# Exploratory Data Analysis with R

## Introduction to R - Part III: R Base Plotting

Xuemao Zhang
East Stroudsburg University

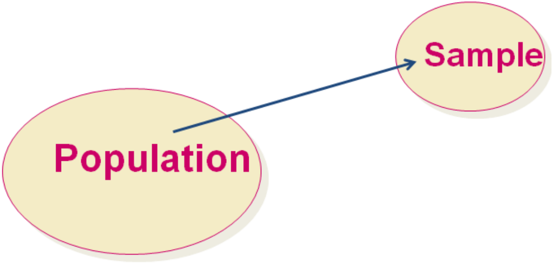
September 14, 2021

# Outline

- Sample and Population
- Types of data
    - Classification of variables and data
- R Graphics
    - Bar charts and Pie charts
    - Colors in R
    - Density plots: histograms and kernel density plots
    - Box plots
    - Scatterplots and Scatter plot matrix
    - Line charts
    - `par()` function
    - Saving a plot

# Sample and Population

- An investigation will typically focus on a well-defined collection of subjects constituting a population of interest.



- Population : The complete collection of all subjects that are being considered.
- Sample: Subcollection of subjects selected from a population.

# Types of data

- A data set is a collection of measurements of one variable or several variables for some individuals or subjects.
- Often, a data set is a file, in which each column (or field) corresponds to an variable(or attribute) and each row corresponds to measurements of all variables for each subject. This type of data sets is called record-based data.

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

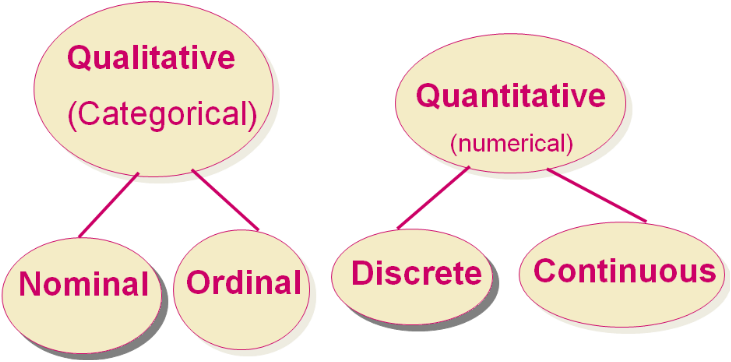
- There are other types of data sets.

# Types of data

- A **variable** (or attribute) is a property or characteristic that can vary from one subject to another or frome one time to another.
  - A data set is are obtained by measuring variables.
- Examples
  - Hair color
  - Body temperature
  - time to failure of a computer component.

# Types of data

- Classification of variables by the type of measurements



Qualitative (Categorical)

Nominal

Ordinal

Quantitative (numerical)

Discrete

Continuous

# Types of data - Categorical Variables

- **Categorical variables** take category or label/name values, and place an individual into one of several groups.
  - They cannot be used for computations.
- Categorical variables can be further classified using levels of measurement by looking at what is being measured.
  - **Nominal**, when there is no natural ordering among the categories.
    - Common examples would be gender, eye color, ethnicity or social security numbers.
  - **Ordinal**, when there is a natural order among the categories, such as, ranking scales or letter grades.
    - Examples: Course grades A, B, C, D, or F; Ranks Gold, Silver, Bronze.
    - However, ordinal variables are still categorical and do not provide precise measurements.
    - Differences between data values either cannot be determined or are meaningless.

# Types of data - Numerical Variables

- **Numerical variables** take numerical values, and represent some kind of measurement.
- Numerical variables are often further classified by the number of values:
  - **Discrete**, when the variable takes on a finite or countably infinite number of values.
    - Most often these variables indeed represent some kind of **count** such as the number of prescriptions an individual takes daily.
  - **Continuous**, when the variable takes infinitely many values corresponding to the points on a real line interval
    - Units should be provided.
    - Our precision in measuring these variables is often limited by our instruments.
    - Common examples would be height (inches), weight (pounds), or time to recovery (days).

# Types of data

**Univariate and Multivariate data:**

- A **univariate** data set consists of observations on a single variable.
  - For example, the following sample of lifetimes (hours) of brand D batteries put to a certain use is a numerical univariate data set:

    5.6 5.1 6.2 6.0 5.8 6.5 5.8 5.5

- We have **bivariate** data when observations are made on each of two variables.
  - Example: A data set consists of a (height, weight) pair for each basketball player on a team, with the first observation as (72, 168), the second as (75, 212), and so on.

- **Multivariate** data arises when observations are made on more than one variable (so bivariate is a special case of multivariate).

# Types of data

**Parameter and Statistic:**

- Parameter is a numerical summary describing some variable of a population.
  - For example, population mean $\mu$, population proportion $p$
- Statistic is a numerical summary describing some variable of a sample
  - A statistic is an estimator of some parameter in a population.
  - For example, sample mean $\bar{x}$, sample proportion $\hat{p}$

# R Graphics

- R has strong graphic capabilities. `plot()` is a generic function for plotting of R objects.
- There are many plot functions which are specific to some tasks.
- Titles, legends and annotations.
  - `main` gives the main title, `sub` the subtitle.
  - `legend()`. The position can be “bottomleft”, “bottomright”, “topleft”, “topright” or exact coordinates.
  - `xlab` specifies the X-axis label; `ylab` specifies the Y-axis label.
  - `xlim` specifies the range of the X-axis; `ylim` specifies the range of the Y-axis.
  - `mtext()` puts some texts in the margin. The margin can be at the bottom (1), the left (2), the top (3) or the right (4).
  - `text()`puts Text in the graph.
  - We can add mathematical symbols using `expression()`.
  - The type of a plot can be : `n`(none), `p`(points), `l`(lines) etc.
  - For more information, type `?title` and `?text` in R console.

# Bar chart

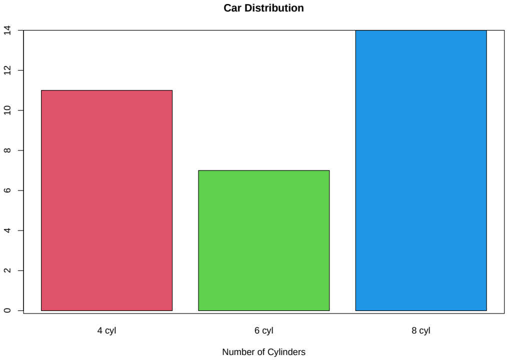
- Bar chart is for categorical data.
- `barplot()` specifies the height of each bar and (optionally) a vector of labels for each bar.

```
mtcars$cyl = as.factor(mtcars$cyl)  # convert cyl to a factor
counts= table(mtcars$cyl)
counts;  #get the count of 4, 6 & 8 cylinder cars
```

```
##
##  4  6  8
## 11  7 14
```

# Bar chart

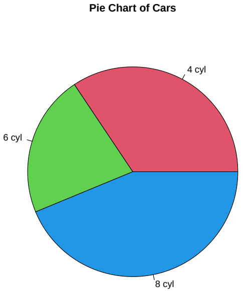
```
barplot(counts,main="Car Distribution", xlab="Number of Cylinders",
        col=c(2,3,4),names.arg=c("4 cyl", "6 cyl", "8 cyl"))
# col is used to specified colors
box() #draw a box around the plot
```



# Pie charts

- Pie charts are for categorical data.
- A pie chart presents each category as a slice of a circle so that each slice has a size that is proportional to the whole in each category.
- Pie charts are not recommended since people are able to judge height more accurately than area.

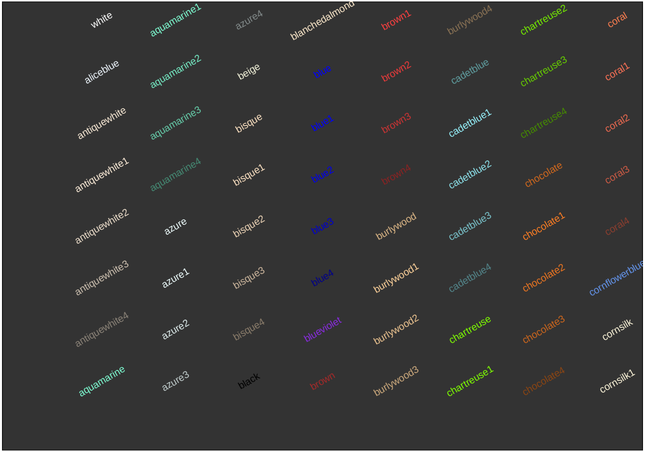
```
counts= table(mtcars$cyl) #create a frequency table
pie(counts,main="Pie Chart of Cars",
        col=c(2,3,4), labels=c("4 cyl", "6 cyl", "8 cyl"))
```



# Colors in R

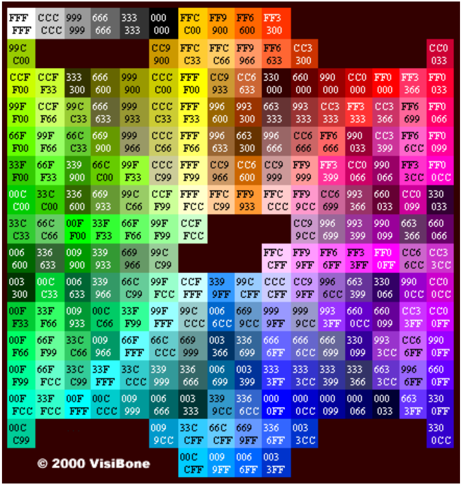
- In R, colors can be specified either by name (e.g col = “red”), integers or hexadecimal colors (#rrggbb). The following shows the first sixty-four color names. See more in the R color cheat sheet or get the colors using the R command

```
colors()
```



# Colors in R

- Colors can be specified using hexadecimal color code, #rrggbb, where rr, gg, and bb refer to color intensity in the red, green, and blue channels, respectively. For more information, see https://stat545.com/colors.html.



| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FFF<br>FFF | CCC<br>CCC | 999<br>999 | 666<br>666 | 333<br>333 | 000<br>000 | FFC<br>C00 | FF9<br>900 | FF6<br>600 | FF3<br>300 | | | | | | |
| 99C<br>C00 | | | | | CC9<br>900 | FFC<br>C33 | FFC<br>C66 | FF9<br>966 | FF6<br>633 | CC3<br>300 | | | | | CC0<br>033 |
| CCF<br>F00 | CCF<br>F33 | 333<br>300 | 666<br>600 | 999<br>900 | CCC<br>C00 | FFF<br>F00 | CC9<br>933 | CC6<br>633 | 330<br>000 | 660<br>000 | 990<br>000 | CC0<br>000 | FF0<br>000 | FF3<br>366 | FF0<br>033 |
| 99F<br>F00 | CCF<br>F66 | 99C<br>C33 | 666<br>633 | 999<br>933 | CCC<br>C33 | FFF<br>F33 | 996<br>600 | 993<br>300 | 663<br>333 | 993<br>333 | CC3<br>333 | FF3<br>333 | CC3<br>366 | FF6<br>699 | FF0<br>066 |
| 66F<br>F00 | 99F<br>F66 | 66C<br>C33 | 669<br>900 | 999<br>966 | CCC<br>C66 | FFF<br>F66 | 996<br>633 | 663<br>300 | 996<br>666 | CC6<br>666 | FF6<br>666 | 990<br>033 | CC3<br>399 | FF6<br>6CC | FF0<br>099 |
| 33F<br>F00 | 66F<br>F33 | 339<br>900 | 66C<br>C00 | 99F<br>F33 | CCC<br>C99 | FFF<br>F99 | CC9<br>966 | CC6<br>600 | CC9<br>999 | FF9<br>999 | FF3<br>399 | CC0<br>066 | 990<br>066 | FF3<br>3CC | FF0<br>0CC |
| 00C<br>C00 | 33C<br>C00 | 336<br>600 | 669<br>933 | 99C<br>C66 | CCF<br>F99 | FFF<br>FCC | FFC<br>C99 | FF9<br>933 | FFC<br>CCC | FF9<br>9CC | CC6<br>699 | 993<br>366 | 660<br>033 | CC0<br>099 | 330<br>033 |
| 33C<br>C33 | 66C<br>C66 | 00F<br>F00 | 33F<br>F33 | 66F<br>F66 | 99F<br>F99 | CCF<br>FCC | | | | CC9<br>9CC | 996<br>699 | 993<br>399 | 990<br>099 | 663<br>366 | 660<br>066 |
| 006<br>600 | 336<br>633 | 009<br>900 | 339<br>933 | 669<br>966 | 99C<br>C99 | | | | FFC<br>CFF | FF9<br>9FF | FF6<br>6FF | FF3<br>3FF | FF0<br>0FF | CC6<br>6CC | CC3<br>3CC |
| 003<br>300 | 00C<br>C33 | 006<br>633 | 339<br>966 | 66C<br>C99 | 99F<br>FCC | CCF<br>FFF | 339<br>9FF | 99C<br>CFF | CCC<br>CFF | CC9<br>9FF | 996<br>6CC | 663<br>399 | 330<br>066 | 990<br>0CC | CC0<br>0CC |
| 00F<br>F33 | 33F<br>F66 | 009<br>933 | 00C<br>C66 | 33F<br>F99 | 99F<br>FFF | 99C<br>CCC | 006<br>6CC | 669<br>9CC | 999<br>9FF | 999<br>9CC | 993<br>3FF | 660<br>0CC | 660<br>099 | CC3<br>3FF | CC0<br>0FF |
| 00F<br>F66 | 66F<br>F99 | 33C<br>C66 | 009<br>966 | 66F<br>FFF | 66C<br>CCC | 669<br>999 | 003<br>366 | 336<br>699 | 666<br>6FF | 666<br>6CC | 666<br>699 | 330<br>099 | 993<br>3CC | CC6<br>6FF | 990<br>0FF |
| 00F<br>F99 | 66F<br>FCC | 33C<br>C99 | 33F<br>FFF | 33C<br>CCC | 339<br>999 | 336<br>666 | 006<br>699 | 003<br>399 | 333<br>3FF | 333<br>3CC | 333<br>399 | 333<br>366 | 663<br>3CC | 996<br>6FF | 660<br>0FF |
| 00F<br>FCC | 33F<br>FCC | 00F<br>FFF | 00C<br>CCC | 009<br>999 | 006<br>666 | 003<br>333 | 339<br>9CC | 336<br>6CC | 000<br>0FF | 000<br>0CC | 000<br>099 | 000<br>066 | 000<br>033 | 663<br>3FF | 330<br>0FF |
| 00C<br>C99 | | | | | 009<br>9CC | 33C<br>CFF | 66C<br>CFF | 669<br>9FF | 336<br>6FF | 003<br>3CC | | | | | 330<br>0CC |
| | | | | | | 00C<br>CFF | 009<br>9FF | 006<br>6FF | 003<br>3FF | | | | | | |

© 2000 VisiBone

# Colors in R

- In R you can call colors by their numbers. The `palette()` function within the `grDevices` library allows a table of colors to be referenced by a numeric index. The default color palette is

```
palette()
```

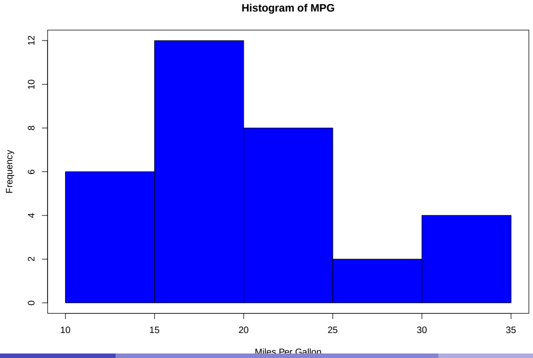
```
1 = "black"
2 = "red"
3 = "green"
4 = "blue"
5 = "cyan"
6 = "magenta"
7 = "yellow"
8 = "gray"
```

- To set these colors as parameters, simply use the index.

# Histograms and kernel density plots

- Histogram is for numerical data.
- A histogram shows a partition of a data set and the number of observations in each class.

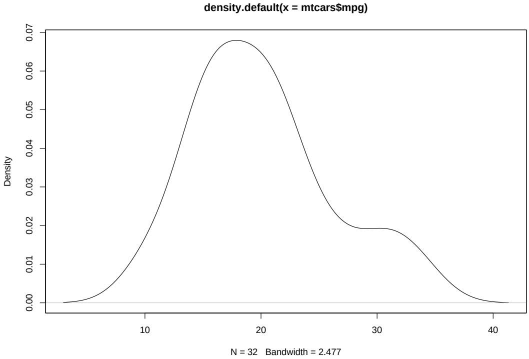
```
hist(mtcars$mpg,  col="blue", xlab="Miles Per Gallon",#breaks = 10,
    main="Histogram of MPG")
#we can use breaks argument to determine the number of bins
box()
```



# Histograms and kernel density plots

- In general, we assume that a data set is from a larger population.
- Kernel density is an **estimate** of the distribution of the variable.

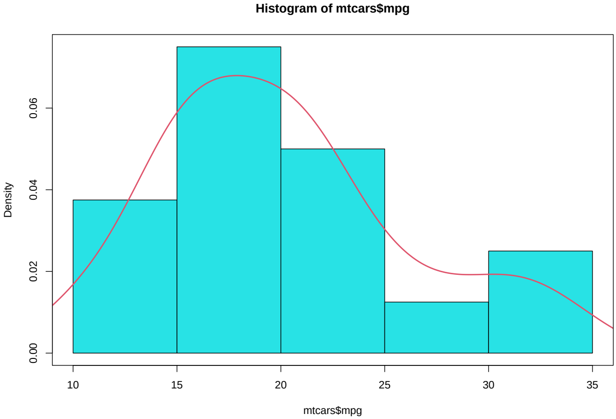
```
d = density(mtcars$mpg) # returns the density data
plot(d) # plots the results
box()
```



# Histograms and kernel density plots

- Relative frequency histogram and density plot

```
hist(mtcars$mpg, col=5,prob=TRUE)#show relative frequencies
lines(density(mtcars$mpg),col=2,lwd=2) # density plot
box()
```

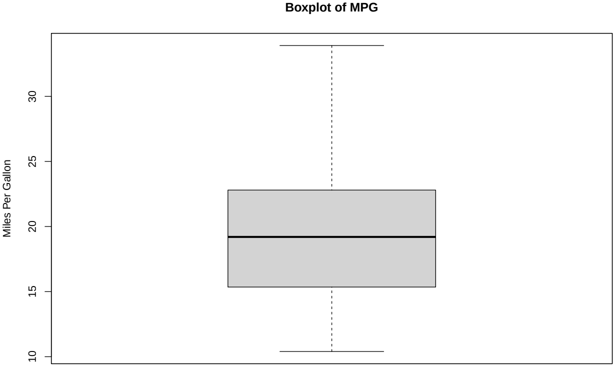


# Box plot

- A box plot (or box and whisker plot) shows the five number summary: Min, $Q_1$,Median, $Q_3$, Max and outliers
  - Min is the minimum value in the data.
  - 25% observations in the sorted data are less than $Q_1$.
  - Mdeian is the absolute center.
  - 75% observations in the sorted data are less than $Q_3$.
  - $Q_3 - Q_1$ is called IQR (Interquartile Range).
  - Max is the maximum value in the data.
  - Outliers are data points far away from other data values.
  - Any values less than $Q1 - 1.5 \times IQR$ or greater than $Q3 + 1.5 \times IQR$ are defined as outliers.
  - The whiskers extend only as far as the minimum data value that is not an outlier and the maximum data value that is not an outlier.
- Boxplots can be created for individual variables or for variables by group using the function `boxplot`.

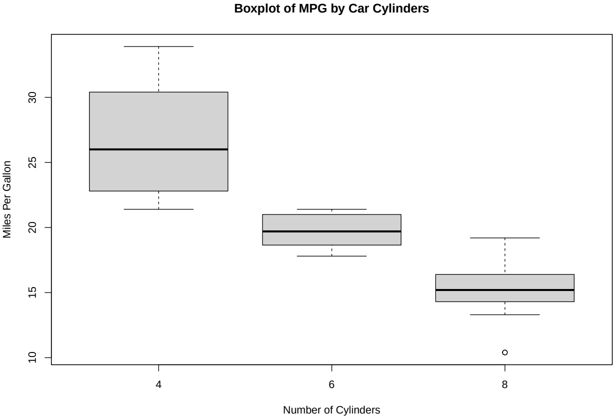
# Box plot

```
boxplot(mtcars$mpg, main="Boxplot of MPG",
        ylab="Miles Per Gallon") # Boxplot of MPG
box()
```



# Box plot

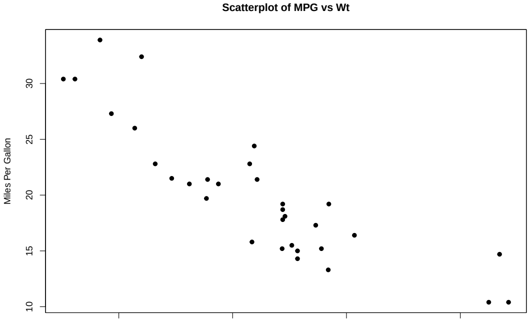
```
boxplot(mpg~cyl,data=mtcars,main="Boxplot of MPG by Car Cylinders",
    xlab="Number of Cylinders", ylab="Miles Per Gallon")
box()
```



# Scatter plot

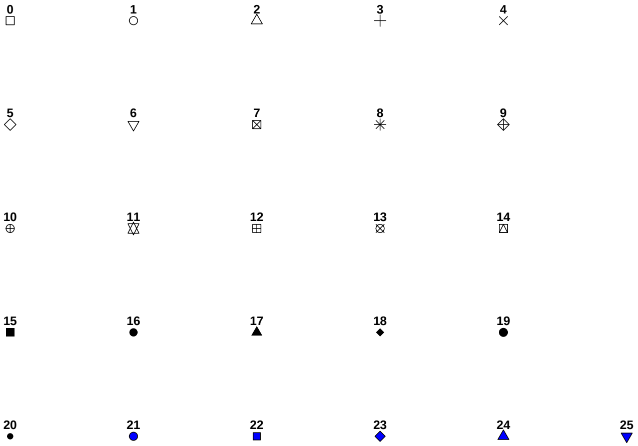
- A scatter plot is used to show the relationship between two variables.
- Each data point with two measurements is plotted on the Cartesian (x,y) plane.

```
plot(mtcars$wt, mtcars$mpg,
     main="Scatterplot of MPG vs Wt",
    xlab="Car Weight", ylab="Miles Per Gallon", pch=19)
#pch is used to specify symbols to use;
#type ?pch or ?points for more information
box()
```



# Scatter plot

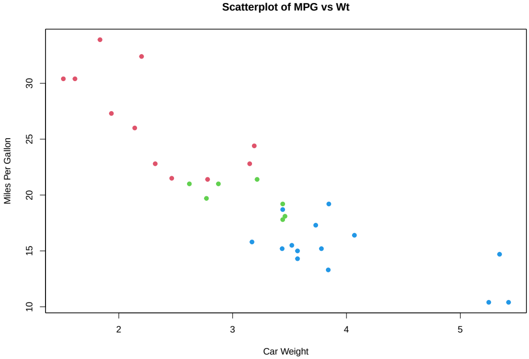
- Points shapes (?pch) available in R:



# Scatter plot

- Colors can be specified for different `cyl`

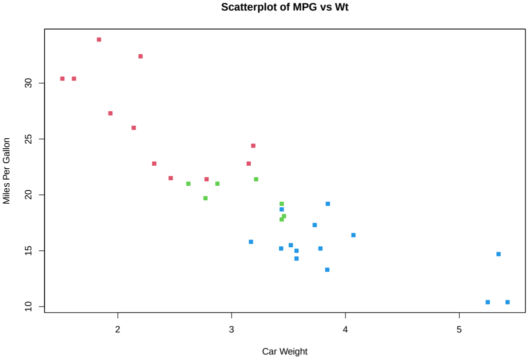
```
plot(mtcars$wt, mtcars$mpg, col=c(2,3,4)[mtcars$cyl],
     main="Scatterplot of MPG vs Wt",
    xlab="Car Weight", ylab="Miles Per Gallon", pch=19)
#pch is used to specify symbols to use;
#type ?pch or ?points for more information
box()
```



# Scatter plot

- Fill color for pch = 21:25.
  - bg fills color for the open plot symbols when pch = 21:25.

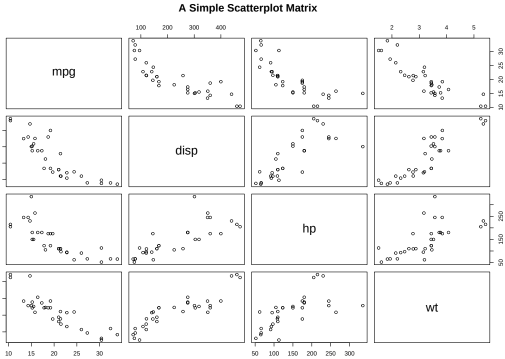
```
plot(mtcars$wt, mtcars$mpg, col=c(2,3,4)[mtcars$cyl],
     main="Scatterplot of MPG vs Wt",
    xlab="Car Weight", ylab="Miles Per Gallon", pch=22,
    bg=c(2,3,4)[mtcars$cyl])
box()
```



# Scatter plot matrix

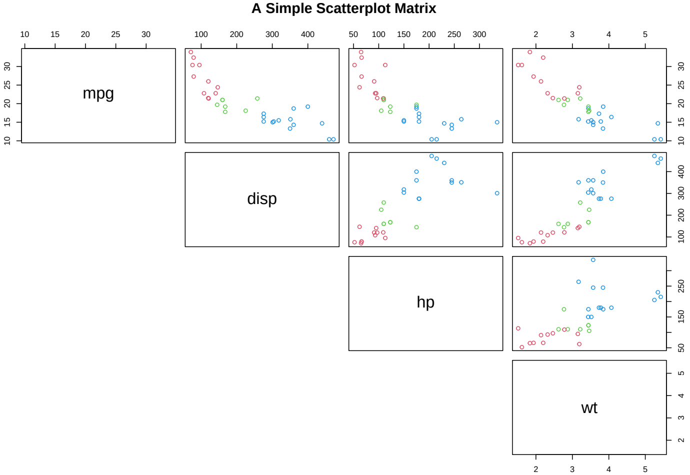
- To see the relationship between any two variables, it is useful to look at the scatter plot matrix.
- `pairs()` function creates beautiful scatter plot matrix.

```
pairs(~mpg+disp+hp+wt,data=mtcars,
   main="A Simple Scatterplot Matrix")
```



# Scatter plot matrix

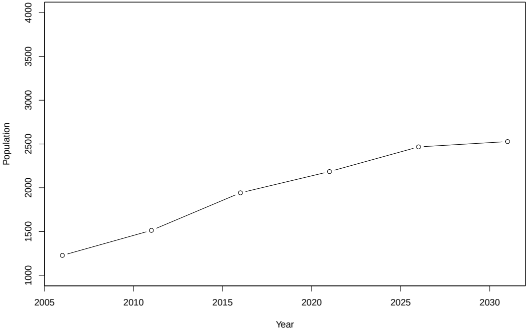
```
pairs(~mpg+disp+hp+wt,data=mtcars, col=c(2,3,4)[mtcars$cyl],
   main="A Simple Scatterplot Matrix",lower.panel = NULL)
```



# Line chart

- Line chart is just a scatter plot by specifying specify type = “b” for points joined line or type = “l” for line.

```
y = c(1227.3, 1513.1, 1942.1, 2184.7, 2466.6, 2527.6)
x = c(2006, 2011, 2016, 2021, 2026, 2031)
plot(x,y, type="b", xlab="Year", ylab="Population",
     xlim=c(2006, 2031), ylim=c(1000, 4000))
box()
```



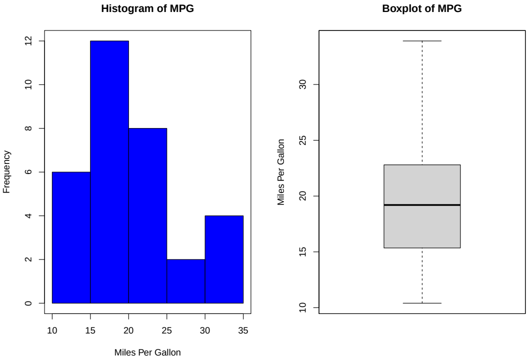
# par() function

- par() defines the default settings for plots such as fonts, colors, axes, titles.
- The format is par(optionname=value, optionname=value, ...)
- If you set parameter values using par(), the changes will be in effect for the rest of the session or until you change them again.
- type ?par in R console to see more information.
- Especially, par() can be used to put multiple graphs in a single plot. The syntax to set the plotting area into a $m \times n$ matrix is

**par(mfrow=c(m,n))**

# par() function

```
par(mfrow=c(1,2))
hist(mtcars$mpg,  col="blue", xlab="Miles Per Gallon",#breaks = 10,
    main="Histogram of MPG")
box()
boxplot(mtcars$mpg, main="Boxplot of MPG",
        ylab="Miles Per Gallon") # Boxplot of MPG
box()
```



# Saving a plot

- All graphs we plot in R programming are displayed on the screen by default.
- The graphs can be saved manually.
- We can save plots as a file on disk with the help of built-in functions.
- We need to call the function `dev.off()` after all the plotting, to save the file and return control to the screen.

# Saving a plot

- To save a plot as jpeg image we need the jpeg() function.

```r
#Save the plot in the working directory
jpeg(file="scatterplot1.jpeg")
plot(mtcars$wt, mtcars$mpg, main="Scatterplot of MPG vs Wt",
    xlab="Car Weight", ylab="Miles Per Gallon", pch=19)
box()
dev.off()
```

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

# Saving a plot

- To save a plot as png image we need the png() function.

```
png(file="scatterplot2.png")
plot(mtcars$wt, mtcars$mpg, main="Scatterplot of MPG vs Wt",
    xlab="Car Weight", ylab="Miles Per Gallon", pch=19)
box()
dev.off()
```

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

# Saving a plot

- To save a plot as bmp image we need the bmp() function.

```
bmp(file="scatterplot3.bmp")
plot(mtcars$wt, mtcars$mpg, main="Scatterplot of MPG vs Wt",
    xlab="Car Weight", ylab="Miles Per Gallon", pch=19)
box()
dev.off()
```

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

# Saving a plot

- To save a plot as tiff format we need the `tiff()` function.

```r
tiff(file="scatterplot4.tiff")
plot(mtcars$wt, mtcars$mpg, main="Scatterplot of MPG vs Wt",
    xlab="Car Weight", ylab="Miles Per Gallon", pch=19)
box()
dev.off()
```

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

# Saving a plot

- We can save our plots as **vector image** in pdf or postscript formats.
- The beauty of vector image is that it is easily resizable. Zooming on the image will not compromise its quality.
- To save a plot as pdf format we need the `pdf()` function.

```
pdf(file="scatterplot5.pdf")
plot(mtcars$wt, mtcars$mpg, main="Scatterplot of MPG vs Wt",
    xlab="Car Weight", ylab="Miles Per Gallon", pch=19)
box()
dev.off()
```

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

# Saving a plot

- To save a plot as ps(postscript) or eps(encapsulated postscript) format we need the `postscript()` function.

```
postscript(file="scatterplot6.eps")
plot(mtcars$wt, mtcars$mpg, main="Scatterplot of MPG vs Wt",
    xlab="Car Weight", ylab="Miles Per Gallon", pch=19)
box()
dev.off()
```

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

# License