Simple linear regression

Prof. Marta Pérez-Casany and Ariel Duarte-López

Data Description

The dataset used along this analysis contains information about some of the featured players that appear in the video game FIFA 2018. Particularly, we use a subsample of the original data set which collects the information about the players of the three following teams: *Manchester City*, *Real Madrid CF* and *FC Barcelona*.

We have collected the following information:

- Name: *factor*. Containts the name of the player and it is a qualitative variable.
- Agility: *numeric*. Contains a measure that summarizes the player's agility. It is a numeric variable that takes values from 36 to 93.
- Acceleration: *numeric*. Contains a measure that summarizes the player's acceleration. It is a numeric variable that takes values from 32 to 94.
- Club: *factor*. Categorical variable with the name of the Clubs to which each player belongs.

Exploratory Analysis

```r
# Loading the whole data set and filtering the players by the Club.
# Additionally, we also filtered the attributes necessary for this analysis.

data <- read.csv('/home/ariel/Documents/classes/CompleteDataset.csv')
fc <- data[ data$Club %in% c('Manchester City','Real Madrid CF', 'FC Barcelona'),
           c('Name', 'Agility', 'Acceleration', 'Club') ]

#Show the first 6 instances of the data set.
head(fc)
```

```
##                  Name Agility Acceleration            Club
## 1  Cristiano Ronaldo      89           89  Real Madrid CF
## 2            L. Messi      90           92    FC Barcelona
## 4           L. Suárez      86           88    FC Barcelona
## 9            T. Kroos      71           60  Real Madrid CF
## 11      Sergio Ramos      79           75  Real Madrid CF
## 12      K. De Bruyne      80           76 Manchester City
```

```r
#Show the last 6 instances of the data set.
tail(fc)
```

```
##                      Name Agility Acceleration            Club
## 12989         D. Faupala      71           70 Manchester City
## 13568 C. Humphreys-Grant      63           68 Manchester City
## 15217         W. Patching      65           68 Manchester City
## 16599             M. Wood      54           63 Manchester City
## 17347     Erik Sarmiento      60           72 Manchester City
## 17962   J. Latibeaudiere      44           57 Manchester City
```

```r
#Show the column types inferred by R when the data is exported.
sapply(fc, class)
```

```
##         Name      Agility Acceleration         Club
##     "factor"     "factor"     "factor"     "factor"
```

In order to understand why R have assigned *factor* to the columns *Agility* and *Acceleration* we take a look to all the values included in these columns.

```r
#Printing the levels of the two numerical variables
levels(fc$Acceleration)[fc$Acceleration]
```

```
##  [1] "89"    "92"    "88"    "60"    "75"    "76"    "75"    "93"   
##  [9] "90"    "77"    "67"    "72"    "55"    "72"    "74"    "48"   
## [17] "77"    "77"    "63"    "54"    "38"    "93"    "73"    "61"   
## [25] "79"    "85"    "76"    "93"    "64"    "72"    "62"    "86"   
## [33] "67"    "61"    "93"    "71"    "83"    "85"    "94"    "93"   
## [41] "58"    "75"    "73"    "89"    "91"    "74"    "72"    "54"   
## [49] "76"    "76"    "80"    "32"    "71"    "79"    "72"    "72"   
## [57] "79"    "75"    "69"    "75"    "57"    "76"    "69"    "89"   
## [65] "82"    "72"    "75"    "76"    "49"    "64"    "76"    "78"   
## [73] "78"    "64"    "67"    "33+10" "52"    "44"    "70"    "68"   
## [81] "68"    "63"    "72"    "57"
```

```r
levels(fc$Agility)[fc$Agility]
```

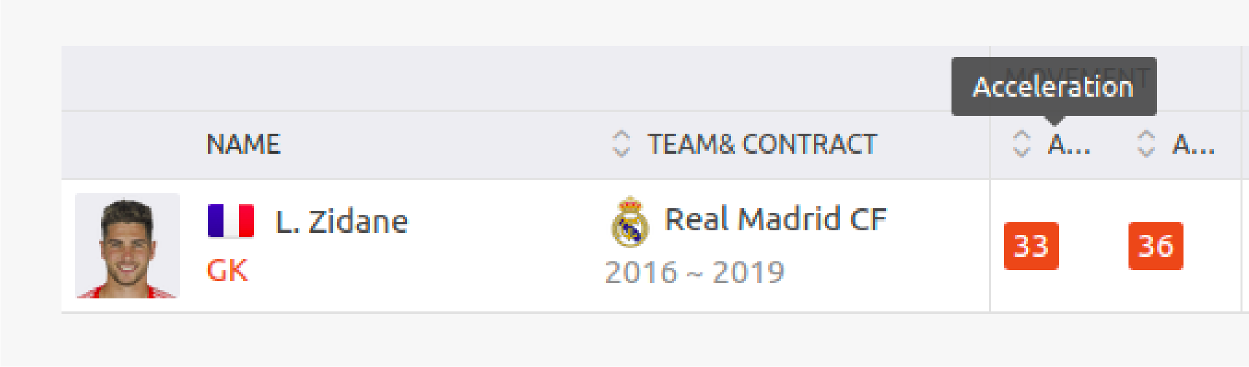
```
##  [1] "89"   "90"   "86"   "71"   "79"   "80"   "93"   "77"   "86"   "77"  
## [11] "72"   "92"   "58"   "79"   "83"   "58"   "77"   "58"   "66"   "60"  
## [21] "37"   "90"   "83"   "60"   "78"   "88"   "82"   "90"   "60"   "68"  
## [31] "69"   "70"   "72"   "58"   "89"   "67"   "82"   "77"   "91"   "93"  
## [41] "64"   "82"   "77"   "93"   "88"   "80"   "60"   "48"   "69"   "70"  
## [51] "83"   "43"   "64"   "71"   "76"   "72"   "68"   "73"   "60"   "73"  
## [61] "55"   "89"   "69"   "78"   "75"   "74"   "80"   "75"   "54"   "65"  
## [71] "79"   "79"   "83"   "71"   "71"   "36+2" "58"   "38"   "71"   "63"  
## [81] "65"   "54"   "60"   "44"
```

We see that there are two values “33+10” and “36+2” that are not numerical. This is the cause of having a wrong column type. The instance with the wrong values is:

```r
# Printing the instances with wrong values.
fc[76,]
```

```
##             Name Agility Acceleration           Club
## 11633 L. Zidane    36+2        33+10 Real Madrid CF
```

As I commented before, the data used here come from the Kaggle’s website. However, it have scraped from the web page: www.sofifa.com.
By taking a look to the original data definition we realize that the values need to be truncated at the two first digits.



Acceleration

NAME	TEAM& CONTRACT	A...	A...
L. Zidane GK	Real Madrid CF 2016 ~ 2019	33	36

.

```
# Truncating the values of Acceleration and converting them to numerical values.
fc[, 'Acceleration'] <-as.numeric(substr(levels(fc$Acceleration)[fc$Acceleration], 0, 2))

# Truncating the values of Agility and converting them to numerical values.
fc[,'Agility'] <- as.numeric(substr(levels(fc$Agility)[fc$Agility], 0, 2))

#Testing that the column types are fine
sapply(fc, class)
```

```
##         Name      Agility Acceleration         Club 
##     "factor"    "numeric"    "numeric"     "factor"
```

Now that the first issue is solved, let us observe the main descriptive statistics of the data:

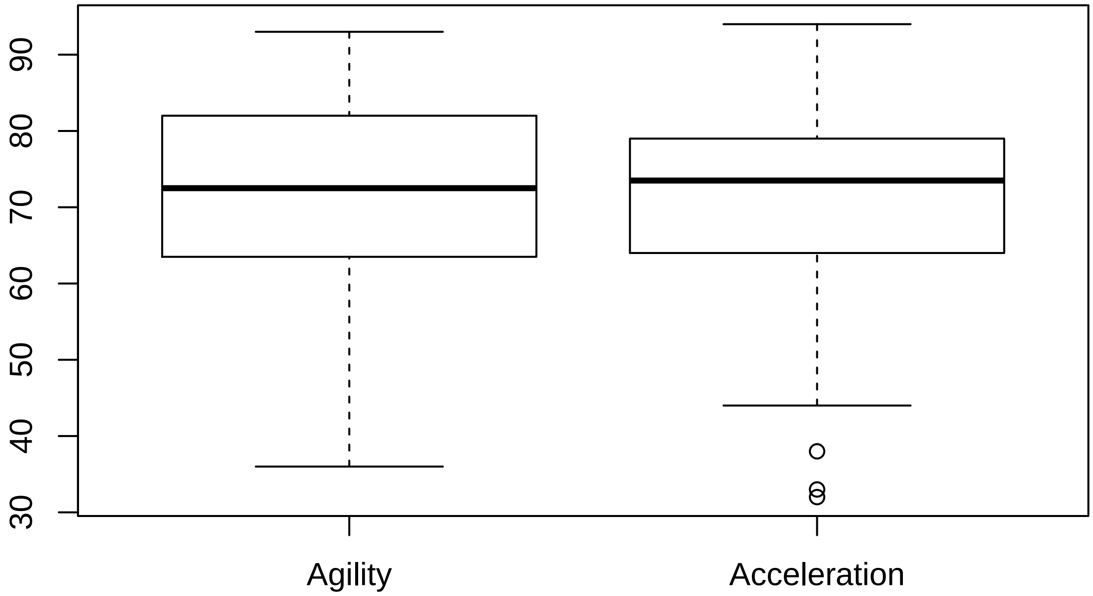
```
# Descriptive statistics related to the numerical variables.
# Note: requires package "pastecs"
round(stat.desc(fc[,c('Acceleration', 'Agility')], basic = T,  desc = T), 2)
```

```
##              Acceleration Agility
## nbr.val             84.00   84.00
## nbr.null             0.00    0.00
## nbr.na               0.00    0.00
## min                 32.00   36.00
## max                 94.00   93.00
## range               62.00   57.00
## sum               6044.00 6035.00
## median              73.50   72.50
## mean                71.95   71.85
## SE.mean              1.48    1.48
## CI.mean.0.95         2.95    2.95
## var                184.41  184.73
## std.dev             13.58   13.59
## coef.var             0.19    0.19
```

At the first sight is not observed other type of abnormalities in our data set. It is important to remark the no existence of **NA** or **NULL** values which can lead to another type of data processing techniques. Additionally, we can observe that both attributes have a similar range and equal variances.

Using the boxplots we can look for outliers in the data, this is the next step of this exploratory analysis.

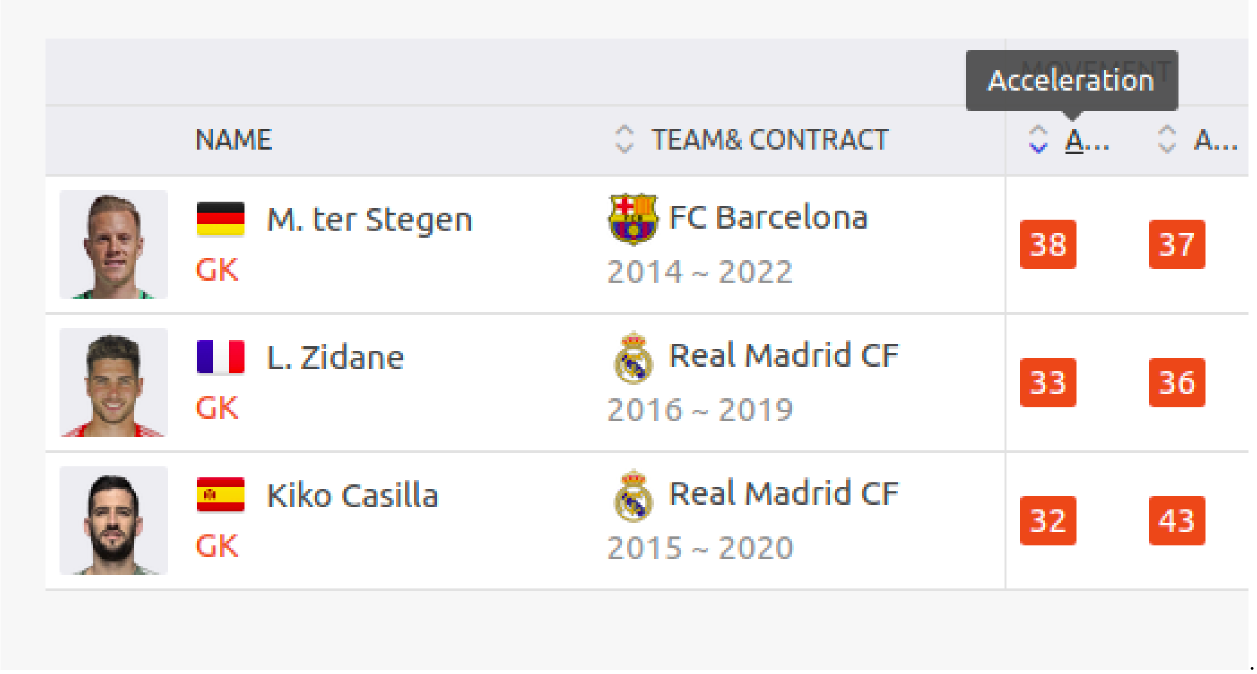
```
# Making the boxplots and filtering the data set.
boxplot(fc[, 2:3])
```



```
fc[fc$Acceleration < 40, ]
```

```
##                Name Agility Acceleration           Club
## 79    M. ter Stegen      37           38    FC Barcelona
## 468    Kiko Casilla      43           32 Real Madrid CF
## 11633     L. Zidane      36           33 Real Madrid CF
```

```
fc <- fc[ -which(fc$Name %in% fc[fc$Acceleration < 40, ]$Name), ]
```



NAME	TEAM& CONTRACT	A...	A...
M. ter Stegen GK	FC Barcelona 2014 ~ 2022	38	37
L. Zidane GK	Real Madrid CF 2016 ~ 2019	33	36
Kiko Casilla GK	Real Madrid CF 2015 ~ 2020	32	43

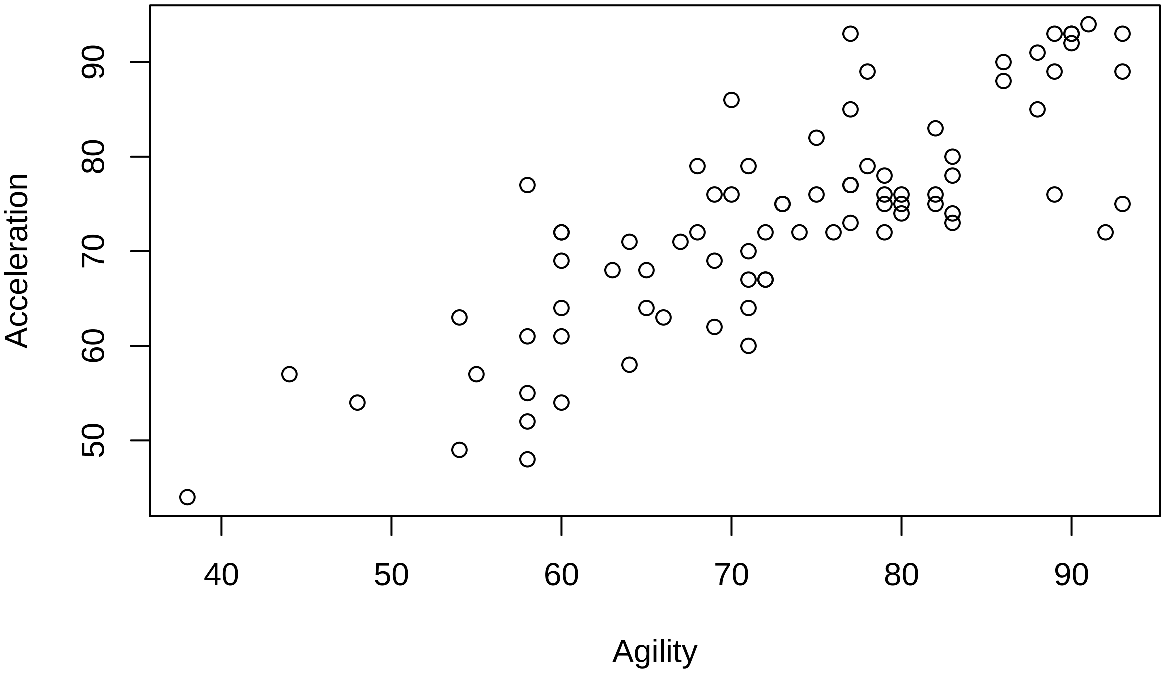
.

The column *Agility* seems not have any outlier, but regarding to the *Acceleration* there are 3 players that could be considered as outliers (they are 1.5 smaller than Q1 - 1.5*IQR). Could we assume that those values are correct? Should we remove them? Why? These values correspond to the goalkeepers of two out of the three teams under analysis. It have sense to consider them as an outlier because the goalkeepers do not need to be running to every point of the field, then it seems reasonable that some of them have a small acceleration.

I have decided to remove them because our analysis is focused on how the Acceleration is affected by the Agility, not in creating a complete team for the match.

To finalize the exploratory analysis let us to create a scatter plot using the two numerical variables in order to see how they are related.

```
# Scatter plot of the numerical variables.
plot(fc$Acceleration ~ fc$Agility, ylab = 'Acceleration', xlab='Agility')
```



```
#Calculating the correlation between these variables.
cor(fc[, 'Acceleration'], fc[, 'Agility'])
```

```
## [1] 0.8145926
```

The scatter plot shows a kind of linear relation between the variables *Agility* and *Acceleration*. The hypothesis of linear relationship between the two variables are also confirmed by the large correlation value obtained. A large correlation value (positive or negative) is always a good sign of linear relationships. On the contrary a correlation value close to zero indicates a non-existing linear relationship.

Some hypothesis to test: Questions.

Now that we have a whole overview of the dataset it is time to state our questions/hypothesis.

1. Is the *Acceleration* of a player influenced by its *Agility*?
2. How much will change the *Acceleration* of a player if the *Agility* is incremented in 2 units?
3. What is the *Agility* required to achieve an *Acceleration* of 81?

To solve these questions we can create a linear model as follows:

Linear Model

Using the procedure *lm* of *R* we can fit a linear model to our data set. In this case we fit a simple linear regression. We have thought that the acceleration has sense to depend on the player agility, that is why we take acceleration as the response variable and agility as the explanatory variable.

```r
# Definition of a simple linear model.
mod1 <- lm(Acceleration ~ Agility, fc)

#Accessing to the statistics calculated in the linear model.
summary(mod1)
```

```
##
## Call:
## lm(formula = Acceleration ~ Agility, data = fc)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.1013  -4.7286   0.0342   5.0510  16.5935
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.37329    4.62644   3.539 0.000676 ***
## Agility      0.77965    0.06246  12.482  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.808 on 79 degrees of freedom
## Multiple R-squared:  0.6636, Adjusted R-squared:  0.6593
## F-statistic: 155.8 on 1 and 79 DF,  p-value: < 2.2e-16
```

The output of the linear model gives us an idea of how the predicted and the predictor variables are related.

A huge number of insights can be extracted from the output of the linear model:

The **residuals** section of this model is telling us that in our model, 50% of the errors are in the range -4.7286 and 5.0510 (1Q and 3Q). For at least one instance there is an under-prediction and an over-prediction of approximately 17 points of acceleration which is a large number.

The **coefficients** section provides information about how relevant are the predictors used in our model. **Small p-values** means that the coefficients are different from zero and they should be included in the model. In our case, both coefficients are statistically different from zero.

The **Multiple R-squared** value is a criterion of how well our model captures and describe the variability in the data set. Models with this value close to one are the best choice to explain the variability of the dependent variable. In this case our model is explaining approximately the 66% of the variability of the Acceleration. This means that 66% of the differences that exists in the acceleration is a consequence of having different levels of agility.

The **Adjusted R-squared**, in this case provides the same information than the Multiple R-square because the linear regression is simple. In case of having more than one explanatory variable, it takes into account the number of parameters of the model. The **adjusted** R^2 is the one considered when one wants to compare models with a different number of parameters.

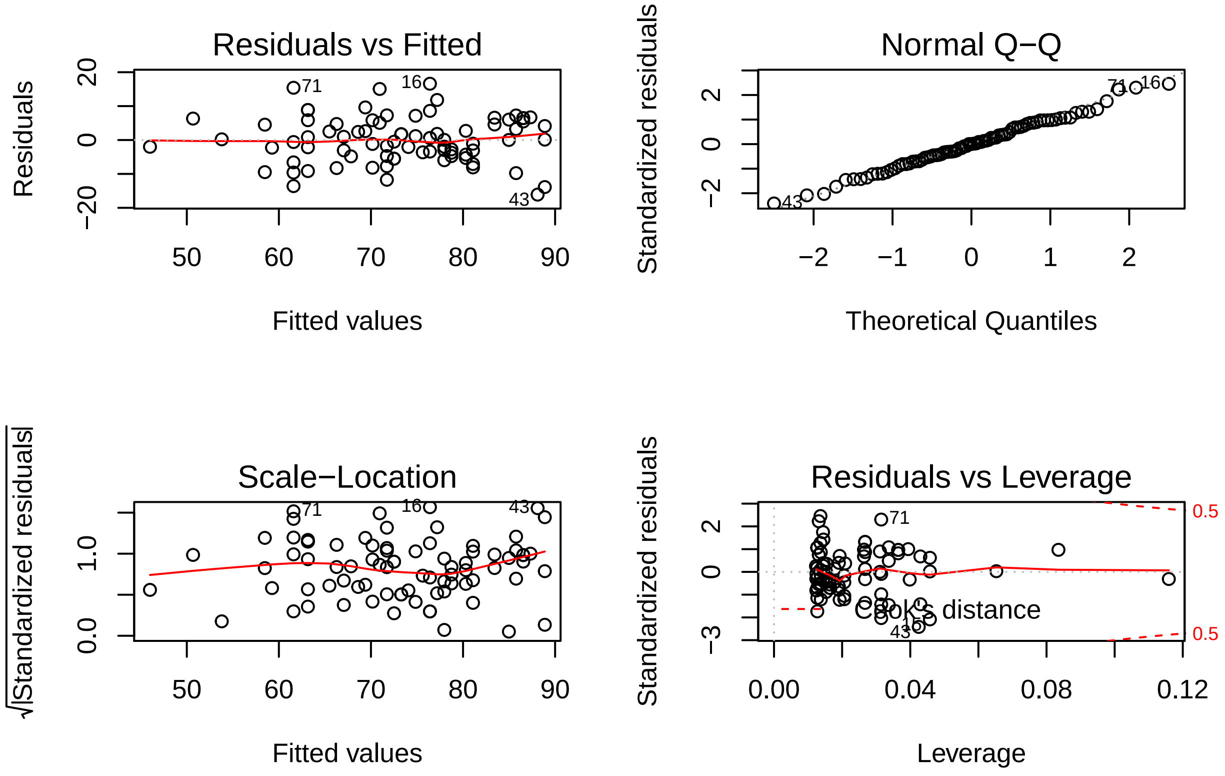
The **Residual Standard Error** give us an idea of the precision of our predictions, 95% of the errors associated with our predictions belong to the interval $(-1.96\,\hat{\sigma}, 1.96\,\hat{\sigma})$. Hence, this can be interpreted as the average amount that the observations deviate from the estimated regression line.

The **Degrees of freedom** are the number of data points, used to estimate the coefficients, minus the number of parameters in the model.

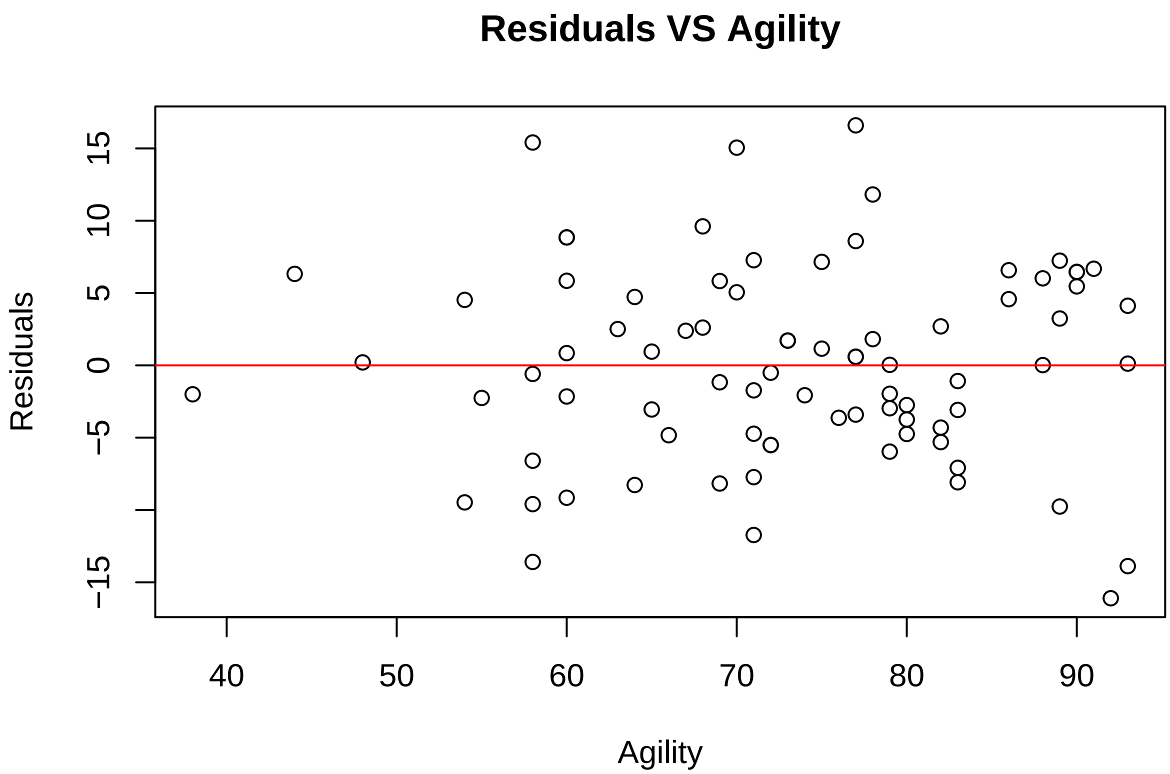
The **F-Statistic** is a good indicator of the existence of a relationship between the response and the predictor variables. It is the statistic that allows to test the *Omnibus test*. Its significance means that our model explains globally an important amount of the variability of the response variable. A F-Statistics far away from 1 is a good sign. On the other hand, this value is highly related to the number of predictors and the number of data points. When the number of data points is large, an F-Statistic a little farther away from 1 is enough to reject the null hypothesis. Otherwise, if the number of data points is small, a large F-Statistic is required to reject the null hypothesis.

Linear Model hypothesis

```r
# Plotting the residual plots of the linear model.
par(mfrow = c(2,2))
plot(mod1)
```



```r
# Specific plot residuals VS the attribute Agility.
par(mfrow = c(1,1))
res <- residuals(mod1)
plot(res ~ fc$Agility, ylab = 'Residuals', xlab = 'Agility', main = 'Residuals VS Agility')
abline(h = 0, col='red')
```



To consider the resulting model as a good model for our data, it is necessary to validate that the hypothesis of the linear regression are fulfilled. This can be performed by means of the visual analysis of the previous residual plots. In this case, we observe that the errors are normally distributed (qq-plot), and that the residual values are independent since no pattern is observed, the same can be stated for the variance of the residuals, it seems to be more or less constant all along the fitted values range.

In what follows it appears the *Anova* table associated to our regression model. Observe that the F-value is equal to 155.8, which is the same value that appears at the end of the summary of our linear model. This is the statistic for the *Omnibus Test*, and we conclude that our covariates globally explain a significant part of the variability present in the response variable.

```
# Anova table related to the linear model.
anova(mod1)
```

```
## Analysis of Variance Table
##
## Response: Acceleration
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Agility    1 7221.1  7221.1  155.81 < 2.2e-16 ***
## Residuals 79 3661.2    46.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Parameter Interpretation

From the summary of the linear model we have that our final model is $Y = 0.77965x + 16.37329 + e$. Meaning that every new agility unit will increase the acceleration in an average of 0.77965 units. On the other hand the intercept is the value of the acceleration when the agility is equal to zero.

Answer hypothesis and Final Comments

After performing the whole analysis, we can state that the Acceleration is linearly related with the Agility of each player. In the particular case when the Agility is increased by two units the Acceleration will increase in average approximately 1.5593 units. Finally, the Agility required for a player to have an Acceleration of 81 is approximately 82.89 units of agility, which is obtained substituting y by 81 in the regression line and isolating the x.