Selection_de_variables

November 22, 2019

[133]:
```
library(leaps)
library(glmnet)
```

0.1 IMPORTATION DES DONNEES

[134]:
```
data_tot = read.csv('./Donnees/Plusieurs_pays/Total.csv', header = TRUE)
data_tot = data_tot[-c(47,60)]
```

Création d'un nouveau dataframe sans les variables `Rating` et `Pays`, utilisé après :

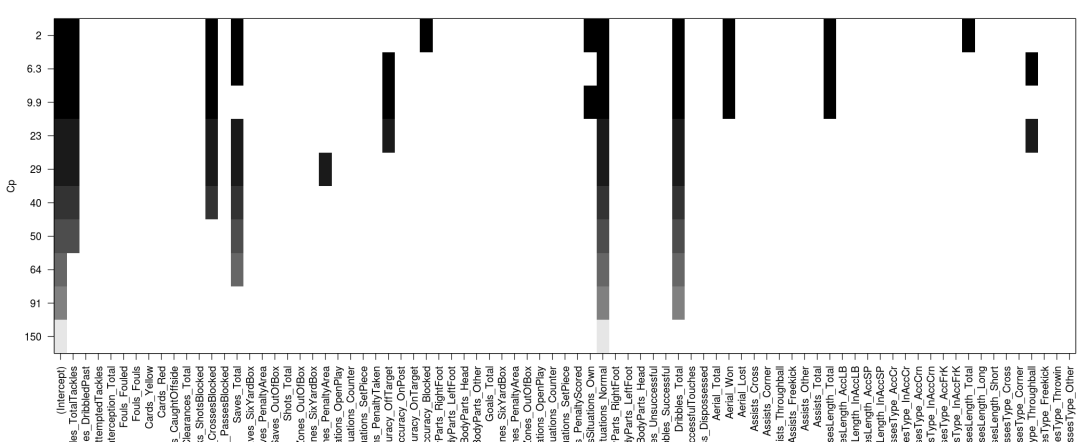
[135]:
```
data = data_tot[,-c(81,82,83,84)]
```

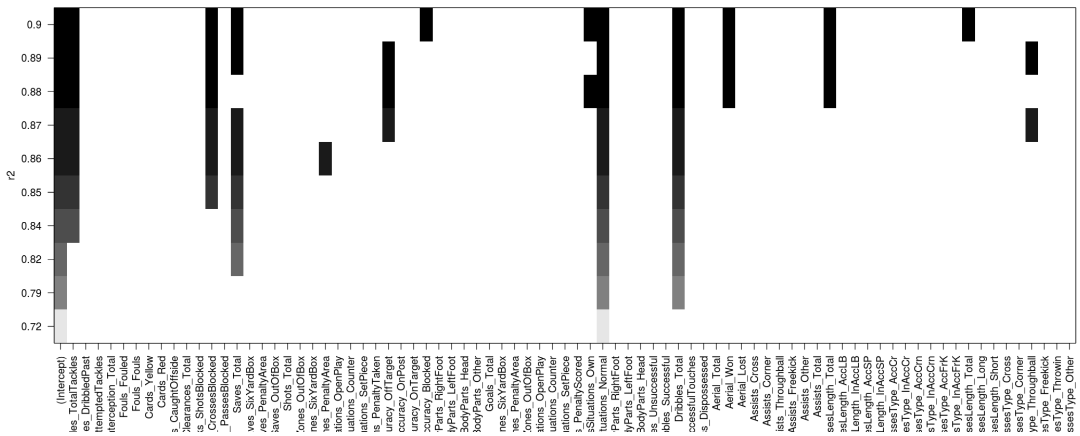
0.2 Sélection de variables

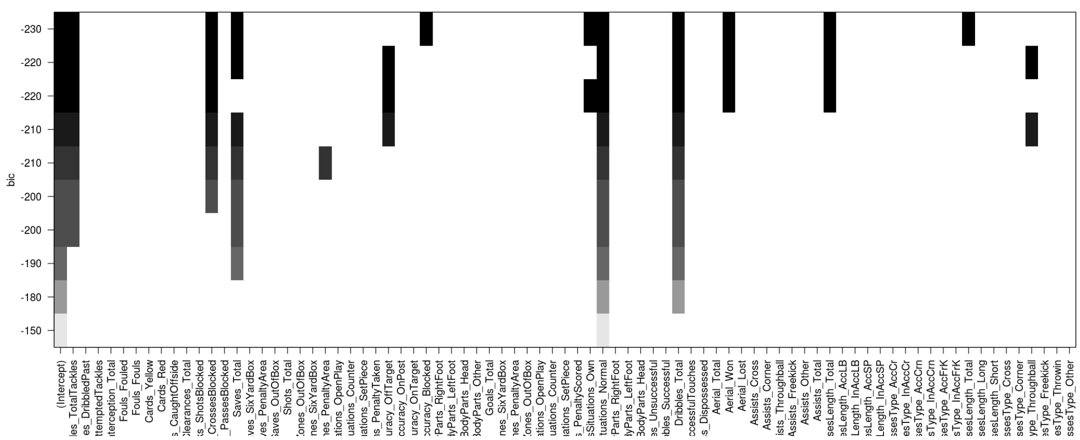
0.2.1 BIC

[136]:
```
choixb <- regsubsets(data_tot$Rating~., data=data,nbest=1, nvmax=10,␣
↪method="seqrep")
```

[137]:
```
options(repr.plot.width=18, repr.plot.height=8)
plot(choixb,scale="Cp")
plot(choixb,scale="r2")
plot(choixb,scale="bic")
```







Les trois différents critères utilisés ci-dessus pour la sélection de modèle (Cp de Mallows, R2 et BIC) semblent donner les mêmes résultats.
Suivant le critère choisi, il faut soit le maximiser (R2), soit le minimiser (Cp et BIC). Dans les deux cas, il s'agit de trouver les variables mises en noir sur la ligne du haut.

[138]:
```
nb_min = which.min(summary(choixb)$bic)
coef(choixb, nb_min)
```

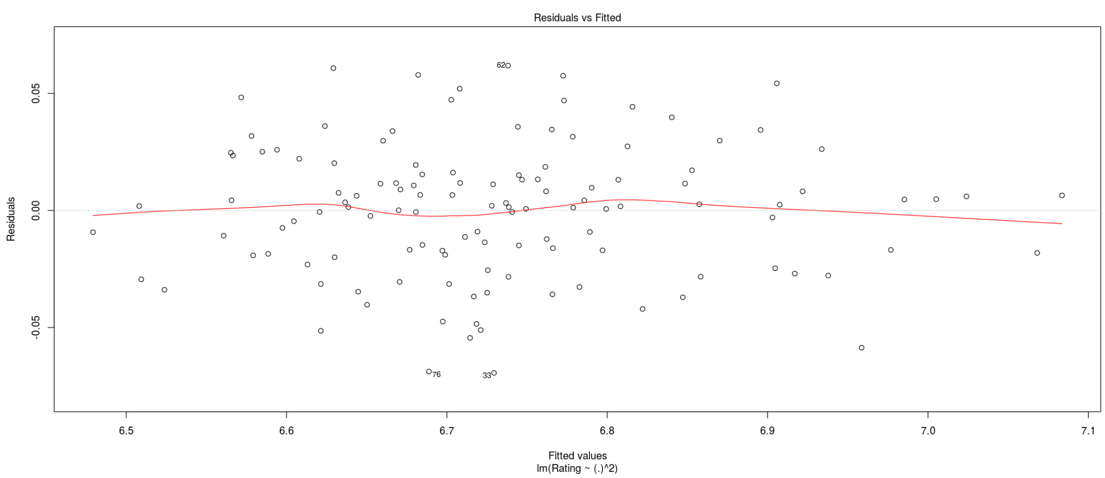
(Intercept)	5.75215912848794	**Tackles_TotalTackles**	0.0116722138782329
Blocks_CrossesBlocked	0.0446894205980564	**Saves_Total**	-0.0197312817417468
ShotsAccuracy_Blocked	-0.0298931882729577	**GoalsSituations_Own**	0.221852796031417
GoalsSituations_Normal	0.194642946229757	**Dribbles_Total**	0.00852515056606557

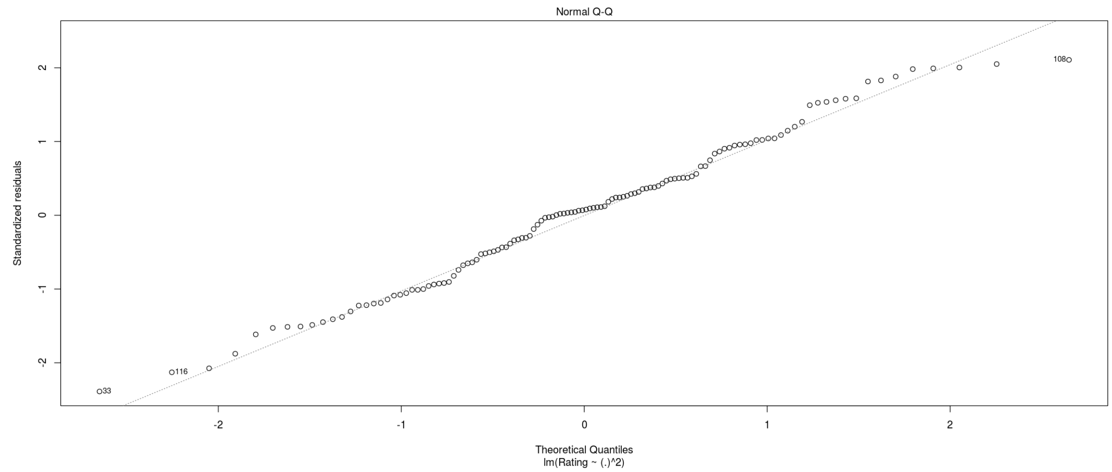
Aerial_Won 0.00705457233332821 **PassesLength_Total** 0.000369394176703441 **KeyPassesLength_Total** 0.0189994394720078

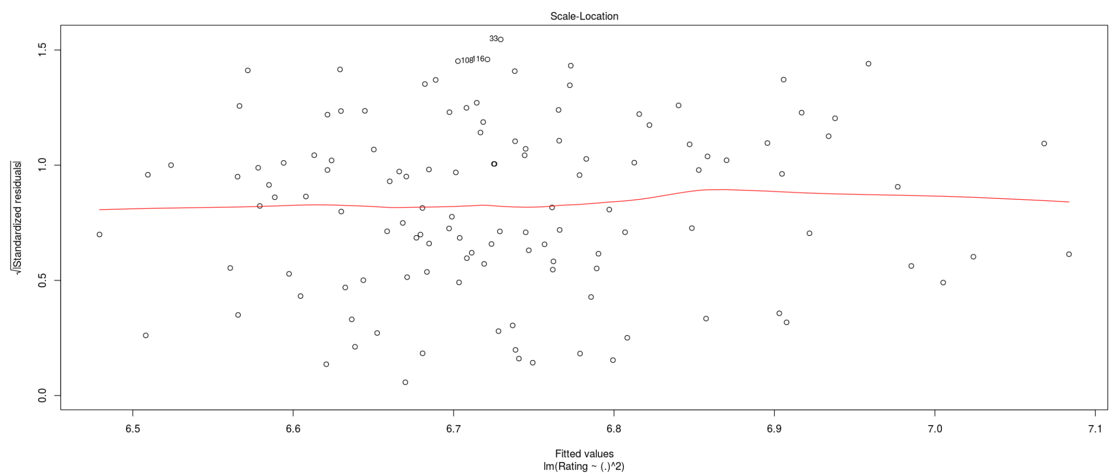
[139]:
```
data_bic = data.frame(data_tot$Rating, data_tot$Tackles_TotalTackles, data_tot$Blocks_CrossesBlocked, data_tot$Saves_Total,
                      data_tot$ShotsAccuracy_Blocked, data_tot$GoalsSituations_Own, data_tot$GoalsSituations_Normal,
                      data_tot$Dribbles_Total, data_tot$Aerial_Won, data_tot$PassesLength_Total, data_tot$KeyPassesLength_Total)

names(data_bic) <- c("Rating", "Tackles_TotalTackles", "Blocks_CrossesBlocked", "Saves_Total", "ShotsAccuracy_Blocked",
                     "GoalsSituations_Own", "GoalsSituations_Normal", "Dribbles_Total", "Aerial_Won",
                     "PassesLength_Total", "KeyPassesLength_Total")
```

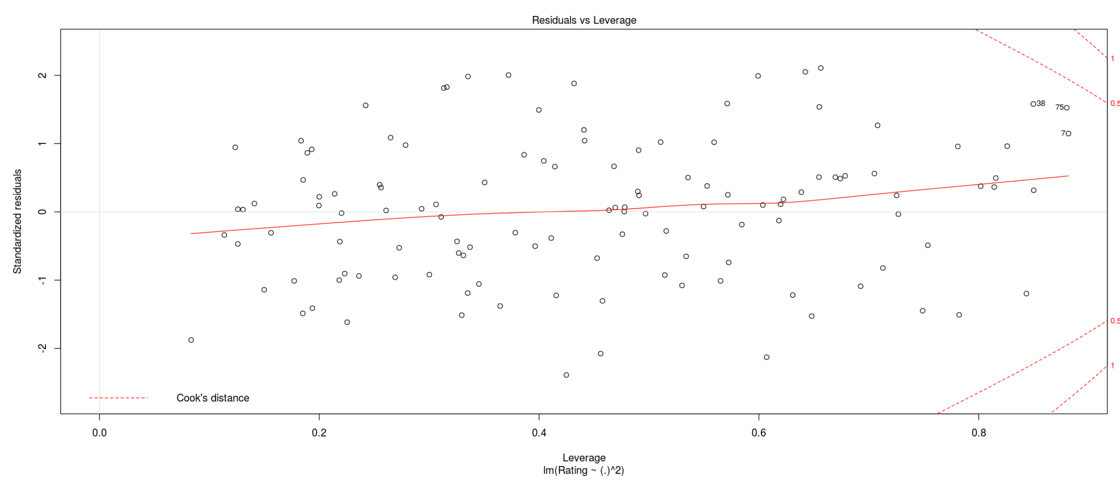
[140]:
```
reg_bic = lm(Rating~(.)^2, data = data_bic)
```

[141]:
```
plot(reg_bic)
```




Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(Rating ~ (.)^2)
108
116
33
-2
-1
0
1
2


Scale-Location
√|Standardized residuals|
Fitted values
lm(Rating ~ (.)^2)
33
108
116
0.0
0.5
1.0
1.5
6.5
6.6
6.7
6.8
6.9
7.0
7.1


Residuals vs Leverage
Standardized residuals
Leverage
lm(Rating ~ (.)^2)
Cook's distance
38
75
7
1
0.5
0.5
1
0.0
0.2
0.4
0.6
0.8
-2
-1
0
1
2

On voit que le graphe des résidus ne présente pas de forme particulière. De plus, le graphe quantile-quantile est plus ou moins aligné (quelques soucis sur les petits et grands quantiles).

On va refaire de la sélection de variables, mais pour chaque pays, et ainsi observer les variables vraiment influentes.

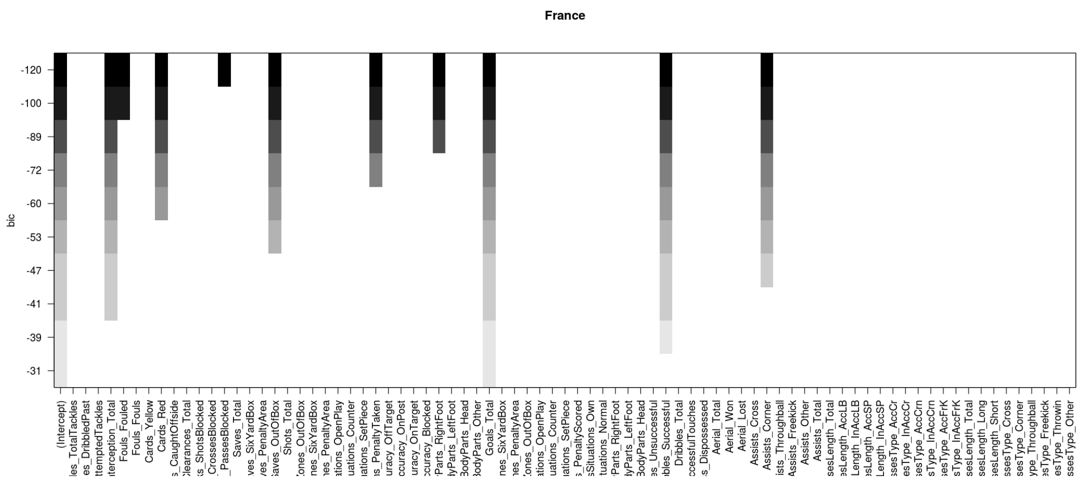
[142]:
```
France = data[data_tot$Pays == "France",]
Allemagne = data[data_tot$Pays == "Allemagne",]
Italie = data[data_tot$Pays == "Italie",]
Espagne = data[data_tot$Pays == "Espagne",]
Argentine = data[data_tot$Pays == "Argentine",]
Angleterre = data[data_tot$Pays == "Angleterre",]
```

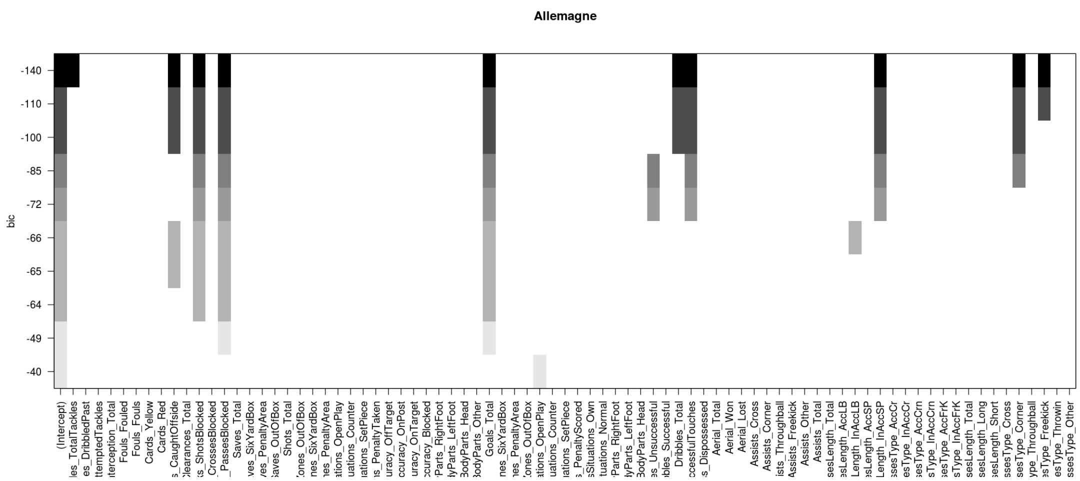
[143]:
```
choix_France <- regsubsets(data_tot[data_tot$Pays=="France",]$Rating~.,␣
 ↪data=France,nbest=1, nvmax=10, method="seqrep")
choix_Allemagne <- regsubsets(data_tot[data_tot$Pays=="Allemagne",]$Rating~.,␣
 ↪data=Allemagne,nbest=1, nvmax=10, method="seqrep")
choix_Italie <- regsubsets(data_tot[data_tot$Pays=="Italie",]$Rating~.,␣
 ↪data=Italie,nbest=1, nvmax=10, method="seqrep")
choix_Espagne <- regsubsets(data_tot[data_tot$Pays=="Espagne",]$Rating~.,␣
 ↪data=Espagne,nbest=1, nvmax=10, method="seqrep")
choix_Argentine <- regsubsets(data_tot[data_tot$Pays=="Argentine",]$Rating~.,␣
 ↪data=Argentine, nbest=1, nvmax=10, method="seqrep")
choix_Angleterre <- regsubsets(data_tot[data_tot$Pays=="Angleterre",]$Rating~.,␣
 ↪data=Angleterre, nbest=1, nvmax=10, method="seqrep")
```

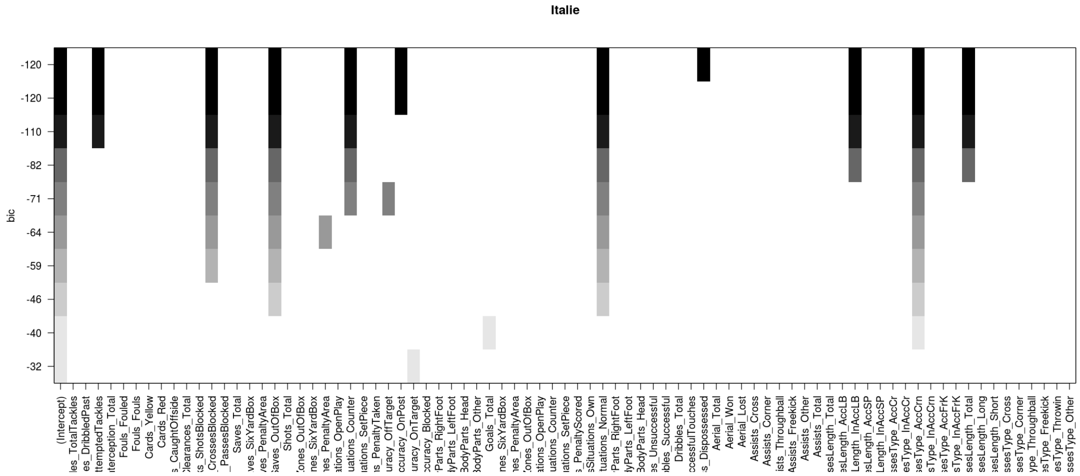
```
Warning message in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
force.in = force.in, :
61  linear dependencies found
Warning message in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
force.in = force.in, :
63  linear dependencies found
Warning message in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
force.in = force.in, :
61  linear dependencies found
Warning message in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
force.in = force.in, :
61  linear dependencies found
Warning message in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
force.in = force.in, :
55  linear dependencies found
Warning message in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
force.in = force.in, :
61  linear dependencies found
```

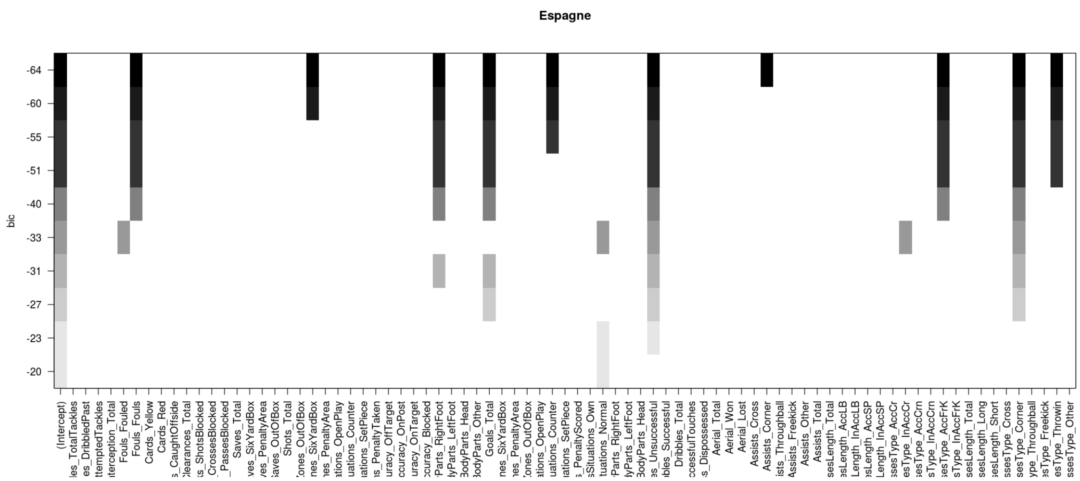
[144]:
```
plot(choix_France,scale="bic", main = "France")
plot(choix_Allemagne,scale="bic", main = "Allemagne")
```

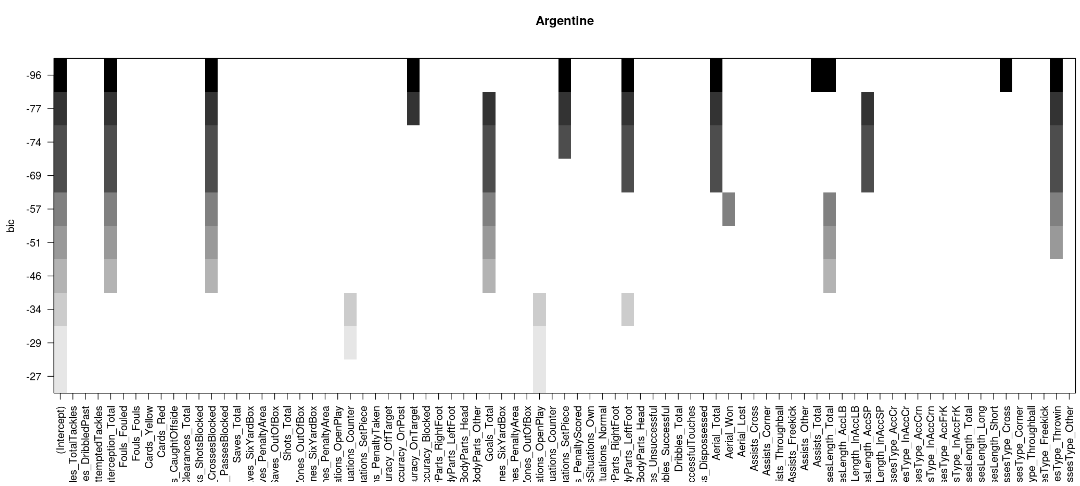
```r
plot(choix_Italie,scale="bic", main = "Italie")
plot(choix_Espagne,scale="bic", main = "Espagne")
plot(choix_Argentine,scale="bic", main = "Argentine")
plot(choix_Angleterre,scale="bic", main = "Angleterre")
```

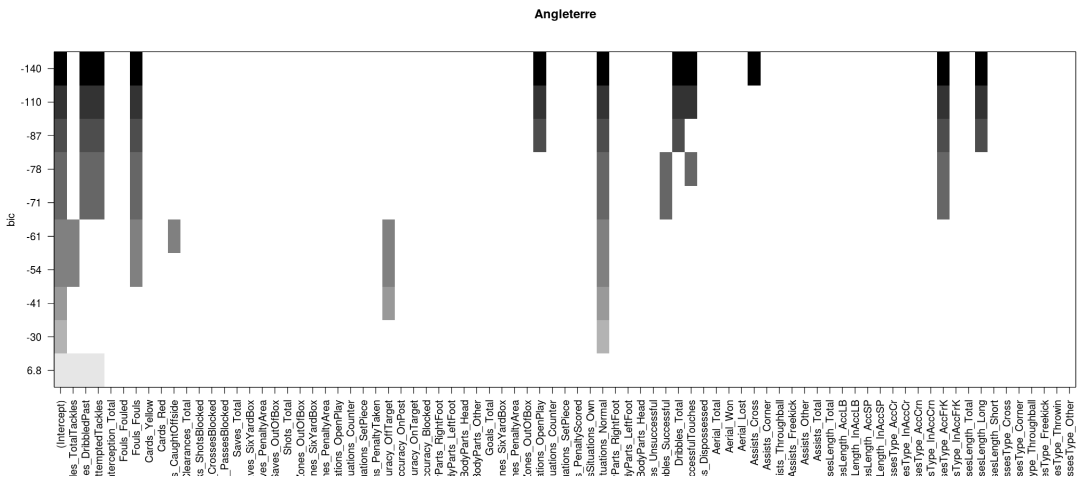





Italie
bic
-120
-120
-110
-82
-71
-64
-59
-46
-40
-32


Espagne
bic
-64
-60
-55
-51
-40
-33
-31
-27
-23
-20


Argentine
bic
-96
-77
-74
-69
-57
-51
-46
-34
-29
-27



On voit que les variables retenues ne sont pas les mêmes d'un pays à l'autre. On va les afficher.

[145]:
```
nb_min = which.min(summary(choix_France)$bic)
coef(choix_France, nb_min)
```

(Intercept) 6.13982138539476 **Interception_Total** 0.0237925361751378 **Fouls_Fouled** 0.00780851950097679 **Cards_Red** -0.599989959343333 **Blocks_PassesBlocked** 0.00953063969054786 **Saves_OutOfBox** -0.158696058583241 **ShotsSituations_PenaltyTaken** -0.299210436279684 **ShotsBodyParts_RightFoot** 0.01058761355573 **Goals_Total** 0.248842297964619 **Dribbles_Successful** 0.0144764285972766 **Assists_Corner** -0.685200840055928

[146]:
```
nb_min = which.min(summary(choix_Allemagne)$bic)
coef(choix_Allemagne, nb_min)
```

(Intercept) 6.3990913593768 **Tackles_TotalTackles** -0.00307838302127169 **OffSides_CaughtOffside** 0.019588686984888 **Blocks_ShotsBlocked** -0.100248995583956 **Blocks_PassesBlocked** 0.0312030351619673 **Goals_Total** 0.254083488003104 **Dribbles_Total** -0.0108021870270855 **PossesionLoss_UnsuccessfulTouches** 0.0329907893164837 **PassesLength_InAccSP** -0.00653498957293113 **KeyPassesType_Corner** 0.0420783100445682 **KeyPassesType_Freekick** 0.0317966473768425

[147]:
```
nb_min = which.min(summary(choix_Italie)$bic)
coef(choix_Italie, nb_min)
```

(Intercept) 6.1219354788306 **Tackles_TotalAttemptedTackles** -0.00631437235721584 **Blocks_CrossesBlocked** 0.0829437516623944 **Saves_OutOfBox** -0.107609112290232 **shotsSituations_Counter** -0.162027230276744 **ShotsAccuracy_OnPost** 0.0470456703888898 **GoalsSituations_Normal** 0.22052772131892 **PossesionLoss_Dispossessed** 0.00214167732977552 **PassesLength_InAccLB** 0.00409971082734128 **PassesType_AccCrn** 0.0741779917647036 **KeyPassesLength_Total** 0.0154845158237651

[148]:
```
nb_min = which.min(summary(choix_Espagne)$bic)
coef(choix_Espagne, nb_min)
```

(Intercept) 5.30226068030442 **Fouls_Fouls** 0.0603328099672517 **ShotsZones_SixYardBox** -0.0494355377391404 **ShotsBodyParts_RightFoot** -0.0951975313204938 **Goals_Total** 0.646177372605525 **GoalsSituations_Counter** 0.311904628461455 **Dribbles_Unsuccessful** -0.0773271313750928 **Assists_Corner** 0.162092345825822 **PassesType_AccFrK** 0.0438940115077385 **KeyPassesType_Corner** 0.445351698903317 **KeyPassesType_Throwin** 0.512586963670849

[149]:
```
nb_min = which.min(summary(choix_Argentine)$bic)
coef(choix_Argentine, nb_min)
```

(Intercept) 5.16725857068163 **Interception_Total** 0.0155505147131027 **Blocks_CrossesBlocked** 0.118954098176166 **ShotsAccuracy_OnTarget** 0.0553495653421166 **GoalsSituations_SetPiece** 0.158874580769146 **GoalsBodyParts_LeftFoot** 0.238693148373298 **Aerial_Total** 0.00628894471029627 **Assists_Total** 0.118270593028633 **PassesLength_Total** 0.000998176423661522 **KeyPassesType_Cross** 0.0280459864635727 **KeyPassesType_Throwin** 0.330646866627707

[150]:
```
nb_min = which.min(summary(choix_Angleterre)$bic)
coef(choix_Angleterre, nb_min)
```

(Intercept) 6.49408039603212 **Tackles_DribbledPast** -0.0586785202081505 **Tackles_TotalAttemptedTackles** 0.0274205807376198 **Fouls_Fouls** -0.0285663008528862 **GoalsSituations_OpenPlay** 0.102413063843859 **GoalsSituations_Normal** 0.235515646057419 **Dribbles_Total** 0.0115469947556096 **PossesionLoss_UnsuccessfulTouches** -0.0103487550216349 **Assists_Cross** 0.0600858505639187 **PassesType_AccFrK** -0.016677853678321 **KeyPassesLength_Long** 0.036631274846773

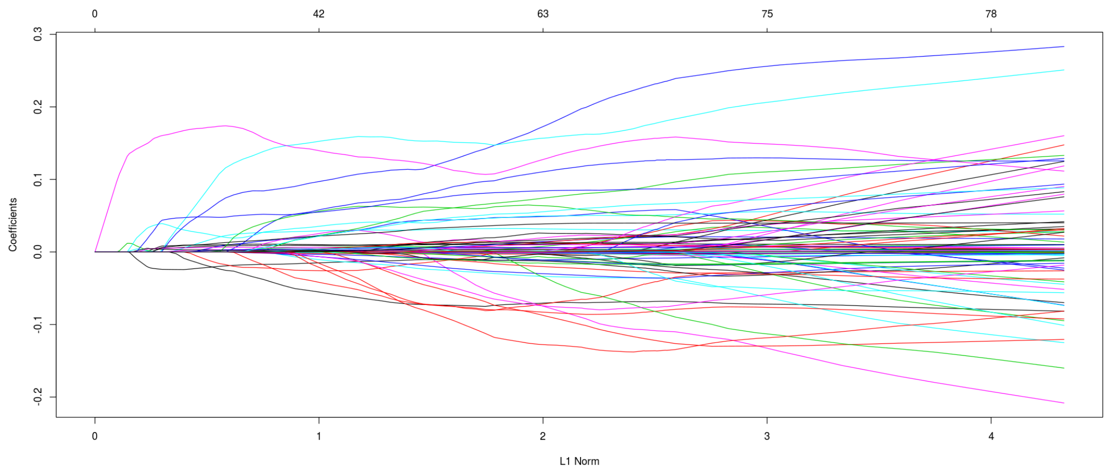
On remarque que c'est seulement en France, Allemagne et Espagne que la variable `Goals_Total` est sélectionnée.

1 Regression LASSO

alpha=1 is the lasso penalty, and alpha=0 the ridge penalty

[19]:
```
m_lasso = glmnet(as.matrix(data), data_tot$Rating, alpha = 1, nlambda = 100)
```

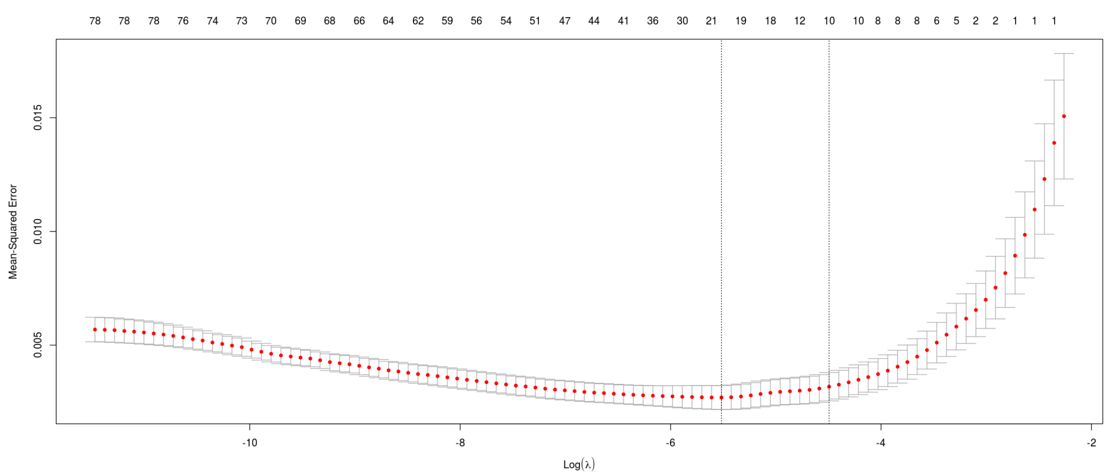
[20]:
```
plot(m_lasso)
```



Cross validation

[21]:
```
cv.out <- cv.glmnet(as.matrix(data), data_tot$Rating, alpha = 1)
```

[22]:
```
plot(cv.out)
```



[23]:
```
bestlam <- cv.out$lambda.min
```

[ ]:
```
predict(m_lasso, type = "coefficients", s = bestlam)
```

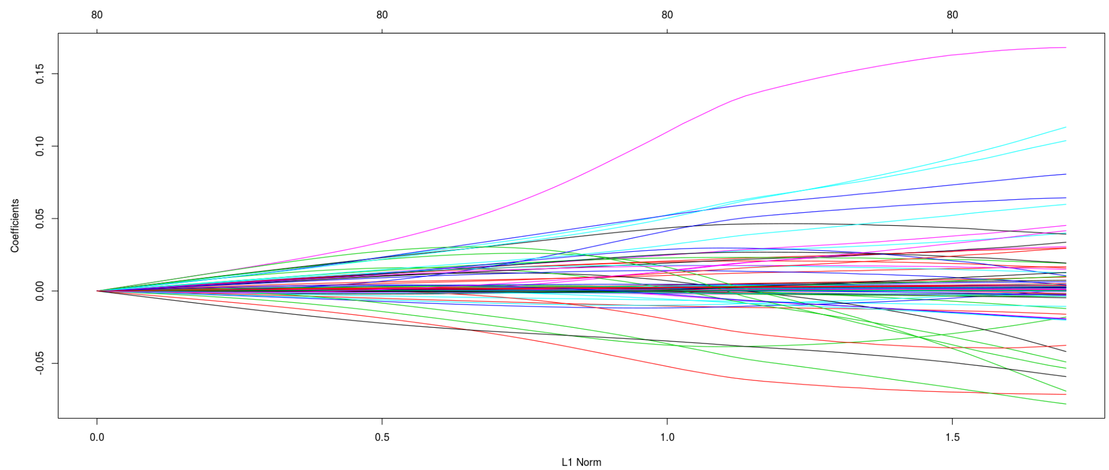
Les variables qu'il semble intéressant de retenir, d'après le modèle de régression Lasso, pour l'ensemble des données, sont : Tackles_TotalTackles ; Interception_Total ; Cards_Red ; Blocks_CrossesBlocked ; Blocks_PassesBlocked ; Saves_Total ; Saves_SixYardBox ; shotsSituations_Counter ; ShotsAccuracy_OffTarget ; GoalsZones_PenaltyArea ; GoalsSituations_PenaltyScored ; GoalsSituations_Own ; GoalsSituations_Normal ; Dribbles_Total ; PossesionLoss_UnsuccessfulTouches

; Aerial_Won ; PassesLength_Total ; PassesLength_AccLB ; PassesType_AccCrn ; KeyPassesLength_Short ; KeyPassesType_Throughball

1.1 Régression RIDGE

[25]:
```
m_ridge = glmnet(as.matrix(data), data_tot$Rating, alpha = 0, nlambda = 100)
```

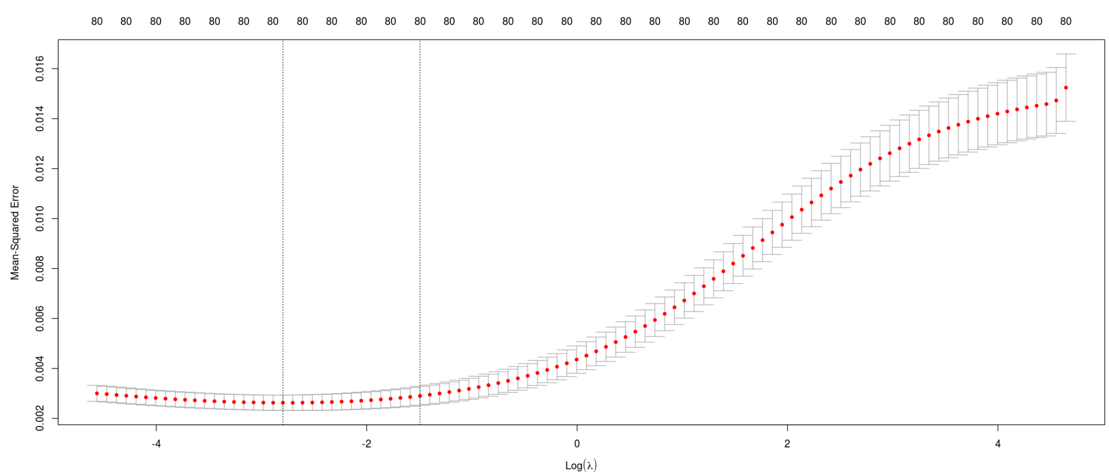
[26]:
```
plot(m_ridge)
```



Cross validation

[27]:
```
ridge.out <- cv.glmnet(as.matrix(data), data_tot$Rating, alpha = 0)
```

[28]:
```
plot(ridge.out)
```



[29]:
```
bestlam_ridge <- ridge.out$lambda.min
```

[30]:
```
p = predict(m_ridge, type = "coefficients", s = bestlam_ridge)
```

[31]:
```
print(p)
```

```
81 x 1 sparse Matrix of class "dgCMatrix"
                                          1
(Intercept)                    5.956521e+00
Tackles_TotalTackles           4.719710e-03
Tackles_DribbledPast          -8.126472e-04
Tackles_TotalAttemptedTackles  1.849083e-03
Interception_Total             3.795019e-03
Fouls_Fouled                   1.357866e-03
Fouls_Fouls                   -2.152593e-03
Cards_Yellow                   1.154406e-03
Cards_Red                     -5.986625e-02
OffSides_CaughtOffside         2.558104e-04
Clearances_Total               1.168195e-03
Blocks_ShotsBlocked           -7.040023e-03
Blocks_CrossesBlocked          1.882854e-02
Blocks_PassesBlocked           2.605218e-03
Saves_Total                   -1.119206e-02
Saves_SixYardBox              -3.841843e-02
Saves_PenaltyArea             -1.078249e-02
Saves_OutOfBox                -8.957045e-03
Shots_Total                    1.035017e-03
ShotsZones_OutOfBox           -1.318405e-03
ShotsZones_SixYardBox          1.216482e-02
ShotsZones_PenaltyArea         2.729240e-03
ShotsSituations_OpenPlay       1.687779e-03
shotsSituations_Counter        2.742786e-02
ShotsSituations_SetPiece      -6.121100e-03
ShotsSituations_PenaltyTaken   1.372436e-03
ShotsAccuracy_OffTarget        5.002412e-03
ShotsAccuracy_OnPost          -6.618545e-03
ShotsAccuracy_OnTarget         4.478433e-03
ShotsAccuracy_Blocked         -8.075487e-03
ShotsBodyParts_RightFoot       1.118899e-03
ShotsBodyParts_LeftFoot        1.230443e-03
ShotsBodyParts_Head            6.265552e-04
ShotsBodyParts_Other          -4.524922e-02
Goals_Total                    1.374912e-02
GoalsZones_SixYardBox          6.161324e-02
GoalsZones_PenaltyArea         2.800827e-02
GoalsZones_OutOfBox            4.605557e-02
GoalsSituations_OpenPlay       2.069039e-02
GoalsSituations_Counter        4.289289e-03
```

```
GoalsSituations_SetPiece            5.838741e-02
GoalsSituations_PenaltyScored       6.028632e-02
GoalsSituations_Own                 1.311188e-01
GoalsSituations_Normal              2.140778e-02
GoalsBodyParts_RightFoot            2.244766e-02
GoalsBodyParts_LeftFoot             2.318594e-02
GoalsBodyParts_Head                 2.963747e-02
Dribbles_Unsuccessful               3.839628e-03
Dribbles_Successful                 3.119941e-03
Dribbles_Total                      2.231146e-03
PossesionLoss_UnsuccessfulTouches   2.667633e-03
PossesionLoss_Dispossessed         -2.022484e-03
Aerial_Total                        4.325855e-04
Aerial_Won                          1.558463e-03
Aerial_Lost                         8.723345e-05
Assists_Cross                       2.666026e-02
Assists_Corner                     -2.753155e-02
Assists_Throughball                -1.119263e-03
Assists_Freekick                    4.934214e-02
Assists_Other                       1.748253e-02
Assists_Total                       1.759461e-02
PassesLength_Total                  7.167326e-05
PassesLength_AccLB                  9.674244e-04
PassesLength_InAccLB               -7.279697e-05
PassesLength_AccSP                  6.201766e-05
PassesLength_InAccSP                5.063930e-04
PassesType_AccCr                    8.908202e-04
PassesType_InAccCr                 -1.644001e-03
PassesType_AccCrn                   1.535835e-02
PassesType_InAccCrn                 3.265237e-03
PassesType_AccFrK                   1.415373e-04
PassesType_InAccFrK                -1.527146e-03
KeyPassesLength_Total               2.477093e-03
KeyPassesLength_Long                2.444908e-03
KeyPassesLength_Short               2.642783e-03
KeyPassesType_Cross                 4.121807e-04
KeyPassesType_Corner               -5.710652e-03
KeyPassesType_Throughball           3.750701e-02
KeyPassesType_Freekick              1.849280e-02
KeyPassesType_Throwin              -3.759308e-02
KeyPassesType_Other                 2.841899e-03
```

Ici c'est beaucoup moins évident de faire de la sélection de variables : les coefficients ne s'annulent pas. Certains sont cependant très petits ($1e-4$).
Si on ne souhaite garder que celles dont le coefficient est au moins de l'ordre de 10^{-2}, on peut citer :

(ancienne version erreur)
Cards_Red ; OffSides_CaughtOffside ; Blocks_ShotsBlocked ; Blocks_CrossesBlocked ;

Saves_Total ; Saves_SixYardBox ; ShotsZones_SixYardBox ; shotsSituations_Counter ; ShotsSituations_SetPiece ; ShotsSituations_PenaltyTaken ; ShotsAccuracy_OnPost ; ShotsAccuracy_Blocked ; ShotsBodyParts_Other ; GoalsZones_SixYardBox ; GoalsZones_PenaltyArea ; GoalsZones_OutOfBox ; GoalsSituations_OpenPlay ; GoalsSituations_Counter ; GoalsSituations_SetPiece ; GoalsSituations_PenaltyScored ; GoalsSituations_Own ; GoalsSituations_Normal ; GoalsBodyParts_RightFoot ; GoalsBodyParts_LeftFoot ; GoalsBodyParts_Head ; Assists_Cross ; Assists_Corner ; Assists_Throughball ; Assists_Freekick ; Assists_Other ; Assists_Total ; PassesType_AccCrn ; PassesType_InAccCrn ; KeyPassesLength_Long ; KeyPassesType_Corner ; KeyPassesType_Throughball ; KeyPassesType_Freekick ; KeyPassesType_Throwin

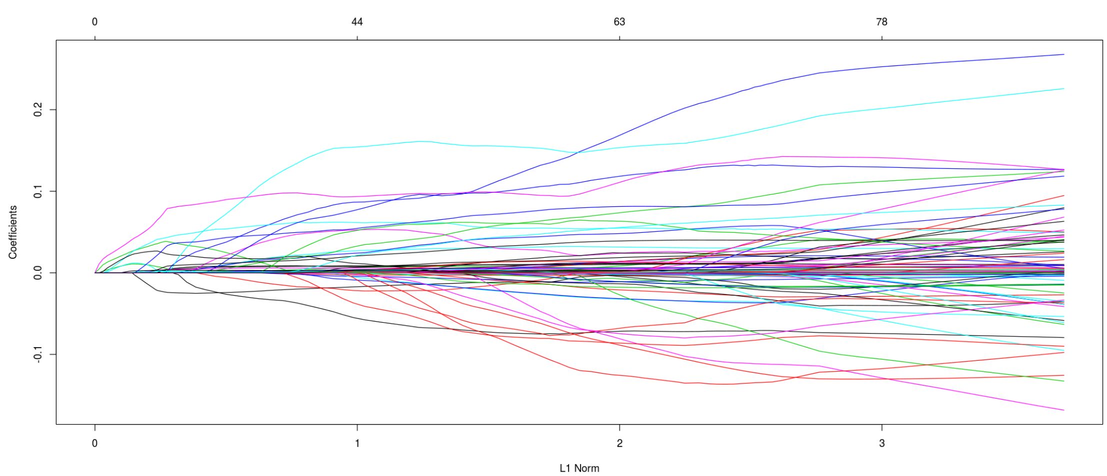
correction :

Cards_Red ; ShotsBodyParts_Other ; Saves_SixYardBox ; KeyPassesType_Throwin ; Assists_Corner ; Saves_Total ; Saves_PenaltyArea ; ShotsZones_SixYardBox ; Goals_Total ; PassesType_AccCrn ; Assists_Other ; Assists_Total ; KeyPassesType_Freekick ; Blocks_CrossesBlocked ; GoalsSituations_OpenPlay ; GoalsSituations_Normal ; GoalsBodyParts_RightFoot ; GoalsBodyParts_LeftFoot ; Assists_Cross ; shotsSituations_Counter ; GoalsZones_PenaltyArea ; GoalsBodyParts_Head ; KeyPassesType_Throughball ; GoalsZones_OutOfBox ; Assists_Freekick ; GoalsSituations_SetPiece ; GoalsSituations_PenaltyScored ; GoalsZones_SixYardBox ; GoalsSituations_Own

1.2 Régression Elastic Net

[32]:
```
m_enet = glmnet(as.matrix(data), data_tot$Rating, alpha = 0.5, nlambda = 100)
```

[33]:
```
plot(m_enet)
```

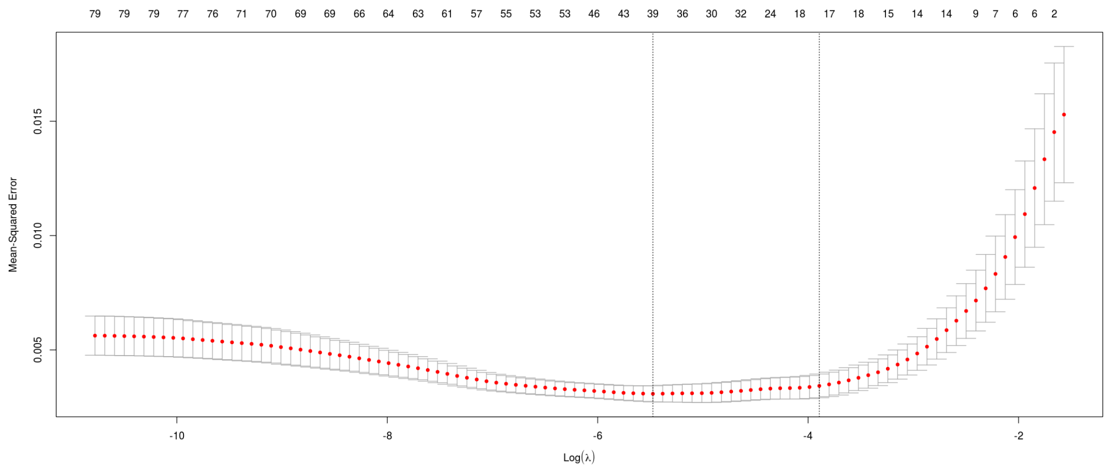


Cross validation

[34]:
```
enet.out <- cv.glmnet(as.matrix(data), data_tot$Rating, alpha = 0.5)
```

[35]:
```
bestlam_enet <- enet.out$lambda.min
```

[36]:
```
plot(enet.out)
```



[37]:
```
predict(m_enet, type = "coefficients", s = bestlam_enet)
```

```
81 x 1 sparse Matrix of class "dgCMatrix"
                                            1
(Intercept)                      5.851398e+00
Tackles_TotalTackles             8.392730e-03
Tackles_DribbledPast             .
Tackles_TotalAttemptedTackles    .
Interception_Total               5.299592e-03
Fouls_Fouled                     8.653163e-04
Fouls_Fouls                     -6.433979e-04
Cards_Yellow                     .
Cards_Red                       -4.514466e-02
OffSides_CaughtOffside           .
Clearances_Total                 1.026858e-03
Blocks_ShotsBlocked             -3.464942e-03
Blocks_CrossesBlocked            2.478552e-02
Blocks_PassesBlocked             2.435930e-03
Saves_Total                     -1.858042e-02
Saves_SixYardBox                -1.862335e-02
Saves_PenaltyArea                .
Saves_OutOfBox                   .
Shots_Total                      .
ShotsZones_OutOfBox              .
ShotsZones_SixYardBox            2.437044e-03
ShotsZones_PenaltyArea           .
ShotsSituations_OpenPlay         .
shotsSituations_Counter          2.167518e-02
ShotsSituations_SetPiece         .
ShotsSituations_PenaltyTaken     .
ShotsAccuracy_OffTarget          7.648981e-03
```

```
ShotsAccuracy_OnPost                     .
ShotsAccuracy_OnTarget                   .
ShotsAccuracy_Blocked                   -5.789549e-03
ShotsBodyParts_RightFoot                 .
ShotsBodyParts_LeftFoot                  .
ShotsBodyParts_Head                      .
ShotsBodyParts_Other                    -2.139722e-02
Goals_Total                              .
GoalsZones_SixYardBox                    8.055209e-02
GoalsZones_PenaltyArea                   6.187840e-02
GoalsZones_OutOfBox                      5.050151e-02
GoalsSituations_OpenPlay                 1.621027e-02
GoalsSituations_Counter                  .
GoalsSituations_SetPiece                 2.308054e-02
GoalsSituations_PenaltyScored            6.892946e-02
GoalsSituations_Own                      1.496238e-01
GoalsSituations_Normal                   9.472339e-02
GoalsBodyParts_RightFoot                 .
GoalsBodyParts_LeftFoot                  .
GoalsBodyParts_Head                      .
Dribbles_Unsuccessful                    .
Dribbles_Successful                      .
Dribbles_Total                           5.803370e-03
PossesionLoss_UnsuccessfulTouches        2.177396e-03
PossesionLoss_Dispossessed              -1.841598e-03
Aerial_Total                             .
Aerial_Won                               3.184567e-03
Aerial_Lost                              .
Assists_Cross                            .
Assists_Corner                           .
Assists_Throughball                      .
Assists_Freekick                         4.275328e-02
Assists_Other                            .
Assists_Total                            .
PassesLength_Total                       1.967232e-04
PassesLength_AccLB                       5.335410e-04
PassesLength_InAccLB                     .
PassesLength_AccSP                       3.885485e-05
PassesLength_InAccSP                     .
PassesType_AccCr                         .
PassesType_InAccCr                       .
PassesType_AccCrn                        1.737853e-02
PassesType_InAccCrn                      .
PassesType_AccFrK                        .
PassesType_InAccFrK                      .
KeyPassesLength_Total                    2.334731e-03
KeyPassesLength_Long                     .
KeyPassesLength_Short                    3.133755e-03
```

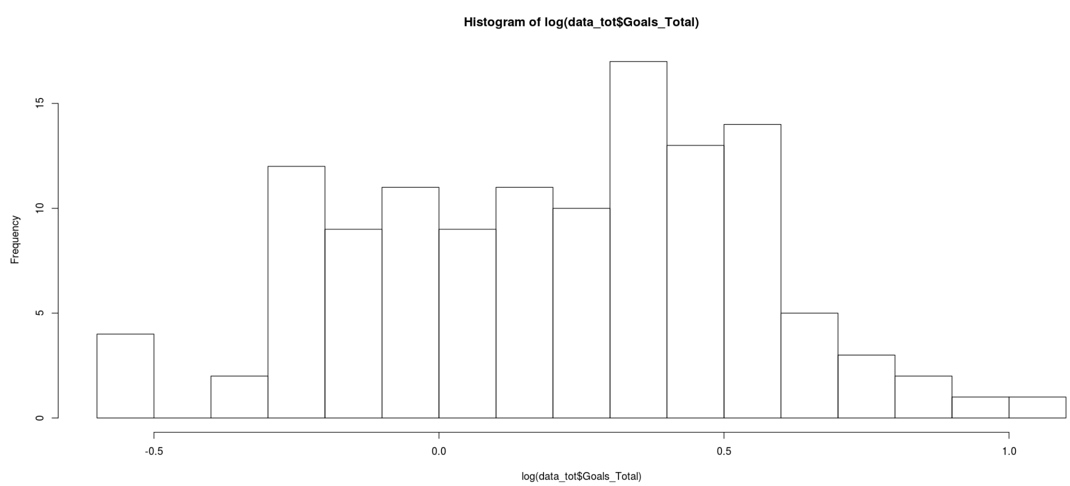
```
KeyPassesType_Cross                .
KeyPassesType_Corner               .
KeyPassesType_Throughball          5.187752e-02
KeyPassesType_Freekick             1.068406e-02
KeyPassesType_Throwin             -2.980732e-04
KeyPassesType_Other                1.861048e-03
```

2 Variable à expliquer = nombre de buts sur la totalité des matches

On fait le choix d'expliquer, en utilisant une régression de Poisson le nombre de buts marqués par une équipe sur 100 matches. Ce choix a été fait car une regression de Poisson permet d'expliquer une variable de comptage, donc entière. Or, on dispose des moyennes de buts marqués par match, ce qui est logique car tous les championnats n'ont pas autant de matches. Cependant, cette moyenne n'est pas un nombre entier. On va donc multiplier par le nombre de matches joués les moyennes empiriques de buts par saison par équipe.

[62]:
```
buts = trunc(data_tot$Goals_Total * data_tot$Nombre)
```

[64]:
```
hist(log(data_tot$Goals_Total), breaks=12)
```



On peut assimiler le log de la variable Buts à la répartition d'une loi normale.

[70]:
```
data_but = data_tot[,-c(34:46,81,82,83,84)]
```

[71]:
```
fit.add = glm(buts~. , data=data_but, family=poisson)
```

[ ]:
```
s = step(fit.add)
```

[73]:
```
s$coefficients
```

(Intercept) 2.30452561452045 **Tackles_TotalTackles** 0.0256914763679399 **Blocks_ShotsBlocked** 0.0926843496287709 **Blocks_PassesBlocked** -0.0547146702077624 **Saves_Total** 0.11325392255371 **Saves_OutOfBox** -0.371303584536457 **Shots_Total**

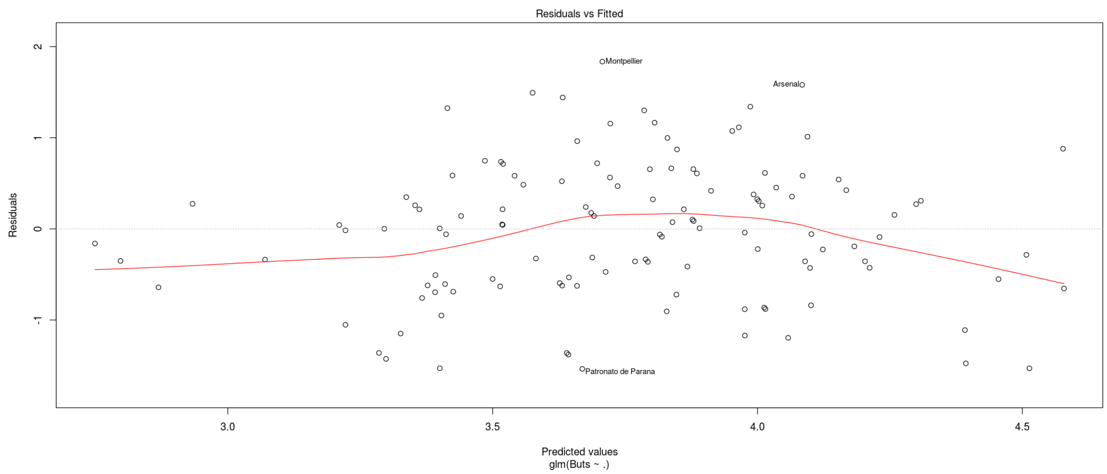
0.101432305700044 **ShotsZones_SixYardBox** 0.134113232438393 **ShotsSituations_PenaltyTaken** 0.601762612150438 **ShotsAccuracy_OffTarget** -0.220785670330016 **ShotsAccuracy_OnPost** 0.252740204468019 **PossesionLoss_UnsuccessfulTouches** -0.0243267135807765 **Aerial_Total** -0.592380422841699 **Aerial_Won** 0.601661181379041 **Aerial_Lost** 0.590886224609563 **Assists_Cross** 0.4761685378259 **Assists_Throughball** 0.968536158784699 **Assists_Other** 0.60778132725629 **PassesLength_Total** -0.000640868346658859 **PassesLength_AccLB** 0.00792215387936705 **PassesType_AccFrK** 0.0434951001029664 **KeyPassesType_Cross** 0.0867929968286926

[74]:
```
data_add_poisson = data.frame(buts, data_tot$Blocks_ShotsBlocked,␣
↪data_tot$Blocks_PassesBlocked,
                        data_tot$Saves_Total,
                        data_tot$Saves_OutOfBox, data_tot$Shots_Total,
                        data_tot$ShotsAccuracy_OffTarget, data_tot$Assists_Cross,
                        data_tot$Assists_Throughball, data_tot$Assists_Other,␣
↪data_tot$PassesLength_Total,
                        data_tot$PassesType_AccFrK, data_tot$ShotsAccuracy_OnPost,
                        data_tot$Tackles_TotalTackles,␣
↪data_tot$ShotsZones_SixYardBox, data_tot$ShotsSituations_PenaltyTaken,
                        data_tot$PossesionLoss_UnsuccessfulTouches,␣
↪data_tot$Aerial_Total, data_tot$Aerial_Won, data_tot$Aerial_Lost,
                        data_tot$PassesLength_AccLB, data_tot$KeyPassesType_Cross)

names(data_add_poisson) <- c("Buts", "Blocks_ShotsBlocked",␣
↪"Blocks_PassesBlocked", "Saves_Total",
                        "Saves_OutOfBox", "Shots_Total",
                        "ShotsAccuracy_OffTarget",
                        "Assists_Cross", "Assists_Throughball", "Assists_Other",
                        "PassesLength_Total", "PassesType_AccFrK",␣
↪"ShotsAccuracy_OnPost",
                        "Tackles_TotalTackles", "ShotsZones_SixYardBox",␣
↪"ShotsSituations_PenaltyTaken",
                        "PossesionLoss_UnsuccessfulTouches", "Aerial_Total",␣
↪"Aerial_Won", "Aerial_Lost",
                        "PassesLength_AccLB", "KeyPassesType_Cross")
```

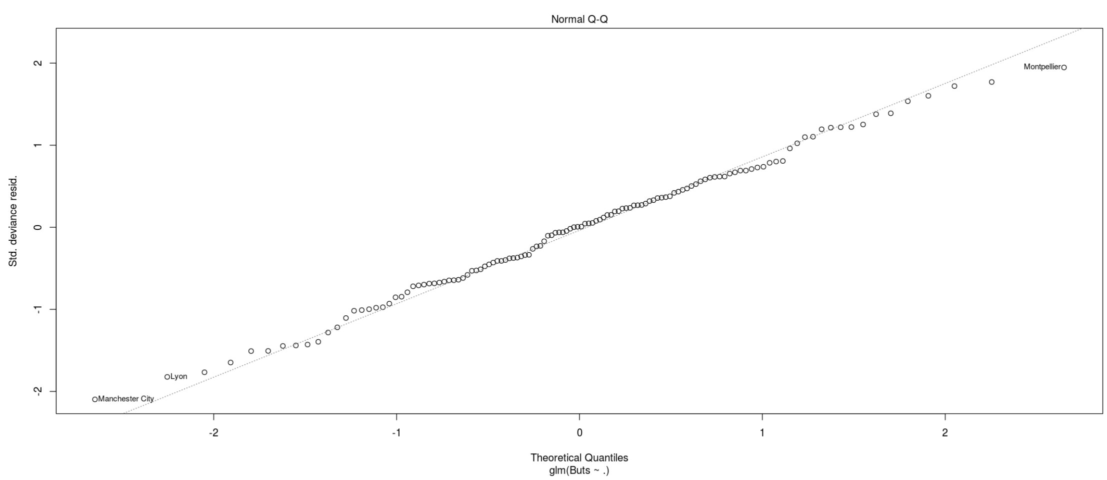
[75]:
```
rownames(data_add_poisson) = rownames(data_tot)
```

[76]:
```
fit_poisson = glm(Buts~., data = data_add_poisson, family = poisson)
```

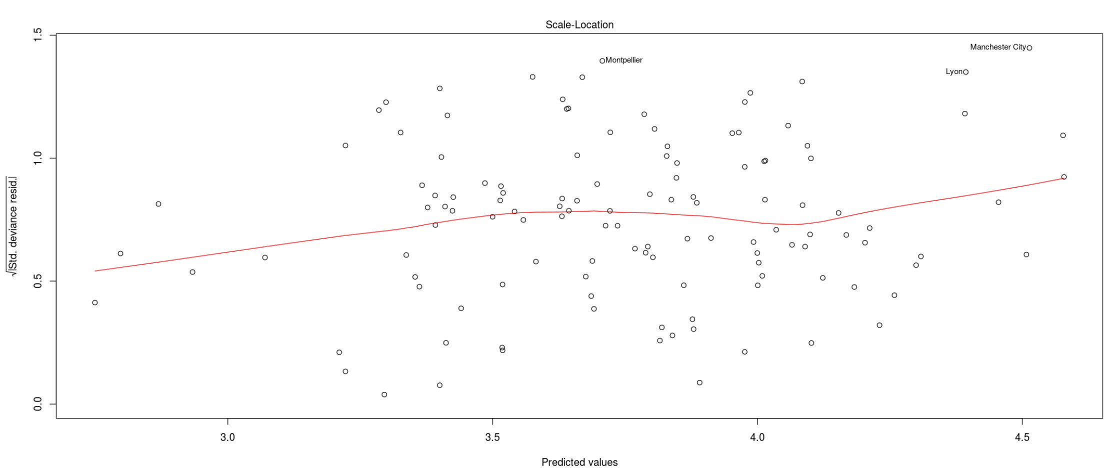
[79]:
```
plot(fit_poisson)
```



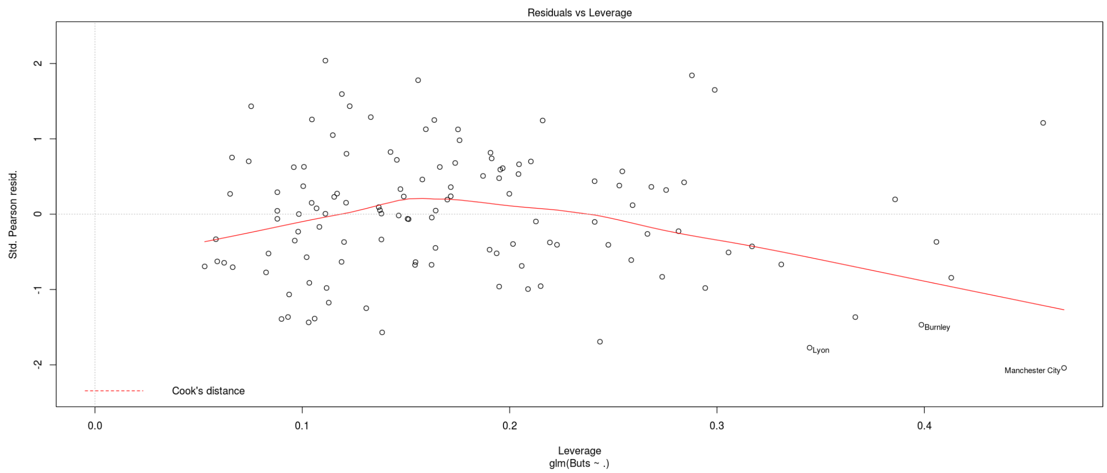
Residuals vs Fitted
Montpellier
Arsenal
Patronato de Parana
Residuals
Predicted values
glm(Buts ~ .)



Normal Q-Q
Montpellier
Lyon
Manchester City
Std. deviance resid.
Theoretical Quantiles
glm(Buts ~ .)



Scale-Location
Montpellier
Manchester City
Lyon
√|Std. deviance resid.|
Predicted values
glm(Buts ~ .)



[84]:
```
R_squared = 1 - fit_poisson$deviance/fit_poisson$null.deviance
print(paste("R^2 du modèle de poisson :", R_squared))
```

```
[1] "R^2 du modèle de poisson : 0.911716331492632"
```

Le R^2 est très proche de 1 donc le modèle de régression de Poisson additif semble bien expliquer la variable du nombre de buts.

Ces résultats ne sont cependant pas représentatifs de la réalité : on a utilisé le nombre de buts total, dépendant du nombre de matches joués, mais on a utiliser les valeurs moyennées sur le reste des données. On va donc refaire l'opération, avec des variables comptées sur l'ensemble de la saison.

[108]:
```
data_but_tot = data_but * data_tot$Nombre
```

[109]:
```
fit.add_tot = glm(buts~. , data=data_but_tot, family=poisson)
```

[ ]:
```
s = step(fit.add_tot)
```

[152]:
```
s$coefficients
```

(Intercept) 2.30791575555754 **Saves_Total** 0.00286040661310479 **Saves_SixYardBox** -0.0101805250933053 **Saves_OutOfBox** -0.00367989518126465 **Shots_Total** 0.00449424799738137 **ShotsAccuracy_OffTarget** -0.00477485493404608 **ShotsAccuracy_Blocked** -0.00475952723234816 **ShotsBodyParts_LeftFoot** -0.000916358770659708 **Aerial_Total** -0.000223939626730078 **Assists_Total** 0.0176475908588984 **PassesLength_Total** 0.000166355805657692 **PassesLength_AccSP** -0.000187632114451 35 **PassesType_AccFrK** 0.000429703540661475 **KeyPassesType_Freekick** 0.00435990414305181 **KeyPassesType_Throwin** -0.010693203461035

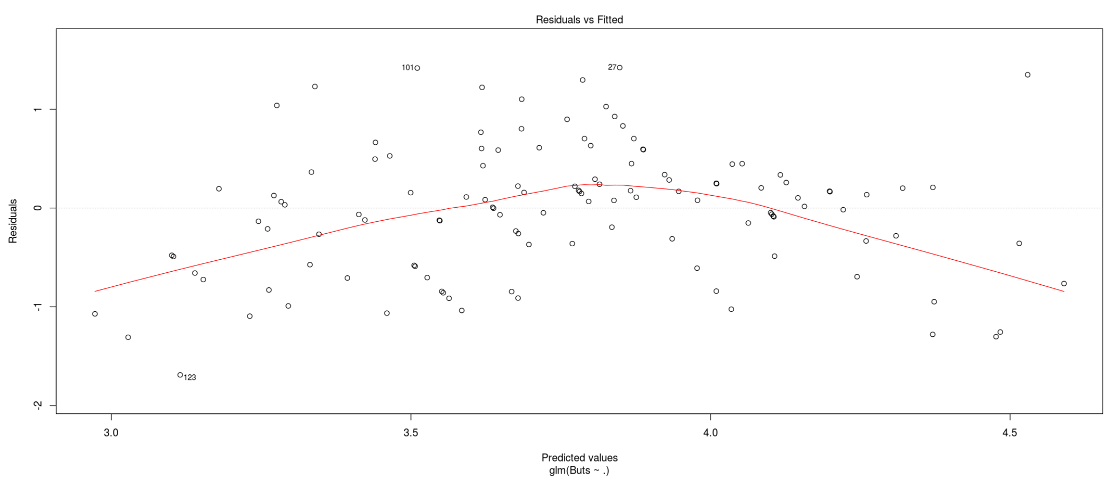
[94]:
```
data_add_poisson_tot = data.frame(buts,
                        data_but_tot$Saves_Total,
                        data_but_tot$Saves_OutOfBox, data_but_tot$Shots_Total,
```

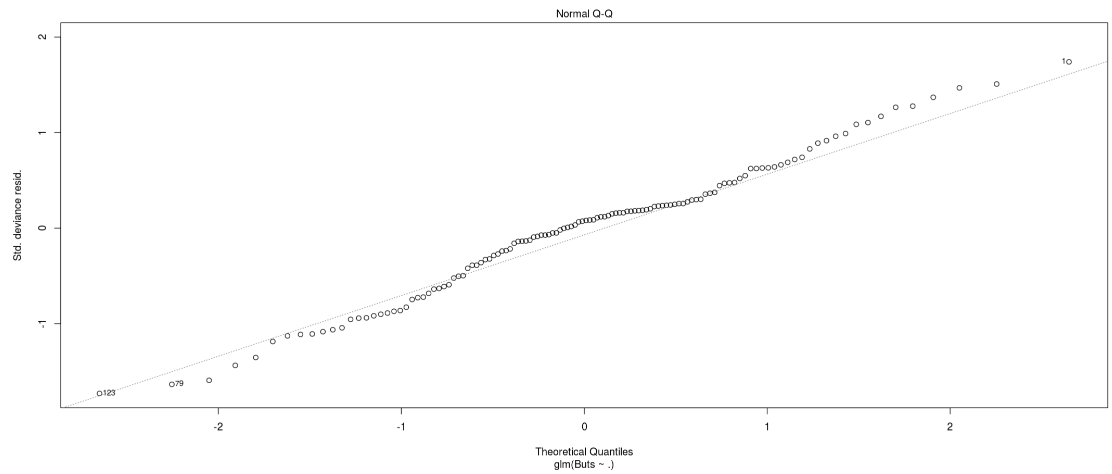
```
                     data_but_tot$ShotsAccuracy_OffTarget,␣
↪data_but_tot$PassesLength_Total,
                     data_but_tot$PassesType_AccFrK,
                     data_but_tot$Aerial_Total, data_but_tot$Saves_SixYardBox,␣
↪data_but_tot$ShotsAccuracy_Blocked,
                     data_but_tot$ShotsBodyParts_LeftFoot,␣
↪data_but_tot$Assists_Total, data_but_tot$PassesLength_AccSP,
                     data_but_tot$KeyPassesType_Freekick,␣
↪data_but_tot$KeyPassesType_Throwin)

names(data_add_poisson_tot) <- c("Buts", "Saves_Total",
                     "Saves_OutOfBox", "Shots_Total",
                     "ShotsAccuracy_OffTarget",
                     "PassesLength_Total", "PassesType_AccFrK",
                     "Aerial_Total", "Saves_SixYardBox",␣
↪"ShotsAccuracy_Blocked", "ShotsBodyParts_LeftFoot",
                     "Assists_Total", "PassesLength_AccSP",␣
↪"KeyPassesType_Freekick", "KeyPassesType_Throwin")
```

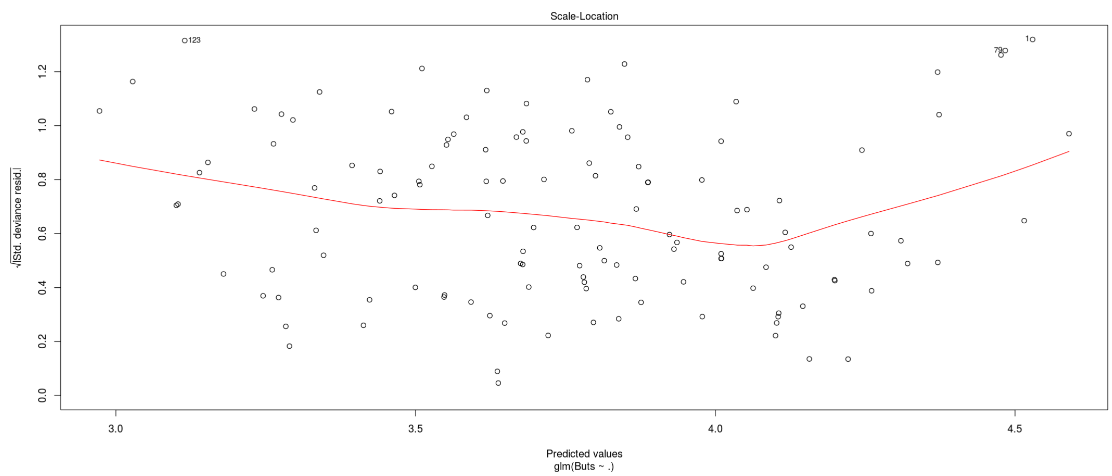
[ ]:
```
rownames(data_add_poisson_tot) = rownames(data_tot)
```

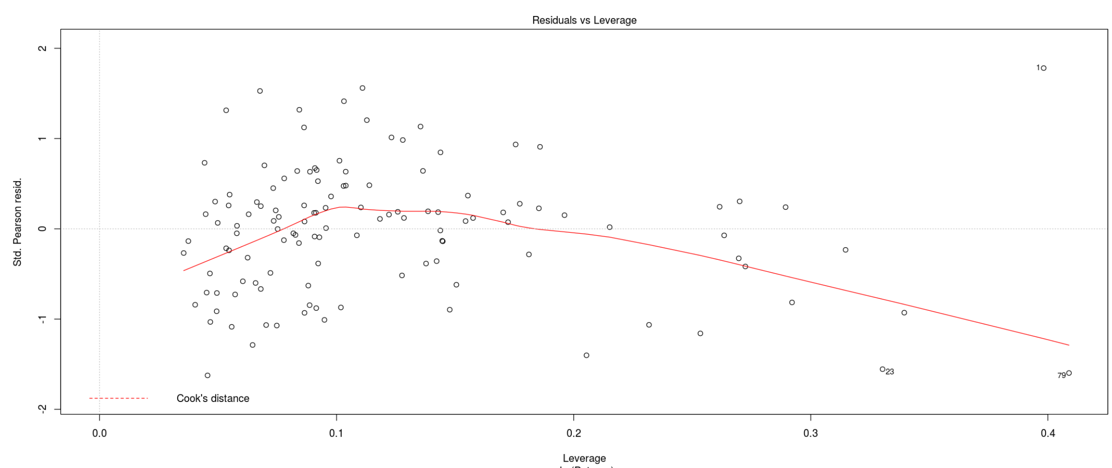
[95]:
```
fit_poisson_tot = glm(Buts~., data = data_add_poisson_tot, family = poisson)
```

[97]:
```
plot(fit_poisson_tot)
```




Normal Q-Q
Std. deviance resid.
Theoretical Quantiles
glm(Buts ~ .)
123
79
1


Scale-Location
√|Std. deviance resid.|
Predicted values
glm(Buts ~ .)
123
79
1


Residuals vs Leverage
Std. Pearson resid.
Leverage
glm(Buts ~ .)
Cook's distance
1
23
79

[99]:
```
R_squared = 1 - fit_poisson_tot$deviance/fit_poisson_tot$null.deviance
print(paste("R^2 du modèle de poisson :", R_squared))
```

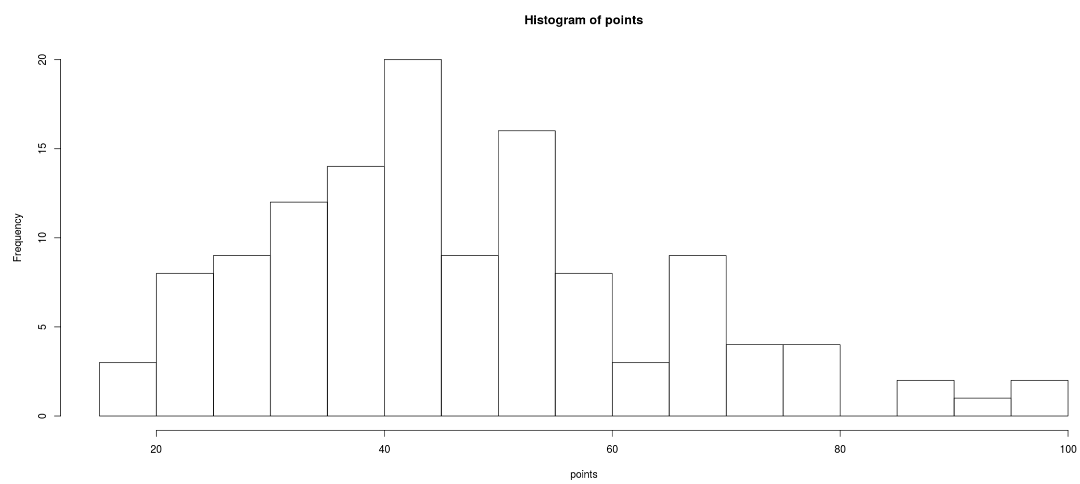
```
[1] "R^2 du modèle de poisson : 0.93401160079489"
```

Le R^2 est très proche de 1 donc le modèle de régression de Poisson additif semble bien expliquer la variable du nombre de buts. C'est bizarre que ce R^2 soit meilleur que le précédent, car on observe des formes dans les résidus (une banane).

3 Variable = nombre de points à l'issue de la saison

[110]:
```
points = data_tot$Points
```

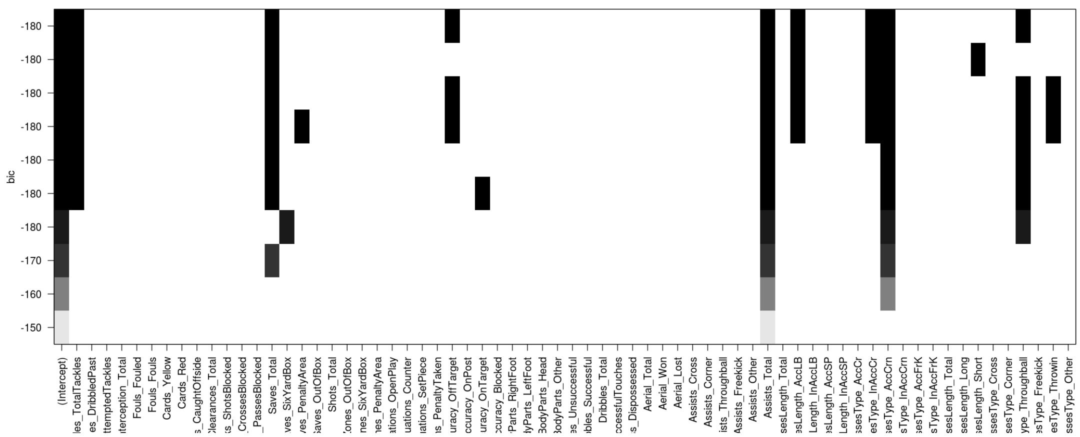
[132]:
```
hist(points, breaks=12)
```



[111]:
```
data_but_tot$Points = points
```

[113]:
```
choix_points <- regsubsets(data_but_tot$Points~., data = data_but_tot, nbest=1,␣
↪nvmax=10, method="seqrep")
```

[116]:
```
options(repr.plot.width=18, repr.plot.height=8)
plot(choix_points,scale="bic")
```



```
[117]: nb_min = which.min(summary(choix_points)$bic)
coef(choix_points, nb_min)
```

(Intercept) 1.53482745865072 **Tackles_TotalTackles** 0.026729570042122 **Saves_Total** -0.158339020183132 **ShotsAccuracy_OffTarget** 0.0762969643304097 **Assists_Total** 0.646769161318233 **PassesLength_AccLB** 0.0130399143603524 **PassesType_InAccCr** -0.0357919576420787 **PassesType_AccCrn** 0.162860763925574 **KeyPassesType_Throughball** 0.307728920847933

Les variables influentes sur le nombre de points sont les variables ci-dessus.

```
[118]: data_bic_points = data.frame(points,
                          data_but_tot$Tackles_TotalTackles,
                          data_but_tot$Saves_Total,␣
 ↪data_but_tot$ShotsAccuracy_OffTarget,
                          data_but_tot$Assists_Total,␣
 ↪data_but_tot$PassesLength_AccLB,
                          data_but_tot$PassesType_InAccCr,
                          data_but_tot$PassesType_AccCrn,␣
 ↪data_but_tot$KeyPassesType_Throughball)

names(data_bic_points) <- c("Points", "Tackles_TotalTackles",
                          "Saves_Total", "ShotsAccuracy_OffTarget",
                          "Assists_Total",
                          "PassesLength_AccLB", "PassesType_InAccCr",
                          "PassesType_AccCrn", "KeyPassesType_Throughball")
```

```
[120]: rl_points = lm(formula = Points~.,data = data_bic_points)
```

```
[128]: r_squared = summary(rl_points)$r.squared
print(paste("R^2 pour un modéle linéaire :", r_squared))
```

```
[1] "R^2 pour un modéle linéaire : 0.837341705382205"
```

Le R^2 est moins bon que losqu'on explique le nombre de buts avec une régression loglinéaire. On s'en doutait un peu car le fait de calculer les points au classement n'est pas équivalent au nombre de buts. Ca ne dépend que de l'issue du match, et pas de son déroullement.

Peut être qu'une autre régression serait à envisager, mais on ne sait pas laquelle.

[ ]:

```
```