Tutorial eksplorasi data menggunakan R

Sandy Hardian Susanto Herho
sandy.herho@igdore.org

Dasapta Erwin Irawan
r-win@office.itb.ac.id

Pendahuluan

Apa itu R?

R merupakan bahasa pemrograman dan lingkungan perangkat lunak yang digunakan untuk analisis statistik, pemodelan data, visuliasi grafik, dan pelaporan data. R bersifat sumber terbuka dan gratis diunduh oleh siapapun. Tutorial ini ditujukan untuk membantu pemula untuk memulai eksplorasi data dengan menggunakan R.

Instalasi

1. Kunjungi https://docs.conda.io/projects/conda/en/latest/user-guide/install/ untuk instalasi Miniconda versi 3.
2. Ikuti prosedur instalasi (jangan lupa atur PATH -nya).
3. Ikuti prosedur instalasi git pada situs: https://git-scm.com/book/en/v2/Getting-Started-Installing-Git.
4. Buku *Command Line Interface* (CLI), jalankan perintah:

```
git clone git@github.com:sandyherho/eksploR.git
```

5. Lakukan instalasi lingkungan virtual conda dengan menjalankan perintah di folder hasil *cloning* tersebut:

```
conda env create -f environment.yml
```

6. Aktifkan lingkungan virtual dengan perintah:

```
conda activate r-env
```

7. Untuk mengaktifkan Jupyter Notebook, jalankan perintah:

```
jupyter notebook
```

8. Jika sudah terbuka di *browser* masing - masing sesi R interaktif dapat dimulai.
9. Untuk mengakhiri sesi tekan tombol `<CTRL> + C` di CLI, dan jalankan perintah sebagai berikut untuk menonaktifkan lingkungan virtual conda:

```
conda deactivate
```

Dasar - dasar pemrograman

Aritmatika

```
1 + 2
```

3

```
5 - 3
```

2

```
5.25 + 3.95
```

9.2

```
3 * 2
```

6

```
15 / 2
```

7.5

```
2 ^ 3
```

8

```
# operator modulus

15/2
```

7.5

```
15 %% 2
```

1

```
# urutan operasi
15 * 20 + 50 / 2
```

325

```
15 * 20 + (50 / 2)
```

325

```
15 * (20 + 50) / 2
```

525

Variabel

```
# komentar
v <- 10
```

```
print(v)
```

```
[1] 10
```

```
v
```

10

```
uang <- 100000
uang
```

1e+05

```
tiket.bioskop <- 2e5 # paling sering digunakan di
komunitas R
tiket.bioskop
```

2e+05

```
tiketBioskop <- 2.05e5
tiketBioskop
```

205000

```
tiket_bioskop <-2.1e5
tiket_bioskop
```

210000

```
popcorn <- 5e4
popcorn
```

50000

```
tiket.bioskop <- tiket.bioskop + popcorn + 2e4
print(tiket.bioskop)
```

```
[1] 270000
```

Tipe - tipe data

Numerik

```
b <- 2
b
```

2

```
d <- 3.5
d
```

3.5

```
class(d)
```

'numeric'

```
class(b)
```

'numeric'

Integer dan float di R dianggap sebagai tipe data numerik

Logical

```
TRUE
```

TRUE

```
FALSE
```

FALSE

```
T
```

TRUE

```
F
```

FALSE

```
a <- TRUE
a
```

TRUE

```
class(a)
```

'logical'

Character

```
"Halo"
```

'Halo'

```
'Halo'
```

'Halo'

```
txt <- "Hello world!"
print(txt)
```

```
[1] "Hello world!"
```

```
class(txt)
```

'character'

Vektor

```
num <- c(1,2,3,4,5)
```

```
class(num)
```

'numeric'

```
txt <- c('a', 'b', 'c')
txt
```

1. 'a'
2. 'b'
3. 'c'

```
class(txt)
```

'character'

```
l <- c(T,F)
l
```

1. TRUE
2. FALSE

```
class(l)
```

'logical'

```
v1 <- c(TRUE, 10, 20)
v1
```

1. 1
2. 10
3. 20

```
class(v1) # logical dikonversi menjadi numeric
```

'numeric'

```
v2 <- c('a', 'b', 20)
v2
```

1. 'a'
2. 'b'
3. '20'

```
class(v2) # numeric dikonversi menjadi character
```

'character'

```
v3 <- c(TRUE, 128, 'Meteorologi')
v3
```

1. 'TRUE'
2. '128'
3. 'Meteorologi'

```
class(v3) # logical & numeric dikonversi menjadi character
```

'character'

```
# names : metode penamaaan vektor (metode 1)
hari <- c(1,2,3,4,5,6,7)
hari
```

1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7

```
names(hari) <- c('sen', 'sel', 'rab', 'kam', 'jum', 'sab',
'min')
```

```
hari
```

sen
: 1

sel
: 2

rab
: 3

kam
: 4

jum
: 5

sab
: 6

min
: 7

```
# names : metode penamaaan vektor (metode 2)
b <-  c('sen', 'sel', 'rab', 'kam', 'jum', 'sab', 'min')
names(hari) <- b
hari
```

sen
: 1

sel
: 2

rab
: 3

kam
: 4

jum
: 5

sab
: 6

min
: 7

```
hari['sen']
```

sen: 1

```
hari[1]
```

sen: 1

Operasi - operasi vektor

```
v1 <- c(1,2,3,4,5)
v2 <- c(6,7,8,9,10)
```

```
v1 + v2 # element-by-element
```

1. 7
2. 9
3. 11
4. 13
5. 15

```
v1 - v2
```

1. -5
2. -5
3. -5
4. -5
5. -5

```
v1 * v2
```

1. 6
2. 14
3. 24
4. 36
5. 50

```
v2 / v2
```

1. 1
2. 1
3. 1
4. 1
5. 1

```
sum(v1) # jumlah seluruh v1
```

15

```
mean(v1) # rata2 v1
```

3

```
sd(v1) # std v1
```

1.58113883008419

```
max(v2)
```

10

```
min(v2)
```

6

```
prod(v1) # mengalikan seluruh elemen di vektor
```

120

```
prod(v2)
```

30240

```
b <- sum(v1)
print(b)
```

```
[1] 15
```

Operator - operator perbandingan

```
4 > 5
```

FALSE

```
7 > 4
```

TRUE

```
10 >= 5
```

TRUE

```
7 <= 5
```

FALSE

```
8 == 8
```

TRUE

```
7 != 15
```

TRUE

```
7 != 7
```

FALSE

```
# Operator perbandingan pada vektor
v <- c(1,2,3,4,5)
v < 2
```

1. TRUE
2. FALSE
3. FALSE
4. FALSE
5. FALSE

```
v == 3
```

1. FALSE
2. FALSE
3. TRUE
4. FALSE
5. FALSE

```
v2 <- c(10,20,30,40,50)
v2
```

1. 10
2. 20
3. 30
4. 40
5. 50

```
v < v2
```

1. TRUE
2. TRUE
3. TRUE

4. TRUE
5. TRUE

Pengindeksan dan pemotongan vektor

```
v1 <- c(10,20,30,40,50)
v2 <- c('a', 'b', 'c', 'd', 'e')
```

```
v1[2] # pengindeksan dimulai dari 1
```

20

```
v1[5]
```

50

```
v2[c(3,4,5)]
```

1. 'c'
2. 'd'
3. 'e'

```
v3 <- c(1,2,3,4,5,6,7,8,9,10)
v3[7:10]
```

1. 7
2. 8
3. 9
4. 10

```
v3[3:5]
```

1. 3
2. 4

3. 5

```
b <- c(1,2,3,4,5,6)
names(b) <- c('I', 'G', 'D', 'O', 'R', 'E')
b
```

I
: 1

G
: 2

D
: 3

O
: 4

R
: 5

E
: 6

```
b[2]
```

G: 2

```
b['G']
```

G: 2

```
b[c(2,3)]
```

G
: 2

D
: 3

```
b[c('G', 'D')]
```

G

2

D

3

```
# Operator perbandingan
b
```

I

1

G

2

D

3

O

4

R

5

E

6

```
b[b > 3]
```

O

4

R

5

E

6

```
e <-  b > 3
```

```
b[e]
```

O

4

R

5

E

6

Matriks

Mendefinisikan matriks

```
b <- 1:10
b
```

1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10

```
matrix(b) # 10 x 1
```

1
2
3
4
5
6
7
8
9
10

```
matrix(b, nrow=2, ncol= 5) # by column
```

1	3	5	7	9
2	4	6	8	10

```
matrix(b, nrow=2, ncol= 5, byrow = T) # by row
```

1	2	3	4	5
6	7	8	9	10

```
matrix(1:12, nrow = 4, byrow=TRUE) # 4 x 3 by row
```

1	2	3
4	5	6
7	8	9
10	11	12

```
# Mendefinisikan matriks dari vektor
fb <- c(250,255,260,263,265) # bayangkan sebagai harga
saham
ms <- c(455,460,465,479, 470)
```

```
saham <- c(fb, ms)
saham
```

1. 250
2. 255
3. 260
4. 263
5. 265
6. 455
7. 460
8. 465
9. 479
10. 470

```
matriks.saham <- matrix(saham, nrow=2, byrow=T)
matriks.saham
```

250	255	260	263	265
455	460	465	479	470

```
# Menamakan baris dan kolom
```

```
hari <- c('sen', 'sel', 'rab', 'kam', 'jum')
perusahaan <- c('fb', 'ms')
```

```
colnames(matriks.saham) <- hari
rownames(matriks.saham) <- perusahaan
```

```
matriks.saham
```

	SEN	SEL	RAB	KAM	JUM
fb	250	255	260	263	265
ms	455	460	465	479	470

Aritmatika matriks

```
mat <- matrix(1:25, nrow=5, byrow=T)
mat
```

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

```
mat * mat # element-by-element
```

1	4	9	16	25

36	49	64	81	100
121	144	169	196	225
256	289	324	361	400
441	484	529	576	625

```
mat / mat
```

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

```
mat^2
```

1	4	9	16	25
36	49	64	81	100
121	144	169	196	225
256	289	324	361	400
441	484	529	576	625

```
1 / mat
```

1.00000000	0.50000000	0.33333333	0.25000000	0.20000000
0.16666667	0.14285714	0.12500000	0.11111111	0.10000000
0.09090909	0.08333333	0.07692308	0.07142857	0.06666667
0.06250000	0.05882353	0.05555556	0.05263158	0.05000000
0.04761905	0.04545455	0.04347826	0.04166667	0.04000000

```
# Operator perbandingan di matriks
```

```
mat > 10
```

FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE
TRUE	TRUE	TRUE	TRUE	TRUE
TRUE	TRUE	TRUE	TRUE	TRUE
TRUE	TRUE	TRUE	TRUE	TRUE

```
mat[mat > 10]
```

1. 11
2. 16
3. 21
4. 12
5. 17
6. 22
7. 13
8. 18
9. 23
10. 14
11. 19
12. 24
13. 15
14. 20
15. 25

```
mat
```

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

```
# Perkalian matriks
mat %*% mat
```

215	230	245	260	275
490	530	570	610	650
765	830	895	960	1025
1040	1130	1220	1310	1400
1315	1430	1545	1660	1775

Operasi - operasi di matriks

```
# Mendefinisikan matriks dari vektor
fb <- c(250,255,260,263,265)
ms <- c(455,460,465,479, 470)
saham <- c(fb, ms)
matriks.saham <- matrix(saham, nrow=2, byrow=T)
colnames(matriks.saham) <- c('sen', 'sel', 'rab', 'kam',
'jum')
rownames(matriks.saham) <-  c('fb', 'ms')
matriks.saham
```

	SEN	SEL	RAB	KAM	JUM
fb	250	255	260	263	265
ms	455	460	465	479	470

```
colSums(matriks.saham) # penjumlahan pada kolom
```

```
sen
705
sel
715
rab
725
kam
742
jum
735
```

```
rowSums(matriks.saham)
```

fb
: 1293

ms
: 2329

```
rowMeans(matriks.saham)
```

fb
: 258.6

ms
: 465.8

```
colMeans(matriks.saham)
```

sen
: 352.5

sel
: 357.5

rab
: 362.5

kam
: 371

jum
: 367.5

```
# Menambahkan kolom dan baris ke matriks
```

```
google <- c(175,180,185,195,190)
saham.int <- rbind(matriks.saham, google)
saham.int
```

	SEN	SEL	RAB	KAM	JUM
fb	250	255	260	263	265
ms	455	460	465	479	470
google	175	180	185	195	190

```
# Menambahkan kolom ke matriks
rata2 <- rowMeans(saham.int)
rata2
```

fb
: 258.6

ms
: 465.8

google
: 185

```
saham.int <- cbind(saham.int, rata2)
saham.int
```

	SEN	SEL	RAB	KAM	JUM	RATA2
fb	250	255	260	263	265	258.6
ms	455	460	465	479	470	465.8
google	175	180	185	195	190	185.0

Seleksi dan pengindeksan matriks

```
v <- c(10,20,30,40,50)
v
```

1. 10
2. 20
3. 30
4. 40
5. 50

```
v[3]
```

```
mat <- matrix(1:25, nrow=5, byrow=T)
mat
```

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

```
mat[1,] # baris 1, seluruh kolom
```

1. 1
2. 2
3. 3
4. 4
5. 5

```
mat[2,3]
```

8

```
mat[3,4]
```

14

```
mat[,3]
```

1. 3
2. 8
3. 13
4. 18
5. 23

```
mat
```

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

```
mat[,5]
```

1. 5
2. 10
3. 15
4. 20
5. 25

```
mat[1:3,] # baris 1 sampai 3
```

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15

```
mat[,4:5] # kolom 4 hingga 5
```

4	5
9	10
14	15
19	20
24	25

```
mat[1:2, 1:3] # baris 1 sampai 2, kolom 1 sampai 3
```

1	2	3
6	7	8

```
mat
```

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

```
mat[3:5, 3:5]
```

13	14	15
18	19	20
23	24	25

Fungsi `factor()`

```
vek.warna <- c('merah', 'hijau', 'biru', 'merah', 'merah',
'hijau', 'biru')
```

```
vek.warna
```

1. 'merah'
2. 'hijau'
3. 'biru'
4. 'merah'
5. 'merah'
6. 'hijau'
7. 'biru'

```
fact.warna <- factor(vek.warna, ordered=T,
levels=c('merah', 'hijau', 'biru'))
fact.warna
```

1. merah
2. hijau
3. biru
4. merah
5. merah
6. hijau
7. biru

- **Levels**:

```
summary(fact.warna)
```

merah
: 3

hijau
: 2

biru
: 2

```
summary(vek.warna)
```

```
   Length     Class      Mode
        7 character character
```

Data Frame

Pengenalan

Kita banyak menggunakan data frame di dalam kegiatan analisis data, karena matriks hanya mampu menampung tipe data yang seragam.

```
state.x77 # built-in df
```

	POPULATION	INCOME	ILLITERACY	LIFE EXP	MURDER	HS GRAD	FROST	AREA
Alabama	3615	3624	2.1	69.05	15.1	41.3	20	50708
Alaska	365	6315	1.5	69.31	11.3	66.7	152	566432
Arizona	2212	4530	1.8	70.55	7.8	58.1	15	113417
Arkansas	2110	3378	1.9	70.66	10.1	39.9	65	51945
California	21198	5114	1.1	71.71	10.3	62.6	20	156361
Colorado	2541	4884	0.7	72.06	6.8	63.9	166	103766
Connecticut	3100	5348	1.1	72.48	3.1	56.0	139	4862
Delaware	579	4809	0.9	70.06	6.2	54.6	103	1982
Florida	8277	4815	1.3	70.66	10.7	52.6	11	54090
Georgia	4931	4091	2.0	68.54	13.9	40.6	60	58073
Hawaii	868	4963	1.9	73.60	6.2	61.9	0	6425
Idaho	813	4119	0.6	71.87	5.3	59.5	126	82677
Illinois	11197	5107	0.9	70.14	10.3	52.6	127	55748
Indiana	5313	4458	0.7	70.88	7.1	52.9	122	36097
Iowa	2861	4628	0.5	72.56	2.3	59.0	140	55941
Kansas	2280	4669	0.6	72.58	4.5	59.9	114	81787
Kentucky	3387	3712	1.6	70.10	10.6	38.5	95	39650
Louisiana	3806	3545	2.8	68.76	13.2	42.2	12	44930
Maine	1058	3694	0.7	70.39	2.7	54.7	161	30920
Maryland	4122	5299	0.9	70.22	8.5	52.3	101	9891
Massachusetts	5814	4755	1.1	71.83	3.3	58.5	103	7826
Michigan	9111	4751	0.9	70.63	11.1	52.8	125	56817
Minnesota	3921	4675	0.6	72.96	2.3	57.6	160	79289
Mississippi	2341	3098	2.4	68.09	12.5	41.0	50	47296
Missouri	4767	4254	0.8	70.69	9.3	48.8	108	68995
Montana	746	4347	0.6	70.56	5.0	59.2	155	145587
Nebraska	1544	4508	0.6	72.60	2.9	59.3	139	76483
Nevada	590	5149	0.5	69.03	11.5	65.2	188	109889

	POPULATION	INCOME	ILLITERACY	LIFE EXP	MURDER	HS GRAD	FROST	AREA
New Hampshire	812	4281	0.7	71.23	3.3	57.6	174	9027
New Jersey	7333	5237	1.1	70.93	5.2	52.5	115	7521
New Mexico	1144	3601	2.2	70.32	9.7	55.2	120	121412
New York	18076	4903	1.4	70.55	10.9	52.7	82	47831
North Carolina	5441	3875	1.8	69.21	11.1	38.5	80	48798
North Dakota	637	5087	0.8	72.78	1.4	50.3	186	69273
Ohio	10735	4561	0.8	70.82	7.4	53.2	124	40975
Oklahoma	2715	3983	1.1	71.42	6.4	51.6	82	68782
Oregon	2284	4660	0.6	72.13	4.2	60.0	44	96184
Pennsylvania	11860	4449	1.0	70.43	6.1	50.2	126	44966
Rhode Island	931	4558	1.3	71.90	2.4	46.4	127	1049
South Carolina	2816	3635	2.3	67.96	11.6	37.8	65	30225
South Dakota	681	4167	0.5	72.08	1.7	53.3	172	75955
Tennessee	4173	3821	1.7	70.11	11.0	41.8	70	41328
Texas	12237	4188	2.2	70.90	12.2	47.4	35	262134
Utah	1203	4022	0.6	72.90	4.5	67.3	137	82096
Vermont	472	3907	0.6	71.64	5.5	57.1	168	9267
Virginia	4981	4701	1.4	70.08	9.5	47.8	85	39780
Washington	3559	4864	0.6	71.72	4.3	63.5	32	66570
West Virginia	1799	3617	1.4	69.48	6.7	41.6	100	24070
Wisconsin	4589	4468	0.7	72.48	3.0	54.5	149	54464
Wyoming	376	4566	0.6	70.29	6.9	62.9	173	97203

```
USPersonalExpenditure
```

	1940	1945	1950	1955	1960
Food and Tobacco	22.200	44.500	59.60	73.2	86.80
Household Operation	10.500	15.500	29.00	36.5	46.20
Medical and Health	3.530	5.760	9.71	14.0	21.10
Personal Care	1.040	1.980	2.45	3.4	5.40
Private Education	0.341	0.974	1.80	2.6	3.64

```
# Mengetahui daftar built-in df
# data()
```

```
head(state.x77) # 6 baris pertama
```

	POPULATION	INCOME	ILLITERACY	LIFE EXP	MURDER	HS GRAD	FROST	AREA
Alabama	3615	3624	2.1	69.05	15.1	41.3	20	50708
Alaska	365	6315	1.5	69.31	11.3	66.7	152	566432
Arizona	2212	4530	1.8	70.55	7.8	58.1	15	113417
Arkansas	2110	3378	1.9	70.66	10.1	39.9	65	51945
California	21198	5114	1.1	71.71	10.3	62.6	20	156361
Colorado	2541	4884	0.7	72.06	6.8	63.9	166	103766

```
tail(state.x77) # 6 baris terakhir
```

	POPULATION	INCOME	ILLITERACY	LIFE EXP	MURDER	HS GRAD	FROST	AREA
Vermont	472	3907	0.6	71.64	5.5	57.1	168	9267
Virginia	4981	4701	1.4	70.08	9.5	47.8	85	39780
Washington	3559	4864	0.6	71.72	4.3	63.5	32	66570
West Virginia	1799	3617	1.4	69.48	6.7	41.6	100	24070
Wisconsin	4589	4468	0.7	72.48	3.0	54.5	149	54464
Wyoming	376	4566	0.6	70.29	6.9	62.9	173	97203

```
str(state.x77) # struktur dari df
```

```
 num [1:50, 1:8] 3615 365 2212 2110 21198 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:50] "Alabama" "Alaska" "Arizona" "Arkansas"
...
  ..$ : chr [1:8] "Population" "Income" "Illiteracy" "Life
Exp" ...
```

```
summary(state.x77) # sari statistik dari df berdasarkan
kolom
```

```
   Population        Income       Illiteracy       Life
Exp
 Min.   :  365   Min.   :3098   Min.   :0.500   Min.
:67.96
 1st Qu.: 1080   1st Qu.:3993   1st Qu.:0.625   1st
Qu.:70.12
 Median : 2838   Median :4519   Median :0.950   Median
:70.67
 Mean   : 4246   Mean   :4436   Mean   :1.170   Mean
:70.88
 3rd Qu.: 4968   3rd Qu.:4814   3rd Qu.:1.575   3rd
Qu.:71.89
 Max.   :21198   Max.   :6315   Max.   :2.800   Max.
:73.60
     Murder          HS Grad          Frost
Area
 Min.   : 1.400   Min.   :37.80   Min.   :  0.00   Min.
:  1049
 1st Qu.: 4.350   1st Qu.:48.05   1st Qu.: 66.25   1st
Qu.: 36985
 Median : 6.850   Median :53.25   Median :114.50   Median
: 54277
 Mean   : 7.378   Mean   :53.11   Mean   :104.46   Mean
: 70736
 3rd Qu.:10.675   3rd Qu.:59.15   3rd Qu.:139.75   3rd
Qu.: 81162
 Max.   :15.100   Max.   :67.30   Max.   :188.00   Max.
:566432
```

```
# mendefinisikan df
nama <- c('Agus', 'Sugio', 'Bayu', 'Atmo', 'Roy')
umur <- c(42,35,37,28,27)
kawin <- c(F,F,T,F,T)
```

```
data.frame(nama, umur, kawin)
```

NAMA	UMUR	KAWIN
Agus	42	FALSE
Sugio	35	FALSE
Bayu	37	TRUE
Atmo	28	FALSE
Roy	27	TRUE

```
df <- data.frame(nama, umur, kawin)
df
```

NAMA	UMUR	KAWIN
Agus	42	FALSE
Sugio	35	FALSE
Bayu	37	TRUE
Atmo	28	FALSE
Roy	27	TRUE

```
str(df)
```

```
'data.frame':   5 obs. of  3 variables:
 $ nama : Factor w/ 5 levels "Agus","Atmo",..: 1 5 3 2 4
 $ umur : num  42 35 37 28 27
 $ kawin: logi  FALSE FALSE TRUE FALSE TRUE
```

```
summary(df)
```

```
    nama        umur        kawin
 Agus :1   Min.   :27.0   Mode :logical
 Atmo :1   1st Qu.:28.0   FALSE:3
 Bayu :1   Median :35.0   TRUE :2
 Roy  :1   Mean   :33.8
 Sugio:1   3rd Qu.:37.0
           Max.   :42.0
```

Seleksi dan pengindeksan Data Frame

```
df
```

NAMA	UMUR	KAWIN
Agus	42	FALSE
Sugio	35	FALSE
Bayu	37	TRUE

NAMA	UMUR	KAWIN
Atmo	28	FALSE
Roy	27	TRUE

```
df[3, ] # ambil baris ketiga
```

	NAMA	UMUR	KAWIN
3	Bayu	37	TRUE

```
df[,1]
```

1. Agus
2. Sugio
3. Bayu
4. Atmo
5. Roy

▶ **Levels**:

```
df[,'nama']
```

1. Agus
2. Sugio
3. Bayu
4. Atmo
5. Roy

▶ **Levels**:

```
df[1:4, c('nama', 'umur')]
```

NAMA	UMUR
Agus	42
Sugio	35
Bayu	37
Atmo	28

```
df$umur
```

1. 42
2. 35
3. 37
4. 28
5. 27

```
df[,'umur']
```

1. 42
2. 35
3. 37
4. 28
5. 27

```
# fungsi subset
df
```

NAMA	UMUR	KAWIN
Agus	42	FALSE
Sugio	35	FALSE
Bayu	37	TRUE
Atmo	28	FALSE
Roy	27	TRUE

```
subset(df,subset = kawin == T)
```

	NAMA	UMUR	KAWIN
3	Bayu	37	TRUE
5	Roy	27	TRUE

```
subset(df, subset = umur > 30)
```

NAMA	UMUR	KAWIN
Agus	42	FALSE
Sugio	35	FALSE
Bayu	37	TRUE

```
# Mengurutkan dataframe
urut.umur <- order(df['umur'])
urut.umur
```

1. 5
2. 4
3. 2
4. 3
5. 1

```
df[urut.umur,]
```

	NAMA	UMUR	KAWIN
5	Roy	27	TRUE
4	Atmo	28	FALSE
2	Sugio	35	FALSE
3	Bayu	37	TRUE
1	Agus	42	FALSE

```
umur.terbalik <- order(-df['umur'])
umur.terbalik
```

1. 1
2. 3
3. 2

4. 4
5. 5

```
df[umur.terbalik,]
```

	NAMA	UMUR	KAWIN
1	Agus	42	FALSE
3	Bayu	37	TRUE
2	Sugio	35	FALSE
4	Atmo	28	FALSE
5	Roy	27	TRUE

```
urut.umur <- order(df$umur)
df[urut.umur, ]
```

	NAMA	UMUR	KAWIN
5	Roy	27	TRUE
4	Atmo	28	FALSE
2	Sugio	35	FALSE
3	Bayu	37	TRUE
1	Agus	42	FALSE

Operasi - operasi data frame

Mendefinisikan data frame

```
c1 <- 1:10
c2 <- letters[1:10]
print(c1)
print(c2)
```

```
 [1]  1  2  3  4  5  6  7  8  9 10
 [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j"
```

```
df <- data.frame(c1,c2)
df
```

C1	C2
1	a
2	b
3	c
4	d
5	e
6	f
7	g
8	h
9	i
10	j

```
df <- data.frame(kolom1 = c1, kolom2 = c2) # nama kolom
bisa kita ubah sesuka kita!
df
```

KOLOM1	KOLOM2
1	a
2	b
3	c
4	d
5	e
6	f
7	g
8	h
9	i
10	j

Mendapatkan info tentang data frame

```
nrow(df) # jumlah baris
```

```
ncol(df) # jumlah kolom
```

2

```
colnames(df)  # nama kolom
```

1. 'kolom1'
2. 'kolom2'

```
rownames(df) # nama baris
```

1. '1'
2. '2'
3. '3'
4. '4'
5. '5'
6. '6'
7. '7'
8. '8'
9. '9'
10. '10'

```
str(df) # struktur data frame
```

```
'data.frame':   10 obs. of  2 variables:
 $ kolom1: int  1 2 3 4 5 6 7 8 9 10
 $ kolom2: Factor w/ 10 levels "a","b","c","d",..: 1 2 3 4
5 6 7 8 9 10
```

```
summary(df) # sari statistik
```

```
     kolom1          kolom2
 Min.   : 1.00   a      :1
 1st Qu.: 3.25   b      :1
 Median : 5.50   c      :1
 Mean   : 5.50   d      :1
 3rd Qu.: 7.75   e      :1
 Max.   :10.00   f      :1
                 (Other):4
```

Referensi sel

```
df
```

KOLOM1	KOLOM2
1	a
2	b
3	c
4	d
5	e
6	f
7	g
8	h
9	i
10	j

```
df[5,1]
```

5

```
df[[5,1]]
```

5

```
df[1,'kolom1']
```

1

```
df[5, 'kolom1']
```

5

```
df[[5,'kolom1']]
```

5

```
df[8,'kolom1'] <- -999 # mengubah nilai
df
```

KOLOM1	KOLOM2
1	a
2	b
3	c
4	d
5	e
6	f
7	g
-999	h
9	i
10	j

```
df[[8,'kolom1']] <- 8
df
```

KOLOM1	KOLOM2
1	a
2	b
3	c
4	d
5	e
6	f
7	g
8	h
9	i

KOLOM1	KOLOM2
10	j

Referensi baris dan kolom

```
df[2,]
```

	KOLOM1	KOLOM2
2	2	b

```
df[1:3,]
```

KOLOM1	KOLOM2
1	a
2	b
3	c

```
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

```
mtcars$mpg
```

1. 21
2. 21
3. 22.8
4. 21.4
5. 18.7
6. 18.1
7. 14.3
8. 24.4
9. 22.8
10. 19.2
11. 17.8
12. 16.4
13. 17.3
14. 15.2
15. 10.4
16. 10.4
17. 14.7
18. 32.4
19. 30.4
20. 33.9
21. 21.5
22. 15.5
23. 15.2
24. 13.3
25. 19.2
26. 27.3
27. 26
28. 30.4
29. 15.8
30. 19.7
31. 15
32. 21.4

```
mtcars[,1]
```

1. 21
2. 21
3. 22.8
4. 21.4
5. 18.7
6. 18.1
7. 14.3
8. 24.4
9. 22.8
10. 19.2
11. 17.8
12. 16.4
13. 17.3
14. 15.2
15. 10.4

16. 10.4
17. 14.7
18. 32.4
19. 30.4
20. 33.9
21. 21.5
22. 15.5
23. 15.2
24. 13.3
25. 19.2
26. 27.3
27. 26
28. 30.4
29. 15.8
30. 19.7
31. 15
32. 21.4

```
mtcars[,'mpg']
```

1. 21
2. 21
3. 22.8
4. 21.4
5. 18.7
6. 18.1
7. 14.3
8. 24.4
9. 22.8
10. 19.2
11. 17.8
12. 16.4
13. 17.3
14. 15.2
15. 10.4
16. 10.4
17. 14.7
18. 32.4
19. 30.4
20. 33.9
21. 21.5
22. 15.5
23. 15.2
24. 13.3
25. 19.2
26. 27.3
27. 26
28. 30.4
29. 15.8
30. 19.7

31. 15
32. 21.4

```
mtcars[['mpg']]
```

1. 21
2. 21
3. 22.8
4. 21.4
5. 18.7
6. 18.1
7. 14.3
8. 24.4
9. 22.8
10. 19.2
11. 17.8
12. 16.4
13. 17.3
14. 15.2
15. 10.4
16. 10.4
17. 14.7
18. 32.4
19. 30.4
20. 33.9
21. 21.5
22. 15.5
23. 15.2
24. 13.3
25. 19.2
26. 27.3
27. 26
28. 30.4
29. 15.8
30. 19.7
31. 15
32. 21.4

```
mtcars[1] # indeks lokasi kolom:1, cara ini mereferensi
kolom mpg sbg data frame
```

	MPG
Mazda RX4	21.0
Mazda RX4 Wag	21.0

	MPG
Datsun 710	22.8
Hornet 4 Drive	21.4
Hornet Sportabout	18.7
Valiant	18.1
Duster 360	14.3
Merc 240D	24.4
Merc 230	22.8
Merc 280	19.2
Merc 280C	17.8
Merc 450SE	16.4
Merc 450SL	17.3
Merc 450SLC	15.2
Cadillac Fleetwood	10.4
Lincoln Continental	10.4
Chrysler Imperial	14.7
Fiat 128	32.4
Honda Civic	30.4
Toyota Corolla	33.9
Toyota Corona	21.5
Dodge Challenger	15.5
AMC Javelin	15.2
Camaro Z28	13.3
Pontiac Firebird	19.2
Fiat X1-9	27.3
Porsche 914-2	26.0
Lotus Europa	30.4
Ford Pantera L	15.8
Ferrari Dino	19.7
Maserati Bora	15.0
Volvo 142E	21.4

```
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

	MPG	CYL	DISP	HP	DRAT	WT	QSEC	VS	AM	GEAR	CARB
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

```
mtcars[c('mpg', 'cyl')]
```

	MPG	CYL
Mazda RX4	21.0	6
Mazda RX4 Wag	21.0	6
Datsun 710	22.8	4
Hornet 4 Drive	21.4	6
Hornet Sportabout	18.7	8
Valiant	18.1	6
Duster 360	14.3	8
Merc 240D	24.4	4
Merc 230	22.8	4
Merc 280	19.2	6
Merc 280C	17.8	6
Merc 450SE	16.4	8
Merc 450SL	17.3	8
Merc 450SLC	15.2	8
Cadillac Fleetwood	10.4	8
Lincoln Continental	10.4	8
Chrysler Imperial	14.7	8
Fiat 128	32.4	4
Honda Civic	30.4	4
Toyota Corolla	33.9	4
Toyota Corona	21.5	4
Dodge Challenger	15.5	8
AMC Javelin	15.2	8
Camaro Z28	13.3	8
Pontiac Firebird	19.2	8
Fiat X1-9	27.3	4
Porsche 914-2	26.0	4
Lotus Europa	30.4	4

	MPG	CYL
Ford Pantera L	15.8	8
Ferrari Dino	19.7	6
Maserati Bora	15.0	8
Volvo 142E	21.4	4

Menambahkan baris dan kolom

```
c1 <- c(10,20,30,40,50)
c2 <- letters[c(1:5)]
df <- data.frame(kol1=c1,kol2=c2)
df
```

KOL1	KOL2
10	a
20	b
30	c
40	d
50	e

```
df1 <- data.frame(kol1=128,kol2='Meteorologi')
df1
```

KOL1	KOL2
128	Meteorologi

```
# menambahkan df1 ke df
df <- rbind(df,df1)
df
```

KOL1	KOL2
10	a
20	b
30	c
40	d

KOL1	KOL2
50	e
128	Meteorologi

```
c3 <- c(11:16)
c3
```

1. 11
2. 12
3. 13
4. 14
5. 15
6. 16

```
# menambahkan kolom ke df
df <- cbind(df,kol3=c3)
df
```

KOL1	KOL2	KOL3
10	a	11
20	b	12
30	c	13
40	d	14
50	e	15
128	Meteorologi	16

```
df$kol4 <- c(20,25,30,35,40,45)
df
```

KOL1	KOL2	KOL3	KOL4
10	a	11	20
20	b	12	25
30	c	13	30
40	d	14	35

KOL1	KOL2	KOL3	KOL4
50	e	15	40
128	Meteorologi	16	45

```
df$kol5 <- df$kol1 * 2
df
```

KOL1	KOL2	KOL3	KOL4	KOL5
10	a	11	20	20
20	b	12	25	40
30	c	13	30	60
40	d	14	35	80
50	e	15	40	100
128	Meteorologi	16	45	256

Mengatur penamaan kolom

```
colnames(df) # mengetahui nama kolom
```

1. 'kol1'
2. 'kol2'
3. 'kol3'
4. 'kol4'
5. 'kol5'

```
colnames(df) <- c('A', 'B', 'C', 'D', 'E') # penamaan
ulang nama kolom
df
```

A	B	C	D	E
10	a	11	20	20
20	b	12	25	40
30	c	13	30	60
40	d	14	35	80
50	e	15	40	100
128	Meteorologi	16	45	256

```
colnames(df)[1] <- 'X' # penamaan ulang kolom secara
individual
df
```

X	B	C	D	E
10	a	11	20	20
20	b	12	25	40
30	c	13	30	60
40	d	14	35	80
50	e	15	40	100
128	Meteorologi	16	45	256

```
colnames(df)[c(2,3)] <- c('Y','Z')
df
```

X	Y	Z	D	E
10	a	11	20	20
20	b	12	25	40
30	c	13	30	60
40	d	14	35	80
50	e	15	40	100
128	Meteorologi	16	45	256

Menyeleksi banyak baris dan kolom

```
df[1:3,]
```

X	Y	Z	D	E
10	a	11	20	20
20	b	12	25	40
30	c	13	30	60

```
# menyeleksi seluruh baris, kecuali no 3
df[-3,]
```

	X	Y	Z	D	E
1	10	a	11	20	20
2	20	b	12	25	40
4	40	d	14	35	80
5	50	e	15	40	100
6	128	Meteorologi	16	45	256

```
# penyeleksian kondisional
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

```
mtcars[mtcars$mpg > 20,]
```

	MPG	CYL	DISP	HP	DRAT	WT	QSEC	VS	AM	GEAR	CARB
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2

	MPG	CYL	DISP	HP	DRAT	WT	QSEC	VS	AM	GEAR	CARB
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

```
mtcars[mtcars$mpg > 20 & mtcars$cyl == 6,]
```

	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
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1

```
mtcars[mtcars$mpg > 20 & mtcars$cyl == 6, c('mpg',
'cyl','hp')]
```

	MPG	CYL	HP
Mazda RX4	21.0	6	110
Mazda RX4 Wag	21.0	6	110
Hornet 4 Drive	21.4	6	110

```
subset(mtcars, mpg > 20 & cyl == 6) # pakai built-in
subset function
```

	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
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1

```
df[,1:3]
```

X	Y	Z
10	a	11
20	b	12
30	c	13
40	d	14
50	e	15
128	Meteorologi	16

```
df[,c(3,5)]
```

Z	E
11	20
12	40
13	60
14	80
15	100
16	256

```
df[,c('Z', 'E')]
```

Z	E
11	20
12	40
13	60
14	80
15	100
16	256

Menangani data kosong

```
df
```

X	Y	Z	D	E
10	a	11	20	20
20	b	12	25	40
30	c	13	30	60
40	d	14	35	80
50	e	15	40	100
128	Meteorologi	16	45	256

```
is.na(df) # cek data kosong
```

X	Y	Z	D	E
FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE

```
any(is.na(df))
```

FALSE

```
any(is.na(df$X))
```

FALSE

```
# Membuat na jadi 0
any(is.na(df)) <- 0.5 # error karena ga ada na
```

```
Error in any(is.na(df)) <- 0.5: could not find function
"any<-"
Traceback:
```

```
any(is.na(df$D)) <- mean(df$D)  # error karena ga ada na
```

```
Error in any(is.na(df$D)) <- mean(df$D): could not find
function "any<-"
Traceback:
```

List

Kumpulan berbagai macam tipe data di R

```
v <- c(1,2,3,4,5)
M <- matrix(1:10, nrow=2)
c1 <- c('Ignatius', 'Laynez', 'Faber', 'Xaverius',
'Kanisius')
c2 <- c(42,37,28,45,43)
```

```
df <- data.frame(Nama = c1, ID = c2)
df
```

NAMA	ID
Ignatius	42
Laynez	37
Faber	28
Xaverius	45
Kanisius	43

```
# Pendefinisian list
l <- list(v,M,df)
l
```

1.
    a. 1
    b. 2
    c. 3
    d. 4
    e. 5

2.

1	3	5	7	9
2	4	6	8	10

3.

NAMA	ID
Ignatius	42
Laynez	37
Faber	28
Xaverius	45
Kanisius	43

```
# penamaaan ulang indeks list

l2 <- list(sampel_vektor = v, sampel_matriks = M,
sample_data_frame = df)
l2
```

$sampel_vektor

1. 1
2. 2
3. 3
4. 4
5. 5

$sampel_matriks

1	3	5	7	9
2	4	6	8	10

$sample_data_frame

NAMA	ID
Ignatius	42
Laynez	37
Faber	28
Xaverius	45
Kanisius	43

```
l2[1]
```

$sampel_vektor =

1. 1
2. 2
3. 3
4. 4
5. 5

```
l2['sampel_vektor']
```

$sampel_vektor =

1. 1
2. 2
3. 3
4. 4
5. 5

```
l2$sampel_vektor
```

1. 1
2. 2
3. 3
4. 4
5. 5

```
l2[['sampel_vektor']]
```

1. 1
2. 2
3. 3
4. 4
5. 5

```
print(class(l2['sampel_vektor']))
print(class(l2[1]))
```

```
[1] "list"
[1] "list"
```

```
print(class(l2$sampel_vektor))
print(class(l2[['sampel_vektor']]))
```

```
[1] "numeric"
[1] "numeric"
```

```
# Mengombinasikan dua list
l3 <- c(l,l2)
l3
```

[[1]]

1. 1
2. 2
3. 3
4. 4
5. 5

[[2]]

1	3	5	7	9
2	4	6	8	10

[[3]]

NAMA	ID
Ignatius	42
Laynez	37
Faber	28
Xaverius	45
Kanisius	43

$sampel_vektor

1. 1
2. 2
3. 3
4. 4
5. 5

$sampel_matriks

1	3	5	7	9
2	4	6	8	10

$sample_data_frame

NAMA	ID
Ignatius	42
Laynez	37
Faber	28
Xaverius	45
Kanisius	43

```
str(l2)
```

```
List of 3
 $ sampel_vektor    : num [1:5] 1 2 3 4 5
 $ sampel_matriks   : int [1:2, 1:5] 1 2 3 4 5 6 7 8 9 10
 $ sample_data_frame:'data.frame':  5 obs. of  2
variables:
  ..$ Nama: Factor w/ 5 levels "Faber","Ignatius",..: 2 4
1 5 3
  ..$ ID  : num [1:5] 42 37 28 45 43
```

```
summary(l2)
```

```
                  Length Class      Mode
sampel_vektor      5     -none-     numeric
sampel_matriks    10     -none-     numeric
sample_data_frame  2     data.frame list
```

Penanganan data tabular

Penanganan data csv

```
getwd() # tahu di mana posisi kita saat ini
```

'/home/ronggolawe/coding_repo/tutorialStatdasR/notebooks'

```
setwd("/home/ronggolawe/coding_repo/tutorialStatdasR/notebooks") # mengatur posisi kita
```

```
getwd()
```

'/home/ronggolawe/coding_repo/tutorialStatdasR/notebooks'

```
# Membaca csv
data <- read.csv("../data/gaji.csv")
data
```

ID	NAMA	GAJI	JURUSAN
1	Petrus	1000000	Teologi
2	Matius	2000000	Filsafat
3	Markus	5000000	Meteorologi
4	Barnabas	10000000	Teknik Informatika
5	Thomas	20000000	Sistem Informasi
6	Ignatius	500000	Pendidikan Agama
7	Aisyah	25000000	Teknik Elektro
8	Supriyanto	1500000	Ilmu Perpustakaan

```
# menuliskan csv
head(mtcars)
```

	MPG	CYL	DISP	HP	DRAT	WT	QSEC	VS	AM	GEAR	CARB

	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

```
write.csv(mtcars, file = "../data/tes.csv")
```

```
class(data)
```

'data.frame'

```
# menuliskan csv dari data frame
c1 <- c(10,20,30,40,50)
c2 <- c('A', 'B', 'C', 'D', 'E')
df <- data.frame(c1,c2)
df
```

C1	C2
10	A
20	B
30	C
40	D
50	E

```
write.csv(df, file = '../data/tes2.csv')
```

```
# Untuk mengetahui secara lebih lanjut, perintahkan:
# help(read.csv)
```

Penanganan data excel

```
library(readxl) # memuat pustaka readxl
```

```
excel_sheets("../data/contoh.xlsx")
```

1. 'Sheet1'
2. 'Sheet2'

```
# membaca file excel
df <- read_excel("../data/contoh.xlsx", sheet = "Sheet1")
df
```

NO	NAMA DEPAN	NAMA BELAKANG	JENIS KELAMIN	NEGARA	USIA	ID
1	Fernando	Sanchez	Pria	Meksiko	28	1562
2	Sandy	Herho	Pria	Indonesia	27	1582
3	Mara	Hashimoto	Wanita	Jepang	25	2587
4	Philip	Gent	Pria	Belgia	32	2468
5	Satya	Narendra	Pria	India	42	6548
6	Vincenza	Welland	Wanita	Amerika Serikat	40	3598
7	Rudy	Salim	Pria	Indonesia	65	7865
8	Gaston	Brumm	Pria	Amerika Serikat	24	2456
9	Etta	Hurn	Wanita	Britania Raya	34	1785

```
summary(df)
```

```
       No     Nama Depan         Nama Belakang      Jenis
Kelamin
 Min.   :1   Length:9           Length:9
Length:9
 1st Qu.:3   Class :character   Class :character   Class
:character
 Median :5   Mode  :character   Mode  :character   Mode
:character
```

```
 Mean   :5

 3rd Qu.:7

 Max.   :9

    Negara               Usia             ID
 Length:9           Min.   :24.00   Min.   :1562
 Class :character   1st Qu.:27.00   1st Qu.:1785
 Mode  :character   Median :32.00   Median :2468
                    Mean   :35.22   Mean   :3383
                    3rd Qu.:40.00   3rd Qu.:3598
                    Max.   :65.00   Max.   :7865
```

```
str(df)
```

```
tibble [9 × 7] (S3: tbl_df/tbl/data.frame)
 $ No           : num [1:9] 1 2 3 4 5 6 7 8 9
 $ Nama Depan   : chr [1:9] "Fernando" "Sandy" "Mara"
"Philip" ...
 $ Nama Belakang: chr [1:9] "Sanchez" "Herho" "Hashimoto"
"Gent" ...
 $ Jenis Kelamin: chr [1:9] "Pria" "Pria" "Wanita" "Pria"
...
 $ Negara       : chr [1:9] "Meksiko" "Indonesia" "Jepang"
"Belgia" ...
 $ Usia         : num [1:9] 28 27 25 32 42 40 65 24 34
 $ ID           : num [1:9] 1562 1582 2587 2468 6548 ...
```

```
mean(df$Usia)
```

35.2222222222222

```
df1 <- read_excel("../data/contoh.xlsx", sheet='Sheet2')
df1
```

BILANGAN	KUADRAT
1	1
2	4
3	9
4	16

BILANGAN	KUADRAT
5	25

```
# menulis file excel
library(writexl)
```

```
c1 <- c(1:5)
c2 <- 6:10
df2 <- data.frame(c1,c2)
df2
```

C1	C2
1	6
2	7
3	8
4	9
5	10

```
write_xlsx(df2, "../data/tes.xlsx")
```

Konsep - konsep inti pemrograman

Operator - operator logika

`&` (AND)

```
b <- 15
b
```

15

```
b < 20
```

TRUE

```
b > 10
```

TRUE

```
b > 10 & b < 20 # TRUE & TRUE = TRUE
```

TRUE

```
b > 30 & b < 20 # FALSE & TRUE = FALSE
```

FALSE

```
(b > 5) & (b < 25) & (b == 15) # TRUE & TRUE & TRUE = TRUE
```

TRUE

```
(b < 5) & (b < 25) & (b == 15) # FALSE & TRUE & TRUE =
FALSE
```

FALSE

`|` (OR)

```
(b==10) | (b < 25) # FALSE | TRUE = TRUE
```

TRUE

```
(b > 10) | (b < 10)
```

TRUE

```
(b > 10) | (b < 10) | (b == 12)
```

TRUE

```
TRUE | FALSE
```

TRUE

```
FALSE | FALSE
```

FALSE

`!` (NOT)

```
(b == 10) | (b < 25)
```

TRUE

```
!((b == 10) | (b < 25))
```

FALSE

```
!T
```

FALSE

```
!(b > 10)
```

FALSE

Penerapan operator - operator logika pada data frame

```
# Kita juga dapat menggunakan operator - operator logika
pada data frame
df <- mtcars
```

```
head(df)
```

	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

```
df[df$mpg > 20,'mpg']
```

1. 21
2. 21
3. 22.8
4. 21.4
5. 24.4
6. 22.8
7. 32.4
8. 30.4

9. 33.9
10. 21.5
11. 27.3
12. 26
13. 30.4
14. 21.4

```
subset(df, mpg > 20)
```

	MPG	CYL	DISP	HP	DRAT	WT	QSEC	VS	AM	GEAR	CARB
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

```
df[(df$mpg > 20) & (df$cyl > 4),]
```

	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
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1

```
df[(df$mpg > 20) | (df$cyl > 4),]
```

	MPG	CYL	DISP	HP	DRAT	WT	QSEC	VS	AM	GEAR	CARB
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1

	MPG	CYL	DISP	HP	DRAT	WT	QSEC	VS	AM	GEAR	CARB
Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

Pernyataan `if`, `else`, dan `else if`

```
if (5 > 3){
    print('Bener, Bro!')
}
```

```
[1] "Bener, Bro!"
```

```
if (5 < 3){
    print('Bener, Bro!')
}

# tidak ada output karena kondisi FALSE
```

```
if (5 < 3){
    print('Bener, Bro!')
}else{
    print('Salah, Bro!')
}
```

```
[1] "Salah, Bro!"
```

```
a <- 10
b <- 20

if (a > b){
    print('a lebih besar dari b.')
}else{
    print('a lebih kecil dari b.')
}
```

```
[1] "a lebih kecil dari b."
```

```
minuman <- 'Coca Cola'

if (minuman == 'Kopi'){
    print('Ngopi, Bro!')
}else if (minuman == 'Coca Cola'){
    print('Mantap!')
}else if (minuman == 'air putih'){
    print('Bagus, Bro buat kesehatan.')
}else{
    print('Terserah mau minum apa, Bro yang penting
halal.')
}
```

```
[1] "Mantap!"
```

Pengulangan `while`

```
b <- 0
while (b < 10){
    print(b)
    b = b + 1
}
```

```
[1] 0
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
```

```
b <- 0
while (b < 10){
    print(paste0('b sama dengan ',b))
    b = b + 1
}
```

```
[1] "b sama dengan 0"
[1] "b sama dengan 1"
[1] "b sama dengan 2"
[1] "b sama dengan 3"
[1] "b sama dengan 4"
[1] "b sama dengan 5"
[1] "b sama dengan 6"
[1] "b sama dengan 7"
[1] "b sama dengan 8"
[1] "b sama dengan 9"
```

```
b <- 0
while (b < 10){
    print(paste0('b sama dengan ',b))
    b = b + 1
    if (b == 10){
        print('b sama dengan 10. Pengulangan selesai.')
    }
}
```

```
[1] "b sama dengan 0"
[1] "b sama dengan 1"
[1] "b sama dengan 2"
[1] "b sama dengan 3"
[1] "b sama dengan 4"
[1] "b sama dengan 5"
[1] "b sama dengan 6"
[1] "b sama dengan 7"
[1] "b sama dengan 8"
[1] "b sama dengan 9"
[1] "b sama dengan 10. Pengulangan selesai."
```

```r
b <- 0
while (b < 10){
    print(paste0('b sama dengan ',b))
    b = b + 1
    if (b == 10){
        print('b sama dengan 10. Pengulangan selesai.')
        print('Yuk Belajar pemrograman R!')
    }
}
```

```
[1] "b sama dengan 0"
[1] "b sama dengan 1"
[1] "b sama dengan 2"
[1] "b sama dengan 3"
[1] "b sama dengan 4"
[1] "b sama dengan 5"
[1] "b sama dengan 6"
[1] "b sama dengan 7"
[1] "b sama dengan 8"
[1] "b sama dengan 9"
[1] "b sama dengan 10. Pengulangan selesai."
[1] "Yuk Belajar pemrograman R!"
```

```r
# penggunaan break() untuk mengakhiri pengulangan
b <- 0
while (b < 10){
    print(paste0('b sama dengan ',b))
    b = b + 1
    if (b == 10){
        print('b sama dengan 10. Pengulangan selesai.')
        break()
        print('Yuk Belajar pemrograman R!')
    }
}
```

```
[1] "b sama dengan 0"
[1] "b sama dengan 1"
[1] "b sama dengan 2"
[1] "b sama dengan 3"
[1] "b sama dengan 4"
[1] "b sama dengan 5"
[1] "b sama dengan 6"
[1] "b sama dengan 7"
[1] "b sama dengan 8"
[1] "b sama dengan 9"
[1] "b sama dengan 10. Pengulangan selesai."
```

```
b <- 0
while (b < 10){
    print(paste0('b sama dengan ',b))
    b = b + 1
    if (b == 5){
        print('b sama dengan 5. Pengulangan selesai.')
        break()
        print('Yuk Belajar pemrograman R!')
    }
}
```

```
[1] "b sama dengan 0"
[1] "b sama dengan 1"
[1] "b sama dengan 2"
[1] "b sama dengan 3"
[1] "b sama dengan 4"
[1] "b sama dengan 5. Pengulangan selesai."
```

Pengulangan `for`

```
v <- c(1,2,3,4,5)
```

```
v
```

1. 1
2. 2
3. 3
4. 4
5. 5

```
for (i in v){
    print(i)
}
```

```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

```
for (i in v){
    print('Halo')
}
```

```
[1] "Halo"
[1] "Halo"
[1] "Halo"
[1] "Halo"
[1] "Halo"
```

```
buah2an <- c('Apel', 'Jeruk', 'Pisang', 'Mangga')

for (buah in buah2an){
    print(buah)
}
```

```
[1] "Apel"
[1] "Jeruk"
[1] "Pisang"
[1] "Mangga"
```

```
# Pengulangan for pada list
c1 <- c(10,20,30,40,50)
c2 <- c('A', 'B', 'C', 'D', 'E')
df <- data.frame(c1,c2)
df
```

C1	C2
10	A
20	B
30	C
40	D
50	E

```
v
```

1. 1
2. 2

3. 3
4. 4
5. 5

```
l <- list(v,df)
l
```

1.
    a. 1
    b. 2
    c. 3
    d. 4
    e. 5
2.

C1	C2
10	A
20	B
30	C
40	D
50	E

```
for (i in l){
    print(i)
}
```

```
[1] 1 2 3 4 5
  c1 c2
1 10  A
2 20  B
3 30  C
4 40  D
5 50  E
```

```
# Pengunaan pengulangan for pada matriks
mat <- matrix(1:25, nrow=5, byrow=T)
mat
```

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

```
for (num in mat){
    print(num)
}
```

```
[1] 1
[1] 6
[1] 11
[1] 16
[1] 21
[1] 2
[1] 7
[1] 12
[1] 17
[1] 22
[1] 3
[1] 8
[1] 13
[1] 18
[1] 23
[1] 4
[1] 9
[1] 14
[1] 19
[1] 24
[1] 5
```

```
[1] 10
[1] 15
[1] 20
[1] 25
```

```
# Pengulangan for bersarang
for (i in 1:5){
    print(i^2)
}
```

```
[1] 1
[1] 4
[1] 9
[1] 16
[1] 25
```

```
for (i in 1:5){
    for (j in 1:2){
        print(i*j)
    }
}
```

```
[1] 1
[1] 2
[1] 2
[1] 4
[1] 3
[1] 6
[1] 4
[1] 8
[1] 5
[1] 10
```

Proses kerjanya:

1. 1×1
2. 1×2
3. 2×1
4. 2×2
5. 3×1
6. 3×2, dst...

Fungsi

```
# sintaks
nama_fungsi <- function(arg_1,arg_2, arg_3 = 10){
    # Kode yang hendak dieksekusi
    hasil <- arg_1 + arg_2
    return(hasil)
}
```

```
salam <- function(){
    print('Halo!')
}
```

```
salam()
```

```
[1] "Halo!"
```

```
salam <- function(nama){
    print(paste('Halo', nama, '!'))
}
```

```
salam('Sandy')
```

```
[1] "Halo Sandy !"
```

```
salam() # error karena tidak ada nama default
```

```
Error in paste("Halo", nama, "!"): argument "nama" is
missing, with no default
Traceback:
```

```
1. salam()

2. print(paste("Halo", nama, "!"))   # at line 2 of file
<text>

3. paste("Halo", nama, "!")   # at line 2 of file <text>
```

```r
salam <- function(nama = 'Priska'){
    print(paste('Halo', nama, '!'))
}
```

```r
salam()
```

```
[1] "Halo Priska !"
```

```r
salam('Sandy')
```

```
[1] "Halo Sandy !"
```

```r
penjumlahan <- function(b1,b2){
    print(b1 + b2)
}
```

```r
penjumlahan(10,20)
```

```
[1] 30
```

```r
# Harusnya pakai return
penjumlahan <- function(b1,b2){
    return(b1 + b2)
}
```

```r
x <- penjumlahan(10,20) # dapat disimpan di variabel
x
```



```r
# Jangkauan variabel
kuadrat <- function(x){
    hasil <- x^2
    return(hasil)
}
```

```
out <- kuadrat(5)
out
```



```
hasil # ga ada karena bersifat lokal
```

```
Error in eval(expr, envir, enclos): object 'hasil' not
found
Traceback:
```

```
var <- 'Variabel global'
internet <- 'Jaringan global'

jaringan <- function(internet){
    print(var)
    internet <- 'Jaringan lokal'
    print(internet)
}
```

```
jaringan()
# redefinisi variabel lokal internet
```

```
[1] "Variabel global"
[1] "Jaringan lokal"
```

```
print(internet) # di luar fungsi berlaku variabel global
```

```
[1] "Jaringan global"
```

Konsep - konsep pemrograman lanjut

Fitur - fitur *built-in*

`seq()`: Mendefinisikan sikuen

```
seq(0,10, by=2)
```

1. 0
2. 2
3. 4
4. 6
5. 8
6. 10

```
seq(0,100, by = 10)
```

1. 0
2. 10
3. 20
4. 30
5. 40
6. 50
7. 60
8. 70
9. 80
10. 90
11. 100

```
seq(0,30, by = 2)
```

1. 0
2. 2
3. 4
4. 6
5. 8
6. 10

7. 12
8. 14
9. 16
10. 18
11. 20
12. 22
13. 24
14. 26
15. 28
16. 30

`sort()` : Mengurutkan vektor

```
v <- c(2,7,1,49,54,32)
v
```

1. 2
2. 7
3. 1
4. 49
5. 54
6. 32

```
sort(v) # dari kecil ke besar
```

1. 1
2. 2
3. 7
4. 32
5. 49
6. 54

```
sort(v, decreasing = T) # dari besar ke kecil
```

1. 54
2. 49
3. 32
4. 7
5. 2
6. 1

```
nama <- c('s', 'a', 'n', 'd', 'y')
nama
```

1. 's'
2. 'a'
3. 'n'
4. 'd'
5. 'y'

```
sort(nama)
```

1. 'a'
2. 'd'
3. 'n'
4. 's'
5. 'y'

```
nama <- c('s', 'a', 'n', 'd', 'Y')
sort(nama)
```

1. 'a'
2. 'd'
3. 'n'
4. 's'
5. 'Y'

```
nama <- c('s', 'a', 'n', 'd', 'Y','A')
sort(nama)
```

1. 'a'
2. 'A'
3. 'd'
4. 'n'
5. 's'
6. 'Y'

`rev()`: Membalikan elemen di dalam suatu objek

```
b <- seq(1,10)
b
```

1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10

```
rev(b)
```

1. 10
2. 9
3. 8
4. 7
5. 6
6. 5
7. 4
8. 3
9. 2
10. 1

```
d <- c('a','b','e','d')
d
```

1. 'a'
2. 'b'
3. 'e'
4. 'd'

```
rev(d)
```

1. 'd'
2. 'e'
3. 'b'
4. 'a'

`str()`: Menunjukkan struktur dari suatu objek

```
str(b)
```

```
 int [1:10] 1 2 3 4 5 6 7 8 9 10
```

```
str(mtcars)
```

```
'data.frame':   32 obs. of  11 variables:
 $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8
19.2 ...
 $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
 $ disp: num  160 160 108 258 360 ...
 $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92
3.92 ...
 $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num  16.5 17 18.6 19.4 17 ...
 $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
 $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
 $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
 $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

```
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
```

`append()`: Menggabungkan objek

```
v1 <- seq(1,5)
v2 <- seq(10,30, by=10)
```

```
append(v1,v2)
```

1. 1
2. 2
3. 3
4. 4
5. 5
6. 10
7. 20
8. 30

Memeriksa dan mengonversi tipe data pada objek - objek R

```
v1
```

1. 1
2. 2
3. 3
4. 4
5. 5

```
is.vector(v1)
```

TRUE

```
is.data.frame(v1)
```

FALSE

```
is.data.frame(mtcars)
```

TRUE

```
as.list(v1)
```

1. 1
2. 2
3. 3
4. 4
5. 5

```
as.matrix(v1)
```

1
2
3
4
5

Fungsi - fungsi `apply`

```
v <- seq(10,50,by=10)
v
```

1. 10
2. 20
3. 30
4. 40
5. 50

```
sample(v,2) # mengambil dua buah sampel acak dari vektor
```

1. 30
2. 20

```
sample(1:100,5)
```

1. 98
2. 17
3. 78
4. 100
5. 40

```
v <- 1:5
v
```

1. 1
2. 2
3. 3
4. 4
5. 5

```
tambah_acak <- function(x){
    acak <- sample(1:100,1)
    return(x + acak)
}
```

```
tambah_acak(10)
```

63

```
hasil <- tambah_acak(20)
hasil
```

91

`lapply()`: dalam bentuk list

```
lapply(v, tambah_acak)
# outputnya dalam bentuk list
```

1. 80
2. 95
3. 36
4. 102
5. 8

`sapply()`: dalam bentuk vektor

```
sapply(v,tambah_acak)
```

1. 52
2. 8
3. 84
4. 22
5. 8

```
v1 <- seq(5,25, by=5)
kuadrat <- function(bil){
    return(bil^2)
}
```

```
kuadrat(5)
```

25

```
lapply(v, kuadrat)
```

1. 1
2. 4
3. 9
4. 16
5. 25

```
sapply(v, kuadrat)
```

1. 1
2. 4
3. 9
4. 16
5. 25

Fungsi anonim

```
v
```

1. 1
2. 2
3. 3
4. 4
5. 5

```
kuadrat <- function(bil){
    return(bil^2)
}
```

```
sapply(v, function(bil){bil^2}) # fungsi anonim
```

1. 1
2. 4
3. 9
4. 16
5. 25

Fungsi `apply` dengan banyak *input*

```
v
```

1. 1
2. 2
3. 3
4. 4
5. 5

```
tambah_dua_bil <- function(b1,b2){
    return(b1+b2)
}
```

```
tambah_dua_bil(20,30)
```

50

```
sapply(v, tambah_dua_bil) # error
```

```
Error in FUN(X[[i]], ...): argument "b2" is missing, with
no default
Traceback:
```

```
1. sapply(v, tambah_dua_bil)

2. lapply(X = X, FUN = FUN, ...)

3. FUN(X[[i]], ...)
```

```
sapply(v, tambah_dua_bil, b2 = 10)
```

1. 11
2. 12
3. 13
4. 14
5. 15

Ekspresi regular : RegEx

```
txt <- "Halo semuanya! Selamat Pagi! Cuaca lagi bagus, nih
buat touring."
```

```
txt
```

'Halo semuanya! Selamat Pagi! Cuaca lagi bagus, nih buat touring.'

```
grepl("Halo",txt) # kata "Halo" ada di txt
```

TRUE

```
grepl("Malam", txt)
```

FALSE

```
grepl("halo", txt) # Sifatnya case-sensitive
```

FALSE

```
v <- c('a','d','k', 'l','t','k')
grepl('k',v)
```

1. FALSE
2. FALSE
3. TRUE
4. FALSE
5. FALSE
6. TRUE

```
grep('k',v) # outputnya indeks
```

1. 3
2. 6

```
grep('a',v)
```

1

Fungsi - fungsi matematika

`abs()`: menghitung nilai absolut

```
abs(-2)
```

2

```
v <- c(-3,-5,7,10)
abs(v)
```

1. 3
2. 5
3. 7
4. 10

`sum()`: menghitung penjumlahan seluruh elemen

```
sum(2,4,6)
```

12

```
v <- c(2,3,4,5)
sum(v)
```

`mean()`: menghitung rata - rata aritmatika

```
mean(v)
```

3.5

```
mean(c(3,4,5))
```

4

`round()`: membulatkan nilai

```
round(2.777645)
```

3

```
round(2.777645, digits=2)
```

2.78

```
round(2.777645, 4)
```

2.7776

Dates dan *Timestamps*

```
Sys.Date() # waktu saat ini
```

2020-07-01

```
d <- Sys.Date()
d
```

2020-07-01

```
class(d)
```

'Date'

```
d <- '1993-03-13'
d
```

'1993-03-13'

```
class(d)
```

'character'

```
# dikonversi menjadi date
b.day <- as.Date(d)
b.day
```

1993-03-13

```
class(b.day)
```

'Date'

```
as.Date('Mar-13-93') # format tidak sesuai
```

```
Error in charToDate(x): character string is not in a
standard unambiguous format
Traceback:
```

```
1. as.Date("Mar-13-93")

2. as.Date.character("Mar-13-93")

3. charToDate(x)

4. stop("character string is not in a standard unambiguous
format")
```

```
as.Date('Mar-13-93', format = '%b-%d-%y')
```

1993-03-13

- `%d`: hari (desimal)
- `%m`: bulan (desimal)
- `%b`: bulan (singkatan)
- `%B`: bulan (tidak disingkat)
- `%y`: tahun (2 digit)
- `%Y`: tahun (4 digit)

```
as.Date('March,01,2009', format= "%B, %d, %Y")
```

2009-03-01

```
# POSIXct
```

```
as.POSIXct('11:03:05', format='%H:%M:%S')
```

```
[1] "2020-07-01 11:03:05 WIB"
```

```
strptime('11:03:05', format = '%H:%M:%S') # lebih banyak
dipakai di pemrograman R
```

```
[1] "2020-07-01 11:03:05 WIB"
```

Manipulasi data

dplyr

```
library(dplyr)
library(nycflights13)
```

```
Attaching package: ‘dplyr’

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

```
head(flights)
```

YEAR	MONTH	DAY	DEP_TIME	SCHED_DEP_TIME	DEP_DELAY	ARR_TIME	SCHED_ARR_TIME	ARR_DELAY	CARRIER
2013	1	1	517	515	2	830	819	11	UA
2013	1	1	533	529	4	850	830	20	UA
2013	1	1	542	540	2	923	850	33	AA
2013	1	1	544	545	-1	1004	1022	-18	B6
2013	1	1	554	600	-6	812	837	-25	DL
2013	1	1	554	558	-4	740	728	12	UA

`filter()`

```
head(filter(flights,month == 5, day == 2, carrier ==
'AA'))
```

YEAR	MONTH	DAY	DEP_TIME	SCHED_DEP_TIME	DEP_DELAY	ARR_TIME	SCHED_ARR_TIME	ARR_DELAY	CARRIER
2013	5	2	539	540	-1	850	840	10	AA
2013	5	2	549	600	-11	823	850	-27	AA
2013	5	2	558	605	-7	855	910	-15	AA
2013	5	2	603	610	-7	729	745	-16	AA
2013	5	2	611	615	-4	900	915	-15	AA
2013	5	2	627	630	-3	736	805	-29	AA

```
head(flights[flights$month == 5 & flights$day == 2 &
flights$carrier == 'AA',]) # ribet
```

YEAR	MONTH	DAY	DEP_TIME	SCHED_DEP_TIME	DEP_DELAY	ARR_TIME	SCHED_ARR_TIME	ARR_DELAY	CARRIER
2013	5	2	539	540	-1	850	840	10	AA

YEAR	MONTH	DAY	DEP_TIME	SCHED_DEP_TIME	DEP_DELAY	ARR_TIME	SCHED_ARR_TIME	ARR_DELAY	CARRIER
2013	5	2	549	600	-11	823	850	-27	AA
2013	5	2	558	605	-7	855	910	-15	AA
2013	5	2	603	610	-7	729	745	-16	AA
2013	5	2	611	615	-4	900	915	-15	AA
2013	5	2	627	630	-3	736	805	-29	AA

`slice()`

```
slice(flights, 1:10) # menyeleksi 10 baris pertama
```

YEAR	MONTH	DAY	DEP_TIME	SCHED_DEP_TIME	DEP_DELAY	ARR_TIME	SCHED_ARR_TIME	ARR_DELAY	CARRIER
2013	1	1	517	515	2	830	819	11	UA
2013	1	1	533	529	4	850	830	20	UA
2013	1	1	542	540	2	923	850	33	AA
2013	1	1	544	545	-1	1004	1022	-18	B6
2013	1	1	554	600	-6	812	837	-25	DL
2013	1	1	554	558	-4	740	728	12	UA
2013	1	1	555	600	-5	913	854	19	B6
2013	1	1	557	600	-3	709	723	-14	EV
2013	1	1	557	600	-3	838	846	-8	B6
2013	1	1	558	600	-2	753	745	8	AA

`arrange()`

```
head(arrange(flights, year, month, day, arr_time))
# mengatur urutan sesuai kolomnya
```

YEAR	MONTH	DAY	DEP_TIME	SCHED_DEP_TIME	DEP_DELAY	ARR_TIME	SCHED_ARR_TIME	ARR_DELAY	CARRIER
2013	1	1	1929	1920	9	3	7	-4	UA
2013	1	1	2121	2040	41	6	2323	43	B6
2013	1	1	2058	2100	-2	8	2359	9	UA
2013	1	1	2120	2130	-10	16	18	-2	B6
2013	1	1	2134	2045	49	20	2352	28	UA
2013	1	1	2312	2000	192	21	2110	191	EV

`select()`

```
head(select(flights, arr_time)) # seleksi kolom arr_time
```

ARR_TIME

ARR_TIME
830
850
923
1004
812
740

```
head(select(flights,carrier)) # seleksi kolom carrier
```

CARRIER
UA
UA
AA
B6
DL
UA

```
head(select(flights, arr_time, carrier, month)) # seleksi
3 kolom
```

ARR_TIME	CARRIER	MONTH
830	UA	1
850	UA	1
923	AA	1
1004	B6	1
812	DL	1
740	UA	1

`rename()`:

```
head(flights)
```

YEAR	MONTH	DAY	DEP_TIME	SCHED_DEP_TIME	DEP_DELAY	ARR_TIME	SCHED_ARR_TIME	ARR_DELAY	CARRIER
2013	1	1	517	515	2	830	819	11	UA
2013	1	1	533	529	4	850	830	20	UA
2013	1	1	542	540	2	923	850	33	AA
2013	1	1	544	545	-1	1004	1022	-18	B6
2013	1	1	554	600	-6	812	837	-25	DL
2013	1	1	554	558	-4	740	728	12	UA

```
rename(flights,new_arr_time = arr_time) # mengubah nama
kolom
```

YEAR	MONTH	DAY	DEP_TIME	SCHED_DEP_TIME	DEP_DELAY	NEW_ARR_TIME	SCHED_ARR_TIME	ARR_DELAY	CAR
2013	1	1	517	515	2	830	819	11	UA

YEAR	MONTH	DAY	DEP_TIME	SCHED_DEP_TIME	DEP_DELAY	NEW_ARR_TIME	SCHED_ARR_TIME	ARR_DELAY	CAR
2013	1	1	533	529	4	850	830	20	UA
2013	1	1	542	540	2	923	850	33	AA
2013	1	1	544	545	-1	1004	1022	-18	B6
2013	1	1	554	600	-6	812	837	-25	DL
2013	1	1	554	558	-4	740	728	12	UA
2013	1	1	555	600	-5	913	854	19	B6
2013	1	1	557	600	-3	709	723	-14	EV
2013	1	1	557	600	-3	838	846	-8	B6
2013	1	1	558	600	-2	753	745	8	AA
2013	1	1	558	600	-2	849	851	-2	B6
2013	1	1	558	600	-2	853	856	-3	B6
2013	1	1	558	600	-2	924	917	7	UA
2013	1	1	558	600	-2	923	937	-14	UA
2013	1	1	559	600	-1	941	910	31	AA
2013	1	1	559	559	0	702	706	-4	B6
2013	1	1	559	600	-1	854	902	-8	UA
2013	1	1	600	600	0	851	858	-7	B6
2013	1	1	600	600	0	837	825	12	MQ
2013	1	1	601	600	1	844	850	-6	B6
2013	1	1	602	610	-8	812	820	-8	DL
2013	1	1	602	605	-3	821	805	16	MQ
2013	1	1	606	610	-4	858	910	-12	AA
2013	1	1	606	610	-4	837	845	-8	DL
2013	1	1	607	607	0	858	915	-17	UA
2013	1	1	608	600	8	807	735	32	MQ
2013	1	1	611	600	11	945	931	14	UA
2013	1	1	613	610	3	925	921	4	B6
2013	1	1	615	615	0	1039	1100	-21	B6
2013	1	1	615	615	0	833	842	-9	DL
…	…	…	…	…	…	…	…	…	…
2013	9	30	2123	2125	-2	2223	2247	-24	EV
2013	9	30	2127	2129	-2	2314	2323	-9	EV
2013	9	30	2128	2130	-2	2328	2359	-31	B6
2013	9	30	2129	2059	30	2230	2232	-2	EV
2013	9	30	2131	2140	-9	2225	2255	-30	MQ

YEAR	MONTH	DAY	DEP_TIME	SCHED_DEP_TIME	DEP_DELAY	NEW_ARR_TIME	SCHED_ARR_TIME	ARR_DELAY	CAR
2013	9	30	2140	2140	0	10	40	-30	AA
2013	9	30	2142	2129	13	2250	2239	11	EV
2013	9	30	2145	2145	0	115	140	-25	B6
2013	9	30	2147	2137	10	30	27	3	B6
2013	9	30	2149	2156	-7	2245	2308	-23	UA
2013	9	30	2150	2159	-9	2250	2306	-16	EV
2013	9	30	2159	1845	194	2344	2030	194	9E
2013	9	30	2203	2205	-2	2339	2331	8	EV
2013	9	30	2207	2140	27	2257	2250	7	MQ
2013	9	30	2211	2059	72	2339	2242	57	EV
2013	9	30	2231	2245	-14	2335	2356	-21	B6
2013	9	30	2233	2113	80	112	30	42	UA
2013	9	30	2235	2001	154	59	2249	130	B6
2013	9	30	2237	2245	-8	2345	2353	-8	B6
2013	9	30	2240	2245	-5	2334	2351	-17	B6
2013	9	30	2240	2250	-10	2347	7	-20	B6
2013	9	30	2241	2246	-5	2345	1	-16	B6
2013	9	30	2307	2255	12	2359	2358	1	B6
2013	9	30	2349	2359	-10	325	350	-25	B6
2013	9	30	NA	1842	NA	NA	2019	NA	EV
2013	9	30	NA	1455	NA	NA	1634	NA	9E
2013	9	30	NA	2200	NA	NA	2312	NA	9E
2013	9	30	NA	1210	NA	NA	1330	NA	MQ
2013	9	30	NA	1159	NA	NA	1344	NA	MQ
2013	9	30	NA	840	NA	NA	1020	NA	MQ

`distinct()`

Untuk menyeleksi nilai - nilai unik

```
distinct(select(flights, carrier)) # nilai - nilai unik
pada kolom carrier
```

CARRIER
UA
AA
B6
DL
EV
MQ
US

CARRIER
WN
VX
FL
AS
9E
F9
HA
YV
OO

```
distinct(select(flights,month))
```

MONTH
1
10
11
12
2
3
4
5
6
7
8
9

`mutate()`

Menambahkan kolom baru di data frame

```
mutate(flights, kol_baru = arr_delay - dep_delay)
```

YEAR	MONTH	DAY	DEP_TIME	SCHED_DEP_TIME	DEP_DELAY	ARR_TIME	SCHED_ARR_TIME	ARR_DELAY	CARRIER
2013	1	1	517	515	2	830	819	11	UA
2013	1	1	533	529	4	850	830	20	UA
2013	1	1	542	540	2	923	850	33	AA
2013	1	1	544	545	-1	1004	1022	-18	B6
2013	1	1	554	600	-6	812	837	-25	DL
2013	1	1	554	558	-4	740	728	12	UA
2013	1	1	555	600	-5	913	854	19	B6
2013	1	1	557	600	-3	709	723	-14	EV
2013	1	1	557	600	-3	838	846	-8	B6
2013	1	1	558	600	-2	753	745	8	AA
2013	1	1	558	600	-2	849	851	-2	B6
2013	1	1	558	600	-2	853	856	-3	B6
2013	1	1	558	600	-2	924	917	7	UA

YEAR	MONTH	DAY	DEP_TIME	SCHED_DEP_TIME	DEP_DELAY	ARR_TIME	SCHED_ARR_TIME	ARR_DELAY	CARRIER
2013	1	1	558	600	-2	923	937	-14	UA
2013	1	1	559	600	-1	941	910	31	AA
2013	1	1	559	559	0	702	706	-4	B6
2013	1	1	559	600	-1	854	902	-8	UA
2013	1	1	600	600	0	851	858	-7	B6
2013	1	1	600	600	0	837	825	12	MQ
2013	1	1	601	600	1	844	850	-6	B6
2013	1	1	602	610	-8	812	820	-8	DL
2013	1	1	602	605	-3	821	805	16	MQ
2013	1	1	606	610	-4	858	910	-12	AA
2013	1	1	606	610	-4	837	845	-8	DL
2013	1	1	607	607	0	858	915	-17	UA
2013	1	1	608	600	8	807	735	32	MQ
2013	1	1	611	600	11	945	931	14	UA
2013	1	1	613	610	3	925	921	4	B6
2013	1	1	615	615	0	1039	1100	-21	B6
2013	1	1	615	615	0	833	842	-9	DL
…	…	…	…	…	…	…	…	…	…
2013	9	30	2123	2125	-2	2223	2247	-24	EV
2013	9	30	2127	2129	-2	2314	2323	-9	EV
2013	9	30	2128	2130	-2	2328	2359	-31	B6
2013	9	30	2129	2059	30	2230	2232	-2	EV
2013	9	30	2131	2140	-9	2225	2255	-30	MQ
2013	9	30	2140	2140	0	10	40	-30	AA
2013	9	30	2142	2129	13	2250	2239	11	EV
2013	9	30	2145	2145	0	115	140	-25	B6
2013	9	30	2147	2137	10	30	27	3	B6
2013	9	30	2149	2156	-7	2245	2308	-23	UA
2013	9	30	2150	2159	-9	2250	2306	-16	EV
2013	9	30	2159	1845	194	2344	2030	194	9E
2013	9	30	2203	2205	-2	2339	2331	8	EV
2013	9	30	2207	2140	27	2257	2250	7	MQ
2013	9	30	2211	2059	72	2339	2242	57	EV
2013	9	30	2231	2245	-14	2335	2356	-21	B6
2013	9	30	2233	2113	80	112	30	42	UA

YEAR	MONTH	DAY	DEP_TIME	SCHED_DEP_TIME	DEP_DELAY	ARR_TIME	SCHED_ARR_TIME	ARR_DELAY	CARRIER
2013	9	30	2235	2001	154	59	2249	130	B6
2013	9	30	2237	2245	-8	2345	2353	-8	B6
2013	9	30	2240	2245	-5	2334	2351	-17	B6
2013	9	30	2240	2250	-10	2347	7	-20	B6
2013	9	30	2241	2246	-5	2345	1	-16	B6
2013	9	30	2307	2255	12	2359	2358	1	B6
2013	9	30	2349	2359	-10	325	350	-25	B6
2013	9	30	NA	1842	NA	NA	2019	NA	EV
2013	9	30	NA	1455	NA	NA	1634	NA	9E
2013	9	30	NA	2200	NA	NA	2312	NA	9E
2013	9	30	NA	1210	NA	NA	1330	NA	MQ
2013	9	30	NA	1159	NA	NA	1344	NA	MQ
2013	9	30	NA	840	NA	NA	1020	NA	MQ

`transmute()`

Sama seperti `mutate()`, namu hanya mengeluarkan *output* kolom baru yang dihasilkan.

```
transmute(flights, kol_baru = arr_delay - dep_delay)
```

KOL_BARU
9
16
31
-17
-19
16
24
-11
-5
10
0
-1
9
-12
32
-4
-7
-7
12
-7
0
19
-8
-4
-17
24
3
1

KOL_BARU
-21
-9
...
-22
-7
-29
-32
-21
-30
-2
-25
-7
-16
-7
0
10
-20
-15
-7
-38
-24
0
-12
-10
-11
-11
-15
NA
NA
NA
NA
NA
NA

`summarise()`

```
summarise(flights, rata2wktTerbang = mean(air_time, na.rm
= T))
# sama seperti fungsi agregate di R
```

RATA2WKTTERBANG
150.6865

```
summarise(flights, JmlhwktTerbang = sum(air_time, na.rm =
T))
```

JMLHWKTTERBANG
49326610

`sample_n()` dan `sample_frac()`

```
sample_n(flights, 3) # mensampel 3 baris acak
```

YEAR	MONTH	DAY	DEP_TIME	SCHED_DEP_TIME	DEP_DELAY	ARR_TIME	SCHED_ARR_TIME	ARR_DELAY	CARRIER

YEAR	MONTH	DAY	DEP_TIME	SCHED_DEP_TIME	DEP_DELAY	ARR_TIME	SCHED_ARR_TIME	ARR_DELAY	CARRIER
2013	8	22	552	600	-8	647	700	-13	US
2013	4	5	1433	1345	48	1813	1700	73	AA
2013	10	29	836	835	1	1100	1050	10	EV

```
sample_frac(flights, 0.01) # mensampel 1 % dari seluruh
baris
```

YEAR	MONTH	DAY	DEP_TIME	SCHED_DEP_TIME	DEP_DELAY	ARR_TIME	SCHED_ARR_TIME	ARR_DELAY	CARRIER
2013	8	3	745	752	-7	856	913	-17	B6
2013	10	19	653	700	-7	953	1003	-10	B6
2013	7	17	2107	2030	37	2224	2211	13	9E
2013	4	4	1059	930	89	1421	1255	86	UA
2013	5	13	1825	1829	-4	2042	2031	11	US
2013	9	15	620	625	-5	835	850	-15	MQ
2013	5	25	1609	1557	12	1930	1908	22	DL
2013	9	1	1550	1600	-10	1819	1849	-30	B6
2013	3	1	703	650	13	846	858	-12	EV
2013	5	30	906	900	6	1146	1210	-24	UA
2013	11	18	955	1000	-5	1326	1333	-7	DL
2013	4	11	1415	1415	0	1611	1610	1	MQ
2013	9	2	603	611	-8	714	722	-8	EV
2013	5	6	1052	1055	-3	1156	1228	-32	UA
2013	10	28	1823	1725	58	2050	2019	31	UA
2013	8	9	2216	2040	96	2344	2154	110	B6
2013	10	6	1747	1732	15	2035	1959	36	FL
2013	2	4	1624	1630	-6	1836	1838	-2	DL
2013	6	17	1956	1959	-3	2123	2140	-17	DL
2013	10	29	1321	1257	24	1425	1414	11	EV
2013	12	24	941	945	-4	1153	1202	-9	EV
2013	4	29	2146	2130	16	35	16	19	B6
2013	3	4	1647	1630	17	1953	1954	-1	B6
2013	7	17	1920	1930	-10	2044	2051	-7	EV
2013	1	15	NA	1359	NA	NA	1656	NA	UA
2013	2	5	1453	1500	-7	1711	1655	16	MQ
2013	1	11	1557	1600	-3	1720	1712	8	US

YEAR	MONTH	DAY	DEP_TIME	SCHED_DEP_TIME	DEP_DELAY	ARR_TIME	SCHED_ARR_TIME	ARR_DELAY	CARRIER
2013	11	16	2108	2059	9	2237	2238	-1	EV
2013	6	11	2013	2015	-2	2240	2308	-28	B6
2013	5	24	758	759	-1	956	1020	-24	EV
…	…	…	…	…	…	…	…	…	…
2013	7	14	825	830	-5	1008	1028	-20	EV
2013	3	28	1535	1540	-5	1736	1820	-44	DL
2013	7	20	2008	2005	3	2317	2300	17	UA
2013	3	29	1952	1955	-3	2230	2240	-10	EV
2013	9	22	1255	1300	-5	1415	1415	0	US
2013	10	6	837	840	-3	1001	955	6	UA
2013	9	24	814	820	-6	1056	1110	-14	B6
2013	2	17	1947	1910	37	2256	2228	28	B6
2013	3	15	1514	1456	18	1813	1759	14	UA
2013	9	11	2334	2125	129	37	2247	110	EV
2013	8	18	839	843	-4	1126	1135	-9	UA
2013	5	30	1651	1529	82	1857	1740	77	EV
2013	10	17	1832	1820	12	2140	2125	15	UA
2013	8	12	1859	1827	32	2047	2033	14	US
2013	2	5	1342	1330	12	1658	1616	42	B6
2013	4	10	1703	1655	8	2129	2010	79	AA
2013	6	25	1336	1100	156	1606	1349	137	DL
2013	9	3	957	1000	-3	1218	1237	-19	DL
2013	7	25	746	745	1	929	1000	-31	9E
2013	3	29	1815	1815	0	2123	2147	-24	B6
2013	5	4	1403	1359	4	1509	1515	-6	EV
2013	4	7	1556	1600	-4	1829	1847	-18	DL
2013	11	7	1557	1610	-13	1810	1819	-9	EV
2013	7	11	801	800	1	1044	1104	-20	DL
2013	6	18	1313	1253	20	1553	1520	33	UA
2013	3	5	745	600	105	1006	825	101	MQ
2013	7	2	1258	1144	74	1433	1316	77	EV
2013	8	21	2144	2152	-8	2240	2258	-18	EV
2013	10	16	752	746	6	1027	1021	6	UA
2013	4	30	2150	2155	-5	2247	2312	-25	UA

Operator *pipe*

```
df <- mtcars
```

```
head(df)
```

	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

Nesting

```
filter(df, mpg > 20)
```

	MPG	CYL	DISP	HP	DRAT	WT	QSEC	VS	AM	GEAR	CARB
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

```
sample_n(filter(df, mpg > 20), size = 5) # dua operasi
secara bersamaan
```

	MPG	CYL	DISP	HP	DRAT	WT	QSEC	VS	AM	GEAR	CARB
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1

```
hasil <- arrange(sample_n(filter(df, mpg > 20), size = 5),
desc(mpg))
# memfilter df untuk mpg > 20
# mengambil 5 baris sampel acak
# mengurutkannya berdasarkan kolom mpg secara terbalik
(descending order)
hasil
```

	MPG	CYL	DISP	HP	DRAT	WT	QSEC	VS	AM	GEAR	CARB
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4

```
# susah dibaca!!!
```

Penugasan berganda

```
a <- filter(df, mpg > 20)
b <- sample_n(a, size = 5)
hasil <- arrange(b, desc(mpg))
hasil
```

	MPG	CYL	DISP	HP	DRAT	WT	QSEC	VS	AM	GEAR	CARB
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1

Operator *pipe*

```
hasil <- df %>% filter(mpg > 20) %>% sample_n(size = 5)
%>% arrange(desc(mpg))
# lebih mudah diBACA!
hasil
```

	MPG	CYL	DISP	HP	DRAT	WT	QSEC	VS	AM	GEAR	CARB
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4

tidyr

- dplyr → manipulasi data.
- tidyr → pembersihan data.

```
library(tidyr)
library(data.table) # punya kelebihan soal waktu eksekusi
dibandingkan data frame
```

```
Attaching package: ‘data.table’

The following objects are masked from ‘package:dplyr’:

    between, first, last
```

`gather()`

Membagi kolom menjadi pasangan *key-value*

```
v <- c('A', 'B', 'C','A', 'B', 'C','A', 'B', 'C')
thn <- c(2020,2021,2022,2020,2021,2022,2020,2021,2022)

q1 <- runif(n = 9, min = 0, max = 100)
q2 <- runif(n = 9, min = 0, max = 100)
q3 <- runif(n = 9, min = 0, max = 100)
q4 <- runif(n = 9, min = 0, max = 100)

df <- data.frame(Perusahaan = v, Tahun = thn, q1, q2, q3,
q4)
df
```

PERUSAHAAN	TAHUN	Q1	Q2	Q3	Q4
A	2020	71.401874	34.04511	75.01010	35.93238
B	2021	6.254729	37.22828	58.15716	5.82554
C	2022	58.959634	87.88993	43.93756	14.01085
A	2020	60.140881	69.06750	33.05298	69.82548
B	2021	12.791612	76.31263	66.74539	73.95468
C	2022	77.511686	15.82185	32.11450	84.96146
A	2020	31.628151	45.47135	77.67680	77.78378
B	2021	87.868777	72.97178	55.51417	39.13752
C	2022	99.692759	15.36461	73.52894	42.38900

```
gather(data = df, key = 'Kuarter', value = 'Keuntungan',
q1:q4)
# 4 kolom dibuat jadi 2 kolom
```

PERUSAHAAN	TAHUN	KUARTER	KEUNTUNGAN
A	2020	q1	71.401874
B	2021	q1	6.254729
C	2022	q1	58.959634
A	2020	q1	60.140881
B	2021	q1	12.791612
C	2022	q1	77.511686
A	2020	q1	31.628151
B	2021	q1	87.868777
C	2022	q1	99.692759
A	2020	q2	34.045112
B	2021	q2	37.228278
C	2022	q2	87.889935
A	2020	q2	69.067497
B	2021	q2	76.312626
C	2022	q2	15.821849

PERUSAHAAN	TAHUN	KUARTER	KEUNTUNGAN
A	2020	q2	45.471355
B	2021	q2	72.971783
C	2022	q2	15.364613
A	2020	q3	75.010101
B	2021	q3	58.157156
C	2022	q3	43.937558
A	2020	q3	33.052977
B	2021	q3	66.745387
C	2022	q3	32.114503
A	2020	q3	77.676802
B	2021	q3	55.514168
C	2022	q3	73.528938
A	2020	q4	35.932385
B	2021	q4	5.825540
C	2022	q4	14.010853
A	2020	q4	69.825478
B	2021	q4	73.954683
C	2022	q4	84.961463
A	2020	q4	77.783779
B	2021	q4	39.137522
C	2022	q4	42.389005

```
# menggunakan fungsi gather() dengan menggunakan operator
pipe

df %>% gather(key = 'Kuarter', value = 'Keuntungan',
q1:q4)
```

PERUSAHAAN	TAHUN	KUARTER	KEUNTUNGAN
A	2020	q1	71.401874
B	2021	q1	6.254729
C	2022	q1	58.959634
A	2020	q1	60.140881
B	2021	q1	12.791612
C	2022	q1	77.511686
A	2020	q1	31.628151
B	2021	q1	87.868777
C	2022	q1	99.692759
A	2020	q2	34.045112
B	2021	q2	37.228278
C	2022	q2	87.889935
A	2020	q2	69.067497
B	2021	q2	76.312626
C	2022	q2	15.821849
A	2020	q2	45.471355
B	2021	q2	72.971783
C	2022	q2	15.364613
A	2020	q3	75.010101
B	2021	q3	58.157156
C	2022	q3	43.937558
A	2020	q3	33.052977
B	2021	q3	66.745387
C	2022	q3	32.114503
A	2020	q3	77.676802
B	2021	q3	55.514168
C	2022	q3	73.528938
A	2020	q4	35.932385
B	2021	q4	5.825540
C	2022	q4	14.010853

PERUSAHAAN	TAHUN	KUARTER	KEUNTUNGAN
A	2020	q4	69.825478
B	2021	q4	73.954683
C	2022	q4	84.961463
A	2020	q4	77.783779
B	2021	q4	39.137522
C	2022	q4	42.389005

`spread()`

```
saham <- data.frame(
    waktu = as.Date('2009-01-01') + 0:9,
    X = rnorm(10,0,1),
    Y = rnorm(10,0,2),
    Z = rnorm(10,0,4)
)
```

```
saham
```

WAKTU	X	Y	Z
2009-01-01	-0.9984820	-0.9431436	-3.33749766
2009-01-02	1.2059759	1.5118200	1.06892403
2009-01-03	-2.2298649	1.0133796	4.37434315
2009-01-04	-0.1377994	-1.5819430	0.52897651
2009-01-05	-0.4644349	-2.3425200	-0.02850392
2009-01-06	0.6431310	-3.0102303	-4.28047760
2009-01-07	-0.9540437	2.1494300	0.59964915
2009-01-08	-0.4403926	0.4593915	-4.46067291
2009-01-09	-1.2862645	1.2761994	1.99288550
2009-01-10	0.3102627	-1.6672000	10.22607936

```
saham.gather <- gather(saham, key = saham, value = harga,
X,Y,Z)
saham.gather
```

WAKTU	SAHAM	HARGA
2009-01-01	X	-0.99848196
2009-01-02	X	1.20597592
2009-01-03	X	-2.22986494
2009-01-04	X	-0.13779945
2009-01-05	X	-0.46443485
2009-01-06	X	0.64313100
2009-01-07	X	-0.95404366
2009-01-08	X	-0.44039255
2009-01-09	X	-1.28626448
2009-01-10	X	0.31026273
2009-01-01	Y	-0.94314360
2009-01-02	Y	1.51182002
2009-01-03	Y	1.01337955
2009-01-04	Y	-1.58194299
2009-01-05	Y	-2.34251996
2009-01-06	Y	-3.01023026
2009-01-07	Y	2.14943001
2009-01-08	Y	0.45939146
2009-01-09	Y	1.27619940
2009-01-10	Y	-1.66720001
2009-01-01	Z	-3.33749766
2009-01-02	Z	1.06892403
2009-01-03	Z	4.37434315

WAKTU	SAHAM	HARGA
2009-01-04	Z	0.52897651
2009-01-05	Z	-0.02850392
2009-01-06	Z	-4.28047760
2009-01-07	Z	0.59964915
2009-01-08	Z	-4.46067291
2009-01-09	Z	1.99288550
2009-01-10	Z	10.22607936

```
spread(data = saham.gather, key = 'saham', value =
'harga')
# menyebar data hasil gather (kebalikan)
```

WAKTU	X	Y	Z
2009-01-01	-0.9984820	-0.9431436	-3.33749766
2009-01-02	1.2059759	1.5118200	1.06892403
2009-01-03	-2.2298649	1.0133796	4.37434315
2009-01-04	-0.1377994	-1.5819430	0.52897651
2009-01-05	-0.4644349	-2.3425200	-0.02850392
2009-01-06	0.6431310	-3.0102303	-4.28047760
2009-01-07	-0.9540437	2.1494300	0.59964915
2009-01-08	-0.4403926	0.4593915	-4.46067291
2009-01-09	-1.2862645	1.2761994	1.99288550
2009-01-10	0.3102627	-1.6672000	10.22607936

```
# menggunakan spread() dengan operator pipe
saham.gather %>% spread(key = 'saham', value = 'harga')
```

WAKTU	X	Y	Z
2009-01-01	-0.9984820	-0.9431436	-3.33749766
2009-01-02	1.2059759	1.5118200	1.06892403
2009-01-03	-2.2298649	1.0133796	4.37434315
2009-01-04	-0.1377994	-1.5819430	0.52897651
2009-01-05	-0.4644349	-2.3425200	-0.02850392
2009-01-06	0.6431310	-3.0102303	-4.28047760
2009-01-07	-0.9540437	2.1494300	0.59964915
2009-01-08	-0.4403926	0.4593915	-4.46067291
2009-01-09	-1.2862645	1.2761994	1.99288550
2009-01-10	0.3102627	-1.6672000	10.22607936

`separate()`

Memisahkan satu kolom ke banyak kolom.

```
df <- data.frame(kol.baru = c('a.x', 'b.y', 'c.z'))
df
```

KOL.BARU
a.x
b.y
c.z

```
separate(df, kol.baru, c("ABC", "XYZ"))
```

ABC	XYZ
a	x
b	y
c	z

```
df <- data.frame(kol.baru = c('a-x', 'b-y', 'c-z'))
df
```

KOL.BARU
a-x
b-y
c-z

```
separate(df, kol.baru, c("ABC", "XYZ"))
```

ABC	XYZ
a	x
b	y
c	z

```
# Sintaks lengkapnya:
separate(data = df, col = kol.baru, c('Pertama', 'Kedua'),
sep="-")
```

PERTAMA	KEDUA
a	x
b	y
c	z

`unite()`

Merupakan kebalikan dari `separate()`. Digunakan untuk menggabungkan kolom.

```
df1 <- separate(data = df, col = kol.baru, c('Pertama',
'Kedua'), sep="-")
df1
```

PERTAMA	KEDUA
a	x
b	y
c	z

```
# menggabungkan jd 1 kolom
unite(df1, kol.gab.baru, Pertama, Kedua, sep = '-')
```

KOL.GAB.BARU
a-x
b-y
c-z

Visualisasi data

Diagram batang

```
library(dplyr)
df <- read.csv('../data/murders.csv')
head(df)
```

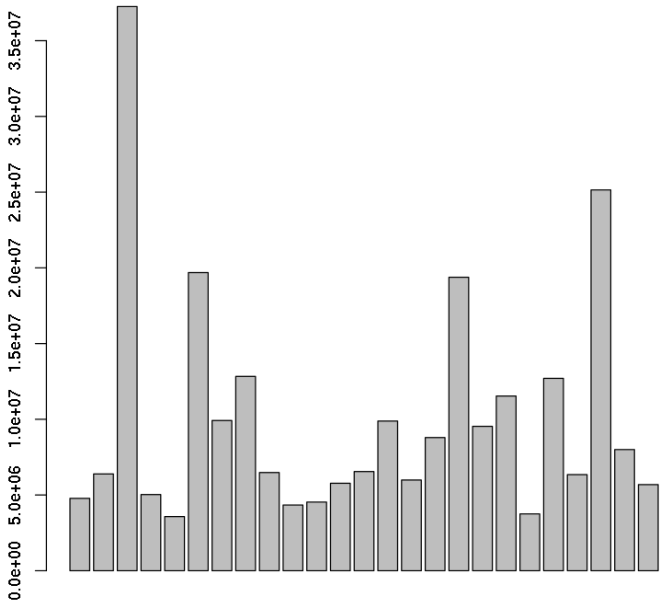
```
Attaching package: ‘dplyr’

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

STATE	ABB	REGION	POPULATION	POPULATIONDENSITY	MURDERS	GUNMURDERS	GUNOW
Alabama	AL	South	4779736	94.65	199	135	0.517
Arizona	AZ	West	6392017	57.05	352	232	0.311
California	CA	West	37253956	244.20	1811	1257	0.213
Colorado	CO	West	5029196	49.33	117	65	0.347
Connecticut	CT	Northeast	3574097	741.40	131	97	0.167
Florida	FL	South	19687653	360.20	987	669	0.245

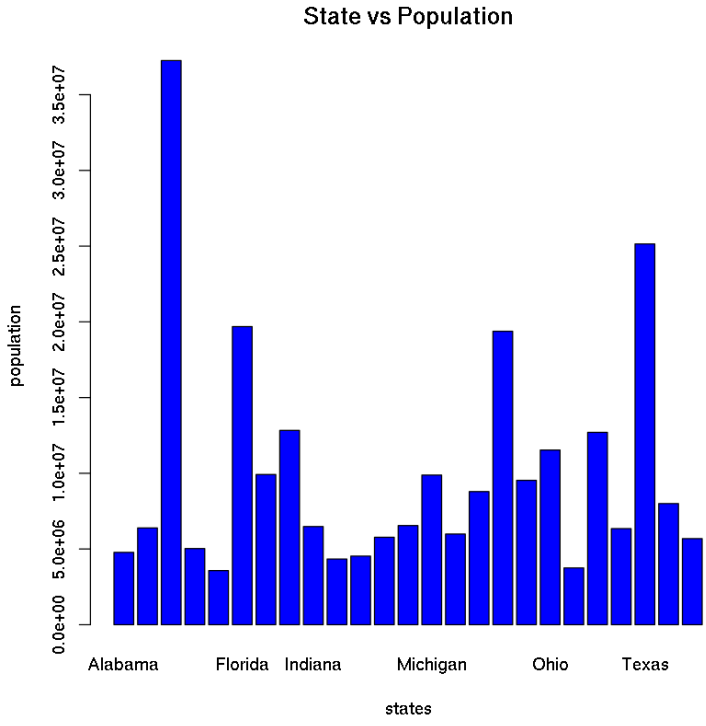
```
subdf <- select(df, state, population, murders)
head(subdf)
```

STATE	POPULATION	MURDERS
Alabama	4779736	199
Arizona	6392017	352
California	37253956	1811
Colorado	5029196	117
Connecticut	3574097	131
Florida	19687653	987

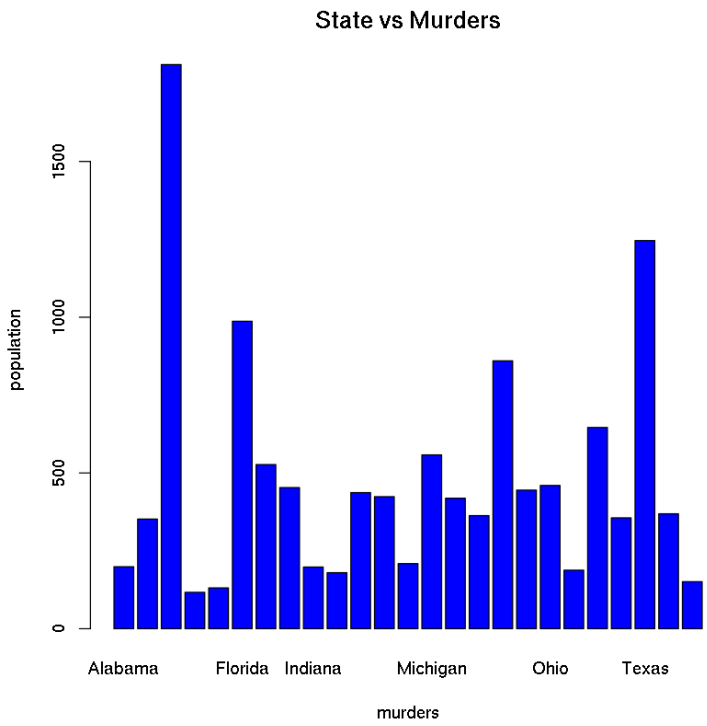
```
barplot(subdf$population)
```




```r
# kostumisasi grafik
barplot(subdf$population,
        xlab='states',
        ylab='population',
        main='State vs Population',
        names.arg = subdf$state,
        col='blue')
```



```r
# kostumisasi grafik murders
barplot(subdf$murders,
       xlab='murders',
       ylab='population',
       main='State vs Murders',
       names.arg = subdf$state,
       col='blue')
```




```r
# mengurutkan df berdasarkan angka pembunuhan (secara terbalik)
dfsort <- arrange(df, desc(murders))

# menseleksi kolom - kolom tertentu
subdfsort <- select(dfsort, state, population, murders)

# Mengambil 5 data tertinggi untuk kasus pembunuhan
topsubdfsort <- head(subdfsort,5)

# PLOT!!!
barplot(topsubdfsort$murders,
       xlab='murders',
       ylab='population',
       main='State vs Murders',
       names.arg = topsubdfsort$state,
       col='blue')
```

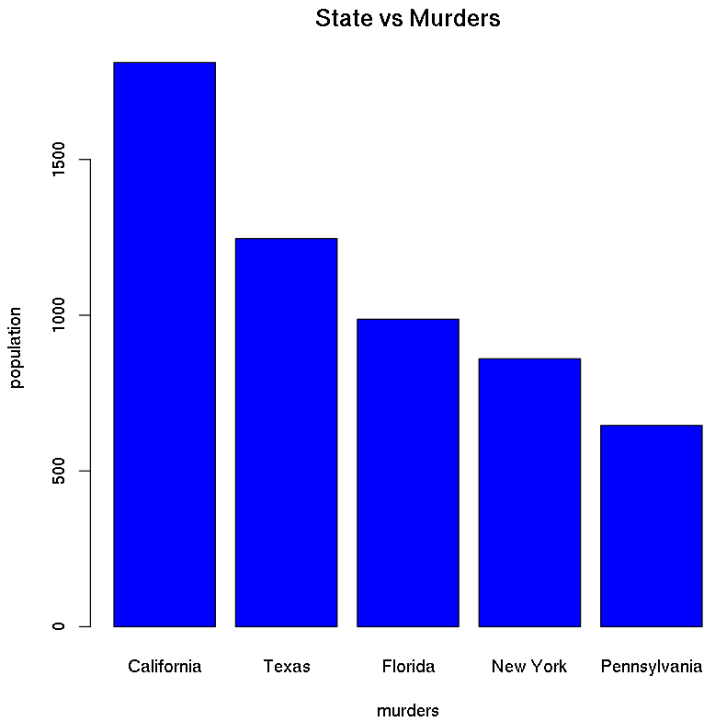


Diagram batang horizontal

```
df <- read.csv("../data/murdersmini.csv")
df
```

STATE	POPULATION	MURDERS
Arizona	6392017	352
Colorado	5029196	117
Georgia	9920000	527
Iowa	3046355	38
Kansas	2853118	100
Maine	1328361	24
Michigan	9883640	558
New York	19378102	860
Texas	25145561	1246
Washington	6724540	151

```
seldf <- select(df, state, murders)
barplot(seldf$murders, horiz=T,
       xlab='Murders', ylab='States',
       main = 'States vs Murders',
       col='blue',
       names.arg = seldf$state)
```

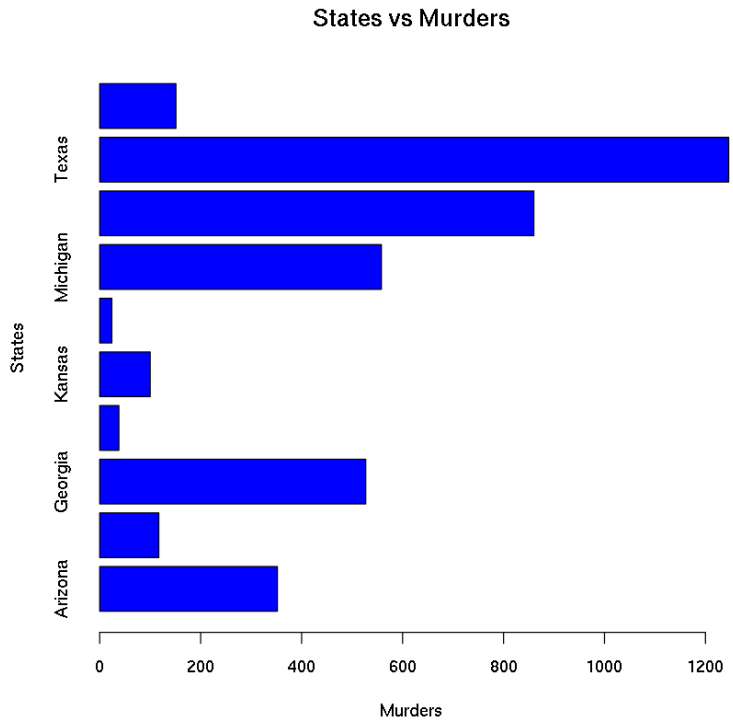


Diagram batang bertumpuk

```
df
```

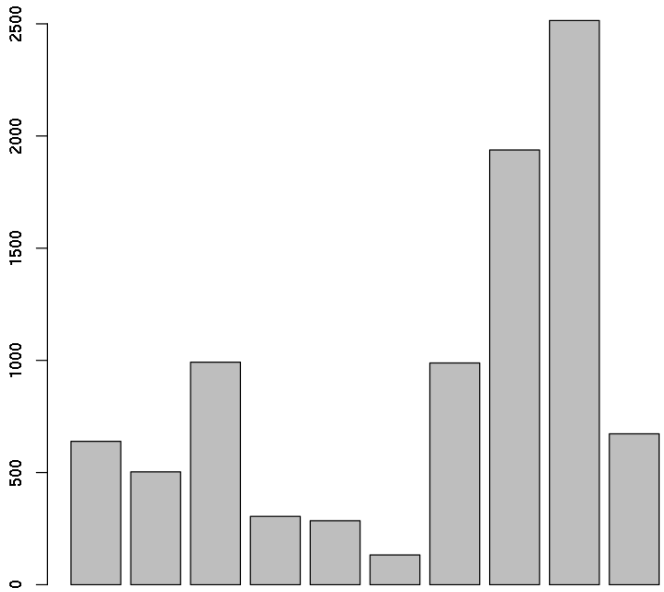
STATE	POPULATION	MURDERS
Arizona	6392017	352
Colorado	5029196	117
Georgia	9920000	527
Iowa	3046355	38
Kansas	2853118	100
Maine	1328361	24
Michigan	9883640	558
New York	19378102	860
Texas	25145561	1246
Washington	6724540	151

```
dfs <- mutate(df,pop = population / 10000)
```

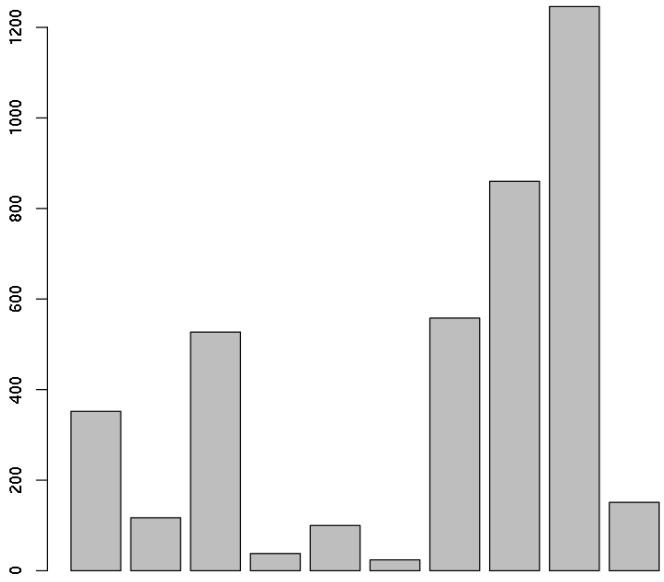
```
names(dfs)
```

1. 'state'
2. 'population'
3. 'murders'
4. 'pop'

```r
dfs <- dfs[c(1,3,4)]
barplot(dfs$pop)
```



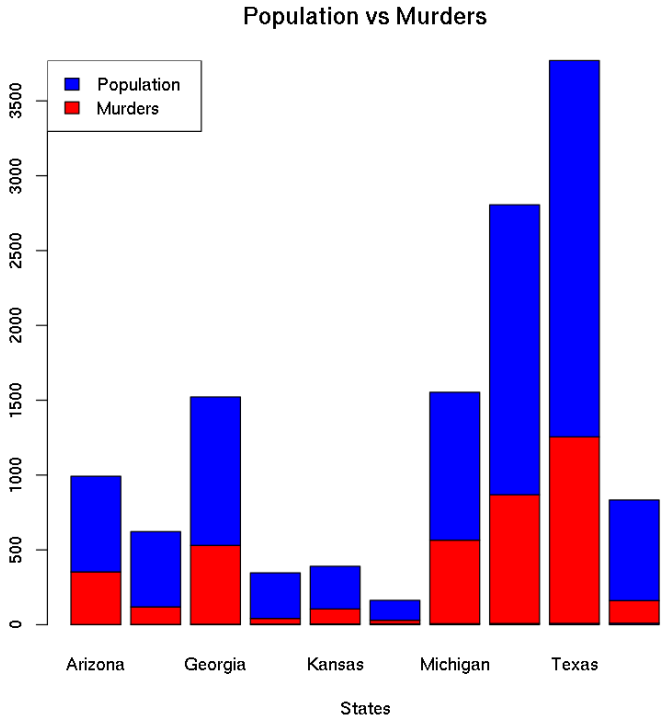

```r
barplot(dfs$murders)
```




```r
mat <- data.matrix(dfs)
mat <- t(mat) # transpos
mat
```

state	1.0000	2.0000	3	4.0000	5.0000	6.0000	7.000	8.00	9.000	10.000
murders	352.0000	117.0000	527	38.0000	100.0000	24.0000	558.000	860.00	1246.000	151.000
pop	639.2017	502.9196	992	304.6355	285.3118	132.8361	988.364	1937.81	2514.556	672.454

```
barplot(mat,
       xlab='States',
       main='Population vs Murders',
       col=c('blue', 'red'),
       names.arg=dfs$state)
legend('topleft', c('Population', 'Murders'),
fill=c('blue', 'red'))
```



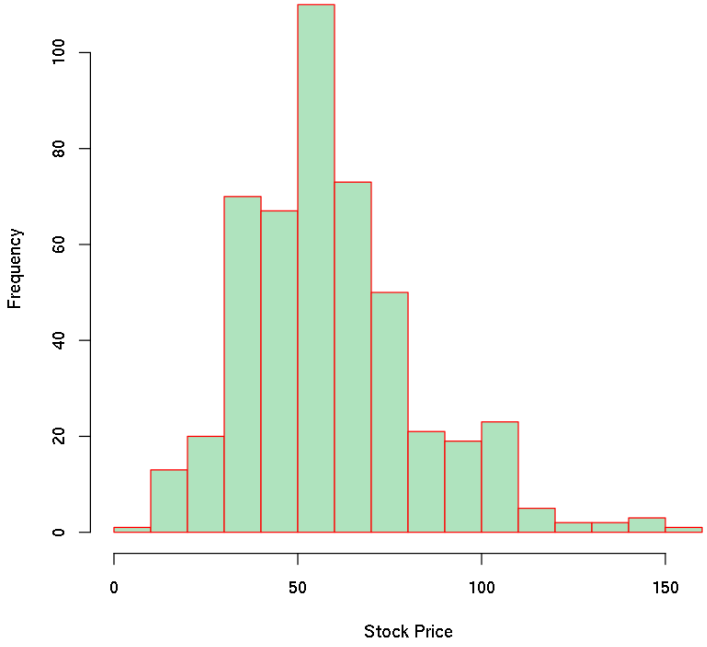
Histogram

```
df <- read.csv('../data/GEStock.csv')
head(df)
```

DATE	PRICE
1/1/70	74.25333
2/1/70	69.97684
3/1/70	72.15857
4/1/70	74.25273
5/1/70	66.66524
6/1/70	67.59318

```
subdf <- select(df, Date, Price)
```

```
hist(subdf$Price,
    xlab='Stock Price',
    main='',
    col='#afe3be',
    border='red',
    breaks = 20) # secara default bins=10
```



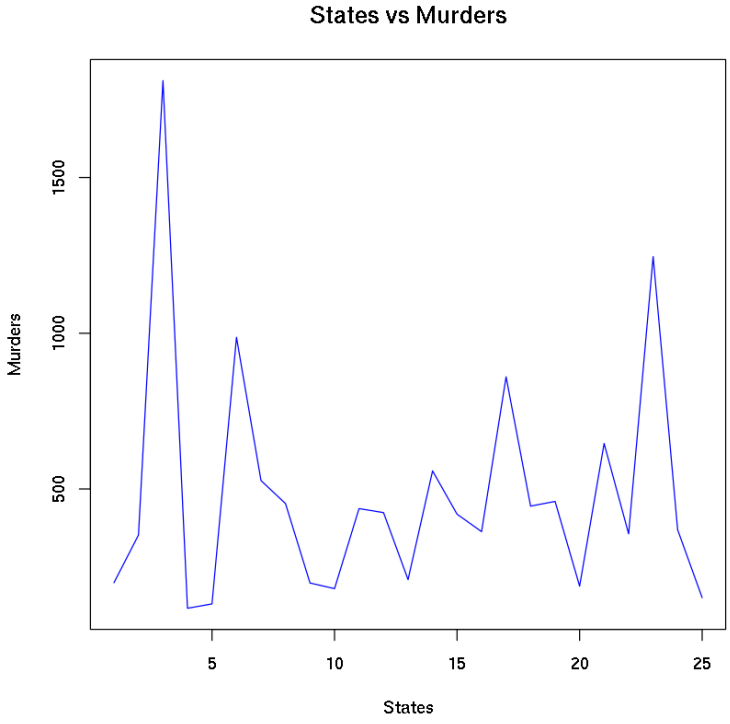
Scatterplot

```
df <- read.csv("../data/murders.csv")
df <- select(df,state,population,murders)
```

```
plot(df$population, df$murders,
    xlab='Population', ylab='Murders',
    main='Population vs Murders', col='red',
    pch = 20)
```

Diagram garis

```
plot(df$murders,type='l',
    xlab='States', ylab='Murders',
    main='States vs Murders',
    col='blue')
```

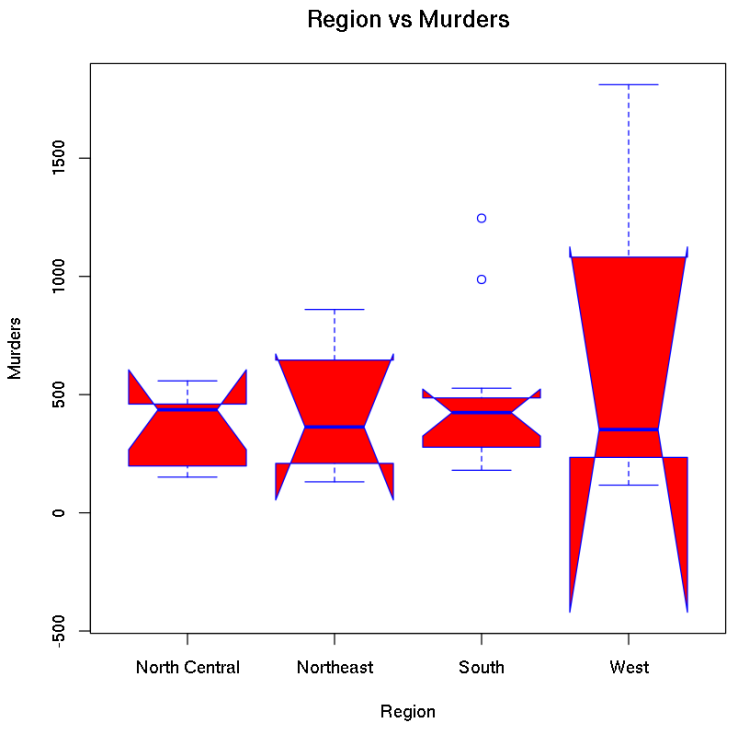


Boxplot

```
df <- read.csv('../data/murders.csv')
df <- select(df, state, population,murders, region)
```

```
boxplot(df$murders ~ df$region,
       xlab='Region', ylab='Murders',
       main='Region vs Murders',
       col='red', border='blue',
       notch=T) # dipisahkan berdasarkan region
```

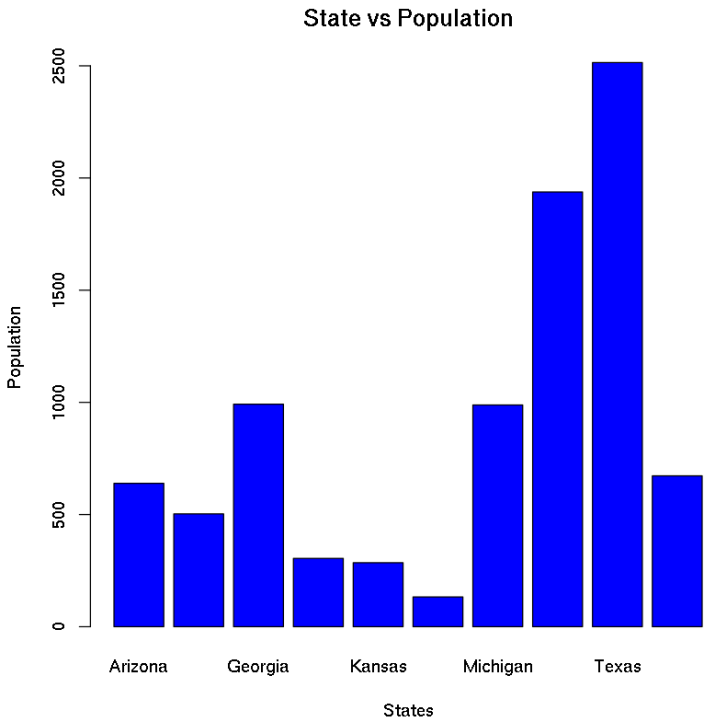
```
Warning message in bxp(list(stats = structure(c(151, 198,
436, 460, 558, 131, 209, :
“some notches went outside hinges ('box'): maybe set
notch=FALSE”
```



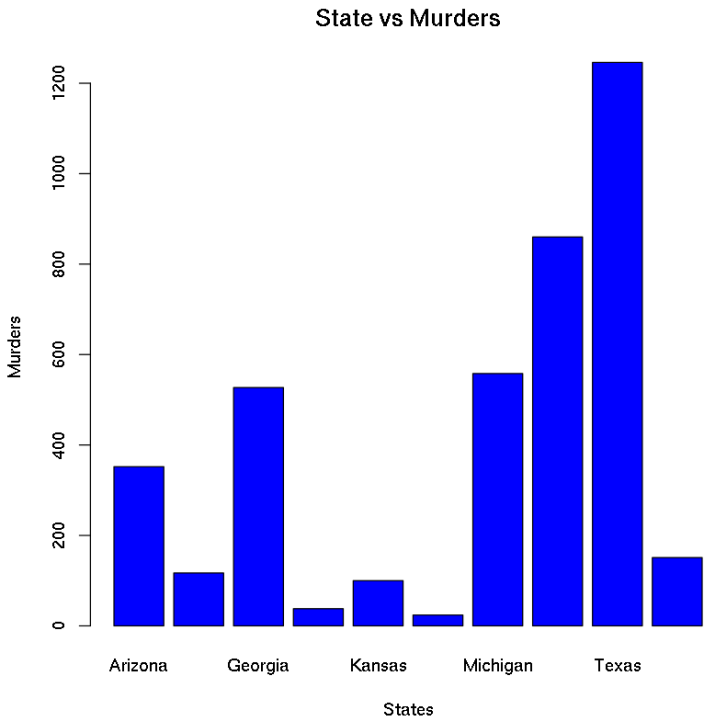
Kombinasi plot

```
df <- read.csv("../data/murdersmini.csv")
df <- mutate(df, pop = population/10000)
df <- df[c(1,3,4)] # seleksi kolom 1, 3, dan 4
```

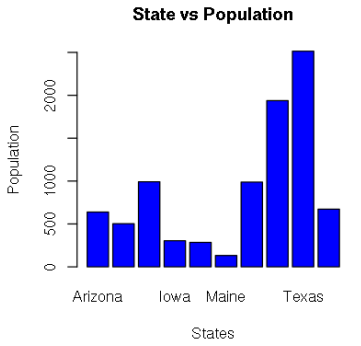
```
barplot(df$pop, xlab='States', ylab='Population',
        main='State vs Population', col='blue',
names.arg=df$state)
```

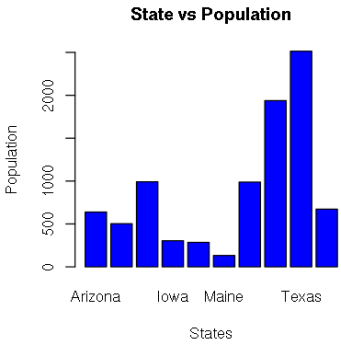



```
barplot(df$murders, xlab='States', ylab='Murders',
        main='State vs Murders', col='blue',
       names.arg=df$state)
```



```r
# Supaya tidak jadi dua plot:
par(mfrow=c(2,2)) # 2 baris, 2 kolom
barplot(df$pop, xlab='States', ylab='Population',
        main='State vs Population', col='blue',
names.arg=df$state)
barplot(df$pop, xlab='States', ylab='Population',
        main='State vs Population', col='blue',
names.arg=df$state)
plot(df$pop, df$murders, xlab='Population',
ylab='Murders',
     col='red',pch=20, main='Population vs Murders')
```





Data kualitatif dan kuantitatif

Pendahuluan

- Di R, data umumnya disimpan dalam bentuk vektor atau data frame.
- Data kualitatif, di dalam statistika dikenal sebagai data kategorikal.
- Data kualitatif dapat disimpan dalam bentuk *Factors*.
- Data kuantitatif, di dalam statistika dikenal sebagai data kontinyu atau data numerik.
- Data kuantitatif dapat disimpan dalam bentuk *Numeric*.

Data kualitatif

Data kualitatif merupakan data non-statistik yang umumnya bersifat tidak terstruktur atau semi-terstruktur.

- Data kualitatif tidak melulu berasal dari pengukuran.
- Data kualitatif dikategorikan berdasarkan sifat - sifat, atribut, label, dll.
- Data ini digunakan untuk interpretasi dan pembuatan hipotesis.
- Data ini tidak dapat dikumpulkan dan dianalisa menggunakan metode - metode konvensional.

Contoh - contoh data kualitatif:

- Jenis kelamin,
- Ukuran sepatu,
- *Rating*.

```
ukuranBaju <- c('S', 'M', 'L', 'XL',
                'XXL', 'M', 'L', 'XL',
                'XXL', 'S', 'M')
ukuranBaju
```

1. 'S'
2. 'M'
3. 'L'
4. 'XL'
5. 'XXL'
6. 'M'
7. 'L'
8. 'XL'
9. 'XXL'
10. 'S'
11. 'M'

```
ukuran_baju <- factor(ukuranBaju) # dijadikan dalam bentuk
Factor
ukuran_baju
```

1. S
2. M
3. L
4. XL
5. XXL
6. M
7. L
8. XL
9. XXL
10. S
11. M

- **Levels**:

```
str(ukuran_baju)
```

```
 Factor w/ 5 levels "L","M","S","XL",..: 3 2 1 4 5 2 1 4 5
3 ...
```

```
summary(ukuran_baju)
```

L
: 2
M
: 3
S
: 2
XL
: 2
XXL
: 2

```
levels(ukuranBaju)
```

```
NULL
```

```
levels(ukuran_baju)
```

1. 'L'
2. 'M'
3. 'S'
4. 'XL'
5. 'XXL'

Visualisasi data kualitatif

Gunakan:

- Diagram batang,
- Diagram lingkaran.

```
ukuran_baju
```

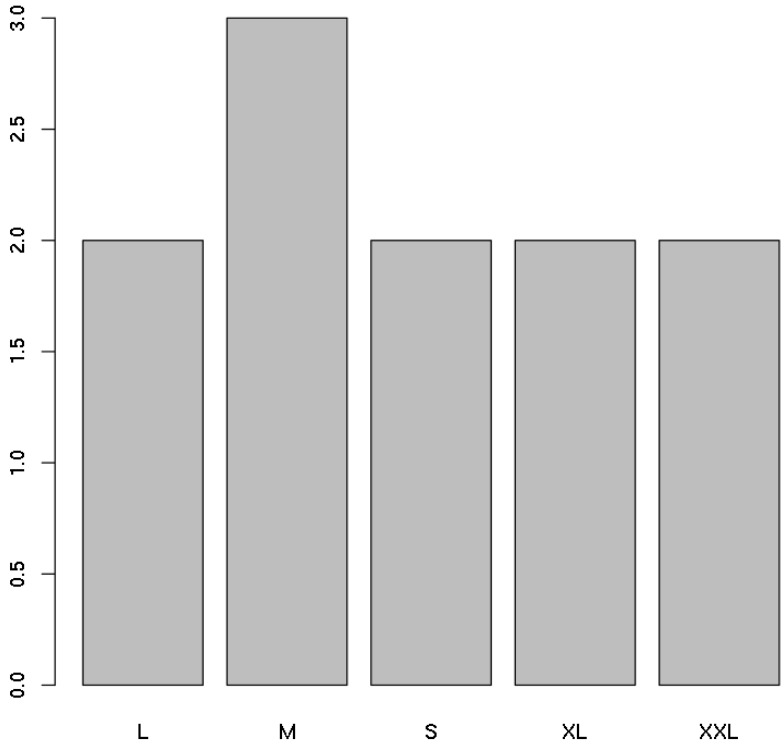
1. S
2. M
3. L
4. XL
5. XXL
6. M
7. L
8. XL
9. XXL
10. S
11. M

▶ **Levels**:

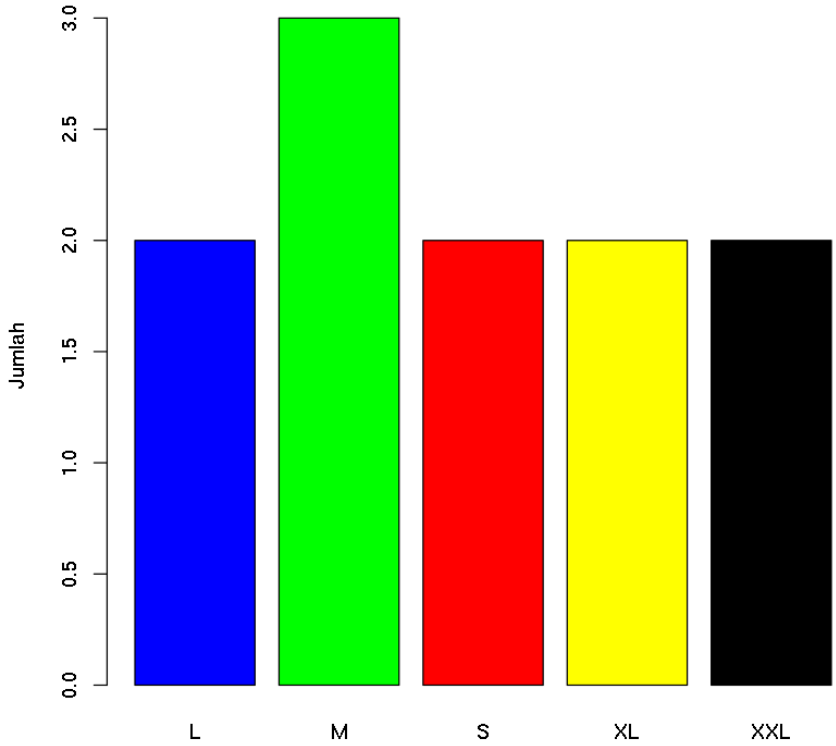
```
tabelUkuranBaju <- table(ukuran_baju)
tabelUkuranBaju
```

```
ukuran_baju
  L   M   S  XL XXL 
  2   3   2   2   2 
```

```
barplot(tabelUkuranBaju)
```




```
# kostumisasi diagram batang
barplot(tabelUkuranBaju,
        col = c('blue', 'green', 'red', 'yellow','black'),
        ylab = 'Jumlah')
```

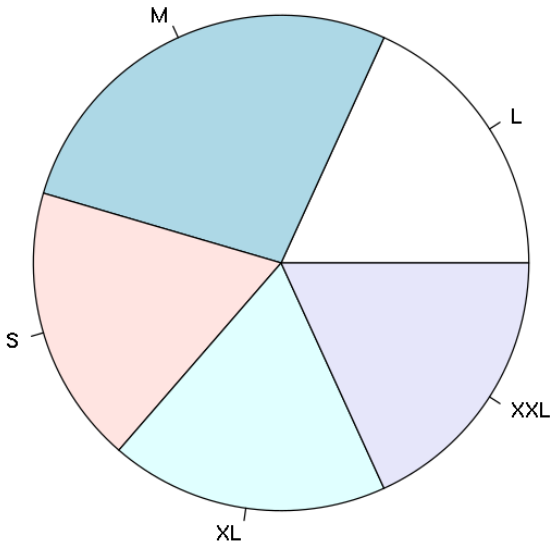



```
# misalkan kita hanya ingin ukuran M
ukuran_baju == 'M'
```

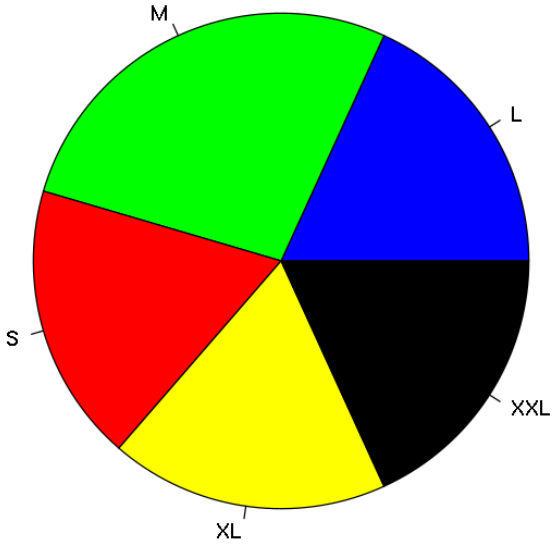
1. FALSE
2. TRUE
3. FALSE
4. FALSE
5. FALSE
6. TRUE
7. FALSE
8. FALSE
9. FALSE
10. FALSE
11. TRUE

```
jmlh_ukuranM = sum(ukuran_baju == 'M')
jmlh_ukuranM
```

```
# Penggunaan diagram lingkaran
pie(tabelUkuranBaju)
```




```
# kostumisasi diagram lingkaran
pie(tabelUkuranBaju, col = c('blue', 'green', 'red',
'yellow','black'))
```




```
kategori_usia <-
factor(c(2,4,3,3,2,1,1,1,2,1,1,4,3,4,2,2,2,1,4,4,4,4,3,2,2
,1))
```

```
levels(kategori_usia)
```

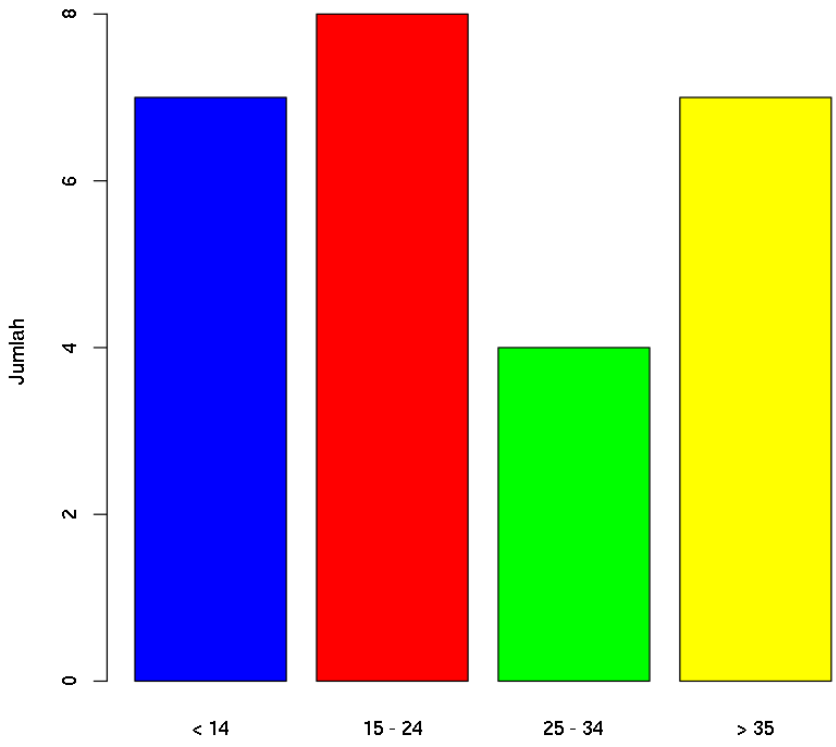
1. '1'
2. '2'
3. '3'
4. '4'

```
# mengubah level kategori usia
levels(kategori_usia) <- c('< 14', '15 - 24', '25 - 34',
'> 35')
```

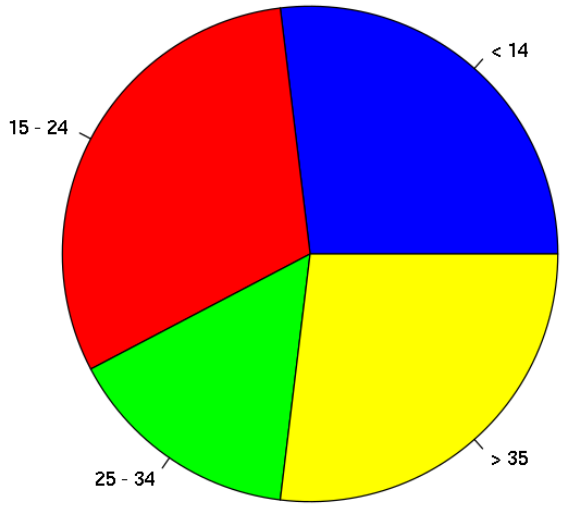
```
tabelKategoriUsia <- table(kategori_usia)
tabelKategoriUsia
```

```
kategori_usia
   < 14 15 - 24 25 - 34    > 35
      7       8       4       7
```

```
barplot(tabelKategoriUsia,
       col=c('blue', 'red', 'green', 'yellow'),
       ylab = 'Jumlah')
```




```
pie(tabelKategoriUsia, col=c('blue', 'red', 'green',
'yellow'))
```



Data kuantitatif

- Data kuantitatif dapat dihitung, diukur, dan diekspresikan secara numerik.
- Data kualitatif sendiri bersifat deskriptif dan konseptual.
- Data kuantitatif bersifat terstruktur.

Contoh:

- Pengukuran temperatur udara,
- Harga saham, dll.

```
# panjang lagu (dalam menit)
lagu <- c(5.3,3.6,5.5,4.7,6.7,4.3,6.2,4.3,4.9,5.1,5.8,4.4)
lagu
```

1. 5.3
2. 3.6
3. 5.5
4. 4.7
5. 6.7
6. 4.3
7. 6.2
8. 4.3
9. 4.9
10. 5.1
11. 5.8

```
```

Visualisasi data kuantitatif

- Histogram
- *Boxplot*
- *strip-chart* → alternatif *boxplot* ketika ukuran sampel kecil.

Histogram

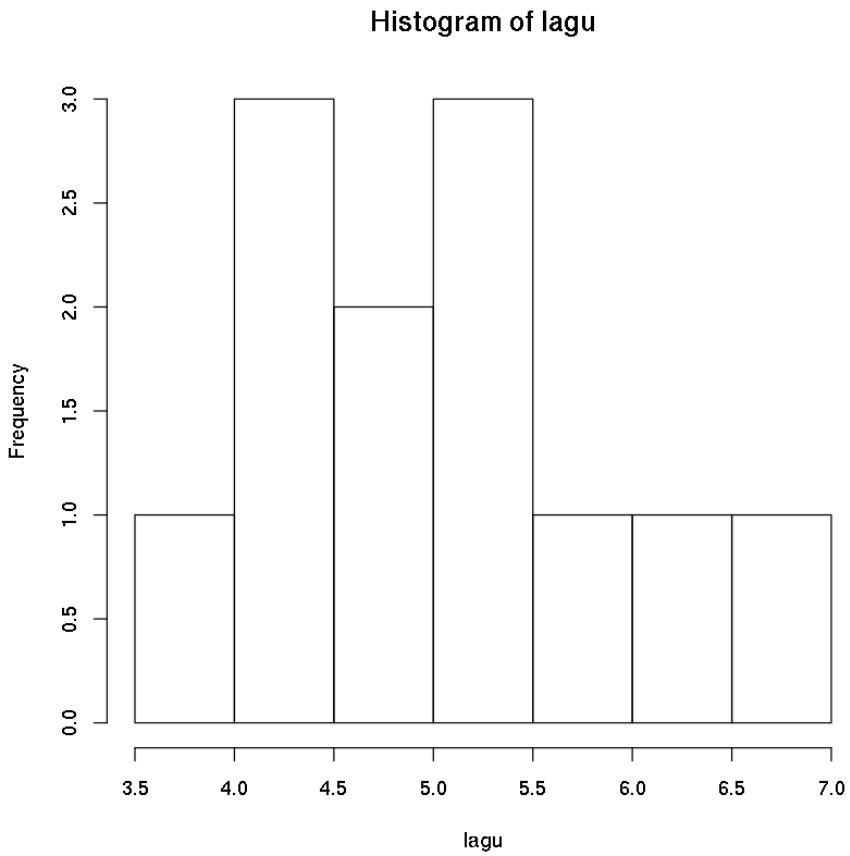
```
length(lagu) # jumlah elemen di dalam vektor lagu
```

```
12
```

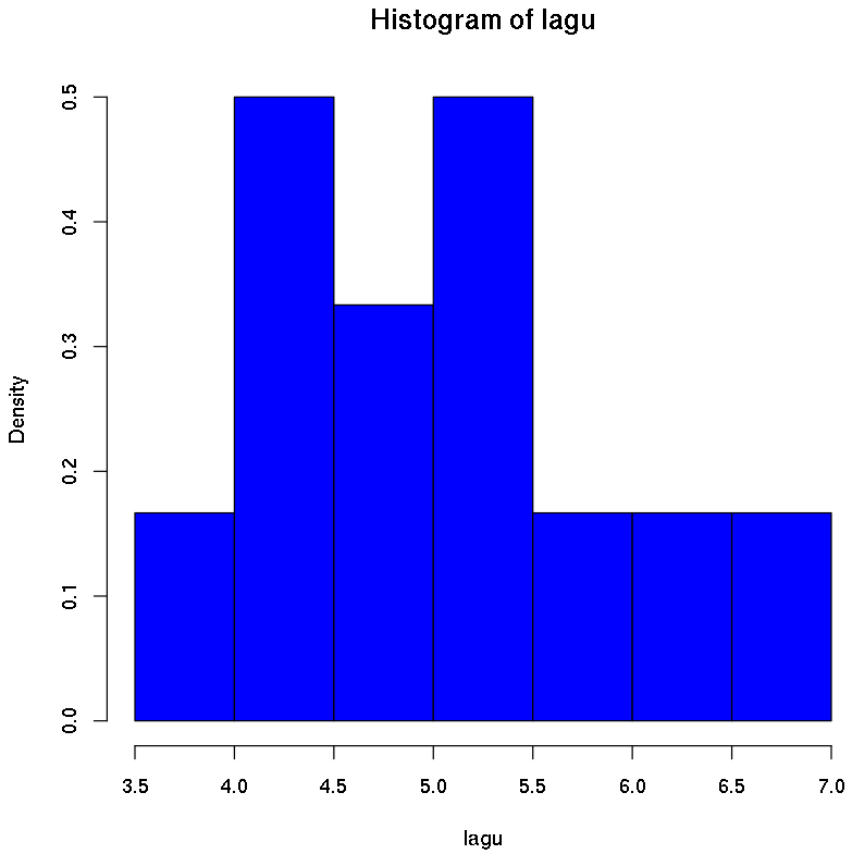
```
summary(lagu)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  3.600   4.375   5.000   5.067   5.575   6.700 
```

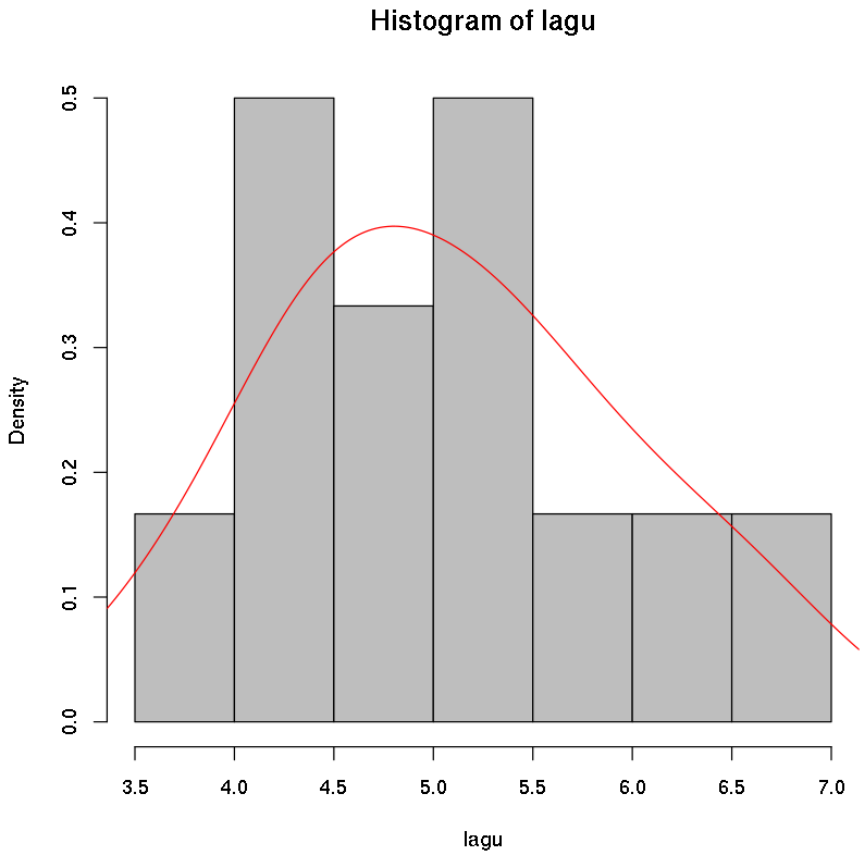
```
hist(lagu)
```




```
hist(lagu, col='blue', prob=T) # pdf: probability density
function
```

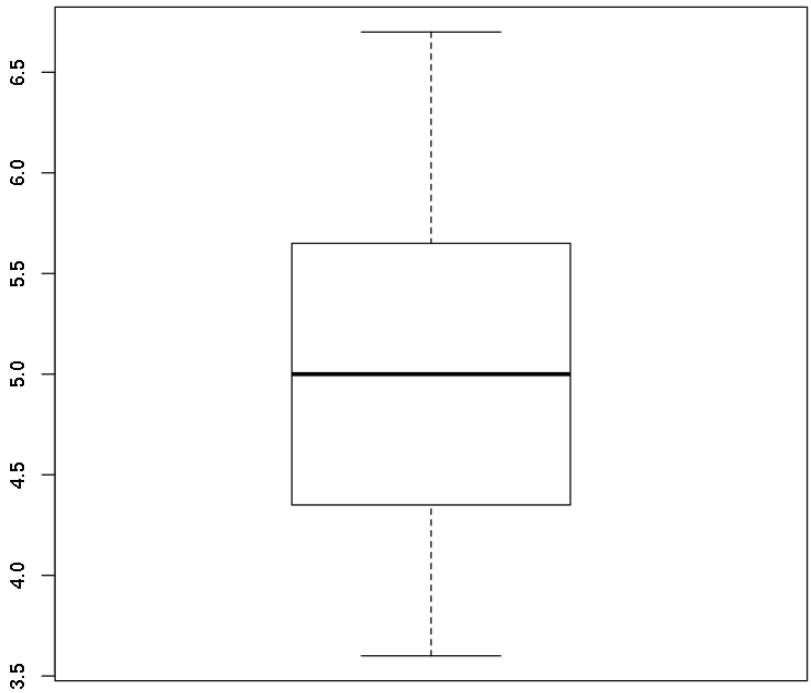


```
hist(lagu, col='grey', prob=T)
lines(density(lagu), col='red')
```



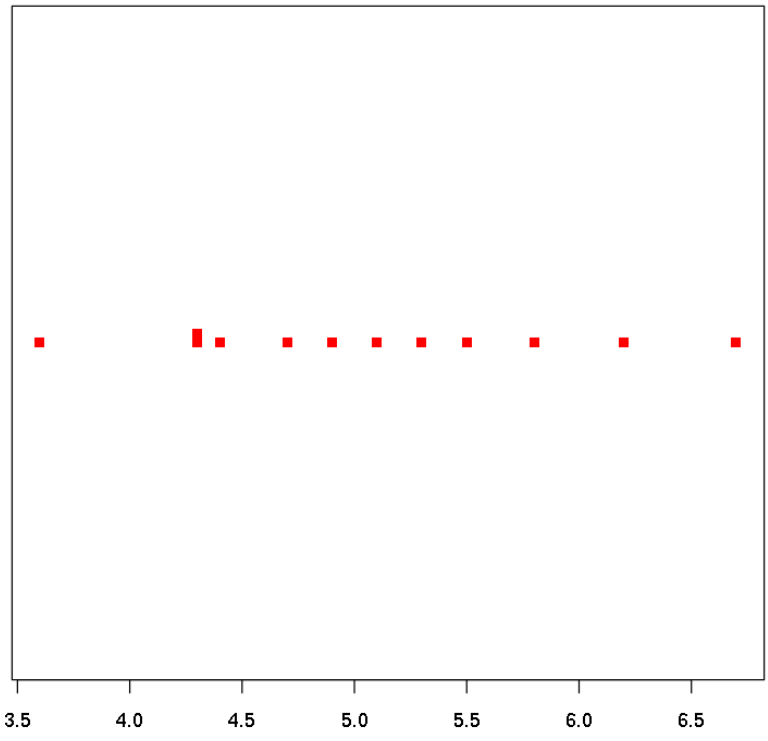
boxplot

```
boxplot(lagu)
```



strip-chart

```
stripchart(lagu,col='red', pch=15, method='stack')
```



Visualisasi data saham

```
library(dplyr)
```

```
Attaching package: ‘dplyr’

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

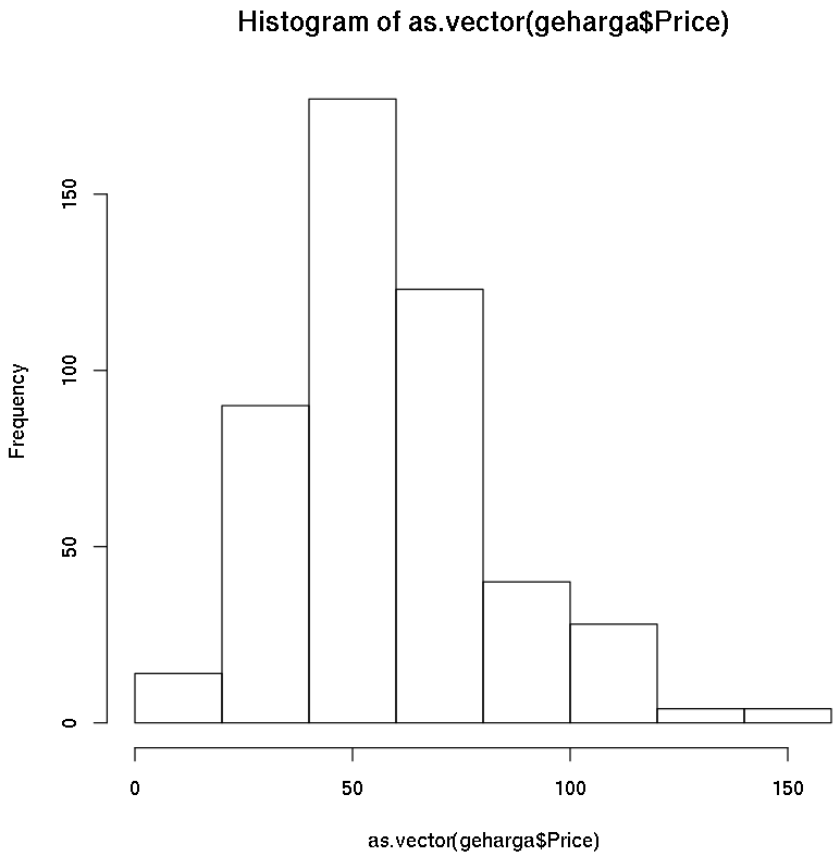
```
gedata <- read.csv("../data/GEStock.csv")
```

```
geharga <- select(gedata, Price)
```

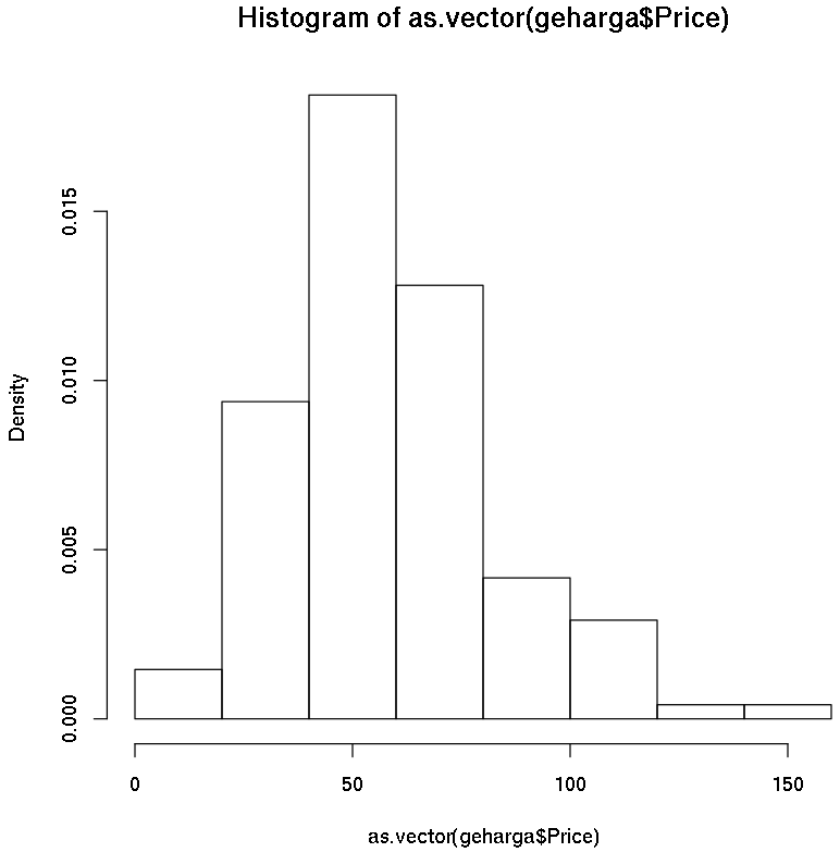
```
summary(geharga)
```

```
     Price
 Min.   :  9.294
 1st Qu.: 44.214
 Median : 55.812
 Mean   : 59.303
 3rd Qu.: 72.226
 Max.   :156.844
```

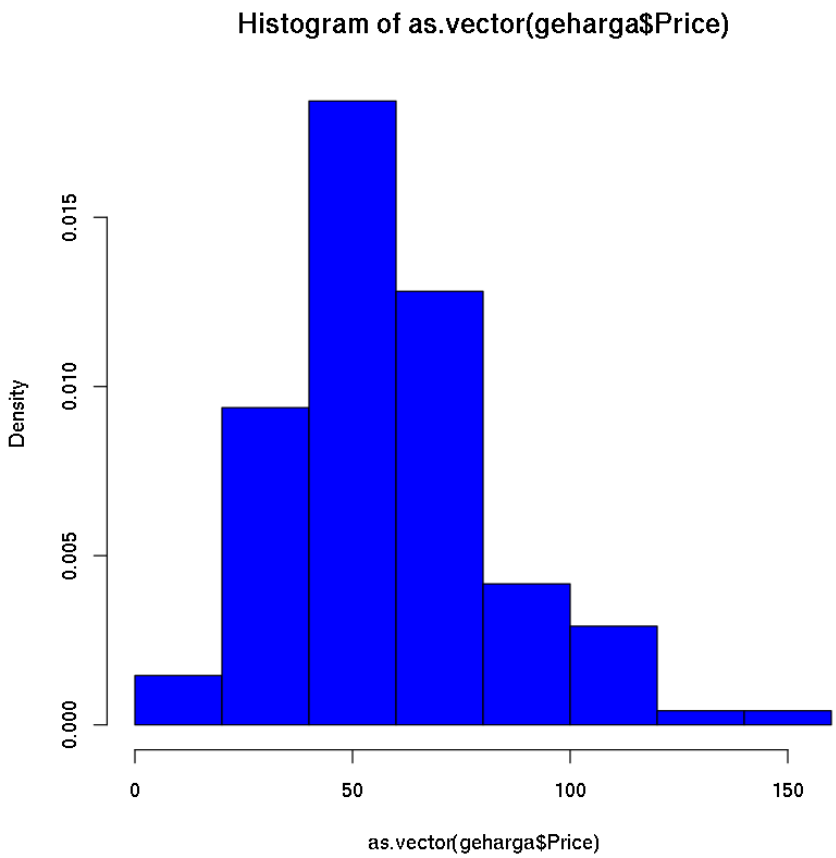
```
hist(as.vector(geharga$Price))
```



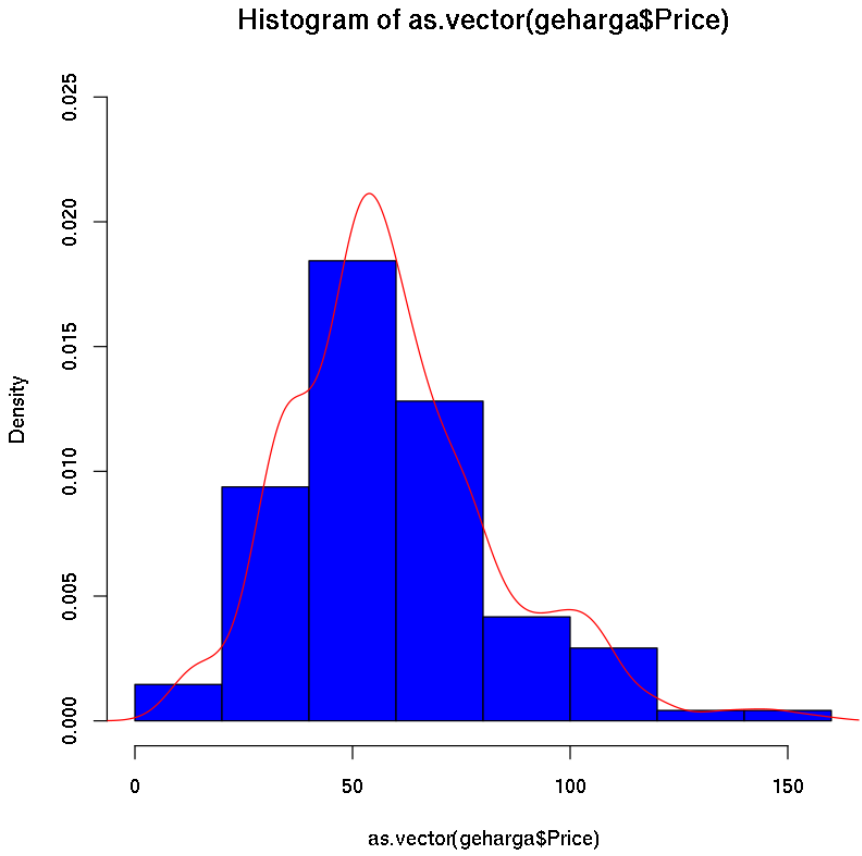

```
hist(as.vector(geharga$Price), prob=T)
```



```
hist(as.vector(geharga$Price), prob=T, col='blue')
```




```
hist(as.vector(geharga$Price), prob=T, col='blue',
ylim=c(0,0.025))
lines(density(geharga$Price), col='red')
```


Histogram of as.vector(geharga$Price)
Density
0.000
0.005
0.010
0.015
0.020
0.025
0
50
100
150
as.vector(geharga$Price)

Statistika Deskriptif

`length()`,`min()`, `max()`, `sum()`, `prod()` dan `sort()`

```
lagu <- c(5.3,3.6,5.5,4.7,6.7,4.3,6.2,4.3,4.9,5.1,5.8,4.4)
lagu
```

1. 5.3
2. 3.6
3. 5.5
4. 4.7
5. 6.7
6. 4.3
7. 6.2
8. 4.3
9. 4.9
10. 5.1
11. 5.8
12. 4.4

```
length(lagu) # banyaknya elemen
```

12

```
max(lagu) # nilai terbesar
```

6.7

```
min(lagu) # nilai terkecil
```

3.6

```
sum(lagu) # total penjumlahan elemen di dalam vektor
```

60.8

```
prod(lagu) # total perkalian elemen di dalam vektor
```

241595726.162817

```
# mengurutkan vektor dari kecil ke besar
sort(lagu)
```

1. 3.6
2. 4.3
3. 4.3
4. 4.4
5. 4.7
6. 4.9
7. 5.1
8. 5.3
9. 5.5
10. 5.8
11. 6.2
12. 6.7

```
# mengurutkan vektor dari besar ke kecil
sort(lagu, decreasing=T)
```

1. 6.7
2. 6.2
3. 5.8
4. 5.5
5. 5.3
6. 5.1
7. 4.9
8. 4.7
9. 4.4
10. 4.3
11. 4.3
12. 3.6

Rata - rata

Jenis rata - rata:

- Rata - rata aritmatika (rata - rata)
- Rata - rata geometri
- Rata - rata harmonik

Rata - rata aritmatika

Rata - rata aritmatika dari suatu sampel adalah penjumlahan dari seluruh sampel, dibagi dengan ukuran sampel.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

```
lagu
```

1. 5.3
2. 3.6
3. 5.5
4. 4.7
5. 6.7
6. 4.3
7. 6.2
8. 4.3
9. 4.9
10. 5.1
11. 5.8
12. 4.4

```
rata2 <- sum(lagu) / length(lagu)
rata2
```

5.06666666666667

```
# pakai fungsi built-in
rata2 <- mean(lagu)
rata2
```

5.06666666666667

Rata - rata geometri

Rata - rata geometri didefinisikan sebagai akar ke-n dari perkalian seluruh sampel.

$$RG = \sqrt[n]{\prod_{i=1}^{n} x_i}$$

```
lagu
```

1. 5.3
2. 3.6
3. 5.5
4. 4.7
5. 6.7
6. 4.3
7. 6.2
8. 4.3
9. 4.9
10. 5.1
11. 5.8
12. 4.4

```
rata2geom <- prod(lagu)^(1/length(lagu))
rata2geom
```

4.99563581610903

```
# Cara yang lebih efisien:
rata2geom <- exp(mean(log(lagu)))
rata2geom
```

4.99563581610903

Aplikasi rata - rata geometri

Umum digunakan di dunia bisnis, misalnya:

- Perhitungan laju pertumbuhan.
- Perhitungan pengembalian portofolio keamanan.

Perhitungan *compounded annual growth rate*. Misalkan ada saham sebuah perusahaan dengan:

- Pertumbuhan sebesar 10 % pada tahun pertama (misalkan awalnya harga saham: $100):
  $100 + 10 = 10$

- Penurunan sebesar 20 % di tahun kedua:
  $100 - 20 = 80$
- Pada tahun ketiga pertumbuhan sebesar 30%:
  $100 + 30 = 130$

Laju pertumbuhan: $\sqrt[3]{110 \times 80 \times 130} = 104,586$

Karena dalam persen, maka:
104.586 - 100 = 4.58 % (laju pertumbuhannya)

```
saham <- c(100 + 10, 100 - 20, 100 + 30)
rg <- exp(mean(log(saham)))
rg
```

104.58643063512

```
rg - 100
```

4.58643063511965

Rata - rata harmonik

Rata - rata harmonik merupakan kebalikan dari rata - rata terbalik dari sampel:

$$\frac{1}{H} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{x_i} \text{ dengan } x_i > 0$$

```
rata2harm <- 1 / mean(1/lagu)
rata2harm
```

4.92500029031758

- Rata - rata harmonik digunakan untuk mencari hubungan perkalian atau pembagian antar pecahan.
- Rata - rata harmonik banyak digunakan untuk merata - ratakan suatu kelajuan.
- Di bidang finansial banyak digunakan untuk menghitung *price-earnings ratio*.

Median dan modus

Median: nilai tengah suatu sampel yang telah diurutkan.

- Pada sampel ganjil:

$$x_{\frac{n+1}{2}}$$

- Pada sampel genap:

$$\frac{x_{\frac{n}{2}} + x_{\frac{n}{2}+1}}{2}$$

Modus merupakan nilai yang paling sering muncul di dalam suatu sampel.

- Suatu sampel dapat mempunyai sebuah modus, lebih dari satu modus, atau tidak mempunyai modus sama sekali.
- Modus dapat mempunyai nilai yang sama dengan rata - rata dan median.

```
sort(lagu)
```

1. 3.6
2. 4.3
3. 4.3
4. 4.4
5. 4.7
6. 4.9
7. 5.1
8. 5.3
9. 5.5
10. 5.8
11. 6.2
12. 6.7

```
median(lagu) # cara menghitung median
```

5

R tidak mempunyai fungsi untuk menghitung modus karena modus jarang digunakan untuk analisis statistik

Pencilan

```
gaji <- c(12,14,18,90,16,19,21) # gaji bulanan dalam juta rupiah
gaji
```

1. 12
2. 14
3. 18
4. 90
5. 16
6. 19
7. 21

```
mean(gaji)
```

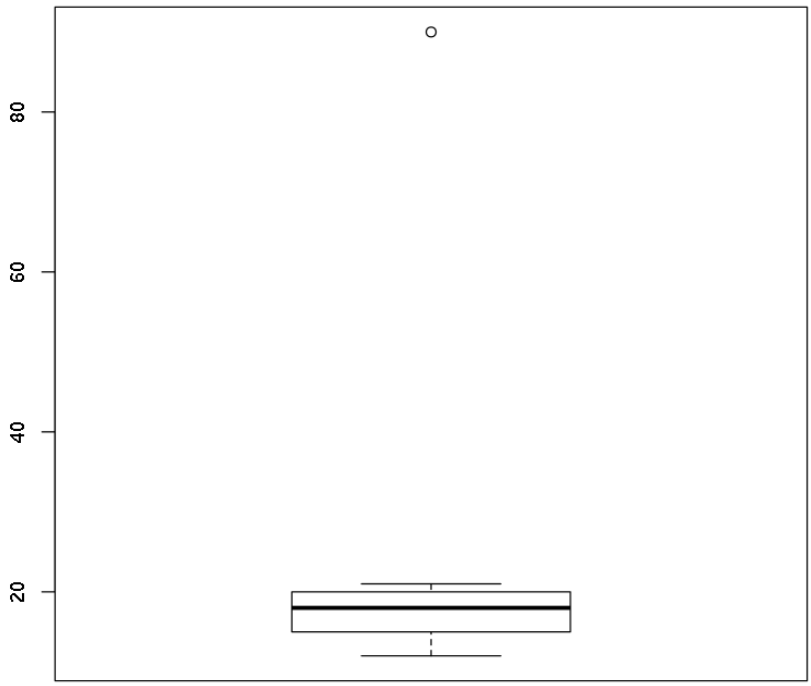
27.1428571428571

```
median(gaji)
```

18

Terdapat perbedaan yang sangat jauh antara rata - rata dan median.

```
boxplot(gaji)
```




```
mean(gaji, trim=0.1)
```

27.1428571428571

```
mean(gaji, trim=0.5)
```

18

```
mean(gaji, trim=0.2)
```

17.6

Kuartil dan kuantil

Kuartil merupakan istilah statistik yang digunakan untuk mendeskripsikan pembagian selang data ke dalam empat interval.

- Kuartil memisahkan data ke dalam tiga titik, yakni kuartil bawah, median, dan kuartil atas.

- Kuartil digunakan untuk menghitung jangkauan antar kuartil (IQR = Q1 - Q3) guna menghitung variabilitas di sekitar median.
- Setiap kuartil memuat 25% dari total data.

Untuk mencari letak kuartil, gunakan persamaan:

$$Q_i = \frac{i(n+1)}{4} \text{ dengan } i = 1, 2, 3$$

```
lagu
```

1. 5.3
2. 3.6
3. 5.5
4. 4.7
5. 6.7
6. 4.3
7. 6.2
8. 4.3
9. 4.9
10. 5.1
11. 5.8
12. 4.4

```
mean(lagu)
```

5.06666666666667

```
median(lagu)
```

5

```
summary(lagu) # digunakan untuk mencari sari data
(termasuk kuartil)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  3.600   4.375   5.000   5.067   5.575   6.700 
```

```
# cara menghitung kuantil:
# sintaks: quantile(data, c(probabilitas1,probabilitas2))
quantile(lagu, c(.25, .75))
```

25%
: 4.375

75%
: 5.575

Varian dan standar deviasi

Digunakan untuk mengukur variabilitas dari suatu data atau akurasi dari parameter - parameter statistik.

- Varian: merupakan ukuran sebaran antar elemen di dalam sampel.
- Standar deviasi: merupakan akar kuadrat dari varian.

$$s^2 = \frac{\sum_{i=1}^{n} (\bar{x} - x_i)^2}{n - 1}$$

```
lagu
```

1. 5.3
2. 3.6
3. 5.5
4. 4.7
5. 6.7
6. 4.3
7. 6.2
8. 4.3
9. 4.9
10. 5.1
11. 5.8
12. 4.4

```
var(lagu) # varian
```

0.787878787878788

```
sd(lagu) # std
```

0.887625364598595

Varian dan standar deviasi pada harga saham

```
library(dplyr)
```

```
Attaching package: ‘dplyr’

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

```
gedata <- read.csv("../data/GEStock.csv")
geprice <- select(gedata,Price)
```

```
ibmdata <- read.csv("../data/IBMStock.csv")
ibmprice <- select(ibmdata, Price)
```

```
var(geprice)
```

	PRICE
Price	575.6425

```
var(ibmprice) # lebih volatil ketimbang general electrics
```

	PRICE
Price	7712.717

```
sd(as.vector(geprice$Price))
```

23.992551305301

```
sd(as.vector(ibmprice$Price))
```

87.822078211186

Korelasi dan kovarian

- Korelasi merupakan metode statistik yang digunakan untuk mengukur derajat relasi antar dua variabel.
- Di dalam dunia finansial, korelasi dapat mengukur pergerakan harga saham.

Korelasi:

$$r = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sqrt{\sum(x - \bar{x})^2}\sqrt{(y - \bar{y})^2}}$$

Hasil korelai selalu berada di antara -1 hingga +1

- Korelasi positif menunjukkan bahwa tren kedua data berada pada arah yang sama.
- Korelasi negatif menunjukkan bahwa tren kedua data berada pada arah yang berbeda.
- Korelasi nol menunjukkan bahwa tidak ada hubungan antar tren kedua data.

```
x <- seq(10,50, by=10)
y <- x
```

```
cor(x,y) # karena data sama maka r = 1
```

1

```
y <- c(50,40,30,20,10)
```

```
cor(x,y) # terbalik r = -1
```

-1

```
x <- c(41,19,23,40,55)
y <- c(94,60,74,71,82)
cor(x,y)
```

0.64810840039477

```
gedates <- select(gedata, Date)
geprice <- select(gedata, Price)
ibmdates <- select(ibmdata, Date)
ibmprice <- select(ibmdata, Price)
```

```
cor(geprice,ibmprice) # secara default menggunakan
korelasi pearson
```

	PRICE
Price	0.1098373

```
cor(geprice,ibmprice, use='complete.obs') # guna menangani
nilai NaN
```

	PRICE
Price	0.1098373

```
cor(geprice, ibmprice, method = 'spearman') # korelasi
spearman
```

	PRICE
Price	0.1665118

```
geprice_vec <- as.vector(geprice$Price)
ibmprice_vec <- as.vector(ibmprice$Price)
```

```
cor.test(geprice_vec,ibmprice_vec, method='pearson')
```

Pearson's product-moment correlation

```
data:  geprice_vec and ibmprice_vec
t = 2.416, df = 478, p-value = 0.01607
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.02053871 0.19739721
sample estimates:
      cor
0.1098373
```

```
cor.test(geprice_vec,ibmprice_vec, method='spearman')
```

Spearman's rank correlation rho

```
data:  geprice_vec and ibmprice_vec
S = 15362788, p-value = 0.0002528
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.1665118
```

```
cor.test(geprice_vec,ibmprice_vec, method='kendall')
```

Kendall's rank correlation tau

```
data:  geprice_vec and ibmprice_vec
z = 3.9796, p-value = 6.902e-05
alternative hypothesis: true tau is not equal to 0
sample estimates:
      tau
0.1215379
```

- Kovarian merupakan metode statistik yang digunakan untuk mengukur hubungan langsung antara kedua variabel.
- Ketika dua harga saham mempunyai perbedaan varian yang besar, maka kovariannya positif, pun begitu sebaliknya (kebalikan dari korelasi).
- Banyak digunakan pada teori potofolio modern di bidang finansial.

```
cov(geprice, ibmprice)
```

	PRICE
Price	231.4354

Contoh kasus perbandingan harga saham

```
cor(geprice,ibmprice)
```

	PRICE
Price	0.1098373

```
cov(geprice,ibmprice)
```

	PRICE
Price	231.4354

```
# mengimpor data saham cocacola
cocadata <- read.csv("../data/CocaColaStock.csv")
cocadates <- select(cocadata, Date)
cocaprice <- select(cocadata, Price)
```

```
cor(geprice, cocaprice)
```

	PRICE
Price	0.1775435

```
cov(geprice, cocaprice)
```

	PRICE
Price	107.2014

Dapat dikatakan harga saham di GE dan CocaCola secara komparatif lebih terhubung daripada GE dengan IBM.

Analisis data bivariat dan multivariat

Data bivariat

Data bivariat mendeskripsikan hubungan antar dua buah variabel. Misalnya:

- Hubungan antara berat dan tinggi badan.
- Hubungan antara risiko penyakit jantung dengan jenis kelamin.
- dll.

Data bivariat, terdiri dari:

- 2 variabel kualitatif
- 1 variabel kualitatif dan 1 variabel kuantitatif.
- 2 variabel kuantitatif.

Data bivariat kualitatif

```
ratings <-
factor(c(2,4,3,3,2,1,1,2,3,4,2,3,3,4,1,3,2,4,3,2,1))
ratings
```

1. 2
2. 4
3. 3
4. 3
5. 2
6. 1
7. 1
8. 2
9. 3
10. 4
11. 2
12. 3
13. 3
14. 4
15. 1
16. 3
17. 2
18. 4
19. 3
20. 2
21. 1

▶ **Levels**:

```
kursus <-
factor(c(1,1,0,0,1,1,0,0,1,0,0,0,1,0,1,0,1,1,1,0,1))
kursus
```

1. 1
2. 1
3. 0
4. 0
5. 1

6. 1
7. 0
8. 0
9. 1
10. 0
11. 0
12. 0
13. 1
14. 0
15. 1
16. 0
17. 1
18. 1
19. 1
20. 0
21. 1

▶ **Levels**:

```
levels(kursus) <- c('R', 'Python')
```

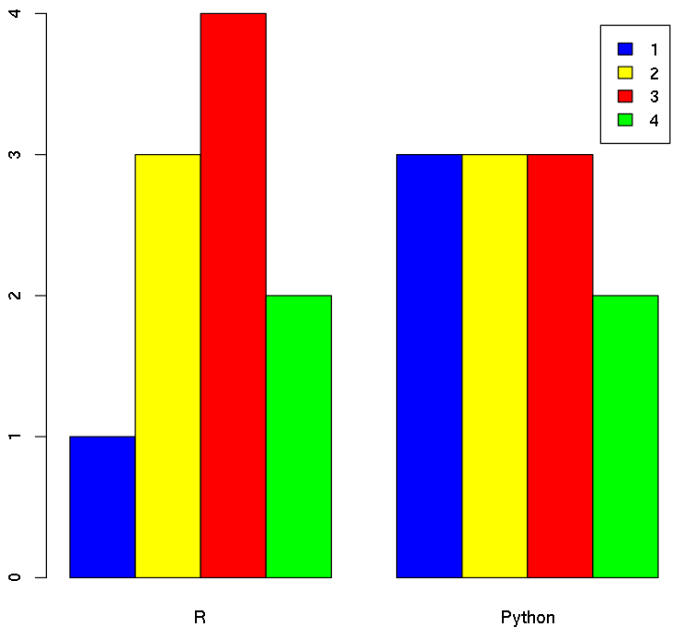
```
table(ratings, kursus)
```

```
       kursus
ratings R Python
      1 1      3
      2 3      3
      3 4      3
      4 2      2
```

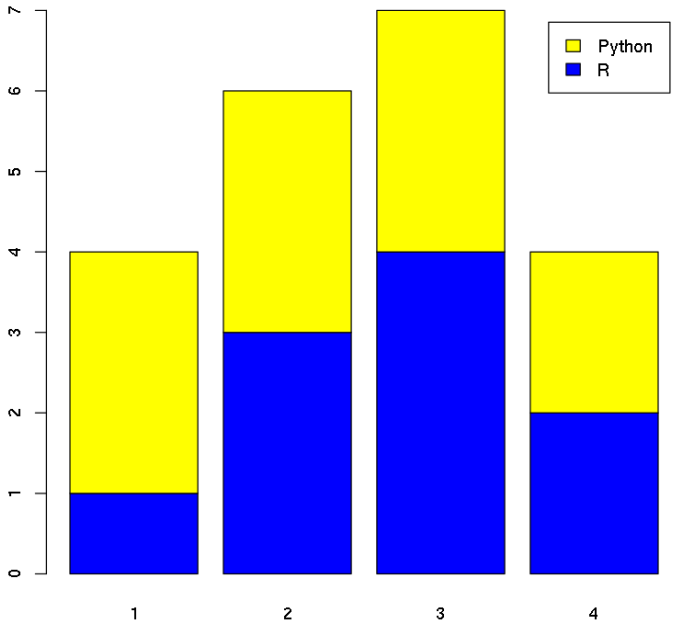
```
table(kursus, ratings)
```

```
        ratings
kursus   1 2 3 4
  R      1 3 4 2
  Python 3 3 3 2
```

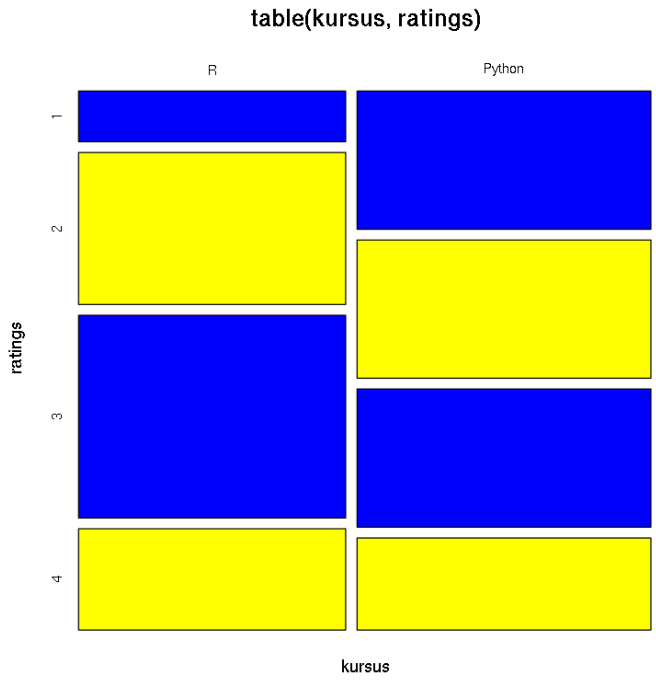
```
barplot(table(ratings, kursus),
       col=c('blue', 'yellow', 'red', 'green'), legend=T,
beside=T)
```




```
barplot(table(kursus, ratings),
        col=c('blue', 'yellow', 'red', 'green'),
        legend.text=T)
```



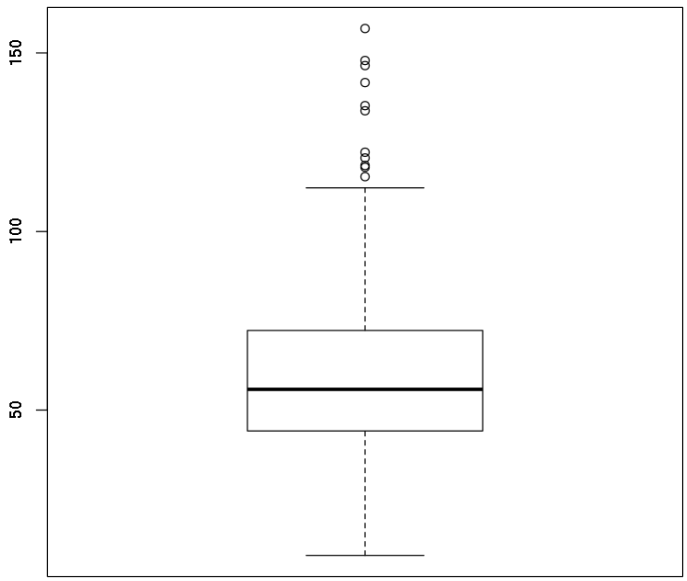

```
mosaicplot(table(kursus,ratings),
           col=c('blue', 'yellow'))
```



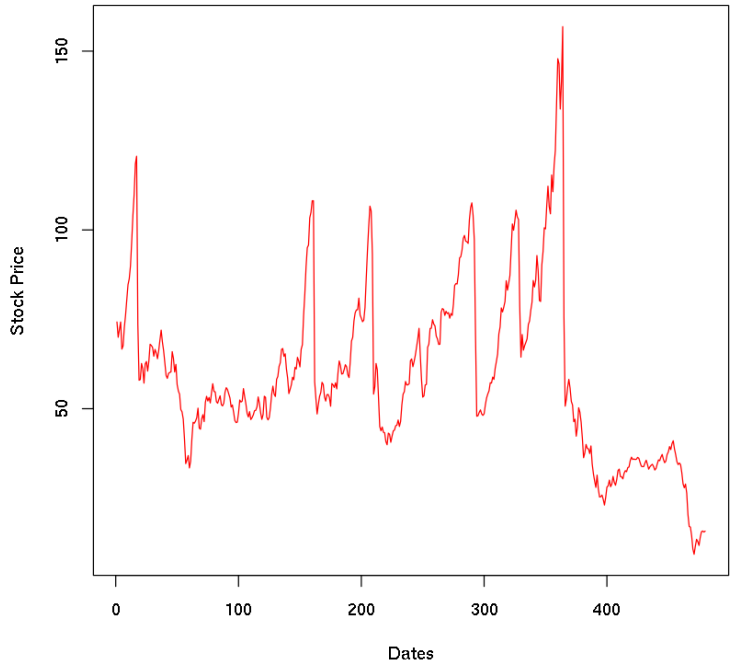
Data bivariat kuantitatif

```
library(dplyr)
df <- read.csv("../data/GEStock.csv")
dates <- select(df, Date)
price <- select(df, Price)
```

```
boxplot(price) # hanya untuk univariat
```



```
plot(df$Price,
    xlab='Dates',
    ylab='Stock Price',
    col='red',
    type='l')
```




```
max(df$Price)
```

156.8436842

```
which(df$Price == max(df$Price)) # indeks maksimum
```

364

```
df[which(df$Price == max(df$Price)),]
```

	DATE	PRICE
364	4/1/00	156.8437

Data multivariat

```
df <- read.csv("../data/murders.csv")
head(df)
```

STATE	ABB	REGION	POPULATION	POPULATIONDENSITY	MURDERS	GUNMURDERS	GUNOW

STATE	ABB	REGION	POPULATION	POPULATIONDENSITY	MURDERS	GUNMURDERS	GUNOW
Alabama	AL	South	4779736	94.65	199	135	0.517
Arizona	AZ	West	6392017	57.05	352	232	0.311
California	CA	West	37253956	244.20	1811	1257	0.213
Colorado	CO	West	5029196	49.33	117	65	0.347
Connecticut	CT	Northeast	3574097	741.40	131	97	0.167
Florida	FL	South	19687653	360.20	987	669	0.245

```
str(df)
```

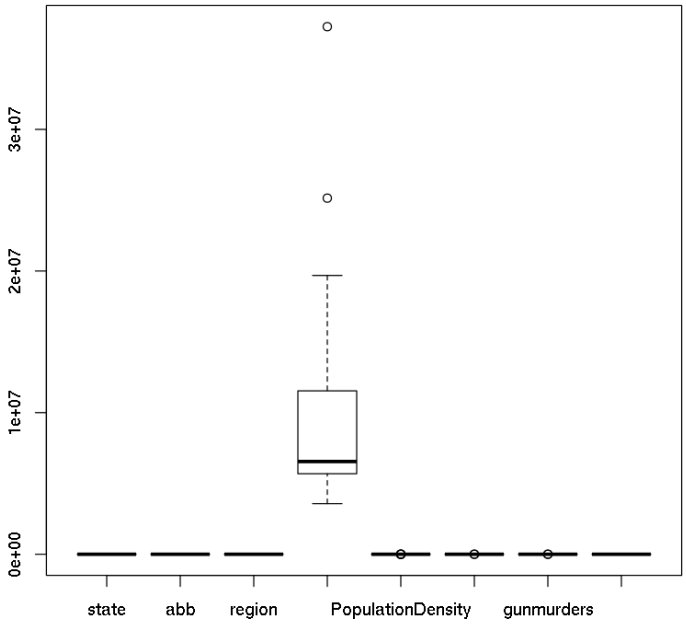
```
'data.frame':   25 obs. of  8 variables:
 $ state            : Factor w/ 25 levels
"Alabama","Arizona",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ abb              : Factor w/ 25 levels
"AL","AZ","CA",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ region           : Factor w/ 4 levels "North
Central",..: 3 4 4 4 2 3 3 1 1 3 ...
 $ population       : int  4779736 6392017 37253956
5029196 3574097 19687653 9920000 12830632 6483802 4339367
...
 $ PopulationDensity: num  94.7 57 244.2 49.3 741.4 ...
 $ murders          : int  199 352 1811 117 131 987 527
453 198 180 ...
 $ gunmurders       : int  135 232 1257 65 97 669 376 364
142 116 ...
 $ gunownership     : num  0.517 0.311 0.213 0.347 0.167
0.245 0.403 0.202 0.391 0.477 ...
```

```
summary(df)
```

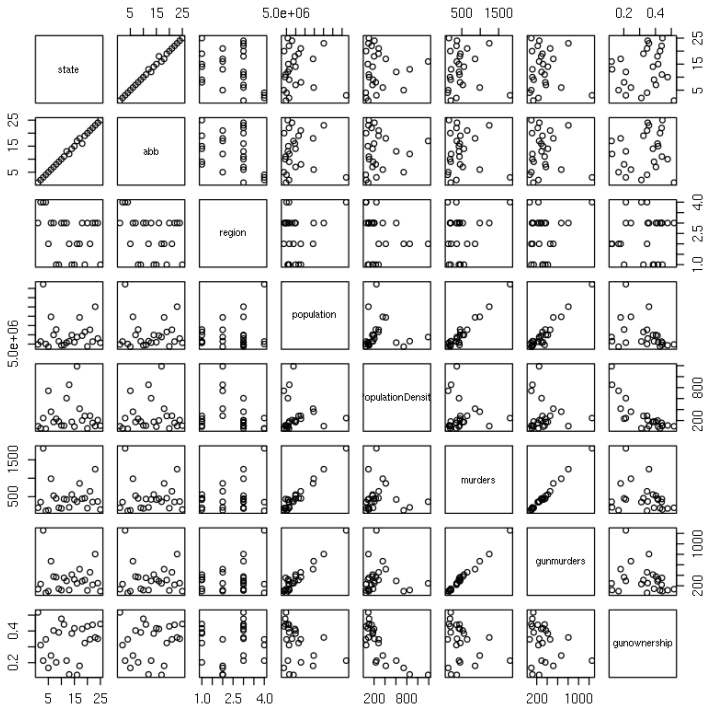
```
         state         abb               region
population
 Alabama    : 1   AL     : 1   North Central: 6   Min.   :
3574097
 Arizona    : 1   AZ     : 1   Northeast    : 5   1st Qu.:
5686986
 California : 1   CA     : 1   South        :11   Median :
6547629
 Colorado   : 1   CO     : 1   West         : 3   Mean
:10155719
 Connecticut: 1   CT     : 1                      3rd
Qu.:11536504
 Florida    : 1   FL     : 1                      Max.
:37253956
 (Other)    :19   (Other):19

 PopulationDensity    murders         gunmurders
gunownership
 Min.   :  49.33   Min.   : 117.0   Min.   :  65.0   Min.
  :0.1230
 1st Qu.: 105.00   1st Qu.: 199.0   1st Qu.: 135.0   1st
Qu.:0.2130
```

```
 Median : 182.50   Median : 419.0   Median : 286.0
Median :0.3510
 Mean   : 282.57   Mean   : 483.4   Mean   : 329.9   Mean
  :0.3305
 3rd Qu.: 285.30   3rd Qu.: 527.0   3rd Qu.: 376.0   3rd
Qu.:0.4170
 Max.   :1189.00   Max.   :1811.0   Max.   :1257.0   Max.
  :0.5170
```

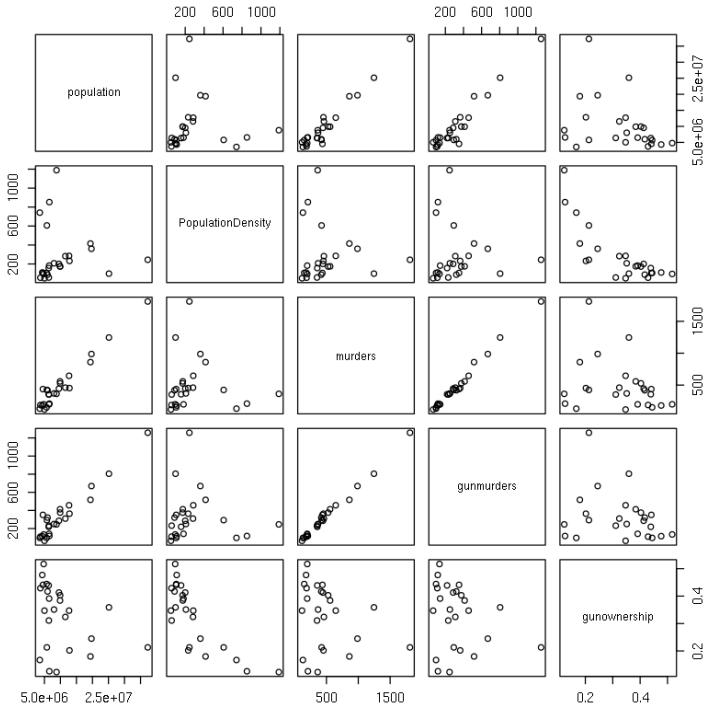
```
boxplot(df)
```




```
plot(df)
```



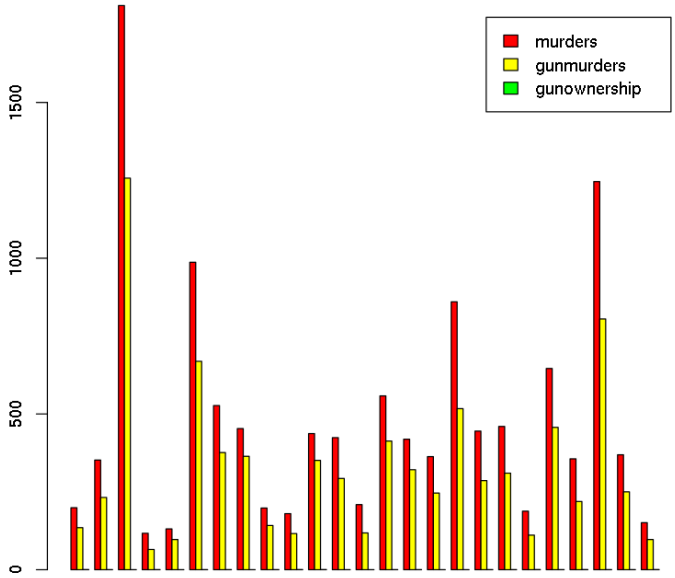

```
pairs(df[,-c(1,2,3)]) # tidak memasukan kolom no 1,2,3
```




```
dfsel <- df[,-c(1,2,3,4,5)] # hanya memasukan data
kuantitatif
```

```
mat <- data.matrix(dfsel) # konversi data terseleksi ke
matriks
mat <- t(mat)
```

```
barplot(mat,
        col=c('red', 'yellow',
              'green'),
        beside=T,
        names.arg=dfsel$state,
        legend.text=T)
```



Distribusi peluang

Pendahuluan

Distribusi peluang merupakan fungsi statistik yang digunakan untuk mendeskripsikan seluruh kemungkinan nilai dari suatu variabel acak.

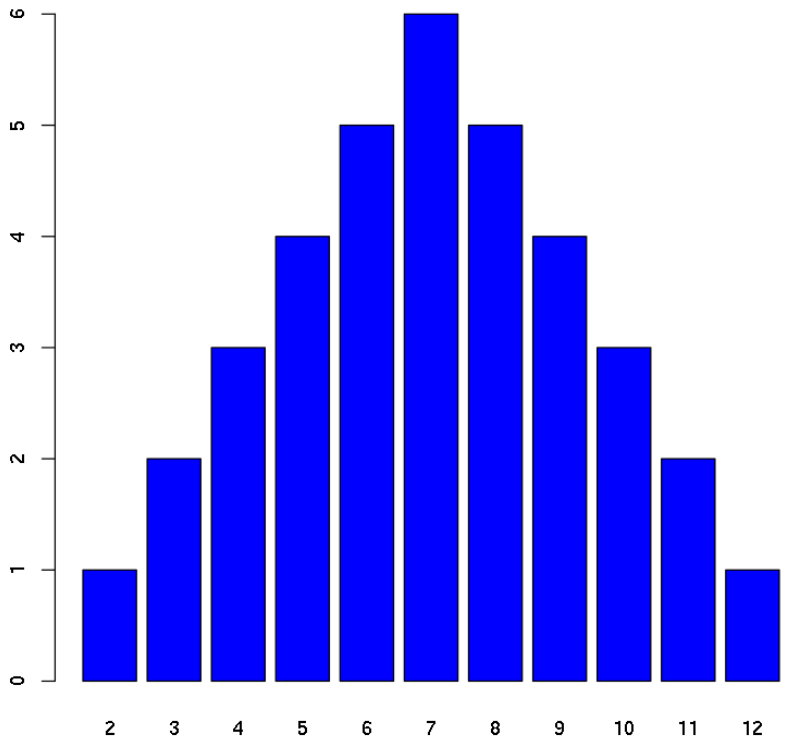
Distribusi peluang bergantung pada:

- Rata - rata
- Standar deviasi
- Kemencengan

Jenis - jenis:

- Distribusi normal
- Distribusi seragam
- Distribusi chi-kuadrat
- Distribusi binomial
- Distribusi Poisson

```
# Contoh pelemparan dua buah dadu

totaldadu <- c(2,3,4,5,6,7,8,9,10,11,12)
posibilitas <- c(1,2,3,4,5,6,5,4,3,2,1)
barplot(posibilitas,
       col='blue',
       names.arg=totaldadu)
```




```
# distribusi acak seragam (0-1)
runif(5)
```

1. 0.123731347266585
2. 0.60765249398537
3. 0.533600943861529
4. 0.681078718043864
5. 0.986109193181619

```
runif(5,1,6) # (1-6)
```

1. 4.62048985203728
2. 3.29562402702868
3. 1.28878939570859
4. 2.99522116803564
5. 1.37944304407574

```
as.integer(runif(5,1,6)) # simulasi lempar dadu 5x
```

1. 1
2. 1
3. 1
4. 1
5. 4

Distribusi seragam

Distribusi seragam adalah distribusi peluang dengan peluang kemunculan nilai yang sama di antara setiap kemungkinannya. Contoh:

- Pelemparan koin.
- Kartu di dalam dek.

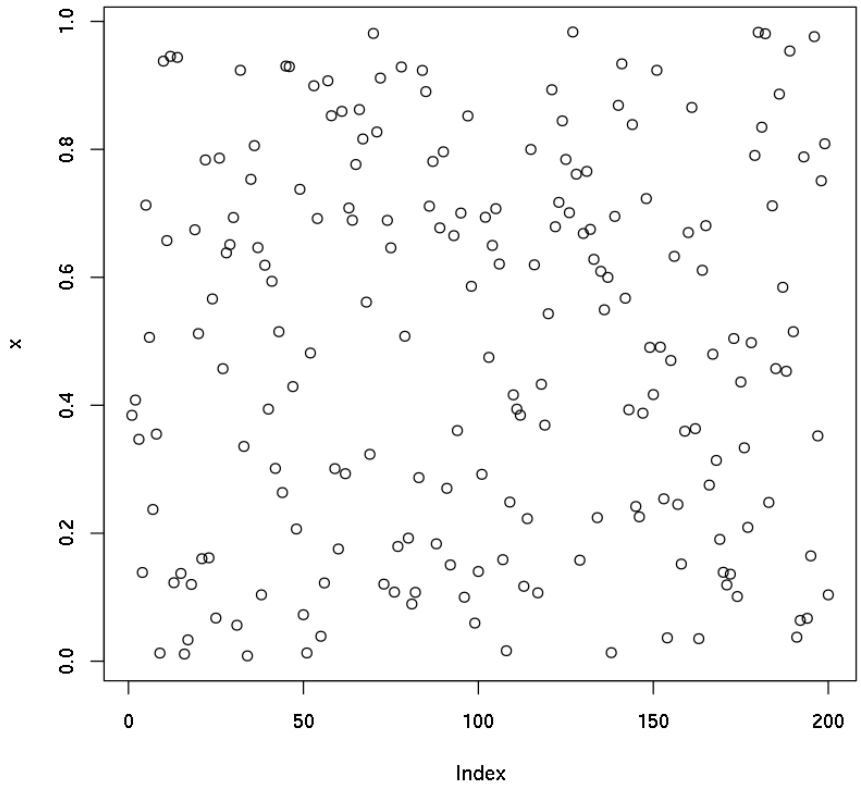
Terdapat dua jenis distribusi seragam:

- Diskrit
- Kontinyu

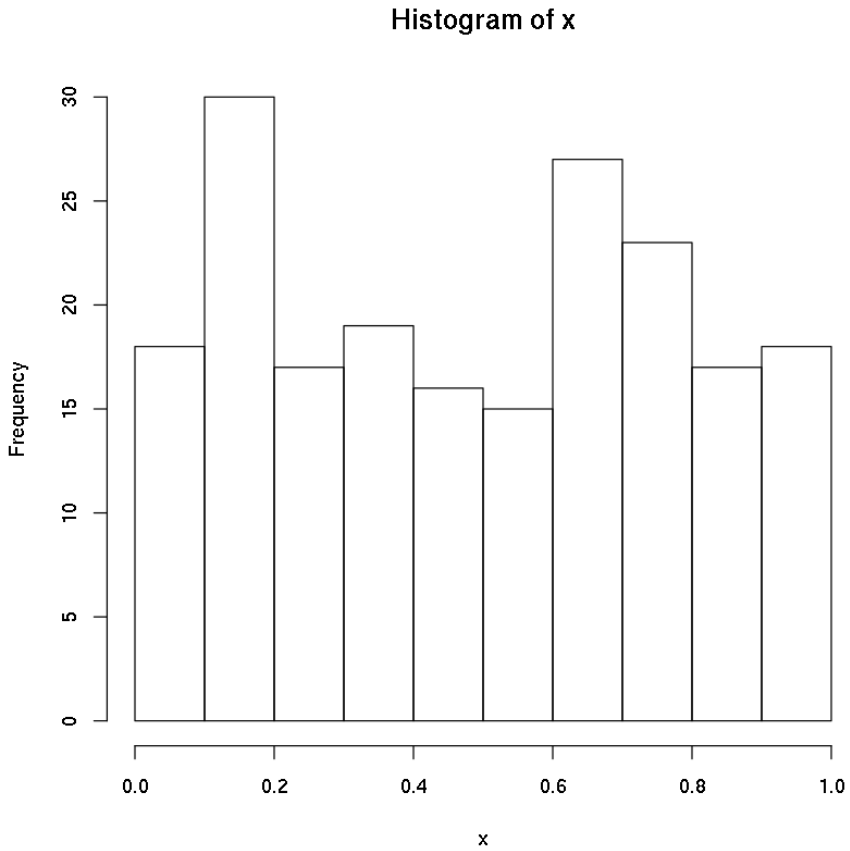
```
runif(10)
```

1. 0.569517947733402
2. 0.27695785346441
3. 0.0101017325650901
4. 0.216280498541892
5. 0.926169243408367
6. 0.289980907225981
7. 0.590355885447934
8. 0.356919593410566
9. 0.0631376963574439
10. 0.385881984839216

```
x <- runif(200)
plot(x)
```


x
Index
0.0
0.2
0.4
0.6
0.8
1.0
0
50
100
150
200

```
hist(x)
```

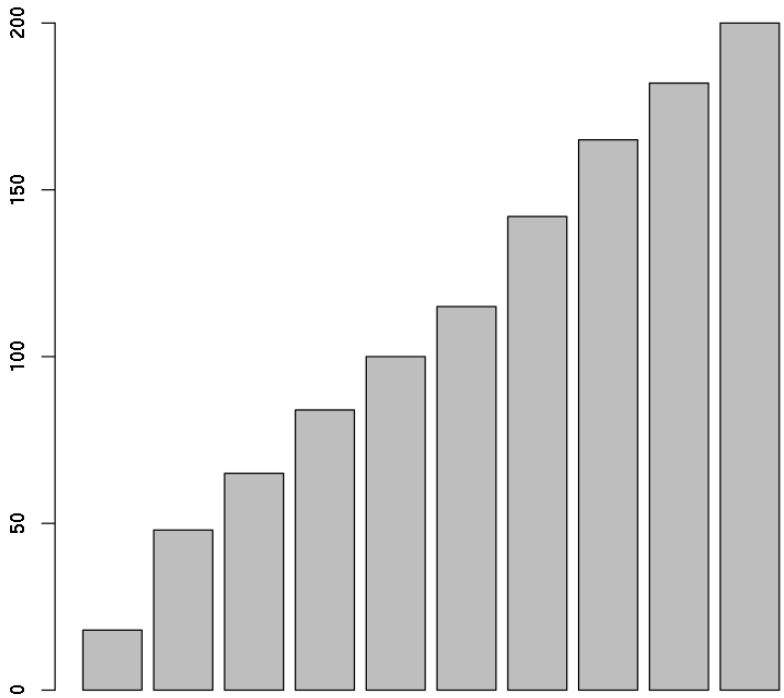

Histogram of x
Frequency
x
0
5
10
15
20
25
30
0.0
0.2
0.4
0.6
0.8
1.0

```
h <- hist(x,plot=F)
h
```

```
$breaks
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

$counts
 [1] 18 30 17 19 16 15 27 23 17 18

$density
 [1] 0.90 1.50 0.85 0.95 0.80 0.75 1.35 1.15 0.85 0.90

$mids
 [1] 0.05 0.15 0.25 0.35 0.45 0.55 0.65 0.75 0.85 0.95

$xname
[1] "x"

$equidist
[1] TRUE

attr(,"class")
[1] "histogram"
```

```
hcum <- cumsum(h$counts)
```

```
barplot(hcum) # distribusi kumulatif
```

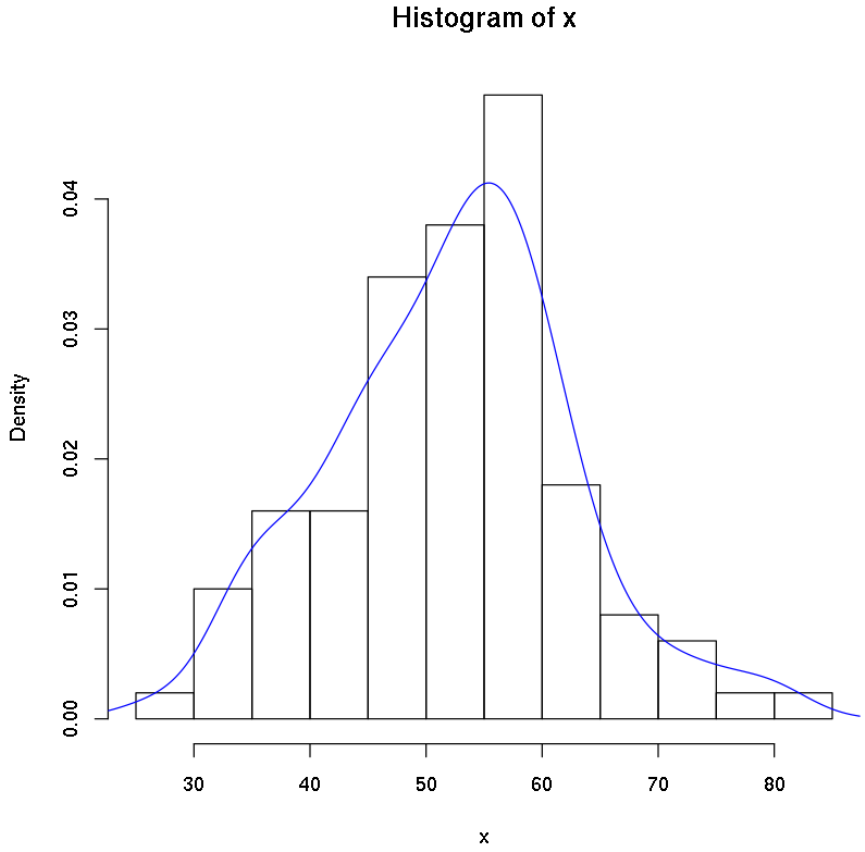


Distribusi normal

Merupakan distribusi peluang yang simetris terhadap rata - ratanya.

```
x <- rnorm(100, 50, 10) # 100 bilangan acak dlm distribusi
normal (rata2 = 50, sd = 10)
```

```
hist(x, probability = T)
lines(density(x), col='blue')
```




```
mean(x)
```

52.2726391744786

```
sd(x)
```

10.3870913897489

Statistika inferensi

- Statistika deskriptif: mengidentifikasi hubungan dari suatu data, menarik kesimpulan dari sampel.
- Statistika inferensi: menarik kesimpulan tentang populasi.

Nilai p

Nilai p merupakan peluang:

- Jika hipotesis awal benar, maka sampel dapat menghasilkan estimasi.
- Menerangkan kemungkinan kita mendapatkan hasil.

Contoh:
Ada warung Zominos Pizza:

- **Komplain** pelanggan bahwa kejunya tidak cukup banyak.
- **Regulasi**: Harus pakai 100 gr keju untuk setiap pizza.

Namun, manajer Zominos tidak dapat mengunjungi seluruh warung di kota tersebut, sehingga diambilah sampel dari beberapa warung.

Pada kasus ini:

- H_0 : (*Trying to provide evidence against*) Menolak klaim pelanggan (Cukup kok kejunya).
- H_1 : Apa yang hendak kita buktikan, yakni tidak cukup keju untuk setiap Pizza.

Significance level (α) = 0,05

Jika nilai p dibawah α, maka kita dapat menolak hipotesis awal.

```
library(dplyr)
df <- read.csv("../data/ZominosCheese.csv")
head(df)
```

```
Attaching package: ‘dplyr’

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

GRAMSCHEESE
85
100
85
98
94
97

```
# peraturannya harus pakai keju 100 gr (mu)
t.test(df,mu=100)
```

One Sample t-test

```
data:  df
t = -0.96396, df = 29, p-value = 0.343
alternative hypothesis: true mean is not equal to 100
95 percent confidence interval:
  95.94179 101.45821
sample estimates:
mean of x
     98.7
```

Nilai p -nya 0,0343, nilai rata2nya = 98,7. Nilai $p > 0,05$, maka kita menolak hipotesis awal.

Derajat kebebasan

Derajat kebebasan merujuk pada ukuran maksimum dari nilai - nilai logikal yang bersifat independen.

$$df = n - 1$$

```
data <- read.csv("../data/ZominosCheese.csv")
head(data)
t.test(data, mu=100)
```

GRAMSCHEESE
85
100
85
98
94
97

```
    One Sample t-test

data:  data
t = -0.96396, df = 29, p-value = 0.343
alternative hypothesis: true mean is not equal to 100
95 percent confidence interval:
  95.94179 101.45821
sample estimates:
mean of x
     98.7
```

$df = 29$

```
df = length(data$GramsCheese) - 1
df
```

29

Selang dan level kepercayaan

- Selang kepercayaan mengukur derajat ketidakpastian atau kepastian.
- Level kepercayaan merupakan persentase dari peluang atau kepastian.

```
t.test(data,mu=100)
```

One Sample t-test

```
data:  data
t = -0.96396, df = 29, p-value = 0.343
alternative hypothesis: true mean is not equal to 100
95 percent confidence interval:
  95.94179 101.45821
sample estimates:
mean of x
     98.7
```

Selang kepercayaan: 95,94179 hingga 101,45821

Pengujian hipotesis

Pengujian hipotesis merupakan prosedur di dalam statistika inferensi

- H_0 (hipotesis awal): Dua populasi tidak berbeda jika merujuk pada sifat tertentu.
- H_1 (hipotesis alternatif): Efek tertentu terjadi pada dua buah populasi.

Jika nilai$-p$ dibawah *significance level*, maka kita dapat menolak hipotesis awal.

Untuk melakukan pengujian hipotesis, kita menggunakan uji$-t$

Uji$-t$ adalah salah satu teknik statistika inferensi yang digunakan untuk menentukan perbedaan signifikan rata - rata antar dua kelompok.

Terdapat 3 jenis uji$-t$:

- Uji sampel tunggal.
- Uji dua sampel.
- Uji sampel berpasangan.

Studi kasus

Zominos mempunyai penawaran beli 1 gratis 1 pada hari tertentu untuk meningkatkan penjualan.

- Sampel 1: Penjualan pada hari - hari promosi.
- Sampel 2: Penjualan pada hari - hari biasa.

Berikut adalah hipotesis nya:

- H_0 : Tidak ada perbedaan antara penjualan di hari - hari promosi dan hari - hari biasa (Penjualan promo = Penjualan biasa).
- H_1 : Terdapat perbedaan (Penjualan promo $\neq$ Penjualan biasa).

(Disebut sebagai *two-tail test*).

- H_0 : Penjualan promo - Penjualan biasa = 0
- H_1 : Penjualan promo - Penjualan biasa $\neq 0$

Jika kita tidak tertarik pada penjualan yang berkurang pada hari - hari promosi, maka uji hipotesis dapat diubah formulasinya:

- H_0 : Penjualan promo - Penjualan biasa ≤ 0.
- H_1 : Penjualan promo - Penjualan biasa > 0.

```
df <- read.csv("../data/ZominosSales.csv")
head(df)
```

OFFERDAYS	NONOFFERDAYS
248.3	215.1
335.2	300.0
338.0	320.6
285.3	276.6
322.2	282.9
283.6	288.1

```
t.test(df$OfferDays, df$NonOfferDays) # Two-tail test
```

Welch Two Sample t-test

```
data:  df$OfferDays and df$NonOfferDays
t = 2.6105, df = 17.229, p-value = 0.01814
alternative hypothesis: true difference in means is not
equal to 0
95 percent confidence interval:
  7.997416 75.042584
sample estimates:
mean of x mean of y
   307.35    265.83
```

Nilai$-p < 0,05$, maka kita menolak H_0 (Penjualan promo - Penjualan biasa = 0).

```
# One-tail test
t.test(df$OfferDays, mu = 265.83) # mu = NonOfferDays
```

One Sample t-test

```
data:  df$OfferDays
t = 4.1576, df = 9, p-value = 0.002456
alternative hypothesis: true mean is not equal to 265.83
95 percent confidence interval:
 284.7592 329.9408
sample estimates:
mean of x
   307.35
```

Nilai$-p < 0,05$, maka kita menolak H_0 (Penjualan promo - Penjualan biasa $\leq$ 0).

Uji chi-kuadrat

Uji chi-kuadrat menilai apakah baris dan kolom pada tabel kontingensi berhubungan secara siginifikan secara statistik.

Terdapat dua jenis uji chi-kuadrat:

- *Test of independence*
- *Goodness-of-Fit Test*
- H_0 : Variabel baris dan kolom dari tabel kontingensi bersifat independen.
- H_1 : Variabel baris dan kolom saling bergantung.

Jika nilai uji chi-kuadrat lebih besar dari *significance level*, maka mengindikasikan bahwa kolom dan baris saling berhubungan satu dengan yang lain.

```
ratings <-
 factor(c(2,4,3,3,2,1,1,2,3,4,2,3,3,4,1,3,2,1,4,3,2,4))
ratings
```

1. 2
2. 4
3. 3
4. 3
5. 2

6. 1
7. 1
8. 2
9. 3
10. 4
11. 2
12. 3
13. 3
14. 4
15. 1
16. 3
17. 2
18. 1
19. 4
20. 3
21. 2
22. 4

▶ **Levels**:

```
kursus <-
factor(c(1,1,1,0,0,0,0,0,1,1,1,0,0,0,0,0,1,1,1,1,1,0))
kursus
```

1. 1
2. 1
3. 1
4. 0
5. 0
6. 0
7. 0
8. 0
9. 1
10. 1
11. 1
12. 0
13. 0
14. 0
15. 0
16. 0
17. 1
18. 1
19. 1
20. 1
21. 1
22. 0

▶ **Levels**:

```
levels(kursus) <- c('R', 'Python')
```

```
kursus
```

1. Python
2. Python
3. Python
4. R
5. R
6. R
7. R
8. R
9. Python
10. Python
11. Python
12. R
13. R
14. R
15. R
16. R
17. Python
18. Python
19. Python
20. Python
21. Python
22. R

- **Levels**:

```
data <- table(ratings, kursus)
data
```

```
       kursus
ratings R Python
      1 3      1
      2 2      4
      3 4      3
      4 2      3
```

```
chisq.test(data)
```

```
Warning message in chisq.test(data):
“Chi-squared approximation may be incorrect”
```

Pearson's Chi-squared test

```
data:  data
X-squared = 2.0095, df = 3, p-value = 0.5704
```

Nilai−p: 0,5704. Maka dapat disimpulkan jika variabel baris dan kolom tidak independen. Secara implisit menyatakan ada hubungan antara kursus dengan ratings.

```
chisq.test(data,simulate.p.value=T)
# Karena datanya kurang, kita simulasikan untuk jumlah
data replikasi yang lebih besar
```

Pearson's Chi-squared test with simulated p-value (based on 2000 replicates)

```
data:  data
X-squared = 2.0095, df = NA, p-value = 0.7076
```

Visualisasi data menggunakan `ggplot2`

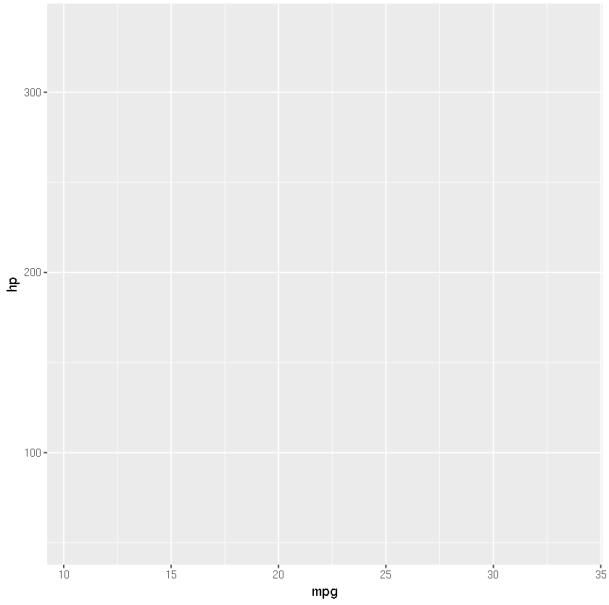
Pendahuluan

- `ggplot2` merupakan pustaka visualisasi pada bahasa pemrograman R.
- Dibangun berdasarkan konsep penambahan lapisan (*layer*) dalam visualisasi.
- Terdapat 7 lapisan: Data, Aesthetics, Geometries, Facets, Statistics, Coordinates, Themes.
- 4 lapisan terakhir tidak wajib, namun dpt digunakan untuk kostumisasi.

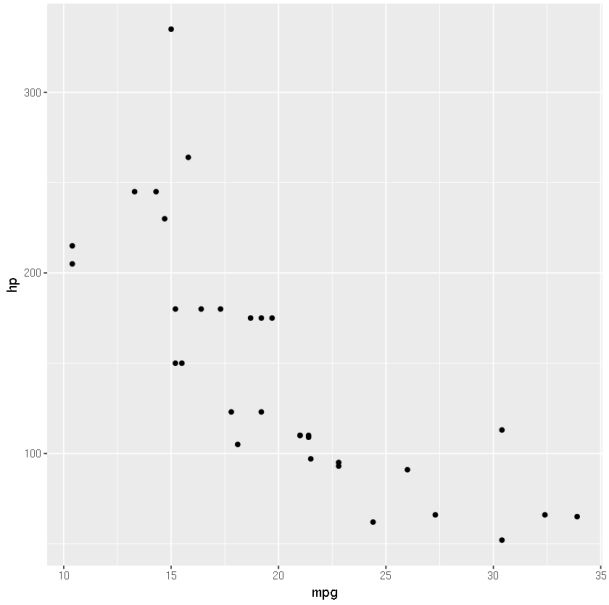
```
library(ggplot2)
```

```
Registered S3 methods overwritten by 'ggplot2':
  method         from
  [.quosures     rlang
  c.quosures     rlang
  print.quosures rlang
```

```
ggplot(data = mtcars, # 1) Lapisan 1: Data
    aes(x = mpg, y = hp)) # 2) Lapisan 2: Aesthetics
```




```
# Lapisan 3): Geometrics
pl <- ggplot(data = mtcars,
             aes(x = mpg, y = hp))
pl + geom_point()
```




```
# 4) Layer 4: Facets
## Membuat kita dapat memplot banyak grafik di dalam satu
kanvas

# 5) Layer 5: Statistics

# 6) Layer 6: Coordinates
## Membatasi limit sumbu-x dan y

# 7) Lapisan 7: Theme
## Menambahkan tema ke dalam suatu plot
```

Histogram

```
library(ggplot2movies)
```

```
head(movies)
```

TITLE	YEAR	LENGTH	BUDGET	RATING	VOTES	R1	R2	R3	R4	...	R9	R10	MPAA	ACTION	ANIMATION
$	1971	121	NA	6.4	348	4.5	4.5	4.5	4.5	...	4.5	4.5		0	0
$1000 a Touchdown	1939	71	NA	6.0	20	0.0	14.5	4.5	24.5	...	4.5	14.5		0	0
$21 a Day Once a Month	1941	7	NA	8.2	5	0.0	0.0	0.0	0.0	...	24.5	24.5		0	1
$40,000	1996	70	NA	8.2	6	14.5	0.0	0.0	0.0	...	34.5	45.5		0	0
$50,000 Climax Show, The	1975	71	NA	3.4	17	24.5	4.5	0.0	14.5	...	0.0	24.5		0	0
$pent	2000	91	NA	4.3	45	4.5	4.5	4.5	14.5	...	14.5	14.5		0	0

```
colnames(movies)
```

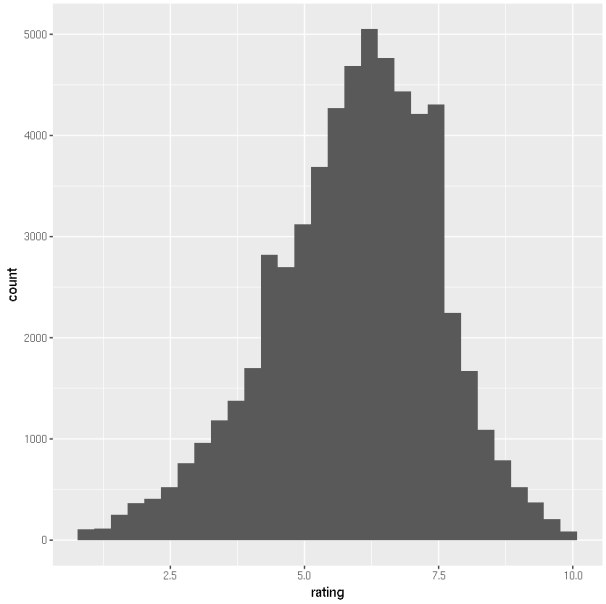
1. 'title'
2. 'year'
3. 'length'
4. 'budget'
5. 'rating'
6. 'votes'
7. 'r1'
8. 'r2'
9. 'r3'
10. 'r4'
11. 'r5'
12. 'r6'

13. 'r7'
14. 'r8'
15. 'r9'
16. 'r10'
17. 'mpaa'
18. 'Action'
19. 'Animation'
20. 'Comedy'
21. 'Drama'
22. 'Documentary'
23. 'Romance'
24. 'Short'

Cheatsheet : https://rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf

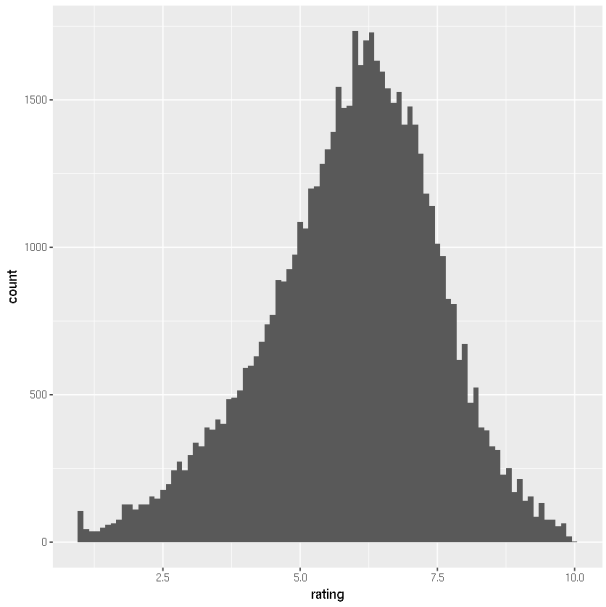
```
pl <- ggplot(data = movies,
             aes(x = rating))
pl + geom_histogram()
```

```
`stat_bin()` using `bins = 30`. Pick better value with
`binwidth`.
```

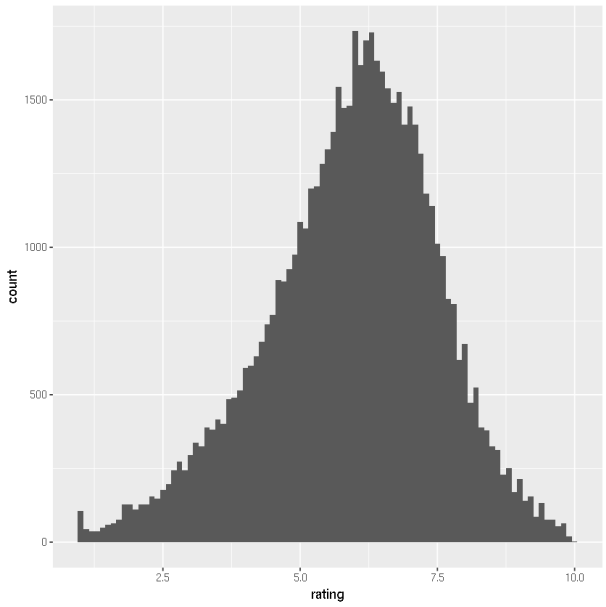



```
# Untuk mengetahui lebih lanjut, perintahkan:
# help("geom_histogram")
```

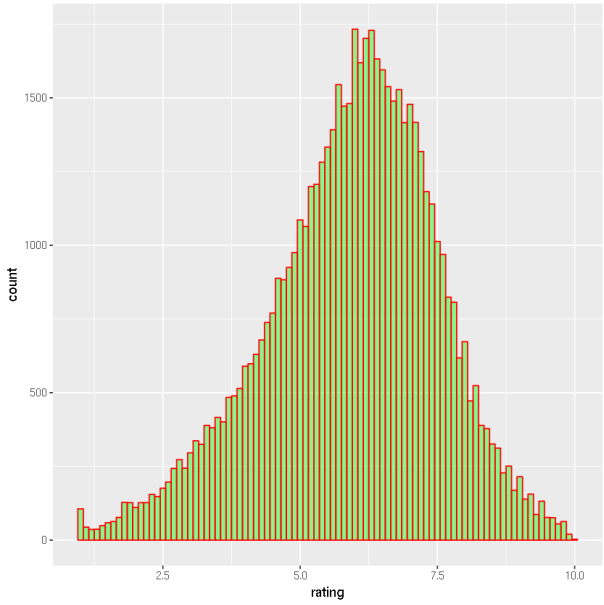
```
pl + geom_histogram(binwidth=0.1) # binwidth = 0.1
```




```
pl + geom_histogram(binwidth=0.1, bins=100) # defaultnya
bins = 30
```

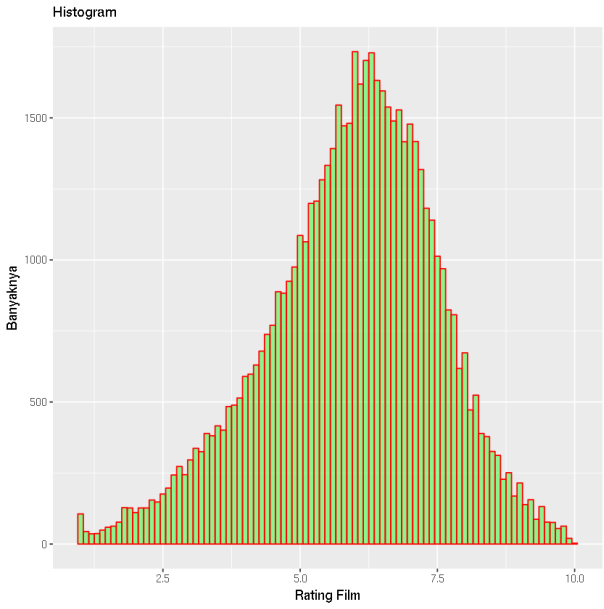



```
pl + geom_histogram(binwidth=0.1, bins=100,
                    color = 'red', fill='green',
                   alpha = 0.4)
```



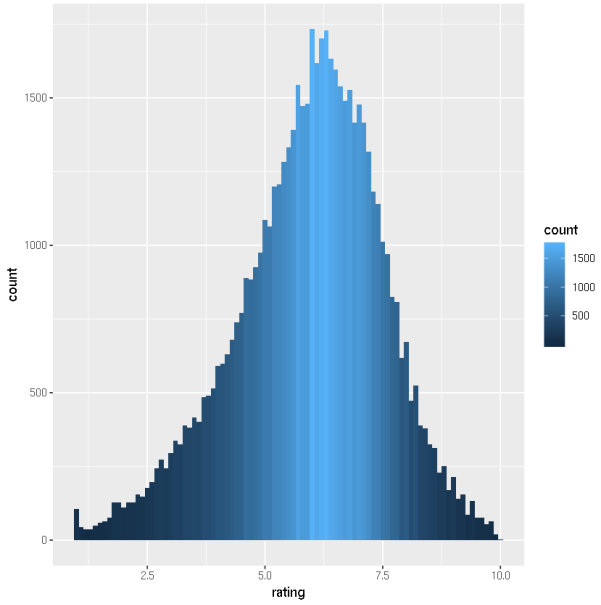
Menambahkan label

```
pl2 <- pl + geom_histogram(binwidth=0.1, bins=100,
                   color = 'red', fill='green',
                   alpha = 0.4)
pl2 + xlab('Rating Film') + ylab('Banyaknya') +
ggtitle("Histogram")
```



Teknik aesthetics lanjutan

```
pl + geom_histogram(binwidth=0.1, aes(fill= ..count..))
```

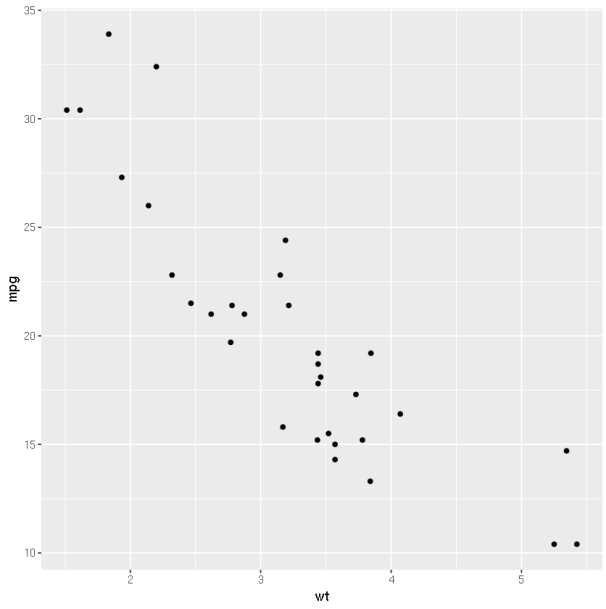


Scatterplot

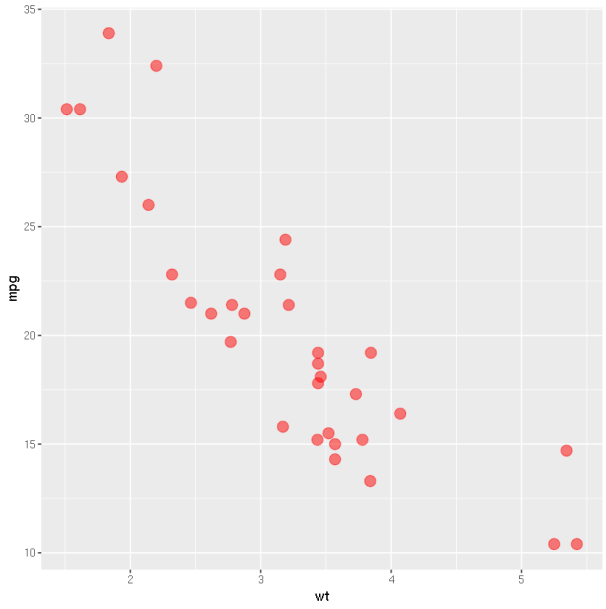
```
df <- mtcars
head(df)
```

	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

```
pl <- ggplot(data = df, aes(x=wt, y=mpg))
pl + geom_point()
```



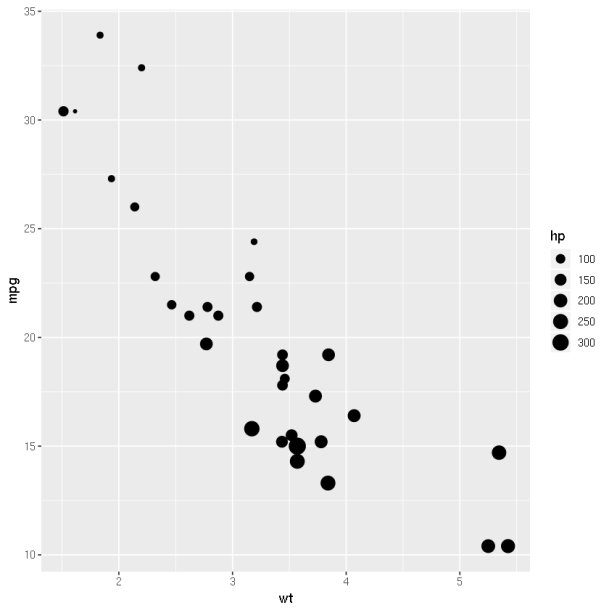

```
pl + geom_point(color = 'red', size=4, alpha = 0.5)
```



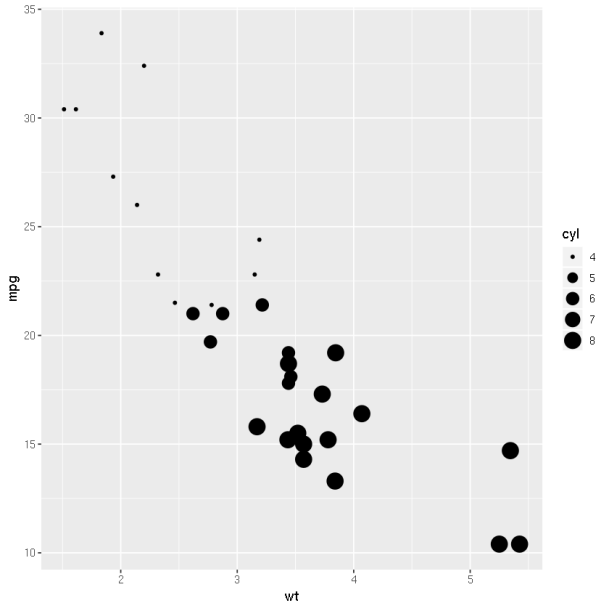

```
# Untuk mengetahui secara lebih lanjut, perintahkan:
# help("geom_point")
```

Menambahkan pemetaan aesthetics

```
pl + geom_point(aes(size = hp)) # ukuran titik berdasarkan
besaran hp
```




```
pl + geom_point(aes(size = cyl))  # ukuran titik
berdasarkan besaran cyl
```

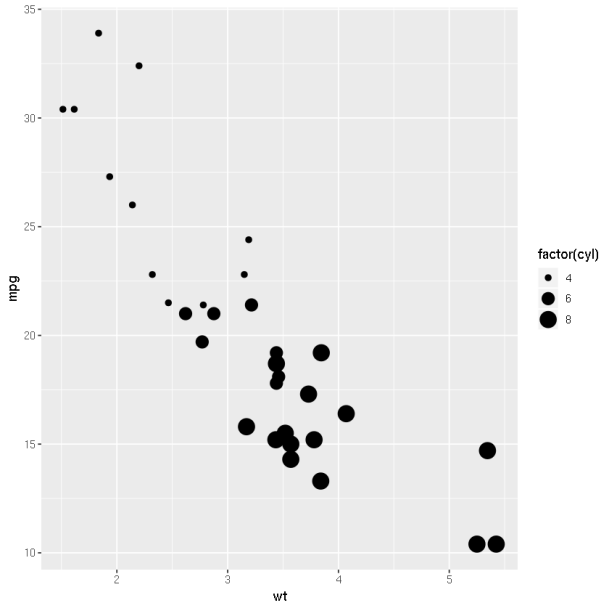



```
df$cyl # bersifat kategorikal, maka kita harus menggunakan fungsi factor()
```

1. 6
2. 6
3. 4
4. 6
5. 8
6. 6
7. 8
8. 4
9. 4
10. 6
11. 6
12. 8
13. 8
14. 8
15. 8
16. 8
17. 8
18. 4
19. 4
20. 4
21. 4
22. 8
23. 8
24. 8
25. 8
26. 4
27. 4
28. 4
29. 8
30. 6
31. 8
32. 4

```
pl + geom_point(aes( size = factor(cyl)))
```

```
Warning message:
“Using size for a discrete variable is not advised.”
```

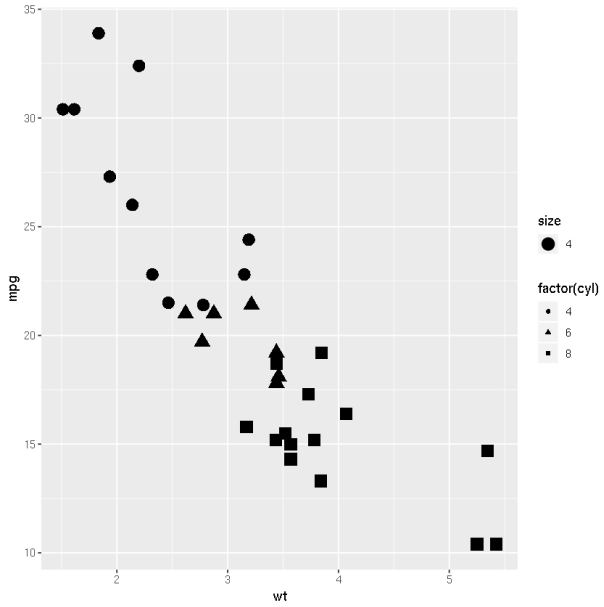


Terdapat pesan:

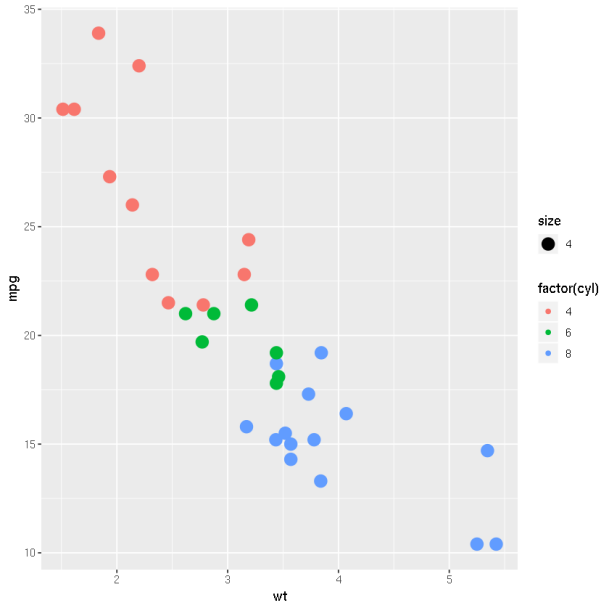
Warning message:
“Using size for a discrete variable is not advised.”

Maka, lebih baik tidak usah digunakan

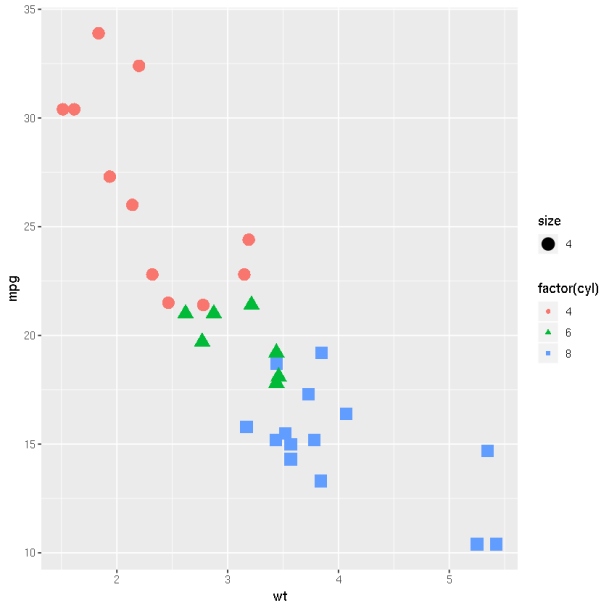
```
pl + geom_point(aes(shape=factor(cyl), size=4)) # pakai
ini lebih baik
```



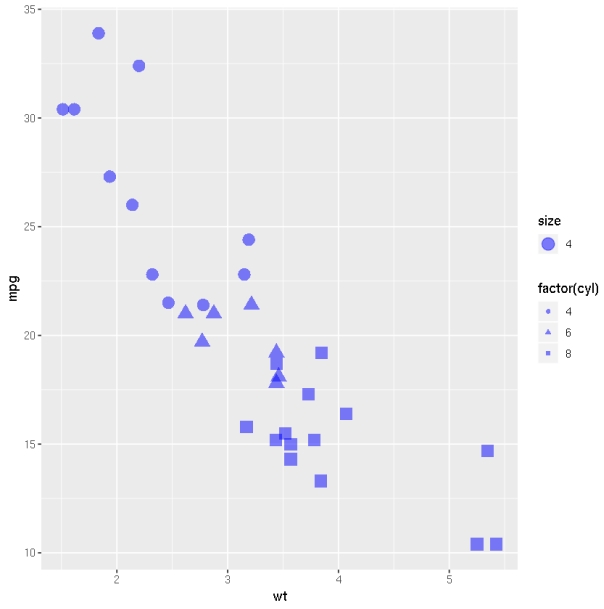

```
# Kita juga dapat membedakan dengan warna
pl + geom_point(aes(color=factor(cyl), size=4))
```




```
# Sintaks lengkap
pl + geom_point(aes(color=factor(cyl), shape =
factor(cyl), size=4))
```

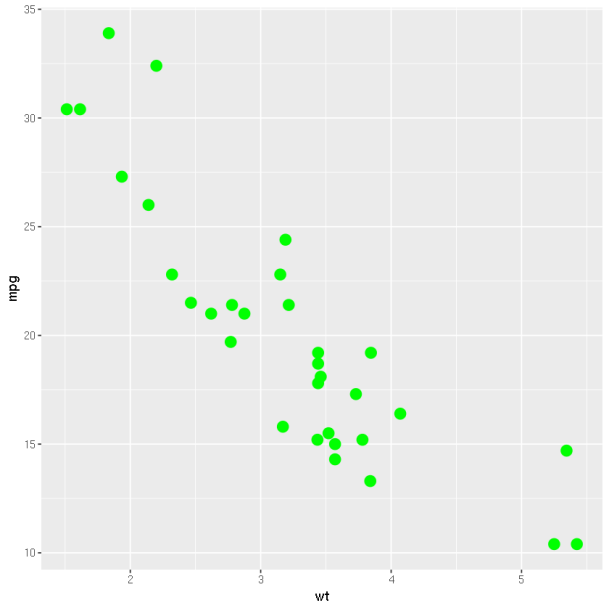



```
pl + geom_point(aes(shape = factor(cyl), size=4),
color='blue', alpha=0.5)
# menambahkan warna di luar aes
```



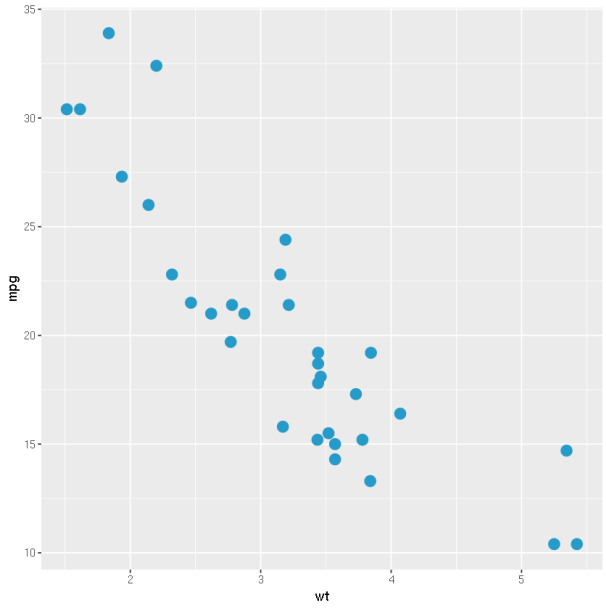
Hex color coding

```
pl + geom_point(size=4, color='green')
```

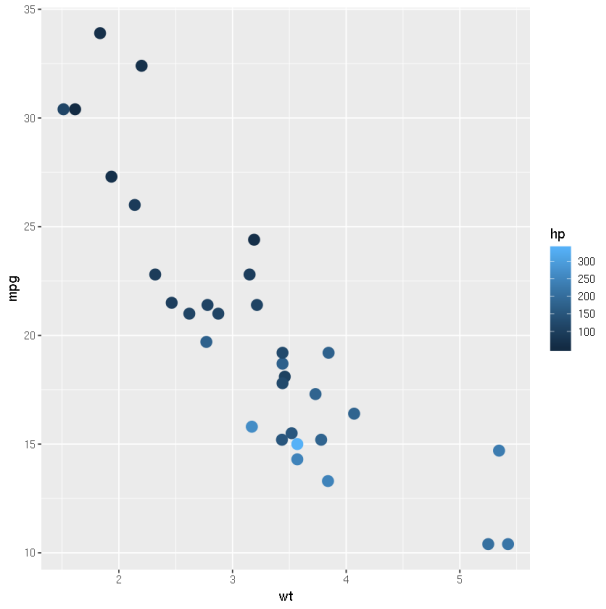


Cari di mesin pencari: hex color code

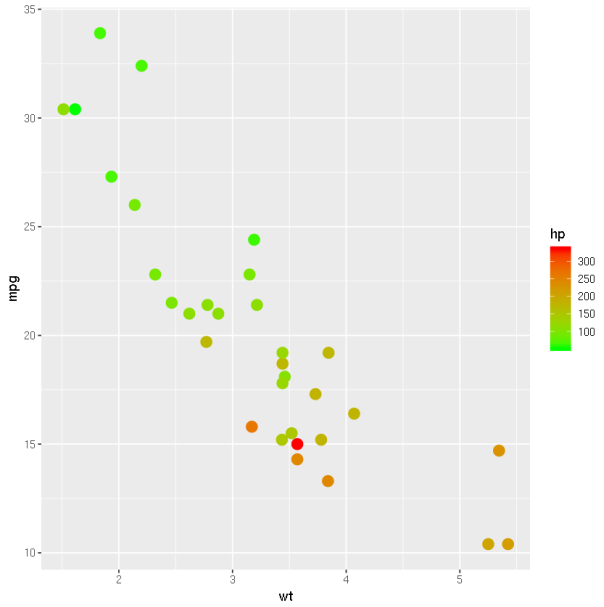
```
pl + geom_point(size=4, color='#269BC9')
```




```
p <- ggplot(df, aes(x=wt, y=mpg))
pl2 <- p + geom_point(aes(color=hp), size=4)
pl2
```

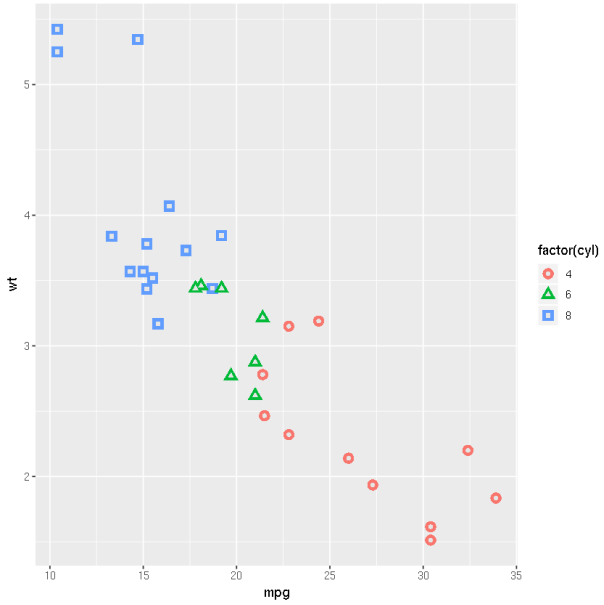



```
pl2 + scale_color_gradient(low='green', high='red')
```




```
# help("geom_point")
```

```
b <- ggplot(mtcars, aes(mpg, wt, shape = factor(cyl)))
b + geom_point(aes(colour = factor(cyl)), size = 4) +
  geom_point(colour = "grey90", size = 1.5)
b + geom_point(colour = "black", size = 4.5) +
  geom_point(colour = "pink", size = 4) +
  geom_point(aes(shape = factor(cyl)))
```



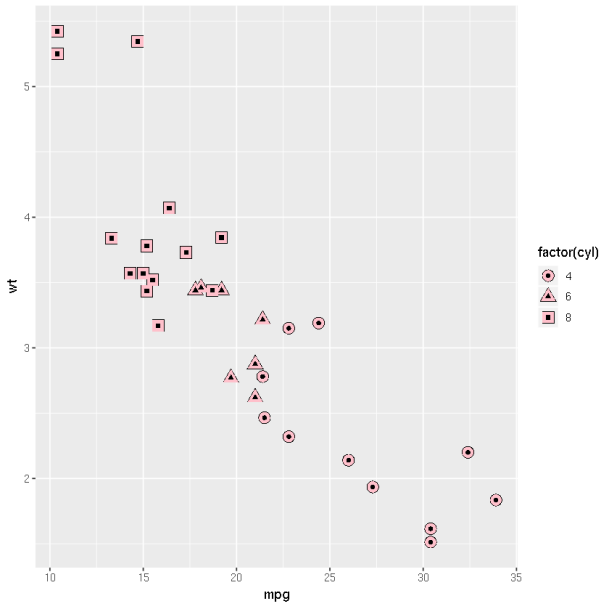


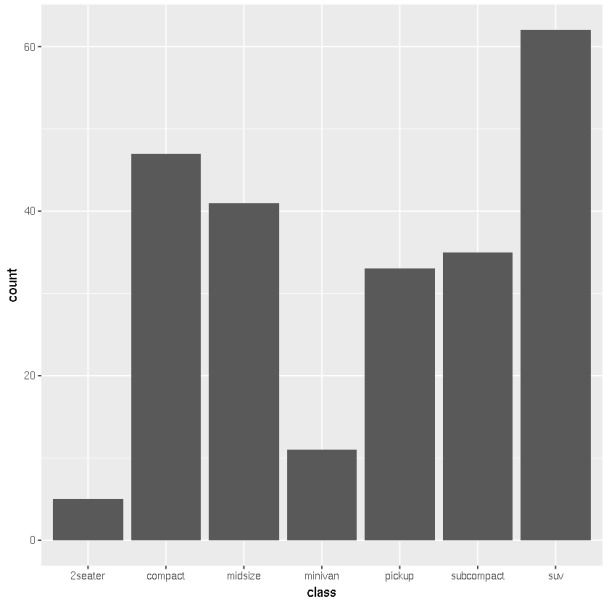
Diagram batang

Umum digunakan untuk menangani data kategorikal

```
df <- mpg
head(mpg)
```

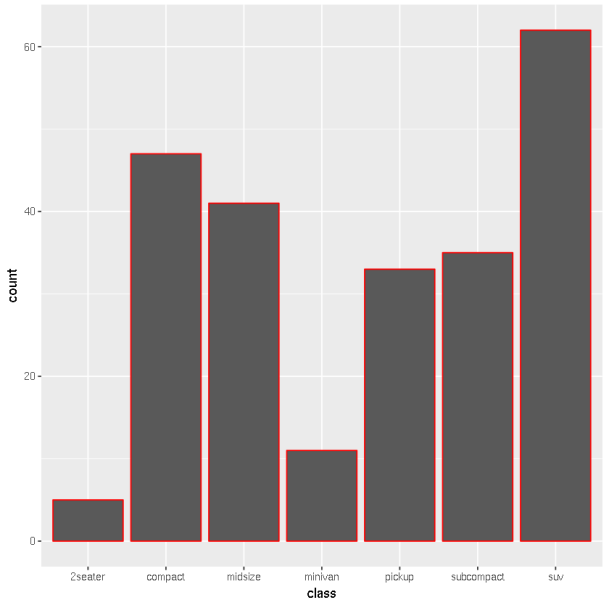
MANUFACTURER	MODEL	DISPL	YEAR	CYL	TRANS	DRV	CTY	HWY	FL	CLASS
audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact

```
pl <- ggplot(df, aes(x=class)) # class : data kategorikal
pl + geom_bar()
```

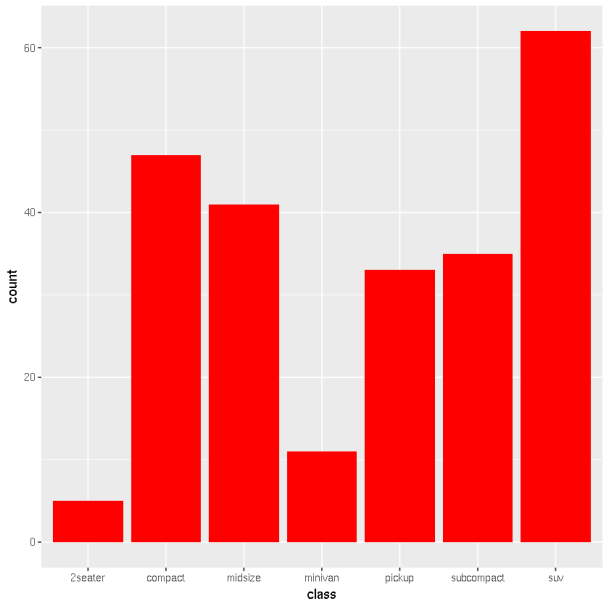



```
# Untuk mengetahui secara lebih lanjut, perintahkan:
# help("geom_bar")
```

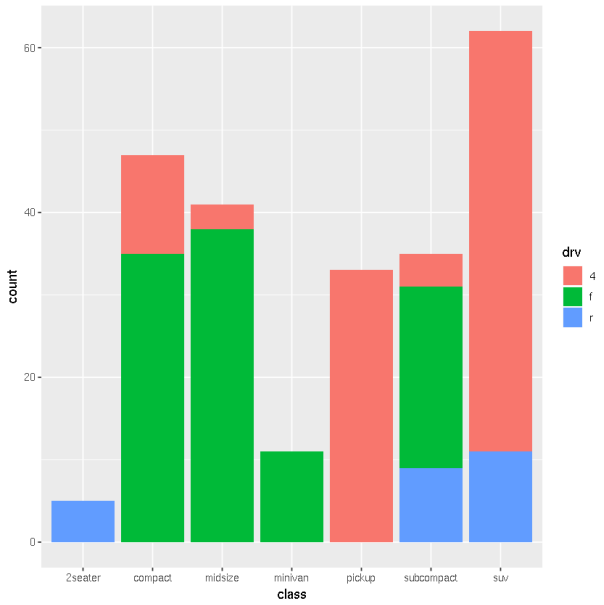
```
pl + geom_bar(color='red')
```



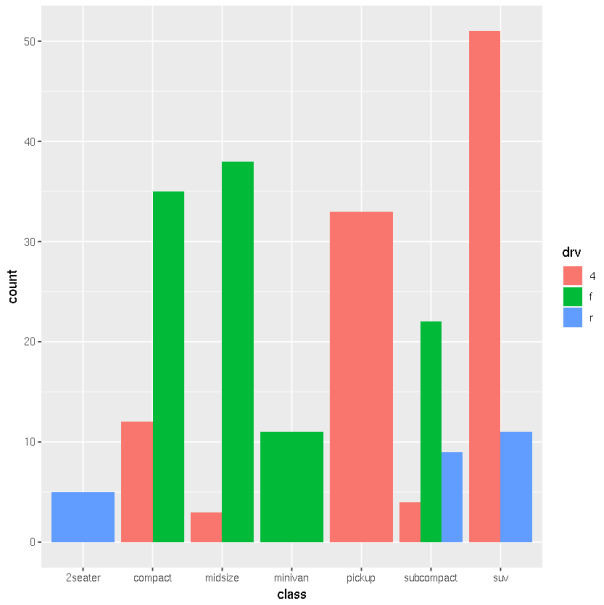

```
pl + geom_bar(fill='red')
```



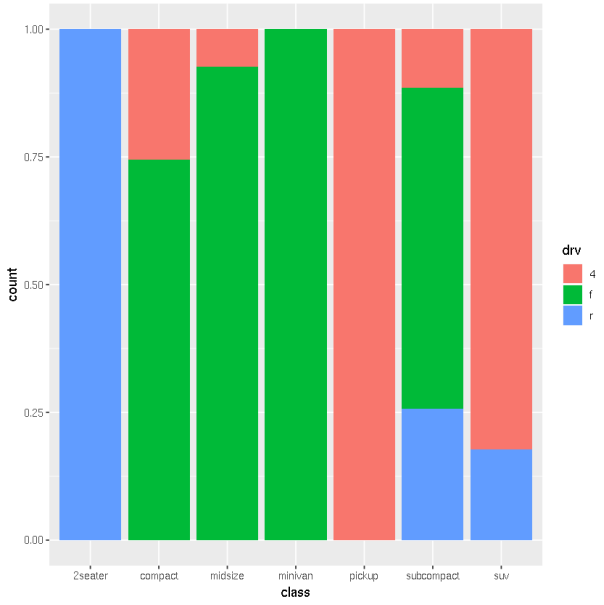

```
pl + geom_bar(aes(fill=drv)) # fill di dasarkan pada
jumlah drv
```




```
pl + geom_bar(aes(fill=drv), position='dodge') #
dipisahkan
```

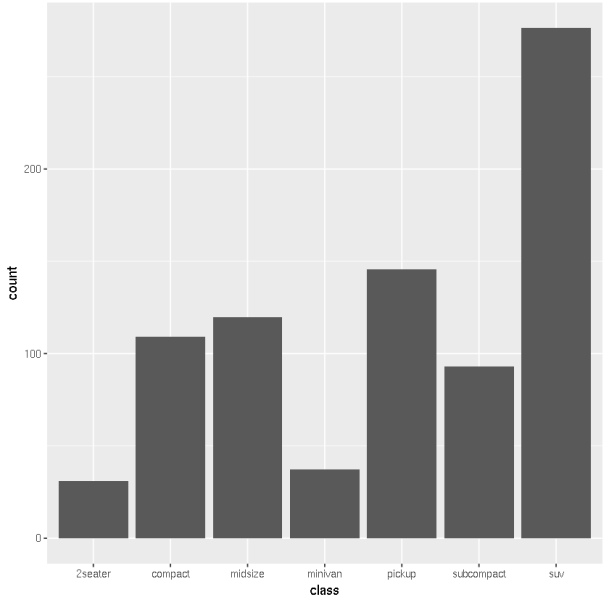



```
pl + geom_bar(aes(fill=drv), position='fill') # dihitung
berdasarkan persentase
```



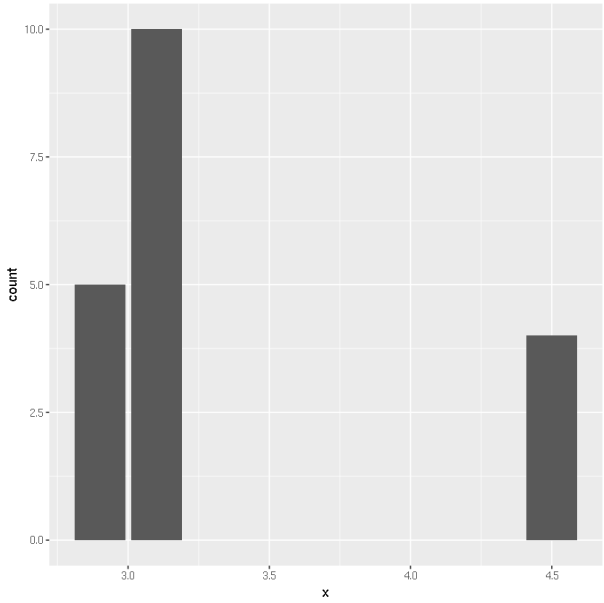

```
# help("geom_bar")
```

```
# Total engine displacement of each class
pl + geom_bar(aes(weight = displ))
```




```
# help("geom_bar")
```

```
# You can also use geom_bar() with continuous data, in
which case
# it will show counts at unique locations
df <- data.frame(x = rep(c(2.9, 3.1, 4.5), c(5, 10, 4)))
ggplot(df, aes(x)) + geom_bar()
```



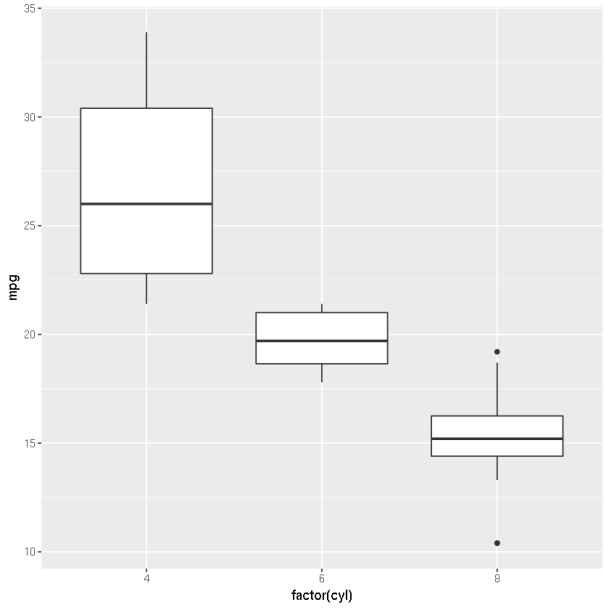
Boxplots

Digunakan untuk menampilkan sari statistik

```
df <- mtcars
head(df)
```

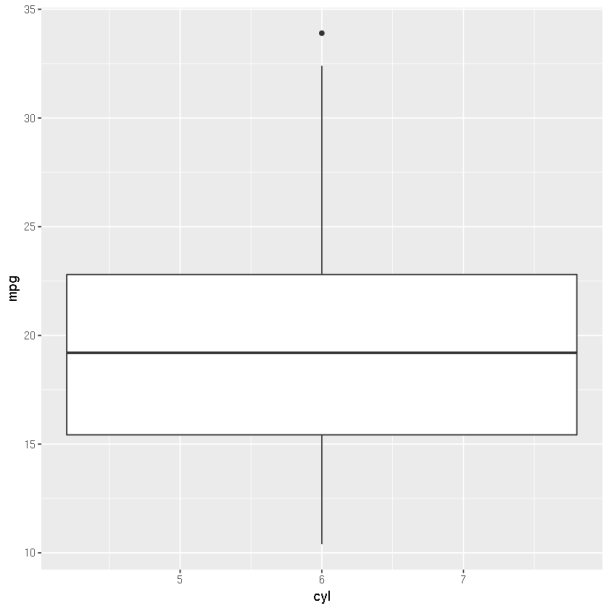
	MPG	CYL	DISP	HP	DRAT	WT	QSEC	VS	AM	GEAR	CARB
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

```
pl <- ggplot(df, aes(x = factor(cyl), y = mpg))
pl + geom_boxplot()
```



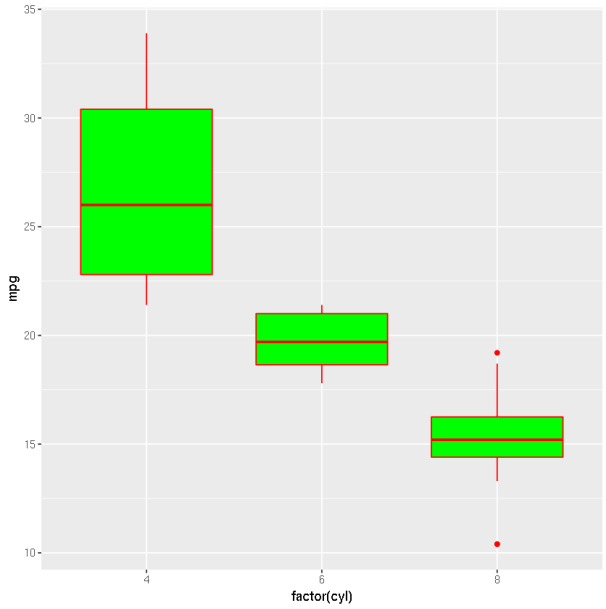
```
pl <- ggplot(df, aes(x = cyl, y = mpg)) # tanpa factor
pl + geom_boxplot()
```

```
Warning message:
“Continuous x aesthetic -- did you forget aes(group=...)?”
```

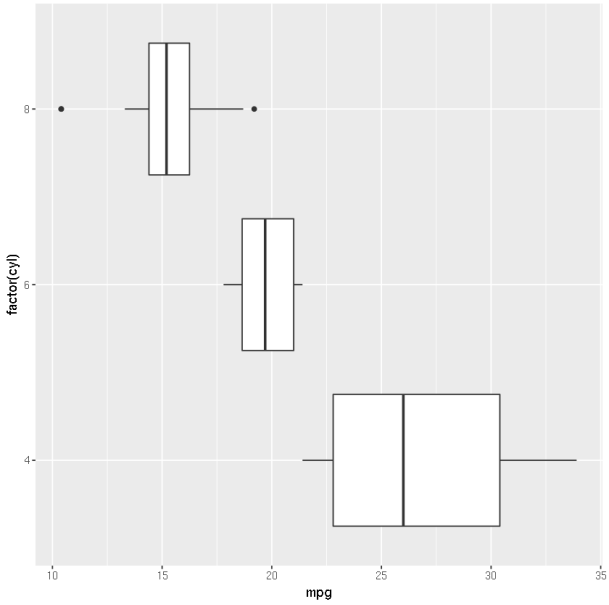



```
# help("geom_boxplot")
```

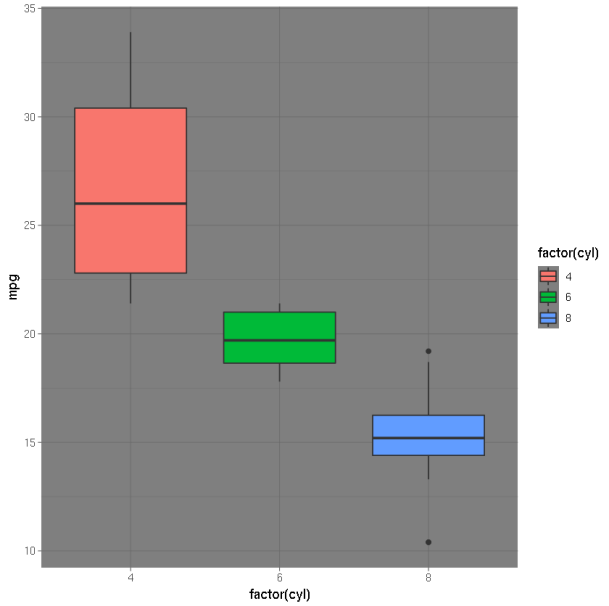
```
pl <- ggplot(df, aes(x = factor(cyl), y = mpg))
pl + geom_boxplot(color='red', fill = 'green')
```




```
# memutar koordinat
pl + geom_boxplot() + coord_flip()
```




```
pl + geom_boxplot(aes(fill=factor(cyl))) + theme_dark()
```



Visualisasi dua variabel

```
head(movies)
```

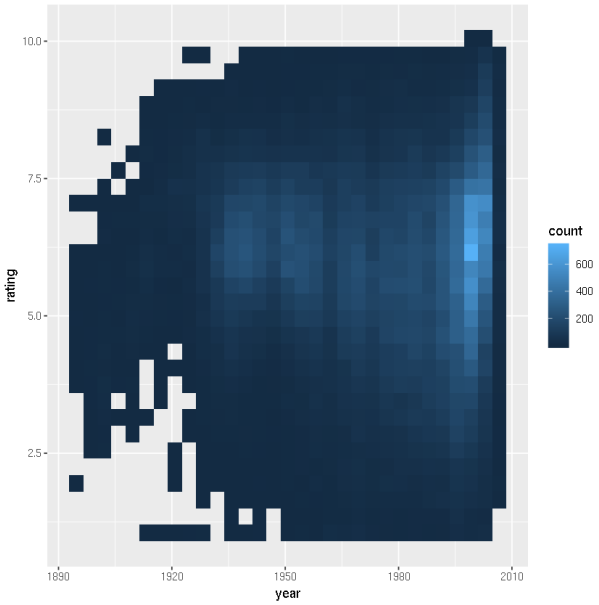
TITLE	YEAR	LENGTH	BUDGET	RATING	VOTES	R1	R2	R3	R4	...	R9	R10	MPAA	ACTION	ANIMATION	
$	1971	121	NA	6.4	348	4.5	4.5	4.5	4.5	...	4.5	4.5		0	0	
$1000 a Touchdown	1939	71	NA	6.0	20	0.0	14.5	4.5	24.5	...	4.5	14.5		0	0	
$21 a Day Once a Month	1941	7	NA	8.2	5	0.0	0.0	0.0	0.0	...	24.5	24.5		0	1	
$40,000	1996	70	NA	8.2	6	14.5	0.0	0.0	0.0	...	34.5	45.5		0	0	
$50,000 Climax Show, The	1975	71	NA	3.4	17	24.5	4.5	0.0	14.5	...	0.0	24.5		0	0	
$pent	2000	91	NA	4.3	45	4.5	4.5	4.5	14.5	...	14.5	14.5		0	0	

```
colnames(movies)
```

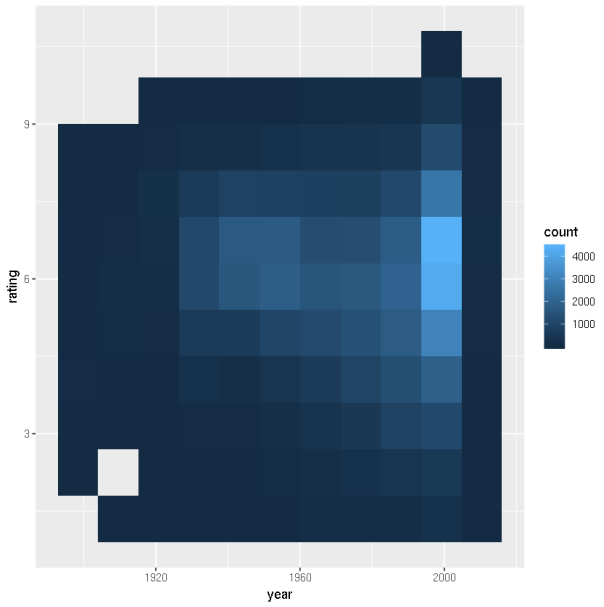
1. 'title'
2. 'year'
3. 'length'
4. 'budget'
5. 'rating'
6. 'votes'
7. 'r1'
8. 'r2'
9. 'r3'
10. 'r4'
11. 'r5'
12. 'r6'
13. 'r7'
14. 'r8'
15. 'r9'
16. 'r10'
17. 'mpaa'
18. 'Action'
19. 'Animation'
20. 'Comedy'
21. 'Drama'
22. 'Documentary'
23. 'Romance'
24. 'Short'

```
pl <- ggplot(movies, aes(x=year, y=rating))
```

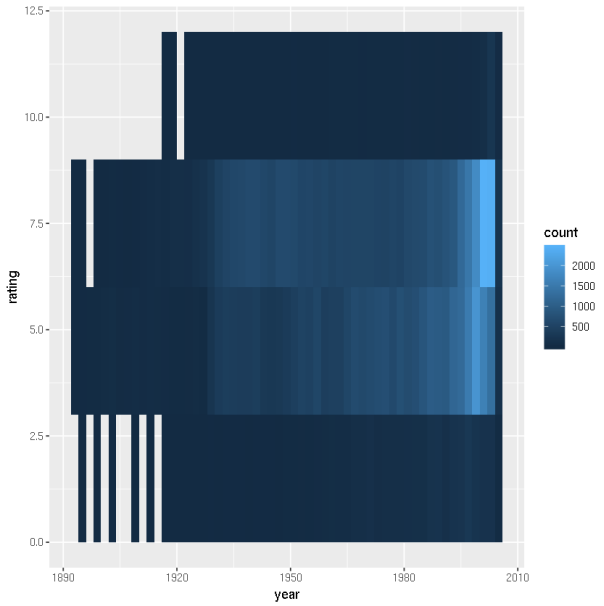
```
pl + geom_bin2d()
# mirip dengan heatmap
# jumlah kejadian dihitung berdasarkan warna
```




```
pl + geom_bin2d(bins=10)
```

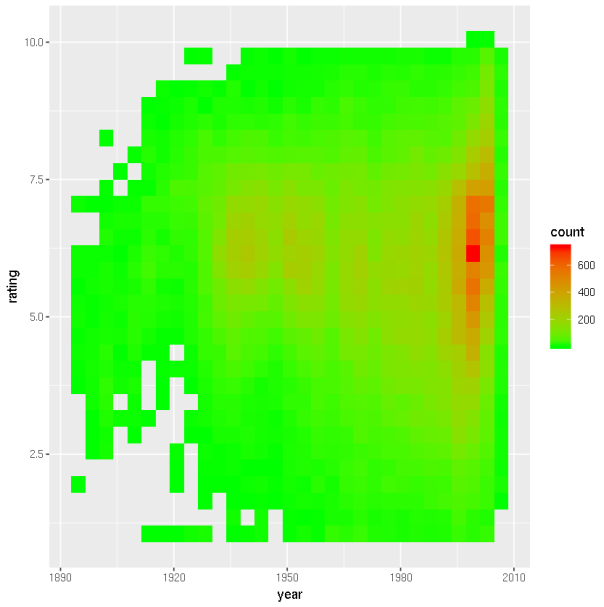



```
pl + geom_bin2d(binwidth=c(2,3), bins=10)
```

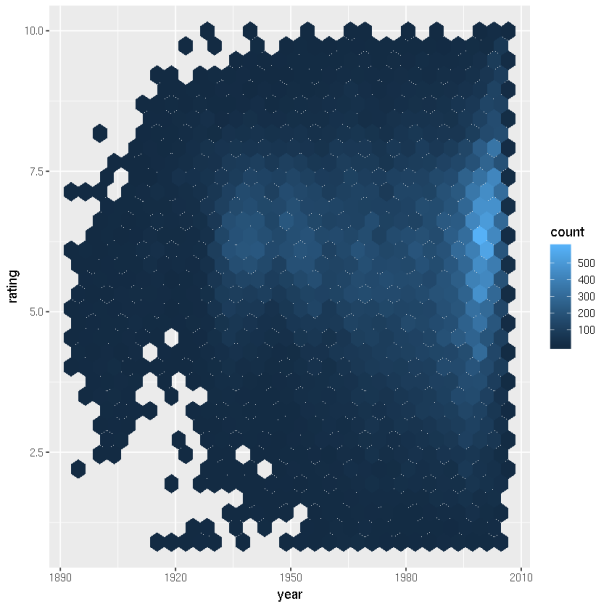



```
# help("geom_bin2d")
```

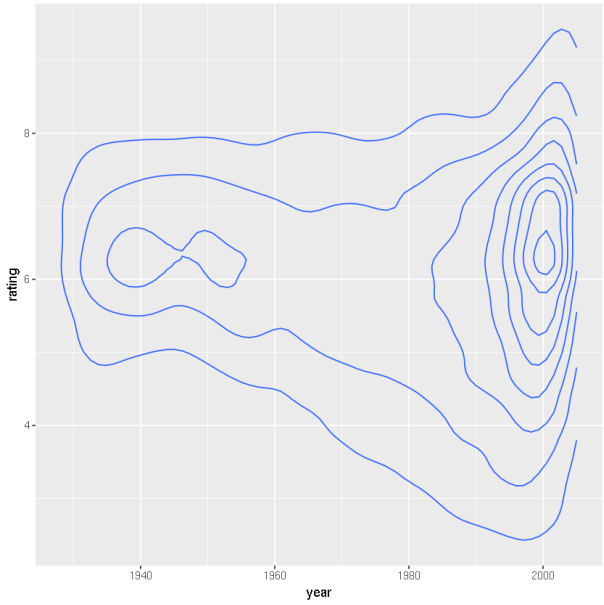
```
# mengubah warna
pl2 <- pl + geom_bin2d()
pl2 + scale_fill_gradient(high = 'red', low='green')
```




```
# mengubah shape jadi hexagon
library(hexbin)
pl + geom_hex()
```




```
pl + geom_density2d()
```



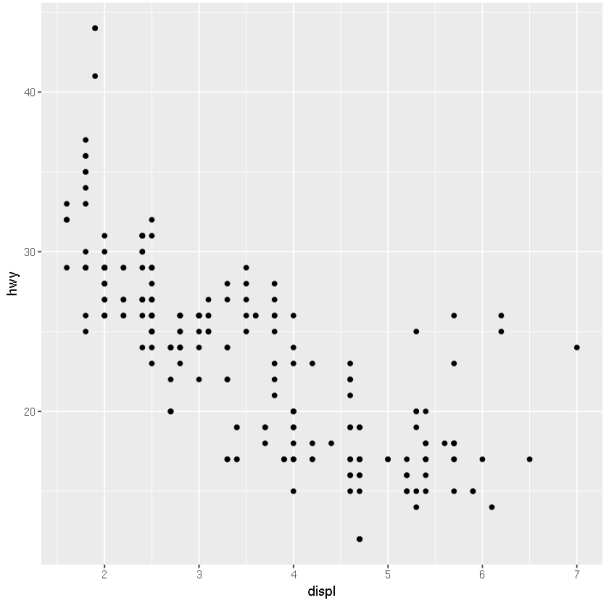
Koordinat dan *faceting*

```
head(mpg)
```

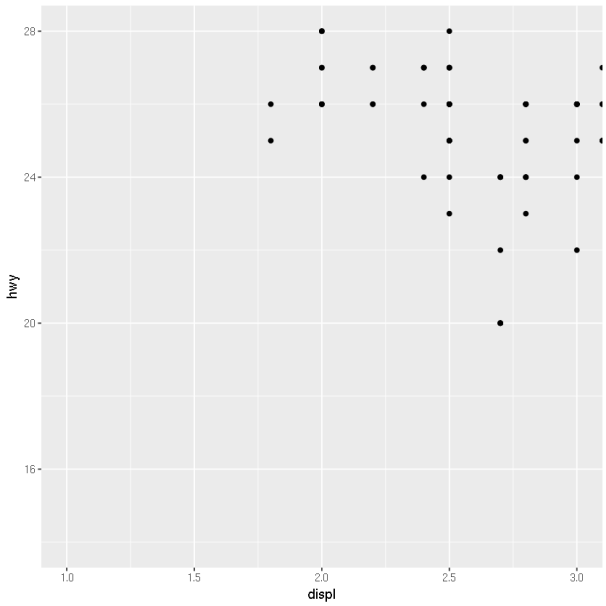
MANUFACTURER	MODEL	DISPL	YEAR	CYL	TRANS	DRV	CTY	HWY	FL	CLASS
audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact

```
# scatterplot simpel
pl <- ggplot(mpg, aes(x=displ, y=hwy)) +
            geom_point()
```

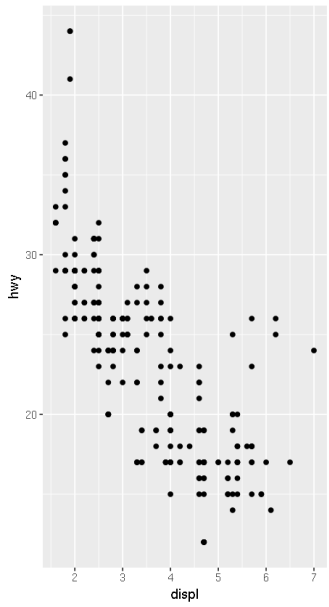
```
pl
```



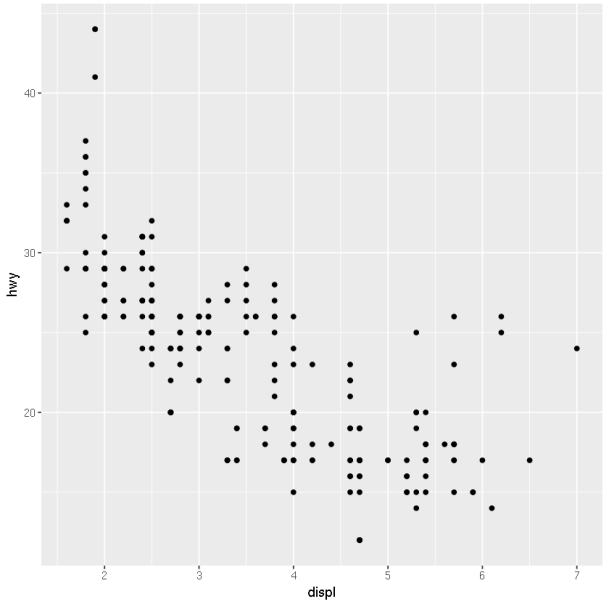

```
# Mengatur limit sumbu-x dan y
pl + coord_cartesian(xlim = c(1,3), ylim = c(14,28))
```



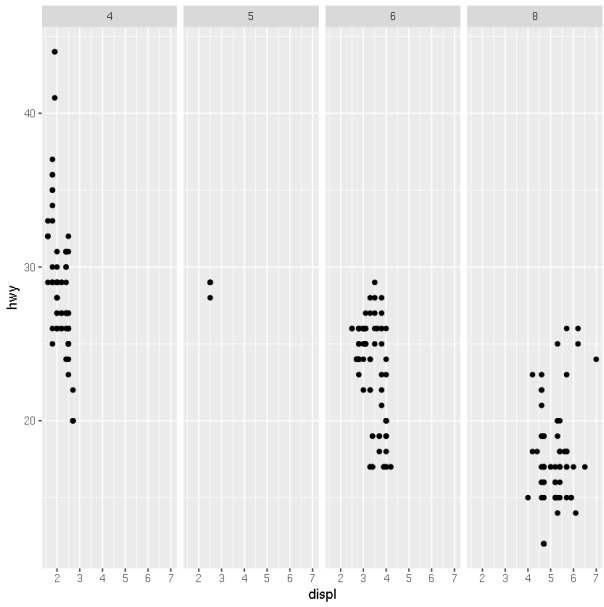

```
# Mengatur rasio sumbu
pl + coord_fixed(ratio = 1/3) # y/x
```




```
# Facets
pl
```

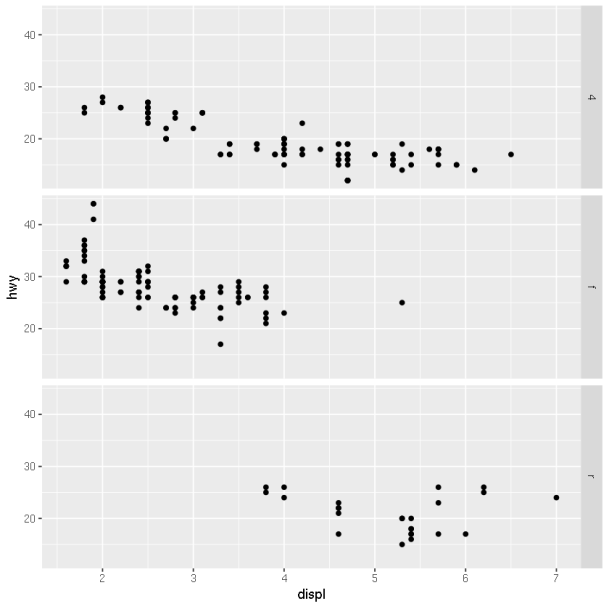



```
pl + facet_grid(.~cyl) # dipisahkan menurut silinder pada
sumbu-x
```

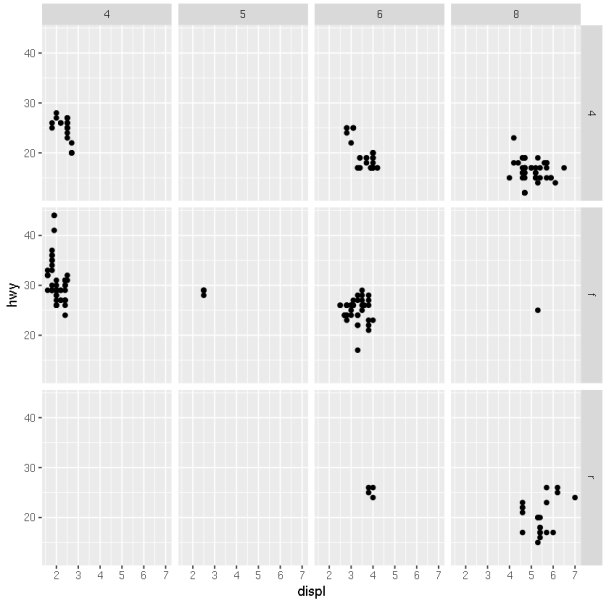


sintaks: `facet_grid(sb-x~sb-y)`

```
pl + facet_grid(drv~.) # membagi facet sumbu-y dengan
menggunakan drv
```

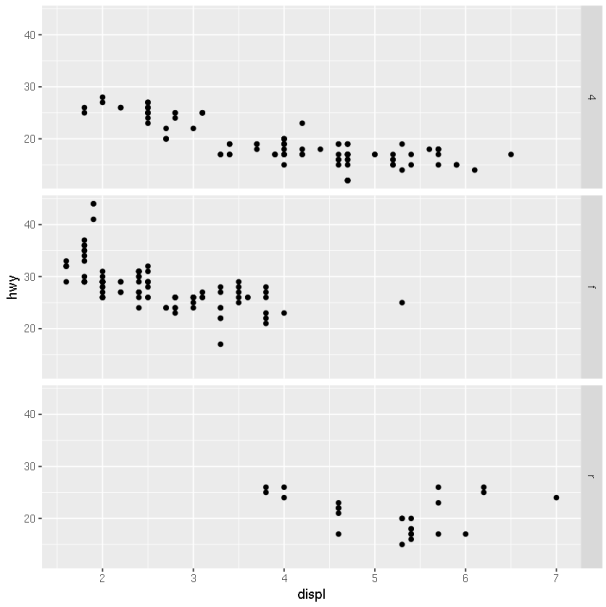



```
pl + facet_grid(drv~cyl)
```

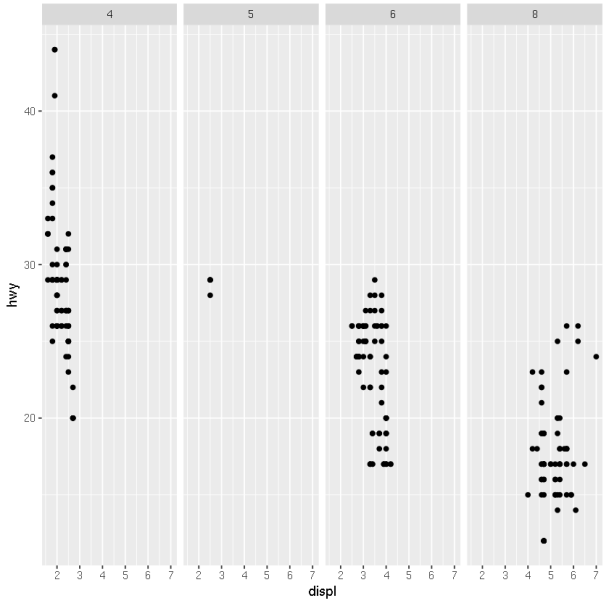



```
# Untuk mengetahui secara lebih lanjut, jalankan perintah:
help("facet_grid")
```

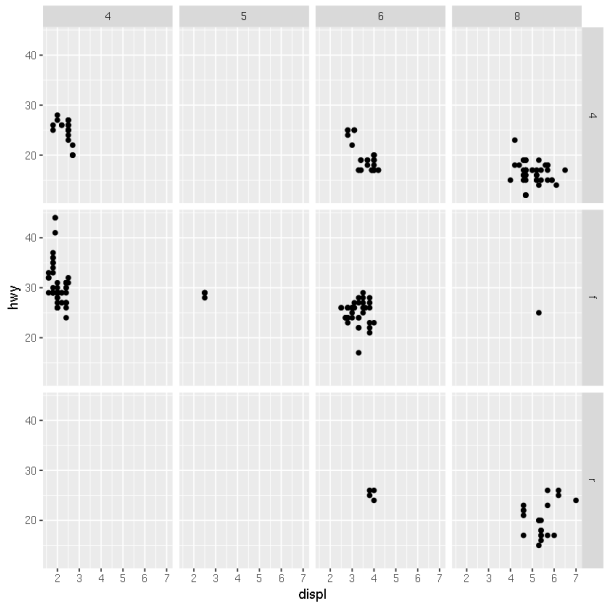
```
# Use vars() to supply variables from the dataset:
(berbasis baris, kolom)
pl + facet_grid(rows = vars(drv))
```




```
pl + facet_grid(cols = vars(cyl))
```




```
pl + facet_grid(vars(drv), vars(cyl))
```

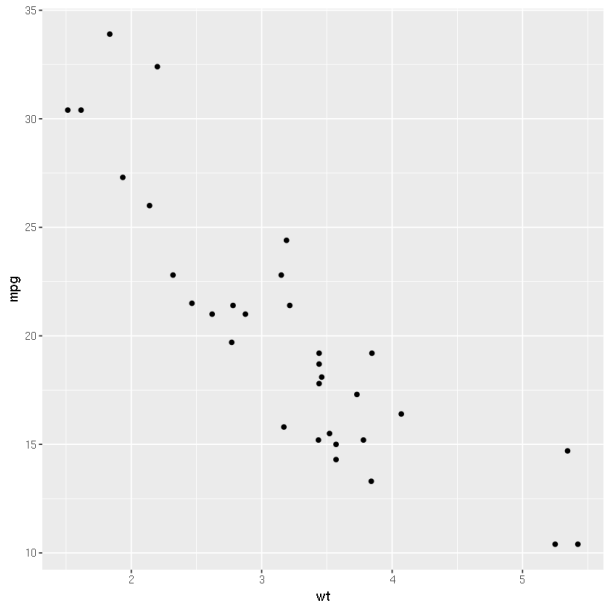


Tema

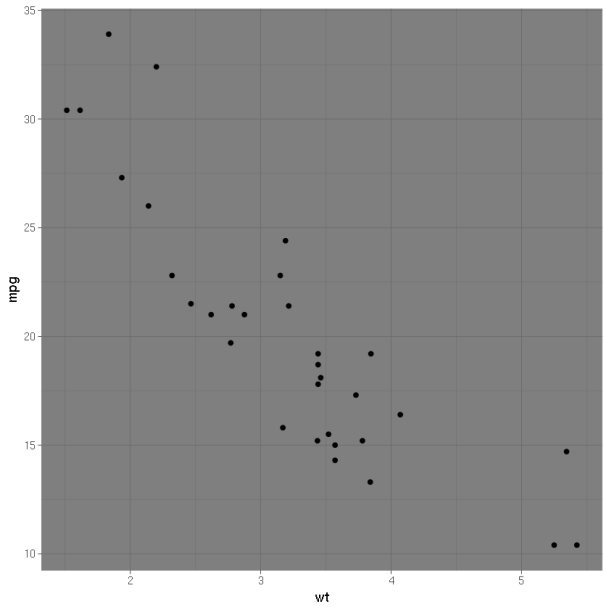
```
df <- mtcars
head(df)
```

	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

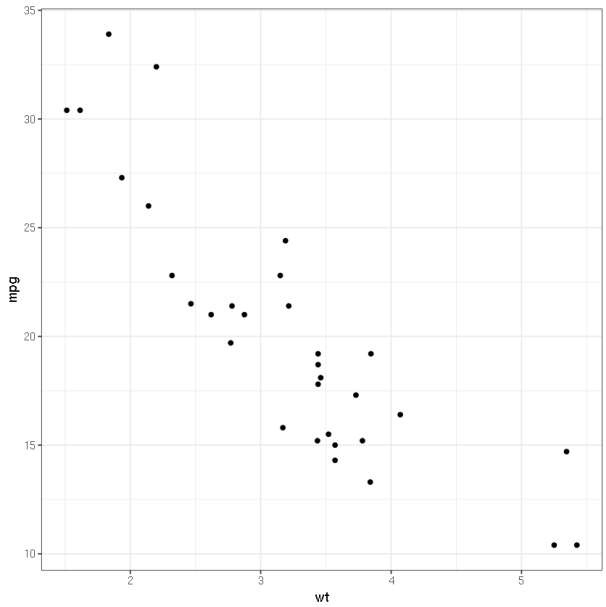
```
pl <- ggplot(df, aes(x=wt, y = mpg)) + geom_point()
pl
```



```
theme_set(theme_dark()) # mengatur tema untuk seluruh plot di dalam script
pl
```



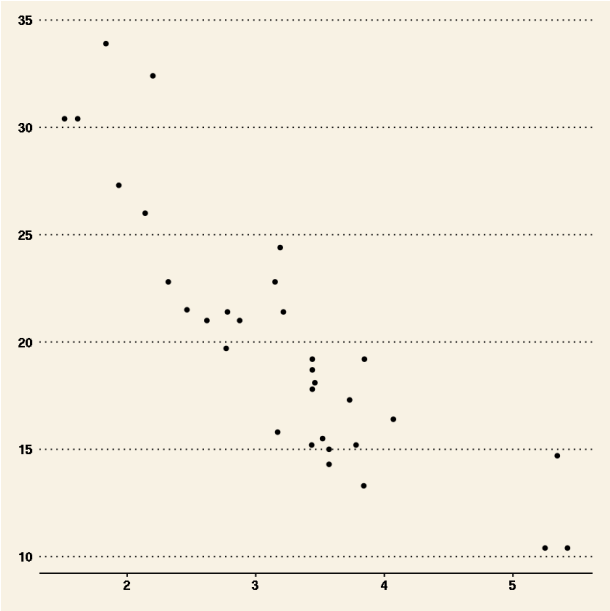

```
pl + theme_bw()
```



Untuk tema tambahan kita dapat menjalankan perintah sebagai berikut:

```
library(ggthemes)
```

```
pl + theme_wsj() # tema dari Wall Street Journal
```


35
30
25
20
15
10
2
3
4
5