Homework 1

Matteo Salvalaggio, Sauraj Verma

15/03/2021

Exercise 1

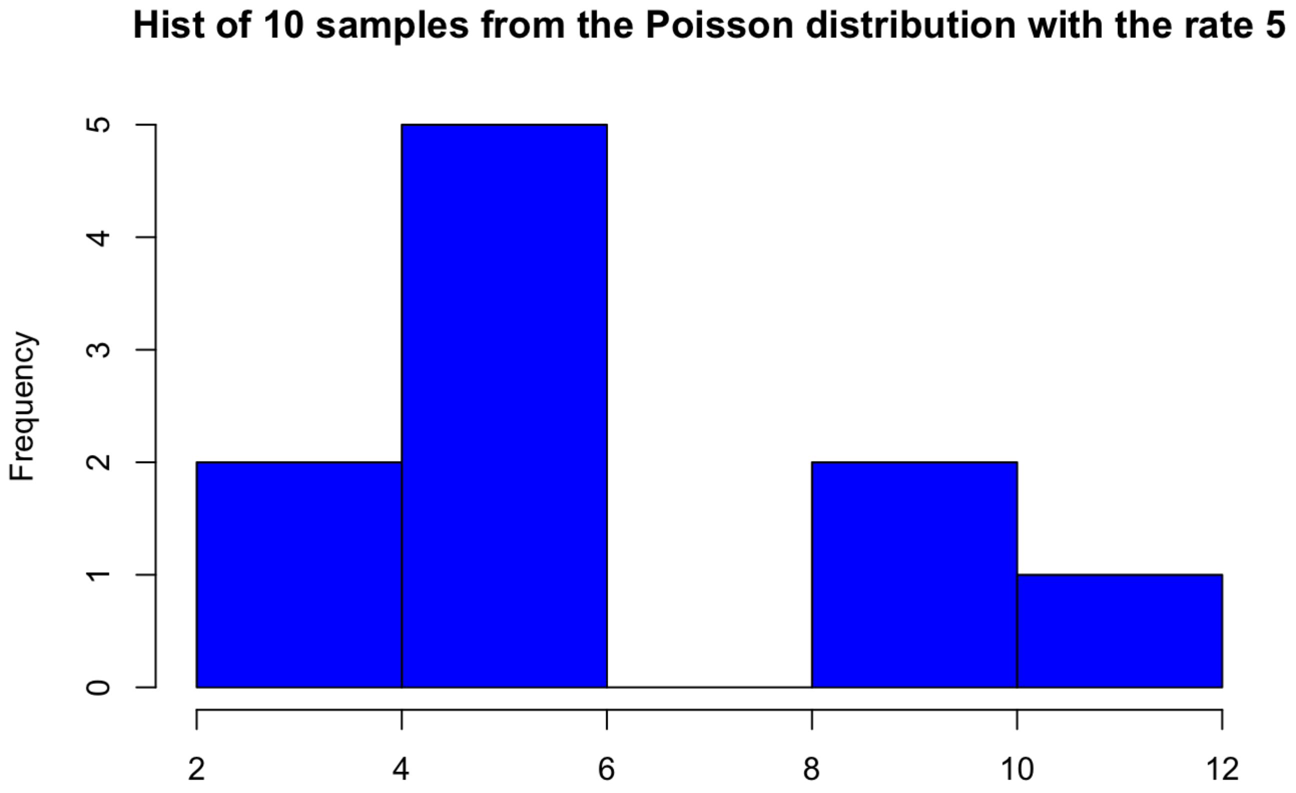
a.

```
library(fitdistrplus)
```

```
## Loading required package: MASS
```

```
## Loading required package: survival
```

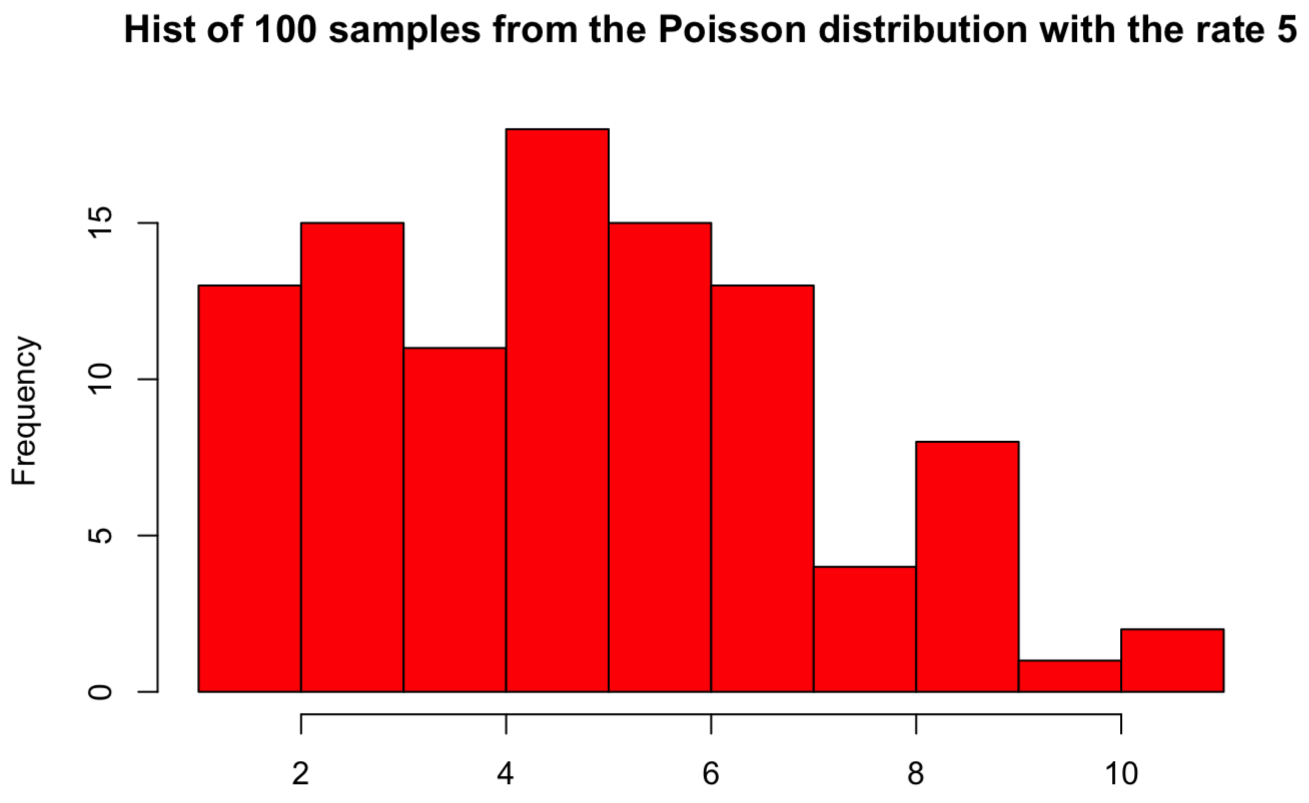
```
n10 <- rpois(10,5) # 10 samples from the Poisson distribution with the rate 5
n100 <- rpois(100,5) # 100 samples from the Poisson distribution with the rate 5
n1000 <- rpois(1000,5) # 1000 samples from the Poisson distribution with the rate 5
n10hist <- hist(n10, col="blue", main = "Hist of 10 samples from the Poisson distributio
n with the rate 5", xlab="")
```



```
n10hist
```

```
## $breaks
## [1]  2  4  6  8 10 12
## 
## $counts
## [1] 2 5 0 2 1
## 
## $density
## [1] 0.10 0.25 0.00 0.10 0.05
## 
## $mids
## [1]  3  5  7  9 11
## 
## $xname
## [1] "n10"
## 
## $equidist
## [1] TRUE
## 
## attr(,"class")
## [1] "histogram"
```

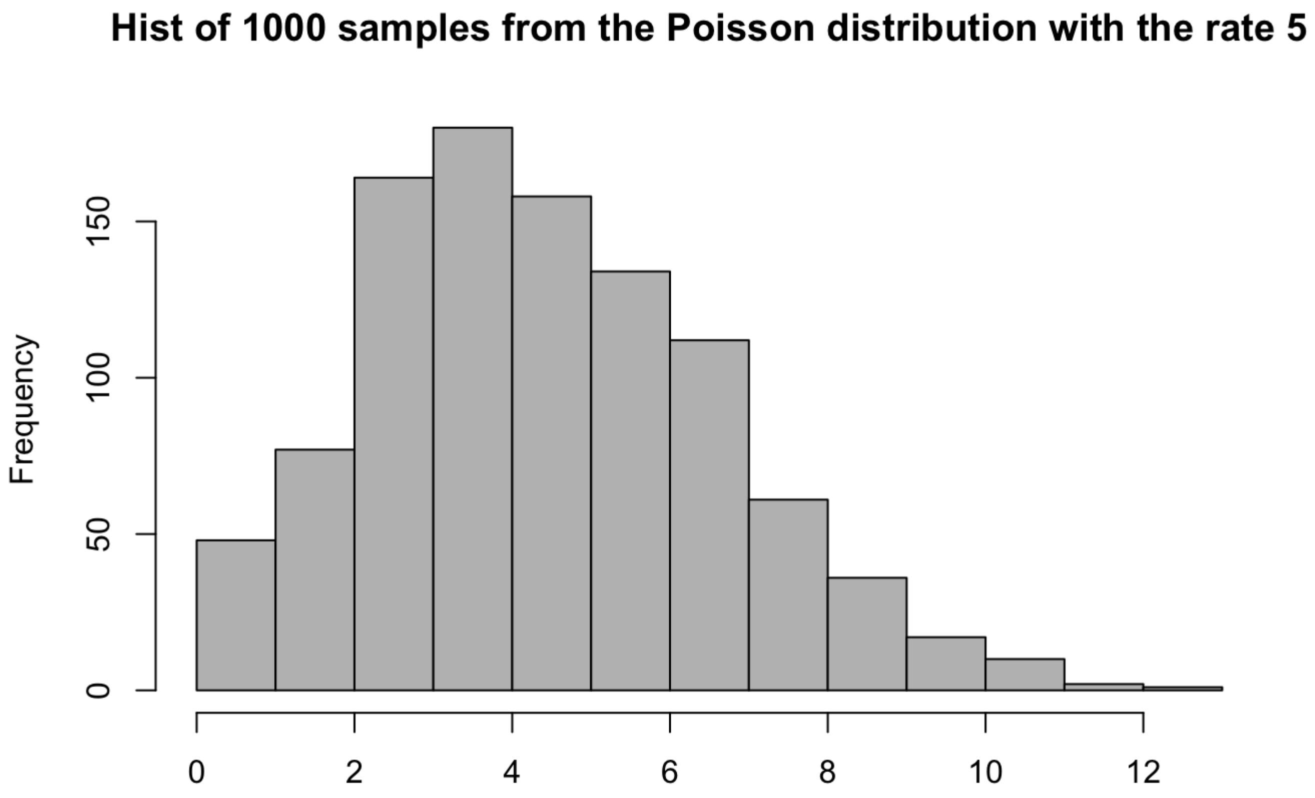
```
n100hist <- hist(n100, col="red", main = "Hist of 100 samples from the Poisson distribut
ion with the rate 5", xlab="")
```



```
n100hist
```

```
## $breaks
##  [1]  1  2  3  4  5  6  7  8  9 10 11
## 
## $counts
##  [1] 13 15 11 18 15 13  4  8  1  2
## 
## $density
##  [1] 0.13 0.15 0.11 0.18 0.15 0.13 0.04 0.08 0.01 0.02
## 
## $mids
##  [1]  1.5  2.5  3.5  4.5  5.5  6.5  7.5  8.5  9.5 10.5
## 
## $xname
## [1] "n100"
## 
## $equidist
## [1] TRUE
## 
## attr(,"class")
## [1] "histogram"
```

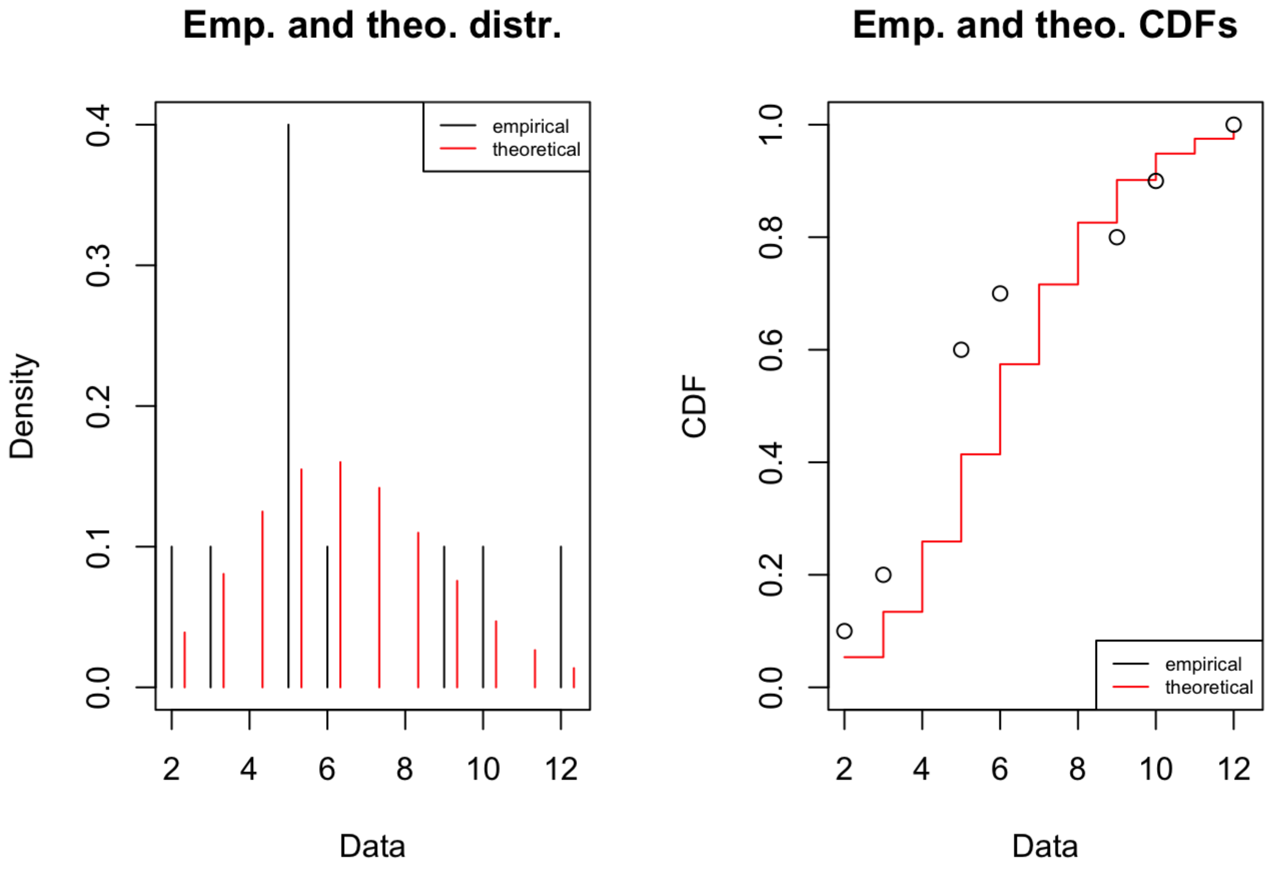
```
n1000hist <- hist(n1000, col="gray", main = "Hist of 1000 samples from the Poisson distr
ibution with the rate 5", xlab="")
```



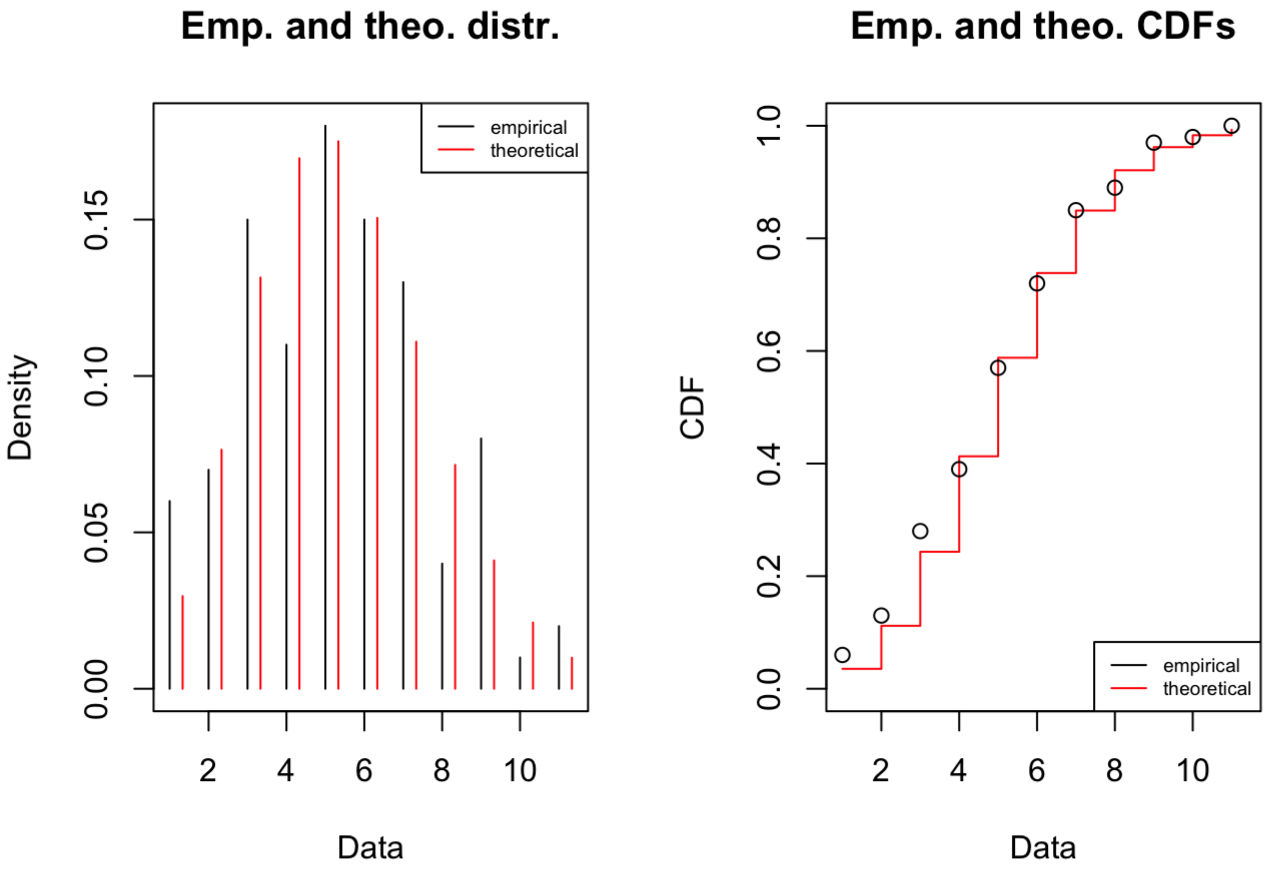
```
n1000hist
```

```
## $breaks
##  [1]  0  1  2  3  4  5  6  7  8  9 10 11 12 13
##
## $counts
##  [1]  48  77 164 180 158 134 112  61  36  17  10   2   1
##
## $density
##  [1] 0.048 0.077 0.164 0.180 0.158 0.134 0.112 0.061 0.036 0.017 0.010 0.002
## [13] 0.001
##
## $mids
##  [1]  0.5  1.5  2.5  3.5  4.5  5.5  6.5  7.5  8.5  9.5 10.5 11.5 12.5
##
## $xname
## [1] "n1000"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

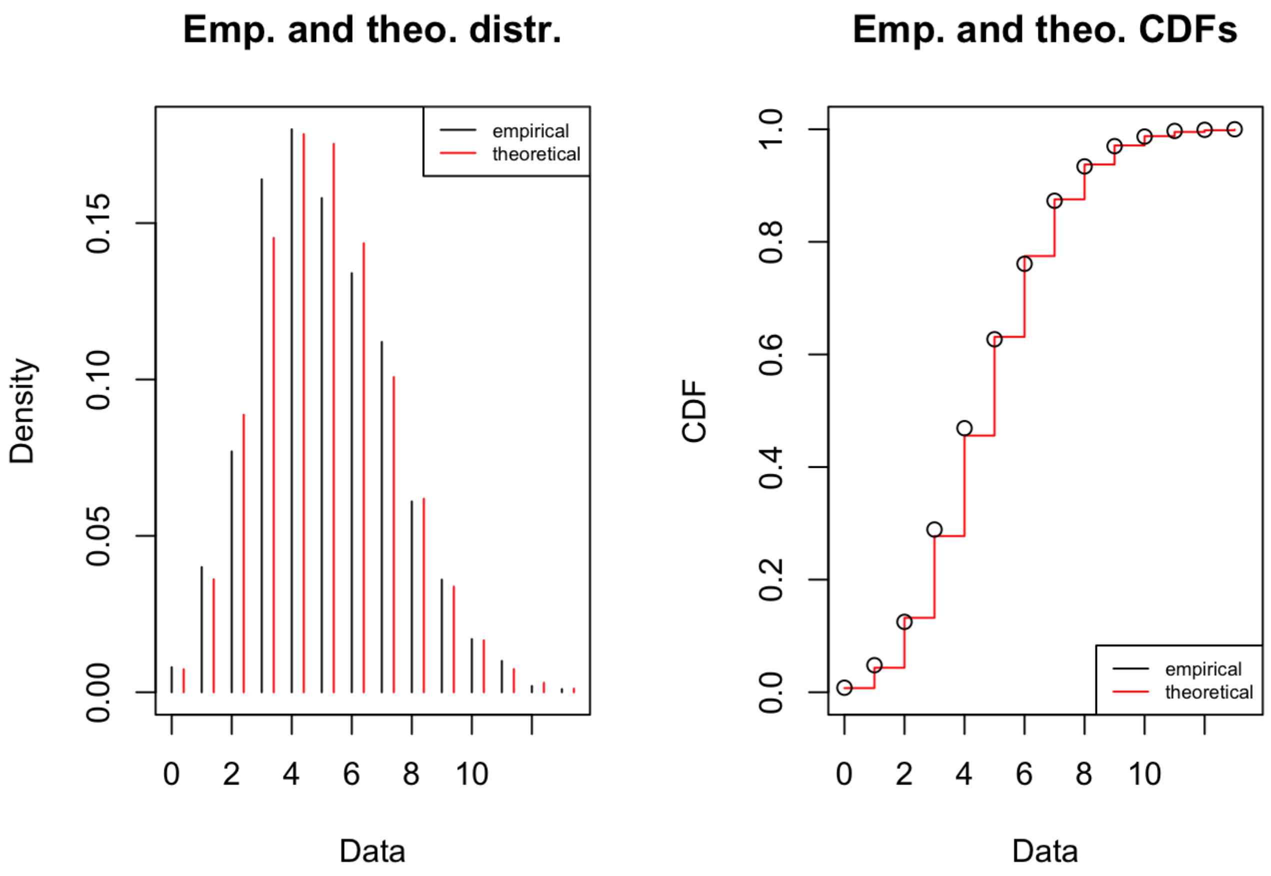
```
qqplotn10 <- plot(fitdist(n10, 'pois')) # Q-Q plot comparing its quantiles (n10) to the
 theoretical quantiles of Poisson distribution
```



```
qqplotn100 <- plot(fitdist(n100, 'pois')) # Q-Q plot comparing its quantiles (n100) to t
he theoretical quantiles of Poisson distribution
```

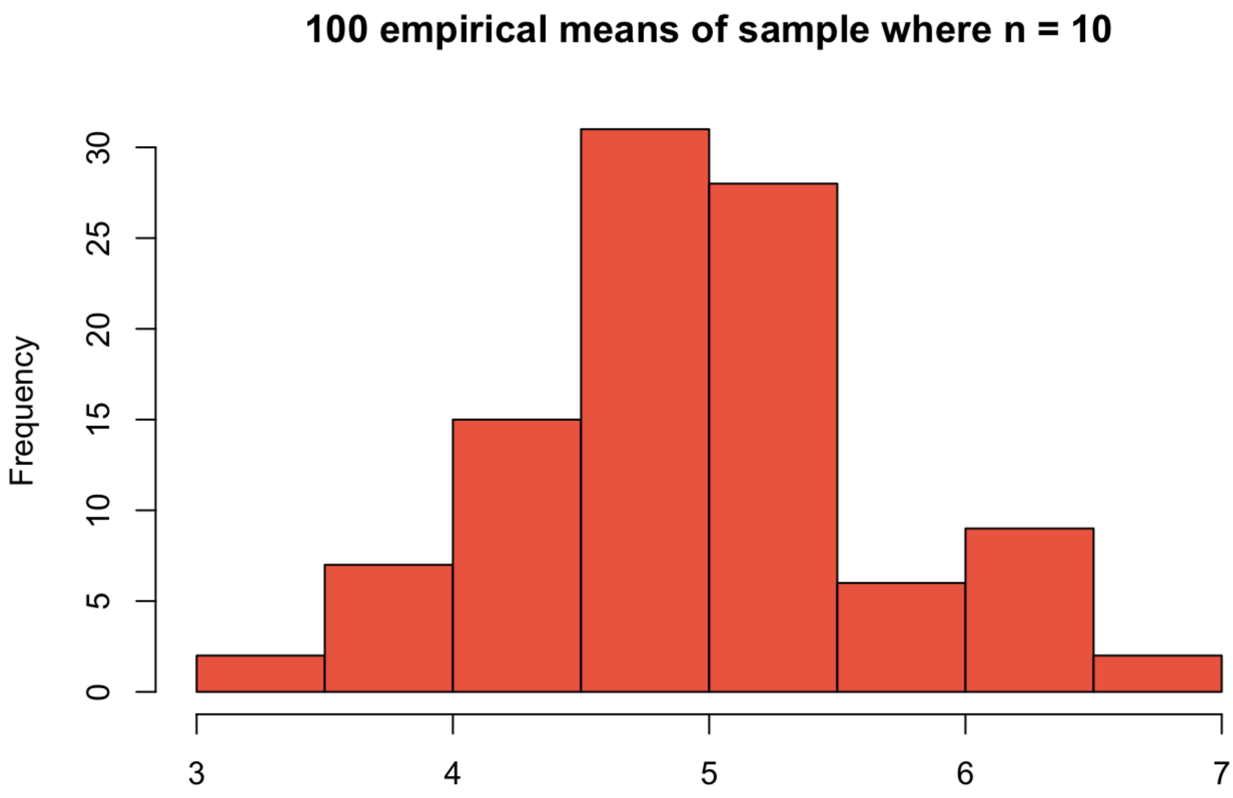


```
qqplotn1000 <- plot(fitdist(n1000, 'pois')) # Q-Q plot comparing its quantiles (n1000) t
o the theoretical quantiles of Poisson distribution
```



b)

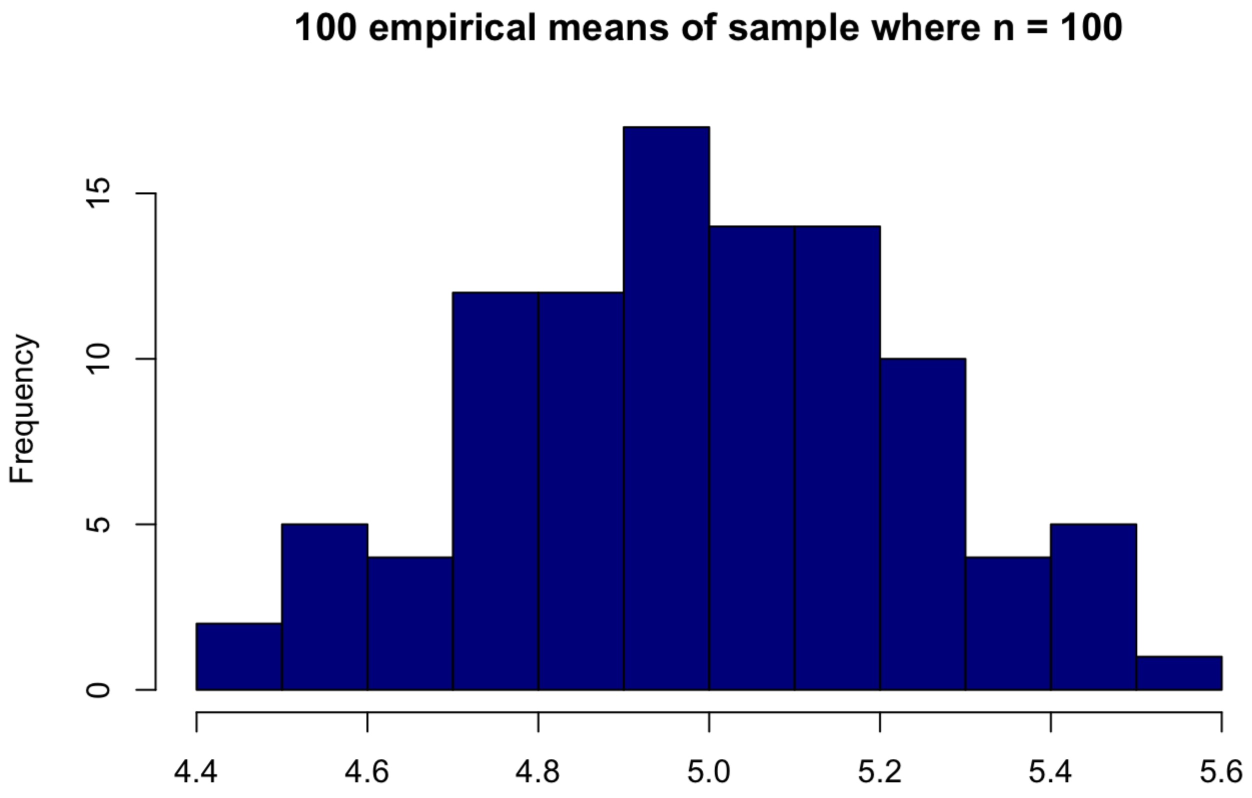
```
sample100n10 <- lapply(1:100, function(x) rpois(10,5))
samplemeann10 <- sapply(sample100n10, mean)
samplemeann10hist <- hist(samplemeann10, col = "coral2", main = "100 empirical means of
 sample where n = 10", xlab="")
```



```
samplemeann10hist
```

```
## $breaks
## [1] 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0
##
## $counts
## [1]  2  7 15 31 28  6  9  2
##
## $density
## [1] 0.04 0.14 0.30 0.62 0.56 0.12 0.18 0.04
##
## $mids
## [1] 3.25 3.75 4.25 4.75 5.25 5.75 6.25 6.75
##
## $xname
## [1] "samplemeann10"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

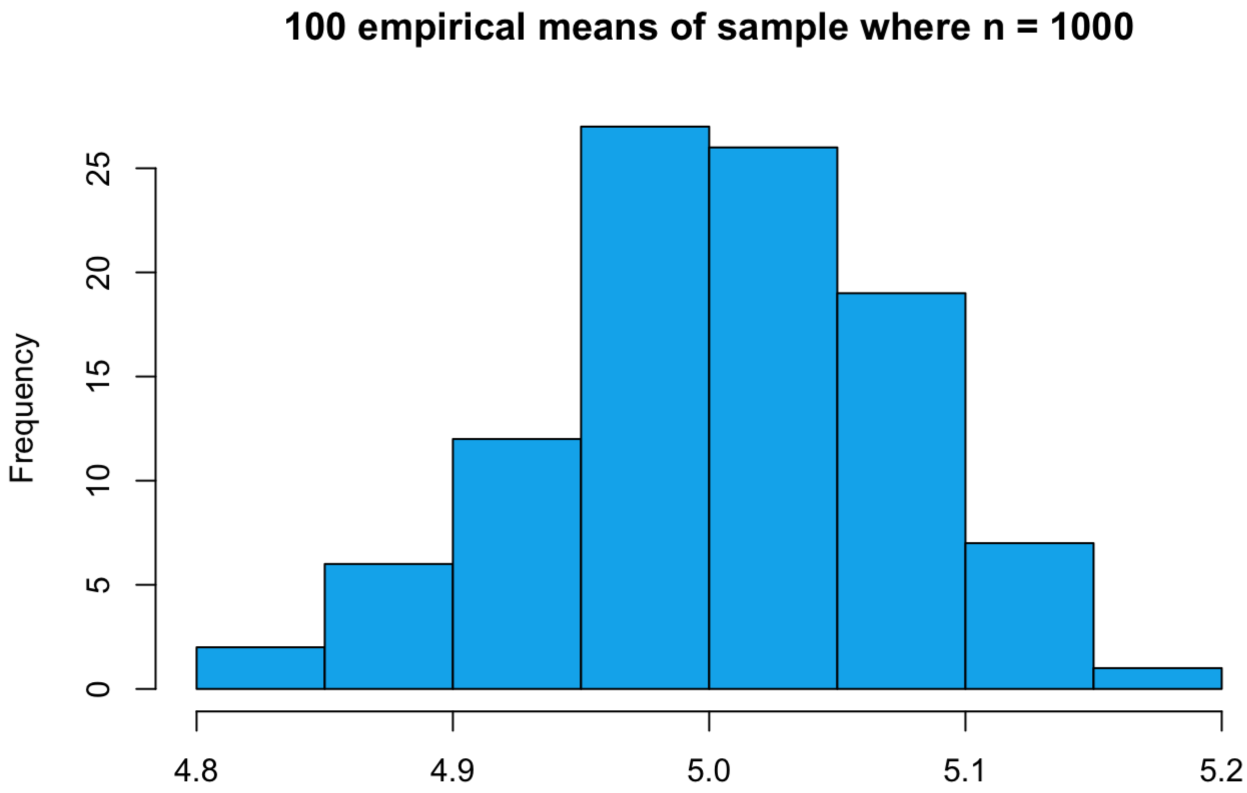
```r
sample100n100 <- lapply(1:100, function(x) rpois(100,5))
samplemeann100 <- sapply(sample100n100, mean)
samplemeann100hist <- hist(samplemeann100, col = "darkblue", main = "100 empirical means
of sample where n = 100", xlab="")
```



```r
samplemeann100hist
```

```
## $breaks
##  [1] 4.4 4.5 4.6 4.7 4.8 4.9 5.0 5.1 5.2 5.3 5.4 5.5 5.6
## 
## $counts
##  [1]  2  5  4 12 12 17 14 14 10  4  5  1
## 
## $density
##  [1] 0.2 0.5 0.4 1.2 1.2 1.7 1.4 1.4 1.0 0.4 0.5 0.1
## 
## $mids
##  [1] 4.45 4.55 4.65 4.75 4.85 4.95 5.05 5.15 5.25 5.35 5.45 5.55
## 
## $xname
## [1] "samplemeann100"
## 
## $equidist
## [1] TRUE
## 
## attr(,"class")
## [1] "histogram"
```

```
sample100n1000 <- lapply(1:100, function(x) rpois(1000,5))
samplemeann1000 <- sapply(sample100n1000, mean)
samplemeann1000hist <- hist(samplemeann1000, col = "deepskyblue2", main = "100 empirical
means of sample where n = 1000", xlab="")
```



```
samplemeann1000hist
```

```
## $breaks
## [1] 4.80 4.85 4.90 4.95 5.00 5.05 5.10 5.15 5.20
##
## $counts
## [1]  2  6 12 27 26 19  7  1
##
## $density
## [1] 0.4 1.2 2.4 5.4 5.2 3.8 1.4 0.2
##
## $mids
## [1] 4.825 4.875 4.925 4.975 5.025 5.075 5.125 5.175
##
## $xname
## [1] "samplemeann1000"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

c. when n increases, the histogram tend to a normal distribution, and this theoretical outcome is justified thanks to the CLT.

Exercise 2

There are two samples, the first one is identified with the individuals in the control group and the second one with the individuals who take the aspirin. We assume that the samples are realization of i.i.d. Bernoulli random variables, $X_1(w), \dots, X_n(w)$ of random variables $X_1, \dots, X_n$ for the former sample with n = 11,034 and $Y_1(w), \dots, Y_m(w)$ of random variables $Y_1, \dots, Y_m$ for the latter sample with m = 11,037. The random variables X_i have as unknown parameter $p_1 \in [0, 1]$ that identifies the probability of having an heart attack and the random variables Y_i have as unknown parameter $p_2 \in [0, 1]$ that identifies the probability of having an heart attack. Therefore, this reasoning leads us to say that the statistical model for the placebo sample is as follow:

$$X \in R, X \sim Binomial(n, p_1)$$

while the statistical model for the aspirin sample is as follow:

$$Y \in R, Y \sim Binomial(m, p_2)$$

Since both of the populations are realizations of n and m Bernoulli random variables respectively. The following process is to formalize the question under an hypothesis test problem. Lets write the contingency table for this problem.

	Control Group	**Aspirin**	**Total**
Heart Attack	189	104	293
NO Heart Attack	10845	10933	21778
Total	11034	11037	22071

The probability of having an heart attack being in the control group is $p_1 = 189/11034 = 0.0171$, while the likelihood of having an heart attack being under aspirin treatment is $p_2 = 104/11037 = 0.0094$, and the probability of having an heart attack in general is $p = 293/22071 = 0.0133$. Therefore the null hypothesis is:

$$H_0 : p_1 = p_2 = p$$

While the alternative hypothesis is:

$$H_1 : p_1 > p_2$$

how many people of the control group would be having an heart attack considering the population percentage of having an heart attack? The calculation is as follow:

$$11034 * 0.0133 = 146.7522 \approx 147$$

while how many people of the treatment group would be having an heart attack considering the population percentage of having an heart attack? The calculation is as follow:

$$11037 * 0.0133 = 146.7921 \approx 147$$

Data observed:

	Control Group	**Aspirin**	**Total**
Heart Attack	189	104	293
NO Heart Attack	10845	10933	21778
Total	11034	11037	22071

Data expected:

	Control group	**Aspirin**	**Total**
Heart Attack	147	147	294
No Heart Attack	10887	10890	21777
Total	11034	11037	22071

Now we need to implement a Chi-squared test where the Chi-squared distribution has 1 degree of freedom:

$$\chi_1^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

where O_i are the observed data, while E_i are the expected ones. The calculation is as follows:

$$\chi_1^2 = \frac{(189 - 147)^2}{147} + \frac{(104 - 147)^2}{147} + \frac{(10845 - 10887)^2}{10887} + \frac{(10933 - 10890)^2}{10890} = 24.9100$$

Considering an $\alpha = 0.05$, the critical value of Chi-squared distribution associated to this value of α is 3.841, yet our value of Chi-squared that we have calculated is bigger than the critical value, and therefore we reject H_0, and we accept H_1. To conclude, we can say that the aspirin has an effect on the heart attack disease.

Exercise 3

Algorithm for simulation

What is the distribution $F^{-1}(U)$, where F^{-1} is the inverse function of F?

To determine the distibution of $F^{-1}(U)$, we need to look at the definition of the quantile function

$$F^{-1}(y) = inf\{x : F(x) \geq y\}, y \in [0, 1]$$

Because $y \in [0, 1]$, it can be mapped through the Uniform random variable U that is uniformally distributed between [0,1].
Now suppose $F^{-1}(U) = X$, where X is a random variable. Then:

$$P(X \leq x) \rightarrow P(F^{-1}(U) \leq x)$$

By taking the inverse to the right of the inequality:

$$P(U \leq F(X))$$

Now since U is a uniform random variable, the probability of $U \leq F(X) = F(X)$. Hence, the distribution of $F^{-1}(U) = X$ is just the distribution F.

Propose an algorithm to sample from the Exponential distribution $Exp(\lambda)$ using a sample from the uniform distribution $U(0, 1)$.

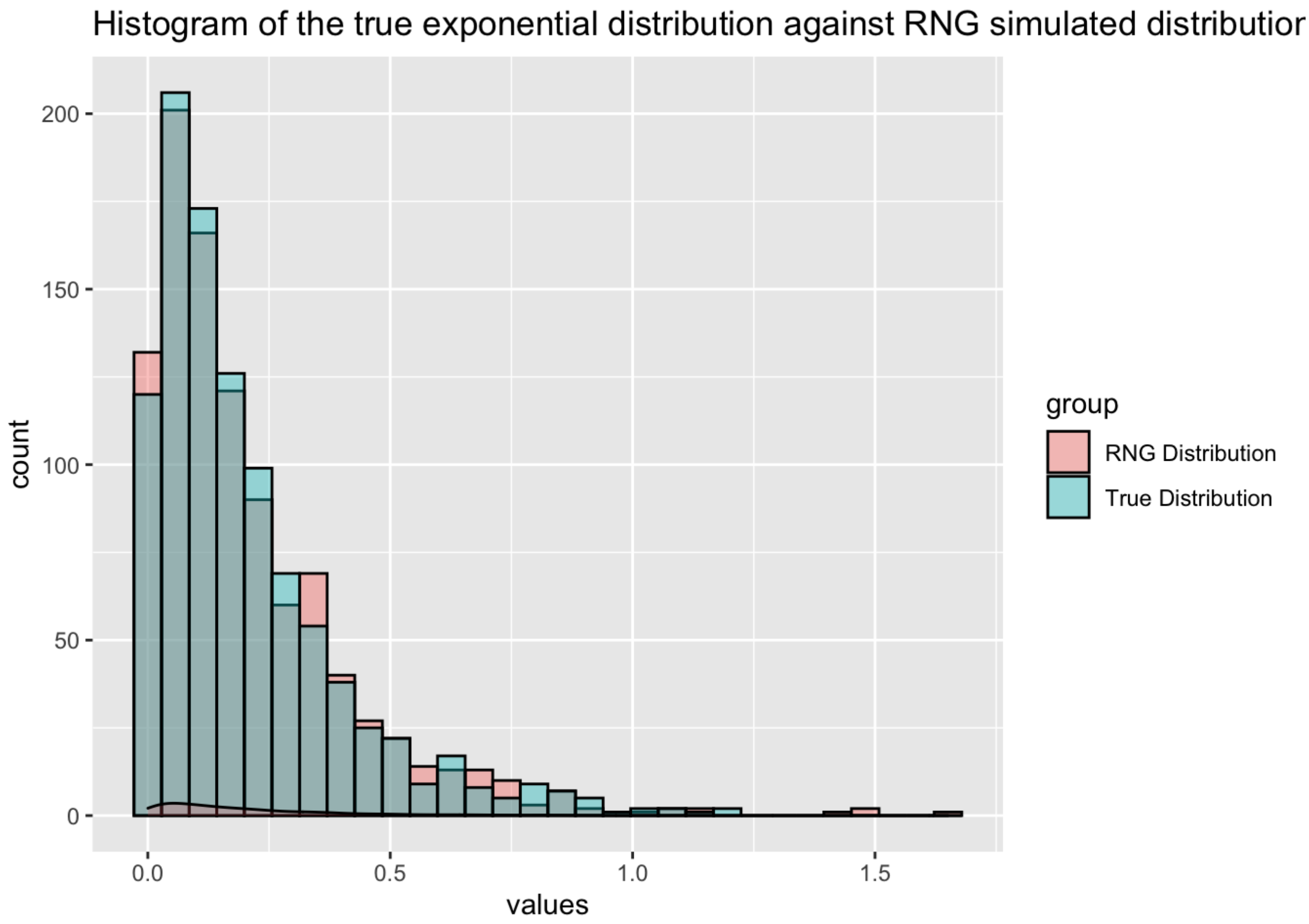
```
#Algorithm for generating exponential distribution through Unif[0,1]
expo_sampler <- function(n,lambda){
  #Empty vector to store the sample
  expo_sample <- c()
  for(i in 1:n){
    #Step 1: Generate a random variable 'X' from the Uniform distribution of [0,1]
    X <- runif(1,min=0,max=1)
    #Step 2: Sample the values from exp(lambda) using it's inverse CDF
    x <- -(log(1-X)/lambda)
    #Step 3: Return the value 'x', which is distributed in 'F', 'F' being the target dis
tribution
    expo_sample[i] <- x
  }
  return(expo_sample)
}
```

Implement this algorithm in R, and use it to generate a 1000-sample distributed according to $Exp(\lambda)$. Compare the distribution of the sample to $Exp(\lambda)$ using a graph.

```
library(ggplot2)
#First, we generate a sample from the exponential distribution of lambda = 5
true_expo_dist <- rexp(1000, rate=5)
#Next, we generate a sample from the exponential distribution we generate with our RNG a
lgorithm
estimated_expo_dist <- expo_sampler(1000, 5)


#Let's observe them to see how similar they are:
distributions <- data.frame(values = c(true_expo_dist,estimated_expo_dist),
                            group = c(rep("True Distribution",1000),
                                      rep("RNG Distribution",1000)))
ggplot(distributions, aes(x=values, fill=group))+
  geom_histogram(position = "identity", alpha=0.4,colour="black")+
  geom_density(alpha=.2, fill="#FF6666")+
  ggtitle("Histogram of the true exponential distribution against RNG simulated distribu
tion")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



####

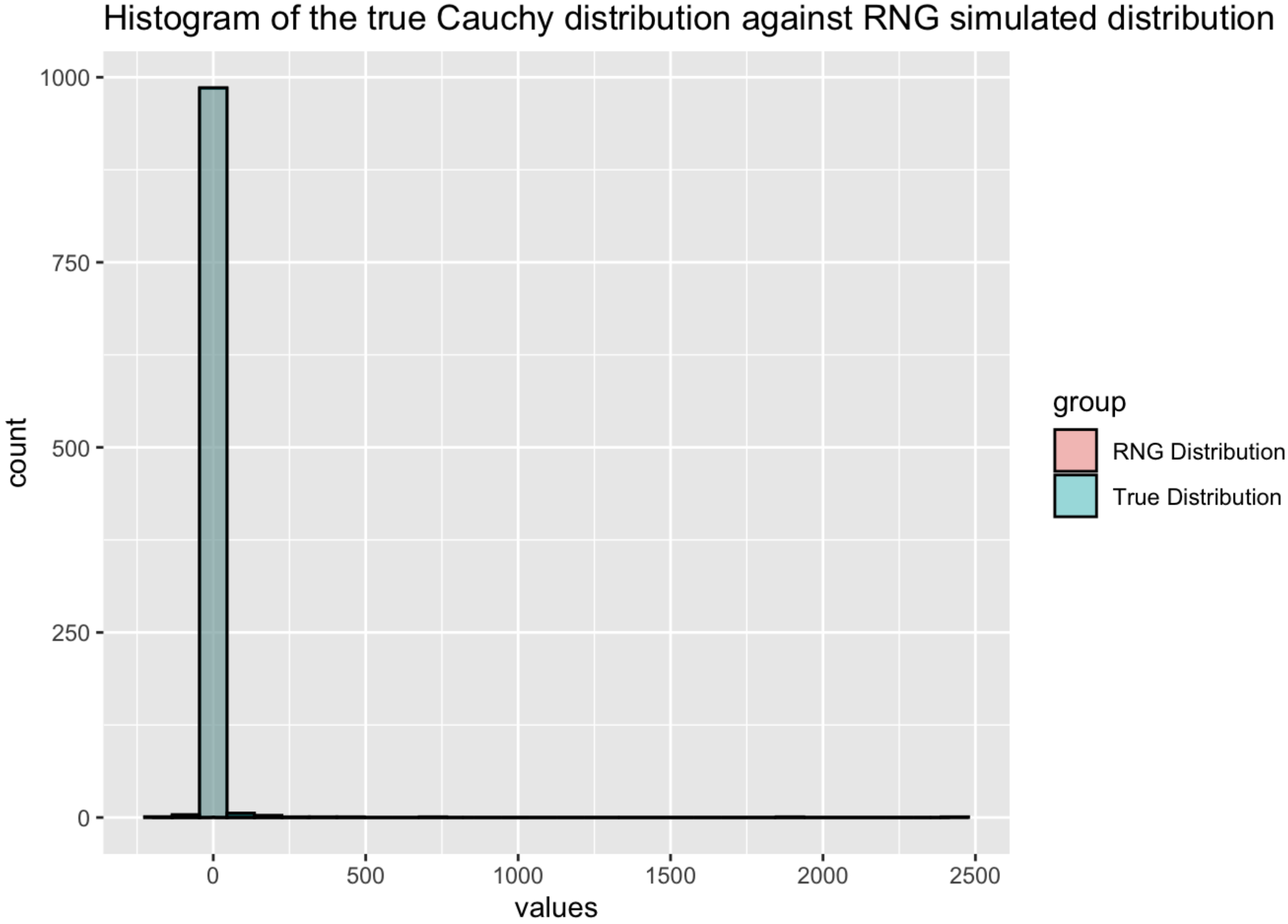
Propose an algorithm to sample from a Cauchy distribution using a sample of the uniform distribution $U(0, 1)$.

```
#Algorithm for generating exponential distribution through Unif[0,1]
cauchy_sampler <- function(n,x0,g){
  #Empty vector to store the sample
  cauchy_sample <- c()
  for(i in 1:n){
    #Step 1: Generate a random variable 'X' from the Uniform distribution of [0,1]
    X <- runif(1,min=0,max=1)
    #Step 2: Sample the values from exp(lambda) using it's inverse CDF
    x <- g*tan((2*X*pi-pi)/2)+x0
    #Step 3: Return the value 'x', which is distributed in 'F', 'F' being the target dis
tribution
    cauchy_sample[i] <- x
  }
  return(cauchy_sample)
}
```

```
library(ggplot2)
#Let's plot the Cauchy distribution against the generated Cauchy samples
true_cauchy_dist <- rcauchy(1000,location=0,scale=1)
estimated_cauchy_dist <- cauchy_sampler(1000, 0,1)


#Let's observe them to see how similar they are:
distributions <- data.frame(values = c(true_cauchy_dist,estimated_cauchy_dist),
                            group = c(rep("True Distribution",100),
                                      rep("RNG Distribution",100)))
ggplot(distributions, aes(x=values, fill=group))+
  geom_histogram(position = "identity", alpha=0.4,colour="black")+
  geom_density(alpha=.2, fill="#FF6666")+
  ggtitle("Histogram of the true Cauchy distribution against RNG simulated distribution"
)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Exercise 4

a.

Under the assumption that the data are realization of i.i.d. Normal random variable, $X_1(w), \ldots, X_n(w)$ of random variable $X_1, \ldots, X_n$. The random variables X_i have as unknown parameters $(\mu, \sigma^2) \in R$. Therefore, the statistical model is:

$$P = \{R^+, N(\mu, \sigma^2), (\mu, \sigma^2) \in R \times R^+\}$$

Moreover, the model is identifiable since the mean and the variance identify the two parameters of the distribution and uniquely generate the distribution of the data.

b.

The equation of the measurement

$$Y = X_i + \epsilon_i$$

where $\epsilon_i \sim N(0, \sigma_e^2)$, and $X \sim N(\mu, \sigma^2)$ is the outcome. We propose as estimator for the area of the field the following estimator:

$$\left(\frac{x_1 + x_2}{2}\right)^2$$

This is the best estimator among the proposed one, since increasing n the variance will be reduced. $T(F) = h(Eg(X))$, where $h(x) = x^2$ and $g(x) = x$ with $E(g(X)) < \infty$.

c.

When we consider n measurements the estimator will be:

$$\frac{1}{n^2}\sum_{i=1}^{n}(x_i)^2$$

Now lets study the asymptotic properties: considering the fact that we are in the case of regular functional,

$$T(F) = h(Eg(X))$$

where $h, g : R \to R$, h continuous and $E|g(X)| < \infty$, then:

$$T(\hat{F_n}) \overset{a.s.}{\to} T(F)$$

Given that the above estimator is just the generalization of s_4 mapped through a continuous function x^2 , the strong consistency is easy to determine since we know that for the empirical mean:

$$M_n x \overset{a.s}{\to} \mu$$

Then with the help of Continuous Mapping Theorem, when the function mapping x^2 is applied, strong consistency is still guaranteed such that:

$$M_n^2 x \overset{a.s}{\to} \mu^2$$

Therefore, the estimator is consistent.

Moreover, since $E|g(X)| < \infty$, and h is a continuous and differentiable function in $E(g(X))$, then:

$$\sqrt{n}(T(\hat{F_n}) - T(F)) \overset{d}{\to} N(0, v(F))$$

where $v(F) = 4\sigma^2\mu^2$ via the help of the Delta method.

Therefore, the estimator is asymptotically normal, where the asymptotic variance is $4\sigma^2\mu^2$. Furthermore, an estimator that is asymptotically normal will have an approximately normal distribution as the sample size gets infinitely large, that is the case of our estimator since the output X is a sum of n normally distributed variables.

Exercise 5

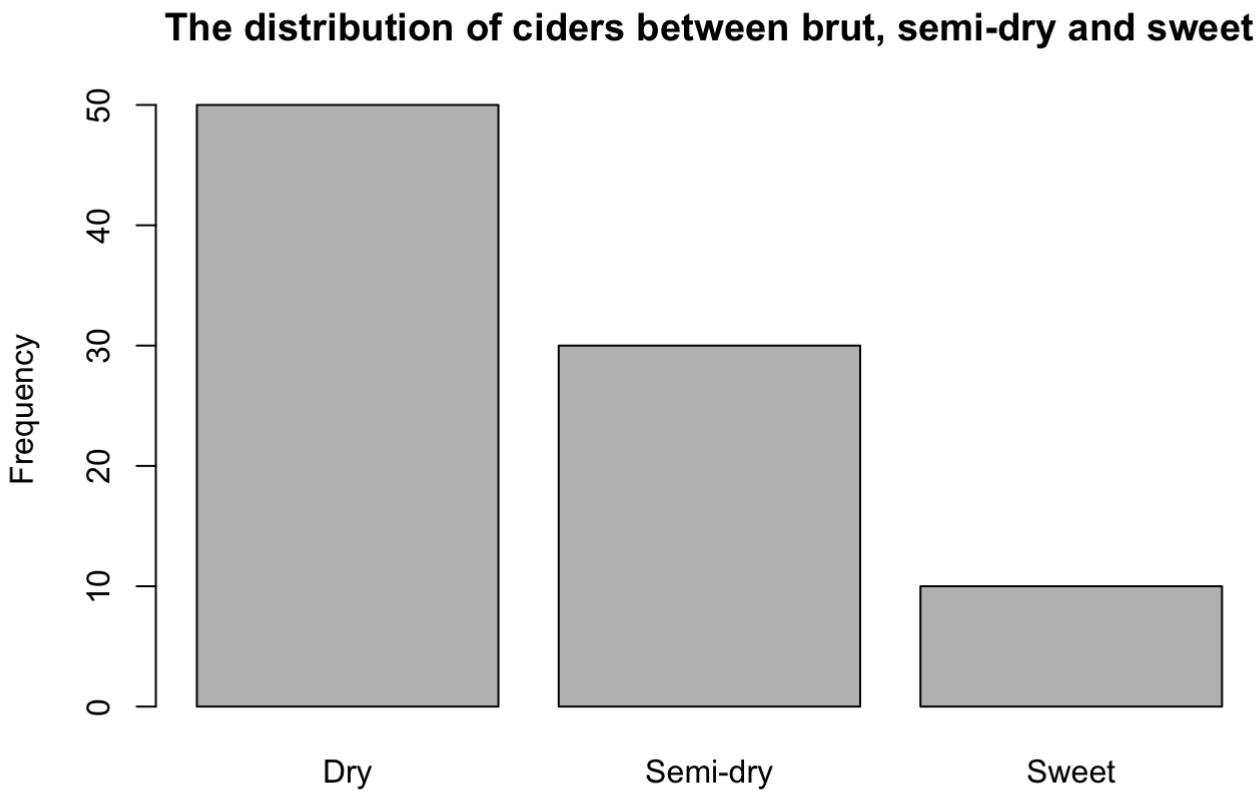
a.

```
df_cinder <- read.csv(file ='cider.csv')
head(df_cinder)
```

```
##   Type Sweetness     Acid Bitterness Astringent
## 1  Dry  4.678571 3.392857   5.857143  2.3214286
## 2  Dry  5.285714 4.285714   4.857143  2.7857143
## 3  Dry  6.500000 4.642857   2.357143  0.7142857
## 4  Dry  5.035714 5.714286   4.642857  3.8214286
## 5  Dry  4.571429 4.607143   4.321429  2.1785714
## 6  Dry  6.071429 3.250000   3.428571  1.1428571
```

b.

```
bar_plot <- barplot(table(df_cinder$Type), main = "The distribution of ciders between br
ut, semi-dry and sweet", ylab = "Frequency")
```

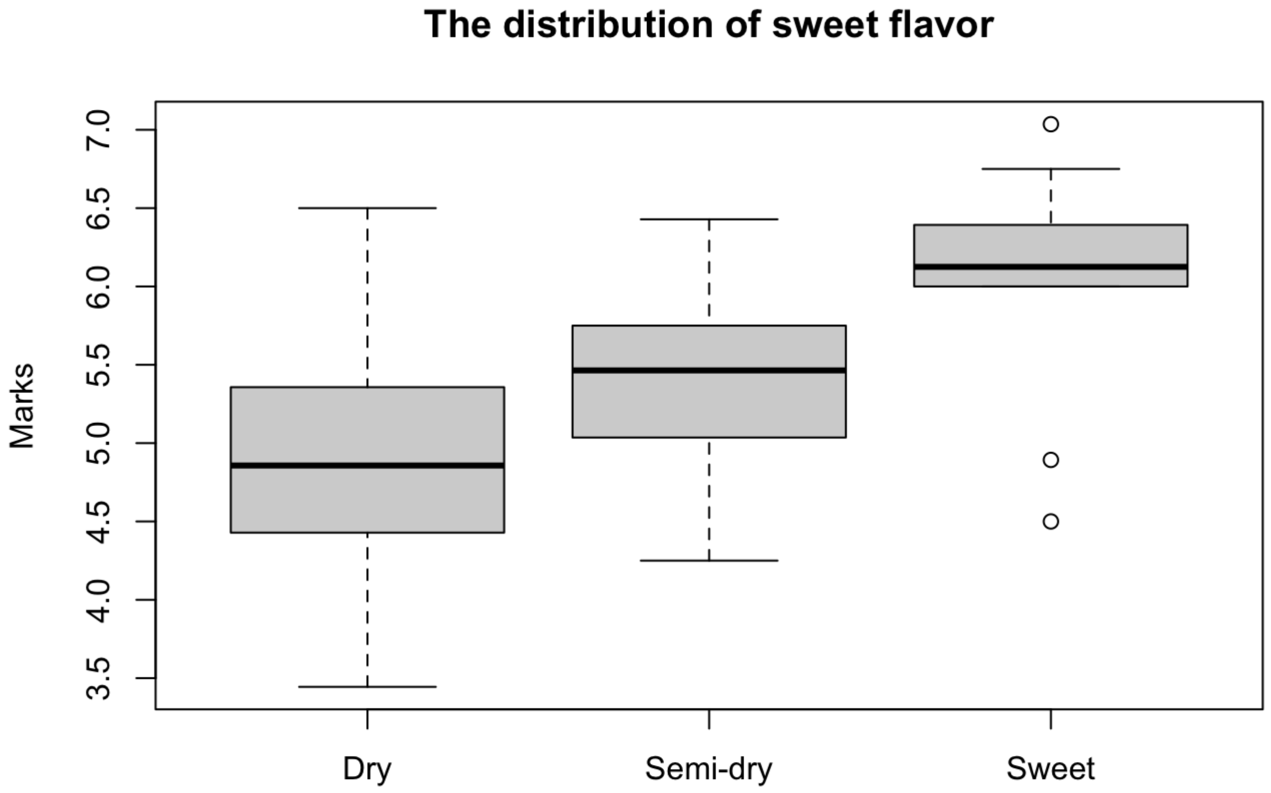


```
bar_plot
```

```
##      [,1]
## [1,]  0.7
## [2,]  1.9
## [3,]  3.1
```

c.

```
box_plot <- boxplot(df_cinder$Sweetness~df_cinder$Type, main = "The distribution of swee
t flavor", xlab="", ylab = "Marks")
```


The distribution of sweet flavor
Marks
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
Dry
Semi-dry
Sweet

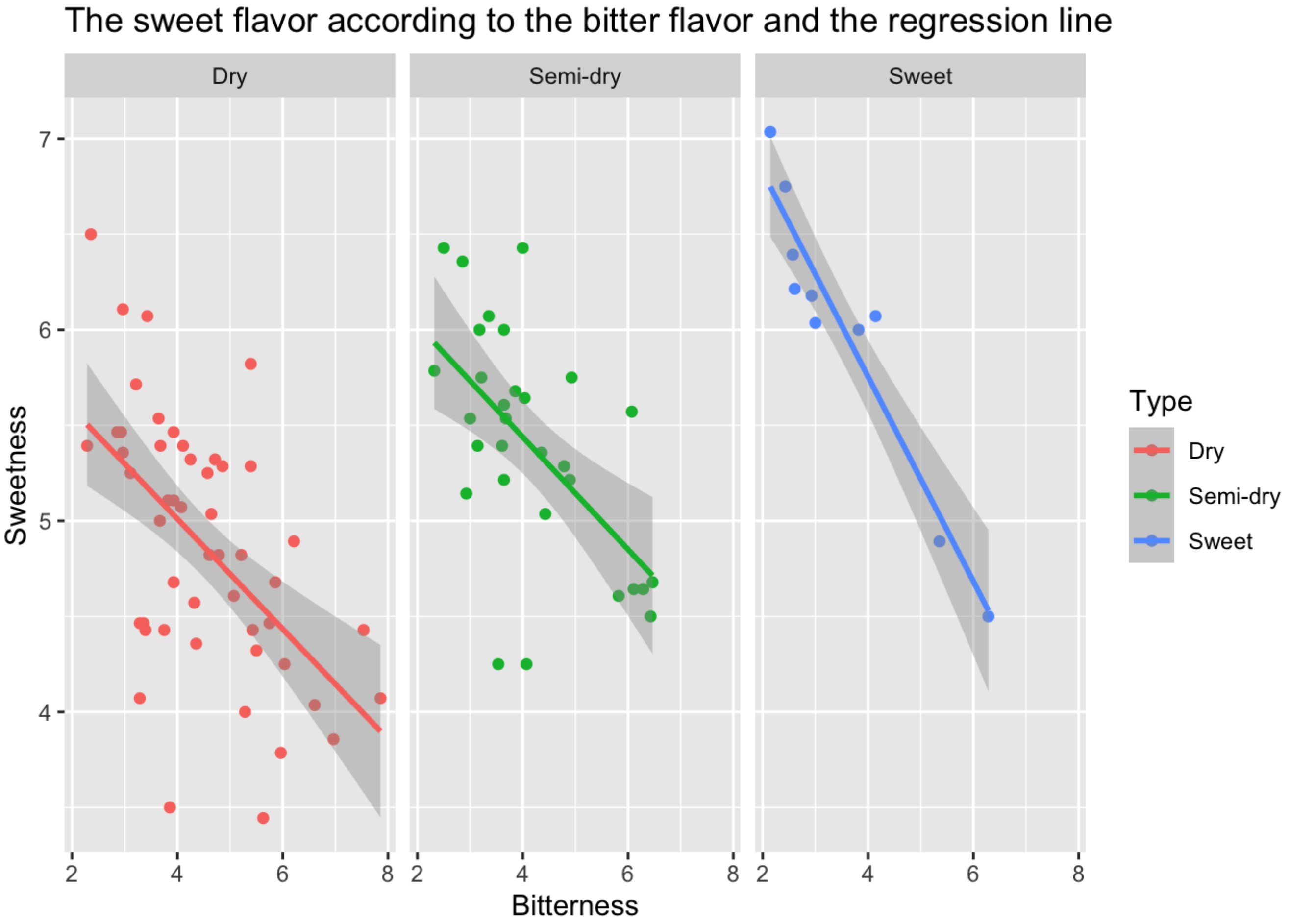
```
box_plot
```

```
## $stats
##          [,1]     [,2]     [,3]
## [1,] 3.444444 4.250000 6.000000
## [2,] 4.428571 5.035714 6.000000
## [3,] 4.857143 5.464286 6.125000
## [4,] 5.357143 5.750000 6.392857
## [5,] 6.500000 6.428571 6.750000
## 
## $n
## [1] 50 30 10
## 
## $conf
##          [,1]     [,2]     [,3]
## [1,] 4.649658 5.258238 5.928713
## [2,] 5.064628 5.670334 6.321287
## 
## $out
## [1] 4.892857 4.500000 7.035714
## 
## $group
## [1] 3 3 3
## 
## $names
## [1] "Dry"      "Semi-dry" "Sweet"
```

d.

```
library(ggplot2)
sweet_bitter <- ggplot(df_cinder, aes(x=Bitterness, y=Sweetness, color=Type)) + geom_poi
nt() + geom_smooth(method='lm') +ggtitle("The sweet flavor according to the bitter flavo
r and the regression line") + facet_wrap(~ Type)
sweet_bitter
```

```
## `geom_smooth()` using formula 'y ~ x'
```



e.

We can say after plotting the graph above that the sweetness and bitterness are negative correlated. Sweeter the wine, less bitter it is, and it makes sense since sweeter the wine less alcoholic it is.