Laboratory Activities
Statistical Methods for Engineers

Nicolás Aguado González
UPV/EHU

December 21, 2022

Contents

0 Introduction

Throughout the subject, we have studied several ways of analyzing the data from the world around us. From descriptive statistics, all the way to inference and probability. Now, we'll summarize all the knowledge and use it to solve several activities in different real-world scenarios. In order to help us process all the math and images, *RStudio* will be used, providing pieces of *R* code along the explanations. The reader is advised to *copy-paste* only the source code contained in the appendix, as otherwise line-breaks may cause some trouble.

1 First Scenario

For this first set of activities, we have to consider the following two games:

- In Game A you flip a fair coin. If the coin comes up Heads you get two dollars, whereas if it comes up Tails you get one dollar.
- In Game B you roll a fair die. If the six-spot comes up, you win twenty-five dollars. If you get 2, 3, 4, or 5, nothing happens. If the one-spot comes up, you lose fifteen dollars.

Now, when understanding the two games, the next two *intuitions* come to mind:

1. If I were to play either Game A or Game B only *once*, I would choose to play *Game B*. That is because when choosing Game A, I'd have 50-50 chance of winning either a dollar or two dollars. But, with Game B, I'd have less chance, but when winning, I'd win 25 dollars. I could also lose 15 dollars, but that's a risk I'm willing to take.
2. For playing a game in the long run (10 000 times), I'd also choose *Game B*. That's because at most, in Game A, I would win 2 dollars, but in Game B, in the long run, I could win $25 - 15 = 10$ dollars for an extended period of time. And because $10 > 2, 4, 6, 8$; without doing extended math, I think the expected value is worth more.

1.1 Game Analysis (x; f(x); E[x] and VAR(x))

Now we'll examine both games. For each one of them, we'll define its random variable, its respective probability mass function and its plot. Also, for both games we'll analyze the expected value and variance.

1.1.1 Game A

For the fair coin, we'll define the random variable as

$$X = \text{Dollars obtained when tossing a fair coin.}$$

From this definition, the probability mass function (p.m.f.) of the game will be:

$$f(x) = \begin{cases} 0,5 & x = 1 \\ 0,5 & x = 2 \\ 0 & \text{otherwise} \end{cases}$$

And we'll plot it like this:

```
moneya <- c(1, 2)
gamea <- c(0.5,0.5)
barplot(gamea, names.arg=moneya, xlab="Money Won (USD$)",
↪  ylab="Probability", ylim=c(0,1))
```

Figure 1: Game A - Barplot

Then, we'll find the *sample* expected value and the *sample* variance of the p.m.f:

```
expecteda <- weighted.mean(moneya, gamea)
vara <- var(moneya)
```

$$E[X] = 1,5 \qquad \text{VAR}(X) = 0,5$$

From this values we can interpret that on average we'll win 1,5 dollars when tossing the coin.

1.1.2 Game B

For the fair die, we'll define the random variable as

$$Y = \text{Resulting dollars when rolling a fair die.}$$

From this definition, the probability mass function (p.m.f.) of the game will be:

$$f(x) = \begin{cases} \frac{4}{6} & x = 0 \\ \frac{1}{6} & x = 25 \\ \frac{1}{6} & x = -15 \end{cases}$$

And we'll plot it like this:

```
moneyb <- c(0, 25, -15)
gameb <- c(4/6, 1/6,1/6)
barplot(gameb, names.arg=moneyb, xlab="Resulting Money (USD$)",
↪  ylab="Probability", ylim=c(0,1))
```

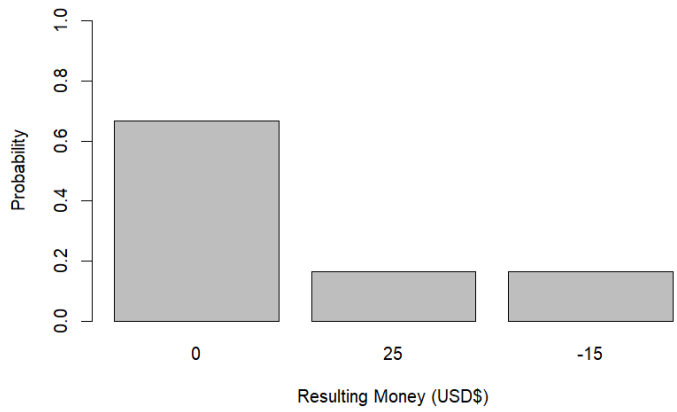

Figure 2: Game B - Barplot

Then, we'll find the *sample* expected value and the *sample* variance of the p.m.f:

```
expectedb <- weighted.mean(moneyb, gameb)
varb <- var(moneyb)
```

$$E[Y] = 1,\overline{6} \qquad \text{VAR}(Y) = 408,\overline{3}$$

From this values we can interpret that on average we'll win money, and we'll get $1,\overline{6}$ dollars when rolling the fair die.

1.2 Sample Simulation

Now, we'll simulate the execution of both games, when the variable n is the times we play. Each simulation will be performed 5 times. We'll analyze each time the resulting mean.

```
# Game A
mean(sample(moneya, prob=gamea, size=n, replace=TRUE))
# Game B
mean(sample(moneyb, prob=gameb, size=n, replace=TRUE))
```

- When we play $n = 100$ times:

Repeat	1	2	3	4	5	$E[X/Y]$
Game A: $\overline{x}$	1,55	1,42	1,53	1,44	1,43	1,5
Game B: $\overline{y}$	0,9	2,9	1,4	2,1	1,7	$1,\overline{6}$

- When we play $n = 1000$ times:

Repeat	1	2	3	4	5	$E[X/Y]$
Game A: $\overline{x}$	1,534	1,493	1,487	1,509	1,481	1,5
Game B: $\overline{y}$	2,24	1,965	1,465	1,665	1,645	$1,\overline{6}$

- When we play $n = 10000$ times:

Repeat	1	2	3	4	5	$E[X/Y]$
Game A: $\overline{x}$	1,493	1,496	1,493	1,494	1,504	1,5
Game B: $\overline{y}$	1,548	1,574	1,573	1,628	1,608	$1,\overline{6}$

We can check that when we repeat more times the experiment, $\overline{x}$ and $\overline{y}$ move closer to the expected value for each game. Game A will also give closer results more quickly. That is because VAR(X) < VAR(Y)

1.3 Probability of the Jackpot in the long run

Now we'll calculate the probability for each game of winning more than 5,000 dollars or more if we play it 10,000 times.

We'll represent the situations as:

$$S^x_{10000} = X_1 + X_2 + \cdots + X_{10000}$$

$$S^y_{10000} = Y_1 + Y_2 + \cdots + Y_{10000}$$

And, because of the Central Limit Theory, we can approximate both situations to the normal distribution:

$$S^x_{10000} \sim N(10000\mu_x, \sqrt{10000\sigma_x^2})$$
$$S^y_{10000} \sim N(10000\mu_y, \sqrt{10000\sigma_y^2})$$

So, we need to find $P(S^x_{10000} > 5000)$ and $P(S^y_{10000} > 5000)$. For that, we'll take advantage of the *pnorm* function in the R programming language, and we'll reuse the values calculated previously for the mean and variance.

```
# Game A
mn <- 10000*expecteda
std <- sqrt(10000*vara)
proba <- 1-pnorm(5000, mean=mn, sd=std)
# Game B
mn <- 10000*expectedb
std <- sqrt(10000*varb)
probb <- 1-pnorm(5000, mean=mn, sd=std)
```

Upon execution of this code, we get:

$$P(S^x_{10000} > 5000) = 1$$

$$P(S^y_{10000} > 5000) = 1$$

We can conclude that almost certainly we'll win 5000 dollars with Game A. When talking about Game B, even though the variance is so big, we can also assure we're winning 5000 dollars when playing 10 000 times.

2 Second Scenario

The *adult* data set is a famous dataset from the *UCI - machine learning repository*. The idea is to predict whether income exceeds \$50K/yr based on census data. This dataset is also known as the "Census Income" dataset. Extraction was done by Barry Becker from the 1994 Census database.

```
adult <- read.table("adult.csv", col.names=c("age",
↪  "workclass", "fnlwgt", "education", "education-num",
↪  "marital-status", "occupation", "relationship", "race",
↪  "sex", "capital-gain", "capital-loss", "hours-per-week",
↪  "native-country", "50K-Prediction"), sep=",", header =
↪  FALSE)
```

Now, we'll examine one variable from the data set and we'll analyze it in depth:

2.1 Hours per Week Worked

From all the possible characteristics an adult can have, the variable that impacts me the most is the hours-per-week worked. It's a continuous variable that represents how much an adult spends working, per week.

```
numdatapoints <- length(adult$hours.per.week)
```

We're working with 32 561 data points, and thus, with 32 561 individuals.

2.1.1 Data Visualization

Now, to visualize and understand better the data, we'll plot it:

```
# Layout to split the screen
layout(mat = matrix(c(1,2),2,1, byrow=TRUE),  height = c(1,8))
# Draw the boxplot and the histogram
par(mar=c(0, 3.1, 1.1, 2.1))
boxplot(adult$hours.per.week , horizontal=TRUE , xaxt="n" ,
↪  col=rgb(0.8,0.8,0,0.5) , frame=F)
par(mar=c(4, 3.1, 1.1, 2.1))
hist(adult$hours.per.week , breaks=25 ,
↪  col=rgb(0.2,0.8,0.5,0.5) , border=F , main="" , xlab="Hours
↪  per Week Worked", ylab="Individuals")
```

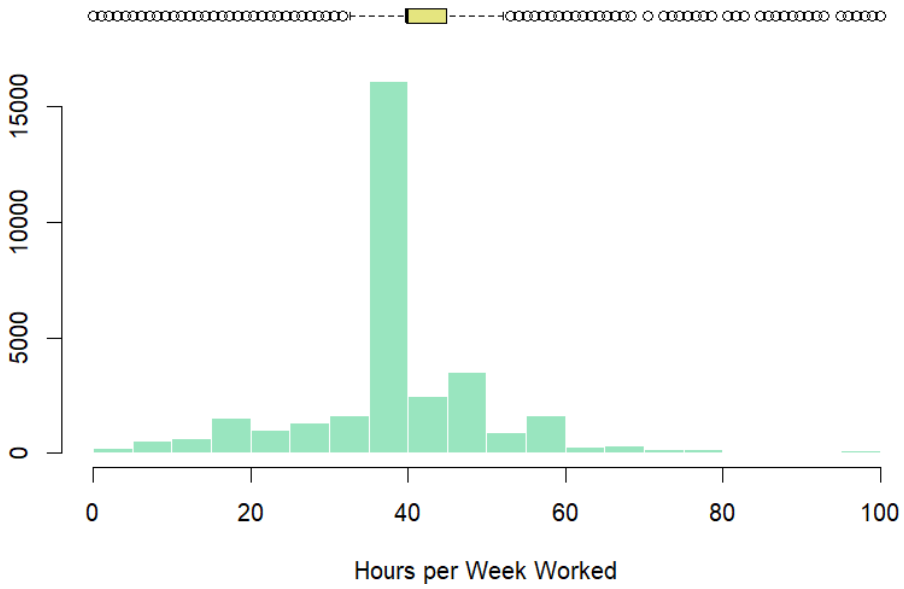

Figure 3: Histogram and Boxplot

When looking at the graphic, we can check that the vast majority of the adults work 35-40 hours a week. But, because we are working with all of the population, we can verify that there are some sectors can work all the way from 20 hours (students, part time jobs) to 60 hours a week (executives, doctors). Of course, sometimes there are outliers and exceptions.

2.1.2 Cental Tendency Measures

If we look at the graphic we can get a rough idea of the data, but sometimes we need more concrete numbers. Now, we'll calculate the mean and median:

```
meanhours <- mean(adult$hours.per.week)
medianhours <- median(adult$hours.per.week)
```

And we get that, on average, adults work 40,437 hours per week. We also get that the value in the middle, that separates data equally, is 40 hours per week. This evaluations corresponds to the one we just made just by looking at the graphic.

2.1.3 Max, min, quartiles

Now, we're interested in the extreme and intermediate values. The simple command in *R* that gives us all the data we need is the *summary()* function.

```
summary(adult$hours.per.week)
```

```
   Min. 1st Qu.  Median  Mean   3rd Qu.    Max.
  1.00   40.00   40.00   40.44   45.00   99.00
```

Just by looking at this result, we know that there is some lucky minority that just works 1 hour a week, and there is some hard-working minority that does 99 hours a week. But, in general, the 1st and the 3rd quartiles confirm what we've been discussing: The majority of the workers do $\sim$ 40 hours a week.

2.1.4 Dispersion

Therefore, and just in case, we can calculate some measures of dispersion to check if our data is dispersed or not.

```
range <- max(adult$hours.per.week)-min(adult$hours.per.week)
irqrange <- IQR(adult$hours.per.week)
```

From this commands we can conclude that that although our range (98) is very big, the interquartile range is only of 5, so the data is very concentrated, but with some outliers.

2.1.5 Chebyshev's inequality

To conclude this variable's analysis, we'll verify if the Chebyshev's inequality applies to the *hours per week* data point. This rule manifests that *at least* $1 - \frac{1}{2^k}\%$ of the data must be contained in the $(\overline{x} - ks, \overline{x} + ks)$ interval.

First, we'll calculate the standard deviation and we'll select a k:

```
sdhours <- sd(adult$hours.per.week)
k <- 2
```

So, with $k = 2$, in theory, *at least* 75% of the data must be contained in the mentioned interval. Let's see if that is the case:

```
minbound <- (meanhours-k*sdhours)
maxbound <- (meanhours+k*sdhours)

obsininterval <- numdatapoints
obsininterval <- obsininterval -
↪  length(adult$hours.per.week[maxbound<adult$hours.per.week])
obsininterval <- obsininterval -
↪  length(adult$hours.per.week[minbound>adult$hours.per.week])

percentageofdatainterval <- obsininterval*100/numdatapoints
```

First, we calculate the upper and lower bounds. Then, we start with the total number of observations in our dataset. After that, we remove the ones that are above the upper bound, and the ones that are below the lower bound. As a result of those calculations, we have in the *obsininterval* variable the number of observations in the $(\overline{x} - ks, \overline{x} + ks)$ interval. Then, we calculate the percentage with a simple rule of 3. So, when we check the value of our resulting variable, we get the output:

```
percentageofdatainterval
```

```
[1] 93.23424
```

And that verifies the result when $k = 2$, because $93,23\% > 75\%$.

2.2 Relationship with other variables

In this section, we'll analyze if our chosen variable (hours per week worked) has any relation to the other (numeric) variables in the *adult* dataset.

2.2.1 Data Visualization

To visualize this quickly, we'll use a *correlogram matrix*. This is better drawn using the *GGally* R package.

```
# Package installation
install.packages("GGally")
library(GGally)
# Plot
ggcorr(adult, method = c("everything", "pearson"))
```

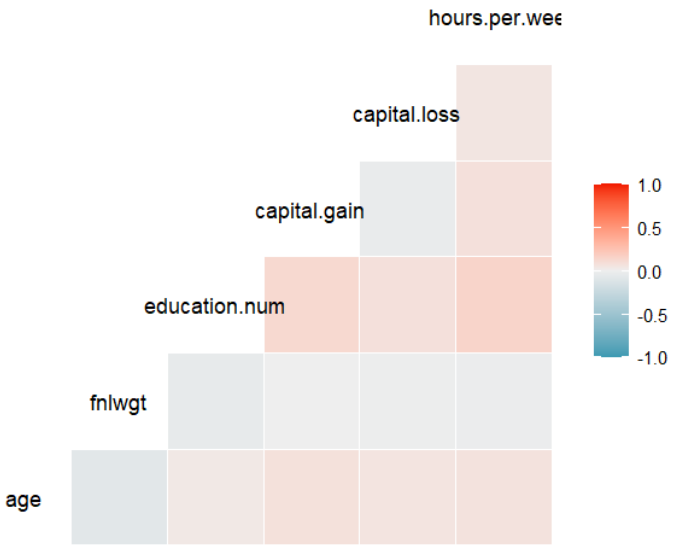

Figure 4: Correlation Matrix

The criteria to check if two variables have a higher correlation is that they have a more *redish* color between them.

Looking at the graphic, we can see that hours per week worked has some relation to the capital gained, and has also relation with the education level. We can appreciate that it has some relation also to the age. The final weight has no relation with hours per week whatsoever.

2.2.2 Correlations

Now, let's calculate the correlation coefficient, to see if our graph conclusions align themselves with the math.

```
cor(adult$education.num, adult$hours.per.week)
cor(adult$age, adult$hours.per.week)
cor(adult$capital.gain, adult$hours.per.week)
```

On the one hand, as for affirming that an adult works more if it has more education, we can not strongly guarantee it, because it has a slight correlation (0.15), but as far as the other variables are concerned, this is the stronger relationship.

On the other hand, to say that an adults works more if he/she has more age and/or gains more is an overestimation. It's true the relationship is there, but it's very small (0.06).

2.2.3 Regression Model

Now, we'll choose the *Hours Per Week* variable and the *Education (level)* variable. We'll plot them, draw the regression line and analyze the results:

```
plot(adult$hours.per.week, adult$education.num, col="#69b3a2",
↪  xlab="Hours Per Week", ylab="Education Level")
abline(lm(adult$education.num ~ adult$hours.per.week))
```

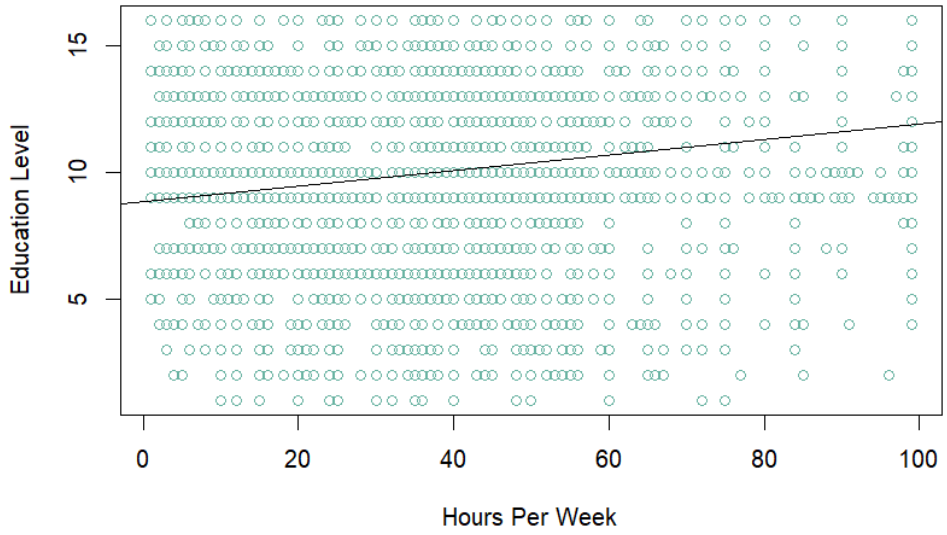

Figure 5: Scatterplot with Regression Line

As we commented previously, data is very dispersed, but there is some *slight* relation between both variables. The regression model drawn by the *lm()* function gives us a very rough approximation:

```
lm(adult$education.num ~ adult$hours.per.week)

Coefficients:
         (Intercept)        adult$hours.per.week
             8.83266                     0.03086
```

With the $y = 0.03x + 8.83$ model we could guess, for example, that a person working 150 hours a week would have an education level of 13.3.

2.3 Context Analysis (Inference)

2.3.1 Confidence Interval

Now, we'll calculate a 95% confidence interval for $\overline{x}$, given the variable *age* in the *adult* dataset. We'll be interested in knowing the average age of our workers.

The confidence interval used in this case will be:

$$\left(\overline{x} - z_{\alpha/2} \cdot \frac{\sigma}{\sqrt{n}}, \overline{x} + z_{\alpha/2} \cdot \frac{\sigma}{\sqrt{n}}\right)$$

Where $\overline{x}$ is characteristic to the sample selected, $n = 100$ is the size of the sample, σ is the standard deviation of the whole datapoint, and $\alpha = (1 - 0.95) = 0.05$

First, we'll simulate a survey of size n and select the sample.

```
n = 100
ci <- 0.95
survey <- sample(adult$age, n)
```

Then, we'll calculate the (sample) mean and the (datapoint) standard deviation.

```
survmean <- mean(survey)
setsd <- sd(adult$age)
```

And finally, we'll calculate the z-score and we'll perform all the calculations

```
survz <- qnorm(p=(1-ci)/2, mean=survmean, sd=setsd)
survlow <- survmean - survz*setsd/sqrt(n)
survhigh <- survmean + survz*setsd/sqrt(n)
```

$$\left(37.87 - 11,13 \cdot \frac{13.64}{10}, 37.87 + 11,13 \cdot \frac{13.64}{10}\right)$$

Now we can say with a 95% confidence that the mean *age* of the workers (μ) will be in the $(22.68, 53.05)$ interval. Just to check, we can assure it:

```
muage <- mean(adult$age)
muage
```

```
[1] 38.58165
```

2.3.2 Hypothesis Testing

Lastly, we're going to pose an hypothesis and then test its the validity. Just like in the confidence intervals, we're going to use $\overline{x}$ as our testing variable. We are going to use a significance level of $\sigma = 0.05$ and our null hypothesis will be:

$$H_0 = \text{The mean age of working adults is equal to 44.}$$

And our alternative hypothesis:

$$H_1 = \text{The mean age of working adults is NOT equal to 44.}$$

Also, we know that because we took a sample of $n = 100$ elements, $\overline{x}$ follows a normal distribution. So, we'll calculate the *p-value* to check if our hypothesis is true:

$$P\left(\frac{|\overline{x} - \mu|}{\sigma/n} > z|\mu = 44\right)$$

```
alpha <- 0.05
estmean <- 44

n = 100
survey <- sample(adult$age, n)
survmean <- mean(survey)
setsd <- sd(adult$age)
# two sided test
pV <- 2*pnorm(survmean,estmean,(setsd/sqrt(n)))
pV
# [1] 0.00038
```

And because our p-value is smaller than our significance level, $(0,00038 < 0,05)$, we reject our initial assumption: the null hypothesis (H_0).

3 Appendix

3.1 Complete R Source Code

(Online Pastebin Link)

```
#
# NICOLAS AGUADO, UPV/EHU, 30 DECEMBER 2022
# STATISTICAL METHODS FOR ENGINEERS
# naguado008@ikasle.ehu.eus
#
#-------------------------------------------------
#
# 1ST SCENARIO
#
#-------------------------------------------------
moneya <- c(1, 2)
gamea <- c(0.5,0.5)
barplot(gamea, names.arg=moneya, xlab="Money Won (USD$)", ylab="
   Probability", ylim=c(0,1))

moneyb <- c(25, 0, -15)
gameb <- c(1/6, 4/6,1/6)
barplot(gameb, names.arg=moneyb, xlab="Resulting Money (USD$)",
   ylab="Probability", ylim=c(0,1))

expecteda <- weighted.mean(moneya, gamea)
vara <- var(moneya)

expectedb <- weighted.mean(moneyb, gameb)
varb <- var(moneyb)

n <-10000
mean(sample(moneya, prob=gamea, size=n, replace=TRUE))
mean(sample(moneyb, prob=gameb, size=n, replace=TRUE))

mn <- 10000*expecteda
std <- sqrt(10000*vara)
proba <- 1-pnorm(5000, mean=mn, sd=std)
proba
mn <- 10000*expectedb
std <- sqrt(10000*varb)
probb <- 1-pnorm(5000, mean=mn, sd=std)
probb
```

```
#-------------------------------------------------
#
# 2ND SCENARIO
#
#-------------------------------------------------
adult <- read.table("adult.csv", col.names=c("age", "workclass", "
    fnlwgt", "education", "education-num", "marital-status", "
    occupation", "relationship", "race", "sex", "capital-gain", "
    capital-loss", "hours-per-week", "native-country", "50K-
    Prediction"), sep=",", header = FALSE)

numdatapoints <- length(adult$hours.per.week)

# Layout to split the screen
layout(mat = matrix(c(1,2),2,1, byrow=TRUE), height = c(1,8))
# Draw the boxplot and the histogram
par(mar=c(0, 3.1, 1.1, 2.1))
boxplot(adult$hours.per.week , horizontal=TRUE , xaxt="n" , col=rgb
    (0.8,0.8,0,0.5) , frame=F)
par(mar=c(4, 3.1, 1.1, 2.1))
hist(adult$hours.per.week , breaks=25 , col=rgb(0.2,0.8,0.5,0.5) ,
    border=F , main="" , xlab="Hours per Week Worked", ylab="
    Individuals")

meanhours <- mean(adult$hours.per.week)
medianhours <- median(adult$hours.per.week)

summary(adult$hours.per.week)

range <- max(adult$hours.per.week)-min(adult$hours.per.week)
irqrange <- IQR(adult$hours.per.week)

sdhours <- sd(adult$hours.per.week)
k <- 2

minbound <- (meanhours-k*sdhours)
maxbound <- (meanhours+k*sdhours)

obsininterval <- numdatapoints
obsininterval <- obsininterval - length(adult$hours.per.week[
    maxbound<adult$hours.per.week])
obsininterval <- obsininterval - length(adult$hours.per.week[
```

```
   minbound>adult$hours.per.week])
percentageofdatainterval <- obsininterval*100/numdatapoints

# Package installation
install.packages("GGally")
library(GGally)
# Plot
ggcorr(adult, method = c("everything", "pearson"))

cor(adult$education.num, adult$hours.per.week)
cor(adult$age, adult$hours.per.week)
cor(adult$capital.gain, adult$hours.per.week)

plot(adult$hours.per.week, adult$education.num, col="#69b3a2", xlab
   ="Hours Per Week", ylab="Education Level")
abline(lm(adult$education.num ~ adult$hours.per.week))

lm(adult$education.num ~ adult$hours.per.week)

n = 100
ci <- 0.95
survey <- sample(adult$age, n)
survmean <- mean(survey)
setsd <- sd(adult$age)
survz <- qnorm(p=(1-ci)/2, mean=survmean, sd=setsd)
survlow <- survmean - survz*setsd/sqrt(n)
survhigh <- survmean + survz*setsd/sqrt(n)
survlow
survhigh
muage <- mean(adult$age)
muage

alpha <- 0.05
estmean <- 44
n = 100
survey <- sample(adult$age, n)
survmean <- mean(survey)
setsd <- sd(adult$age)
#two sided test
pV <- 2*pnorm(survmean,estmean,(setsd/sqrt(n)))
pV
```

3.2 Bibliography and References

1. Slides present in https://egela.ehu.eus
2. Solved exercises present in https://egela.ehu.eus
3. R Programming Language Documentation
4. R Graph Gallery: https://r-graph-gallery.com/
5. Online *paste-site* for the code : https://pastebin.com
6. https://tex.stackexchange.com/questions/107470/getting-section-numbering-to-start-at-0
7. https://latex-tutorial.com/tutorials/lists/
8. https://tex.stackexchange.com/questions/347007/language-and-spell-check
9. https://tex.stackexchange.com/questions/32140/how-to-write-a-function-piecewise-with-bracket-outside
10. https://www.geeksforgeeks.org/r-bar-charts/
11. https://statisticsglobe.com/increase-y-axis-scale-of-barplot-in-r
12. https://www.statology.org/expected-value-in-r/
13. https://www.programmingr.com/statistics/variance/
14. https://r-lang.com/how-to-calculate-variance-in-r-using-var-function/
15. https://math-linux.com/latex-26/faq/latex-faq/article/latex-horizontal-space-qquad-hspace-thinspace-enspace
16. https://iqcode.com/code/other/periodic-number-latex
17. https://latex-tutorial.com/tables-in-latex/
18. *Adult* Dataset: https://archive.ics.uci.edu/ml/datasets/adult
19. https://r-graph-gallery.com/82-boxplot-on-top-of-histogram.html
20. https://tex.stackexchange.com/questions/9363/how-does-one-insert-a-backslash-or-a-tilde-into-latex
21. https://www.physicsread.com/latex-sigma-symbol/

22. https://www.physicsread.com/latex-mu-and-nu-symbol/

23. https://www.tutorialspoint.com/r/r_normal_distribution.htm

24. https://latexhelp.com/latex-underscore/

25. https://www.geeksforgeeks.org/how-to-calculate-quartiles-in-r/

26. https://en.wikipedia.org/wiki/Chebyshev's_inequality

27. https://latex-tutorial.com/tutorials/hyperlinks/

28. https://tex.stackexchange.com/questions/14342/verbatim-environment-that-can-break-long-lines

29. https://tex.stackexchange.com/questions/85200/include-data-from-a-txt-verbatim/85218

30. https://www.rdocumentation.org/packages/mpoly/versions/1.1.1/topics/chebyshev

31. https://www.overleaf.com/learn/latex/Page_numbering

32. https://stackoverflow.com/questions/37893216/how-to-select-specific-rows-in-a-data-set-r

33. https://www.statology.org/confidence-interval-in-r/

34. https://www.geeksforgeeks.org/normal-distribution-in-r/

35. https://r-lang.com/pnorm-function-in-r-with-example/

36. https://www.geeksforgeeks.org/hypothesis-testing-in-r-programming/