BAB

V

R untuk Pengolahan & Analisis Statistik

V.1. Statistika deskriptif

V.1.1. Rerata (*Mean*)

Rerata merupakan deskripsi statistika yang menggambarkan tentang nilai rata-rata dari suatu sample. Perhitungan rerata secara matematis adalah sebagai berikut:

$$mean = 1/n * \sum_{i}^{n} x_i$$

Dalam R terdapat fungsi untuk menghitung nilai rerata sampel. Fungsi yang digunakan adalah `mean(x)` ataupun dengan menggunakan fungsi `summary(x)`. Sebagai contoh, digunakan data dari *datapackage* yang sudah tersedia di R, pilih salah satu data (misalkan Nile). Kemudian hitung nilai rerata sampel, dengan menuliskan

```
> data()
> data(Nile)
> Nile
Time Series:
Start = 1871
End = 1970
Frequency = 1
[1]  1120 1160 963  1210 1160 1160 813  1230 1370 1140 995  935  1110
[14] 994  1020 960  1180 799  958  1140 1100 1210 1150 1250 1260 1220
[28] 1030 1100 774  840  874  694  940  833  701  916  692  1020 1050
[41] 969  831  726  456  824  702  1120 1100 832  764  821  768  845
[54] 864  862  698  845  744  796  1040 759  781  865  845  944  984
[67] 897  822  1010 771  676  649  846  812  742  801  1040 860  874
[80] 848  890  744  749  838  1050 918  986  797  923  975  815  1020
[94] 906  901  1170 912  746  919  718  714  740
```

```
> mean(Nile)
[1] 919.35
> summary(Nile)
  Min.   1st Qu.   Median   Mean    3rd Qu.   Max.
  456.0  798.5     893.5    919.4   1033.0    1370.0
```

Hasil tersebut di atas adalah menunjukkan tentang data Nile yang terdiri dari 100 data dengan nilai rerata 919.35. Selain rerata ada pula nilai statistik lainnya, yaitu minimum, kuartil pertama, nilai tengah (median), kuartil ke tiga dan maksimum. Nilai-nilai tersebut ditampilkan dengan menjalankan fungsi `summary()`.

V.1.2. Nilai Tengah (*median*)

Seperti halnya dengan rerata, R juga menyediakan fungsi untuk mencari nilai tengah (*median*) sampel dengan menuliskan fungsi `median()`.
Sebagai contoh, dapat digunakan data seperti mencari nilai rerata sebelumnya untuk menghitung nilai tengah (median):

```
> median(Nile)
[1] 893.5
```

V.1.3. Nilai minimum dan maksimum

R menyediakan fasilitas untuk mencari nilai minimum dan maksimum suatu data, yaitu dengan digunakan perintah `min()` dan `max()`

```
> min(x)     # untuk nilai minimum, dan
> max(x)    # untuk nilai maksimum
```

V. 2. Grafik

R dilengkapi dengan fasilitas untuk visualisasi statistik dalam bentuk grafik, baik statistik, kontur, map, dll. Sistem grafik di R terdiri dari dua sistem: sistem (dasar/default) yang terdapat dalam paket `graphics` dan sistem *trellis* yang terdapat dalam paket `lattice`. Grafik di R dapat diatur sesuai keperluan. Untuk melihat lebih detil mengenai fitur grafik dalam R, pilih menu Help di menu R kemudian pilih Manual (dalam format *pdf*) atau HTML help. Untuk mendapatkan gambaran langsung tentang grafik dalam R, dapat dilihat dalam fungsi `demo()`, dengan menuliskan

```
> demo()              # untuk melihat jenis-jenis demo
> demo(graphics)      # atau
> demo(image)         # atau
> demo(persp)         # atau
> demo(lattice)       # sebelumnya diperlukan load package dan pilih
                        lattice pada menu di windows R anda; atau
> demo(package = .packages(all.available = TRUE))  # untuk melihat
                        semua jenis demo yang tersedia
```

Ket: untuk R versi 2.1.1, penulisan `Return` *dituliskan setelah penulisan fungsi* `demo()`

Grafik Dasar (*Base*)

Pengantar tentang prosedur grafik dapat dilihat di dokumen "Introduction to R" pada menu **Help ->Manual** (dalam format pdf).

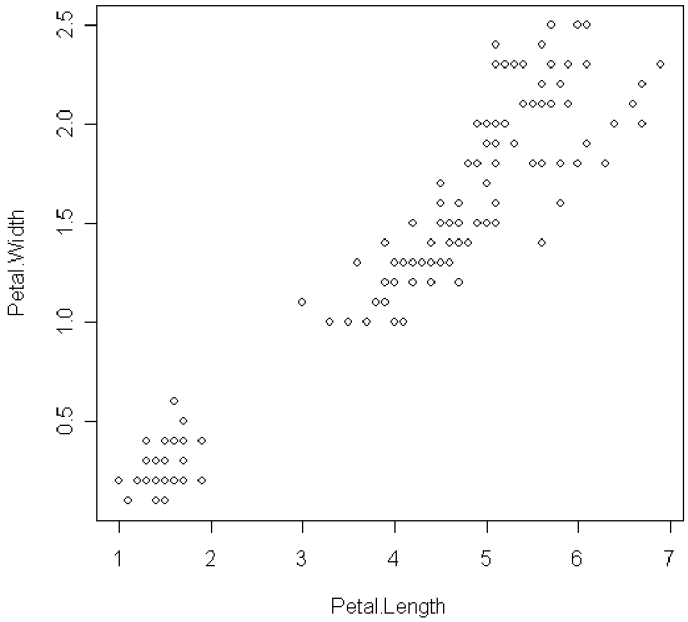
Berikut akan disajikan contoh pembuatan plot secara bertahap diawali dengan model standar hingga pengaturan sesuai dengan yang diinginkan (customize). Contoh berikut adalah pembuatan *scatterplot* untuk `petal.length` yang dibandingkan dengan `petal.width` dari dataset `iris`. Default scatterplot dari dua variabel dihasilkan oleh metod `plot.default`, yang secara otomatis digunakan oleh perintah plot generik dimana argumennya merupakan dua vektor dengan panjang yang sama seperti berikut ini:

```
> data(iris)
> str(iris)

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

Data tersebut di atas menyatakan bahwa data iris terdiri dari 5 variabel dimana setiap variable terdiri dari 150 data observasi. Lima variable tersebut adalah: Sepal.Length, Sepal.Width, Petal.Length, Petal.Width dan Species.

```
> attach(iris)
> plot(Petal.Length, Petal.Width)
```

Hasil dari perintah tersebut terlihat pada Grafik 5.1 (merupakan bentuk grafik secara default). Dalam grafik tersebut, sumbu x dan sumbu y berasal dari argumen pertama (`Petal.Length`) dan argumen kedua (`Petal.Width`), dari pernyataan `plot(`*`argument_1, argument_2`*`)`.



Gambar 5.1: Scatter plot data variabel Petal

Grafik 5.1 di atas dapat dilengkapi untuk menunjukan dependensi argumen dimana sumbu *y* sebagai variabel dependen. Hal tersebut dilakukan dengan menuliskan perintah seperti berikut, dimana variable terikat terletak sebelah kiri:

```
> plot(Petal.Width ~ Petal.Length,)
```

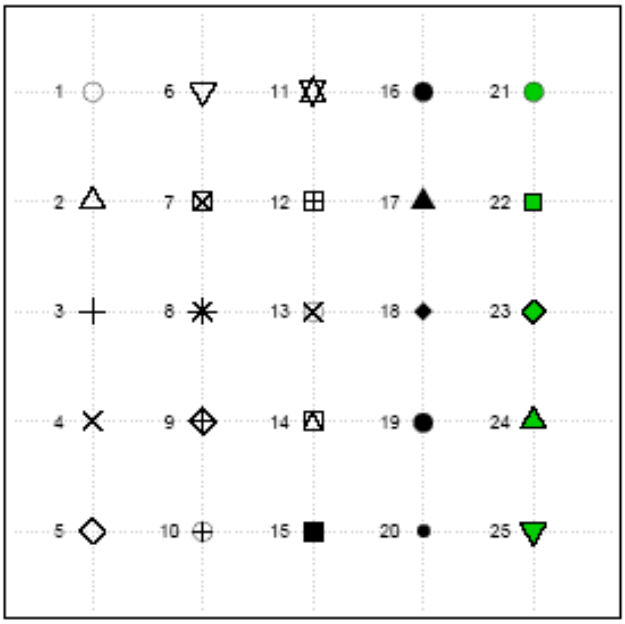
Pada Gambar 5.1 bentuk grafik sangat standart, sehingga perlu dilengkapi dengan beberapa keterangan tambahan untuk memperjelas dan mempermudah dalam melakukan interpretasi grafik. Hal ini dapat dilakukan dengan menambahkan fitur warna atau simbol dalam tampilan grafik. Untuk hal tersebut, R mempunyai fasilitas pewarnaan (yaitu dengan argumen `col`), simbol (dengan argumen `pch`), ukuran (dengan argumen `cex`), label/nama sumbu kordinat (dengan argumen `xlab` dan `ylab`), judul grafik (dengan argumen `main`). Beberapa jenis warna yang disediakan dalam R dapat diketahui dengan menggunakan perintah `colours()`, dimana akan ditampilan daftar warna-warna tersebut.

```
> colours()

[1]   "white"            "aliceblue"      "antiquewhite"
[4]   "antiquewhite1"    "antiquewhite2   " antiquewhite3"
…
[655] "yellow3"          "yellow4"        "yellowgreen"
```

sedangkan fungsi `palette()` dapat digunakan untuk menampilkan daftar warna dalam bilangan numeric:

```
> palette()

[1] "black" "red" "green3" "blue" "cyan" "magenta" "yellow"
[8] "gray"
```

Selain warna, simbol juga dapat digunakan untuk menampilkan tanda plot. Spesifikasi simbol dapat dilakukan dengan menentukan karakter yang akan digunakan (misalkan asterik ”*”) atau kode integer dari simbol tersebut. Gambar 5.2 menunjukkan simbol dan kodenya. Simbol dengan kode 21-26 memiliki *fill* (warna latar) yang dispesifikasikan pada argumen `bg`, spesifikasi warna utama dengan argumen `col` menspesifikasikan garis border.

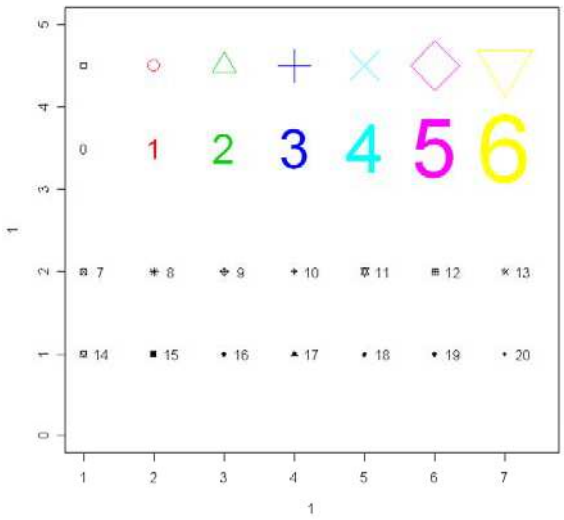


Gambar 5.2: Simbol dan kode dalam R

Berikut ini merupakan contoh yang menampilkan grafik plot yang dilengkapi dengan pewarnaan, modifikasi ukuran dan pemilihan simbol *plotting*. Penulisannya adalah sebagai berikut:

```
> plot(1, 1, xlim=c(1, 7.5), ylim=c(0,5), type="n")
> points(1:7, rep(4.5, 7), cex=1:7, col=1:7, pch=0:6)
> text(1:7,rep(3.5, 7), labels=paste(0:6), cex=1:7, col=1:7)
> points(1:7,rep(2,7), pch=(0:6)+7) # Plot simbol 7 hingga 13
> text((1:7)+0.25, rep(2,7), paste((0:6)+7)) # Label dengan bilangan simbol
> points(1:7,rep(1,7), pch=(0:6)+14) # Plot symbols 14 hingga 20
> text((1:7)+0.25, rep(1,7), paste((0:6)+14)) # Labels dengan bilangan simbol
```

dan akan menghasilkan Gambar 5.3 sebagai berikut:

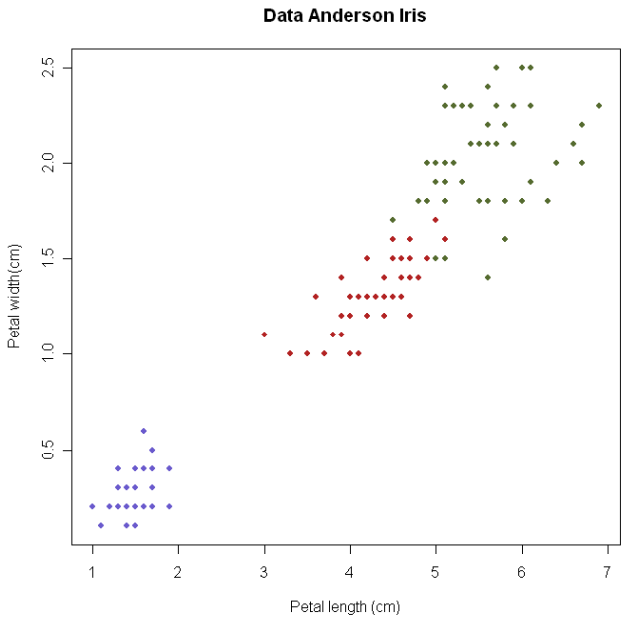


Gambar 5.3: Simbol, Kode dan Warna dalam R

Setelah kita mengetahui bentuk dan kode simbol, maka berikut ini akan dilakukan penggunaan simbol, warna dan modifkasi lain untuk menggambarkan plot/grafik dari contoh sebelumnya dengan menuliskan perintah seperti berikut:

```
> plot(Petal.Length, Petal.Width, pch=20, cex=1.2,
+ xlab=" Petal length (cm)", ylab="Petal width (cm)",
+ main="Data Anderson Iris",
+ col=c("slateblue","firebrick",darkolivegreen")
+ [as.numeric(Species)])
```

Data menunjukkan dengan jelas bahwa ukuran spesies berbeda (Sentosa paling kecil, Versicolor menengah, Virginica terbesar) tetapi rasio petal length dan weight sama untuk ketiga ukuran tersebut.



Gambar 5.4: Grafik *scatterplot* data *Iris*

V.2.1. Histogram

Selain plot, bentuk representasi grafis lainnya yang paling mudah digunakan untuk menggambarkan sebaran data adalah histogram. R menyediakan fasilitas fungsi histogram yang digunakan untuk mengetahui sebaran sampel suatu data. Sebagai catatan: histogram ataupun boxplot, digunakan untuk satu variable.
Sebelum kita mencoba untuk menggunakan fasilitas histogram, maka perlu sedikit penjelasan yang berkaitan dengan histogram, yaitu:

- histogram digunakan untuk mengestimasi fungsi distribusi probabilitas densitas (*probability density function*);

$$f(x) = \lim_{\delta \to 0} \Pr ob(x - \delta < X \le x) / \delta$$

- histogram ditentukan pula oleh bin/lebar batang;
- sumbu-y dalam histogram dapat berupa frekuensi kemunculan atau proporsi;
- tidak ada estimasi statistik yang dapat dibaca langsung dari histogram, namun
- dengan histogram kita dapat menduga kemiringan, sifat/behavior pada tail atau ujung kurva, dan outlier data;
- histogram dapat dibandingkan sebagai suatu distribusi analitik standar.

Selain histogram, R juga menyediakan plot yang fungsinya menyerupai histogram yakni **stem-and-leaf plot** dengan penulisan:

```
> stem(variabel)
```

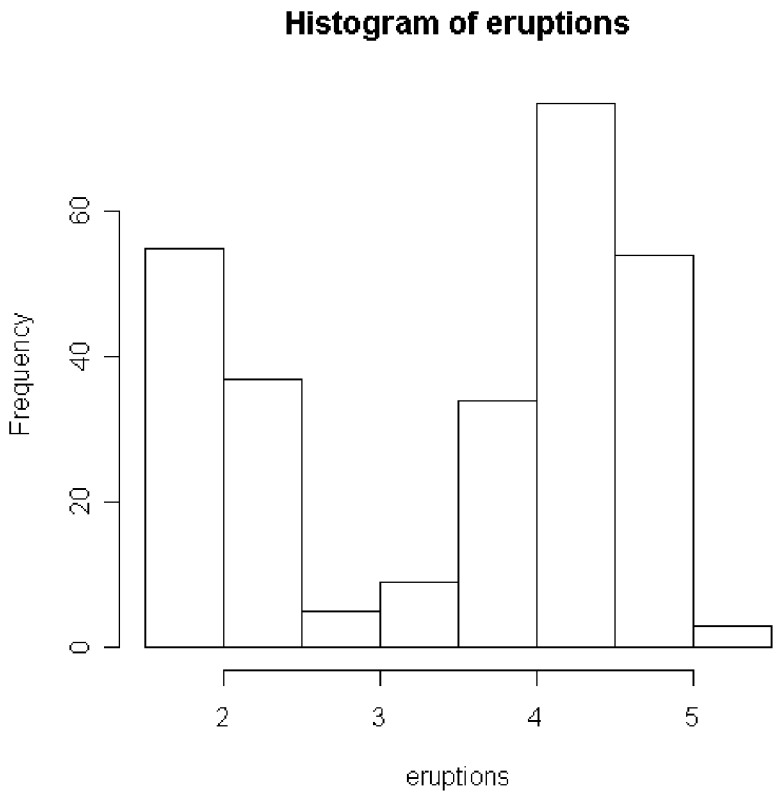
Sebagai contoh, gunakan variabel **eruptions** dari tabel data **faithful**, dengan menuliskan:

```
> attach(faithful)
> summary(eruptions)

  Min.   1st Qu. Median  Mean   3rd Qu.  Max.
  1.600  2.163   4.000   3.488  4.454    5.100

> fivenum(eruptions)

[1] 1.6000 2.1585 4.0000 4.4585 5.1000

> stem(eruptions)

      The decimal point is 1 digit(s) to the left of the |
16 | 070355555588
18 | 000022233333335577777777888822335777888
20 | 00002223378800035778
22 | 0002335578023578
24 | 00228
26 | 23
28 | 080
30 | 7
32 | 2337
34 | 250077
36 | 0000823577
38 | 2333335582225577
```

```
40 | 000000335778888800223355557778
42 | 033355557788002333335555777 8
44 | 022223355577800000000233335777888 8
46 | 00002333577000000235 78
48 | 00000022335800333
50 | 0370
```

Kita juga dapat melihat sebaran data dalam plot histogram yaitu dengan menggunakan fungsi `hist()`

```
> hist(eruptions)
```

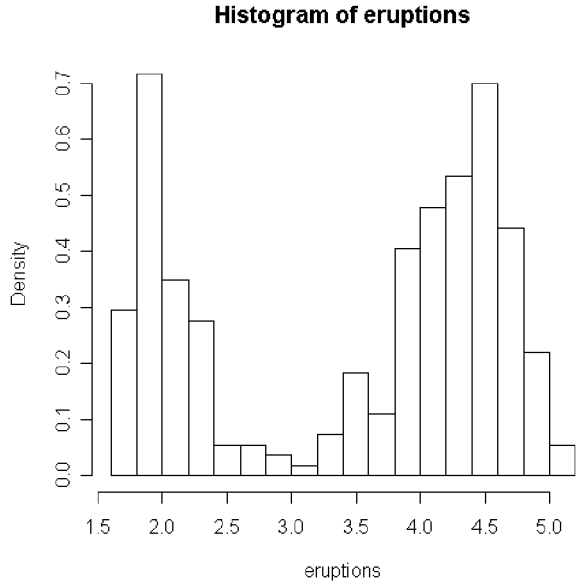


Gambar 5.5: Grafik histogram data *eruptions*

Pada Gambar 5.5 di atas, fungsi `hist()` menggunakan jarak antar batang (disebut *bin*) cukup besar. Untuk membuat *bin* lebih kecil, diperlukan tambahan atribut dengan menuliskan:

```
> hist(eruptions, seq(1.6, 5.2, 0.2), prob=TRUE)
```

Pada perintah di atas, argumen `seq(1.6, 5.2, 0.2)` adalah histogram menggunakan range dari 1.6 hingga 5.2 dengan lebar *bin* 0.2. Sehingga tampilan grafik histogram adalah sebagai berikut:

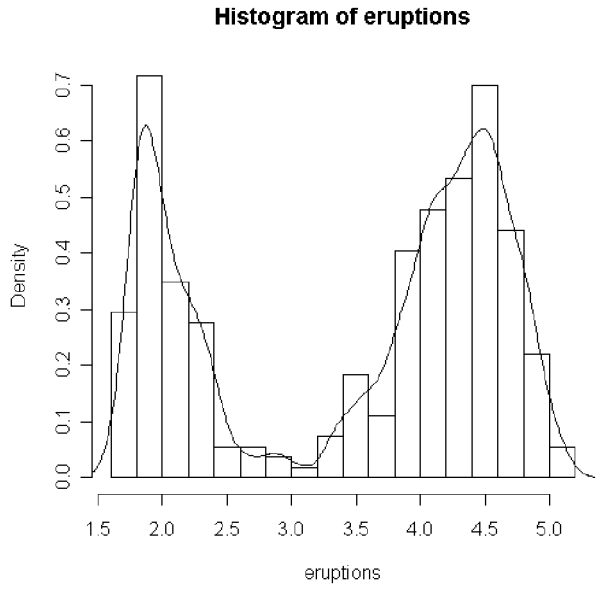


Gambar 5.6: Grafik histogram data *eruption* berdasarkan lebar *bin* 0.2

Gambar 5.6 di atas menunjukkan lebar batang histogram yang lebih kecil dibanding Gambar 5.5. Apabila ingin ditambahkan garis pada data densitas, maka dapat menggunakan fungsi `lines()` seperti berikut:

```
> lines(density(eruptions, bw = 0.1))
```

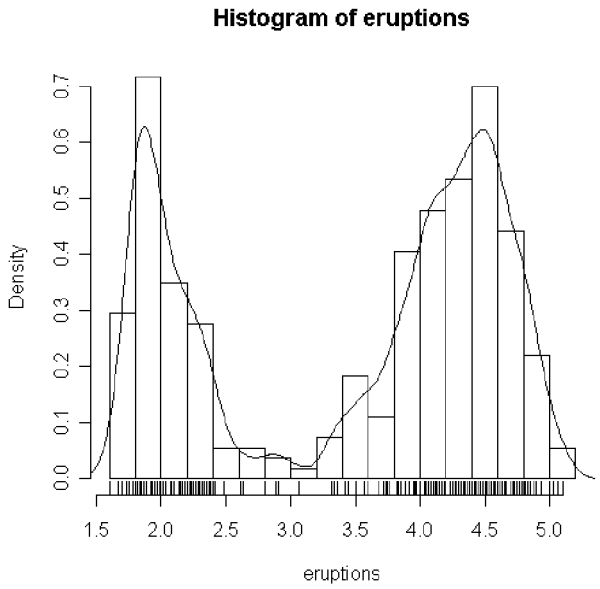
Keterangan: `bw` adalah *bandwidth* (lebar pita), dengan nilainya berdasarkan *trial and error*.



Gambar 5.7: Grafik histogram data *eruptions* dengan *bw* = 0.1

Untuk menampilkan point data aktual digunakan fungsi `rug()` sbb:

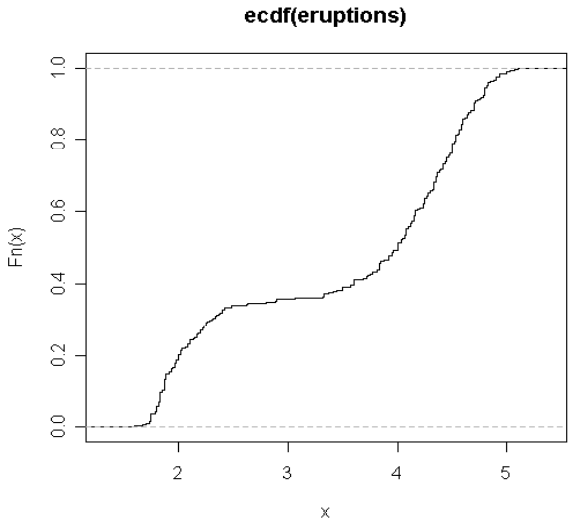
```
> rug(eruptions)
```



Gambar 5.8: Grafik histogram data *eruption* dengan data aktual

Selain histogram, R dapat memplot fungsi distribusi kumulatif empiris dengan menggunakan fungsi `ecdf()`:

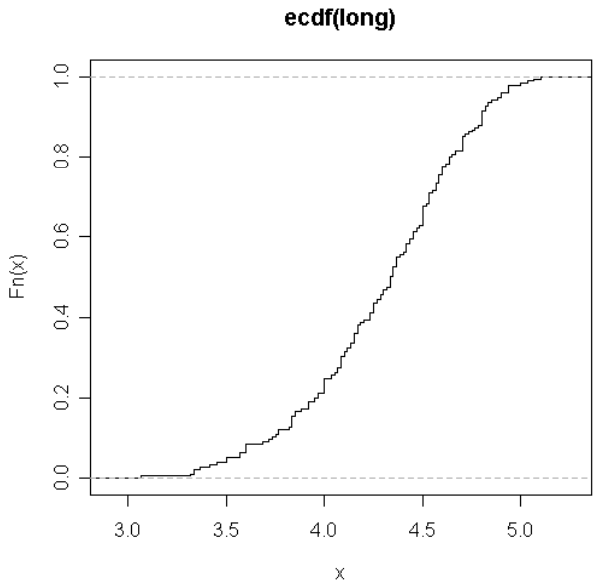
```
> plot(ecdf(eruptions), do.points=FALSE, verticals=TRUE)
```



Gambar 5.9: Plot 1 ecdf data *eruption*

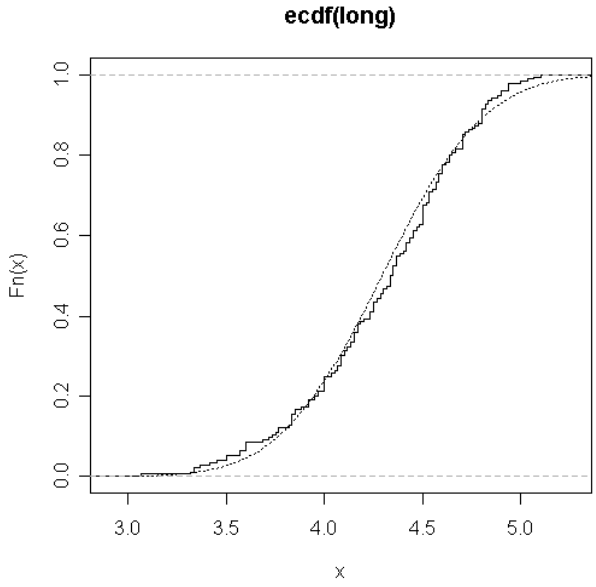
Distribusi `ecdf` di atas masih jauh dari standar distribusi yang ada. Untuk itu dapat dicobakan dengan mencocokkan distribusi normal dan "menutupi" fungsi distribusi kumulatif (`ecdf`) sebelumnya. Penulisannya adalah sebagai berikut:

```
> long <- eruptions[eruptions > 3]
> plot(ecdf(long), do.points=FALSE, verticals=TRUE)
```



Gambar 5.10: Plot 2 ecdf data *eruption*

```
> x <- seq(3, 5.4, 0.01)
> lines(x, pnorm(x, mean=mean(long), sd=sqrt(var(long))), lty=3)
```



Gambar 5.11: Plot 3 ecdf data *eruption*

V.2.2. Q-Q (Quantile – Quantile)

Selain histogram sebagai alat untuk memplot sebaran data suatu variabel adalah Quantile – Quantile (Q-Q) plot. Q-Q plot dapat digunakan untuk memplot variable secara lebih teliti berdasarkan nilai quantile data.
Q-Q plot adalah suatu scatter plot yang membandingkan distribusi empiris dengan *fitted distribution* dalam kaitannya dengan nilai dimensi suatu variabel (misalkan: nilai quantile empiris). Q-Q plot dapat memplot dengan baik jika dataset diperoleh dari populasi yang sudah diketahui.

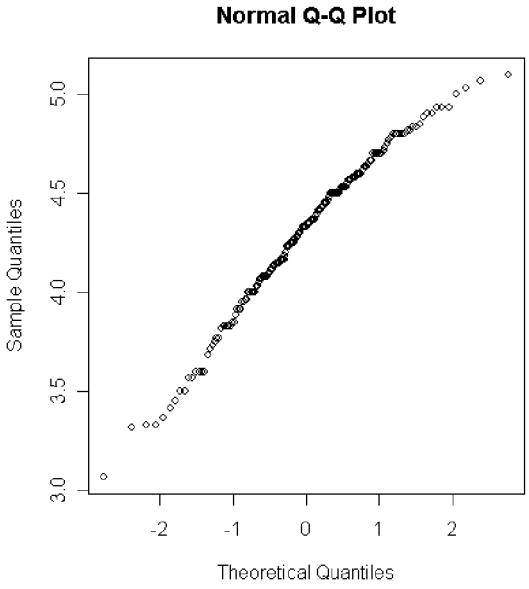
Q-Q plot dalam R dibagi menjadi dua, yaitu:

- `qqnorm(`***`variabel`***`)`; untuk menguji *goodness of fit* dari distribusi Gaussian. `qqnnorm()` disebut juga sebagai plot probabilitas normal.

- `qqplot(`***`variabel`***`)`; untuk sebarang jenis distribusi

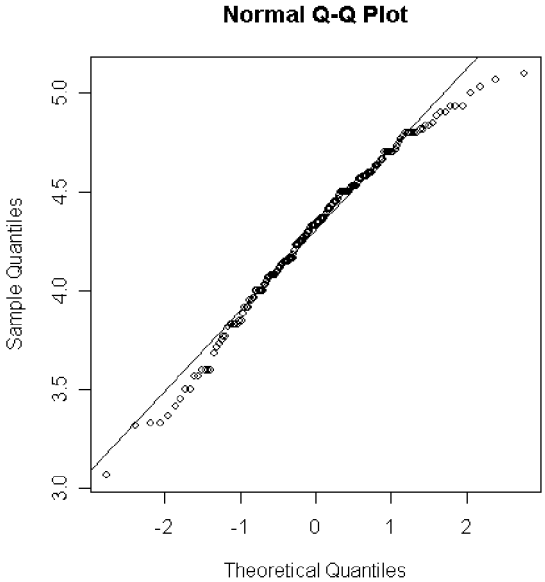
Melanjutkan contoh sebelumnya, untuk data **`eruptions,`** penggunaan Q-Q plot dituliskan sbb:

```
> par(pty="s") # mengatur pembuatan daerah gambar berbentuk kotak
> qqnorm(long)
```



Gambar 5.12: Plot1 *Q-Q Normal*

```
> qqline(long)
```



Gambar 5.13: Plot 2 *Q-Q Normal*

V.2.3. Boxplot

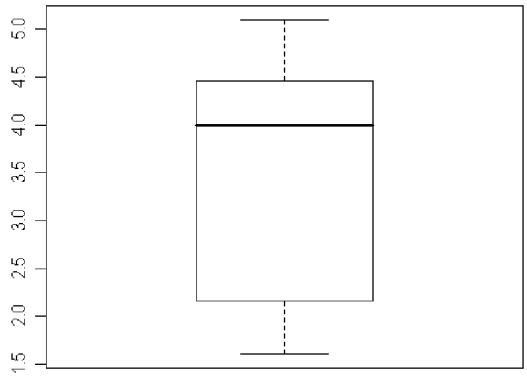
Selain dua alat untuk menggambarkan grafik untuk satu variable yang sudah dijelaskan sebelumnya, terdapat fasilitas `boxplot` yang digunakan untuk melihat sebaran data. Berikut adalah penjelasan tentang fitur dasar `boxplot`:

- Berguna untuk membanding banyak kelompok/grup.
- Dasarnya menggunakan 3 jenis summary: 3 quartil.
- Mudah dalam menampilkan nilai rerata (*mean*).
- Dapat diperluas untuk menampilkan persentil lainnya, terutama pada ujung(*tails*) suatu distribusi.

R menyediakan fitur untuk menampilkan boxplot, dengan menuliskan fungsi `boxplot(variabel)`.
Untuk menjelaskan penggunaan fungsi `boxplot()`, berikut adalah contoh menggambar grafik dengan menggunakan data **`faithful`** seperti pada contoh sebelumnya.

```
> boxplot(eruptions)
```

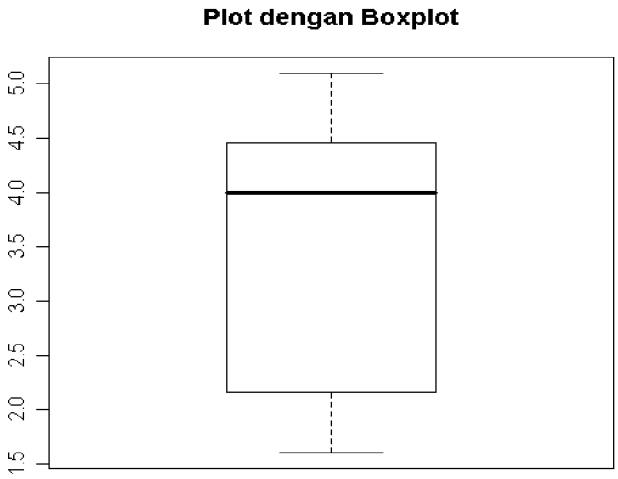
Perintah di atas akan mendapatkan gambar boxplot dari variabel eruption seperti berikut:



Gambar 5.14: Boxplot data *eruption*

Apabila ingin menambahkan judul gambar, maka penulisannya adalah:

```
> boxplot(eruptions, main="Plot dengan Boxplot")
```



Gambar 5.15: Boxplot data *eruption* dengan nama titel

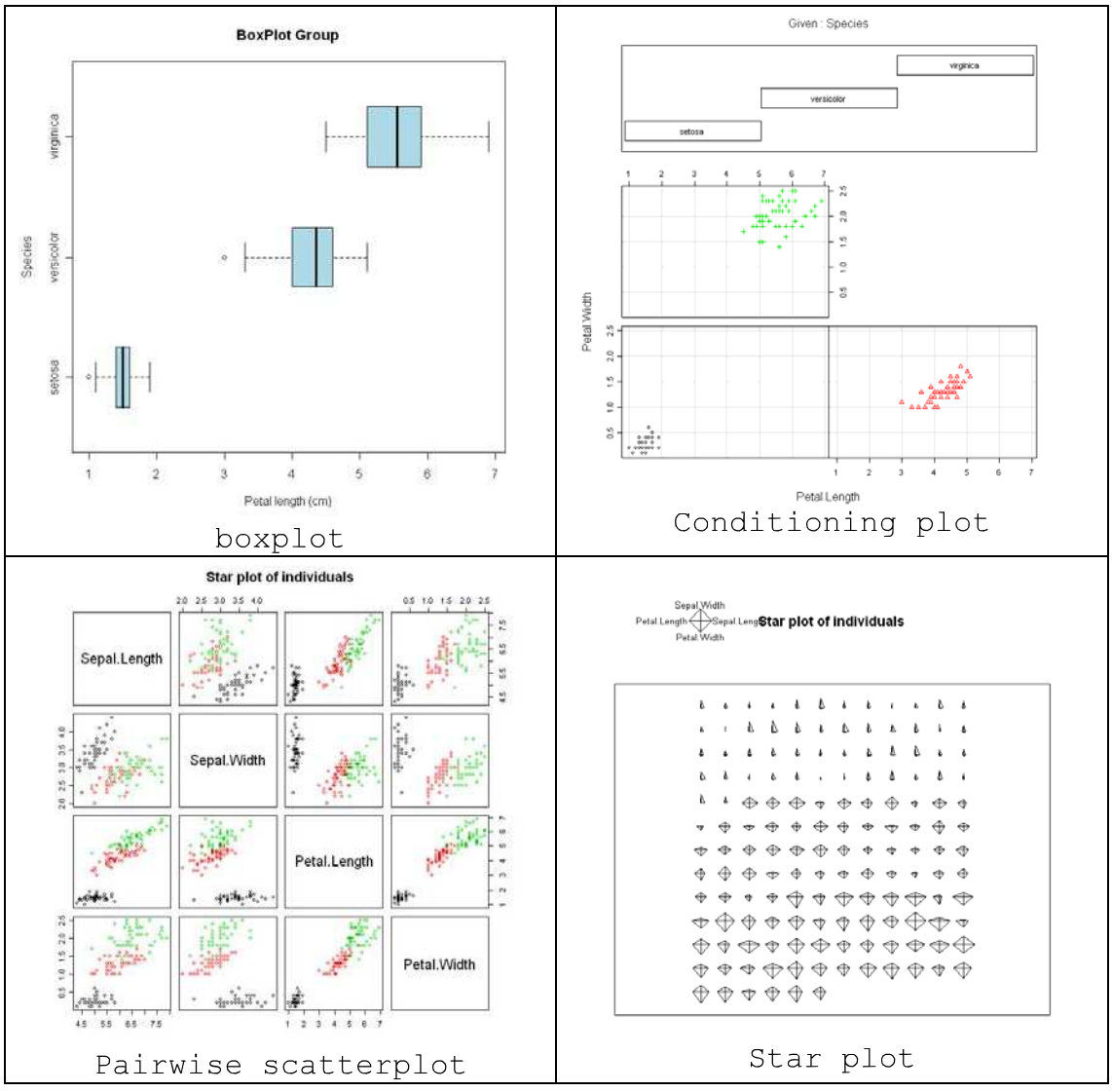
R menyediakan beragam bentuk penyajian grafik plot. Berikut adalah daftar plot grafik dasar yang ada dalam R (beberapa ada yang termasuk dalam instalasi awal dan adapula yang masuk dalam paket lain yang harus didownload dari CRAN):

Tabel 5.1: Jenis plot dalam R

Jenis Fungsi Plot	Keterangan
`assocplot`	Plot Asosiasi (Association)
`barplot`	Plot Batang (Bar)
`boxplot`	Plot Kotak (Box)
`contour`	Plot Contour
`coplot`	Plot Conditioning
`dotchart`	Plot Cleveland Dot
`filled.contour`	Plot Level (Contour)
`fourfoldplot`	Plot Fourfold
`hist`	Histogam
`image`	Menampilkan suatu Warna Image
`matplot`	Plot Kolom suatu Matriks
`mosaicplot`	Plot Mosaic
`pairs`	Matriks Scatterplot
`persp`	Plot Perspektif
`plot`	Plot X-Y Umum
`stars`	Plot Star (Spider/Radar)
`stem`	Plot Stem-and-Leaf
`stripchart`	Plot Scatter 1-D
`sunflowerplot`	Plot Scatter Sunflower

Gambar 5.16 terdiri dari beberapa contoh tampilan grafik plot, yaitu boxplot, a conditioning plot, pairwise scatterplot, dan star plot, yang kesemuanya mengaplikasikan dataset Anderson iris. Perintah di R untuk menggambarkan grafik-grafik tersebut adalah :

```
> boxplot(Petal.Length ~ Species, horizontal=T,
+ col="lightblue", boxwex=.5,
+ xlab="Petal length (cm)", ylab="Species",
+ main="BoxPlot Group")
> coplot(Petal.Width ~ Petal.Length | Species,
+ col=as.numeric(Species), pch=as.numeric(Species))
> pairs(iris[,1:4], col=as.numeric(Species),
+ main="Pairwise scatterplot")
> stars(iris[,1:4], key.loc=c(2,35), mar=c(2, 2, 10, 2),
+ main="Star plot of individuals", frame=T)
```



Gambar 5.16: Contoh grafik plot

V.2.4 Grafik Trellis

Sistem grafik trellis dalam R tersedia dalam paket lattice. Model grafik ini khususnya digunakan untuk visualisasi multivariate apabila relasi antara variable berubah bersama beberapa group factor yang disebut sebagai kondisi (*conditioning*) suatu grafik terhadap factor. Metoda ini menggunakan formula yang similar dengan formula statistic untuk menspesifikasikan variable yang akan diplot serta hubungannya dalam plot.

Untuk memudahkan dalam penggunaannya, jenis plot dibagi berdasarkan banyaknya variabel: satu variable (univariate), dua variable (bivariate), tiga variable (trivariate) dan banyak variable (hypervariate).

Satu varibel/Univariate

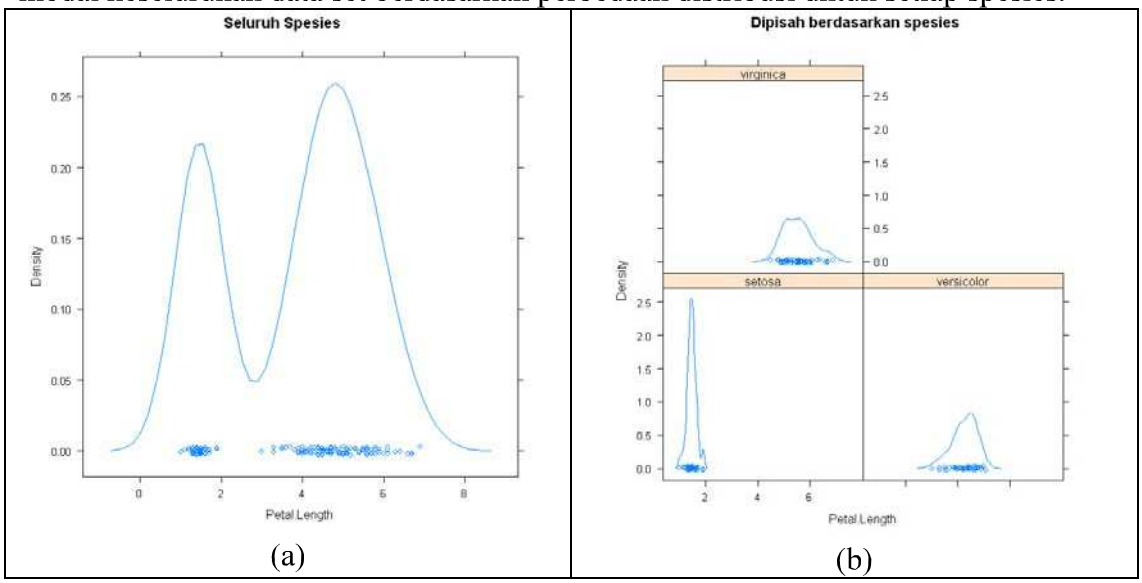
Sebagai salah contoh univariate adalah membuat grafik plot densitas pada keseluruhan data. Pada contoh ini digunakan dataset iris seperti contoh sebelumnya. Berikut adalah metode yang digunakan untuk menampilkan plot densitas:

```
> densityplot(~ Petal.Length, data=iris,main="Seluruh Spesies")
```

Operator ~ tidak memiliki operan di sebelah kiri, karena tidak ada variable terikat (dependent) dalam plot; ini menunjukkan sifat univariate. Petal.Length adalah variable bebas (independent), dan diperoleh plotnya. Gambar 5.17(a) sebelah kiri menunjukkan plot densitas univariate.
Pengkondisian dilakukan dengan menambahkan operator "|", yang dapat dibaca sebagai "pengkondisian pada" (*conditioned on*) satu (beberapa) variable pada sisi kanan operator, seperti pada berikut ini:

```
> densityplot(~ Petal.Length | Species, data=iris)
```

Perintah tersebut akan menampilkan satu panel per spesies; seperti yang ditunjukkan pada Gambar 5.17(b). Pada gambar tersebut tampak jelas bahwa distribusi multi-modal keseluruhan data set berdasarkan perbedaan distribusi untuk setiap spesies.



(a) (b)

Gambar 5.17 Plot densitas triller (a) tanpa dan (b) dengan pengkondisian faktor

Jenis plot untuk satu variable (univariate) adalah seperti pada table 5.2 berikut:

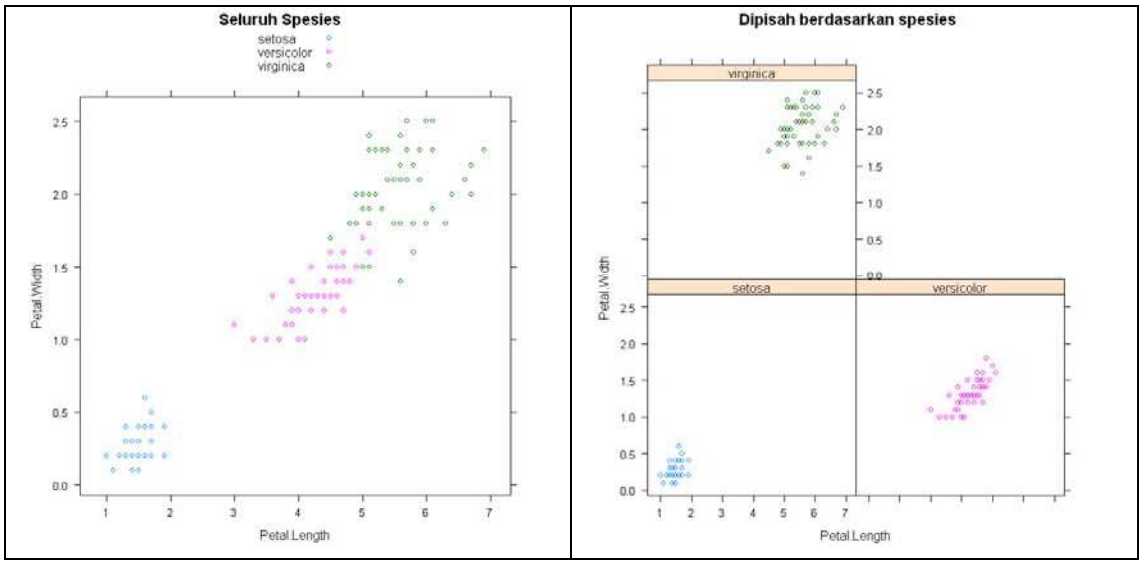
Tabel 5.2: Jenis plot untuk satu variable

Jenis	Keterangan
`assocplot`	Plot hubungan (*association*)
`barchart`	Plot batang (bar)
`bwplot`	Plot box dan whisker
`densityplot`	Plot kepadatan Kernel
`dotplot`	Plot dot
`histogram`	Histogram
`qqmath`	Quantile plot distribusi matematis
`stripplot`	Scatterplot 1 dimensi

Dua variable (Bivariate)

Salah satu metode membuat plot dua variable (bivariate) adalah `xyplot`, dimana sumbu y adalah variable terikat dan sumbu x adalah variable bebas; variable tersebut juga dapat dikondisikan terhadap satu atau lebih kelompok faktor seperti pada perintah berikut.

```
> xyplot(Petal.Width ~ Petal.Length, data=iris,
+ groups=Species, auto.key=T,main="Seluruh Spesies")
> xyplot(Petal.Width ~ Petal.Length | Species, data=iris,
+ groups=Species,main="Dipisah berdasarkan spesies")
```



Gambar 5.18: Scatter plot triller (a) tanpa dan (b) dengan pengkondisian faktor

Gambar 5.18 menunjukkan penggunaan argumen `group` untuk menspesifikasikan perbedaan cara menampilkan grafik (dalam hal ini warna) untuk setiap spesies, dan argument `auto.key` untuk mendapatkan kunci sederhana terhadap warna yang digunakan untuk setiap spesies. Motode lain yang ada di R untuk membentuk grafik dengan bivariate adalah pada table 5.3 berikut:

Tabel 5.3: Jenis plot untuk dua variable

Jenis	Keterangan
`qq`	Plot untuk membandingkan dua distribusi
`xyplot`	Plot scatter

Tiga variable(Trivariate)

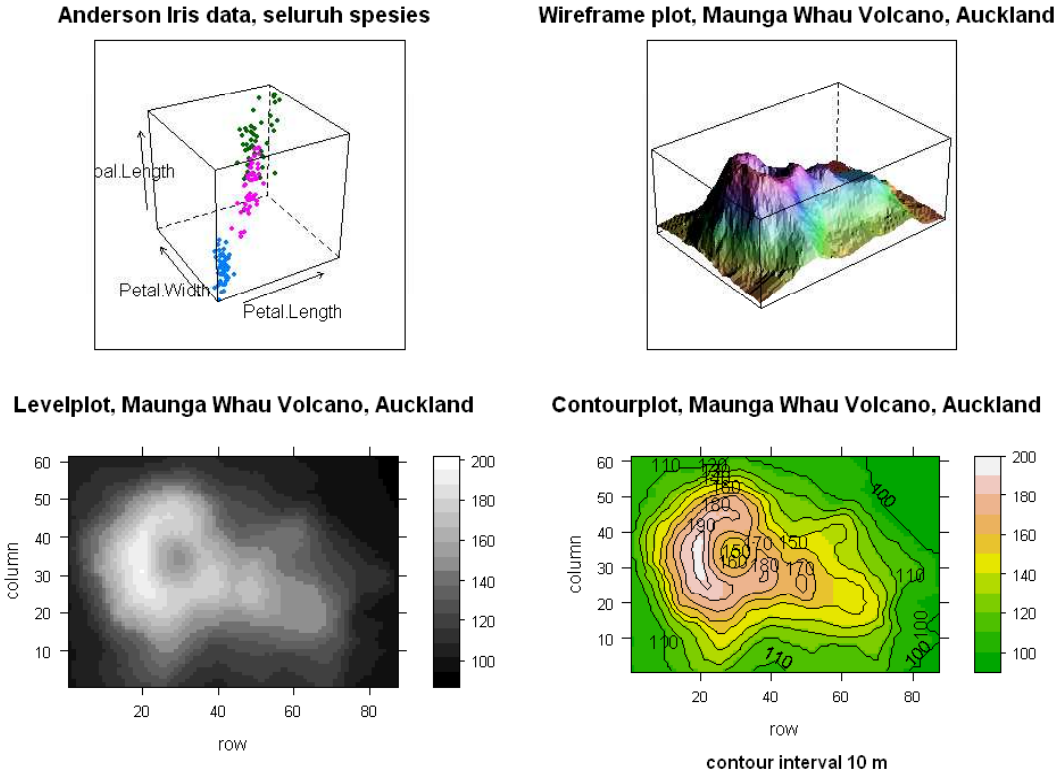
Plot yang paling banyak digunakan untuk trivariate adalah `levelplot` dan `contourplot` untuk melakukan plot 2D dari satu variable respon pada dua variable terikat kontinu (misalkan, elevation vs. dua koordinat), metode wireframe untuk suatu versi grafik 3D, dan metode cloud (awan) untuk scatter plot 3D dari tiga variable. Semua dapat dikondisikan pada suatu factor tertentu. Gambar 5.19 menunjukkan contoh yang dihasilkan dari kode berikut:

```
> pl1 <- cloud(Sepal.Length ~ Petal.Length * Petal.Width,
+ groups=Species,
+ data=iris, pch=20, main="Anderson Iris data, all species",
+ screen=list(z=30, x=-60))
> data(volcano)
> pl2 <- wireframe(volcano,
+ shade = TRUE, aspect = c(61/87, 0.4),
+ light.source = c(10, 0, 10), zoom=1.1, box=F,
+ scales=list(draw=F), xlab="", ylab="", zlab="",
+ main="Wireframe plot, Maunga Whau Volcano, Auckland")
> pl3 <- levelplot(volcano,
+ col.regions=gray(0:16/16),
+ main="Levelplot, Maunga Whau Volcano, Auckland")
> pl4 <- contourplot(volcano, at=seq(floor(min(volcano)/10)*10,
+ ceiling(max(volcano)/10)*10, by=10),
+ main="Contourplot, Maunga Whau Volcano, Auckland",
+ sub="contour interval 10 m",
+ region=T,
+ col.regions=terrain.colors(100))
> print(pl1, split=c(1,1,2,2), more=T)
> print(pl2, split=c(2,1,2,2), more=T)
> print(pl3, split=c(1,2,2,2), more=T)
> print(pl4, split=c(2,2,2,2), more=F)
> rm(pl1, pl2, pl3, pl4)
```

Gambar 5.19 menunjukkan hasil dari perintah di atas. Sebagai catatan, data set volcano merupakan matriks elavasi:

```
> str(volcano)
num [1:87, 1:61] 100 101 102 103 104 105 105 106 107 108 ...
```

Metode levelplot menkonversi ke variable respon (nilai z) dan dua predictor, yaitu baris dan kolom matriks (nilai x dan y). Contoh tersebut menunjukkan metode lattice tingkat tinggi yang melakukan pembentukan grafik sendiri. Hasil dari metode levelplot digambar dengan metode `print`. Metode plot tiga variable adalah seperti Table 5.4:



Gambar 5.19: Grafik plot Trellis trivariate

Tabel 5.4: Jenis plot untuk tiga variable

Jenis	Keterangan
`levelplot`	Plot level
`contourplot`	Plot contour
`cloud`	Plot scatter 3 dimensi
`wireframe`	Permukaan 3 dimensi (similar dengan plot persp di R)

Lebih dari tiga variabel/Hypervariate

Selain motode plot untuk satu, dua, tiga variable, R juga menyediakan plot grafik untuk lebih dari tiga variable (hypervariate) seperti pada Table 5.5 berikut:

Tabel 5.5: Jenis plot untuk lebih dari tiga variable

Jenis	Keterangan
`splom`	Matriks plot scatter
`parallel`	Plot koordinat paralel

V.3. Fungsi Distribusi

Fungsi distribusi merupakan salah satu bahasan penting dalam statistika, terutama dalam analisis data. Fungsi distribusi merupakan salah satu alat pendekatan distribusi suatu data. Fungsi distribusi juga berperan dalam menentukan densitas suatu fungsi

data. Dalam bab ini akan dibahas fungsi distribusi dan fungsi yang berkaitan dengannya.

V.3.1. Jenis fungsi distribusi dalam R

Software R mempunyai koleksi fungsi distribusi standar yang lengkap, yang tersedia dalam paket program R dan dapat ditambah dengan mendownload dalam bentuk paket dari situs R.

Fungsi distribusi di R disediakan untuk memfasilitasi fungsi distribusi kumulatif (Cummulative Distributive Function (CDF)) `P(X ≤ x)`, fungsi probabilitas densitas (Probability Density Function (PDF)), dan fungsi kuantil *(diberikan q, x lebih kecil sedemikian hingga $P(X \leq x) > q$)*. Berikut adalah tabel distribusi di R.

Tabel 5.6: Jenis fungsi distribusi dalam R

Nama Distribusi	Nama fungsi di R	Argument tambahan
Beta	`beta`	shape1, shape2, ncp
Binomial	`binom`	size, prob
Cauchy	`cauchy`	location, scale
Chi-squared	`chisq`	df, ncp
Exponential	`exp`	rate
F	`f`	df1, df1, ncp
Gamma	`gamma`	shape, scale
Geometric	`geom`	Prob
Hypergeometric	`hyper`	m, n, k
Log-normal	`lnorm`	meanlog, sdlog
Logistic	`logis`	location, scale
Binomial negative	`nbinom`	size, prob
Normal	`norm`	mean, sd
Poisson	`pois`	Lambda
t - Student's	`t`	df, ncp
Uniform	`unif`	min, max
Weibull	`weibull`	shape, scale
Wilcoxon	`wilcox`	m, n

V.3.2. Fungsi Probabilitas Densitas (Probability Density Function (PDF))

Fungsi probabilitas densitas merupakan salah satu parameter statistic yang digunakan untuk mengetahui probabilitas terhadap suatu factor yang ada dalam sample. Buku ini akan mengawali pembahasan dengan contoh kasus untuk mempermudah pemahaman tentang fungsi densitas seperti berikut ini:
Terdapat 16 mahasiswa dipilih secara acak dari populasi dimana 30% adalah wanita. Berapa probabilitas sebanyak nol, satu, dua, …, enam belas dari mahasiswa tersebut yang dipilih adalah wanita?. Untuk menghitung probabilitas tersebut akan digunakan beberapa langkah dalam R seperti dibawah ini.

```
> round(dbinom(0:16, 16, 0.3), 3)    # dbinom  artinya d:densitas dan
                                       binom:binomial

[1] 0.003 0.023 0.073 0.146 0.204 0.210 0.165 0.101 0.049 0.019 0.006 0.001
[13] 0.000 0.000 0.000 0.000 0.000
```

Pertama adalah menghitung nilai probabilitas jumlah wanita yang terpilih dari populasi yang dicari tersebut berdasarkan distribusi binomial dengan menggunakan fungsi `dbinom()`. Nilai probabilitas dari masing-masing kejadian tersebut adalah:

Jml Wanita terpilih	Probabilitas
0	0.003
1	0.023
2	0.073
3	0.146
4	0.204
5	0.210
6	0.165
7	0.101
8	0.049
9	0.019
10	0.006
11	0.001
12 - 16	0.000

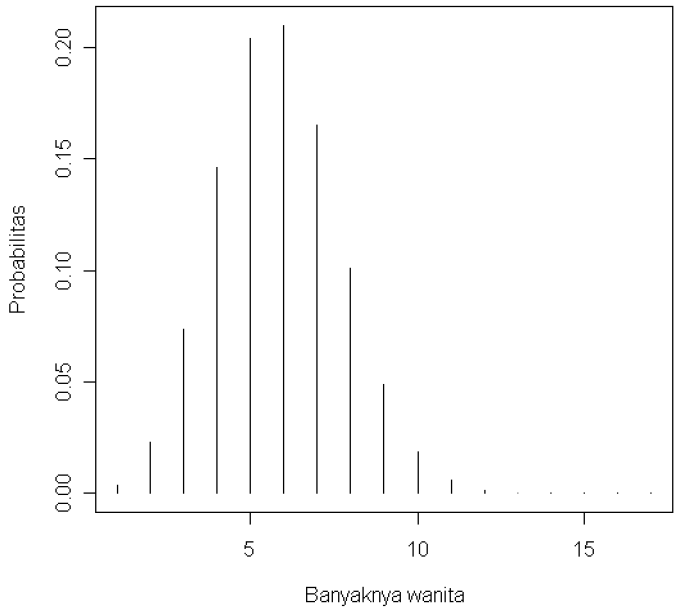
Total kumulatif dari masing-masing nilai probabilitas tersebut adalah 1 (hal ini menunjukkan bahwa nilai probabilitas tersebut merupakan pdf). Kemudian untuk menggambarkan scatter plot nilai probabilitas tersebut digunakan perintah plot seperti berikut:

```
> plot(dbinom(0:16, 16, 0.3), type =”h”, xlab=”   Banyaknya wanita”,
  + ylab=”Probabilitas”)
```

Perintah tersebut menggambarkan grafik plot (Gambar 5.20) dimana sample berdistribusi binomial dimana menghitung probabilitas sukses 0 hingga 16 (0:16) pada suatu populasi yang terdiri dari 16 mahasiswa dimana terdiri dari 0.3 (30%) mahasiswa wanita.

Misalkan terdapat kondisi bahwa hanya 2 dari 16 yang terpilih adalah wanita. Berapa probabilitas dua atau kurang dari jumlah wanita dapat terpilih kembali? Penyelesaian masalah tersebut dapat diselesaikan dengan menggunakan fungsi `pbiom()` seperti perintah program dibawah, dimana argument pertama = 2 yang menyatakan jumlah wanita terpilih, argument kedua adalah 16 yang menyatakan jumlah keseluruhan, argument ketiga menyatakan probabilitas jumlah wanita. Perintah dalam R adalah sebagai berikut:

```
> pbinom(2, 16, 0.3, lower.tail = T) # pbinom, p:  fungsi distribusi
                                       atau probabilitas, binom: binomial
[1] 0.09935968
```

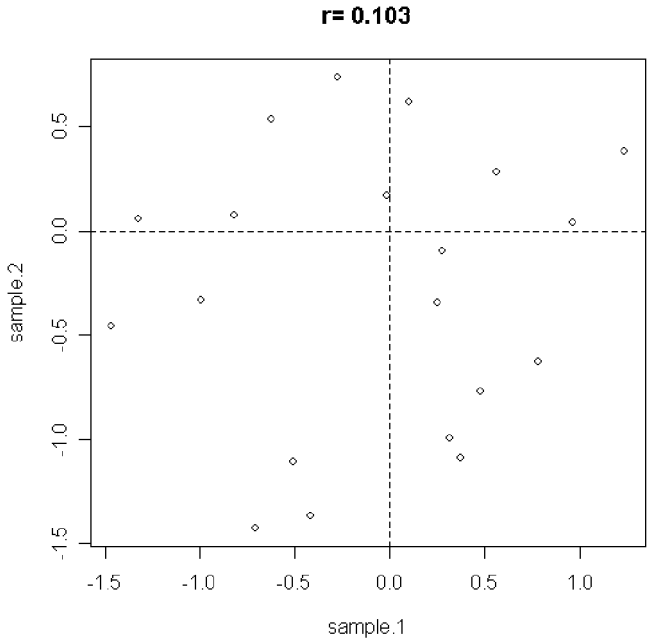


Gambar 5.20: Plot binomial data mahasiswa

(Interpretasi: dalam sample random dari 16 orang populasi dengan 30% adalah wanita, terdapat sekitar 10% bahwa dari sample akan terpilih sebanyak nol, satu atau dua adalah wanita. Maka jika kita lihat, 16 orang sample dengan dua atau kurang wanita, maka kita menduga bahwa terjadi diskriminasi terhadap hal tersebut, maka 10% kesempatan yang kita duga tersebut adalah tanpa dasar yang kuat).

Contoh berikut menyatakan bahwa dua random sample yang saling bebas yang berdistribusi normal seharusnya tidak berkorelasi. Kita dapat mensimulasikan hal ini berulangkali untuk menaksi koefisien korelasi dimana probabilitas error Type I (yaitu menolak hipotesis yang menyatakan tidak terdapat suatu korelasi) adalah 10%. Pertama, kita akan menuliskan:

```
> size <- 20
> sample.1 <- rnorm(size) # rnorm; r: deviasi random; norm:
                                    distribusi normal
> sample.2 <- rnorm(size)
> cor.test(sample.1, sample.2)

Pearson's product-moment correlation

data:  sample.1 and sample.2
t = 0.4394, df = 18, p-value = 0.6656
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.3557135  0.5217580
sample estimates:
      cor
0.1030244

> plot(sample.1, sample.2)
> abline(h=0, lty=2)
> abline(v=0, lty=2)
> title(paste("r=", round(cor(sample.1, sample.2), 3)))
```

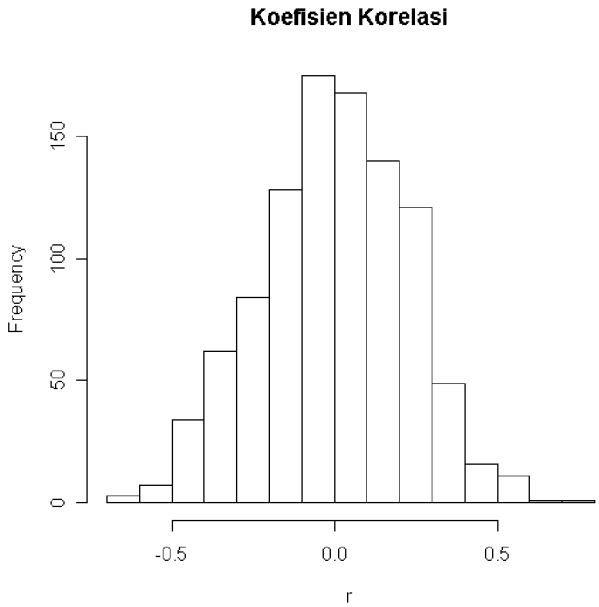


Gambar 5.21: Plot variabel *sample1* dan *sample 2*

Gambar 5.21 menunjukkan bahwa 2 variabel random sample.1 dan sample.2 adalah saling bebas.
Sebagai catatan, karena sifat kerandomannya, hasil yang akan anda peroleh dari hasil percobaan yang dilakukan tentu akan berbeda dengan apa yang disajikan dalam buku ini.

Kemudian, untuk perhitungan yang lebih besar lagi, yakni mencari nilai-nilai statistiknya, perlu mendefinisikan fungsi secara tersendiri seperti berikut ini:

```
> cor.2  <- function(size) {
+        sample.1 <- rnorm(size)
+        sample.2 <- rnorm(size)
+        cor(sample.1, sample.2)
+}
> results <- NULL
> length <- 1000
> for (i  in  1:length) {
+       results[i]  <- cor.2(20)
+}
> hist(results, xlab= "r", main= "Koefisien Korelasi")
> (paste("5% nilai r lebih negative dari",
round(sort(results)[length/20], 3)))

[1] "5% nilai r lebih negative dari -0.386"

> (paste("5% nilai r lebih positif dari",
         round(sort(results)[length - length/20], 3)))

[1] "5% nilai r lebih positif dari 0.35"
```

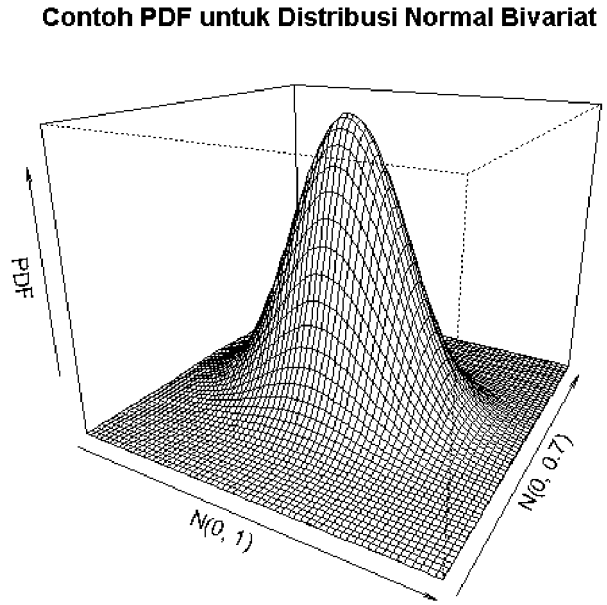


Gambar 5.22: Histogram koefisien korelasi

Sepertinya, terlihat suatu interval antara -0, 36 …. +0,37 yang melingkupi 90% dari koefisien korelasi *r* sample untuk ukuran sample 20. Sekali lagi hasilnya mungkin sedikit berbeda antara yang anda lakukan dengan yang dikerjakan dalam buku ini.

Suatu plot perspektif untuk PDF - distribusi normal bivariat:

```
> range <- seq (-3, 3,  by= 0.1)
> random.1 <- dnorm(range, 0, 1)
> random.2 <- dnorm(range, 0, 0.7)
> mesh <- outer(random.1, random.2, function(x, y) x * y)
> persp(range, range, mesh, phi = 20, theta = 30, expand = 0.8,
+ xlab = "N(0, 1)", ylab = "N(0, 0.7)",zlab = "PDF",
+ main = "Contoh PDF untuk Distribusi Normal Bivariat")
```



Gambar 5.23: Grafik 3D untuk pdf normal bivariat

Baris perintah di atas adalah salah satu contoh pembahasan mengenai fungsi probabilitas densitas (*probability density function* (PDF)) dengan kasus distribusi normal bivariat.

V.3.3. Fungsi Kumulatif Densitas Empirik (Empirical Cummulative Density Function (ECDF))

R menyediakan fitur dengan fungsi `ecdf()` untuk mencari fungsi densitas kumulatif empirik (empirical cummulative density function (ecdf)).. Berikut adalah teori yang berkaitan dengan **`ecdf`**:

- Fungsi distribusi kumulatif suatu populasi dituliskan dalam bentuk

$$F(x) = \Pr ob(X \leq x)$$

- Daerah di bawah fungsi densitas *f(x)* dari *a* ke *b* dituliskan dalam bentuk

$$F(b) - F(a) = \Pr ob(a < X \leq b)$$

- Fungsi distribusi kumulatif dicari dengan mengestimasi nilai *F(x)*, dimana proporsi nilai data $\leq x$
- Histogram yang ditampilakan adalah histogram kumulatif
- Dapat menjadi sempurna jika histogramnya hanya memiliki satu observasi per bin (batang)
- ECDF bersifat unik dan tidak memerlukan *binning*
- Sangat baik untuk menunjukkan perbedaan dalam distribusi keseluruhan di antara dua atau tiga grup yang berlapis/ menumpuk.
- Dapat membaca langsung nilai quantil nya.

V.4. Regresi & ANOVA (Analysis of Variance)

Dalam sub bab ini akan dijelaskan tentang regresi dan ANOVA, yang merupakan salah satu metode dasar statitistik dalam melakukan pengolahan dan analisis data.
Disini akan disajikan teori dan praktek regresi dan anova serta penggunaan R untuk analisis tersebut. Pada tahap awal mungkin akan terasa sedikit rumit, namun diharapkan setelah mencoba beberapa contoh akan menjadi lebih mudah.

Analisis regresi digunakan untuk menjelaskan atau memodelkan hubungan antara suatu variable tunggal `Y`, disebut sebagai variable respon, output atau terikat, dan satu atau lebih variable predictor, input, bebas atau penjelasan (explanatory), $X_1, \ldots, X_p$. Apabila `p = 1` maka disebut regresi sederhana, sedangkan apabila `p > 1` maka disebut regresi berganda atau regresi multivariate. Jika terdapat lebih dari satu variable terikat Y, maka disebut regresi mulrivariate berganda.

Variabel respon harus berbentuk kontinu, sedangkan variabel penjelasan dapat berbentuk kontinu, diskrit ataupun kategori. Sebelum kita masuk ke contoh, maka akan digunakan data pima dari paket library ”faraway” yang sudah tersedia di paket