Serba-serbi R

Kodingan R

@asepandrif

April 26, 2019

## 1. Pendahuluan

Pada pelatihan kali ini, kita akan menggunakan RStudio. Namun perlu diingat bahwa syntax yang digunakan di sini dapat pula digunakan pada RGui. Perlu dicatat bahwa core R yang digunakan adalah versi 3.5.3. Mungkin saja ada beberapa perintah yang berbeda dengan versi R lain.

Pada tulisan ini, bagian yang ada dalam kotak berwarna gelap merupakan syntax atau rangkaian perintah R sedangkan yang berada dalam kotak yang berarwa terang dan isinya dimulai dengan ## merupakan luaran (*output*) syntax yang dieksekusi dan ditampilkan. Cara mengeksekusi syntax di R dapat dilakukan dengan menekan tombol enter pada jendala console, atau ctrl+R, atau ctrl+enter. Penjelasan lebih rinci dapat dilihat pada bagian 1.3.1.Gambar icon Run yaitu:

Gambar 1. Run

**Gambar 1. Run**

### 1.1. Kutipan R

Cara mengutip R sudah disediakan saat pertama kali membuka R atau RStudio. Namun bila lupa mencatatnya, kita tidak perlu menutup dan membuka aplikasi R. Cukup tuliskan kode:

citation()

##   
## To cite R in publications use:  
##   
## R Core Team (2019). R: A language and environment for  
## statistical computing. R Foundation for Statistical Computing,  
## Vienna, Austria. URL https://www.R-project.org/.  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Manual{,  
## title = {R: A Language and Environment for Statistical Computing},  
## author = {{R Core Team}},  
## organization = {R Foundation for Statistical Computing},  
## address = {Vienna, Austria},  
## year = {2019},  
## url = {https://www.R-project.org/},  
## }  
##   
## We have invested a lot of time and effort in creating R, please  
## cite it when using it for data analysis. See also  
## 'citation("pkgname")' for citing R packages.

Jika menggunakan pustaka tambahan yang belum tersedia dalam inti R, misalnya pustaka **cars**, pustaka **mlr**, dan pustaka-pustaka lain, maka anda dapat melihat cara mengutipnya dengan menggunakan syntax: **citation(“nama pustaka”)**. Sebagai contoh, jika menggunakan pustaka **stats** maka kita dapat mencari cara mengutipnya dengan:

citation("stats")

##   
## The 'stats' package is part of R. To cite R in publications use:  
##   
## R Core Team (2019). R: A language and environment for  
## statistical computing. R Foundation for Statistical Computing,  
## Vienna, Austria. URL https://www.R-project.org/.  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Manual{,  
## title = {R: A Language and Environment for Statistical Computing},  
## author = {{R Core Team}},  
## organization = {R Foundation for Statistical Computing},  
## address = {Vienna, Austria},  
## year = {2019},  
## url = {https://www.R-project.org/},  
## }  
##   
## We have invested a lot of time and effort in creating R, please  
## cite it when using it for data analysis. See also  
## 'citation("pkgname")' for citing R packages.

### 1.2. *Working Directory* (wd) dan Libpath

*Working directory* merupakan “wadah” tempat file-file yang akan kita gunakan dan akan kita simpan. Secara *default*, R mengatur *working directory* sesuai dengan pengaturan saat proses instalasi. Namun, kita dapat mengatur ulang *working directory* saat bekerja menggunakan R.

Sedangkan **libpath** merupakan “wadah” tempat pustaka (*packages*) disimpan. Pustaka-pustaka tersebut merupakan program yang digunakan di dalam R. Secara *default*, sudah ada beberapa pustaka saat menginstall R seperti **stats** dan **graphics**. Namun kita dapat menginstall pustaka lain dari dari berbagai sumber seperti CRAN dan Github.

Kita biasanya mengubah-ubah *working directory* tapi tidak dengan libpath. Beberapa syntax yang berhubungan dengan *working directory* dan libpath adalah sebagai berikut.

# Melihat working directory yang aktif  
getwd()

## [1] "D:/"

*Working direktory* kita berada pada folder Documents yang berlokasi lengkap di disk C:/Users/aa/Documents. Untuk mengubah *working directory* dapat menggunakan syntax berikut ini.

# Mengatur working directory  
setwd("D:/Pelatihan R")

Sekarang, *working directory* kita sudah berubah tempat. Untuk mengeceknya gunakan lagi syntax getwd().

# Melihat working directory  
getwd()

## [1] "D:/"

Pada contoh di atas, tidak terdapat perubahan *working directory* karena komputer yang penulis gunakan mengatur *default working directory* pada tempat yang sama dengan saat dilakukan perubahan. Perubahan letak *working directory* akan sangat tergantung pada komputer yang digunakan.

Pustaka R yang terpasang dalam komputer kita dapat dipanggil secara khusus dengan menggunakan fungsi library(). Namun, sebelum mencoba memanggil pustaka, kita bisa tahu letak pustaka R dapat menjalankan syntax sebagai berikut.

# Mengecek libpath  
.libPaths()

## [1] "C:/Users/aa/Documents/R/win-library/3.5"  
## [2] "C:/Program Files/R/R-3.5.3/library"

Dalam komputer yang digunakan penulis, pustaka R terletak pada tempat seperti tertulis di atas. Pada komputer yang anda gunakan, sangat mungkin letaknya berbeda.

### 1.3. Hal-hal yang Perlu Diperhatikan

#### 1.3.1. Menulis dan Mengeksekusi Perintah R

Perintah R ditulis perbaris. Satu baris ditandai dengan sebuah baris baru atau dengan tanda titik koma (;). Saat menggunakan RStudio atau RGui, penulisan syntax dapat dilakukan pada jendela **console** atau menulisnya di dalam jendela **script**. Jika menulis dalam jendela console, kita cukup menekan enter saat sudah selesai menulis satu baris atau mengakhirnya dengan titik koma. Sedangkan jika menggunakan jendela Script, kita dapat menekan tanda Run atau menekan tombol ctrl+enter (RStudio) atau menekan ctrl+R (RGui). Selain itu, kita juga dapat mengklik icon Run. Gambar icon Run dapat dilihat pada Gambar 1.

#### 1.3.2. R Merupakan Bahasa yang *Case Sensitive*

Dalam R, antara a dan A dibaca sebagai objek yang berbeda. Aturan seperti itu disebut dengan *case sensitve*.

#### 1.3.3. Perhatikan Penamaan Objek

Nama objek tidak dapat diawali dengan: 1. angka 2. underscore 3. spasi 4. simbol-simbol seperti @, $, \* dan lain sebagainya

Sangat dianjurkan untuk tidak menggunakan istilah (huruf atau kata) yang sudah dipesan oleh R (*reserved words*). Penggunaan istilah yang sudah dipesan dapat menimbulkan kerancuan dan besar kemungkinan program tidak akan berjalan sesuai dengan keinginan kita. Beberapa contoh istilah yang sudah dipesan adalah: 1. c 2. data 3. dt 4. df 5. function 6. mean

Pengecekkan nama objek dapat dilakukan dengan cara mengkesekusi nama objek. Contohnya kita akan mengecek apakah c dan mean adalah istilah yang sudah dipesan oleh R.

c

## function (...) .Primitive("c")

mean

## function (x, ...)   
## UseMethod("mean")  
## <bytecode: 0x000000000696e018>  
## <environment: namespace:base>

Dari hasil diatas kita tahu bahwa c dan data sudah dipesan. Adapun istilah yang belum dipesan akan menampilkan error saat dieksekusi.

r

## Error in eval(expr, envir, enclos): object 'r' not found

Ketik kita mengeksekusi objek r, muncul pemberitahuan error yang menginformasikan bahwa objek r tidak ada.

#### 1.3.4. Fitur Panduan (*Help*)

Dalam R, kita dapat melihat bantuan dengan menuliskan syntax **help(“nama\_fungsi”)** atau **?nama\_fungsi**. Berikut ini contoh saat akan melihat bantuan untuk fungsi mean.

help(mean)

## starting httpd help server ... done

?mean

Dalam RStudio kita, akan muncul petunjuk pada jendela Help yang secara *default* terletak di sebelah kiri bawah.

## 2. Menjalankan R

### 2.1. Menampilkan objek

Secara umum, R menampilkan luaran di jendela console. Perintah print() menghasilkan luaran sesuai dengan objek yang digunakan. Akan tetapi, untuk menghemat waktu kita dapat menampikan objek tanpa menuliskan fungsi print() secara formal melainkan cukup dengan menulis nama objek lalu kita eksekusi seperti contoh berikut ini.

print("hello, world!")

## [1] "hello, world!"

"hello, world!"

## [1] "hello, world!"

1

## [1] 1

### 2.2. R sebagai Kalkulator

Kaerna sifatnya yang interpreter, maka R dapat digunakan seperti kalkulator. Beberapa contoh operator dan fungsi yang dapat digunakan adalah sebagai berikut.

2 + 1

## [1] 3

sin(0)

## [1] 0

exp(1)

## [1] 2.718282

10^-6

## [1] 1e-06

### 2.3. Penugasan (*Assignment*)

Kita dapat menyimpan suatu objek dalam objek lain. Sebagai ilustrasi, misalkan kita akan menyimpan hasil penjumlahan 2 + 1 ke dalam suatu objek bernama a maka kita dapat melakukannya dengan cara:

a <- 2 + 1

Hasil penjumlahan 2 + 1 sudah dimasukkan ke dalam objek a. Untuk menampilkannya kita dapat menggunakan fungsi print() atau langsung mengeksekusi a.

print(a)

## [1] 3

a

## [1] 3

Operator “<-” adalah operator khas dan disukai pengguna R. Namun, selain operator “<-”, kita juga dapat menggunakan operator lain saat melakukan penugasan. Contoh-contohnya adalah sebagai berikut.

a + 5 -> b  
b

## [1] 8

d = 4 + 5  
d

## [1] 9

e <<- a # digunakan khusus saat pendefinisian objek secara global  
e

## [1] 3

begin <- "mari gunakan R"  
begin

## [1] "mari gunakan R"

Kita dapat menggunakan operator manapun sesuai dengan kesukaan dan kebutuhan.

### 2.4. Tipe class

Semua objek di R memiliki class yang dapat diketahui dengan menggunakan fungsi class(). Ada bermacam-macam tipe class dalam R tetapi pada bagian ini hanya disinggung delapan class yaitu logical, numeric, integer, character, factor, ordered, function, dan formula. Selain keenam jenis ini, masih banyak yang lainnya seperti class matrix, data.frame, list, Date, dan lain sebagainya.

#### 2.4.1. Logical

Tipe class yang dibahas pertama adalah class logical. Kelas ini sangat sederhana hanya terdiri dari dua objek yaitu TRUE dan FALSE. R menyediakan fungsi untuk mengecek kelas suatu objek seperti fungsi class(objek) atau is.jenis\_class(objek).

TRUE

## [1] TRUE

FALSE

## [1] FALSE

class(TRUE)

## [1] "logical"

is.logical(TRUE) # mengecek apakah objek TRUE merupakan logical

## [1] TRUE

Operasi yang berkaitan dengan logika akan menghasilkan objek TRUE atau FALSE. Misalnya:

2 < 1

## [1] FALSE

3\*5 <= 15

## [1] TRUE

4/2 == 2 # == adalah operator sama dengan

## [1] TRUE

5 - 2 != 2 # != adalah operator tidak sama dengan

## [1] TRUE

Selain operator di atas, masih banyak operator lain yang menghasilkan kelas logical seperti “>=”, “>”, “&”, “&&”, “|”, dan “||”.

#### 2.4.2. Numeric

Objek yang memiliki kelas numeric dapat menyimpan desimal maupun tidak.

numerik <- 1  
numerik

## [1] 1

class(numerik)

## [1] "numeric"

is.numeric(numerik) # mengecek apakah objek numerik merupakan logical

## [1] TRUE

2.645

## [1] 2.645

class(2.645)

## [1] "numeric"

#### 2.4.3. Integer

Sekilas mirip dengan numeric. Namun, seperti namanya, integer hanya terdiri dari bilangan bulat.

bulat <- 1:5  
bulat

## [1] 1 2 3 4 5

class(bulat)

## [1] "integer"

is.integer(bulat)

## [1] TRUE

is.numeric(bulat)

## [1] TRUE

Kita dapat merubah bilangan numerik menjadi integer dengan menggunakan fungsi as.integer(). Perubahan yang terjadi menuruti aturan pembulatan bilangan.

as.integer(2.3872)

## [1] 2

as.integer(3.5)

## [1] 3

#### 2.4.4. Character

Character, seperti pada bahasa pemrograman lain, memiliki sifat yang khusus pada R. Objek ber-class karakter dicirikan dengan diapit oleh tanda " " atau ‘’. Kedua tanda itu dapat kita gunakan.

karakter <- "R"  
karakter

## [1] "R"

class(karakter)

## [1] "character"

is.character(karakter)

## [1] TRUE

is.numeric(karakter)

## [1] FALSE

#### 2.4.5. Factor

Factor merupakan kelas yang spesial karena dapat ditampilkan seperti character tapi sebenarnya merupakan class yang digunakan untuk mengklasifikasikan sesuatu. Class factor ini seperti tipe skala pengukuran nominal pada skala pengukuran peubah. Hal yang membedakan antara faktor dan character adalah factor memiliki levels. Perhatikan contoh berikut ini.

faktor.1 <- factor(1)  
faktor.1

## [1] 1  
## Levels: 1

faktor.2 <- factor("R")  
faktor.2

## [1] R  
## Levels: R

class(faktor.1)

## [1] "factor"

is.character(faktor.2)

## [1] FALSE

is.numeric(faktor.1)

## [1] FALSE

Perlu diingat bahwa faktor tidak berclass character.

#### 2.4.6. Ordered Factor

Ordered factor adalah factor yang memiliki tingkatan. Jika dikaitkan dengan skala pengukuran peubah, ordered factor seperti skala pengukuran ordinal.

urutan <- ordered(1)  
urutan

## [1] 1  
## Levels: 1

class(urutan)

## [1] "ordered" "factor"

is.factor(urutan)

## [1] TRUE

is.numeric(urutan)

## [1] FALSE

#### 2.4.7. Function

R memiliki objek function (fungsi) tapi kita juga dapat menciptakan fungsi sendiri. Fungsi yang *default* dari R misalnya fungsi untuk menghitung rata-rata: mean(), simpangan baku: sd(), ragam: var(). Kita dapat menciptakan fungsi sendiri dengan menggunakan fungsi function(). Pada bagian lain, akan dijelaskan lebih banyak tentang menciptakan fungsi sendiri.

fungsi <- function(x) {x^2}  
fungsi

## function(x) {x^2}

class(fungsi)

## [1] "function"

is.function(fungsi)

## [1] TRUE

is.function(mean)

## [1] TRUE

is.function(factor)

## [1] TRUE

#### 2.4.8. Formula

Class formula ini banyak digunakan saat akan melakukan pemodelan.

formula(x~y)

## x ~ y

f = x~y  
class(f)

## [1] "formula"

Tanda tilde (~) pada banyak keyboard terletak di bawah tombol esc. Selain untuk pemodelan, class formula ini juga dapat digunakan saat akan membuat plot berdasarkan kelompok tertentu.

### 2.5. Manipulasi Sederhana Objek R

#### 2.5.1. Vector

Suatu vector hanya dapat diisi oleh objek yang memiliki class yang sama.

vektor.1 <- 1 #skalar  
vektor.1

## [1] 1

is.vector(vektor.1)

## [1] TRUE

Kita dapat membuat vektor yang berisi bilangan berurutan atau dengan pola teratur dengan menggunakan operator “:” dan fungsi seq(). Dalam fungsi seq() ada argumen yang harus diisi yaitu from = , to = , by = atau len = . Dalam tulisan ini, argumen-argumen dalam suatu fungsi, akan digunakan tanpa penjelasan lebih rinci.

vektor.2 <- 1:5 # vektor bilangan bulat secara sekuens  
vektor.2

## [1] 1 2 3 4 5

vektor.21 <- seq(from = 1, to = 5, by = 1)  
vektor.21

## [1] 1 2 3 4 5

vektor.22 <- seq(from = 1, to = 10, by = 2)  
vektor.22

## [1] 1 3 5 7 9

vektor.23 <- seq(from = 1, to = 100, len = 10) # argumen len tidak dapat digunakan bersama dengan argumen by!  
vektor.23

## [1] 1 12 23 34 45 56 67 78 89 100

Kita bisa juga membuat vektor dengan cara lain yaitu dengan menggunakan fungsi c() dan rep().

vektor.3 <- c(1,3,5,7,9,11) # vektor bilangan bulat bilangan tertentu  
vektor.3

## [1] 1 3 5 7 9 11

vektor.4 <- rep(1,times = 5) # vektor bilangan bulat bilangan yang diulang  
vektor.4

## [1] 1 1 1 1 1

vektor.5 <- rep(vektor.2, times = 2) # vektor bilangan bulat bilangan yang diulang  
vektor.5

## [1] 1 2 3 4 5 1 2 3 4 5

vektor.6 <- rep(c(1,2,3), each = 2) # vektor bilangan bulat bilangan yang diulang  
vektor.6

## [1] 1 1 2 2 3 3

Vektor selain berisi bilangan, dapat pula berisi character.

vektor.7 <- c("R", "Statistik") # vektor karakter atau string  
vektor.7

## [1] "R" "Statistik"

vektor.8 <- paste("X",c(1:5),sep = "")  
vektor.8

## [1] "X1" "X2" "X3" "X4" "X5"

vektor.9 <- paste(c("A", "B", "C"), c(1:3), sep = " ")  
vektor.9

## [1] "A 1" "B 2" "C 3"

vektor.10 <- paste(c("A", "B", "C"), c(1:6), sep = "")  
vektor.10

## [1] "A1" "B2" "C3" "A4" "B5" "C6"

Lebih jauh, kita juga dapat membuat vektor yang belum diisi oleh objek.

vektor.11 <- vector() # mendaftarkan objek vektor tapi belum terisi  
vektor.11

## logical(0)

is.vector(vektor.11)

## [1] TRUE

Vektor memiliki panjang vektor tapi tidak memiliki dimensi.

length(vektor.10)

## [1] 6

dim(vektor.10) # mengecek dimensi vektor (vektor tidak berdimensi!)

## NULL

Vektor juga dapat memuat objek dari class lain seperti Date yang merupakan format untuk data tanggal. Contoh pembuatan vektor tanggal diberikan pada bagian manipulasi data.

#### 2.5.2. Matrix

Seperti matriks pada matematika, matriks pada R juga memiliki baris dan kolom. Matriks memiliki dimensi sebanyak dua. Kita dapat menghitung panjang baris dengan fungsi nrow() dan panjang kolom dengan fungsi ncol(). Contoh-contoh berbagai cara membuat objek matriks adalah sebagai berikut.

matriks.1 <- matrix(c(1,2,3,4), nrow = 2, ncol = 2)  
matriks.1

## [,1] [,2]  
## [1,] 1 3  
## [2,] 2 4

dim(matriks.1)

## [1] 2 2

matriks.2 <- matrix(c(1,2,3,4), nrow = 2, ncol = 2, byrow = TRUE)  
matriks.2

## [,1] [,2]  
## [1,] 1 2  
## [2,] 3 4

matriks.3 <- vektor.10  
dim(matriks.3) <- c(2,3)  
matriks.3

## [,1] [,2] [,3]  
## [1,] "A1" "C3" "B5"  
## [2,] "B2" "A4" "C6"

matriks.4 <- cbind(vektor.4, vektor.8)  
matriks.4

## vektor.4 vektor.8  
## [1,] "1" "X1"   
## [2,] "1" "X2"   
## [3,] "1" "X3"   
## [4,] "1" "X4"   
## [5,] "1" "X5"

matriks.5 <- rbind(vektor.3, vektor.6)  
matriks.5

## [,1] [,2] [,3] [,4] [,5] [,6]  
## vektor.3 1 3 5 7 9 11  
## vektor.6 1 1 2 2 3 3

matriks.6 <- t(vektor.3)  
matriks.6

## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] 1 3 5 7 9 11

matriks.7 <- as.matrix(vektor.3)  
matriks.7

## [,1]  
## [1,] 1  
## [2,] 3  
## [3,] 5  
## [4,] 7  
## [5,] 9  
## [6,] 11

matriks.8 <- matrix()  
matriks.8

## [,1]  
## [1,] NA

is.matrix(matriks.8)

## [1] TRUE

Matriks hanya dapat digunakan untuk menyimpan class data yang sama.

#### 2.5.3. Data Frame

Data frame memiliki kelebihan dibanding vektor dan matriks yaitu dapat menyimpan data yang memiliki class berbeda dalam setiap kolomnya. Misalkan kita memiliki data numerik yang disimpan pada kolom pertama dan data kategorik pada kolom kedua. Asalkan panjang kedua kolomnya sama, maka dapat disimpan dengan baik dalam bentuk data frame. Meskipun tetap dapat disimpan dengan baik, data dengan panjang kolom berbeda membutuhkan trik khusus agar dapat informatif. Kita akan mencoba berbagai macam bentuk data frame seperti di bawah ini.

df.1 <- data.frame(c(80, 90, 85, 70, 95, 82),  
 c("B", "A", "A", "C", "A", "B"))  
df.1

## c.80..90..85..70..95..82. c..B....A....A....C....A....B..  
## 1 80 B  
## 2 90 A  
## 3 85 A  
## 4 70 C  
## 5 95 A  
## 6 82 B

dim(df.1)

## [1] 6 2

Bila kita tidak memberikan nama secara spesifik untuk data frame, maka akan diberi nama secara otomatis sesuai dengan kondisi data yang diinput. Pemberian nama kolom dapat dilakukan dengan mudah seperti contoh di bawah ini.

df.2 <- data.frame(X = c(80, 90, 85, 70, 95, 82),  
 Y = c("B", "A", "A", "C", "A", "B"))  
df.2

## X Y  
## 1 80 B  
## 2 90 A  
## 3 85 A  
## 4 70 C  
## 5 95 A  
## 6 82 B

Kita dapat membuat data frame dari berbagai struktur data lain seperti vektor, matriks, dan lain sebagainya.

df.3 <- data.frame(vektor.3)  
df.3

## vektor.3  
## 1 1  
## 2 3  
## 3 5  
## 4 7  
## 5 9  
## 6 11

df.4 <- data.frame(ganjil = vektor.3)  
df.4

## ganjil  
## 1 1  
## 2 3  
## 3 5  
## 4 7  
## 5 9  
## 6 11

df.5 <- data.frame(ganjil = vektor.3,  
 huruf = vektor.10)  
df.5

## ganjil huruf  
## 1 1 A1  
## 2 3 B2  
## 3 5 C3  
## 4 7 A4  
## 5 9 B5  
## 6 11 C6

df.6 <- cbind(df.2, df.5)  
df.6

## X Y ganjil huruf  
## 1 80 B 1 A1  
## 2 90 A 3 B2  
## 3 85 A 5 C3  
## 4 70 C 7 A4  
## 5 95 A 9 B5  
## 6 82 B 11 C6

Meskipun cbind digunakan dalam matriks, tapi jika kolom yang digabung (*bind*) merupakan data frame, maka hasil cbind akan berupa data frame.

df.7 <- data.frame(X = 92,  
 Y = "A")  
df.7

## X Y  
## 1 92 A

df.8 <- rbind(df.2, df.7)  
df.8

## X Y  
## 1 80 B  
## 2 90 A  
## 3 85 A  
## 4 70 C  
## 5 95 A  
## 6 82 B  
## 7 92 A

Meskipun bentuk data frame dan matriks memiliki kemiripan, tapi keduanya sangat berbeda. Jika kita memeriksanya dengan fungsi is.matrix() dan is.data.frame(), maka akan diperoleh keterangan bahwa matriks dan data frame berbeda.

is.matrix(df.6)

## [1] FALSE

is.data.frame(df.6)

## [1] TRUE

#### 2.5.4. Array

Array dapat mempunyai dimensi yang lebih banyak dari dua. Lebih besar daripada matriks dan data frame. Misalkan kita akan membuat array dari suatu bilangan dengan dimensi 3 yang terdiri dari 3 x 1 data terdalam yang ada pada 2 buah lipatan.

arai.1 <- array(c(1.5, 2.8, 3.62, 3, 5.6, 7.24), c(3,1,2))  
arai.1

## , , 1  
##   
## [,1]  
## [1,] 1.50  
## [2,] 2.80  
## [3,] 3.62  
##   
## , , 2  
##   
## [,1]  
## [1,] 3.00  
## [2,] 5.60  
## [3,] 7.24

dim(arai.1)

## [1] 3 1 2

Keterangan dimensi 3,1,2 menunjukkan bahwa data disimpan pada suatu struktur dua dimensi dengan baris sebanyak 3 dan kolom sebanyak 1 yang masing-masing ada pada bagian 1 dan 2. Kita dapat membuat berbagai variasi arrah seperti berikut ini.

arai.2 <- array(matriks.6)  
arai.2

## [1] 1 3 5 7 9 11

arai.3 <- array(matriks.6, c(1,2,3))  
arai.3

## , , 1  
##   
## [,1] [,2]  
## [1,] 1 3  
##   
## , , 2  
##   
## [,1] [,2]  
## [1,] 5 7  
##   
## , , 3  
##   
## [,1] [,2]  
## [1,] 9 11

arai.4 <- array(df.5, c(1,2,3))  
arai.4

## , , 1  
##   
## [,1] [,2]   
## [1,] Numeric,6 factor,6  
##   
## , , 2  
##   
## [,1] [,2]   
## [1,] Numeric,6 factor,6  
##   
## , , 3  
##   
## [,1] [,2]   
## [1,] Numeric,6 factor,6

arai.5 <- NULL  
arai.5

## NULL

Array banyak digunakan dalam pembuatan peta. Pada tulisan ini, array tidak terlalu banyak dijelaskan.

#### 2.5.5. List

Berbagai macam struktur dan kelas data data disimpan dalam list. Inilah yang membuat list sering digunakan untuk menyimpan luaran suatu fungsi.

list.1 <- list(c("Januari", "Februari", "Maret"))  
list.1

## [[1]]  
## [1] "Januari" "Februari" "Maret"

list.2 <- list(bulan = c("Januari", "Februari", "Maret"))  
list.2

## $bulan  
## [1] "Januari" "Februari" "Maret"

list.3 <- list(bulan = c("Januari", "Februari", "Maret"),  
 hari = c(31, 28, 31))  
list.3

## $bulan  
## [1] "Januari" "Februari" "Maret"   
##   
## $hari  
## [1] 31 28 31

list.4 <- list(list.3,  
 fungsi = fungsi)  
list.4

## [[1]]  
## [[1]]$bulan  
## [1] "Januari" "Februari" "Maret"   
##   
## [[1]]$hari  
## [1] 31 28 31  
##   
##   
## $fungsi  
## function (x)   
## {  
## x^2  
## }

## 2.6. Operasi Aritmatika

Dalam R dapat dilakukan berbagai macam operasi aritmatika sederhana sampai yang rumit. Berbagai contoh operasi aritmatika dalam R disajikan di bawah ini.

### 2.6.1. Penjumlahan dan pengurangan pada vektor

2 + 4

## [1] 6

c(1,2,3) - c(2,6,3)

## [1] -1 -4 0

10 - 11

## [1] -1

### 2.6.2. Perkalian dan pembagian pada vektor

vektor.2 \* 2

## [1] 2 4 6 8 10

2^4

## [1] 16

10 / 5

## [1] 2

vektor.5 / 2

## [1] 0.5 1.0 1.5 2.0 2.5 0.5 1.0 1.5 2.0 2.5

5 %% 2 # modulo

## [1] 1

5 %/% 2

## [1] 2

### 2.6.3. Pembulatan desimal

round(2.4)

## [1] 2

round(3.5)

## [1] 4

round(2.5)

## [1] 2

floor(2.6)

## [1] 2

ceiling(2.4)

## [1] 3

### 2.6.4. Penjumlahan matriks

matriks.1 + matriks.2

## [,1] [,2]  
## [1,] 2 5  
## [2,] 5 8

matriks.2 - matriks.1

## [,1] [,2]  
## [1,] 0 -1  
## [2,] 1 0

matriks.1 %\*% matriks.2

## [,1] [,2]  
## [1,] 10 14  
## [2,] 14 20

matriks.1 \* matriks.2 # perkalian antar unsur

## [,1] [,2]  
## [1,] 1 6  
## [2,] 6 16

## 3. Input Data

### 3.1. Internal

# Data yang sudah ada dalam R  
data(iris)  
head(iris) # beberapa data terawal

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 1 5.1 3.5 1.4 0.2 setosa  
## 2 4.9 3.0 1.4 0.2 setosa  
## 3 4.7 3.2 1.3 0.2 setosa  
## 4 4.6 3.1 1.5 0.2 setosa  
## 5 5.0 3.6 1.4 0.2 setosa  
## 6 5.4 3.9 1.7 0.4 setosa

tail(iris) # beberapa data terakhir

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 145 6.7 3.3 5.7 2.5 virginica  
## 146 6.7 3.0 5.2 2.3 virginica  
## 147 6.3 2.5 5.0 1.9 virginica  
## 148 6.5 3.0 5.2 2.0 virginica  
## 149 6.2 3.4 5.4 2.3 virginica  
## 150 5.9 3.0 5.1 1.8 virginica

# Melakukan input dengan berbagai tipe struktur objek  
tb <- c(165, 172, 166, 168)  
tb

## [1] 165 172 166 168

jk <- data.frame(nama = c("Dhini", "Nanda", "Mika", "Herianti", "Asep"),  
 jk = c(0, 1, 0, 0, 1))

Kita juga dapat memanfaatkan fungsi scan. Caranya adalah dengan menggunakan fungsi scan() dan menuliskan di console nilai-nilai yang diinput. Proses input diakhiri dengan enter dua kali. pindai.vektor <- scan() 1: 6 2: 7 3: 7 4: Read 3 items pindai.vektor [1] 6 7 8

Teknik-teknik lebih rinci disajikan pada saat presentasi.

### 3.2. Eksternal

cs <- read.csv("D:/Pelatihan R/carbon stock.csv")  
head(cs)

## plot.site tree.stand root necromass litter herbacious  
## 1 mono 55.21 13.80 2.69 0.85 0.02  
## 2 mono 62.23 15.56 1.11 0.88 0.16  
## 3 mono 42.05 10.51 1.16 0.92 0.03  
## 4 mono 49.69 12.42 0.00 1.03 0.09  
## 5 mono 40.90 10.22 0.09 1.29 0.00  
## 6 mono 48.70 12.17 0.42 0.97 0.03  
## soil.organic.carbon total  
## 1 17.75 90.32  
## 2 75.02 154.95  
## 3 34.72 89.39  
## 4 29.75 92.98  
## 5 48.03 100.54  
## 6 40.13 102.41

Kita juga dapat menggunakan menu bar. Teknik-teknik lebih rinci disajikan pada saat presentasi.

## 4. Manipulasi Data

### 4.1. Vektor

# Mengakses unsur vektor  
vek.1 = 11:15  
vek.1

## [1] 11 12 13 14 15

vek.1[1]

## [1] 11

vek.1[1:3]

## [1] 11 12 13

vek.1[c(1,2,5)]

## [1] 11 12 15

vek.1[-c(1,2)]

## [1] 13 14 15

vek.1[6] <- 10  
vek.1[9] <- 11  
vek.2 <- c("a", "b", "c", "d")  
vek.2[c(1,3)]

## [1] "a" "c"

vek.2[7] <- "e"  
vek.2

## [1] "a" "b" "c" "d" NA NA "e"

vek.1 < 13

## [1] TRUE TRUE FALSE FALSE FALSE TRUE NA NA TRUE

vek.1[vek.1 < 13]

## [1] 11 12 10 NA NA 11

vek.3 <- vek.1[vek.1 < 13]  
vek.3

## [1] 11 12 10 NA NA 11

# Mengecek na  
is.na(vek.3)

## [1] FALSE FALSE FALSE TRUE TRUE FALSE

!is.na(vek.3)

## [1] TRUE TRUE TRUE FALSE FALSE TRUE

vek.4 <- vek.3[!is.na(vek.3)]  
vek.4

## [1] 11 12 10 11

# Menggunakan fungsi subset  
vek.5 <- subset(vek.3, subset = vek.3 < 13 & !is.na(vek.3))  
vek.5

## [1] 11 12 10 11

# Mengakses vektor string atau character  
vek.string.1 <- "R untuk statistik"  
vek.string.1

## [1] "R untuk statistik"

vek.string.2 <- substr(vek.string.1, start = 1, stop = 1)  
vek.string.2

## [1] "R"

vek.string.3 <- substr(vek.string.1, start = 1, stop = 7)  
vek.string.3

## [1] "R untuk"

vek.string.4 <- substr(vek.string.1, start = 3, stop = 1000)  
vek.string.4

## [1] "untuk statistik"

# Vektor tanggal  
x <- c("1jan1960", "2jan1960", "31mar1960", "30jul1960")  
x

## [1] "1jan1960" "2jan1960" "31mar1960" "30jul1960"

class(x)

## [1] "character"

z <- as.Date(x, format = "%d%b%Y")  
z

## [1] "1960-01-01" "1960-01-02" "1960-03-31" "1960-07-30"

class(z)

## [1] "Date"

dates.char <- c("02/27/92", "02/27/92", "01/14/92", "02/28/92", "02/01/92")  
dates

## Error in eval(expr, envir, enclos): object 'dates' not found

class(dates.char)

## [1] "character"

dates <- as.Date(dates, "%m/%d/%y")

## Error in as.Date(dates, "%m/%d/%y"): object 'dates' not found

# Mengurutkan vektor  
order(vek.1, decreasing = TRUE)

## [1] 5 4 3 2 1 9 6 7 8

vek.1.turun <- vek.1[order(vek.1, decreasing = TRUE)]  
vek.1.turun

## [1] 15 14 13 12 11 11 10 NA NA

vek.1.turun.juga <- sort(vek.1, decreasing = TRUE, na.last = TRUE) # hanya berlaku untuk vektor  
vek.1.turun.juga

## [1] 15 14 13 12 11 11 10 NA NA

vek.1.naik <- vek.1[order(vek.1, decreasing = FALSE)]  
vek.1.naik

## [1] 10 11 11 12 13 14 15 NA NA

vek.1.naik.juga.1 <- vek.1[order(vek.1)]  
vek.1.naik.juga.1

## [1] 10 11 11 12 13 14 15 NA NA

vek.1.naik.juga.2 <- sort(vek.1, na.last = TRUE)  
vek.1.naik.juga.2

## [1] 10 11 11 12 13 14 15 NA NA

vek.1.naik.juga.3 <- sort(vek.1, na.last = FALSE)  
vek.1.naik.juga.3

## [1] NA NA 10 11 11 12 13 14 15

vek.1.naik.tanpa.na <- sort(vek.1)  
vek.1.naik.tanpa.na

## [1] 10 11 11 12 13 14 15

### 4.2. Data Frame

# Mengakses data frame  
cs.1 <- cs[1,]  
cs.1

## plot.site tree.stand root necromass litter herbacious  
## 1 mono 55.21 13.8 2.69 0.85 0.02  
## soil.organic.carbon total  
## 1 17.75 90.32

# melihat nama variabel  
names(cs.1)

## [1] "plot.site" "tree.stand" "root"   
## [4] "necromass" "litter" "herbacious"   
## [7] "soil.organic.carbon" "total"

cs.root <- cs[,3]  
cs.root

## [1] 13.80 15.56 10.51 12.42 10.22 12.17 14.28 4.16 14.29 11.98 20.81  
## [12] 10.16 24.88 21.23 11.65 22.95 19.45 8.45 5.80 6.25

cs.root.juga <- cs[,"root"]  
cs.root.juga

## [1] 13.80 15.56 10.51 12.42 10.22 12.17 14.28 4.16 14.29 11.98 20.81  
## [12] 10.16 24.88 21.23 11.65 22.95 19.45 8.45 5.80 6.25

cs.2 <- cs[c(1:5),c(1:3)]  
cs.2

## plot.site tree.stand root  
## 1 mono 55.21 13.80  
## 2 mono 62.23 15.56  
## 3 mono 42.05 10.51  
## 4 mono 49.69 12.42  
## 5 mono 40.90 10.22

cs.3 <- cs[c(1:3,6),c(1,3,4)]  
cs.3

## plot.site root necromass  
## 1 mono 13.80 2.69  
## 2 mono 15.56 1.11  
## 3 mono 10.51 1.16  
## 6 mono 12.17 0.42

cs.mono <- cs[cs$plot.site == "mono",]  
cs.mono

## plot.site tree.stand root necromass litter herbacious  
## 1 mono 55.21 13.80 2.69 0.85 0.02  
## 2 mono 62.23 15.56 1.11 0.88 0.16  
## 3 mono 42.05 10.51 1.16 0.92 0.03  
## 4 mono 49.69 12.42 0.00 1.03 0.09  
## 5 mono 40.90 10.22 0.09 1.29 0.00  
## 6 mono 48.70 12.17 0.42 0.97 0.03  
## 7 mono 57.12 14.28 0.47 0.53 0.07  
## 8 mono 16.62 4.16 2.26 1.16 0.04  
## 9 mono 57.15 14.29 0.15 1.18 0.00  
## 10 mono 47.92 11.98 3.88 0.95 0.02  
## soil.organic.carbon total  
## 1 17.75 90.32  
## 2 75.02 154.95  
## 3 34.72 89.39  
## 4 29.75 92.98  
## 5 48.03 100.54  
## 6 40.13 102.41  
## 7 75.71 148.19  
## 8 29.12 53.36  
## 9 42.00 114.77  
## 10 35.55 100.31

# Menggunakan fungsi subset  
cs.mono.juga <- subset(cs, subset = plot.site == "mono")  
cs.mono.juga

## plot.site tree.stand root necromass litter herbacious  
## 1 mono 55.21 13.80 2.69 0.85 0.02  
## 2 mono 62.23 15.56 1.11 0.88 0.16  
## 3 mono 42.05 10.51 1.16 0.92 0.03  
## 4 mono 49.69 12.42 0.00 1.03 0.09  
## 5 mono 40.90 10.22 0.09 1.29 0.00  
## 6 mono 48.70 12.17 0.42 0.97 0.03  
## 7 mono 57.12 14.28 0.47 0.53 0.07  
## 8 mono 16.62 4.16 2.26 1.16 0.04  
## 9 mono 57.15 14.29 0.15 1.18 0.00  
## 10 mono 47.92 11.98 3.88 0.95 0.02  
## soil.organic.carbon total  
## 1 17.75 90.32  
## 2 75.02 154.95  
## 3 34.72 89.39  
## 4 29.75 92.98  
## 5 48.03 100.54  
## 6 40.13 102.41  
## 7 75.71 148.19  
## 8 29.12 53.36  
## 9 42.00 114.77  
## 10 35.55 100.31

cs.4 <- subset(cs, subset = plot.site == "mono", select = c("plot.site", "root", "litter"))  
cs.4

## plot.site root litter  
## 1 mono 13.80 0.85  
## 2 mono 15.56 0.88  
## 3 mono 10.51 0.92  
## 4 mono 12.42 1.03  
## 5 mono 10.22 1.29  
## 6 mono 12.17 0.97  
## 7 mono 14.28 0.53  
## 8 mono 4.16 1.16  
## 9 mono 14.29 1.18  
## 10 mono 11.98 0.95

cs.4$ket.root <- ifelse(cs.4$root > 10, "Tinggi", "Rendah")  
head(cs.4)

## plot.site root litter ket.root  
## 1 mono 13.80 0.85 Tinggi  
## 2 mono 15.56 0.88 Tinggi  
## 3 mono 10.51 0.92 Tinggi  
## 4 mono 12.42 1.03 Tinggi  
## 5 mono 10.22 1.29 Tinggi  
## 6 mono 12.17 0.97 Tinggi

cs.5 <- cs.4  
cs.5$ket.root <- ifelse(cs.5$root > 12, "Tinggi",  
 ifelse(cs.5$root <= 12 & cs.5$root > 10, "Sedang","Rendah"))  
head(cs.5)

## plot.site root litter ket.root  
## 1 mono 13.80 0.85 Tinggi  
## 2 mono 15.56 0.88 Tinggi  
## 3 mono 10.51 0.92 Sedang  
## 4 mono 12.42 1.03 Tinggi  
## 5 mono 10.22 1.29 Sedang  
## 6 mono 12.17 0.97 Tinggi

# Mengurutkan data frame berdasarkan 1 variabel  
cs.urut.1 <- cs[order(cs$total),]  
head(cs.urut.1)

## plot.site tree.stand root necromass litter herbacious  
## 8 mono 16.62 4.16 2.26 1.16 0.04  
## 18 agro 33.79 8.45 0.07 0.77 0.14  
## 20 agro 24.98 6.25 0.31 0.94 0.10  
## 12 agro 40.65 10.16 0.09 1.26 0.07  
## 19 agro 23.21 5.80 25.10 0.64 0.02  
## 3 mono 42.05 10.51 1.16 0.92 0.03  
## soil.organic.carbon total  
## 8 29.12 53.36  
## 18 35.96 79.17  
## 20 47.63 80.22  
## 12 32.31 84.55  
## 19 33.37 88.14  
## 3 34.72 89.39

cs.urut.2 <- cs[order(cs$root, cs$necromass, decreasing = c(T,F)),]  
head(cs.urut.2)

## plot.site tree.stand root necromass litter herbacious  
## 13 agro 99.53 24.88 0.25 0.88 0.12  
## 16 agro 91.80 22.95 5.35 0.50 0.38  
## 14 agro 84.93 21.23 0.00 0.61 0.09  
## 11 agro 83.23 20.81 0.84 1.16 0.17  
## 17 agro 77.79 19.45 0.23 0.60 0.06  
## 2 mono 62.23 15.56 1.11 0.88 0.16  
## soil.organic.carbon total  
## 13 56.51 182.17  
## 16 50.51 171.50  
## 14 46.33 153.21  
## 11 38.35 144.55  
## 17 67.24 165.37  
## 2 75.02 154.95

# Mengurutkan data frame berdasarkan lebih dari 1 variabel  
df.X <- data.frame(Faktor.1 = rep(rep(c(1:3), each = 3), times = 2),  
 Faktor.2 = rep(rep(c(1:3),times = 3), times = 2),  
 r = rep(c(1,2), each = 9))  
df.X

## Faktor.1 Faktor.2 r  
## 1 1 1 1  
## 2 1 2 1  
## 3 1 3 1  
## 4 2 1 1  
## 5 2 2 1  
## 6 2 3 1  
## 7 3 1 1  
## 8 3 2 1  
## 9 3 3 1  
## 10 1 1 2  
## 11 1 2 2  
## 12 1 3 2  
## 13 2 1 2  
## 14 2 2 2  
## 15 2 3 2  
## 16 3 1 2  
## 17 3 2 2  
## 18 3 3 2

df.X[order(df.X$Faktor.1, df.X$Faktor.2, decreasing = c(T,F)),]

## Faktor.1 Faktor.2 r  
## 7 3 1 1  
## 16 3 1 2  
## 8 3 2 1  
## 17 3 2 2  
## 9 3 3 1  
## 18 3 3 2  
## 4 2 1 1  
## 13 2 1 2  
## 5 2 2 1  
## 14 2 2 2  
## 6 2 3 1  
## 15 2 3 2  
## 1 1 1 1  
## 10 1 1 2  
## 2 1 2 1  
## 11 1 2 2  
## 3 1 3 1  
## 12 1 3 2

# Memberi nama variabel  
colnames(df.X) <- c("Dosis", "Waktu", "Ulangan")  
df.X

## Dosis Waktu Ulangan  
## 1 1 1 1  
## 2 1 2 1  
## 3 1 3 1  
## 4 2 1 1  
## 5 2 2 1  
## 6 2 3 1  
## 7 3 1 1  
## 8 3 2 1  
## 9 3 3 1  
## 10 1 1 2  
## 11 1 2 2  
## 12 1 3 2  
## 13 2 1 2  
## 14 2 2 2  
## 15 2 3 2  
## 16 3 1 2  
## 17 3 2 2  
## 18 3 3 2

df.X.Dosis.2 <- subset(df.X, Dosis == 2)  
df.X.Dosis.2

## Dosis Waktu Ulangan  
## 4 2 1 1  
## 5 2 2 1  
## 6 2 3 1  
## 13 2 1 2  
## 14 2 2 2  
## 15 2 3 2

rownames(df.X.Dosis.2) <- 1:6  
df.X.Dosis.2

## Dosis Waktu Ulangan  
## 1 2 1 1  
## 2 2 2 1  
## 3 2 3 1  
## 4 2 1 2  
## 5 2 2 2  
## 6 2 3 2

# Melakukan merge data frame  
nilai.1 = data.frame(nama = c("Andri", "Ofi", "Amel", "Dewa", "Sam","Juli"),  
 ujian.1 = c(80, 75, 82, 65, 72, 95))  
nilai.1

## nama ujian.1  
## 1 Andri 80  
## 2 Ofi 75  
## 3 Amel 82  
## 4 Dewa 65  
## 5 Sam 72  
## 6 Juli 95

nilai.2 = data.frame(nama = c("Ofi", "Amel", "Andri", "Dewa", "Sam"),  
 ujian.2 = c(77, 90, 85, 80, 92))  
nilai.2

## nama ujian.2  
## 1 Ofi 77  
## 2 Amel 90  
## 3 Andri 85  
## 4 Dewa 80  
## 5 Sam 92

nilai.irisan = merge(nilai.1, nilai.2)  
nilai.irisan

## nama ujian.1 ujian.2  
## 1 Amel 82 90  
## 2 Andri 80 85  
## 3 Dewa 65 80  
## 4 Ofi 75 77  
## 5 Sam 72 92

nilai.gabungan.1 = merge(nilai.1, nilai.2, all = TRUE)  
nilai.gabungan.1

## nama ujian.1 ujian.2  
## 1 Amel 82 90  
## 2 Andri 80 85  
## 3 Dewa 65 80  
## 4 Juli 95 NA  
## 5 Ofi 75 77  
## 6 Sam 72 92

nilai.gabungan.2 = merge(nilai.1, nilai.2, all = TRUE, sort = FALSE)  
nilai.gabungan.2

## nama ujian.1 ujian.2  
## 1 Andri 80 85  
## 2 Ofi 75 77  
## 3 Amel 82 90  
## 4 Dewa 65 80  
## 5 Sam 72 92  
## 6 Juli 95 NA

### 4.3. Matrix

mat.1 <- matrix(c(1,2,3,4), nrow = 2, ncol = 2, byrow = T)  
mat.1

## [,1] [,2]  
## [1,] 1 2  
## [2,] 3 4

mat.2 <- matrix(c(11:16), nrow = 2, ncol = 3, byrow = F)  
mat.2

## [,1] [,2] [,3]  
## [1,] 11 13 15  
## [2,] 12 14 16

mat.3 <- rbind(mat.1, mat.2)

## Error in rbind(mat.1, mat.2): number of columns of matrices must match (see arg 2)

mat.3

## Error in eval(expr, envir, enclos): object 'mat.3' not found

mat.4 <- cbind(mat.1, mat.2)  
mat.4

## [,1] [,2] [,3] [,4] [,5]  
## [1,] 1 2 11 13 15  
## [2,] 3 4 12 14 16

mat.5 <- mat.1 %\*% mat.2  
mat.5

## [,1] [,2] [,3]  
## [1,] 35 41 47  
## [2,] 81 95 109

# determinan matriks  
determinan <- det(mat.1)  
determinan

## [1] -2

# inverse matriks  
inverse <- solve(mat.1)  
inverse

## [,1] [,2]  
## [1,] -2.0 1.0  
## [2,] 1.5 -0.5

# eigen value dan eigen vektor  
eigen(mat.1)

## eigen() decomposition  
## $values  
## [1] 5.3722813 -0.3722813  
##   
## $vectors  
## [,1] [,2]  
## [1,] -0.4159736 -0.8245648  
## [2,] -0.9093767 0.5657675

vektor.diag <- diag(mat.4)  
vektor.diag

## [1] 1 4

unsur <- mat.4[2:4]  
unsur

## [1] 3 2 4

vektor.baris <- mat.4[1,]  
vektor.baris

## [1] 1 2 11 13 15

mat.6 <- mat.4[1:2, c(1,3)]  
mat.6

## [,1] [,2]  
## [1,] 1 11  
## [2,] 3 12

### 4.4. List

list.41 <- list(vektor = vektor.1,  
 data.frame = cs.mono,  
 matriks = mat.1)  
list.41

## $vektor  
## [1] 1  
##   
## $data.frame  
## plot.site tree.stand root necromass litter herbacious  
## 1 mono 55.21 13.80 2.69 0.85 0.02  
## 2 mono 62.23 15.56 1.11 0.88 0.16  
## 3 mono 42.05 10.51 1.16 0.92 0.03  
## 4 mono 49.69 12.42 0.00 1.03 0.09  
## 5 mono 40.90 10.22 0.09 1.29 0.00  
## 6 mono 48.70 12.17 0.42 0.97 0.03  
## 7 mono 57.12 14.28 0.47 0.53 0.07  
## 8 mono 16.62 4.16 2.26 1.16 0.04  
## 9 mono 57.15 14.29 0.15 1.18 0.00  
## 10 mono 47.92 11.98 3.88 0.95 0.02  
## soil.organic.carbon total  
## 1 17.75 90.32  
## 2 75.02 154.95  
## 3 34.72 89.39  
## 4 29.75 92.98  
## 5 48.03 100.54  
## 6 40.13 102.41  
## 7 75.71 148.19  
## 8 29.12 53.36  
## 9 42.00 114.77  
## 10 35.55 100.31  
##   
## $matriks  
## [,1] [,2]  
## [1,] 1 2  
## [2,] 3 4

ls(list.41)

## [1] "data.frame" "matriks" "vektor"

list.41[[1]]

## [1] 1

list.41$vektor

## [1] 1

list.41$data.frame

## plot.site tree.stand root necromass litter herbacious  
## 1 mono 55.21 13.80 2.69 0.85 0.02  
## 2 mono 62.23 15.56 1.11 0.88 0.16  
## 3 mono 42.05 10.51 1.16 0.92 0.03  
## 4 mono 49.69 12.42 0.00 1.03 0.09  
## 5 mono 40.90 10.22 0.09 1.29 0.00  
## 6 mono 48.70 12.17 0.42 0.97 0.03  
## 7 mono 57.12 14.28 0.47 0.53 0.07  
## 8 mono 16.62 4.16 2.26 1.16 0.04  
## 9 mono 57.15 14.29 0.15 1.18 0.00  
## 10 mono 47.92 11.98 3.88 0.95 0.02  
## soil.organic.carbon total  
## 1 17.75 90.32  
## 2 75.02 154.95  
## 3 34.72 89.39  
## 4 29.75 92.98  
## 5 48.03 100.54  
## 6 40.13 102.41  
## 7 75.71 148.19  
## 8 29.12 53.36  
## 9 42.00 114.77  
## 10 35.55 100.31

## 6. Looping dan Control Flow

### 6.1 Looping

#### 6.1.1 For

for (i in 1:10){  
 print(paste("ulangan ke-", i, sep = ""))  
}

## [1] "ulangan ke-1"  
## [1] "ulangan ke-2"  
## [1] "ulangan ke-3"  
## [1] "ulangan ke-4"  
## [1] "ulangan ke-5"  
## [1] "ulangan ke-6"  
## [1] "ulangan ke-7"  
## [1] "ulangan ke-8"  
## [1] "ulangan ke-9"  
## [1] "ulangan ke-10"

#### 6.1.2 While

j = 1  
while (j < 11) {  
 print(paste("ulangan ke-", j, sep = ""))  
 j = j + 1  
}

## [1] "ulangan ke-1"  
## [1] "ulangan ke-2"  
## [1] "ulangan ke-3"  
## [1] "ulangan ke-4"  
## [1] "ulangan ke-5"  
## [1] "ulangan ke-6"  
## [1] "ulangan ke-7"  
## [1] "ulangan ke-8"  
## [1] "ulangan ke-9"  
## [1] "ulangan ke-10"

### 6.2 Control Flow

m <- 10   
n <- 10  
ctr <- 0  
mymat <- matrix(0,m,n)  
for(i in 1:m) {  
 for(j in 1:n) {   
 if(i==j) {   
 break;  
 } else {  
 # you assign the values only when i<>j  
 mymat[i,j] = i\*j  
 ctr=ctr+1  
 }  
 }  
 print(i\*j)   
}

## [1] 1  
## [1] 4  
## [1] 9  
## [1] 16  
## [1] 25  
## [1] 36  
## [1] 49  
## [1] 64  
## [1] 81  
## [1] 100

### 6.3 Menyimpan Hasil Looping

for(i in c(2,4,6,8,10)){  
 x <- i  
 x.kuadrat <- i^2  
 hasil.1 <- data.frame(x,x.kuadrat)  
}  
hasil.1

## x x.kuadrat  
## 1 10 100

hasil.2 <- data.frame()  
for(i in c(2,4,6,8,10)){  
 x <- i  
 x.kuadrat <- i^2  
 df.x <- data.frame(x,x.kuadrat)  
 hasil.2 <- rbind(hasil.2, df.x)  
}  
hasil.2

## x x.kuadrat  
## 1 2 4  
## 2 4 16  
## 3 6 36  
## 4 8 64  
## 5 10 100

### 6.4 Membuat fungsi sendiri  
fungsiGanjil <- function(x) {  
 index = x %% 2 == 1  
 hasil = x[index]  
 return(hasil)  
}  
fungsiGanjil(1:10)

## [1] 1 3 5 7 9

fungsiGanjilGenap <- function(x) {  
 index.ganjil = x %% 2 == 1  
 hasil.ganjil = x[index.ganjil]  
 index.genap = x %% 2 ==0  
 hasil.genap = x[index.genap]  
 return(list(ganjil = hasil.ganjil, genap = hasil.genap))  
}  
  
fungsiGanjilGenapPilihan <- function(x,pilihan) {  
 if (pilihan == "ganjil") {  
 return(hasil.ganjil = x[x %% 2 == 1])  
 } else if (pilihan == "genap") {  
 return(hasil.genap = x[x %% 2 ==0])  
 }  
}  
fungsiGanjilGenapPilihan(1:10,"ganjil")

## [1] 1 3 5 7 9

fungsiGanjilGenapPilihan(1:10,"genap")

## [1] 2 4 6 8 10

fungsiKorelasiPearson <- function(x,y) {  
 if(is.numeric(x) == FALSE) {  
 stop("Variabel X harus numeric")  
 } else if(is.numeric(y) == FALSE) {  
 stop("Variabel Y harus numeric")  
 } else {  
 var.x <- var(x)  
 var.y <- var(y)  
 cov.xy <- cov(x,y)  
 r <- cov.xy/(sqrt(var.x\*var.y))  
 return(r)  
 }  
}

## 7. Statistik Deskriptif

### 7.1. Statistik

data(mtcars)  
head(mtcars)

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Mazda RX4 21.0 6 160 110 3.90 2.620 16.46 0 1 4 4  
## Mazda RX4 Wag 21.0 6 160 110 3.90 2.875 17.02 0 1 4 4  
## Datsun 710 22.8 4 108 93 3.85 2.320 18.61 1 1 4 1  
## Hornet 4 Drive 21.4 6 258 110 3.08 3.215 19.44 1 0 3 1  
## Hornet Sportabout 18.7 8 360 175 3.15 3.440 17.02 0 0 3 2  
## Valiant 18.1 6 225 105 2.76 3.460 20.22 1 0 3 1

# Ringkasan data termasuk statistik lima serangkai  
summary(mtcars)

## mpg cyl disp hp   
## Min. :10.40 Min. :4.000 Min. : 71.1 Min. : 52.0   
## 1st Qu.:15.43 1st Qu.:4.000 1st Qu.:120.8 1st Qu.: 96.5   
## Median :19.20 Median :6.000 Median :196.3 Median :123.0   
## Mean :20.09 Mean :6.188 Mean :230.7 Mean :146.7   
## 3rd Qu.:22.80 3rd Qu.:8.000 3rd Qu.:326.0 3rd Qu.:180.0   
## Max. :33.90 Max. :8.000 Max. :472.0 Max. :335.0   
## drat wt qsec vs   
## Min. :2.760 Min. :1.513 Min. :14.50 Min. :0.0000   
## 1st Qu.:3.080 1st Qu.:2.581 1st Qu.:16.89 1st Qu.:0.0000   
## Median :3.695 Median :3.325 Median :17.71 Median :0.0000   
## Mean :3.597 Mean :3.217 Mean :17.85 Mean :0.4375   
## 3rd Qu.:3.920 3rd Qu.:3.610 3rd Qu.:18.90 3rd Qu.:1.0000   
## Max. :4.930 Max. :5.424 Max. :22.90 Max. :1.0000   
## am gear carb   
## Min. :0.0000 Min. :3.000 Min. :1.000   
## 1st Qu.:0.0000 1st Qu.:3.000 1st Qu.:2.000   
## Median :0.0000 Median :4.000 Median :2.000   
## Mean :0.4062 Mean :3.688 Mean :2.812   
## 3rd Qu.:1.0000 3rd Qu.:4.000 3rd Qu.:4.000   
## Max. :1.0000 Max. :5.000 Max. :8.000

# Menghitung jumlah dari masing-masing baris (jumlah dapat diganti dengan fungsi apapun)  
apply(mtcars, MARGIN = 1, FUN = sum)

## Mazda RX4 Mazda RX4 Wag Datsun 710   
## 328.980 329.795 259.580   
## Hornet 4 Drive Hornet Sportabout Valiant   
## 426.135 590.310 385.540   
## Duster 360 Merc 240D Merc 230   
## 656.920 270.980 299.570   
## Merc 280 Merc 280C Merc 450SE   
## 350.460 349.660 510.740   
## Merc 450SL Merc 450SLC Cadillac Fleetwood   
## 511.500 509.850 728.560   
## Lincoln Continental Chrysler Imperial Fiat 128   
## 726.644 725.695 213.850   
## Honda Civic Toyota Corolla Toyota Corona   
## 195.165 206.955 273.775   
## Dodge Challenger AMC Javelin Camaro Z28   
## 519.650 506.085 646.280   
## Pontiac Firebird Fiat X1-9 Porsche 914-2   
## 631.175 208.215 272.570   
## Lotus Europa Ford Pantera L Ferrari Dino   
## 273.683 670.690 379.590   
## Maserati Bora Volvo 142E   
## 694.710 288.890

rowSums(mtcars)

## Mazda RX4 Mazda RX4 Wag Datsun 710   
## 328.980 329.795 259.580   
## Hornet 4 Drive Hornet Sportabout Valiant   
## 426.135 590.310 385.540   
## Duster 360 Merc 240D Merc 230   
## 656.920 270.980 299.570   
## Merc 280 Merc 280C Merc 450SE   
## 350.460 349.660 510.740   
## Merc 450SL Merc 450SLC Cadillac Fleetwood   
## 511.500 509.850 728.560   
## Lincoln Continental Chrysler Imperial Fiat 128   
## 726.644 725.695 213.850   
## Honda Civic Toyota Corolla Toyota Corona   
## 195.165 206.955 273.775   
## Dodge Challenger AMC Javelin Camaro Z28   
## 519.650 506.085 646.280   
## Pontiac Firebird Fiat X1-9 Porsche 914-2   
## 631.175 208.215 272.570   
## Lotus Europa Ford Pantera L Ferrari Dino   
## 273.683 670.690 379.590   
## Maserati Bora Volvo 142E   
## 694.710 288.890

# Menghitung rata-rata dari masing-masing kolom (rata-rata dapat diganti dengan fungsi apapun)  
apply(mtcars, MARGIN = 2, FUN = mean)

## mpg cyl disp hp drat wt   
## 20.090625 6.187500 230.721875 146.687500 3.596563 3.217250   
## qsec vs am gear carb   
## 17.848750 0.437500 0.406250 3.687500 2.812500

colMeans(mtcars)

## mpg cyl disp hp drat wt   
## 20.090625 6.187500 230.721875 146.687500 3.596563 3.217250   
## qsec vs am gear carb   
## 17.848750 0.437500 0.406250 3.687500 2.812500

# Menghitung rata-rata agregat (rata-rata dapat diganti dengan fungsi apapun)  
aggregate(list(rataan.mpg = mtcars$mpg), list(carb = mtcars$carb), mean)

## carb rataan.mpg  
## 1 1 25.34286  
## 2 2 22.40000  
## 3 3 16.30000  
## 4 4 15.79000  
## 5 6 19.70000  
## 6 8 15.00000

aggregate(list(rataan.mpg = mtcars$mpg),  
 list(carb = mtcars$carb, gear = mtcars$gear), mean)

## carb gear rataan.mpg  
## 1 1 3 20.33333  
## 2 2 3 17.15000  
## 3 3 3 16.30000  
## 4 4 3 12.62000  
## 5 1 4 29.10000  
## 6 2 4 24.75000  
## 7 4 4 19.75000  
## 8 2 5 28.20000  
## 9 4 5 15.80000  
## 10 6 5 19.70000  
## 11 8 5 15.00000

aggregate(list(rataan.mpg = mtcars$mpg, rataan.wt = mtcars$wt),   
 list(carb = mtcars$carb, gear = mtcars$gear), mean)

## carb gear rataan.mpg rataan.wt  
## 1 1 3 20.33333 3.046667  
## 2 2 3 17.15000 3.560000  
## 3 3 3 16.30000 3.860000  
## 4 4 3 12.62000 4.685800  
## 5 1 4 29.10000 2.072500  
## 6 2 4 24.75000 2.683750  
## 7 4 4 19.75000 3.093750  
## 8 2 5 28.20000 1.826500  
## 9 4 5 15.80000 3.170000  
## 10 6 5 19.70000 2.770000  
## 11 8 5 15.00000 3.570000

# Menyimpan hasil menjadi suatu objek  
rataan.mpg <- aggregate(list(rataan.mpg = mtcars$mpg), list(carb = mtcars$carb), mean)  
rataan.mpg

## carb rataan.mpg  
## 1 1 25.34286  
## 2 2 22.40000  
## 3 3 16.30000  
## 4 4 15.79000  
## 5 6 19.70000  
## 6 8 15.00000

simbaku.mpg <- aggregate(list(simbaku.mpg = mtcars$mpg), list(carb = mtcars$carb), sd)  
simbaku.mpg

## carb simbaku.mpg  
## 1 1 6.001349  
## 2 2 5.472152  
## 3 3 1.053565  
## 4 4 3.911081  
## 5 6 NA  
## 6 8 NA

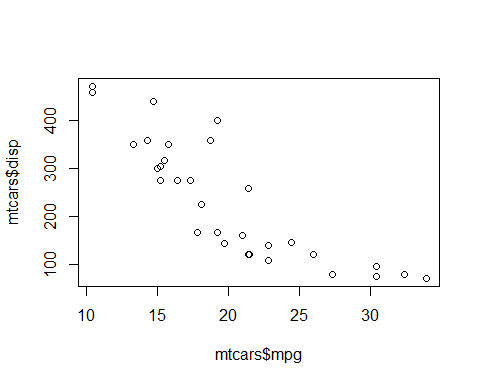
rataan\_simbaku.1 <- cbind(rataan.mpg, simbaku.mpg)  
rataan\_simbaku.2 <- merge(rataan.mpg, simbaku.mpg)

### 7.2. Visualisasi

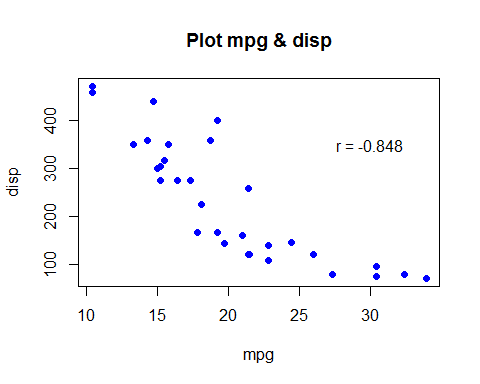
data(mtcars)  
head(mtcars)

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Mazda RX4 21.0 6 160 110 3.90 2.620 16.46 0 1 4 4  
## Mazda RX4 Wag 21.0 6 160 110 3.90 2.875 17.02 0 1 4 4  
## Datsun 710 22.8 4 108 93 3.85 2.320 18.61 1 1 4 1  
## Hornet 4 Drive 21.4 6 258 110 3.08 3.215 19.44 1 0 3 1  
## Hornet Sportabout 18.7 8 360 175 3.15 3.440 17.02 0 0 3 2  
## Valiant 18.1 6 225 105 2.76 3.460 20.22 1 0 3 1

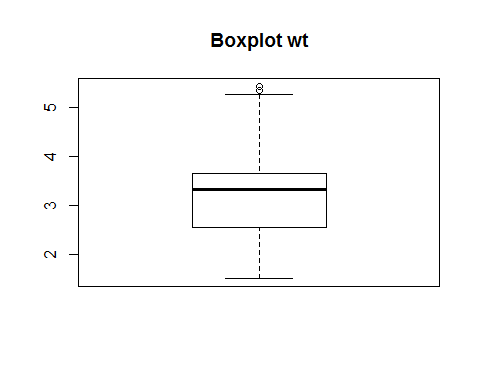
plot(mtcars$mpg, mtcars$disp)



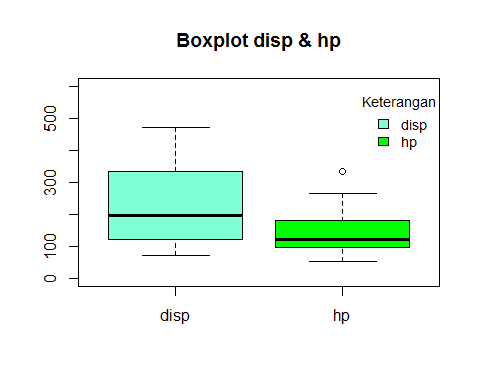
plot(mtcars$mpg, mtcars$disp,  
 xlab = "mpg", ylab = "disp",  
 main = "Plot mpg & disp",  
 pch = 16,  
 col = "blue")  
text(x = 30, y = 350,  
 paste("r =", round(cor(mtcars$mpg, mtcars$disp),3)))



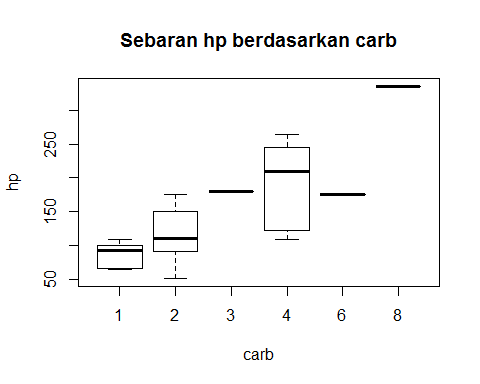
boxplot(mtcars$wt, main = "Boxplot wt")



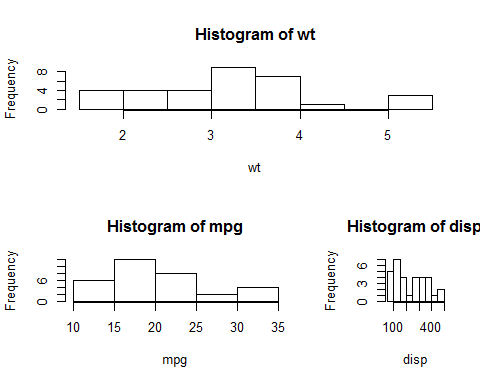
boxplot(mtcars[,c("disp","hp")],  
 main = "Boxplot disp & hp",  
 col = c("aquamarine", "green"),  
 ylim = c(0,600))  
legend(x = 2.1, y = 590,  
 legend = c("disp","hp"),  
 fill = c("aquamarine", "green"),  
 border = "black",  
 box.lty = 0,  
 title = "Keterangan",  
 cex = 0.9)



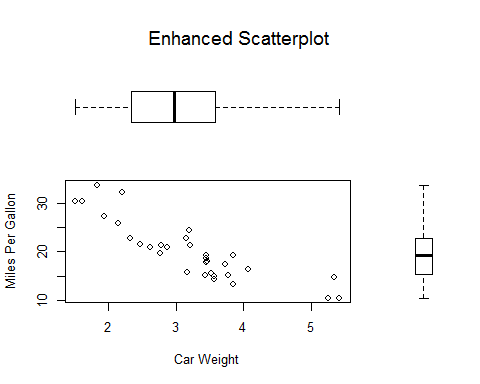
boxplot(hp~carb, data = mtcars,  
 xlab = "carb",  
 ylab = "hp",  
 main = "Sebaran hp berdasarkan carb")



# membuat beberapa plot dalam satu bingkai  
attach(mtcars) # menguraikan variabel  
layout(matrix(c(1,1,2,3), 2, 2, byrow = TRUE),   
 widths=c(2,1))  
hist(wt)  
hist(mpg)  
hist(disp)



detach(mtcars) # kebalikan dari attach  
  
par(fig=c(0,0.8,0,0.7))  
plot(mtcars$wt, mtcars$mpg, xlab="Car Weight",  
 ylab="Miles Per Gallon")  
par(fig=c(0,0.8,0.4,1),new=TRUE)  
boxplot(mtcars$mpg, horizontal = TRUE, axes=FALSE)  
par(fig=c(0.7,1,0,0.7),new=TRUE)  
boxplot(mtcars$mpg, axes=FALSE)  
mtext("Enhanced Scatterplot", side=3, outer=TRUE, line=-3, cex = 1.2)



layout(1) # mengembalikan pengaturan default jumlah baris dan kolom layout plot

## 8. Pustaka (Packages) R

### 8.1. Menginstall Pustaka

Ada banyak pustaka yang dapat diunduh secara gratis di CRAN atau Github. Hanya saja, di sini kita akan mencoba 3 pustaka.

install.packages("psych")

## Installing package into 'C:/Users/aa/Documents/R/win-library/3.5'  
## (as 'lib' is unspecified)

## Error in contrib.url(repos, "source"): trying to use CRAN without setting a mirror

install.packages("summarytools")

## Installing package into 'C:/Users/aa/Documents/R/win-library/3.5'  
## (as 'lib' is unspecified)

## Error in contrib.url(repos, "source"): trying to use CRAN without setting a mirror

install.packages("ggplot2")

## Installing package into 'C:/Users/aa/Documents/R/win-library/3.5'  
## (as 'lib' is unspecified)

## Error in contrib.url(repos, "source"): trying to use CRAN without setting a mirror

### 8.2. Menggunakan Pustaka

Setelah pustaka diinstall, maka dapat dipanggil. #### 8.2.1. Memanggil Pustaka

library(psych)  
library(summarytools)  
library(ggplot2)

##   
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':  
##   
## %+%, alpha

Data yang digunakan adalah data internal yaitu data iris

# Memanggil data iris  
data(iris)  
str(iris) # struktur data iris

## 'data.frame': 150 obs. of 5 variables:  
## $ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...  
## $ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...  
## $ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...  
## $ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...  
## $ Species : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...

head(iris)

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 1 5.1 3.5 1.4 0.2 setosa  
## 2 4.9 3.0 1.4 0.2 setosa  
## 3 4.7 3.2 1.3 0.2 setosa  
## 4 4.6 3.1 1.5 0.2 setosa  
## 5 5.0 3.6 1.4 0.2 setosa  
## 6 5.4 3.9 1.7 0.4 setosa

#### 8.2.2. Menggunakan Pustaka

# pustaka psych  
citation("psych") # cara mengutip pustaka psych

##   
## To cite the psych package in publications use:  
##   
## Revelle, W. (2018) psych: Procedures for Personality and  
## Psychological Research, Northwestern University, Evanston,  
## Illinois, USA, https://CRAN.R-project.org/package=psych Version  
## = 1.8.12.  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Manual{,  
## title = {psych: Procedures for Psychological, Psychometric, and Personality Research},  
## author = {William Revelle},  
## organization = { Northwestern University},  
## address = { Evanston, Illinois},  
## year = {2018},  
## note = {R package version 1.8.12},  
## url = {https://CRAN.R-project.org/package=psych},  
## }

describe(iris)

## vars n mean sd median trimmed mad min max range skew  
## Sepal.Length 1 150 5.84 0.83 5.80 5.81 1.04 4.3 7.9 3.6 0.31  
## Sepal.Width 2 150 3.06 0.44 3.00 3.04 0.44 2.0 4.4 2.4 0.31  
## Petal.Length 3 150 3.76 1.77 4.35 3.76 1.85 1.0 6.9 5.9 -0.27  
## Petal.Width 4 150 1.20 0.76 1.30 1.18 1.04 0.1 2.5 2.4 -0.10  
## Species\* 5 150 2.00 0.82 2.00 2.00 1.48 1.0 3.0 2.0 0.00  
## kurtosis se  
## Sepal.Length -0.61 0.07  
## Sepal.Width 0.14 0.04  
## Petal.Length -1.42 0.14  
## Petal.Width -1.36 0.06  
## Species\* -1.52 0.07

describeBy(iris, group = iris$Species)

##   
## Descriptive statistics by group   
## group: setosa  
## vars n mean sd median trimmed mad min max range skew  
## Sepal.Length 1 50 5.01 0.35 5.0 5.00 0.30 4.3 5.8 1.5 0.11  
## Sepal.Width 2 50 3.43 0.38 3.4 3.42 0.37 2.3 4.4 2.1 0.04  
## Petal.Length 3 50 1.46 0.17 1.5 1.46 0.15 1.0 1.9 0.9 0.10  
## Petal.Width 4 50 0.25 0.11 0.2 0.24 0.00 0.1 0.6 0.5 1.18  
## Species\* 5 50 1.00 0.00 1.0 1.00 0.00 1.0 1.0 0.0 NaN  
## kurtosis se  
## Sepal.Length -0.45 0.05  
## Sepal.Width 0.60 0.05  
## Petal.Length 0.65 0.02  
## Petal.Width 1.26 0.01  
## Species\* NaN 0.00  
## --------------------------------------------------------   
## group: versicolor  
## vars n mean sd median trimmed mad min max range skew  
## Sepal.Length 1 50 5.94 0.52 5.90 5.94 0.52 4.9 7.0 2.1 0.10  
## Sepal.Width 2 50 2.77 0.31 2.80 2.78 0.30 2.0 3.4 1.4 -0.34  
## Petal.Length 3 50 4.26 0.47 4.35 4.29 0.52 3.0 5.1 2.1 -0.57  
## Petal.Width 4 50 1.33 0.20 1.30 1.32 0.22 1.0 1.8 0.8 -0.03  
## Species\* 5 50 2.00 0.00 2.00 2.00 0.00 2.0 2.0 0.0 NaN  
## kurtosis se  
## Sepal.Length -0.69 0.07  
## Sepal.Width -0.55 0.04  
## Petal.Length -0.19 0.07  
## Petal.Width -0.59 0.03  
## Species\* NaN 0.00  
## --------------------------------------------------------   
## group: virginica  
## vars n mean sd median trimmed mad min max range skew  
## Sepal.Length 1 50 6.59 0.64 6.50 6.57 0.59 4.9 7.9 3.0 0.11  
## Sepal.Width 2 50 2.97 0.32 3.00 2.96 0.30 2.2 3.8 1.6 0.34  
## Petal.Length 3 50 5.55 0.55 5.55 5.51 0.67 4.5 6.9 2.4 0.52  
## Petal.Width 4 50 2.03 0.27 2.00 2.03 0.30 1.4 2.5 1.1 -0.12  
## Species\* 5 50 3.00 0.00 3.00 3.00 0.00 3.0 3.0 0.0 NaN  
## kurtosis se  
## Sepal.Length -0.20 0.09  
## Sepal.Width 0.38 0.05  
## Petal.Length -0.37 0.08  
## Petal.Width -0.75 0.04  
## Species\* NaN 0.00

# pustaka summarytools  
citation("summarytools")

##   
## To cite package 'summarytools' in publications use:  
##   
## Dominic Comtois (2019). summarytools: Tools to Quickly and  
## Neatly Summarize Data. R package version 0.9.3.  
## https://CRAN.R-project.org/package=summarytools  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Manual{,  
## title = {summarytools: Tools to Quickly and Neatly Summarize Data},  
## author = {Dominic Comtois},  
## year = {2019},  
## note = {R package version 0.9.3},  
## url = {https://CRAN.R-project.org/package=summarytools},  
## }

view(dfSummary(iris))

## Switching method to 'browser'

## Output file written: C:\Users\aa\AppData\Local\Temp\Rtmpmsypuc\file45502c457884.html

# pustaka ggplot2  
library(ggplot2)  
head(iris)

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 1 5.1 3.5 1.4 0.2 setosa  
## 2 4.9 3.0 1.4 0.2 setosa  
## 3 4.7 3.2 1.3 0.2 setosa  
## 4 4.6 3.1 1.5 0.2 setosa  
## 5 5.0 3.6 1.4 0.2 setosa  
## 6 5.4 3.9 1.7 0.4 setosa

ggplot(iris, aes(x = Species, y = Sepal.Length, fill = Species)) +   
 geom\_boxplot() +  
 labs(title = "Sepal Length",  
 subtitle = "Berdasarkan Species")

