G4.P-1

David Emanuel Craciusecu y Laura Pérez Medeiro

October 11, 2019

The test laboratory will be divided into two parts:

1. An exercise mentored by the teacher, in which all the contents explained in the fisrt unit will be revised. The files used in this exercise are:

1.1 ”Satelites.txt”, which contains information about the different ratious of Uranus’ moons. The aim of the exercise is obtained the same result as we get into the theorical class.

```
> while (!"satelites.txt" %in% list.files(getwd()))
+ {
+   print("Data file not found. Add \"satelites.txt\" to the current directory.")
+   invisible(readline(prompt="Press [enter] to continue"))
+ }
>         satelites <- read.table("satelites.txt")
>         satelites

           nombre radio
1        CORDELIA    13
2          OFELIA    16
3          BIANCA    22
4         CRESIDA    33
5       LESDEMONA    39
6         JULIETA    42
7       ROSALINDA    27
8         BELINDA    34
9  LUNA-1986U1020    20
10       CALIBANO    30
11       LUNA-119    20
12     LUNA_119U2    15

>         radius <- satelites $radio
>         radius

 [1] 13 16 22 33 39 42 27 34 20 30 20 15
```

It is important to take into account that the working directory must be the same to the file’s directory when read.table is going to be used. Otherwise, we must indicate the absolute route where the file is.

After reading the data, different analyzes will be applied to those Uranus’ moon radius:

a) Calculate absolute and relative satellite radius frequencies: ABSOLUTE FRECUENCY:

```
> absoluteFreq    <- function(set) {table(set)}
> absoluteFreq(radius)
```

```
set
13 15 16 20 22 27 30 33 34 39 42
 1  1  1  2  1  1  1  1  1  1  1
```

ACUMULATIVE ABSOLUTE FRECUENCY:

```
> cumAbsoluteFreq <- function(set) {cumsum(absoluteFreq(set))}
> cumAbsoluteFreq(radius)
```

```
13 15 16 20 22 27 30 33 34 39 42
 1  2  3  5  6  7  8  9 10 11 12
```

RELATIVE FRECUENCY

```
> relativeFreq <- function(set) {table(set) / length(set)}
```

ACUMULATIVE RELATIVE FRECUENCY

```
> cumRelativeFreq <- function(set) {cumsum(relativeFreq(set))}
```

b) Arithmetic mean

```
> arithmeticMean <- function(set, usrTrim = 0) (mean(set, trim = usrTrim))
> arithmeticMean(radius)
```

```
[1] 25.91667
```

c) Measures of dispersion, where the following page was used as a reference for this section: http://iridl.ldeo.columbia.edu/dochelp/StatTutorial/Dispersion/index.htmlIntro

- RANGE:

```
> range <- function(set) {max(set) - min(set)}
> range(radius)
```

```
[1] 29
```

- STANDARD DEVIATION

```
> stdDeviation <- function(set)
+ {
+    sd(set) * (sqrt((length(set) - 1) / length(set)))
+ }
> stdDeviation(radius)
```

```
[1] 9.277736
```

- VARIANCE:

```
> variance <- function(set) {var(set) * (length(set) - 1 / length(set))}
> variance(radius)
```

```
[1] 1118.993
```

- ROOT MEAN SQUARE:

```
> rootMeanSqr <- function(set) {sqrt(mean(set ^ 2))}
> rootMeanSqr(radius)
```

```
[1] 27.52726
```

- ROOT MEAN SQUARE ANOMALY:

```
> rootMeanSqrAn <- function(set) {sqrt(sum(set - mean(set)) ^ 2) / length(set)}
> rootMeanSqrAn(radius)
```

```
[1] 1.184238e-15
```

- INTERQUARTILE RANGE:

```
> interQuartRange <- function(set) {IQR(set)}
> interQuartRange(radius)
```

```
[1] 14.25
```

- MEDIAN ABSOLUTE DEVIATION

```
> medAbsDeviation <- function(set) {mad(set)}
> medAbsDeviation(radius)
```

```
[1] 12.6021
```

d) Finally, measures of order:
-MEDIAN:

```
> getMedian    <- function(set) {median(set)}
> getMedian(radius)
```

```
[1] 24.5
```

-MODE:

```
> getMode      <- function(set) {mfv(set)}
>
```

-QUARTILES:

```
> getQuartiles <- function(set) {quantile(set)}
> getQuartiles(radius)
```

```
   0%   25%   50%   75%  100%
13.00 19.00 24.50 33.25 42.00
```

-54th QUANTILE:

```
> getQuantiles <- function(set, range = 0) {quantile(set, probs = range)}
> getQuantiles(radius)
```

```
0%
13
```

1.2 Now, the file will be ”cardata.sav” where the same analyzes of data will be applied to mpg. But this time, it is not necessary to calculate the 54th and the frequencies.

It is necessary to import the foreign’s library to read fata from SPSS files in R.

```
> library(foreign)
```

Then, the file is read, only data related to mpg is load into the variable mpg and we use a filter to delete all unusefull data and group then.

```
> dataset = read.spss("cardata.sav", to.data.frame=TRUE)
> mpg = dataset$mpg
> mpg = mpg[!is.na(mpg)]
> mpg
```

```
  [1] 36.1 19.9 19.4 20.2 19.2 20.5 20.2 25.1 20.5 19.4 20.6 20.8 18.6 18.1 19.2 17.7 18.1
 [20] 30.9 23.2 23.8 21.5 19.8 22.3 20.2 20.6 17.0 17.6 16.5 18.2 16.9 15.5 19.2 18.5 35.7
 [39] 23.9 34.2 34.5 28.4 28.8 26.8 33.5 32.1 28.0 26.4 24.3 19.1 27.9 23.6 27.2 26.6 25.8
 [58] 39.0 34.7 34.4 29.9 22.4 26.6 20.2 17.6 28.0 27.0 34.0 31.0 29.0 27.0 24.0 23.0 38.0
 [77] 38.0 26.0 22.0 36.0 27.0 27.0 32.0 28.0 31.0 43.1 20.3 17.0 21.6 16.2 31.5 31.9 25.4
 [96] 41.5 34.3 44.3 43.4 36.4 30.4 40.9 29.8 35.0 33.0 34.5 28.1 30.7 36.0 44.0 32.8 39.4
[115] 27.2 21.1 23.9 29.5 34.1 31.8 38.1 37.2 29.8 31.3 37.0 32.2 46.6 40.8 44.6 33.8 32.7
[134] 39.1 35.1 32.3 37.0 37.7 34.1 33.7 32.4 32.9 31.6 25.4 24.2 37.0 31.0 36.0 36.0 34.0
[153] 38.0 32.0
```

Now, data is ready to be analyzes so we can use the same functions as it has been used in the previous section.