101 Default R Graphs

Cevi Herdian

2019-05-15

Contents

Chapter 1

Preface

Note:

-An R Notebook is an R Markdown document with chunks that can be executed independently and interactively, with output visible immediately beneath the input.

-Notebook output are available as HTML, PDF, Word, or Latex.

-This Notebook as HTML is preferably open with Google Chrome.

-R-Code can be extracted as Rmd file under the button "Code" in the notebook.

Why R?:

- Free and open source
- Built for statistical computing
- Visualization tools
- Tools for building Models

Learning Objectives:

- How to create beautiful, useful, and insightful graphics and charts
- How to customize the look of them

  All languages have their inconsistencies include R Programming. This documentation helps us to create visualizing in default R graphics before to far away using the packages.

Change log update:

- 18.12.2018
- 19.12.2018
- 23.12.2018
- 24.12.2018
- 13.05.2019
- 14.05.2019
- 15.05.2019

Preferences:

- R Programming/Graphics
- Tutorials for learning R
- RDocumentation
- Statistical tools for high-throughput data analysis

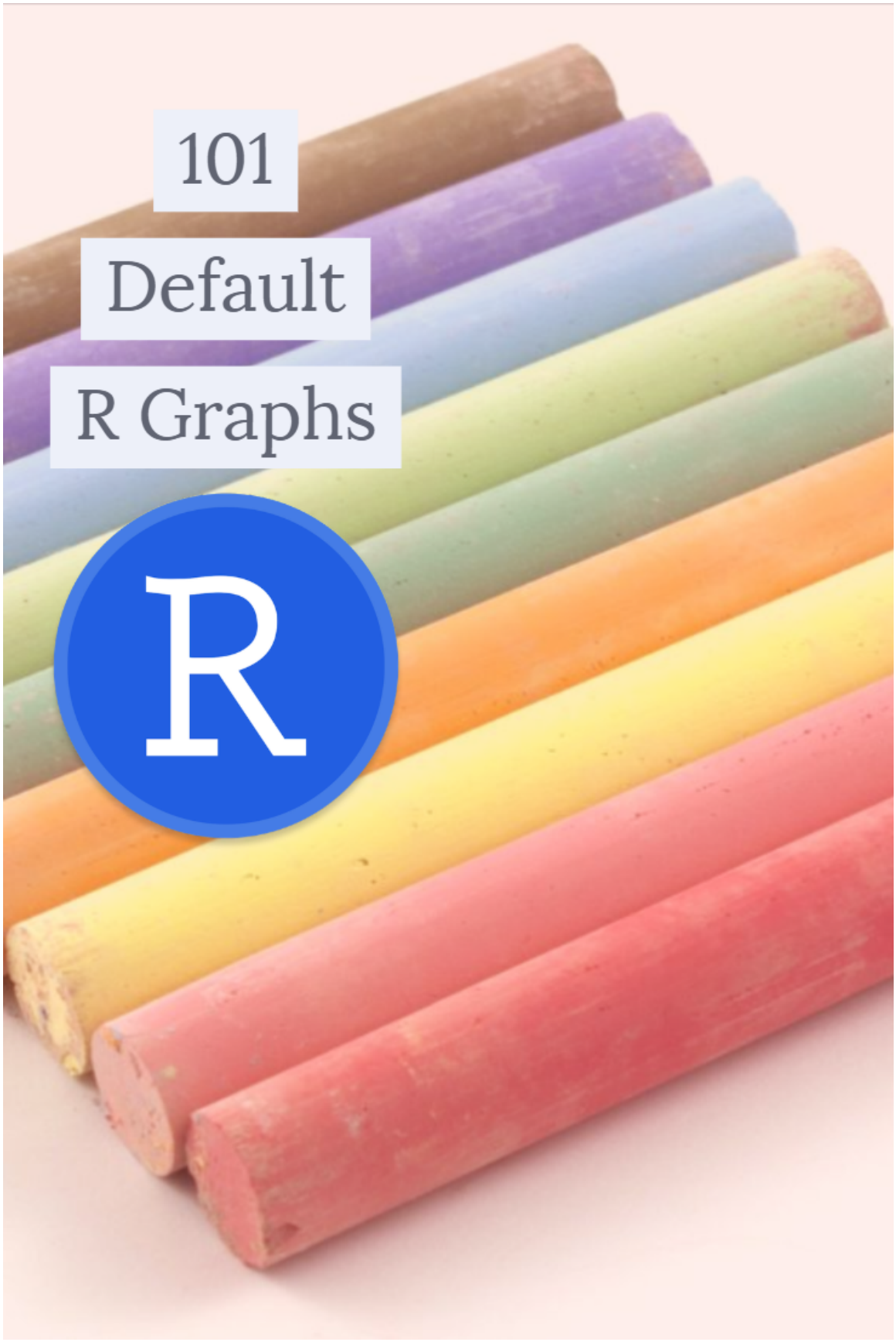


Figure 1.1:

- statmethods
- cognitiveclass.ai-Data Visualization with R
- Comprehensive Guide to Data Visualization in R
- datascienceplus
- Hands-On Programming with R
- R for Data Science
- R Markdown: The Definitive Guide
- R bookdown: Authoring Books and Technical Documents with R Markdown
- blogdown: Creating Websites with R Markdown

License:

Chapter 2

Creating and Saving Graphs in R

Creating graphs:

Used `mtcars` (Motor Trend Car Road Tests from default dataset) dataset for this section:

```
?mtcars
```

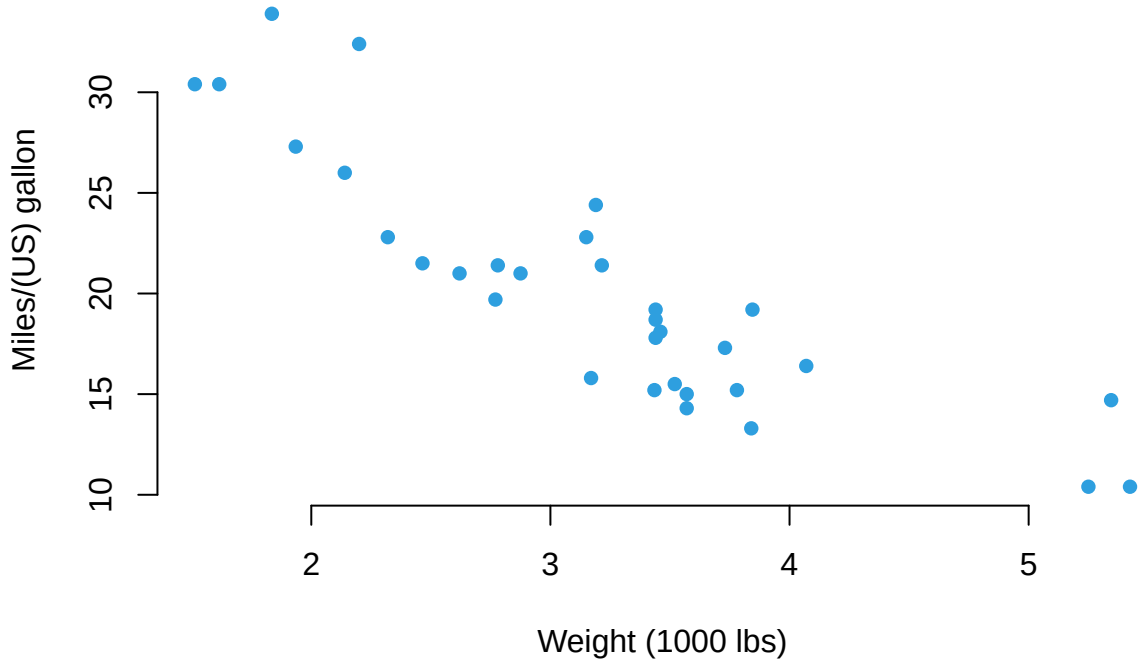
```
## starting httpd help server ... done
```

```
head(mtcars,5)
```

```
##                   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
```

Created a graphs with `plot()` function:

```
plot(x = mtcars$wt, y = mtcars$mpg,
     pch = 16, frame = FALSE,
     xlab = "Weight (1000 lbs)", ylab = "Miles/(US) gallon", col = "#2E9FDF")
```



Saving graphs:

There are two ways of saving graphs in R

1. In RStudio IDE:

Plots panel -> Export -> Save as Image or Save as PDF

```
#preference:http://www.sthda.com/english/wiki/r-base-graphs
```

2. Using R codes:

- 1. Choose the format
- 2. Create the graphs
- 3. Enter the dev.off() command

Example->

Note: The file you save are in the current working directory.

The command to get directory is `getwd()`

```
# 1. Choose the format
png("rplot.jpg") #width = 25, height = "25"

# 2. Create the graphs
plot(x = mtcars$wt, y = mtcars$mpg,
     pch = 16, frame = FALSE,
     xlab = "wt", ylab = "mpg", col = "#2E9FDF")
```

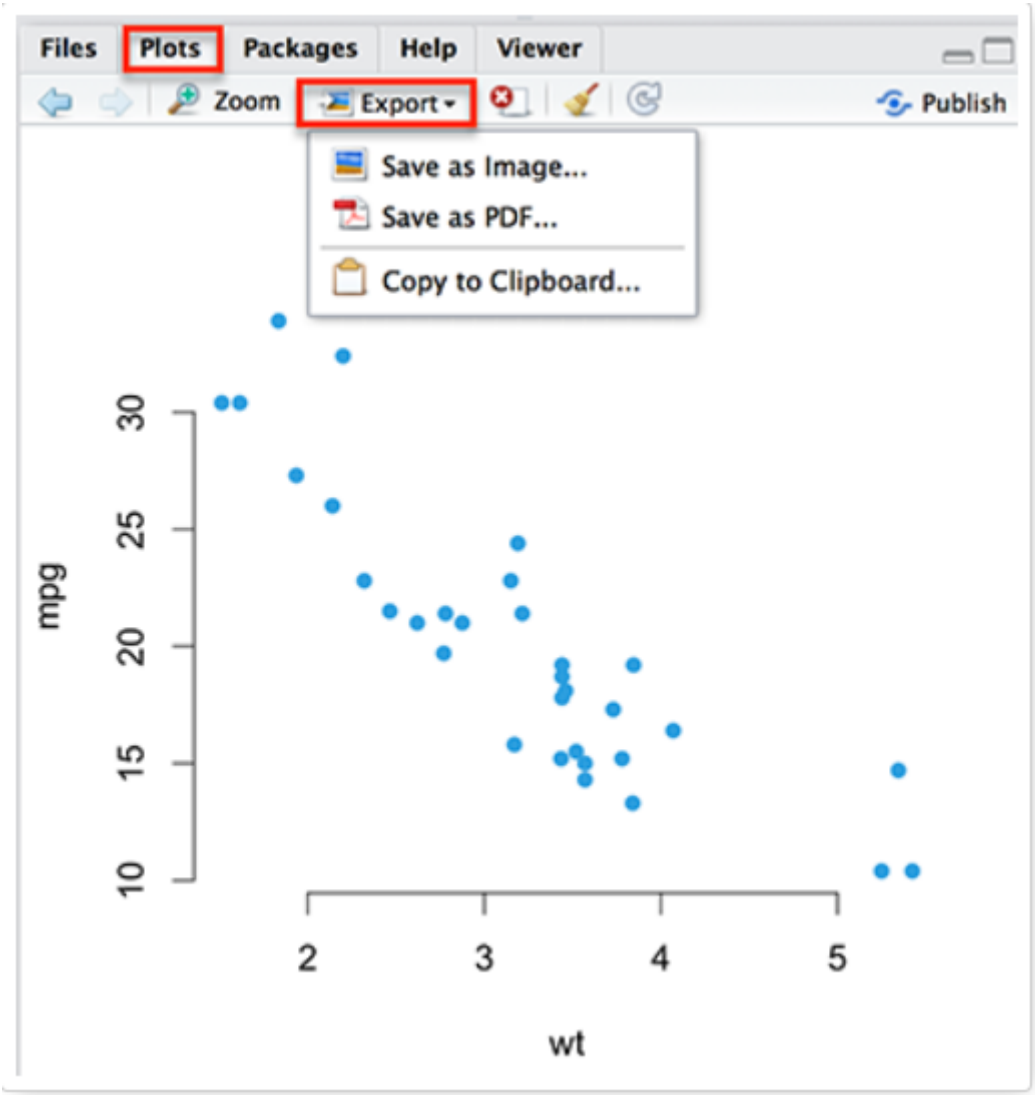

Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
Save as Image...
Save as PDF...
Copy to Clipboard...
mpg
10
15
20
25
30
2
3
4
5
wt

Figure 2.1:

```
# 3. Save the file with dev.off() command in directory
dev.off()
```

```
## pdf
##   2
```

File formats for exporting plots are:

- `pdf("rplot.pdf")` : pdf file
- `png("rplot.png")` : png file
- `jpeg("rplot.jpg")` : jpeg file
- `postscript("rplot.ps")`: postscript file
- `bmp("rplot.bmp")` : bmp file
- `win.metafile("rplot.wmf")` : windows metafile

Chapter 3

Standard graphical formatting in R

Introduction plot() function:

`plot()` function is generic function for plotting of R objects in basic graphs.

- `par()` : the default settings (rows x columns) for plots.
- `plot()` : the main function.
- There are many other plot functions which are specific to some tasks such as `hist()`, `boxplot()`, etc. Most of them take the same arguments as the `plot()` function.

Formula:

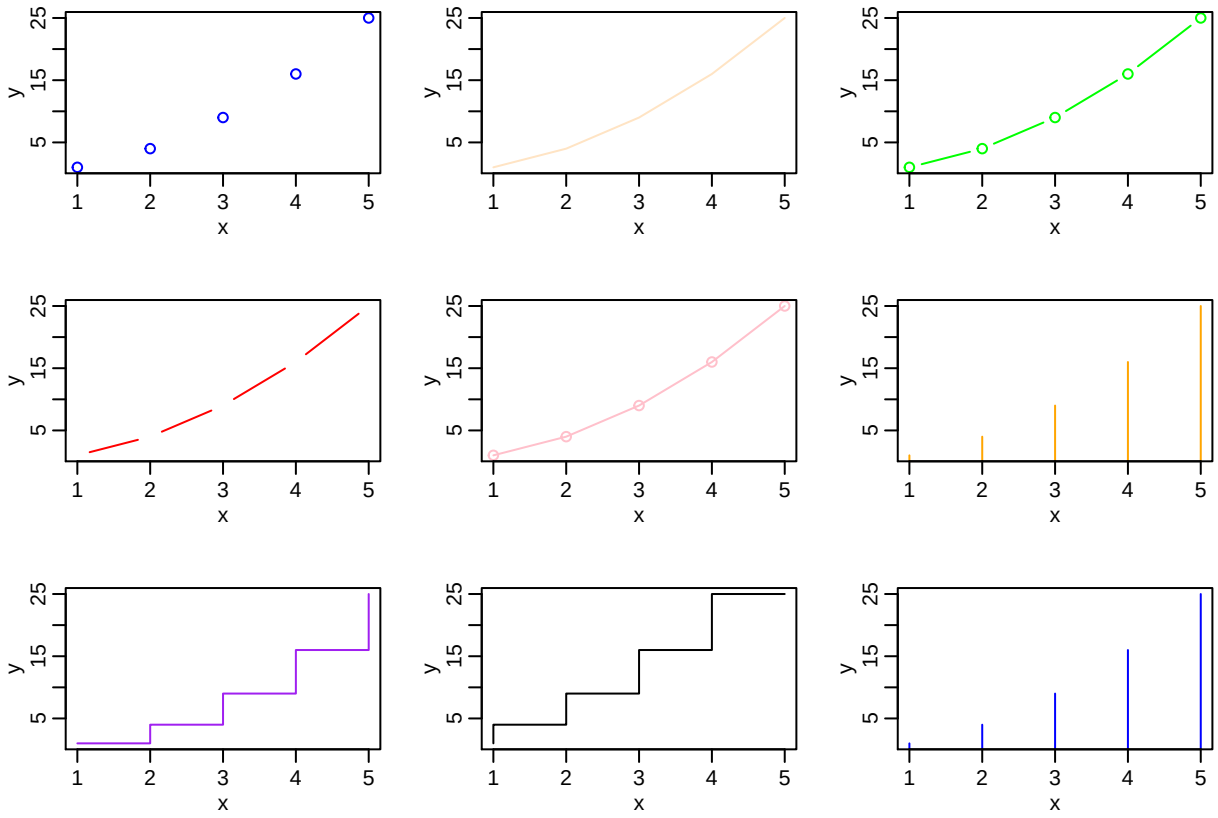
plot(x, y, type="p")

- **x** and **y**: the coordinates of points to plot
- `type`:
  - "p" for points,
  - "l" for lines,
  - "b" for both,
  - "c" for the lines part alone of "b",
  - "o" for both 'overplotted',
  - "h" for 'histogram' like (or 'high-density') vertical lines,
  - "s" for stair steps,
  - "S" for other steps, see 'Details' below,
  - "n" for no plotting.

Example:

```
par(mfrow=c(3,3), mar=c(3,3,1,0)+.5, mgp=c(1.6,.6,0))  # set up the graphics
x<-1:5; y=x*x
plot(x, y, type="p",col="blue")
plot(x, y, type="l",col="bisque")
plot(x,y, type="b",col="green")
plot(x, y, type="c",col="red")
plot(x, y, type="o",col="pink")
plot(x,y, type="h",col="orange")
```

```
plot(x, y, type="s",col="purple")
plot(x, y, type="S",col="black")
plot(x,y, type="h",col="blue")
```



For more color type used the sintaks `color()`

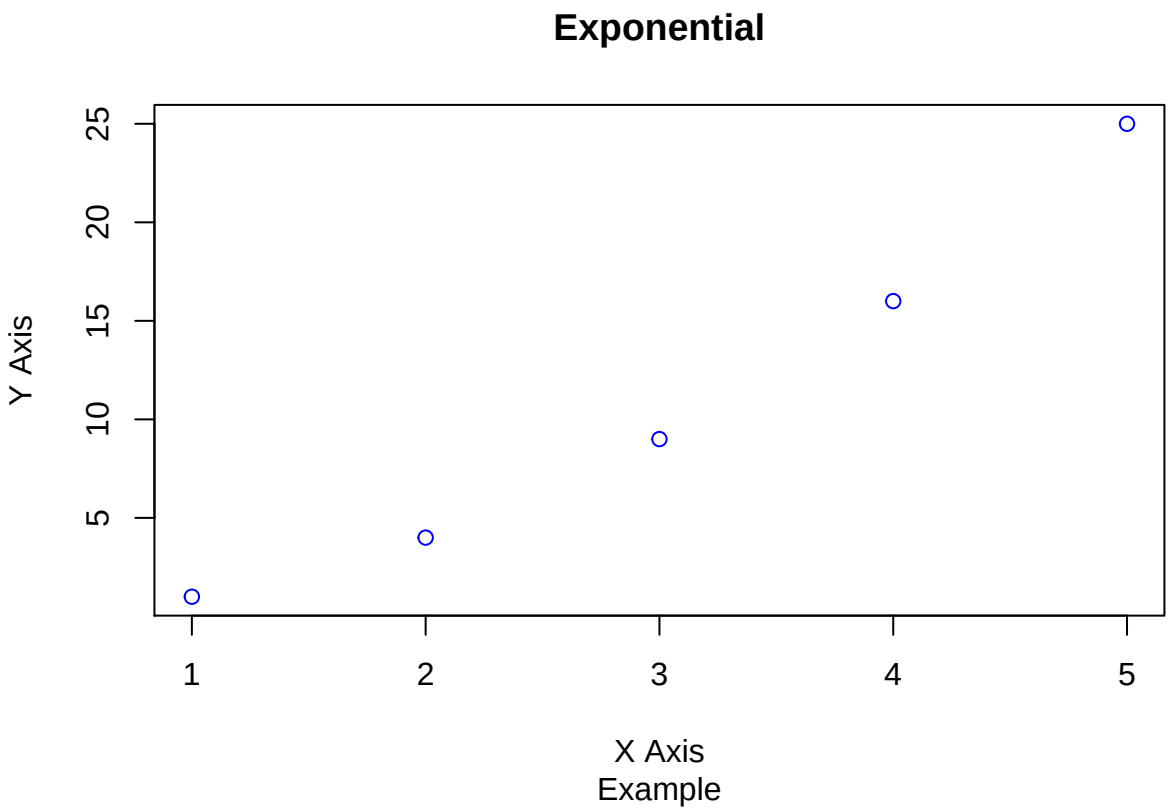
More info about `plot()` go to RDocumentation here

Titles:

- `main="test"`: main titles
- `sub="text"`: subtitle
- `xlab="test"`: the name of the x axis
- `ylab="test"`: the name of the y axis.

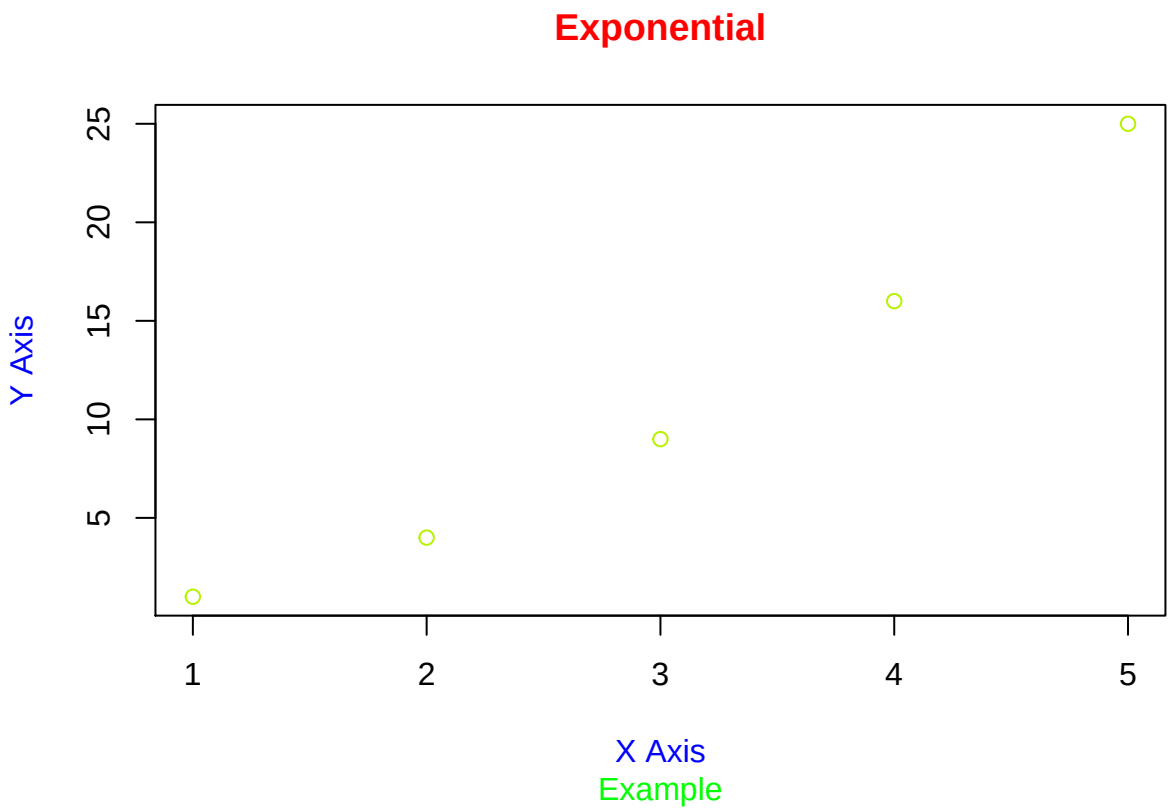
====Main title and axis labels====

```
x<-1:5; y=x*x
plot(x, y, type="p",col="blue",main="Exponential",sub="Example",ylab="Y Axis",xlab="X Axis")
```



====Title colors====

```
x<-1:5; y=x*x
plot(x, y, type="p",col="#B7F200",main="Exponential",sub="Example",ylab="Y Axis",xlab="X Axis",
     col.main="red", col.lab="blue", col.sub="green")
```

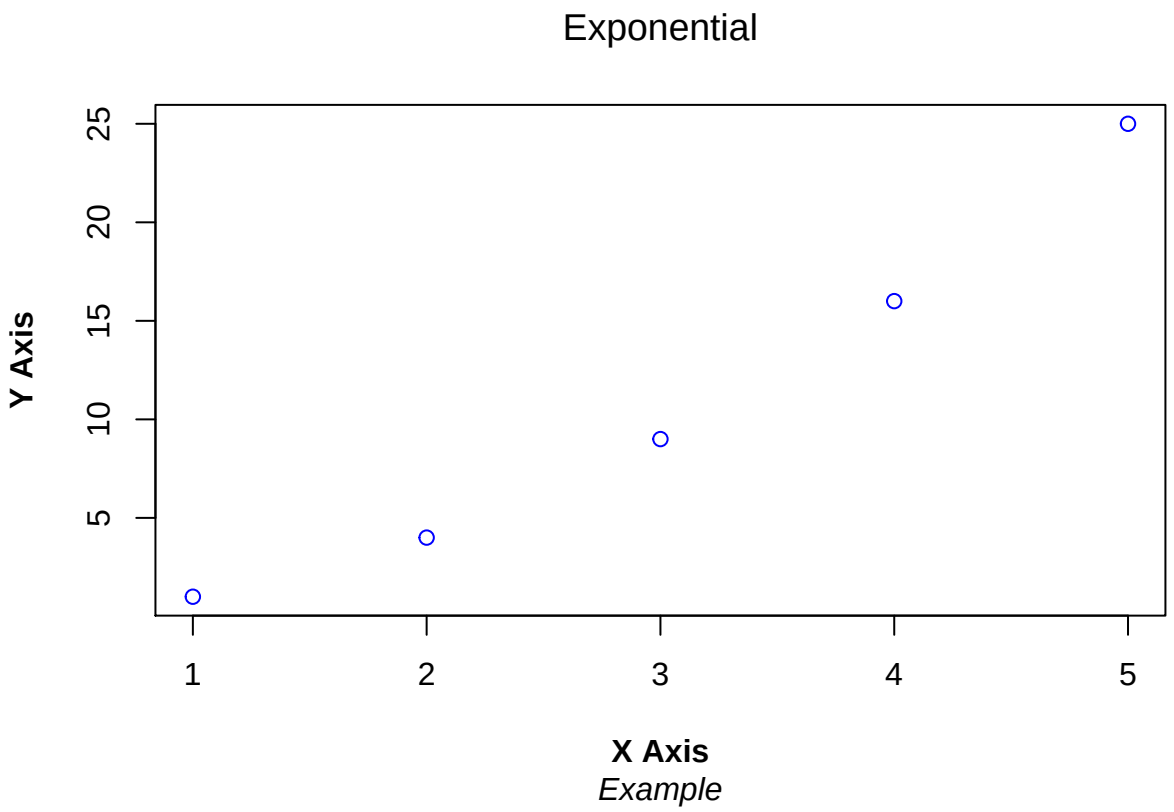


====The font style====

The possible values for the font style are :

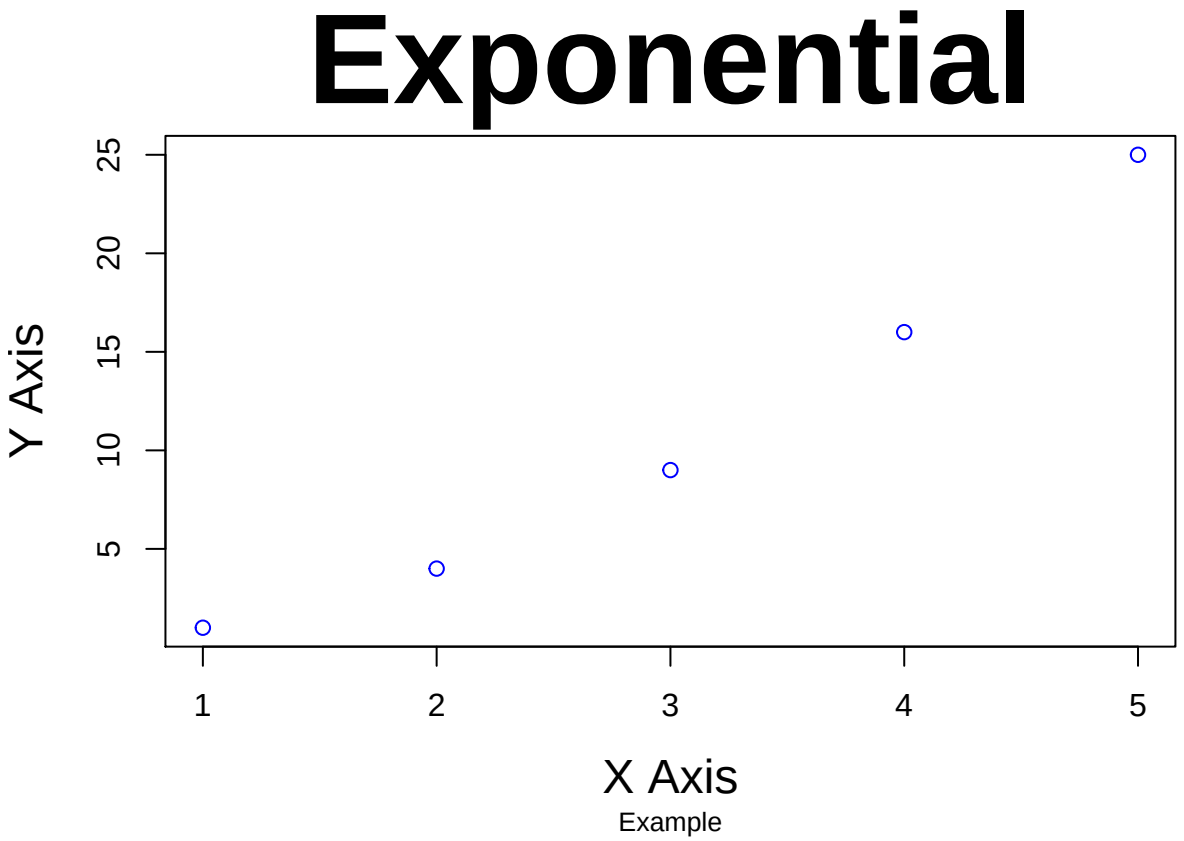
1: normal text 2: bold 3: italic 4: bold and italic 5 : Symbol font

```
x<-1:5; y=x*x
plot(x, y, type="p",col="blue",main="Exponential",sub="Example",ylab="Y Axis",xlab="X Axis",
     font.main=1, font.lab=2, font.sub=3
     )
```



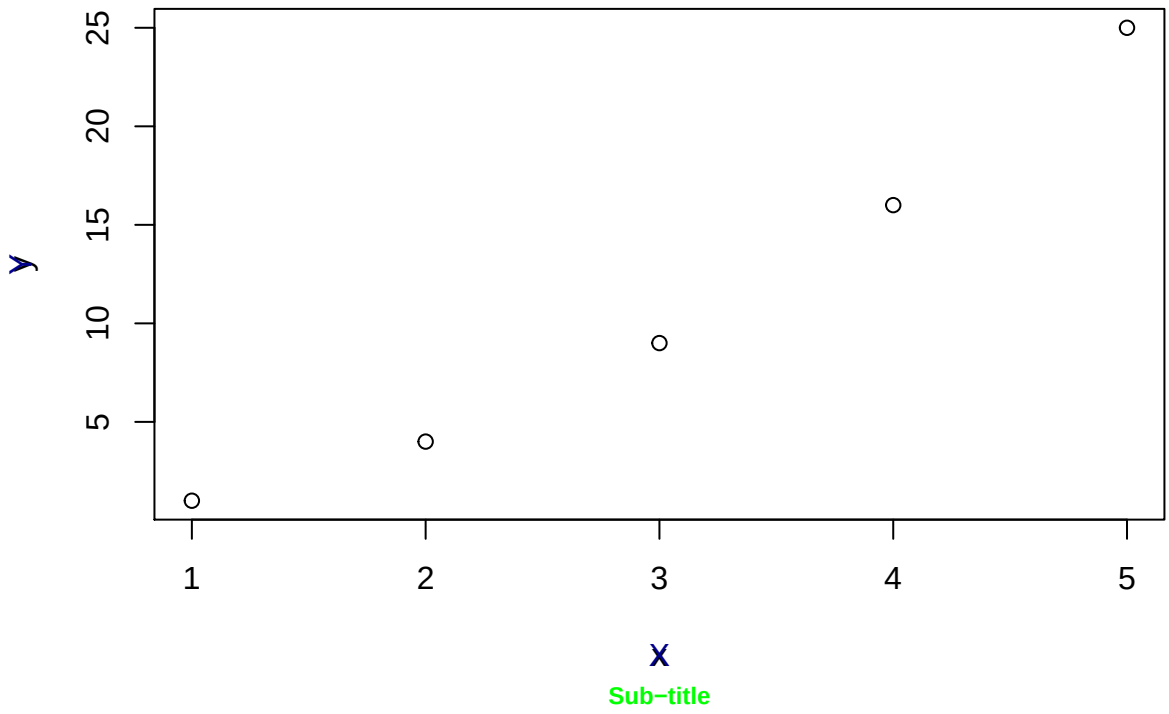
====The font size====

```
x<-1:5; y=x*x
plot(x, y, type="p",col="blue",main="Exponential",sub="Example",ylab="Y Axis",xlab="X Axis",
     cex.main=4, cex.lab=1.5, cex.sub=0.8
     )
```



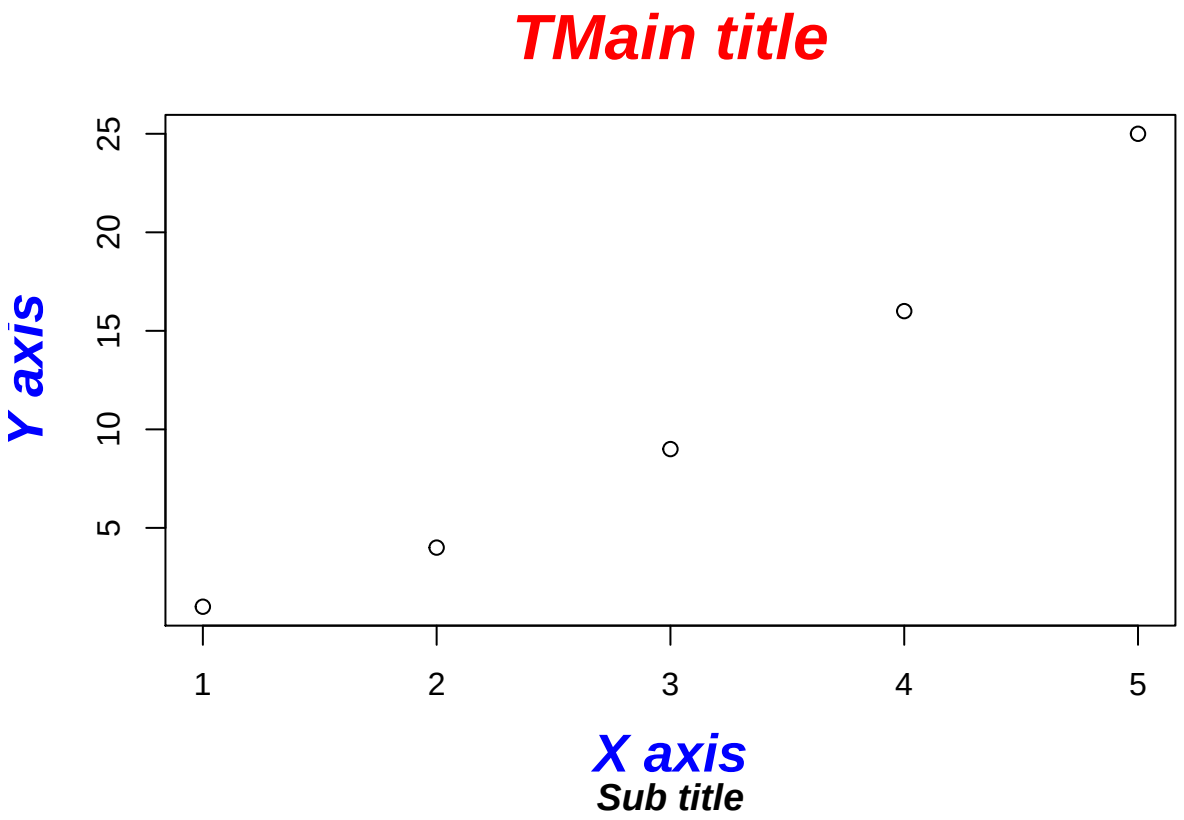
====Use the `title()` function====

```
x<-1:5; y=x*x
plot(x, y, type="p")
title(main = "Main title", sub = "Sub-title",
      xlab = "X", ylab = "Y",
      cex.main = 2,   font.main= 4, col.main= "red",
      cex.sub = 0.75, font.sub = 2, col.sub = "green",
      col.lab ="darkblue"
      )
```



====Customize the titles using `par()` function====

```
par(
  # Change the colors
  col.main="red", col.lab="blue", col.sub="black",
  # Titles in italic and bold
  font.main=4, font.lab=4, font.sub=4,
  # Change font size
  cex.main=2, cex.lab=1.7, cex.sub=1.2
  )
plot(x, y, type="p",
     main="TMain title",
        xlab="X axis",
        ylab="Y axis",
        sub="Sub title"
     )
```



Legend:

`legend()`: the position can be “bottomleft”, “bottomright”, “topleft”, “topright” or exact coordinates.

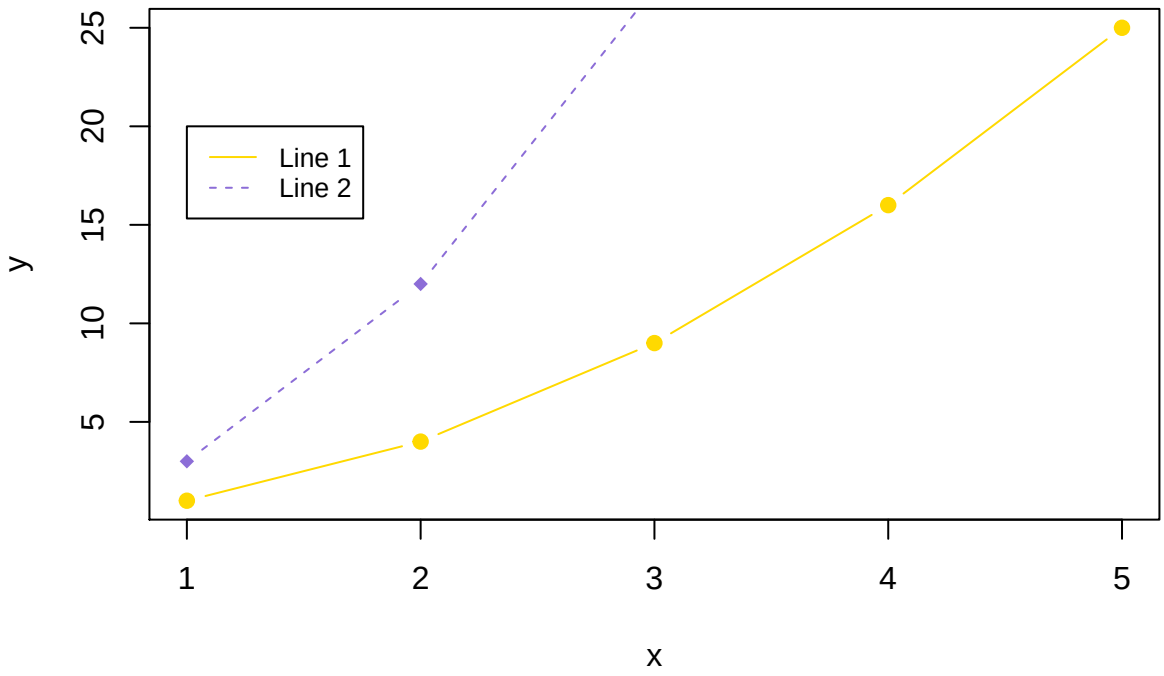
====R legend function====

Formula:

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

Example:

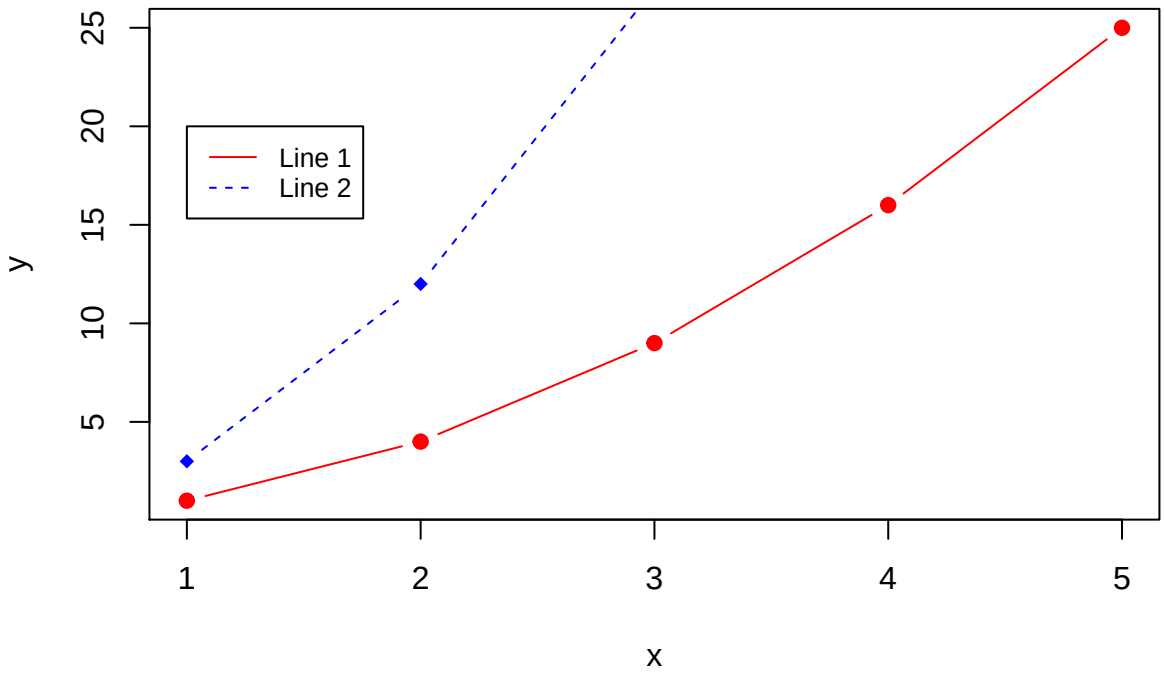
```
# Generate example data
x<-1:5; y1=x*x; y2=3*y1
plot(x, y1, type="b", pch=19, col="#FFD900", xlab="x", ylab="y")
# Add a line
lines(x, y2, pch=18, col="#8C6DD7", type="b", lty=2)
# Add a legend
legend(1, 20, legend=c("Line 1", "Line 2"),
       col=c("#FFD900", "#8C6DD7"), lty=1:2, cex=0.8)
```



Create R function to avoid repeating graphs code:

```
# Generate example data
make_plot<-function(){
x<-1:5; y1=x*x; y2=3*y1
plot(x, y1, type="b", pch=19, col="red", xlab="x", ylab="y")
# Add a line
lines(x, y2, pch=18, col="blue", type="b", lty=2)
# Add a legend
legend(1, 20, legend=c("Line 1", "Line 2"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
}
```

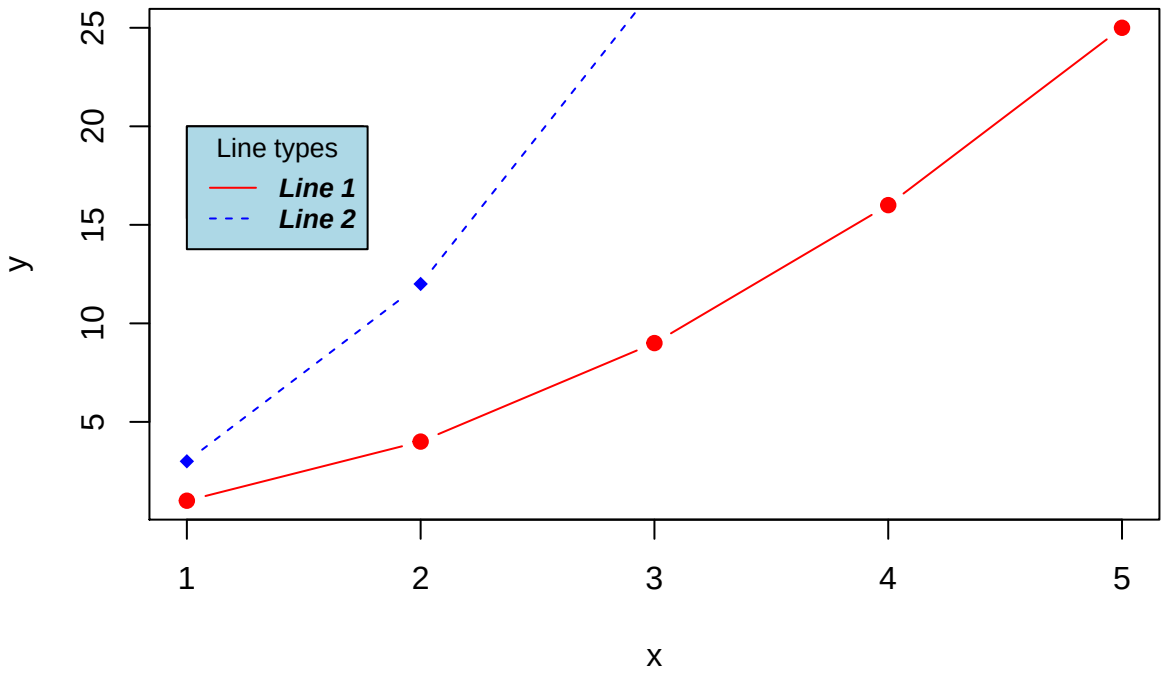
```
make_plot()
```



====Title, text font and background color of the legend box====

- title: The title of the legend
- text.font: an integer specifying the font style of the legend text; possible values are :
  - 1: normal
  - 2: bold
  - 3: italic
  - 4: bold and italic
  - bg: background color of the legend box

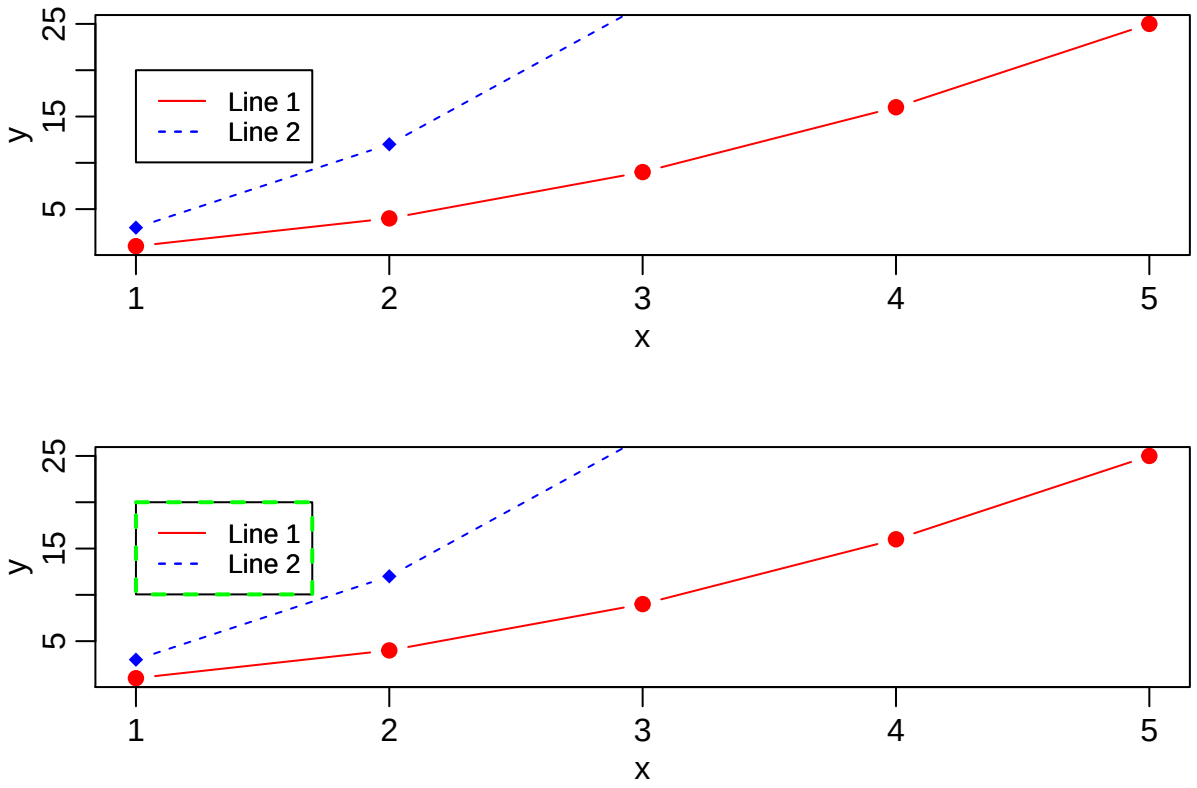
```
make_plot()

# Add a legend to the plot
legend(1,20, legend=c("Line 1", "Line 2"),
       col=c("red", "blue"), lty=1:2, cex=0.8,
       title="Line types", text.font=4, bg="lightblue")
```



====Border of the legend box====

- box.lty: modify the line type
- box.lwd: modify the width
- box.col: modify the color

```
par(mfrow=c(2,1), mar=c(3,3,1,0)+.5, mgp=c(1.6,.6,0))  # set up the graphics
# Remove legend border using box.lty = 0
make_plot()
legend(1,20, legend=c("Line 1", "Line 2"),
       col=c("red", "blue"), lty=1:2, cex=0.8,
       box.lty=0)
# Change the border
make_plot()
legend(1,20, legend=c("Line 1", "Line 2"),
       col=c("red", "blue"), lty=1:2, cex=0.8,
       box.lty=2, box.lwd=2, box.col="green")
```



====The legend position by keywords====

Using the following keywords : “bottomright”, “bottom”, “bottomleft”, “left”, “topleft”, “top”, “topright”, “right” and “center”.

```
par(mfrow=c(2,1), mar=c(3,3,1,0)+.5, mgp=c(1.6,.6,0))  # set up the graphics
# Remove legend border using box.lty = 0
make_plot()
legend("center", legend=c("Line 1", "Line 2"),
       col=c("red", "blue"),box.lty=0)
make_plot()
legend("topright", legend=c("Line 1", "Line 2"),
       col=c("red", "blue"),
       box.lty=0)
```

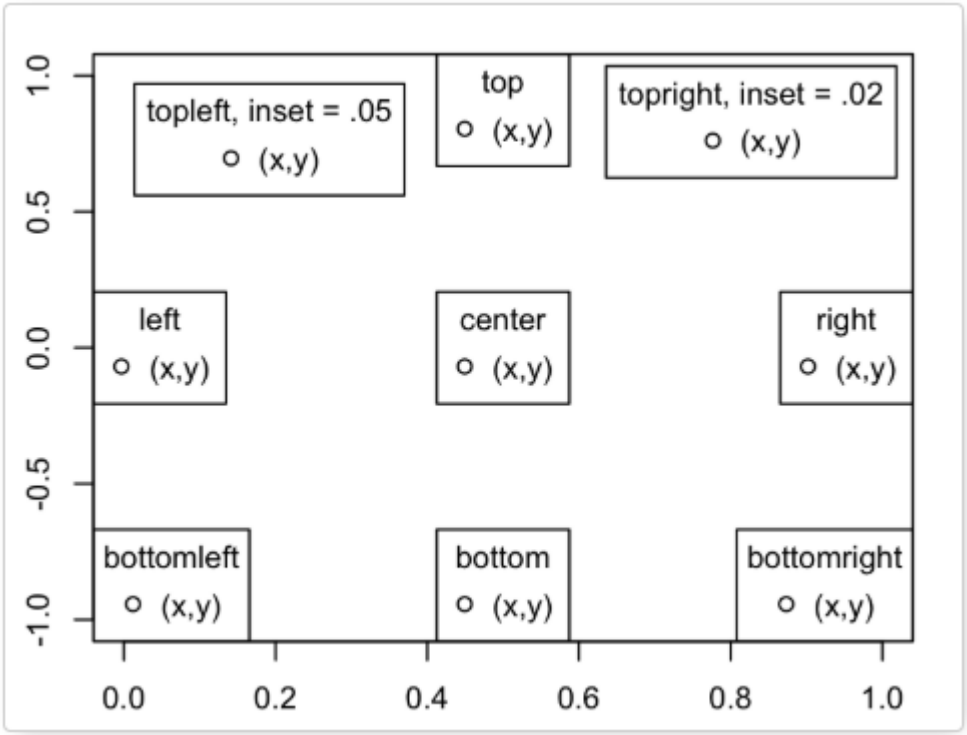
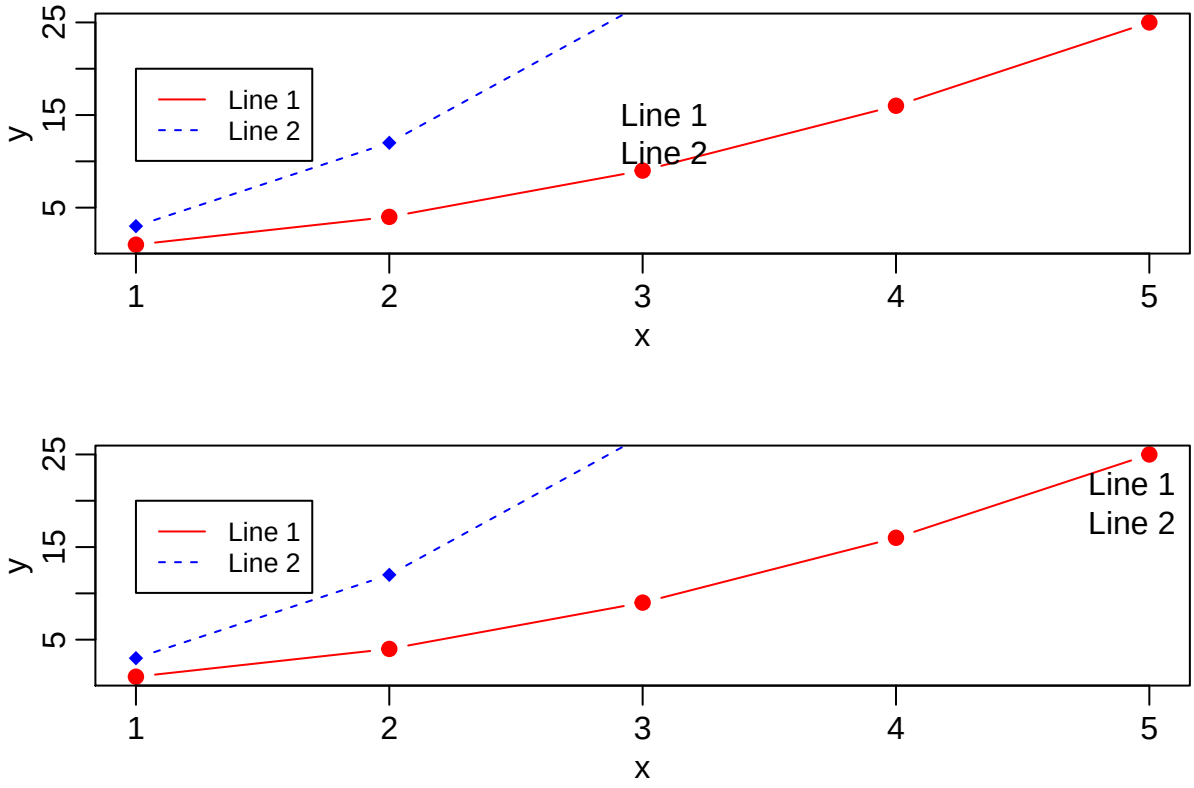

topleft, inset = .05
(x,y)
top
(x,y)
topright, inset = .02
(x,y)
left
(x,y)
center
(x,y)
right
(x,y)
bottomleft
(x,y)
bottom
(x,y)
bottomright
(x,y)
1.0
0.5
0.0
-0.5
-1.0
0.0
0.2
0.4
0.6
0.8
1.0

Figure 3.1:


Line 1
Line 2
Line 1
Line 2
y
x
25
15
5
1
2
3
4
5
Line 1
Line 2
Line 1
Line 2
y
x
25
15
5
1
2
3
4
5

Text:

`text()` and `mtext()` R functions can be used To add a text to a plot.

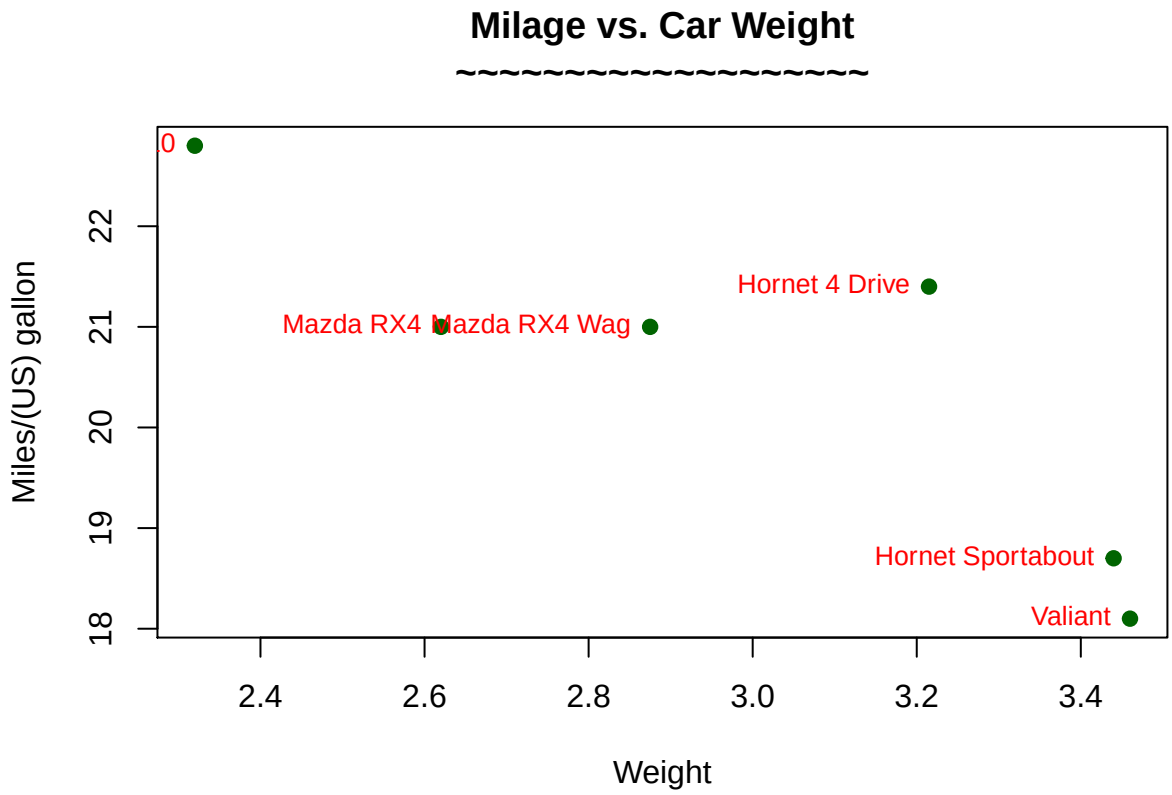
====Texts within the graph====

Formula:

text(x, y, labels)

Example:

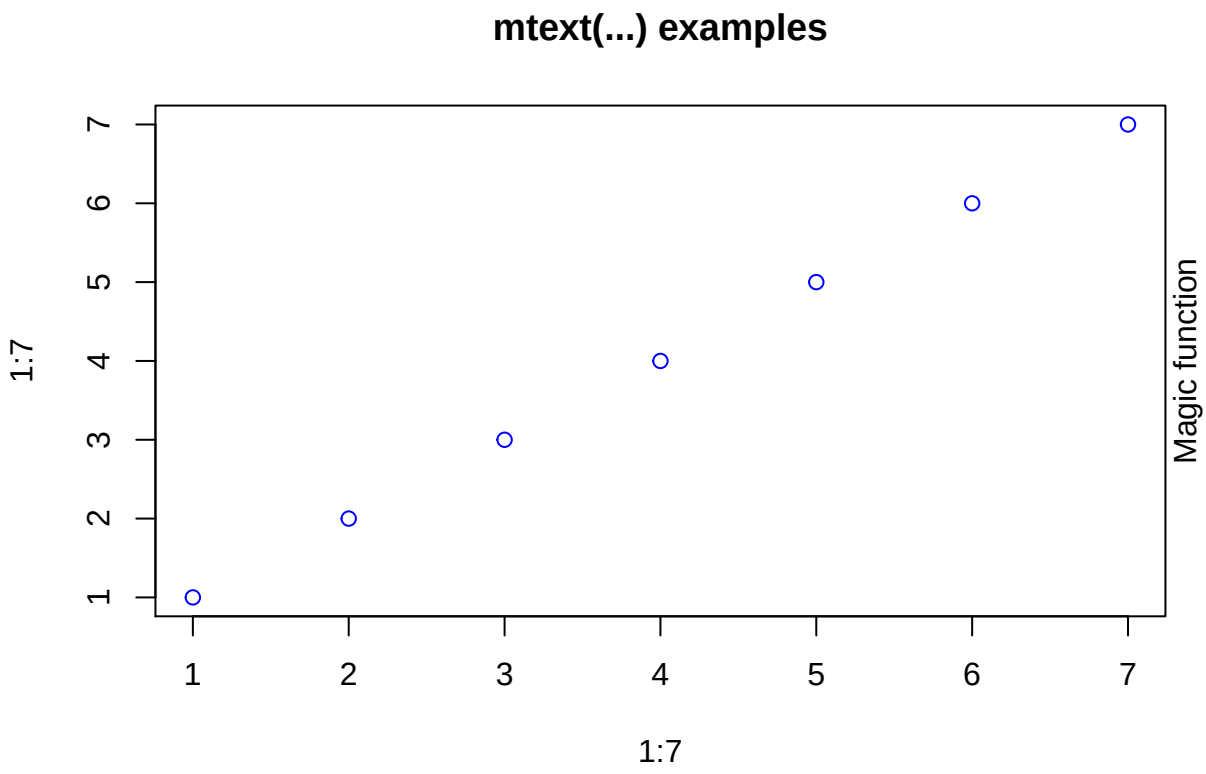
```
d<-head(mtcars)
plot(d[,'wt'], d[,'mpg'],
     main="Milage vs. Car Weight\n~~~~~~~~~~~~~~~~~~",
      xlab="Weight", ylab="Miles/(US) gallon",
      pch=19, col="darkgreen")
text(d[,'wt'], d[,'mpg'],  row.names(d),
     cex=0.85, pos=2,col="red")
```



====Text in the margins of the graph====

- text : the text to be written
- side : an integer specifying the side of the plot; Possible values are :
- 1: bottom
- 2: left
- 3: top
- 4: right

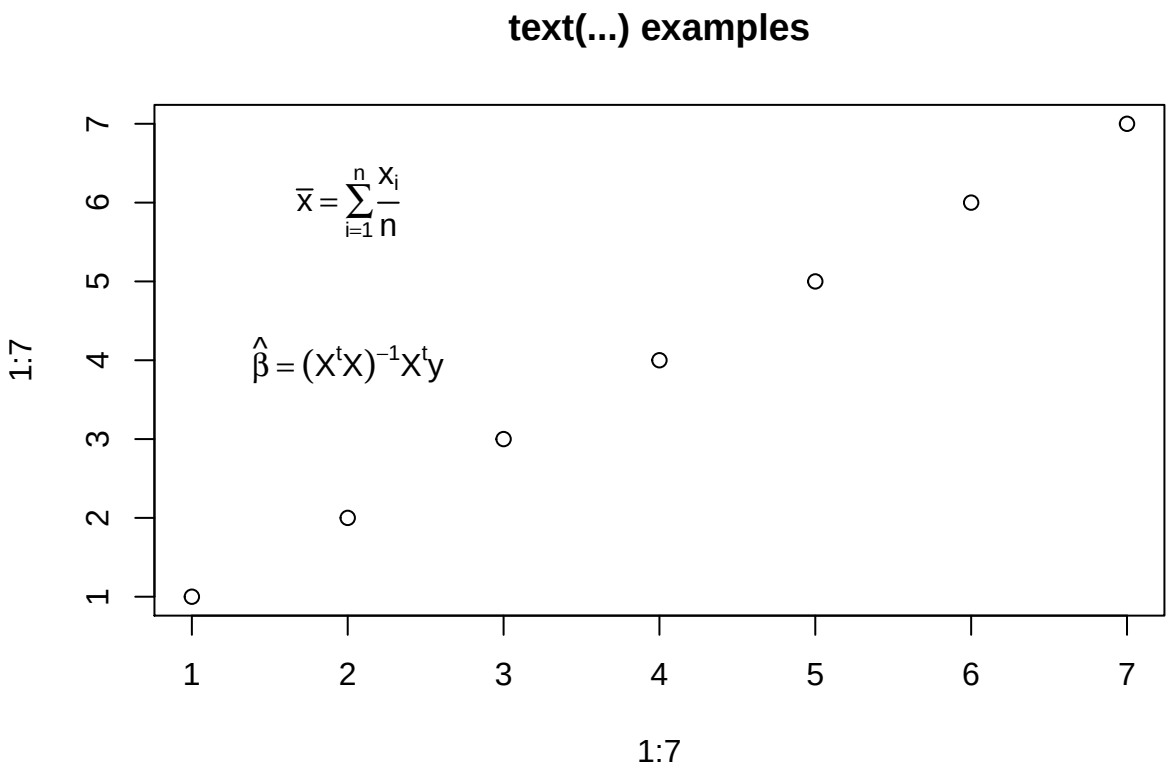
```r
plot(1:7, 1:7,
     main="mtext(...) examples",col="blue")
mtext("Magic function", side=4)
```



====Mathematical annotation within the graph====

Example:

```r
plot(1:7, 1:7,
     main="text(...) examples",col="black")
text(2, 4, expression(hat(beta) == (X^t * X)^{-1} * X^t * y))
text(2, 6, expression(bar(x) == sum(frac(x[i], n), i==1, n)))
```



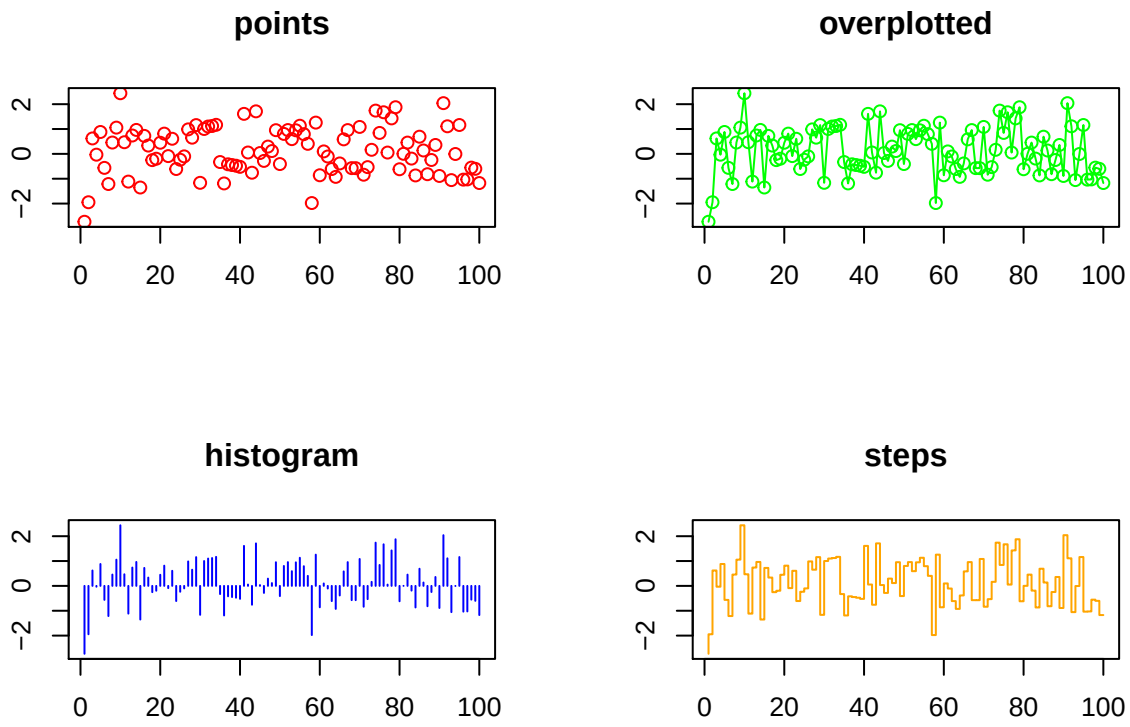
Types of Plot:

An option for plot types can be:

`type`: * “p” for points,

```
  * "l" for lines,

  * "b" for both,

  * "c" for the lines part alone of "b",

  * "o" for both 'overplotted',

  * "h" for 'histogram' like (or 'high-density') vertical lines,

  * "s" for stair steps,

  * "S" for other steps, see 'Details' below,

  * "n" for no plotting.
```

```
x <- rnorm(100)
par(mfrow = c(2,2))
plot(x, type = "p", main = "points", ylab = "", xlab = "",col="red")
plot(x, type = "o", main = "overplotted", ylab = "", xlab = "",col="green")
plot(x, type = "h", main = "histogram", ylab = "", xlab = "",col="blue")
plot(x, type = "S", main = "steps", ylab = "", xlab = "",col="orange")
```



Axis:

====Add an axis to a plot====

Formula:

axis(side, at=NULL, labels=TRUE)

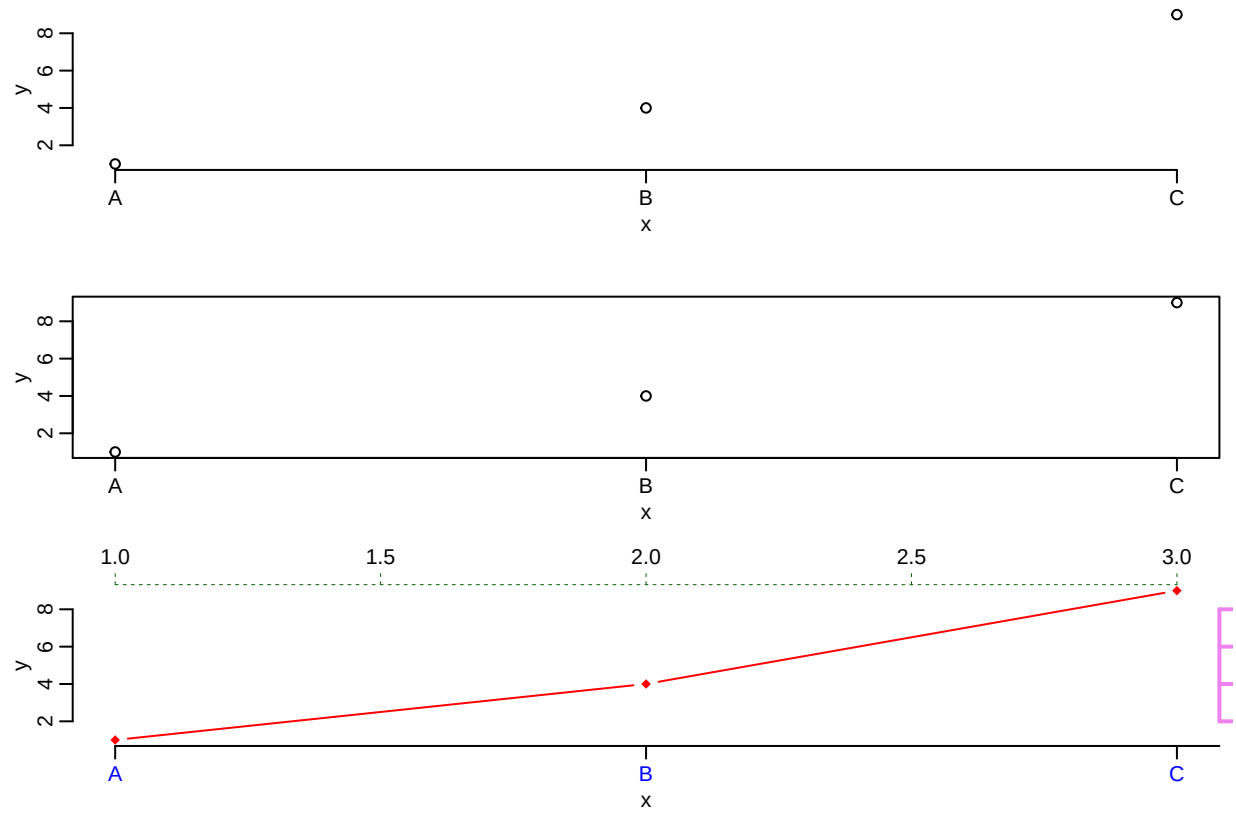
Side :

- 1: below
- 2: left
- 3: above
- 4: right
- at: The position
- labels: Texts for tick-mark labels.

Example :

```
par(mfrow=c(3,1), mar=c(3,3,1,0)+.5, mgp=c(1.6,.6,0))  # set up the graphics
x<-1:3; y=x*x
# Example 1
plot(x, y, axes = FALSE)
axis(side=1, at = 1:3, labels=LETTERS[1:3])
axis(2)
# Example 2
plot(x, y, axes = FALSE)
```

```
axis(side=1, at=1:3, labels=LETTERS[1:3])
axis(2)
box() #- To make it look like "usual" plot
#Example3:
plot(x, y, pch=18, col="red", type="b",
     frame=FALSE, xaxt="n") # Remove x axis
axis(1, 1:4, LETTERS[1:4], col.axis="blue")
axis(3, col = "darkgreen", lty = 2, lwd = 0.5)
axis(4, col = "violet", col.axis = "dark violet", lwd = 2)
```



====Axis scale====

- xlim: the limit of x axis; format : xlim = c(min, max)
- ylim: the limit of y axis; format: ylim = c(min, max)

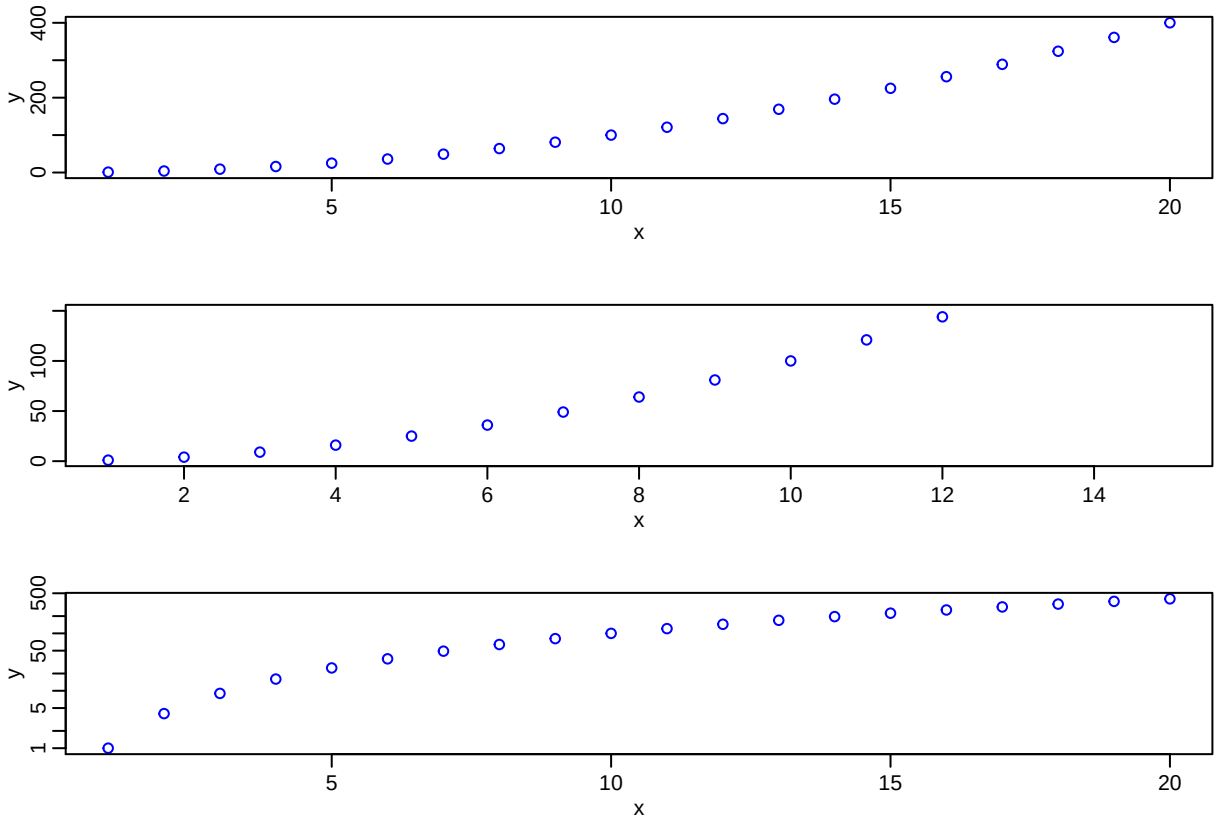
Transformation to log scale:

- log = "x"
- log = "y"
- log = "xy"*

Example:

```
par(mfrow=c(3,1), mar=c(3,3,1,0)+.5, mgp=c(1.6,.6,0))  # set up the graphics
x<-1:20; y=x*x
# Simple graph
plot(x, y,col="blue")
```

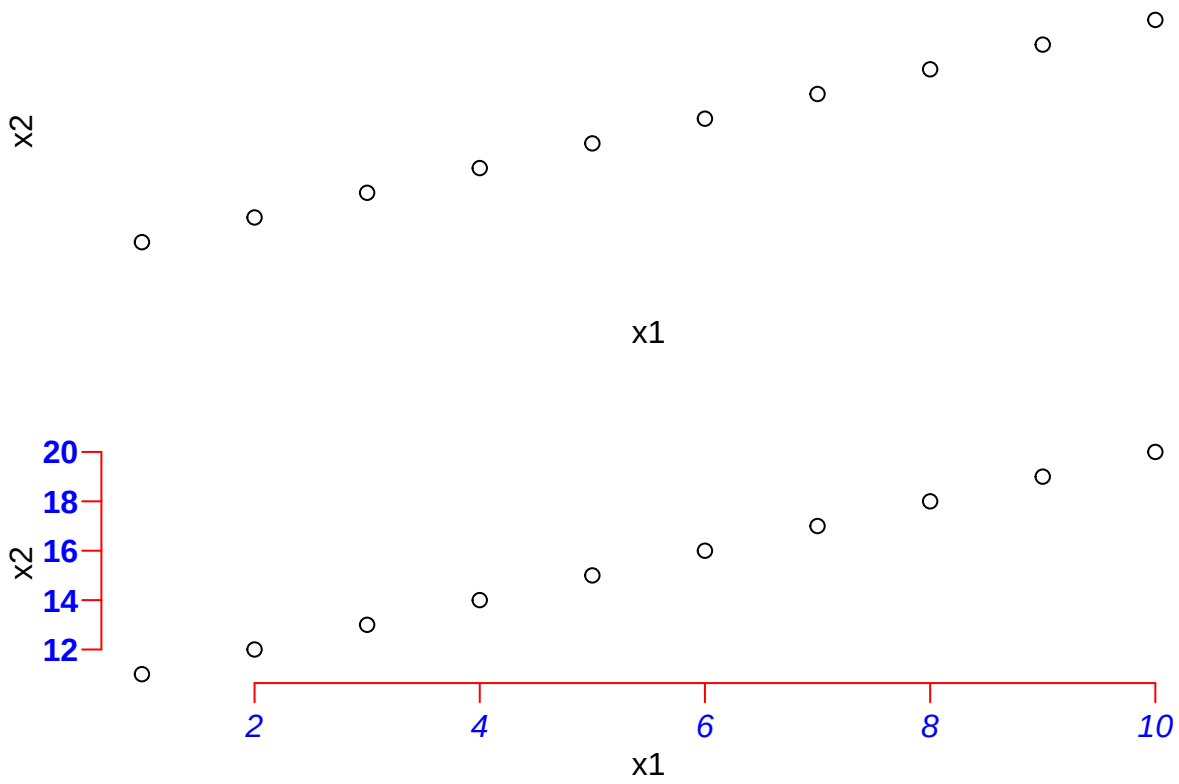
```r
# Enlarge the scale
plot(x, y, xlim=c(1,15), ylim=c(1,150),col="blue")
# Log scale
plot(x, y, log="y",col="blue")
```



====Axes fonts====

Remove them with `axes=FALSE`:

```r
par(mfrow=c(2,1), mar=c(3,3,1,0)+.5, mgp=c(1.6,.6,0))  # set up the graphics
x1<-1:10
x2<-11:20
plot(x1,x2,axes=FALSE)
plot(x1,x2,axes=FALSE)
axis(1,col="red",col.axis="blue",font.axis=3)
axis(2,col="red",col.axis="blue",font.axis=2,las=2)
```

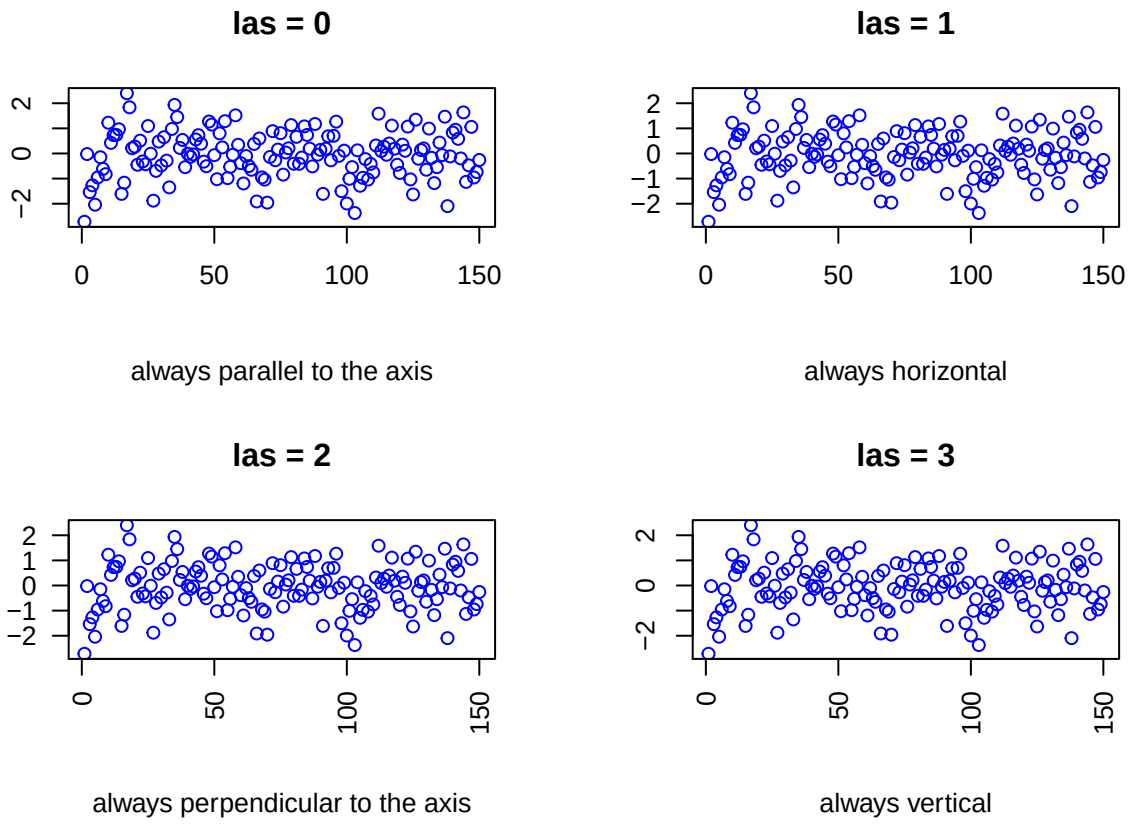


====Axes positions====

Use `las` function to specified the style of axis labels:

- 0 : always parallel to the axis [default]
- 1 : always horizontal
- 2 : always perpendicular to the axis
- 3 : always vertical

```
x1 <- rnorm(150)
par(mfrow = c(2,2))
plot(x1, las = 0, main = "las = 0", sub = "always parallel to the axis", xlab = "", ylab = "",col="blue
plot(x1, las = 1, main = "las = 1", sub = "always horizontal", xlab = "", ylab = "",col="blue")
plot(x1, las = 2, main = "las = 2", sub = "always perpendicular to the axis", xlab = "", ylab = "",col=
plot(x1, las = 3, main = "las = 3", sub = "always vertical", xlab = "", ylab = "",col="blue")
```



Margins `par()`:

Formula:

par(mfrow=c(2,1), mar=c(3,3,1,0)+.5, mgp=c(1.6,.6,0)) # set up the graphics

See `?par` to learn more about the topic.

Colors:

Use by name (e.g col = "red") or as a hexadecimal RGB triplet (such as col = "#FFCC00").

====Default color in R====

```
colors() # list the r colors
```

```
##   [1] "white"               "aliceblue"           "antiquewhite"
##   [4] "antiquewhite1"       "antiquewhite2"       "antiquewhite3"
##   [7] "antiquewhite4"       "aquamarine"          "aquamarine1"
##  [10] "aquamarine2"         "aquamarine3"         "aquamarine4"
##  [13] "azure"               "azure1"              "azure2"
##  [16] "azure3"              "azure4"              "beige"
##  [19] "bisque"              "bisque1"             "bisque2"
##  [22] "bisque3"             "bisque4"             "black"
##  [25] "blanchedalmond"      "blue"                "blue1"
##  [28] "blue2"               "blue3"               "blue4"
##  [31] "blueviolet"          "brown"               "brown1"
##  [34] "brown2"              "brown3"              "brown4"
##  [37] "burlywood"           "burlywood1"          "burlywood2"
##  [40] "burlywood3"          "burlywood4"          "cadetblue"
```

```
##  [43] "cadetblue1"          "cadetblue2"          "cadetblue3"
##  [46] "cadetblue4"          "chartreuse"          "chartreuse1"
##  [49] "chartreuse2"         "chartreuse3"         "chartreuse4"
##  [52] "chocolate"           "chocolate1"          "chocolate2"
##  [55] "chocolate3"          "chocolate4"          "coral"
##  [58] "coral1"              "coral2"              "coral3"
##  [61] "coral4"              "cornflowerblue"      "cornsilk"
##  [64] "cornsilk1"           "cornsilk2"           "cornsilk3"
##  [67] "cornsilk4"           "cyan"                "cyan1"
##  [70] "cyan2"               "cyan3"               "cyan4"
##  [73] "darkblue"            "darkcyan"            "darkgoldenrod"
##  [76] "darkgoldenrod1"      "darkgoldenrod2"      "darkgoldenrod3"
##  [79] "darkgoldenrod4"      "darkgray"            "darkgreen"
##  [82] "darkgrey"            "darkkhaki"           "darkmagenta"
##  [85] "darkolivegreen"      "darkolivegreen1"     "darkolivegreen2"
##  [88] "darkolivegreen3"     "darkolivegreen4"     "darkorange"
##  [91] "darkorange1"         "darkorange2"         "darkorange3"
##  [94] "darkorange4"         "darkorchid"          "darkorchid1"
##  [97] "darkorchid2"         "darkorchid3"         "darkorchid4"
## [100] "darkred"             "darksalmon"          "darkseagreen"
## [103] "darkseagreen1"       "darkseagreen2"       "darkseagreen3"
## [106] "darkseagreen4"       "darkslateblue"       "darkslategray"
## [109] "darkslategray1"      "darkslategray2"      "darkslategray3"
## [112] "darkslategray4"      "darkslategrey"       "darkturquoise"
## [115] "darkviolet"          "deeppink"            "deeppink1"
## [118] "deeppink2"           "deeppink3"           "deeppink4"
## [121] "deepskyblue"         "deepskyblue1"        "deepskyblue2"
## [124] "deepskyblue3"        "deepskyblue4"        "dimgray"
## [127] "dimgrey"             "dodgerblue"          "dodgerblue1"
## [130] "dodgerblue2"         "dodgerblue3"         "dodgerblue4"
## [133] "firebrick"           "firebrick1"          "firebrick2"
## [136] "firebrick3"          "firebrick4"          "floralwhite"
## [139] "forestgreen"         "gainsboro"           "ghostwhite"
## [142] "gold"                "gold1"               "gold2"
## [145] "gold3"               "gold4"               "goldenrod"
## [148] "goldenrod1"          "goldenrod2"          "goldenrod3"
## [151] "goldenrod4"          "gray"                "gray0"
## [154] "gray1"               "gray2"               "gray3"
## [157] "gray4"               "gray5"               "gray6"
## [160] "gray7"               "gray8"               "gray9"
## [163] "gray10"              "gray11"              "gray12"
## [166] "gray13"              "gray14"              "gray15"
## [169] "gray16"              "gray17"              "gray18"
## [172] "gray19"              "gray20"              "gray21"
## [175] "gray22"              "gray23"              "gray24"
## [178] "gray25"              "gray26"              "gray27"
## [181] "gray28"              "gray29"              "gray30"
## [184] "gray31"              "gray32"              "gray33"
## [187] "gray34"              "gray35"              "gray36"
## [190] "gray37"              "gray38"              "gray39"
## [193] "gray40"              "gray41"              "gray42"
## [196] "gray43"              "gray44"              "gray45"
## [199] "gray46"              "gray47"              "gray48"
## [202] "gray49"              "gray50"              "gray51"
```

```
## [205] "gray52"             "gray53"             "gray54"
## [208] "gray55"             "gray56"             "gray57"
## [211] "gray58"             "gray59"             "gray60"
## [214] "gray61"             "gray62"             "gray63"
## [217] "gray64"             "gray65"             "gray66"
## [220] "gray67"             "gray68"             "gray69"
## [223] "gray70"             "gray71"             "gray72"
## [226] "gray73"             "gray74"             "gray75"
## [229] "gray76"             "gray77"             "gray78"
## [232] "gray79"             "gray80"             "gray81"
## [235] "gray82"             "gray83"             "gray84"
## [238] "gray85"             "gray86"             "gray87"
## [241] "gray88"             "gray89"             "gray90"
## [244] "gray91"             "gray92"             "gray93"
## [247] "gray94"             "gray95"             "gray96"
## [250] "gray97"             "gray98"             "gray99"
## [253] "gray100"            "green"              "green1"
## [256] "green2"             "green3"             "green4"
## [259] "greenyellow"        "grey"               "grey0"
## [262] "grey1"              "grey2"              "grey3"
## [265] "grey4"              "grey5"              "grey6"
## [268] "grey7"              "grey8"              "grey9"
## [271] "grey10"             "grey11"             "grey12"
## [274] "grey13"             "grey14"             "grey15"
## [277] "grey16"             "grey17"             "grey18"
## [280] "grey19"             "grey20"             "grey21"
## [283] "grey22"             "grey23"             "grey24"
## [286] "grey25"             "grey26"             "grey27"
## [289] "grey28"             "grey29"             "grey30"
## [292] "grey31"             "grey32"             "grey33"
## [295] "grey34"             "grey35"             "grey36"
## [298] "grey37"             "grey38"             "grey39"
## [301] "grey40"             "grey41"             "grey42"
## [304] "grey43"             "grey44"             "grey45"
## [307] "grey46"             "grey47"             "grey48"
## [310] "grey49"             "grey50"             "grey51"
## [313] "grey52"             "grey53"             "grey54"
## [316] "grey55"             "grey56"             "grey57"
## [319] "grey58"             "grey59"             "grey60"
## [322] "grey61"             "grey62"             "grey63"
## [325] "grey64"             "grey65"             "grey66"
## [328] "grey67"             "grey68"             "grey69"
## [331] "grey70"             "grey71"             "grey72"
## [334] "grey73"             "grey74"             "grey75"
## [337] "grey76"             "grey77"             "grey78"
## [340] "grey79"             "grey80"             "grey81"
## [343] "grey82"             "grey83"             "grey84"
## [346] "grey85"             "grey86"             "grey87"
## [349] "grey88"             "grey89"             "grey90"
## [352] "grey91"             "grey92"             "grey93"
## [355] "grey94"             "grey95"             "grey96"
## [358] "grey97"             "grey98"             "grey99"
## [361] "grey100"            "honeydew"           "honeydew1"
## [364] "honeydew2"          "honeydew3"          "honeydew4"
```

```
## [367] "hotpink"              "hotpink1"             "hotpink2"
## [370] "hotpink3"             "hotpink4"             "indianred"
## [373] "indianred1"           "indianred2"           "indianred3"
## [376] "indianred4"           "ivory"                "ivory1"
## [379] "ivory2"               "ivory3"               "ivory4"
## [382] "khaki"                "khaki1"               "khaki2"
## [385] "khaki3"               "khaki4"               "lavender"
## [388] "lavenderblush"        "lavenderblush1"       "lavenderblush2"
## [391] "lavenderblush3"       "lavenderblush4"       "lawngreen"
## [394] "lemonchiffon"         "lemonchiffon1"        "lemonchiffon2"
## [397] "lemonchiffon3"        "lemonchiffon4"        "lightblue"
## [400] "lightblue1"           "lightblue2"           "lightblue3"
## [403] "lightblue4"           "lightcoral"           "lightcyan"
## [406] "lightcyan1"           "lightcyan2"           "lightcyan3"
## [409] "lightcyan4"           "lightgoldenrod"       "lightgoldenrod1"
## [412] "lightgoldenrod2"      "lightgoldenrod3"      "lightgoldenrod4"
## [415] "lightgoldenrodyellow" "lightgray"            "lightgreen"
## [418] "lightgrey"            "lightpink"            "lightpink1"
## [421] "lightpink2"           "lightpink3"           "lightpink4"
## [424] "lightsalmon"          "lightsalmon1"         "lightsalmon2"
## [427] "lightsalmon3"         "lightsalmon4"         "lightseagreen"
## [430] "lightskyblue"         "lightskyblue1"        "lightskyblue2"
## [433] "lightskyblue3"        "lightskyblue4"        "lightslateblue"
## [436] "lightslategray"       "lightslategrey"       "lightsteelblue"
## [439] "lightsteelblue1"      "lightsteelblue2"      "lightsteelblue3"
## [442] "lightsteelblue4"      "lightyellow"          "lightyellow1"
## [445] "lightyellow2"         "lightyellow3"         "lightyellow4"
## [448] "limegreen"            "linen"                "magenta"
## [451] "magenta1"             "magenta2"             "magenta3"
## [454] "magenta4"             "maroon"               "maroon1"
## [457] "maroon2"              "maroon3"              "maroon4"
## [460] "mediumaquamarine"     "mediumblue"           "mediumorchid"
## [463] "mediumorchid1"        "mediumorchid2"        "mediumorchid3"
## [466] "mediumorchid4"        "mediumpurple"         "mediumpurple1"
## [469] "mediumpurple2"        "mediumpurple3"        "mediumpurple4"
## [472] "mediumseagreen"       "mediumslateblue"      "mediumspringgreen"
## [475] "mediumturquoise"      "mediumvioletred"      "midnightblue"
## [478] "mintcream"            "mistyrose"            "mistyrose1"
## [481] "mistyrose2"           "mistyrose3"           "mistyrose4"
## [484] "moccasin"             "navajowhite"          "navajowhite1"
## [487] "navajowhite2"         "navajowhite3"         "navajowhite4"
## [490] "navy"                 "navyblue"             "oldlace"
## [493] "olivedrab"            "olivedrab1"           "olivedrab2"
## [496] "olivedrab3"           "olivedrab4"           "orange"
## [499] "orange1"              "orange2"              "orange3"
## [502] "orange4"              "orangered"            "orangered1"
## [505] "orangered2"           "orangered3"           "orangered4"
## [508] "orchid"               "orchid1"              "orchid2"
## [511] "orchid3"              "orchid4"              "palegoldenrod"
## [514] "palegreen"            "palegreen1"           "palegreen2"
## [517] "palegreen3"           "palegreen4"           "paleturquoise"
## [520] "paleturquoise1"       "paleturquoise2"       "paleturquoise3"
## [523] "paleturquoise4"       "palevioletred"        "palevioletred1"
## [526] "palevioletred2"       "palevioletred3"       "palevioletred4"
```

```
## [529] "papayawhip"          "peachpuff"           "peachpuff1"
## [532] "peachpuff2"          "peachpuff3"          "peachpuff4"
## [535] "peru"                "pink"                "pink1"
## [538] "pink2"               "pink3"               "pink4"
## [541] "plum"                "plum1"               "plum2"
## [544] "plum3"               "plum4"               "powderblue"
## [547] "purple"              "purple1"             "purple2"
## [550] "purple3"             "purple4"             "red"
## [553] "red1"                "red2"                "red3"
## [556] "red4"                "rosybrown"           "rosybrown1"
## [559] "rosybrown2"          "rosybrown3"          "rosybrown4"
## [562] "royalblue"           "royalblue1"          "royalblue2"
## [565] "royalblue3"          "royalblue4"          "saddlebrown"
## [568] "salmon"              "salmon1"             "salmon2"
## [571] "salmon3"             "salmon4"             "sandybrown"
## [574] "seagreen"            "seagreen1"           "seagreen2"
## [577] "seagreen3"           "seagreen4"           "seashell"
## [580] "seashell1"           "seashell2"           "seashell3"
## [583] "seashell4"           "sienna"              "sienna1"
## [586] "sienna2"             "sienna3"             "sienna4"
## [589] "skyblue"             "skyblue1"            "skyblue2"
## [592] "skyblue3"            "skyblue4"            "slateblue"
## [595] "slateblue1"          "slateblue2"          "slateblue3"
## [598] "slateblue4"          "slategray"           "slategray1"
## [601] "slategray2"          "slategray3"          "slategray4"
## [604] "slategrey"           "snow"                "snow1"
## [607] "snow2"               "snow3"               "snow4"
## [610] "springgreen"         "springgreen1"        "springgreen2"
## [613] "springgreen3"        "springgreen4"        "steelblue"
## [616] "steelblue1"          "steelblue2"          "steelblue3"
## [619] "steelblue4"          "tan"                 "tan1"
## [622] "tan2"                "tan3"                "tan4"
## [625] "thistle"             "thistle1"            "thistle2"
## [628] "thistle3"            "thistle4"            "tomato"
## [631] "tomato1"             "tomato2"             "tomato3"
## [634] "tomato4"             "turquoise"           "turquoise1"
## [637] "turquoise2"          "turquoise3"          "turquoise4"
## [640] "violet"              "violetred"           "violetred1"
## [643] "violetred2"          "violetred3"          "violetred4"
## [646] "wheat"               "wheat1"              "wheat2"
## [649] "wheat3"              "wheat4"              "whitesmoke"
## [652] "yellow"              "yellow1"             "yellow2"
## [655] "yellow3"             "yellow4"             "yellowgreen"
```

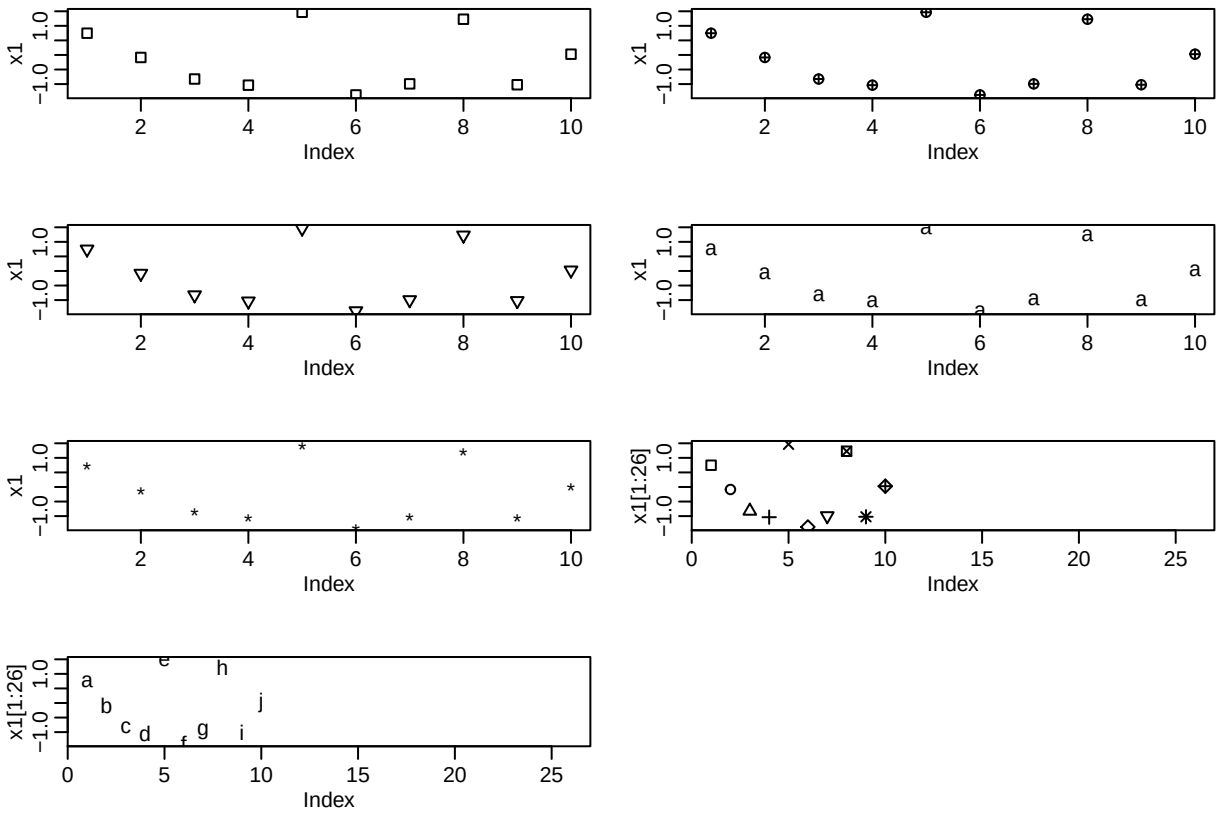
More info about color scheme extractor code in R:

- Adobe Color (Adobe Kuler)
- Color Schema Designer
- Color Calculator
- Another products

Points:

Using `pch=""`. It has values between 1 till 25 or a single character.

```
par(mfrow=c(4,2), mar=c(3,3,1,0)+.5, mgp=c(1.6,.6,0))  # set up the graphics
x1<-rnorm(10)
plot(x1, type = "p", pch = 0)
plot(x1, type = "p", pch = 10)
plot(x1, type = "p", pch = 25)
plot(x1, type = "p", pch = "a")
plot(x1, type = "p", pch = "*")
plot(x1[1:26], type = "p", pch = 0:25)
plot(x1[1:26], type = "p", pch = letters)
```



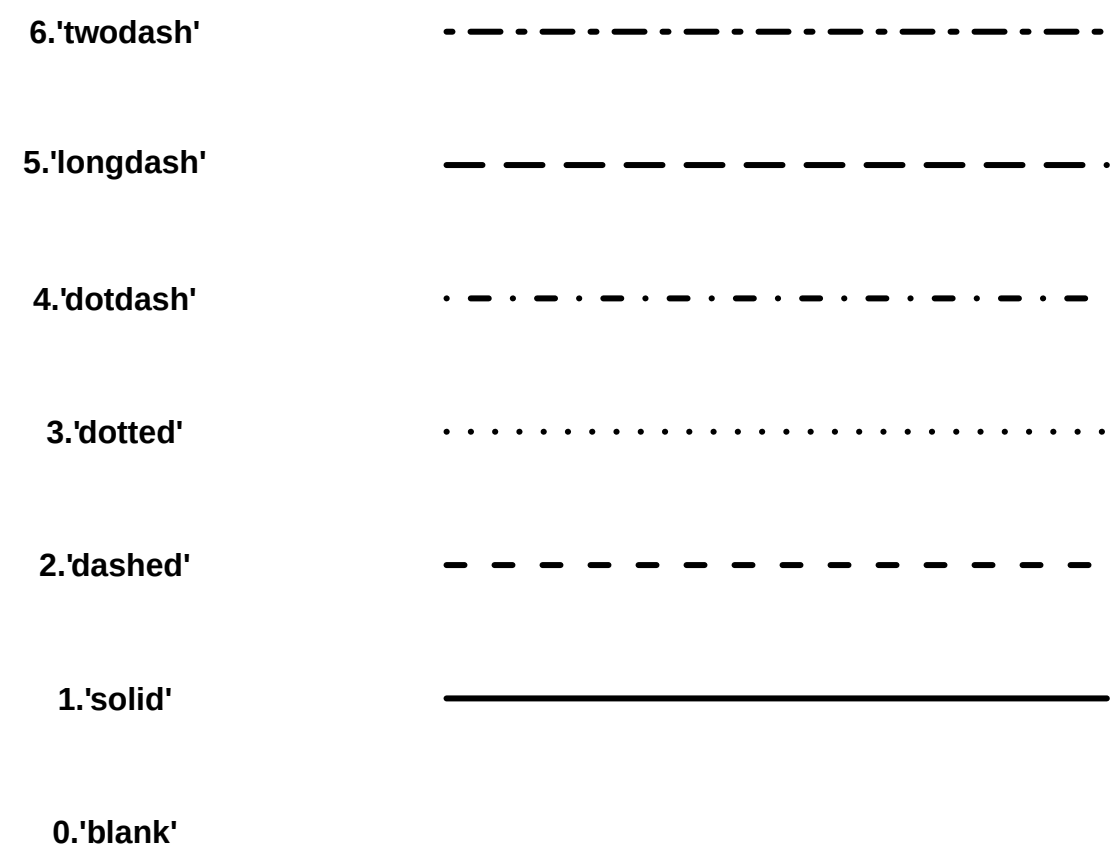
Lines:

====Line types in R : `lty`====

`lty` for changing the lines type and `lwd` to change line width. The argument is a string ("blank", "solid", "dashed", "dotted", "dotdash", "longdash", or "twodash") or an integer (0=blank, 1=solid (default), 2=dashed, 3=dotted, 4=dotdash, 5=longdash, 6=twodash)

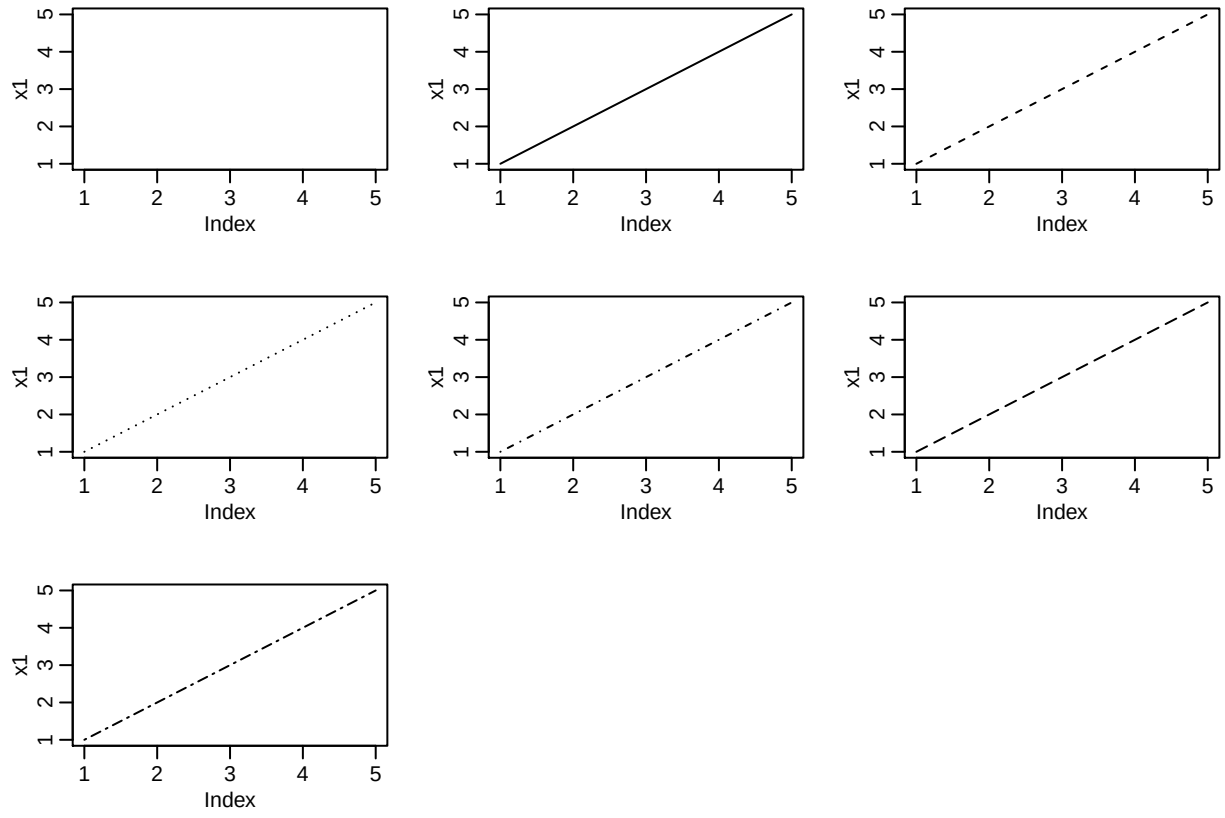
```
#Line types
#++++++++++++++++++++++++++++++++++++++++++++
generateRLineTypes<-function(){
  oldPar<-par()
  par(font=2, mar=c(0,0,0,0))
  plot(1, pch="", ylim=c(0,6), xlim=c(0,0.7),  axes=FALSE,xlab="", ylab="")
  for(i in 0:6) lines(c(0.3,0.7), c(i,i), lty=i, lwd=3)
  text(rep(0.1,6), 0:6, labels=c("0.'blank'", "1.'solid'", "2.'dashed'", "3.'dotted'",
                                 "4.'dotdash'", "5.'longdash'", "6.'twodash'"))
```

```
  par(mar=oldPar$mar,font=oldPar$font )
}
generateRLineTypes()
```



Example:

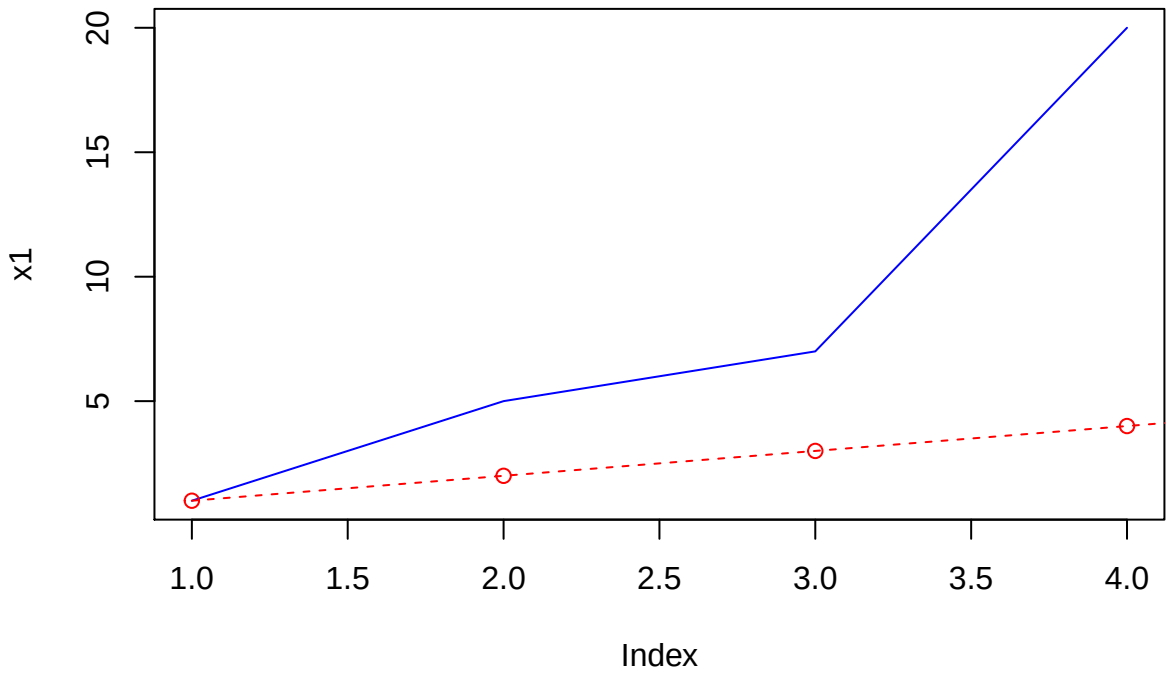
```
par(mfrow=c(3,3), mar=c(3,3,1,0)+.5, mgp=c(1.6,.6,0))  # set up the graphics
x1<-1:5
plot(x1, type = "l", lty = "blank")
plot(x1, type = "l", lty = "solid")
plot(x1, type = "l", lty = "dashed")
plot(x1, type = "l", lty = "dotted")
plot(x1, type = "l", lty = "dotdash")
plot(x1, type = "l", lty = "longdash")
plot(x1, type = "l", lty = "twodash")
```



====An additional `lines()` on a graph====

Example:

```
x1<-c(1,5,7,20)
x2<-1:10
plot(x1, type = "l", lty = "solid",col="blue")
lines(x2, type = "o", lty = "dashed", col = "red")
```



====Straight `abline()`====

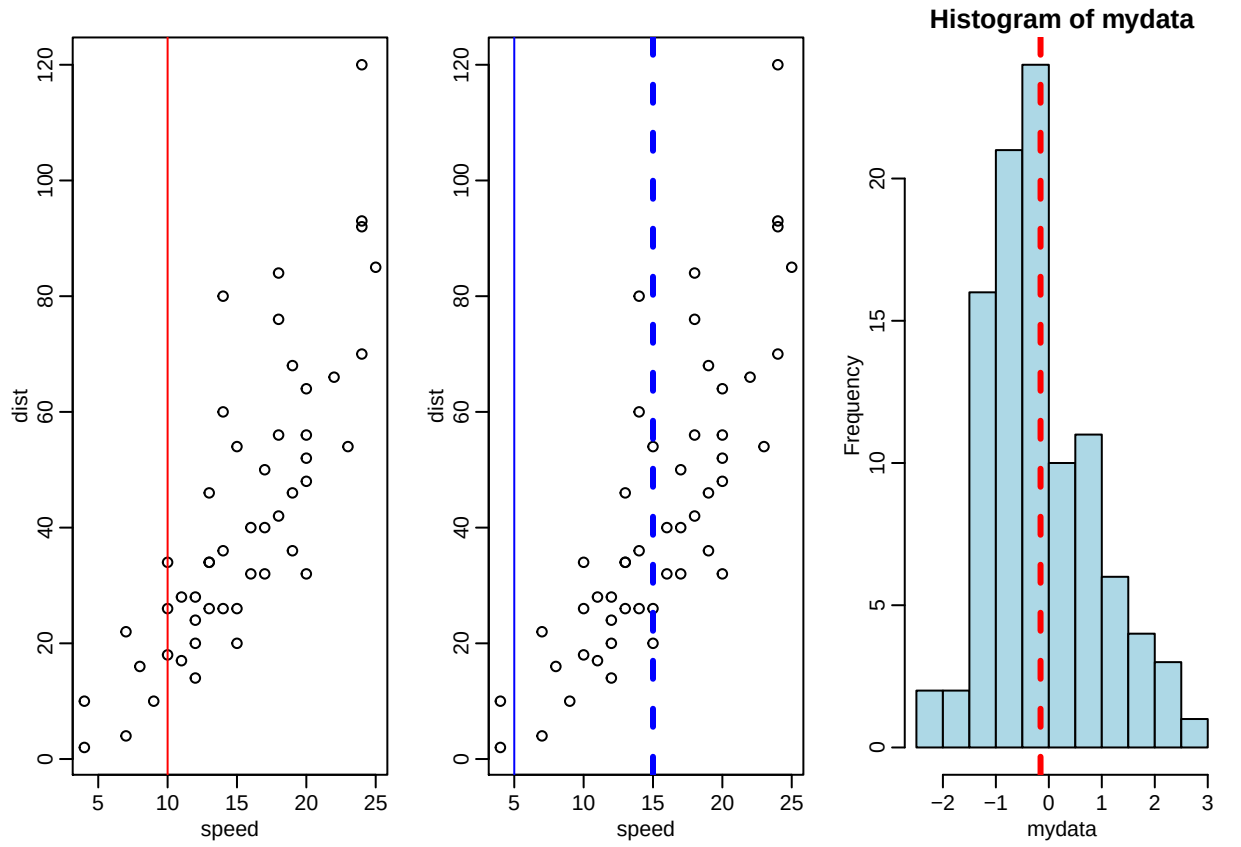
`abline()` can be used to add vertical, horizontal or regression lines.

Formula:

abline(a=NULL, b=NULL, h=NULL, v=NULL, ...)

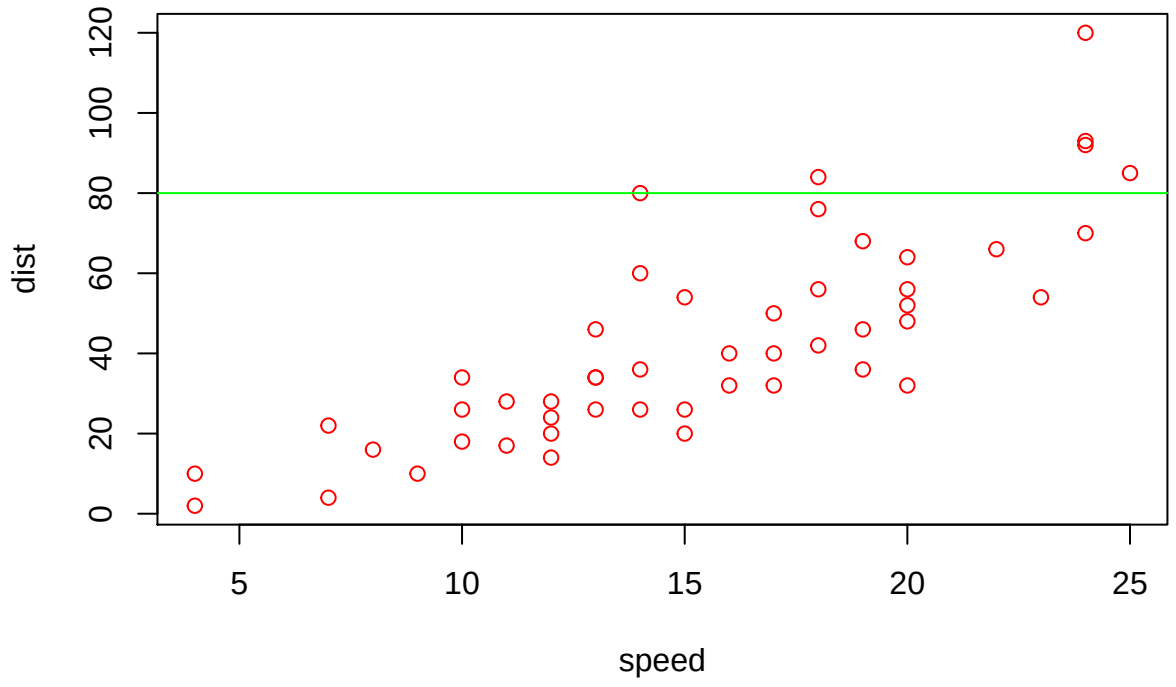
Vertical line:

```
par(mfrow=c(1,3), mar=c(3,3,1,0)+.5, mgp=c(1.6,.6,0))  # set up the graphics

#Example-1: Add one line
plot(cars)
abline(v=10, col="red")

#Example-2: add 2 lines
# change line colors, sizes and types
plot(cars)
abline(v=c(5,15), col=c("blue", "blue"), lty=c(1,2), lwd=c(1, 3))

#Example-3:
set.seed(1234); mydata<-rnorm(100)
hist(mydata, col="lightblue")
abline(v = mean(mydata), col="red", lwd=3, lty=2)
```



Horizontal line:

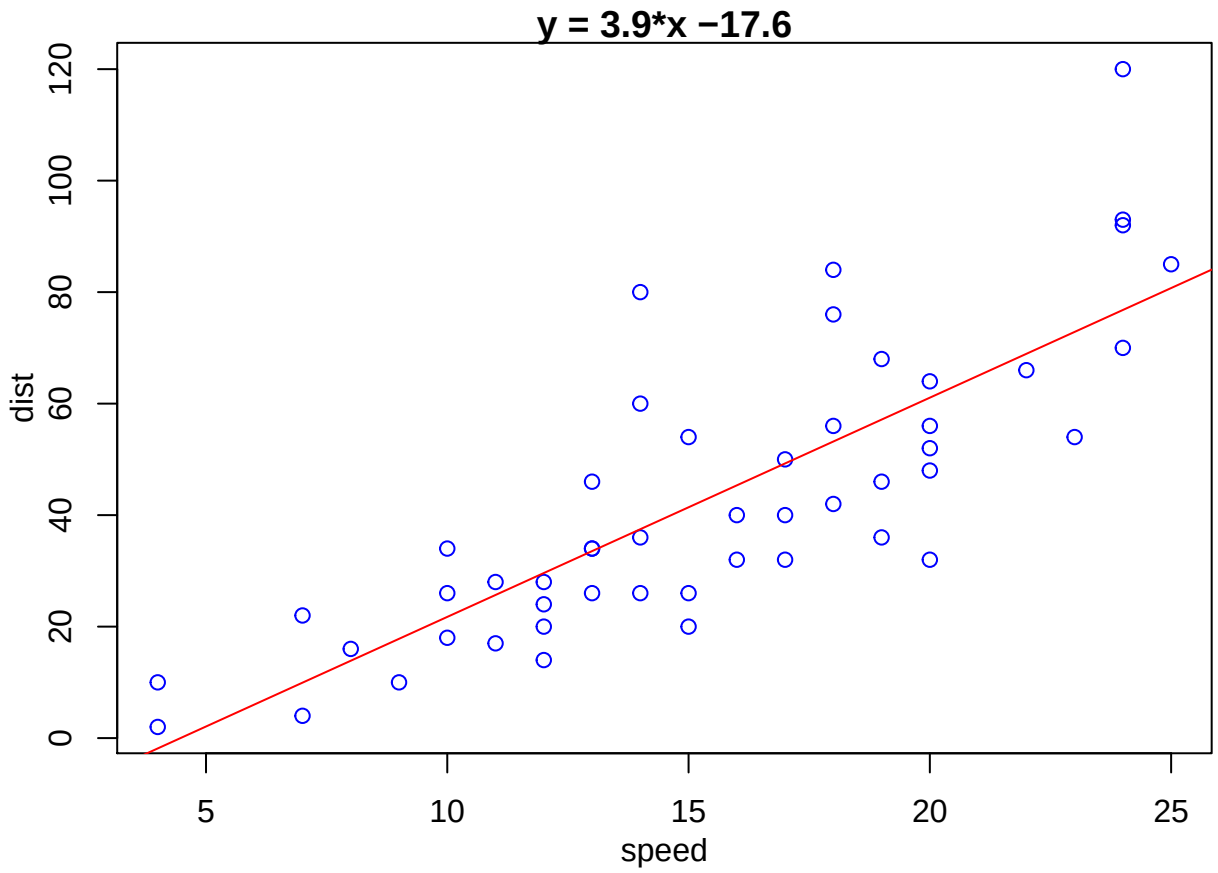
```
plot(cars,col="red")
abline(h=80, col="green")
```



Regression line:

`lm()` function to fit linear model.

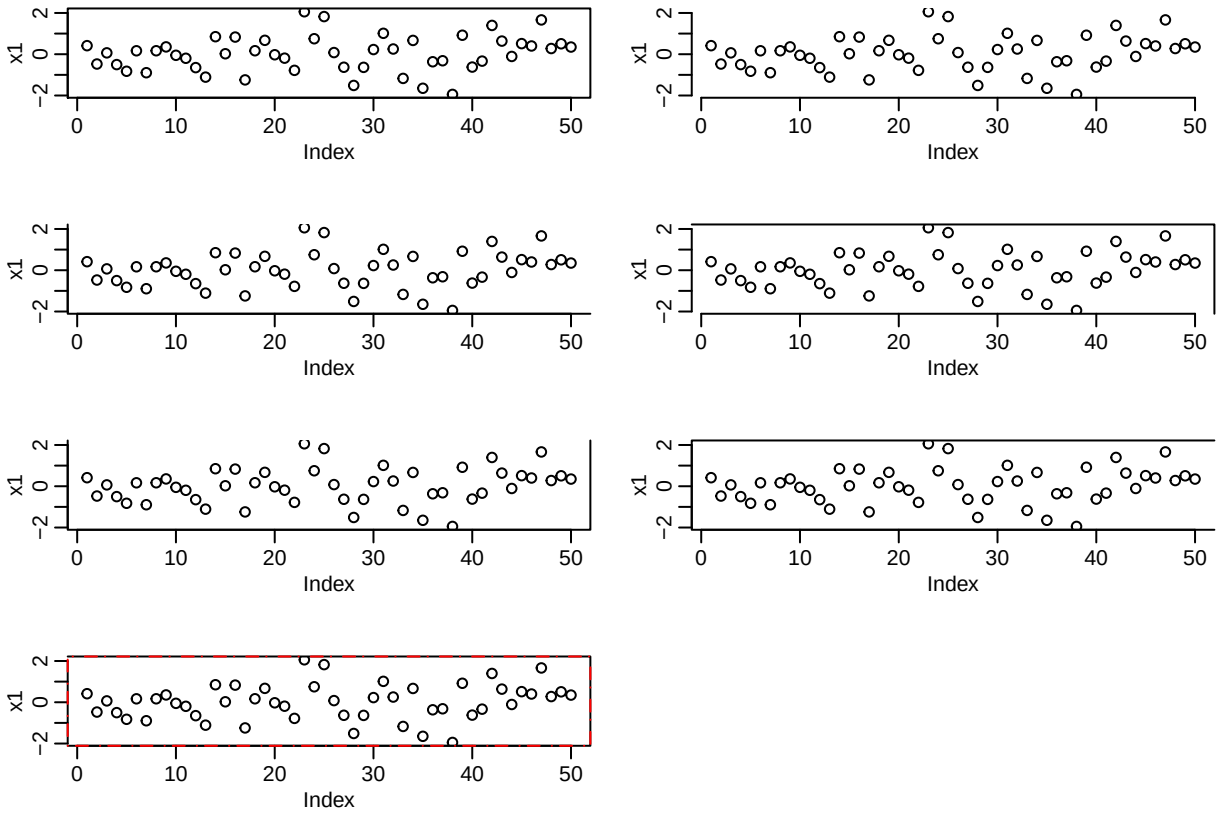
```
par(mgp=c(2,1,0), mar=c(3,3,1,1))
# Fit regression line
require(stats)
reg<-lm(dist ~ speed, data = cars)
coeff=coefficients(reg)
# equation of the line :
eq = paste0("y = ", round(coeff[2],1), "*x ", round(coeff[1],1))
# plot
plot(cars, main=eq,col="blue")
abline(reg, col="red")
```



Boxes:

`bty` specifies the box type.

```
par(mfrow=c(4,2), mar=c(3,3,1,0)+.5, mgp=c(1.6,.6,0))  # set up the graphics
x1<-rnorm(50)
plot(x1, bty = "o") # the default
plot(x1, bty = "n") # no box
plot(x1, bty = "l")
plot(x1, bty = "7")
plot(x1, bty = "u")
plot(x1, bty = "c")
plot(x1, bty = "]")
box(lty = '1375', col = 'red')
```

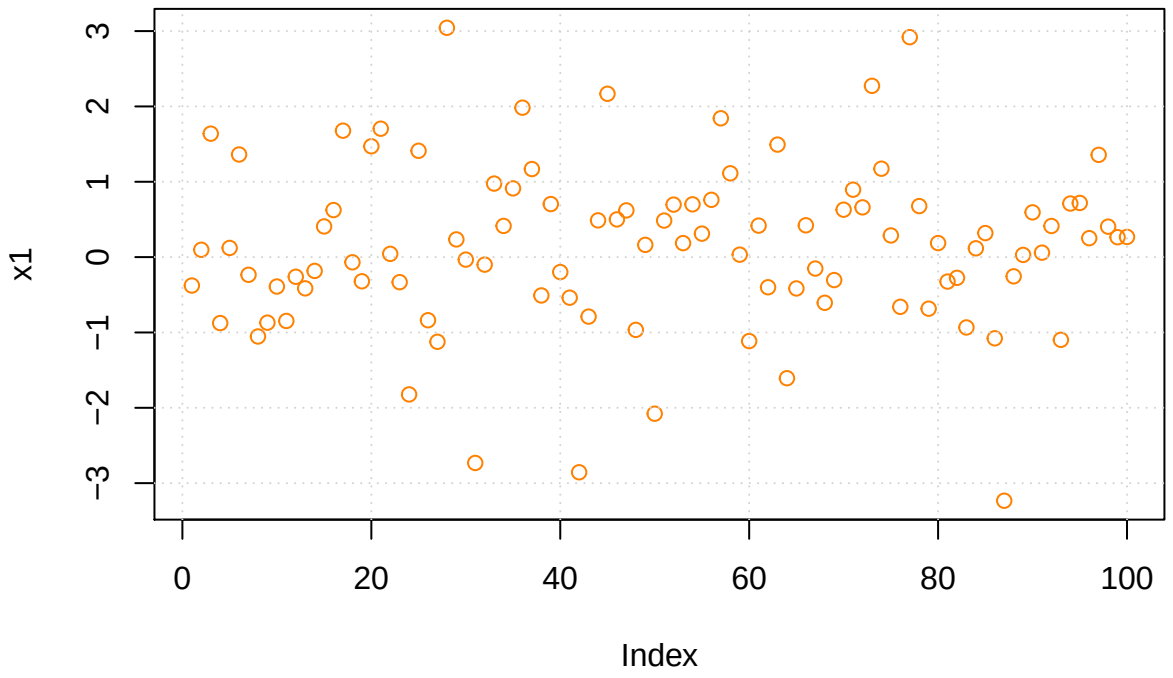


Grid:

`grid()` adds a grid to the current graph.

Example:

```
x1<-rnorm(100)
plot(x1, col="#FF8100")
grid()
```

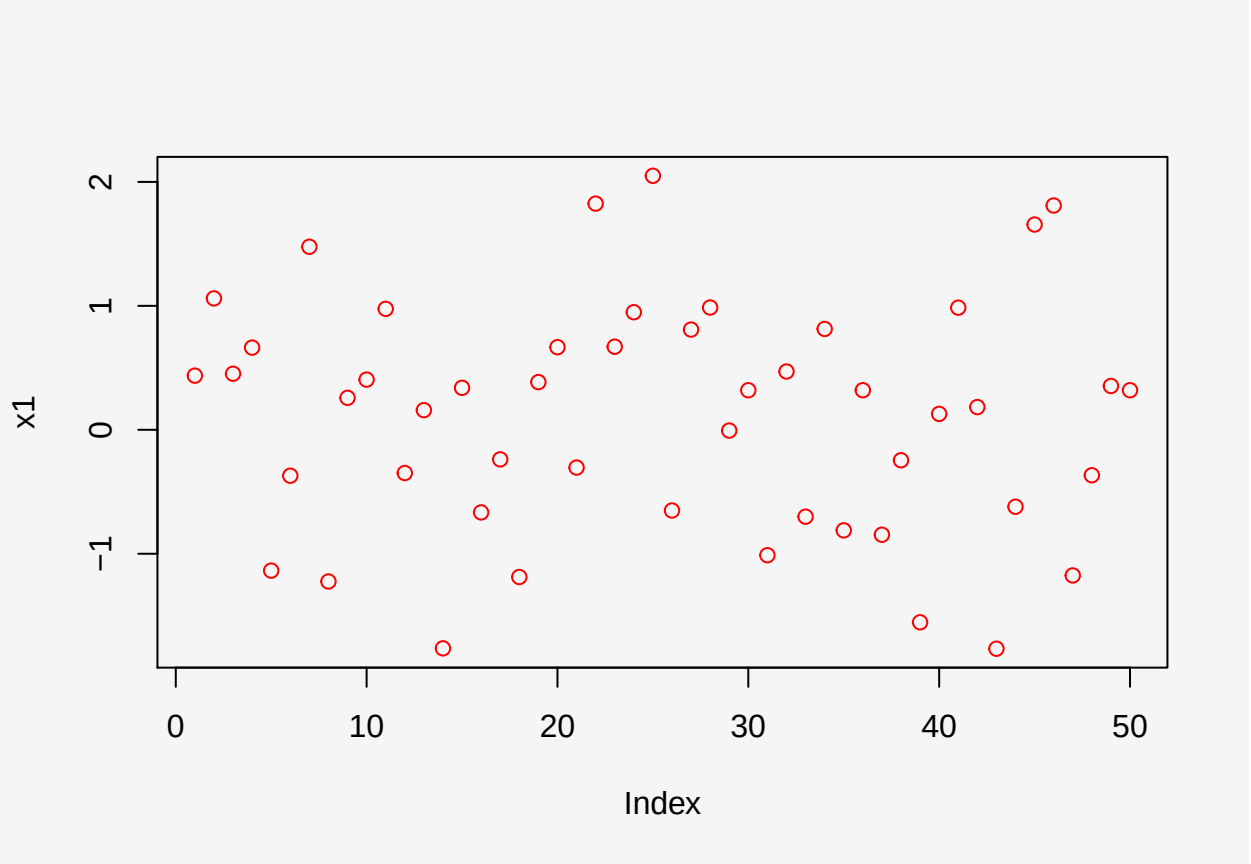


Background:

Change the background color with `par(bg=)`.

Example:

```
x1<-rnorm(50)
par(bg="whitesmoke")
plot(x1, bty ="o",col="red") # the default
```

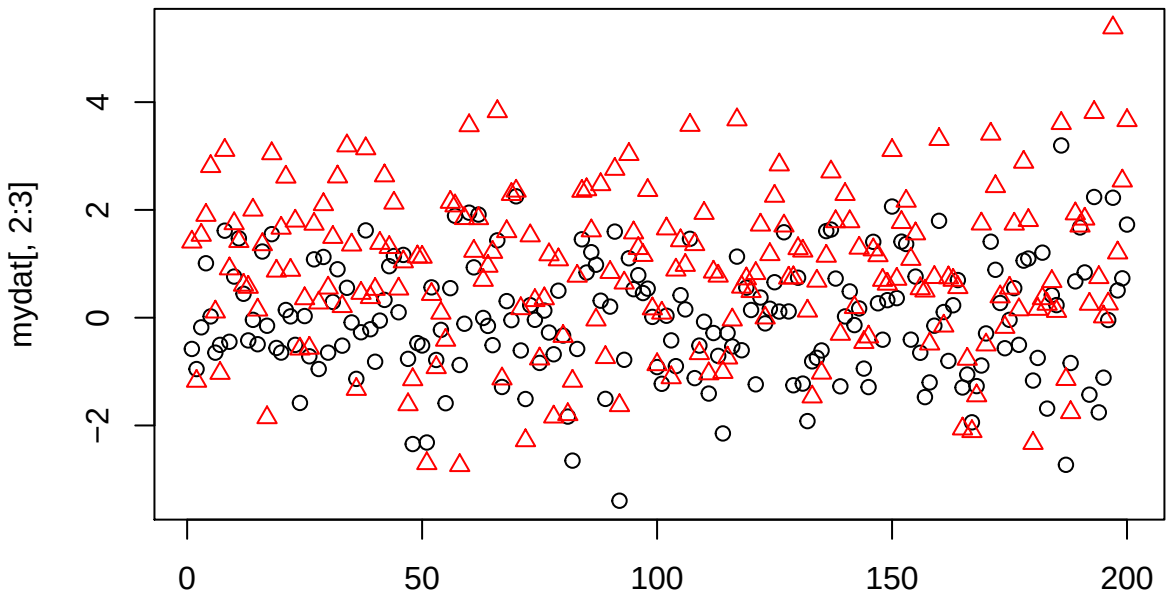


Overlaying plots:

`matplot()`

Example:

```
N <- 200
x1 <- rnorm(N)
x2 <- rnorm(N) + x1 + 1
y <- 1 + x1 + x2 + rnorm(N)
mydat <- data.frame(y,x1,x2)
matplot(mydat[,2:3], pch = 1:2)
```

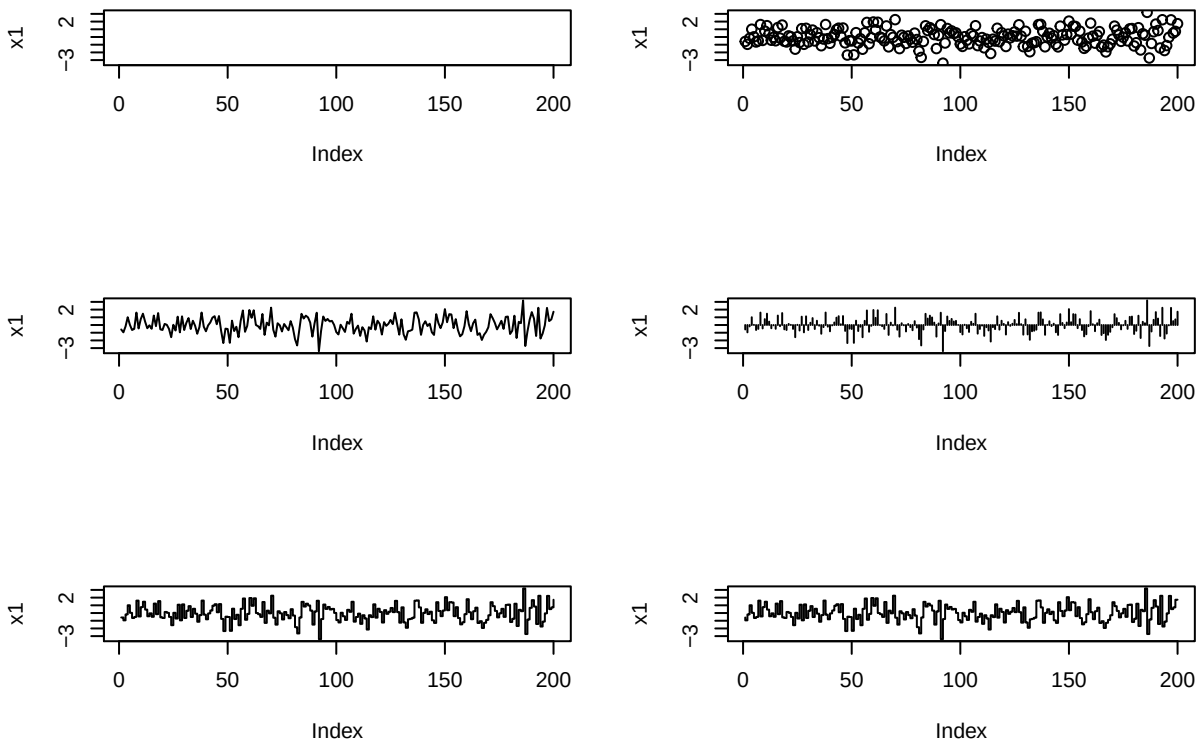


Multiple plots:

- `par()` for display multiple figures
- `mfrow = c(3,2)` : 3 row x 2 column
- `mfcol = c(3,2)` : same with mfrow but another positions

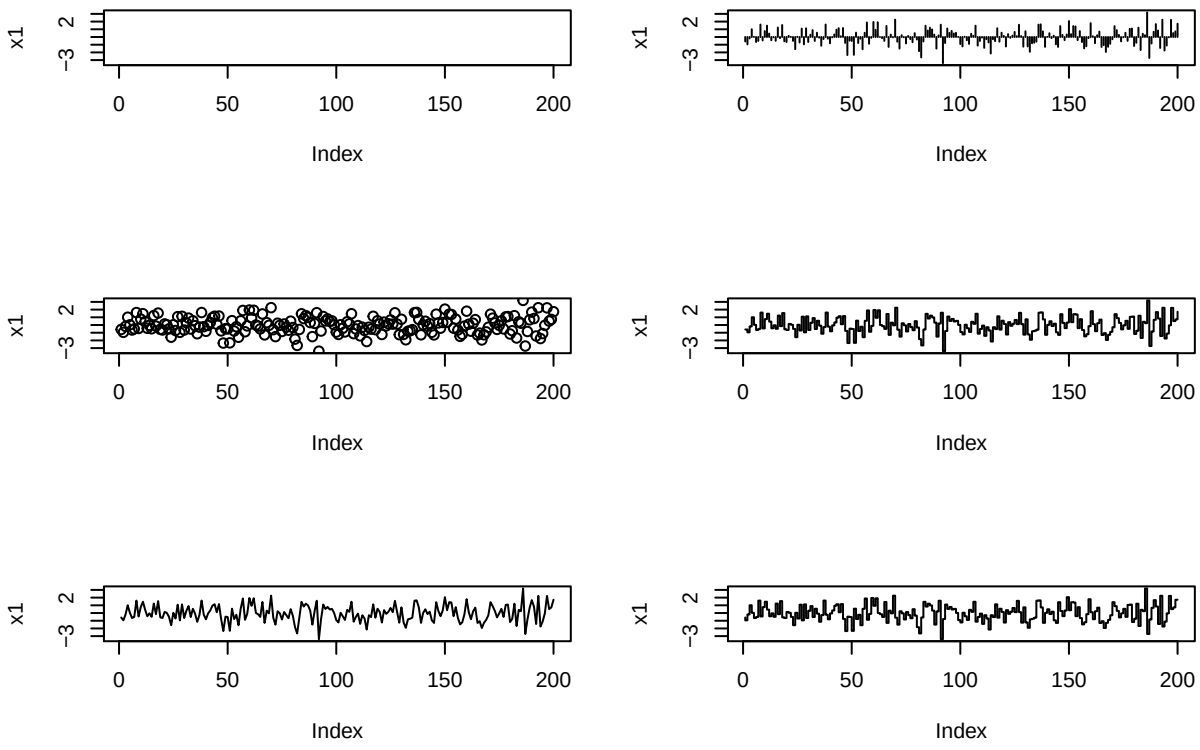
Example-1:

```
par(mfrow = c(3,2))
plot(x1, type = "n")
plot(x1, type = "p")
plot(x1, type = "l")
plot(x1, type = "h")
plot(x1, type = "s")
plot(x1, type = "S")
```



Example-2:

```
par(mfcol = c(3,2))
plot(x1, type = "n")
plot(x1, type = "p")
plot(x1, type = "l")
plot(x1, type = "h")
plot(x1, type = "s")
plot(x1, type = "S")
```

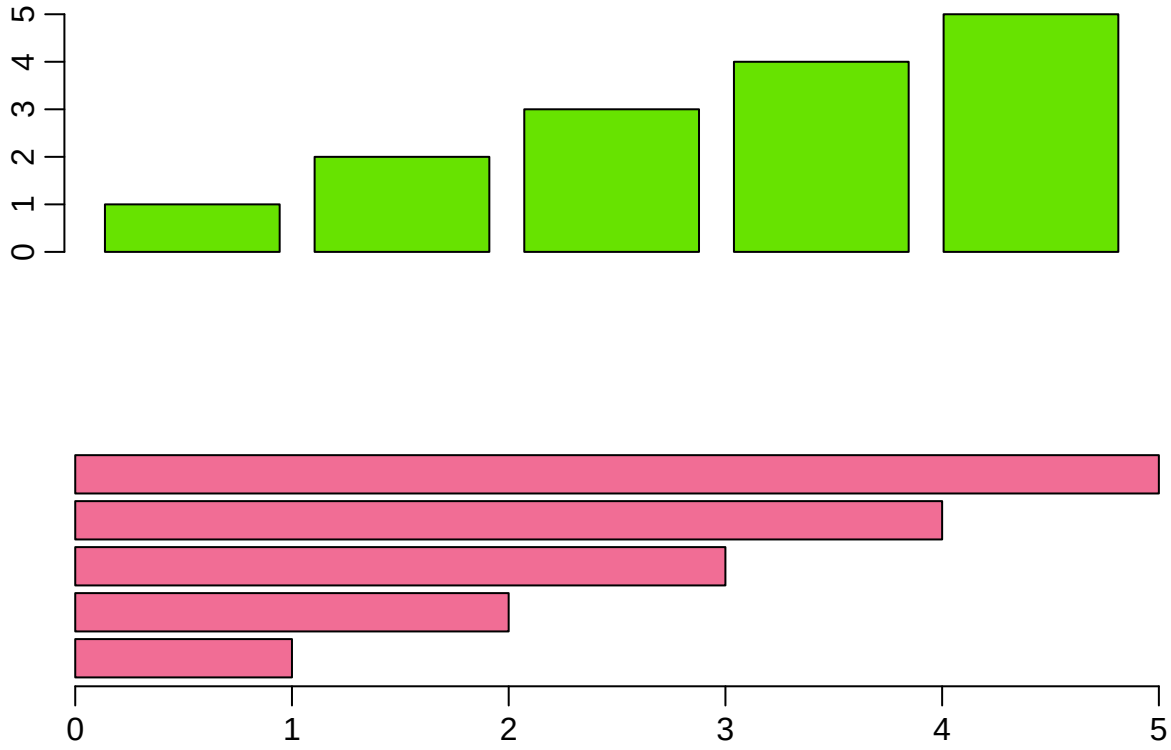

x1
−3
2
0
50
100
150
200
Index

Chapter 4

Default Bar plots in R

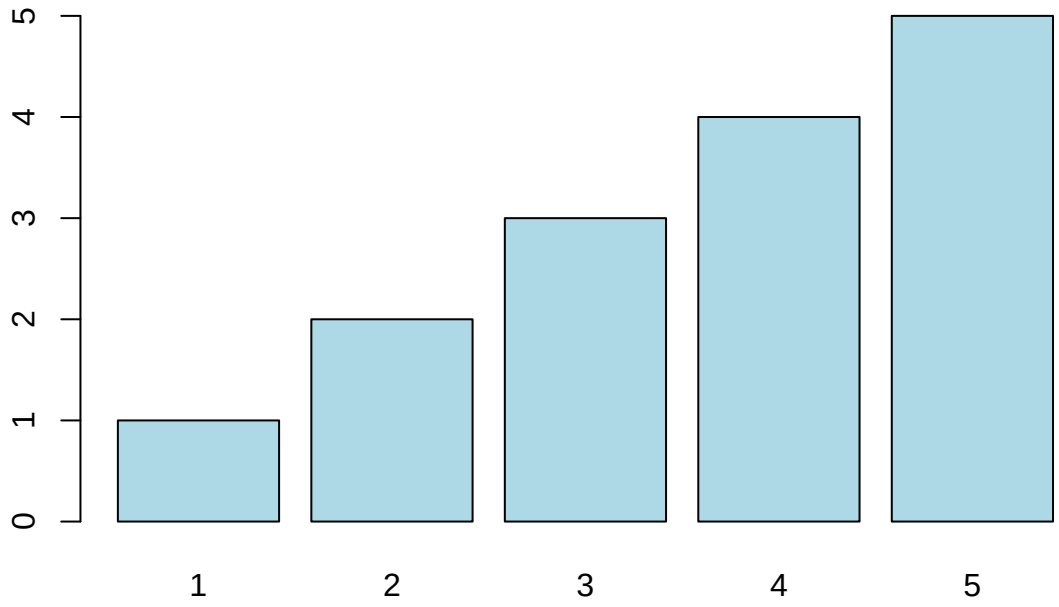
Basic bar plots:

```
par(mfrow=c(2,1), mar=c(3,3,1,0)+.5, mgp=c(1.6,.6,0))
x<-c(1:5)
barplot(x,col="#67E300")
barplot(x,horiz=TRUE,col="#F16D95")
```



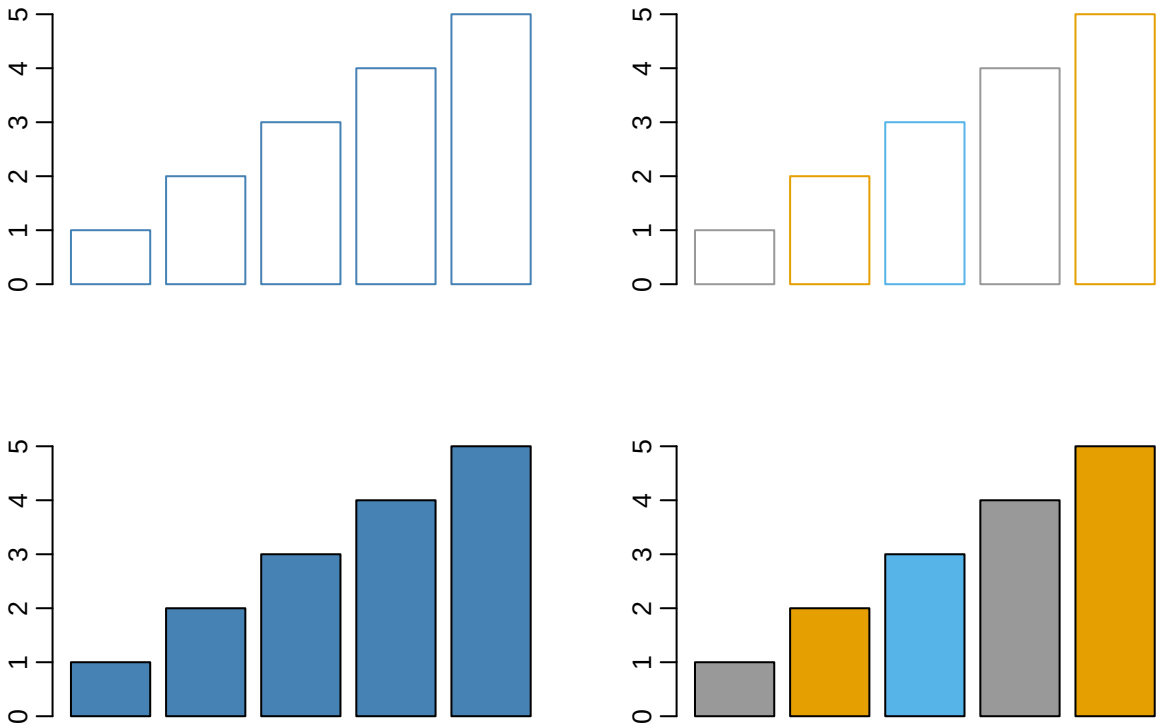
Change group names:

```
barplot(x, names.arg = c("1", "2", "3","4","5"), col="lightblue")
```



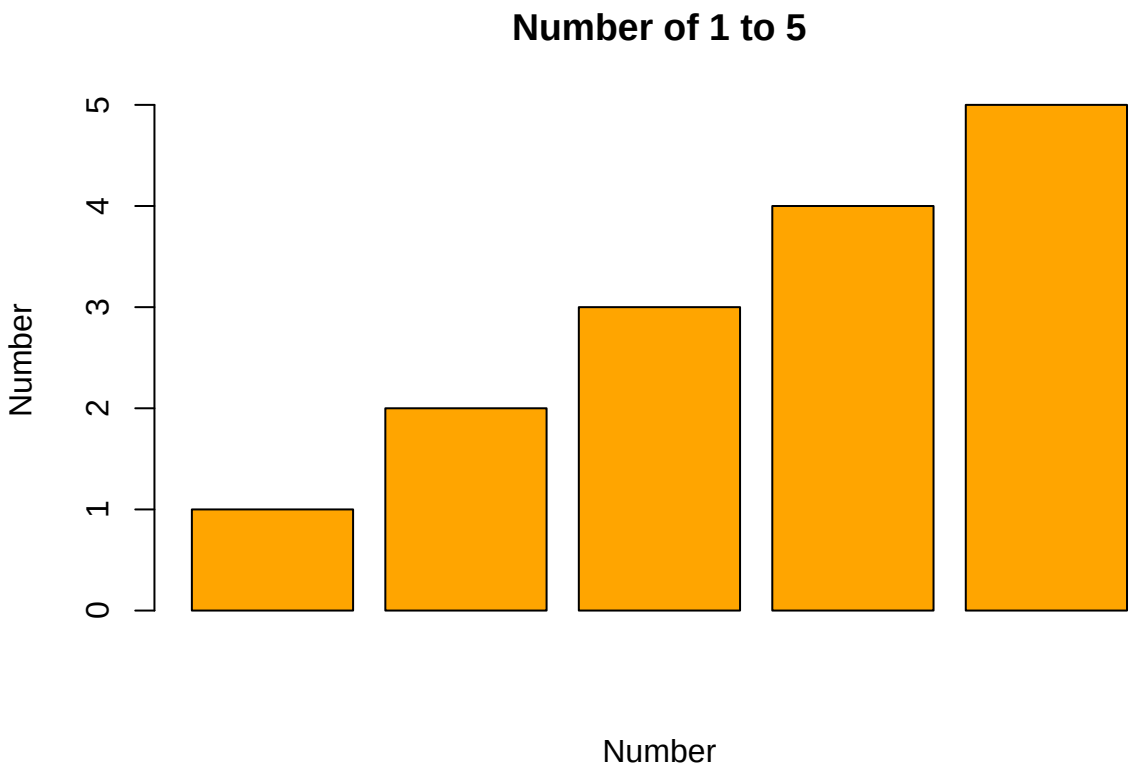
Change color:

```
par(mfrow=c(2,2), mar=c(3,3,1,0)+.5, mgp=c(1.6,.6,0))
x<-c(1:5)
# Change border and fill color using one single color
barplot(x, col = "white", border = "steelblue")
# Change the color of border.
#  Use different colors for each group
barplot(x, col = "white",
        border = c("#999999", "#E69F00", "#56B4E9"))
# Change fill color : single color
barplot(x, col = "steelblue")
# Change fill color: multiple colors
barplot(x, col = c("#999999", "#E69F00", "#56B4E9"))
```



Change main title and axis labels:

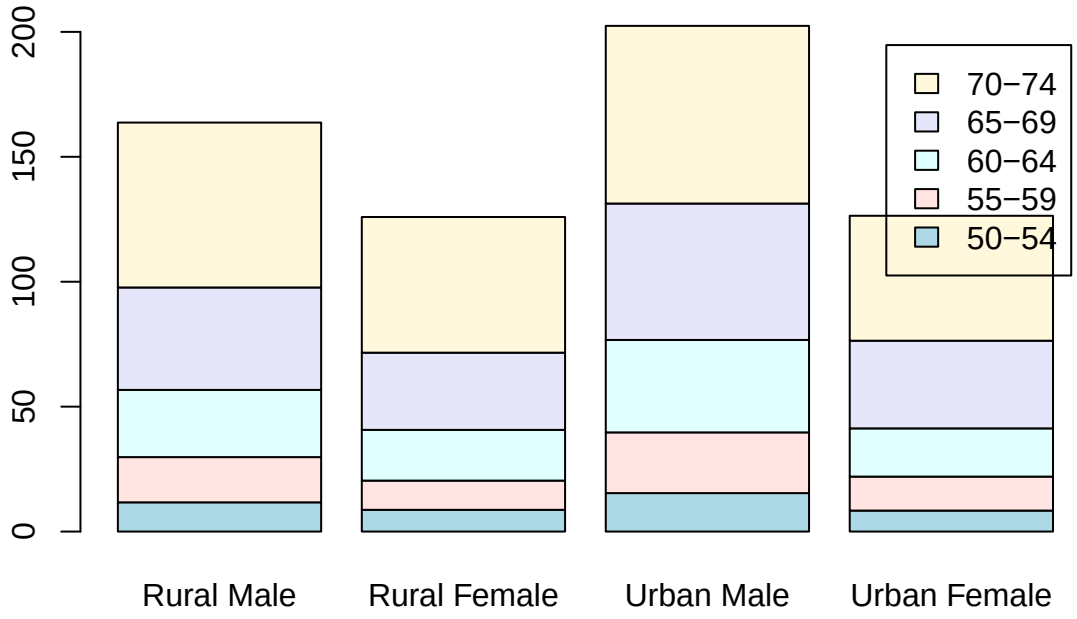
```
# Change axis titles
# Change color (col = "orange") and remove frame
x<-c(1:5)
barplot(x, main = "Number of 1 to 5",
        xlab = "Number", ylab = "Number", col="orange")
```



Stacked bar plots:

Example-1:

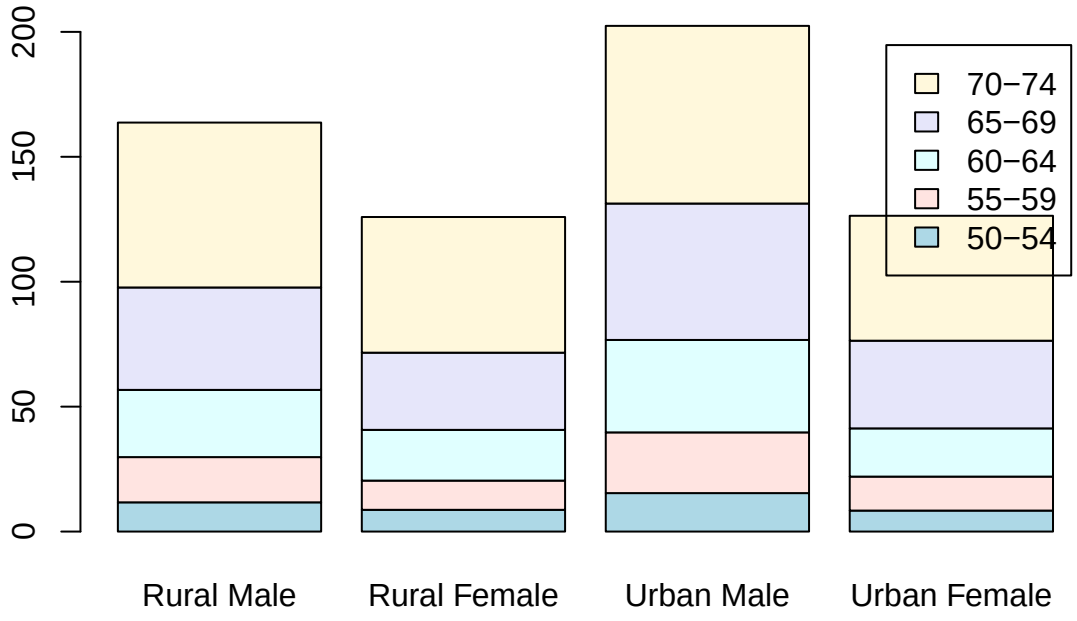
```
barplot(VADeaths,
        col = c("lightblue", "mistyrose", "lightcyan",
                "lavender", "cornsilk"),
       legend = rownames(VADeaths))
```



Stacked bar plots:

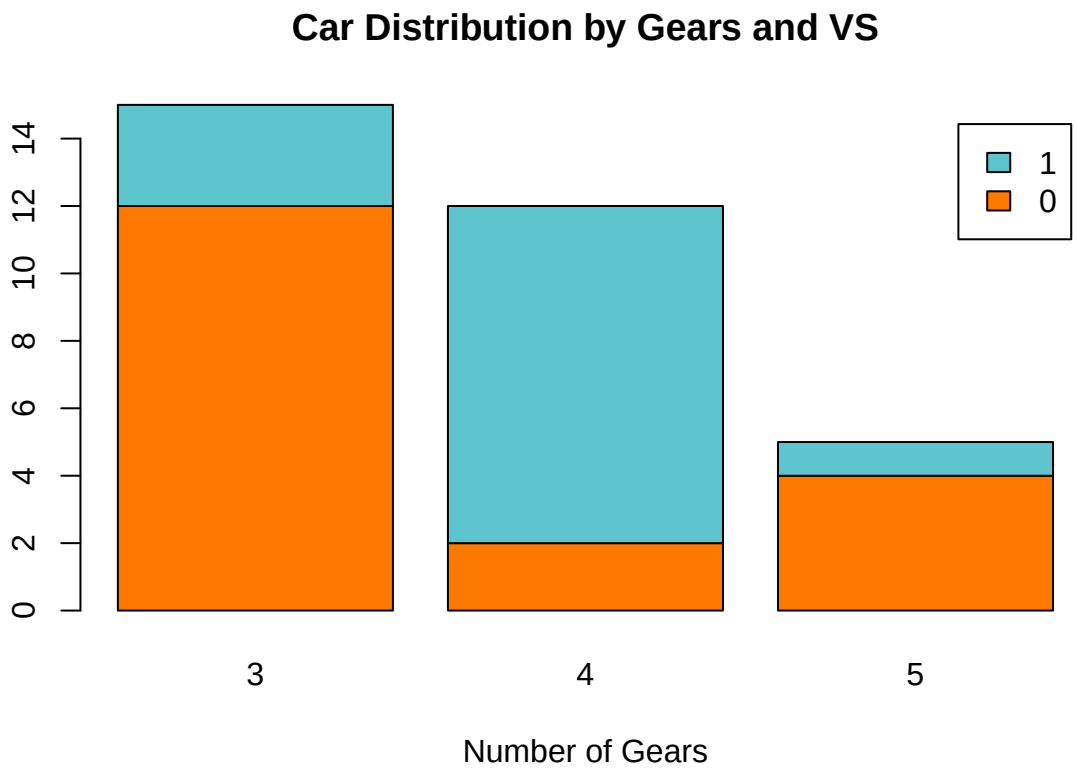
Example-1:

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



Example-2:

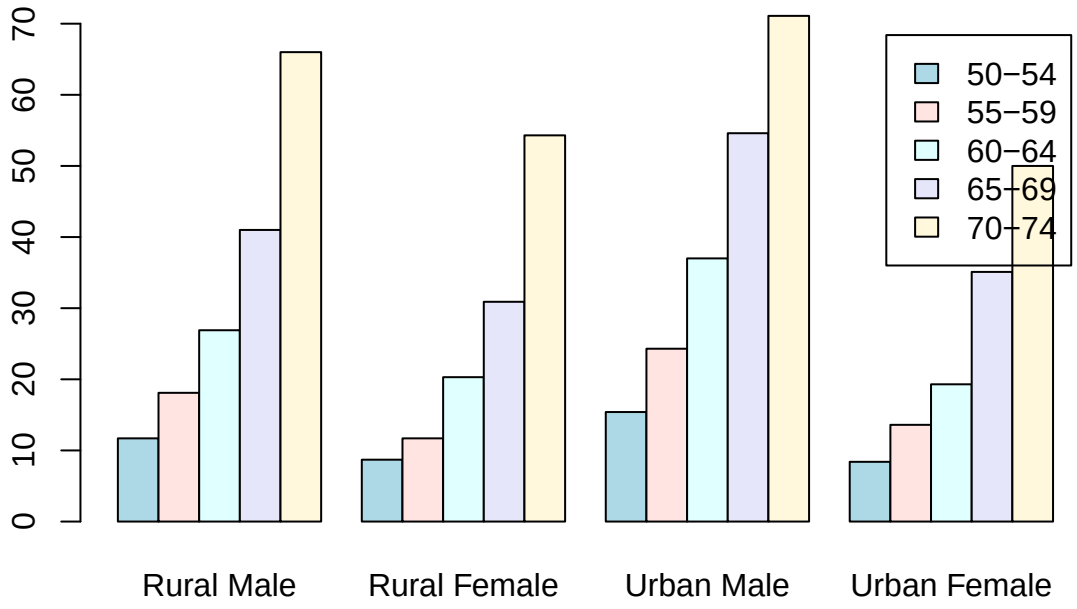
```
# Stacked Bar Plot with Colors and Legend
counts <- table(mtcars$vs, mtcars$gear)
barplot(counts, main="Car Distribution by Gears and VS",
  xlab="Number of Gears", col=c("#FF7800","#5EC4CD"),
  legend = rownames(counts))
```



Grouped bar plots:

Example-1:

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

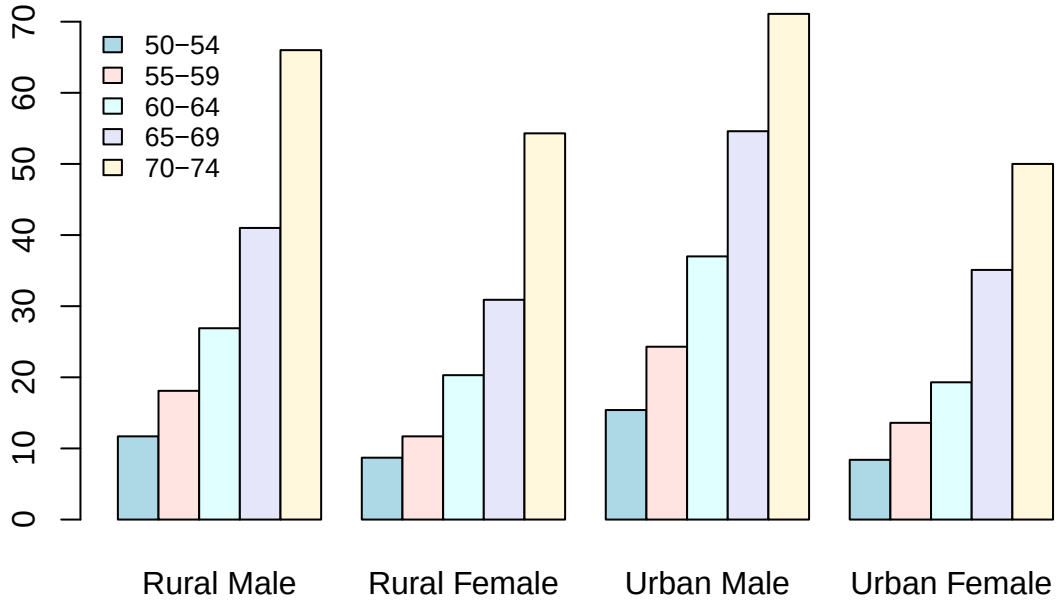


Example-2:

- box.lty = 0: Remove the box around the legend

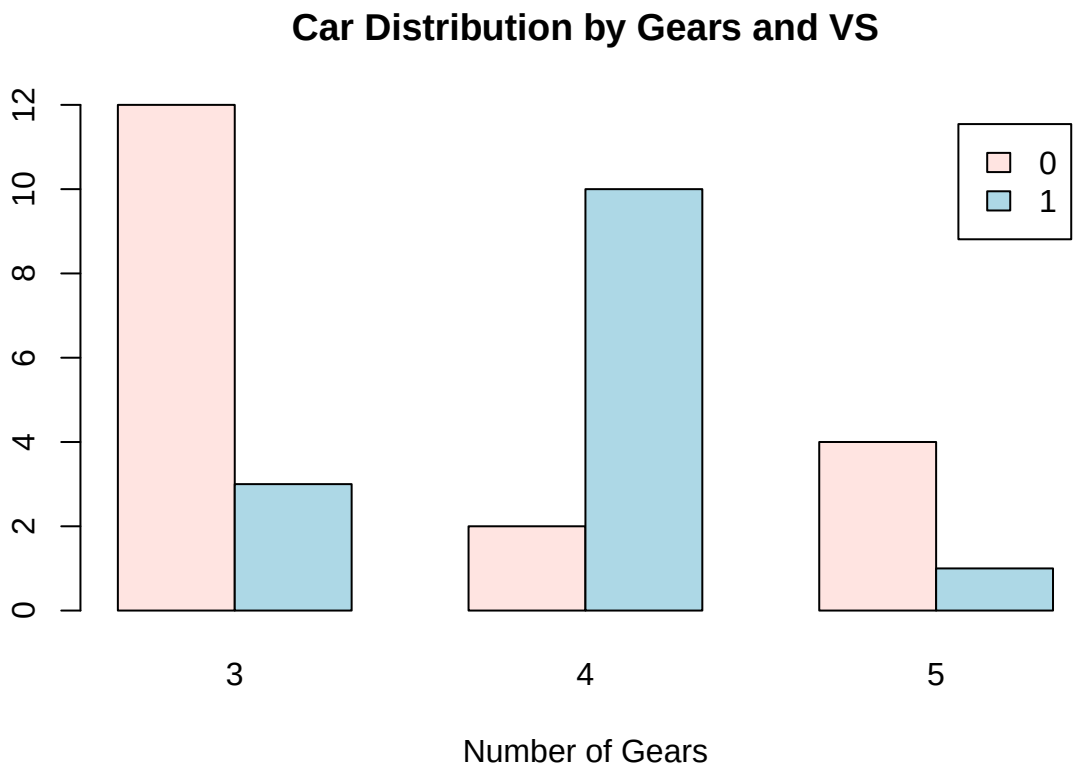
- cex = 0.8: legend text size

```
# Define a set of colors
my_colors <- c("lightblue", "mistyrose", "lightcyan",
               "lavender", "cornsilk")
# Bar plot
barplot(VADeaths, col = my_colors, beside = TRUE)
# Add legend
legend("topleft", legend = rownames(VADeaths),
       fill = my_colors, box.lty = 0, cex = 0.8)
```



Example-3:

```
# Grouped Bar Plot
counts <- table(mtcars$vs, mtcars$gear)
barplot(counts, main="Car Distribution by Gears and VS",
  xlab="Number of Gears", col=c("mistyrose","lightblue"),
  legend = rownames(counts), beside=TRUE)
```



Note: If you need more formatting, see the **3-Standard R graphs formatting**.

Chapter 5

Default Histogram and Density Plots in R

A Histogram is NOT a Bar Chart:

`Preference:https://www.edrawsoft.com/histogram-vs-bar-chart.php`

- Histograms VS. Bar Charts
- A Histogram is NOT a Bar Chart

Simple Histograms:

- x: a numeric vector
- breaks: breakpoints between histogram cells.

```
# Simple Histogram
par(mfrow=c(2,1), mar=c(3,3,1,0)+.5, mgp=c(1.6,.6,0))
hist(mtcars$mpg,col="mistyrose")
hist(mtcars$mpg, breaks =30,col="pink")
```

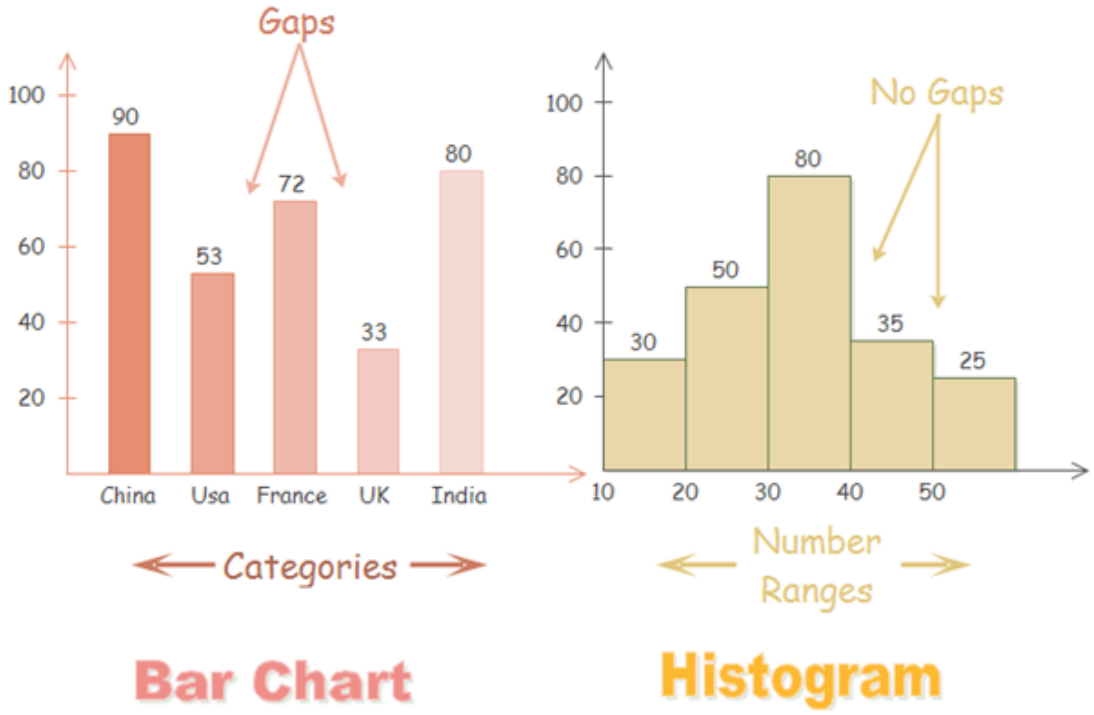
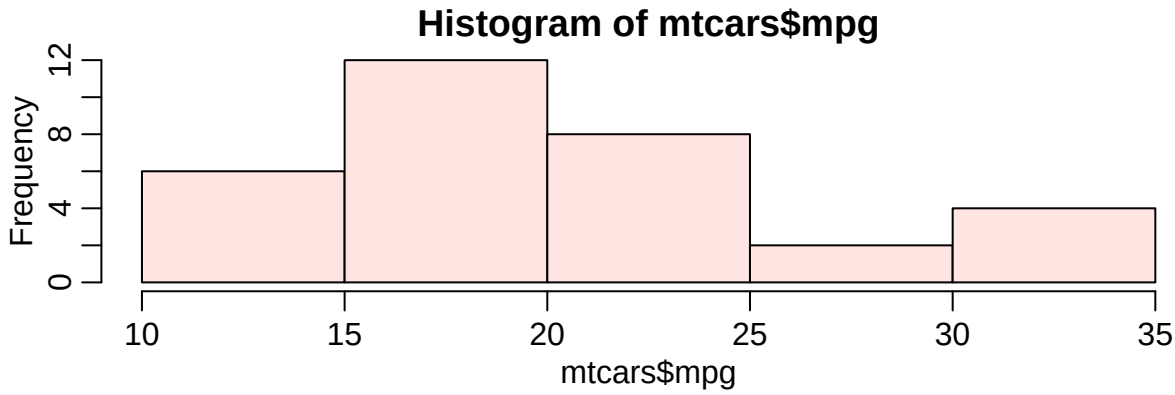
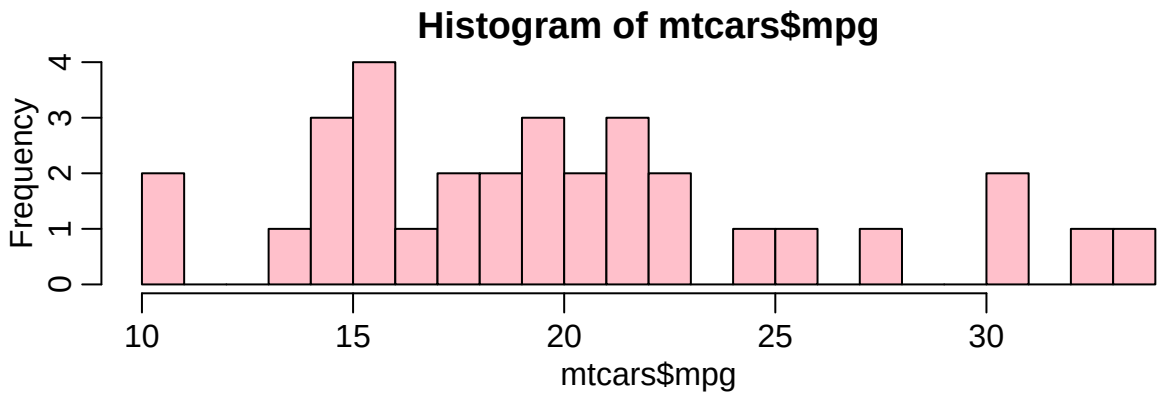

Gaps
100
80
60
40
20
90
53
72
33
80
China
Usa
France
UK
India
Categories
Bar Chart
No Gaps
100
80
60
40
20
30
50
80
35
25
10
20
30
40
50
Number
Ranges
Histogram

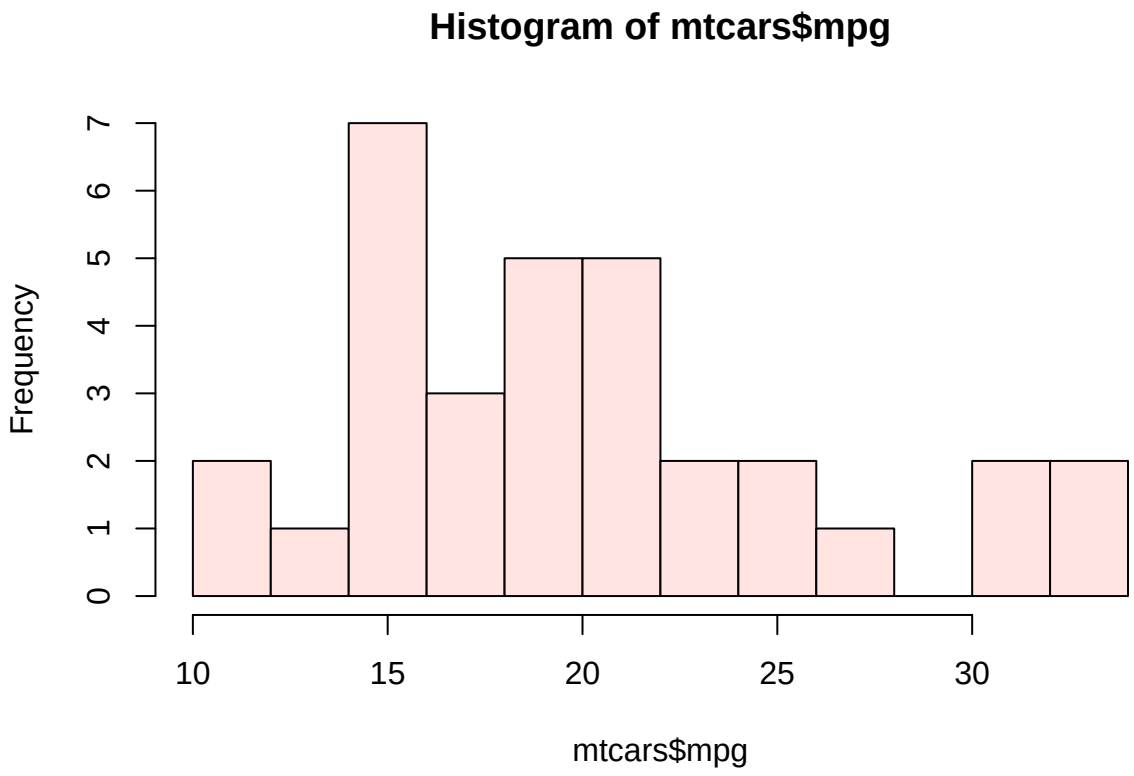
Figure 5.1:


Histogram of mtcars$mpg
Frequency
0
4
8
12
10
15
20
25
30
35
mtcars$mpg


Histogram of mtcars$mpg
Frequency
0
1
2
3
4
10
15
20
25
30
mtcars$mpg

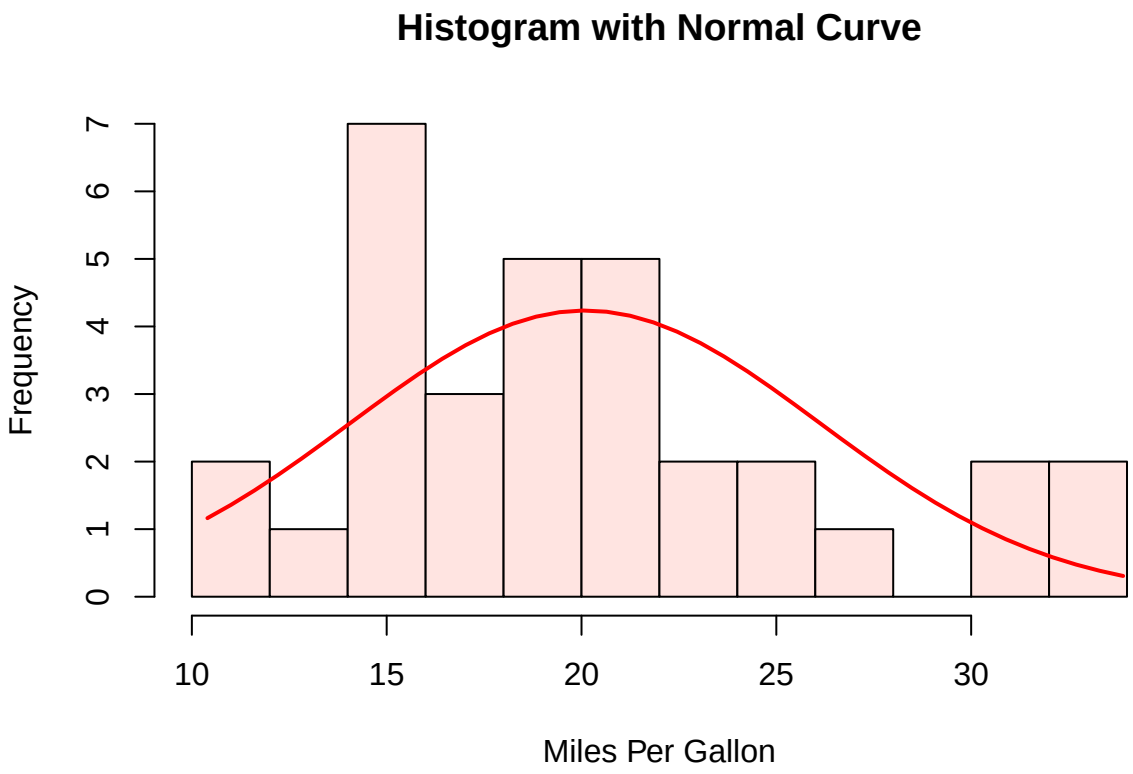
Colored Histogram:

```
hist(mtcars$mpg, breaks=12, col="mistyrose")
```



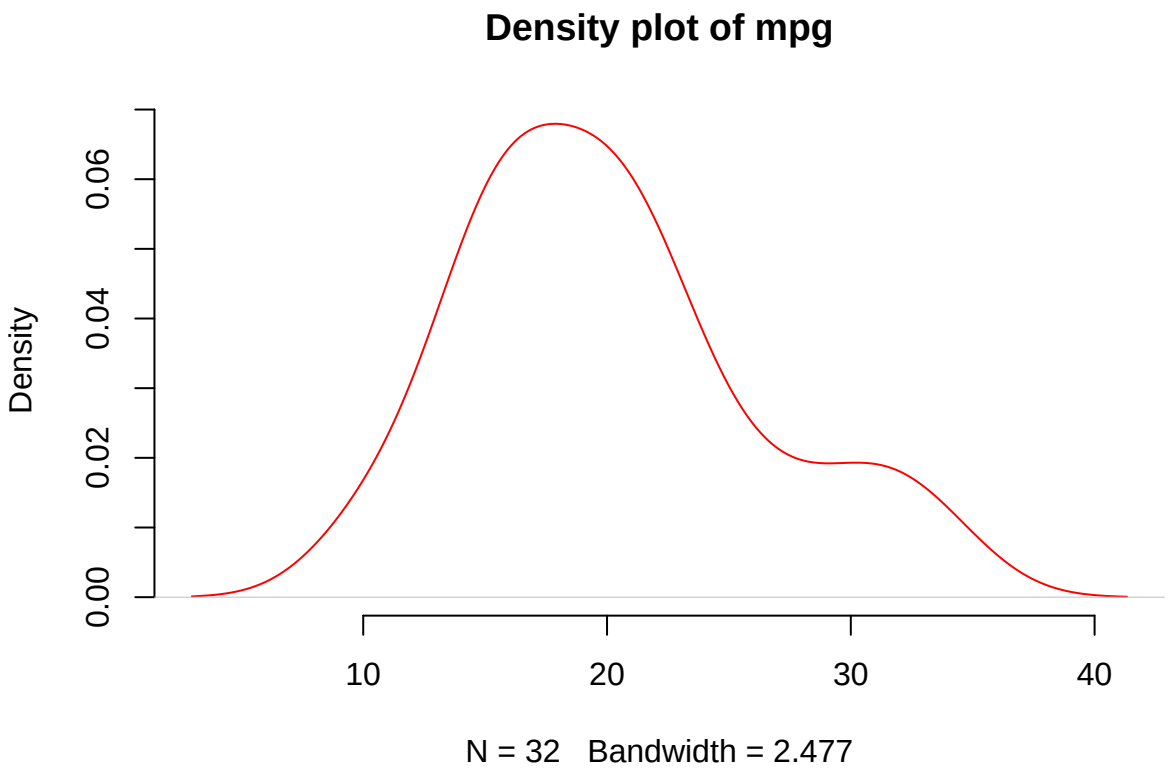
Add a Normal Curve:

```
x <- mtcars$mpg
h<-hist(x, breaks=10, col="mistyrose", xlab="Miles Per Gallon",
   main="Histogram with Normal Curve")
xfit<-seq(min(x),max(x),length=40)
yfit<-dnorm(xfit,mean=mean(x),sd=sd(x))
yfit <- yfit*diff(h$mids[1:2])*length(x)
lines(xfit, yfit, col="red", lwd=2)
```



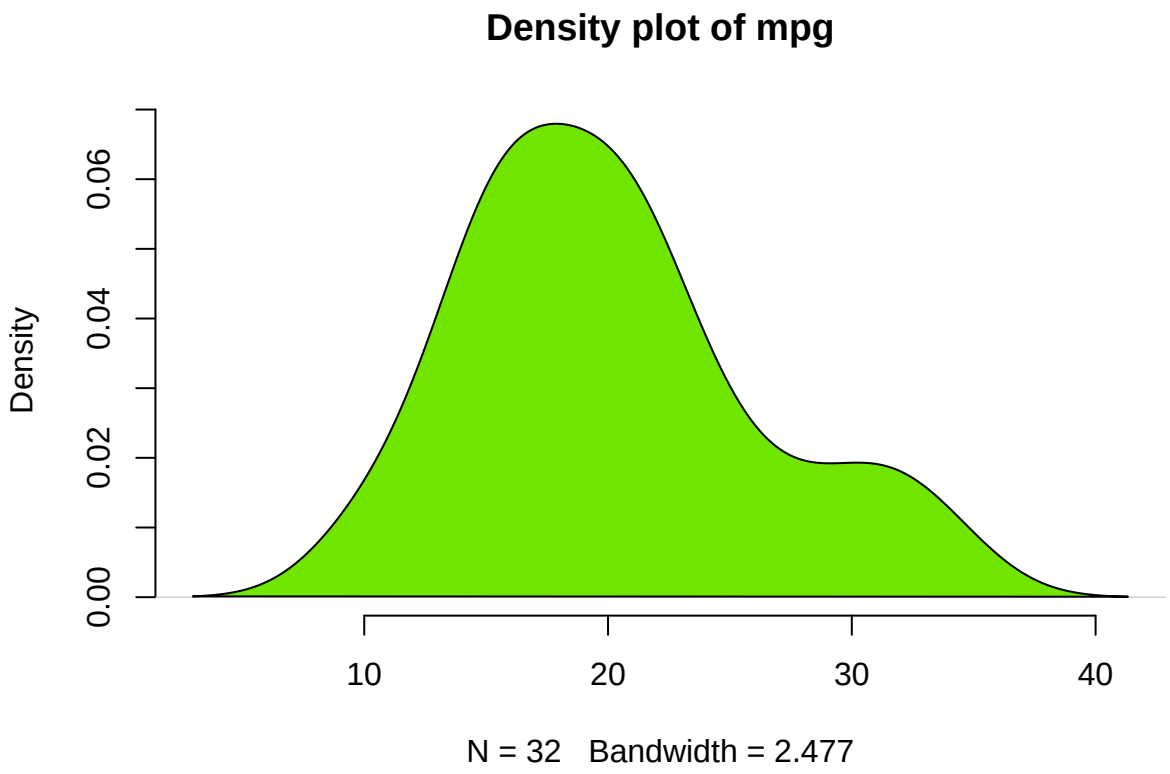
Density plots: density():

```
# The density data
dens <- density(mtcars$mpg)
# plot density
plot(dens, frame = FALSE, col = "red",
     main = "Density plot of mpg")
```



Density plot using polygon():

```
plot(dens, frame = FALSE, col = "lightblue",
     main = "Density plot of mpg")
polygon(dens, col = "#70E500")
```



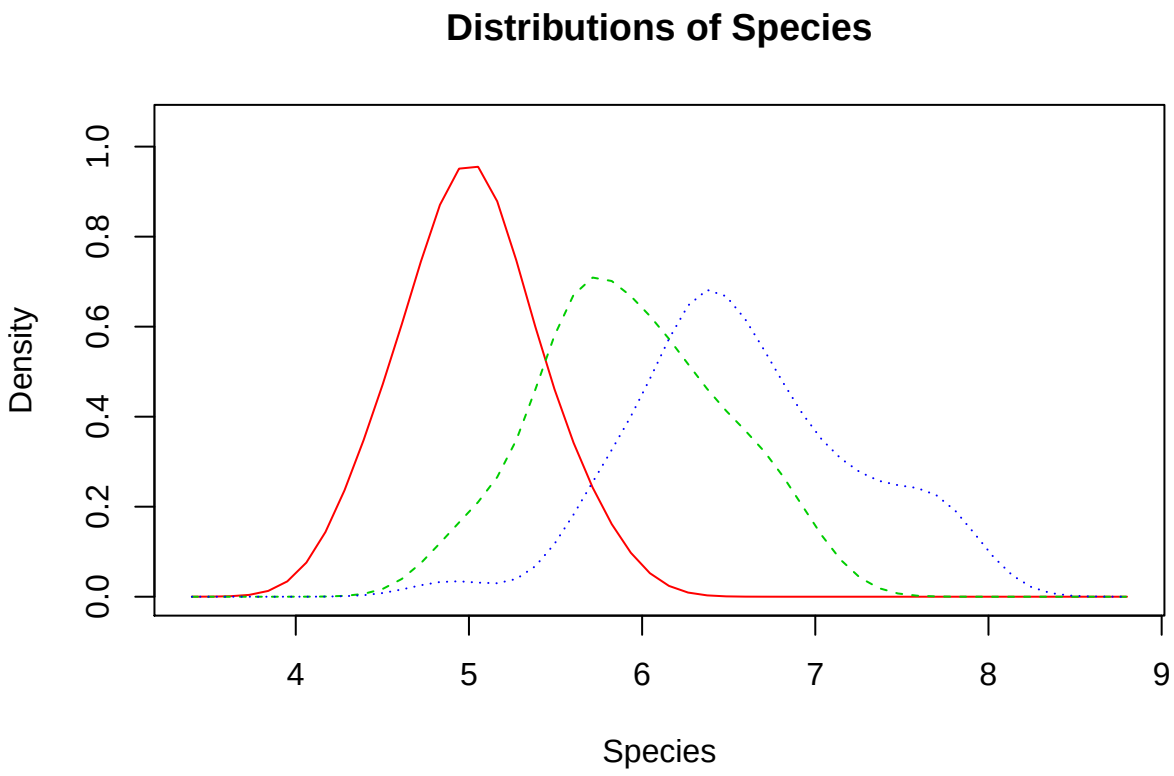
Additional->Comparing density Plots:

Example-1:

```
#install.packages("sm")
library(sm)
```

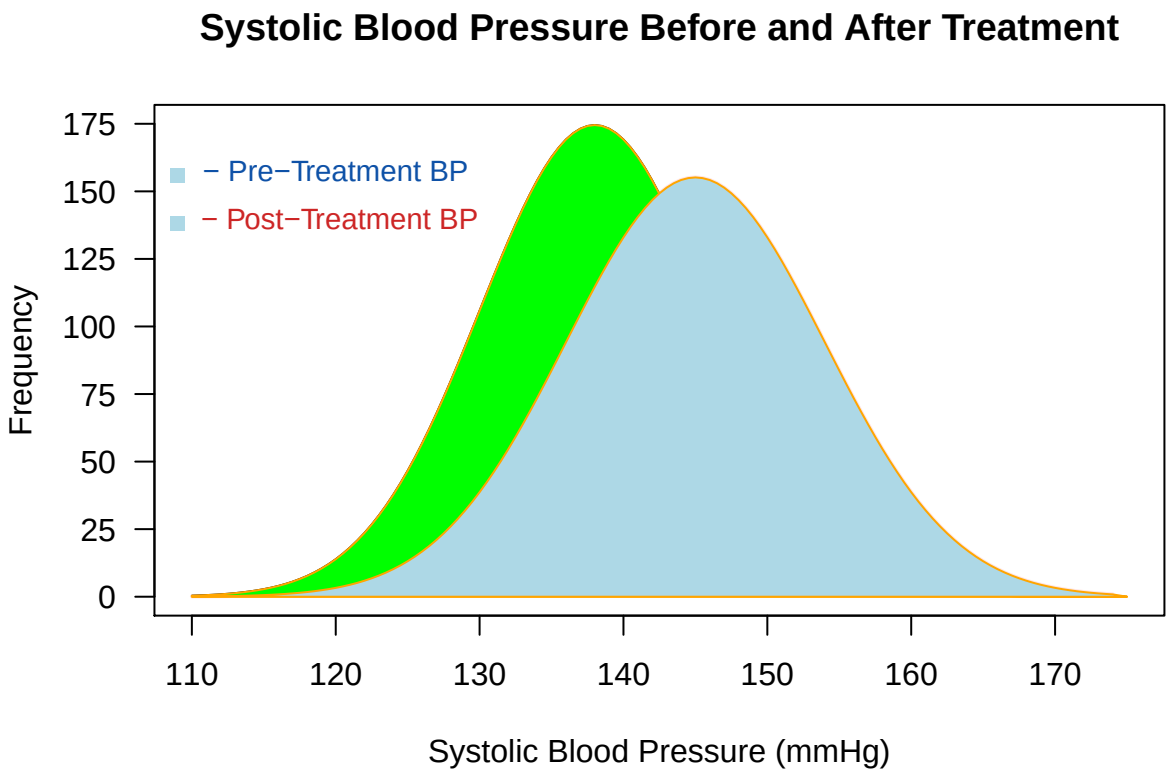
```
## Package 'sm', version 2.2-5.6: type help(sm) for summary information
```

```
sm.density.compare(iris$Sepal.Length, iris$Species, xlab="Species")
title(main="Distributions of Species")
```



Example-2:

```
x <- seq(from = 110, to = 174, by = 0.5)
y1 <- dnorm(x, mean = 145, sd = 9)
y2 <- dnorm(x, mean = 138, sd = 8)
plot(x, y1, type="l", lwd=2, col="mistyrose",
     main="Systolic Blood Pressure Before and After Treatment",
     xlab = "Systolic Blood Pressure (mmHg)",
     ylab = "Frequency", yaxt="n",
     xlim = c(110, 175), ylim = c(0, 0.05))
lines(x, y2)
polygon(c(110,x,175),c(0,y2,0), col="green",
     border = "orange")
polygon(c(117,x,175),c(0,y1,0), col="lightblue",
     border = "orange")
ylab=c(seq(from=0, to=175, by=25))
y=c(seq(from=0, to=0.05, length.out = 8))
axis(2,at=y,labels=ylab, las=1)
text(x = 120, y = 0.045, "- Pre-Treatment BP", col = "#0E51A7", cex = 0.9)
text(x = 120, y = 0.04, " - Post-Treatment BP", col = "firebrick3", cex = 0.9)
points(109, 0.0445, pch = 15, col = "lightblue")
points(109, 0.0395, pch = 15, col = "lightblue")
```


Systolic Blood Pressure Before and After Treatment
Pre−Treatment BP
Post−Treatment BP
Frequency
175
150
125
100
75
50
25
0
110
120
130
140
150
160
170
Systolic Blood Pressure (mmHg)

Chapter 6

Default Line Plots in R

Note that the function `lines()` need a `plot()`.

Basic line plots:

Example-1:

```
par(mfrow=c(2,1), mar=c(3,3,1,0)+.5, mgp=c(1.6,.6,0))
x <- 1:10
y1 <- x*x
y2  <- 3*y1
# Basic stair steps plot
plot(x, y1, type = "S",plot="pink")
```

```
## Warning in plot.window(...): "plot" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "plot" is not a graphical parameter
```

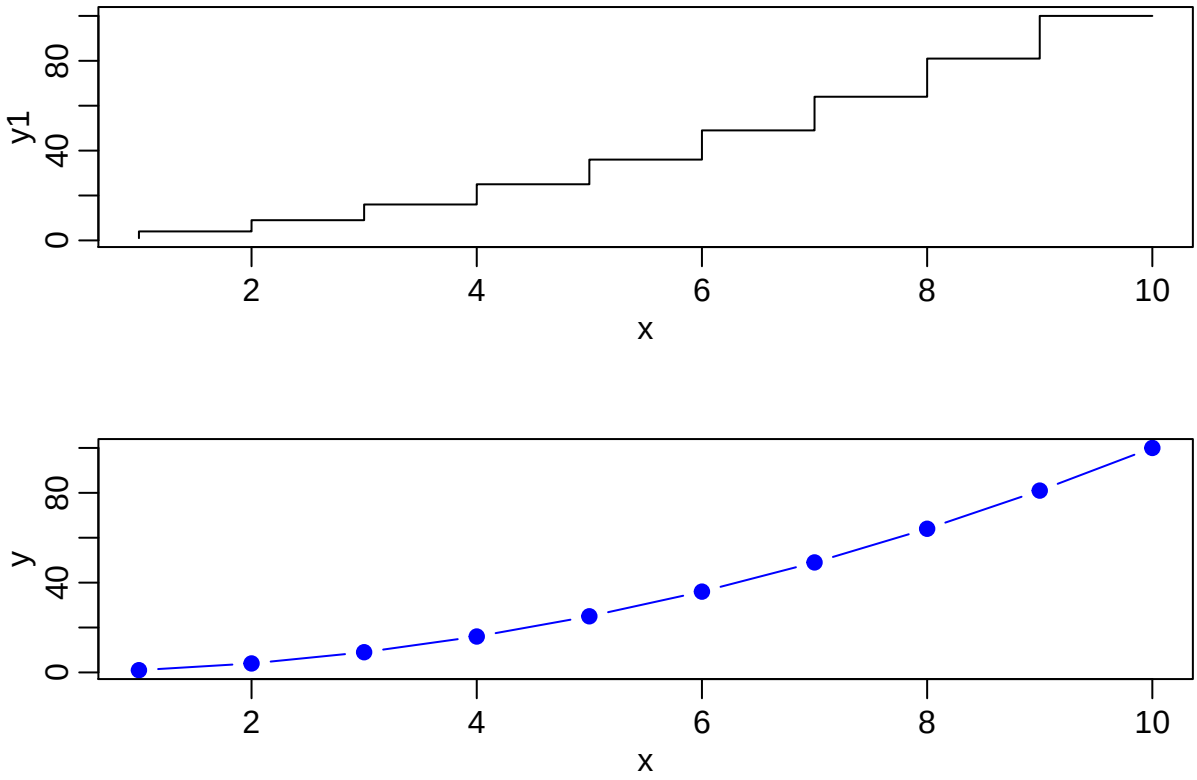
```
## Warning in axis(side = side, at = at, labels = labels, ...): "plot" is not
## a graphical parameter
```

```
## Warning in axis(side = side, at = at, labels = labels, ...): "plot" is not
## a graphical parameter
```

```
## Warning in box(...): "plot" is not a graphical parameter
```

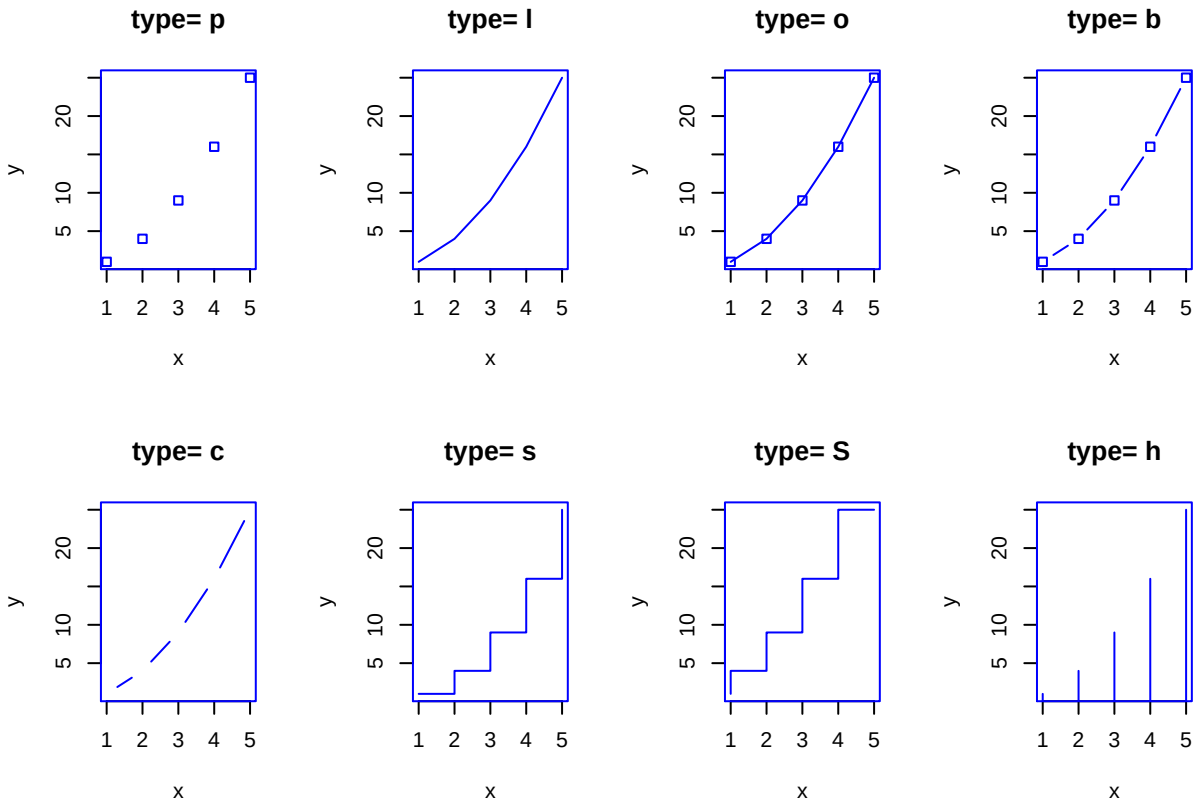
```
## Warning in title(...): "plot" is not a graphical parameter
```

```
# Points and line
plot(x, y1, type = "b", pch = 19,
     col = "blue", xlab = "x", ylab = "y")
```



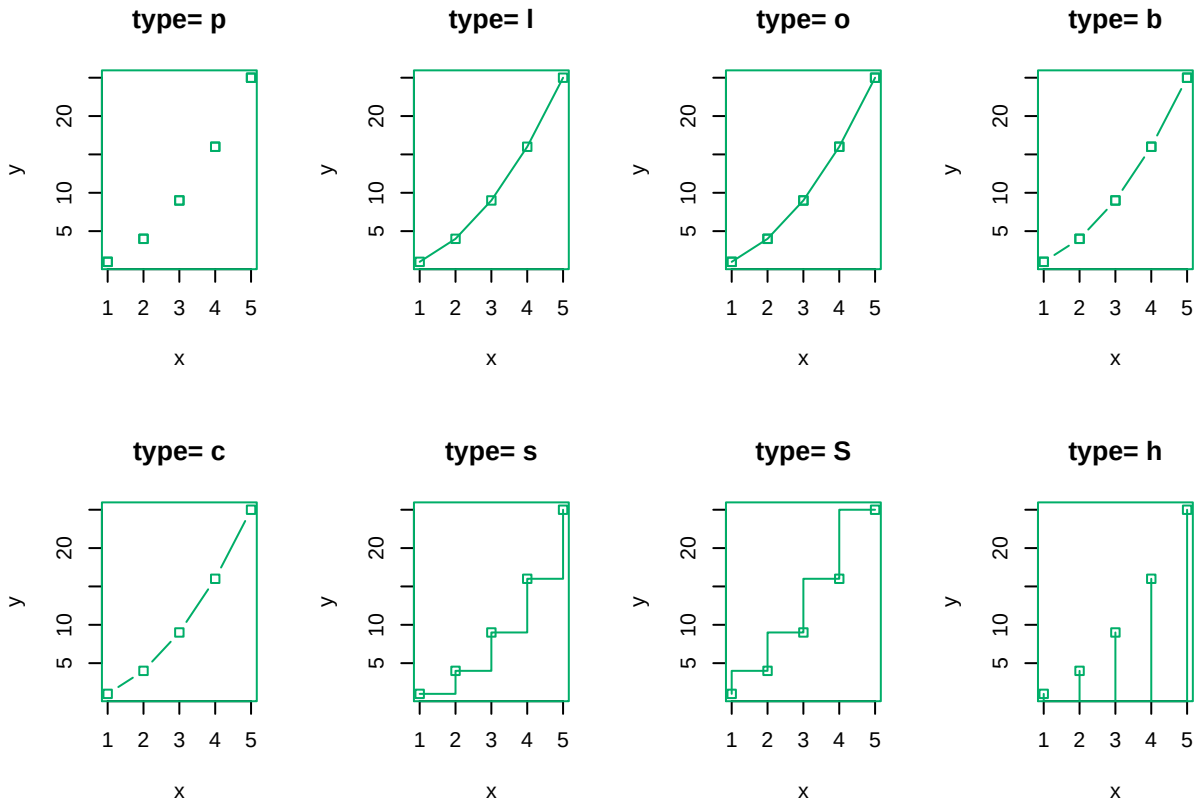
Example-2: As functions for different types

```
x <- c(1:5); y <- x*x # create example data
par(pch=22, col="blue") # plotting symbol and color
par(mfrow=c(2,4)) # all plots on one page
opts = c("p","l","o","b","c","s","S","h")
for(i in 1:length(opts)){
  heading = paste("type=",opts[i])
  plot(x, y, type="n", main=heading)
  lines(x, y, type=opts[i])
}
```



Example-3: Plot the points type=“”

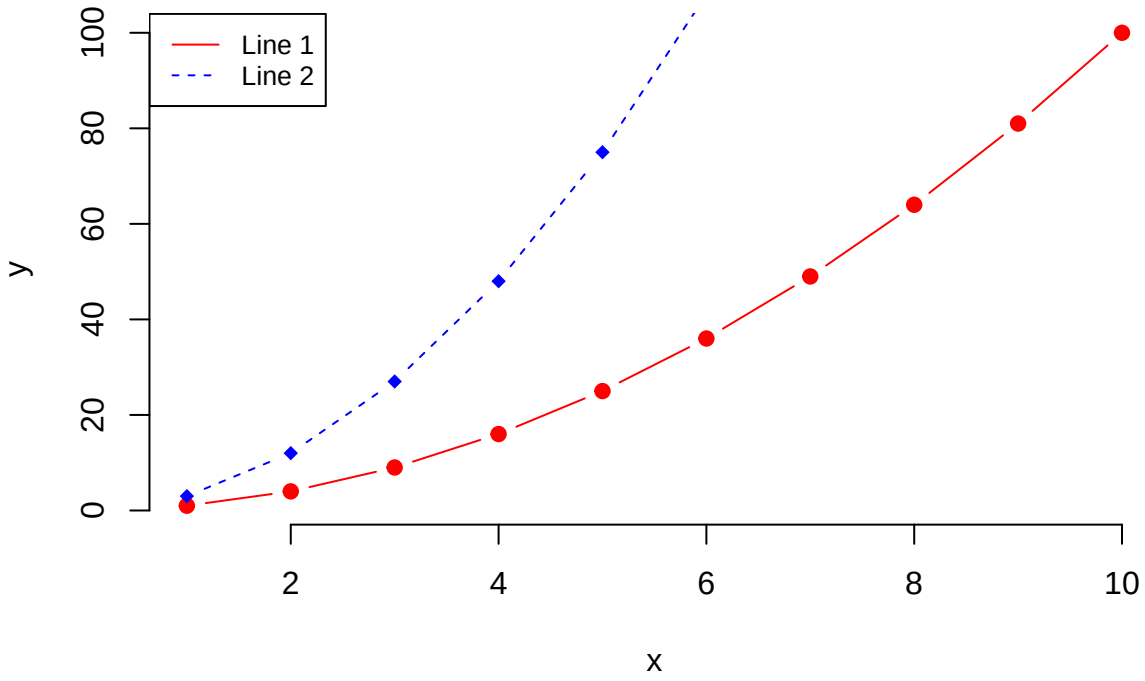
```
x <- c(1:5); y <- x*x # create example data
par(pch=22, col="#00AE68") # plotting symbol and color
par(mfrow=c(2,4)) # all plots on one page
opts = c("p","l","o","b","c","s","S","h")

for(i in 1:length(opts)){
  heading = paste("type=",opts[i])
  plot(x, y, main=heading)
  lines(x, y, type=opts[i]) }
```



Multiple lines plots:

Example-1:

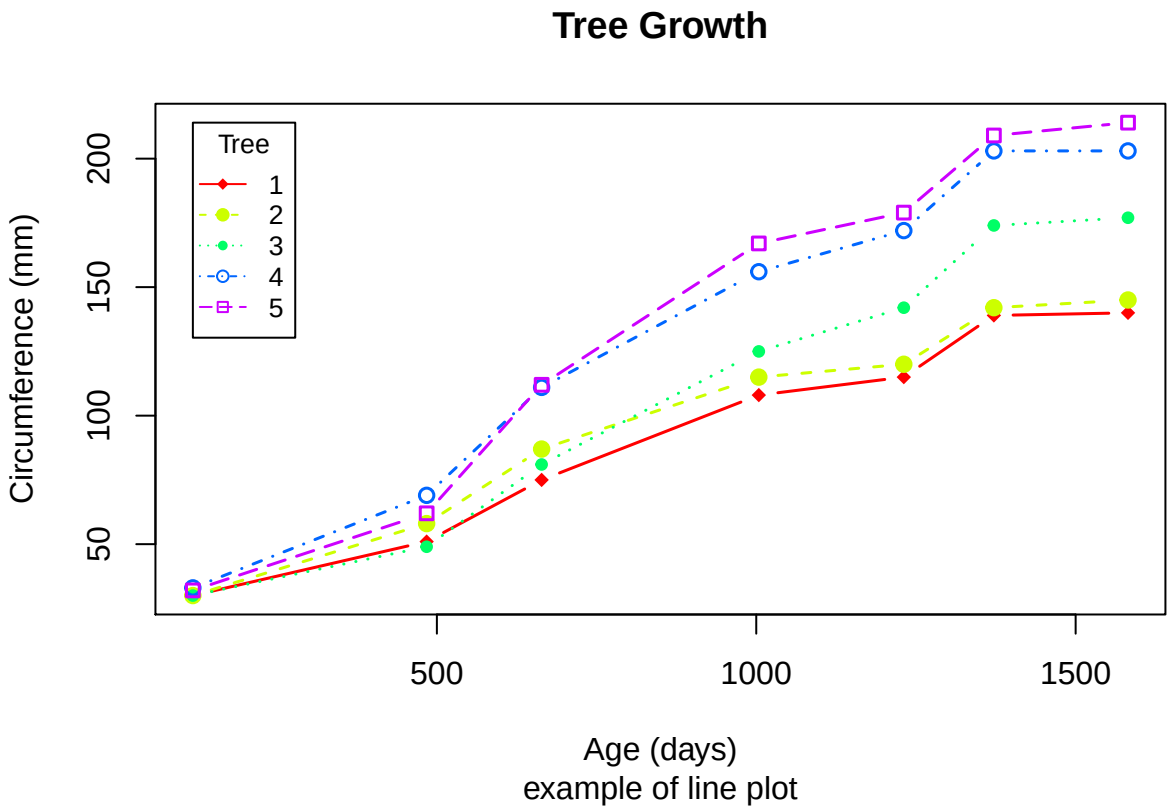
```
# First line
x <- 1:10
y1 <- (x*x)
plot(x, y1, type = "b", frame = FALSE, pch = 19,
     col = "red", xlab = "x", ylab = "y")
# A second line
lines(x, y2, pch = 18, col = "blue", type = "b", lty = 2)
# Add a legend to the plot
legend("topleft", legend=c("Line 1", "Line 2"),
       col=c("red", "blue"), lty = 1:2, cex=0.8)
```



Example-2: As functions for different types

```
# Create Line Chart
# convert factor to numeric for convenience
Orange$Tree <- as.numeric(Orange$Tree)
ntrees <- max(Orange$Tree)

# get the range for the x and y axis
xrange <- range(Orange$age)
yrange <- range(Orange$circumference)

# set up the plot
plot(xrange, yrange, type="n", xlab="Age (days)",
   ylab="Circumference (mm)" )
colors <- rainbow(ntrees)
linetype <- c(1:ntrees)
plotchar <- seq(18,18+ntrees,1)

# add lines
for (i in 1:ntrees) {
  tree <- subset(Orange, Tree==i)
  lines(tree$age, tree$circumference, type="b", lwd=1.5,
    lty=linetype[i], col=colors[i], pch=plotchar[i])
}

# add a title and subtitle
title("Tree Growth", "example of line plot")
```

```
# add a legend
legend(xrange[1], yrange[2], 1:ntrees, cex=0.8, col=colors,
   pch=plotchar, lty=linetype, title="Tree")
```



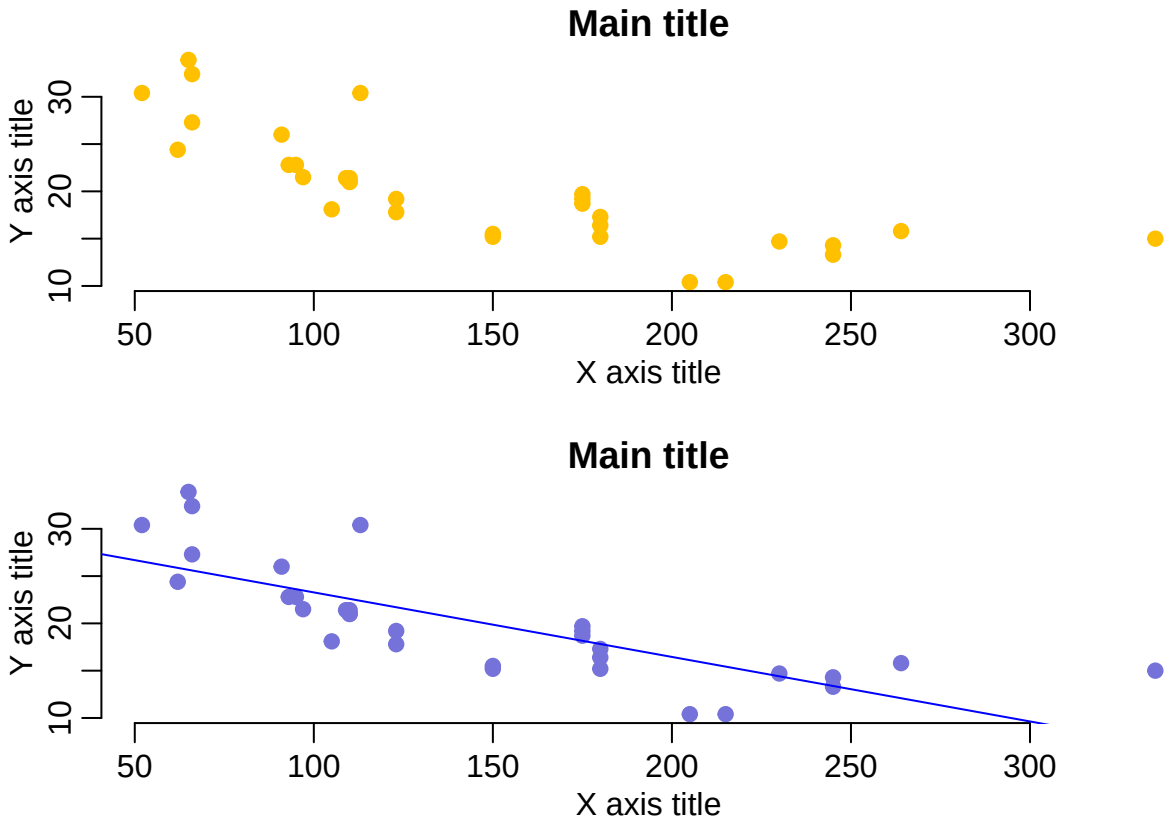
Chapter 7

Default Scatter Plots in R

R base scatter plot: plot():

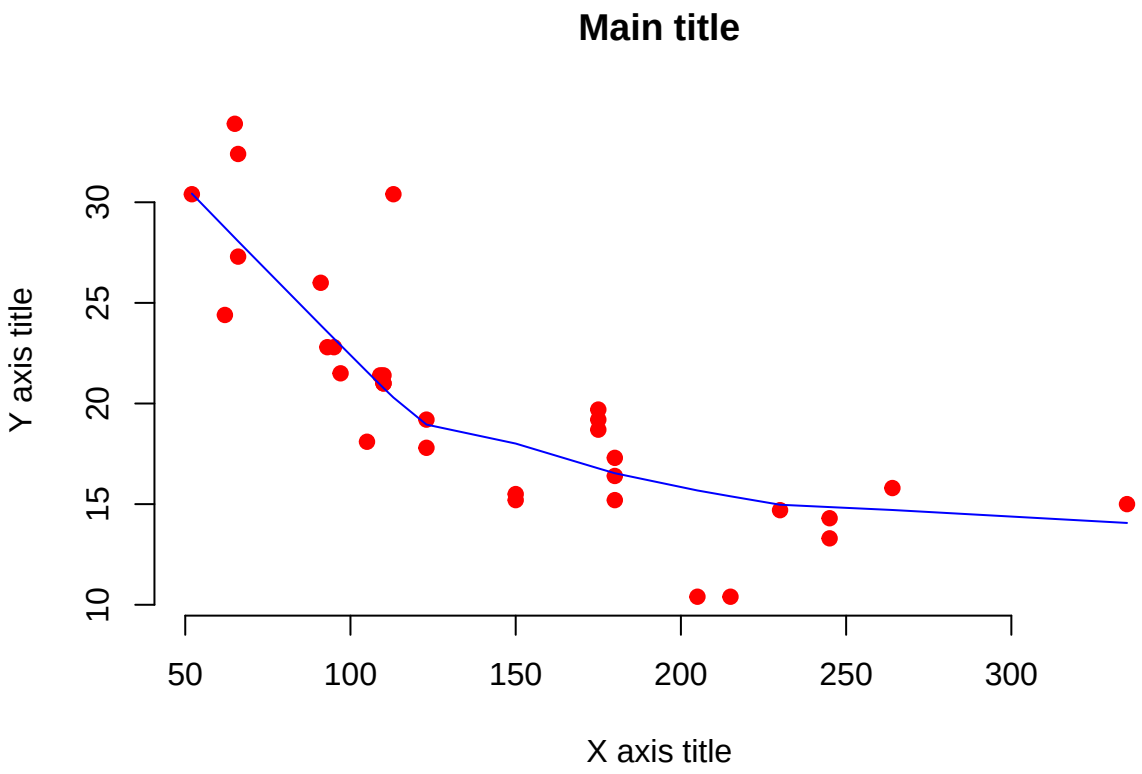
Example-1:

```
par(mfrow=c(2,1), mar=c(3,3,1,0)+.5, mgp=c(1.6,.6,0))
x <- mtcars$hp
y <- mtcars$mpg
# Plot with main and axis titles
# Change point shape (pch = 19) and remove frame.
plot(x, y, main = "Main title",
     xlab = "X axis title", ylab = "Y axis title",
     pch = 19, frame = FALSE,col="#FFC000")
# Add regression line
plot(x, y, main = "Main title",
     xlab = "X axis title", ylab = "Y axis title",
     pch = 19, frame = FALSE,col="#7573D9")
abline(lm(y ~ x, data = mtcars), col = "blue")
```



Example-2:

```
x <- mtcars$hp
y <- mtcars$mpg
# Add loess fit
plot(x, y, main = "Main title",
     xlab = "X axis title", ylab = "Y axis title",
     pch = 19, frame = FALSE,col="red")
lines(lowess(x, y), col = "blue")
```



Addtional->Enhanced scatter plots:

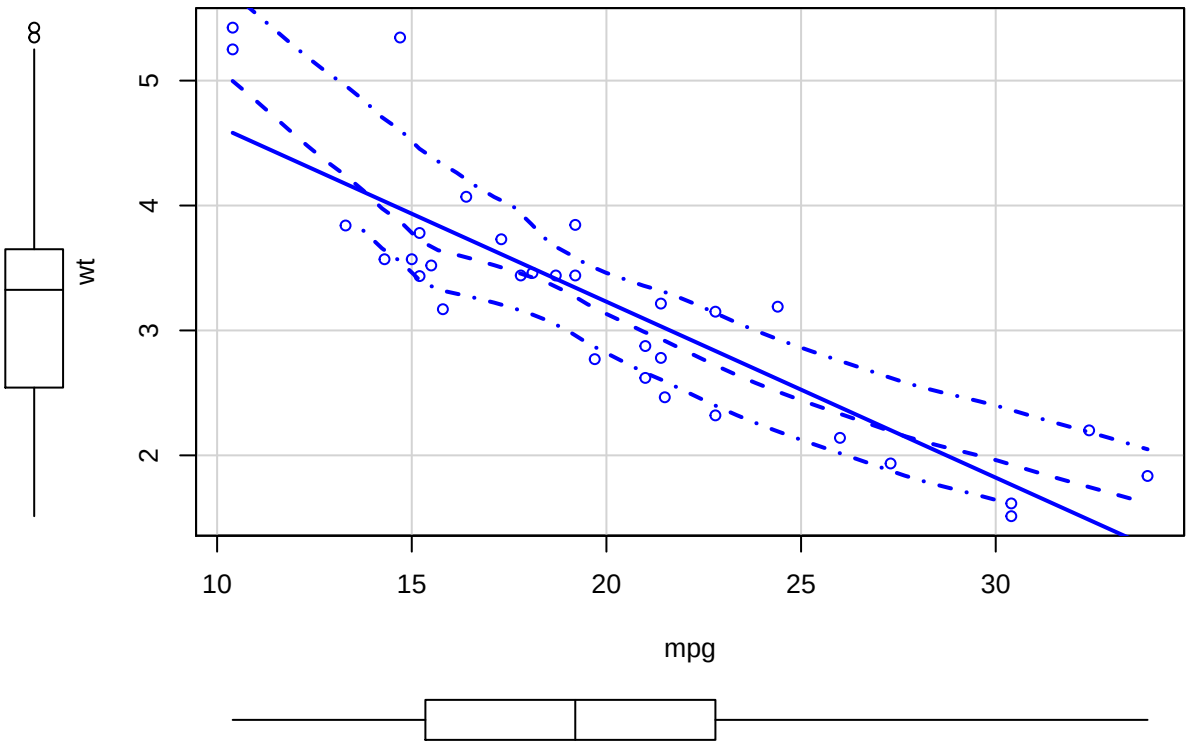
The components of the plot contains:

- the points
- the regression line (in green)
- the smoothed conditional spread (in red dashed line)
- the non-parametric regression smooth (solid line, red)

```
#install.packages("car")
library("car")
```

```
## Loading required package: carData
```

```
scatterplot(wt ~ mpg, data = mtcars)
```

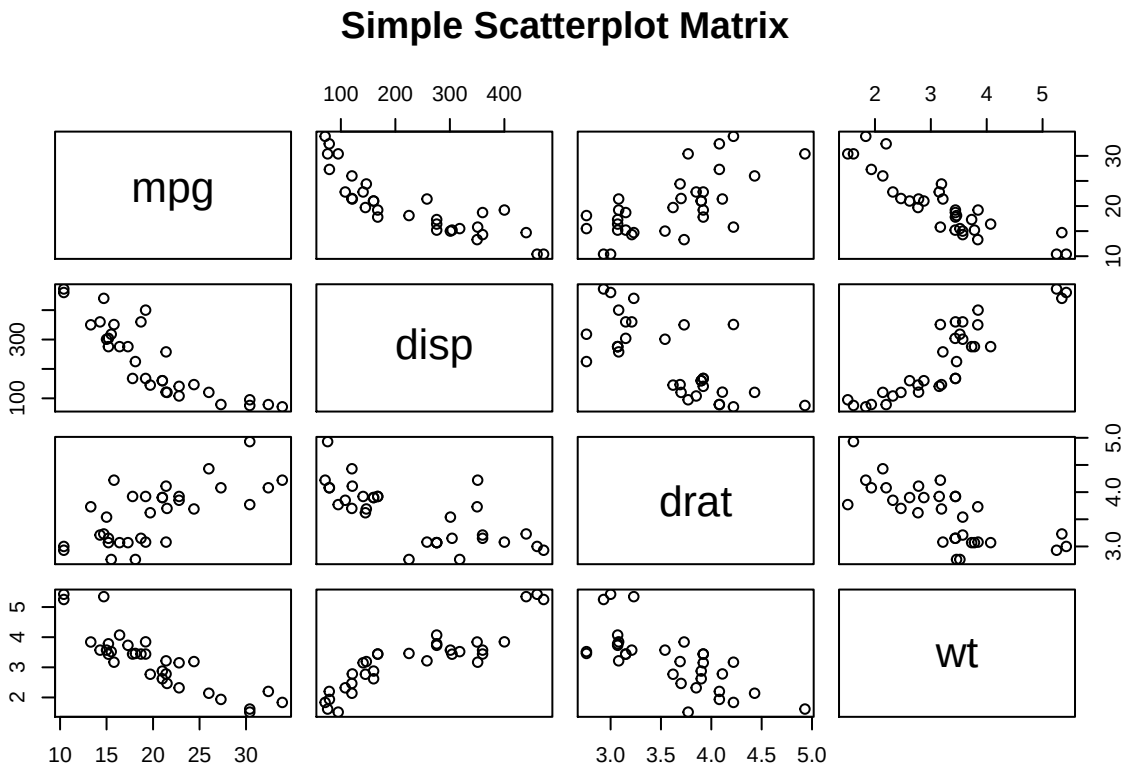

8
wt
5
4
3
2
10
15
20
25
30
mpg

Chapter 8

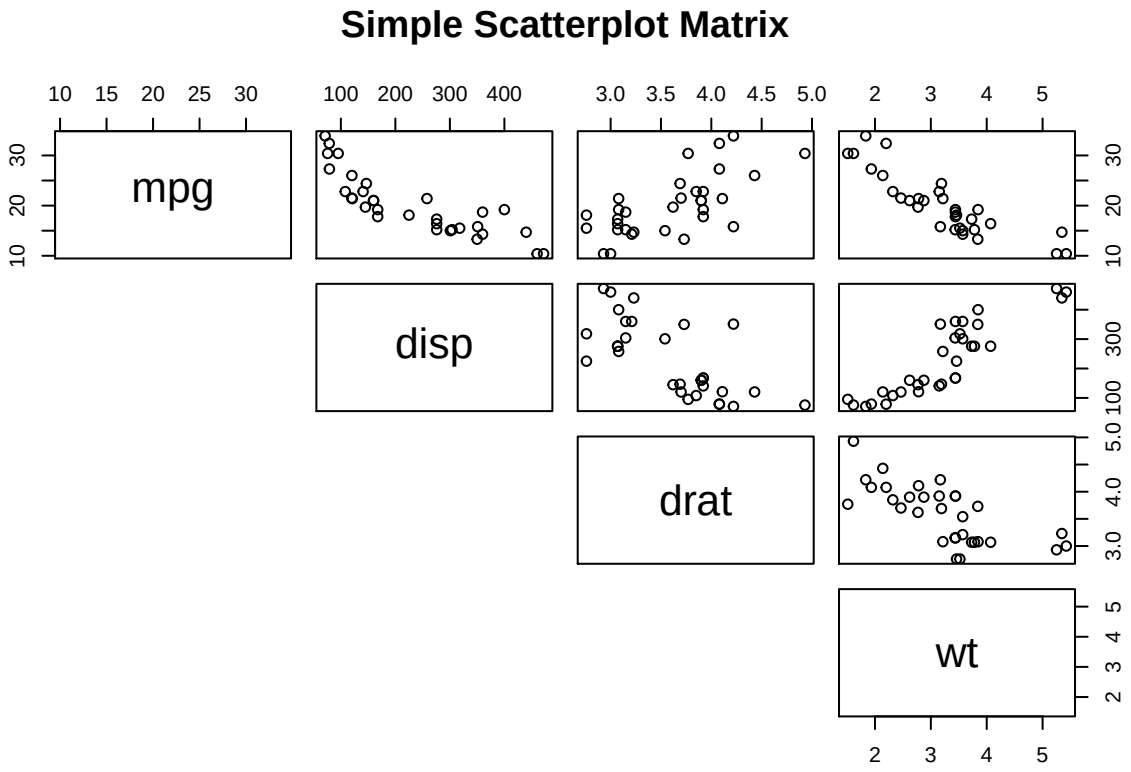
Default Scatter Plot Matrices in R

Scatter Plot Matrices:

```
par(mfrow=c(2,1))
pairs(~mpg+disp+drat+wt,data=mtcars,
   main="Simple Scatterplot Matrix")
```

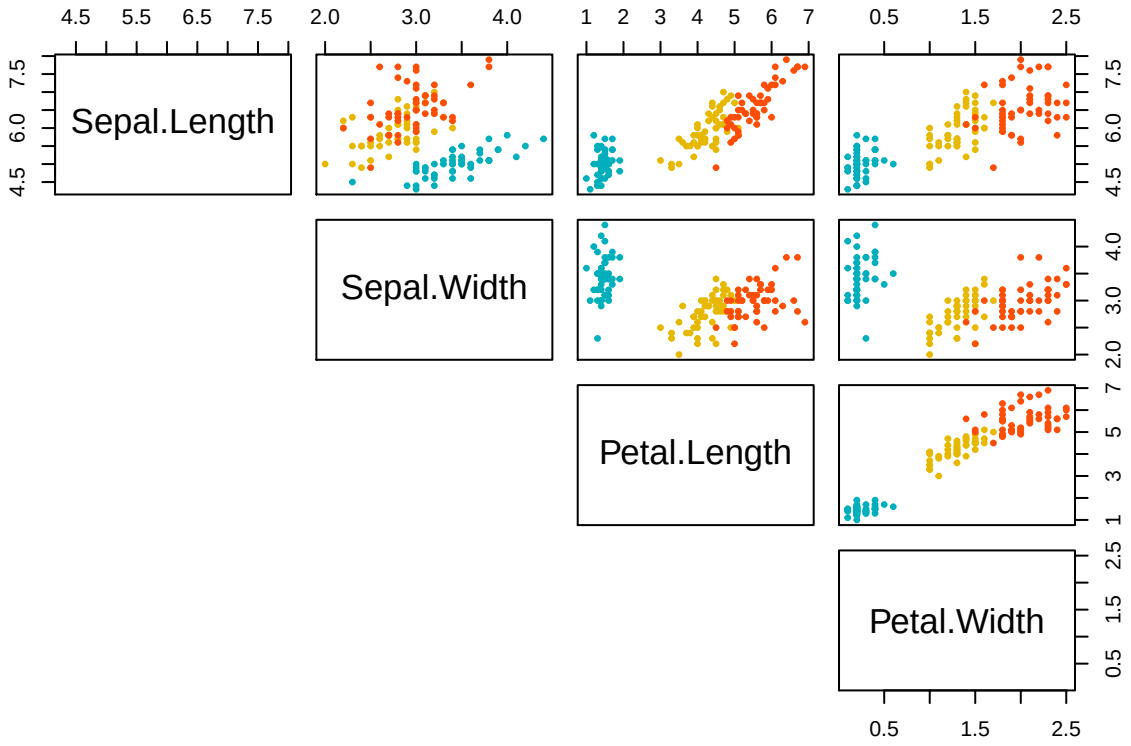



```
pairs(~mpg+disp+drat+wt,data=mtcars,lower.panel = NULL,
   main="Simple Scatterplot Matrix")
```



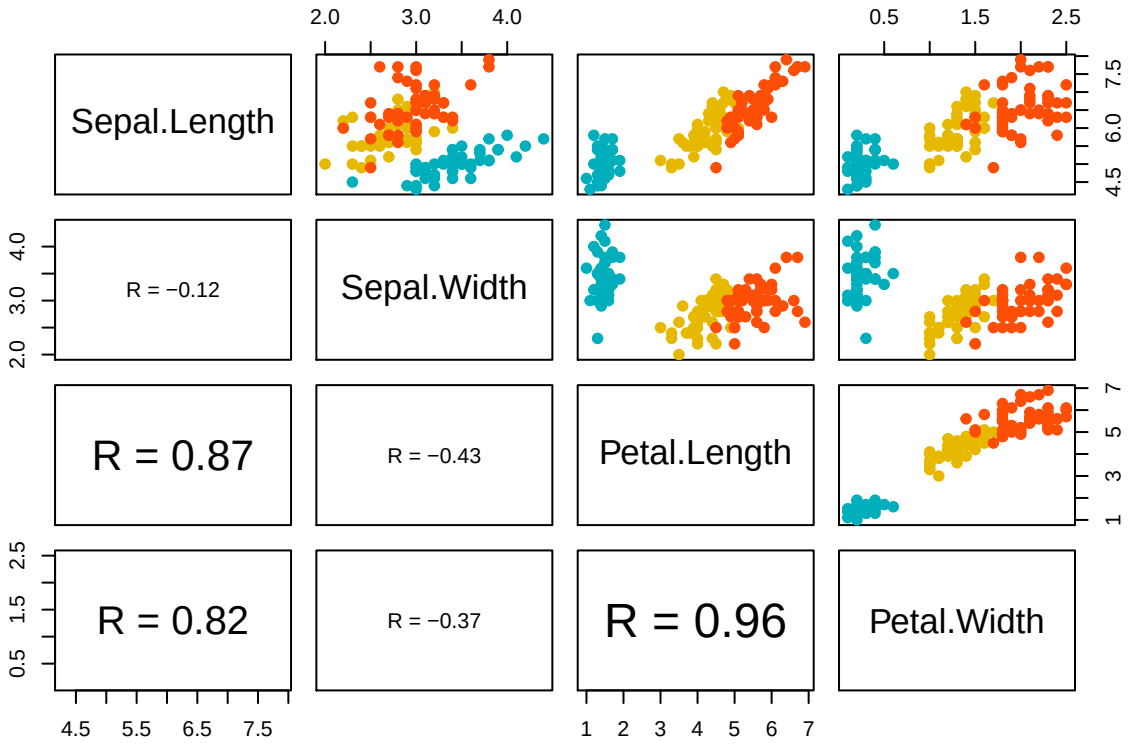
Color points by groups Scatter plots:

```
my_col <- c("#00AFBB", "#E7B800", "#FC4E07")
pairs(iris[,1:4], pch = 19,  cex = 0.5,
      col = my_col[iris$Species],
      lower.panel=NULL)
```



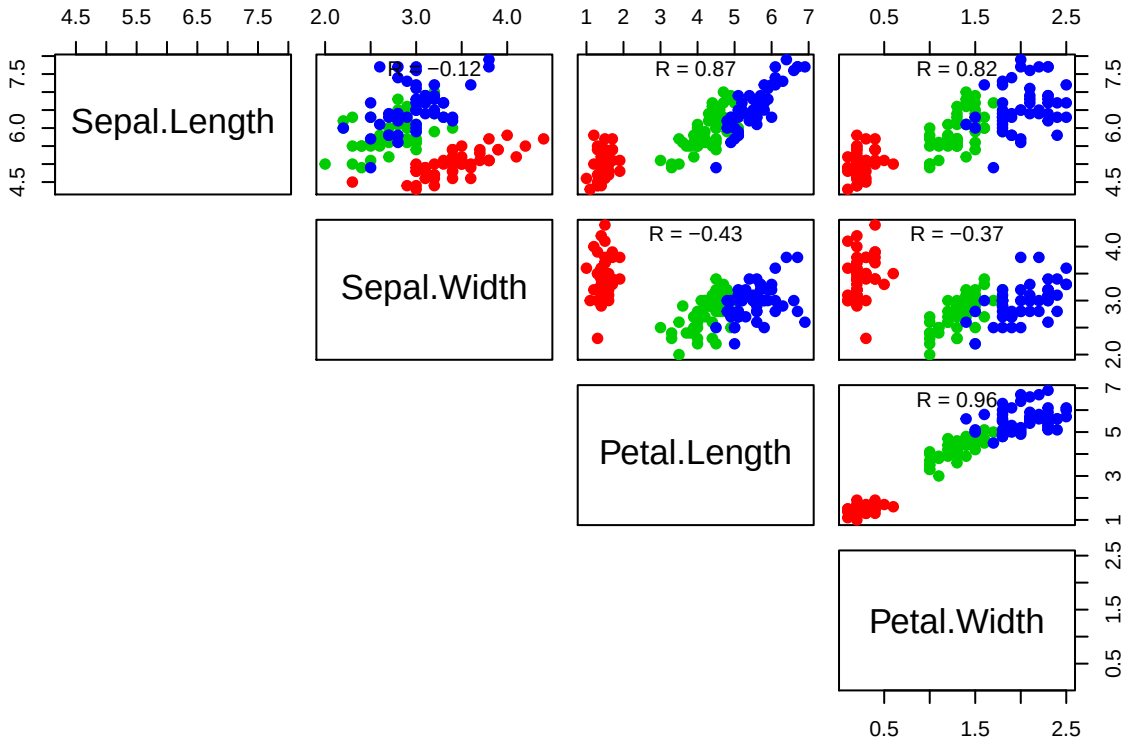
Add correlations on the lower panels:

```
# Correlation panel
my_col <- c("#00AFBB", "#E7B800", "#FC4E07")
panel.cor <- function(x, y){
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    r <- round(cor(x, y), digits=2)
    txt <- paste0("R = ", r)
    cex.cor <- 0.8/strwidth(txt)
    text(0.5, 0.5, txt, cex = cex.cor * r)
}
# Customize upper panel
upper.panel<-function(x, y){
  points(x,y, pch = 19, col = my_col[iris$Species])
}
# Create the plots
pairs(iris[,1:4],
      lower.panel = panel.cor,
      upper.panel = upper.panel)
```



Add correlations on the scatter plots:

```
my_col <- c("#00AFBB", "#E7B800", "#FC4E07")
# Customize upper panel
upper.panel<-function(x, y){
  points(x,y, pch=19, col=c("red", "green3", "blue")[iris$Species])
  r <- round(cor(x, y), digits=2)
  txt <- paste0("R = ", r)
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  text(0.5, 0.9, txt)
}
pairs(iris[,1:4], lower.panel = NULL,
      upper.panel = upper.panel)
```



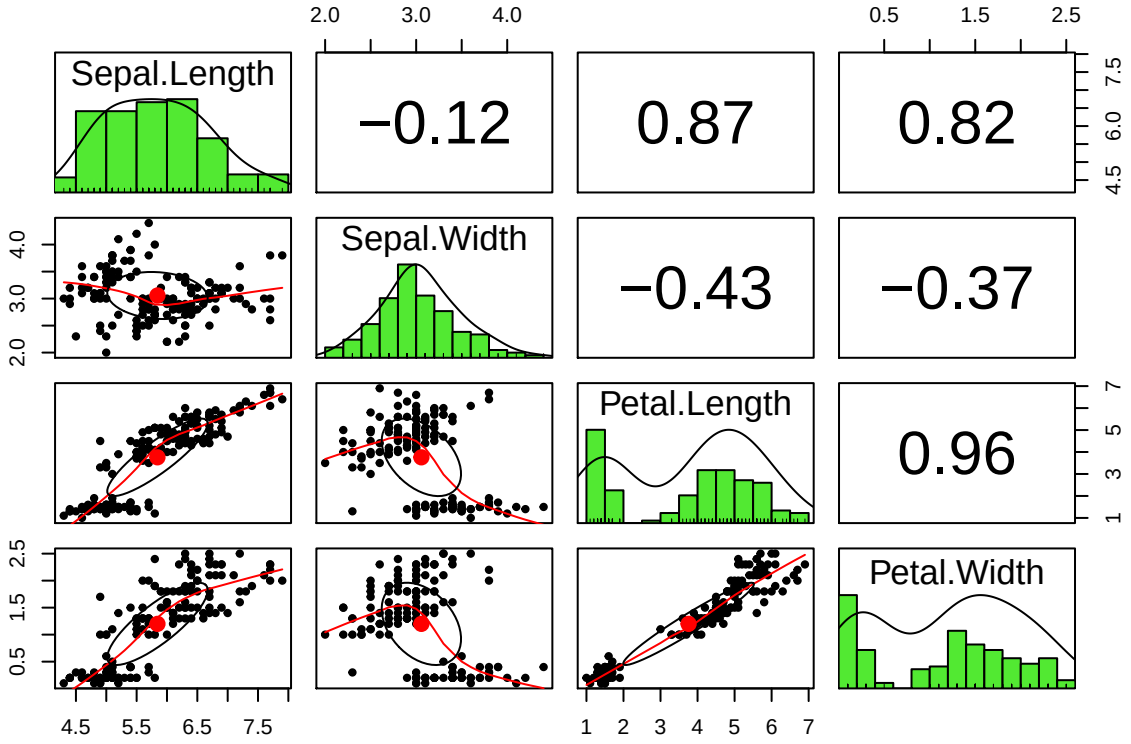
Additional-> the R package psych:

```
#install.packages("psych")
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following object is masked from 'package:car':
##
##     logit
```

```
pairs.panels(iris[,-5],
             method = "pearson", # correlation method
             hist.col = "#52E932",
             density = TRUE,  # show density plots
             ellipses = TRUE # show correlation ellipses
             )
```


Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
−0.12
0.87
0.82
−0.43
−0.37
0.96
2.0
3.0
4.0
0.5
1.5
2.5
4.5
5.5
6.5
7.5
1
2
3
4
5
6
7

Chapter 9

Strip charts:1-D scatter plots

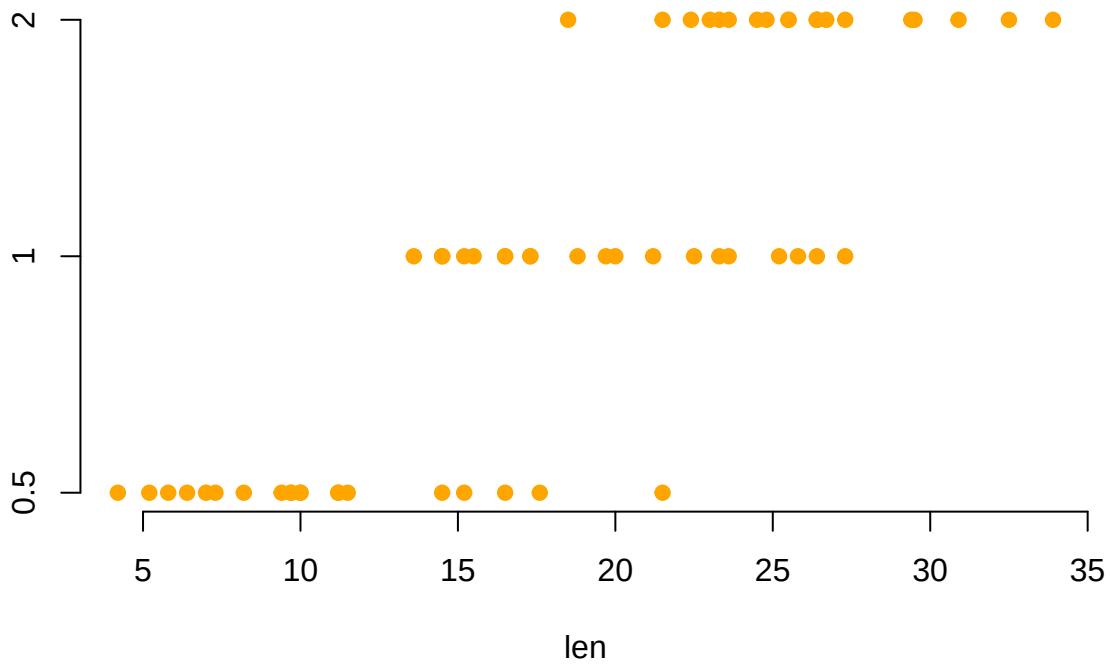
Formula:

stripchart(x, data = NULL method = “overplot”, jitter = 0.1)

- x: the data from which the plots are to be produced. Allowed values are one or a list of numeric vector, each corresponding to a component plot.
- data: a data.frame (or list) from which the variables in x should be taken.
- Method: the method to be used to separate coincident points. Allowed values are one of “overplot”, “jitter” or “stack”.
- jitter: when method = “jitter” is used, jitter gives the amount of jittering applied.

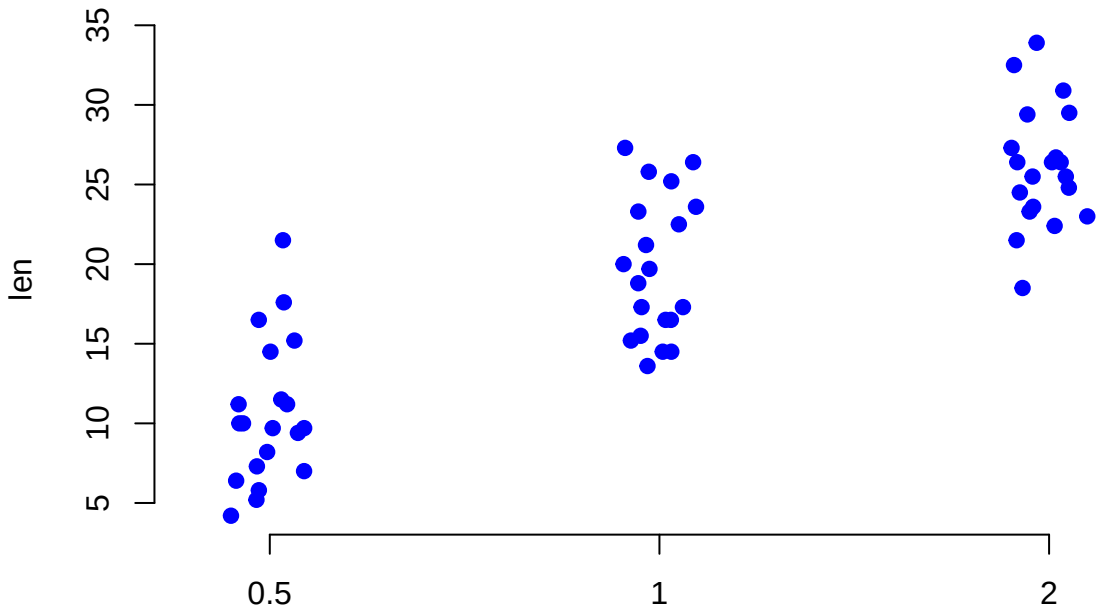
Example-1:

```
# Plot len by dose
stripchart(len ~ dose, data = ToothGrowth,
         pch = 19, frame = FALSE,col="orange")
```



Example-2:

```
# Vertical plot using method = "jitter"
stripchart(len ~ dose, data = ToothGrowth,
         pch = 19, frame = FALSE, vertical = TRUE,
         method = "jitter",col="blue")
```



Example-3:

```
# Change point shapes (pch) and colors by groups
# add main title and axis labels
stripchart(len ~ dose, data = ToothGrowth,
         frame = FALSE, vertical = TRUE,
         method = "jitter", pch = c(21, 18, 16),
         col = c("#00BF32", "#E69F00", "#56B4E9"),
         main = "Length by dose", xlab = "Dose", ylab = "Length")
```


Length by dose
Length
5
10
15
20
25
30
35
0.5
1
2
Dose

Chapter 10

Default Dot Plots in R

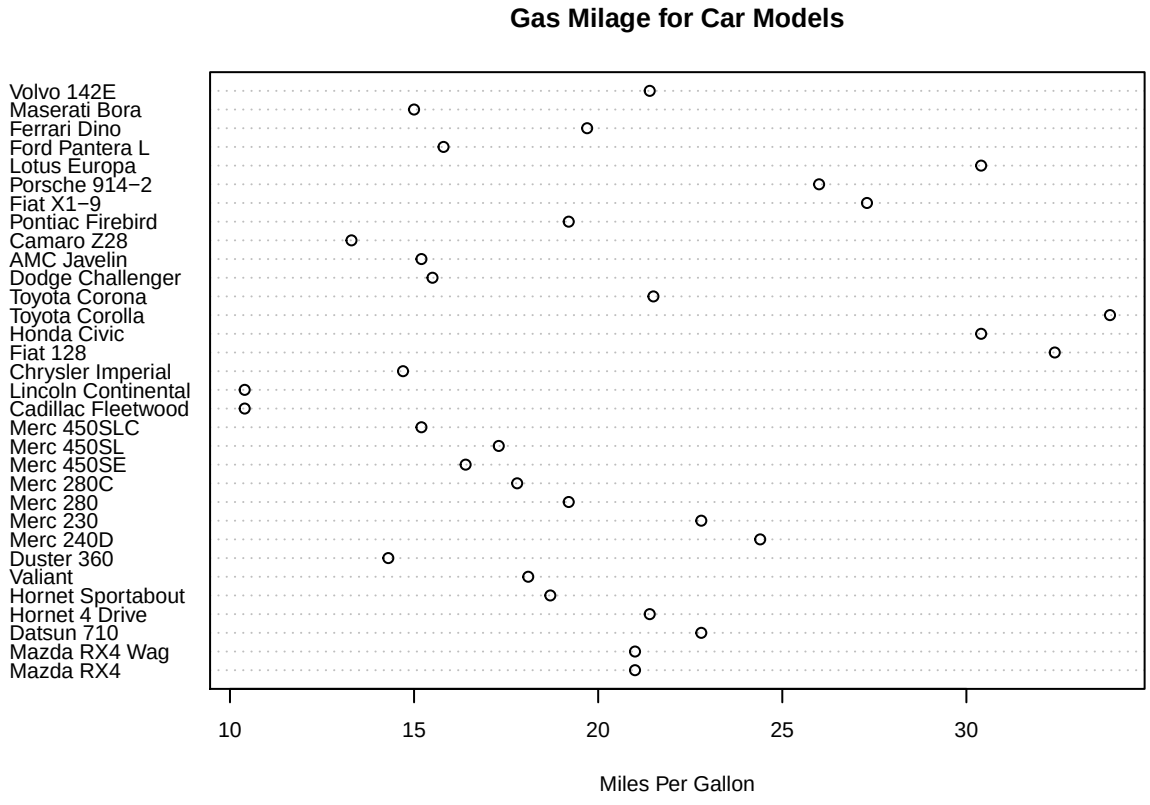
Formula:

dotchart(x, labels = NULL, groups = NULL, gcolor = par("fg"), color = par("fg"))

Simple Dot Plots numeric vectors:

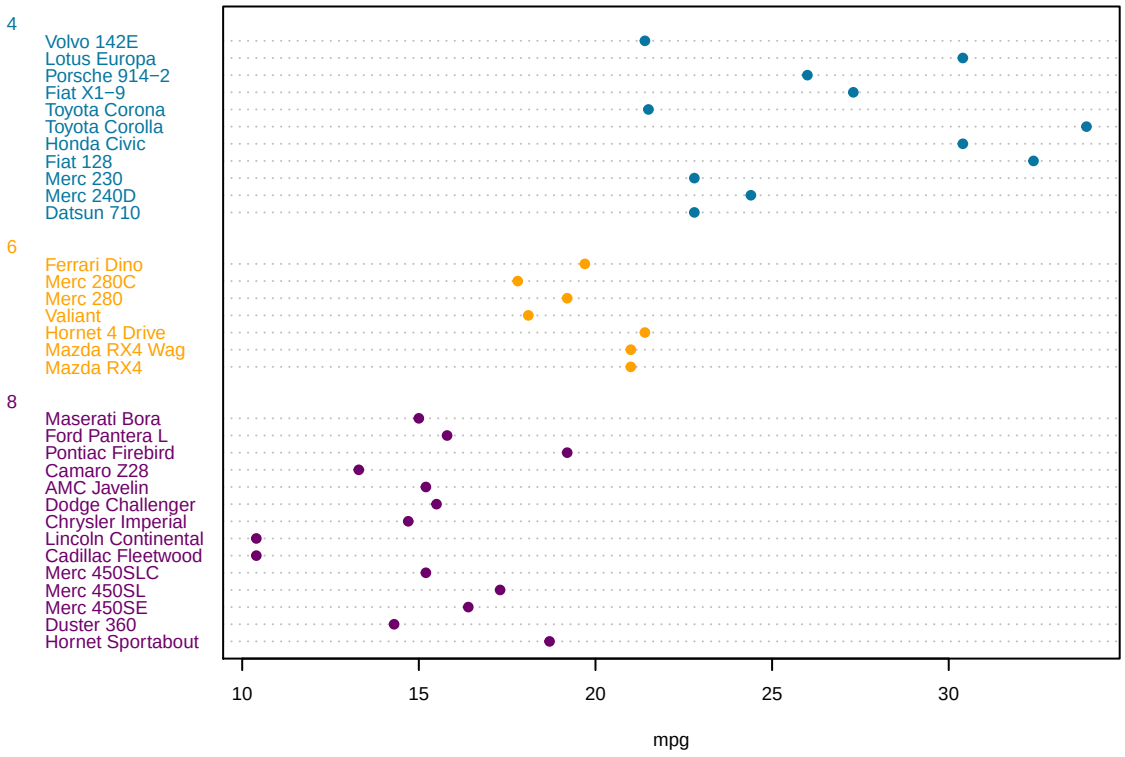
Example-1:

```
# Simple Dot Plot
dotchart(mtcars$mpg,labels=row.names(mtcars),cex=.7,
   main="Gas Milage for Car Models",
   xlab="Miles Per Gallon")
```



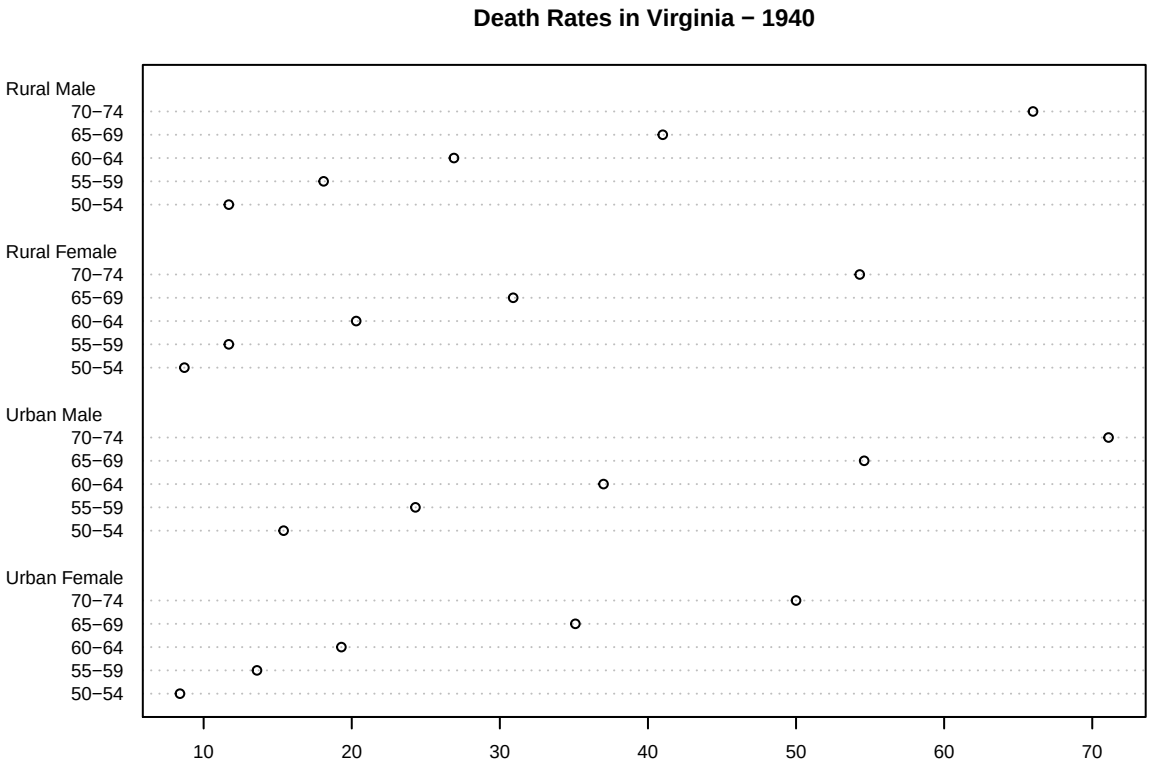
Example-2:

```
# Plot and color by groups cyl
grps <- as.factor(mtcars$cyl)
my_cols <- c("#0776A0", "#FFA100", "#6E0069")
dotchart(mtcars$mpg, labels = row.names(mtcars),
         groups = grps, gcolor = my_cols,
         color = my_cols[grps],
         cex = 0.6,  pch = 19, xlab = "mpg")
```



Dot Plot matrix:

```
dotchart(VADeaths, cex = 0.6,
         main = "Death Rates in Virginia - 1940")
```


Death Rates in Virginia − 1940
Rural Male
70−74
65−69
60−64
55−59
50−54
Rural Female
70−74
65−69
60−64
55−59
50−54
Urban Male
70−74
65−69
60−64
55−59
50−54
Urban Female
70−74
65−69
60−64
55−59
50−54
10
20
30
40
50
60
70

Chapter 11

Default Pie Charts in R

Formula:

pie(x, labels = names(x), radius = 0.8)

- x: a vector of non-negative numerical quantities. The values in x are displayed as the areas of pie slices.
- labels: character strings giving names for the slices.
- radius: radius of the pie circle. If the character strings labeling the slices are long it may be necessary to use a smaller radius.

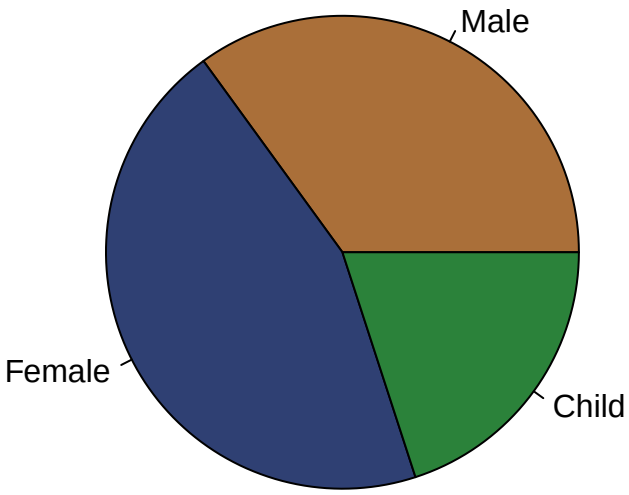
Basic pie charts: -> pie()

Example-1:

```
df <- data.frame(
  group = c("Male", "Female", "Child"),
  value = c(35, 45, 20)
  )
df
```

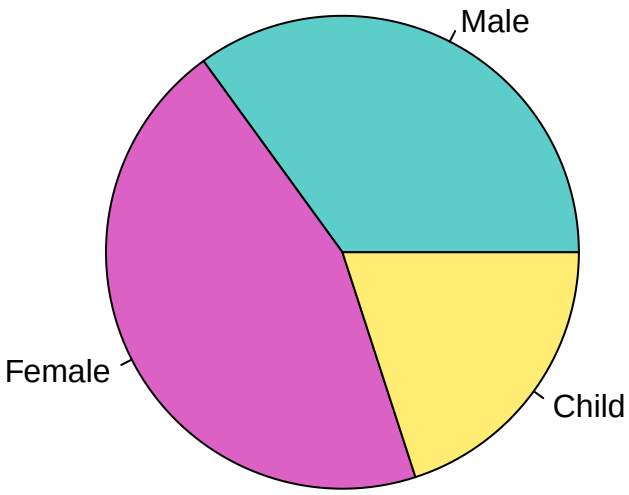
```
##    group value
## 1   Male    35
## 2 Female    45
## 3  Child    20
```

```
pie(df$value, labels = df$group, radius = 1,col=c("#AA6F39","#2F4073","#2B823A"))
```



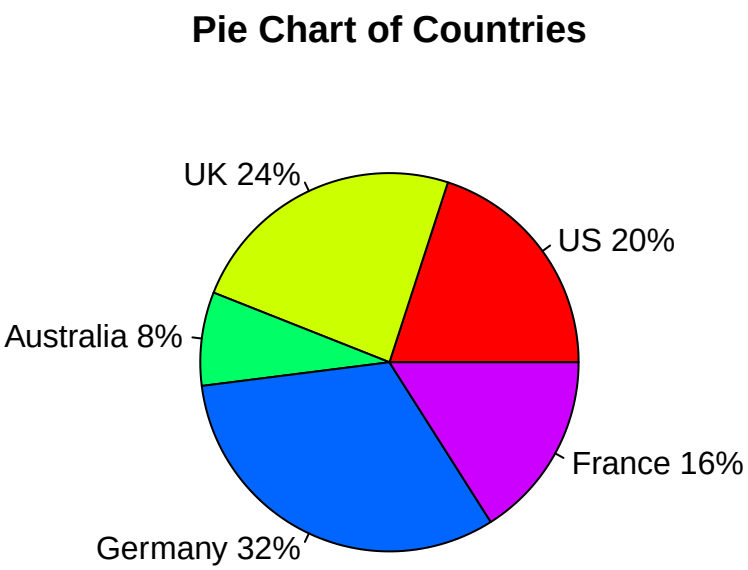
Example-2:

```
# Change colors
pie(df$value, labels = df$group, radius = 1,
    col = c("#5CCDC9", "#DB62C4", "#FFED73"))
```



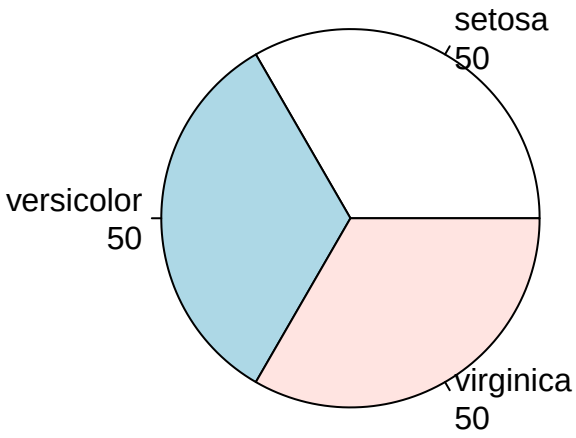
Example-3:

```
# Pie Chart with Percentages
slices <- c(10, 12, 4, 16, 8)
lbls <- c("US", "UK", "Australia", "Germany", "France")
pct <- round(slices/sum(slices)*100)
lbls <- paste(lbls, pct) # add percents to labels
lbls <- paste(lbls,"%",sep="") # ad % to labels
pie(slices,labels = lbls, col=rainbow(length(lbls)),
   main="Pie Chart of Countries")
```



Example-4:

```
# Pie Chart from data frame with Appended Sample Sizes
mytable <- table(iris$Species)
lbls <- paste(names(mytable), "\n", mytable, sep="")
pie(mytable, labels = lbls,
   main="Pie Chart of Species\n (with sample sizes)")
```


Pie Chart of Species
(with sample sizes)
setosa
50
versicolor
50
virginica
50

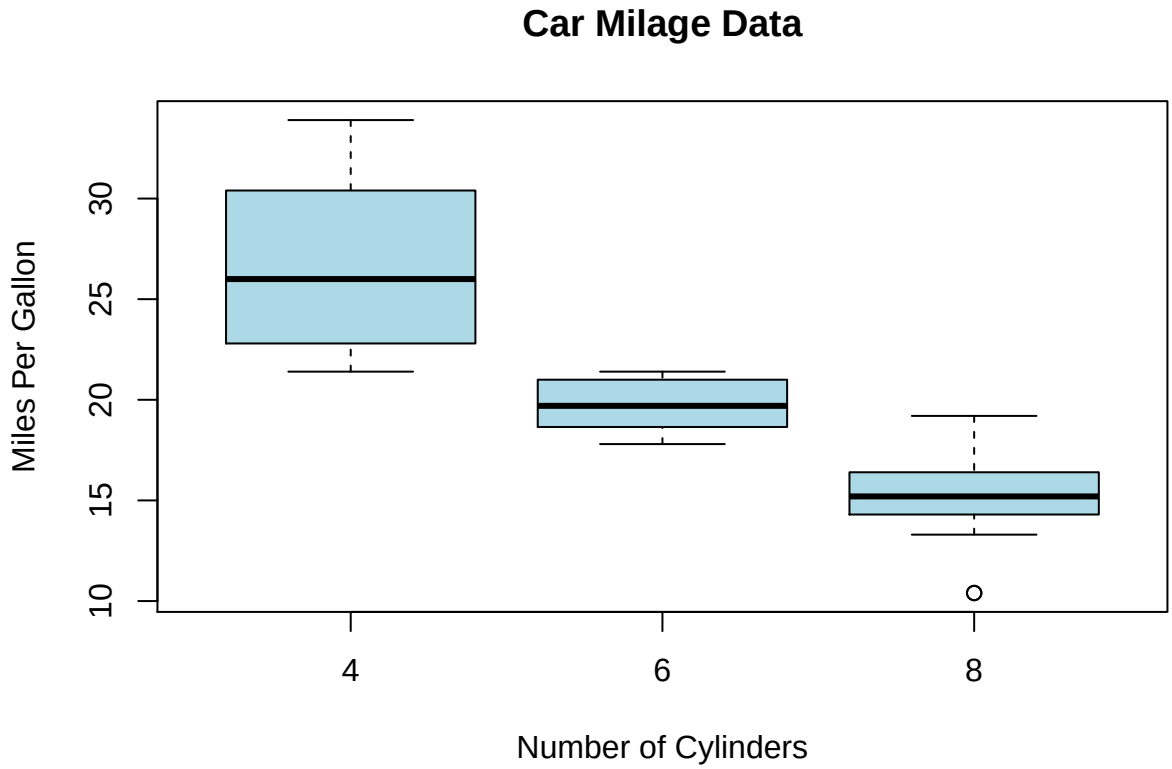
Chapter 12

Default Box Plots in R

Basic box plots:

Example-1:

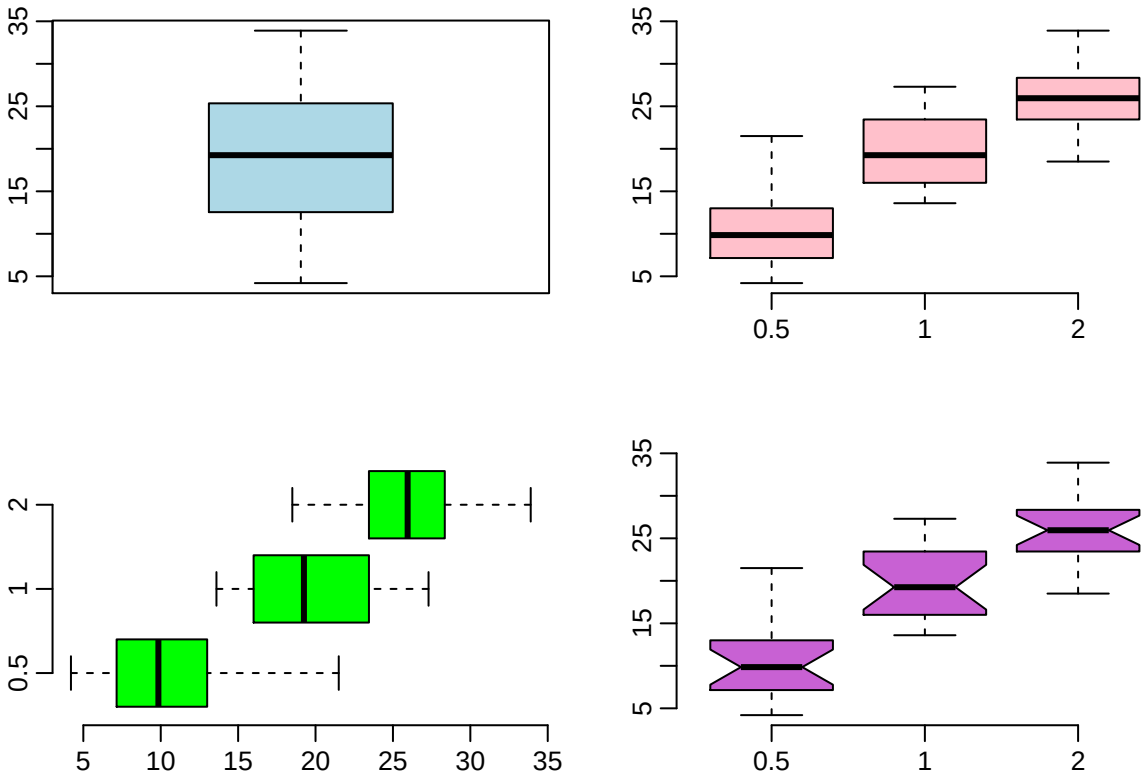
```
# Boxplot of MPG by Car Cylinders
boxplot(mpg~cyl,data=mtcars, main="Car Milage Data",
   xlab="Number of Cylinders", ylab="Miles Per Gallon",col="lightblue")
```



Example-2:

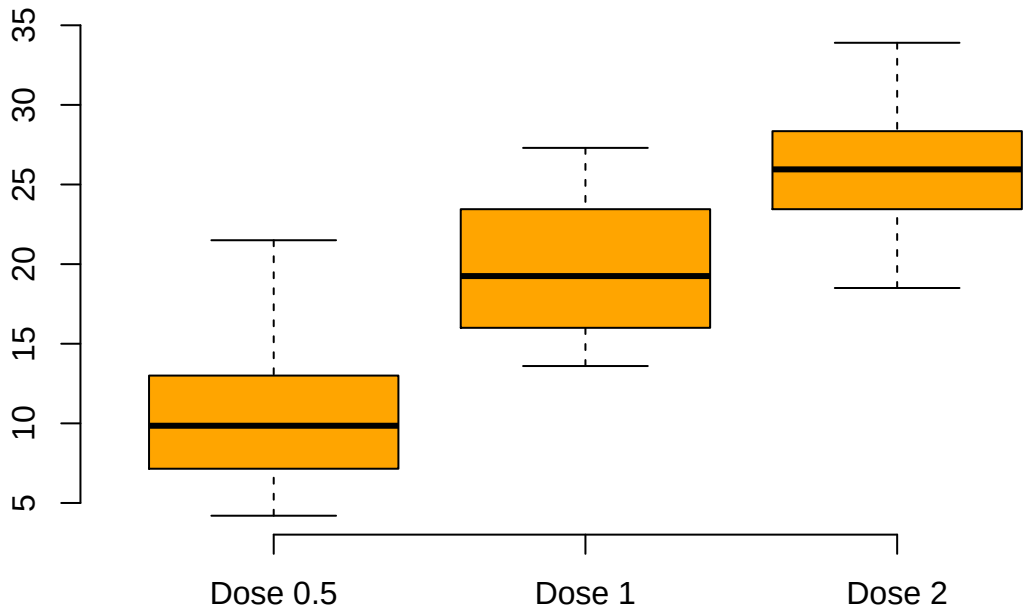
```
par(mfrow=c(2,2), mar=c(3,3,1,0)+.5, mgp=c(1.6,.6,0))
# Box plot of one variable
```

```r
boxplot(ToothGrowth$len,col="lightblue")
# Box plots by groups (dose)
# remove frame
boxplot(len ~ dose, data = ToothGrowth, frame = FALSE,col="pink")
# Horizontal box plots
boxplot(len ~ dose, data = ToothGrowth, frame = FALSE,
        horizontal = TRUE,col="green")
# Notched box plots
boxplot(len ~ dose, data = ToothGrowth, frame = FALSE,
        notch = TRUE,col="#C861D3")
```



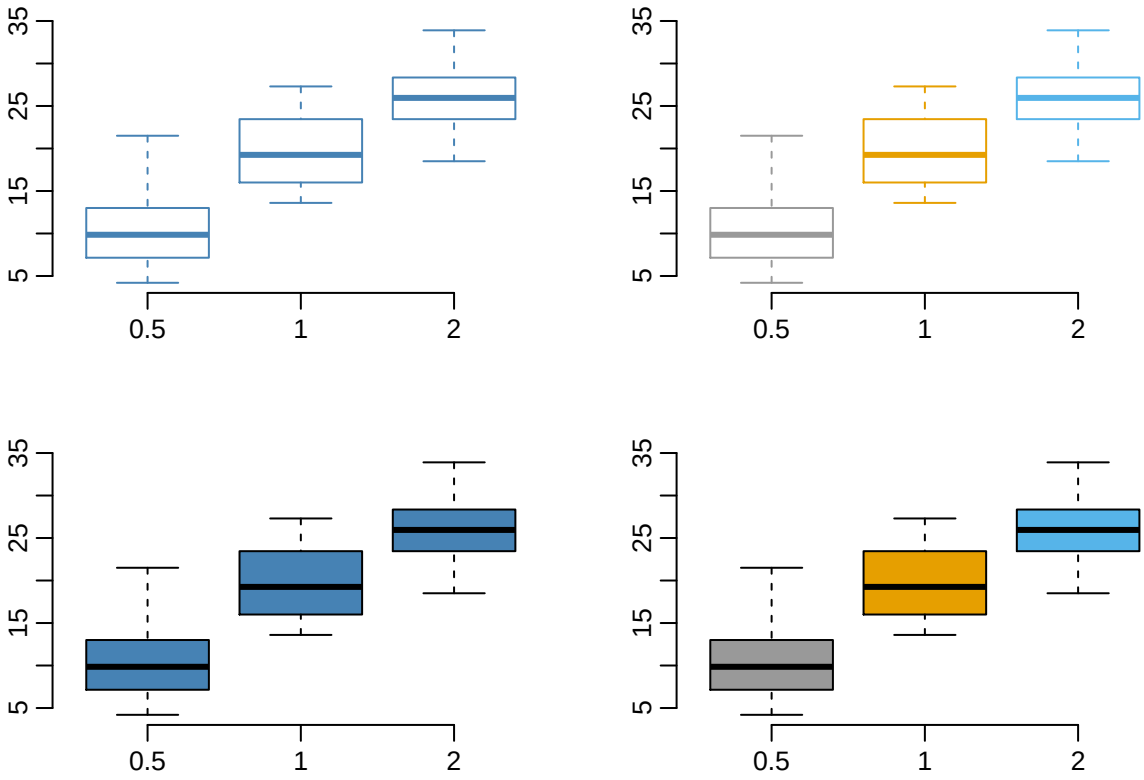
Change group names:

```r
boxplot(ToothGrowth$len ~ ToothGrowth$dose, data = ToothGrowth, frame = FALSE,
        names = c("Dose 0.5", "Dose 1", "Dose 2"),col="orange")
```



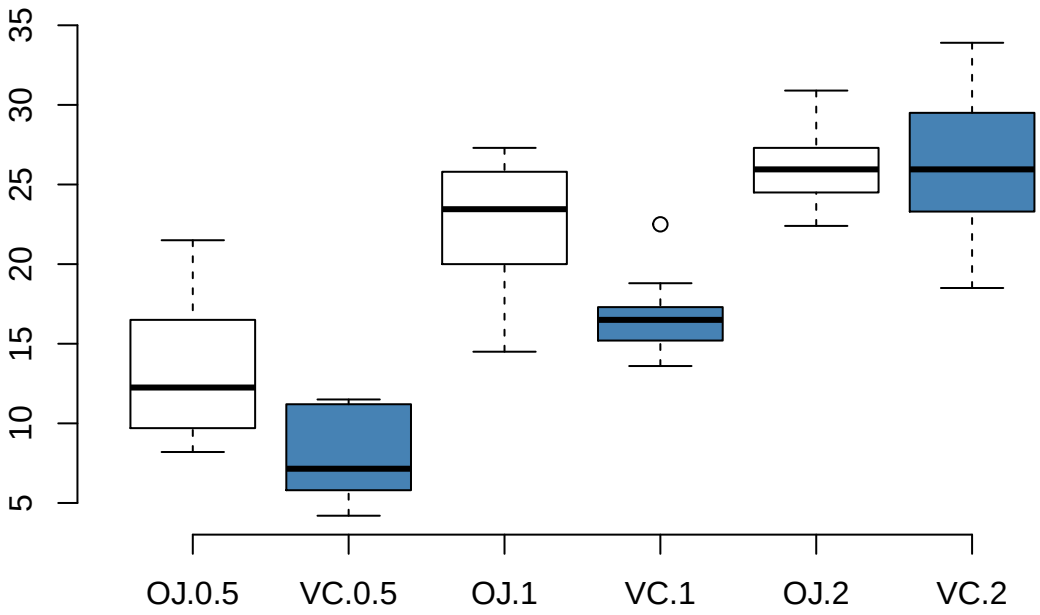
Change color:

```
par(mfrow=c(2,2), mar=c(3,3,1,0)+.5, mgp=c(1.6,.6,0))

# Change the color of border using one single color
boxplot(len ~ dose, data = ToothGrowth, frame = FALSE,
        border = "steelblue")
# Change the color of border.
#  Use different colors for each group
boxplot(len ~ dose, data = ToothGrowth, frame = FALSE,
        border = c("#999999", "#E69F00", "#56B4E9"))
# Change fill color : single color
boxplot(len ~ dose, data = ToothGrowth, frame = FALSE,
        col = "steelblue")
# Change fill color: multiple colors
boxplot(len ~ dose, data = ToothGrowth, frame = FALSE,
        col = c("#999999", "#E69F00", "#56B4E9"))
```



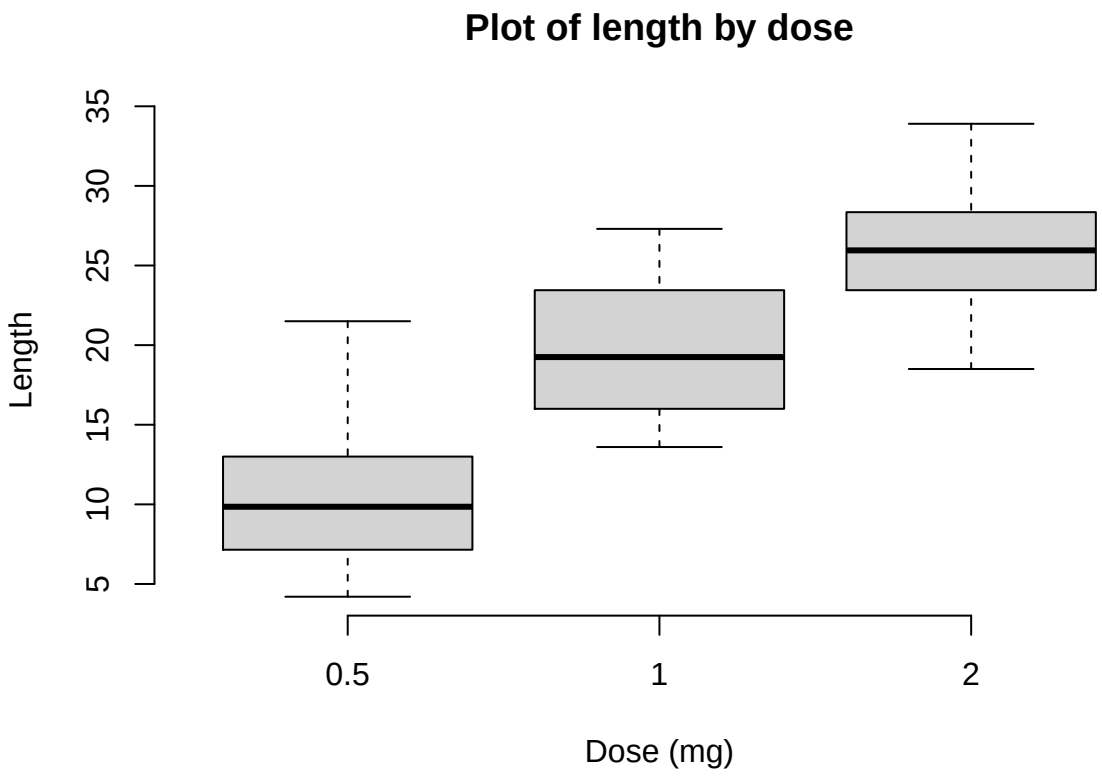
Multiple box plots:

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



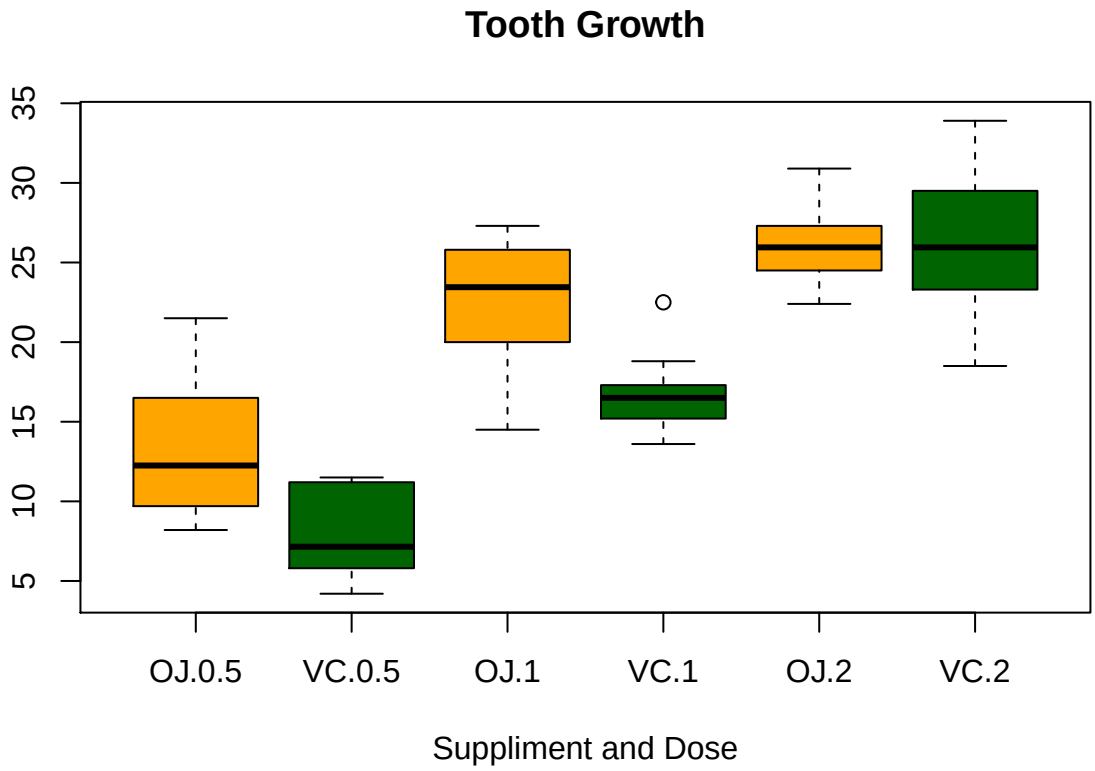
Main title and axis labels:

```
# Change axis titles
# Change color (col = "gray") and remove frame
# Create notched box plot
boxplot(len ~ dose, data = ToothGrowth,
        main = "Plot of length by dose",
        xlab = "Dose (mg)", ylab = "Length",
        col = "lightgray", frame = FALSE)
```



Notched Boxplot:

```
# Notched Boxplot of Tooth Growth Against 2 Crossed Factors
# boxes colored for ease of interpretation
boxplot(len~supp*dose, data=ToothGrowth, notch=FALSE,
  col=(c("orange","darkgreen")),
  main="Tooth Growth", xlab="Suppliment and Dose")
```

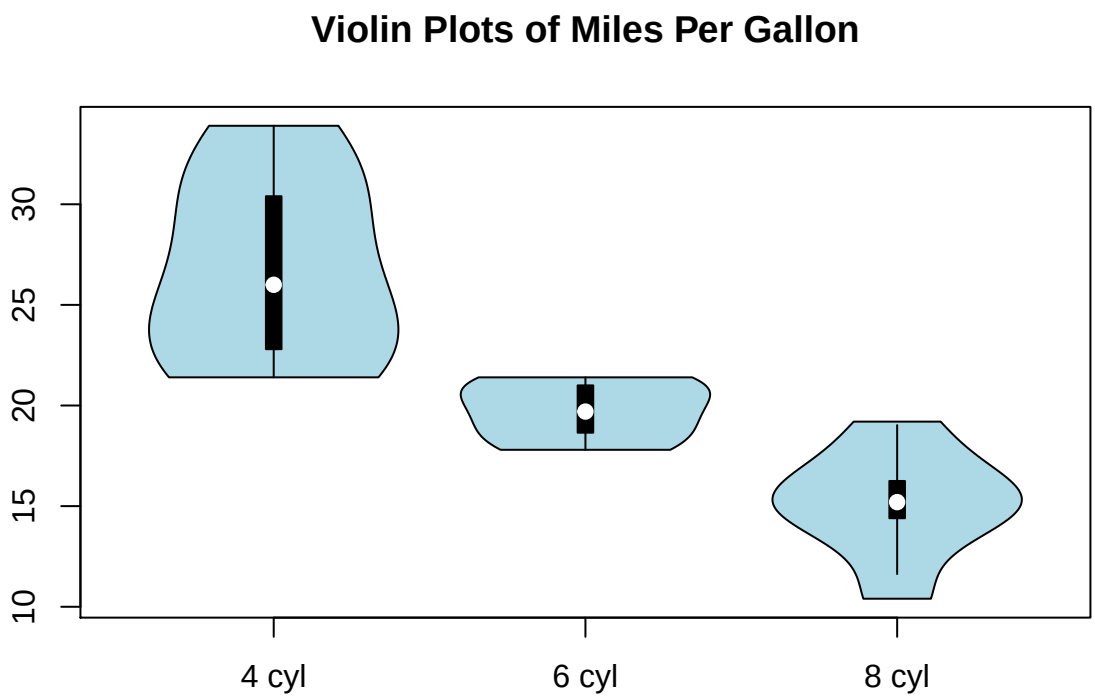


Additional box plots->

- `vioplot` function from **vioplot** package
- `bagplot(x, y)` function from **aplpack** package

Example-1:

```
# Violin Plots
#install.packages("vioplot")
library(vioplot)
x1 <- mtcars$mpg[mtcars$cyl==4]
x2 <- mtcars$mpg[mtcars$cyl==6]
x3 <- mtcars$mpg[mtcars$cyl==8]
vioplot(x1, x2, x3, names=c("4 cyl", "6 cyl", "8 cyl"),
   col="lightblue")
title("Violin Plots of Miles Per Gallon")
```

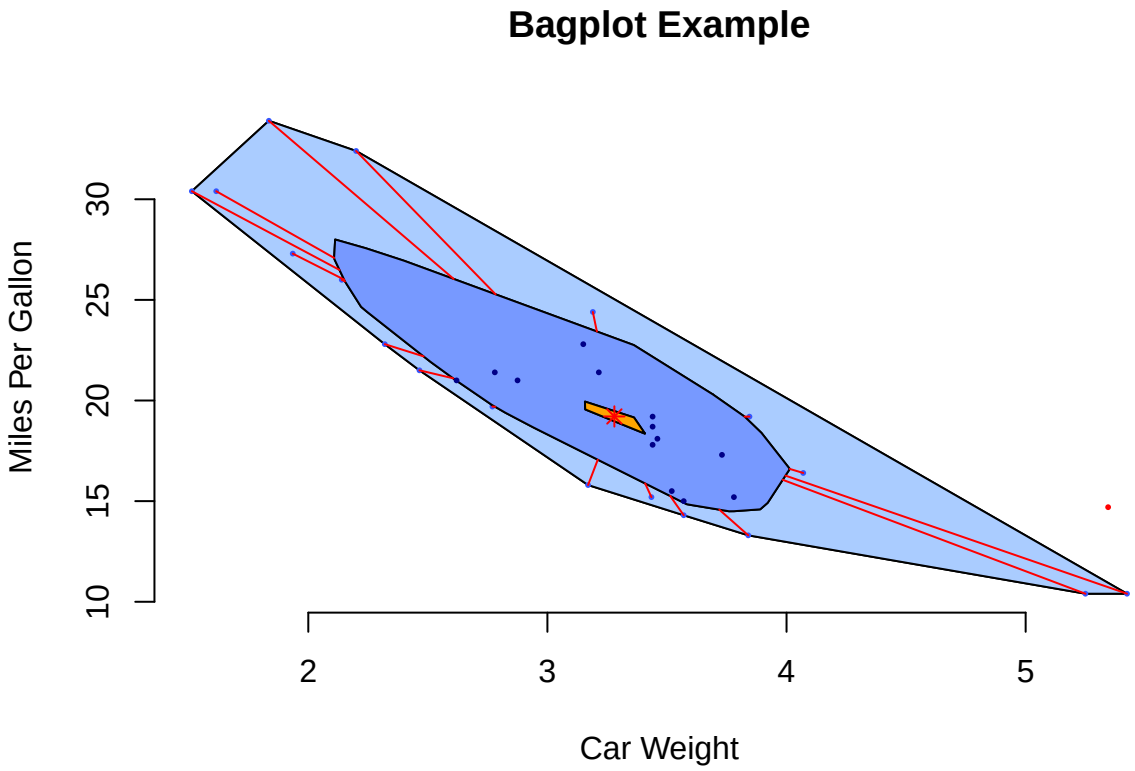


Example-2:

```
# Example of a Bagplot
#install.packages("aplpack")
library(aplpack)
```

```
## Loading required package: tcltk
```

```
attach(mtcars)
bagplot(wt,mpg, xlab="Car Weight", ylab="Miles Per Gallon",
  main="Bagplot Example")
```

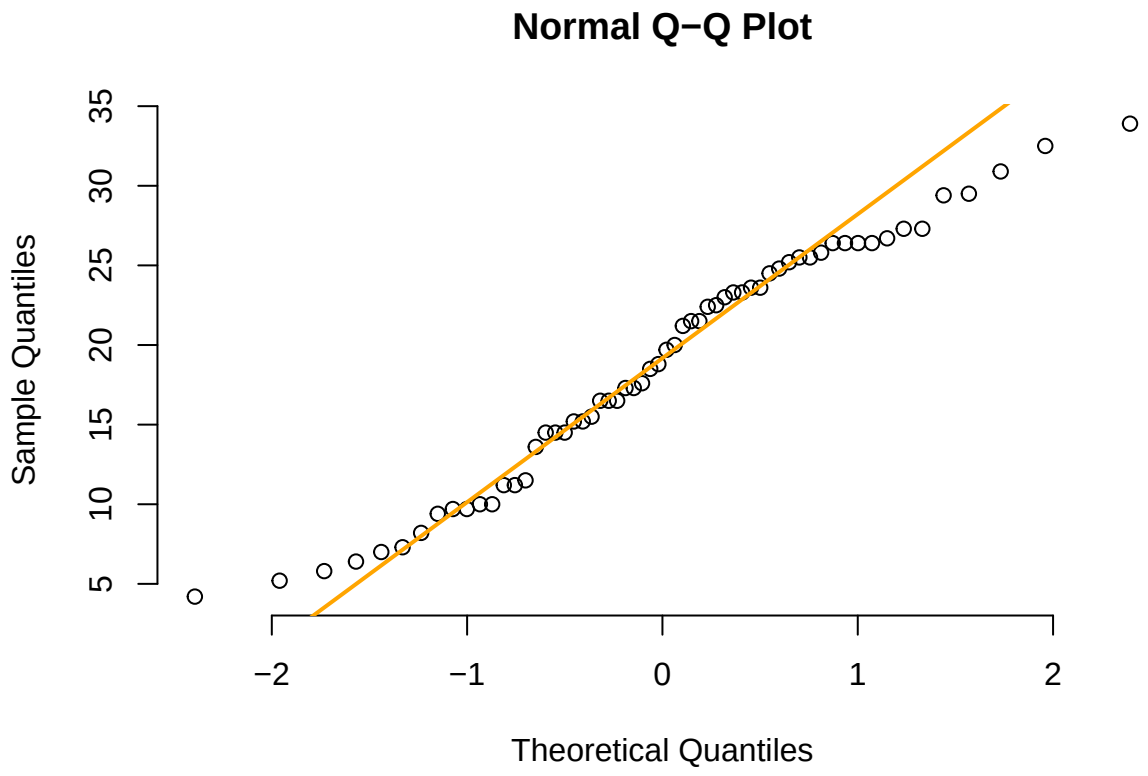

Bagplot Example
Miles Per Gallon
10
15
20
25
30
2
3
4
5
Car Weight

Chapter 13

QQ-Plots: Quantile-Quantile Plots

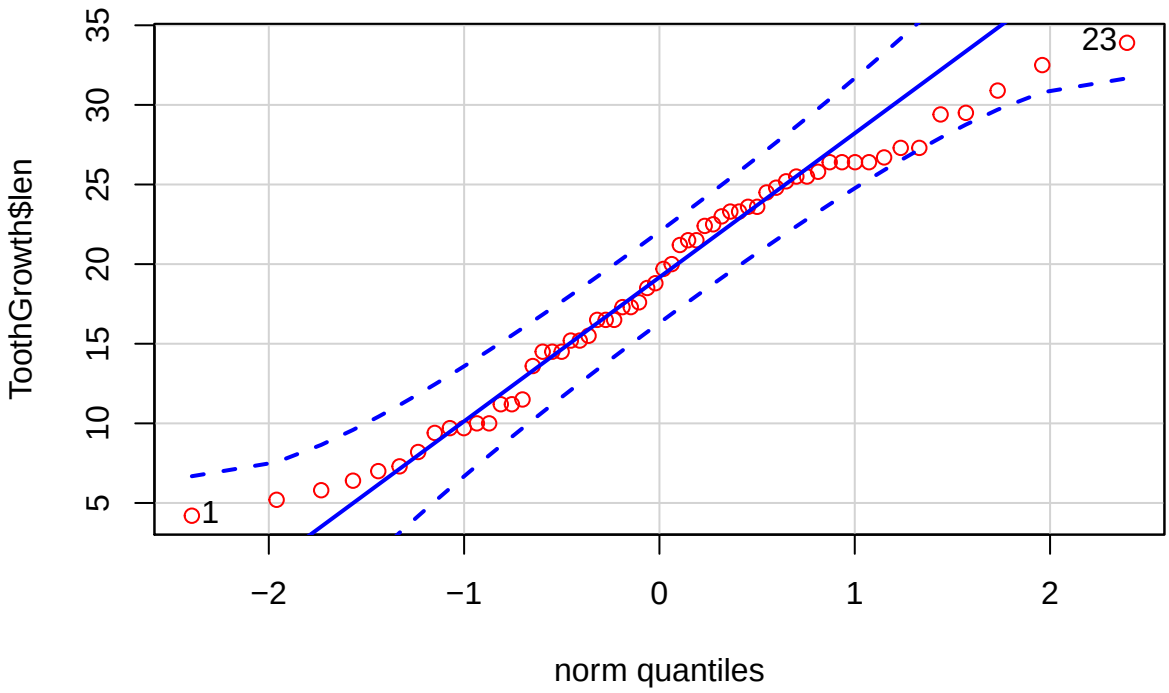
- qqnorm(): produces a normal QQ plot of the variable
- qqline(): adds a reference line

```
qqnorm(ToothGrowth$len, pch = 1, frame = FALSE)
qqline(ToothGrowth$len, col = "orange", lwd = 2)
```



Additional -> qqPlot() in car package:

```
#install.packages("car")
library("car")
qqPlot(ToothGrowth$len,col="red")
```




```
## [1] 23  1
```

Chapter 14

Means and Confidence Intervals

With `plotmeans()` function from **gplots** package

```
#install.packages("gplots")
library(gplots)
```

```
##
## Attaching package: 'gplots'
```

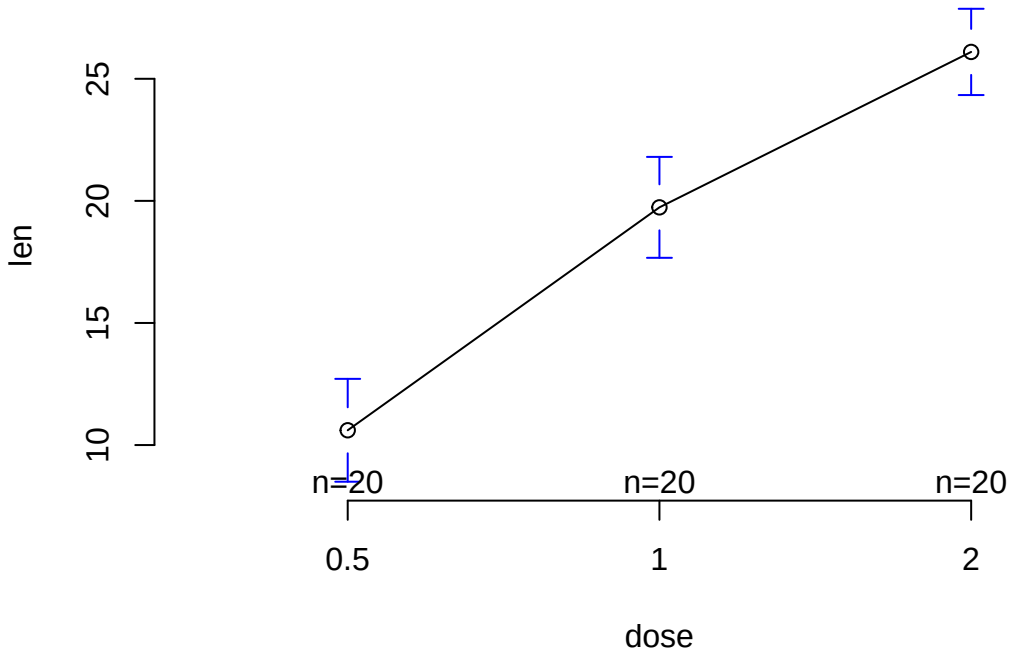
```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
# Plot the mean of teeth length by dose groups
plotmeans(len ~ dose, data = ToothGrowth, frame = FALSE)
```

```
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "frame" is not a
## graphical parameter
```

```
## Warning in axis(1, at = 1:length(means), labels = legends, ...): "frame" is
## not a graphical parameter
```

```
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "frame" is not a
## graphical parameter
```


len
10
15
20
25
n=20
n=20
n=20
0.5
1
2
dose