
Virasakdi Chongsuvivatwong

# Analisa Data Epidemiologi

## Menggunakan R dan Epicalc

Alih Bahasa : Zurnila Marli Kesuma

Epidemiology Unit
Prince of Songkla University
THAILAND

# Analisa Data Epidemiologi
## Menggunakan R dan Epicalc

Virasakdi Chongsuvivatwong

Alih Bahasa : Zurnila Marli Kesuma

Epidemiology Unit
Prince of Songkla University
**THAILAND**

# Kata Pengantar

Analisa data sangatlah penting dalam riset epidemiologi. Kapasitas fasilitas komputasi yang semakin meningkat, menggerakkan seni keilmuan bidang epidemiology menuju kesamaan arah dengan kemajuan bidang komputasi. Dewasa ini, banyak sekali paket statistik yang digunakan secara meluas oleh para epidemiolog di seluruh dunia. Bagi Negara maju, biaya untuk perangkat lunak bukanlah suatu persoalan besar. Akan tetapi, bagi negara berkembang, biayanya sering terlalu besar. Beberapa peneliti di negara-negara berkembang akhirnya menggunakan perangkat lunak duplikat atau bajakan.

Paket perangkat lunak yang bebas biaya tersedia dalam jumlah yang terbatas, baik dalam jumlahnya maupun dalam kesiapan penggunaannya. EpiInfo, misalnya, bebas biaya dan dapat digunakan un tuk data entri dan analisa data sederhana.. Tetapi, bagi analisa data yang lebih canggih paket tersebut memiliki banyak kekurangan dan keterbatasan di banyak aspek. Sebagai contoh, paket ini tidak layak untuk manipulasi data dalam kajian longitudinal. Fasilitas paket ini untuk analisa regresi tidak dapat mengatasi pengukuran berulang dan permodelan multi level. Fasilitas penampilan grafisnya juga sangat terbatas.

Sebuah perangkat lunak yang menjanjikan dan relatif baru serta tersedia secara cuma-cuma adalah R. Didukung oleh para ahli statistik terkemuka di seluruh dunia, R memiliki hampir semua yang dibutuhkan seorang analis data epidemiologi. Namun, sulit untuk belajar dan menggunakannya bila dibandingkan dengan paket statistik yang sama untuk analisis data epidemiologi seperti Stata. Tujuan buku ini adalah untuk menjembatani kesenjangan tersebut dengan membuat R menjadi mudah dipelajari bagi para peneliti dari negara berkembang dan juga untuk mempromosikan penggunaannya.

Pengalaman saya selama lebih dari dua puluh tahun dalam pembelajaran epidemiologi khususnya mengajar analisis data. Terinspirasi oleh semangat filosofi perangkat lunak *open source*, saya telah berusaha keras mengeksplorasi potensi dan penggunaan R. Selama empat tahun, saya telah mengembangkan paket add-on untuk R yang memungkinkan peneliti baru menggunakan

perangkat lunak ini secara menyenangkan.. Lebih dari 20 bab catatan kuliah dan latihan-latihan dipersiapkan bersama dengan dataset yang mempersiapkan pembaca belajar secara mandiri.

Didukung oleh WHO, TDR dan Thailand Research Fund, saya juga menjalankan sejumlah lokakarya untuk perangkat lunak ini di negara berkembang seperti Thailand, Myanmar, Korea Utara, Maladewa dan Bhutan, dimana R dan Epicalc sangat diterima. Dengan pengalaman ini, saya dengan ini mengusulkan bahwa penggunaan software ini harus didukung oleh para peneliti epidemiologi, terutama bagi mereka yang tidak mampu membeli paket perangkat lunak komersial yang mahal.

R adalah sebuah lingkungan yang dapat menangani dataset secara bersamaan. Pengguna mendapatkan akses ke variabel dalam setiap dataset baik dengan menyalinnya ke path pencarian atau dengan memasukkan nama dataset sebagai awalan**.** Ketika membuat variabel atau memodifikasi yang sudah ada, tanpa awalan nama dataset, variabel baru diisolasi dari dataset induk nya. Jika awalan adalah pilihan, data asli berubah tapi salinan dalam path pencarian tidak berubah**.** Hati-hati pengguna harus menghapus salinan dalam path pencarian dan recopy dataset baru ke dalamnya. Prosedur dalam aspek ini agak janggal. Jika tidak rapi akhirnya akan berakhir dengan salinan terlalu banyak dalam path pencarian overloading sistem atau akan membingungkan si analist untuk memastikan di mana variabel sebenarnya terletak.

Epicalc menyajikan solusi konsep bagi pekerjaan umum di mana analis data bekerja pada satu dataset pada suatu waktu dengan hanya menggunakan beberapa perintah. Dalam Epicalc pengguna hampir dapat menghilangkan perlunya menspesifikasikan dataset dan dapat menghindari overloading dari jalur pencarian dengan sangat efektif dan efisien. Selain itu, merapikan memori sangatlah mudah untuk dilakukan, Epicalc memudahkan pula untuk mengenali variabel dengan mengadopsi label variabel atau deskripsi yang telah dibuat dari perangkat lunak lain, seperti SPSS atau Stata, atau secara lokal disiapkan oleh Epicalc itu sendiri.

R memiliki fungsi grafik yang sangat powerful sehingga pengguna harus menghabiskan waktu untuk mempelajarinya. Epicalc memanfaatkan kekuatan ini dengan memproduksi plot distribusi yang baik secara otomatis setiap kali satu variabel diringkas. Suatu rincian dari variabel pertama dengan variabel kategori kedua juga sederhana dan hasil grafisnya secara otomatis ditampilkan.

Strategi grafik otomatis ini juga diterapkan pada tabulasi satu arah dan tabulasi dua arah. Deskripsi variabel dan label atau kategori nilai sepenuhnya tereksploitasi dengan grafik deskriptif.

Fungsi epidemiologi tambahan yang ditambahkan oleh Epicalc termasuk perhitungan ukuran sampel, tabulasi pemadanan 1: n (n dapat bervariasi), kappa statistik, menggambar kurva ROC dari tabel atau dari hasil regresi logistik, plot populasi piramida dari usia dan jenis kelamin dan ikuti lanjut plot.

R memiliki beberapa fungsi pemodelan regresi canggih seperti regresi logistik multinomial, regresi logistik ordinal, analisis kelangsungan hidup dan multi-level pemodelan. Dengan menggunakan tabel Epicalc dari odds ratio dan 95% selang kepercayaan, maka naskah sederhana dapat dipindahkan ke dalam dokumen naskah dengan hanya memerlukan sedikit modifikasi.

Meskipun penggunaan Epicalc menunjukkan cara kerja yang berbeda dengan **R** yang konvensional, instalasi on of Epicalc tidak memberikan efek apapun terhadap setiap fungsi yang tersedia dan atau yang baru yang ada di R. Fungsi-fungsi di Epicalc hanyalah untuk meningkatkan efisiensi analisi data dan membuat R menjadi lebih mudah digunakan.

Buku ini intinya tentang mempelajari R dengan penekanan pada Epicalc. Para pembaca seharusnya memiliki latar belakang dalam dasar-dasar penggunaan computer. Dengan R, Epicalc dan data set yang disediakan, para pengguna harus mampu untuk mengikuti setiap konsep pembelajaran data manajemen, teori statistika yang terkait dan berlatih analisis data serta membuat grafik dengan baik.

Dalam empat bab pertama diperkenalkan konsep R dan penanganan sederhana elemen-elemen dasar seperti skalar, vektor, matriks, array dan data frames. Bab 5 membahas tentang eksplorasi data sederhana. Variabel tanggal dan waktu didefinisikan di dalam Bab 6 dan investigasi wabah dibahas secara mendalam dalam Bab 7. Statistik deskriptif dah tabulasi satu arah secara otomatis disertai dengan grafiknya, sehingga hampir tidak mungkin ada informasi penting yang terlupakan. Akhirnya, plot waktu untuk paparan dan penyakit diplot dengan serangkaian command yang diperlihatkan. Bab 8 melanjutkan investigasi lanjutan untuk memeriksa wabah tabulasi dua arah. Berbagai jenis kajian tentang resiko, seperti risk ratio dan *protective efficacy*, dianalisa secara numeric dan grafik.

Bab 9 menjangkau analisa dari suatu dataset untuk menangani tingkat asosiasi atau odds ratios. Tabulasi bertingkat, Mantel-Haenzsel odds ratio, dan uji homogenitas dijelaskan secara detail. Semua hasilnya dilengkapi dengan plot yang simultan.. Dengan grafik-grafik tersebut, konsep pembauran menjadi semakin mudah dipahami.

Sebelum meneruskan lebih jauh, pembaca dapat pembaca memiliki latihan menyeluruh tentang , *data cleaning* dan manipulasi data yang standar dalam Bab 10. Scatter plots, regersi linier sederhana dan analisi varians dibahas dalam Bab 11. Scatter plot bertingkat untuk memperjelas konsep pembauran dan interaksi variable keluaran yang kontinu diberikan di Bab 12. Model kelengkungan (Curvilinear) didiskusikan di Bab 13. Model linier diperluas ke *generalized linear* di Bab 14.

Untuk variable keluaran yang biner ,Bab 15 memperkenalkan regresi logistic dengan perbandingan tambahan dengan *stratified cross-tabulation* dipelajari di Bab 9. Konsep *matched case control study* didiskusikan di Bab 16 tabulasi untuk pemadanan 1:1 and 1:n. Akhirnya, regresi logistic bersyarat diterapkan. Bab 17 memperkenalkan regresi logistik polytomus menggunakan *case-control study* dimana satu tipe case dibandingkan dengan dua tipe grup control. Regesi logistic ordinal diterapkan untuk keluaran yang diinginkan.dalam Bab 18.

Untuk studi cohort, dengan paparan kelompok datasets, Regresi Poisson digunakan di Bab 19. Regresi Extra-Poisson untuk *overdispersion* juga didiskusikan. Diskusi juga menyertakan permodelan dengan distribusi negative binomial error. Multi-level modelling and longitudinal data analisis didiskusikan di Bab 20.

Untuk studi cohort dengan individual follow-up times, analisa survival didiskusikan di Bab21 dan Cox proportional hazard model diperkenalkan di Bab 22. Pada Bab 23 fokusnya adalah menganalisa dataset tentang sikap, yang banyak digunakan dalam ilmu-ilmu sosial. Bab 24 berkaitan dengan langkah-langkah menghitung ukuran sampel dan teknik dokumentasi yang harus dikuasai oleh professional data analissi dibahas di Bab 25.

Beberapa saran dan strategi penanganan data berukuran besar dibahas di Bab 26. Buku ini diakhiri dengan peragaan perintah `tableStack`,yang secara yang secara dramatis memperpendek dan merapikan penyusunan tabel dengan teknik khusus *copy* dan *paste* ke dalam naskah.

Pada akhir setiap bab beberapa referensi diberikan untuk bacaan lebih lanjut. Kebanyakan bab juga diakhiri dengan beberapa soal untuk berlatih. Solusi untuk soal-soal tersebut diberikan pada akhir buku.

## Warna

Dianggap bahwa pembaca buku ini akan secara teratur berlatih perintah-perintah (commands) dan melihat hasilnya di layar. Penjelasan di dalam teks, kadang-kadang dengan menggambarkan warna dari grafik yang muncul dalam warna hitam dan putih.di dalam buku ini. (alasannya murni untuk mengurangi biaya cetak). Akan tetapi, dalam versi elektroniknya , ditampilkan versi yang berwarna.

## Penjelasan bentuk-bentuk yang digunakan dalam buku ini.

**MASS** Paket R atau library
**`Attitudes`** R dataset
`plot` Fungsi di R
*summ* Fungsi di Epicalc (huruf miring)
'`abc`' Object di R
'pch' Argument dalam suatu fungsi
'saltegg' Variable di dalam suatu data frame
"**data.txt**" Suatu file data dalam disk

# Daftar Isi

BAB 1

# Penggunaan R

Pada bab ini difokuskan pada penggunan utama **R**, meliputi instalasi, bagaimana menggunakan help, sintaks perintah **R** dan dokumentasi tambahan. Ingat pula bahwa buku ini ditulis untuk pengguna Windows, namun **R** juga bekerja pada system operasi yang lain.

## Instalasi

**R** terdistribusi dibawah bentuk GNU General Public License. Software tersebut secara bebas tersedia untuk penggunaan dan berdistribusi dibawah bentuk license ini. Versi **R 3.2.0** dan Epicalc beserta dokumentasinya dapat diunduh dengan mengetikkan perintah berikut pada R console.

- Install.packages ("epicalc" , repos = http://medipe.psu.ac.th/epicalc")

Hal pertama untuk menginstal **R** adalah masuk ke website CRAN dan pilih system operasi yang sesuai pada bagian atas layar. Untuk pengguna Windows klik pada link Windows dan ikuti link pada subdirektori base . Dalam halaman ini anda dapat mengunduh file pengaturan untuk Windows yaitu R-2.6.1-win32.exe. Klik link tersebut dan tekan tombol "Save".

File set-up **R** berukuran sekitar 30Mb. Untuk menjalankan instalasi double-click pada file ini dan ikuti instruksi selanjutnya. Setelah instalasi, icon shortcut **R** akan tampil pada layar desktop. Klik kanan pada icon tersebut untuk mengubah start-up propertiesnya. Gantikan folder default 'Start in' dengan folder anda sendiri. Folder ini merupakan tempat dimana **R** akan bekerja. Anda dapat membuat lebih dari satu ikon shortcut dengan folder start-in yang berbeda untuk setiap pekerjaan yang akan dilakukan.

Misalkan pekerjaan yang berhubungan dengan buku ini akan disimpan dalam sebuah folder yang dinamakan 'C:\RWorkplace'. The 'Properties' of the icon should have the 'Start in:' text box filled with 'C:\RWorkplace' (tidak boleh menggunakan tanda quote ' dan '. tanda tersebut digunakan dalam buku ini untuk mengindikasikan objek atau nama teknis)

**R** mendeteksi bahasa utama suatu sistem operasi pada komputer dan coba gunakan kotak menu dan dialog dalam bahasa tersebut. Sebagai contoh, jika anda menjalankan **R** dalam Windows XP menggunakan bahasa China, kotak menu dan dialog akan muncul dalam bahasa china. Karena buku ini ditulis awal dalam bahasa Inggris maka disarankan bahasa yang digunakan adalah bahasa Inggris sehingga respon pada computer pengguna akan serupa dengan yang dibuku. Pada tab 'Shortcut' ikon **R,** tambahkan Language=en pada akhir 'Target'. Masukkan spasi sebelum kata 'Language'.


Maka kotak teks 'Target' untuk versi ikon R-2.6.1 adalah :

**"C:\Program Files\R\R-2.6.1\bin\Rgui.exe" Language=en**

Agar bisa menggunakan buku ini secara efisien, editor teks khusus seperti *Crimson Editor* or *Tinn-R* harus diinstall pada computer anda. Sebagai tambahan, paket Epicalc harus diinstal dan dimuat.

## Text Editors

### Crimson Editor

Software ini dapat diinstal secara konvensional seperti software lainnya yaitu dengan menjalankan file setup.exe dan mengikuti instruksi.

Crimson Editor memiliki beberapa fitur menarik yang dapat membantu pengguna saat bekerja menggunakan **R**. Fitur tersebut sangat bagus digunakan untuk editing script atau command files menggunakan berbagai program software seperti file C++, PHP dan HTML. Nomor baris dapat ditampilkan dan tanda kurung buka dan tutup dapat disesuaikan. Fitur ini penting karena fitur tersebut umumnya digunaka dalam bahasa perintah **R**.

Proses instalasi dan set-up untuk Crimson Editor akan dijelaskan pada Chapter 25.

### *Tinn-R*

Tinn-R mungkin merupakan editor teks file terbaik untuk digunakan dalam konjungsi terhadap program **R**. Editor ini didesain secara khusus untuk bekerja dalam file script **R**. Sebagai tambahan untuk sintaks kode **R**, Tinn-R dapat berinteraksi dengan **R** menggunakan menu dan bar tool tertentu. Ini berari bahwa bagian perintah dapat disorot dan dikirim ke dalam **R** console (sourced) dengan sekali klik pada tombol. Tinn-R dapat diunduh dari internet pada situs www.sciviews.org/Tinn-R.

## Memulai Program R

Setelah pemodifikasian start-up properties dari ikon **R**, double-klik pada ikon **R** yang ada pada desktop. Program dimulai dan output berikut akan ditampilkan pada **R** console.

```
R version 2.6.1 (2007-11-26)
Copyright (C) 2007 The R Foundation for Statistical
  Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
```

```
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
>
```

Output diatas dihasilkan dari **R** version 2.6.1, yang dirilis pada 26 November 2007. Paragraf kedua mendeklarasikan dan menjelaskan secara singkat mengenai garansi dan perizinan. Paragraf ketiga memberikan informasi mengenai kontibutor dan bagaimana mencari **R** dalam publikasi. Paragraf keempat menyarankan beberapa perintah untuk penggunaan pertama yang dapat dicoba.

Dalam buku ini, perintah **R** bermula dengan tanda ">", serupa dengan yang ditunjukkan dalam jendela **R** console. Setelah memulai dengan ">" maka ketik perintah yang akan digunakan. Dalam dokumen ini, baik perintah **R** dan baris output akan ditampilkan dalam font Courier New sedangkan teks penjelasnya dalam font Times New Roman. Perintah Epicalc ditampilka dalam *italic*, sedangkan perintah standar **R** ditampilkan dalam font normal.

Sebagai latihan, tutup program sebelumnya. Klik tanda pada sudut atas kanan jendela program atau ketik perintah berikut pada **R** console:

```
> q()
```

Kotak dialog akan menampilkan pertanyaan "Save workspace image?" dengan tiga pilihan: "Yes", "No" dan "Cancel". Pilih "Cancel" untuk melanjutkan. Jika anda memilih "Yes", dua file baru akan terbentuk dalam folder pekerjaan anda. Perintah sebelumnya yang telah diketik pada **R** console akan disimpan kedalam file yang dinamakan '.Rhistory' sementara workspace yang baru saja digunakan disimpan ke dalam file yang disebut ".**Rdata**". Ingat bahwa dua file ini tidak memiliki awalan. Dalam bahasan penghitungan selanjutnya, saat **R** dimulai pada folder ini, image dari pekerjaan sebelumnya akan diperoleh kembali secara otomatis bersama dengan history perintah. Penggunaan **R** selanjutnya dalam cara ini (berhenti bekerja dan menyimpan *image workspace*) dihasilkan dalam dua file ini akan bertambah besar. Umumnya salah satunya akan memulai **R** lagi setiap waktu sehingga disarankan untuk selalu memilih "No" saat akan menyimpan *workspace*. Sebagai alternatif dapat diketik:

```
> q("no")
```

untuk keluar tanpa menyimpan *image workspace* dan mencegah kotak pesan dialog muncul.

Ingat bahwa sebelum keluar dari **R** anda dapat menyimpan *image workspace* dengan mengetik:

```
> save.image("C:/RWorkplace/myFile.RData")
```

dimana 'myFile' merupakan nama file anda. Kemudian saat anda keluar dari **R** pilih "No".

## R libraries & packages

**R** dapat didefinisikan sebagai sebuah lingkungan fungsi dimana banyak teknik statistika klasik dan modern dapat diterapkan. Beberapa dari teknik ini dibangun sebagai dasar lingkungan **R**, tetapi kebanyakan disediakan dalam bentuk *packages* (paket). Sebuah *packages* merupakan koleksi sederhana dari fungsi, dataset beserta dokumentasinya. *Library* merupakan koleksi *package* yang khusus memuat direktori tunggal dalam komputer.

Terdapat sekitar 25 *packages* tersedia dalam **R** (dinamakan *packages* "standard" atau "recommended") dan banyak lainnya juga tersedia diseluruh website CRAN. Hanya 7 dari paket ini dimuat kedalam memori saat **R** dieksekusi. Untuk mengetahui paket mana yang baru saja dimuat ke dalam memori, anda dapat mengetik:

```
> search()
[1] ".GlobalEnv"        "package:methods"   "package:stats"
[4] "package:graphics"  "package:grDevices" "package:utils"
[7] "package:datasets"  "Autoloads"         "package:base"
```

Daftar diatas merupakan pencarian pintas **R**. Saat **R** diperintahkan melakukan suatu pekerjaan, daftar akan mencari objek khusus untuk dikerjakan. Pertama, akan dicari kedalam lingkungan global '.GlobalEnv'. Ini akan selalu menjadi posisi pertama pencarian. Jika **R** tidak bisa menemukan apa yang diinginkan, maka akan dicari pada posisi kedua pencarian, dalam kasus ini "package:methods" dan seterusnya. Fungsi lainnya yang termasuk dalam satu *loaded packages* selalu tersedia selama sesi **R**.

## *Epicalc package*

Epicalc *package* dapat diunduh pada R console dengan mengetikkan

- Install.packages ("epicalc" , repos = http://medipe.psu.ac.th/epicalc")

Epicalc merupakan kepanjangan dari 'Epidmiological calculator'.

*Package* Epicalc di *update* dari waktu ke waktu. Nomor versi berada pada akhiran. Sebagai contoh "**epicalc_2.6.1.6.zip**" merupakan file biner yang digunakan pada system operasi Windows dan versi Epicalc 2.6.1.6. Versi terbaru dibuat untuk mengatur error pada program, untuk memperbaiki fitur fitur fungsi yang ada dan untuk menambahkan fungsi fungsi baru.

File **"epicalc_*version*.zip"** ('version' meningkat sesuai waktu) merupakan file *compressed* yang penuh dengan kumpulan paket Epicalc untuk sistem operasi Windows. Instalasi paket ini harus dilakukan didalam **R** itu sendiri. Umumnya hanya ada satu sesi instalasi yang dibutuhkan kecuali anda ingin mengganti paket lama dengan paket baru dan dengan nama yang sama. Anda juga harus reinstall paket ini jika anda menginstal versi terbaru **R**.

Untuk menginstal Epicalc, klik 'Packages' pada menu bar dibagian atas jendela. Pilih 'Install packages from local zip files...". Saat jendela navigasi muncul, *browse* untuk menemukan filenya dan buka file tersebut.

Instalasi berhasil jika tampil dalam bentuk berikut:

```
> utils:::menuInstallLocal()
package 'epicalc' successfully unpacked and MD5 sums checked
updating HTML package descriptions
```

Sekarang instalasi sudah selesai; bagaimanapun fungsi dalam Epicalc belum tersedia sebelum perintah berikut dijalankan:

```
> library(epicalc)
```

Ingat untuk menggunakan huruf kecil. Saat *console* menerima perintah, kita dapat mengetahui bahwa perintah telah diterima. Sebaliknya error atau peringatan akan dilaporkan.

Peringatan umumnya merupakan laporan dari sebuah ketidaksesuain. Peringatan ini kebanyakan tidak terlalu serius. Ini berarti bahwa sebuah objek (biasanya sebuah fungsi) dengan nama yang sama sudah ada sebelumnya dalam lingkungan kerja **R**. Pada kasus ini, **R** akan memberi prioritas kepada objek yang

lebih dahulu dari pencarian pintas. Perintah diatas harus diketik setiap kali sesi baru **R** dijalankan.

### *Meng-update packages*

Kapanpun versi baru *packagei* dirilis maka disarankan untuk di *update* dengan menghapus versi yang lama dan memuat versi baru. Untuk menghapus the Epicalc *package*, anda dapat mengetik perintah berikut pada **R** console:

```
> detach(package:epicalc)
```

Setelah mengetik perintah diatas, anda dapat menginstal versi paket yang baru seperti yang telah dijelaskan sebelumnya. Jika terdapat masalah, anda dapat keluar dan memulai kembali **R**.

### *RProfile.site*

Saat **R** dijalankan, secara bersamaan perintah dieksekusi dalam file**"RProfile.site"**, yang berlokasi di folder 'C:\Program Files\R\R-2.7.0\etc'. Ingat untuk menggantikan versi **R** dengan yang telah anda install. Dengan memasukkan perintah library(epicalc) dalam file **"RProfile.site"** kapanpun **R** dijalankan, *package* Epicalc secara otomatis dimuat dan siap untuk digunakan. Anda dapat mengedit file ini dan menyisipkan perintah diatas.

File **"RProfile.site"**ditampilkan dalam bentuk:

```
library(epicalc)

# Things you might want to change
# options(papersize="a4")
# options(editor="notepad")
# options(pager="internal")

# to prefer Compiled HTML help
# options(chmhelp=TRUE)

# to prefer HTML help
# options(htmlhelp=TRUE)
```

## Bantuan On-line

Bantuan online sangat berguna saat menggunakan software, khususnya untuk

pengguna pemula. Belajar secara otodidak juga sangat mungkin dari bantuan online **R**, meskipun dengan beberapa kesulitan. Penulis menyarankan kombinasi penggunaan buku ini sebagai tutorial dan bantuan online sebagai referensi manual.

Dokumentasi bantuan online tampil dalam tiga versi berbeda pada **R**. Versi *default* menampilkan bantuan informasi pada jendela terpisah didalam **R**. Format ini ditulis dalam bahasa sederhana yang dapat dibaca oleh **R** dan dapat pula dikonversikan ke dalam LaTeX yang digunakan untuk menghasilkan cetakan manual. Versi lainnya yang dapat diatur dalam file "**Rprofile.site**" merupakan bantuan HTML (htmlhelp=TRUE) dan kumpulan bantuan HTML (chmhelp=TRUE). Versi terakhir merupakan spesifikasi Windows dan jika dipilih, dokumentasi bantuan akan muncul dalam *viewer* bantuan Windows. Setiap format bantuan memiliki kelebihan tersendiri dan anda bebas memilih format mana yang anda inginkan.

Untuk permulaan, ketik

```
> help.start()
```

Sistem akan membuka *web browser* dari menu utama **R**. Pengenalan **R** merupakan bab yang harus dibaca oleh semua pengguna **R** dan harus dicoba. Bahasan menarik lainnya adalah 'Packages'. Klik untuk melihat paket apa yang anda punya. Jika paket Epicalc sudah selesai dimuat, kemudian namanya akan muncul pada daftar. Klik 'Epicalc' untuk melihat daftar fungsi yang tersedia. Klik masing masing fungsi dan anda akan melihat bantuan untuk setiap fungsi. Informasi ini dapat pula diperoleh dengan mengetik 'help(*myFun*)' pada **R** console, dimana *'myFun*' merupakan nama fungsi. Untuk mendapatkan bantuan pada fungsi 'help' anda dapat mengetik,

```
> help(help)
```

Atau dengan mengetik

```
> ?help
```

Untuk pencarian dengan kata kunci yang anda inginkan

```
> help.search("...")
```

Gantikan titik titik diatas dengan kata kunci yang ingin dicari. Fungsi ini juga membolehkan anda untuk mencari dengan kata kunci lebih dari satu. Anda dapat menggunakan ini untuk menyaring pertanyaan saat anda mendapatkan

banyak hasil pencarian.

Pengguna sering ingin mengetahui bagaimana untuk mendapatkan fungsi analisis statistika lainnya yang tidak terdapat dalam paket yang baru diinstal. Caranya adalah melakukan pencarian pada website CRAN menggunakan fitur 'search' pada sisi kiri halaman web dan Google akan melakukan pencarian dalam CRAN. Hasilnya akan lebih banyak dan berguna. Selanjutnya pengguna dapat memilih website yang diinginkan untuk pembelajaran lebih jauh.

Sekarang ketik

```
> search()
```

Anda dapat melihat "package:epicalc" pada daftar. Jika paket Epicalc belum dimuat maka fungsi yang ada didalamnya tidak tersedia untuk digunakan.

Memiliki paket Epicalc dalam pencarian berarti bahwa kita dapat menggunakan seluruh fungsi yang ada dalam paket. Paket lainnya dapat dipanggil saat akan digunakan. Contohnya, paket **survival** dibutuhkan dalam analisis survival. Kita akan membahas ini pada bab selanjutnya.

Urutan pencarian pintas terkadang menjadi penting. Untuk pengguna Epicalc direkomendasikan bahwa penambahan *library* seharusnya dilakukan lebih awal pada saat memulai sesi **R**, misalnya sebelum membaca dan melampirkan *data frame*. Hal ini untuk memastikan bahwa dataset aktif akan berada pada posisi kedua pencarian. Detail lebih lanjut akan dijelaskan pada Chapter 4.

## Penggunaan R

Tujuan dasar **R** adalah menampilkan perhitungan sederhana.

```
> 1+1
[1] 2
```

Saat anda mengetik '1+1' dan menekan tombol <Enter>, **R** akan menampilkan hasil perhitungan yaitu 2.

Untuk akar kuadrat 25:

```
> sqrt(25)
[1] 5
```

Kata didepan tanda kurung buka disebut 'function'. *Entity* didalam tanda kurung disebut 'argument'. Maka pada contoh diatas, 'sqrt()' adalah sebuah fungsi dan argument nya adalah 25 maka akan menghasilkan nilai 5.

Untuk mencari nilai *e*:

```
> exp(1)
[1] 2.718282
```

Eksponen 1 sama dengan nilai *e*, yaitu sekitar 2.7. Secara serupa, nilai eksponensial dari -5 atau $e^{-5}$ adalah

```
> exp(-5)
  [1] 0.006738
```

## *Sintaks perintah R*

**R** akan menghitung jika perintah yang dimasukkan benar. Misalnya jika jumlah kurung tertutup lebih sedikit dari kurung terbuka dan ketika tombol <Enter> ditekan, baris baru akan dimulai dengan tanda '+', mengindikasikan bahwa **R** menunggu kelengkapan perintah. Setelah jumlah kurung tertutup berjumlah sama dengan kurung buka, perhitungan dilakukan dan hasilnya akan muncul.

```
> log(3.8
+ )
[1] 1.335001
```

Bagaimanapun, jika jumlah kurung tertutup melebihi kurung terbuka, hasil berupa sintaks error atau gramatikal komputer.

```
> log(3.2))
Error: syntax error
```

## *Objek R*

Pada perhitungan sederhana diatas, hasil segera ditampilkan pada layar dan tidak disimpan. Untuk menampilkan perhitungan dan menyimpan hasil dalam sebuah objek, ketik:

```
> a = 3 + 5
```

Kita dapat memeriksa apakah tugas ini telah sukses dengan mengetik nama objek yang baru:

```
> a
```

```
[1] 8
```

Secara sederhana, tugas ditulis dengan cara berikut.

```
> a <- 3 + 5
> a
[1] 8
```

Untuk pengguna pemula, tidak terdapat perbedaan penggunaan antara = dan <- . Perbedaan diaplikasikan pada level pemograman **R** dan tidak akan didiskusikan disini. Meskipun <- sedikit lebih rumit untuk diketik dari pada =, teknik sebelumnya lebih diutamakan untuk menghindari kebingungan dengan operator perbandingan (==). Ingat bahwa tidak ada spasi antara komponen dari operator penugasan <-.

Sekarang buatlah objek kedua yang disebut 'b' yaitu akar kuadrat dari 36.

```
> b <- sqrt(36)
```

Maka jumlahkan kedua objek tersebut.

```
> a + b
[1] 14
```

Kita juga dapat menghitung nilai pada sisi kiri dan menempatkan hasil pada objek baru yang disebut 'c' pada sisi kanan, menggunakan operator penugasan ->.

```
> a + 3*b -> c
> c
[1] 26
```

Bagaimanapun, perintah tersebut tidak bekerja.

```
> a + 3b -> c
Error: syntax error
```

**R** tidak mengenal '3b'. Simbol * dibutuhkan sebagai tanda perkalian.

Nama objek dapat berisi lebih dari satu huruf.

```
> xyx <- 1
> xyx
[1] 1
```

Hal yang tidak masuk akal dapat juga diketik pada **R** console seperti:

```
> qwert
```

```
Error: Object "qwert" not found
```

Apa yang diketik diatas benar secara sintaks tetapi 'qwert' bukan fungsi yang dikenal dan bukan objek yang terdefinisi.

Sebuah titik dapat pula digunakan sebagai pembatas nama objek.

```
> baht.per.dollar <- 40
> baht.per.dollar
[1] 40
```

Pada akhirnya, saat suatu objek diketik pada **R** console, program akan mencoba menampilkan nilai dari objek tersebut. Jika tanda = atau <- atau -> bertemu, nilai akan disimpan pada objek sebelah kiri = dan <- atau sebelah kanan ->.

## *Objek karakter atau string*

Karakter atau string berarti alphanumerik atau huruf. Contoh dibawah terdiri dari nama orang beserta alamat. Tipe objek ini tidak dapat digunakan untuk kalkulasi. Nomor telepon dan kode pos juga objek string

```
> A <- "Prince of Songkla University"

> A
[1] "Prince of Songkla University"
```

**R** merupakan program sensitif, jadi 'A' tidak sama dengan 'a'.

```
> a
[1] 8

> A
[1] "Prince of Songkla University"
```

## *Memasukkan komentar pada baris perintah*

Pada buku ini, seperti kebanyakan dokumen pemograman lainnya, penulis biasanya menyisipkan beberapa komentar sebagai bagian dokumentasi untuk mengingatkan penulis atau menunjukkan beberapa isu khusus kepada pembaca.

**R** mengabaikan kata yang diikuti simbol #. Tetapi sebuah kalimat dapat digunakan sebagai perintah. Contoh:

Mengabaikan setiap kata menggunakan dengan lambang #. Selanjutnya,

kalimat-kalimat berikut dapat digunakan untuk perintah berikut: Misalnya:

```
> 3*3 = 3^2   # This gives a syntax error
> 3*3 == 3^2  # This is correct syntax-wise.
> 3*2 == 3^2  # Correct syntax but the result is FALSE
```

### *Logical: TRUE dan FALSE*

Pada perintah berikut:

```
> 3*3 == 3^2
[1] TRUE
```

tetapi

```
> 3*2 == 3^2
[1] FALSE
```

Ingat bahwa kita membutuhkan dua tanda “sama dengan” untuk memeriksa kesamaan, tetapi hanya satu untuk penugasan.

```
> 3*2 < 3^2
[1] TRUE
```

### *Logical connection menggunakan & (logical 'and')*

Kedua objek TRUE dan FALSE merupakan objek logical. Koneksi lebih dari satu objek akan dihasilkan dalam TRUE atau FALSE. Jika keseluruhan TRUE, maka hasil akhir adalah TRUE, contoh:

```
> TRUE & TRUE
[1] TRUE
```

Kombinasi FALSE dengan objek logical lainnya selalu FALSE.

```
> TRUE & FALSE
[1] FALSE

> FALSE & FALSE
[1] FALSE
```

Ingat bahwa

```
> (FALSE & TRUE) == (TRUE & FALSE)
[1] TRUE
```

Tanpa menggunakan tanda kurung, penghitungan dilakukan dari kiri ke kanan.

```
> FALSE & TRUE == TRUE & FALSE
[1] FALSE
```

### *Logical connection dengan | (logical 'or')*

Jenis koneksi ini mencari semua objek TRUE.

```
> TRUE | TRUE
[1] TRUE

> TRUE | FALSE
[1] TRUE

> 3*3 == 3^2 | 3*2 == 3^2
[1] TRUE
```

### *Nilai TRUE dan FALSE*

Secara numerik, TRUE sama dengan 1 dan FALSE bernilai 0.

```
> TRUE == 1
[1] TRUE

> FALSE == 0
[1] TRUE

> (3*3 == 3^2) + (9 > 8)
[1] 2
```

Setiap nilai dalam tanda kurung adalah TRUE yang bernilai . Penambahan dua objek TRUE bernilai 2. Bagaimanapun,

```
> 3*3 == 3^2 + 9 > 8
Error: syntax error in "3*3 == 3^2 + 9 >"
```

Ini didasarkan pada urutan rumit dari sebuah operasi. Meskipun demikian, selalu lebih baik jika menggunakan tanda kurung untuk spesifikasi urutan pasti penghitungan.

Mari kita tinggalkan **R** untuk sementara. Pilih "Yes" untuk pertanyaan: "Save work space image?".

Ingat bahwa menjawab "No" adalah tanggapan yang lebih baik dalam buku ini seperti penulis sarankan ketik

```
> q("no")
```

untuk mengakhiri setiap sesi **R**. Menjawab "Yes" disini hanya sebagai latihan pemahaman konsep dari *workspace images*, yang akan dijelaskan pada Chapter 2.

## *Referensi*

IntrroductiontoR.ISBN3-900051-12-7.
R Language Definition. ISBN 3-900051-13-5.

Kedua referensi diatas dapat didownload dari website CRAN.

## Latihan

Soal 1.

Rumus untuk mencari ukuran sampel dalam survey deskriptif adalah

$$n = \frac{1.96^2}{\delta^2}\pi(1-\pi)$$

Dimana n adalah ukuran sampel, $\pi$ adalah prevalensi di dalam populasi (janganlah dibingungkan oleh konstan dan *pi)*dan $\delta$ adalah setengah dari lebar 95% selang kepercayaan (presisi).

Hitung ukuran sampel yangdibutuhkan jika prevalensi diduga menjadi 30% dari populasi dan 95% selang kepercayaan tidak akan melebihi dari taksiran prevalensi sebesar 5%.

Soal 2.

Ubahlah prevalensi diatas menjadi 5% dan andaikan masing-masing sisi dari 95% selang kepercayaan tidak akan melebihi dari taksiran prevalensi sebesar 2%.

Soal 3.

Istilah 'logit' dinotasikan dengan 'log{P/(1-P)}' dimana P adalah resiko atau prevalensi dari suatu penyakit. Hitung dari nilai prevalensi berikut: 1%, 10%, 50%, 90% and 100%.

# BAB 2

# Vektor

Pada bab sebelumnya kita telah memperkenalkan kalkulasi sederhana dan bagaimana menyimpan hasilnya. Pada bab ini, kita akan belajar sekilas mengenai isu yang lebih kompleks.

## *History dan saved objek*

Diluar **R**, jika anda menguji folder pekerjaan, anda dapat melihat dua file baru yaitu **".Rdata"** yang merupakan lingkungan pekerjaan yang disimpan dari sesi terakhir **R** dan **".Rhistory"** yang merekam semua perintah dari sesi **R** sebelumnya. **".Rdata"** adalah file biner dan hanya dikenali oleh program **R** sedangkan **".Rhistory**" adalah file teks dan dapat diedit menggunakan berbagai teks editor seperti Notepad, Crimson Editor atau Tinn-R.

Bukalah **R** dari ikon desktop. Anda akan melihat ini pada baris terakhir:

```
[Previously saved workspace restored]
```

Ini berarti bahwa **R** telah menyimpan perintah dari sesi **R** sebelumnya (atau history) dan objek disimpan pada sesi ini. Tekan tanda panah diatas dan anda akan melihat perintah sebelumnya (keduanya benar dan tidak benar). Tekan <Enter> pada perintah; hasilnya akan muncul jika anda melanjutkan bekerja

pada sesi sebelumnya.

```
> a
[1] 8

> A
[1] "Prince of Songkla University"
```

Kedua nya '`a`' dan '`A`' disimpan pada sesi sebelumnya.

**Catatan**:

---

Image yang disimpan pada sesi sebelumnya hanya mengandung sebuah objek dalam '.GlobalEnv', yang merupakan posisi pertama dalam pencarian pintas. Keseluruhan pencarian tidak disimpan. Misalkan, sebarang library secara manual dimuat setiap kali kita memulai **R** (dari pengaturan file "**Rprofile.site**" yang kita modifikasi pada bab sebelumnya). Meskipun dalam aturan seperti ini, tanpa memperhatikan apakah workspace image telah disimpan atau tidak pada sesi sebelumnya, Epicalc akan selalu ada pada pencarian pintas).

Jika anda akan menghapus objek dari lingkungan dan history, keluar dari **R** tanpa menyimpan. Kembali pada folder 'start in' dan hapus dua file **".Rhistory**" dan **".Rdata"**. kemudian mulai kembali **R** dan tidak terdapat pesan yang mengindikasikan penyimpanan dari workspace sebelumnya dan tidak ada perintah sebelumnya.

## Rangkaian (Concatenation)

Objek dengan tipe yang sama, misalkan numerik dengan numerik, string dengan string, dapat diurutkan. Pada kenyataannya, sebuah vektor merupakan sebuah objek yang diurutkan, tidak ada lagi pembagian objek dengan tipe yang sama.

Untuk mengurutkan, fungsi '`c()`' digunakan minimal satu objek *atomized* sebagai argument. Buatlah sebuah vektor sederhana dengan bilangan bulat 1,2 dan 3 sebagai elemennya.

```
> c(1,2,3)
[1] 1 2 3
```

Vektor ini memiliki tiga elemen: 1, 2 and 3. Tekan tanda panah atas untuk menunjukkan kembali perintah ini dan ketik tanda panah kanan untuk

menampilkan hasil dalam objek baru yang dinamakan 'd'.

```
> c(1,2,3) -> d
> d
```

Lakukan beberapa perhitungan dengan objek 'd' dan perhatikan hasilnya.

```
> d + 4
> d - 3
> d * 7
> d / 10
> d * d
> d ^ 2
> d / d
> d == d
```

Sebagai tambahan, sebuah kata dapat digunakan untuk menciptakan vektor string.

```
;> B <- c("Faculty of Medicine","Prince of Songkla
  University")
> B
[1] "Faculty of Medicine"       "Prince of Songkla
  University"
```

## *Vektor bilangan*

Terkadang pengguna ingin membuat sebuah vektor bilangan dengan pola tertentu. Perintah berikut akan membuat sebuah vektor bilangan bulat dari 1 sampai 10.

```
> x <- 1:10; x
 [1]  1  2  3  4  5  6  7  8  9 10
```

Untuk lima kali pengulangan bilangan 13:

```
> rep(13, times=5)
[1] 13 13 13 13 13
```

Fungsi '`rep`' digunakan untuk menggandakan nilai suatu argument. Untuk mengurutkan bilangan -1 hingga 11 dengan selang 3 bilangan, ketik:

```
> seq(from = -1, to = 11, by = 3)
[1] -1  2  5  8 11
```

Pada kasus ini `seq` merupakan fungsi dengan tiga argumen 'from', 'to' dan 'by'. Fungsi ini dapat dieksekusi dengan paling kurang dua parameter, , 'from' dan

'to', karena parameter 'by' mempunyai nilai *default* 1 (atau -1 jika to' lebih kecil dari 'from').

```
> seq(10, 23)
 [1] 10 11 12 13 14 15 16 17 18 19 20 21 22 23

> seq(10, -3)
 [1] 10  9  8  7  6  5  4  3  2  1  0 -1 -2 -3
```

Urutan dari argumen 'from', 'to' dan 'by' diasumsikan jika kata diabaikan.

Saat eksplisiti diberikan, urutan bisa dirubah

```
> seq(by=-1, to=-3, from=10)
```

Aturan urutan argument ini diaplikasikan dalam semua fungsi. Untuk lebih jelas tentang `seq` gunakan fitur *help*.

## Subsetting vektor dengan indeks vektor

Pada kebanyakan contoh, hanya sebuah bagian tertentu dari vektor yang digunakan. Mari asumsikan kita memiliki sebuah vektor dengan bilangan 3 hingga 100 dengan selang 7 bilangan. Berapa nilai pada bilangan kelima?

```
> seq(from=3, to=100, by=7) -> x
> x
 [1]  3 10 17 24 31 38 45 52 59 66 73 80 87 94
```

Kenyataannya vektor tidak berakhir pada bilangan 100 tetapi 94 karena untuk step yang lebih jauh akan melebihi 100.

```
> x[5]
[1] 31
```

Bilangan yang ada didalam kurung siku '`[]`' disebut subscript. Subscript menotasikan posisi atau pemilihan vektor utama. Pada kasus ini, nilai pada posisi kelima dari vektor 'x' adalah 31. Jika ditanya posisi keempat, keenam dan ketujuh maka ketik:

```
> x[c(4,6,7)]
[1] 24 38 45
```

Ingat bahwa pada contoh ini, objek yang ada didalam subscript bisa menjadi

sebuah vektor, maka fungsi urutan `c` dibutuhkan disini untuk memenuhi sintaks **R**. Sintaks berikut tidak bisa diterima oleh **R:**

```
> x[4,6,7]
Error in x[4, 6, 7] : incorrect number of dimensions
```

Untuk memilih '`x`' dengan empat elemen pertama diabaikan, ketik:

```
> x[-(1:4)]
 [1] 31 38 45 52 59 66 73 80 87 94
```

Tanda negative didepan vektor subscript mendefinisikan penghapusan elemen '`x`' yang berkorespondensi dengan posisi yang dispesifikasi oleh vektor subscript.

Dengan cara yang sama vektor string dapat pula di subscript.

```
> B[2]
[1] "Prince of Songkla University"
```

## *Menggunakan vektor subscript untuk pemilihan subset*

Sebuah vektor merupakan kumpulan bilangan atau huruf (string). Penggunaan syarat dalam hasil subscript dalam sebuah subset vektor utama. Sebagai contoh, untuk memilih bahkan untuk satu bilangan vektor '`x`', ketik:

```
> x[x/2 == trunc(x/2)]
[1] 10 24 38 52 66 80 94
```

Fungsi `trunc` berguna untuk memotong atau menghilangkan desimal. Syarat bahwa '`x`' dibagi oleh 2 sama dengan nilai tanpa desimal adalah benar iff (if dan only if) '`x`' adalah bilangan genap. Hasil serupa dapt pula diperoleh dengan menggunakan fungsi `subset`.

```
> subset(x, x/2==trunc(x/2))
```

Hanya jika bilangan ganjil yang dipilih, maka operator pembanding dapat dirubah secara sederhana menjadi `!=` yang berarti 'not equal'.

```
> subset(x, x/2!=trunc(x/2))
[1]  3 17 31 45 59 73 87
```

Untuk memilih elemen '`x`' yang lebih besar dari 30:

```
> x[x>30]
 [1] 31 38 45 52 59 66 73 80 87 94
```

## *Fungsi berhubungan dengan vektor manipulasi*

**R** dapat menghitung statistik vektor dengan sangat mudah.

```
> fruits <- c(5, 10, 1, 20)
> summary(fruits)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    1.0     4.0     7.5     9.0    12.5    20.0
> sum(fruits)
[1] 36
```

Terdapat 36 total buah yang ada.

```
> length(fruits)   # number of different types of fruits
[1] 4
> mean(fruits)          # mean of number of fruits
[1] 9
> sd(fruits)     # standard deviation
[1] 8.205689
> var(fruits)    # variance
[1] 67.33333
```

## *Vektor non-numeric*

Mari buat sebuah vektor string yang dinamakan '`person`' dan terdiri dari 11 elemen.

```
> person <- c("A","B","C","D","E","F","G","H","I","J","K")
```

Sebagai alternatif dan lebih praktis:

```
> person <- LETTERS[1:11]
```

Sekarang periksa kelas objek '`person`' dan '`fruits`'

```
> class(person)
[1] "character"

> class(fruits)
[1] "numeric"
```

Tipe karakter digunakan untuk penyimpanan nama individual. Untuk menyimpan jenis kelamin, mula-mula kode numerik diberikan: 1 untuk laki-laki, 2 untuk wanita.

```
> sex <- c(1,2,1,1,1,1,1,1,1,1,2)
> class(sex)
```

```
[1] "numeric"
> sex1 <- as.factor(sex)   # Creating sex1 from sex
```

Fungsi `as.factor` memaksa objek '`sex`' menjadi sebuah *factor* yang merupakan tipe data kategori dalam **R**.

```
> sex1
 [1] 1 2 1 1 1 1 1 1 1 1 2
Levels: 1 2
```

Ada dua level jenis kelamin.

```
> class(sex1)
[1] "factor"
> is.factor(sex)
[1] FALSE
> is.factor(sex1)
[1] TRUE
```

Sekarang coba uji label 'sex1'.

```
> levels(sex1) <- c("male", "female")
```

Level '`sex`' merupakan vektor string.

```
> sex1
 [1] male   female male   male   male   male   male
   [8] male   male   male   female
Levels: male female
```

## *Mengurutkan element vektor*

Buatlah vektor usia dengan 11 elemen.

```
> age <- c(10,23,48,56,15,25,40,21,60,59,80)
```

Untuk mengurutkannya:

```
> sort(age)
 [1] 10 15 21 23 25 40 48 56 59 60 80
```

Fungsi `sort` mengurutkan elemen vektor dalam urutan ascending (dari nilai terkecil ke nilai terbesar). Bagaimanapun vektor aslinya tidah dirubah.

```
> median(age)
[1] 40
```

Nilai median sebesar 40. Untuk mendapatkan nilai kuantil, gunakan fungsi

quantile.

```
> quantile(age)
  0%  25%  50%  75% 100%
10.0 22.0 40.0 57.5 80.0
```

Jika argumen lainnya diabaikan (*default*) persentil ke-0, ke-25, ke-50, ke-75 dan ke-100 juga ditampilkan.

```
> quantile(age, prob = .3)
30%
 23
```

## *Membuat faktor dari sebuah vektor*

Vektor kelompok usia dapat diperoleh dengan menggunakan fungsi `cut`.

```
> agegr <- cut(age, breaks=c(0,15,60,100))
```

Fungsi ini menciptakan 3 kelompok berbeda yang kita namakan 'children', 'adults' dan 'elderly'. Ingat bahwa argumen minimum dan maksimum dalam fungsi `cut` merupakan batas paling luar.

```
> is.factor(agegr)
[1] TRUE
> attributes(agegr)
$levels
[1] "(0,15]"   "(15,60]"  "(60,100]"
$class
[1] "factor"
```

Objek '`agegr`' merupakan sebuah faktor dengan level yang ditunjukkan diatas. Kita bisa memerikas korenpondensi antara '`age`' dan '`agegr`' menggunakan fungsi `data.frame`, yang mengombinasikan (tetapi tidak disimpan) 2 variabel dalam data frame dan menampilkan hasilnya. Lebih jelas mengenai fungsi ini akan dipaparkan pada Chapter 4.

```
> data.frame(age, agegr)
   age    agegr
1   10   (0,15]
2   23  (15,60]
3   48  (15,60]
4   56  (15,60]
5   15   (0,15]
6   25  (15,60]
```

```
7   40  (15,60]
8   21  (15,60]
9   60  (15,60]
10  59  (15,60]
11  80 (60,100]
```

Bisa diperhatikan bahwa orang ke-5 yang berusia 15 tahun diklasifikasi ke dalam kelompok pertama dan orang ke-9 yang berusia 60 tahun berada pada kelompok kedua. Label untuk setiap kelompok menggunakan kurung siku pada akhir argument yang menandakan bahwa bilangan terakhir termasuk dalam grup (termasuk pemotongan). Untuk memperoleh tabel frekuensi kelompok usia, ketik:

```
> table(agegr)
agegr
  (0,15]  (15,60] (60,100]
       2        8        1
```

Terdapat dua anak-anak, delapan dewasa dan seorang lanjut usia.

```
> summary(agegr) # same result as the preceding command
> class(agegr)
[1] "factor"
```

Vektor kelompok usia merupakan sebuah faktor atau vektor kategori. Vektor tersebut dapat ditransformasikan menjadi vektor numerik sederhana menggunakan fungsi '`unclass`' yang akan dijelaskan lebih detail pada Bab 3.

```
> agegr1 <- unclass(agegr)
> summary(agegr1)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   2.000   2.000   1.909   2.000   3.000

> class(agegr1)
[1] "integer"
```

Variabel kategori, misalnya jenis kelamin, ras dan agama harus selalu difaktorkan. Kelompok usia pada contoh ini adalah sebuah faktor walaupun kelompok tersebut memiliki pola terurut. Pendeklarasian vektor sebagai faktor sangatlah penting, khususnya saat menggunakan analisis regresi yanga akan didiskusikan pada bab berikutnya.

Nilai yang tidak terkelompok dari sebuah faktor digunakan pada saat nilai numerik (atau bilangan bulat) dari faktor dibutuhkan. Misalnya, jka kita memiliki

sebuah dataset yang mengandung variabel 'sex' , dikelompokkan sebagai faktor, dan kita ingin menggambar scatter plot dimana warna titik plot diklasifik'col = unclass(sex)'. Untuk detail akan dijelaskan pada bab selanjutnya.

## Data hilang (Missing values)

Data hilang biasanya muncul dari data yang tidak dikumpulkan. Sebagai contoh, data usia yang tidak ada berasal dari seseorang yang tidak memberikan informasi mengenai usianya. Di dalam **R**, data hilang dinotasikan dengan 'NA', kepanjangan dari 'Not Available'. Perhitungan yang melibatkan NA akan menghasilkan output dalam NA pula.

```
> b <- NA
> b * 3
[1] NA

> c <- 3 + b
> c
[1] NA
```

Karena contoh data hilang berada dalam deret vektor, ketik perintah berikut:

```
> height <- c(100,150,NA,160)
> height
[1] 100 150  NA 160

> weight <- c(33, 45, 60,55)
> weight
[1] 33 45 60 55
```

Diantara empat sampel dalam contoh ini, keseluruhan bobot (berat badan) tersedia tetapi ada satu data hilang pada tinggi badan.

```
> mean(weight)
[1] 48.25

> mean(height)
[1] NA
```

Kita dapat memperoleh rataan tinggi badan tetapi rataan tinggi badan tidak dapat diperoleh, meskipun panjang vektor tersedia.

```
> length(height)
[1] 4
```

Untuk mendapatkan rataan dari semua elemen yang ada, elemen NA harus dihilangkan.

```
> mean(height, na.rm = TRUE)
[1] 136.6667
```

Argumen 'na.rm' berarti 'not available (value) removed', dan sama halnya saat NA dihilangkan dengan menggunakan fungsi `na.omit()`.

```
> length(na.omit(height))
[1] 3

> mean(na.omit(height))
[1] 136.6667
```

Dengan demikian `na.omit` merupakan fungsi independen yang mengabaikan nilai hilang dari objek argument. 'na.rm' adalah argumen internal dari statistik deskriptif sebuah vektor.

## Latihan

Masalah 1.

Hitung nilai $1^2 + 2^2 + 3^2 ... + 100^2$

Masalah 2.

Misalkan '`y`' merupakan sebuah deret bilangan bulat antara 1 hingga 1,000.

Hitung jumlah dari elemen '`y`' yang merupakan kelipatan 7.

Masalah 3.

Berikut adalah tinggi badan (cm) dan berat badan (kg) dari 10 anggota keluarga:

```
            ht wt
Niece      120 22
Son        172 52
GrandPa    163 71
Daughter   158 51
Yai        153 51
GrandMa    148 60
Aunty      160 50
Uncle      170 67
Mom        155 53
Dad        167 64
```

Buatlah sebuah vektor yang dinamakan 'ht' yang berkorespondensi dengan 11 anggota keluarga. Buatlah nama anggota keluarga menjadi nama atribut vektor.

Buat sebuah vektor yang disebut 'wt' yang berkorespondensi dengan berat badan anggota keluarga.

Hitung body mass index (BMI) setiap orang dimana

BMI = berat badan/ tinggi badan$^2$.

Periksa siapa saja yang memiliki nilai BMI tertinggi dan terendah serta hitung standar deviasi BMI.

BAB 3

# Array, Matriks, dan Tabel

Data riil untuk analisis jarang merupakan sebuah vektor. Dalam banyak kasus, data tersebut merupakan dataset yang terdiri dari banyak baris atau catatan dan banyak kolom atau variabel. Dalam **R**, dataset ini disebut kerangka data (*data frames*). Sebelum membahas mengenai *data frames*, mari kita pelajari hal sederhana seperti array, matriks dan tabel. Mendapatkan konsep serta keahlian dalam mengatasi tipe objek seperti ini akan memberi kesempatan pengguna untuk memanipulasi data dengan efektif dan efisien pada masa mendatang.

## Array

Array secara umum dapat diartikan sebagai sesuatu yang tersusun dengan baik. Dalam matematika dan komputasi, sebuah array terdiri dari nilai nilai yang tersusun dalam baris dan kolom. Sebuah dataset dasarnya merupakan sebuah array. Kebanyakan paket statistik ditangani hanya dengan satu dataset atau array pada saat tertentu. **R** memiliki kemampuan khusus untuk mengatasi beberapa array dan dataset secara bersamaan. Hal ini karena **R** merupakan

program berorientasi objek. Selain itu, **R** menginterpretasikan baris dan kolom dalam cara yang sangat sama.

## *Merubah vektor menjadi array*

Biasanya sebuah vektor tidak memiliki dimensi.

```
> a <- (1:10)
> a
 [1]  1  2  3  4  5  6  7  8  9 10

> dim(a)
NULL
```

Merubah vektor menjadi array sangatlah sederhana. Hanya dengan mendeklarasikan atau memasukkan kembali dimensi jumlah baris dan kolom seperti,

```
> dim(a) <- c(2,5)
> a
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    3    5    7    9
[2,]    2    4    6    8   10
```

Angka angka dalam kurung siku merupakan subskrip baris dan kolom. Perintah command `dim(a) <- c(2,5)` mengubah vektor menjadi array yang terdiri dari 2 baris dan 5 kolom.

## *Ekstraksi sel, kolom, baris dan subarray menggunakan subscripts*

Sementara mengekstraksi sebuah himpunan bagian vektor hanya membutuhkan satu komponen angka atau vektor, maka array membutuhkan dua komponen. Masing-masing elemen array dapat dibedakan dengan memberi nama array mengikuti dua subscript dipisahkan oleh koma dalam kurung siku. Subscript yang pertama menyatakan pemilihan baris, subscript yang kedua menyatakan pemilihan kolom. Masing-masing baris dan kolom dapat diekstraksi dengan menghilangkan salah satu komponen, tetapi koma tetap ada.

```
> a[1,]     # for the first row and all columns of array 'a'
> a[,3]     # for all rows of the third column
> a[2,4]    # extract 1 cell from the 2nd row and 4th column
> a[2,2:4]  # 2nd row, from 2nd to 4th columns
```

Kedua perintah `a[,]` dan `a[]` memilih semua baris dan kolom dari '`a`' . Array tersebut juga memiliki 3 dimensi.

```
> b <- 1:24
> dim(b) <- c(3,4,2)          # or b <- array(1:24, c(3,4,2))

> b
, , 1
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12

, , 2
     [,1] [,2] [,3] [,4]
[1,]   13   16   19   22
[2,]   14   17   20   23
[3,]   15   18   21   24
```

Nilai pertama dari dimensi menyatakan jumlah baris, kemudian jumlah kolom dan yang terakhir adalah jumlah tingkatan.

Elemen dari array tiga dimensi dapat diekstraksi dengan cara yang sama.

```
> b[1:3,1:2,2]
     [,1] [,2]
[1,]   13   16
[2,]   14   17
[3,]   15   18
```

Dalam kenyataannya, sebuah array dapat memiliki dimensi yang lebih tinggi, tetapi untuk kebanyakan analisis epidemiologi jarang digunakan atau dibutuhkan.

## *Menggabungkan Vektor*

Berbeda dengan merubah vektor, sebuah array dapat dibuat dari penggabungan vektor, baik dengan kolom ( menggunakan fungsi `cbind`) atau dengan baris (menggunakan fungsi `rbind`) . Mari kembali pada vektor buah.

```
> fruits <- c(5, 10, 1, 20)
```

Anggap orang kedua membeli buah tetapi dalam jumlah yang berbeda dengan orang pertama.

```
> fruits2 <- c(1, 5, 3, 4)
```

Untuk menggabungkan '`fruits`' dengan '`fruits2`', dimana kedua vektor tersebut memiliki ukuran yang sama, ketik:

```
> Col.fruit <- cbind(fruits, fruits2)
```

Kita dapat member nama untuk baris dari array tersebut:

```
> rownames(Col.fruit) <-
  c("orange","banana","durian","mango")
> Col.fruit
        fruits fruits2
orange       5       1
banana      10       5
durian       1       3
mango       20       4
```

Atau, penggabungan dapat dilakukan menggunakan baris.

```
> Row.fruit <- rbind(fruits, fruits2)
> colnames(Col.fruit) <-
  c("orange","banana","durian","mango")
> Row.fruit
        orange banana durian  mango
fruits       5     10      1     20
fruits2      1      5      3      4
```

## *Transposisi sebuah array*

Transposisi array berarti menukar baris dan kolom sebuah array. Pada contoh diatas, '`Row.fruits`' merupakan transposisi dari '`Col.fruits`' dan sebaliknya. Transposisi array diperoleh dengan menggunakan fungsi `t`.

```
> t(Col.fruit)
> t(Row.fruit)
```

## *Statistika dasar mengenai array*

Total jumlah buah-buahan yang dibeli kedua orang diatas diperoleh dengan mengetik:

```
> sum(Col.fruit)
```

Dan total jumlah nenas diperoleh dari:

```
> sum(Col.fruit[2,])
```

Untuk memasukkan statistik deskriptif masing-masing pembeli, ketik:

```
> summary(Col.fruit)
```

Dan untuk memasukkan statistik deskriptif masing-masing jenis buah :

```
> summary(Row.fruit)
```

Sekarang misalkan ditambahkan '`fruits3`' tetapi tidak ada jenis buah yang ditambah.

```
> fruits3 <- c(20, 15, 3, 5, 8)
> cbind(Col.fruit, fruits3)
        fruits fruits2 fruits3
orange       5       1      20
banana      10       5      15
durian       1       3       3
mango       20       4       5
Warning message:
number of rows of result is not a multiple of vector length
  (arg 2) in: cbind(Col.fruit, fruits3)
```

Ingat bahwa elemen terakhir '`fruits3`' dihilangkan sebelum ditambahkan.

```
> fruits4 <- c(1,2,3)
> cbind(Col.fruit, fruits4)
        fruits fruits2 fruits4
orange       5       1       1
banana      10       5       2
durian       1       3       3
mango       20       4       1
Warning message:
number of rows of result is not a multiple of vector length
  (arg 2) in: cbind(Col.fruit, fruits4)
```

Ingat bahwa '`fruits4`' ukurannya lebih pendek dibanding panjang argument vektor pertama. Pada situasi seperti ini **R** secara otomatis memakai kembali elemen vektor yang lebih pendek, memasukkan elemen pertama dari '`fruits4`' kedalam baris keempat, dengan pemberitahuan.

## *String arrays*

Sama halnya dengan vektor, sebuah array dapat mengandung karakter string.

```
> Thais <- c("Somsri", "Daeng", "Somchai", "Veena")

> dim(Thais) <- c(2,2); Thais
     [,1]     [,2]
[1,] "Somsri" "Somchai"
[2,] "Daeng"  "Veena"
```

Ingat bahwa elemen digabungkan secara kolom, bukan baris, secara berurutan.

## *Array "implicit" dari dua vector yang sama panjang*

Dua vektor, khususnya yang memiliki panjang yang sama, dapat berhubungan satu sama lain tanpa penggabungan formal.

```
> cities <- c("Bangkok","Hat Yai","Chiang Mai")
> postcode <- c(10000, 90110, 50000)
> postcode[cities=="Bangkok"]
[1] 10000
```

Ini memberikan hasil yang sama sebagai

```
> subset(postcode, cities=="Bangkok")
[1] 10000
```

Untuk vektor tunggal, banyak cara untuk mengidentifikasi urutan elemen tertentu. Misalnya, untuk menemukan indeks "Hat Yai" dalam vektor kota, empat perintah berikut semuanya memberikan hasil yang serupa.

```
> (1:length(cities))[cities=="Hat Yai"]
> (1:3)[cities=="Hat Yai"]
> subset(1:3, cities=="Hat Yai")
> which(cities=="Hat Yai")
```

Ingat bahwa saat sebuah vektor karakter digabungkan dengan vektor numerik, vektor numerik dipaksakan kedalam vektor karakter, karena semua elemen array harus memiliki tipe yang sama.

```
> cbind(cities,postcode)
     cities       postcode
[1,] "Bangkok"    "10000"
[2,] "Hat Yai"    "90110"
[3,] "Chiang Mai" "50000"
```

## Matriks

Matriks merupakan array dimensi dua. Matriks memiliki beberapa sifat dan operasi matematika yang digunakan dibelakang statistika komputasi seperti analisis faktor, model linear umum dan sebagainya.

Pengguna paket statistik tidak perlu menggunakan matriks secara langsung tetapi beberapa hasil analsis dalam bentuk matriks, keduanya ditampilkan pada layar yang mudah dilihat dan tersembunyi sebagai *returned object* yang dapat digunakan nanti. Untuk tujuan latihan, kita akan menguji kovarian matriks, yang merupakan *returned object* dari analsis regresi pada bab selanjutnya.

## Tabel

Sebuah tabel merupakan array yang menekankan pada hubungan antara nilai-nilai dalam sel. Biasanya, sebuah tabel merupakan hasil dari analisis, misalnya tabulasi silang antara variabel kategori (menggunakan fungsi `table`).

Misalkan enam orang pasien yang terdiri dari laki-laki, perempuan, perempuan, laki-laki, perempuan dan perempuan dating ke sebuah klinik. Jika kode 1 (laki-laki) dan kode 2 (perempuan), maka untuk membuatnya dalam **R** ketik:

```
> sex <- c(1,2,2,1,2,2)
```

Sama halnya jika kita mengkategori umur pasien muda atau tua dan tiga pasien pertama umurnya masih muda, dua pasien berikutnya sudah tua dan pasien terakhir masih muda, dan kode untuk ketiga klasifikasi ini adalah 1 (muda) dan 2 (tua), sehingga kita bisa membuatnya di **R** dengan mengetik:

```
> age <- c(1,1,1,2,2,1)
```

Misalkan juga bahwa pasien ini pernah mengunjungi klinik satu hingga enam kali, secara berurutan.

```
> visits <- c(1,2,3,4,5,6)

> table1 <- table(sex, age); table1
   age
sex 1 2
  1 1 1
  2 3 1
```

Ingat bahwa `table1` memberikan hitungan setiap kombinasi dari vektor `sex` dan `age` sementara '`table2`' (dibawah) memberikan jumlah angka kunjungan berdasarkan empat kombinasi berbeda dari `sex` dan `age`.

```
> table2 <- tapply(visits, list(Sex=sex, Age=age), FUN=sum)

> table2
   Age
Sex  1 2
  1  1 4
  2 11 5
```

Untuk memasukkan rataan setiap tipe kombinasi:

```
> tapply(visits, list(Sex=sex, Age=age), FUN=mean)

   Age
Sex     1 2
  1 1.000 4
  2 3.667 5
```

Meskipun '`table1`' memiliki kelas *table*, kelas '`table2`' tetap sebuah matriks. Dapat juga diubah sederhana menggunakan fungsi `as.table`.

```
> table2 <- as.table(table2)
```

## *Ringkasan sebuah tabel dari ringkasan suatu array (Summary of table vs summary of array)*

Dalam **R**, menggunakan ringkasan fungsi kedalam sebuah tabel menunjukkan uji kebebasan chi squared.

```
> summary(table1)
Number of cases in table: 6
Number of factors: 2
Test for independence of all factors:
        Chisq = 0.375, df = 1, p-value = 0.5403
        Chi-squared approximation may be incorrect
```

Sebaliknya, menerapkan `summary` untuk array non tabel menghasilkan statistik deskriptif untuk setiap kolom.

```
> is.table(Col.fruits)
[1] FALSE
```

```
> summary(Col.fruits)
       fruits        fruits2
 Min.   : 1.0   Min.   :1.00
 1st Qu.: 4.0   1st Qu.:2.50
 Median : 7.5   Median :3.50
 Mean   : 9.0   Mean   :3.25
 3rd Qu.:12.5   3rd Qu.:4.25
 Max.   :20.0   Max.   :5.00

> fruits.table <- as.table(Col.fruits)
> summary(fruits.table)
Number of cases in table: 49
Number of factors: 2
Test for independence of all factors:
        Chisq = 6.675, df = 3, p-value = 0.08302
        Chi-squared approximation may be incorrect

> fisher.test(fruits.table)
        Fisher's Exact Test for Count Data
data:  fruits.table
p-value = 0.07728
alternative hypothesis: two.sided
```

## Lists

Sebuah array membuat semua sel dari kolom dan baris yang berbeda untuk menjadi tipe yang sama. Jika sebarang sel merupakan sebuah karakter maka semua sel akan dipaksa menjadi sebuah karakter. Berbeda dengan daftar. Hal tersebut dapat menjadi sebuah campuran dari berbagai tipe objek yang berbeda dipaksakan menjadi satu kesatuan. Campuran tersebut dapat berupa vektor, array, tabel atau tipe objek lainnya.

```
> list1 <- list(a=1, b=fruits, c=cities)
> list1
$a
[1] 1

$b
[1]  5 10  1 20

$c
```

```
[1] "Bangkok"   "Hat Yai"   "Chiang Mai"
```

Ingat bahwa argument fungsi `list` terdiri dari serangkaian objek baru yang diberi nilai dari obyek yang sudah ada. Saat ditampilkan, setiap nama baru dimulai dengan tanda dollar $.

Pembuatan daftar bukan suatu pekerjaan umum dalam analisis data. Bagaimanapun, sebuah daftar terkadang dibutuhkan dalam argument beberapa fungsi.

Menghilangkan objek dari memori computer juga membutuhkan daftar argument untuk fungsi `rm`.

```
> rm(list=c("list1", "fruits"))
```

Hal ini equivalen dengan

```
> rm(list1); rm(fruits)
```

Sebuah daftar dapat juga dikembalikan dari hasil analisis, tetapi muncul dalam kelas khusus.

```
> sample1 <- rnorm(10)
```

Ini menghasilkan sampel dari 10 nomor dari distribusi normal.

```
> qqnorm(sample1)
```

Fungsi `qqnorm` memplotkan kuantil sample, atau mengurut nilai observasi bersama dengan kuantil teoritik, atau nilai ekspektasi yang berhubungan jika data berdistribusi normal sempurna. Ini digunakan sekedar demonstrasi fungsi `list.`

```
> list2 <- qqnorm(sample1)
```

Penyimpanan hasil ke dalam objek disebut '`list2`'.

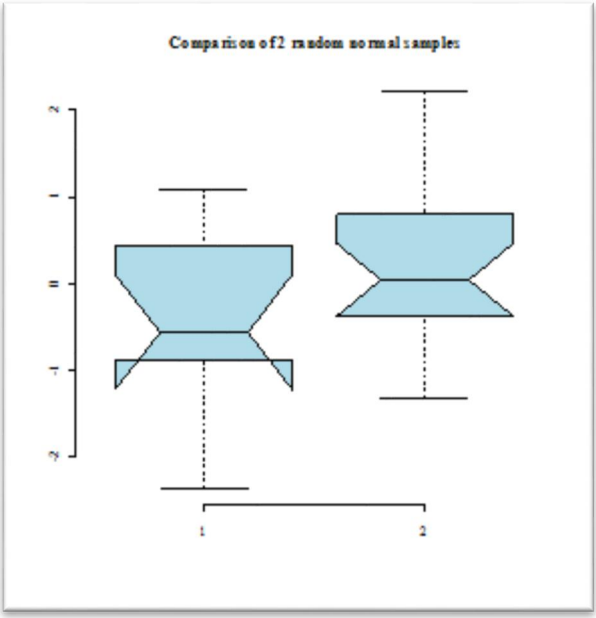
```
> list2
$x
 [1]  0.123 -1.547 -0.375  0.655  1.000  0.375 -0.123
 [8] -1.000 -0.655  1.547

$y
 [1] -0.4772 -0.9984 -0.7763  0.0645  0.9595 -0.1103
 [7] -0.5110 -0.9112 -0.8372  2.4158
```

Perintah `qqnorm(sample1)` digunakan metode grafik untuk menguji

normalitas. Sementara hal itu menghasilkan grafik dalam layar, juga mengembalikan daftar koordinat x dan y, yang dapat disimpan dan digunakan untuk kalkulasi lebih lanjut.

Sama halnya, perintah berikut mengembalikan berbagai daftar objek untuk menghasilkan plot boxplot. Lihta halaman bantuan untuk beberapa contoh menarik.

```
> sample2 <- rnorm(20)
> bp <- boxplot(sample1, sample2)> bp
$stats
         [,1]      [,2]
[1,] -2.34570 -1.308507
[2,] -0.89004 -0.372543
[3,] -0.55554  0.046435
[4,]  0.42912  0.803616
[5,]  1.08444  2.208447

$n
[1] 10 20

$conf
         [,1]     [,2]
[1,] -1.21465 -0.36910
[2,]  0.10356  0.46197

$out
numeric(0)

$group
numeric(0)

$names
[1] "1" "2"

> bxp(bp, notch=TRUE, boxfill="lightblue", frame=FALSE,
  outl=FALSE, main="Comparison of 2 random normal samples")
```


Comparison of 2 random normal samples
2
1
0
-1
-2
1
2

## Latihan

Soal1.

Tunjukkan cara sederhana untuk membuat array di bawah ini:

```
     [,1][,2][,3][,4][,5][,6][,7][,8][,9][,10]
[1,]    1   2   3   4   5   6   7   8   9   10
[2,]   11  12  13  14  15   16  7  18  19   20
```

Soal 2.

Lakukan proses *extract* dari array di atas untuk kolom bernomor ganjil.

Soal 3.

*Cross-tabulation* antara status suatu penyakit dan paparan dugaan (putative) adalah sebagai berikut:

| | Diseased | Non-diseased |
|---|---|---|
| ***Exposed*** | 15 | 20 |
| ***Non-exposed*** | 30 | 22 |

Buatlah tabel dengan **R** dan tampilkan uji chi-squared dan uji exact Fisher.

BAB 4

# Data Frames

Dalam bab sebelumnya, contoh diberikan pada array dan daftar. Dalam bab ini, frame data akan menjadi fokus utama. Untuk sebagian besar peneliti, hal ini kadang-kadang disebut dataset. Namun, dataset lengkap dapat berisi lebih dari satu frame data. Dimana berisi data nyata yang peneliti harus bekerja dengan sebagian besarnya.

## *Perbandingan dari array dan data frame*

Banyak aturan yang digunakan untuk array juga berlaku untuk data frame. Sebagai contoh, struktur utama data frame terdiri dari kolom (atau variabel) dan baris (atau catatan). Aturan untuk mengikat subscripting, kolom atau baris dan pemilihan subset dalam array secara langsung diterapkan pada data frame.

Data frame bagimanapun sedikit lebih rumit dari array. Semua kolom dalam array dipaksa untuk menjadi karakter jika hanya satu sel berupa karakter. Sebuah data frame, di sisi lain, dapat memiliki kelas yang berbeda dari kolom. Sebagai contoh, data frame dapat terdiri dari kolom 'idnumber', yang merupakan numerik dankolom 'nama', yang merupakan karakter.

Data frame juga dapat memiliki atribut tambahan. Sebagai contoh, setiap variabel dapat memiliki deskripsi variabel panjang. Faktor dalam data frame sering memiliki 'level' atau label nilai. Atribut ini dapat ditransfer dari dataset asli dalam format lain seperti Stata atau SPSS. Mereka juga dapat dibuat di R selama analisis.

## *Mendapatkan data frame dari file teks*

Data dari berbagai sumber dapat dimasukkan dengan menggunakan banyak program perangkat lunak yang berbeda. Mereka dapat ditransfer dari satu format ke format yang lain melalui format file ASCII. Pada Windows, file teks adalah file ASCII yang paling umum, biasanya memiliki ekstensi **"txt".** Ada beberapa file lain dalam format ASCII, termasuk file **". R"** , file perintah yang dibahas dalam bab 25.

Data dari banyak program perangkat lunak dapat diekspor atau disimpan sebagai file ASCII. Dari *Excel*, program spreadsheet yang sangat umum digunakan, data dapat disimpan sebagai format **".csv"** (comma separated values). Ini adalah cara yang mudah untuk menghubungkan antara file spreadsheet *Excel* dan *R.* Cukup buka file Excel dan simpan sebagai (‘save as’) format csv.

Sebagai contoh, misalkan file **"csv1.xls"** yang awalnya sebuah spreadsheet *Excel*. Setelah 'save as' ke dalam format csv, file output disebut **"csv1.csv"**, yang isinya adalah:

```
"name","sex","age"
"A","F",20
"B","M",30
"C","F",40
```

Perhatikan bahwa karakter yang ditutupi dalam tanda kutip dan pembatas (pemisah variabel) adalah koma. Kadang-kadang file mungkin tidak mengandung tanda kutip, seperti dalam file "csv2.csv".

```
name,sex,age
A,F,20
B,M,30
C,F,40
```

Untuk kedua file, perintah R untuk membaca dataset adalah sama.

```
> a <- read.csv("csv1.csv", as.is=TRUE)
```

```
> a
  name sex age
1    A   F  20
2    B   M  30
3    C   F  40
```

Argumen **'as.is'** disetel ke TRUE untuk menyimpan semua variabel seperti mereka. Hal ini belum ditentukan, karakter akan dipaksa menjadi faktor. 'Nama' variabel tidak harus menjadi faktor tapi 'jenis kelamin' yang seharusnya. Karenanya, perintah berikut harus diketik:

```
> a$sex <- factor(a$sex)
```

Catatan pertama bahwa objek 'a' memiliki kelas data frame dan kedua bahwa nama-nama variabel dalam data frame 'a' harus dirujuk menggunakan notasi tanda dolar. Jika tidak, **R** akan memberitahu Anda bahwa objek 'jenis kelamin' tidak dapat ditemukan.

```
> class(a)    # "data.frame"
```

Untuk file dengan spasi (spasi dan tab) sebagai pemisah, seperti dalam file **"data1.txt"**, perintah untuk menggunakannya adalah **read.table.**

```
> a <- read.table("data1.txt", header=TRUE, as.is=TRUE)
```

File **"data2.txt"** adalah ditempatkan dalam bidang format tanpa bidang pemisah.

```
namesexage
1AF20
2BM30
3CF40
```

Untuk membaca sedemikian file, fungsi **read.fwf** lebih disukai. Baris pertama, yang sebagai header, harus dilewati. Lebar dari tiap variabel dan nama kolom harus ditentukan oleh pengguna.

```
> a <- read.fwf("data2.txt", skip=1, width=c(1,1,2),
  col.names = c("name", "sex", "age"), as.is=TRUE)
```

## Entri data dan analisis

Perlakuan di bagian atas dengan menciptakan data frame dengan cara membaca data yang dibuat di luar dari program R, seperti Excel. Hal ini juga memungkinkan untuk memasukkan data secara langsung ke R dengan menggunakan fungsi data.entry. Namun, jika ukuran datanya besar (katakanlah lebih dari 10 kolom dan / atau lebih dari 30 baris), kemungkinan kesalahan yang dilakukan besar dengan spreadsheet atau teks mode entri data. Sebuah software khusus dirancang untuk entri data, seperti Epidata, yang lebih sesuai. Situs web mereka: http://www.epidata.dk. Epidata memiliki fasilitas untuk mengatur kendala berguna seperti cek jangkauan, melompat otomatis dan pelabelan variabel dan nilai-nilai (kode) untuk setiap variabel. Ada transfer langsung antara Epidata dan R (menggunakan 'read.epiinfo') tapi direkomendasikan untuk mengekspor data dari Epidata (menggunakan prosedur ekspor di dalam perangkat lunak itu) ke format Stata dan menggunakan fungsi read.dta untuk membaca dataset ke R. Pengeksporan data ke dalam format Stata mempertahankan banyak atribut dari variabel, seperti label variabel dan deskripsi.

### *Pembersihan memori dan membaca data*

Pada tipe R console:

```
> rm(list=ls())
```

Fungsi **rm** singkatan dari "menghapus". Perintah di atas akan menghapus semua objek dalam ruang kerja. Untuk melihat apa objek sedang dalam jenis ruang kerja:

```
> ls()
character(0)
```

Perintah **ls ()** menunjukkan daftar objek dalam ruang kerja saat ini. Nama (s) dari benda memiliki karakter kelas. Hasil "karakter (0)" berarti bahwa tidak ada benda biasa di lingkungan.

Jika Anda tidak melihat "karakter (0)" dalam output tetapi sesuatu yang lain, itu berarti benda-benda yang tersisa dari sesi R sebelumnya. Ini akan terjadi jika Anda setuju untuk menyimpan gambar ruang kerja sebelum keluar dari R. Untuk menghindari hal ini, hentikan R dan menghapus file "Rdata.", yang terletak di folder kerja Anda, atau mengubah nama itu jika Anda ingin

menjaga ruang kerja dari sesi sebelumnya R.

Atau, untuk menghapus semua objek dalam ruang kerja saat ini tanpa berhenti dari R, adalah dengan mengetik:

```
> zap()
```

Perintah ini akan menghapus semua objek biasa dari memori R. Benda biasa termasuk data frame, vektor, array, dll . Fungsi objek terhindar dari penghapusan.

## Dataset termasuk dalam Epicalc

Kebanyakan paket add-on (penambahan) untuk R berisi dataset yang digunakan untuk demonstrasi dan pengajaran. Untuk memeriksa apakah dataset tersedia di semua paket yang dimuat dalam R, ketik:

```
> data()
```

Anda akan melihat nama dan deskripsi dari beberapa dataset dalam berbagai kemasan, seperti **dataset** dan **epicalc**. Dalam buku ini, sebagian besar contoh menggunakan dataset dari paket Epicalc.

## Membaca dalam data

Mari kita coba untuk memuat sebuah dataset Epicalc.

```
> data(Familydata)
```

Perintah data memuat/memanggil dataset Familydata ke dalam ruang kerja R. Jika tidak ada kesalahan, Anda dapat melihat objek ini di ruangkerja.

```
> ls()
[1] "Familydata"
```

### *Melihat isi data frame*

Jika data frame kecil seperti ini (11 catatan, 6 variabel), cukup ketik nama untuk melihat keseluruhan dataset.

```
> Familydata
   code age  ht wt money sex
1     K   6 120 22     5   F
2     J  16 172 52    50   M
3     A  80 163 71   100   M
4     I  18 158 51   200   F
5     C  69 153 51   300   F
6     B  72 148 60   500   F
7     G  46 160 50   500   F
8     H  42 163 55   600   F
9     D  58 170 67  2000   M
10    F  47 155 53  2000   F
11    E  49 167 64  5000   M
```

Untuk mendapatkan nama-nama variabel (dalam urutan) dari data frame, Anda dapat mengetik:

```
> names(Familydata)
[1] "code"  "age"   "ht"    "wt"    "money" "sex"
```

Fungsi lain yang dapat digunakan untuk mengeksplorasi struktur data adalah **str.**

```
> str(Familydata)
'data.frame':   11 obs. of  6 variables:
 $ code : chr  "K" "J" "A" "I" ...
 $ age  : int  6 16 80 18 69 72 46 42 58 47 ...
 $ ht   : int  120 172 163 158 153 148 160 163 170 155 ...
 $ wt   : int  22 52 71 51 51 60 50 55 67 53 ...
 $ money: int  5 50 100 200 300 500 500 600 2000 2000 ...
 $ sex  : Factor w/ 2 levels "F","M": 1 2 2 1 1 1 1 1 2  ...
=============+=== remaining output omitted =====+===========
```

### *Ringkasan statistik dari data frame*

Sebuah eksplorasi cepat dari dataset adalah mendapatkan ringkasan statistik dari semua variabel. Hal ini dapat dicapai dalam satu perintah.

```
> summary(Familydata)
     code                age              ht
```

```
 Length:11          Min.   : 6.0   Min.   :120
 Class :character   1st Qu.:30.0   1st Qu.:154
 Mode  :character   Median :47.0   Median :160
                    Mean   :45.7   Mean   :157
                    3rd Qu.:63.5   3rd Qu.:165
                    Max.   :80.0   Max.   :172
       wt           money       sex
 Min.   :22.0   Min.   :   5   F:7
 1st Qu.:51.0   1st Qu.: 150   M:4
 Median :53.0   Median : 500
 Mean   :54.2   Mean   :1023
 3rd Qu.:62.0   3rd Qu.:1300
 Max.   :71.0   Max.   :5000
```

Fungsi ringkasan adalah dari perpustakaan dasar. Ini memberikan ringkasan statistik dari setiap variabel. Untuk variabel kontinu seperti 'usia', 'berat', 'ht' dan 'uang', statistik deskriptif non-parametrik seperti minimum, kuartil pertama, median, kuartil ketiga dan maksimum, serta mean (parametrik) akan ditampilkan. Tidak ada informasi tentang standar deviasi atau jumlah observasi. Untuk variabel kategori, seperti 'seks', tabulasi frekuensi ditampilkan. Variabel 'kode' pertama adalah variabel karakter. Karena itu tidak ada ringkasan untuk itu.

Bandingkan hasil ini dengan versi ringkasan statistik menggunakan fungsi **summ** dari paket Epicalc.

```
> summ(Familydata)
Anthropometric and financial data of a hypothetical family
No. of observations = 11
  Var. name Obs.  mean    median s.d.    min.   max.
1 code
2 age       11    45.73   47     24.11   6      80
3 ht        11    157.18  160    14.3    120    172
4 wt        11    54.18   53     12.87   22     71
5 money     11    1023.18 500    1499.55 5      5000
6 sex       11    1.364   1      0.505   1      2
```

Fungsi summ memberikan output lebih ringkas, menunjukkan satu variabel per baris. Jumlah observasi dan deviasi standar yang termasuk dalam laporan menggantikan nilai-nilai kuartil pertama dan ketiga dalam fungsi ringkasan (summary function) asli dari perpustakaan dasar. Statistika deskriptif untuk variabel faktor menggunakan nilai-nilai mereka yang tidak dikelompokkan. Nilai-nilai 'F' dan 'M' untuk variabel 'seks' telah digantikan masing-masing oleh kode

1 dan 2 . Hal ini karena R menafsirkan variabel faktor berupa tingkat, di mana setiap tingkat disimpan sebagai bilangan bulat mulai dari 1 untuk tingkat pertama faktor. Variabel faktor yang tidak dikelompokkan mengubah kategori atau tingkat ke bilangan bulat. Diskusi lebih lanjut tentang faktor akan muncul kemudian.

Dari output di atas statistik yang sama dari variabel yang berbeda dimasukkan ke dalam kolom yang sama. Informasi tentang setiap variabel diselesaikan tanpa ada yang hilang karena jumlah pengamatan semua 11. Minimum dan maksimum yang akan ditampilkan mendekati satu sama lain memungkinkan berbagai variabel untuk dapat ditentukan dengan mudah .

Selain itu, ringkasan statistik untuk setiap variabel yang mungkin dengan kedua pilihan fungsi. Hasilnya mirip dengan ringkasan statistik dari seluruh dataset.Cobalah perintah berikut:

```
> summary(Familydata$age)
> summ(Familydata$age)
> summary(Familydata$sex)
> summ(Familydata$sex)
```

Perhatikan bahwa **summ**, bila diterapkan ke variabel, secara otomatis memberikan output grafis. Ini akan diuji lebih rinci dalam bab-bab selanjutnya.

## *Mengekstrak subset dari data frame*

Sebuah frame data memiliki sistem subscripting yang mirip dengan array. Untuk memilih hanya kolom ketiga dari Familydata, ketik:

```
> Familydata[,3]
 [1] 120 172 163 158 153 148 160 163 170 155 167
```

Ini adalah sama dengan

```
> Familydata$ht
```

Perhatikan bahwa subscripting data frame **Familydata** dengan tanda dolar ($) dan nama variabel hanya akan mengekstrak variabel tersebut. Hal ini karena data frame juga merupakan jenis list (daftar) (lihat bab sebelumnya).

```
> typeof(Familydata)
[1] "list"
```

Untuk mengambil lebih dari satu variabel, kita dapat menggunakan salah satu

nomor indeks dari variabel atau nama. Sebagai contoh, jika kita ingin menampilkan hanya yang 3 catatan pertama 'ht', 'berat' dan 'seks', maka kita dapat mengetikkan:

```
> Familydata[1:3,c(3,4,6)]
   ht wt sex
1 120 22   F
2 172 52   M
3 163 71   M
```

Kita juga bisa mengetikkan :

```
> Familydata[1:3,c("ht","wt","sex")]

   ht wt sex
1 120 22   F
2 172 52   M
3 163 71   M
```

Kondisi dalam subskrip dapat menjadi
kriteria pilihan, seperti memilih perempuan.

```
> Familydata[Familydata$sex=="F",]
   code age  ht wt money sex
1     K   6 120 22     5   F
4     I  18 158 51   200   F
5     C  69 153 51   300   F
6     B  72 148 60   500   F
7     G  46 160 50   500   F
8     H  42 163 55   600   F
10    F  47 155 53  2000   F
```

Perhatikan bahwa ekspresi kondisional harus diikuti dengan tanda koma untuk menunjukkan pilihan semua kolom. Selain itu, dua tanda sama dengan dibutuhkan dalam ekspresi kondisional. Ingat bahwa salah satu tanda sama dengan merupakan penugasan.

Metode lain dari pemilihan adalah dengan menggunakan fungsi subset.

```
> subset(Familydata, sex=="F")
```

Untuk memilih hanya variabel 'ht' dan 'berat' diantara perempuan adalah dengan cara :

```
> subset(Familydata, sex=="F", select = c(ht,wt))
```

Perlu diketahui bahwa perintah untuk memilih subset tidak memiliki efek permanen pada data frame . Pengguna harus menyimpan ini menjadi sebuah objek baru jika akan digunakan lebih lanjut.

### *Menambahkan variabel ke data frame*

Seringkali kita perlu membuat variabel baru dan menambahkan ke data frame yang ada. Sebagai contoh, kita mungkin ingin membuat variabel baru bernama 'log10money' yang sama dengan log basis 10 dari uang saku.

```
> Familydata$log10money <- log10(Familydata$money)
```

Atau kita dapat menggunakan fungsi transformasi.

```
> Familydata <- transform(Familydata,
  log10money=log10(money))
```

Data frame sekarang berubah dengan tambahan variabel baru 'log10money ".Hal ini dapat diperiksa dengan perintah berikut.

```
> names(Familydata)
> summ(Familydata)

Anthropometric and financial data of a hypothetic family
No. of observations = 11

  Var. name  Obs.  mean    median  s.d.    min.  max.
1 code
2 age        11    45.73   47      24.11   6     80
3 ht         11    157.18  160     14.3    120   172
4 wt         11    54.18   53      12.87   22    71
5 money      11    1023.18 500     1499.55 5     5000
6 sex        11    1.364   1       0.505   1     2
7 log10money 11    2.51    2.7     0.84    0.7   3.7
```

Menghapus variabel dari data frame

Sebaliknya, jika kita ingin mengeluarkan variabel dari data frame, hanya dengan memberikan tanda minus di depan subskrip kolom:

```
> Familydata[,-7]
   code age  ht wt money sex
1     K   6 120 22     5   F
2     J  16 172 52    50   M
3     A  80 163 71   100   M
4     I  18 158 51   200   F
```

```
5      C  69 153 51   300   F
6      B  72 148 60   500   F
7      G  46 160 50   500   F
8      H  42 163 55   600   F
9      D  58 170 67  2000   M
10     F  47 155 53  2000   F
11     E  49 167 64  5000   M
```

Perhatikan lagi bahwa ini hanya menampilkan bagian yang diinginkan dan tidak memiliki efek permanen pada data frame . Perintah berikut akan menghapus secara permanen variabel dan mengembalikan data frame kembali ke keadaan semula.

```
> Familydata$log10money <- NULL
```

Menempatkan nilai NULL ke variabel dalam data frame setara dengan menghapus variabel tersebut.

Pada tahap ini, adalah mungkin bahwa Anda telah membuat beberapa kesalahan pengetikan. Beberapa dari mereka mungkin cukup serius untuk membuat data frame **Familydata** terdistorsi atau bahkan tidak tersedia dari lingkungan. Anda selalu bisa menyegarkan lingkungan R dengan menghapus semua benda, kemudian dibaca lagi dalam dataset.

```
> zap()
> data(Familydata)
```

## Melampirkan data frame ke path (jalur) pencarian

Mengakses variabel dalam frame data dengan awalan variabel dengan nama dari data yang rapi namun sering membingungkan, terutama jika data frame dan nama variabel yang panjang. Menempatkan atau memasang data frame ke dalam path pencarian menghilangkan kebutuhan awalan nama variabel yang membosankan dengan data frame. Untuk memeriksa langkah pencarian, ketik:

```
> search()
 [1] ".GlobalEnv"        "package:epicalc"
 [3] "package:methods"   "package:stats"
 [5] "package:graphics"  "package:grDevices"
```

```
 [7] "package:utils"     "package:datasets"
 [9] "package:foreign"   "Autoloads"
[11] "package:base"
```

Penjelasan umum dari pencarian () diberikan dalam Bab 1. Data frame kita tidak dalam path pencarian. Jika kita mencoba untuk menggunakan variabel dalam data frame yang tidak dalam path pencarian, kesalahan akan terjadi.

```
> summary(age)
Error in summary(age) : Object "age" not found
```

Cobalah perintah berikut:

```
> attach(Familydata)
```

The search path now contains the data frame in the second position.

Path pencarian sekarang berisi data frame di posisi kedua.

```
> search()
 [1] ".GlobalEnv"        "Familydata"        "package:methods"
 [4] "package:datasets" "package:epicalc"
  "package:survival"
 [7] "package:splines"  "package:graphics"
  "package:grDevices"
[10] "package:utils"     "package:foreign"  "package:stats"
[13] "Autoloads"         "package:base"
```

Karena 'usia' ada di dalam Familydata, yang sekarang dalam path pencarian, perhitungan statistik pada 'usia' sekarang menjadi mungkin.

```
> summary(age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   6.00   30.00   47.00   45.73   63.50   80.00
```

Melampirkan data frame ke path pencarian ini mirip dengan memuat paket menggunakan fungsi **library**. Data frame terlampir, serta paket dimuat,sebenarnya dibaca ke dalam memori R dan ditempatkan dalam memori sampai terpisah. Hal ini berlaku bahkan jika data frame asli telah dihapus dari memori.

```
> rm(Familydata)
> search()
```

Data frame **Familydata** yang masih dalam path pencarian memungkinkan setiap variabel dalam data frame akan digunakan.

```
> age
 [1]  6 16 80 18 69 72 46 42 58 47 49
```

Memuat ***library*** yang sama berulang-ulang tidak berpengaruh pada path pencarian tetapi melampirkan kembali data frame yang sama akan membebani sumber daya sistem.

```
> data(Familydata)
> attach(Familydata)

      The following object (s) are masked from Familydata
  ( position 3 ) :
         age code ht money sex wt
```

Variabel-variabel ini sudah di posisi kedua dari path pencarian. Melampirkan lagi hal ini dapat menciptakan konflik dalam nama variabel.

```
> search()
 [1] ".GlobalEnv"        "Familydata"       "Familydata"
 [4] "package:methods"   "package:datasets"
  "package:epicalc"
 [7] "package:survival"  "package:splines"
  "package:graphics"
[10] "package:grDevices" "package:utils"
  "package:foreign"
[13] "package:stats"     "Autoloads"        "package:base"
```

Path pencarian sekarang berisi dua objek bernama Familydata di posisi 2 dan 3. Keduanya memiliki lebih atau kurang set yang sama dari variabel dengan nama yang sama. Ingat bahwa setiap kali suatu perintah yang diketik dan tombol Enter ditekan, pertamanya sistem akan memeriksa apakah suatu objek dalam lingkungan global. Jika tidak, R memeriksa apakah komponen dari path pencarian yang tersisa, yaitu, variabel dalam data frame terlampir atau fungsi dalam salah satu paket dimuat.

Berulang kali pemuatan ***library*** yang sama tidak menambah path pencarian karena R mengetahui bahwa isi di ***library*** tidak berubah selama sesi yang sama. Namun, data frame dapat berubah setiap saat selama sesi tunggal, seperti yang terlihat pada bagian sebelumnya dimana variabel 'log10money' ditambahkan dan kemudian dihapus. Data frame yang melekat pada posisi 2 mungkin akan berbeda dengan objek dengan nama yang sama di posisi pencarian lain. Kebingungan muncul jika sebuah objek independen (misalnya vektor) yang dibuat di luar data frame (dalam lingkungan global) dengan nama

yang sama dengan data frame atau jika dua data frame yang berbeda dalam path pencarian masing-masing berisi sebuah variabel dengan nama yang sama. Konsekuensinya dapat menjadi masalah.

Selain itu, semua elemen dalam path pencarian menempati memori sistem. Data frame **Familydata** dalam path pencarian menempati jumlah memori yang sama dengan yang di ruang kerja saat ini. Menggandakan memori tidak menjadi masalah serius jika data frame kecil. Namun, berulang kali melampirkan pada data frame yang besar dapat menyebabkan ***R*** tidak dapat mengeksekusi karena memori tidak cukup.

Dengan alasan ini, ini merupakan praktik pertama yang baik, untuk menghapus data frame dari path pencarian setelah tidak diperlukan lagi. Kedua, menghapus objek dari lingkungan menggunakan rm (list = ls ()) ketika mereka tidak diperlukan lagi. Ketiga, tidak mendefinisikan objek baru (misalnya vektor atau matriks) yang mungkin memiliki nama yang sama dengan data frame dalam path pencarian. Sebagai contoh, kita tidak harus menciptakan vektor baru yang disebut **Familydata** seperti yang kita sudah memiliki data frame Familydata dalam path pencarian.

Mengeluarkan kedua versi **Familydata** dari path pencarian.

```
> detach(Familydata)
> detach(Familydata)
```

Perhatikan bahwa perintah ***detachAllData*** () dalam Epicalc menghapus semua lampiran ke data frame. Perintah zap () tidak sama, melainkan menghapus semua objek yang bukan fungsi (non-function objects). Dengan kata lain, perintah zap () adalah setara dengan rm (list = lsNoFunction ()) dan diikuti oleh detachAllData ().

## Perintah 'use' di Epicalc

Melampirkan ke dan memisahkan dari data frame sering membosankan dan rumit dan jika ada lebih dari satu data frame di ruang kerja maka pengguna harus berhati-hati bahwa mereka melekat ke data frame yang benar saat

bekerja dengan data mereka. Kebanyakan analisis data hanya berurusan dengan data frame tunggal. Untuk mengurangi langkah-langkah melampirkan dan memisahkan, Epicalc berisi perintah yang disebut **use** yang memudahkan proses. Pada konsol R ketik:

```
> zap()
> data(Familydata)
> use(Familydata)
```

Perintah **use()** membaca dalam file data dari Dbase (.dbf), Stata (.dta), SPSS (.sav), EpiInfo (.rec) dan nilai dipisahkan koma dengan format (.csv) , serta mereka yang berasal dari pra-paket yang disertakan dengan R. Data frame **Familydata** dilengkapi dengan Epicalc. Jika Anda ingin membaca dataset dari format file Stata, seperti **"family.dta"**, cukup ketik **use ("family.dta")** tanpa mengetikkan perintah data di atas. Dataset akan disalin ke memori dalam data frame standar yang disebut **.data.** Jika **.data** sudah ada, maka akan ditimpa oleh data frame baru. Para **Familydata** asli, bagaimanapun akan tetap.

Bahkan, semua dataset di Epicalc awalnya adalah salah satu format file dari **.dta, .rec, .csv** atau **.txt.** Dataset ini dalam format aslinya dapat didownload dari http://medipe.psu.ac.th/Epicalc/. Jika Anda men-download file dan mengatur direktori kerja untuk R ke folder default **"C: \ RWorkplace"**, Anda tidak perlu mengetik **data (Familydata)** dan **use (Familydata)**, melainkan cukup mengetik:

```
> use("family.dta")
```

File Stata asli akan dibaca ke R dan disimpan sebagai .data. Jika berhasil, ia tidak akan membuat perbedaan apakah Anda mengetik **data (Familydata)** diikuti oleh **use (Familydata)** atau hanya menggunakan **("family.dta").**

Di sebagian besar buku ini, kami memilih untuk memberitahu Anda untuk mengetik **data (Familydata)** dan **use (Familydata)** bukan penggunaan **("family.dta")** karena dataset sudah dalam paket Epicalc, yang sudah tersedia ketika Anda menggunakan Epicalc ke titik ini. Namun, menempatkan**"filename.extension"** sebagai argumen seperti **use ("family.dta")** dalam bab ini atau **use ("timing.dta")** dalam bab berikutnya, dan sebagainya, dapat memberikan pengertian yang sebenarnya dari membaca file aktual bahkan dari pendekatan yang digunakan dalam buku ini.

Perintah **use** juga secara otomatis menempatkan data frame, .data, ke dalam

path pencarian. Dengan mengetikkan :

```
> search()
```

Anda akan melihat bahwa .data di posisi kedua dari path pencarian. ketik:

```
> ls()
```

Anda akan melihat hanya objek **Familydata**, dan bukan **.data** karena nama objek ini dimulai dengan sebuah titik dan diklasifikasikan sebagai objek tersembunyi..Dalam rangka untuk menunjukkan bahwa **.data** benar-benar dalam memori, ketik :

```
> ls(all=TRUE)
```

Anda akan melihat .data dalam posisi pertama dari daftar.

**. data tahan untukzap ()**

Ketik argument berikut di konsol R:

```
> zap()
> ls(all=TRUE)
```

Objek **Familydata** hilang tapi **.data** masih ada. Namun, keterikatan pada path pencarian sekarang hilang

```
> search()
```

Untuk meletakkannya kembali ke path pencarian, kita harus melampirkan secara manual.

```
> attach(.data)
```

Keuntungan dari **use ()** tidak hanya menghemat waktu dengan membuat lampiran dan melepaskan yang tidak perlu, tapi .data ditempatkan dalam path pencarian serta dibuat data frame standar. Jadi **des ()** adalah sama dengan **des (.data)**, **summ ()** setara dengan **summ (.data).**

```
> des()
> summ()
```

Urutan perintah **zap, data (datafile), use (datafile), des ()** dan **summ ()** direkomendasikan untuk memulai analisis dihampir semua dataset dalam buku ini. Sejumlah perintah lain dari paket Epicalc berdasarkan strategi ini membuat

**.data** data frame default dan eksklusif melekat pada path pencarian (semua data frame lainnya akan dikeluarkan, kecuali argumen 'clear =FALSE' ditentukan dalam **fungsi use**). Untuk analisis data sederhana, perintah **use ()** sudah cukup untuk membuat pengaturan ini. Dalam banyak kasus dimana data yang dibaca butuh untuk dimodifikasi, disarankan untuk mengubah nama atau menyalin data frame final ke **.data**. Kemudian melepaskan dari data lama .data dan lampirkan kembali ke dalam path pencarian yang paling update.

Strategi ini tidak memiliki efek pada fungsi standar **R**. Pengguna Epicalc masih dapat menggunakan perintah lain dari **R** sementara masih menikmati manfaat dari Epicalc.

---

## Latihan

Dengan beberapa dataset yang disediakan Epicalc, gunakan perintah terakhir *(zap, data, use, des, summ)* untuk melihat/mengakses data tersebut dengan cepat.

BAB 5

# Eksplorasi Data Sederhana

## Eksplorasi Data Menggunakan Epicalc

Di bab sebelumnya, kita telah mempelajari "commands" *zap* untuk membersihkan "workspace" dan memori, *use* untuk membaca file data dan *codebook*, *des* dan *summ* untuk menginisialkan eksplorasi kerangka data (data frame), ingat bahwa semua ini adalah Epicalc commands. Fungsi *use* menempatkan kerangka data kedalam sebuah objek tersembunyi yang bisa dipanggil **.data**, secara otomatis terlampir pada search path. Di bab ini, kita akan bekerja dengan lebih banyak contoh kerangka data sebaik-baiknya cara untuk mengeksplor variabel individu.

```
> zap()
> data(Familydata)
> use(Familydata)
> des()

Anthropometric and financial data of a hypothetical family

No. of observations = 11
  Variable      Class           Description
1 code          character
2 age           integer         Age(yr)
3 ht            integer         Ht(cm.)
```

```
4 wt            integer         Wt(kg.)
5 money         integer         Pocket money(B.)
6 sex           factor
```

Garis pertama setelah command *des()* menunujukkan *label* data, yang mendeskripsikan teks untuk dataframe. Biasanya dihasilkan oleh software yang digunakan untuk memasukkan data, seperti Epidata atau Stata. Baris berikutnya menunjukkan nama variabel dan deskripsi dari masing-masing variabel. Untuk variabel 'code' berjenis character sedangkan 'sex' adalah sebuah faktor. Sedangkan variabel yang lain berjenis integer. Suatu variable character tidak digunakan untuk perhitungan statistik tetapi hanya bertujuan untuk memberikan label secara sederhana atau untuk merekam hasil identifikasinya. Pemanggilan kembali sebuah faktor yakni yang disebut **R** merupakan suatu pengelompokkan atau grup variabel. Variabel integer yang tersisa ('age', 'ht', 'wt' and 'money') merupakan variabel kontinu secara intuisi. Variabel 'code' dan 'sex' tidak mempunyai deskripsi variabel karena tidak dicantumkan selama persiapan dari data sebelumnya untuk entry data.

```
> summ()

Anthropometric and financial data of a hypothetical family

No. of observations  = 11

  Var. name Obs.  mean    median s.d.    min.   max.
1 code
2 age       11    45.73   47     24.11   6      80
3 ht        11    157.18  160    14.3    120    172
4 wt        11    54.18   53     12.87   22     71
5 money     11    1023.18 500    1499.55 5      5000
6 sex       11    1.364   1      0.505   1      2
```

Sebagaimana disebutkan didalam bab sebelumnya, command *summ* menghasilkan ikhtisar statistik dari semua variabel dalam default kerangka data, dalam kasus ini **.data**. masing-masing dari enam variabel mempunyai 11 observasi, yang berarti bahwa tidak ada nya nilai yang hilang di dalam dataset tersebut. Selama variabel 'code' merupakan kelas 'character' (seperti ditunjukkan dari command '*des()*' diatas, informasi mengenai variabel ini tidak ditunjukkan. Umur merupakan subyek dalam dataset dengan rentang dari 6-80 (tahun). Tinggi badan mereka berkisar antara 120-172 (cm), dan berat badan mereka berkisar antara 22-71 (kg). Untuk variabel 'uang' berkisar dari 5-

5,000 (baht). Nilai mean dan median umur, tinggi badan dan berat badan saling mendekati sehingga menunjukkan adanya hubungan dengan distribusi-ketakmencengengan. Variabel 'uang' memiliki nilai mean lebih besar daripada nilai median signifikan bahwa distribusinya condong ke kanan. Variable terakhir, 'sex', adalah sebuah faktor. Bagaimanapun, statistiknya berdasarkan nilai dari variabel yang tidak dikelompokkan. Kita bisa lihat bahwa ada dua tingkatan, jika nilai minimum adalah 1 dan nilai maksimum adalah 2. Untuk faktornya, semua nilai disimpan sebagai integer didalamnya misalnya hanya 1 atau 2 dalam kasus ini. Nilai mean dari 'sex' adalah 1.364 mengindikasikan bahwa 36.4% dari subyek mempunyai level kedua dari faktor tersebut (dalam kasus ini adalah pria). Jika sebuah faktor mempunyai lebih dari dua tingkatan, maka nilai mean tidak memerlukan interpretasi.

## *Codebook*

Fungsi dari *summ* memberikan ringkasan statistik dari masing-masing variabel, baris demi baris. Ini sangat berguna untuk variabel numerik tetapi kurang berguna untuk faktor, khususnya dengan lebih dari dua level. Epicalc mempunyai fungsi lain yang bisa memberikan ringkasan statistik untuk variabel numerik dan tabel frekuensi dengan tingkatan label dan kode untuk faktor.

```
> codebook()

 Anthropometric and financial data of a hypothetical family

code     :
A character vector
 ==================
age      :       Age(yr)
  obs. mean   median  s.d.   min.   max.
  11   45.727 47      24.11  6      80
 ==================
ht       :       Ht(cm.)
  obs. mean    median  s.d.   min.   max.
  11   157.182 160     14.3   120    172
 ==================
wt       :       Wt(kg.)
  obs. mean   median  s.d.   min.   max.
  11   54.182 53      12.87  22     71
 ==================
money    :       Pocket money(B.)
```

```
  obs. mean     median  s.d.    min.   max.
  11   1023.182 500     1499.55 5      5000
 ==================
sex      :
Label table: sex1
  code Frequency Percent
F    1         7    63.6
M    2         4    36.4
 ==================
```

Tidak seperti hasil dari fungsi *summ*, *codebook* berhubungan dengan masing-masing variabel dalam dataframe secara lebih mendetail. Jika ada sebuah label variabel, maka ditampilkan outputnya. Untuk faktor, nama dari tabel untuk label tingkatan ditunjukkan dan kode untuk tingkatan ditampilkan dalam kolom, diikuti oleh frekuensi dan persentase dari distribusi. Oleh karena itu, fungsi tersebut sangat berguna. Outputnya bisa digunakan untuk menulis tabel dari data awal dari naskah yang berasal dari dataframe.

Output tersebut mengkombinasikan deskripsi variabel dengan ringkasan statistik untuk semua variabel numerik. Untuk 'sex', merupakan sebuah faktor, label asli dinamakan 'sex1' yakni 1 = W dan 2 = P. Ada 7 wanita dan 4 pria didalam keluarga tersebut.

Catatan pada tabel untuk label code dari sebuah faktor dengan mudah bisa dikerjakan didalam fase persiapan data entry menggunakan Epidata dengan pengaturan dari file ".chk". Jika suatu data dieksport kedalam format Stata, kemudian tabel dari masing-masing label variabel akan dieksport ke seluruh dataset. Tabel label yang dilewati atribut dalam koresponding data kerangka. Epicalc command *codebook* seluruhnya berguna dalam atribut ini yang mengizinkan pengguna untuk melihat dan mendokumenkan skema code sebagai referensi yang akan datang.

Kita juga bisa mengeksplor variabel individu secara lebih detil dengan beberapa commands yang sama yaitu *des* dan *summ* secara menggantikan nama variabel didalam tanda kurung.

```
> des(code)

'code' is a variable found in the following source(s):

 Var. source  Var. order Class     # records Description
 .data        1          character 11
```

Outputnya mengatakan bahwa 'code' ada didalam **.data**. Andaikan kita bisa membuat sebuah objek yang juga dipanggil 'code', tetapi posisinya secara bebas ditempatkan diluar data kerangka yang tersembunyi.

```
> code <- 1
> des(code)

'code' is a variable found in the following source(s):

 Var. source  Var. order Class      # records Description
 .GlobalEnv               numeric    1
 .data        1           character  11
```

Output tersebut mengatakan bahwasanya ada dua 'codes'. Yang pertama akhir-akhir ini digunakan untuk menyatakan lingkungan global. Sedangkan yang kedua adalah variabel didalam dataframe, **.data**. Untuk mencegah adanya kekacauan, kita akan menghapus objek yang baru dibuat yaitu 'code'.

```
> rm(code)
```

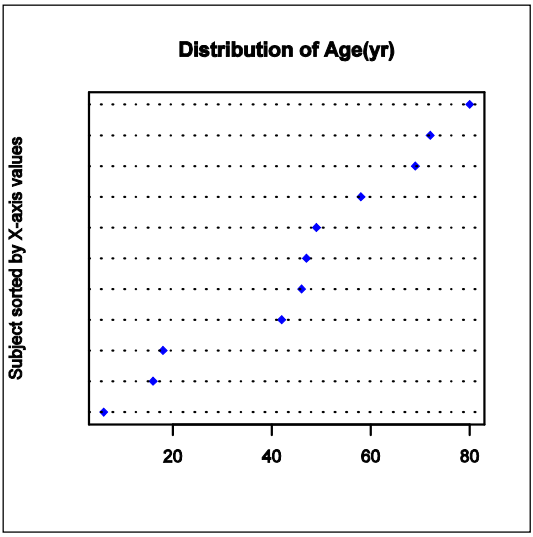
Setelah memindahkan 'code' dari lingkungan global, command *des*() terakhir akan mendeskripsikan variabel 'code', yang merupakan bagian dari **.data**, dan bisa digunakan kembali. Menggunakan *des*() dengan variabel lain menunjukkan hasil yang serupa.

Sekarang coba ikuti command berikut ini:

```
> summ(code)
```

Hal ini menyebabkan terjadinya error karena 'code' merupakan objek berkarakter. Selanjutnya ketikkan:

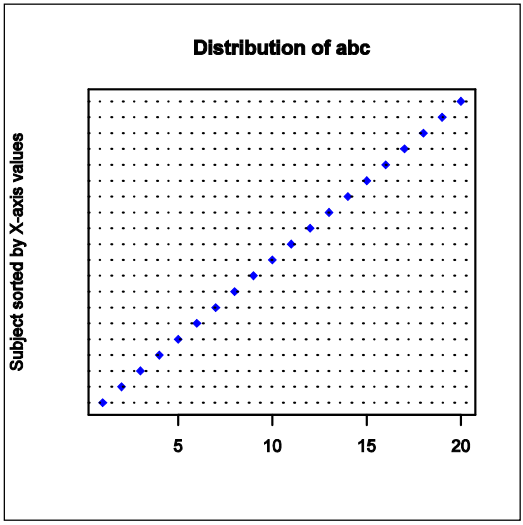
```
> summ(age)
  Obs.  mean    median  s.d.   min.   max.
  11    45.727  47      24.11  6      80
```




Hasil yang didapat serupa dengan apa yang kita lihat dari *summ*. Oleh karena itu, selama argumen command *summ* merupakan variabel tunggal, grafiknya juga bisa ditunjukkan dari distribusi umur.

Judul dari grafik tersebut mendeskripsikan variabel setelah kata "distribusi dari". Jika variabel yang tidak mempunyai deskripsi, nama variabel akan dijelaskan didalamnya. Sekarang kita ikuti command dibawah ini:

```
> abc <- 1:20
> summ(abc)
  Obs.   mean   median  s.d.   min.   max.
    20   10.5    10.5   5.916  1      20
```




Objek '`abc`' mempunyai distribusi yang seragam dan sempurna dimana titiknya mendekati garis lurus.

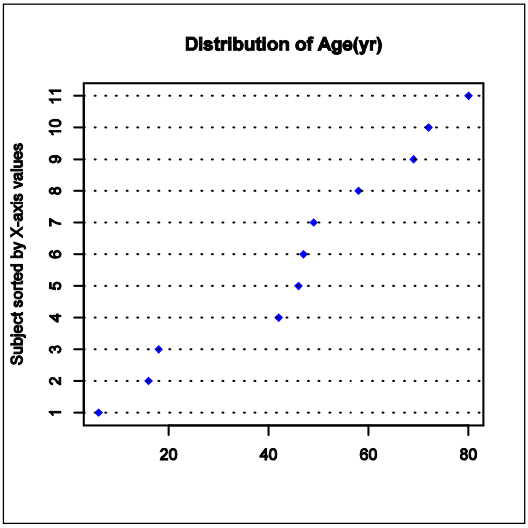
Grafik yang dihasilkan dari command *`summ`* merupakan diagram yang sudah diurutkan. Sebuah dot chart mempunyai satu sumbu axis (dalam kasus ini X-axis) mewakili rentang variabel. Sumbu axis yang lain yaitu the Y-axis, dilabeli 'Subyek yang diurutkan oleh nilai dari sumbu X-axis', mewakili masing-masing subyek atau pengamatan yang diurutkan oleh nilai dari variabel. Untuk objek '`abc`', nilai terkecil adalah 1, yang diplotkan disebelah kiri bawah, lalu 2, 3, 4 dst. Observasi terakhir adalah 20, yang diplotkan disebelah kanan atas. Pertambahan nilai bertambah semakin tinggi dari satu pengamatan ke pengamatan selanjutnya. Kenaikan yang terjadi terus menerus, sehingga menunjukkan garis lurus yang sempurna.

Untuk melihat grafik umur maka ketikkan:

```
> summ(age)
> axis(side=2, 1:length(age))
```

Command pada sumbu '`axis`' menambahkan tanda petik dan label nilai pada sumbu axis yang telah ditentukan (dalam kasus ini, 'side=2' menunjukkan sumbu Y-axis). Tanda petik (tick) menempatkan nilai 1, 2, 3, sampai 11 (yang

merupakan panjang dari vektor umur). Tanda petik diabaikan secara default dimana jika vektornya terlalu panjang, sehingga akan terlalu padat/banyak. Dalam sesi ini, tanda petik akan memfasilitasi diskusi.




Untuk memfasilitasi pertimbangan yang lebih detil, vektor umur yang telah diurutkan ditunjukkan oleh grafik berikut.

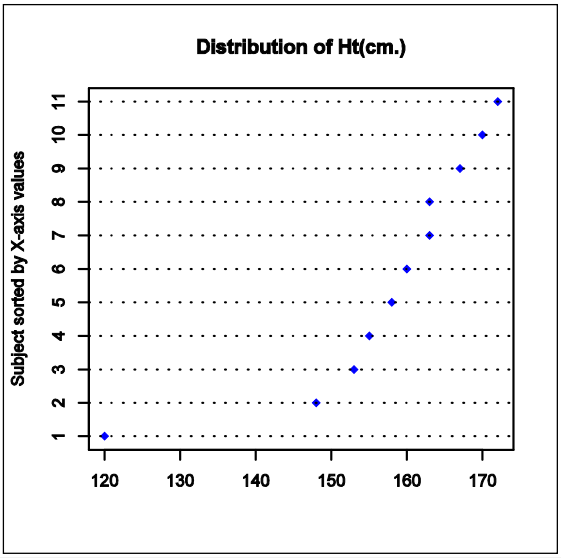
```
> sort(age)
 [1]  6 16 18 42 46 47 49 58 69 72 80
```

Kenaikan hubungan pada sumbu X-axis dari pengamatan pertama (6 tahun) ke pengamatan kedua (16 tahun) lebih besar dari kedua ke pengamatan ketiga (18 tahun). Dengan demikian kita mengamati kenaikan yang curam dalam sumbu Y-axis untuk pasangan kedua. Dari pengamatan ketiga hingga pengamatan keempat (42 tahun), kenaikannya lebih besar dari tahap pertama; kemiringannya relatif datar. Dengan kata lain, tidak adanya titik antara 20 dan 40 tahun. Nilai keempat, kelima, keenam dan ketujuh relatif saling mendekati, dengan demikian adanya kenaikan yang curam pada sumbuY-axis.

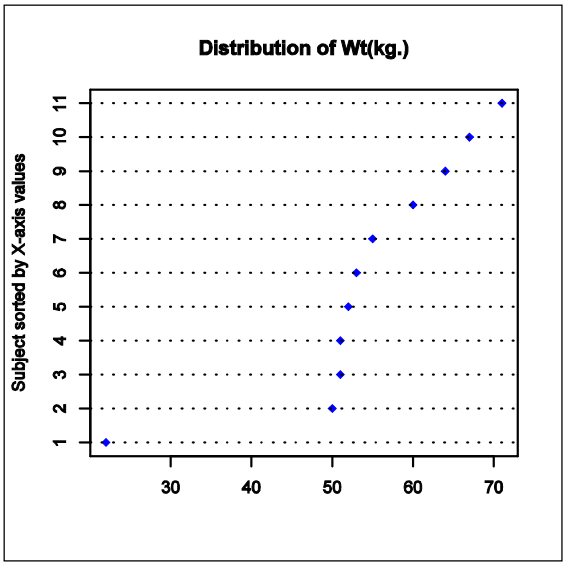
```
> summ(ht)
  Obs.  mean    median  s.d.   min.   max.
  11    157.182 160     14.303 120    172
> axis(side=2, 1:length(ht))
```

```
> sort(ht)
 [1] 120 148 153 155 158 160 163 163 167 170 172
```

Distribusi tinggi badan ditampilkan dengan grafik yang menarik. Subyek terpendek (120cm) lebih pendek dari subyek sebelumnya. Faktanya, seorang anak perempuan diantara orang dewasa. Ada dua orang (ketujuh dan kedelapan) dengan tinggi badan yang sama (163cm). Kenaikan pada sumbu Y-axis adalah vertikal.




```
> summ(wt)
> axis(side=2, 1:length(wt))
```

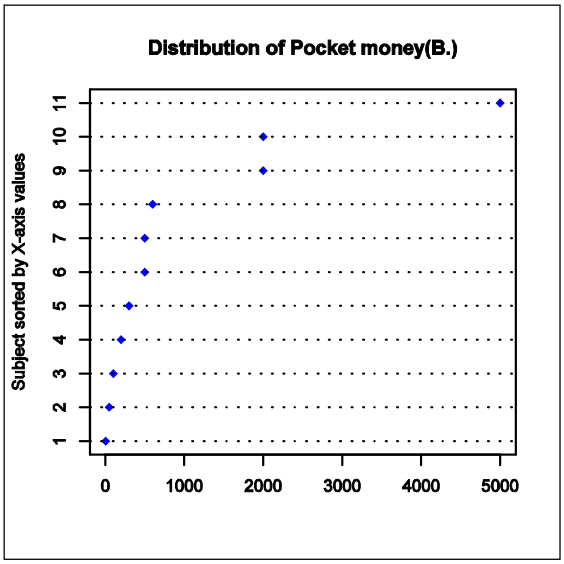



Level yang lebih tinggi dari pengelompokkan berat badan daripada tinggi badan dari pengematan kedua hingga pengamatan ketujuh; ada enam orang yang mempunyai berat badan yang serupa. Dari pengamatan kedelapan sampai pengamatan kesebelas, distribusinya cukup seragam.

Untuk distribusi dari variabel uang, ketikkan:

```
> summ(money)
```

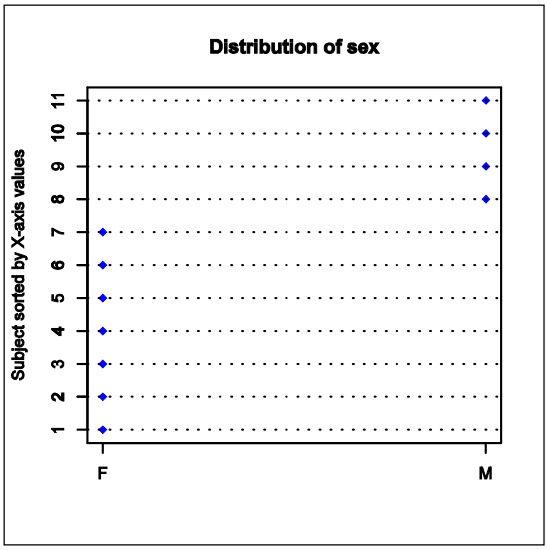
Uang mempunyai distribusi kemencengan. Tujuh orang pertama mengantongi uang kurang dari 1,000 baht. Dua orang selanjutnya mengenatongi uang sekitar 2,000 baht sebaliknya yang terakhir mengantongi uang 5,000 baht, semakin jauh ( sumbu X-axis) dari yang lain. Ini merupakan apa yang disebut dengan teori distribusi eksponensial.




Selanjutnya amati distribusi dari variabel jenis kelamin berikut.

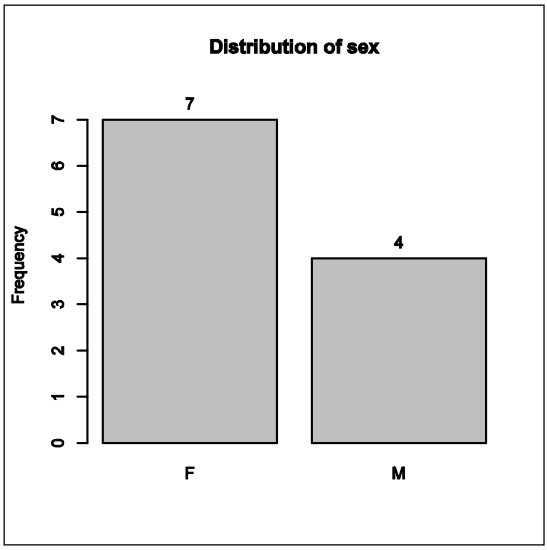
```
> summ(sex)
  Obs.   mean    median   s.d.    min.    max.
  11     1.364   1        0.5     1       2
```

Grafik menunjukkan ada empat dari sebelas (36.4%, ditunjukkan secara statistik) merupakan pria. Ketika variabel faktornya telah diberikan label, nilai tersebut akan menunjukka nama dari kelompoknya.



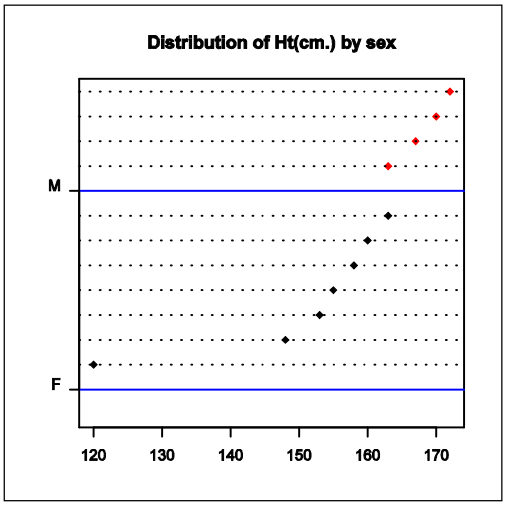

Faktanya, hasil yang lebih baik bisa dihasilkan dengan mengetikkan

```
> tab1(sex)
sex :

        Frequency Percent Cum. percent
F               7    63.6         63.6
M               4    36.4        100.0
  Total        11   100.0        100.0
```




Dimana dua jenis kelamin, kita bisa membandingkan distribusi tinggi badan oleh jenis kelamin.

```
> summ(ht, by=sex)
For sex = F
  Obs.  mean   median  s.d.   min.   max.
  7     151    155     14.514 120    163

For sex = M
  Obs.  mean   median  s.d.   min.   max.
  4     168    168.5   3.916  163    172
```



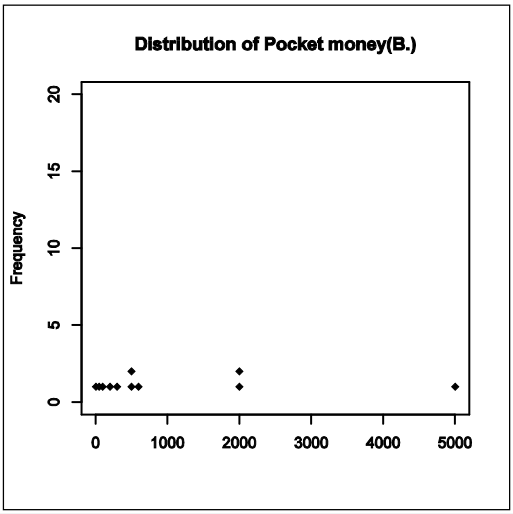

Jelas terlihat bahwasanya, pria lebih tinggi dari wanita.

## *Dotplot*

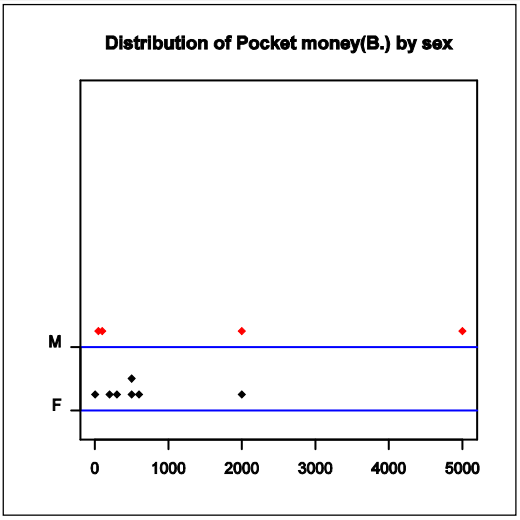
Dalam penambahan `summ` dan `tabl`, *Epicalc* mempunyai alat eksplorasi yang lain disebut `dotplot`.

```
> dotplot(money)
```

Grafik dihasilkan dari command `summ` memplotkan nilai peringkat individual, `dotplot` membagi skala kedalam beberapa binary yang sama dan berukuran kecil (default = 40) dan meletakkan hasilnya masing-masing kedalam "corresponding bin". Dari gambar diatas, ada tiga pengamatan yang lebih banyak terletak disebelah kiri dan satu disebelah kanan. Plotnya sangat mirip dengan histogram kecuali nilai aslinya muncul pada sumbu X-axis. Banyak orang lebih mengenal dot plot daripada dot chart yang dihasilkan oleh `summ`. Bagaimanapun, plot yang terakhir memberikan informasi yang lebih detil dengan keakuratan yang lebih baik. Jika ukuran sampelnya kecil, plot yang dihasilkan oleh `summ` lebih informatif. Sedangkan ukuran sampelnya besar (diatas 200), `dotplot` lebih mudah dipahami oleh orang banyak.




```
> dotplot(money, by=sex)
```




Command *summ* dengan mudah menghasilkan grafik yang sangat mendukung. Salah satunya bisa menunjukkan informasi yang lebih. **R** bisa menyajikan untuk banyak tujuan, tetapi seorang user harus menyediakan waktu yang banyak untuk mempelajarinya.

Andaikan digambar sebuah dot chart yang telah diurutkan untuk tinggi badan. Command dibawah seharusnya diikuti pertahap untuk melihat perubahan dari grafik yang dihasilkan dari setiap baris. Jika anda membuat satu kesalahan yang serius maka dengan mudah bisa dimulai lagi dari baris pertama. Menggunakan tombol panah "up", command sebelumnya bisa diedit sebelum dieksekusi lagi.

```
> zap()
> data(Familydata)
> use(Familydata)
> sortBy(ht)
> .data
```

Command *sortBy*, tidak seperti equivalent `sort` dari library **base**, mempunyai efek yang permanen pada **`.data`**. Keseluruhan dataframe telah diurutkan secara ascending berdasarkan nilai tinggi badan.

```
> dotchart(ht)
```

Setelah data diurutkan, maka kenaikan polanya tidak akan terlihat.

```
> dotchart(ht, col=unclass(sex), pch=18)
```

Penunjukkan warna-warna yang terpisah untuk setiap jenis kelamin dapat dilakukan dengan fungsi '`unclass`'. Dimana 'jenis kelamin' adalah sebuah faktor, fungsi dari unclassing memberikan vektor numerik dengan 1 untuk level pertama (wanita) dan 2 level kedua (pria). Warna bisa dispesifikasikan dalam beberapa cara dalam **R**. Satu cara yang sederhana untuk menggunakan tabel warna sederhana dikenal dengan *palette*. Default palette mempunyai 9 warna, dimana nomor 1 menunjukkan warna hitam, nomor 2 menunjukkan warna merah, sampai nomor 9 menunjukkan warna abu-abu. Kemudian titik hitam menunjukkan "wanita" dan titik merah menunjukkan "pria". Lebih detil bagaimana melihat atau memanipulasi palette bisa ditemukan dihalaman bantuan.

Untuk menambah sumbu y-axis, ketikkan command berikut ini:

```
> axis(side=2,at=1:length(ht), labels=code, las=1)
```

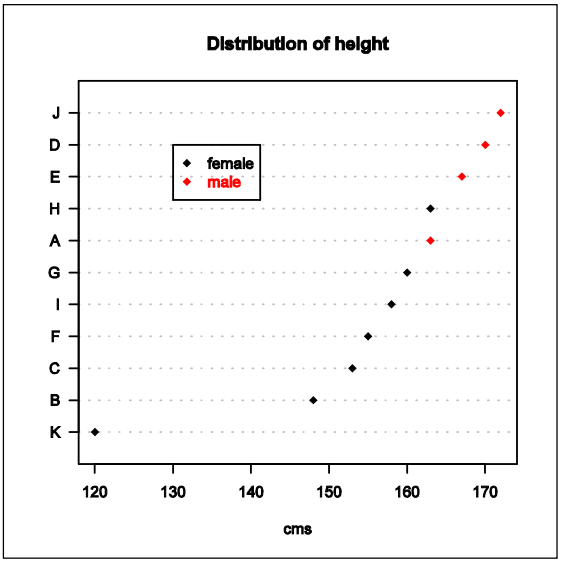
Argumen 'las' merupakan parameter grafis, yang menentukan orientasi dengan memberikan tanda label pada sumbu. Jika '`las=1`', semua label akan horisontal dengan sumbu. Untuk menambah legenda maka bisa menggunakan command '`legend`':

```
> legend(x=130, y=10, legend=c("female","male"), pch=18,
  col=1:2, text.col=1:2)
```

Argumen 'pch' untuk memplotkan karakterter. Kode 18 bermakna simbol yang ditandai dengan bentuk diamond yang udah ditebalkan dan lebih jelas dari `pch=1` (sebuah titik yang berlubang). Catatan bahwa 'col' untuk simbol plot yang berwarna dan 'text.col' untuk mewarnakan teks dalam legenda.

Untuk menambahkan judul ketikkan:

```
> title(main="Distribution of height")
> title(xlab="cms")
```




Untuk meringkaskan, setelah menggunakan `(datafile)`, *`des`* dan *`summ`*, variable individu bisa dieksplor secara sederhana oleh *`summ`*`(var.name)` dan *`summ`*`(var.name, by=group.var)`. Dalam penambahan ringkasan statistics, dot chart yang diurutkan bisa lebih informatif. Command *`dotplot`* dalam keakurasian nilai individu dengan frekuensi dot plots, serupa dengan histogram. Lebih jauh menggunakan command ini akan didemonstrasikan jika jumlah pengamatan lebih besar.

## Latihan

Cobalah simulasi dibawah ini dengan berbagai variasi ukuran sampel dan jumlah grup. Bandingkan grafik dengan tipe berbeda menggunakan tiga commands, summ, dotplot dan boxplot. Untuk masing-masing kondisi, tipe grafik mana yang terbaik?

## sampel ukuran kecil, dua grup.

```
> grouping1 <- rep(1:2, times=5)
> random1   <- rnorm(10, mean=grouping1, sd=1)
> summ(random1, by=grouping1)
> dotplot(random1, by=grouping1)
> boxplot(random1 ~ grouping1)
```

## sampel ukuran sedang, tiga grup.

```
> grouping2 <- c(rep(1, 10),rep(2, 20), rep(3, 45))
> random2   <- rnorm(75, mean=grouping2, sd=1)
> summ(random2, by=grouping2)
> dotplot(random2, by=grouping2)
> boxplot(random2 ~ grouping2, varwidth=TRUE, col=1:3,
  horizontal=TRUE, las=1)
```

## sampel ukuran besar, empat grup.

```
> grouping3 <- c(rep(1, 100), rep(2, 200), rep(3,450),
  rep(4, 1000))
> random3   <- rnorm(1750, mean=grouping3, sd=1)
> summ(random3, by=grouping3)
> dotplot(random3, by=grouping3)
> boxplot(random3 ~ grouping3, varwidth=TRUE, col=1:4,
  horizontal=TRUE, las=1)
```

Grafik mana yang terbaik dari perbedaan kondisi diatas?

BAB 6

# Tanggal dan Waktu

Salah satu tujuan dari studi epidemiologi adalah untuk menggambarkan distribusi status kesehatan penduduk dalam hal waktu, tempat dan orang. Sebagian besar data analisis, lebih berurusan dengan seseorang dari waktu dan tempat. Dalam bab ini, penjelasan akan difokuskan pada perihal waktu.

Satuan waktu mencakup abad, tahun, bulan, hari, jam, menit dan detik. Unit yang paling umum yang terlibat langsung dalam penelitian epidemiologi adalah hari. Lokasi kronologis hari adalah tanggal, yang merupakan fungsi serial tahun, bulan dan hari.

Ada beberapa contoh umum penggunaan tanggal dalam studi epidemiologi. Tanggal lahir diperlukan untuk perhitungan usia yang akurat. Dalam sebuah investigasi wabah, deskripsi tanggal eksposur dan onset adalah penting untuk perhitungan masa inkubasi. Dalam tindak lanjut penelitian, waktu tindak lanjut biasanya ditandai dengan tanggal kunjungan. Dalam analisis survival, tanggal mulai pengobatan dan menilai hasil adalah unsur yang dibutuhkan untuk menghitung waktu kelangsungan hidup.

## Perhitungan fungsi yang terkait dengan tanggal.

Bekerja dengan tanggal dapat menyebabkan perhitungan menjadi rumit. Ada tahun kabisat, bulan dengan jumlah hari yang berbeda, hari dalam seminggu dan bahkan lompatan detik. Tanggal bahkan dapat disimpan dalam era yang berbeda tergantung pada kalender. Tugas dasar dalam bekerja dengan tanggal adalah untuk menghubungkan waktu dari tanggal tetap untuk tampilan berbagai format tanggal yang biasa digunakan oleh orang.

Perangkat lunak yang berbeda menggunakan tanggal awal yang berbeda untuk menghitung tanggal. Ini disebut epoch. R menggunakan hari pertama tahun 1970 sebagai epoch (hari 0). Dengan kata lain, tanggal yang disimpan sebagai jumlah hari dimulai sejak 1 Januari 1970, dengan nilai negatif untuk tanggal yang lebih awal. Cobalah berikut ini di konsol R:

```
> a <- as.Date("1970-01-01")
> a
[1] "1970-01-01"
> class(a)
[1] "Date"
> as.numeric(a)
[1] 0
```

Perintah pertama di atas menciptakan 'sebuah' objek dengan Tanggal kelas. Ketika dikonversi ke numerik, nilai adalah 0. Hari ke 100 adalah

```
> a + 100
[1] "1970-04-11"
```

Tampilan default format R untuk sebuah objek Tanggal adalah format ISO. Format Amerika ', hari bulan tahun,' dapat diperoleh dengan

```
> format(a, "%b %d, %Y")
[1] "Jan 01, 1970"
```

'Format' Fungsi menampilkan 'a' objek dalam mode yang dipilih oleh pengguna. '% b' menunjukkan bulan dalam bentuk tiga-karakter disingkat. '% d' menunjukkan nilai hari dan '% Y' menunjukkan nilai tahun, termasuk abad.

Dalam beberapa kondisi sistem operasi, seperti sistem operasi Windows Thailand, '% b' dan '% a' tidak dapat bekerja atau mungkin ada beberapa masalah dengan font. Cobalah perintah berikut:

```
> Sys.setlocale("LC_ALL", "C")
```

Sekarang coba perintah format di atas lagi. Kali ini, sudah dapat bekerja. R memiliki 'locale' atau lokasi kerja yang ditetapkan oleh sistem operasi, yang bervariasi dari negara ke negara. "C" adalah ibu pertiwi R dan bahasa "C" adalah bahasa Inggris Amerika. '% A' dan '% a' adalah format mewakili hari kerja penuh dan disingkat, sedangkan '% B' dan '% b' masing-masing mewakili bulan. Hal ini bergantung pada bahasa dan sistem operasi.

Cobalah berikut ini

```
> b <- a + (0:3)
> b
```

Kemudian ubahlah bahasa dan lihatlah efek pada konsol R dan perangkat grafis.

```
> setTitle("German"); summ(b)
> setTitle("French"); summ(b)
> setTitle("Italian"); summ(b)
```

Perintah setTitle merubah lokal serta kata-kata tetap dari lokal untuk mencocokkannya. Untuk melihat apa bahasa yang saat ini tersedia dalam Epicalc coba:

```
> titleString()
> titleString(return.look.up.table=TRUE)
```

Perhatikan bahwa semua bahasa-bahasa ini menggunakan karakter teks ASCII standar. Hasil ditampilkan dari perintah ini akan tergantung pada sistem operasi. Thailand dan Cina versi Windows dapat memberikan hasil yang berbeda. Anda dapat mencoba setTitle dengan Lokal yang berbeda. Untuk mengatur ulang sistem untuk nilai-nilai asli default Anda, ketik

```
> setTitle("")
```

Untuk bahasa dengan non-standar karakter ASCII, tiga frase sering digunakan dalam Epicalc ("Distribution of", "by", dan "Frequency") dapat diubah ke bahasa Anda sendiri. Untuk lebih jelasnya lihat bantuan untuk fungsi titleString.

Manipulasi string judul, label variabel dan tingkat faktor menggunakan bahasa Anda sendiri memungkinkan Anda dapat memiliki grafik otomatis disesuaikan dengan kebutuhan Anda sendiri. Namun ini agak terlalu rumit untuk ditunjukkan dalam buku ini. Pembaca yang tertarik dapat menghubungi penulis untuk informasi lebih lanjut.

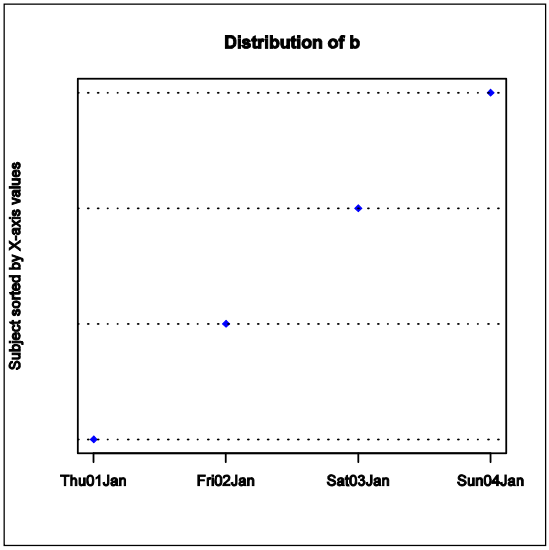
Epicalc menampilkan hasil dari fungsi summ dalam format ISO untuk menghindari bias negara. Hasil grafis hanya dalam kisaran beberapa hari, seperti vektor 'b', memiliki sumbu X label tanda centang dalam format '% a% d% b'. Perhatikan bahwa '% a' menunjukkan hari kerja dalam bentuk tiga-karakter disingkat.

Dalam hal ini tanggal tidak ditampilkan, hanya pemecahan masalah dengan mengetik:

```
> Sys.setlocale("LC_ALL", "C")
```

Kemudian, periksa apakah format tanggal yang berisi '% a' dan '% b' bekerja.

```
> format(b, "%a %d%b%y")
[1] "Thu 01Jan70" "Fri 02Jan70" "Sat 03Jan70" "Sun 04Jan70"

> summ(b)
  obs. mean       median     s.d.   min.       max.
  4    1970-  01-02 1970-01-02 <NA>   1970-01-01 1970-01-04
```



## Membaca pada sebuah variabel tanggal

Setiap perangkat lunak memiliki cara sendiri dalam membaca tanggal. Mentransfer variabel tanggal dari salah satu perangkat lunak ke perangkat lunak lain terkadang dihasilkan dalam bentuk 'character' untuk tanggal yang tidak dapat langsung dihitung oleh perangkat lunak tersebut.

R dapat dibaca dalam variabel tanggal dari file Stata secara langsung tapi tidak versi lama EpiInfo dengan format <dd/mm/yy>. Hal ini akan dibaca sebagai 'character' atau 'Asis'.

Ketika membaca data dari format file koma dipisahkan variabel (. Csv), hal ini merupakan kebiasaan yang baik untuk menempatkan sebuah argumen 'as.is = TRUE' pada perintah read.csv untuk menghindari variabel tanggal diubah menjadi faktor.

Hal ini diperlukan untuk mengetahui cara membuat variabel tanggal dari format karakter. Buatlah vektor dari tiga tanggal yang disimpan sebagai karakter:

```
> date1 <- c("07/13/2004","08/01/2004","03/13/2005")
> class(date1)
[1] "character"

> date2 <- as.Date(date1, "%m/%d/%Y")
```

Format atau urutan dari karakter asli harus dilihat terlebih dahulu. Dalam elemen pertama dari 'date1', '13 ', yang bisa hanya hari (karena hanya ada 12 bulan), berada di posisi tengah, sehingga'% d 'juga harus di posisi tengah. Garis miring '/' memisahkan bulan, hari dan tahun. Ini harus sejalan dengan format dalam perintah as.Date.

```
> date2
[1] "2004-07-13" "2004-08-01" "2005-03-13"

> class(date2)
[1] "Date"
```

Format tanggal standar adalah "% Y-% m-% d". Untuk mengubah ke format yang umum digunakan dalam Epicalc :

```
> format(date2, "%d%b%y")
[1] "13Jul04" "01Aug04" "13Mar05"
```

Format lain dapat lebih dieksplorasi dengan perintah berikut:

```
> help(format.Date)
> help(format.POSIXct)
```

Dalam hal ini semua hari, bulan dan tahun tidak harus selalu disajikan. Misalnya, jika bulan saja yang akan ditampilkan, Anda dapat mengetik:

```
> format(date2, "%B")
[1] "July"   "August" "March"
```

Untuk memasukkan hari dari minggu

```
> format(date2, "%a-%d%b")
[1] "Tue-13Jul" "Sun-01Aug" "Sun-13Mar"

> weekdays(date2)
[1] "Tuesday" "Sunday"  "Sunday"
```

Sama halnya dengan

```
> format(date2, "%A")
```

Sebaliknya, jika ada dua atau lebih variabel yang merupakan bagian tanggal:

```
> day1 <- c("12","13","14");
> month1 <- c("07","08","12")
> paste(day1, month1)
[1] "12 07" "13 08" "14 12"

> as.Date(paste(day1,month1), "%d %m")
[1] "2007-07-12" "2007-08-13" "2007-12-14"
```

Fungsi Paste menggabungkan dua variabel karakter. Ketika nilai tahun diabaikan, R otomatis menambahkan tahun ini pada sistem dalam komputer.

## Menangani variabel waktu

Sebuah objek Tanggal berisi nilai tahun, bulan dan hari. Untuk waktu, nilai-nilai jam, menit dan detik harus tersedia.

Sebuah sampel dataset yang melibatkan sejumlah variabel waktu dikumpulkan dari peserta lokakarya pada 14 Desember 2004, pertanyaan berkisar tentang

karakteristik pribadi, kapan mereka pergi tidur, bangun, dan tiba di Lokakarya. Lokakarya tersebut dimulai pada pukul 8:30 pagi.

```
> zap()
> data(Timing)
> use(Timing)
```

Catatan:

_____

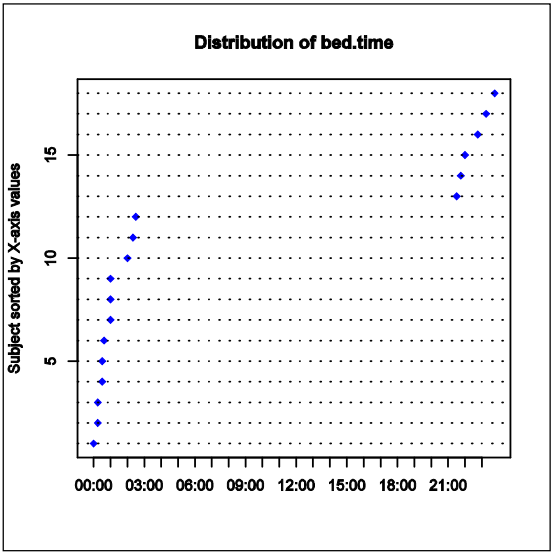
file asli untuk Dataset ini dalam format Stata dan disebut "timing.dta". Jika Anda telah men-download file ini ke direktori kerja (seperti yang dijelaskan dalam bab sebelumnya), Anda hanya dapat mengetikkannya menggunakan ("timing.dta").

```
> des()

Timing questionnaire
No. of observations =18

   Variable      Class         Description
1  id            integer
2  gender        factor
3  age           integer
4  marital       factor
5  child         integer       No. of children
6  bedhr         integer       Hour to bed
7  bedmin        integer       Min. to bed
8  wokhr         integer       Hour woke up
9  wokmin        integer       Min. woke up
10 arrhr         integer       Hour arrived at wkshp
11 arrmin        integer       Min. arrived at wkshp

> summ()

Timing questionnaire

No. of observations = 18

   Var. name Obs.  mean   median  s.d.   min.   max.
1  id        18    9.5    9.5     5.34   1      18
2  gender    18    1.611  2       0.502  1      2
3  age       18    31.33  27.5    12.13  19     58
4  marital   18    1.611  2       0.502  1      2
```

```
5  child     18    0.33   0       0.59   0       2
6  bedhr     18    7.83   1.5     10.34  0       23
7  bedmin    18    19.83  17.5    17.22  0       45
8  wokhr     18    5.61   6       1.61   1       8
9  wokmin    18    23.83  30      17.2   0       49
10 arrhr     18    8.06   8       0.24   8       9
11 arrmin    18    27.56  29.5    12.72  0       50
```

Untuk membuat variabel yang sama dengan waktu para peserta pergi tidur, digunakan fungsi ISOdatetime

```
> bed.time <- ISOdatetime(year=2004, month=12, day=14,
  hour=bedhr, min=bedmin, sec=0, tz="")
> summ(bed.time)
             Min.              Median                Mean
  Max.
2004-12-14 00:00 2004-12-14 01:30 2004-12-14 08:09 2004-12-
  14 23:45
```

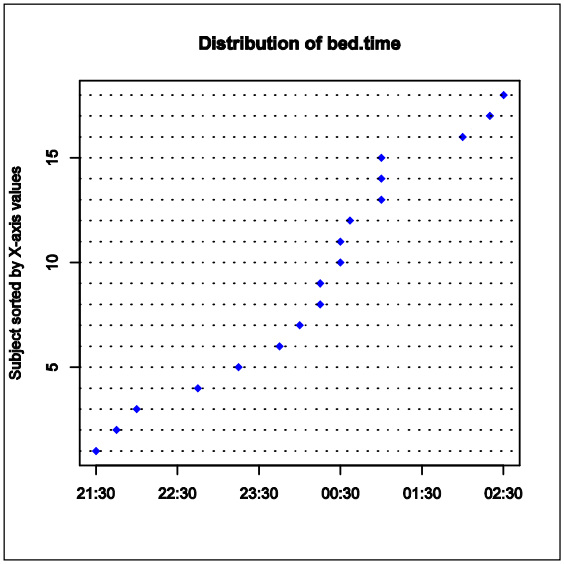


Grafik menunjukkan waktu terganggu. Bahkan, hari harus dihitung berdasarkan waktu peserta pergi tidur. Jika peserta pergi tidur antara 12:00 (tengah hari) dan 12:00 (tengah malam), jadi hari tersebut haruslah 13 Desember, jika tidak hari tersebut haruslah 14 Desember, hari lokakarya. Untuk menghitung ulang jenis hari:

```
> bed.day <- ifelse(bedhr > 12, 13, 14)
```

Fungsi ifelse memilih argumen kedua jika argumen pertama adalah TRUE, ketiga sebaliknya.

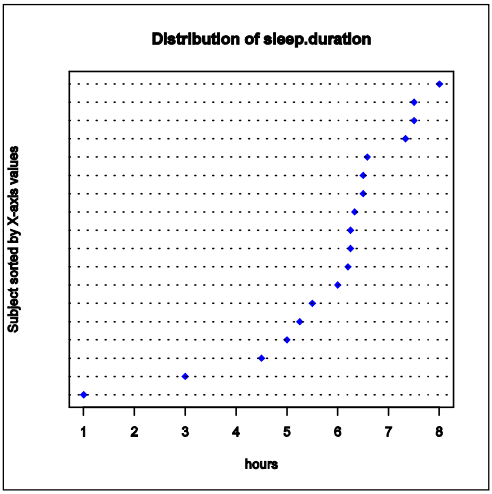
```
> bed.time <- ISOdatetime(year=2004, month=12, day=bed.day,
  hour=bedhr, min=bedmin, sec=0, tz="")

> summ(bed.time)
            Min.            Median              Mean
  Max.
2004-12-13 21:30 2004-12-14 00:22 2004-12-14 00:09 2004-12-
  14 02:30
```




Setelah ini, waktu bangun dan waktu kedatangan dapat dibuat dan diperiksa.

```
> woke.up.time <- ISOdatetime(year=2004, month=12, day=14,
```

```
  hour=wokhr, min=wokmin, sec=0, tz="")

> summ(woke.up.time)
            Min.           Median             Mean
  Max.
2004-12-14 01:30 2004-12-14 06:10 2004-12-14 06:00 2004-12-
  14 08:20

'Woke.up.time' objek terlihat normal, meskipun satu atau dua
  peserta bangun terlalu pagi. Untuk menghitung durasi
  tidur:
> sleep.duration <- difftime(woke.up.time, bed.time)

> summ(sleep.duration)
  Obs.  mean   median  s.d.   min.   max.
  18    5.844  6.25    1.7    1      8
```




Sebuah pilihan yang tepat untuk unit 'sleep.duration' dipilih, tetapi dapat diubah oleh pengguna jika diinginkan. Seseorang tidur sangat sedikit.

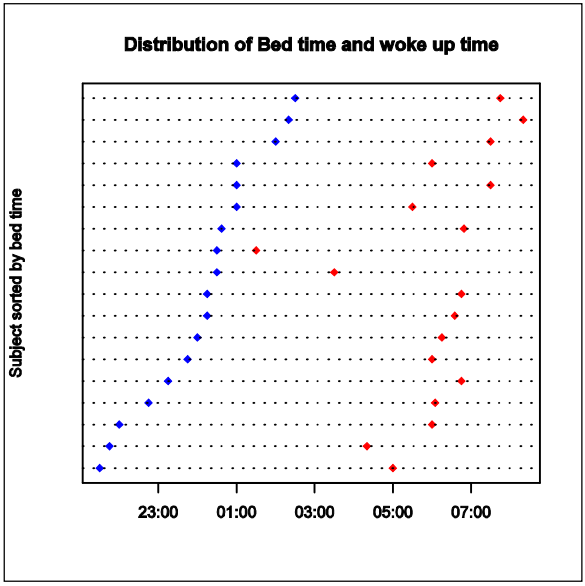
### *Menampilkan dua variabel pada satu grafik*

Perintah summ pada Epicalc tidak sesuai digunakan untuk menampilkan dua variable secara bersamaan. Dotchart asli dari R is the preferred graphical method.

```
> sortBy(bed.time)
> plot(bed.time, 1:length(bed.time),
  xlim=c(min(bed.time),max(woke.up.time)), pch=18,
  col="blue", ylab=" ", yaxt="n")
```

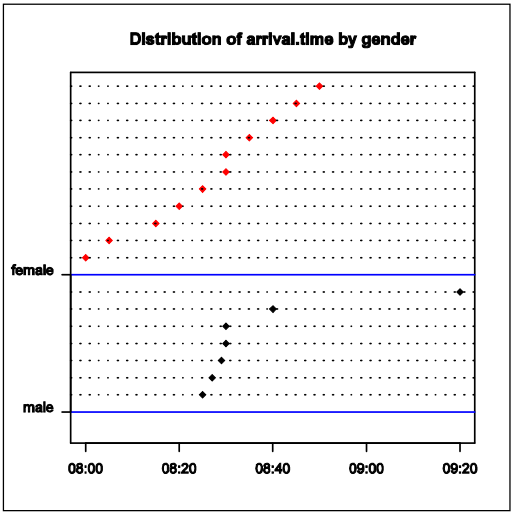
Argumen 'xlim' (batas sumbu-x) diatur menjadi minimum 'bed.time' dan maksimum 'woke.up.time'. Pada yaxt Argumen = "n" terdapat label centang pada sumbu Y-.

```
> n <- length(bed.time)
> segments(bed.time, 1:n, woke.up.time, 1:n)
> points(woke.up.time, 1:n, pch=18, col="red")
> title(main="Distribution of Bed time and Woke up time")
```



Akhirnya, waktu kedatangan di lokakarya dibuat

```
> arrival.time <- ISOdatetime(year=2004, month=12, day=14,
  hour=arrhr, min=arrmin, sec=0, tz="")

> summ(arrival.time)
            Min.           Median             Mean
  Max.
2004-12-14 08:00 2004-12-14 08:30 2004-12-14 08:30 2004-12-
  14 09:20

> summ(arrival.time, by=gender)

For gender = male
            Min.           Median             Mean
  Max.
2004-12-14 08:25 2004-12-14 08:30 2004-12-14 08:37 2004-12-
  14 09:20

For gender = female
            Min.           Median             Mean
  Max.
2004-12-14 08:00 2004-12-14 08:30 2004-12-14 08:26 2004-12-
  14 08:50
```



Perintah summ bekerja relatif baik dengan variabel waktu. Dalam kasus ini, hal ini menunjukkan bahwa perempuan lebih dari laki-laki. Waktu kedatangan untuk wanita cukup bervariasi. Beberapa dari mereka datang lebih awal karena mereka harus mempersiapkan ruang lokakarya. Kebanyakan laki-laki yang tidak mempunyai tugas tiba tepat pada waktunya. Ada satu laki-laki yang sedikit terlambat dan laki-laki yang terlambat hampir satu jam.

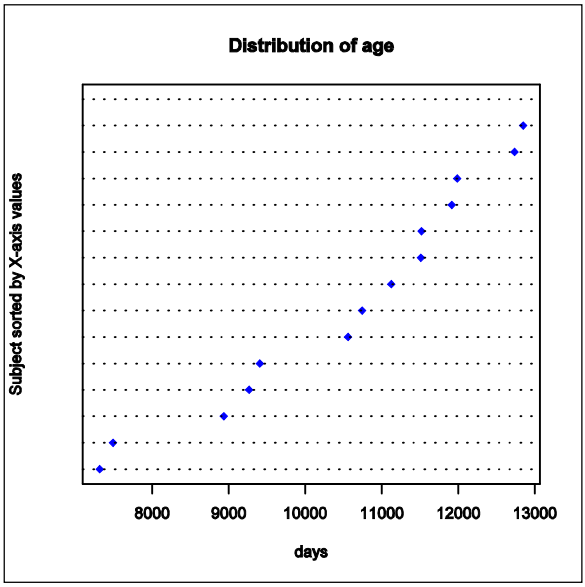
### *Usia dan difftime*

Perhitungan usia dari tanggal lahir biasanya memberikan hasil yang lebih akurat daripada memperoleh usia dari wawancara langsung. Dataset berikut ini berisi tanggal lahir subjek yang dapat kita gunakan untuk mencoba perhitungan usia.

```
> zap()
> data(Sleep3)
> use(Sleep3)
> des()

Sleepiness among the participants in a workshop
No. of observations =15
  Variable     Class        Description
1 id           integer      code
2 gender       factor       gender
3 dbirth       Date         Date of birth
4 sleepy       integer      Ever felt sleepy in workshop
5 lecture      integer      Sometimes sleepy in lecture
6 grwork       integer      Sometimes sleepy in group work
7 kg           integer      Weight in Kg
8 cm           integer      Height in cm
Tanggal Anilisis adalah 13 desember 2004.
> age <- as.Date("2004-12-13") - dbirth

variabel 'age' mempunyai class difftime yang dapat dilihat
  dengan mengetik:
> class(age)
[1] "difftime"

Unit dari  age adalah 'days'.
> attr(age, "unit")
[1] "days"
```

Untuk menampilkan age:

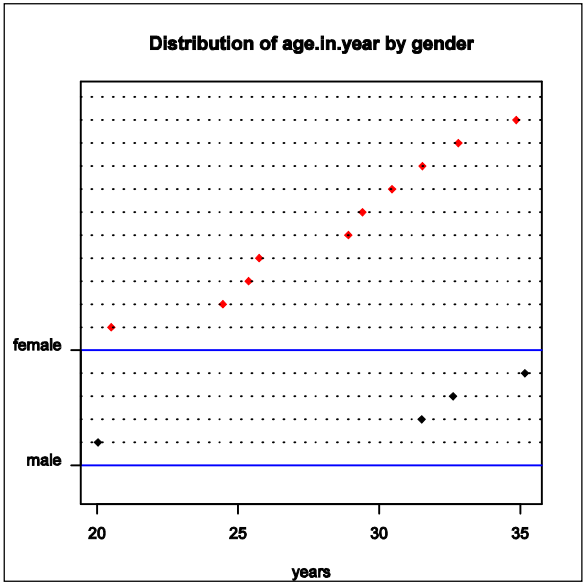
```
> age
Time differences of  7488, 10557,  8934,  9405, 11518,
  11982, 10741, 11122, 12845,  9266, 11508, 12732, 11912,
  7315,    NA days

> summ(age)
  Obs.  mean   median  s.d.    min.   max.
  15    10520  10930   1787.88 7315   12850
```




Perhatikan satu tidak mempunyai nilai. Untuk mengubah age menjadi years:

```
> age.in.year <- as.numeric(age)/365.25
> summ(age.in.year)
  Obs.  mean   median  s.d.   min.   max.
  14    28.81  29.93   4.89   20.03  35.17

> summ(age.in.year, by=gender)
For gender = male
  Obs.  mean   median  s.d.   min.   max.
  4     29.83  32.06   6.712  20.03  35.17
```

```
For gender = female
  Obs.   mean    median   s.d.    min.    max.
    10    28.4    29.16    4.353   20.5    34.86
```




Perhatikan bahwa ada garis putus-putus kosong di bagian atas kelompok perempuan. Ini nilai yang hilang. Pria memiliki ukuran sampel yang jelas lebih kecil dengan kisaran yang sama dengan perempuan, tetapi pengamatan memiliki nilai yang relatif tinggi.

## Latihan

Dalam dataset Waktu:

Hitung waktu sejak bangun hingga kedatangan di lokakarya.

Plot waktu tidur, waktu bangun dan waktu kedatangan pada sumbu yang sama.

BAB 7

# Investigasi Wabah: Gambaran Waktu

Menginfestigasi wabah adalah tugas umum seorang epidemiologis. Bab ini menjelaskan bagaimana data dapat digambarkan secara efektif. Waktu dan tanggal dari tipe data tidak dipersiapkan dengan baik dan harus dimodifikasi lebih lanjut untuk memenuhi kebutuhan analisis deskriptif.

Pada tanggal 25 Agustus 1990, petugas kesehatan lokal di Provinsi Buri Supan Thailand melaporkan terjadinya wabah penyakit gastrointestinal akut pada hari olahraga penyandang cacat nasional. Dr Lakkana Thaikruea dan rekan-rekannya melakukan investigasi. Dataset tersebut disebut **Outbreak**.. Kebanyakan nama variabel telah cukup jelas. Variabel yang dikodekan sebagai 0 = tidak, 1 = ya dan 9 = hilang / tidak diketahui, untuk tiga item makanan yang dikonsumsi oleh peserta: 'beefcurry' (daging sapi kari), 'saltegg' (telur asin) dan 'air'. Juga pada menu kue sus, kue sebesar jari yang disi dengan kocokan susu dan dibungkus dengan lapisan gula. Variabel ini mencatat jumlah potongan yang dimakan oleh setiap peserta. Nilai yang hilang diberi kode sebagai berikut: 88 = "dimakan tapi tidak ingat berapa banyak", sedangkan kode 90 menunjukkan informasi yang benar-benar hilang (tidak diingat). Beberapa peserta mengalami gejala gastrointestinal, seperti: mual, muntah, sakit perut

dan diare. Usia masing-masing peserta dicatat dalam tahun dengan 99 mewakili nilai yang hilang. variabel 'Exptime' dan 'onset' adalah paparan dan waktu timbulnya gejala, dalam format karakter, atau 'Asis' dalam R terminologi.

## *Pencarian Cepat*

Mari kita lihat data. Ketik sintak berikut di konsol R:

```
> zap()
> data(Outbreak)
> use(Outbreak)
> des()

No. of observations =1094

  Variable     Class          Description
1  id          numeric
2  sex         numeric
3  age         numeric
4  exptime     AsIs
5  beefcurry   numeric
6  saltegg     numeric
7  eclair      numeric
8  water       numeric
9  onset       AsIs
10 nausea      numeric
11 vomiting    numeric
12 abdpain     numeric
13 diarrhea    numeric

> summ()

No. of observations = 1094

  Var. name  valid obs. mean   median  s.d.   min.   max.
1  id        1094      547.5  547.5   315.95  1      1094
2  sex       1094      0.66   1       0.47    0      1
3  age       1094      23.69  18      19.67   1      99
4  exptime
5  beefcurry 1094      0.95   1       0.61    0      9
```

```
6  saltegg   1094     0.96  1     0.61   0     9
7  eclair    1094    11.48  2    27.75   0    90
8  water     1094     1.02  1     0.61   0     9
9  onset
10 nausea    1094     0.4   0     0.49   0     1
11 vomiting  1094     0.38  0     0.49   0     1
12 abdpain   1094     0.35  0     0.48   0     1
13 diarrhea  1094     0.21  0     0.41   0     1
```

Pertama kita tentukan kasus nya, memeriksa waktunya akan dilakukan dalam bab ini dan menyelidiki penyebabnya pada bagian berikutnya.

## Definisi kasus

Telah disepakati di kalangan para peneliti bahwa sebuah kasus harus didefinisikan sebagai orang yang memiliki salah satu dari empat gejala: 'mual', 'muntah', 'abdpain' atau 'diare'. Sebuah kasus dapat dihitung sebagai berikut:

```
> case <- (nausea==1)|(vomiting==1)|(abdpain==1)|(diarrhea==1)
```

Untuk memasukkan variabel baru ke dalam .data, kita menggunakan fungsi label.var, yang akan dijelaskan secara rinci dalam Bab 10..

```
> label.var(case, "diseased")
```

obyek ‘case’ sekarang dimasukkan ke dalam data sebagai variabel ke 14 beserta deskripsi variabel. Perhatikan bahwa kelas harus *logical* (logis).

```
> des()
```

### *Waktu Terpapar*

Untuk menghitung waktu terpapar, pertama mari lihat struktur variabel berikut.

```
> str(exptime)
Class 'AsIs' chr [1:1094] "25330825180000" "25330825180000"...
```

Nilai dari variabel ini berisi empat belas digit. Empat digit pertama mewakili tahun wabah di Era Buddhis (BE) kalender, yang sama

dengan AD + 543. Angka 5 dan 6 berisi dua digit yang mewakili bulan, 7 dan 8 mewakili hari, 9 dan 10 jam , 11 dan 12 menit dan 13 dan 14 detik.

```
> day.exptime <- substr(exptime, 7, 8)
```

Perintah R susbtr (dari substring), adalah untuk mengekstrak bagian karakter dari vektor. Pertama, mari kita lihat pada hari paparan.

```
> tab1(day.exptime)
day.exptime :
      Frequency   %(NA+) cum.%(NA+)   %(NA-) cum.%(NA-)
25         1055    96.4      96.4     100       100
<NA>         39     3.6     100.0       0       100
 Total     1094   100.0     100.0     100       100
```

Hasil hari terpapar adalah 25 Agustus untuk semua catatan/record (abaikan 39 nilai yang hilang). Kita dapat mengambil waktu terpapar dengan cara yang sama.

```
> hr.exptime <- substr(exptime, 9, 10)
> tab1(hr.exptime)
```

Semua nilai tampaknya dapat diterima, dengan mode pada 18 jam.

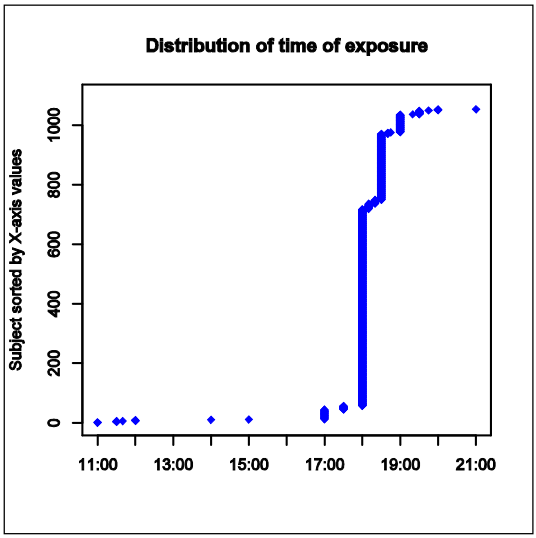
```
> min.exptime <- substr(exptime, 11, 12)
> tab1(min.exptime)
```

Ini juga dapat diterima, meskipun diketahui bahwa kebanyakan menit telah dibulatkan ke jam terdekat atau setengah jam. Sekarang waktu paparan sudah dapat dihitung.

```
> time.expose <- ISOdatetime(year=1990, month=8, day=day.exptime, hour=hr.exptime,
    min=min.exptime, sec=0)
```

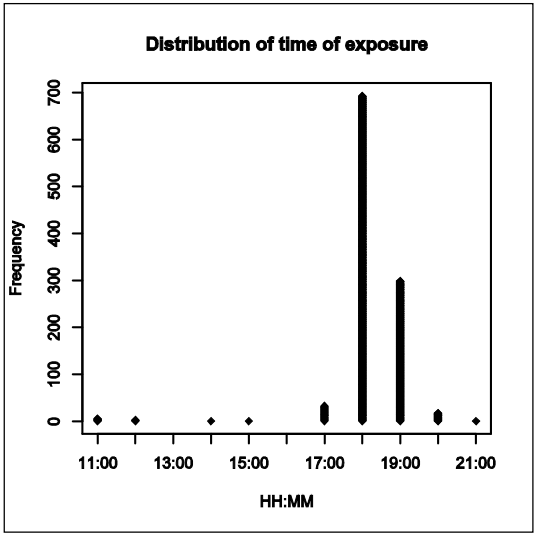
Kemudian, variabel diberi label dalam rangka untuk mengintegrasikan ke dalam frame data default.

```
> label.var(time.expose, "time of exposure")
> summ(time.expose)
        Min.          Median          Mean          Max.
1990-08-25 11:00 1990-08-25 18:00 1990-08-25 18:06  1990-08-25 21:00
```




Sebuah plot titik juga dapat dihasilkan.

```
> dotplot(time.expose)
```



Hampir seluruh waktu paparan terjadi selama makan malam; 06:00-7:00,sementara hanya sedikit yang terjadi selama makan siang.

### *Timing the onset (waktu timbulnya gejala)*

Eksplorasi data menunjukkan bahwa tiga non-kasus memiliki non-blank waktu terjadinya.

```
> sum(!is.na(onset[!case]))  # 3
```

Fungsi is.na mengidentifikasi elemen dalam vektor yang memiliki NA. Untuk sederhananya kita pastikan bahwa 'onset' variable secara eksklusif digunakan hanya untuk kasus-kasus saja.

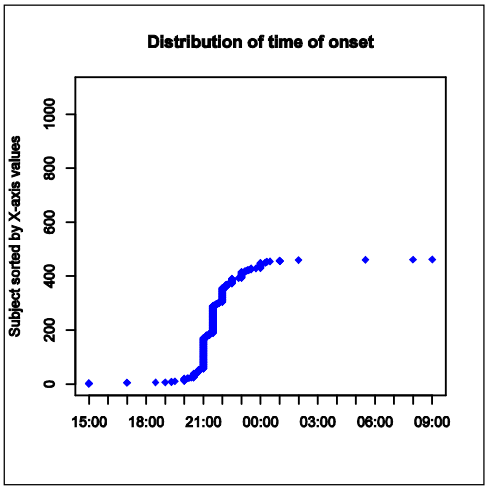
```
> onset[!case] <- NA
```

Ekstraksi (pencarian) waktu timbulnya gejala mirip dengan waktu terpapar.

```
> day.onset <- substr(onset, 7, 8)
> tab1(day.onset)
day.onset :
      Frequency   %(NA+) cum.%(NA+)   %(NA-) cum.%(NA-)
25          429    39.2      39.2    92.9      92.9
26           33     3.0      42.2     7.1     100.0
<NA>        632    57.8     100.0     0.0     100.0
 Total     1094   100.0     100.0   100.0     100.0
```

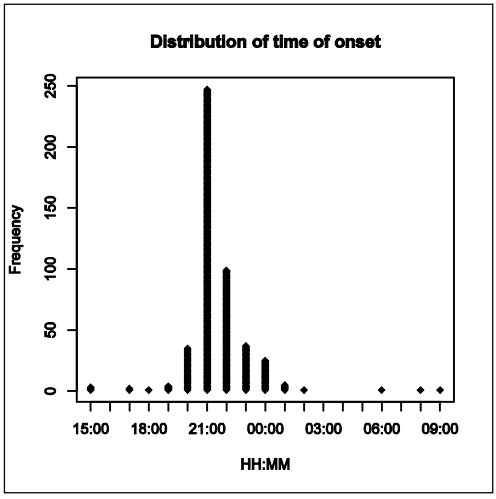
Dari subyek yang diwawancarai, 57,8% tidak memiliki 'onset' dan setelah itu didapatkan variabel 'day.onset' . Hal ini disebabkan subjek tidak bisa ingat dengan baik apakah mereka memiliki gejala atau tidak. Di antara mereka yang melaporkan waktu terjadinya, 429 terjadi pada 25 Agustus. Dan 33 yang tersisa terjadi pada hari setelahnya.

```
 > hr.onset <- substr(onset, 9, 10)
> tab1(hr.onset)
> min.onset <- substr(onset, 11, 12)
> tab1(min.onset)
> time.onset <- ISOdatetime(year=1990, month=8, day=day.onset, hour=hr.onset,
    min=min.onset, sec=0, tz="")
> label.var(time.onset, "time of onset")
> summ(time.onset)
```




```
        Min.         Median         Mean         Max.
1990-08-25 15:00 1990-08-25 21:30 1990-08-25 21:40  1990-08-26 09:00
```

Bagian atas grafik kosong karena banyak nilai-nilai yang hilang . Mungkin tampilan visual yang lebih baik dapat diperoleh dengan menggunakan plot titik.



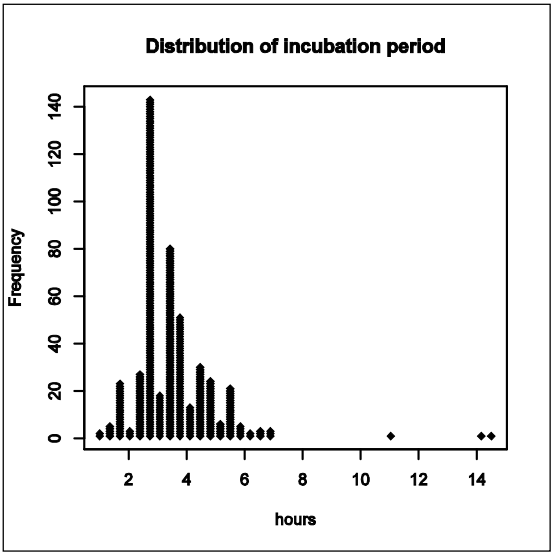
Kedua grafik di atas merupakan kurva single-peak klasik dari penyebaran penyakit, yang menunjukkan satu sumber penyebab. Kasus yang paling awal terjadi pada pukul 03:00 sore tanggal 25 Agustus. Mayoritas kasus mulai merasa sakit pada waktu tengah malam. Pada pagi berikutnya, hanya beberapa kasus yang terjadi. Kasus terakhir yang dilaporkan terjadi pada pukul 09:00 tanggal 26 Agustus.

## *Incubation period (Masa Inkubasi)*

Analisis untuk masa inkubasi sangatlah mudah.

```
> incubation.period <- time.onset - time.expose
> label.var(incubation.period, "incubation period")
> summ(incubation.period)
 Valid obs. mean   median s.d.   min.   max.
 462        3.631  3.5    1.28   1      14.5
> dotplot(incubation.period, las=1)
```

Masa inkubasi memilki median 3,5 jam dengan kemiringan ke kanan.



## Plot berpasangan *(Paired plot)*

Kita sekarang mencoba menempatkan waktu paparan dan waktu terjadinya gejala pada grafik yang sama. Sebuah grafik yang terurut biasanya memberikan lebih banyak informasi, sehingga seluruh data frame sekarang berurut.

```
> sortBy(time.expose)
```

Dengan ukuran sampel yang besar, ada baiknya grafik dibatasi hanya untuk mem plot waktu paparan 'time.exposure' dan waktu timbulnya gejala 'time.onset'. Penggabungan ini disimpan sebagai data frame lain yang disebut 'data.for.graph'.

```
> data.for.graph <- subset(.data, (!is.na(time.onset) & !is.na(time.expose)), select =
    c(time.onset, time.expose))

> des(data.for.graph)
No. of observations =462
  Variable     Class          Description
1 time.onset   POSIXt
2 time.expose  POSIXt
```

Hanya ada dua variabel dalam data frame. Semua nilai-nilai yang tidak diketahui telah dihapus dan hanya tersisa 462 catatan untuk di plotkan.

```
> n <- nrow(data.for.graph)
> with(data.for.graph, {
    plot(time.expose, 1:n, col="red", pch=20,
    xlim = c(min(time.expose), max(time.onset)),
    main = "Exposure time & onset of food poisoning outbreak",
    xlab = "Time (HH:MM)", ylab = "Subject ID" )
 } )
```

Pola plot terlihat mirip dengan yang dihasilkan oleh summ (time.expose). Karakter titik, "PCH ', diatur menjadi 20, dimana plotnya berupa lingkaran padat dan kecil, untuk menghindari terlalu banyak tumpang tindih titik-titik. Batas-batas pada sumbu horizontal adalah dari waktu minimum paparan sampai maksimum waktu kejadian, sehingga memungkinkan titik kejadian untuk diletakkan pada grafik yang sama. Titik-titik ini ditambahkan dalam perintah berikut:

```
> with(data.for.graph, {
```

```
 points(time.onset, 1:n, col="blue", pch=20)
 } )
```

Dua set titik dipasangkan oleh banyak subjek. Sebuah garis yang menghubungkan masing-masing pasangan sekarang ditarik oleh perintah segmen.

```
> with(data.for.graph, {
 segments(time.expose, 1:n, time.onset, 1:n, col = "grey45")
 } )
```

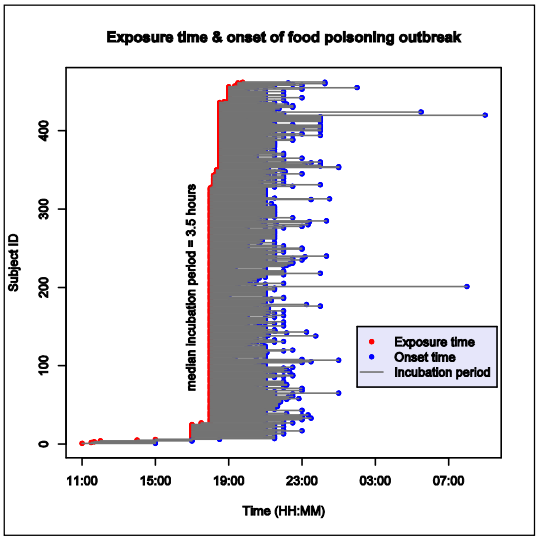
Daftar lengakap dari nama warna yang dapat digunakan di R dapat ditemukan di colours().
Sebuah legenda dimasukkan untuk membuat grafik tampak lebih jelas.

```
> legend(x = ISOdatetime(1990,8,26,2,0,0), y = 150,
   legend=c("Exposure time","Onset time","Incubation period"),
   pch=c(20,20,-1), lty=c(0,0,1),col=c("red","blue","grey45"),
   bg="lavender")
```

Sudut kiri atas legenda terletak di kuadran kanan bawah grafik dengan koordinat x di 2 am dan y di 150. Legenda ini terdiri dari tiga item seperti yang ditunjukkan oleh karakter vektor. Karakter titik dan warna dari legenda di ditentukan sesuai dengan yang ada di dalam grafik. Argumen terakhir, masa inkubasi, 'PCH' sama dengan -1 menunjukkan tidak ada titik yang digambar. Jenis garis,'lty', dari paparan dan waktu kejadian adalah 0 (tidak ada garis) sedangkan untuk periode inkubasi adalah 1 (garis padat/utuh). Warna-warna dari titik-titik dan garis sesuai dengan yang ada di grafik. Latar belakang legenda diberi warna lavender untuk menggantikan setiap garis atau titik balik legenda. Akhirnya, beberapa teks yang menjelaskan statistik kunci dari variabel ini ditempatkan di dalam area plot pada 5 pm dan berpusat pada 200.

```
> text(x = ISOdatetime(1990, 8, 25, 17, 0, 0), y = 200, labels = "median incubation period
   = 3.5 hours", srt = 90)
```

Bagian tengah dari teks dalam grafik terletak pada x = 19:00 dan y = 200 dalam grafik. Parameter'srt' berasal dari 'rotasi string'. Dalam kasus ini rotasi 90 derajat akan menghasilkan gambar yang terbaik. Karena warna latar belakang sudah abu-abu, teks putih akan cocok.




Analisis dari waktu data telah selesai. Frame data utama .Data disimpan agar dapat digunakan kembali pada bab selanjutnya.

```
> save(.data, file = "Chapter7.Rdata")
```

## *Referensi*

Thaikruea, L., Pataraarechachai, J., Savanpunyalert, P., Naluponjiragul, U. 1995 An unusual outbreak of food poisoning. Southeast Asian J Trop Med Public Health 26(1):78-85.

## Latihan

Kita catat waktu asli variabel ‘onset’ kanan dari awal menggunakan perintah:

```
> Onset [kasus!] <- NA
```

Unutk data frame yang kita lewatkan untuk bab selanjutnya, apakah variabel ‘onset’ berubah? Jika tidak, mengapa dan bagaimana perubahan tetap dari data frame yang kita gunakan?

Catatan: Dataset **Outbreak** yang dibangun tidak boleh dimodifikasi.

BAB 8

# Investigasi Wabah: Penilaian Resiko

Langkah selanjutnya dalam menganalisis wabah adalah penyesuaian dengan level resiko. Namun, pertama mari kita memuat data yang disimpan dari bab sebelumnya.

```
> zap()
> load("Chapter7.Rdata")
> ls(all=TRUE)      # .data is there
> search()    # No dataset in the search path
> use(.data)
> search()    # .data is ready for use
> des()
```

## Recoding data hilang

Terdapat sejumlah variable yang perlu direcoding. Variable pertama yang di recoding adalah 'age'. Perintah Epicalc *recode* digunakan disini. Fungsi ini akan dijelaskan lebih detail pada chapter 10.

```
> recode(var = age, old.value = 99, new.value = NA)
```

Variable dengan skema recoding yang sama, dengan 9 data hilang, adalah 'beefcurry', 'saltegg' dan 'air'. Mereka dapat direcode ulang bersama-sama dalam satu langkah sebagai berikut:

```
> recode(vars = c(beefcurry, saltegg, water), 9, NA)
```

Ketiga variabel dapat juga diubah menjadi faktor dengan label nilai yang melekat.

```
> beefcurry <- factor(beefcurry, labels=c("No","Yes"))
> saltegg <- factor(saltegg, labels=c("No","Yes"))
> water <- factor(water, labels=c("No","Yes"))
> label.var(beefcurry, "Beefcurry")
> label.var(saltegg, "Salted egg")
> label.var(water, "Water")
```

Untuk 'eclair', nilai hilang absolute adalah 90. Ini yang harus direcoding pertama kali, kemudian periksa kembali data frame untuk nilai yang hilang.

```
> recode(eclair, 90, NA)
> summ()
```

Keseluruhan variabel kelihatan normal kecuali 'eclair' yang masih mengandung nilai 80, ini berarti " makan tetapi tidak mengingat berapa banyak yang dimakan". Kita akan menganalisis hubungan tersebut dengan fungsi 'case' dengan mempertimbangkan variabel 'eclair' sebagai variabel kategori berurut.

Pada tahap ini, tabulasi silang dapat ditampilkan dengan menggunakan perintah Epicalc *tabpct*.

```
> tabpct(eclair, case)
```

Lebar kolom grafik mosaik diatas menunjukkan frekuensi relatif dari kategori tersebut. Frekuensi tertinggi adalah 2 potong diikuti oleh 0 dan 1 . Angka-angka lainnya memiliki frekuensi yang relatif rendah, terutama 5 catatan di mana 'Eclair' diberi kode sebagai 80.

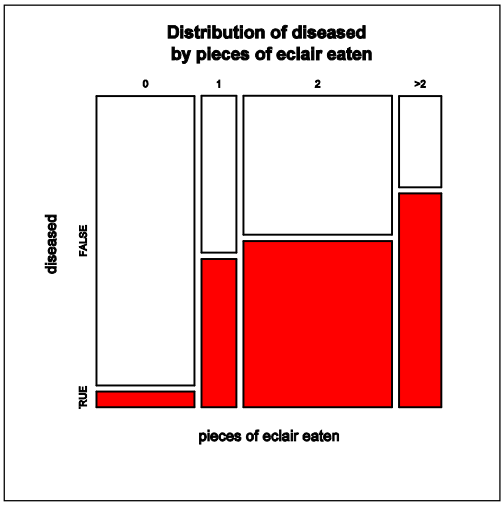
Ada kecenderungan peningkatan area merah atau tingkat serangan dari kiri ke kanan yang menunjukkan bahwa risiko telah meningkat ketika lebih banyak potongan kue sus yang dikonsumsi. Kami akan menggunakan distribusi dari proporsi ini untuk membentuk kelompok konsumsi kue sus. Kolom pertama dari konsumsi nol memiliki tingkat serangan yang sangat rendah, oleh karena itu kolom tersebut harus merupakan kategori yang terpisah. Hanya sedikit yang mengambil setengah potong dan ini dapat dikombinasikan dengan orang yang mengambil satu potong kue. Orang yang mengkonsumsi 2 potong harus dimasukkan dalam satu kategori karena mereka memiliki frekuensi yang tinggi. Lainnya yang mengkonsumsi lebih dari 2 potong harus dikelompokkan dalam kategori lainnya. Akhirnya yang dikodekan sebagai '80' akan dikeluarkan karena jumlah konsumsi yang tidak diketahui serta frekuensinya yang rendah.

```
> eclairgr <- cut(eclair, breaks = c(0, 0.4, 1, 2, 79),
  include.lowest = TRUE, labels=c("0","1","2",">2"))
```

Argumen 'include.lowest' diatur menjadi TRUE untuk menunjukkan bahwa Éclair 0 harus termasuk dalam kategori terendah.

Untuk latihan melabelkan variabel baru agar dapat menggambarkan serta memasukkannya kedalam **`.data`**, perintah *`label.var`* dapat digunakan.

```
> label.var(eclairgr, "pieces of eclair eaten")
> tabpct(eclairgr, case)
======== lines omitted =========
Row percent
                       diseased
pieces of eclair eaten   FALSE    TRUE  Total
                     0      279      15    294
                         (94.9)   (5.1)  (100)
                     1       54      51    105
                         (51.4)  (48.6)  (100)
                     2      203     243    446
                         (45.5)  (54.5)  (100)
                     >2      38      89    127
                         (29.9)  (70.1)  (100)
 ======== lines omitted =========
```




Laju atau persentase serangan penyakit dalam setiap kategori penyebaran, seperti ditunjukkan dalam golongan dari kolom TRUE, meningkat dari 5.1% diantara mereka yang tidak mengkonsumsi kue apapun hingga 70.1% diantara mereka yang banyak memakan kue sus. Output grafik yang diperoleh sama dengan sebelumnya kecuali kelompoknya yang lebih ringkas.

Sekarang kita punya variabel kontinu 'eclair' dan variabel kategori 'eclairgr'. Langkah selanjutnya adalah membuat sebuah penyebaran biner untuk kue sus.

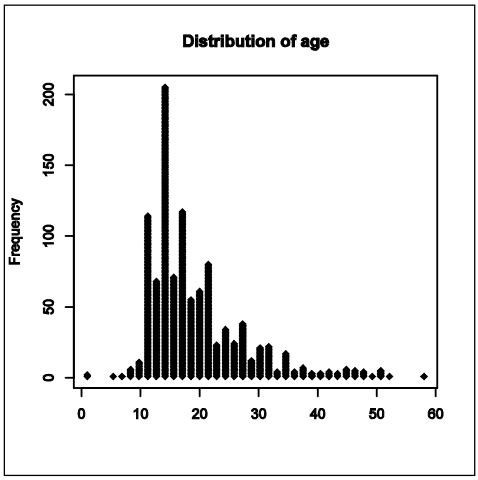
```
> eclair.eat <- eclair > 0
> label.var(eclair.eat, "eating eclair")
```

Variabel penyebaran dikotomi tidak sama dengan yang lainnya, ('beefcurry', 'saltegg' and 'water').

## Eksplorasi usia dan jenis kelamin

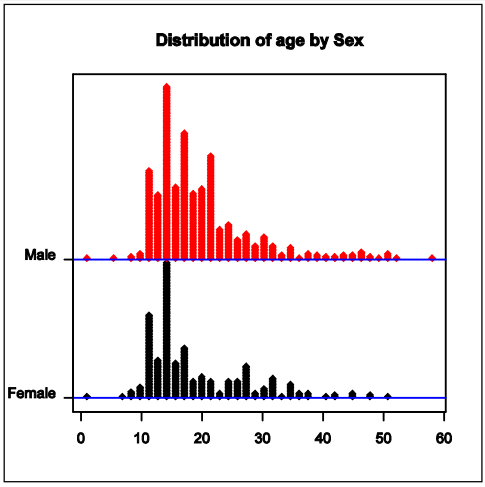
Eksplorasi sederhana pada usia dapat dilakukan dengan menggunakan perintah *summ* dan *dotplot* seperti berikut :

```
> summ(age); dotplot(age)
```



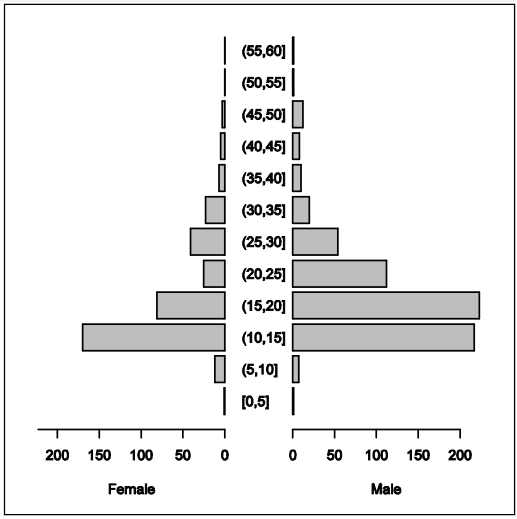

Distribusi usia diklasifikasikan berdasarkan jenis kelamin dapat dengan mudah dilakukan melalui:

```
> sex <- factor(sex, labels=c("Female","Male"))
> summ(age, by = sex)
> dotplot(age, by = sex)
```



Alternatif untuk menggambar piramida penduduk usia dan jenis kelamin, dengan menggunakan fungsi Epicalc *pyramid*, sebagai berikut:

```
> pyramid(age, sex)
```




Dari grafik hasil, laki-laki muda dewasa (usia 10-20 tahun) mendominasi. Lebar batang dapat pula dirubah sehingga memiliki kelompok usia yang lebih sedikit.

```
> pyramid(age, sex, binwidth = 15)
```

Tabel yang dihasilkan fungsi piramida dapat juga ditampilkan dengan cara berikut:

```
> pyramid(age, sex, printTable=TRUE)

Tabulation of age by sex (frequency).
         sex
age       Female Male
  [0,5]        1    1
  (5,10]      12    7
  (10,15]    170  217
  (15,20]     81  223
  (20,25]     25  112
  (25,30]     41   54
  (30,35]     23   20
  (35,40]      7   10
```

```
  (40,45]      5    8
  (45,50]      3   12
  (50,55]      0    1
  (55,60]      0    1
```

Persentase (untuk setiap jenis kelamin) dapat juga ditampilkan.

```
> pyramid(age, sex, printTable=TRUE, percent="each")
  Tabulation of age by sex (percentage of each sex).
        Female   Male
[0,5]    0.272  0.150
(5,10]   3.261  1.051
(10,15] 46.196 32.583
(15,20] 22.011 33.483
(20,25]  6.793 16.817
(25,30] 11.141  8.108
(30,35]  6.250  3.003
(35,40]  1.902  1.502
(40,45]  1.359  1.201
(45,50]  0.815  1.802
(50,55]  0.000  0.150
(55,60]  0.000  0.150
```

Akhirnya, kedua table dan kelompok umur dapat disimpan sebagai **R** objects untuk keperluan mendatang.

```
> (age.tab <- pyramid(age, sex))
> ageGrp <- age.tab$ageGroup
> label.var(ageGrp, "Age Group")
> des()
> des("age*")

No. of observations =1094
   Variable      Class           Description
3  age           numeric
20 ageGrp        factor          Age Group
```

Fungsi *des* dapat juga menampilkan variabel dengan menggunakan wild card matching.

```
> des("????????")

No. of observations =1094
   Variable      Class           Description
11 vomiting      numeric
13 diarrhea      numeric
```

```
18 eclairgr      factor          pieces of eclair eaten
```

Kami telah menghabiskan waktu untuk belajar fitur-fitur dari Epicalc untuk eksplorasi data. Mari kita kembali ke analisis risiko, yang merupakan fitur utama dari Epicalc.

## Perbandingan Resiko: Risk Rasio dan resiko yang ditimbulkan

Pada dasarnya ada dua metode untuk membandingkan risiko penyakit dalam kelompok sebaran yang berbeda.

Rasio resiko – RR (disebut juga relative risk) merupakan rasio resiko terserang penyakit bagi yang telah terserang (exposed) dibandingkan dengan yang tidak terserang penyakit (non-exposed). Hal itu mengindikasikan berapa kali resiko akan meningkat selama penderita mengubah status dari exposed menjadi non-exposed. Peningkatan dianggap dalam perkalian, sehingga dalam notasi matematika disebut model multiplikatif.

Dalam sisi lain resiko menunjukkan jumlah resiko yang diperoleh atau hilang seiring penderita berubah dari exposed menjadi non-exposed. Peningkatannya absolut dan memiliki model aditif dalam notasi matematika.

Perintah Epicalc *cs* digunakan untuk menganalisis hubungan semacam ini.

```
> cs(case, eclair.eat)

       eating eclair
case     FALSE TRUE Total
  FALSE  279   300  579
  TRUE   15    383  398
  Total  294   683  977

         Rne   Re   Rt
  Risk   0.05  0.56 0.41
                                      Estimate Lower95
  Upper95
Risk difference (attributable risk)   0.51     0.44    0.58
Risk ratio                            10.99    8       15.1
Attr. frac. exp. -- (Re-Rne)/Re       0.91
Attr. frac. pop. -- (Rt-Rne)/Rt*100 % 87.48
```

'Rne', 'Re' dan 'Rt' merupakan resiko non-exposed, resiko exposed dan total populasi. 'Rne' dalam hal ini adalah 15/294 = 0.05. Demikian juga 'Re' adalah 383/683 = 0.56 dan 'Rt' senilai 398/977 = 0.41. Selisih resiko adalah 'Re' - 'Rne', peningkatan absolute 50% sementara rasio resiko 'Re' / 'Rne', peningkatan sebesar 11 kali lipat. Resiko terserang penyakit pada orang yang memakan kue sus bisa saja berkurang sebesar 91% dan resiko diantara keseluruhan peserta dalam karnaval olahraga yang tidak mengkonsumsi kue sus dapat juga berkurang sebesar 87.5%.

Risk ratio merupakan indikator penting untuk sebab-akibat. Rasio risiko di atas 10 sangat menyarankan sebuah hubungan sebab-akibat.

Selisih resiko memiliki implikasi lebih terhadap kesehatan masyarakat dibandingkan dengan rasio resiko. Rasio resiko yang tinggi mungkin tidak menjadi kepentingan dalam kesehatan masyarakat jika penyakit sangat jarang terjadi. Sedangkan selisih resiko mengukur secara langsung masalah kesehatan dan kebutuhan pelayanan kesehatan. Mereka yang mengkonsumsi kue sus memiliki peluang yang besar (55%) menderita gejala. Penurunan 51% secara substansial mengurangi beban peserta permainan olahraga dan pelayanan rumah sakit.

Perbedaan fraksi populasi menunjukkan bahwa sejumlah kasus dapat dikurangi sebesar 87% pada kue sus yang belum terkontaminasi. Wabah ini berlaku sementara jika kita bandingkan dengan masalah kronis seperti penyakit kardiovaskular dan kanker. Bahkan level yang relatif rendah dari fraksi perbedaan resiko dalam populasi tembakau, katakanlah 20%, dapat menyebabkan sejumlah besar sumber daya dihabiskan dalam pelayanan kesehatan.

Persebaran perbedaan fraksi memiliki sedikit hubungan dengan tingkat penyebaran penyakit dalam populasi. Hal ini sama dengan $1 - RR^{-1}$, dan ini merupakan cara lain untuk menampilkan rasio resiko.

Kita punya kue sus sebagai penyebab penyakit. Ada beberapa intervensi yang dapat mencegah penyebaran penyakit seperti vaksinasi, pendidikan, penegakan hukum dan perbaikan lingkungan. Dalam contoh berikut ini, mari asumsikan bahwa tidak memakan kue sus sebagai proses pencegahan.

```
s> eclair.no <- !eclair.eat    # The ! sign means "NOT"
> cs(case, eclair.no)
```

```
       eclair.no
case    FALSE TRUE Total
  FALSE 300   279  579
  TRUE  383   15   398
  Total 683   294  977

       Rne   Re   Rt
  Risk  0.56  0.05 0.41
                                      Estimate Lower95  Upper95
 Risk difference (absolute change) -0.51    -0.44    -0.58
 Risk ratio                         0.09     0.12     0.07
 protective efficacy (%)            90.9
 Number needed to treat (NNT)       1.96
```

Resiko antara exposed (tidak mengkonsumsi kue sus) lebih rendah dibandingkan non-exposed (mengkonsumsi kue sus). Selisih resiko berubah tanda menjadi negative. Rasio resiko menuju nilai yang kecil 0.09. Meskipun tampilan sebaran fraksi berbeda dan populasi fraksi berbeda, perintah menunjukkan keberhasilan pencegahan dan jumlah yang diperlukan untuk diobati (needed to treat (NNT)).

Dari nilai keberhasilan pencegahan, sebaran untuk program pencegahan resiko komsumer kue sus telah dikurangi (unexposed dibawah kondisi hipotikal) sebesar 90.9%. NNT hanya kebalikan dari negatif selisih resiko. Pengurangan resiko 0.51 muncul dari intervensi dalam satu individual. Penurunan 1 diharapkan muncul dari intervensi pada individual 1/0.51 atau 1.96. Intervensi NNT yang tinggi akan dibutuhkan untuk dibagikan ke banyak individual untuk menghindari kejadian yang tidak diinginkan. Level terendah yang paling mungkin dari NNT adalah 1 atau pencegahan sempurna yang juga mempunyai efektifitas perlindungan 100%. NNT adalah bagian pengukuran kelayakan teknologi intervensi (baik pencegahan maupun pengobatan). Untuk menghindari tipe serupa dari kejadian yang tidak diinginkan, intervensi dengan NNT yang rendah lebih disukai daripada NNT yang tinggi, meskipun biaya juga harus diperhitungkan.

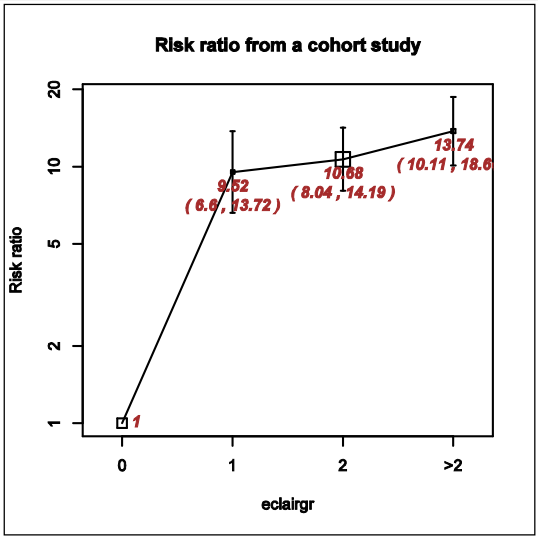
## Hubungan Dosis-respons

Salah satu kriteria untuk sebab-akibat adalah bukti adanya hubungan dosis-respon. Jika Jika penyebaran dosis lebih tinggi dikaitkan dengan tingkat resiko

yang lebih tinggi secara linear, maka penyebaran tersebut mungkin menjadi penyebab.

Kita sekarang mengeksplorasi hubungan antara risiko terkena penyakit dan jumlah kue sus yang dikonsumsi.

```
> cs(case, eclairgr)
                eclairgr
case             0    1     2     >2
  FALSE          279  54    203   38
  TRUE           15   51    243   89

  Absolute risk  0.05 0.49  0.54  0.7
  Risk ratio     1    9.52  10.68 13.74
  lower 95% CI        6.6   8.04  10.11
  upper 95% CI        13.72 14.19 18.66

Chi-squared = 237.12 , 3 d.f., P value = 0
Fisher's exact test (2-sided) P value = 0
```




Rasio resiko meningkat seiring meningkatnya dosis penyebaran kue sus. Tingkatan dari tidak mengkonsumsi kue sus menjadi kelompok pertama (konsumsi diatas dua potong kue) cukup luas sedangkan untuk peningkatan lebih jauh ditunjukkan pada slope yang agak mendatar. P-value pada output

untuk keduanya samadengan nol. Pada kenyataannya, kedua nilai tersebut tidak benar-benar bernilai 0, tetapi telah dibulatkan sampai 3 desimal. Pembulatan desimal dari odd rasio dan resiko relatif adalah dua dan P-value bernilai tiga. Lihat halaman bantuan untuk informasi argumen lebih lanjut.

Sebelum menyelesaikan bab ini, data saat ini disimpan untuk penggunaan lebih lanjut.

```
> save(.data, file = "Chapter8.Rdata")
```

## Latihan

Hitung perbedaaan resiko dan rasio resiko dari 'beefcurry', 'saltegg' and 'water'. Apakah signifikan secara statistik? Jika iya, kenapa?

BAB 9

# Odds Rasio, Pembauran, dan Interaksi

Setelah melakukan berbagai penilaian parameter risiko dari peserta dalam wabah di bab terakhir, sekarang kami fokus pada pembauran antara berbagai jenis makanan.

Penilaian risiko dalam bab ini berubah dari aspek kemungkinan penyebab. Langkah berikutnya dalam menganalisis wabah adalah menguraikan tingkatan risiko. Pertama kita akan memuat data yang disimpan dari bab sebelumnya.

```
> zap()
> load("Chapter8.Rdata")
> use(.data)
```

## Odds dan Odds Rasio

Odds rasio memiliki makna yang berkaitan dengan probabilitas. Jika p adalah probabilitas, p / (1-p) dikenal sebagai odds. Sebaliknya, probabilitas akan sama dengan odds / (odds +1).

```
> tab1(case)
        Frequency Percent
FALSE         625    57.1
TRUE          469    42.9
  Total      1094   100.0
```

Probabilitas menjadi kasus adalah 469/1094 atau 42,9%. Dalam hal ini di mana non-kasus yang dikodekan sebagai 0 dan kasus dikodekan sebagai 1, probabilitasnya adalah

```
> mean(case)
```

Di sisi lain odds menjadi kasus adalah 469/625 = 0,7504, atau

```
> mean(case)/(1 - mean(case))
```

Perhatikan bahwa ketika ada nilai-nilai yang hilang dalam variabel, fungsi mean harus mengubah 'na.rm' argumen menjadi TRUE. Misalnya kemungkinan makan kue sus adalah:

```
> m.eclair <- mean(eclair.eat, na.rm = TRUE)
> m.eclair /(1 - m.eclair)
[1] 2.323129
```

Saat probabilitas selalu berkisar dari 0 sampai 1, sebuah odds berkisar dari 0 sampai tak terhingga. Untuk studi cohort kita dapat menghitung antara rasio odds exposed yang menjadi kasus vs odds non-exposed.

```
> table(case, eclair.eat)
       eclair.eat
case    FALSE TRUE
  FALSE   279  300
  TRUE     15  383
```

Metode konvensional untuk menghitung rasio odds :

```
> (383/300)/(15/279)
[1] 23.746
```

Ini adalah nilai yang sama sebagai rasio odds yang terbuka di antara kasus dan kalangan non-kasus.

```
> (383/15)/(300/279)
```

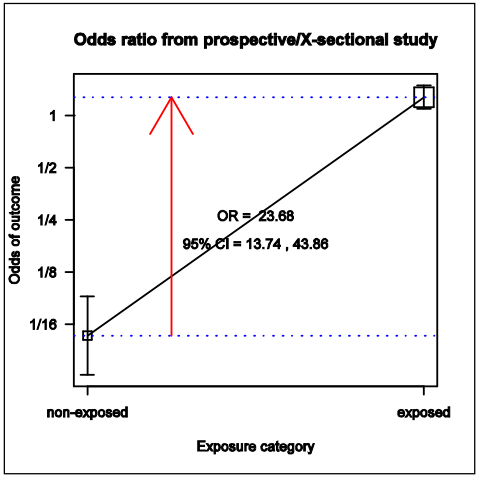
Hal ini juga sama dengan rasio antara cross-produk.

```
> (383 * 279)/(300 * 15)
```

Epicalc memiliki fungsi cc yang menghasilkan rasio odds, dengan interval kepercayaan 95%, melakukan uji chi-kuadrat dan uji eksak Fisher serta gambaran grafik sebagai penjelasan

```
> cc(case, eclair.eat)
       eating eclair
case     FALSE TRUE Total
  FALSE    279  300   579
  TRUE      15  383   398
  Total    294  683   977
OR =  23.68
95% CI = 13.74 43.86
Chi-squared = 221.21 ,  1 d.f. , P value = 0
Fisher's exact test (2-sided) P value = 0
```

Nilai rasio odds dari fungsi cc sedikit berbeda dari perhitungan yang telah kita lakukan. Hal ini dikarenakan fungsi cc menggunakan metode yang tepat untuk menghitung rasio odds.



Garis-garis vertikal dari grafik yang dihasilkan menunjukkan estimasi dan interval kepercayaan 95% dari dua kemungkinan yang sakit, non-exposed di sebelah kiri dan exposed di sebelah kanan, dihitung dengan metode konvensional. Ukuran kotak yang diperkirakan mencerminkan ukuran sampel relatif setiap subkelompok. Ada lebih banyak exposed daripada non-exposed. Kelompok non-exposed memiliki nilai estimasi sedikit di bawah 1/16 karena nilai sebenarnya adalah 15/279. Perkiraan nilai estimasi exposed adalah 383/300 atau sedikit lebih tinggi dari 1. Nilai yang terakhir ini lebih dari 23 kali dari nilai sebelumnya.

```
> fisher.test(table(case, eclair.eat))$estimate
odds ratio
  23.681

> fisher.test(table(case, eclair.eat))$conf.int
[1] 13.736 43.862
attr(,"conf.level")
[1] 0.95
```

## Pembauran dan mekanismenya

Untuk 'saltegg', rasio odds dapat juga dihitung.

```
> cc(case, saltegg)
       saltegg
case     0    1 Total
  FALSE 66  554   620
  TRUE  21  448   469
  Total 87 1002  1089
OR =  2.54
95% CI = 1.51 4.44
Chi-squared = 13.82 ,  1 d.f. , P value = 0
Fisher's exact test (2-sided) P value = 0
```

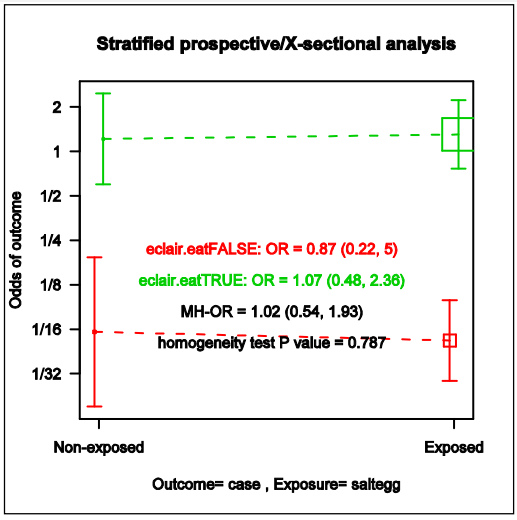
Total catatan yang valid untuk perhitungan adalah 1.089, dimana lebih tinggi 977 dari hasil cross-tabulasi antara 'case' dan 'eclair.eat'. Nilai odds ratio tidak setinggi nilai statistik tetapi signifikan. Sesuai dengan analisis rasio odds untuk 'Eclair', ukuran dari kotak di sebelah kanan jauh lebih besar daripada yang di sebelah kiri, hal ini menunjukkan sebagian besar dari eksposur.

Baik eclairs (kue sus) dan salted eggs (telur asin) memiliki odds rasio yang signifikan dan dikonsumsi oleh sebagian besar peserta. Mari kita memeriksa hubungan antara kedua variable ini.

```
> cc(saltegg, eclair.eat, graph = FALSE)
       eating eclair
saltegg FALSE TRUE Total
  0        53   31    84
  1       241  647   888
  Total   294  678   972
OR =  4.58
95% CI = 2.81 7.58
Chi-squared = 47.02 ,  1 d.f. , P value = 0
Fisher's exact test (2-sided) P value = 0
```

Hanya ada satu penyebab yang nyata dan yang lainnya hanya pembauran. Dengan kata lain, peserta yang mengkonsumsi saltegg (telur asin) juga cenderung untuk mengkonsumsi eclair (kue sus). Analisis bertingkat memberikan rincian pembauran sebagai berikut.

```
> mhor(case, saltegg, eclair.eat)
```




```
Stratified analysis by  eclair.eat
                     OR lower lim. upper lim. P value
```

```
eclair.eat FALSE  0.874      0.224       5.00   0.739
eclair.eat TRUE   1.073      0.481       2.36   0.855
M-H combined      1.023      0.541       1.93   0.944
M-H Chi2(1) = 0 , P value = 0.944
Homogeneity test, chi-squared 1 d.f.=0.07, P value = 0.787
```

Analisis atas hubungan antara penyakit dan saltegg (telur asin) yang dikelompokkan berdasarkan tingkat konsumsi eclair (kue sus) berdasarkan catatan yang memiliki nilai valid dari 'case', 'eclair.eat' dan 'saltegg'. Ada dua bagian utama dari hasil tersebut. Bagian pertama menyangkut rasio odds paparan kepentingan dalam setiap strata yang didefinisikan oleh variabel ketiga, dalam kasus ini 'eclair.eat' serta rasio odds dan chi-kuadrat statistik yang dihitung dengan teknik Mantel-Haenszel. Bagian kedua menunjukkan apakah rasio odds strata ini dapat dikombinasikan. Kami akan fokus pada bagian pertama pada tahap ini dan kembali ke bagian kedua nanti.

Dalam kedua strata, odds ratio yang dekat dengan 1 dan secara statistik tidak signifikan. Kemiringan dari dua garis yang agak datar. Rasio odds Mantel-Haenszel (MH), yang juga disebut adjusted rasio odds atau rasio odds yang disesuaikan, merupakan berat rata-rata dari dua rasio odds, yang juga dekat dengan 1. Baik rasio odds stratum-specific dan rasio odds MH tidak berbeda secara signifikan dari 1 tetapi rasio odds crude secara signifikan berbeda. Distorsi dari hasil crude yang berasal dari hasil yang disesuaikan (adjusted) disebut pembauran.

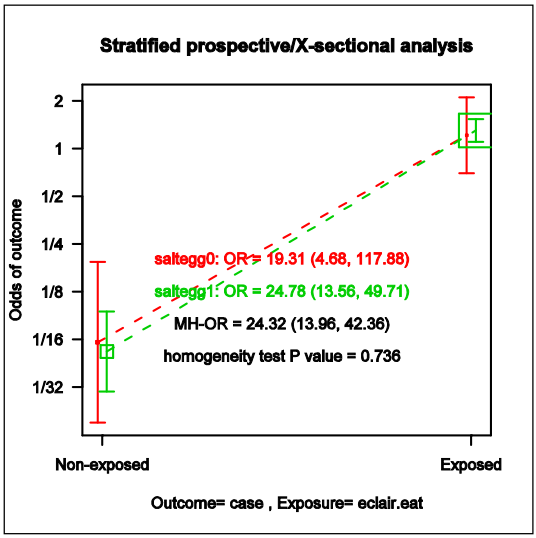
Mekanisme dari pembauran ini dapat dijelaskan dengan menggunakan grafik di atas. Garis atas dari grafik menunjukkan subset atau strata subyek yang mengkonsumsi eclair (kue sus) sedangkan garis bawah mewakili mereka yang tidak mengkonsumsinya. Garis atas terletak jauh di atas garis bawah hal ini berarti bahwa subset dari yang mengkonsumsi eclair (kue sus) memiliki risiko jauh lebih tinggi daripada yang tidak mengkonsumsi eclair (kue sus). Jarak antara dua garis tersebut adalah antara 16-32 kali lipat dari odds. Penting untuk dicatat bahwa distribusi subjek dalam penelitian ini tidak seimbang dalam kaitannya dengan konsumsi eclair dan saltegg. Di sisi kanan (konsumen saltegg), terdapat lebih banyak yang mengkonsumsi eclair (kotak atas) daripada yang tidak mengkonsumsinya (kotak bawah). Pusat dari sisi kanan kemudian cenderung lebih dekat ke lokasi dari kotak atas. Sebaliknya, di sisi kiri, atau mereka yang tidak mengkonsumsi saltegg, jumlah konsumen yang tidak mengkonsumsi eclair (yang diwakili oleh ukuran kotak lebih rendah) adalah

lebih tinggi dari konsumen yang mengkonsumsi eclair. Oleh karena itu pusat dari sisi kiri adalah salah untuk cenderung lebih dekat ke kotak yang lebih rendah. Dengan kata lain, ketika dua strata digabungkan, (berat rata-rata) kemungkinan antara konsumen saltegg (telur asin) adalah berpenyakit, oleh karena itu lebih dekat ke kotak atas. Sebaliknya untuk sisi kiri di mana berat rata-rata kemungkinan mendapatkan penyakit adalah benar harus lebih dekat ke kotak yang lebih rendah. Sebuah peluang rata-rata lebih tinggi di sisi kanan mengarah pada crude rasio odds yang lebih tinggi dari satu. Crude odds ratio ini menyesatkan kita pada pemikiran bahwa saltegg (telur asin) adalah penyebab lain dari penyakit dimana pada kenyataannya itu hanya dibaurkan oleh eclair. Tingkat pembauran dicatat hanya jika kedua dari dua kondisi berikut terpenuhi.

Pertama, faktor stratifikasi harus merupakan faktor risiko independen. Kedua, harus ada hubungan yang signifikan antara faktor stratifikasi dan eksposur dari ketertarikan.

Sekarang kita periksa apakah hubungan antara penyakit dan eclair ini dibaurkan oleh saltegg.

```
> mhor(case, eclair.eat, saltegg)
Stratified analysis by  saltegg
               OR lower lim. upper lim.  P value
saltegg 0    19.3       4.68      117.9 6.06e-07
saltegg 1    24.8      13.56       49.7 2.42e-51
M-H combined 24.3      13.96       42.4 8.12e-49
M-H Chi2(1) = 215.63 , P value = 0
Homogeneity test, chi-squared 1 d.f. = 0.11 , P value =
  0.736
```



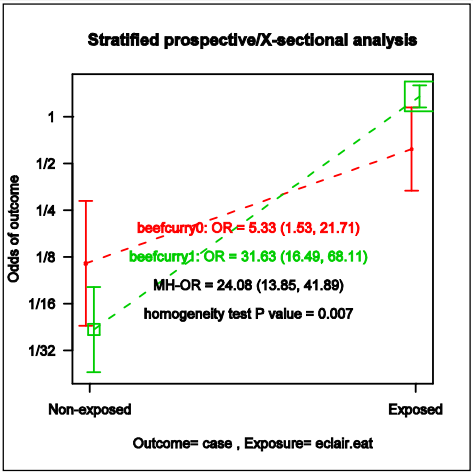

Dikelompokkan berdasarkan 'saltegg', odds rasio eclair.eat di kedua strata (19,3 dan 24,8) dan MH rasio odds (24,3) yang kuat dan dekat dengan crude rasio odds (23,68).

Secara grafis, dua garis strata yang sangat dekat menunjukkan bahwa 'saltegg' bukan merupakan faktor risiko independen. Dalam setiap kelompok exposed dan non-exposed, oleh karena itu kemungkinan untuk penyakit yang dekat dan kemungkinan berat rata-rata tidak dipengaruhi oleh jumlah subjek. Jadi variabel yang tidak dapat membaurkan variabel lain tidak dapat menjadi faktor risiko independen.

## Interaksi dan efek modifikasi

Mari kita menganalisis hubungan antara mengkonsumsi eclair dan berkembangnya penyakit gastrointestinal akut lagi, namun kali ini menggunakan 'beefcurry' sebagai faktor stratifikasi.

```
> mhor(case, eclair.eat, beefcurry)
Stratified analysis by  beefcurry
                OR lower lim. upper lim.  P value
beefcurry 0   5.33       1.53       21.7 3.12e-03
beefcurry 1  31.63      16.49       68.1 4.79e-56
M-H combined 24.08      13.85       41.9 1.39e-48
M-H Chi2(1) = 214.56 , P value = 0
Homogeneity test, chi-squared 1 d.f. = 7.23 , P value =
  0.007
```

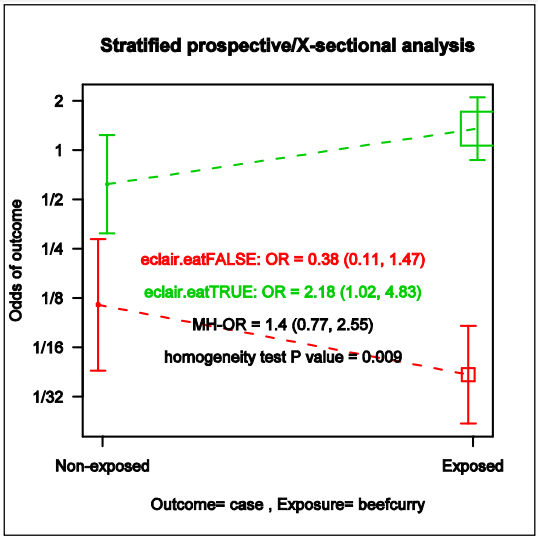



Kemiringan odds rasio dari dua strata saling silang. Di antara mereka yang tidak mengkonsumsi beef curry (kari daging sapi), kemungkinan mendapatkan penyakit di antara mereka yang tidak mengkonsumsi eclair sedikit di bawah 1 dari 6. Kemungkinan meningkat menjadi lebih dari 1 dalam 2 bagi mereka yang mengkonsumsi eclair saja. Peningkatan ini adalah 5.33 kali lipat atau 5.33 rasio odds. Sebaliknya, peluang garis bawah antara mereka mengkonsumsi beef curry saja (titik kiri dari garis hijau) adalah suatu tempat antara 1 di 32 dan 1 di 16, yang merupakan kelompok risiko terendah dalam grafik. Namun kemungkinan meningkat secara tajam ke lebih dari 1 di antara konsumen yang mengkonsumsi baik eclair dan beef curry. Uji homogenitas dalam baris terakhir menyimpulkan bahwa rasio odds yang tidak homogen. Dalam statistik, ini disebut interaksi yang signifikan. Dalam epidemiologi, efek dari 'Eclair' telah diubah oleh 'beefcurry'. Mengkonsumsi beef curry meningkatkan efek berbahaya dari eclair

atau meningkatkan kerentanan orang untuk mendapatkan sakit dengan mengkonsumsi eclair.

Kami sekarang memeriksa efek dari 'beefcurry' dikelompokkan oleh 'eclair.eat'.

```
> mhor(case, beefcurry, eclair.eat)
Stratified analysis by  eclair.eat
                    OR lower lim. upper lim. P value
eclair.eat FALSE 0.376      0.111       1.47  0.1446
eclair.eat TRUE  2.179      1.021       4.83  0.0329
M-H combined     1.401      0.769       2.55  0.2396
M-H Chi2(1) = 1.38 , P value = 0.24
Homogeneity test, chi-squared 1 d.f. = 6.78 , P value =
  0.009
```




Efek dari beer curry di antara mereka yang tidak mengkonsumsi eclair cenderung menjadi protektif tapi tanpa signifikansi statistik. Rasio odds antara konsumen yange mengkonsumsi eclair adalah 2,18 dengan signifikansi statistik. Tes homogenitas juga menyimpulkan bahwa dua odds rasio yang tidak homogen. Faktor stratifikasi eclair telah memodifikasi efek dari beef curry factor non-signifikan protektif menjadi faktor risiko yang signifikan.

Tabulasi dan grafik bertingkat sangat berguna dalam menjelaskan pembauran dan interaksi. Namun, mereka dibatasi hanya untuk dua atau tiga variabel. Untuk dataset dengan sejumlah besar variabel, dibutuhkan regresi logistik. Kami menempatkan variabel 'eclair.eat' baru ke dalam .data dengan menggunakan label.var dan menyimpan seluruh data frame untuk penggunaan nantinya dengan regresi logistik.

## Latihan

Analisa pengaruh air minum terhadap kemungkinan penyakit. Periksa apakah itu pembauran dengan mengkonsumsi kue sus atau makanan lain. Periksa interaksinya.

# BAB 10

# Manajemen Data Dasar

### *Pembersihan Data*

Dataset sebelumnya relatif bersih. Mari kita lihat sebuah dataset tidak bersih (uncleaned) yang berasal dari sebuah klinik keluarga berencana di pertengahan tahun 1980. Skema coding dapat dilihat dari

```
> help(Planning)
```

Pembersihan akan memungkinkan Anda untuk belajar fungsi Epicalc untuk pengelolaan data.

```
> zap()
> data(Planning)
> des(Planning)
```

Perhatikan bahwa semua nama-nama variabel dalam upper case. Untuk mengkonversikan menjadi kasus sederhana, cukup ketik perintah berikut.

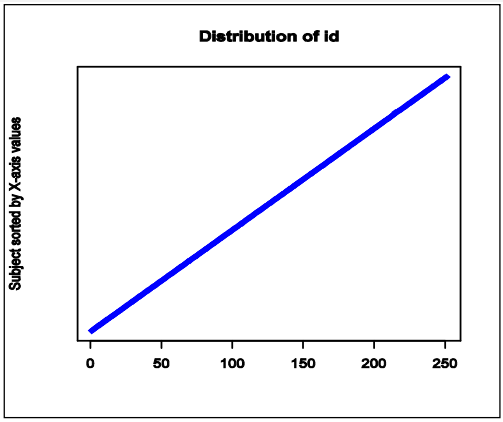
```
> names(Planning) <- tolower(names(Planning))
> use(Planning)
> summ()

No. of observations = 251
```

```
   Var. name Obs.  mean   median  s.d.   min.   max.
1  id        251   126    126     72.6   1      251
2  age       251   27.41  27      4.77   18     41
3  relig     251   1.14   1       0.59   1      9
4  ped       251   3.83   3       2.32   0      9
5  income    251   2.84   2       2.38   1      9
6  am        251   20.66  20      5.83   15     99
7  reason    251   1.55   1       0.86   1      9
8  bps       251   137.74 110     146.84 0      999
9  bpd       251   97.58  70      153.36 0      999
10 wt        251   52.85  51.9    11.09  0      99.9
11 ht        251   171.49 154     121.82 0      999
```

## Mengidentifikasi duplikasi ID

Mari kita lihat lebih dekat pada objek 'id'. Variabel ini merupakan nomor identifikasi unik untuk subjek.

```
> summ(id)
  Valid obs. mean     median  s.d.    min.    max.
  251        125.996  126     72.597  1       251
```




Grafik terlihat cukup merata (berdistribusi normal). Namun, rata-rata id (125,996) tidak sama dengan apa yang seharusnya.

```
> mean(1:251)
[1] 126
```

There must be some duplication and/or some gaps within these id numbers. Looking carefully at the graph, there is no noticeable irregularity.

To check for duplication, we can type the following:

Harus ada beberapa duplikasi dan / atau beberapa kesenjangan dalam angka-angka id ini . Lihat hati-hati pada grafik, tidak ada penyimpangan yang terlihat.

Untuk memeriksa duplikasi, kita bisa ketik berikut:

```
> any(duplicated(id))
[1] TRUE
```

Hasilnya memberitahu kita bahwa sebenarnya ada setidaknya satu id diduplikasi. Untuk menentukan id dari tipe duplikat:

```
> id[duplicated(id)]
[1] 215
```

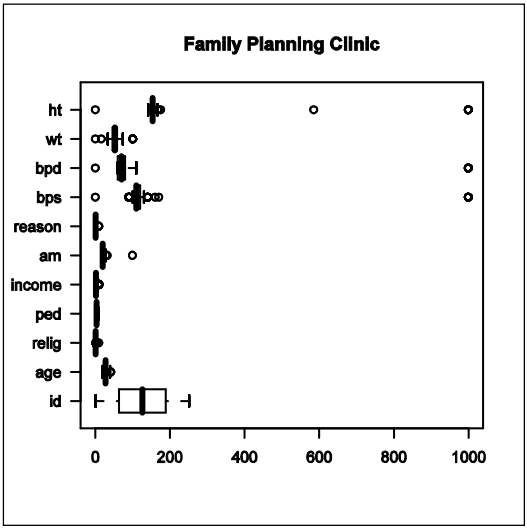
Kita melihat bahwa id = 215 memiliki satu duplikat. Pemeriksaan lebih lanjut dari data menunjukkan bahwa jumlah rekor adalah 215 dan 216. Ini dua catatan yang harus diselidiki dimana salah satunya tidak benar. Salah satu dari mereka harus diubah menjadi 'id' = 216.

## Data yang hilang

File ini belum siap untuk analisis. Seperti sering terjadi, data dikodekan menggunakan angka outlier untuk mewakili kode yang hilang.

Kami pertama mengeksplorasi data dengan boxplots.

```
> boxplot(.data, horizontal=T, las=1, main="Family Planning
  Clinic")
```




Nilai outlier dari 'bps', 'bph' dan 'ht' agak jelas. Ini dikonfirmasi dengan statistik numerik dari perintah summ yang terlihat sebelumnya dalam bab ini.

Dalam dataset ini, nilai '9 'merupakan kode hilang untuk agama (variabel 3),pendidikan pasien (4 variabel), kelompok pendapatan (5 variabel) dan alasan untuk keluarga berencana (7 variabel).

Ada empat metode untuk mengubah nilai-nilai yang hilang (NA). Metode pertama didasarkan pada menggantikan fungsi (fungsi pengganti), yang menangani satu vektor atau variabel pada suatu waktu. Yang kedua menggunakan ekstraksi dan mengindeks dengan subskrip'[]'. Metode ini dapat menangani baik vektor atau array (beberapa variabel pada waktu yang sama). Metode ketiga adalah berdasarkan perintah transform.

Ketiga metode menggunakan perintah yang asli ke R. Metode keempat menggunakan perintah recode (mengkode ulang) dari Epicalc, yang sejauh ini merupakan metode yang paling sederhana.

Kita akan menggunakan fungsi pengganti untuk variabel ke-3, 'relig', ekstraksi dan pengindeksan variabel untuk tanggal 4 sampai 7 , 'ped', 'am', 'income' dan 'reason', mengubah/mentransformasi untuk variabel 'wt', dan akhirnya recode (mengkode ulang) untuk variabel sisa yang diperlukan.

### *Mengganti nilai dalam data frame*

Kita ingin mengganti semua kejadian dari 9 dengan nilai yang hilang 'NA'. Fungsi pengganti menangani hanya satu variabel pada suatu waktu.

```
> summ(relig)
```

Kita ingin mengganti semua kejadian dari 9 dengan nilai yang hilang 'NA'.

```
> replace(relig, relig==9, NA) -> .data$relig
```

Ada tiga argumen penting untuk fungsi pengganti; vektor target, vektor indeks dan nilai. Lihat bantuan online untuk informasi lebih rinci tentang penggunaannya.
Argumen pertama, 'relig', adalah vektor yang berisi nilai target yang harus diganti. Argumen kedua, 'relig == 9', adalah vektor indeks menetapkan kondisi, dalam hal ini, setiap kali 'relig' adalah sama dengan 9. Argumen akhir, 'NA', adalah nilai baru yang akan menggantikan nilai lama 9. Jadi, setiap kali 'relig "adalah sama dengan 9, maka akan diganti dengan' NA '.

Perhatikan bahwa vektor indeks, atau kondisi untuk perubahan, tidak perlu vektor yang sama sebagai vektor target. Sebagai contoh, seseorang mungkin ingin memaksa nilai tekanan darah diastolik hilang jika tekanan darah sistoliknya hilang.
Kedua, replace adalah sebuah fungsi, bukan perintah. Ini tidak berpengaruh pada nilai-nilai asli. Nilai-nilai yang diperoleh dari fungsi ini harus ditugaskan dengan nilai-nilai asli menggunakan operator penugasan, '->' atau'<- '.
Sekarang, variabel telah berubah.

Ada tiga argumen penting untuk fungsi pengganti; vektor target, vektor indeks dan nilai. Lihat bantuan online untuk informasi lebih rinci tentang penggunaannya.

Argumen pertama, 'relig', adalah vektor yang berisi nilai target yang harus diganti. Argumen kedua, 'relig == 9', adalah vektor indeks menetapkan kondisi, dalam hal ini, setiap kali 'relig' adalah sama dengan 9. Argumen akhir, 'NA', adalah nilai baru yang akan menggantikan nilai lama 9. Jadi, setiap kali 'relig "adalah sama dengan 9, maka akan diganti dengan' NA '.

Perhatikan bahwa vektor indeks, atau kondisi untuk perubahan, tidak perlu

vektor yang sama sebagai vektor target. Sebagai contoh, seseorang mungkin ingin memaksa nilai tekanan darah diastolik hilang jika tekanan darah sistoliknya hilang.

Kedua, replace adalah sebuah fungsi, bukan perintah. Ini tidak berpengaruh pada nilai-nilai asli. Nilai-nilai yang diperoleh dari fungsi ini harus ditugaskan dengan nilai-nilai asli menggunakan operator penugasan, '->' atau'<-'.

Sekarang, variabel telah berubah.

```
> summ(.data$relig)
  Obs.  mean   median  s.d.   min.   max.
  250   1.108  1       0.31   1      2
```

Ada satu subjek dengan nilai yang hilang meninggalkan 250 catatan untuk perhitungan statistik. Subyek yang tersisa memiliki nilai satu dan dua hanya untuk'agama'.

## *Mengubah nilai-nilai dengan ekstraksi dan pengindeksan*

Variabel pertama yang diganti dengan metode ini adalah satu 6, 'am', yang menunjukkan usia saat perkawinan pertama.

```
> summ(.data$am)
  Valid obs. mean   median  s.d.   min.   max.
  251        20.657 20      5.83   15     99
```

Nilai 99 merupakan kode nilai yang hilang selama entri data. Perhatikan bahwa mean, median dan standar deviasi tidak benar karena ini pengkodean dari nilai-nilai yang hilang. Bahkan menggunakan metode sebelumnya, alternatif adalah:

```
> .data$am[.data$am==99] <- NA
```

Dengan tiga komponen yang sama dari target vektor, kondisi dan nilai penggantian, perintah terakhir ini sedikit lebih mudah daripada yang di atas yang menggunakan fungsi pengganti.

Metode ini juga dapat digunakan untuk banyak variabel dengan kode hilang sama. Sebagai contoh,, variabel keempat, kelima dan ketujuh semua menggunakan nilai 9 sebagai kode untuk nilai yang hilang.

```
> .data[,c(4,5,7)][.data[,c(4,5,7)]==9] <- NA
```

Semua variabel keempat, kelima dan ketujuh dari data yang memiliki nilai 9 digantikan dengan 'NA'. Perintah di atas dapat dijelaskan sebagai berikut. Ada dua lapisan subset dari data yang ditandai dengan '[]'.

`'.data[,c(4,5,7)]'` 'berarti ekstrak semua baris dari kolom 4, 5 dan 7, (' PED ','pendapatan 'dan' alasan ').

`'[.data[,c(4,5,7)]==9]` ' berarti subset dari setiap kolom tertentu di mana baris adalah sama dengan 9.

'<- NA' berarti epression di sebelah kiri adalah untuk diberi nilai yang hilang (NA).
Jadi, untuk keempat variabel, setiap elemen di mana nilai sama dengan 9 akan digantikan oleh NA.

## *Transformasi variabel dalam data frame*

Fungsi transformasi melakukan pekerjaan yang sama seperti metode yang dijelaskan sebelumnya di atas. Sebagai contoh, untuk mengubah 'wt'

```
> transform(.data, wt=ifelse(wt>99, NA, wt)) -> .data
```

Ekspresi dalam fungsi memberitahu R untuk menggantikan nilai-nilai 'wt' yang lebih besar dari 99 dengan nilai NA. Obyek yang dihasilkan disimpan ke dalam data frame.
Sekarang memeriksa 'wt' variabel di dalam frame data.

```
> summ(.data$wt)
  Valid obs. mean   median  s.d.   min.   max.
  246        51.895 51.45   8.91   0      73.8
```

Perhatikan dua outlier disisi kiri grafik. Mirip dengan hasil dari metode sebelumnya, tidak mengubah mengubah variabel 'wt' di dalam frame data dalam langkah pencarian.

```
> summ(wt)
  Valid obs. mean   median  s.d.   min.   max.
  251        52.851 51.9    11.09  0      99.9
```

Perhatikan bahwa frame data ditransformasikan tidak menyimpan label variabel atau deskripsi dengan itu. Data baru yang memiliki semua deskripsi variabel akan dihapus. Jadi metode ini mengurangi kekuatan Epicalc.

## Recoding (menkode Ulang) nilai dengan menggunakan Epicalc

Fungsi recode dalam Epicalc diciptakan untuk membuat transformasi data lebih mudah. Mirip dengan perintah lain di Epicalc, sebagai contoh *use, des, summ, tab1, dan label.var,* perintah recode yang dibatasi untuk pengaturan kepunyaan data sebagai data frame standar.

Kita memerlukan pengganti nilai-nilai '999 'untuk nilai yang hilang untuk variabel' bps ',' bpd 'dan' ht '. Perintahnya sederhana. Dan akan dimulai dengan 'bps'.

```
> recode(var=bps, old.value=999, new.value=NA)
> summ(.data)
```

Perhatikan bahwa variabel 'bps' telah berubah. Bahkan, recode telah otomatis terlepas dari data frame lama dan melekat ke yang baru, seperti yang ditunjukkan di bawah ini.

```
> summ(bps)
  Valid obs. mean    median  s.d.   min.   max.
  244        113.033 110     14.22  0      170
```

Variabel 'bps' di .data dan bahwa dalam jalur pencarian telah disinkronkan. Jumlah record yang valid dikurangi menjadi 244 dan maksimal 170 sekarang tidak 999. Perbaruan otomatis ini juga mempengaruhi variabel lain dalam langkah pencarian yang kita ubah sebelumnya.

```
> summ(am)
  Valid obs. mean   median  s.d.   min.   max.
  250        20.344 20      3.06   15     31
```

Ketika variabel 'am' digunakan sebagai argumen summ, program akan mencari objek independen yang disebut 'am', yang tidak ada. Kemudian terlihat dalam langkah pencarian. Karena data frame dalam langkah pencarian ('search () [2]') telah diperbarui dengan data baru., Variabel 'am' yang digunakan sekarang adalah salah satu update yang telah diubah dari perintah dalam bagian sebelumnya . Perintah recode membuat manipulasi variabel sederhana daripada tiga metode R standar di atas.

Perintah recode dapat lebih disederhanakan:

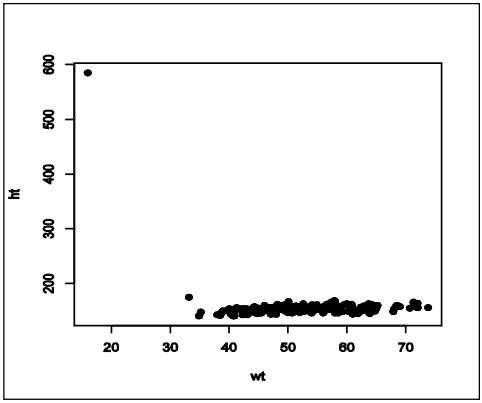
```
> recode(bpd, 999, NA)
> recode(ht, 999, NA)
> summ()
```

Semua maxima telah diperbaiki tetapi minima dari 0 juga hilang nilai untuk empat variabel terakhir ditambah 'ped'. Kita dapat menggunakan recode untuk mengubah semua nol ke nilai-nilai yang hilang dalam satu langkah.

```
> recode(c(ped, bps, bpd, wt, ht), 0, NA)
> summ()
No. of observations = 251
   Var. name Obs.  mean   median  s.d.   min.  max.
============ variables #1, #2, #3 omitted =========
4  ped       226   3.3    2       1.66   2     7
============ variables #5, #6, #7 omitted =========
8  bps       243   113.5  110     12.25  90    170
9  bpd       243   72.02  70      9.9    60    110
10 wt        245   52.11  51.5    8.28   16    73.8
11 ht        245   155.3  153     28.08  141   585
```

Berat minimum 16kg dan tinggi maksimum 585 cm adalah nilai meragukan dan sebenarnya tidak harus diterima. Setiap berat di bawah 30kg dan setiap tinggi di atas 200cm juga harus diperlakukan sebagai nilai hilang (kecuali ada alasan yang sangat baik untuk meninggalkan mereka sebagai bukan nilai hilang). Sebuah plot pencar (scatter plot) juga berguna di sini.

```
> plot(wt, ht, pch=19)
```

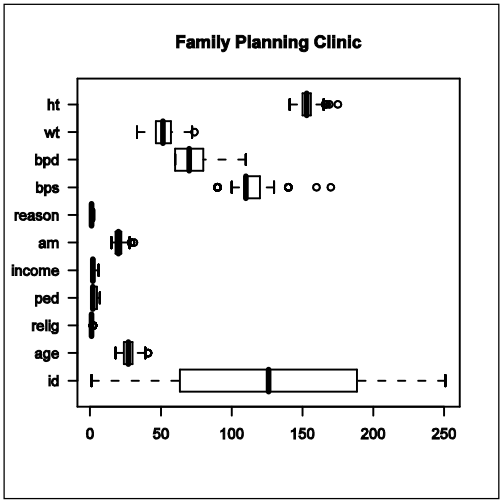


Outlier jelas terlihat (pada pojok kiri atas). Untuk memperbaiki kesalahan jenis ini:

```
> recode(wt, wt < 30, NA)
> recode(ht, ht > 200, NA)
> summ()
```

Perlu dicatat bahwa setelah pembersihan, ukuran sampel agak berkurang dari nilai aslinya yaitu 251. Boxplot semua variabel sekarang memiliki penampilan yang berbeda.

```
> boxplot(.data, horizontal=T, main="Family Planning
  Clinic", las=1)
```




## Pelabelan variabel dengan 'label.var'

Ketika hanya ada beberapa variabel di dataset, yang semuanya adalah untuk tujuan umum, seperti 'age', 'sex', atau 'education', penamaan tidak menjadi masalah. Namun, ketika ada sejumlah besar variabel, sulit untuk memiliki nama

intuitif yang dapat dimengerti untuk setiap variabel. Sebuah sistem label memisahkan variabel dari nama variabel adalah cara dokumentasi yang lebih baik.

R tidak datang dengan fasilitas membangun pelabelan variabel. Akan tetapi, Epicalc menambahkan fasilitas yang berguna dalam cara yang sederhana.

```
> names(.data)
[1] "id"     "age"    "relig"  "ped"  "income" "am"
[7] "reason" "bps"     "bpd"    "wt"   "ht"
```

Kemudian, sebuah label yang sesuai atau keterangan untuk masing-masing variabel dapat dibuat satu per satu.

```
> label.var(id, "Id code")
```

Pada tahap ini, pemeriksaan deskripsi dataset akan mengungkapkan deskripsi dari variabel pertama.

```
> des()
No. of observations =251
   Variable      Class           Description
1  id            numeric         Id code
2  age           numeric
3  relig         numeric
========= subsequent lines omitted ==========
```

Sebuah deskripsi variabel saja juga dapat ditampilkan.

```
> des(id)

'id' is a variable found in the following source(s):

 Var. source  Var. order Class   # records Description
 .data        1          numeric 251
```

Sekarang akan dilengkapi semua label variabel lainnya.

```
> label.var(age, "age")
> label.var(relig, "religion")
> label.var(ped, "eduction")
> label.var(income, "monthly income")
> label.var(am, "age(yr) 1st marriage")
> label.var(reason, "reason for fam. plan.")
> label.var(bps, "systolic BP")
> label.var(bpd, "diastolic BP")
```

```
> label.var(wt, "weight (kg)")
> label.var(ht, "height (cm)")
> des()


No. of observations =251
   Variable     Class          Description
1  id           numeric        ID code
2  age          numeric        age
3  relig        numeric        religion
4  ped          numeric        eduction
5  income       numeric        monthly income
6  am           numeric        age(yr) 1st marriage
7  reason       numeric        reason for fam. plan.
8  bps          numeric        systolic BP
9  bpd          numeric        diastolic BP
10 wt           numeric        weight (kg)
11 ht           numeric        height (cm)
```

Dalam hal ini disarankan untuk membuat setiap nama label dengan nama yang pendek karena akan sering digunakan dalam proses tampilan grafis otomatis dan tabulasi.

### *Pelabelan variabel kategorik*

Pelabelan nilai dari variabel kategorik adalah praktik yang baik. Ini adalah bagian dari dokumentasi penting. Selama analisis, variabel berlabel jauh lebih mudah untuk dipahami dan dijelaskan daripada variable tanpa label.

Seperti disebutkan sebelumnya, cara terbaik untuk label variabel selama persiapan dari pengentrian data menggunakan perangkat lunak entri data. Namun, terkadang seseorang dapat menemukan sebuah dataset tanpa label, seperti yang langsung diimpor/diambil dari format EpiInfo, 'txt' atau 'csv'. Oleh karena itu penting untuk mengetahui bagaimana untuk melabelkan/memberi keterangan variabel dalam R.

Dalam contoh kita tentang data keluarga berencana, variabel 'ped' (tingkat pendidikan pasien) adalah variabel kategorik tanpa label. Bahkan, pada tahap ini, bukan benar-benar sebuah variabel kategoris. Ketika kita meringkas statistik, baik dengan ringkasan perintah *(. data)* atau dengan *summ*, kedua output menunjukkan mean, median dan standar

deviasi, menunjukkan variabel numerik terus menerus.

```
> summary(ped)
   Min. 1st Qu. Median  Mean 3rd Qu.   Max.    NA's
  2.000   2.000  2.000  3.296  5.000  7.000  25.000

> summ(ped)
  Obs.  mean   median  s.d.   min.   max.
  226   3.296  2       1.66   2      7
```

Perhatikan bahwa tidak ada hitungan untuk kategori 1 dari 'ped'. Berdasarkan skema pengkodean:

1 = tidak ada pendidikan, 2 = sekolah dasar, 3 = sekolah menengah, 4 = sekolah tinggi,
5 = sekolah kejuruan, 6 = sarjana, 7 = lain.

Data adalah numerik dan karena itu perlu untuk dikonversi menjadi faktor. Label dapat dimasukkan ke dalam daftar 7 elemen.

```
> label.ped <- list(None="1", Primary="2", "Secondary
  school"="3", "High school"="4", Vocational="5", "Bachelor
  degree"="6", Others="7")
```

Setiap label harus ditutupi dalam tanda kutip ganda (”) jika mengandung spasi, selain itu ini bersifat opsional. Sebagai contoh, seseorang dapat memiliki: Tidak ada = "1" atau "Tidak" = "1".

Untuk mengkonversi vektor numerik untuk satu kategorik dapat menggunakan 'faktor'fungsi.

```
> educ <- factor(ped, exclude = NULL)
```

Variabel baru adalah hasil dari pemfaktoran nilai-nilai 'ped' di .data. Argumen ’exclude’ diatur ke 'NULL' menunjukkan tidak ada kategori (bahkan hilang atau 'NA') akan dikeluarkan dalam proses pemfaktoran.

```
> summary(educ)
   2    3    4    5    6    7 <NA>
 117   31   20   26   16   16   25
```

Kita dapat memeriksa label dari sebuah objek faktor menggunakan perintah tingkat.

```
> levels(educ)
```

```
[1] "2" "3" "4" "5" "6" "7" NA
```

Ada tujuh tingkat yang diketahui, mulai dari "2" ke "7" dan satu tingkat hilang (NA). Perhatikan bahwa angka-angka ini sebenarnya karakter atau nama grup. Tidak ada "1" dalam data dan secara koresponden dihilangkan dalam tingkat.

Tingkat untuk kode harus diubah menjadi kata-kata bermakna seperti yang didefinisikan sebelumnya.

```
> levels(educ) <- label.ped
> levels(educ)
[1] "None"             "Primary"          "Secondary school"
[4] "High school"      "Vocational"       "Bachelor degree"
[7] "Others"
```

## Penambahan variabel ke data frame

Perhatikan bahwa variabel 'educ' tidak di dalam data frame .data. Ingat bahwa R memiliki kapasitas untuk menangani lebih dari satu objek secara bersamaan. Namun, meskipun ada kemungkinan untuk menganalisis data dengan variabel diluar data frame .data, disarankan menggabungkan semua variabel penting ke dalam data frame utama .data, terutama jika pemilahan apapun dilakukan. Selain itu, variabel dapat memiliki label deskriptif. Lebih penting lagi, bila perlu, data frame keseluruhan termasuk variabel lama dan baru dapat ditulis ke dalam format data lain dengan mudah (lihat fungsi 'write.foreign' dalam foreign package / paket asing).

```
> des() # same as before
```

Untuk menggabungkan variabel baru yang berasal dari data frame .data, hanya label nama variabel sebagai berikut.

```
> label.var(educ, "education")
```

Kemudian memeriksa ulang.

```
> des()
No. of observations =251
   Variable      Class           Description
1  id            numeric         ID code
```

```
============ Variables # 2 to 11 omitted =======
12 educ          factor          education
```

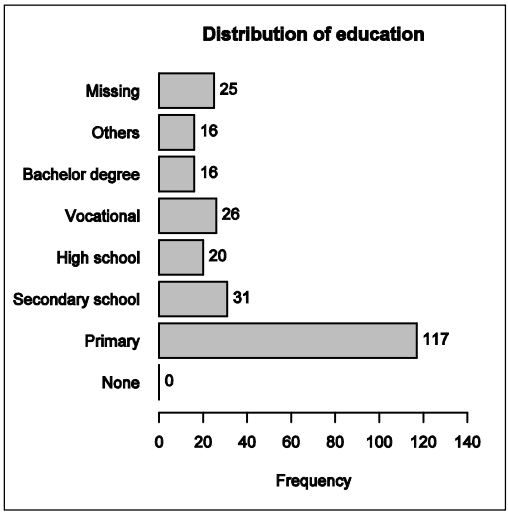
Untuk variable di luar .data, perintah *label.var* sebenarnya menyelesaikan lima tugas.

- Variabel baru dimasukkan ke dalam data frame.data,
- Variabel baru diberi label dengan keterangan,
- Data frame lama dipisah,
- Data lama di luar data frame yang '*free*' (bebas) di dihapus, kecuali argumen 'pack = FALSE' yang ditentukan,
- Data frame yang baru melekat ke langkah pencarian.

### *Perintah tabulasi atau arah*

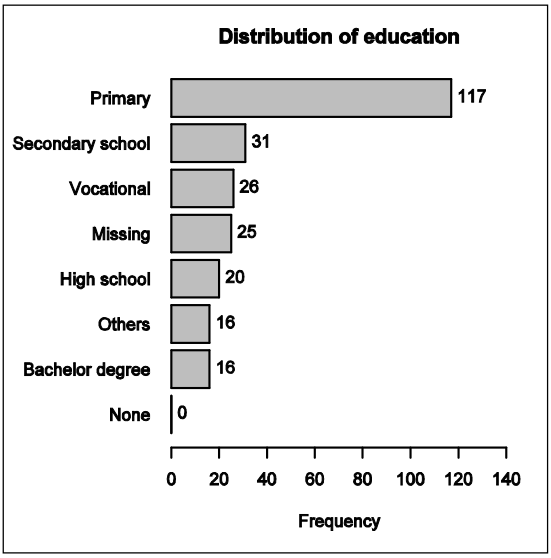
Variabel pendidikan baru dapat ditabulasikan.

```
> tab1(educ)
educ: education
                Frequency   %(NA+)   %(NA-)
None                    0      0.0      0.0
Primary               117     46.6     51.8
Secondary school       31     12.4     13.7
High school            20      8.0      8.8
Vocational             26     10.4     11.5
Bachelor degree        16      6.4      7.1
Others                 16      6.4      7.1
NA's                   25     10.0      0.0
   Total              251    100.0    100.0
```




Tabel dan grafik menunjukkan bahwa mata pelajaran yang paling hanya memiliki pendidikan dasar. Sebuah grafik bar horisontal diproduksi ketika jumlah kelompok melebihi 6 dan label kelompok terpanjang memiliki lebih dari 8 karakter. Tabulasi juga dapat diurutkan.

```
> tab1(educ, sort.group = "decreasing")
educ : education
                Frequency   %(NA+)   %(NA-)
Primary               117     46.6     51.8
Secondary school       31     12.4     13.7
Vocational             26     10.4     11.5
NA's                   25     10.0      0.0
High school            20      8.0      8.8
Bachelor degree        16      6.4      7.1
Others                 16      6.4      7.1
None                    0      0.0      0.0
  Total               251    100.0    100.0
```




Secara alternatif penyortiran dapat di tingkatkan.

```
> tabl(educ, sort.group = "increasing")
educ : education
                 Frequency   %(NA+)   %(NA-)
None                     0      0.0      0.0
Bachelor degree         16      6.4      7.1
Others                  16      6.4      7.1
High school             20      8.0      8.8
NA's                    25     10.0      0.0
Vocational              26     10.4     11.5
Secondary school        31     12.4     13.7
Primary                117     46.6     51.8
  Total                251    100.0    100.0
```

Sebuah meja pensortir/pengurutan dan grafik batang lebih mudah untuk dibaca dan dilihat ketika tidak ada urutan kategori. Namun, sebagian tingkat pendidikan diurutkan secara alami, sehingga grafik yang tidak diurutkan mungkin lebih baik.

## Mengurangi kategori

Kadang-kadang variabel kategorik mungkin memiliki terlalu banyak tingkatan. Analis mungkin ingin menggabungkan dua atau lebih kategori bersama menjadi satu.Sebagai contoh, tingkat kejuruan dan sarjana, yang merupakan tingkat ke-5 dan ke-6, dapat digabungkan menjadi satu tingkat yang disebut 'tersier'. Kita dapat melakukan ini dengan membuat sebuah variabel baru, yang kemudian dimasukkan ke dalam .data di akhir.

```
> ped2 <- educ
> levels(ped2)[5:6] <- "Tertiary"
> label.var(ped2, "level of education")
> des()
> tab1(ped2)

ped2 : level of education

                  Frequency   %(NA+)   %(NA-)
None                      0      0.0      0.0
Primary                 117     46.6     51.8
Secondary school         31     12.4     13.7
High school              20      8.0      8.8
Tertiary                 42     16.7     18.6
Others                   16      6.4      7.1
NA's                     25     10.0      0.0
  Total                 251    100.0    100.0
```

Dua kategori telah digabungkan menjadi satu memberikan 42 mata pelajaran yang memiliki tingkat pendidikan tersier.

## *Kesimpulan*

Dalam bab ini, kita telah melihat sebuah dataset dengan banyak data pembersihan yang dibutuhkan. Dalam praktek nyata, sangat penting untuk memiliki langkah-langkah preventif untuk meminimalkan kesalahan selama pengumpulan data dan entri data. Sebagai contoh, sebuah kendala dari range cek diperlukan dalam entri data. Nilai-nilai yang hilang lebih baik dimasukkan dengan kode hilang yang spesifik untuk perangkat lunak. Dalam EpiInfo, Stata dan SPSS ini adalah tanda periode '.' atau hanya dibiarkan kosong.

Salah satu cara terbaik untuk memasukkan data adalah dengan menggunakan perangkat lunak EpiData, yang dapat mengatur rentang hukum dan beberapa

pemeriksaan logis lainnya serta label variabel dan nilai-nilai dengan cara yang mudah. Jika ini telah dilakukan dengan benar, maka perintah yang sulit digunakan dalam bab ini tidak akan diperlukan. Dalam bab-bab yang tersisa, kita akan menggunakan dataset yang telah benar dimasukkan, dijaga untuk nilai-nilai yang hilang dan diberi label dengan benar.

Setiap kali suatu variabel diubah, ini adalah praktik yang baik untuk memperbarui variabel di dalam data frame yang terlampir dengan di luar.

Cara terbaik untuk memodifikasi data adalah dengan menggunakan recode, yang merupakan perintah Epicalc yang kuat . Hal ini dapat bekerja dengan satu variabel atau beberapa variabel dengan skema pengkodean ulang yang sama atau pengkodean ulang sebuah variabel atau variabel di bawah kondisi. Akhirnya, cara terbaik untuk memperbarui data frame dengan variabel baru atau yang diubah adalah dengan menggunakan label.var. Perintah ini tidak hanya label variabel untuk digunakan lebih lanjut tetapi juga update dan menggabungkan data frame dengan variabel luar. Lampiran data frame baru secara otomatis, membuat manipulasi data dalam R lebih halus dan sederhana.

Ada banyak fungsi-fungsi lainnya yang lebih maju dalam manajemen data R yang tidak tercakup dalam bab ini. Ini termasuk aggregate, reshape dan merge, dan pembaca didorong untuk mengeksplorasi perintah-perintah yang sangat berguna dan kuat ini pada mereka sendiri.

## Latihan

Dataset VCT berisi data dari kuesioner yang melibatkan pekerja seks perempuan dari Phuket, Thailand pada 2004.
Membaca file di R dan menggunakan perintah dalam bab ini untuk membersihkan data.

# BAB 10

# Manajemen Data Dasar

### *Pembersihan Data*

Dataset sebelumnya relatif bersih. Mari kita lihat sebuah dataset tidak bersih (uncleaned) yang berasal dari sebuah klinik keluarga berencana di pertengahan tahun 1980. Skema coding dapat dilihat dari

```
> help(Planning)
```

Pembersihan akan memungkinkan Anda untuk belajar fungsi Epicalc untuk pengelolaan data.

```
> zap()
> data(Planning)
> des(Planning)
```

Perhatikan bahwa semua nama-nama variabel dalam upper case. Untuk mengkonversikan menjadi kasus sederhana, cukup ketik perintah berikut.

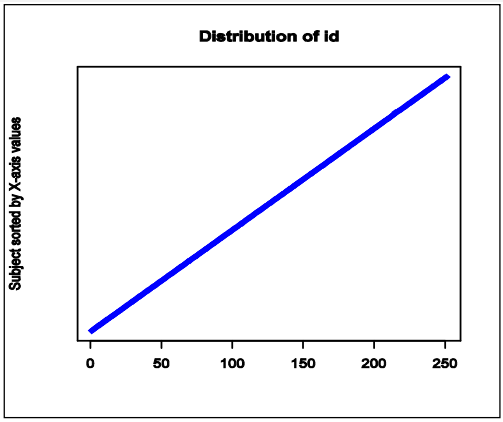
```
> names(Planning) <- tolower(names(Planning))
> use(Planning)
> summ()

No. of observations = 251
```

```
   Var. name Obs.  mean   median  s.d.   min.   max.
1  id        251   126    126     72.6   1      251
2  age       251   27.41  27      4.77   18     41
3  relig     251   1.14   1       0.59   1      9
4  ped       251   3.83   3       2.32   0      9
5  income    251   2.84   2       2.38   1      9
6  am        251   20.66  20      5.83   15     99
7  reason    251   1.55   1       0.86   1      9
8  bps       251   137.74 110     146.84 0      999
9  bpd       251   97.58  70      153.36 0      999
10 wt        251   52.85  51.9    11.09  0      99.9
11 ht        251   171.49 154     121.82 0      999
```

## Mengidentifikasi duplikasi ID

Mari kita lihat lebih dekat pada objek 'id'. Variabel ini merupakan nomor identifikasi unik untuk subjek.

```
> summ(id)
  Valid obs. mean     median  s.d.    min.    max.
  251        125.996  126     72.597  1       251
```




Grafik terlihat cukup merata (berdistribusi normal). Namun, rata-rata id (125,996) tidak sama dengan apa yang seharusnya.

```
> mean(1:251)
[1] 126
```

There must be some duplication and/or some gaps within these id numbers. Looking carefully at the graph, there is no noticeable irregularity.

To check for duplication, we can type the following:

Harus ada beberapa duplikasi dan / atau beberapa kesenjangan dalam angka-angka id ini . Lihat hati-hati pada grafik, tidak ada penyimpangan yang terlihat.

Untuk memeriksa duplikasi, kita bisa ketik berikut:

```
> any(duplicated(id))
[1] TRUE
```

Hasilnya memberitahu kita bahwa sebenarnya ada setidaknya satu id diduplikasi. Untuk menentukan id dari tipe duplikat:

```
> id[duplicated(id)]
[1] 215
```

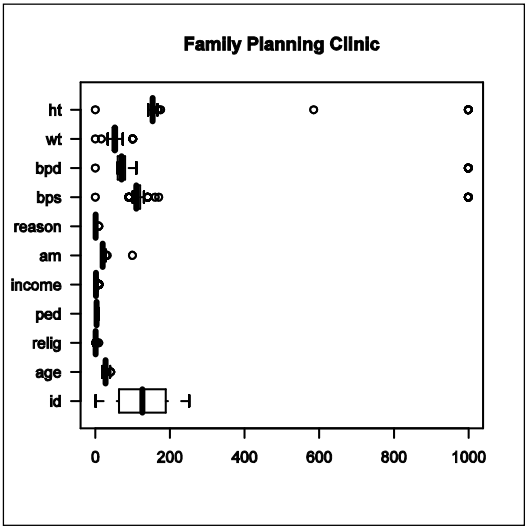
Kita melihat bahwa id = 215 memiliki satu duplikat. Pemeriksaan lebih lanjut dari data menunjukkan bahwa jumlah rekor adalah 215 dan 216. Ini dua catatan yang harus diselidiki dimana salah satunya tidak benar. Salah satu dari mereka harus diubah menjadi 'id' = 216.

## Data yang hilang

File ini belum siap untuk analisis. Seperti sering terjadi, data dikodekan menggunakan angka outlier untuk mewakili kode yang hilang.

Kami pertama mengeksplorasi data dengan boxplots.

```
> boxplot(.data, horizontal=T, las=1, main="Family Planning
  Clinic")
```




Nilai outlier dari 'bps', 'bph' dan 'ht' agak jelas. Ini dikonfirmasi dengan statistik numerik dari perintah summ yang terlihat sebelumnya dalam bab ini.

Dalam dataset ini, nilai '9 'merupakan kode hilang untuk agama (variabel 3),pendidikan pasien (4 variabel), kelompok pendapatan (5 variabel) dan alasan untuk keluarga berencana (7 variabel).

Ada empat metode untuk mengubah nilai-nilai yang hilang (NA). Metode pertama didasarkan pada menggantikan fungsi (fungsi pengganti), yang menangani satu vektor atau variabel pada suatu waktu. Yang kedua menggunakan ekstraksi dan mengindeks dengan subskrip'[]'. Metode ini dapat menangani baik vektor atau array (beberapa variabel pada waktu yang sama). Metode ketiga adalah berdasarkan perintah transform.

Ketiga metode menggunakan perintah yang asli ke R. Metode keempat menggunakan perintah recode (mengkode ulang) dari Epicalc, yang sejauh ini merupakan metode yang paling sederhana.

Kita akan menggunakan fungsi pengganti untuk variabel ke-3, 'relig', ekstraksi dan pengindeksan variabel untuk tanggal 4 sampai 7 , 'ped', 'am', 'income' dan 'reason', mengubah/mentransformasi untuk variabel 'wt', dan akhirnya recode (mengkode ulang) untuk variabel sisa yang diperlukan.

### *Mengganti nilai dalam data frame*

Kita ingin mengganti semua kejadian dari 9 dengan nilai yang hilang 'NA'. Fungsi pengganti menangani hanya satu variabel pada suatu waktu.

```
> summ(relig)
```

Kita ingin mengganti semua kejadian dari 9 dengan nilai yang hilang 'NA'.

```
> replace(relig, relig==9, NA) -> .data$relig
```

Ada tiga argumen penting untuk fungsi pengganti; vektor target, vektor indeks dan nilai. Lihat bantuan online untuk informasi lebih rinci tentang penggunaannya.
Argumen pertama, 'relig', adalah vektor yang berisi nilai target yang harus diganti. Argumen kedua, 'relig == 9', adalah vektor indeks menetapkan kondisi, dalam hal ini, setiap kali 'relig' adalah sama dengan 9. Argumen akhir, 'NA', adalah nilai baru yang akan menggantikan nilai lama 9. Jadi, setiap kali 'relig "adalah sama dengan 9, maka akan diganti dengan' NA '.

Perhatikan bahwa vektor indeks, atau kondisi untuk perubahan, tidak perlu vektor yang sama sebagai vektor target. Sebagai contoh, seseorang mungkin ingin memaksa nilai tekanan darah diastolik hilang jika tekanan darah sistoliknya hilang.
Kedua, replace adalah sebuah fungsi, bukan perintah. Ini tidak berpengaruh pada nilai-nilai asli. Nilai-nilai yang diperoleh dari fungsi ini harus ditugaskan dengan nilai-nilai asli menggunakan operator penugasan, '->' atau'<- '.
Sekarang, variabel telah berubah.

Ada tiga argumen penting untuk fungsi pengganti; vektor target, vektor indeks dan nilai. Lihat bantuan online untuk informasi lebih rinci tentang penggunaannya.

Argumen pertama, 'relig', adalah vektor yang berisi nilai target yang harus diganti. Argumen kedua, 'relig == 9', adalah vektor indeks menetapkan kondisi, dalam hal ini, setiap kali 'relig' adalah sama dengan 9. Argumen akhir, 'NA', adalah nilai baru yang akan menggantikan nilai lama 9. Jadi, setiap kali 'relig "adalah sama dengan 9, maka akan diganti dengan' NA '.

Perhatikan bahwa vektor indeks, atau kondisi untuk perubahan, tidak perlu

vektor yang sama sebagai vektor target. Sebagai contoh, seseorang mungkin ingin memaksa nilai tekanan darah diastolik hilang jika tekanan darah sistoliknya hilang.

Kedua, replace adalah sebuah fungsi, bukan perintah. Ini tidak berpengaruh pada nilai-nilai asli. Nilai-nilai yang diperoleh dari fungsi ini harus ditugaskan dengan nilai-nilai asli menggunakan operator penugasan, '->' atau'<-'.

Sekarang, variabel telah berubah.

```
> summ(.data$relig)
  Obs.  mean   median  s.d.   min.   max.
  250   1.108  1       0.31   1      2
```

Ada satu subjek dengan nilai yang hilang meninggalkan 250 catatan untuk perhitungan statistik. Subyek yang tersisa memiliki nilai satu dan dua hanya untuk'agama'.

## *Mengubah nilai-nilai dengan ekstraksi dan pengindeksan*

Variabel pertama yang diganti dengan metode ini adalah satu 6, 'am', yang menunjukkan usia saat perkawinan pertama.

```
> summ(.data$am)
  Valid obs. mean   median  s.d.   min.   max.
  251        20.657 20      5.83   15     99
```

Nilai 99 merupakan kode nilai yang hilang selama entri data. Perhatikan bahwa mean, median dan standar deviasi tidak benar karena ini pengkodean dari nilai-nilai yang hilang. Bahkan menggunakan metode sebelumnya, alternatif adalah:

```
> .data$am[.data$am==99] <- NA
```

Dengan tiga komponen yang sama dari target vektor, kondisi dan nilai penggantian, perintah terakhir ini sedikit lebih mudah daripada yang di atas yang menggunakan fungsi pengganti.

Metode ini juga dapat digunakan untuk banyak variabel dengan kode hilang sama. Sebagai contoh,, variabel keempat, kelima dan ketujuh semua menggunakan nilai 9 sebagai kode untuk nilai yang hilang.

```
> .data[,c(4,5,7)][.data[,c(4,5,7)]==9] <- NA
```

Semua variabel keempat, kelima dan ketujuh dari data yang memiliki nilai 9 digantikan dengan 'NA'. Perintah di atas dapat dijelaskan sebagai berikut. Ada dua lapisan subset dari data yang ditandai dengan '[]'.

`'.data[,c(4,5,7)]'` 'berarti ekstrak semua baris dari kolom 4, 5 dan 7, (' PED ','pendapatan 'dan' alasan ').

`'[.data[,c(4,5,7)]==9]` ' berarti subset dari setiap kolom tertentu di mana baris adalah sama dengan 9.

'<- NA' berarti epression di sebelah kiri adalah untuk diberi nilai yang hilang (NA).
Jadi, untuk keempat variabel, setiap elemen di mana nilai sama dengan 9 akan digantikan oleh NA.

## *Transformasi variabel dalam data frame*

Fungsi transformasi melakukan pekerjaan yang sama seperti metode yang dijelaskan sebelumnya di atas. Sebagai contoh, untuk mengubah 'wt'

```
> transform(.data, wt=ifelse(wt>99, NA, wt)) -> .data
```

Ekspresi dalam fungsi memberitahu R untuk menggantikan nilai-nilai 'wt' yang lebih besar dari 99 dengan nilai NA. Obyek yang dihasilkan disimpan ke dalam data frame.
Sekarang memeriksa 'wt' variabel di dalam frame data.

```
> summ(.data$wt)
  Valid obs. mean   median  s.d.   min.   max.
  246        51.895 51.45   8.91   0      73.8
```

Perhatikan dua outlier disisi kiri grafik. Mirip dengan hasil dari metode sebelumnya, tidak mengubah mengubah variabel 'wt' di dalam frame data dalam langkah pencarian.

```
> summ(wt)
  Valid obs. mean   median  s.d.   min.   max.
  251        52.851 51.9    11.09  0      99.9
```

Perhatikan bahwa frame data ditransformasikan tidak menyimpan label variabel atau deskripsi dengan itu. Data baru yang memiliki semua deskripsi variabel akan dihapus. Jadi metode ini mengurangi kekuatan Epicalc.

## Recoding (menkode Ulang) nilai dengan menggunakan Epicalc

Fungsi recode dalam Epicalc diciptakan untuk membuat transformasi data lebih mudah. Mirip dengan perintah lain di Epicalc, sebagai contoh *use, des, summ, tab1, dan label.var,* perintah recode yang dibatasi untuk pengaturan kepunyaan data sebagai data frame standar.

Kita memerlukan pengganti nilai-nilai '999 'untuk nilai yang hilang untuk variabel' bps ',' bpd 'dan' ht '. Perintahnya sederhana. Dan akan dimulai dengan 'bps'.

```
> recode(var=bps, old.value=999, new.value=NA)
> summ(.data)
```

Perhatikan bahwa variabel 'bps' telah berubah. Bahkan, recode telah otomatis terlepas dari data frame lama dan melekat ke yang baru, seperti yang ditunjukkan di bawah ini.

```
> summ(bps)
  Valid obs. mean    median  s.d.   min.   max.
  244        113.033 110     14.22  0      170
```

Variabel 'bps' di .data dan bahwa dalam jalur pencarian telah disinkronkan. Jumlah record yang valid dikurangi menjadi 244 dan maksimal 170 sekarang tidak 999. Perbaruan otomatis ini juga mempengaruhi variabel lain dalam langkah pencarian yang kita ubah sebelumnya.

```
> summ(am)
  Valid obs. mean   median  s.d.   min.   max.
  250        20.344 20      3.06   15     31
```

Ketika variabel 'am' digunakan sebagai argumen summ, program akan mencari objek independen yang disebut 'am', yang tidak ada. Kemudian terlihat dalam langkah pencarian. Karena data frame dalam langkah pencarian ('search () [2]') telah diperbarui dengan data baru., Variabel 'am' yang digunakan sekarang adalah salah satu update yang telah diubah dari perintah dalam bagian sebelumnya . Perintah recode membuat manipulasi variabel sederhana daripada tiga metode R standar di atas.

Perintah recode dapat lebih disederhanakan:

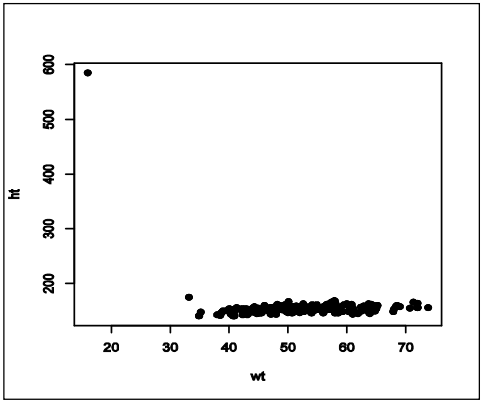
```
> recode(bpd, 999, NA)
> recode(ht, 999, NA)
> summ()
```

Semua maxima telah diperbaiki tetapi minima dari 0 juga hilang nilai untuk empat variabel terakhir ditambah 'ped'. Kita dapat menggunakan recode untuk mengubah semua nol ke nilai-nilai yang hilang dalam satu langkah.

```
> recode(c(ped, bps, bpd, wt, ht), 0, NA)
> summ()
No. of observations = 251
   Var. name Obs.  mean   median  s.d.   min.  max.
============ variables #1, #2, #3 omitted =========
4  ped       226   3.3    2       1.66   2     7
============ variables #5, #6, #7 omitted =========
8  bps       243   113.5  110     12.25  90    170
9  bpd       243   72.02  70      9.9    60    110
10 wt        245   52.11  51.5    8.28   16    73.8
11 ht        245   155.3  153     28.08  141   585
```

Berat minimum 16kg dan tinggi maksimum 585 cm adalah nilai meragukan dan sebenarnya tidak harus diterima. Setiap berat di bawah 30kg dan setiap tinggi di atas 200cm juga harus diperlakukan sebagai nilai hilang (kecuali ada alasan yang sangat baik untuk meninggalkan mereka sebagai bukan nilai hilang). Sebuah plot pencar (scatter plot) juga berguna di sini.

```
> plot(wt, ht, pch=19)
```

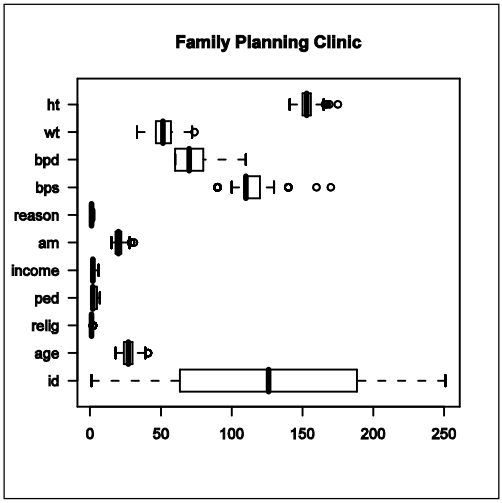


Outlier jelas terlihat (pada pojok kiri atas). Untuk memperbaiki kesalahan jenis ini:

```
> recode(wt, wt < 30, NA)
> recode(ht, ht > 200, NA)
> summ()
```

Perlu dicatat bahwa setelah pembersihan, ukuran sampel agak berkurang dari nilai aslinya yaitu 251. Boxplot semua variabel sekarang memiliki penampilan yang berbeda.

```
> boxplot(.data, horizontal=T, main="Family Planning
  Clinic", las=1)
```




## Pelabelan variabel dengan 'label.var'

Ketika hanya ada beberapa variabel di dataset, yang semuanya adalah untuk tujuan umum, seperti 'age', 'sex', atau 'education', penamaan tidak menjadi masalah. Namun, ketika ada sejumlah besar variabel, sulit untuk memiliki nama

intuitif yang dapat dimengerti untuk setiap variabel. Sebuah sistem label memisahkan variabel dari nama variabel adalah cara dokumentasi yang lebih baik.

R tidak datang dengan fasilitas membangun pelabelan variabel. Akan tetapi, Epicalc menambahkan fasilitas yang berguna dalam cara yang sederhana.

```
> names(.data)
[1] "id"     "age"    "relig"  "ped"  "income" "am"
[7] "reason" "bps"     "bpd"    "wt"   "ht"
```

Kemudian, sebuah label yang sesuai atau keterangan untuk masing-masing variabel dapat dibuat satu per satu.

```
> label.var(id, "Id code")
```

Pada tahap ini, pemeriksaan deskripsi dataset akan mengungkapkan deskripsi dari variabel pertama.

```
> des()
No. of observations =251
   Variable      Class           Description
1  id            numeric         Id code
2  age           numeric
3  relig         numeric
========= subsequent lines omitted ==========
```

Sebuah deskripsi variabel saja juga dapat ditampilkan.

```
> des(id)

'id' is a variable found in the following source(s):

 Var. source  Var. order Class   # records Description
 .data        1          numeric 251
```

Sekarang akan dilengkapi semua label variabel lainnya.

```
> label.var(age, "age")
> label.var(relig, "religion")
> label.var(ped, "eduction")
> label.var(income, "monthly income")
> label.var(am, "age(yr) 1st marriage")
> label.var(reason, "reason for fam. plan.")
> label.var(bps, "systolic BP")
> label.var(bpd, "diastolic BP")
```

```
> label.var(wt, "weight (kg)")
> label.var(ht, "height (cm)")
> des()


No. of observations =251
   Variable     Class          Description
1  id           numeric        ID code
2  age          numeric        age
3  relig        numeric        religion
4  ped          numeric        eduction
5  income       numeric        monthly income
6  am           numeric        age(yr) 1st marriage
7  reason       numeric        reason for fam. plan.
8  bps          numeric        systolic BP
9  bpd          numeric        diastolic BP
10 wt           numeric        weight (kg)
11 ht           numeric        height (cm)
```

Dalam hal ini disarankan untuk membuat setiap nama label dengan nama yang pendek karena akan sering digunakan dalam proses tampilan grafis otomatis dan tabulasi.

### *Pelabelan variabel kategorik*

Pelabelan nilai dari variabel kategorik adalah praktik yang baik. Ini adalah bagian dari dokumentasi penting. Selama analisis, variabel berlabel jauh lebih mudah untuk dipahami dan dijelaskan daripada variable tanpa label.

Seperti disebutkan sebelumnya, cara terbaik untuk label variabel selama persiapan dari pengentrian data menggunakan perangkat lunak entri data. Namun, terkadang seseorang dapat menemukan sebuah dataset tanpa label, seperti yang langsung diimpor/diambil dari format EpiInfo, 'txt' atau 'csv'. Oleh karena itu penting untuk mengetahui bagaimana untuk melabelkan/memberi keterangan variabel dalam R.

Dalam contoh kita tentang data keluarga berencana, variabel 'ped' (tingkat pendidikan pasien) adalah variabel kategorik tanpa label. Bahkan, pada tahap ini, bukan benar-benar sebuah variabel kategoris. Ketika kita meringkas statistik, baik dengan ringkasan perintah *(. data)* atau dengan *summ*, kedua output menunjukkan mean, median dan standar

deviasi, menunjukkan variabel numerik terus menerus.

```
> summary(ped)
   Min. 1st Qu. Median  Mean 3rd Qu.   Max.    NA's
  2.000   2.000  2.000  3.296  5.000  7.000  25.000

> summ(ped)
  Obs.  mean   median  s.d.   min.   max.
  226   3.296  2       1.66   2      7
```

Perhatikan bahwa tidak ada hitungan untuk kategori 1 dari 'ped'. Berdasarkan skema pengkodean:

1 = tidak ada pendidikan, 2 = sekolah dasar, 3 = sekolah menengah, 4 = sekolah tinggi,
5 = sekolah kejuruan, 6 = sarjana, 7 = lain.

Data adalah numerik dan karena itu perlu untuk dikonversi menjadi faktor. Label dapat dimasukkan ke dalam daftar 7 elemen.

```
> label.ped <- list(None="1", Primary="2", "Secondary
  school"="3", "High school"="4", Vocational="5", "Bachelor
  degree"="6", Others="7")
```

Setiap label harus ditutupi dalam tanda kutip ganda (”) jika mengandung spasi, selain itu ini bersifat opsional. Sebagai contoh, seseorang dapat memiliki: Tidak ada = "1" atau "Tidak" = "1".

Untuk mengkonversi vektor numerik untuk satu kategorik dapat menggunakan 'faktor'fungsi.

```
> educ <- factor(ped, exclude = NULL)
```

Variabel baru adalah hasil dari pemfaktoran nilai-nilai 'ped' di .data. Argumen ’exclude’ diatur ke 'NULL' menunjukkan tidak ada kategori (bahkan hilang atau 'NA') akan dikeluarkan dalam proses pemfaktoran.

```
> summary(educ)
   2    3    4    5    6    7 <NA>
 117   31   20   26   16   16   25
```

Kita dapat memeriksa label dari sebuah objek faktor menggunakan perintah tingkat.

```
> levels(educ)
```

```
[1] "2" "3" "4" "5" "6" "7" NA
```

Ada tujuh tingkat yang diketahui, mulai dari "2" ke "7" dan satu tingkat hilang (NA). Perhatikan bahwa angka-angka ini sebenarnya karakter atau nama grup. Tidak ada "1" dalam data dan secara koresponden dihilangkan dalam tingkat.

Tingkat untuk kode harus diubah menjadi kata-kata bermakna seperti yang didefinisikan sebelumnya.

```
> levels(educ) <- label.ped
> levels(educ)
[1] "None"             "Primary"          "Secondary school"
[4] "High school"      "Vocational"       "Bachelor degree"
[7] "Others"
```

## Penambahan variabel ke data frame

Perhatikan bahwa variabel 'educ' tidak di dalam data frame .data. Ingat bahwa R memiliki kapasitas untuk menangani lebih dari satu objek secara bersamaan. Namun, meskipun ada kemungkinan untuk menganalisis data dengan variabel diluar data frame .data, disarankan menggabungkan semua variabel penting ke dalam data frame utama .data, terutama jika pemilahan apapun dilakukan. Selain itu, variabel dapat memiliki label deskriptif. Lebih penting lagi, bila perlu, data frame keseluruhan termasuk variabel lama dan baru dapat ditulis ke dalam format data lain dengan mudah (lihat fungsi 'write.foreign' dalam foreign package / paket asing).

```
> des() # same as before
```

Untuk menggabungkan variabel baru yang berasal dari data frame .data, hanya label nama variabel sebagai berikut.

```
> label.var(educ, "education")
```

Kemudian memeriksa ulang.

```
> des()
No. of observations =251
   Variable      Class           Description
1  id            numeric         ID code
```

```
============ Variables # 2 to 11 omitted =======
12 educ          factor          education
```

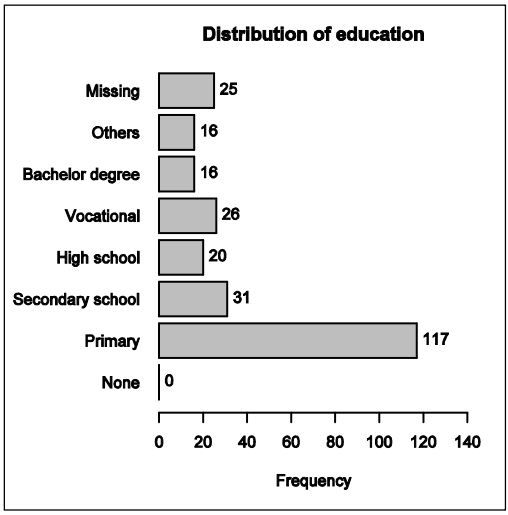
Untuk variable di luar .data, perintah *label.var* sebenarnya menyelesaikan lima tugas.

- Variabel baru dimasukkan ke dalam data frame.data,
- Variabel baru diberi label dengan keterangan,
- Data frame lama dipisah,
- Data lama di luar data frame yang '*free*' (bebas) di dihapus, kecuali argumen 'pack = FALSE' yang ditentukan,
- Data frame yang baru melekat ke langkah pencarian.

### *Perintah tabulasi atau arah*

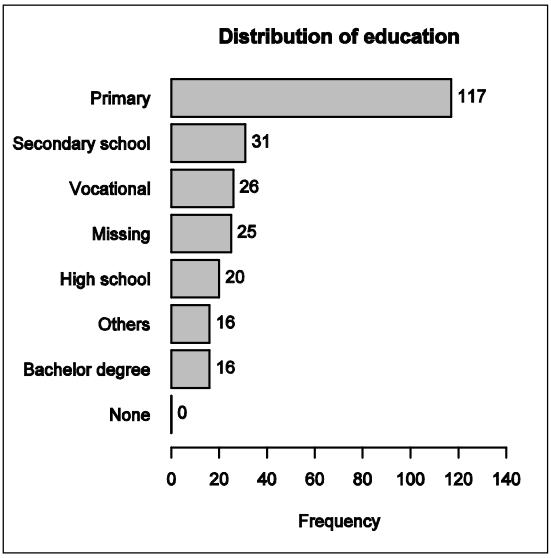
Variabel pendidikan baru dapat ditabulasikan.

```
> tab1(educ)
educ: education
                Frequency   %(NA+)   %(NA-)
None                    0      0.0      0.0
Primary               117     46.6     51.8
Secondary school       31     12.4     13.7
High school            20      8.0      8.8
Vocational             26     10.4     11.5
Bachelor degree        16      6.4      7.1
Others                 16      6.4      7.1
NA's                   25     10.0      0.0
   Total              251    100.0    100.0
```




Tabel dan grafik menunjukkan bahwa mata pelajaran yang paling hanya memiliki pendidikan dasar. Sebuah grafik bar horisontal diproduksi ketika jumlah kelompok melebihi 6 dan label kelompok terpanjang memiliki lebih dari 8 karakter. Tabulasi juga dapat diurutkan.

```
> tab1(educ, sort.group = "decreasing")
educ : education
                Frequency   %(NA+)   %(NA-)
Primary               117     46.6     51.8
Secondary school       31     12.4     13.7
Vocational             26     10.4     11.5
NA's                   25     10.0      0.0
High school            20      8.0      8.8
Bachelor degree        16      6.4      7.1
Others                 16      6.4      7.1
None                    0      0.0      0.0
  Total               251    100.0    100.0
```




Secara alternatif penyortiran dapat di tingkatkan.

```
> tabl(educ, sort.group = "increasing")
educ : education
                 Frequency   %(NA+)   %(NA-)
None                     0      0.0      0.0
Bachelor degree         16      6.4      7.1
Others                  16      6.4      7.1
High school             20      8.0      8.8
NA's                    25     10.0      0.0
Vocational              26     10.4     11.5
Secondary school        31     12.4     13.7
Primary                117     46.6     51.8
  Total                251    100.0    100.0
```

Sebuah meja pensortir/pengurutan dan grafik batang lebih mudah untuk dibaca dan dilihat ketika tidak ada urutan kategori. Namun, sebagian tingkat pendidikan diurutkan secara alami, sehingga grafik yang tidak diurutkan mungkin lebih baik.

## Mengurangi kategori

Kadang-kadang variabel kategorik mungkin memiliki terlalu banyak tingkatan. Analis mungkin ingin menggabungkan dua atau lebih kategori bersama menjadi satu.Sebagai contoh, tingkat kejuruan dan sarjana, yang merupakan tingkat ke-5 dan ke-6, dapat digabungkan menjadi satu tingkat yang disebut 'tersier'. Kita dapat melakukan ini dengan membuat sebuah variabel baru, yang kemudian dimasukkan ke dalam .data di akhir.

```
> ped2 <- educ
> levels(ped2)[5:6] <- "Tertiary"
> label.var(ped2, "level of education")
> des()
> tab1(ped2)

ped2 : level of education

                  Frequency   %(NA+)   %(NA-)
None                      0      0.0      0.0
Primary                 117     46.6     51.8
Secondary school         31     12.4     13.7
High school              20      8.0      8.8
Tertiary                 42     16.7     18.6
Others                   16      6.4      7.1
NA's                     25     10.0      0.0
  Total                 251    100.0    100.0
```

Dua kategori telah digabungkan menjadi satu memberikan 42 mata pelajaran yang memiliki tingkat pendidikan tersier.

## *Kesimpulan*

Dalam bab ini, kita telah melihat sebuah dataset dengan banyak data pembersihan yang dibutuhkan. Dalam praktek nyata, sangat penting untuk memiliki langkah-langkah preventif untuk meminimalkan kesalahan selama pengumpulan data dan entri data. Sebagai contoh, sebuah kendala dari range cek diperlukan dalam entri data. Nilai-nilai yang hilang lebih baik dimasukkan dengan kode hilang yang spesifik untuk perangkat lunak. Dalam EpiInfo, Stata dan SPSS ini adalah tanda periode '.' atau hanya dibiarkan kosong.

Salah satu cara terbaik untuk memasukkan data adalah dengan menggunakan perangkat lunak EpiData, yang dapat mengatur rentang hukum dan beberapa

pemeriksaan logis lainnya serta label variabel dan nilai-nilai dengan cara yang mudah. Jika ini telah dilakukan dengan benar, maka perintah yang sulit digunakan dalam bab ini tidak akan diperlukan. Dalam bab-bab yang tersisa, kita akan menggunakan dataset yang telah benar dimasukkan, dijaga untuk nilai-nilai yang hilang dan diberi label dengan benar.

Setiap kali suatu variabel diubah, ini adalah praktik yang baik untuk memperbarui variabel di dalam data frame yang terlampir dengan di luar.

Cara terbaik untuk memodifikasi data adalah dengan menggunakan recode, yang merupakan perintah Epicalc yang kuat . Hal ini dapat bekerja dengan satu variabel atau beberapa variabel dengan skema pengkodean ulang yang sama atau pengkodean ulang sebuah variabel atau variabel di bawah kondisi. Akhirnya, cara terbaik untuk memperbarui data frame dengan variabel baru atau yang diubah adalah dengan menggunakan label.var. Perintah ini tidak hanya label variabel untuk digunakan lebih lanjut tetapi juga update dan menggabungkan data frame dengan variabel luar. Lampiran data frame baru secara otomatis, membuat manipulasi data dalam R lebih halus dan sederhana.

Ada banyak fungsi-fungsi lainnya yang lebih maju dalam manajemen data R yang tidak tercakup dalam bab ini. Ini termasuk aggregate, reshape dan merge, dan pembaca didorong untuk mengeksplorasi perintah-perintah yang sangat berguna dan kuat ini pada mereka sendiri.

## Latihan

Dataset VCT berisi data dari kuesioner yang melibatkan pekerja seks perempuan dari Phuket, Thailand pada 2004.
Membaca file di R dan menggunakan perintah dalam bab ini untuk membersihkan data.

BAB 12

# Regresi Linier Bertingkat

Dataset yang dikumpulkan selama penelitian biasanya mengandung banyak variabel. Hal ini sering berguna untuk melihat hubungan antara dua variabel dalam tingkat yang berbeda dari sepertiga lainnya, variabel kategorik.

### *Contoh : Tekanan Darah Sistolik*

Sebuah survei kecil telah dilakukan pada tekanan darah . Tujuannya adalah untuk melihat efek hipertensi terhadap subjek dengan melakukan penambahan garam meja pada makanan mereka.

```
> zap()
> data(BP);  use(BP)
> des()

cross-sectional survey on BP & risk factors
No. of observations =100
  Variable      Class           Description
1 id            integer         id
2 sex           factor          sex
3 sbp           integer         Systolic BP
4 dbp           integer         Diastolic BP
5 saltadd       factor          Salt added on table
6 birthdate     Date
```

```
> summ()
cross-sectional survey on BP & risk factors
No. of observations = 100
Var. name  Obs.  mean       median     s.d.   min.
  max.
id         100   50.5       50.5       29.01  1          100
sex        100   1.55       2          0.5    1          2
sbp        100   154.34     148        39.3   80         238
dbp        100   98.51      96         22.74  55         158
saltadd     80   1.538      2          0.502  1          2
birthdate  100   1952-10-11 1951-11-17 <NA>   1930-11-14
  1975-12-08
```

Perhatikan bahwa maksimum dari tekanan darah sistolik dan diastoliknya cukup tinggi. Ada 20 nilai yang hilang pada *'saltadd'* (penambahan garam) dan sekarang frekuensi dari variabel kategori *'sex'* (jenis kelamin) dan *'saltadd'* (penambahan garam) diperiksa.

```
> summary(data.frame(sex, saltadd))
     sex     saltadd
 male  :45   no  :37
 female:55   yes :43
             NA's:20
```

Langkah selanjutnya adalah membuat variabel umur baru dari tanggal lahir. Perhitungan ini didasarkan pada tanggal survey, 12 Maret 2001.

```
> age.in.days <- as.Date("2001-03-12") - birthdate
```

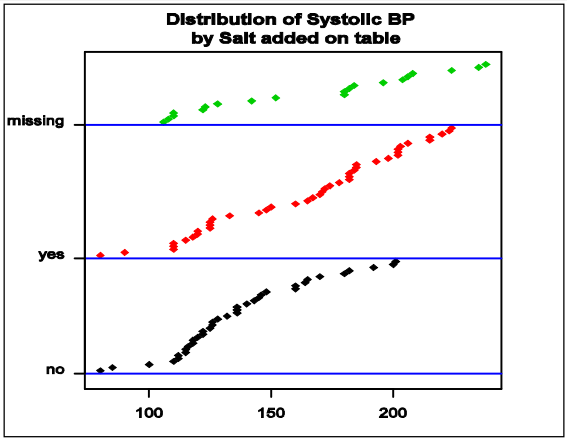
Ada tahun kabisat setiap empat tahun. Oleh karena itu, rata-rata tahun akan menjadi 365,25 hari.

```
> class(age.in.days)
[1] "difftime"
> age <- as.numeric(age.in.days)/365.25
```

Fungsi as.numeric diperlukan untuk mengubah unit umur (difftime), jika tidak maka pemodelan tidak akan mungkin dilakukan.

```
> summ(sbp, by = saltadd)

For saltadd = no
  Obs.  mean   median  s.d.   min.   max.
  37    137.5  132     29.624 80     201
For saltadd = yes
```

```
  Obs.  mean   median  s.d.   min.   max.
  43    163    171     39.39  80     224
For saltadd = missing
  Obs.  mean   median  s.d.   min.   max.
  20    166.9  180     45.428 106    238
```




Pengkodean ulang nilai-nilai yang hilang ke dalam kategori lain

Kelompok nilai yang hilang memiliki median dan rata-rata tekanan darah sistolik tertinggi. Dalam rangka untuk membuat variabel baru dengan jenis tiga tingkat:

```
> saltadd1 <- saltadd
> levels(saltadd1) <- c("no", "yes", "missing")
> saltadd1[is.na(saltadd)] <- "missing"
> summary(saltadd1)
     no     yes missing
     37      43      20
> summary(aov(age ~ saltadd1))
             Df  Sum Sq Mean Sq F value Pr(>F)
saltadd1      2   114.8    57.4  0.4484   0.64
Residuals    97 12421.8   128.1
```

Karena tidak ada cukup bukti bahwa kelompok yang hilang adalah penting dan untuk alasan tambahan dari kesederhanaan, kita akan mengabaikan kelompok ini dan melanjutkan analisis dengan variabel asli 'saltadd' (penambahan garam) yang hanya terdiri dari dua tingkat. Sebelum melakukan hal ini, model regresi sederhana dan garis regresi yang pertama dipasang.

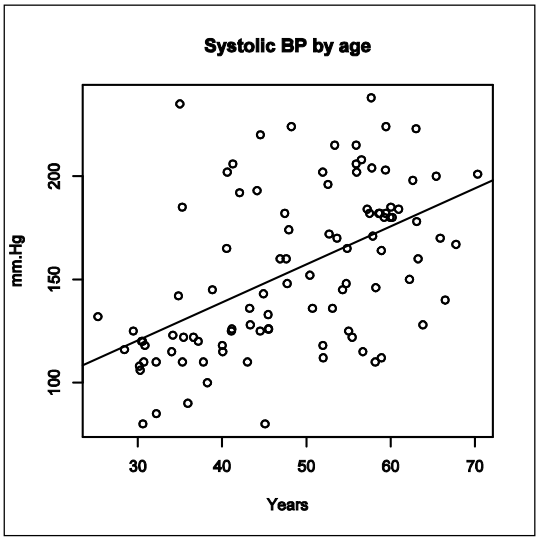
```
> lm1 <- lm(sbp ~ age)
> summary(lm1)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  65.1465    14.8942   4.374 3.05e-05
age           1.8422     0.2997   6.147 1.71e-08
Residual standard error: 33.56 on 98 degrees of freedom
Multiple R-Squared: 0.2782,     Adjusted R-squared: 0.2709
F-statistic: 37.78 on 1 and 98 DF,  p-value: 1.712e-08
```

Meskipun nilai *R-squared* tidak terlalu tinggi, kecil nya nilai *P-Value* menunjukkan pengaruh penting dari usia pada tekanan darah sistolik.

Sebuah *scatterplot* usia terhadap tekanan darah sistolik sekarang ditampilkan dengan penambahan garis regresi dengan menggunakan fungsi '*abline'*, yang telah disebutkan sebelumnya dalam bab 11. Fungsi ini dapat menerima bentuk argumen yang berbeda, termasuk objek regresi.

Jika objek ini memiliki metode 'koefisiens', dan ia mengembalikan vektor dengan panjang 1, maka nilai tersebut diambil sebagai kemiringan garis yang melalui titik asal, jika dua nilai pertama diambil sebagai *intercept* dan kemiringan, seperti kasus untuk 'lm1'.

```
> plot(age, sbp, main = "Systolic BP by age", xlab =
  "Years",
    ylab = "mm.Hg")

> coef(lm1)
(Intercept)         age
    65.1465      1.8422

> abline(lm1)
```




Eksplorasi berikutnya dari residu menunjukkan penyimpangan yang tidak signifikan dari normalitas dan tidak ada pola. Rincian ini dapat diadopsi dari teknik yang dibahas dalam bab sebelumnya. Langkah berikutnya adalah untuk menyediakan pola plot yang berbeda untuk kelompok yang berbeda dari kebiasaan garam.

```
> lm2 <- lm(sbp ~ age + saltadd)
> summary(lm2)
====================
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  63.1291    15.7645   4.005 0.000142
age           1.5526     0.3118   4.979 3.81e-06
saltaddyes   22.9094     6.9340   3.304 0.001448
---
Residual standard error: 30.83 on 77 degrees of freedom
Multiple R-Squared: 0.3331,     Adjusted R-squared: 0.3158
F-statistic: 19.23 on 2 and 77 DF,  p-value: 1.68e-07
```

Pada rata-rata, selisih satu tahun usia meningkatkan tekanan darah sistolik sebesar 1,5 mmHg. Penambahan garam meja meningkatkan tekanan darah sistolik secara signifikan sekitar 23 mmHg.

Serupa dengan metode yang digunakan dalam bab sebelumnya, langkah berikut membuat sebuah frame kosong untuk plot:

```
> plot(age, sbp, main="Systolic BP by age", xlab="Years",
  ylab="mm.Hg", type="n")
```

Tambahkan lingkaran biru berongga untuk subjek yang tidak menambahkan garam meja.

```
> points(age[saltadd=="no"], sbp[saltadd=="no"], col="blue")
```

Kemudian tambahkan poin merah penuh untuk subjek yang menambahkan garam meja.

```
> points(age[saltadd=="yes"], sbp[saltadd=="yes"],
col="red", pch = 18)
```

Perhatikan bahwa titik-titik merah yang sesuai untuk mereka yang menambahkan garam meja lebih tinggi dari lingkaran biru. Tugas terakhir adalah dengan menarik dua garis regresi yang terpisah untuk masing-masing kelompok.

Karena model 'lm2' berisi 3 koefisien, yang perintah *abline* sekarang memerlukan argumen 'a' sebagai *intercept* dan 'b' sebagai *slope* (kemiringan).

```
> coef(lm2)
(Intercept)         age  saltaddyes
  63.129112    1.552615   22.909449
```

Sekarang kita memiliki dua garis regresi untuk digambar, satu untuk setiap kelompok. *Intercept* untuk bukan pengguna garam akan menjadi koefisien pertama dan untuk pengguna garam akan menjadi yang pertama ditambah ketiga. Kemiringan untuk kedua kelompok adalah sama. Jadi *intercept* untuk bukan pengguna garam adalah:

```
> a0 <- coef(lm2)[1]
```

Untuk pengguna garam, *intercept* adalah koefisien pertama ditambah koefisien ketiga:

```
> a1 <- coef(lm2)[1] + coef(lm2)[3]
```

Untuk kedua kelompok, kemiringan adalah tetap di:

```
> b <- coef(lm2)[2]
```

Sekarang garis regresi pertama (yang lebih rendah) adalah ditarik dengan

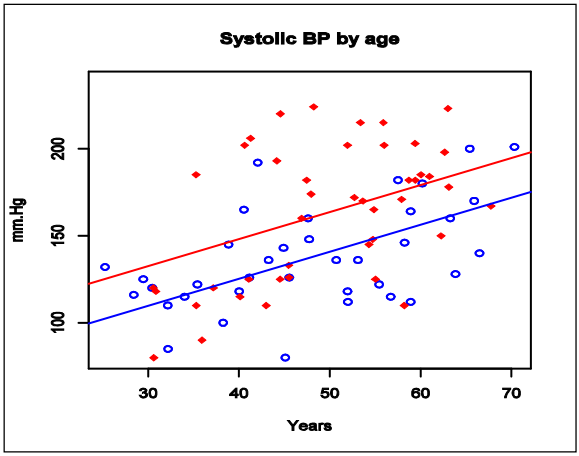
warna biru, kemudian yang lain merah.

```
> abline(a = a0, b, col = "blue")
> abline(a = a1, b, col = "red")
```

Perhatikan bahwa sumbu X tidak dimulai dari nol. Jadi *intercept* berada di luar bingkai plot.

Garis merah adalah untuk titik-titik merah penambah garam dan garis biru untuk titik-titik biru bukan penambah garam. Dalam model ini, usia memiliki efek tetap yang tidak tergantung (independen) pada tekanan darah sistolik.

Lihatlah distribusi titik-titik dari dua warna, titik merah lebih tinggi daripada yang biru, terutama pada bagian kanan grafik. Untuk kesesuaian baris dengan kemiringan yang berbeda, dibuatlah model yang baru, yaitu model dengan interaksi.




Langkah berikutnya adalah menyiapkan model dengan kemiringan yang berbeda (atau berbeda 'b' untuk argumen *abline*) untuk baris yang berbeda. Model ini memerlukan interaksi jangka antara 'saltadd' dan 'umur'.

```
> lm3 <- lm(sbp ~ age * saltadd)
> summary(lm3)
Call:
lm(formula = sbp ~ age * saltadd)
===============
```

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     78.0066    20.3981   3.824 0.000267 ***
age              1.2419     0.4128   3.009 0.003558 **
saltaddyes     -12.2540    31.4574  -0.390 0.697965
age:saltaddyes   0.7199     0.6282   1.146 0.255441
---
Multiple R-Squared: 0.3445,     Adjusted R-squared: 0.3186
F-statistic: 13.31 on 3 and 76 DF,  p-value: 4.528e-07
```

Pada bagian formula model, 'usia * saltadd' adalah sama dengan 'usia + saltadd + usia : saltadd'. Keempat koefisien ditampilkan dalam ringkasan dari model.Mereka juga dapat diperiksa sebagai berikut.

```
> coef(lm3)
   (Intercept)           age    saltaddyes  age:saltaddyes
    78.0065572    1.2418547    -12.2539696       0.7198851
```

Koefisien pertama adalah intercept dari garis yang dipasang di kalangan bukan pengguna garam.

Untuk intercept pengguna garam, suku (istilah) dan keempat semua nol (sejak usia sama dengan nol) tetapi yang ketiga harus disimpan seperti itu. Suku ini negatif. Intercept pengguna garam tersebut lebih rendah dibandingkan dengan bukan pengguna garam.

```
> a0 <- coef(lm3)[1]
> a1 <- coef(lm3)[1] + coef(lm3)[3]
```

Untuk kemiringan bukan pengguna garam , koefisien kedua saja sudah cukup karena yang pertama dan ketiga tidak terlibat dengan setiap unit kenaikan usia dan suku keempat memiliki 'saltadd' 0. Kemiringan untuk kelompok pengguna garam termasuk koefisien kedua dan keempat sejak 'saltaddyes' adalah 1.

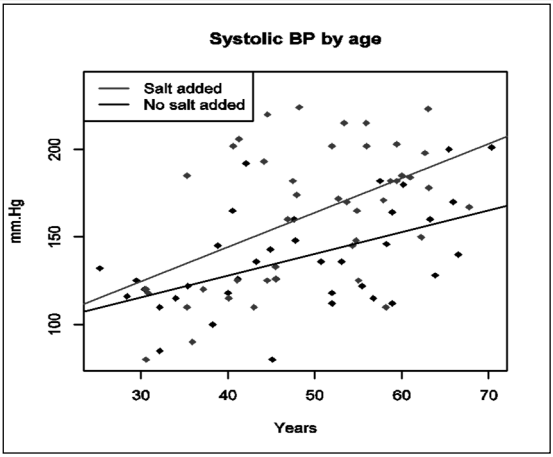
```
> b0 <- coef(lm3)[2]
> b1 <- coef(lm3)[2] + coef(lm3)[4]
```

Suku-suku ini digunakan untuk menggambar dua garis regresi.

Menggambar kembali grafik tetapi kali ini dengan warna hitam mewakili bukan penambah garam .

```
> plot(age, sbp, main="Systolic BP by age", xlab="Years",
  ylab="mm.Hg", pch=18, col=as.numeric(saltadd))
```

```
> abline(a = a0, b = b0, col = 1)
> abline(a = a1, b = b1, col = 2)
> legend("topleft", legend = c("Salt added", "No salt
  added"),
  lty=1, col=c("red","black"))
```




Perhatikan bahwa 'as.numeric (saltadd)' mengubah tingkat faktor ke bilangan bulat 1 (hitam) dan 2 (merah), yang masing-masing mewakili bukan penambah garam dan penambah garam. Kode-kode warna berasal dari palet warna R.

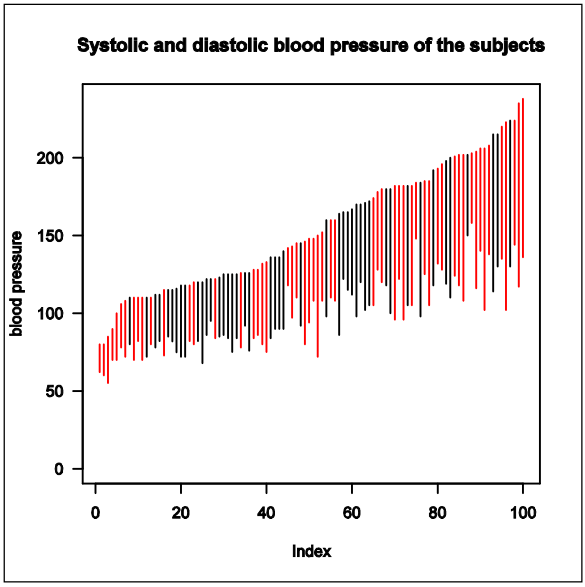
Model ini menunjukkan bahwa pada usia muda, tekanan darah sistolik dari kedua kelompok tidak jauh berbeda seperti dua garis yang berdekatan di sebelah kiri plot.Sebagai contoh, pada usia 25, perbedaannya adalah 5.7mmHg. Bertambahnya usia meningkatkan perbedaan antara dua kelompok. Pada usia 70 tahun, perbedaannya (selisihnya) sebesar 38mmHg. (Untuk mempermudah, prosedur untuk perhitungan kedua tingkat perbedaan, dilewati dalam catatan ini). Dalam aspek ini, usia memodifikasi efek penambahan garam meja.

Di sisi lain kemiringan usia 1.24mmHg per tahun di antara mereka yang tidak menambahkan garam tetapi menjadi 1,24 0,72 = 1.96mmHg antara penambah garam. Jadi, penambahan garam memodifikasi efek usia. Interaksi adalah istilah statistik sedangkan efek modifikasi adalah istilah epidemiologi setara.

Koefisien istilah interaksi 'umur: saltaddyes' tidak signifikan secara statistik. Kedua kemiringan hanya berbeda secara kebetulan.

## Latihan

Plot tekanan darah sistolik dan diastolik dari subjek, menggunakan warna merah untuk laki-laki dan biru untuk perempuan seperti yang ditunjukkan pada gambar berikut. [Petunjuk:segmen]




Periksa apakah ada perbedaan yang signifikan dari tekanan darah diastolik antara laki-laki dan perempuan setelah penyesuaian untuk usia.

# BAB 13

# Hubungan Kelengkungan (Curvilinear Relationship)

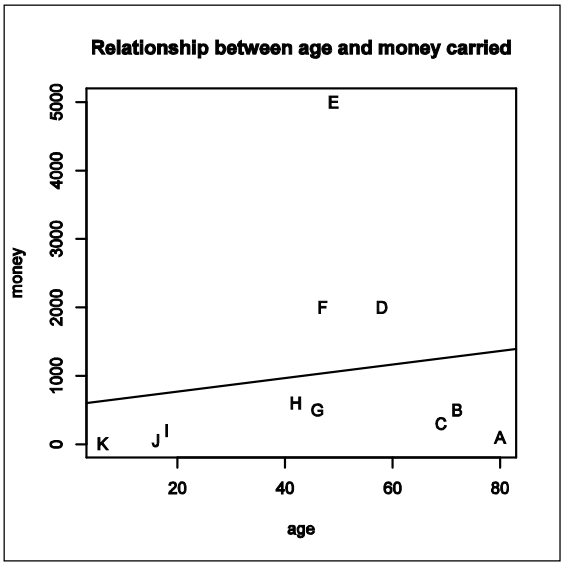
Contoh: Pembawaan Uang (uang yang dibawa) dan usia

Bab ini kembali menggunakan data keluarga dan mengeksplorasi hubungan antara uang yang dibawa dan usia.

```
> zap()
> data(Familydata)
> use(Familydata)
> des()
> plot(age, money, pch=" ")
```

Perintah di atas ekuivalen dengan :

```
> plot(age, money, type="n")
```




Untuk menempatkan 'kode' sebagai teks pada titik-titik, tambahkan judul dan garis regresi, ketik perintah berikut:

> text(age, money, labels = code)

```
> title("Relationship between age and money carried")
> lm1 <- lm(money ~ age)
> abline(lm1)
```

'lm1' objek dapat diperoleh dengan menggunakan fungsi penjumlahan.

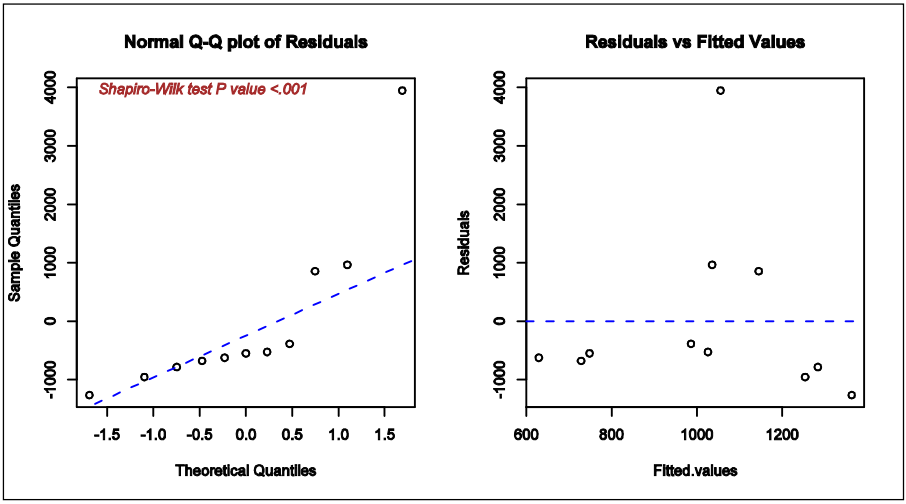
```
> summary(lm1)
============
Residual standard error: 1560 on 9 degrees of freedom
Multiple R-Squared: 0.0254,   Adjusted R-squared: -0.08285
F-statistic: 0.2349 on 1 and 9 DF,  p-value: 0.6395
```

Nilai R-squared sangat kecil untuk mengindikasikan kemiskinan. Hal ini dikonfirmasi oleh garis regresi kemiskinan pada grafik sebelumnya. Orang-orang

yang memiliki umur sekitar 40-60 tahun cenderung untuk membawa lebih banyak uang dibandingkan dengan kelompok usia lainnya.

Memeriksa residual.

```
> Residuals <- resid(lm1)
> Fitted.values <- fitted(lm1)
> windows(9,5)
> opar <- par(mfrow=c(1,2))
> shapiro.qqnorm(Residuals)
> plot(Fitted.values, Residuals, main="Residuals vs Fitted")
> abline(h=0, lty=3, col="blue")
```




Dari plot di atas diketahui bahwa residual tidak berdistribusi normal.

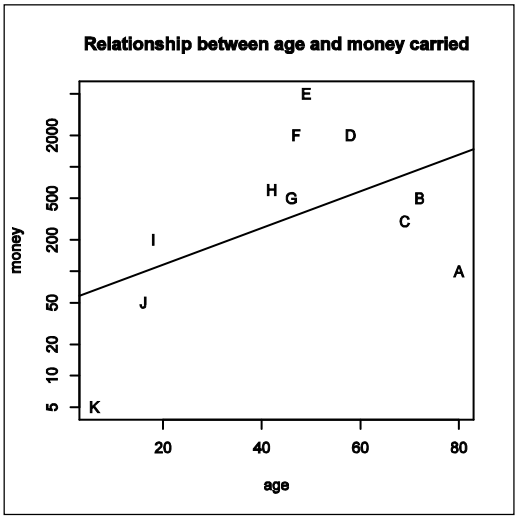
Untuk mengatur ulang perangkat grafis kembali ke jenis pengaturan original, gunakan sintak di bawah ini :

```
> par(opar)
```

Variasi dalam uang biasanya berdistribusi eksponensial. Dengan menguunakan logaritma dapat membantu memperoleh model yang cocok.

```
> plot(age, money, type="n", log = "y",
  main = "Relationship between age and money carried")
```

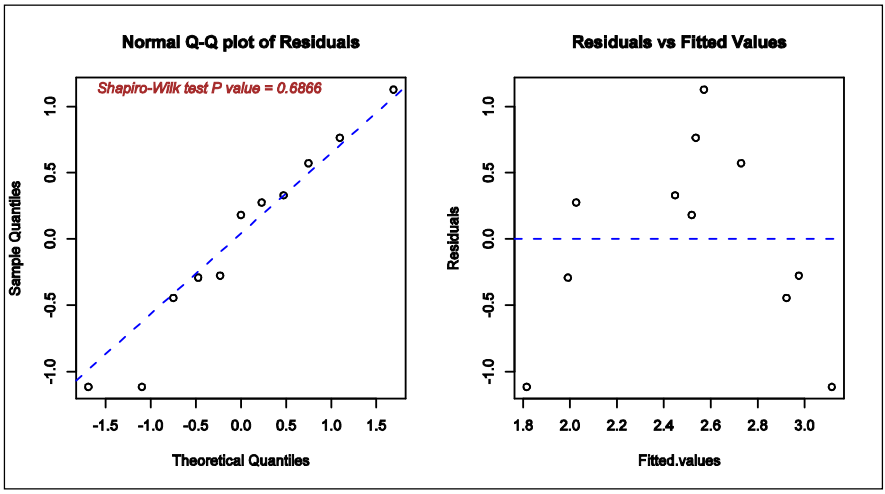
```
> text(age, money, labels = code)
> lm2 <- lm(log10(money) ~ age)
> abline(lm2)
```




Terlihat bahwa titik-titik pengamatannya tidaklah menghasilkan informasi yang memadai. Residual dapat diperiksa dengan cara berikut. Dengan skala log dari sumbu y, distribusi hubungan cenderung lengkung. Sehingga , menggambarkan garis lurus regresi melalui titik pengamatan yang berpola demikian tidaklah tepat.

Residu dapat diperiksa sebagai berikut:

```
> Residuals <- resid(lm2)
> Fitted.values <- fitted(lm2)
> windows(9,5)
> opar <- par(mfrow=c(1,2))
> shapiro.qqnorm(Residuals)
> plot(Fitted.values, Residuals, main="Residuals vs Fitted")
> abline(h=0, lty=3, col="blue")
```




Indicating that perhaps we need to include a quadratic term of age in the model. Residunya sekarang terlihat berdistribusi normal. Namun, nilai-nilai residunya yang tinggi di tengah kisaran nilai *fitted* (nilai kesesuaian), menunjukkan bahwa mungkin kita perlu memasukkan bentuk kuadrat usia dalam model.

Agar sesuai dengan garis regresi di bawah skala log tapi dengan (skala non-log) linier nilai akan terlalu rumit. Sebuah cara yang lebih baik akan mengubah 'uang' menjadi variabel baru pada skala log basis 10 dan cocok dengan model baru dengan rentang kuadrat usia.

```
> lm3 <- lm(log10(money) ~ age + I(age^2))
> summary(lm3)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.102650   0.338502    0.30  0.76944
age          0.125355   0.017641    7.11  0.00010
I(age^2)    -0.001268   0.000201   -6.30  0.00023

Residual standard error: 0.332 on 8 degrees of freedom
Multiple R-Squared: 0.875,      Adjusted R-squared: 0.844
F-statistic:   28 on 2 and 8 DF,  p-value: 0.000243
```

Kedua nilai baik yang disesuaikan (adjusted) dan non-adjusted R-squared bernilai tinggi. Penambahan istilah usia kuadrat meningkatkan model secara

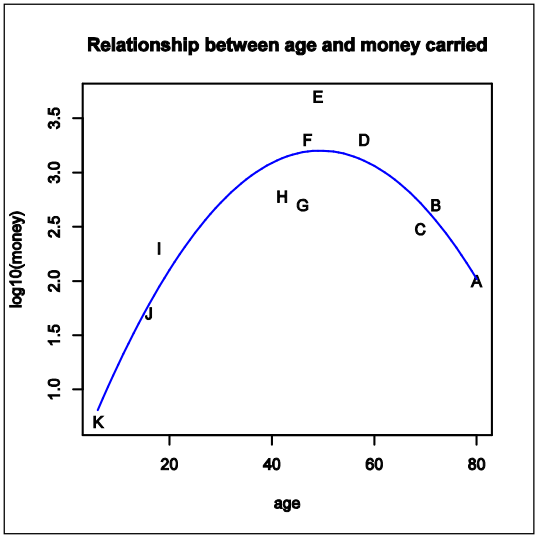
substansial dan secara statistik signifikan. Langkah selanjutnya adalah menyesuaikan dengan garis regresi, tugas yang tidak mudah.

Sebuah garis regresi adalah garis yang menghubungkan nilai-nilai kesesuaian. Ada sangat sedikit titik dari nilai kesesuaian dalam model. Sebuah frame data baru sekarang dibuat untuk memasukkan variabel 'umur' baru mulai dari 6 to 80 (yang merupakan rentang usia subyek kita) dan usia-kuadrat yang sesuai panjang.

```
> new <- data.frame(age = 6:80, age2 = (6:80)^2)
```

Kemudian nilai prediksi dari data frame dihitung berdasarkan model terakhir.

```
> predict1 <- predict.lm(lm3, new)
> plot(age, log10(money), type="n", ylab = "log10(money)",
   main="Relationship between age and money carried",)
> text(age, log10(money), labels = code)
> lines(new$age, predict1, col = "blue")
```




**Nilai maksimum dalam model kuadrat**

Model kuadrat menjelaskan bahwa, orang muda katakan "K" yang berusia 5 tahun membawa sangat sedikit uang. Peubah uang meningkat bersamaan dengan usia dan memuncak di antara 40-50 tahun. Kemudian nilai menurun

saat usia bertambah.

Nilai prediksi maksimum adalah

```
> max(predict1)
[1] 3.2012
```

Nilai uang yang sesuai adalah

```
> 10^max(predict1)
[1] 1589.4
```

Nilai umur adalah

```
> new$age[which.max(predict1)]
[1] 49
```

Namun, perhitungan matematis yang lebih tepat dari koefisien dapat diperoleh sebagai berikut:

```
> coef(lm4)
(Intercept)          age    I(age^2)
  0.1026501   0.1253546  -0.0012677

> a <- coef(lm3)[3]
> b <- coef(lm3)[2]
> c <- coef(lm3)[1]
> x <- -b/(2*a); x   # 49.441
```

Nilai yang sesuai di sumbu Y adalah

```
> y <- a * x^2 + b * x + c
> y   # 3.20148
```

Terakhir, nilai uang yang sesuai adalah:

```
> 10^y  # 1590.3
```

Kesimpulan dari model ini adalah bahwa pada usia 49 tahun, rata-rata orang akan membawa sekitar 1.590 baht. Jumlah ini lebih rendah dari nilai sebenarnya dari uang yang dibawa oleh "E", yaitu sebesar 5.000 baht atau lebih dari tiga kali lebih tinggi.

```
> 10^(log10(money)[code=="E"]-y)  # 3.1441
```

## Model lengkung bertingkat

Ada laki-laki dan perempuan dalam keluarga. Sebagai latihan, dua kurva paralel akan digunakan untuk menyesuai dengan data.

```
> lm4 <- lm(log10(money) ~ sex + age + I(age^2))
> summary(lm4)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.027252   0.338084    0.08  0.93801
sexM         0.239320   0.207432    1.15  0.28648
age          0.126284   0.017305    7.30  0.00016
I(age^2)    -0.001288   0.000198   -6.51  0.00033

Residual standard error: 0.325 on 7 degrees of freedom
Multiple R-Squared: 0.895,      Adjusted R-squared: 0.85
F-statistic: 19.9 on 3 and 7 DF,  p-value: 0.000834
```

Model 'lm4' memberikan sedikit lebih tinggi nilai R-squared dari 'LM3'. Jenis kelamin ("M" dibandingkan dengan "F") tidak signifikan. Kami menggunakan model ini untuk latihan.

```
> plot(age, log10(money), type="n", ylab = "log10(money)"
    main = "Relationship between age and money carried")
> text(age, log10(money), labels=code, col=unclass(sex))
```

Perhatikan bahwa baris pertama adalah sama seperti plot sebelumnya. Namun baris kedua, membedakan jenis kelamin dengan warna. Ketika 'Jenis kelamin', yang merupakan faktor, adalah *unclassed*, nilai-nilai menjadi urutan numerik dari tingkat. "F" berkode 1 dan "M" adalah kode 2 seperti yang diberikan dalam palet warna default **R**.

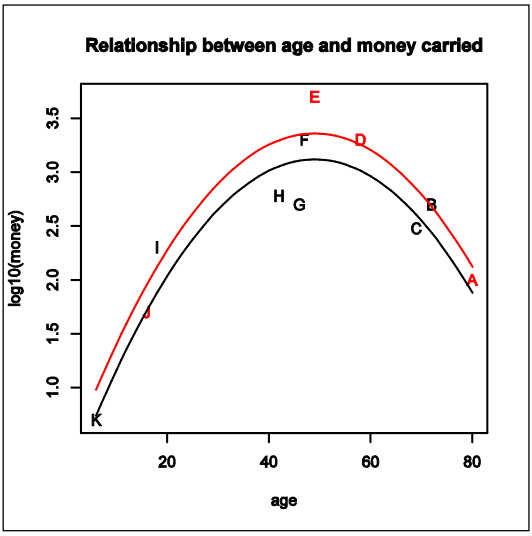
```
> age.frame2.male <- data.frame(age = 6:80, age2 = (6:80)^2,
   sex = factor ("M"))
> predict2.male <- predict.lm(lm4, age.frame2.male)
```

Perintah pertama menciptakan sebuah frame data yang berisi variabel yang digunakan dalam 'lm4'. Perhatikan bahwa 'jenis kelamin' di sini adalah terbatas pada laki-laki. Perintah kedua menciptakan vektor baru berdasarkan 'lm4' dan frame data baru. Pertama kita menarik garis untuk laki-laki.

```
> lines(age.frame2.male$age, predict2.male, col = 2)
```

Baris untuk perempuan.

```
> age.frame2.female <- data.frame(age = 6:80, age2 =
  (6:80)^2, sex = factor("F"))
> predict2.female <- predict.lm(lm4, age.frame2.female)
> lines(age.frame2.female$age, predict2.female, col=1)
```




Garis merah terletak konsisten di atas garis hitam, karena model kami tidak termasuk bentuk interaksi. Untuk setiap nilai usia, laki-laki cenderung untuk membawa uang 102,4 atau 1,738 kali lebih daripada perempuan. Namun, perbedaannya tidak signifikan.

## *Dari usia ke kelompok usia*

Sejauh ini, kita telah menganalisis efek usia sebagai variabel kontinu. Dalam sebagian besar analisis data epidemiologi, usia sering ditransformasikan ke variabel kategoris dengan memotong ke dalam kelompok umur. Untuk dataset kecil, kita membagi subjek menjadi anak-anak, dewasa dan lanjut usia subyek dengan titik memotong 20 dan 40 tahun dengan dua ekstrim 0 dan 85 tahun.

```
> agegr <- cut(age, breaks = c(0, 20, 60, 85),
   labels = c("Child", "Adult", "Elder"))
```

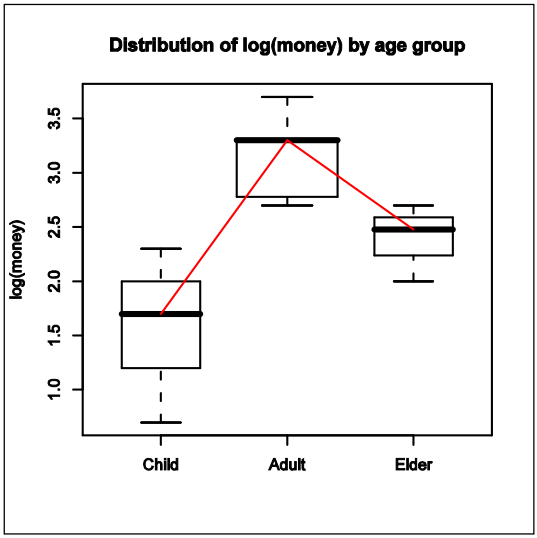
Metode pemotongan telah dijelaskan dalam Bab 2. Di sini, kami menempatkan label tertentu untuk menggantikan nama-nama bin default "(0,20 ]","( 20,60]" dan "(60,80]". Kemudian garis ditarik untuk bergabung dengan rata-rata log (uang) dari kelompok usia, yang di baris ketiga dari 'a'.

Untuk menggambarkan perubahan log (uang) dengan usia, serangkaian plot kotak yang digambar dengan parameter statistik disimpan dalam sebuah objek baru 'a'.

```
> a <- boxplot(logmoney ~ agegr, varwidth = TRUE)
```

Kemudian garis ditarik untuk bergabung dengan rata-rata log (uang) dari kelompok usia, yang di baris ketiga dari 'a'.

```
> lines(x = 1:3, y = a$stats[3, ], col = "red")
> title(main = "Distribution of log(money) by age group",
   ylab = "log(money)")
```



## Pemodelan dengan variabel kategori bebas

Sebuah model baru menambahkan variabel 'agegr' kategoris.

```
> lm5 <- lm(log10(money) ~ sex + agegr)
> summary(lm5)
=================== Lines omitted ===============
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.510      0.343    4.40   0.0031
sexM           0.169      0.351    0.48   0.6436
agegrAdult     1.578      0.408    3.87   0.0062
agegrElder     0.826      0.456    1.81   0.1129
=================== Lines omitted ===============
```

Ada dua parameter kelompok umur dalam model; "Dewasa" dan "Lansia". Tingkat pertama, "Anak", yang dihilangkan karena tingkat rujukan. Ini berarti tingkat lainnya akan dibandingkan dengan tingkat ini. Dewasa dilakukan $10^{1.578}$ atau uang sekitar 38 kali lebih dari anak-anak, yang secara statistik signifikan. Lansia dilakukan $10^{0.8257}$ = 6,7 kali lebih banyak uang daripada anak-anak, tetapi tidak signifikan secara statistik.

Kita bisa memeriksa pola kontras sebagai berikut:

```
> contrasts(agegr)
       Adult Elder
Child      0     0
Adult      1     0
Elder      0     1
```

Kolom-kolom matriks adalah variabel yang muncul dalam model. Baris menunjukkan semua tingkatan. Kolom 'Dewasa' dalam model adalah sama dengan 1 jika ‘agegr’ sama dengan "dewasa" dan nol sebaliknya. Kolom 'Lansia' adalah 1 ketika 'agegr' adalah "Lansia" dan nol sebaliknya. Tidak ada kolom 'Anak'. Ketika kedua 'Dewasa' dan 'Lansia' adalah sama dengan nol, model kemudian memprediksi nilai 'agegr' menjadi "Anak". Jika "Dewasa" diperlukan untuk menjadi tingkat acuan, kontras dapat diubah.

```
> contrasts(agegr) <- contr.treatment(levels(agegr), base=2)
```

Perintah di atas merubah kelompok ke tingkat 2.

```
> contrasts(agegr)
       Child Elder
```

```
Child       1      0
Adult       0      0
Elder       0      1
```

Kolom 'Dewasa' sekarang hilang. Tipe lain dari kontras juga dapat ditentukan. Lihat referensi untuk lebih jelasnya.

```
> summary(lm(log10(money) ~ sex + agegr))
================== Lines omitted =================
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)     3.088      0.286   10.78  1.3e-05
sexM            0.169      0.351    0.48   0.6436
agegrChild     -1.578      0.408   -3.87   0.0062
agegrElder     -0.752      0.408   -1.84   0.1079
================== Lines omitted =================
```

Perhatikan bahwa koefisien 'Anak' adalah negatif bahwa 'Dewasa' dari model 'lm5'. Selain itu, secara signifikan, orang tua tidak membawa uang kurang dari orang dewasa.

## Referensi

Venables, W. N. and Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth edition. Springer.

## Latihan

Apa yang akan terjadi di 'lm3' jika log basis 2 digunakan selain log basis 10? Apakah kesimpulan akan sama?

BAB 14

# Generalized Linear Models

## *Dari lm ke glm*

Pemodelan linear menggunakan fungsi lm didasarkan pada metode kuadrat terkecil. Konsepnya adalah meminimalkan residual jumlah kuadrat. Pemodelan dari lm ekuivalen dengan analisis varian yang menggunakan fungsi aov. Perbedaannya adalah bahwa yang pertama berfokus pada koefisien variabel bebas sedangkan yang kedua berfokus pada jumlah kuadrat.

Pemodelan linear umum (GLM) seperti sebutannya, lebih umum dari model linear. Metode ini berdasarkan fungsi likelihood. Saat likelihood bernilai maksimum maka diperoleh koefisien dan varians (serta standard error) dari variable bebas. Sedangkan pemodelan linear klasik mengasumsikan variable terikat didefinisikan dalam skala kontinu, seperti kekurangan darah pada contoh sebelumnya, (dan asumsi normalitas error dan varians konstan), GLM dapat mengatasi hasil yang berupa proporsi, distribusi Poisson (berhingga) dan lainnya seperti distribusi gamma dan binomial negatif.

Kita akan mulai dengan hasil pada skala kontinu pada contoh sebelumnya, kekurangan darah dan infeksi cacing.

```
> zap()
> data(Suwit)
> use(Suwit)
> bloodloss.lm <- lm(bloss ~ worm)
> summary(bloodloss.lm)
```

Hasilnya juga telah diperoleh pada bab sebelumnya.

Kita kita tampilkan model regresi linear umum menggunakan fungsi glm. Untuk fungsi glm standar *family* merupakan distribusi Gaussian, maka argumen 'family' dapat diabaikan.

```
> bloodloss.glm <- glm(bloss ~ worm)
> summary(bloodloss.glm)
Call:
glm(formula = bloss ~ worm)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-15.8461  -10.8118    0.7502    4.3562   34.3896

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 10.847327   5.308569   2.043   0.0618
worm         0.040922   0.007147   5.725 6.99e-05

(Dispersion parameter for gaussian family taken to be
  188.882)

    Null deviance: 8647.0  on 14  degrees of freedom
Residual deviance: 2455.5  on 13  degrees of freedom
AIC: 125.04

Number of Fisher Scoring iterations: 2
```

Dengan menggunakan data frame dan formula yang sama, contohnya 'bloss ~ worm', hasil yang diperoleh menggunakan fungsi 'lm' dan 'glm' untuk residual (disebut deviasi residual dalam 'glm'), koefisien dan standar errornya adalah sama. Bagaimanapun, terdapat banyak atribut pada fungsi terakhir.

## Model attributes

```
> attributes(bloodloss.lm)
$names
 [1] "coefficients"  "residuals"   "effects"     "rank"
 [5] "fitted.values" "assign"      "qr"
  "df.residual"
 [9] "xlevels"       "call"        "terms"       "model"
$class
[1] "lm"

> attributes(bloodloss.glm)
$names
 [1] "coefficients" "residuals"  "fitted.values"
 [4] "effects"      "R"          "rank"
 [7] "qr"           "family"     "linear.predictors"
[10] "deviance"     "aic"        "null.deviance"
[13] "iter"         "weights"    "prior.weights"
[16] "df.residual"  "df.null"    "y"
[19] "converged"    "boundary"   "model"
[22] "call"         "formula"    "terms"
[25] "data"         "offset"     "control"
[28] "method"       "contrasts"  "xlevels"
$class
[1] "glm" "lm"
```

Ingat bahwa 'bloodloss.glm' juga memiliki kelas *lm* sebagai tambahan ke dalam *glm*. Dua himpunan atribut serupa dengan sub-elemen untuk 'bloodloss.glm '. sub-elemen dengan nama yang sama pada dasarnya serupa. Dalam hal ini, 'deviance' dari perintah glm adalah sama dengan jumlah kuadrat residual.

```
> sum(bloodloss.glm$residuals^2)
[1] 2455.468
> bloodloss.glm$deviance
[1] 2455.468
```

Demikian pula, 'null.deviance' sama dengan total jumlah kuadrat dari selisih jumlah kekurangan darah individual dan rataannya.

```
> sum((bloss-mean(bloss))^2)
[1] 8647.044
> bloodloss.glm$null.deviance
[1] 8647.044
```

Beberapa atribut dalam 'glm' jarang digunakan tetapi beberapa tidak, seperti

'aic' yang sangat membantu. Akan ada diskusi lebih lanjut mengenai hal ini pada bab berikutnya.

Some of the attributes in of the 'glm' are rarely used but some, such as 'aic', are very helpful. There will be further discussion on this in future chapters.

## Attributes of model summary

```
> attributes(summary(bloodloss.lm))
$names
 [1] "call"          "terms"         "residuals"     "coefficients"
 [5] "aliased"       "sigma"         "df"            "r.squared"
 [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
$class
[1] "summary.lm"

> attributes(summary(bloodloss.glm))
$names
$names
 [1] "call"           "terms"          "family"
 [4] "deviance"       "aic"            "contrasts"
 [7] "df.residual"    "null.deviance"  "df.null"
[10] "iter"           "deviance.resid" "coefficients"
[13] "aliased"        "dispersion"     "df"
[16] "cov.unscaled"   "cov.scaled"

$class
[1] "summary.glm"
```

Sebagian besar proporsi elemen dari kedua himpunan atribut mengulang model ini. Atribut tambahan termasuk didalamnya R squared dalam model 'lm' dan matriks kovarians ('cov.unscaled') dalam kedua model. Matriks kovarians digunakan untuk penghitungan standard error dan interval kepercayaan koefisien 95%.

## Matriks Kovarians

Ketika terdapat dua atau lebih variable penjelas dan independent, variasi kolektif dinotasikan sebagai kovarians (dibandingkan dengan varians variabel tunggal). Kovarians tersebut disimpan sebagai matriks simetris karena satu variabel dapat berkovari dengan variabel lainnya. Sebuah matriks kovarians dapat diskalakan 'scaled' atau tidak diskalakan 'unscaled'. Salah satu dari model 'lm' memberikan 'cov.unscaled' sedangkan 'glm' dapat menampilkan keduanya.

```
> vcov(bloodloss.glm)    # or
  summary(bloodloss.glm)$cov.scaled
             (Intercept)          worm
(Intercept) 28.18090491 -2.822006e-02
worm        -0.02822006  5.108629e-05
> summary(bloodloss.glm)$cov.unscaled
               (Intercept)          worm
(Intercept)  0.1491983716 -1.494057e-04
worm        -0.0001494057  2.704665e-07
```

matriks kovarians terakhir dapat pula diperoleh dari summary model linear sederhana.

```
> summary(bloodloss.lm)$cov.unscaled
```

Faktor scaling sebenarnya merupakan penyebaran atau kuadrat jumlah, yang merupakan jumlah dari kuadarat residual dibagi dengan derajat kebebasan residual. Dengan demikian matriks pertama dapat diperoleh dari

```
> summary(bloodloss.glm)$cov.unscaled *
  summary(bloodloss.glm)$dispersion
atau
> summary(bloodloss.lm)$cov.unscaled *
  summary(bloodloss.lm)$sigma^2
```

atau

```
> summary(bloodloss.lm)$cov.unscaled *
  sum(summary(bloodloss.lm)$residuals^2)/13
```

Matriks kovarian terskala digunakan untuk menghitung standard error koefisien. Bentuk diagonal dari matriks ini - dimana nama baris sama dengan nama kolom- merupakan nilai varians koefisien dengan nama yang serupa. Dengan menggunakan akar kuadrat dari bentuk ini akan menghasilkan koefisien standard error.

### *Menghitung standard error, nilai t dan selang kepercayaan 95%*

Standar error untuk 'worm' adalah

```
> vcov(bloodloss.glm)[2,2]^.5 -> se2
> se2
[1] 0.0071475
Hal ini dapat diperiksa terhadap summary koefisien.
> coef(summary(bloodloss.glm))
             Estimate Std. Error  t value    Pr(>|t|)
(Intercept) 10.847327 5.3085690  2.043362  0.06183205
worm         0.040922 0.0071475  5.725392  0.00006990
```

Selanjutnya, 't value' dapat dihitung dari pembagian koefisien terhadap standard error.

```
> coef(summary(bloodloss.glm))[2,1] /
  summary(bloodloss.glm)$cov.scaled[2,2]^.5 -> t2
> t2
```

atau

```
> 0.04092205 / 0.007147467 # 5.7254
```

P value merupakan peluang 't' dapat berada pada nilai ini atau nilai yang lebih ekstrim. Nilai yang lebih ekstrim dapat berada dikedua sisi atau tanda t value. Untuk itu, nilai P value dihitung dengan

```
> pt(q=t2, df=13, lower.tail=FALSE) * 2
[1] 6.9904e-05
```

Nilai ini sama dengan summary koefisiennya. Penjelasan lebih lanjut mengenai penrhitungan peluang dari distribusi t dapat diketahui dari 'help(TDist)' atau 'help(pt)'.

Akhirnya untuk menghitung selang kepercayaan 95% digunakan:

```
> beta2 <- coef(summary(bloodloss.glm))[2,1]; beta2
[1] 0.04092205
> ci2 <- beta2 + qt(c(0.025, 0.975), 13)*se2; ci2
[1] 0.02548089 0.05636321
```

Pada kenyataannya, **R** mempunyai perintah untuk menghitung selang kepercayaan 95% model dengan menggunakan:

```
> confint(bloodloss.lm)
                2.5 %    97.5 %
```

```
(Intercept) -0.621139 22.315793
worm         0.025481  0.056363
```

Hasilnya sama tetapi waktu yang digunakan lebih cepat. Ingat bahwa perintah confint(bloodloss.glm)menghasilkan perbedaaan selang kepercayaan yang rendah. Hal ini disebabkan karena fungsi tetap menggunakan distribusi distribusi t walaupun penggunaannya tidak layak.

### *Bagian lain 'glm'*

Seperti yang disebutkan sebelumnya, pemodelan linear atau 'lm', setelah digeneralisasi menjadi 'glm', dapat menampung lebih banyak pilihan variabel independen. Model harus memiliki *family*. Untuk memeriksanya digunakan:

```
> family(bloodloss.glm) # or bloodloss$family
Family: gaussian
Link function: identity
```

Pemodelan lm ekuivalen dengan glm yang termasuk dalam keluarga 'gaussian'. Fungsi penghubungnya adalah 'identity', yang berarti bahwa variabel independent tidak ditransformasi. Tipe 'family' dan link' lainnya akan diperlihatkan pada bab selanjutnya.

Karena fungsi penghubung adalah 'identity', 15 nilai dari predictor linear untuk keluarga 'glm' sama dengan nilai dugaannya (untuk kedua model 'lm' dan glm')

```
> all(fitted(bloodloss.glm) == predict(bloodloss.glm))
 [1] TRUE
```

Model 'glm' menghasilkan error menggunakan 'deviance'. Untuk model linear, nilai ini sama dengan jumlah kuadrat residual.

```
> bloodloss.glm$deviance
[1] 2455.468

> sum(summary(bloodloss.lm)$res^2)
[1] 2455.468
```

Interpretasi untuk error menggunakan model 'glm' sama dengan model linear: nilai deviasi yang besar mengindikasikan dugaan yang lemah.

Pemodelan linear umum memakai iterasi numerik untuk mendapatkan maksimum likelihood. Nilai maksimum likelihood yang diperoleh kecil karena maksimum likelihood adalah hasil kali peluang. Bentuk logaritma ini lebih baik

untuk digunakan. Mkasimum log likelihood dapat diperoleh dari fungsi berikut:

```
> logLik(bloodloss.glm)
'log Lik.' -59.51925 (df=3).
```

Semakin tinggi log likelihood (tidak negatif), semakin layak model tersebut. bagaimanapun, setiap model mempunyai masing-masing parameter penjelas. Banyak sekali parameter dapat berdampak kurang efisienPada saat mencocokkan model, yang diinginkan adalah mendapatkan model yang sangat mendekati kenyatan yang sebenarnya. Atribut model yang menyeimbangkan log-likelihood dengan jumlah parameter adalah nilai AIC yang merupakan singkatan dari "Akaike Information Criterion". Nilai ini sama dengan -2×log-likelihood + k×*npar*, dimana *k* merupakan faktor penalty (umumnya 2) dan *npar* menunjukkan jumlah parameter dalam model dugaan. Likelihood yang tinggi atau kelayakan yang baik akan ditunjukkan dengan nilai AIC yang rendah. Meskipun sejumlah besar parameter juga dihasilkan dalam AIC yang tinggi. Jumlah parameter untuk model ini adalah 3. AIC kita peroleh dari:

```
> -2*as.numeric(logLik(bloodloss.glm))+2*3
[1] 125.0385

> AIC(bloodloss.glm)
[1] 125.0385
```

AIC sangat berguna saat harus memilih antara model dari himpunan dataset yang sama. AIC dan atribut penting lainnya akan didiskusikan lebih jelas dalam bab selanjutnya.

## Referensi

Dobson, A. J. (1990). An Introduction to Generalized Linear Models. London: Chapman and Hall.
McCullagh P. and Nelder, J. A. (1989). Generalized Linear Models. London: Chapman and Hall.

## Latihan

Dalam dataset **BP**, gunakan perintah glm untuk menganalisa model dugaan tekanan darah systolic dari usia dan tambahkan table salt dengan dan tanpa interaksi. Gunakan untuk memilih model yang paling efisien. Uji pula asumsi kenormalannya.

BAB 15

# Regresi Logistik

## Distribusi dari keluaran biner

Dalam data epidemiologi, kebanyakan keluaran seringkali berupa dalam bentuk biner atau dikotomi. Sebagai contoh, dalam investigasi penyebab terjadi penyakit, status untuk keluaran yaitu penyakit, adalah terkena penyakit vs tidak terserang penyakit. Dalam studi mortalitas, keluarannya berupa hidup vs meninggal.

Untuk variabel kontinu seperti tinggi dan berat badan, jumlah representatif tunggal untuk populasi atau sample adalah mean dan median. Untuk data dikotomi, jumlah representatif adalah proporsi atau persentasi dari suatu tipe keluaran. Sebagai contoh, 'prevalensi' adalah proporsi populasi dengan penyakit tertentu. *Case-fatality* adalah proporsi terjadi kematian diantara orang yang terserang penyakit.

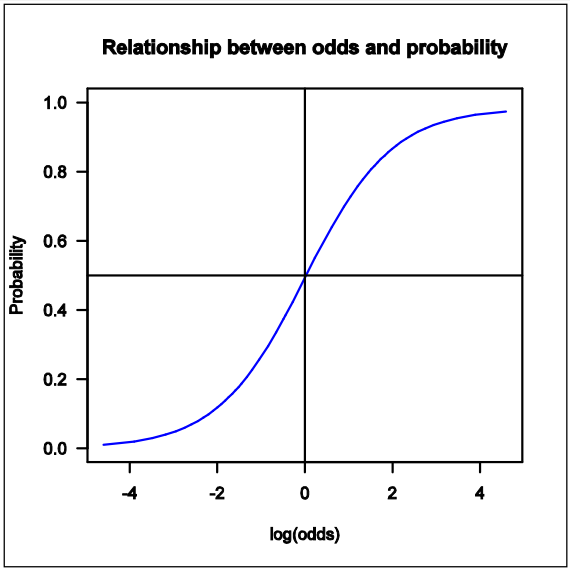
Istilah lainnya yang berhubungan adalah 'probability'. Proporsi adalah istilah mudah dan sederhana. Peluang menunjukkan kemungkinan yang lebih teoritis. Dalam kasus variabel dikotomi, proporsi digunakan untuk mengestimasi peluang.

Untuk penghitungan, keluaran sering direpresentasikan dengan 1 dan 0. Prevalensi merupakan mean dari jumlah yang terserang penyakit antara sampel

penelitian. Contohnya jika ada 50 sampel, 7 orang terserang penyakit (kode 1), 43 tidak terserang penyakit (kode 0) maka mean sama dengan 7/50 = 0.14, yang merupakan prevalensi.

Peluang digunakan berdasarkan kesederhanaannya. Untuk kalkulasi yang kompleks seperti regresi logistik, log(odds) atau logit adalah lebih baik. Jika P merupakan peluang terkena penyakit, 1-P merupakan peluang tidak terkena penyakit. Maka odd samdengan P/(1-P). hubungan antara peluang dan odds, terutama log(odds) dapat diplotkan :

```
> p <- seq(from=0, to=1, by=.01)
> odds <- p/(1-p)
> plot(log(odds), p, type="l", col="blue",
  ylab="Probability",
  main="Relationship between odds and probability", las=1)
> abline(h=.5)
> abline(v=0)
```




Peluang minimum 0, maksimum 1 dengan nilai tengah 0.5. The odds memiliki nilai korespodensinya pada saat 0, tak berhingga dan 1. Log(odds) atau yang sering disebut logit, memiliki kenaikan linear dengan –infinity dan +infinity dan 0 untuk titik tengah. Kurva diatas disebut kurva logistic. Menjadi linear dan

berskala seimbang, logit adalah skala yang lebih pantas untuk keluaran biner dibandingkan dengan peluang itu sendiri. Pemodelan logit (Y|**X**) ~ **βX** adalah bentuk umum dari regresi logistik. Hal ini berarti bahwa logit Y menghasilkan **X** (atau dibawah pengaruh **X**), dimana **X** menotasikan satu atau lebih variabel independent, dapat ditentukan dengan menjumlahkan keluaran antara setiap koefisien dengan nilai X nya.

Misalkan terdapat variabel independen dan dependen: $X_1$ and $X_2$. **βX** akan menjadi $\beta_0 + \beta_1X_1 + \beta_2X_2$, dimana $\beta_0$ merupakan intercept.

Dalam bahasan medis, keluaran biner Y (disebut juga dikotomi) sering merupakan often disease vs non-disease, dead vs alive, etc. **X** dapat berupa usia, jenis kelamin atau variabel prognostik lainnya. Antara variabel X ini, satu atau beberapa variabel diuji dengan spesifik hipotesis. Lainnya juga merupakan pembaur potensial yang disebut covariat.

Secara matematika, terlihat bahwa Pr(Y|**X**) sama dengan exp(**βX**)/(1 + exp(**βX**)). Sehingga regresi logistik sering digunakan untuk menghitung peluang keluaran dengan set of exposures diketahui. Contohnya, estimasi peluang untuk terserang penyakit jika diketahui himpunan usia, jenis kelamin dan kelompok perilaku, etc.

## *Contoh: Pembusukan Gigi*

Himpunan data **Decay** merupakan himpunan data sederhana yang terdiri dari dua variabel: 'decay' variabel biner dan 'strep' variabel kontinu.

```
> zap()
> data(Decay)
> use(Decay)
> des()
No. of observations =436
  Variable      Class          Description
1 decay         numeric        Any decayed tooth
2 strep         numeric        CFU of mutan strep.

> summ()
No. of observations =436
  Var. name Obs.  mean   median  s.d.   min.   max.
1 decay     436   0.63   1       0.48   0      1
2 strep     436   95.25  105     53.5   0.5    152.5
```

'decay' merupakan variabel dependen yang mengindikasikan apakah seseorang mengalami pembusukan gigi paling kurang satu gigi (1) atau tidak ada gigi busuk (0). Variabel independen 'strep' merupakan jumlah colony forming units (CFU) bakteri streptococci, sekelompok bakteri penyebab gigi busuk.

Prevelesi mengalami pembusukan gigi sama dengan mean dari variabel 'decay' yaitu 0.63. Untuk melihat variabel 'strep' ketik:

```
> summ(strep)
```

Plot menunjukkan bahwa luas dominannya memiliki nilai pada sekitar 150. Karena distribusi natural dari bakteri adalah logaritma, variabel yang ditransformasi dibuat dan digunakan sebagai variabel independen.

```
> log10.strep <- log10(strep)
> label.var(log10.strep, "Log strep base 10")
> glm0 <- glm(decay~log10.strep, family=binomial,
  data=.data)
> summary(glm0)
===============
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)    -2.554      0.518   -4.93  8.4e-07
log10.strep     1.681      0.276    6.08  1.2e-09
===============
AIC: 535.83
```

Kedua koefisien intercept dan 'log10.strep' adalah signifikan secara statistik.

Pr(>|z|)untuk 'log10.strep' merupakan P value dari uji Wald's. Uji ini untuk memeriksa apakah P value 1.681 berbeda signifikan dari 0 dan untuk kasus ini P value signifikan.

Intercept dugaan sebesar -2.554. Ini berarti bahwa pada saat 'log10.strep' sama dengan 0 (or strep bakteri equals 1 CFU), logit memiliki paling sedikit satu gigi rusak adalah -2.55. selanjutnya kita dapat menghitung nilai odd dan peluang.

```
> exp(-2.554) -> baseline.odds
> baseline.odds
[1] 0.07777
> baseline.odds/(1+baseline.odds) -> baseline.prob
> baseline.prob
[1] 0.072158
```

Terdapat Odd 0.077 atau peluang 7.2% untuk memiliki paling sedikit satu gigi

rusak jika jumlah CFU dari mutan bakteri streptococcus berada pada 1 CFU.

Koefisien 'log10.strep' sebesar 1.681. Untuk setiap unit kenaikan 'log10.strep', atau kenaikan 10 CFU, logit akan meningkat sebesar 1.681. peningkatan logit adalah konstan tetapi tidak dengan kenaikan peluang karena yang terakhir tidak dalam skala linear. Peluang untuk setiap titik CFU dihitung dengan menggantikan kedua koefisien yang diperolehd dari model. Sebagai contoh, pada 100 CFU, peluangnya adalah:

```
> coef(glm0)[1] + log10(100)*coef(glm0)[2]
(Intercept)
     0.8078
```

Untuk melihat hubungan keseluruhan dataset:

```
> plot(log10.strep, fitted(glm0))
```

Natural logistic kurva ditunjukkan secara terpisah. Untuk lebih jelas, interval sumbu X dan Y keduanya diperluas agar mendapatkan kurva yang sesuai.

```
> plot(log10.strep, fitted(glm0), xlim = c(-2,4),
  ylim=c(0,1), xlab=" ", ylab=" ", xaxt="n", las=1)
```

Vektor lainnya dengan nama serupa 'log10.strep' dibuat dalam bentuk data frame untuk memplotkan garis dugaan dalam grafik yang sama.

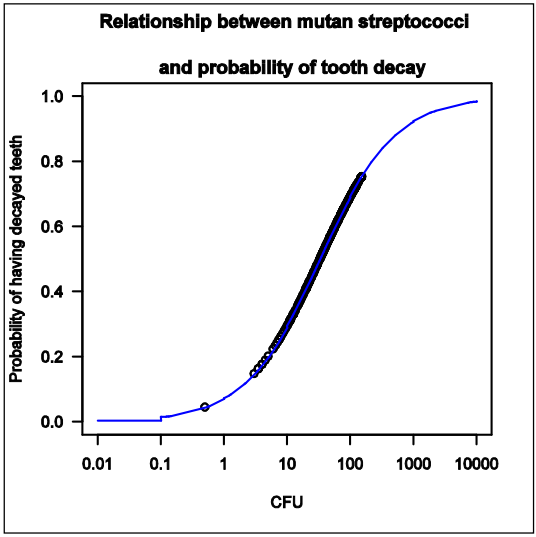
```
> newdata <- data.frame(log10.strep=seq(from=-2, to=4,
  by=.01))
> predicted.line <-
  predict.glm(glm0,newdata,type="response")
```

Nilai untuk estimasi garis pada perintah diatas harus dalam skala yang sama seperti variabel dependen. Karena variabel berada antara 0 atau 1, garis yang diprediksi harus berada diantaranya, ie. estimasi peluang untuk setiap nilai log10(strep).

```
> lines(newdata$log10.strep, predicted.line, col="blue")

> axis(side=1, at=-2:4, labels=as.character(10^(-2:4)))

> title(main="Relationship between mutan streptococci \n
  and probability of tooth decay", xlab="CFU",
  ylab="Probability of having decayed teeth")
```

Ingat bahwa penggunaan '\n' pada perintah diatas untuk memisahkan judul

yang panjang dalam dua garis.




## Regresi logistik dengan variabel independen biner

Contoh data pembusukan gigi diatas merupakan variabel kontinu 'log10.strep' seperti variabel independent. Dalam kebanyakan epidemiologi dataset, variabel independent sering berupa kategori. Ingat bahwa kita mempunyai dataset untuk penyebaran keracunan makanan di Thailand yang telah dianalisa pada Bab 7-9. Pada bab ini, kita akan menggunakan regresi logistik untuk menentukan model yang layak saat penyebab penyakit berupa variabel kategori. Pembaca disarankan untuk membandingkan hasil regresi logistik pada bab ini dengan analisis bertingkat pada bab sebelumnya.

```
> zap()
> load("chapter9.Rdata")
> use(.data)
> des()
```

Kita memodelkan 'case' sebagai variabel biner dan mengambil 'eclair.eat' sebagai variabel penjelas.

```
> glm0 <- glm(case ~ eclair.eat, family=binomial,
  data=.data)
> summary(glm0)
Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)      -2.923      0.265  -11.03   <2e-16
eclair.eatTRUE    3.167      0.276   11.48   <2e-16
=================== Lines omitted =================
```

Tampilan bagian atas sebenarnya adalah sebuah matriks yang diperoleh:

```
> coef(summary(glm0))
```

Epicalc memanipulasikan matriks dan memberikan tambahan tampilan agar lebih mudah dimengerti oleh kebanyakan epidemiologis.

```
> logistic.display(glm0)
Logistic regression predicting diseased

               OR (95% CI)          P(Wald's test) P(LR-
  test)
eating eclair  23.75 (13.82,40.79)  < 0.001        < 0.001

Log-likelihood =  -527.6075
No. of observations =  977
AIC value = 1059.2
```

Odd rasio regresi logistik diperoleh dari eksponensiasi estimasi, i.e. 23.75 yang muncul dari:

```
> exp(coef(summary(glm0))[2,1])
```

Interval kepercayaan 95% dari odd rasio diperoleh dari:

```
> exp(coef(summary(glm0))[2,1] + c(-1,1) * 1.96 *
  coef(summary(glm0))[2,2])
```

Nilai ini sama dengan perhitungan sederhana dari tabel 2-by-2 yang telah lebih dulu dijelaskan pada Bab 9. log-likelihood dan nilai AIC akan dibahas selanjutnya.

Nilai standar dalam *logistic.display* adalah 95% untuk interval kepercayaan dan digit diperlihatkan dalam dua decimal.

```
> args(logistic.display)
> help(logistic.display)
```

anda dapat merubah nilai standar tersebut dengan menambahkan argumen ekstra dalam perintah.

```
> logistic.display(glm0, alpha=0.01, decimal=2)
```

Jika data frame telah dimasukkan dalam perintah glm (menggunakan argumen 'data'), output akan menunjukkan gambaran variabel daripada nama variabel. The P value dari uji Wald's sama dengan seperti yang diperlihatkan oleh matriks koefisien yang diperoleh,

```
> coef(summary(glm0))
```

Ouput dari tampilan logistic juga mengandung hasil 'LR-test', yang memeriksa apakah likehood dari model yang diberikan, 'glm0', akan berbeda secara signifikan dari model tanpa 'eclair.eat', yang mana pada kasus ini akam menjadi model "null". Untuk variabel independen dua level, uji LR tidak menambahkan informasi penting lebih jauh karena uji Wald's telah dipergunakan untuk menguji hipotesis. Saat variabel independen memiliki lebih dari dua level, uji LR lebih penting dibandingkan dengan uji Wald's seperti yang diperlihatkan oleh contoh berikut:

```
> glm1 <- glm(case ~ eclairgr, family=binomial, data=.data)
> logistic.display(glm1)

Logistic regression predicting diseased

                      OR(95%CI)            P(Wald's test)
  P(LR-test)
pieces of eclair eaten:                                    <
  0.001
             ref.=0
   1                  17.57 (9.21,33.49)   < 0.001
   2                  22.27 (12.82,38.66)  < 0.001
   >2                 43.56 (22.89,82.91)  < 0.001

Log-likelihood =  -516.8236
No. of observations =  972
AIC value = 1041.6
```

Dengan menginterpretasikan uji Wald's, seseorang akan menyimpulkan bahwa keseluruhan level konsumsi kue sus akan signifikan. **R** mengasumsikan bahwa level pertama dari faktor independen merupakan level referen. Jika kita melevel ulang level referen menjadi 2 potong kue sus, uji Wald's akan memberikan hasil

yang berbeda.

```
> eclairgr <- relevel(eclairgr, ref="2")
> pack()
> glm2 <- glm(case ~ eclairgr, family=binomial, data=.data)
> logistic.display(glm2)

Logistic regression predicting diseased

                      OR(95%CI)         P(Wald's test)
  P(LR-test)
pieces of eclair eaten: ref.=2                          <
  0.001
   0                  0.04 (0.03,0.08)  < 0.001
   1                  0.79 (0.52,1.21)  0.275
   >2                 1.96 (1.28,2.99)  0.002
============================================================
  ==========
```

Hasil menunjukkan bahwa memakan satu potong kue sus tidak mengurangi resiko secara signifikan dibandingkan dengan memakan dua potong kue.

Sementara uji Wald's bergantung pada level reference dari variabel penjelas, uji LR fokus hanya pada kontribusi variabel seperti keseluruhan dan mengabaikan level referen. Kita akan mendiskusikan hal ini pada bab selanjutnya.

Selanjutnya cobalah variabel penjelas 'saltegg'.

```
> glm3 <- glm(case ~ saltegg, family = binomial, data=.data)
> logistic.display(glm3)

Logistic regression predicting case

              OR (95% CI)       P(Wald's test)  P(LR-test)
saltegg:      2.54 (1.53,4.22)  < 0.001         < 0.001
     Yes vs No

Log-likelihood =  -736.998
No. of observations =  1089
AIC value = 1478
```

Odd rasio untuk 'saltegg' signifikan secara statistic dan sama seperti yang terlihat pada tabulasi silang Bab 9. Jumlah catatan yang valid juga lebih tinggi dibandingkan dengan model yang mengandung 'eclairgr'.

**Catatan**:

_____

Seseorang selalu hati-hati saat menganalisis data yang mengandung data hilang. Metode untuk mengatasi masalah data hilang diluar batas buku ini dan untuk alas an kesederhanaan maka akan diabaika. Pembaca disarankan untuk memahami analsis data hilang untuk mendukung analisis selanjutnya.

Untuk memeriksa apakah odd rasio dibaurkan oleh 'eclairgr', dua variabel penjelas dimasukkan bersama dalam model selanjutnya.

```
> glm4 <- glm(case ~ eclairgr + saltegg, family=binomial)
> logistic.display(glm4, crude.p.value=TRUE)

Logistic regression predicting case

        crude OR(95%CI)  P value  adj. OR(95%CI)   P(Wald)
  P(LR-test)
eclairgr: ref.=2                                          <
  0.001
  0     0.04 (0.03,0.08) < 0.001  0.04 (0.03,0.08) < 0.001
  1     0.79 (0.52,1.21) 0.275    0.79 (0.51,1.21)   0.279
  >2    1.96 (1.28,2.99) 0.002    1.96 (1.28,2.99)   0.002

saltegg: 2.37 (1.4,3.99) 0.001    1.01 (0.53,1.93)   0.975
  0.975
   Yes vs No

Log-likelihood =  -516.823
No. of observations =  972
AIC value = 1043.6
```

Odd rasio variabel penjelas dalam 'glm4' adjusted untuk lainnya. Odd rasio sederhana sama persis dari model sebelumnya dengan hanya variabel tunggal. P value 'saltegg' ditunjukkan oleh 0.001 berdasarkan pengelompokan. Pada kenyataannya 0.00112 yang tidak lebih kecil dari 0.001. Epicalc, untuk alasan estetika, menampilkan P value sebagai

'< 0.001' kapanpun nilai original lebih kecil 0.001.

The adjusted odds ratios of 'eclairgr' tidak mengubah saran bahwa itu tidak dibaurkan oleh 'saltegg', meskipun odd rasio 'saltegg', berubah seiring kesatuan dan sekarang memiliki P value yang besar. Perbedaan antara adjusted odds

ratio dan odd rasio sederhana adalah indikasi bahwa 'saltegg' dibaurkan oleh 'eclairgr', yang merupakan faktor resiko bebas. Adjusted odds ratios ini hampir sama dengan yang diperoleh dari metode Mantel-Haenszel pada Bab 9.

Sekarang kita punya model yang terdiri dari dua variabel penjelas, kita dapat membandingkan model 'glm4' dan 'glm2' menggunakan perintah *lrtest*.

```
> lrtest(glm4, glm2)
Likelihood ratio test for MLE method
Chi-squared 1 d.f. =  0.0009809 , P value =  0.975
```

P value sebesar 0.975 serupa seperti 'P(LR-test)' dari 'saltegg' yang diperoleh dari perintah sebelumnya. Uji menentukan apakah penghapusan 'saltegg' dari model akan membawa perbedaan signifikan. Jika terdapat lebih dari satu variabel penjelas, 'P(LR-test)' dari *logistic.display* diperoleh dari perintah *lrtest* yang membandingkan model sebelumnya dengan model dengan salah satu atau beberapa variabel dihilangkan,sementara variabel lainnya tetap digunakan.

Regresi logistik memberikan sekaligus adjusted odds ratios secara bersamaan. Metode Mantel-Haenszel hanya memberikan odd rasio variabel utamanya. Keuntungan lainnya adalah regresi logistik dapat menangani perkalian kovariat secara serempak.

```
> glm5 <- glm(case~eclairgr+saltegg+sex, family=binomial)
> logistic.display(glm5)

Logistic regression predicting case

       crude OR(95%CI)    adj. OR(95%CI)    P(Wald's test)
  P(LR-test)
eclairgr: ref.=2                                          <
  0.001
   0   0.04 (0.03,0.08)   0.04 (0.02,0.07)  < 0.001
   1   0.79 (0.52,1.21)   0.75 (0.49,1.16)  0.2
   >2  1.96 (1.28,2.99)   1.82 (1.19,2.8)   0.006

saltegg: 2.37 (1.41,3.99) 0.92 (0.48,1.76)  0.807
  0.808
   Yes vs No
sex:     1.58 (1.19,2.08) 1.85 (1.35,2.53)  < 0.001       <
  0.001
   Male vs Female
```

```
Log-likelihood =  -509.5181
No. of observations =  972
AIC value = 1031.0
```

Variabel penjelas ketiga 'sex' merupakan faktor resiko independent lainnya. Karena wanita merupakan reference level, maka laki-laki mengalami peningkatan odd 90% dibandingkan dengan wanita. Variabel ini bukan merupakan pembaur untuk variabel sebelumnya karena odds ratio telah dirubah secara substansial (dari 'glm4'). Alasan untuk tidak membaur adalah kurangnya asosiasi dengan variabel penjelas sebelumnya. Dengan kata lain, laki-laki dan wanita tidak berbeda dalam hal konsumsi kue sus dan telur asin.

## Interaksi

Bentuk interaksi terdiri paling kurang dua variabel, salah satunya harus merupakan variabel kategori. Jika terjadi interaksi, pengaruh suatu variabel akan bergantung pada status variabel lainnya dan dengan demikian mereka tidak independen. Bentuk interaksi dalam **R** dapat dikelompokkan dalam dua cara: 'x1*x2' atau 'x1:x2'. Bentuk tersebut ekuivalen dengan 'x1+ x2+x1:x2'.

Berlatih dengan model berikut ini dimana variabel 'eclairgr' dan 'beefcurry' dikelompokkan sebagai bentuk interaksi.

```
> glm6 <- glm(case ~ eclairgr*beefcurry, family=binomial)
> logistic.display(glm6, decimal=1)

Logistic regression predicting diseased

           crude OR(95%CI) adj. OR(95%CI) P(Wald's test)
  P(LR-test)
eclairgr: ref.=2                                        <
  0.001
   0       0 (0,0.1)       0.1 (0,0.5)    0
   1       0.8 (0.5,1.2)   0.5 (0.1,2.5)  0.39
   >2      2 (1.3,3)       0.5 (0.1,3)    0.41

beefcurry: 2.7 (1.6,4.6)   1.4 (0.5,3.6)  0.53          <
  0.001
   (Yes vs No)
```

```
eclairgr:beefcurry: ref.=2:No
  0.03
   0:Yes    -              0.3 (0.1,1.2)   0.09
   1:Yes    -              1.7 (0.3,9.7)   0.52
   >2:Yes   -              4.8 (0.7,33.9)  0.11

Log-likelihood =  -511.8
No. of observations =  972
AIC value = 1039.6
```

Pada bentuk terakhir , 'eclairgr:beefcurry', merupakan bentuk interaksi. Interpretasi P value yang diperoleh dari uji Wald's menunjukkan bahwa interaksi mungkin tidak signifikan. Bagaimanapun, P value dari uji LR lebih penting dan menentukan. Nilai 0.03 mengindikasikan bahwa 'eclairgr' dan 'beefcurry' keduanya saling dependen satu sama lain. Odd rasio untuk bentuk interaksi tidak dapat diaplikasikan.

Pembaca dapat melakukan relevel variabel menjadi original reference level (ref = "0") dan membandingkan keluarannya.

```
> eclairgr <- relevel(eclairgr, ref="0")
> pack()
> glm7 <- glm(case~eclairgr*beefcurry, family=binomial,
  data=.data)
> logistic.display(glm7)
```

### *Seleksi Stepwise variabel independen*

Bahasan berikutbya menunjukkan seleksi model stepwise dalam **R**.

Pertama, dibuat subset dataset untuk memastikan bahwa semua variabel memiliki catatn valid (non missing). Ingat bahwa perintah glm juga mengizinkan pengelompokan a subset of the dataset. Model selanjutnya menggunakan variabel 'eclair.eat' daripada 'eclairgr' untuk menyederhanakan output.

```
> complete.data <- subset(.data, subset=!is.na(eclair.eat)
  & !is.na(beefcurry) & !is.na(saltegg) & !is.na(sex))

> glm8 <- glm(case ~ eclair.eat * beefcurry + saltegg + sex,
  family = binomial, data=complete.data)
Model mungkin terlalu berlebihan. Kita biarkan R memilih
  model dengan AIC terendah.
> modelstep <- step(glm8, direction = "both")
```

```
Start:  AIC= 1038.5
 case ~ eclair.eat * beefcurry + saltegg + sex
                       Df Deviance   AIC
- saltegg                1     1026  1036
<none>                           1026  1038
- eclair.eat:beefcurry   1     1030  1040
- sex                    1     1039  1049

Step:  AIC= 1036.5
 case ~ eclair.eat + beefcurry + sex + eclair.eat:beefcurry
                       Df Deviance   AIC
<none>                           1026  1036
- eclair.eat:beefcurry   1     1030  1038
+ saltegg                1     1026  1038
- sex                    1     1039  1047
```

Pada awalnya AIC adalah 1038.5. Perintah step memindahkan setiap variabel independent dan membandingkan penurunan derajat kebebasan, deviasi dan AIC terbaru. Hasil meningkat bedasarkan AIC. Urutan paling atas memiliki AIC terendah dan merupakan yang terbaik. Pada langkah pertama, pemindahan 'saltegg' akan menghasilkan AIC minimum dan akan digunakan untuk langkah selanjutnya.

Pada tahap kedua tidak dilakukan pemindahan terhadap variabel independen yang tersisa dan diperoleh AIC terendah. Maka proses seleksi berhenti dengan variabel yang tersisa. Sekarang kita periksa hasilnya.

```
> summary(modelstep)
=================== Lines omitted ==================
Coefficients:
                      Estimate  St. Error z value Pr(>|z|)
(Intercept)             -2.672      0.494   -5.41  6.3e-08
eclair.eatTRUE           2.067      0.601    3.44  0.00059
beefcurry               -0.903      0.573   -1.58  0.11484
sexMale                  0.586      0.163    3.59  0.00033
eclair.eatTRUE:beefcurry 1.412      0.685    2.06  0.03923
=================== Lines omitted ==================
```

Model terakhir dengan variabel 'saltegg' dikeluarkan. Jenis kelamin menjadi faktor resiko independent. Konsumsi kue sus merupakan faktor resiko yang ditambah dengan konsumsi kari daging. Memakan kari dengan sendirinya menjadi faktor pelindung. Bagaimanapun, ketika dimakan dengan kue sus, odd rasio meningkat dan menjadi positif.

Harus diingat bahwa regresi stepwise terbatas terhadap ekspolari dan sering tidak sesuai untuk pengujian hipotesis tertentu, dimana kebanyakan studi epidemiologi dirancang seperti itu. Regresi ini cenderung menghilangkan semua variabel independent yang tidak signifikan dalam model. Pada pengujian hipotesis, satu atau beberapa variabel bebas digunakan untuk pengujian. Odd rasio dan selang kepercayaannya harus tetap dihitung sesuai signifikansi statistika.

## Interpretasi odds ratio

Perhatikan model terakhir yang diperoleh dengan seksama.

```
> logistic.display(modelstep, crude=FALSE)

Logistic regression predicting case

                             adj. OR(95%CI)    P(Wald's)
  LR-test
eclair.eat                   7.9 (2.43,25.66)  < 0.001     <
  0.001
beefcurry: Yes vs No         0.4 (0.13,1.25)   0.115       <
  0.001
sex: Male vs Female          1.8 (1.31,2.47)   < 0.001     <
  0.001
eclair.eatTRUE:beefcurryYes  4.1 (1.07,15.71)  0.039
  0.048

Log-likelihood =  -513.2296
No. of observations =  972
AIC value = 1036.5
```

All the three variables 'eclair.eat', 'beefcurry' and 'sex' are dichotomous. The odds ratio for 'sex' is that of males compared to females. For 'eclair.eat' it is TRUE vs FALSE and for 'beefcurry', "Yes" is compared to "No".

The independent variable 'sex' has an odds ratio of approximately 1.8, which means that males have approximately a 1.8 times higher risk than females. The other two variables, 'eclair.eat' and beefcurry, are interacting. The odds ratio of 'eclair.eat' depends on the value of 'beefcurry' and vice versa. Three terms 'eclair.eat', 'beefcurry' and their interaction term 'eclair.eat:beefcurry' need to

be considered simultaneously.

If 'beefcurry' is "No" (did not eat beef curry), the 'eclair.eat:beefcurry' term is 0. The odds ratio for eclair.eat for this subgroup is therefore only 7.9. Among the beef curry eaters, the interaction term should be multiplied by 1 (since 'eclair.eat' and 'beefcurry' are both 1), the odds ratio is then 7.9 × 4.1 or approximately 32.4.

The required odds ratio can be obtained from computing the product of the appropriate odds ratio of the individual variables. However, the standard errors and 95% confidence interval cannot be easily computed from the above result.

A better way to get the odds ratio and 95% confidence interval for 'eclair.eat' among 'beefcurry' eaters is to relevel the variable and run the model again.

```
> complete.data$beefcurry <-
  relevel(complete.data$beefcurry, ref="Yes")
> glm9 <- glm(case ~ eclair.eat * beefcurry + sex,
  family = binomial, data = complete.data)

> logistic.display(glm9, crude=FALSE)

Logistic regression predicting case

                        adj. OR (95%CI)   P(Wald's test)
  P(LR-test)
eclair.eat              32.4 (16.9,62.3)  < 0.001         <
  0.001
beefcurry: No vs Yes    2.47 (0.8,7.59)     0.115         <
  0.001
sex: Male vs Female     1.8  (1.31,2.47)  < 0.001         <
  0.001
eclair.eatTRUE:         0.24 (0.06,0.93)    0.039
  0.048
    beefcurryNo

Log-likelihood =  -513.2296
No. of observations =  972
AIC value = 1036.5
```

Odd rasio dan interval kepercayaan 95% 'eclair.eat' diantara mereka yang memakan kari daging sekarang berada pada baris pertama karena bentuk 'beefcurry' pada baris kedua dan bentuk interaksi pada baris terakhir sama

sama bernilai 0.

### *Format data lainnya*

Himpunan data diatas berdasarkan catatan individual. Terkadang regresi diharuskan menampilkan tabel jumlah.

```
> zap()
> data(ANCtable)
> ANCtable
> use(ANCtable)
> death  <- factor(death)
> levels(death) <- c("no","yes")
> anc <- factor(anc)
> levels(anc) <- c("old","new")
> clinic <- factor(clinic)
> levels(clinic) <- c("A","B")
> pack()
```

Fungsi Epicalc *pack* mengidentifikasi semua vektor bebas dengan panjang yang sama seperti jumlaha catatan yang ada dalam **.data** dan menambahkan kedalam data.frame. vektor bebas ini kemudian dihilangkan dari lingkungan global.

```
> .data
  death anc clinic Freq
1    no old      A  176
2   yes old      A   12
3    no new      A  293
4   yes new      A   16
5    no old      B  197
6   yes old      B   34
7    no new      B   23
8   yes new      B    4
```

Ini merupakan format dengan 'Freq' menjadi variabel yang menotasikan jumlah sampel untuk setiap kategori. Variabel ini menambahkan argumen 'weight' dalam model.

```
> glm(death ~ anc+clinic, binomial, weight=Freq, data=.data)
```

Koefisiennya sama seperti yang digunakan dalam himpunan data yang asli, **ANCdata** tetapi derajat kebebasannya berbeda.

Format data lainnya untuk regresi logistik mungkin digunakan dimana sejumlah

kasus dan kontrol dari sebaran yang sama berada dalam baris yang sama tetapi kolom yang berbeda.

```
> .data$condition <- c(1,1,2,2,3,3,4,4)
> data2 <- reshape(.data, timevar="death", v.name="Freq",
  idvar="condition", direction="wide")
```

Variabel 'condition' dibuat sebagai fasilitas reshaping. Data yang direshape, data2 hanya memiliki empat baris dari data jika dibandingkan dengan **.data** yang memiliki 8 baris.

```
> data2
  anc clinic condition Freq.no Freq.yes
1 old      A         1     176       12
3 new      A         2     293       16
5 old      B         3     197       34
7 new      B         4      23        4
```

Kolom pertama untuk setiap baris adalah 'row.names' dari data frame. Data frame ini dapat dituliskan menjadi text file dengan 'row.names' dan variabel 'condition' (variabel ketiga) diabaikan.

Regresi logistic untuk 'data2' dapat diperoleh dengan cara:

```
> glm(cbind(Freq.yes, Freq.no) ~ anc + clinic, data=data2,
   family=binomial)
```

Pada ruas kiri formula diatas merupaka hasil penggabungan dua kolom frekuensi keluaran. Bagian lain perintah menyisakan format yang sama untuk case-by-case. Koefisien dan standard error dari perintah ini sama dengan yang diatas. Bagaimanapun, penyimpangan residual dan AIC sangat kecil berdasarkan jumlah derajat kebebasan yang rendah.

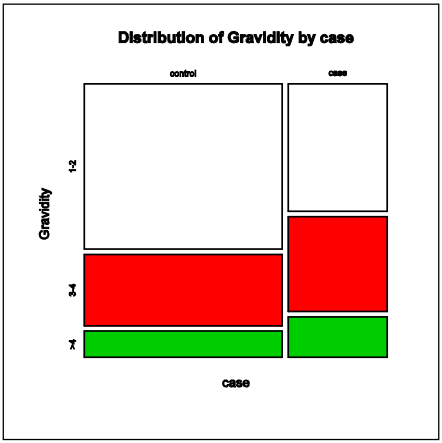
Format data case-by-case umumnya sesuai dengan analisis data actual. Format dalam **ANCtable** dan 'data2', yang terkadang ditemukan merupakan perhatian utama secara teori.

### *Lebih dari 2 strata*

Dataset **Ectopic** merupakan data yang diperoleh dari studi case-control yang menguji hipotesis apakah previous induced abortion merupakan faktor resiko untuk current ectopic pregnancy. Ada tiga kelompok yang dipelajari: ectopic pregnancy patients ('EP'), current clients who came for an induced abortion

('IA') and those who came for delivery ('deli'). Untuk memudahkan, pada bahasan ini, dua grup terakhir dikombinasikan dan dimasukkan sebagai kontrol sementara grup pertama dianggap sebagai kasus. Sebaran utama adalah 'hia' atau sejarah previous induced abortion dan pembaur potensial adalah 'gravi' atau level gravidity. Cobalah perintah **R** berikut:

```
> zap()
> data(Ectopic)
> use(Ectopic)
> des()

  No. of observations = 723
  Variable     Class       Description
1 id           integer
2 outc         factor      Outcome
3 hia          factor      Previous induced abortion
4 gravi        factor      Gravidity

> summ()

No. of observations = 723

  Var. name Obs.   mean   median  s.d.   min.   max.
1 id        723    362    362     208.86 1      723
2 outc      723    2      2       0.817  1      3
3 hia       723    1.545  2       0.498  1      2
4 gravi     723    1.537  1       0.696  1      3

> tab1(outc, graph=F)
> tab1(hia, graph=F)
> tab1(gravi, graph=F)
> case <- outc == "EP"
> case <- factor(case)
> levels(case) <- c("control","case")
> tabpct(case, gravi)
```

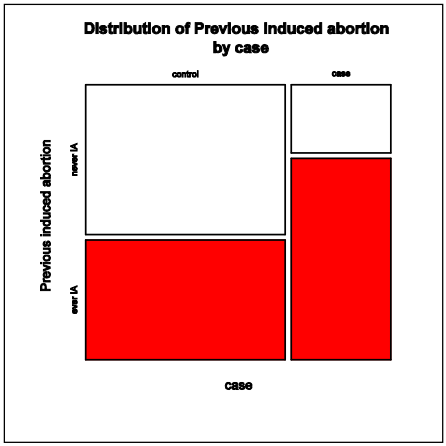



Kasus tersebut memiliki level of gravidity yang tinggi.

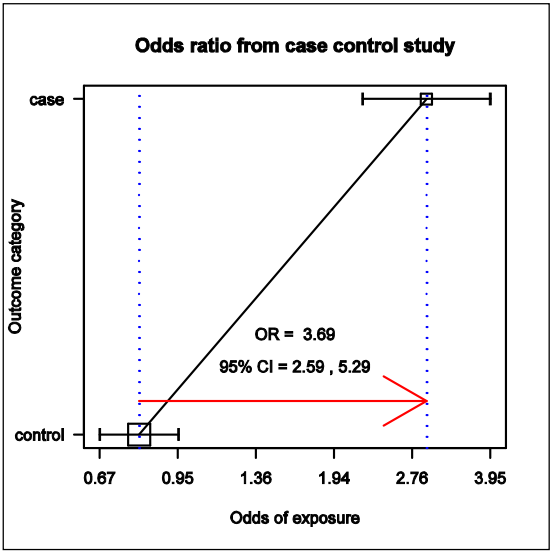
```
> tabpct(case, hia) -> case.hia
```

Perintah diatas hanay akan menampilkan grafik saja, karena kita telah menyimpan keluaran menjadi objek. Inspeksi objek ini dapat dilakukan dengan mengetik nama objek.

Kasus tersebut juga memiliki experience of induced abortion yang besar.



```
> cc(case, hia, design = "case-control")

         hia
case       no yes Total
  control 268 214   482
  case     61 180   241
  Total   329 394   723

OR =  3.689
95% CI = 2.595 5.291
Chi-squared = 59.446 ,  1 d.f. , P value = 0
Fisher's exact test (2-sided) P value = 0
```



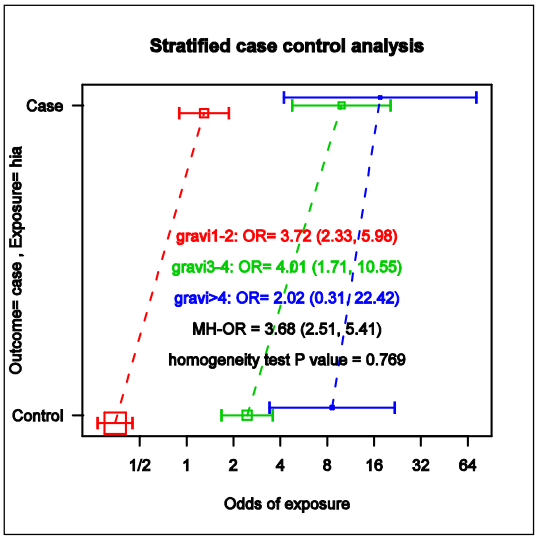

Grafik ini diperjelas dengan 'design' = "case-control", meskipun orientasinya disesuaikan dengan variabel penjelas. Sebaran odd diantara kasus ditunjukkan sebelah kanan (nilai tertinggi).

Next we adjust for gravidity.

```
> mhor(case, hia, gravi, design="case-control")

Stratified analysis by  gravi
               OR lower lim. upper lim.  P value
gravi 1-2    3.72      2.328       5.98 6.26e-09
```

```
gravi 3-4     4.01       1.714       10.55 3.52e-04
gravi >4      2.02       0.307       22.42 4.62e-01
M-H combined 3.68       2.509        5.41 6.12e-12

M-H Chi2(1) = 47.29 , P value = 0
Homogeneity test, chi-squared 2 d.f. = 0.52 , P value =
  0.769
```

The stratified analysis menunjukkan 3 strata gravidity dan menghubungkan 3 garis sebaran dalam grafik. Sebaran odd untuk induced abortion meningkat (bergerak maju keruas kanan) seiring gravidity. Odd diantara kelompok kontrol lebih rendah (lebih kekiri) untuk setiap stratum kelompok gravidity. Kemiringan 3 garis sama sama sama mengindikasikan interaksi yang rendah dan hal ini didukung dengan P value dari uji homogenitas. MH kombinasi odd rasio sama dengan crude odds ratio, hal ini menunjukkan sedikit pengaruh pembauran oleh gravidity.




Untuk regresi logitik dapat digunakan fungsi glm seperti sebelumnya.

```
> glm1 <- glm(case ~ hia, family = binomial)
```

Serupa dengan bahasan sebelumnya, *logistic.display* dapat digunakan untuk memperoleh sebaran odd rasio dan selang kepercayaan 95% untuk induced

abortion.

```
> logistic.display(glm1)

Logistic regression predicting case : case vs control

                   OR(95%CI)       P(Wald's test) P(LR-test)
hia:               3.7 (2.63,5.2)  < 0.001        < 0.001
  ever IA vs never IA

Log-likelihood =  -429.3863
No. of observations =  723
AIC value = 862.77

> glm2 <- glm(case ~ hia + gravi, binomial)
> logistic.display(glm2)

Logistic regression predicting case : case vs control

          crude OR(95%CI)  adj. OR(95%CI)  P(Wald's test)
  P(LR-test)
hia:      3.7 (2.6,5.2)    3.7 (2.5,5.4)   < 0.001         <
  0.001
  ever IA vs never IA

gravi: ref.=1-2                                            1
  3-4     1.7 (1.2,2.4)    1.0 (0.7,1.5)   0.989
  >4      2.0 (1.2,3.2)    1.0 (0.6,1.7)   0.992

Log-likelihood =  -429.3861
No. of observations =  723
AIC value = 866.77
```

AIC dari model 'glm1' lebih rendah dari 'glm2', ini mengindikasikan model yang lebih layak. Kasus ectopic pregnancies hampir mendekati 3.7 kali odd sebaran sebelumnya untuk induced abortion dibandingkan dengan kelompok kontrol. Gravidity tidak memiliki penagruh terhadap keluaran dan bukan merupakan pembaur.

## Referensi

Hosmer Jr DW & Lemeshow S (2004). Applied Logistic Regression, 2nd Edition.
Kleinbaum DG & Klein M. Logistic Regression. A self-learning Text (2nd Edition). Springer-Verlag New York, Inc. August 2002.

## Latihan

Soal 1.

Dengan menggunakan data frame 'complete.data', hitunglah nilai odds ratio dan 95% selang kepercayaan untuk for paparan kombinasi antara 'eclair.eat' dan 'beefcurry' menggunakan kelompok yang terekspose to neither eclair maupun beef curry sebagai group rujukannya.

Soal 2.

Gunakanlah **ANCtable** dataset dan fungsi xtabs untuk membuat table 2x2 bertingkat. Lalu gunakan pula fungsi *mhor* untuk menganalisa *adjusted odds ratio.*

Hint: 'help(xtabs)', 'help(mhor)'.

Soal 3.

Gunakan **Hakimi** dataset untuk melakukan analisa yang sama dengan soal no.2

Soal 4.

Pada **Ectopic** dataset, lakukan unclass 'gravi' dan gunakan regresi logistik untuk memeriksa hubungan *dose response* (trend linier) antara graviditas dan risiko kehamilan ektopik, setelah disesuaikan dengan dampak dari aborsi sebelumnya ('hia').

BAB 16

# Studi Kasus Kontrol Berpasangan

Contoh pada bab sebelumnya memiliki kasus dan control yang diperoleh secara terpisah. Dalam studi kasus control berpasangan, ketika sebuah kasus diambil, sebuah kontol atau himpunan control (lebih dai satu orang) dapat diambil untuk dipasangkan dengan kasus dalam beberapa parameter seperti umur dan jenis kelamin dan kondisi lainnya (seperti saudara atau tetangga). Jika deret control dipilih berdasarkan umur dan jenis kelamin yang sesuai - hal ini bertujuan untuk menghindari ketidakseimbangan - maka himpunan data seharusnya dianalisis dalam keadaan tidak berpasangan. Banyak buku yang cukup bagus untuk mengetahui bagaimana menganalisis studi kasus control, khususnya dalam pengaturan berpasangan, dan pembaca dapat membaca referensi pada akhir bab ini.

Contoh-contoh pada bab ini ditampilkan hanya untuk demonstrasi semata. Ukuran sampel nya cukup kecil untuk menghasilkan kesimpulan yang solid. Bagaimanapun juga metode ini tetap dapat diaplikasikan untuk studi kasus control berpasangan lainnya.

Dalam analsis himpunan berpasangan, perbandingan dibuat dalam setiap himpunan berpasangan dimana suatu deret dipasangkan dengan deret lainnya. Pada bab ini, himpunan data **VC1to1** and **VC1to6** terdiri dari data yang diperoleh dari pengujian studi kasus control berpasangan dimana merokok, minum alcohol dan berkerja pada industry karet merupakan faktor resiko pada kanker oesophageal. Setiap kasus dipasangkan dengan tetangga dari grup usia dan jenis kelamin yang sama. Rasio kesesuaian beragam mulai dari 1:1 hingga 1:6. File **VC1to6** merupakan dataset yang penuh dimana **VC1to1** memiliki jumlah kontrol per kasus yang direduksi menjadi 1 untuk semua himpunan berpasangan. File terakhir akan digunakan terlebih dahulu untuk analisis ini.

```
> zap()
> data(VC1to1)
> use(VC1to1)
> des()

No. of observations = 52
  Variable      Class           Description
1 matset        numeric
2 case          numeric
3 smoking       numeric
4 rubber        numeric
5 alcohol       numeric

> summ()
No. of observations = 52

  Var. name obs.   mean   median  s.d.   min.   max.
1 matset    52     13.5   13.5    7.57   1      26
2 case      52     0.5    0.5     0.5    0      1
3 smoking   52     0.81   1       0.4    0      1
4 rubber    52     0.33   0       0.47   0      1
5 alcohol   52     0.52   1       0.5    0      1

> head(.data)
  matset case smoking rubber alcohol
1      1    1       1      0       0
2      1    0       1      0       0
3      2    1       1      0       1
4      2    0       1      1       0
5      3    1       1      1       0
6      3    0       1      1       0
```

Terdapat 26 pasang padanan seperti yang ditunjukkan dalam variabel terurut 'matset'. Kode variabel 'case' adalah 1 untuk orang yang sakit dan 0 untuk tidak sakit. Sekarang kita membentuk kembali data untuk memfasilitasi eksplorasi data.

```
> wide <- reshape(.data, timevar="case",
  v.names=c("smoking",
   "rubber", "alcohol"), idvar="matset", direction="wide")

> head(wide,3)
matset smoking.1 rubber.1 alcohol.1 smoking.0 rubber.0
  alcohol.0
1      1         1        0         0         1        0
3      2         1        0         1         1        1
5      3         1        1         0         1        1
```

Data frame original **.data** memiliki variabel yang disusun dalam bentuk *long*. Setiap record menyajikan satu subjek/sampel. Data frame yang baru 'wide' dalam bentuk yang luas. Setiap record menyajikan pasangan yang bersesuaian. Tabulasi silang kebiasaan merokok dari kasus dan kontrol dalam setiap pasangan dapat dengan mudah dilakukan.

```
> attach(wide)
> table(smoking.1, smoking.0, dnn=c("smoking in case",
    "smoking in control"))
              smoking in control
smoking in case  0  1
              0  0  5
              1  5 16
```

Argumen optional 'dnn' dalam perintah table diatas membolehkan nama dimensi menjadi lebih spesifik, memudahkan interpretasi. Dari tabulasi silang ini, tidak terdapat pasangan padanan dimana kedua kasus dan kontrol merupakan bukan perokok. Terdapat 16 pasang padanan dimana kasus dan kontrol nya adalah perokok. Ada 5 pasangan dimana kasusnya merupakan perokok tetapi kontrolnya tidak (sudut kiri bawah). Dan lima pasang terakhir (sudut kanan atas), kontrol nya merupakan perokok sementara kasusnya tidak.

The level of contrast of history of smoking between the two based on matched pairs is called a conditional odds ratio. Itu merupakan nilai sel pada sudut kiri bawah yang dibagi oleh sel sudut kanan atas. Pada kasus ini odd rasio bersyarat adalah 5/5 = 1 (terkadang disebut juga odd rasio McNemar). Pada kenyataannya, ini berarti bahwa rasio jumlah tidak seimbang antara paparan

kasus versus paparan kontrol adalah 1.

*Epicalc* mempunyai fungsi *matchTab* yang dapat digunakan untuk menganalisis himpunan berpasangan ( tidak butuh 1 kasus per 1 kontrol) dari dataset original seperti berikut:

```
> detach(wide)
> matchTab(case, smoking, strata=matset)

Number of controls = 1
                    No. of controls exposed
No. of cases exposed 0  1
                     0 0  5
                     1 5 16

Odds ratio by Mantel-Haenszel method = 1

Odds ratio by maximum likelihood estimate (MLE) method = 1
 95%CI= 0.29 , 3.454
```

Dua metode memberikan nilai yang sama untuk odd rasio. Metode MLE juga menghasilkan dugaan interval kepercayaaan 95%.

## Pemadanan 1:n

Jika terdapat masalah serius dalam kekurangan kasus penyakit, rasio terbaik pemadanan adalah satu kasus per kontrol. Sumber yang mengkoleksi data dari setiap individu akan sangat lebih efisien tanpa memperhatikan apakah sampel/subjek adalah sebuah kasus atau kontrol. Bagaimanapun juga saat penyakit yang jadi perhatian menjadi langka, seringkali cost-effective untuk meningkatkan jumlah kontrol setiap kasus. Efesiensi (khususnya sumber yang mengoleksi data dari kontrol tambahan) menurun tetapi hal ini berarti bahwa kajian akan segera selesai.

Sekarang kita akan menganalisis full dataset, dimana setiap kasus kemungkinan memiliki 1 dan 6 kontrol berpadanan.

```
> zap()
> data(VC1to6); use(VC1to6)
> des()
```

```
> summ()

No. of observations = 119

================= lines omitted ============
> .data
    matset case smoking rubber alcohol
1        1    1       1      0       0
2        1    0       1      0       0
3        2    1       1      0       1
4        2    0       1      1       0
================= lines omitted ============
116     26    0       0      0       0
117     26    0       1      1       0
118     26    0       0      0       0
119     26    0       1      1       1
Akan sangat tidak mudah untuk membentuk kembali data ini
  menjadi bentuk yang luas. Mari gunakan fungsi Epicalc
  matchTab.
> matchTab(case, smoking, strata=matset)

Number of controls = 1
                     No. of controls exposed
No. of cases exposed 0 1
                    0 0 0
                    1 0 3

Number of controls = 2
                     No. of controls exposed
No. of cases exposed 0 1 2
                    0 0 0 1
                    1 1 1 0
================= lines omitted ============
Number of controls = 6
                     No. of controls exposed
No. of cases exposed 0 1 2 3 4 5 6
                    0 0 0 0 1 0 0 0
                    1 0 0 0 0 0 1 2

Odds ratio by Mantel-Haenszel method = 1.988

Odds ratio by maximum likelihood estimate (MLE) method =
  2.066
 95%CI= 0.678 , 6.299
```

Perintah diatas menghasilkan enam table berdasarkan himpunan padanan dari ukuran yang sama (kasus per kontrol). Tabel terakhir menampilkan 4 himpunan padanan dengan enam kontrol setiap kasus. Salah satunya memiliki kasus non-paparan dan tiga keluar dari paparan kontrol. Satunya memiliki paparan kasus san lima dari enam kontrol non-paparan. Dua himpunan terakhir memiliki kasus dan keenam kontrolnya terpapar. Odd rasio dari dua dataset yang berlainan tidak terlalu berbeda. Bagaimanapun pengaruh merokok berdasarkan outcome tetap tidak signifikan secara statistik sepanjang selang kepercayaan 95% dari odd rasio masih mengandung nilai 1.

## Regresi Logistik untuk pemadanan 1:1

Seperti yang dibahas diatas, odd rasio bersyarat 1:1 untuk studi kasus control berpasangan adalah berdasarkan rasio ketidakseimbangan paparan antara kasus dan kontrol dari pasangan padanan yang sama. Dari pandangan pemodelan, perbedaan hasil outcome dalam himpunan berpasangan adalah bergantung pada perbedaan paparan antara kasus dan kontrol. Perbedaan yang terlebih dahulu bernilai 1 karena outcome kasus adalah samadengan 1 dan outcome kontrol adalah samavdengan 0. Perbedaan paparan dihitung dalam himpunan berpasangan.

```
> zap()
> data(VC1to1);  use(VC1to1)
> use(wide)
> smoke.diff <- smoking.1 - smoking.0
> alcohol.diff <- alcohol.1 - alcohol.0
> rubber.diff <- rubber.1 - rubber.0
> outcome.1 <- rep(1, 26) # 26 cases with outcome being 1
> outcome.0 <- rep(0, 26) # 26 controls with outcome being 0
> outcome.diff <- outcome.1 - outcome.0
> cbind(outcome.diff, smoke.diff, alcohol.diff)
> pack()
> summ()
No. of observations = 26

   Var. name    obs. mean   median  s.d.   min.   max.
1  matset       26   13.5   13.5    7.65   1      26
2  smoking.1    26   0.81   1       0.4    0      1
```

```
3  rubber.1     26   0.31   0       0.47   0      1
4  alcohol.1    26   0.65   1       0.49   0      1
5  smoking.0    26   0.81   1       0.4    0      1
6  rubber.0     26   0.35   0       0.49   0      1
7  alcohol.0    26   0.38   0       0.5    0      1
8  alcohol.diff 26   0.27   0       0.6    -1     1
9  outcome.0    26   0      0       0      0      0
10 outcome.1    26   1      1       0      1      1
11 outcome.diff 26   1      1       0      1      1
12 rubber.diff  26   -0.04  0       0.6    -1     1
13 smoking.diff 26   0      0       0.63   -1     1
```

Ingat bahwa variabel 'outcome.diff' adalah 1 untuk keseluruhan record karena outcome untuk kasus sama dengan 1 dan untuk kontrol adalah 0 sedangkan perbedaan dalam paparan terhadap alkohol, karet dan rokok dapat menjadi 1 (kasus exposed dan kontrol tidak exposed), 0 (kedua kontrol dan kasus exposed atau keduanya tidak exposed) dan -1 (kasus tidak exposed tetapi kontrol exposed).

Sekarang kita akan menampilkan bagaimana regresi logistik memperkirakan perbedaan outcome dari kebiasaan merokok yang berbeda.

```
> co.lr1 <- glm(outcome.diff ~ smoke.diff-1, binomial)
> summary(co.lr1)

Call:
glm(formula = outcome.diff ~ smoke.diff-1, family=binomial)
Coefficients:
           Estimate Std. Error z value Pr(>|z|)
smoke.diff    0.000      0.632       0        1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 36.044  on 26  degrees of freedom
Residual deviance: 36.044  on 25  degrees of freedom
AIC: 38.04
```

Dalam model glm diatas, perbedaan outcome ( yang selalu bernilai 1 untuk alasan diatas) diprediksi dengan pebedaan kebiasaan merokok. Terdapat bentuk tambahan '-1' pada bagian sebelah kanan formula yang mengindikasikan bahwa intercept seharusnya dihilangkan dari model. Umumnya, intercept merupakan nilai dugaan variabel terikat (variabel pada bagian kiri formula) saat semua variabel bebasnya bernilai 0. Pada regresi logistik bersyarat tidak

terdapat intercept yang disebabkan perbedaan outcome fixed to 1, dan logit bernilai 0.

Dengan koefisien 0, odd rasio exp(0) = 1, yang bernilai sama dengan hasil yang diperoleh dari tabulasi berpasangan. Interval kepercayaan 95% dari odd rasio dapat diperoleh dari:

```
> exp(confint.default(co.lr1))
             2.5 % 97.5 %
smoke.diff 0.2895 3.4542
```

Nilai ini jelas serupa dengan yang diperoleh melalui tabulasi berpadanan. Epicalc dapat menampilkan hasil dalam bentuk yang lebih tepat.

```
> logistic.display(co.lr1)
Logistic regression predicting outcome.diff

                  OR(95%CI)      P(Wald's test) P(LR-test)
smoke.diff        1 (0.29,3.45)  1              -

Log-likelihood = -18.0218
No. of observations = 26
AIC value = 38.0437
```

Ingat kemba;I bahwa kelebihan regresi logistic adalah kemampuannya untuk mengatasi lebih dari satu variabel terikat. Jalankan model regresi kembali dan tambahkan bentuk alkohol.

```
> co.lr2 <- glm(outcome.diff ~ smoke.diff + alcohol.diff-1,
  binomial)
> logistic.display(co.lr2, decimal=1)

Logistic regression predicting outcome.diff

               crude OR(95%CI) adj.OR(95%CI)   P(Wald's)
  P(LR-test)
smoke.diff     1 (0.3,3.5)     0.7 (0.2,2.9)  0.66       0.66
alcohol.diff   4.5 (1,20.8)    4.8 (1,23.2)   0.05       0.03

Log-likelihood = -15.513
No. of observations = 26
AIC value = 35.026
```

Pengenalan 'alcohol.diff' telah mengubah koefisien 'smoke.diff' secara substansial mengindikasikan bahwa merokok dirancukan (confounded) oleh konsumsi alkohol.

## Regresi logistik bersyarat

Analisis regresi logistic diatas yang didasarkan pada manipulasi data, tetap tidak mudah untuk dilakukan. Analis statistik perlu membentuk kembali dataset dan membuat perbedaan nilai dalam variabel terikat dan variabel bebas. Lebih lanjut metode ini hanya dapat diaplikasikan untuk padanan 1:1.

Metode yang lebih sederhana analisis multivariate dari dataset **VC1to1** adalah menggunakan perintah clogit (kependekan dari 'conditional logit') dari paket **survival**. Dataset orginal dalm format yang panjang dapat digunakan.

```
> zap()
> library(survival)
> use(.data)
> clogit1 <- clogit(case ~ smoking+alcohol+strata(matset))
> summary(clogit1)

  n= 52
         coef exp(coef) se(coef)      z    p
smoking -0.314      0.73    0.708 -0.444 0.66
alcohol  1.572      4.81    0.803  1.957 0.05

        exp(coef) exp(-coef) lower .95 upper .95
smoking      0.73      1.369     0.182      2.92
alcohol      4.81      0.208     0.998     23.23
Rsquare= 0.092   (max possible= 0.5 )
Likelihood ratio test= 5.02  on 2 df,   p=0.0814
Wald test            = 3.83  on 2 df,   p=0.147
Score (logrank) test = 4.62  on 2 df,   p=0.0991
```

Bagian diatas dari outcome menjelaskan bahwa perintah clogit memanggil perintah umum lainnya coxph. Jika perintah panggilan digunakan, hasilnya akan sama saja.

```
> coxph(formula = Surv(rep(1, 52), case) ~ smoking + alcohol
  + strata(matset), method = "exact")
```

Odd rasio dan interval kepercayaan 95% dari clogit sama dengan yang diperoleh dengan pemodelan perbedaan. Bagian terakhir mengandung beberapa hasil pengujian, masing-masing mengindikasikan bahwa model tidak berbeda signifikan dengan model null (model yang tidak mengandung variabel prediktor apapun).

Fungsi Epicalc *clogistic.display* dapat digunakan untuk mendapatkan outpur yang lebih baik.

```
> clogistic.display(clogit1)
```

```
Conditional logistic regression predicting case : 1 vs 0

                 crude OR(95%CI)   adj. OR(95%CI)    P(Wald)
  P(LR)
smoking: 1 vs 0  1.0 (0.29, 3.45)  0.73 (0.18,2.92)  0.66
  0.655
alcohol: 1 vs 0  4.5 (0.97,20.83)  4.81 (1,23.23)    0.05
  0.025

No. of observations =  52
```

## Referensi

Breslow NE, Day NE (1980). The Analysis of Case-Control Studies (Statistical Methods in Cancer Research, Vol. 1). Int Agency for Research on Cancer.

## Latihan

Soal1.
Lakukan pemadanan tabulasi untuk paparan alkohol dalam **VC1to6**Bandingkan hasilnya dengan paparan alkohol tersebut jika digunakan regresi logistik bersyarat.
Soal2..
Mengacu pada log likelihood dan nilai AIC pada bab sebelumnya tentang model linier umum.(generalized linear model).
Model regresi logistik bersyarat tidaklah memberikan informasi tentang log likelihood dan nilai AIC, akan tetapi model ini member informasi tentang log likelihood bersyarat, yang juga menunjukkan tingkat kesesuaian model. Log likelihood bersyarat ini dapat digunakan untuk perbandingan model bersarang dari data set yang sama.

```
> clogit3 <- clogit(case ~ smoking + alcohol +rubber + strata(matset))
> attributes(clogit3)
> clogit3$loglik
[1] -37.89489 -31.89398
```

Elemen 'loglik' dari masing-masing perintah clogit (analog dengan 'logLik' pada glm) terdiri atas dua two sub-elemen. Sub-elemen yang pertama, Likelihood bersyarat sebagai *null model*, yang sama untuk setiap model Log likelihood bersyarat .The second sub-element is specific to the particular model. Yang kedua, perbedaaan mutlak antara dua sub-elemen tersebut sama dengan uji perbandingan likelihood untuk model. Hasil dari uji ini dapat dilihat dari display pada model.

Cobalah model-model yang berbeda dan bandingkan nilai dari model regresi logistik bersyaratnya. Pilihlah model yang terbaik.

# BAB 17

# Regresi Logistik Polytomous

Regresi logistik sudah sangat dikenal untuk pemodelan keluaran yang biner. Dalam beberapa keadaan, keluarannya dapat memiliki lebih dari dua kategori yang tak beraturan.

Dalam bab 15 kita melihat dataset **`Ectopic`**, yang berasal dari studi pengujian hipotesis apakah diinduksi aborsi sebelumnya merupakan faktor risiko untuk kehamilan **`Ectopic`** saat ini (EP). Hasilnya memiliki dua kelompok kontrol: subyek datang untuk layanan induksi aborsi (IA) dan wanita yang melahirkan bayi (Deli). Kedua kelompok yang digunakan untuk mewakili intra-uterus kehamilan. Hasil dalam penelitian ini memiliki tiga kategori nominal.

### *Tabulation*

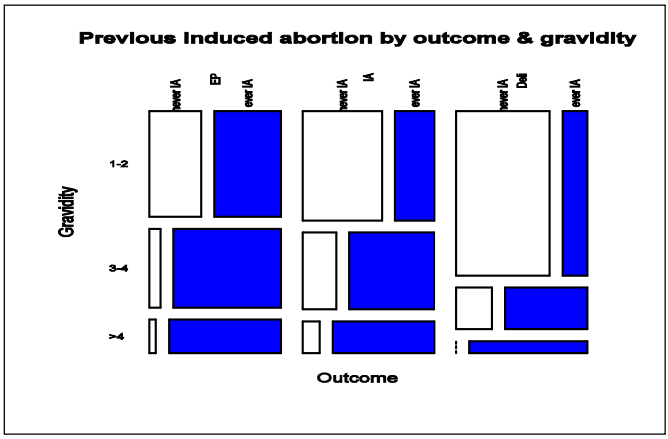
```
> zap()
> data(Ectopic);  use(Ectopic)
> des()

No. of observations =723
  Variable      Class           Description
1 id            integer
2 outc          factor          Outcome
```

```
3 hia           factor          Previous induced abortion
4 gravi         factor          Gravidity

> tabpct(outc, hia, graph=FALSE)
Original table
       Previous induced abortion
Outcome  never IA  ever IA  Total
  EP           61      180    241
  IA          110      131    241
  Deli        158       83    241
  Total       329      394    723

Row percent
       Previous induced abortion
Outcome  never IA  ever IA  Total
   EP           61      180    241
           (25.3)   (74.7)  (100)
   IA          110      131    241
           (45.6)   (54.4)  (100)
   Deli        158       83    241
           (65.6)   (34.4)  (100)
```

```
Column percent
       Previous induced abortion
Outcome  never IA       %  ever IA       %
  EP           61  (18.5)      180  (45.7)
  IA          110  (33.4)      131  (33.2)
  Deli        158  (48.0)       83  (21.1)
  Total       329   (100)      394   (100)
```

Dua cara tabulasi mengungkapkan proporsi tertinggi (74,7%) dari yang pernah IA dalam kelompok EP dibandingkan dengan 54,4% pada IA dan 34,4% pada kelompok Deli.

```
> table1 <- table(outc, gravi, hia)
> plot(table1, col=c("white", "blue"), las=4, main="Previous
   induced abortion by outcome & gravidity", xlab="Outcome",
   ylab= "Gravidity")
```




Plot mosaik memberikan informasi yang rumit. Kolom plot adalah hasil, yang terbagi menjadi EP, IA dan Deli, seperti dijelaskan sebelumnya. Ukuran dari 3 "kolom" adalah sama (241 subjek). Setiap baris mewakili tiga tingkat graviditas (jumlah kehamilan): masing-masing 1-2, 3-4 dan > 4. Distribusi graviditas antara EP dan kelompok IA kurang lebih sama, yaitu sekitar setengah memiliki 1-2 kehamilan, sedangkan di antara para wanita yang datang untuk memberikan bayi, persentase dalam kelompok ini jauh lebih tinggi (sekitar 75%) . Akhirnya, informasi dapat diperoleh dari warna yang berbeda. Daerah biru mewakili kaum perempuan yang mengalami induksi aborsi sebelumnya sedangkan putih melambangkan mereka yang tidak. Dalam setiap kolom, seperti persentase tampaknya meningkat dengan graviditas, yaitu perempuan yang memiliki graviditas tinggi akan memiliki tingkat yang lebih tinggi dari paparan induksi aborsi di masa lalu. Perbandingan antara tiga kolom, yang merupakan hipotesis utama dari studi ini, menunjukkan bahwa proporsi warna biru adalah tertinggi di antara kelompok EP.

## Regresi Logistik Polytomous menggunakan R

Regresi logistik Polytomous, kadang-kadang disebut regresi logistik multinomial, digunakan ketika hasilnya mengandung lebih dari dua kategori. Dalam kasus ini, kode untuk hasilnya adalah: 1 = EP, 2 = IA dan 3 = Deli. Perintah untuk regresi

tersebut terkandung dalam paket 'nnet', paket yang didasarkan pada konsep jaringan saraf.

```
> library(nnet)
> multi1 <- multinom(outc ~ hia); multi1
# weights:  9 (4 variable)
initial  value 794.296685
final  value 753.732244
converged
Call:
multinom(formula = outc ~ hia)

Coefficients:
     (Intercept) hiaever IA
IA       0.58958   -0.90735
Deli     0.95170   -1.72585

Residual Deviance: 1507.5
AIC: 1515.5
```

Bagian atas dari output menyangkut proses iterasi dari jaringan saraf. Bagian penting untuk epidemiologi adalah dalam bagian 'Koefisien:'. Interpretasi dari koefisien regresi logistik polytomous agak rumit, terutama ketika desain memiliki satu kelompok kasus dan lebih dari satu kelompok kontrol.

Ada tiga kategori hasil. Yang pertama, 'EP', adalah referensi terhadap yang dua perbandingan yang dibuat. Risiko untuk menjadi EP dalam hal ini adalah kembali ke kemungkinan tidak EP dalam dataset. Karena penelitian ini adalah studi kasus kontrol, nilai-nilai intercept harus diabaikan. Bagian yang paling penting adalah koefisien dari 'hia'.

Bagi mereka yang memiliki riwayat induksi aborsi, yang logit menjadi IA dalam kehamilan dengan perubahan -0,90735 unit. Hal ini setara dengan rasio odds (rasio ganjil) exp (-0,90735) atau 0,403.

"Kemungkinan memiliki kehamilan intra-uterus (dan akhirnya datang untuk induksi aborsi) dikurangi dengan faktor 0,403 jika subjek memiliki sejarah induksi aborsi" dapat diulang sebagai "kemungkinan memiliki kehamilan ektopik (dan karena itu tidak dalam kelompok IA) meningkat dengan 1/0.403, atau faktor 2.48 ".

Demikian pula, rasio odds (rasio ganjil) untuk EP menggunakan Deli sebagai kontrol adalah 1 / e-1,7258539 = 5,617.

Perlu diingat bahwa dalam bab tentang regresi logistik, rasio odds untuk riwayat induksi aborsi sebelumnya menggunakan dua kelompok gabungan yang diperoleh sebagai berikut:

```
> logistic.display(glm(outc=="EP" ~ hia, binomial))
Logistic regression predicting outc == "EP"

                        OR(95%CI)       P(Wald's test) P(LR-test)
hia: ever IA vs never IA  3.7 (2.63,5.2)  < 0.001        < 0.001

Log-likelihood = -429.3863
No. of observations = 723
AIC value = 862.772
```

Rasio odds dari regresi logistik dalam bab 15 dari 3,695 adalah antara dua rasio odds dihitung dari regresi logistik polytomous dalam bab ini.

Standar error (kesalahan standar) dapat diperoleh dengan perintah berikut:

```
> summary(multi1) -> s1; s1
========== coefficient section omitted =============
Std. Errors:
     (Intercept) hiaever IA
IA       0.15964    0.19666
Deli     0.15074    0.20081
========== correlation section omitted ============
```

Hanya bagian standar error yang ditampilkan, karena bagian koefisien dimunculkan di atas dengan perintah sebelumnya dan bagian korelasi tidak terkait langsung di sini.

Untuk mendapatkan nilai z untuk setiap sel, adalah sebagai berikut:

```
> coef(s1) / s1$st -> z; z
     (Intercept) hiaever IA
IA        3.6932    -4.6139
Deli      6.3136    -8.5943
```

Tingginya kadar 'z' menunjukkan koefisien beberapa kali nilai standard error. Dengan kata lain, koefisien jauh dari 0, di mana didasarkan pada hipotesis nol (tidak ada hubungan). P-value dapat diperoleh dengan:

```
> pnorm(abs(z), lower.tail=FALSE)*2 -> p.values
> p.values
     (Intercept) hiaever IA
IA    2.2143e-04 3.9513e-06
```

```
Deli  2.7264e-10 8.3774e-18
```

Perhatikan bahwa nilai absolut dari 'z' digunakan sebelum menghitung P-Values.

Interval kepercayaan 95% dari koefisien dapat dihitung berdasarkan koefisien dan standar error.

```
> coeff.lower.95ci <- coef(s1) - qnorm(.975) * s1$st
> coeff.lower.95ci
> coeff.upper.95ci <- coef(s1) + qnorm(.975) * s1$st
> coeff.upper.95ci
```

Rasio odds dan interval kepercayaan 95% dapat dicapai dari eksponensiasi dari koefisien serta dari nilai batas atas dan bawah 95% dari CI.

### *Tampilan hasil Regresi Polytomous*

Proses komputasi di atas cukup rumit. Untuk menyederhanakan jumlah mengetik dan untuk mendapatkan hasil output yang rapi, mlogit.display dari *Epicalc* dapat digunakan pada ringkasan model.

```
> mlogit.display(multi1)

Outcome =outc; Referent group = EP
            IA
            Coeff./SE       RRR(95%CI)
(Intercept)  0.59/0.16***    -
hiaever IA  -0.91/0.197***  0.404(0.275,0.593)

            Deli
            Coeff./SE       RRR(95%CI)
(Intercept)  0.95/0.151***   -
hiaever IA  -1.73/0.201***  0.178(0.12,0.264)

Residual Deviance: 1507.464
AIC = 1515.464
```

Format output telah dimodifikasi agar sesuai dengan halaman. P-values adalah kode dengan jumlah tanda bintang sesuai dengan yang digunakan dalam ringkasan dari 'GLM' dan model 'lm'. Rasio ganjil untuk *intercept* tidak relevan dan karena itu dihilangkan. Seperti telah dibahas sebelumnya,rasio ganjil di sini

tidak untuk resiko kehamilan ektopik tetapi untuk resiprokal mereka.

Untuk memasukkan variabel 'gravi' dalam model, adalah sebagai berikut :

```
> multi2 <- multinom(outc ~ hia + gravi)
> mlogit.display(multi2)
```

Opsional, tiga perintah teratas dapat dikombinasikan dan diganti dengan satu di bawah yang memberikan hasil yang sama

```
> mlogit.display(multinom(outc ~ hia + gravi))
# weights:  15 (8 variable)
initial  value 794.296685
iter  10 value 744.763718
final  value 744.587307
converged

Outcome =outc; Referent group = EP

            IA
            Coeff./SE       RRR(95%CI)
(Intercept)  0.51/0.165**    -
hiaever IA  -1.11/0.223***  0.33(0.213,0.511)
gravi3-4     0.39/0.224     1.472(0.95,2.283)
gravi>4      0.47/0.295     1.599(0.897,2.85)

            Deli
            Coeff./SE       RRR(95%CI)
(Intercept)  1.02/0.154***  -
hiaever IA  -1.49/0.222***  0.225(0.146,0.348)
gravi3-4    -0.47/0.24      0.628(0.392,1.004)
gravi>4     -0.7/0.366      0.499(0.243,1.022)

Residual Deviance: 1489.175
AIC = 1505.175
```

Sekali lagi, format output telah dimodifikasi agar sesuai dengan halaman. Tak satu pun dari koefisien dan rasio odds dari kemungkinan graviditas dalam model ini adalah signifikan. Namun, model ini memiliki penyimpangan yang jauh lebih rendah dibandingkan dengan model sisa 'multi1'. Penurunannya adalah dari 1507.464 ke 1489.175 atau sebesar 18,289 unit pada biaya memperkenalkan empat parameter lebih (tingkat graviditas dua untuk dua hasil) dapat dianggap bermanfaat karena P-Value dari chi-kuadrat sebesar

18,289 dengan 4 derajat kebebasan adalah 0,001. Selain itu, nilai AIC dari model 'multi2' sebesar 1505,175 jelas lebih kecil daripada yang dari 'multi1' yaitu 1515,464.

Untuk kesimpulan akhir, setelah penyesuaian untuk graviditas, riwayat induksi aborsi sebelumnya secara signifikan meningkatkan risiko kehamilan ektopik. Rasio odds adalah 1/.33 atau 3.03 jika klien saat ini meminta untuk induksi aborsi yang digunakan sebagai kelompok rujukan dan 1/.225 atau 4,4 jika perempuan yang melahirkan bayi adalah kelompok rujukan. Hal ini juga diketahui bahwa induksi aborsi sering diulang. Klien saat ini untuk layanan ini biasanya mengalami induksi aborsi lebih daripada populasi umum. Pasien kehamilan ektopik memiliki pengalaman bahkan lebih terhadap induksi aborsi daripada kelompok ini. Oleh karena itu, riwayat induksi aborsi sangat mungkin menjadi faktor risiko untuk kehamilan ektopik.

## *Pemilihan kelompok hasil rujukan*

Variabel hasil didalam suatu regresi logistik polytomous biasanya mengandung faktor lebih dari dua tingkat. Tingkat pertama biasanya diambil sebagai tingkat rujukan. Hasil analisis yang sama dapat diperoleh dengan menciptakan tiga dummy variabel hasil dan menggunakan mereka dalam format matriks dengan fungsi cbind.

```
> ep <- outc == "EP"
> ia <- outc == "IA"
> deli <- outc == "Deli"
> multi3 <- multinom(cbind(ep,ia,deli) ~ hia+gravi)
> summary(multi3)

> mlogit.display(multi3)
```

Perintah di atas akan memberikan hasil yang sama seperti yang dari 'multi2' kecuali bahwa nama-nama kelompok hasil dalam huruf kecil.

Karena kolom pertama selalu digunakan sebagai kelompok rujukan, kita dapat memanfaatkan metode ini untuk mengacak urutan variabel hasil dalam rangka untuk mengubah kelompok rujukan. Sebagai contoh, untuk menggunakan 'deli' sebagai tingkat rujukan, 'deli' dimasukkan sebagai kolom pertama dari matriks hasil:

```
> multi4 <- multinom(cbind(deli,ep,ia) ~ hia+gravi)
> mlogit.display(multi4)

Outcome =cbind(deli, ep, ia); Referent group = deli

            ep
            Coeff./SE       RRR(95%CI)
(Intercept) -1.02/0.154***  -
hiaever IA   1.49/0.222***   4.443(2.877,6.861)
gravi3-4     0.47/0.24       1.593(0.996,2.55)
gravi>4      0.7/0.366       2.005(0.979,4.107)

            ia
            Coeff./SE       RRR(95%CI)
(Intercept) -0.51/0.131***  -
hiaever IA   0.38/0.215      1.466(0.963,2.233)
gravi3-4     0.85/0.237***   2.346(1.475,3.732)
gravi>4      1.16/0.369**    3.205(1.554,6.607)
```

Output relatif mudah untuk menafsirkan. Menggunakan pengiriman sebagai hasil rujukan, bagi seorang wanita dengan riwayat induksi aborsi, kemungkinan yang 'EP' atau memiliki kehamilan ektopik dalam pengakuan ini meningkat 4,443 kali lipat (yang sangat signifikan) dan bahwa untuk menjadi (mengulang) pasien induksi aborsi hanya meningkat 47 persen (OR = 1,466, yang mana tidak signifikan). Di sisi lain, meningkatkan graviditas berarti akan meningkatkan risiko kehamilan ektopik secara signifikan, dan dalam hubungan kebiasaan respon dari dosis, meningkatkan kesempatan untuk menjadi klien untuk layanan induksi aborsi dalam kunjungan saat ini.

## Latihan

Dalam trial uji coba vaksin pada 120 tikus, 75 diberikan vaksin ('vac'= 1) sedangkan 45 lainnya diberi plasebo ('vac'= 0). Diantaranya adalah 35 tikus muda ('agegr' = 0) dan 85 tikus tua ('agegr' = 1)

Ada tiga tingkat hasil: 1 = tidak ada perubahan, 2 = menjadi imun, 3 = meninggal

| Outcome | vac | agegr | total |
|---|---|---|---|
| 1 | 0 | 0 | 25 |
| 1 | 0 | 1 | 15 |
| 1 | 1 | 0 | 4 |
| 1 | 1 | 1 | 8 |
| 2 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 |
| 2 | 1 | 0 | 25 |
| 2 | 1 | 1 | 35 |
| 3 | 0 | 0 | 3 |
| 3 | 0 | 1 | 1 |
| 3 | 1 | 0 | 2 |
| 3 | 1 | 1 | 1 |

Soal 1.

Apakah ada perbedaan pada kelompok usia antara dua kelompok penerima vaksinasi?

Soal 2.

Apakah ada hubungan antara kelompok usia dan hasil?

Soal 3.

Apakah ada perbedaan hasil antara kelompok perlakuan dan plasebo vaksin

# BAB 18

# Regresi Logistik Ordinal

Pada bab sebelumnya, semua variabel dimana faktor-faktornya diperlakukan sebagai variabel kategori berurut. Regresi logistik *polytomous* berhubungan dengan memprediksi hasil (outcomes) yang bersifat kategori tetapi tidak berurut. Dalam banyak keadaan, hasil (outcome) memiliki beberapa cara pengurutan. Menggunakan regresi logistik *polytomous* untuk situasi tersebut akan menghilangkan kemampuan untuk mendeteksi asosiasi seperti menyalahartikan cara variabel terikat berhubungan dengan variabel penjelas.

## *Faktor Terurut*

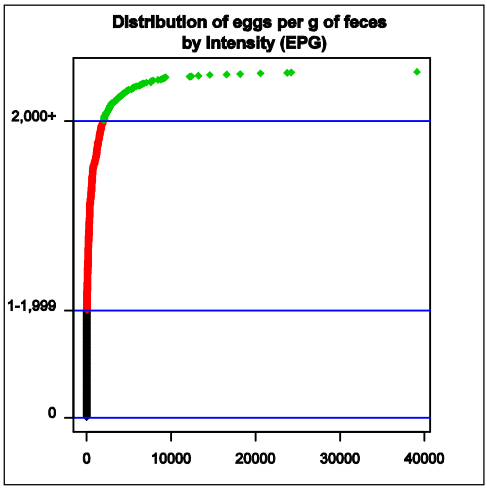
Bab ini menggunakan kumpulan dari dari sebuah survey terhadap infeksi cacing tambang di wilayah selatan Thailand yang dilakukan pada tahun 1993. Tujuannya adalah untuk mengetahui pengaruh usia dan penggunaan sepatu ('shoes') terhadap intensitas infeksi.

```
> library(nnet) # For polytomous logistic regression
> library(MASS) # For ordinal logistic regression
> zap()
> data(HW93)
> use(HW93)
> des()
```

```
No. of observations = 637
  Variable      Class           Description
1 id            integer
2 epg           numeric         eggs per g of faeces
3 age           integer
4 shoes         factor          Shoe wearing
5 intense       factor          Intensity (EPG)
6 agegr         factor          Age group

> summ()
  No. of observations = 637
  Var. name Obs.  mean    median  s.d.    min. max.
1 id        637   325.38  325     185.79  1    646
2 epg       637   1141.85 207     2961.82 0    39123
3 age       637   25.94   23      19.47   2    78
4 shoes     637   1.396   1       0.489   1    2
5 intense   637   1.834   2       0.652   1    3
6 agegr     637   1.667   2       0.608   1    3
```

Variabel 'intense' merupakan bentuk kategori dari variabel 'epg'.

```
> summ(epg, by=intense)
For intense = 0
  obs. mean   median  s.d.   min.   max.
  197  0      0       0      0      0

For intense = 1-1,999
  obs. mean   median  s.d.    min.   max.
  349  539    345     512.368 23     1910

For intense = 2,000+
  obs. mean   median  s.d.     min.   max.
  91   5930   3960    5792.453 2020   39100
```




### *Menggunakan Regresi Logistik Polytomous*

```
> poly.hw <- multinom(intense ~ agegr + shoes)
> mlogit.display(poly.hw)

Outcome =intense; Referent group = 0
               1-1,999
               Coeff./SE       RRR(95%CI)
(Intercept)    0.29/0.138*     -
agegr15-59 yrs 0.87/0.216***   2.39(1.56,3.65)
agegr60+ yrs   0.77/0.41       2.16(0.97,4.82)
shoesyes       -0.48/0.212*    0.62(0.41,0.94)

               2,000+
               Coeff./SE        RRR(95%CI)
(Intercept)    -0.97/0.204***   -
agegr15-59 yrs 1.03/0.306***    2.8(1.54,5.1)
agegr60+ yrs   1.8/0.478***     6.05(2.37,15.44)
shoesyes       -1.34/0.317***   0.26(0.14,0.49)

Residual Deviance: 1196.8
AIC = 1212.8
```

Untuk infeksi ringan (1-1,999 epg), orang dewasa muda memiliki resiko tertinggi daripada anak-anak. Untuk infeksi berat (2,000+ epg), orang dewasa muda dan orang tua memiliki masing-masing 2.8 dan 6.1 kali resiko tertinggi daripada anak-anak. Penggunaan sepatu memiliki efek perlindungan terhadap infeksi, ringan dan berat, dengan odd rasio masing-masing 0.62 dan 0.262.

## Pemodelan ordinal terikat

Sebagai alternatif, karena intensitas merupakan variabel terikat berurut, maka dapat dicoba regresi ordinal logistik. Perintah `polr` dari paket **MASS** akan menjalankannya. Tetapi pertama kita harus memberitahukan **R** bahwa variabel terikatnya berurutan.

```
> class(intense)       # "factor"
> intense.ord <- ordered(intense)
> class(intense.ord)   # "ordered" "factor"
> ord.hw <- polr(intense.ord ~ agegr + shoes)
> summary(ord.hw)

Coefficients:
                    Value Std. Error   t value
agegr15-59 yrs  0.7744521  0.1834157  4.222388
agegr60+ yrs    1.2797213  0.3226504  3.966278
shoesyes       -0.7234746  0.1780106 -4.064223

Intercepts:
               Value   Std. Error t value
0|1-1,999      -0.6301  0.1293    -4.8726
1-1,999|2,000+  2.0745  0.1579    13.1363

Residual Deviance: 1204.920
AIC: 1214.920
```

Model regresi ordinal logistik memiliki 2 intercept, satu untuk setiap titik potong variabel. Nilai intercept ini tidak terlalu berarti sehingga dapat diabaikan pada tahap ini. Koefisien dari semua variabel penjelas dibagi oleh dua titik potong variabel terikat. Titik potong yang pertama, logit untuk mendapatkan infeksi apapun (intense= 1- 1,999 and 2,000 + epg vs no infection) berkurang sebesar 72% jika subjek memakai sepatu, sehingga itu adalah logit pada titik

potong kedua (intensity of 2,000 + epg vs any lower levels of intensity). Kedua koefisien yang positif mengindikasi bahwa resiko infeksi akan meningkat sesuai waktu. Penggunaan sepatu memiliki koefisien negatif mengindikasikan bahwa penggunaan sepatu melindungi kedua tingkat infeksi.

```
> summary(ord.hw) -> s1
> attributes(s1)
```

Untuk menghitung P value untuk 'shoes', ketik:

```
> coef(s1)
> t <- coef(s1)[,3]
> df <- s1$df.residual
> pt(abs(t), df, lower.tail=F)
agegr15-59 yrs   agegr60+ yrs       shoesyes
  1.386181e-05   4.067838e-05   2.713385e-05
     0|1-1,999 1-1,999|2,000+
  6.969506e-07   2.521906e-35
```

Perintah diatas mendefinisikan 't' dan 'df' pada ringkasan regresi. Perintah terakhir menggunakan nilai mutlak 't' untuk menghitung P value dua arah. Semua P value signifikan.

## *'ordinal.or.display'*

Epicalc memiliki fungsi untuk menampilkan odds ratio ordinal dan interval kepercayaan sebesar 95%.

```
> ordinal.or.display(ord.hw)
               Ordinal OR lower95ci upper95ci P.value
agegr15-59 yrs 2.169      1.517     3.116     1.39e-05
agegr60+ yrs   3.596      1.913     6.788     4.07e-05
shoesyes       0.485      0.341     0.686     2.71e-05
```

Kesimpulan yang diperoleh dari model regresi ordinal logistik adalah intensitas infeksi meningkat signifikan terhadap kelompok umur dan tingkat signifikan berkurang dengan penggunaan sepatu. Pada setiap titik potong intensitas infeksi, rata-rata, penggunaan sepatu berasosiasi dengan penurunan sebesar 0.48 atau sebagian dari *odds* mereka yang tidak menggunakan sepatu.

## Referensi

Venables, W. N. and Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth edition. Springer.

## Latihan

Tingkat rasa sakit setelah pengobatan (1 = no pain, 2 = some pain, 3 = severe pain) diukur setelah perawatan sekelompok subjek dengan penghilang rasa sakit (Drug = 1) terhadap placebo (Drug = 0) pada laki-laki (1) dan wanita (0) dengan data sebagai berikut :

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Male | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| Drug | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| Pain | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
| Total | 3 | 5 | 15 | 10 | 5 | 7 | 8 | 5 | 10 | 10 | 10 | 2 |

Analisis pengaruh obat tersebut dengan penyesuaian terhadap jenis kelamin dengan menggunakan regresi logistik polytomous dan logistik ordinal.

BAB 19

# Regresi Poisson dan Binomial Negatif

### *Distribusi Poisson*

Di alam, suatu kejadian biasanya terjadi dalam jumlah waktu yang sangat kecil. Pada sebarang titik waktu, peluang menghadapi kejadian seperti itu sangat kecil. Daripada peluang, pengukuran difokuskan terhadap kepadatan, yang berarti kejadian 'dihitung' selama periode waktu. Sementara waktu adalah dimensi satu, konsep yang sama berlaku terhadap kepadatan jumlah objek yang kecil dalam ruang dua atau tiga dimensi.

Saat satu kejadian bebas dari kejadian lain, proses terjadinya adalah acak. Secara matematika, dapat dibuktikan bahwa dalam kondisi ini, kepadatan dalam berbagai satuan waktu yang berubah dengan varians sama dengan kepadatan rata-rata. Saat kemungkinan terjadinya kejadian dipengaruhi oleh beberapa faktor, sebuah model dibutuhkan untuk menjelaskan dan memprediksi kepadatan. Keragaman antara strata yang berbeda dijelaskan oleh faktor-faktor. Dengan setiap distribusi strata adalah acak.

## *Regresi Poisson*

Regresi Poisson berhubungan dengan variabel terikat yang terhitung di alam (seluruhnya angka atau bilangan bulat). Independen kovariat mirip dengan yang ditemui dalam regresi linear dan logistik.

Dalam epidemiologi, regresi Poisson digunakan untuk menganalisis data kohort berkelompok, melihat kepadatan kejadian setiap waktu yang diberikan oleh subjek dari kelompok dengan karakteristik serupa.

Regresi Poisson merupakan salah satu dari tiga model generalized linear models (GLM) yang umum digunakan dalam kajian epidemiologi. Dua model lainnya yang lebih umum digunakan adalah regresi linear dan logistik, yang telah dibahas pada bab sebelumnya.

Terdapat 2 asumsi utama untuk regresi Poisson. Pertama, *risk* homogen antara setiap titik waktu diberikan oleh subjek berbeda yang memiliki kelompok karakteristik serupa (seperti jenis kelamin, kelompok umur) dan periode sama. Kedua, asimtotik, atau semakin membesar ukuran sampel, rata-rata hitungannya sama dengan varians.

## *Keuntungan Model Regresi Poisson*

Kemudahan metode regresi linear (asumsikan varians konstan, error normal) tidak sesuai untuk data hitungan karena empat alasan utama:

1. model mungkin menyebabkan prediksi jumlah negatif,
2. varians variabel terikat dapat meningkat,
3. error tidak berdistribusi normal,
4. jumlah nol sulit diatasi dalam transformasi.

Rregresi Poisson mengurangi beberapa masalah yang dihasilkan oleh berbagai teknik regresi lainnya. Sebagai contoh, dalam regresi logistik, subjek berbeda dapat memiliki variabel bebas dengan titik waktu berbeda. Menganalisis faktor resiko sementara mengabaikan perbedaan titik waktu adalah tindakan salah. Dalam analisis *survival* menggunakan regresi Cox (dibahas pada bab 22), hanya *hazard ratio* dan kepadatan kejadian setiap subkelompok tidak dihitung. Para analis dan pembaca mungkin tidak memiliki gambaran yang jelas pada statistik deskriptif pada *baseline risks* ini. Dengan kata lain, regresi Poisson menghasilkan

'baseline incidence density' dan 'incidence density ratio' diantara strata.

### *Contoh: Kajian peleburan Montana*

Kumpulan data **Montana** diekstraksi dari kajian kerja kelompok yang dilakukan untuk menguji asosiasi antara penyakit pernapasan dan penyebaran arsenik dalam industri, setelah penyesuaian dengan berbagai faktor resiko lainnya. Variabel terikat utama adalah 'respdeath'. Berikut merupakan hitungan jumlah kematian antara 'personyrs' atau subjek dari setiap tahun masing-masing kategori. Varibel lainnya merupakan kovariat independen termasuk kelompok umur 'agegr', periode pekerjaan 'period', waktu permulaan pekerjaan 'start' dan tingkat penyebaran selama periode kajian 'arsenic'. Pertama bacalah data kemudian periksa variabelnya.

```
> zap()
> data(Montana)
> use(Montana)
> summ()

No. of observations = 114

  Var. name Obs.  mean    median  s.d.   min.  max.
1 respdeath 114   2.42    1       3.3    0     19
2 personyrs 114   1096.41 335.15  2123.1 4.2   12451
3 agegr     114   2.61    3       1.1    1     4
4 period    114   2.4     2       1.09   1     4
5 start     114   1.46    1       0.5    1     2
6 arsenic   114   2.47    2       1.11   1     4
```

```
> des()
No. of observations = 114

  Variable      Class           Description
1 respdeath     integer
2 personyrs     numeric
3 agegr         integer
4 period        integer
5 start         integer
6 arsenic       integer
```

Empat variabel terakhir diklasifikasikan sebagai integer. Kita perlu memberitahu **R** untuk menginterpretasinya sebagai variabel kategori, atau faktor, dan masukkan label untuk setiap tingkatan. Hal ini dapat dilakukan dengan perintah `factor` dengan sebuah agumen 'labels' didalamnya.

```
> agegr <- factor(agegr, labels=c("40-49","50-59","60-
  69","70-79"))
> period <- factor(period, labels=c("1938-1949", "1950-
  1959", "1960-1969", "1970-1977"))
> start <- factor(start, labels=c("pre-1925", "1925 &
  after"))
> arsenic1 <- factor(arsenic, labels=c("<1 year", "1-4
  years","5-14 years", "15+ years"))

> label.var(agegr, "Age group")
> label.var(period, "Period of employment")
> label.var(start, "Era of starting employment")
> label.var(arsenic1, "Amount of exposure to arsenic")
> des()

No. of observations =114
  Variable      Class           Description
1 respdeath     integer
2 personyrs     numeric
3 agegr         factor          Age group
4 period        factor          Period of employment
5 start         factor          Era of starting employment
6 arsenic       integer
7 arsenic1      factor          Amount of exposure to
  arsenic
```

Kita tetap menjaga variabel original 'arsenic' tanpa perubahan untuk digunakan kemudian.

### *Rincian kejadian dengan umur dan waktu*

Mari kita melihat rincian setiap tahun dari umur dan waktu. Pertama, buat sebuah tabel untuk total setiap tahun:

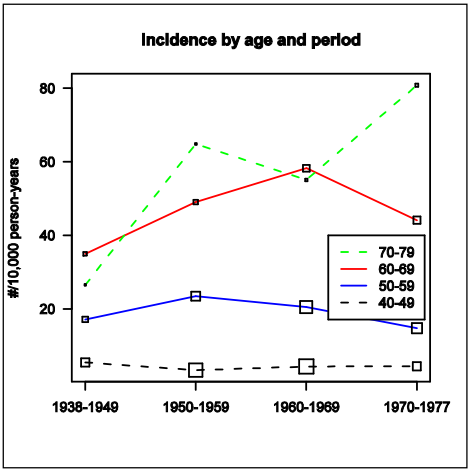
```
> tapply(personyrs, list(period, agegr), sum) ->
  table.pyears
```

Gunakan prosedur yang sama untuk jumlah kematian, dan hitung tabel kejadian per 10,000 setiap tahun untuk setiap sel.

```
> tapply(respdeath, list(period, agegr), sum) ->
  table.deaths
> table.inc10000 <- table.deaths/table.pyears*10000
> table.inc10000
              40-49    50-59    60-69    70-79
1938-1949 5.424700 17.13102 34.95107 26.53928
1950-1959 3.344638 23.47556 49.01961 64.82632
1960-1969 4.341516 20.49375 58.23803 55.06608
1970-1977 4.408685 14.77747 44.09949 80.81413
```

Sekarang, buat sebuah plot deret waktu dari kejadian:

```
> plot.ts(table.inc10000, plot.type="single", xlab=" ",
   ylab="#/10,000 person-years", xaxt="n", col=c("black",
   "blue","red","green"), lty=c(2,1,1,2), las=1)

> points(rep(1:4,4), table.inc10000, pch=22,
  cex=table.pyears
   / sum(table.pyears) * 20)

> title(main = "Incidence by age and period")

> axis(side = 1, at = 1:4, labels = levels(period))

> legend(3.2,40, legend=levels(agegr)[4:1], col=c("green",
   "red", "blue", "black"), bg = "white", lty=c(2, 1, 1, 2))
```



Grafik diatas menunjukkan bahwa kelompok umur yang paling tua umumnya berasosiasi dengan resiko yang tinggi. Dengan kata lain, ukuran sampel (tampak dari ukuran kotak setiap titik) menurun terhadap umur.

Peluang pengaruh membingungkan dari umur dapat lebih baik diperiksa dengan regresi Poisson.

## Pemodelan dengan regresi Poisson

```
> mode11 <- glm(respdeath ~ period, offset = log(personyrs),
    family = poisson)
> summary(mode11)
==============================
Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)      -6.4331     0.1715 -37.511   <2e-16
period1950-1959   0.2365     0.2117   1.117   0.2638
period1960-1969   0.3781     0.2001   1.889   0.0588
period1970-1977   0.4830     0.2036   2.372   0.0177

AIC: 596
==============================
```

Pilihan 'offset = log(personyrs)' memungkinkan variabel 'personyrs' menjadi penyebut untuk jumlah 'respdeath'. Transformasi logaritma dibutuhkan karena untuk model linear umum Poisson, fungsi penghubung adalah logaritma natural, dan standar penghubung keluarga Poisson adalah penghubung log.

Kriteria penting dalam memilih fungsi penghubung untuk berbagai keluarga distribusi adalah untuk memastikan bahwa nilai dugaan model berada dalam batas yang wajar. Menentukan penghubung log (standar  untuk Poisson) memastikan bahwa jumlah nilai dugaan semuanya lebih besar atau sama dengan nol.

**Note**:

Untuk rincian lebih lanjut mengenai penghubung standar berbagai keluarga distribusi yang berhubungan dengan pemodelan linear umum, lihat pada **R** 'help(family)'.

Model pertama dengan regresi Poisson dengan 'period' merupakan satu-satunya variabel penjelas yang menyatakan bahwa laju kematian meningkat terhadap waktu. Model dapat diuji untuk kesesuaian model dan diperiksa apakah asumsi Poisson disebutkan sebelumnya pada bab ini tidak dipenuhi.

## Uji kesesuaian model

Untuk menguji kesesuaian model Poisson, ketik:

```
> poisgof(mode11)
$results
[1] "Goodness-of-fit test for Poisson assumption"

  $chisq
[1] 369.27

  $df
[1] 110

  $p.value
[1] 9.5784e-30
```

Komponen '$chisq' sebenarnya dihitung dari deviasi model, parameter mencerminkan tingkat error. Nilai chi-squared yang besar dengan derajat kebebasan kecil menyebabkan tidak terpenuhi secara signifikan asumsi Poisson ($p < 0.05$). Jika hanya P value yang diinginkan, maka perintahnya menjadi lebih singkat.

```
> poisgof(mode11)$p.value
```

P value yang sangat kecil mengindikasikan ketidaksesuaian.

**Note:**

Harus diingat bahwa metode ini berdasarkan asumsi bahwa sampel berukuran besar. Alternatif metode adalah model regresi binomial negative dan periksa jika parameter berbeda dari 1, seperti yang akan ditunjukkan pada bagian akhir bab ini.

Sekarang ditambahkan variabel penjelas yang kedua 'agegr' dalam model.

```
> mode12 <- glm(respdeath~agegr+period,
  offset=log(personyrs),
```

```
   family = poisson)
> AIC(mode12)  # 396.64
```

AIC telah meningkat dengan baik dari model 'model1' ke 'model2' yang mengindikasikan ketidaksesuai pada model pertama.

```
> poisgof(mode12)$p.value   # 0.00032951
```

Bagaimanapun model kedua, 'model2', masih tidak memenuhi asumsi Poisson.

```
> mode13 <- glm(respdeath ~ agegr, offset = log(personyrs),
   family = poisson)
> AIC(mode13) # 394.47
> poisgof(mode13)$p.value   # 0.0003295
```

Megganti 'period' lebih lanjut mengurangi nilai AIC tetapi masih tidak memenuhi asumsi Poisson seperti model sebelumnya. Langkah selanjutnya adalah menambahkan variabel penjelas utama 'arsenic1'.

```
> mode14 <- glm(respdeath ~ agegr+arsenic1,
  offset=log(personyrs), family = poisson)
> summary(mode14)
Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)         -7.995      0.224  -35.74  < 2e-16
agegr50-59           1.462      0.245    5.96  2.5e-09
agegr60-69           2.350      0.238    9.87  < 2e-16
agegr70-79           2.599      0.256   10.14  < 2e-16
arsenic11-4 years    0.804      0.158    5.10  3.4e-07
arsenic15-14 years   0.596      0.206    2.89   0.0038
arsenic115+ years    0.998      0.176    5.67  1.4e-08

    Null deviance: 376.02  on 113  degrees of freedom
Residual deviance: 122.25  on 107  degrees of freedom
AIC: 355.0

> poisgof(mode14)$p.value   # 0.14869
```

Model terakhir, 'model4', memiliki AIC yang sangat rendah dibanding 'model3' dan memenuhi asumsi. Satu atau banyak tahun penyebaran arsenik berasosiasi dengan pengaruh tinggi terhadap penyakit pernafasan.

### *Linear dose response relationship*

Secara alternatif, meskipun arsenik dimasukkan sebagai variabel kategori, tetapi dapat dimasukkan ke dalam model sebagai kontinu variabel. Jika P value signifikan maka hal ini menyiratkan bahwa terdapat *dose-response relations*

antara penyebaran arsenik dan resiko penyakit. Variabel original 'arsenic' dimasukkan pada model berikutnya.

```
> model5 <- glm(respdeath~agegr+arsenic,
  offset=log(personyrs),
    family=poisson)
> summary(model5)
============================
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -8.2416     0.2327  -35.42  < 2e-16
agegr50-59    1.4572     0.2454    5.94  2.9e-09
agegr60-69    2.3236     0.2379    9.77  < 2e-16
agegr70-79    2.5572     0.2558   10.00  < 2e-16
arsenic       0.3358     0.0524    6.40  1.5e-10
AIC: 360.31

> poisgof(model5)$p.value    # 0.069942
```

Meskipun bentuk linear signifikan, nilai AIC pada '`model5`' lebih besar dari '`model4`'. Maka lebih baik tetap menjadikan arsenik sebagai faktor. Bagaimanapun, dari model '`model4`' tidak terdapat kenampakan kenaikan resiko kematian lebih dari 4 tahun penyebaran arsenik maka hal itu dapat menjadi kombinasi yang baik hanya dalam dua level.

```
> arsenic2 <- arsenic1
> levels(arsenic2) <- c("<1 year", rep("1+ years", 3))
> label.var(arsenic2, "Exposure to arsenic")
> model6 <- glm(respdeath ~ agegr + arsenic2,
  offset=log(personyrs), family=poisson)
> summary(model6)
============================
Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)       -8.009      0.223  -35.86  < 2e-16
agegr50-59         1.470      0.245    5.99  2.0e-09
agegr60-69         2.366      0.237    9.98  < 2e-16
agegr70-79         2.624      0.255   10.30  < 2e-16
arsenic21+ years   0.811      0.121    6.70  2.1e-11
============================
AIC: 353.8

> poisgof(model6)$p.value    # 0.13999
```

Pada bagian ini, kita akan menerima '`model6`' sebagai model pilihan karena model tersebut memiliki nilai AIC terkecil antara semua model yang telah dicoba. Kita menyimpulkan bahwa penyebaran arsenik untuk paling sedikit satu tahun akan meningkatkan resiko penyakit sebesar exp(0.8109) atau 2.25 kali dari signifikan statistik.

## Kepadatan kejadian

Pada model Poisson, hasilnya berupa sebuah hitungan (*count*). Dalam model linear umum, hubungan antara nilai keluaran (diukur dalam data dan diprediksi oleh model menggunakan nilai dugaan) dan prediktor linear bergantung pada fungsi penghubung. Fungsi penghubung menghubungkan nilai rata-rata keluaran dan prediktor linearnya. Umumnya, fungsi penghubung untuk distribusi Poisson adalah logaritma natural. Dengan offset menjadi log(setiap waktu), nilai keluarannya menjadi log(kepadatan kejadian).

Matriks 'table.inc10000' (telah dibuat sebelumnya) memberikan gambaran kasar mengenai kepadatan kejadian kelompok umur dan waktu. Setiap model regresi Poisson diatas dapat digunakan untuk menghitung prediksi kepadatan kejadian saat variabel-variabel dalam model diketahui. Misalnya, untuk menghitung kepadatan kejadian dari 100,000 populasi orang yang berumur antara 40-49 tahun yang tercemar arsenik dalam waktu kurang dari satu tahun menggunakan '`model6`', ketik:

```
> newdata <- as.data.frame(list(agegr="40-49",
   arsenic2="<1 year", personyrs=100000))
> predict(model6, newdata, type="response")
[1] 33.257
```

Populasi ini memiliki estimasi kepadatan kejadian sebesar 33.26 per 100,000 setiap tahun.

### *Rasio kepadatan kejadian*

Dalam sebuah kasus studi kontrol, odds ratio digunakan untuk membandingkan pola penyebaran antara kasus dan kontrol. Dalam studi kelompok, nilai ini sama dengan rasio antara *odds* terkena penyakit antar kelompok tercemar dan tidak tercemar. Jika penyakitnya jarang terjadi, *the odds* mendekati peluang atau

resiko. Rasio resiko untuk dua kelompok disebut 'risk ratio' atau 'relative risk'.

Dalam studi kelompok sebenarnya, subjek tidak selalu memiliki durasi follow up. yang sama. Resiko relatif mengabaikan durasi follow up tersebut. Meskipun itu bukan pengukuran yang baik untuk perbandingan resiko dua kelompok. Pada bab ini, semua subjek menggabungkan waktu follow-up dan jumlahnya disebut 'person time', yang kemudian digunakan sebagai penyebut untuk setiap peristiwa, menghasilkan 'incidence density'. Membandingkan kepadatan kejadian antara dua kelompok subjek berdasarkan status penyebarannya adalah lebih adil daripada membandingkan resiko kasar. Rasio antara kepadatan kejadian dua kelompok disebut *incidence density ratio* (IDR), yang merupakan bentuk revisi dari resiko relatif.

Pada '`model6`', untuk menghitung rasio kepadatan kejadian antara subjek tercemar arsenik untuk satu tahun atau lebih terhadap subjek tercemar arsenik untuk dibawah satu tahun, kita dapat membagi antar kejadian sebelumnya berdasarkan kelompok sebelumnya.

```
> levels(newdata$arsenic2) <- c("<1 year", "1+ years")
> newdata <- rbind(newdata, list(agegr="40-49",
    arsenic2="1+ years", personyrs=100000))
> newdata
  agegr arsenic2 personyrs
1 40-49  <1 year     1e+05
2 40-49 1+ years     1e+05
> id <- predict(model6, newdata, type="response")
> idr.arsenic <- id[2]/id[1]
> idr.arsenic
[1] 2.2499
```

Prosedur diatas dimulai dengan menambahkan baris baru untuk frame data '`newdata`' yang memiliki semua hal yang sama seperti baris pertama kecuali variabel 'arsenic2' yaitu 1+ years". Kepadatan kejadian dari dua kondisi akan dihitung selanjutnya. IDR kemudian diperoleh dari pembagian kepadatan kejadian untuk arsenic2="<1 year" dengan arsenic2="1+ years".

Jalan yang lebih pendek untuk memdapatkan IDR adalah dengan mengeksponensialkan koefisien variabel 'arsenic', yang merupakan koefisien kelima dalam model.

```
> coef(model6)
  (Intercept)  agegr50-59  agegr60-69  agegr70-79 arsenic21+
```

```
     -8.00865     1.47015     2.36611     2.62375     0.81087

> exp(coef(model6)[5])
arsenic21+ years
          2.2499
```

### *'idr.display' to get 95% CI of IDR*

Langkah berikut ini menjelaskan bagaimana selang kepercayaan 95% dari IDR untuk semua variabel diperoleh.

```
> coeff <- coef(model6)
> coeff.95ci <- cbind(coeff, confint(model6))
```

Ingat bahwa `confint(glm6)` menyediakan selang kepercayaan 95% untuk koefisien dalam model.

```
> IDR.95ci <- round(exp(coeff.95ci), 1)[-1,]
```

Nilai yang diperlukan diperoleh dengan mengeksponensialkan matriks terakhir dimana baris pertama atau intercept dihilangkan. Hasil dibulatkan menjadi 1 desimal untuk tampilan lebih baik. Kemudian kolom matriks diberi nama dan diperoleh selang kepercayaan 95%.

```
> colnames(IDR.95ci) <- c("IDR", "lower95ci", "upper95ci")
> IDR.95ci
```

Langkah yang lebih sederhana adalah dengan menggunakan perintah *`idr.display`* dalam Epicalc.

```
> idr.display(model6, decimal=1)

Poisson regression predicting respdeath with offset =
  log(personyrs)

            crude IDR(95%CI)  adj. IDR(95%CI)  P(Wald's)  P(LR-
  test)
agegr: ref.=40-49                                          < 0.001
   50-59      4.5 (2.8,7.3)    4.3 (2.7,7)      < 0.001
   60-69     11.3 (7.1,17.9)  10.7 (6.7,17)     < 0.001
   70-79     14.5 (8.8,23.8)  13.8 (8.4,22.7)   < 0.001

arsenic2      2.5 (2,3.1)     2.2 (1.8,2.9)     < 0.001     < 0.001
   1+ years vs <1 year

Log-likelihood =  -171.9
```

```
No. of observations =  114
AIC value = 353.8
```

Perintah *idr.display* memberikan hasil hingga 3 desimal. Hal ini dapat dengan mudah diubah oleh pengguna.

## Regresi binomial negatif

Ingat kembali regresi Poisson, salah satu asumsi untuk model yang valid adalah rata-rata dan varians jumlah variabel adalah sama. Distribusi binomial negatif adalah bentuk yang lebih umum dari distribusi

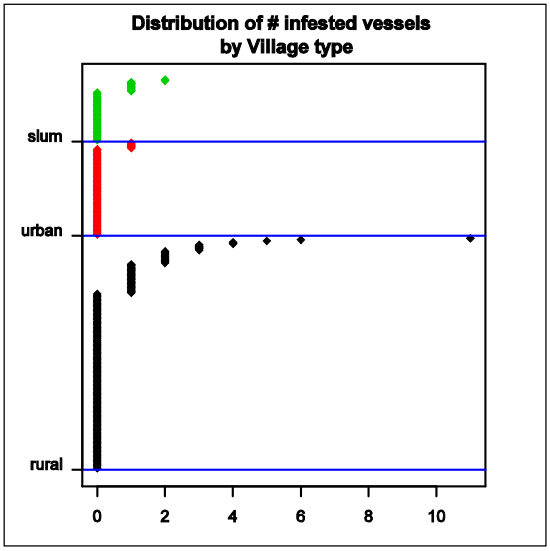
Ingat kembali regresi Poisson, salah satu asumsi untuk model yang valid yaitu rata-rata dan varians variabel count adalah sama. Distribusi binomial negatif adalah bentuk yang lebih umum dari distribusi yang biasa digunakan untuk data count response, mengikuti penyebaran terbesar atau varians of counts. Dalam kenyataannya, hal itu hampir sama untuk varians dari outcome menjadi lebih besar daripada rata-rata. Jika variabel count terlalu menyebar, regresi Poisson mengabaikan standar error variabel terikat. Saat overdispersion jelas, satu solusi untuk menetapkan bahwa error berdistribusi binomial negatif.

Regresi binomial negatif memberikan koefisien yang sama seperti dalam regresi Poisson tetapi menghasilkan standar error yang lebih besar. Interpretasi hasil sama seperti regresi Poisson.

Misalkan sebuah contoh jumlah kontainer air yang dipenuhi larva nyamuk dalam sebuah survey lapangan. Data terdapat dalam kumpulan data **DHF99**.

```
> library(MASS)
> data(DHF99); use(DHF99)
> des()
No. of observations = 300
  Variable      Class           Description
1 houseid       integer         no
2 village       integer         Village
3 education     factor          Educational level
4 containers    integer         # infested vessels
5 viltype       factor          Village type
> summ()
```

```
No. of observations = 300

  Var. name   obs. mean    median   s.d.    min.    max.
1 houseid     300  174.27  154.5    112.44  1       385
2 village     300  48.56   51       32.25   1       105
3 education   300  2.09    1        1.455   1       5
4 containers  299  0.35    0        1.01    0       11
5 viltype     300  1.56    1        0.754   1       3

> summ(containers, by=viltype)

For viltype = rural
  obs. mean   median  s.d.   min.   max.
  179  0.492  0       1.251  0      11

For viltype = urban
  obs. mean   median  s.d.   min.   max.
  72   0.069  0       0.256  0      1

For viltype = slum
  obs. mean   median  s.d.   min.   max.
  48   0.25   0       0.526  0      2
```



Fungsi untuk menampilkan model linear umum binomial negatif adalah `glm.nb`.. Fungsi ini terdapat dalam **MASS** library. Sebagai tambahan, fungsi yang sangat membantu untuk memilih model terbaik terbaik berdasarkan AIC adalah fungsi `step`, yang terdapat dalam **stats** library (a default library loaded on start-up).

```
> model.poisson <- step(glm(containers ~ education +
  viltype, family=poisson, data=.data))
> model.nb <- step(glm.nb(containers ~ education + viltype,
  data=.data))

> coef(model.poisson)
 (Intercept) viltypeurban  viltypeslum
  -0.7100490   -1.9571792   -0.6762454

> coef(model.nb)
 (Intercept) viltypeurban  viltypeslum
  -0.7100490   -1.9571792   -0.6762454
```

Kedua model berakhir dengan 'viltype' selanjutnya akan dipilih. Koefisiennya bernilai sama. Model Poisson memiliki overdispersion signifikan tetapi bukan model binomial negatif. .

```
> poisgof(model.poisson)$p.value
[1] 0.0043878

> poisgof(model.nb)$p.value
[1] 1
```

Nilai AIC untuk model binomial negatif juga lebih baik (lebih kecil) daripada AIC dalam model Poisson.

```
> model.poisson$aic
[1] 505.92

> model.nb$aic
[1] 426.23
```

Akhirnya, perbedaaan utama yang diuji adalah standar error dari koefisien, selang kepercayaan 95% dan P value.

```
> summary(model.poisson)$coefficients
               Estimate Std. Error   z value     Pr(>|z|)
(Intercept)  -0.7100490  0.1066000 -6.660873 2.722059e-11
viltypeurban -1.9571792  0.4597429 -4.257117 2.070800e-05
```

```
viltypeslum  -0.6762454  0.3077286 -2.197538 2.798202e-02

> summary(model.nb)$coefficients
               Estimate Std. Error   z value     Pr(>|z|)
(Intercept)  -0.7100490  0.1731160 -4.101578 4.103414e-05
viltypeurban -1.9571792  0.5255707 -3.723912 1.961591e-04
viltypeslum  -0.6762454  0.4274174 -1.582166 1.136116e-01

> idr.display(model.poisson)
               IDR lower95ci upper95ci P value
viltypeurban 0.141     0.057     0.348   0.000
viltypeslum  0.509     0.278     0.930   0.028

> idr.display(model.nb)
               IDR lower95ci upper95ci P value
viltypeurban 0.141      0.05     0.396   0.000
viltypeslum  0.509      0.22     1.175   0.114
```

Standar error dari model binomial negatif adalah sedikit lebih besar dari standar error model Poisson dalam selang kepercayaan 95% dan P value yang besar. Dari regresi Poisson, kedua komunitas urban dan wilayah kumuh memiliki resiko signifikan lebih rendah untuk nyamuk berkerumun (masing-masing sekitar 14% dan separuh pengurangan). Bagaimanapun, berdasarkan regresi binomial, hanya komunitas urban memiliki resiko signifikan lebih rendah.

## Referensi

Agresti, A. (1996). An Introduction to Categorical Data Analysis. New York: John Wiley and Sons.

Agresti, A. (2002). Categorical Data Analysis. Hoboken, NJ: John Wiley and Sons.

Powers, D.A., Xie, Y. (2000). Statistical Methods for Categorical Data Analysis. San Diego: Academic Press.

Long, J.S. (1997). Regression Models for Categorical and Limited Dependent Variables. Thousand Oakes, CA: Sage Publications.

Vermunt, J.K. (1997). Log-linear Models for Event Histories. Thousand Oakes, CA: Sage Publications.

## Latihan

Gunakan `step` untuk memilih taksiran model incidence dencity terbaik dari **`Montana`** dataset. Uji kesesuaian model Poisson. Hitung perbandingan *incidence density* untuk variable-variabel independen yang signifikan. Tetapkan model regresi binomial negative untuk menguji theta (parameter dispersi) dan standar errornya sebelum menyimpulkan apakah ada bukti terjadinya *overdispersion.*

# BAB 20

# Pengenalan Pemodelan Multi-level

Terdapat banyak sebutan untuk pemodelan multi-level dan semuanya sama serta setiap sebutan memiliki implikasi masing-masing, misalnya pemodelan berhierarki,pemodelan pengaruh campuran, pemodelan dengan pengaruh acak

Dalam kajian epidemiologi, variable sering memiliki hirarki. Misalkan pada pengukuran tekanan darah setiap individu yang dapat memiliki lebih dari satu pengukuran. Pada kasus ini, individu perorangan berada pada hirarki tertinggi daripada setiap pengukuran. Individu, bagaimanapun, memiliki keluarga, semua anggotanya yang mungkin memberikan beberapa variabel bebas, seperti etnis, tempat tinggal dll. Pada dasarnya keluarga biasanya merupakan bagian dari sebuah desa dan sebagaianya. Jadi hirarki dapat berupa Negara, provinsi, kabupaten, desa, keluarga dan pengukuran. Beberapa variabel bebas akan berada pada tingkat pengukuran individu, seperti waktu pengukuran. Beberapa variabel dapat merupakan orde hirarki tertinggi, seperti jenis kelamin dan umur (individu), etnis (keluarga) dan jarak dari ibukota (desa). Variabel independen pada tingkat yang berbeda dari hirarki tidak boleh

diperlakukan dengan cara yang sama. Untuk alas an ini pemodelan multi-level juga disebut pemodelan berhirarki.

Dalam berbagai aspek, pemodelan biasanya berguna untuk menjelaskan hubungan dari variabel-variabel secara informatif dan efektif. Dalam pemodelan sederhana, dimana jumlah kelompoknya tidak besar, katakan *m* kelompok etnis, jumlah parameter yang digunakan untuk menjelaskan pengaruh 'ethnic' adalah *m*-1 karena satu kelompok yang diabaikan digunakan sebagai acuan kelompok. Jika ukuran sampel besar dan *m* kecil maka jumlah parameter yang digunakan tidak terlalu besar. Dengan kata lain, jika ukuran sampel kecil tetapi jumlah kelompoknya besar, misalkan 50 subjek dengan beberapa kali pengukuran tekanan darah, pengelompokan variabel akan memiliki terlalu bnyak level untuk dimasukkan ke dalam model. Untuk melakukan, sebuah nilai rata-rata kelompok dihitung dan anggota individu diperlakukan secara pengaruh acak tanpa parameter. Pada situasi ini, pemodelan multi-level juga disebut pemodelan dengan pengaruh acak. Bagaimanapun, pengaruh acak harus selalu memiliki rataan, yang digunakan untuk mengestimasi keseluruhan pengaruh. Rataan atau keseluruhan pengaruh disebut pengaruh tetap. Dengan penggabungan penagruh tetap dan pengaruh acak dalam model yang sama, multi-level pemodelan juga disebut 'pemodelan pengaruh campuran'.

Pemodelan multi-level relatif lebih baru jika dibandingkan dengan tipe pemodelan yang sama seperti regresi linear dan regresi Poisson. Terdapat variasi metode iterasi numerik untuk perhitungan kofisien dan standar error. Mereka secara umum memberikan estimasi terdekat tetapi standar error, varians dan kovarians yang berbeda. Contoh-contoh pada bab ini hanya terbatas pada fungsi glmmPQL' atau model Generalized Linear Mixed menggunakan Penalized Quasi-Likelihood. Hal tersebut dapat mengatasi seluruh families yang digunakan dalam GLM dengan argumen sama dalam perintah kecuali bentuk tambahan yang mendefinisikan pengaruh acak dan tetap. Pembaca disarankan untuk mengeksplorasi fungsi lainnya seperti lme (linear mixed effects) dan nlme (non-linear mixed effects).

## *Dari analisis bertingkat hingga pemodelan pengaruh acak*

Analisis pengaruh dari penambahan garam pada makanan pada bab 12 telah menghasilkan dua strata, masing-masing dengan jumlah subjek yang relatif

tinggi. Faktor stratifikasi (penambahan garam) memiliki dua tingkatan 'yes/no' tetapi hanya memiliki satu parameter dalam model.

Dalam sebuah pengaturan dengan jumlah strata yang tinggi, masing-masing dengan jumlah daftar yang relatif kecil, termasuk strata individual akan menambah terlalu banyak parameter dalam model, maka mengurangi efisiensi penjelasan (terlalu banyak parameter digunakan untuk menjelaskan kumpulan data kecil). Untuk menyelesaikan masalah ini, setiap strata adalah ditunjukkan oleh rataan strata dan setiap sampel strata diambil secara acak dari himpunan strata dalam populasi. Oleh karena itu, dengan mengabaikan seberapa besar jumlah strata tersebut, akan hanya terdapat dua parameter dari faktor stratifikasi: rataan dan varians (atau standar deviasi)

## *Contoh: Pengukuran Orthodontik*

Sebuah contoh untuk situasi diatas, dan perintah untuk penghitungan tersedia dalam library **nlme**. Pertumbuhan 27 anak (16 laki-laki dan 11 perempuan) telah diperoleh dengan mengukur jarak dari rongga pituitary hingga rongga pterygomaxillary. Pengukuran dibuat pada setiap anak setiap 4 tahun (umur 8, 10, 12 dan 14 tahun).

Data merupakan orde berhirarki. Seorang anak memiliki lebih dari satu catatan pengukuran secara keseluruhan. Catatan atau pengukuran individu berada pada level 1 sementara individu anak berada pada level 2. Umur juga berada pada level 1 karena umur dapat berubah dalam subjek individual, meskipun variasi konstan untuk semua anak. Dengan kata lain, jenis kelamin pada level 2 yang disesuaikan untuk setiap anak.

Setiap anak dapat diumpakan sebagai sebuah stratum. 27 strata diambil secara acak dari populasi strata tak berhingga.

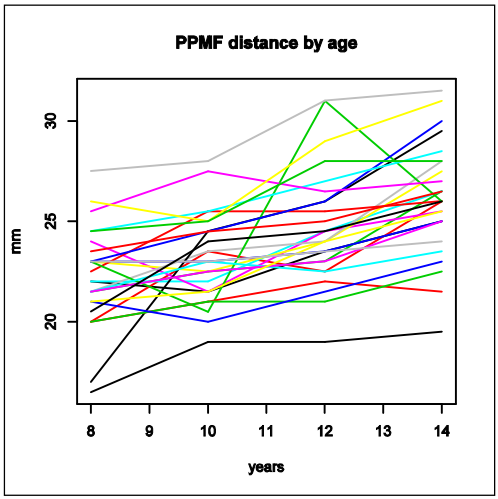
Untuk pemodelan multi-level paling sederhana, koefisien dari 'age', atau slope dari garis regresi, diestimasi sebagai sebagai parameter tunggal, yaitu semua subjek diasumsikan memiliki laju pertumbuhan yang sama. Untuk intercept, model mengestimasi populasi 'mean intercept' dan standar deviasi populasi dari intercept. Intercept memiliki 'random effects' (untuk individu anak) sedangkan slope memiliki 'fixed effect' untuk keseluruhan kelompok. Kombinasi dua tipe pengaruh random dan pengaruh tetap, model sering dinamakan 'mixed model'.r

Setelah library **nlme** dibuka, kumpulan data **Orthodont** dapat digunakan. Hati-hati karena beberapa nama variabel pada frame data berikut dimulai dengan kasus diatas.

```
> zap()
> library(MASS)     # For the glmmPQL command
> library(nlme)     # For the example dataset
> data(Orthodont)
> .data <- as.data.frame(Orthodont)
> use(.data);  des()

No. of observations =108
  Variable      Class           Description
1 distance      numeric         distance
2 age           numeric         age
3 Subject       factor          Subject
4 Sex           factor          Sex

> summ()
  Var. name Obs.   mean   median  s.d.   min.   max.
1 distance  108    24.02  23.75   2.93   16.5   31.5
2 age       108    11     11      2.25   8      14
3 Subject   108    14     14      7.825  1      27
4 Sex       108    1.407  1       0.494  1      2
```

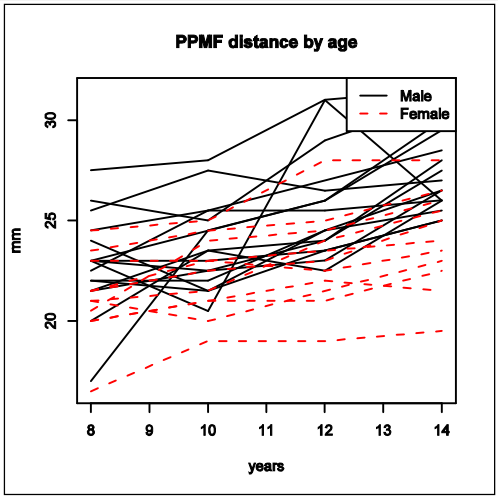
Sebuah plot selanjutnya digunakan untuk menggambarkan data. *Epicalc* memiliki fungsi yang disebut *followup.plot*, dimana plot adalah keluaran untuk masing-masing subjek dalam keseluruhan waktu.

```
> followup.plot(id=Subject, time=age, outcome=distance,
    line.col="multicolor")
> title(main="PPMF distance by age", ylab="mm",
  xlab="years")
```




Untuk melihat apakah terdapat perbedaan jenis kelamin, kita gantikan argumen 'lines' dengan argumen 'by' dalam perintah.

```
> followup.plot(id=Subject,time=age,outcome=distance,by=Sex)
> title(main="PPMF distance by age", ylab="mm",
  xlab="years")
```



Pada kedua plot jelas terlihat bahwa seiring umur meningkat, jarak juga mengalami peningkatan. Laju individual bersilangan menuju jumlah yang pasti. Dengan kata lain, garis tertinggi dan terendah cukup konsisten. Laki-laki secara umum memiliki jarak rongga pituitary ke rongga pterygomaxillary yang lebih besar.

## Model intercept acak (*Random intercepts model*)

Untuk pemodelan multi-level, setiap subjek dikatakan sebagai stratum . Untuk model pertama ini , slope diharuskan sama. Terdapat 27 intercept; terlalu banyak untuk menjadikan mereka parameter. Sehingga sebuah intercept rataan dihitung dan sisanya digunakan sebagai pengaruh acak.

```
> model0 <- glmmPQL(distance ~ age, random = ~1 | Subject,
    data = .data, family = gaussian)
```

Perintah diatas menciptakan model generalized linear multi-level (glmm) menggunakan metode iterasi Penalized Quasi-Likelihood (PQL). Variabel terikat adalah 'distance'. Variabel bebas adalah 'age', yang memiliki pengaruh tetap(untuk semua subjek). Pengaruh acak (yang ditandai oleh kata 'random') merupakan konstanta bernilai 1. Level paling atas dari model (ikuti tanda '|') adalah 'Subject' karena subjek yang sama memiliki 4 pengulangan pengukuran. Dengan kata lain, 'Subject' merupakan level tertinggi. Perintah glmmPQL menangani argumen 'family' dalam model dengan cara yang sama dengan perintah glm. Karena error diasumsikan berdistribusi normal, keluarga dispesifikasikan sebagai 'gaussian'.

```
> summary(model0)
Linear mixed-effects model fit by maximum likelihood
 Data: .data
  AIC BIC logLik
   NA  NA     NA

Random effects:
 Formula: ~1 | Subject
        (Intercept) Residual
StdDev:    2.072142 1.422728

Variance function:
 Structure: fixed weights
```

```
 Formula: ~invwt
Fixed effects: distance ~ age
               Value Std.Error DF  t-value p-value
(Intercept) 16.761111 0.8020244 80 20.89851       0
age          0.660185 0.0617993 80 10.68272       0
 Correlation:
    (Intr)
age -0.848

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-3.68695131 -0.53862941 -0.01232442  0.49100161  3.74701484

Number of Observations: 108
Number of Groups: 27
```

Nilai 'AIC' dan 'BIC' diperoleh dari 'logLik', log likelihood. Mereka akan digunakan untuk membandingkan level kesesuaian dengan model lainnya dengan menggunakan data yang sama dan metode iterasi yang sama pula. Ingat bahwa AIC sama dengan -2×logLik + 2×*npar* dan BIC sama dengan to -2×logLik + log(*n*)×*npar*, dimana *npar* merupakan jumlah parameter dalam model (dalam model ini, empat; yaitu, intercept standar deviasi dan residual, yang merupakan pengaruh acak, dan koefisien dari intercept tetap dan pengaruh tetap umur) dan *n* adalah jumlah observasi (108).

Pengaruh acak menunjukkan diri mereka sebagai standar deviasi error. Terdapat dua bagian bagian error. Bagian pertama merupakan standar deviasi dari perbedaan antara intercept tetap dan intercept dari subjek individual. Bagian kedua adalah standar deviasi residual atau perbedaan antara nilai akhir prediksi dan nilai diamati untuk setiap subjek. Tidak terdapat koefisien untuk bentuk pengaruh acak ini karena rataannya mendekati nol. Hal ini disebabkan karena mereka diasumsikan berdistribusi normal.

Bagian tetap dari ringkasan, sama halnya dengan model regresi sederhana, terdiri dari koefisien dan standar error. Koefisien intercept yaitu 16.76. Hal ini berarti bahwa pada rata-rata usia 0 tahun, jarak PPMF untuk anak-anak diharapkan sebesar 16.67 mm. Koefisien umur adalah 0.66. Hal ini berarti untuk setiap ulangtahun, rata-rata anak diharapkan memiliki 0.66 mm panjang jarak PPMF. Koefisien ini signifikan secara statistik sebagai standar error yang relatif kecil, hasil dari t-value besar dan P value yang bernilai kecil. Residual standardised dalam kelompok (atau dalam anak) adalah berdistribusi dengan

derajat kesimetrian karena median mendekati nol serta kuartil tertinggi dan terendah relatif berjarak sama dari median, yaitu minimum dan maksimum. Pada akhirnya, model menunjukkan bahwa terdapat 27 anak memberikan 108 observasi.

## *Model atribut dan grafik*

Model memiliki banyak atribut didalamnya. Kita akan mengerjakan hanya beberapa saja.

```
> attributes(model0)
$names
 [1] "modelStruct"  "dims"      "contrasts"    "coefficients"
 [5] "varFix"       "sigma"     "apVar"        "logLik"
 [9] "numIter"      "groups"    "call"         "terms"
[13] "method"       "fitted"    "residuals"    "fixDF"
[17] "na.action"    "data"      "family"

$class
[1] "glmmPQL" "lme"
```

Atribut paling penting adalah koefisien.

```
> coef(model0)
$fixed
(Intercept)         age
 16.7611111   0.6601852
$random
$random$Subject
    (Intercept)
M16  -0.9152788
M05  -0.9152788
M02  -0.5798146
================
================
F04   0.7620421
F11   2.1038989
```

Ada dua macam bagian koefisien: bagian tetap dan bagian acak. Bagian tetap yang ditunjukkan pada summary merupakan rata-rata untuk keseluruhan 27 strata (anak-anak). Fixed intercept sebesar is 16.761111, yang berarti bahwa jarak estimasi (rata-rata) pada kelahiran (saat usia 0 tahun) adalah sebesar

16.76 mm. Untuk setiap kenaikan umur, jarak PPMF meningkat mendekati dua-pertiga millimeter (0.66). Bagian kedua atau bagian acak menyatakan 'random intercepts only' karena tidak terdapat variabel pada bagian ini seperti yang ditunjukkan oleh 'random ~ 1'. Ada 27 (koefisien tambahan untuk) intercept, satu untuk setiap anak. Untuk anak pertama (M16) yang memiliki intercept acak negatif , atau jarak permulaan, intercept rataan untuk bagian tetap (16.76) harus dikurangi 0.9152788. Anak kedua (M05) berbagi intercept yang sama. Secara keseluruhan, interval intercept acak antara -4.940849 (F10) hingga +4.899434 (M10).

Terdapat banyak atribut lainnya yang dapat dipergunakan. Salah satunya adalah 'fitted(model0)', yang mengandung nilai dugaan atau prediksi untuk setiap titik observasi.

```
> model0$fitted
     fixed Subject
1   22.043  25.377
2   23.363  26.697
3   24.683  28.017
4   26.004  29.338
5   22.043  21.463
6   23.363  22.783
7   24.683  24.104
8   26.004  25.424
==== Up to 108th person ==========
```

Ada dua kolom nilai dugaan: bagian tetap (rata-rata setiap titik waktu) dan acak (oleh Subject). Pada kenyataannya, bagian tetap hanya memiliki empat nilai untuk memprediksi nilai rata-rata setiap nilai umur.

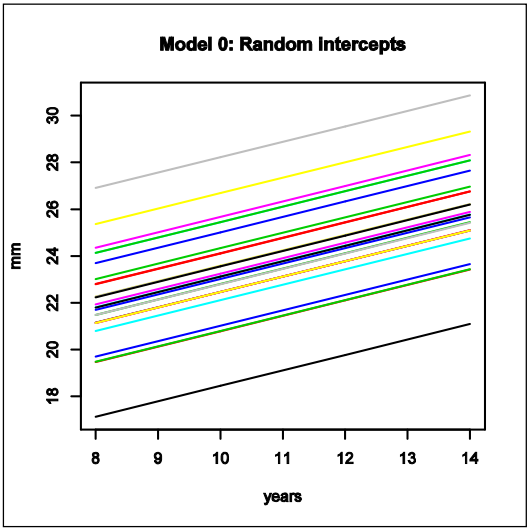
```
> tab1(model0$fitted[, 1])
[model0$fitted  1 :
                  Frequency Percent
22.0425925925926        27      25
23.3629629629630        27      25
24.6833333333333        27      25
26.0037037037037        27      25
  Total                108     100
```

Setiap nilai memiliki 27 catatan pengulangan. Dengan kata lain, hanya terdapat empat bentuk dari pengaruh tetap , masing-masing diperoleh dari keseluruhan subjek (27). Komponen kedua adalah memprediksi nilai intercept untuk setiap

subjek, yang saling berbeda antara anak yang satu dengan yang lain.

```
> followup.plot(id=Subject, time=age,
  outcome=fitted(model0),
    line.col="multicolor")

> title(main="Model 0: random intercepts", ylab="mm",
    xlab="years")
```

Koordinat X untuk setiap garis merupakan umur setiap anak. Koordinat terikat Y merupakan nilai dugaan untuk jarak PPMF. Ingat bahwa terdapat dua kolom untuk nilai dugaan (untuk pengaru tetap dan , acak). Plot menggunakan kolom kedua, dimana merupakan nilai dugaan untuk setiap anak (pengaruh aacak). Warna beragam sesuai perintah 'Subject'.



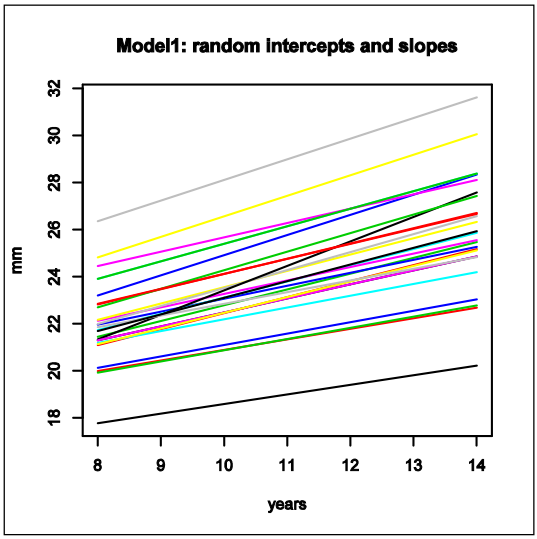

Model terdiri dari koefisien slope, dimasukkan intercept hanya sebagai variabel acak. Model selanjutnya menunjukkan pengaruh umur menjadi acak terhadap nilai rata-rata.

## Model dengan slope acak

```
> model1 <- glmmPQL(distance ~ age, random = ~age | Subject,
    data = .data, family = gaussian)

> summary(model1)
Linear mixed-effects model fit by maximum likelihood
 Data: .data
   AIC BIC logLik
    NA  NA     NA

Random effects:
 Formula: ~age | Subject
 Structure: General positive-definite, Log-Cholesky
  parametrization
            StdDev    Corr
(Intercept) 2.2023778 (Intr)
age         0.2152392 -0.585
Residual    1.3103646
Variance function:
 Structure: fixed weights
 Formula: ~invwt
Fixed effects: distance ~ age
                Value Std.Error DF   t-value p-value
(Intercept) 16.761111 0.7689227 80 21.798174       0
age          0.660185 0.0706254 80  9.347699       0
 Correlation:
    (Intr)
age -0.849

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-3.30002923 -0.48692999  0.00739127  0.48148182  3.92211226

Number of Observations: 108
Number of Groups: 27
```

Sama halnya dengan 'model0', grafik dapat diplotkan dengan menggunakan perintah berikut,

```
> followup.plot(id=Subject, time=age, outcome=fitted(glmm1),
    line.col="multicolor")

> title(main="Model1: random intercepts and slopes",
    ylab="mm", xlab="years")
```




Model 'model0' setara dengan analisis bertingkat tanpa interaksi sedangkan 'model1' setara dengan penetapan bentuk interaksi. Model kedua menunjukkan bahwa setiap anak memiliki jarak dasar tersendiri (intercept) serta laju pertumbuhan mereka sendiri.

Grafik tersebut menunjukkan slope yang berbeda untuk subjek yang berbeda pula. Slope atau kemiringan sekarang merupakan pengaruh acak serta pengaruh tetap.

Pada bagian pengaruh acak, umur memiliki standar deviasi 0.215 mm yang relative lebih kecil jika dibandingkan dengan keacakan intercept (2.2 mm) dan residual (1.3 mm). Variasi karena perbedaan dalam laju pertumbuhan jarak PPMF antara subyek adalah kecil dibandingkan dengan variasi dalam baseline dan tingkat pertumbuhan rata-rata. Korelasi antara usia dan intercept adalah negatif (-0,585) dalam pengaruh acak menunjukkan bahwa kemiringan dari subyek cenderung datar sebagai peningkatan level intercept –Y.

Koefisien pengaruh acak untuk kemiringan dan umur tidak berbeda dengan 'model0'. Pada kenyataan koefisien adalah sama dengan glm sederhana.

```
> summary(glm(distance ~ age, family=gaussian))
```

Standar error dari glm jauh lebih tinggi daripada model multi-level. Model advanced ini memperbaiki ketepatan perkiraan. Pada contoh ini 'model1' memiliki standar error yang lebih luas disbanding 'model0'. Saat pengaruh umur dipartisi secara individu, pengaruh usia keseluruhan mengurangi presisi.

Kita punya variabel independent lainnya yaitu 'Sex'. Hal ini akan menarik untuk memeriksa apakah anak laki-laki memiliki jarak lebih besar daripada anak perempuan dan apakah tingkat pertumbuhan yang berbeda antara kedua jenis kelamin.

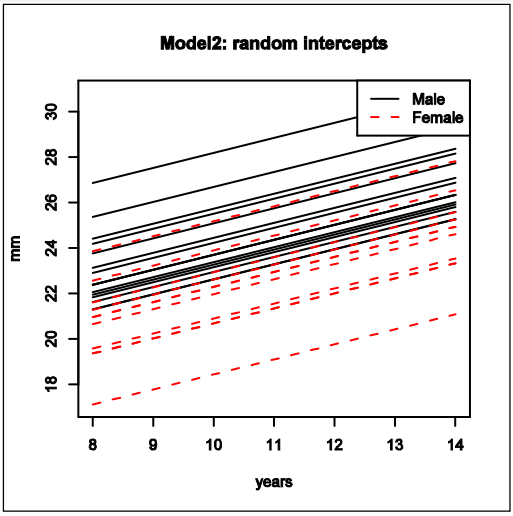
```
> model2 <- glmmPQL(distance ~ age + Sex, random = ~1 |
    Subject, data = .data, family = gaussian)

> summary(model2)
Linear mixed-effects model fit by maximum likelihood
 Data: .data
  AIC BIC logLik
   NA  NA     NA

Random effects:
 Formula: ~1 | Subject
        (Intercept) Residual
StdDev:    1.730079 1.422728

Variance function:
 Structure: fixed weights
 Formula: ~invwt
Fixed effects: distance ~ age + Sex
                Value Std.Error DF   t-value p-value
(Intercept) 17.706713 0.8315459 80 21.293729  0.0000
age          0.660185 0.0620929 80 10.632212  0.0000
SexFemale   -2.321023 0.7430668 25 -3.123572  0.0045
========= Remaining parts of output omitted ========
```

'Sex' diperkenalkan sebagai pengaruh tetap murni. Bahkan, tidak dapat menjadi

Garis pertumbuhan sekarang sudah dipisahkan oleh 'Sex'.

```
> followup.plot(id=Subject, time=age,
  outcome=fitted(model2),
    by=Sex)

> title(main="Model2: random intercepts", ylab="mm",
```

```
  xlab="years")
```




Jelas terlihat bahwa garis untuk laki-laki cenderung berada dibagian atas plot sedangkan perempuan cenderung berada dibagian bawah

Untuk menguji apakah laju berbeda antara dua jenis kelamin, akan diperkenalkan sebuah bentuk interaksi antara usia dan jenis kelamin

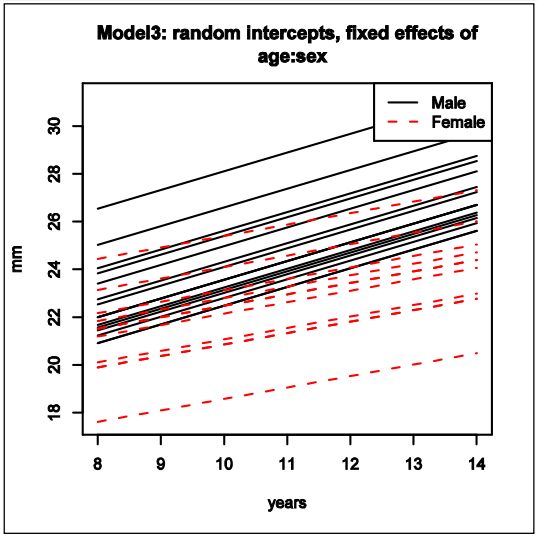
```
> model3 <- glmmPQL(distance ~ age*Sex, random = ~1 |
  Subject, data = .data, family = gaussian)
> summary(model3)
Linear mixed-effects model fit by maximum likelihood
 Data: .data
  AIC BIC logLik
   NA  NA     NA

Random effects:
 Formula: ~1 | Subject
        (Intercept) Residual
StdDev:    1.740851 1.369159

Variance function:
 Structure: fixed weights
 Formula: ~invwt
Fixed effects: distance ~ age * Sex
                  Value Std.Error DF   t-value p-value
```

```
(Intercept)    16.340625 0.9814310 79 16.649795  0.0000
age             0.784375 0.0779963 79 10.056564  0.0000
SexFemale       1.032102 1.5376069 25  0.671239  0.5082
age:SexFemale -0.304830 0.1221968 79 -2.494580  0.0147
========= Remaining parts of output omitted ========
```

Interaksi antara umur dan jenis kelamin adalah signifikan. Koefisien pengaruh utama dari 'Female' sebesar 1.03, menunjukkan bahwa dengan menggunakan asumsi pertumbuhan linear, pada kelahiran (dimana usia 0 tahun), perempuan memiliki jarak PPMF rata-rata lebih lama 1.03mm dibandingkan dengan anak laki-laki.

Koefisien interaksi sebesar -0.30483 menunjukkan bahwa untuk setiap kenaikan satu tahun, perempuan akan memiliki jarak PPMF rata-rata lebih pendek 0.3mm dibandingkan dengan anak laki-laki. Dengan kata lain, perempuan memiliki jarak PPMF lebih pendek dan tingkat pertumbuhan yang lebih kecil.

```
> followup.plot(id=Subject, time=age,
  outcome=fitted(model3),
    by=Sex)

> title(main="Model3: random intercepts, fixed effects of
    age:sex", ylab="mm", xlab="years")
```



Sebagai kesimpulan, masing-masing anak memiliki jarak dasar PPMF yang berbeda. Anak perempuan cenderung memiliki jarak PPMF lebih tinggi pada saat lahir. Namun, anak laki-laki memiliki tingkat pertumbuhan yang lebih cepat.

## *Catatan untuk paket lme4*

Pemodelan pengaruh campuran adalah subjek yang bergerak cepat. Telah diperkenalkan sebuah paket baru dalam **R** version 2.4.1 yang disebut **lme4**. Paket berisi fungsi yang disebut lmer, yang lebih efisien daripada fungsi glmmPQL dalam paket **MASS** dan dapat menampung lebih banyak type tersarang yang lebih rumit. Sebagai contoh, analisis kunjungan klinis dapat tersarang secara simultan oleh pasien dan dokter. Sementara fitur ini lebih canggih daripada apa yang telah ditunjukkan dalam bab ini, paket baru ini memberikan hasil yang sama untuk tersarang sederhana. Namun hal itu masih dalam tahap percobaan. Sebagai contoh, nilai-nilai dugaan tidak dapat dengan mudah diperoleh. Ketika paket ini sepenuhnya dikembangkan maka mungkin dapat mengganti isi dalam bab ini.

## Referensi

Pinheiro, J. C. & Bates, D. M. 2000. Mixed-Effects Models in S and S-PLUS. New York: Springer.

Raudenbush, S. W. & Bryk, A. S. 2002. Hierarchical Linear Models: Applications and Data Analysis Methods. 2nd ed. Thousand Oaks CA: Sage.

## Latihan

Himpunan data **Bang** terdiri dari sekumpulan data dari 'Survey Fertility Bangladesh 1988 '.

```
> zap()
> data(Bang)
> use(Bang)
> label.var(woman, "woman ID")

# Response variable
> label.var(user, "current contraceptive use")
> label.var(age_mean, "age(yr) centred around mean")
> living.children <- factor(living.children)
> label.var(living.children, "No. of children living")
```

Soal1.
Gunakan glmmPQL untuk menghitung pengaruh dari sejumlah anak yang hidup ,usia dan tinggal didaerah perkotaan dengan peluang penggunaan kontrasepsi dikalangan wanita. Hitung odd ratio dengan .interval kepercayaan 95%

Soal2.
Apakah sejumlah anak yang hidup memiliki hubungan dosis respon linear terhadap penggunaan kontrasepsi?

Soal 3.

Apakah usia dapat merupakan pengaruh acak?

Soal4.
Apakah usia memiliki pengaruh yang sama antara perempuan perkotaan dan pedesaan dalam penggunaan kontrasepsi?

# BAB 21

# Analisis Survival

Dalam studi kohort, seseorang ditindaklanjuti dari waktu permulaan hingga akhir penelitian atau hingga waktu tindak lanjut telah telah diakhiri oleh outcome, mana yang lebih dahulu. Durasi event-free merupakan outcome yang penting. Untuk kejadian yang tidak diinginkan, outcome yang diharapkan merupakan durasi event-free yang lebih lama.

Untuk subjek dengan kejadian yang terjadi sebelum akhir penelitian, total waktu durasinya diketahui. Untuk subjek yang dilakukan dengan waktu berakhir tanpa kejadian apapun, status terakhir disebut 'censored' karena durasi waktu kejadian sebenarnya tidak diketahui atau 'censored' oleh penelitian. Meskipun pada akhirnya variabel dependen untuk setiap sampel atau subjek terdiri dari 'time' dan 'status'. Secara matematika, status 1 jika terjadi peristiwa dan 0 untuk lainnya.

## *Contoh: Usia Pernikahan*

Sebuah data manajemen workshop telah dilakukan pada tahun 1997. Setiap orang dari 27 peserta diminta untuk memberikan informasi personal tentang

jenis kelamin, tahun lahir, tingkat pendidikan, status pernikahan serta tahun menikah (untuk mereka yang telah menikah). Tujuan analisis ini adalah untuk menggunakan metode analisis survival untuk menguji himpunan data berikut.

```
> library(survival)
> data(Marryage)
> use(Marryage)
> des()

No. of observations =27

  Variable      Class           Description
1 id            integer
2 sex           factor
3 birthyr       integer         year of birth
4 educ          factor          level of eduction
5 marital       factor          marital status
6 maryr         integer         year of marriage
7 endyr         integer         year of analysis

> summ()

No. of observations = 27
  Var. name Obs.  mean    median  s.d.   min.   max.
1 id        27    14      14      7.94   1      27
2 sex       27    1.667   2       0.48   1      2
3 birthyr   27    1962.15 1963    6.11   1952   1972
4 educ      27    1.519   2       0.509  1      2
5 marital   27    1.593   2       0.501  1      2
6 maryr     16    1987.56 1988    5.18   1979   1995
7 endyr     27    1997    1997    0      1997   1997
```

Gunakan perintah berikut untuk melihat kode variabel faktor :

```
> codebook()

id       :
  obs. mean   median  s.d.   min.   max.
  27   14     14      7.94   1      27

 ==================
sex      :
Label table: sexlab
       code Frequency Percent
male      1         9    33.3
```

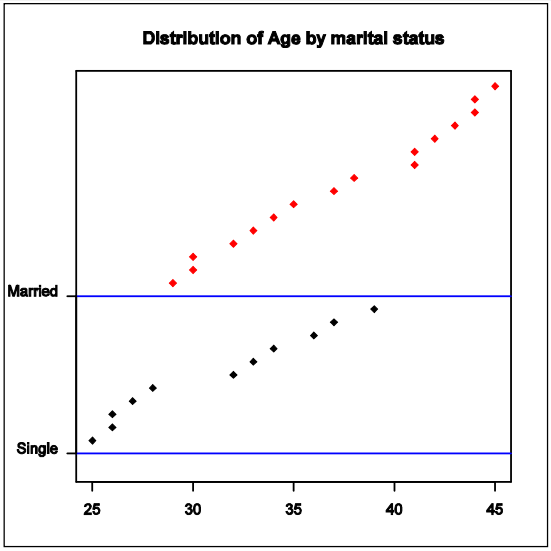
```
female    2        18    66.7

 ==================
birthyr          :       year of birth
  obs. mean     median  s.d.   min.   max.
  27   1962.148 1963    6.11   1952   1972

 ==================
educ     :       level of education
Label table: educlab
          code Frequency Percent
bach-        2        13    48.1
>bachelor    3        14    51.9

 ==================
marital          :       marital status
Label table: marlab
        code Frequency Percent
Single     1        11    40.7
Married    2        16    59.3

 ==================
maryr    :       year of marriage
  obs. mean     median  s.d.   min.   max.
  16   1987.562 1988    5.18   1979   1995

 ==================
endyr    :       year of analysis
  obs. mean   median  s.d.   min.   max.
  27   1997   1997    0      1997   1997
 ==================
```

Ingat bahwa kode sesungguhnya untuk variabel 'educ' adalah 2 = bach-, 3 = >bachelor, seperti yang ditunjukkan pada output perintah *codebook*. Cara ini menunjukkan bagaimana kode didefinisikan dalam program entri data original, dan label tabel yang berasosiasi dengan setiap variabel kategori tetap digunakan dalam data. Bagaimanapun pada output fungsi *summ*, kode numerik untuk 'educ' ditampilkan sebagai 1 (bach-) dan 2 (>bachelor). Penyimpangan ini terjadi karena tidak adanya pengklasifikasian level variabel faktor pada output perintah *summ*. Saat R mengubah sesuatu menjadi sebuah faktor, level pertama akan selalu memiliki kode 1. Kode numerik ini tidak membingungkan dengan skema pengkodean original. Pada kenyataannya, kode

tersebut hanya digunakan selama entri data original tidak pernah digunakan dalam analisis data.

Variabel 'endyr', fixed at 1997, digunakan untuk penghitungan usia dan usia pada saat menikah.

```
> age <- endyr - birthyr
> label.var(age, "Age")
> summ(age, by = marital)
For marital = Single
  Obs.   mean   median  s.d.   min.   max.
  11     31.18  32      4.996  25     39

For marital = Married
  Obs.   mean   median  s.d.   min.   max.
  16     37.38  37.5    5.596  29     45
```




Ada 16 orang peserta yang telah menikah (59%). Lebih jelasnya peserta yang telah menikah memiliki usia lebih tua dari yang belum menikah.

```
> age.marr <- maryr - birthyr
> label.var(age.marr, "Age at marriage")
> summ(.data[,8:9])

No. of observations = 27

  Var. name  obs. mean   median  s.d.   min.   max.
1 age        27   34.85  34      6.11   25     45
2 age.marr   16   27.94  27.5    2.77   25     36
```

Diantara 16 peserta yang telah menikah, usia rata-rata waktu menikah adalah 27.94 tahun.

Inti dari keseluruhan analisis survival ini berhubungan dengan “time-to-event”. Pada himpunan data berikut kita menggunakan usia sebagai variabel waktu dan pernikahan sebagai kejadiannya. Pada kebanyakan studi epidemiologi 'time' biasanya dianggap sebagai durasi tindak lanjut (follow up) dan kejadian biasanya terjadi pada waktu yang tidak diinginkan, seperti kematian atau penyakit yang kambuh tiba-tiba. Data ini berasal dari survei cross-sectional, sedangkan kebanyakan data untuk analsis survival berasal dari studi tindak lanjut (follow up studies). Bagaimanapun, prosedur yang digunakan untuk dataset sederhana ini dapat diaplikasikan untuk tipe data survival lainnya.

## Objek Survival dalam R

Library **survival** berisi semua fungsi yang diperlukan untuk menganalisis tipe data survival. Untuk menganalisis data ini, kita perlu membuat sebuah objek dari kelas *Surv*, yang mengombinasikan informasi data dan status pada objek tunggal.variabel status harus numerik atau logical. Jika numerik, ada dua pilihan. Nilai harus berupa 0=censored dan 1=event, atau 1=censored dan 2=event. Jika logical, FALSE=censored dan TRUE=event. Dalam himpunan data **`Marryage`**, 'marital' merupakan faktor dan harus dikonversikan kesalah satu bentuk yang disebutkan diatas. Kita akan memilih bentuk logical, tetapi secara acak.

```
> married <- marital == "Married"
> time <- ifelse(married, age.marr, age)
```

Perhatikan bahwa waktu untuk sampel yang telah menikah dan belum menikah diperoleh secara berbeda. Untuk sampel yang telah menikah, kita tahu bahwa

durasi waktu merupakan usia saat mereka menikah. Waktu survival mereka berhenti pada tahun terjadi pernikahan. Untuk subjek yang belum menikah, kita tidak mengetahui durasi waktu mereka. Jadi usia mereka saat ini digunakan sebagai gantinya. Objek survival pernikahan sekarang dapat dibuat dan dibandingkan dengan variabel lainnya.

```
> (surv.marr <- Surv(time, married))
[1] 26  26  29  25+ 26  26+ 28  28  28  36+ 36  39+ 29  33+
[15] 25  31  27  34+ 37+ 26  27+ 25  27  26+ 28+ 30  32+

> head(data.frame(age, age.marr, married, surv.marr))
  age age.marr married surv
1  44       26    TRUE   26
2  43       26    TRUE   26
3  45       29    TRUE   29
4  25       NA   FALSE   25+
5  37       26    TRUE   26
6  26       NA   FALSE   26+
```

Untuk tiga sampel pertama dan sampel kelima dimana kesemua mereka telah menikah, nilai '`surv.marr`' sama dengan usia saat mereka menikah. Untuk sampel keempat dan keenam, nilainya sama dengan usia mereka saat ini. Tanda positif menunjukkan bahwa 'time' sebenarnya diluar nilai nilai tersebut tetapi telah disensor. Para peserta ini belum menikah pada saat workshop.

Untuk eksplorasi lebih lanjut, subset variabel yang diurutkan berdasarkan 'time' ditampilkan dengan menggunakan perintah berikut.

```
> cbind(age, sex, age.marr, married,
  surv.marr)[order(time),]
      age sex age.marr married time status
 [1,]  25   1       NA       0   25      0
 [2,]  32   2       25       1   25      1
 [3,]  29   1       25       1   25      1
 [4,]  44   1       26       1   26      1
 [5,]  43   2       26       1   26      1
 [6,]  37   2       26       1   26      1
 [7,]  26   2       NA       0   26      0
 [8,]  34   1       26       1   26      1

======== subsequent lines omitted ========
```

Objek '`Surv`' terdiri dari 'time' dan 'status'. Orang pertama, laki-laki single berusia 25 tahun. Waktunya adalah 25 dan statusnya adalah 0 dan kejadiannya

disensor. Orang kedua merupakan wanita 32 tahun yang menikah pada usia 25 tahun, maka 25 adalah waktunya. Kejadian (pernikahan) telah terjadi sehingga statusnya sama dengan 1, dll.

## Tabel Kehidupan

Tabel kehidupan merupakan tabulasi survival, kejadian dan peluang survival dari waktu ke waktu. Metode klasik untuk analsis ini dalam populasi umum telah dikembangkan secara baik selama berabad-abad. Secara umum, metode ini melibatkan perhitungan peluang survival kumulatif yang dihasilkan dari peluang survival pada setiap langkah. Untuk dataset sederhana, keseluruhan table kehidupan dapat diperoleh dari:

```
> fit <- survfit(surv.marr)
> summary(fit, censor=TRUE)
Call: survfit(formula = surv.marr)
 time n.risk n.event survival std.err lower95CI upper95CI
   25     27       2    0.926  0.0504     0.832     1.000
   26     24       4    0.772  0.0820     0.627     0.950
   27     18       2    0.686  0.0926     0.526     0.894
   28     15       3    0.549  0.1025     0.380     0.791
   29     11       2    0.449  0.1054     0.283     0.711
   30      9       1    0.399  0.1048     0.238     0.668
   31      8       1    0.349  0.1029     0.196     0.622
   32      7       0    0.349  0.1029     0.196     0.622
   33      6       0    0.349  0.1029     0.196     0.622
   34      5       0    0.349  0.1029     0.196     0.622
   36      4       1    0.262  0.1080     0.117     0.588
   37      2       0    0.262  0.1080     0.117     0.588
   39      1       0    0.262  0.1080     0.117     0.588
```

Baris pertama output menyatakan bahwa pada waktu 25 (saat semua peserta berusia 25 tahun), ada 27 sampel/subjek, dua diantaranya menikah pada saat itu. Peluang survival (peluang menikah pada usia ini) dihitung menggunakan (27-2)/27 = 0.926. Pada kenyataannya ada seseorang berusia 25 tahun yang tidak ditampilkan. Orang ini disensor (tidak menikah) sehingga termasuk dalam baris ini tapi tidak di baris berikutnya.

Pada baris kedua, ada 24 peserta yang tersisa yang telah mencapai atau melalui

ulang tahun ke 26 ( menuju 27, 2 kejadian dan 1 disensor pada penghujung tahun ke 25). Pada saat ini, 4 kejadian terjadi dan karena baris ketiga menyatakan bahwa hanya 18 orang tersisa pada titik waktu berikutnya, 2 subjek harus telah disensor. Peluang survival pada waktu 26 adalah (24 - 4)/24 = 0.833. Saat mengalikan nilai ini dengan peluang sebelumnya pada abris pertama, peluang kumulatif adalah (25/27) x (20/24) = 0.772. Perhitungan peluang survival kumulatif ini berlanjut dengan cara serupa hingga akhir himpunan data. Ingat bahwa pada titik waktu 32, 33, 34, 37 dan 39 tahun, tidak ada kejadian yang terjadi (*n* event = 0). Maka peluangnya tidak berubah.

Tabel kehidupan Kaplan-Meier diatas merupakan modifikasi dari bentuk metode demografi klasik dimana interval waktunya tetap (biasanya pada setiap 5 tahun usia) dan penyesuaian untuk informasi yang tidak komplit dari *exact time* diambil dari account.

## Kurva Kaplan-Meier

Ringakasan objek survival berikut mengungkapkan banyak sub-objek.

```
> km1 <- summary(fit, censor=T)
> attributes(km1)
$names
[1] "surv"     "time"     "n.risk"   "n.event"  "conf.int"
  "std.err"  "lower"    "upper"    "call"
$class
[1] "summary.survfit"
```

Kita dapat menggunakan objek '`km1`' ini untuk memplotkan 'time' vs 'surv', dan menghasilkan plot stepped line yang disebut kurva survival atau kurva Kaplan-Meier.

```
> plot(km1$time, km1$surv, type="s")
```

Jika '`xlim=c(25, 40)`' ditambahkan pada perintah , kurva akan menjadi sangat serupa dengan yang dihasilkan oleh perintah standar.

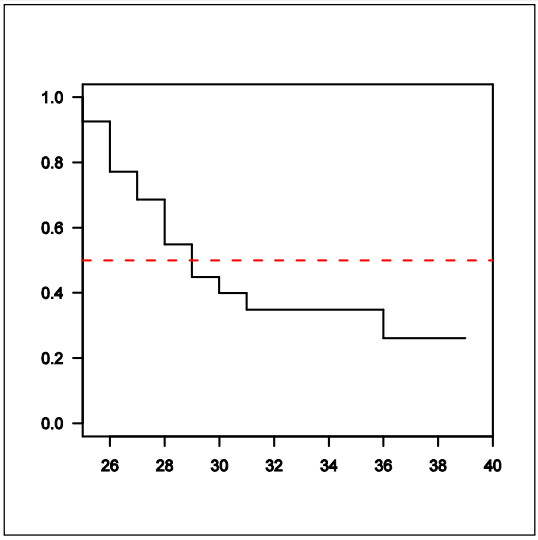
```
> plot(fit, xlim=c(25, 40))
```

Saat hanya ada satu kurva diplotkan, dua garis interval kepercayaan 95% dan tanda waktu untuk subjek yang disensor juga dimasukkan kedalam plot. Untuk lebih jelas, plot tersebut dapat diatur menjadi FALSE.

```
> plot(fit, conf.int=F, mark.time=F, xlim=c(25, 38), las=1)
```

Sumbu vertical merupakan peluang survival dan sumbu horizontal adalah waktu. Jika sebuah garis horizontal ditarik pada peluang 50%, itu akan melintasi kurva survival pada titik median waktu survival. Jika kurang dari setengah subjek pernah mengalami kejadian tersebut maka median waktu survival tidak terdefinisi.

```
> abline(h=.5, lty=2, col="red")
```




Dalam himpunan data ini, median waktu survival (usia pada waktu menikah) adalah 29 tahun. Nilai ini akan ditunjukkan saat diketik objek '`fit`'.

```
> fit
Call: survfit(formula = surv.marr)

      n  events  median 0.95LCL 0.95UCL
     27      16      29      27      36
```

The numbers at risk at various time points can also be displayed on the plot.

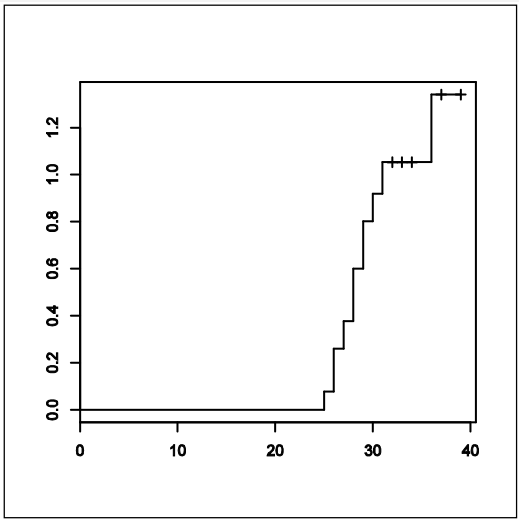
```
> stimes <- seq(from=20, to=40, by=5)
> sfit <- summary(fit, times = stimes)
> sfit
Call: survfit(formula = surv.marr)
```

```
 time n.risk n.event survival std.err lower95%CI upper95% CI
   20     27       0    1.000  0.0000      1.000       1.000
   25     27       2    0.926  0.0504      0.735       0.981
   30      9      12    0.399  0.1048      0.200       0.592
   35      4       1    0.349  0.1029      0.162       0.545

> n.risk <- sfit$n.risk
> n.time <- sfit$time
> mtext(n.risk, side=1, line=2, at=stimes, cex=0.8)
```

## Laju Hazard Kumulatif

Laju hazard merupakan proporsi kegagalan per unit waktu. Dalam studi epidemiologi, laju hazard dapat bervariasi dari waktu ke wakt. Secara grafik, lebih baik menggunakan laju kumulatif karena relative lebih mudah untuk melihat perubahan laju dengan kemiringan kurva kumulatif.

```
> plot(fit, conf.int=FALSE, fun="cumhaz")
```

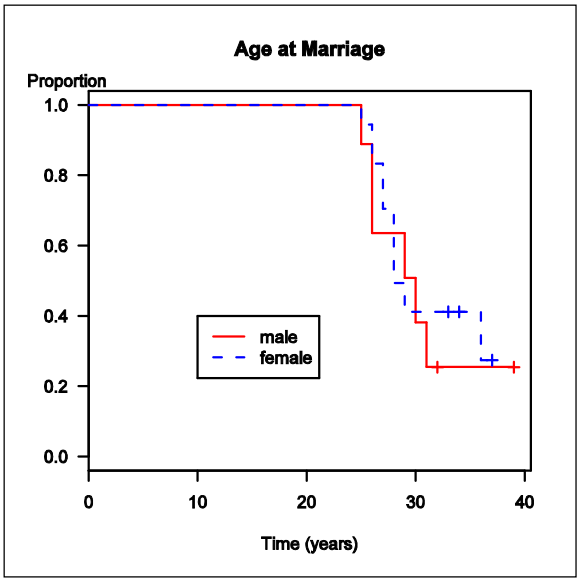



Dalam 25 tahun pertama, lereng datar karena tidak adanya peristiwa. Dari usia 25-31 kemiringannya relatif curam, ini mengindikasikan laju pernikahan yang

tinggi selama durasi usia ini. Peningkatan drastic terjadi pada usia 36 tahun. Diakhir kurva, laju tidak terlalu tepat karena kecilnya ukuran sampel dalam periode ini.

Ringkasan survival dapat diperoleh dari tingkat yang berbeda dari variabel faktor dengan menambahkan bentuk tersebut dalam fungsi `survfit`. Perkalian kurva survival dapat pula ditampilkan dalam grafik yang sama.

```
> fit <- survfit(surv.marr ~ sex)
> fit
Call: survfit(formula = surv.marr ~ sex)

            n events median 0.95LCL 0.95UCL
sex=male    9      6     30      26     Inf
sex=female 18     10     28      28     Inf

> summary(fit)

Call: survfit(formula = surv.marr ~ sex)

                sex=male
time n.risk n.event survival std.err lower 95%CI upper 95%CI
  25      9       1    0.889   0.105       0.706       1.000
  26      7       2    0.635   0.169       0.377       1.000
  29      5       1    0.508   0.177       0.257       1.000
  30      4       1    0.381   0.172       0.157       0.924
  31      3       1    0.254   0.155       0.077       0.838

                sex=female
time n.risk n.event survival std.err lower 95%CI upper 95%CI
  25     18       1    0.944  0.0540      0.8443       1.000
  26     17       2    0.833  0.0878      0.6778       1.000
  27     13       2    0.705  0.1117      0.5169       0.962
  28     10       3    0.494  0.1287      0.2961       0.823
  29      6       1    0.411  0.1309      0.2204       0.768
  36      3       1    0.274  0.1419      0.0994       0.756

> plot(fit, col=c("red", "blue"), lty=c(1,2), las=1)
> title(main="Age at Marriage", xlab="Time (years)")
> mtext(side=3, text="Proportion", at=-2)

> legend(10,.4, legend=c("male", "female"), lty=c(1,2),
    col=c("red", "blue"))
```




Saat terjadi perkalian kurva survival, garis selang kepercayaan 95% diabaikan. Kurva yang muncul serupa mengindikasikan bahwa laki-laki dan perempuan dalam acara wokshop menikah pada laju yang sama. Lebih jelas perbandingan antar grup dipaparkan secara detail pada subbab selanjutnya.

## *Perbandingan Statistik antara kurva survival*

Kurva survival dapat diuji untuk perbedaan statistik dengan perintah `survdiff`

```
> survdiff(surv.marr ~ sex)
Call:
survdiff(formula = surv.marr ~ sex)

             N Observed Expected (O-E)^2/E (O-E)^2/V
sex=male     9        6     5.37    0.0746     0.125
sex=female 18       10    10.63    0.0376     0.125

 Chisq= 0.1  on 1 degrees of freedom, p= 0.724
```

Dengan ukuran sampel kecil, perbedaan dapat secara sederhana

dijelaskan. Perintah `survdiff` sebenarnya memiliki 5 argumen, yang terakhir bernama 'rho', yang mengelompokkan jenis uji yang akan digunakan. Saat rho = 0 (by default) maka uji log-rank atau Mantel-Haenszel chi-squared akan digunakan. Uji ini membandingkan jumlah peristiwa yang diharapkan dalam setiap grup dengan nilai observasi. Jika level perbedaan antara kedua grup terlalu tinggi, nilai chi-squared akan membesar dan P value mengecil, hal ini menunjukkan bahwa kurva berbeda secara signifikan. Jika rho = 1 maka modifikasi Peto dari uji Gehan-Wilcoxon (kadang disebut uji Peto) akan digunakan yang akan memberikan banyak detail tentang kejadian sebelumnya.

### *Perbandingan Bertingkat*

Berikut merupakan asosiasi signifikan antara jenis kelamin dan pendidikan.

```
> cc(sex, educ)
        educ
sex      bach- >bachelor Total
  male       1         8     9
  female    12         6    18
  Total     13        14    27
OR =  0.07
95% CI = 0.001 0.715
Chi-squared = 7.418 ,  1 d.f. , P value = 0.006
Fisher's exact test (2-sided) P value = 0.013
```

Para wanita kelihatan memiliki tingkat pendidikan yang lebih tinggi. Pengaruh jenis kelamin dalam survival dengan penyesuaian untuk pendidikan dapat diperoleh dengan:

```
> survdiff(surv.marr ~ sex + strata(educ))
Call:
survdiff(formula=surv.marr ~ sex + strata(educ))

            N Observed Expected (O-E)^2/E (O-E)^2/V
sex=male    9        6     5.61    0.0266    0.0784
sex=female 18       10    10.39    0.0144    0.0784

 Chisq= 0.1  on 1 degrees of freedom, p= 0.779
```

Pengaruh penyesuaian tidak terlalu berbeda dengan yang tidak dilakukan penyesuaian. Kurangnya pembauran pada kasus ini karena sedikitnya pengaruh independen dari pendidikan pada usia pernikahan.

Kita akan membahas tentang hal ini pada bab selanjutnya.

```
> save.image(file = "Marryage.Rdata")
```

## Referensi

Kleinbaum D, Klein M (2005). Survival Analysis: A Self-Learning Text.

Hosmer Jr D, Lemeshow S (1999). Applied Survival Analysis: Regression Modeling of Time to Event Data.

## Latihan

Dataset **Compaq** berisi data dari studi lanjutan (*follow up study)* tentang kanker payudara di Eropa yang mengevaluasi apakah pasien di rumah sakit swasta ('rumah sakit') memiliki kelangsungan hidup lebih baik ('tahun').

Soal 1.

Periksalah distribusitentang tahun kematian dan *censoring*.

Soal 2.

Gambarkan kurva Kaplan-Meier untuk setiap grup rumah sakit dengan tanda censoring yang ditunjukkan dalam kurva. Tampilkan banyaknya resiko (*risk)* pada tingkat interval waktu yang masuk akal.

Soal 3.

Uji tingkat signifikansinya dengan dan tanpa penyesuaian (*adjustment*) untuk potensial pembauran yang mungkin; : age ('agegr'), stage of disease ('stage'dan socio-economic level ('ses').

# BAB 22

# Regresi Cox

## *Model proporsional hazard Cox*

Sama halnya dengan tipe variabel respon lainnya, variabel ketahanan dapat diuji menggunakan lebih dari satu respon menggunakan pemodelan regresi. Terdapat banyak pilihan 'parametric regression' untuk objek ketahanan. Masing-masing memiliki asumsi spesifik mengenai distribusi peluang ketahanan selama pengamatan (dinamakan fungsi hazard). Dalam studi epidemiologi, regresi yang paling sering digunakan untuk analisis ketahanan adalah regresi Cox, yang tidak memiliki asumsi mengenai fungsi hazard.

Sementara model regresi parametrik mengikuti prediksi peluang ketahanan pada setiap titik waktu, regresi Cox fokus pada pengujian perbedaan peluang ketahanan setiap kelompok dengan penyesuaian faktor. Asumsi yang terpenting ialah memenuhi 'proportional hazards'.

Secara matematika, laju hazard $h=h(t)$ merupakan sebuah fungsi yang bergantung pada $n$ kovariat bebas $\boldsymbol{X}$, dimana $\boldsymbol{X}$ menotasikan vektor $X_1, X_2, X_3 \ldots, X_n$ dimana $X_i$, $i = 1, 2, 3, \ldots n$, dan $t$ waktu. Fungsi hazard dapat juga ditulis sebagai $h(t, \boldsymbol{X})$. Ini menyatakan bahwa penjumlahan pengaruh dari satu kelompok terhadap kelompok lain merupakan proporsi konstan.

Berdasarkan asumsi proporsional hazard:

$$h(t, X) = h_0(t)e^{\sum \beta_i X_i}$$

Ruas kiri persamaan diatas menyatakan bahwa hazard dipengaruhi oleh waktu dan kovariat. Ruas kanannya terdiri $h_0(t)$, yang merupakan fungsi hazard mula-mula saat semua $X_i$ bernilai nol. Fungsi hazard mula-mula dikalikan *e* yang dipangkatkan dengan penjumlahan seluruh kovariat terboboti dengan koefisien estimasi $\beta_i$.

Akibatnya,

$$\frac{h(t, X)}{h_0(t)} = e^{\sum \beta_i X_i}$$

Ruas kiri merupakan proporsi atau rasio antara hazard kelompok dengan variabel bebas ***X*** dibagi hazard mula-mula. Ruas kanan merupakan eksponensial jumlah keluaran koefisien estimasi dan vector kovariat $X_i$, yang bebas dari waktu, misalnya diasumsikan konstan sepanjang waktu. Maka $e^{\beta iXi}$ adalah kenaikan hazard, atau rasio hazard karena pengaruh bebas dari variabel ke $i^{th}$.

Kapanpun terjadi sebuah kejadian, peluang bersyarat, atau proporsi subjek antara kelompok yang berbeda dalam memperoleh hazard, diasumsikan konstan.

Kita akan menggunakan data dari bab sebelumnya untuk menguji pengaruh bebas dari jenis kelamin dalam usia pernikahan.

```
> zap()
> library(survival)
Loading required package: splines
> load("Marryage.Rdata")
> use(.data)
> cox1 <- coxph(surv.marr ~ sex)
> cox1
================================
            coef exp(coef) se(coef)      z    p
sexfemale -0.170     0.844    0.522 -0.325 0.74
```

Koefisien negatif dan tidak signifikan. Rasio hazard (exp(coef)) sebesar 0.844 menyarankan pengurangan keseluruhan 16% laju hazard jenis kelamin perempuan dibandingkan laki-laki. Untuk mendapatkan selang kepercayaan

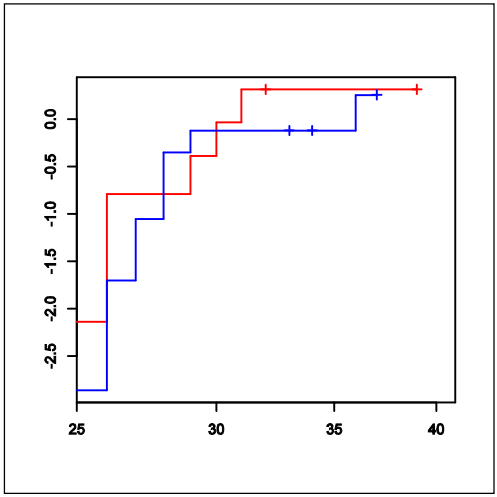
95%, dibutuhkan ringkasan objek 'coxph'.

```
> summary(cox1)
================================
          exp(coef) exp(-coef) lower .95 upper .95
sexfemale     0.844       1.19     0.304      2.35
================================
```

## Uji asumsi proporsional hazard

Berdasarkan grafik, kurva dua jenis kelamin dapat dibandingkan setelah sumbu vertikal telah ditransformasi menggunakan -log(log(y)) dan diplotkan terhadap log(waktu). Jika dua kurva sejajar, asumsi proporsional hazard tidak dapat dilanggar (asumsi terpenuhi).

```
> fit <- survfit(surv.marr ~ sex)
> plot(fit, conf.int=FALSE, fun="cloglog", xlim=c(25,41),
    col=c("red", "blue"))
```



Dua kurva diatas saling bersilangan lebih dari satu kali. Sulit untuk menilai apakah asumsi tidak terpenuhi melalui grafik tersebut. Uji formal asumsi proporsional hazard dapat dilakukan melalui:

```
> cox.zph(model1) -> diag1; diag1
             rho    chisq     p
sexfemale 0.00756 0.000883 0.976
```

Bukti untuk memenuhi asumsi proporsional hazard sangat lemah. Hasil diagnostik ini dapat lebih jauh dieksplorasi.

### *Tren waktu rasio hazard*

Atribut ini dapat diringkas dalam sebuah grafik dengan memplotkan perubahan beta (koefisien estimasi) sepanjang waktu.

```
> diag1$x  # x coordinates for plotting time
> diag1$y  # y coordinates for plotting beta coefficients
> plot(cox.zph(model1))
```

Grafik ini harus dibaca bersamaan dengan hasil sebelumnya dalam bab dimana kejadian dan informasi jenis kelamin subjek diurutkan berdasarkan waktu.

```
> data.frame(age, sex, age.marr, married,
    surv.marr)[order(time),]
   age    sex age.marr married surv.marr
4   25   male       NA   FALSE       25+
15  32 female       25    TRUE       25
22  29   male       25    TRUE       25
1   44   male       26    TRUE       26
2   43 female       26    TRUE       26
=======================================
```

Pertama dua kejadian terjadi pada tahun ke-25 dimana seorang laki-laki dan wanita menikah. Hazard dalam 'diag1$y' sebesar 1.43 dan -2.92. Pada tahun ke-26, terdapat empat kejadian dari dua laki-laki (beta = -3.16) dan dua wanita (beta = 1.19). Nilai ganda dari hasil beta menunjukkan sebuah peringatan tetapi tidak terlalu serius. Poin berikutnya diplotkan dalam bentuk yang sama. Sebuah garis dibuat melewati nilai beta untuk menggambarkan tingkat stabilitas koefisien sepanjang waktu. Peluang menikah untuk wanita lebih rendah

daripada laki-laki saat mereka berusia dibawah 26 tahun atau berusia diatas 29 tahun. Jadi wanita memiliki peluang yang besar untuk menikah. Namun, uji tersebut menunjukkan bahwa hasil yang diperoleh dapat dijelaskan dengan sederhana secara kebetulan.

Untuk perkalian kovariat digunakan prinsip yang sama.

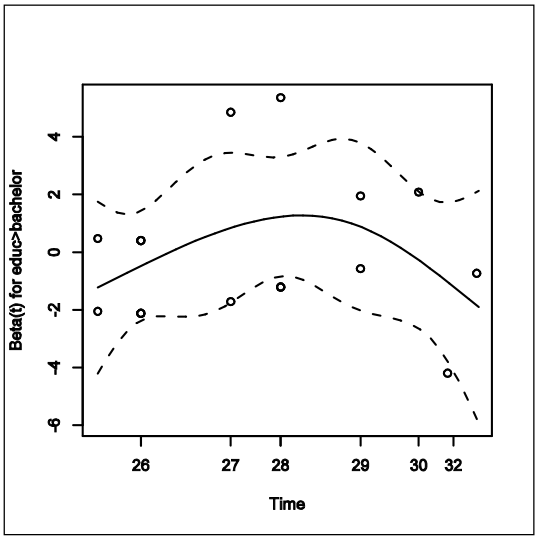
```
> cox2 <- coxph(surv.marr ~ sex + educ)
> cox2
> summary(cox2)
=================================================
             exp(coef) exp(-coef) lower.95 upper.95
sexfemale        0.831       1.20    0.230     2.99
educ>bachelor    0.975       1.03    0.278     3.42
=================================================
> cox.zph(cox2) -> diag2; diag2
                 rho   chisq     p
sexfemale     0.0246 0.00885 0.925
educ>bachelor 0.0321 0.01547 0.901
GLOBAL            NA 0.01604 0.992
```

Hasil pengujian dipisahkan oleh setiap variabel. Akhirnya, uji keseluruhan menunjukkan bahwa hasil tidak signifikan.

```
> diag2$x # x coordinates for plotting time: same as diag1
> diag2$y # two columns, one for each variable
> plot(cox.zph(cox2), var=1) # for the first variable of y
```

Koefisien jenis kelamin dengan penyesuaian untuk variabel pendidikan tidak terlalu banyak berubah.

```
> plot(cox.zph(cox2), var=2)
```




Laju hazard untuk pernikahan orang-orang yang memiliki pendidikan tinggi meningkat pada umur 27-29 tahun. Pada usia akhir duapuluhan, mereka memiliki kesempatan sedikit lebih tinggi untuk menikah dibanding dengan orang-orang yang memiliki tingkat pendidikan rendah. Kebalikan untuk sisa waktu (umur) lainnya. Sekali lagi, perbedaan ini tidak signifikan dan dapat dijelaskan secara kebetulan.

## Regresi Cox bertingkat

Contoh diatas memiliki sedikit subjek, dan tidak terlalu mengejutkan bahwa hasilnya tidak signifikan. Sekarang kita menggunakn kembali dataset kanker **Compaq**, yang digunakan sebagai latihan pada bab sebelumnya. Tujuan utama sekarang adalah untuk menguji apakah pasien kanker payudara dalam rumah sakit umum atau swasta memiliki perbedaan tingkat (laju) ketahanan hidup setelah peyesuaian tingkat, status sosial ekonomi dan umur.

```
> zap()
> data(Compaq)
> use(Compaq)
> des(); summ(); codebook()
> surv.ca <- Surv(year, status)
> cox3 <- coxph(surv.ca ~ hospital + stage + ses + agegr)
> summary(cox3)
Call:
coxph(formula=surv.ca ~ hospital+stage+ses+agegr)
  n= 1064
                     coef exp(coef) se(coef)      z       p
hospitalPrivate  -0.4224     0.655    0.142 -2.971 3.0e-03
stageStage 2      0.7682     2.156    0.123  6.221 5.0e-10
stageStage 3      2.4215    11.263    0.156 15.493 0.0e+00
stageStage 4      1.3723     3.944    0.190  7.213 5.5e-13
sesHigh-middle   -0.0944     0.910    0.133 -0.712 4.8e-01
sesPoor-middle    0.0341     1.035    0.178  0.192 8.5e-01
sesPoor          -0.4497     0.638    0.144 -3.126 1.8e-03
agegr40-49        0.2574     1.294    0.164  1.569 1.2e-01
agegr50-59        0.4923     1.636    0.164  2.999 2.7e-03
agegr60+          1.4813     4.399    0.159  9.343 0.0e+00
```

Pasien pada rumah sakit swasta memiliki dua-pertiga resiko (hazard) dibandingkan dengan pasien pada rumah sakit umum setelah peyesuaian status, status social ekonomi dan umur. Untuk memeriksa apakah ketiga variabel kategori layak untuk dimasukkan kedalam model, perintah `step` untuk regresi stepwise dapat digunakan.

```
> step(cox3)
Start:  AIC= 4854.56
 surv.ca ~ hospital + stage + ses + agegr

           Df    AIC
<none>        4854.6
- ses       3 4860.2
- hospital  1 4862.0
- agegr     3 4951.6
- stage     3 5059.9
===== Further output omitted due to redundancy ====
```

Nilai AIC sangat rendah saat tidak ada variabel yang dihilangkan. Oleh karena itu, semua variabel harus dijaga. Selanjutnya dinilai asumsi proporsional hazard.

```
> cox.zph(cox3)
```

```
                              rho   chisq       p
hospitalPrivate hospital  0.03946  0.6568 0.41768
stageStage 2              0.05406  1.1629 0.28086
stageStage 3             -0.09707  3.6786 0.05512
stageStage 4             -0.10222  4.2948 0.03823
sesHigh-middle            0.00968  0.0367 0.84818
sesPoor-middle           -0.04391  0.7612 0.38297
sesPoor                   0.10409  4.4568 0.03476
agegr40-49               -0.07835  2.3831 0.12266
agegr50-59               -0.09297  3.2339 0.07213
agegr60+                 -0.09599  3.5242 0.06048
GLOBAL                         NA 23.3117 0.00965
```

Tingkat tertinggi dan kelompok sosial-ekonomi terendah berkontribusi banyak terhadap nilai statistik chi-square. Pengujian keseluruhan menghasilkan nilai P value signifikan dan hal ini menunjukkan bahwa asumsi tidak terpenuhi. Solusi yang mungkin adalah melakukan analisis bertingkat pada salah satu variabel kategori, misalkan 'stage'.

```
> cox4 <- coxph(surv.ca ~ hospital+strata(stage)+ses+agegr)
> cox.zph(cox4)
                              rho  chisq      p
hospitalPrivate hospital  0.04407  0.797 0.3720
sesHigh-middle            0.00801  0.025 0.8743
sesPoor-middle           -0.04857  0.920 0.3376
sesPoor                   0.09747  3.785 0.0517
agegr40-49               -0.07366  2.097 0.1476
agegr50-59               -0.08324  2.565 0.1093
agegr60+                 -0.08521  2.761 0.0966
GLOBAL                         NA 10.297 0.1724
```

Menggunakan 'stage' sebagai faktor bertingkat menurunkan semua nilai all chi-square dan asumsi proporsi hazard terpenuhi.

```
> summary(cox4)
Call:
coxph(formula = surv.ca ~ hospital + strata(stage) + ses +
  agegr)
  n= 1064
                     coef exp(coef) se(coef)      z      p
hospitalPrivate   -0.4049     0.667    0.141 -2.866 0.0042
sesHigh-middle    -0.1078     0.898    0.133 -0.811 0.4200
sesPoor-middle     0.0374     1.038    0.179  0.209 0.8300
sesPoor           -0.4201     0.657    0.144 -2.926 0.0034
```

```
agegr40-49          0.2532     1.288    0.164  1.542 0.1200
agegr50-59          0.4703     1.600    0.165  2.857 0.0043
agegr60+            1.4514     4.269    0.159  9.141 0.0000
```

Koefisien '`cox4`' hamper sama dengan koefisien '`cox3`'. Perhatikan faktor 'stage' diabaikan. Regresi Cox bertingkat mengabaikan faktor bertingkat. Karena tujuan kita adalah untuk mendokumentasikan perbedaan antara tipe rumah sakit, koefisien untuk variabel lainnya tidak terlalu dibutuhkan jika kovariat dengan baik disesuiakan.

## Referensi

Kleinbaum D, Klein M (2005). Survival Analysis: A Self-Learning Text.

Hosmer Jr D, Lemeshow S (1999). Applied Survival Analysis: Regression Modeling of Time to Event Data.

## Latihan

Masalah 1.

Dapatkah 2 variabel lainnya (status sosial-ekonomi dan umur) digunakan sebagai faktor bertingkat?

Masalah 2.

Gunakan perintah `plot(cox.zph)` untuk '`cox3`' dan '`cox4`' untuk menguji perubahan rasio hazard dari rumah sakit swasta sepanjang waktu. Diskusikan pola residualnya.

BAB 23

# Menganalisis Data Tentang Sikap

***Kumpulan data tentang 'Sikap'***

Meskipun studi tentang sikap adalah di bidang ilmu-ilmu sosial, ahli epidemiologi harus memiliki beberapa ide tentang metode dasar analisis ini jenis data. Sebuah kuesioner pada sikap biasanya berisi pertanyaan dimana responden menentukan tingkat kesepakatan untuk pernyataan. Tingkat ini sering disebut sebagai skala Likert. Biasanya skala lima-titik digunakan, namun tujuh dan bahkan sembilan poin skala juga dapat digunakan. Meskipun sebagian besar digunakan di bidang psikometri, jenis skala penilaian kadang-kadang digunakan dalam studi epidemiologi seperti yang melibatkan kualitas hidup.

Epicalc menawarkan fungsi tableStack untuk menampilkan distribusi dari nilai dari beberapa variabel yang memiliki skala penilaian yang sama. Hal ini juga mendeteksi bagian-bagian yang perlu dibalik sebelum skor item/bagian dijumlahkan atau dirata-ratakan.

Dataset Sikap berasal dari sebuah survei tentang sikap terkait dengan layanan antara staf rumah sakit. Rinciannya dapat dicari dari perintah berikut.

```
> help(Attitudes)
> data(Attitudes)
> use(Attitudes)
> des()
> summ()
```

Untuk mendapatkan jumlah secara bersamaan dari setiap item kuesioner ketik:

```
> tableStack(qa1:qa18)
     1  2  3  4  5  count mean sd  description
qa1  0  0  7  54 75 136   4.5  0.6 I have pride in my job
qa2  0  2  13 60 61 136   4.3  0.7 I'm happy to give service
qa3  30 52 25 20 9  136   2.5  1.2 I feel difficulty in
  giving service
qa4  0  0  10 90 36 136   4.2  0.6 I can improve my service
qa5  0  3  5  39 89 136   4.6  0.7 A service person must
  have patience
qa6  17 19 58 29 12 135   3    1.1 I would change my job if
  given ...
qa7  0  3  7  68 58 136   4.3  0.7 Devoting some personal
  time will...
qa8  0  5  20 59 52 136   4.2  0.8 Hard work will improve
  oneself
qa9  0  0  4  41 91 136   4.6  0.5 Smiling leads to trust
qa10 1  1  16 74 44 136   4.2  0.7 I feel bad if I cannot
  give service
qa11 6  20 35 52 23 136   3.5  1.1 A client is not always
  right
qa12 2  26 45 49 13 135   3.3  0.9 Experienced clients
  should not ...
qa13 13 54 37 22 10 136   2.7  1.1 A client violating the
  regulation..
qa14 0  13 45 62 16 136   3.6  0.8 Understanding colleagues
  will ...
qa15 0  2  18 82 33 135   4.1  0.7 Clients like this place
  due to ...
qa16 36 53 21 16 8  134   2.3  1.2 Clients who expect our
  smiling ...
qa17 4  41 32 44 14 135   3.2  1.1 Clients are often self-
  centred
qa18 2  1  13 87 33 136   4.1  0.7 Clients should be better
  served
  Total score         130   67.1 4.9
  Average score       130   3.7  0.3
```

Semua item skala membagi respon yang sama mulai dari 1 sampai 5 meskipun kita dapat melihat dari output yang beberapa item memiliki jumlah nol untuk skala 1 dan 2. Fungsi *tableStack* menentukan nilai minimum dan maksimum dari semua variabel yang dipilih. Ini dengan mudah dapat diubah dengan memodifikasi argumen 'minlevel' dan 'maxlevel' ke fungsi, yang diatur untuk "otomatis" secara default. Empat statistik dihitung untuk setiap variabel: jumlah, sarana dan standar deviasi disajikan secara default, sementara median juga tersedia tetapi harus disetel TRUE jika diinginkan. Argumen lainnya termasuk 'var.labels', yang mengontrol tampilan deskripsi variabel, dan 'total', yang mengontrol tampilan dari nilai total dan rata-rata di bagian bawah.

Nilai total dan rata-rata tidak tepat jika salah satu item harus dibalik. Satu dapat menebak item untuk membalikkan berdasarkan kata-kata dari pertanyaan dan pada tingkat lebih rendah oleh distribusi terbalik dibandingkan dengan item lainnya. Item dapat ditentukan dengan 'vars.to.reverse' argumen. Sebagai contoh, jika item 3, 6 dan 16 dianggap skala dalam arah terbalik untuk barang-barang lainnya, maka kedua item harus ditentukan pada argumen 'vars.to.reverse' sebagai berikut:

```
> tableStack(qa1:qa18, vars.to.reverse=c(qa3,qa6,qa16))
               Reversed 1  2  3  4  5  count mean sd
qa1                   .  0  0  7  54 75 136   4.5  0.6
qa2                   .  0  2  13 60 61 136   4.3  0.7
qa3                   x  9  20 25 52 30 136   3.5  1.2
qa4                   .  0  0  10 90 36 136   4.2  0.6
qa5                   .  0  3  5  39 89 136   4.6  0.7
qa6                   x  12 29 58 19 17 135   3    1.1
qa7                   .  0  3  7  68 58 136   4.3  0.7
qa8                   .  0  5  20 59 52 136   4.2  0.8
qa9                   .  0  0  4  41 91 136   4.6  0.5
qa10                  .  1  1  16 74 44 136   4.2  0.7
qa11                  .  6  20 35 52 23 136   3.5  1.1
qa12                  .  2  26 45 49 13 135   3.3  0.9
qa13                  .  13 54 37 22 10 136   2.7  1.1
qa14                  .  0  13 45 62 16 136   3.6  0.8
qa15                  .  0  2  18 82 33 135   4.1  0.7
qa16                  x  8  16 21 53 36 134   3.7  1.2
qa17                  .  4  41 32 44 14 135   3.2  1.1
qa18                  .  2  1  13 87 33 136   4.1  0.7
 Total score                                  130   69.6 5.9
 Average score                                130   3.9  0.3
```

Item terbalik ditampilkan dengan tanda silang (x) pada kolom berjudul "Terbalik", menunjukkan bahwa skala telah terbalik untuk item itu. Statistik untuk nilai total dan rata-rata kemungkinan akan berubah karena arah skala terbalik dari item. Cara alternatif untuk memilih item untuk mengatur 'membalikkan' argumen ke TRUE.

```
> tableStack(qa1:qa18, reverse=TRUE)
```

Fungsi akan menghitung korelasi antara skor masing-masing item terhadap skor rata-rata tertimbang dari semua yang tersisa. Item yang berkorelasi negatif dengan rata-rata ini akan secara otomatis terbalik. Dalam Sikap dataset, ini adalah item 3, 6, 12, 13, 16 dan 17.

## Table Stack untuk variabel logis dan faktor

Semua pertanyaan dalam sikap dataset adalah bilangan bulat, sehingga memungkinkan untuk mendapatkan statistik untuk setiap item serta mereka untuk skor total dan mean. Jika kelas variabel tidak numerik, hanya menghitung frekuensi ditampilkan. Mari kita menjelajahi dataset Oswego, yang berisi data tentang 75 orang di bawah investigasi untuk mengetahui penyebab keracunan makanan akut setelah pesta makan malam.

```
> data(Oswego)
> use(Oswego)
> des()

No. of observations = 75
   Variable      Class           Description
1  age           numeric
2  sex           AsIs
3  timesupper    numeric
4  ill           logical
5  onsetdate     AsIs
6  onsettime     numeric
7  bakedham      logical
8  spinach       logical
9  mashedpota    logical
```

```
10 cabbagesal    logical
11 jello         logical
12 rolls         logical
13 brownbread    logical
14 milk          logical
15 coffee        logical
16 water         logical
17 cakes         logical
18 vanilla       logical
19 chocolate     logical
20 fruitsalad    logical
```

Untuk mendapatkan persentase, atur 'by' dengan "none".

```
> tableStack(bakedham:fruitsalad)
           No Yes count
bakedham   29  46    75
spinach    32  43    75
mashedpota 37  37    74
cabbagesal 47  28    75
jello      52  23    75
rolls      38  37    75
brownbread 48  27    75
milk       71   4    75
coffee     44  31    75
water      51  24    75
cakes      35  40    75
vanilla    21  54    75
chocolate  27  47    74
fruitsalad 69   6    75
```

Atau, prevalensi pemakan (Yes) dapat ditampilkan dengan menetapkan 'prevalensi' argumen ke TRUE.

```
> tableStack(bakedham:mashedpota, by="none",
  prevalence=TRUE)
                        Total
bakedham = Yes
  prevalence     46/75(61.3%)

spinach = Yes
  prevalence     43/75(57.3%)

mashedpota = Yes
  prevalence       37/74(50%)
```

Kembali ke data Sikap dan mengubah semua variabel faktor-faktor. Hal ini sering terjadi ketika pilihan-pilihan diberi label selama entri data.

```
> data(Attitudes)
> use(Attitudes)
> scales <- list("strongly agree"=1, "agree"=2, "neutral"=3,
  "disagree"=4, "strongly disagree"=5)
> for(i in 4:21){
   .data[,i] <- factor(.data[,i])
   levels(.data[,i]) <- scales
  }
```

Urutan perintah di atas hanya mengkonversi kolom keempat ke kolom ke-21 dari data (item 'qa1': 'qa21') faktor-faktor dan memberikan nilai-nilai setiap item label sesuai dengan unsur-unsur dalam 'skala'. Ini adalah tingkat dari item.

```
> des()
```

Semua item sekarang harus menjadi faktor. Menggunakan fungsi *tableStack* dengan frame data baru akan menghasilkan statistik yang tidak ditampilkan.

```
> tableStack(qa1:qa18)
```

Perhatikan bahwa kolom sekarang, telah berlabel. Jika jumlahan statistik yang diinginkan maka orang akan perlu untuk *unclass* semua variabel dalam frame data sebelum menggunakan fungsi. Jika frame data berisi banyak variabel, ini akan menjadi tugas yang cukup melelahkan. Epicalc memiliki fungsi untuk *unclass* semua variabel di dalam frame data yang dihasilkan dalam variabel yang dikonversi ke bilangan bulat, yaitu *unclassDataframe.*

```
> unclassDataframe(qa1:qa18)
> des()
> tableStack(qa1:qa18, reverse=TRUE)
```

## Cronbach's alpha

Untuk data survei tentang sikap, , langkah selanjutnya dalam analisis ini adalah menghitung koefisien reliabilitas, yaitu *Cronbach's alpha*, yang merupakan ukuran konsistensi internal dari survei kuesioner. Sebuah analisis sikap data

survei tidak akan pernah diterima oleh jurnal-jurnal ilmu pengetahuan yang paling sosial, kecuali *Cronbach's alpha* telah dihitung.

Secara singkat, koefisien ini mencerminkan tingkat korelasi antara semua item dari skala yang sama. Kadang-kadang disebut koefisien reliabilitas karena mencerminkan konsistensi antara item. Jika nilai koefisien ini terlalu rendah (katakanlah kurang dari 0,7), skala dianggap memiliki konsistensi internal yang agak rendah, dan total atau rata-rata skor dihitung dari item-item yang tidak konsisten mungkin tidak benar mencerminkan domain yang pertanyaan-pertanyaannya mencoba untuk mengukur.

Fungsi *alpha* dari Epicalc menghitung *Cronbach's alpha*, dan memungkinkan pengguna untuk melihat efek dari pergatian setiap item pada kedua koefisien dan korelasi antara setiap item dan yang tersisa.

Argumen untuk fungsi mirip dengan fungsi *tableStack*. Argumen pertama adalah vektor dari nama variabel (tanpa tanda kutip) atau indeks kolom dari variabel dalam frame data.

```
> alpha(qa1:qa18, var.labels=FALSE)
Number of items in the scale = 18
Sample size = 136
Average inter-item correlation = 0.1461

Cronbach's alpha: cov/cor computed with
  'pairwise.complete.obs'
      unstandardized value = 0.708
        standardized value = 0.7549

Item(s) reversed and new alpha if the item omitted:
   Reversed Alpha    Std.Alpha r(item,rest) description
qa1    .    0.685817 0.732773  0.461288     I have pride in
  my job
qa2    .    0.674703 0.725548  0.556550     I'm happy to
  give
qa3    x    0.699929 0.749653  0.282889     I feel
  difficulty in
qa4    .    0.686278 0.730758  0.467432     I can improve my
qa5    .    0.691590 0.739174  0.329396     A service person
  must
qa6    x    0.682247 0.739252  0.392348     I would change
  my job
qa7    .    0.674438 0.722168  0.563173     Devoting some
```

```
  personal
qa8    .    0.677646 0.728148  0.484181     Hard work will
  improve
qa9    .    0.691061 0.736795  0.410533     Smiling leads to
  trust
qa10   .    0.708569 0.755929  0.153067     I feel bad if I
  cannot
qa11   .    0.729312 0.764704  0.007361     A client is not
  always
qa12   x    0.720390 0.765974  0.057229     Experienced
  clients
qa13   x    0.693353 0.748163  0.303587     A client
  violating the
qa14   .    0.710688 0.757130  0.128318     Understanding
  colleagues
qa15   .    0.685665 0.733966  0.415518     Clients like
  this place
qa16   x    0.692937 0.744674  0.311757     Clients who
  expect our
qa17   x    0.720186 0.764488  0.088212     Clients are
  often self…
qa18   .    0.695617 0.744489  0.296922     Clients should
  be...
```

Fungsi pertama menghitung matriks kovarians antara semua variabel yang dipilih. Matriks ini kemudian digunakan untuk menghitung rata-rata korelasi antar-item.

Kedua, koefisien alpha *unstandardized* dan *standardized* (baik yang belum dibakukan maupun yang sudah dibakukan) dihitung berdasarkan formula, yang dapat ditemukan dalam sebagian besar buku pelajaran. Nilai *unstandardized* cocok jika semuanya membagi nilai *coding* yang sama, seperti sikap dataset dimana semua item memiliki skala 1 sampai 5. *Alpha* yang sudah dibakukan adalah sesuai ketika variabel kode pada skala yang berbeda, yang kurang umum ditemukan dalam sebuah studi pada sikap.

Terakhir, sebuah tabel ditampilkan, dengan item yang telah otomatis terbalik ditandai dengan 'x', mirip dengan perintah *tableStack* tanpa argumen 'by' diberikan. Kolom 'alpha' dan 'Std. 'alpha adalah koefisien alpha *unstandardized* dan *standardized*, masing-masing, diperoleh ketika setiap variabel dihilangkan dari perhitungan.

Fungsi ini juga 'membalikkan' argumen, nilai default menjadi TRUE. Jika diatur

ke FALSE, maka skala semua item diasumsikan diukur dalam arah yang sama. Dalam dataset ini yang akan menghasilkan nilai alpha lebih rendah dan paling mungkin untuk kesimpulan yang salah.

Dari keluaran sebelumnya, koefisien *unstandardized* adalah 0,71 dan item yang bisa dihapus untuk meningkatkan (kenaikan) koefisien alpha item 10, 11, 12, 14 dan 17.

Analisis lebih lanjut bisa dikejar dengan kelalaian berturut-turut dari item. Sebuah pilihan yang berhasil item akan memiliki kuesioner dengan item tidak terlalu banyak namun dengan koefisien alpha diterima tinggi. Pertimbangkan menghapus angka 11, karena itu menghasilkan koefisien alpha tertinggi jika itu dihapus dan juga memiliki korelasi terendah dengan semua itemlainnya.

```
> alpha(c(qa1:qa10, qa12:qa18))
```

Kedua koefisien *alpha unstandardized* dan *standardized* telah meningkat. Seperti yang ditunjukkan oleh bagian ketiga dari hasil, koefisien alpha dapat lebih ditingkatkan dengan menghilangkan angka 12.

```
> alpha(c(qa1:qa10, qa13:qa18))
```

dan kemudian item 17.

```
> alpha(c(qa1:qa10, qa13:qa16, qa18))
```

dan kemudian item 14.

```
> alpha(c(qa1:qa10, qa13, qa15:qa16, qa18))
```

dan kemudian item 10.

```
> alpha(c(qa1:qa9, qa13, qa15:qa16, qa18))
```

Penghapusan lebih lanjut dari item yang tidak menghasilkan perbaikan pada koefisien alpha. Secara keseluruhan, 5 item yang dihapus dari 18 item asli untuk sampai pada model terbaik. Ini tugas yang agak membosankan dapat diotomatisasi dengan menggunakan fungsi lain Epicalc disebut *alphaBest*.

```
> alphaBest(qa1:qa18)
$best.alpha
[1] 0.7620925

$removed
qa11 qa12 qa17 qa14 qa10
  14   15   20   17   13
```

```
$remaining
 qa1  qa2  qa3  qa4  qa5 qa6 qa7 qa8 qa9 qa13 qa15 qa16 qa18
   4    5    6    7    8   9  10  11  12   16   18   19   21
```

*Cronbach's alpha* terbaik dicapai dengan indeks dari item yang dihapus dan sisanya terdaftar. Nilai-nilai dari kedua vektor adalah indeks dari variabel dalam frame data. Sebagai contoh, pertama kita dihapus 'qa11', yang merupakan variabel 14, kemudian 'qa12', yang merupakan, 15 'qa17', yang merupakan ke-20, dan seterusnya. Demikian pula, variabel yang tersisa adalah 'qa1', mana variabel 4, 'qa2', yang merupakan, 5 'qa3', yang merupakan 6, dll.

Secara default, fungsi memilih model terbaik berdasarkan koefisien *alpha unstandardized*. Jika pilihan terbaik adalah didasarkan pada standar alfa, kemudian *'standardized*' harus di set ke TRUE.

```
> alphaBest(qa1:qa18, standardized=TRUE)
```

Hasil yang persis sama dalam hal ini, karena semua item memiliki skala yang sama. Menyimpan item yang dihapus dan item yang tersisa sebagai indeks memiliki keuntungan yang sangat penting seperti yang ditunjukkan berikutnya. Vektor 'sisa' item dapat disimpan dan digunakan lebih lanjut dalam perintah *tableStack* dijelaskan sebelumnya.

```
> alphaBest(qa1:qa18)$remaining -> wanted
```

Fungsi *tableStack* menerima sebuah vektor argumen integer untuk 'vars'. Untuk mendapatkan set akhir terbaik dari item, dengan membalikkan yang diperlukan, langkah berikutnya adalah dengan menggunakan perintah *tableStack* pada item yang ingin menyimpan item dari hasil untuk objek **R**.

```
> tableStack(vars=wanted, reverse=TRUE, var.labels=FALSE) ->
  b
```

Perhatikan bahwa nilai rata-rata sekarang telah meningkat 3,7-4,0 menggunakan metode (mungkin naif) asli untuk menjaga semua item dan tanpa perlu menyelidiki untuk membalikkan item. Obyek disimpan 'b' berisi nilai rata-rata dan total, yang dapat disimpan kembali ke frame data default untuk pengujian hipotesis lebih lanjut.

```
> mean.score <- b$mean.score
> total.score <- b$total.score
> pack()
```

```
> des()
> t.test(mean.score ~ sex)    # p-value = 0.7348
```

Cara alternatif menampilkan hasil dari pengujian hipotesis untuk perbedaan antara dua jenis kelamin di item dan skor rata-rata akan menjadi:

```
> tableStack(vars=c(wanted, mean.score), by=sex, var.=FALSE)
```

Fungsi menentukan uji statistik yang sesuai digunakan untuk semua variabel. Jika distribusi tidak normal, maka tes Wilcoxon rank sum digunakan sebagai pengganti uji t-. Rincian perintah *tableStack* menggunakan argument 'by' dijelaskan dalam Bab 27 - "Table Stacking for a Manuscript"

## Ringkasan

Singkatnya, ketika Anda punya dataset pada sikap, itu adalah ide yang baik untuk mengeksplorasi variabel dengan *tableStack*, awalnya tanpa salah satu item terbalik. Hati-hati melihat distribusi komparatif dari item dan membaca setiap pertanyaan (atau deskripsi variabel) untuk mendapatkan ide dari arah, baik positif atau negatif, skala item. Item yang harus dibalik biasanya yang berlawanan dengan distribusi untuk sebagian yang tersisa. Jika variabel adalah faktor, gunakan *unclassDataframe* untuk mengkonversikannya ke bilangan bulat. Sebenarnya tidak ada perlu untuk menyimpan nilai total atau rata-rata pada tahap ini. Periksa *Cronbach's alpha* menggunakan fungsi alpha dan kemudian *alphaBest* untuk mendapatkan subset terbaik dari item yang memaksimalkan alfa. Simpan hasil ke sebuah obyek dan menempatkan 'tersisa' item sebagai argumen 'vars' untuk perintah *tableStack* akhir dengan 'reverse = TRUE'. Nilai total dan rata-rata model yang dipilih terbaik dengan item dengan benar terbalik dapat disimpan dan siap untuk analisa lebih lanjut.

## Referensi

Cronbach, L. J. 1951. Coefficient alpha and internal structure of tests. *Psychometrika*, 16: 297–334.

## Latihan

Download dan install library **psy** dari website CRAN. Coba buka dataset. **expsy**

```
> library(psy)
> data(expsy)
> des(expsy)
> head(expsy)
```

Tentukan mana dari semua item (it1 untuk it10) yang perlu dihilangkan. Cari item terbaik dengan menggunakan koefisien Cronbach's alpha.

BAB 24

# Perhitungan Ukuran Sampel

Perhitungan ukuran sampel sangat penting dalam studi epidemiologi. Dalam kebanyakan survey, ukuran populasinya sangat besar, sebagai akibatnya biaya untuk mengoleksi data dari semua subjek akan sangat tinggi. Dalam studi klinikal, mengambil terlalu banyak subjek dalam sebuah penelitian tidak hanya menimbulkan permasalahan dalam hal manajemen dan finansial tetapi juga dapat menjadi permasalahan etik. Jika sebuah kesimpulan dapat diperoleh dari ukuran sampel kecil, menggunakan lebih banyak subjek dapat menimbulkan resiko pada kelompok subjek yang perlakuannya rendah. Di sisi lain, survey dengan ukuran sampel yang terlalu kecil tidak akan mampu mendeteksi pengaruh yang siginifikan secara statistik.

## *Fungsi untuk menghitung ukuran sampel*

Penggunaan fungsi untuk menhitung ukuran sampel akan memudahkan pengguna pemula **R** untuk memahami prinsip dari argumen lebih cepat dan berarti.

*Epicalc* menyediakan empat fungsi untuk pengukuran besar sampel. Yang pertama untuk survei *prevalence*. Fungsi kedua untuk perbandingan dua proporsi yang digunakan dalam studi kasus-kontrol, studi *cross-sectional*, studi kohort atau *randomised controlled trial*. Fungsi ketiga digunakan untuk perbandingan dua rataan dan fungsi terakhir untuk penarikan sampel kualitas *assurance*.

Sebagai tambahan untuk perhitungan ukuran sampel, terdapat dua fungsi untuk menghitung *power* dalam penelitian komparatif, yaitu untuk membandingkan dua rataan dan untuk membandingkan dua proporsi.

## Survei Lapangan

Tujuan survei lapangan umumnya untuk mendokumentasikan kepastian (*prevalence*) populasi dalam keadaan tertentu, seperti infeksi helminthic atau ulasan tentang pelayanan kesehatan seperti program imunisasi. Ukuran sampel yang digunakan bergantung pada estimasi kepastian dan tingkat error kepastian yang dapat diterima oleh peneliti. Dalam kebanyakan situasi, sampling kluster sering digunakan. Keuntungan metode sampling ini adalah mengurangi waktu dan biaya untuk perjalanan mengoleksi data.

Sebagai contoh, sampling acak sederhana mungkin akan memerlukan 96 orang dari 96 desa yang akan didata. Hal ini tentu akan menyulitkan. Jumlah desa yang akan didata dapat dikurangi menjadi 30 dan ukuran sampel dapat digantikan dengan penarikan lebih banyak subjek dari setiap desa. Sedikit peningkatan pada besar sampel diimbangi dengan pengurangan biaya perjalanan. Bagaimanapun, tehnik sampling kluster mengatasi masalah lainnya. Subjek pada desa yang sama cenderung menjadi sama dengan subjek lainnya dari desa lainnya dalam hal resiko penyakit, pelayanan dll. Dengan kata lain, subjek dipilih dari kluster sama yang biasanya tidak 'independent'. Meskipun ukuran sampel diestimasi dari tehnik sampling acak sederhana harus dibuat untuk menghadapi masalah ini, 'alikeness among the same cluster' (atau 'design effect').

Fungsi `n.for.survey` dalam Epicalc digunakan dalam penghitungan ukuran sampel untuk sebuah survey. Untuk melihat argument dari fungsi ini, ketik:

```
> args(n.for.survey)
function(p, delta = 0.5 * min(c(p, 1 - p)), popsize = FALSE,
  deff = 1, alpha = 0.05)
```

Argumen fungsi ini adalah:

***p***: Estimasi peluang dalam bentuk proporsi 0 dan 1.

***delta****:* Perbedaan antara estimasi nilai *prevalence* dan batas interval kepercayaan. Sebagai contoh, jika 'p' diestimasi sebesar 30% tetapi kita tetap dapat menerima bahwa maksimum error dapat mencapai *prevalence* 50%, maka 'delta' adalah 0.5 - 0.3 = 0.2. jika delta tidak diberikan, maka nilai yang ditetapkan adalah *p* atau *1-p*, yang lebih kecil. Secara umum, *delta* memiliki banyak pengaruh dalam ukuran sampel jika dibandingkan dengan *p*. Saat *p* bernilai kecil maka *delta* lebih kecil dari *p*. Sebaliknya, batas terendah interval kepercayaan akan bernilai negatif atau batas teratas akan lebih besar dari 100%, dimana kedua tidak valid. Nilai yang ditetapkan (*default value*) dapat lebih diterima untuk nilai *prevalence* yang lebih kecil (misalkan dibawah 15%) atau *prevalence* yang cukup besar (misalkan diatas 80%). Jika nilai *prevalence* berada diantara nilai ini, maka setengah *p* (atau *1-p*) akan lebih tidak tepat. Pengguna dapat menggunakan *delta* yang lebih kecil.

***popsize***: Ukuran populasi berhingga. Ukuran populasi ini yang dapat digunakan dalam sebuah servey. Ukuran populasi kecil akan membutuhkan ukuran sampel yang relatih kecil. Jika nilainya FALSE, maka akan diabaikan dan diasumsikan bahwa populasinya sangat besar. Umumnya saat ukuran melebihi nilai yang ada, misalkan 5000, peningkatan lainnya dalam populasi akan memberi pengaruh yang kecil pada ukuran sampel.

***deff***: Pengaruh model, yang merupakan faktor penyesuaian untuk sampling kluster seperti yang dijelaskan diatas. Berdasarkan definisi, untuk sampling acak sederhana, *deff* benilai 1. Dalam penarikan sampel kluster dengan ukuran kluster yang besar serta level kesamaan diantara subjek dalam kluster yang sama juga besar, *deff* dapat membesar dan akan memenuhi ukuran sampel.

***alpha****:* merupakan peluang error tipe I. Dalam keadaan standar, alpha diatur sebesar 0.05 dan interval kepercayaan dari $p \pm delta$ adalah 95% batas kepercayaan dari *prevalence*. Dengan permintaan akurasi yang tinggi, sebagai contoh, pada kepercayaan 99%, ukuran sampel yang diperlukan juga akan meningkat.

Jika survey dilakukan dengan *prevalence* yang kecil (lebih kecil dari 15%), dalam populasi yang besar, semua nilai yang telah ditetapkan (*default value*) dapat diterima. Perintah yang digunakan adalah:

```
> n.for.survey(p=.05)

Sample size for survey.
Assumptions:
  Proportion       = 0.05
  Confidence limit = 95 %
  Delta            = 0.025 from the estimate.
  Sample size      = 292
```

Fungsi mengatur nilai 'alpha' sebesar 0.05 dan interval kepercayaan 95%. Argumen 'delta' diatur secara otomatis menjadi setengah dari 5% yaitu 0.025. Pengaruh model, 'deff', tidak diberikan jadi diatur sebesar 1. Ukuran sampel diasumsikan menjadi sangat besar dan tidak digunakan dalam penghitungan ukuran sampel.

Pada kesimpulannya, fungsi menyarankan bahwa jika sebuah batas kepercayaan 95% dari 5% $\pm$ 2.5% (dari 2.5% hingga 7.5%) dibutuhkan untuk estimasi proporsi dari 0.05 dalam populasi yang besar maka ukuran sampel yang dibutuhkan adalah 292.

Jika *prevalence* bernilai kecil, 'deff' untuk sampling kluster biasanya mendekati *unity*. Ukuran sampel yang dihitung tetap dapat digunakan bahkan jika sampling kluster digunakan karena kecilnya nilai kepastian (*prevalence*).

Jika nilai *prevalence* yang diestimasi mendekati 50%, delta 25% adalah sangat besar. Akan lebih baik jika direduksi menjadi $\pm$5% atau $\pm$10% dari *prevalence*. Jika sampling kluster digunakan pada kondisi ini, nilai 'deff' biasanya lebih dari 1.

Misalkan, dalam 30-cluster sampling standar pada pencakupan tugas imunisasi dimana kepastian estimasi mendekati 80%, nilai 'deff' akan berada disekitar nilai 2. Ukuran populasi dalam kasus ini biasanya besar dan limit kepercayaan 99% bahkan dibutuhkan 95%. Pada kasus ini, perhitungan yang disarankan adalah:

```
> n.for.survey(p =.8, delta =.1, deff=2, alpha=.01)

Sample size for survey.
```

```
Assumptions:
  Proportion       = 0.8
  Confidence limit = 99 %
  Delta            = 0.1 from the estimate.
  Design effect    = 2

  Sample size      = 212
```

Dengan total ukuran sampel 212 dan 30 kluster, rata rata ukuran setiap kluster adalah 212/30 = 7 subjek. Ukuran sampel ini dapat digunakan dalam survei standar untuk memperkirakan pencakupan imunisasi pada negara berkembang.

## Perbandingan dua proporsi

Perbandingan dua proporsi pada penelitian epidemiologi sering dilakukan.

Fungsi berikut dinamakan *n.for.2p* untuk tujuan ini. Seperti sebelumnya, argumen berikut ini diperlukan untuk dilakukan pengujian:

```
> args(n.for.2p)
function(p1, p2, alpha = 0.05, power = 0.8, ratio=1)
```

Dalam studi kasus-kontrol, proporsi (p1) dari subjek yang tersebar dengan faktor resiko diantara kasus (grup penderita) dibandingkan dengan proporsi (p2) dari subjek yang tersebar diantara kontrol (grup non-penderita).

Pada studi kohort, peluang (p1) terinfeksi penyakit diantara kelompok tersebar dibandingkan dengan peluang (p2) diantara kelompok non-tersebar.

Dalam percobaan *randomised controlled*, peluang (p1) sembuh diantara subjek yang diberikan pengobatan baru dibandingkan dengan peluang (p2) sembuh diantara subjek yang diberikan pengobatan lama.

Argumen alpha merupakan peluang error tipe I. Jika dua kelompok sebenarnya memiliki proporsi yang sama pada level populasi (hipotesis null benar), dengan ukuran sampel dari perhitungan ini, akan ada kesempatan 'alpha' bahwa hipotesis akan ditolak. Dalam kata lain, perbedaan dalam dua sampel akan diputuskan signifikan secara statistik. Seperti sebelumnya, adalah hal yang umum untuk mengatur nilai alpha sebesar 0.05.

*Power* dari sebuah studi merupakan peluang menolak hipotesis null saat

hipotesis tersebut salah. Pada situasi ini, peluang perkiraan perbedaan signifikan secara statistik dari proporsi sebuah populasi, yang pada kenyataannya sama besarnya dengan sampel. Hal ini cukup diterima dimana level *power*nya 80%. Error tipe kedua adalah 1-*power*, merupakan peluang tidak menolak hipotesis null saat hipotesis tersebut salah. Para ilmuwan biasanya mengizinkan nilai peluang yang lebih besar untuk error tipe II dibandingkan error tipe I. Menolak pengobatan baru yang sebenarnya lebih baik dari pengobatan yang lama mungkin disebabkan kurang seriusnya pengobatan yang baru sehingga menggantikan perlakuan yang lama dengan yang baru yang sebenarnya tidak lebih baik.

'ratio' merupakan rasio jumlah subjek dalam sampel 1 dengan jumlah subjek dalam sampel 2. Untuk ketiga tipe penelitian ini, ukuran sampel yang paling efisien (ukuran terkecil dari total sampel yang dapat menguji hipotesis) diperoleh saat rasio antara dua tingkatan kelompok adalah 1:1. Sebagai contoh, jika koleksi data untuk setiap subjek sesuai, perbandingan dua kelompok perlakuan (pengobatan) setiap 50 subjek sangat lebih baik dibandingkan dengan perbandingan 5 subjek dalam satu kelompok melawan 95 subjek dalam kelompok lainnya. Dalam keadaan tertentu, seperti pada saat penyakit langka diteliti, penelitian akan lebih cepat diselesaikan dengan lebih dari satu kontrol per kasus. Sebagai tambahan, dalam penelitian *cross-sectional*, status subjek pada sebaran dan outcome tidak diketahui dari awal; sampel tidak direncanakan. Pada kondisi ini dimana rasio bukan 1:1, nilai rasio harus ditentukan dalam kalkulasi.

Misalkan jika resiko ditetapkan sebesar 50% diantara kelompok penyakit dan 20% diantara grup kontrol, ukuran sampel minimal yang dibutuhkan untuk mendeteksi perbedaan ini bagi studi kontrol dapat dihitung oleh:

```
> n.for.2p(p1=0.5, p2=0.2)

Estimation of sample size for testing Ho: p1==p2
Assumptions:

     alpha = 0.05
     power = 0.8
        p1 = 0.5
        p2 = 0.2
     n2/n1 = 1
```

```
Estimated required sample size:

        n1 = 45
        n2 = 45
   n1 + n2 = 90
```

Penggunaan fungsi ini tidak rumit, hanya 'p1' dan 'p2' yang perlu dimasukkan. Argumen lainnya akan diatur menjadi nilai *default* secara otomatis. Kesimpulannya, hanya 45 kasus dan 45 kontrol digunakan untuk menguji hipotesis tidak adanya asosiasi. Jika penyakitnya langka, misalkan hanya 10 kasus setiap tahun dan peneliti ingin menyempurnakan studi lebih awal, maka dia dapat meningkatkan rasio kasus-kontrol menjadi 1:4

```
> n.for.2p(p1=0.5, p2=0.2, ratio=4)

Estimation of sample size for testing Ho: p1==p2
Assumptions:

     alpha = 0.05
     power = 0.8
        p1 = 0.5
        p2 = 0.2
     n2/n1 = 4

Estimated required sample size:

        n1 = 27
        n2 = 108
   n1 + n2 = 135
```

Ingat bahwa rasio sama dengan n2/n1. Studi ini dapat diselesaikan kurang dari 3 tahun dari 4.5 tahun sebenarnya. Meningkatkan rasio diatas memiliki pengaruh kecil terhadap reduksi jumlah kasus tetapi pengaruh besar jelas terlihat pada peningkatan jumlah kontrol. Misalkan, sebuah rasio 1 kasus per 9 kontrol akan mengurangi ukuran sampel menjadi 23 kasus (4 kasus direduksi) tetapi meningkatkan jumlah kontrol menjadi 207 (peningkatan mencapai 100).

Peningkatan *power* dari 0.8 menjadi 0.9 juga meningkatkan kebutuhan ukuran sampel. Atur rasio menjadi 1:1

```
> n.for.2p(p1=0.5, p2=0.2, power=0.9)
```

Output perintah diatas diabaikan, bagaimanapun 58 kasus dan 58 kontrol yang dibutuhkan (peningkatan sebesar 29% dari ukuran sampel diperlukan pada keduanya).

### *Hubungan antara p1, p2 dan odds rasio dalam studi kasus - kontrol*

Kita harus konsisten dengan pernyataan diatas, odd rasio merupakan rasio dua peluang sebaran yaitu p1/(1-p1) / {p2/(1-p2)}.

```
> .5/(1-.5)/(.2/(1-.2))
[1] 4
```

Mengatur 'p1' dan 'p2' untuk penghitungan ukuran sampel bagi studi kasus-kontrol digunakan untuk seterusnya. Bagaimanapun juga dalam beberapa contoh, terdapat permintaan untuk menghitung ukuran sampel berdasarkan proporsi sebaran dalam populasi umum (yang sama dengan nilai proporsi antar kontrol berdasarkan kelangkaan penyakit) dan odd rasio. Dengan kata lain, 'p2' dan odd rasio diketahui. Maka akan dihitung nilai 'p1'.

Sebagai contoh, jika proporsi sebaran antar populasi (p2) sam dengan 30%, dan odd rasio adalah 2, maka proporsi sebaran antar kasus (p1) serta ukuran sampel yang digunakan dapat dihitung menggunanakan:

```
> p2 <- 0.3
> or <- 2
> odds2 <- p2/(1-p2)
> odds1 <- or*odds2
> p1 <- odds1/(1+odds1);  p1
 [1] 0.4615385
> n.for.2p(p1,p2)

Estimation of sample size for testing Ho: p1==p2
Assumptions:

     alpha = 0.05
     power = 0.8
        p1 = 0.4615385
        p2 = 0.3
     n2/n1 = 1

Estimated required sample size:
        n1 = 153
```

```
        n2 = 153
   n1 + n2 = 306
```

Ukuran sampel yang dibutuhkan lebih besar dari contoh sebelumnya karena odd rasio dideteksi mendekati *unity*. Dengan kata lain, tingkat perbedaan yang dideteksi bernilai lebih kecil.

## *Studi kohort dan percobaan randomised controlled*

Diketahui bahwa 'p1' dan 'p2' adalah laju sukses antara dua perlakuan atau kelompok perlakuan. Perhitungan akan dilakukan selanjutnya.

Pada kenyataannya, apakah perhitungan dilakukan berdasarkan laju kesuksesan atau laju kegagalan, hasilnya tetaplah sama. Sebagai contoh, jika perlakuan A memberikan laju sukses 90% dan perlakuan B 80%, kita dapat mengatakan bahwa perlakuan A dan B memiliki laju gagal 10% dan 20%. Perhitungan ukuran sampel ada kedua kasus akan menghasilkan output yang sama.

```
> n.for.2p(p1=0.9, p2=0.8)

===== details omitted =========
        n1 = 219
        n2 = 219
   n1 + n2 = 438

> n.for.2p(p1=.1, p2=.2)

===== details omitted =========
        n1 = 219
        n2 = 219
   n1 + n2 = 438
```

## *Studi Cross-sectional: pengujian hipotesis*

Survei cross-sectional memiliki dua tujuan, pertama untuk mendokumentasikan kepastian suatu keadaan (sebuah penyakit atau keadaan sebaran, bahkan keduanya), kedua untuk menguji asosiasi antara sebaran dan outcome. Ukuran sampel untuk pengujian hipotesis ini berbeda dengan tujuan deskriptif ( yang telah dibahas diatas).

Perhitungan ukuran sampel untuk komponen kedua (pengujian hipotesis) studi cross-sectional harus berdasarkan fungsi `n.for.2p`. Serupa dengan studi

kohort dan percobaan *randomised controlled*, proporsi 'p1' dan 'p2' harus berorientasi pada outcome setiap kelompok sebaran dimana 'p1' sama dengan proporsi dari outcome positif antar kelompok tersebar dan 'p2' sama dengan proporsi dari outcome positif antar kelompok non-tersebar.

Pada sisi lain, nilai 'ratio' merupakan rasio antara kelompok tersebar dan non-tersebar yang harus diestimasi dari kepastian (*prevalence*) penyebaran.

Sebagai contoh, dalam sebuah survei, kepastian penyebaran diestimasi sebesar 20%, maka peluang terkena penyakit adalah 20% dan 5% antara populasi tersebar dan non-tersebar.

Dengan kepastian penyebaran sebesar 20% maka rasionya adalah n2:n1 yaitu 0.8/0.2 = 4.

```
> n.for.2p(p1=0.2, p2=0.05, ratio=4)
Estimation of sample size for testing Ho: p1==p2
Assumptions:

     alpha = 0.05
     power = 0.8
        p1 = 0.2
        p2 = 0.05
     n2/n1 = 4

Estimated required sample size:
        n1 = 48
        n2 = 192
   n1 + n2 = 240
```

Total ukuran sampel dalam survei cross-sectional untuk pengujian hipotesis adalah 240 subjek. Termasuk 48 orang tersebar dan 192 orang non-tersebar.

Ukuran sampel ini harus diperiksa untuk penyesuaian dengan objek lainnya, misalkan untuk menggambarkan kepastian penyebaran yang diestimasi sebesar 20%.

```
> n.for.survey(p=0.2)

Sample size for survey.
Assumptions:
  Proportion       = 0.2
  Confidence limit = 95 %
  Delta            = 0.1 from the estimate.
```

```
  Sample size      = 61
```

Ukuran sampel pada studi deskriptif lebih kecil daripada pengujian hipotesis. Dengan demikian ukuran sampel terakhir (240 subjek) yang sebaiknya digunakan.

## Perbandingan dua rataan

Dalam epidemiologi, perbandingan dua rataan tidak lazim dilakukan seperti halnya perbandingan dua proporsi. Hal ini terjadi karena keputusan *public health* yang didasarkan pada keyakinan outcome dikotomis dan kurangnya perbedaan level dari nilai rataan. Bagaimanapun, terdapat pula banyak outcome kesehatan yang diukur pada skala kontinu, perbedaan rataan yang dapat menjadi perhatian penting sosial. Contoh dari outcome kontinu adalah kecerdasan, penyebab penyakit dan kualitas hidup.

Dua sampel rataan biasanya memiliki dua standar deviasi yang berbeda. Dengan demikian fungsi untuk perhitungan ini membutuhkan sedikit tambahan argumen.

```
> args(n.for.2means)
function(mu1, mu2, sd1, sd2, ratio=1, alpha=0.05, power=0.8)
```

Secara intuisi, notasi digunakan untuk seterusnya. Terdapat empat argument yang wajib pengguna masukkan dalam fungsi, yaitu dua rataan dan standar deviasi yang sesuai.

***Catatan***: _____

*Pembaca harus hati-hati karena argumen fungsi yang mengandung tanda samadengan diikuti oleh nilai optional. Nilai sebelah kanan tanda merupakan nilai default yang digunakan fungsi saat argumen diabaikan. Argumen yang tidak mengandung tanda sama dengan adalah diwajibkan. Jika diabaikan maka akan terjadi error.*

Seperti pada contoh, misalkan sebuah agen therapeutic baru diharapkan dapat mengurangi rataan penyebab penyakit dari 0.8 menjadi 0.6 dalam sebuah kelompk subjek dan standar deviasi yang diharapkan adalah 0.2 dan 0.25. Untuk menghitung ukuran sampelnya, ketik perintah berikut:

```
> n.for.2means(mu1=0.8, mu2=0.6, sd1=0.2, sd2=0.25)

Estimation of sample size for testing Ho: mu1==mu2
Assumptions:
     alpha = 0.05
     power = 0.8
       mu1 = 0.8
       mu2 = 0.6
       sd1 = 0.2
       sd2 = 0.25

Estimated required sample size:
        n1 = 21
        n2 = 21
   n1 + n2 = 42
```

Eksperimen anastesi akan membutuhkan 21 subjek dalam setiap grup.

Pada kenyataannya, rumus matematika untuk perhitungan ukuran sampel tidak membutuhkan nilai pasti dari 'mu1' dan 'mu2'. Jika selisih antara rataan dan standar deviasi adalah tetap, mengubah dua rataan tidak akan berpengaruh pada ukuran sampel yang dihitung. Dengan demikian hasil yang sama diperoleh dari perintah berikut (output dihilangkan).

```
> n.for.2means(mu1=0.4, mu2=0.2, sd1=0.2, sd2=0.25)
```

## Pengambilan sampel lot penjaminan kualitas

Lot quality assurance sampling (LQAS) awalnya diterapkan pada proses manufaktur. Sebuah perusahaan mengambil sampel untuk memeriksa apakah *lot* produk siap dikirim. Jika persentase cacat diperkirakan lebih tinggi daripada level tertentu, *lot* akan ditolak. Jika tidak, semua *lot* akan dikirimkan ke pasar.

Perbedaan antara LQAS dan metode sampling lainnya adalah bahwa LQAS tidak memperkirakan persentase cacat yang tepat. LQAS hanya memeriksa apakah tingkat *acceptable* dilampaui. Ukuran sampel yang diperlukan untuk proses ini lebih kecil dibandingkan dengan proporsi kepastian (*prevalence*). Dengan demikian, biaya pemeriksaan menurun sangat jauh jika kualitas analisis komponen individu tinggi.

Sistem kesehatan mengadopsi LQAS terutama untuk pengawasan proporsi masalah. Sebagai contoh, dalam proses kualitas asuransi obat anti-TB di selatan Thailand, pengadaan essay dan pengujian terputusnya obat adalah cukup mahal. Metode LQAS digunakan untuk menghitung ukuran sampel minimal yang masih cukup untuk menguji apakah kualitasnya diterima.

Misalkan proporsi *acceptable* tertinggi dari specimen cacat ditetapkan sebesar 1 persen. Jika penelitian menyarankan bahwa proporsi actual berada pada level ini atau lebih rendah, maka *lot* diterima. Jika tidak semua, *lot* akan ditolak. Proporsi aktual (apakah itu lebih tinggi atau lebih rendah daripada level *acceptable* ini) tidak penting. Jika ukuran sampel terlalu kecil, misalkan 20, dan bahkan jika semua spesimen yang dipilih secara acak akan diterima, ia tetap tidak akan tentu bahwa kurang dari 1% dari seluruh spesimen yang cacat. Jika ukuran sampel terlalu besar, katakanlah 1000, maka anda telah menyia-nyiakan semua spesimen yang diuji. Ukuran sampel seperti ini sangat berlebihan.

Dengan ukuran sampel optimal, salah satu spesimen yang dipilih secara acak adalah cacat, proporsi *acceptable* keseluruhan *lot* akan diharapkan untuk dilampaui. Salah satu cara termudah untuk memahami hal ini adalah dengan melihat hasil perhitungan.

```
> n.for.lqas(p=0.01)

     Lot quality assurance sampling

                              Method = Normal approximation
                     Population size = 10000
  Maximum defective sample accepted = 0
     Probability of defect accepted = 0.01
                               Alpha = 0.05
                Sample size required = 262
```

Dari perhitungan ini, ambang batas untuk proporsi cacat (p) ditetapkan sebesar 1%. Ukuran sampelnya adalah 262. Ukuran *lot* diasumsikan sebesar 10,000 secara *default*. Sampel cacat maksimum yang diterima adalah 0 (secara *default*). Hal ini berarti bahwa jika terdapat dalah satu dari 262 spesimen yang cacat, proporsi 1% dianggap melebihi dan *lot* ditolak. Dengan ukuran sampel ini, peneliti akan mengambil sampel acak dari 262 spesimen dan memeriksa satu persatu. Jika semua spesimen lulus uji, sisa *lot* dari 10,000-262 = 9,738 spesimen dapat dipasarkan. Jika tidak, keseluruhan dari 1,000 akan ditolak.

Ada beberapa parameter pengontrol ukuran sampel. Alpha (error tipe I) umumnya ditetapkan sebesar 5%. Hal ini berarti bahwa jika hipotesis null (pesentase kecacatan kurang dari 5%) benar, terdapat peluang 5% bahwa akan ada setidaknya satu spesimen diantara keseluruhan sampel 262. Jika alpha ditetapkan menjadi kriteria yang lebih tinggi, misalkan 2.5%, maka ukuran sampel juga akan meningkat.

Proporsi ambang batas untuk sampel diterima berbanding terbalik dengan ukuran sampel. Jika ambang meningkat, misalkan sebesar 3%, ukuran sampel yang dibutuhkan akan dikurangi (hanya 87 yang dibutuhkan).

Sampel cacat maksimum yang diterima ditetapkan pada 0 secara default untuk meminimumkan ukuran sampel. Secara teori bisa berapapun. Bagaimanapun, semakin besar jumlah sampel cacat maka semakin besar pula ukuran sampelnya.

## Penentuan Power untuk perbandingan dua proporsi

Terkadang pembaca menemukan sebuah studi yang menyatakan tidak ada perbedaan signifikan antara dua kelompok. Orang mungkin meragukan apakah studi memiliki *power* yang cukup untuk mendeteksi perbedaan siginifikan jika sebuah perbedaan siginifikan secara klinikal terdapat pada tingkat populasi. Misalkan sebuah percobaan dengan 105 subjek dalam satu perlakuan pengobatan terdiri dari 35 kegagalana versus 50 subjek pada sebuah placebo dengan 20 kegagalan. Untuk membuat tabel data hipotesis ini, anda dapat mengetik perintah berikut:

```
> table1 <- c(35,70,20,30)
> dim(table1) <- c(2,2)
> table1 <- as.table(table1)
> cc(cctable=table1)
        A  B Total
A      35 20    55
B      70 30   100
Total 105 50   155
OR =  0.751
95% CI = 0.354 1.606
Chi-squared = 0.658 ,  1 d.f. , P value = 0.417
Fisher's exact test (2-sided) P value = 0.474
```

Odd rasio 0.75 memiliki selang kepercayaan yang besar. Hal ini mungkin menarik untuk diketahui berapa besar power dari ukuran sampel untuk studi khusus ini jika odd rasio sebesar 0.5 dan laju kegagalan antar kelompok placebo adalah sama.

```
> odds.placebo <- 20/30
> odds.treat <- .5 * odds.placebo
> p.placebo <- 20/50
> p.treat <- odds.treat/(1+odds.treat)
> power.for.2p(p1=p.treat, p2=p.placebo, n1=105, n2=50)
     alpha = 0.05
        p1 = 0.25
        p2 = 0.4
        n1 = 105
        n2 = 50
     power = 0.4082
```

Ukuran sampel yang digunakan pada studi ini hanya memiliki 40% peluang untuk menemukan perbedaan signifikan mengingat bahwa perlakuan pengobatan memiliki odd rasio 0.5. Oleh karena itu penelitian ini tidak meyakinkan.

Ingat bahwa *power* bergantung pada ukuran perbedaan yang akan dideteksi. Untuk mendapatkan signifikansi statistik untuk perbedaan yang besar akan membutuhkan ukuran sampel yang lebih kecil daripada untuk mendeteksi perbedaan yang kecil jika *power*nya tetap sama.

## Penentuan Power untuk perbandingan dua rataan

Misalkan sebuah studi menyatakan bahwa dalam percobaan *randomised controlled*, nutrisi-mikro diberikan kepada 100 siswa dan placebo kepada 100 siswa lainnya yang dipilih secara acak. Pada akhir tahun, rataan $\pm$ standard deviasi skor IQ dalam dua kelompok masing-masing $98 \pm 10.1$ dan $95 \pm 11.7$.

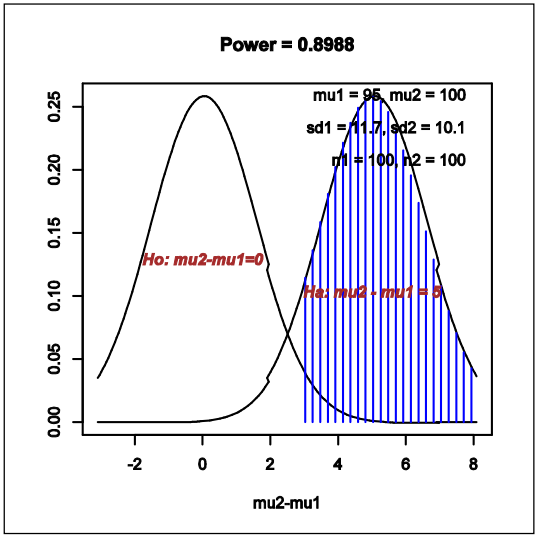
P*ower* berapakah yang digunakan untuk menentukan perbaikan 5 unit (IQ baru = 100) jika parameter dalam kelompok plasebo dan standar deviasi dari kelompok perlakuan ini tidak diubah?

Misalkan kelompok 1 menyatakan siswa dengan pengobatan placebo dan

kelompok 2 merupakan siswa-siswa yang menerima pengobatan baru.

```
> power.for.2means(mu1=95, mu2=100, sd1=11.7, sd2=10.1,
  n1=100, n2=100)
     alpha = 0.05
       mu1 = 95
       mu2 = 100
        n1 = 100
        n2 = 100
       sd1 = 11.7
       sd2 = 10.1
     power = 0.8988
```

Dengan ukuran sampel yang realtif besar ini, *power* yang digunakan untuk mendeteksi perbedaan 5 titik IQ berdasarkan asumsi ini adalah 90%.



## Latihan

Soal 1.

Hitung ukuran sampel maksimum yang dibutuhkan untuk mengestimasi kepastian prevalensi infeksi saluran pernafasan, dengan presisi 5%, dalam sebuah populasi yang terdiri anak-anak 1-5 tahun pada wilayah tertentu di sebuah Negara berkembang.

Soal 2.

Sebuah studi kasus-kontrol dilakukan untuk mengetahui efesiensi vaksin pencegahan tubekolosis anak menggunakan placebo. Asumsikan bahwa 50% kontrol tidak diberi vaksin. Jika jumlah kasus dan kontrolnya sama, berapa ukuran sampel yang dibutuhkan untuk mendeteksi, dengan power 80% dan 5% error tipe I serta odd rasio minimal 2 dalam populasi tersebut?

Soal 3.

Pengujian secara acak dilakukan untuk membandingkan dua perlakuan baru bertujuan untuk meningkatkan berat badan anak kurang gizi dengan sebuah kelompok kontrol. Manfaatnya adalah terjadi peningkatan rataan berat badan 2.5kg dan standar deviasinya berubah menjadi 3.5kg.

Berapa ukuran sampel yang digunakan, asumsikan kelompok kontrol lebih besar 2 kali dari masing-masing kelompok perlakuan dan power 80% digunakan untuk setiap perbandingan?

# BAB 25

# Dokumentasi

Data dapat dianalisis secara interaktif seperti yang disajikan pada bab-bab sebelumnya atau dalam model kumpulan seperti ditunjukkan pada bab ini.

## *Menggunakan analisis interaktif*

Dalam model interaktif, jenis analisis member perintah secara langsung ke dalam console dan, jika tidak terdapat error, akan diperoleh output spesifik untuk perintah tersebut. Hal ini berguna untuk pengguna pemula. Mengetik dan membaca perintah dari console merupakan proses pembelajaran natural. Fase pembelajaran dalam pengetikan perintah perintah ini terkadang sering mengalami kesalahan, baik secara sintaks atau lainnya. Kesalahan yang paling umum terjadi adalah erros sintaks atau penyalahgunaan aturan yang ditetapkan oleh software. Contohnya termasuk tanda kurung yang tidak sama, tanda kutip tidak seimbang dan kelalaian pembatas (seperti koma). Kesalahan ini sangat mudah untuk dikoreksi. Pengguna dapat dengan mudah menekan tanda panah atas untuk mengambil perintah sebelumnya

Pada tahap awal analisis, analist perlu berkenalan dengan dataset dan variabel. Tahap ini sering disebut 'Exploratory data analysis', yang dilakukan secara

interaktif. Analisis ini dapat dilakukan dalam Epicalc dengan mengikuti langkah berikut:

Awali dengan pembersihan memori:

```
> zap()
```

Muat library yang diperlukan untuk beberapa tujuan analisis seperti `library(survival)` untuk analisis data survival, `library(nlme)` and `library(MASS)` untuk pemodelan multi-level.

## *Membaca file data*

Jika dataset dalam format EpiInfo (file extension = ".rec"), Stata (".dta"), SPSS (".sav"), atau nilai yang dipisahkan koma (".csv") maka akan lebih mudah membaca file data dengan menggunakan perintah *use*`("filename")` dari library Epicalc.

Jika data berada pada format yang lain, periksa apakah baris pertama merupakan *header* (nama variabel) dan periksa tipe pemisah variabelnya. Perintah yang sesuai untuk membaca dataset adalah `read.table` dari library **base**.

Untuk format file data yang lainnya, ketik:

```
help.start()
```

Pilih 'packages' dan kemudian pilih 'foreign'.

Eksplorasi kelas dan deskripsi dari variabel menggunakan *des*`()`. Untuk lebih cepat mengeksplorasi ringkasan statistik variabel gunakan *summ*`()`.

Eksplorasi setiap variabel dalam satu waktu menggunakan perintah *summ*`(varname)`. Perhatikan pula minimum dan maksimum serta perhatikan grafik untuk melihat distribusi secara jelas. Jelajahi juga variabel kategori menggunakan *codebook*`()` dan *tab1*`(varname)`.

Simpan perintah yang telah diketik menggunakan `savehistory("filename")`. File yang telah tersimpan memiliki ekstensi ".r" atau ".rhistory". File ini menyimpan semua perintah yang telah diketik. Perintah-perintah ini akan digunakan untuk analisis lebih lanjut.

Ingat bahwa 'varname' dan 'filename' pada daftar diatas seharusnya digantikan

dengan nama variabel dan nama file yang sesuai. Perintah diketik selama waktu interaktif sering mengandung error atau kesalahan. Karena perintah ini akan digunakan lagi selanjutnya, maka perintah error tersebut harus dibersihkan menggunakan editor teks. Langkah selanjutnya adalah membuka file yang telah tersimpan menggunakan sebuah editor teks. Editor teks yang direkomendasikan adalah Crimson Editor dan Tinn-R.

## Editor Crimson

Ada banyak teks editor yang baik tersedia untuk mengedit file perintah. Teks editor yang baik harus mampu menunjukkan jumlah baris dan tanda kurung yang sesuai. Program Notepad yang merupakan bagian Windows tidak memiliki fitur ini dan program tersebut tidak cocok untuk file dengan perintah yang sangat panjang. Program-program yang disarankan saat ini adalah Crimson Editor dan Tinn-R, yang keduanya merupakan software yang umum ditemukan.

Gunakan Windows Explorer untuk membuat file teks. Secara default, Windows akan menawarkan penamaan file, misalkan 'New Text Document.txt'. Jangan gunakan nama ini. Lebih baik memilih nama yang sesuai dengan tujuan file, misalnya 'Chapter1' atau 'HIV' dan pastikan anda memasukkan ekstensi '.R' atau '.r'. Doubel klik pada file baru ini. Jika asosiasi komputer anda telah berhasil diset up, komputer anda harus membuka file tersebut dalam Crimson Editor atau Tinn-R. Jika tidak, klik kanan dan pilih 'Open with' kemudian pilih Crimson Editor (cedt.exe) atau Tinn-R (Tinn-R.exe).

Bagian berikut hanya membahas tentang Crimson Editor. Anda dapat menggunakan file baru ini untuk penyesuaian Crimson Editor dan file yang diinginkan.

Pilih 'View', 'Tool bars/Views'. Periksa 'Tool bar', 'MDI file tabs' dan 'Status bar'. Jika anda ingin mengetahui apa yang mereka kerjakan, tinggal hapus centangan satu per satu.

Ingat bahwa Crimson Editor mampu menampung beberapa file yang dibuka secara bersamaan. Setiap file yang telah dirubah namun belum disimpan akan ditandai oleh titik merah pada tab File MDI. Titik ini akan menjadi hijau setelah file disimpan.

Dari menu bar pilih 'Document', 'Syntax types'. Perhatikan jika **R** berada dalam daftar tipe file yang dikenal. Jika tidak , pilih 'Customize...' pada bagian paling bawah daftar. Kotak dialog 'Preference' akan muncul bersamaan dengan 'Syntax Type' dibawah option 'File'. Dalam daftar 'Syntax Type', geser kebawah hingga terlihat posisi '-Empty-' pertama dan pilih dengan menggunakan mouse. Posisikan kursor pada kotak teks 'Description' dan ketik R dan ketik R. Di dekat 'Lang Spec', ketik 'R.spc', dan untuk 'Keywords' ketik 'R.key'. Terakhir klik 'OK'. Spesifikasi bahasa dan kata kunci untuk program **R** akan tersedia untuk setiap file yang dibuka menggunakan Crimson Editor. Tetapi file perintah tetap belum secara otomatis berasosiasi dengan Crimson Editor. Pengguna harus mengaktif ini dengan mengklik 'Document', 'Syntax types' dan pilih 'R' dalam daftar.

Terakhir, untuk jumlah baris, klik 'Tool' pada menu bar dan 'Preferences...'. dalam kota 'Preferences...'., sorot 'Visual'. Pilih 'Show line numbers', 'Highlight active line' dan 'Highlight matching pairs'.

## Tinn-R

Manfaat menggunakan Tinn-R jika dibandingkan Crimson Editor adalah kemampuannya untuk beinteraksi dengan dirinya sendiri. Pengguna dapat mengetik perintah dalam editor Tinn-R dan mengirim mereka ke dalam **R** console baris per baris, diblok garis, atau bahkan keseluruhan file perintah. Tinn-R memiliki banyak fitur menarik yang sama dengan Crimson Editor yang membuat bekerja menggunakan **R** lebih mudah dan nyaman.

Memeriksa jumlah baris sangat disarankan oleh penulis. Hal ini diatur pada menu View. Mereka yang ingin menggunakan fungsi kunci dengan memakai mouse 'hotkeys' **R**, dibawah menu R . Preferensi penulis adalah untuk mengatur F2 untuk pengiriman garis tunggal, F4 untuk mengirim blok yang dipilih, F5 untuk mengirim file perintah saat ini yang belum disimpan dan F6 untuk menyimpan file dan mengirim sebagai 'source'. Fungsi kunci F3 untuk pencarian (dan mencari lagi).

### *Mengedit file perintah*

File perintah dapat menjadi sangat mudah atau rumit, bergantung pada sifat pekerjaannya. Untuk file perintah dengan beberapa langkah mudah, tingakt

kesulitannya tidak tinggi. Pengeditan juga tidak telalu susah. Tugas editing dilakukan dengan cara berikut:

Buka file history yang telah disimpan menggunakan Crimson Editor atau Tinn-R. periksa beberapa baris yang memiliki sintaks yang salah. Baris terakhir 'savehistory("filename.r")' harus dihapus karena itu tidak dibutuhkan dalam model kumpulan (batch mode).

Periksa kesalahan pengetikan dengan menghapus baris perintah yang error.

Hilangkan semua perintah yang double.

Periksa struktur perintahnya. Pastikan perintah tersebut terdiri dari urutan perintah yang benar seperti yang disebutkan diatas (dengan *zap*, *use*, dll) .

Jika anda menggunakan Crimson Editor, anda dapat menyalin blok perintah dan menempelkannya pada **R** console. Jika anda menggunakan Tinn-R, anda cukup menyorot perintah yang ingin anda kirim ke **R** dengan menggunakan mouse, klik ikon “send” (atau tekan hotkey) untuk menampilkan operasinya. Copi dan paste memiliki manfaat untuk melihat perbedaan warna perintah (red) dan output (blue) pada **R** console. Bagaimanapun, kesalahan atau error di tengah-tengah blok besar perintah mungkin luput dari perhatian. Jika blok perintah mengandung error, maka simpan dan kirim perintah tersebut sehingga *source* akan berhenti pada baris yang menagndung error. Misalkan,

```
Error in parse(file, n = -1, NULL, "?") : syntax error at
3: library(nlme
4: use("Orthodont.dta")
```

Laporan sintaks yang error biasanya mengandung nomor baris yang error. Pada contoh diatas, error terjadi pada baris 3 (hilang tanda kurung tutup). Kembali pada file perintah dan buat koreksi yang tepat diikuri penyimpanan file.

Bahkan pada saat semua sintaks error telah dihapus, bisa saja masih terdapat tipe error lainnya, seperti kesalahan pengetikan perintah, objek tidak ditemukan atau file yang tidak bisa dibuka. Pada situasi ini, console akan memberikan hasil dalam console pada baris error. Tetapi nomor baris tidak akan diketahui. Kembali pada file perintah dan koreksi error kemudian kembali ke **R** console dan jalankan kembali perintah berikut:

```
> source("filename.r", echo=TRUE)
```

Baris yang harus dihilangkan oleh **R**, seperti komentar penulis atau perintah

yang ingin diabaikan oleh analis untuk saat ini dapat dimulai dengan #. Sangat direkomendasikan untuk memasukkan komentar ke dalam file perintah agar memudahkan pembaca lainnya lebih mudah mengikuti.

Jumlah perintah yang diketik pada file perintah harus optimal. Ini merupakan latihan yang baik untuk memiliki baris perintah yang baru yang mengandung satu kumpulan tindakan serupa. Misalkan, perintah untuk membuat variabel kategori yang baru dari variabel kontinu dan memeriksa distribusi variabel baru tersebut (menggunakan `tabl(newvar)`) harus diletakkan bersamaan. Eksekusi file perintah pada tahap ini akan memudahkan analis untuk mengecek bagian ini secara cepat. Setelah variabel baru terbentuk, baris `tabl(newvar)` mungkin tidak diperlukan lagi dan dapat dihapus atau diabaikan dengan menambahkan '#' sebelumnya.

Salah satu manfaat **R** adalah kemampuan grafiknya. Membuat grafik dapat melibatkan banyak langkah pengerjaan dan mungkin diperlukan penambahan parameter grafis. Ini merupakan ide yang bagus untuk membuat grafik sederhana dalam sebuah perintah. Parameter lainnya seperti 'pch' (point character), 'lty' (line type), 'xlab' (X-axis label), 'col' (colour) dll, dapat ditambahkan pada pengeditan perintah selanjutnya. Akhirnya, grafik yang bagus membutuhkan beberapa baris perintah untuk menghasilkan grafik tersebut.

## *Breaking di dalam file perintah*

Karena ada beberapa perintah dalam file perintah yang dijalankan secara terus menerus, hasil yang keluar dari console bisa terlalu banyak disimpan dalam *buffer* console atau terlalu banyak dibaca. Sebuah grafik yang dibentuk dari baris perintah juga akan ditimpa oleh grafik sebelumnya. Hal ini sering dibutuhkan untuk menginterupsi file perintah yang bertujuan untuk melihat rincian atau grafik dibeberapa titik. Untuk melakukannya, masukkan sebuah baris dimana *break* dibutuhkan. Ketik beberapa sebarang kata seperti 'xxx' pada baris tersebut. Simpan dan jalankan file. Saat file perintah dieksekusi pada titik ini, **R** tidak mengerti apa yang harus dilakukan dan akan berhenti dengan pesan error. Output sebelum 'xxx' dapat dieksplorasi dan setiap grafik yang baru ditampilkan akan disimpan.

Perintah pada **R** console `source("filename.r")` untuk menjalankan file perintah dapat dengan mudah di ulang dengan menekan tanda panah atas

kemudian <Enter>. Mengubah titik *break* 'xxx' dari satu tempat ke tempat yang lain dalam file perintah diikuti dengan penyimpanannya dan *resourcing* pada **R** console merupakan metode standar penggunaan yang baik dari file perintah yang ada.

## *Eksekusi pada bagian tertentu file perintah*

Metode diatas, sekali diterapkan, memastikan bahwa bahwa file perintah tidak memiliki sintaks yang error dan sistem bekerja dengan baik hingga titik 'xxx'. Bagaimanapun metode ini memungkinkan penambahan waktu yang banyak jika beberapa file data dan file perintah terlalu besar atau proses perhitungan cukup lama oleh CPU. Terkadang, analis perlu memotong sesi-sesi sebelumnya untuk mendapatkan hasil yang cepat pada sesi di bagian file perintah selanjutnya. Ini dapat dilakukan jika dan hanya jika sesi sebelumnya bukan merupakan persyaratan dari sesi yang dibutuhkan. Misalkan, sesi yang diawali dengan *`zap()`* atau `rm(list=ls())` akan menghapus semua objek dan lampiran. Setiap bagian sebelumnya mungkin bisa dilewati tanpa banyak masalah.

## *Memotong beberapa baris pada file perintah*

Jika beberapa baris harus dipotong tetapi tanpa menghapusnya, langkah yang paling mudah adalah dengan meletakkan # didepannya. Namun, jika terlalu banyak baris yang harus dipotong dan semua baris tersebut berdekatan, pemotongan dapat dilakukan dengan '`if(){...}`'. jika ekspresi dalam kurung pertama adalah FALSE, maka semua perintah dalam kurung kurawal {} akan dipotong. Jadi untuk memotong bagian yang besar, cukup memasukkan satu baris dengan perintah:

```
if(FALSE){
```

sebelum bagian yang ingin dipotong, dan sebaris dengan kurung kurawal pada akhir sesi. Keseluruhan sesi mengandung kurung kurawal akan diabaikan.

Masalah utama pada metode ini adalah menemukan dan menghapus kurung kurawal yang sesuai ketika pemotongan tidak lagi diperlukan dan file perintah sudah tidak terpakai untuk waktu yang lama. Crimson Editor dan Tinn-R memiliki fitur untuk mencocokkan tetapi kurung buka taua kurung tutup yang dicari mungkin saling berjauhan satu sama lain. Untuk mencegah terjadi

kebingungan, beberpa baris kosong harus dimasukkan sebelum baris perintah 'if(FALSE){' dan setelah tanda kurung tutup yang sesuai. Garis-garis kosong akan membuat bagian yang diabaikan mudah dideteksi.

## Menyimpan teks output

Terdapat sejumlah metode untuk menyimpan teks output.

Cara yang paling mudah adalah menandakan area teks menggunakan mouse dan menyalinnya ke dalam clipboard sebelum ditempelkan pada area yang merupakan bagian teks dokumen.

Metode alternatif adalah dengan menggunakan perintah `sink(file = "myFile.txt")` untuk mengubah semua teks output menjadi bentuk file yang dinamakan "**myFile.txt**". Lihat 'help(sink)' untuk keterangan lebih lanjut tentang penggunaan fungsi ini. Untuk kembali pada mode interaktif, yaitu penghentian konversi dari output menjadi file, gunakan perintah `sink()`. Penggunaan `sink` tergabung dalam file perintah atau dilakukan secara manual.

Kelemahan penggunaan `sink` muncul ketika tidak ada error yang terjadi dalam perintah berikutnya. Karena outputnya dirubah dalam bentuk file, bukan pada layar, penggunaan tidaka akan mendeteksi error dan proses akan berhenti. Jika hal ini terjadi, solusinya dengan mengetik `sink()` dalam console. Perintah ini akan mengembalikan ke dalam rute yang benar pada layar. Error dapat diselidiki dalam file output. Untuk mencegah hal ini, perintah `sink` harus digunakan hanya saat semua perintah yang telah diuji bebas dari error, misalkan tidak ada lagi 'xxx' atau lainnya. Perintah `sink(file="myFile.txt")` dapat diletakkan pada awal file perintah dan `sink()` diletakkan pada akhir file tersebut. Kemudian kirim file perintah itu ke dalam **R** menggunakan perintah `source("command file")`.

Mungkin cara yang paling sederhana dan cara terbaik untuk menyimpan output teks adalah dengan dengan memilih 'Save to File...' pada menu bar. Cara ini langsung menyimpan semua output pada console menjadi file teks. Tujuan file default adalah "**lastsave.txt**" dan nama tersebut bisa dirubah.

***Catatan**: _____*

*Metode terakhir tidak akan menyimpan output jika perintah 'clear console' telah digunakan. Sebagai tambahan, terdapat batas untuk jumlah baris yang dapat disimpan. **R** membatasi jendela console 5000 atau kurang baris yang dapat disimpan. Oleh karena itu, metode ini digunakan hanya jika output yang dihasilkan tidak terlalu panjang.*

## Menyimpan grafik

Penyimpanan perubahan grafik menjadi file lebih sederhana daripada penyimpanan perubahan teks output. Menyalin garfik pada clipboard dan kemudian menempelkannya pada sebuah program misalkan dokumen Word atau pada slide presentasi PowerPoint adalah sangat mudah. Klik jendela grafik dan pilih 'File' dari menu bar dan 'Copy' pada clipboard. Pilih bentuk Bitmap atau Metafile jika software yang dituju menerima format ini. Metafile sedikit lebih kecil dalam hal ukuran dan baris yang lebih tajam. Format Bitmap tidak memiliki gambar yang tajam saat ukurannya diperbesar. Sebagai alternative, grafik dapat disimpan dalam berbagai format lain seperti JPEG, postscript atau PDF.

Untuk menyimpan grafik saat perintah dijalankan dari file tertentu, ketik 'xxx' setelah perintah grafik untuk menghentikan eksekusi perintah lebih lanjut. Kemudian salin atau simpan grafik seperti yang disebutkan diatas.

Atau, daripada menunjukkan grafik di layar, lebih baik jika grafik dialihkan ke dalam sebuah file dengan menggunakan salah satu perintah grafis berikut:

```
bmp("filename.bmp")
jpeg("filename.jpg")
png("filename.jpg")
win.metafile("filename.wmf")
pdf("filename.pdf")
```

Setiap perintah ini menyediakan perangkat grafis dan harus diikuti oleh perintah yang menghasilkan grafik sebenarnua. Saat perintah yang menghasilkan grafik dieksekusi, jangan lupa untuk mematikan perangkat untuk menulis konten grafik kedalam file dan *rerouting* output grafik pada layar.

```
dev.off()
```

Metode *rerouting* berguna karena keseluruhan proses dari file perintah tidak perlu diinterupsi ditengah jalan dengan metode yang disebutkan dalam paragraph sebelumnya.

Konsep pengubahan perangkat grafis setelah membuat grafik serupa dengan menggunakan perintah the `sink`, yang membutuhkan `sink()` terakhir untuk menyimpan dan menutup file. Perintah dibawah menghasilkan grafik summary dari variabel 'age' dari dataset **`Outbreak`** dalam Epicalc. Grafik diarahkan pada file yang dinamakan "**graph1.jpg**".

```
> zap()
> data(Outbreak)
> use(Outbreak)
> jpeg("graph1.jpg")
> summ(age)
> dev.off()
```

Proses re-routing dapat diselesaikan baik secara interaktif atau dalam sebuah perintah jika tidak terdapat error dalam perintah grafik.

# BAB 26

# Strategi Penanganan Data Berukuran Besar

Data yang disajikan dalam paket Epicalc dan digunakan dalam buku ini relatif kecil, baik dari segi jumlah unsur maupun jumlah variabel. Dalam kehidupan sehari-hari, seorang analis data sering menangani lebih dari 50 variabel dan ribuan elemen data. Dalam proses analisis ini dibutuhkan memori komputasi dalam jumlah yang sangat besar, CPU yang bekerja cepat, hard disk yang besar, serta strategi yang efisien dalam menangani data. Tanpa persyaratan tersebut, proses analisa data dapat memakan waktu yang terlalu lama atau bahkan tidak mungkin untuk dilakukan.

## *Menghapus Memori R*

**R** dapat menangani objek yang banyak dalam satu waktu. Jika jumlah memory yang digunakan terbatas, objek-objek yang tidak penting dapat dihapus dari lingkungan kerja dan dihilangkan dari seluruh kerangka data yang tidak penting. Oleh karena itu, untuk memulai sebuah program sebaiknya gunakan perintah berikut ini.

```
> zap()
> detachAllData()
```

## Simulasi Data Berukuran Besar

Untuk menghindari penggunaan data yang terlalu besar, buatlah kerangka data yang baru yang terdiri dari 30.000 elemen dan 161 variabel. Fungsi `rnorm` digunakan untuk menghasilkan bilangan-bilangan acak dari sebuah distribusi normal yang standar.

```
> data1 <- rnorm(30000*160)
> dim(data1) <- c(30000, 160)
> data1 <- data.frame(id=1:30000, data1)
```

Variabel yang pertama disebut 'id', sedangkan nama untuk variabel lainnya dapat ditentukan dengan menggunakan dua `for` loop berlapis dan konstanta **R** yang sudah tersedia yaitu '`letters`', yang terdiri dari sejumlah huruf kecil dalam alfabet Bahasa Inggris. Loop yang paling luar menghasilkan karakter pertama untuk setiap penamaan variabel-variabel tersebut (a – h). Loop yang paling dalam memberikan angka 1 – 20 untuk huruf-huruf ini. Dalam hal ini, huruf dan angka dipisahkan oleh sebuah titik.

```
> namesVar <- NULL
> for (i in letters[1:8])
  {
    for(j in 1:20){
      namesVar <- c(namesVar, paste(i, j, sep="."))
    }
  }
> names(data1)[2:161] <- namesVar
```

Kemudian definisikan setiap variabel dengan menggunakan fungsi `attr`. Proses ini biasanya hanya memerlukan waktu yang sangat singkat, tergantung pada tingkat kecepatan kerja komputer yang digunakan.

```
> attr(data1, "var.labels")[1] <- "ID number"
> for(i in 2:161){
   attr(data1, "var.labels")[i] <- paste("Variable No.", i)
  }
> use(data1)
```

### *Menentukan Subhimpunan dari Sekumpulan Variabel*

Setelah memasukkan perintah-perintah pada peti **R**, biasanya akan bergulir keluaran yang besar pada layar, yang menunjukkan pemandangan yang aneh. Untuk menunjukkan sebuah subhimpunan dari variabel dalam kerangka data

tersebut, masukkan argumen 'select' dalam fungsi *des*.

```
> des(select=1:20)
```

Dengan demikian, yang akan muncul hanyalah 10 variabel pertama beserta kelas dan keterangannya. Setelah itu kita akan melihat dua puluh variabel berikutnya.

```
> des(select=21:40)
```

... dan seterusnya. Dengan melihat sekilas sekitar 20 variabel pada satu waktu, maka user dapat memperhatikan penjelasan dari variabel-variabel tersebut dengan lebih cermat, tanpa harus menggulirkan kursor ke atas dan ke bawah.

Jika hanya ingin melihat variabel yang namanya diawali oleh huruf “a”, ketik:

```
> des(select="a*")
```

Dalam hal ini, variabel yang dimaksud berjumlah 20.

Untuk melihat penjelasan dari variable yang dimulai dengan "a." yang hanya diikuti oleh satu karakter, ketik:

> des(select="a.?")

## *Menentukan Satu Sub-sampel*

Bekerja dengan data yang sangat besar dapat menghabiskan waktu yang sangat lama. Ketika mencoba menjalankan perintah **R,** akan lebih baik jika subhimpunan dari elemen-elemen data ditentukan sehingga dapat mengurangi durasi waktu yang diperlukan. Ketika perintah tersebut bekerja dengan baik, maka perintah ini dapat diterapkan pada seluruh himpunan data lainnya. Fungsi Epicalc *keepData* dapat digunakan untuk menentukan sebuah subhimpunan dari elemen-elemen data dari kerangka data secara keseluruhan.

```
> keepData(sample=300)
```

Jumlah elemen dari kerangka data **.data** akan diubah dari 30.000 menjadi 300 dengan jumlah dan keterangan variabel yang sama, seperti yang tertulis berikut ini.

```
> des(.data)
```

Perhatikan bahwa beberapa baris pertama membaca:

```
  (subset)
```

```
No. of observations =300
    Variable      Class           Description
1   id            integer         ID number
2   a.1           numeric         Variable No. 2
========= lines omitted=========================
```

yang menunjukkan bahwa **.data** merupakan sebuah subhimpunan dari elemen data yang semula.

Jika ingin menggunakan kerangka data semula, ketik:

```
> use(data1)
```

Sebuah alternatif untuk menentukan jumlah dari elemen untuk disimpan secara acak adalah dengan menentukan sebuah persentase dari elemen data yang semula.hal ini dapat dilakukan dengan memilih sebuah bilangan antara 0 dan 1 untuk argumen 'sample'.

```
> keepData(sample=0.01)
```

Perintah di atas hanya akan mengumpulkan 300 elemen dari jumlah elemen data semula. Kriteria untuk menyimpan subhimpunan elemen data dapat ditentukan dengan menggunakan argumen 'subset':

```
> keepData(subset=a.1 < 0)
```

Dengan demikian, akan terlihat pengurangan dari jumlah elemen data, bukan variabel.

```
> des()
```

Banyaknya pengurangan tersebut kira-kira setengahnya mengingat variabel 'a.1' dihasilkan dari sebuah distribusi normal yang standar, yang memiliki nilai rata-rata 0 dan simetris mengenai rata-rata ini.

Metode penentuan subhimpunan ini dapat diaplikasikan pada kerangka data yang sebenarnya, contohnya membuat subhimpunan berdasarkan jenis kelamin atau sebuah kelompok usia tertentu.

### *Pengeluaran Data Data exclusion*

Fungsi *keepData* dapat digunakan untuk mengeluarkan variabel. Kembali pada data semula dan keluarkan variabel antara 'a.1' and 'g.20'.

```
> use(data1)
> keepData(exclude = a.1:g.20)
> des()
```

Variabel dari 'a.1' to 'g .20' sudah dikeluarkan, tetapi harus diingat bahwa jumlah elemen datanya tetap sama.

Untuk mengeluarkan 10 elemen terakhir dari setiap sesi, fitur wildcard pada Epicalc dapat digunakan.

```
> use(data1)
> keepData(exclude = "????")
> des()
```

Semua variabel yang namanya terdiri dari empat karakter sudah dikeluarkan.

Sebagaimana yang telah dibahas sebelumnya, jika ukuran dari kerangka datanya sangat besar, analis dapat memilih satu atau lebih dari strategi di atas untuk mengurangi ukuran tersebut. Dengan demikian, analisis yang lebih jauh dapat dilakukan dengan lebih cepat. Jika semua perintah didokumentasikan di dalam sebuah file seperti yang ditunjukkan pada bab sebelumnya, dan perintah tersebut terorganisir dengan baik, beberapa baris pertama dari file tersebut dapat diedit untuk menggunakan seluruh kerangka data semula dalam proses analisa akhir.

BAB 27

# Menyusun Tabel untuk Naskah

Pembaca buku ini mungkin bertanya-tanya mengapa uji statistik sederhana seperti uji-t, uji chi-squared dan uji-uji non-parametrik jarang disebutkan atau dibahas secara detil di sini. Uji-uji tersebut sering digunakan dalam perbandingan awal kelompok-kelompok, yang ditampilkan sebagai tabel pertama dalam kebanyakan naskah epidemiologis. Semua uji statistik ini dapat dihasilkan oleh satu perintah Epicalc tunggal, `tableStack`.

Pada bab 23, perintah ini digunakan secara luas yang paralel dengan perintah `alpha` dan `alphaBest` untuk menampilkan distribusi dari setiap variabel. Tujuan lainnya yang tidak kalah pentingnya ialah untuk menghitung rata-rata dan nilai total di mana elemen-elemennya dibalik secara benar ketika dibutuhkan.

Pada bab ini, fungsi yang sama juga digunakan secara luas tetapi argumen 'by' dimasukkan. Hasilnya dapat langsung dimasukkan ke dalam naskah.

## Konsep 'tableStack'

Naskah yang epidemiolis dan klinis sering memiliki tujuan untuk menguji hipotesis tertentu dalam subjek-subjek kemanusiaan. Subjek-subjek ini biasanya dikelompokkan berdasarkan jenis paparan (dalam sebuah kelompok atau study intervensi) atau hasil (dalam studi kontrol kasus) dari kepentingan. Variabel pengelompokkan ini mula-mula dianalisa untuk mengamati karakteristik dasar pada tabel pertama dari naskah dan untuk melihat varipabel-variabel dari uji hipotesis pada tabel kedua. Orientasi dari tabel-tabel tersebut biasanya memiliki variabel kelompok sebagai kolom dan variabel lainnya sebagai baris.

Dalam prakteknya, jika variabel pada baris merupakan sebuah faktor, maka tabulasi-silang dari variabel-variabel tersebut dapat diperoleh dengan menggunakan fungsi `table` dari **base** library atau *`tabpct`* dari Epicalc. Hal ini tergantung pada uji statistik dengan menggunakan uji chi-squared atau uji Fisher's exact.

Jika variabel baris berada pada skala kontinyu, tabel yang dibutuhkan dapat diperoleh dengan menggunakan fungsi `tapply` atau `aggregate` dalam paket **base** dan **stats** dari **R**, berturut-turut, yang memberikan satu statistik dari setiap subgrup pada satu waktu atau *`aggregate.numeric`* dari paket Epicalc, yang menunjukkan sejumlah statistik dari subgrup tersebut. Jika data terdistribusi dengan normal, nilai rata-rata dan standar deviasi merupakan dua statistik yang biasanya muncul. Untuk data yang miring atau tidak berdistribusi normal, median dan jarak interkuartil (persentil ke-25 dan ke-75) lazim digunakan. sementara itu, untuk data yang berdistribusi normal, digunakan uji-t (untuk melakukan pengujian antara dua grup) dan anova satu arah (untuk melakukan pengujian lebih dari dua grup). Untuk data yang tidak normal, uji-uji non-parametrik sering diandalkan, seperti uji penjumlahan bertingkat Wilcoxon untuk 2 grup dan uji Kruskal-Wallis untuk lebih dari 2 grup.

Ketika mengerjakan hal di atas, analis harus melewati berbagai tahapan eksplorasi distribusi, menghitung statistik yang berbeda untuk subgrup dan kemudian menyalin hasilnya ke dalam naskah, biasanya dilakukan dengan format durasi waktu tertentu. Proses yang melelahkan ini dapat diselesaikan melalui fungsi Epicalc *`tableStack`*, yang membuat dan mengatur beberapa tabel dengan statistik yang sesuai ke dalam satu tabel yang cocok.

### *Contoh*

Semua himpunan data dengan sedikitnya satu variabel faktor dapat digunakan untuk percobaan. Mari memulai dengan himpunan data **`Familydata`**, yang merupakan sebuah himpunan data kecil yang digunakan pada bab 4.

```
> zap
> data(Familydata)
> use(Familydata)
> des()

Anthropometric and financial data of a hypothetical family
No. of observations = 11
  Variable      Class           Description
1 code          character
2 age           integer         Age(yr)
3 ht            integer         Ht(cm.)
4 wt            integer         Wt(kg.)
5 money         integer         Pocket money(B.)
6 sex           factor
```

Data tersebut hanya memiliki satu variabel faktor, yaitu 'sex'. Berikut ini adalah sebuah tabel yang terdiri dari seluruh variabel berdasarkan jenis kelamin dalam sebuah format yang tersusun rapi.

```
> tableStack(vars=2:5, by=sex)
                F             M            Test stat.        P
  value
Age(yr)                                    t (9 df): t = 0.5
  0.627
  mean(SD)     42.9(24.3)     50.8(26.6)

Ht(cm.)                                    Rank sum: W = 0.5
  0.014
  median(IQR)  155(150.5,159) 168.5(166,170.5)

Wt(kg.)                                    Rank sum: W = 3
  0.047
  median(IQR)   51(50.5,54)    65.5(61,68)

Pocket money(B.)                           t (9 df): t = 1.33
  0.218
  mean(SD)      586.4(656.1)   1787.5(2326.1)
```

Argumen numerik 'vars' dapat digantikan dengan nama-nama variabel.

```
> tableStack(age:money, by=sex)
```

Tabel keluaran terdiri empat variabel yang berasal dari ke-2 hingga ke-5 (vars=2:5) pada himpunan data tersebut. Usia terdistribusi dengan normal, dengan demikian uji-t diperlukan untuk menguji perbedaan antara usia rata-rata dari laki-laki dan perempuan. Statistik ujinya kecil (t=0.5), dengan derajat kebebasan 9 dan nilai P yang tidak signifikan.

Tinggi dan berat antara laki-laki dan perempuan jelas berbeda. Kedua variabel memiliki distribusi tak normal sehingga yang muncul adalah mediannya, bukan mean. Jarak interkuartil (IQR) juga turut muncul, namun tidak untuk standar deviasi (SD). Selain itu, dalam hal ini yang digunakan adalah uji penjumlahan bertingkat Wilcoxon, bukan uji-t..

Uang saku berdistribusi normal dan telah dilakukan uji-t dengan hasil yang tidak signifikan. Perlu diingat bahwa untuk ukuran sampel yang kecil seperti contoh di atas, kesimpulan kita tidak begitu kuat.

Untuk melihat asumsi terhadap normalitas dari sisa ‘money’, ketik

```
> shapiro.test(lm(money ~ sex)$residuals)

        Shapiro-Wilk normality test

data:  lm(money ~ sex)$residuals
W = 0.8722, p-value = 0.08262
```

Selain itu, asumsi dari varian yang sama dari sisa dapat dilihat melalui

```
> bartlett.test(money ~ sex)

        Bartlett test of homogeneity of variances

data:  money by sex
Bartlett's K-squared = 5.8683, df = 1, p-value = 0.01542
```

Epicalc menetapkan tingkatan yang jelas untuk uji Shapiro-Wilk dan uji Bartlett yang dipergunakan untuk menukar hasilnya dengan tidak lagi menggunakan uji-t tetapi beralih pada uji penjumlahan bertingkat Wilcoxon pada P>0.01, bukan P>0.05. Perintah yang terakhir mempunyai nilai P sebesar 0.015, tapi tidak cukup untuk melakukan proses penukaran ini.

Percobaan dengan menggunakan variabel-variabel lainnya di dalam himpunan data tersebut dapat membantu memahami alasan memilih uji-uji parametrik dan non-parametrik. Pengguna juga dapat menentukan fitur-fitur keluaran yang berbeda, misalnya pengguna tidak ingin menampilkan hasil uji statistik, nama jenis uji yang digunakan, dan variabel-varabel untuk menggunakan uji-uji statistik non-parametrik. Contohnya:

```
> tableStack(age:money, by=sex, test=FALSE)
> tableStack(age:money, by=sex, name.test=FALSE)
> tableStack(age:money, by=sex, iqr=c(age, money))
```

## *Contoh Lainnya*

Argumen 'by' pada fungsi *tableStack* juga dapat memiliki lebih dari 2 tingkatan.

```
> data(Ectopic)
> use(Ectopic)
> des()

No. of observations = 723
  Variable      Class           Description
1 id            integer
2 outc          factor          Outcome
3 hia           factor          Previous induced abortion
4 gravi         factor          Gravidity

> table(outc)
outc
  EP   IA Deli
 241  241  241

> tableStack(hia:gravi, by=outc, var.labels=FALSE)
               EP         IA       Deli        Test stat.  P
  value
3 : hia                                     Chi(2) = 78.72  <
  0.001
```

```
   never IA  61(25.3)  110(45.6) 158(65.6)
   ever IA  180(74.7)  131(54.4)  83(34.4)

4 : gravi                                        Chi(4) = 46.18  <
  0.001
   1-2      117(48.5)  121(50.2)  182(75.5)
   3-4       87(36.1)   85(35.3)   46(19.1)
   >4        37(15.4)   35(14.5)   13(5.4)
```

Perhatikan ketika 'var.labels' FALSE, yang muncul adalah indeks variabel dan nama variabel, bukan label variabel. Hal ini tidak dapat melakukan proses 'copy & paste' ke dalam naskah, namun berguna untuk eksplorasi data. Kategori abnormal untuk variabel baris seperti tingkatan label yang salah, atau baris dengan angka yang terlalu kecil, dapat menunjukkan sebuah kebutuhan untuk melakukan pengkodean ulang terhadap variabel tersebut sebelum versi terakhir siap untuk digunakan.

Ketika variabel baris merupakan sebuah faktor, tabulasi silang untuk variabel tersebut terhadap variabel ‘by’ muncul. Tabel tersebut menunjukkan 241 EP (wanita dengan kehamilan ectopic). Sebanyak 180 orang dari jumlah tersebut, yaitu 74%, pernah mengalami aborsi. Jumlah ini jauh lebih besar dari kelompok IA (54%) dan kelompok delivery (34%). Uji chi-squared sangat signifikan (P < 0.001). Pada kelompok 'gravi', persentase memiliki 1-2 kali kehamilan sebelumnya lebih besar dari persentase yang dimiliki oleh kelompok 'Deli' dan perbedaan dari gravidity juga sangat signifikan. Hubungan antara keluaran (variabel kolom) dan lebih dari satu variabel baris menunjukkan adanya kemungkinan munculnya permasalahan di luar dugaan yang membutuhkan analisa lebih lanjut.

Pengaturan awal argumen dapat diubah-ubah, seperti membuat keluaran dari pengujian hipotesis menjadi FALSE atau menunjukkan persentase kolom.

```
> tableStack(hia:gravi, by=outc, test=FALSE)
                                  EP        IA      Deli
Previous induced abortion
   never IA                61(25.3) 110(45.6) 158(65.6)
   ever IA                180(74.7) 131(54.4)  83(34.4)
Gravidity
   1-2                    117(48.5) 121(50.2) 182(75.5)
   3-4                     87(36.1)  85(35.3)  46(19.1)
   >4                      37(15.4)  35(14.5)   13(5.4)
> tableStack(hia:gravi, by=outc, test=FALSE, percent="row")
```

Perhatikan bahwa 'percent' harus dijadikan "row" jika ingin membandingkan persentase dari variabel keluaran yang telah dirancang untuk varabel kolom.

Dua kerangka data di atas memiliki variabel baris yang terdiri dari hanya satu tipe, variabel atau faktor kontinyu. Berikut ini adalah percobaan dengan kombinasi dari keduanya.

```
> data(Cars93, package="MASS")
> use(Cars93)
> des()
> tableStack(vars=4:25, by=Origin)
                          USA         non-USA  Test stat.    P
  value
Min.Price                                      Rank sum test
  0.812
  median(IQR) 14.5(11.4,19.4)  16.3(9.1,22.9)

Price                                          Rank sum test
  0.672
  median(IQR) 16.3(13.5,20.7) 19.1(11.6,26.7)

Max.Price                                      Rank sum test
  0.489
  median(IQR)   18.4(15,24.5) 21.7(12.9,28.5)

MPG.city                                       Rank sum test
  0.037
  median(IQR)       20(18,23)       22(19,26)

MPG.highway                                    Rank sum test
  0.191
  median(IQR)       28(26,30)       30(25,33)

AirBags                                        Chi(2) = 0.48
  0.786
  Driver & Passenger  9(18.8)         7(15.6)
  Driver only        23(47.9)        20(44.4)
  None               16(33.3)          18(40)
DriveTrain                                     Chi(2) = 0.17
  0.919
  4WD                 5(10.4)         5(11.1)
  Front              34(70.8)        33(73.3)
  Rear                9(18.8)         7(15.6)

Cylinders                                      Fisher's test
  0.011
  3                      0(0)          3(6.7)
  4                  22(45.8)          27(60)
```

```
  5                      0(0)          2(4.4)
  6                   20(41.7)        11(24.4)
  8                    6(12.5)          1(2.2)
  rotary                 0(0)          1(2.2)
=================== remaining lines omitted =================
```

Beberapa dari variabel tersebut seperti variabel-variabel yang berhubungan dengan price, tingkat pemakaian bahan bakar dan arus, baik itu tidak berdistribusi normal atau memiliki varian yang sangat berbeda antara kedua mobil asalnya, diuji dengan uji penjumlahan bertingkat non-parametrik. Variabel kontinyu lainnya diuji dengan uji-t. Di sini terdapat empat variabel faktor, yaitu lokasi airbags (AirBags), tipe drive train (DriveTrain) dan ketersediaan manual transmission (Man.trans.avail), yang seluruhnya diuji dengan uji chi-squared. Sementara itu, jumlah silinder (Cylinders) mematahkan asumsi-asumsi dari uji chi-squared, sehingga digunakan uji Fisher's exact. Nilai P dua-sisi ini sangat kecil sehingga menunjukkan bahwa pola silinder dari mobil yang berasal dari US dan mobil yang non-US sangat berbeda.

## Kolom total

Jika diperlukan, kolom tambahan dari total dapat dimunculkan.

```
> tableStack(vars=4:25, by=Origin, total.column=TRUE)
```

Dalam hal ini, tabel terlihat lebih baik jika kolom uji dihilangkan.

```
> tableStack(vars=4:25, by=Origin, total.column=T, test=F)

                                 USA          non-USA           Total
Min.Price
  median(IQR)          14.5(11.4,19.4)  16.3(9.1,22.9) 4.7(10.8,20.3)

Price
  median(IQR)          16.3(13.5,20.7) 19.1(11.6,26.7) 7.7(12.2,23.3)

Max.Price
  median(IQR)            18.4(15,24.5) 21.7(12.9,28.5) 9.6(14.7,25.3)

MPG.city
  median(IQR)               20(18,23)       22(19,26)       21(18,25)

MPG.highway
```

```
  median(IQR)              28(26,30)       30(25,33)      28(26,31)

AirBags
   Driver & Passenger       9(18.8)         7(15.6)       16(17.2)
   Driver only             23(47.9)        20(44.4)       43(46.2)
   None                    16(33.3)          18(40)       34(36.6)
================== remaining lines omitted =================
```

Dalam beberapa kasus, hanya kolom total yang berguna ketika ditampilkan. Contohnya, dalam himpunan data **Compaq**, tabel yang pertama dapat menjadi keterangan dari subjek dalam hal tahapan, kelompok usia, jenis kelamin, dll.

```
> data(Compaq)
> use(Compaq)
> des()
> tableStack(vars=4:6, by="none")
                   Total
stage
   Stage 1     530(49.8)
   Stage 2     390(36.7)
   Stage 3       81(7.6)
   Stage 4       63(5.9)

Age group
   <40         296(27.8)
   40-49       285(26.8)
   50-59       243(22.8)
   60+         240(22.6)

ses
   Rich        279(26.2)
   High-middle   383(36)
   Poor-middle 154(14.5)
   Poor        248(23.3)
```

String "none" dapat digantikan dengan nilai yang dikutip dengan hasil yang sama.

```
> tableStack(vars=4:6, by="junk")
```

## Mengirim 'tableStack' dan tabel lainnya ke dalam naskah

**R** memilki fungsi untuk menulis sebuah matriks, tabel atau kerangka data ke dalam sebuah file dari variabel yang dipisahkan oleh koma (csv) yang dapat dibaca oleh Excel. Setelah dibaca dalam Excel, tabel dapat disalin ke dalam naskah dengan mudah.

```
> table1 <- tableStack(vars=4:25, by=Origin, data=Cars93)
> write.csv(table1, file="table1.csv")
> getwd()
```

Perintah yang terakhir menunjukkan direktori kerja yang terkini, yang terdapat dalam file "**table1.csv**". Untuk melihat hasilnya, masuk ke direktori tersebut dan buka file yang dimaksud melalui Excel. Setelah itu, lakukan “copy dan paste” terhadap tabel keluaran ke dalam dokumen naskah.

Teknik ini juga dapat bekerja dengan seri *display* dalam Epicalc, seperti *`regress.display`*, *`logistic.display`*, dan lain-lain.

```
> glm1 <- glm(Origin ~ Price + AirBags + DriveTrain,
  binomial, data=Cars93)
> logistic.display(glm1) -> glm1.display
> attributes(glm1.display)
$names
[1] "first.line" "table"      "last.lines"

$class
[1] "display" "list"

> table2 <- glm1.display$table

> write.csv(table2, file="table2.csv")
```

Kemudian lihatlah hasil yang diperoleh.

# Bab 1

Soal 1

```
> p <- 0.3
> delta <- 0.05
> n <- 1.96^2*p*(1-p)/delta^2 ; n # 322.6944.
```

Maka subjek yang dibutuhkan adalah 323.

Soal 2

```
> p <- .05; delta <- .02
> n <- 1.96^2*p*(1-p)/delta^2 ; n # 456.19
```

Maka dibutuhkan 457 subjek.

Soal 3

```
> log(.01/(1-.01)) # -4.59512
> log(.1/(1-.1))   # -2.197225
> log(.5/(1-.5))   #  0
> log(.9/(1-.9))   #  2.197225
> log(1/(1-1))     #  Inf
```

Cara lain adalah dengan membuat vector yang terdiri atas nilai peluang-peluangnya.

```
> p <- c(.01, .1, .5, .9, 1)
> log(p/(1-p))
[1] -4.5951 -2.1972  0.0000  2.1972     Inf
```

Perhatikan bahwa dalam R fungsi c digunakan untuk menggabungkan nilai-nilai ke dalam suatu vektor. Anda akan menemukan bahwa fungsi ini sangat berguna dan digunakan di dalam buku ini.

## Bab 2

Soal 1.

```
> sum(1:100*1:100)  # or sum((1:100)^2)
[1] 338350
```

Soal 2.

```
> x <- 1:1000
> x7 <- x[x/7==trunc(x/7)]  # or x7 <- x[x%%7==0]
> sum(x7)
[1] 71071
```

Soal 3.

```
> ht <- c(120,172,163,158,153,148,160,170,155,167)
> wt <- c(22,52,71,51,51,60,50,67,53,64)
> names <- c("Niece", "Son", "GrandPa", "Daughter", "Yai",
  "GrandMa", "Aunty", "Uncle", "Mom", "Dad")
> names(ht) <- names
> names(wt) <- names
> cbind(ht,wt)
> bmi <- wt/(ht/100)^2

> sort(bmi)
   Niece      Son   Aunty1 Daughter      Yai
15.27778 17.57707 19.53125 20.42942 21.78649
     Mom      Dad    Uncle  GrandPa  GrandMa
22.06035 22.94812 23.18339 26.72287 27.39226

> summary(bmi)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  15.28   19.76   21.92   21.69   23.12   27.39

> sd(bmi)
[1] 3.742951
```

Kesimpulannya, 'Niece' memiliki BMI terendah yaitu 15.27 kg/m$^2$ dan 'GrandMa' memiliki BMI tertinggi yaitu 27.39 kg/m$^2$. BMI rata-rata adalah 21.7 kg/m$^2$ dan standard deviasi adalah 3.7 kg/m$^2$.

## Bab 3

Soal 1

Ada lebih dari satu metoda yang benar.

Metoda pertama

```
> a1 <- rbind(1:10, 11:20)
> a1
```

Metoda kedua

```
> a2 <- matrix(1:20, nr=2, byrow=TRUE)
> a2
```

Metoda ketiga

```
> a2 <- t(cbind(1:10, 11:20))
> a2
```

Soal 2

```
> a1[,seq(from=1, to=10, by=2)]
```

Soal 3

```
> table1 <- cbind(c(15,30), c(20,22)); table1
> rownames(table1) <- c("Exposed","Non-exposed")
> colnames(table1) <- c("Diseased","Non-diseased")
> table1
> help(chisq.test)
> help(fisher.test)
> chisq.test(table1) # with Yates' continuity correction
> chisq.test(table1, correct=FALSE) # without
> fisher.test(table1) # default atlernative is "two.sided"
> fisher.test(table1, alternative="greater")
> fisher.test(table1, alternative="less")
```

## Bab 5

Pemilihan plot bergantung pada ukuran sampel, Secara mendasar, apa yang perlu dperhatikan adalah mudah dimengerti, membedakan diantara nilai-nilai atau hanya dari tampilannya saja. Hal ini juga bergantung pada siapa penggunanya informasi yang menjadi target.

| **Nilai dari masing-masing elemen pada skala** | |
|---|---|
| Dotchart | Nilai-nilai asalnya semua tersimpan |
| Dotplot | Setiap nilai dipaksa untuk masuk ke dalam *bins.* |
| Box plot | Hanya nilai pencilan yang dimunculkan. Sedangkan nilai-nilai lainnya akan tergabungkan dalam suatu bagian pada *box.* |
| **Power untuk membedakan masing-masing nilai yang tidak seragam** | |
| Dotchart | Diskriminasi berdaya tinggi. Bahkan perbedaan kecil bisa diperhatikan jika ukuran sampel tidak besar |
| Dotplot | Karena nilai-nilai yang berdekatan sering dipaksa masuk ke *bins* yang sama, kekuatan (daya) diskriminasi nya hilang. |
| Boxplot | Daya diskriminasi buruk, karena sebagian besar titik tidak muncul di dalam *box* |

| **Persepsi untuk nilai-nilai dari distribusi frekwensi** | |
|---|---|
| Dotchart | Ruang kosong dalam grafik segera akan menyampaikan informasi bahwa tidak ada data di daerah tersebut. Yang naik rata atau lambat menunjukkan frekuensi rendah sedangkan yang naik tajam atau curam menunjukkan frekuensi tinggi. Pengguna harus dibimbing untuk memberikan interpretasi yang tepat. |
| Dotplot | Grafik ini memberikan informasi terbaik tentang frekuensi relatif. Interpretasinya dapat dilakukan dengan mudah. |
| Boxplot | Panjang kotak adalah kontra-intuitif. Karena kotak ini dibagi menjadi dua bagian dengan kurang lebih jumlah data yang sama, bagian pendek berarti kepadatan tinggi dan bagian panjang berarti kepadatan rendah. Banyak orang tidak memiliki pengetahuan ini |

| | |
|---|---|
| | untuk menafsirkan hasilnya. |

| **Informasi tentang ukuran sampel dalam setiap stratum** | |
|---|---|
| Dotchart | Ketebalan lapisan ditentukan oleh ukuran sampel. |
| Dotplot | Ketebalan lapisan ditentukan oleh ketinggian *bins* yang paling sering, oleh karena itu, dapat secara visual terdistorsi. |
| Boxplot | Ketika 'varwidth = TRUE', seperti yang ditunjukkan dalam perintah, lebar setiap kotak ditentukan oleh ukuran sampel, tetapi tidak dalam proporsi linier.. |

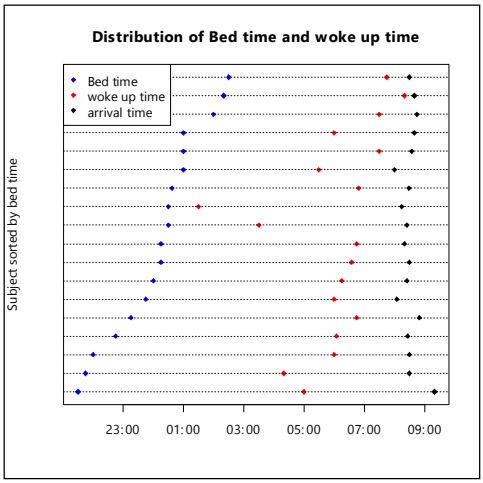
| **Data yang hilang** | |
|---|---|
| Dotchart | Data yang hilang ditempatkan sebagai ruang kosong di bagian atas setiap strata. |
| Dotplot | Data yang hilang tidak ditampilkan. |
| Boxplot | Data yang hilang tidak ditampilkan. |

| **Kesesuaian terkait dengan ukuran sampel dan jumlah strata** | |
|---|---|
| Dotchart | Paling sesuai ketika ukuran sampel tidak terlalu besar misalnya <200. Besar jumlah strata dapat menjadi masalah, terutama ketika ukuran sampel di antara strata yang sangat tidak seimbang. |
| Dotplot | Permasalahannya serupa dengan 'Summ (var)' pada masalah stratifikasi. Namun, 'dotplot' lebih bersahabat ketika ukuran sampel besar |
| Boxplot | Mengingat hanya ada 5 nilai-nilai vektor, grafik ini tidak terbebani oleh ukuran sampel yang besar. Dalam analisis stratifikasi, ukuran sampel strata tidak proporsional dengan lebar kotak bahkan jika 'varwidth = TRUE' dikenakan. Jadi grafik dapat mengakomodasi masalah ini cukup baik. Disisi lain, panjang dari kotak dapat mengelabui ukuran sampel seperti yang disebutkan. Informasi keseluruhan pada ukuran sampel pada umumnya berkurang padabox plot. Simpul Median diperkenalkan untuk menunjukkan |

| | |
|---|---|
| | 95% selang kepercayaan dari median. Sebuah simpul yang lebih kecil menunjukkan ukuran sampel yang lebih besar atau menunjukkan tingkat dispersi yang lebih rendah. Akan tetapi, penggunaan simpul ini tidak populer. |

## Bab 6

```
> zap()
> data(Timing)
> use(Timing)
> bed.day <- ifelse(bedhr > 20, 12, 13)
> bed.time <- ISOdatetime(year=2004, month=12, day=bed.day,
  hour=bedhr, min=bedmin, sec=0, tz="")
> woke.up.time <- ISOdatetime(year=2004, month=12, day=13,
  hour=wokhr, min=wokmin, sec=0, tz="")
> arrival.time <- ISOdatetime(year=2004, month=12, day=13,
  hour=arrhr, min=arrmin, sec=0, tz="")
> from.woke.to.work <- arrival.time - woke.up.time
> summ(from.woke.to.work)
> sortBy(bed.time)
> par(bg="cornsilk")
> plot(bed.time, 1:length(bed.time), xlim=c(min(bed.time),
  max(arrival.time)), pch=18, col="blue", ylab=" ", yaxt="n")
> points(woke.up.time, 1:length(woke.up.time), pch=18, col=2)
> points(arrival.time, 1:length(arrival.time), pch=18, col=1)
> abline(h=1:length(arrival.time), lty=3)
> title(main="Distribution of Bed time and woke up time")
```




```
> title(ylab="Subject sorted by bed time")
> legend("topleft", legend=c("Bed time", "woke up time", "arrival
  time"), pch=18, col=c("blue","red","black"), bg="cornsilk")
```

## Bab 7

No. Seperti yang terlihat dari.

```
> addmargins(table(.data$onset, .data$case))
```

Ada tiga non-kasus yang dilaporkan onset. 'Onset' yang telah berubah adalah vector bebas yang dapat dibuat dengan command ini:

```
> onset[!case] <- NA
```

Dalam command ini, baik 'onset' maupun 'case' ada dalam posisi kedua dalam path '`search()`', yang dapat di *attached* dengan mengkopi **`.data`**. Dari command:

```
> onset[!case] <- NA
```

Ada tiga duplikat 'onset'. Duplikat pertama dan kedua **`.data`** dan dalam 'search()[2]' yang tidak akan berubah. Dua duplikat tersebut saling berbeda dari vektor bebas yang dibuat dengan command berikut:

Untuk mendapatkan permanen efek, *recode* command *recode* dalam Epicalc harus digunakan:

```
> recode(onset, !case, NA)
```

Lalu, cek lagi:

```
> addmargins(table(.data$onset, .data$case))
```

Melalui metode ini , vector bebas 'onset'ini akan dihilangkan. Vector-vektor di dalam **.data** dan di dalam 'search()[2]' juga akan secara otomatis disinkronisasikan ke nilai baru.

Bagaimanapun, variabel, 'time.onset', a *POSIXt* objek kelas, tidak bermasalah. Menggunakan variable di dalam **.data** dalam bab-bab berikutnya dapat dilakukan..

## Bab 8

Beefcurry' dan 'saltegg' keduanya signifikan secara atribut risk dan risk ratio. Hal ini dapat dijelaskan bahwa makanan tersebut telah terkontaminasi. Kenyataannya adalah bahwa peningkatan resiko dari pola konsumsi ini dikarenakan adakan pembauran (confounding). Masalah ini akan didiskusikan pada bab yang lain.

```
> cc(case, water) # OR =1.14, 95%CI = 0.47, 2.85
> table(case, eclair.eat, water)
```

Perhatikan bahwa nilai sell adalah nol untuk kasus orang-orang yang tidak mengkonsumsi kue sus maupun air. Command lanjutan yang berikut memberikan nilai MH odds ratio akan tetapi bukan nilai OR yang spesifik, dan juga memuat hasil dari uji homogenitas.

```
> mhor(case, eclair.eat, water)
  # MH OR = 24.3, 95% CI = 14.11, 41.7

> mhor(case, water, eclair.eat)
  # MH OR = 1.56, 95% CI =  0.60, 4.06
```

Untuk stratifikasi dengan beef curry, tidak ada satupun sell yang bernilai nol. Uji homogenitas memungkinkan untuk dilakukan.

```
> table(case, beefcurry, water)
> mhor(case, beefcurry, water)
```

Cross grafik, Nilai P untuk uji homogenitas = 0.018

```
> mhor(case, water, beefcurry)
```

Cross grafik, Nilai P untuk uji homogenitas = 0.016

*Perhatikan interaksi kuat antara* **beef curry** *dengan eclair dan dengan air, membutuhkan penjelasan dari sudut pandang ilmu biologi.*

## .Bab 10

*Solusi dihilangkan.*

## Bab 11

```
> des()
> plot(smoke, log(deaths))
> plot(SO2, log(deaths))
> plot(log(smoke), log(deaths))
> plot(log(SO2), log(deaths))
```

Dari empat plot yang ada, plot terakhir adalah yang terbaik.

```
> lm1 <- lm(log(deaths) ~ smoke)
> summary(lm1)$r.squared # 0.47
```

```
> lm2 <- lm(log(deaths) ~ SO2)
> summary(lm2)$r.squared # 0.59
> lm3 <- lm(log(deaths) ~ log(smoke))
> summary(lm3)$r.squared # 0.43
> lm4 <- lm(log(deaths) ~ log(SO2))
> summary(lm4)$r.squared # 0.66
```

Nilai R-kuadrat '`lm4`' sama dengan model berikut: (gunakan log berbasis 2):

```
> lm5 <- lm(log2(deaths) ~ log2(SO2))
> summary(lm4)$r.squared # 0.66
```

Koefisien log(SO2) dari '`lm4`' dan log2(SO2) dari '`lm5`' adalah sama: 0.45843.

Untuk setiap kenaikan unit log2 (SO2), maka log2 (kematian) meningkat sebesar 0,458 unit. Demikian pula, untuk setiap peningkatan unit loge (SO2), loge (SO2) juga meningkat sebesar 0,458 unit. Koefisien ini saling independen dari dasar logaritmanya. . Ini berarti bahwa hubungan antara kedua variabel cukup powerful.e. Mengingat x adalah angka positif, untuk setiap x kali kenaikan SO2 , jumlah kematian akan meningkat sebesar $x^{0.45843}$ kali.

```
> plot(log2(SO2), log2(deaths))
> abline(lm5)
```

Dari koefisien regresi dan grafik, , Pada saat konsentrasi SO2 di udarameningkat dua kali lipat, jumlah kematian akan meningkat sebesar. $2^{0.45843}$ atau 1.374 times. Permodelan untuk variable keluaran yang diskret, dapat memberikan hasil yang lebih baik dengan menggunakan regresi Poisson pada bab 19.

## Bab 12

```
> zap()
> data(BP)
> use(BP)
> age.in.days <- as.Date("2001-03-12") - birthdate
> age <- as.numeric(age.in.days)/365.25
> sortBy(sbp)
> plot(sbp,ylim=c(0,max(sbp)),pch=" ",ylab="blood pressure")
> n <- length(sbp)
> segments(1:n, sbp, 1:n, dbp, col=unclass(sex))
> title(main="Systolic and diastolic blood pressure of the
  subjects")
> summary(lm(dbp ~ sex + age))
=======================
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)  48.9647     9.4928   5.158 1.32e-06
sexfemale     7.2243     4.0798   1.771   0.0797
age           0.9412     0.1813   5.192 1.14e-06
=======================
```

*Setelah dilakukan penyesuaian untuk usia, maka perbedaan diantara jenis kelamin secara statistik tidak signifikan.*

## Bab 13

Semua kesimpulan berbasiskan logaritma yang didapatkan adalah saling independen dan pasti sama nilainya,

```
> log2money <- log2(money)
> summary(lm6 <- lm(log2money ~ age + age2))
============================
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.340996   1.124481   0.303 0.769437
age          0.416419   0.058602   7.106 0.000101
age2        -0.004211   0.000668  -6.304 0.000232
---
> coef(lm6)
 (Intercept)          age         age2
 0.340996352  0.416418830 -0.004211267
> coef(lm4)
 (Intercept)          age         age2
 0.102650130  0.125354559 -0.001267718
> coef(lm4) / coef(lm6)
(Intercept)         age        age2
    0.30103     0.30103     0.30103
```

Besarnya unit pada sumbu horizontal dalam model, lm4 adalah 30% berada di dalam `lm6`. Proporsinya adalah log 2 berbasis 10.

```
> log10(2)  # 0.30103
```

Atau proporsi antara dua logaritma:

```
> log(2)/log(10)
```

Pada perhitungan nilai harapan usia di mana mendapatkan uang dalam jumlah yang maksimal

```
> a1 <- coef(lm6)[3]
> b1 <- coef(lm6) [2]
> c1 <- coef(lm6) [1]
> x1 <- -b1/(2*a1); x1   # 49.44104
> y1 <- a1 * x1^2 + b1 * x1 + c1
> y1; 2^y1  # 1590.304
```

Uang dibawa adalah maksimum pada usia 49,4 dan perkiraan besarnya uang adalah 1.590,3 baht. Hasil ini sama dengan yang dari lm4, yang menggunakan basis logaritma 10.

## Bab 14

Command berikut berasal dari bab sebelumnya

```
> data(BP)
> use(BP)
> des()
> age.in.days <- as.Date("2001-03-12") - birthdate
> age <- as.numeric(age.in.days)/365.25
> saltadd1 <- saltadd
> levels(saltadd1) <- c("no", "yes", "missing")
> saltadd1[is.na(saltadd)] <- "missing"
```

Command di bawah ini spesifik untuk bab ini.

```
> glm1 <- glm(sbp ~ age * saltadd, family=gaussian)
> glm2 <- glm(sbp ~ age + saltadd, family=gaussian)
> glm3 <- glm(sbp ~ age, family=gaussian)
> glm1$aic
  [1] 781.1646

> glm2$aic
  [1] 780.535

> glm3$aic
  [1] 990.425
```

Dari ketiga model, `glm2` bernilai AIC terendah. . Itu sebabnya model ini adalah model yang terbaik.

```
> summary(glm2)
===============
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)  63.1291    15.7645   4.005 0.000142 ***
age           1.5526     0.3118   4.979 3.81e-06 ***
saltaddyes   22.9094     6.9340   3.304 0.001448 **
---
    Null deviance: 109757  on 79  degrees of freedom
Residual deviance:  73192  on 77  degrees of freedom
AIC: 780.53
```

## Bab 15

Soal 1

```
> use(complete.data)
> eclair.beefcurry <- eclair.eat + (beefcurry=="Yes")
> tab1(eclair.beefcurry)
> eclair.beefcurry <- factor(eclair.beefcurry)
> levels(eclair.beefcurry) <- c("none","either","both")
> pack()
> glm1 <- glm(case ~ eclair.beefcurry, binomial, data=.data)
> logistic.display(glm1)

Logistic regression predicting case

                        adj. OR(95%CI)      P(Wald's test) P(LR-test)
eclair.beefcurry: ref.="none":                             < 0.001
   either               0.79 (0.29,2.19)    0.651
   both                 11.88 (4.65,30.36)  < 0.001

Log-likelihood = -534.7787
No. of observations = 972
AIC value = 1075.5574
```

Model ini berhubungan dengan dua `terminology` yang berhubungan yaitu eclair dan beef curry. Terminologi terakhir memuat jawabannya.

Soal 2

```
> zap()
> data(ANCtable); use(ANCtable)
> death <- factor(death, labels=c("no","yes"))
> anc <- factor(anc, labels=c("old","new"))
> clinic <- factor(clinic, labels=c("A","B"))
> data1 <- data.frame(death, anc, clinic, Freq)
> xtable <- xtabs(Freq~death+anc+clinic)
> mhor(mhtable=xtable)
Stratified analysis by  clinic
```

```
                OR lower lim. upper lim. P value
clinic A     0.801      0.346       1.90   0.556
clinic B     1.008      0.238       3.22   1.000
M-H combined 0.863      0.454       1.64   0.649

M-H Chi2(1) = 0.21 , P value = 0.649
Homogeneity test, chi-squared 1 d.f. = 0.11 , P value = 0.742
```

Setelah distratifikasi dengan clinic, disimpulkan tidak ada perbedaan dalam mortality antara dua metode pelayanan antenatal.

Soal 3

```
> zap()
> data(Hakimi)
> use(Hakimi)
> treatment <- 2 - treatment
> table(treatment)
> label.var(treatment, "Treatment")
> cc(dead, treatment)

       treatment
dead      0   1 Total
  0     196 204   400
  1      28  37    65
  Total 224 241   465

OR =  1.269
95% CI = 0.725 2.242
Chi-squared = 0.786 ,  1 d.f. , P value = 0.375
Fisher's exact test (2-sided) P value = 0.423

> mhor(dead, treatment, malpres, graph=TRUE)
Stratified analysis by  malpres
                OR lower lim. upper lim. P value
malpres 0    0.672      0.335       1.32  0.2655
malpres 1    6.688      0.940      81.48  0.0386
M-H combined 0.911      0.514       1.62  0.7453

M-H Chi2(1) = 0.105 , P value = 0.745
Homogeneity test, chi-squared 1 d.f.=5.596, P value=0.018
```

Crude dan adjusted odds ratio hasilnya berbeda.namun uji homogenitas member hasil

signifikan, hal ini menjelaskan bahwa strata specific odds ratios tidak dapat digabungkan. Pada saat 'malpres'=1, efek perlakuan terhadap kematian adalah signifikan.

```
> summary(glm(dead ~ treatment, binomial) -> cox1)
> summary(glm(dead ~ treatment + malpres, binomial) -> cox2)
> summary(glm(dead ~ treatment*malpres, binomial) -> cox3)
> summary(glm(dead ~ treatment*malpres+birthwt*treatment,
  binomial) -> cox4)
> step(cox4)
```

Kita menyimpulkan bahwa ada bukti terjadinya interaksi yang signifikan antara 'treatment' dan 'malpres'. Berat lahir cukup signifikan. Model terbaik adalah:

```
> m <- glm(dead ~ treatment*malpres+birthwt, family=binomial)
> logistic.display(m, decimal=1)

Logistic regression predicting dead

           crude OR(95%CI)     adj. OR(95%CI)      P(Wald)  P(LR-test)
treatment: 1.3 (0.7,2.2)       0.6 (0.3,1.1)       0.12     0.12
  1 vs 0

malpres:   13.2 (6.2,27.7)     1.6 (0.3,8.6)       0.6      0.61
  1 vs 0

birthwt    0.9985(0.998,0.999) 0.9986(0.998,0.999) < 0.001  < 0.001

treatment:malpres  -           14.4 (2,103.3)      0.01     0

Log-likelihood =  -154.5335
No. of observations =  465
AIC value = 319.07
```

Soal 4

```
> data(Ectopic)
> use(Ectopic)
> case <- outc == "EP"
> case <- factor(case)
> levels(case) <- c("control", "case")
> gravi1 <- unclass(gravi)
> m1 <- glm(case ~ hia + gravi1, family=binomial)
> logistic.display(m1, dec=1, crude=FALSE)

Logistic regression predicting case : case vs control
```

```
                        adj. OR(95%CI)    P(Wald test) P(LR-test)
hia: ever IA vs never IA 3.7 (2.5,5.4)     < 0.001      < 0.001

gravi1 (cont. var.)      1.0 (0.78,1.28)   1            1

Log-likelihood = -429.386
No. of observations = 723
AIC value = 864.773
```

*Tidak ada bukti hubungan dosis-respons linier antara graviditas dan risiko kehamilan ektopik, setelah disesuaikan untuk 'hia'.*

## Bab 16

Soal 1

```
> zap()
> library(survival)
> use(VC1to6)
> matchTab(case, alcohol, strata = matset)
===============
Odds ratio by Mantel-Haenszel method = 5.386

Odds ratio by maximum likelihood estimate (MLE) method = 5.655
 95%CI= 1.811 , 17.659

> clogit.display(clogit(case ~ alcohol + strata(matset)))
Call:
coxph(formula = Surv(rep(1, 119L), case) ~ alcohol +
  strata(matset), method = "exact")

  n= 119
        coef exp(coef) se(coef)    z      p
alcohol 1.73      5.66    0.581 2.98 0.0029

        exp(coef) exp(-coef) lower .95 upper .95
alcohol      5.66      0.177      1.81      17.7

Rsquare= 0.089   (max possible= 0.471 )
Likelihood ratio test= 11.1  on 1 df,   p=0.000843
Wald test            = 8.9  on 1 df,   p=0.00286
```

```
Score (logrank) test = 10.7  on 1 df,   p=0.00105
```

Soal 2

```
> clogit3 <- clogit(case ~ smoking + alcohol + rubber +
  strata(matset))
> clogit2 <- clogit(case ~ alcohol + rubber + strata(matset))
> clogit1 <- clogit(case ~ alcohol + strata(matset))
> clogit3$loglik
> clogit2$loglik
> clogit1$loglik
> clogit3
===============
Likelihood ratio test=12 on 3 df, p=0.00738 n=119
> clogit2
===============
Likelihood ratio test=11.5 on 2 df, p=0.00314 n=119
> clogit1
===============
Likelihood ratio test=11.1 on 1 df, p=0.000843 n=119
```

log likelihood bersyarat dan uji likelihood ratio dari 'clogit1', meskipun yang terkecil di antara ketiganya, memiliki derajat kebebasan terendah . Model ini hanya terdiri dari 'alkohol', yang sangat signifikan secara statistik sedangkan semua variabel independen dua lainnya tidak. Semua fakta ini menunjukkan bahwa 'clogit1' harus menjadi model pilihan.

Kita dapat mengkonfirmasi hal ini dengan menggunakan uji rasio kemungkinan:

```
> lrtest(clogit3, clogit2)
Likelihood ratio test for Cox regression & conditional logistic
  regression
Chi-squared 1 d.f. =  0.4743344 , P value =  0.491
```

Memiliki satu kelebihan derajat kebebasan dengan sedikit peningkatan di likelihood tidak banyak berguna. Oleh karena itu, 'clogit2' harus lebih baik dari 'clogit3'. Variabel bebas 'smoking' sekarang dihapus.

Dengan cara yang sama, kita sekarang menguji apakah perlu mempertahankan variabel 'rubber'.

```
> lrtest(clogit2, clogit1)
Likelihood ratio test for Cox regression & conditional logistic
  regression
Chi-squared 1 d.f. =  0.383735 , P value =  0.5356
```

Ternyata, model 'clogit2' dan 'clogit1' tidak signifikan secara statistic. Pilihan saat ini harus 'clogit1'. Minum minuman keras adalah satu-satunya prediktor yang signifikan untuk kanker esofagus.

## Bab 17

*Set up* data:

```
> zap()
> outcome <- gl(n=3, k=4)
> levels(outcome) <- c("nochange","immuned","dead")
> vac <- gl(n=2, k=2, length= 12)
> levels(vac) <- c("placebo","vaccine")
> agegr <- gl(n=2, k=1, length=12)
> levels(agegr) <- c("young","old")
> total <- c(25,15,4,8,1,0,25,35,3,1,2,1)
> .data <- data.frame(outcome, vac, agegr, total)
> .data
```

Soal 1

```
> table1 <- xtabs(total ~ agegr+vac, data=.data)
> table1
> cc(cctable=table1) # OR = 2.552, P value = .023
```

Soal 2

```
> table2 <- xtabs(total~agegr+outcome, data=.data)
> table2
> fisher.test(table2) # p-value = 0.226
```

Soal 3

```
> table3 <- xtabs(total ~ outcome + vac, data=.data)
> table3
> fisher.test(table3) # p-value < 2.2e-16
> multi3 <- multinom(outcome ~ vac + agegr, weights=total,
  data=.data)
> s3 <- summary(multi3)
> mlogit.display(multi3) # AIC = 137.13
```

Recreate a model with age group removed.

```
> multi4 <- multinom(outcome ~ vac, weights=total, data=.data)
> s4 <- summary(multi4)
> mlogit.display(multi4) # AIC = 134.471
```

Model 'multi4' memiliki nilai AIC yang lebih rendah daripada 'multi3'. Itu sebabnya kelompok umur tidak tepat berada di dalam model. Dari command sebelumnya, disimpulkan bahwa vaksinasi meningkatkan kesempatan meningkatnya kekebalan dengan nilai odd ratio 200 yang cukup tinggi signifikansinya. Juga harus dicatat bahwa vaksin juga (tidak signifikan) meningkatkan kemungkinan kematian.

## Bab 18

```
> zap()
> library(nnet)
> library(MASS)
> male <- c(rep(0, times=6), rep(1, times=6))
> drug <- rep(c(0,1), times=6)
> pain <- rep(1:3, times=4)
> total <- c(3,5,15,10,5,7,8,5,10,10,10,2)
```

Untuk regresi logistik polytomous

```
> pain.cat <- factor(pain)
> levels(pain.cat) <- c("nill","mild","severe")
> pain.ord <- ordered(pain.cat)
> model.polytom <- multinom(pain.cat ~ drug + male,
  weights=total)
> summary(model.polytom)
> mlogit.display(model.polytom)
```

Menunjukkan dampak obat yang signifikan hanya pada sakit yang parah saja.. AIC = 191.623. Untuk regresi logistik ordinal:

```
> model.ord <- polr(pain.ord ~ drug + male, weights=total)
> summary(model.ord)
```

Nilai AIC = 189.037, yang lebih baik (lebih rendah) daripada model polytomous

```
> ordinal.or.display(model.ord)
```

*Kesimpulannya, kedua nya, baik obat maupun menjadi laki-laki memiliki penurunan yang signifikan pada rasa nyeri.*

## Bab 19

Model berikut dijalankan setelah *men set up* variabel dengan menggunakan perintah dalam teks.

```
> model.pois <- step(glm(respdeath ~ agegr + period + arsenic2 +
  start, offset=log(personyrs), family=poisson, data=.data))
> summary(model.pois)
> idr.display(model.pois)
```

Perhatikan bahwa menggunakan 'arsenic2' dalam model lebih baik daripada menggunakan 'arsenik' menunjukkan tidak ada bukti hubungan dosis-respons. Selain itu, pekerja yang mulai bekerja dari 1925 secara signifikan memiliki risiko jauh lebih rendah daripada mereka yang telah dimulai sebelumnya.> poisgof(model.pois) # p.value = 0.40591

Tidak ada bukti terjadinya *over disperse.*

```
> model.nb <- glm.nb(respdeath ~ agegr + arsenic2 + start +
  offset(log(personyrs)), data=.data)
> summary(model.nb)  # theta = 49, S.E. = 107
```

Regresi Poisson dapat digunakan menggantikan regresi binomial negative.. Perhatikan bahwa masalah konvergensi akan muncul jika semua variable dimasukkan ke dalam model binomial negative. Kemungkinan besar karena jumlah data yang relative kecil.

## Bab 20

Soal 1

```
> model.bang1 <- glmmPQL(user ~ urban+ age_mean+ living.children,
  random=~1 | district, binomial, data=.data)
> summary(model.bang1)
```

Untuk menghitung 95% selang kepercayaan dari odds ratio:

```
> exp(intervals(model.bang1)$fixed)
```

Perhatikan bahwa wanita urban memiliki dua kali kemungkinan menggunakan kontrasepsi dibandingkan dengan perempuan pedesaan. Peningkatan 1 tahun usia dikaitkan dengan penurunan sekitar 3 persen dari odd penggunaan.

Soal 2

Dari output terakhir, peningkatan jumlah anak yang hidup tidak memiliki hubungan dosis-respon linier dengan penggunaan. Kemungkinan hampir dua kali lipat jika perempuan memiliki dua anak dan hampir tiga kali lipat untuk wanita dengan tiga anak yang hidup. Namun, karena jumlah melebihi tiga, odd dari penggunaan tidak meningkat lagi lebih jauh.

Soal 3

```
> model.bang2 <- glmmPQL(user ~ urban + age_mean +
  living.children, random = ~ age_mean | district,
  family=binomial, data=.data)
> logLik(model.bang1) # -4244.312 (df=8)
> logLik(model.bang2) # -4243.606 (df=10)
> lrtest(model.bang1, model.bang2) # P value=0.4933
```

Memasukkan usia ke dalam efek random berakibat terjadinya redundan.

Soal 4

```
> model.bang3 <- glmmPQL(user ~ urban * age_mean +
  living.children, random=~1 | district, family=binomial,
  data=.data)
> summary(model.bang3) # P value for interaction = 0.3887

> lrtest(model.bang1, model.bang3)
# Error:  Likelihood gets worse with more variables. Test not
  executed
```

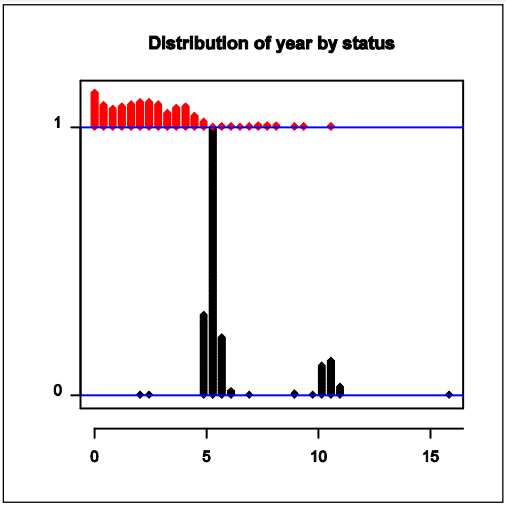
Bukti bahwa usia mempunyai efek yang berbeda di dalam areal perkotaan dan pedesaan tidak ditemukan.

# Bab 21

```
> zap()
> data(Compaq)
> use(Compaq)
> des()
> summ()
```

Soal 1

```
> summ(year, by = status)
> abline(v=c(5,6))
> dotplot(year, by=status)
```

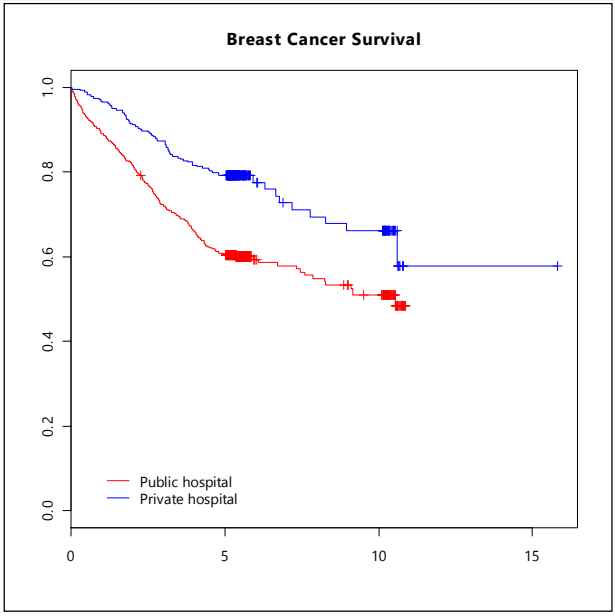



Catatan bahwa kematian berdistribusi dalam lima tahun pertama dimana saat itu hanya ada dua pengamatan censor. Dengan kata lain, ada cukup banyak censoring antara tahun ke 5 dan tahun ke 6 dimana hanya ada sedikit kasus kemstian dalam periode itu. Puncak censor yang kedua terjadi setelah tahun ke 10. Dimana hanya ada 1 pasien yang hidup selama 15.8 tahun dan censor pada saat masa studi berakhir.. Pengelompokan bergantian antara kematian dan censor tidak dapat dilakukan jika analisis tentang eksploratori data tidak dilakukan dengan hati-hati.

Soal 2

```
> surv.ca <- Surv(year, status)
```

```
> plot(survfit(surv.ca ~ hospital), col = c("red", "blue"),
  legend.text = levels(hospital), main="Breast Cancer Survival")
```




Perhatikan censoring sangat padat terjadi dengan tiba-tiba tahun ke 5 dan tahun ke 10.

Soal 3

```
> survdiff(surv.ca ~ hospital)
> survdiff(surv.ca ~ hospital + strata(stage))
> survdiff(surv.ca ~ hospital + strata(agegr))
> survdiff(surv.ca ~ hospital + strata(ses))
```

Perbedaan pertahanan hidup dari pasien dari kedua jenis rumah sakit sangat signifikan meskipun dilakukan beberapa penyesuaian. Perhatikan bahwa penyesuaian hanya dapat dilakukan satu variabel pada satu waktu menggunakan pendekatan ini. Penyesuaian multivariat dengan menggunakan regresi Cox disajikan dalam bab 22.

# Bab 22

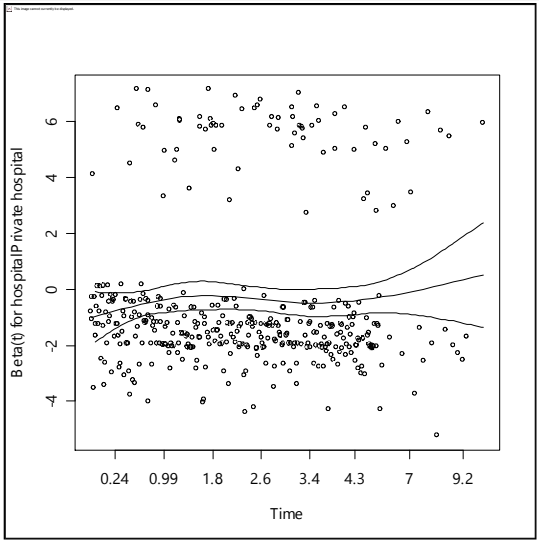
Soal 1

```
> coxph(surv.ca ~ hospital + stage + strata(ses) + agegr) ->
  model5
> cox.zph(model5)  # Global test p value = 0.00802
> coxph(surv.ca ~ hospital + stage + ses + strata(agegr)) ->
  model6
> cox.zph(model6)  # Global test p value = 0.00494
```

Model didasarkan pada stratifikasi dengan status sosio ekonomi dan dengan umur tetapi masih juga mengganggu asumsi proportional hazard.

Soal 2

```
> plot(cox.zph(model4), var = 1)
```




Ratio hazard terlihat relative stabil dan  and agak mengumpul di sisi negative untuk sebagian besar periode waktu. Fitur penting yang muncul dari grafik adalah bahwa ada dua kelompok residu. Beberapa nilai positif ekstrim ditemukan di bagian atas plot sebahagian besar pengamatan berada di dalam range 0 – 3 unit beta. Hal ini menunjukkan bahwa data boleh jadi berasal dari  lebih dari satu kelompok pasien. Sayangnya, kami tidak bisa menyelidiki lebih lanjut temuan ini.

## Bab 23

Soal 1

```
> help(expsy)
> summ(expsy)
```

Item 'it1' ke 'it10' semua berbagi nilai skala yang sama (1: low ke 4: high). Dataset yang tersedia biasanya cukup kecil untuk muncul di layar.

```
> expsy
```

Perhat.ikan data yang hilang

```
> use(expsy)
> alpha(it1:it10)  # 4 items reversed
> alphaBest(it1:it10)$remaining -> wanted
> tableStack(vars=wanted, reverse=TRUE) -> b
```

## Bab 24

Soal 1

An estimate of the population prevalence is not known. However, we can obtain a range of sample sizes required corresponding to a range of values for p, say from 0.1 to 0.9.

```
> p <- seq(0.1,0.9,0.1)
> d <- 0.05
> n.for.survey(p, delta = d)

Sample size for survey.
Assumptions:
  Confidence limit = 95 %
  Delta            = 0.05 from the estimate.

    p delta   n
1 0.1  0.05 138
2 0.2  0.05 246
3 0.3  0.05 323
4 0.4  0.05 369
5 0.5  0.05 384
6 0.6  0.05 369
7 0.7  0.05 323
8 0.8  0.05 246
```

```
9 0.9  0.05 138
```

Kita lihat dari output di atas bahwa ukuran sampel maksimum didapatkan jika p sama dengan 0.5. Hal ini benar untuk setiap survey dimana tidak ada informasi awal tentang dugaan prevalensi dan presisinya sudah tetap.Untuk situasi seperti ini pilihan yang paling aman adalah dengan memasang asumsi bahwa p = 0.5.

Soal 2

```
> p2 <- 0.5; or <- 2
> odds2 <- p2/(1-p2)
> odds1 <- or*odds2
> p1 <- odds1/(1+odds1); p1
 [1] 0.6666667
> n.for.2p(p1,p2)
Estimation of sample size for testing Ho: p1==p2
Assumptions:
     alpha = 0.05
     power = 0.8
        p1 = 0.6666667
        p2 = 0.5
     n2/n1 = 1

Estimated required sample size:

        n1 = 148
        n2 = 148
   n1 + n2 = 296
```

Sekitar 300 subjek dibutuhkan

Soal 3

Manfaat berharga adalah 2.5kg dan karena kita tidak tahu mean yang sebenarnya dalam tiga kelompok, kita dapat mengganti nilai-nilai untuk 'mu1' dan 'mu2', selama perbedaannya adalah 2,5. Juga, mengingat bahwa kita melakukan dua perbandingan, Tingkat kesalahan tipe 1 (alpha) akan menjadi 0,02, menggantikan nilai alpha yang konvensional yaitu 0,05 . Ukuran sampel yang diperlukan dapat diperoleh sebagai berikut:

```
> n.for.2means(mu1=10, mu2=12.5, sd1=3.5, sd2=3.5, ratio=2,
alpha=0.02)

Estimation of sample size for testing Ho: mu1==mu2
```

```
====== assumptions omitted ======
Estimated required sample size:
         n1 = 31
         n2 = 61
   n1 + n2 = 92
```

Dengan demikian 61 kontrol yang diperlukan, sedangkan 31 masing-masing diperlukan dalam dua kelompok perlakuan, memberikan ukuran sampel total yang dibutuhkan sebesar 123. Catatan, jika standar deviasi dalam setiap kelompok tersebut meningkat menjadi 4.5kg, ukuran sampel yang dibutuhkan meningkat menjadi 200.

# Indeks

## *T*



## *U*



## *V*



## *W*

# Fungsi-Fungsi yang Ada dalam Epicalc

| | |
|---|---|
| addMissingRecords | Menambahkan data yang hilang ke dalam kumpulan data longitudinal. |
| adjust | Menyesuaikan dan menstandarisasikan nilai rataan, proporsi dan rate. |
| aggregate.numeric | Menghitung ringkasan statistic untuk variable yang numeric. |
| aggregate.plot | Plot ringkasan statistic untuk variable yang numeric terhadap grupnya. |
| alpha, alphaBest | Cronbach's alpha |
| auc | Area dibawah kurva *time-concentration*. |
| be2ad | Mengubah tahun dari B.E. ke A.D. |
| cc | Menghitung dan membuat grafik Odds ratio |
| ci | Selang kepercayaan peluang, rataan, dan incidence |
| codebook | Codebook dari data frame |
| des | Deskripsi data frame atau variable |
| detachAllData | Melepaskan semua kerangka data (data frame) |
| dotplot | Dot plot |
| expand | Memperluas kerangka data agregat |
| fillin | *Rectangularize* kerangka data |
| followup.plot | Membuat plot followup Longitudinal |
| kap,… | Kappa statistics |
| keepData | Menyimpan bagian dari variable atau rekod |
| label.var | Manipulasi variable. |
| lagVar | Membuat vektor dari nilai lag atau nilai berikutnya . |
| logistic.display | Tabel untuk multivariate odds ratio, incidence density dll |
| lookup | Mengkodekan kembali beberapa nilai dari suatu variabel |
| lroc,… | Kurva ROC |

| | |
|---|---|
| lrtest | Uji perbandingan Likelihood |
| lsNoFunction | Lis *non-function objects* |
| markVisits | Menandai *visits of subjects* dalam format panjang |
| matchTab | Tabulasi *Matched* |
| mhor | Menghitung dan membuat grafik Odds ratio |
| n.for.2means,… | Menghitung ukuran sampel |
| pack | Manipulasi Variabel |
| poisgof | Ukuran kecocokan untuk permodelan data hitungan |
| power.for.2means | Menghitung Power untuk rataan dua sampel dan proporsi |
| power.for.2p | Menghitung Power untuk rataan dua sampel dan proporsi |
| pyramid | Pyramid populasi |
| recode | Recode variabel |
| rename | Mengganti nama variable dalam standar (default) kerangka data |
| setTitle | Mengatur bahasa dalam judul grafik Epicalc |
| shapiro.qqnorm | Normal Q-Q plots dengan Shapiro-Wilk's test |
| sortBy | Manipulasi variabel |
| summ | Meringkaskan dengan grafik |
| tab1 | Tabulasi satu arah |
| tableStack | Tabulasi variabel dalam bentuk *stack* |
| tabpct | Tabulasi dua arah |
| titleString | Menggantikan kata-kata yang umum digunakan dalam judul grafik Epicalc |
| unclassDataframe | Faktor-faktor yang tidak ada di dalam kelompok kerangka data standar |
| use | Perintah yang cepat untuk membaca data dan melampirkannya *( attach)* |
| zap | Menghapus objek dan dan melepaskan (detach) semua kerangka data |

# Dataset yang Ada di Epicalc

| | |
|---|---|
| ANCdata | Dataset tentang pengaruh metode perawatan baru antenatal pada tingkat kematian |
| ANCtable | Dataset tentang pengaruh metode perawatan baru antenatal pada tingkat kematian (berupa tabel) |
| Attitudes | Dataset dari survey tentang sikap diantara para staff rumah sakit |
| BP | Dataset tentang tekanan darah dan faktor-faktor penentu (*determinant)* |
| Bang | Dataset dari survey tingkat kesuburan di Bangladesh, 1988 |
| Compaq | Dataset tentang kelangsungan hidup penderita kanker |
| DHF99 | Dataset untuk latihan tentang infestasi predictor larva nyamuk |
| Decay | Dataset tentang kerusakan gigi dan mutan streptokokus |
| Ectopic | Dataset dari studi kasus-kontrol melihat sejarah aborsi sebagai faktor risiko kehamilan ektopik |
| Familydata | Dataset dari keluarga terhipotesis |
| HW93 | Dataset dari penelitian tentang prevalensi cacing tambang dan intensitasnya |
| Hakimi | Dataset tentang pengaruh pelatihan personil terhadap tingkat kematian |
| IudAdmit | Dataset tentang penerimaan kasus untuk uji IUD |
| Marryage | Dataset tentang tingkat usia perkawinan |
| Montana | Dataset paparan arsenic dan and kegagalan pernafasan |
| Oswego | Dataset dari wabah keracunan makanan di Amerika Serikat |
| Outbreak | Dataset dari wabah keracunan makanan karnaval hari olah raga , Thailand tahun 1990 |
| Planning | Dataset untuk latihan *cleaning, labelling* dan *recoding* |
| SO2 | Dataset tentang pencemaran udara dan kematian di Inggris |
| Sleep3 | Dataset tentang rasa mengantuk dalam suatu *workshop* |
| Suwit | Infeksi cacing tambang dan kehilangan darah: SEAJTM 1970 |
| Timing | Dataset pada waktu tidur, bangun tidur dan ketibaan di suatu |

| | |
|---|---|
| | *workshop* |
| VC1to1, | Datasets pada studi *matched case-control* kanker esofagus |
| Xerop | Dataset dari studi di Indonesian tentang kekurangan vitamin A dan resiko infeksi pernafasan |

# Tentang Epicalc

Open source dan perangkat lunak bebas telah menjadi andalan bagi para peneliti, terutama di negara-negara berkembang, di mana kebutuhan untuk perangkat lunak komputer dan biaya beberapa aplikasi perangkat lunak sering bertentangan. Meningkatnya kompleksitas proyek penelitian dan persyaratan analitis yang terkait menyebabkan pula perkembangan R di akhir tahun 1990-an. Versi R saat ini , sebuah perangkat lunak *open source* statistik awalnya ditulis oleh Robert Gentleman dan Ross Ihaka dari Departemen Statistik University of Auckland, adalah hasil dari upaya kolaboratif yang melibatkan kontribusi dari seluruh dunia. R menyediakan berbagai teknik statistik dan grafis, dan sangat lah meluas cakupannya. Program Khusus untuk Riset dan Pelatihan di Tropical Diseases (TDR) yang disponsori oleh UNICEF / UNDP / Bank Dunia / WHO telah mendukung penyusunan R add-on paket, Epicalc, yang memungkinkan R untuk lebih mudah menangani data epidemiologi. Epicalc, yang ditulis oleh Virasakdi Chongsuvivatwong dari Prince of Songkla University, Hat Yai, Thailand, telah diterima dengan baik oleh anggota tim inti-R dan paket dapat didownload dari CRAN (Komprehensif R Arsip Jaringan) <http://www.cran r-project.org.> yang dicerminkan oleh 69 lembaga akademik di 29 negara., Epicalc juga telah disambut baik oleh mahasiswa dan pengguna . Di satu sisi, hal tersebut membantu analis data dalam eksplorasi data dan manajemen. Di sisi lain, ia membantu epidemiolog muda untuk mempelajari istilah kunci dan konsep berdasarkan hasil numerik dan grafis analisis.

**Steven Wayling**
**Program Khusus untuk Riset dan Pelatihan di Tropical Diseases (TDR)**
**World Health Organization**
**Oktober, 2007**

**Zurnila Marli Kesuma** menyelesaikan Program S3 di bidang Epidemiologi (Biostatistika) PSU Thailand. Saat ini menjadi dosen senior di Prodi Statistika Unsyiah dan aktif mengembangkan laboratorium Biostatistika yang fokus pada penelitian bidang kesehatan dan komunitas. Beberapa hasil penelitiannya telah dipublikasikan pada Jurnal Internasional terkemuka dan prosiding nasional/internasional.

**Alih Bahasa :**
Zurnila Marli Kesuma

**Virasakdi Chongsuvivatwong** adalah Profesor dibidang *Community Medicine*. Beliau mendirikan Jurusan Epidemiologi di PSU Thailand tahun 1986 dan mengembangkannya menjadi Program Pascasarjana Internasional pada tahun 1992. Di tahun 2004, beliau pula yang mengembangkan Institute For Research and Development for Southern Health dan menjadi presiden Deep South Relief and Reconsiliation Foundation yang dibentuk pada tahun 2010. Hingga saat ini beliau masih aktif menjadi pembicara dan peneliti di berbagai event internasional.

**Pengarang:**
Virasakdi Chongsuvivatwong

**Editor:**
Edward McNeil
Edward.m@psu.ac.th

**Dipublikasikan dengan dukungan:**

Special Programme for Research & Training in Tropical Diseases (TDR) sponsored by UNICEF/UNDP/World Bank/WHO