SDA- Assignment 1

Jose Chacon (2699643)

Exercises

Exercise 1.5

The first part of this exercise was to create a function called *CLT_unif* with parameters n and m. This function will do 3 main things. First, it will draw n samples of size m from the uniform distribution on the interval (0,1). This was achieved by using the combination of two methods: *replicate* and *runif*. The *runif* method will draw samples of size "m" from the uniform distribution in the interval (0,1), and the method *runif* will repeat the process "m" times.

Secondly, this function will also plot a scaled histogram of the n sample-specific means. Therefore, we have to calculate the sample of the "m" samples; we do this with the function *colMeans* (it will calculate the mean of the columns of a data frame; in this case dim(data frame)=1, so we get exactly what we want).

Finally, the function will also plot the curve of a normal distribution with a mean of 0.5 and a standard deviation of $1/\sqrt{12 * m}$. To create this plot we have to define what the x and y-axis should look like. First x, is a sequence of numbers starting in the minimum of the samples and ending in the maximum of the samples, with the continuous range in-between. On the other hand, the y-axis will create the form of the known normal distribution with the help of *dnorm*, which evaluates the density of a normal distribution at a specific point.

For the second part of the exercise, we called the function four times with different parameters to understand its behavior. Shortly, the 4 plots will be displayed. The following parameters were used:

- n=50, m=30
- n=50, m=200
- n=300, m=30
- n=300, m=200

Conclusion: Depending on the m and n parameters, the histogram and the normal distribution curve will change. One perfect example to illustrate this is the changes in the histogram if we increase n. Then the size of each sample will increase and the histograms will have more items to represent the distribution of the sample means. Therefore, it will resemble the noraml distribution. Conversely, if we decrease n, we will have less bars (and info).

Similarly, if we increase the parameter m, then the size of each sample will also increase. Therefore, the standard deviation of the expected distribution of the sample means will decrease. This will create a narrower normal density curve. In conclusion, if m increases we approach the normal distribution with mean 0.5 and a standard deviation of $1/\sqrt{12 * m}$.

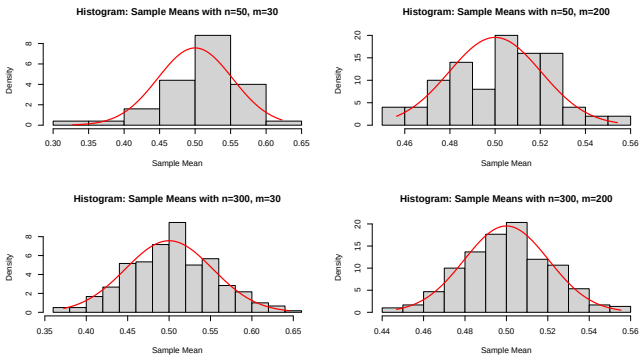


Figure 1: Histograms with different parameters 1.5

Exercise 1.6

In this last exercise, we were given a data frame with the military spending per capita. We have to perform graphical and numerical studies in different subsets of the data frame. First, we analyzed a univariate data frame that contained the military expenses per capita in 2020. With help of the function *summary*, we got all the general information about the data frame. The results are presented in the next table.

Statistic	Value
Min.	0.00
1st Qu.	19.98
Median	82.00
Mean	263.28
3rd Qu.	319.35
Max.	2507.58
st Dev.	427.35

Table 1: Summary Statistics: Military expenditure 2020

We also created some graphical representations of the data set to better understand it. A boxplot, a histogram, and a scatter plot were created. These plots will be presented.

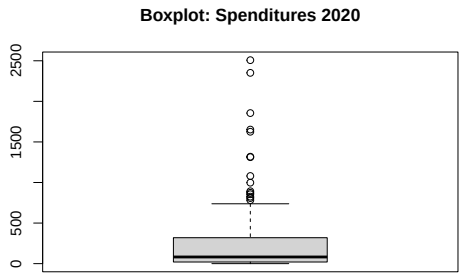
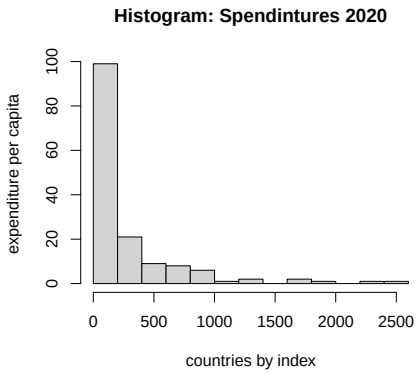
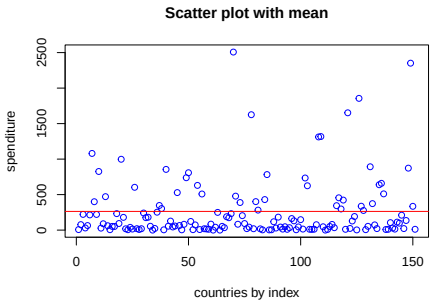


Figure 2: Boxplot of the military expenditure 2020



[Histogram]



[Scatterplot]

Figure 3: Graphical Summaries for the Military Expenditure

Secondly, we performed an analysis in a bivariate data frame. In this case, we had two years to compare: 1988 and 2020. We also used the same method *summary*. The results are displayed in the next table. Lastly, we also computed the correlation of these two datasets, which is 0.91. This means that both data sets are very highly correlated. Additionally, we also performed a graphical description of the data set. We did this by creating a scatterplot having the values of 1988 as the x-axis and the values of 2020 as the y-axis. Before plotting this graph, we made sure that we deleted all rows of the data frame that were not completed. We achieved this with the method *na.omit*.

	1988	2020
Min.	0.00	0.00
1st Qu.	11.33	20.29
Median	34.86	82.55
Mean	175.80	317.22
3rd Qu.	229.92	422.88
Max.	1650.15	2507.58

Table 2: Summary Statistics 1988 vs 2020

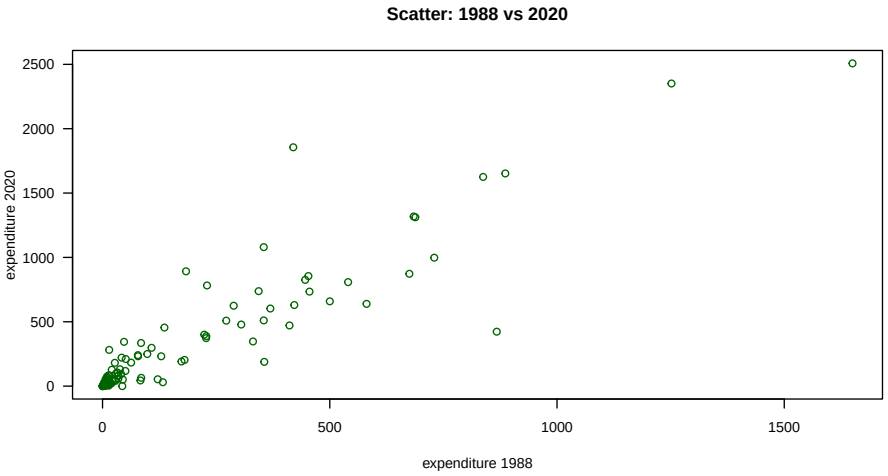


Figure 4: Scatterplot with a mean line

Appendix

In this first section of the appendix, we can find the code for the assignment 1.4

0.1 code exercise 1.4

```
norm <- function(n, mu, signma){
  set.seed(2023210)
  samples = rnorm(n, mu, signma)[1:n]
  quants = quantile(samples,probs= c(0.05,0.5,0.95), type=7)
  loc = mean(samples)
  spread = sd(samples)
  stud_no = 2699643
  myList = list(quants, loc, spread, stud_no)
  save(myList, file="C:\\Users\\josec\\OneDrive\\Documentos\\VU\\Statisitcal Data Analysis\\Ass
}

norm(100, 1, 1)
```

0.2 Code exercise 1.5

```
CLT_unif <- function(n, m) {
  samples = replicate(n, runif(m))
  means = colMeans(samples)

  hist(means, main=paste("Histogram: Sample Means with n=", n, ", m=", m, sep=""),
  xlab="Sample Mean", prob=TRUE)
```

```
  x = seq(min(means), max(means), length.out=100)
  y = dnorm(x, mean=0.5, sd=1/sqrt(12*m))
  lines(x, y, type="l", col="red", lwd=2)
}
par(mfrow=c(2,2))
CLT_unif(50,30)
CLT_unif(50,200)
CLT_unif(300,30)
CLT_unif(300,200)
```

0.3 Code exercise 1.6

This part of the is for the univariate data frame.

```
summary(ME2020)
sd(ME2020)
var(ME2020)

hist(ME2020, main="Histogram: Spendintures 2020",
xlab='countries by index', ylab='expenditure per capita')

boxplot(ME2020, main="Boxplot: Spenditures 2020")

#scatter plot with a mean line
x = seq(1,length(ME2020),1)
plot(x, ME2020, col='blue', main="Scatter plot with mean",
    xlab='countries by index',ylab='spenditure')

abline(h=avg_spending_2020, col='red')
```

This is the code for the bivariate data frame.

```
temp = na.omit(military_bivariate)

summary(temp)
correlation = cor(temp$military_expenditure_1988,temp$military_expenditure_2020)

#plotting spenditure 1988 vs 2020
plot(temp$military_expenditure_1988, temp$military_expenditure_2020,
    main="Scatter: 1988 vs 2020", xlab='expenditure 1988', ylab='expenditure 2020', col='dark g
```