Linear Models (LSTAT2120) - Project

Bartolomeo Aurèle, Gengler Rémi, Lamy Lionel

January 2021

Introduction and objective

Note: all the tables, figures and the code used in this project can be found on github.

For this project, we will use a dataset that contains all the FIFA 19 player characteristics (FIFA is a football simulation video game, and our dataset come from the 2019's edition). This dataset contains `18059` observations (all the different players created in the game) and `34` variables (characteristics such as Age, Nationality, Wage, Stamina, ...).

The purpose of our project is to predict the wage of a player based on all his features. This is a very interesting question because it could help club managers to know if some of their players are being underpaid (that would push the player to leave) or overpaid (that could be a threat to club finances). It could also help recruiters to have an idea of the price to pay for a player, depending on the features he is looking for. Finally, it would be a helpful tool for young players, for example to know on which feature they have to work in order to have their wage increased.

For this purpose we will use linear models only. So for a start we will look more precisely at our dataset and see if the classical assumptions for linear models are respected. That is, we check if

1. The X matrix of explanatory variables is of full rank (i.e. $rk(X) = p$)
2. X is a fixed matrix
3. The error terms $\epsilon_1, \ldots, \epsilon_n$ are independent
4. $E[\epsilon] = 0$
5. $Var(\epsilon) = \sigma^2 I_n$

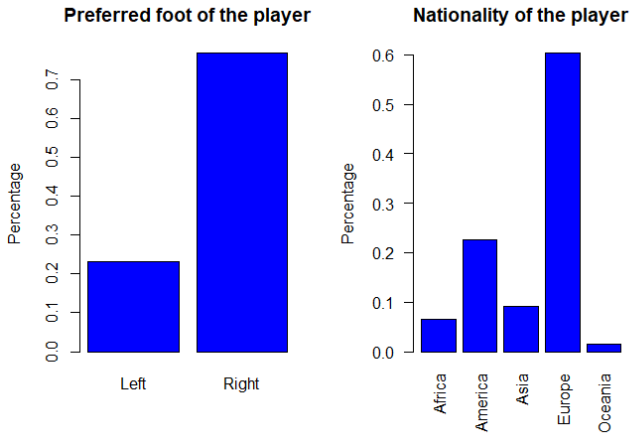
We will also check for nonlinearity, influential observations, multicollinearity (this is linked to assumption 1), heteroskedasticity (this is assumption 5) and autocorrelation (this is linked to assumption 3). All these concepts will be reexplained in the appropriate section. If some hypotheses seem clearly not satisfied, we will then take remedial actions. We will retain different models based on several regression methods seen in course and will compare them based on robust criterions (for example, their ability to predict correctly the value of the wage for a completely new player).

Data cleaning and separtion of the dataset

We start by doing a complete case analysis (i.e. eliminating all observations that contain missing values). We can do that without introducing any bias in our analysis, because observations containing missing values constitute an extremely small portion of our dataset (`48 of 18207` observations, so less than `0.3%`). We then improve the usability of the dataset by changing some units. Initially, the variable *Nationality* that records the nationality of the player was encoded as the country of the player. Since there were too much levels and some of them seemed to be irrelevant, we decided to modify it manually and replace these levels by the associated continents. Finally, we separated randomly `20%` of our observations. These observations will be used for prediction, but not for model estimation.

Descriptive analysis

Among the 34 variables, 27 take values in percentage : it mainly concerns the game features such as agility, dribbling, ball control... The other quantitative variables are the age (given in years), the wage (given in thousands of euros), the height (given in centimeters), the weight (given in kilograms) and the value of the player (given in million of euros). We should also emphasize the fact that we have two categorical variables called *Preferred_Foot* (with two levels : Left and Right) and *Nationality* (with five levels : Africa, America, Asia, Europe and Oceania). We can look at the two corresponding barplots.

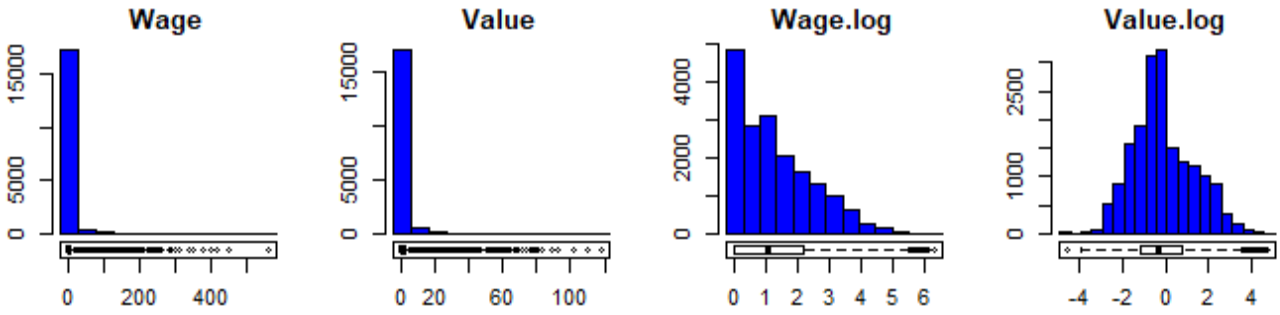


We observe that right-footed players are four times as many as left-footed players. We also see that more than half of the players are European, while the 4 other continents don't exceed 25% individually.

Now we can take a look at the table below that shows some interesting descriptive values about our quantitative variables. More precisely, we have calculated the mean, the standard deviation, the skewness and the kurtosis for each of them. It would be too massive to describe precisely each result of this table since we have a lot of variables, but we will try to outline the most important ones.

Let's start with the *Age* variable. Players age have mean `25`, with a `4.7` standard deviation. We see that the variable is right skewed (nothing surprising, some players can still play at 40 but none can play at 5). The kurtosis is slightly less than 3, which means that the variable is maybe a bit light-tailed.

Now we can look at our variable of interest, the wage. It seems that it is not at all following a normal distribution. In fact, we have a a skewness of `7.9` which means that it is very right-tailed. Furthermore, it has a kurtosis of `102` so the *Wage* variable is extremely thick in the tails. We can back up this claim by looking at the histogram and boxplot of the wage. This variable is definitely not normal.



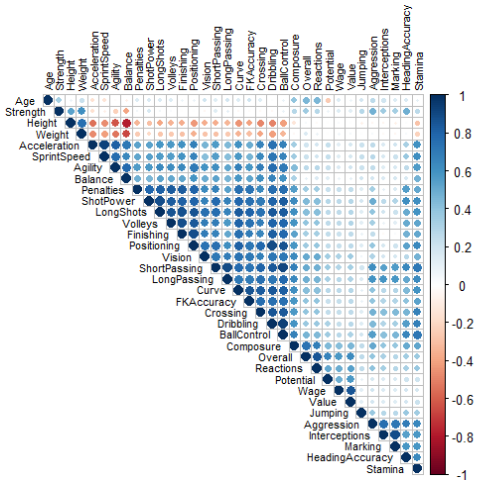
Similarly from the *Wage* variable, we see in the table that the value of the player is also right-skewed and very thick in its tails. The histogram/boxplot figure of the *Value* variable above confirms this tendency.

We can see in the histograms above that, if we take the log of these two variables (*Wage* and *Value*), we obtain much more suitable distributions. The distribution is less agglutinated near zero, and the distribution seems even normally distributed.

Weight and height of the players seem to be normally distributed (maybe a bit right-skewed for the weight) around `75kg` and `181cm`, respectively.

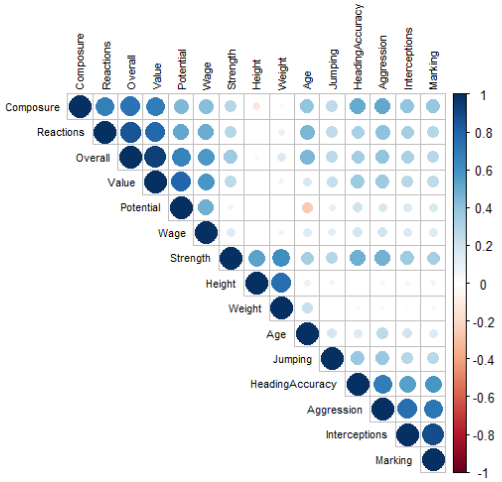
The remaining variables are all about players abilities. Like we have already said, these variables take values between 0 and 100. We will not describe each of them specifically, since they are all quite similar. In general, they have mean between 45 and 65 and standard deviation between 10 and 20. In addition, they have skewness near 0 and kurtosis near 3.

We now look at the correlation matrix. The aim of this step is to detect high pairwise correlations (and therefore multicollinearity). Note that even if there were no pairwise correlations, it would not mean that there is no multicollinearity. In fact, there can be multicollinearity invisible in the correlation matrix because multicollinearity can come frome linear dependance between 3 or more variables.



We have here the upper part of the correlation matrix. We see that we have a triangle of variables that have very high pairwise correlations (lots of those correlations are even superior to `0.8`). We are not very surprised of that, because we could expect that good players have globally good abilities and conversely, bad players have on average bad abilities. Thus, those variables contains almost the same information as what we already have in the *Overall* variable. Considering this fact, we decided to discard those variables in order to avoid too much multicollinearity problems.

After this suppression, we see that the following correlation matrix is better, in the sense that we have no longer an enormous block of high correlated variables.



Again, this does not mean that we have no multicollinearity at all. We will look after the model selection, with more sophisticated tools.

Model Selection

We will now start to select variables to include in our model. We will use the two following types of model selection: a stepwise regression using Akaike information criterion (AIC) and a LASSO procedure.

Stepwise Regression using AIC

We will use a stepwise function that finds a model that minimizes AIC, using a step-by-step approach. As a reminder, the AIC, just like the adjusted R-squared, is a measure of the quality of the model that penalizes the

addition of new variables. Let $\hat{L}$ be the maximized value of the likelihood function, so the lower the AIC, the better the goodness of fit of the model.

$$AIC = 2k - 2\log(\hat{L})$$

This is what we need in order to do a stepwise regression. The method will check whether the AIC keeps decreasing while we include or remove more and more variables. This is what we need in order to do a stepwise regression. The method will check whether the AIC keeps decreasing while we include more and more variables.

In the stepwise summary in the appendix we see that the selection discards 9 variables : Overall, Weight, Jumping, Interceptions, Composure, Wage, NationalityAmerica, NationalityOceania, Preferred_FootRight.

LASSO Procedure

The LASSO estimator is comparable to the OLS in the sense that the goal is to minimize the sum of squared residuals. But the main difference lies in the fact that it imposes a constraint on the L1 norm of the model parameter β. Indeed, for a certain coefficient $t > 0$ to be determined, we impose that

$$\sum_{j=1}^{p} |\beta_j| \leq t$$

We can see the results in the appendix.

Final Model (without interactions)

The table above shows a model comparison for those 3 models. We use various criteria such as the log-likelihood, the AIC, the adjusted R-squared and the deviance. We decide to take as final model the Lasso log(Y) model with log of the *Wage* variable because it has almost as good goodness of fit criteria as the Complete model, but it has only 7 variables. The final model is then given by -1.004 (Intercept) + 0.029 Age + 0.022 Overall + 0.004 Reactions + 0.001 Composure + 0.57 Value + -0.115 NationalityAmerica + 0.153 NationalityAsia.

Interactions

We test different possible interactions in the following : we add each time an interaction term in the basis model (final model) and then look at the properties of the new model. The first (1) model with interaction contains an interaction term between the *Preferred_FootLeft* variable and the *Marking* variable. We then consider interaction between *Preferred_FootLeft* and *Value* (2), interaction between *NationalityEurope* and *Potential* (3) and interaction between *NationalityAfrica* and *Value* (4).

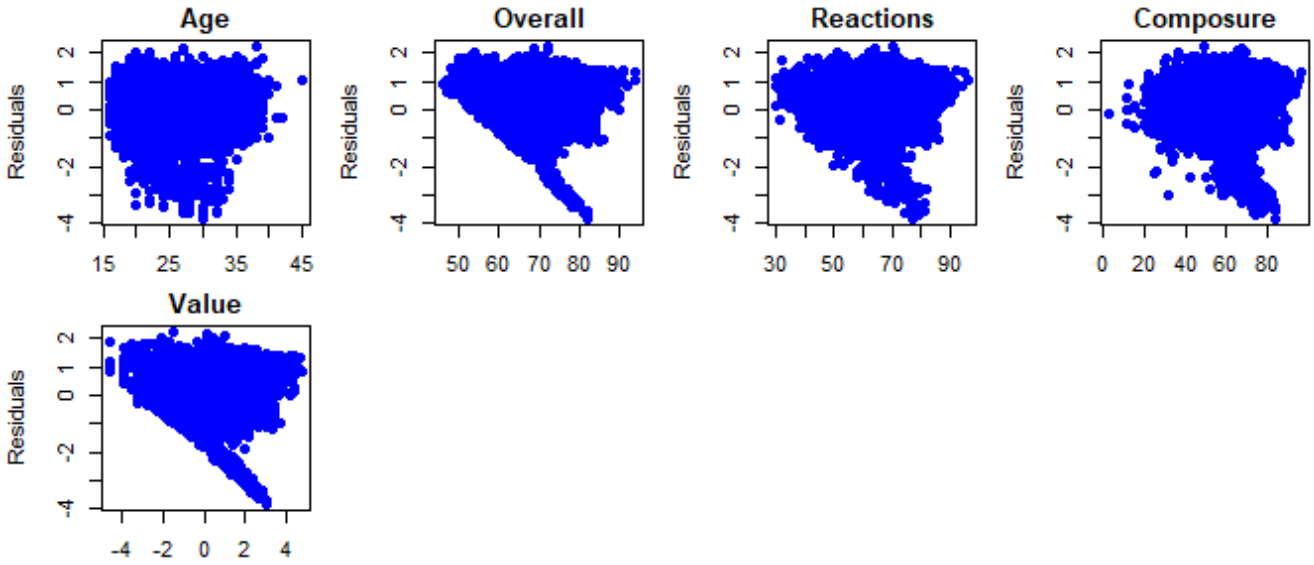
In the comparative table above, we see that the interactions we are testing does not improve a lot our criteria of goodness of fit : it is possible to get an AIC and an adjusted R-square slightly better than in the basis model (that's the case when we consider the interaction between *NationalityEurope* and *Potential*) but this difference isn't so much considerable. Adjusted R-squared only reaches `0.682` at best instead of `0.68` in the model without interactions and R-squared remains mainly constant for the different tested models. Therefore we will work with the model without any interaction.

Underlying Hypotheses Testing

We will now check for nonlinearity, outliers and influential observations, multicollinearity, heteroskedasticity, autocorrelation and normality of the residuals. If some of these hypotheses are not fulfilled, we will try to take remedial actions.

Nonlinearity

We would like to check whether it was a good choice to make a linear model regression. In particular, we would like to check if the regression function is linear. In order to do that, we will look at scatter plots of the residuals e_i against X_{ij}, with $j = 1, \ldots, p-1$.



We are looking for non-linear patterns on the above plots of the residuals and, if necessary, take remedial actions to cope with that problem. For example, if we see a quadratic scatterplot, we would be tempted to add a quadratic term for this explaining variable. In our scatterplots, we don't see evidence of hidden quadratic relations, although the scatterplots for the *Overall* and *Value* variables tend to show that there is probably a non-linear relation between those terms and our variable of interest. We choose to keep going with the initial model, because the non-linear formula seems to be not straightforward, and therefore we fear to complicate too much the model if we try to deal with it. In addition, recall that we have already taken the log of the *Wage* variable, so we have not so much possibilities remaining.

Outliers

We will first check for outliers with respect to X. For that, we calculate the leverages that we can find with the following formula

$$h_{ii} = (1, X_{i1}, \ldots, X_{i,p-1})(X'X)^{-1}(1, X_{i1}, \ldots, X_{i,p-1})'$$

In fact, they are the elements on the diagonal of the projection matrix $H = X(X'X)^{-1}X'$. Then, we say that the observation X_i is an outlier if $h_{ii} > \frac{2p}{n}$.

Using the criteria described previously, we obtain `548` outliers with respect to X (about `3.8%` of the players). Then, we check for outliers with respect to Y (the wage). We say that Y_i is an outlier if

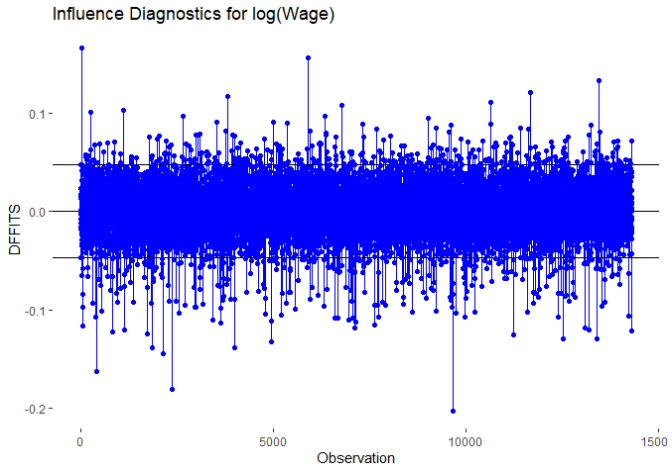
$$|d_i^*| > t_{n-p-1;1-\alpha/2}$$

where

$$d_i^* = e_i \left(\frac{n-p-1}{SSE(1-h_{ii}) - e_i^2} \right)^{1/2}$$

Again, our decision rule allows us to measure the number of outliers with respect to Y. We have `530` outliers of that kind out of `14326` players, which is quite low again (about `3.7%`). Since $t_{14309;0.975} \approx 1.96$, under our criteria, all the outliers with respect to Y are represented in red on the plot in appendix.

We now look at the influential observations for the fitted values. Recall that the i-th observation is influential if $\left|DFFITS_i\right| > 2\sqrt{\frac{p}{n}}$, where $DFFITS_i = d_i^* \sqrt{\frac{h_{ii}}{1-h_{ii}}}$.



We only have **513** influential observations for the fitted values : that's not much again (less than 4% of the players). Since $2\sqrt{\frac{p}{n}} \approx 0.07$ in our case, under our criteria, all the influential observations are the observations that exceed the thresholds on the previous plot.

In addition to that kind of influential observations, we have to consider the influential observations for the regression coefficients.

The decision rule seen in class asserts that the i-th observation is influential if :

$$\left|DFBETAS_{k,i}\right| > \frac{2}{\sqrt{n}}, \text{where}$$

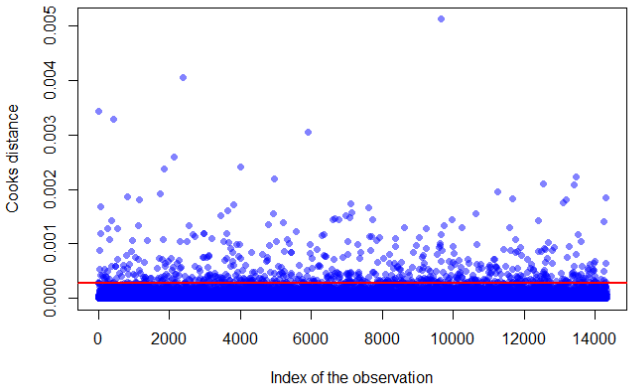
$$DFBETAS_{k,i} = \frac{\hat{\beta}_k - \hat{\beta}_{k(i)}}{\sqrt{MSE_{(i)} c_k}} \text{ and } c_k = (X'X)^{-1}_{kk}$$

In our case, $\frac{2}{\sqrt{n}} \approx 0.02$. Thus, every plot of $DFBETAS$ in appendix corresponds to the coefficient associated with the concerned variable. Then we see that there aren't much influential points for the regression coefficient associated with the *Value* variable whereas the regression coefficient associated with the *Preferred_FootLeft* variable has a lot of influential observations.

To have an overview of all the influential points, we look at the Cook's distance. Recall that Cook's distance is defined as

$$D_i = \frac{(\hat{\beta} - \hat{\beta}_{(i)})' X' X (\hat{\beta} - \hat{\beta}_{(i)})}{pMSE} = \frac{(\hat{Y} - \hat{Y}_{(i)})'(\hat{Y} - \hat{Y}_{(i)})}{pMSE}$$

Note that D_i measures the influence of the i-th observation on the coefficients and on the fitted values.



Here, the Cook's distances are very low (very close to 0 for the wide majority). We choose as decision rule that the influential points are the ones for which the Cook's distance D_i is bigger than $\frac{4}{n}$ (threshold represented in red on the previous plot) where n is the number of observations (common criteria).

Multicollinearity

We will use the Variance Inflation Factors (VIF) to determine if there is multicollinearity problems. Recall that the VIF for $\hat{\beta}_k$ is defined by

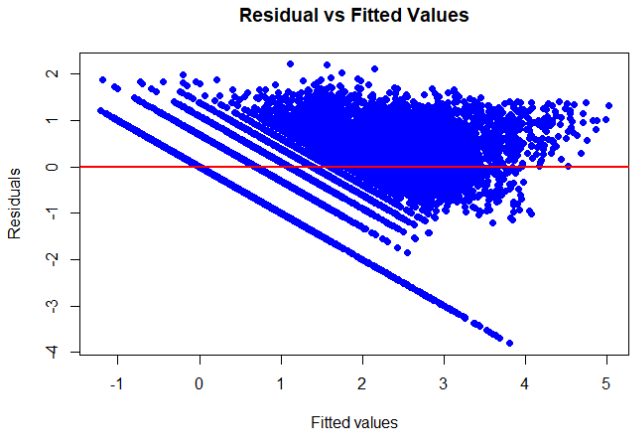
$$VIF_k = \frac{1}{1-R_k^2} \text{ for } k = 1,\dots,p-1 \text{ where } R_k^2$$

is the coefficient of determination of a regression of X_k on $X_1,\dots,X_{k-1},X_{k+1},\dots,X_{p-1}$. The decision rule to decide if there is a multicollinearity problem is the following : if either the maximum VIF is higher than `10` or if the average VIF is considerably larger than `1` (A VIF_k greater than `10` means that the corresponding R_k^2 is larger than `0.9`, which is a sign of approximate collinearity). This method, instead of the pairwise correlation matrix that we analyzed before, has the advantage to detect also multicollinearity relations involving more than `2` variables.

Since we have VIF of `36.922` (for *Overall*) and `30.072` (for *Value*), based on our decision rule, there is a multicollinearity problem. Moreover, the mean of all these VIF is of `11.429`, which is also considerably larger than than one. Nevertheless, we can not apply Ridge Regression since the underlying assumptions are not satisfied. Indeed, as shown before, the linearity criterion isn't that clear. Moreover, we can show that there is heteroskedasticity (see next section). As a result, we have to cope with that multicollinearity problem in the following.

Heteroskedasticity

Based on the scatter plots of the residuals e_i against X_{ij}, with $j = 1,\dots,p-1$ in the study of nonlinearity, we can strongly suspect that there is heteroskedasticity. Let's have a look at the scatter plot of the residuals against the fitted values. If the variance of the residuals is considerably variable, we discard the hypothesis of homoskedasticity, meaning that there is heteroskedasticity.



It seems that the more the fitted values are big, the more our residuals tends to be negative. It is a bit complicated to decide only on basis of this graph whether there is heteroskedasticity, so we decide to do a Breusch-Pagan test. Note that the null hypothesis of this test states the homoskedasticity whereas in the alternative hypothesis the variance of the residuals is dependent on the observation (i.e. player) considered (equivalent to heteroskedasticity).

```
##
##  Breusch Pagan Test for Heteroskedasticity
##  -----------------------------------------
##  Ho: the variance is constant
##  Ha: the variance is not constant
##
##                                    Data
##  ------------------------------------------------------------------------------
##  Response : log(Wage)
##  Variables: Age Overall Reactions Composure Value NationalityAmerica NationalityAsia
##
##             Test Summary (Unadjusted p values)
##  ---------------------------------------------------------
##   Variable                 chi2       df          p
##  ---------------------------------------------------------
##   Age                   37.671208     1     8.373135e-10
##   Overall              651.094698     1    1.291327e-143
##   Reactions            497.374568     1    3.541720e-110
##   Composure            425.108726     1     1.885765e-94
##   Value                734.256508     1    1.063290e-161
##   NationalityAmerica     5.716542     1     1.680580e-02
##   NationalityAsia       37.184744     1     1.074511e-09
##  ---------------------------------------------------------
##   simultaneous         775.657526     7    3.318437e-163
##  ---------------------------------------------------------
```

With a p-value of `3.318e-163`, it is perfectly clear that we can reject the null hypothesis of homoskedasticity. Hopefully, we have done a robust inference so our $\hat{\beta}$ estimator is consistent. As a result, we don't really have to consider remedial actions such as the method of weighted least squares and Box-Cox transformation.

Autocorrelation

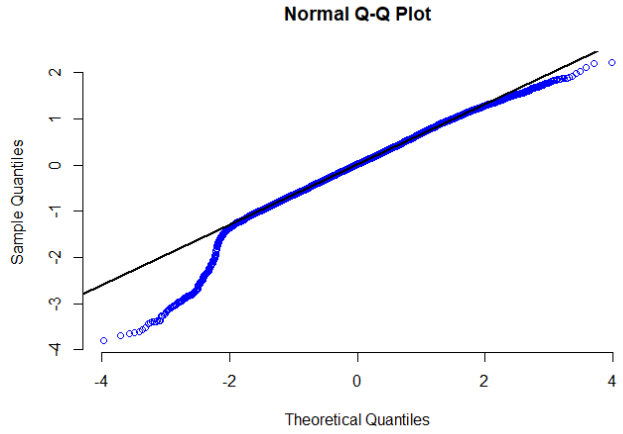
Autocorrelation is discussed through the Breusch-Godfrey test. Recall that this test looks at non-autocorrelation (H_0) versus autocorrelation (H_1). The null hypothesis can be written as $corr(\varepsilon_t, \varepsilon_{t-k}) = 0$ for $k = 1, \ldots, p$.

```
##
##  Breusch-Godfrey test for serial correlation of order up to 8
##
## data:  model
## LM test = 4.0562, df = 8, p-value = 0.852
```

The p-value of this Breush-Godfrey test tells us that we can not reject the null hypothesis of no autocorrelation since `0.852 > 0.05`. Therefore, we don't take remedial actions in this case.

Normality of the residuals

In order to test the normality of the residuals, we can first consider a Quantile-Quantile plot of the residuals. In this plot, we compare the empirical quantiles of the residuals with those of a normal distribution.



Therefore, the points on this graph should follow the straight line. We see that this is not the case for the first quantiles, even though it seems to get better and better then. We will do a Jarque-Bera test to take the decision of rejecting normality of the residuals or not. This test compares the coefficients of skewness S and kurtosis κ with the theoretical values for a normal distribution. Recall that for a normal distribution, skewness is equal to 0 and kurtosis is equal to 3 (this is our null hypothesis H_0). The Jarque-Bera test statistic is then given by

$$JB = \frac{n}{6}\left[\hat{S^2} + \frac{(\hat{\kappa} - 3)^2}{4}\right] \sim \chi_2^2$$

under H_0.

```
##
##  Jarque-Bera Normality Test
##
## data:  model$residuals
## JB = 2880.8, p-value < 2.2e-16
## alternative hypothesis: greater
```

With the result of the test above, we reject the normality of the error terms. This could be the result of an inadequate model, in fact this could mean that the error is not random. Like we have already said in the nonlinearity section, a complicated nonlinearity relation could maybe fix this problem, but it would be out of the scope of this project.

Significance of the Coefficients

We show below the estimates and significance test for the variables that are included in our model.

```
##
## Call:
## lm(formula = formula.lassology, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8093 -0.4251  0.0177  0.4563  2.2133
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -1.0040132  0.2830001  -3.548  0.00039 ***
## Age                 0.0288180  0.0026975  10.683  < 2e-16 ***
## Overall             0.0220176  0.0051001   4.317 1.59e-05 ***
## Reactions           0.0037351  0.0012782   2.922  0.00348 **
## Composure           0.0014380  0.0007785   1.847  0.06475 .
## Value               0.5698624  0.0226438  25.166  < 2e-16 ***
## NationalityAmerica -0.1149382  0.0143658  -8.001 1.33e-15 ***
## NationalityAsia     0.1533940  0.0212636   7.214 5.71e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7004 on 14318 degrees of freedom
```

```
## Multiple R-squared:  0.6797, Adjusted R-squared:  0.6796
## F-statistic:  4342 on 7 and 14318 DF,  p-value: < 2.2e-16
```

All our coefficients are significant at an α−level of 0.01. We note that all the coefficients estimates β_j have a positive sign, apart from *NationalityAmerica* and the intercept. In fact, beeing American seems to increase your salary by 890$. Since we are working with the log of the *Wage*, even a negative coefficient increases the salary. Being Asian, moreover, increases your wage by 1700$. An explantion for that could be that rising clubs in country like Qatar or Saudi Arabia recruits manly asian players, and pay them well. The explanation for the coefficient of *NationalityAmerica* could be that South American players are really praised in top Europeean football clubbs.

Coefficient Linear Combination

We will now test if a linear combination of our coeficients holds. We are interested in the effect of the *Overall* variable. The objective will be to see if its effect is compensated by the effect of the other coefficients relative to player mindset characteristics. We will therefore test the following hypothesis :

$$H_0 : \beta_{Overall} = \frac{\beta_{Reactions} + \beta_{Composure}}{2}$$

```
## Linear hypothesis test
##
## Hypothesis:
## - 2 Overall  + Reactions  + Composure = 0
##
## Model 1: restricted model
## Model 2: log(Wage) ~ Age + Overall + Reactions + Composure + Value + NationalityAmerica +
##     NationalityAsia
##
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1  14319 7029.8
## 2  14318 7023.1  1    6.7256 13.711 0.0002139 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We see in the table above that we can reject the restricted model (i.e. the model under H_0). We can conclude that the effect of the *Overall* variable is not equal to the mean of the effect of the players mindset characteristics.

Test of nullity of a subset of coefficients

We would like to see if we can simplify our model a bit to a restricted model. For that, we will check if we can say that some of the β_j (actually, the ones of *Reaction* and *Composure*) are simultaneously equal to 0. We want to test them because we suspect that their role is not crucial to determine the salary of the player. Our test statistic is

$$\frac{(SSE_0 - SSE)/q}{SSE/(n-p)} \sim F_{q,n-p}$$

The idea behind this test is that under H_0, SSE and SSE_0 should be close. We will therefore reject H_0 if the statistic is beyond a critical value (for a $F_{q,n-p}$ distribution).

```
## Linear hypothesis test
##
## Hypothesis:
## Reactions = 0
## Composure = 0
##
## Model 1: restricted model
## Model 2: log(Wage) ~ Age + Overall + Reactions + Composure + Value + NationalityAmerica +
##     NationalityAsia
##
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
```

```
## 1  14320 7030.0
## 2  14318 7023.1  2     6.951 7.0856 0.0008401 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In the result table above, we look at the following null hypothesis :

$$H_0 : \beta_{Reactions} = \beta_{Composure} = 0$$

We see that we have a p-value of `0.0008` for this F-test. Therefore, at our nominal level, we can reject H_0. So we reject the assertion that $\beta_{Reactions}$ and $\beta_{Composure}$ are null simultaneously. It seems that the behaviour of the player has then an impact on its wage. Maybe, club managers want players to be exemplary to give a good impression of the club.

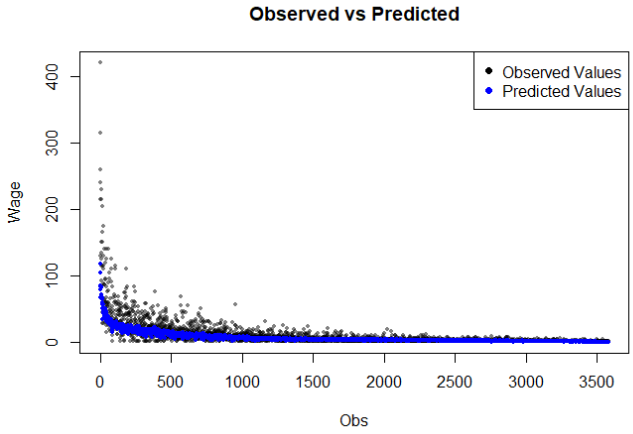
Predictions

We are now interested in calculating confidence intervals for the observations that we left apart in the beginning. Thus, now we are not working with our training dataset anymore (80% of the initial data), but with our validation dataset (the 20% left).

We know that a prediction interval at level $(1-\alpha)100\%$ for a new vector observation x_h is

$$\hat{Y}_h \pm t_{n-p;1-\alpha/2} S(1 + x_h'(X'X)^{-1}x_h)^{1/2}$$

It is wider than the confidence region for $E[Y_h]$ because for Y_h we also have the incertitude of the error ϵ.



we calculate that only `3.32%` of our prediction intervals contains the observed values, which is very poor. However, we can see above the actual values of the *Wage* variable of the validation dataset, next to the predicted ones. We see a certain fit with the observed data, which is good.

Conclusion

Our main goal was to make football managers know the wage they have to pay their players, in order to be around the average of what other clubs do. That is a useful information because it could be a precious help to the club accounting. We started by selecting the variables that would compose our model. We did that using several methods of selection (AIC-stepwise, LASSO). We also checked if we had better goodness of fit criteria by taking the log of the response variable, and that was the case. We selected a model that was not too complicated but that had still a good fit to the training data. It was the LASSO model. We were not surprised to see that the value of the player was significant. In fact, the value of the player is one of the variables that help the most to predict his wage. Unsurprisingly, some game characteristics play a role too.

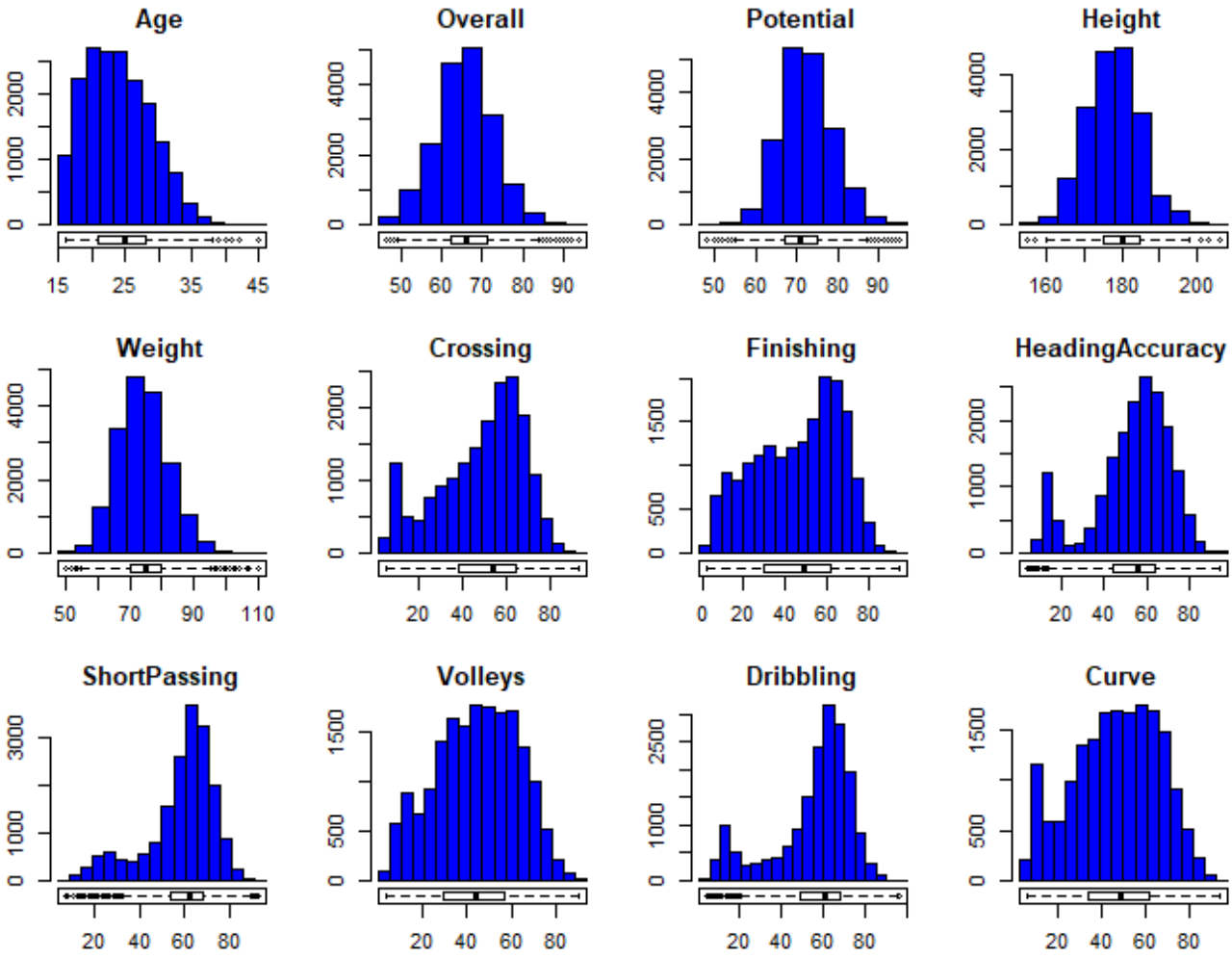
We then tested some hypotheses on our selected model. We saw that we have approximate multicollinearity, nonlinearities, heteroskedasticity and non normality of the error term. We tried to take remedial actions against that but we think that the nonlinear relation in the true model is very complicated, since we did not manage to improve the model with simple analytic transformations of our variables. We saw that we have some outliers and influential observations, but not so much in comparison to the size of the dataset. Of course we did not delete them because this phenomenon might hide the lack of an important explanatory variable, such as player popularity, for example. We tried then some linear hypotheses, but we had to reject all of them.

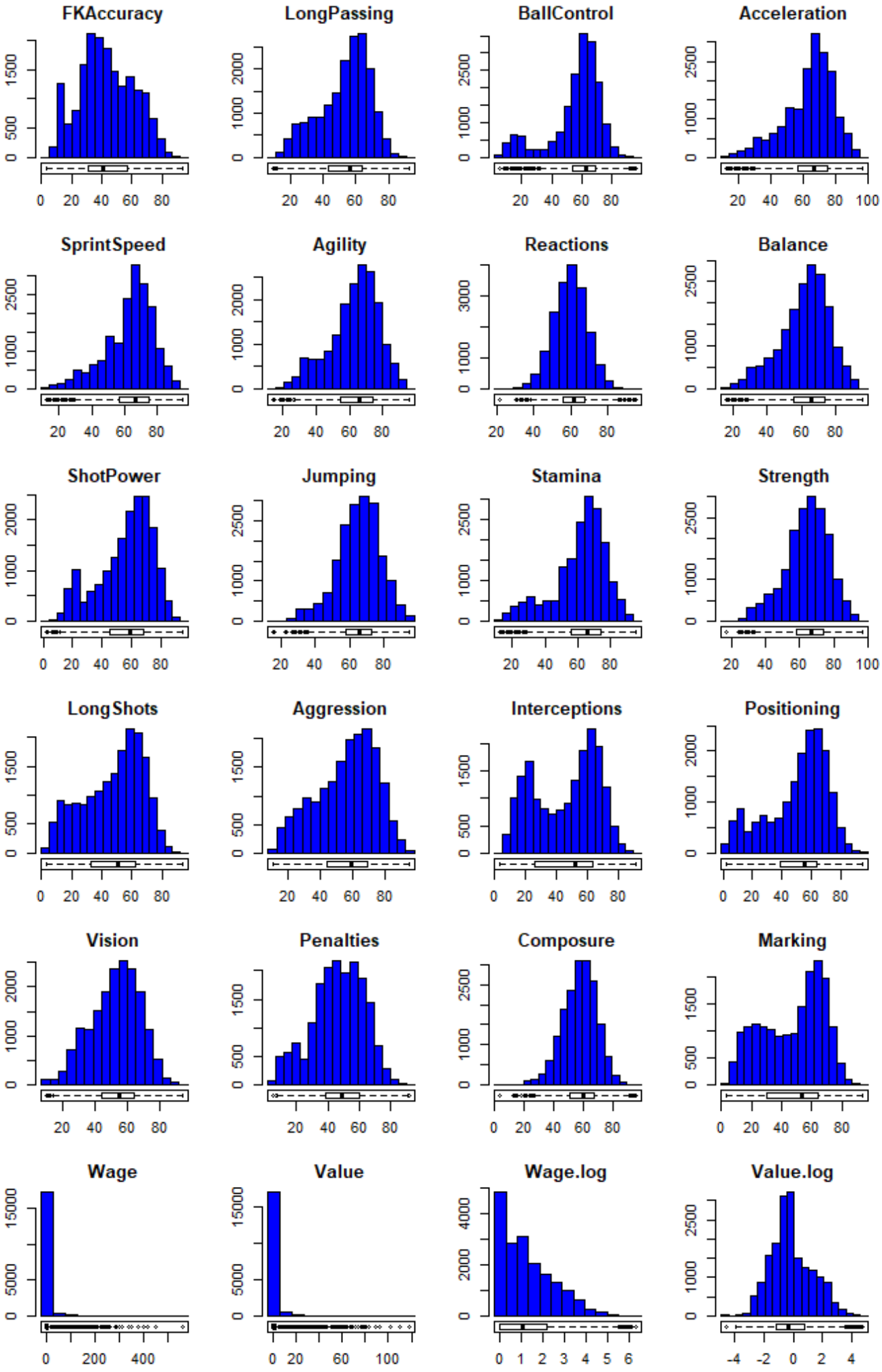
Finally, and maybe the most important part of the project, the prediction. Unfortunately, we have a severe drawback of our model for this section, our model has very bad predictive power. Our prediction intervals only cover the true value observed in the validation dataset only in 3.3% of the cases...

As a conclusion, we could say that this project was very interesting because it was a dive into a realistic practical problem. Furthermore, the fail to predict correctly our response variables with linear models pushes us to learn more about other regression techniques, and therefore diversify our statistics advanced techniques.

Appendix

Histograms and boxplots




FKAccuracy
LongPassing
BallControl
Acceleration
SprintSpeed
Agility
Reactions
Balance
ShotPower
Jumping
Stamina
Strength
LongShots
Aggression
Interceptions
Positioning
Vision
Penalties
Composure
Marking
Wage
Value
Wage.log
Value.log

Model selection

Stepwise

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## log(Wage) ~ Age + Overall + Potential + Height + Weight + HeadingAccuracy +
##     Reactions + Jumping + Strength + Aggression + Interceptions +
##     Composure + Marking + Value + NationalityAfrica + NationalityAmerica +
##     NationalityAsia + NationalityEurope + NationalityOceania +
##     Preferred_FootLeft + Preferred_FootRight
##
## Final Model:
## log(Wage) ~ Age + Potential + Height + HeadingAccuracy + Reactions +
##     Strength + Aggression + Marking + Value + NationalityAfrica +
##     NationalityAsia + NationalityEurope + Preferred_FootLeft
##
##
##                    Step Df     Deviance Resid. Df Resid. Dev       AIC
## 1                                          14306   6933.838 -10355.84
## 2 - Preferred_FootRight  0 0.0000000000     14306   6933.838 -10355.84
## 3  - NationalityOceania  0 0.0000000000     14306   6933.838 -10355.84
## 4             - Jumping  1 0.0008423988     14307   6933.839 -10357.84
## 5             - Overall  1 0.1041677658     14308   6933.943 -10359.63
## 6  - NationalityAmerica  1 0.1707073946     14309   6934.114 -10361.27
## 7           - Composure  1 0.6117907978     14310   6934.725 -10362.01
## 8             - Weight  1 0.8849700016     14311   6935.610 -10362.18
## 9       - Interceptions  1 0.9491383014     14312   6936.560 -10362.22
```

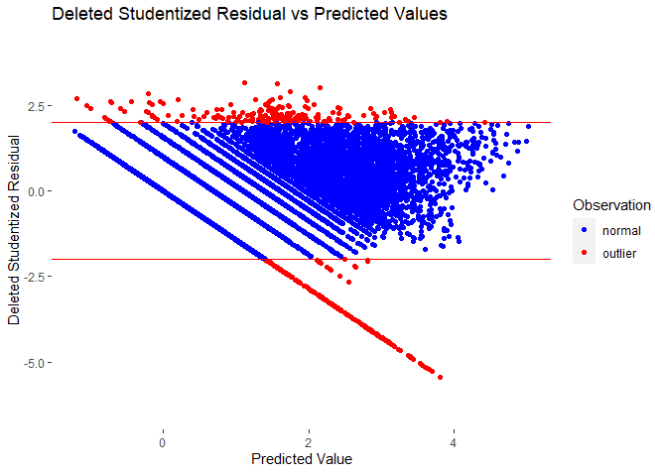
LASSO

```
##
## Call:
## lm(formula = formula.lassology, data = train)
##
## Coefficients:
##        (Intercept)                 Age             Overall           Reactions
##          -1.004013            0.028818            0.022018            0.003735
##          Composure               Value  NationalityAmerica     NationalityAsia
##           0.001438            0.569862           -0.114938            0.153394
```

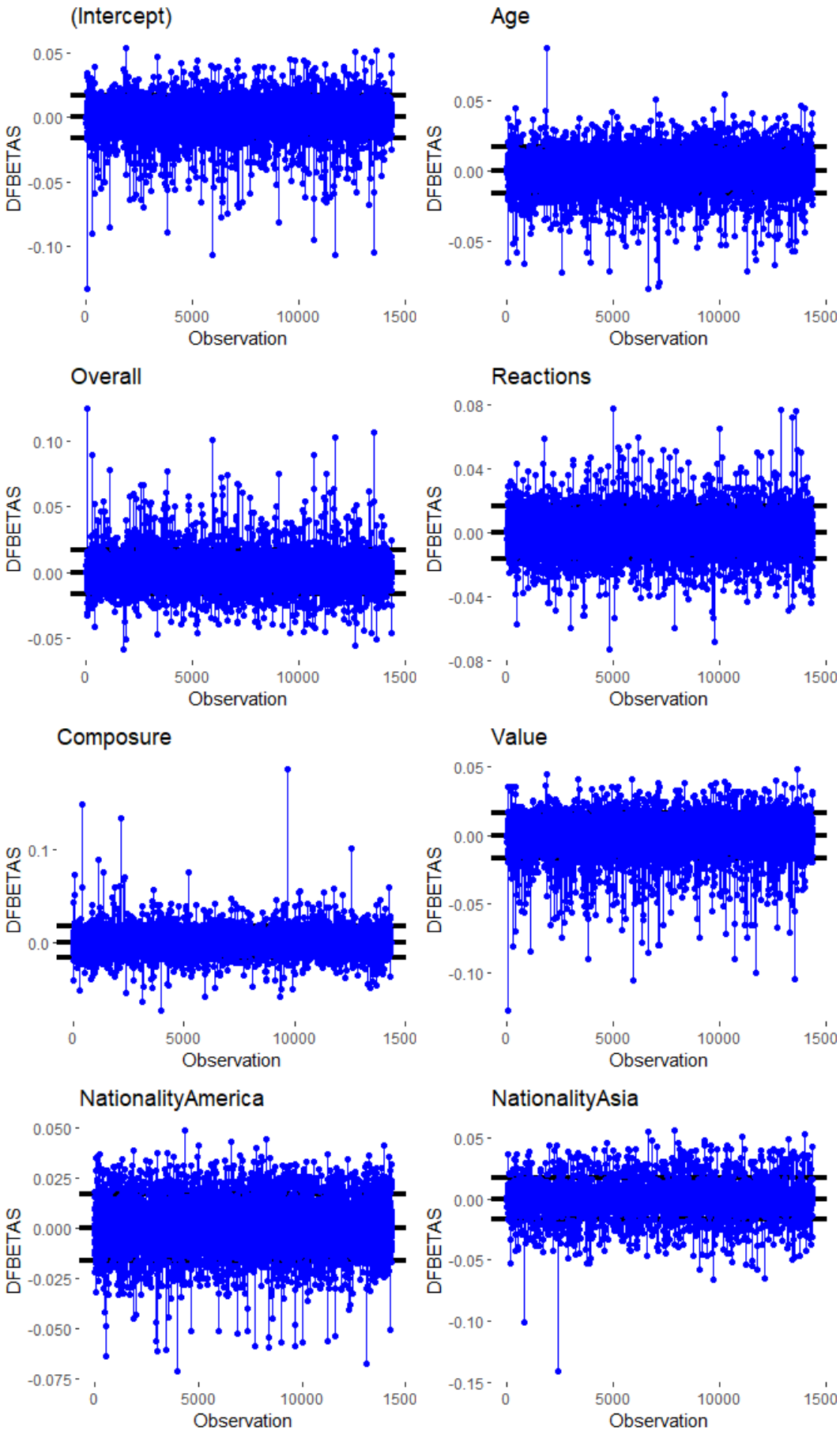
```
##
## Call:   glmnet(x = data.matrix(within(train, {        Wage = NULL })), y = log(train$Wage), family =
"gaussian", alpha = 1)
##
##    Df  %Dev  Lambda
## 1   0  0.00 1.00000
## 2   2 11.35 0.91110
## 3   2 20.85 0.83020
## 4   2 28.73 0.75640
## 5   2 35.28 0.68920
## 6   2 40.71 0.62800
## 7   2 45.22 0.57220
## 8   2 48.97 0.52140
## 9   2 52.08 0.47510
## 10  2 54.66 0.43290
## 11  2 56.80 0.39440
## 12  2 58.58 0.35940
## 13  2 60.05 0.32740
## 14  2 61.28 0.29840
## 15  2 62.30 0.27190
## 16  2 63.14 0.24770
## 17  2 63.84 0.22570
## 18  2 64.43 0.20560
## 19  2 64.91 0.18740
## 20  2 65.31 0.17070
## 21  2 65.64 0.15560
## 22  2 65.92 0.14170
## 23  2 66.15 0.12920
## 24  2 66.34 0.11770
## 25  3 66.51 0.10720
## 26  3 66.65 0.09770
## 27  3 66.77 0.08902
## 28  3 66.87 0.08111
## 29  3 66.95 0.07391
## 30  3 67.02 0.06734
## 31  4 67.08 0.06136
## 32  4 67.13 0.05591
## 33  4 67.17 0.05094
```

```
## 34  7 67.23 0.04641
## 35  7 67.36 0.04229
## 36  7 67.46 0.03853
## 37  7 67.55 0.03511
## 38  8 67.62 0.03199
## 39  9 67.69 0.02915
## 40  9 67.78 0.02656
## 41  9 67.85 0.02420
## 42  9 67.92 0.02205
## 43 10 67.97 0.02009
## 44 10 68.02 0.01831
## 45 11 68.06 0.01668
## 46 11 68.10 0.01520
## 47 11 68.13 0.01385
## 48 11 68.15 0.01262
## 49 11 68.17 0.01150
## 50 11 68.19 0.01048
## 51 12 68.21 0.00954
## 52 14 68.23 0.00870
## 53 16 68.25 0.00792
## 54 16 68.27 0.00722
## 55 16 68.28 0.00658
## 56 17 68.30 0.00600
## 57 17 68.31 0.00546
## 58 17 68.32 0.00498
## 59 18 68.33 0.00454
## 60 18 68.34 0.00413
## 61 18 68.34 0.00376
## 62 18 68.35 0.00343
## 63 18 68.35 0.00313
## 64 18 68.36 0.00285
## 65 18 68.36 0.00260
## 66 18 68.36 0.00236
## 67 20 68.37 0.00215
## 68 20 68.37 0.00196
## 69 20 68.37 0.00179
## 70 20 68.37 0.00163
## 71 20 68.37 0.00148
## 72 20 68.37 0.00135
## 73 20 68.38 0.00123
## 74 20 68.38 0.00112
## 75 20 68.38 0.00102
## 76 20 68.38 0.00093
```

Outliers



Influence diagnostics



Code

```
library(ggplot2)

# ===
```

```r
# Setup and helper functions


opt.digits = 3
opt.barcolor = "blue"

models.comparison = data.frame()

addComparison = function(model, model_name){
  to_combine = data.frame(
    Model = as.character(model_name),
    Vars = length(names(coef(model)))-1,
    LogLik = ifelse(is.null(logLik(model)[1]), yes=NA, no=round(logLik(model)[1], opt.digits)),
    AIC = round(AIC(model), opt.digits),
    R2 = round(summary(model)$r.squared, opt.digits),
    Adjusted.R2 = round(summary(model)$adj.r.squared, opt.digits),
    Deviance = ifelse(is.null(deviance(model)), yes=NA, no=round(deviance(model), opt.digits))
  )
  rbind(
    models.comparison,
    to_combine
  )
}

# Remove text and change color of ggplot
changePlot = function(p){
  p$layers[[5]] = NULL
  p$layers[[4]] = NULL
  p$layers[[3]]$aes_params$colour = opt.barcolor
  p$layers[[1]]$aes_params$colour = opt.barcolor
  p$layers[[2]]$aes_params$colour = "black"
  p$layers[[3]]$aes_params$shape = 16
  return (p)
}

# Remove background and grid of ggplot
removeBackground = function(p){
   p$theme = list(panel.grid.major = element_blank(), panel.grid.minor = element_blank(), panel.background =
element_blank())
  return (p)
}

# Descriptive analysis : Mean, Sd, Skewness and Kurtosis
library(moments)
library(formattable)
compute.desc.quantitative = function(data){

  addStats = function(col, name){

    return(data.frame(
      Variable=name,
      Mean = round(mean(col, na.rm = T), opt.digits),
      Std.Deviation = round(sd(col), opt.digits),
      Skewness = round(skewness(col), opt.digits),
      Kurtosis = round(kurtosis(col), opt.digits)
    ))
  }

  for (c in colnames(data)){
    if (exists("results")){
      results = rbind(results, addStats(data[[c]], c))
    }else{
      results = addStats(data[[c]], c)
    }
  }

  results.order = order(as.character(results$Variable))
  results = results[results.order,]
  row.names(results) = NULL
  return (results)

}

# Histograms with boxplots below
draw.hist.boxplots = function(data, nrow=2, ncol=2, boxframe=T){


  layout(mat=matrix(seq(nrow*ncol*2), 2*nrow, 1*ncol, byrow=F), heights = rep(c(4,1.5), ncol))
  drawPlots = function(coname){

    par(mar=c(0, 2, 2, 2))
    hist(data[[coname]], axes=T, main=coname, col = opt.barcolor, xlab="", ylab="Count", xaxt="n")
    par(mar=c(3, 2.3, 0, 2.3))
    boxplot(data[[coname]], horizontal = T, frame=boxframe)

  }

  invisible(sapply(colnames(data), function (x) drawPlots(x)))

}
```

```r
# ===
# Data importation

set.seed(28122020)
original = read.csv("./sources/foot_last.csv", sep=";", encoding = "utf8", skipNul = T, stringsAsFactors = F)
original = na.omit(original)
original = original[-which(original == 0, arr.ind = T)[,1],] # remove rows where value is 0

# Dummies variables
dummies = data.frame(model.matrix(~ Nationality - 1, data=original),
                     model.matrix(~ Preferred_Foot - 1, data=original))

# Transformations and cleaning
dataset = within(original, {
  Weight  = round(as.numeric(Weight*0.453592))
  Height = round(sapply(Height, function(x)
    (as.numeric(strsplit(x, "'")[[1]][1])*12 + as.numeric(strsplit(x, "'")[[1]][2]))*2.54))

  Value = Value_M #* 10^6
  Wage = Wage_K #* 1000

  Value_M = NULL
  Wage_K = NULL
  Nationality = NULL
  Preferred_Foot = NULL
})

# ===

# Histogram of qualitative variable
par(mfrow=c(1,2))
barplot(prop.table(table(original$Preferred_Foot)), main="Preferred foot of the player", col = opt.barcolor,
ylab="Percentage")
barplot(prop.table(table(original$Nationality)),  main="Nationality  of  the  player",  col  =  opt.barcolor,
ylab="Percentage",las=2)


# Print desc
desc.quantitative = compute.desc.quantitative(dataset)
formattable(desc.quantitative, align=c("l", rep("c", ncol(desc.quantitative)-1)))


# Boxplot and Histogram
draw.hist.boxplots(
              data.frame(Wage=dataset$Wage,       Value=dataset$Value,              Wage.log=log(dataset$Wage),
Value.log=log(dataset$Value)),
  1, 4, TRUE
)

# ===
# Correlation Matrix

library(corrplot)
corrplot(cor(dataset), type="upper", order="hclust", tl.col="black", tl.srt=90, tl.cex = 0.7)

# Removing of "similar" variables
clean_dataset = dataset[,c(1:5, 8, 19, 22, 24, 26:27, 31, 32:34)]
clean_dataset$Value = log(clean_dataset$Value)

# New correlation matrix
corrplot(cor(clean_dataset), type="upper", order="hclust", tl.col="black", tl.srt=90, tl.cex = 0.7)

library(olsrr)
library(lmtest)

library(MASS)
library(glmnet)

# ===
# Split into training and validation sets

tmp = sample(nrow(clean_dataset), round(.80*nrow(clean_dataset)), replace=F)

train = cbind(clean_dataset[tmp,], dummies[tmp,])
valid = cbind(clean_dataset[-tmp,], dummies[-tmp,])

rm(tmp)

# ===
# Model selection

# Complete Model
```

```r
formula.complete.logy = as.formula("log(Wage) ~ .")
model.complete.logy = lm(formula.complete.logy, data=train)

models.comparison = addComparison(model.complete.logy, "Complete log(Y)")


# Stepwise AIC log(Y)
model.aiclogy = stepAIC(model.complete.logy, direction = "both", trace = F, k = 2)

models.comparison = addComparison(model.aiclogy, "Stepwise AIC log(Y)")
aicdiff = setdiff(names(train), names(model.aiclogy$coefficients))

# LASSO log(Y)
lasso.cvlogy = cv.glmnet(x=data.matrix(within(train, {Wage = NULL})), y = log(train$Wage), family =
"gaussian", alpha = 1)
lasso.coeflogy = coef(lasso.cvlogy, 0.04641) # Best lambda in our opinion

formula.lassology = as.formula(paste0("log(Wage) ~ ", paste(rownames(lasso.coeflogy)[which(abs(lasso.coeflogy)
> 0)][0:-1], collapse=" + ")))

model.lassology = lm(formula.lassology, data=train)
models.comparison = addComparison(model.lassology, "Lasso log(Y)")


# ===
# Chosen model

choice = 3
model = model.lassology

formattable(models.comparison, list(
  formattable::area(row=choice) ~ formatter("span", style= ~ style(font.weight="bold"))
))

formula.model = paste0("log(Wage) ~ ", paste(names(coef(model))[0:-1], collapse=" + "))

# ===
# Variables interactions

model.interaction1 = lm(paste0(formula.model, " + Preferred_FootLeft * Marking"), data = train)
model.interaction2 = lm(paste0(formula.model, " + Preferred_FootLeft * Value"), data = train)
model.interaction3 = lm(paste0(formula.model, " + NationalityEurope * Potential"), data = train)
model.interaction4 = lm(paste0(formula.model, " + NationalityAfrica * Value"), data = train)

# Reset the model comparison df
models.comparison = data.frame()
models.comparison = addComparison(model, "Final Model")
models.comparison = addComparison(model.interaction1, "Model 1")
models.comparison = addComparison(model.interaction2, "Model 2")
models.comparison = addComparison(model.interaction3, "Model 3")
models.comparison = addComparison(model.interaction4, "Model 4")

formattable(models.comparison, align=c("l", "c"))

# ===
# Non-linearity relations

par(mfrow=c(2,4))
for(i in 2:6){
  par(mar=c(2,4,2,1))
    plot(x = unlist(model$model[i]), y=model$residuals, main = names(coef(model))[i], ylab = "Residuals",
pch=16, col=opt.barcolor)
}

# Outliers with respect to X

explanatory = train[, names(coef(model))[0:-1]]
pn = (1+ncol(explanatory))*2/nrow(explanatory)

# Matrix computation
mafull = as.matrix(explanatory)
macross = crossprod(mafull, mafull)
mainv = solve(macross)

# Use sapply for speed improvement
hii = sapply(seq(1, nrow(explanatory)),
  function(i){
    ma = as.matrix(explanatory[i,])
    return(ma %*% tcrossprod(mainv, ma))
  }
)

outliers_X = length(which(hii > pn))
```

```r
# Outliers with respect to Y

rStudentModel = as.matrix(rstudent(model))
quantileStudent = qt(0.975, nrow(explanatory)-ncol(explanatory)-2)

outliers_Y = sum(sapply(seq(1, nrow(explanatory)),
  function(i){
    return(abs(rStudentModel[i]) > quantileStudent)
  }
))


# Influential Obsevations

DFFITS = sapply(seq(1, nrow(explanatory)),
  function(i){
    return(rStudentModel[i] * sqrt(hii[i]/(1-hii[i])))
  }
)

influential_fitted = length(which(abs(DFFITS) > 2*sqrt((1+ncol(explanatory))/nrow(explanatory))))

# Plot influential
p = ols_plot_dffits(model, print_plot = F)
removeBackground(changePlot(p$plot))

plot(cooks.distance(model), xlab="Index of the observation", ylab="Cooks distance", pch=16,
     col = rgb(0,0,255, alpha=125, maxColorValue = 255))
abline(h=4/14326, col="red", lwd=2)

# ===
# Multicollinearity

p = ols_vif_tol(model)
p = within(p, {
  Tolerance = round(Tolerance, opt.digits)
  VIF = round(VIF, opt.digits)
})
formattable(p, align=c("l", "c", "c"))

# ===
# Heteroskedasticity : Residuals vs fitted
plot(model$fitted.values, model$residuals, xlab="Fitted values", ylab="Residuals", pch=16,
     col=opt.barcolor, main = "Residual vs Fitted Values")
abline(h=0, col=2, lwd=2)

# Breusch-Pagan test
t = ols_test_breusch_pagan(model, rhs = T, multiple=T)

# ===
# Autocorrelation : Breusch-Godfrey test
t = bgtest(model, order = 8)


# ===
# Normality of the error term

# QQplot
qqnorm(model$residuals, pch = 1, frame = FALSE, col = opt.barcolor)
qqline(model$residuals, lwd = 2)

# Jarque-Bera test
jarque.test(model$residuals)


# ===
# Coefficient Significance
summary(model)

# Linear Hypothesis
library(car)
hyp = c(0, 0, -2, 1, 1, 0, 0, 0)
t = linearHypothesis(model, hypothesis.matrix = hyp)

# Nullity of a small subset
t = linearHypothesis(model, c("Reactions=0", "Composure=0"))


# ===
# Predictions

predictions = exp(predict(model, newdata = valid, interval = "confidence"))
plot(1:nrow(valid), valid$Wage, pch=16, cex=0.5, xlab="Obs", ylab="Wage", main="Observed vs Predicted",
     col=rgb(0, 0, 0, alpha=125, max=255))
points(1:nrow(valid), predictions[,1], col=opt.barcolor, pch=16, cex=0.5)
```

```r
legend("topright", legend = c("Observed Values", "Predicted Values"), pch = c(16,16), col=c(1,4))

# How many obs are in the CI
aimed = length(which(valid$Wage >= predictions[,2] & valid$Wage <= predictions[,3]))
k=aimed / nrow(valid)*100



# ===
# Appendix

draw.hist.boxplots(data.frame(dataset,  Wage.log=log(dataset$Wage), Value.log=log(dataset$Value)), 1, 4, T)

model.aiclogy$anova

print(model.lassology)
lasso.cvlogy$glmnet.fit

t = ols_plot_resid_stud_fit(model, print_plot = F)
t$plot$layers[[3]] = NULL;t$plot$layers[[3]] = NULL;
t = removeBackground(t$plot)
plot(t)

library(cowplot)
p = ols_plot_dfbetas(model, print_plot = F)
p = lapply(p$plots, function(x) {
  x = removeBackground(changePlot(x))
  x$layers[[2]]$aes_params$size = 1.5
  x$labels$title = substr(x$labels$title, start=27, stop=100)
  return(x)
})
plot_grid(plotlist = p, ncol = 2)

# ===
# END
```