Modèle linéaire multiple

Rapport écrit par:

Enzo LERICHE, Samuel DARMALINGON, Zakaria GHORAB

BUT Science des données FI EMS

2024-03-28

IUT de Paris - Rives de Seine
Université Paris Cité

Table des matières

Introduction

Présentation du projet

Consigne sur sujet : *L'objectif de ce projet est d'appliquer les connaissances acquises pendant le cours de modèle linéaire à utiliser sur R. Notre problématique est :*

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 précisement, il s'agit d'une personne qui a récolté ses données sur une période de 87 parties. Notre base de données contient 16 variables que nous pourrons vous présenter juste après.

Réponse à l'exercice

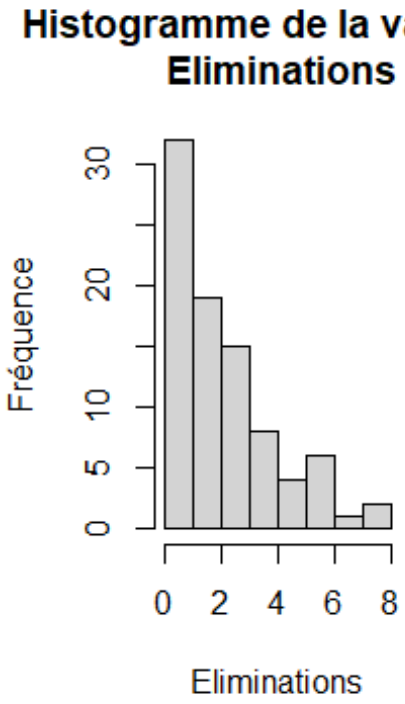
Importation des données

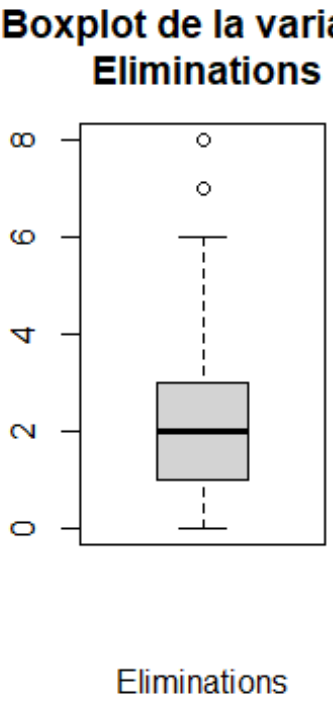
```
setwd("C:/Users/User/OneDrive/Documents/BUT/2eme annee/modèle linéaire")
data <- read_excel("fortnite_statistics.xlsx")
head(data, 3)
```

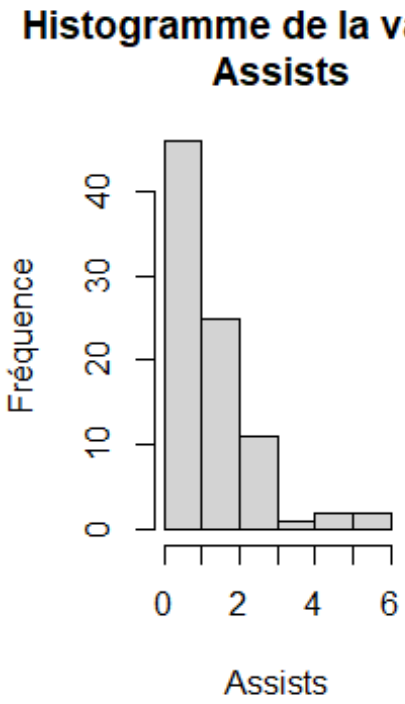
```
## # A tibble: 3 × 16
##   Date                `Time of Day`       Placed `Mental State`
Eliminations
##   <dttm>              <dttm>               <dbl> <chr>
<dbl>
## 1 2018-04-10 00:00:00 1899-12-31 18:00:00     27 sober
2
## 2 2018-04-10 00:00:00 1899-12-31 18:00:00     45 sober
1
## 3 2018-04-10 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>
```

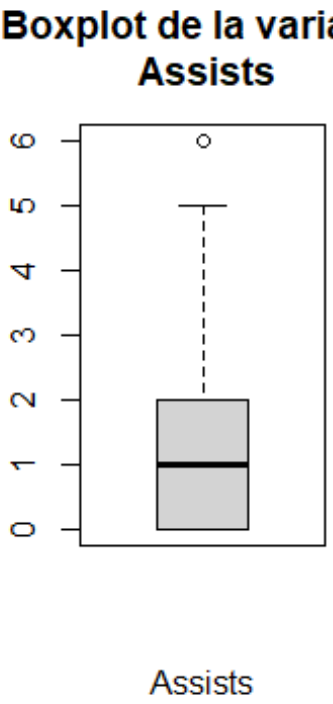
Étude descriptive univariée

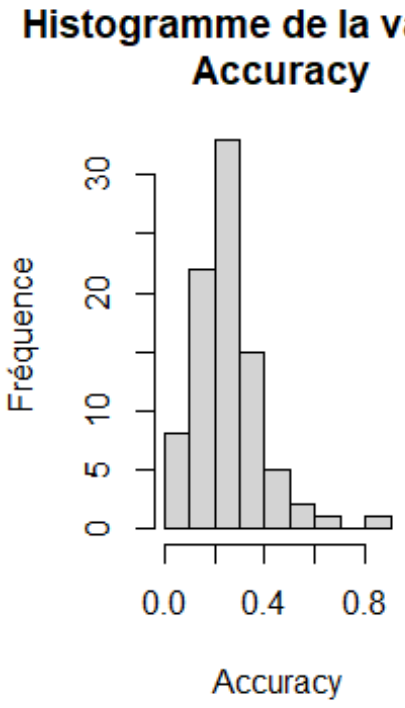
```
noms_variables <- c("Eliminations", "Assists", "Accuracy", "Hits", "Head
Shots", "Damage Taken", "Damage to Players")

par(mfrow=c(1,2))
for (i in noms_variables){
  hist(unlist(data[,i]),main=paste("Histogramme de la variable\n", i),
       xlab=i, ylab="Fréquence")
  boxplot(unlist(data[,i]),main=paste("Boxplot de la variable\n", i),
          xlab=i, ylab="")
}
```

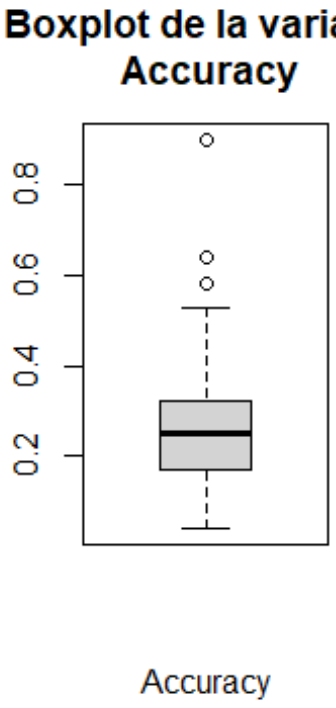

Histogramme de la varial Eliminations
Fréquence
0
5
10
20
30
0
2
4
6
8
Eliminations

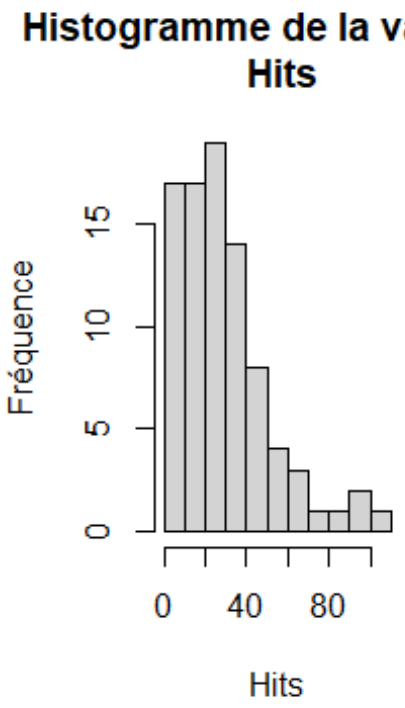

Boxplot de la variable Eliminations
0
2
4
6
8
Eliminations

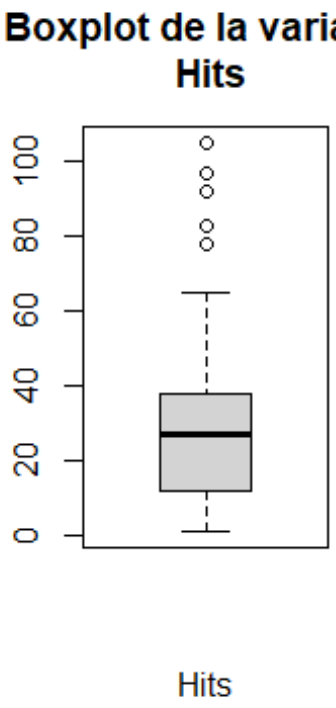

Histogramme de la varial Assists
Fréquence
0
10
20
30
40
0
2
4
6
Assists

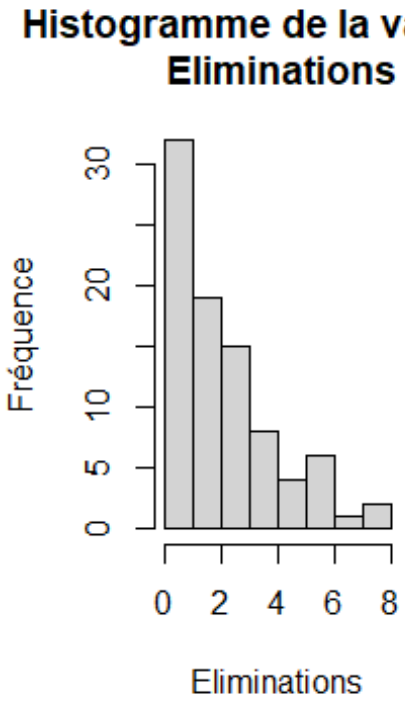

Boxplot de la variable Assists
0
1
2
3
4
5
6
Assists

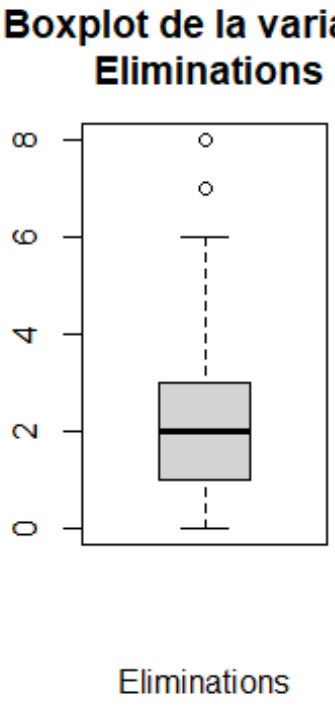

Histogramme de la varial
Accuracy
Fréquence
0
5
10
20
30
0.0
0.4
0.8
Accuracy

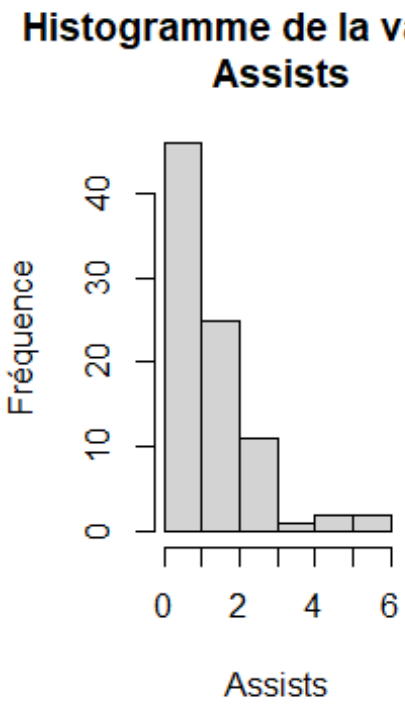

Boxplot de la variable
Accuracy
0.2
0.4
0.6
0.8
Accuracy

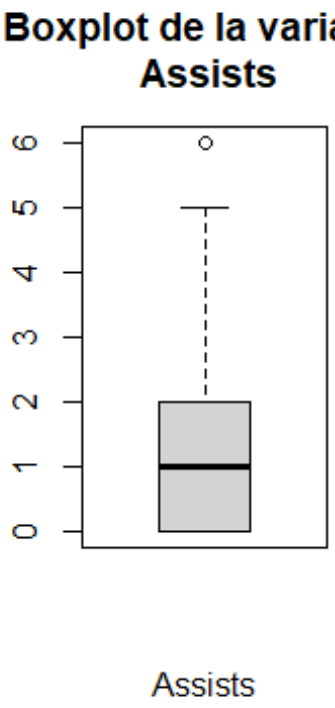

Histogramme de la varial
Hits
Fréquence
0
5
10
15
0
40
80
Hits

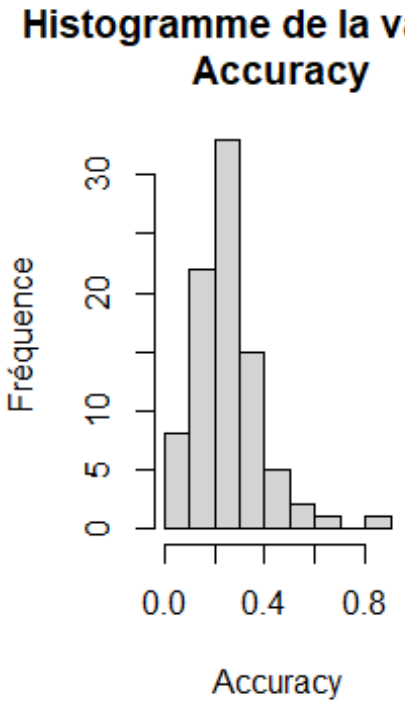

Boxplot de la variable
Hits
0
20
40
60
80
100
Hits

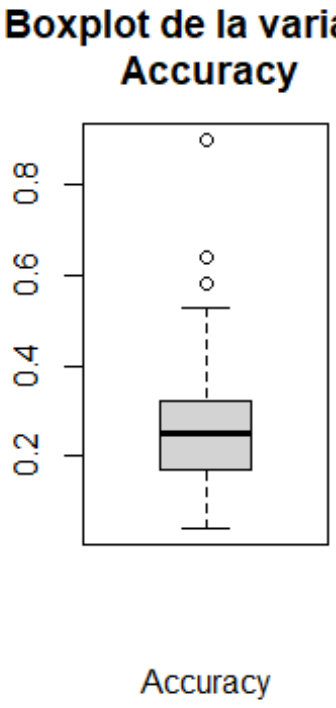

Histogramme de la varial Eliminations
Fréquence
0
5
10
20
30
0
2
4
6
8
Eliminations

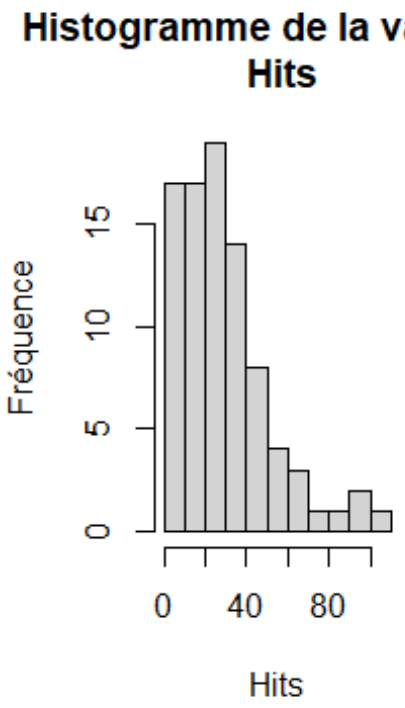

Boxplot de la variable Eliminations
0
2
4
6
8
Eliminations

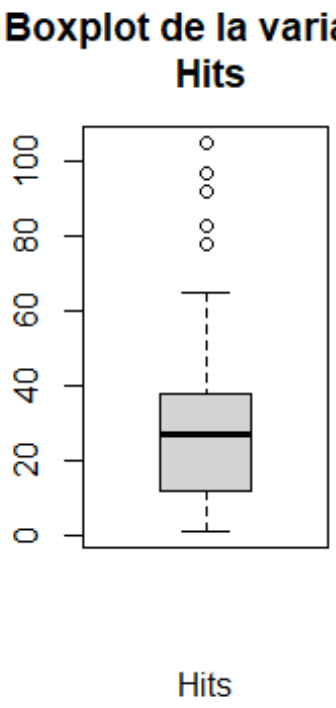

Histogramme de la varial Assists
Fréquence
0
10
20
30
40
0
2
4
6
Assists

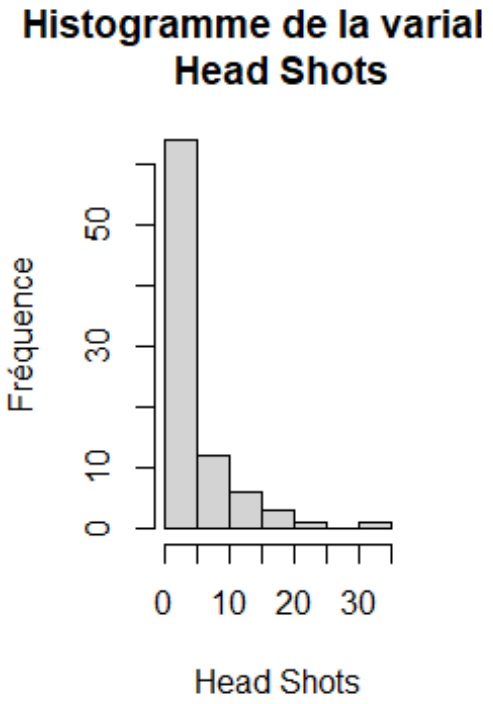

Boxplot de la variable Assists
0
1
2
3
4
5
6
Assists

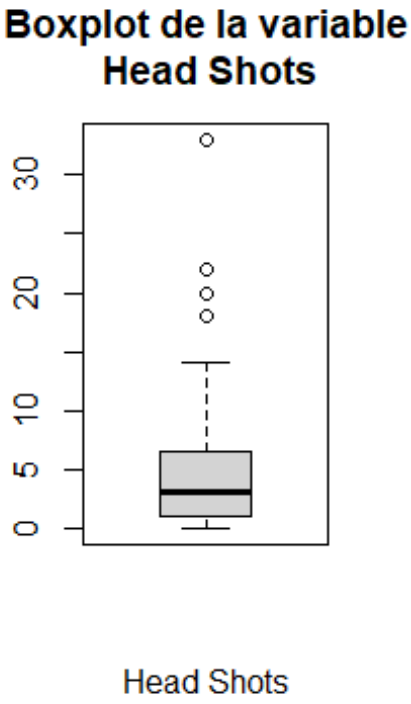

Histogramme de la varial
Accuracy
Fréquence
0
5
10
20
30
0.0
0.4
0.8
Accuracy

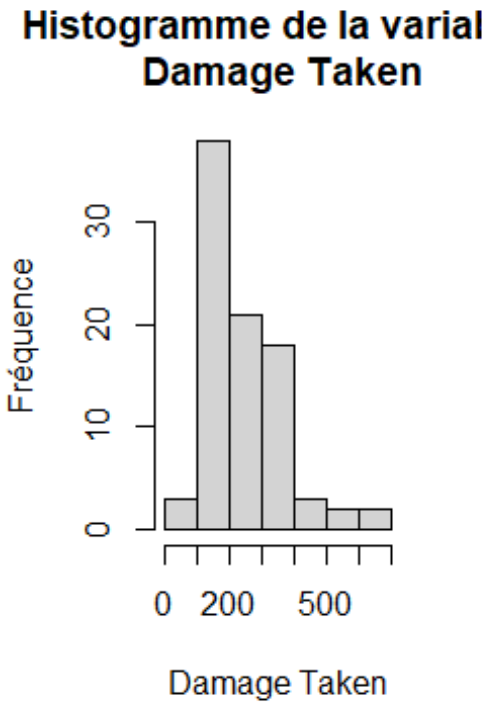

Boxplot de la variable
Accuracy
0.2
0.4
0.6
0.8
Accuracy

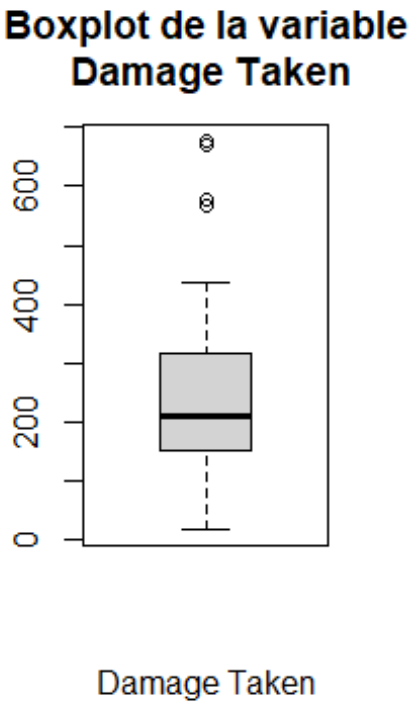

Histogramme de la varial
Hits
Fréquence
0
5
10
15
0
40
80
Hits

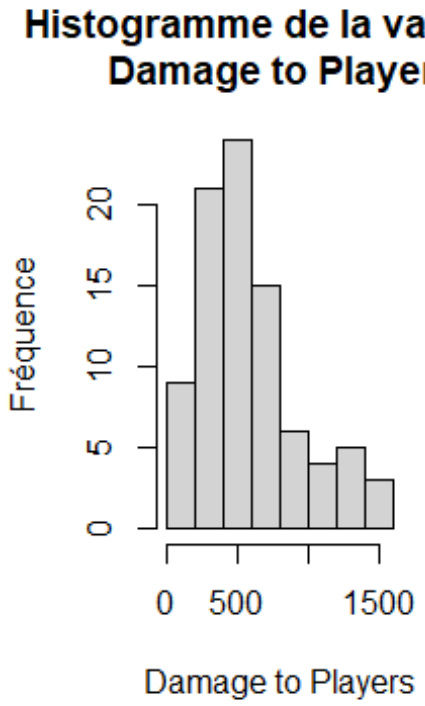

Boxplot de la variable
Hits
0
20
40
60
80
100
Hits

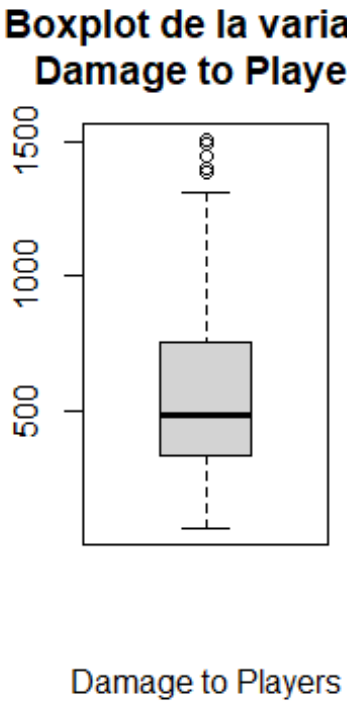

Histogramme de la varial
Head Shots
Fréquence
0
10
30
50
0
10
20
30
Head Shots


Boxplot de la variable
Head Shots
0
5
10
20
30
Head Shots


Histogramme de la varial
Damage Taken
Fréquence
0
10
20
30
0
200
500
Damage Taken

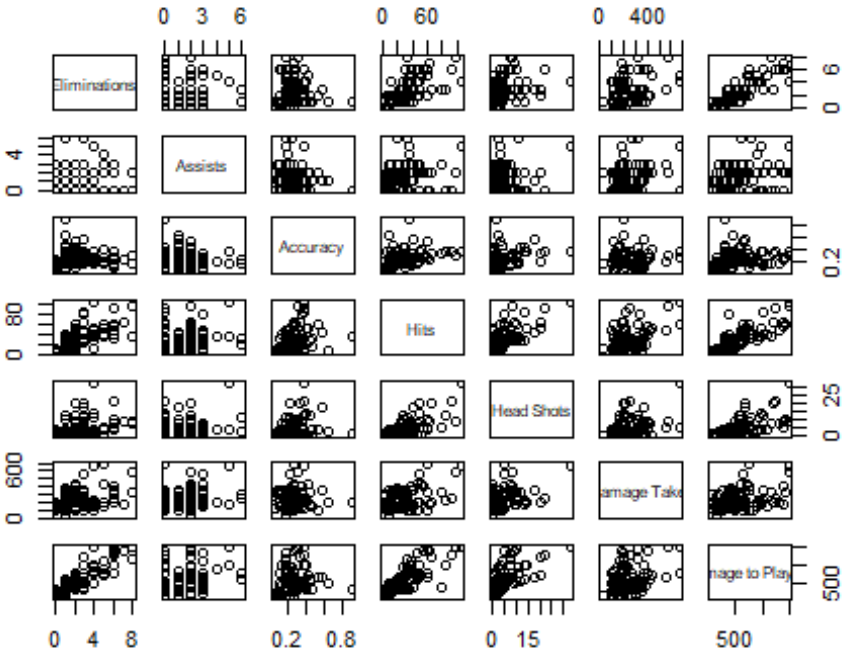

Boxplot de la variable
Damage Taken
0
200
400
600
Damage Taken





Matrice de corrélation

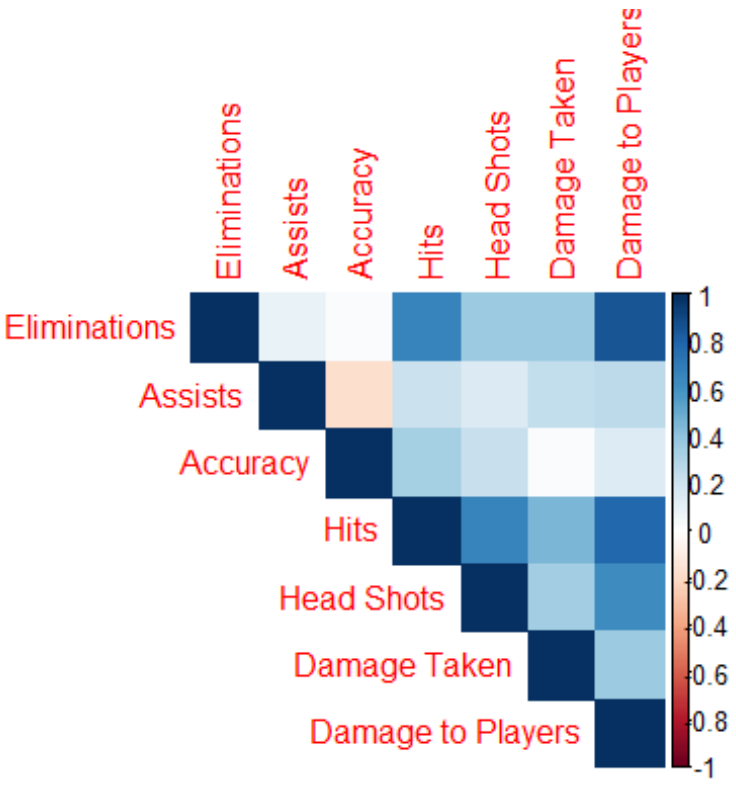
```
fortnite <- data[,noms_variables]
plot(fortnite)
```



```r
cor(fortnite)
```

```
##                   Eliminations     Assists    Accuracy      Hits Head
Shots
## Eliminations        1.00000000  0.09452124  0.02173760 0.6659952
0.3633493
## Assists             0.09452124  1.00000000 -0.17067214 0.2112163
0.1574874
## Accuracy            0.02173760 -0.17067214  1.00000000 0.3311535
0.2254906
## Hits                0.66599523  0.21121625  0.33115347 1.0000000
0.6695894
## Head Shots          0.36334927  0.15748735  0.22549061 0.6695894
1.0000000
## Damage Taken        0.36904282  0.24818386  0.02630038 0.4566420
0.3496391
## Damage to Players   0.85315123  0.26580156  0.14019576 0.7881844
0.6239366
##                   Damage Taken Damage to Players
## Eliminations        0.36904282         0.8531512
## Assists             0.24818386         0.2658016
## Accuracy            0.02630038         0.1401958
## Hits                0.45664198         0.7881844
## Head Shots          0.34963907         0.6239366
## Damage Taken        1.00000000         0.3653123
## Damage to Players   0.36531231         1.0000000
```

```r
corrplot(cor(fortnite), method="color", type="upper")
```

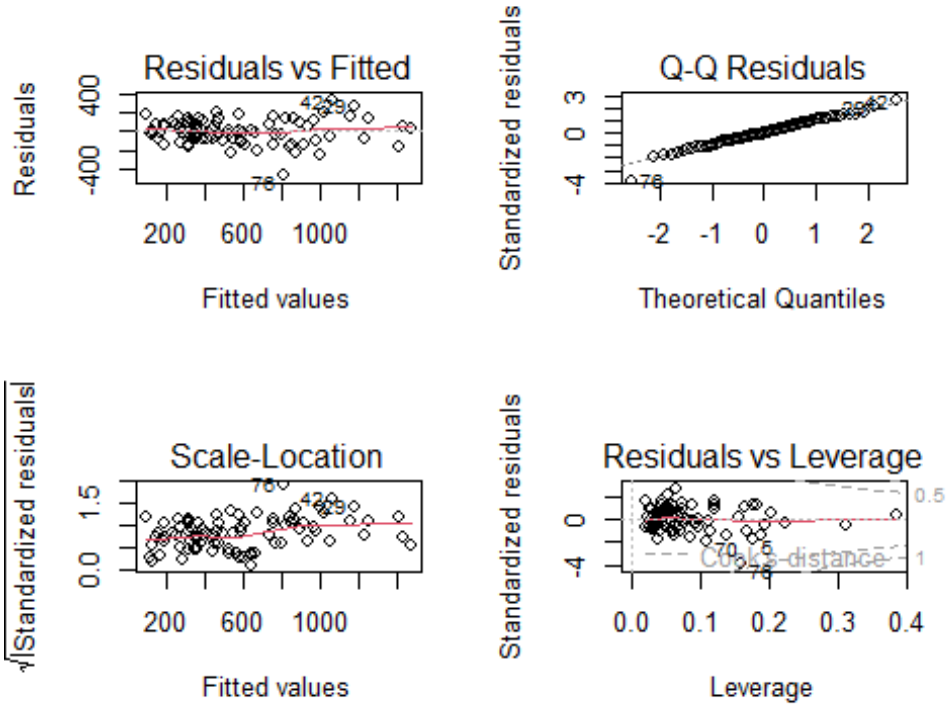


La matrice des corrélations met en évidence plusieurs relations entre les différentes variables de notre ensemble de données Fortnite. En se concentrant sur la variable "Damage to Players", on observe qu'elle présente la corrélation la plus élevée avec "Eliminations" (0.853). Cela suggère qu'il existe une forte association entre les dégâts infligés aux joueurs et le nombre d'éliminations réalisées.

En regardant plus globalement la matrice des corrélations, on constate que les autres variables ne présentent pas de corrélations aussi fortes entre elles. Par exemple, les corrélations entre "Assists" et les autres variables sont relativement faibles, avec des coefficients allant de -0.17 à 0.24.

Construction du modèle linéaire multiple

```
mod <- lm(formula= `Damage to Players` ~., data=fortnite)
par(mfrow=c(2,2))
plot(mod)
```




```
summary(mod)

##
## Call:
## lm(formula = `Damage to Players` ~ ., data = fortnite)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -449.18  -77.07   -4.56   87.09  317.44
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     85.2123    45.4313   1.876 0.064354 .
## Eliminations   125.7025    10.5595  11.904  < 2e-16 ***
## Assists         39.8324    10.8317   3.677 0.000425 ***
## Accuracy       112.0133   119.2945   0.939 0.350575
## Hits             2.4010     1.2067   1.990 0.050039 .
## `Head Shots`    16.9561     3.2807   5.168  1.7e-06 ***
## `Damage Taken`  -0.2455     0.1283  -1.914 0.059187 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 128.6 on 80 degrees of freedom
## Multiple R-squared:  0.8774, Adjusted R-squared:  0.8682
## F-statistic: 95.39 on 6 and 80 DF,  p-value: < 2.2e-16
```

Les hypothèses du modèle linéaire sont validées.

Dans la sortie R, nous avons les estimations de nos paramètres du modèle:

- b = 85.2123
- Eliminations = 125.7025
- Assists = 39.8324
- Accuracy = 112.0133
- Hits = 2.4010
- Head Shots = 16.9561
- Damage Taken = -0.2455

Nous allons maintenant tester un à un si ces paramètres d'éspérance sont nuls ou non en prenant en compte la présence des autres variables :

- b : pvaleur = 0.064354, on ne rejette pas H0, on n'a pas mis en évidence que b est différent de 0 au risque 5%.
- Eliminations : pvaleur < 2e-16, on rejette H0, on a mis en évidence que le paramètre "Eliminations" était différent de 0 au risque 5%.
- Assists : pvaleur = 0.000425, on rejette H0, on a mis en évidence que le paramètre "Assists" était différent de 0 au risque 5%.
- Head Shots : pvaleur = 1.7e-06, on rejette H0, on a mis en évidence que le paramètre "Head Shots" était différent de 0 au risque 5%.

Cela veux dire que que lorque le joueur va effectuer des éliminations, asssiter aux éliminations et faire des tirs dans la tête, il va infligé des dommages aux autres joueurs avec un risuqe de de tromper de 5%.

- Damage Taken : pvaleur = 0.059187, on ne rejette pas H0, on n'a pas mis en évidence que le paramètre "Damage Taken" est différent de 0 au risque 5%.
- Accuracy : pvaleur = 0.350575, on ne rejette pas H0, on n'a pas mis en évidence que le paramètre "Accuracy" est différent de 0 au risque 5%.
- Hits : pvaleur = 0.050039, on ne rejette pas H0, on n'a pas mis en évidence que le paramètre "Hits" est différent de 0.

Cela veut dire que lorsque le joueur va recevoir des dégâts, qu'il va tirer, et qu'on va mesurer son pourcentage de tir. Cela ne signifie qu'il ne va pas infligé des dommages aux autres joueurs avec un risque de de tromper de 5%.

On va maintenant tester au risque 5% la contribution globale des variables explicatives sur le nombre de dégât infligés aux joueurs ce qui correspond au test de Fisher. Le sortie R nous

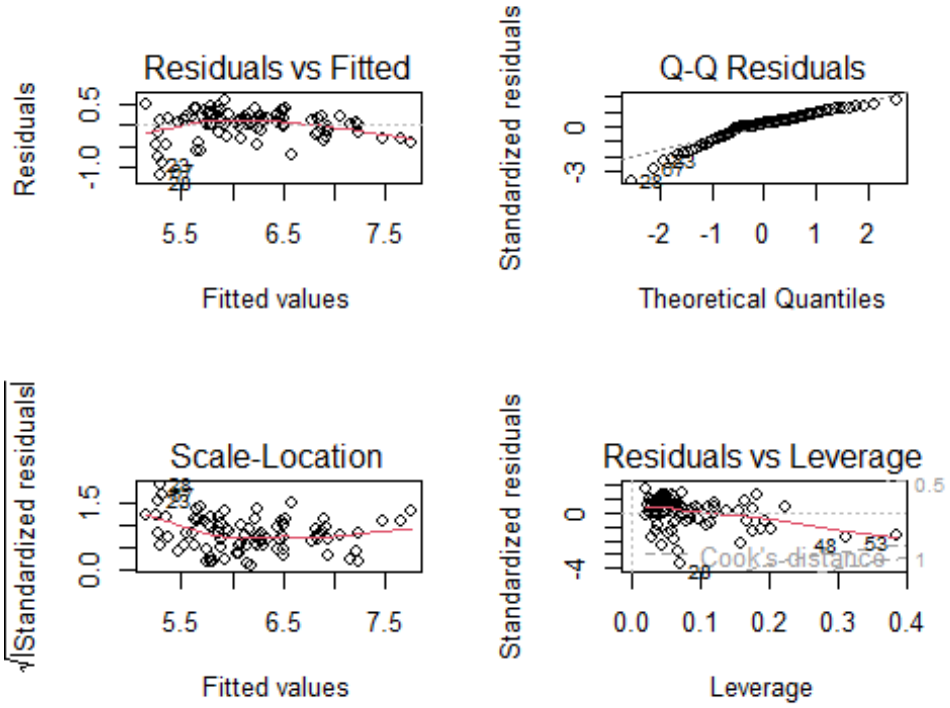
donne une pvaleur < 2.2e-16, ce qui veut dire que au moins une des variables du modèle sert à expliquer le nombre de dégât infligés aux joueurs.

La sortie R nous montre un R^2 très grand, égale à 0.8774. Mais on sait que le R^2 n'est pas un critère très intéressant car il augmente quand le nombre de paramètre augmente. On va donc regarder le R^2 ajustés qui lui ne prend pas en compte le nombre de paramètre. Ici, le R^2 ajustés est égal à 0.8682, c'est à dire que 0.8682 soit 86,82% de la variabilité de Y (le nombre de dégât infligés aux joueurs) est expliqué par notre modèle.

Notre modèle est plus intéressant que pas de modèle, mais certaines variables semble moins intéressante à garder. Il faudrait faire une selection de variable pour faire un meilleur modèle.

Construction du modèle linéaire multiple pour log(y+1)

```
modlog <- lm(formula= log(`Damage to Players`+1) ~., data=fortnite)
par(mfrow=c(2,2))
plot(modlog)
```




```
summary(modlog)

##
## Call:
## lm(formula = log(`Damage to Players` + 1) ~ ., data = fortnite)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.18241 -0.16324  0.05724  0.19867  0.57015
```

```
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     5.0526093  0.1188663  42.507  < 2e-16 ***
## Eliminations    0.2352832  0.0276279   8.516 7.85e-13 ***
## Assists         0.1099187  0.0283400   3.879 0.000214 ***
## Accuracy        1.0411662  0.3121222   3.336 0.001292 **
## Hits            0.0015631  0.0031572   0.495 0.621899
## `Head Shots`    0.0263177  0.0085837   3.066 0.002958 **
## `Damage Taken` -0.0003282  0.0003356  -0.978 0.331097
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3365 on 80 degrees of freedom
## Multiple R-squared:  0.7689, Adjusted R-squared:  0.7515
## F-statistic: 44.35 on 6 and 80 DF,  p-value: < 2.2e-16
```

Les hypothèses du modèle linéaire sont validées.

Dans la sortie R, nous avons les estimations de nos paramètres du modèle:

- b = 5.0526093
- Eliminations = 0.2352832
- Assists = 0.1099187
- Accuracy = 1.0411662
- Hits = 0.0015631
- Head Shots = 0.0263177
- Damage Taken = -0.0003282

Nous allons maintenant tester un à un si ces paramètres d'éspérances sont nuls ou non en prenant en compte la présence des autres variables :

- b : pvaleur < 2e-16, on rejette H0, on a mis en évidence que le paramètre "Eliminations" était différent de 0.
- Eliminations : pvaleur = 7.85e-13, on rejette H0, on a mis en évidence que le paramètre "Eliminations" était différent de 0.
- Assists : pvaleur = 0.000214, on rejette H0, on a mis en évidence que le paramètre "Assists" était différent de 0.
- Accuracy : pvaleur = 0.001292, on rejette H0, on a mis en évidence que le paramètre "Assists" était différent de 0.
- Head Shots : pvaleur = 0.002958, on rejette H0, on a mis en évidence que le paramètre "Head Shots" était différent de 0.

Cela veux dire que que lorque le joueur va effectuer des éliminations, asssiter aux éliminations, le pourcentage du nombre de tir du joueur et faire des tirs dans la tête, il va infligé des dommages aux autres joueurs avec un risuqe de de tromper de 5%.

- Damage Taken : pvaleur = 0.331097, on ne rejette pas H0, on n'a pas mis en évidence que le paramètre "Damage Taken" est différent de 0.
- Hits : pvaleur = 0.621899, on ne rejette pas H0, on n'a pas mis en évidence que le paramètre "Hits" est différent de 0.

Cela veut dire que lorsque le joueur va recevoir des dégâts,et qu'il va tirer, il ne va pas infligé des dommages aux autres joueurs avec un risuqe de de tromper de 5%.

On va maintenant tester au risque 5% la contribution globale des variables explicatives sur le nombre de dégât infligés aux joueurs ce qui correspond au test de Fisher. Le sortie R nous donne une pvaleur < 2.2e-16, ce qui veut dire que au moins une des variables du modèle sert à expliquer le nombre de dégât infligés aux joueurs.

La sortie R nous montre un R^2 très grand, égale à 0.7689. Mais on sait que le R^2 n'est pas un critère très intéressant car il augmente quand le nombre de paramètre augmente. On va donc regarder le R^2 ajustés qui lui ne prend pas en compte le nombre de paramètre. Ici, le R^2 ajustés est égal à 0.7515, c'est à dire que 0.8682 soit 86,82% de la variabilité de Y (le nombre de dégât infligés aux joueurs) est expliqué par notre modèle.

Notre modèle est plus intéressant que pas de modèle, mais certaines variables semble moins intéressante à garder. Il faudrait faire une selection de variable pour faire un meilleur modèle.

Pour comparer les deux modèles, le plus interessant semble le premier modèle (mod). En effet, il a un meilleur R^2 et R^2 ajustés que le modèle 2 (modlog). Par contre les conclusions semblent pas vraiment si différentes entre les 2 modèles. Dans les 2 modèles la conclusion finale était que au moins une des variables sert à expliquer le nombre de dégât infligés aux joueurs et que notre modèle était plus intéressant que pas de modèle.

Selection de variable

Nombre de modèle possible à comparer

Pour savoir le nombre de modèle à tester il faut calculé 2^p ou p représente le nombre de variables.

Ici on a p = 6.

Donc :

Avec leaps

Nous allons maintenant utiliser leaps pour pouvoir représenter les meilleurs modèles en fonction des différents critères vu en cours.

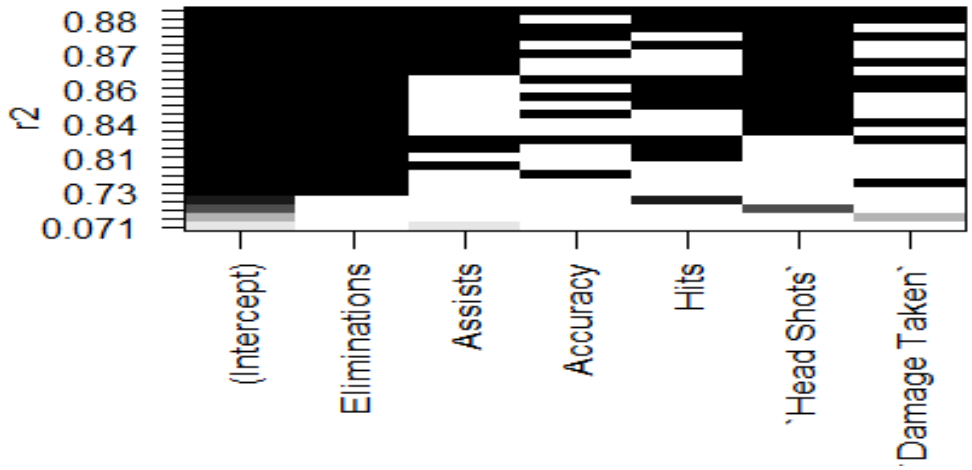
Nous allons utiliser les 3 critères suivants :

- R^2

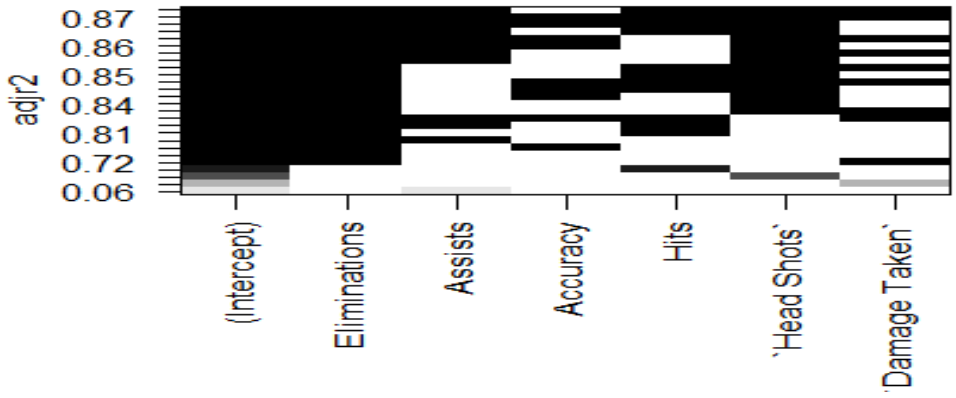
-R^2 ajusté

- BIC

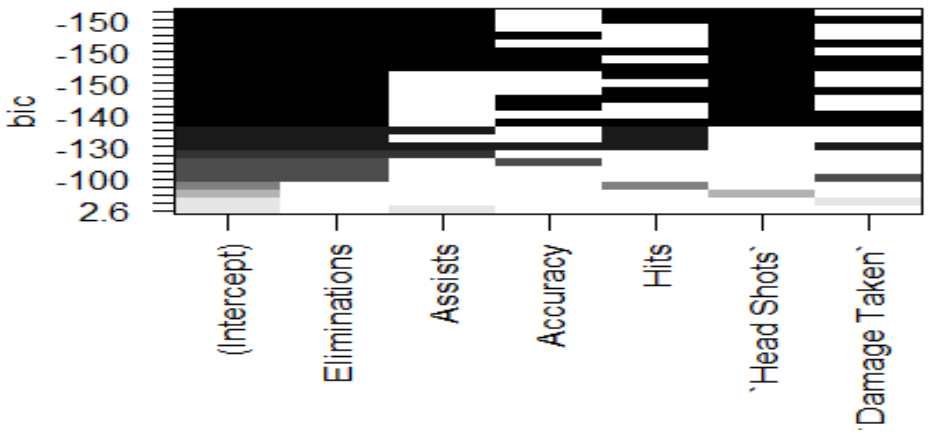
```
a <- regsubsets(`Damage to Players`~., data = fortnite, method =
"exhaustive", nbest = 5)

par(mfrow=c(1,1))
plot(a, scale = "r2")
```




```
plot(a, scale = "adjr2")
```



```
plot(a, scale = "bic")
```



Avec le R^2, le modèle qui est considéré comme le meilleur contient toutes les variables, c'est-à-dire: Eliminations, Assists, Accuracy, Hits, Head Shots, Damage Taken.

Avec le R^2 ajusté le modèle qui est considéré comme le meilleur contient les variables suivantes : Eliminations, Assists, Hits, Head Shots, Damage Taken.

Avec le BIC le modèle qui est considéré comme le meilleur contient les variables suivantes : Eliminations, Assists, Hits, Head Shots.

On passe donc d'un modèle qui garde 6 variables avec le R^2 à un modèle qui garde 4 variables avec le BIC.

Selection ascendante (Forward)

Avec AIC :

```
m1 <- lm(`Damage to Players`~1, data = fortnite)
step(m1,scope=list(upper=~Eliminations+Assists+Accuracy+Hits+`Head
Shots`+`Damage Taken`),
     direction="forward", trace=FALSE, k = 2)

##
## Call:
## lm(formula = `Damage to Players` ~ Eliminations + `Head Shots` +
##     Assists + Hits + `Damage Taken`, data = fortnite)
##
## Coefficients:
##    (Intercept)    Eliminations    `Head Shots`         Assists
Hits
##       114.6386        122.7004         16.8624         37.0351
2.8749
```

```
## `Damage Taken`
##        -0.2547
```

En faisant la selection ascendante nous trouvons le meilleur modèle qui contient les variables suivantes: Eliminations, Head Shots, Assists, Hits, Damage Taken.

Avec BIC :

```
n <- nrow(data)
step(m1,scope=list(upper=~Eliminations+Assists+Accuracy+Hits+`Head
Shots`+`Damage Taken`),
     direction="forward", trace=F, k = log(n))

##
## Call:
## lm(formula = `Damage to Players` ~ Eliminations + `Head Shots` +
##     Assists + Hits, data = fortnite)
##
## Coefficients:
##  (Intercept)  Eliminations  `Head Shots`       Assists          Hits
##       77.878       120.260        16.331        33.272         2.499
```

En faisant la selection ascendante nous trouvons le meilleur modèle qui contient les variables suivantes: Eliminations, Head Shots, Assists, Hits.

Selection descendante (Backward)

Avec AIC :

```
step(mod,direction="backward", trace=F, k = 2)

##
## Call:
## lm(formula = `Damage to Players` ~ Eliminations + Assists + Hits +
##     `Head Shots` + `Damage Taken`, data = fortnite)
##
## Coefficients:
##    (Intercept)    Eliminations         Assists            Hits    `Head
Shots`
##       114.6386        122.7004         37.0351          2.8749
16.8624
## `Damage Taken`
##        -0.2547
```

En faisant la selection descendante nous trouvons le meilleur modèle qui contient les variables suivantes: Eliminations, Head Shots, Assists, Hits, Damage Taken.

Avec BIC :

```
step(mod,direction="backward", trace=F, k = log(n))
```

```
##
## Call:
## lm(formula = `Damage to Players` ~ Eliminations + Assists + Hits +
##     `Head Shots`, data = fortnite)
##
## Coefficients:
##  (Intercept)  Eliminations       Assists          Hits  `Head Shots`
##       77.878       120.260        33.272         2.499        16.331
```

En faisant la selection descendante nous trouvons le meilleur modèle qui contient les variables suivantes: Eliminations, Head Shots, Assists, Hits.

Methode Stepwise

Avec AIC :

```
step(m1, scope=list(upper=~Eliminations+Assists+Accuracy+Hits+`Head
Shots`+`Damage Taken`),
     direction="both", trace=F, k = 2)
```

```
##
## Call:
## lm(formula = `Damage to Players` ~ Eliminations + `Head Shots` +
##     Assists + Hits + `Damage Taken`, data = fortnite)
##
## Coefficients:
##    (Intercept)    Eliminations    `Head Shots`         Assists
Hits
##       114.6386        122.7004         16.8624         37.0351
2.8749
## `Damage Taken`
##        -0.2547
```

En faisant la selection étape par étape avec stepwise nous trouvons le meilleur modèle qui contient les variables suivantes: Eliminations, Head Shots, Assists, Hits, Damage Taken.

Avec BIC :

```
step(m1, scope=list(upper=~Eliminations+Assists+Accuracy+Hits+`Head
Shots`+`Damage Taken`),
     direction="both", trace=F, k = log(n))
```

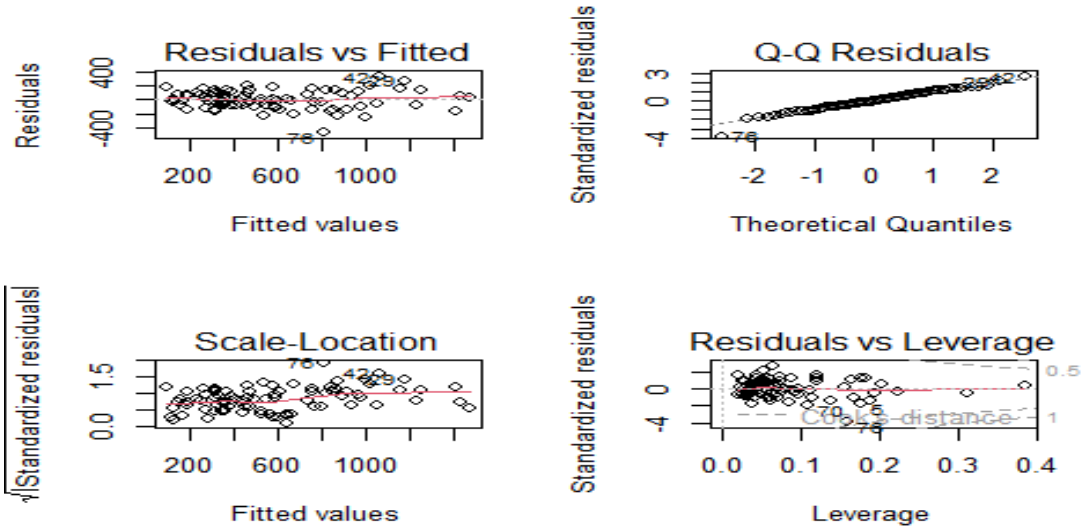
```
##
## Call:
## lm(formula = `Damage to Players` ~ Eliminations + `Head Shots` +
##     Assists + Hits, data = fortnite)
##
## Coefficients:
##  (Intercept)  Eliminations  `Head Shots`       Assists          Hits
##       77.878       120.260        16.331        33.272         2.499
```

En faisant la selection étape par étape avec stepwise nous trouvons le meilleur modèle qui contient les variables suivantes: Eliminations, Head Shots, Assists, Hits.

Verification des hyphothèses pour les modèles choisis

Modèle complet

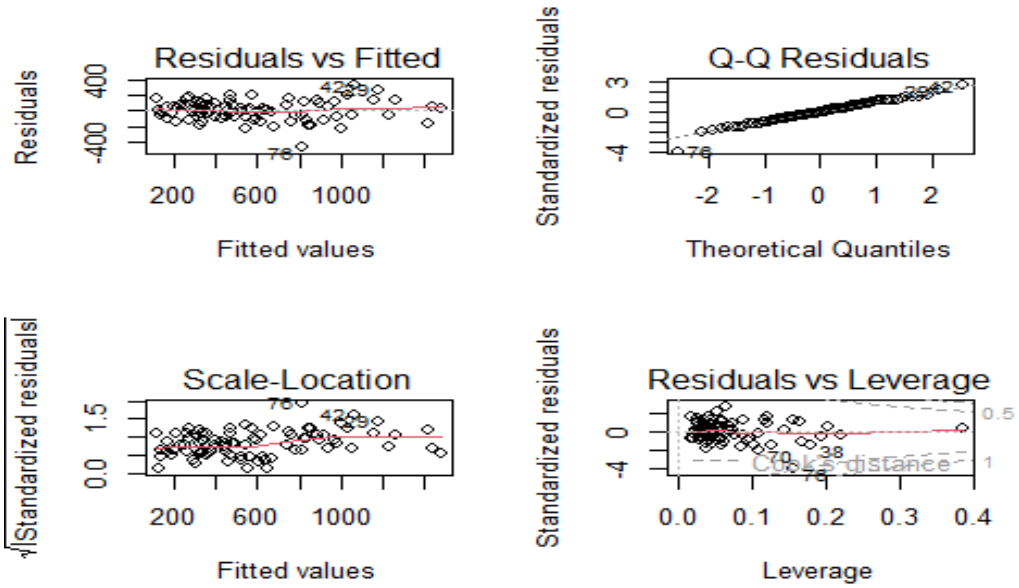
```
mod <- lm(formula= `Damage to Players` ~., data=fortnite)
par(mfrow=c(2,2))
plot(mod)
```



Les hyptohèses sont vérifiées pour le modèle complet.

Modèle AIC

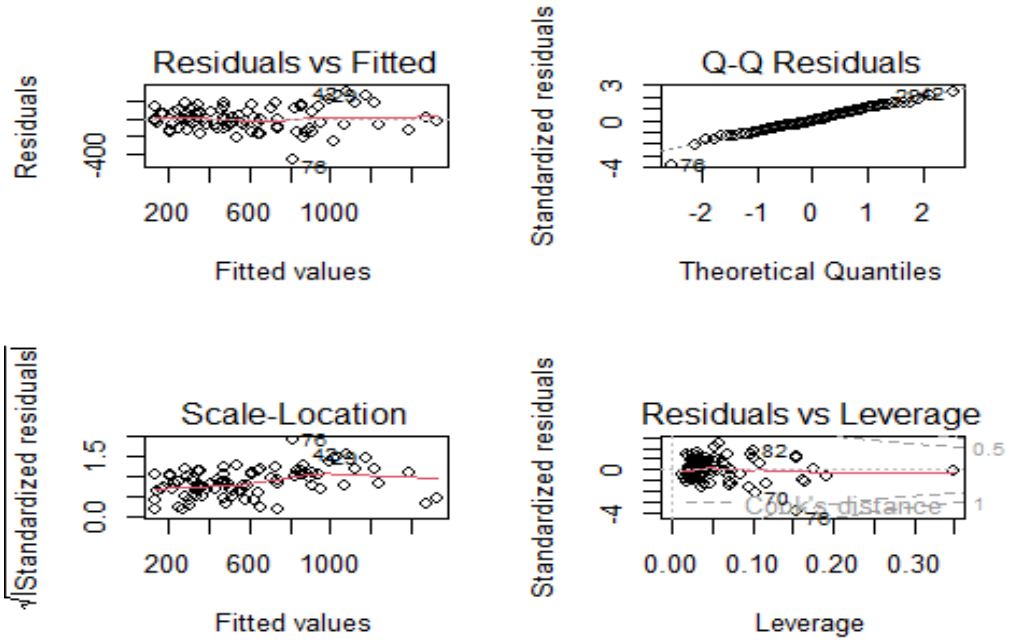
```
mod_retenu_aic <- lm(`Damage to Players`~Eliminations+`Head
Shots`+Assists+Hits+`Damage Taken`, data = fortnite)
par(mfrow=c(2,2))
plot(mod_retenu_aic)
```



Les hyptohèses sont vérifiées pour le modèle AIC.

Modèle BIC

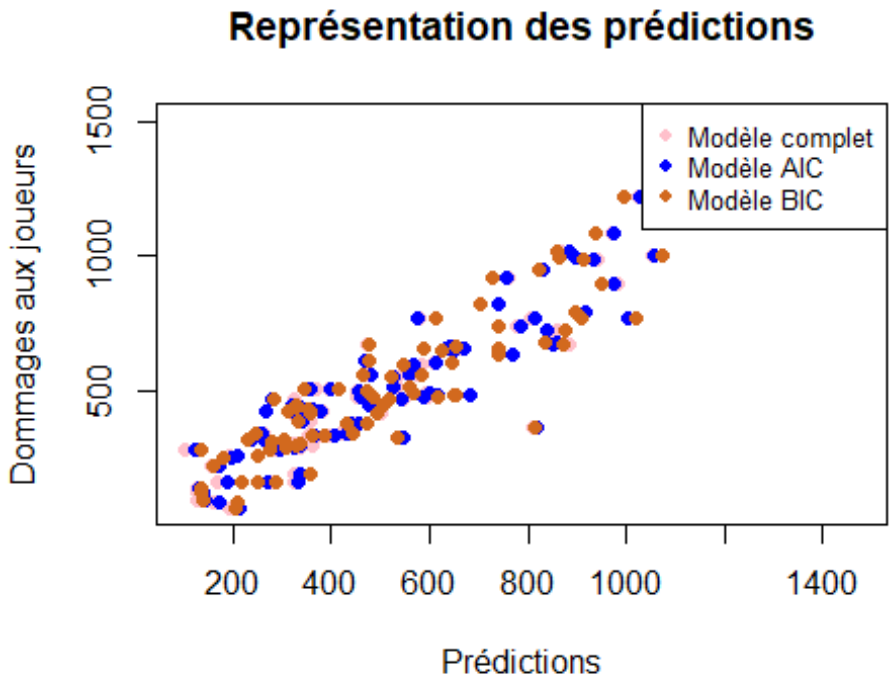
```
mod_retenu_bic<- lm(`Damage to Players`~Eliminations+`Head
Shots`+Assists+Hits, data = fortnite)
par(mfrow=c(2,2))
plot(mod_retenu_bic)
```



Les hyptohèses sont vérifiées pour le modèle BIC.

Représentation des predictions

```
par(mfrow=c(1,1))
predict_modele_complet <- predict(mod, fortnite,interval="prediction",
level=0.95)[,1]
predict_modele_aic <- predict(mod_retenu_aic,fortnite,interval="prediction",
level=0.95)[,1]
predict_modele_bic <- predict(mod_retenu_bic,fortnite,interval="prediction",
level=0.95)[,1]

legend_labels <- c("Modèle complet", "Modèle AIC", "Modèle BIC")

plot(predict_modele_complet, fortnite$`Damage to Players`, col="pink",
pch=16,
     xlab="Prédictions", ylab="Dommages aux joueurs",
     main="Représentation des prédictions")
points(predict_modele_aic, fortnite$`Damage to Players`, col="blue", pch=16)
points(predict_modele_bic, fortnite$`Damage to Players`, col="chocolate",
pch=16)
legend("topright", legend=legend_labels, col=c("pink", "blue", "chocolate"),
pch=16, cex=0.8)
```



De manière générale que ce soit pour le modèle AIC ou BIC, les points de prédiction semble très proches des données réelles.