Analyse de données: Projet final, données BASEBALL

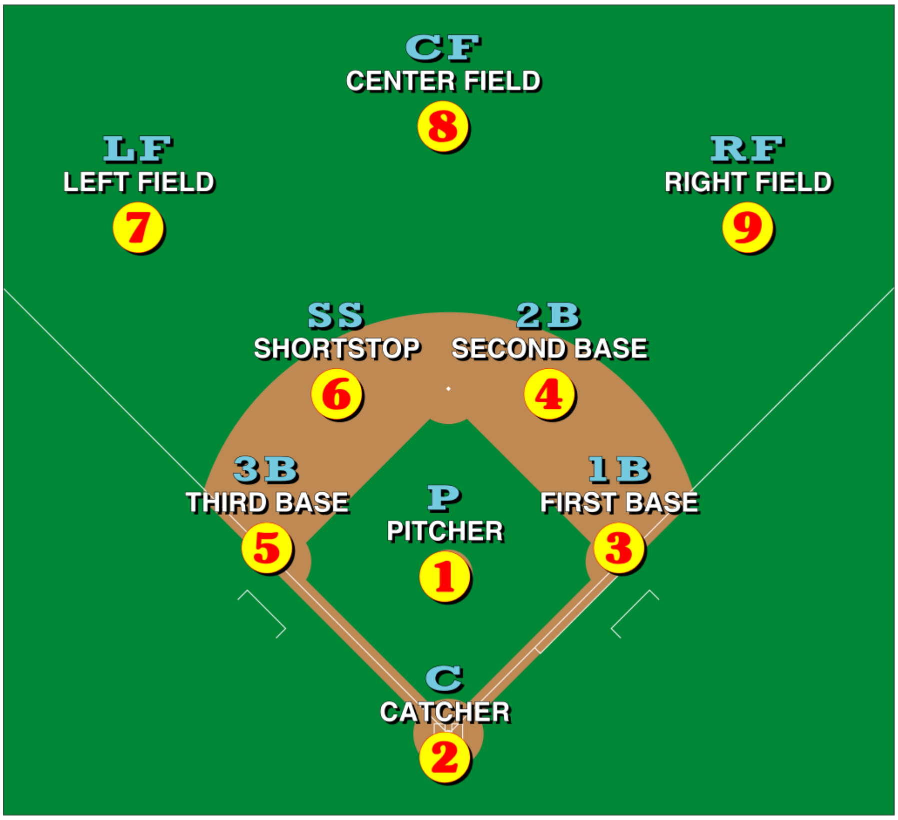
Prédive GOPINATHAN & Jovana KRSTEVSKA

1/12/2021

Introduction

Le baseball est un sport collectif. Il se joue avec une batte pour frapper une balle lancée et des gants pour rattraper la balle. Au baseball, un **frappeur**(hitter) ou batteur est un joueur dont le rôle est de frapper la balle avec sa batte. Pour ce projet nous travaillons avec des données sur les salaires de 1987 des **frappeurs** (hitters) obtenues par l'édition du 20 Avril, 1987 de *Sports Illustrated*, enrichies par des données sur toutes leurs carrières et également performances de 1986 venant de *1987 Baseball Encyclopedia Update*. Nous avons également, les compositions des teams de 1986, grâce à *Elias Sports Bureau*.

Tout d'abord, nous devons nous familiariser avec ces données. Regardons les positions possibles pour un joueur de baseball.



Baseball players position

Dans notre base de données nous n'avons que des observations concernant les **frappeurs** (hitters). plus précisément, nous avons **263** observations. Voici une brève description des quelques positions :

- **Lanceur (Pitcher)** : le lanceur doit analyser chaque frappeur, choisir quels lancers utiliser en fonctions des qualités et défauts de son adversaire.
- **Receveur (Catcher)** : le rôle est primordial lors des phases défensives. Positionné face au lanceur et derrière le batteur, il est le seul joueur de champ à pouvoir interagir à la fois avec le lanceur et avec les joueurs de champ.
- **Première Base (First Base)** : a mission est d'empêcher le batteur d'atteindre le premier but, ou il est positionné.
- **Deuxième base** : les joueurs de deuxième base doivent attraper la balle et la lancer en première base pour enregistrer des retraits. Le joueur lui-même enregistre une assistance. Il est aussi chargé de commencer et de compléter des doubles jeux.
- **Troisième base** : Les joueurs de troisième base sont habituellement des droitiers. Ils sont chargés d'attraper la balle et la lancer en première base, et aussi d'attraper la balle frappée en l'air.
- **Arrêt-court (Shortstop)** : les arrêts-courts jouent la même position que les joueurs de deuxième base, la seule différence étant qu'ils sont plus loin de la première base, et doivent donc la lancer avec plus de force Dans la base de données originale, il y avait plein d'erreurs qui étaient corrigées ensuite. Nous disposons de la version corrigée. Voici un petit aperçu de nos données:

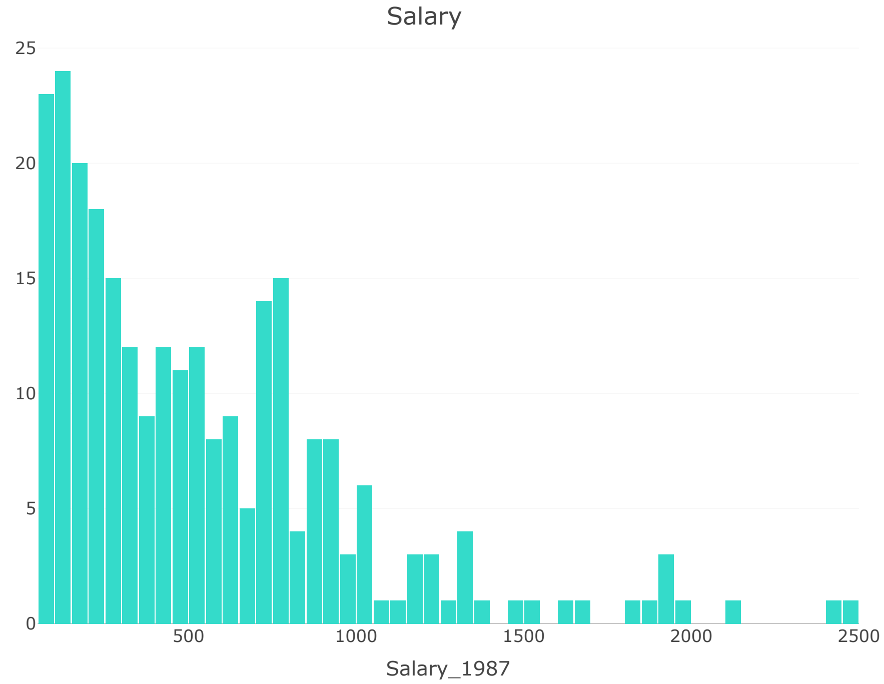
```
## 'data.frame':    263 obs. of  24 variables:
##  $ Name             : Factor w/ 322 levels "Al Newman","Alan Ashby",..: 2 7 8 10 6 1 14 11 9 3 ...
##  $ Bat_times_86     : int  315 479 496 321 594 185 298 323 401 574 ...
##  $ Hits_86          : int  81 130 141 87 169 37 73 81 92 159 ...
##  $ Home_runs_1986   : int  7 18 20 10 4 1 0 6 17 21 ...
##  $ Runs_1986        : int  24 66 65 39 74 23 24 26 49 107 ...
##  $ Runs_batted_1986 : int  38 72 78 42 51 8 24 32 66 75 ...
##  $ Walks_1986       : int  39 76 37 30 35 21 7 8 65 59 ...
##  $ Longevity        : int  14 3 11 2 11 2 3 2 13 10 ...
##  $ Bat_times_career : int  3449 1624 5628 396 4408 214 509 341 5206 4631 ...
##  $ Hits_career      : int  835 457 1575 101 1133 42 108 86 1332 1300 ...
##  $ Home_runs_career : int  69 63 225 12 19 1 0 6 253 90 ...
##  $ Runs_career      : int  321 224 828 48 501 30 41 32 784 702 ...
##  $ Runs_batted_career: int  414 266 838 46 336 9 37 34 890 504 ...
##  $ Walks_career     : int  375 263 354 33 194 24 12 8 866 488 ...
##  $ League_1986      : Factor w/ 2 levels "A","N": 2 1 2 2 1 2 1 2 1 1 ...
##  $ Division_1986    : Factor w/ 2 levels "E","W": 2 2 1 1 2 1 2 2 1 1 ...
##  $ Team_1986        : Factor w/ 24 levels "Atl.","Bal.",..: 9 21 14 14 16 14 10 1 7 8 ...
##  $ Position_1986    : Factor w/ 25 levels "13","1B","1O",..: 11 2 22 2 24 5 24 24 15 24 ...
##  $ Put_outs_1986    : int  632 880 200 805 282 76 121 143 0 238 ...
##  $ Assists_1986     : int  43 82 11 40 421 127 283 290 0 445 ...
##  $ Errors_1986      : int  10 14 3 4 25 7 9 19 0 22 ...
##  $ Salary_1987      : num  475 480 500 91.5 750 ...
##  $ League_1987      : Factor w/ 2 levels "A","N": 2 1 2 2 1 1 1 2 1 1 ...
##  $ Team_1987        : Factor w/ 24 levels "Atl.","Bal.",..: 9 21 5 14 16 13 10 1 7 8 ...
```

On voit que nous avons deux ligues, la **ligue A (Américaine)** et la **ligue N (Nationale)** qui ont deux divisions chacune, **Ouest (W)** et **Est (E)**.

Notre but c'est d'étudier si les baseballeurs sont convenablement payés selon leur performance, d'un coté de l'année dernière mais aussi de toute leure carrière. La variable à expliquer est donc **le salaire des baseballeurs en 1987**, ici appelée **Salary_1987**. Voici un petit récapitulatif de cette variable:

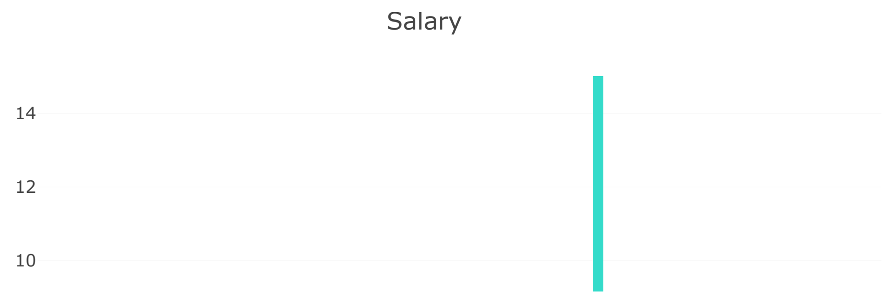
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    67.5   193.0   430.0   542.2   750.0  2460.0
```

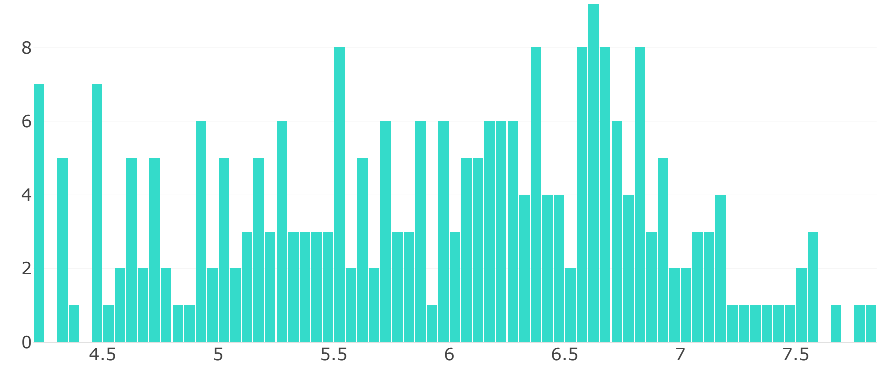
Voici un histogramme du salaire:



Pour essayer de rendre la cible plus symétrique, on va essayer de prédire plutôt **log(Salary_1987)**. Voici un petit résumé sur cette version de la cible, et l'histogramme correspondant.

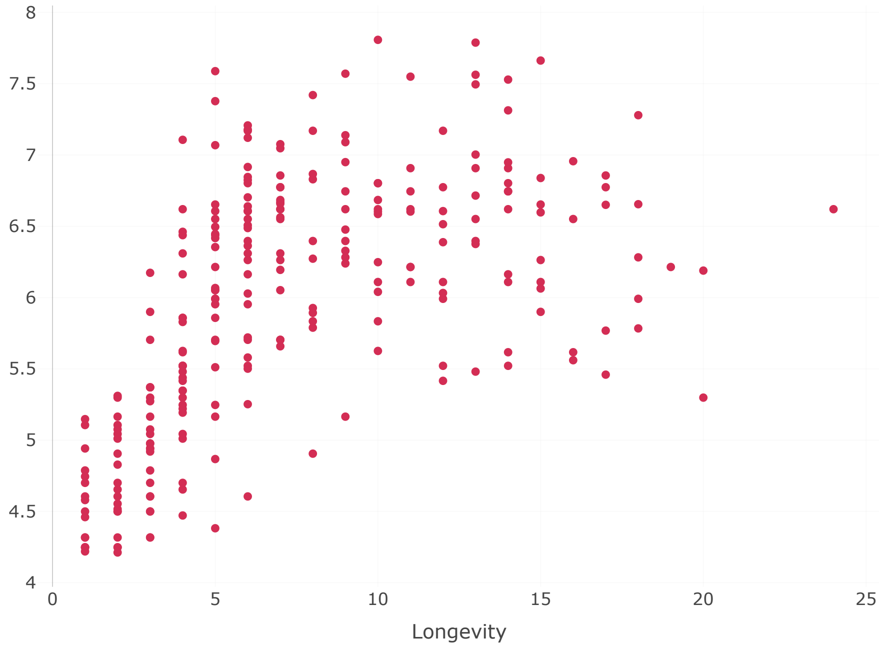
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.212   5.263   6.064   5.945   6.620   7.808
```





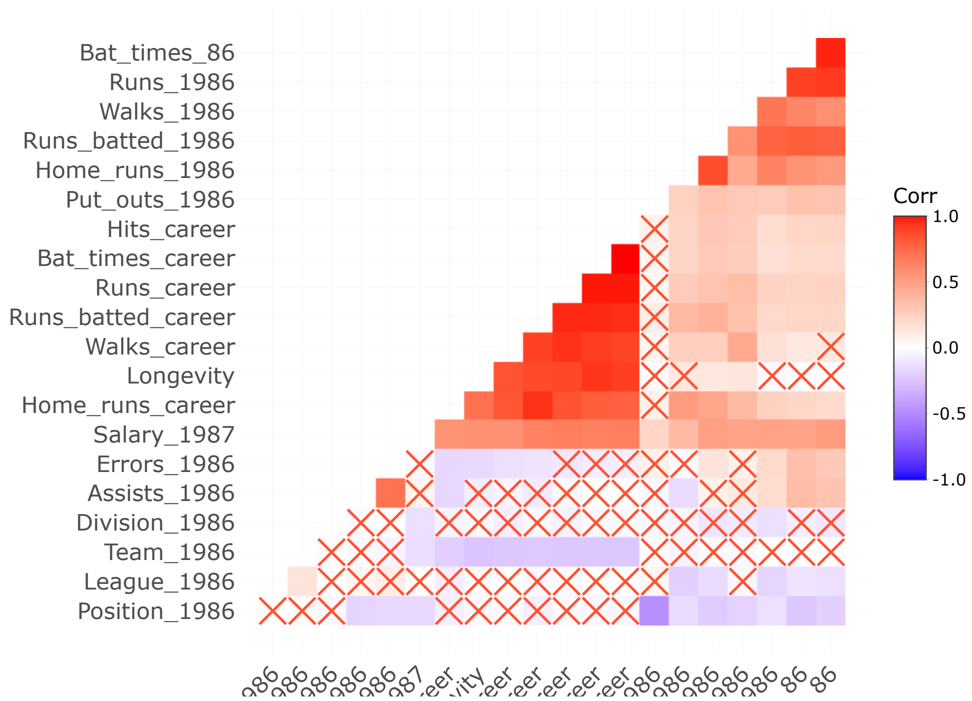
On voit bien que les valeurs sont plus symétriquement distribuées, ainsi on évitera d'avoir des modèles instables.

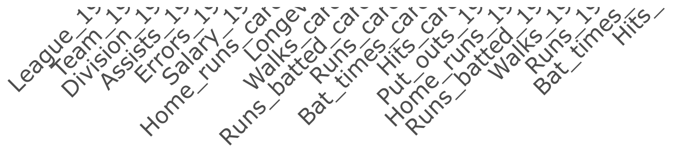
Pour bien voir le lien entre la performance des joueurs et leur salaire, intuitivement la durée de leur carrière est très importante. Une première remarque qu'on peut faire est que le salaire n'est pas linéairement explicable par les années d'activités, ce qui peut paraître contre-intuitif. Ceci n'exclut pas la possibilité que ces deux variables sont non-linéairement dépendants.



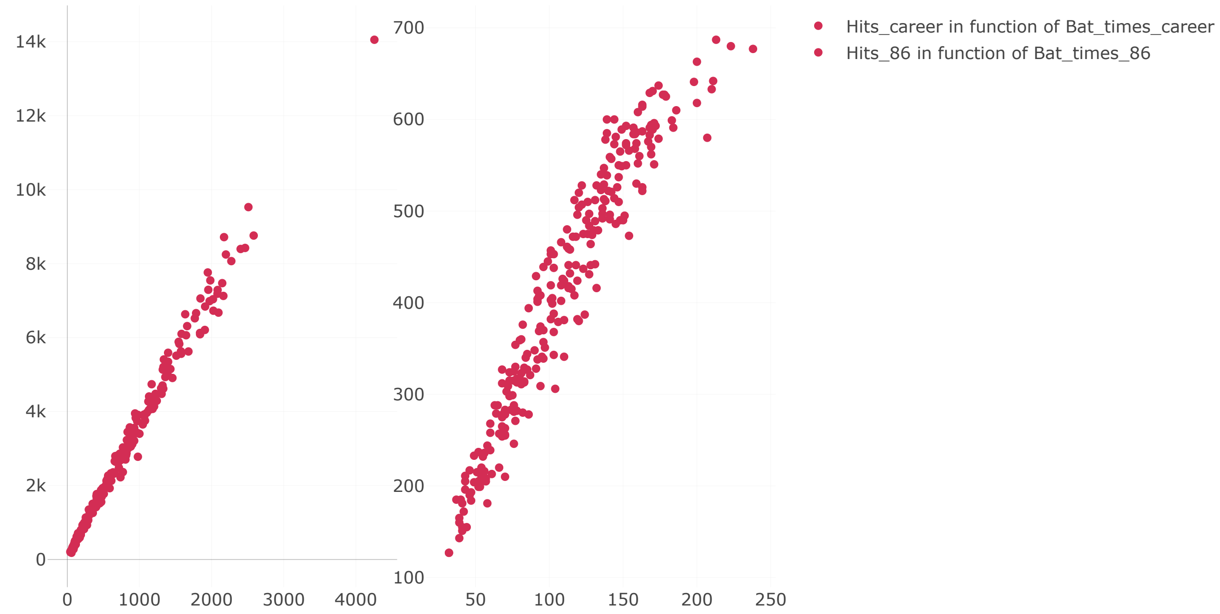
Vérifions les corrélations entre les variables dont on dispose:

Correlation entre les variables et le target





Nous remarquons que beaucoup de variables sont très fortement corrélées entre elles. Ceci nous permettra de réduire le nombre de variables importantes dans la modélisation. Voici une comparaison: **Hits_career** vs. **Bat_times_career** et les mêmes pour l'année 1986: **Hits_86** vs. **Bat_times_86**.



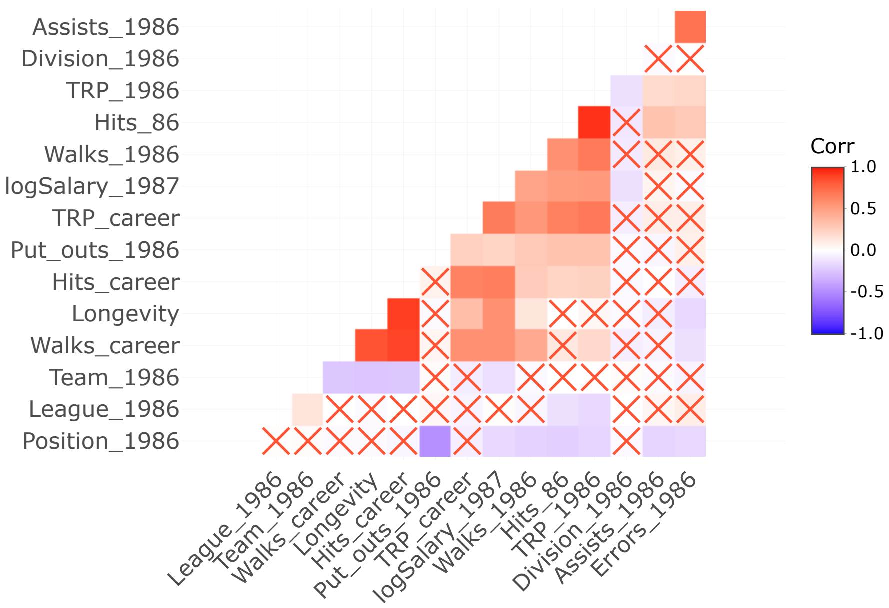
Nous décidons arbitrairement d'en garder que **Hits_career** et **Hits_86**. Donc, on enlève les colonnes **Bat_times_career** et **Bat_times_86**.

Un expert du baseball, **Earnshaw Cook** avait suggéré de considérer une variable que l'on notera **Total Runs Produced (TPR)**, qui donne le nombre total de runs produits par les joueurs, divisé par le nombre d'année de leur activité sportive de haut niveau. Techniquement, ce que l'on obtient c'est une moyenne de runs qu'un joueur a fait par an. On obtient cette variable par la formule suivante:

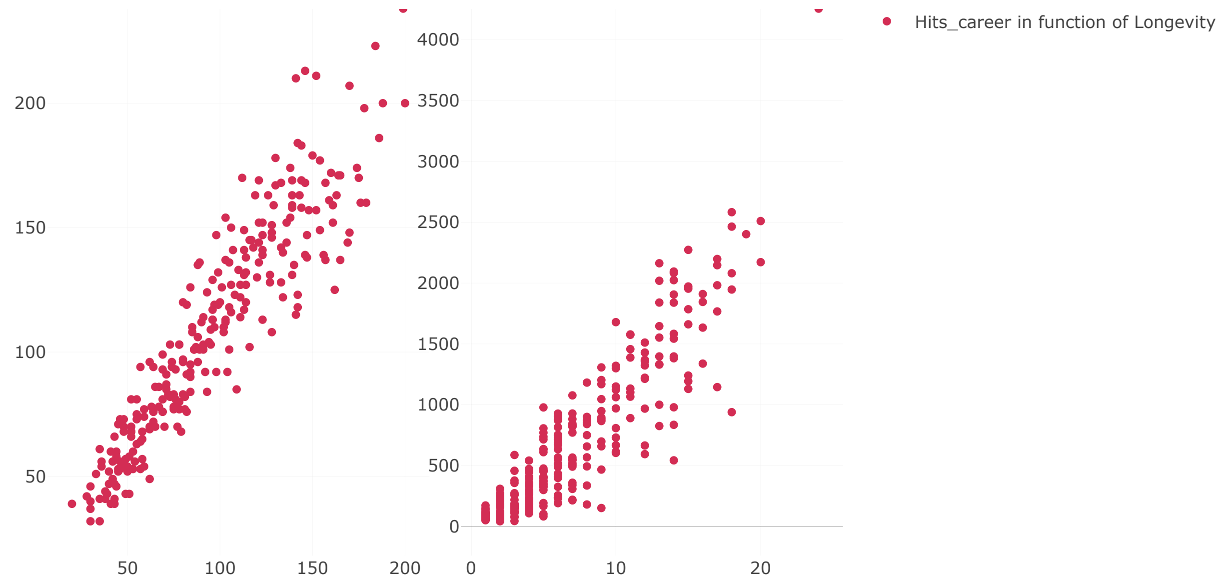
$$TPR = (runs + runs\_batted\_in - home\_runs)/years$$

Cette nouvelle variable nous permet de nous libérer de 6 variables, et d'en ajouter 2 nouvelles, donc de diminuer le nombre de features de 4.

Voici la nouvelle matrice de correlation.



Nous remarquons une correlation forte entre **Hits_86** et **TRP_1986**, ce qui est logique, en considérant le calcul de **TRP**.



On enlève la variable **Hits_86** et la variable **Hits_Career**.

On continue à travailler avec ces variables-là:

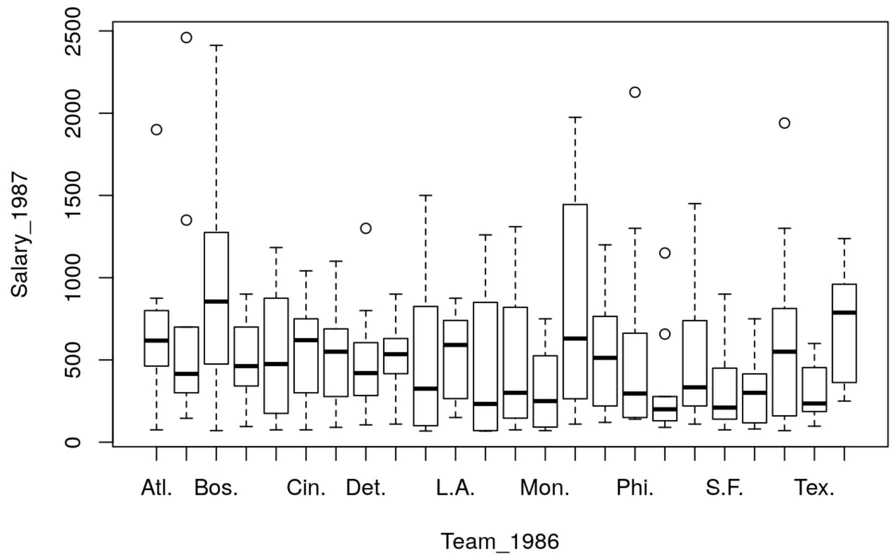
```
## [1] "Walks_1986"    "Longevity"     "Walks_career"  "League_1986"
## [5] "Division_1986" "Team_1986"     "Position_1986" "Put_outs_1986"
## [9] "Assists_1986"  "Errors_1986"   "TRP_career"    "TRP_1986"
```

Variables categorielles

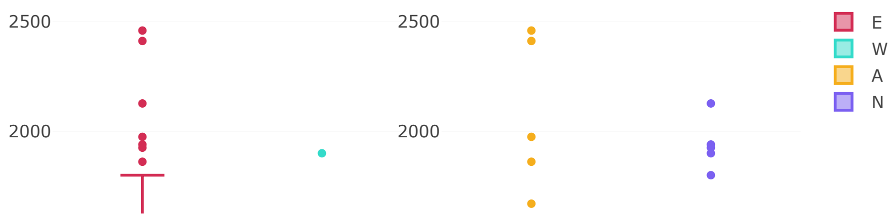
Les vairables categorielles dans notre jeu de données sont:

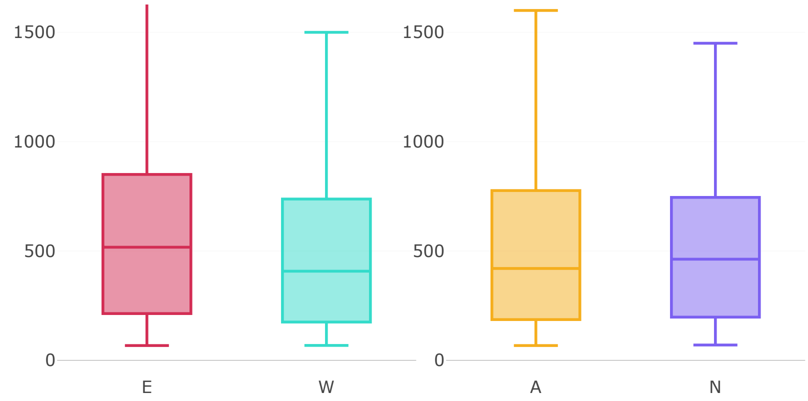
- League_1986 (A ou N)
- Division_1986 (E ou W)
- Teal_1986 (il y en a 25 au total)
- Position_1986

Regardons comment les salaires sont distribués selon l'équipe dans laquelle ils appartiennent.

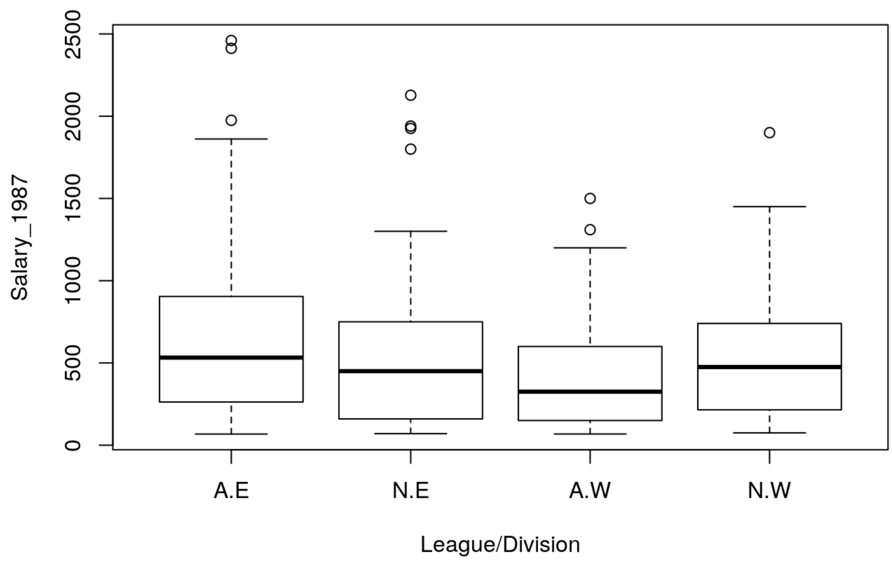


Regardons comment sont distribués les salaires des baseballeurs selon la **Division** et la **Ligue** dans laquelle ils appartiennent.





Ensuite, nous allons étudier l'impact de ces deux variables sur le salaire.



On constate que la distribution des salaires en fonction des leagues et des divisions n'est pas la même : ces deux facteurs semblent donc impacter le salaire des joueurs.
Pour le vérifier, on va réaliser une **anova à deux facteurs**.

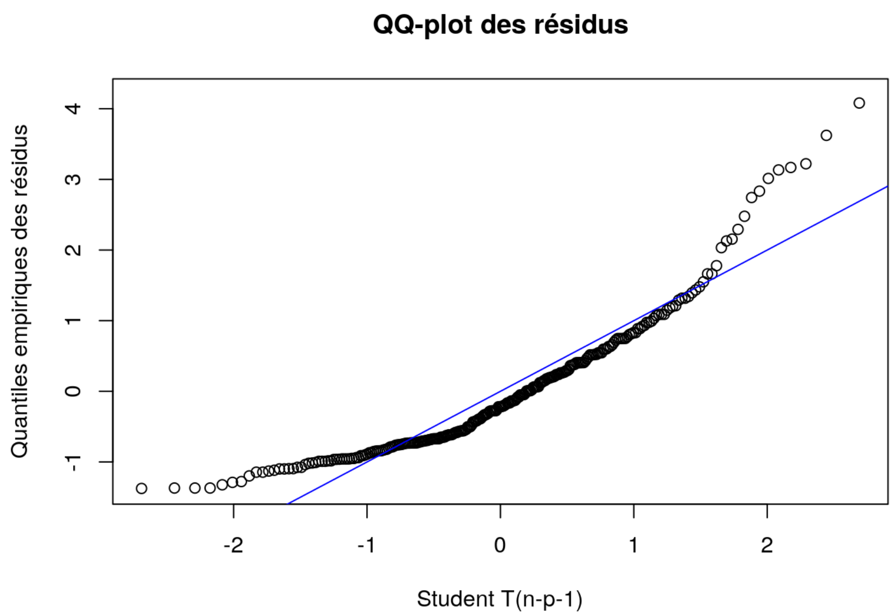
```
##
## Call:
## lm(formula = Salary_1987 ~ League_1986 * Division_1986, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -603.35 -321.41  -95.28  227.22 1789.15
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  670.85      53.69  12.495  < 2e-16 ***
## League_1986N                 -98.50      78.08  -1.262  0.20823
## Division_1986W              -249.44      75.12  -3.320  0.00103 **
## League_1986N:Division_1986W   187.37     109.40   1.713  0.08797 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 442.7 on 259 degrees of freedom
## Multiple R-squared:  0.043,  Adjusted R-squared:  0.03191
## F-statistic: 3.879 on 3 and 259 DF,  p-value: 0.009727
```

La p-valeur du test associé à la league est très élevée. Cependant on ne peut pas rejeter immédiatement l'hypothèse selon laquelle l'interraction league/division influe sur le salaire. Vérifions le avec le tableau d'anova :

```
## Analysis of Variance Table
##
## Response: Salary_1987
##                            Df   Sum Sq Mean Sq F value   Pr(>F)
## League_1986                 1      452     452  0.0023 0.961721
## Division_1986               1  1705570 1705570  8.7011 0.003471 **
## League_1986:Division_1986   1   574971  574971  2.9333 0.087969 .
## Residuals                 259 50768712  196018
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

La p-valeur de l'interraction est de 0.9, on rejette donc l'hypothèse nulle au niveau 10% mais pas au niveau 5%.

Vérifions maintenant l'hyphotèse de normalité des résidus :



Les résidus ne s'alignent pas correctement sur la première bissectrice, il est possible que ce soit du à des salaires très élevés (ceux des stars de la league) ou très bas, en effet rien ne dit que le niveau est homogène et les salaires dépendent principalement du niveau des joueurs. Pour vérifier cela on pourrait par exemple examiner les valeurs aberrantes. On ne peut cependant pas rejeter notre modèle pour autant sans faire une étude plus approfondie.

Analyse des composantes principales (ACP)

Cette méthode d'analyse des données (et de réduction de dimensionnalité) s'appelle également **décomposition orthogonale aux valeurs propres** ou **POD** (anglais : proper orthogonal decomposition). Elle consiste à transformer des variables liées entre elles (dites « corrélées » en statistique) en nouvelles variables décorrélées les unes des autres. Ces nouvelles variables sont nommées **composantes principales**, ou **axes principaux**.

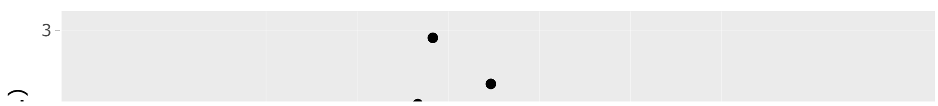
En pratique, l'analyse des composantes principale nous aide à diminuer le nombre de variables utilisées dans un modèle, et rendre l'information moins redondante. Mais elle peut aussi être utilisée pour nous indiquer les variables les plus importantes pour décrire toute (ou quasiment toute) la variance décrite par toutes les variables. Autrement dit, à quelles variables du jeu de données initial peut-on nous restreindre pour expliquer une bonne partie de la totalité d'information disponible.

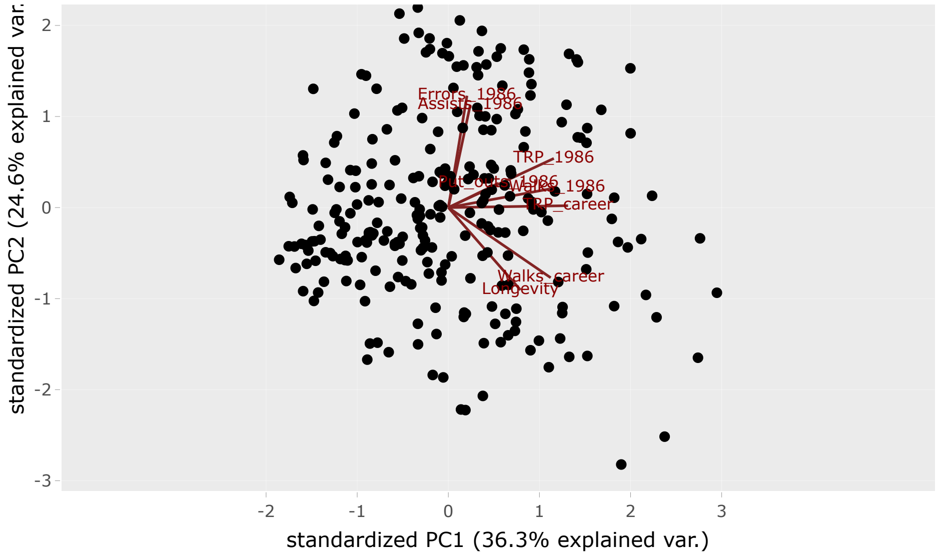
L'ACP n'étant pas très stable pour les données categorielles, on les exclut temporairement. On travaille donc, avec les variables suivantes:

```
## [1] "Walks_1986"    "Longevity"     "Walks_career"  "Put_outs_1986"
## [5] "Assists_1986"  "Errors_1986"   "TRP_career"    "TRP_1986"
```

Et voici un plot de notre ACP, les deux premières composantes principales:

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.7044 1.4037 1.1391 0.87967 0.67513 0.52667 0.49687
## Proportion of Variance 0.3631 0.2463 0.1622 0.09673 0.05698 0.03467 0.03086
## Cumulative Proportion  0.3631 0.6095 0.7716 0.86836 0.92534 0.96001 0.99087
##                            PC8
## Standard deviation     0.27025
## Proportion of Variance 0.00913
## Cumulative Proportion  1.00000
```





Nous pouvons remarquer que 60.9% de la variance est expliquée par seulement les deux premières composantes principales. Regardons ces vecteurs de près:

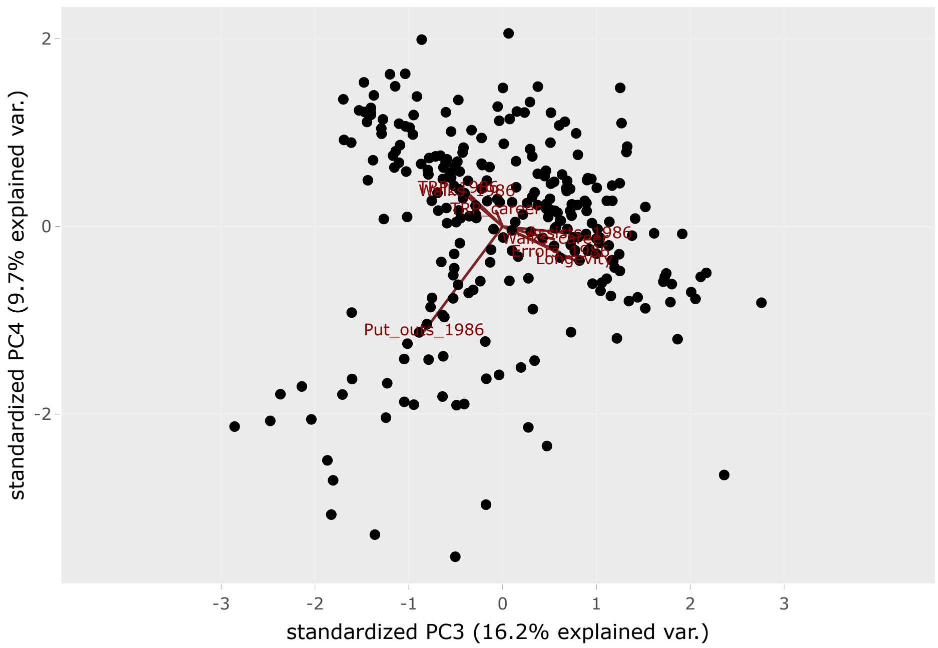
```
##                    PC1          PC2
## Walks_1986    0.45765099  0.102167538
## Longevity     0.30408866 -0.422139014
## Walks_career  0.43147193 -0.359274923
## Put_outs_1986 0.21153048  0.124713143
## Assists_1986  0.09239206  0.524580088
## Errors_1986   0.07854436  0.572841833
## TRP_career    0.50455154  0.009632819
## TRP_1986      0.44485359  0.251602432
```

Les composantes principales sont constituées des valeurs propres de l'orthogonalisation. Le plus la valeur propre est grande pour une variable de départ, le plus elle est importante dans l'explication d'information.

Nous remarquons que la valeur propre la plus grande pour la première composante principale est celle correspondant à **TRP_career**. Ceci n'est pas du tout étonnant, vu que le nombre de runs qu'un joeur à fait au cours de sa carrière joue intuitivement un grand rôle dans le salaire qui gagne.

La valeur propre la plus grande pour la deuxième composante principale est celle qui correspond à **Errors_1986**.

Regardons la troisième et la quatrième composante principale:



Ces résultats ne sont pas très significatifs, car ces deux composantes principales expliquent seulement 25.9% de la variance. Donc, au total, juste avec les 4 premières composantes principales nous expliquons **86.8%** de la variance totale. Et voici les vecteurs correspondants à la troisième et la quatrième composante principale.

```
##                       PC3         PC4
## Walks_1986    -0.21899372  0.27302092
## Longevity      0.43911299 -0.26740365
## Walks_career   0.32107566 -0.10358781
## Put_outs_1986 -0.48076395 -0.83082650
## Assists_1986   0.47319473 -0.06139313
## Errors_1986    0.35497738 -0.20706049
## TRP_career    -0.04224353  0.13042985
## TRP_1986      -0.27072359  0.29882545
```

La valeur propre la plus grande pour la troisième variable et quatrième variable est **Put_outs_1986**.

Cependant, ces résultats ne sont pas vraiment utiles pour comprendre l'impact de ces variables sur la cible. Pour cela on peut avoir recours à d'autres méthodes.

Analyse descendante de l'importance des variables

Essayons d'estimer les variables les plus importantes pour la cible - le salaire des baseballeurs en 1987, en effectuant une analyse **descendante** sur une **regression linéaire multiple** (backward feature importance analysis).

C'est-à-dire, on va modéliser le problème en utilisant toutes les variables disponibles, et ensuite, on enlèvera celle qui présente la p-value la plus grande, donc, celle qui a été le moins utile pour décrire la cible.

Nous commençons par utiliser toutes les variables:

```
## 
## Call:
## lm(formula = y ~ ., data = X_back)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -1.4798 -0.3752  0.0360  0.4144  1.1481 
## 
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)    
## (Intercept)    3.9584234  0.2217276  17.853  < 2e-16 ***
## Walks_1986     0.0087936  0.0026232   3.352 0.000926 ***
## Longevity      0.1287538  0.0145325   8.860  < 2e-16 ***
## Walks_career  -0.0013041  0.0003240  -4.025 7.56e-05 ***
## League_1986    0.1431358  0.0686633   2.085 0.038120 *  
## Division_1986 -0.1244497  0.0663045  -1.877 0.061691 .  
## Team_1986     -0.0035858  0.0050614  -0.708 0.479328    
## Position_1986 -0.0077358  0.0085568  -0.904 0.366837    
## Put_outs_1986  0.0001290  0.0001441   0.896 0.371366    
## Assists_1986   0.0004969  0.0003284   1.513 0.131527    
## Errors_1986   -0.0157080  0.0072612  -2.163 0.031467 *  
## TRP_career     0.0113028  0.0016946   6.670 1.63e-10 ***
## TRP_1986       0.0025481  0.0015196   1.677 0.094828 .  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.5289 on 250 degrees of freedom
## Multiple R-squared:  0.6582, Adjusted R-squared:  0.6418 
## F-statistic: 40.11 on 12 and 250 DF,  p-value: < 2.2e-16
```

Nous observons que la p-value la plus grande (0.479328) est obtenue pour la variable **Team_1986**. Donc, on l'enlève pour notre modèle suivant.

```
##
## Call:
## lm(formula = y ~ ., data = X_back)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.44360 -0.37023  0.03064  0.41295  1.18657
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.9200119  0.2147829  18.251  < 2e-16 ***
## Walks_1986     0.0087588  0.0026202   3.343 0.000956 ***
## Longevity      0.1293539  0.0144934   8.925  < 2e-16 ***
## Walks_career  -0.0012857  0.0003227  -3.985 8.86e-05 ***
## League_1986    0.1354528  0.0677341   2.000 0.046603 *
## Division_1986 -0.1252469  0.0662291  -1.891 0.059760 .
## Position_1986 -0.0078395  0.0085470  -0.917 0.359908
## Put_outs_1986  0.0001285  0.0001439   0.893 0.372740
## Assists_1986   0.0004889  0.0003279   1.491 0.137219
## Errors_1986   -0.0151129  0.0072053  -2.097 0.036952 *
## TRP_career     0.0112738  0.0016925   6.661 1.71e-10 ***
## TRP_1986       0.0025175  0.0015175   1.659 0.098374 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5284 on 251 degrees of freedom
## Multiple R-squared:  0.6575, Adjusted R-squared:  0.6425
## F-statistic:  43.8 on 11 and 251 DF,  p-value: < 2.2e-16
```

Ensuite, on enlève la variable **Put_outs_1986**, car sa p-valeur est la plus grande: 0.372740.

```
##
## Call:
## lm(formula = y ~ ., data = X_back)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.41902 -0.37966  0.02556  0.42577  1.23147
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.9536783  0.2113628  18.706  < 2e-16 ***
## Walks_1986     0.0089381  0.0026114   3.423 0.000723 ***
## Longevity      0.1292584  0.0144872   8.922  < 2e-16 ***
## Walks_career  -0.0013014  0.0003220  -4.041 7.07e-05 ***
## League_1986    0.1408621  0.0674355   2.089 0.037726 *
## Division_1986 -0.1265131  0.0661873  -1.911 0.057082 .
## Position_1986 -0.0115136  0.0074885  -1.538 0.125423
## Assists_1986   0.0004220  0.0003191   1.323 0.187157
## Errors_1986   -0.0144472  0.0071637  -2.017 0.044785 *
## TRP_career     0.0114611  0.0016787   6.827 6.43e-11 ***
## TRP_1986       0.0025939  0.0015145   1.713 0.087995 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5282 on 252 degrees of freedom
## Multiple R-squared:  0.6564, Adjusted R-squared:  0.6428
## F-statistic: 48.14 on 10 and 252 DF,  p-value: < 2.2e-16
```

Et puis on enlève **Assists_1986**, et ainsi de suite. A la fin, on obtient un classement des variables par rapport à leur importance dans la détermination de la cible pour un modèle de regression linéaire multiple.

1. **TRP_career**
2. **Longevity**
3. **Walks_1986**
4. **Walks_career**
5. **Division_1986**

Modèles de regression linéaire multiple

Résumons quelques remarques essentielles qu'on a pu faire tout au long de notre analyse:

- La modélisation du problème par un modèle de regression linéaire multiple en utilisant seulement les variables initiales dont on dispose ne donne pas des résultats très satisfaisants, comme on l'avait vu par le score *R^2 dans l'analyse descendante.
- Le salaire des joeurs de baseball a forcément un lien avec les années d'activité du joeur, et ce lien est juste "partiellement" linéaire.
- La variable **TRP_career** est très importante pour l'explication de la cible (vu dans l'analyse descendante), et en général elle joue un grand rôle dans la quantité d'information totale (vu dans la partie ACP)

- Pour qu'un modèle linéaire soit assez stable pour les prédictions des salaires de l'année 1987, il faut y inclure une variable de performance de l'année 1986.

En vue de la première remarque, on essayera de créer un modèle de regression linéaire multiple avec un nombre de features pas plus grand que 5.

En vue de la deuxième remarque, en plus de la variable **Longevity**, on va considérer son carré et son cube. C'est-à-dire:

$$\text{Longevity\_squared} = \text{Longevity}^2$$
$$\text{Longevity\_cubic} = \text{Longevity}^3$$

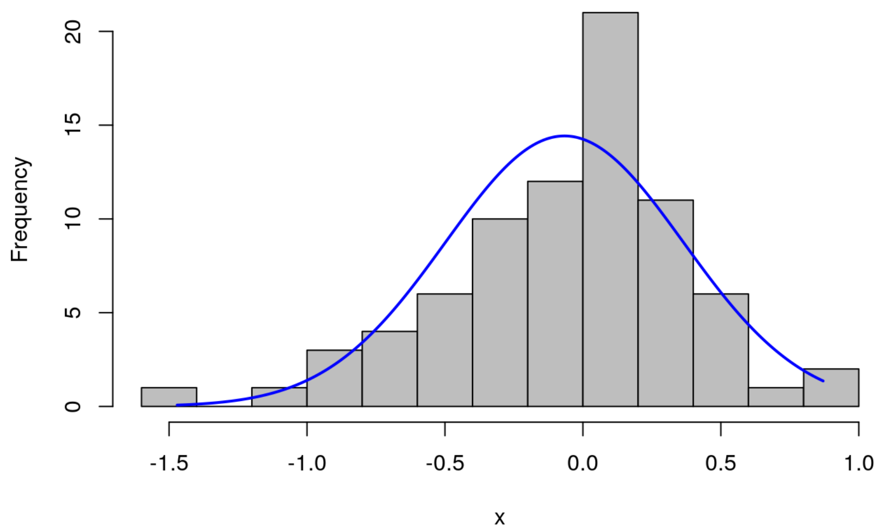
En vue de la troisième remarque, on va inclure **TRP_career** dans notre modèle.

Et finalement, en vue de la dernière remarque, on chosit la variable **Walks_1986**, comme une variable de performance des joueurs de l'année 1986.

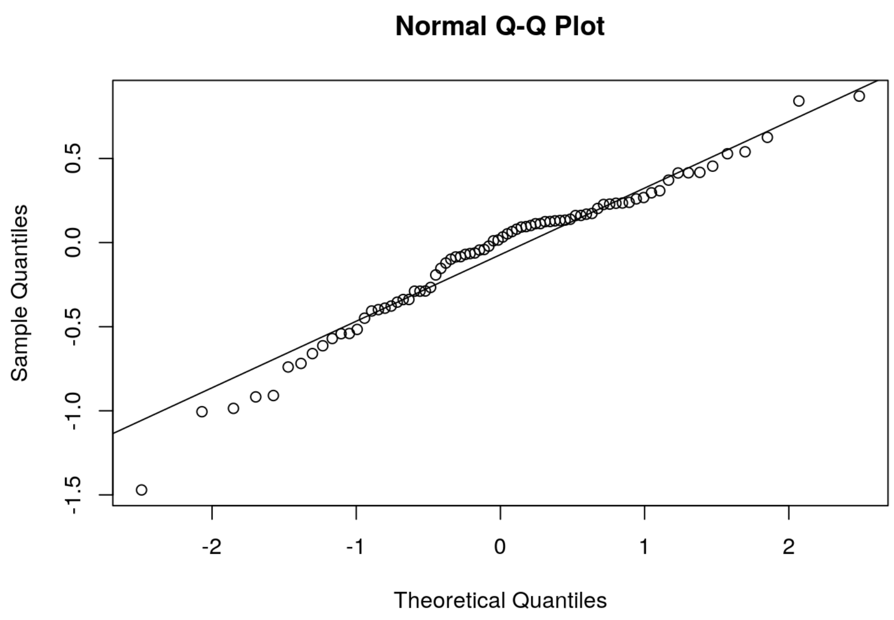
Essayons donc un modèle de regression linéaire multiple qui utlise: **Longevity**, **Longevity_squared**, **Longevity_cubic**, **TRP_career** et **Walks_1986**. Voici ce que l'on obtient:

```
##
## Call:
## lm(formula = y_train ~ ., data = X_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.04284 -0.18930 -0.00764  0.19940  0.69952
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.7959292  0.1089346  25.666  < 2e-16 ***
## Longevity      0.6408712  0.0441547  14.514  < 2e-16 ***
## TRP_career     0.0121570  0.0007837  15.511  < 2e-16 ***
## Walks_1986     0.0044713  0.0011985   3.731 0.000256 ***
## years_squared -0.0490035  0.0053276  -9.198  < 2e-16 ***
## years_cubic    0.0010954  0.0001861   5.886 1.91e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2954 on 179 degrees of freedom
## Multiple R-squared:  0.8898, Adjusted R-squared:  0.8868
## F-statistic: 289.2 on 5 and 179 DF,  p-value: < 2.2e-16
```

Nous séparons le jeu de données que nous avons en **train** (70% des observations) et **test** (30% des observations). Les observations dans chaque ensemble sont aléatoirement choisies, en respectant les tailles mentionnées entre parenthèses.



On voit que les résidus sont bien distribuées selon une loi normale, on le vérifie grace aux graphiques suivants:

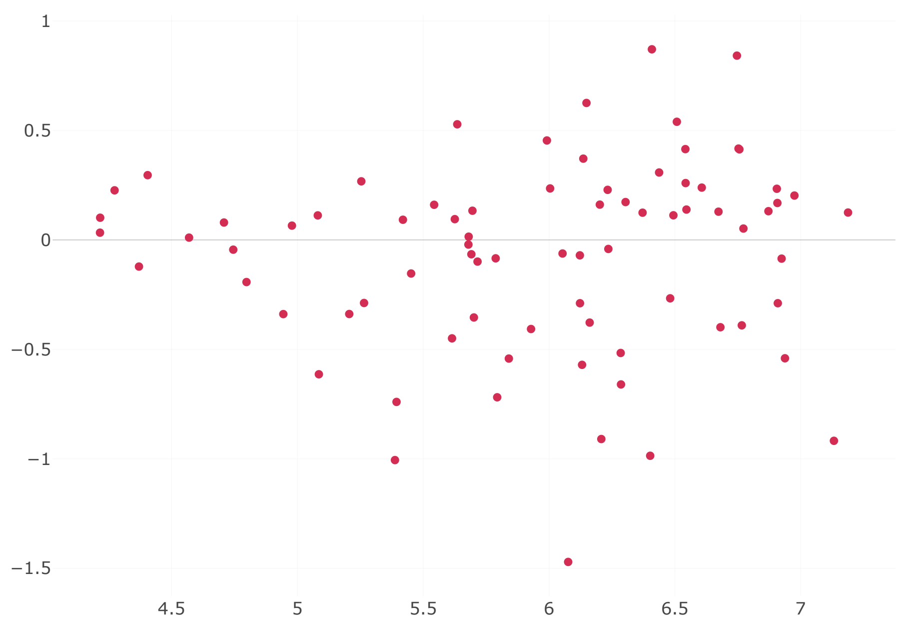


Et voici la Root Mean Squared Error sur le test set:

```
## [1] "RMSE on test set:"
```

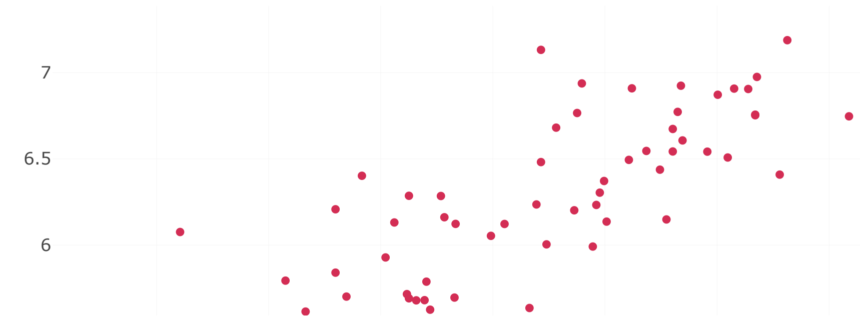
```
## [1] 256.0595
```

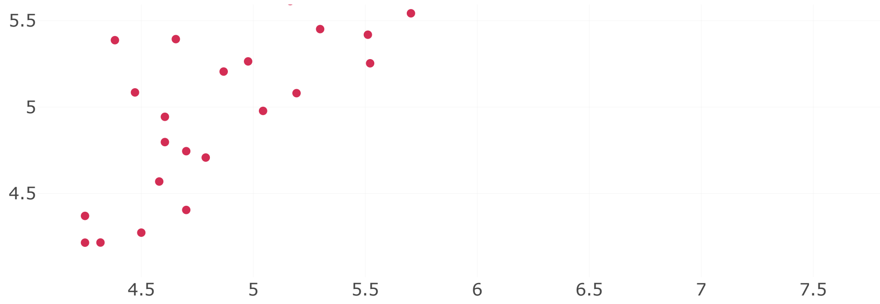
Voici les résidus en fonction des prédictions:



Nous observons un assez grand écart pour certaines valeurs, ce qui est indicateur des valeurs aberrantes dans le jeu de données, nous devons les enlever ou les corriger pour avoir un modèle plus robuste et précis.

En revanche, nous observons une bonne tendance linéaire lorsqu'on affiche les predictions en fonction des vraies valeurs pour le salaire dans le test set:





Conclusion

Nous pouvons affirmer que les salaires des joueurs sont bien expliqués par leur performance tout au long de leur carrière et de la performance de l'année d'avant. Cependant, les modèles que nous avons utilisé sont très sensibles aux valeurs aberrantes et une étude plus approfondie (de préférance en collaboration avec des experts de baseball) est demandée afin de produire des modèles plus stables et robustes.

Comme le lien entre les variables et la cible n'est pas forcément linéaire, un modèle non-linéaire serait plus convenable. Même des simples modèles de machine learning (comme **Random Forest**, **Gradient Boosting**) donnent des résultats bien meilleurs qu'une regression linéaire multiple, malgrè le bon choix des variables.