Basic skills in Bootstrap using R

Ziv Shkedy, Hasselt University (August, 2020)

Introduction

An Introduction to the Bootstrap: skills and theory...

This book contains all materials and R code used to produce the examples and results for the course "computer intensive methods using R: an introduction". We focus in this book on both skills and the theory behind. In other words, we focus on the question "How to do it ?" and not just of the question "why to do it ?". The approach we take in the book is to open the black box and to program the procedures and not to just use existing R functions/packages to produce the results. All examples are illustrated using the R software. We follow the book "An Introduction to the Bootstrap" by Efron and Tibshirani and illustrate the main concepts of bootstrap using the examples/illustrations from their book

R ?

Only basic knowledge in R is required. All the models discussed in the book can be fitted using the `lm()` and `glm()` functions in R. The datasets used for illustrations are available in R and many of them are part of the `bootstrap` R package. To run the code smoothly, this package should be installed.

```
library(bootstrap)
```

A for loop in R

Random samples from $N(0,1)$

A for loop in R is a loop in which we repeatedly ask R to do the same action in each step of the loop. For example, suppose that we would like to draw a sample of 10 observations from $N(0,1)$. In R this can be done using the code

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

The sample is

```
x
```

```
##  [1]  0.7151394  0.3025032 -0.2803666 -0.5759206  0.5728859  0.7599535
##  [7]  0.3393546 -0.1204576  1.5885959  0.0154381
```

and the sample mean, the R object mx,

```
mx<-mean(x)
mx
```

```
## [1] 0.3317126
```

Suppose that we would like to draw a sample of 10 observations from $N(0,1)$ 1000 times. To do this we can use a "for loop" in the following way:

```
mx<-c(1:1000)
for(i in 1:1000)
{
x<-rnorm(10,0,1)
mx[i]<-mean(x)
}
```

Note that the R object mx is a vector contains 1000 sample means. A histogram of the sample means is shown in Figure~@ref(fig:figchp11)

```r
hist(mx,nclass=20)
```

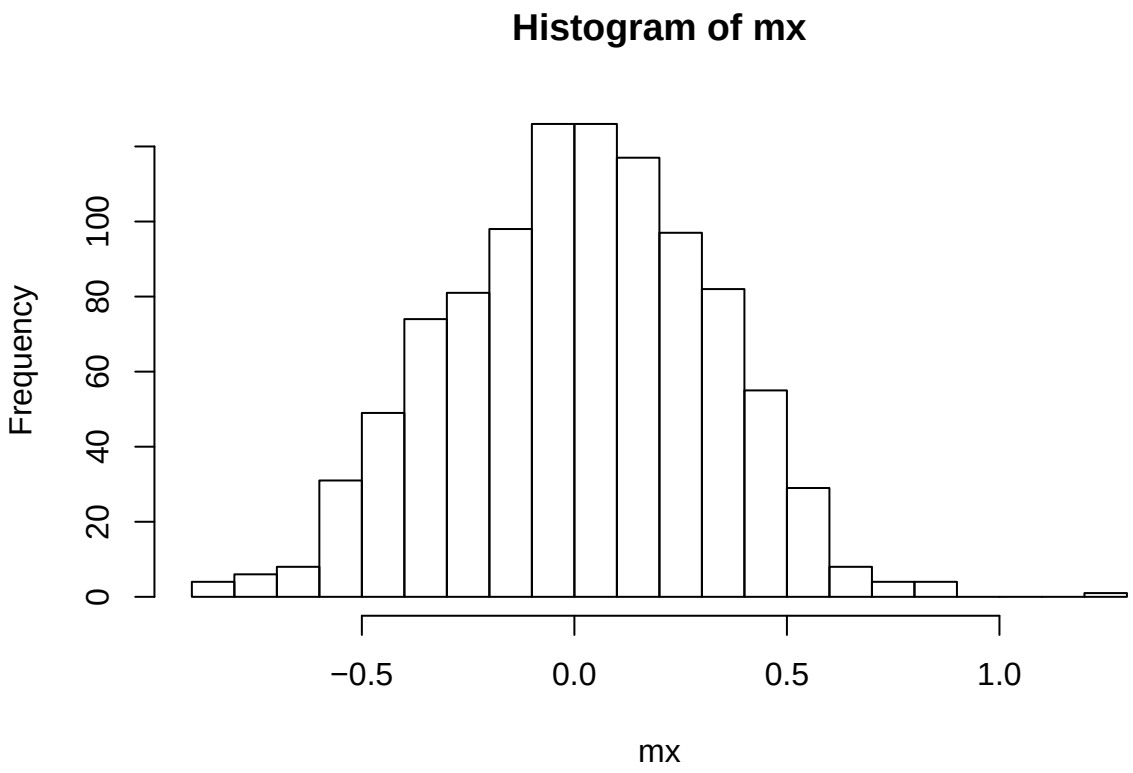


Figure 1: Histogram for 1000 sample means

Just do it

If you did not have a problem to understand the R code above, you will not have any problem to understand the R code that we use to produce all the output for the examples discuss in this book. If you had a difficulty to understand the "for loop" example above you need a short training, at a beginner level, in R.

YouTube tutorial: the for loop in R

For a short online YouTube introduction by Richard Webster about a for loop in R see YTBootstrap 1.

R course online

An introductory course for R is able online in the >eR-BioStat website. See Rcourse.

Notation and book's structure

Throughout the book we keep, as much as we can, the notaion presented in Efron and Tibshirani (1993) "An Introduction to the Bootstrap". The ineractive book is organized in 6 parts:

- Part 1: Introduction.
- Part 2: Basic concepts of bootstrap.
- Part 3: Bootstrap confidence intervals.
- Part 4: Resampling based inferen
- Part 5: Bootstrap methods for linear and generalized linear models.
- Part 6: Bias estimation, Jackknife, Cross-Validation and Non parametric regression.

Links to power point presentation and YouTube tutorials for each topic are given as a part of the book.

Part I
Introduction

The empirical distribution function and the plug-in principle

In the first part of this book we cover the basic concepts that will be used in later chapters of the book:

- Sampling from a population.
- The plug-in principle.
- The empirical distribution.
- The accuracy of the sample mean, the standard error and estimated standard error.

Slides

Slides for the first part that cover the topics of sampling for a population, the accuracy of the sample means and the glug-in principle can be found here: Slides1.

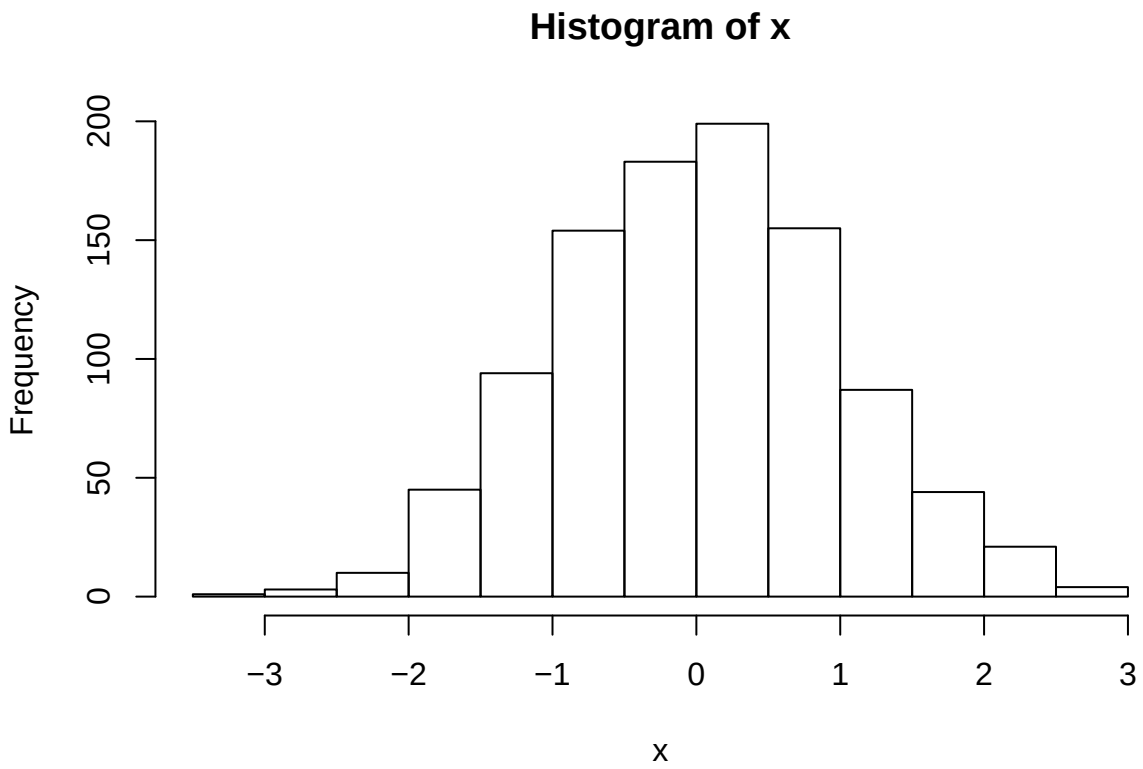
Sampling for a population

Let $x_1, \ldots, x_n$ be a random sample from a probability distribution F and let $\theta = \theta(F)$ be a parameter of interest. For example, let F be a standard normal distribution, $F = N(\mu = 0, \sigma^2 = 1)$ and consider a sample of 1000 observations,

```
x<-rnorm(1000,0,1)
```

The distribution of the sample is shown in Figure~@ref(fig:figchp21)

```
hist(x)
```



The parameter of interest is the mean, $\mu = 0$, and the parameter estimate $\hat{\mu} = \bar{x}$ is equal to

```
mean(x)
```

```
## [1] 0.01266752
```

We note that $\bar{x}$ is defined on the sample, that is $\hat{\mu} = \bar{x} = T(x_1, \ldots, x_n) = \frac{1}{n}\sum_{i=1}^{n} x_i$.

Example: the low school data

The law school data gives information about a random sample of size $n = 15$ from the population of 82 USA law schools. The data consists of two measurements: LSAT (average score of a national law test) and GPA (average undergraduate grade-point average). The R object law82 is a data frame contains data for the whole population of 82 law schools. The data frame law is the sample of 15 law schools (out of the 82). The left panel in Figure @ref(fig:figchp22) below present the population of 82 scools while the right panel the random sample of 15 schools out the population.

```
par(mfrow=c(1,2))
plot(law82$LSAT,law82$GPA,xlim=c(450,700),ylim=c(2.5,3.5))
points(law$LSAT,law$GPA,pch="o",col=2)
plot(law$LSAT,law$GPA,pch="o",xlim=c(450,700),ylim=c(2.5,3.5))
```

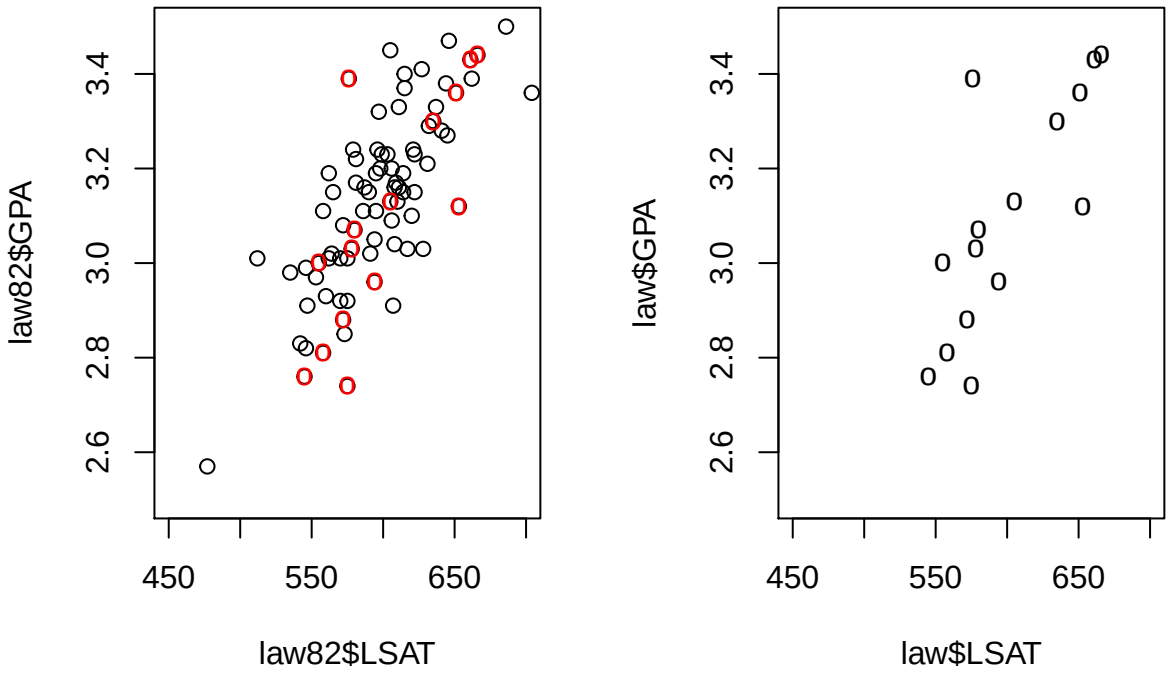


Figure 2: The low schools data

The population correlation (the parameter) is equal to

```
cor(law$LSAT,law$GPA)
```

```
## [1] 0.7763745
```

and the sample correlations (the parameter estimate) is

```
cor(law$LSAT,law$GPA)
```

```
## [1] 0.7763745
```

The plug-in principle

Let $x_1, \ldots, x_n$ be a random sample from F and $\theta = t(F)$ the paramter of interest. The plug-in estimate for θ is same function t defined on the observed data (i.e., the emperical distribution),

$$\hat{\theta} = t(x_1, \ldots, x_n).$$

Example: the low school data

For the low school example, the parameter of primary interest is the correlation between LAST and PGA in the population of the 82 schools

```
cor(law82$LSAT,law82$GPA)
```

```
## [1] 0.7599979
```

while the plag-in estimate is the correlation between the two variables in the sample of 15 schools

```
cor(law$LSAT,law$GPA)
```

```
## [1] 0.7763745
```

The empirical distribution

Let us assume that $x_1, \ldots, x_n$ is a random sample from a probability distribution $F(\theta)$. The empirical distribution function, $\hat{F}$ is defined to be the discrete distribution that puts probability of $\frac{1}{n}$ on each value of x_i. An example of a density function of $N(0,1)$ and an approximation of the density by a histogram is shown in Figure @ref(fig:figchp23).

```
par(mfrow=c(1,2))
x<-seq(from=-3,to=3,length=1000)
dx<-dnorm(x,0,1)
plot(x,dx,type="l")
title("F=N(0,1)")
x1<-rnorm(10000,0,1)
hist(x1,nclass=50,col=0,probability=T,main="Sample from N(0,1), n=10000")
```

As mentioned above, for a sample from F of size n, the empirical distribution puts a probability of $\frac{1}{n}$ on each observations. Let $\hat{f}_k$ the observed frequency, $k = 1, 2, \ldots K$,

$$\hat{f}_k = \frac{\#x = k}{n}.$$

The empirical distribution, $\hat{F}$ is given by

$$\hat{F} = (\hat{f}_1, \ldots, \hat{f}_K).$$

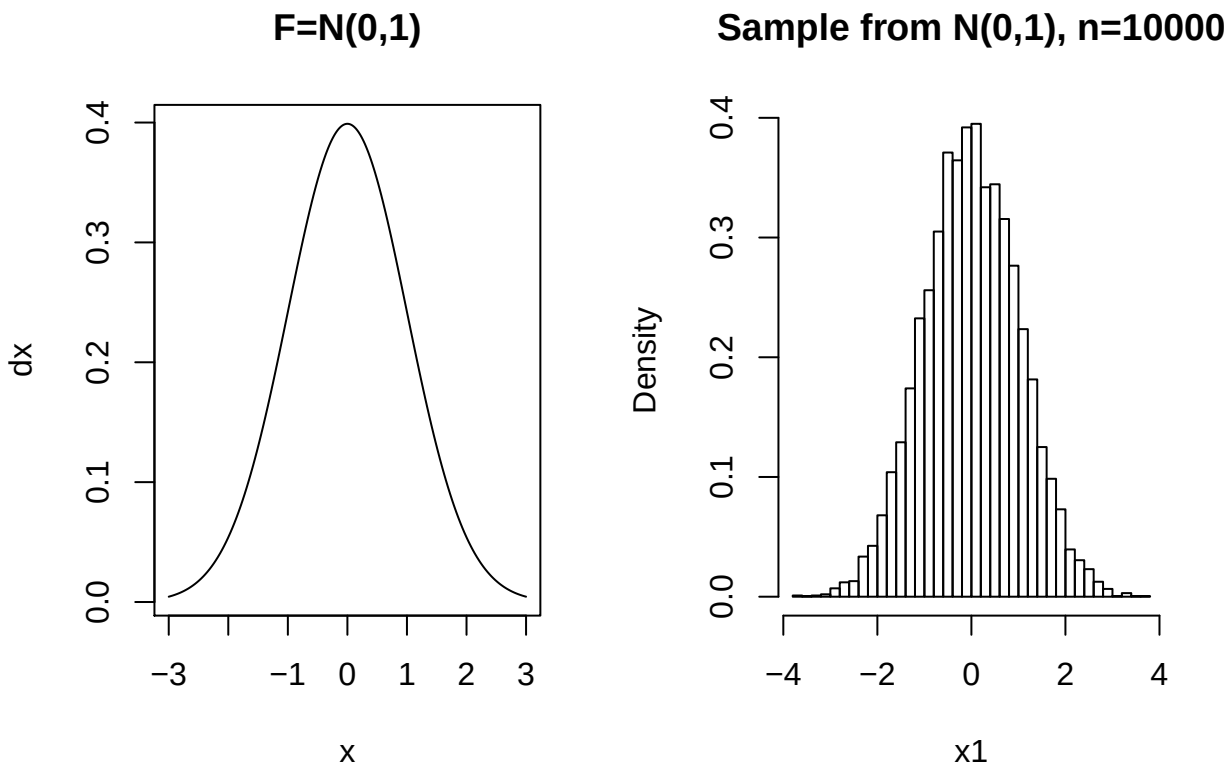


Figure 3: Left: density of N(0,1). Right: a random sample

Example: $F = N(0,1)$, $n = 50$

Consider a sample of 50 observations from $N(0,1)$. The lower panels Figure @ref(fig:figchp24a) shows the cumulative distribuion (F, left panel) while the right panel shows the emprical distribuion $\hat{F}$.

```
par(mfrow=c(2,2))
x2<-rnorm(50,0,1)
x2<-sort(x2)
px<-pnorm(x2,0,1)
dx<-dnorm(x,0,1)
plot(x,dx,type="l")
title("Density N(0,1)")
hist(x2,nclass=10,col=0,probability=T,main="Random sample: n=50")
plot(x2,px,type="l")
title("F")
n<-length(x2)
px.e<-c(1:length(x2))/n
plot(x2,px.e,type="s")
title("hat(F),n=50")
```

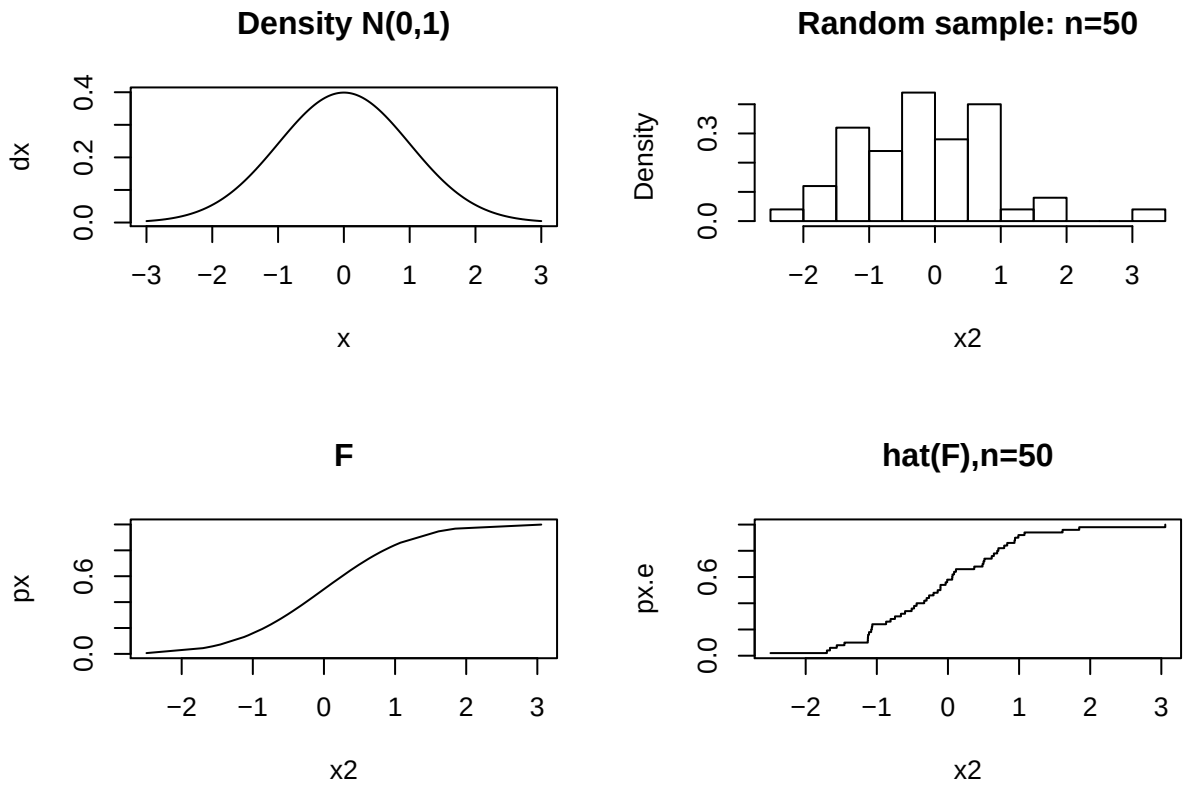


Figure 4: The emprical distribution

Example: $F = N(0,1)$, $n = 100$ **and** $n = 1000$

Examples for $F = N(0,1)$ and the empirical distributions for $n = 100$ and $n = 1000$ are shown in Figure @ref(fig:figchp25).

```
par(mfrow=c(1,2))
x2<-rnorm(100,0,1)
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",col=2)
title("n=100")
x2<-rnorm(1000,0,1)
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",col=2)
title("n=1000")
```

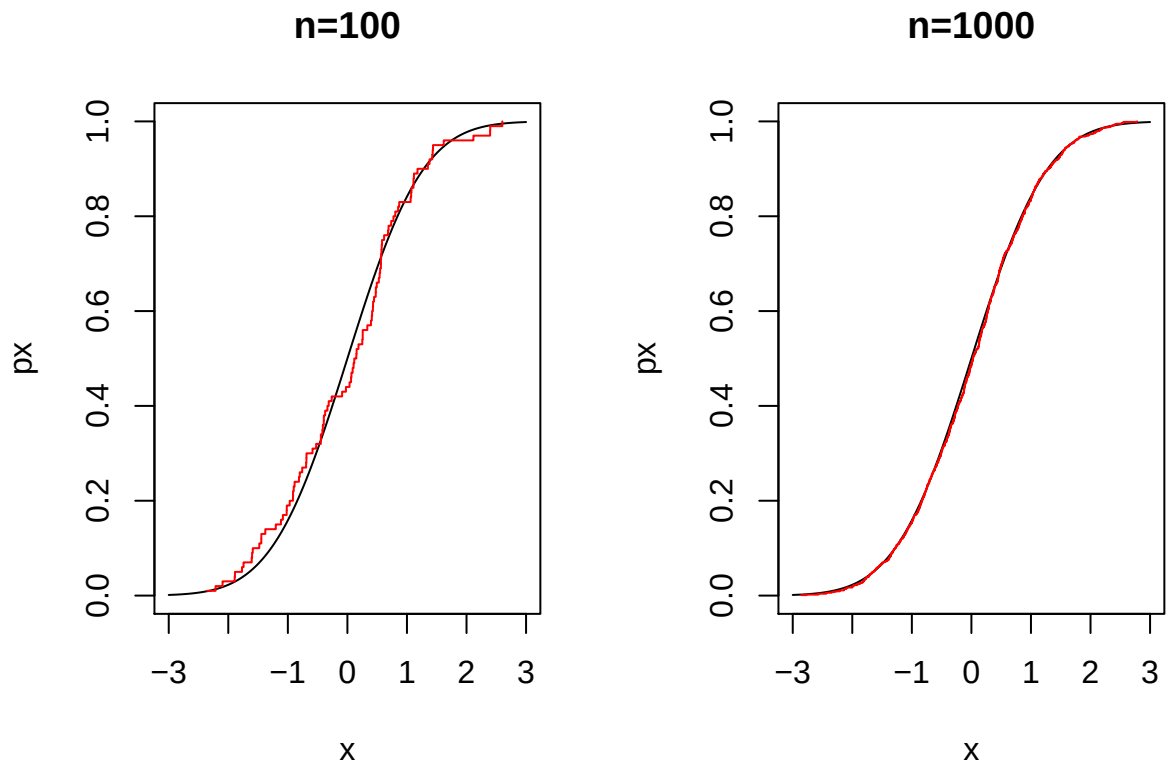


Figure 5: Emprical distribution with n=100 and n=1000

The accuracy of the sample mean

Let $x_1, \ldots, x_n$ is a random sample from a probability distribution F. Let us assume that θ is the population mean, σ^2 is the population variance and $\hat{\theta}$ is the sample mean. The variance of $\hat{\theta}$ is given by

$$Var(\hat{\theta}) = \frac{\sigma^2}{n},$$

and it can be estimated by

$$\frac{s^2}{n}, \quad s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 .$$

Here, $s^2 = \hat{\sigma}^2$.

Example: the mouse data

The mouse dataset is a small randomized experiment with 16 mice, 7 belong to treatment group and 9 belong to control group. Treatment was intended to prolong survival time after a test surgery. Survival times for the control group are given by

```
mouse.c
```

```
## [1]  52 104 146  10  50  31  40  27  46
```

```
mean(mouse.c)
```

```
## [1] 56.22222
```

and the survival times for the treatment group are

```
mouse.t
```

```
## [1]  94 197  16  38  99 141  23
```

The mean survival time is given by

```
mean(mouse.t)
```

```
## [1] 86.85714
```

The standard error of the sample mean is given by

$$\sqrt{\frac{\sigma^2}{n}}$$

For the mouse data the standard error for the two groups are given, respectivly, by

```
sqrt(var(mouse.c)/9)
```

```
## [1] 14.13897
```

```
sqrt(var(mouse.t)/7)
```

```
## [1] 25.23549
```

Example: a random samples from $N(0,1)$

Let us consider a random sample from $N(0,1)$ with $n = 100$ shown in Figure @ref(fig:figchp31),

```
x<-rnorm(100,0,1)
hist(x,nclass=50,probability=TRUE)
x1<-seq(from=-3,to=3,length=1000)
dx<-dnorm(x1,0,1)
lines(x1,dx)
```

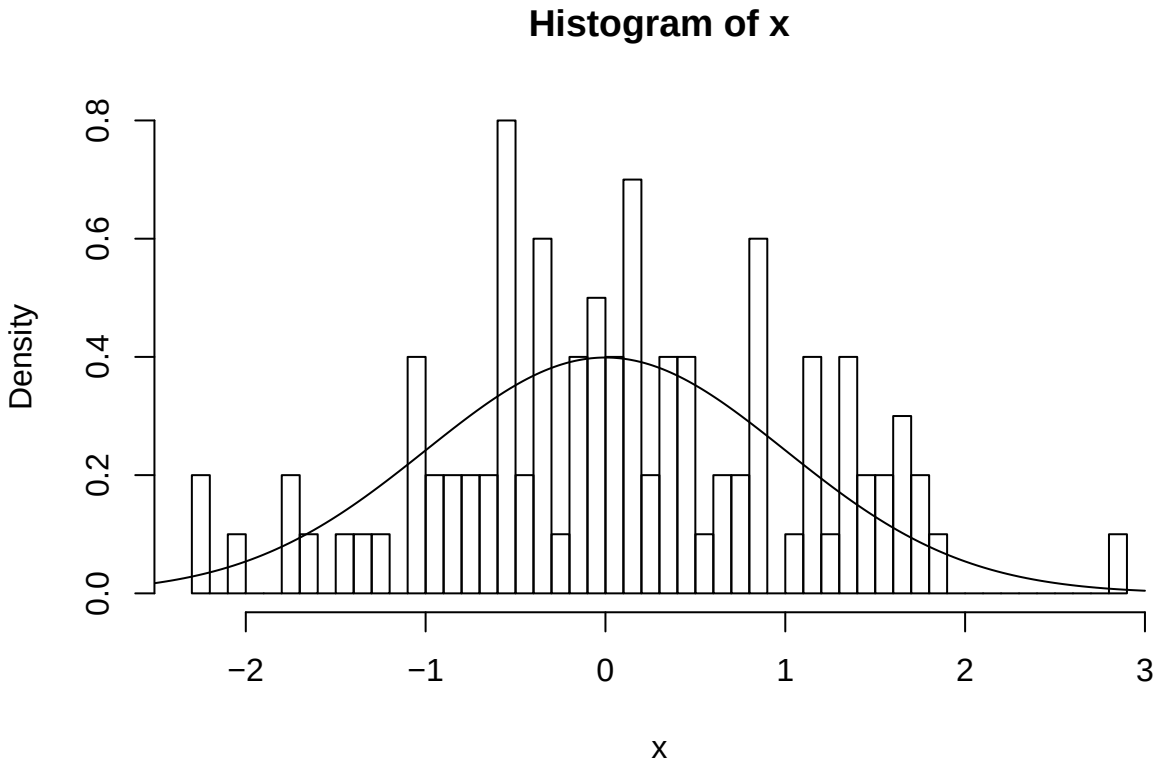


Figure 6: A random sample from N(0,1).

The sample mean is equal to

```
mean(x)
```

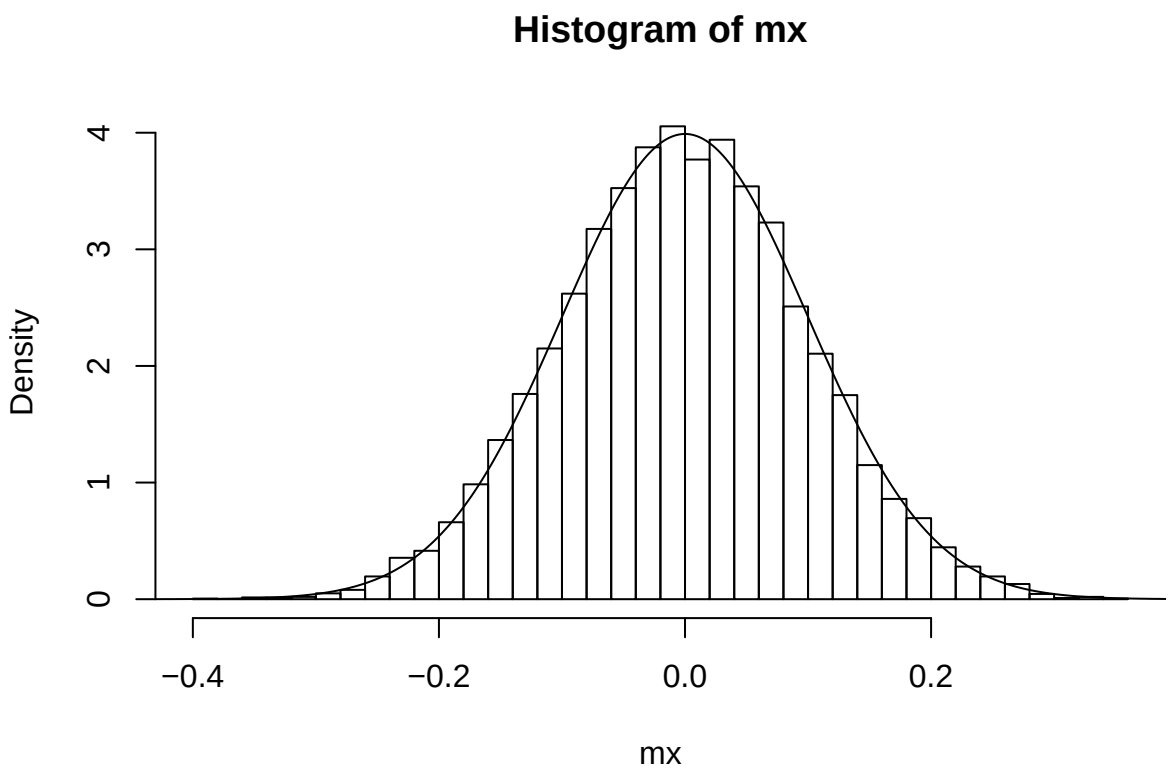
```
## [1] 0.09480468
```

Supose that we can resample 10000 samples of size 100 from $N(0,1)$ and estimate the mean with the sample mean $\bar{x}$.

```
mx<-c(1:10000)
for(i in 1:10000)
{
x<-rnorm(100,0,1)
mx[i]<-mean(x)
}
```

We now can estimate the distribution of $\bar{x}$ based on the estimated values for μ, $\bar{x}_1, \ldots, \bar{x}_{10000}$. The historgam for $\bar{x}$ and density plot for $N(\mu, \frac{\sigma^2}{n})$ are shown in Figure @ref(fig:figchp32)

```r
hist(mx,nclass=50,probability=TRUE)
x1<-seq(from=-1,to=1,length=1000)
dx<-dnorm(x1,0,sqrt(1/100))
lines(x1,dx)
```



Note that the standard error of $\bar{x}$, since $\sigma^2 = 1$ and $n = 100$ is equal to

$$\sqrt{\frac{1}{100}} = 0.1$$

For our example, the mean of $\bar{x}$ over the 10000 similated datasets is

```r
mean(mx)
```

```
## [1] -0.001268023
```

and the variance and standard error of the sample means are given, respectively, by

```r
c(var(mx),sqrt(var(mx)))
```

```
## [1] 0.009932229 0.099660568
```

very closed to the theoretical value of 0.01 (σ^2/n) and 0.1 ($\sqrt{\sigma^2/n}$) as expected.

Part II

The basic bootstrap

The bootstrap estimate of the standard error for $\hat{\theta}$

Introduction

The bootstap is a computer intensive method that can be used to

- Estimate variability of estimators.
- Estimate probabilities and quantiles related to test statistics or to construct confidence Intervals.
- explore the shape of the distribution of estimators or test statistics.

The basic idea behind the bootstrap is that, given a sample from $F(\theta)$, we repeatedly sample from the empirical distribution $\hat{F}$ in order to approximate the distribution of $\hat{\theta}$. In this chapter we focus on the estimation of the standard error of the sample mean, $\bar{x}$, and We discuss two bootstrap procedures:

- Parametric bootstrap.
- Non parametric bootstrap.

Although we mainly discuss the problem of estimation of the standard errors of sample mean, the method is developed to be fully automatic and can be applied for any estimator of interest./newline

Slides

Slides for the second part of the book that covers the topic of the basic bootstrap can be found here: Slides2.

YouTube tutorial: the sample function in R

The non prametric bootstrap procedures discussed in this book are based on a for loop in which the resampling is done using the sample() R function. A YouTube tutorial by Ian Dworkin is focused on the sample() function in R can be found here:YTBootstrap 2.

YouTube tutorial: the basic basic bootstrap

For a short online introduction by Dr. Sarveshwar Inani about the basic bootstrap procedures and the implementation in R see YTBootstrap 3.

Notaion

Let F be a probability distribution function with unknown parameter(s) θ. Let $x_1, \ldots, x_n$ be a random sample from F, $\theta = t(F)$ is the parameter(s) of primary interest and $\hat{\theta} = t(x_1, \ldots, x_n)$ is the plug-in estimate for θ. Our aim in this chapter is to estimate the standard error and the distribution of $\hat{\theta}$.

Non parametric bootstap

The Non parametric bootstrap algorithm consists of sampling for the empirical distribution $\hat{F}$. And it is defined as follows:

- Draw B bootstrap samples $x^*(b) = (x_1^*(b), \ldots, x_n^*(b))$, with replacement from (the empirical distribution) $x_1, \ldots, x_n$ $(b = 1, \ldots B)$.
- Evaluate the bootstrap replications

$$\hat{\theta}^*(b) = T(x^{*(b)}) \;\; b = 1, \ldots, B$$

- Estimate $se_F(\hat{\theta})$, $se_{\hat{F}_n}(\hat{\theta}^*)$ by the sample deviation of the B replications

$$\hat{se}_B = \sqrt{\frac{1}{B-1}\sum_{i=1}^{B}[\hat{\theta}^*(b) - \hat{\theta}^*(.)]^2}$$

where

$$\hat{\theta}^*(.) = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}^*(b).$$

Example: Estimation of the standard error of the sample mean using non parametric bootstrap

Consider a sample of 10 observations form an unknown distribution

```
x <- c(11.201, 10.035, 11.118, 9.055, 9.434, 9.663, 10.403, 11.662, 9.285,8.84)
```

With sample variance and standard error for $\bar{x}$ given by

```
var(x)
```

```
## [1] 0.9726152
```

```
var(x)/10
```

```
## [1] 0.09726152
```

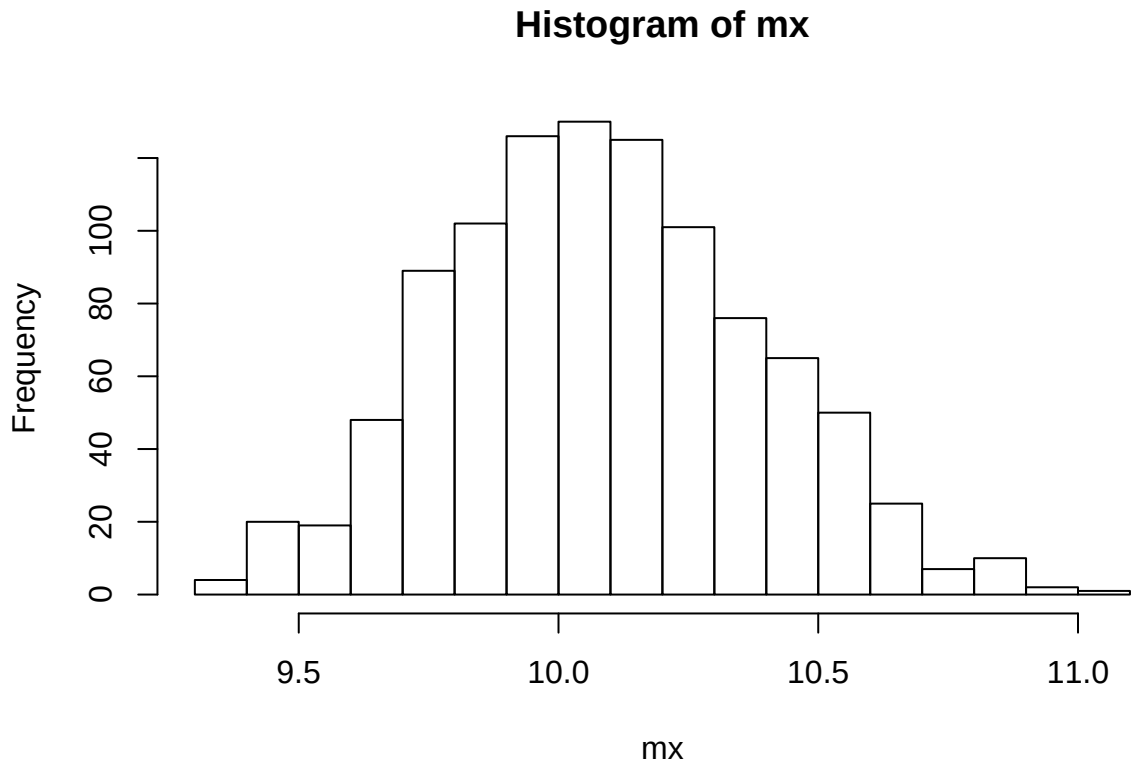
```
n<-length(x)
```

Non parametric bootstrap algorethim consists of resampling B bootstrap samples from $\hat{F}$. For the bth bootstrap sample (the bth setp in the "for loop") we use the function sample(x,n,replace=T)

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

Note that the R object mx is used to store the bootstrap replicates, mx=$(\hat{\theta}^*_1, \ldots, \hat{\theta}^*_B)$. A bootstrap estimate for the distribution of $\hat{\mu} = \bar{x}$ is shown in Figure @ref(fig:figchp41)

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



A bootstrap estimate for the standard error for $\bar{x}$:

```
var(mx)
```

compared with the classical estimate $\hat{\sigma}/\sqrt{n}$=0.09726152.

Parametric bootstap

Let $F(\theta)$ the probability function of interest. The parametric bootstrap estimate of the standard error of $\hat{\theta}$ is defined as

$$se_{\hat{F}_{prm}}(\hat{\theta}^*),$$

Here, $\hat{F}_{prm}$ is a parametric distribution that we used for $F(\theta)$. Note that since θ is an unknown parameter, we use the plag-in estimate $\hat{\theta}$. The parametric bootstrap algorithm is as follow:

- Make a parametric assumption about the probability distribution F.
- Estimate θ from the sample using the plug-in estimate $\hat{\theta}$.
- Draw B bootstrap samples (of size n) from $F(\hat{\theta})$.
- Evaluate the bootstrap replications

$$\hat{\theta}^*(b) = T(x^{*(b)}) \;\; b = 1, \ldots, B.$$

- Estimate $se_F(\hat{\theta})$, $se_{\hat{F}_n}(\hat{\theta}^*)$ by the sample deviation of the B replications

$$\hat{se}_B = \sqrt{\frac{1}{B-1}\sum_{i=1}^{B}[\hat{\theta}^*(b) - \hat{\theta}^*(.)]^2},$$

where

$$\hat{\theta}^*(.) = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}^*(b).$$

Example: Estimation of the standard error of the sample mean using parametric bootstrap

Let us assume that the sample $x_1, \ldots, x_n$ was drawn from $N(\mu, \sigma^2)$.Since μ and σ^2 are unknown we estimate both parameters using thier plug-in estimates,

```
MLx<-mean(x)
MLx
```

```
## [1] 10.0696
```

```
Varx<-var(x)
Varx
```

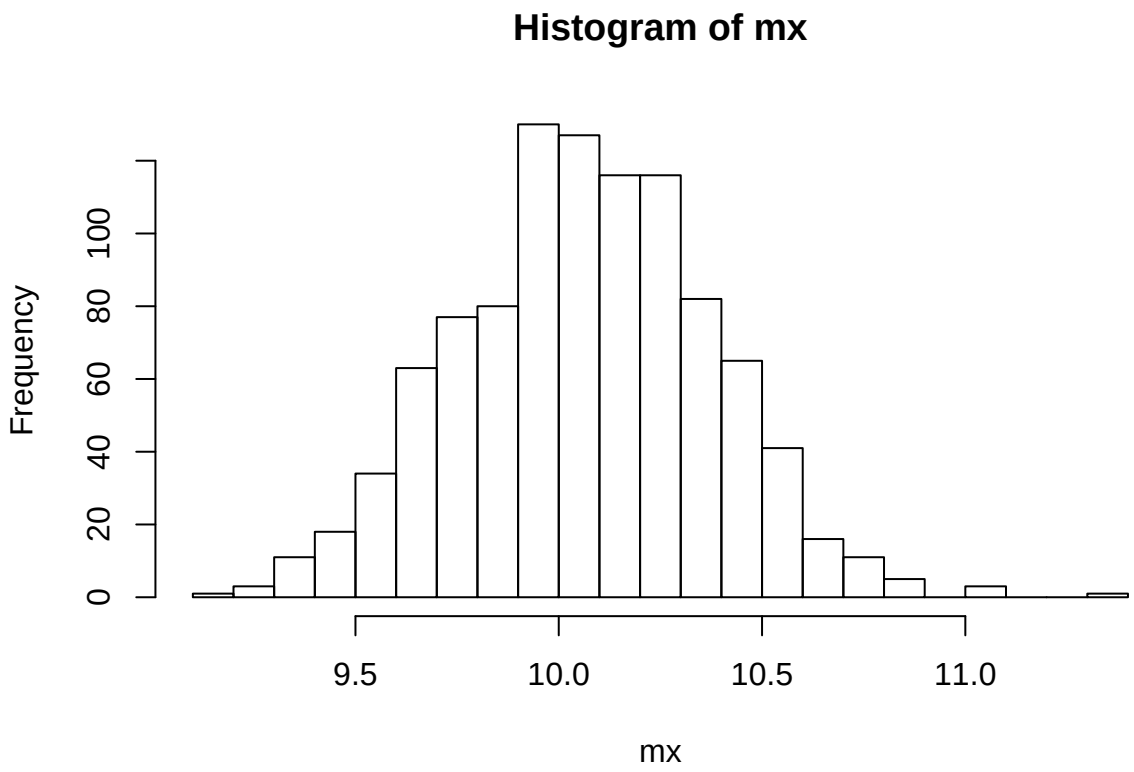
```
## [1] 0.9726152
```

Parametric bootstrap assuming $F = N(\mu, \sigma^2)$ and an empirical distribution $\hat{F} = N(\hat{\mu}, \hat{\sigma}^2)$ with 1000 bootstrap samples is given below, in each bootstrap iteration we use the function rnorm(n,MLx,sqrt(Varx)) to select a sample from $N(\hat{\mu}, \hat{\sigma}^2)$.

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

Figure @ref(fig:figchp42) shows the bootstrap estimate for the distribution of $\hat{mu} = \bar{x}$,

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



A bootstrap estimate for the standard error for $\bar{x}$:

```
var(mx)
```

```
## [1] 0.09655227
```

Example: the correlation coefficient.

YouTube tutorial: resampling pairs

The resampling procedure discussed in this section is required to resample pairs of observations (x_i, y_i) in order to keep the correlation stricture of the data. For a short online introduction, by Ian Dworkin, about the sample() function in R and how to resample pairs (start at the end of the tutorial at $5:45$), see YTBootstrap 3.

Estimating the distribution of the sample correlation with bootstrap

Consider the following data (Neter et al.~(1996), p. 653) on population (in thousands) and expenditures for a new food product (per capita in dollars)

```
#x:population (in thousands)
#y:expenditures
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)
```

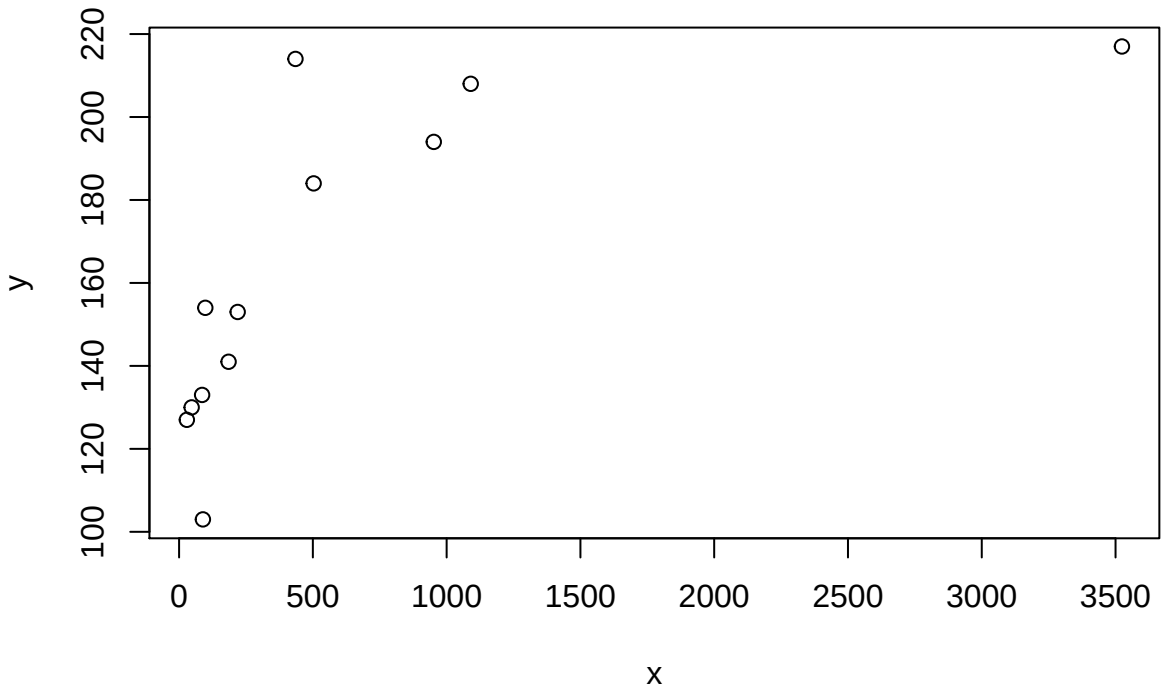
```
##          x   y
```

```
##  [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
```

```
cor.obs<-cor(x,y)
```

A scaterpot of the data is shown below.

```
plot(x,y)
```



The parameter of interest is the correation between the population and expenditures, $\rho(x, y)$. The sample correlation, $\hat{\rho}(x, y)$, is equal to

```
cor.obs
```

```
## [1] 0.6737664
```

Non parametric bootstrap

Our aim is to estimate the standard error and the distribution of the sample corelation. A non parametric bootstrap consists of resampling pairs from the bivariate data (x_i, y_i). Since we would like to keep the correlation structure as it is in the sample, we need to resample pairs. This implies that within the bootstrap loop the observation (x_i, y_i) will resample together. This can be done using an index vector:

```
n<-length(x)
n
```

```
## [1] 12
```

```
index<-c(1:n)
index
```

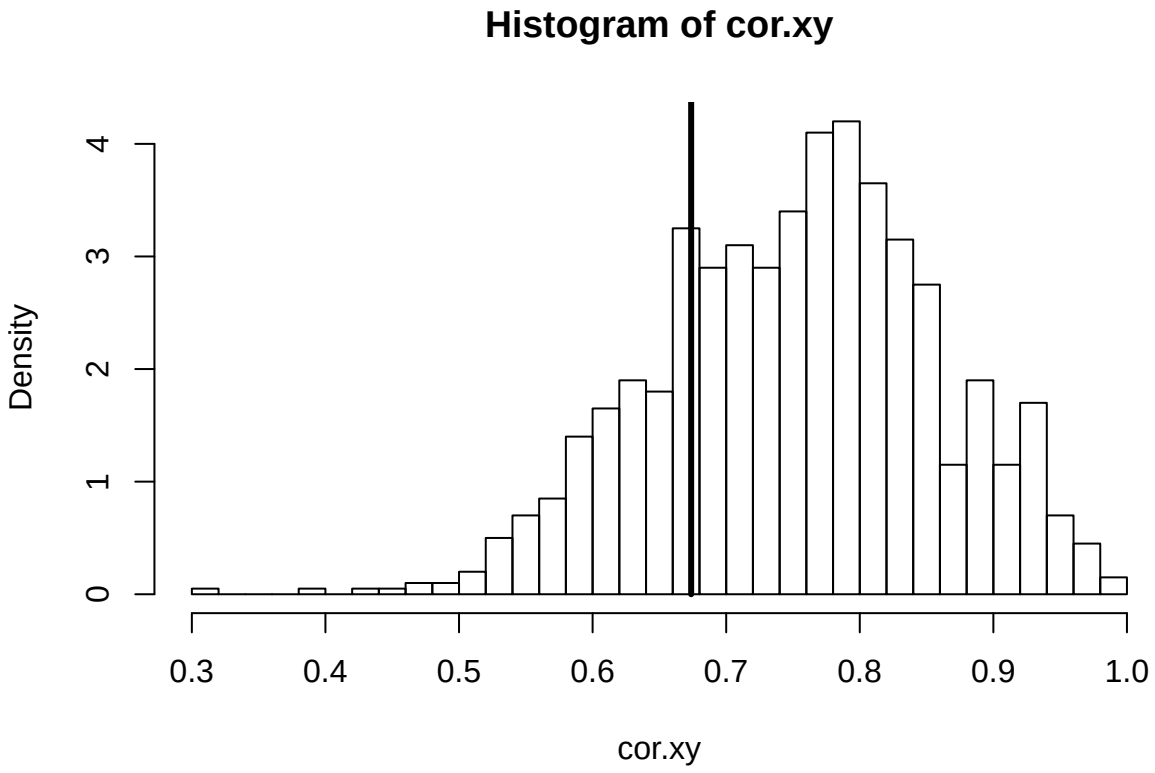
```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12
```

For the bootstrap we use the following code:

```
B<-1000
obs.8<-cor.xy<-c(1:B)
for(i in 1:B)
{
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)
}
```

Note that for each bootstrap sample, we first bootstarp the index vector, sample(index,n,replace=T), and use this bootstrap sample ro resample pairs. The distribution of the bootstrap rplicates is an estimate for the distribution of $\rho(\hat{x}, y)$ and shown in Figure @ref(fig:figchp43)

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



The standard error of $\hat{\rho}$ is given by

```
sqrt(var(cor.xy))
```

```
## [1] 0.1049373
```

Parametric bootstrap

For the parametric bootstrap we assume that the joint distribution of x and y is bivariate normal distribution given by

$$\begin{pmatrix} x \\ y \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_x \\ \mu y \end{pmatrix}, \Sigma\right).$$

Here, Σ is a 2×2 the covariance matrix given by

$$\Sigma = \begin{pmatrix} \sigma_x^2 & cov(x,y) \\ cov(x,y) & \sigma_y^2 \end{pmatrix}.$$

Since all the parameters are unknown we use their plug-in estimates $\hat{\mu}_x$ (MLx),$\hat{\mu}_y$ (MLy) and $\hat{\Sigma}$ (sigma).

```
cor.obs<-cor(x,y)
dat.mat<-cbind(x,y)
n<-length(x)
MLx<-mean(x)
MLy<-mean(y)
sigma<-cov(dat.mat)
```

The code for the parametric bootstrap is given below.

```
B<-1000
cor.xy<-c(1:B)
for(i in 1:B)
{
dat.b <- rmvnorm(n=n, mean=c(MLx,MLy),sigma=sigma)
cor.xy[i]<-cor(dat.b)[1,2]
}
```

Figure @ref(fig:figchp44) shows the distribution of the bootstrap replicates for $\hat{\rho}$.

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

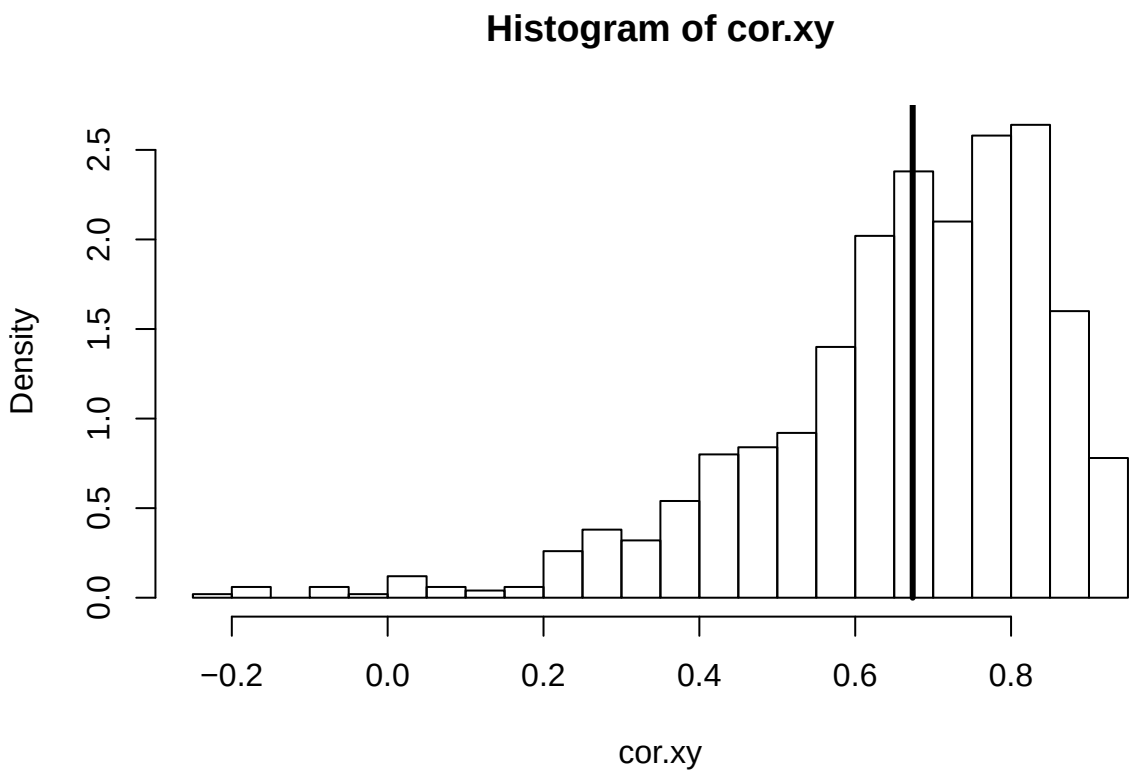


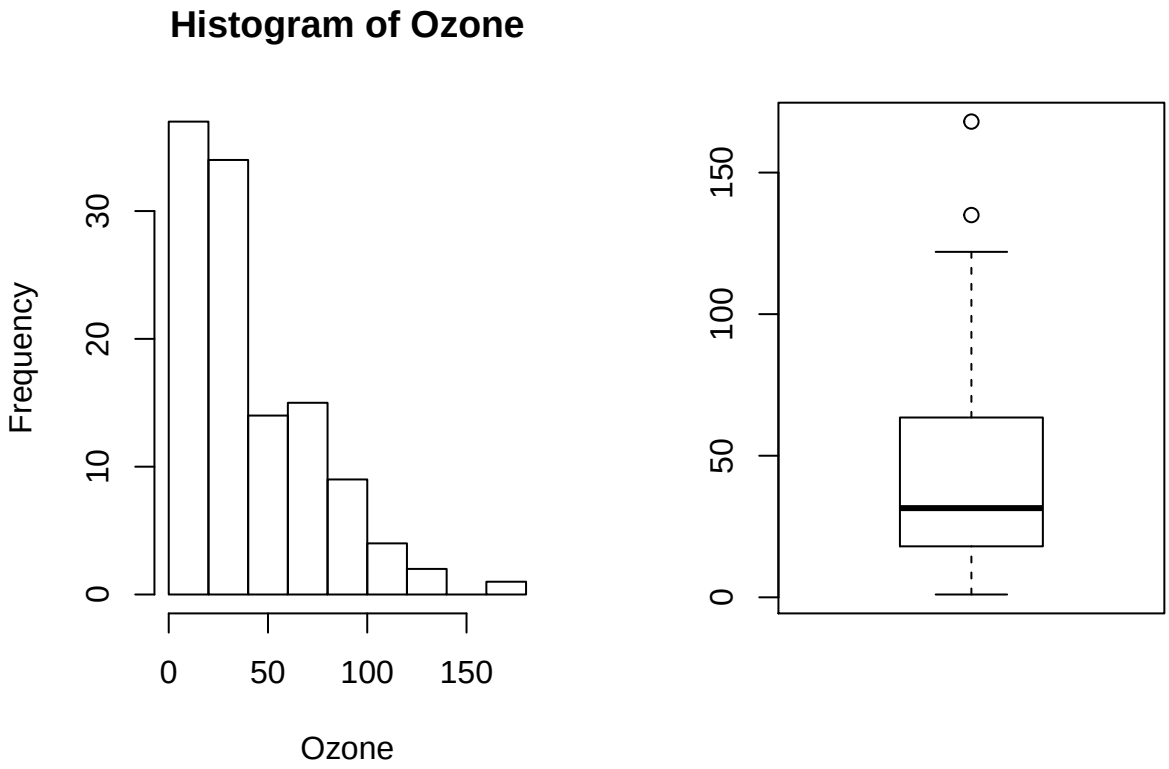
Figure 7: Parametric bootstrap: the distribution of the bootstrap replicates for the correlation.

Example: the airquality data.

The airquality data gives information about the daily air quality measurements in New York, May to September 1973. For the analysis presented below we focus on the mean ozone in parts per billion from 1300 to 1500 hours at Roosevelt Island. Histogram and boxplot for the Ozone are shown in Figure @ref(fig:figchp45)

```
par(mfrow=c(1,2))
attach(airquality)
Ozone<-na.omit(Ozone)
```

```
hist(Ozone)
boxplot(Ozone)
```



The parameters of primiry interest are the 25%, 50% and 75% quantiles of the ozone distribution. The sample quantiles are equal to

```
quantile(Ozone,probs=c(0.25,0.5,0.75))
```

```
##   25%   50%   75%
## 18.00 31.50 63.25
```

To calculate the standard error of the quantiles we implement a non parametruc bootstrap procedure. In each bootstrap step, we resample with replacment from the Ozone data using the function sample() in the follwoing way: sample(Ozone,size=116,replace=TRUE).

```
B<-10000
q.boot<-matrix(0,B,3)
for(b in 1:B)
{
Ozone.boot<-sample(Ozone,size=116,replace=TRUE)
q.boot[b,]<-quantile(Ozone.boot,probs=c(0.25,0.5,0.75))
}
```

Boxplot for the quantiles' distribution is shown in Figure @ref(fig:figchp46).

```
par(mfrow=c(2,2))
hist(q.boot[,1],nclass=50,main="q_25")
hist(q.boot[,2],nclass=50,main="q_50")
hist(q.boot[,3],nclass=50,main="q_75")
```

```
boxplot(q.boot[,1],q.boot[,2],q.boot[,3])
```

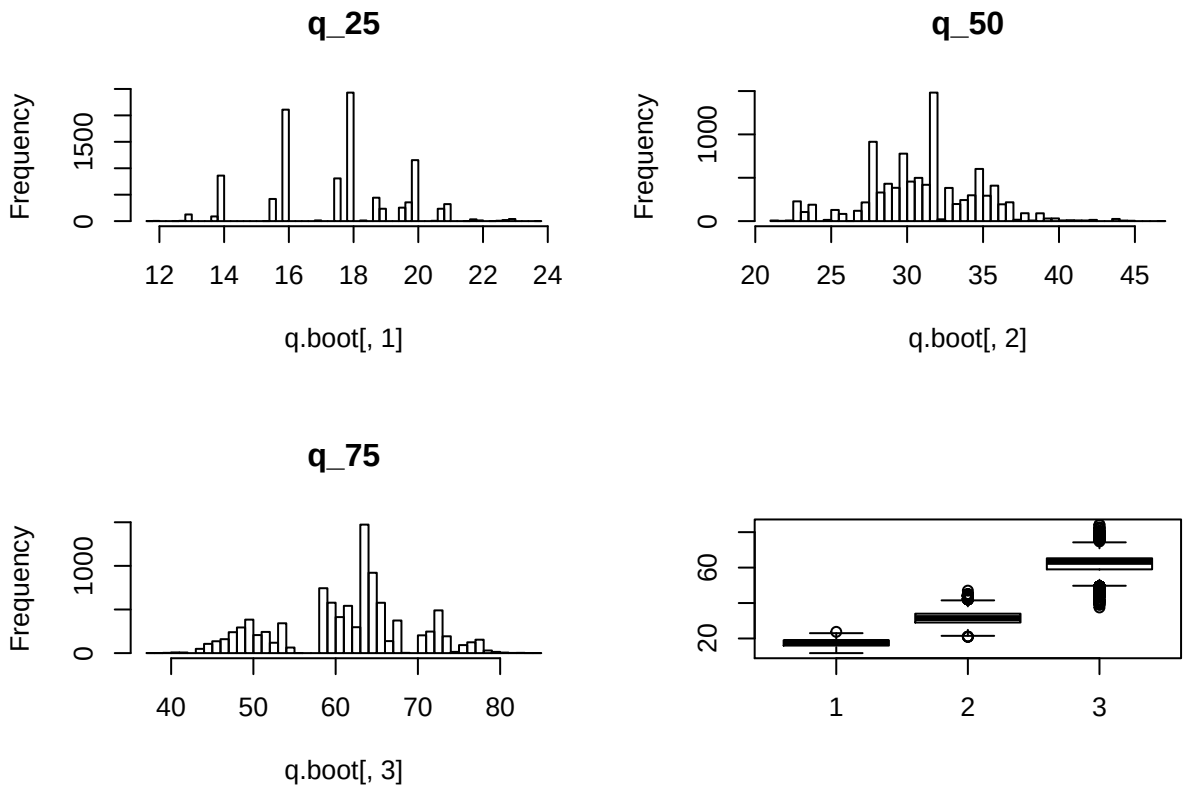


Figure 8: Bootstrap replicates for the quantiles of the airquality data.

Example: the score data.

The score dataset (the R object scor) provides information about 88 students who took examinations in 5 subjects. Some where with open book and other with closed book.The first 6 students and the boxplot for the scores are shown below

```
head(scor)
```

```
##   mec vec alg ana sta
## 1  77  82  67  67  81
## 2  63  78  80  70  81
## 3  75  73  71  66  81
## 4  55  72  63  70  68
## 5  63  63  65  70  63
## 6  53  61  72  64  73
```

```
boxplot(scor)
```

Our main interst is the covariance between the scores in statistics and algebra. The scatterplot matrix is shown in Figure @ref(fig:figchp48)

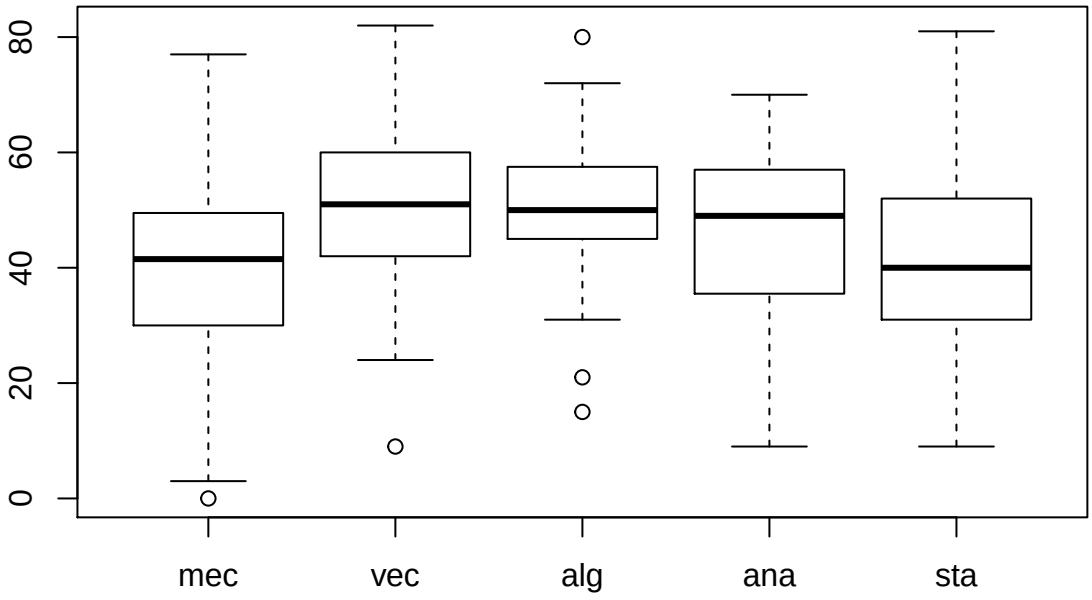


Figure 9: Boxplot for the score data.

```
pairs(scor)
```

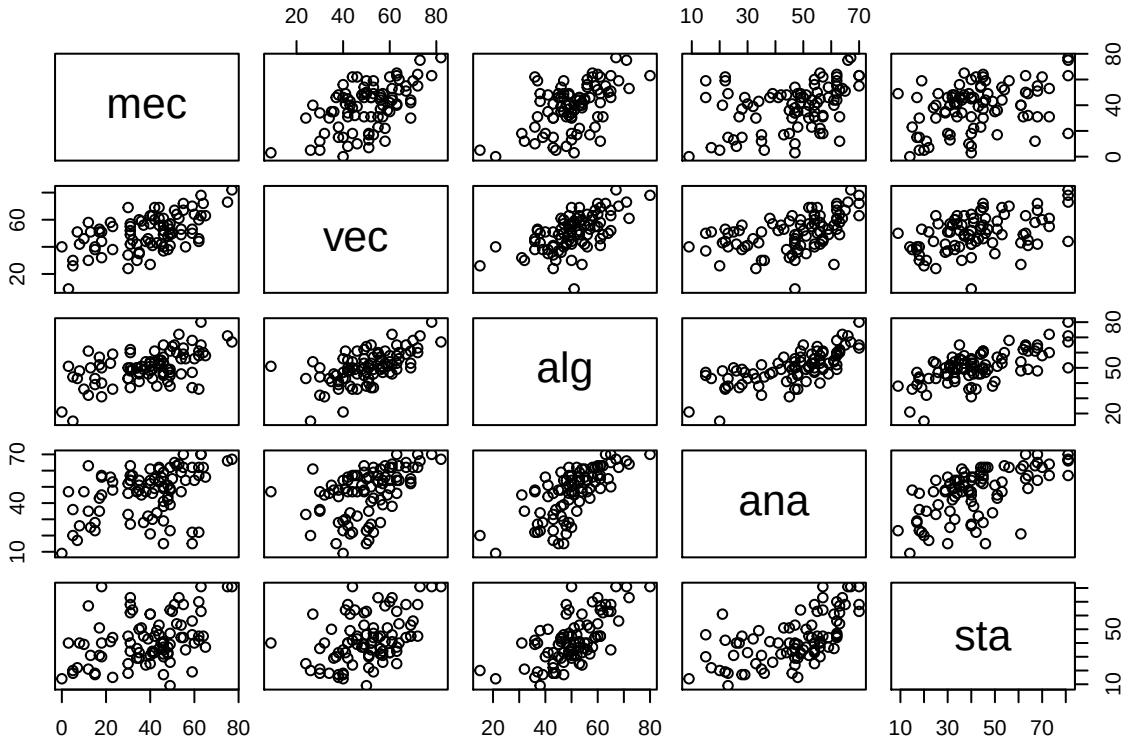


Figure 10: Scatterplot matrix for the score data.

The sample covariance matrix

```
cov(scor)
```

```
##          mec       vec       alg       ana       sta
## mec 305.7680 127.22257 101.57941 106.27273 117.40491
## vec 127.2226 172.84222  85.15726  94.67294  99.01202
## alg 101.5794  85.15726 112.88597 112.11338 121.87056
## ana 106.2727  94.67294 112.11338 220.38036 155.53553
## sta 117.4049  99.01202 121.87056 155.53553 297.75536
```

The sample covariance between statistics and algebra is equal to

```
cov(scor)[3,5]
```

```
## [1] 121.8706
```

To estimate the distribution and the standard error of $\Sigma_{3,5}$ we conduct a non parametric bootstrap of 1000 steps. Note that the R object sig35 is a vector contains the bootstrap replicates for $\Sigma^*_{3,5}$.

```
n<-length(scor$sta)
B<-1000
index<-c(1:n)
sig35<-ratio.b<-c(1:B)
```

```
for(i in 1:B)
{
index.b<-sample(index,n,replace=TRUE)
scor.b<-scor[index.b,]
cov.b<-cov(scor.b)
sig35[i]<-cov.b[5,3]
ratio.b[i]<-mean(scor.b$sta/scor.b$alg)
}
```

The distribtuion of the bootstarp replicates for $\hat{\Sigma}_{3,5}$ is presented in Figure @ref(fig:figchp49).

```
hist(sig35,nclass=50)
lines(c( 121.8706,121.8706),c(0,500),col=2,lwd=3)
```

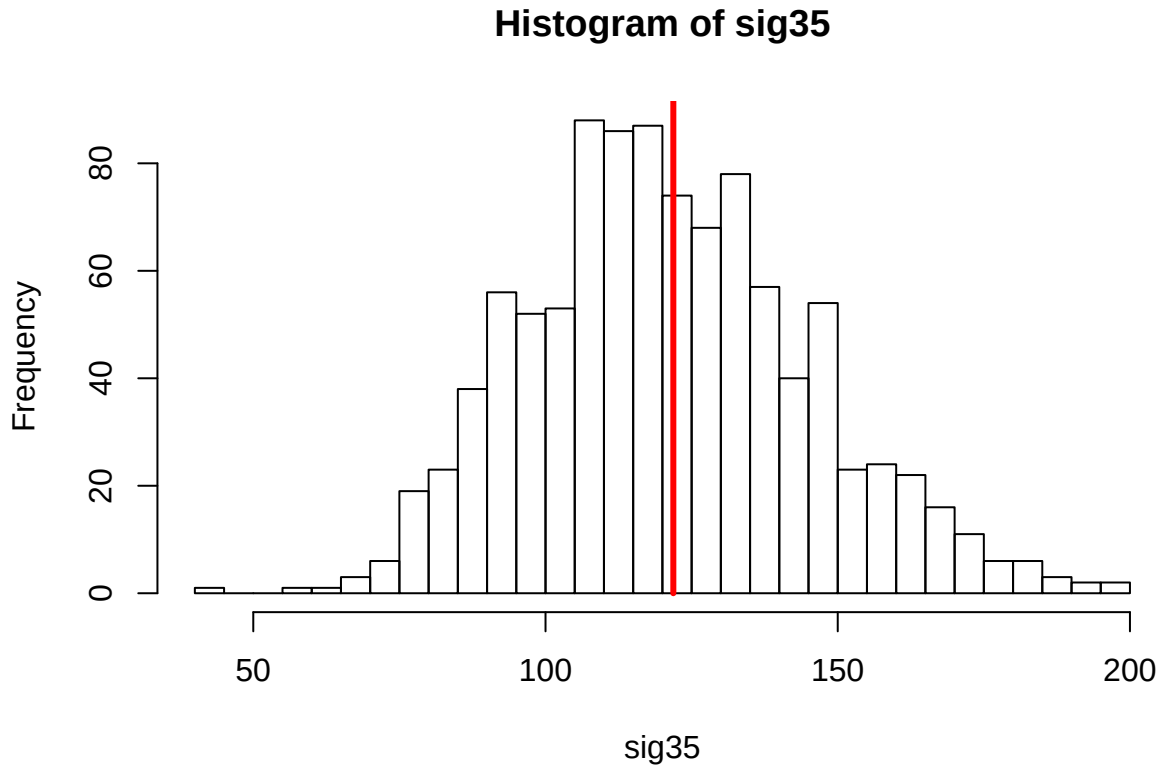


Figure 11: The distribtuion of the bootstarp replicates for the covariance.

Standard error for $\hat{\Sigma}_{3,5}$ is equal to

```
var(sig35)
```

```
## [1] 580.6884
```

```
sqrt(var(sig35))
```

```
## [1] 24.09748
```

Part III

Bootstrap confidence intervals

Confidence intervals

Introduction

In this part of the book we discuss confidence intervals for population parameters. We discuss several topics include

- Classical confidence intervals.
- Improved bootstrap normal confidence intervals.
- Percentile intervals.
- Bootstrap t intervals.
- The transformation respecting properties of a confidence interval.
- The BCa method.

Slides

Slides for the third part of the book that covers the topic of the bootstrap confidence intervals can be found here: Slides3.

YouTube tutorial: bootstrap confidence intervals

A very basic YouTube tutorial by Matthew E. Clapham is focused on a non parametric bootstrap procedure in R for the calculation of confidence intervals for the mean. See YTBootstrap 4.

Classical confidence intervals

Let $\hat{\theta}$ be an estimator for the parameter θ (based on a sample of size n) and suppose we want an interval with left and right tail errors equal to α. Let the respective quantiles of $\hat{\theta} - \theta$ be defined as

$$P(\hat{\theta} - \theta \leq \xi_\alpha) = \alpha = P(\hat{\theta} - \theta \geq \xi_{1-\alpha}).$$

Then the $(1 - 2\alpha)100\%$ equi-tailed interval is given by

$$[\hat{\theta} - \xi_{1-\alpha}, \hat{\theta} - \xi_\alpha].$$

Usually the distribution and hence the quantiles of $\hat{\theta} - \theta$ are unknown and have to be estimated using the bootstrap. As in many situations, both the parametric and the nonparametric bootstrap can be used for these purposes.

First we consider a special case where the distribution of $\hat{\theta} - \theta$ is assumed to be (approximately) normal.

Example: the mouse data - classical C.I using $N(0, 1)$

We consider the treatment group of the mouse data for which $\bar{z} = 86.85714$.

```
z <- c(94, 197, 16, 38, 99, 141, 23)
z
```

```
## [1]  94 197  16  38  99 141  23
```

```
t.hat<-mean(z)
se.t.hat<-sqrt(var(z)/7)
t.hat
```

```
## [1] 86.85714
```

The standard error of the sample mean is equal to

```
se.t.hat
```

```
## [1] 25.23549
```

Let us assume that $Z_i \sim N(\mu, \sigma^2)$. In this case the $(1-\alpha) \times 100\%$ confidence intervals is given by

$$(\hat{(}\theta) - \frac{\sigma}{\sqrt{n}} \times C_\alpha, \hat{(}\theta) + \frac{\sigma}{\sqrt{n}} \times C_\alpha).$$

Here, C_α is the quantile for the $N(0,1)$ distribution that satisfies

$$P(Z \leq C_\alpha) = 1 - \frac{\alpha}{2}.$$

For $\alpha = 0.1$, the critical value (the 95% quantile of $N(0,1)$) is equal to

```
C.alpha<-qnorm(0.95,0,1)
C.alpha
```

```
## [1] 1.644854
```

and it is shown in the Figure @ref(fig:figchp51).

```
x<-seq(from=-3,to=3,length=1000)
dx<-dnorm(x,0,1)
plot(x,dx,type="l")
lines(c(C.alpha,C.alpha),c(0,1),col=2)
```

Thus, a 90% C.I for the mean is given by

```
c(t.hat-se.t.hat*C.alpha,t.hat+se.t.hat*C.alpha)
```

```
## [1]  45.34846 128.36583
```

Example: the mouse data - classical C.I using t distribution

Alternativly, we can use a t_6 to choose the quantile for the C.I. In this case a 90% C.I for the mean (37.82,135.89) is given as a part of output below

```
t.test(z,conf.level=0.9)
```

```
##
##  One Sample t-test
##
## data:  z
## t = 3.4419, df = 6, p-value = 0.01377
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##   37.82004 135.89425
## sample estimates:
## mean of x
```

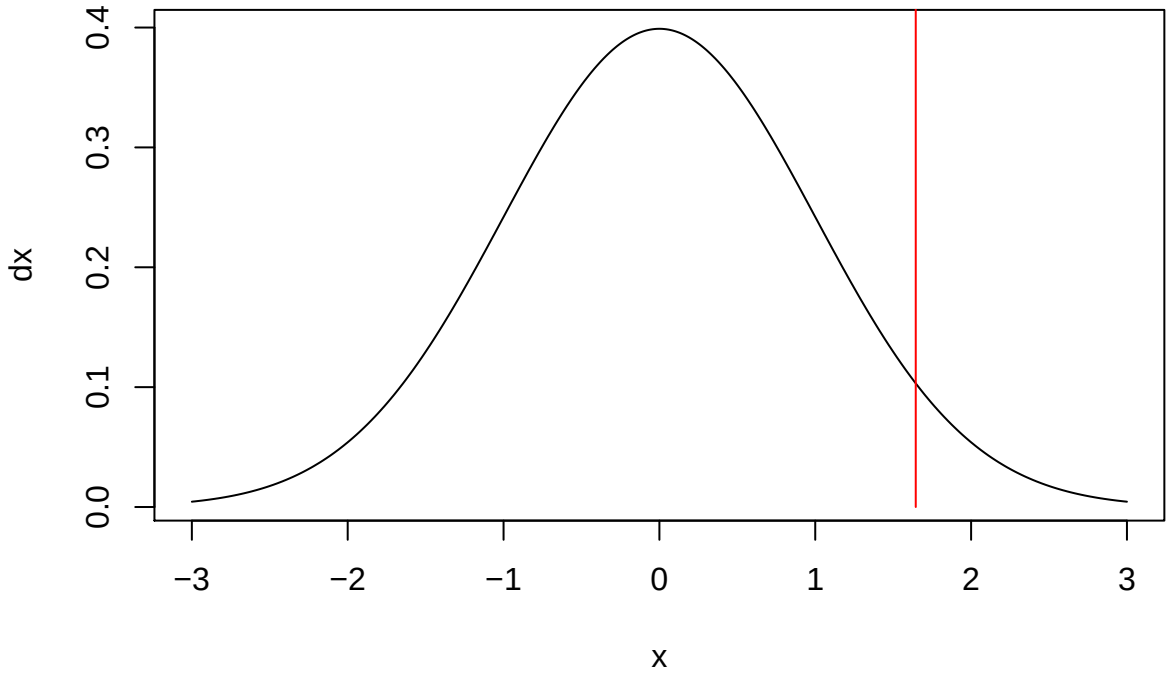


Figure 12: Critical value from N(0,1).

```
##  86.85714
```

Bootstrap confidence intervals

The main idea behind the bootstrap confidence intervals is to avoid the parametric assumption about the distribution of $\hat{\theta} - \theta$ and to estimate the distribution using bootstrap techniques. Both the parametric and non-parametric bootstrap procedures, discussed Chapter 4 of the book, can be used.

Improved normal confidence intervals

Suppose $\hat{\theta} - \theta$ is (approximately) normal, so

$$\hat{\theta} \sim N(\beta, \nu),$$

where β represents the (asymptotic) bias

$$\beta = E(\hat{\theta}) - \theta,$$

and ν the (asymptotic) variance $\nu = Var(\hat{\theta})$. In case β and ν are known, a $(1-2\alpha)100\%$ confidence interval for θ is given by

$$(\hat{\theta} - \beta - C_\alpha\sqrt{\nu}, \hat{\theta} - \beta - C_\alpha\sqrt{\nu}),$$

where $\Phi(C_\alpha) = 1 - \alpha$.

In case that the bias and variance are unknown, conditional on the observed sample, both the bias and variance of $\hat{\theta}$ can be estimated using the actual value of the parameter estimate and a set of bootstrap replicates $\hat{\theta}_b^*, b = 1, \ldots, B$

$$\hat{\beta} = \bar{\hat{\theta}}^* - \hat{\theta} = B^{-1}\sum_{b=1}^{B}\hat{\theta}_b^* - \hat{\theta}.$$

This estimate will be discussed further in Chapter 12. The standard error can be estimated by

$$\hat{\nu} = (B-1)^{-1}\sum_{b=1}^{B}(\hat{\theta}_b^* - \bar{\theta}^*)^2,$$

leading to the confidence interval

$$(\hat{\theta} - \hat{\beta} + C_\alpha\sqrt{\hat{\nu}}, \hat{\theta} - \hat{\beta} - C_\alpha\sqrt{\hat{\nu}}).$$

Example: the mouse data

In the first stage we implement a non parametric bootstrap,

```
B=10000
t.boot<-c(1:B)
for(b in 1:B)
{
x.boot<-sample(z,size=length(z),replace=T)
t.boot[b]<-mean(x.boot)
}
```

The R object t.boot is use to store the bootstrap replicates. The bias estimate is calculated by

```
bias.t<-mean(t.boot)-t.hat
bias.t
```

```
## [1] 0.1620429
```

The distribution of the bootstrap replicates is shown in Figure @ref(fig:figchp52)

```
hist(t.boot,nclass=20,col=0,probability=TRUE)
lines(c(t.hat,t.hat),c(0,20),col=2)
lines(c(mean(t.boot),mean(t.boot)),c(0,20),col=4)
```

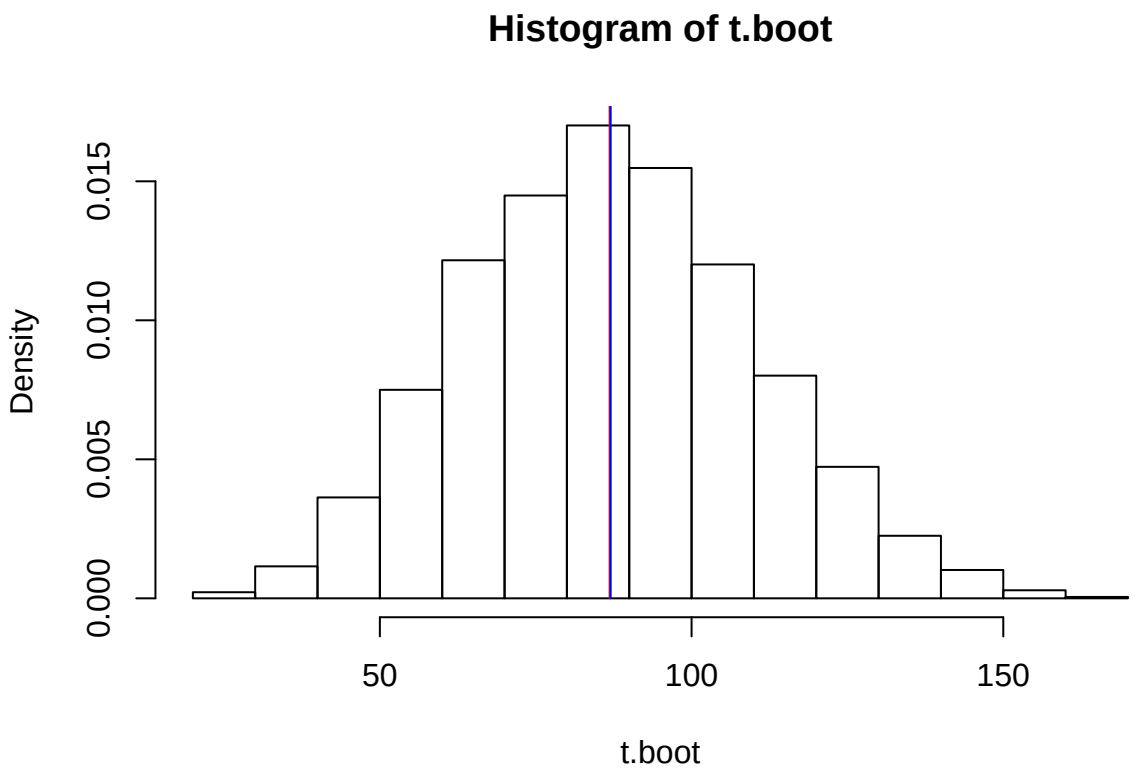


Figure 13: Distribution of the bootstrap replicates.

The bootstrap estimate for the standard error of the sample mean is equal to

```
vt<-sqrt(var(t.boot))
vt
```

```
## [1] 23.12449
```

For a 90% C.I the critical value of $N(0,1)$ is give by

```
C.alpha<-qnorm(0.95,0,1)
C.alpha
```

```
## [1] 1.644854
```

leading to a 90% confidence interval of

```
c(t.hat-bias.t-se.t.hat*C.alpha,t.hat-bias.t+se.t.hat*C.alpha)
```

```
## [1]  45.18641 128.20379
```

Note that the classical C.I, calculated in the previous section is (45.34846,128.36583).

Basic bootstrap intervals

Without any distributional assumptions on $\hat{\theta}$, basic bootstrap intervals can be based on bootstrap estimates for the quantiles $\xi_{1-\alpha}$ and ξ_{α}. We can use the corresponding quantiles of the distribution of the bootstrap replicates.The bootstrap C.I in this case is given by

$$(2\hat{\theta} - \hat{\theta}^*_{((B+1)(1-\alpha))}, 2\hat{\theta} - \hat{\theta}^*_{((B+1)(\alpha))}),$$

where $\hat{\theta}^*_{(b)}, k = 1, \ldots, B$ denote the ordered bootstrap replicates. Here and in the sequel, we assume that B and α are chosen such that $(B+1)\alpha$ and $(B+1)(1-\alpha)$ are integers.

Example: the mouse data

A non parametric bootstrap procedure for the mouse data can be implemented as follow

```
B=10000
t.boot<-c(1:B)
for(b in 1:B)
{
x.boot<-sample(z,size=length(z),replace=T)
t.boot[b]<-mean(x.boot)
}
```

For a 90% C.I, the 95% and 5% quantiles of the bootstrap replicates are given, respectively, by

```
t.up<-quantile(t.boot,probs=c(0.95))
t.lo<-quantile(t.boot,probs=c(0.05))
c(t.up,t.lo)
```

```
##       95%        5%
## 126.42857  49.14286
```

The upper and lower quantiles are shown in Figure @ref(fig:figchp53).

```
hist(t.boot,nclass=20,col=0,probability=TRUE)
lines(c(t.up,t.up),c(0,20),col=2)
lines(c(t.lo,t.lo),c(0,20),col=2)
```

A 90% C.I:

```
c(2*t.hat-t.up,2*t.hat-t.lo)
```

```
##       95%        5%
##  47.28571 124.57143
```

Percentile intervals

The bootstrap percentile C.I is based on the distribution of the bootstrap replicates for $\hat{\theta}$ and it is equal to

$$[\theta^*_{((B+1)\alpha)}, \theta^*_{((B+1)(1-\alpha))}].$$

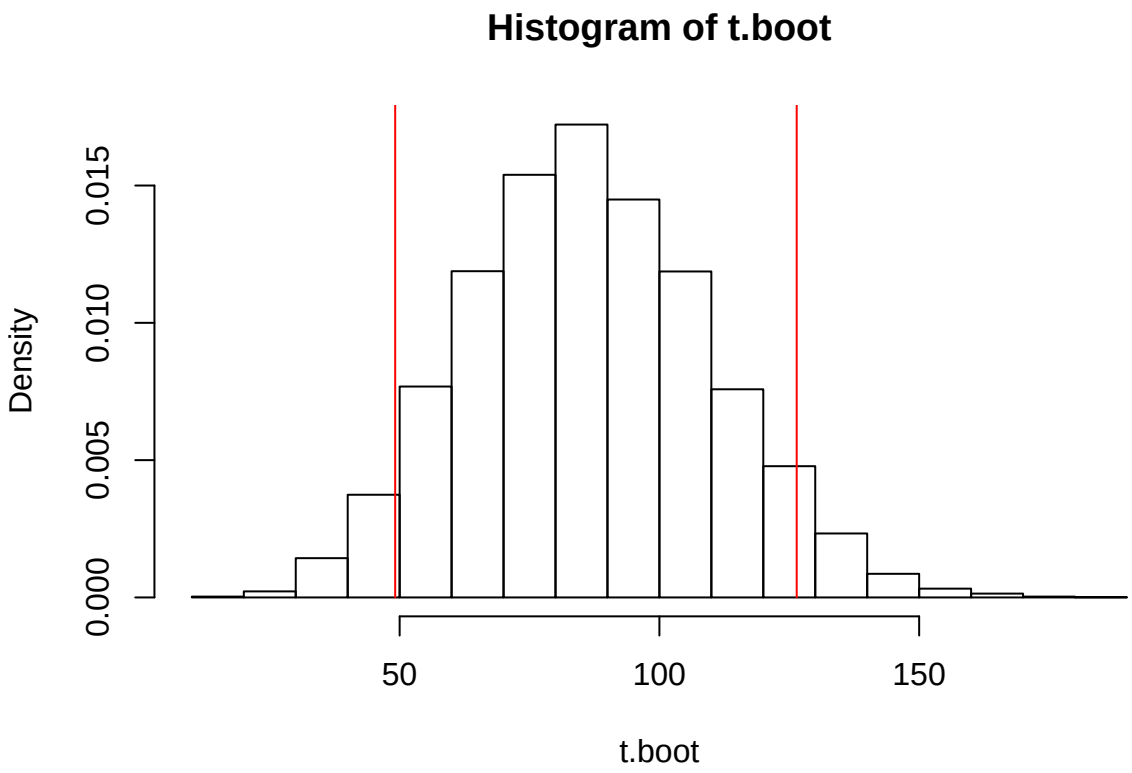


Figure 14: Bootstrap replicated and C.I.

The percentile C.I is a transformation respecting interval. Suppose that there is some (monotonic & nondecreasing) transformation $\phi = m(\theta)$ which has a symmetric distribution. Using this symmetry, the $(1-2\alpha)100\%$ basic confidence interval for $\phi = m(\theta)$ is given is by

$$[m(\theta^*_{((B+1)\alpha)}), m(\theta^*_{((B+1)(1-\alpha))})].$$

This type of bootstrap interval was initially recommended in place of the basic bootstrap interval, but the method turned out not to work well for the nonparametric bootstrap (even when a suitable transformation m exists). Adjusted percentile methods overcoming these difficulties are discussed in Section 5.2.8.

Example: percentile intervals for the mouse data

We use a non parametric bootstrap with $B = 10000$ to approximate the distribution of $\hat{\theta}$.

```
B=10000
t.boot<-c(1:B)
for(b in 1:B)
{
x.boot<-sample(z,size=length(z),replace=T)
t.boot[b]<-mean(x.boot)
}
```

A 90% percentile interval is given by

```
quantile(t.boot,probs=c(0.05,0.95))
```

```
##        5%       95%
##  49.14286 126.44286
```

This interval is visualized in Figure @ref(fig:figchp54)

```
lo<-quantile(t.boot,probs=c(0.05,0.95))[1]
up<-quantile(t.boot,probs=c(0.05,0.95))[2]
hist(t.boot,probability=T,nclass=50)
lines(c(lo,lo),c(0,0.2),col=4)
lines(c(up,up),c(0,0.2),col=4)
```

Bootstrap t intervals

The bootstrap t interval is based on the distribution of a pivotal statistic such as

$$Z = \frac{\hat{\theta} - \theta}{\sqrt{V}},$$

where V is an estimator for $\nu = Var(\hat{\theta})$.

The distribution the bootstap replicates

$$Z^* = \frac{\hat{\theta}^* - \theta}{\sqrt{V^*}},$$

can then be used to estimate the quantiles of Z. For B bootstrap samples, this leads to the interval

$$(\theta - \sqrt{v}z^*_{((1-\alpha)(B+1))}, \theta - \sqrt{v}z^*_{(\alpha(B+1))}).$$

These intervals are known as "Studentized intervals". The method is superior to the previous ones. But it needs a variance estimator and does not respect transfrmation of the form $\phi = m(\theta)$.

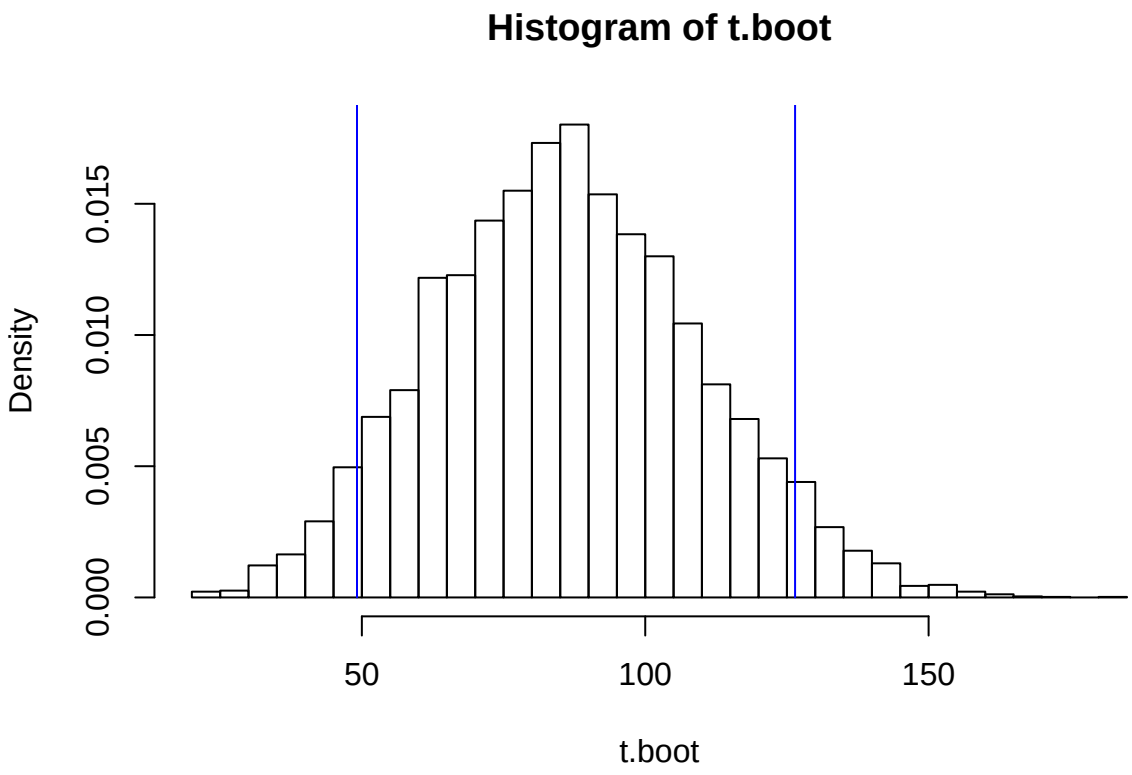


Figure 15: Precentile interval.

Example: the mouse data

Our starting point is a non parametric bootstrap with B=10000.

```
B=10000
t.boot<-c(1:B)
for(b in 1:B)
{
x.boot<-sample(z,size=length(z),replace=T)
se.boot<-sqrt(var(x.boot)/length(z))
t.boot[b]<-(mean(x.boot)-t.hat)/se.boot
}
```

Quantiles for bootstrap replicates z^* and for $t_{(6)}$ (note that $n = 7$):

```
quantile(t.boot,probs=c(0.05,0.95))
```

```
##        5%       95%
## -2.193178  1.682441
```

```
qt(0.95,6)
```

```
## [1] 1.94318
```

```
qt(0.05,6)
```

```
## [1] -1.94318
```

Histogram of the bootstrap replicates z^* with density of $t_{(6)}$ and the quantiles is presnted in Figure @ref(fig:figchp56)

```
hist(t.boot,probability=T,nclass=100,ylim=c(0,0.5),xlim=c(-10,10))
xx<-seq(from=-3,to=3,length=1000)
dx2<-dt(xx,6)
lines(xx,dx2,col=2)
lines(rep(qt(0.95,6),2),c(0,0.4),col=2)
calpha<-quantile(t.boot,probs=c(0.95))
lines(rep(calpha,2),c(0,0.4),col=4)
```

```
qt(0.95,6)
```

```
## [1] 1.94318
```

```
quantile(t.boot,probs=c(0.95))
```

```
##      95%
## 1.682441
```

The bootstrap t interval:

```
up<-quantile(t.boot,probs=c(0.95))
lo<-quantile(t.boot,probs=c(0.05))
c(t.hat+se.t.hat*lo,t.hat+se.t.hat*up)
```

```
##        5%       95%
##  31.51123 129.31436
```

Confidence interval based on $t_{(6)}$ distribution:

```
t.test(z,conf.level=0.9)
```

```
##
```

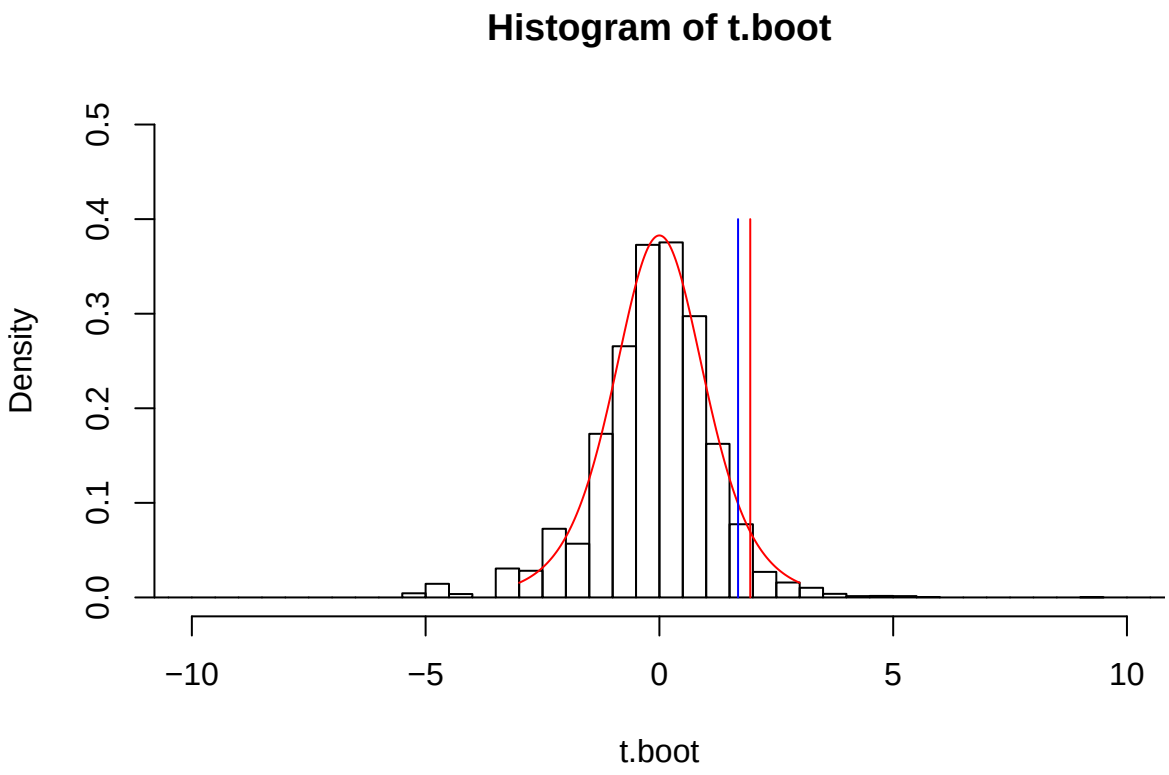


Figure 16: Bootstrap replicates for z.

```
##  One Sample t-test
##
## data:  z
## t = 3.4419, df = 6, p-value = 0.01377
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##   37.82004 135.89425
## sample estimates:
## mean of x
##  86.85714
```

The BCa method

The BC_a interval is a bootstrap interval based on a percentile interval which is corrected for possible bias. Here, the choices of α_1 and α_2 (for the quantiles to be used in the interval) are determined by

$$\alpha_1 = \Phi\left(\hat{z}_0 + \frac{\hat{z}_0 + z_\alpha}{1 - \hat{a}(\hat{z}_0 + z_\alpha)}\right),$$

and

$$\alpha_2 = \Phi\left(\hat{z}_0 + \frac{\hat{z}_0 + z_{1-\alpha}}{1 - \hat{a}(\hat{z}_0 + z_{1-\alpha})}\right),$$

where

$$\hat{z}_0 = \Phi^{-1}\left(\frac{\{\#\hat{\theta}_b^* < \hat{\theta}\}}{B}\right),$$

and

$$\hat{a} = \frac{\sum_{i=1}^n (\hat{\theta}^{(\cdot)} - \hat{\theta}^{(-i)})^3}{6\{\sum_{i=1}^n (\hat{\theta}^{(\cdot)} - \hat{\theta}^{(-i)})^2\}^{3/2}},$$

with

$$\hat{\theta}^{(\cdot)} = \frac{1}{n}\sum_{i=1}^n \hat{\theta}^{(-i)}.$$

The bias-correction $\hat{z}_0$ measures the median bias of $\hat{\theta}^*$. The acceleration $\hat{a}$ is based on so-called "jackknife values" $\hat{\theta}^{(-i)}$ and is called the acceleration because it refers to the rate of change of the standard error of t with respect to the true parameter value θ. The jackknife procedure is discused in Chapter 13.

The BC_a method has two advantages:

- It is transformation respecting.
- It is second-order accurate, meaning that error probabilties go to zero at rate $1/n$, compared to first-order accurate (a rate of $1/\sqrt{n}$) for the basic bootstrap and percentile method.

Example: the spatial test data

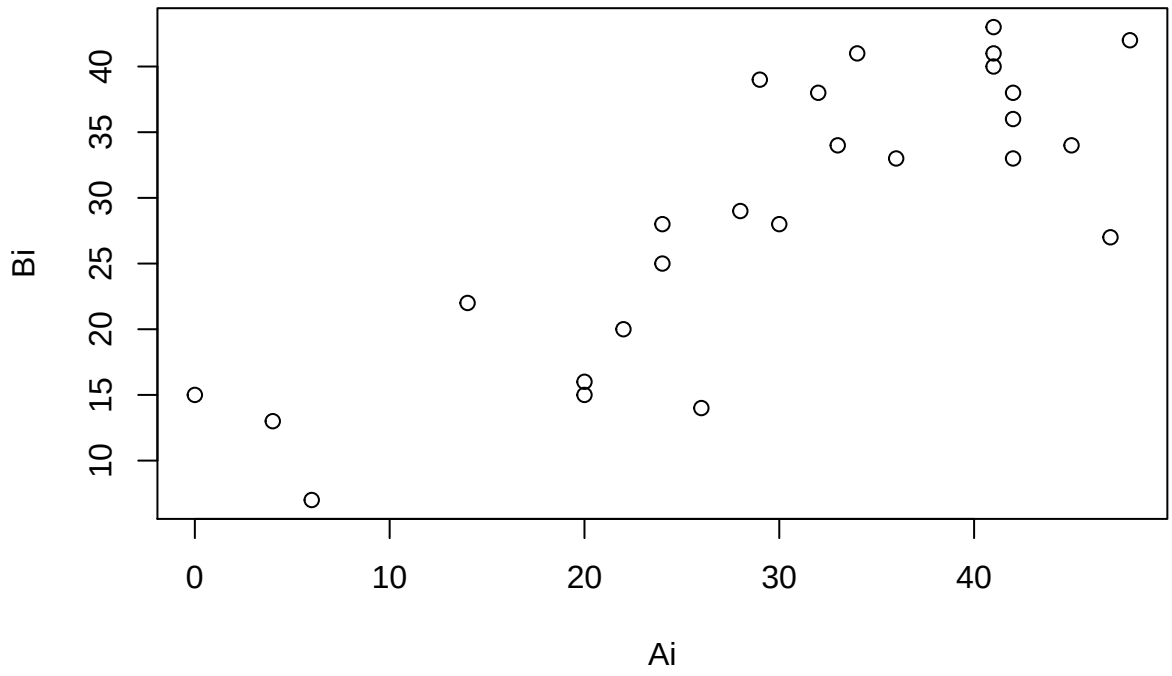
The spatial contains information about twenty-six neurologically impaired children that have each taken two tests of spatial perception, called "A" and "B".

```
attach(spatial)
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     B
```

A scaterplot of the data and the sample covariance matrix are whon below.

```
Ai<-spatial$A
Bi<-spatial$B
plot(Ai,Bi)
```




```
mean(Ai)
```

```
## [1] 29.65385
```

```
mean(Bi)
```

```
## [1] 28.88462
```

```
cov(spatial)
```

```
##          A        B
## A 178.3954 116.9585
## B 116.9585 113.7862
```

The parameter of primary interest is the variance of the variable A, σ_A^2. The sample variance is equal to $\hat{\sigma}_A^2 = 178.3954$. We use the function bcanon() from the R package bootstrap to calculate the BC_a interval.

```
x<-Ai
theta <- function(x){var(x)}
results <- bcanon(x,1000,theta,alpha=c(0.025,0.975))
```

A 95% C.I is given by

```
results
```

```
## $confpoints
##      alpha bca point
## [1,] 0.025  110.0738
## [2,] 0.975  289.9462
##
## $z0
## [1] 0.1687415
##
## $acc
## [1] 0.06124012
##
## $u
##  [1] 171.2433 184.0833 181.7900 185.8100 179.2233 179.2233 181.7900 179.2233
##  [9] 183.2900 180.2500 175.6233 175.2100 161.5833 147.7233 185.3433 185.7100
## [17] 185.0100 157.3100 185.5900 184.4433 172.7900 180.2500 184.4433 185.2500
## [25] 185.8233 180.2500
##
## $call
## bcanon(x = x, nboot = 1000, theta = theta, alpha = c(0.025, 0.975))
```

Part IV
Bootstrap inference

Bootstrap tests

Introduction

In this chapter we focus on resampling based inference and, as before, the bootstrap algorithm should reflect the original data generation mechanism. In addition, there is a second important rule: the bootstrap simulation should conducted under the null hypothesis.
The idea is to generate (repeatedly) new bootstrap data, reflecting the null hypothesis, recalculate the test statistic and in this way to simulate the null distribution of the test statistic. These bootstrap test values can then be used to compute a monte-carlo p-value (=bootstrap p-values). We discuss in this chapter non parametric bootstrap procedures for the population mean(s) in the setting of

- One sample tests.
- Two samples tests.

Other bootstrap tests for different settings such as correlation tests, test for proportions, linear regression, generalized linear models etc. will be discussed in later chapters.

Slides

Slides for the fourth part of the book about bootstrap and permutations tests can be found here: Slides4.

The One sample problem

We consider the treatment group of the mouse data. The survival times of seven mice, in days, after surgery are: 94,197,16,38,99,141,23 with mean survival time equal to 86.9 days and ML estimate for the standard deviation

$$\bar{\sigma} = \left(\frac{\sum_{i=1}^{n} (z_i - \bar{z})^2}{n-1} \right)^{0.5} = 66.8.$$

Suppose that we want to test the null hypothesis

$$H_0 : \mu_z = 129.$$

Classical tests under normality assumption

Test for one population using $N(0,1)$

Assuming a normal population, the distribution of the sample mean under the null hypothesis is given by

$$\bar{z} \sim N(129, \sigma^2/n).$$

Therefore, the test statistic

$$t(z) = \frac{\bar{z} - 129}{\sigma/\sqrt{n}},$$

follows a standard normal distribution. The P-value can be calculated by

$$\Phi\left(\frac{86.9-129}{\sigma/\sqrt{n}}\right).$$

In practice, σ is unknown so we use the estimate $\bar{\sigma}$,

$$\Phi\left(\frac{86.9-129}{66.8/\sqrt{7}}\right)=\phi(-1.67)=0.05.$$

A One sample t test

In R one sample t test for the hypotheses

$$H_0:\mu_z=129, H_1:\mu_z\leq 129,$$

can be conducted using the R function t.test().Note that the argument mu=129 and alternative="less" implies that wish to test the hypotheses formulated above.

```
z <- c(94, 197, 16, 38, 99, 141, 23)
z
```

```
## [1]  94 197  16  38  99 141  23
```

```
mean(z)
```

```
## [1] 86.85714
```

```
sd(z)
```

```
## [1] 66.76683
```

```
t.test(z,mu=129,alternative="less")
```

```
##
##  One Sample t-test
##
## data:  z
## t = -1.67, df = 6, p-value = 0.07298
## alternative hypothesis: true mean is less than 129
## 95 percent confidence interval:
##      -Inf 135.8942
## sample estimates:
## mean of x
##  86.85714
```

The text statistic $t(z)=\frac{\bar{z}-129}{\sigma/\sqrt{n}}$ is equal to

```
t.test(z,mu=129)$statistic
```

```
##         t
## -1.669984
```

and since the p value=0.072 we do not reject the null hypothesis.

One sample bootstrap test

Recall that the bootstrap simulation should satisfy the null hypothesis. Suppose that we resample from the empirical distribution $\hat{F}$ and consider the following bootstrap algorithm

- Simulate B samples of size n with replacement from z.
- Calculate the test statistic values $t(z^*)_1, \ldots, t(z^*)_b, \ldots, t(z^*)_B$, independently under the null hypothesis model,

$$t(z^*)_b = \frac{\bar{z^*} - 129}{\bar{\sigma^\star}/\sqrt{n}}, \quad b = 1, \ldots, B.$$

- Compute the Monte Carlo P-value:

$$p_{mc} = \frac{1 + (\#\{t(z^*)_b \geq t\})}{B + 1}.$$

The main problem is that the algorithm above dose not satisfy the null hypothesis $H_0 : \mu_z = 129$ since the mean of the empirical distribution is not 129 so it cannot be used for inference.

A non parametric bootstrap algorithm under H_0.

$\hat{F}$ is not an appropriate estimate for F since it dose not obey H_0, the mean of $\hat{F}$ is not 129. We need to obtain an estimate for F which has the mean of 129. This can be done if we define a new variable, $\tilde{z}$, such that

$$\tilde{z}_i = z_i - \bar{z} + 129.$$

The bootstrap algorithm for the one sample problem is as follows.

- Simulate B samples of size n with replacement from $\tilde{z}$.
- Calculate the test statistic values $t(z^*)_1, \ldots, t(z^*)_b, \ldots, t(z^*)_B$, independently under the null hypothesis model,

$$t(z^*)_b = \frac{\bar{z^*} - 129}{\bar{\sigma^*}/\sqrt{n}}, \quad b = 1, \ldots, B.$$

- Compute the Monte Carlo P-value

$$p_{mc} = \frac{1 + (\#\{t(z^*)_b \geq t\})}{B + 1}$$

Implemention in R

In the first step we substruct $\bar{z}$ and add μ_0 to the observed data,

```
mz <- mean(z)
mz
```

```
## [1] 86.85714
```

```
n <- length(z)
z.tilde <- z - mz + 129
mean(z.tilde)
```

```
## [1] 129
```

Note that $\bar{\tilde{z}} = 129$. Next we resample from $\tilde{z}$.

```
nz<-length(z)
t.obs<-t.test(z,mu=129)$statistic
B<-5000
t.boot<-c(1:B)
for(b in 1:B)
```

```
{
z.b<-sample(z.tilde,size=nz,replace=T)
t.boot[b]<-t.test(z.b,mu=129)$statistic
}
```

Monte-carlo p-value is calculated by

```
Pmc<-(1+sum(t.boot < t.obs))/(B+1)
Pmc
```

```
## [1] 0.09918016
```

Since Pmc > 0.05 we do not reject the null hypothesis. The distribution of the test statistics under the null hypothesis is shown in Figure @ref(fig:figchp61).

```
hist(t.boot,nclass=50,probability=T)
lines(c(t.obs,t.obs),c(0,1),lwd=3,col=2)
```

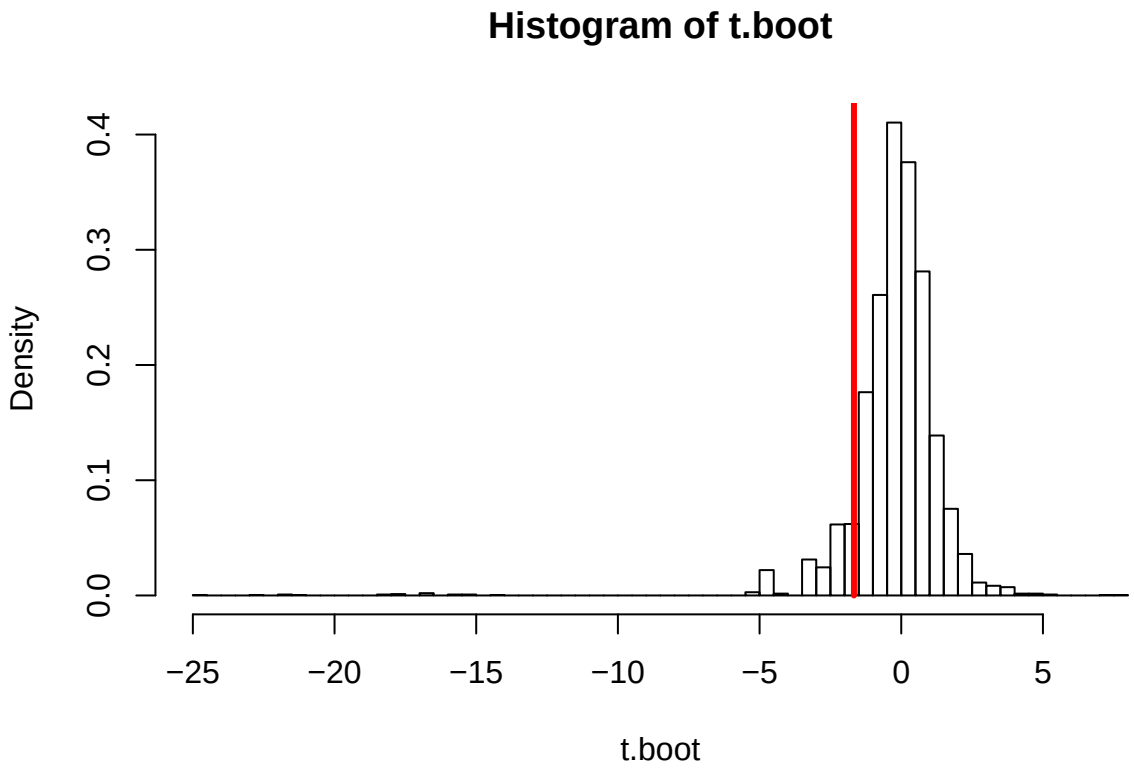


Figure 17: The distribution of the test statistic under the hull hypothesis.

The two samples problem

Consider the following setting: the response of a treated subject is denoted by Y with mean μ_y and for a control by X with mean μ_X, with distribution function $G(\mu_Y)$ and $F(\mu_X)$, respectively. The no-effect hypothesis can be translated into

$$H_0 : G = F,$$

or

$$H_0 : \mu_Y = \mu_X.$$

Let $x_1, \ldots, x_m$ and $y_1, \ldots, y_n$ be iid samples from $F(\mu_X)$ and $G(\mu_Y)$. Let $(z_1, z_2, \ldots, z_m, Z_{m+1}, \ldots, z_{m+n}) = (x_1, \ldots, x_m, y_1, \ldots, y_n)$.

Two samples t test

Consider the test statistic

$$t(z) = \frac{\bar{x} - \bar{y}}{\bar{\sigma}\sqrt{1/m + 1/n}},$$

with the pooled sample variance given by

$$\bar{\sigma} = \left(\frac{\sum_{i=1}^{m}(x_i - \bar{x})^2 + \sum_{i=1}^{n}(y_i - \bar{y})^2}{n + m - 2} \right)^{\frac{1}{2}}$$

Note that we assume here that the variances in the two populations are equal.
Under the null hypothesis $H_0 : \mu_Y = \mu_X$ and

$$t(z) \sim t_{n+m-2}.$$

Under the null hypothesis we expect that $t(z)$ will be "small". Therefore, we reject the null hypothesis for a "small" value of the statistic. Formally, for significant level α and one sided test, we reject H_0 if

$$t(z) > t_{(\alpha, m+n-2)}.$$

Example: the mouse data - classical two sample t test

For illustraion, we use the mouse data and test the null hypothesis that the mean of the control group is equal to the mean of the treatment group. Data are boxplot for the two groups are whon below.

```
y <- c(10, 27, 31, 40, 46, 50, 52, 104, 146)
x <- c(16, 23, 38, 94, 99, 141, 197)
y
```

```
## [1]  10  27  31  40  46  50  52 104 146
```

```
x
```

```
## [1]  16  23  38  94  99 141 197
```

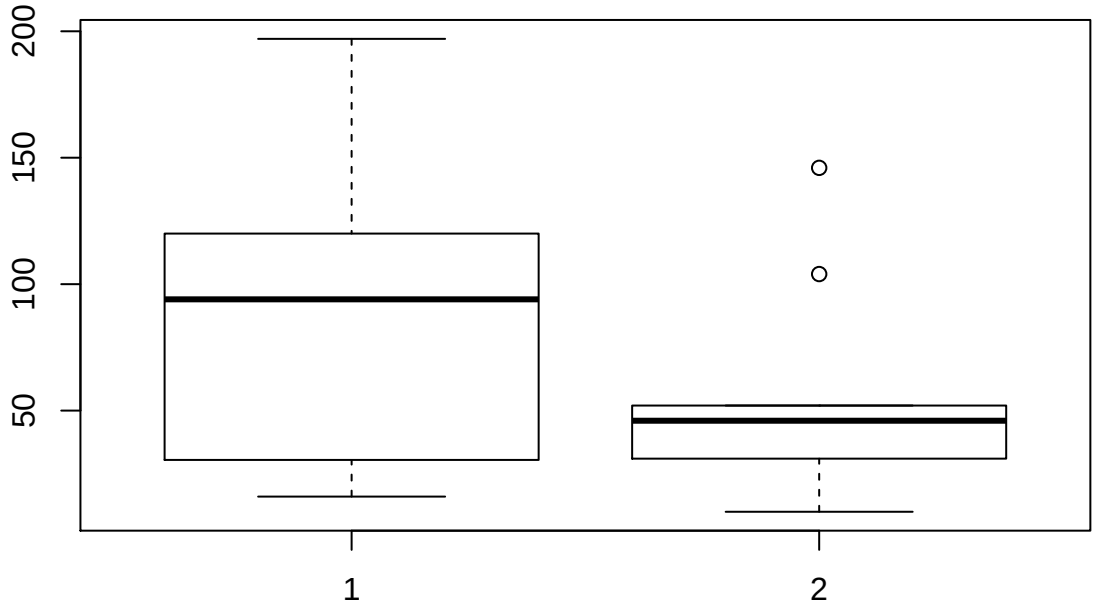
```
mean(y)
```

```
## [1] 56.22222
```

```
mean(x)
```

```
## [1] 86.85714
```

```
boxplot(x,y)
```



The p-value for a two sample t test, assuming a $t_{(14)}$ distribution for the test statistic, is equal to 0.28 indicates that the null hypothesis cannot be rejected.

```
t.test(x,y,var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  x and y
## t = 1.1214, df = 14, p-value = 0.281
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -27.95786  89.22770
## sample estimates:
## mean of x mean of y
##  86.85714  56.22222
```

Bootstrap test (Algorithm I)

YouTube tutorial: two samples t-test

An example for the implementation of bootstrap procedure for two-samples t-test is given by Matthew E. Clapham in YTBootstrap 6. This YouTube tutorial covers also permutations test that we discussed in the next chapter of the book

YouTube tutorial: bootstrap tests with R (the two-samples problem)

The tutorial by Mike Marin is focused on hypothesis testing in two-samples setting and discuss the implementation of a non parametric bootstrap procedure for two test statistics. See YTBootstrap 7.

Using the mean difference as test statistic

As before the bootstrap algorithm should reflect the original data generation mechanism and, for hypotheses testing, there is a second important rule, namely the bootstrap simulation should be counducted under null hypothesis.

In practice, we draw a sample of size $n+m$ with replacement from z. The first n are considered to be observations from $F(\mu_X)$ and the remaining m are considered as observations from $G(\mu_Y)$. In this section, we consider the following test statistic (see also the example in Mike Marin's tutorial):

$$t(z) = \bar{x} - \bar{y}.$$

A non parametric bootstrap algorithm for testing $H_0 : \mu_Y = \mu_X$ is given by

- Simulate B bootstrap samples of size $m+n$ with replacement from z. The first n are considered as observations $x^\star$ and the remaining m as observations $y^\star$.
- Calculate the test statistic values $t(z^*)_1, \ldots, t(z^*)_b, \ldots, t(z^*)_B$, independently under the null hypothesis,

$$t(z^*)_b = \bar{x^*} - \bar{y^\star}, \quad b = 1, \ldots, B.$$

- Compute the Monte Carlo P-value

$$p_{mc} = \frac{1 + (\#\{t(z^*)_b \geq t\})}{B+1}.$$

In R, we first need to define a new vector (z) that contains the joint sample:

```
z <- c(x, y)
z
```

```
##  [1]  16  23  38  94  99 141 197  10  27  31  40  46  50  52 104 146
```

The R object t.obs is the observed test statistic $(\bar{x} - \bar{y})$

```
m <- length(x)
n <- length(y)
mn <- m + n
t.obs <- mean(x) - mean(y)
t.obs
```

```
## [1] 30.63492
```

The non parametric bootstrap precedure is given below.

```
B<-5000
t.boot<-c(1:B)
for(b in 1:B)
{
z.b<-sample(z,size=mn,replace=T)
x.b<-z.b[1:n]
y.b<-z.b[(n+1):mn]
```

```
t.boot[b]<-mean(x.b)-mean(y.b)
}
```

As expcted, the distribution of the bbotstrap test statistics $t(z^*)_1, \ldots, t(z^*)_b, \ldots, t(z^*)_B$ under H_0, shown in Figure @ref(fig:figchp62), is centered around zero.

```
hist(t.boot,nclass=50,probability=T)
lines(c(t.obs,t.obs),c(0,1),lwd=3,col=2)
```

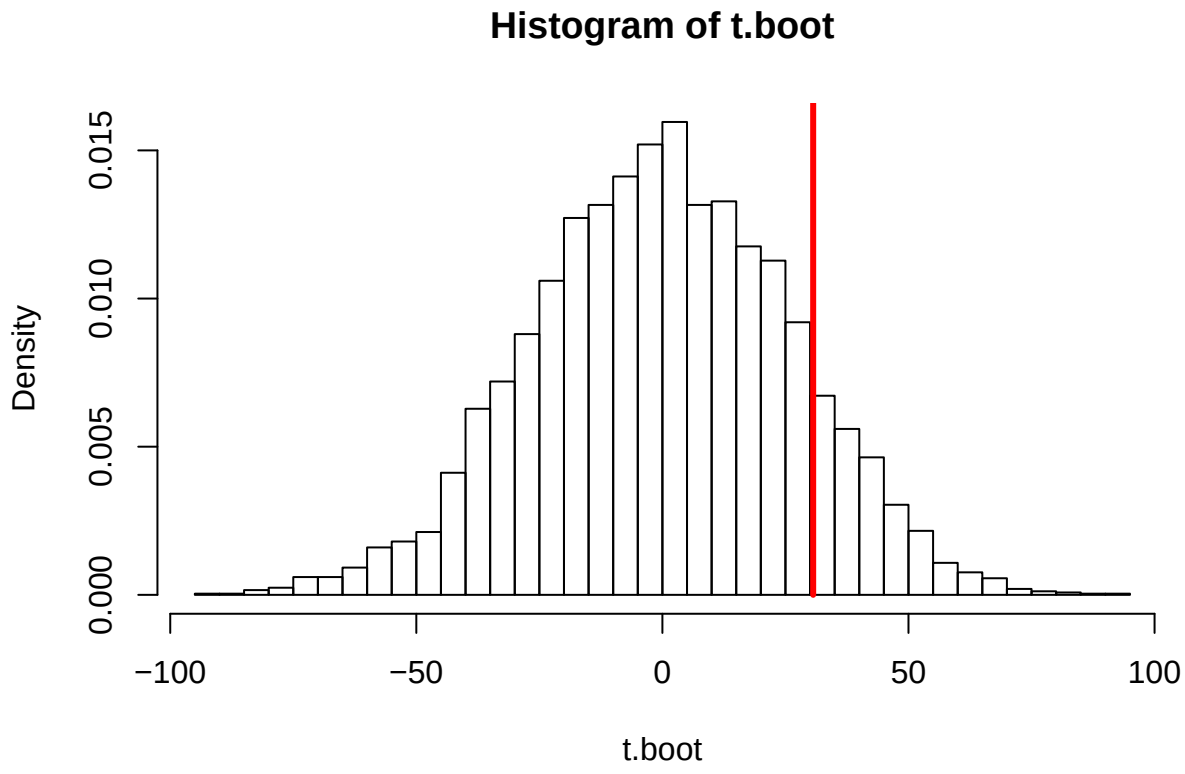


Figure 18: Distribution of sample means diference under the hull hypothesis.

The monte-carlo p-value for one sided alternative is equal to

```
(sum(t.boot>t.obs)+1)/(B+1)
```

```
## [1] 0.119976
```

A two-samples t-test

The test statistic defined in the previous section does not take the sample variability into account. Alternativly, we can use the usual two samples t-test statistic given by

$$t(z) = \frac{\bar{x} - \bar{y}}{\bar{\sigma}\sqrt{1/m + 1/n}},$$

The observed test statistic is equal to 1.059.

```
t.obs <- t.test(x,y)$statistic
t.obs
```

```
##        t
## 1.059062
```

The non parametric bootstrap precedure for testing $H_0 : \mu_Y = \mu_X$ is as follows

- Simulate B samples of size $m + n$ with replacement from z. The first n are considered as observations $x^\star$ and the remaining m as observations $y^\star$.
- Calculate the test statistic values $t(z^*)_1, \ldots, t(z^*)_b, \ldots, t(z^*)_B$, independently under the null hypothesis,

$$t(z^*)_b = \frac{\bar{x^*} - \bar{y^\star}}{\bar{\sigma^\star}\sqrt{1/m + 1/n}}, \quad b = 1, \ldots, B.$$

- Compute the Monte Carlo P-value

$$p_{mc} = \frac{1 + (\#\{t(z^*)_b \geq t\})}{B + 1}.$$

The procedure above can be implemented in R using the follwoing code

```
B<-2500
t.boot<-c(1:B)
for(b in 1:B)
{
z.b<-sample(z,size=mn,replace=T)
x.b<-z.b[1:n]
y.b<-z.b[(n+1):mn]
t.boot[b]<-t.test(x.b,y.b)$statistic
}
```

The distribution of the bootstrap test statistics under the null hypothesis and the bootstrap p value is presented in Figure @ref(fig:figchp63)

```
hist(t.boot,nclass=50,probability=T)
lines(c(t.obs,t.obs),c(0,1),lwd=3,col=2)
```

```
Pmc<-(1+sum(t.boot > t.obs))/(B+1)
Pmc
```

```
## [1] 0.1719312
```

The bootstrap p value is similar to the two sample t test below and in both cases we do not reject the null hypothesis. Note that in both cases, $H_0 : \mu_X > \mu_Y$

```
t.test(x,y,alternative="greater",var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  x and y
## t = 1.1214, df = 14, p-value = 0.1405
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -17.48178       Inf
## sample estimates:
## mean of x mean of y
##  86.85714  56.22222
```

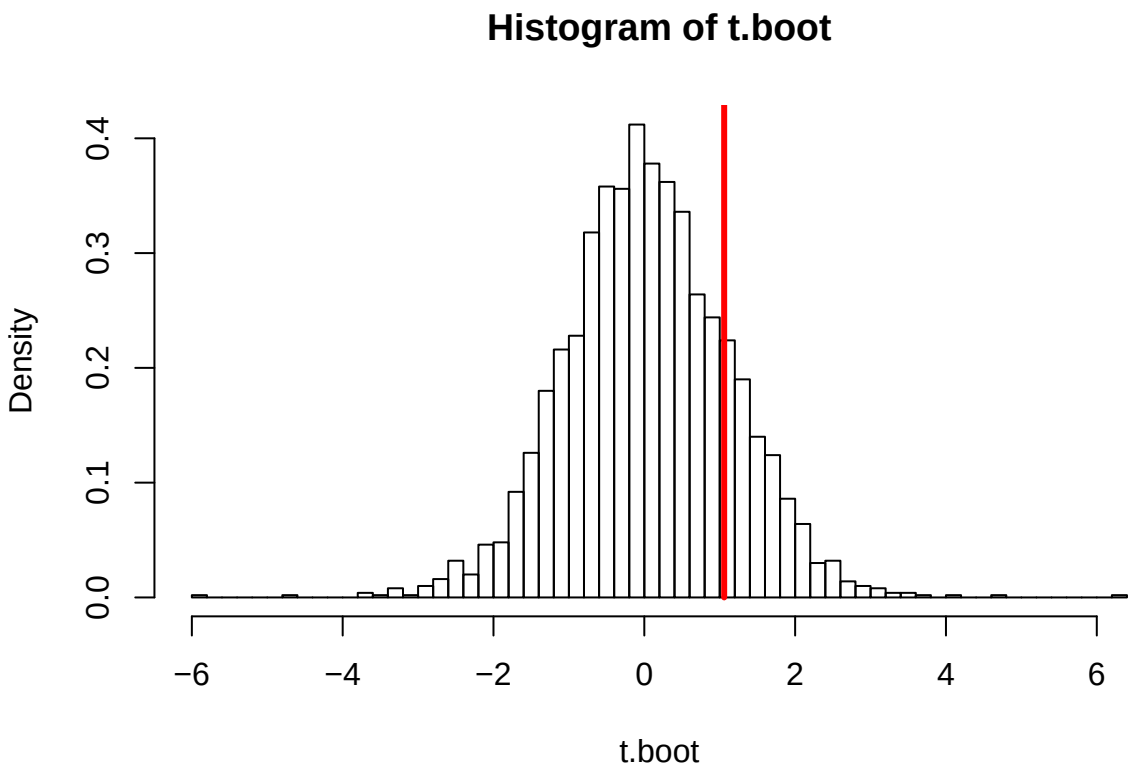


Figure 19: Distribution of the two-samples test statistics under the null hypothesis.

Bootstrap test (Algorithm II)

For this bootstrap algorithm both samples are transformed and centered around the overall mean of the joint sample z. In the sedonc step, we resample from $\tilde{x}$ and $\tilde{y}$. The non parametric bootstrap procedure is given below.

- Define the vectors $\tilde{x}$ and $\tilde{y}$.
- Simulate B samples of size m and n with replacement from $\tilde{x}$ and $\tilde{y}$.
- Calculate the bootstrap sample means $\bar{\tilde{x}}_b^*$ and $\bar{\tilde{y}}_b^*$
- Calculate the test statistic values $t(z^*)_1, \ldots, t(z^*)_b, \ldots, t(z^*)_B$, independently under the null hypothesis,

$$t(z^*)_b = \frac{\bar{x^*} - \bar{y^\star}}{\bar{\sigma^\star}\sqrt{1/m + 1/n}}, \quad b = 1, \ldots, B.$$

- Compute the Monte Carlo P-value

$$p_{mc} = \frac{1 + (\#\{t(z^*)_b \geq t\})}{B + 1}.$$

For the mouse data, the above bootstrap algorithm is implemented in R in the follwing way. First we define $\tilde{x}$ and $\tilde{y}$.

```
y <- c(10, 27, 31, 40, 46, 50, 52, 104, 146)
x <- c(16, 23, 38, 94, 99, 141, 197)
z<-c(x,y)
m <- length(x)
n <- length(y)
my<-mean(y)
mx<-mean(x)
mz<-mean(z)
x.tilde<-x-mx+mz
y.tilde<-y-my+mz
```

Note that the mean of $\tilde{x}$ and $\tilde{y}$ are equal to the overall mean as requested.

```
c(mean(x.tilde),mean(y.tilde))
```

```
## [1] 69.625 69.625
```

Next, we resample with replacent from both $\tilde{x}$ and $\tilde{y}$.

```
t.obs <- t.test(x,y)$statistic
t.obs
```

```
##        t
## 1.059062
```

```
B<-1000
t.boot<-c(1:B)
for(b in 1:B)
{
x.b<-sample(x.tilde,m,replace=TRUE)
y.b<-sample(y.tilde,n,replace=TRUE)
t.boot[b]<-t.test(x.b,y.b)$statistic
}
```

The distribution of the bootstrap test statistics and the monte-carlo p value are shown in Figure @ref(fig:figchp64). Since the p-value > 0.05 we do not rehect the null hypothesis.

```
hist(t.boot,nclass=50,probability=T)
lines(c(t.obs,t.obs),c(0,1),lwd=3,col=2)
```

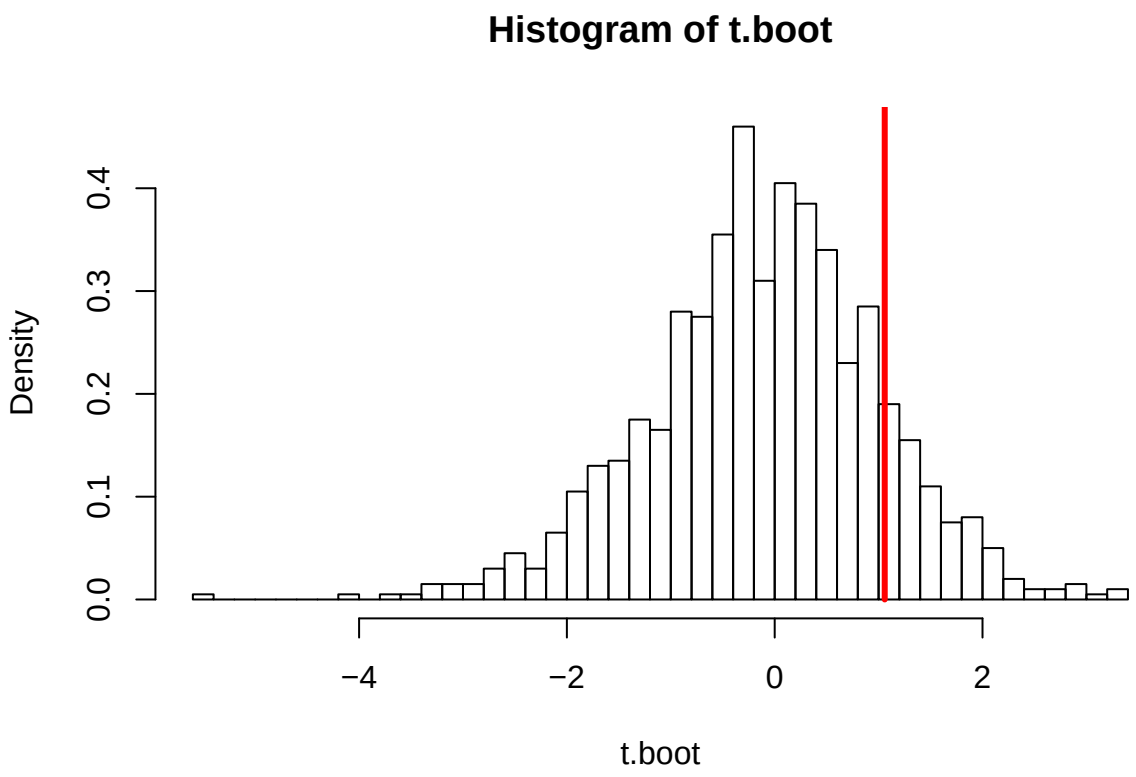


Figure 20: Distribution of the test statistic under the null hypothesis.

```
Pmc<-(1+sum(t.boot > t.obs))/(B+1)
Pmc
```

```
## [1] 0.1338661
```

The bootstrap test for correlation

The cars data

The cars data gives the speed of 50 cars and the distances taken to stop and shown in Figure @ref(fig:figchp70). The data were recorded in the 1920s.

```
plot(cars$speed,cars$dist)
```

Let x be the car speed and y be the stopping distance.

```
x<-cars$speed
y<-cars$dist
```

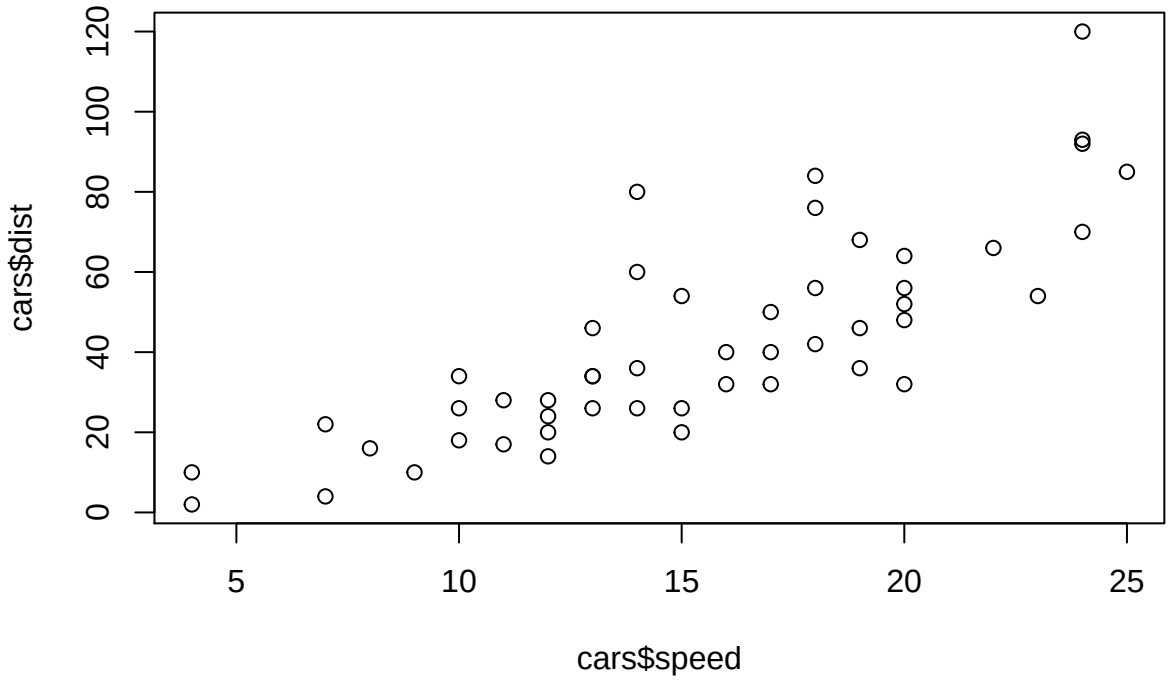


Figure 21: The cars dataset. Cars' speed versus the stoping distance.

In this chapter, the parameter of primary interest is the correlation $\rho(x, y)$. The sample correlation is equal to

```
rho<-cor(x,y)
rho
```

```
## [1] 0.8068949
```

Our aim is to test the null hypothesis

$$H_0 : \rho = 0.$$

Non parametric bootstrap

Given the pairs $(x_1, y_1), \ldots, (x_n, y_n)$, a possible test statistic, consider the above null hypothes that x and y are independent, can be based on the sample correlation $\hat{\rho}$.The nonparametric bootstrap test for zero correlation does not assume anything about the joint and marginal distributions of x and y. A way to generate new uncorrelated x and y bootstrap values is to resample both separately, with replacement from their empirical distribution.

```
n<-length(x)
B<-1000
coeff.boot<-c(1:B)
for(b in 1:B)
{
x.boot<- sample(x,size=n,replace=TRUE)
y.boot<- sample(y,size=n,replace=TRUE)
coeff.boot[b]<-cor(x.boot,y.boot)
}
```

The distribution of the bootstrap test statistics, $\hat{\rho}_1^*, \hat{\rho}_2^*, \ldots, \hat{\rho}_B^*$ is shown in Figure @ref(fig:figchp71a)

```
hist(coeff.boot,nclass=50,xlim=c(-1,1))
lines(c(rho,rho),c(0,500),col=2,lwd=3)
```

with monte-carlo p value given by

```
(1+sum(abs(coeff.boot)>rho))/(B+1)
```

```
## [1] 0.000999001
```

Note that the bootstrap procedure above is NOT the same as the non-parametric bootstrap procedure for the case that a confidence interval for ρ is of interest (that was discussed in Section 4.4). For this case we need to resamples pairs and the bootstrap algorithm is implemented as follows:

```
n<-length(x)
index<-c(1:n)
B<-1000
coeff.boot<-c(1:B)
index<-c(1:n)
for(b in 1:B)
{
index.b<- sample(index,size=n,replace=TRUE)
x.boot<- x[index.b]
y.boot<- y[index.b]
coeff.boot[b]<-cor(x.boot,y.boot)
}
```

The estimate for the distribution of $\hat{rho}$ is shown in Figure @ref(fig:figchp71b).

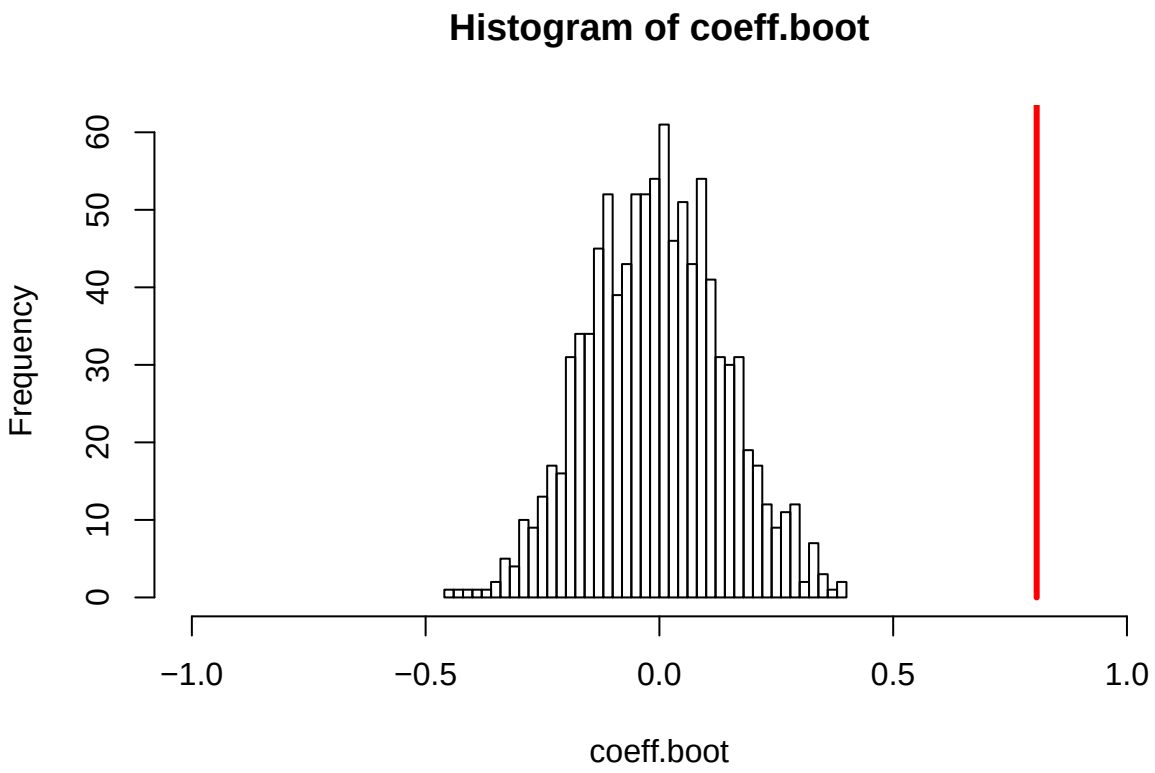


Figure 22: Non parametric bootstrap: distribution of the test statistic under the null hypothesis.

```
hist(coeff.boot,nclass=50)
lines(c(rho,rho),c(0,500),col=2,lwd=3)
```

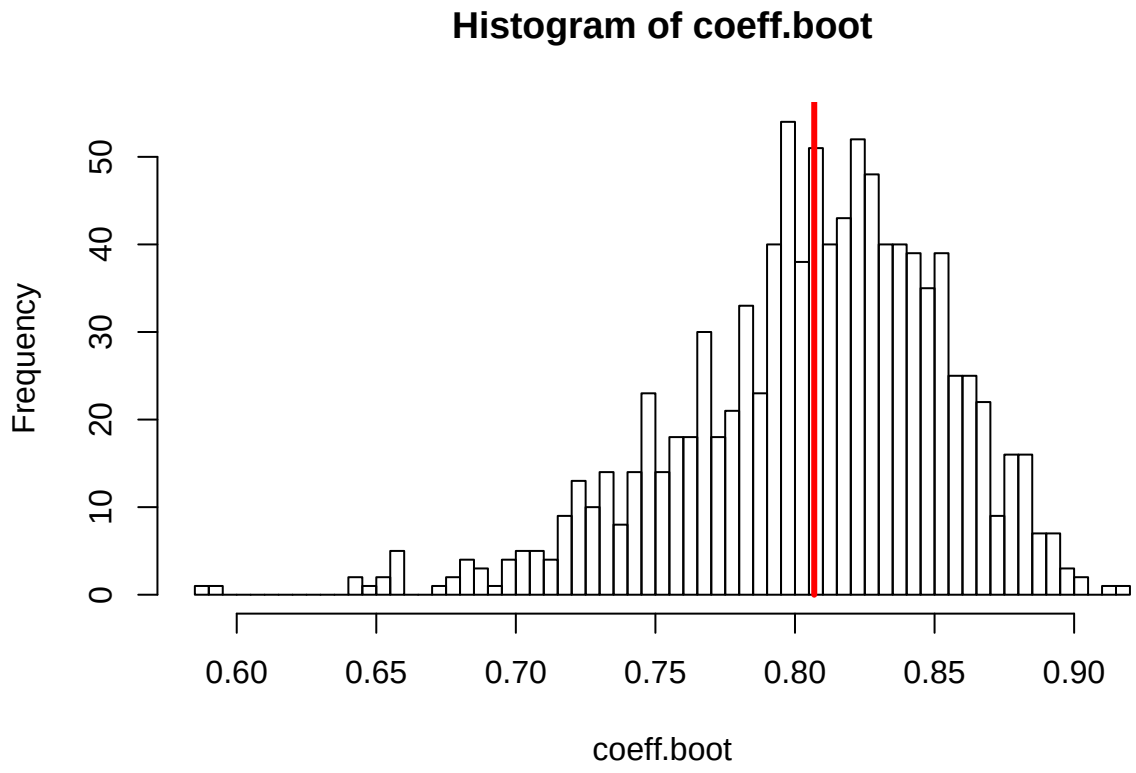


Figure 23: Non parametric bootstrap: distribution of the bootstrap replicates when the resampling is done by pairs.This distribution CANNOT be used for inference

Parametric bootstrap

For parametric bootstrap we need to make an assumption about the joint distribution $F(x, y, \theta)$ under the null hypothesis. Let us assume that the joint distribution of x and y is a bivariate normal distribution. Since the null hypothesis states that x and y are independent, it implies that the marginal distributions of x and y are two independent normal distributions (so $\rho = 0$). Hence, under H_0, $x \sim N(\mu_x, \sigma_x^2)$ and $y \sim N(\mu_y, \sigma_y^2)$. We cannot resample from these distributions since the parameters are unknown and therefore we will replace them with the plug-in estimators and resample from

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

The pulg-in estimators for $\hat{F}_x$ and $\hat{F}_y$

```
MLx<-mean(x)
MLy<-mean(y)
Sigx<-sqrt(var(x))
Sigy<-sqrt(var(y))
```

A parametric bootstrap is implemented using the follwoing code:

```
n<-length(x)
B<-1000
coeff.boot<-c(1:B)
for(b in 1:B)
{
x.boot<- rnorm(n,MLx,Sigx)
y.boot<- rnorm(n,MLy,Sigy)
coeff.boot[b]<-cor(x.boot,y.boot)
}
```

Figure @ref(fig:figchp72) shows the distribution of the bootstrap test statistics under H_0.

```
hist(coeff.boot,nclass=50,xlim=c(-1,1))
lines(c(rho,rho),c(0,500),col=2,lwd=3)
```

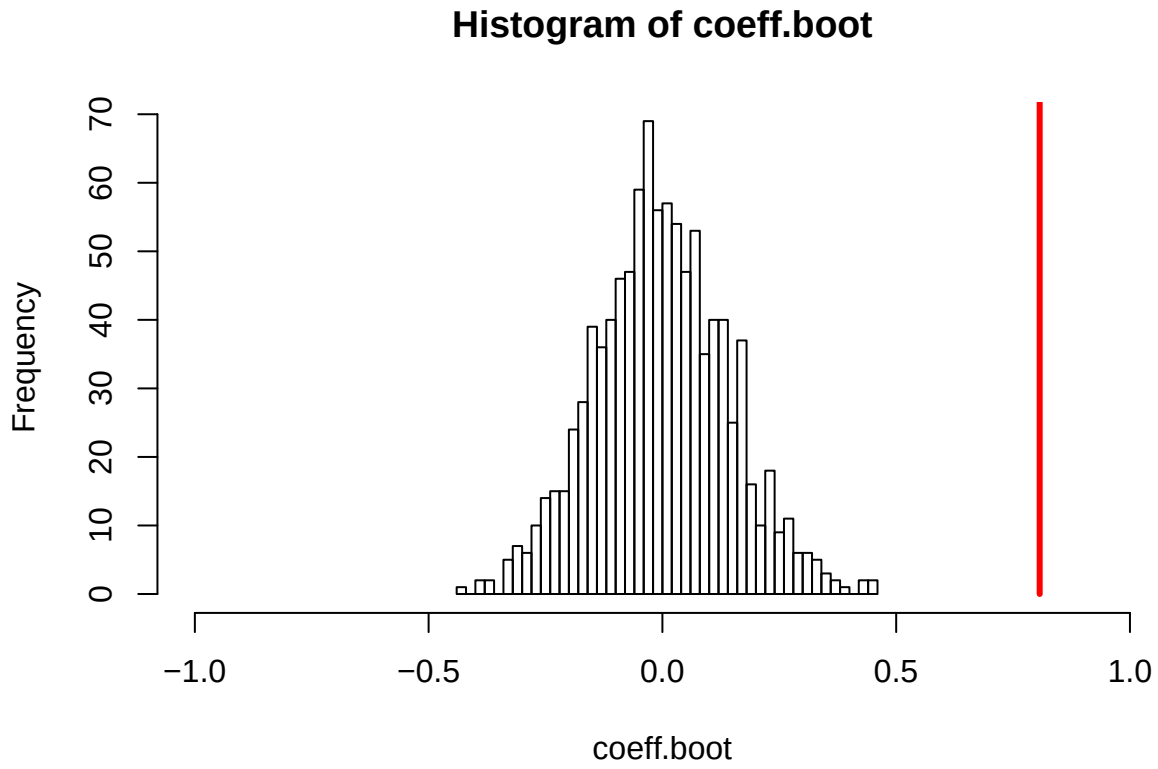


Figure 24: Parametric bootstrap: distribution of the test statistic under the null hypothesis.

```
(1+sum(abs(coeff.boot)>rho))/(B+1)
```

```
## [1] 0.000999001
```

Note that instead of re sampling from two independent normal distributions, we can sample pairs from the null bivariate normal distribution

$$\begin{pmatrix} x \\ y \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_x \\ \mu y \end{pmatrix}, \Sigma_{H_0}\right).$$

Here, Σ_{H_0} is a 2×2 the covariance matrix given by

$$\Sigma = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix}.$$

Premutaions test

Resampling without replacement

The nonparametric bootstrap procedures that were discussed in the previous chapter are based on an algorithm of sampling with replacement. An alternative is to resample without replacement. Consider the two samples problem in which we would like to test the null hypothesis

$$H_0 : \mu_x = \mu_y.$$

Let $(x_1, \ldots, x_{n\_}$ and $(y_1, \ldots, y_m)$ the samples from the two populations and the combined sample $z = (x, y)$

The non parametric bootstrap precedure without replacent for testing H_0 is as follows

- Simulate B samples of size $m + n$ without replacement from z. The first n are considered as observations $x^\star$ and the remaining m as observations $y^\star$.
- Calculate the test statistic values $t(z^*)_1, \ldots, t(z^*)_b, \ldots, t(z^*)_B$, independently under the null hypothesis,

$$t(z^*)_b = \frac{\bar{x^*} - \bar{y^\star}}{\bar{\sigma^\star}\sqrt{1/m + 1/n}}, \quad b = 1, \ldots, B.$$

Its conditional null distribution is determined by that of $z_1, \ldots, z_n$, which is uniform over all permutations of random selection of n observations from the vector $z_1, \ldots, z_{n+m}$. For a selection of B random permutations of $z_1, \ldots, z_{n+m}$ we can compute the Monte Carlo p-value in the same way that it was calculated in Section 6.3.2.3.

YouTube tutorial: Permutation tests in R

A tutorial by Ian Dworkin about permutation test in R. The tutorial is focused on a One-Way ANOVA with two groups using a resampling procedure without replacement. This example I equivalent to a permutation test for two samples t-test. Within the context of this course, this example is similar to the examples discussed in the chapter aboutlinear model. For the tutorial click here: YTBootstrap 8.

Premutaions test for the mouse data

The data and boxplot for the mouse data are shwon in Figure @ref(fig:figchp81)

```
x<-c(94,197,16,38,99,141,23)
y<-c(52,104,146,10,51,30,40,27,46)
z<-c(x,y)
z
```

```
##  [1]  94 197  16  38  99 141  23  52 104 146  10  51  30  40  27  46
```

```
n<-length(x)
m<-length(y)
boxplot(x,y)
```

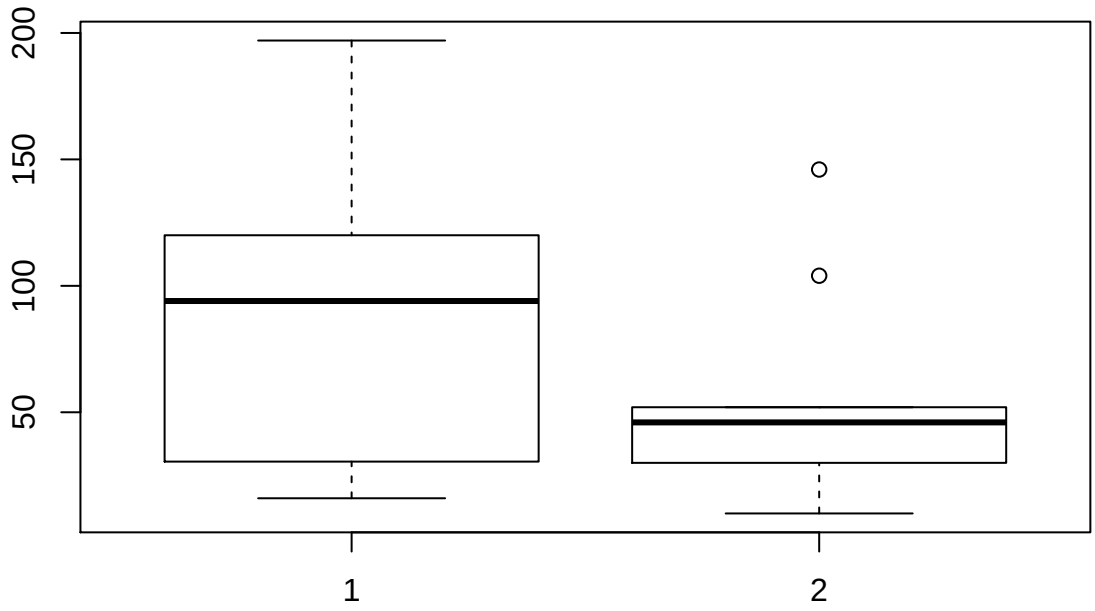


Figure 25: Survival time of mice per group

Recall that a classical two samples t-test leads to the following results:

```
t.test(x,y,alternative="greater",var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  x and y
## t = 1.1208, df = 14, p-value = 0.1406
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -17.50517       Inf
## sample estimates:
## mean of x mean of y
##  86.85714  56.22222
```

```
t.obs<-t.test(x,y,alternative="greater",var.equal=TRUE)$statistic
```

For the permutation test, we first need to define the combine sample (z):

```
z<-c(x,y)
r.x<-rep("x",n)
r.y<-rep("y",m)
r.x<-c(r.x,r.y)
data.frame(sort(z),r.x[order(z)])
```

```
##    sort.z. r.x.order.z..
## 1        10             y
## 2        16             x
## 3        23             x
## 4        27             y
## 5        30             y
## 6        38             x
## 7        40             y
## 8        46             y
## 9        51             y
## 10       52             y
## 11       94             x
## 12       99             x
## 13      104             y
## 14      141             x
## 15      146             y
## 16      197             x
```

The permutation procedure is implemented using a non parametric bootstrap in which the resampling is done without replacement (i.e., using the argument: replace=FALSE).

```
B<-1000
v.boot<-t.boot<-c(1:B)
for(b in 1:B)
{
z.boot<-sample(z,size=n+m,replace=FALSE)
v.boot[b]<-var(z.boot)
x.boot<-z.boot[1:n]
y.boot<-z.boot[(n+1):(n+m)]
t.boot[b]<-t.test(x.boot,y.boot,alternative="greater",var.equal=TRUE)$statistic
}
```

Figure @ref(fig:figchp82) shows the distribution of the permutation test statistics and Monte Carlo p value. Similar to the analysis presnted in Chapter 6, we do not reject the null hypothesis.

```
hist(t.boot,nclass=50,probability=TRUE)
lines(c(t.obs,t.obs),c(0,10),col=3,lwd=3)
```

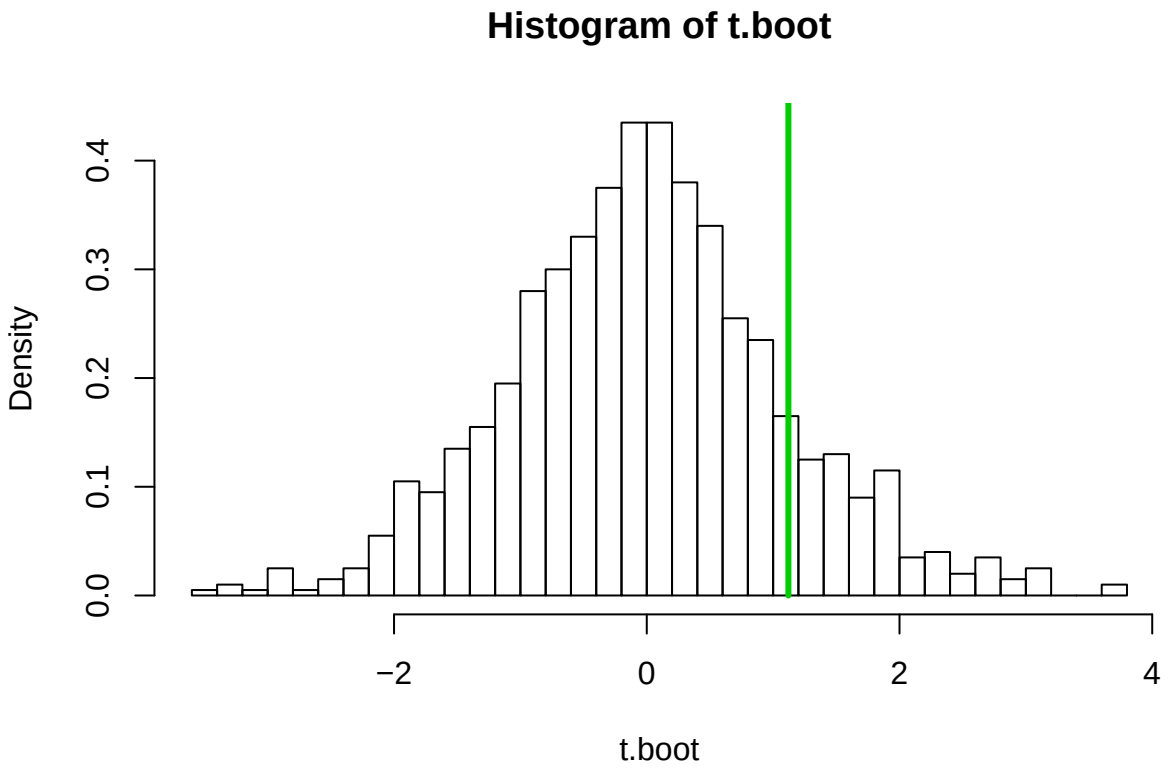


Figure 26: Permutaion test statistics

```
(1+sum(t.boot>t.obs))/(B+1)
```

```
## [1] 0.1398601
```

Resampling based inference for Binary data

The setting

Let $x_1, \ldots, x_n$ and $y_1, \ldots, y_m$ two samples from the populations: $x_i \sim B(\pi_1, n)$ and $y_i \sim B(\pi_2, m)$. Both π_1 and π_2 are unknown parameters. We wish to test the null hypothesis

$$H_0 : \pi_1 = \pi_2.$$

We use a two sample t test statistic for proporions

$$t = \frac{\hat{\pi}_1 - \hat{\pi}_2}{\sqrt{\pi(1-\pi)\left(\frac{1}{n}+\frac{1}{m}\right)}} \sim N(0;1).$$

Here, $\hat{\pi}_1$ and $\hat{\pi}_2$ are the plug-in estimators for the two unknown proportions. Since π is unknown we estimate it by the pooled proporion given by

$$\hat{\pi} = \frac{n\hat{\pi}_1 + m\hat{\pi}_2}{n+m},$$

as an estimate for the proportion under H_0. Under the null hypothesis, the test statistic, given below, follows a standard normal distribution.

$$t = \frac{\hat{\pi}_1 - \hat{\pi}_2}{\sqrt{\hat{\pi}(1-\hat{\pi})\left(\frac{1}{n}+\frac{1}{m}\right)}} \sim N(0;1).$$

Example: Relationship between aspirin use and myocardial infarction

A five-year randomized study (discussed in Agresti, 0000 testing whether regular intake of aspirin reduces mortality from cardiovascular disease. Every other day, the male physicians participating in the study took either one aspirin tablet or a placebo. The data, in a table format, are given below.

```
phs <- matrix(c(189,10845,104,10933),byrow=TRUE,ncol=2)
dimnames(phs) <- list(Group=c("Placebo","Aspirin"),MI=c("Yes","No"))
phs
```

```
##          MI
## Group     Yes    No
##   Placebo 189 10845
##   Aspirin 104 10933
```

Classical test for $H_0 : \pi_1 = \pi_2$.

To calculate the difference between the two proportions we define the following R objects:

```
n.P<-189+10845
n.A<-104+10933
Placebo<-c(rep(1,189),rep(0,10845))
Aspirin<-c(rep(1,104),rep(0,10933))
P.Placebo<-189/(189+10845)
P.Aspirin<-104/(104+10933)
```

Note that the R objects Placebo and Aspirin are binary vectors of the two sampels. The difference $\hat{\pi}_1 - \hat{\pi}_2$:

```
P.Placebo-P.Aspirin
```

```
## [1] 0.007706024
```

The pool proportion, $\hat{\pi}$:

```
P.tot<-(189+104)/(189+10845+104+10933)
P.tot
```

```
## [1] 0.01327534
```

Test statistics and p value (based on $N(0,1)$ distribution) indicate that the null hypothesis should be rejected.

```
z<-(P.Placebo-P.Aspirin)/sqrt(P.tot*(1-P.tot)*(1/n.P+1/n.A))
z
```

```
## [1] 5.001388
```

```
1-pnorm(z,0,1)
```

```
## [1] 2.845948e-07
```

Parametric bootstrap test for $H_0 \pi_1 = \pi_2$.

For parametric bootstrap, we assume that , under H_0, $x_i \sim B(\pi, n)$ and $y_i \sim B(\pi, m)$. Since π is unknown we use the pooled samples proportion $\hat{\pi}$. The parametric bootstrap is implemented below. Note that here, $\hat{F}_x = B(\hat{\pi}, n)$.

```
B<-1000
z.b<-c(1:B)
for(i in 1:B)
{
S1<-sum(rbinom(n.P,1,P.tot))
P.P.b<-S1/(189+10845)
S2<-sum(rbinom(n.A,1,P.tot))
P.A.b<-S2/(104+10933)
P.tot.b<-(S1+S2)/(n.P+n.A)
z.b[i]<-(P.P.b-P.A.b)/sqrt(P.tot.b*(1-P.tot.b)*(1/n.P+1/n.A))
}
```

The ditribution of the bootstrap test statistics unfer H_0 is presented in Figure @ref(fig:figchp91)

```
hist(z.b,nclass=50,xlim=c(-6,6),probability=TRUE)
lines(c(z,z),c(0,100),col=2,lwd=2)
```

The bootstrap p-value<0.05 and therefore the null hypothesis is rejected.

```
(1+sum(abs(z.b)>abs(z)))/(B+1)
```

```
## [1] 0.000999001
```

Non parametric bootstrap test for $H_0 \pi_1 = \pi_2$.

For non parametric bootstrap, we define a new vector $(z.i)$ for the joint sample:

```
z.i<-c(Placebo,Aspirin)
nz<-length(z.i)
nz
```

```
## [1] 22071
```

A non-parametric bootstrap procedure consists of resampling B bootstrap samples with replacement from the joint sample, which is the same algorithm that was used for the two sample problems in Chapter 6.

```
B<-1000
z.b<-c(1:B)
for(i in 1:B)
{
z.boot<-sample(z.i,size=nz,replace =TRUE)
```

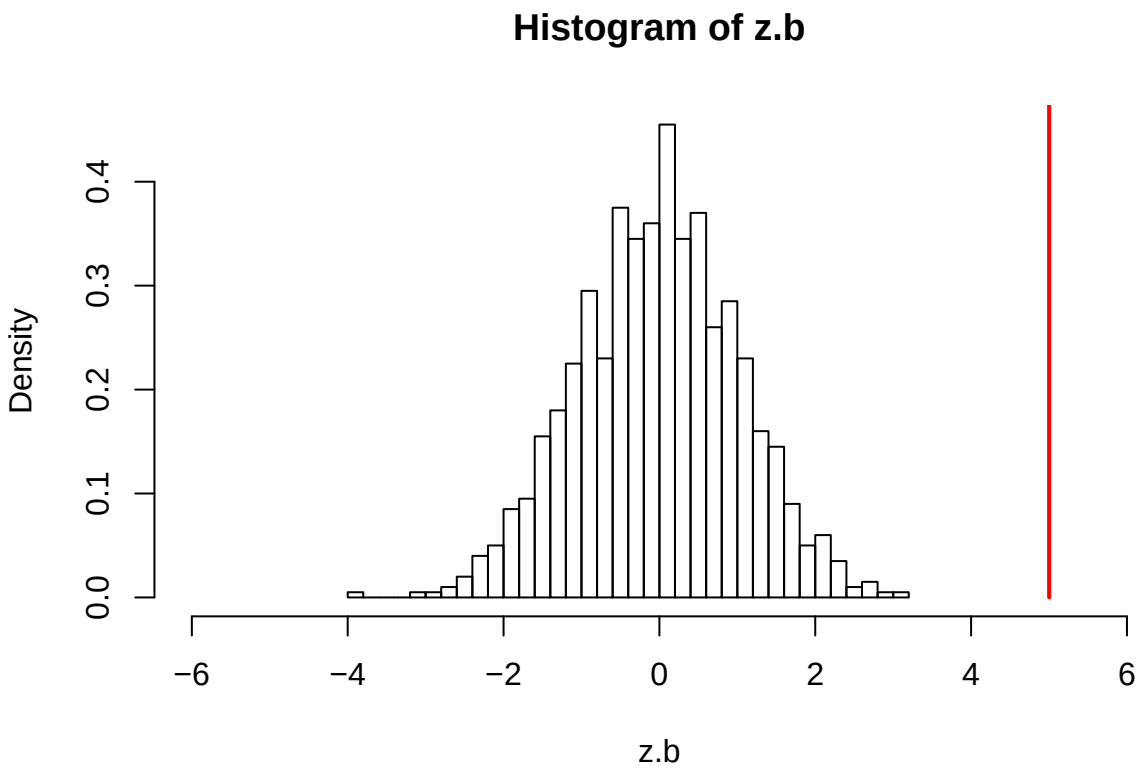


Figure 27: Non parametric bootstrap: test statistics under the null hypothesis

```
x.boot<-z.boot[1:n.P]
y.boot<-z.boot[(n.P+1):nz]
S1<-sum(x.boot)
P.P.b<-S1/(189+10845)
S2<-sum(y.boot)
P.A.b<-S2/(104+10933)
P.tot.b<-(S1+S2)/(n.P+n.A)
z.b[i]<-(P.P.b-P.A.b)/sqrt(P.tot.b*(1-P.tot.b)*(1/n.P+1/n.A))
}
```

The distribution of the bootstrap test statistics under H_0 is shown in Figure @ref(fig:figchp92).

```
hist(z.b,nclass=50,xlim=c(-6,6),probability=TRUE)
lines(c(z,z),c(0,100),col=2,lwd=2)
```

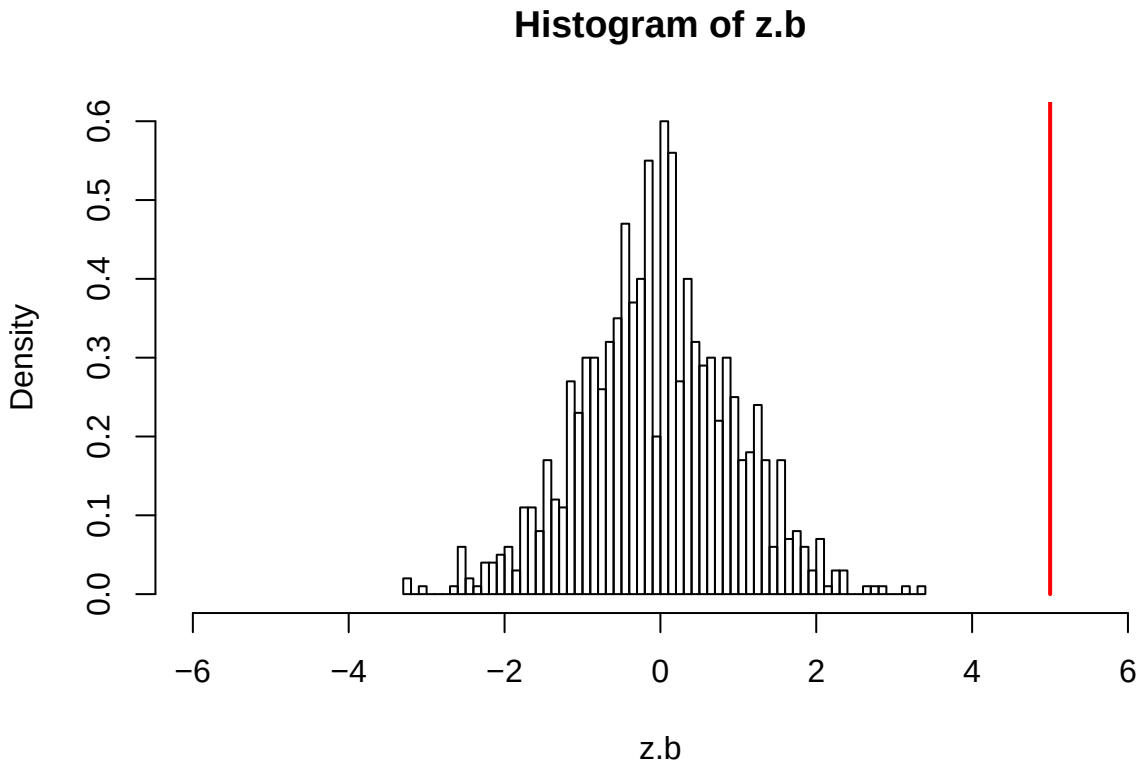


Figure 28: Parametric bootstrap: test statistics under the null hypothesis

The bootstrap p-value<0.05 and similar to the previous section, the null hypothesis is rejected.

```
(1+sum(abs(z.b)>abs(z)))/(B+1)
```

```
## [1] 0.000999001
```

Permutaion test for $H_0 \pi_1 = \pi_2$.

A Permutation procedure is similar to the non-parametric bootstrap procedure above but, now we resample B bootstrap samples WITHOUT replacement from the joint sample (we use the argument: replace =FALSE).

```
z.b<-c(1:B)
z.i<-c(Placebo,Aspirin)
nz<-length(z.i)
nz
```

```
## [1] 22071
```

```
for(i in 1:B)
{
z.boot<-sample(z.i,size=nz,replace =FALSE)
x.boot<-z.boot[1:n.P]
y.boot<-z.boot[(n.P+1):nz]
S1<-sum(x.boot)
P.P.b<-S1/(189+10845)
S2<-sum(y.boot)
P.A.b<-S2/(104+10933)
P.tot.b<-(S1+S2)/(n.P+n.A)
z.b[i]<-(P.P.b-P.A.b)/sqrt(P.tot.b*(1-P.tot.b)*(1/n.P+1/n.A))
}
```

Figure @ref(fig:figchp93) shows the distribution of the permutaion test statistics under H_0.

```
hist(z.b,nclass=50,xlim=c(-6,6),probability=TRUE)
lines(c(z,z),c(0,100),col=2,lwd=2)
```

The permutaion p value indicates that the null hypothesis should be rejected.

```
(1+sum(abs(z.b)>abs(z)))/(B+1)
```

```
## [1] 0.000999001
```

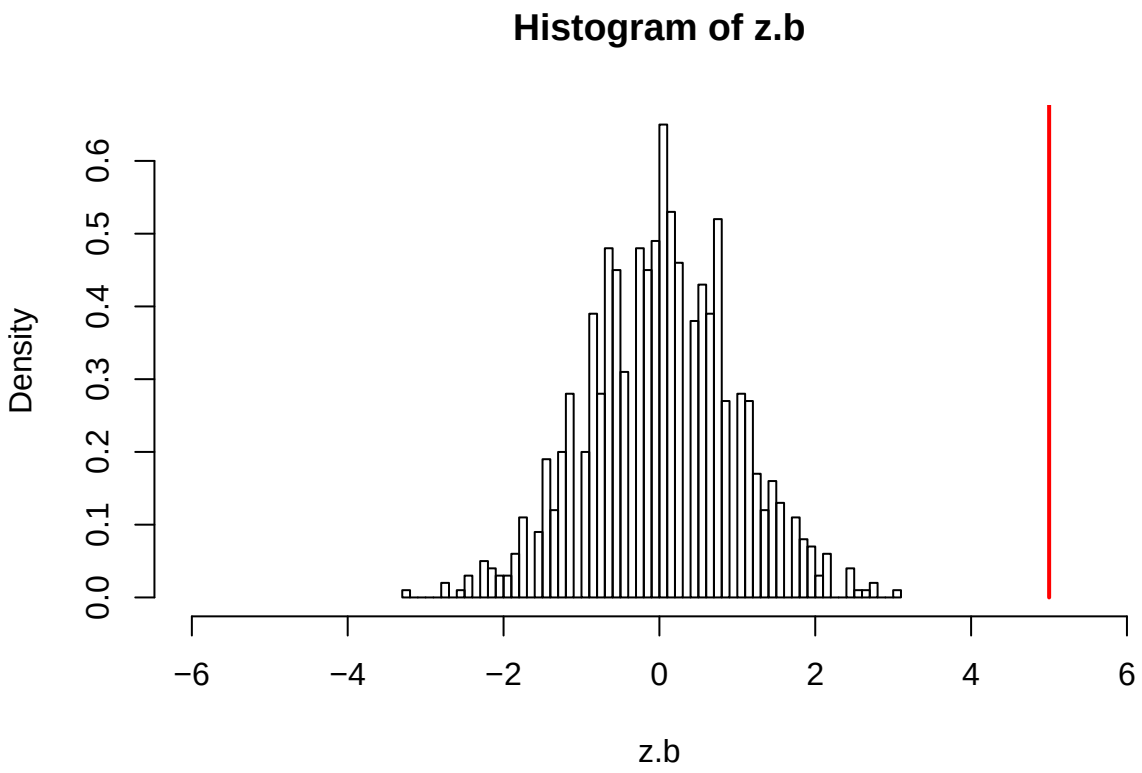


Figure 29: Permutaion test: test statistics under the null hypothesis

Part V
Modeling

Linear models

Introduction

The setting and notation

Consider data (x_i, y_i), $i = 1, \ldots, n$, is a $1 \times P + 1$ column-vector containing the values of the explanatory variables (predictors) and y_i is the dependent variable. In parametric linear regression, we assume the following model, for $i = 1, \ldots, n$,

$$y_i = x_i^T \beta + \varepsilon_i,$$

where $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ (independent). The regression coefficients $(\beta_0, \beta_1, \ldots, \beta_P)$ can be estimated using the ordinary least squares method or by a more robust method like e.g. least trimmed squares regression.

In this chapter we discuss three bootstrap procedures within the context of linear regression models:

- Non parametric bootstrap.
- Semi-parametric bootstrap
- Parametric bootstrap.

Slides

Slides for the fifth part of the book about bootstrap methods for linear regression and generalized linear models can be found here: Slides5.

YouTube tutorial: bootstrap application for linear regression.

A YouTube Ian Dworkin presents a bootstrap application for multiple linear regression using the *lm* function in R. The tutorial is focused on estimation and C.Is. See YTBootstrap 9.

Non parametric bootstrap

As usual, we do not need to make any distributional assumption if we implement a non-parametric bootstrap procedure to estimate the unknown parameters in the linear model defined above. Note that bootstrap procedures for inference will be discussed in Section 10.2.3. Since our main focus in this section is estimation and confidence intervals we need to keep the correlation structure of the data fixed. Therefore, the resampling procedure consists of resampling pairs of observations. In what follows, we focus on a simple linear regression model. We apply the following bootstrap for a simple linear regression model:

Let the observed data be $z_i = (x_i, y_i)$, $i = 1, \ldots, n$. We apply the following bootstrap:

*Take B resamples of size n

$$\begin{array}{cccc} z_1^*(1), & \ldots, & z_n^*(1) \\ z_1^*(2), & \ldots, & z_n^*(2) \\ \vdots & & \\ z_1^*(B), & \ldots, & z_n^*(B) \end{array}$$

*For each bootstrap sample, $b = 1, \ldots, B$ fit the model

$$y_i^* = \beta_0 + \beta_1 x_i^* + \varepsilon_i,$$

and estimate the unknown parameters in the models

$$\beta_1^* \ldots, \beta_B^*$$

* Use the above bootstrap replicates for $\beta = (\beta_0 + \beta_1)$ to estimate the standard error, confidence intervals, distribution etc.

Example: the hormone data

The hormone data contains information about the amount in milligrams of anti-inflammatory hormone remaining in 27 devices after a certain number of hours (hrs) of wear. The data consists of two variables:

- Hormone level (response)
- Hours (predictor)

The data are shown in Figure @ref(fig:figchp101)

```
plot(hormone$hrs,hormone$amount)
```

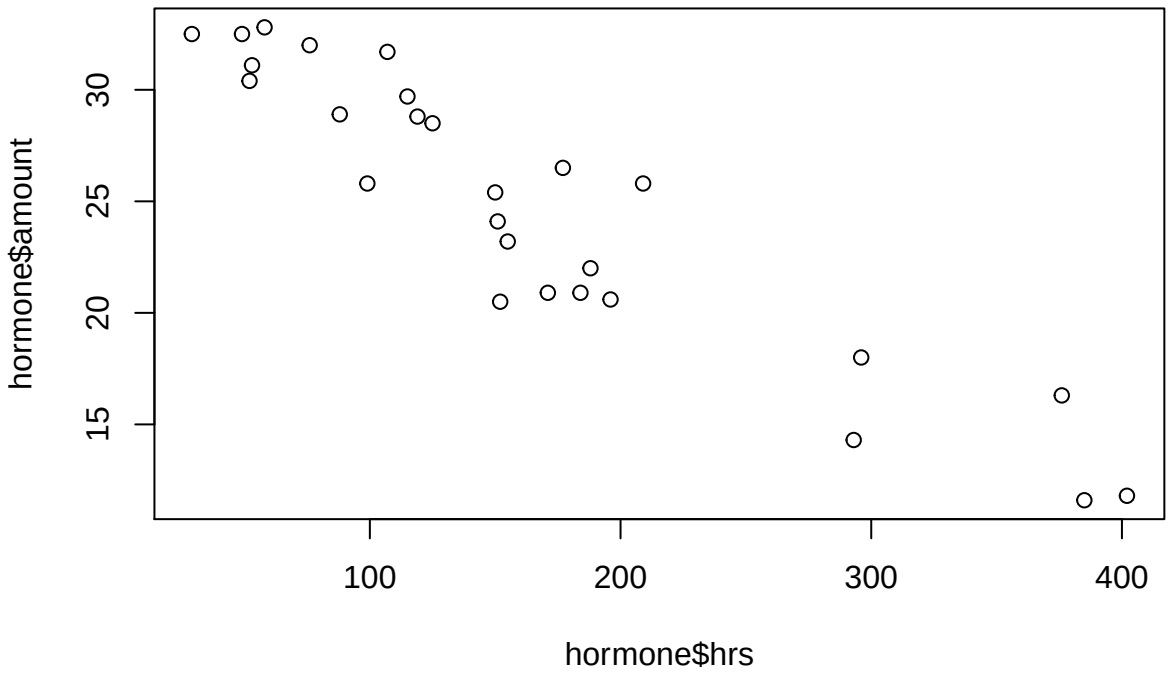


Figure 30: The hormone data.

Linear regression model

We fit a simple linear regreesion model with hours as predictor

```
n<-length(hormone$amount)
fit.lm<-lm(hormone$amount~hormone$hrs)
summary(fit.lm)
```

```
##
## Call:
## lm(formula = hormone$amount ~ hormone$hrs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9357 -1.7282 -0.0229  1.7388  3.7323
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.167528   0.867197   39.40  < 2e-16 ***
## hormone$hrs -0.057446   0.004464  -12.87 1.58e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.378 on 25 degrees of freedom
## Multiple R-squared:  0.8688, Adjusted R-squared:  0.8636
## F-statistic: 165.6 on 1 and 25 DF,  p-value: 1.584e-12
```

```
beta0<-summary(fit.lm)$coeff[1,1]
beta1<-summary(fit.lm)$coeff[2,1]
sigma<-2.378
```

Parameter estimates are equal to

```
c(beta0,beta1)
```

```
## [1] 34.1675282 -0.0574463
```

Data and predicated model are shown in Figure @ref(fig:figchp102)

```
plot(hormone$hrs,hormone$amount)
lines(hormone$hrs,fit.lm$fit)
```

Non parametric bootstrap for the hormone data - estimation and C.Is

To implement a non-parametric bootstrap procedure we need to resample pairs. This can be done using an index vectoras done in Section 4.4.3.

```
index<-c(1:n)
index
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## [26] 26 27
```

A non parametric bootstrap

```
B<-10000
beta0.b<-beta1.b<-c(1:B)
for (i in 1:B)
{
index.b<-sample(index,n,replace=TRUE)
hormone.b<-hormone[index.b,]
fit.lm.b<-lm(hormone.b$amount~hormone.b$hrs)
beta0.b[i]<-summary(fit.lm.b)$coeff[1,1]
beta1.b[i]<-summary(fit.lm.b)$coeff[2,1]
}
```

Figure 31: The hormone data and fitted model

The distributions of the bootstrap replicates are presented in Figure @ref(fig:figchp103a).

```
par(mfrow=c(1,2))
hist(beta0.b,nclass=50)
hist(beta1.b,nclass=50)
```

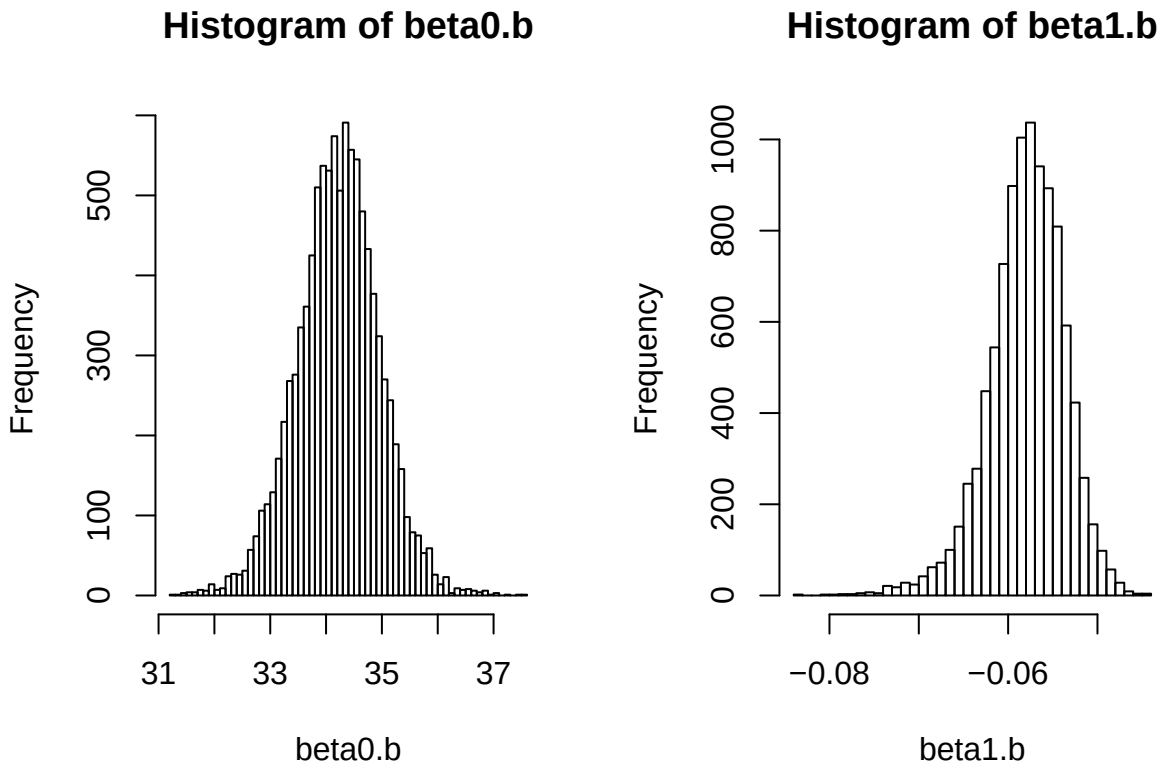


Figure 32: Distribution of the bootstrap replicates for the intercept and slope.

95% C.Is for β_0 and β_1. Note that to test the null hypothesis we need to calculate the bootstrap p value in the same way that we did in Chapter 6.

```
quantile(beta0.b,probs=c(0.025,0.975))
```

```
##     2.5%    97.5%
## 32.74655 35.67514
```

```
quantile(beta1.b,probs=c(0.025,0.975))
```

```
##        2.5%       97.5%
## -0.06754573 -0.05040092
```

Non parametric bootstrap for the hormone data - inference

Our aim is to test the null hypothesis $H_0 : \beta_1 = 0$ and therefore the resampling procedure should reflect the null hypothesis. For the current example, it implies that we should resample the response and keep the predictor fixed (this is in contrast with the resampling procedure in the previous section in which we resample

pairs).
To resample the bootstrap replicates for the response we use `sample(variable,n,replace=TRUE)`.

```
B<-10000
beta0.b<-beta1.b<-c(1:B)
for (i in 1:B)
{
hormone.b<-sample(hormone$amount,n,replace=TRUE)
fit.lm.b<-lm(hormone.b~hormone$hrs)
beta0.b[i]<-summary(fit.lm.b)$coeff[1,1]
beta1.b[i]<-summary(fit.lm.b)$coeff[2,1]
}
```

The distribution of the parameter estimate under H_0 is shown in Figure @ref(fig:figchp104). Note that to test the null hypothesis we need to calculate the bootstrap p value in the same way that as explained in Chapter 6.

```
par(mfrow=c(1,1))
hist(beta1.b,nclass=50,xlim=c(-0.065,0.065))
lines(c(beta1,beta1),c(0,1000),col=2)
```

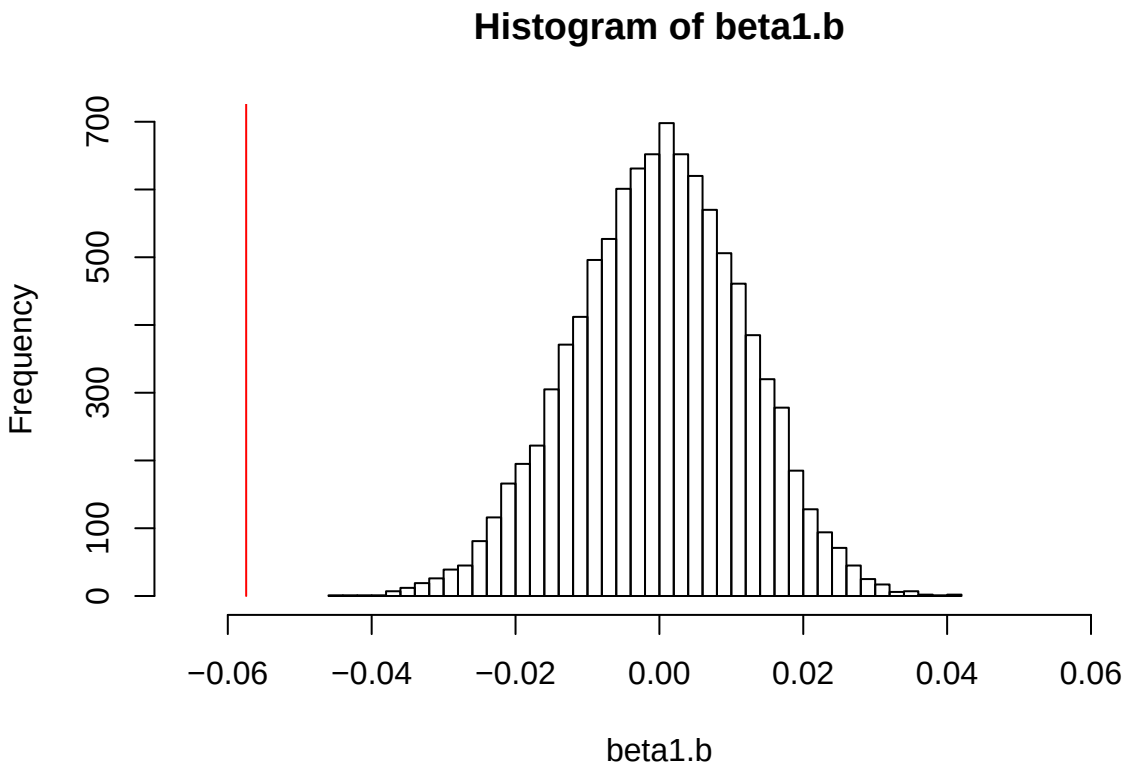


Figure 33: Non parametric bootstrap: bootstrap replicates under the null hypothesis.

Semi-parametric bootstrap: resampling residual

In this section we resample from the empirical distribution F_{e_i} which is an estimate to the unknown probability distribution F_{ε_i}. The semi-parametric bootstrap algorithm consists of the following steps:

- Fit the model

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i,$$

and estimate the unknown parameters by $\hat{\beta}_0$ and $\hat{\beta}_1$.

- Estimate the error term by the residuals:

$$\hat{\varepsilon}_i = e_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i).$$

- Bootstrap the residuals in the following way:
  - For $b = 1, \ldots, B$ resamples the residual vector and obtained the bootstrap sample

$$e_1^*, \ldots, e_n^*.$$

  - for B bootstrap samples we have

$$\begin{array}{cccc} e_1^*(1), & \ldots, & e_n^*(1) \\ e_1^*(2), & \ldots, & e_n^*(2) \\ \vdots & & \\ e_1^*(B), & \ldots, & e_n^*(B) \end{array}$$

* Calculate the bootstrap samples $(y^{*}_{1},x_{1}),\dots, y^{*}_{n},x_{n})$ by

$$y_i^* = \hat{\beta}_0 + \hat{\beta}_1 x_i + e_i^*.$$

* Fit the model

$$y_i^* = \beta_0 + \beta_1 x_i + u_i,$$

and obtained the bootstrap replicates $\hat{\beta}*_0$ and $\hat{\beta}*_1$

- Use the above bootstrap replicates for β to estimate the standard error, confidence intervals, distribution etc.

Semi-parametric bootstrap for the hormone data - estimation and C.Is

In the first step we fit the linear model and estimate the error (the vector ei).

```
y<-hormone$amount
x<-hormone$hrs
fit.lm <- lm(y ~ x)
ei <- fit.lm$resid
```

The semi parametric bootstrap loop consists of two steps: (1) calculate the bootstrap replicates for the response $y_1^*, \ldots, y_n^*$ and (2) fit the linear model
$y_i^* = \beta_0 + \beta_1 x_i + u_i$ and obtain the bootstrap replicates for the unknown parameters. In the code below, we resample from the residual vector using the function sample() and calculate $y_1^*, \ldots, y_n^*$ by fit.lm$coeff[1] + fit.lm$coeff[2]*x + e.boot.

```
n <- length(x)
B <- 1000
beta0.b <- beta1.b <- c(1:B)
for(i in 1:B) {
    e.boot <- sample(ei, size = n, replace = T)
    y.boot <- fit.lm$coeff[1] + fit.lm$coeff[2]*x + e.boot
    x.boot <- x
    fit.boot <- lm(y.boot ~ x.boot)
    beta0.b[i] <- fit.boot$coeff[1]
    beta1.b[i] <- fit.boot$coeff[2]
}
```

The distribution of the bootstrap replicates for intercept and slope are shown in Figure @ref(fig:figchp105).

```
par(mfrow=c(1,2))
hist(beta0.b,nclass=50)
hist(beta1.b,nclass=50)
```

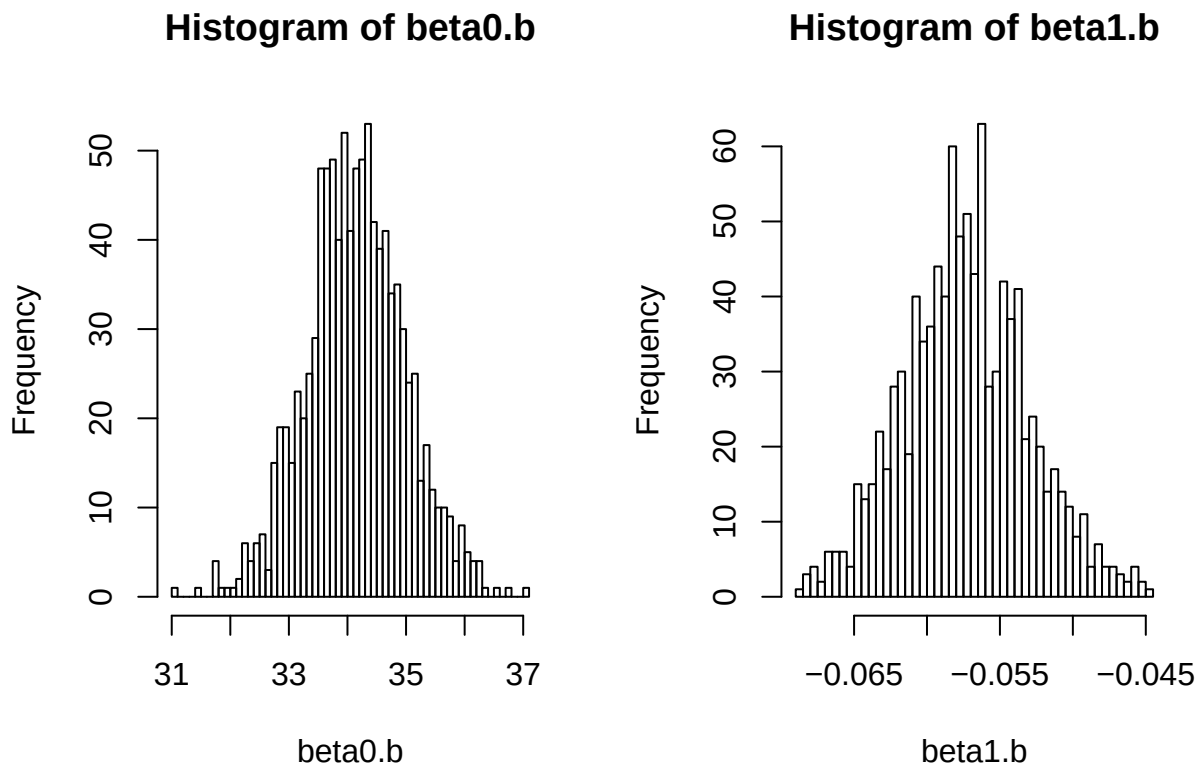


Figure 34: Semi-parametric bootstrap: distribution of the bootstrap replicates

95% C.Is for the intercept and slope.

```
quantile(beta0.b,probs=c(0.025,0.975))
```

```
##     2.5%    97.5%
## 32.45675 35.87903
```

```
quantile(beta1.b,probs=c(0.025,0.975))
```

```
##        2.5%       97.5%
## -0.06560279 -0.04841744
```

Semi-parametric bootstrap for the hormone data - inference

To implement the semi-parametric bootstrap procedure for inference, in the first step we fit the null model and calculate the residuals under the null, $e_{0,i}$ (the R object ei.0 bellow). Note that the null hypothesis $H_0 : \beta_1 = 0$ imlies that $y_i = \beta_0 + \varepsilon_i$.

```
fit.lm<-lm(hormone$amount~hormone$hrs)
beta0<-summary(fit.lm)$coeff[1,1]
beta1<-summary(fit.lm)$coeff[2,1]
y<-hormone$amount
x<-hormone$hrs
fit.lm.0 <- lm(y ~ 1)
summary(fit.lm.0)
```

```
##
## Call:
## lm(formula = y ~ 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.089  -3.939   1.111   5.361   8.111
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   24.689      1.239   19.92   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.439 on 26 degrees of freedom
```

```
ei.0 <- fit.lm.0$resid
```

The semi parametric bootstrap loop for inference ($H_0 : \beta_1 = 0$) consists of following steps:

- Resample the residuals vector ($e_{0,i}$, that were obtained under H_0). In the code below this setp is done in the line sample(ei.0, size = n, replace = T).
- Calculate the bootstrap replicates for the response $y_1^*, \ldots, y_n^*$ under the null hypothesis in the following way

$$y_i^* = \hat{\beta}_0 + e_{0,i}^*.$$

In the code below this setp is done in the line fit.lm.0$coeff[1] + e.boot. * Fit the model $y_i^* = \beta_0 + \beta_1 x_i + u_i$ and obtain the bootstrap replicates under H_0.

The code for the semi-parametric procedure is given below.

```
n <- length(x)
B <- 1000
beta0.b <- beta1.b <- c(1:B)
for(i in 1:B) {
    e.boot <- sample(ei.0, size = n, replace = T)
```

```
    y.boot <- fit.lm.0$coeff[1] + e.boot
    x.boot <- x
    fit.boot <- lm(y.boot ~ x.boot)
    beta0.b[i] <- fit.boot$coeff[1]
    beta1.b[i] <- fit.boot$coeff[2]
}
```

The distribution of the parameter estimates under H_0 is shown in Figure @ref(fig:figchp106).

```
par(mfrow=c(1,1))
hist(beta1.b,nclass=50,xlim=c(-0.065,0.065))
lines(c(beta1,beta1),c(0,1000),col=2)
```

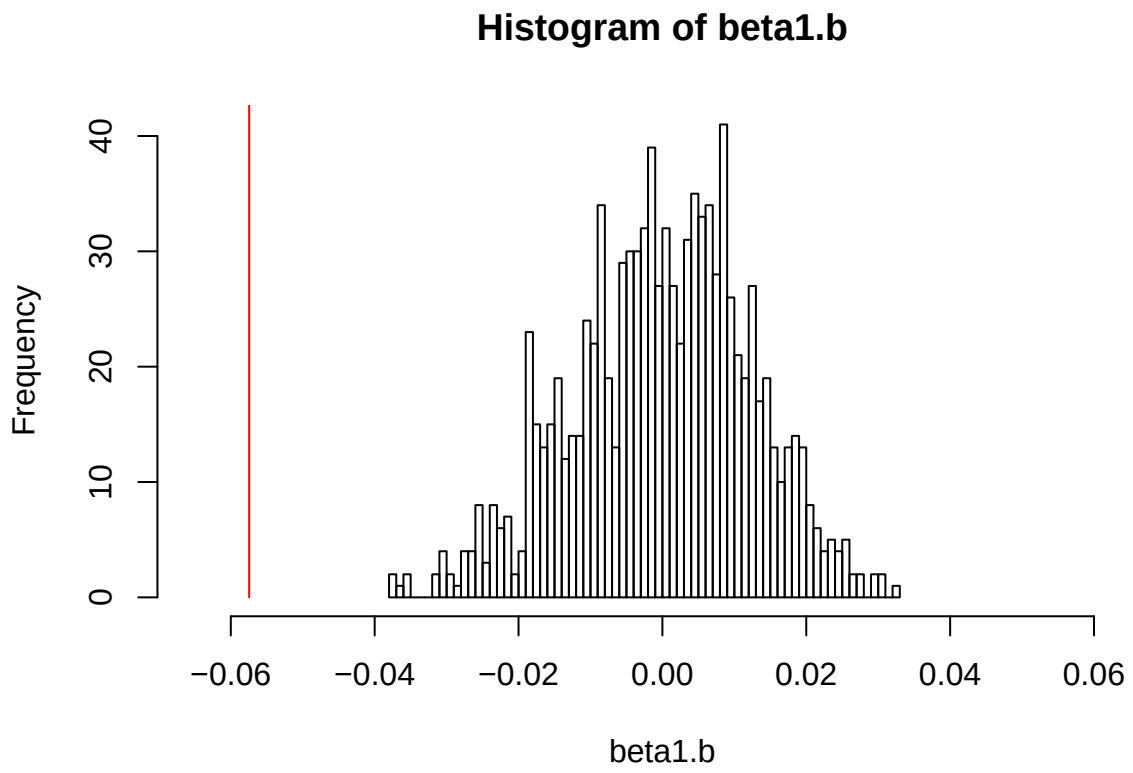


Figure 35: Semi-parametric bootstrap: distribution of the bootstrap replicates for the slopeunder the null hypothesis.

Parametric bootstrap

For parametric bootstrap, we resample from an empirical distribution F_{y_i} for which we make a an assumption about the parametric form, i.e., we assume that $y_i|x_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2)$. Since we do not know β_0, β_1 and σ^2 we replace them with their plug-in estimates. The parametric bootstrap algorithm can be implemented as follows.

- Fit the linear model

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i,$$

and estimate the unknown parameters by $\hat{\beta}_0$, $\hat{\beta}_1$ and $\hat{\sigma}^2$.

- For the bootstrap loop $b = 1, \ldots, B$ resamples the response from the empirical distribution

$$y_i^*|x_i \sim N(\hat{\beta}_0 + \hat{\beta}_1 x_i, \hat{\sigma}^2).$$

- for B bootstrap samples we have

$$\begin{array}{ccc} (y_1^*(1), x_1) & \ldots, & (y_n^*(1), x_n) \\ (y_1^*(2), x_1) & \ldots, & (y_n^*(2), x_n) \\ \vdots & & \\ (y_1^*(B), x_1) & \ldots, & (y_n^*(B), x_n) \end{array}$$

- For each bootstrap sample fit the model

$$y_i^* = \beta_0 + \beta_1 x_i + u_i,$$

and obtained the bootstrap replicates $\hat{\beta}*_0$ and $\hat{\beta}*_1$

- Use the above bootstrap replicates for β to estimate the standard error, confidence intervals, distribution etc.

Parametric bootstrap for the hormone data - estimation and C.Is

For parametrc bootstrap, we first fit the linear model and estimate the unknown parameters.

```
fit.lm<-lm(hormone$amount~hormone$hrs)
beta0<-summary(fit.lm)$coeff[1,1]
beta1<-summary(fit.lm)$coeff[2,1]
sigma<-2.378
```

Within the bootstrap loop we resample from $N(\hat{\beta}_0 + \hat{\beta}_1 x_i, \sigma^2)$ using the R function rnorm(n,mu,sigma). In the code below this is done using a nested for loop.

```
B<-10000
beta0.b<-beta1.b<-c(1:B)
amount.b<-c(1:n)
for (i in 1:B)
{
for(j in 1:n)
{
amount.b[j]<-rnorm(1,beta0+beta1*hormone$hrs[j],sigma)
}
fit.lm.b<-lm(amount.b~hormone$hrs)
beta0.b[i]<-summary(fit.lm.b)$coeff[1,1]
beta1.b[i]<-summary(fit.lm.b)$coeff[2,1]
}
```

Figure @ref(fig:figchp107) shows the distribution of the bootstrap replicates.

```
par(mfrow=c(1,2))
hist(beta0.b,nclass=50)
hist(beta1.b,nclass=50)
```

95% C.Is for β_0 and β_1.

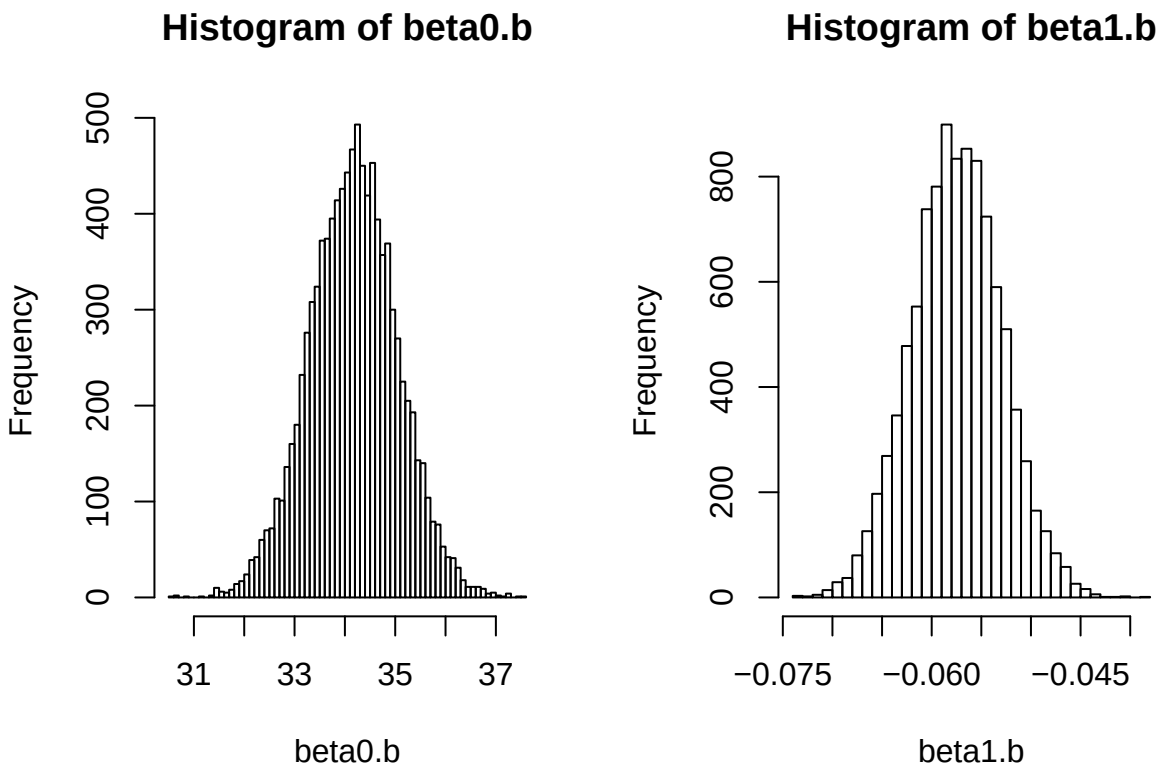


Figure 36: Parametric bootstrap: distribution of the bootstrap replicates.

```
quantile(beta0.b,probs=c(0.025,0.975))
```

```
##     2.5%    97.5%
## 32.43234 35.89116
```

```
quantile(beta1.b,probs=c(0.025,0.975))
```

```
##        2.5%       97.5%
## -0.06629443 -0.04845029
```

Parametric bootstrap for the hormone data - inference

A parametric bootstrap for inference consists of resampling the bootstrap replicates from a known distribution under the null hypothesis. In our example we resample from $N(\hat{\beta}_0, \sigma^2)$. The plug-in parameter estimates $\hat{\beta}_0$ and σ^2 obtained from the null model $y_i = \beta_0 + \varepsilon_i$.

```
fit.lm.0<-lm(hormone$amount~1)
summary(fit.lm.0)
```

```
##
## Call:
## lm(formula = hormone$amount ~ 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.089  -3.939   1.111   5.361   8.111
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   24.689      1.239   19.92   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.439 on 26 degrees of freedom
```

```
beta0<-summary(fit.lm)$coeff[1,1]
beta1<-summary(fit.lm)$coeff[2,1]
sigma<-sqrt(6.439)
c(beta0,sigma)
```

```
## [1] 34.167528  2.537518
```

Within the parametric bootstrap loop we resample the bootstrap replicate y_i^* from $N(\hat{\beta}_0, \sigma^2)$ using the function rnorm(1,beta0,sigma). Similar to Section 10.4.1 this is done using a double for loop.

```
B<-10000
beta0.b<-beta1.b<-c(1:B)
amount.b<-c(1:n)
for (i in 1:B)
{
for(j in 1:n)
{
amount.b[j]<-rnorm(1,beta0,sigma)
}
fit.lm.b<-lm(amount.b~hormone$hrs)
beta0.b[i]<-summary(fit.lm.b)$coeff[1,1]
```

```
beta1.b[i]<-summary(fit.lm.b)$coeff[2,1]
}
```

The distribution of the bootstrap replicates for the slope under the null hypothesis is shown in Figure @ref(fig:figchp108).

```
hist(beta1.b,nclass=50,xlim=c(-0.065,0.065))
lines(c(beta1,beta1),c(0,1000),col=2)
```

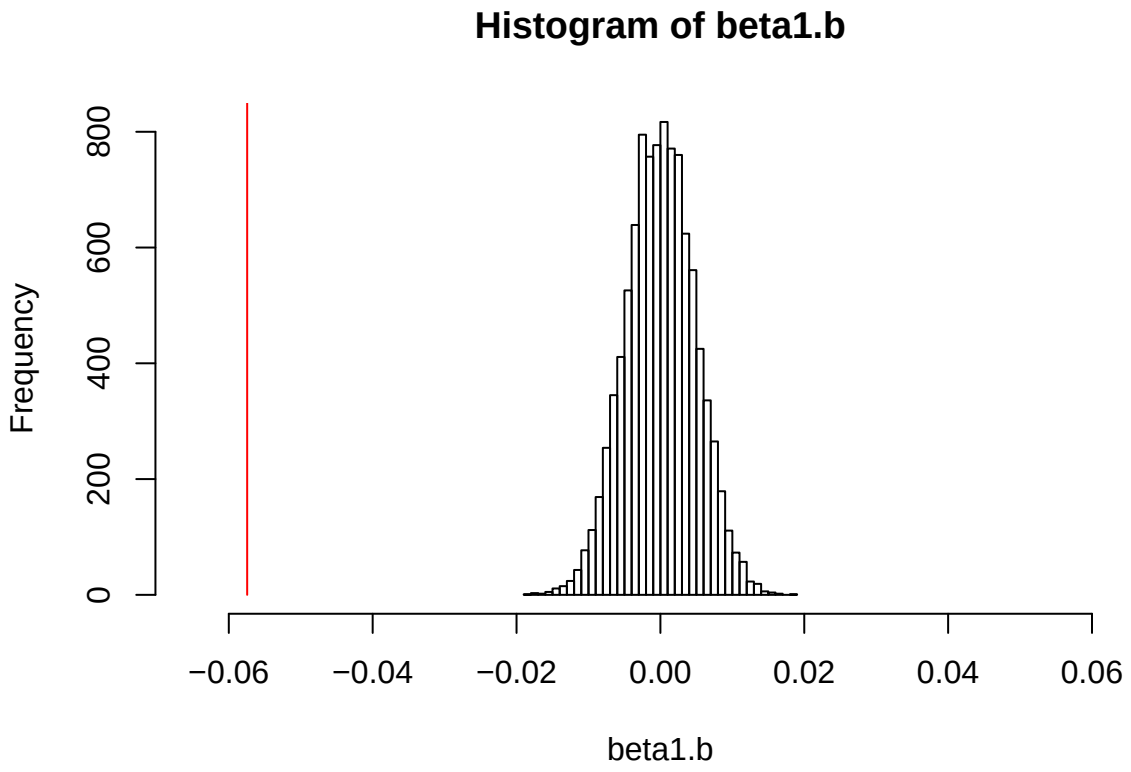


Figure 37: Parametric bootstrap: distribution of the bootstrap replicates under the null hypothesis.

Example: the cell survival data

The data

Fourteen cell plates were exposed to different levels of radiation. The observed response was the proportion of cells which survived the radiation exposure. The response in plate 13 was considered uncertain by the investigator. Data are shown in Figure @ref(fig:figchp109) and the data frame below.

```
library(bootstrap)
plot(cell$dose,cell$log.surv,ylim=c(-12,0))
```

```
y<-cell$log.surv
x<-cell$dose
data.frame(x,y)
```

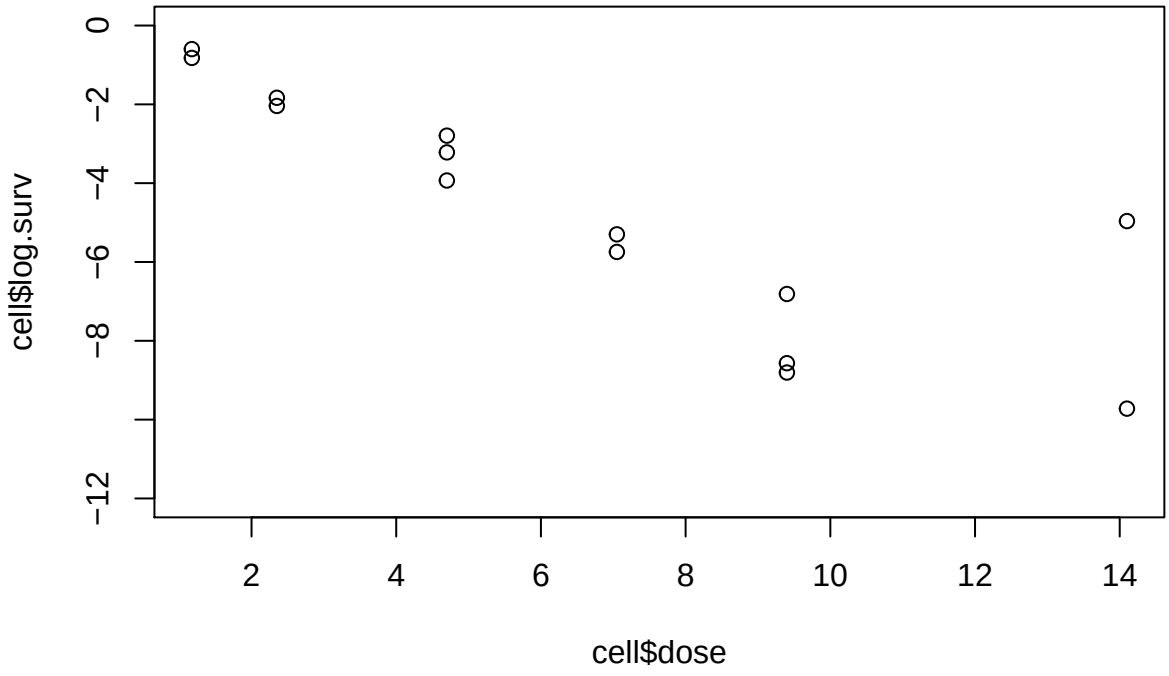


Figure 38: The cell survival data.

```
##         x      y
## 1   1.175 -0.821
## 2   1.175 -0.598
## 3   2.350 -1.833
## 4   2.350 -2.040
## 5   4.700 -3.219
## 6   4.700 -3.932
## 7   4.700 -2.794
## 8   7.050 -5.298
## 9   7.050 -5.745
## 10  9.400 -6.812
## 11  9.400 -8.805
## 12  9.400 -8.568
## 13 14.100 -4.962
## 14 14.100 -9.721
```

As mentioned above, there is uncertainty about the response for plate 13.

```
c(x[13],y[13])
```

```
## [1] 14.100 -4.962
```

```
x2<-x^2
x13<-x[-c(13)]
x132<-x13^2
y13<-y[-c(13)]
```

Quadratic models with and without observation 13 (n=13)

We fit two quadratic models to the data, with and without observation 13

$$Y_i = \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i.$$

Note that both models are fitted without an intercept.

```
plot(cell$dose,cell$log.surv,ylim=c(-12,0))
fit.lm2<-lm(y~-1+x+x2) #observation 13 in included
fit.lm2a<-lm(y13~-1+x13+x132) #observation 13 in not included
#summary(fit.lm2)
#summary(fit.lm2a)
lines(x,fit.lm2$fit,col=2)
lines(x13,fit.lm2a$fit,col=4)
```

Models for all data (n=14)

For the complete data, a linear and quadratic models are fitted.

$$y_i = \beta_1 x_i + \varepsilon_i,$$
$$y_i = \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i.$$

Data and fitted models are shown in Figure @ref(fig:figchp111).

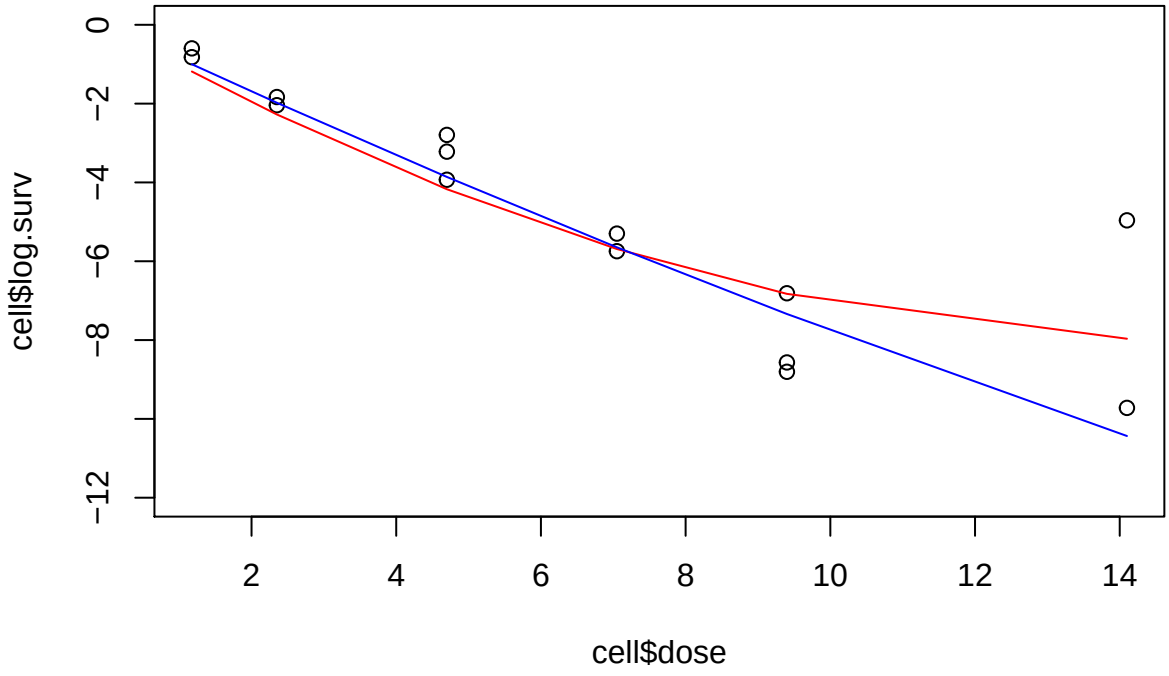


Figure 39: The cell survival data and fitted models (n=13)

```
plot(cell$dose,cell$log.surv,ylim=c(-12,0))
fit.lm1<-lm(y~-1+x)
fit.lm2<-lm(y~-1+x+x2)
#summary(fit.lm1)
#summary(fit.lm2)
lines(x,fit.lm1$fit)
lines(x,fit.lm2$fit,col=2)
```

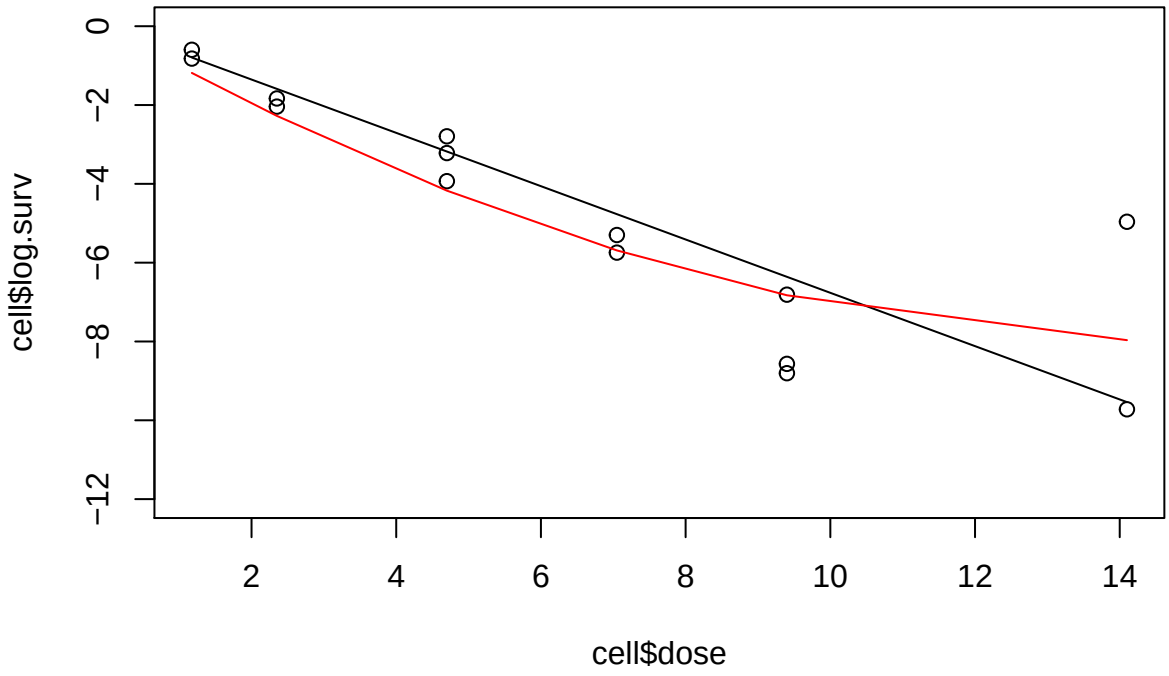


Figure 40: The cell survival data and fitted models (n=14)

Robust regression models for all data (n=14)

Three models are fitted to the complete data, a linear and quadratic regression models (estimated by OLS) and a robust regression model using the R package *rlm*. Data and fitted models are shown in Figure @ref(fig:figchp112).

```
library(rlm)
library(MASS)
```

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

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:rlm':
##
```

```
##     rlm
```

```
plot(cell$dose,cell$log.surv,ylim=c(-12,0))
fit.lm1<-lm(y~-1+x)
fit.lm2<-lm(y~-1+x+x2)
lines(x,fit.lm1$fit)
lines(x,fit.lm2$fit,col=2)
fit.rob<-rlm(y~-1+x)
#summary(fit.rob)
lines(x,fit.rob$fit,col=6)
```

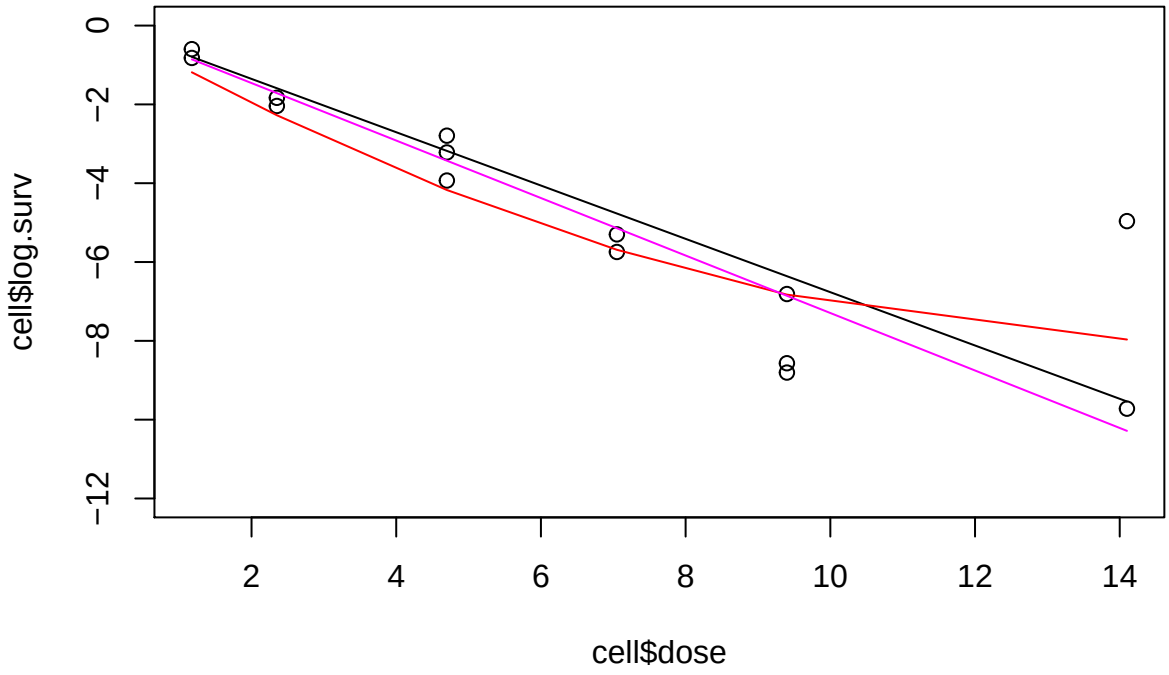


Figure 41: The cell survival data and fitted models (n=14)

Calculating the residuals scores

We define the residual by

$$y_i - \hat{\beta}_1 x_i,$$

and we calculate three residuals scores:

- The mean residuals sum of square.
- The median residual.
- The mean sum of absolute residuals.

We estimate β by minimizing the above scores using a grid search over a sequence of β.

```
beta<-seq(from=-0.9,to=-0.5,length=10000)
RAS<-RSS<-MSR<-c(1:10000)
n<-length(x)
for(i in 1:10000)
{
y.hat<-beta[i]*x
r<-(y-y.hat)
r.q<-r^2
RSS[i]<-(1/n)*sum(r.q)
MSR[i]<-median(r.q)
RAS[i]<-(1/n)*sum(abs(r))
}
m.RSS<-min(RSS)
m.MSR<-min(MSR)
m.RAS<-min(RAS)
beta.rss<-beta[RSS==m.RSS]
beta.msr<-beta[MSR==m.MSR]
beta.ras<-beta[RAS==m.RAS]
```

The OLS parameter estimate and the residual sum of squares score are shown in Figure @ref(fig:figchp113c).

```
plot(beta,RSS,type="l")
points(beta.rss,m.RSS,pch="+",cex=3)
```

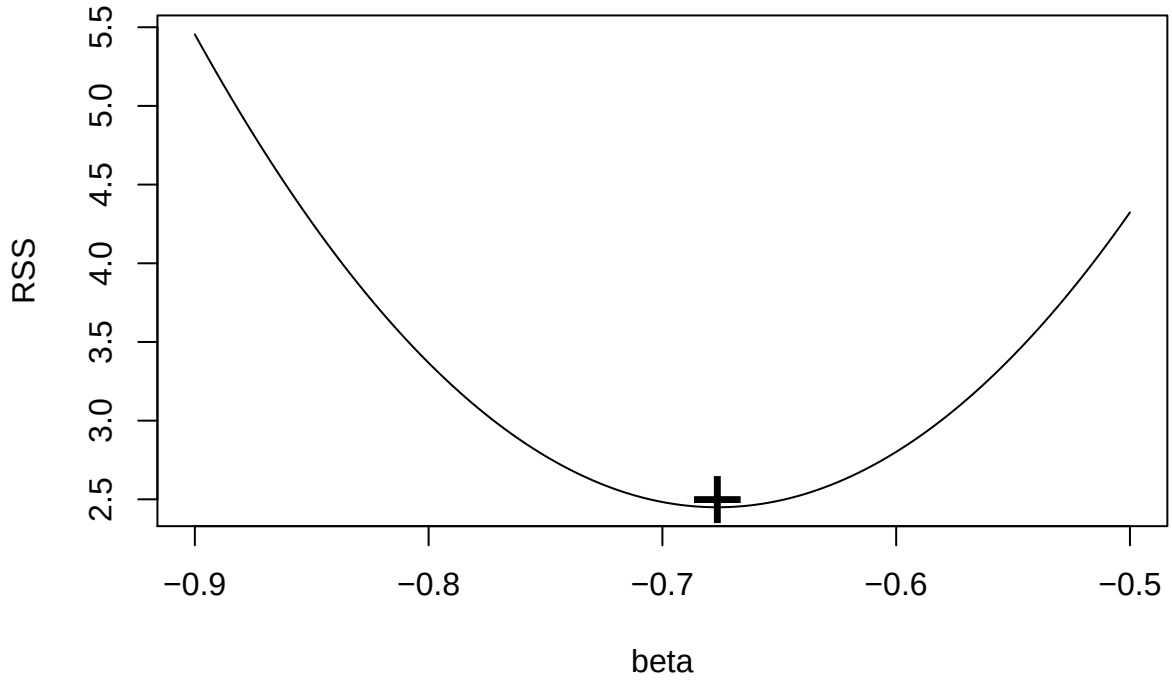


Figure 42: Residual sum of squares (OLS).

```
m.RSS
```

```
## [1] 2.449635
```

```
beta.rss
```

```
## [1] -0.6764976
```

The median residual parameter estimate and the median residuals score are shown in Figure @ref(fig:figchp113b). .

```
plot(beta,MSR,type="l")
points(beta.msr,m.MSR,pch="+",cex=3)
```

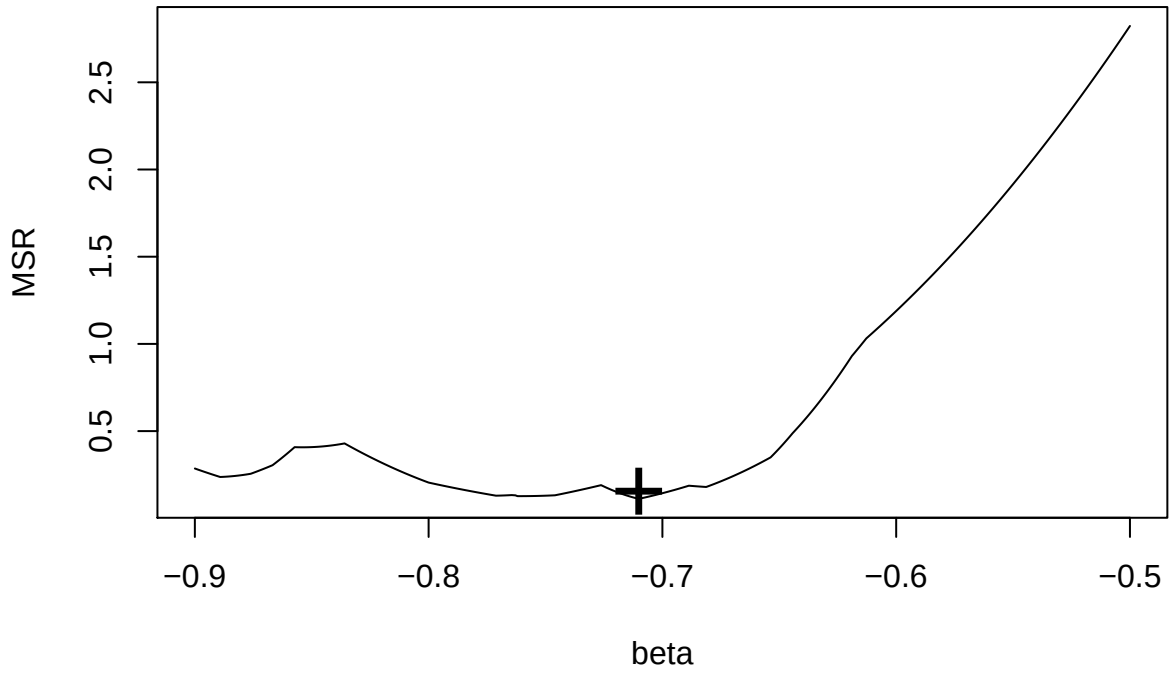


Figure 43: Median residuals

```
m.MSR
```

```
## [1] 0.1114881
```

```
beta.msr
```

```
## [1] -0.710101
```

The mean absolute residual parameter estimate and the median sum of absolute residuals score are shown in Figure @ref(fig:figchp114).

```
plot(beta,RAS,type="l")
points(beta.ras,m.RAS,pch="+",cex=3)
```

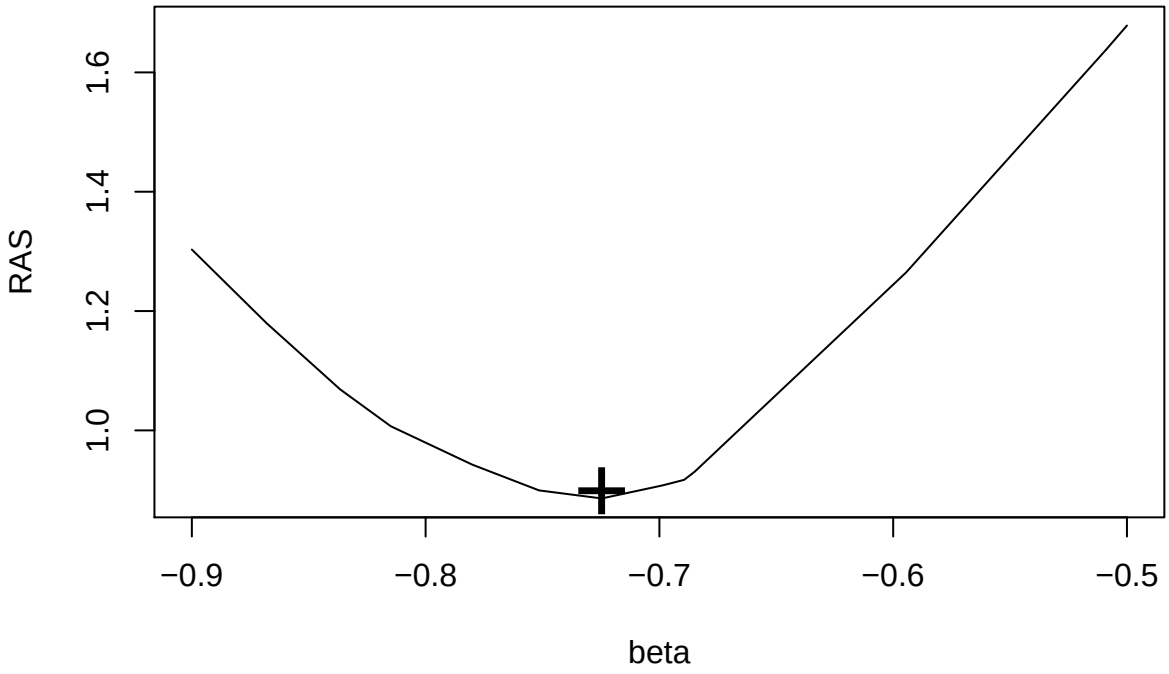


Figure 44: median sum of absolute residuals.

```
m.RAS
```

```
## [1] 0.8859395
```

```
beta.ras
```

```
## [1] -0.7247025
```

Figure @ref(fig:figchp115) shows the data and estimated models.

```
plot(x,y)
lines(unique(x),unique(x)*beta.rss,col=2)
lines(unique(x),unique(x)*beta.msr,col=3)
lines(unique(x),unique(x)*beta.ras,col=4)
legend(10,-2,c("OLS","MSR","RAS"),lty=c(1,1,1),col=c(2,3,4))
```

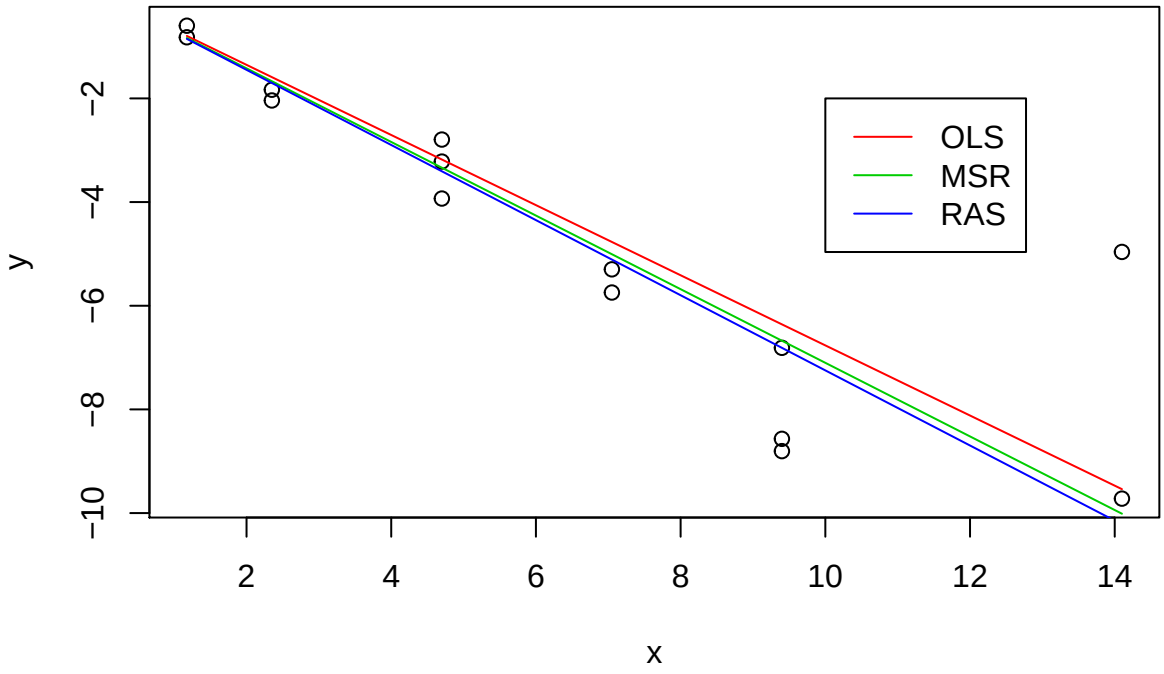


Figure 45: data and fitted models

Parameter estimates and residuals scores for the three models are shown below.

```
mm<-c(m.RSS,m.MSR,m.RAS)
beta.i<-c(beta.rss,beta.msr,beta.ras)
data.frame(mm,beta.i)
```

```
##          mm     beta.i
## 1 2.4496353 -0.6764976
## 2 0.1114881 -0.7101010
## 3 0.8859395 -0.7247025
```

Generelized linear models

GLM: The setting

Example 1: dose-response models for binary data

This section is focused on an example about dose-response modeling for binary data. We use logistic regression models and discuss parametric bootstrap procedure implemented to construct a confidence intervals for the parameters of interest and for the predicted probabilities.

The beetle data

The beetle data consists of a dose response experiment in which beetles were expose to an increasing does of a chemical compound (the predictor). The response is the number of animals that died during the experiment. Data are shown below. The R object beetles is the number of animals per dose level.

```
Dose<-c(1.6907,1.7242,1.7552,1.7842,1.8113,1.8369,1.8610,1.8839)
beetles<-c(59,60,62,56,63,59,62,60)
killed<-c(6,13,18,28,52,53,61,60)
beetle<-data.frame(Dose,beetles,killed)
beetle
```

```
##     Dose beetles killed
## 1 1.6907      59      6
## 2 1.7242      60     13
## 3 1.7552      62     18
## 4 1.7842      56     28
## 5 1.8113      63     52
## 6 1.8369      59     53
## 7 1.8610      62     61
## 8 1.8839      60     60
```

Logistic regression model for the beetle data

Let y_j be the number of beetle died at does level d_j, $y_j \sim B(n_j, \pi(d_j))$. We assume that the probability to die in the experiment is does dependent,

$$\pi(d_j) = \frac{e^{\alpha+\beta d_j}}{1+e^{\alpha+\beta d_j}}.$$

For a logit link function

$$log\left(\frac{\pi}{1-\pi}\right),$$

the linear predictor is given by

$$\eta = \alpha + \beta d_j.$$

The above model is a logistic regression model, i.e., a GLM for binary data with a logit link function. In R, the model can be fitted using the glm() function. Data and fitted model are shown in Figure @ref(fig:figchp111a).

```
p.b<-killed/beetles
unkilled<-beetles-killed
Proportionkilled<-p.b
par(mfrow=c(1,1))
plot(Dose,Proportionkilled, main="Proportion of the killed beetles",ylim=c(0,1))
fit.beetles<-glm(cbind(killed,unkilled)~Dose,family=binomial(link = "logit"))
lines(Dose,fit.beetles$fit)
```

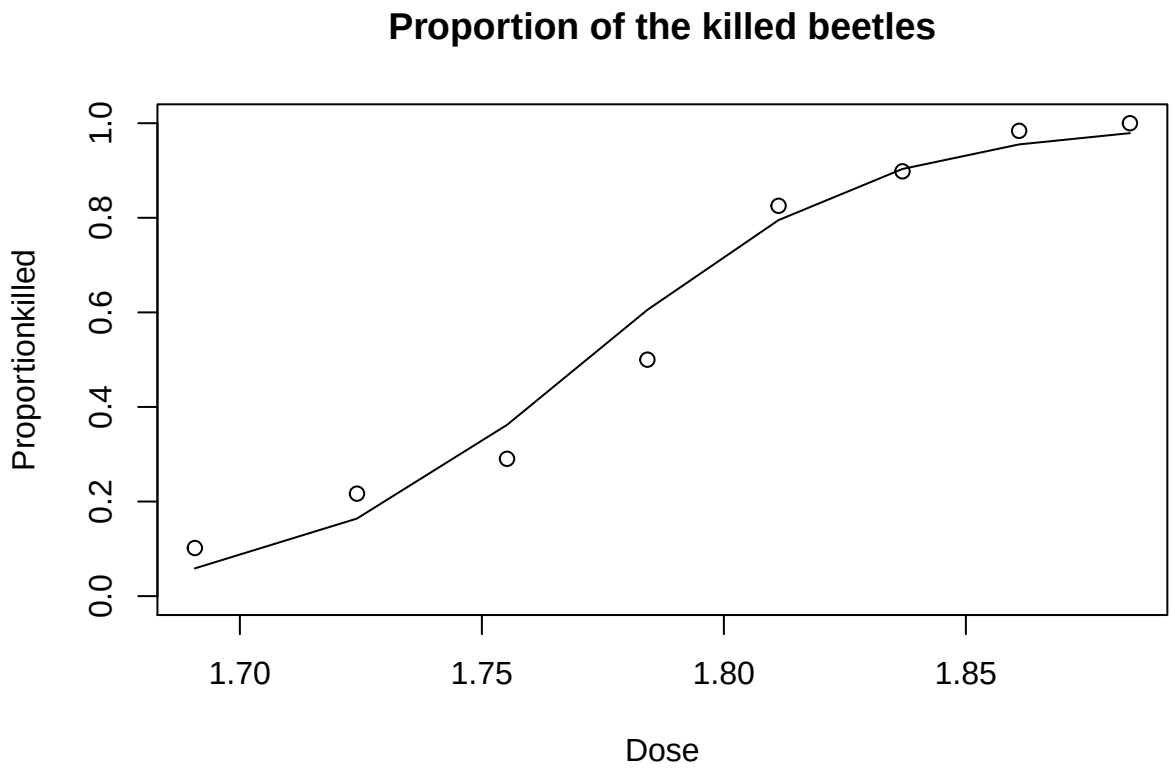


Figure 46: The Beetle data.

Parameter estimates are whon in the panel below.

```
summary(fit.beetles)$coefficients
```

```
##             Estimate Std. Error   z value     Pr(>|z|)
## (Intercept) -60.71745   5.180701 -11.71993 1.007549e-31
## Dose         34.27033   2.912134  11.76811 5.698445e-32
```

Parametric bootstrap: estimation

A parametric bootstrap consists of resampling for $B(n_j, \pi(d_j))$. Since the probability is unknown we use the sample proportions (the R object p.d).

```
p.b<-beetle$killed/beetle$beetles
p.b
```

```
## [1] 0.1016949 0.2166667 0.2903226 0.5000000 0.8253968 0.8983051 0.9838710
## [8] 1.0000000
```

Note that then bootstrap procedure is implemented using a double for loop. In the inner loop we resample the bootstrap replicates (y^*) to create the bootstrap samples

$$y_j^* \; simB(n_j, \hat{\pi}(d_j)),$$

for which the corresponding R code is given by pos.boot[i]<-sum(rbinom(beetles[i],1,p.b[i])). In the outer loop we resample B bootstrap samples of y^*

```
B<-1000
prob.boot<-matrix(0,8,B)
test.stat<-coeff.boot<-matrix(0,2,B)
pos.boot<-c(1:8)
for(b in 1:B) #outer loop
{
for(i in 1:8) #inner loop
{
pos.boot[i]<-sum(rbinom(beetles[i],1,p.b[i]))
}             # end of inner loop
neg.boot<-beetles-pos.boot
fit.boot<-glm(cbind(pos.boot,neg.boot)~Dose,family=binomial(link = "logit"))
prob.boot[,b]<-fit.boot$fit
coeff.boot[,b]<-fit.boot$coefficients
test.stat[,b]<-summary(fit.boot)$coefficients[,3]
}              # end of outer loop
```

Figure @ref(fig:figchp112a) shows the distribution of the bootstrap replicates of $\hat{\alpha}$ and $\hat{\beta}$.

```
par(mfrow=c(1,2))
hist(coeff.boot[1,],main="alpha",nclass=25)
lines(c(fit.beetles$coefficients[1],fit.beetles$coefficients[1]),c(0,550),
      col=2,lwd=2)
hist(coeff.boot[2,],main="beta",nclass=25)
lines(c(fit.beetles$coefficients[2],fit.beetles$coefficients[2]),c(0,300),
      col=2,lwd=2)
```

95% percentile intervals for α and β are given, respectively, by

```
quantile(coeff.boot[1,], probs = c(0.025, 0.975))
```

```
##      2.5%     97.5%
## -72.46830 -52.22517
```

```
quantile(coeff.boot[2,], probs = c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 29.56506 40.86814
```

We can construct a prediction interval for $\hat{\pi}(d_j)$ using the bootstrap replicates for the prediction values (the R object prob.boot) shown in Figure @ref(fig:figchp113a).

```
par(mfrow=c(1,1))
plot(Dose,Proportionkilled, main="Proportion of the killed beetles",ylim=c(0,1))
lines(Dose,fit.beetles$fit)
for(b in 1:B)
{
```

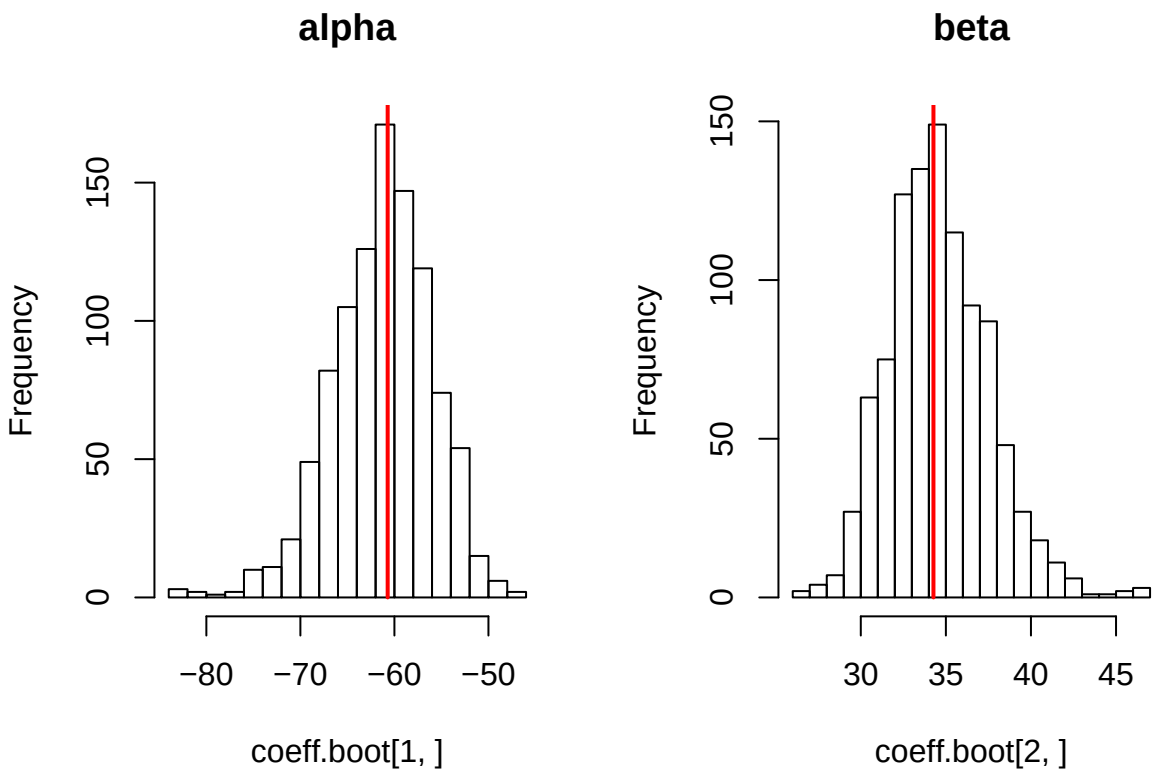


Figure 47: Bootstrap replicates for alpha and beta

```
lines(Dose,prob.boot[,b],col=3)
}
lines(Dose,fit.beetles$fit,col=1,lwd=3)
points(Dose,Proportionkilled)
```

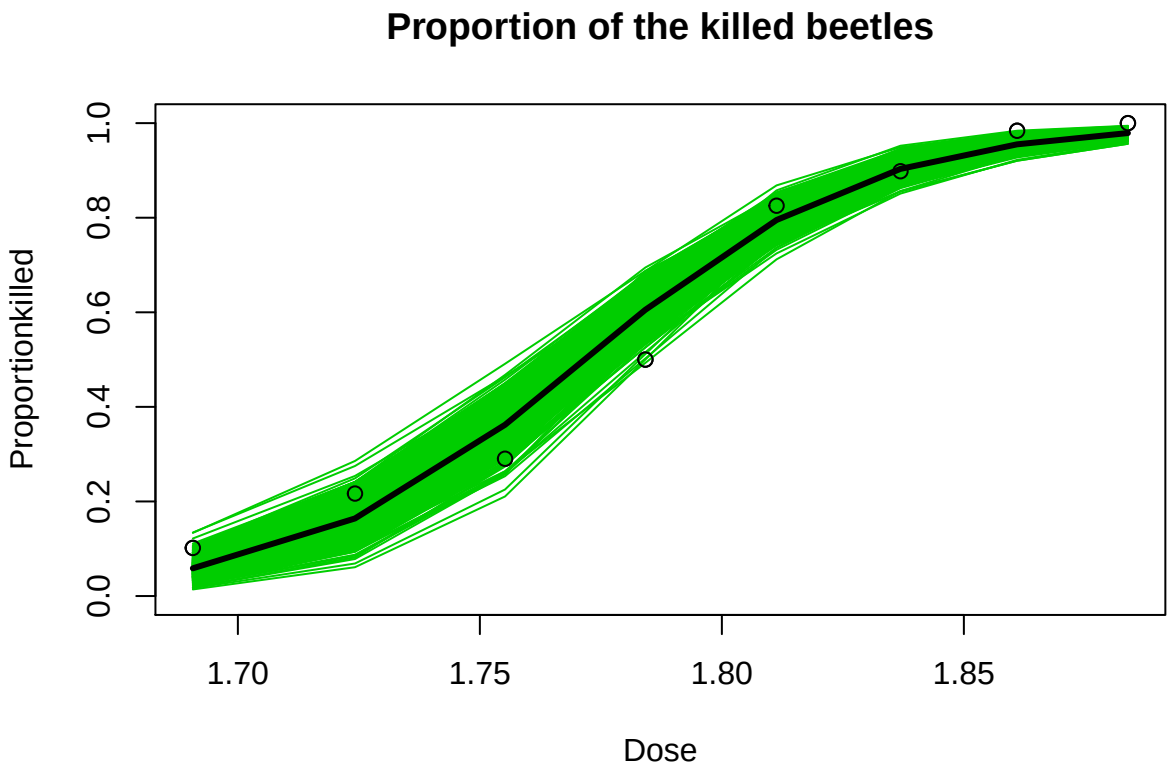


Figure 48: The beetle data and bootstrap replicates for the fitted model

Example: the malaria data

In this section we discuss parametric bootstrap procedures for binomial data. In Section 11.3.2 we focus on estimation and C.I while Section 11.3.3 is devoted to inference.
The malaria data gives information about the results of serological test for Malaria and consists of three variables, the age group of the subject, the number of seropositive (posi) and the total number of subjects in each age group (ntot).

```
agei<-c(1.5,4.0,7.5,12.5,17.5,25.0,35.0,47.0,60)
posi<-c(8,6,18,14,20,39,19,25,44)
ntot<-c(123,132,182,140,138,161,133,92,74)
negi<-ntot-posi
cbind(agei,posi,negi)
```

```
##       agei posi negi
##  [1,]  1.5    8  115
##  [2,]  4.0    6  126
##  [3,]  7.5   18  164
```

```
##  [4,] 12.5   14  126
##  [5,] 17.5   20  118
##  [6,] 25.0   39  122
##  [7,] 35.0   19  114
##  [8,] 47.0   25   67
##  [9,] 60.0   44   30
```

Logistic regression for the malaria data

Let y_j be the number of individlas that were infected at age a_j, $y_j \sim B(n_j, \pi(a_j))$. Here $\pi(a_j)$ is the probability of infection (the prevalence). Our aim is to model the probability of infection which is assumed to be age dependent

$$\pi(a) = \frac{e^{\alpha+\beta\times age}}{1+e^{\alpha+\beta\times age}}.$$

For a model with a logit link function, the linear predictor is given by

$$\eta = \alpha + \beta \times age.$$

In R the model is fitted using the following code:

```
fit.malaria1<-glm(cbind(posi,negi)~agei,family=binomial(link = "logit"))
```

Figure @ref(fig:figchp114a) shows the data and estimated prevalence.

```
summary(fit.malaria1)
```

```
##
## Call:
## glm(formula = cbind(posi, negi) ~ agei, family = binomial(link = "logit"))
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.78685  -1.31863  -0.05053   0.66752   2.38275
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.714074   0.151740 -17.886   <2e-16 ***
## agei         0.044672   0.004511   9.904   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 124.037  on 8  degrees of freedom
## Residual deviance:  21.865  on 7  degrees of freedom
## AIC: 66.388
##
## Number of Fisher Scoring iterations: 4
```

```
plot(agei,posi/ntot)
lines(agei,fit.malaria1$fit)
```

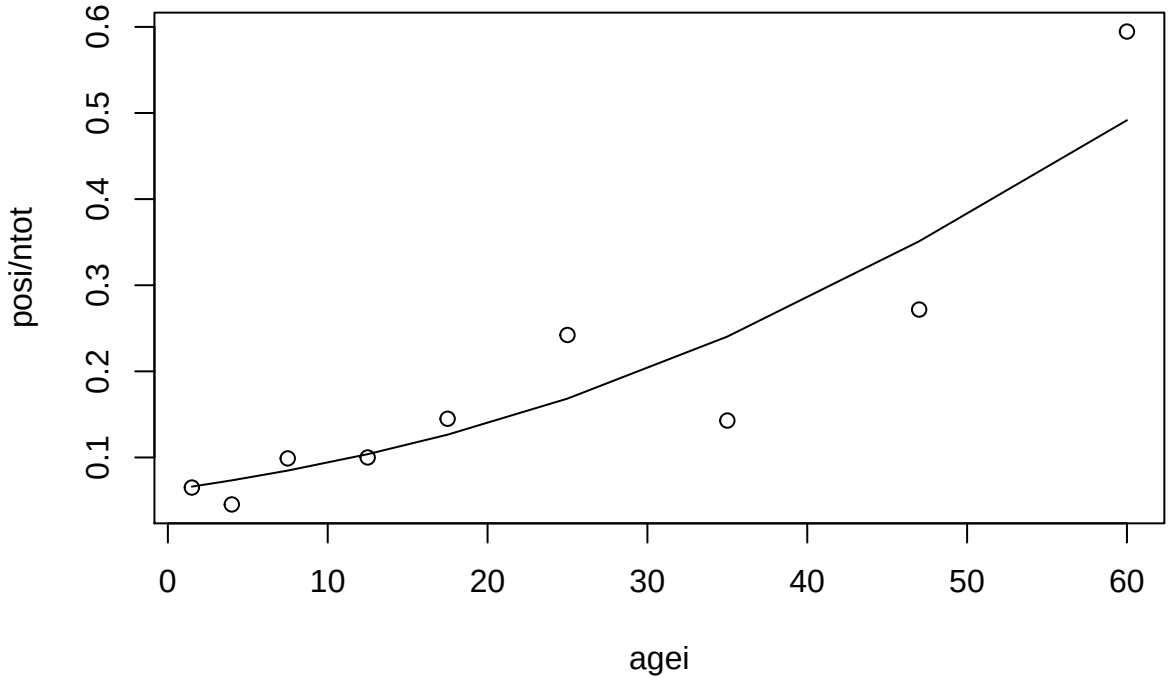


Figure 49: The malaria data and fitted model

Parametric bootstrap: estimation and C.Is

A parametric bootstrap for this example is similar to the procedure implement for the dose-response model in Section 11.2. Note that the bootstrap procedure is implemented using a double for loop in which the bootstrap replicates are sampled in the inner loop.

```
p.b<-posi/ntot
B<-1000
prob.boot<-matrix(0,9,B)
test.stat<-coeff.boot<-matrix(0,2,B)
pos.boot<-c(1:9)
for(b in 1:B)
{
for(i in 1:9)
{
pos.boot[i]<-sum(rbinom(ntot[i],1,p.b[i]))
}
neg.boot<-ntot-pos.boot
fit.boot<-glm(cbind(pos.boot,neg.boot)~agei,family=binomial(link = "logit"))
prob.boot[,b]<-fit.boot$fit
coeff.boot[,b]<-fit.boot$coefficients
test.stat[,b]<-summary(fit.boot)$coefficients[,3]
}
```

The distribution of the bootstrap replicates and 95% C.Is are shown in Figure @ref(fig:figchp115a)

```
par(mfrow=c(1,2))
hist(coeff.boot[1,],main="alpha",nclass=25)
lines(c(fit.malaria1$coefficients[1],fit.malaria1$coefficients[1]),c(0,550),
      col=2,lwd=2)
hist(coeff.boot[2,],main="beta",nclass=25)
lines(c(fit.malaria1$coefficients[2],fit.malaria1$coefficients[2]),c(0,300),
      col=2,lwd=2)
```

95% percentile intervals for α and β are given, respectively, by

```
quantile(coeff.boot[1,], probs = c(0.025, 0.975))
```

```
##      2.5%     97.5%
## -3.006058 -2.428175
```

```
quantile(coeff.boot[2,], probs = c(0.025, 0.975))
```

```
##       2.5%      97.5%
## 0.03595143 0.05324394
```

Bootstrap replicates for the predicted values are whown in Figure @ref(fig:figchp116a).

```
par(mfrow=c(1,1))
plot(agei,p.b, ylab="sero prevalence",ylim=c(0,1))
lines(agei,fit.malaria1$fit)
for(b in 1:B)
{
lines(agei,prob.boot[,b],col=3)
}
lines(agei,fit.malaria1$fit,col=1,lwd=3)
points(agei,p.b)
```

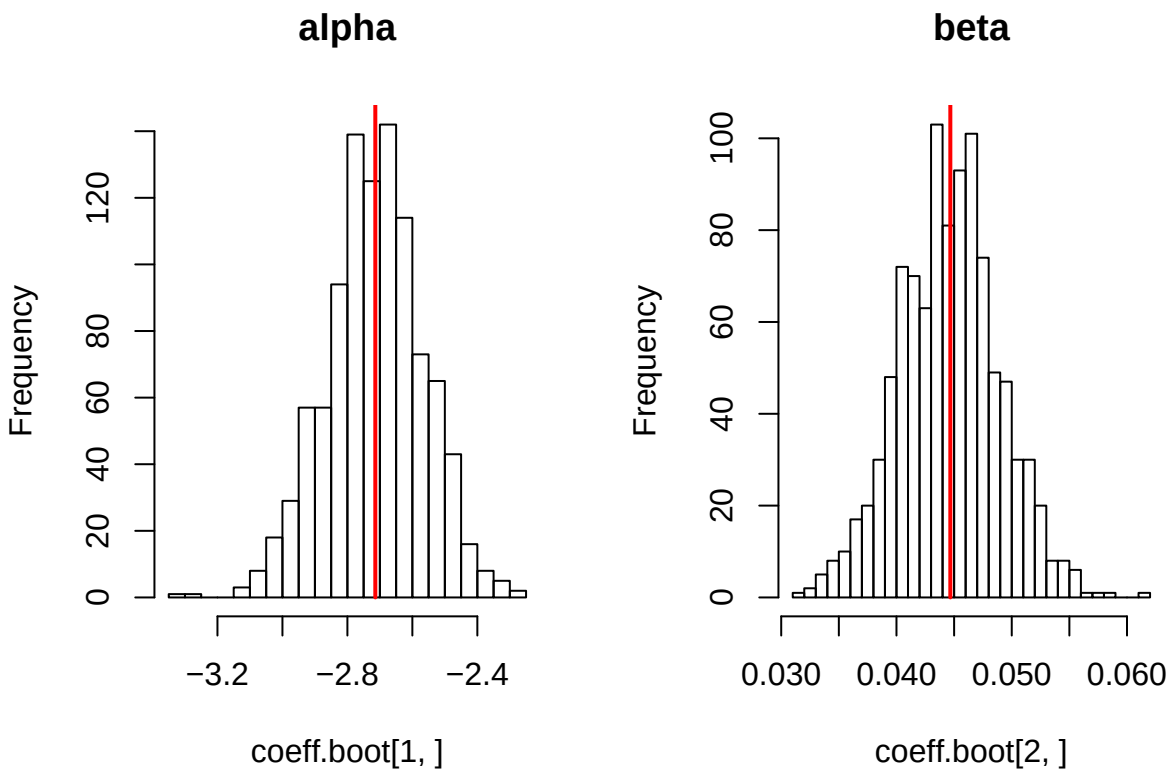


Figure 50: Distribution of the bootstrap replicates for alpha and beta

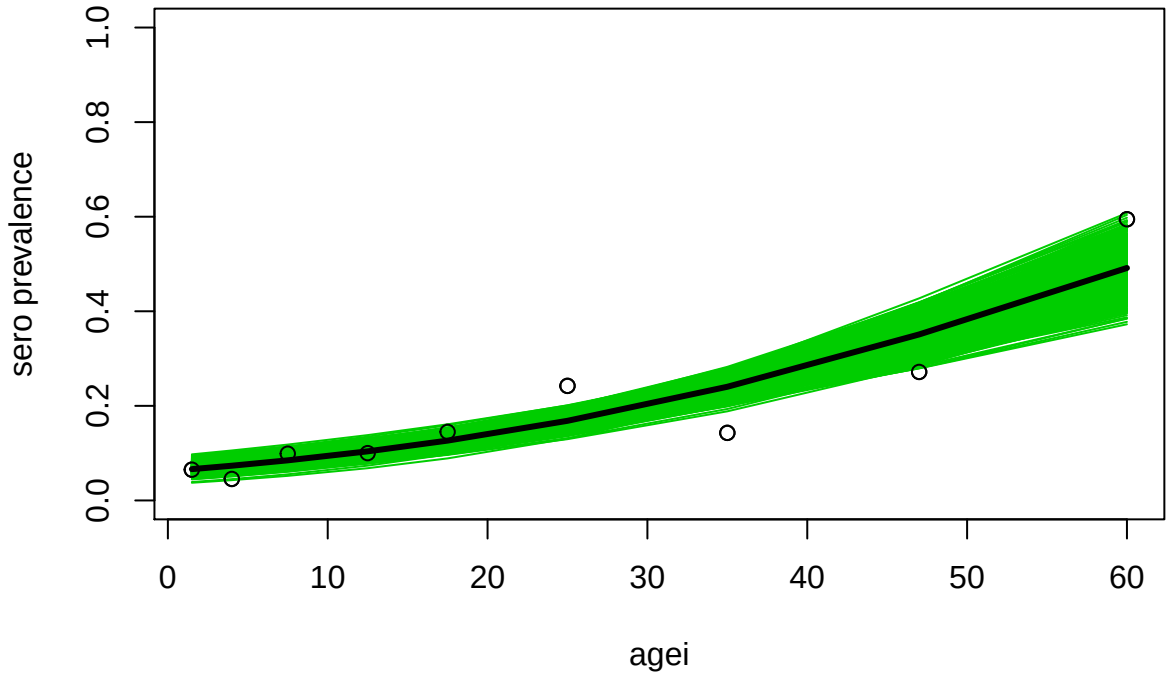


Figure 51: The malaria data and the bootstrap replicates for the fitted model

Parametric bootstrap: inference

Suppose that we would like to test the hypothesis that age has not effect of the probability of infection. That is

$$H_0 : \beta = 0.$$

The null model (of no age effect) can be fitted using the following code

```
fit.malaria0<-glm(cbind(posi,negi)~1,family=binomial(link = "logit"))
summary(fit.malaria0)
```

```
##
## Call:
## glm(formula = cbind(posi, negi) ~ 1, family = binomial(link = "logit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.2517  -2.5363  -0.6784   2.5303   8.3548
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.62690    0.07874  -20.66   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 124.04  on 8  degrees of freedom
## Residual deviance: 124.04  on 8  degrees of freedom
## AIC: 166.56
##
## Number of Fisher Scoring iterations: 4
```

The probability of infection inder H_0 is equal at each dose level (i.e., the overall prevalence in teh sample) is equal to 0.164.

```
eta<- -1.62690
prob.i<-exp(eta)/(1+exp(eta))
prob.i<-rep(prob.i,9)
prob.i
```

```
## [1] 0.1642555 0.1642555 0.1642555 0.1642555 0.1642555 0.1642555 0.1642555
## [8] 0.1642555 0.1642555
```

Note that these are the fitted values obtained for the null model.

```
fit.malaria0$fit
```

```
##         1         2         3         4         5         6         7         8
## 0.1642553 0.1642553 0.1642553 0.1642553 0.1642553 0.1642553 0.1642553 0.1642553
##         9
## 0.1642553
```

The overall proportion of infection can be seen in Figure @ref(fig:figchp116b).

```
plot(agei,posi/ntot)
lines(agei,prob.i,col=2)
```

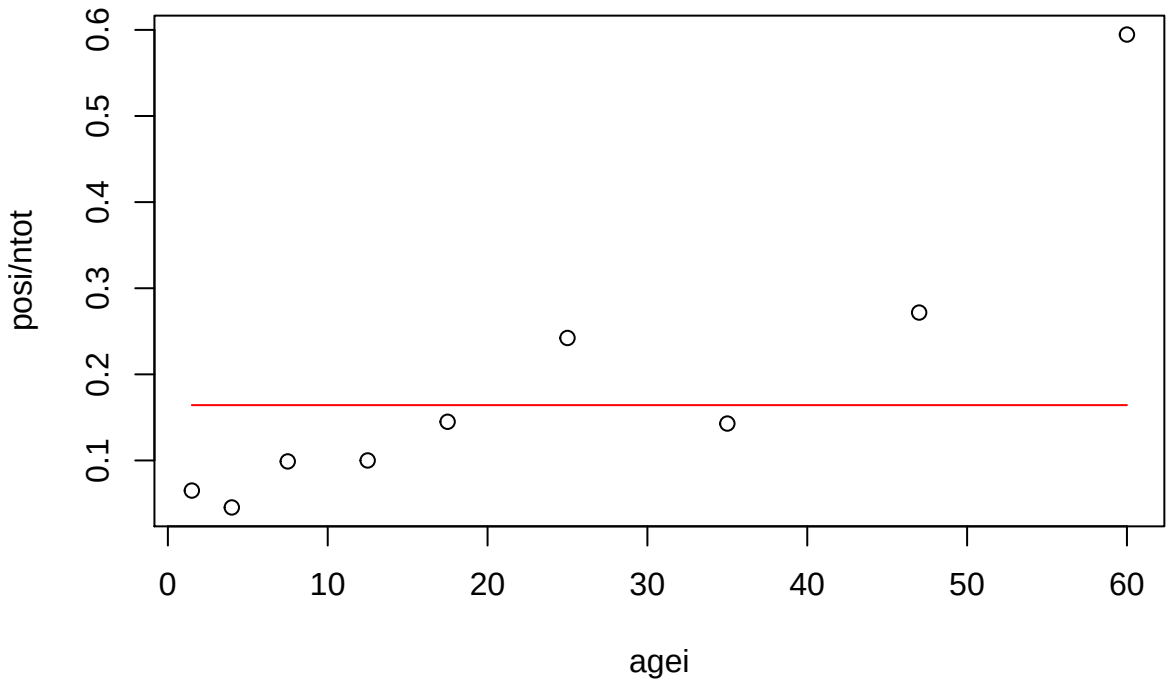


Figure 52: The probabbiliy of infection under the null hypothesis

The parametric bootstrap consists of resampling from a binomial distribution under H_0,

$$y_j^* \sim B(n_j, \pi_{H_0}).$$

With π_{H_0} =prob.i=0.164...

```
B<-1000
prob.boot<-matrix(0,9,B)
test.stat<-coeff.boot<-matrix(0,2,B)
pos.boot<-c(1:9)
for(b in 1:B)
{
for(i in 1:9)
{
pos.boot[i]<-sum(rbinom(ntot[i],1,prob.i[i]))
}
neg.boot<-ntot-pos.boot
fit.boot<-glm(cbind(pos.boot,neg.boot)~agei,family=binomial(link = "logit"))
prob.boot[,b]<-fit.boot$fit
coeff.boot[,b]<-fit.boot$coefficients
test.stat[,b]<-summary(fit.boot)$coefficients[,3]
}
```

The distributions of bootstrap replicates for $\hat{\beta}_b^*$ and the teststatistic $t_{\beta,b}^*$ with the observed values are shown in Figure @ref(fig:figchp117a). The monte carlo p value, in both cases, is equal to $\frac{1}{B+1}$ and we reject the null hypothesis.

```
par(mfrow=c(1,2))
hist(coeff.boot[2,],main="beta",xlim=c(-0.025,0.05))
lines(c(fit.malaria1$coefficients[2],fit.malaria1$coefficients[2]),c(0,300),
      col=2,lwd=2)
hist(test.stat[2,],main="t-beta",xlim=c(-10,10))
lines(c(9.903614,9.903614),c(0,300),
      col=2,lwd=2)
```

Example: the stress data

In this section we focus on application of parametric and non parametric bootstrap procedures for estimation and inference for a Poisson regression model.

The data and fitted model

The stress data gives information about 41 events which had occurred within the last a period of 18 months. The data consist of two variables: the response is the number of respondents that experienced at least one event (by month) and the predictor id the month. Data are shown in the panel below.

```
month<-c(1:18)
respondents<-c(15,11,14,17,5,11,10,4,8,10,7,9,11,3,6,1,1,4)
stress<-data.frame(month,respondents)
stress
```

```
##    month respondents
## 1      1          15
## 2      2          11
```

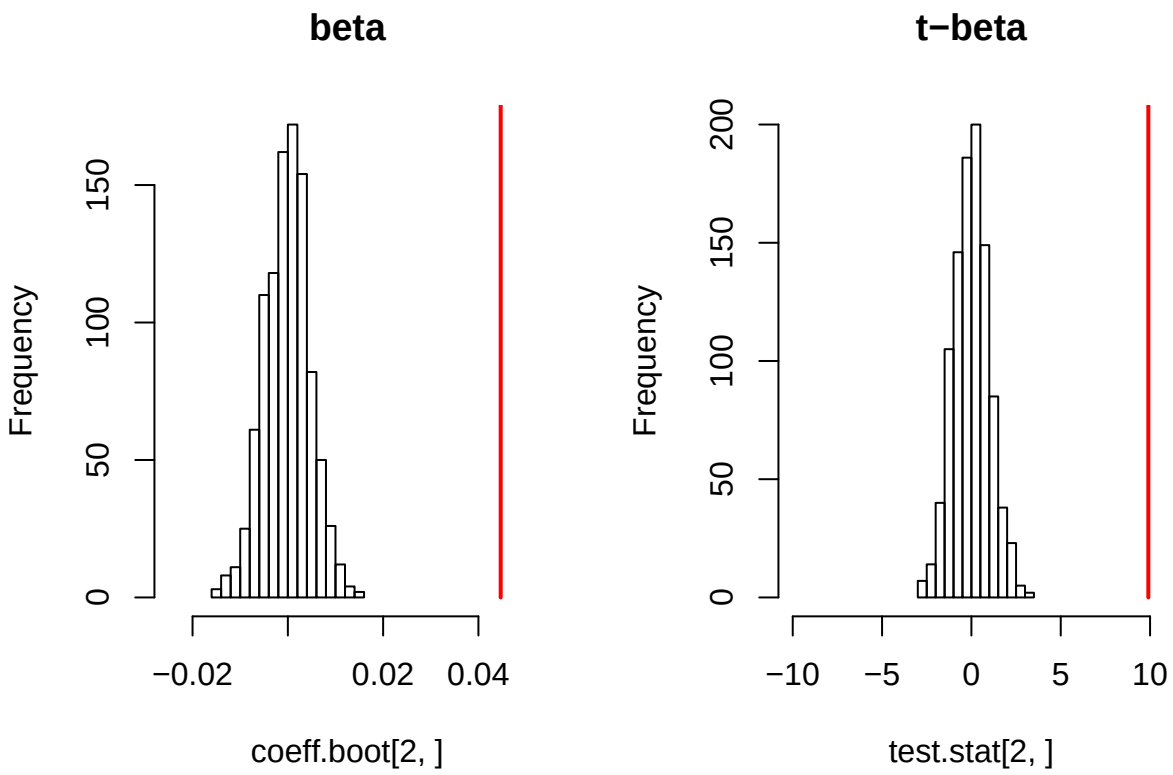


Figure 53: The distribution of the bootstrap replicates and test statistics under the null hypothesis.

```
## 3      3          14
## 4      4          17
## 5      5           5
## 6      6          11
## 7      7          10
## 8      8           4
## 9      9           8
## 10    10          10
## 11    11           7
## 12    12           9
## 13    13          11
## 14    14           3
## 15    15           6
## 16    16           1
## 17    17           1
## 18    18           4
```

Poisson regression for the stress data

Let y_i be the number of respondents that experienced at least one event in the ith month. Since the number of respondents is a count data we assume a Poisson distribution, that is

$$y_i \sim Poisson(\mu_i).$$

Here, μ_i is the mean number of respondents at the ith month. We formulate a model with log link function with a linear predictor given by

$$log(\mu_i) = \alpha + \beta \times x_i.$$

Here, x_i is the month. The above model can be fitted in R using the glm() function with the argument family=poisson. Data and predicted model are shown in Figure @ref(fig:figchp118a).

```
respGLM <- glm(respondents ~ month, family=poisson, data=stress)
summary(respGLM)
```

```
##
## Call:
## glm(formula = respondents ~ month, family = poisson, data = stress)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9886  -0.9631   0.1737   0.5131   2.0362
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.80316    0.14816  18.920  < 2e-16 ***
## month       -0.08377    0.01680  -4.986 6.15e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 50.843  on 17  degrees of freedom
```

```
## Residual deviance: 24.570  on 16  degrees of freedom
## AIC: 95.825
##
## Number of Fisher Scoring iterations: 5
```

```
plot(respondents ~ month, xlab = "Months",ylab = "Subjects", xlim=c(0,20), ylim=c(0,20))
lines(month,respGLM$fit)
```

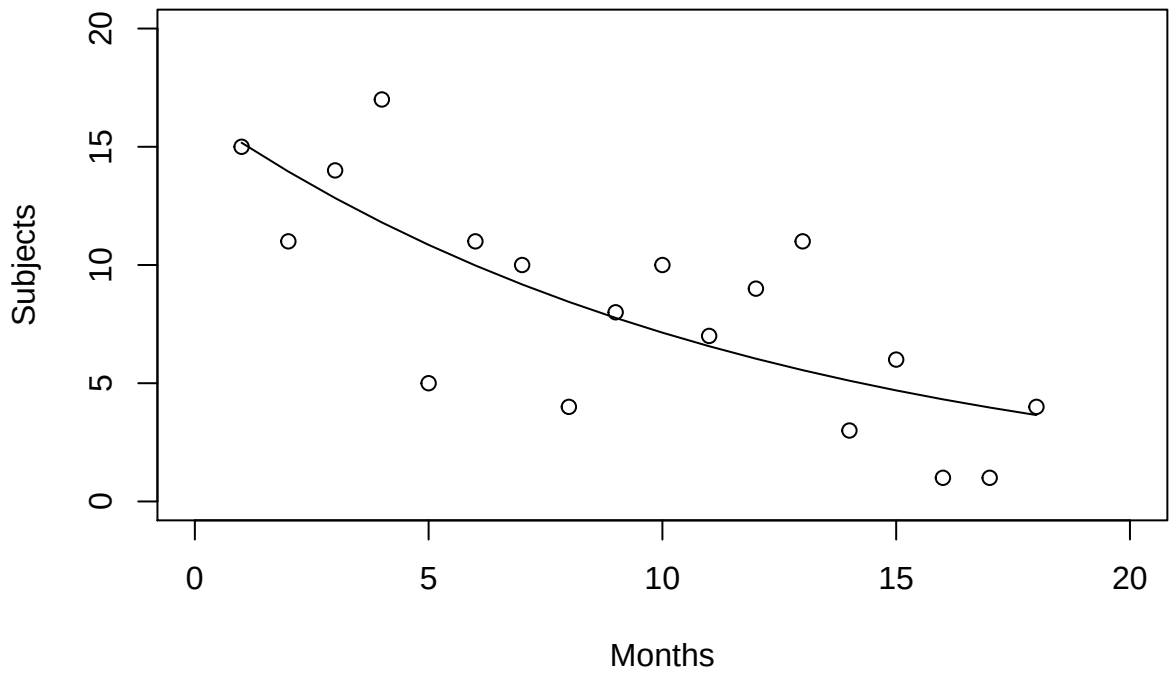


Figure 54: The stress data and fitted model.

Parameter estimates are shown below.

```
summary(respGLM)
```

```
##
## Call:
## glm(formula = respondents ~ month, family = poisson, data = stress)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9886  -0.9631   0.1737   0.5131   2.0362
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.80316    0.14816  18.920  < 2e-16 ***
## month       -0.08377    0.01680  -4.986 6.15e-07 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 50.843  on 17  degrees of freedom
## Residual deviance: 24.570  on 16  degrees of freedom
## AIC: 95.825
##
## Number of Fisher Scoring iterations: 5
```

Non parametric bootstrap: Estimation and C.Is

A non parametric bootstrap procedure with the aim to construct C.Is requires to resample pairs. In the code below this is done using an index vector. Note that at each bootstrap iteration, the bootstrap sample is stress.b<-stress[index.b,].

```
n<-length(stress[,1])
index<-c(1:n)
B<-10000
beta0.b<-beta1.b<-c(1:B)
for (i in 1:B)
{
index.b<-sample(index,n,replace=TRUE)
stress.b<-stress[index.b,]
fit.glm.b<-glm(stress.b$respondents ~ stress.b$month, family=poisson)
beta0.b[i]<-summary(fit.glm.b)$coeff[1,1]
beta1.b[i]<-summary(fit.glm.b)$coeff[2,1]
}
```

Figure @ref(fig:figchp119a) ahows the distribution of the bootstrap replicates and 95% C.Is for the parameters.

```
par(mfrow=c(1,2))
hist(beta0.b,nclass=50)
hist(beta1.b,nclass=50)
```

95% percentile intervals for α and β are given, respectively, by

```
quantile(beta0.b,probs=c(0.025,0.975))
```

```
##     2.5%    97.5%
## 2.485419 3.110579
```

```
quantile(beta1.b,probs=c(0.025,0.975))
```

```
##        2.5%       97.5%
## -0.12530637 -0.04609202
```

Non parametric bootstrap: inference

In this section we would like to test the null hypothesis that the month has no effect on the number of events,

$$H_0 : \beta = 0.$$

For a non parametric bootstrap, this implies that we need to resample under the null hypothesis, i.e. we need to resample the months and the respondents separately.

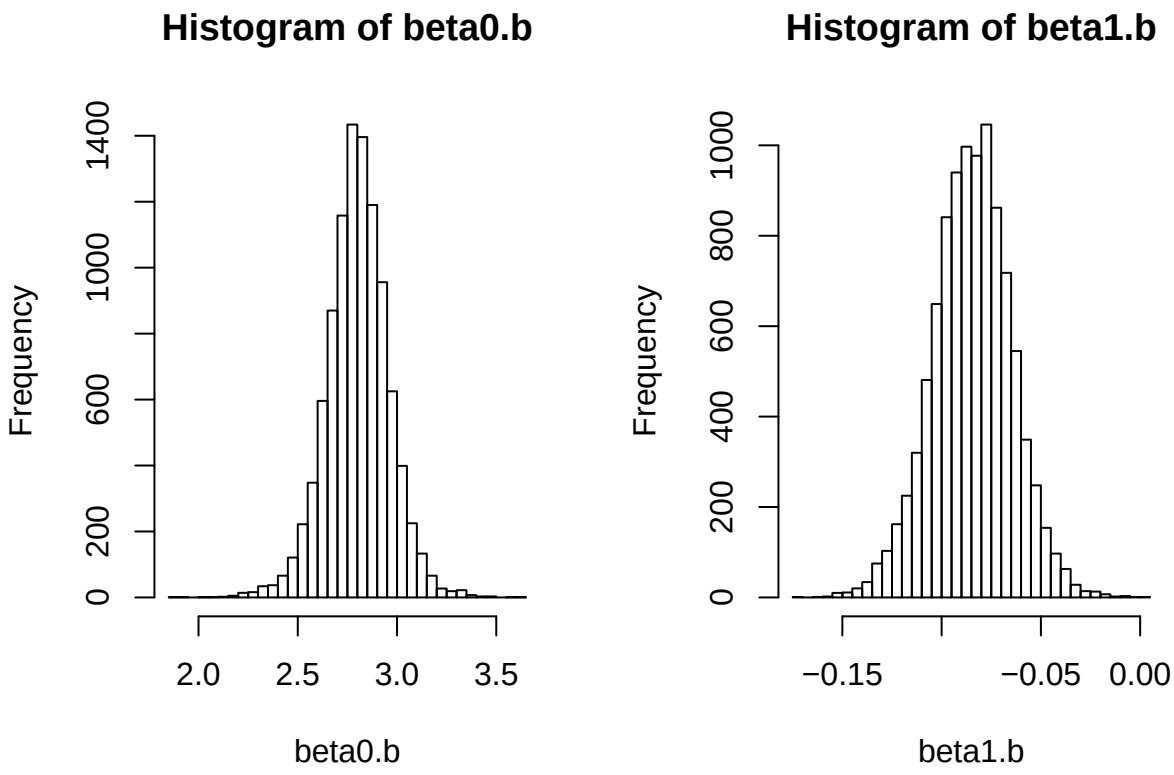


Figure 55: The distribution of the bootstrap repliacres for the intercept and slope.

```
attach(stress)
```

```
## The following objects are masked _by_ .GlobalEnv:
##
##     month, respondents
```

```
n<-length(stress[,1])
B<-10000
beta0.b<-beta1.b<-c(1:B)
for (i in 1:B)
{
#stress.b<-stress[index.b,]
respondents.b<-sample(respondents, size=n, replace = TRUE)
month.b<-sample(month,size=n, replace = TRUE)
fit.glm.b<-glm(respondents.b ~ month.b, family=poisson)
beta0.b[i]<-summary(fit.glm.b)$coeff[1,1]
beta1.b[i]<-summary(fit.glm.b)$coeff[2,1]
}
```

As a test statistic we use $\hat{\beta}$. The observed slope is equal to -0.08377.

```
respGLM <- glm(respondents ~ month, family=poisson, data=stress)
beta1.obs<-summary(respGLM)$coeff[2,1]
print(beta1.obs)
```

```
## [1] -0.08376906
```

The distribution of the test statistic under the null is shown in Figure @ref(fig:figchp1110).

```
par(mfrow=c(1,1))
hist(beta1.b,nclass=50)
lines(c(beta1.obs,beta1.obs),c(0,1000),col=2)
```

The Monte carlo p value is smaller than 0.05, and the null hypothesis is rejected.

```
(1+sum(abs(beta1.b)>abs(beta1.obs)))/(B+1)
```

```
## [1] 0.0039996
```

Parametric bootstrap: inference

For a parametric bootstrap, we need to resample the bootstrap replicates for y from a Poisson distribution taking the null hypothesis into account,

$$y_i \sim Poisson(\mu_{H_0,i}).$$

Here, μ_{H_0} is the mean under H_0. For an estimate of the mean under the null we use the sample mean $\hat{\mu} = \bar{x} = 8.166667$.

```
hat.mu<-mean(respondents)
hat.mu
```

```
## [1] 8.166667
```

At each bootstrap iteration, we use the function rpois(n,hat.mu) to resample the bootstrap replicates. Note that, in this acse, we fixed the values of the months.

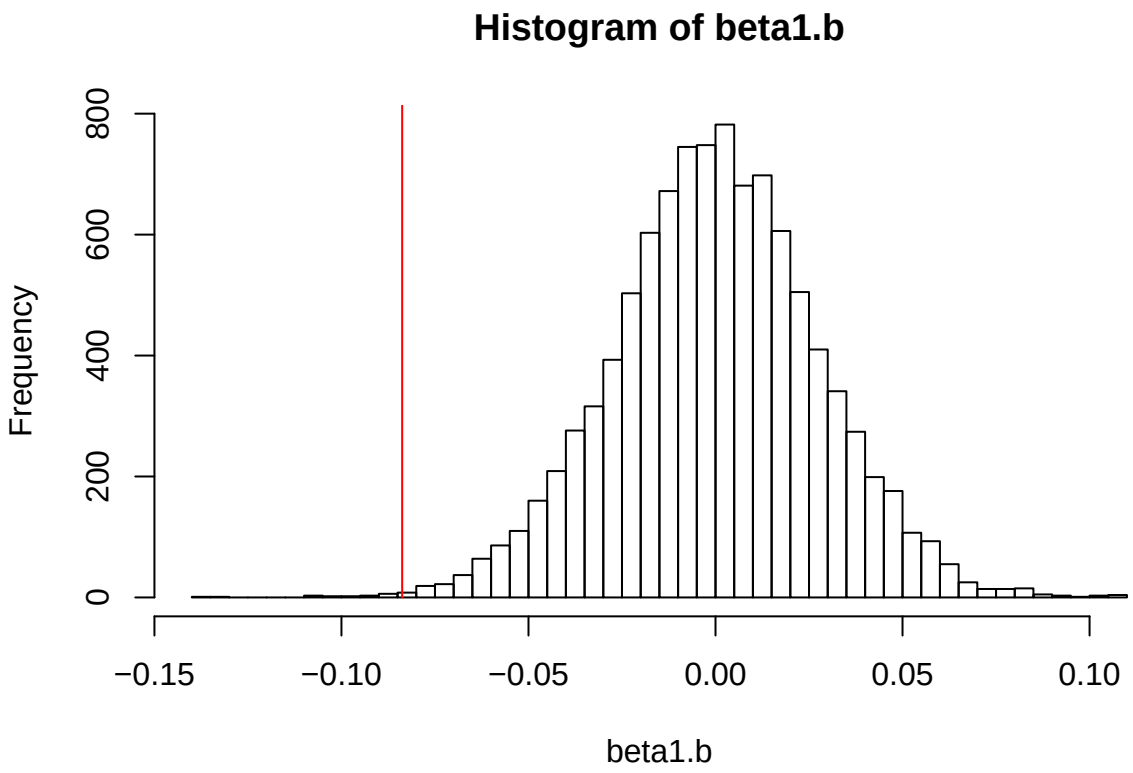


Figure 56: The distribution of the test statistic under the null hypothesis.

```
n<-length(stress[,1])
B<-10000
beta0.b<-beta1.b<-c(1:B)
for (i in 1:B)
{
respondents.b<-rpois(n,hat.mu)
month.b<-month
fit.glm.b<-glm(respondents.b ~ month.b, family=poisson)
beta0.b[i]<-summary(fit.glm.b)$coeff[1,1]
beta1.b[i]<-summary(fit.glm.b)$coeff[2,1]
}
```

The distribution of the test statistic under the null is shown in Figure @ref(fig:figchp1111). The vertical red line is the observed test statistic.

```
par(mfrow=c(1,1))
hist(beta1.b,nclass=50,xlim=c(-0.12,0.12))
lines(c(beta1.obs,beta1.obs),c(0,1000),col=2)
```

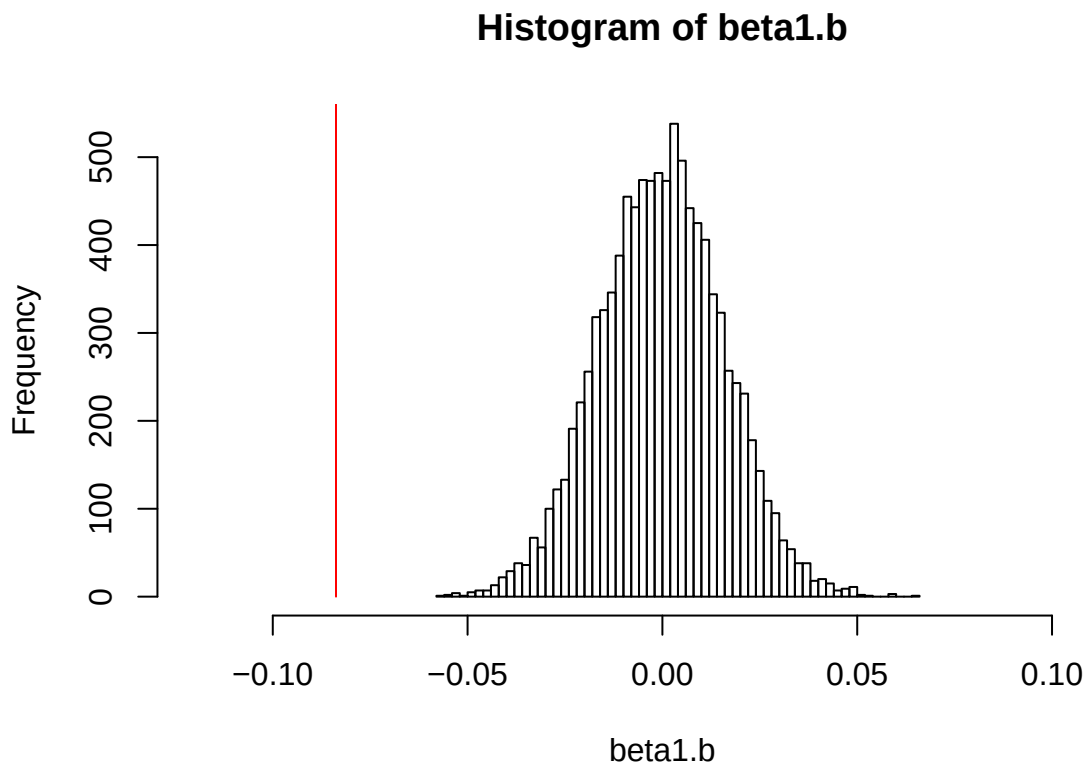


Figure 57: The distribution of the test statistic under the null hypothesis

The monte carlo p value is smaller than 0,05 indicating that the null hypothesis should be rejected.

```
(1+sum(abs(beta1.b)>abs(beta1.obs)))/(B+1)
```

```
## [1] 9.999e-05
```

Part VI
Selected topics

Estimation of bias

Introduction and notation

In this chapter we focus on bias estimation for a plug-in estimates. Let $F(\theta)$ be an unknown probability distribution, θ be the parameter of primary interest, $\theta = t(F)$ and let $\hat{\theta}$ the plug-in estimate. The (asymptotic) bias is given by

$$E(\hat{\theta}) - \theta.$$

For example, if θ is the mean of F and $\hat{theta} = \bar{x}$ is the sample mean we can show that the sample mean is an unbiased estimate for the mean since $E(\hat{\theta} = E(\bar{x}) = \theta$. In the chapter we discuss the application of the bootstrap to calculate the bias of any parameter estimate of interest.

Slides

Slides for the sixth part of the book that covers materials for the next three chapters about bias estimation, cross-validation and jackknife can be found here: Slides6.

Estimating the bias using bootstrap

Let us assume that $x_1 \ldots, x_n$ is a random sample from $F(\theta)$ and that a bootstrap procedure (parametric or non parametric) was applied to the data. Let

$$\hat{\theta}^*_1, \ldots, \hat{\theta}^*_B,$$

be the bootstrap replicates. For any parameter estimate of interest, the bootstrap estimate for $E(\hat{\theta})$ is the mean of the bootstrap replicates given by

$$\bar{\hat{\theta}}^* = \frac{1}{B} \sum_{b=1}^{B} \hat{\theta}^*_b.$$

The bootstrap estimates for the bias is defined as:

$$\bar{\hat{\theta}}^* - \hat{\theta}.$$

Example: the patch data

The patch data consists of 8 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. Data is shown below.

```
library(bootstrap)
patch
```

```
##   subject placebo oldpatch newpatch     z     y
## 1       1    9243    17649    16449  8406 -1200
## 2       2    9671    12013    14614  2342  2601
## 3       3   11792    19979    17274  8187 -2705
## 4       4   13357    21816    23798  8459  1982
## 5       5    9055    13850    12560  4795 -1290
## 6       6    6290     9806    10157  3516   351
```

```
## 7       7   12412    17208    16570  4796  -638
## 8       8   18806    29044    26325 10238 -2719
```

Let z_i bet he difference between the new patch and the old patch, $z_i = old_i - new_i$and let y_i bet he difference between the old patch to the placebo patch, $y_i = old_i - placebo_i$. The parameter of interest the ratio

$$\theta = \frac{E(z)}{E(y)}.$$

The parameter estimate is the sample ratio

$$\frac{\bar{z}}{\bar{y}}.$$

The sample means $\bar{y}$ and $\bar{z}$ are shown below.

```
y<-patch$y
z<-patch$z
c(mean(y),mean(z))
```

```
## [1] -452.250 6342.375
```

The observed ratio:

```
theta.obs<-mean(y)/mean(z)
theta.obs
```

```
## [1] -0.0713061
```

Our main interest is to estimate the bias of the ratio. A non parametric bootstrap procedure for the ratio can be implemented as follows.

```
n<-length(z)
B<-1000
index<-c(1:n)
theta.boot<-c(1:B)
for(i in 1:B)
{
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)
}
```

Distribution of the bootstrap replicates and the observed ratio are shown in Figure @ref(fig:figchp121a).

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

The estimate for the bias, $\bar{\hat{\theta}}^* - \hat{\theta}$, is given by

```
m.boot <- mean(theta.boot)
b.boot <- m.boot - theta.obs
b.boot
```

```
## [1] 0.004715343
```

Figure @ref(fig:figchp122a) visualizes the bias and the bootstrap replicates.

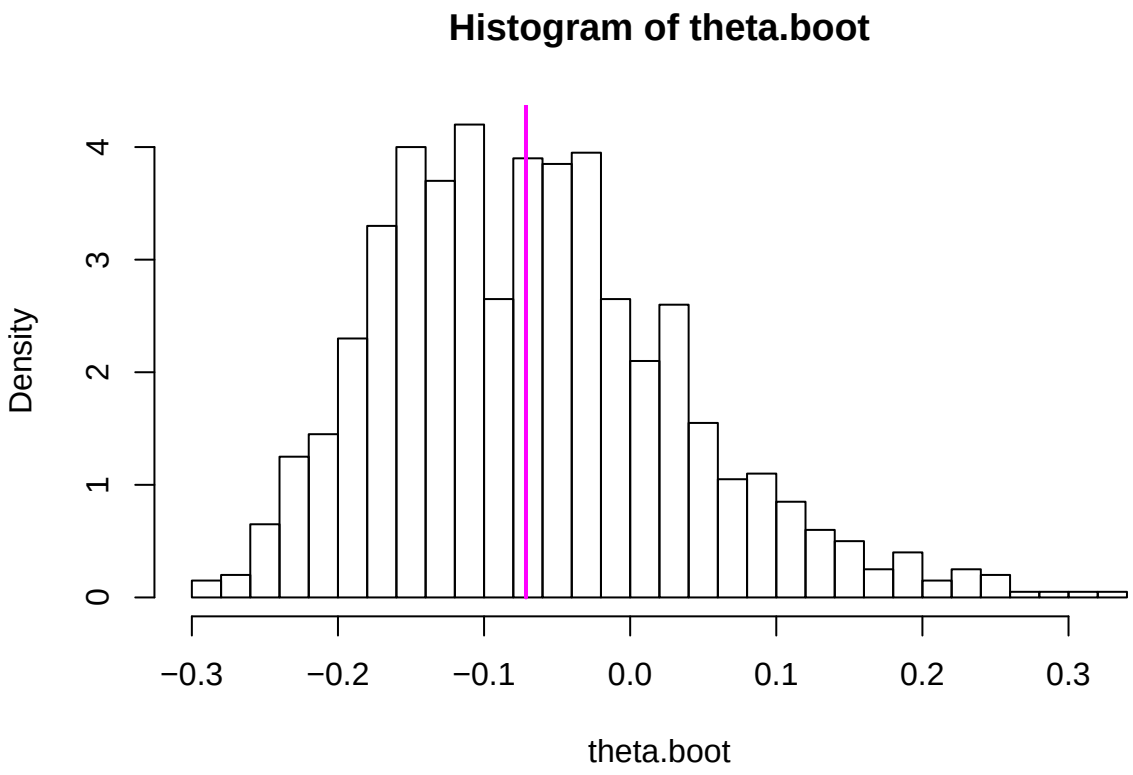


Figure 58: The distribution of the bootstrap replicates.

```
plot(c(1:B),theta.boot)
abline(m.boot,0,col=2)
abline(theta.obs,0,col=4)
```

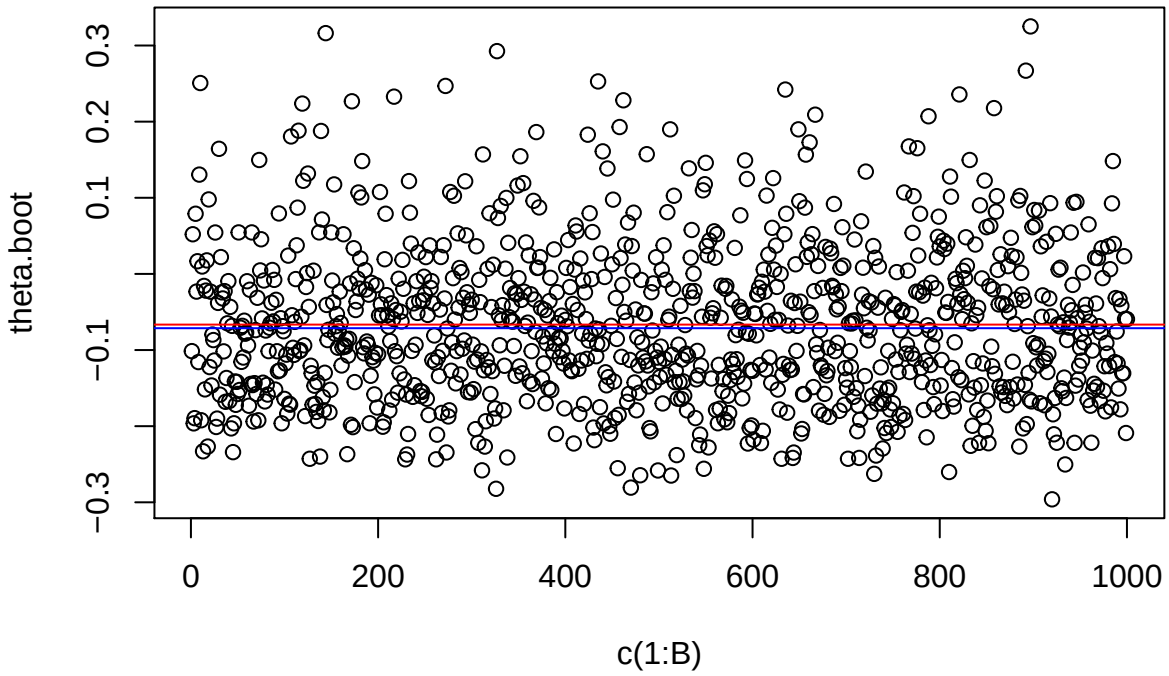


Figure 59: The distribution of the bootstrap replicates and bias estimation.

A zoom in to the bias and the bootstrap replicates is shown in Figure @ref(fig:figchp123a).

```
plot(c(1:B),theta.boot,ylim=c(-0.1,-0.06))
abline(m.boot,0,col=2)
abline(theta.obs,0,col=4)
```

The jackknife

The jackkinfe procedure

The jackknife is a resampling procedure that can be used to calculate the standard error or bias of an estimate. In each resampling step of the jackknife loop, one observation from the original sample is left out and the plag in estimate is calculated. Let $(x_1 \ldots, x_{i-1}, x_i, x_{i+1} x_n)$ be a random sample from the unknown probability distribution $F(\theta)$. Our aim is to estimate the unknown parameter(s) θ and it standard error and possible bias. The jackknife sample consists of leaving out one observation from the observed sample. The ith jackknife sample, consists of $n-1$ observations (without x_i), is given by

$$(x_1 \ldots, x_{i-1}, x_{i+1} x_n).$$

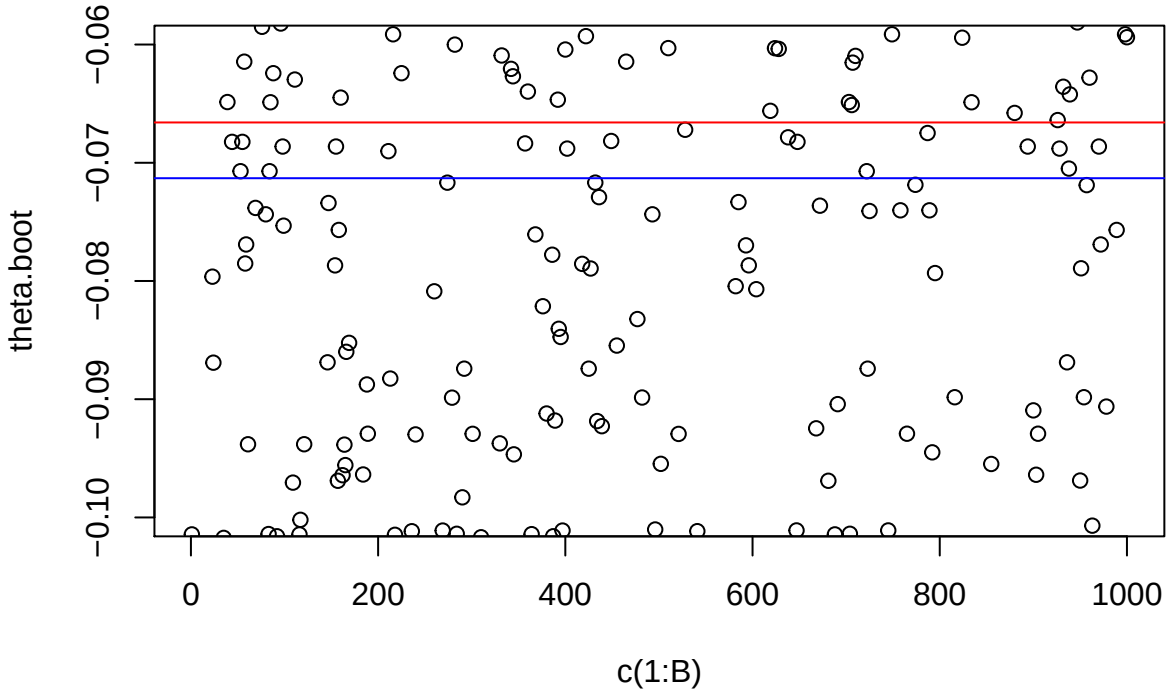


Figure 60: Zoom in to the the distribution of the bootstrap replicates and bias estimation.

Let us assume that θ is the mean of $F(\theta)$. Let $\hat{\theta}^{(-i)}$ be the mean of the ith jackknife sample given by

$$\hat{\theta}^{(-i)} = \frac{1}{n-1} \sum_{j,j \neq i} x_j.$$

The jackknife estimate for the mean is equal to

$$\hat{\theta}^{(.)} = \frac{1}{n} \sum_{i} \hat{\theta}^{(-i)}.$$

The jackknife standard error for $\hat{\theta}$ is given by

$$SE\hat{\theta} = \frac{n-1}{n} \sum_{i=1}^{n} \left(\hat{\theta}^{(-i)} - \hat{\theta}^{(.)} \right)^2.$$

The term $(n-1)$ is called the inflation factor. The inflation factor is needed since the jackknife replicates have less variability around the sample mean due to the fact that at each iteration only one observation is taken out from the observed sample. Therefore, if we do not use the inflation factor we will under estimate the variability of the parameter estimate. This point will be illustrated in the flowing sections.

The jackknife estimate for the bias:

$$(n-1) \left(\hat{\theta}^{(.)} - \hat{\theta} \right).$$

Example: the airquality data

The airquality data gives information about the daily air quality in New York, May to September 1973. In this section we focus on the mean Ozone in parts per billion from 1300 to 1500 hours at Roosevelt Island (see Figure Figure @ref(fig:figchp131a)). The parameter of primary interest is the mean ozone level.

```
x <- na.omit(airquality$Ozone)
hist(x, col = 0)
```

```
mean(x)
```

```
## [1] 42.12931
```

```
n <- length(x)
n
```

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

```
var(x)/n
```

```
## [1] 9.381039
```

We apply both a non parametric bootstrap and a jackknife procedures to the data. A Non parametric bootstrap procedure for the airquality data can be implemented using the code below. A histogram for the bootstrap replicates is shown in Figure @ref(fig:figchp132a).

```
B<-1000
m.x<-c(1:B)
for(i in 1:B)
{
```

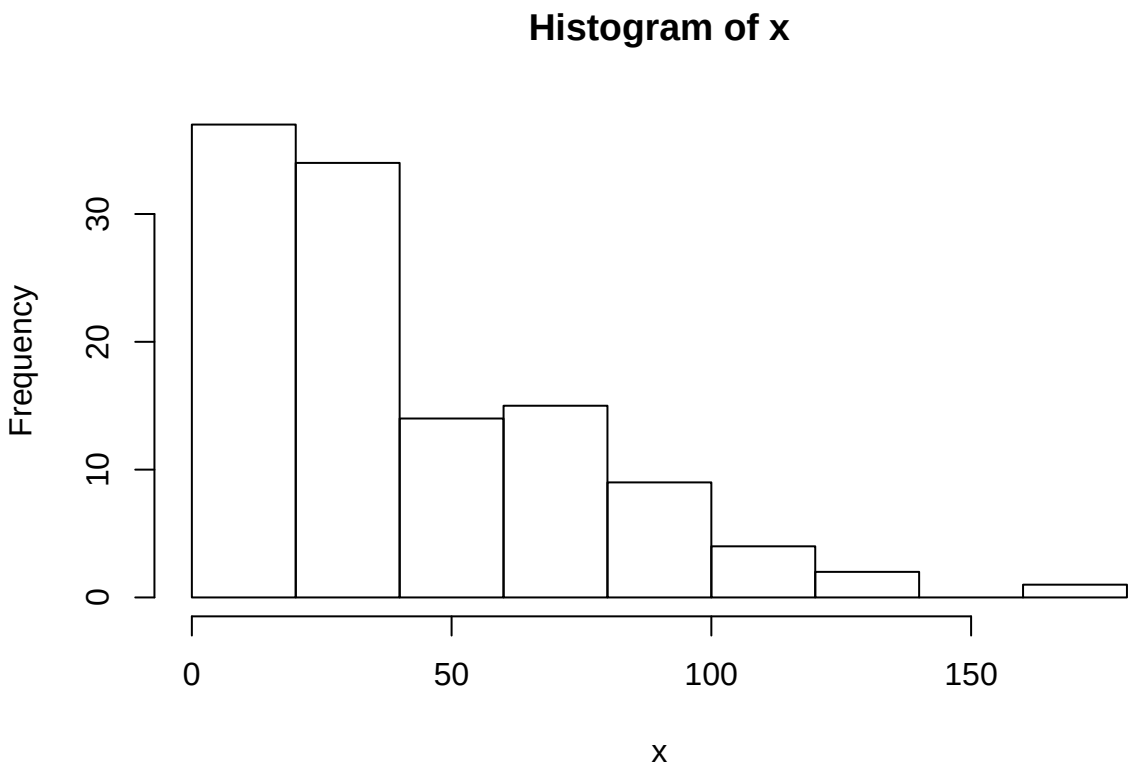


Figure 61: The airquality data: Ozone level.

```
x.boot<-sample(x,n,replace=TRUE)
m.x[i]<-mean(x.boot)
}
hist(m.x,nclass=50)
```

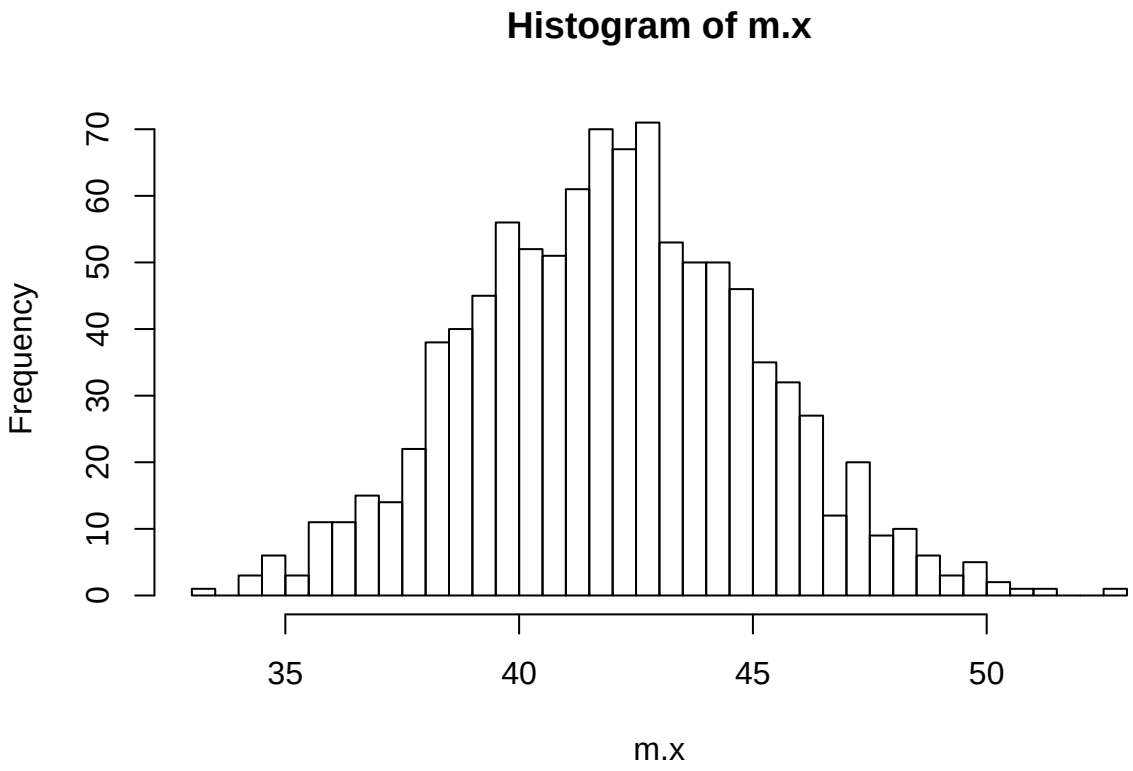


Figure 62: Distribution of the bootstrap replicates for the mean ozone level.

A Jackknife procedure for the airquality data is implmented using a “for loop”" in which at each setp of the loop one observation is left out of the data. In the cose below, this is done in the line x.jack <- x[- c(i)].

```
mean(x)
```

```
## [1] 42.12931
```

```
n <- length(x)
var(x)/n
```

```
## [1] 9.381039
```

```
m.boot <- m.jack <- c(1:n)
 for(i in 1:n) {
    x.jack <- x[ - c(i)]
    m.jack[i] <- mean(x.jack)
    x.boot <- sample(x, size = n, replace = T)
    m.boot[i] <- mean(x.boot)
}
```

The distributions of the bootstrap and jackknife replicates is shown in Figure @ref(fig:figchp132b). Note that for the jackknife we did not include yet the inflation factor.

```
par(mfrow=c(2,1))
hist(m.boot,nclass=10,xlim=c(30,50),main="bootstrap replicates")
hist(m.jack,nclass=10,xlim=c(30,50),main="jackknife replicates")
```

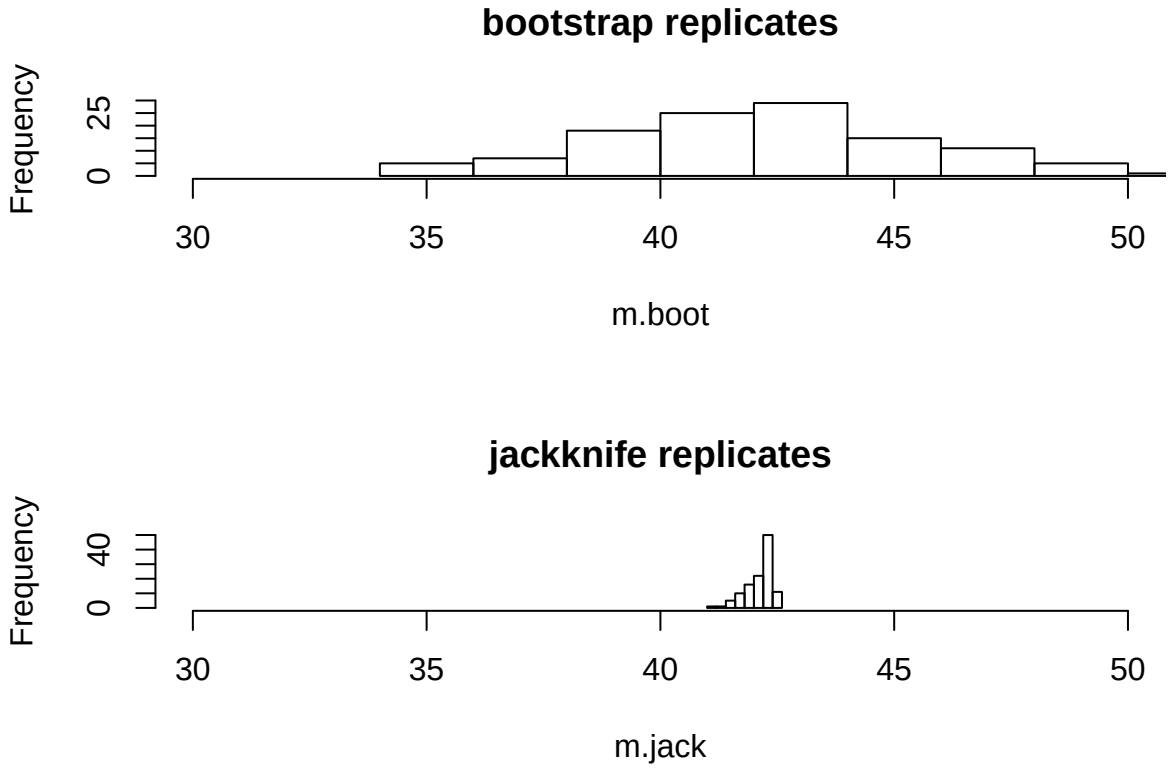


Figure 63: Distribution of the bootstrap replicates and jackknife replicates for the mean ozone level.

Figure @ref(fig:figchp133a) showns the distributions of the (centered) bootstrap and jackknife replicates . Note that for this Figure, the the inflation factor is included for the jackknife replicates in the lower panel but not in the middle panel.

```
m.b<-mean(m.boot)
m.j<-mean(m.jack)
par(mfrow=c(3,1))
hist(m.boot-m.b,nclass=10,main=" ",xlim=c(-10,8))
hist(m.jack-m.j,nclass=10,main=" ",xlim=c(-10,8))
hist(sqrt(n-1)*(m.jack-m.j),nclass=10,main=" ",xlim=c(-10,8))
```

Example: the patch data

The patch data, discussed in the previous chapter, consists of 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. The parameter of primary interest is the mean ozone level. The parameter estimate of primary interest is the sample ratio $\frac{\bar{z}}{\bar{y}}$.

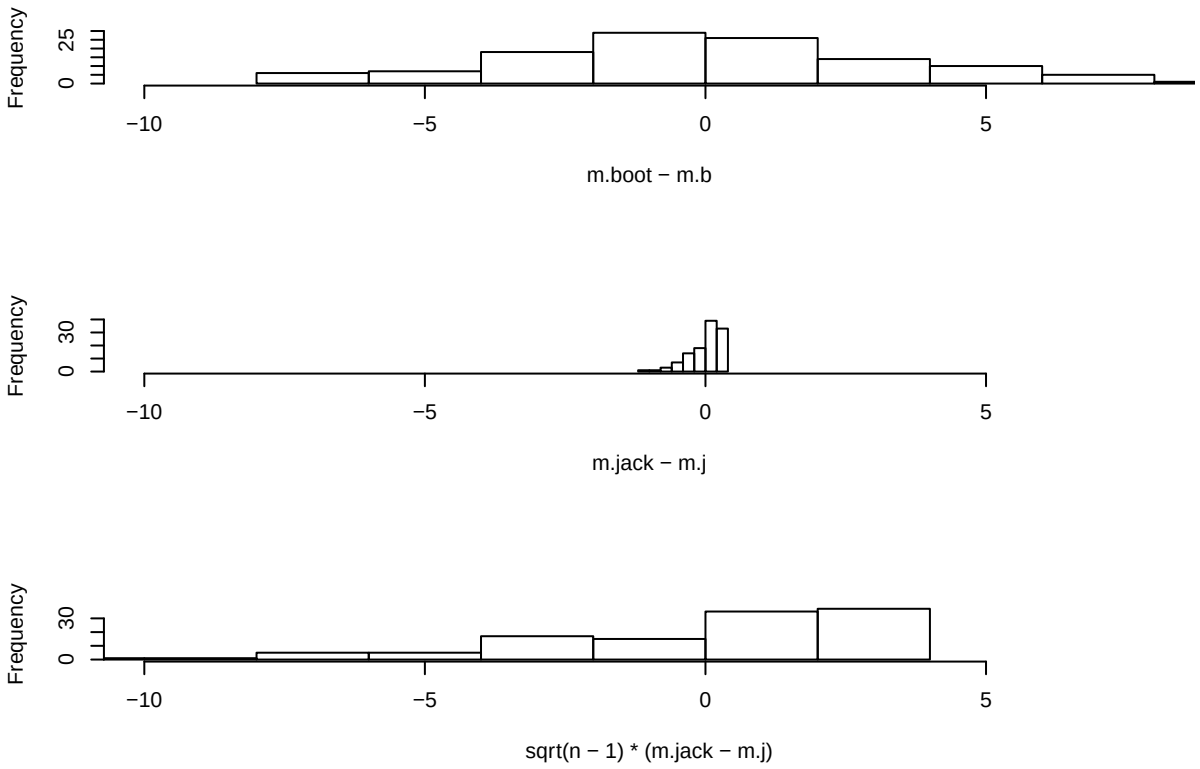


Figure 64: Distribution of the bootstrap replicates and jackknife replicates (with and without inflation factor) for the mean ozone level.

```
z<-patch$z
y<-patch$y
theta.obs <- mean(y)/mean(z)
```

The observed ratio is equal to

```
theta.obs
```

```
## [1] -0.0713061
```

The jackknife procedure for the patch data.

```
n <- length(z)
m.jack <- c(1:n)
for(i in 1:n) {
    cat(i)
    z.jack <- z[ - c(i)]
    y.jack <- y[ - c(i)]
    m.jack[i] <- mean(y.jack)/mean(z.jack)
}
```

```
## 12345678
```

The jackknife estimate for the bias, $(n-1)\left(\hat{\theta}^{(\cdot)}-\hat{\theta}\right)$,

```
(n - 1) * (mean(m.jack) - theta.obs)
```

```
## [1] 0.008002488
```

Note that we need to include the inflation factor to calculate the bias.

Cross validation

Cross validation for prediction

One problem related to prediction is the fact that the data is used twice: (1) to estimate the model and (2) to predict the response. This can lead to a problem of an underestimation of the error that will be made when we predict the response of interest.
A procedure that can be used to overcome this problem is cross validation (CV), which consists of separating the data into two parts. The first part is called the training set (on which the model is fitted) while the second part, called the test set, is left for evaluation. The model is built using the training set alone and model assessment is done using the testing set alone.
The procedure above consists of splitting the data into two part and it is called 2-fold CV. A K -fold cross validation procedure consists of the following steps:

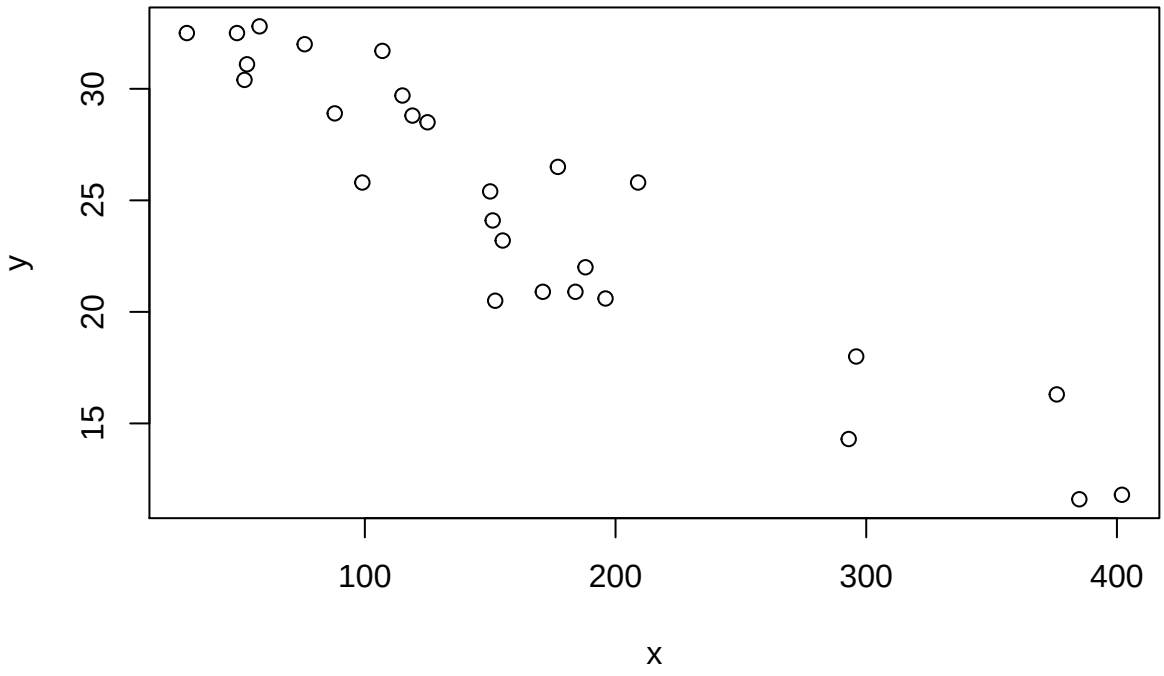
- Split the sample into K equal parts.
- For the kth part, fit the model to the other $K-1$ parts and calculate the prediction error of the fitted model when prediction is done on the kth part.
- Repeat for$ k=1,2,..,K$.

A Leave-One-Out Cross Validation (LOOCV) is a K-fold cross validation for which $K=n$. This implies that n separate times, one observation is left out and the model is trained on $n-1$ observations and evaluated on the observation that left out.

Example: the hormone data

Figure @ref(fig:figchp140a) shows the hormone data gives information about the amount in milligrams of anti-inflammatory hormone remaining in 27 devices, after a certain number of hours (hrs) of wear.

```
y<-hormone$amount
x<-hormone$hrs
plot(x,y)
```



A simple linear regression model of the form $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$ was fitted with the amount anti-inflammatory hormone (y) as a response and hours (x) as predictor.

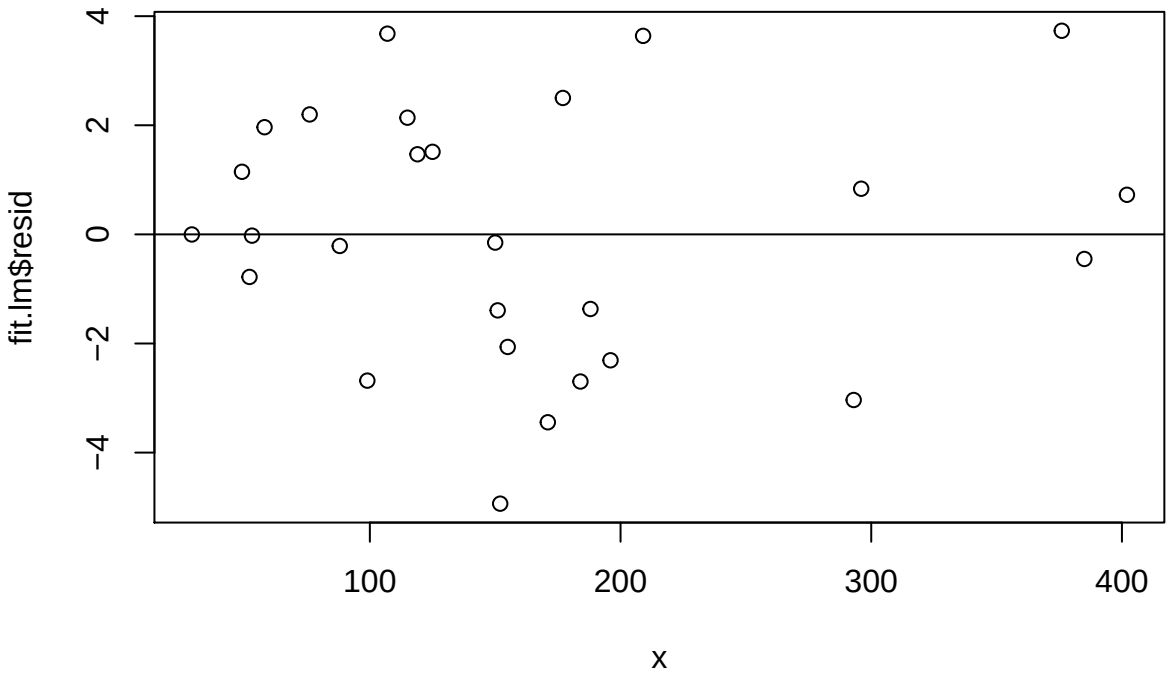
```
fit.lm<-lm(y~x)
summary(fit.lm)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9357 -1.7282 -0.0229  1.7388  3.7323
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.167528   0.867197   39.40  < 2e-16 ***
## x           -0.057446   0.004464  -12.87 1.58e-12 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.378 on 25 degrees of freedom
## Multiple R-squared:  0.8688, Adjusted R-squared:  0.8636
## F-statistic: 165.6 on 1 and 25 DF,  p-value: 1.584e-12
```

Residuals from the fitted model are shown in Figure @ref(fig:figchp141a),

```
plot(x,fit.lm$resid)
abline(0,0)
```



Leave one out CV for the hormone data

We implement a leave-one-out-cross-validation procedure for the hormone dataset. At each step, one observation is left out and the regression model is fitted to the $n-1$ observations. The cross validated residual is calculated for the observation that left out.

```
n <- length(x)
beta.cv <- fit.cv <- c(1:n)
for(i in 1:n) {
    x.cv <- x[ - c(i)]
    y.cv <- y[ - c(i)]
    fit.lm.cv <- lm(y.cv ~ x.cv)
    fit.cv[i] <- fit.lm.cv$coeff[1] + fit.lm.cv$coeff[2]*x[i]
    beta.cv[i] <- fit.lm.cv$coeff[2]
}
```

Note that, in the code below, the cross validated residual, $y_i - \hat{y}^{(-i)}$ is calculated in the line y - fit.cv.

```
res.cv <- y - fit.cv
```

The CV residual score $\frac{\sum_i \left(y_i - \hat{y}^{(-i)}\right)^2}{n}$ is higher than the OLS residual score $\frac{\sum_i (y_i - \hat{y}_i)^2}{n}$,

```
cv.score <- sqrt(sum((res.cv^2))/n)
ols.score <- sqrt(sum((y - fit.lm$fit)^2)/27)
c(cv.score,ols.score)
```

```
## [1] 2.455137 2.288386
```

OLS residuals and CV residuals are shown in Figure @ref(fig:figchp142a). Note that the OLS residuals are greater, as expected, from the CV residuals.

```
plot(x,fit.lm$resid,ylim=c(-5,5))
points(x,res.cv,pch="+",col=2)
abline(0,0)
```

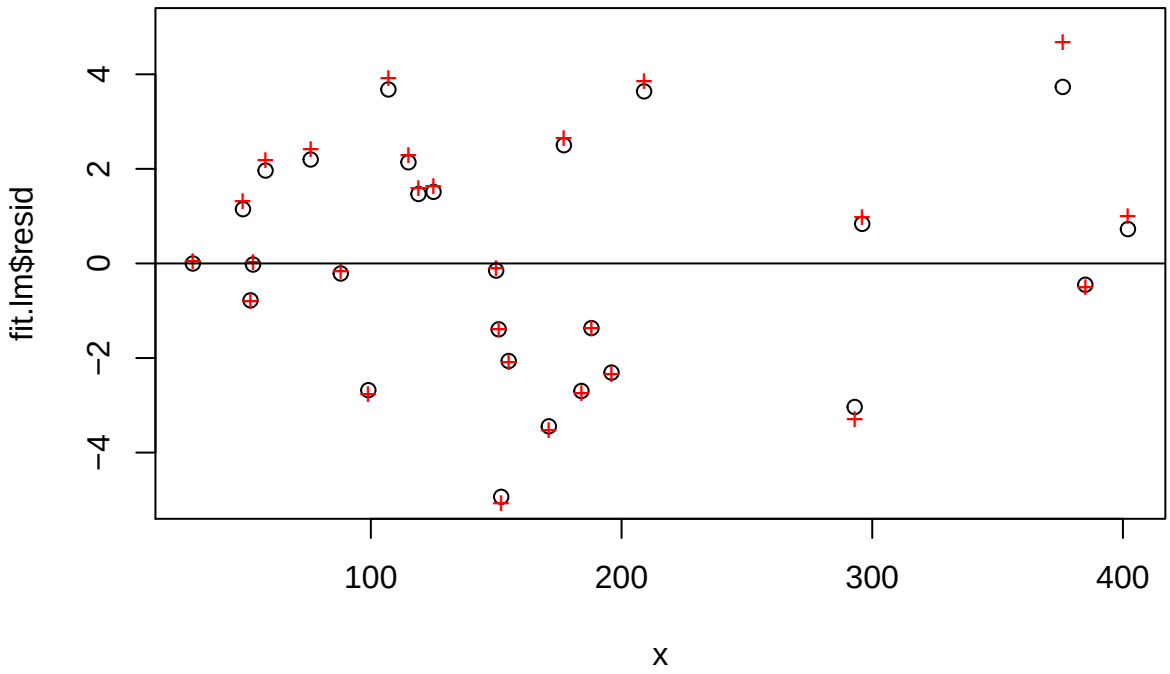


Figure 65: Residuals (OLS) and CV residuals.

Figure @ref(fig:figchp143a) shows the parameter estimate for the slope obtained within the LOOCV procedure.

```
plot(x,beta.cv)
abline(fit.lm$coeff[2],0)
```

Let us focus on observation 10, (x_{10}, y_{10}):

```
c(x[10],y[10])
```

```
## [1] 376.0  16.3
```

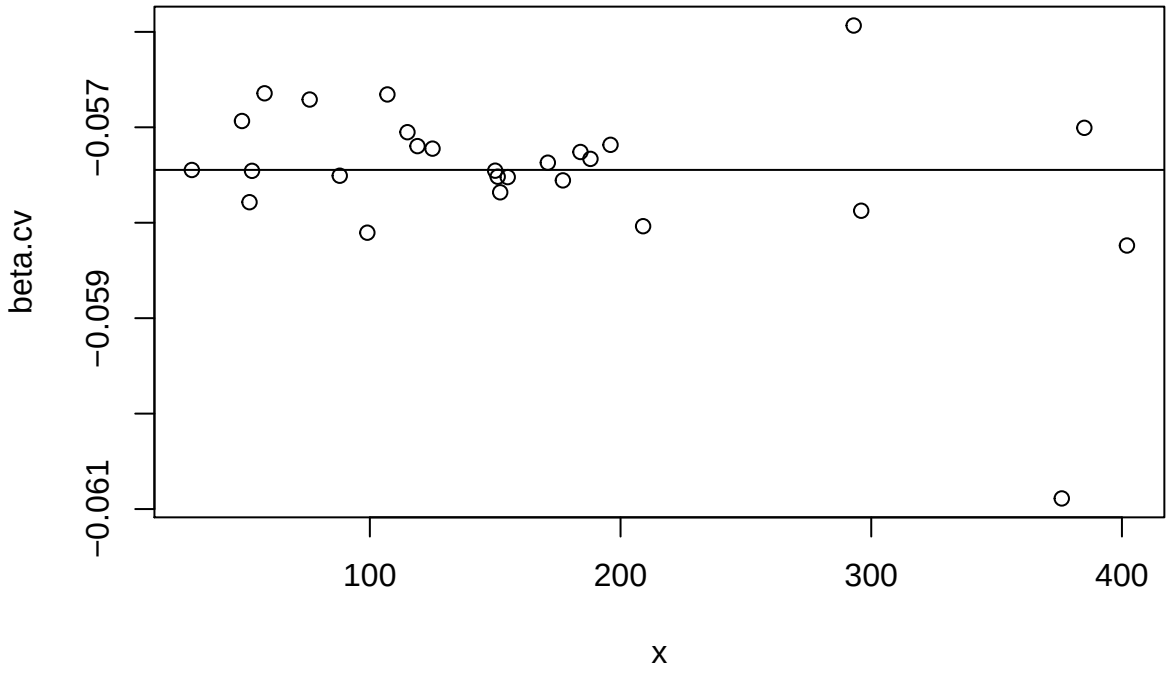


Figure 66: Slope values for the LOOCV

An illustration of leave out observation 10 from the data and fitting the linear model, with and without observation 10 is shown in Figure @ref(fig:figchp144a).

```
plot(x,y)
lines(x,fit.lm$fit)
x.10<-x[-c(10)]
y.10<-y[-c(10)]
points(x.10,y.10,col=2)
fit.lm.10<-lm(y.10~x.10)
lines(x.10,fit.lm.10$fit,col=2)
```

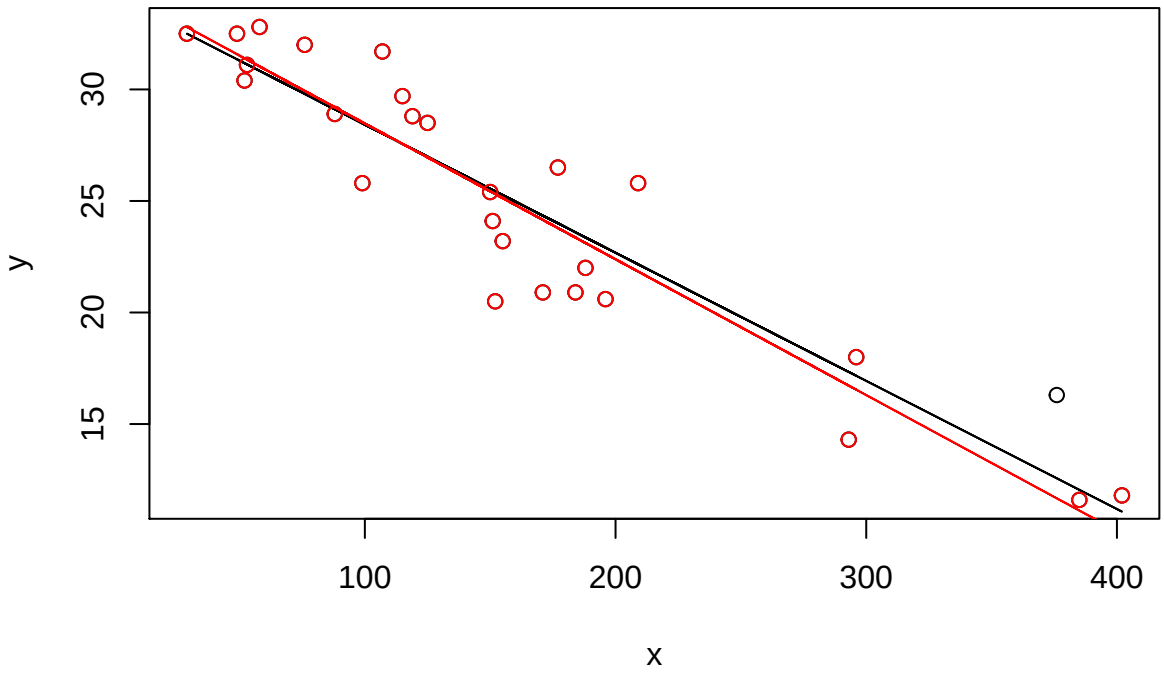


Figure 67: The hormone data: fitted models with and without observation 10.

Note that for the original data, $\hat{\beta}_1 = -0.0574$ while for the data without observation 10, $\hat{\beta}_1 = -0.0608$. This difference can be clearly seen in Figure @ref(fig:figchp143a).

```
summary(fit.lm.10)
```

```
##
## Call:
## lm(formula = y.10 ~ x.10)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8090 -1.7622 -0.1338  1.6717  3.9617
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.564057   0.857131   40.33  < 2e-16 ***
## x.10        -0.060889   0.004666  -13.05 2.16e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.274 on 24 degrees of freedom
## Multiple R-squared:  0.8765, Adjusted R-squared:  0.8713
## F-statistic: 170.3 on 1 and 24 DF,  p-value: 2.158e-12
```

Part VII

Scatterplot smoothing with LOESS

Non parametric regression (I)

Slides

Slides for the last part of the book about non parametric regression be found here: Slides7.

Example: the fuel data

The fuel dataset gives information on cars taken from the April, 1990 issue of Consumer Reports and consists of two variables:

- The mileage: a numeric vector of gas mileage in miles/gallon as tested by CU.
- Weight: a numeric vector of the car's weight.

```
fuel.frame<-read.table('C:/projects/cim/UpdatesSlides_2017/Data/fuel.txt')
names(fuel.frame)<-c("ID","Weight","Mileage")
par(mfrow=c(1,1))
x<-fuel.frame$Weight
y<-fuel.frame$Mileage
y<-y[order(x)]
x<-sort(x)
```

Parametric and non parametric regression models for the fuel data

Our aim is to model the relationship between the car's weight and mileage, to predict the mileage of a car with weight of 3200 Kg and to estimate the standard error for a prediction for this specific car's weight.

Modeling the relationship between the car's weight and mileage

We consider two possible models for the car mileage. A linear regression model given by

$$y_i = \beta_1 + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i,$$

And a non parametric model of the form

$$y_i = r(x_i) + \varepsilon_i.$$

Here, $r(x_i)$ is assume to be a smooth function of the car's weight.The second model is estimated by a loess model with smoothing parameters equal to 0.5 and 0.75. Data and estimated models are shown below.

```
plot(x,y)
fit.lm<-lm(y~x+x^2)
par(mfrow=c(1,1))
plot(x,y)
lines(x,fit.lm$fit,lwd=2)
fit.lo1<-loess(y~x)
fit.lo2<-loess(y~x,span=0.5)
lines(x,fit.lo1$fit,lwd=2,col=4)
lines(x,fit.lo2$fit,lwd=2,col=3)
legend(3000,35,c("regression","loess:lambda=0.75","loess: lambda=0.5"),col=c(1,4,3),lty=c(1,1,1))
```

We estimate the car mileage for a weight of 3200, $\hat{E}(y_i|x_i = 3200$.

```
x<-3200
newdat<-data.frame(x)
predict(fit.lm,newdat)
```

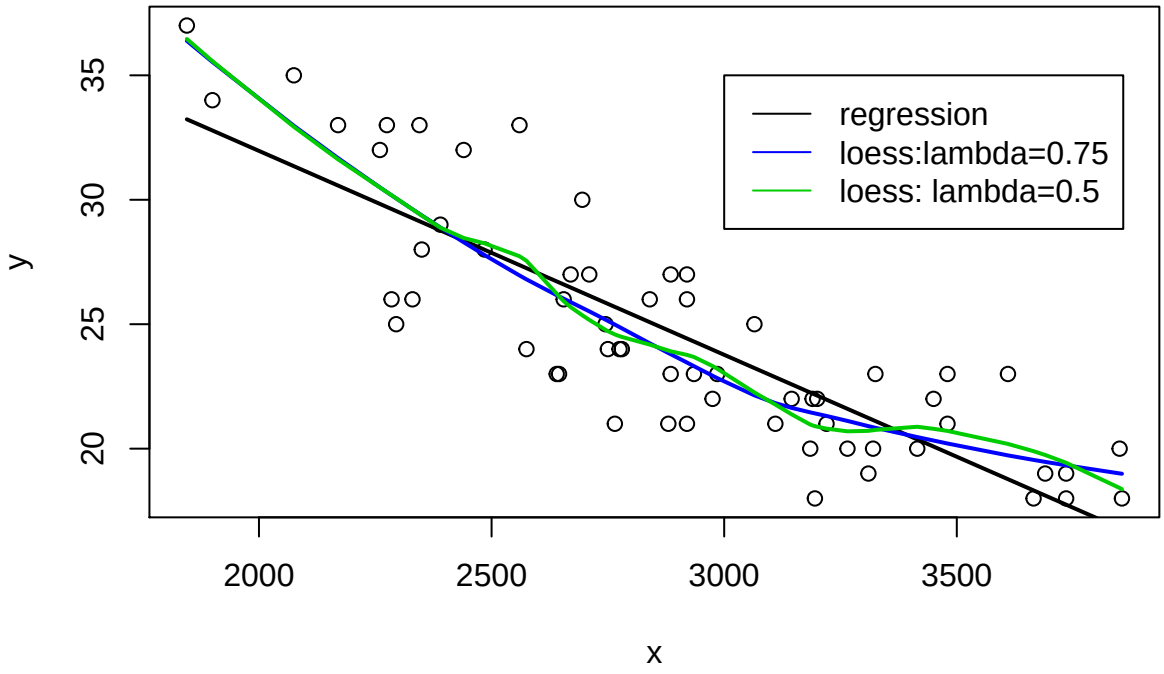


Figure 68: The fuel data: parametric and non parametric regression models

```
##        1
## 22.13231
```

```
predict(fit.lo2,newdat)
```

```
##        1
## 20.88211
```

```
predict(fit.lo1,newdat)
```

```
##        1
## 21.39581
```

Non parametric bootstrap for the fuel data

We use non parametric bootstrap to estimate the standard error for $\hat{E}(y_i|x_i = 3200)$ obtained from the linear regression and the loess modes.

```
x<-fuel.frame$Weight
y<-fuel.frame$Mileage
y<-y[order(x)]
x<-sort(x)
n<-length(x)
index<-c(1:n)
B<-1000
x.boot<-y.boot<-fit.b.lm<-fit.b.lo<-matrix(0,n,B)
x.b<-3000
newdat<-data.frame(x.b)
pred.lm<-pred.lo<-c(1:B)
for(i in 1:B){
boot.i<-sample(index,n,replace=T)
x.b<-x[boot.i]
y.b<-y[boot.i]
y.b<-y.b[order(x.b)]
x.b<-sort(x.b)
fit.lm.i<-lm(y.b~x.b+x.b^2)
fit.lo.i<-loess(y.b~x.b,span=0.5)
x.boot[,i]<-x.b
y.boot[,i]<-y.b
fit.b.lm[,i]<-fit.lm.i$fit
fit.b.lo[,i]<-fit.lo.i$fit
pred.lo[i]<-predict(fit.lo.i,newdat)
pred.lm[i]<-predict(fit.lm.i,newdat)
}
```

Data and bootstrap predicted models for the linear regression model are shown below.

```
plot(x,y,ylim=c(15,40))
for(i in 1:B)
{
lines(x.boot[,i],fit.b.lm[,i],col=4)
}
```

Data and bootstrap predicted models for the non parametric regression model are shown below.

```
plot(x,y,ylim=c(15,40))
for(i in 1:B)
```

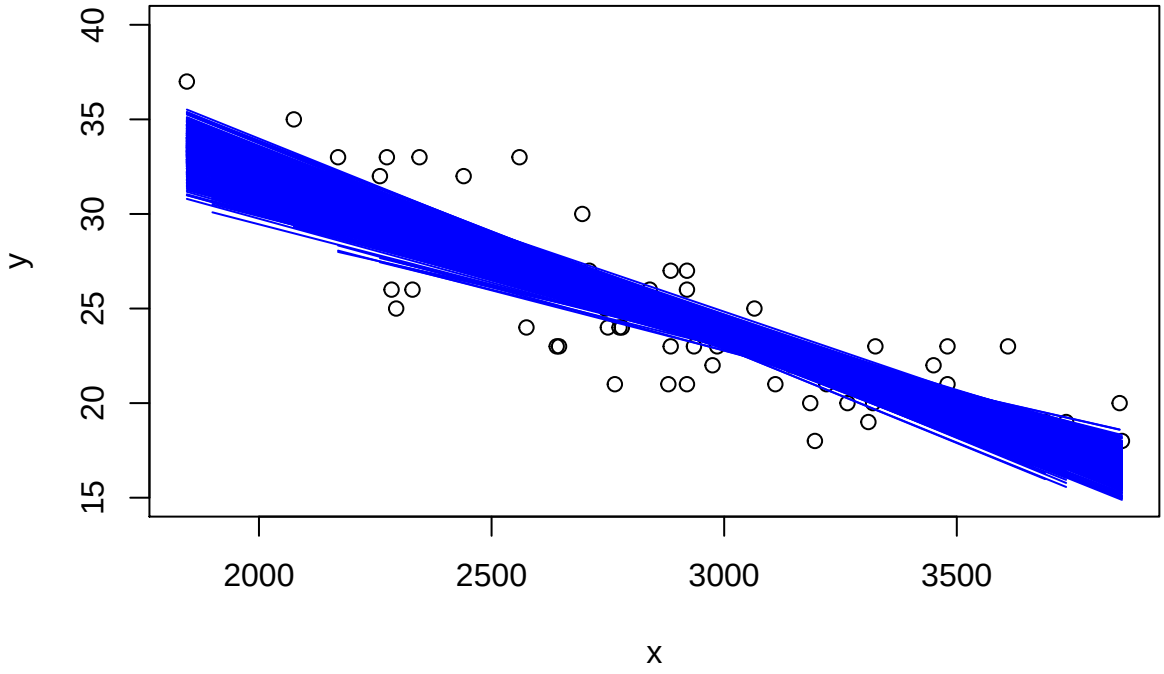


Figure 69: The fuel data and bootstrap replicates for the fitted models (parametric regression).

```
{
lines(x.boot[,i],fit.b.lo[,i],col=4)
}
```

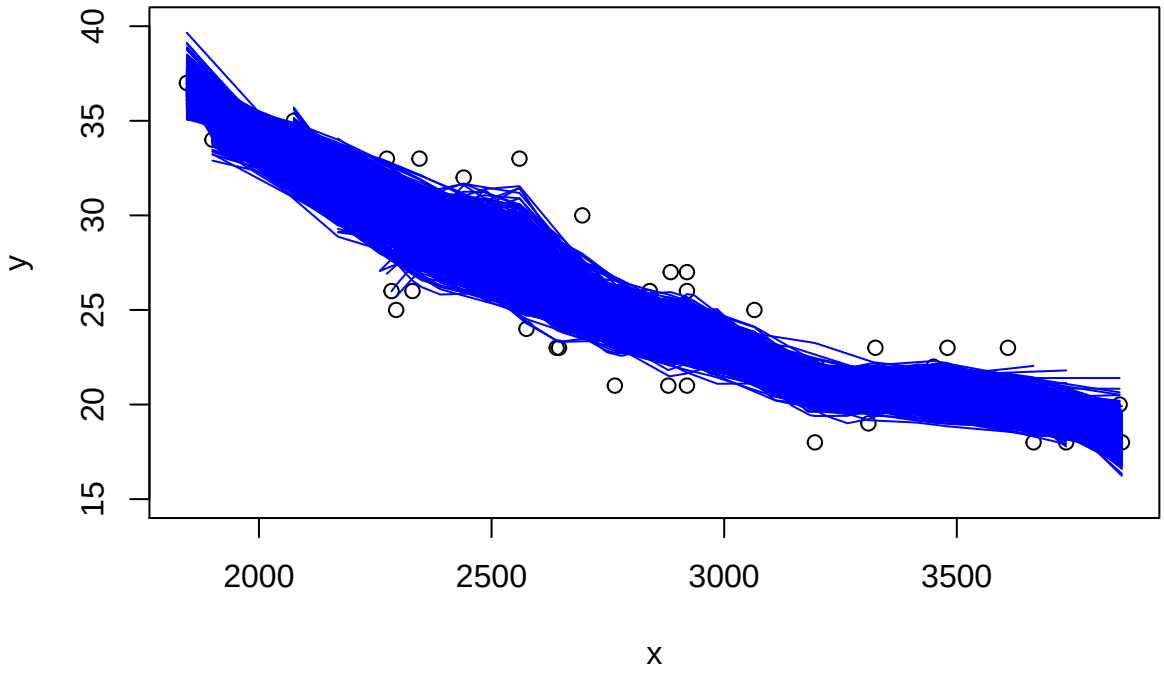


Figure 70: The fuel data and bootstrap replicates for the loess models (lambda=0.5)

Prediction and C.I for the predicted values

```
var(pred.lm)
```

```
## [1] 0.1034462
```

```
var(pred.lo)
```

```
## [1] 0.3933433
```

The distribution of the predictions $\hat{E}(y_i|x_i = 3000)$ and C.Is are shown in the Figures below.

```
cir<-quantile(pred.lm,probs=c(0.025,0.975))
cilo<-quantile(pred.lo,probs=c(0.025,0.975))
par(mfrow=c(1,2))
hist(pred.lm,col=0,nclass=25,probability=T,main=" ")
lines(c(cir[1],cir[1]),c(0,1),lwd=2,col=3)
lines(c(cir[2],cir[2]),c(0,1),lwd=2,col=3)
title("linear regression: r(3000)")
```

```
hist(pred.lo,col=0,nclass=25,probability=T,main=" ")
lines(c(cilo[1],cilo[1]),c(0,1),lwd=2,col=3)
lines(c(cilo[2],cilo[2]),c(0,1),lwd=2,col=3)
title("loess: r(3000)")
```

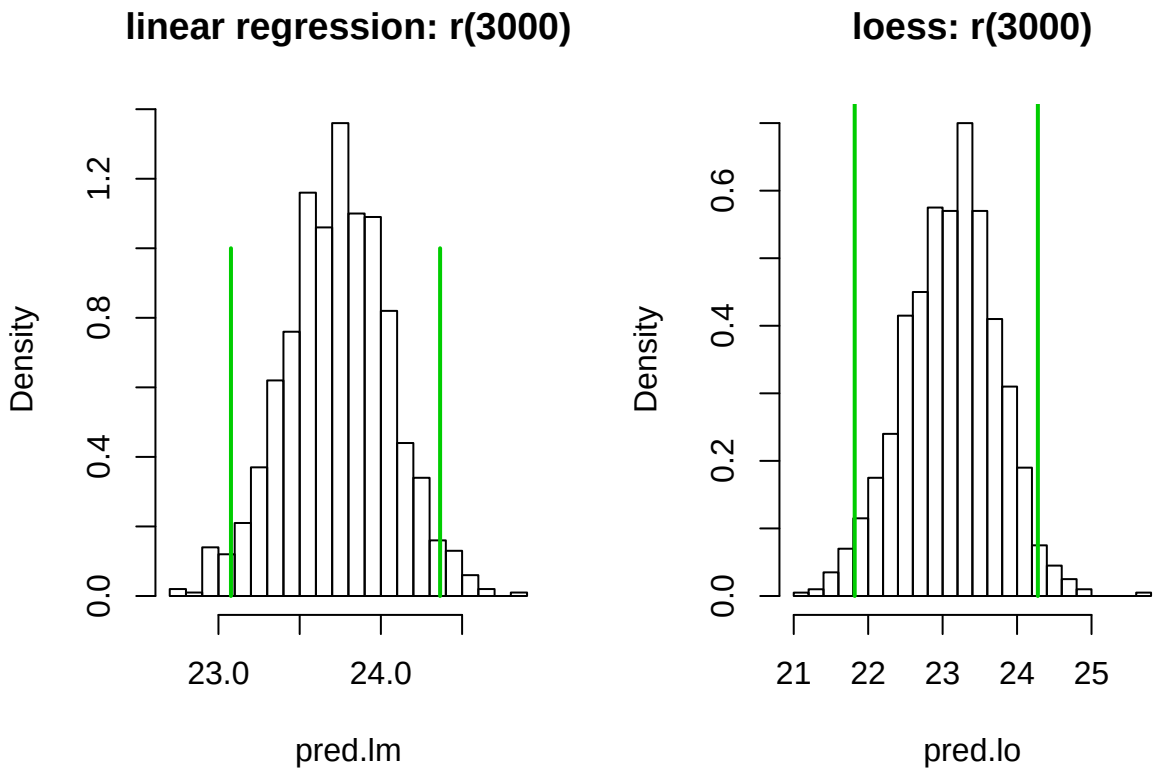


Figure 71: Confidence intervals for the predicted values at x=3000

95% C.I for $E(y_i|x_i = 3200)$ are given by

```
cir
```

```
##     2.5%    97.5%
## 23.07691 24.36456
```

```
cilo
```

```
##     2.5%    97.5%
## 21.81882 24.27948
```

Non parametric regression (II)

The airquality data

The air quality data gives information about the daily air quality measurements in New York, May to September 1973. For the analysis presented in this chapter we focus on the mean ozone in parts per billion

from 1300 to 1500 hours at Roosevelt Island (the response) and the average wind speed in miles per hour at 0700 and 1000 hours at LaGuardia Airport (predictor). Data ($n = 111$) are shown below.

```
xxx<-na.omit(airquality)
y<-xxx$Ozone
x<-xxx$Wind
plot(x,y)
```

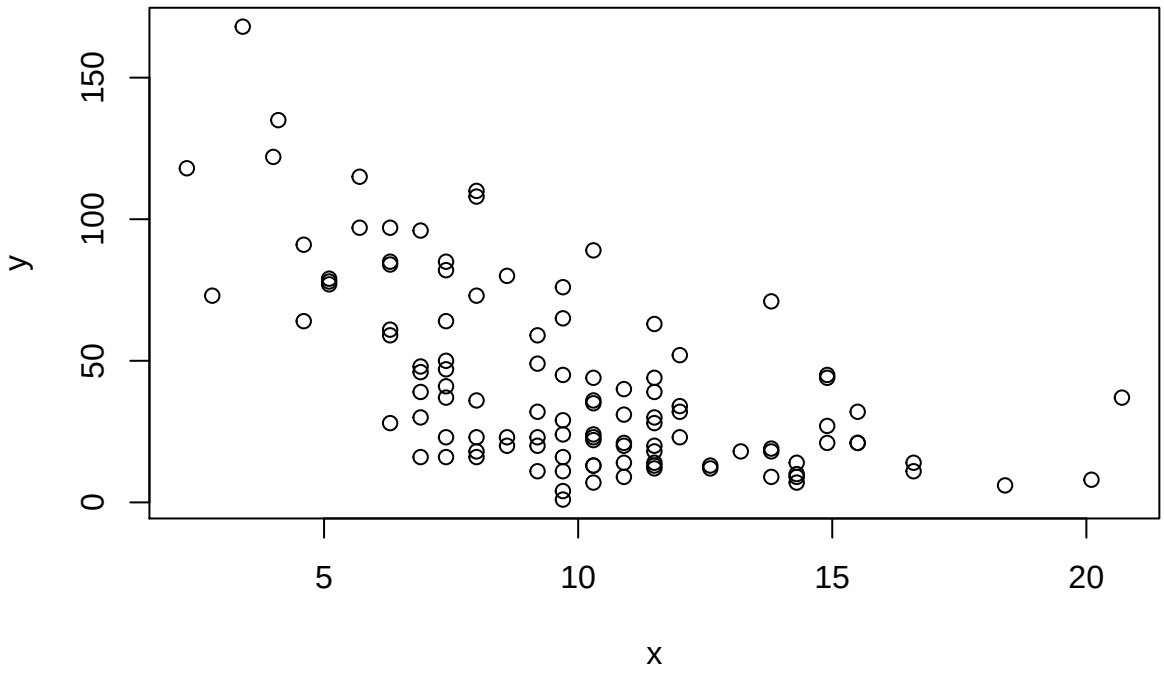


Figure 72: The airquality data: Ozone vs. Wind speed

We specify two model that describe the relationship between the wind speed and the ozone level,

$$H_0 : E(y_i|x_i) = constat,$$
$$H_1 : E(y_i|x_i) = r(x_i).$$

Here, $r(x)$ is a smooth function of x. The model spesficyed under the null hypothesis assume that x does not have an effect on y.

Non parametric model under H_0

We wish to construct a confidence band under the null Model. This can be done by resampling the ozone levels y and fix the wind speed x. We use a loess model to estimate $E(y^*|x)$.

```
nx<-length(x)
B<-1000
nx
```

```
## [1] 111
```

```
boot.x<-boot.fit<-matrix(0,length(x),B)
for(i in 1:B)
{
#x.boot<-sample(x, size=nx, replace=T)
y.boot<-sample(y, size=nx, replace=T)
boot.lo<-loess(y.boot ~ x, degree = 1, span = 0.75)
boot.fit[,i]<-boot.lo$fit
#boot.x[,i]<-x.boot
}
```

Next we smooth the data and plot the data, the estimated model and the models that were estimated under the null hypothesis.

```
par(mfrow=c(1,1))
plot(x,y)
fit.lo<-loess(y~x)
lines(sort(x),fit.lo$fit[order(x)],lwd=2)
for(i in 1:B)
{
#lines(sort(boot.x[,i]),boot.fit[,i][order(boot.x[,i])],col=5)
lines(sort(x),boot.fit[,i][order(x)],col=5)
}
lines(sort(x),fit.lo$fit[order(x)],lwd=2)
```

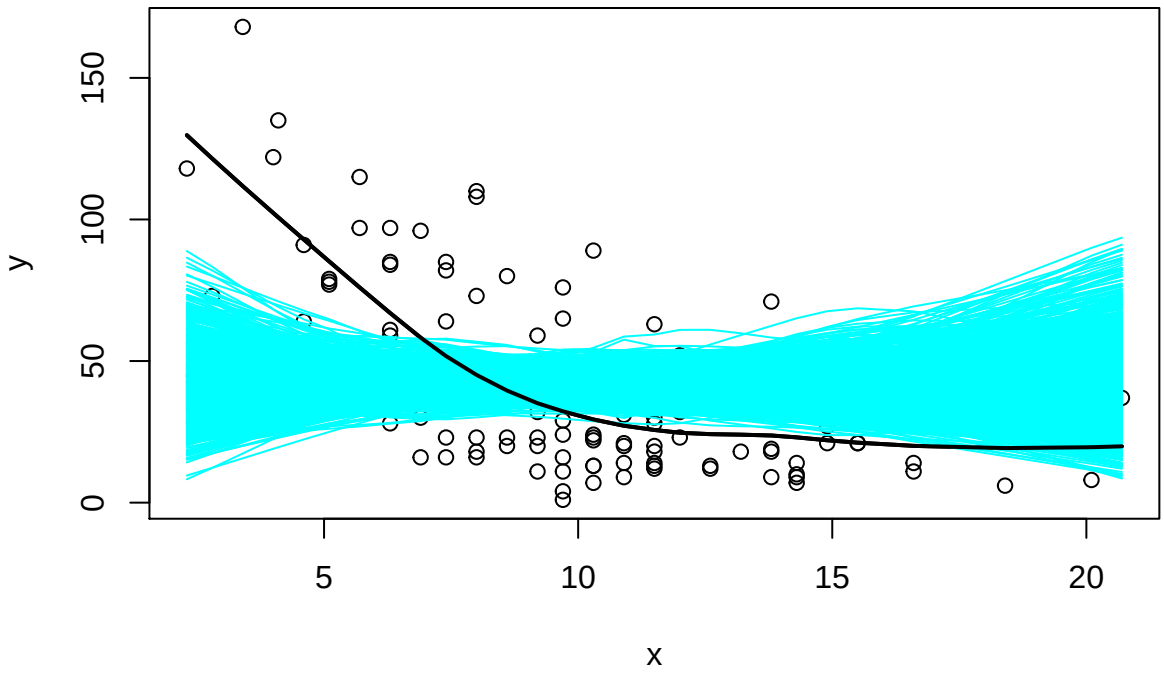


Figure 73: The airquality data, fitted loess model and loess models fitted under the no effect hypothesis