League of Legends: Detecting Smurfs and Analyzing Rank vs Total Games Played

David Pham

June 28, 2022

I. Overview

League of Legends is one of the most popular online games ever made. As of this report, there are around 115 million active monthly players worldwide and roughly 3 million matches played daily. There is no doubt that this highly addictive game has made its way into history.

In this report, we will be examining if there is a correlation between your League of Legends rank and the amount of games you've played, won, lost, as well as your Summoner Level. Furthermore, we will see if this is enough information to detect smurfs, players that create new accounts and promote unbalanced matchmaking in games/ranks that they do not belong in.

Explanation of Ranked

Assuming the reader has a good grasp on what the game's objective is about, we will briefly go over the competitive game mode, the Ranked queue. Whenever you finish a game on any gamemode in League of Legends (with the exception of custom matches), you get **experience** (also abbreviated as **XP**) which contributes to your **Summoner Level**. This level is an indicator on roughly how much time you have spent in the game. When you are level 30, you will be eligible to play in the Ranked game mode. As an extremely rough estimate, many players have reported that it takes around 200-300 games to obtain 30 levels.

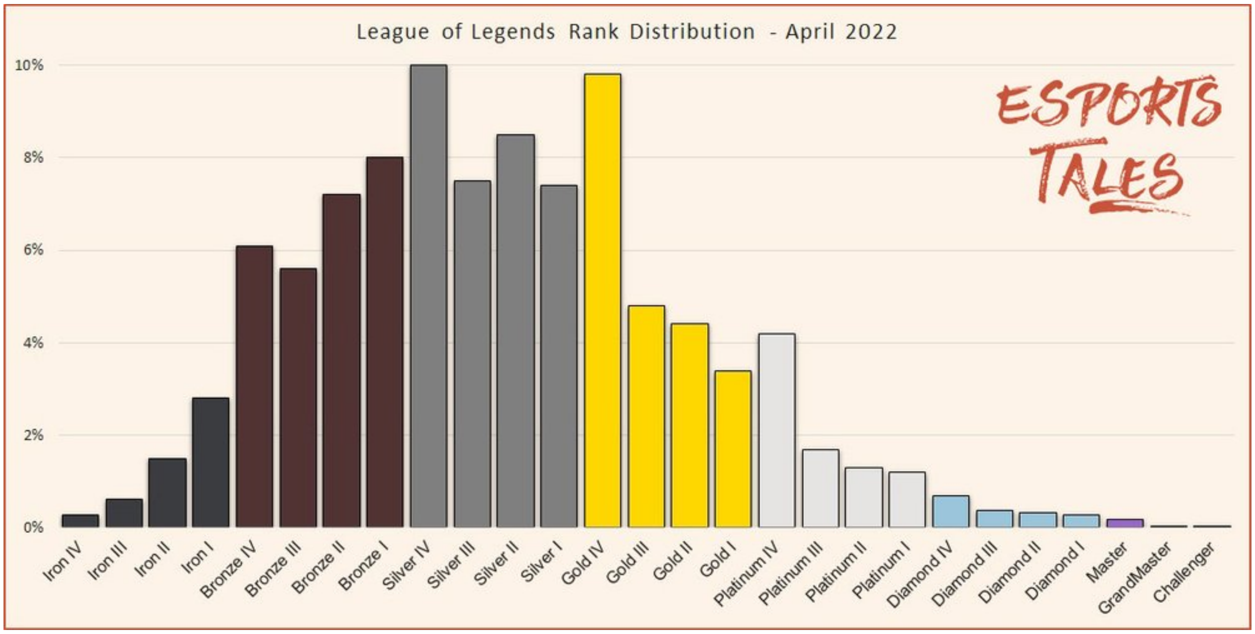
When you start out, you will be **unranked**. The first 10 ranked games you play are served to measure what your approximate skill level is. The better you and your team perform and the faster you win, the more likely you are to get a higher rank.

The Ranked Distribution There are a total of 9 **tiers** in the ranking system. From lowest to highest, these are:

- Iron
- Bronze
- Silver
- Gold
- Platinum
- Diamond
- Master
- Grandmaster
- Challenger

Furthermore, with the exception of Master, Grandmaster, and Challenger, each tier has four **divisions** that you can get placed in (e.g., Bronze **I**). The fourth (IV) division is the lowest, while the first (I) division is the highest. When you win enough games in the first division, you'll get promoted to the next tier. Contrarily, you get demoted to the previous tier if you lose enough matches in the bottom division.

According to *EsportsTales*, this is what the worldwide ranked distribution looks like (as of April 2022):



A good chunk of players lie in between Silver IV and Gold IV; in fact, around 43.2% of the entire player base lays here. Aside from this, the rank distribution is positively skewed (right-skewed).

Smurfs Quickly discussing this section, a 'smurf' is a term given to people who create new accounts and disguise themselves in lower skill brackets. This is extremely common in many competitive video games, and promotes unfair matchmaking. Riot has done a better job taking care of this issue over the years, but smurfing is still prevalent no matter where you go. I was curious to see if raw data was enough to detect smurfs. We will look at factors such as win ratio, summoner level, and other indicators.

II. Data and Wrangling

A user from the Riot Games API Discord community, Canisback, randomly sampled 18347 ranked players from the North American League of Legends servers. From their .csv file, they managed to obtain the actual **leagues** of the players (e.g., Galio's Warlords, Nocturne's Dervish, etc.), a bracket containing a group of players in that tier, in the form of a `league_id`.

In Python, we randomly sample 1000 players from Canisback's sample by converting the .csv file into a Pandas dataframe, shuffling it, and then iterating through the first 1000 randomized entries. The maximum and most efficient number of requests that could be accessed from the Riot API by using this method is around 1000 (1000 requests took approximately 100 minutes, on average on my computer). Tying this in with reducing our margin of error to approximately 3%, 1000 samples should be an ideal sample size.

Note that the information provided are from *leagues*, and not actual *players*. Each league is a dictionary containing a few key elements that we are interested in:

- The `tier` of the league (Iron, all the way up to Challenger).
- A paginated list containing all the players in that current league (in no particular order).

Therefore, we randomly select the first player from each league.

For every element in this list, it is a dictionary containing information about an individual player's stats. Some notable details we extract are:

- The unique summonerId that is associated with every League of Legends account.
- The rank/division of this summoner (Again, 4 is the lowest, 1 is the highest).
- The number of wins this summoner has in the ranked gamemode.
- The number of losses this summoner has in ranked gamemode.
- The total number of games played this summoner has in ranked (wins + losses).

Finally, using the summonerId key, we obtain a few more indicators that are not directly related to rank:

- The summoner level of a player. (Remember, this is a good indicator on how long someone has played League of Legends)
- The **total** number of mastery points a player has on their account. You accumulate mastery points on characters by simply playing them. Generally, someone with a large amount of total mastery points has played the game quite often.
- The highest mastery points of a player on a single character. If this number is large, then that means this player has spent a lot of time playing this one character.
- A player's total mastery score. Another progression metric similar to the mastery points. However, a player's performance is also included.
- The highest mastery point proportion. This is the percentage of the player's highest mastery point character relative to their total mastery points.
- The number of champions played. A League of Legends character is referred to as a **champion**.

All of these factors may help us determine if the number of champions you play and how often you play them may have an impact on your performance/rank.

Here is a summary of the file we will be using:

```
## 'data.frame':    1000 obs. of  11 variables:
##  $ tier                 : Factor w/ 6 levels "IRON","BRONZE",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ rank                 : Factor w/ 4 levels "IV","III","II",..: 4 1 1 2 2 1 4 2 1 1 ...
##  $ wins                 : int  15 11 17 16 168 3 289 161 53 9 ...
##  $ losses               : int  15 11 12 26 180 10 311 184 49 3 ...
##  $ total_games_played   : int  30 22 29 42 348 13 600 345 102 12 ...
##  $ summoner_level       : int  423 44 38 74 179 40 136 89 232 59 ...
##  $ total_mastery_points  : int  3431274 26894 155118 350568 1019680 121276 789564 948472 1876645 2489
##  $ highest_mastery_points: int  153810 13905 30208 252485 189576 32040 194069 43544 314862 100573 ..
##  $ total_mastery_score   : int  572 18 85 42 197 57 96 289 244 74 ...
##  $ number_champs_played  : int  158 12 55 19 96 36 50 116 104 54 ...
##  $ hmp_proportion        : num  0.0448 0.517 0.1947 0.7202 0.1859 ...
```

Upon glancing the data to look for unusual observations, I remove one case where the summoner level is less than 30. Recall, you must be level 30 or above to be eligible to play Ranked. We now have 999 observations.

Next, we create four new variables:

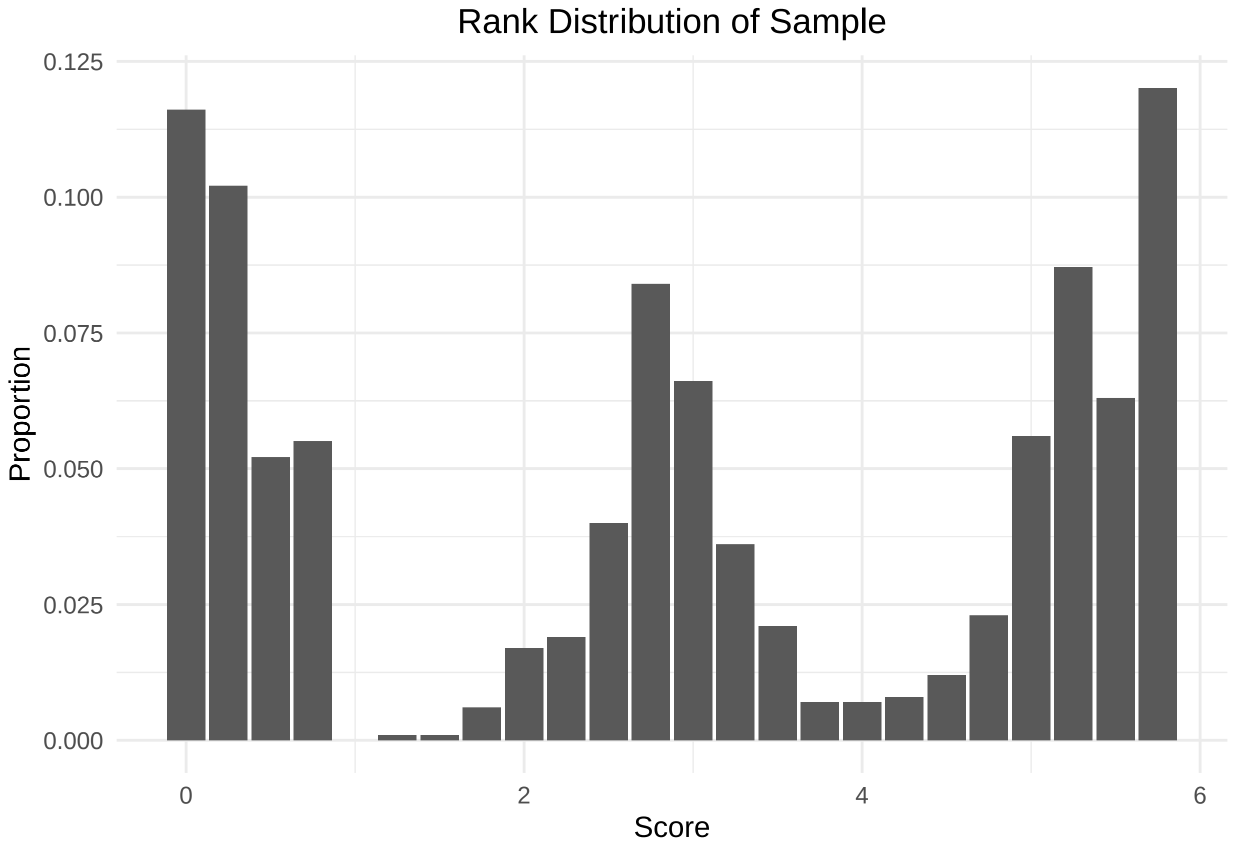
- A win_ratio variable, which gives us the percentage of the amount of games players have won out of their total games played.

- A `tier_score` variable, which gives a rating from 0-5, depending on what tier you are (Master and Challenger have been omitted since Canisback's data did not contain any players from these tiers). For reference:
    - Iron = 0
    - Bronze = 1
    - Silver = 2
    - Gold = 3
    - Platinum = 4
    - Diamond = 5
- A `rank_score` variable, which gives a rating from 0-0.75, depending on what rank you are. For reference:
    - IV = 0
    - III = 0.25
    - II = 0.5
    - I = 0.75
- Finally, the `score` variable adds each player's `tier_score` and `rank_score` into one variable, making it a simpler response variable to predict instead of tier and rank being separate entities. As an example, Silver III would result in a score of 2 + 0.25 = 2.25.

Here is a summary of the new, whole dataset:

```
## 'data.frame':    999 obs. of  15 variables:
##  $ tier                  : Factor w/ 6 levels "IRON","BRONZE",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ rank                  : Factor w/ 4 levels "IV","III","II",..: 4 1 1 2 2 1 4 2 1 1 ...
##  $ wins                  : int  15 11 17 16 168 3 289 161 53 9 ...
##  $ losses                : int  15 11 12 26 180 10 311 184 49 3 ...
##  $ total_games_played    : int  30 22 29 42 348 13 600 345 102 12 ...
##  $ summoner_level        : int  423 44 38 74 179 40 136 89 232 59 ...
##  $ total_mastery_points  : int  3431274 26894 155118 350568 1019680 121276 789564 948472 1876645 2489
##  $ highest_mastery_points: int  153810 13905 30208 252485 189576 32040 194069 43544 314862 100573 ..
##  $ total_mastery_score   : int  572 18 85 42 197 57 96 289 244 74 ...
##  $ number_champs_played  : int  158 12 55 19 96 36 50 116 104 54 ...
##  $ hmp_proportion        : num  0.0448 0.517 0.1947 0.7202 0.1859 ...
##  $ win_ratio             : num  0.5 0.5 0.586 0.381 0.483 ...
##  $ tier_score            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ rank_score            : num  0.75 0 0 0.25 0.25 0 0.75 0.25 0 0 ...
##  $ score                 : num  0.75 0 0 0.25 0.25 0 0.75 0.25 0 0 ...
```

One problem we encounter is that the sample we had collected deviates fairly from the true ranked distribution:

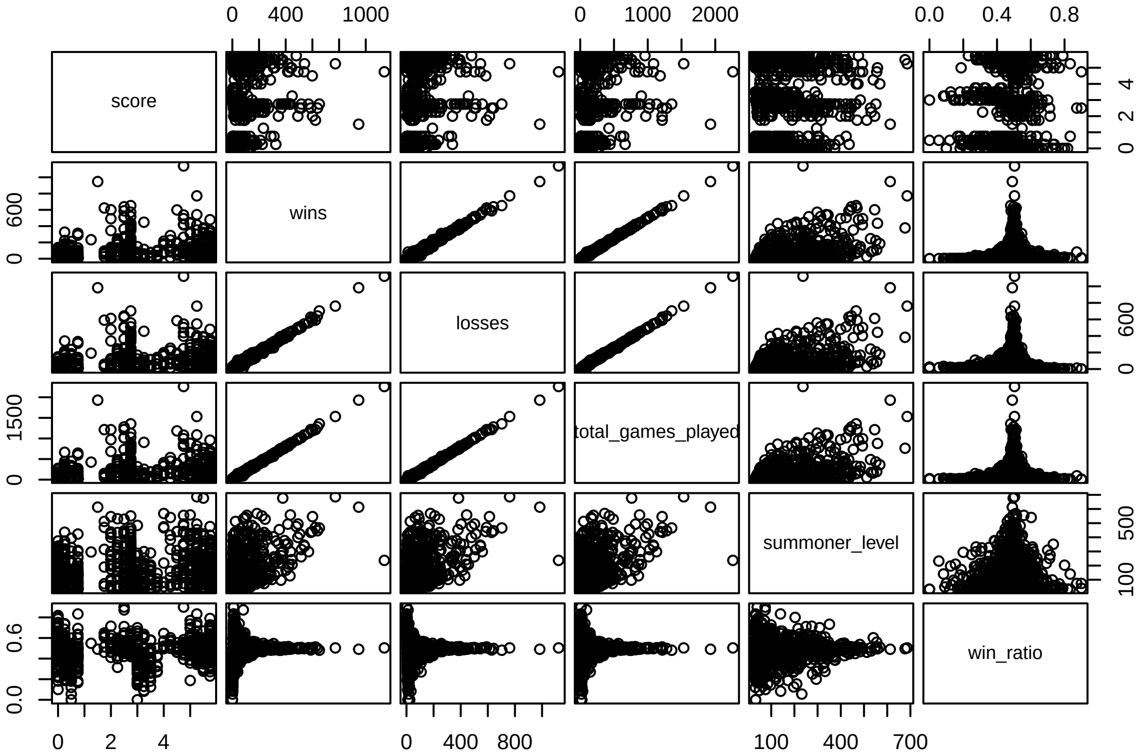


Unluckily, our sample has more of an emphasis on the two opposite spectrums of the ranks; Iron and Diamond by themselves make up over 50% of the ranked proportion in this sample. By design, this report may generate inaccurate findings and conclusions.

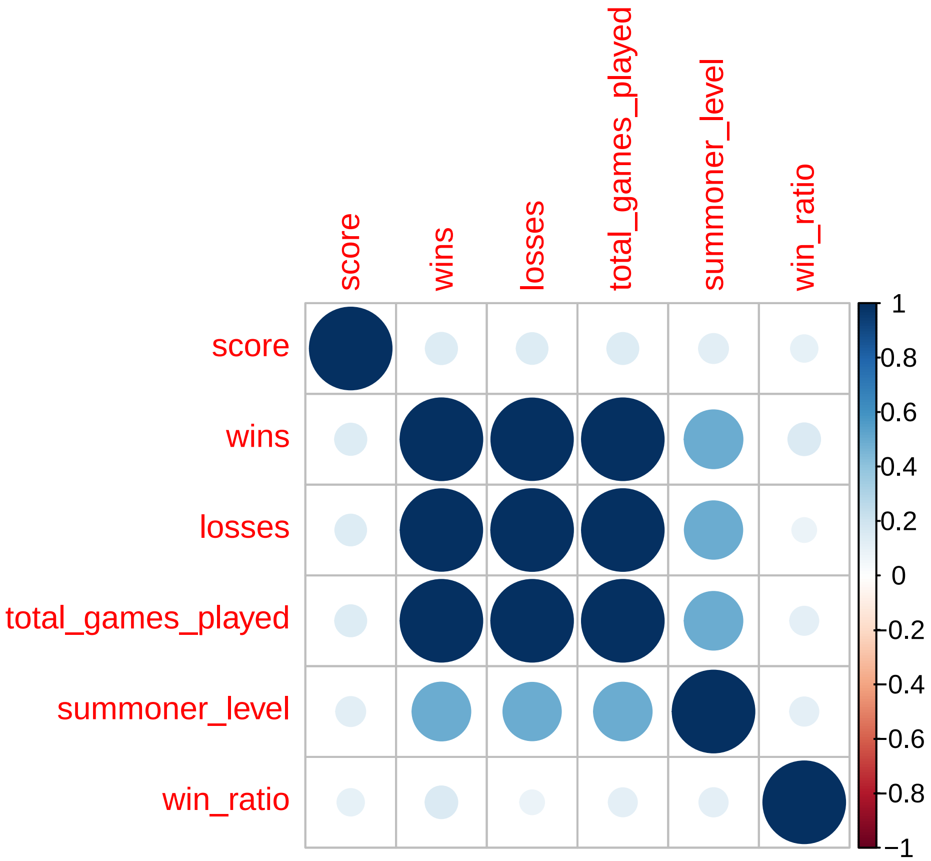
III. Explanatory Data Analysis

Listing the variables by category, `tier` and `rank` are both ordinal variables, while `wins`, `losses`, `total games played`, `summoner level`, `total mastery points`, `highest mastery points`, `total mastery score`, `number of champions played`, `tier score`, `rank score` and `score` are discrete variables. The only continuous variables are `win ratio` and `highest mastery score proportion`.

The main variables we will be looking to analyze are "wins", "losses", "total_games_played", "summoner_level" and "win_ratio" as the independent variables, and "score" as the response variable. Here are all the pairwise correlations and the scatterplot matrix for these pairs of variables in the data:



Most of these are plots are certainly redundant, so we will overlook most of them. Looking at the first row of plots however, it seems as if we graphed gibberish; there is certainly no clear linear relationship between the “score” variable and any single predictor (while holding all others constant). Furthermore, there is no way of determining outliers that may look like smurfs due to everything being so scattered.



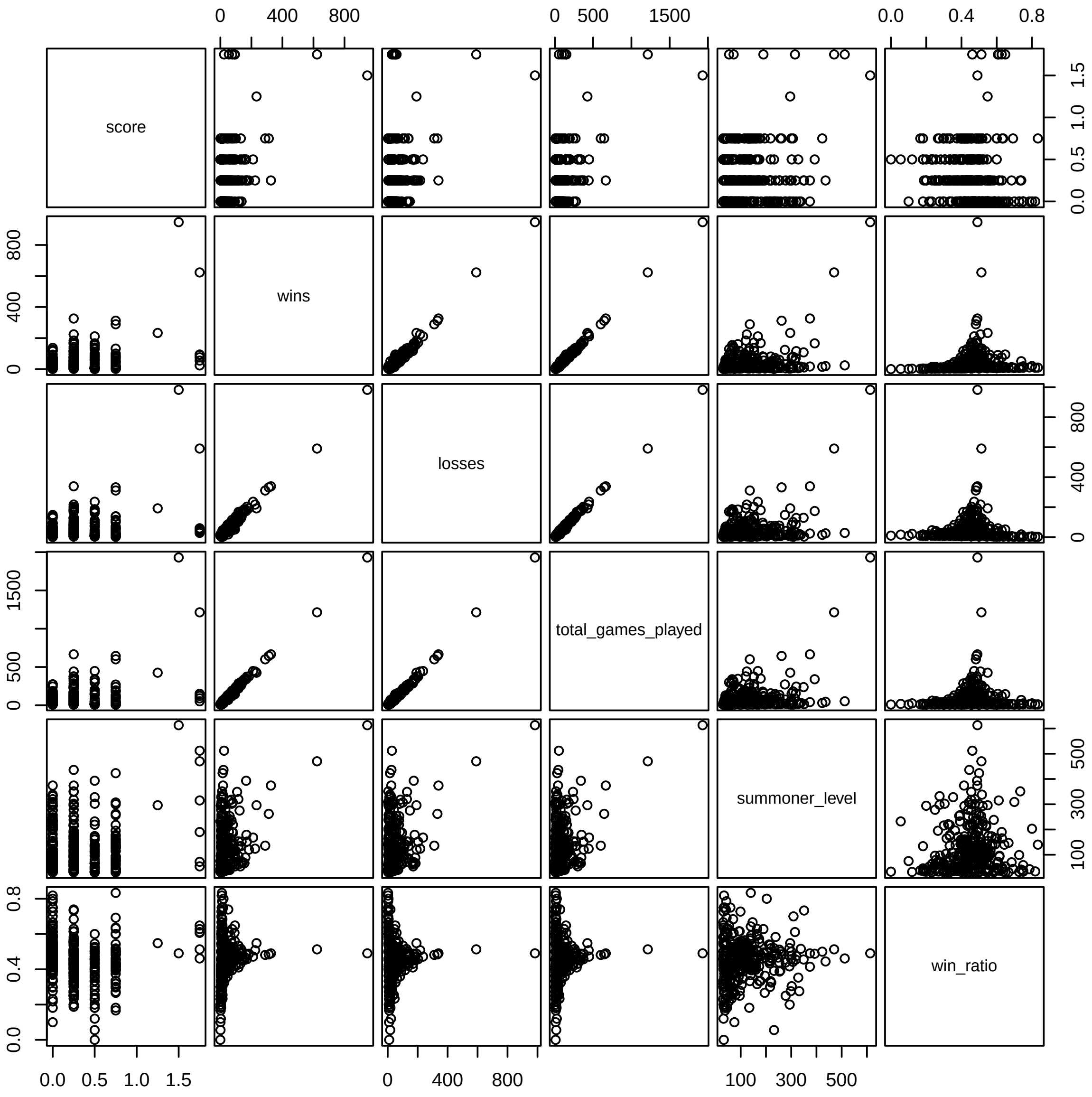
The pairwise correlations confirm this, as the coefficients between 'score' and each predictor are very low (refer to the first column).

Furthermore, an interesting remark is that there is almost a one-to-one correspondance with the amount of wins and losses/total games played a player has. Perhaps there is no easy correlation with your rank after all… You win as much as you lose. This is actually intended by the matchmaking system in League of Legends and will be discussed at a later stage.

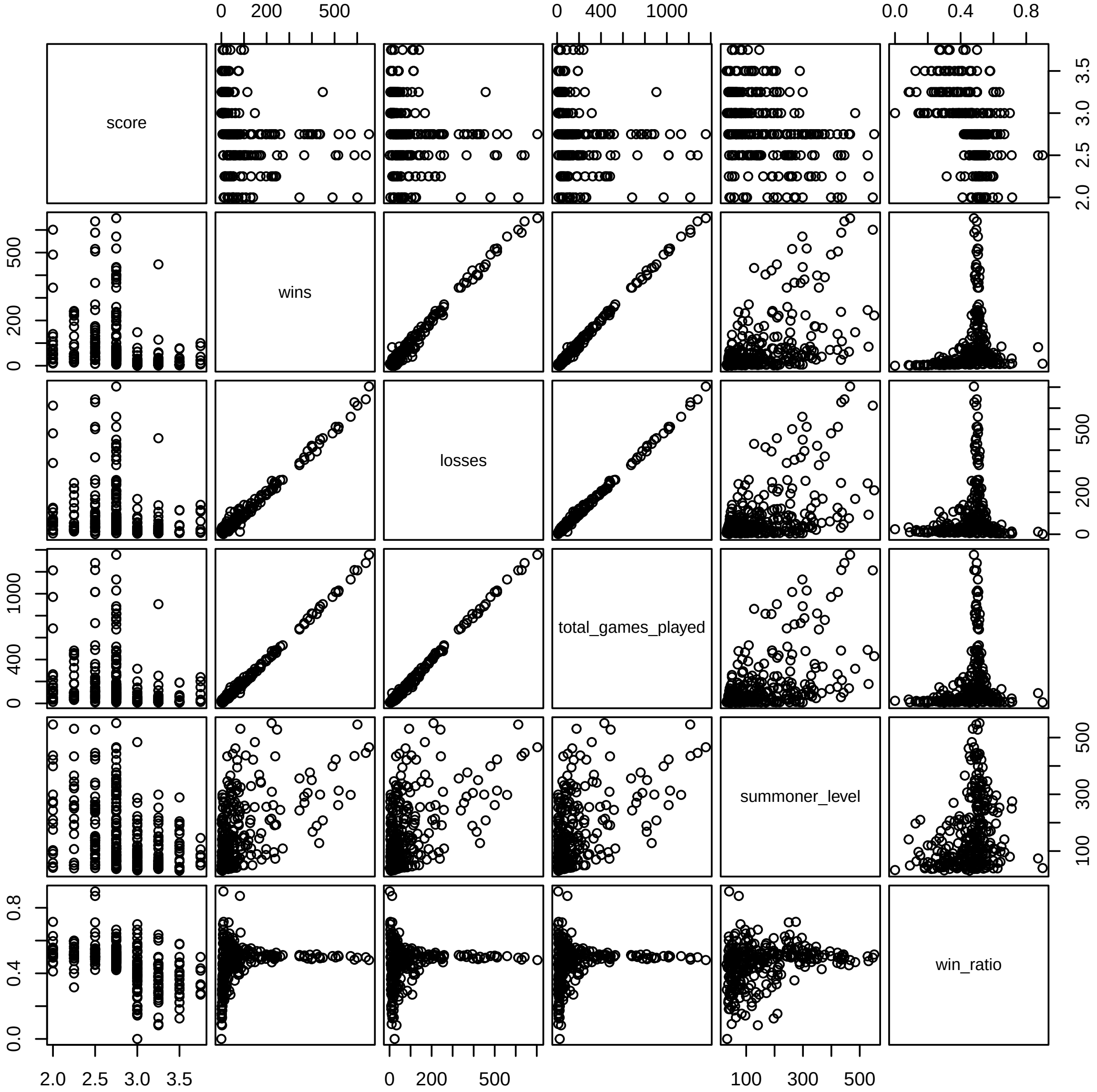
Also, we know that there is multicollinearity here; wins, losses, and total games played are a perfect example of variables being heavily correlated with each other, so we will omit wins and losses in the MLR section of the report, and focus on total games played, win ratio and summoner level instead.

Another thing to note is that it looks like the response for each graph in the top row is divided/clustered into three major sections: players with a 'score' value from 0-2, 2-4, and 4-6 seem to have diferent trends. We'll try to stratify the data into three subsets and see if we can zoom in on a possible relationship.

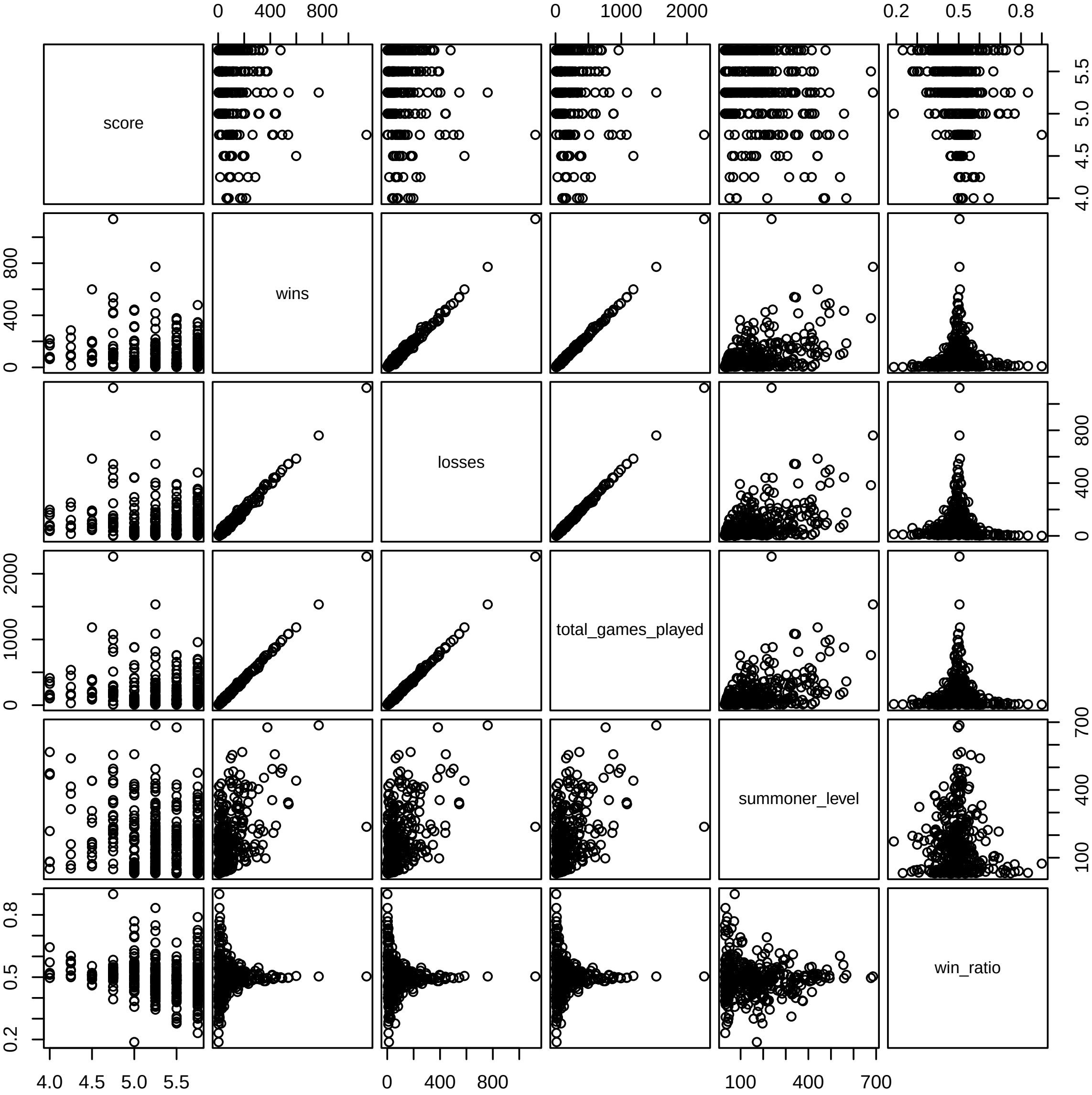
Here is the scatterplot matrix for players only in Iron and Bronze:



This is the scatterplot matrix for players in Silver and Gold:



Finally, the scatterplot matrix for players in Platinum and Diamond:



Unfortunately… it doesn't look like there's anything special here either. Focusing in on the top row of the scatterplot matrices, assuming we hold all other predictors constant, the data is scrambled everywhere and there does not seem to be a linear relationship between any of these predictors and what rank you are.

However, in the last scatterplot matrix for Platinum and Diamond players, there is a fairly high probability that there are smurfs in this subset. To elaborate, there is an extremely low chance that anyone below summoner level 100 would already be in the top ~10% of players. Furthermore, a win rate above 60% in this bracket (or in general) is considered phenomenal, so there may be evidence of Master/Challenger players in this dataset!

Using this filter, this is what some of these smurfs look like:

```
## Rows: 24
## Columns: 15
## $ tier                    <fct> PLATINUM, PLATINUM, DIAMOND, DIAMOND, DIAMOND, ~
## $ rank                    <fct> IV, I, III, IV, IV, I, IV, I, IV, I, III, I, IV~
## $ wins                    <int> 67, 9, 13, 14, 36, 8, 6, 15, 6, 10, 14, 8, 10, ~
## $ losses                  <int> 37, 1, 5, 6, 16, 4, 4, 4, 4, 4, 9, 3, 3, 4, 6, ~
## $ total_games_played      <int> 104, 10, 18, 20, 52, 12, 10, 19, 10, 14, 23, 11~
## $ summoner_level          <int> 51, 74, 32, 33, 99, 51, 52, 41, 78, 47, 68, 41,~
## $ total_mastery_points    <int> 203587, 391712, 14153, 22512, 904456, 259550, 2~
## $ highest_mastery_points <int> 53340, 73763, 11223, 9419, 89613, 23150, 26348,~
## $ total_mastery_score     <int> 103, 129, 6, 12, 267, 148, 107, 11, 186, 107, 1~
## $ number_champs_played    <int> 76, 67, 3, 7, 122, 97, 60, 7, 109, 62, 92, 63, ~
## $ hmp_proportion          <dbl> 0.26200101, 0.18830927, 0.79297675, 0.41839908,~
## $ win_ratio               <dbl> 0.6442308, 0.9000000, 0.7222222, 0.7000000, 0.6~
## $ tier_score              <dbl> 4, 4, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,~
## $ rank_score              <dbl> 0.00, 0.75, 0.25, 0.00, 0.00, 0.75, 0.00, 0.75,~
## $ score                   <dbl> 4.00, 4.75, 5.25, 5.00, 5.00, 5.75, 5.00, 5.75,~
```

A genuine level 31 player who is already in the top 0.3% of players in North America… Quite hard to believe.

Next, let's take a look at the tables depicting the strength of the relationships for each predictor and the response, by tier classification.

```
##      Predictor            Cor. Coeff. for Score (in Iron and Bronze)
## [1,] "wins"               "0.3162"
## [2,] "losses"             "0.3094"
## [3,] "total_games_played" "0.3133"
## [4,] "summoner_level"     "0.4639"
## [5,] "win_ratio"          "-0.1787"

##      Predictor            Cor. Coeff. for Score (in Silver and Gold)
## [1,] "wins"               "-0.368"
## [2,] "losses"             "-0.3241"
## [3,] "total_games_played" "-0.3468"
## [4,] "summoner_level"     "-0.3134"
## [5,] "win_ratio"          "-0.4839"

##      Predictor            Cor. Coeff. for Score (in Platinum and Diamond)
## [1,] "wins"               "-0.1862"
## [2,] "losses"             "-0.143"
## [3,] "total_games_played" "-0.1647"
## [4,] "summoner_level"     "-0.2224"
## [5,] "win_ratio"          "-0.0683"
```

We even have *negative* correlations with your League of Legends rank, which doesn't even make much sense. The strongest, positive linear relationship with 'score' (again, by holding all predictors constant) is summoner level in the Iron and Bronze tiers. Even at best, the relationship is moderately positive. Why is that not the case in other tiers?

IV. Methods and Models?

We have run into a stump. That is, none of the SLR model assumptions hold for the response and any of the individual predictors.

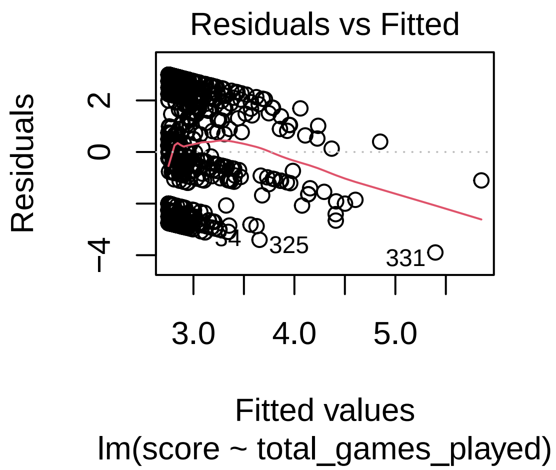
Recall the four normal error SLR model assumptions (LINE!):

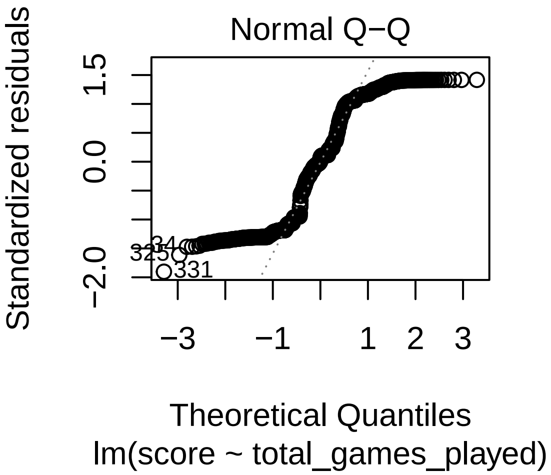
1. Linearity: The mean of the error at each value of the predictor is **zero**.
2. Independence: The errors are independent.
3. Normally Distributed: The errors at each value of the predictor are *normally distributed*.
4. Equal Variances: These errors have equal variances at each value of the predictor.

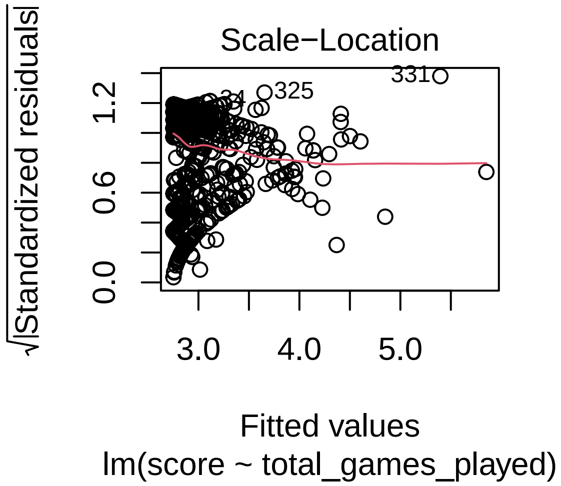
We'll take a look at the four diagnostic plots between the response and each individual predictor to confirm our initial statement.

First, for total games played and score,

- Linearity is violated (decreasing trend in the residuals vs fitted plot)
- Normality assumption is violated (tails are too skewed in Normal Q-Q plot)
- Constant variances assumption is also violated (due to outliers)

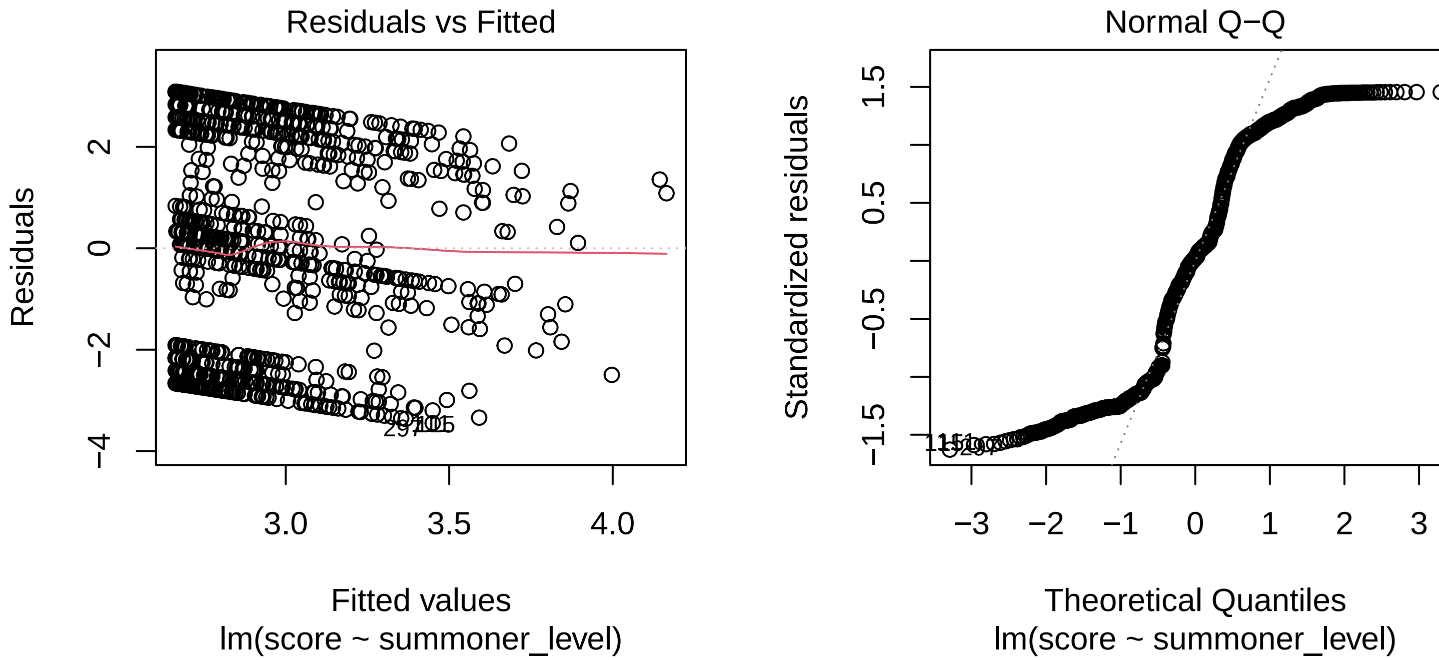






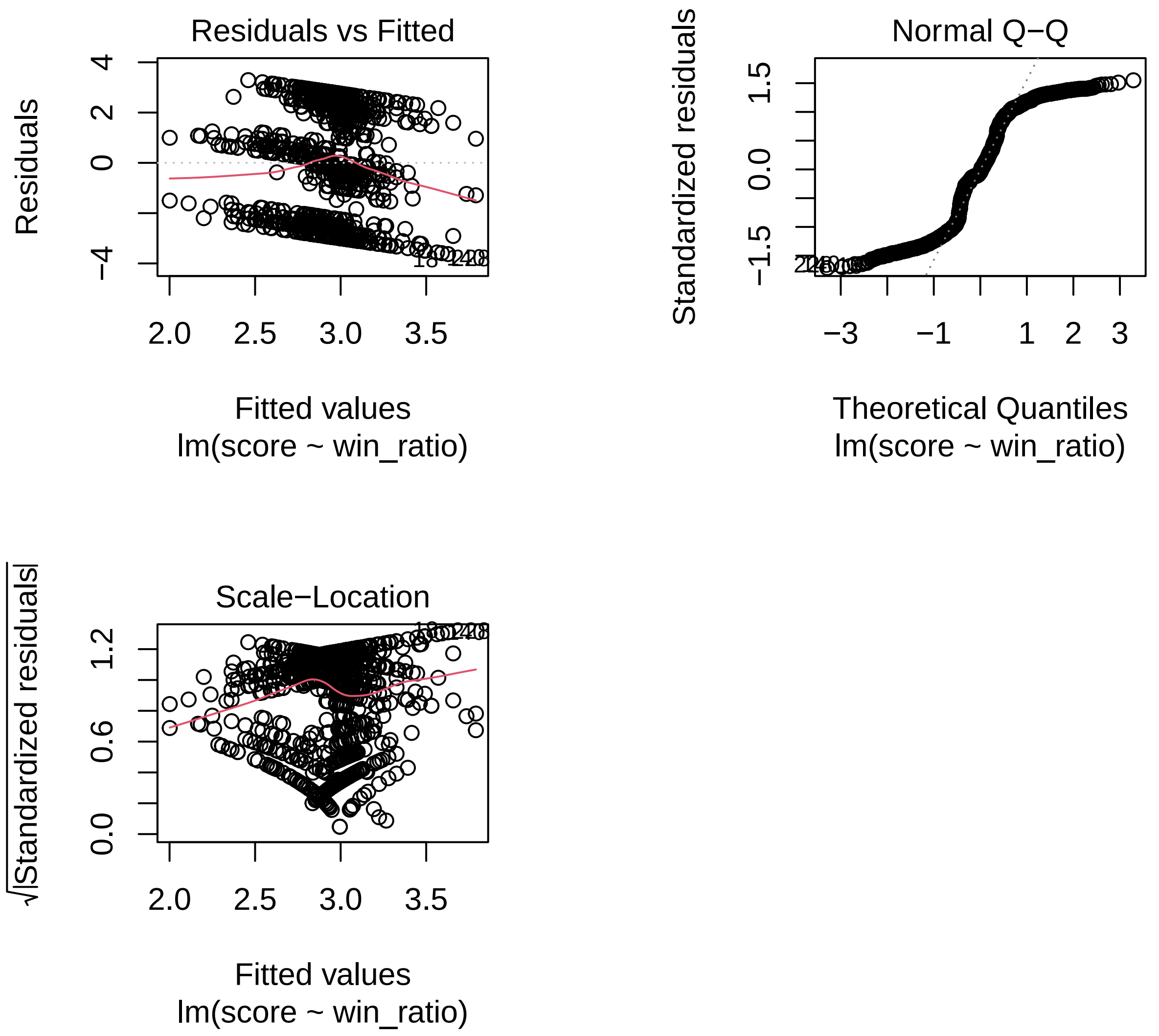
Next, for summoner level and score,

- Linearity is violated (decreasing trend in the residuals vs fitted plot)
- Normality assumption is violated (tails are too skewed in Normal Q-Q plot)



Finally, for win ratio and score,

- Linearity is violated (general decreasing trend in the residuals vs fitted plot)
- Normality assumption is violated (tails are too skewed in Normal Q-Q plot)
- Constant variances assumption is also violated (generally increasing trend)



With so many problems flying towards us at once, we halt our regression analysis. I do not believe that linear regression is suitable for this data. We need a better plan. We need more data and predictors, as well as better methods that can help us generate more accurate conclusions.

V. Discussions and Limitations

To conclude this 'chapter', we will talk about the obstacles that were presented to us throughout the analysis.

Sampling

My knowledge on sampling is very limited, as I have not taken the course on sampling theory yet. I am not sure if taking a sample of a sample is appropriate for this report, and even if so, I should have considered bootstrap sampling. The sample we obtained was not very representative of the true population, so any conclusions may not have reflected it either. However, it is unclear if I can repeatedly sample due to the restraints on Riot's API. (Despite not liking our sample, we will keep n = 1000?)

Linearity

As reported in the previous section, none of the SLR conditions are satisfied. We may not be able to generate any sound conclusions.

Predictors

Amount of games played, summoner level and win ratio are certainly not enough to establish a relationship. We need more data from the actual game to determine if there exists an underlying relationship.

A pivoting point has been established, and aside from our original objective, a new question presents itself:

Is it better to one-trick a character to climb the ranks of League of Legends, or is it safer and more efficient to diversify, and play multiple champions in multiple lanes?