HOMEWORK 1

Elena Gómez Espinosa

UC3M, 2021

```
library(ggplot2)
```

Data preprocessing

Exploring the data

In this project I am going to used a database about call of duty players skills. Call of Duty is a first-person shooter video game. I am aware that the database is not too large, but I started the project with a bigger one and I had plenty of problems because my laptop did not have power enough to execute some models, so I decided to use a smaller one in order to do the hole project without any problem.

```
setwd("C:/Users/elepu/OneDrive/Escritorio/1º CUATRI/APRENDIZAJE ESTADÍSTICO/project 1")
data = read.csv(file = "cod.csv",
                header = T, sep = ",", dec = ".")
```

As I have said, the variables are related with call of duty players skills.Here they are explained:

name: the name for each player wins : number of times the player win a match kills : number of kills the player made in all his matches kdRatio : kill/deaths ratio that means, if a player has 10 kills and 5 deaths, his KD ratio is equal to 2. A KD ratio of 1 means that the player got killed exactly as many times as he successfully eliminated his opponents killstreak : kill a number of enemy players without dying. level : is the player grade losses : total number of losing prestige: it is an optional Mode that players can choose after they progress to Level 55 and max hits : number of times the player damaged another player timePlayed : the time spent by every player playing Call of Duty in hours headshots : number of times the player hit the others with headshots averageTime : average time gamesPlayed : number of times the player play multiplayer match assists : number of times player damaging an enemy but a teammate gets the kill. misses : the number of times the player miss the hit xp : Experience Points (XP) are a numerical quantity exclusive to multiplayer that dictates a player’s level and progress in the game. scorePerMinute :a measure of how many points players are gaining per unit time. shots :total number of shots the player did deaths : number of time the player gots killed in the game. The data base is from https://www.kaggle.com/aishahakami/call-of-duty-players (https://www.kaggle.com/aishahakami/call-of-duty-players)

Firstly, we are going to take a look to become familiar with the data base

```
str(data)
```

```
## 'data.frame':    1558 obs. of  19 variables:
##  $ name          : chr  "RggRt45#4697369" "JohniceRex#9176033" "bootybootykill#1892064" "J
NaCo#5244172" ...
##  $ wins          : int  0 0 0 3 0 684 4 186 741 26 ...
##  $ kills         : int  0 0 66 2 2 27011 162 1898 21803 349 ...
##  $ kdRatio       : num  0 0 1.03 0.4 0.2 ...
##  $ killstreak    : int  0 0 0 0 0 18 4 13 26 7 ...
##  $ level         : int  1 1 9 1 1 177 6 37 185 12 ...
##  $ losses        : int  0 0 0 0 0 10 2 7 29 4 ...
##  $ prestige      : int  0 110 110 0 110 110 0 2 111 0 ...
##  $ hits          : int  0 0 0 0 0 98332 568 5111 81361 996 ...
##  $ timePlayed    : int  0 7 32 3 5 1366 8 550 2442 44 ...
##  $ headshots     : int  0 0 16 0 1 5113 35 485 3894 40 ...
##  $ averageTime   : num  0 7 32 3 5 ...
##  $ gamesPlayed   : int  0 0 0 0 0 588 4 150 864 15 ...
##  $ assists       : int  0 0 1 0 0 6063 68 488 4029 138 ...
##  $ misses        : int  0 0 0 0 0 305319 4836 39978 327230 4844 ...
##  $ xp            : int  0 700 48300 1150 1000 3932335 24485 458269 4269370 72765 ...
##  $ scorePerMinute: num  0 0 0 0 0 ...
##  $ shots         : int  0 0 0 0 0 403651 5404 45089 408591 5840 ...
##  $ deaths        : int  0 16 64 5 10 25321 256 3332 21032 786 ...
```

```
head(data)
```

```
##                     name wins kills  kdRatio killstreak level losses prestige
## 1        RggRt45#4697369    0     0 0.000000          0     1      0        0
## 2     JohniceRex#9176033    0     0 0.000000          0     1      0      110
## 3 bootybootykill#1892064    0    66 1.031250          0     9      0      110
## 4          JNaCo#5244172    3     2 0.400000          0     1      0        0
## 5  gomezyayo_007#6596687    0     2 0.200000          0     1      0      110
## 6   Brxndoon7-LK#4002715  684 27011 1.066743         18   177     10      110
##    hits timePlayed headshots averageTime gamesPlayed assists misses      xp
## 1     0          0         0    0.000000           0       0      0       0
## 2     0          7         0    7.000000           0       0      0     700
## 3     0         32        16   32.000000           0       1      0   48300
## 4     0          3         0    3.000000           0       0      0    1150
## 5     0          5         1    5.000000           0       0      0    1000
## 6 98332       1366      5113    2.323129         588    6063 305319 3932335
##   scorePerMinute  shots deaths
## 1          0.000      0      0
## 2          0.000      0     16
## 3          0.000      0     64
## 4          0.000      0      5
## 5          0.000      0     10
## 6        255.672 403651  25321
```

```
summary(data)
```

```
##      name                wins          kills            kdRatio
##  Length:1558        Min.   :   0   Min.   :    0.0   Min.   :0.0000
##  Class :character   1st Qu.:   0   1st Qu.:    4.0   1st Qu.:0.2614
##  Mode  :character   Median :  10   Median :  191.5   Median :0.7328
##                     Mean   : 153   Mean   : 3753.0   Mean   :0.6371
##                     3rd Qu.: 168   3rd Qu.: 3445.8   3rd Qu.:0.9553
##                     Max.   :3519   Max.   :66935.0   Max.   :3.0000
##    killstreak          level            losses          prestige
##  Min.   :  0.000   Min.   :  1.00   Min.   : 0.000   Min.   :  0.00
##  1st Qu.:  0.000   1st Qu.:  1.00   1st Qu.: 0.000   1st Qu.:  0.00
##  Median :  5.000   Median : 11.00   Median : 2.000   Median : 14.00
##  Mean   :  6.895   Mean   : 44.41   Mean   : 4.998   Mean   : 47.66
##  3rd Qu.: 12.000   3rd Qu.: 51.00   3rd Qu.: 8.000   3rd Qu.:110.00
##  Max.   :235.000   Max.   :435.00   Max.   :80.000   Max.   :117.00
##       hits            timePlayed       headshots        averageTime
##  Min.   :     0.0   Min.   :   0.0   Min.   :    0.0   Min.   :   0.000
##  1st Qu.:     0.0   1st Qu.:   4.0   1st Qu.:    1.0   1st Qu.:   2.000
##  Median :   214.5   Median :  51.0   Median :   32.0   Median :   3.031
##  Mean   : 10330.2   Mean   : 425.9   Mean   :  630.7   Mean   :  21.428
##  3rd Qu.:  9015.5   3rd Qu.: 485.5   3rd Qu.:  602.8   3rd Qu.:   9.086
##  Max.   :209851.0   Max.   :7479.0   Max.   :11719.0   Max.   :1349.000
##   gamesPlayed        assists            misses             xp
##  Min.   :   0.0   Min.   :    0.0   Min.   :     0   Min.   :       0
##  1st Qu.:   0.0   1st Qu.:    0.0   1st Qu.:     0   1st Qu.:    2106
##  Median :   3.0   Median :   36.5   Median :  1308   Median :   63968
##  Mean   : 116.7   Mean   :  685.8   Mean   : 45357   Mean   :  872633
##  3rd Qu.: 110.5   3rd Qu.:  609.8   3rd Qu.: 40907   3rd Qu.:  828669
##  Max.   :3745.0   Max.   :14531.0   Max.   :965775   Max.   :14970539
##  scorePerMinute       shots            deaths
##  Min.   :  0.00   Min.   :      0   Min.   :    0
##  1st Qu.:  0.00   1st Qu.:      0   1st Qu.:   14
##  Median : 56.79   Median :   1565   Median :  269
##  Mean   :107.87   Mean   :  55687   Mean   : 3875
##  3rd Qu.:221.65   3rd Qu.:  50781   3rd Qu.: 3699
##  Max.   :413.80   Max.   :1166620   Max.   :67888
```

To begin with, we are going to prepare our database.Preparing the input is the most important part of the process,not having a clean database before developing different models makes next steps more difficult. We have data, but we have to clean it because if not it can be useless.

Missing values We are going to see if there are missing values in the database.

```
sum(is.na(data))
```

```
## [1] 0
```

There are not missing values, so we can continue without removing any variable or observation.

Irrelevant variables From my point of view, all the variables of the database are interesting. But there is one that will not be very usefull later. We will remove “prestige” because it refers to modes that the player can choose after progress to level 55.

```
typeof(data$prestige)
```

```
## [1] "integer"
```

```
data$prestige=NULL
```

It should not be consider as a integer but if we convert it into another type of data which is not number or integer, it would be an useless variables for the models, so we remove it.

Converting variables

```
str(data)
```

```
## 'data.frame':    1558 obs. of  18 variables:
##  $ name          : chr  "RggRt45#4697369" "JohniceRex#9176033" "bootybootykill#1892064" "J
NaCo#5244172" ...
##  $ wins          : int  0 0 0 3 0 684 4 186 741 26 ...
##  $ kills         : int  0 0 66 2 2 27011 162 1898 21803 349 ...
##  $ kdRatio       : num  0 0 1.03 0.4 0.2 ...
##  $ killstreak    : int  0 0 0 0 0 18 4 13 26 7 ...
##  $ level         : int  1 1 9 1 1 177 6 37 185 12 ...
##  $ losses        : int  0 0 0 0 0 10 2 7 29 4 ...
##  $ hits          : int  0 0 0 0 0 98332 568 5111 81361 996 ...
##  $ timePlayed    : int  0 7 32 3 5 1366 8 550 2442 44 ...
##  $ headshots     : int  0 0 16 0 1 5113 35 485 3894 40 ...
##  $ averageTime   : num  0 7 32 3 5 ...
##  $ gamesPlayed   : int  0 0 0 0 0 588 4 150 864 15 ...
##  $ assists       : int  0 0 1 0 0 6063 68 488 4029 138 ...
##  $ misses        : int  0 0 0 0 0 305319 4836 39978 327230 4844 ...
##  $ xp            : int  0 700 48300 1150 1000 3932335 24485 458269 4269370 72765 ...
##  $ scorePerMinute: num  0 0 0 0 0 ...
##  $ shots         : int  0 0 0 0 0 403651 5404 45089 408591 5840 ...
##  $ deaths        : int  0 16 64 5 10 25321 256 3332 21032 786 ...
```

As we can see, all the variables except the name are consider as numbers or integers. There are not any variable that we have to convert because the type of each variables makes sense.

Creating new variables Now, we are going to create new variables using the ones that we already have.

The first variable that we are going to create is hsps which is shotshead/shots.

```
for (i in 1:nrow(data)){
  if (data$headshots[i]==0){
    data$hsps[i]=0
  }
  if (data$shots[i]==0){
    data$hsps[i]=0
  }else{
  data$hsps[i]=data$headshots[i]/data$shots[i]
  }
}
```

We are also going to create the percentage of games won taking into account the games played. The new variable will be called wpRation

```
for (i in 1:nrow(data)){
  if (data$wins[i]==0){
    data$wpRatio[i]=0
  }
  if (data$gamesPlayed[i]==0){
    data$wpRatio[i]=0
  }else{
  data$wpRatio[i]=data$wins[i]/data$gamesPlayed[i]
  }
}
```

The last variable that we will create will be called ksk and it is killstreak/kills, the number of times that the player kill a number of enemy players without dying over the total number of times that the player kill an enemy player.

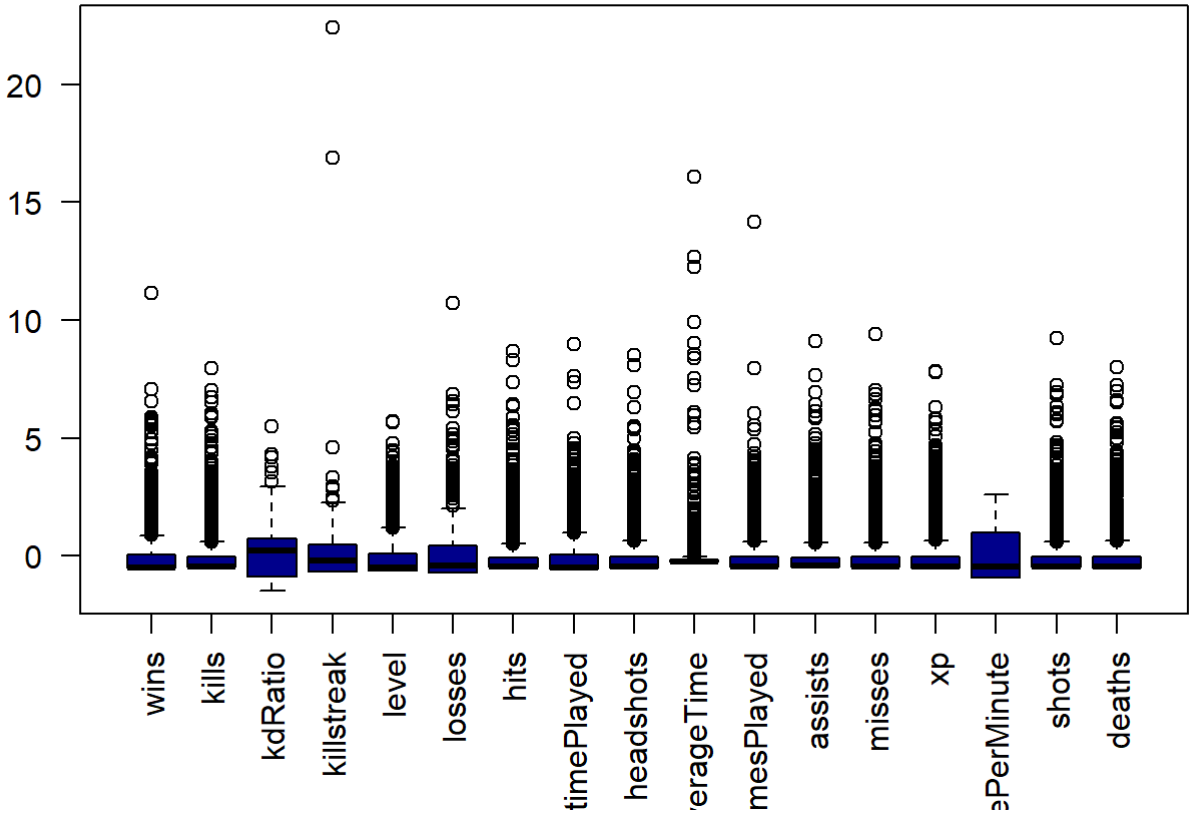
```
for (i in 1:nrow(data)){
  if (data$kills[i]==0){
    data$ksk[i]=0
  }
  if (data$killstreak[i]==0){
    data$ksk[i]=0
  }else{
  data$ksk[i]=data$killstreak[i]/data$kills[i]
  }
}
```

After using some tools I return to that point of the project, because I have realised that the variables that I had created difficult the proccess instead of helping. So although I had created with the aim of help, I think that is better to remove them.

```
data$hsps=NULL
data$wpRatio=NULL
data$ksk=NULL
```

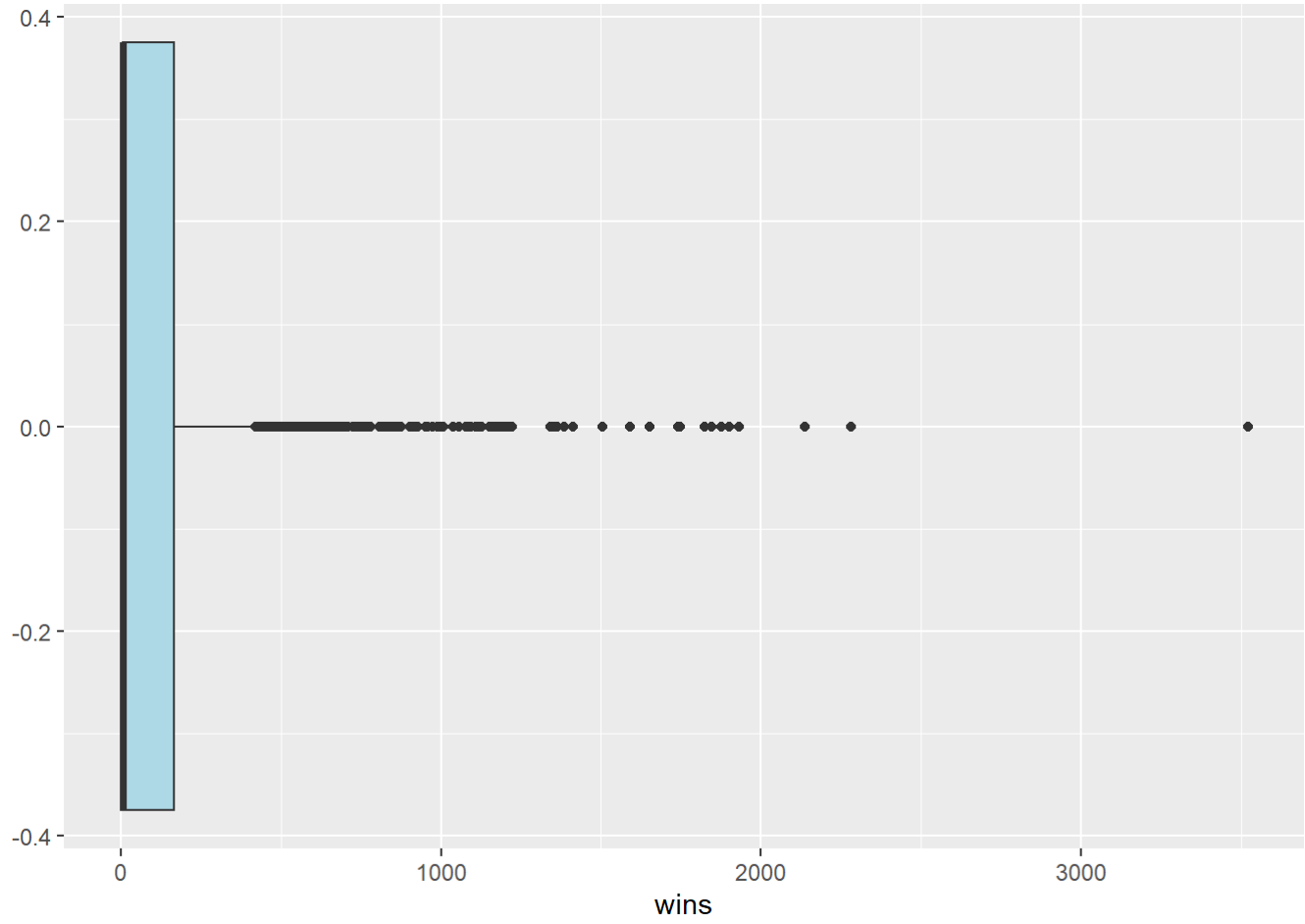
Outliers We are going to represent all the numerical and integers variable in different boxplots in order to know if aur database has oultiers. We are going to scale the data before plotting it.

```
data1=data[-1]
boxplot(scale(data1), las=2, col="darkblue")
```

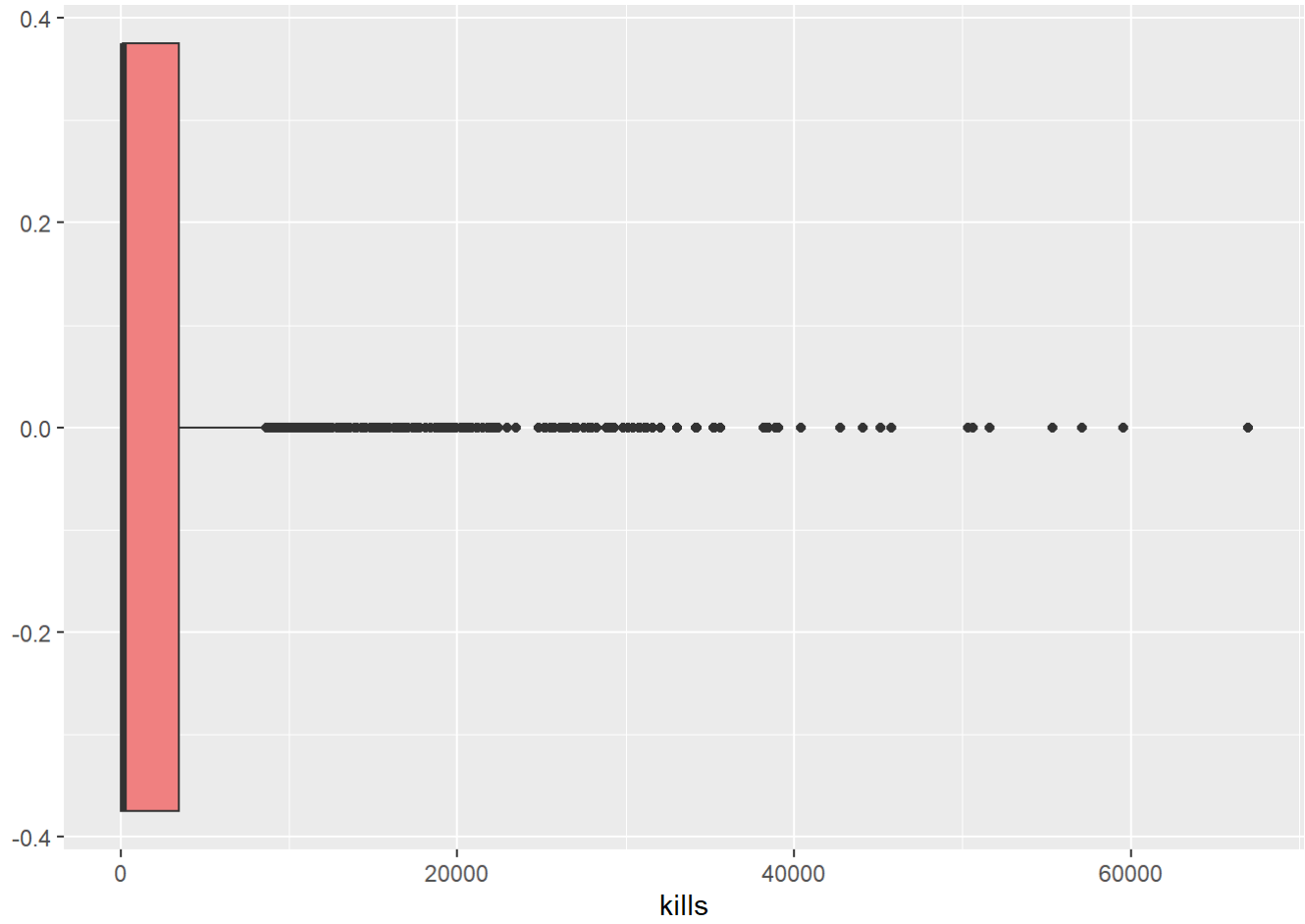


As we can see there are plenty outliers in our database. We will plot some variables alone to see outliers better.

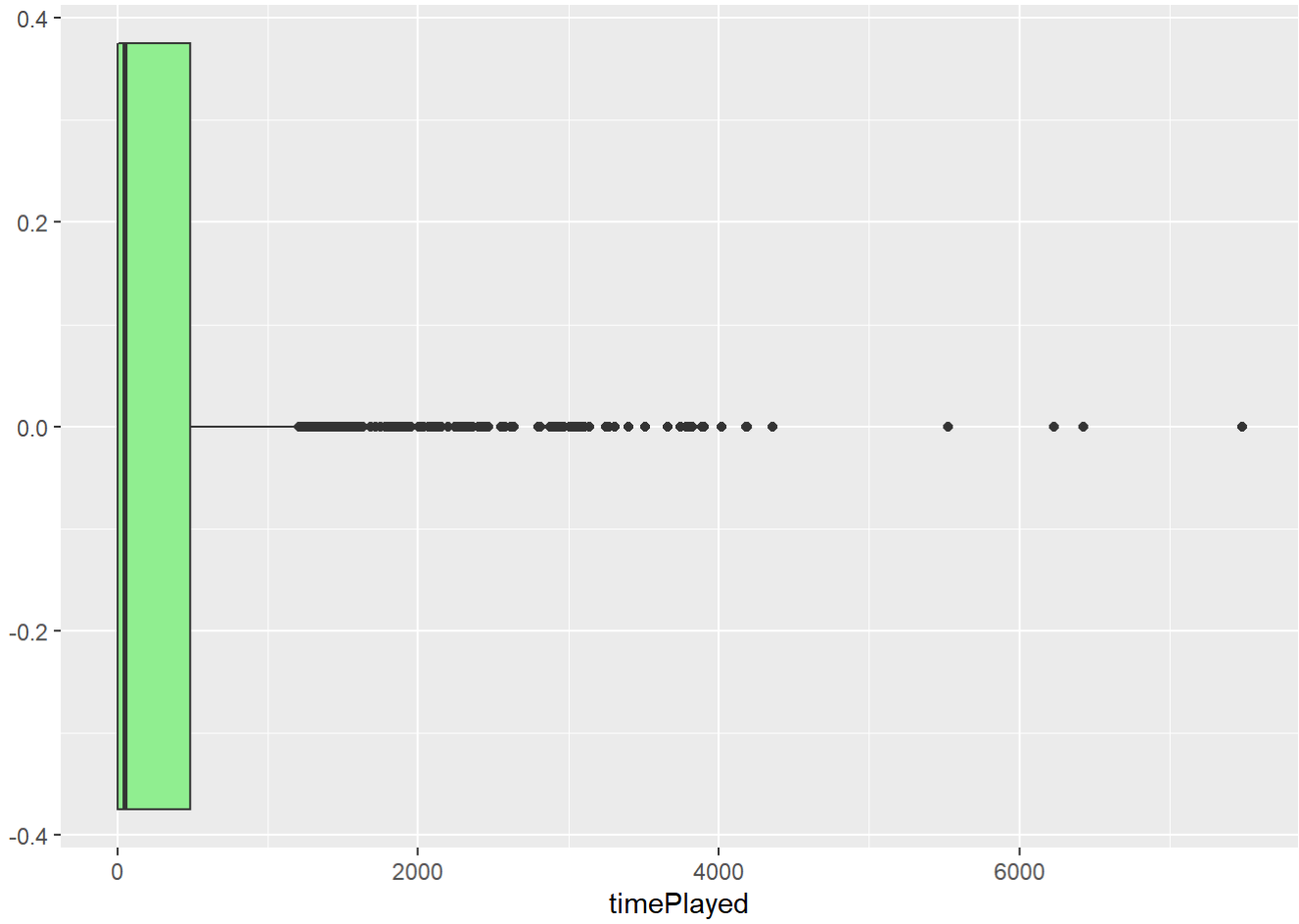
```
ggplot(data)+aes(x =wins)+geom_boxplot(fill="#ADD8E6")
```



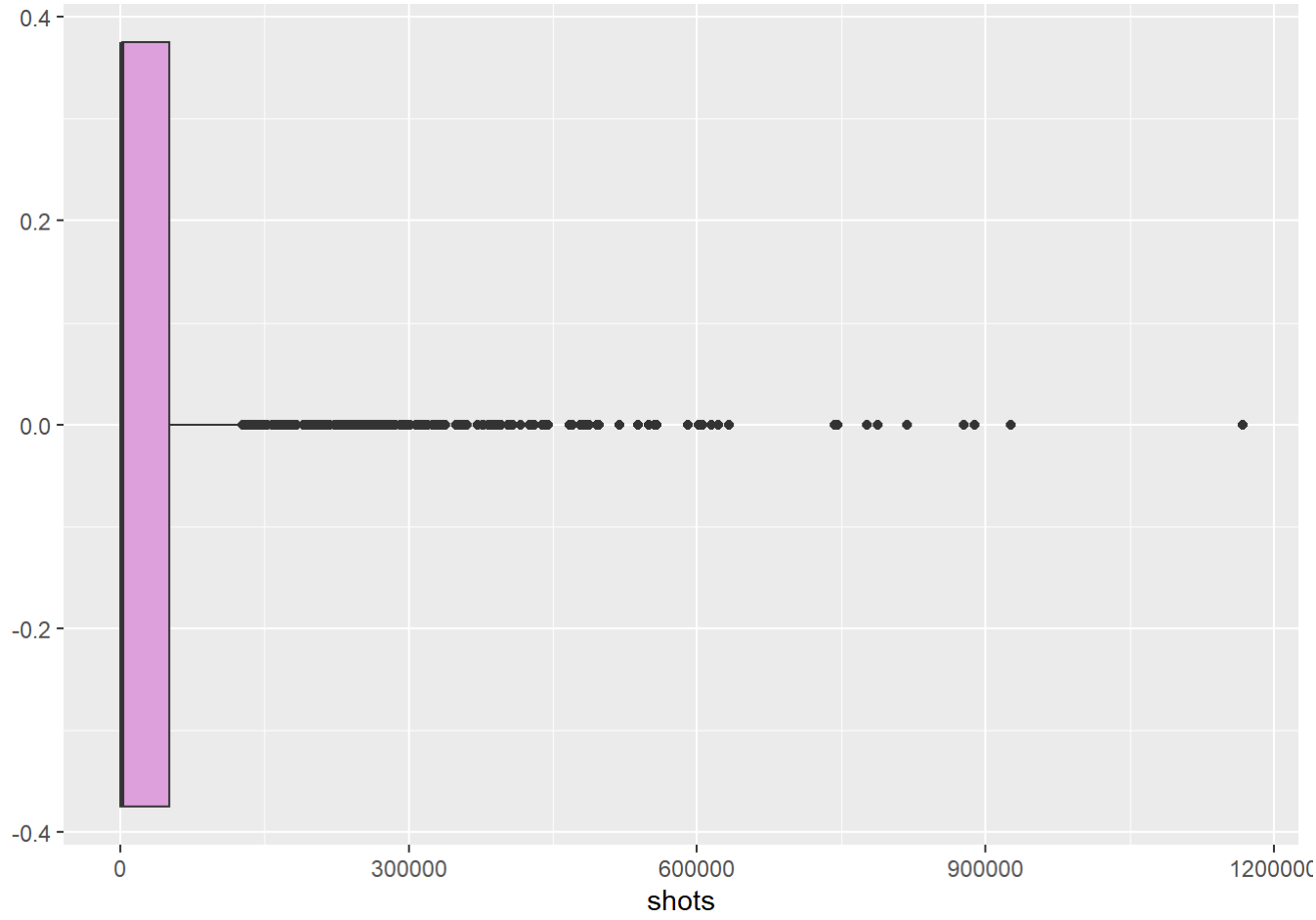

```
ggplot(data)+aes(x = kills)+geom_boxplot(fill="#F08080")
```



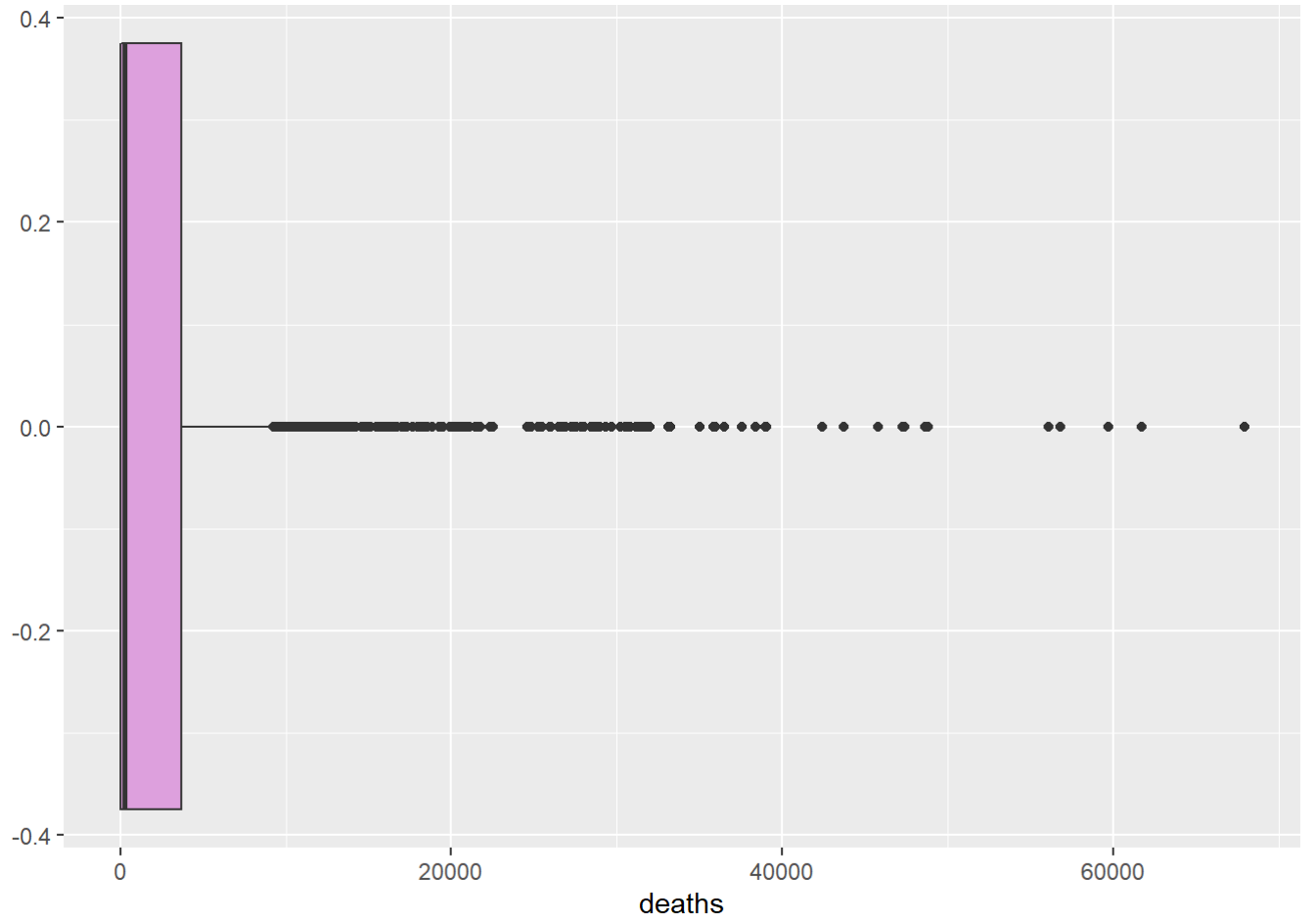
```
ggplot(data)+aes(x = timePlayed)+geom_boxplot(fill="#90EE90")
```




```
ggplot(data)+aes(x = shots)+geom_boxplot(fill="#DDA0DD")
```




```
ggplot(data)+aes(x = deaths)+geom_boxplot(fill="#DDA0DD")
```



There are too many outliers, but I think that this occurs because the values of different observations are very different. I decide no to remove any observation because I think that every observation will be important for the analysis.

As they are to many outliers, we can divide the players into levels for visualize the data. (inspiration from kaggle)

```
#Counting the numbers of players in our classification
for (i in 1:nrow(data)){
  if (data$wins[i]<=50){
    data$level[i]="1 low level"
  }
  if (50<data$wins[i] & data$wins[i]<=100){
    data$level[i]="2 mid level"
  }
  if (100<=data$wins[i] & data$wins[i]<=200){
    data$level[i]="3 high level"
  }
  if (data$wins[i]>200){
    data$level[i]="4 gods"
  }
}
typeof(data$level)
```

```
## [1] "character"
```

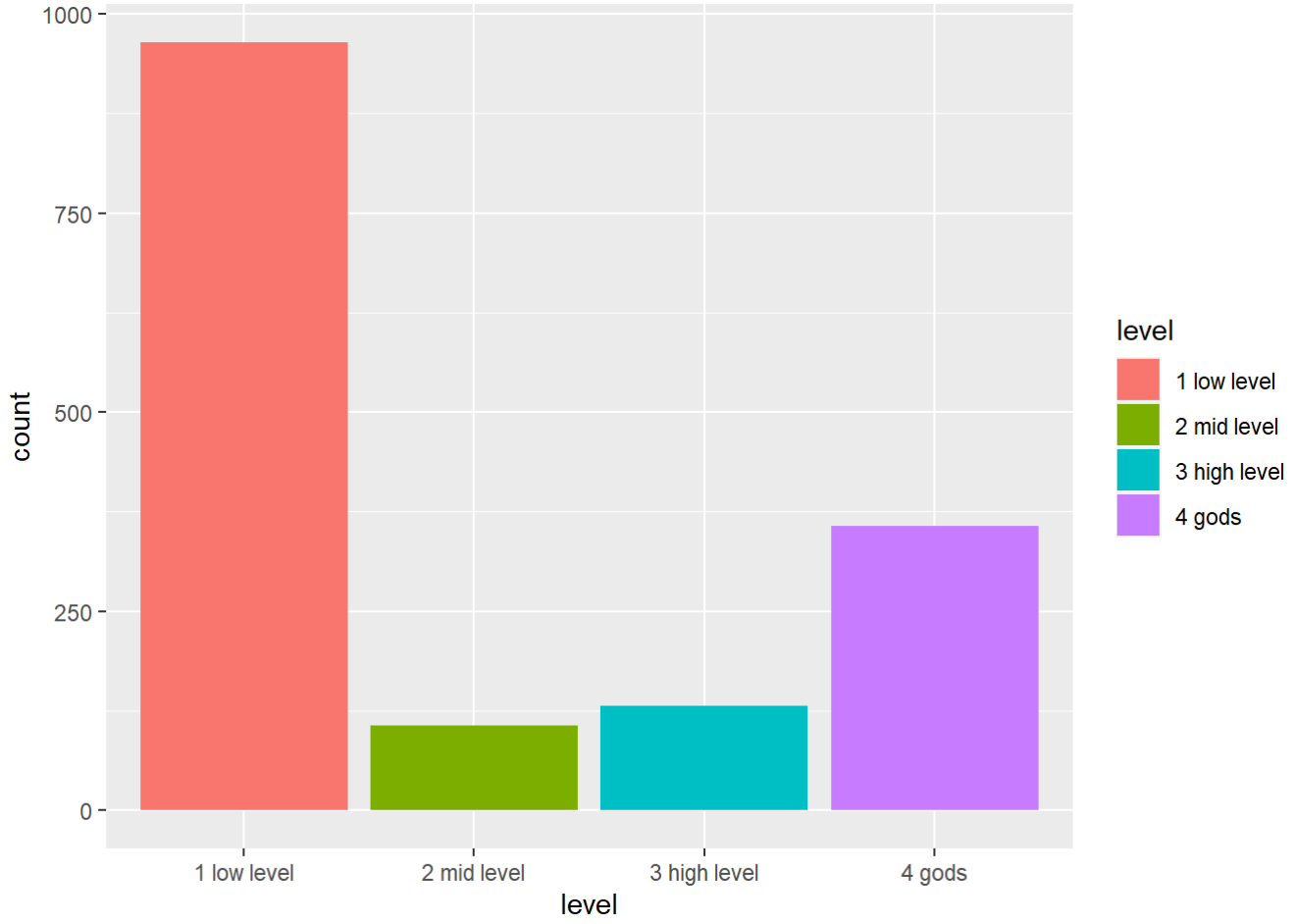
```
data$level=as.factor(data$level)
```

Visualization tools to get insights before the tools:

We are going to plot some different variables in order to understand them better and to see the relationship between some variables.

First of all we are going to plot a barplot using the variable “level” that we have just created.

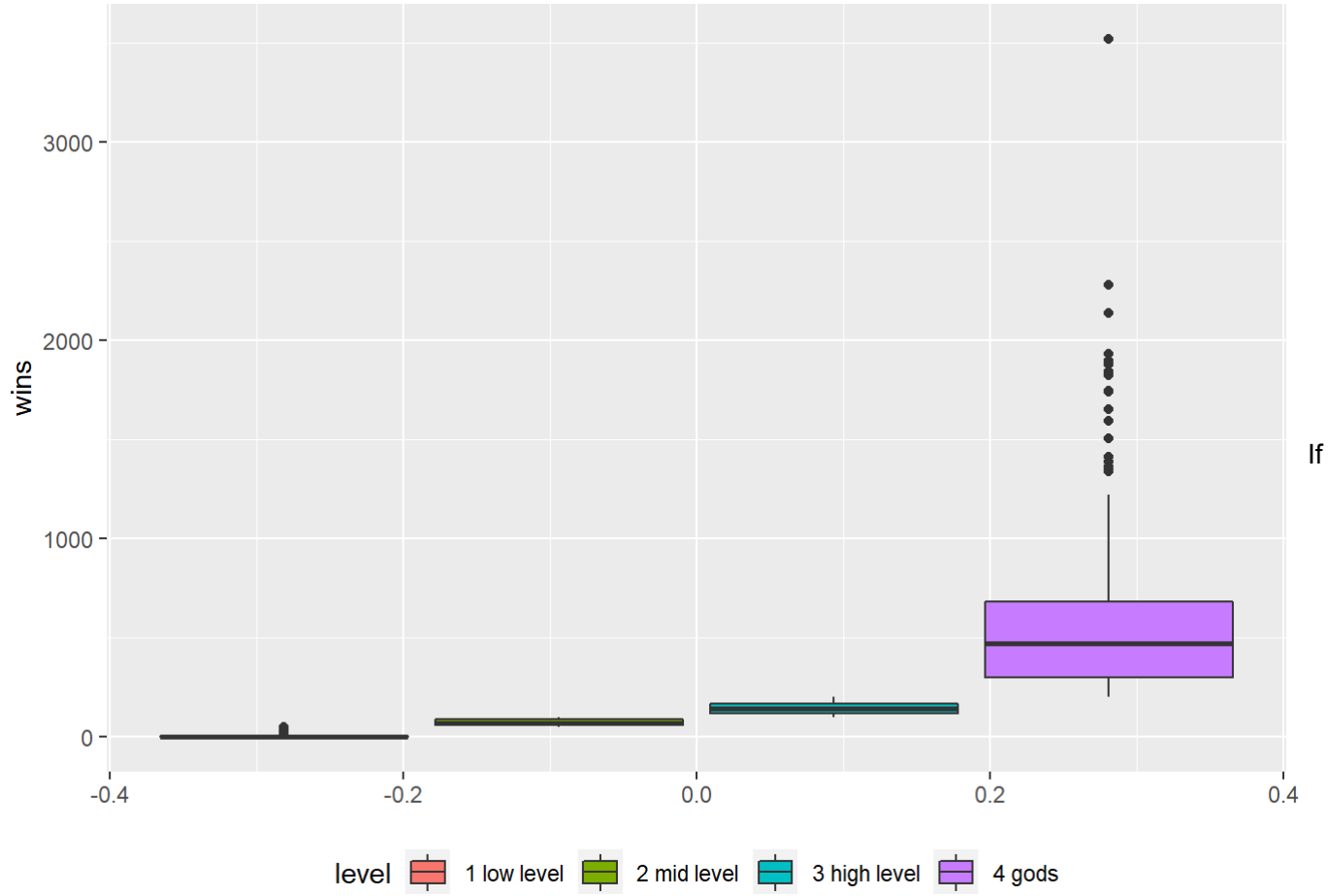
```
ggplot(data) + aes(x =level) + geom_bar(aes(fill=level))
```



As we can see most of the players are in a low level. But there are also plenty of players that are in very high levels.

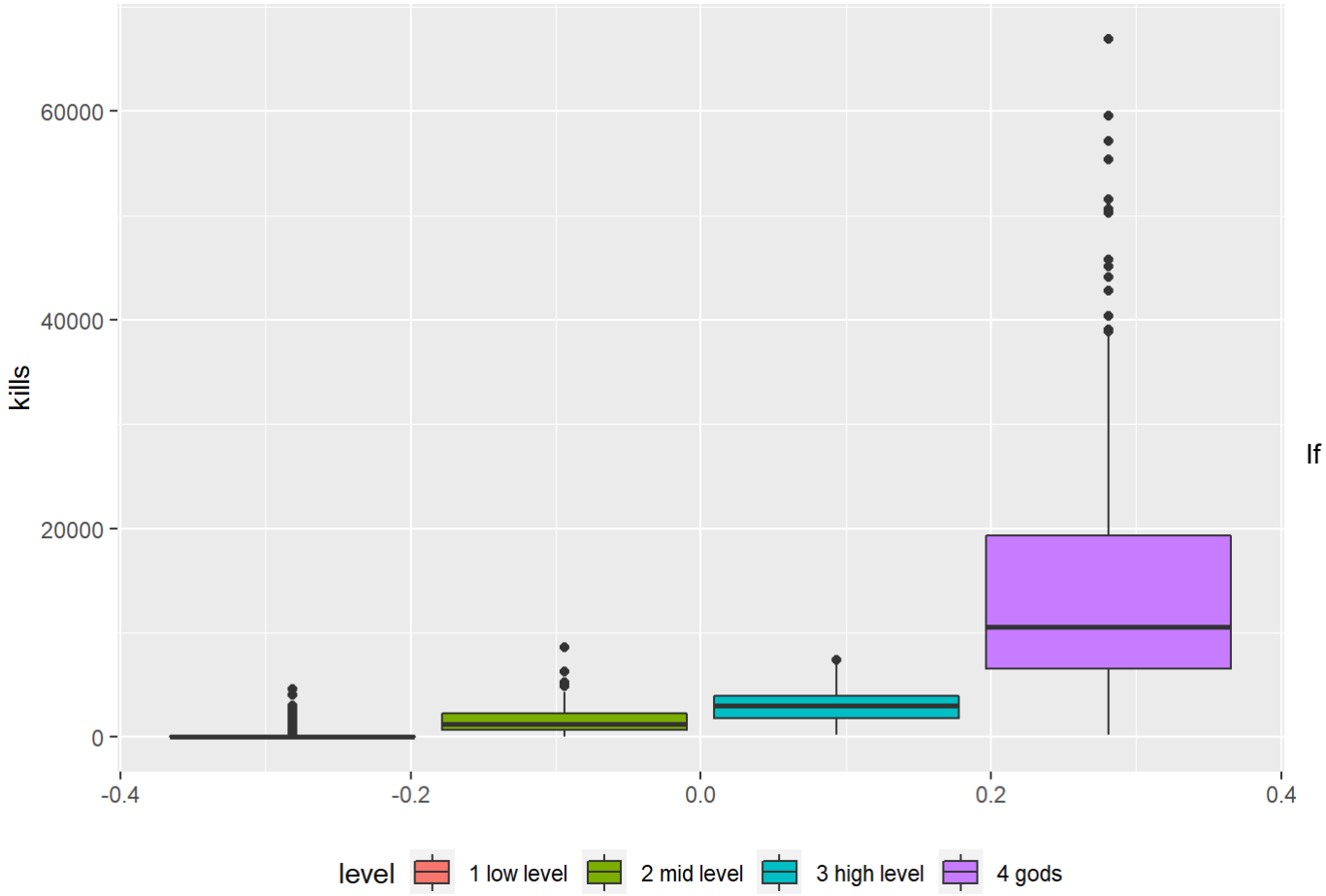
Now we are going to visualize the same variables that we have visualize to see outliers, but dividing by levels. We will realise that dividing by levels there are going to be less outliers

```
ggplot(data)+aes(y = wins, fill = level)+
geom_boxplot()+theme(legend.position = "bottom")
```



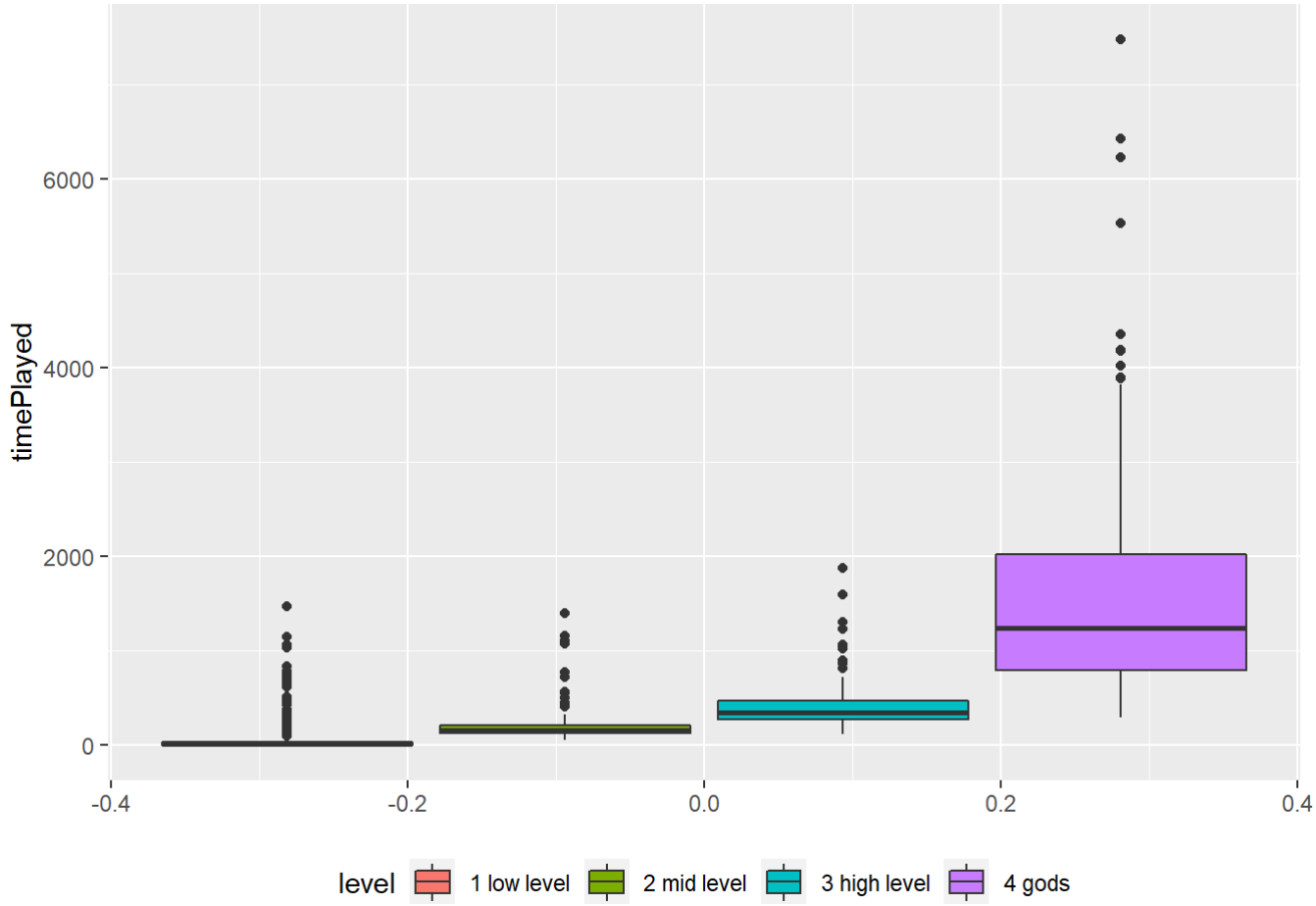
If the player is in a better lever, he wins more times.

```
ggplot(data)+aes(y = kills, fill = level)+
geom_boxplot()+theme(legend.position = "bottom")
```



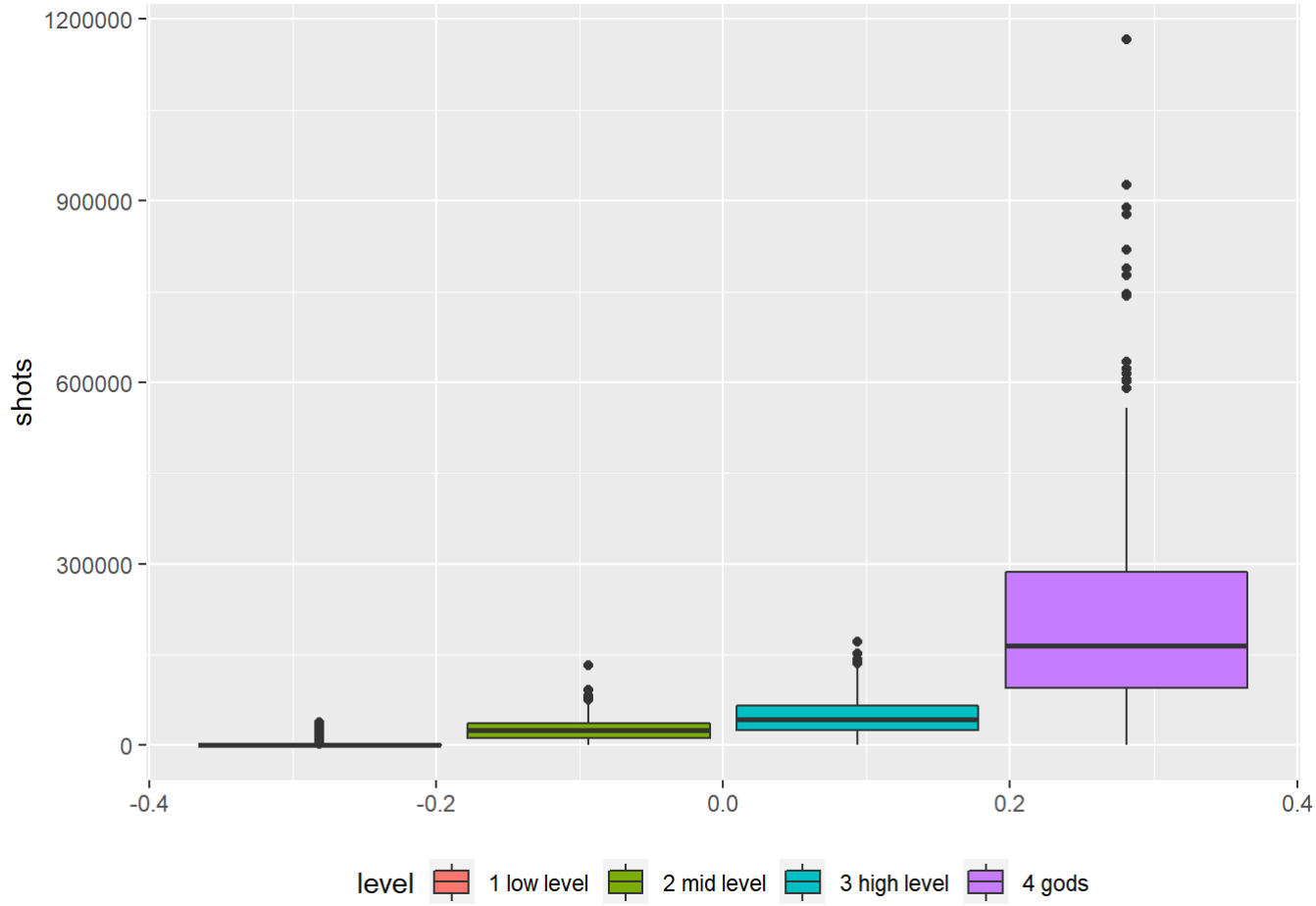
If the player is in a better level, he kill more enemies.

```
ggplot(data)+aes(y = timePlayed, fill = level)+
geom_boxplot()+theme(legend.position = "bottom")
```



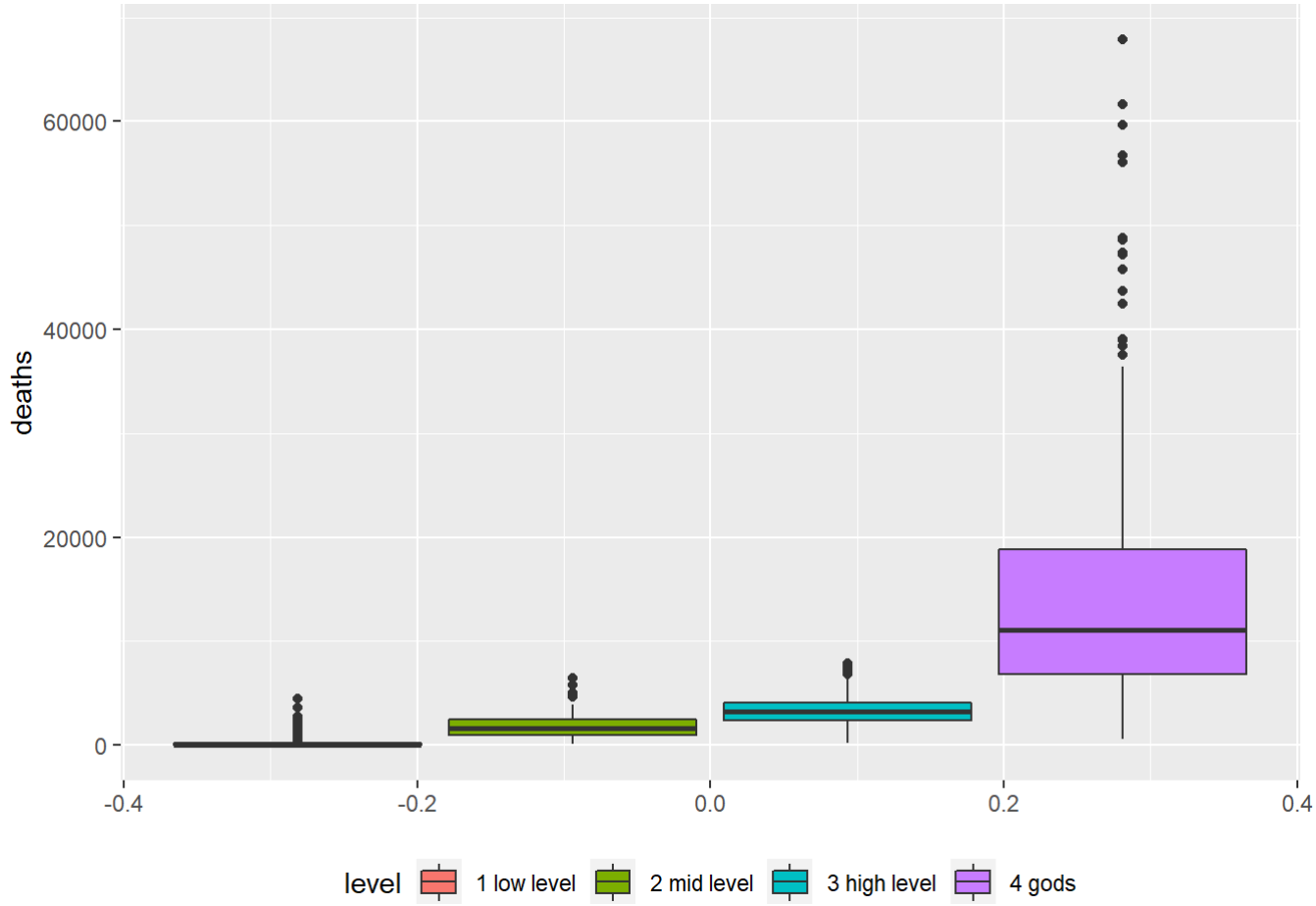
Players who are in a higher level play more time.

```
ggplot(data)+aes(y =shots, fill = level)+
geom_boxplot()+theme(legend.position = "bottom")
```



Player who are in a higher level shot more times.

```
ggplot(data)+aes(y = deaths, fill = level)+
geom_boxplot()+theme(legend.position = "bottom")
```



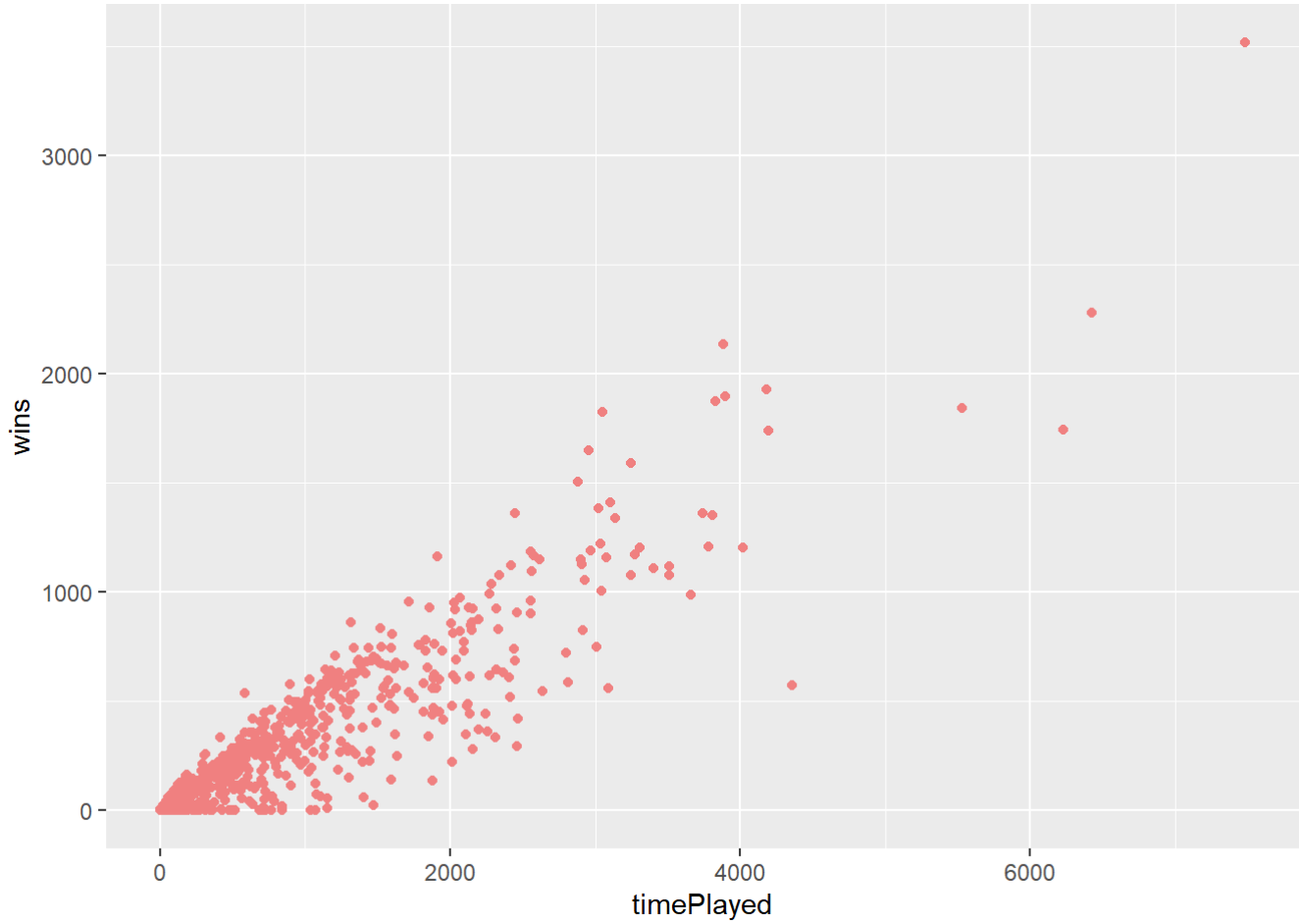
Player who are in a higher level die more times.

As we can see in the previous boxplots, there are also outliers but no as many as in the boxplot in which we have no split by levels. This occurs because there are a huge different between the values which belong to different levels, because if we divide the values into levels, we are also dividing the outliers into level.

We will plot some different scatterplots to discover if there exist a relationship between two variables.

Let´s start plotting the scatterplot of timePlayed and wins.

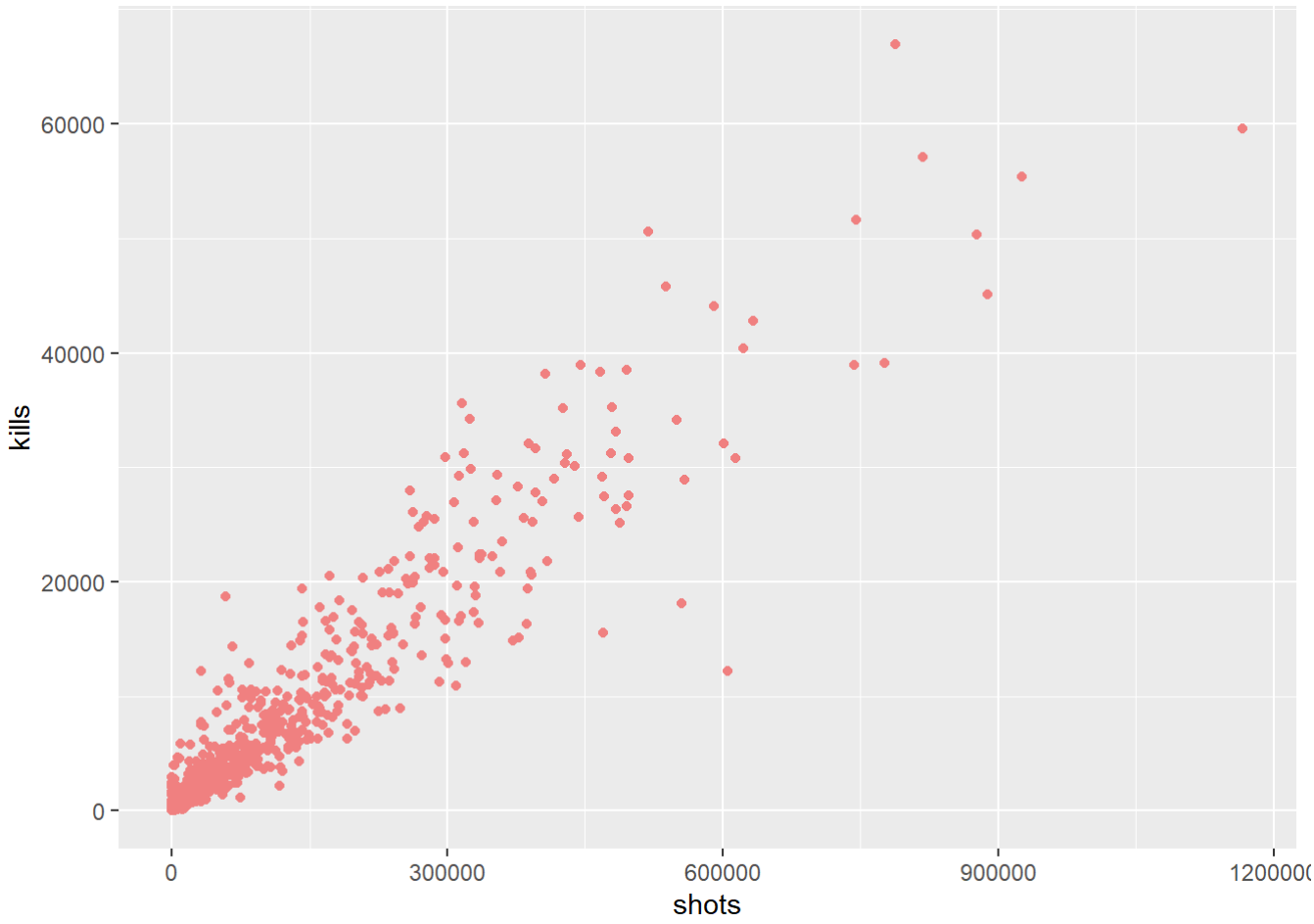
```
ggplot(data) + aes(x =timePlayed, y=wins) +
geom_point(color="#F08080")
```



As we can see in the plot these two variables have a strong relationship. If a player plays more time, he wins more times.

Let´s also see the relation ship between shots and kills.

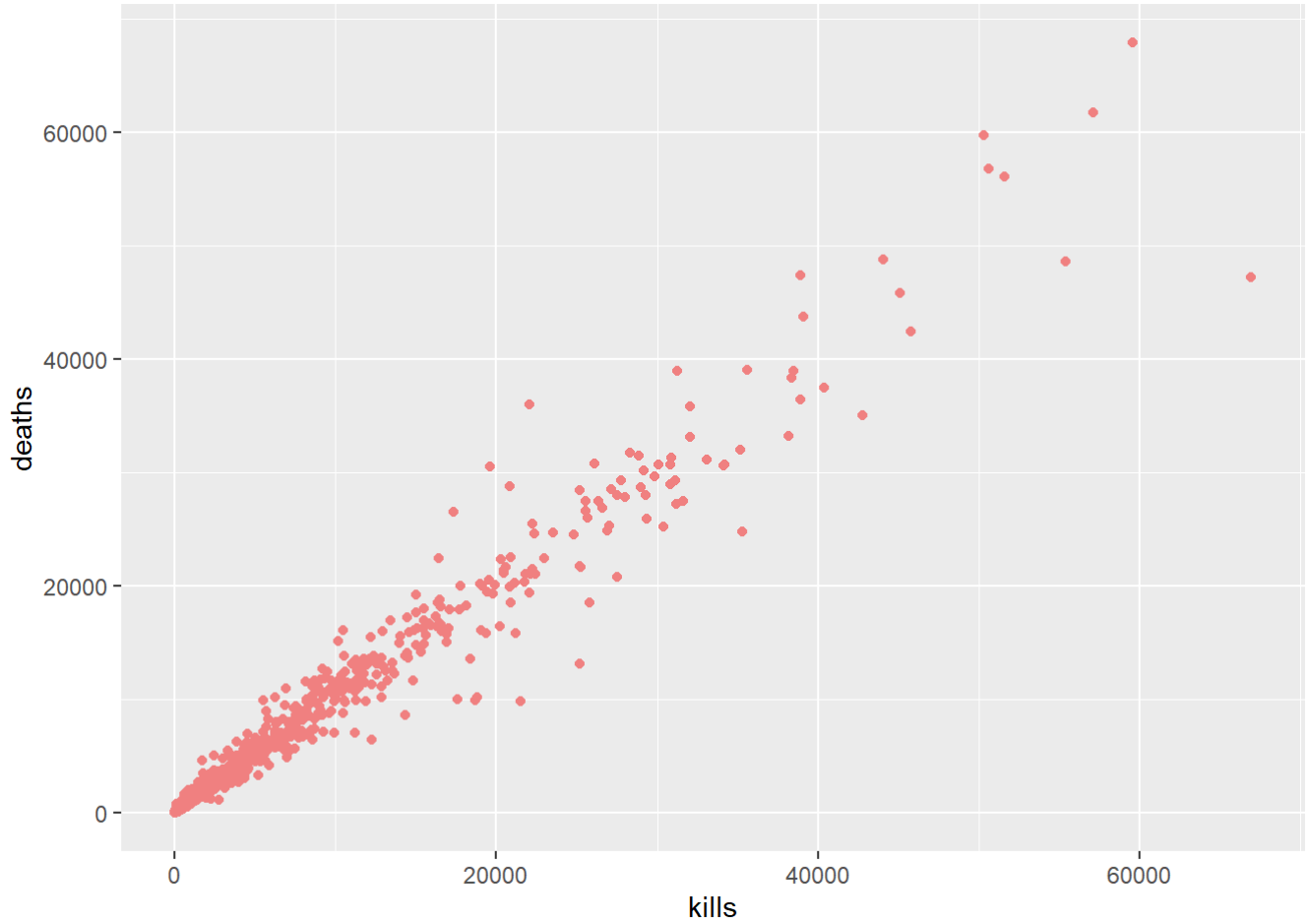
```
ggplot(data) + aes(x =shots, y=kills) +
geom_point(color="#F08080")
```



They also have a strong relationship. If a player makes more shots, he kill more people.

Finally, we will see the relationship between deaths and kills

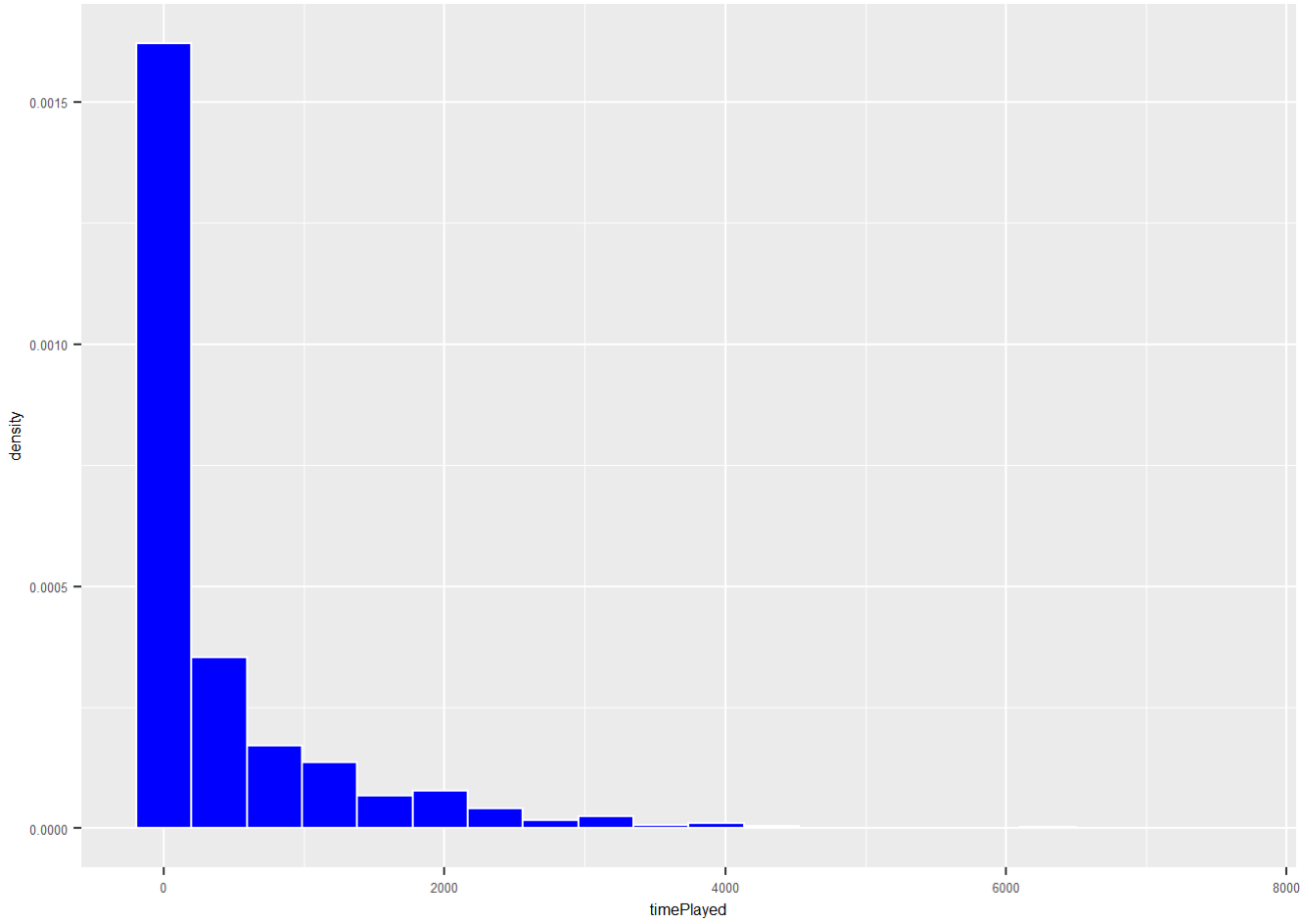
```
ggplot(data) + aes(x =kills, y=deaths) +
geom_point(color="#F08080")
```



As we can see, there is a strong relationship between the number of times that a player kill an enemy and the number of times that a player deaths.

Finally, let´s plot the variable “timePlayed”

```
ggplot(data) + aes(x = timePlayed) +
geom_histogram(aes(y = ..density..),
bins = 20,fill="blue", color = "white")+
theme(text = element_text(size = 6))
```



In there we can see that in the database there are plenty of people that do not play too much. Most of the people of the database do not played too much.

##PRINCIPAL COMPONENT ANALYSIS

The first tool that we will use is PCA. This tool represents the multivariate information with a smaller number of variables without loosing much information. This tool is usefull if variables are correlated, and as we have seen in the previous scatterplots, we have correlated variables.

```
library(tidyverse)
library(GGally)
library(factoextra)
```

First of all, we are going to prepare the input for the PCA. We create an specific database equal to the original one, but we remove the values that are not numeric or integers.

```
names = data[,1]
data_pca=data
data_pca$level=NULL
data_pca$name=NULL
dim(data)
```

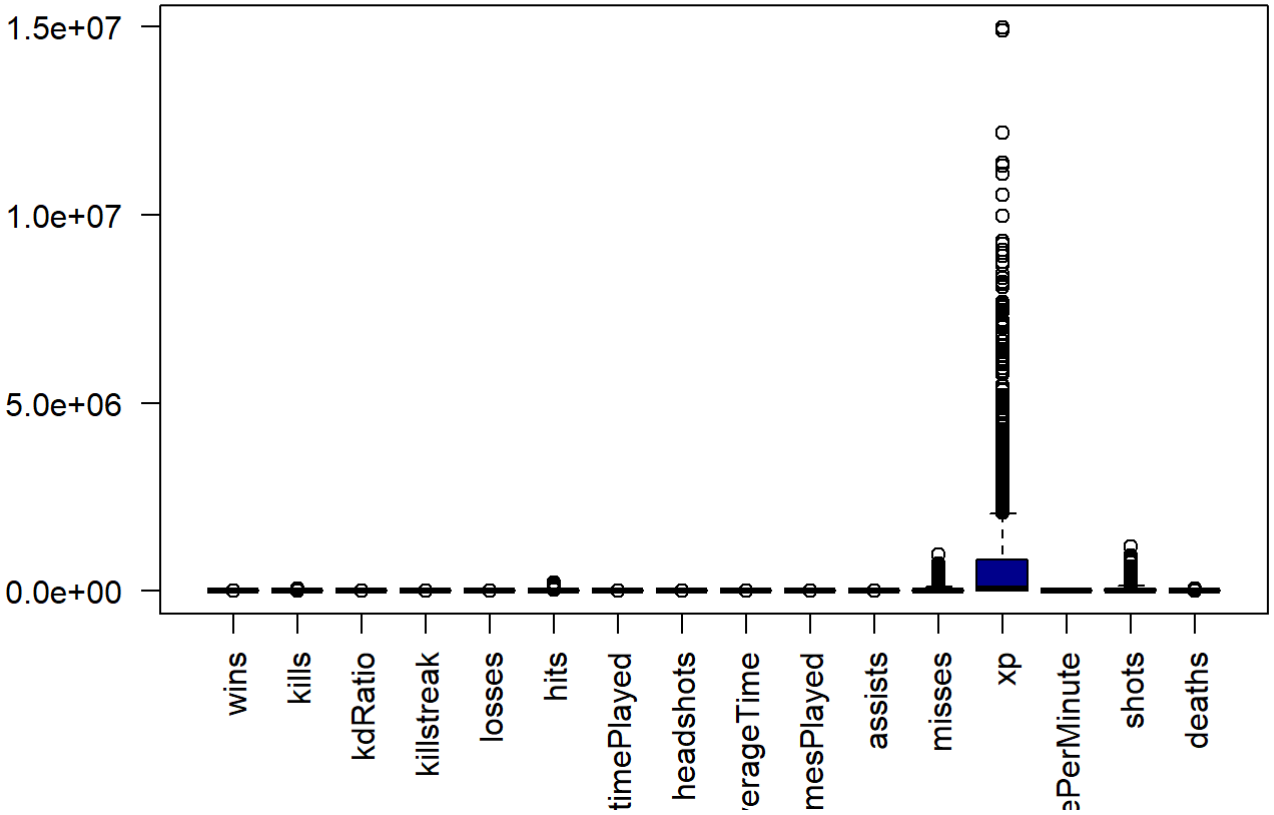
```
## [1] 1558   18
```

```
summary(data)
```

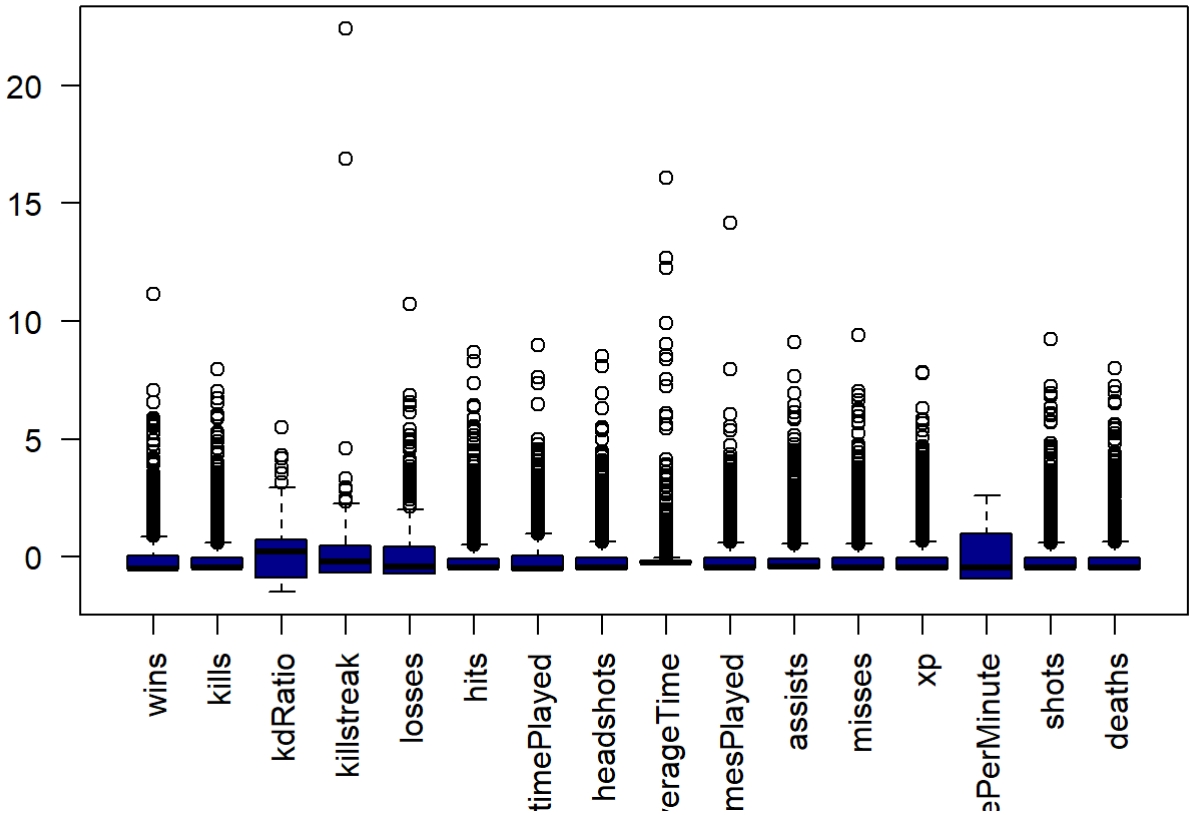
```
##      name                wins          kills            kdRatio
##  Length:1558        Min.   :   0   Min.   :    0.0   Min.   :0.0000
##  Class :character   1st Qu.:   0   1st Qu.:    4.0   1st Qu.:0.2614
##  Mode  :character   Median :  10   Median :  191.5   Median :0.7328
##                     Mean   : 153   Mean   : 3753.0   Mean   :0.6371
##                     3rd Qu.: 168   3rd Qu.: 3445.8   3rd Qu.:0.9553
##                     Max.   :3519   Max.   :66935.0   Max.   :3.0000
##    killstreak               level         losses            hits
##  Min.   :  0.000   1 low level :964   Min.   : 0.000   Min.   :     0.0
##  1st Qu.:  0.000   2 mid level :106   1st Qu.: 0.000   1st Qu.:     0.0
##  Median :  5.000   3 high level:131   Median : 2.000   Median :   214.5
##  Mean   :  6.895   4 gods      :357   Mean   : 4.998   Mean   : 10330.2
##  3rd Qu.: 12.000                      3rd Qu.: 8.000   3rd Qu.:  9015.5
##  Max.   :235.000                      Max.   :80.000   Max.   :209851.0
##    timePlayed       headshots        averageTime        gamesPlayed
##  Min.   :   0.0   Min.   :    0.0   Min.   :   0.000   Min.   :   0.0
##  1st Qu.:   4.0   1st Qu.:    1.0   1st Qu.:   2.000   1st Qu.:   0.0
##  Median :  51.0   Median :   32.0   Median :   3.031   Median :   3.0
##  Mean   : 425.9   Mean   :  630.7   Mean   :  21.428   Mean   : 116.7
##  3rd Qu.: 485.5   3rd Qu.:  602.8   3rd Qu.:   9.086   3rd Qu.: 110.5
##  Max.   :7479.0   Max.   :11719.0   Max.   :1349.000   Max.   :3745.0
##     assists            misses             xp           scorePerMinute
##  Min.   :    0.0   Min.   :     0   Min.   :       0   Min.   :  0.00
##  1st Qu.:    0.0   1st Qu.:     0   1st Qu.:    2106   1st Qu.:  0.00
##  Median :   36.5   Median :  1308   Median :   63968   Median : 56.79
##  Mean   :  685.8   Mean   : 45357   Mean   :  872633   Mean   :107.87
##  3rd Qu.:  609.8   3rd Qu.: 40907   3rd Qu.:  828669   3rd Qu.:221.65
##  Max.   :14531.0   Max.   :965775   Max.   :14970539   Max.   :413.80
##      shots             deaths
##  Min.   :      0   Min.   :    0
##  1st Qu.:      0   1st Qu.:   14
##  Median :   1565   Median :  269
##  Mean   :  55687   Mean   : 3875
##  3rd Qu.:  50781   3rd Qu.: 3699
##  Max.   :1166620   Max.   :67888
```

Let´s start repeating a descriptive analysis Firstly, we plot the data in a boxplot

```
boxplot(data_pca, las=2, col="darkblue")
```



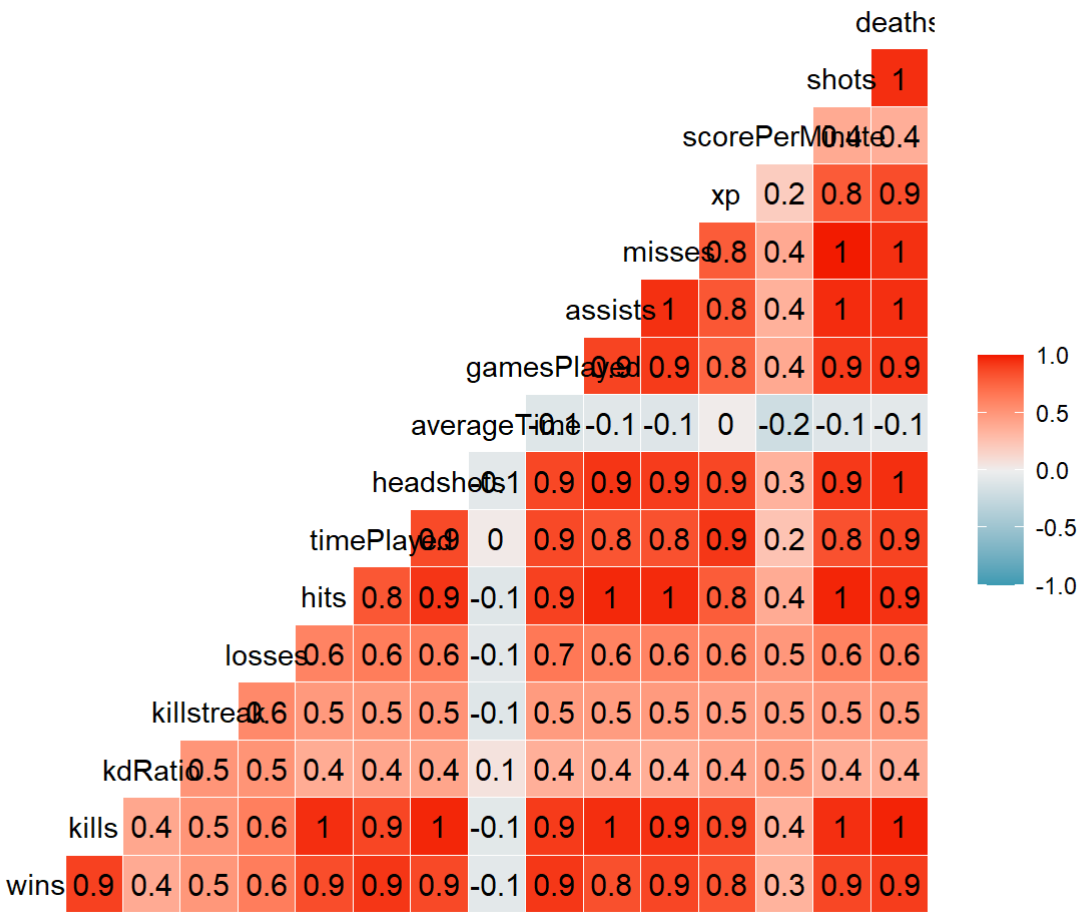

```
boxplot(scale(data_pca), las=2, col="darkblue")
```



correlation between variables

Now, we plot the data in a way that allows as to see the correlation between the variables. As more red, more correlated.(PCA consists on replace a large number of correlated variables by a smaller number of uncorrelated ones)

```
ggcorr(data_pca, label = T)
```



As we can see there are lots of variables correlated

PCA

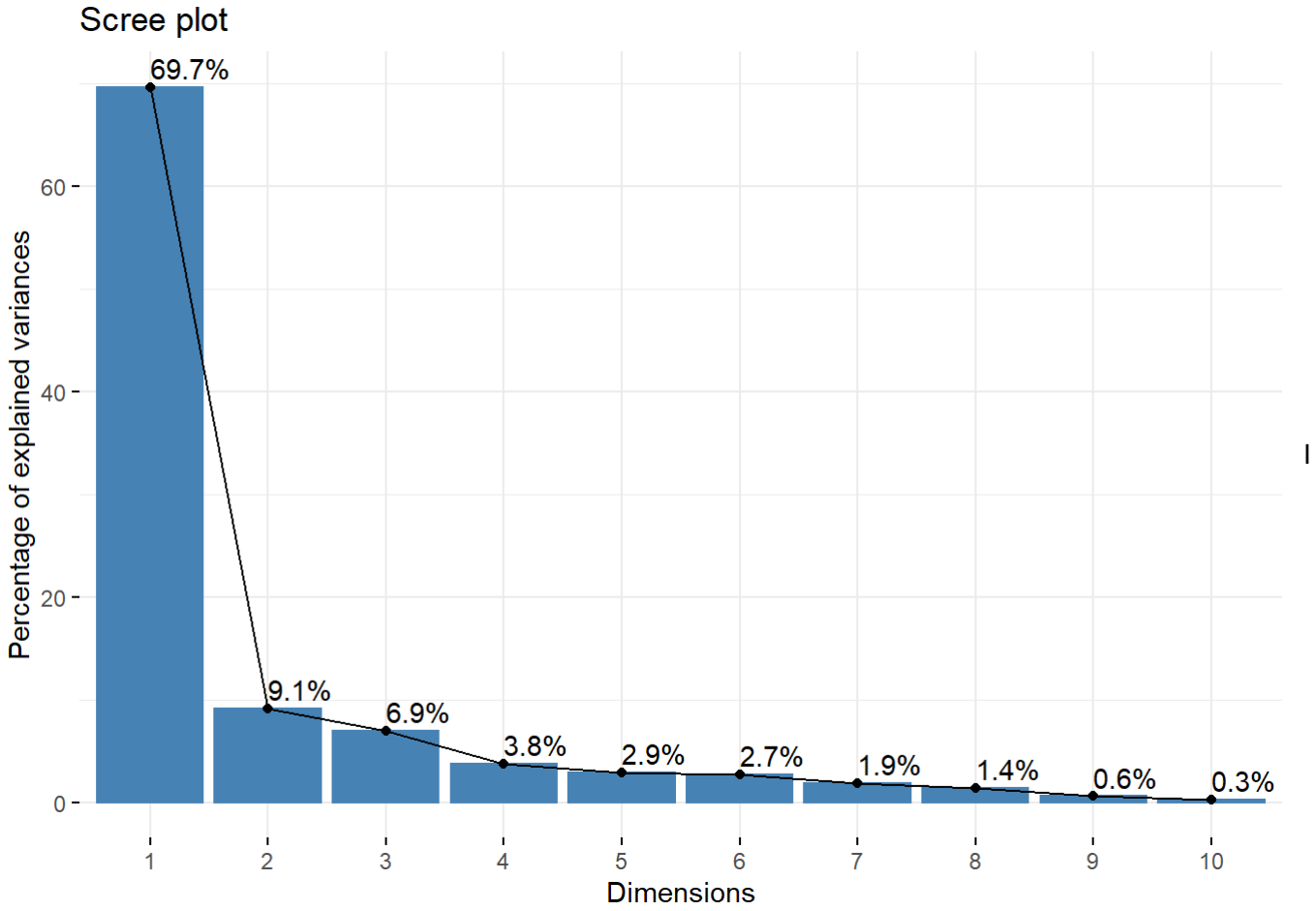
Let´s apply PCA (we use standarized data)

```
pca = prcomp(data_pca, scale=T)
summary(pca)
```

```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     3.3386 1.20979 1.05389 0.77621 0.68338 0.6573 0.54622
## Proportion of Variance 0.6966 0.09147 0.06942 0.03766 0.02919 0.0270 0.01865
## Cumulative Proportion  0.6966 0.78812 0.85754 0.89519 0.92438 0.9514 0.97003
##                           PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.47143 0.30267 0.20864 0.20295 0.18643 0.15061 0.12888
## Proportion of Variance 0.01389 0.00573 0.00272 0.00257 0.00217 0.00142 0.00104
## Cumulative Proportion  0.98392 0.98964 0.99237 0.99494 0.99711 0.99853 0.99957
##                           PC15      PC16
## Standard deviation     0.08315 2.305e-05
## Proportion of Variance 0.00043 0.000e+00
## Cumulative Proportion  1.00000 1.000e+00
```

Number of chosen components After doing PCA, let´s plot the percentage of variability that explains each principal component

```
fviz_screeplot(pca, addlabels = TRUE)
```

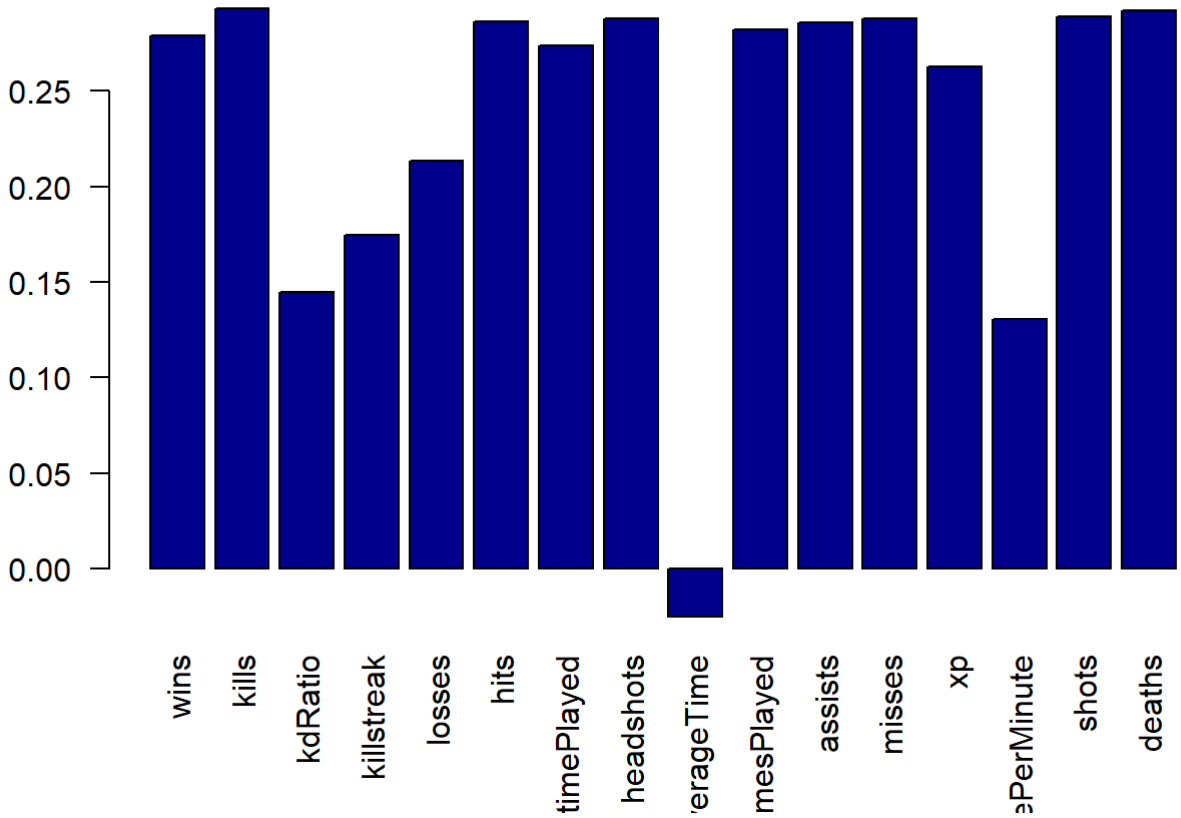


´m going to choose 3 components.I think that is the best option because with two components we we explain around 85% of variability and I think that it is enough

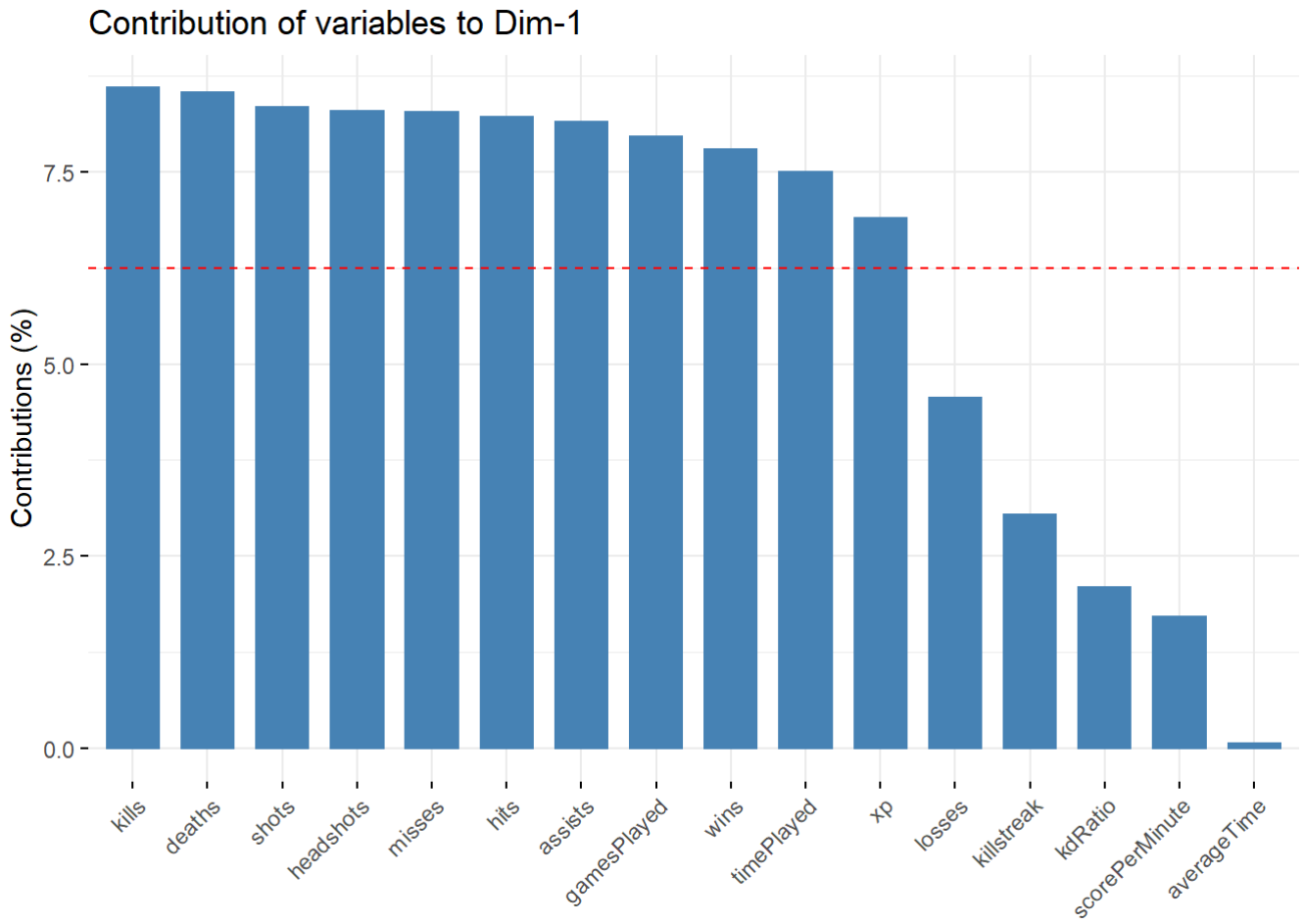
Interpretation of components Now, we will plot the contributions of each variable in each component and we will try to interpret each component

First component

```
barplot(pca$rotation[,1], las=2, col="darkblue")
```



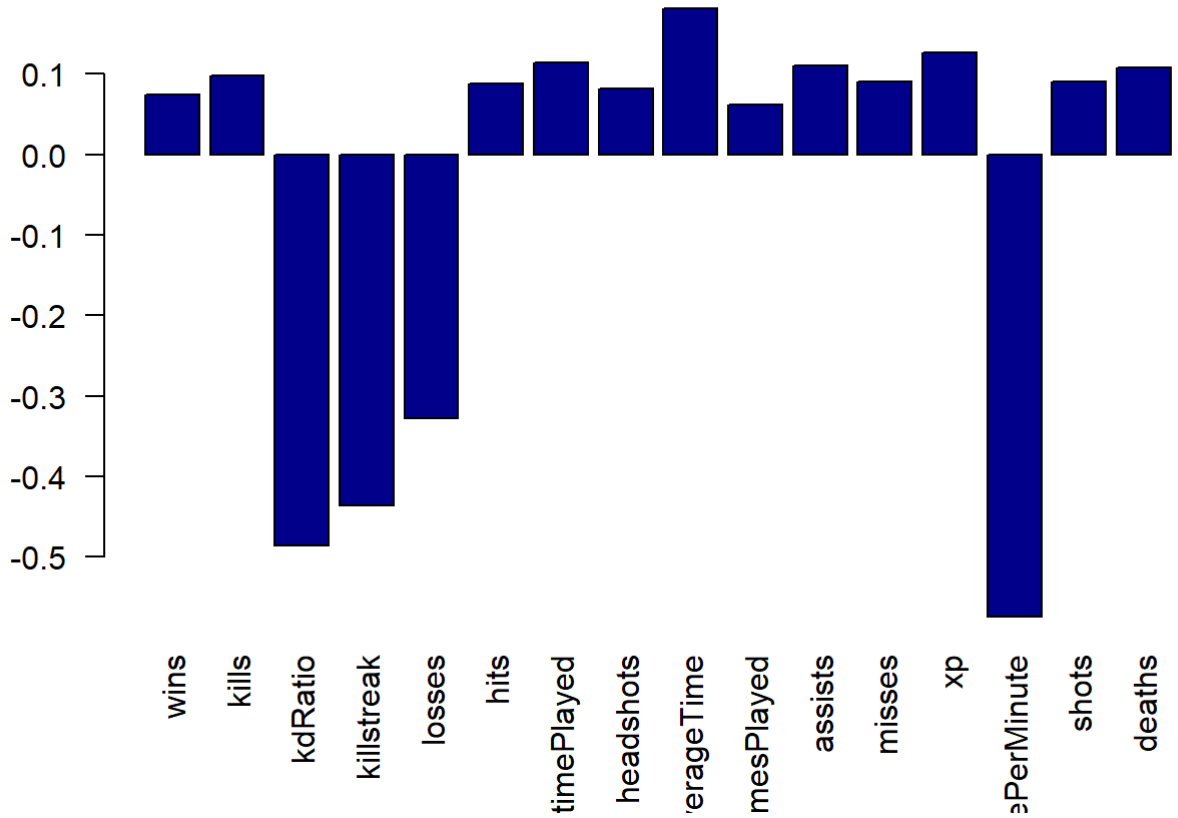

```
fviz_contrib(pca, choice = "var", axes = 1)
```



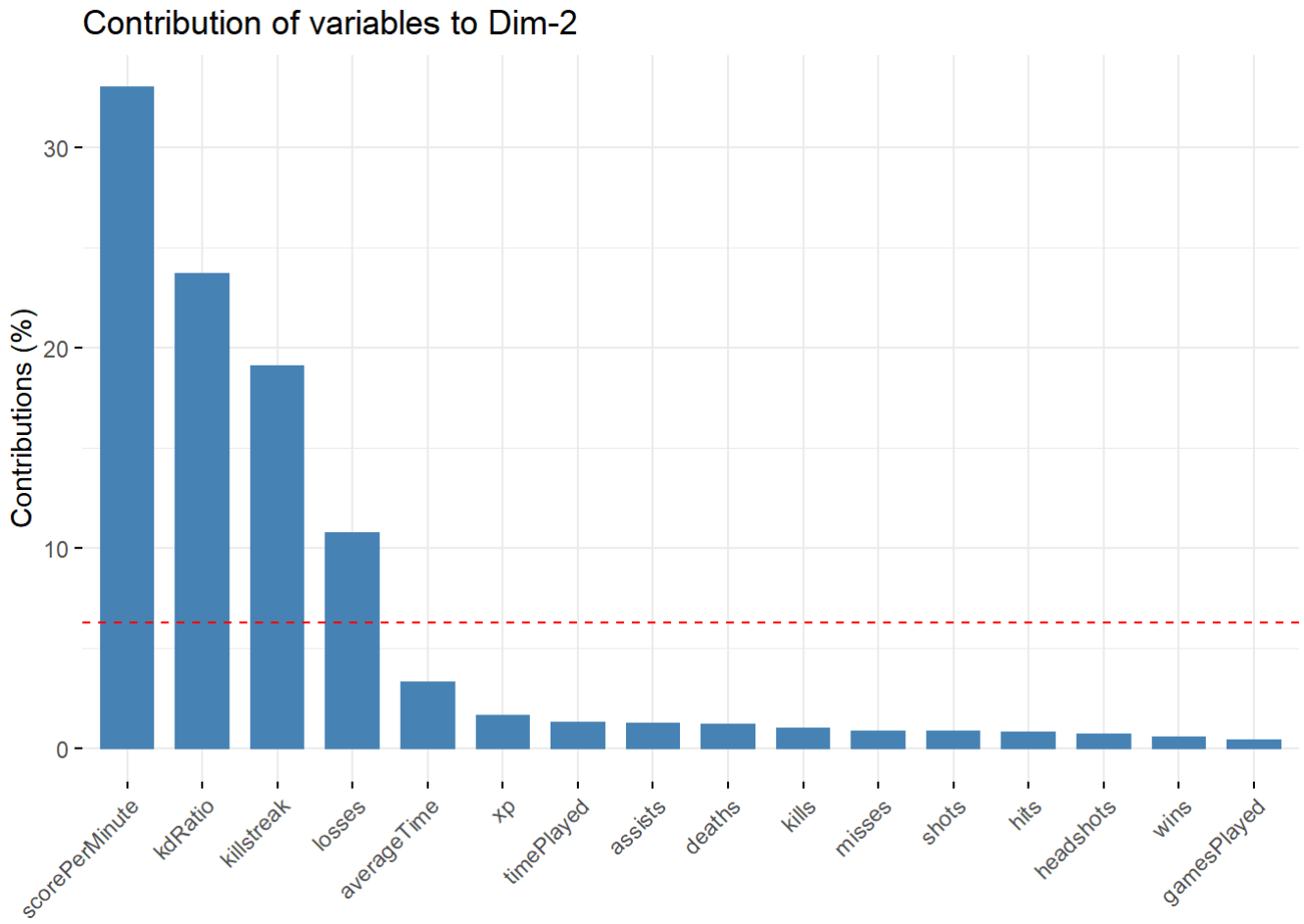
In PC1 the variables that contributes the most are kills, dealths, shots, headshots, missed,hits,assists,gamesPlayed, wins and xp. First component could be an weighted average about the variables.

Second component

```
barplot(pca$rotation[,2], las=2, col="darkblue")
```



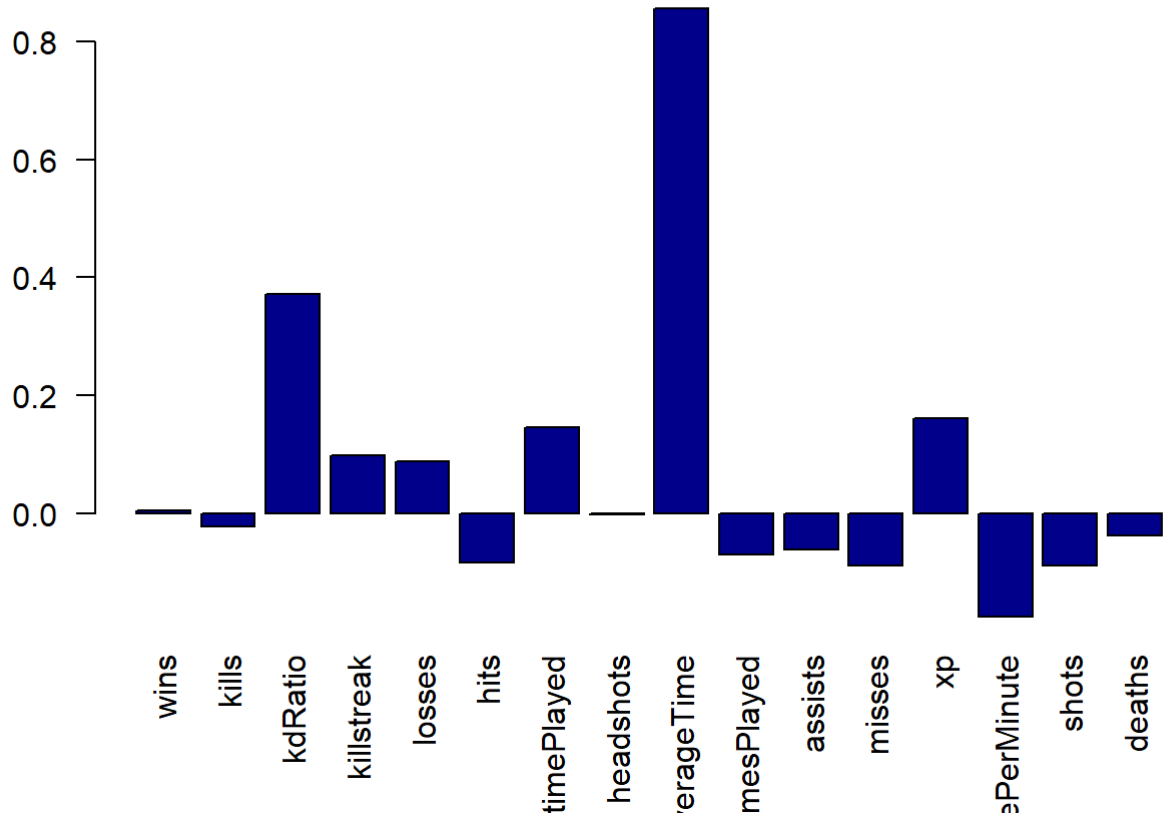

```
fviz_contrib(pca, choice = "var", axes = 2)
```



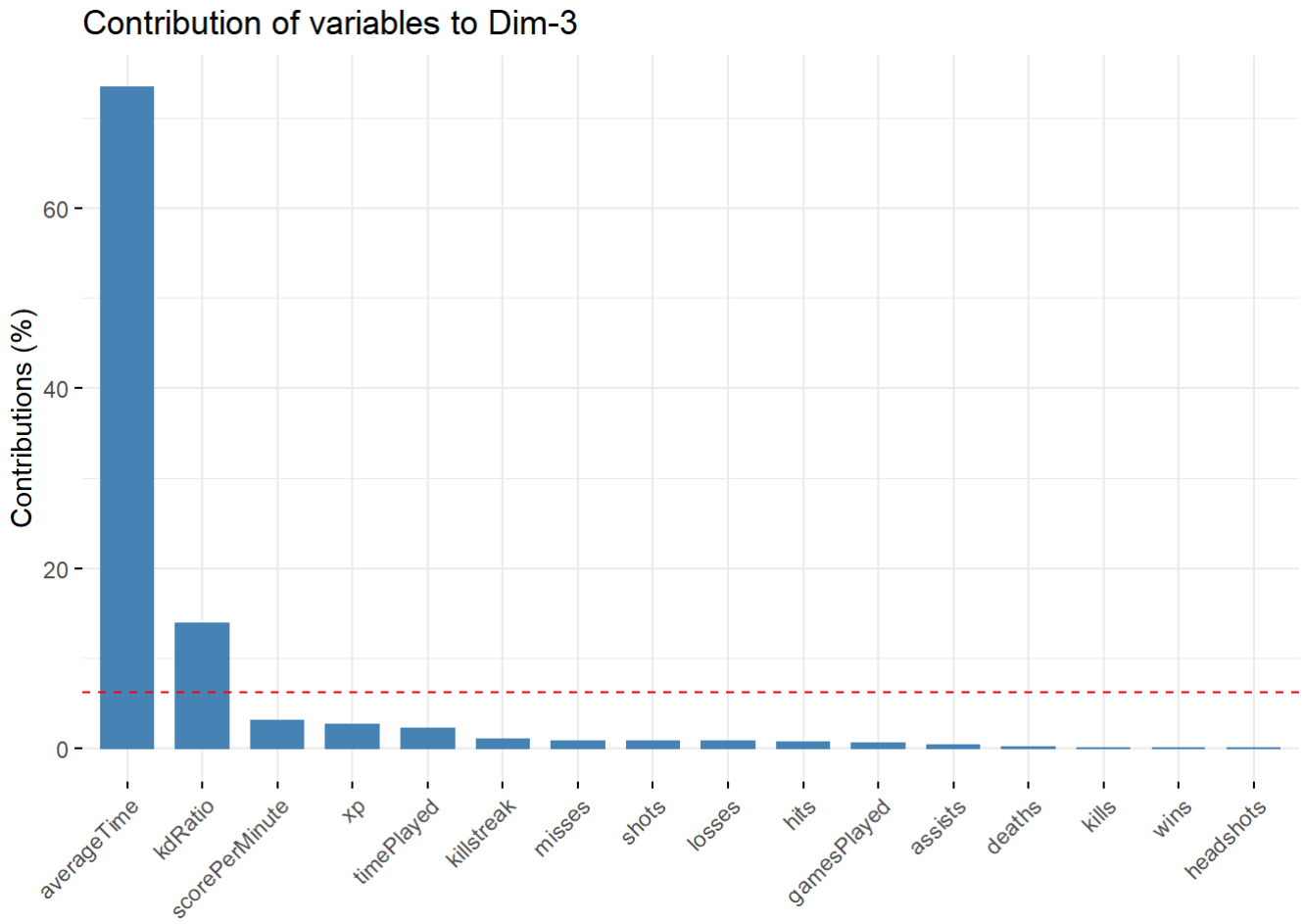
In PC2 the variables that contributes the most are scorePerMinute, KdRatio, killstreak and losses as is shown in the plot. Maybe the second component is be obtained by contrasting variables, but I don´t understand how it is really obtained.

Third component

```
barplot(pca$rotation[,3], las=2, col="darkblue")
```



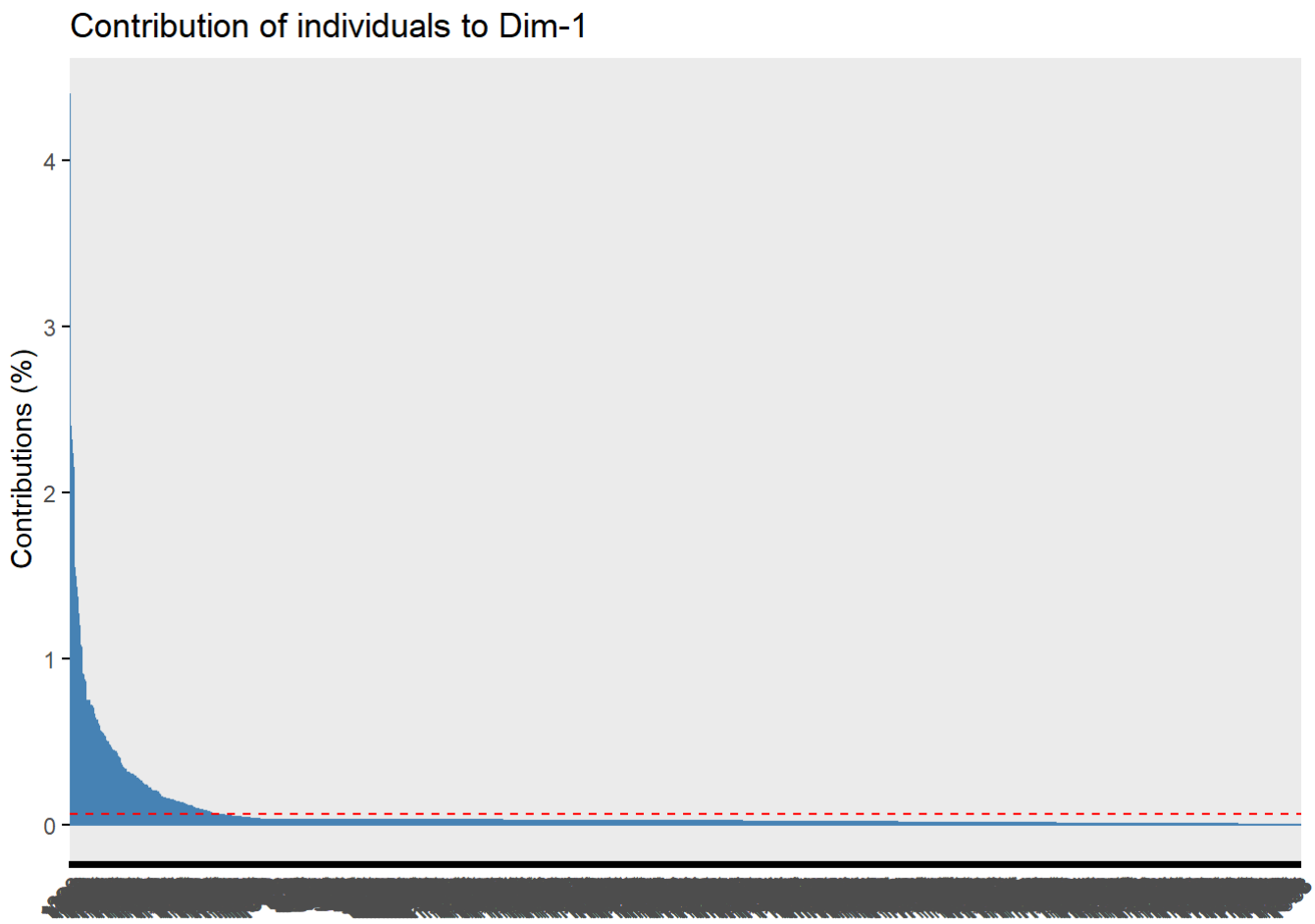

```
fviz_contrib(pca, choice = "var", axes = 3)
```



In the third component the variables that contributes the most are averageTime kdRatio, scorePerMinute, killstreak, losses, wpRatio, averageTime,hsps, as we can see in the plot. I think that it have been obtained contrasting groups of variables, but i don´t know the specific way.

contribution of each player first component Now, we are going to see the contribution of all players to the first component.

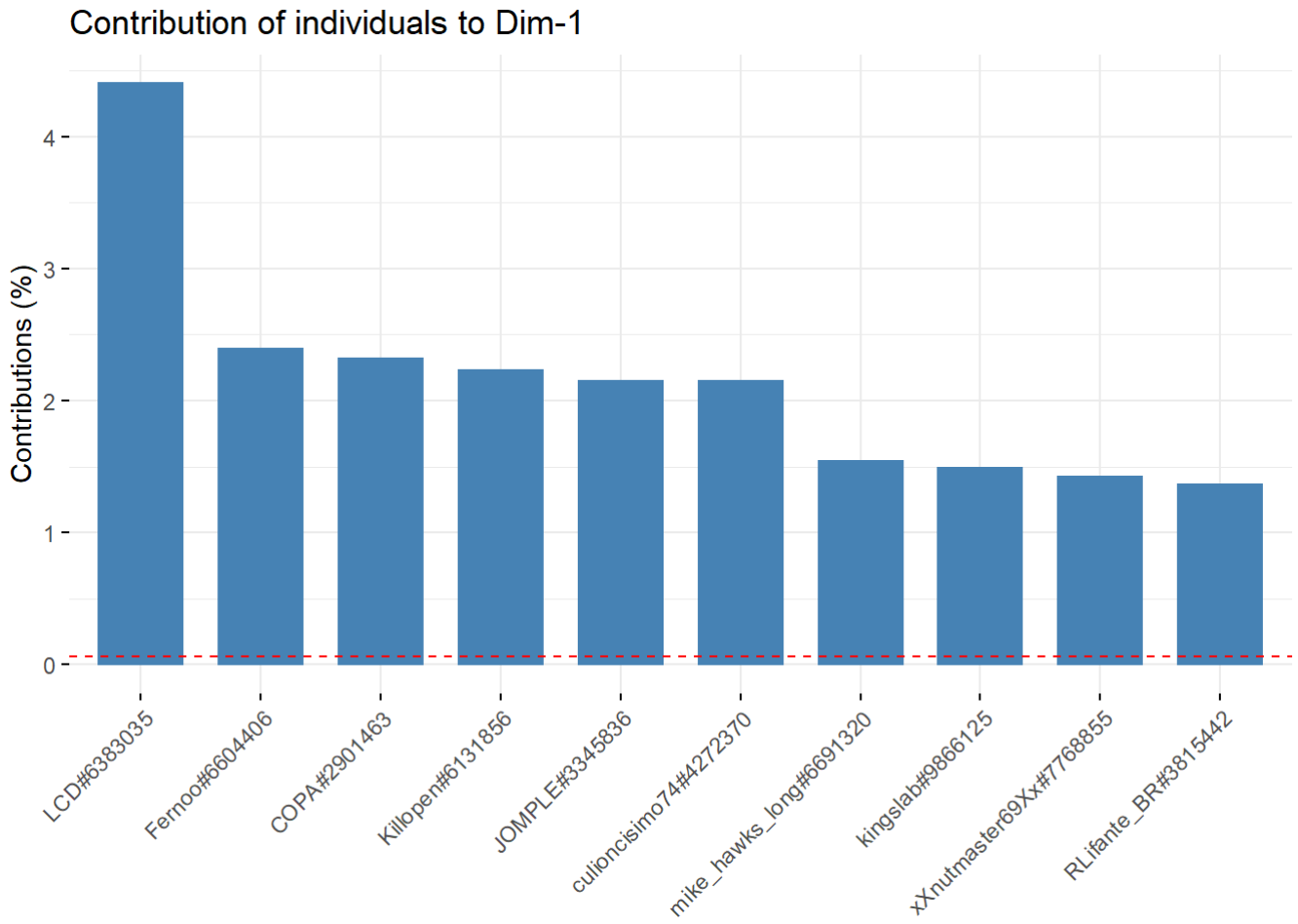
```
fviz_contrib(pca, choice = "ind", axes = 1)
```



This plot does not allow us to distinguish the names of the player so

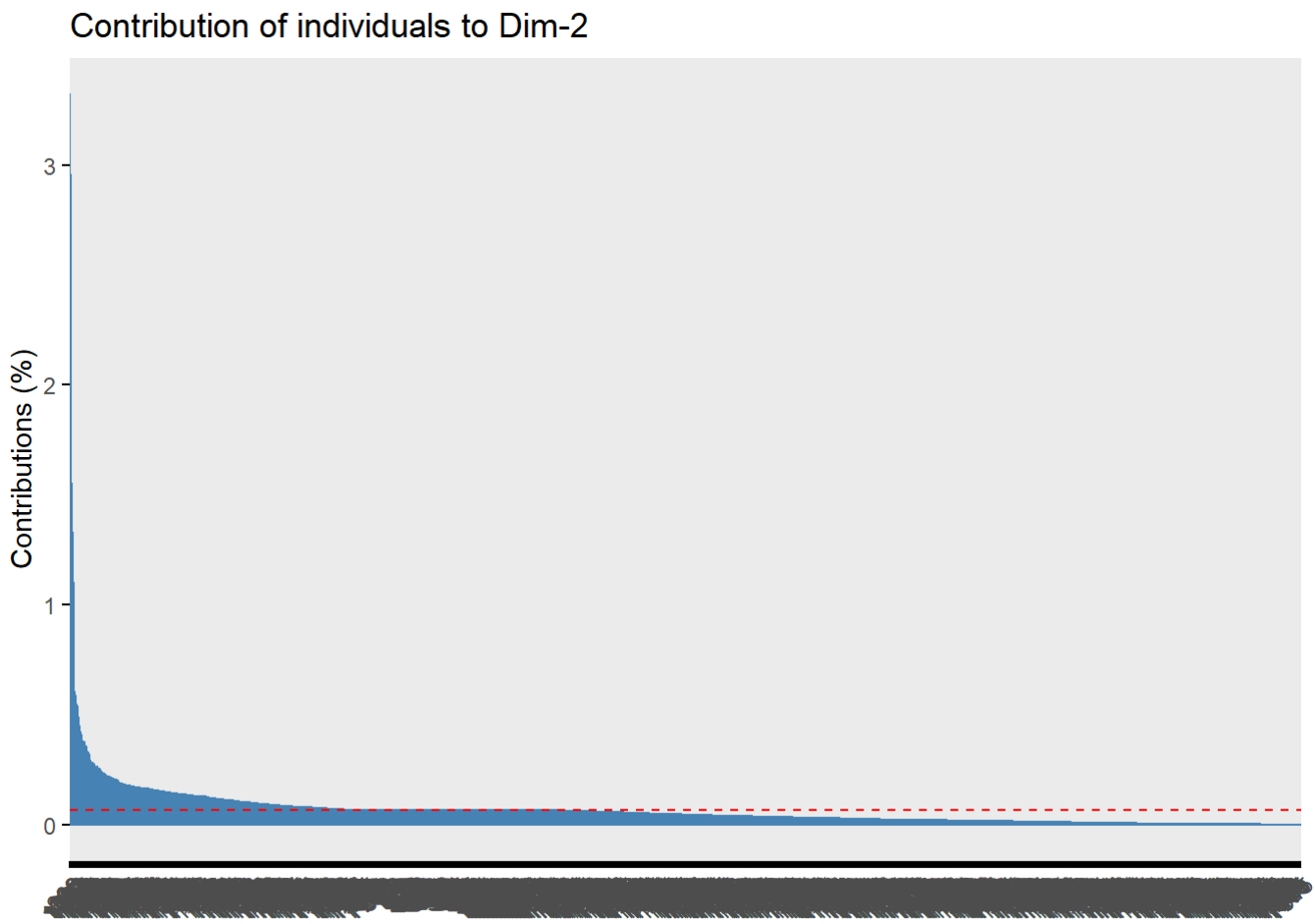
now, let´s see the contribution of the 10 first players

```
names_z1 = names[order(get_pca_ind(pca)$contrib[,1],decreasing=T)]
fviz_contrib(pca, choice = "ind", axes = 1, top=10)+scale_x_discrete(labels=names_z1)
```



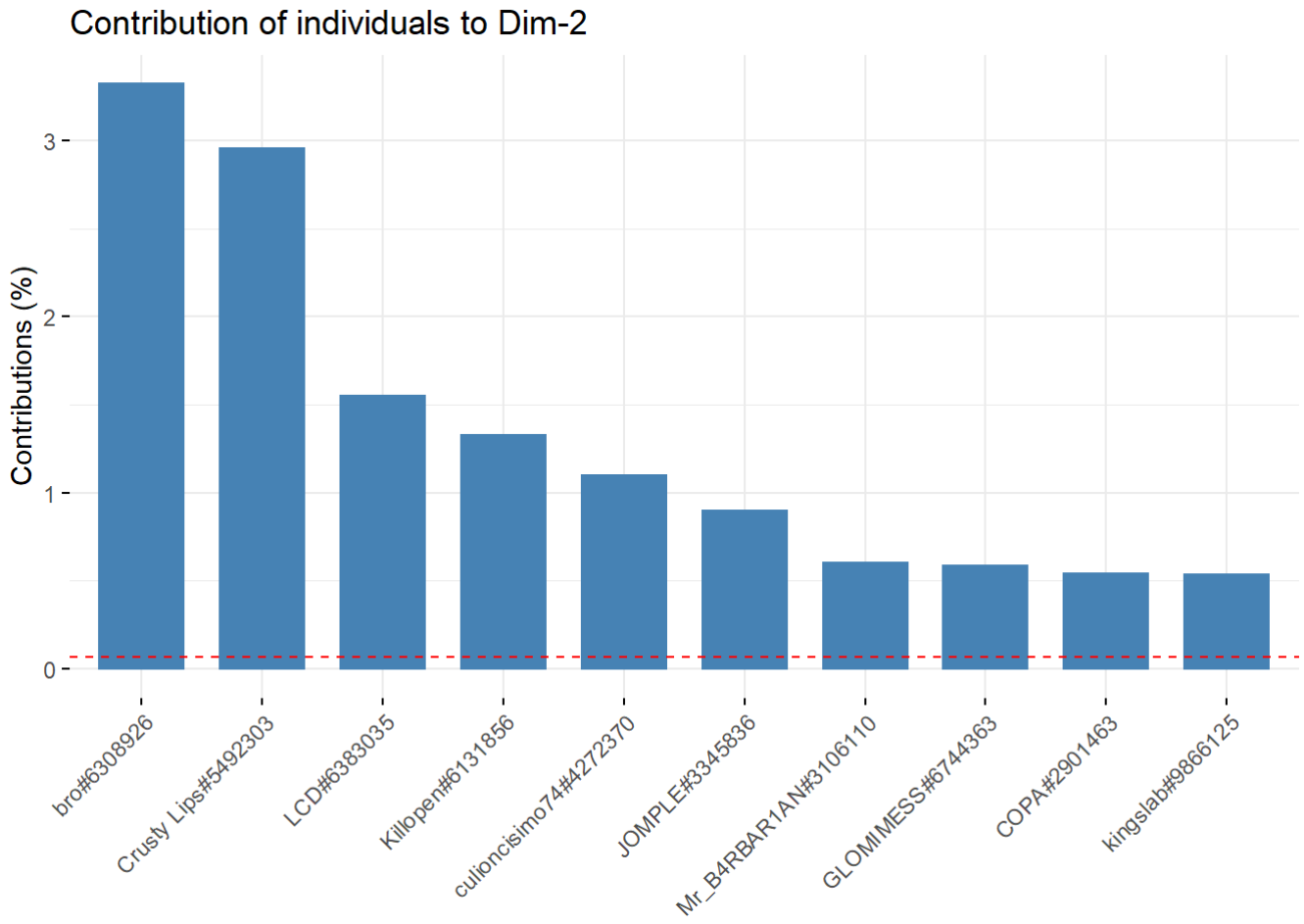
We will repeat this process with the two remaining components **second component** We repeat the process with the second component. We plot the contribution of all players to the second component.

```
fviz_contrib(pca, choice = "ind", axes = 2)
```



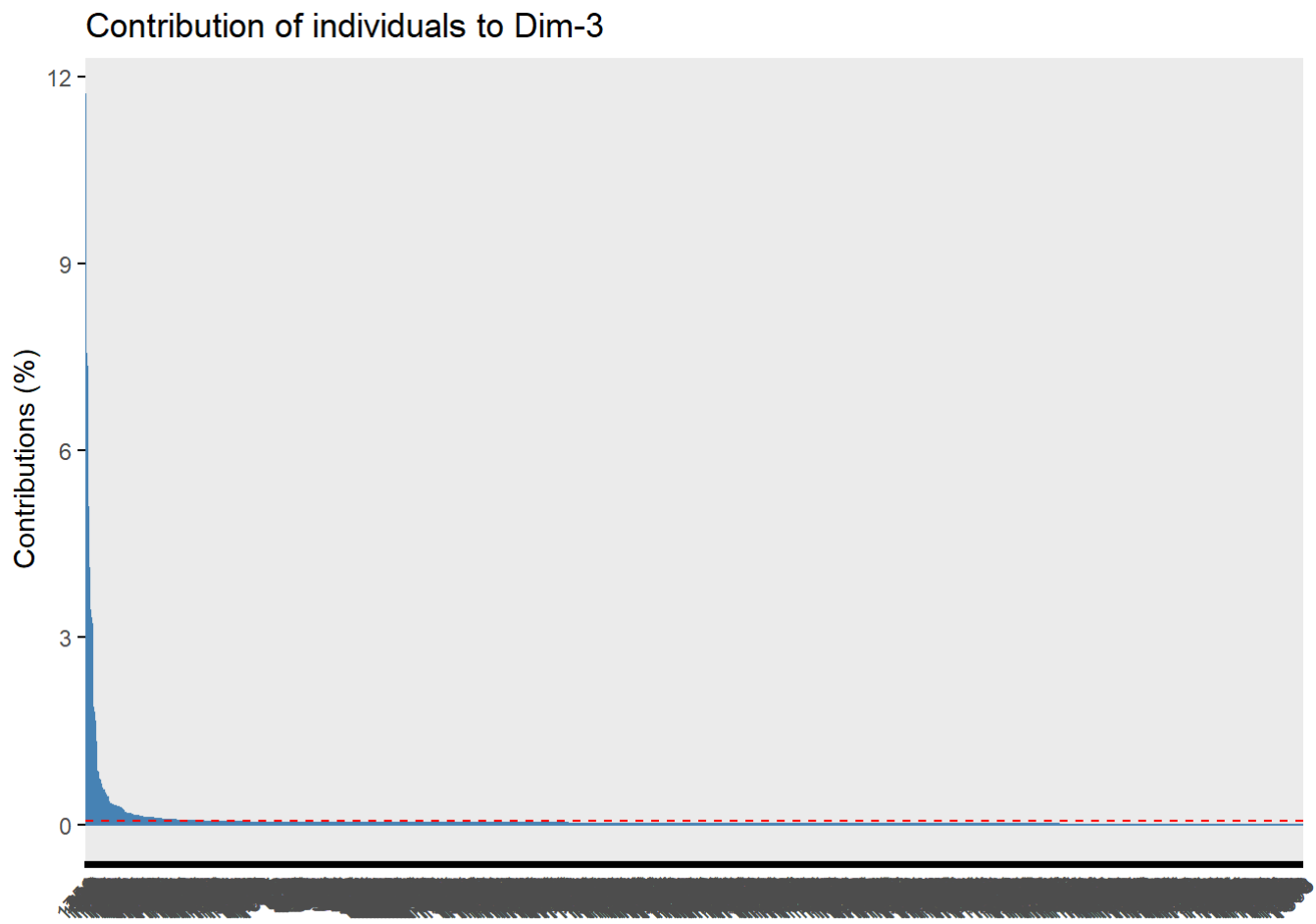
Now, let´s see the contribution of the 10 first players to the second component.

```
names_z1 = names[order(get_pca_ind(pca)$contrib[,2],decreasing=T)]
fviz_contrib(pca, choice = "ind", axes = 2, top=10)+scale_x_discrete(labels=names_z1)
```



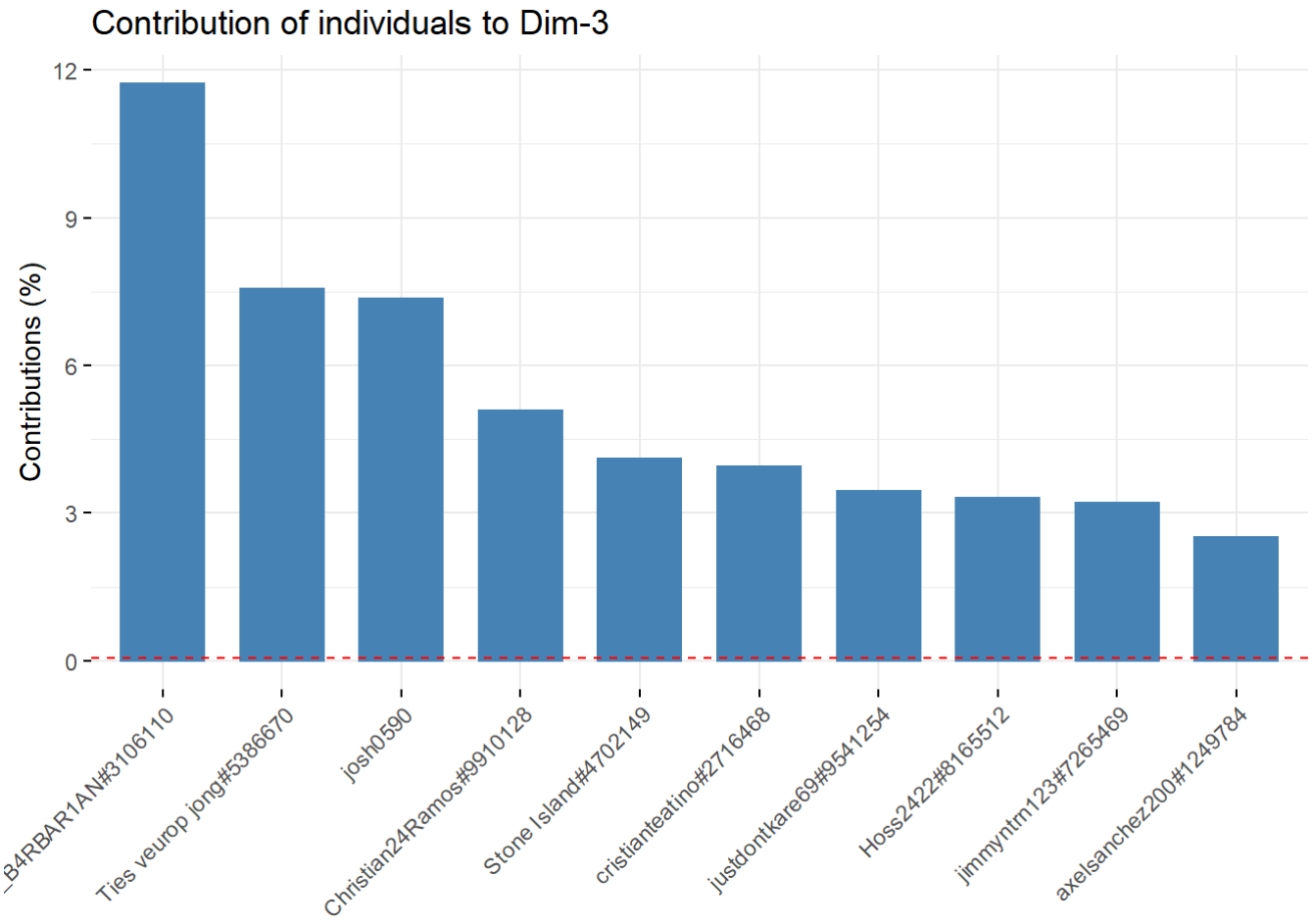
Third component We repeat the process with the third component. We plot the contribution of all players to the third component.

```
fviz_contrib(pca, choice = "ind", axes = 3)
```



now, let´s see the contribution of the 10 first players

```
names_z1 = names[order(get_pca_ind(pca)$contrib[,3],decreasing=T)]
fviz_contrib(pca, choice = "ind", axes = 3, top=10)+scale_x_discrete(labels=names_z1)
```



##Factor analysis

Let´s continue with an analytical tool to reduce the dimension. This tool find the relationship between latent variables and indicators.

```
library(tidyverse)
library(VIM)
library(Quandl)
library(VIM)
library(lubridate)
library(GGally)
library(factoextra)
library(quantmod)
```

First of all, we are going to prepare the data

```
data_FA=data
data_FA$name=NULL
data_FA$level=NULL
```

Let´s see with happens with a *5*-factor model

```
factor=factanal(data_FA, factors=5, rotation="none", scores="regression")
factor
```

```
##
## Call:
## factanal(x = data_FA, factors = 5, scores = "regression", rotation = "none")
##
## Uniquenesses:
##           wins          kills        kdRatio     killstreak         losses
##          0.051          0.005          0.477          0.488          0.352
##           hits     timePlayed      headshots    averageTime    gamesPlayed
##          0.034          0.005          0.035          0.926          0.041
##        assists         misses             xp scorePerMinute          shots
##          0.049          0.005          0.031          0.447          0.005
##         deaths
##          0.022
##
## Loadings:
##                Factor1 Factor2 Factor3 Factor4 Factor5
## wins            0.931           0.190   0.192
## kills           0.983                           0.168
## kdRatio         0.407   0.581   0.124
## killstreak      0.507   0.487           0.105
## losses          0.642   0.437   0.121   0.175
## hits            0.964          -0.187
## timePlayed      0.919           0.381
## headshots       0.958                           0.212
## averageTime                     0.219  -0.128
## gamesPlayed     0.941                   0.260
## assists         0.960          -0.142
## misses          0.977          -0.179
## xp              0.880           0.354  -0.221   0.140
## scorePerMinute  0.361   0.545  -0.226   0.270
## shots           0.979          -0.181
## deaths          0.982
##
##                Factor1 Factor2 Factor3 Factor4 Factor5
## SS loadings     10.951   1.071   0.561   0.293   0.155
## Proportion Var   0.684   0.067   0.035   0.018   0.010
## Cumulative Var   0.684   0.751   0.786   0.805   0.814
##
## Test of the hypothesis that 5 factors are sufficient.
## The chi square statistic is 24682.22 on 50 degrees of freedom.
## The p-value is 0
```

```
cbind(factor$loadings, factor$uniquenesses)
```

```
##                    Factor1       Factor2      Factor3       Factor4
## wins            0.93055724 -0.0188440686  0.189759771  1.922672e-01
## kills           0.98261595 -0.0099593817 -0.021876216  2.822642e-02
## kdRatio         0.40671211  0.5809467663  0.123921562  2.409497e-03
## killstreak      0.50721988  0.4867915830  0.063759132  1.046858e-01
## losses          0.64153099  0.4369142652  0.121256027  1.751238e-01
## hits            0.96391194  0.0149289405 -0.186688890 -2.363107e-02
## timePlayed      0.91876726 -0.0034105744  0.380569333  8.504002e-05
## headshots       0.95829987  0.0318596672  0.003280960 -1.726749e-02
## averageTime    -0.06540128 -0.0666709786  0.219084572 -1.282169e-01
## gamesPlayed     0.94082055 -0.0227852851 -0.004214859  2.595046e-01
## assists         0.95962778 -0.0069339998 -0.141838433 -3.975168e-02
## misses          0.97713660  0.0006446079 -0.179061693 -2.628661e-02
## xp              0.87995373  0.0334202838  0.354258774 -2.211447e-01
## scorePerMinute  0.36103397  0.5445148091 -0.226473208  2.695138e-01
## shots           0.97942543  0.0033740305 -0.181401894 -2.590880e-02
## deaths          0.98194054 -0.0379033136 -0.017321336  6.359820e-02
##                    Factor5
## wins           -0.097658733 0.05119000
## kills           0.168110447 0.00500000
## kdRatio         0.070228618 0.47675450
## killstreak      0.048585050 0.48845172
## losses         -0.009922259 0.35174904
## hits            0.041132113 0.03354809
## timePlayed     -0.077737172 0.00500000
## headshots       0.212020482 0.03538423
## averageTime    -0.019730465 0.92611825
## gamesPlayed    -0.075032642 0.04134738
## assists         0.089375815 0.04938059
## misses         -0.094574224 0.00500000
## xp              0.139517505 0.03069591
## scorePerMinute -0.043065316 0.44710312
## shots          -0.069142690 0.00500000
## deaths          0.090035733 0.02190153
```

The uniqueness is very high for some variables. And a high uniqueness for a variable indicates that the factors do not account well for its variance. But it is low for most of the variables

Let´s prove with three factors rotation varimax , and Barlett estimation for scores

```
factor= factanal(data_FA, factors =5, rotation="varimax", scores="Bartlett")
factor
```

```
##
## Call:
## factanal(x = data_FA, factors = 5, scores = "Bartlett", rotation = "varimax")
##
## Uniquenesses:
##           wins          kills        kdRatio     killstreak         losses
##          0.051          0.005          0.477          0.488          0.352
##           hits     timePlayed      headshots    averageTime    gamesPlayed
##          0.034          0.005          0.035          0.926          0.041
##        assists         misses             xp scorePerMinute          shots
##          0.049          0.005          0.031          0.447          0.005
##         deaths
##          0.022
##
## Loadings:
##                Factor1 Factor2 Factor3 Factor4 Factor5
## wins            0.852   0.334           0.334
## kills           0.936   0.308   0.101          -0.119
## kdRatio         0.202   0.692
## killstreak      0.316   0.633
## losses          0.446   0.647           0.155
## hits            0.915   0.279   0.214
## timePlayed      0.850   0.371  -0.273   0.247
## headshots       0.906   0.341                  -0.143
## averageTime                    -0.265
## gamesPlayed     0.864   0.298   0.183   0.301
## assists         0.921   0.268   0.166
## misses          0.925   0.266   0.198           0.173
## xp              0.836   0.375  -0.345
## scorePerMinute  0.149   0.586   0.421
## shots           0.928   0.270   0.202           0.152
## deaths          0.936   0.283   0.107
##
##                Factor1 Factor2 Factor3 Factor4 Factor5
## SS loadings      9.233   2.706   0.670   0.313   0.109
## Proportion Var   0.577   0.169   0.042   0.020   0.007
## Cumulative Var   0.577   0.746   0.788   0.808   0.814
##
## Test of the hypothesis that 5 factors are sufficient.
## The chi square statistic is 24682.22 on 50 degrees of freedom.
## The p-value is 0
```

```
cbind(factor$loadings, factor$uniquenesses)
```

```
##                    Factor1     Factor2     Factor3      Factor4      Factor5
## wins            0.85166888  0.33403485 -0.01779433  0.333980257  0.005091422
## kills           0.93604107  0.30763473  0.10107791  0.005605162 -0.118783572
## kdRatio         0.20172312  0.69238306  0.00321603 -0.054889272 -0.009930423
## killstreak      0.31630187  0.63288163  0.09714341  0.033992342 -0.021084218
## losses          0.44615482  0.64657680  0.08096203  0.155482639 -0.008973988
## hits            0.91530220  0.27864419  0.21368755 -0.049040501  0.054443307
## timePlayed      0.84969425  0.37055725 -0.27299038  0.246981526  0.014296465
## headshots       0.90614349  0.34148401  0.06355960 -0.048481892 -0.143239723
## averageTime    -0.03659227 -0.04545603 -0.26481891  0.002594885  0.002867073
## gamesPlayed     0.86374721  0.29751994  0.18274772  0.301072310  0.005843583
## assists         0.92060698  0.26758337  0.16630462 -0.061869846  0.004094351
## misses          0.92521608  0.26604735  0.19836696  0.015786634  0.173464844
## xp              0.83646849  0.37500778 -0.34535175 -0.042078451 -0.089196872
## scorePerMinute  0.14923208  0.58602090  0.42111140  0.068992857  0.069561415
## shots           0.92788095  0.26976130  0.20227030  0.003492435  0.151606382
## deaths          0.93628057  0.28293996  0.10686528  0.074679925 -0.066494145
##
## wins           0.05119000
## kills          0.00500000
## kdRatio        0.47675450
## killstreak     0.48845172
## losses         0.35174904
## hits           0.03354809
## timePlayed     0.00500000
## headshots      0.03538423
## averageTime    0.92611825
## gamesPlayed    0.04134738
## assists        0.04938059
## misses         0.00500000
## xp             0.03069591
## scorePerMinute 0.44710312
## shots          0.00500000
## deaths         0.02190153
```

We have obtained the same results as beefore.

Now, let´s try it with *7* factors

```
factor=factanal(data_FA, factors = 7, rotation="none", scores="regression")
factor
```

```
##
## Call:
## factanal(x = data_FA, factors = 7, scores = "regression", rotation = "none")
##
## Uniquenesses:
##           wins          kills        kdRatio     killstreak         losses
##          0.049          0.005          0.480          0.493          0.356
##           hits     timePlayed      headshots    averageTime    gamesPlayed
##          0.005          0.005          0.035          0.918          0.041
##        assists         misses             xp scorePerMinute          shots
##          0.034          0.005          0.024          0.408          0.005
##         deaths
##          0.005
##
## Loadings:
##                Factor1 Factor2 Factor3 Factor4 Factor5 Factor6 Factor7
## wins            0.923           0.218   0.128  -0.139          -0.101
## kills           0.985                                           0.147
## kdRatio         0.402   0.537   0.128   0.118   0.144   0.136
## killstreak      0.505   0.449           0.151           0.147
## losses          0.637   0.395   0.135   0.208           0.136
## hits            0.972          -0.177           0.103
## timePlayed      0.907           0.405
## headshots       0.961                                           0.184
## averageTime            -0.122   0.211
## gamesPlayed     0.939                   0.166  -0.186
## assists         0.967          -0.125
## misses          0.979          -0.146                          -0.118
## xp              0.871           0.378  -0.143   0.206           0.104
## scorePerMinute  0.366   0.540  -0.211   0.197  -0.170   0.227
## shots           0.982          -0.152
## deaths          0.985                          -0.109
##
##                Factor1 Factor2 Factor3 Factor4 Factor5 Factor6 Factor7
## SS loadings     10.945   0.960   0.577   0.197   0.187   0.138   0.132
## Proportion Var   0.684   0.060   0.036   0.012   0.012   0.009   0.008
## Cumulative Var   0.684   0.744   0.780   0.792   0.804   0.813   0.821
##
## Test of the hypothesis that 7 factors are sufficient.
## The chi square statistic is 23702.19 on 29 degrees of freedom.
## The p-value is 0
```

```
cbind(factor$loadings, factor$uniquenesses)
```

```
##                    Factor1     Factor2      Factor3      Factor4      Factor5
## wins            0.92328711  0.01034672  0.217929780  0.128179515 -0.138558104
## kills           0.98536067  0.02175436  0.004950627 -0.006300911 -0.010815429
## kdRatio         0.40175125  0.53695391  0.128377540  0.117644000  0.144059728
## killstreak      0.50486068  0.44850060  0.073543625  0.151493210  0.021679541
## losses          0.63734683  0.39532927  0.134865989  0.207883076 -0.022990262
## hits            0.97230191 -0.02634175 -0.177345891  0.081130461  0.103065833
## timePlayed      0.90654154 -0.01292741  0.405395214  0.023721961  0.014018620
## headshots       0.96132762  0.04514893  0.024448473  0.001329191  0.058377229
## averageTime    -0.07061141 -0.12196039  0.211174034 -0.042541812  0.093560792
## gamesPlayed     0.93928871  0.01573041  0.024227634  0.166260663 -0.186433271
## assists         0.96701836 -0.04353466 -0.125024318  0.016273528  0.065172849
## misses          0.97871794  0.01525422 -0.145663450 -0.057380545 -0.018715290
## xp              0.87069472  0.03465202  0.377771434 -0.143144629  0.206023048
## scorePerMinute  0.36599912  0.54034698 -0.211471692  0.197068442 -0.170416743
## shots           0.98231397  0.00739165 -0.152429830 -0.031229452  0.004433039
## deaths          0.98517293 -0.02626932  0.014556748 -0.019658942 -0.109260851
##                    Factor6     Factor7
## wins           -0.070076608 -0.10088439 0.04922425
## kills          -0.045018705  0.14705357 0.00500000
## kdRatio         0.135784357  0.02595811 0.48007323
## killstreak      0.146735075  0.02268155 0.49314470
## losses          0.135764521 -0.02772833 0.35619006
## hits            0.013211553  0.01416139 0.00500000
## timePlayed      0.007641209 -0.08899199 0.00500000
## headshots      -0.028700737  0.18355229 0.03528688
## averageTime     0.082632788 -0.01606069 0.91763354
## gamesPlayed    -0.088363220 -0.07838066 0.04054804
## assists         0.058350123  0.07187227 0.03426869
## misses         -0.008272401 -0.11788400 0.00500000
## xp             -0.001760850  0.10414877 0.02419296
## scorePerMinute  0.226523804 -0.04478371 0.40810012
## shots          -0.004213937 -0.09326605 0.00500000
## deaths          0.046798903  0.09528232 0.00500000
```

We obtain similar result of the uniqueness as before, (as for 5 factors)

Let´s prove with three factors rotation varimax , and Barlett estimation for scores

```
factor=factanal(data_FA, factors = 7, rotation="varimax", scores="Bartlett")
factor
```

```
##
## Call:
## factanal(x = data_FA, factors = 7, scores = "Bartlett", rotation = "varimax")
##
## Uniquenesses:
##           wins          kills        kdRatio     killstreak         losses
##          0.049          0.005          0.480          0.493          0.356
##           hits     timePlayed      headshots    averageTime    gamesPlayed
##          0.005          0.005          0.035          0.918          0.041
##        assists         misses             xp scorePerMinute          shots
##          0.034          0.005          0.024          0.408          0.005
##         deaths
##          0.005
##
## Loadings:
##                Factor1 Factor2 Factor3 Factor4 Factor5 Factor6 Factor7
## wins            0.848   0.333           0.348
## kills           0.934   0.308   0.102
## kdRatio         0.199   0.683                   0.113
## killstreak      0.318   0.630
## losses          0.449   0.646           0.145
## hits            0.926   0.283   0.171                           -0.148
## timePlayed      0.843   0.362  -0.268   0.271
## headshots       0.905   0.338                   0.111  -0.117
## averageTime                    -0.279
## gamesPlayed     0.864   0.302   0.168   0.299
## assists         0.930   0.273   0.117
## misses          0.925   0.271   0.189                   0.170
## xp              0.825   0.360  -0.309           0.252
## scorePerMinute  0.150   0.625   0.369          -0.198
## shots           0.930   0.275   0.186                   0.141
## deaths          0.939   0.288                                   0.115
##
##                Factor1 Factor2 Factor3 Factor4 Factor5 Factor6 Factor7
## SS loadings      9.240   2.732   0.552   0.324   0.154   0.081   0.053
## Proportion Var   0.578   0.171   0.035   0.020   0.010   0.005   0.003
## Cumulative Var   0.578   0.748   0.783   0.803   0.813   0.818   0.821
##
## Test of the hypothesis that 7 factors are sufficient.
## The chi square statistic is 23702.19 on 29 degrees of freedom.
## The p-value is 0
```

```
cbind(factor$loadings, factor$uniquenesses)
```

```
##                    Factor1     Factor2     Factor3       Factor4       Factor5
## wins            0.84786622  0.33256308 -0.01690716  3.475935e-01  0.0003384932
## kills           0.93388579  0.30809504  0.10187497  2.021466e-02  0.0784791569
## kdRatio         0.19934911  0.68267902  0.01318491 -2.347829e-02  0.1125239802
## killstreak      0.31794104  0.63042004  0.08056337  3.659799e-02  0.0096441039
## losses          0.44948644  0.64561398  0.04885699  1.447323e-01 -0.0266185922
## hits            0.92613519  0.28334335  0.17068632 -6.328945e-02 -0.0434742259
## timePlayed      0.84325211  0.36165756 -0.26758182  2.714277e-01  0.0710134107
## headshots       0.90499497  0.33842098  0.06395461 -2.849362e-02  0.1105920534
## averageTime    -0.03538484 -0.05349263 -0.27925617  5.389182e-05 -0.0015260186
## gamesPlayed     0.86400474  0.30230197  0.16831410  2.991124e-01 -0.0563531526
## assists         0.92987823  0.27270947  0.11734458 -8.484410e-02 -0.0499835668
## misses          0.92522395  0.27127274  0.18867744  2.283919e-02 -0.0475448050
## xp              0.82519197  0.35973277 -0.30893684  1.039586e-02  0.2515303536
## scorePerMinute  0.15032898  0.62502861  0.36866334  1.491979e-02 -0.1981626703
## shots           0.92995475  0.27491204  0.18617589  6.514563e-03 -0.0470025020
## deaths          0.93903180  0.28788995  0.07882593  5.815136e-02 -0.0481382938
##                     Factor6       Factor7
## wins            0.0019986975  0.0137736364 0.04922425
## kills          -0.0941021674  0.0487336229 0.00500000
## kdRatio         0.0166797803 -0.0212896985 0.48007323
## killstreak     -0.0206047987  0.0069443770 0.49314470
## losses         -0.0233935171 -0.0135687058 0.35619006
## hits            0.0128497323 -0.1479592322 0.00500000
## timePlayed      0.0361019148  0.0397187655 0.00500000
## headshots      -0.1168030727  0.0197858748 0.03528688
## averageTime     0.0002045139  0.0020984086 0.91763354
## gamesPlayed    -0.0160648972 -0.0180826193 0.04054804
## assists        -0.0235045093 -0.0516765296 0.03426869
## misses          0.1701097178  0.0005433855 0.00500000
## xp             -0.0079097797  0.0811029333 0.02419296
## scorePerMinute  0.0267712909  0.0496685540 0.40810012
## shots           0.1409371694 -0.0277950501 0.00500000
## deaths         -0.0718645665  0.1154238929 0.00500000
```

We again obtain similar results.

Let´s prove with *10* factors

```
factor=factanal(data_FA, factors = 10, rotation="none", scores="regression")
factor
```

```
##
## Call:
## factanal(x = data_FA, factors = 10, scores = "regression", rotation = "none")
##
## Uniquenesses:
##           wins          kills        kdRatio     killstreak         losses
##          0.044          0.005          0.360          0.467          0.289
##           hits     timePlayed      headshots    averageTime    gamesPlayed
##          0.005          0.005          0.030          0.615          0.005
##        assists         misses             xp scorePerMinute          shots
##          0.030          0.005          0.005          0.396          0.005
##         deaths
##          0.005
##
## Loadings:
##                Factor1 Factor2 Factor3 Factor4 Factor5 Factor6 Factor7 Factor8
## wins            0.932           0.105           0.241
## kills           0.985                                           0.134
## kdRatio         0.407   0.589   0.138   0.191           0.195
## killstreak      0.506   0.503
## losses          0.646   0.407                   0.157   0.184
## hits            0.964          -0.190          -0.115   0.103
## timePlayed      0.921           0.335           0.146
## headshots       0.961                          -0.119           0.144
## averageTime                     0.212   0.535                           0.182
## gamesPlayed     0.945          -0.102           0.286
## assists         0.961          -0.131          -0.126
## misses          0.972          -0.182                          -0.118
## xp              0.886           0.427          -0.149
## scorePerMinute  0.360   0.575  -0.266  -0.131                          -0.121
## shots           0.975          -0.184
## deaths          0.984                                  -0.107   0.110
##                Factor9 Factor10
## wins
## kills
## kdRatio         0.162
## killstreak
## losses         -0.158   0.173
## hits
## timePlayed
## headshots
## averageTime
## gamesPlayed
## assists
## misses
## xp
## scorePerMinute          0.133
## shots
## deaths
##
##                Factor1 Factor2 Factor3 Factor4 Factor5 Factor6 Factor7 Factor8
## SS loadings     10.980   1.102   0.580   0.350   0.271   0.121   0.093   0.085
## Proportion Var   0.686   0.069   0.036   0.022   0.017   0.008   0.006   0.005
## Cumulative Var   0.686   0.755   0.791   0.813   0.830   0.838   0.844   0.849
##                Factor9 Factor10
```

```
## SS loadings      0.081    0.070
## Proportion Var   0.005    0.004
## Cumulative Var   0.854    0.858
##
## Test of the hypothesis that 10 factors are sufficient.
## The chi square statistic is 23268.35 on 5 degrees of freedom.
## The p-value is 0
```

```
cbind(factor$loadings, factor$uniquenesses)
```

```
##                    Factor1      Factor2     Factor3      Factor4      Factor5
## wins            0.93177358  0.029196900  0.10455524 -0.083978385  0.241006277
## kills           0.98525859  0.005090733 -0.01457203  0.006601045 -0.065279216
## kdRatio         0.40702314  0.588811921  0.13797569  0.191127462 -0.042043786
## killstreak      0.50616904  0.502645267  0.04490819 -0.052532386  0.015247467
## losses          0.64648233  0.407445650  0.07433877  0.009428797  0.156763033
## hits            0.96428179 -0.009335638 -0.18958525 -0.008186802 -0.115057830
## timePlayed      0.92094302  0.009184487  0.33529343  0.005480798  0.146412777
## headshots       0.96146464  0.036438532  0.02364437  0.008830722 -0.119382834
## averageTime    -0.06650415 -0.045420257  0.21177139  0.534530014 -0.007885207
## gamesPlayed     0.94525680 -0.021891489 -0.10221340  0.004726945  0.285953908
## assists         0.96115326 -0.026782472 -0.13094511  0.023764547 -0.125777596
## misses          0.97160779  0.004951191 -0.18154916  0.003469399 -0.035608684
## xp              0.88573045 -0.015218662  0.42739456 -0.005397977 -0.148635927
## scorePerMinute  0.35965432  0.575258355 -0.26621347 -0.131259526  0.098408680
## shots           0.97499510  0.002248722 -0.18397957  0.001261996 -0.050945955
## deaths          0.98404719  0.004961091 -0.03260548 -0.004293293 -0.002540399
##                     Factor6      Factor7     Factor8       Factor9     Factor10
## wins           -0.022359864 -0.023224280  0.05581067  0.0397729228 -0.074606465
## kills          -0.004905968  0.133606156 -0.01094855  0.0001632652 -0.042608690
## kdRatio         0.194993520 -0.012202000 -0.06537343  0.1622945975 -0.039742880
## killstreak      0.095288858  0.024107681  0.01339806 -0.0877290568 -0.035634270
## losses          0.183574383  0.027791353 -0.08564082 -0.1579891891  0.172959600
## hits            0.102624984 -0.015932799  0.07095481  0.0075791682  0.007606199
## timePlayed     -0.017150344 -0.085634618  0.07190111 -0.0067373674 -0.007870482
## headshots       0.034806749  0.143918419  0.01618941 -0.0568693034 -0.060419145
## averageTime    -0.044774742 -0.063382343  0.18195323  0.0939185779 -0.016571427
## gamesPlayed     0.060733906  0.047580781 -0.05209547  0.0011327835  0.010301086
## assists         0.039407980  0.038405566  0.04662616 -0.0216467477  0.080757495
## misses         -0.052925520 -0.118129473 -0.04277032 -0.0033899202 -0.008215433
## xp              0.034075125 -0.006637883 -0.06326049  0.0058517397  0.015086563
## scorePerMinute  0.063284451  0.041164497 -0.12074393  0.0877306558  0.132900488
## shots          -0.023499918 -0.099208660 -0.02127939 -0.0013128103 -0.005236196
## deaths         -0.106771175  0.109594464  0.03003890  0.0064024143  0.034826250
##
## wins           0.04357656
## kills          0.00500000
## kdRatio        0.35989144
## killstreak     0.46728423
## losses         0.28925157
## hits           0.00500000
## timePlayed     0.00500000
## headshots      0.03029571
## averageTime    0.61466323
## gamesPlayed    0.00500000
## assists        0.02974419
## misses         0.00500000
## xp             0.00500000
## scorePerMinute 0.39640327
## shots          0.00500000
## deaths         0.00500000
```

Now we obtain lower values for the uniqueness

Let´s prove with three factors rotation varimax , and Barlett estimation for scores

```
factor=factanal(data_FA, factors = 10, rotation="varimax", scores="Bartlett")
factor
```

```
##
## Call:
## factanal(x = data_FA, factors = 10, scores = "Bartlett", rotation = "varimax")
##
## Uniquenesses:
##           wins          kills        kdRatio     killstreak         losses
##          0.044          0.005          0.360          0.467          0.289
##           hits     timePlayed      headshots    averageTime    gamesPlayed
##          0.005          0.005          0.030          0.615          0.005
##        assists         misses             xp scorePerMinute          shots
##          0.030          0.005          0.005          0.396          0.005
##         deaths
##          0.005
##
## Loadings:
##                Factor1 Factor2 Factor3 Factor4 Factor5 Factor6 Factor7 Factor8
## wins            0.817   0.289   0.234           0.382
## kills           0.927   0.287   0.184                           0.108
## kdRatio         0.189   0.728   0.159   0.177
## killstreak      0.315   0.613   0.132                                   0.172
## losses          0.437   0.611   0.177           0.115   0.312
## hits            0.943   0.283
## timePlayed      0.779   0.264   0.444   0.111   0.310
## headshots       0.893   0.306   0.220                           0.128
## averageTime                             0.618
## gamesPlayed     0.872   0.289                   0.319   0.168
## assists         0.938   0.258
## misses          0.940   0.277                                  -0.163
## xp              0.748   0.241   0.607
## scorePerMinute  0.198   0.671  -0.153  -0.259
## shots           0.945   0.279                                  -0.134
## deaths          0.931   0.268   0.151
##                Factor9 Factor10
## wins
## kills
## kdRatio
## killstreak
## losses
## hits                   -0.114
## timePlayed
## headshots
## averageTime
## gamesPlayed
## assists
## misses
## xp
## scorePerMinute  0.116
## shots
## deaths          0.130
##
##                Factor1 Factor2 Factor3 Factor4 Factor5 Factor6 Factor7 Factor8
## SS loadings      9.027   2.576   0.837   0.542   0.379   0.150   0.082   0.062
## Proportion Var   0.564   0.161   0.052   0.034   0.024   0.009   0.005   0.004
## Cumulative Var   0.564   0.725   0.778   0.811   0.835   0.844   0.850   0.853
##                Factor9 Factor10
```

```
## SS loadings      0.048    0.031
## Proportion Var   0.003    0.002
## Cumulative Var   0.856    0.858
##
## Test of the hypothesis that 10 factors are sufficient.
## The chi square statistic is 23268.35 on 5 degrees of freedom.
## The p-value is 0
```

```
cbind(factor$loadings, factor$uniquenesses)
```

```
##                    Factor1     Factor2     Factor3     Factor4      Factor5
## wins            0.81737505  0.28920545  0.23431348 -0.04248133  0.381971964
## kills           0.92679582  0.28733402  0.18427420 -0.04164255  0.035611863
## kdRatio         0.18910483  0.72835839  0.15851582  0.17742451 -0.008786652
## killstreak      0.31463290  0.61332258  0.13180636 -0.07834883  0.054273257
## losses          0.43702520  0.61068898  0.17743168 -0.06792984  0.114896729
## hits            0.94268753  0.28300765  0.03757831 -0.07495889 -0.030221867
## timePlayed      0.77853470  0.26373335  0.44364592  0.11087062  0.309896967
## headshots       0.89290934  0.30609703  0.22012835 -0.03140255 -0.013951388
## averageTime    -0.04212493 -0.04187478  0.01446023  0.61771201 -0.001918051
## gamesPlayed     0.87177099  0.28922181  0.04411787 -0.07948530  0.319113343
## assists         0.93777691  0.25821434  0.08733403 -0.04292151 -0.060350551
## misses          0.93961497  0.27687732  0.04608390 -0.07869589  0.044799141
## xp              0.74839067  0.24058914  0.60655981  0.08303835  0.023257018
## scorePerMinute  0.19777294  0.67126237 -0.15263244 -0.25936595  0.034159601
## shots           0.94482956  0.27941107  0.04468566 -0.07837159  0.030702199
## deaths          0.93051247  0.26833999  0.15093937 -0.03707133  0.089163343
##                     Factor6       Factor7       Factor8       Factor9
## wins            0.023740784 -0.0025177484  0.0348542450  0.0165624536
## kills           0.012364413  0.1078876088  0.0272918929  0.0020560151
## kdRatio        -0.076764262  0.0172709129 -0.0474445055 -0.0929375601
## killstreak      0.038602024  0.0094217113  0.1716589897 -0.0033530501
## losses          0.311815095 -0.0009057663  0.0172932028  0.0071231140
## hits            0.020944233 -0.0062197326  0.0142242091 -0.0683708472
## timePlayed      0.082035504 -0.0441279377  0.0576335979  0.0318833787
## headshots       0.029576488  0.1277198198  0.0939894602 -0.0313164430
## averageTime    -0.005059003  0.0001987828 -0.0026961571  0.0001034506
## gamesPlayed     0.168331294  0.0309311481 -0.0880278282 -0.0361259383
## assists         0.075539694  0.0219083876  0.0007517529  0.0178468691
## misses          0.008164399 -0.1625325006 -0.0077188019  0.0020033585
## xp              0.030581783  0.0187661911 -0.0002467338 -0.0156073920
## scorePerMinute  0.010766499 -0.0288837505 -0.0881477042  0.1162943296
## shots           0.010643831 -0.1335031520 -0.0035710167 -0.0114181448
## deaths          0.032228193  0.0659211279  0.0235293647  0.1295844525
##                    Factor10
## wins           -3.730997e-03 0.04357656
## kills           6.383930e-02 0.00500000
## kdRatio         6.013697e-03 0.35989144
## killstreak     -1.631057e-03 0.46728423
## losses         -3.741484e-03 0.28925157
## hits           -1.142732e-01 0.00500000
## timePlayed     -3.429838e-02 0.00500000
## headshots       4.561173e-02 0.03029571
## averageTime    -4.022084e-05 0.61466323
## gamesPlayed     5.408712e-02 0.00500000
## assists        -6.737538e-02 0.02974419
## misses          2.328429e-02 0.00500000
## xp              1.236889e-02 0.00500000
## scorePerMinute -2.822218e-03 0.39640327
## shots          -2.852177e-03 0.00500000
## deaths          4.828884e-02 0.00500000
```

The values are the same the same as the previous

After this analysis I would chose 5 factors, because although with 10 factors the uniqueness is less,I think that 10 factors are too many if we only have 16 variables. And the input if we choose 7 is similar as if we chose 5, so I think that 5 factors is a better option.

##Clustering

Let´s use the last tool, clustering. Clustering is a tool that classify the observations in a data matrix into homogeneous groups, so that observations of the same group should be similar, and observations in different groups should be different. This tool measures similarity taking into account distances. It is subjective

```
library(factoextra)
library(cluster)
library(mclust)
library(kernlab)
```

There are different types of clustering *Kmeans* We will start with kmeans. Clustering consists on randomly select K centroids, assign each point to the nearest centroid, recalculate centroid based on assigned classes and repeat this process until the centroids do not change. Before we have to scale the data because the variables of our database are very different. And obviously, we have to remove factors and characters.

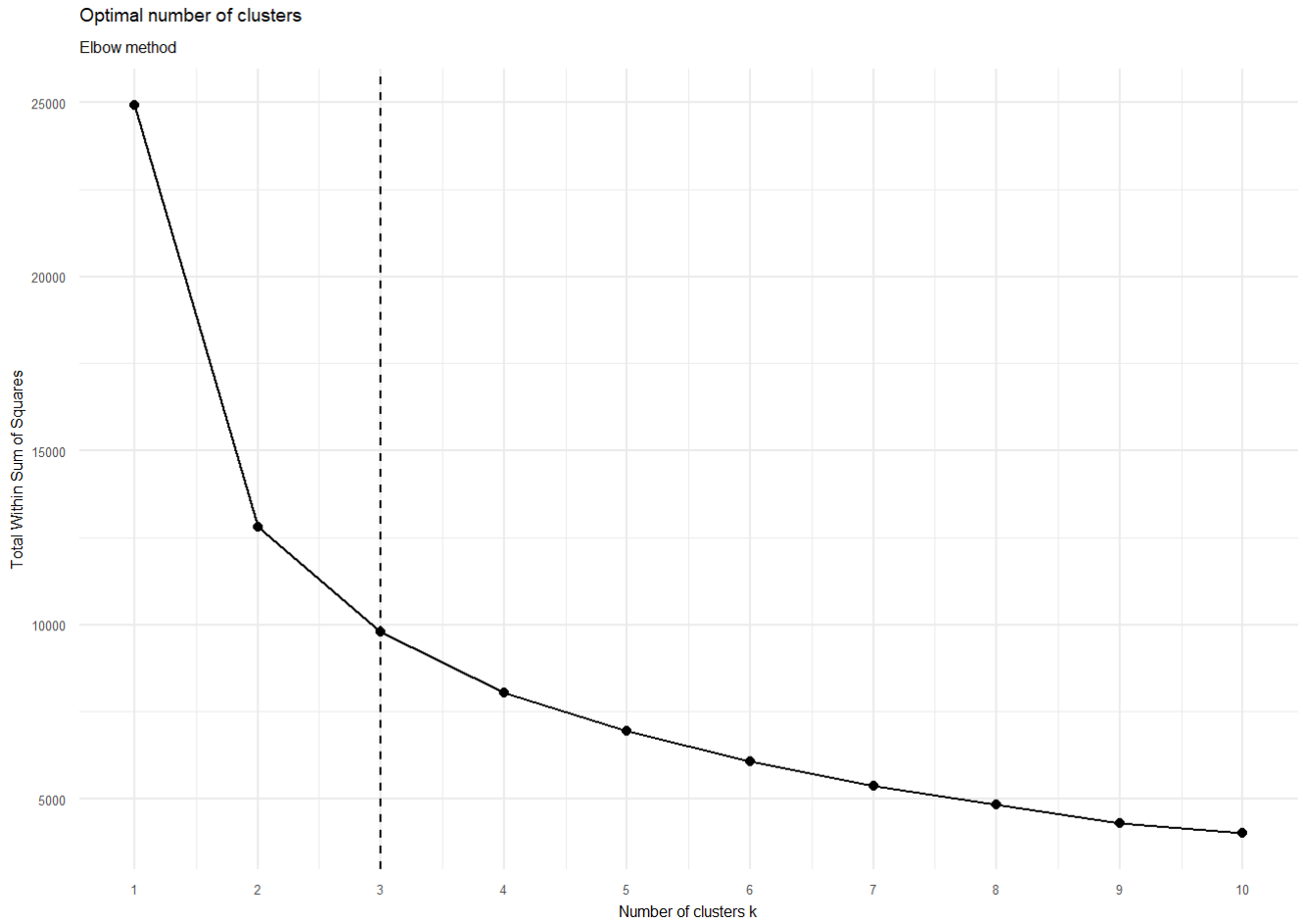
```
data_cl=data
data_cl$name=NULL
data_cl$level=NULL
data_cl=scale(data_cl)
```

We will use the elbow method to compute the optimal number of clusters

```
evaluation = data.frame(clusters = 1:10, WSS = 0)
for (i in 1:10){
km = kmeans(data_cl,
center = evaluation$clusters[i],
nstart = 25)
evaluation$WSS[i] = sum((data_cl - km$centers[km$cluster,])^2)
}
```

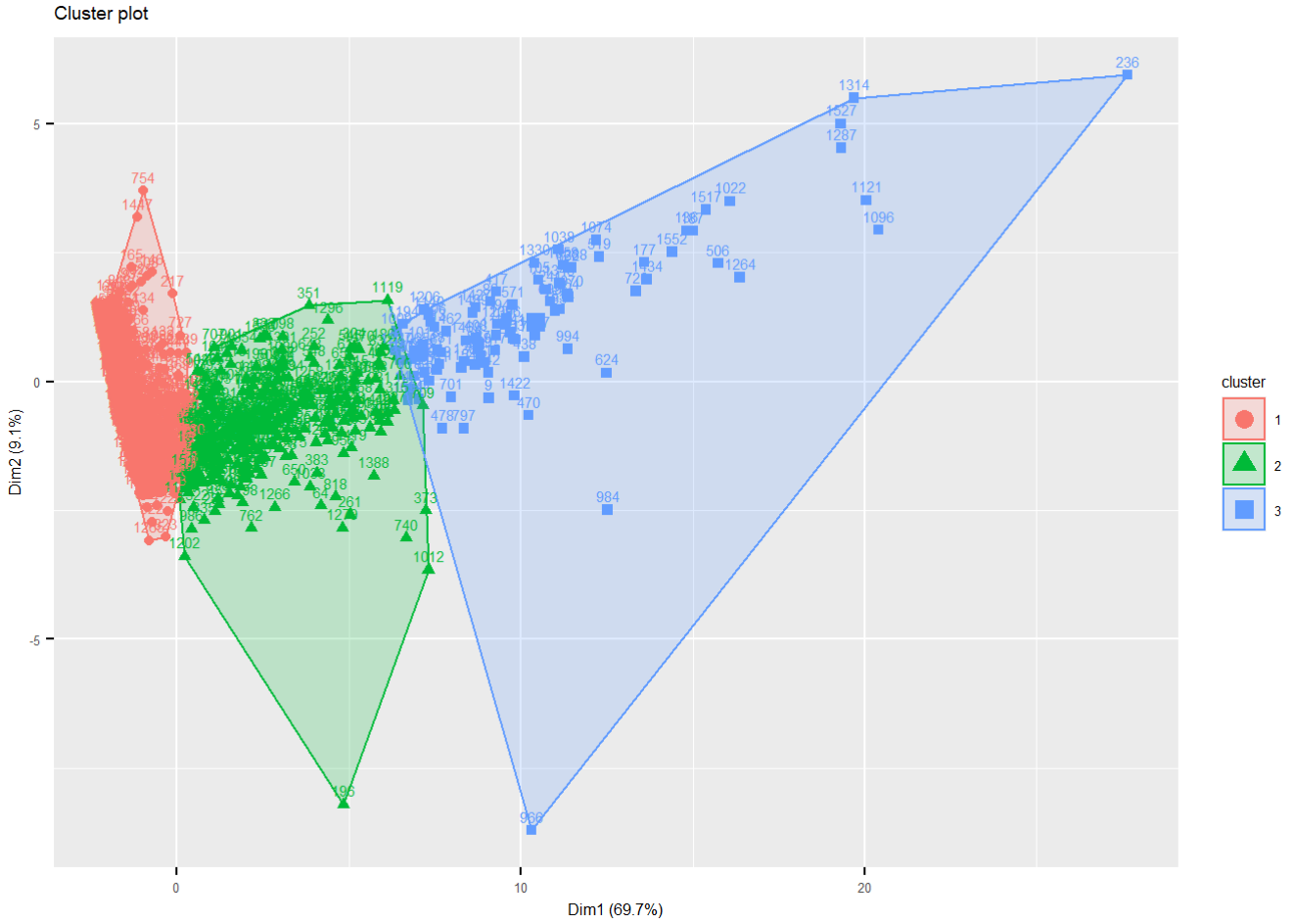
We plot the result

```
ggplot(evaluation) + aes(x = clusters, y = WSS) +
geom_point() + geom_line() +
geom_vline(xintercept = 3, linetype = 2) +
scale_x_continuous(breaks = 1:10) +
theme_minimal() + theme(text = element_text(size = 6))+
labs(title = "Optimal number of clusters", subtitle = "Elbow method") +
xlab("Number of clusters k") + ylab("Total Within Sum of Squares")
```



Following the elbow method I think that the optimal number of cluster is 3

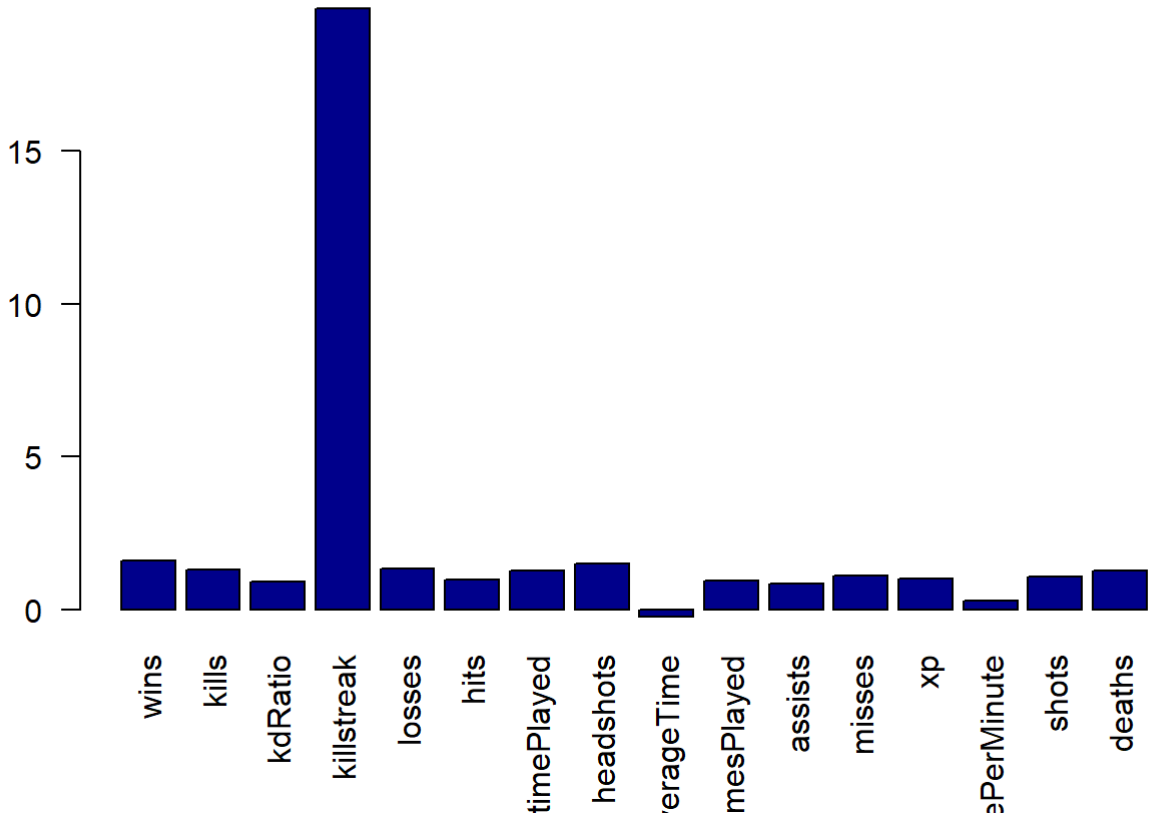
```
fit= kmeans(data_cl, center = 3, nstart = 1000)
fviz_cluster(fit, data = data_cl, labelsize = 6) +
theme(text = element_text(size = 6))
```



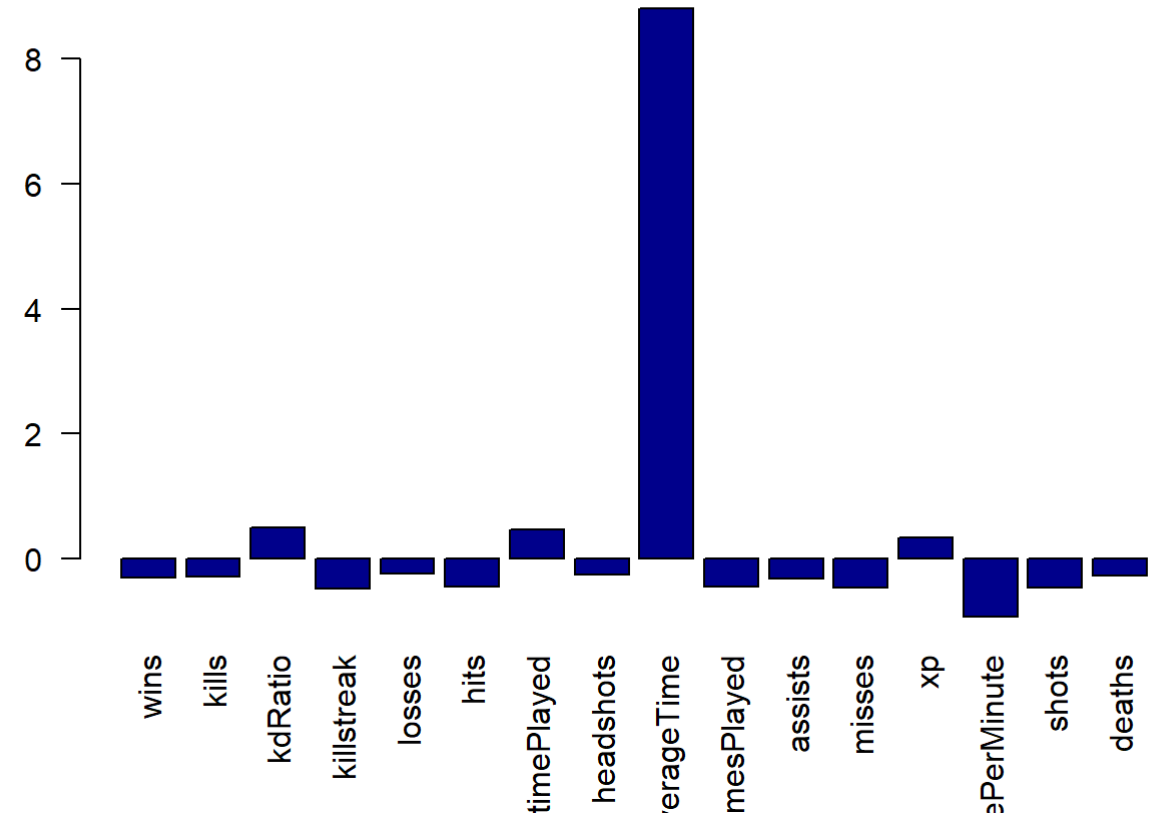
I have removed the name of the player because it is impossible to distinguish them because they are very close

Let´s plot each center in order to understand better the variables that are in each center.

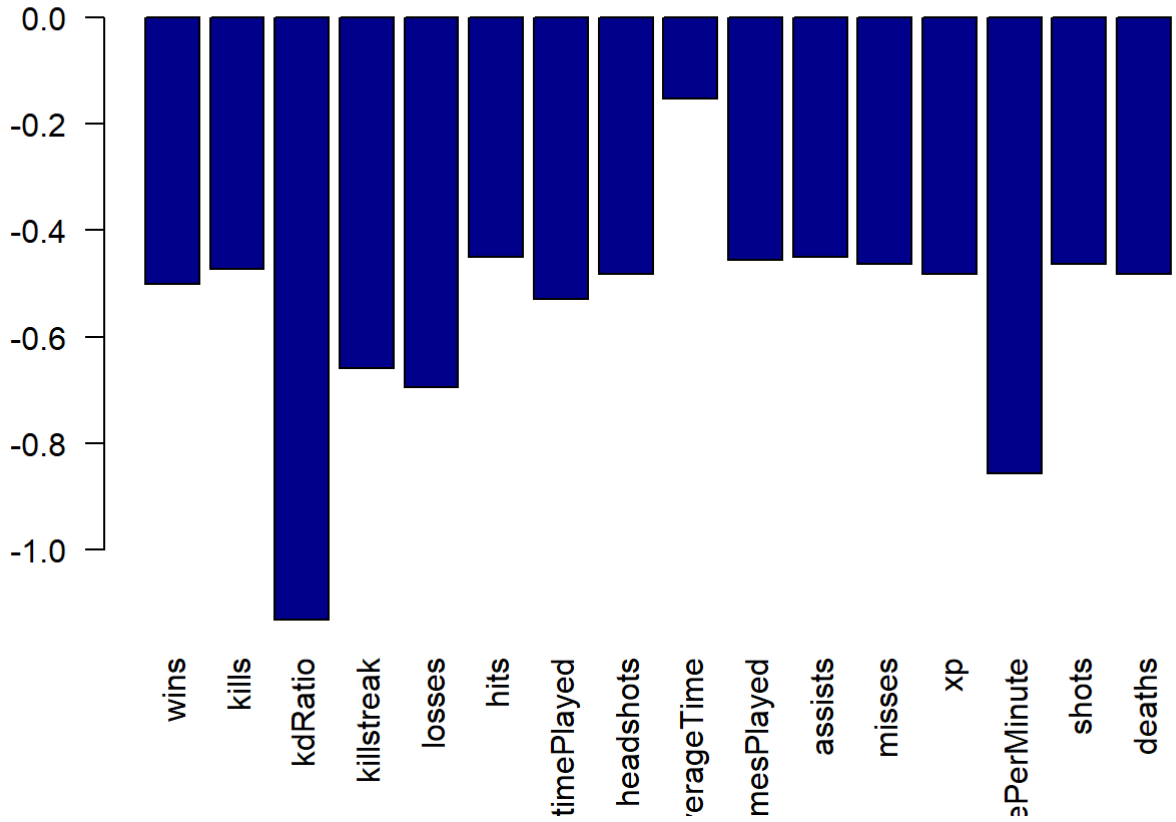
```
centers=km$centers
barplot(centers[1,], las=2, col="darkblue")
```




```
barplot(centers[2,], las=2, col="darkblue")
```



```
barplot(centers[3,], las=2, col="darkblue")
```

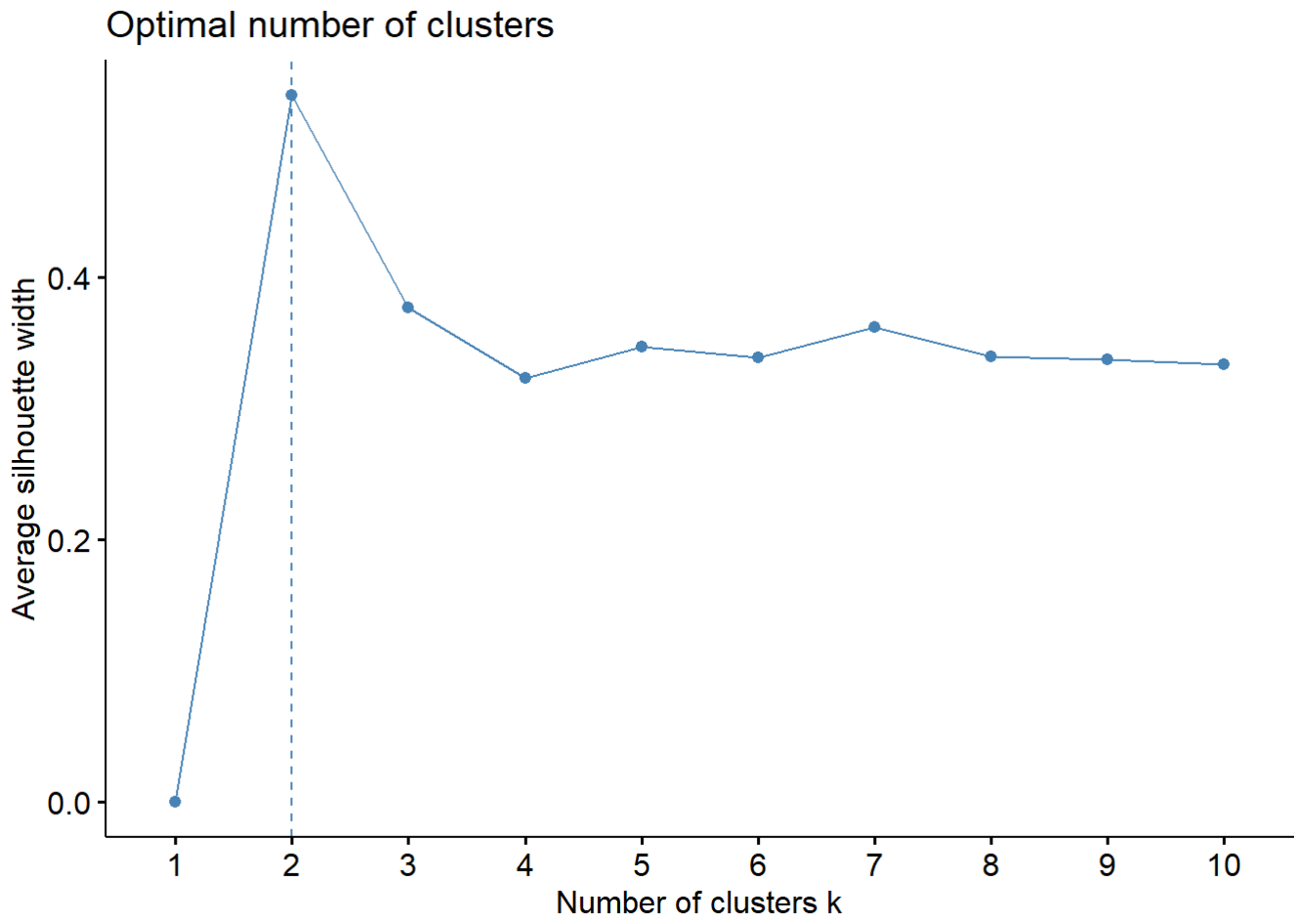


PAM

Let´s continue with K-medoids or PAM. It is similar to k-means,but now the centers are indeed observations (medoids) instead of means

As before, we are going to plot what happens with the first 10 centers in order to choose the best option.

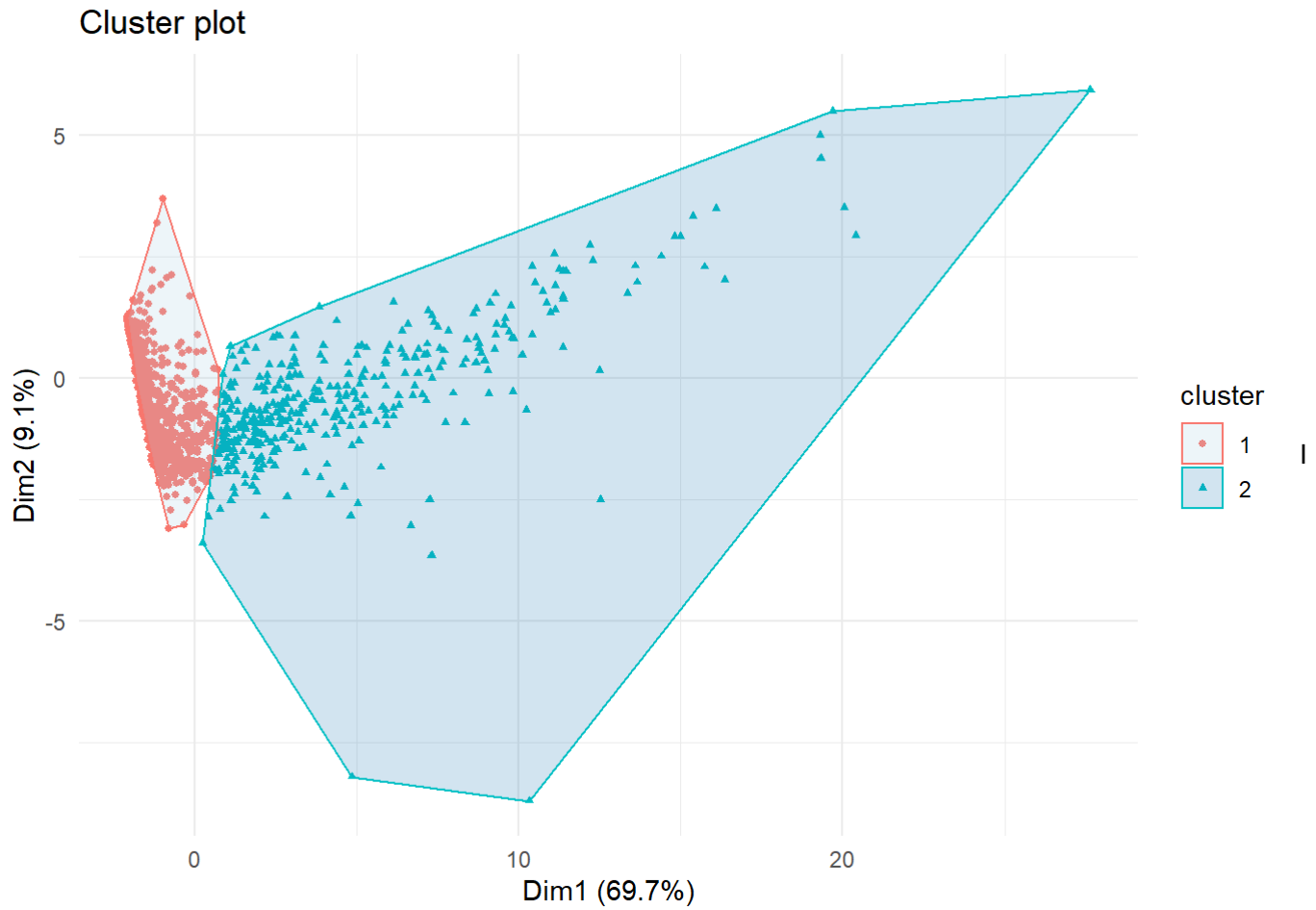
```
fviz_nbclust(data_cl, pam, method = 'silhouette', k.max = 10)
```



After look at the plot, I think that 2 centers is the best option

```
cl.pam=eclust(data_cl, "pam", stand=TRUE, k=2, graph=F)

fviz_cluster(cl.pam, data = X, geom = c("point"), pointsize=1)+
  theme_minimal()+scale_fill_brewer(palette="Paired")
```



have removed the names because they can not be distinguish, they are very close.

Kernel k-means

Now, we will work with Kernel k-means, partitioning clustering based on a non-linear distance (Firstly we have to create a matrix)

```
cl.ker=kkmeans(as.matrix(data_cl), centers=3, kernel="rbfdot")
```

```
## Using automatic sigma estimation (sigest) for RBF or laplace kernel
```

```
centers(cl.ker)
```

```
##           [,1]       [,2]       [,3]       [,4]       [,5]       [,6]
## [1,] -0.5043448 -0.4728850 -1.2814718 -0.6724031 -0.7063522 -0.4499970
## [2,]  0.4036362  0.3974577  0.5974453  0.4875194  0.5271127  0.3846973
## [3,] -0.3856234 -0.4122958  0.1632740 -0.3789791 -0.4381058 -0.4096328
##           [,7]       [,8]        [,9]      [,10]     [,11]      [,12]
## [1,] -0.5352781 -0.4827829 -0.19898523 -0.4551107 -0.450990 -0.4631069
## [2,]  0.4143434  0.4035472  0.04283219  0.3854364  0.380669  0.3938126
## [3,] -0.3717650 -0.4149725  0.15868343 -0.4045893 -0.397515 -0.4159807
##          [,13]      [,14]      [,15]      [,16]
## [1,] -0.4843881 -0.9052930 -0.4628853 -0.4832268
## [2,]  0.3876966  0.7245269  0.3940121  0.4022755
## [3,] -0.3704503 -0.6922020 -0.4168175 -0.4109681
```

```
size(cl.ker)
```

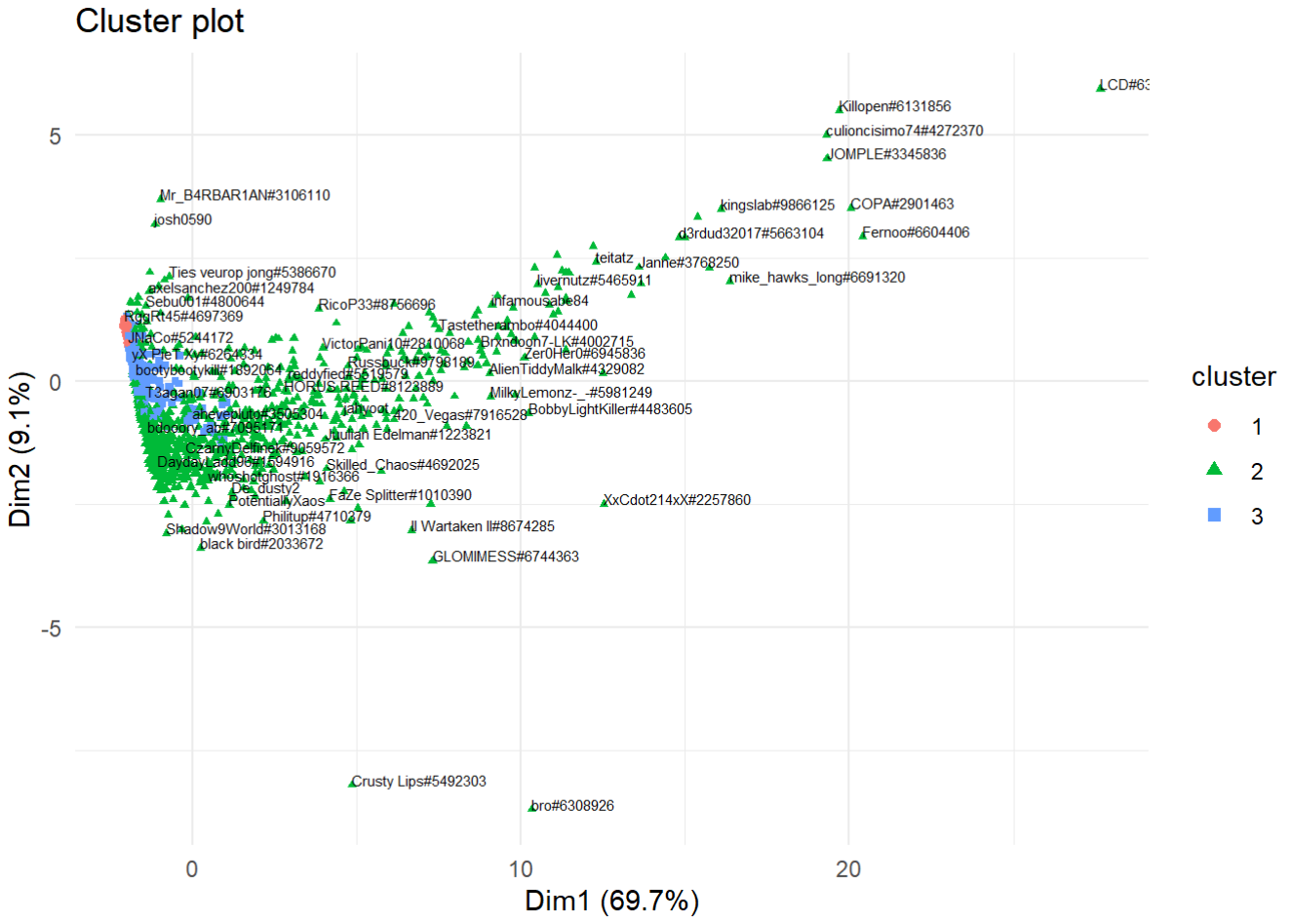
```
## [1] 424 825 309
```

```
withinss(cl.ker)
```

```
## [1]   801.9906 19014.8771   836.9562
```

```
cl1.ker = list(data = data_cl, cluster = cl.ker@.Data)

fviz_cluster(cl1.ker, geom = c("point"), ellipse=F,pointsize=1)+
  theme_minimal()+geom_text(label=names,hjust=0, vjust=0,size=2,check_overlap = T)+scale_fill
_brewer(palette="Paired")
```



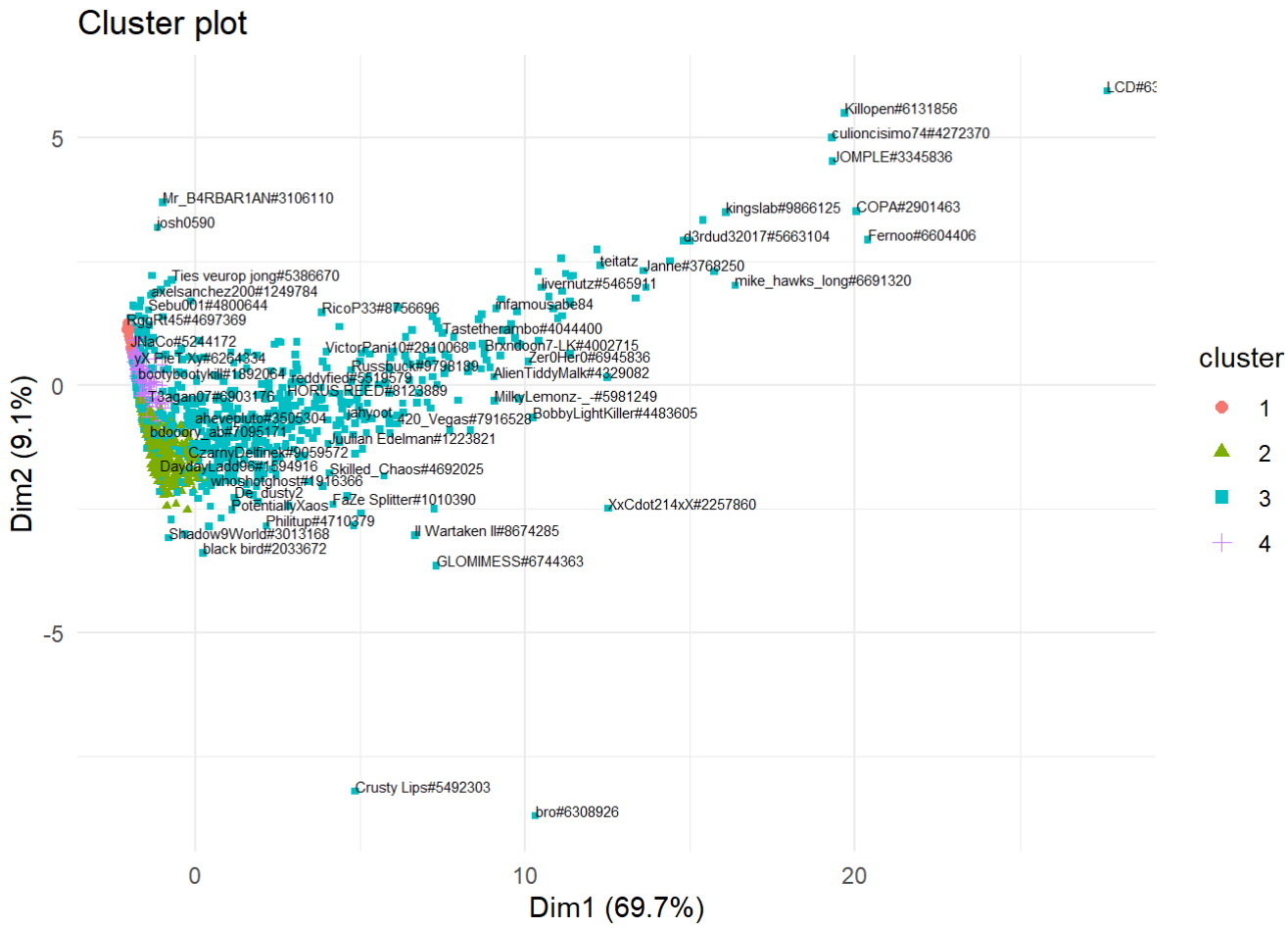
I have decide two select 4 centers as before, but when I have seen the plot I have decide that 5 centers would be a better option.

```
cl.ker=kkmeans(as.matrix(data_cl), centers=4, kernel="rbfdot")
```

```
## Using automatic sigma estimation (sigest) for RBF or laplace kernel
```

```
cl1.ker = list(data = data_cl, cluster = cl.ker@.Data)

fviz_cluster(cl1.ker, geom = c("point"), ellipse=F,pointsize=1)+
  theme_minimal()+geom_text(label=names,hjust=0, vjust=0,size=2,check_overlap = T)+scale_fill
_brewer(palette="Paired")
```



I think that 4 centers is better for this type of clustering, because in the plot the difference between groups is more clear.

Hierarchical clustering

Let´s continue with a different clustering, Hierarchical clustering, specifically we will plot a dendogram, which is different if we compare it with the previous ones.

We will prove with 4 and 5 centers, and after see the plot we will decide the one that we prefer

```
d = dist(data_cl, method = "euclidean")
hc = hclust(d, method = "complete")

#fviz_dend(hc, k = 4, cex = 0.35, rect = T) +
#theme(text = element_text(size = 6))
```

```
#fviz_dend(hc, k = 5, cex = 0.35, rect = T) +
#theme(text = element_text(size = 6))
```

I have been trying this code, but my laptop does not have power enough to plot this two dendograms, so I can not choose if I prefer 4 centers or 5.

And that is the hole analysis of the database using unsupervised learning. As we have seen using unsupervised models such as PCA, FACTOR ANALYSIS AND CLUSTERING, we can reduce the dimension of our data and group them, taking into account if they are similar and they have something in common. The disadvantage of this tools is that everything is subjective, so it is very difficult to know if we are doing well.