

Pengenalan bahasa pemrograman R: untuk non-programmer

Dasapta Erwin Irawan dan Prana Ugi

Jan 19, 2016

Contents

1	BAB 1 PENDAHULUAN	2
1.1	Sekilas R	2
1.2	Sejarah R	2
1.3	Mengapa R	3
1.4	Mengenal Lingkungan R	4
1.5	Instalasi R	4
1.6	Menggunakan buku ini	6
1.7	References	6
2	BAB 2 DATA DALAM R	6
2.1	Data punya peran sentral	7
2.2	Tipe data yang dikenal R	9
2.3	Data disimpan sebagai <i>Objects</i> (obyek)	12
2.4	Mengimpor data ke dalam R	17
2.5	References	19
3	BAB 3 SYNTAX R	19
3.1	<i>Function</i> , <i>arguments</i> dan <i>packages</i>	19
3.2	Syntax perintah dalam R	22
3.3	References	22

1 BAB 1 PENDAHULUAN

1.1 Sekilas R

Beberapa tahun yang lalu, kalau anda melihat huruf “R”, maka anda akan memikirkan alfabet ke-18. Tapi saat ini “R” juga bisa berarti bahasa pemrograman yang trendi. R kini telah memiliki lebih dari 4000 paket tambahan (*add-on package*) atau biasa disebut *package* saja, masing-masing *package* tersebut umumnya memiliki *mailing list* dengan anggota ribuan, selain itu juga ada banyak sekali R group di media sosial seperti Google+, LinkedIn, dan Twitter. Situs [R-evolution](#) menampilkan jumlah *package* yang diunggah tiap tahun dan grafik para pengembang yang membuat lebih dari 10 *package*.

Saat ini juga rutin diadakan pertemuan pengguna R di berbagai negara yang disebut *R Meetup*. Saat ini telah diadakan 155 pertemuan dengan jumlah peserta 46.607, di 127 kota di 31 negara, menurut situs [R user group](#).

R yang awalnya diperuntukkan untuk analisis statistik, saat ini telah berkembang aplikasinya hingga dapat melakukan manipulasi data GIS serta menampilkannya. Ditambah lagi dengan era *data analysis* atau akrab disebut *big data*, maka perkembangan R menjadi tidak terbendung lagi.

Aplikasi R yang terus meluas didukung oleh berkembangnya bahasa pemrograman lainnya, seperti python, Ruby dll. Salah satu contohnya adalah R markdown. Bahasa *markup* (*markup language*) yang lebih mudah dari LaTeX atau html sekalipun. Dengan menggunakan R markdown, saat ini anda bisa membuat *blogpost* atau naskah buku dengan R, seperti halnya naskah yang sedang anda baca saat ini. Penjelasan lebih detail mengenai Markdown akan saya jelaskan dalam bagian yang lain.

1.2 Sejarah R

Menurut situs [R project](#) R adalah bahasa dan lingkungan untuk komputasi statistik dan grafis. R adalah proyek berjenis *open source* GNU, mirip dengan bahasa “S” yang dikembangkan oleh Bell Laboratories (dulu dikenal sebagai AT&T, sekarang bernama Lucent Technologies) oleh John Chambers dkk. Sekarang anda tahu kan bahwa R bukanlah satu-satunya bahasa pemrograman yang bernama unik.

Perintah dasar dalam bahasa R telah menyediakan berbagai *tool* untuk pemodelan statistik linear dan nonlinear, analisis *time-series*, klasifikasi, analisis klaster, dan analisis grafis. Kemampuan ini terus berkembang dengan adanya ribuan paket tambahan yang diunggah ke server CRAN tiap tahunnya.

Testimoni dari para pengembang dan pengguna R menyebutkan bahwa salah satu kelebihanannya adalah sifatnya yang dapat diulang (*reproducible*) dan mampu menghasilkan keluaran grafis berkualitas tinggi untuk keperluan publikasi ilmiah. Jadi mengapa anda tidak belajar mulai sekarang. Anda tidak akan sendiri karena kesulitan anda pasti ada jawabnya.

1.3 Mengapa R

Mengapa orang menggunakan R? Beberapa alasannya adalah sebagai berikut.

1.3.1 R gratis, *open source*, dan *cross platform*

Alasan utama pasti karena gratis. Tidak hanya gratis tapi juga *open source*. Apa artinya? Kita tidak hanya bisa membuat paket tambahan saja, tetapi juga bisa memodifikasi kode dasar R. Tentunya ini diperuntukkan bagi yang memahami dasar pemrograman atau yang memang profesinya sebagai pengembang (*developer*). Namun demikian, dengan banyaknya paket tambahan R, maka mestinya sudah banyak pula tutorial untuk membuatnya. Bahkan saat ini, saya memperhatikan, kebanyakan mahasiswa master atau PhD di luar negeri yang menggunakan statistik dan R dalam risetnya, membuat *package* sendiri sesuai kebutuhan risetnya. R juga tersedia untuk tiga sistem operasi mayor atau istilah kerennya *cross platform*, yaitu: Linux (Ubuntu, Fedora), Mac OS, dan Windows.

1.3.2 R mendukung prinsip *reproducibility*

R adalah aplikasi berbasis *command line*, artinya setiap perintah harus diketik sebagai baris perintah. Tidak seperti SPSS, Statistica, atau Minitab, yang berbasis *point and click* serta *drag and drop*. Karena menggunakan baris perintah, maka anda dapat berbagi hasil analisis dengan mudah, cukup dengan mengirimkan baris kode perintah dan datasetnya. Dengan menjalankan ulang baris perintah terhadap dataset tersebut, maka orang lain, misal rekan kerja anda, dapat mengulang proses dan analisis yang anda kerjakan secara persis sama. Prinsip ini disebut sebagai *reproducibility*. Anda tidak perlu mengirimkan *screenshot* layar dan grafik berformat jpeg hanya untuk menjelaskan proses yang anda kerjakan dan hasilnya. Rekan kerja anda pun cukup meng-copy paste kode perintah ke dalam layar R Studio atau R base-nya. Apakah R Studio dan R base itu. Akan dijelaskan tidak lama lagi.

1.3.3 R menghasilkan visualisasi yang berkualitas tinggi

R juga mendukung para peneliti untuk membuat grafik yang berkualitas baik, atau biasa disebut *publication quality images*. Ini karena redaksi jurnal ilmiah biasanya mensyaratkan resolusi dan format grafik khusus, yang tidak dapat dibuat dengan menggunakan piranti lunak *spreadsheet* biasa seperti Ms Excel atau LibreOffice Sheet. Fasilitas ini tentu saja sangat membantu kita yang berprofesi sebagai dosen dan peneliti. Visualisasi yang dapat dibuat dengan R, tapi tidak dengan Excel, misalnya:

- menyandingkan dua atau lebih grafik dengan format sumbu yang sama. Cara yang biasa dilakukan adalah meng-copy beberapa grafik dan menempelkannya secara manual. Akibatnya format sumbu, kisaran nilai maks dan min dapat saja berbeda.
- menampilkan identitas data pada grafik. Excel hanya dapat memunculkan nilai sumbu x atau y, tapi tidak dapat menampilkan identitas data.

1.4 Mengenal Lingkungan R

R bukan hanya sebuah bahasa pemrograman. Ia adalah lingkungan pengembangan terintegrasi (*integrated development environment* atau sering disingkat IDE). Beberapa komponen penyusun R dapat dijelaskan secara singkat sebagai berikut:

- **R base:** ini adalah inti dari instalasi R. Tanpa menginstalasi komponen ini, R tidak akan berjalan dalam sistem anda. R base sudah memiliki perangkat (*tools*) yang lengkap untuk keperluan analisis statistik umum. Jadi tanpa package tambahan, anda sudah dapat mulai menggunakannya. Daftar *package* dan fungsi dasar R dapat dilihat pada situs [R base packages](#).
- **R IDE:** adalah piranti lunak (aplikasi) terintegrasi. Ada dua jenis yang bisa anda pilih: apakah [R Studio](#) atau [R Commander](#). Masing-masing memiliki kelebihan dan kelemahan. Tapi dalam buku ini, kita akan memakai R Studio. R Studio dapat membantu anda menulis skrip kode, menjalankannya, mengevaluasinya, dan membuat visualisasinya.
- **Package tambahan:** ini adalah pengembangan dari kode dasar R base. Tiap *package* memiliki kemampuan spesifik, tergantung kepada si pembuatnya. Terdapat ribuan *package* di server CRAN saat ini. Akan sangat membingungkan. Untuk itu tiap *package* yang diunggah di CRAN, wajib memiliki dokumentasi atau *manual book*. Dokumentasi tersebut memiliki spesifikasi minimum agar seragam, yang memudahkan para pengguna, yaitu kita para non-programmers.

1.5 Instalasi R

Sekarang mulailah “pekerjaan kotor kita”, yaitu menginstalasi R ke dalam PC atau laptop kita. Untuk itu coba perhatikan beberapa hal berikut ini:

- Spesifikasi komputer/laptop: Tidak ada spesifikasi khusus untuk R, tetapi prinsip utamanya adalah makin besar data yang anda gunakan, makin kompleks analisis yang anda lakukan, akan memerlukan spesifikasi prosesor dan RAM yang makin besar. Jadi ini akan sangat bergantung kepada kebutuhan anda. Untuk keperluan pembelajaran gunakan saja komputer yang anda miliki sekarang.
- Sistem operasi (OS): Seperti yang telah saya sampaikan sebelumnya, R berjalan di semua OS: Linux (bisa Ubuntu, Fedora dll), Mac OS, dan tentunya Windows. Jangan khawatir, yang manapun OS yang anda pakai, spesifikasi R nya akan sama persis.
- Apa saja yang perlu anda unduh dan install:
 - R base: Inti dari R.
 - * Kunjungi [Situs R Project](#)

- * Pilih *mirror server*. Pilih server yang ada di Indonesia. Klik **CRAN mirror** di dalam kotak “Getting Started”. Cari server di Indonesia. Ada dua, silahkan anda pilih:
 - [Server BPPT](#)
 - [Server Universitas Jember](#)
 - * Setelah server CRAN-BPPT terbuka, klik versi R sesuai dengan OS yang anda miliki.
- R Studio: lingkungan pemrograman.
 - Kunjungi [Situs R Studio](#)
 - Klik menu **Products** > **RStudio** > klik tombol **Download RStudio Desktop**. Secara otomatis R Studio akan membaca OS yang anda pakai dan proses pengunduhan akan segera dimulai.

Atau

- Anda bisa langsung buka halaman <http://www.rstudio.com/products/rstudio/download/>, Pilih versi RStudio. Pilihan *installer* yang ada per tanggal 04 September 2014 adalah (sekarang pasti sudah berubah banyak):
 - RStudio 0.98.1049 - Windows XP/Vista/7/8 ukuran file 48.2 MB tanggal update 2014-09-02
 - RStudio 0.98.1049 - Mac OS X 10.6+ (64-bit) ukuran file 37.8 MB tanggal update 2014-09-02
 - RStudio 0.98.1049 - Debian 6+/Ubuntu 10.04+ (32-bit) ukuran file 56.3 MB tanggal update 2014-09-02
 - RStudio 0.98.1049 - Debian 6+/Ubuntu 10.04+ (64-bit) ukuran file 58 MB tanggal update 2014-09-02
 - RStudio 0.98.1049 - Fedora 13+/openSUSE 11.4+ (32-bit) ukuran file 56.6 MB tanggal update 2014-09-02
 - RStudio 0.98.1049 - Fedora 13+/openSUSE 11.4+ (64-bit) ukuran file 57.9 MB tanggal update 2014-09-02

Setelah proses pengunduhan selesai, jalankan file program instalasinya:

- Untuk Linux: jalankan file **xRstudioxx.deb** dan ikuti perintahnya.
- Untuk Mac OSX: jalankan file **xRstudioxx.dmg** dan ikuti perintahnya
- Untuk Windows: jalankan file **xRstudioxx.exe** dan ikuti perintahnya

1.6 Menggunakan buku ini

Buku ini merupakan buku tutorial R pertama dalam Bahasa Indonesia. Kami berdua bukanlah programmer. Saya, penulis pertama, berlatar belakang pendidikan geologi dan Prana Ugi, penulis kedua, menyelesaikan pendidikan sarjana dan magisternya di bidang matematika dan statistik. Jadi kami merasakan sulitnya belajar bahasa pemrograman, sehingga kami menggunakan cara yang paling sederhana untuk menjelaskan berbagai fungsi dan operasi R dalam buku ini. Pada banyak tempat kami melakukan pengulangan dan alur cerita yang lambat, yang mungkin akan membosankan bagi para programmer.

Untuk menggunakan buku ini mudah dan sederhana. Kalau anda pemula, maka mulailah dari awal. Tapi kalau anda telah memiliki pengalaman dengan R sebelumnya, dapat kiranya memilih berbagai contoh aplikasi yang kami berikan. Anda cukup menyediakan:

- PC atau laptop dengan spesifikasi yang memadai? Saya pernah menginstalasi R di sebuah netbook dengan prosesor yang minim dan ternyata tetap jalan dengan baik. Tentunya data yang anda baca tidak dapat terlalu banyak bila menggunakan laptop jenis ini.
- Instalasi R base dan R Studio.
- Koneksi internet, karena anda akan sering mencoba mencari contoh-contoh lainnya milik orang lain yang telah diunggah di [Rpubs](#) misalnya, atau anda mencari jawaban atas masalah anda dari [Stackoverflow](#).
- Anda tidak perlu mengetik ulang kode yang ada dalam buku ini, karena kami menyediakan kodenya secara online di situs [DasaptaErwin](#). Copy kode yang ada di situs tersebut sesuai dengan bab dan sub bab yang anda perlukan, kemudian paste ke dalam *script window* R Studio.

Mudah bukan.

1.7 References

- [Server CRAN](#)
- [R Studio](#)
- [Quick R Tutorial](#)
- [Dasapta Erwin Irawan blog](#)
- [R tutor](#)

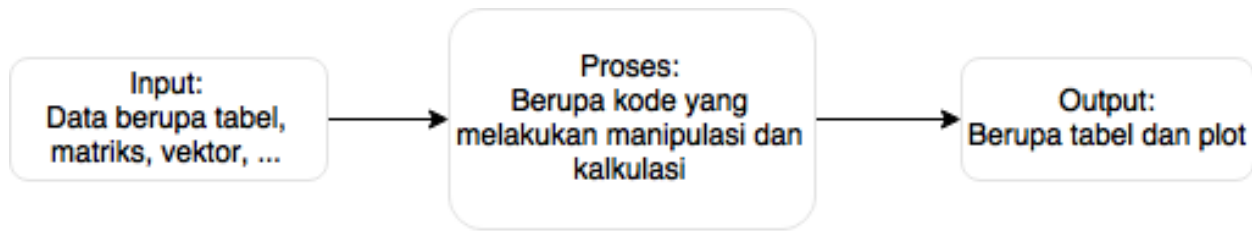
2 BAB 2 DATA DALAM R

```
... By the way, the word "data" is plural, so ordinarily you would not say "data is" or
... [King WB] (http://ww2.coastal.edu/kingw/statistics/R-tutorials/objects.html)
```

2.1 Data punya peran sentral

Data adalah salah satu bagian sentral dari R, bagian yang lainnya adalah kode dan output. Oleh karenanya kami menempatkan bagaimana menyiapkan dan memanipulasi data di bab kedua. Dalam diagram berikut ini dapat anda lihat bahwa mekanisme kerja dalam R adalah:

- input: berupa data
- proses: berupa kumpulan kode perintah
- output: berupa tabel dan plot

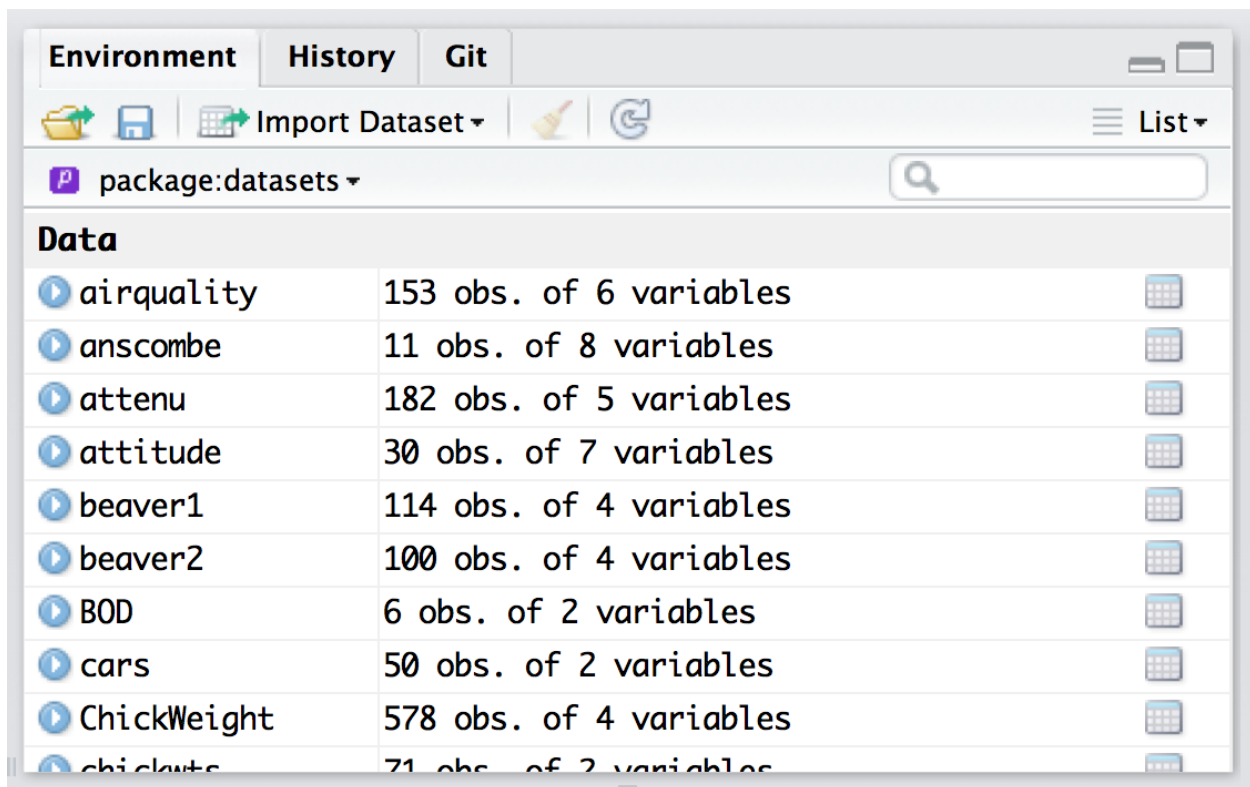
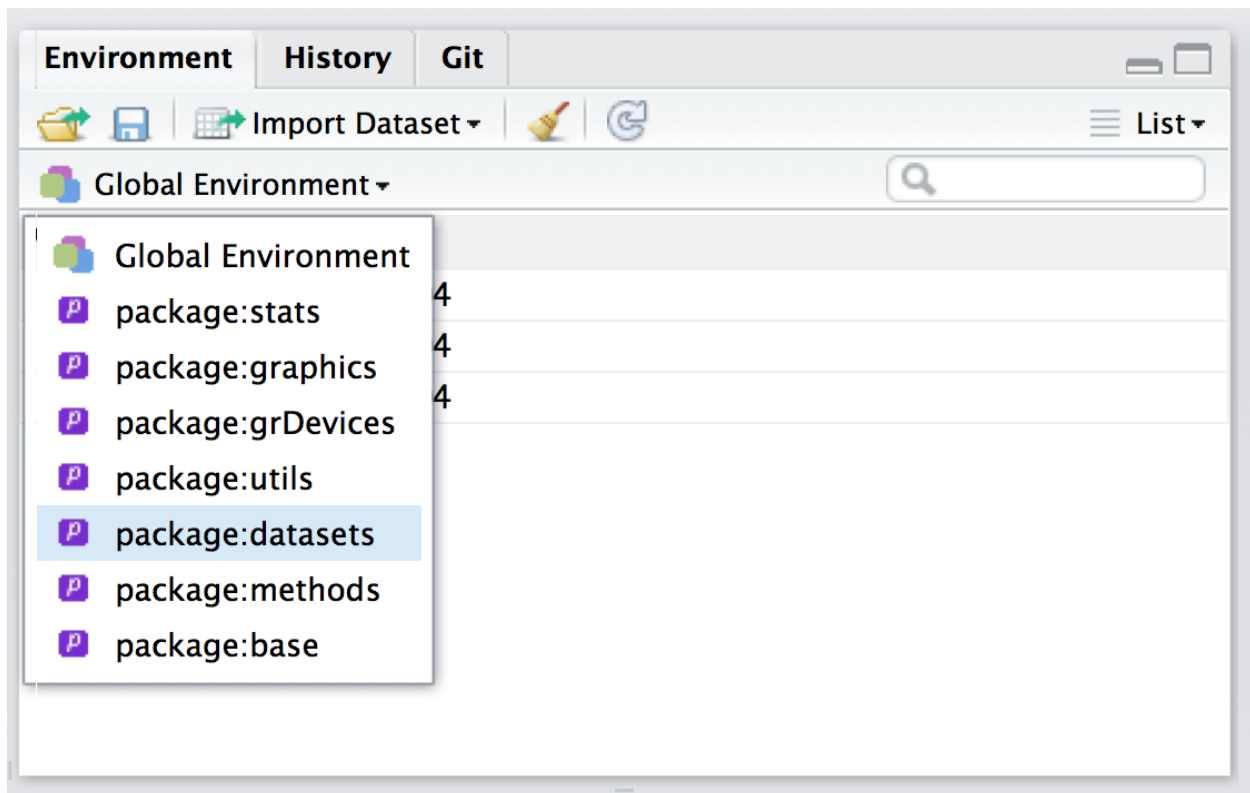


Data adalah bahan baku R. Ia berperan sebagai input. Kalau anda mahasiswa pasti sering berhadapan dengan *spreadsheet* maka itulah salah satu contoh data, dalam hal ini adalah data eksternal. Anda harus mengimpor data tersebut ke dalam format yang dapat dibaca oleh R. Pada bagian berikutnya akan kami perlihatkan cara untuk melakukan langkah tersebut.

R memiliki beberapa data yang terinstalasi secara *default* di dalamnya. Ini dibuat untuk keperluan latihan biasanya, beberapa data yang sering dipakai sebagai contoh misalnya:

- **iris**: data iris hasil pengukuran (dalam cm) geometri 50 spesies bunga Iris (*Iris setosa*, *Iris versicolor*, dan *Iris virginica*).
- **mtcars**: data pengukuran 10 spesifikasi 32 merek mobil yang diambil dari majalah Motor Trend US terbitan tahun 1974.
- **rivers**: data panjang 141 sungai utama di Amerika Utara (dalam mil) yang dikompilasi oleh USGS.

Coba anda lihat jendela **Global Environments** kanan atas. Klik menu *dropdown* (seperti pada gambar berikut), maka anda akan melihat banyak dataset yang terinstalasi dalam sistem R. Masing-masing dataset tersebut bersifat bebas pakai dan dapat digunakan dalam latihan R anda.



Berikut ini adalah bagaimana cara melihat isi dataset. *Copy paste* baris per baris perintah berikut ini setelah R> di jendela console (kiri bawah) atau *copy paste* seluruhnya di jendela Script (kiri atas)


```

# for "iris" data
iris          # to see the complete data. It's messy.
?iris         # to see the name and the source of the data.
head(iris)    # to see the first six rows of the data.
tail(iris)    # to see the last six rows of the data.
str(iris)     # to see the structure and type of the data, eg (num = numeric, chr = character)
class(iris)   # to see what kind of object the data is.

# for "mtcars" data
mtcars        # to see the complete data. It's messy.
?mtcars       # to see the name and the source of the data.
head(mtcars)  # to see the first six rows of the data.
tail(mtcars)  # to see the last six rows of the data.
str(mtcars)   # to see the structure and type of the data, eg (num = numeric, chr = character)
class(mtcars) # to see what kind of object the data is.

# for "rivers" data
rivers        # to see the complete data. It's messy.
?rivers       # to see the name and the source of the data.
head(rivers)  # to see the first six rows of the data.
tail(rivers)  # to see the last six rows of the data.
str(rivers)   # to see the structure and type of the data, eg (num = numeric, chr = character)
class(rivers) # to see what kind of object the data is.

```

2.2 Tipe data yang dikenal R

Ada beberapa tipe data yang bisa diproses oleh R:

- *Numeric*
- *Integer*
- *Complex*
- *Logical*
- *Character*

2.2.1 *Numeric*

Tipe data numerik adalah semua angka dari 0 sampai 9. Bisa positif, bisa negatif. Bisa mengandung pecahan (*fraction*) atau tidak. Yang penting adalah tidak bercampur dengan karakter.

```

x <- 1704      # assign a value to object named x
x              # print the value of x
class(x)       # print the class name of x

```

```

y <- 17.04      # assign a value to object named y
y              # print the value of y
class(y)       # print the class name of y

```

2.2.2 *Integer*

Jenis data integer sama dengan numerik, tapi tidak mengandung pecahan, atau angka yang bulat. Nilai integer boleh positif dan negatif.

Untuk memastikan bahwa nilai yang kita *assign* untuk suatu obyek adalah integer, kita bisa menggunakan perintah `as.integer()`. Jadi kita bisa juga memerintahkan R untuk menyimpan angka non pecahan sebagai integer.

```

x <- 1704
class(x)

y <- -17.04
class(y)

z1 <- as.integer(17.04)
z1
class(z1)

z2 <- as.integer(1704)
z2
class(z2)

```

Bisa dilihat bedanya? Apa yang anda lihat?

2.2.3 *Complex*

Bilangan kompleks adalah bilang yang memiliki struktur: $a + b_i$. Jadi mengandung bilangan a yang riil (*real*) dan bilangan b_i yang imajiner.

```

c <- 1 + 2i      # create a complex number
c              # print the value of z
class(c)       # print the class name of z

```

2.2.4 *Logical*

Nilai logikal dalam R biasanya bernilai `TRUE` atau `FALSE`.

```

x <- 1704
y <- 1511
z <- x > y    # is x larger than y? and store in z
z            # print the logical value in z
class(z)     # print the class name of z

```

Berikut adalah beberapa simbol untuk operasi logika baku dalam R:

- “&” (dan/*and*),
- “|” (atau/*or*),
- “!” (negasi/*negation*).

```

d <- TRUE
e <- FALSE
d & e      # u AND v
d | e      # u OR v
!d         # negation of u

```

2.2.5 *Character*

Data berjenis karakter adalah yang terdiri dari huruf (A-Z). Tipe data ini tidak akan diproses oleh R sebagai angka.

```

e <- "Ali"
e
class(e)

```

```

c <- "Adi"
c
class(c)

```

Tapi kita juga bisa mengetik angka sebagai teks atau memerintahkan R untuk membaca angka sebagai teks.

```

g <- "3.14"
g
class(g)

```

```

h <- as.character(3.14)
h
class(h)

```

2.3 Data disimpan sebagai *Objects* (obyek)

Data baik yang kita ketik secara langsung maupun yang diimpor dari luar akan disimpan ke dalam R sebagai obyek.

2.3.1 Penamaan obyek

Setiap obyek harus memiliki nama, seperti halnya yang sudah anda lakukan di beberapa baris di atas. Obyek sangat sensitif, begitu anda mengganti nilai yang tersimpan dalam suatu obyek, maka R akan langsung menindih obyek tersebut. Atau anda menggunakan nama obyek yang sama dengan nilai yang berbeda, maka obyek yang terakhirlah yang akan disimpan oleh R.

Anda bisa memberi nama obyek sesuka anda. Aturan dasarnya adalah tidak boleh mengandung **spasi** atau simbol **dash**. Jadi anda bisa membuat nama obyek yang:

- panjang, tapi jangan terlalu panjang.
- campuran antara huruf kecil dan besar.
- mengandung simbol *underscore* (`_`) dan titik (`.`).

Berikut ini adalah beberapa contohnya:

- satu huruf seperti: `x`, `X`. Untuk kode yang pendek boleh saja, tapi kalau kode anda sudah panjang, akan susah mengingatnya.
- gabungan dua karakter seperti: `x2`, `X2`. Masih susah mengingatnya.
- gabungan tiga karakter seperti: `x.2`, `x_2`.
- sebuah frasa seperti: `myData`, `MyData`, `my_data`, `my.data`.
- tapi jangan terlalu panjang seperti: `my.data.from.the.geology.experiment`
- atau mengandung nama seseorang, seperti: `flinstone`, `Flinstone`, `FLINSTONE`, atau `Fred.Flinstone`
- atau jangan pula menggunakan nama perintah baku dalam R, misal: `mean`, `sd`, `plot`, dll.

2.3.2 Mencoba membuat obyek

Selain mengimpor data dari luar, anda juga dapat memasukkan data secara langsung dari prompt R (`R>`). Ada tiga cara untuk meng-assign data kepada suatu obyek dalam R. Coba ketik beberapa baris berikut (selalu akhiri dengan **enter**). Simbol `#` digunakan untuk memberikan komentar. Semua yang diketik di belakang simbol `#` tidak akan diproses oleh R. Kita coba langsung ya agar anda terbiasa dengan *look and feel* syntax dalam R.

Ketikkan setiap baris di bawah ini setelah `R>` dan akhir dengan **enter**. Anda akan membuat obyek pertama anda bernama `x` berisi angka 1704.

```
# cara pertama (dengan beberapa variasi)
x<-1704
x
```

```
x <- 1704
x
```

```
x    <-    1704
x
```

```
# atau cara kedua
```

```
y=1704
y
```

```
y = 1704
y
```

```
z    =    1704
z
```

```
# atau cara ketiga
```

```
1704->z
z
```

```
1704 -> z
z
```

```
1704    <-    z
z
```

Bagaimana hasilnya?

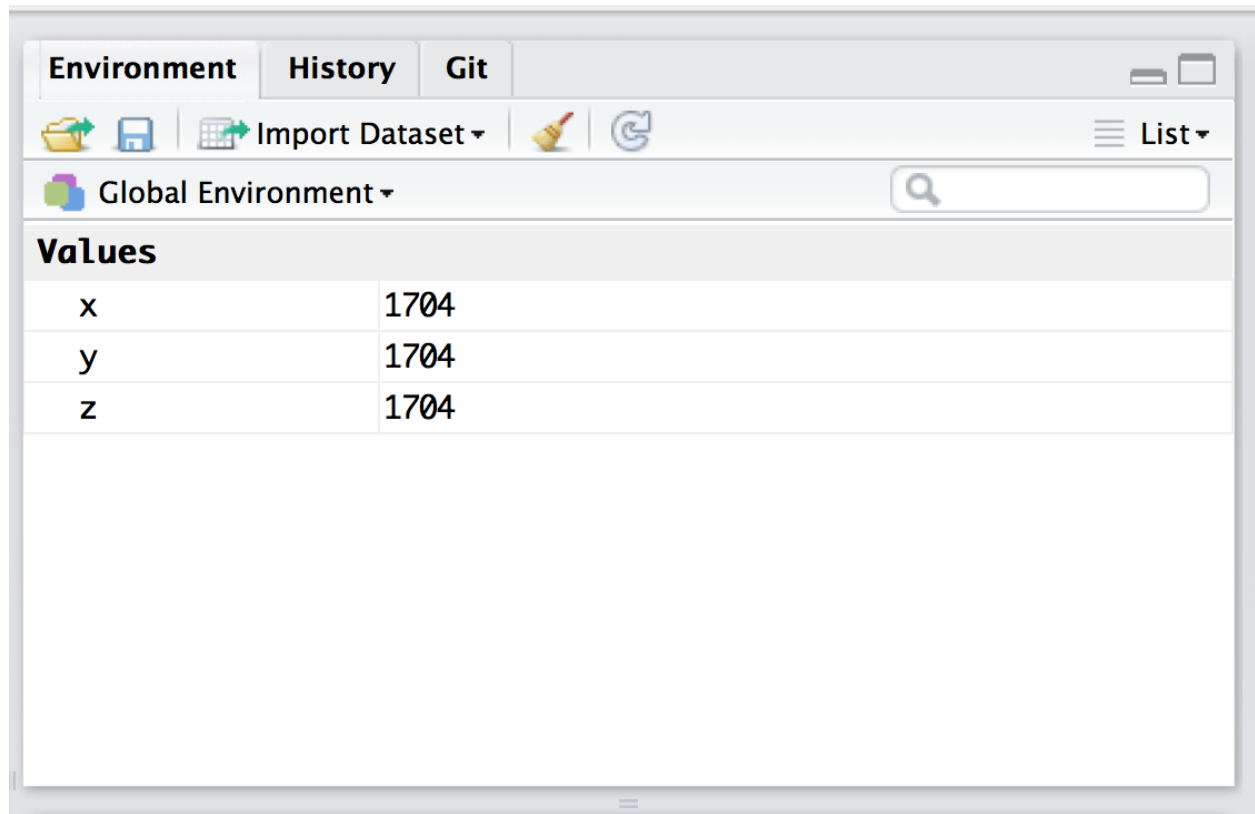
Sama bukan.

Jadi dalam hal ini spasi tidak mempengaruhi R untuk mengerti perintah anda. Namun hal ini tidak berlaku umum. Nilai 1704 dimuat oleh R ke dalam memorinya dalam bentuk obyek bernama `x` untuk diproses lebih lanjut.

Simbol `<-` (tanpa spasi diantaranya) dan `=` mempunyai efek sama, tetapi tidak selalu. Namun simbol `=` akan sering digunakan dalam *argument*. Jadi agar tidak bingung biasakan menggunakan simbol `<-`.

Sekarang kita coba lihat ada berapa obyek anda punya di memori R. Dengan perintah berikut, anda akan lihat bahwa di memori R sudah ada tiga obyek bernama: `x`, `y`, `z`. Coba lihat

jendela `Environment` di kanan atas, atau ketik `ls()`.



2.3.3 Jenis obyek

Semua data yang bisa dibaca oleh R, dinamakan obyek/*Objects*. Berikut ini adalah beberapa jenis obyek dalam R:

- *vector*
- *list*
- *array*
- *matrix*
- *data frames*

2.3.3.1 Vektor

Karena bidang saya sumber daya air, maka coba kita lihat data “rivers”. Menurut perintah `class(rivers)` data ini berjenis vector. Beda dengan “iris” dan “mtcars” yang berjenis data frame. Vektor adalah jenis obyek yang paling sederhana. Sekarang kita coba membuat vektor sederhana.

```
c("Abi", "Aci", "Adi", "Afi", "Agi") # vector 1
c(76, 78, 79, 80, 83, 87) # vector 2
c(38, 34, 37, 36, 31, 27) # vector 3
```

Di atas anda sudah membuat tiga buah vektor, tapi keduanya belum memiliki nama. Ingat aturannya, setiap obyek, termasuk vektor, harus bernama. Sekarang kita coba beri nama.

```
nama <- c("Abi", "Aci", "Adi", "Afi", "Agi", "Ali") # vector 1
tahun.lahir <- c(76, 78, 79, 80, 83, 87) # vektor 2 named tahun.lahir
usia <- c(38, 36, 35, 34, 31, 27) # vektor 3 named usia
```

```
nama
str(nama)
class(nama)
```

```
tahun.lahir
str(tahun.lahir)
class(tahun.lahir)
```

```
usia
str(usia)
class(usia)
```

Setiap vektor akan terdiri dari nilai vektor dan no index atau no urut. Kalau anda ingin melihat data ketiga dalam vektor “nama”, “tahun.lahir”, dan “usia”, maka gunakan perintah berikut.

```
nama[3]
tahun.lahir[3]
usia[3]
```

Hasilnya adalah “Adi”, 79, dan 35.

Contoh lain, kalau anda ingin melihat data tertentu dalam vektor “rivers”, maka gunakan perintah berikut.

```
rivers[115]          # to see data no 115
rivers[10:20]        # to see data no 10 hingga 20
rivers[c(10, 15, 18, 20)] # to see data no 10, 15, 18 dan 20
```

Kalau anda melihat tulisan NA, maka artinya ada kolom yang kosong dalam data anda.

2.3.3.2 *lists*

List adalah gabungan dari campuran berbagai obyek yang tidak sebangun. Misal kita ingin membuat list bernama “daftar” dengan menggabungkan vektor “nama”, “tahun.lahir”, dan “usia” dengan angka 5.

```
daftar <- list("nama", "tanggal.lahir", "usia", 5)
daftar
str(daftar)
class(daftar)
```

2.3.3.3 *arrays*

Sebuah array bisa tersusun oleh matriks dan tabel. Kita bisa membuat sebuah array dengan tiga baris, empat kolom, dan dua tabel.

```
# make an array of number from 1 to 24
# of 2 tables, each table has 3 rows, 4 columns.
array1 <- array(1:24, dim=c(3,4,2))
array1
```

2.3.3.4 *matriks/matrices*

Sebuah matriks adalah kumpulan elemen data yang disusun secara dua dimensi, terdiri dari m baris dan n kolom.

Kita coba buat ya dengan menggunakan perintah:

```
namamatriks <- matrix(c(dataelement), nrow = m, ncol = n)
```

```
matriksA <- matrix(c(2, 4, 3, 1, 5, 7), nrow=2, ncol=3)
```

```
# or you can edit the code this way
```

```
matriksA <- matrix(c(2, 4, 3, 1, 5, 7),
                  nrow=2,
                  ncol=3)
```

```
matriksA
```

```
matriksA2 <- matrix(c(2, 4, 3, 1, 5, 7),
                   nrow=2,
                   ncol=3,
                   byrow=TRUE)
```

```
matriksA2
```

```
# here's the second matrix
```

```
matriksB <- matrix(c(2, 4, 3, 1, 5, 7),
                  nrow=3,
                  ncol=2)
```

```
matriksB
```

```
matriksB2 <- matrix(c(2, 4, 3, 1, 5, 7),
                   nrow=3,
                   ncol=2,
                   byrow=T)
```

```
matriksB2
```

perintah `byrow = TRUE` akan membuat matriks diisi berdasarkan baris (row).

2.3.3.5 *data frames*

Data frame adalah kumpulan dari vektor dengan dimensi yang sama. Bedakan dengan *list* ya. Jadi kita bisa membuat sebuah data frame dari awal, membaca sebuah tabel dan menyimpannya dalam sebuah data frame.

Ingat data frame “iris” atau “mtcars”.

```
class(iris)
str(iris)
class(mtcars)
str(mtcars)
```

Masih ingat tiga vektor kita: “nama”, “tahun.lahir”, dan “usia”? Karena ketiganya terdiri dari jumlah baris yang sama, yakni enam baris, maka kita bisa membuat data frame darinya. Ingat aturannya adalah vektor harus sama ukurannya, terutama jumlah barisnya agar bisa digabungkan dalam satu data frame.

```
str(nama)
str(tahun.lahir)
str(usia)
df <- data.frame(nama, tahun.lahir, usia)
df
```

Anda akan melihat bahwa data frame “iris” terdiri dari lima variabel (dalam kolom) dan 150 observasi (dalam baris). Data frame “mtcars” terdiri dari 11 variabel dan 32 observasi. Tipe obyek data frame akan sering kita pakai dalam R.

2.4 Mengimpor data ke dalam R

Tiap piranti lunak (P/L) statistik biasanya akan membuat format data khusus, misal: format **sav** untuk *SPSS*, format **dta** untuk *Stata*, atau format **sta** untuk *Statistica*. Semuanya adalah format biner (*binary*). Jadi kalau tidak punya P/L nya maka anda tidak akan bisa membuka filenya.

R agak aneh sedikit. Ia bisa membuka file data dengan format yang umum, yakni text file atau ASCII, misalnya: **csv** (*comma separated value*) atau **txt**. Ini yang membuat R bisa dipakai di OS apa saja. Dalam buku ini kita akan banyak menggunakan format **csv**. Tapi dengan sedikit *tweaking* dan menginstalasi package bernama **foreign** atau **readr** anda akan bisa membuka format apa saja, termasuk **xls** (dan variannya), **sav**, dan **dta**.

Sekarang kita coba untuk mengimpor data dari beberapa jenis format yang umum ke dalam R. Untuk melakukan hal ini kita punya fungsi dasar **read.tabel**. Fungsi ini sangat handal untuk mengimpor data berformat text (ASCII) seperti txt dan csv. Kita coba selangkah demi selangkah.

Dalam latihan ini file data perlu kita unduh lebih dahulu. Data yang diberikan adalah data kualitas air yang diambil di beberapa mata air di kawasan Bandung Selatan. Tabel tersebut disimpan di situs [blog saya](#) yang merupakan pendamping buku ini.

Ada dua cara:

1. anda bisa mengunduh lebih dahulu empat jenis file yang disertakan di halaman situs tersebut atau,
2. anda bisa secara langsung membacanya dari R.

```
# Membaca file format *.csv secara langsung dari R
# dan disimpan dalam obyek bernama "data_csv"
data_csv <- read.csv("http://dasaptaerwin.net/wp/wp-content/uploads/2016/01/data_csv.csv")

# anda dapat melihat jendela "Environment"
# anda akan melihat ada 7 observasi (baris) dan 33 variabel (kolom)
# Coba baca obyek dengan beberapa perintah berikut
View(data_csv)
head(data_csv)
tail(data_csv)

# membuka file format *.txt
data_txt <- read.table("http://dasaptaerwin.net/wp/wp-content/uploads/2016/01/data_txt.txt")
# lihat di jendela "Environment" ada 9 obs dan 33 variable.

# coba perintah berikut
data_txt <- read.table("http://dasaptaerwin.net/wp/wp-content/uploads/2016/01/data_txt.txt")
# lihat di jendela "Environment" ada 8 obs dan 33 variable.
# apa yang tidak sama? Jumlah barisnya. Apa yang beda?
View(data_txt)
# anda bisa melihat nama "Situ Kince" yang mestinya berada dalam satu kolom dipenggal me

# membuka file zip
unzip("http://dasaptaerwin.net/wp/wp-content/uploads/2016/01/data_zip.zip", unzip="inter")
# ada pesan kesalahan dan unzip tidak berhasil.
# oleh karena itu kami sarankan anda menyimpan file dalam format *.csv atau *.xls

# Membuka file format *.xls menggunakan package "readr"
install.packages("readxl") # instalasi package
library(readxl) # loading library
# https://github.com/hadley/readxl

data_xls <- read_excel("http://dasaptaerwin.net/wp/wp-content/uploads/2016/01/data_xls.xls")

# didapatkan pesan kesalahan
```

```
# harus diunduh lebih dulu baru dibuka.
```

```
data_xls <- read_excel("/Users/dasaptaerwinirawan/Documents/2016/Github/Rforbeginners/ta  
View(data_xls)
```

```
# Anda dapat melihat ada 7 obs dan 33 variables
```

Dari latihan di atas, kami sarankan:

- simpanlah file data dalam format `*.csv` atau `*.xls`, manfaatkan package seperti `readxl`, `foreign`, atau `gdata` untuk membaca berbagai format file data lainnya.
- unduhlah terlebih dahulu file data sebelum diimpor ke dalam R.

2.5 References

- [Chongsuvivatwong, V, 2012, Analysis of Epidemiological Data using R and Epicalc Package, Epidemiology Unit Prince of Songkla University THAILAND](#)
- [Chi Yau, R Tutorial Blog, accessed 17 April 2014](#)
- [King, WB, R Tutorial, Coastal Carolina University](#)
- [Wikipedia online encyclopedia](#)
- [Readxl R-package](#)
- [Foreign package tutorial](#)
- [Data import tutorial](#)

3 BAB 3 SYNTAX R

3.1 *Function, arguments* dan *packages*

Sekarang kita coba mengenal fungsi (function) dan package. Fungsi adalah kumpulan perintah yang sifatnya iteratif dan dapat diulang. Sedangkan argumen (*arguments*) adalah spesifikasi rinci dari fungsi sesuai dengan kebutuhan kita. Dari bab sebelumnya anda sebenarnya telah mengenal fungsi.

Bila anda mengetik `head(nama dataframe)`, maka `head` adalah fungsi. Dalam bab-bab berikutnya anda akan berlatih menggunakan banyak fungsi lainnya.

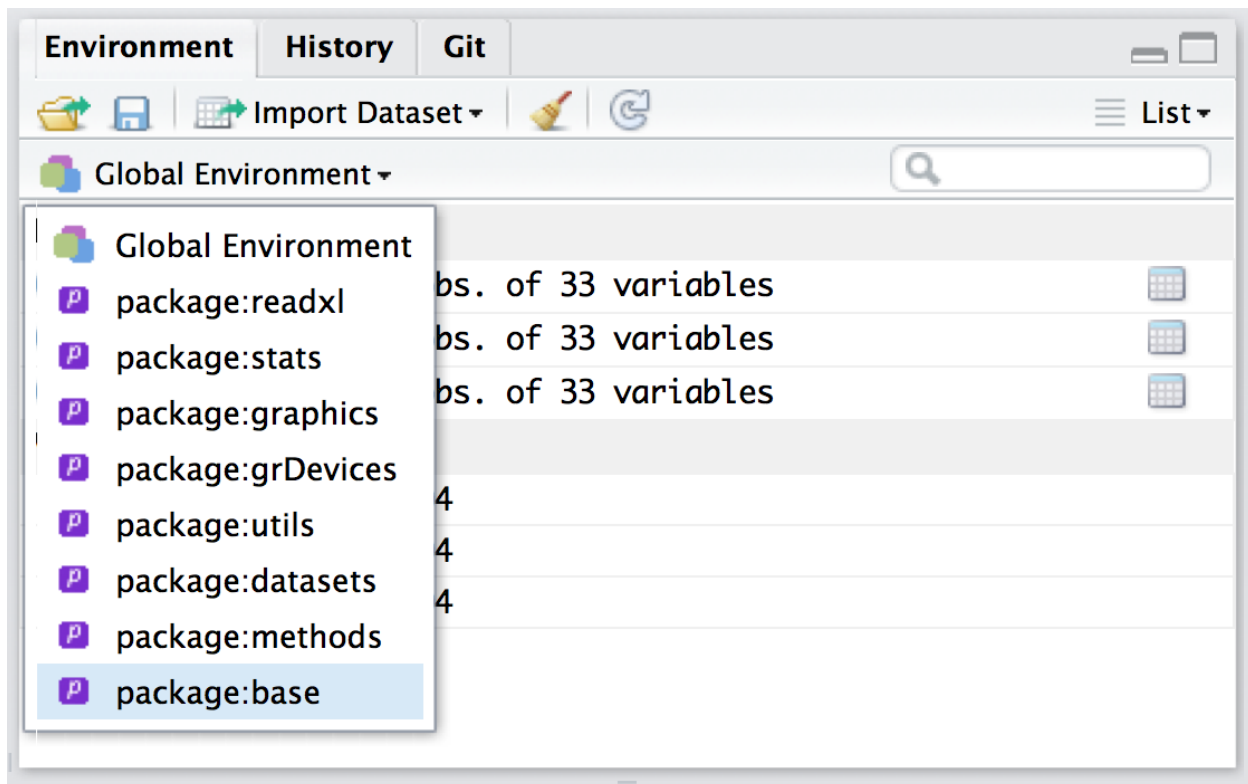
Kemudian bila anda memerintahkan R untuk membuka suatu file (misal bernama `data.csv`), maka perintahnya adalah `read.csv(data.csv, header=TRUE)`. Dalam proses tersebut, maka anda menggunakan fungsi `read.csv` dan argumen `header=TRUE` untuk meminta R membaca judul kolom (*header*).

Sudah jelas ya antara fungsi dan argumen? Bila masih bingung, jangan khawatir, anda akan makin paham sejalan dengan anda terus berlatih. Bilamana anda bingung bagaimana cara menggunakan fungsi tertentu, R menyediakan file bantuan (*help file*) yang dapat muncul di jendela **Viewer** (kanan bawah) bila anda mengetik `?nama fungsi`, misal: `?read.csv`.

Anda dapat membaca file bantuan tersebut yang formatnya telah dibakukan, yaitu diawali dengan pendahuluan, syntax (cara penulis), daftar argumen, dan contoh.

Selanjutnya adalah **package**. Ia adalah sebuah kemasan yang berisi kumpulan atau kompilasi dari berbagai fungsi sejenis atau data. Karena R adalah aplikasi *open source*, maka pengembangannya berbasis komunitas. Package yang semula dikembangkan oleh individu pengguna R suatu saat akan dimasukkan sebagai package baku (*Base package*) dalam instalasi R. Per hari ini (18 Januari 2016), server R yang bernama [CRAN](#) telah menyimpan sebanyak 7791 package dalam berbagai jenis dan kategori.

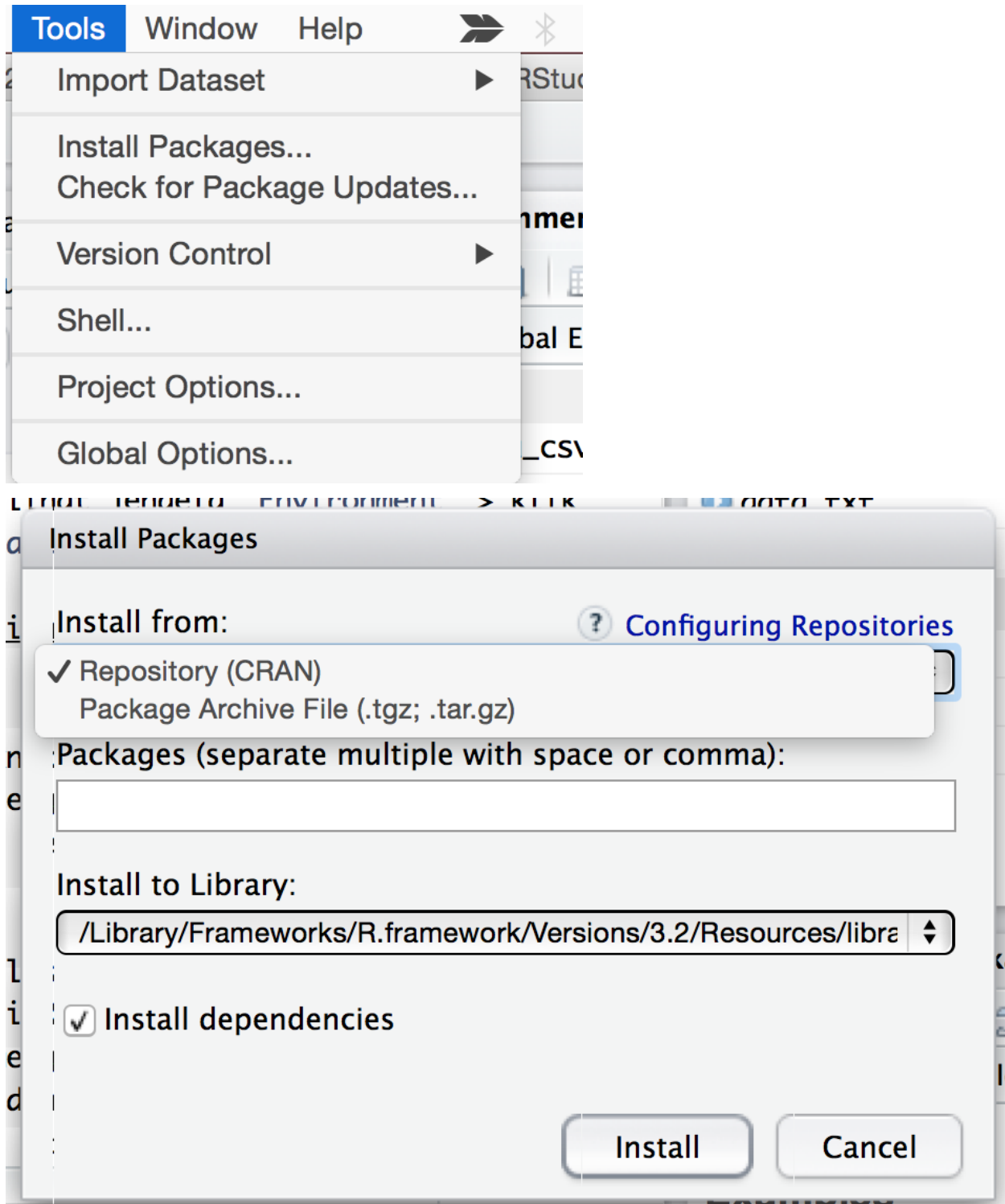
Untuk melihat package baku, anda dapat melihat jendela **Environment** > klik **Global Environment** > pilih **packages: base**, atau ketik `library()`.



Untuk menginstalasi package tambahan, teknik yang paling umum adalah menginstalasi langsung dari server CRAN dengan cara: `install.packages("nama package")`. R akan secara otomatis memilih repositori CRAN dan menginstalasi package yang dimaksud. Untuk mengetahui berbagai package yang sesuai dengan kebutuhan analisis data, anda dapat mengakses situs [CRAN task views](#). Dari situs tersebut dapat beberapa kategori package, misal: Bayesian, Multivariate, dan Time series

Karena satu dan lain hal, terkadang instalasi dengan cara di atas tidak dapat dilakukan, biasanya karena perbedaan versi R. Untuk itu anda dapat mencoba menginstalasi: + dari zip file. Anda perlu mengunduh zip file package-nya terlebih dahulu sebelum menginstalasinya dengan menggunakan perintah `install.packages.zip(/dir/nama package)` atau gunakan menu **Tools** > **install packages** > pilih **install from package archives**, seperti pada gambar berikut. + dari situs Github. Para pengembangan package umumnya akan menyimpan kode pemro-

gramannya di repositori [Github](#). Namun anda perlu menginstalasi package `devtools` terlebih dahulu. Pertama untuk instalasi package `devtools` ketik `install.packages("devtools")`, `library(devtools)`, kemudian instalasi dari Github dapat dilakukan dengan mengetik `install_github("username/packageName")`.



3.2 Syntax perintah dalam R

Seperti halnya bahasa pemrograman lainnya, anda harus mengetahui bagaimana cara memerintahkan R melalui fungsi dan argumen. Cara penulisannya (syntax) adalah: `obyek <- fungsi(argumen1, argumen2, ...)`. Coba anda ingat lagi bagaimana cara: + mengimpor file data (`read.csv(xxxx)`) + melihat isi data (`View(xxxx)`)

maka formatnya akan sama.

Dalam hal argumen yang banyak dan panjang, anda dibolehkan memenggalnya seperti contoh berikut ini.

```
obyek <- fungsi(argumen 1,                argumen 2,                argumen
3)
```

Dengan cara ini maka kita dapat dengan mudah melihat bagian argumen yang salah.

3.3 References

- [R Studio packages](#)
- [CRAN task views](#)
- [Tutorial Git and Github](#)