The >eR-Biostat initiative

Making R based education materials in statistics accessible for all

Basic concepts in exploratory data analysis (EDA) and computational statistics

Developed by

Martin Otava (Hasselt Univesrsity), Dinberu Seyoum (Jmma University) and Ziv Shkedy(Hasselt Univesrsity)

Visit us on Facebook ER-BioStat

GitHub https://github.com/eR-Biostat

Email: erbiostat@gmail.com

twitter @erbiostat

E-learning system using R

Biostatistics

The course was developed as a part of the >eR-BioStat initiative.

Most of the datasets used in the course are available as R objects.

External datasets are available in the GitHub page of the course.

Overview

- Exploratory data analysis.
- Basic Bootstrap methods.
- Computational statistics: likelihood functions & optimization.

Reference

- Exploratory data analysis:
  - selected chapters from the book “Understanding robust and exploratory data analysis”:
    - Chapter 3: BOXPLOTS AND BATCH COMPARISON
    - Chapter 10: COMPARING LOCATION ESTIMATORS: TRIMMED MEANS, MEDIANS, AND TRIMEAN
    - Chapter 11: M-ESTIMATORS OF LOCATION: AN OUTLINE OF THE THEORY
- Basic Bootstrap methods:
  - selected chapters from the book “An Introduction to the Bootstrap”:
    - Chapter 4: The empirical distribution function and the plug-in
    - principle
    - Chapter 5: Standard errors and estimated standard errors
    - Chapter 6 The bootstrap estimate of standard error

Part 1:
Exploratory Data Analysis: An introduction

Outline

- Chapter 1: Measure of Location
- Chapter 2: Spread
- Chapter 3: Resistance
- Chapter 4: Robustness

Chapter 1
Location

Introduction

- Location is the center of the distribution
- The place where the data are concentrated.

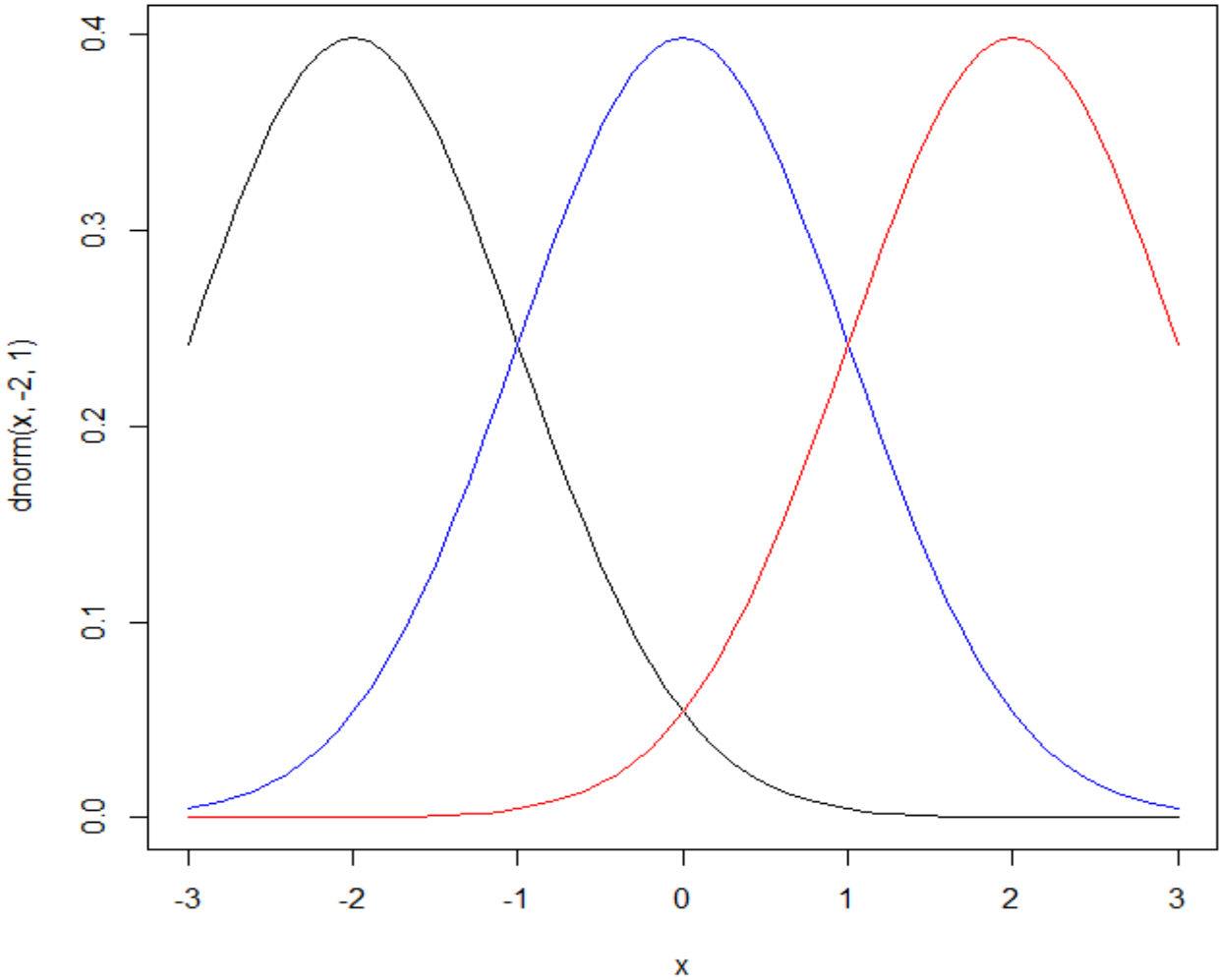
- A fundamental task in many statistical analyses is to estimate a location parameter for the distribution; i.e., to find a typical or central value that best describes the data.

- Consider a normal PDF with μ and δ^2

$$f(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Introduction

- Example: three density functions for μ = -2, 0 and 2 (black , blue and red). The distributions are shifted relative to each other and the value of μ determines the shift.



R-code

```
> x<-seq(-3,3,0.1)
> plot(x,dnorm(x, -2, 1),type="l")
> lines(x,dnorm(x, 0, 1),col="blue")
> lines(x,dnorm(x, 2, 1),col="red")
```

Location estimators

In real life μ is unknown and need to be estimated from the data.

The estimator for μ is called location estimator.

Numerical summaries:

- Mean
- Median
- Trimmed mean

Graphical displays:

- Strip plot
- Dot plot
- Boxplot

Numerical summaries for location

- Most common summary statistics: **sample mean**
- Other estimators: the **median** and the **trimmed mean**

- If the data comes from symmetric distribution the mean gives an estimate for the location of the center of the distribution.

- What if the data comes from non symmetric distribution ?
- How should we choose an estimator among the three?
- What is the difference between the mean, median and trimmed mean ?

Order statistics

- Definition:

  Let $X_1, X_2, \ldots, X_n$ be a sample of size n.

  The sorted values from smallest to largest,

  $X_{(1)}, X_{(2)}, \ldots, X_{(n)}$

  are called the order statistic.

- The k^{th} entry in the list, $x_{(k)}$, is the k^{th} order statistic, and approximately $100(k/n)\%$ of the observations fall below $x_{(k)}$.
- The sample quantiles are related to the order statistics.

Order statistics

- For example, consider the sample 35, 80, 105, 96 and 35. The sorted values are 35, 35, 80, 96,105 imply that:
  - The minimum value of the data is the first ordered statistics
  - $x_{(1)} = 35 = \min\{35, \ldots, 105)$
  - $x_{(3)} = 80$
  - The maximum is $x_{(5)} = 105$

- How to do it with R

```
> sort(x)  # to sort the data set
> quantile(x, probs = c(0.25, 0.37))
# first quartile, and 37th sample quantile
```

Numerical summaries

L-estimators

- Definition:

  Let $X_{(1)} \le X_{(2)} \le \ldots \le X_{(n)}$ be the order statistics of a sample of size n, and let $a_1, a_2, \ldots, a_n$ be weights, such that $0 \le a_i \le 1$ and

$$\sum_{i=1}^{n} a_i = 1$$

- L-estimator T_L with weights $a_1, a_2, \ldots, a_n$ is

$$T_L = \sum_{i=1}^{n} a_i X_{(i)}$$

Numerical summaries: Sample mean

- **Sample mean**
- It is an L-estimator.

- Let $a_i = 1/n$, for i=1,2,…,n. Then

$$T = \sum_{i=1}^{n} a_i X_{(i)} = \sum_{i=1}^{n} \frac{1}{n} X_{(i)} = \overline{X}$$

- In our example the sample mean is

$$\overline{X} = \frac{1}{5}(35 + \ldots + 105) = 70.2$$

Numerical summaries: Median

- **Median**
- It is L-estimator.
- The median is the value such that 50% of the data are less than or equal to.

- If *n* is odd, the median is equal to $X_{((n+1)/2)}^{th}$ ordered statistics.
- If *n* is even, the median is average of $X_{(n/2)}^{th}$ and $X_{(n/2+1)}^{th}$ order statistics.

Numerical summaries: Median

- For example, the median of

  35, 80, 105, 96, 35

  is 80 which is also $x_{(3)}$.

- If we define the weights:

$$a_i = \begin{cases} 1 \text{ if } i = \dfrac{n-1}{2} + 1 \\ 0 \quad otherwise \end{cases}$$

- Then it follows that

$$T = \sum_{i=1}^{n} a_i X_{(i)} = 1 \cdot X_{\left(\frac{n+1}{2}\right)} = median$$

Numerical summaries: Median

- When n is even then

$$a_i = \begin{cases} 0.5 \ \textit{if} \ \mathrm{i} = \dfrac{\mathrm{n}}{2}, \dfrac{\mathrm{n}}{2}+1 \\ 0 \qquad \text{otherwise} \end{cases}$$

- And the median is

$$T = \sum_{i=1}^{n} a_i X_{(i)} = 0.5 \cdot X_{\left(\frac{n}{2}\right)} + 0.5 \cdot X_{\left(\frac{n}{2}+1\right)} = median$$

Numerical summaries: Trimmed mean

- **Trimmed mean**
- The trimmed mean is the mean of the sample obtained after trimming a certain proportion of the observations at the upper and lower tails.
- For most statistical applications, 5% to 25% of the ends are discarded.

Numerical summaries: Trimmed mean

- For example, consider a sample of size 10:

  25, 27, 39, 57, 57, 63, 69, 75, 76, 94

- The 10%-trimmed mean, $T_{(0.2)}$, is the average of the 8 observations.

  27, 39, 57, 57, 63, 69, 75, 76

  which remain after trimming 10% of the observations at each tail, i.e the largest and the smallest values of the sample (25 and 94).

- The 10%-trimmed mean is therefore obtained by discarding 20% of the data, hence the notation $T_{(0.2)}$.

  In this example $T(0.2) = 57.875$.

Numerical summaries: Trimmed mean

- A 20%-trimmed mean, $T_{(0.4)}$, is the average of the data after trimming 20% of the data at each side, i.e.

$$T_{(0.4)} = \frac{1}{6}\sum_{i=3}^{8} x_i = \frac{1}{6}(39+57+57+63+69+75) = 60$$

- The α%-trimmed mean belongs to the L-estimators class. If we define

$$a_i = \begin{cases} \dfrac{1}{n-2\alpha n} & \text{if } n\alpha+1 \le i \le n-n\alpha \\ 0 & \text{otherwise} \end{cases}$$

Numerical summaries: Trimmed mean

- Then in our example, for n = 10 and α = 0.2 *(*40% of the observations are trimmed), we have:

- $n - 2\alpha n = 10 - 4 = 6$, $n\alpha + 1 = 3$ and $n - n\alpha = 8$

$$T_{(2\alpha)} = \sum_{i=1}^{n} \alpha_i x_{(i)} = \frac{1}{n - 2\alpha n} \sum_{i=n\alpha+1}^{n-n\alpha} x_{(i)}$$

$$= \frac{1}{6} \sum_{i=3}^{8} x_{(i)}$$

Location estimator in R

- Consider the sample :

  2, 4, 7, 1, 2, 9, 6, 4, 10, 18.

```
> x<-c(2,4,7,1,2,9,6,4,10,18)
> x
 [1]  2  4  7  1  2  9  6  4 10 18
> sort(x)
 [1]  1  2  2  4  4  6  7  9 10 18
```

Location estimator in R

- The sample mean would be

```
> mean(x)
[1] 6.3
```

- Since *n is* even, the median is the average between the two observations in the center, $\tilde{X} = \frac{4+6}{2} = 5$

```
> median(x)
[1] 5
```

- Trimmed mean is following:

```
> mean(x,trim=0.1) # 10% trimmed mean
[1] 5.5
> mean(x,trim=0.2) # 20% trimmed mean
[1] 5.333333
```

Graphical display of location

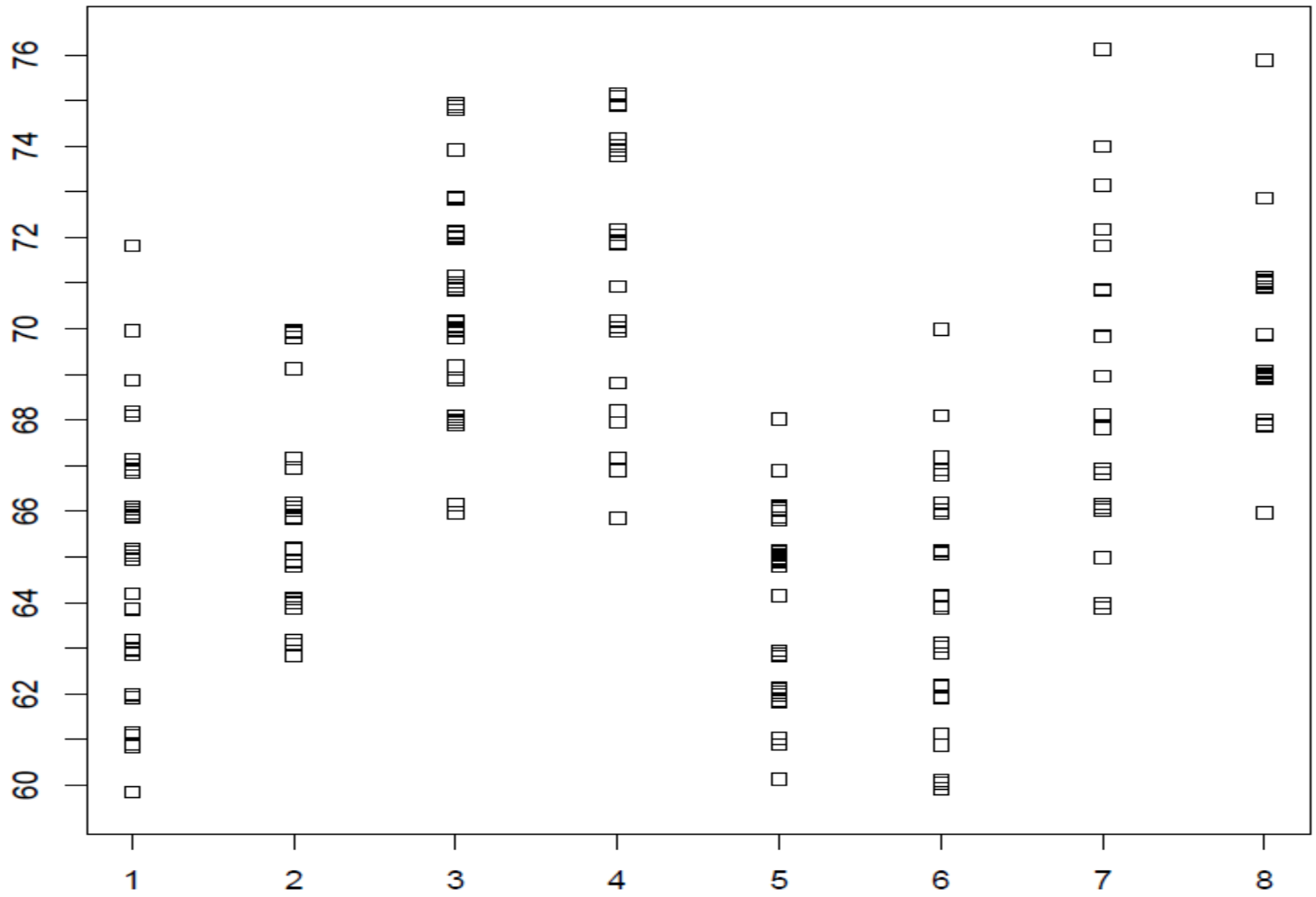
Graphical Display of location

Graphical displays:

- Strip plot
- Dot plot
- Boxplot

Graphical Display of location: Strip plot

- **Strip plot**



Graphical Display of location: Strip plot

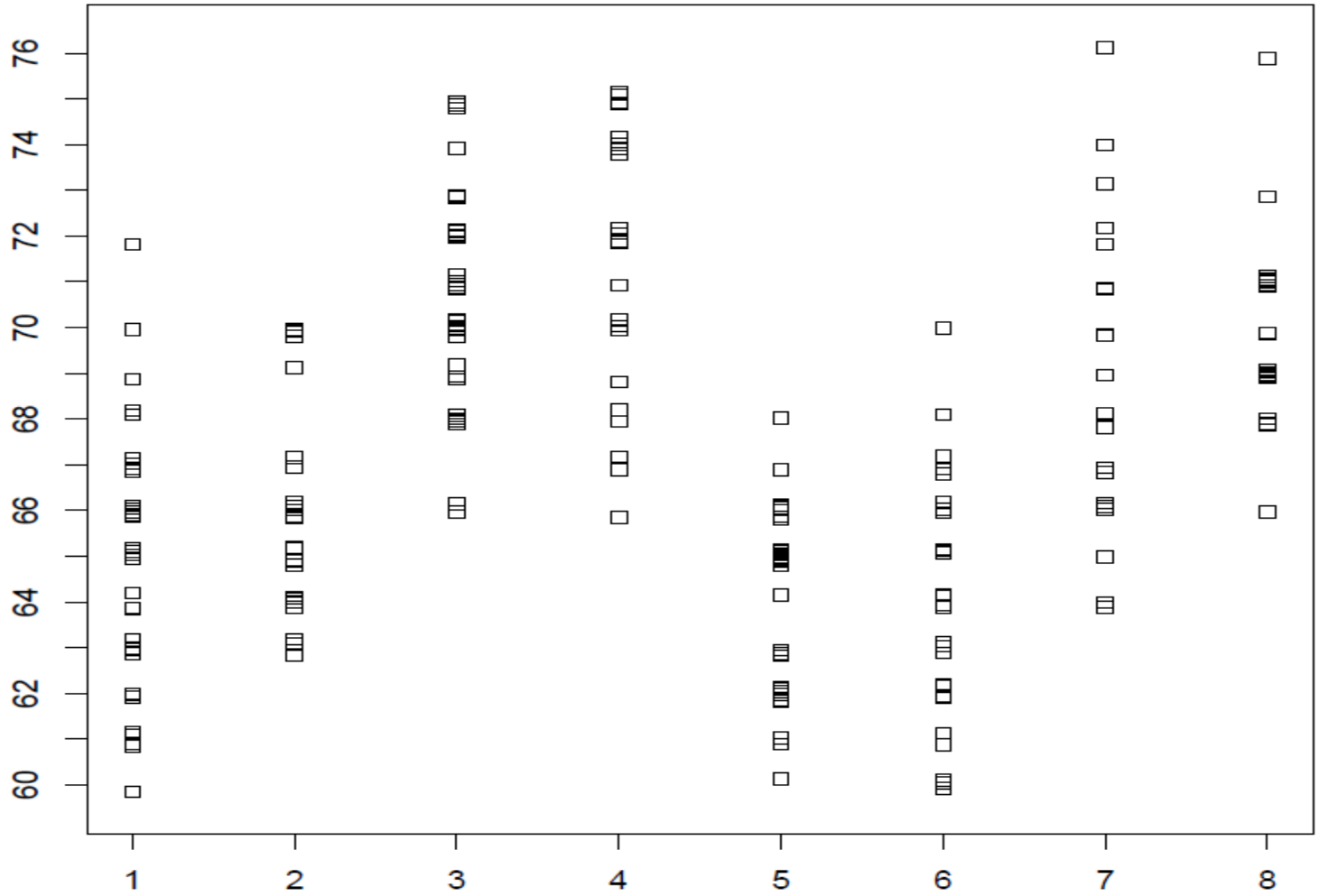
- Ideal for summarizing a univariate data set.

- Example data set:
  - singer data set

    Eight groups of different types of singers (gender and voice type combination)

Graphical Display of location: Strip plot

```
> stripplot(voice.part~height, data=singer,
  cex=0.5, jitter=TRUE)
```



Graphical Display of location: Strip plot

- Pattern in the data: It is easy to distinguish between women (Sopranos and Altos) and men (Tenors and Basses). Women are clearly shorter than men.
- Is this the only pattern in the data? Let us look at the groups mean.

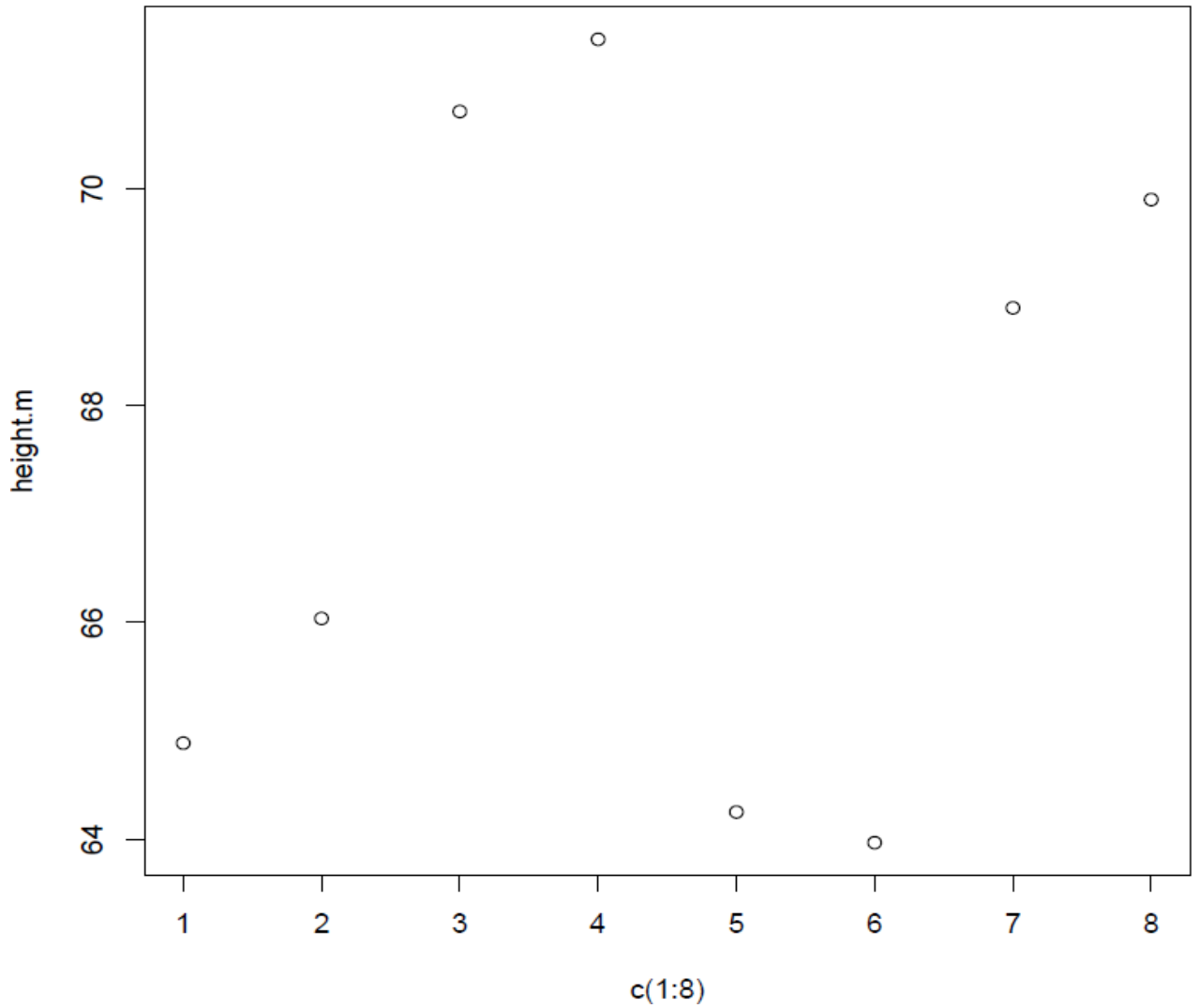
```
> attach(singer)
> tapply(height, voice.part, mean)
Bass 2       Bass 1   Tenor 2  Tenor 1   Alto 2
71.38462  70.71795  69.90476  68.90476  66.03704

Alto 1  Soprano 2  Soprano 1
64.88571 63.96667  64.25
```

Graphical Display of location: Strip plot

- On average men are taller than women.
- Singers with lower voice are taller than singers with higher voice. For example, the average of the two bass groups (71.38 and 70.71 for Bass 1 and Bass 2 respectively) are higher than the average in the tenor groups (69.90 and 68.90 for Tenor 1 and Tenor 2 respectively).
- Among women, the sopranos are shorter, on average, than the altos.

Graphical Display of location: Dot plot

- **Dot plot**



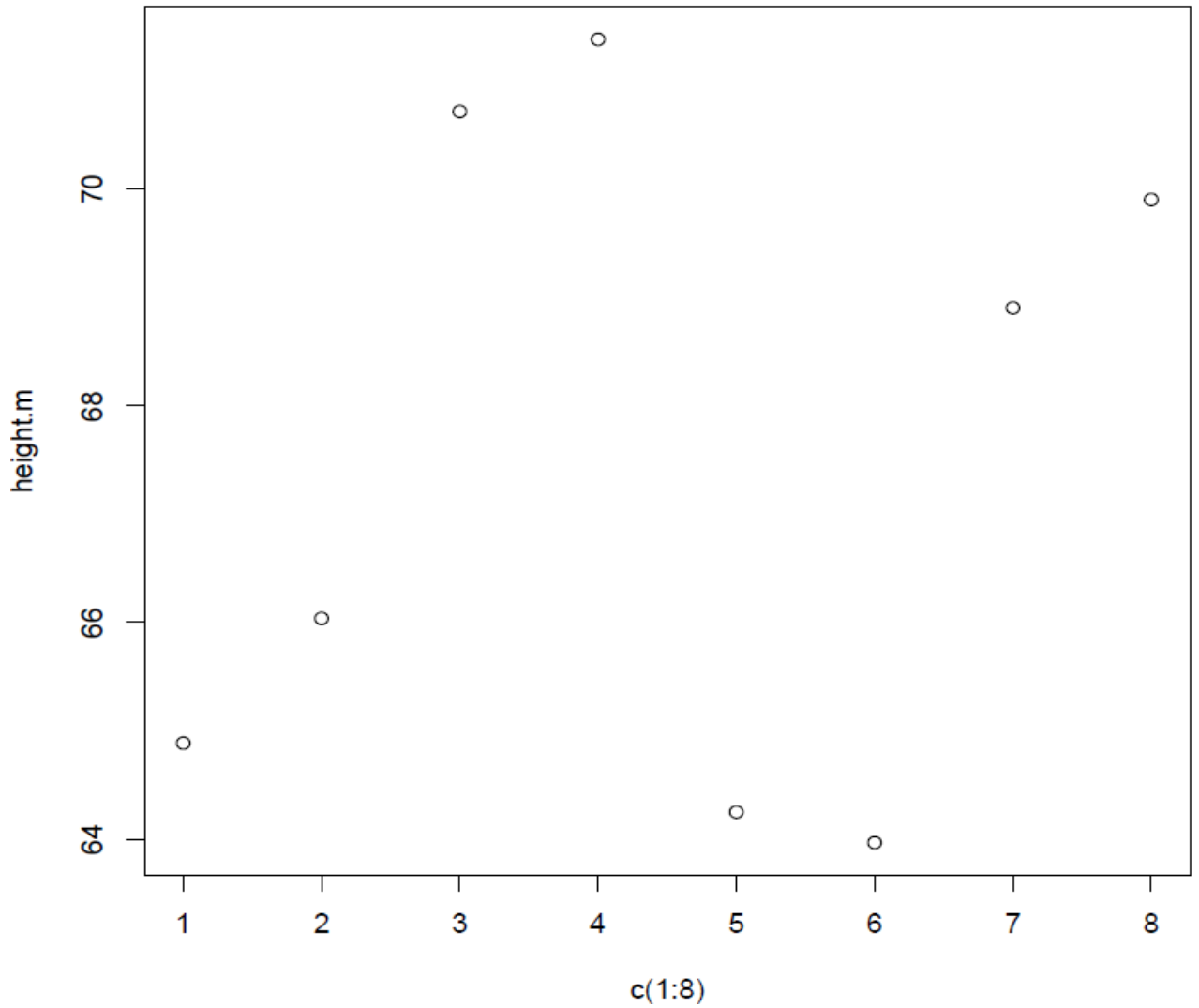
Graphical Display of location: Dot plot

- The dot plot is a graphical display of the group means.
- The location of each group is summarized by the mean and it appears in different strip.

```
> height.m <- tapply(height,voice.part,mean)
  #calculate the mean for each group
> dotplot(names(height.m)~height.m,cex=1.25)
```

Graphical Display of location: Dot plot

- Dot plot for the singers height by voice group



Graphical Display of location: Dotplot

- Male singers are taller than female singers (all the means from the men group are greater than the means from the women group).
- Within each gender group, singers with lower voices are taller: basses than tenors and altos than sopranos.
- Within three voice groups (except Sopranos), the singers with a lower voice (denoted as 2) are higher than singers with higher voice (denoted as 1).

Graphical Display of location: Dot plot

Example

- Look the data '*mtcars*' that found in R
- To see the available data on R software

```
> data()
```

- For the detail of the data

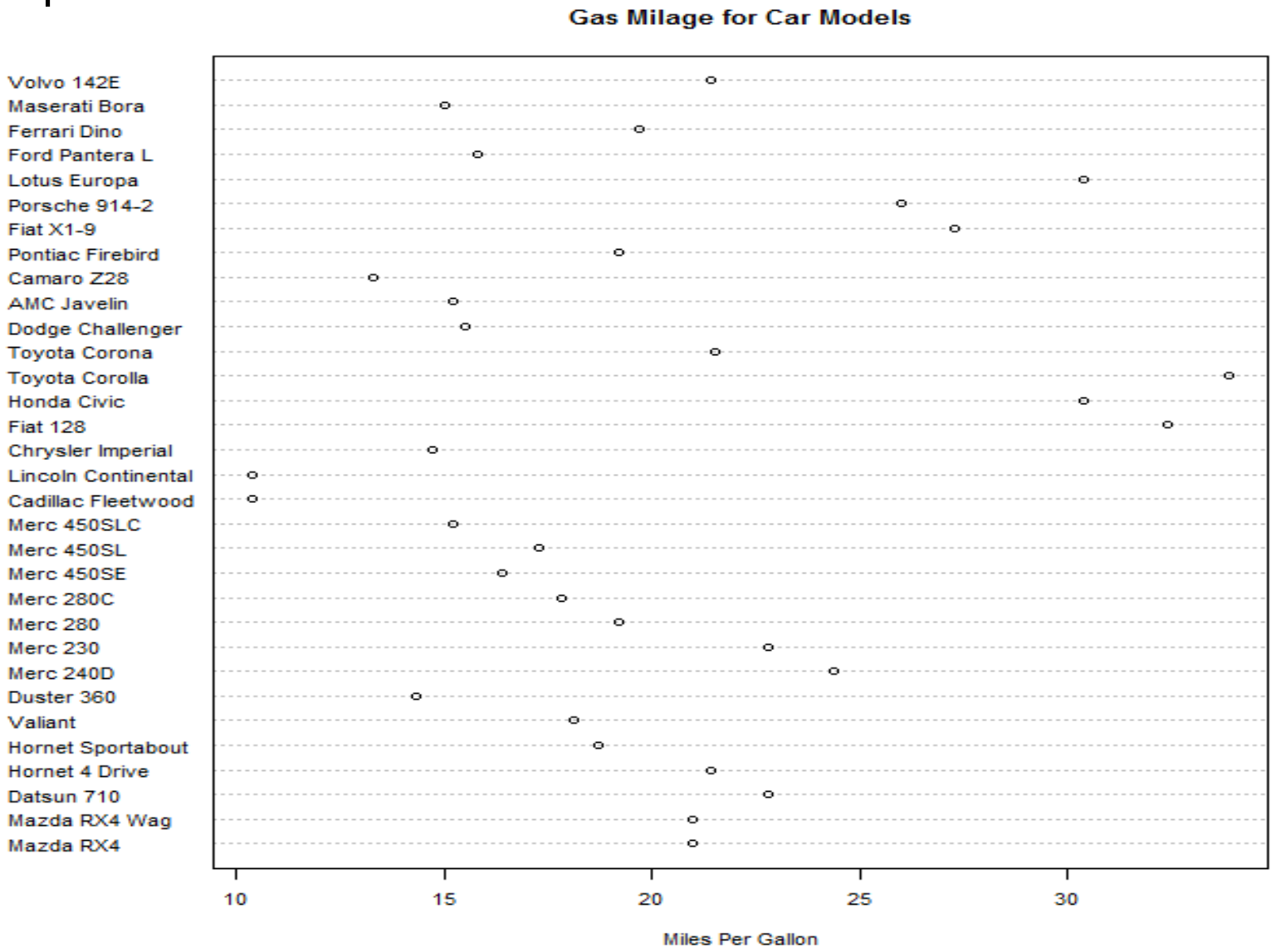
```
> str(mtcars) # or
> help(mtcars)
```

- Simple Dotplot

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

Graphical Display of location: Dot plot

Dot plot

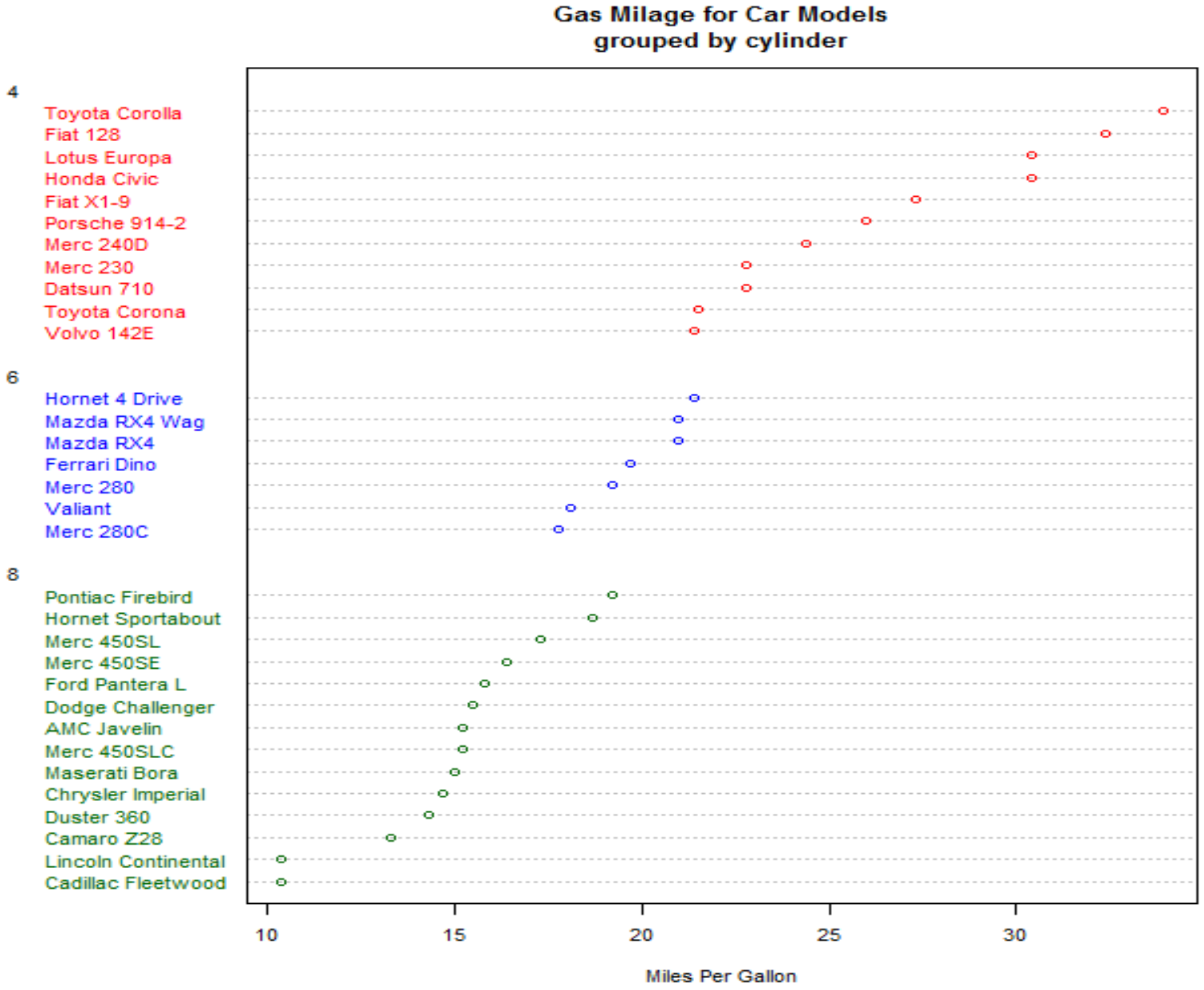


Graphical Display of location: Dot plot

```
> # Dotplot: Grouped Sorted and Colored
> # Sort by mpg, group and color by cylinder
> x <- mtcars[order(mtcars$mpg),] # sort by mpg
> x$cyl <- factor(x$cyl) # it must be a factor
> x$color[x$cyl==4] <- "red"
> x$color[x$cyl==6] <- "blue"
> x$color[x$cyl==8] <- "darkgreen"
> dotchart(x$mpg,labels=row.names(x),cex=.7,
+ groups= x$cyl, main="Gas Milage for Car
  Models\ngrouped by cylinder", xlab="Miles Per
  Gallon", gcolor="black", color=x$color)
```

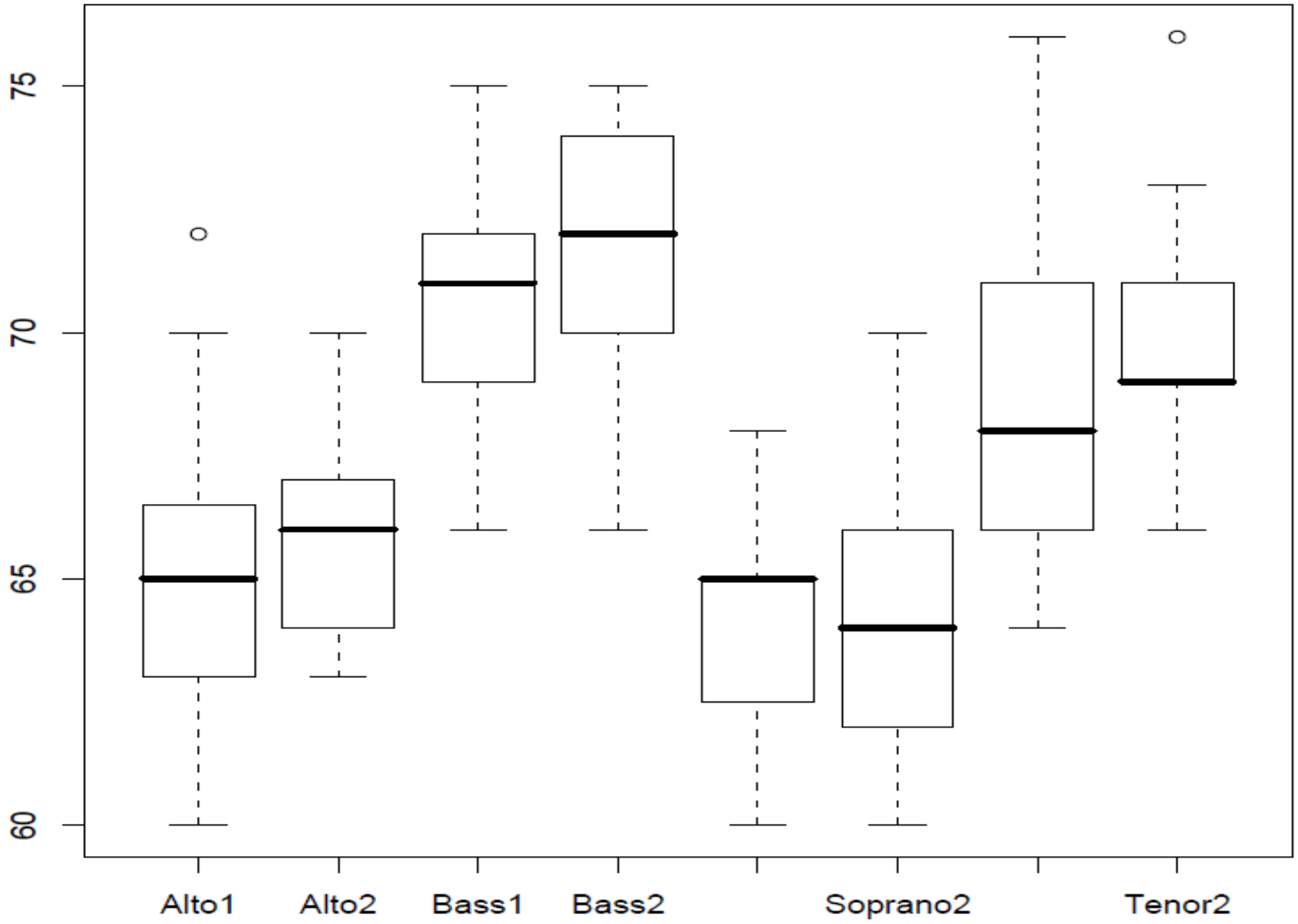
Graphical Display of location: Dot plot

By group



Graphical Display of location: Box plot

- **Boxplot**

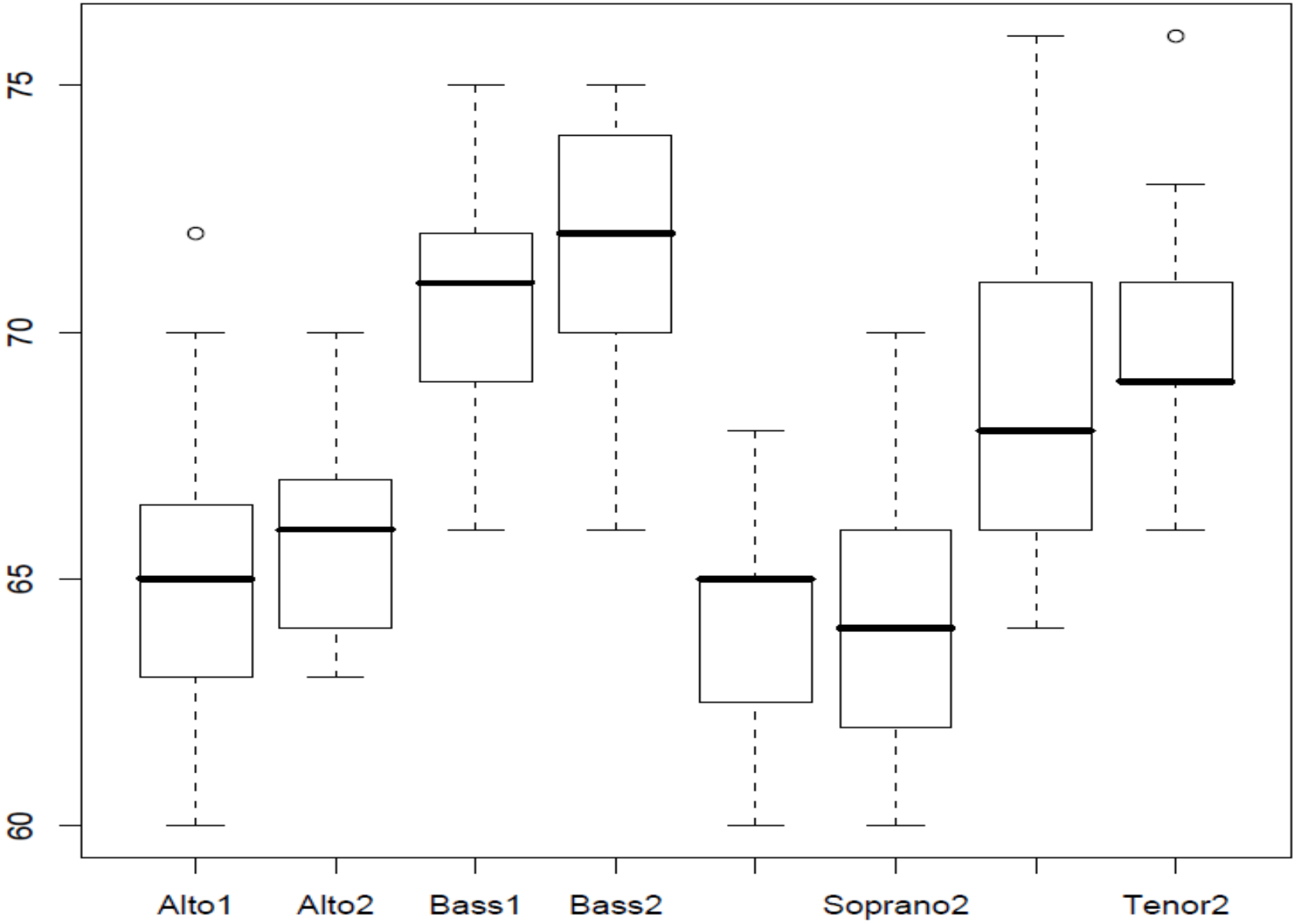


Graphical Display of location: Box plot

- Displays both location and spread (see later)
- The location of each group is summarized by the median (the vertical line inside the box).

```
> bwplot(voice.part ~ height, data=singer,
  xlab="Height (inches)")
```

Graphical Display of location: Boxplot



Practical session 1

Exercises 1

1. Draw two random samples of size 100 from the following distributions $N(0, 0.5^2)$ and $N(2, 0.5^2)$ and produce the histograms of these samples on one page.

   a) Use the function `rnorm()` to generate the samples in the following way:

```
> x <- rnorm(sample size, mean, standard
  deviation) # general call of rnorm
> x1 <- rnorm(100, 0, 0.5)
> x2 <- rnorm(100, 2, 0.5)
```

Exercises 1

b) Use the par() function in order to define the number of figures in one page:

```
> par(mfrow=c(2,1)) #put two figures in one page, two figures in one column
> hist(x1)
> hist(x2)
```

c) What is the difference between the two histograms ?

d) Does the two distributions look the same (in terms of shape)?

Exercises 2

2. a) Draw a random sample from of 25 observations from N(0,2)

```
> x1 <- rnorm(1000, 0, 2)
```

b) Draw a random sample from of 25 observations from χ^2_3

c) Plot the histogram for the two samples.

```
> hist(x1)
> hist(x2)
```

d) Calculate the mean T(0.2) and median for both samples and compare.

```
> mean(x1)
> mean(x1, trim = 0.1)
> median(x1)
> mean(x2)
> mean(x2, trim = 0.1)
> median(x2)
```

Exercises 3

3. a) Draw a random sample from N(0,1).

```
> x<-rnorm(10,0,1)
```

b) Calculate the mean and T(0.2).

```
> mean(x)
> mean(x,trim=0.1)
```

c) Change the largest value in the sample and calculate once again the location estimators.

```
> x<-sort(x)
> x[10]<-100*x[10]
> mean(x)
> mean(x,trim=0.1)
```

d) What is your conclusion?

Exercises 4

4. a) Use the fuel frame dataset and compare the distribution of the mileage by car type with strip plot.

```
> attach(fuel.frame)
> names(fuel.frame)
> stripplot(Type~Mileage,data=fuel.frame)
```

b) Compare the distribution of the mileage by car type with boxplot.

```
> bwplot(Type~Mileage,data=fuel.frame)
```

c) What is the difference between the boxplot and the stripplot ?

Exercises 5

5. a) Draw two samples (n=250) from N(0,1) and χ^2_1 and use stem-and-leaf to investigate the shape.

   b) Calculate the mean, median, T(0.2) and T(0.4) for the two samples. What are the differences that you see ?

Chapter 2
Spread

Introduction

- Until now we summarized the distribution of the data with location estimators
- In this chapter we will focus on the **spread**.

- Spread of a distribution measures how close the data are to each other, how concentrated are the data around the location of the distribution.

Introduction

- Consider the following hypothetical samples:

  -1, 0 , 1

  -50, 0, 50
- Both samples are symmetric around 0.
- The location estimators for both samples are the same (0).
- The data in the first sample range from -1 to 1, in the second sample the data range from -50 to 50.
- The variability in the second sample is higher.

Introduction

- Spread Estimators:
- **Standard deviation**
- The most simple measure for spread is the sample variance given by:

$$S_x^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2$$

- **Fourth-spared**
- A more robust estimator for the spread of the distribution is the fourth-spread (the **interquartile range=IQR**) given by

Fourth-spread = upper fourth – lower fourth

Standard deviation and Four-spread

- The fourth-spread is the difference between the 75% and the 25% quantiles of the data.
- It is the range of 50% of the data in the center of the distribution
- It is more robust estimator than the variance since it is not influenced from outliers at the tails as the variance (see later).

- Consider a sample of 5 observations:

24, 35, 39, 50, 60

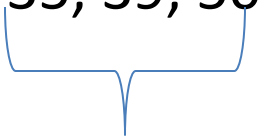
50-35=15

- The fourth-spread is 15 and the sample variance 192.3.

Standard deviation and Four-spread

- Now, suppose that we change the sample to

24, 35, 39, 50, 800



50-35=15

- The fourth-spread remains the same
- The sample variance now is equal to 116,520.3.

- Hence, sample variance is sensitive to change, but four-spread is not.

Median Absolute Deviation

MAD - Median Absolute Deviation

- Another robust measure of spread is the MAD, which is the Median of the Absolute Deviation from the median, given by

$$MAD = Median \left| X_i - M \right|$$

- Where M is the sample median

- In our example the median is 39.

```
> xi <- c(24,35,39,50,60)
> median(xi)
[1] 39
```

Median Absolute Deviation

- When we subtract the median we have

```
> xi-median(xi)
[1] -15 -4  0 11 21 # xi-39
```

- The sorted absolute values are

```
> abs(xi-median(xi))
[1] 15  4  0 11 21 # |xi-39|
> sort(abs(xi-median(xi)))
[1] 0  4 11 15 21 # sort values of |xi-39|
```

- Finally, the MAD is the median of

(0, 4, 11, 15, 21)

```
> median(abs(xi-median(xi))) # the MAD
[1] 11
```

Median Absolute Deviation

- When we change the maximum value of the sample from 60 to 800 the MAD does not change.

```
> xi[5] <- 800 # change the maximum value to 800
> xi
[1] 24 35 39 50 800
# the sample after the change
> mad(xi, constant=1) # calculate the MAD
[1] 11
```

Graphical display of spread

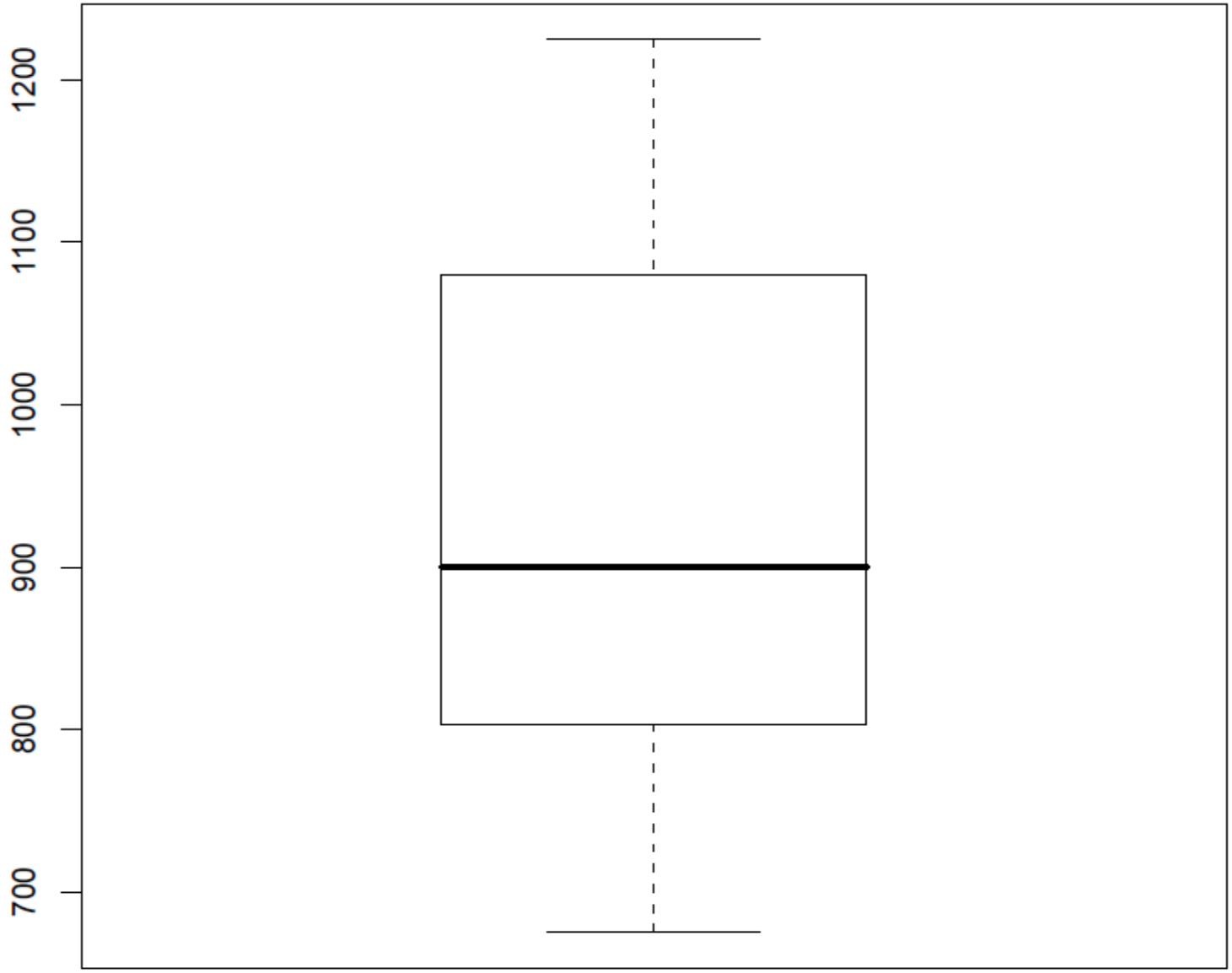
Graphical display: Boxplot

- Boxplot is a graphical display which shows the location, the spread and the shape of the distribution.
- The location is summarized by the median
- The spread is summarized by the fourth-spread which is simply the length of the box in the boxplot (the shape will be discussed in chapter 4).

- The figure found below shows the boxplot for the urban students.

Graphical display: Boxplot

Boxplot of the SAT score for the Urban students.



Graphical display: Boxplot

- The upper and lower adjacent values are given by

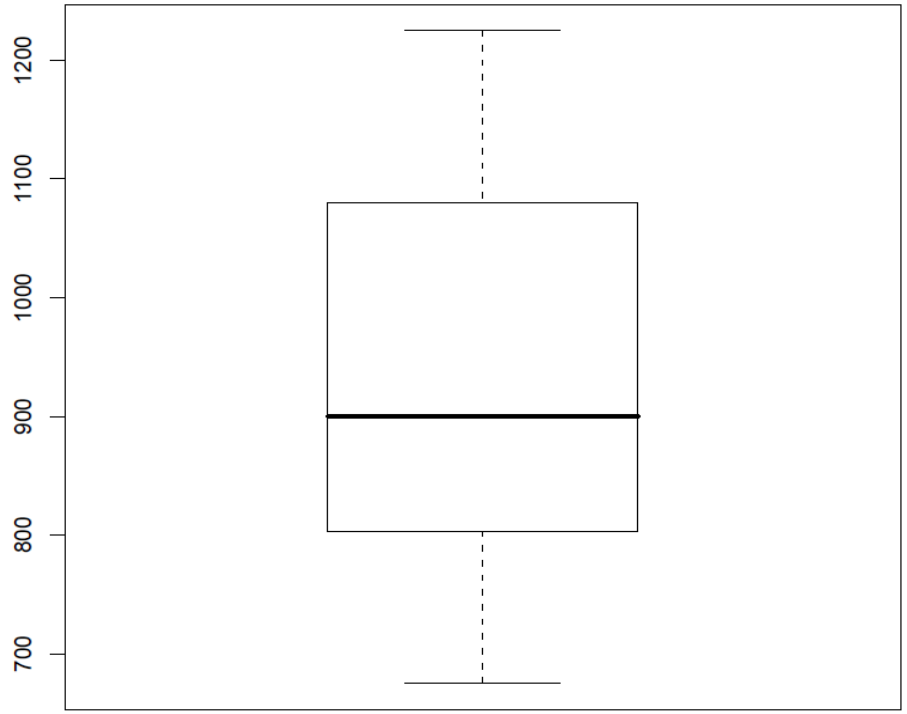
$$\text{Upper adjacent value} = \min\{\max(x), Q3 + 1.5(Q3 - Q1)\}$$
$$\text{Lower adjacent value} = \max\{\min(x), Q1 - 1.5(Q3 - Q1)\}$$

- For the urban students these values are

$$\text{Upper adjacent value} = \min\{1225, 1080 + 1.5 * 277\} = 1225$$
$$\text{Upper adjacent value} = \max\{675, 803 - 1.5 * 277\} = 675$$

Graphical display: Boxplot

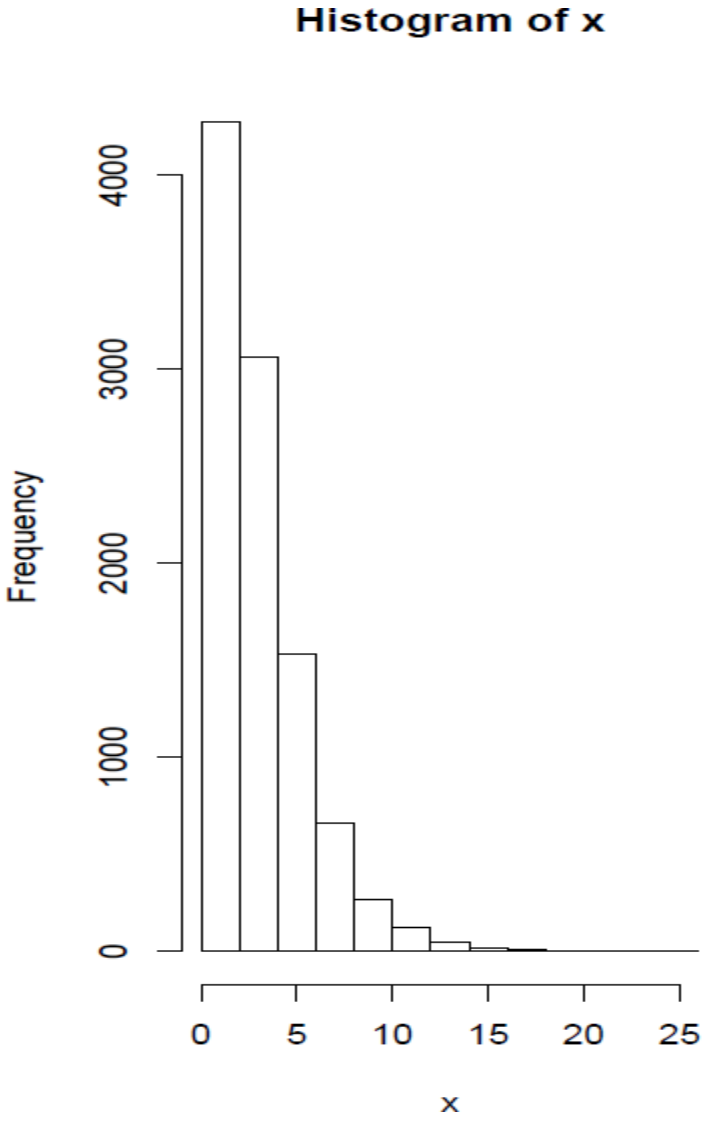
- The SAR data.
- The upper and lower adjacent values are used to identify outliers.
- Observations higher than the upper adjacent value or smaller than the lower adjacent value are considered to be outliers.

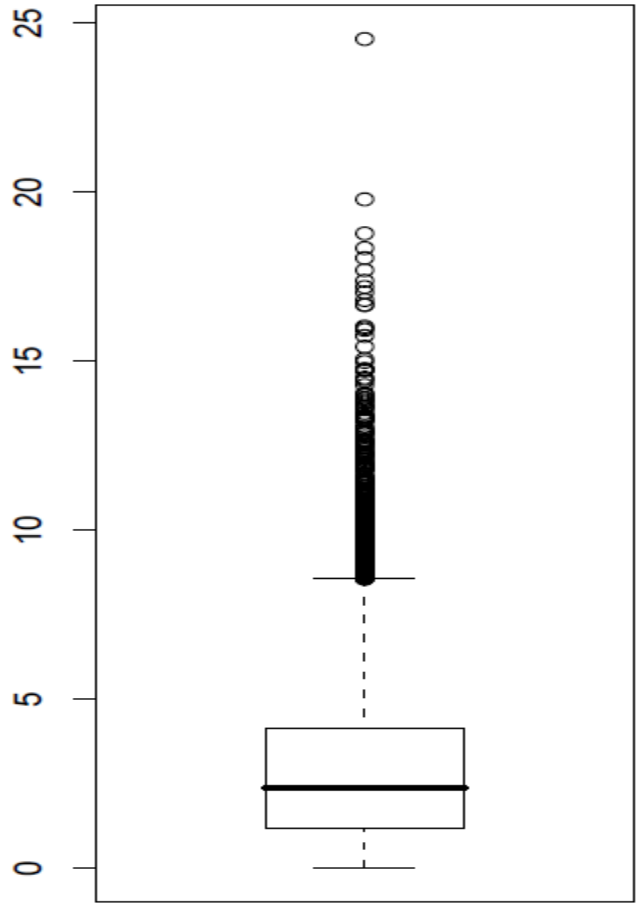



```
attach(Sat)
> boxplot(SAT[Place == "Urban"], boxcol=0, medcol=1)
```

Graphical display: Boxplot

Histogram and Boxplot of skewed distribution





Graphical display: Boxplot

Comparing spread with boxplots

- Boxplots are commonly used to compare the distribution of several samples.
- In the SAT dataset the aim of the analysis is to compare the distributions of the urban and rural students.
- We can see that the fourth-spread in the rural sample is equal to 85, which is smaller than the fourth-spread in the urban sample (277).

Graphical display: Boxplot

```
> quantile(Urban.sat)
0%     25%     50%     75%     100%
675    803     900     1080    1225
> quantile(Rural.sat)
0%    25%    50%    75%    100%
500   765    812    850    974

> diff(quantile(Urban.sat, probs = c(0.25, 0.75)))
277
> diff(quantile(Rural.sat, probs = c(0.25, 0.75)))
85
```

Graphical display: Boxplot

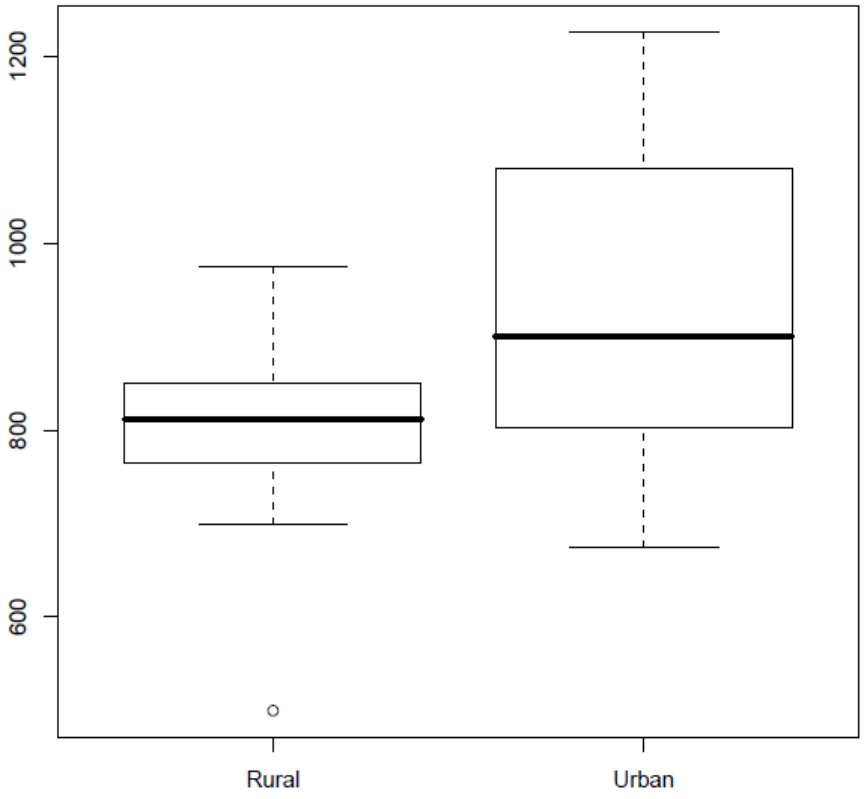
- This difference in the spread is visualized in figure below where we can see that the box for the urban sample is heigher than the box of the rural sample.

- This indicates that the spread among the urban students is greater than the spread among the rural students.

- Note that there is one outlier in the rural sample with SAT score equal to 500. The lower adjacent value in the rural sample is equal to 765 – 1.5 *85 = 637.5 and therefore the student with SAT score of 500 is considered to be an outlier.

Graphical display: Boxplot

Boxplot of the SAT scores for rural and urban students




```
boxplot(split(SAT,Place), names=c("Rural","Urban"),
boxcol=0,medcol=1)
```

Graphical display: Boxplot

Example:

- Let us see the data for life expectancy of 44 countries.
- Read the data here

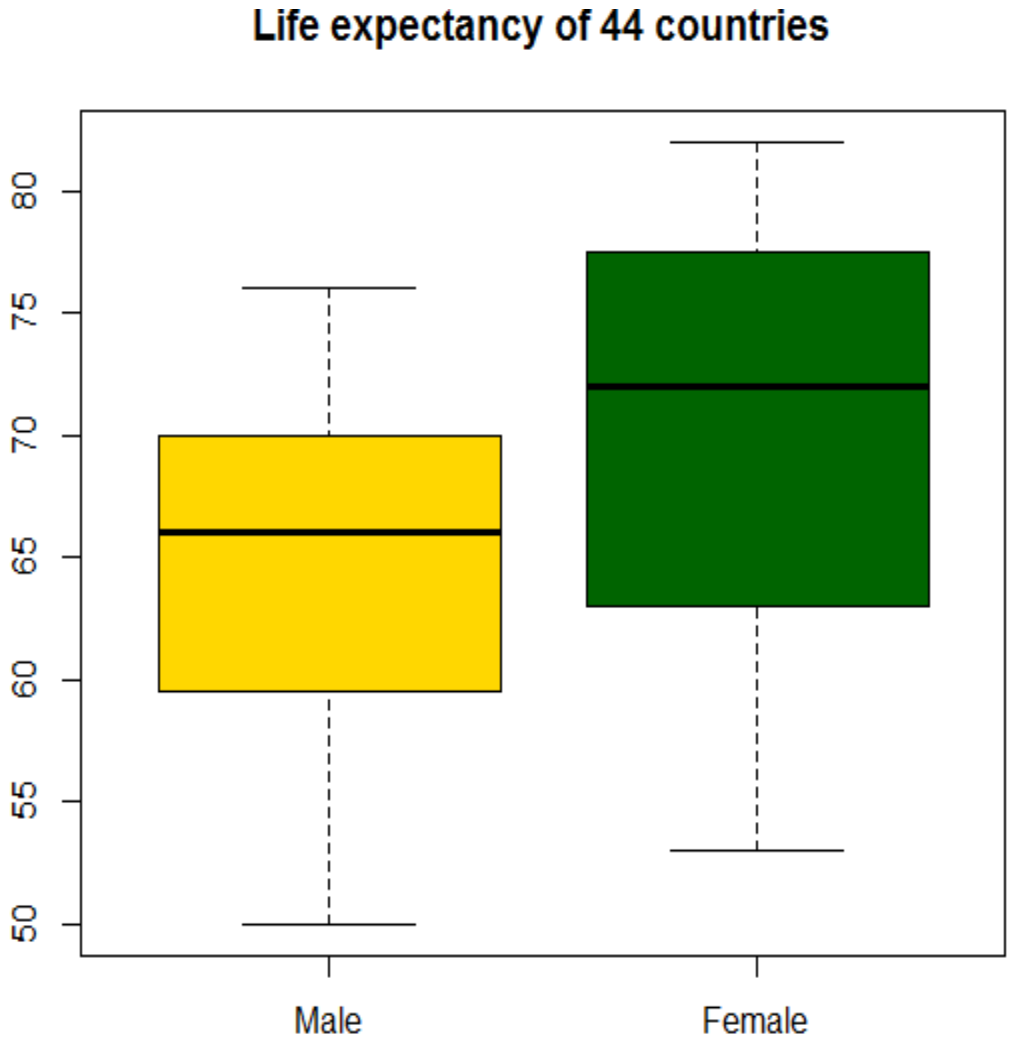
```
> tt <-
  read.csv('C:\\Users\\uhstudent\\Documents\\life.
  expe.csv', header = T, dec=",", sep=";")
> attach(tt)
```

- Plot the box plot for female and male life expectancy. Compare the spread regarding the life expectancy in the box plot

```
> boxplot(M.life.exp, F.life.exp, col =
  c("gold","darkgreen"), names =
  c("Male","Female"), outlier = TRUE, main = "Life
  expectancy of 44 countries")
```

Graphical display: Boxplot

- Life expectancy for male and female in 44 countries.



Graphical display: Boxplot

- The height of box in male life expectancy is smaller than the female. Therefore, the spread in male life expectancy is less than the female.
- Indeed, the male fourth spread value is smaller than female fourth spread:

  Male forth spread=9.75

  Female forth spread=14.25

Graphical display: Boxplot

```
> quantile(M.life.exp,probs = c(0.25, 0.75))
     25%      75%
   59.75    69.50
> quantile(F.life.exp,probs = c(0.25, 0.75))
     25%      75%
   63.00    77.25
> diff(quantile(M.life.exp, probs = c(0.25, 0.75)))
9.75
> diff(quantile(F.life.exp, probs = c(0.25, 0.75)))
14.25
```

Variance, fourth-spread and MAD

- The panel below shows the spread estimators for the two samples in the SAT dataset.

```
> #Sx, Q3-Q1 and MAD for the SAT dataset
                Urba    rural    ratio (urban/rural)
1    Sx       176.57   120.37                1.466894
2    Q3-Q1    277.00    85.00                3.258824
3    MAD      149.00    47.00                3.170213
```

Variance, fourth-spread and MAD

- The ratio between the MAD of the two samples, 149/47 = 3.17, indicates that the dispersion in the urban sample is about three times higher than the dispersion in the rural sample.
- The fourth-spread ratio, 277/85 = 3.26.
- The ratio between the samples standard deviations, 176.57/120.37 = 1.46, is about the half of the MAD and the fourth-spread ratios.

- The reason for that is the lack of resistance of the standard deviation against outliers. When the outlier score (500) was omitted from the rural sample the standard deviation was dropped to 82.2 and the ratio increase to 176.57/82.2 = 2.15.

Variance, fourth-spread and MAD

Hypothetical example

- The two hypothetical samples, identical except the maximum observation (80 in the first sample and 800 in the second).
- The ratios for the MAD and fourth-spread are both equal to 1. This indicates that the dispersion in the two sample is the same.
- On the other hand, the ratio between the standard deviations is 266.01/19.56 = 13.59.

```
> x1 <- c(24,35,39,50,60,60,75,80) # sample 1
> x2 <- c(24,35,39,50,60,60,75,800) # sample 2
> spread.ratio
                    1          2      ratio:2/1
1    Sx        19.560   266.0131        13.59295
2    Q3-Q1     25.750    25.7500         1.00000
3    MAD       18.000    18.0000         1.00000
```

Practical session 2

Exercise 1

1. a) Draw random sample (n=100) from N(0,1), calculate the variance, MAD and interquartile range.

```
x <- rnorm(100,0,1)
var(x)
mad(x)
quantile(x,prob=c(0.25,0.75))
diff(quantile(x,prob=c(0.25,0.75)))
```

Exercises

b) Add to the sample very large number and calculate again the summaries for spread. The variance changed much more than the MAD and the interquartile range. Why ?

```
x <- c(x,50)
var(x)
mad(x)
quantile(x,prob=c(0.25,0.75))
diff(quantile(x,prob=c(0.25,0.75)))
```

Exercises 2

2. Comparing distributions with qqplot and boxplot

a) Normal model: draw two samples from normal distribution, N(0,1) and N(2,1).

What is the difference between the two distributions? Try to make a figure of the true model (BY HAND).

b) Calculate the 5 Quantiles, and produce a boxplot of the two samples.

The quantiles of X1 are higher than the quantiles of X2.

How came?

```
quantile(x1)
quantile(x2)
boxplot(x1,x2)
```

Exercises

c) Plot boxplot and qqplot in one page.

```
par(mfrow=c(1,2))
boxplot(x1,x2)
qqplot(x1,x2)
abline(0,1)
```

Chapter 3
Resistance

Introduction

- **Resistance**: A resistant estimator is one that is not influenced by extremely high or low data values (outliers).
- I.e., resistant estimator pay much attention to the main body of the data and little to outliers.
- A resistant estimator produces results that change only slightly when a small part of the data is changed.

- For instance: The mean is not a resistant estimator of location because we can make the mean as large as we want by changing the size of only one data value.
- The median, on the other hand, is more resistant.

Resistance: Mean vs Median

- **Mean:**
- The sample mean is simply the arithmetic average of the observations:

$$\overline{X} = \frac{x_1 + x_2 + \ldots + x_n}{n} = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i$$

- Merits: natural, easy to compute, has nice mathematical properties
- Demerit: has no resistance to extreme values or outliers

- If any value $x_k \to \infty$, mean will be affected.

Resistance: Mean vs Median

- **Median**:
- The middle value of a sample data, $\tilde{X}$

- Merit: It is resistant estimator to extreme values
- Demerits: It is not sensitive for change of a specific value.
- Depends only on the values of the middle observations and not sensitive to the extreme values.
- The median will tolerate up to 50% gross errors on the data values.

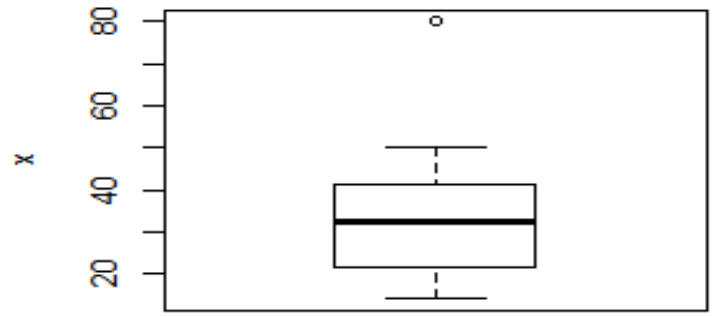
Resistance: Mean vs Median

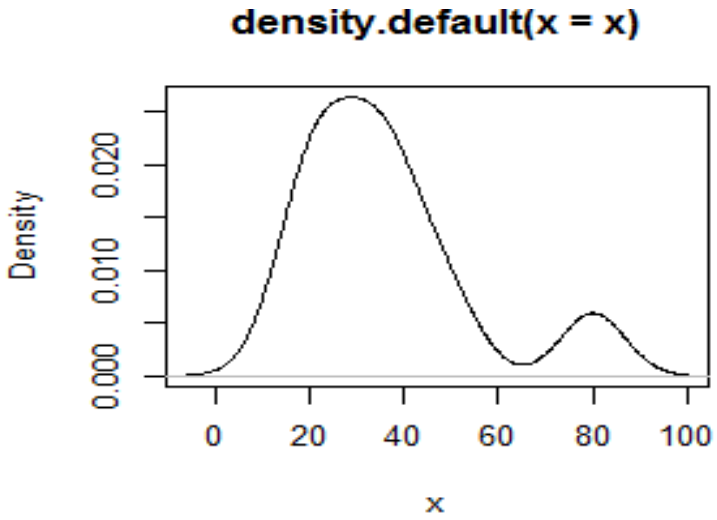
- Example: Class size for Introductory biostatistics course in 20 departments in Jimma University.

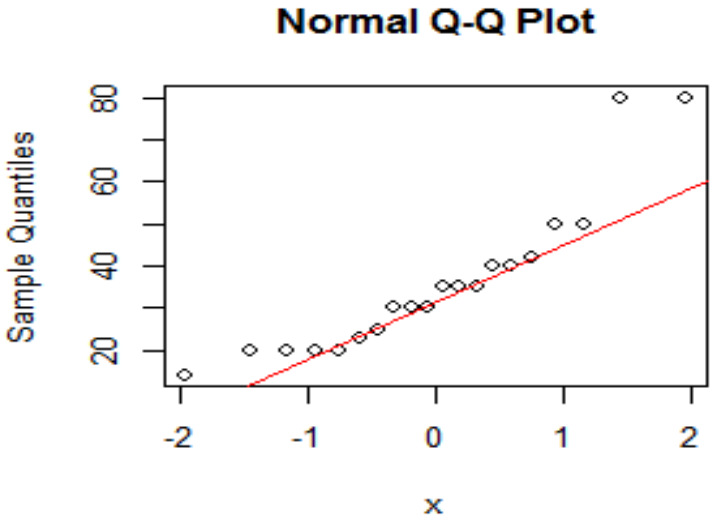
```
x<-c(14,20,20,20,20,23,25,30,30,30,35,35,35,40,40,42,50,50,80,80)
mean(x)
[1] 35.95
median(x)
[1] 32.5
```

Resistance: Mean vs Median

- Box-plot, density plot and Q-Q normal probability plot







Resistance: Mean vs Median

- We can clearly see the extreme values for the given data on the plots.

```
> x<-c(14,20,20,20,20,23,25,30,30,30,35,35,35,40,
  40,42,50,50,80,80)
> par(mfrow=c(2,2))
> boxplot(x,ylab="x")
> plot(density(x),xlab="x")
> qqnorm(x,xlab="x")
> qqline(x,col=2)
```

Resistance: Mean vs Median

- Exclude the extreme values and recompute mean and median to see effects.
- Discard one observation from the end. Both mean and median are changed (originally: mean 35.95 and median 32.5).

```
x.1<c(14,20,20,20,20,23,25,30,30,30,35,35,35,40,40,
  42,50,50,80)
mean(x.1)
[1] 33.63
median(x.1)
[1] 30
```

Resistance: Mean vs Median

- If we exclude 14 from the beginning and 80 from the end

```
x.2<c(20,20,20,20,23,25,30,30,30,35,35,35,40,40,42,
  50,50,80)
> mean(x.2)
[1] 34.72
> median(x.2)
[1] 32.5
```

- The mean has changed, but median value now is not changed (originally: mean 35.95 and median 32.5).
- Why?

Resistance: Trimmed mean

- **Trimmed mean**

- Measure designed to address the sensitivity of the sample mean to extreme observations.
- A measure of center that is more resistant than the mean.
- Is also a simple way to delete outliers in the observed data x.

- Compute 5% trimmed mean for the above data

```
mean (x, trim=0.05)
[1] 34.722
```

Resistance: Trimmed mean

- Note: 0% and 50% trimmed mean are equal to sample mean and median respectively.

```
mean(x, trim=0)
[1] 35.95
mean(x, trim=0.5)
[1] 32.5
```

L-estimators

Resistance: L-estimators

L-estimators

- The difference between mean, median and trimmed mean is related to their resistance against extreme values and change in the data values.
- The weights of the estimator playing an important role in resistance. For L-estimators

$$T_L = \sum_{i=1}^{n} a_i X_{(i)}$$

Resistance: L-estimators

- Sample mean: all the weights are all equal to 1/n.
- All the observations influence the value of the estimator with the same weight.

- Median: only 1 or 2 observations have positive weights.
- Only this middle part influence the value of estimator.

- Trimmed mean: all observations after trimming have same weights.
- Trimmed observations do not have any influence at all.

Resistance: Weights

- Consider a sample of the 15 largest cities in the US in 1960
- Original number of inhabitants is divided by 10,000.

```
> pop.15 <-c(778,355,248,200,167,94,94,88,76,75,74,74,70,68,63)
> sort(pop.15)
[1] 63 68 70 74 74 75 76 88 94 94 167 200 248 355 778
```

Resistance: Weights

- The mean of the population of the 15 largest cities is 1,682,667, the median of the population is 880,000.

```
> mean(pop.15)
[1] 168.2667
> median(pop.15)
[1] 88
```

- The reason for this difference is the unusual large population in New-York (7,780,000) and Chicago (3,550,000).
- These two extreme values push the mean towards large values. The median, on the other hand, is not influenced by these extreme values (the weights are 0!).

Resistance: Weights

- For the 10%-trimmed mean, that is the average of the 13 cities without New York (7,780,000) and New Orleans (630,000), the mean drops to 1,294,615.
- The 20%-trimmed mean (Chicago and Dallas with 3,550,000 and 680,000 are further excluded) drops further 1,046,670, much closer to median.

```
> mean(pop.15,trim=0.1)
[1] 129.4615
> mean(pop.15,trim=0.2)
[1] 104.6667
```

Resistance: Weights

- Note that the 0%-trimmed mean is simply the sample mean and the 50%-trimmed mean is the sample median.

```
> mean(pop.15,trim=0)
[1] 168.2667
> mean(pop.15,trim=0.5)
[1] 88
```

Resistance: Weights

- Let us focus on the two largest observations in the dataset:

  New-York with a population of 7,780,000 and Chicago with a population of 3,550,000.
- The table below shows the contribution of these two observations to the estimators.

Location estimator	Weight for 778	Weight for 355
Mean	1/15	1/15
Mean (α=0.1)	0	1/13
Mean (α=0.2)	0	0
Median	0	0

Resistance: Weights

- Mean: the weights for both cities are 1/15, so the contributions to the mean are (1/15)*(778) and (1/15) * (355).
- 10%-trimmed mean: the smallest and largest observations are trimmed. Hence the weight of 778 is 0 and the weight of 355 is 1/13.
- 20%-trimmed mean and median: the weights are 0 for both observations and therefore the two largest values do not influence these estimators.

Location estimator	Weight for 778	Weight for 355
Mean	1/15	1/15
Mean (α=0.1)	0	1/13
Mean (α=0.2)	0	0
Median	0	0

Resistance: Breakdown point

Breakdown Point

- The breakdown point of an estimator is the proportion of incorrect observations (i.e. arbitrarily large observations) an estimator can handle before giving an infinite result.

- Example, for the given sample data

  $x_1, x_2, x_3, \ldots, x_n$, the arithmetic mean is calculated as

$$\overline{X} = \frac{x_1 + x_2 + \ldots + x_n}{n} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- This estimator has a breakdown point of 0, because we can make arbitrarily large just by changing any of $x_1, x_2, x_3, \ldots, x_n$

Resistance: Breakdown point

- The higher the breakdown point of an estimator, the more robust or resistant it is.

- A breakdown point cannot exceed 50% because if more than half of the observations are contaminated, it is not possible to distinguish between the underlying distribution and the contaminating distribution.
- Therefore, the maximum breakdown point is 0.5 and there are estimators which achieve such a breakdown point.

- For example, the median has a breakdown point of 0.5. The $\alpha\%$ trimmed mean has breakdown point of $\alpha\%$.

Sensitivity curve

Resistance: Sensitivity curve

- The sensitive curve of the estimator T, defined for {1,2,3,…,n}, at the sample x_1, x_2, x_3, …, x_{n-1} is

$$SC(x, x_1, \ldots, x_{n-1}, T) = T_n(x_1, \ldots, x_{n-1}, x) - T_{n-1}(x_1, \ldots, x_{n-1})$$

- Hence, the sensitive curve measures the change in T_n when a new observation is included in the sample.

- Note that the sensitivity curve do not focus of extreme values but measure the change of the estimator for any change in the data

Resistance: Sensitivity curve

- Mean with original sample:

$$T_{n-1}(x_1,\ldots,x_{n-1}) = \frac{1}{n-1}\sum_{i=1}^{n-1} x_i$$

- Mean with new observation:

$$T_n(x_1,\ldots,x_{n-1},x) = \frac{1}{n}\sum_{i=1}^{n-1} x_i + \frac{1}{n}x_n$$

- Sensitivity curve for mean:

$$SC(x,x_1,\ldots,x_{n-1},T) = T_n(x_1,\ldots,x_{n-1},x) - T_{n-1}(x_1,\ldots,x_{n-1})$$

$$= \frac{1}{n}x_n + \frac{1}{n(n-1)}\sum_{i=1}^{n-1} x_i$$

Resistance: Sensitivity curve

- Let us see the change for L-estimators due to new observation

```
> x<-c(14,20,20,20,20,23,25,30,30,30,35,
  35,35,40,40,42,50,50,80,80)
> mean(x)
[1] 35.95
> median(x)
[1] 32.5
> x.new <- c(12,x)
> x.new
 [1] 12 14 20 20 20 20 23 25 30 30 30 35 35 35 40
  40 42 50 50 80 80
> mean(x.new)
[1] 34.80952
> median(x.new)
[1] 30
```

- The mean and median changed for new value 12 in the data.

Resistance: Sensitivity curve

```
> x<-c(14,20,20,20,20,23,25,30,30,30,35,
     35,35,40,40,42,50,50,80,80)
> mean(x)
[1] 35.95
> median(x)
[1] 32.5
> x.new <- c(10,85,x)
> x.new
 [1] 10 85 14 20 20 20 20 23 25 30 30 30 35 35 35
  40 40 42 50 50 80 80
> mean(x.new)
[1] 37
> median(x.new)
[1] 32.5
```

- The median has not changed, but the mean is changed.

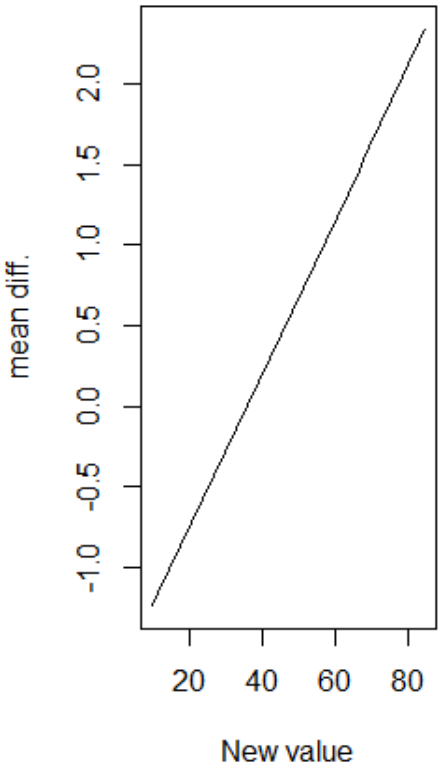
Resistance: Sensitivity curve

- Plot sensitivity curve for range of possible new observations.
- Add 50 new observations one by one to the original data set and see the new mean and median value for new set of data.
- The new observations are between 10 and 85.

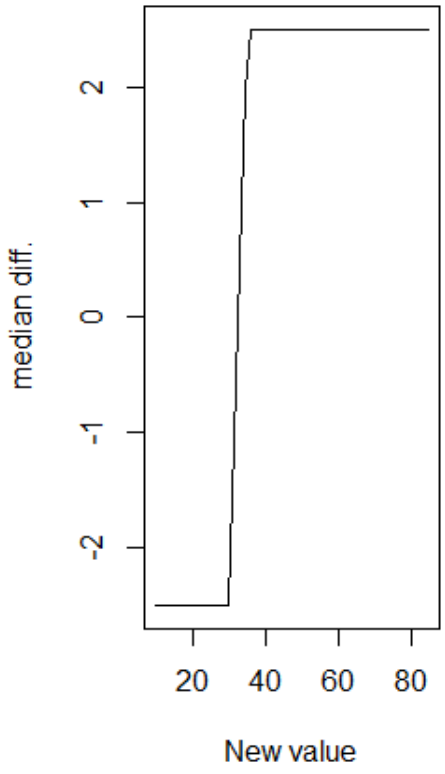
```
for(i in 1:50){
new sample = old sample +
      new observation [i]
compute new mean
compute new median
sensitivity. mean =
      new mean - mean
sensitivity. mean =
      new median - median
 }
```

Resistance: Sensitivity curve

- Sensitivity curve for mean and median of the given data



SC for the Mean



SC for the Median

The mean changes for any change in the data.

The median changes only when the data at the center of the sample are changed.

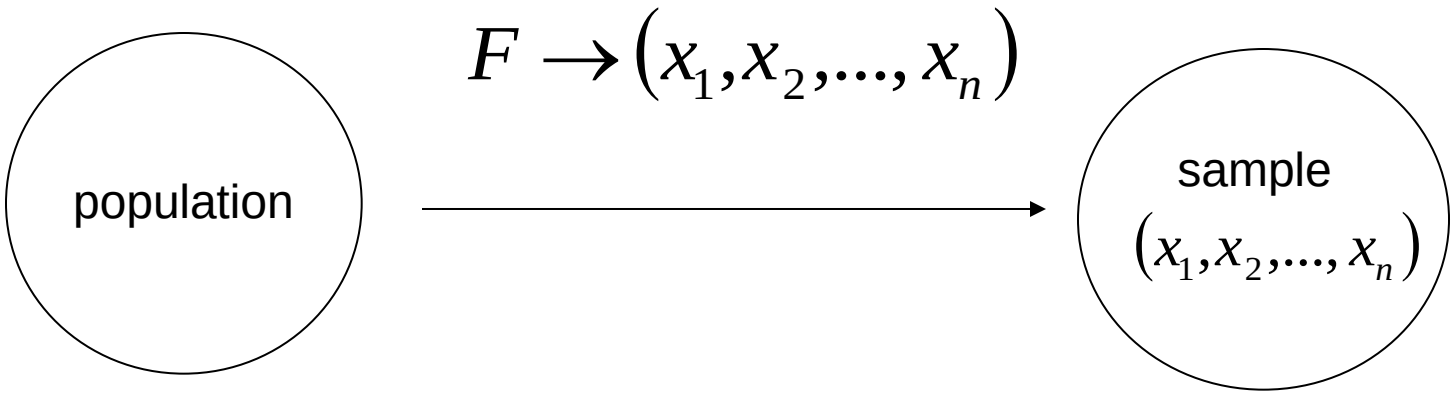
Resistance: Sensitivity curve

```
> mean <- mean(x)
> median <- median(x)
>  new.value <- seq(from=10,to=85,length=50)
>  mean.i <- c(1:50)
>  median.i <- c(1:50)
>  for(i in 1:50){
+  x.new <- c(new.value[i],x)
       # new.value[i] is the "new observation"
+  mean.i[i] <- mean(x.new)-mean
+  median.i[i] <- median(x.new)-median
+ }
>  par(mfrow=c(1,2))
>  plot(new.value,mean.i,type="l",xlab="New
  value",ylab="mean diff.")#mean difference
>  plot(new.value,median.i,type="l",xlab="New
  value",ylab="median diff.")#median difference
```

Chapter 4
Robustness

Introdution

- In the previous chapter we focused on the influence of single observation on the estimator.
- In this chapter we focus on distributions.
- We observed random sample from the probability distribution *F*.



- Resistance: how sensitive is the estimator for change in the data.
- Robustness : how sensitive is the estimator for change in the distribution of the data.

Introdution

- In the previous chapter we focused on the influence of single observation on the estimator.
- In this chapter we focus on distributions.

- Resistance: how sensitive is the estimator for change in the data.
- Robustness : how sensitive is the estimator for change in the distribution of the data.

Introdution

- Which estimator is the best when data are drawn from normal distribution ? (the sample mean)
- Does the best estimator for location for normal distribution is also the best estimator for location for heavy-tailed distribution ? (No)
- But what if the data is not normally distributed ?
- How robust is the estimator against distribution assumptions ?

Graphical Display of distribution: Q-Q plot

- Example data:

```
> x<-c(14,20,20,20,20,23,25,30,30,30,
    35,35,35,40,40,42,50,50,80,80)
> hist(x)
> plot(density(x),xlab="x")
> qqnorm(x)
> qqline(x,col=2)
```

- Explore data graphically to conclude about its distribution:

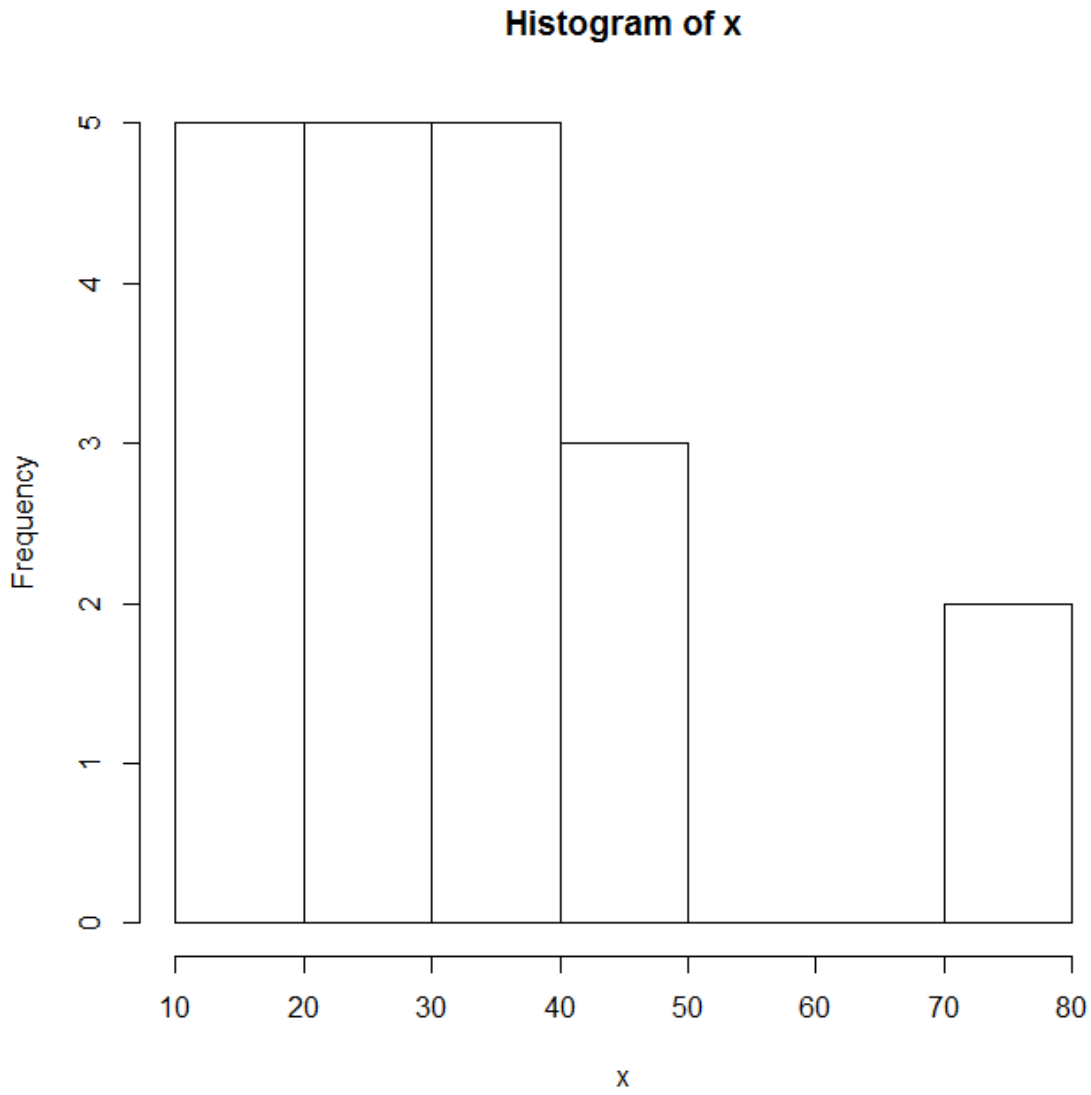
  Histogram

  Density plot

  Q-Q plot

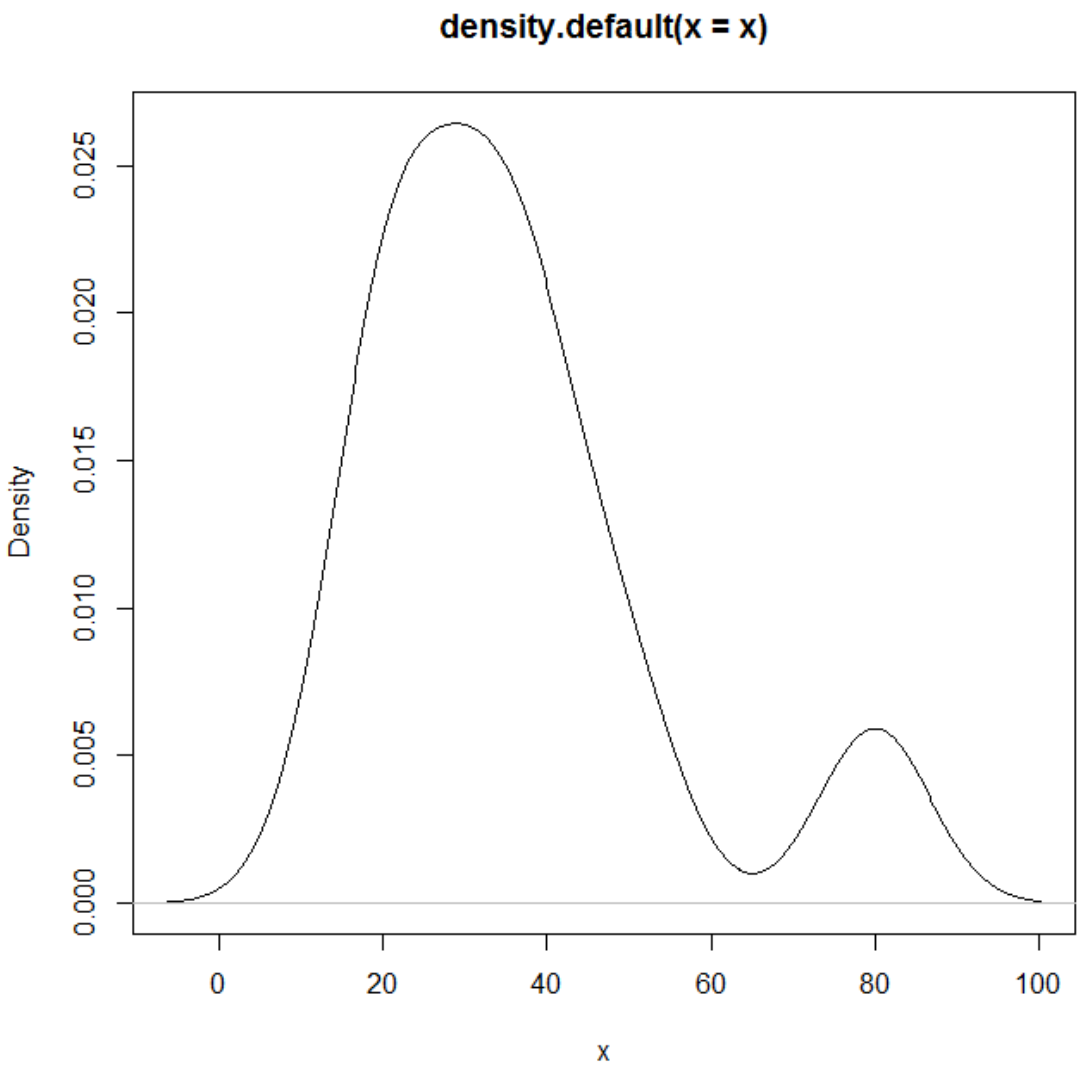
Graphical Display of distribution: Histogram

- Histogram



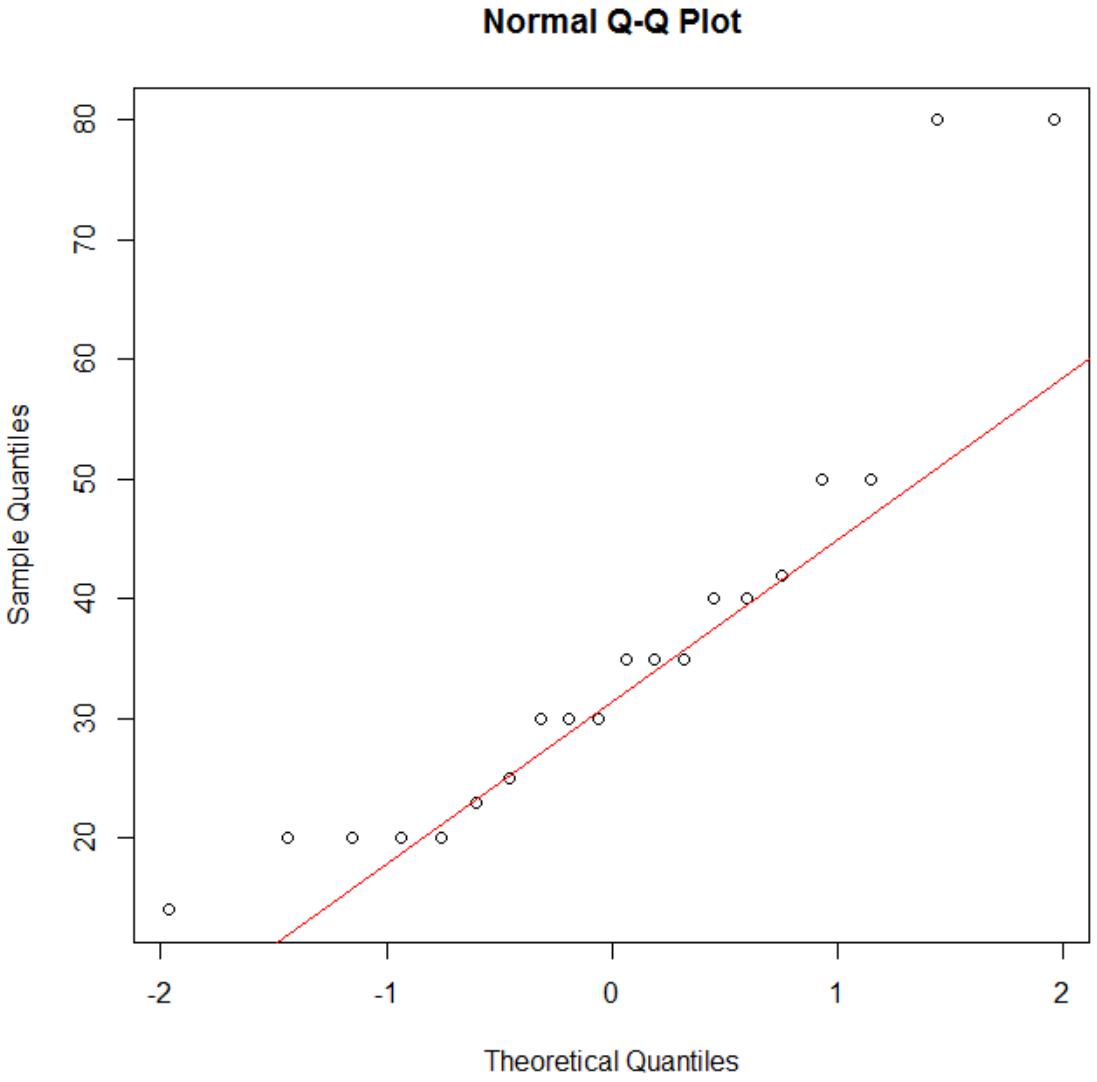
Graphical Display of distribution: Density plot

- Density plot



Graphical Display of distribution: Q-Q plot

- Density plot



Graphical Display of distribution

- **Histogram** shows data distribution of data.
- It is quite granular, because we have just few unique values in our data set and they repeat several times.
- Several observations appear far on the right side.

- **Density plot** is smooth version of histogram.
- Estimated density again shows two peaks.
- It suggests that the data does not come from normal distribution.

- **Q-Q plot** compare quantiles of normal distribution with quantiles of the data set.
- Deviation from straight line suggests non-normality.

Robustness: Contaminated Normal Distribution

- Contaminated normal distribution is a mixture of two normal distribution with means 0 but different variances.
- The contaminated normal probability density function is given as:

$$P(x) = p \cdot N(0, sd_1) + (1 - p) \cdot N(0, sd_2)$$

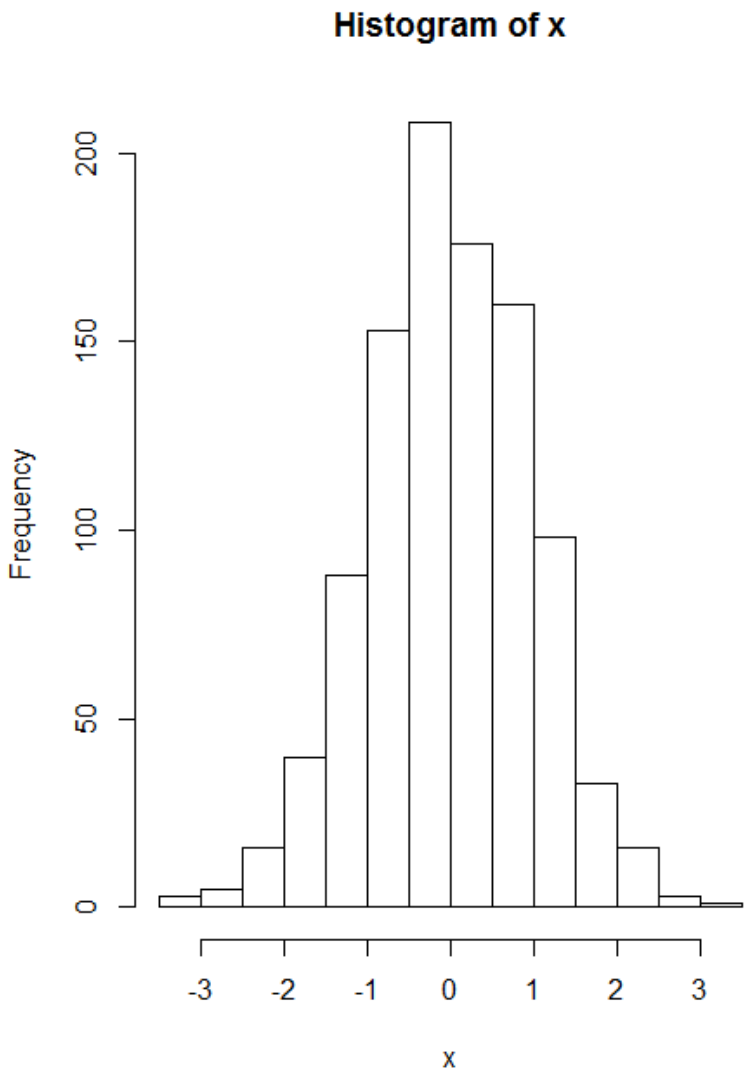
p is contamination percentage

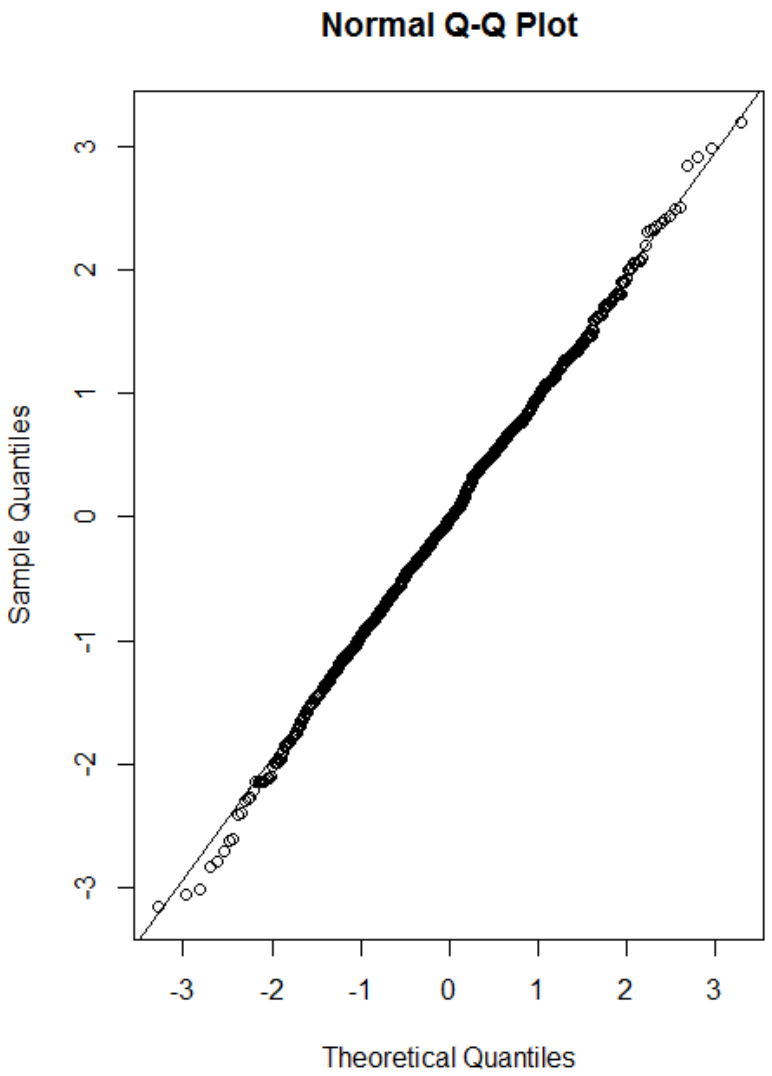
sd_i is the i^{th} standard deviation

- Let us see standard normal distribution (no contamination) compared to the 5% (p=0.05) contaminated normal probability distribution with sd_1=10, sd_2=1.

Robustness

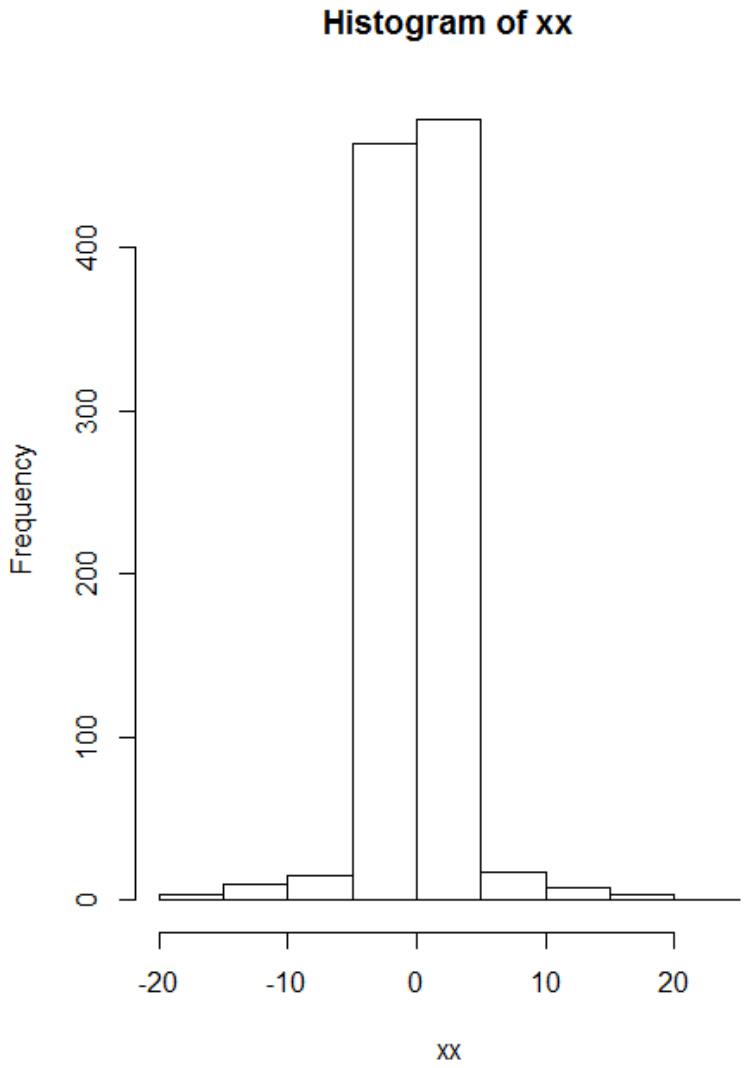
Standard normal distribution

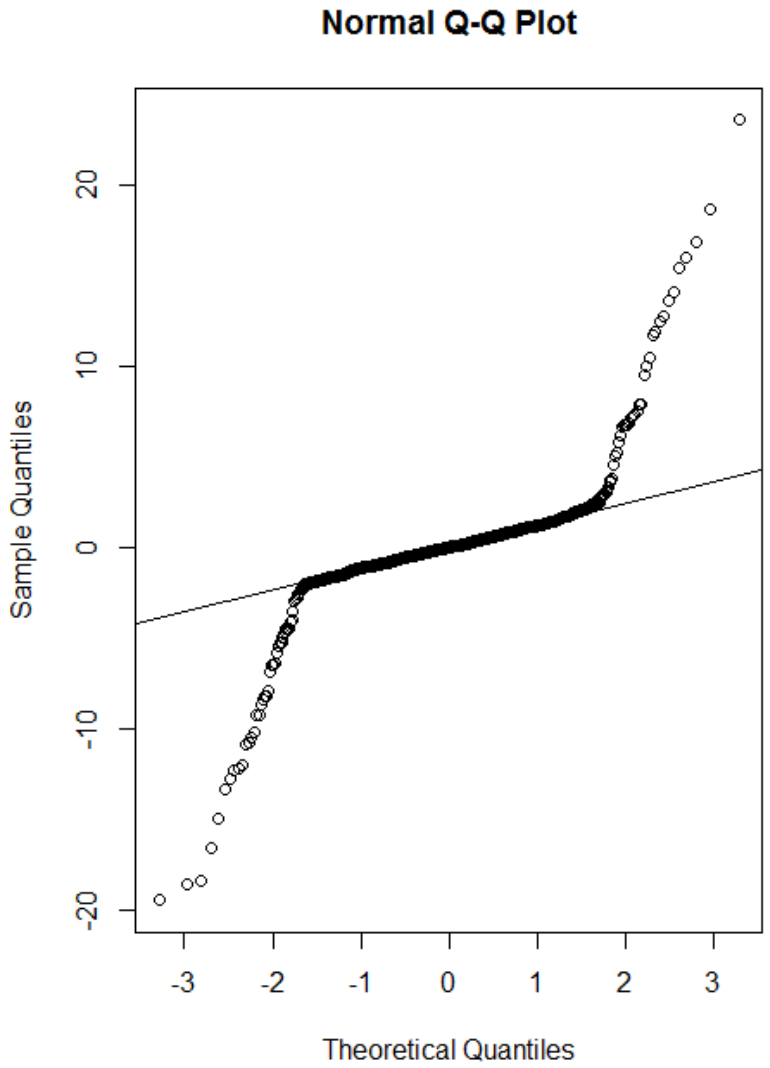




Robustness

5% contaminated normal distribution





Robustness

```
> par(mfrow=c(1,2))
> x <- rnorm(1000,0,1)
> hist(x)# not contaminated
> qqnorm(x)
> qqline(x)

# For contaminated
> par(mfrow=c(1,2))
> x1 <- rnorm(900,0,1)
> x2 <- rnorm(100,0,10)
> xx <- c(x1,x2)   # contaminated
> hist(xx)
> qqnorm(xx)
> qqline(xx)
```

Robustness: Relative efficiency

Relative Efficiency

- The relative efficiency of two estimator T_1 and T_2 on samples from distribution F is

$$RE(T_1, T_2, F) = \frac{Var(T_1, F)}{Var(T_2, F)}$$

- For two estimators, if $RE(T_1, T_2, F) < 1$ then $Var(T_1, F) < Var(T_2, F)$.
- This means that T_1 is better estimator than T_2 for the distribution F.

Robustness: Relative efficiency

- **Relative efficiency of location estimators for standard normal distribution**

- Let us compare mean, median and trimmed mean:

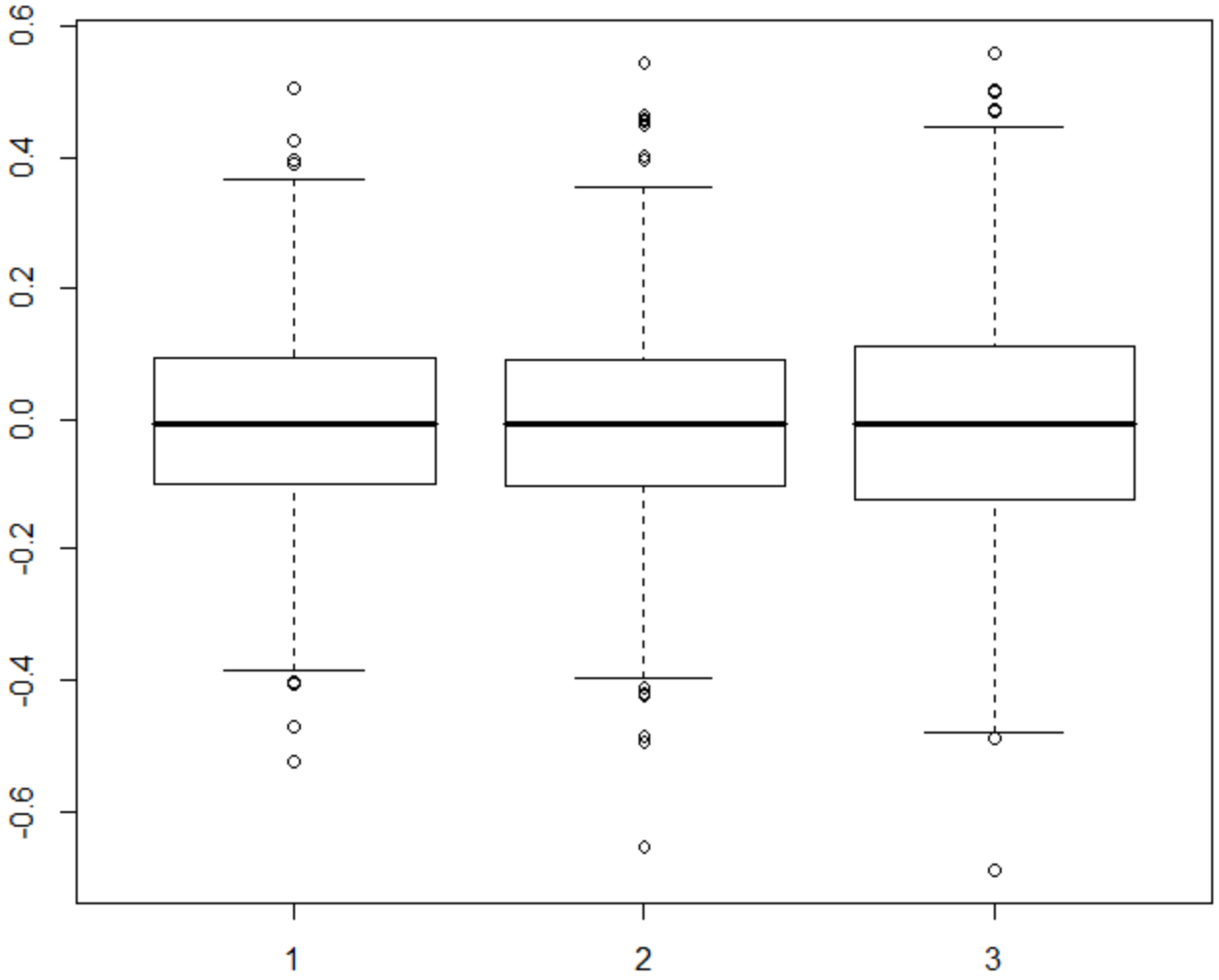
1. Generate 1000 random data sets with 50 observations each from standard normal distribution (no contamination).
2. Generate 1000 random data sets with 50 observations each from the 10% (p=0.1) contaminated normal probability distribution with sd_1=10, sd_2=1.
3. Compute ratio of variances for var(mean)/var (median) and var(mean)/var(trimmed) for both distributions.
4. Draw box plot for mean, median and trimmed mean for both distributions.

Robustness: Relative efficiency

```
> # Standard normal distribution
> x.mean <- x.mean.t <- x.med <- c(1:1000)
> for(i in 1:1000){
+ x.1 <- rnorm(50,0,1)
+ x.mean[i] <- mean(x.1)
+ x.mean.t[i] <- mean(x.1,trim=0.2)
+ x.med[i] <- median(x.1)
+ }

> var(x.mean)/var(x.med)
[1] 0.6496134
> var(x.mean)/var(x.mean.t)
[1] 0.8757091
> boxplot(x.mean,x.mean.t,x.med)
```

Robustness: Relative efficiency

- Standard normal distribution

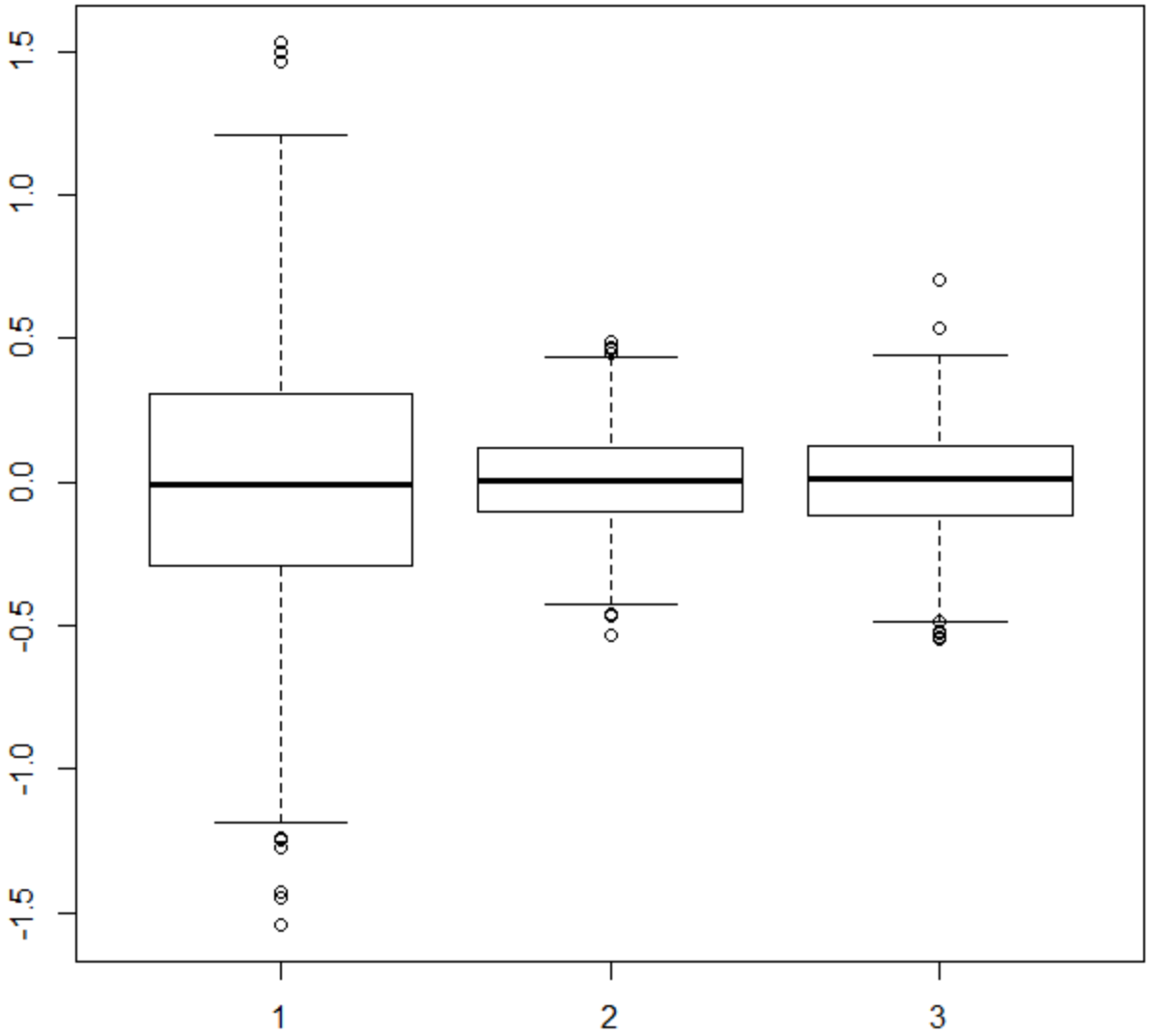


Robustness: Relative efficiency

```
> # Contaminated normal distribution
x.mean <- x.mean.t <- x.med <- c(1:1000)
> for(i in 1:1000){
+ x.1 <- rnorm(45,0,1)
+ x.2<-rnorm(5,0,10)
+ xx=c(x.1,x.2)
+ x.mean[i] <- mean(xx)
+ x.mean.t[i] <- mean(xx,trim=0.2)
+ x.med[i] <- median(xx)
+ }
> var(x.mean)/var(x.med)
[1] 5.959209
> var(x.mean)/var(x.mean.t)
[1] 7.517097
> boxplot(x.mean,x.mean.t,x.med)
```

Robustness: Relative efficiency

10% contaminated normal distribution.



Robustness: Relative efficiency

- **Standard normal distribution:**

  Variance of the mean is smaller than the variance of median and trimmed mean.

  Relative efficiency of the mean is smaller than 1.

  Hence, sample mean is most efficient location estimator for mean.

- **10% contaminated normal distribution:**

  Variance of the mean is much higher than the variance of median and trimmed mean.

  Relative efficiency of the mean is far above 1.

  Hence, sample mean is not robust against contamination.

  Median is doing much better job.

  Trimmed mean works well too, because we trimmed 10% at each side, which trimmed out contaminated part of distribution.

Robustness: Relative efficiency

- What if the distribution is not normal and it has heavier tails (higher probability of very large or small values, than for normal distribution)?

- Let us investigate Student's t distribution with 3 degrees of freedom:

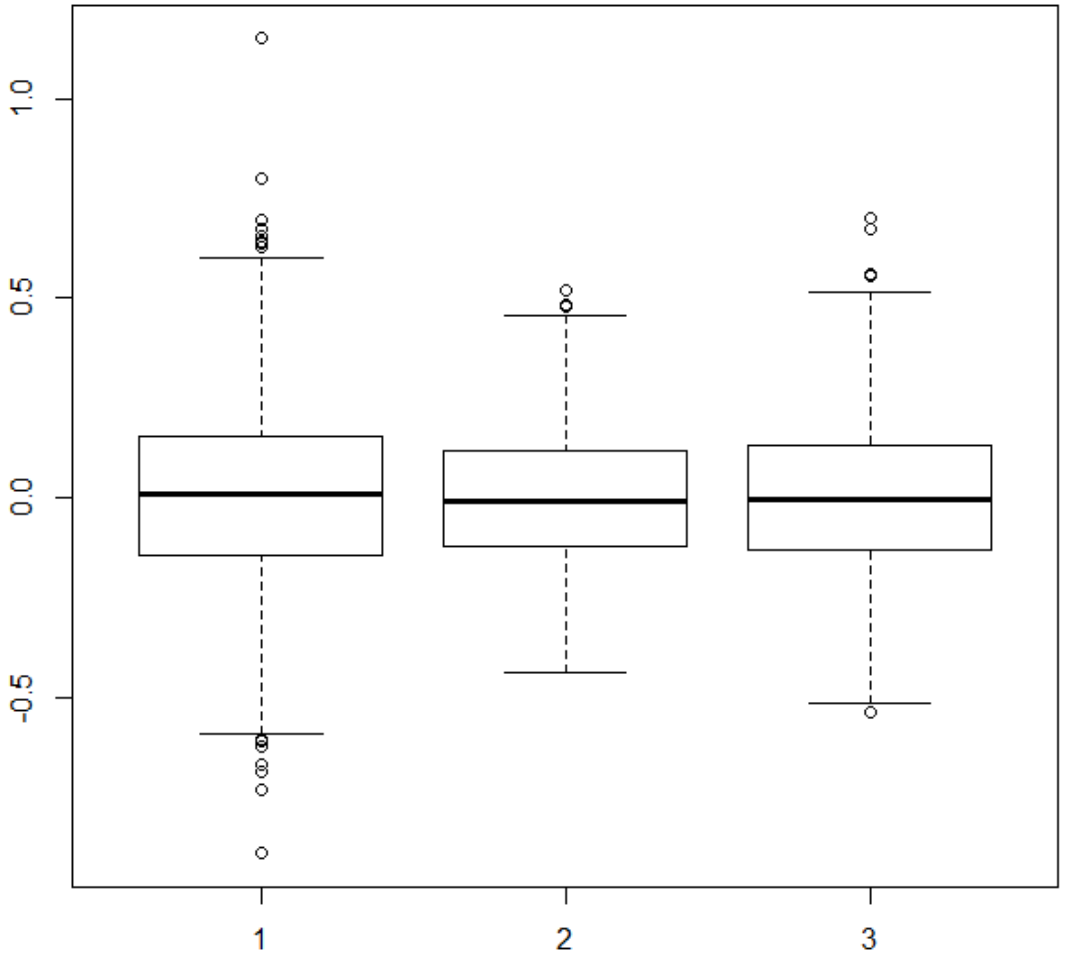
1. Take 50 sample from $t_{(3)}$.
2. Repeat it 1000 times.
3. Calculate the variance mean, median and trimmed mean.
4. Draw box plot for mean, median and trimmed mean.

Robustness: Relative efficiency

```
> # t(3) distribution
x.mean <- x.mean.t <- x.med <- c(1:1000)
> for(i in 1:1000) {
+ x.1 <- rt(50,3)
+ x.mean[i] <- mean(x.1)
+ x.mean.t[i] <- mean(x.1,trim=0.2)
+ x.med[i] <- median(x.1)
+ }
> var(x.mean)/var(x.med)
[1] 1.547816
> var(x.mean)/var(x.mean.t)
[1] 1.881402
> boxplot(x.mean,x.mean.t,x.med)
```

Robustness: Relative efficiency

- Box plot for mean, trimmed mean and median of $t_{(3)}$



Robustness: Relative efficiency interpretation

- We can see that already for $t_{(3)}$ that is quite close to normal distribution, the variance of sample mean is high.

- Sample mean is not robust with respect to heavier tails.
- The same conclusion as for contaminated normal distribution.

- **Conclusion about sample mean:**

  When we believe in normality of our data, sample mean is most efficient estimator of location.

  When we have doubts about normality, sample mean is not robust and so misleading. Instead, we can use trimmed mean or median.

Practical session 3

Robustness: Exercises

1. For the different values of p (e.g. 0.05,0.1,0.15,0.2):
    1. Take a sample of size 1000 from a contaminated model:

        $$f(y) = p * N(0, sd=100) + (1-p) * N(0,1)$$

    2. Plot the histogram of the sample.
    3. Plot the Q-Q plot of the sample.
    4. Evaluate graphically normality or non-normality of the sample.

Robustness: Exercises

2. For the different values of p (e.g. 0.05,0.1,0.15,0.2):
    1. Take 1000 samples of size 50 from a contaminated model:

       $$f(y)= p * N(0, sd=100) + (1-p) * N(0,1)$$

    2. For each one of the samples calculate the mean, the trimmed mean T(0.2) and the median.
    3. Plot the boxplot of the three estimators.
    4. Can you see any difference between the three estimators?

.

PART 2:
An Introduction to Bootstrap Methods

Outline

- Chapter 1: The empirical distribution function and the plug-in principle.
- Chapter 2: The basic bootstrap
  - Bootstrap estimate of the standard error for the mean.
  - The correlation coefficient.
- Chapter 3: Estimation of bias.

The empirical distribution function and the plug-in principle

The probability distribution

Let X be a random variable such that

$$X \sim F(\theta)$$

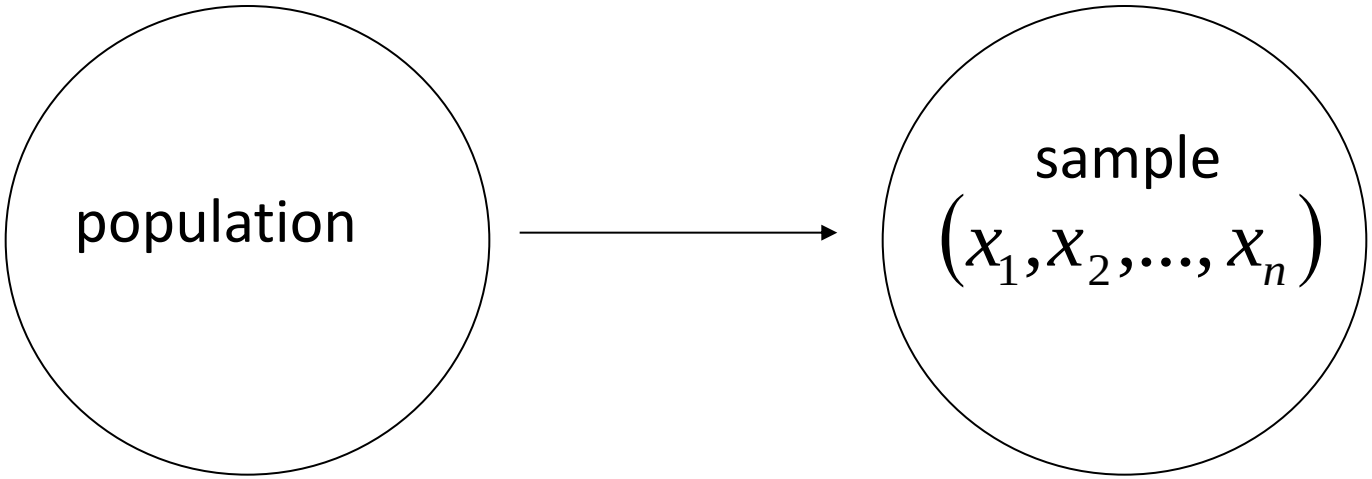
F is the probability distribution of X

θ is an unknown parameter

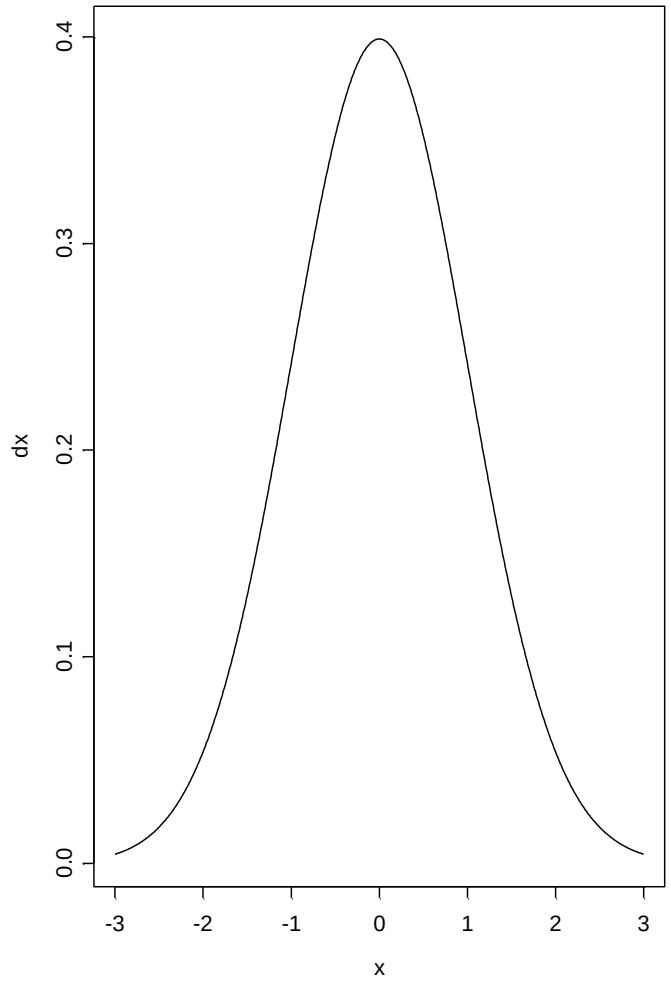
A random sample from F

We observed a random sample from the probability distribution F

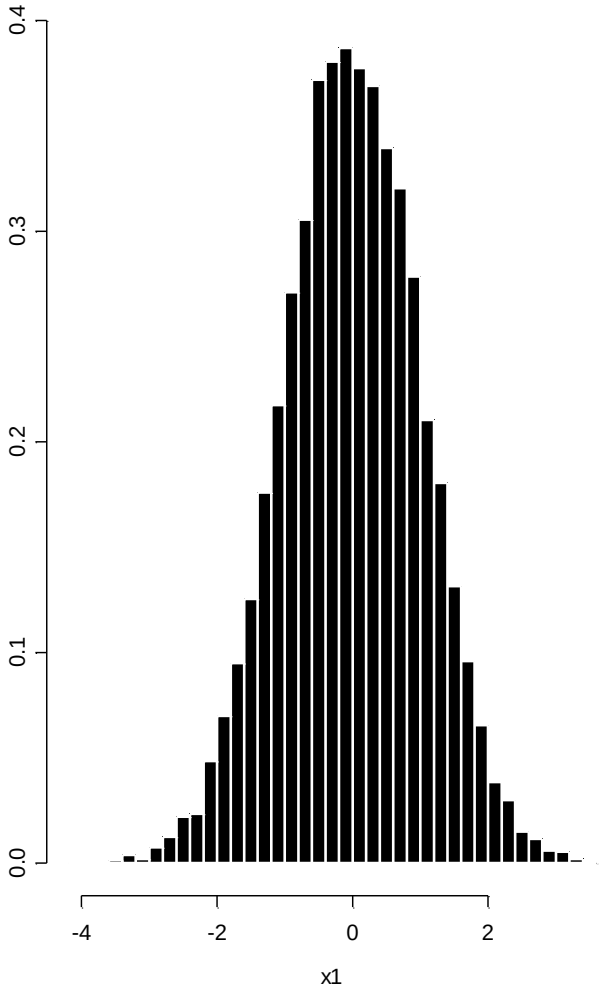
$$F \rightarrow (x_1, x_2, \ldots, x_n)$$



N(0,1)



A sample of 10000 observations from N(0,1)



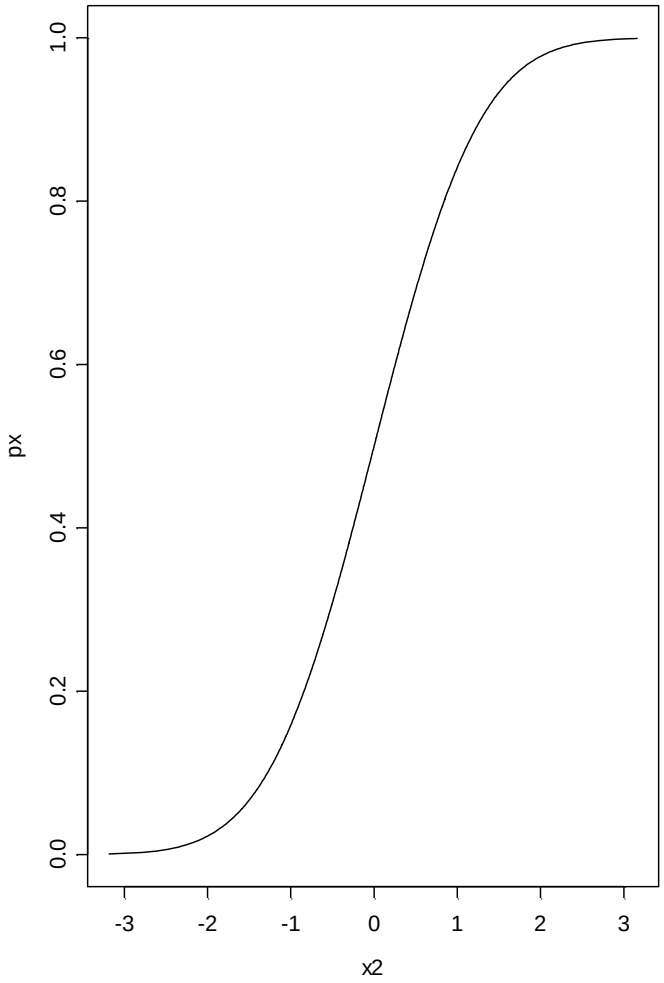
The empirical distribution

The empirical distribution function is defined to be the discrete distribution that puts probability of 1/n on each value of x_i

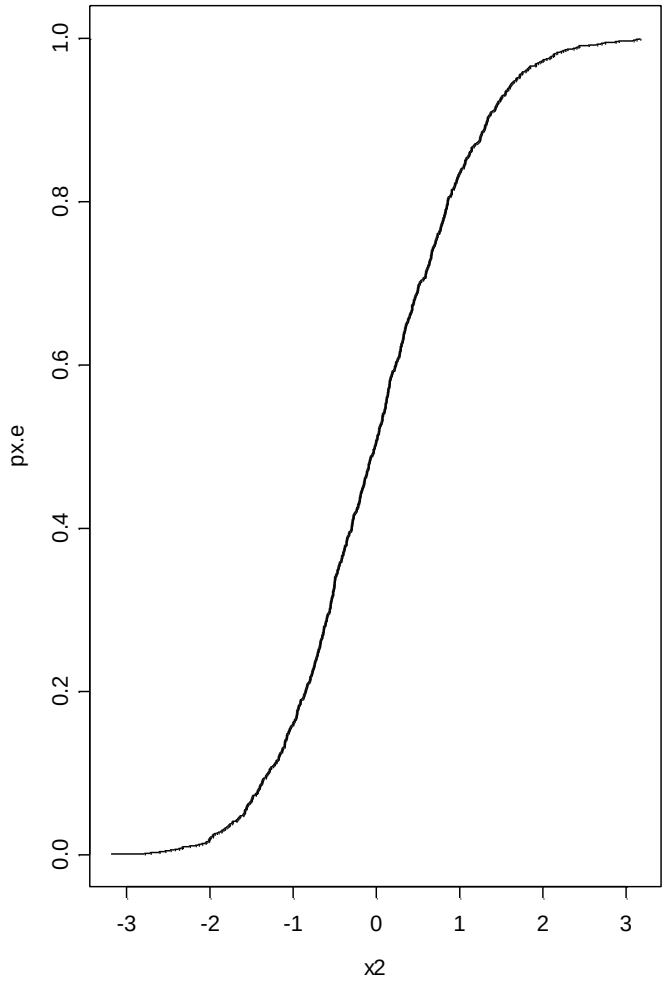
$$F \rightarrow \left(x_1, x_2, \ldots, x_n\right)$$

$$P(A) = \hat{F} = \frac{\#(x_i \in A)}{n}$$

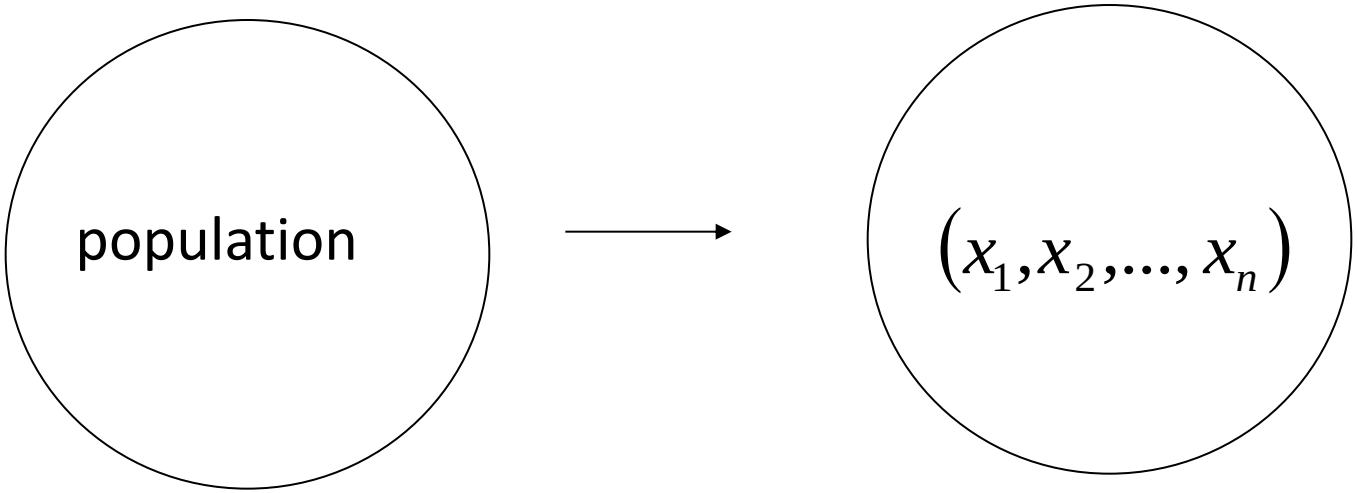
The probability distribution N(0,1)



The empirical probability distribution of a sample (n=500)



The empirical distribution



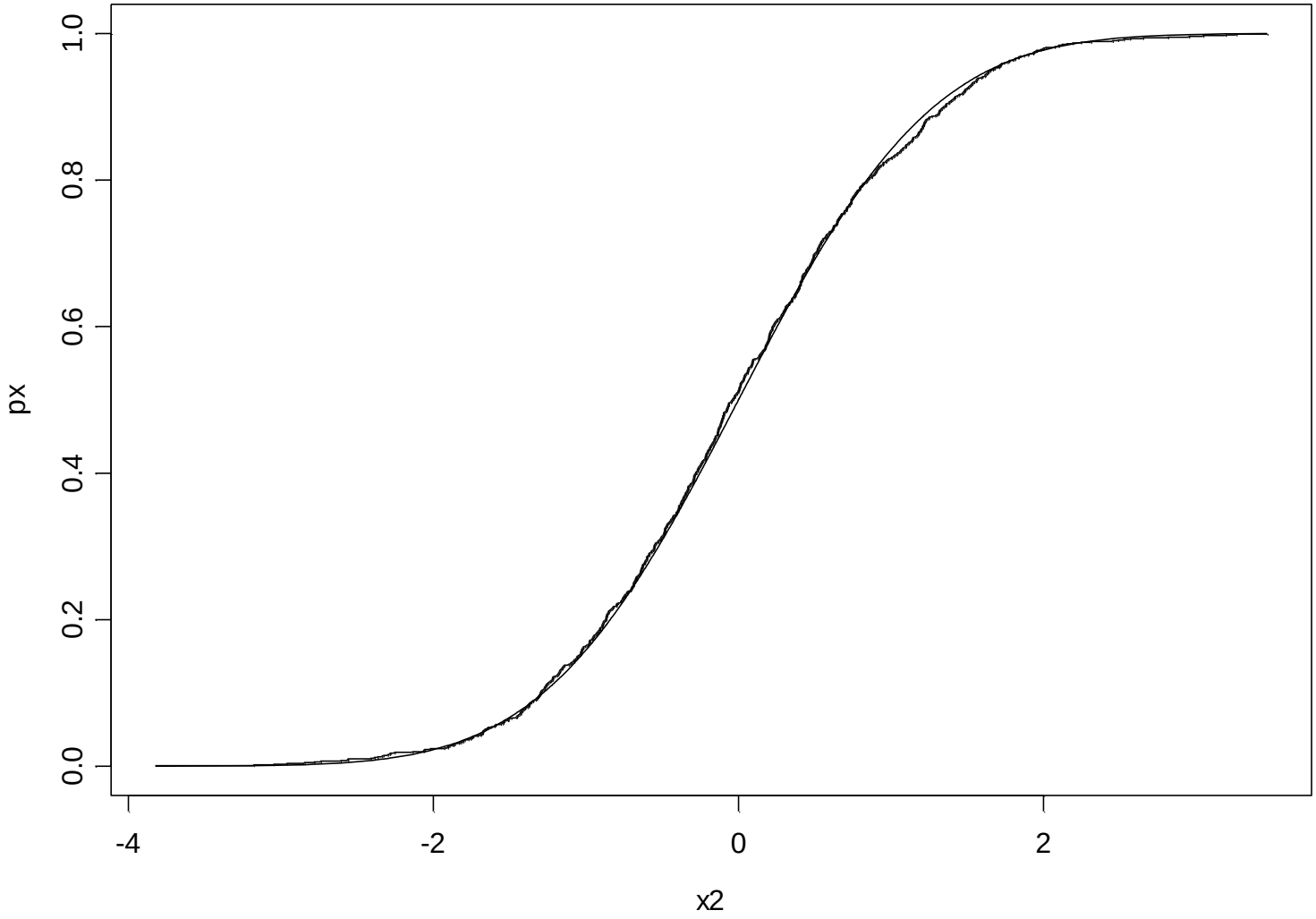
F

$\hat{F}$

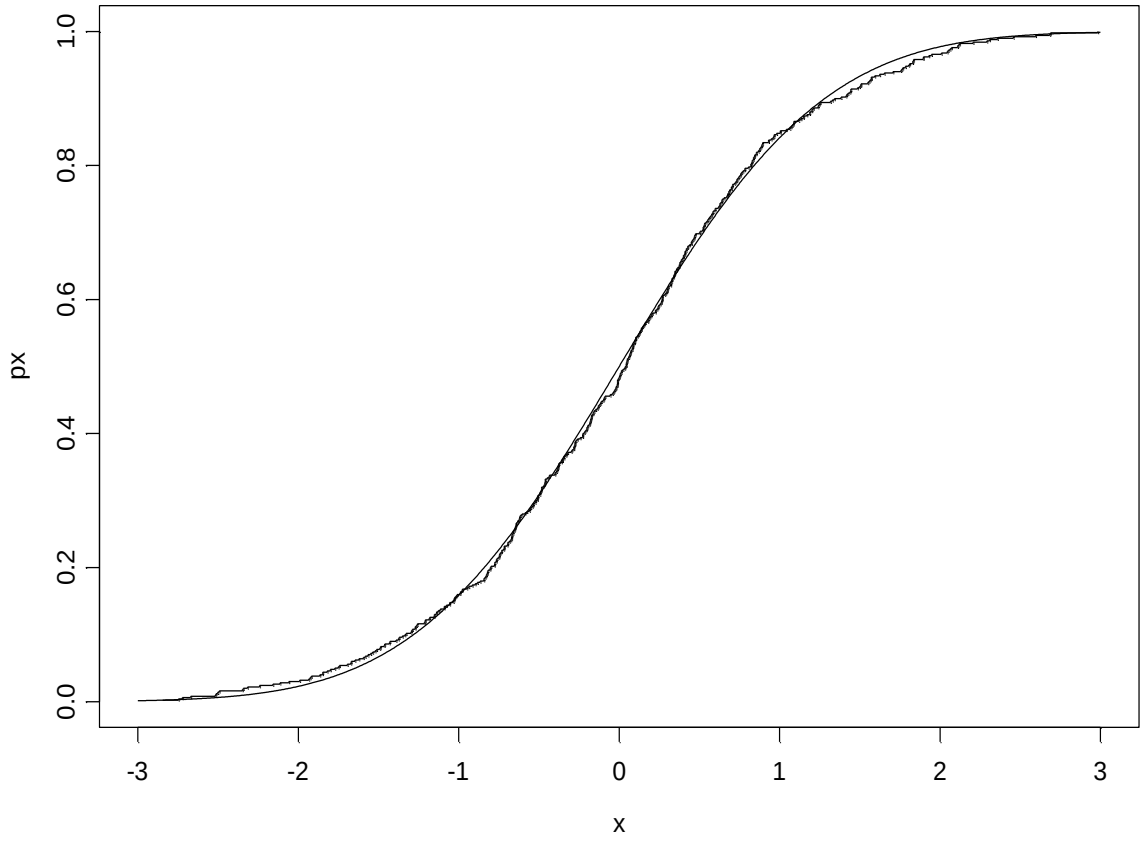
The probability distribution

The empirical probability distribution

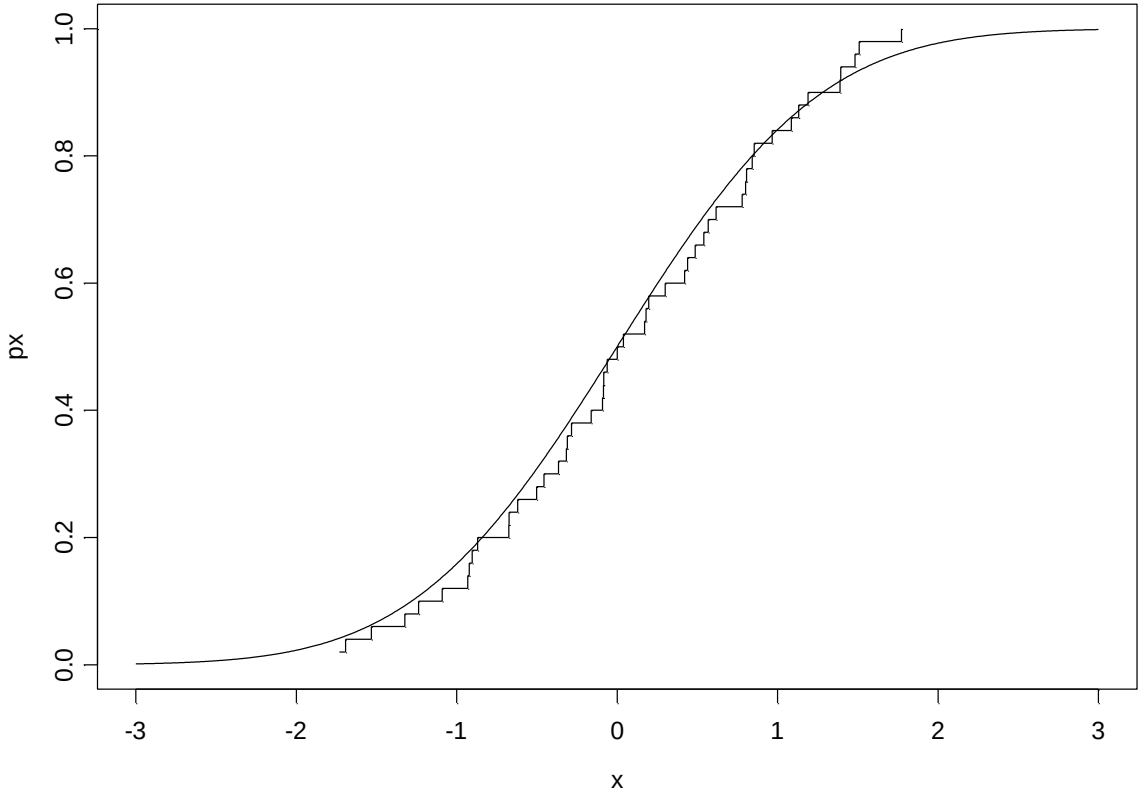
The empirical distribution



A sample of 500 observations from N(0,1)

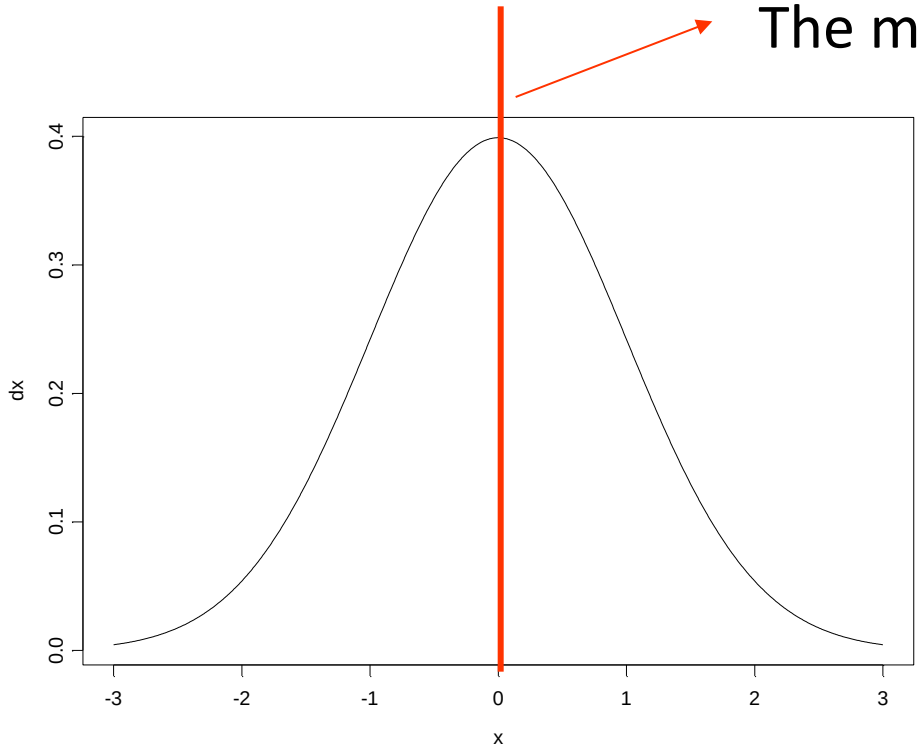


A sample of 50 observations from N(0,1)



A parameter

- A parameter θ is a function of the probability distribution F



$$\theta = t(F)$$

A statistic

parameter

$$\theta = t(F)$$

A random sample from F

$$F \rightarrow (x_1, x_2, \ldots, x_n)$$

A statistic is a function of the observed sample **x**

$$\hat{\theta} = t(\hat{F})$$

The mean for B(n,p)

$$F = B(n, p)$$

$$x_i = \begin{cases} 1 & p \\ 0 & (1-p) \end{cases}$$

$$\theta = E_F(x)$$

$$\theta = t(F) = np$$

The plug-in principle

The plug-in estimate of the parameter

$$\theta = t(F)$$

is defined as

$$\hat{\theta} = t(\hat{F})$$

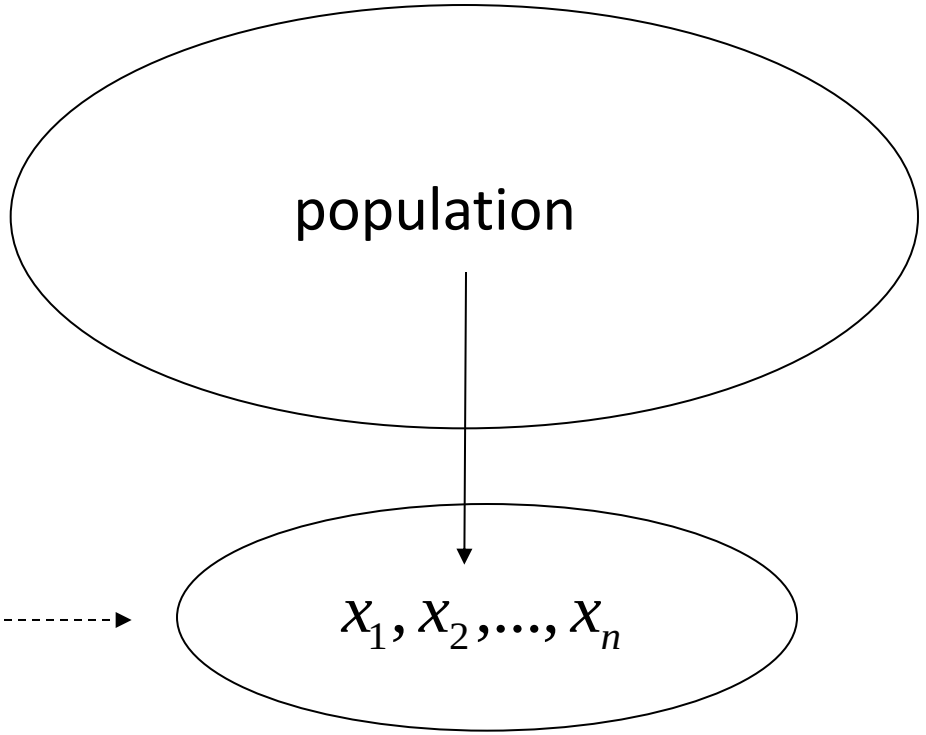
We use the same function from F, *t(F)* on the empirical distribution

The population mean and the parameter estimate from the sample

$$F = N(\mu, \sigma^2)$$

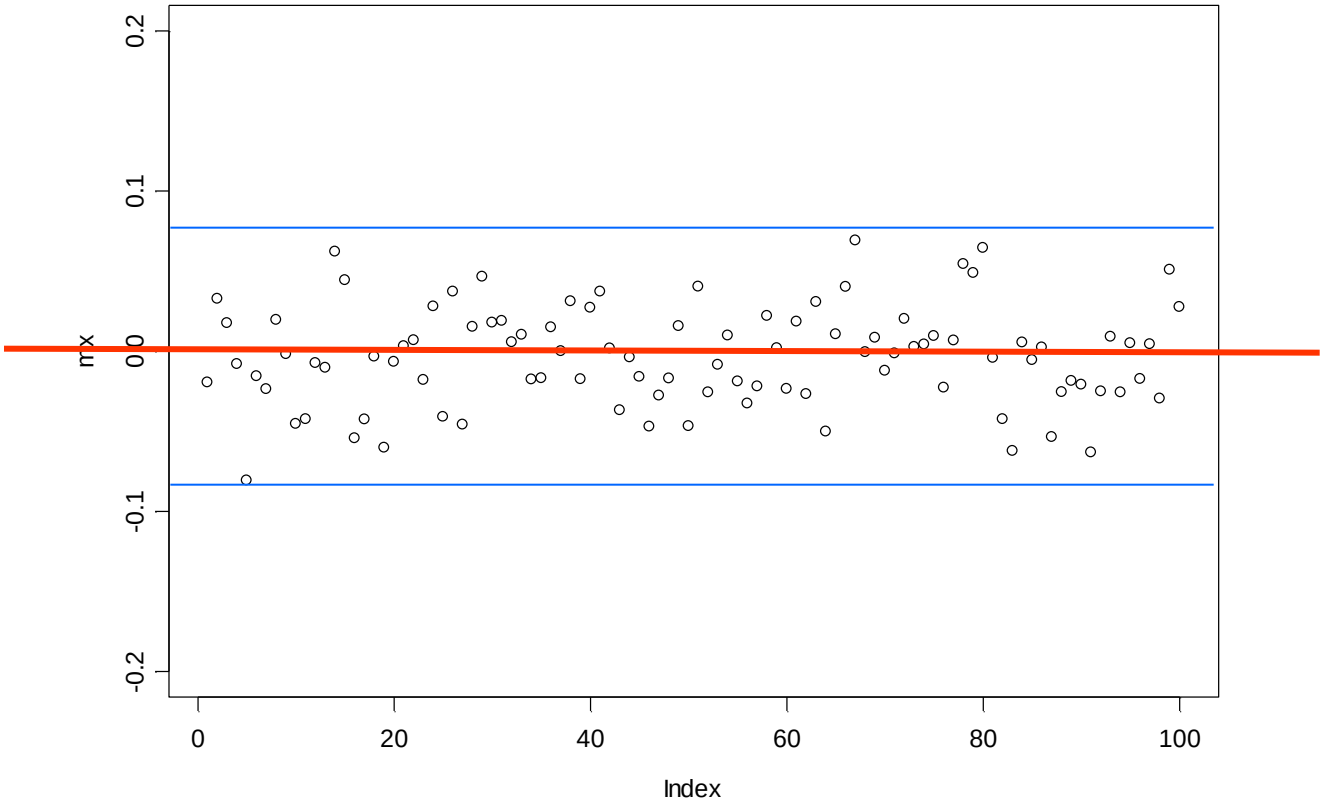
$$\mu = E_F(x)$$

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$$



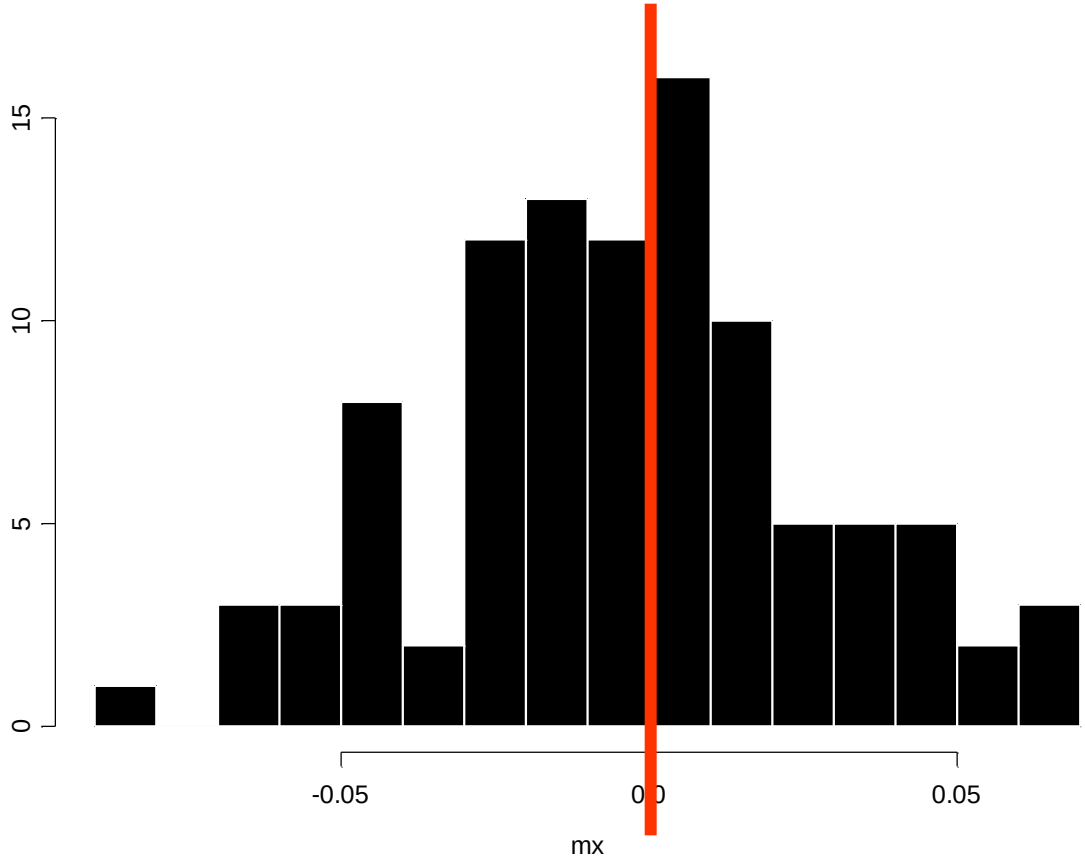
100 sample of size 50 from N(0,1)

$\mu = E_F(x)$



100 samples of size 50 from N(0,1)

$$\mu = E_F(x)$$



The standard error of the sample mean (1)

$$X \sim (\mu_F, \sigma_F{}^2)$$

$$\mu_F = E_F(x)$$

$$\sigma_F^2 = Var_F = E_F[(x - \mu_F)^2]$$

The standard error of the sample mean (2)

population

$$X \sim (\mu_F, \sigma_F{}^2)$$

$$\mu_F = E_F(x)$$

sample

$$F \rightarrow (x_1, x_2, \ldots, x_n)$$

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$E(\bar{x}) = \frac{1}{n} E(\sum_{i=1}^{n} x_i) = \mu_F$$

The standard error of the sample mean (3)

population

$$X \sim (\mu_F, {\sigma_F}^2)$$

$$\sigma_F^2 = E_F[(x - \mu_F)^2]$$

sample

$$F \rightarrow (x_1, x_2, ..., x_n)$$

$$\hat{\sigma}_F = \left\{ \frac{1}{n^2} \sum_{i=1}^{n} (x_i - \bar{x})^2 \right\}^{\frac{1}{2}}$$

The standard error of the sample mean (4)

population

$$X \sim (\mu_F, {\sigma_F}^2)$$

$$\sigma_F^2 = E_F[(x-\mu_F)^2]$$

sample

$$F \rightarrow (x_1, x_2, \ldots, x_n)$$

$$Var(\bar{x}) = \frac{1}{n^2} Var(\sum_{i=1}^{n} x_i) = \frac{{\sigma_F}^2}{n}$$

$$S.E(\bar{x}) = \frac{\sigma_F}{\sqrt{n}}$$

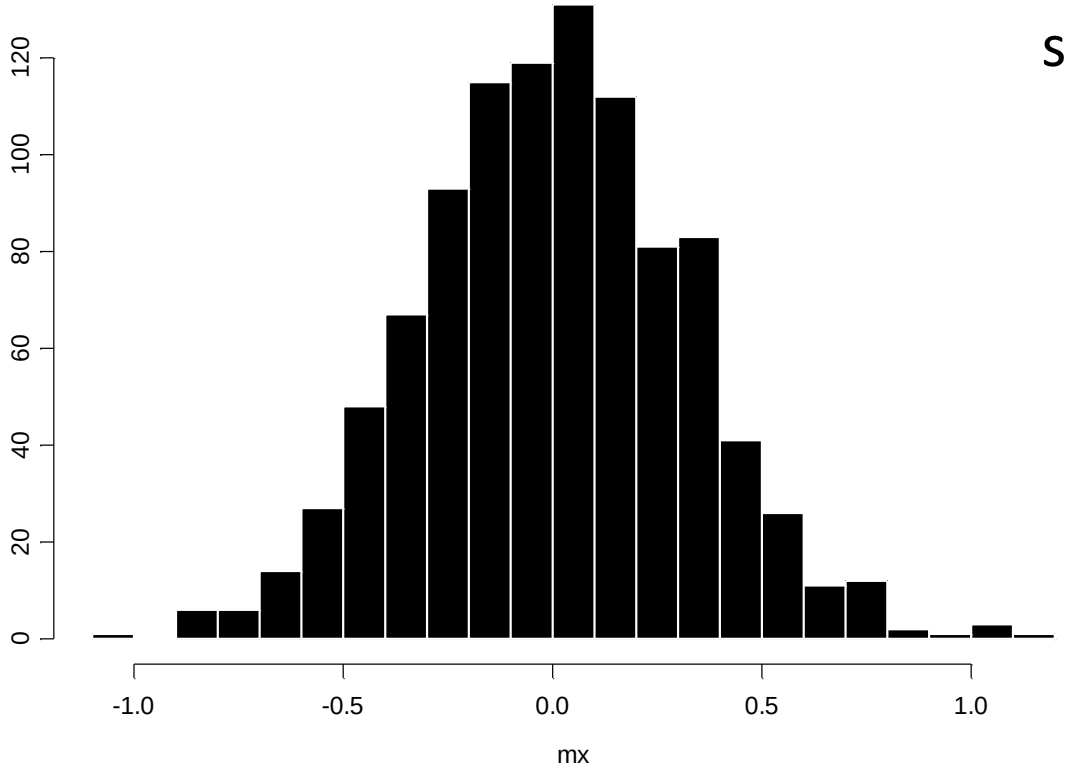
Distribution of the sample mean

$$X \sim (\mu_F, {\sigma_F}^2)$$

$$F \rightarrow (x_1, x_2, \ldots, x_n)$$

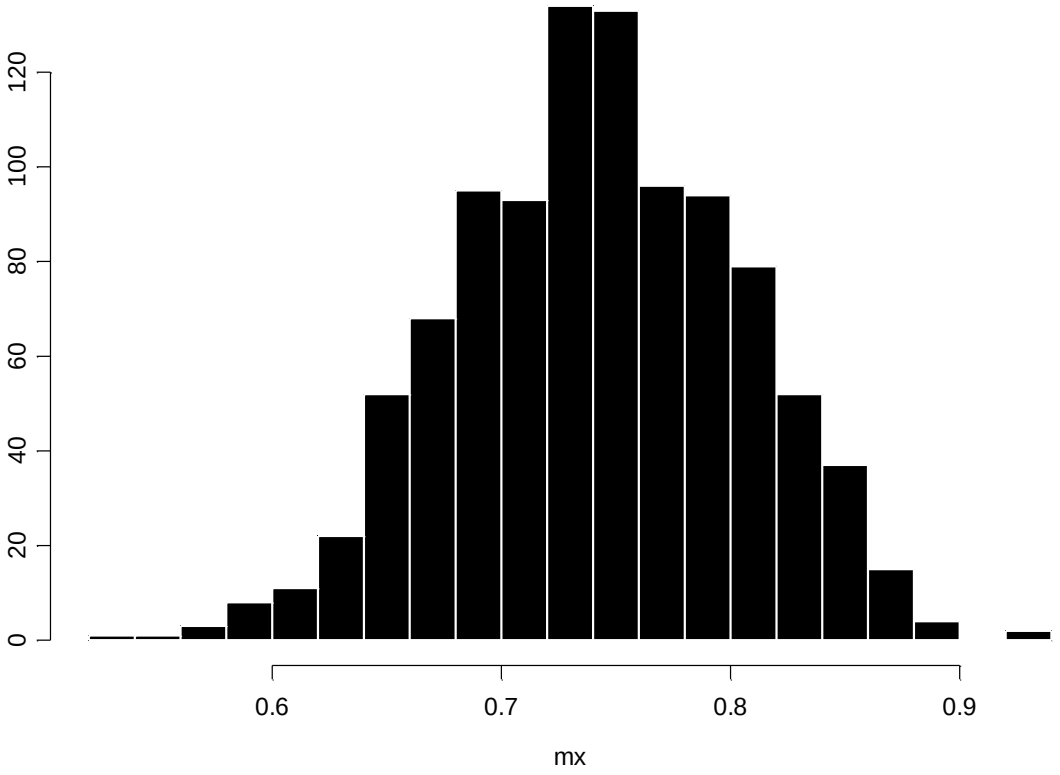
$$\bar{x} \sim N(\mu_F, \frac{\sigma_F^2}{n})$$

Distribution of the sample mean



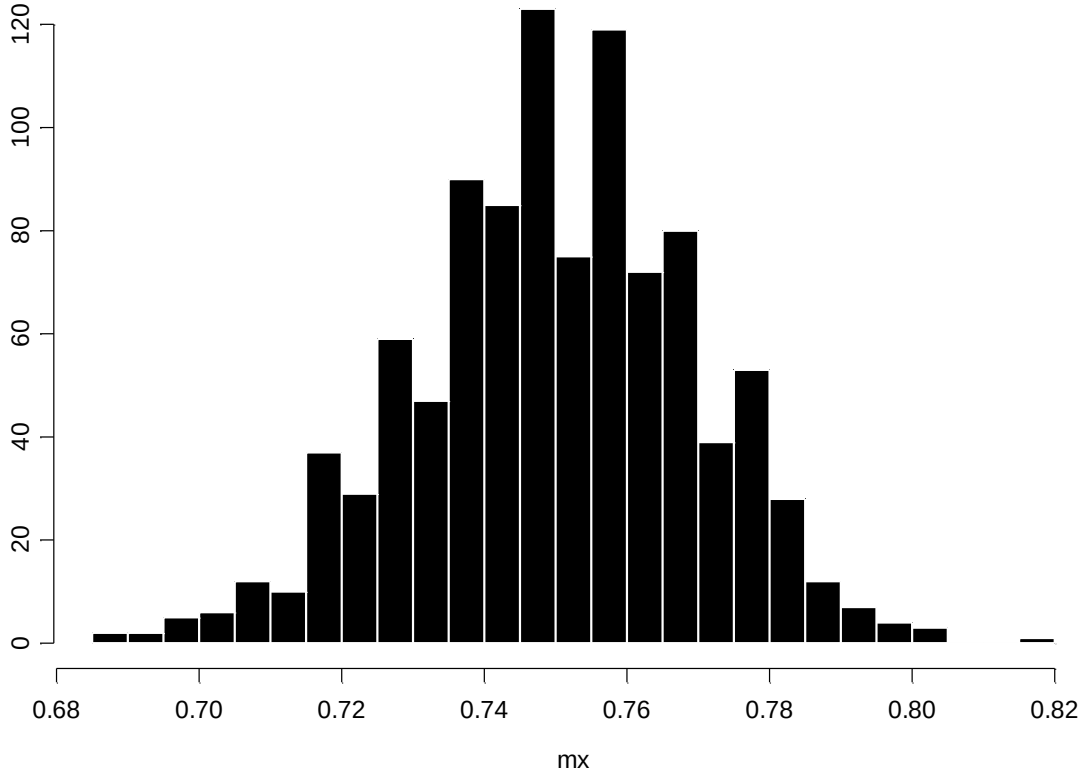
1000 samples of
size 50 from N(0,1)

Distribution of the sample mean



1000 samples of size 50 from B(1,0.75)

Distribution of the sample mean



1000 samples of size 500 from B(1,0.75)

R code: the empirical distribution

```
par(mfrow=c(1,1))
x<-seq(from=-3,to=3,length=1000)
dx<-dnorm(x,0,1)
plot(x,dx,type="l")

x1<-rnorm(10000,0,1)
hist(x1,nclass=50,col=1,probability=T)

par(mfrow=c(1,2))
x2<-rnorm(1000,0,1)
x2<-sort(x2)
px<-pnorm(x2,0,1)
plot(x2,px,type="l")
n<-length(x2)
px.e<-c(1:length(x2))/n
plot(x2,px.e,type="s")
```

R code: the sample mean from N(0,1)

```
par(mfrow=c(1,1))
x2<-rnorm(50,0,1)
x2<-sort(x2)
x<-seq(from=-3,to=3,length=1000)
px<-pnorm(x,0,1)
plot(x,px,type="l")
n<-length(x2)
px.e<-c(1:length(x2))/n
lines(x2,px.e,type="s")

nsim<-1000
mx<-c(1:nsim)
for(i in 1:nsim){
x<-rnorm(10,0,1)
mx[i]<-mean(x)
}
hist(mx,nclass=20,col=1)
```

R code: sample mean from B(1,0.75)

```
nsim<-1000
mx<-c(1:nsim)
for(i in 1:nsim){
x<-rbinom(500,1,0.75)
mx[i]<-mean(x)
}


hist(mx,nclass=20,col=1)
```

The basic bootstrap

The observed data

A sample of 10 observations:

```
> x <- c(11.201, 10.035, 11.118, 9.055, 9.434,
9.663, 10.403, 11.662, 9.285,8.84)
> mean(x)
[1] 10.0696
```

We wish to estimate the standard error of the sample mean

$$S.E(\bar{x}) = \frac{\sigma_F}{\sqrt{n}}$$

Parametric and nonparametric bootstrap

nonparametric bootstrap

$$F \rightarrow (x_1, x_2, \ldots, x_n)$$

We resample from the empirical distribution

parametric bootstrap

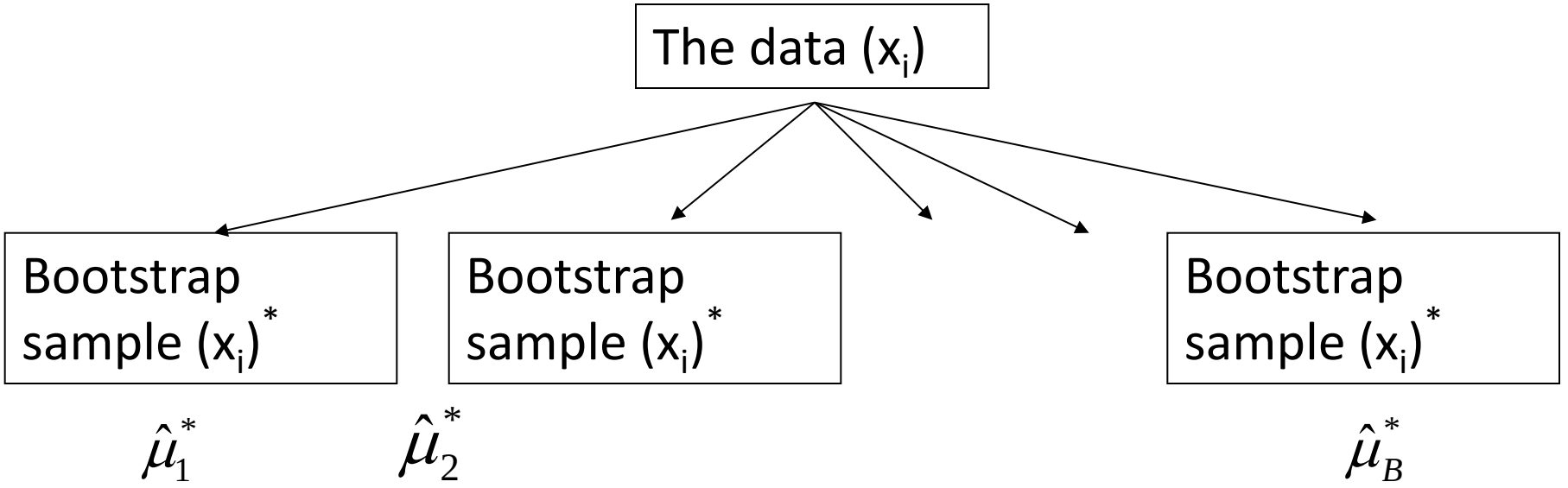
We assume a parametric model for F

$$F(\theta)$$

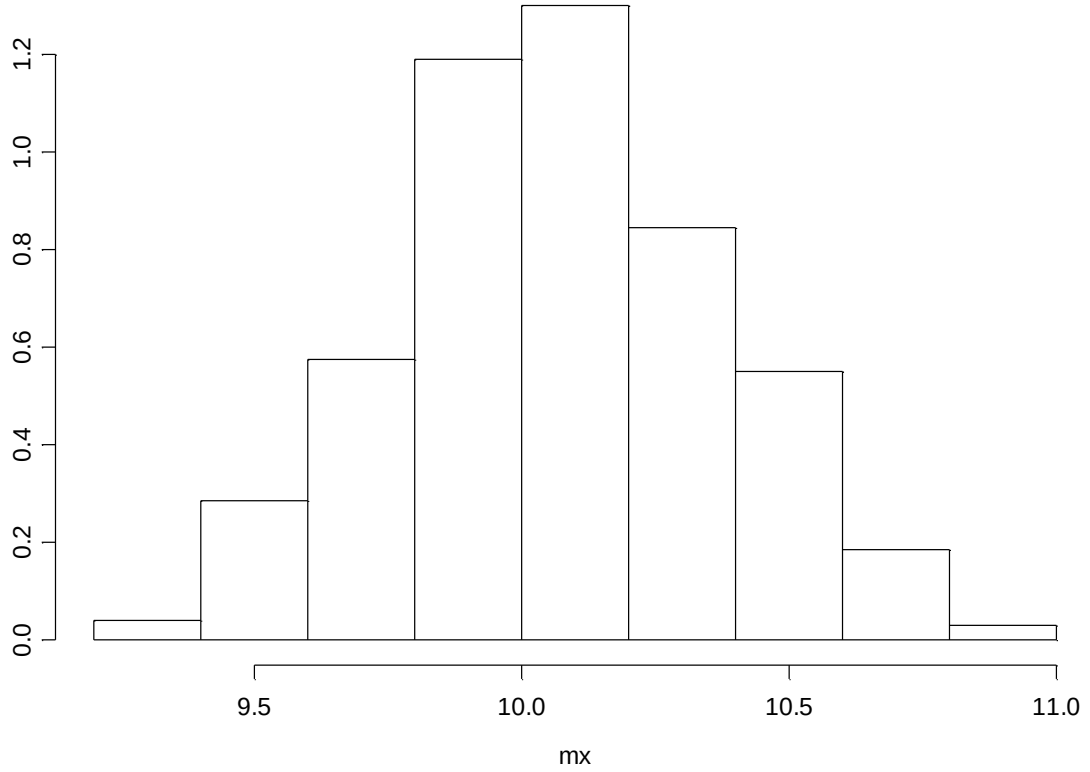
We resample from

$$F(\hat{\theta})$$

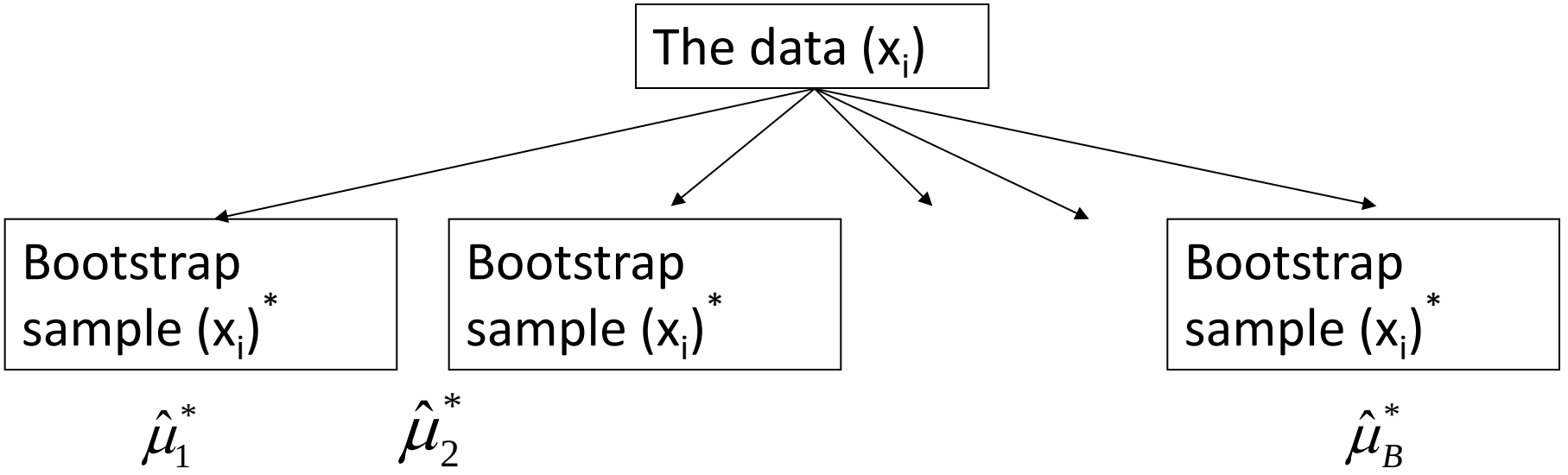
Nonparametric bootstrap



Nonparametric bootstrap



Nonparametric bootstrap



$$S.E.(\hat{\mu}) = \left\{ \frac{1}{B-1} \sum_{b=1}^{B} \left(\hat{\mu}_b^* - \hat{\mu}^* \right)^2 \right\}^{0.5}$$

R code

```
> var(mx)
[1] 0.09357364
```

The estimated standard error 0.093

```
n<-length(x)
B<-1000
mx<-c(1:B)
for(i in 1:B){
cat(i)
boot.i <- sample(x, n,
  replace=T)
mx[i]<-mean(boot.i)
}
```

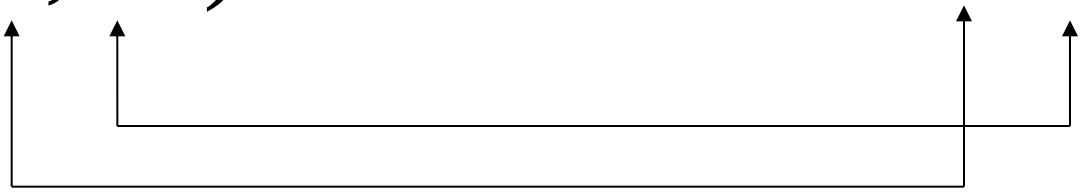
Parametric bootstrap

We assume a parametric model for F

We estimate F by

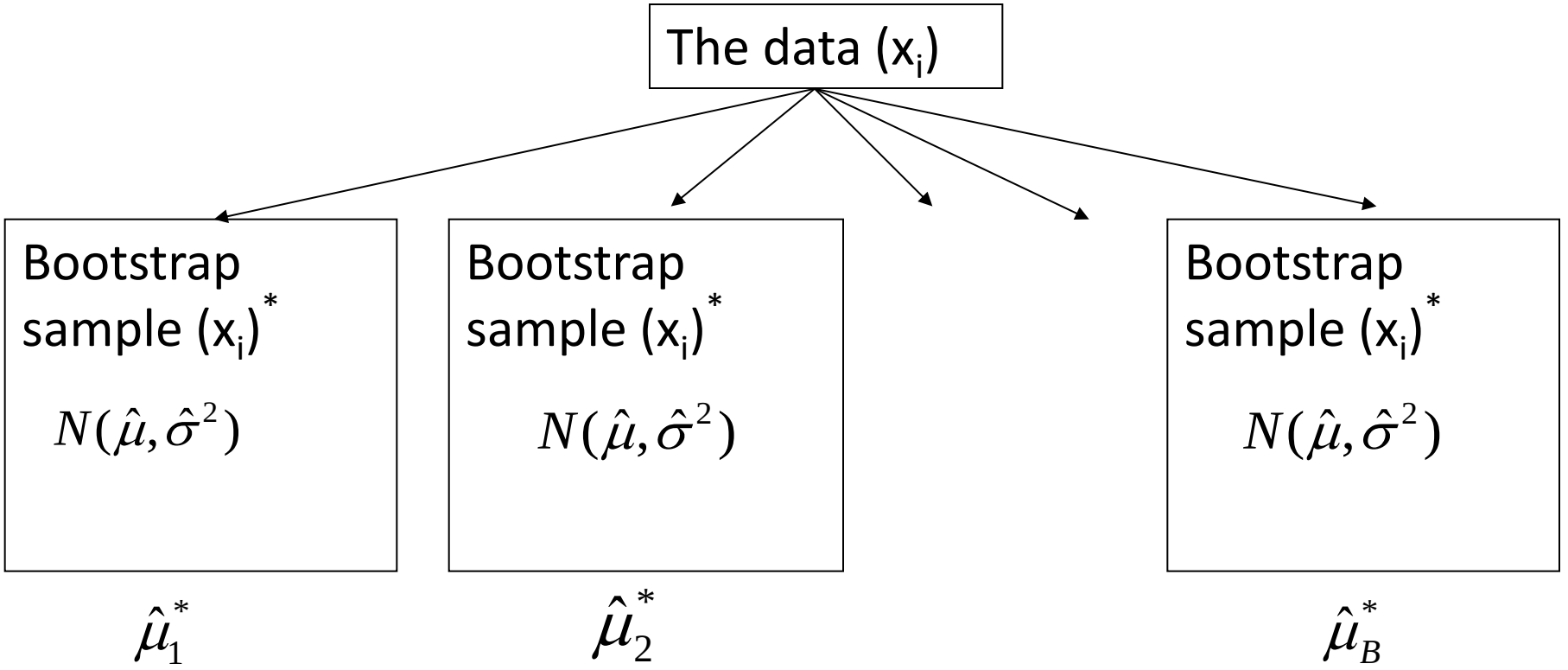
$$F = N(\mu, \sigma^2)$$

$$\hat{F} = N(\hat{\mu}, \hat{\sigma}^2)$$



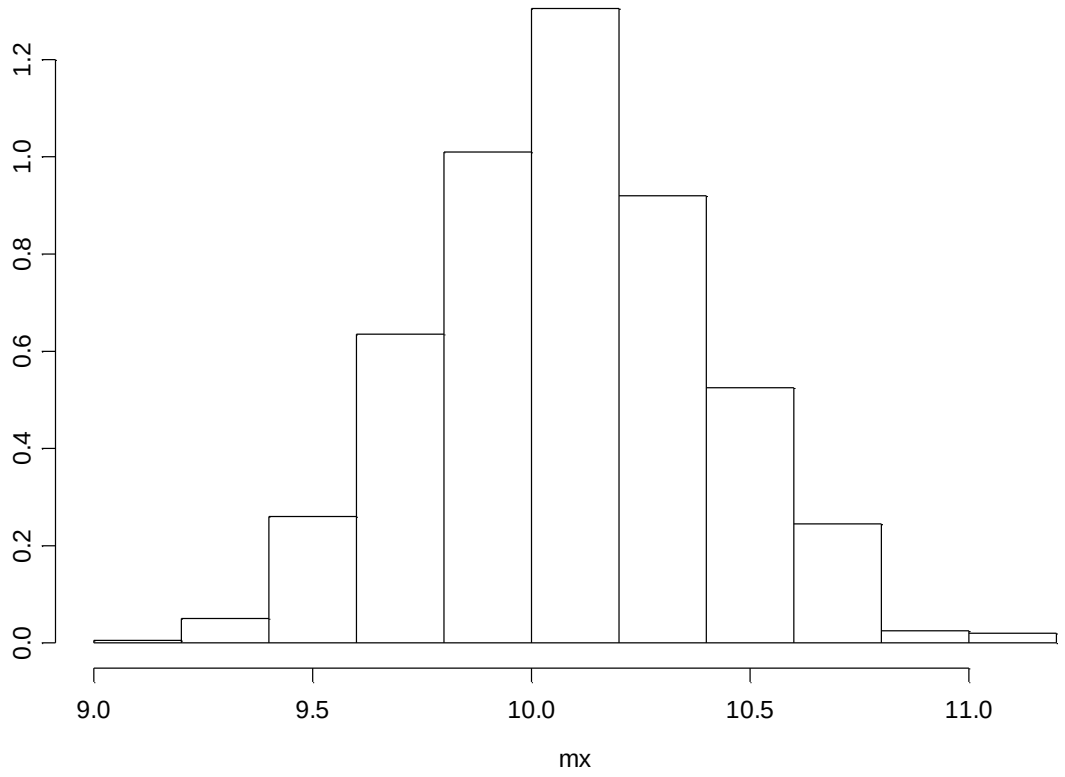
We replace the unknown parameters in F with their plug-in estimates

Parametric bootstrap



$$S.E.(\hat{\mu}) = \left\{ \frac{1}{B+1} \sum_{b=1}^{B} \left(\hat{\mu}_b^* - \hat{\mu}^* \right)^2 \right\}^{0.5}$$

Parametric bootstrap



R code

```
> var(mx)
[1] 0.1007613
```

Bootstrap estimate for the standard error for the mean

```
B<-1000
MLx<-mean(x)
Varx<-var(x)
mx<-c(1:B)
for(i in 1:B){
cat(i)
boot.i<-rnorm(n, MLx,
  sqrt(Varx))
mx[i]<-mean(boot.i)
}
```

The correlation coefficient

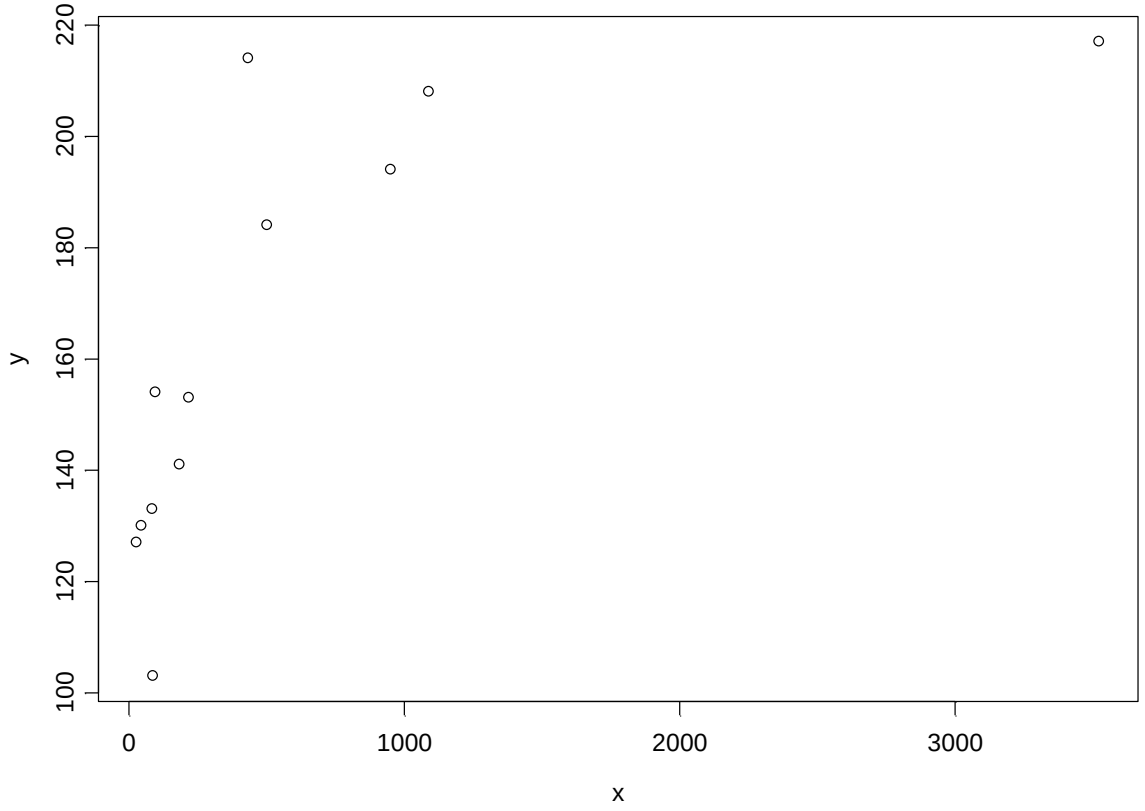
The sample

```
        x   y
 [1,]   29 127
 [2,]  435 214
 [3,]   86 133
 [4,] 1090 208
 [5,]  219 153
 [6,]  503 184
 [7,]   47 130
 [8,] 3524 217
 [9,]  185 141
[10,]   98 154
[11,]  952 194
[12,]   89 103
```

Observed correlation

```
> cor(x, y)
[1] 0.6738982
```

The sample




```
> cor(x,y)
[1]
0.6738982
```

The bootstrap algorithm

The observed sample

x_1	y_1
x_2	y_2
.	.
.	.
.	.
x_{12}	y_{12}

We resample the **pair** (x_i, y_i) with replacement

n=12

x^*_1	y^*_1
x^*_2	y^*_2
.	.
.	.
.	.
x^*_{12}	y^*_{12}

The bootstrap sample

The bootstrap algorithm

The bootstrap sample

$$
\begin{array}{cc}
x^*_1 & y^*_1 \\
x^*_2 & y^*_2 \\
\cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot \\
x^*_{12} & y^*_{12}
\end{array}
\longrightarrow \quad \hat{\rho}^*_b(x^*, y^*)
$$

For each bootstrap sample we calculate the correlation

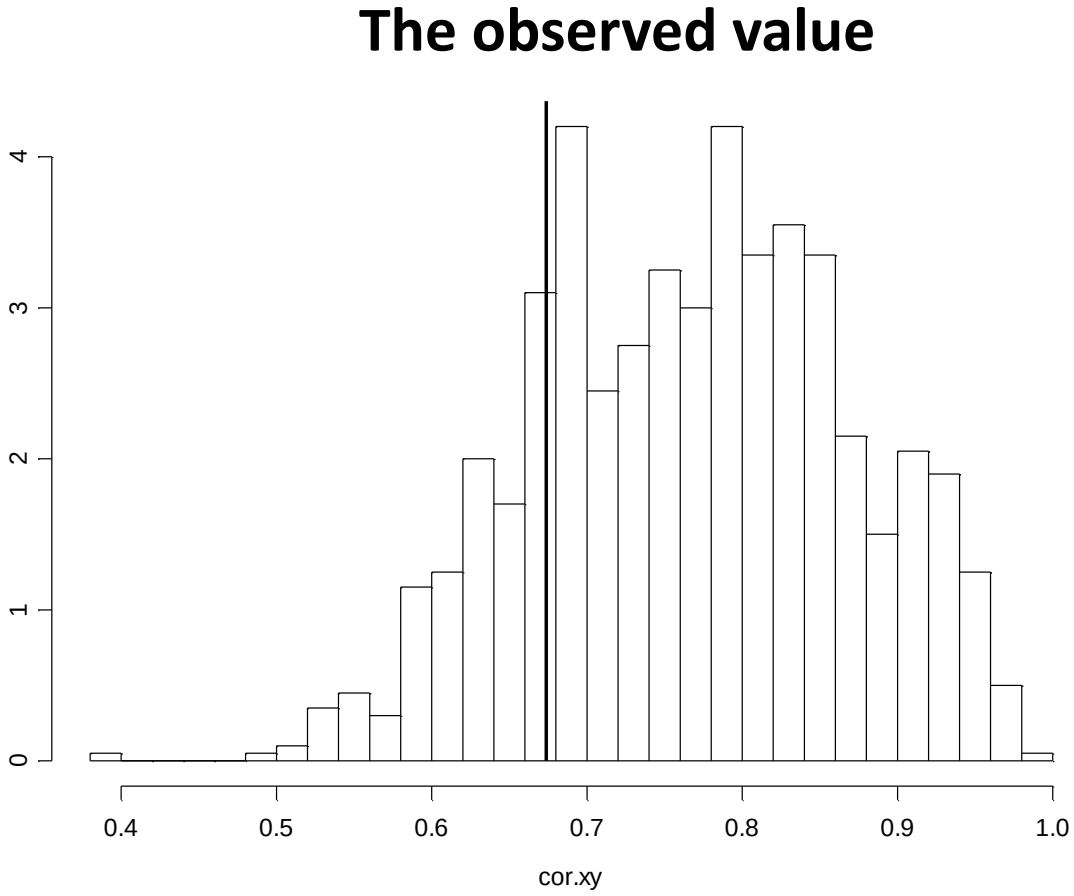
The bootstrap algorithm

B bootstrap replicates

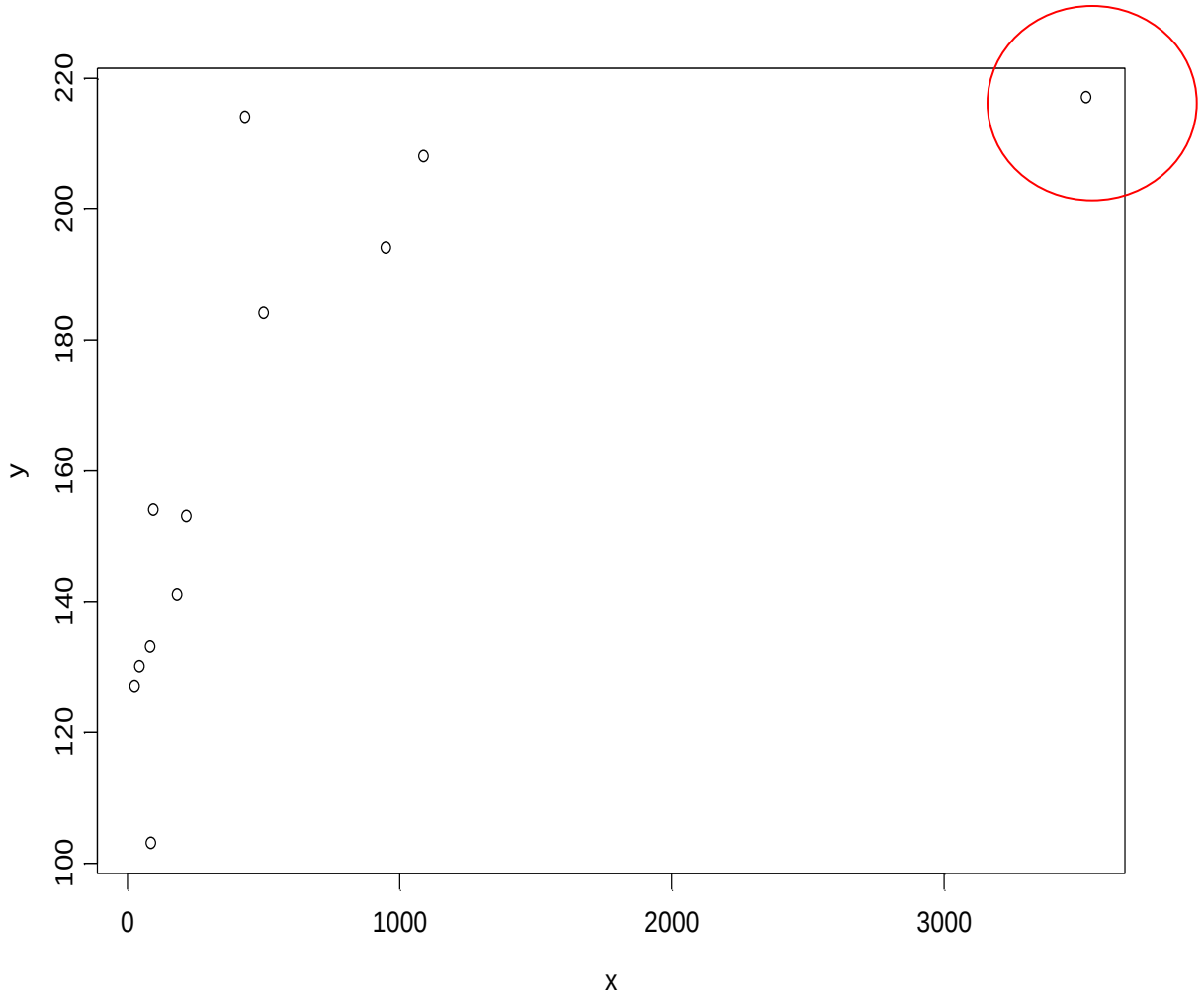
$\hat{\rho}_1^*(x^*, y^*)$ $\hat{\rho}_2^*(x^*, y^*)$ $\hat{\rho}_B^*(x^*, y^*)$

$$S.E(\hat{\rho}) = \left\{ \frac{1}{B-1} \sum_{b=1}^{B} \left(\hat{\rho}_b^* - \hat{\rho}^* \right)^2 \right\}^{\frac{1}{2}}$$

1000 bootstrap replicates for the correlation



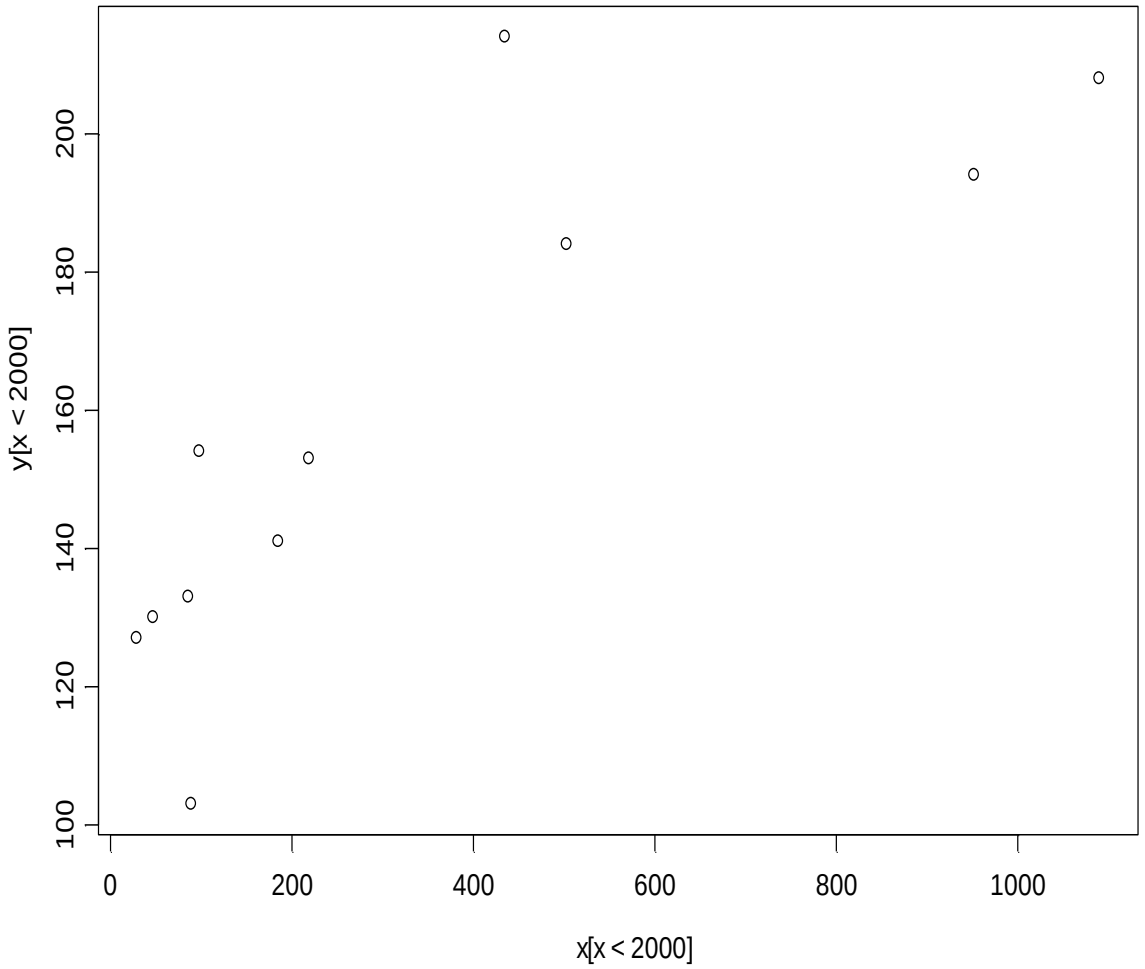
The observed sample



Observation 8

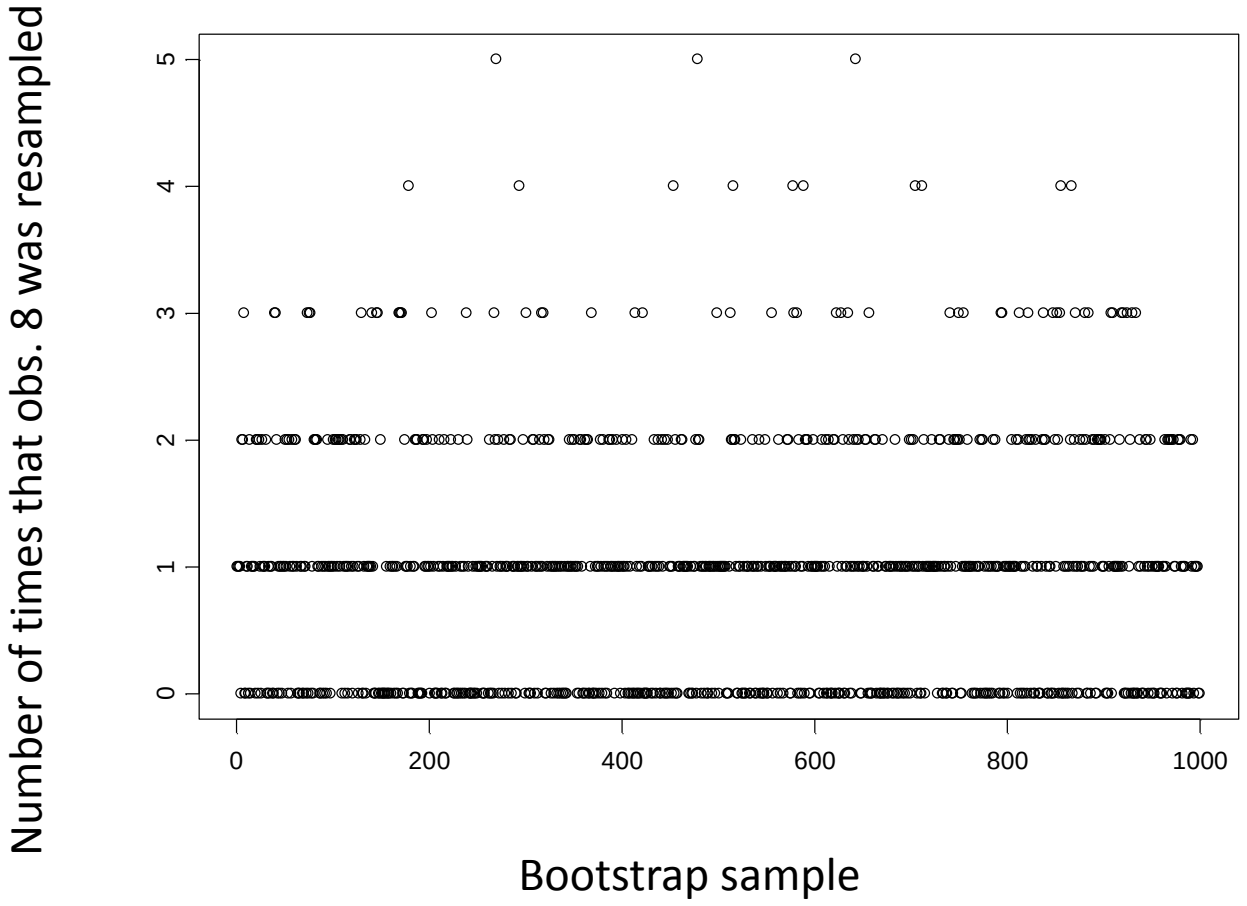
(3524,217)

Data without observation 8

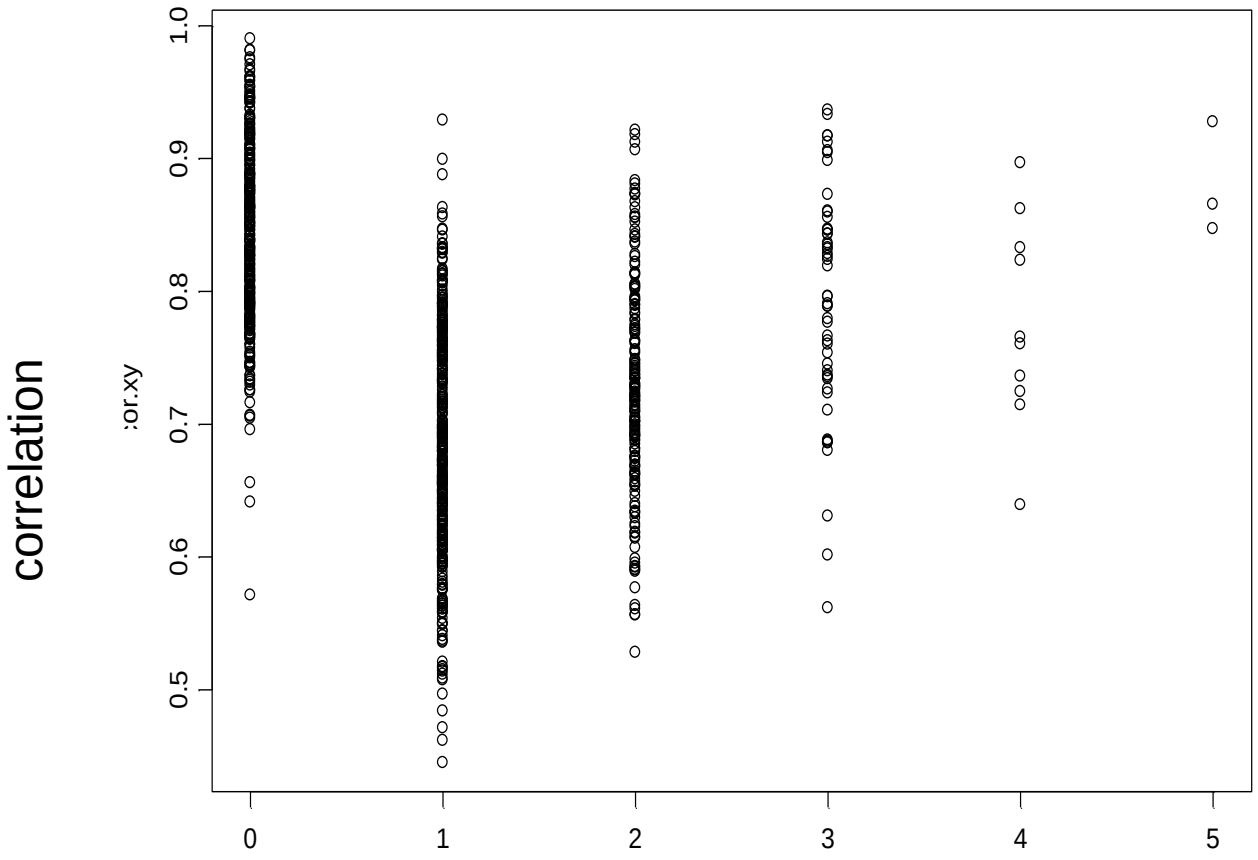


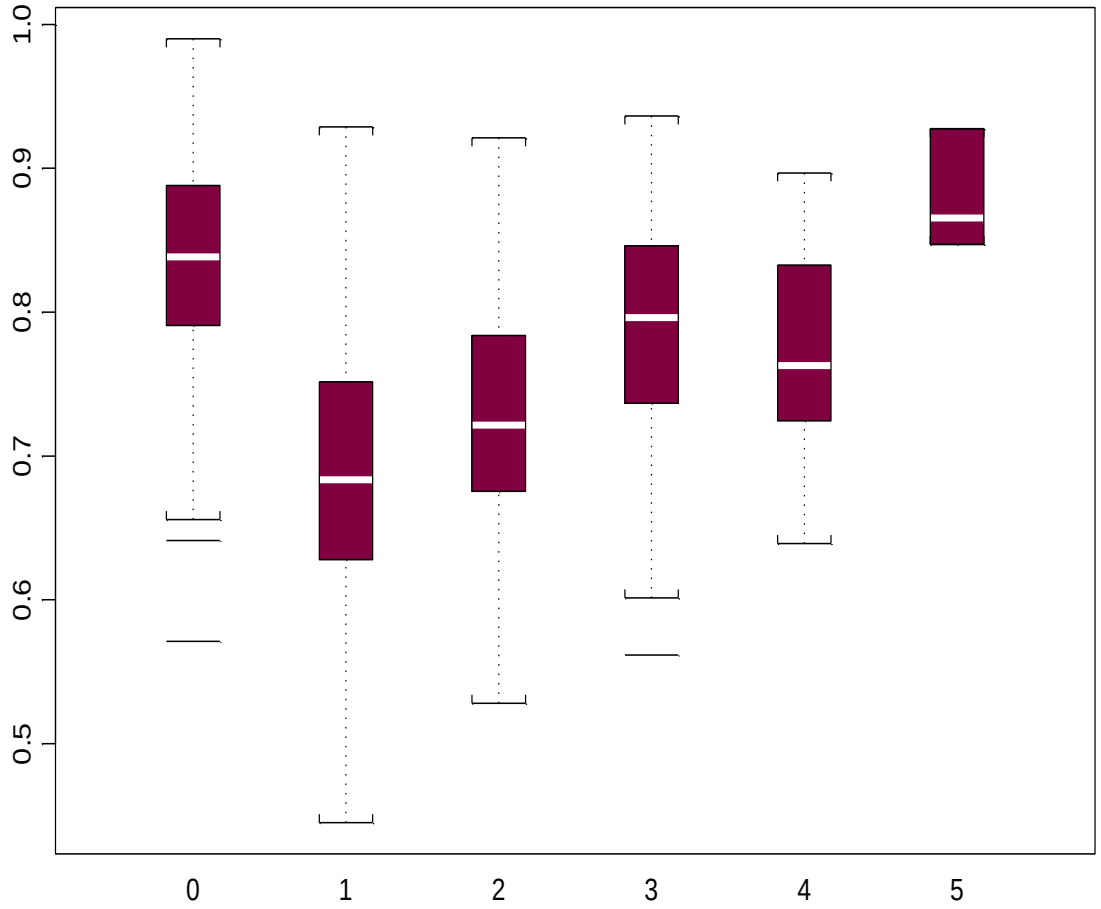

```
> cor(x[x < 2000], y[x < 2000])
[1] 0.820564
```

How many time observation 8 was resample in each bootstrap sample ?



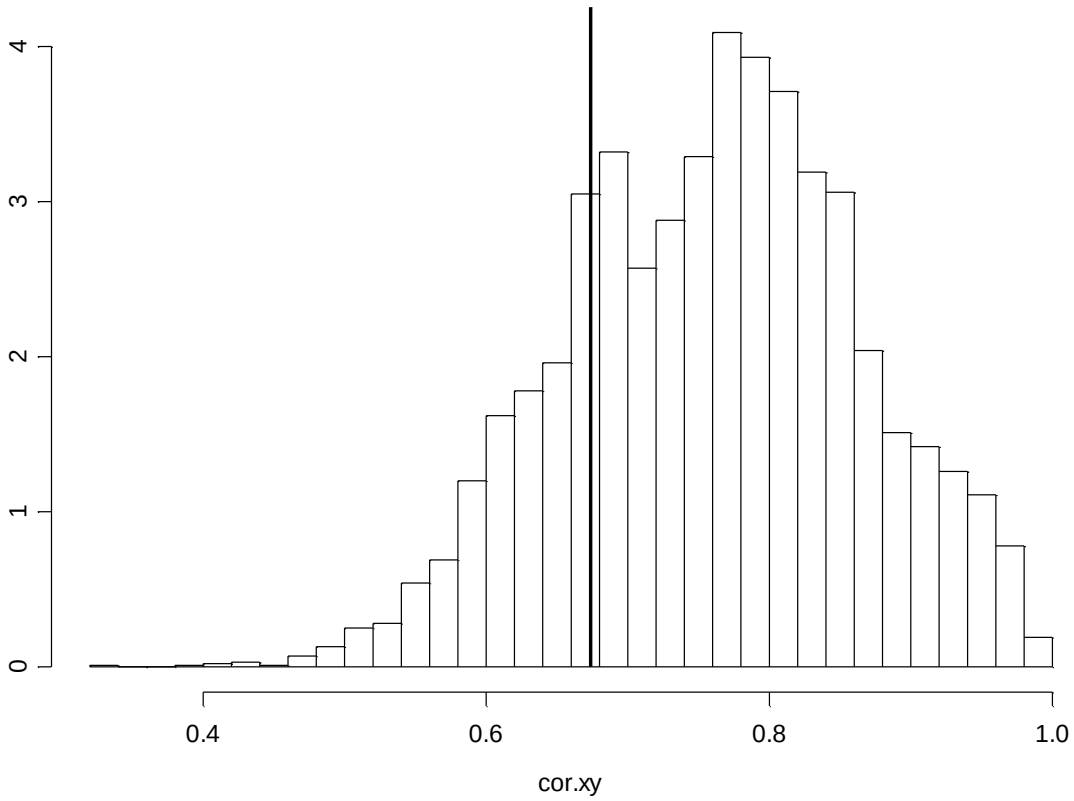
The influence of observation 8



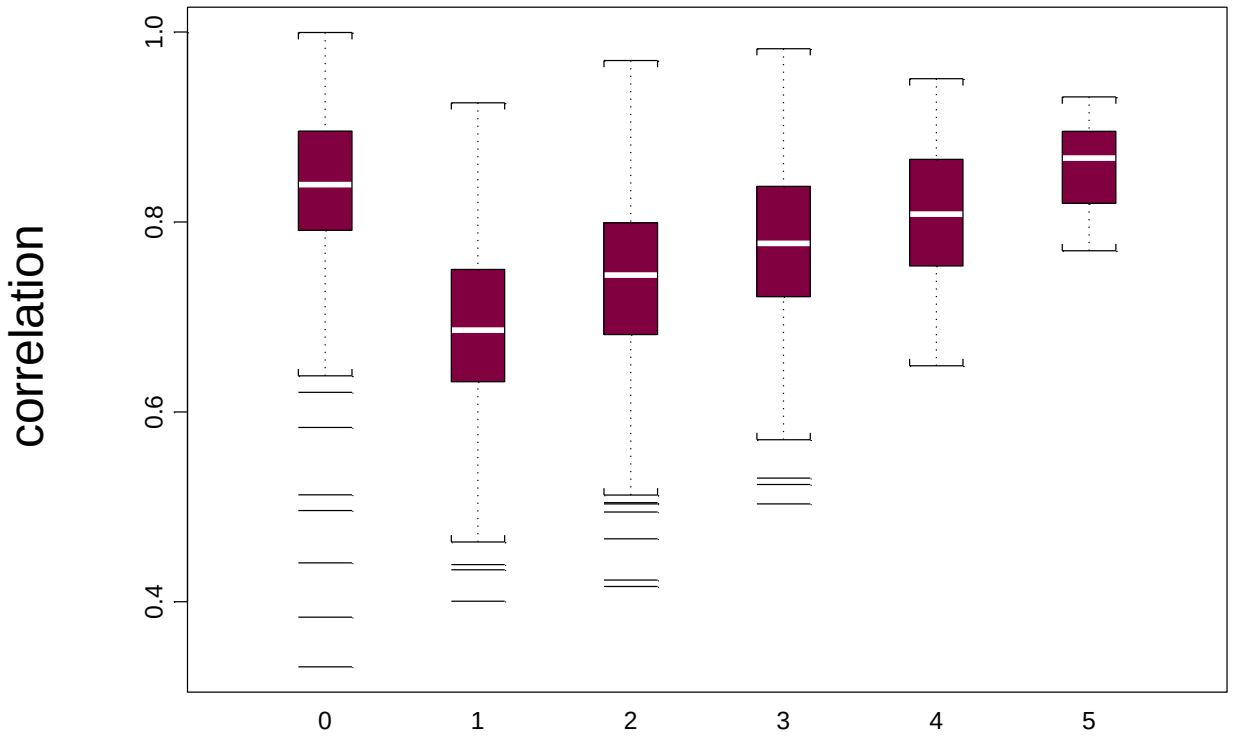


Number of times that obs. 8 was resample

B=5000

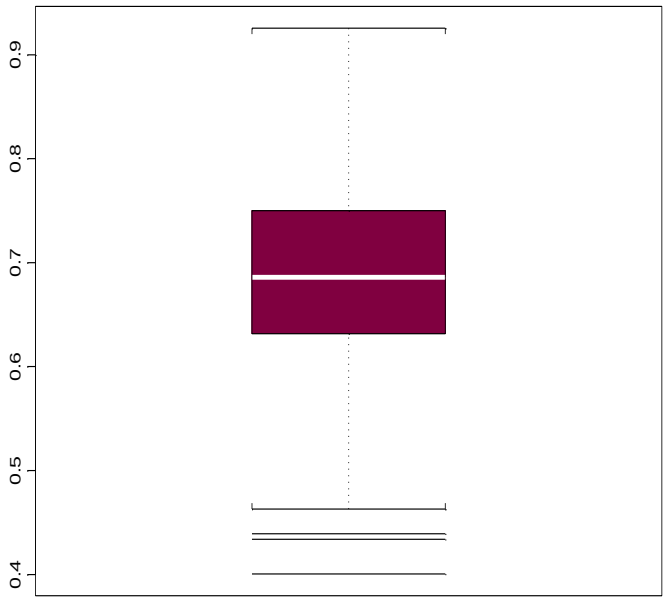


B=5000



Number of times that obs. 8 was resample in each bootstrap sample

The influence of observation 8



Boxplot for the bootstrap replicates (of the correlation) for the samples in which observation 8 was sampled only once

```
> mean(cor.xy[
      obs.8 == 1])
[1] 0.6881557

> cor(x, y)
[1] 0.6737664
```

R code

```
x<-c(29,435,86,1090,219,503,47,3524,185,98,952,89)
y<-c(127,214,133,208,153,184,130,217,141,154,194,103)
cbind(x,y)
plot(x,y)
cor.obs<-cor(x,y)
n<-length(x)
index<-c(1:n)
B<-1000
obs.8<-cor.xy<-c(1:B)
for(i in 1:B){
cat(i)
boot.i<-sample(index,n,replace=T)
obs.8[i]<-sum(boot.i==8)
x.b<-x[boot.i]
y.b<-y[boot.i]
cor.xy[i]<-cor(x.b,y.b)
}

plot(obs.8)
plot(obs.8,cor.xy)
boxplot(split(cor.xy,obs.8))

hist(cor.xy,probability=T,col=0,nclass=30)
lines(c(cor.obs,cor.obs),c(0,10),lwd=3)
```

Estimation of the bias using bootstrap

The probability distribution

Let X be a random variable such that

$$X \sim F(\theta)$$

F is the probability distribution of X.

θ is an unknown parameter to be estimated.

We assume that

$$\theta = t(X_1, X_2, \ldots, X_N)$$

The empirical distribution

The empirical distribution function is defined to be the discrete distribution that puts probability of 1/n on each value of x_i

$$F \rightarrow \left(x_1, x_2, \ldots, x_n\right)$$

$$P(A) = \hat{F} = \frac{\#(x_i \in A)}{n}$$

The plug-in principle

The plug-in estimate of the parameter

is defined as

$$\theta = t(F)$$

$$\hat{\theta} = t(\hat{F})$$

We use the same function from F, *t(F)* on the empirical distribution

Bias

Bias: on average, how far is the statistic from the parameter ?

$$bias_F = bias_F(\hat{\theta}, \theta) = E_F(\hat{\theta}) - \theta$$

observed

unknown parameter

The bootstrap estimate for the bias

We would like to apply the bootstrap method in order to estimate the bias of a statistics.

This could be very useful if the distribution of the statistics in unknown.

$$bias_{\hat{F}} = bias_{\hat{F}}(\hat{\theta}, \hat{\theta}^*) = E_{\hat{F}}(\hat{\theta}^*) - \hat{\theta}$$

Observed statistics

The bootstrap algorithm

The observed data

$$x_1, x_2, \ldots\ldots, x_n$$

B bootstrap samples

$$x_1^*, x_2^*, \ldots\ldots, x_n^* \qquad x_1^*, x_2^*, \ldots\ldots, x_n^* \qquad x_1^*, x_2^*, \ldots\ldots, x_n^*$$

The bootstrap replicates

$$\theta_1^* \qquad \theta_b^* \qquad \theta_B^*$$

The bootstrap estimate for the bias

We first approximate the distribution of the statistics using bootstrap.

We estimate the expected value of the bootstrap replicates by

$$\hat{\theta}^* = \frac{\sum_{b=1}^{B} \hat{\theta}_b^*}{B} = \hat{E}_{\hat{F}}(\hat{\theta}^*)$$

The bootstrap estimate for the bias

The estimate for the bias

$$\hat{b}_{\hat{F}} = \hat{\theta}^* - \hat{\theta}$$

The mean of the bootstrap replicates

The observed statistics

Example: the patch data

- Eight subjects used medical patches design to decrease the level of a certain hormone in the blood.
- Each subject was measured three times, at baseline (using a placebo patch), using old patch and using new patch.

Example: the patch data

Bioequivalence study.

The FDA criterion for bioequivalence is that the expected value of the new patch match the expected value of the new patches so that

$$\frac{\left|E(new\quad patch)-E(old\quad patch)\right|}{E(old\quad patch)-E(placebo\quad patch)}\leq 0.2$$

The test statistic

We define 2 variables

$$z = oldpatch - placebo$$
$$y = newpatch - oldpatch$$

Ratio statistic

$$\theta = \frac{E_F(y)}{E_F(z)}$$

F is the joint distribution of y and z

What is the distribution of θ ?

The plug in estimate

parameter

$$\theta = \frac{E_F(y)}{E_F(z)}$$

parameter estimate

$$\hat{\theta} = \frac{\bar{y}}{\bar{z}} = \frac{1/8\sum_{i=1}^{8} y_i}{1/8\sum_{i=1}^{8} z_i}$$

The bootstrap algorithm

The observed data

$$x_i = (z_i, y_i)$$

$$x_1, x_2, \ldots\ldots\ldots, x_n$$

Since the distribution of the statistic is unknown we use bootstrap to approximate the distribution.

We resample pairs.

B bootstrap samples

$$x_1^*, x_2^*, \ldots\ldots\ldots, x_n^* \quad \theta_1^*$$

$$x_1^*, x_2^*, \ldots\ldots\ldots, x_n^* \quad \theta_b^* \longrightarrow \hat{\theta}_b^* = \frac{\bar{y}^*}{\bar{z}^*} = \frac{1/8\sum_{i=1}^{8} y_i^*}{1/8\sum_{i=1}^{8} z_i^*}$$

$$x_1^*, x_2^*, \ldots\ldots\ldots, x_n^* \quad \theta_B^*$$

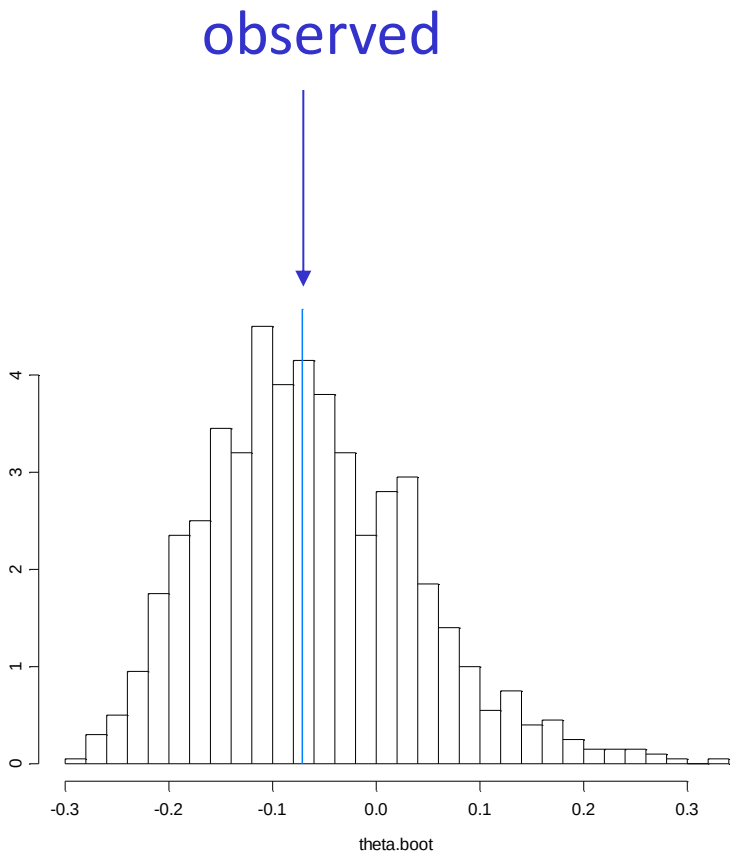
Data and observed statistic

```
 V1    V2    V3
1  9243 17649 16449
2  9671 12013 14614
3 11792 19979 17274
4 13357 21816 23798
5  9055 13850 12560
6  6290  9806 10157
7 12412 17208 16570
8 18806 29044 26325
```

```
> mean(y)
[1] -452.25
> mean(z)
[1] 6342.375
> theta.obs <- mean(y)/mean(z)
> theta.obs
[1] -0.0713061
```

The observed ratio is -0.0713

The bootstrap replicates

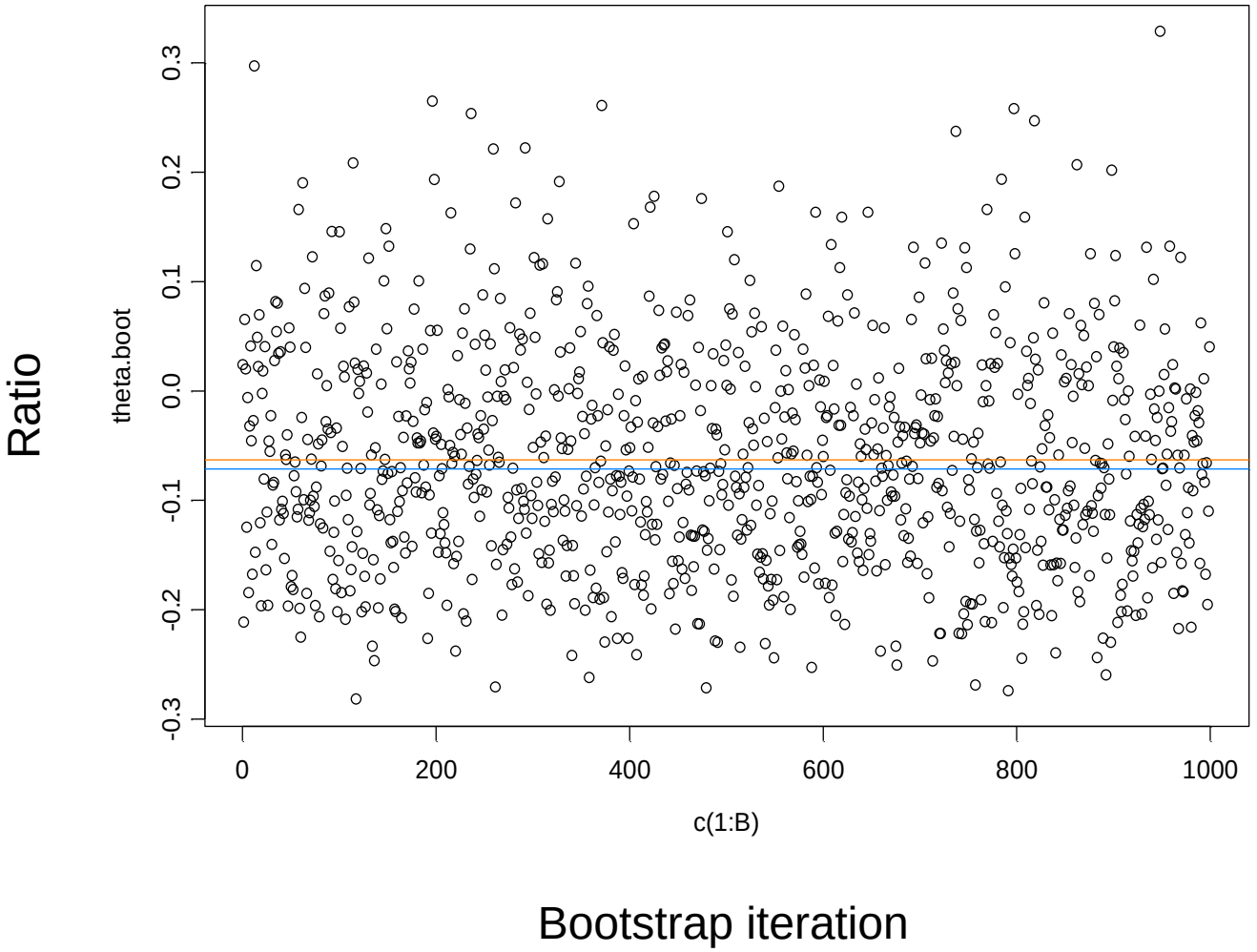


1000 bootstrap replicates.

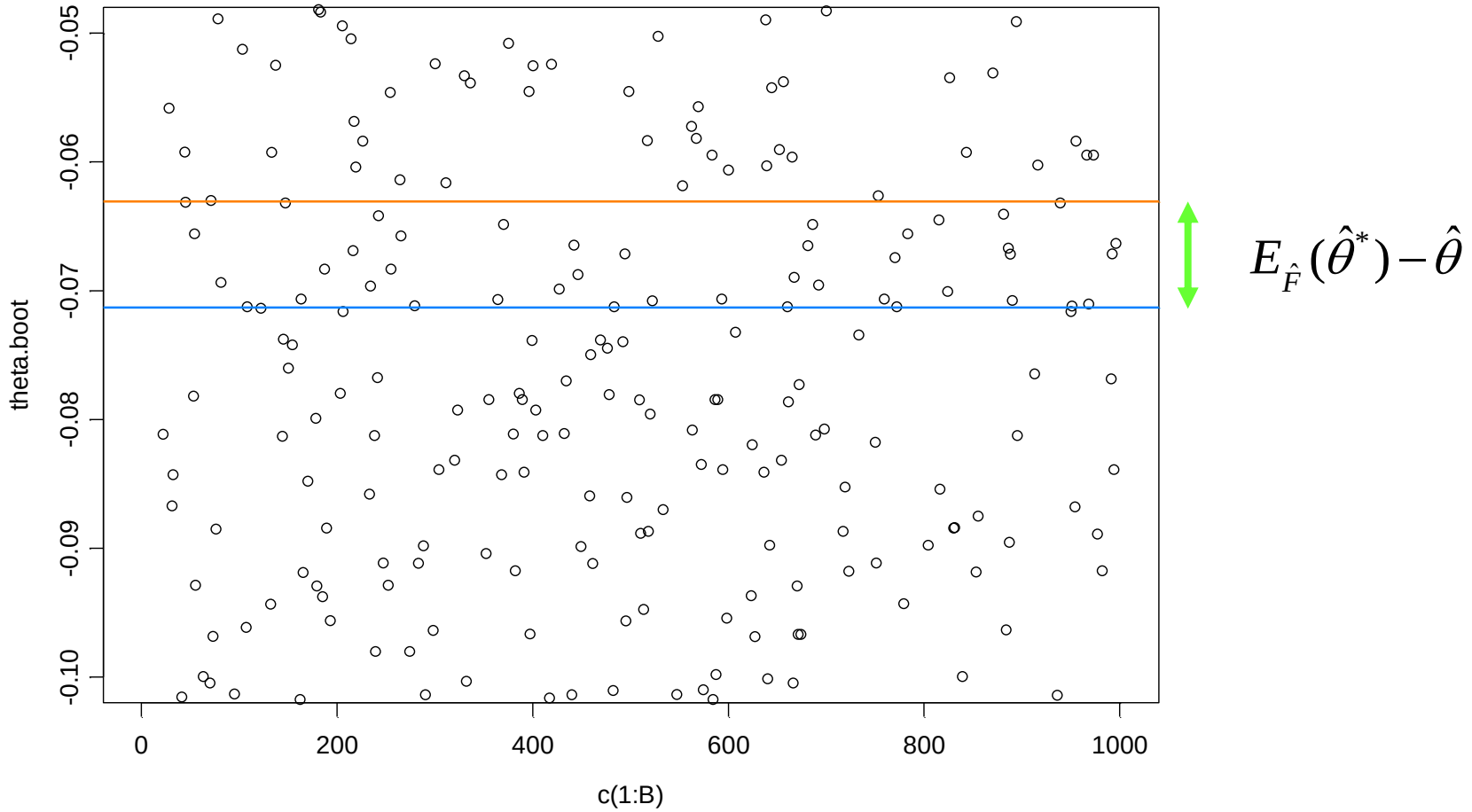
Asymmetric distribution for the ratio.

```
n<-length(z)
B<-1000
index<-c(1:n)
theta.boot<-c(1:B)
for(i in 1:B){
cat(i)
i.boot<-sample(index, size=n,
   replace=T)
y.boot<-y[i.boot]
z.boot<-z[i.boot]
theta.boot[i]<-
   mean(y.boot)/mean(z.boot)
}

hist(theta.boot,col=0,nclass=30,prob
   ability=T)
lines(c(theta.obs,theta.obs),c(0,5),
   lwd=2,col=6)
```

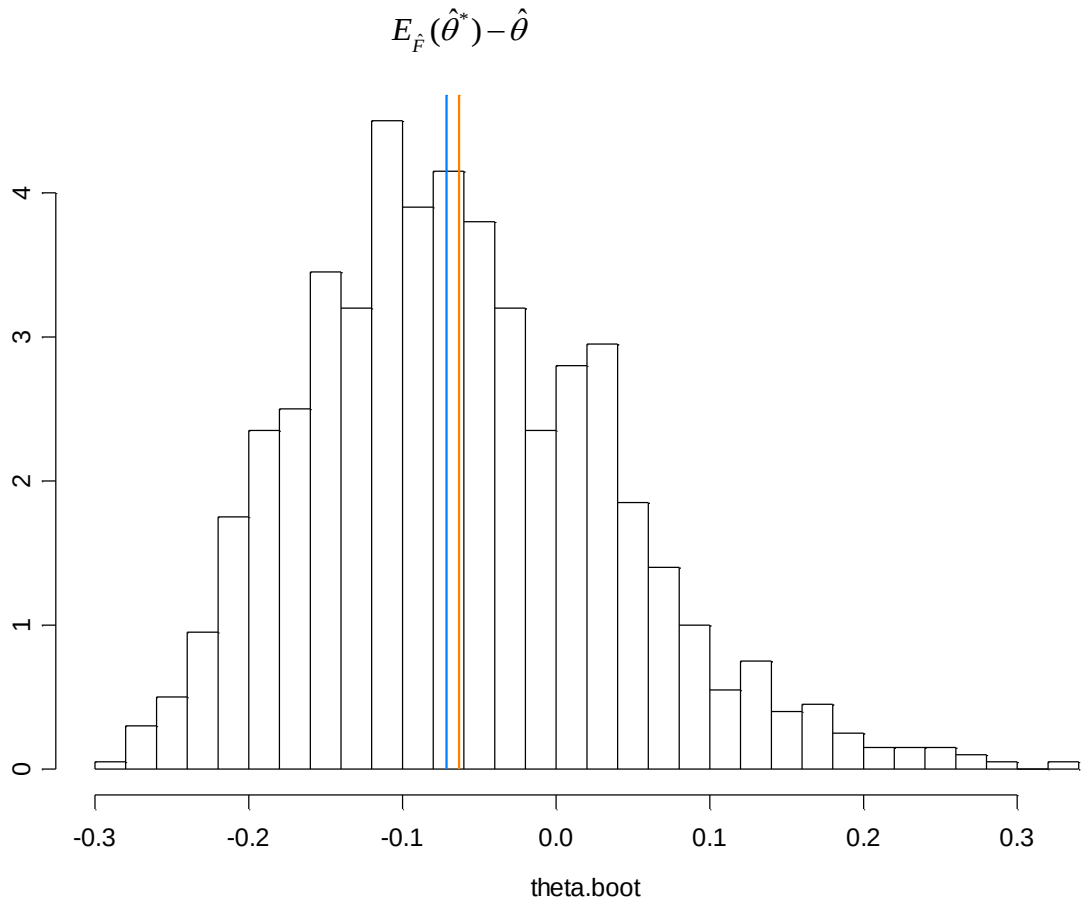
The bootstrap replicates and the bias



The bootstrap replicates and bias



The bootstrap replicates and bias



Estimate for the bias

$$\hat{b}_{\hat{F}} = \hat{\theta}^* - \hat{\theta}$$

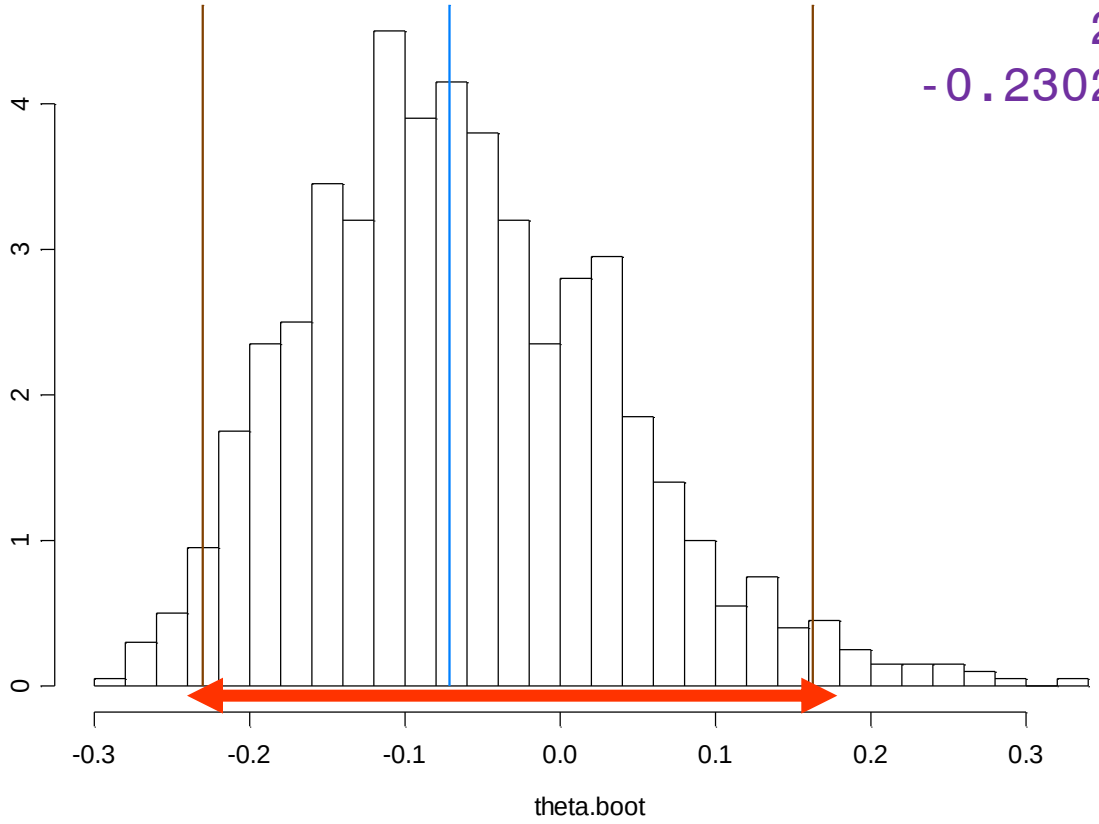
```
> m.boot <- mean(theta.boot)
> b.boot <- m.boot - theta.obs
> b.boot
[1] 0.008231291
```

```
b.boot <- m.boot - theta.obs
```

The mean of the bootstrap replicates

The observed statistic

Bioequivalence ?




```
> quantile(theta.boot,
probs = c(0.025, 0.975))
      2.5%      97.5%
-0.2302189 0.1626573
```

$$|\theta| \leq 0.2$$

PART 3:
Statistical Computing: likelihood functions and maximum likelihood estimators

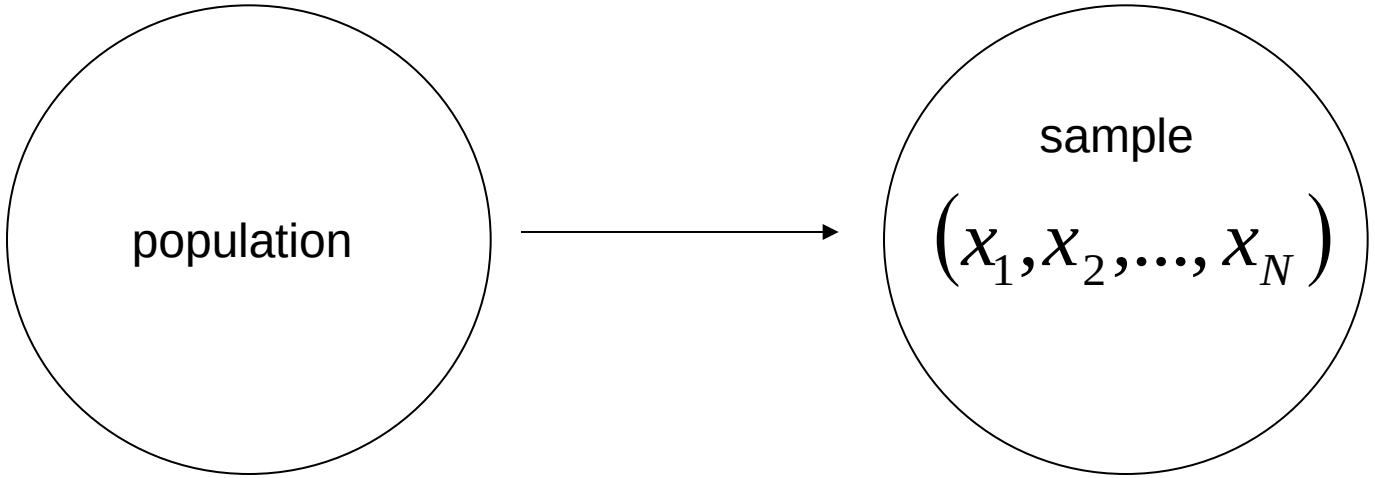
Outline

- Chapter 1:
- Chapter 2:
- Chapter 3:

Introduction

- We observed a random sample from the probability distribution F.

$$F \rightarrow (x_1, x_2, \ldots, x_N)$$



- Parametric representation of F
- We know F except some parameter(s): $F = F(\theta)$

Introduction

- Based on observed data set $x_1, \ldots, x_N$, we would like to draw conclusion about parameter θ.

- **Likelihood function** theory allows us to obtain such estimate.

- Point estimator that arises from likelihood function is called **Maximum likelihood estimator** (MLE).

Likelihood function: Definition

- Likelihood function is a function of the parameter of the statistical model, given fixed outcome (observed sample).
- It indicates how likely is a particular parameter to produce an observed sample.

- Formally: the *likelihood* of a set of parameter values given some observed outcomes is equal to the *probability* of those observed outcomes under various possible values of θ.

Likelihood vs. Probability

- Probability is a function of the outcome given a fixed parameter value.

- *Example:*
- Probability*:*

  "Given that I have flipped a coin 100 times and it is a fair coin, what is the *probability* of it landing heads-up every time? “
- Likelihood:

  "Given that I have flipped a coin 100 times and it has landed heads-up 100 times, what is the *likelihood* of the coin being fair?"

Likelihood Function for Discrete Distribution

Discrete Case:

- Let *X* be a random variable with a discrete probability distribution *P* depending on a parameter θ.

  Then the function

$$L(\theta \mid x) = P_\theta(x) = P_\theta(X = x)$$

  considered as a function of θ, is called the likelihood function (of θ, given the outcome *x* of *X*).

Likelihood Function for Continuous Distribution

Continuous case:

- Let X be a random variable with a continuous probability distribution with density function f depending on a parameter θ. Then the function

$$L(\theta \mid x) = f_\theta(x)$$

considered as a function of θ, is called the likelihood function.

Bernoulli distribution

- Binary outcome 0/1.
- Know also as Alternative distribution.
- Bernoulli probability function is given as

$$P(x; p) = p^x \cdot (1-p)^{1-x} \quad \text{for} \quad \text{x} = 0,1$$

- Hence $P(x = 1; p) = p$

  $P(x = 0; p) = (1-p)$

- If we denote success as 1, then Bernoulli distribution represents experiment of one trial with probability of success given by parameter *p*.

Bernoulli distribution

- Likelihood function for one observation:

$$L(p;x) = p^x \cdot (1-p)^{1-x}\ ,\ p \in [0,1]$$

- Knowing that x=0,1, *L* is still function of *p*:

$$L(p;x=0) = (1-p)$$
$$L(p;x=1) = p$$

- Note: likelihood function is the same function as probability function, only evaluated with respect to different parameter.

Bernoulli distribution

- We have data set that consists of number of observations $x_{1,\ldots,}x_N$

- Assume:

  $x_{1,\ldots,}x_N$ are independent

  $x_{1,\ldots,}x_N$ are identically distributed: Bernoulli(*p*)

- Each x_i represents one observation of situation when experiment with 1 trial with probability of success *p* resulted into successes (x_i=1) or not (x_i=0)
- Identically distributed: there is identical parameter *p*
- Independent: probability of x_1=1 and x_2=1 simultaneously is simply multiplication of their own probabilities of success

$$p \cdot p$$

Bernoulli distribution

- Likelihood function for all the observations

$$L(p; x_1, \ldots, x_N) = \prod_{i=1}^{N} p^{x_i} (1-p)^{1-x_i}$$

$$= p^{\sum_{i=1}^{N} x_i} (1-p)^{\sum_{i=1}^{N} (1-x_i)}$$

$$= p^{\sum_{i=1}^{N} x_i} (1-p)^{N - \sum_{i=1}^{N} x_i}$$

- This function involves the parameter (p) given the data ($x_{1,\ldots,}x_N$).
- Likelihood function depends only on the sum of $x_{1,\ldots,}x_N$.

Bernoulli distribution

- We are estimating how likely various values of *p* are given the observed data.

- Example:

  We have *N*=10 independent trials.

  So, the possible value of sum of $x_{1,\ldots,}x_N$ would be: 0,1, 2, …, 10.

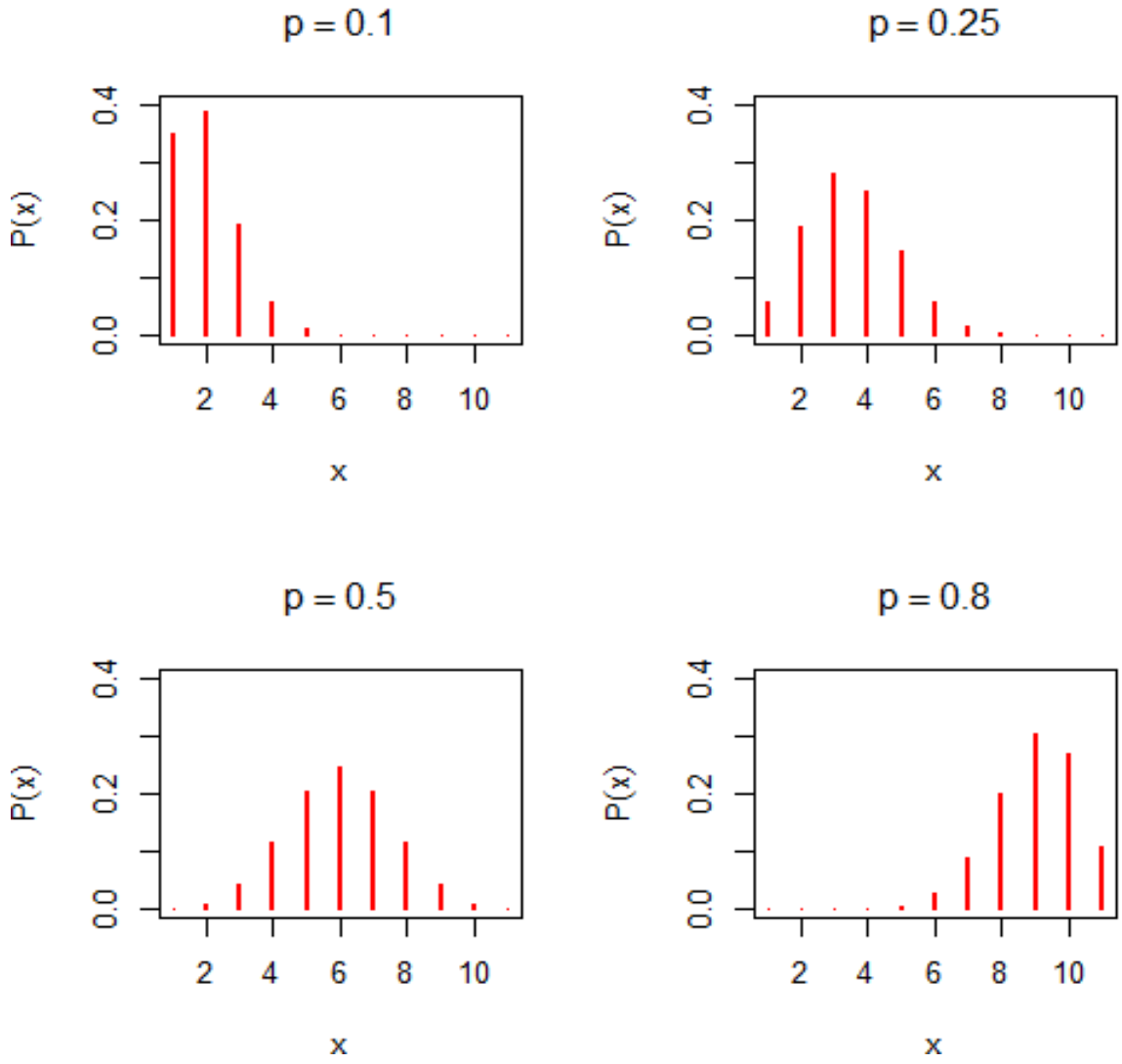
  Let us see the probabilities of different values of *x* for different *p* value.

- For each value of *p*, we can construct PDF (probability distribution function), how likely are different values sum of $x_{1,\ldots,}x_N$, given *p*.

  [In this case, it actually corresponds to PDF of binomial distribution with probability of success equal to *p, N* number of trials and sum of $x_{1,\ldots,}x_N$ as number of successes.]

Bernoulli distribution

Line graph of PDF for N=10



Bernoulli distribution

```
> par(mfrow=c(2,2))
> N <- 10
> p <- c(0.1,0.25,0.5,0.8)
> x <- 0:10
> for(i in 1:4){
+ x1 <- dbinom(x,N,p[i])
+ plot(x1, type="h", xlab="x",
  ylab=expression(P(x)),
+ ylim=c(0,0.4), lwd=2, col="red")
+ j <- p[i]
+ title(main = substitute(p == j,list(j=j)))
+ }
```

Bernoulli distribution

- Likelihood function is function of *p* given $x_{1,\ldots,}x_N$.

- Assume further that we got this observed data:

  0,0,1,0,1,1,0,0,0,1

  so the of sum of $x_{1,\ldots,}x_N$ equals 4.

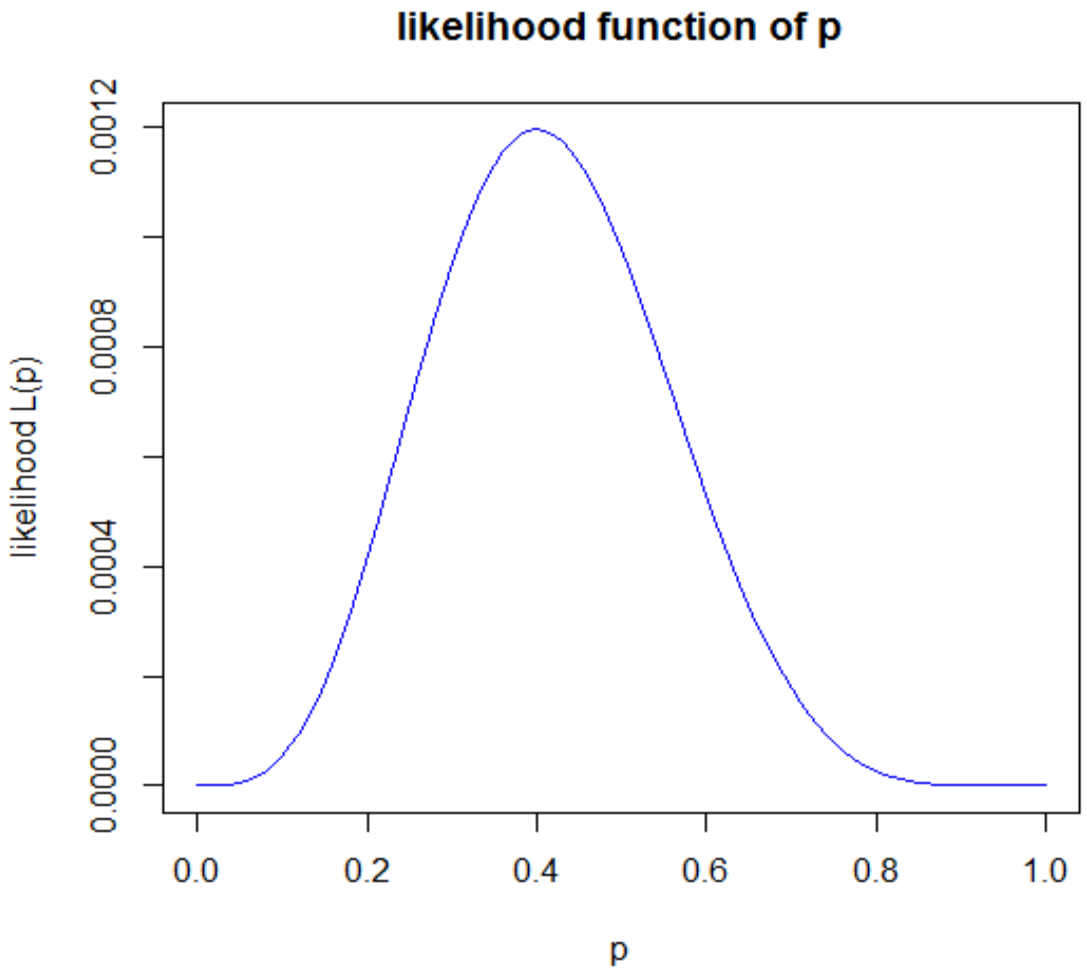
- Then, likelihood function equals

$$L(p; x_1, \ldots, x_N) = p^4(1-p)^{N-4}$$

- For more observations, likelihood function and PDF can be different, but only with respect to constant multiplication.
- They are always proportional.

Bernoulli distribution

Likelihood probability plot



Bernoulli distribution

```
> p <- seq(0,1,0.01)
> d <- length(p)
> N <- 10
> x <- 4
> j <- m <- seq(1:d)
> for(i in 1:d){
+ j[i] <- p[i]^x*(1-p[i])^(N-x)
+ }
> plot(p, j, type="l", col=4, ylab="likelihood
  L(p)", main="likelihood function of p")
```

Bernoulli distribution: Log-likelihood Function

- The log transformation of likelihood function.
- Reason is simplification of further computations.

- Log-likelihood function of Bernoulli probability distribution

$$l(p; x_1, \ldots, x_N) = \ln L(p; x_1, \ldots, x_N)$$

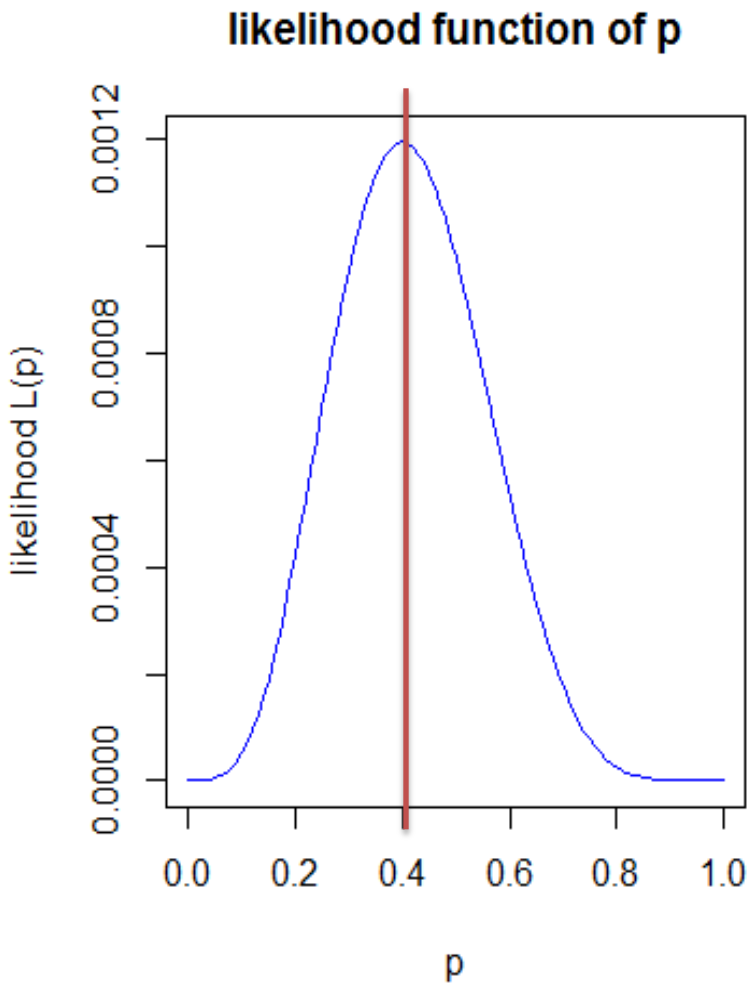
$$l(p; x_1, \ldots, x_N) = \sum_{i=1}^{N} x_i \cdot \ln p + \left(N - \sum_{i=1}^{N} x_i \right) \cdot \ln(1 - p)$$

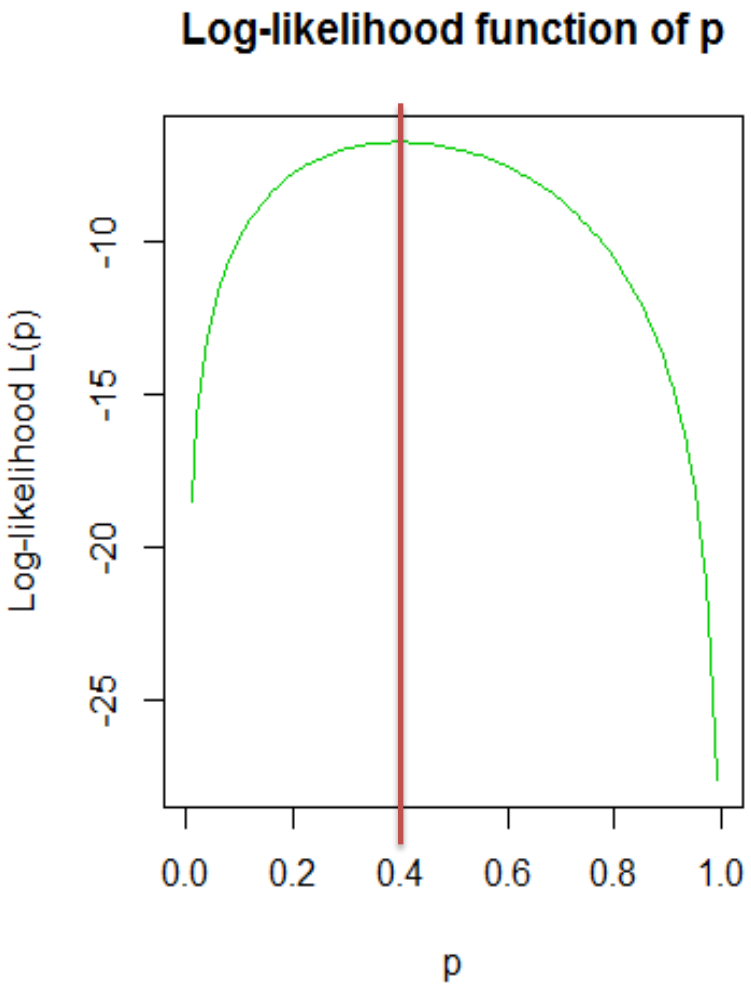
Bernoulli distribution: Log-likelihood Function

- We will be interested in maximum of likelihood function.
- The log-likelihood function and the likelihood function always have the peak (maximum) for the same value of the parameter(s).
- It is monotone transformation of original likelihood function.
- Simplification: at the log part, products become sum and ratios become differences.

Bernoulli distribution: Log-likelihood Function

Plot for likelihood and log-likelihood





Bernoulli distribution: Log-likelihood Function

```
> p <- seq(0,1,0.01)
> d <- length(p)
> N <- 10
> x <- 4
> j <- log.lik <- m <- seq(1:d)
> for (i in 1:d){
+ log.lik[i] <- x*log(p[i])+(N-x)*log(1-p[i])
+ j[i] <- p[i]^x*(1-p[i])^(N-x)
+ }
> par(mfrow=c(1,2))
> plot(p, j, type="l",col=4, ylab="likelihood
  L(p)", main="likelihood function of p")
> plot(p, log.lik, type="l", col=3, ylab="Log-
  likelihood L(p)", main="Log-likelihood function
  of p")
```

Bernoulli distribution: Log-likelihood Function

- Trick in R
- Log-likelihood is written in R

```
log.lik[i] <- x*log(p[i])+(N-x)*log(1-p[i])
```

but we can compute it also through

```
dbinom(x, size=N, prob=p[i], log=TRUE)
```

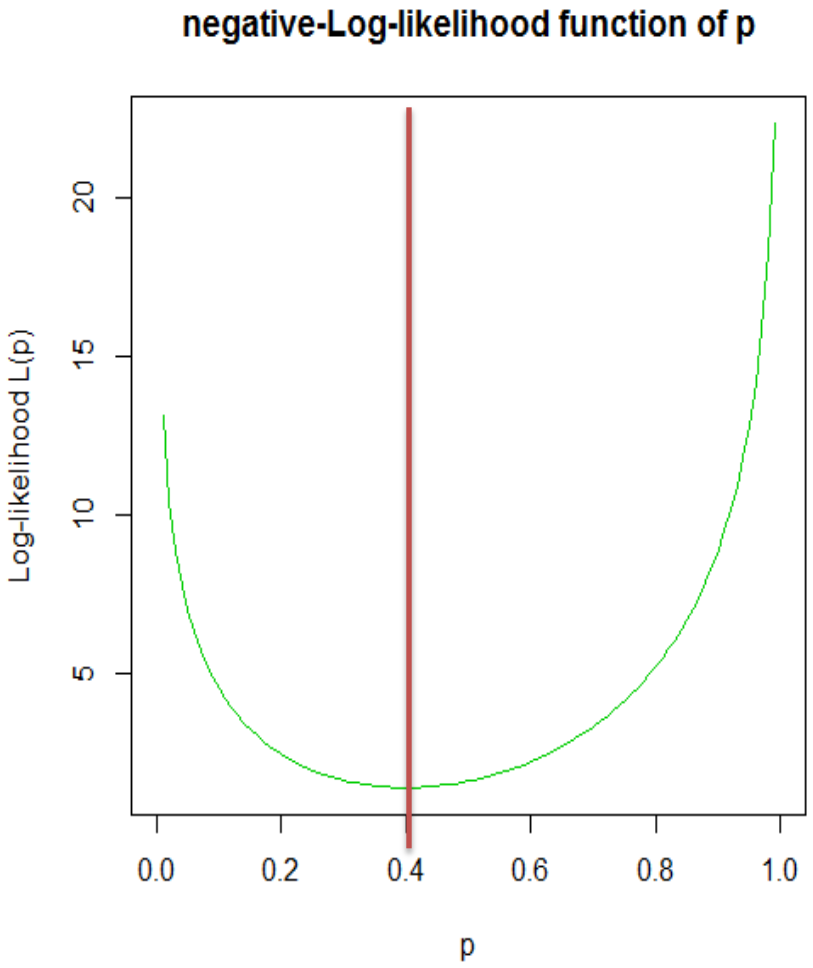
- The dbinom function calculates the binomial likelihood for a specified number of successes x, probability *p*, and number of trials *N*

- The curves will not be same, but they are proportional: their difference is just constant multiplication.
- For maximization, proportionality is all we need.

Bernoulli distribution: Log-likelihood Function

- The negative log likelihood plot

```
> - dbinom(x, prob=p, size=N, log=TRUE)
```



Bernoulli distribution: MLE

Maximum Likelihood estimator (MLE)

- The value of the parameter(s) that maximize the likelihood function is called the maximum likelihood estimate.

- "Most likely value of parameter, given the data."

- In this example you can find the MLE analytically by finding the value of the parameter where the likelihood function is flat (i.e., at the peak), by setting the first derivative to zero.

Bernoulli distribution: MLE

$$\frac{\partial l(p; x_1, \ldots, x_N)}{\partial p} = \sum_{i=1}^{N} x_i \cdot \frac{1}{p} - \left(N - \sum_{i=1}^{N} x_i \right) \cdot \frac{1}{(1-p)}$$

$$0 = \sum_{i=1}^{N} x_i \cdot \frac{1}{p} - \left(N - \sum_{i=1}^{N} x_i \right) \cdot \frac{1}{(1-p)}$$

$$\sum_{i=1}^{N} x_i - p \sum_{i=1}^{N} x_i = Np - p \sum_{i=1}^{N} x_i$$

$$p = \frac{\sum_{i=1}^{N} x_i}{N}$$

Bernoulli distribution: MLE

- Then the MLE will be

$$\hat{p} = \frac{\sum_{i=1}^{N} x_i}{N}$$

```
> mle.p <- x/N
> mle.p
[1] 0.4
```

The variance of $\hat{p}$

$$Var(\hat{p}) = \frac{\hat{p}(1-\hat{p})}{N}$$

```
> mle.p <- x/N
> var.p <- mle.p*(1-mle.p)/N
> var.p
[1] 0.024
```

Bernoulli distribution: MLE

Maximization

- So far, we have seen the value that maximizes the likelihood function of Bernoulli PDF graphically.
- Also, we saw analytical solution for MLE.

- Log likelihood can be sometimes so complicated that analytical solution is no longer possible.
- Hence, numerical optimization of function to find MLE is needed.

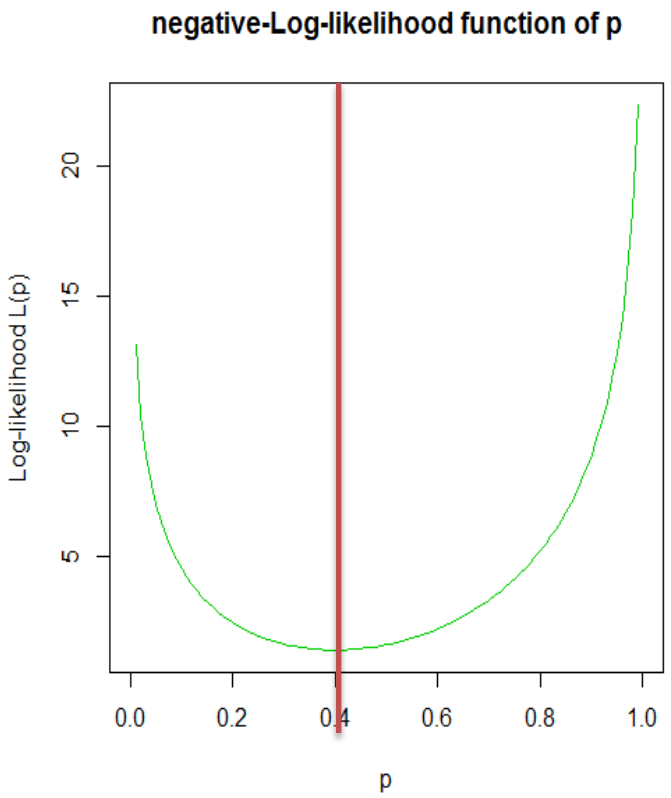
- As example, we search the value that maximizes the Bernoulli likelihood function by *optimize* function.

Bernoulli distribution: MLE

```
> maxim <- function (p,x,N){
+ p^x*(1-p)^(N-x)
+ }
> pmax <- optimize(maxim, c(0, 1), tol = 0.000001,
  N = 10, x=4, maximum=TRUE)
> pmax
```

Out put

```
$maximum
[1] 0.4
$objective
[1] 0.001194391
```



- The value that maximizes the likelihood function (MLE) is 0.4.

Bernoulli distribution: MLE

- The maximum likelihood estimator is

$$\hat{p} = \sum_{i=1}^{N} x_i / N$$

```
> mle.p <- 4/10 = 0.4
```

- The numerical maximization result is 0.4.

- The MLE of *p* that we found here analytically and by *optimize* function is the same.

Binomial Distribution

Binomial probability distribution

- Binomial distribution describes the probability of *x* number of successes in *n* independent trials with probability of success is *p*.

- Binomial probability mass function is given as

$$P(x, p, n) = \binom{n}{x} (p)^x (1-p)^{n-x} \quad \text{for} \quad \mathrm{x} = 0,1,\ldots,\mathrm{n}$$

$$\textit{where,} \qquad \binom{n}{x} = \frac{n!}{x!(n-x)!}$$

Binomial Distribution

- Likelihood function of binomial distribution for one observation:

$$L(p; n, x) = \binom{n}{x} (p)^x (1-p)^{n-x}$$

- This function involves the parameter (*p*) given the data (*x* and *n*)
- We are estimating how likely various values of *p* are given the observed data.

Binomial Distribution

- Data set consists of number of observations $x_{1,\ldots,}x_N$

- Assume:

  $x_{1,\ldots,}x_N$ are independent

  $x_{1,\ldots,}x_N$ are identically distributed: B(n,p)

- Each x_i represents one observation of situation when experiment with *n* trials with probability of success *p* resulted into x_i successes.

- Example:

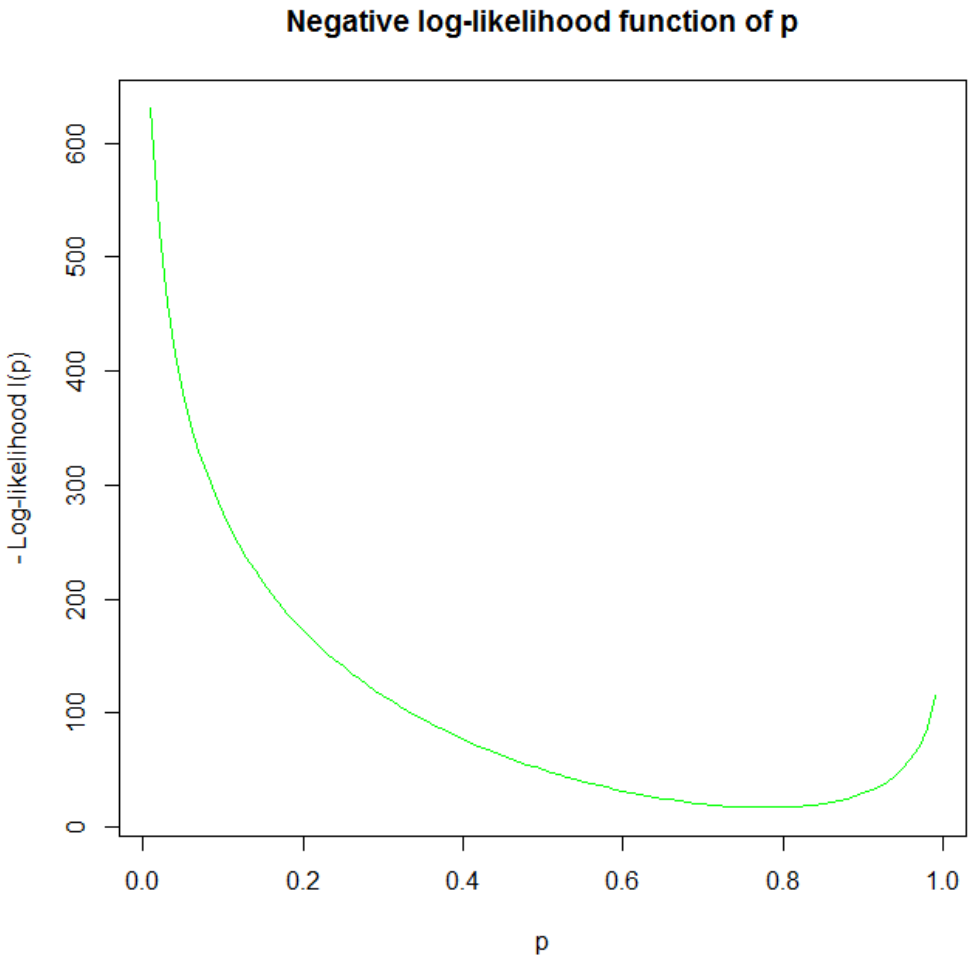
  We perform standardized test in 10 classes with each having 20 students. Numbers of successful students in each class are:

  15, 13, 15, 16, 16, 17, 16, 16, 16, 16

Binomial Distribution

- The negative log likelihood plot looks like

```
> -sum(dbinom(x, prob=p, size=n, log=TRUE))
```



Binomial Distribution

- Likelihood function of binomial probability distribution

$$L(p \mid x, n) = \prod_{i=1}^{N} \binom{n}{x_i} (p)^{x_i} (1-p)^{n-x_i}$$

$$l(p \mid x, n) = \sum_{i=1}^{N} \ln\left[\binom{n}{x_i}\right] + \sum_{i=1}^{N} x_i \cdot \ln(p) + \sum_{i=1}^{N} (n - x_i) \ln(1-p)$$

- First term does not depend on *p*, so it will be omitted when we proceed to derivative. So, log-likelihood is proportional to

$$l(p \mid x, n) \propto \sum_{i=1}^{N} x_i \cdot \ln(p) + \sum_{i=1}^{N} (n - x_i) \ln(1-p)$$

Binomial Distribution

- The solution is very similar to Bernoulli case.
- Both distribution are closely related.
- We saw that sum of Bernoulli distributed variables follows binomial distribution.

- Analytical solution of MLE for Bernoulli variable is:

$$\hat{p} = \frac{\sum_{i=1}^{N} x_i}{n \cdot N}$$

- Example: $\hat{p} = 146/200 = 0.73$

Poisson Distribution

Poisson probability distribution

- Typically used to describe count data without restriction on maximum.

- The probability mass function is given as:

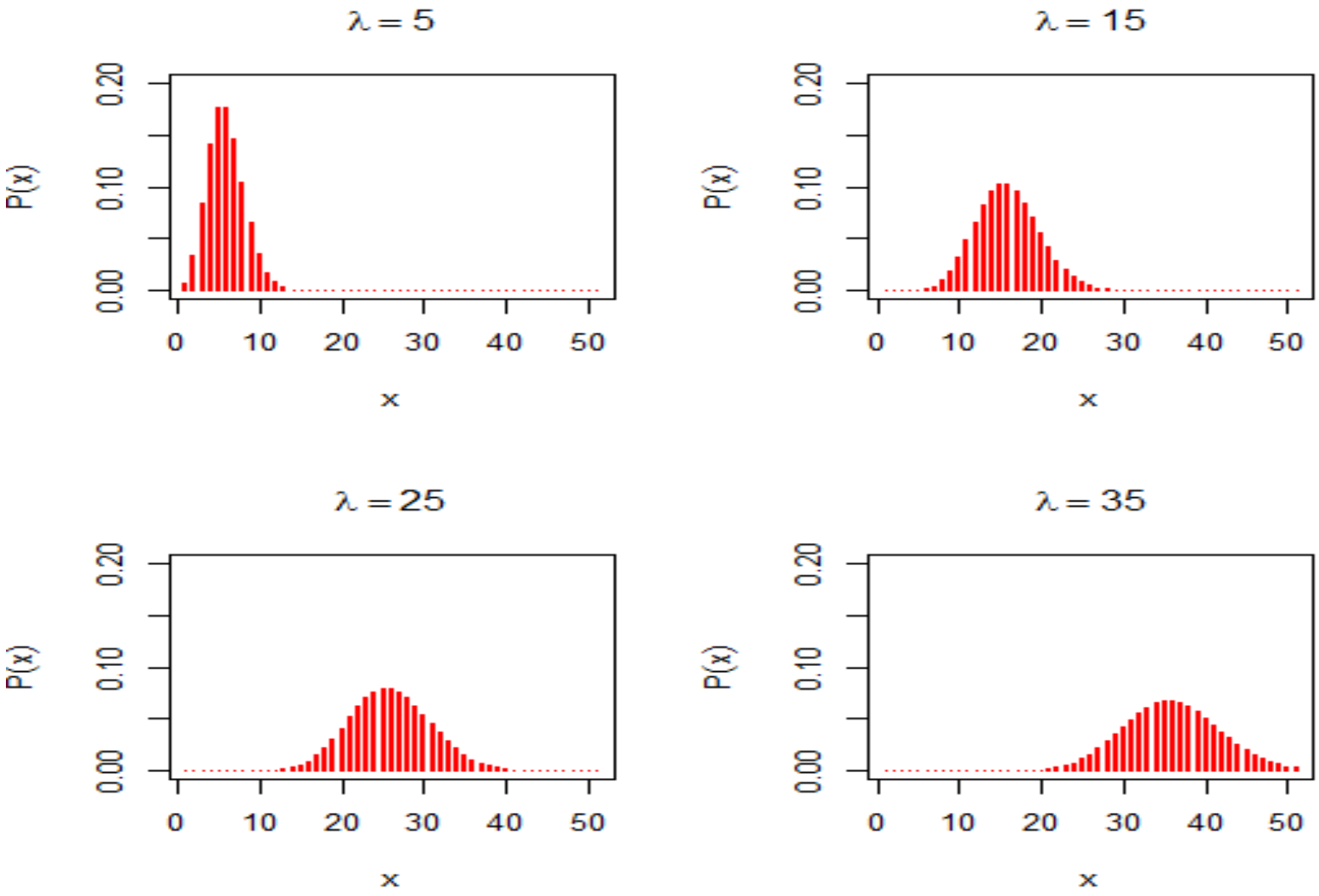
$$P(x;\lambda) = P(X = x) = \frac{\lambda^x e^{-\lambda}}{x!}, x \in [0, \infty)$$

- The parameter is λ.
- It represents frequency of event in some time period.
- For a given λ, what is the probability of X values?

- Let us see for different value of X, and also λ.

Poisson Distribution

- Line plot of Poisson PDF for different lambda



Poisson Distribution

```
> par(mfrow=c(2,2))
> x <- seq(0,50,1)
> lambda <- c(5,15,25,35)
> t <- length(lambda)
> for (i in 1:4){
+ prob <- dpois(x, lambda[i], log = FALSE)
+ plot(prob, type="h", xlab="x",
+ ylab = expression(P(x)),
+ ylim=c(0,.2),lwd=2, col="red")
+ j <- lambda[i]
+ title(main = substitute(lambda == j, list(j=j)))
+ }
```

Poisson Distribution

- Data set consists of number of observations $x_{1,\ldots,}x_N$

- Assume:

  $x_{1,\ldots,}x_N$ are independent

  $x_{1,\ldots,}x_N$ are identically distributed: Pois(λ)

- Each x_i represents number of observation of particular event that occurred in fixed period of time during experiment with frequency of events λ.

- Example 1: We observed the following data from Poisson distribution

  5, 0, 1, 1, 0, 3, 2, 3, 4, 1

Poisson Distribution

- Likelihood function of Poisson probability distribution

$$L(\lambda; x_1,\ldots, x_N) = \prod_{i=1}^{N} f(x_i;\lambda)$$

$$= \frac{\prod_{\mathrm{i}=1}^{\mathrm{N}} \lambda^{\mathrm{x_i}} e^{-\lambda}}{x_i!} = \frac{\lambda^{\sum_{i=1}^{N} x_i} e^{-N\lambda}}{x_1! x_2! \ldots x_N!}$$

- For Example 1, likelihood function is

$$L(\lambda; x_1,\ldots, x_N) = \frac{e^{-10\lambda}\lambda^{\sum x_i}}{\prod x_i!} = \frac{e^{-10\lambda}\lambda^{20}}{207360}$$

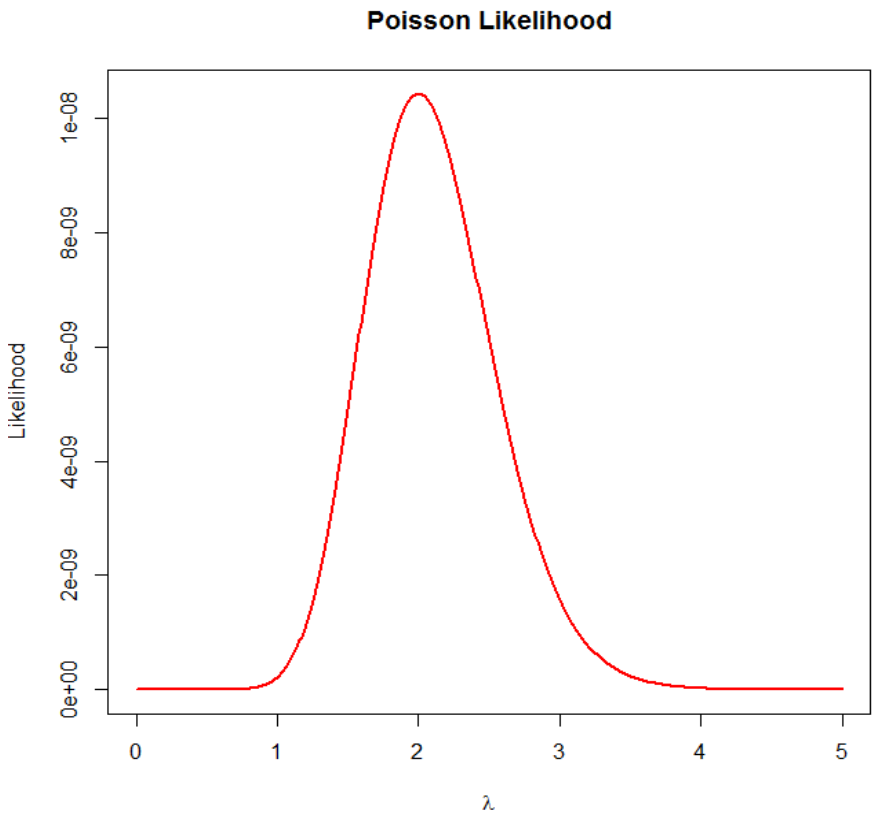
Poisson Distribution

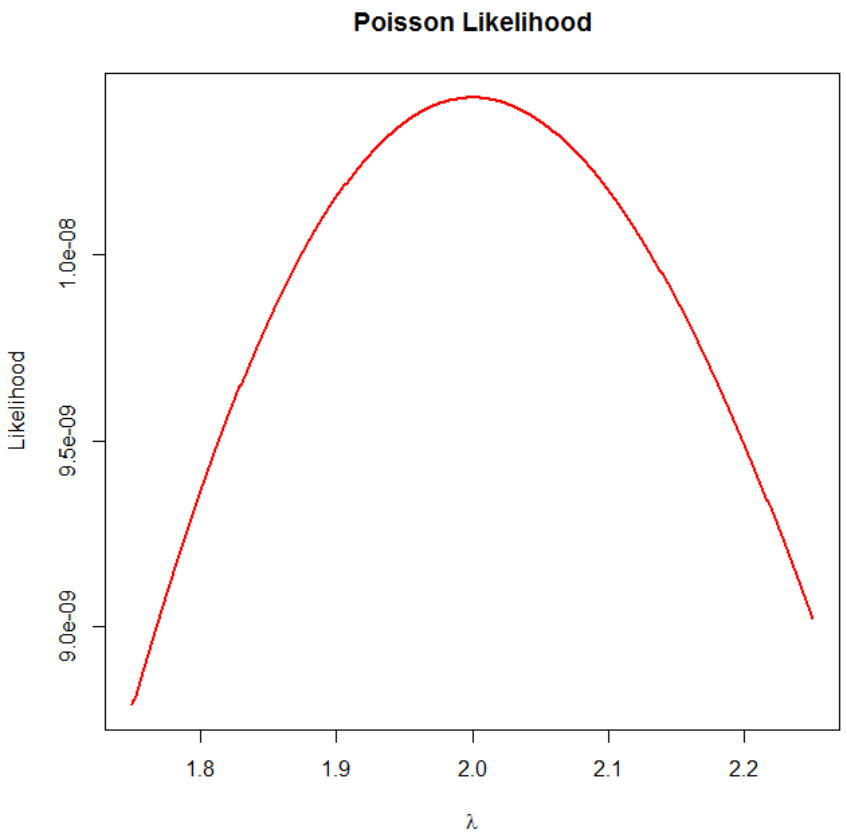
- We need to search over values of λ to find the value of MLE for λ

- Let us start with graphical representation.
- Let us investigate the λ for a sequence of values from 0 to 5 in increments of 0.01.
- Then, plot for different sequence of λ (between 1.75 and 2.25) and compare the plot to the previous one.

Poisson Distribution

- Plot of likelihood functions for two ranges of λ





Poisson Distribution

```
> lambda <- seq(0,5,0.01)
> like <- ((exp(-10*lambda))*(lambda^20))/207360
> plot(lambda, like, type="l", col=2, lwd=2,
  main="Poisson Likelihood",
  xlab=expression(lambda), ylab="Likelihood")
```

- For different sequence

```
> lambda2 <- seq(1.75,2.25,0.001)
> like2 <- (exp(-10*lambda2)*lambda2^20)/207360
> plot(lambda2, like2, type="l", col=2, lwd=2,
  main="Poisson Likelihood",
  xlab=expression(lambda), ylab="Likelihood")
```

Poisson Distribution

- In the Poisson likelihood, the denominator, $\prod_{i=1}^{n} x_i!$ is a constant in the likelihood function that doesn't depend on λ.
- This term divides the value of the likelihood by value of 207360, regardless of the value of λ.
- Dividing by a constant doesn't change the location of the maximum at all.
- Hence, same as for binomial distribution, it does not influence value of MLE.

- Therefore we can drop this term.

$$L(\lambda; x_1, \ldots, x_N) \propto \lambda^{\sum_{i=1}^{N} x_i} e^{-N\lambda}$$

Poisson Distribution

- Full log-likelihood of Poisson distribution would be

$$l(\lambda; x_1, \ldots, x_N) = \sum_{i=1}^{N} x_i \cdot \ln(\lambda) - n\lambda - \ln\left(\prod_{i=1}^{N} x_i!\right)$$

- After dropping unnecessary term, the log-likelihood is proportional to:

$$l(\lambda; x_1, \ldots, x_N) \propto \sum_{i=1}^{N} x_i \cdot \ln(\lambda) - n\lambda$$

- Note: derivative of dropped term would be equal to zero, so the term would not influence our results.

Poisson Distribution

- Derivate the log-likelihood function with respect to the parameter λ and equate it zero

$$\frac{\partial l(\lambda; x_1,\ldots,x_N)}{\partial p} = \frac{\sum_{i=1}^{N} x_i}{\lambda} - N\lambda = 0$$

- The maximum likelihood estimator would be

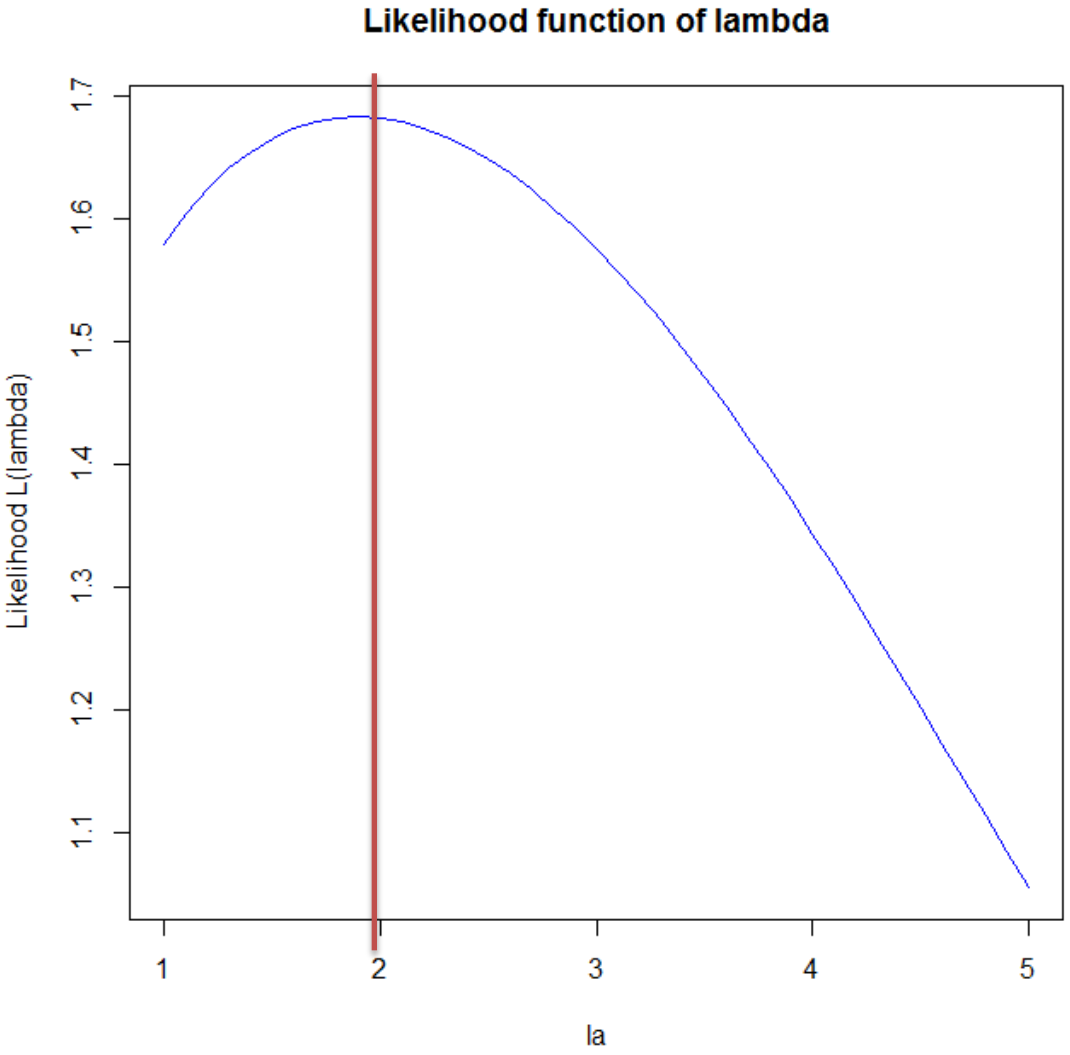
$$\hat{\lambda} = \frac{\sum_{i=1}^{N} x_i}{N}$$

- Therefore, the MLE for the Example 1 would be:

```
> x <- c(5,0,1,1,0,3,2,3,4,1)
> mle.la2 <- sum(x)/length(x)
> mle.la2
[1] 2
```

Poisson Distribution

Plot of likelihood function for Example 1



Poisson Distribution

- Example 2:

  Number of births per every 6 hours seen in Jimma University.

  Suppose this data follows Poisson distribution.

  Which parameter value is the most plausible for this observed outcome?

- Data:

  2, 2, 0, 1, 3, 6, 4, 4, 4, 1

```
> x <- c(2, 2, 0, 1, 3, 6, 4, 4, 4, 1)
> mle.la <- sum(x)/length(x)
> mle.la
[1] 2.7
```

Poisson Distribution

```
> lam <- seq(1,5,0.1)
> t <- length(lam)
> x <- c(2, 2, 0, 1, 3, 6, 4, 4, 4, 1)
> su <- sum(x)
> me <- mean(x)
> pro <- c(1:t)
> for (i in 1:t){
+ pro[i]<-(lam[i]^su)*(exp(-3*lam[i]))/288
+ }
> plot(lam, pro, type="l", col=4,
  ylab="likelihood", main="likelihood function")
```

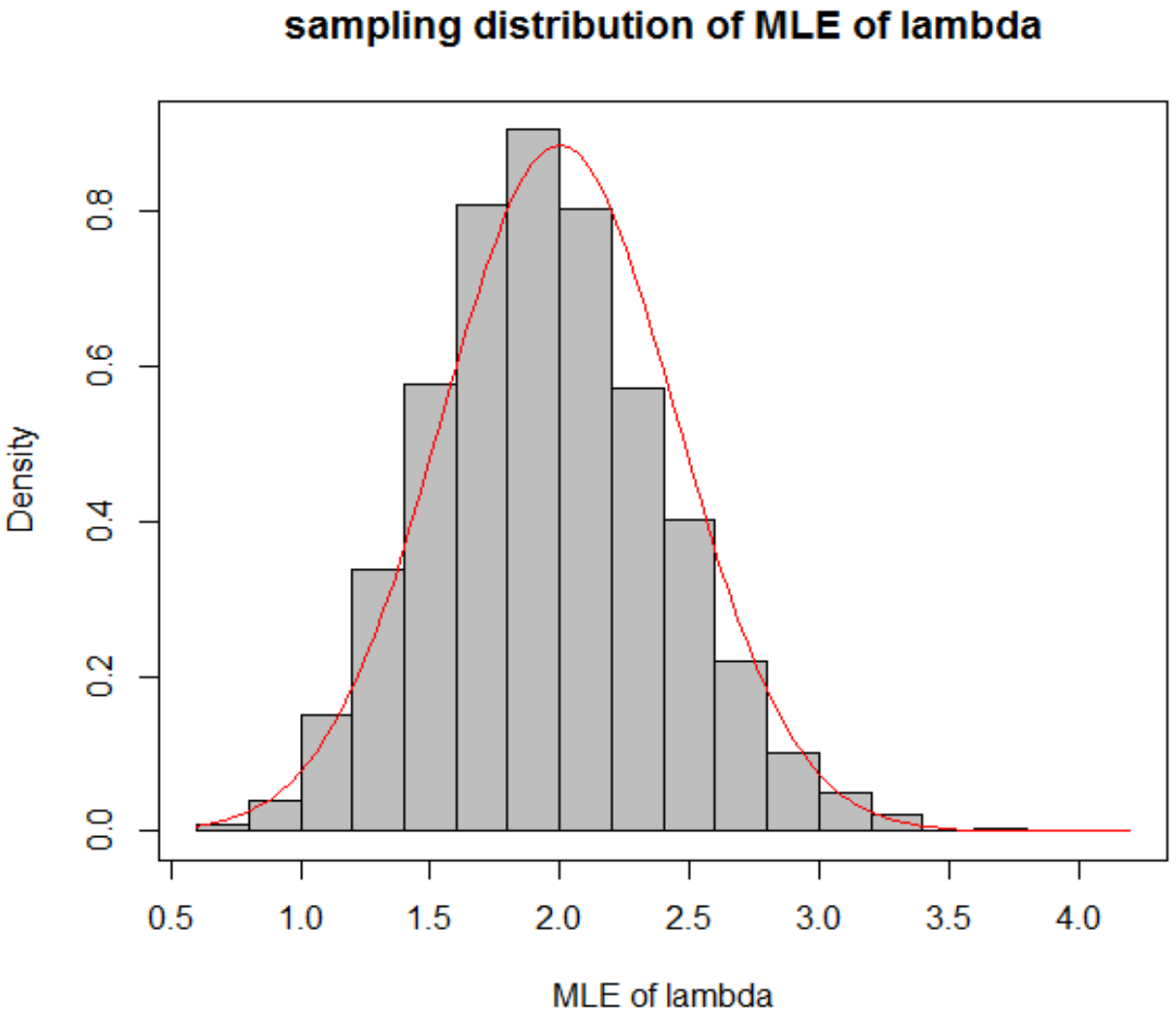
Poisson Distribution: Bootstrap for MLE

- Sampling distribution of MLE of Poisson

1. We calculated the estimated lambda from our data λ_o.
2. Take 5000 random samples from Poisson distribution with parameter value λ_o.
3. Compute the MLE of λ for each sample.
4. Make histogram for estimated MLEs and compute the standard error.

Poisson Distribution: Bootstrap for MLE

- Histogram of estimated lambda



Poisson Distribution: Bootstrap for MLE

```
> poi.data <- c(5,0,1,1,0,3,2,3,4,1)
> lambda <- mean(poi.data)
> t <- 5000
> x <- boot.mean <- seq(1:t)
> for (i in 1:t){
+ boot.sample <- rpois(10,lambda)
+ boot.mean[i] <- mean(boot.sample)
+ }
> lambda
[1] 2
> mean(boot.mean)
[1] 2.00444
> sd(boot.mean)
[1] 0.444342
```

Poisson Distribution: Bootstrap for MLE

```
> hist(boot.mean, freq = FALSE, col = "grey",
  xlab="MLE of lambda", main="sampling
  distribution of MLE of lambda")
> box()
> curve(dnorm(x, mean=mean(boot.mean),
  sd=sd(boot.mean)), col='red', add=TRUE)
```

Poisson Distribution: Numerical MLE

- Example 3:

```
> y <- c(4,1,4,3,0,2,3,1,0,1)
```

Given this data, find the value that maximizes the log-likelihood function of Poisson distribution.

- We search maximum numerically using *optim* function.

- Note, that the maximum likelihood estimator of Poisson is given as

$$\hat{\lambda} = \frac{\sum_{i=1}^{N} x_i}{N}$$

```
> y <- c(4,1,4,3,0,2,3,1,0,1)
> mean(y)
[1] 1.9
```

Poisson Distribution: Numerical MLE

```
> poisson.lik <- function(mu,y){
+ y <- c(4,1,4,3,0,2,3,1,0,1)
+ N <- length(y)
+ b <- sum(y)
+ logl <- b*log(mu)-N*mu
+ }
> xmax <- optimize(poisson.lik, c(0, 5), tol =
  0.000001, maximum=TRUE)
> xmax
$maximum
[1] 1.9
$objective
[1] -6.804776
```

Poisson Distribution: Numerical MLE

- The value 1.9 maximize the log-likelihood function for Poisson distribution, given the data.
- Hence, it is the MLE estimate.
- It is same as analytical solution.

- The value -6.804776 is maximum value of the log-likelihood function.
- It is the value that function have in MLE.

- We are not interested in values of function itself.
- In general maximization problem, it can be relevant.

Optimize function

- General optimization tool in R for one parameter.
- Several optimization methods available.
- Numerical iterative procedures that updates starting values to reach minimum/maximum.

- Offers flexibility to control the procedure (maximum number of iterations, tolerance for stopping rules, tracing the procedure and much more)
- Different types of stopping rules.

- Returns maximum that function achieves and the value in which the maximum is achieved.
- For MLE, only the latter is important.

Poisson Distribution: Exercise

- Exercise: Let us do numerical estimation for the data found above

```
> y <- c(5,0,1,1,0,3,2,3,4,1)
```

- As we have seen the MLE of λ is 2.
- Compute the MLE by optimize function.

Normal distribution

- **Normal Probability Distribution**
- The probability density function of normal distribution $N(\mu, \sigma^2)$ is given as:

$$f(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

- For the normal distribution, there are two parameters, μ and σ^2.
- Parameter μ represents mean, parameter σ^2 variance.
- Often both of them are unknown.
- So instead of only one parameter to estimate we must find MLE for both μ and σ^2.

Normal distribution

- The normal likelihood function is given as:

$$L(\mu, \sigma^2; x_i) = (2\pi\sigma^2)^{-\frac{1}{2}} e^{-\frac{1}{2}[(x_i-\mu)^2/\sigma^2]}$$

- Let us assume that we have observations $x_{1,\ldots,}x_N$ that are all independent and distributed according to $N(\mu, \sigma^2)$.
- Then, log-likelihood function is

$$\ln L(\mu, \sigma^2; x_1, \ldots, x_N) = -\frac{N}{2}\ln(2\pi) - \frac{N}{2}\ln(\sigma^2) - \frac{1}{2}\sum_{i=1}^{N}\frac{(x_i-\mu)^2}{\sigma^2}$$

- To get the MLE of mean and variance we need to derivate this function with respect to each parameter.

Normal distribution

- Derivatives with respect to mean and variance

$$\frac{\partial l}{\partial \mu} = \frac{1}{\sigma^2}\sum_{i=1}^{N}(x_i - \mu)$$

$$\frac{\partial l}{\partial \sigma} = -\frac{N}{\sigma} + \sigma^{-3}\sum_{i=1}^{N}(x_i - \mu)^2$$

- Now, the derivative should be equate to zero. Then, MLEs would be

$$\hat{\mu} = \frac{\sum_{i=1}^{N} x_i}{N} \qquad \hat{\sigma}^2 = \frac{\sum_{i=1}^{N}(x_i - \hat{\mu})^2}{N}$$

Normal distribution

- Note: it is important that MLE of μ does not depend on σ^2
- That allows us to first compute MLE of μ and then plug-in this estimate into formula for σ^2

- If μ would depend on σ^2 *(*and σ^2 on μ*),* we have to use iterative numerical procedures to maximize simultaneously.
- It is in general much more difficult and it is happening for most of distributions with 2 parameters.

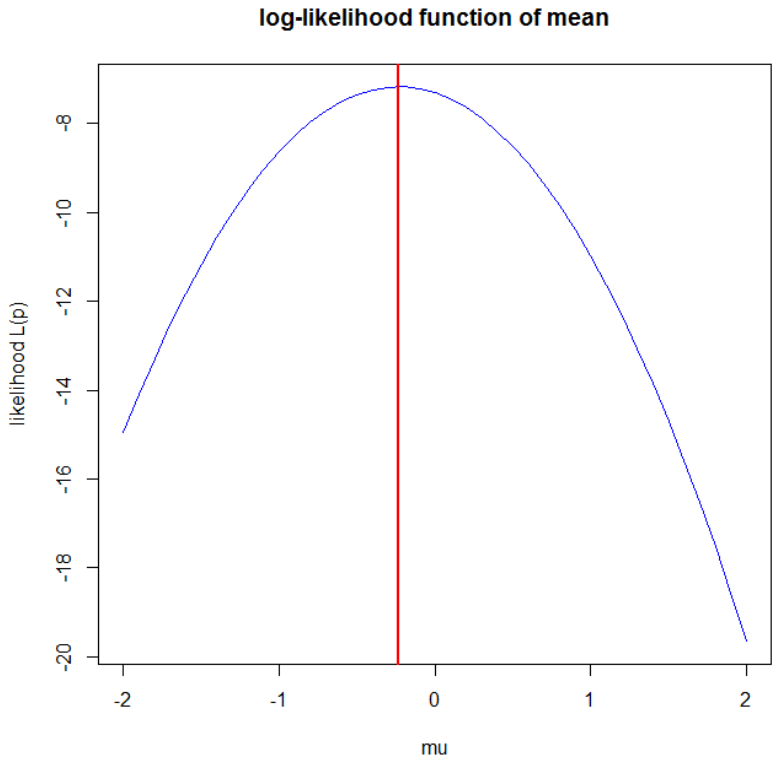
- Optim function shows before works for more parameters.

Normal distribution

- Example 1: The data set below is obtained from a normal population with variance 1 and unknown mean.

```
> y <- c(1.364, 0.235, -0.846, -0.285, -1.646)
```

- Let us see the normal log-likelihood function curve for unknown mean and known variance to see the plausible value of population mean



Normal distribution

```
> normal.lik1 <- function(mu,sigma2,y){
+ N <- length(y)
+ t <- length(mu)
+ logl <- array(1,t)
+ for (i in 1:t){
+ logl[i] <- -0.5*N*log(2*pi)-0.5*N*log(sigma2)-
  (1/(2*sigma2))*sum((y-mu[i])**2)}
> return(logl)
> }
> y <- c(1.364, 0.235, -0.846, -0.285, -1.646)
> mu <- seq(-2, 2,0.1)
> sigma2 <- 1
> logl <- normal.lik1(mu,sigma2,y)
> plot(mu, logl, type="l",col=6, ylab="likelihood
  L(p)", main="log-likelihood function of mean")
```

Normal distribution

- Variance is known here, $\sigma^2 = 1$.
- Using formula $\hat{\mu} = \frac{\sum_{i=1}^{N} x_i}{N}$
- We get MLE for μ:

```
> mean(y)
[1] -0.2356
```

Normal distribution

- Example 2: Again, values are drawn from normal distribution with variance 1 and unknown mean μ.

```
> y2 <- c(0.9898790, 1.4552127, -1.5438397, -
2.3792939,-0.3809298)
```

- We know that MLE of μ can be obtained using formula:

```
> mean(y2)
[1] -0.3717943
```

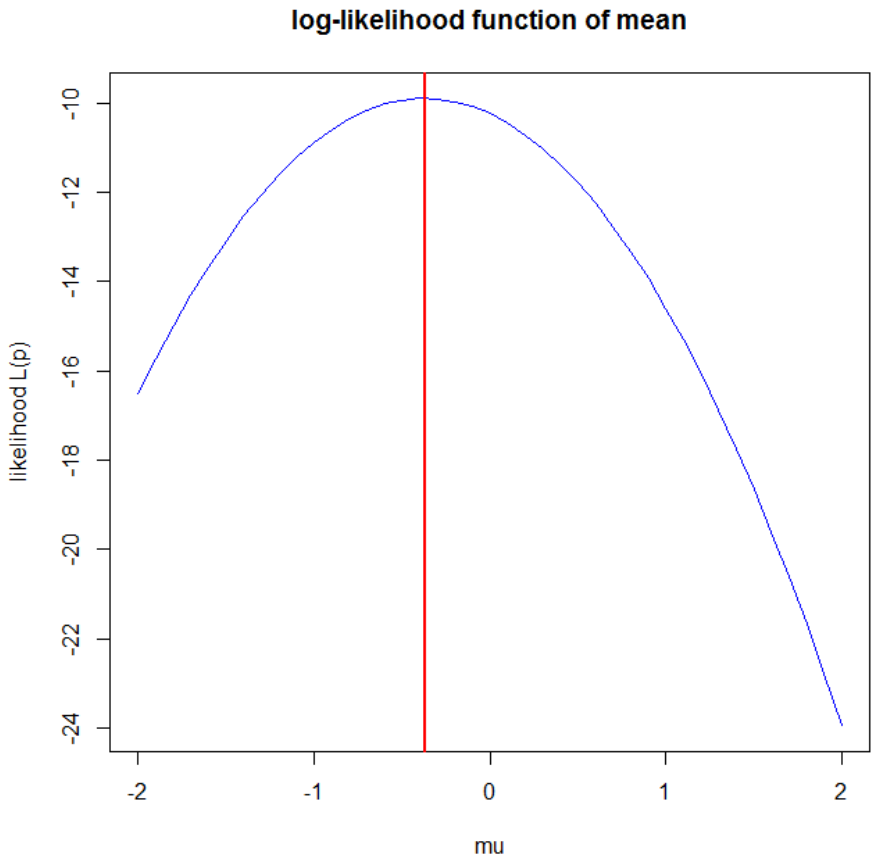
- Now, we go to find MLE through numerical maximization of the log-likelihood function.

Normal distribution

```
> g <- function(mu){
+ sigma2 <- 1
+ y2 <- c(0.9898790,1.4552127,-1.5438397,-
  2.3792939,-0.3809298)
+ N <- length(y2)
+ t <- y-mu
+ h <- t**2
+ logl <- -0.5*N*log(2*pi)-0.5*N*log(sigma2)-
  (1/(2*sigma2))*sum(h)
+ }
> xmax <- optimize(g, c(-2, 2), maximum=TRUE)
```

Normal distribution

```
> xmax
$maximum
[1] -0.3717943
$objective
[1] -9.892661
```



- The value -0.3717943 is the MLE of μ for a give data. Log-likelihood function has a maximum point for this value.
- The value -9.892661 is the maximum value for the log-likelihood function .

Normal distribution

- Example 3: We assume same observations, i.e. values are drawn from normal distribution, but now with unknown variance σ^2 and unknown mean μ.

```
> y2 <- c(0.9898790, 1.4552127, -1.5438397, -
2.3792939,-0.3809298)
```

- In normal probability distribution there are two unknown parameters: the population mean and variance.
- There is MLE for both parameters. It can be found by simultaneous maximization of log-likelihood function.

Normal distribution

- Using formulas

$$\hat{\mu} = \frac{\sum_{i=1}^{N} x_i}{N} \qquad \hat{\sigma}^2 = \frac{\sum_{i=1}^{N} (x_i - \hat{\mu})^2}{N}$$

- We get MLEs

```
> mean(y2)
[1] -0.3717943
> mean((y2-mean(y2))^2)
[1] 2.119187
```

- Other option is numerical optimization for both parameters simultaneously.
- Optimize cannot be used for two parameters!
- Instead, more general function optim can be used in very similar way.

Optim function

- General optimization tool in R for any number of parameters.
- Flexibility even better than for optimize.

- Several differences in notation:

1. Instead of range, the starting values are specified.
2. All parameters are grouped into one vector for input.
3. As default, minimization is performed!

   Either define function as minus log-likelihood or use control$fnscale=-1 option (see below in code).

- The results usually do not depend on starting values, but it can happen (local maxima). More starting values should be tested to see, if there is any change in results.

Normal distribution

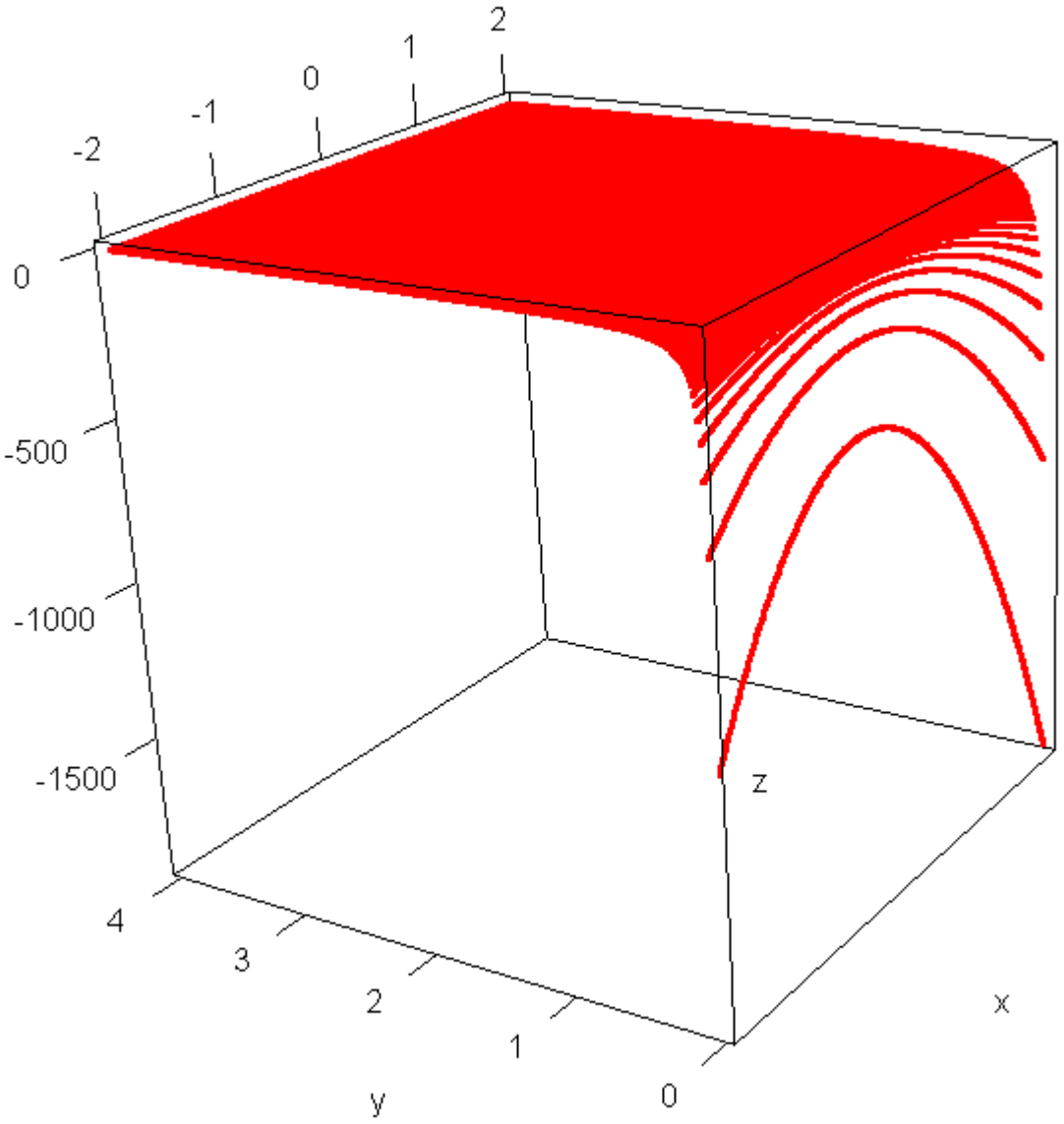
```
> g2 <- <-function(theta){
+ y2 <- c(0.9898790,1.4552127,-1.5438397,-
  2.3792939,-0.3809298)
+ mu <- theta[1]
+ sigma2 <- theta[2]
+ N <- length(y2)
+ logl<- -0.5*N*log(2*pi)-0.5*N*log(sigma2)-
  (1/(2*sigma2))*sum((y2-mu)^2)
+ }
> max.theta <- optim(c(0,1), g2, method="Nelder-
  Mead", control = list(fnscale = -1))
```

Normal distribution

```
$par
[1] -0.3718688  2.1184747
$value
[1] -8.972275
$counts
function gradient
      55       NA
$convergence
[1] 0
$message
NULL
```

- The MLEs are -0.3719558 for μ and 2.1191054 for σ^2.
- The values are same as results that we got using exact formulas.

Normal distribution



Normal distribution

```
> library(rgl)
>
> x <- rep(seq(-2,2,by=0.01),401)
> y <- rep(seq(0.01,4.01,by=0.01),each=401)
> theta <- data.frame(x,y)
> z <- apply(theta, MARGIN=1, g2)
> plot3d(x,y,z,col="red")
```