

# Projet R : Fortnite

Écrit par : Enzo LERICHE, Franck TANKHAPANYA

2023-12-18

# Contents

<b>Introduction</b>	<b>3</b>
Présentation du projet . . . . .	3
Présentation de Fortnite . . . . .	3
Présentation du jeu de données . . . . .	3
<b>Analyse</b>	<b>4</b>
Importation des données . . . . .	4
Descriptions des données . . . . .	4
Création de nouvelles variables . . . . .	5
Choix des variables . . . . .	6
Séparation de la base de données . . . . .	6
Description des données choisies . . . . .	6
Résumés statistiques par variable . . . . .	7
Avec une boucle . . . . .	7
Avec une fonction . . . . .	15
Résumés statistiques croisés . . . . .	16
Groupe pour les placements . . . . .	17
Graphique pour le placements peut importe l'état high ou sober . . . . .	18
Graphique pour du placement pour l'état high . . . . .	18
Graphique pourdu placement pour l'état sober . . . . .	19
Facet grid . . . . .	20
Évolution du placement en fonction de la date et de l'heure (graphique compliqué) . . . . .	21
Parti de Test . . . . .	23
Test de comparaison de moyennes . . . . .	23
Corrélation et test de corrélation . . . . .	24
Test $\chi^2$ d'indépendance . . . . .	26
<b>Conclusion</b>	<b>27</b>

# Introduction

## Présentation du projet

***Consigne sur sujet :** L'objectif de ce projet est d'appliquer les connaissances acquises pendant le cours de programmation statistique avec R. Chaque groupe doit s'appropriier sa base de données, en faire des résumés statistiques et réaliser différents graphiques d'analyse descriptive. Les graphiques et les statistiques diffèrent en fonction des sujets, il faut choisir lesquels sont pertinents à réaliser.*

## Présentation de Fortnite

Fortnite, un jeu vidéo développé par Epic Games, a pris d'assaut la scène du gaming depuis son lancement en 2017. Avec son mode Battle Royale très populaire, le jeu rassemble 100 joueurs sur une île où la mission est de rester le dernier survivant.

Les différents modes de jeu, tels que Solo (individuel), Duo (en équipe de deux), Trio (en équipe de trois) et Squad (en équipe de quatre), offrent une expérience sociale et stratégique. Jouer avec des amis pour atteindre la victoire ajoute une dynamique amusante au jeu.

## Présentation du jeu de données

Notre jeu de données est composé de variables récoltées dans le jeu vidéo Fortnite. Plus précisément, il s'agit d'une personne qui a récolté ses données sur une période de 80 parties. Il cherche des conclusions possibles entre la sobriété et les statistiques dans le jeu. Il se demande si la sobriété lui permet d'être meilleur. Notre base de données contient 16 variables que nous pourrons vous présenter juste après.

# Analyse

## Importation des données

Nous importons notre jeu de données sur Rstudio. Pour cela, nous utilisons la library “*readxl*”, pour pouvoir utiliser la fonction “*read\_excel*”. Nous avons nommé notre data frame “*data*” pour plus de simplicité.

```
library(readxl)
```

```
## Warning: le package 'readxl' a été compilé avec la version R 4.3.2
```

```
data <- read_excel("Z:/R/projet/fortnite_statistics.xlsx")
head(data, 3)
```

```
## # A tibble: 3 x 16
##   Date           'Time of Day'      Placed 'Mental State' Eliminations
##   <dtm>          <dtm>          <dbl> <chr>          <dbl>
## 1 2018-04-10 00:00:00 1899-12-31 18:00:00      27 sober            2
## 2 2018-04-10 00:00:00 1899-12-31 18:00:00      45 sober            1
## 3 2018-04-10 00:00:00 1899-12-31 18:00:00      38 high             3
## # i 11 more variables: Assists <dbl>, Revives <dbl>, Accuracy <dbl>,
## #   Hits <dbl>, 'Head Shots' <dbl>, 'Distance Traveled' <dbl>,
## #   'Materials Gathered' <dbl>, 'Materials Used' <dbl>, 'Damage Taken' <dbl>,
## #   'Damage to Players' <dbl>, 'Damage to Structures' <dbl>
```

Nous avons rajouté “*head(data, 3)*” qui permet d’afficher les 3 premières lignes de notre base de données. On peut donc bien vérifier la présence des 16 colonnes.

## Descriptions des données

Précisément voici la liste des variables :

Variables	Types	Descriptions	Pertinence
Date	Date	Date du jour jouée	non
Time of Day	Time	Heure de la journée jouée	non
Placed	Numérique	Placement à la fin de la partie	Oui
Mental State	Caractère	Indique si le joueur était intoxiqué par la marijuana ou non	Oui
Eliminations	Numérique	Nombre d’éliminations faites durant la partie	Oui
Assists	Numérique	Nombre d’assistances faites durant la partie	Oui
Revives	Numérique	Nombre de réanimations faites durant la partie	Non
Accuracy	Numérique	Correspond à la précision	Oui
Hits	Numérique	Nombre de tirs réussis durant la partie	Oui
Head Shots	Numérique	Nombre de tirs dans la tête faits durant la partie	Oui
Distance Traveled	Numérique	Correspond à la distance parcourue en mètres durant la partie	Non
Materials Gathered	Numérique	Nombre de matériaux récupérés	Non
Materials Used	Numérique	Nombre de matériaux utilisés	Non
Damage Taken	Numérique	Correspond aux dommages subis	Oui

Variables	Types	Descriptions	Pertinence
Damage to Players	Numérique	Correspond aux dommages infligés aux joueurs	Oui
Damage to Structures	Numérique	Correspond aux dommages infligés aux structures	Non

Parmi les 16 variables, 8 sont vraiment pertinentes pour analyser un potentiel lien entre la sobriété et les performances en jeux.

Les variables relatives à la date et à l'heure pourraient également être pertinentes, dans le but d'explorer d'autres éventuels liens, en dehors de l'influence de la marijuana, sur les performances individuelles.

## Création de nouvelles variables

On décide de créer une nouvelle colonne nommée “*score*”, cette colonne aura 4 valeurs possibles. On crée des classes : [1,10] correspond à “TOP”, [10,25] correspond à “BON”, [25,50] correspond à “MOYEN”, [50,100] correspond à “MAUVAIS”. Pour cela on utilise des conditions ifelse. Une fois la nouvelle variable créée on décide de transformer la colonne en une variable catégorielle, cela permet aussi pour les graphiques d'avoir un ordre prédéfini.

```
library(dplyr)
```

```
## Warning: le package 'dplyr' a été compilé avec la version R 4.3.2
```

```
##
```

```
## Attachement du package : 'dplyr'
```

```
## Les objets suivants sont masqués depuis 'package:stats':
```

```
##
```

```
## filter, lag
```

```
## Les objets suivants sont masqués depuis 'package:base':
```

```
##
```

```
## intersect, setdiff, setequal, union
```

```
data <- data %>%
  mutate(score = ifelse(Placed <= 10, "TOP",
                        ifelse(Placed <= 25, "BON",
                              ifelse(Placed <= 50, "MOYEN", "MAUVAIS"))))
data$score <- factor(data$score, levels = c("TOP", "BON", "MOYEN", "MAUVAIS"))
```

En analysant les données, on s'est rendu compte que la colonne “Time of Day” n'était pas très utilisable à cause de la date qui était comprise dedans. On a donc créé une nouvelle variable qui contient notre vraie date dans la colonne “Date” et l'heure de partie contenue dans la variable “Time of Day”. Pour cela on a créé une variable temporaire qui est utile uniquement pour créer la deuxième variable qui est celle qui nous intéresse. La première est nommée “Heure” qui contient uniquement l'heure donnée dans la colonne “Time of Day”. La deuxième variable nommée “DateHeure” est une concaténation de la variable “Date” et “Heure”.

```
data <- data %>%
  mutate(Heure = format(`Time of Day`, "%H:%M:%S"),
         DateHeure = paste(Date, Heure))
```

## Choix des variables

Pour nos futures analyses, on décide de mettre tous les indicateurs qui nous intéressent dans un vecteur. Le nom de ce vecteur est “noms\_variables”.

```
noms_variables <- c("Placed", "Eliminations", "Assists",  
                  "Accuracy", "Hits", "Head Shots",  
                  "Damage Taken", "Damage to Players", "score")
```

## Séparation de la base de données

Sachant que notre but est de déterminer une potentielle amélioration des performances de jeu grâce à la marijuana, on décide de créer 2 nouveaux data frames qui correspondent chacun à un état mentale. En effet, on a créé un data frame pour l'état mental correspondant à “high” et un autre pour l'état mental correspondant à “sober”. Ce qui nous permettra par la suite de réaliser des graphiques.

```
data_high <- data[data$`Mental State` == "high", noms_variables ]  
data_sober <- data[data$`Mental State` == "sober", noms_variables ]
```

## Description des données choisies

### *nombre de lignes*

```
nrow(data)
```

```
## [1] 87
```

Pour compter le nombre de lignes, on a utilisé la fonction nrow. Le resultat obtenu est de 87 lignes.

### *nombre de colonnes*

```
ncol(data)
```

```
## [1] 19
```

Pour compter le nombre de colonnes, on a utilisé la fonction ncol. Le resultat obtenu est de 19 colonnes.

### *nombre de valeurs manquantes*

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

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

Pour compter le nombre de valeurs manquantes, on a fait une somme du nombre de valeurs manquantes. Ici, la valeur est de 0, c'est à dire qu'il n'y a aucune valeur manquante.

### *nombre de parties pour les différents états*

```
nrow(data_high)
```

```
## [1] 42
```

```
nrow(data_sober)
```

```
## [1] 45
```

Chaque ligne permet de calculer respectivement le nombre de lignes contenues dans les data frames. Le nombre de ligne correspond aux nombres de parties jouées par rapport à la variable état mental. Ici, nous trouvons 42 parties où la personne est sous l'emprise de marijuana et 45 où la personne est sobre.

## Résumés statistiques par variable

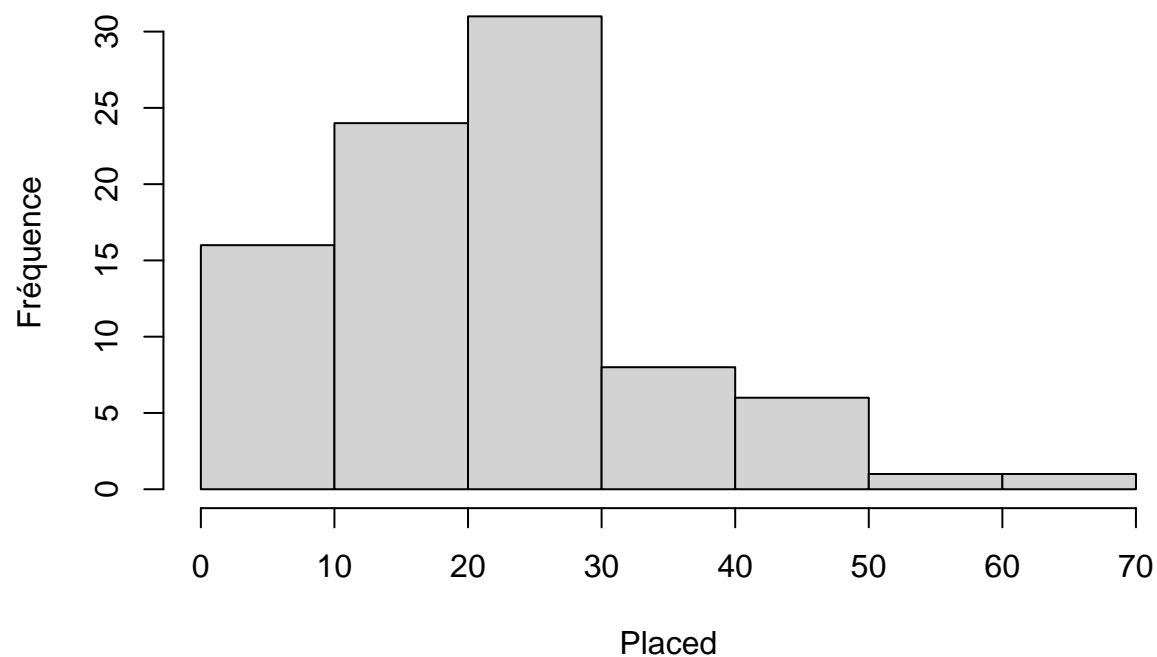
### Avec une boucle

On utilise une boucle qui permet de faire les résumés statistiques pour chaque variables.

```
noms_variables <- c("Placed", "Eliminations", "Assists",
                    "Accuracy", "Hits", "Head Shots",
                    "Damage Taken", "Damage to Players")
for (i in noms_variables){
  cat("Résumé pour la variables",i, "\n")
  print(summary(data[,i]))
  cat(" Sd.      :", sd(unlist(data[, i])), "\n")
  cat(" Effectif:", nrow(data[, i]), "\n")
  hist(unlist(data[,i]),main=paste("Histogramme de la variable\n", i),
       xlab=i, ylab="Fréquence")
  cat("\n")
}
```

```
## Résumé pour la variables Placed
##      Placed
##  Min.   : 1.00
## 1st Qu.:15.00
##  Median :21.00
##   Mean  :22.05
## 3rd Qu.:28.50
##   Max.  :66.00
##   Sd.   :13.14579
## Effectif: 87
```

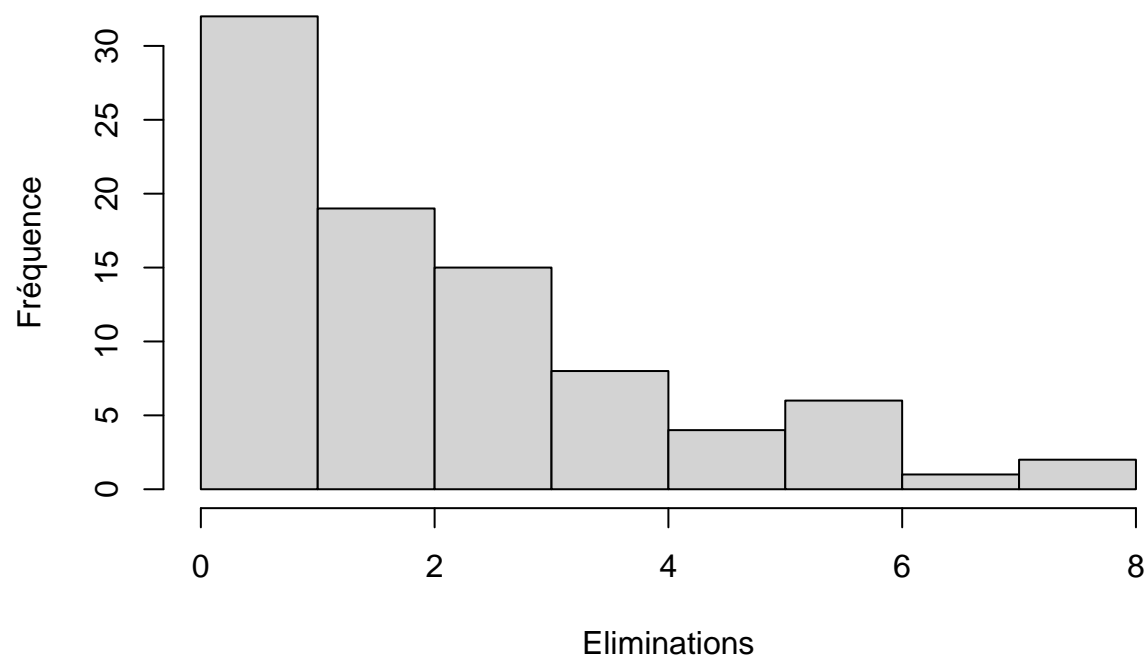
**Histogramme de la variable  
Placed**



```
##
## Résumé pour la variables Eliminations
## Eliminations
## Min. :0.000
## 1st Qu.:1.000
## Median :2.000
## Mean :2.517
## 3rd Qu.:3.000
## Max. :8.000
## Sd. : 1.885453
## Effectif: 87
```

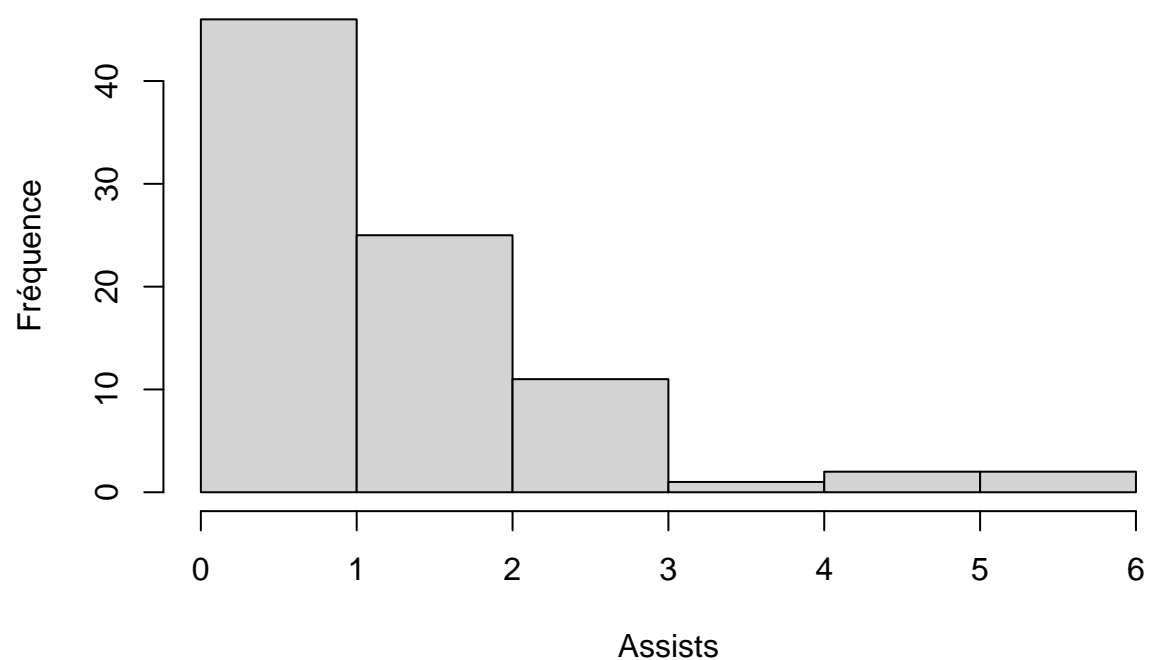


## Histogramme de la variable Eliminations

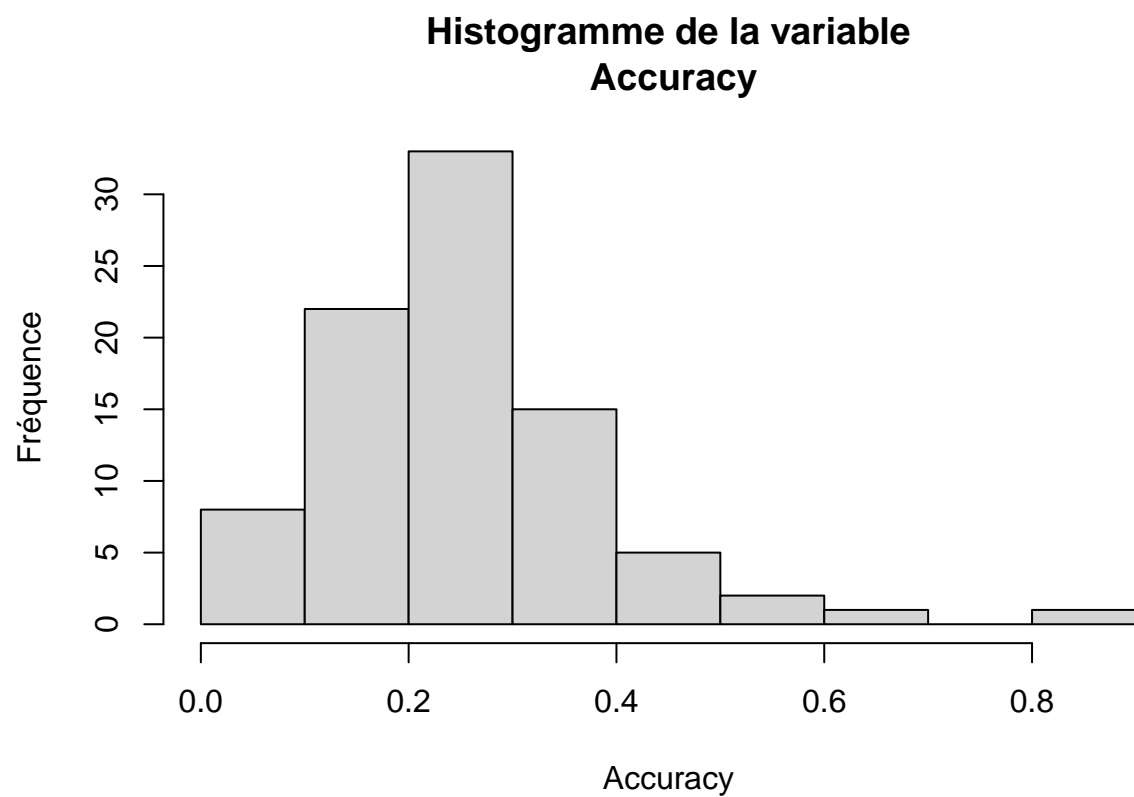


```
##
## Résumé pour la variables Assists
##   Assists
##   Min.   :0.000
##   1st Qu.:0.000
##   Median :1.000
##   Mean   :1.483
##   3rd Qu.:2.000
##   Max.   :6.000
##   Sd.    : 1.388173
##   Effectif: 87
```

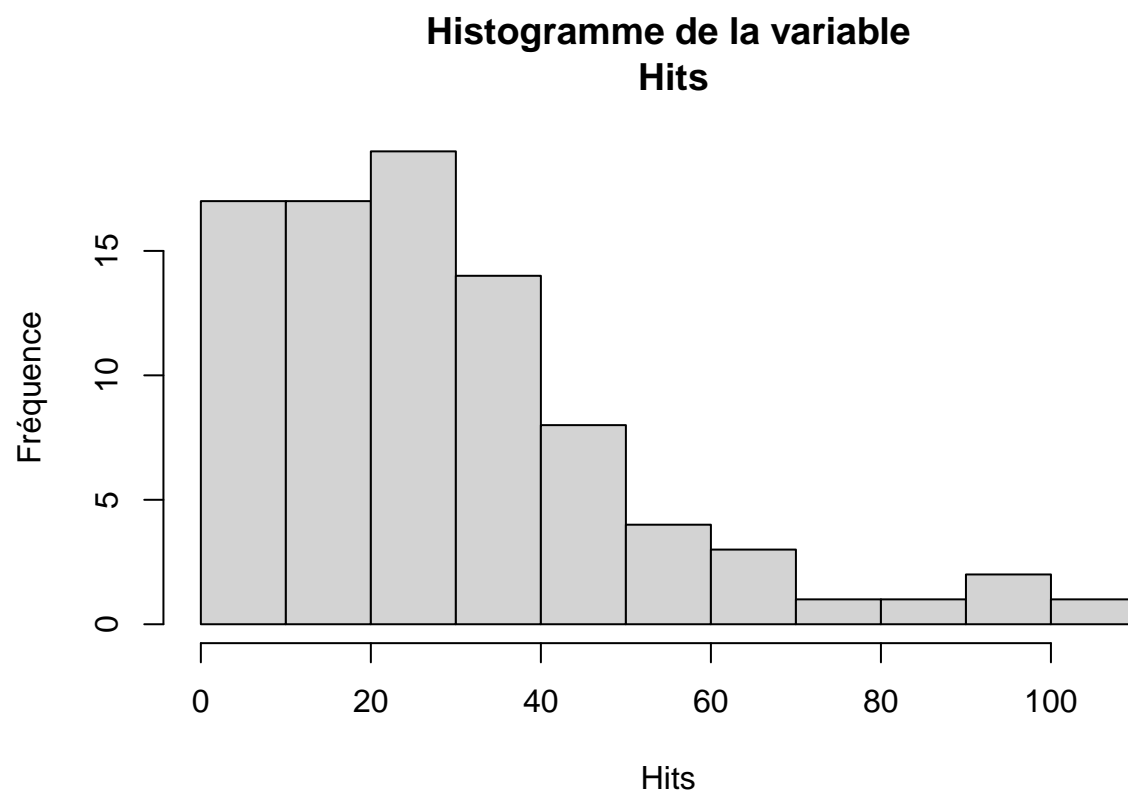
## Histogramme de la variable Assists



```
##
## Résumé pour la variables Accuracy
## Accuracy
## Min. :0.0400
## 1st Qu.:0.1700
## Median :0.2500
## Mean :0.2601
## 3rd Qu.:0.3200
## Max. :0.9000
## Sd. : 0.1347133
## Effectif: 87
```

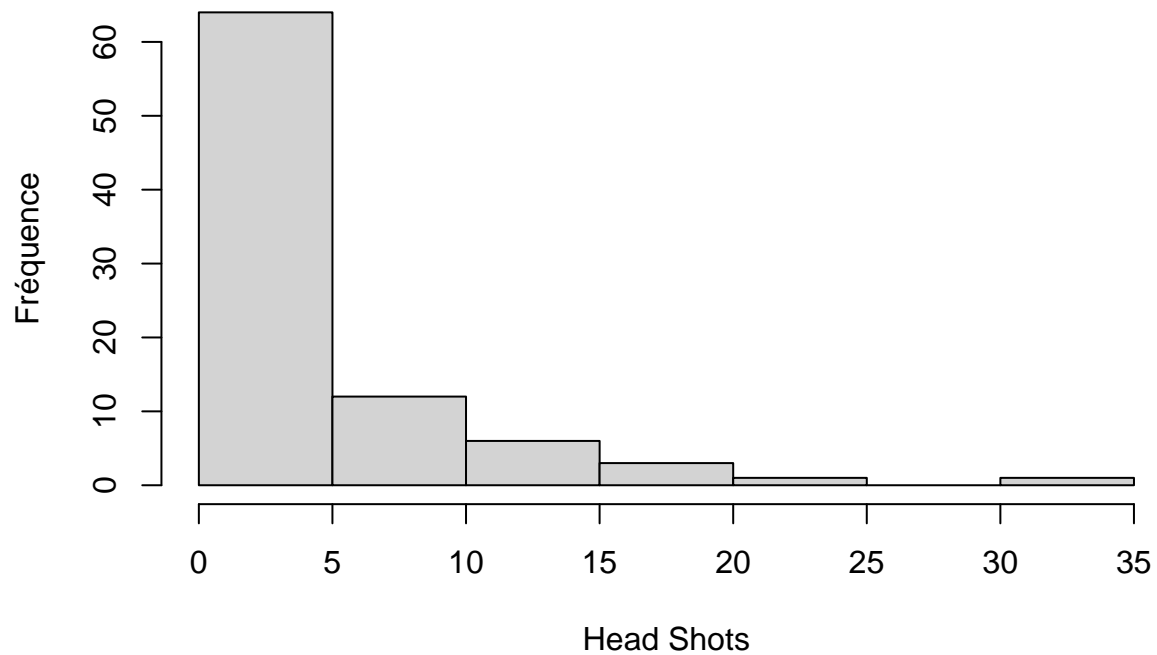


```
##
## Résumé pour la variables Hits
##      Hits
##  Min.   : 1.00
## 1st Qu.: 12.00
##  Median : 27.00
##   Mean  : 29.74
## 3rd Qu.: 38.00
##   Max.  :105.00
##   Sd.   : 22.0936
## Effectif: 87
```



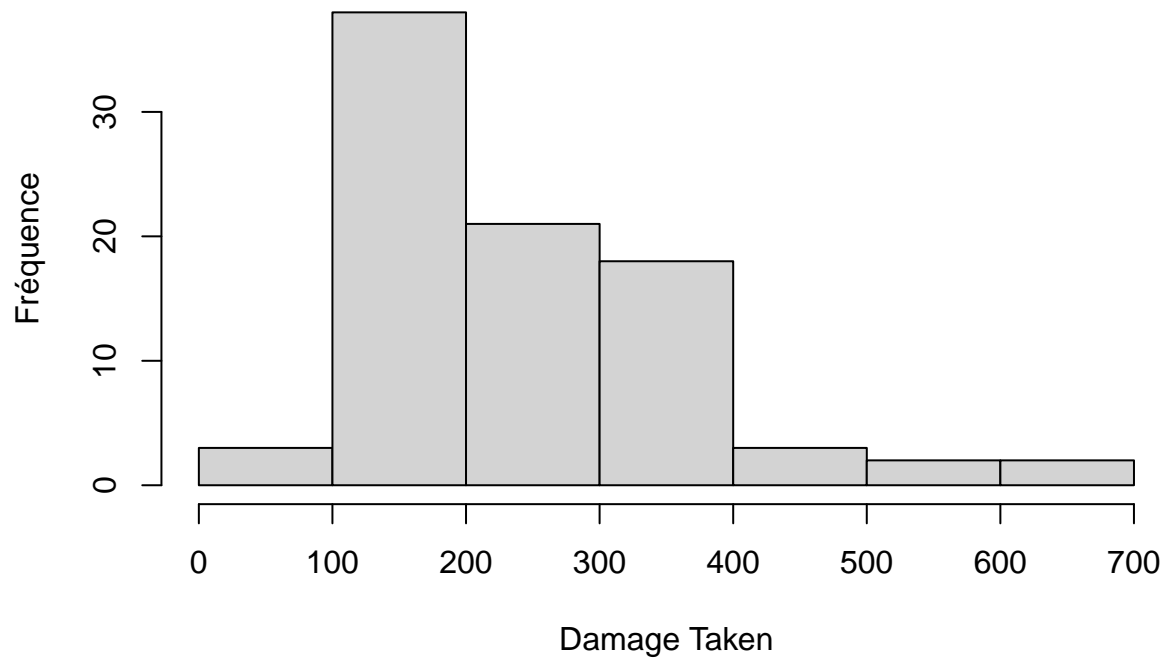
```
##
## Résumé pour la variables Head Shots
##   Head Shots
##   Min.   : 0.000
##   1st Qu.: 1.000
##   Median : 3.000
##   Mean   : 4.747
##   3rd Qu.: 6.500
##   Max.   :33.000
##   Sd.    : 5.777298
##   Effectif: 87
```

## Histogramme de la variable Head Shots

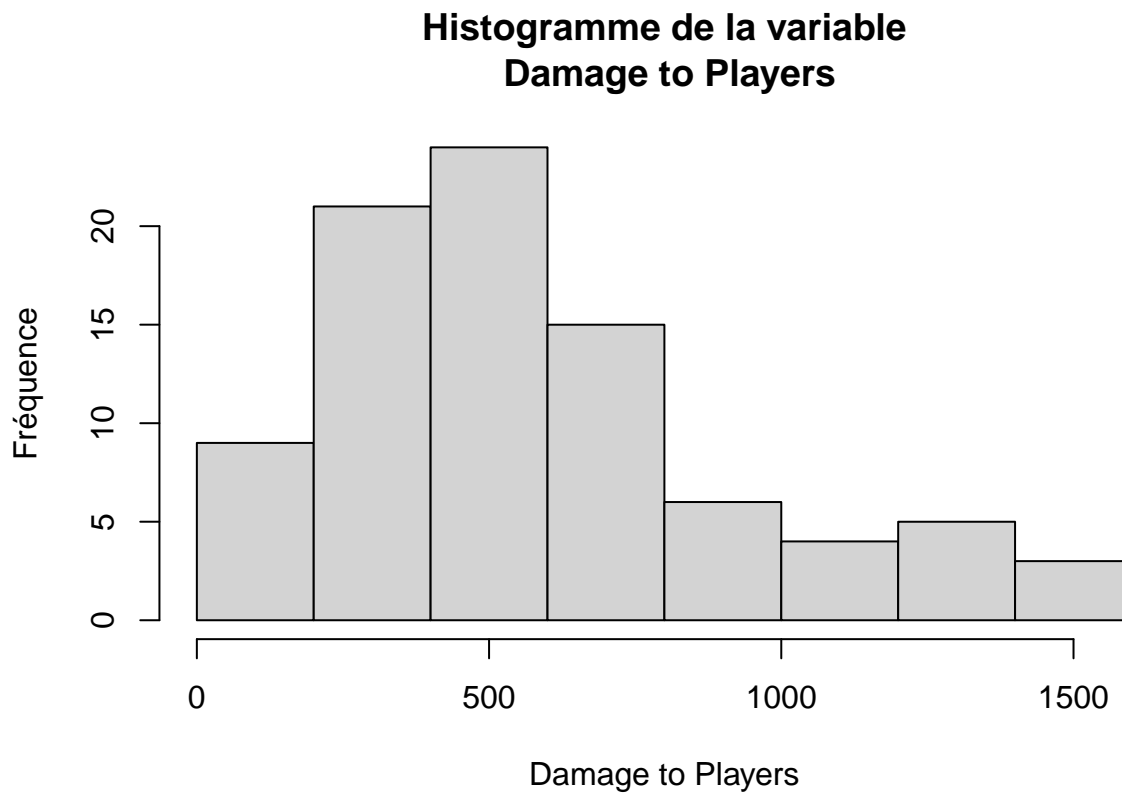


```
##
## Résumé pour la variables Damage Taken
##   Damage Taken
##   Min.   : 19.0
##   1st Qu.:154.0
##   Median :209.0
##   Mean   :244.2
##   3rd Qu.:316.5
##   Max.   :677.0
##   Sd.    :124.9374
##   Effectif: 87
```

## Histogramme de la variable Damage Taken



```
##
## Résumé pour la variables Damage to Players
## Damage to Players
## Min.   : 60.0
## 1st Qu.: 328.5
## Median : 481.0
## Mean   : 581.8
## 3rd Qu.: 750.0
## Max.   :1507.0
## Sd.    : 354.1724
## Effectif: 87
```



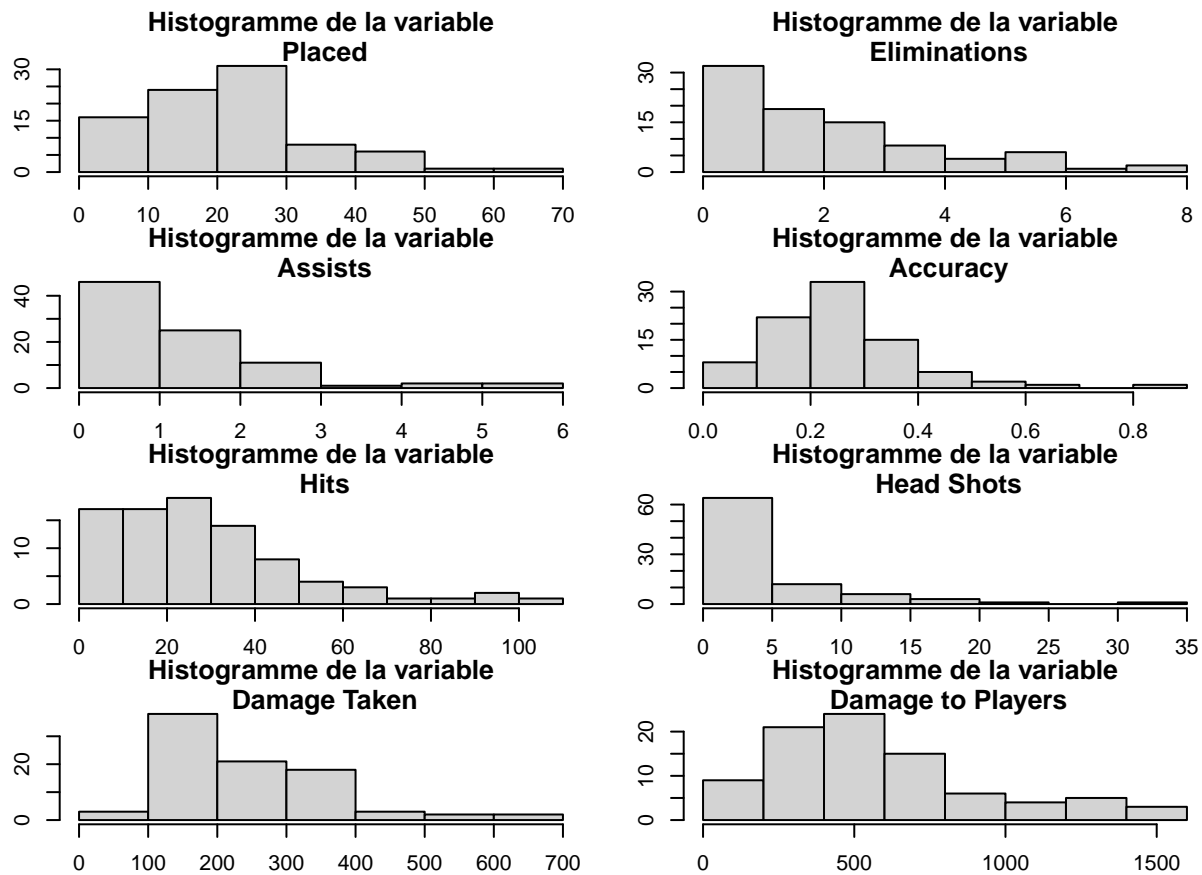
#### Avec une fonction

Il est aussi possible de faire la même chose avec une fonction. Nous procédons avec cette méthode car c'était demandé dans le sujet.

```
par(mar = c(2, 2, 2, 2))
par(mfrow = c(4, 2))

generation_histogramme <- function(data, variable) {
  for (i in noms_variables){
    hist(unlist(data[,i]),main=paste("Histogramme de la variable\n", i),
         xlab=i, ylab="Fréquence")
  }
}

# Exemple d'utilisation de la fonction avec la variable "Placed"
generation_histogramme(data, noms_variables)
```

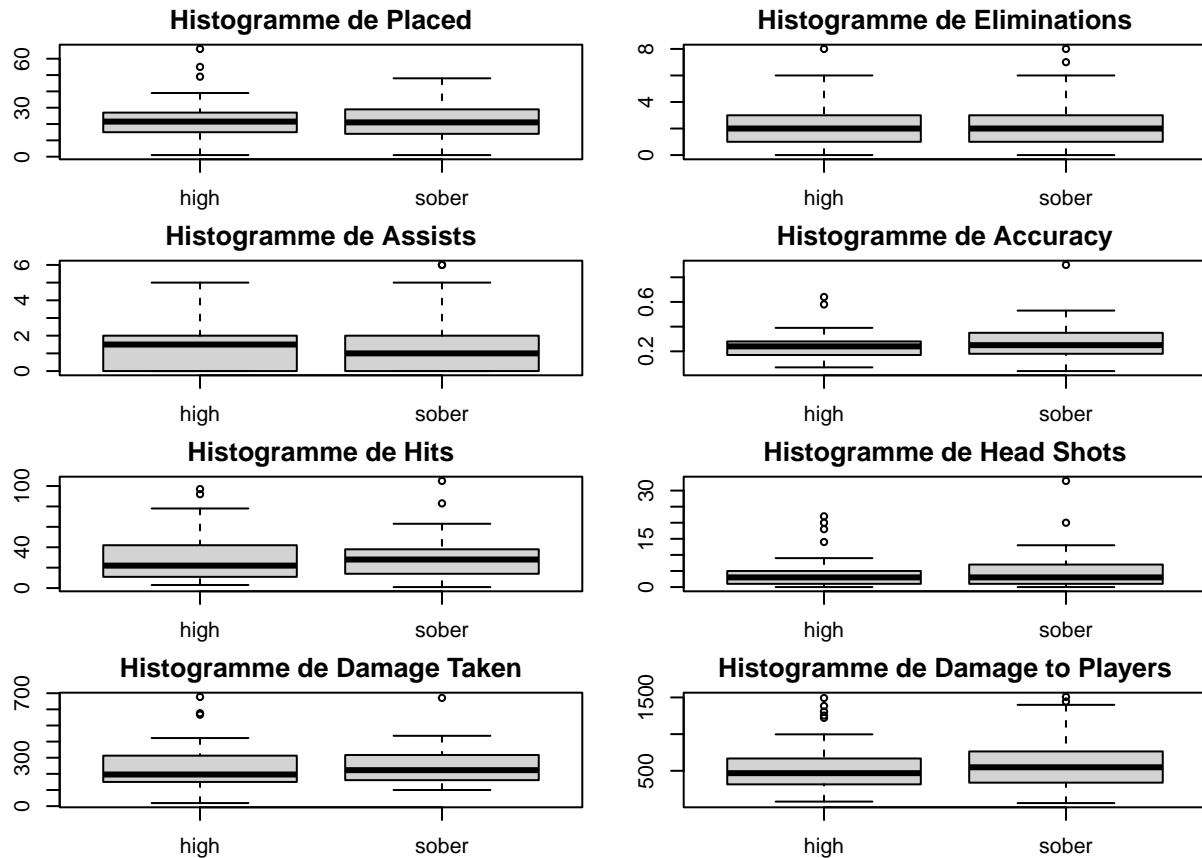


## Résumés statistiques croisés

On décide de faire des boxplots en croisant chaque variable avec l'état mental, ce qui permet de comparer les résultats des performances entre "high" et "sober".

```
par(mar = c(2, 2, 2, 2))
par(mfrow = c(4, 2))
for (i in noms_variables){
  boxplot(unlist(data[,i])~data$`Mental State`, main=paste("Histogramme de",i),xlab=i)
}
```





On s'intéresse au nombre d'éliminations pendant les différentes parties en prenant en compte l'état mental. On fait donc un tableau croisé qui nous permettra d'obtenir le nombre de parties qui correspond aux éliminations.

```
table(data$`Mental State`, data$Eliminations)
```

```
##
##      0  1  2  3  4  5  6  7  8
## high  3 13  7  9  5  1  3  0  1
## sober  4 12 12  6  3  3  3  1  1
```

On s'est aussi intéressé aux scores par rapport à l'état mental. Grâce au tableau croisé en sortie, on va pouvoir comparer les différents classements entre "high" et "sober". Vous verrez que prochainement, on décidera de le représenter aussi graphiquement pour que ce soit plus clair visuellement.

```
table(data$`Mental State`, data$score)
```

```
##
##      TOP BON MOYEN MAUVAIS
## high   7  23   10     2
## sober   9  19   17     0
```

## Groupe pour les placements

On s'intéresse maintenant principalement au placement. On va donc représenter visuellement cette variable pour voir les tendances et les évidences s'il y en a.

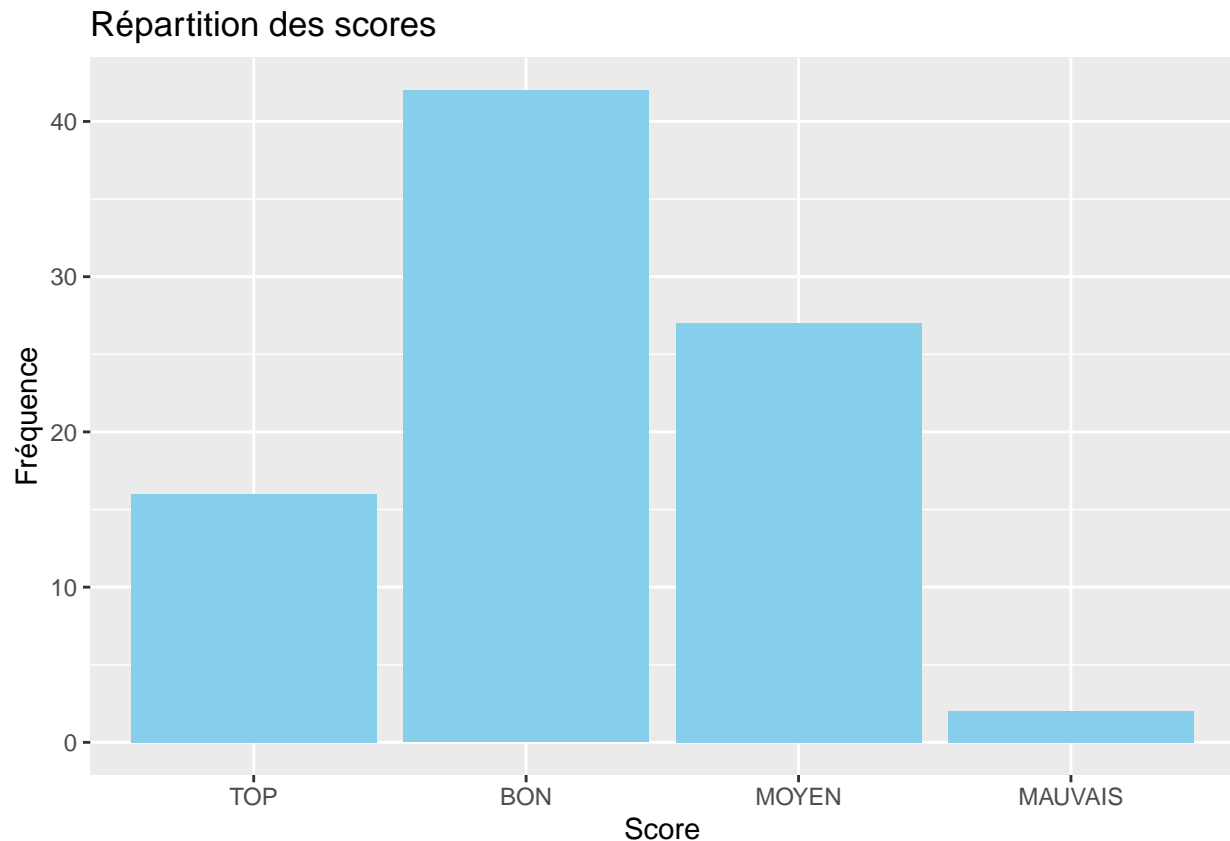
## Graphique pour le placements peut importe l'état high ou sober

On va représenter graphique la variable placement en ne faisant pas attention à l'état mental. Cela nous permettra d'avoir une vision de l'ensemble de notre jeu de données.

```
library(ggplot2)
```

```
## Warning: le package 'ggplot2' a été compilé avec la version R 4.3.2
```

```
ggplot(data, aes(x = score)) +  
  geom_bar(fill = "skyblue") +  
  labs(title = "Répartition des scores",  
        x = "Score",  
        y = "Fréquence")
```



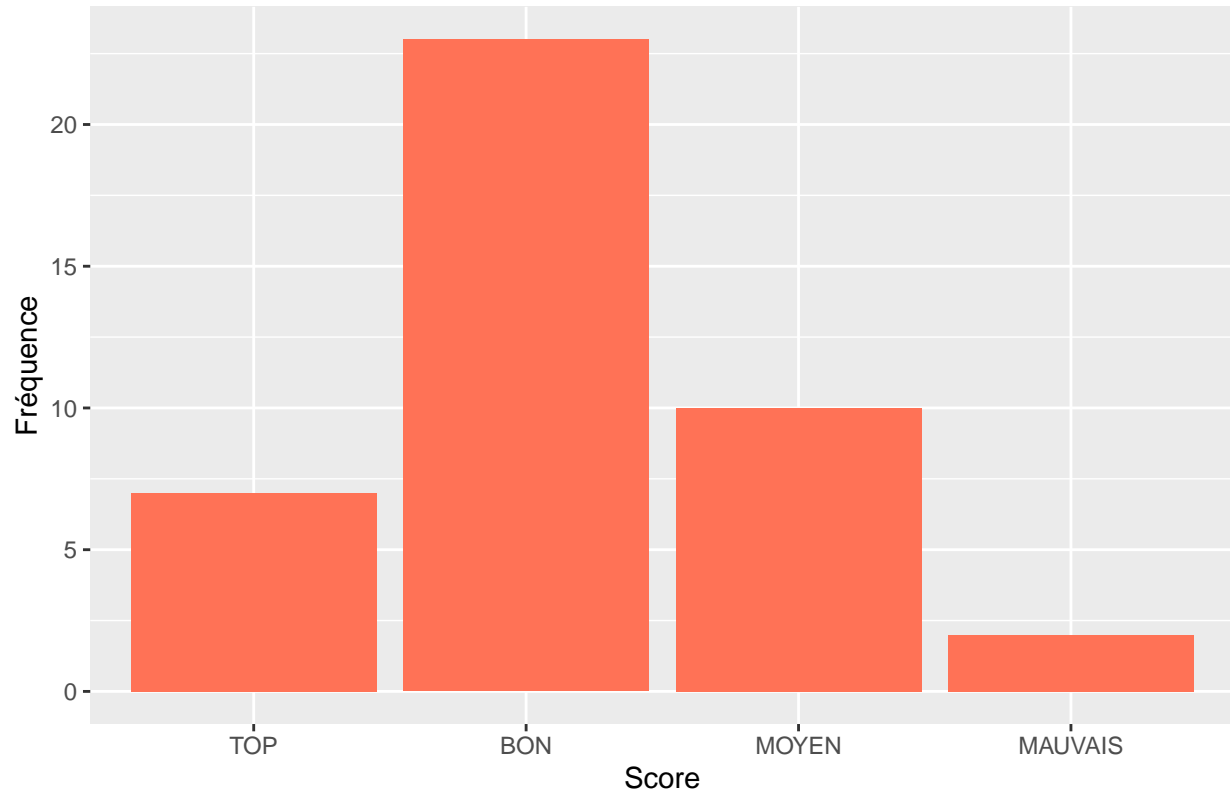
## Graphique pour du placement pour l'état high

Après avoir vu les tendances générales du placement sur notre jeu de données, on va s'intéresser à comparer les différents états mentaux pour la variable "Placement". On va commencer par faire par rapport à "high", c'est à dire quand la personne a fumé de la majijuana.

```
ggplot(data_high, aes(x = score)) +  
  geom_bar(fill = "#FF7256") +  
  labs(title = "Répartition des scores high",
```

```
x = "Score",  
y = "Fréquence")
```

Répartition des scores high

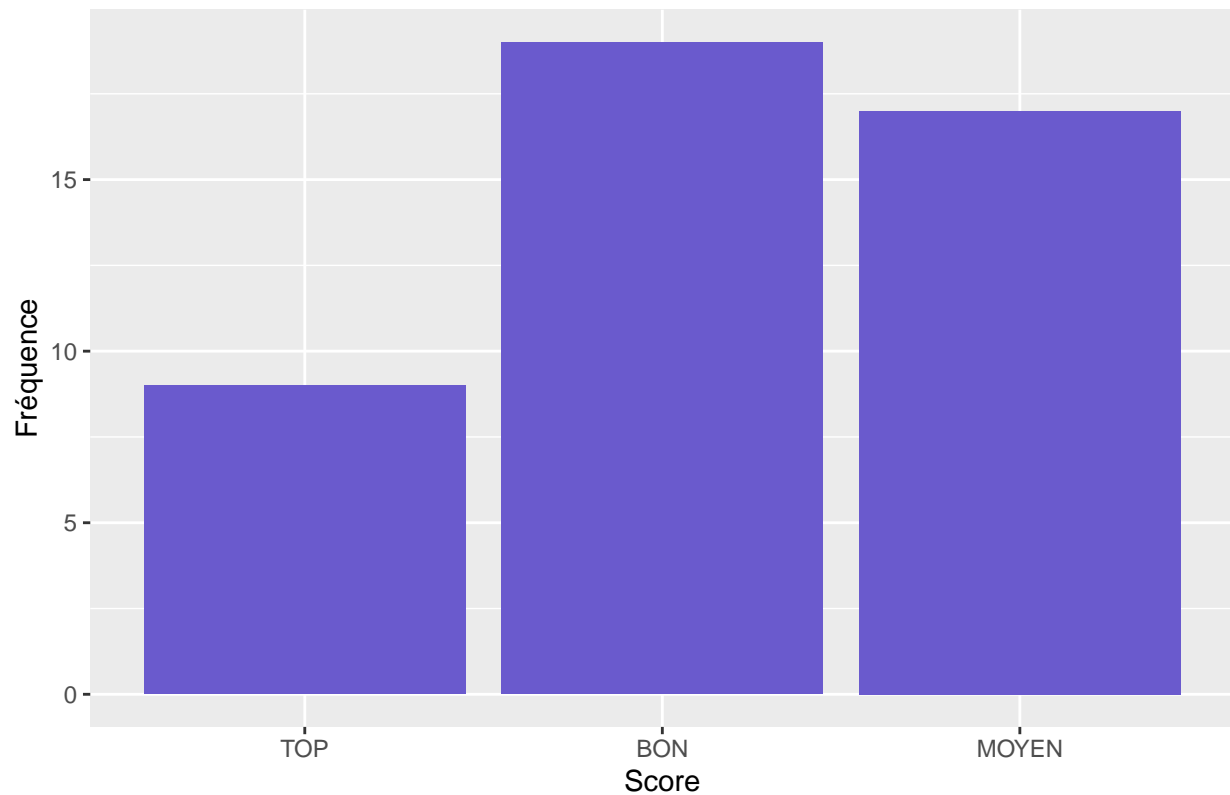


#### Graphique pour du placement pour l'état sober

Après avoir fait pour la valeur "high", on fait maintenant pour la valeur "sober". On pourra ensuite comparer les 2 graphiques.

```
ggplot(data_sober, aes(x = score)) +  
  geom_bar(fill = "slateblue") +  
  labs(title = "Répartition des scores sober",  
        x = "Score",  
        y = "Fréquence")
```

## Répartition des scores sober



Première chose marquante, on remarque que le graphique sober n'a que 3 barres. En effet, en étant sobre, la personne a fait 0 mauvaise partie.

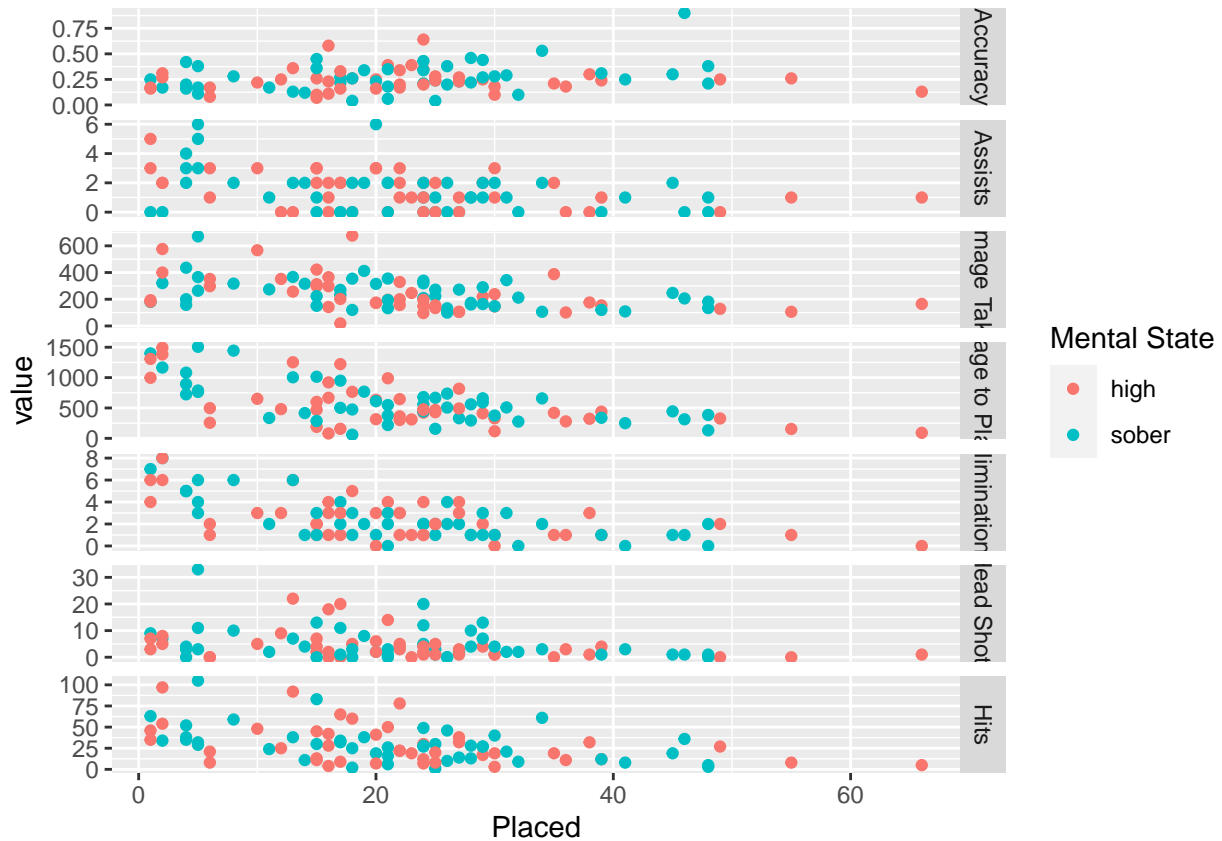
## Facet grid

On va utiliser facetgrid pour pouvoir afficher plusieurs sous-graphiques en un seul graphique.

```
library(tidyr)
```

```
## Warning: le package 'tidyr' a été compilé avec la version R 4.3.2
```

```
data2 <- data%>%select("Mental State","Placed", "Eliminations", "Assists",  
                      "Accuracy", "Hits", "Head Shots",  
                      "Damage Taken", "Damage to Players")  
  
table_longue <- data2 %>% gather( -Placed,-`Mental State`, key = "var", value="value")  
  
ggplot(table_longue, aes(Placed,value, color=`Mental State`)) +geom_point()+  
  facet_grid(rows=var~., scales = "free_y")
```



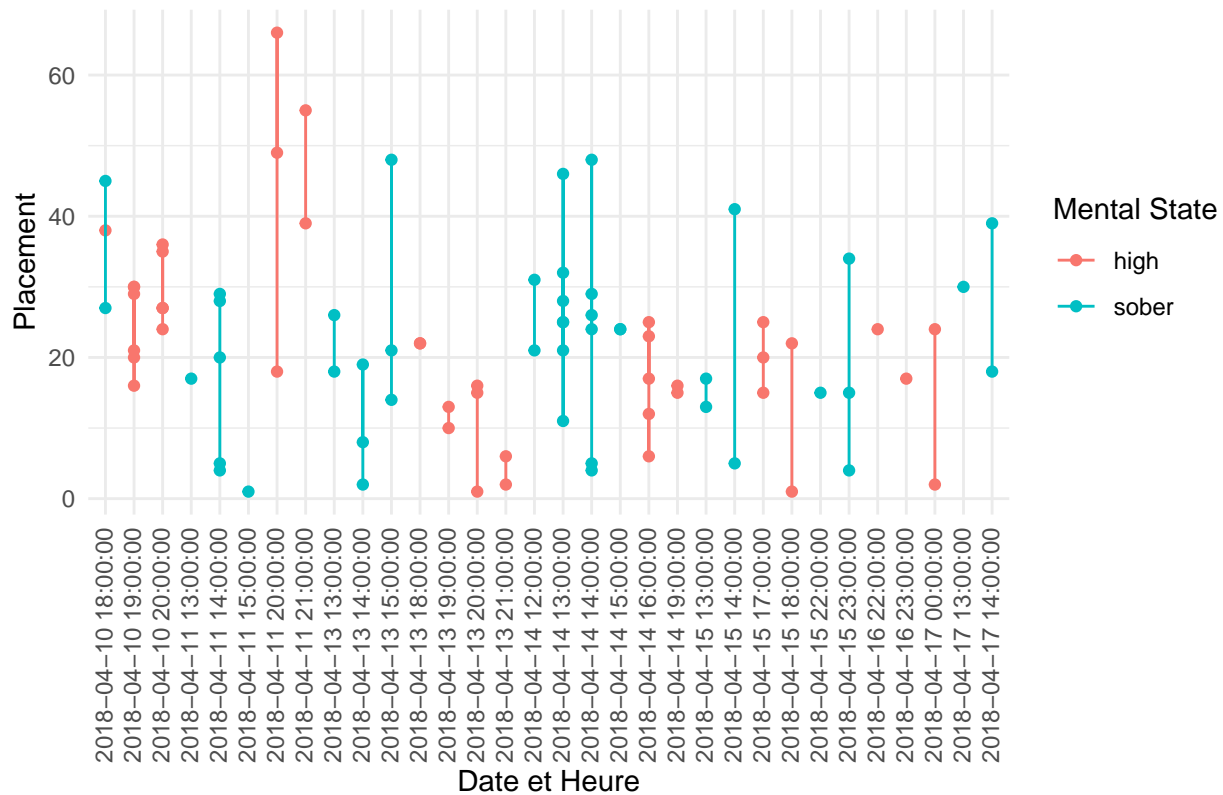
## Évolution du placement en fonction de la date et de l'heure (graphique compliqué)

On aimerait faire un graphique qui permet de voir l'évolution du placement en fonction de la date et de l'heure. On aimerait aussi afficher une courbe pour l'état mental correspondant à "high" et une autre pour "sober".

```
p <- ggplot(data, aes(x = DateHeure, y = Placed, color = `Mental State`)) +
  geom_point() +
  geom_line() +
  labs(title = "Évolution du Placement en fonction de la Date et de l'Heure",
       x = "Date et Heure",
       y = "Placement",
       color = "Mental State") +
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.4, hjust=1))
```

p

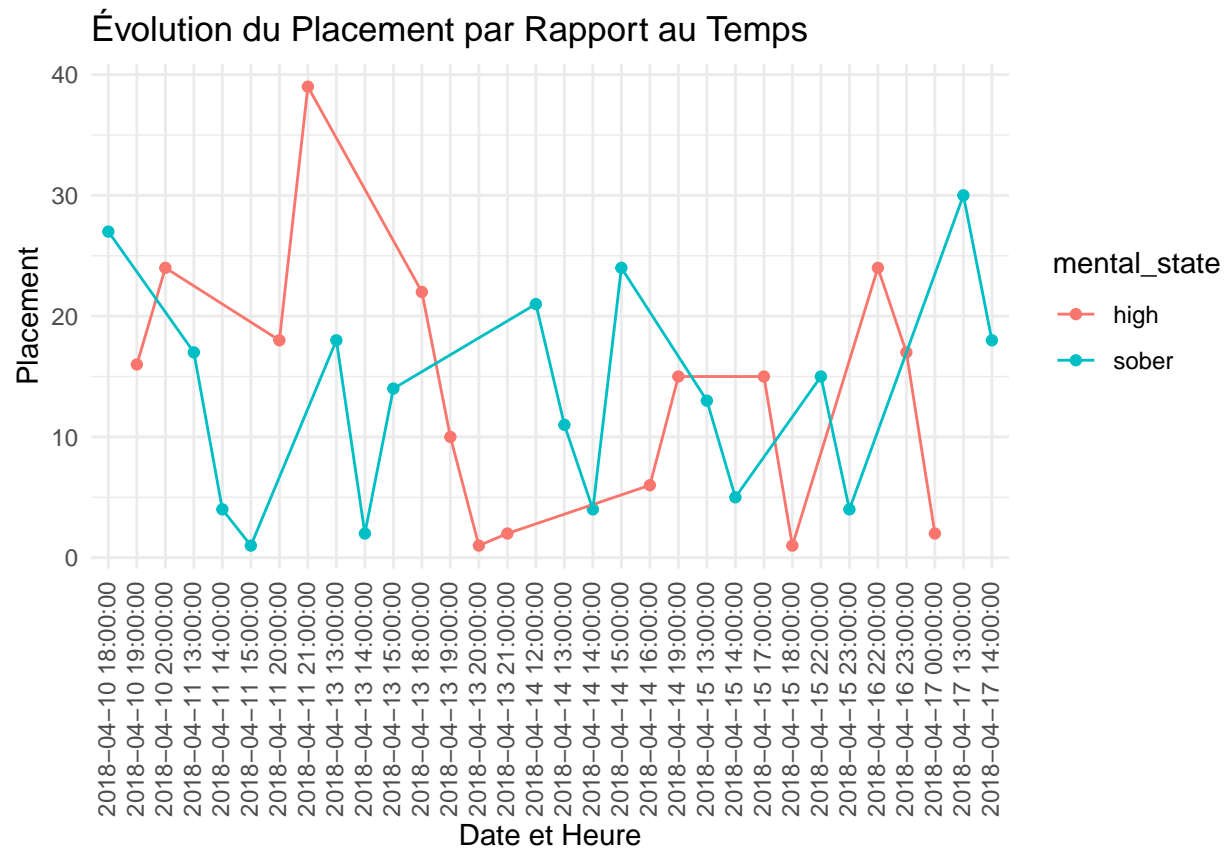
## Évolution du Placement en fonction de la Date et de l'Heure



On ne peut pas tracer la droite d'évolution car nous disposons uniquement de l'heure et pas des minutes. Si l'on voudrait tracer la droite d'évolution, il faudrait supprimer des valeurs, ou, faire un tri des données, ce qui serait beaucoup moins pertinent.

```
max_placement <- data %>%
  group_by(DateHeure) %>%
  summarise(max_placement = min(Placed),
            mental_state = first(`Mental State`))

# Étape 4: Tracer la courbe évolutive avec différentes couleurs pour chaque mental state
ggplot(max_placement, aes(x = DateHeure, y = max_placement, color = mental_state, group = mental_state))
  geom_point() +
  geom_line() +
  labs(title = "Évolution du Placement par Rapport au Temps",
       x = "Date et Heure",
       y = "Placement") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.4, hjust=1))
```



## Parti de Test

On va maintenant réaliser différents tests qui nous permettront de comparer les données entre les différents états mentaux ("high" et "sober").

### Test de comparaison de moyennes

Parmi les 9 variables, on en choisit 3 : Eliminations, Accuracy, Damage to Player On teste si la marijuana augmente ou non les performances de jeu

```
indicateurs <- c("Eliminations", "Accuracy", "Damage to Players")

resultat <- data.frame(Indicateur = indicateurs, p_valeur = NA)

for (i in seq_along(indicateurs)) {
  x <- data[data$Mental State == "high", indicateurs[i]]
  y <- data[data$Mental State == "sober", indicateurs[i]]

  # Test t unilatéral supérieur
  resultat_test <- t.test(x, y, alternative = "greater")

  resultat$p_valeur[i] <- resultat_test$p.value
}
```

```
print(resultat)
```

```
##          Indicateur  p_valeur
## 1      Eliminations 0.5326318
## 2          Accuracy 0.8408620
## 3 Damage to Players 0.7291757
```

Les pvalueur trouvées sont très grandes. Par exemple, pour la variable 'Damage to Players' la pvalueur trouvée est de 0.73, c'est à dire qu'avec des niveaux alpha < 73% on ne pourra pas rejeter H0. On n'a donc pas mis en évidence que la moyenne de 'Damage to Players' pour l'état mental 'high' est supérieure à celle de l'état mental 'sober'. De manière générale, pour les 3 indicateurs, on ne rejette pas H0, on n'a pas mis en évidence que fumer de la marijuana augmente la performance de jeu.

### Corrélation et test de corrélation

On s'intéresse maintenant à la corrélation. Pour ça, on va utiliser la fonction `corrplot` de la librairie `corrplot`, et de la fonction `cor` qui permet de calculer la corrélation entre 2 variables.

```
library(corrplot)
```

```
## Warning: le package 'corrplot' a été compilé avec la version R 4.3.2
```

```
## corrplot 0.92 loaded
```

```
variables <- c("high", "sober")

for (i in variables){
  datax <- data[data$`Mental State` == i, noms_variables ]

  cat("----- Résultat pour la variable :", i, "-----", "\n")

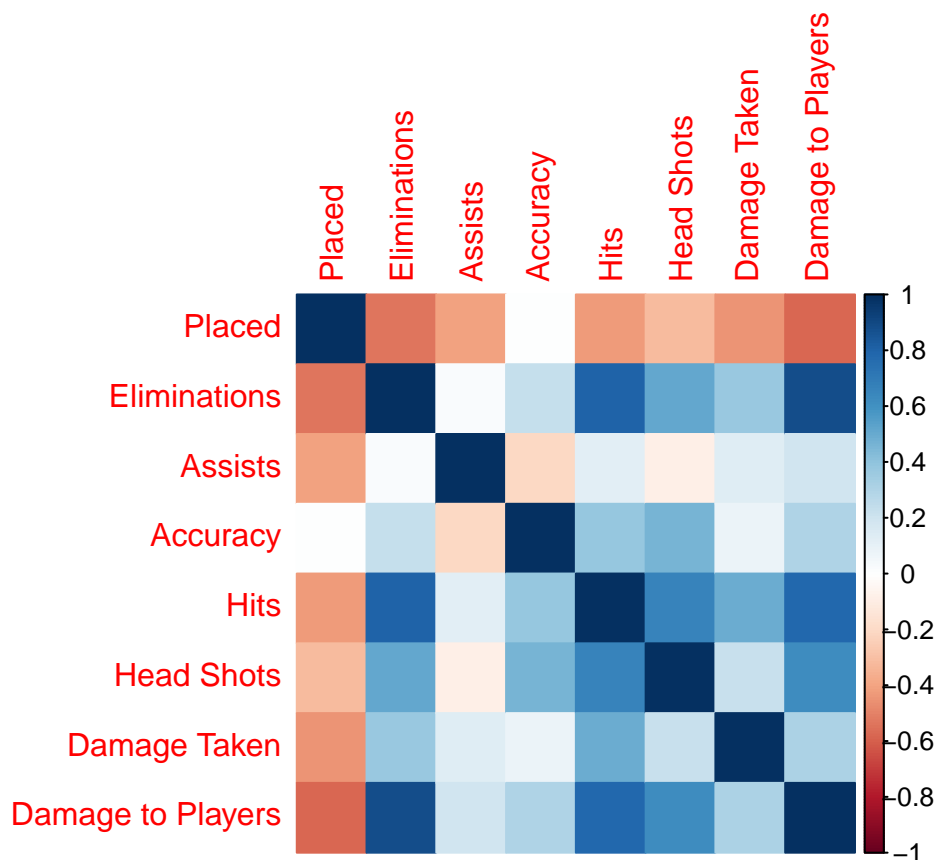
  corrplot(cor(datax), method="color")

  print(cor(datax, method = "pearson"))

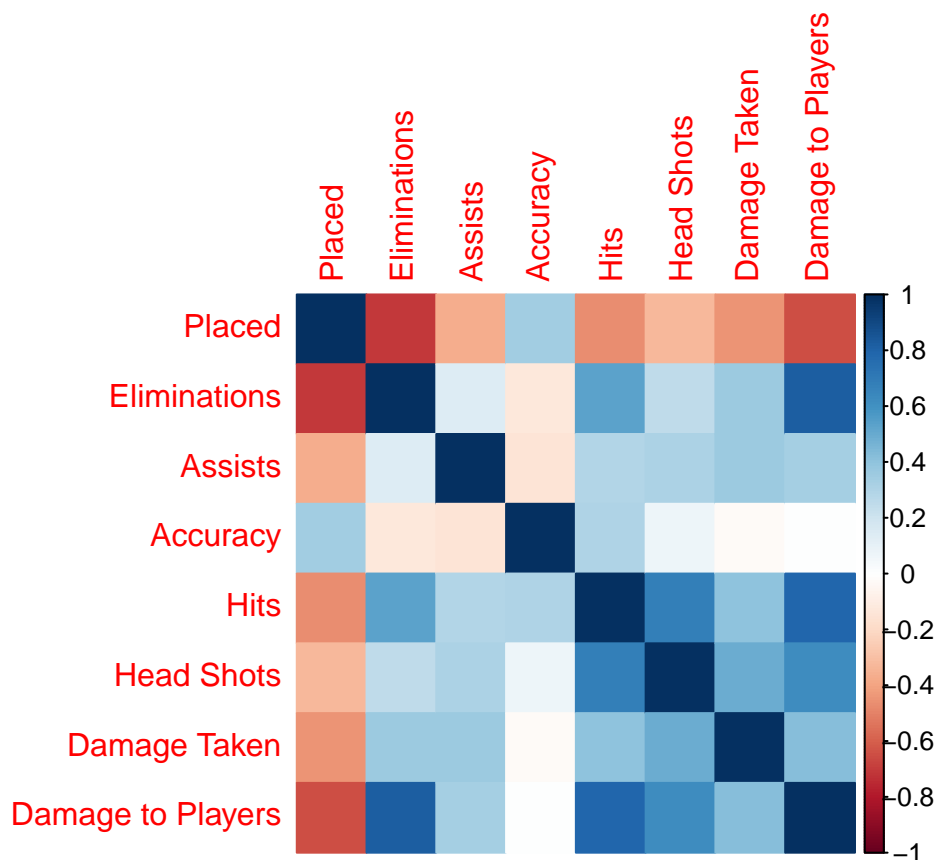
  cat("\n")
}
```

```
## ----- Résultat pour la variable : high -----
```





```
##          Placed Eliminations    Assists    Accuracy    Hits
## Placed      1.000000000 -0.5316330 -0.40283548  0.002200524 -0.4207654
## Eliminations -0.531633019  1.0000000  0.02165740  0.239766642  0.8066383
## Assists      -0.402835481  0.0216574  1.000000000 -0.204635934  0.1203533
## Accuracy      0.002200524  0.2397666 -0.20463593  1.000000000  0.3840000
## Hits         -0.420765376  0.8066383  0.12035331  0.384000010  1.0000000
## Head Shots   -0.315685461  0.5111840 -0.08528707  0.462237740  0.6692479
## Damage Taken -0.444979705  0.3792567  0.13506688  0.084653704  0.4958506
## Damage to Players -0.575434563  0.8897839  0.19504874  0.303282846  0.7810106
##
##          Head Shots Damage Taken Damage to Players
## Placed      -0.31568546 -0.4449797      -0.5754346
## Eliminations  0.51118397  0.3792567      0.8897839
## Assists      -0.08528707  0.1350669      0.1950487
## Accuracy      0.46223774  0.0846537      0.3032828
## Hits         0.66924793  0.4958506      0.7810106
## Head Shots    1.00000000  0.2210406      0.6261506
## Damage Taken  0.22104063  1.0000000      0.3185007
## Damage to Players 0.62615065  0.3185007      1.0000000
##
## ----- Résultat pour la variable : sober -----
```



##	Placed	Eliminations	Assists	Accuracy	Hits
## Placed	1.0000000	-0.7079099	-0.3691717	0.344867831	-0.4686511
## Eliminations	-0.7079099	1.0000000	0.1463706	-0.124440946	0.5357554
## Assists	-0.3691717	0.1463706	1.0000000	-0.147965564	0.2923738
## Accuracy	0.3448678	-0.1244409	-0.1479656	1.000000000	0.3013759
## Hits	-0.4686511	0.5357554	0.2923738	0.301375886	1.0000000
##	Head Shots	Damage Taken	Damage to Players		
## Placed	-0.32111770	-0.44339067	-0.645794650		
## Eliminations	0.25543203	0.36670528	0.825486461		
## Assists	0.31635320	0.36917873	0.331177461		
## Accuracy	0.07367078	-0.02854647	0.009212568		
## Hits	0.68287067	0.40755359	0.799171020		
## Head Shots	1.00000000	0.49117122	0.625906064		
## Damage Taken	0.49117122	1.00000000	0.425915387		
## Damage to Players	0.62590606	0.42591539	1.000000000		

### Test $\chi^2$ d'indépendance

On aimerait faire le test du  $\chi^2$ .

```
tableau <- table(data$`Mental State`, data$score)
tableau
```

```
##
##           TOP BON MOYEN MAUVAIS
##   high      7  23   10      2
##   sober      9  19   17      0
```

Le test du  $\chi^2$  repose selon l'hypothèse que les effectifs théoriques soient suffisamment grands (supérieur à 5). Ici, ce n'est pas le cas, donc l'approximation peut devenir inexacte, et le résultat du test peut ne pas être fiable.

```
chisq.test(tableau)
```

```
## Warning in chisq.test(tableau): L'approximation du Chi-2 est peut-être
## incorrecte
```

```
##
## Pearson's Chi-squared test
##
## data:  tableau
## X-squared = 4.3475, df = 3, p-value = 0.2263
```

La p-valeur associée au test du  $\chi^2$  est de 0.2263, ce qui est relativement élevé. Cela suggère qu'il n'y a pas suffisamment de preuves pour rejeter l'hypothèse nulle selon laquelle les variables 'Mental State' et 'score' sont indépendantes.

## Conclusion

**Rappel de la problématique :** *Est-ce que la marijuana a un impact sur les performance de jeu ?*

Nos analyses ne permettent pas de conclure que la performance en partie est améliorée sous l'emprise de la marijuana. En effet, pour le test de comparaison du  $\chi^2$ , nos résultats ne permettent pas de conclure sur l'indépendance des variables. De plus, avec les résultats obtenus, on ne peut pas dire qu'il inflige plus de dégâts en étant sous l'emprise de stupéfiants qu'en étant sobre.