First Delivery Statistics

Eduardo Alarcón & Alfonso Pineda

2022-11-13

# 1.Introduction:

The first delivery of the final project of Statistics of the degree of Computer Science. In this document, we (Alfonso Pineda and Eduardo Alarcón) will be showing the histogram of the main variable we have chosen for our project, is the of a song, which is a value assigned by the spoify algorith to try and categorize songs and if they make people more or less energetic. As well, we will have a Box Plot and the statistical Measures on the same block as the histogram, the first one.

On the second block, we will show the of a song which we have tested to be the variable that has the most relation with the main variable. We are also going to show a Histogram, a Box Plot and the Statistical Measurements.

Lastly, we will show the Scatter Plot and the Linear Model between the energy and the loudness

$\newpage$

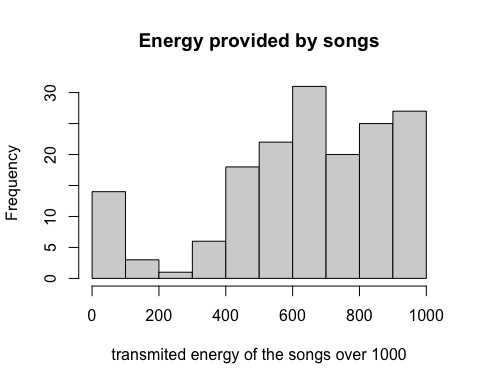
# 1st Block:

Including the data from the excel: The first thing we need to do is import the data we are going to work with.

library(readxl)  
SpotifySongs <- read\_excel("songstats.xlsx")  
# View(SpotifySongs)

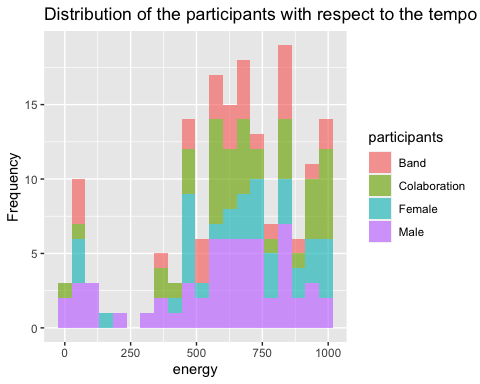
### Histogram Then, we need to create the histogram, using R

energy <- SpotifySongs$energy  
hist(energy, xlab = "transmited energy of the songs over 1000",  
 main = "Energy provided by songs")



We have also created the histogram differentiating if the artist is a Male, Female, Band or Collaboration between different artists (we used different colors to view them):

# {r, fig.height = 4, fig.width = 6}  
  
suppressWarnings(library(ggplot2))  
SpotifySongs$participants <- "Male"  
SpotifySongs$participants[SpotifySongs$GenderGroup == "F"] <- "Female"  
SpotifySongs$participants[SpotifySongs$GenderGroup == "Band"] <- "Band"  
SpotifySongs$participants[SpotifySongs$GenderGroup == "Colab"] <- "Colaboration"  
  
qplot(energy, data=SpotifySongs, geom=c("histogram"), fill=participants,   
 alpha=I(.65), main="Distribution of the participants with respect to the tempo",   
 xlab="energy", ylab="Frequency", bins=20)

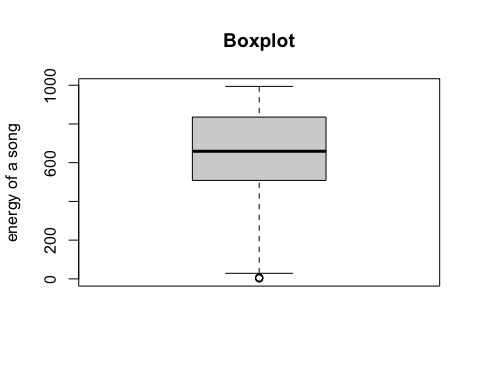


#Use bins=num to set the number of intervals

## Boxplot

Then, we also need to create a Box Plot:

#{r, fig.height = 4, fig.width = 5}  
boxplot(energy, ylab = "energy of a song",   
 main = "Boxplot")



As we can see from the Box Plot there are some extreme outliers.

We can see from the Box Plot that the Histogram is not symmetric at all.

# Statistical Measures

Now, it’s time for us to calculate the statistical measures of the main variable, These measures are: the , the , the , the , the , the , and the .

First, we need to store the variable as Data in R, then, we ask R to describe the variable, which outputs the measures we need, as well as the number of elements there are, in this case N: 167

energy<-SpotifySongs$energy  
suppressWarnings(library(summarytools))  
# Describe the variable energy  
descr(energy)

## Descriptive Statistics   
## energy   
## N: 167   
##   
## energy  
## ----------------- --------  
## Mean 633.62  
## Std.Dev 259.35  
## Min 2.40  
## Q1 500.00  
## Median 659.00  
## Q3 836.00  
## Max 994.00  
## MAD 250.56  
## IQR 327.00  
## CV 0.41  
## Skewness -0.81  
## SE.Skewness 0.19  
## Kurtosis 0.12  
## N.Valid 167.00  
## Pct.Valid 100.00

# 2nd Block:

On this second part we will test which of the variables we have on our study has the best correlation with the main variable. To asses this, we will use the next block of R:

SpotifySongs <- read\_excel("songstats.xlsx")  
dance <- SpotifySongs$danceability  
energy <- SpotifySongs$energy  
loud <- SpotifySongs$loudness  
speech <- SpotifySongs$speechiness  
accous <- SpotifySongs$acousticness  
live <- SpotifySongs$liveness  
valence <- SpotifySongs$valence  
tempo <- SpotifySongs$tempo  
duration <- SpotifySongs$duration\_ms  
# Choose best second variable  
cat("Correlation between loud and Danceability\n")

## Correlation between loud and Danceability

cor(loud, SpotifySongs$danceability)

## [1] -0.4811005

cat("Correlation between loud and energy\n")

## Correlation between loud and energy

cor(loud, SpotifySongs$energy)

## [1] -0.8125021

cat("Correlation between loud and Loudness\n")

## Correlation between loud and Loudness

cor(loud, SpotifySongs$loudness)

## [1] 1

cat("Correlation between loud and speechiness\n")

## Correlation between loud and speechiness

cor(loud, SpotifySongs$speechiness)

## [1] -0.2661057

cat("Correlation between loud and acousticness\n")

## Correlation between loud and acousticness

cor(loud, SpotifySongs$acousticness)

## [1] 0.66467

cat("Correlation between loud and liveness\n")

## Correlation between loud and liveness

cor(loud, SpotifySongs$liveness)

## [1] -0.07253627

cat("Correlation between loud and valence\n")

## Correlation between loud and valence

cor(loud, SpotifySongs$valence)

## [1] -0.3773111

cat("Correlation between loud and tempo\n")

## Correlation between loud and tempo

cor(loud, tempo)

## [1] -0.335712

cat("Correlation between loud and duration\_ms\n")

## Correlation between loud and duration\_ms

cor(loud, SpotifySongs$duration\_ms)

## [1] 0.2083979

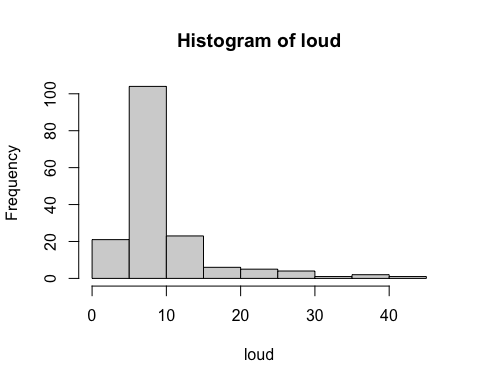
#The best correlation found is: Energy & Loudness, with a correlation of 0.8125021

With the previous results, we choose the variable loudness. These are the statistical variables of the loudness, as well as the Histogram and Box Plot:

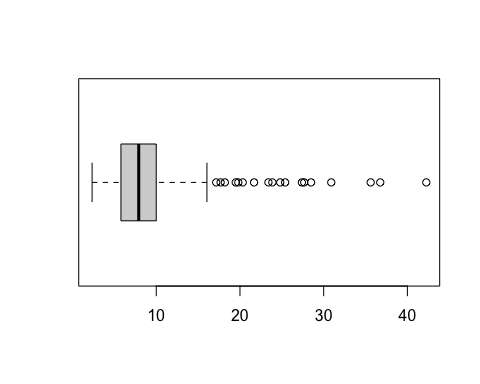
SpotifySongs <- read\_excel("songstats.xlsx")  
loud<-SpotifySongs$loudness  
descr(loud)

## Descriptive Statistics   
## loud   
## N: 167   
##   
## loud  
## ----------------- --------  
## Mean 9.53  
## Std.Dev 6.55  
## Min 2.34  
## Q1 5.78  
## Median 7.91  
## Q3 10.03  
## Max 42.26  
## MAD 3.15  
## IQR 4.22  
## CV 0.69  
## Skewness 2.46  
## SE.Skewness 0.19  
## Kurtosis 6.83  
## N.Valid 167.00  
## Pct.Valid 100.00

# Histogram/Box-Plot of the secondary variable  
hist(loud)



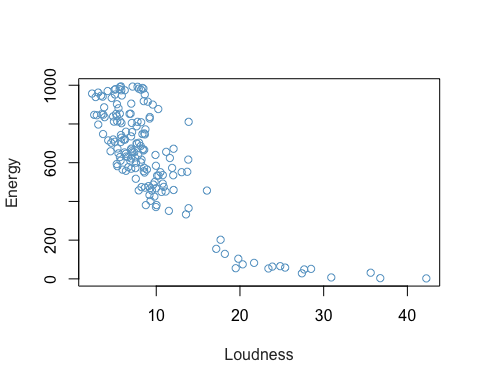
boxplot(loud, horizontal = TRUE)



# 3rd Block:

On the last block, we will see the Scatter Plot and Linear Model between the main variable, the tempo and the speechiness

# Scatter plot without linear model of tempo and speechiness  
plot(  
 loud,  
 energy,   
 xlab = "Loudness",  
 ylab = "Energy",  
 col.lab = "gray19",  
 col="skyblue3"  
)



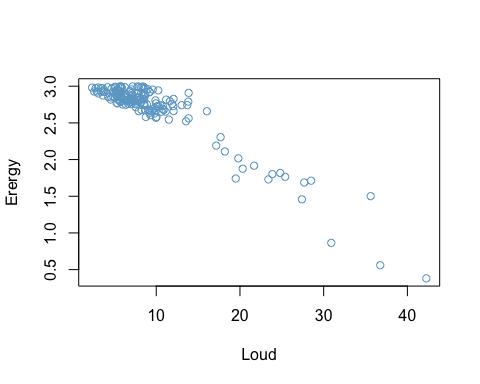
## Scatter Plot with the Linear Model:

The scatter plot created with the speechiness is:

#energy <- log(energy)  
energy <- log10(SpotifySongs$energy)  
loud <- SpotifySongs$loudness  
  
RegressionModel <- lm(loud ~ energy, data=SpotifySongs)  
print(RegressionModel)

##   
## Call:  
## lm(formula = loud ~ energy, data = SpotifySongs)  
##   
## Coefficients:  
## (Intercept) energy   
## 22.52456 -0.02051

plot(  
 loud,  
 energy,  
 xlab = "Loud",  
 ylab = "Erergy",  
 col="skyblue3"  
)  
abline(RegressionModel, col="tomato3")

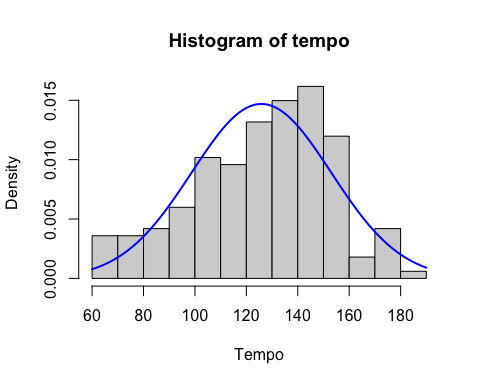


# Model Fitting.

SpotifySongs <- read\_excel("songstats.xlsx")  
suppressWarnings(library(summarytools))   
energy <- SpotifySongs$energy  
descr(tempo)

## Descriptive Statistics   
## tempo   
## N: 167   
##   
## tempo  
## ----------------- --------  
## Mean 125.87  
## Std.Dev 27.16  
## Min 60.17  
## Q1 108.67  
## Median 129.47  
## Q3 145.01  
## Max 180.04  
## MAD 27.60  
## IQR 36.06  
## CV 0.22  
## Skewness -0.43  
## SE.Skewness 0.19  
## Kurtosis -0.41  
## N.Valid 167.00  
## Pct.Valid 100.00

hist(tempo, probability = TRUE, xlab = "Tempo")  
curve(dnorm(x, mean(tempo), sd(tempo)), col="blue", lwd=2, add=TRUE, yaxt="n")



Partition <- hist(tempo, plot=FALSE)  
Partition

## $breaks  
## [1] 60 70 80 90 100 110 120 130 140 150 160 170 180 190  
##   
## $counts  
## [1] 6 6 7 10 17 16 22 25 27 20 3 7 1  
##   
## $density  
## [1] 0.0035928144 0.0035928144 0.0041916168 0.0059880240 0.0101796407  
## [6] 0.0095808383 0.0131736527 0.0149700599 0.0161676647 0.0119760479  
## [11] 0.0017964072 0.0041916168 0.0005988024  
##   
## $mids  
## [1] 65 75 85 95 105 115 125 135 145 155 165 175 185  
##   
## $xname  
## [1] "tempo"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"

library(fitdistrplus)

## Loading required package: MASS

## Loading required package: survival

normalfit <- fitdist(tempo, "norm")