Northeastern University

ALY6015 Intermediate Analytics

2019 Spring Quarter

Instructor: Emre Ozdemir

Week 6 Final Project Report

Team #4: Huiwen Li

Jingjing Xie

Completion Date: May 18, 2019

**Introduction**

In this final project, we are going to predict the value of a FIFA player by using FIFA 19 complete player dataset (from Kaggle). Firstly, we will clean the original data, such as dropping some unnecessary variable, dealing with missing values and formatting data. Then, we will do some data exploration to analyse some variables and find the meaning behind it and to explore the relationship between dependent variable and independent variables. After that, we are going to build a linear regression model to predict the value of a FIFA player. We try three different methods (OLS, stepwise selection, Lasso) to select necessary independent variables and select a best model. Finally, we will evaluate the final model using three methods: 5-fold cross validation, predicted R square and MES.

**Analysis**

**1. Data pre-processing**

**1.1. Preview the dataset**

Firstly, we read the dataset and get a head of the dataset by using “head()” function. From the head of the dataset, we can know that there are 18207 observations and 89 variables, which show the latest FIFA 2019 players attributes, like Age, Nationality, Overall rating, Value… We also check the type of each variables by using the function of “str()”, and we find there are types of variables : one is integer, the other is factor. For example, Name, Nationality, Value, Wage are factor variables; Overall rating, Potential rating are integer variables.

**1.2. Drop some unnecessary columns**

1. Drop all the position-specific rating columns (from 29 to 54 columns)
2. Drop some columns which are not useful, such as 'ID', 'Photo', 'Flag', 'Club.Logo', 'Jersey.Number', 'Joined', 'Special', 'Loaned.From', 'Body.Type'.
3. Drop all rows with NA values. As there are large observations (18207 observations), it doesn’t significantly reduce the dataset dropping the rows with NA values.

After dropping the unnecessary columns, we preview the dataset again, and we find that there are only 47 variables now, and there is no NA value in the dataset.

**1.3. Transform currency into number**

The columns “Wage” ,“Value” and “Release.Clause” present how much money a player earned, which is a currency format. In order to predict the value of a player in the analysis part, we need to transform currency into numbers.

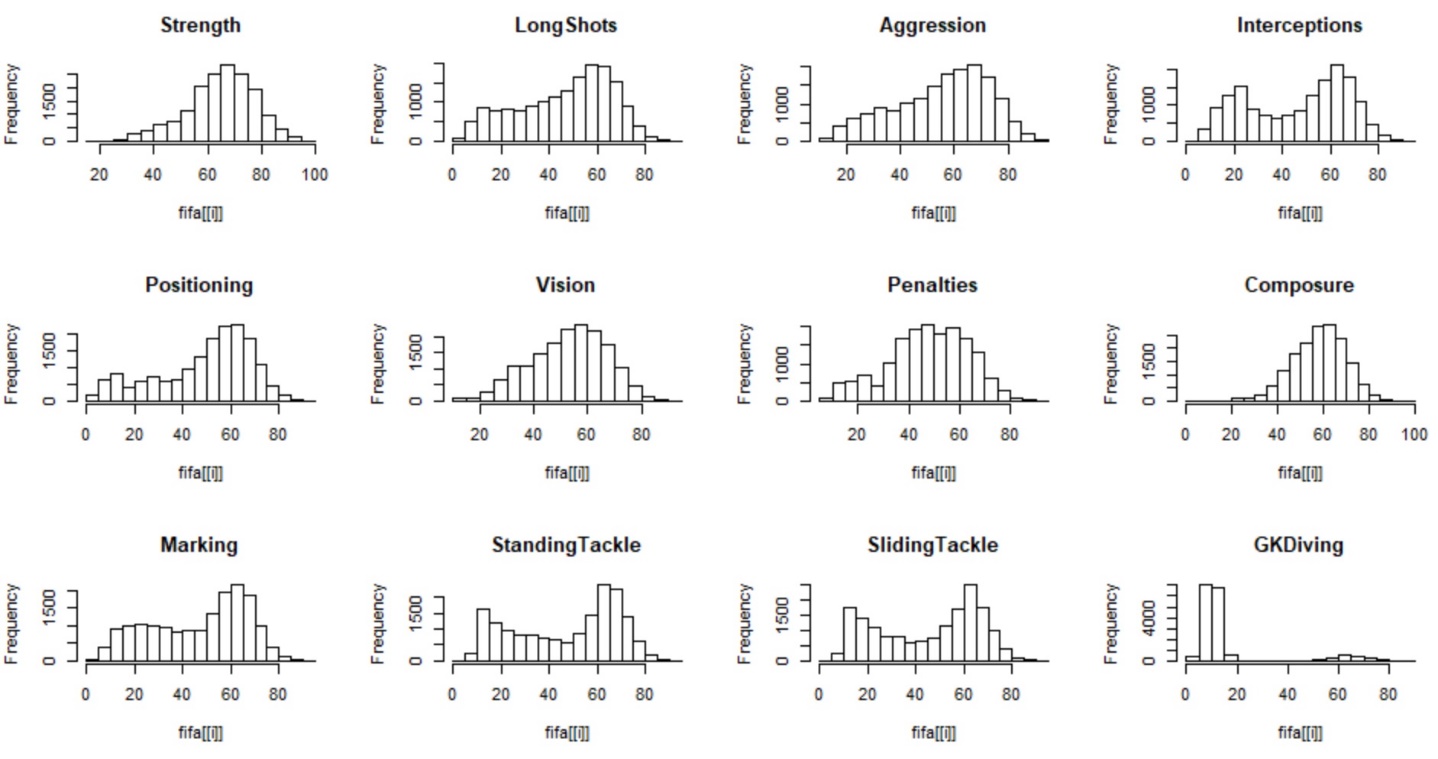
**1.4. Redefine positions**

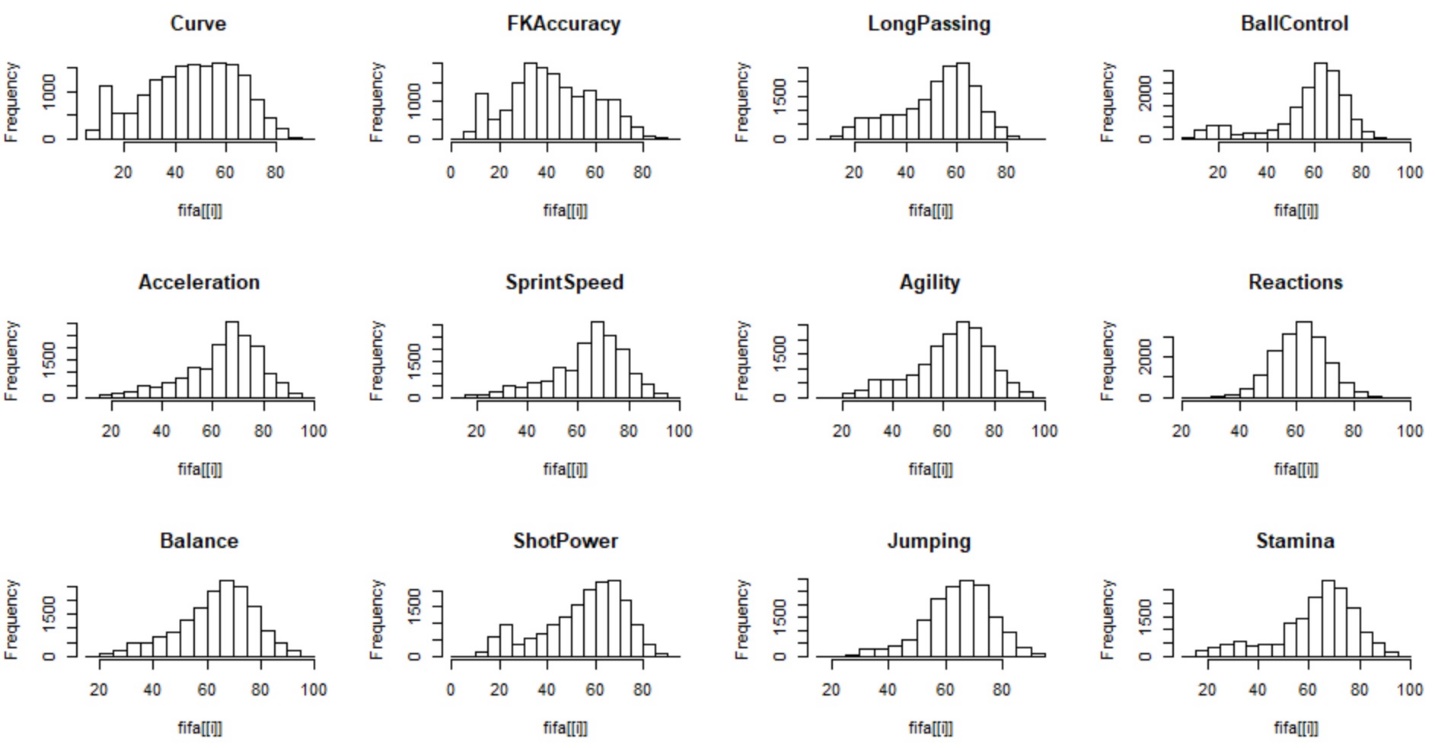
Column “position” shows the football pitch field positions for players. This column indicates detailed position, which include 27 positions. So we simplify the positions, for example, “CAM” Central Attacking Midfielder and “RDM” Right Defending Midfielder are all “Midfielder”, so we redefine “CAM” and “RDM” to be “Midfielder”.

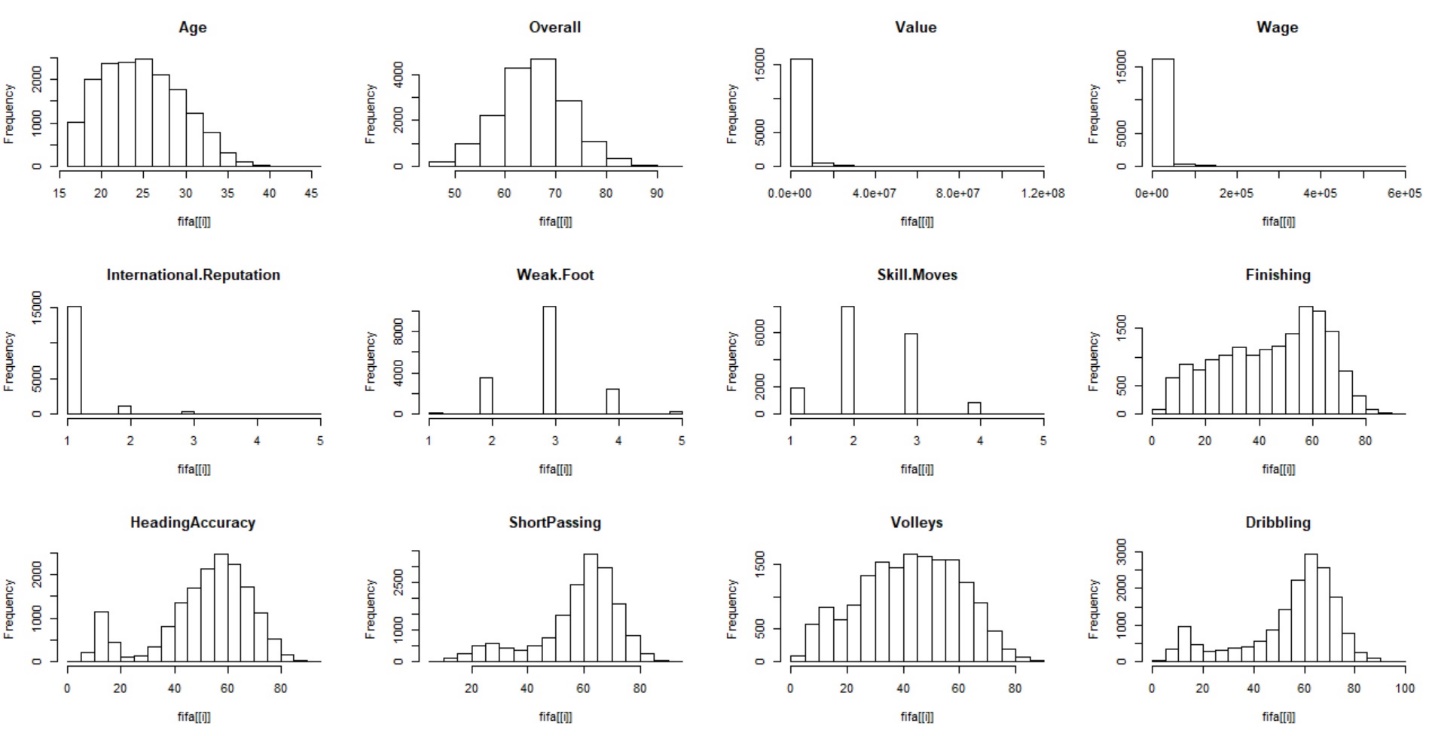
**2.** **Exploratory data analysis**

**2.1. Each variable distribution**

Firstly, we get the distribution of each variable.







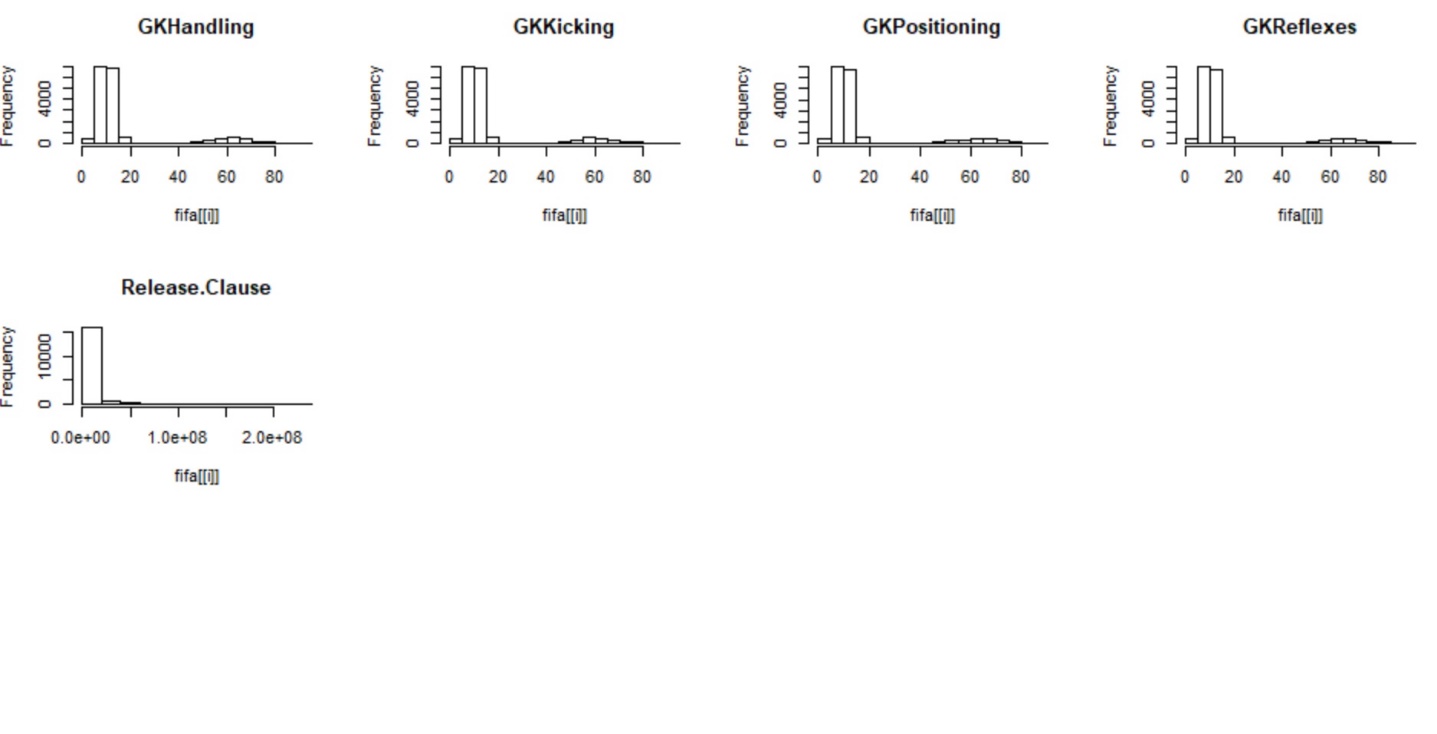


Figure 1: correlation plot between dependent variables and predictors

From the distribution of each variable, we can see that most of the variables are normal distributed, except for some variables about money, such as “Value”, “Wage”, “Release.Clause”, and some variables about special position skills, such as “GKHandling”, “GKKicking”, “GKPositioning”, “GKReflexes”, which are the scores of goal keepers’ skills.

As “Value”, “Wage”, “Release.Clause” are highly skewed, we will normalized these values using log-transformation.

**2.2. Normalized highly skewed values (“Value”, “Wage”, “Release.Clause”)**

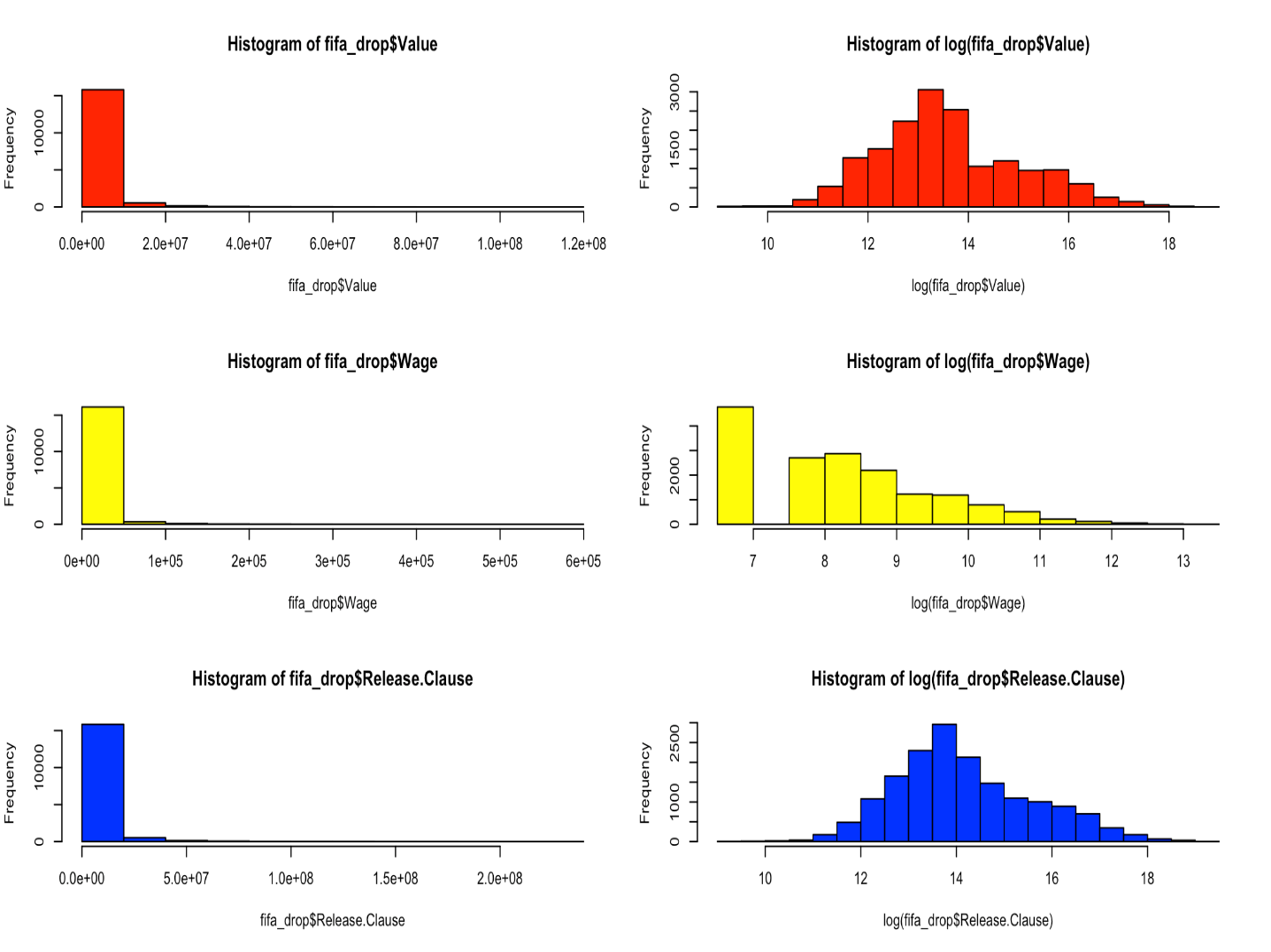


Figure 2: Histogram of “Value”, “Wage” and “Release.Clause”

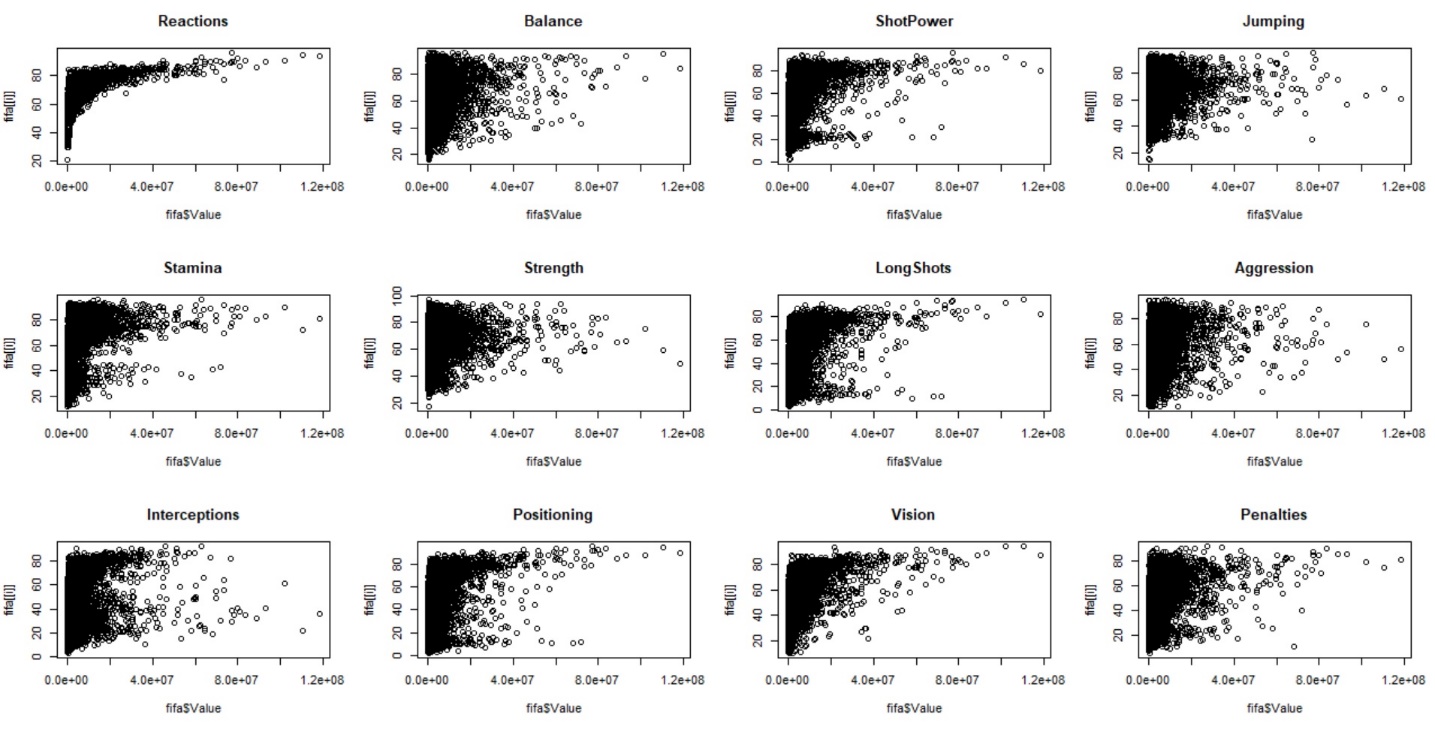
From this figure, we can find that the log-format of “Value” and “Release.Clause” is

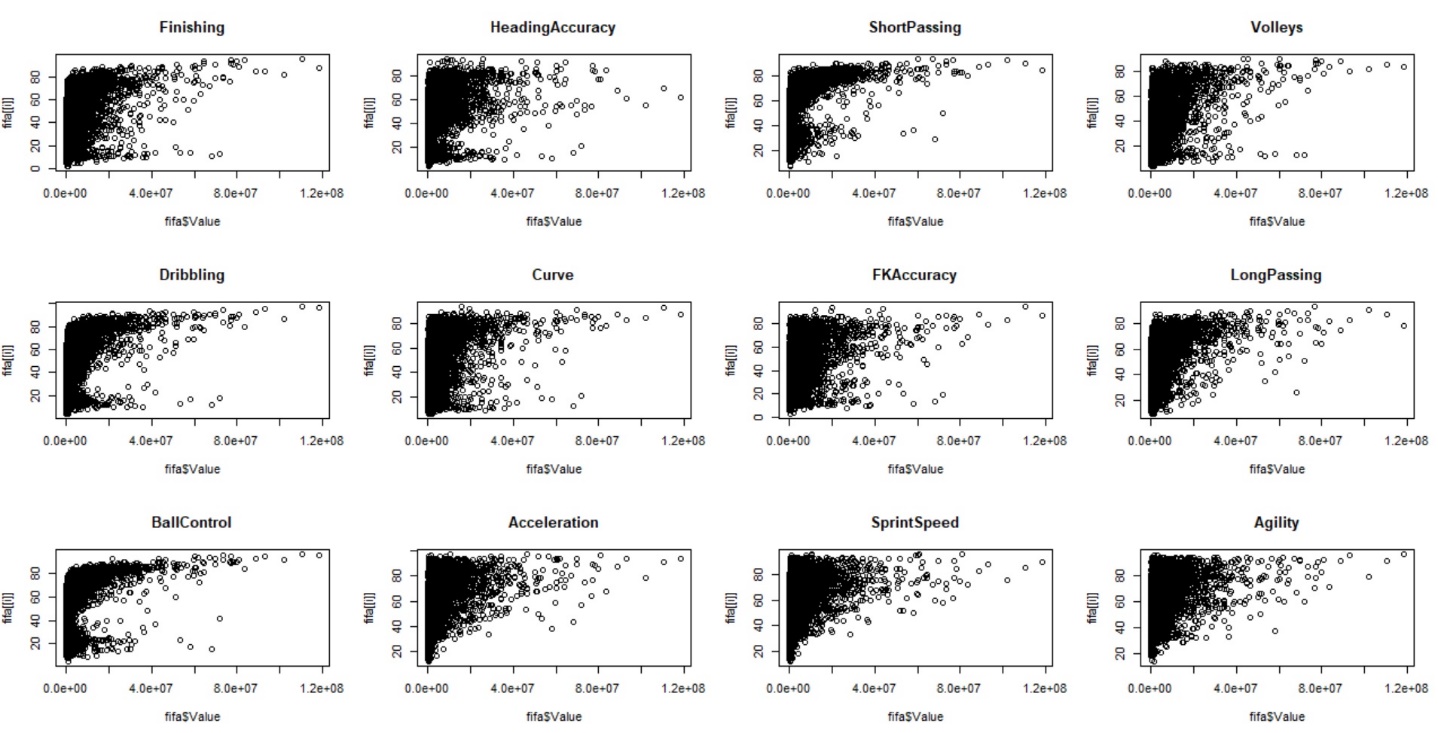
normally distributed. And “Wage” is not that highly skewed.

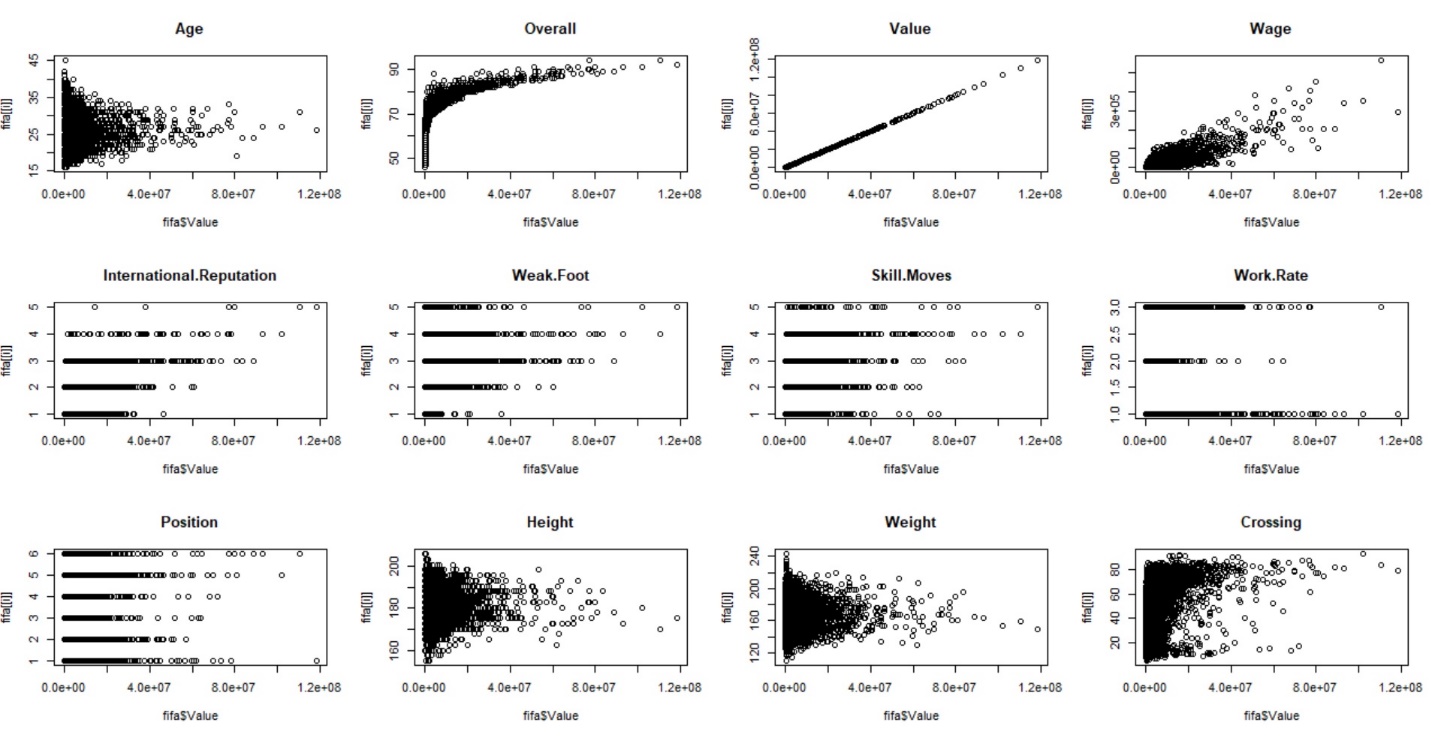
**2.3. Correlation plot between “Value” and other variables**

The following figure is the correlation plot between “Value” and other variables. From this plot, we can find that “Value” has strong positive relationship with “Wage”, “Overall” and “Release.Clause” and also has some positive relationship with some skills, such as

“Reaction”, “Vision” etc.







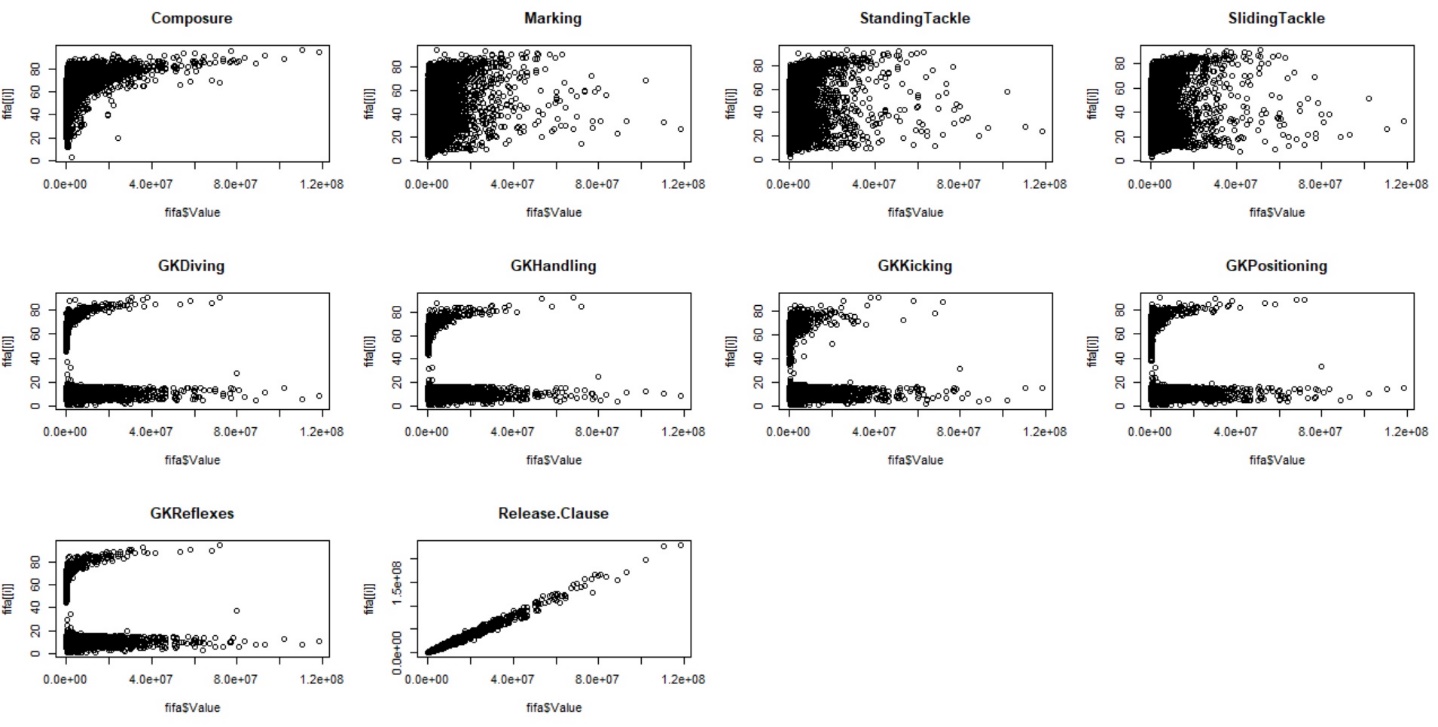


Figure 3: scatter plot of each variables

This correlation also shown in the heatmap as below. From this heatmap, we can find that “Value” has strong positive relationship with “Wage”, “Overall”, “International.Reputation” and “Release.Clause”. Also, we find that there are some relationship between independent variables, which is multicollinearity. Such as “Marking” and “SlidingTracking” are highly related.

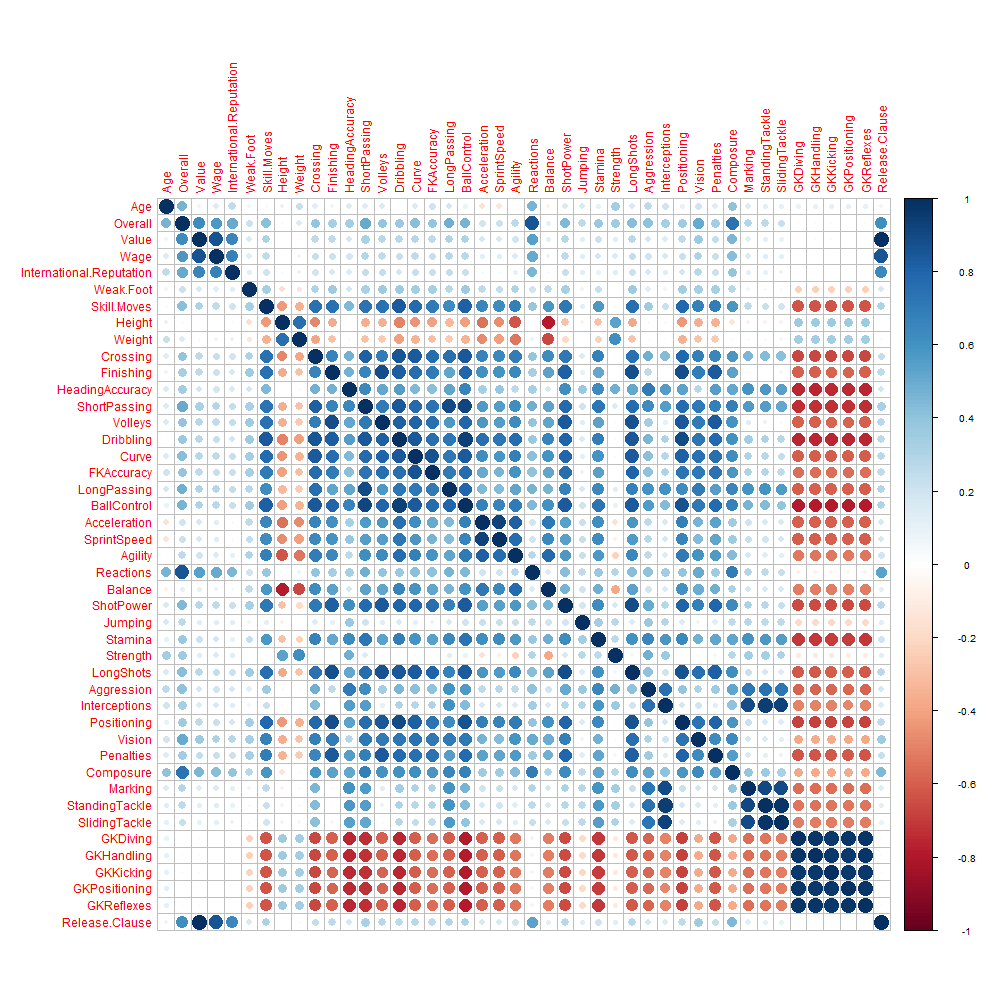


Figure 4: heatmap of each variables

**2.4 Top 20 highest value for players**

This figure shows the top 20 highest value of players. We can see that the value of Neymar.Jr is highest, following with L.messi.

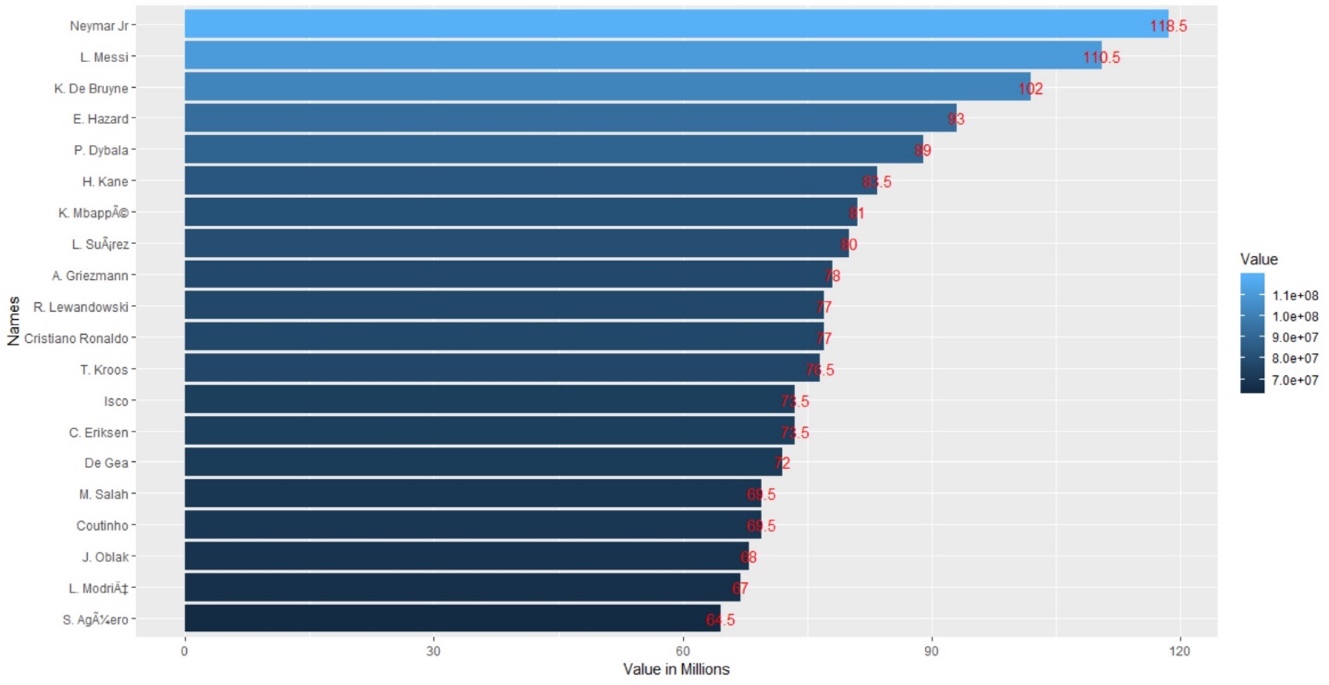


Figure 5: top 20 highest value for players

**3. Randomly Split data into train and test**

After exploratory data analysis with summary statistics and various graphs, we are ready to perform model building. Before that we need to randomly split the data into training and testing set. We choose to use 80% of the original data for training the model and the remaining 20% to test the model. Here below are the R codes for splitting the data.

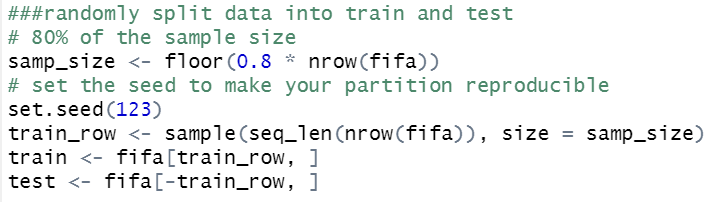
****

Figure 6: Randomly split data into train and test

**4. Building models**

**4.1. Ordinary Least Square (OLS) Regression**

As stated earlier, the main objective of the project is to build a model to predict player’s Value with given information, the response variable in the model is Value. We will iteratively fit a model by using OLS regression. The first OLS model we try is the full model, including all variables except Value as the independent variables.

**lm\_fit1 <- lm(Value ~ ., data=train)**

The results of lm\_fit1 model is shown in the following table.

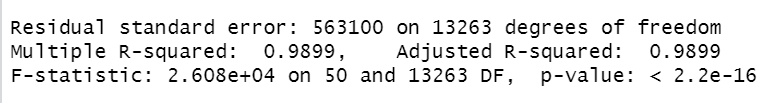


Figure 7: Results of lm\_fit1

The lm\_fit1 model indicates 14 significant predictor variables and they are Overall, Wage, International.Reputation, Position, Crossing, Finishing, Volleys, FKAccuracy, LongPassing, Stamina, Marking, StandingTackle, GKHandling, Release.Clause. The Adjusted R2 is pretty good 0.9899 but the Residual Standard Errors are huge due to large numbers in Wage and Release.Clause. Next we need to check the residual plots next to confirm the model performance.

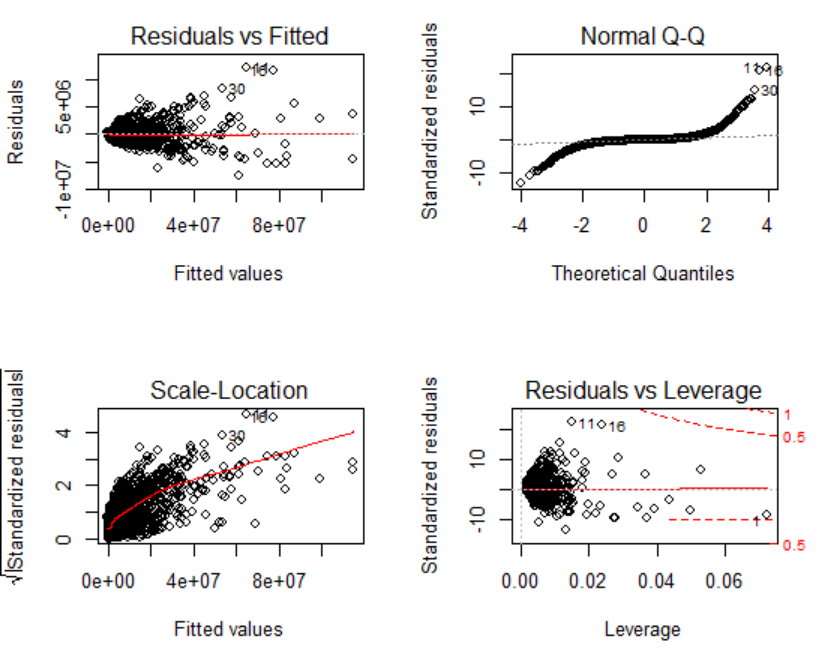
****

Figure 8: Residual plots of lm\_fit1

From the above graph, it suggests the residuals do not exhibit homoscedasticity or equal variance, as they tend to cluster at where fitted values are lower but get very much dispersed when fitted values are higher. In addition, the residuals are not normally distributed according to the normal Q-Q plot. All these information suggest lm\_fit1 is not a good model hence we should improve it.

As shown in the exploratory data analysis, the distributions of Value, Wage and Release.Clause are highly skewed to the right. We can apply logarithm on these variables to adjust the distribution. Hence we try the following model in lm\_fit2.

**lm\_fit2 <- lm(log(Value) ~ Overall + log(Wage) + International.Reputation + Position +**

**Crossing + Finishing + Volleys + FKAccuracy + LongPassing + Stamina +**

**Marking + StandingTackle + GKHandling + log(Release.Clause), data=train)**

The results of lm\_fit1 model is shown in the following table.

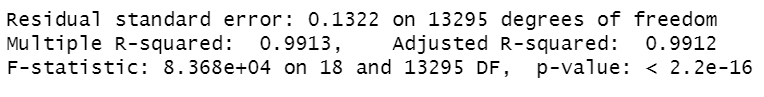
****

Figure 9: Results of lm\_fit2

By looking at the Adjusted R2 it is slightly improved to 0.9912 and the Residual Standard Error is 0.1322 which is much better compared to lm\_fit1 model. Let’s take a look at the residual plots.

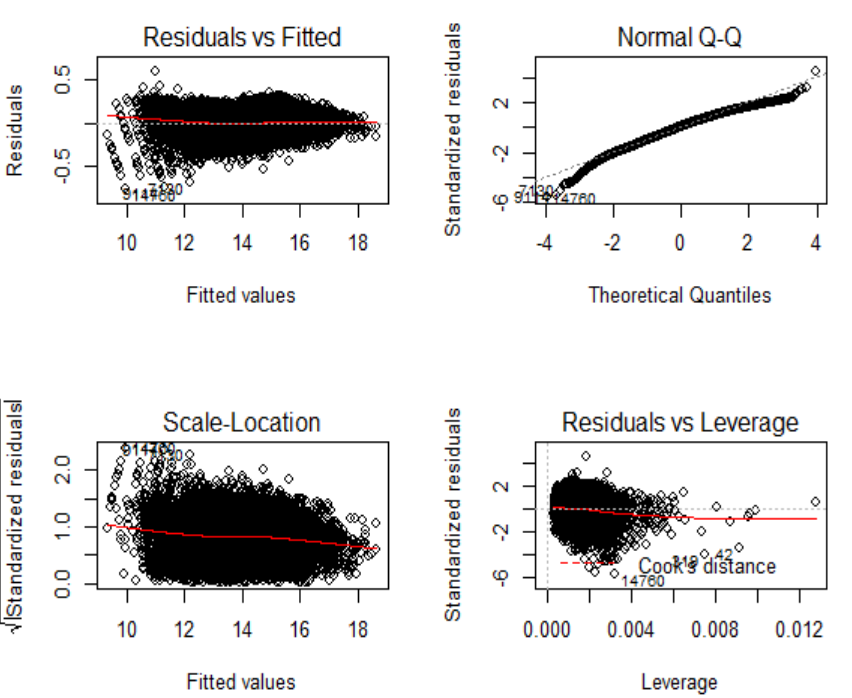


Figure 10: Residual plots of lm\_fit2

From the above graph, the residuals do not clearly show patterns at different fitted values and are also very close to normal distribution. No obvious heteroscedasticity pattern is found. The lm\_fit2 model satisfies most of the assumptions, at the same time achieves high Adjusted R2, hence we are happy with it. However, before we can conclude the final OLS model, we still need to check for multicollinearity of the predictor variables.

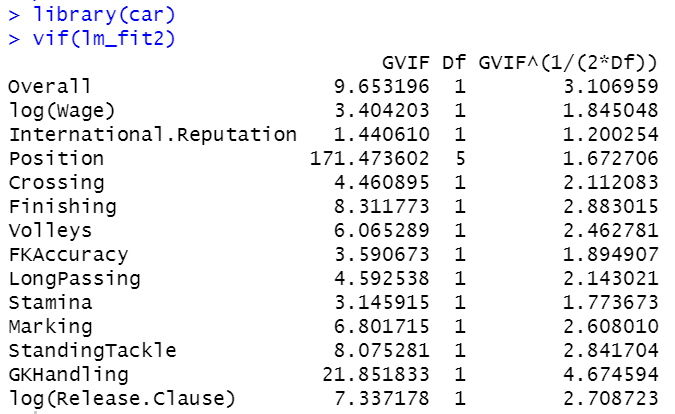


Figure 11: Multicollinearity results of lm\_fit2

The VIF score exceeding 10 are signs of serious multicollinearity. Based on the above results, Position and GKHandling have serious multicollinearity hence we should remove both from the model. We will define lm\_fit3 model as follows.

**lm\_fit3 <- lm(log(Value) ~ Overall + log(Wage) + International.Reputation +**

**Crossing + Finishing + Volleys + FKAccuracy + LongPassing +**

**Stamina + Marking + StandingTackle + log(Release.Clause),**

**data=train)**

The results of lm\_fit3 model is shown in the following table.

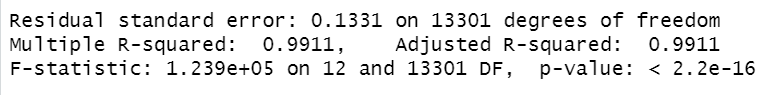
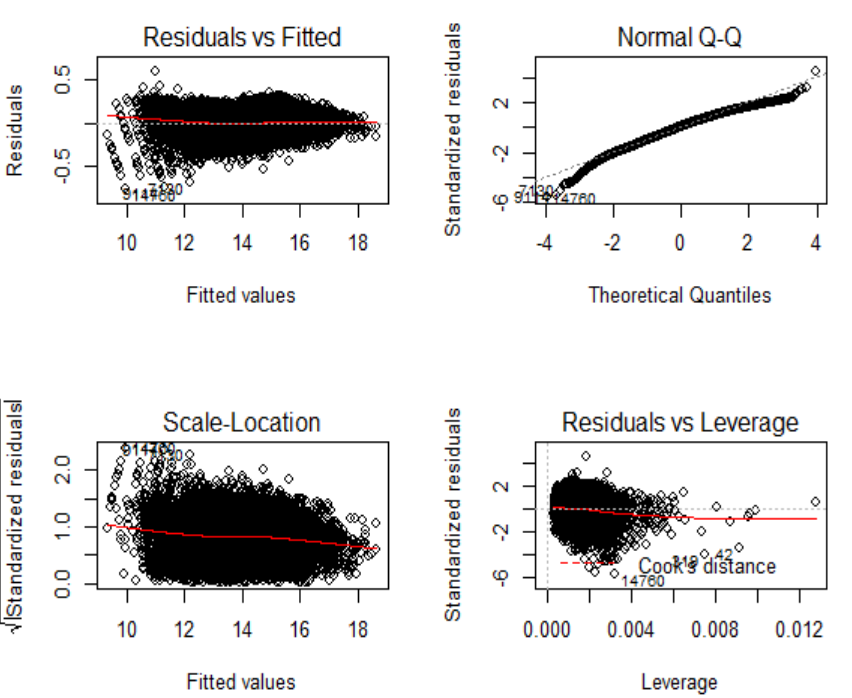


Figure 12: Results of lm\_fit3

Without Position and GKHandling, the Adjusted R2 did not really change and the Residual Standard Error is almost the same. Let’s check the residual plots.



Graph : Residual plots of lm\_fit3

Figure 13: residual plots of lm\_fit3

From the above residuals plots we do not observe clear patterns at various fitted values. The residuals are also close to normal distribution and do not have obvious heteroscedasticity pattern. Therefore we can conclude lm\_fit3 as the final OLS model.

**4.2. Stepwise Regression Model**

After trying manually fitting the model (Ordinary Least Square model), we will now try to use stepwise variable selection to find the best model based on AIC value. From the results, the final step model consists of 32 variables, as shown below.

**step\_fit <- lm(log(Value) ~ Age + Overall + log(Wage) + International.Reputation +**

**Weak.Foot + Skill.Moves + Position + Weight + Crossing +**

**HeadingAccuracy + Volleys + Dribbling + FKAccuracy + LongPassing +**

**Acceleration + Agility + Balance + Jumping + Stamina + Strength +**

**Aggression + Interceptions + Positioning + Vision + Penalties +**

**Composure + Marking + SlidingTackle + GKDiving + GKHandling +**

**GKPositioning + log(Release.Clause), data = train)**

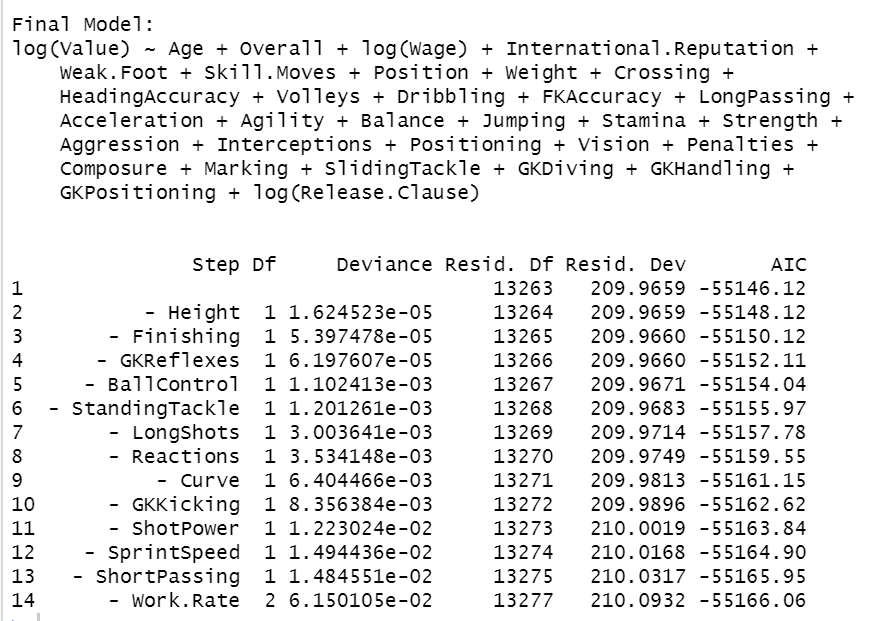
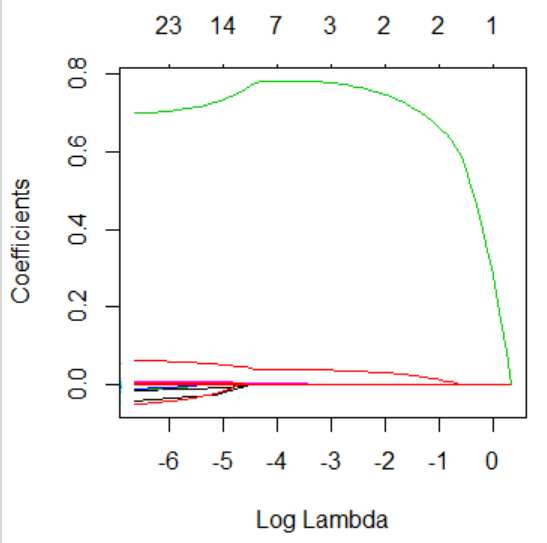
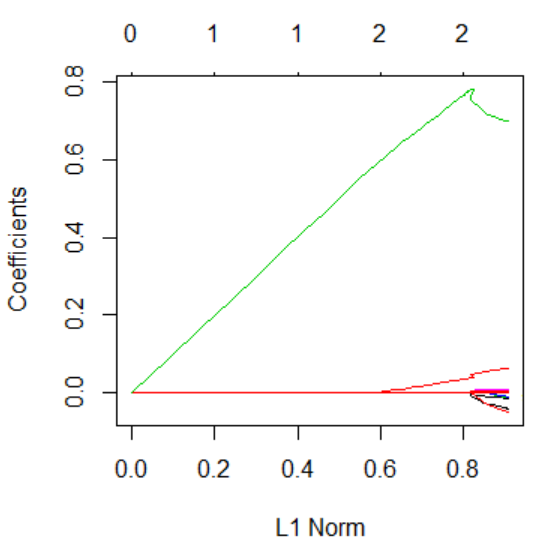


Figure 14 : Results of step\_fit

**4.3. LASSO Regression Model**

The output models by OLS and Stepwise regressions are not very ideal as both models consist of 10 or more predictor variables. Although the percentage of variance explained by both models are high, too many predictor variables make the models difficult to interpret. Hence we would like to try LASSO regularization which penalizes some insignificant variables and shrinks their coefficients to zero. At the same time LASSO regularization takes care of multicollinearity among variables.

The first step is to build the LASSO model with response and predictor variables. Next is to use plot.glmnet () function to plot the path of each predictor variable coefficients against the L1 norm and log lambda of the beta vector and we obtained the following two graphs. They show us path of each of predictor variable coefficients against the L1 norm of the beta vector.



Graph 15: Graphs of variable coefficient path

After that we use cv.glmnet () function to get the cross validation curve and the value of lambda that minimizes the mean cross validation error which is 0.0252. We then use this minimum lambda value to get the estimated beta matrix, as shown in the table below. According to the result, most of the predictor coefficients have been shrunk to zero, except six preditors: Overall, Skill.Moves, Finishing, Stamina, Positioning, Log(Release.Clause). That means the remaining six variables are important in explaining the variation in Value. The final model obtained from LASSO is:

**lasso\_fit <- lm(log(Value) ~ Overall + Skill.Moves + Finishing + Stamina + Positioning +**

**log(Release.Clause), data = train)**

**5. Model Performance Evaluation**

**5.1. AIC and BIC**

AIC and BIC are both penalized-likelihood criteria. They are often used for comparing models and selecting the best subset of predictor variables in regression analysis. AIC estimates the relative amount of information lost by a given model: the less information a model loses, the higher the quality of that model. In the process of estimating the information lost, AIC takes into the consideration of the trade-off between the goodness of fit of the model and the simplicity of the model. In other words, AIC tackles the risks of overfitting and underfitting. BIC is similar to AIC, the only difference is the size of the penalty. BIC penalizes model complexity more heavily.

|  |  |  |
| --- | --- | --- |
| Methods | AIC | BIC |
| Ordinary Least Square | -15902 | -15797 |
| Stepwise Selection | -17381 | -17096 |
| LASSO Regularization | -15348 | -15289 |

Figure 16: AIC and BIC of various models

According to the selection criteria, the lower the AIC/BIC the better the model. Hence the model from stepwise selection method has the best performance.

**5.2. K-fold cross validation**

As we obtain three different models from OLS, stepwise selection and LASSO regularization, we will use cross validation method to compare which model is better with lower mean square error.

With 5-fold cross validation on the model from Ordinary Least Square method, we have overall mean square error = 0.0177.

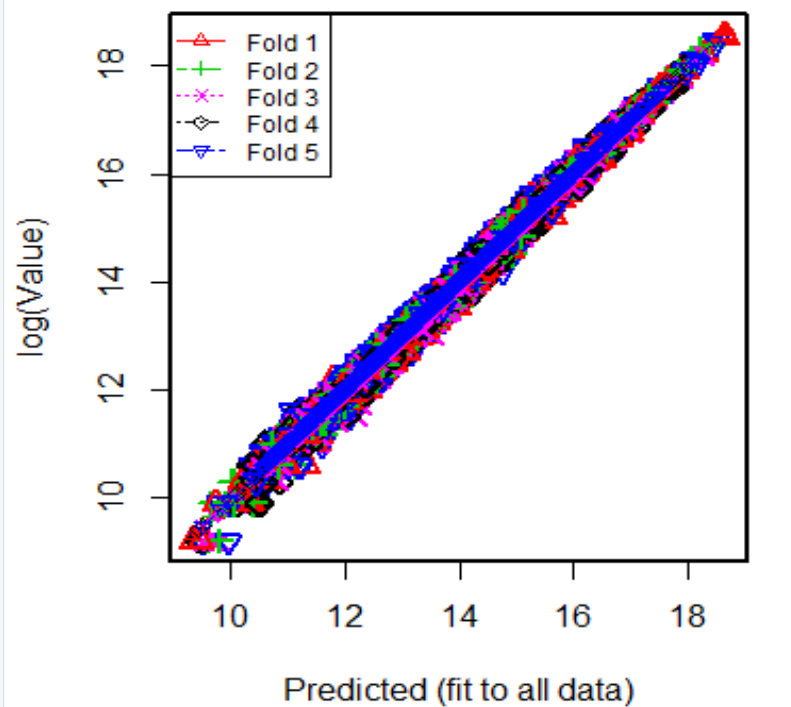


Figure 17: Cross-validation result on OLS model

With 5-fold cross validation on the model from Stepwise selection method, we have overall mean square error = 0.0159.

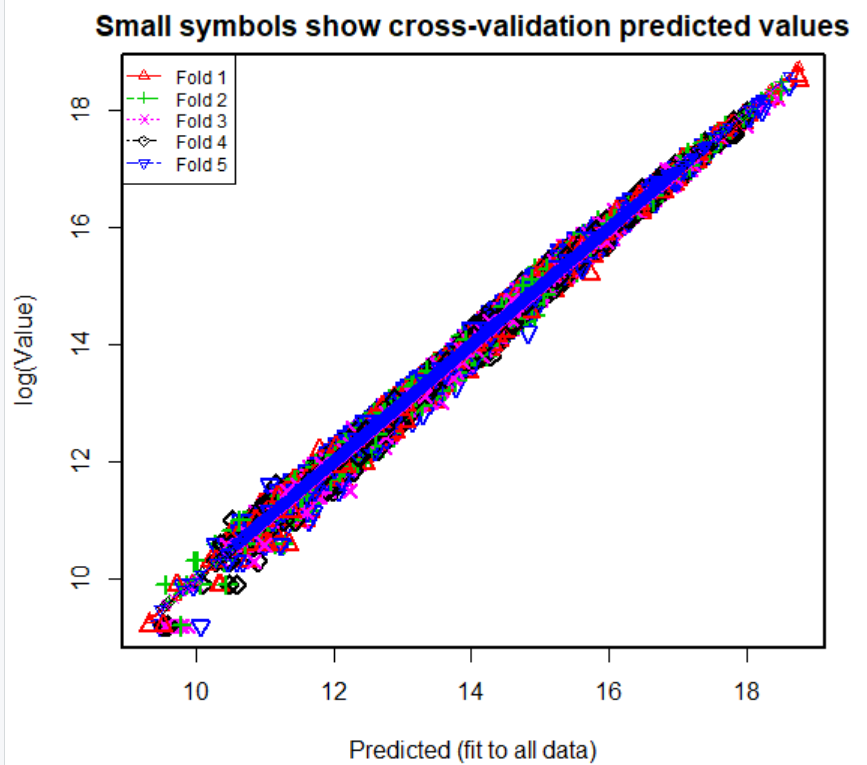


Figure 18 : Cross-validation result on Step model

With 5-fold cross validation on the model from LASSO regularization method, we have overall mean square error = 0.0185.

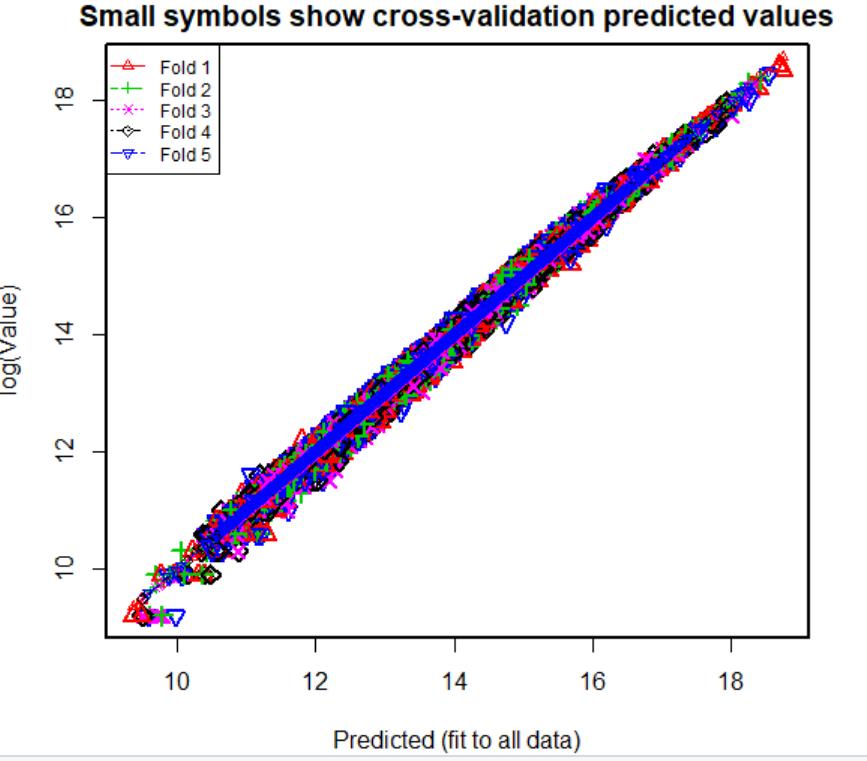


Figure 19 : Cross-validation result on LASSO model

|  |  |
| --- | --- |
| Model | Mean Square Error |
| Ordinary Least Square | 0.0177 |
| Stepwise Selection | 0.0159 |
| LASSO Regularization | 0.0185 |

From the above results, we see the performance of all three models are good with low mean square error (MSE). The Stepwise model has the lowest MSE 0.0159, followed by OLS model then LASSO model. The cross-validation plots also suggest good fit of the models. Let us summarize the cross-validation results in the following table.

Figure 20: Cross-validation result of all three models

**5.3. Prediction R2 and MSE**

In order to access the predicting abilities of each model, we predict fitted y values of each model with test data and calculate the predicted R2 for all the models. The predicted R2 value indicates the percentage of variation in the response variable being explained by the model. Therefore with higher predicted R2 value the model performs better. The results including MSE are summarized in the following table. The results show all three models have almost the same predicted R2 values, while MSE are also very close to each other.

|  |  |  |
| --- | --- | --- |
| Model | MSE | Predicted R2 |
| Ordinary Least Square | 0.0179 | 99.1% |
| Stepwise Selection | 0.0179 | 99.2% |
| LASSO Regularization | 0.0186 | 99.1% |

Figure 21: Prediction results of all models

The plots of predicted y values versus the actual y values for all three models are shown below. For all the models, all the points are close to a regressed diagonal line, which indicates good fit of the models. Since the performances of all the models are very close, we will select the simplest LASSO model because it provides simplicity and easy interpretation with only six predictor variables. Hence the final model is

**log(Value) ~ Overall + Skill.Moves + Finishing + Stamina +**

**Positioning + log(Release.Clause)**

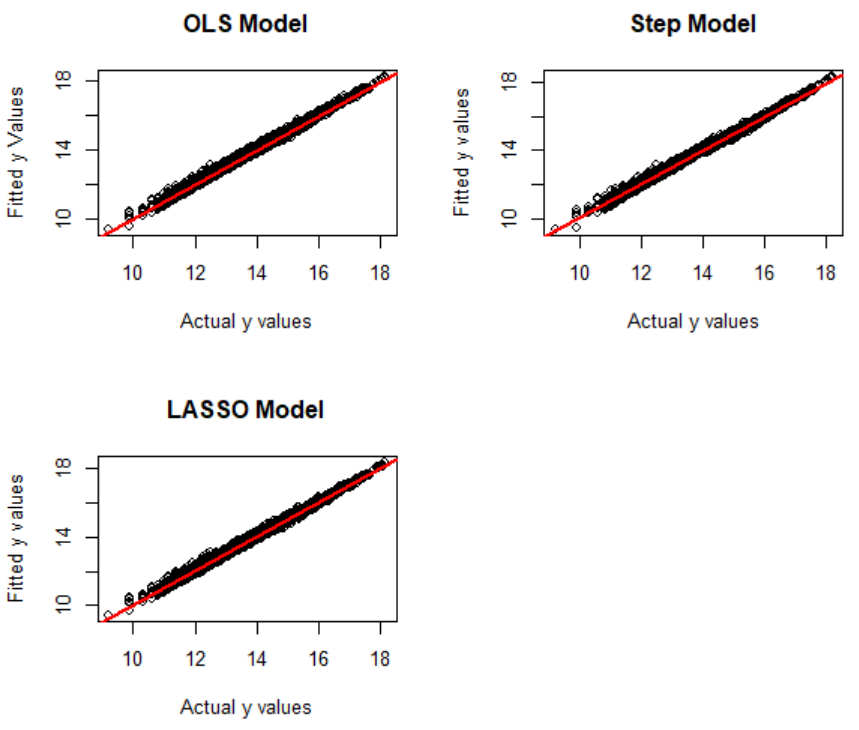


Figure : Plots of fitted y against actual y

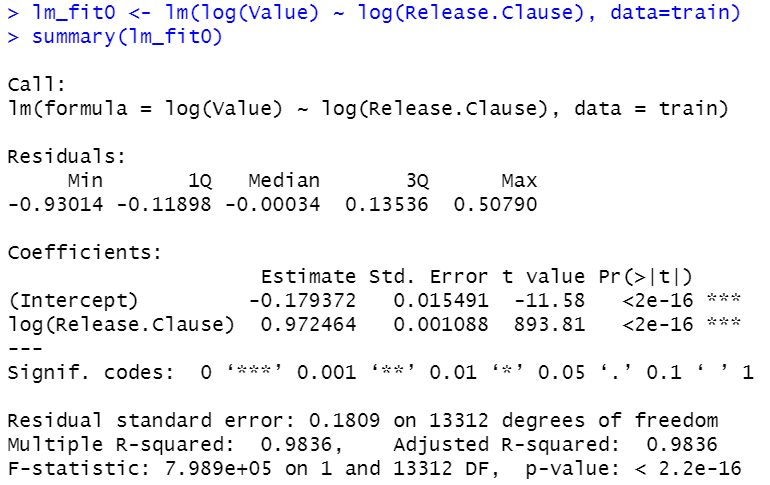
Figure 22: The plots of predicted y values versus the actual y values for all three models

**Conclusion**

We have tried three methods to build the prediction models and used various evaluation metrices to access the performance of each model. Since all the three models have very similar performances, according to the principle of parsimony we select the model from LASSO regularization because it is relatively simple with easy interpretability that there are only 6 predictor variables. However, Stepwise selection method produces a model with 32 predictor variables whereas OLS method with 12 predictor variables.

For the next step, we would like to try using random forest algorithm to fit another linear regression model. The above three methods we have tried in building the model are similar in the sense that they produce a single complicated and complex model. Random forest uses ensemble technique in which the basic idea is to generate lots of models by training on Training Set and at the end combine them through averaging the output. Averaging the Trees helps us reduce the variance and also improve the predictive performance on test set and eventually avoid overfitting. We would like to see if random forest algorithm will improve the model by reducing the square error.

We have a major concern in our model that the adjusted R2 is nearly perfect at 99%. Initially we suspected that the owner of the data set actually used the same method to arrive at the “Value”. However, that is not the real underlying cause. Thanks to professor who helped us identified the real cause – information leakage from the predictor variable “Release.Cause”. According to the definition, data leakage is the creation of unexpected additional information in the training data, allowing a model or machine learning algorithm to make unrealistically good predictions. In our case, we observe that there is a nearly perfect correlation between “Value” and “Release.Cause” from the correlation heatmap. And we have tried to use only “Release.Clause” to predict the “Value” and we get adjusted R2 at 98.36%, as shown in the regression result in the following figure. All these information confirm that there is a data leakage in our model. Hence, we would like to re-model it with OLS and LASSO by removing “Release.Clause” and check the model performance again. In the future we should be extra careful when we have a perfect correlation between a predictor and response variable. If that happens, it is better to leave this predictor out of the model. Another indicator could be a nearly perfect model accuracy or adjusted R2. It will always be a good practice to conduct a thorough research on why some of the predictor variables are highly correlated to the response variable.



**Reference**

AIC vs. BIC. (n.d.) PennSate The Methodology Center. Retrieved from https://www.methodology.psu.edu/resources/aic-vs-bic/

Akaike information criterion. (n.d.) Wikipedia. Retrieved from https://en.wikipedia.org/wiki/Akaike\_information\_criterion

David Lillis, Linear Models in R: Diagnosing Our Regression Model, retrieved from https://www.theanalysisfactor.com/linear-models-r-diagnosing-regression-model/

Martin, K. G. (n.d.). Assessing the Fit of Regression Models. The Analysis Factor. Retrieved from https://www.theanalysisfactor.com/assessing-the-fit-of-regression-models/

Shaikh, R. (2018). Cross Validation Explained: Evaluating estimator performance. Towards Data Science. Retrieved from https://towardsdatascience.com/cross-validation-explained-evaluating-estimator-performance-e51e5430ff85

YellowRoad. (2017). Beware of Data Leakage. Retrieved from https://blog.myyellowroad.com/beware-of-data-leakage-f6c307009ad9