BUSAD40 - Busniess Statistics

Lecture Note 1

Norman Lo

Spring 2021

Can one be a good data analyst without being a half-good programmer? The short answer to that is, 'No.' The long answer to that is, 'No.'

— Frank Harrell, 1999 S-PLUS User Conference, New Orleans (October 1999), quoted by the R function fortune in the CRAN package fortunes

I. What's Statistics?

The term statistics can refer to numerical facts such as averages, medians, percentages, and maximums that help us understand a variety of business and economic situations.

Statistics can also refer to the art and science of collecting, analyzing, presenting, and interpreting data.

Applications in Business and Economics

Accounting

Public accounting firms use statistical sampling procedures when conducting audits for their clients.

Economics

Economists use statistical information in making forecasts about the future of the economy or some aspect of it.

Finance

Financial advisors use price-earnings ratios and dividend yields to guide their investment advice.

Marketing

Electronic point-of-sale scanners at retail checkout counters are used to collect data for a variety of marketing research applications.

Production

A variety of statistical quality control charts are used to monitor the output of a production process.

Information Systems

A variety of statistical information helps administrators assess the performance of computer networks.

Data and Data Sets

Data are the facts and figures collected, analyzed, and summarized for presentation and interpretation. All the data collected in a particular study are referred to as the data set for the study.

Categorical and Quantitative Data

Data can be classified as being categorical or quantitative. The statistical analysis that is appropriate depends on whether the data for the variable are categorical or quantitative. In general, there are more alternatives for statistical analysis when the data are quantitative.

Categorical Data

- Labels or names are used to identify an attribute of each element
- Often referred to as qualitative data
- Use either the nominal or ordinal scale of measurement
- Can be either numeric or nonnumeric
- Appropriate statistical analyses are rather limited

Quantitative Data

- Quantitative data indicate how many or how much.
- Quantitative data are always numeric.
- Ordinary arithmetic operations are meaningful for quantitative data.

Scales of Measurement

Scales of measurement include

Nominal: Data are labels or names used to identify an attribute of the element. A nonnumeric label or numeric code may be used.

Example:

Students at a university are classified by the school in which they are enrolled using a nonnumeric label such as Business, Humanities, Education, and so on. Alternatively, a numeric code could be used for the school variable (e.g. 1 denotes Business, 2 denotes Humanities, 3 denotes Education, and so on).

Ordinal: The data have the properties of nominal data and the order or rank of the data is meaningful. A nonnumeric label or numeric code may be used.

Example:

Students at a university are classified by their class standing using a nonnumeric label such as Freshman, Sophomore, Junior, or Senior. Alternatively, a numeric code could be used for the class standing variable (e.g. 1 denotes Freshman, 2 denotes Sophomore, and so on).

Interval: The data have the properties of ordinal data, and the interval between observations is expressed in terms of a fixed unit of measure. Interval data are always numeric.

Example:

Melissa has an SAT score of 1985, while Kevin has an SAT score of 1880. Melissa scored 105 points more than Kevin.

Ratio: Data have all the properties of interval data and the ratio of two values is meaningful. Ratio data are always numerical. Zero value is included in the scale.

Example: Price of a book at a retail store is 200 dollars, while the price of the same book sold online is 100 dollars. The ratio property shows that retail stores charge twice the online price.

Note: The scale determines the amount of information contained in the data. The scale indicates the data summarization and statistical analyses that are most appropriate.

Statistical Studies

Observational

In observational (nonexperimental) studies no attempt is made to control or influence the variables of interest.

Example:

Survey studies of smokers and nonsmokers are observational studies because researchers do not determine or control who will smoke and who will not smoke.

Experimental

In experimental studies the variable of interest is first identified. Then one or more other variables are identified and controlled so that data can be obtained about how they influence the variable of interest.

The largest experimental study ever conducted is believed to be the 1954 Public Health Service experiment for the Salk polio vaccine. Nearly two million U.S. children (grades 1- 3) were selected.

Two Main Branches of Statistics

Descriptive Statistics

Most of the statistical information in newspapers, magazines, company reports, and other publications consists of data that are summarized and presented in a form that is easy to understand. Such summaries of data, which may be tabular, graphical, or numerical, are referred to as descriptive statistics.

Example:

The manager of Hudson Auto would like to have a better understanding of the cost of parts used in the engine tune-ups performed in her shop.

Statistical Inference

The process of using data analysis to deduce properties of an underlying distribution of probability. Inferential statistical analysis infers properties of a population, for example by testing hypotheses and deriving estimates.

Example:

The data analyst of Microsoft would like to know if people would return to their web site with a deep blue color background or light blue color background.

II. Descriptive Statistic in R

Objectives:

1. Measures of Central Tendency and Variability of the Data
2. Summarizing Data for a Categorical Variable: Categorical data use labels or names to identify categories of like items.
3. Summarizing Data for a Quantitative Variable: Quantitative data are numerical values that indicate how much or how many.
4. Summarizing Data for Two Variables Using Tables and Graphical Displays

Setting Up the Working Directory

Before starting any project in R, it's important to locate your current working directory and setting the correct directory. It helps you find the R script file and loading the data more effectively.

```
# Identify your current working directory
getwd()
```

```
## [1] "C:/Users/lokma/Desktop/Teaching/Saint_Mary's/BUSAD40_FA2020/ClassNotes"
```

```
# Setting the working directory for your project or work
# setwd("Enter your working directory here!")
```

Loading the Data

We are using the **lsr** library for the demo. To install the package to your machine, you can use the command **install.packages()** and put **"lsr"** inside the brackets. Once the package is installed, you can call the package in R with the command **library()**. The data we are using is inside the data folder and you can load the Rdata file with the command **load** and include the path to the file. Once the data is loaded into R, we can start working on the data set.

```
# Import the lsr library
# install.packages("lsr")
library(lsr)

# Loading the Australian Football League (AFL) data set
load("data/aflsmall.Rdata")

# Check the variables in the file
who()
```

```
##    -- Name --      -- Class --   -- Size --
##    afl.finalists   factor        400
##    afl.margins     numeric       176
```

```
# Print the value of margins
print(afl.margins)
```

```
##   [1]  56  31  56   8  32  14  36  56  19   1   3 104  43  44  72   9  28  25
##  [19]  27  55  20  16  16   7  23  40  48  64  22  55  95  15  49  52  50  10
##  [37]  65  12  39  36   3  26  23  20  43 108  53  38   4   8   3  13  66  67
##  [55]  50  61  36  38  29   9  81   3  26  12  36  37  70   1  35  12  50  35
##  [73]   9  54  47   8  47   2  29  61  38  41  23  24   1   9  11  10  29  47
##  [91]  71  38  49  65  18   0  16   9  19  36  60  24  25  44  55   3  57  83
## [109]  84  35   4  35  26  22   2  14  19  30  19  68  11  75  48  32  36  39
## [127]  50  11   0  63  82  26   3  82  73  19  33  48   8  10  53  20  71  75
## [145]  76  54  44   5  22  94  29   8  98   9  89   1 101   7  21  52  42  21
## [163] 116   3  44  29  27  16   6  44   3  28  38  29  10  10
```

Measures of Central Tendency

One of the most important measure in the study of statistics is the **Central Tendency**. Often, we measure the central tendency of the data by either it's **mean**, **median**, or **mode**. If the measures are computed for data from a sample, they are called sample **statistics**. If the measures are computed for data from a population, they are called population **parameters**. A sample statistic is referred to as the point estimator of the corresponding population parameter.

Mean

The mean provides a measure of central location. The mean of a data set is the average of all the data values. The sample mean x bar is the point estimator of the population mean μ.

Median

The median of a data set is the value in the middle when the data items are arranged in ascending order. Whenever a data set has extreme values, the median is the preferred measure of central location. The median is the measure of location most often reported for annual income and property value data. A few extremely large incomes or property values can inflate the mean.

Mode

The mode of a data set is the value that occurs with greatest frequency. The greatest frequency can occur at two or more different values. If the data have exactly two modes, the data are bimodal. If the data have more than two modes, the data are multimodal.

```
# Calculate the mean value in R
mean(afl.margins)
```

```
## [1] 35.30114
```

```
# Find the median value in R
median(afl.margins)
```

```
## [1] 30.5
```

```
# Find the mode value in R (Core R has no function for mode)
# Option 1:
table(afl.finalists)
```

```
## afl.finalists
##         Adelaide         Brisbane          Carlton      Collingwood
##               26               25               26               28
##         Essendon          Fitzroy        Fremantle          Geelong
##               32                0                6               39
##         Hawthorn        Melbourne  North Melbourne    Port Adelaide
##               27               28               28               17
##         Richmond         St Kilda           Sydney       West Coast
##                6               24               26               38
## Western Bulldogs
##               24
```

```
# Option 2: (modeOf function from lsr package)
modeOf(afl.finalists)
```

```
## [1] "Geelong"
```

```
# Trimmed mean
# The mean function in R has several interesting parameters, one of them is "trim"
dataSet <- c(-100, 2, 3, 4, 5, 6, 7, 8, 9, 10)

# Regular mean includes all values in the vector
mean(dataSet)
```

```
## [1] -4.6
```

```
# Trimmed mean that trims the 10% outliners from the lower and higher ends
mean(dataSet, trim = 0.10)
```

```
## [1] 5.5
```

Measures of Variability

It is often desirable to consider measures of variability (dispersion), as well as measures of location. For example, in choosing supplier A or supplier B we might consider not only the average delivery time for each, but also the variability in delivery time for each.

Range

The range of a data set is the difference between the largest and smallest data value. It is the simplest measure of variability. It is very sensitive to the smallest and largest data values.

Interquartile Range

The interquartile range of a data set is the difference between the third quartile and the first quartile. It is the range for the middle 50% of the data. It overcomes the sensitivity to extreme data values.

Variance

The variance is a measure of variability that utilizes all the data. It is based on the difference between the value of each observation and the mean (x bar for a sample, mu for a population). The variance is useful in comparing the variability of two or more variables.

Standard Deviation

The standard deviation of a data set is the positive square root of the variance. It is measured in the same units as the data, making it more easily interpreted than the variance.

```
# Find the Range of the data
max(afl.margins)
```

```
## [1] 116
```

```
min(afl.margins)
```

```
## [1] 0
```

```r
range(afl.margins)
```

```
## [1]   0 116
```

```r
# Find the Interquartile Range of the data
# quantile function in R
quantile(afl.margins)
```

```
##     0%    25%    50%    75%   100%
##   0.00  12.75  30.50  50.50 116.00
```

```r
# 50% quantile
quantile(afl.margins, probs = 0.5)
```

```
##  50%
## 30.5
```

```r
# Find the 25% and 75% quantile range
quantile(afl.margins, probs = c(0.25, 0.75))
```

```
##   25%   75%
## 12.75 50.50
```

```r
# Calculate the interquartile range
IQR(afl.margins)
```

```
## [1] 37.75
```

```r
# Mean Absolute Deviation
# Create a new vector
x <- c(4, 7, 10, 5, 6)

# Step 1: Calculate the mean of the data
xBar <- mean(x)

# Step 2: Calculate the absolute deviations from the mean
ad <- abs(x - xBar)

# Step 3: Calculate the mean absolute deviation
mad <- mean(ad)
print(mad)
```

```
## [1] 1.68
```

```r
# Use the aad() function from lsr package
aad(x)
```

```
## [1] 1.68
```

```r
# Variance of the data
# Step 1: Calculate the mean of the data
xBar <- mean(x)

# Step 2: Calculate the absolute deviations from the mean
sqDev <- (x - xBar)^2

# Step 3: Calculate the mean absolute deviation
va <- mean(sqDev)
print(va)
```

```
## [1] 4.24
```

```r
# Use the var() function from base R
# NOTE: the var() function in base R calculates the sample variance,
# not the population variance
var(x)
```

```
## [1] 5.3
```

```r
# Standard Deviation of the data
# Option 1:
sqrt(var(x))
```

```
## [1] 2.302173
```

```r
# Option 2: Use the sd() function in base R
# NOTE: the sd() function in base R calculates the sample standard deviation,
# not the population parameters
sd(x)
```

```
## [1] 2.302173
```

Descriptive Statistics Summary Table in R

Descriptive statistics is the term given to the analysis of data that helps describe, show, or summarize data in a meaningful way such that, for example, patterns might emerge from the data. Descriptive statistics do not, however, allow us to make conclusions beyond the data we have analyzed or reach conclusions regarding any hypotheses we might have made. They are simply a way to describe our data.

```r
# Summary of a vector of data
summary(afl.margins)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   12.75   30.50   35.30   50.50  116.00
```

```
summary(afl.finalists)
```

```
##         Adelaide         Brisbane          Carlton      Collingwood
##               26               25               26               28
##         Essendon          Fitzroy        Fremantle          Geelong
##               32                0                6               39
##         Hawthorn        Melbourne  North Melbourne    Port Adelaide
##               27               28               28               17
##         Richmond         St Kilda           Sydney       West Coast
##                6               24               26               38
## Western Bulldogs
##               24
```

```
# Summarizing a data frame
# load the clinicaltrial data frame
load("data/clinicaltrial.Rdata")
who(TRUE)
```

```
##    -- Name --      -- Class --   -- Size --
##    ad              numeric       5
##    afl.finalists   factor        400
##    afl.margins     numeric       176
##    clin.trial      data.frame    18 x 3
##     $drug          factor        18
##     $therapy       factor        18
##     $mood.gain     numeric       18
##    dataSet         numeric       10
##    mad             numeric       1
##    sqDev           numeric       5
##    va              numeric       1
##    x               numeric       5
##    xBar            numeric       1
```

```
# Print the first 6 rows of the data
head(clin.trial)
```

```
##       drug    therapy mood.gain
## 1  placebo no.therapy       0.5
## 2  placebo no.therapy       0.3
## 3  placebo no.therapy       0.1
## 4 anxifree no.therapy       0.6
## 5 anxifree no.therapy       0.4
## 6 anxifree no.therapy       0.2
```

```
# Summarize the data frame
summary(clin.trial)
```

```
##        drug         therapy    mood.gain
##  placebo :6   no.therapy:9   Min.   :0.1000
##  anxifree:6   CBT       :9   1st Qu.:0.4250
##  joyzepam:6                  Median :0.8500
```

```
##                             Mean   :0.8833
##                             3rd Qu.:1.3000
##                             Max.   :1.8000
```

```r
# Aggregate the data by group and summarized with the mean value
aggregate(formula = mood.gain ~ drug + therapy, data = clin.trial, FUN = mean)
```

```
##       drug    therapy mood.gain
## 1  placebo no.therapy  0.300000
## 2 anxifree no.therapy  0.400000
## 3 joyzepam no.therapy  1.466667
## 4  placebo        CBT  0.600000
## 5 anxifree        CBT  1.033333
## 6 joyzepam        CBT  1.500000
```

```r
# Aggregate the data by group and summarized with the standard deviation
aggregate(formula = mood.gain ~ drug + therapy, data = clin.trial, FUN = sd)
```

```
##       drug    therapy mood.gain
## 1  placebo no.therapy 0.2000000
## 2 anxifree no.therapy 0.2000000
## 3 joyzepam no.therapy 0.2081666
## 4  placebo        CBT 0.3000000
## 5 anxifree        CBT 0.2081666
## 6 joyzepam        CBT 0.2645751
```

```r
# Crosstab
# Create the crosstab with base R
crosstab <- table(clin.trial$drug, clin.trial$therapy)
print(crosstab)
```

```
##
##            no.therapy CBT
##   placebo           3   3
##   anxifree          3   3
##   joyzepam          3   3
```

```r
# Create frequency table
margin.table(crosstab, 1)  # drug freq. by summed over therapy
```

```
##
##  placebo anxifree joyzepam
##        6        6        6
```

```r
margin.table(crosstab, 2)  # therapy freq. by summed over drug
```

```
##
## no.therapy        CBT
##          9          9
```

```r
# Create proportion table
prop.table(crosstab, 1)
```

```
##
##             no.therapy CBT
##   placebo          0.5 0.5
##   anxifree         0.5 0.5
##   joyzepam         0.5 0.5
```

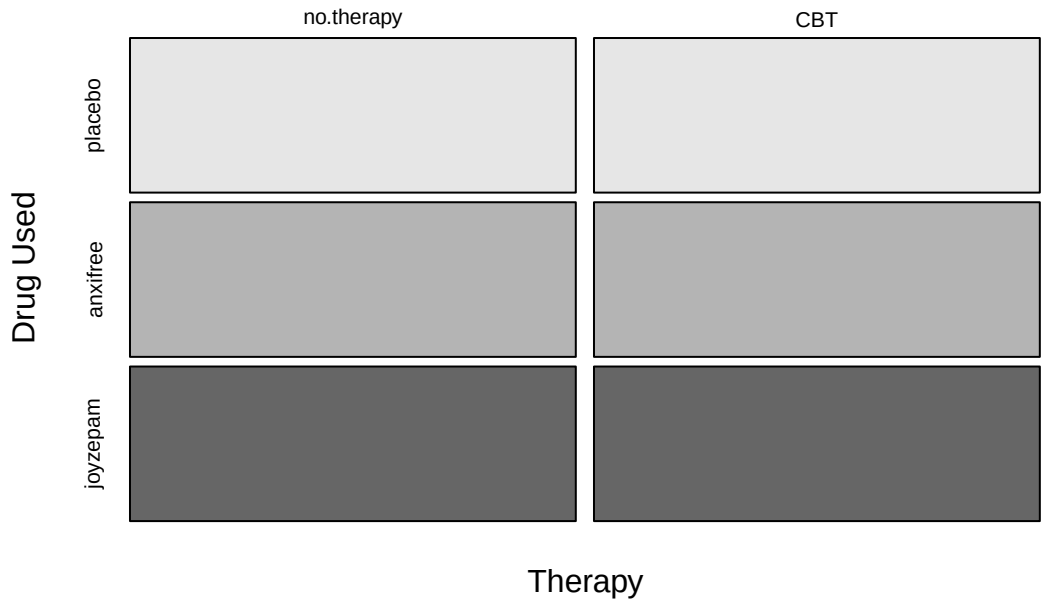
```r
prop.table(crosstab, 2)
```

```
##
##             no.therapy       CBT
##   placebo   0.3333333 0.3333333
##   anxifree  0.3333333 0.3333333
##   joyzepam  0.3333333 0.3333333
```

```r
# Load the descr package
# install.packages("descr")
library(descr)
```

```
## Warning: package 'descr' was built under R version 3.6.3
```

```r
crosstab(clin.trial$drug, clin.trial$therapy, xlab = "Therapy", ylab = "Drug Used")
```

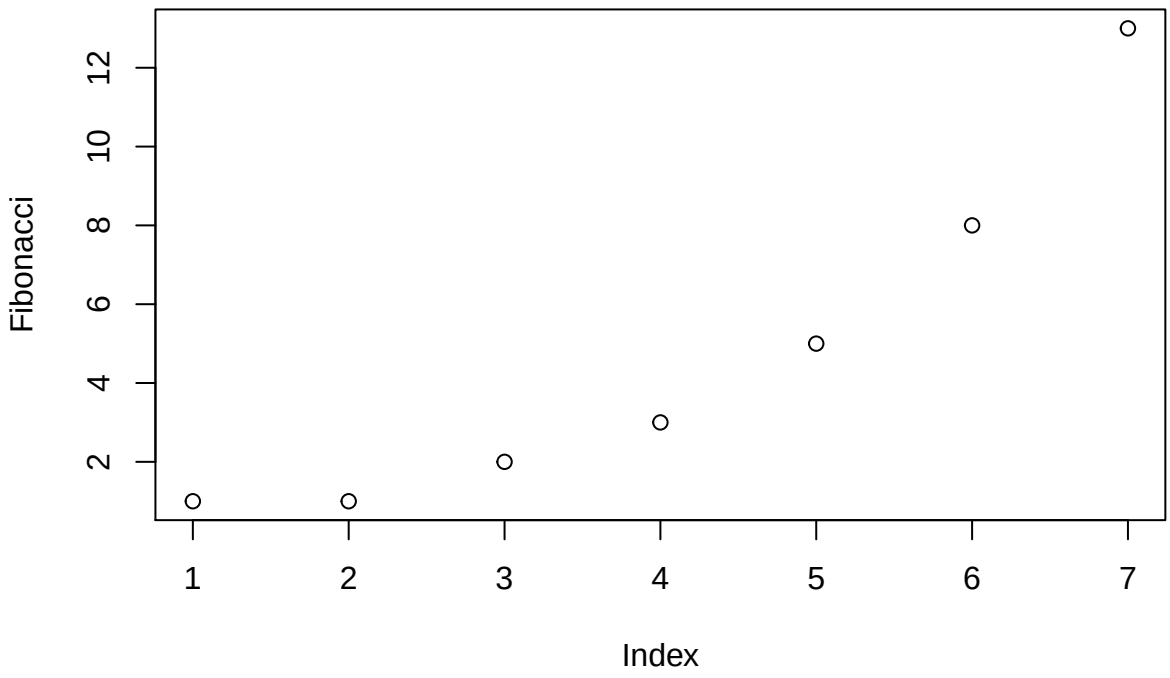


```
##    Cell Contents
## |-------------------------|
## |                   Count |
## |-------------------------|
##
## ==========================================
##                   clin.trial$therapy
## clin.trial$drug    no.therapy   CBT   Total
## ------------------------------------------
## placebo                     3     3       6
## ------------------------------------------
## anxifree                    3     3       6
## ------------------------------------------
## joyzepam                    3     3       6
## ------------------------------------------
## Total                       9     9      18
## ==========================================
```

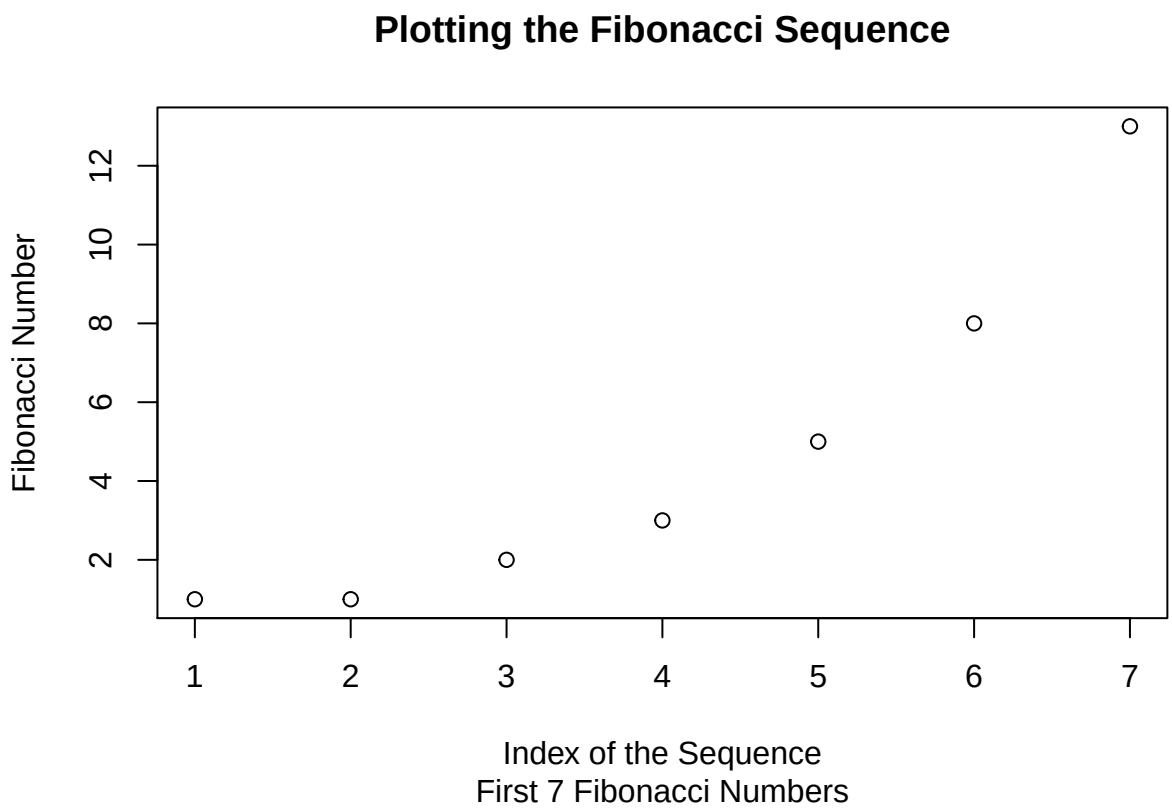
Visualization for Single Variable Data

The descriptive statistics in tabular format is useful to aggregate and summarize the data, however, it is not easy to see the pattern of the data. Visualizing the data helps to find any intuitive insight and pattern of the data more effectively. In most case, we use **bar chart** and **pie chart** to display **Categorical Data** and use **histogram** to display **Quantitative Data**. To visualize two variables case, we can use the **scatter plot** for two quantitative variables and **box plot** for quantitative and categorical combination.

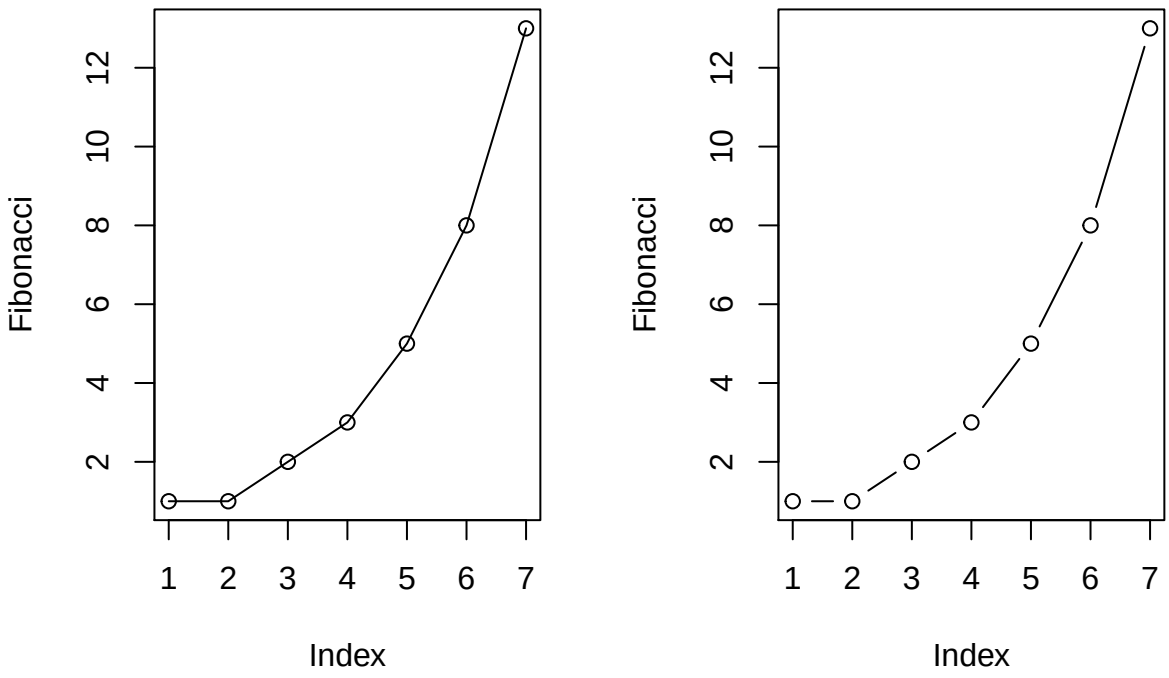
```r
# Example to demonstrates the base R plotting function
Fibonacci <- c(1, 1, 2, 3, 5, 8, 13)

# Plotting the vector, with its index
plot(Fibonacci)
```



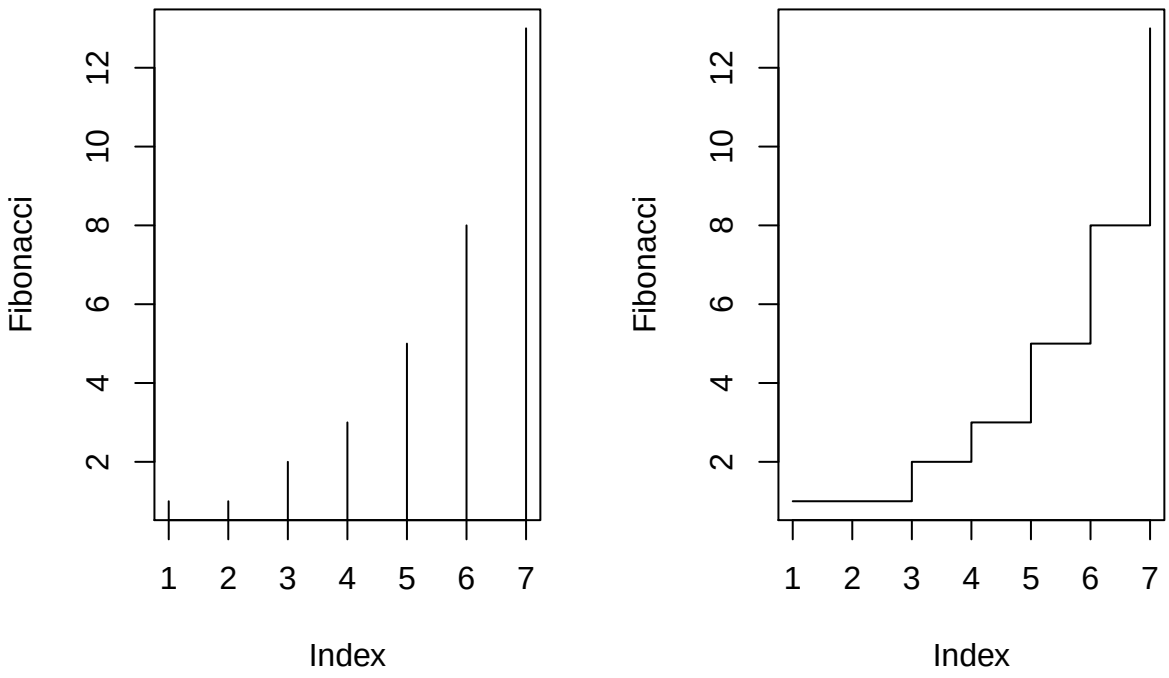

```r
# Customizing the title and axis labels
plot(Fibonacci,                                  # the data to plot
     main = "Plotting the Fibonacci Sequence", # the title
     sub = "First 7 Fibonacci Numbers",         # the sub-title
     xlab = "Index of the Sequence",            # x-axis label
     ylab = "Fibonacci Number")                 # y-axis label
```



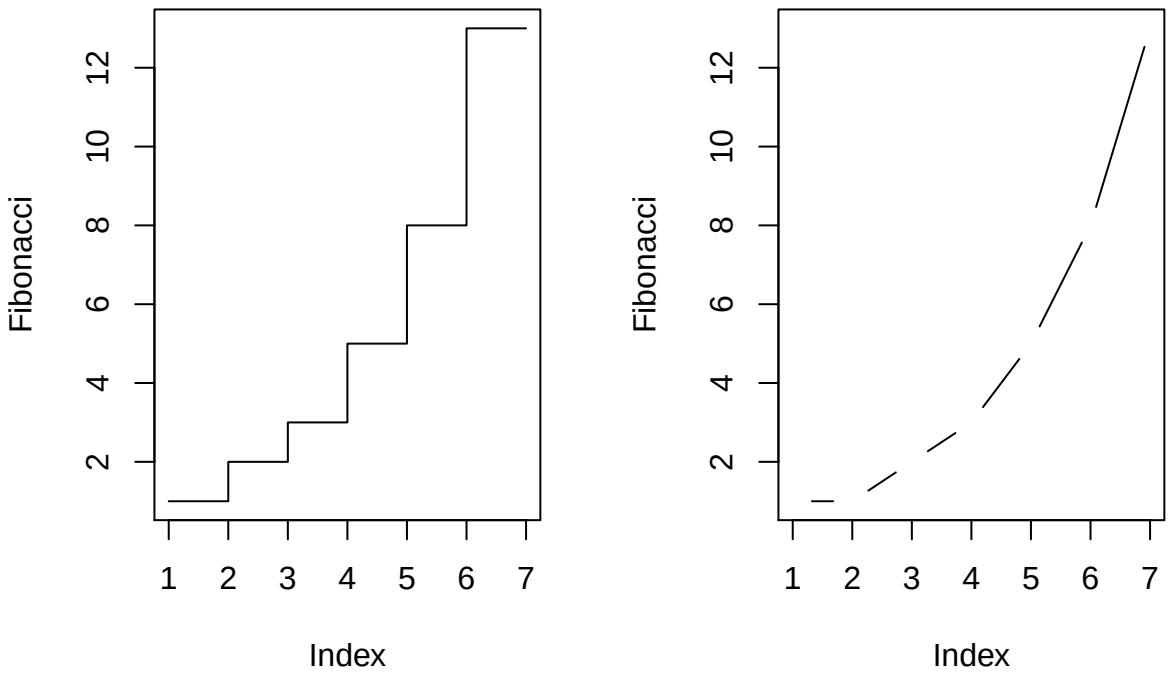

```r
# Different types of graph
par(mfrow = c(1,2))
plot(Fibonacci, type = "o")  # draw the line on top of the points
plot(Fibonacci, type = "b")  # draw both points and lines, but don't overplot
```



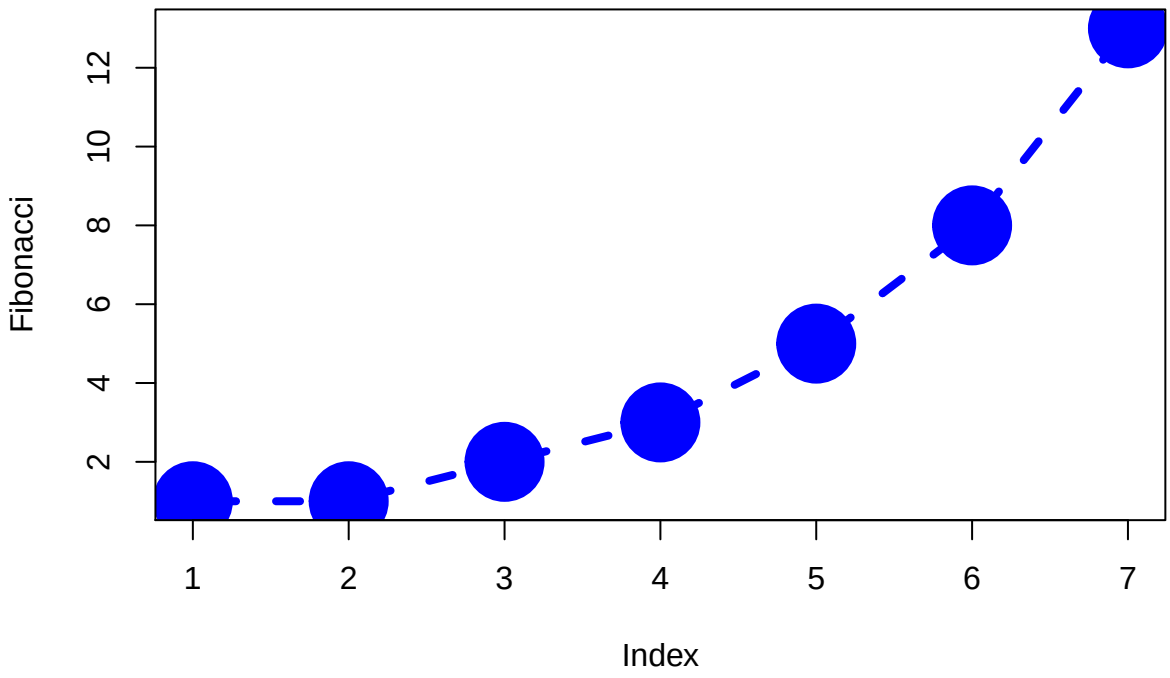

```
par(mfrow = c(1,2))
plot(Fibonacci, type = "h")  # draw histogram-like vertical bars
plot(Fibonacci, type = "s")  # draw a staircase, going horizontally then vertically
```



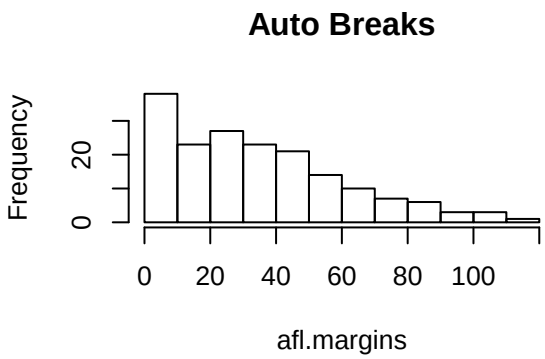

```
par(mfrow = c(1,2))
plot(Fibonacci, type = "S")  # draw a Staircase, going vertically then horizontally
plot(Fibonacci, type = "c")  # draw only the connecting lines from the "b" version
```

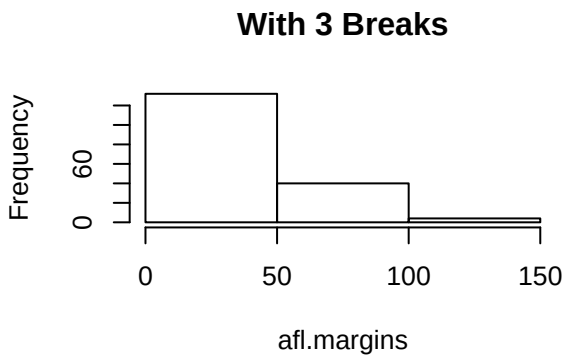


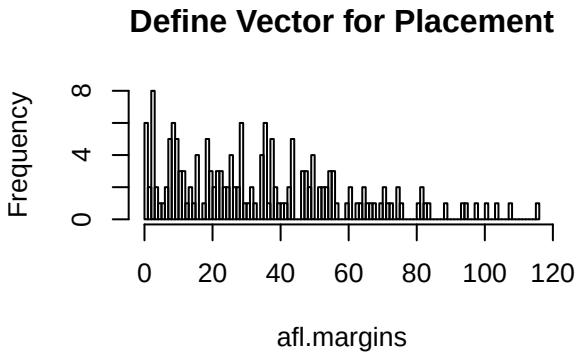

```
# Change other features of a plot
par(mfrow=c(1,1))
plot(Fibonacci,       # the data set
     type = "b",      # plot both points and lines
     col = "blue",    # change the plot color to blue
     pch = 19,        # plotting character is a solid circle
     cex = 5,         # plot it at 5x the usual size
     lty = 2,         # change line type to dashed
     lwd = 4)         # change line width to 4x the usual
```

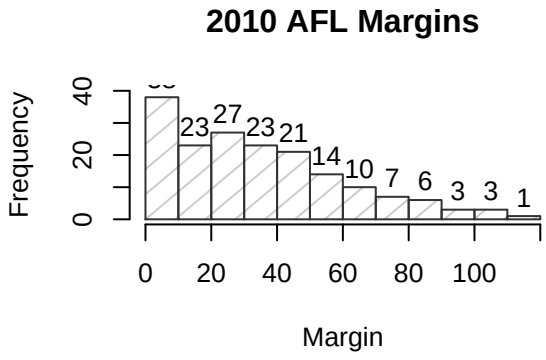



```
## Histogram
# Loading the Australian Football League (AFL) data set
load("data/aflsmall.Rdata")

# Plotting the histogram
par(mfrow = c(2,2))

hist(afl.margins,
     main = "Auto Breaks")

hist(afl.margins, breaks = 3,
     main = "With 3 Breaks")

hist(afl.margins, breaks = 0:116,
     main = "Define Vector for Placement")

hist(afl.margins,               # the data set
     main = "2010 AFL Margins", # title of the histogram
     xlab = "Margin",           # x-axis label
     density = 10,              # shading lines: 10 per inch
     angle = 40,                # set the angle of the shading lines is 40 degree
     border = "gray20",         # set the color of the borders of the bars
     col = "gray80",            # set the color of the shading lines
     labels = TRUE,             # add frequency labels to each bar
     ylim = c(0, 40))           # change the scale of the y-axis
```





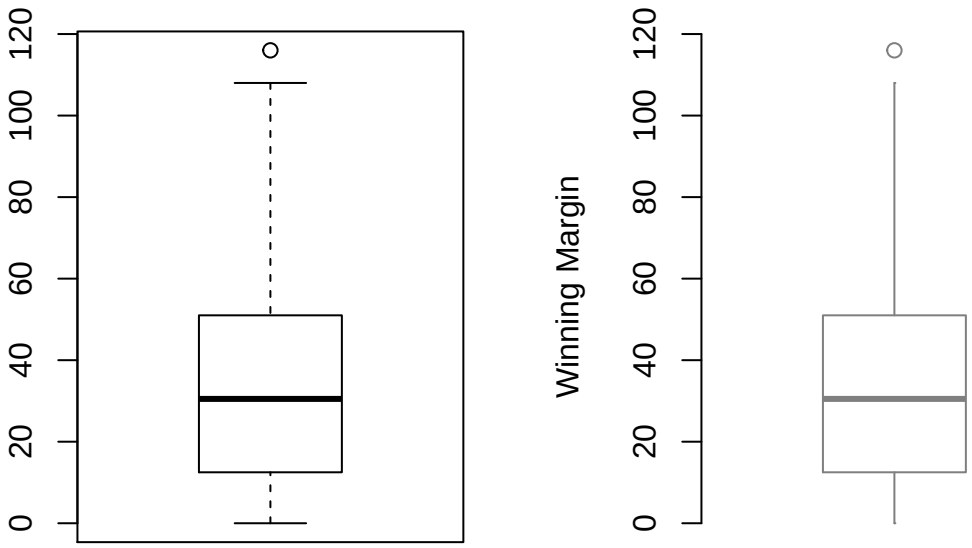





```
## Boxplot
# Take a look of the margins data set
summary(afl.margins)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   12.75   30.50   35.30   50.50  116.00
```

```
# Plot the basic boxplot for the margins data set
par(mfrow = c(1,2))
boxplot(afl.margins)

# Change the plot features
boxplot(afl.margins,                # the data set
        xlab = "AFL Games, 2010", # x-axis label
        ylab = "Winning Margin",  # y-axis label
        border = "grey50",        # dim the border of the box
        frame.plot = FALSE,       # don't draw a frame
        staplewex = 0,            # don't draw staples
        whisklty = 1)             # solid line for whisker
```



AFL Games, 2010

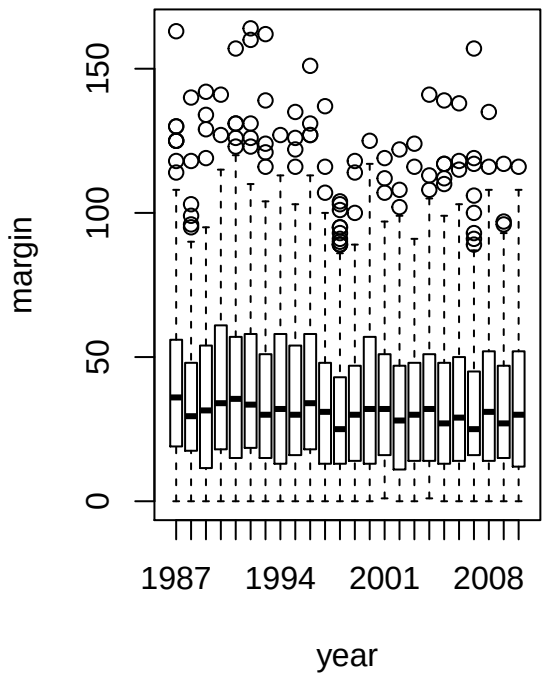
```
# Boxplots by Categories
# Load the aflsmall2 data set
load("data/aflsmall2.Rdata")
who(TRUE)
```

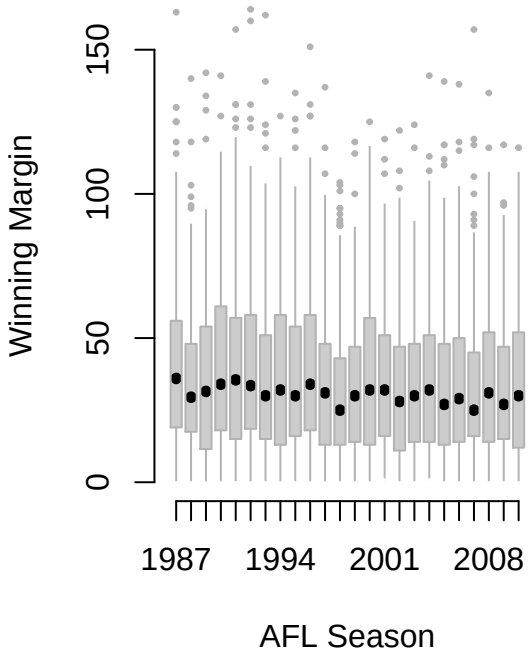
```
##    -- Name --      -- Class --   -- Size --
##    ad              numeric       5
##    afl.finalists   factor        400
##    afl.margins     numeric       176
##    afl2            data.frame    4296 x 2
##     $margin        numeric       4296
##     $year          numeric       4296
##    clin.trial      data.frame    18 x 3
##     $drug          factor        18
##     $therapy       factor        18
##     $mood.gain     numeric       18
##    crosstab        table         3 x 2
##    dataSet         numeric       10
##    Fibonacci       numeric       7
##    mad             numeric       1
##    sqDev           numeric       5
##    va              numeric       1
##    x               numeric       5
##    xBar            numeric       1
```

```r
# print the head of the data set
head(afl2)
```

```
##   margin year
## 1     33 1987
## 2     59 1987
## 3     45 1987
## 4     91 1987
## 5     39 1987
## 6      1 1987
```

```r
# Plot the margin data across different years
par(mfrow = c(1,2))
boxplot(margin ~ year,
        data = afl2)

# Change the features of the plot
boxplot(margin ~ year,
        data = afl2,
        xlab = "AFL Season",       # x-axis label
        ylab = "Winning Margin",   # y-axis label
        frame.plot = FALSE,        # don't draw a frame
        Staplewex = 0,             # don't draw staples
        staplecol = "White",       # fixes a tiny display issue
        boxwex = 0.75,             # narrow the boxes slightly
        boxfill = "grey80",        # lightly shade the boxes
        whisklty = 1,              # solid line for whiskers
        whiskcol = "grey70",       # dim the whiskers
        boxcol = "grey70",         # dim the box borders
        outcol = "grey70",         # dim the outliers
        outpch = 20,               # outliers as solid dots
        outcex = 0.5,              # shrink the outliers
        medlty = "blank",          # no line for the medians
        medpch = 20,               # instead, draw solid dots
        sedlwd = 1.5)              # make them larger
```






```
## Scatterplots
# Load the parenthood data set
load("data/parenthood.Rdata")
who()
```

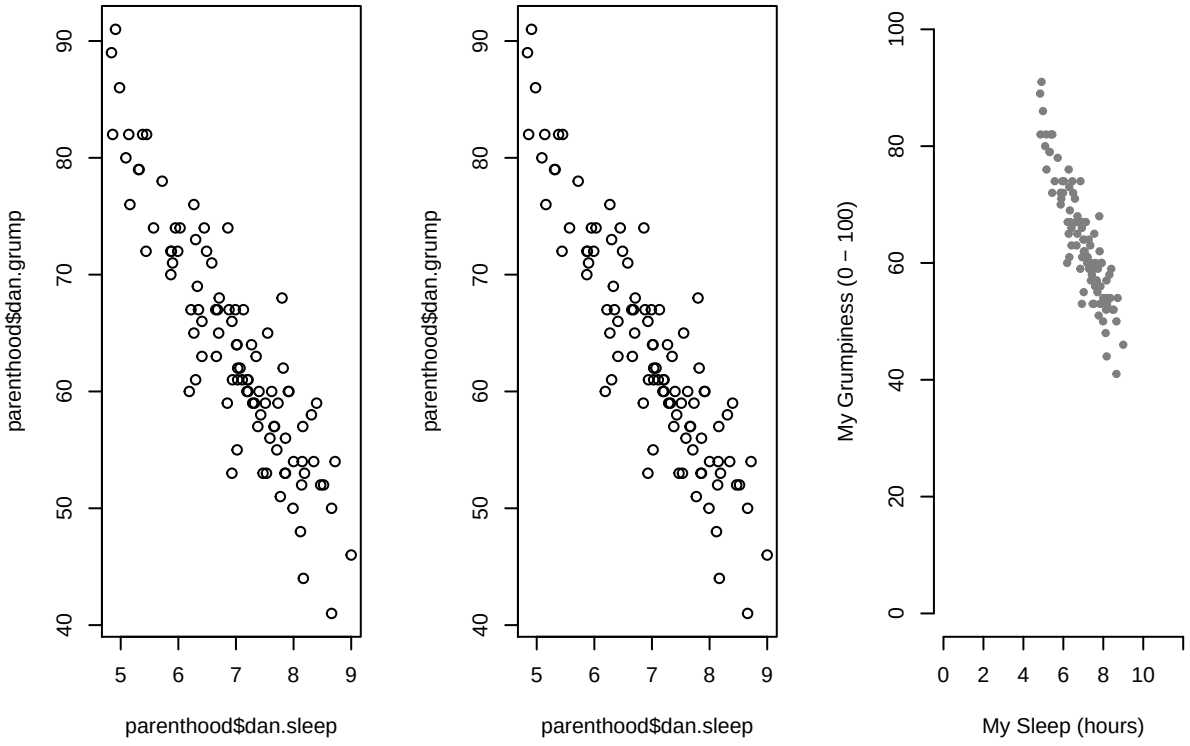
```
##    -- Name --      -- Class --   -- Size --
##    ad              numeric       5
##    afl.finalists   factor        400
##    afl.margins     numeric       176
##    afl2            data.frame    4296 x 2
##    clin.trial      data.frame    18 x 3
##    crosstab        table         3 x 2
##    dataSet         numeric       10
##    Fibonacci       numeric       7
##    mad             numeric       1
##    parenthood      data.frame    100 x 4
##    sqDev           numeric       5
##    va              numeric       1
##    x               numeric       5
##    xBar            numeric       1
```

```
# Print the head of the data
head(parenthood)
```

```
##   dan.sleep baby.sleep dan.grump day
```

```
## 1      7.59       10.18        56   1
## 2      7.91       11.66        60   2
## 3      5.14        7.92        82   3
## 4      7.71        9.61        55   4
## 5      6.68        9.75        67   5
## 6      5.99        5.04        72   6
```

```r
# Plot dan.sleep against dan.grump
par(mfrow = c(1,3))
plot(x = parenthood$dan.sleep,
     y = parenthood$dan.grump)

# Plot without explicitly calling the parameters
plot(parenthood$dan.sleep,
     parenthood$dan.grump)

# Changing the features of the plot
plot(parenthood$dan.sleep, parenthood$dan.grump,
     xlab = "My Sleep (hours)",          # x-axis label
     ylab = "My Grumpiness (0 - 100)",   # y-axis label
     xlim = c(0,12),                     # scale the x-axis
     ylim = c(0,100),                    # scale the y-axis
     pch = 20,                           # change the plot type
     col = "gray50",                     # dim the dots slightly
     frame.plot = FALSE)                 # don't draw a box
```



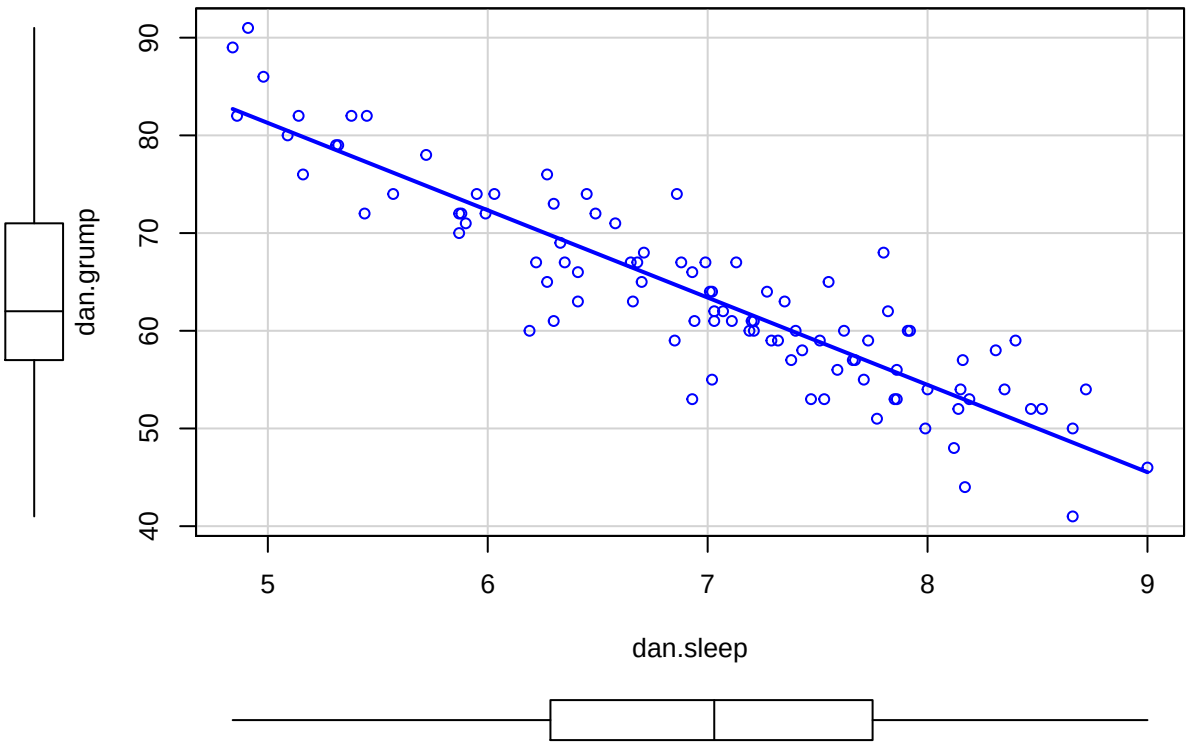
```
# An alternative for scatterplot is to use the scatterplot() function in the "car" package
# load the car package
library(car)
```

```
## Warning: package 'car' was built under R version 3.6.2
```

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

```
## Warning: package 'carData' was built under R version 3.6.1
```

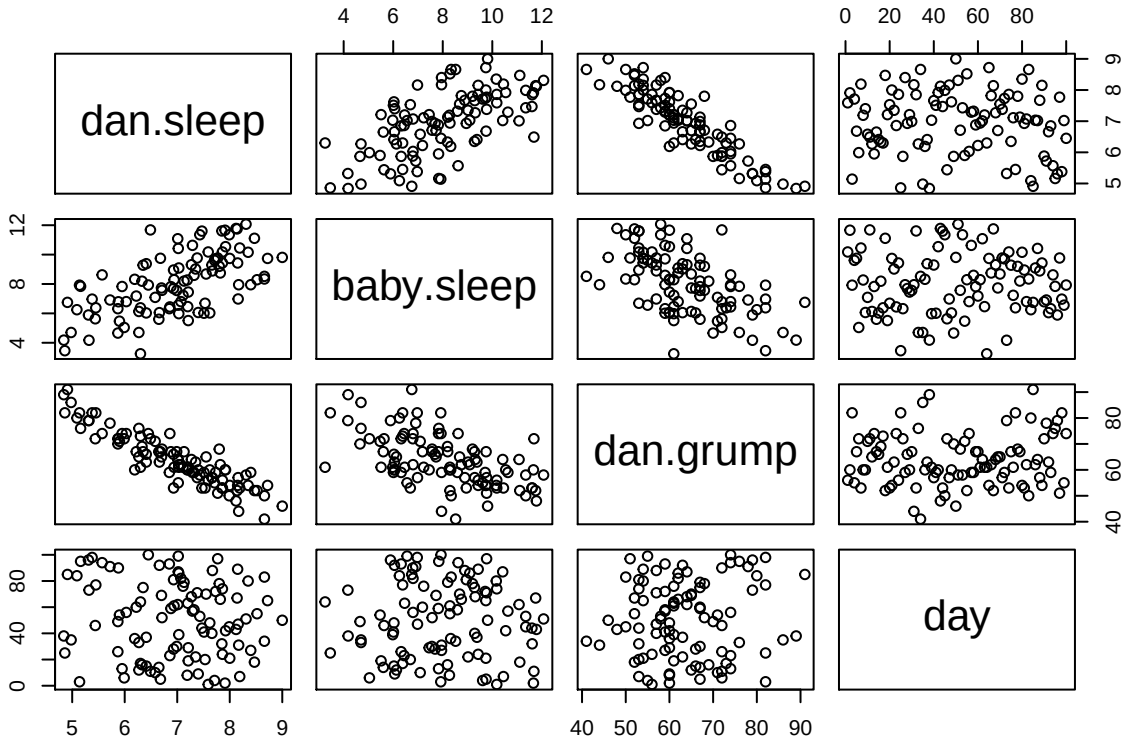
```
# Plot the data with scatterplot() function
par(mfrow=c(1,1))
scatterplot(dan.grump ~ dan.sleep, data = parenthood, smooth = FALSE)
```




```
# Scatterplot Matrix (Pair Plots)
# Display the correlations matrix of the data
cor(parenthood)
```

```
##              dan.sleep  baby.sleep   dan.grump         day
## dan.sleep   1.00000000  0.62794934 -0.90338404 -0.09840768
## baby.sleep  0.62794934  1.00000000 -0.56596373 -0.01043394
## dan.grump  -0.90338404 -0.56596373  1.00000000  0.07647926
## day        -0.09840768 -0.01043394  0.07647926  1.00000000
```

```
# Create a scatterplot matrix by using the pairs() function
pairs(parenthood)
```




```
# Bar Graphs
# Load the afl data set
load("data/aflsmall.Rdata.")

# Create a simple numeric vector for finalist's frequency
freq <- tabulate(afl.finalists)
print(freq)
```

```
##  [1] 26 25 26 28 32  0  6 39 27 28 28 17  6 24 26 38 24
```

```
# Create a new vector for all teams
teams <- levels(afl.finalists)
print(teams)
```

```
##  [1] "Adelaide"          "Brisbane"          "Carlton"           "Collingwood"
##  [5] "Essendon"          "Fitzroy"           "Fremantle"         "Geelong"
##  [9] "Hawthorn"          "Melbourne"         "North Melbourne"   "Port Adelaide"
## [13] "Richmond"          "St Kilda"          "Sydney"            "West Coast"
## [17] "Western Bulldogs"
```

```r
# Create a bar graph displays final frequency of each team
par(mfrow = c(1,3))
barplot(freq)

# Plot the bar graph with team names
barplot(freq, names.arg= teams)

# Change the graph features
barplot(freq,
        names.arg = teams,
        las= 2,                                    # rotate the label
        ylab = "Number of Finals",                 # y-axis label
        main = "Finals Played by Team, 1987-2010", # figure title
        density = 10,                              # shade the bars
        angle = 20)                                # shading lines angle
```