

PRAKTIKUM 4

FUNGSI DALAM R

Tujuan Umum Praktikum

Mahasiswa praktikum mampu memahami dan menerapkan teknik visualisasi data dengan menggunakan RStudio untuk menganalisis dan menyajikan data secara efektif melalui grafik yang sesuai, serta mampu memilih jenis visualisasi yang tepat berdasarkan karakteristik data.

Tujuan Khusus

1. Mahasiswa dapat mengenali berbagai jenis grafik dasar yang digunakan untuk visualisasi data, seperti scatter plot, histogram, dan boxplot.
2. Mahasiswa mampu menggunakan fungsi dasar visualisasi di R seperti `plot()`, `hist()`, dan `boxplot()` untuk membuat grafik dari data.
3. Mahasiswa dapat memahami cara menambahkan elemen visualisasi tambahan seperti judul, label sumbu, dan warna pada plot di R.
4. Mahasiswa mampu menginterpretasi grafik yang dihasilkan dari data untuk menggambarkan pola atau tren dalam data.
5. Mahasiswa dapat membuat visualisasi yang lebih kompleks dengan menggunakan paket tambahan seperti `ggplot2` untuk membuat grafik yang lebih interaktif dan estetik.
6. Mahasiswa dapat membandingkan dan memilih visualisasi yang tepat berdasarkan jenis data yang digunakan (numerik atau kategorikal).
7. Mahasiswa mampu melakukan modifikasi dan kustomisasi visualisasi data sesuai dengan kebutuhan analisis, seperti mengubah skala sumbu, mengganti warna, dan menambah anotasi.

Visualisasi Data Menggunakan Fungsi `plot`

Fungsi `plot()` merupakan fungsi umum yang digunakan untuk membuat plot pada R. Format dasarnya adalah sebagai berikut:

```
plot(x, y, type="p")
```

Catatan:

- x dan y: titik koordinat plot Berupa variabel dengan panjang atau jumlah observasi yang sama.
- type: jenis grafik yang hendak dibuat. Nilai yang dapat dimasukkan antara lain:
 - type="p" : membuat plot titik atau scatterplot. Nilai ini merupakan default pada fungsi plot().
 - type="l" : membuat plot garis.
 - type="b" : membuat plot titik yang terhubung dengan garis.
 - type="o" : membuat plot titik yang ditimpa oleh garis.
 - type="h" : membuat plot garis vertikal dari titik ke garis $y=0$.
 - type="s" : membuat fungsi tangga.
 - type="n" : tidak membuat grafik plot sama sekali, kecuali plot dari axis. Dapat digunakan untuk mengatur tampilan suatu plot utama yang diikuti oleh sekelompok plot tambahan.

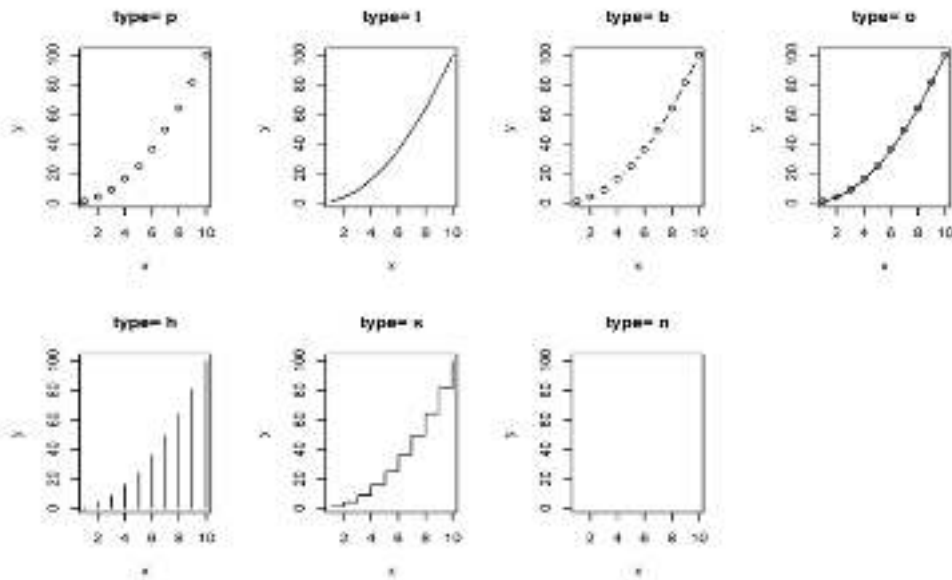
Contoh untuk masing-masing grafik tersebut. Berikut adalah contoh sintaks dan hasil plot yang disajikan pada Gambar berikut ini:

```
# membuat vektor data
x <- c(1:10); y <- x^2

# membuat jendela grafik dengan 2 baris dan 4 kolom
par(mfrow=c(2,4))

# loop
type <- c("p","l","b","n","h","o","s","n")

for (i in type){
  plot(x,y, type=i,
       main=paste("type=", i))
}
```



contoh kali ini merupakan data suatu fungsi matematika. Berikut adalah sintaks yang digunakan:

```
set.seed(123)
x <- seq(from=0, to=10, by=0.1)
y <- x^2*exp(-x/2)*(1+rnorm(n=length(x), mean=0, sd=0.05))

par(mfrow=c(1,2),

    # mengatur margin grafik

    mar=c(4,4,1.5,1.5),

    # mengatur margin sumbu

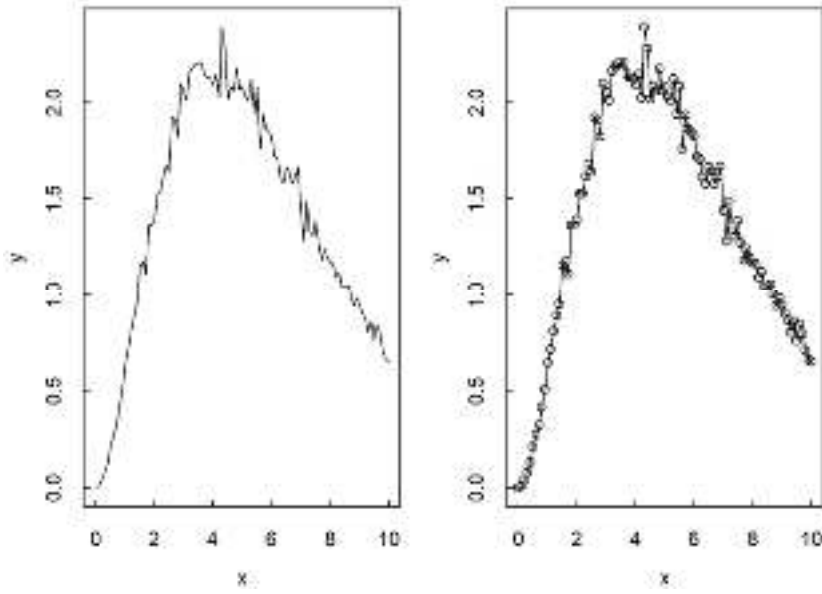
    mex=0.8,

    # arah tick sumbu koordinat

    tcl=0.3)

plot(x, y, type="l")

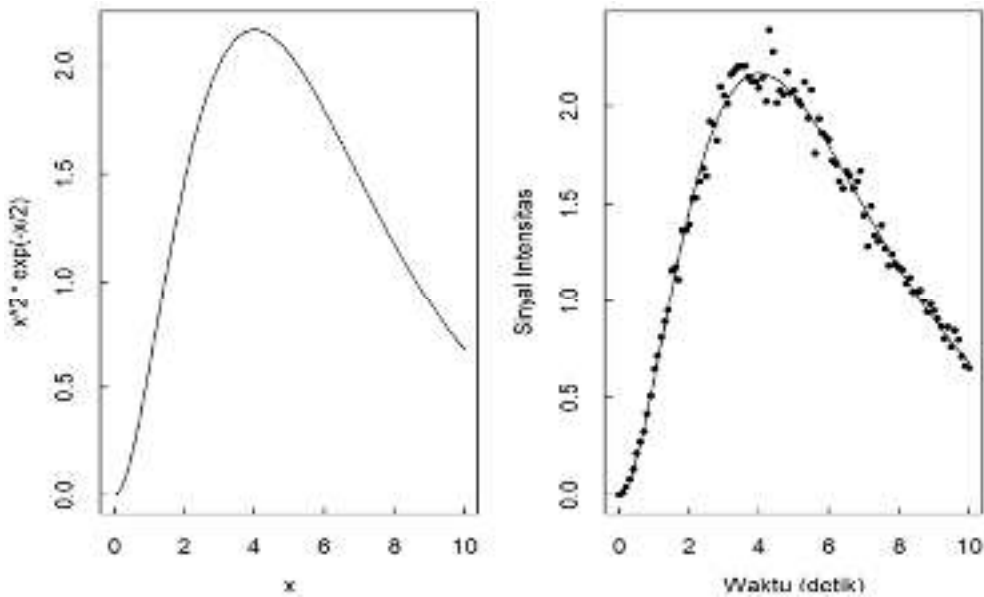
plot(x, y, type="o")
```



Fungsi lain yang dapat digunakan untuk membuat kurva suatu persamaan matematis adalah fungsi `curve()`. Berbeda dengan fungsi `plot()` yang perlu menspesifikasi objek pada sumbu x dan y, fungsi `curve()` hanya perlu menspesifikasi objek sumbu x saja. Format fungsi `curve()` adalah sebagai berikut:

```
curve(expr, from = NULL, to = NULL, add = FALSE)
```

```
par(mfrow=c(1,2),  
  
# mengatur margin grafik  
  
mar=c(4,4,1.5,1.5),  
  
# mengatur margin sumbu  
mex=0.8,  
  
# arah tick sumbu koordinat  
tcl=0.3)  
  
# Grafik kiri  
  
curve(expr=x^2*exp(-x/2),  
  
from=0, to=10)  
  
# Grafik kanan  
  
plot(x, y, pch=19, cex=0.7,  
xlab="Waktu (detik)",  
  
yleb="Sinyal Intensitas")  
  
curve(expr=x^2*exp(-x/2),  
  
from=0, to=10, add=TRUE)
```



Gambar 3.3: Visualisasi menggunakan fungsi curve (sebelah kiri) dan visualisasi menggunakan fungsi plot dan curve (sebelah kanan)

Visualisasi Lainnya

Visualisasi lainnya yang sering digunakan antara lain: histogram, density plot, bar plot, dan box plot.

Bar Plot

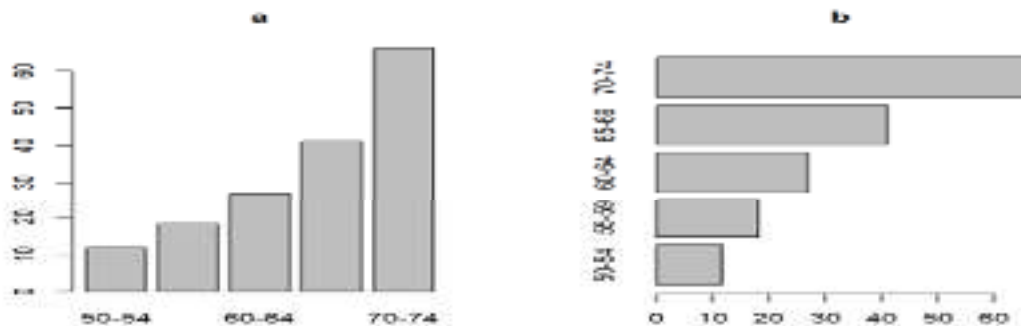
Barplot pada R dapat dibuat menggunakan fungsi `barplot()`. Untuk lebih memahaminya berikut disajikan contoh barplot menggunakan dataset `VADeaths`. Untuk memuatnya jalankan sintaks berikut:

```
VADeaths
```

```
##      Rural Male Rural Female Urban Male Urban Female
## 50-54      11.7      8.7      15.4      8.4
## 55-59      18.1     11.7     24.3     13.6
## 60-64     26.9     26.3     37.0     19.3
## 65-69     41.8     38.0     54.6     35.1
## 70-74     66.8     54.3     71.1     58.8
```

Contoh bar plot untuk variabel `Rural Male` disajikan pada Gambar

```
par(mfrow=c(1,2))
barplot(VADeaths[, "Rural Male"], main="a")
barplot(VADeaths[, "Rural Male"], main="b", horiz=TRUE)
```

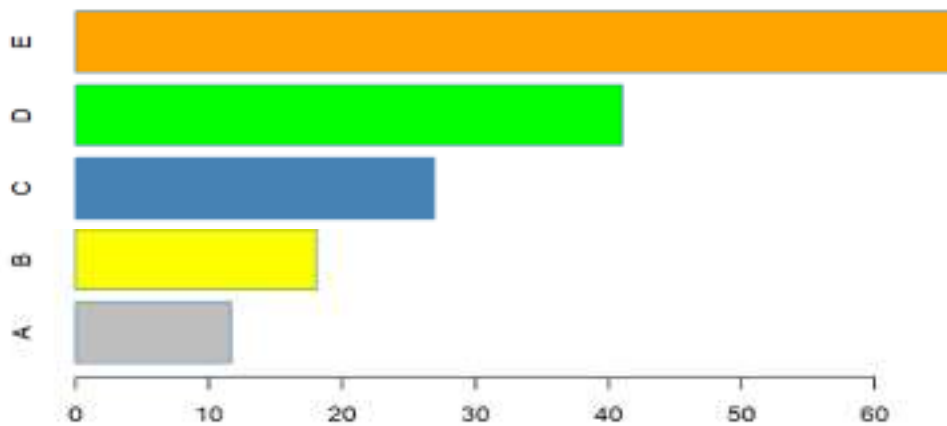


Gambar 3.4: a. bar plot vertikal; b. bar plot horizontal

```
par(mfrow=c(1,1))
```

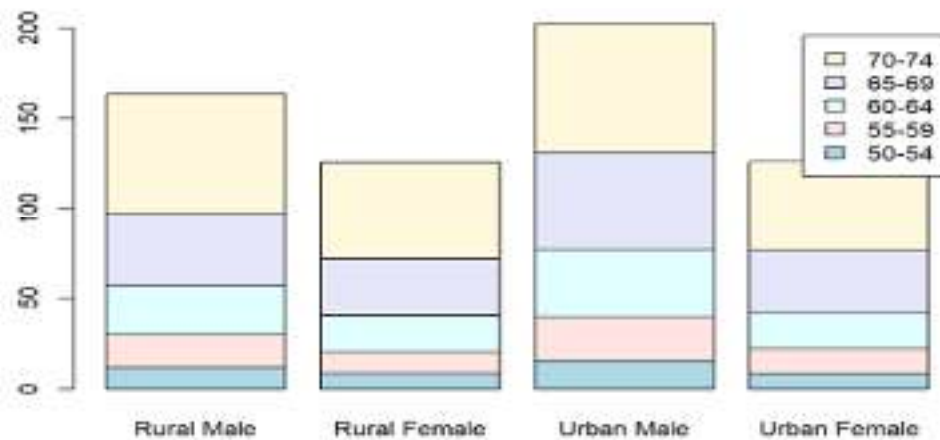
Kita dapat mengubah warna pada masing-masing bar, baik outline bar maupun box pada bar. Selain itu kita juga dapat mengubah nama grup yang telah dihasilkan sebelumnya. Berikut sintaks untuk melakukannya dan output yang dihasilkan pada Gambar berikut.

```
barplot(VADeaths[, "Rural Male"],  
        # ubah warna outline menjadi steelblue  
        border="steelblue",  
        # ubah warna box  
        col= c("grey", "yellow", "steelblue", "green", "orange"),  
        # ubah nama grup dari A sampai E  
        names.arg = LETTERS[1:5],  
        # ubah orientasi menjadi horizontal  
        horiz=TRUE)
```

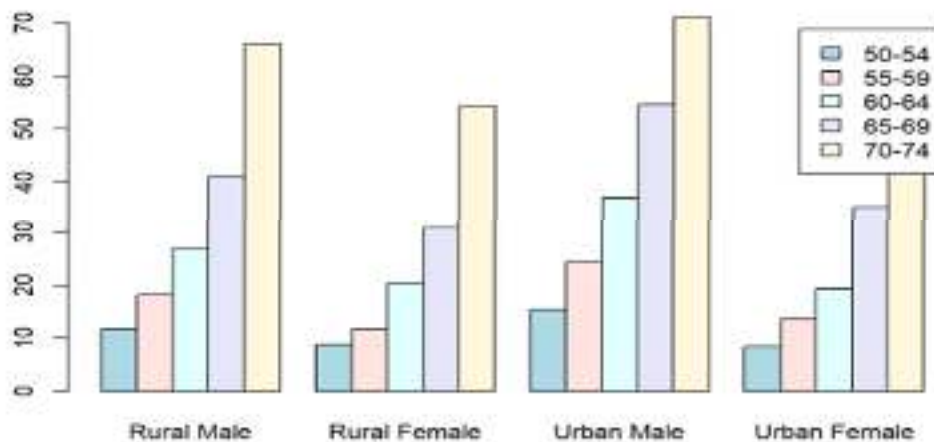


Untuk bar plot dengan *multiple group*, tersedia dua pengaturan posisi yaitu *stacked bar plot* (menunjukkan proporsi penyusun pada masing-masing grup) dan *grouped bar plot* (melihat perbedaan individual pada masing-masing grup). Pada Gambar berikut.

```
# stacked  
barplot(VADeaths,  
        col = c("lightblue", "mistyrose", "lightcyan",  
                "lavender", "cornsilk"),  
        legend = rownames(VADeaths))
```



```
# grouped
barplot(VADeaths,
        col = c("lightblue", "mistyrose", "lightcyan",
                "lavender", "cornsilk"),
        legend = rownames(VADeaths), beside = TRUE)
```



Histogram dan Density Plot

Fungsi `hist()` dapat digunakan untuk membuat histogram pada R. Secara sederhana fungsi tersebut didefinisikan sebagai berikut:

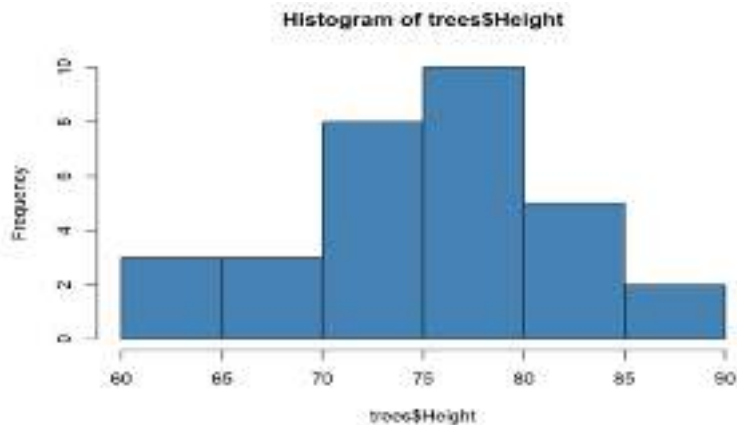
```
hist(x, breaks="Sturges")
```

Catatan:

- `x`: vektor numerik
- `breaks` *breakpoints* antar sel histogram

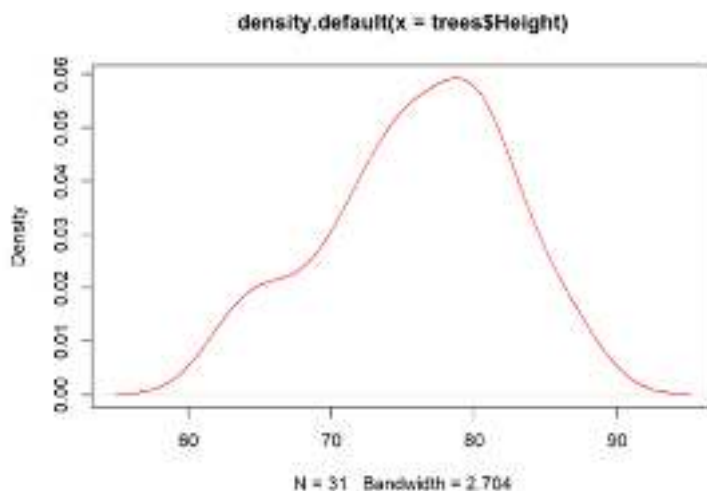
Pada dataset trees akan dibuat histogram variabel Height. Untuk melakukannya jalankan sintaks berikut:

```
hist(trees$Height)
```



Density plot pada R dapat dibuat menggunakan fungsi density(). Berbeda dengan fungsi hist(), fungsi ini tidak langsung menghasilkan grafik densitas. Fungsi density() hanya menghitung kernel densitas pada data. Densitas yang telah dihitung selanjutnya diplotkan menggunakan fungsi plot(). Berikut adalah sintaks dan output yang dihasilkan pada Gambar berikut.

```
# menghitung kernel density  
dens <- density(trees$Height)  
  
# plot densitas dengan outline merah  
plot(dens,col="red")
```

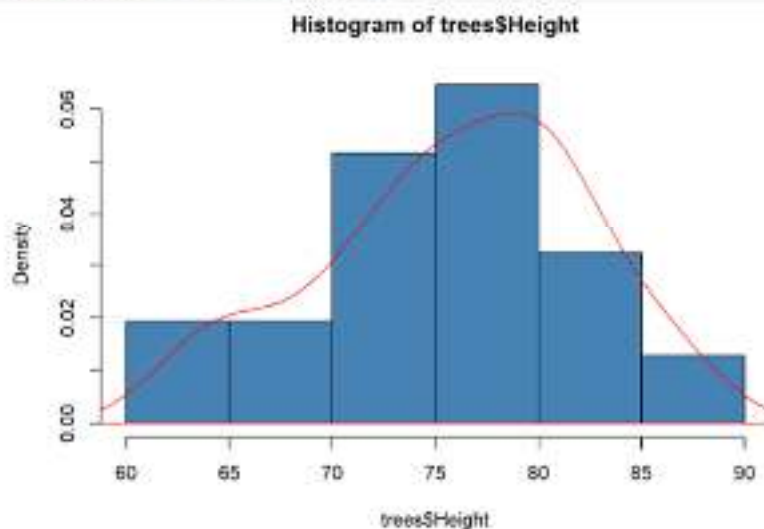


Kita juga dapat menambahkan grafik densitas pada histogram sehingga mempermudah pembacaan pada histogram. Untuk melakukannya kita perlu mengubah kernel histogram dari frekuensi menjadi density dengan menambahkan argumen `freq=FALSE` pada fungsi `hist()`. Selanjutnya tambahkan fungsi `polygon()` untuk memplotkan grafik densitas. Berikut adalah sintak dan output yang dihasilkan pada Gambar berikut.

```
# menghitung kernel density
dens <- density(trees$Height)

# histogram
hist(trees$Height, freq=FALSE, col="steelblue")

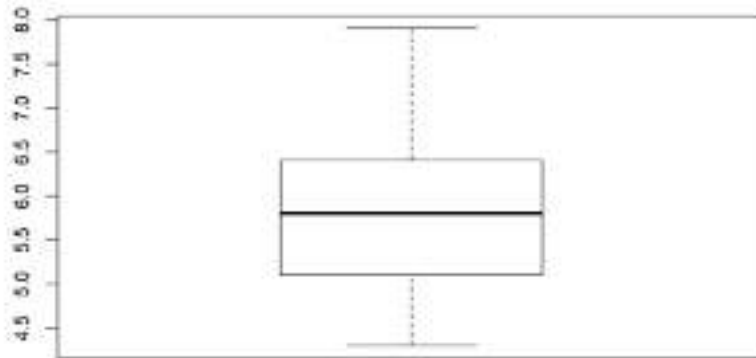
# tambahkan density plot
polygon(dens, border="red")
```



Box plot

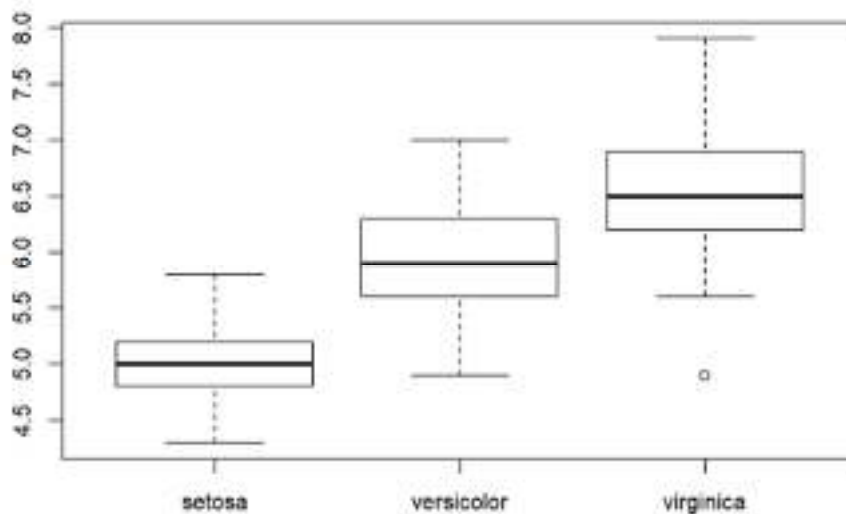
Box plot pada R dapat dibuat menggunakan fungsi `boxplot()`. Berikut adalah sintaks untuk membuat boxplot variabel `Sepal.Length` pada dataset `iris` dan output yang dihasilkan pada Gambar berikut.

```
boxplot(iris$Sepal.Length)
```



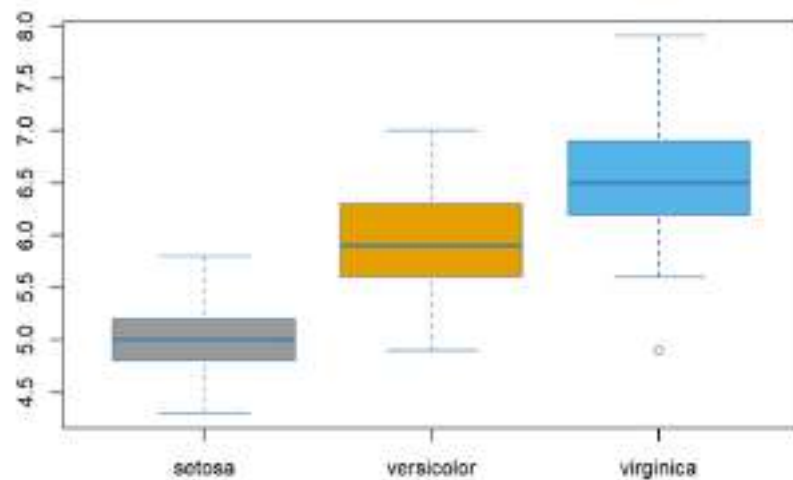
Boxplot juga dapat dibuat berdasarkan variabel factor. Hal ini berguna untuk melihat perbedaan distribusi data pada masing-masing grup. Pada sintaks berikut dibuat boxplot berdasarkan variabel Species. Output yang dihasilkan disajikan pada Gambar berikut.

```
boxplot(iris$Sepal.Length~iris$Species)
```



Kita juga dapat mengubah warna outline dan box pada boxplot. Berikut adalah contoh sintaks yang digunakan untuk melakukannya dan output yang dihasilkan disajikan pada Gambar berikut.

```
boxplot(iris$Sepal.Length~iris$Species,
        # ubah warna outline menjadi steelblue
        border = "steelblue",
        # ubah warna box berdasarkan grup
        col= c("#999999", "#E69F00", "#56B4E9"))
```



Kita juga dapat membuat boxplot pada *multiple group*. Data yang digunakan untuk contoh tersebut adalah dataset ToothGrowth. Berikut adalah sintaks untuk memuat dataset tersebut:

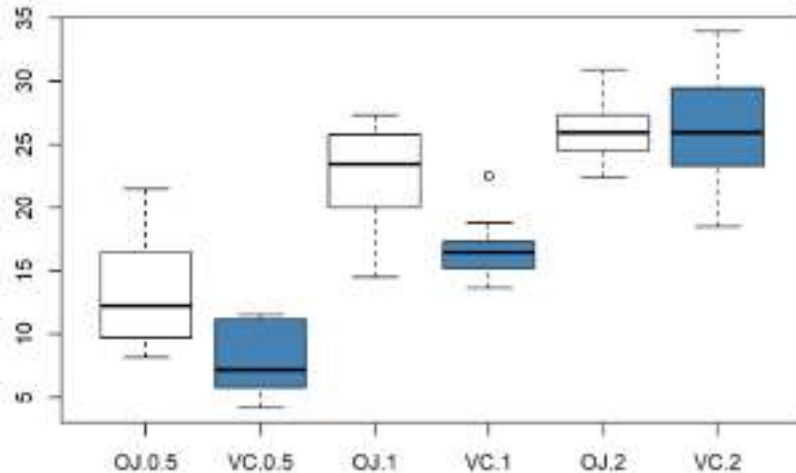
```
# ubah variable dose menjadi factor
ToothGrowth$dose <- as.factor(ToothGrowth$dose)

# print
head(ToothGrowth)
```

```
##   len supp dose
## 1  4.2   VC  0.5
## 2 11.5   VC  0.5
## 3  7.3   VC  0.5
## 4  5.8   VC  0.5
## 5  6.4   VC  0.5
## 6 18.8   VC  0.5
```

Contoh sintaks dan output boxplot *multiple group* disajikan pada Gambar berikut.

```
boxplot(len ~ supp*dose, data = ToothGrowth,
        col = c("white", "steelblue"))
```



Kustomisasi Parameter Grafik

Pada bagian ini penulis akan menjelaskan cara untuk kustomisasi parameter grafik seperti:

- menambahkan judul, legend, teks, axis, dan garis.
- mengubah skala axis, simbol plot, jenis garis, dan warna.

Menambahkan Judul

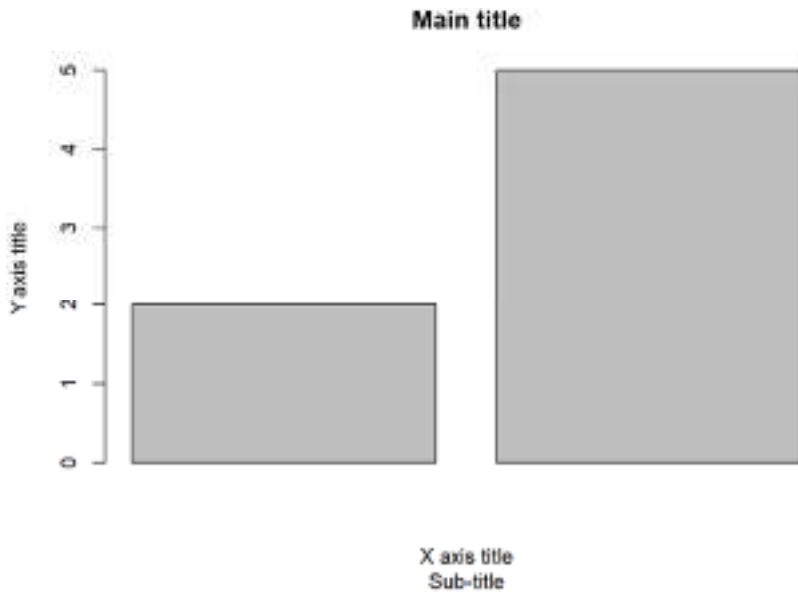
Pada grafik di R, kita dapat menambahkan judul dengan dua cara, yaitu: pada plot melalui parameter dan melalui fungsi `plot()`. Kedua cara tersebut tidak berbeda satu sama lain pada parameter input.

Untuk menambahkan judul pada plot secara langsung, kita dapat menggunakan argumen tambahan sebagai berikut:

- main**: teks untuk judul.
- xlab**: teks untuk keterangan axis X.
- ylab**: teks untuk keterangan axis y.
- sub**: teks untuk sub-judul.

Berikut contoh sintaks penerapan masing-masing argumen tersebut beserta dengan output yang dihasilkan pada Gambar berikut.

```
# menambahkan judul
barplot(c(2,5), main="Main title",
        xlab="X axis title",
        ylab="Y axis title",
        sub="Sub-title")
```



kita juga dapat melakukan kustomisasi pada warna, *font style*, dan ukuran font judul. Untuk melakukan kustomisasi pada warna pada judul, kita dapat menambahkan argumen sebagai berikut:

- a. **col.main**: warna untuk judul.
- b. **col.lab**: warna untuk keterangan axis.
- c. **col.sub**: warna untuk sub-judul

Untuk kustomisasi font judul, kita dapat menambahkan argumen berikut:

- a. **font.main**: *font style* untuk judul.
- b. **font.lab**: *font style* untuk keterangan axis.
- c. **font.sub**: *font style* untuk sub-judul.

Penting!

Nilai yang dapat diinputkan antara lain:

- 1: untuk teks normal
- 2: untuk teks cetak tebal
- 3: untuk teks cetak miring
- 4: untuk teks cetak cetak dan miring
- 5: untuk font simbol

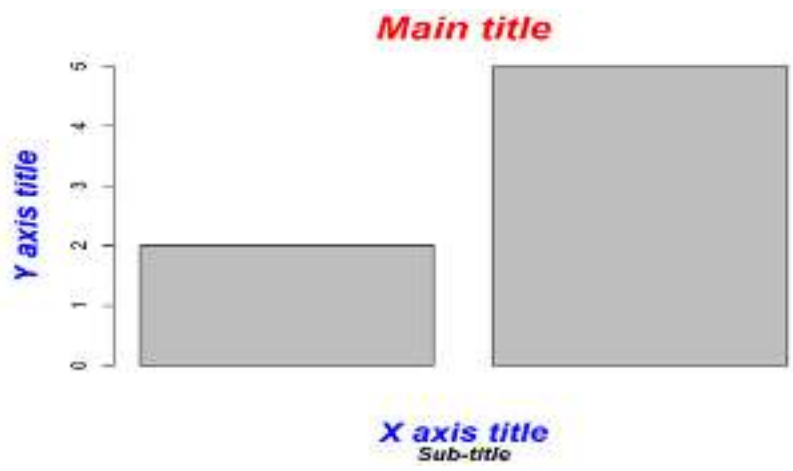
Sedangkan untuk ukuran font, kita dapat menambahkan variabel berikut:

- a. **cex.main**: ukuran teks judul.

- b. **cex.lab**: ukuran teks keterangan axis.
- c. **cex.sub**: ukuran teks sub-judul.

Berikut sintaks penerapan seluruh argumen tersebut beserta output yang dihasilkan pada Gambar

```
# menambahkan judul
barplot(c(2,5),
        # menambahkan judul
        main="Main title",
        xlab="X axis title",
        ylab="Y axis title",
        sub="Sub-title",
        # kustomisasi warna font
        col.main="red",
        col.lab="blue",
        col.sub="black",
        # kustomisasi font style
        font.main=4,
        font.lab=4,
        font.sub=4,
        # kustomisasi ukuran font
        cex.main=2,
        cex.lab=1.7,
        cex.sub=1.2)
```



Menambahkan Legend

Fungsi `legend()` pada R dapat digunakan untuk menambahkan legend pada grafik. Format sederhananya adalah sebagai berikut.

```
legend(x, y=NULL, legend, fill, col, bg)
```

Catatan:

- `x` dan `y`: koordinat yang digunakan untuk posisi legend.
- `legend`: teks pada legend
- `fill`: warna yang digunakan untuk mengisi box disamping teks legend.
- `col`: warna garis dan titik disamping teks legend.
- `bg`: warna latar belakang legend box.

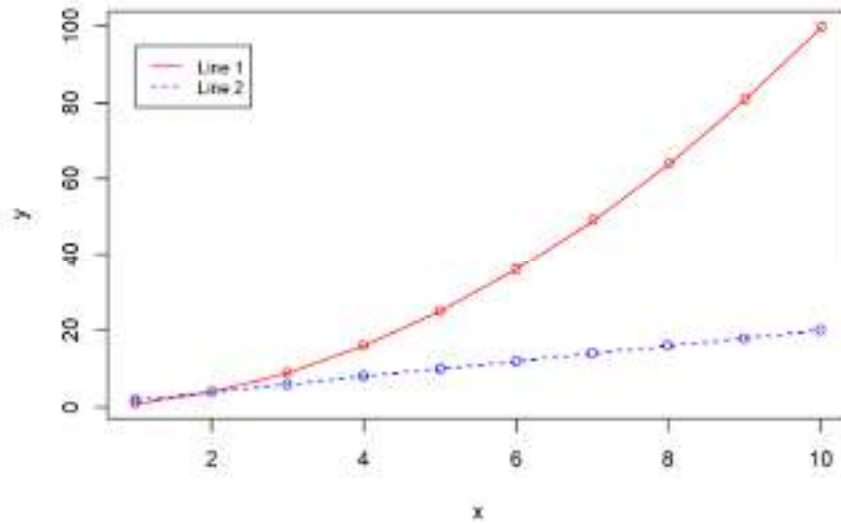
Berikut adalah contoh sintaks dan output penerapan argumen disajikan pada Gambar berikut.

```
# membuat vektor numerik
x <- c(1:10)
y <- x^2
z <- x*2

# membuat line plot
plot(x,y, type="o", col="red", lty=1)

# menambahkan line plot
lines(x,z, type="o", col="blue", lty=2)

# menambahkan legend
legend(1, 95, legend=c("Line 1", "Line 2"),
      col=c("red", "blue"), lty=1:2, cex=0.8)
```



Kita dapat menambahkan judul, merubah font, dan merubah warna backgroud pada legend.

Argumen yang ditambahkan pada legend adalah sebagai berikut:

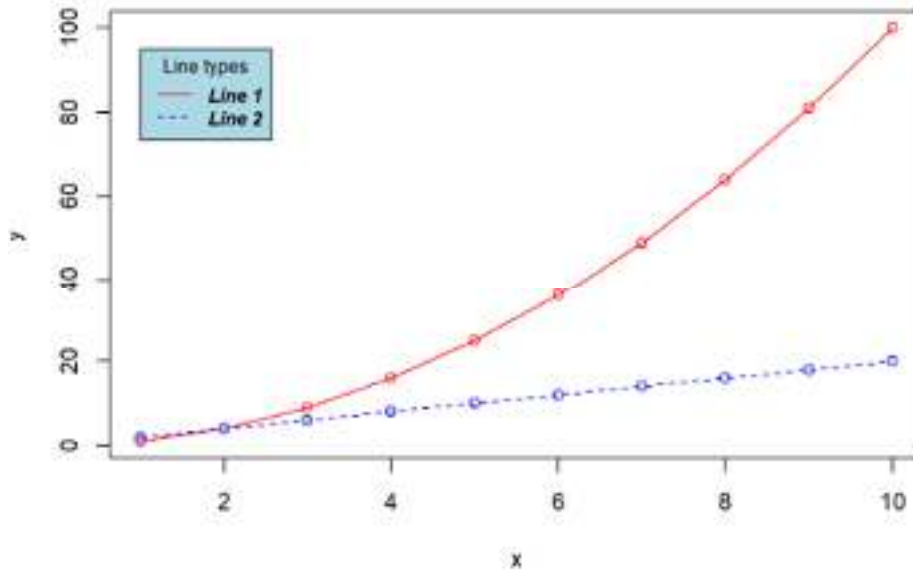
- a. **title:** Judul legend
- b. **text.font:** integer yang menunjukkan *font style* pada teks legend. Nilai yang dapat dimasukkan adalah sebagai berikut:
 - o **1:** normal
 - o **2:** cetak tebal
 - o **3:** cetak miring
 - o **4:** cetak tebal dan miring.
- c. **bg:** warna background legend box.

Berikut adalah penerapan sintaks dan output yang dihasilkan pada Gambar berikut.

```
# membuat line plot
plot(x,y, type="o", col="red", lty=1)

# menambahkan line plot
lines(x,z, type="o", col="blue", lty=2)

# menambahkan legend
legend(1, 95, legend=c("Line 1", "Line 2"),
      col=c("red", "blue"), lty=1:2, cex=0.8,
      title="Line types", text.font=4, bg='lightblue')
```



Kita dapat melakukan kustomisasi pada border dari legend melalui argumen `box.lty=(jenis garis)`, `box.lwd=(ukuran garis)`, dan `box.col=(warna box)`. Berikut adalah penerapan argumen tersebut beserta output yang dihasilkan pada Gambar berikut.

```
# membuat line plot
plot(x,y, type="o", col="red", lty=1)

# menambahkan line plot
lines(x,z, type="o", col="blue", lty=2)

# menambahkan legend
legend(1, 95, legend=c("Line 1", "Line 2"),
      col=c("red", "blue"), lty=1:2, cex=0.8,
      title="Line types", text.font=4, bg='white',
      box.lty=2, box.lwd=2, box.col="steelblue")
```

