

Project 3 158: Multiple Linear Regression

Tesfa Asmara and Kevin Loun

4/09/2022

Contents

1	Introduction	1
2	Hypothesis	1
3	Feature Engineering	2
4	Interaction Variables	2
5	Computational Model	3
6	Statistical Model	3
6.1	Checking for Outliers	4
6.2	Checking For Multicollinearity	11
	Bibliography	12

1 Introduction

The dataset for this project contains 10,000 League of Legends ranked matches from the North American region with 775 variables offered through the Riot Games API, provided on Kaggle (Games 2021a)(James 2020). Each match is pulled from players who rank Gold in the League system, a ranking system that matches players of a similar skill level to play with and against each other. Amongst North American players, the Gold skill level was the second most common tier, achieved by 27.7 percent of players, or approximately 49.86 million players when considered against Riot Games' player base of 180 million ("Distribution of League of Legends (Lol) Summoners in North America as of October 2021, by Tier" 2021)(Games 2021b). This dataset will be referred to as `lol10`.

For this project, the following variables are of interest: time spent crowd controlling others, map side, longest time spent living, kills, gold earned, and total damage dealt. A figure including all the relevant variables and their description is attached at the end.

2 Hypothesis

We consider the following research question: Are one or more of the independent variables, `b_gold_earned`, `b_kills`, `b_longest_time_spent_living`, or `b_time_c_cing_others` in the model useful in predicting the future values of `b_total_damage_dealt`?

3 Feature Engineering

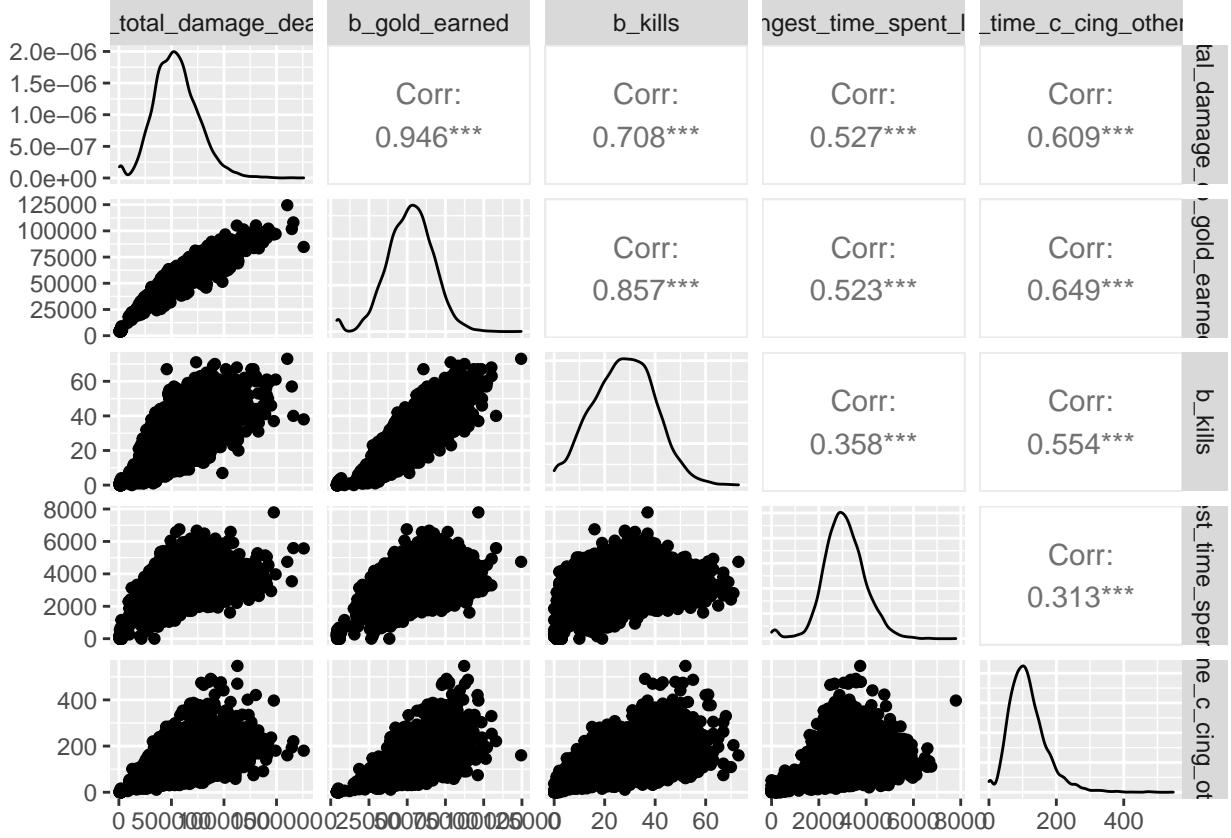


Figure 1: Correlation Matrix of Variables

For $n = 5$ observations, Table B.6 in ALSM is employed to assess whether or not the magnitude of the correlation coefficient supports the reasonableness of the normality assumption. The feature engineering we conducted was minimal.

4 Interaction Variables

We wish to test formally in the lol10 dataset whether interaction terms between the four explanatory variables should be included in the regression model. We therefore need to consider the following regression model: ,

$$\begin{aligned}
b_{\text{total_damage_dealt}} = & \beta_0 + \beta_1(b_{\text{gold_earned}}) + \\
& \beta_2(b_{\text{kills}}) + \beta_3(b_{\text{longest_time_spent_living}}) + \\
& \beta_4(b_{\text{time_c_cing_others}}) + \beta_5(b_{\text{gold_earned}} \times b_{\text{kills}}) + \\
& , \quad \beta_6(b_{\text{gold_earned}} \times b_{\text{longest_time_spent_living}}) + \beta_7(b_{\text{gold_earned}} \times b_{\text{time_c_cing_others}}) + \\
& \beta_8(b_{\text{kills}} \times b_{\text{longest_time_spent_living}}) + \beta_9(b_{\text{kills}} \times b_{\text{time_c_cing_others}}) + \\
& \beta_{10}(b_{\text{longest_time_spent_living}} \times b_{\text{time_c_cing_others}}) + \epsilon
\end{aligned} \tag{1}$$

We wish to test whether any interaction terms are needed. We do so by performing a partial F-test by fitting both the reduced and full models separately and thereafter comparing them using the `anova()` function.

Since $F \approx 297.3748455$ (p-value ≈ 0), we reject the null hypothesis $H_0 : \beta_5 = \beta_6 = \beta_7 = \beta_8 = \beta_9 = \beta_{10} = 0$ at the $\alpha = 0.05$ level of significance. This means that the interaction terms do not contribute significant information to the `b_total_damage_dealt` once the explanatory variables `b_gold_earned`, `b_kills`, `b_longest_time_spent_living`, `b_time_c_cing_others` have been taken into consideration.

5 Computational Model

From our domain experience, we consider the following parsimonious models: ,

$$\begin{aligned}
b_{\text{total_damage_dealt}} = & \beta_0 + \beta_1(b_{\text{gold_earned}}) + \\
& , \quad \beta_2(b_{\text{kills}}) + \epsilon
\end{aligned} \tag{2}$$

and ,

$$\begin{aligned}
b_{\text{total_damage_dealt}} = & \beta_0 + \beta_1(b_{\text{gold_earned}}) + \\
& , \quad \beta_2(b_{\text{kills}}) + \beta_3(b_{\text{longest_time_spent_living}}) +, \\
& \epsilon
\end{aligned} \tag{3}$$

We compare the two models using cross-validation prediction error.

In comparing which model is better, the CV RMSE provides information on how well the model did predicting each $1/v$, where $v =$ the number of folds, hold out sample. We can compare the model RMSE to the original variability seen in the `b_total_damage_dealt` variable. The original variability (measured by standard deviation) of `b_total_damage_dealt` was 2.1223789×10^5 . After running Model 1, the remaining variability (measured by RMSE averaged over the folds) is 2661.4238251; after running Model 2, the remaining variability (measured by RMSE averaged over the folds) is 2677.1891693.

Hence, the better computational model is ,

$$\begin{aligned}
b_{\text{total_damage_dealt}} = & \beta_0 + \beta_1(b_{\text{gold_earned}}) + \\
& , \quad \beta_2(b_{\text{kills}}) + \epsilon
\end{aligned} \tag{4}$$

6 Statistical Model

For the four predictors in the `lol10` data, we know there are $2^4 = 16$ possible models. The four-parameter model ,

$$\begin{aligned} b_{\text{total_damage_dealt}} = & \beta_0 + \beta_1(b_{\text{gold_earned}}) + \\ , & \beta_2(b_{\text{kills}}) + \beta_3(b_{\text{longest_time_spent_living}}) +, \\ & \beta_4(b_{\text{time_c_cing_others}}) + \epsilon \end{aligned} \quad (5)$$

is identified as best by the $R_{a,p}$ criterion; it has $\max(R_{a,p}) = 4$ and will serve as the selected model.

For the test data, the linear model we selected has $R^2 = 0.9321065$ and a $R^2_{adj} = 0.9319978$. Therefore, 0.0093211% of the variability in `b_total_damage_dealt` for players who rank Gold in the North American region is explained by the variables `b_gold_earned`, `b_kills`, `b_longest_time_spent_living`, `b_time_c_cing_others`.

6.1 Checking for Outliers

We first apply an outlier test on our data to see if there are any significant Bonferroni P values that could indicate potential outliers in our dataset.

```
##      rstudent unadjusted p-value Bonferroni p
## 1700  11.241725    1.2239e-28   3.0646e-25
## 2495   7.024868    2.7520e-12   6.8910e-09
## 1259   6.870322    8.0580e-12   2.0177e-08
## 1972   6.279619    3.9899e-10   9.9908e-07
## 413    5.727456    1.1417e-08   2.8588e-05
## 335    5.684848    1.4615e-08   3.6596e-05
## 1857   5.284712    1.3679e-07   3.4253e-04
## 1063   5.258432    1.5761e-07   3.9465e-04
## 1666   5.097518    3.6990e-07   9.2624e-04
## 1656   4.559145    5.3838e-06   1.3481e-02
```

The test output shows potential outliers at observations 1700, 2495, 1259, 1972, 413, 335, 1857, 1063, 1666, and 1656. While this test is important for identifying a potentially significant outlying observation, it is not the ultimate indicator of outliers and checking for patterns in our outliers so we can proceed to generate a plot of Dfbetas to see if these points reappear as outliers.

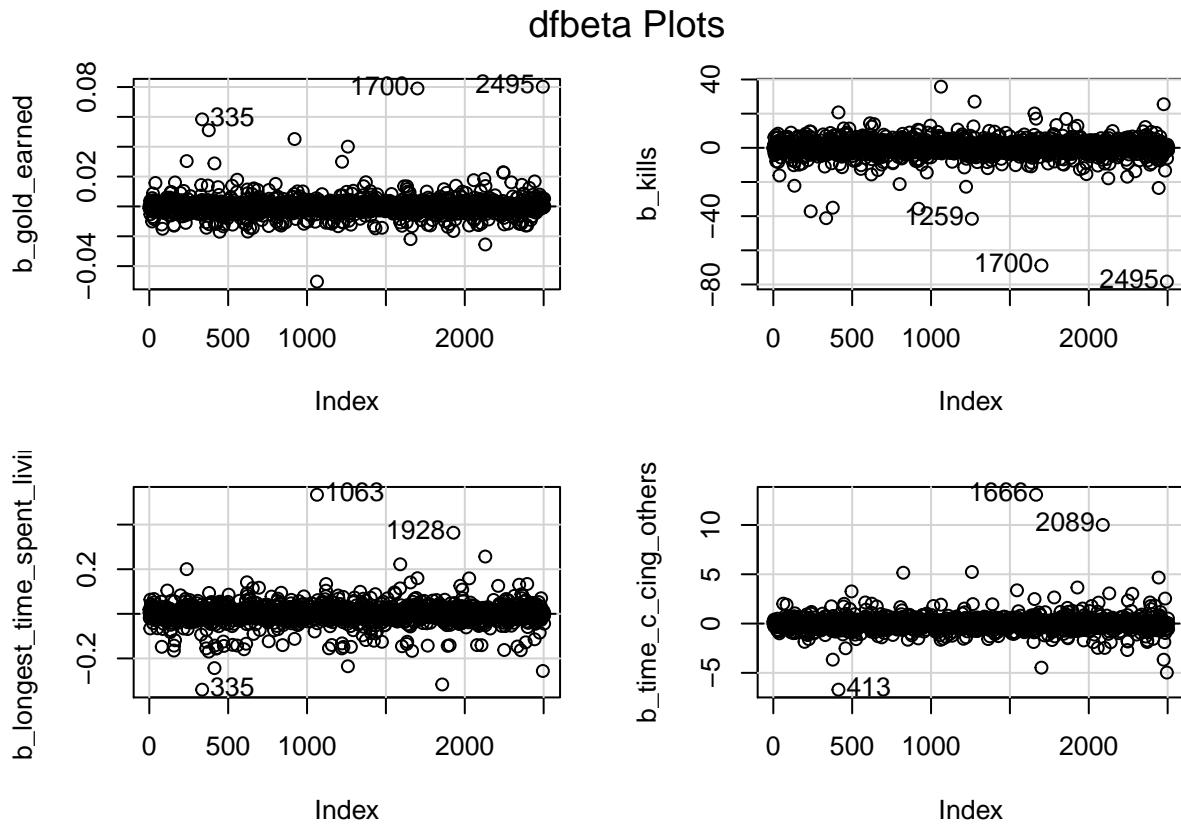


Figure 2: Plots for DfBetas

```
## NULL
```

We notice that the same outlying points appear on the dfbeta plots further indicating that these are outliers that could affect our model. In the leverage plot for each predictor we see a similar outcome; outliers at observation 1700 and 2495 for each predictor variable.

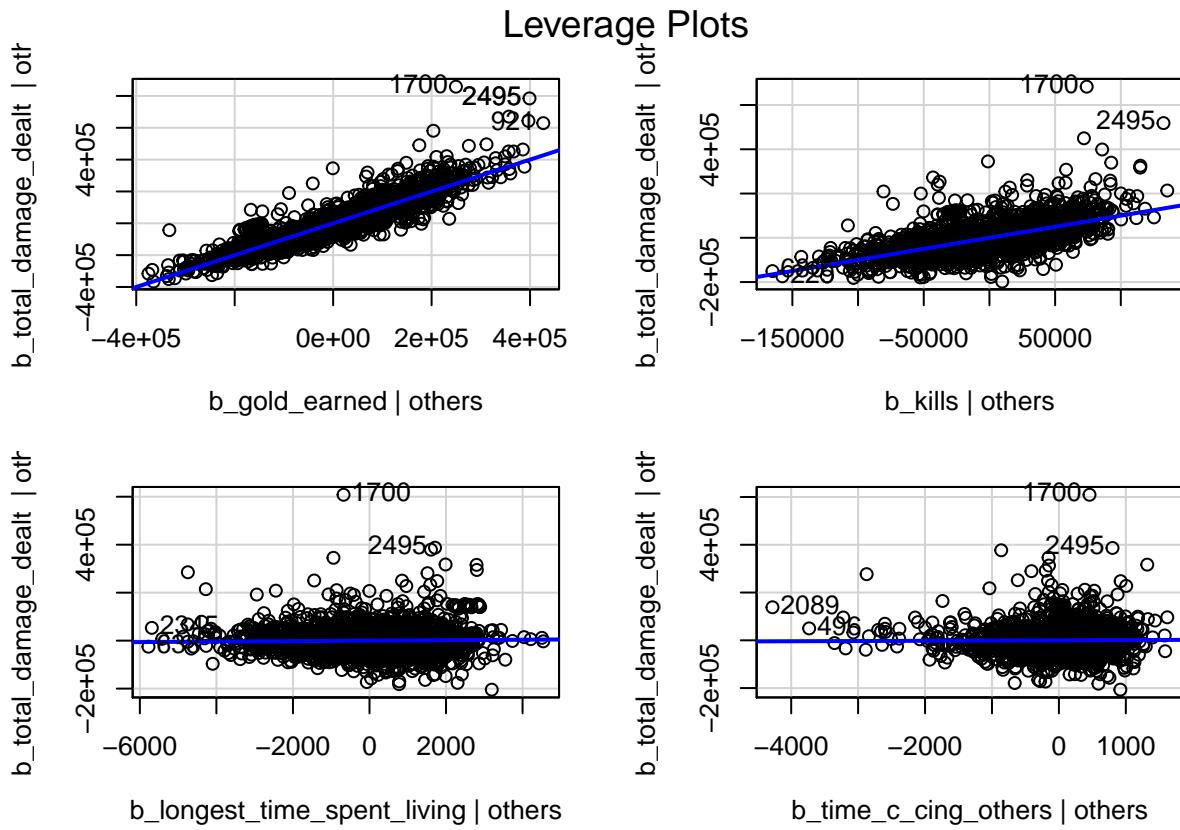


Figure 3: Leverage Plots for Outliers

To ensure that the outliers aren't an underlying cause of high correlation between our variables, we plotted our data in an added variable plot to visualize our outliers in plots where our predictors are independent of one another.

Added-Variable Plots

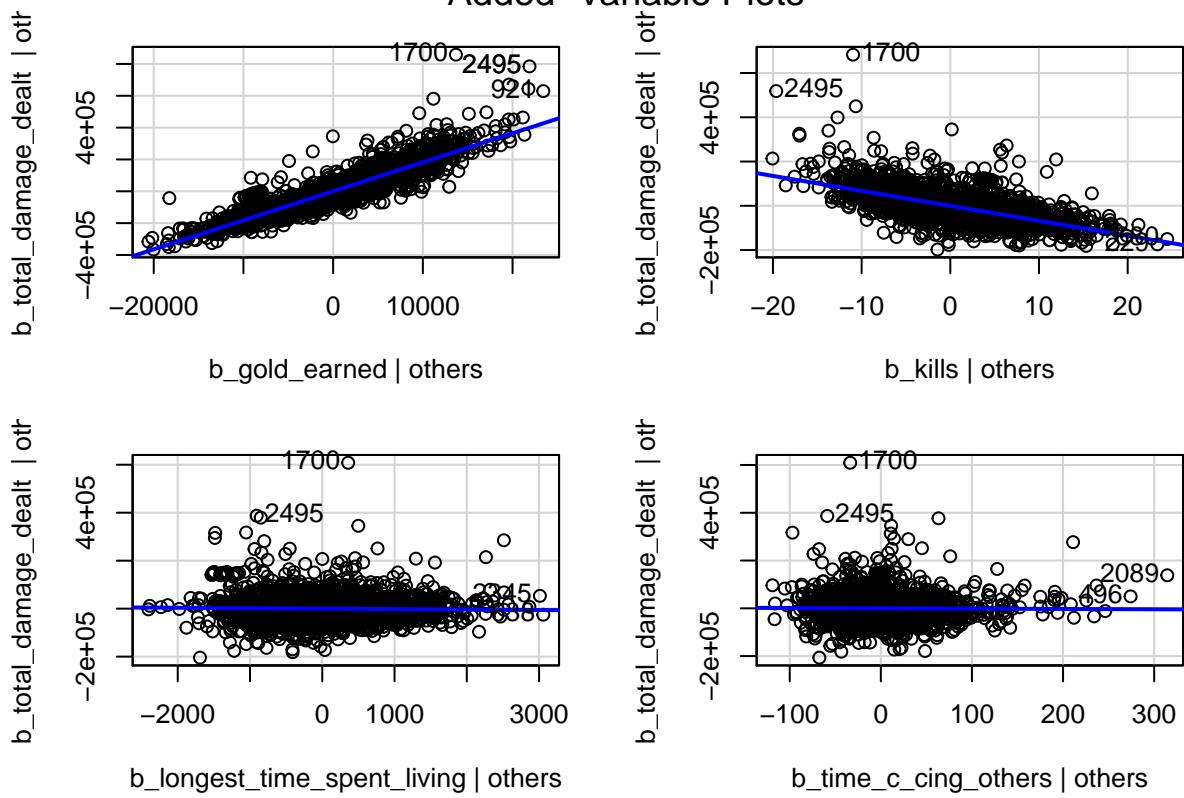


Figure 4: Added Variable Plot for each Predictor

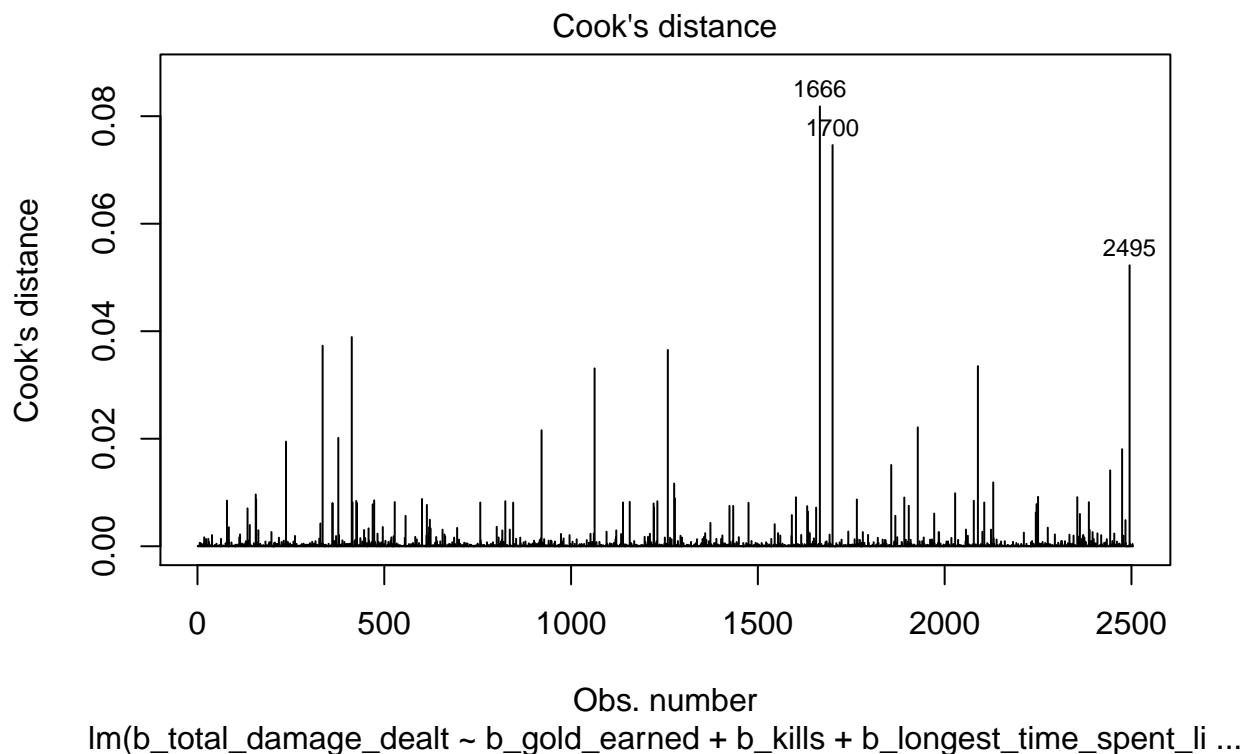


Figure 5: Cook's Distance per Observation

Influence Plot

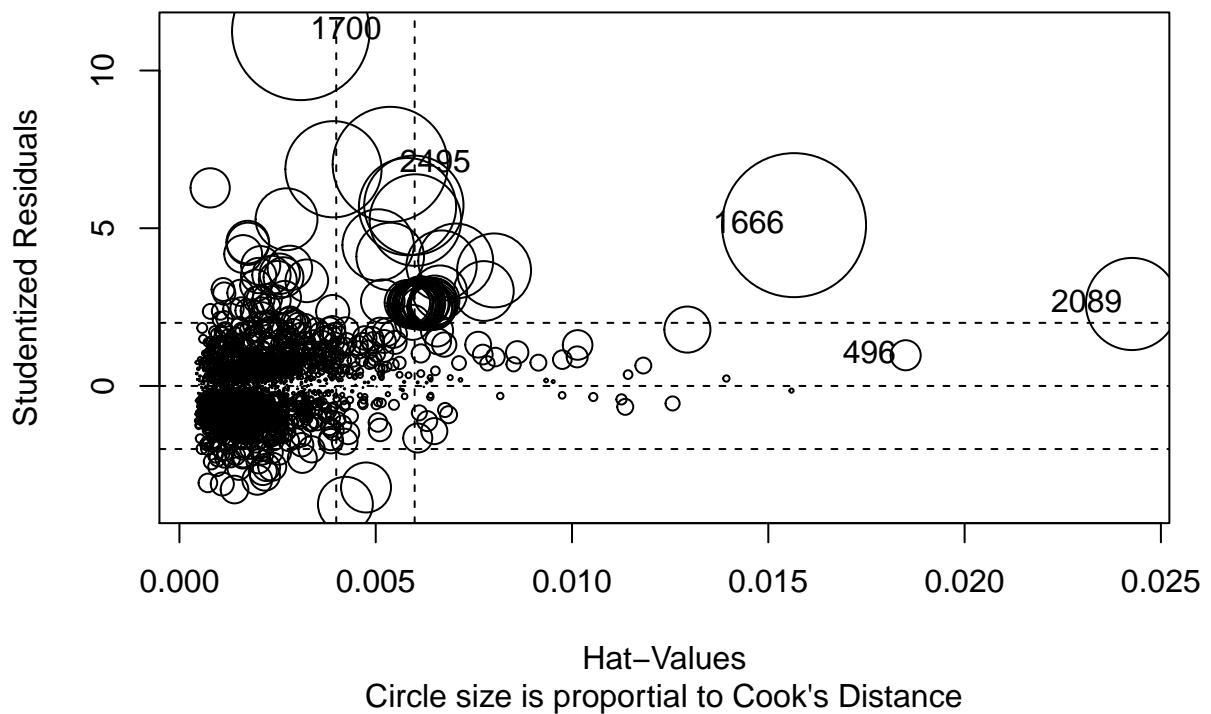


Figure 6: Influence plot for Cook's Distance

```
##          StudRes      Hat      CookD
## 496    0.9728662 0.018498547 0.003567732
## 1666   5.0975175 0.015654895 0.081833242
## 1700  11.2417253 0.003090876 0.074621180
## 2089   2.5985059 0.024262284 0.033502519
## 2495   7.0248676 0.005368642 0.052262053
```

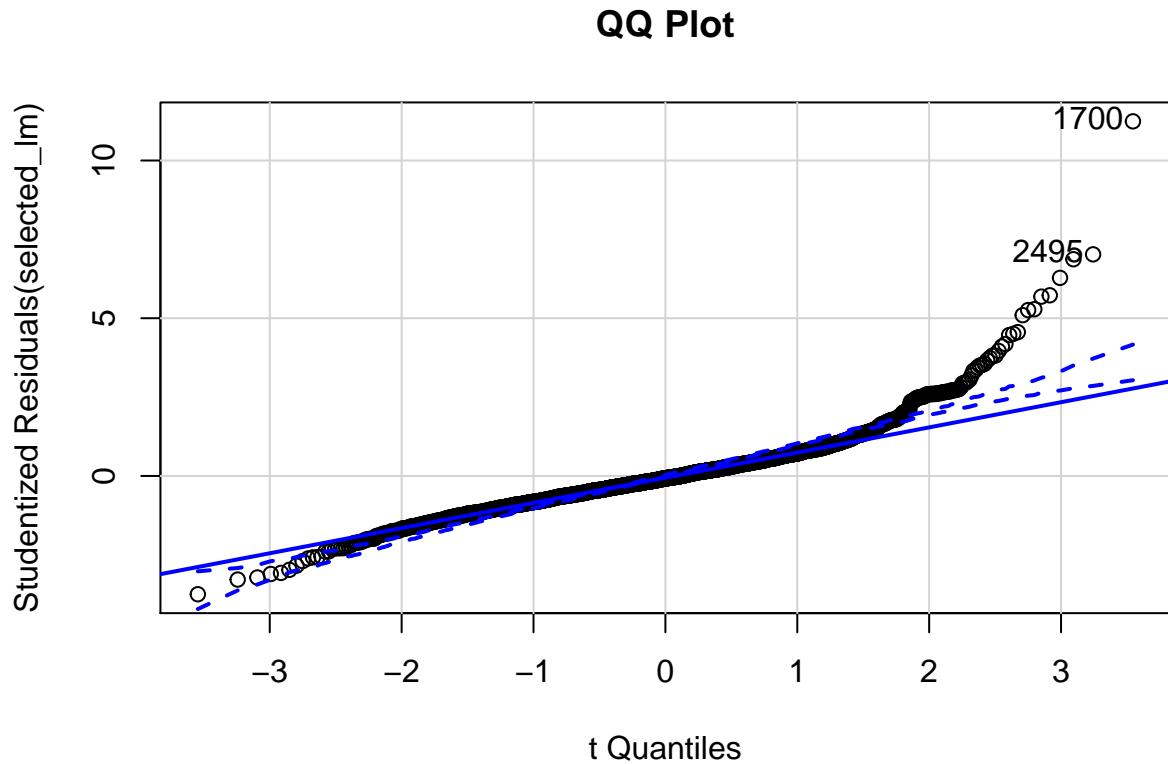


Figure 7: QQ Plot and Distribution of Studentized Residuals

```
## [1] 1700 2495
```

Distribution of Studentized Residuals

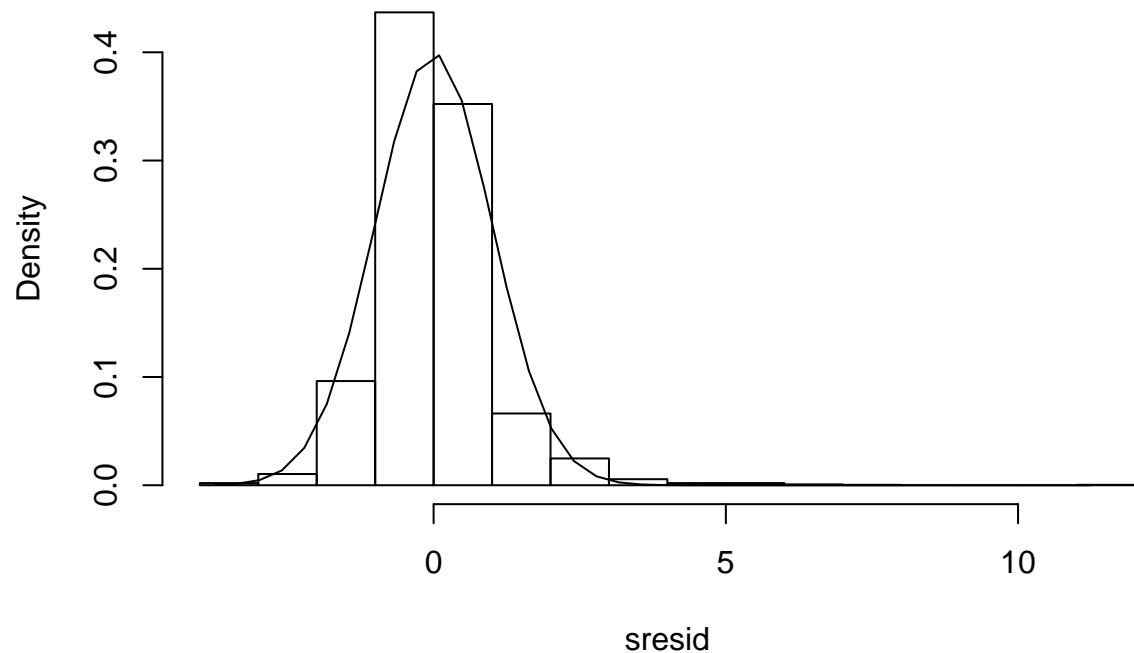


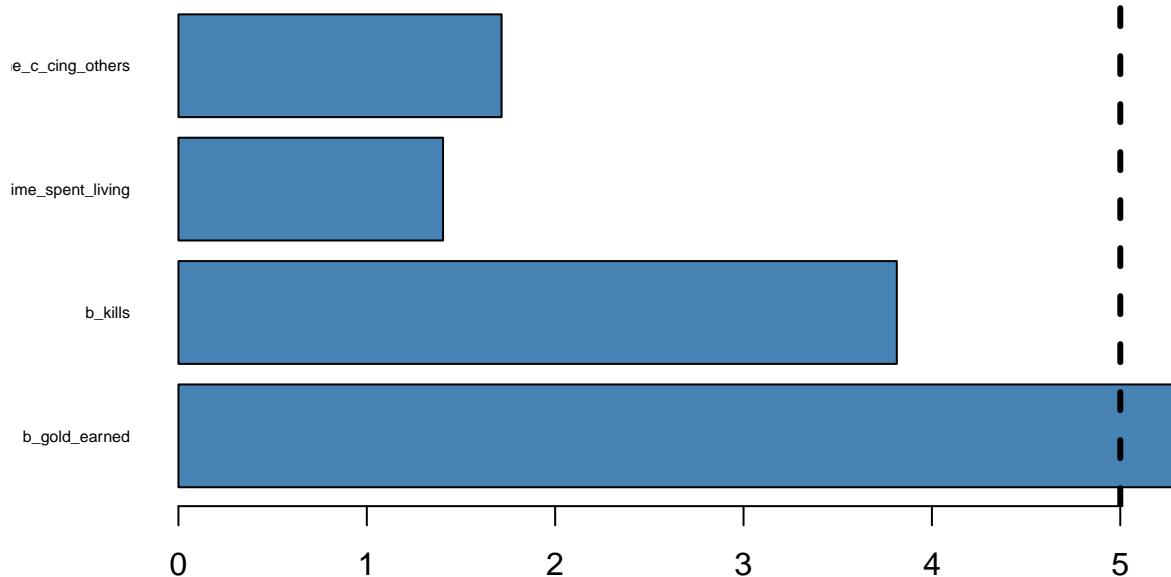
Figure 8: QQ Plot and Distribution of Studentized Residuals

6.2 Checking For Multicollinearity

```
# Evaluate Collinearity
vif(selected_lm) # variance inflation factors

##          b_gold_earned            b_kills
##      5.308366      3.813736
##  b_longest_time_spent_living    b_time_c_cing_others
##      1.404775      1.715489
```

VIF Values



- Interpret your β coefficients to the best of your ability.
- Report the R^2 and Adjusted- R^2 values on the test data.
- A complete analysis of the residuals and influence points.
- Try to give an interpretation of the model that makes sense.
- Give CIs for a mean predicted value and a future predicted value for at least one combination of X's (from your final linear model).
- Summarize your report.
- Add pretty language.
- Fix Variables and their descriptions table for MLR./
- Report on any relationships between the explanatory variables.

Bibliography

- “Distribution of League of Legends (Lol) Summoners in North America as of October 2021, by Tier.” 2021. *Statista*. <https://www.statista.com/statistics/807298/league-of-legends-player-tier/>.
- Games, Riot. 2021a. “Riot Games Api.” *Riot Developer Portal*. <https://developer.riotgames.com/apis>.
- . 2021b. *Twitter*. Twitter. <https://twitter.com/riotgames/status/1455172784938651649?s=20&t=AQmQGrTa1ijf6u3cEDPZcg>.

James. 2020. “League of Legends Ranked Match Data from Na.” *Kaggle*. <https://www.kaggle.com/jamesbting/league-of-legends-ranked-match-data-from-na>.

Variable	Description
b_kills	The number of kills obtained by summoners on the blue side of the map.
b_gold_earned	The gold obtained by summoner 1 on the blue side of the map.
b_longest_time_spent_living	Sum of the longest time spent alive by summoners on the blue side of the map.
b_time_c_cing_others	The total time spend crowd controlling enemy players by summoners of the blue side of the map.
b_total_damage_dealt	Total damage done by summoners on the blue side of the map.

Figure 9: Variables and their descriptions