MTH 256 FINAL PROJECT

DeHorn, Nathan E.

Staci White

Table of Contents

MTH 256 – Statistical Models

Final Project Overview

This project focuses on the widely popular computer game “League of Legends”, a real-time multiplayer battle in which players assume the role of a “summoner” and control a “champion” to fight against enemy teams. Two teams composed of five players each occupy a map full of various objectives, and the first team to destroy the enemy team’s “nexus” wins the game.

I conducted this experimental procedure to explain what model of play the best League of Legends players follow, and to discover which player’s model is the most effective. It would be assumed that if players modify their play style to mirror that of the best players, then they would perform better and experience more wins.

1 Introduction

League of Legends is a game that has many options/choices for players (within the game). These will be described in more detail in Section 2: Data and Exploratory Analysis, however, the presence of these options and choices make the game very customizable, giving players a wide variety of possible courses of action, as well as many different ways to respond to other players' actions.

Due to the nature of the game League of Legends, along with it being one of the most popular online games around today, many players attempt to develop their own customized strategies to play the game. For example, some players believe that gold is an extremely important resource, and that the team with the most gold will usually win. Players with this mindset will focus on accumulating minion kills, killing jungle monsters, or securing objectives, relatively easy ways to earn gold. Most of these strategies, however, are based on educated guesses. Gold allows you to buy items, and items allow your character to become stronger, and the more valuable items you can buy, the stronger you get. This thought process is why many players believe gold is important to winning games.

The process described in this report can also be applied to sport information to determine the best strategy to win a baseball game, or financial records to see what the best course of action is for a company. These are techniques that are certainly not limited to the scope of the data overviewed in this report.

2 Data and Exploratory Analysis

The dataset that I will be using was compiled by a Java program (that I wrote) that pooled together data from the game creator's (Riot Games) databases. The program's source code is available online if you are curious at https://github.com/natedehorn/StatsProject. This project also makes use of a few different API's, which are included below with credit:

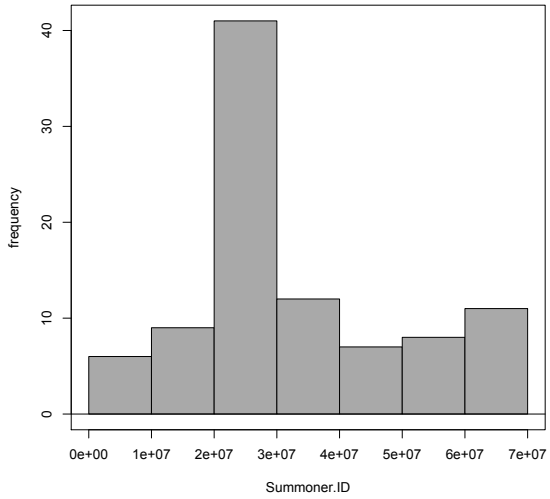
- User "rithms"' project entitled "riot-api-java" found at: https://github.com/rithms/riot-api-java.
    - This was used as an intermediary to Riot's database, allowing me to interact with their database and pull the exact data that I needed for purposes of this project.
- Apache's POI project found at: https://poi.apache.org/.
    - This was used to convert the data grabbed from Riot's database into an excel sheet.

The data set currently contains information about the top 100 summoners in North America, providing values for 21 different metrics. This data was compiled on October 14th, 2015, so data regarding the top 100 summoners may have changed (in fact, the top 100 summoners may not be the same as they were when the data was collected). Regardless, the data used is still meaningful and thereby useful.

The variables that are found in the dataset are explained in detail below:

- Summoner Name – The summoner's name. This is the name that players are identified by in-game.
- Summoner ID – This is a unique ID assigned to each summoner used within Riot's Database to differentiate between summoners. {Primary Key}

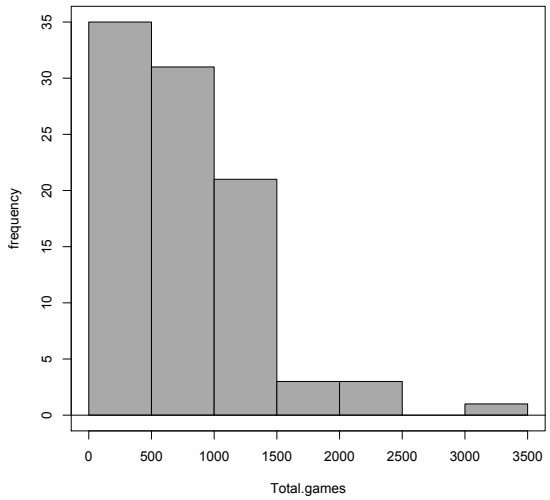
SUMMONER ID	
MEAN	31960803
STANDARD DEVIATION	17664275
MINIMUM	1496
MAXIMUM	69301989



- Champion ID – This is a unique ID assigned to each champion (different characters within the game) used within the Riot Database to differentiate between champions. An ID of "0" contains aggregate summoner info.
- Total games – This is the total amount of games played in the "Ranked 5v5" game mode.

TOTAL GAMES

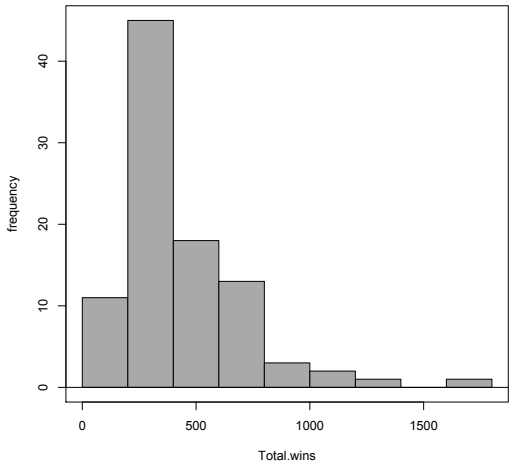
MEAN	763.8404
STANDARD DEVIATION	534.8825
MINIMUM	82.00000
MAXIMUM	3315.000



- Total wins – The total amount of wins a summoner has in the "Ranked 5v5" game mode.

TOTAL WINS

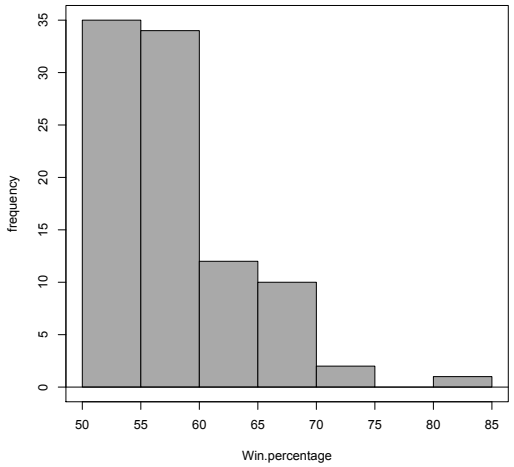
MEAN	424.8723
STANDARD DEVIATION	268.4199
MINIMUM	44.00000
MAXIMUM	1685.000



- Win percentage – This is the win percentage of a summoner in the "Ranked 5v5" game mode, calculated by $\frac{Total\ wins}{Total\ games} * 100$.

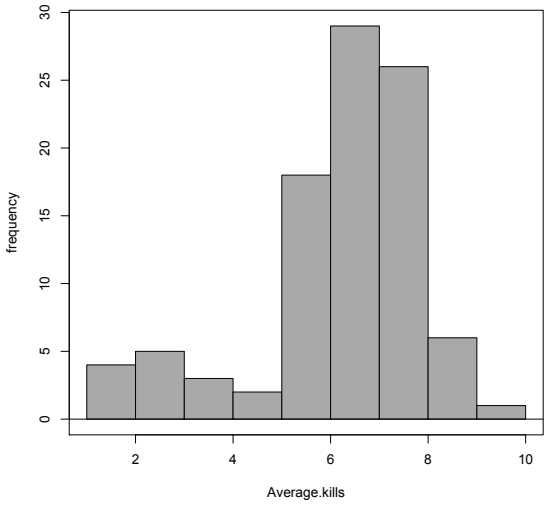
WIN PERCENTAGE

MEAN	58.07542
STANDARD DEVIATION	5.498759
MINIMUM	50.63853
MAXIMUM	80.95238



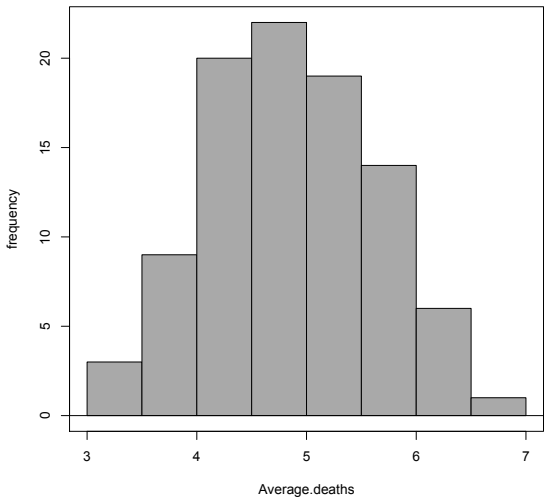
- Average kills – This is the average amount of kills a summoner gets in a single game.

AVERAGE KILLS	
MEAN	6.195425
STANDARD DEVIATION	1.632189
MINIMUM	1.826792
MAXIMUM	9.305288



- Average deaths – This is the average amount of deaths a summoner has in a single game.

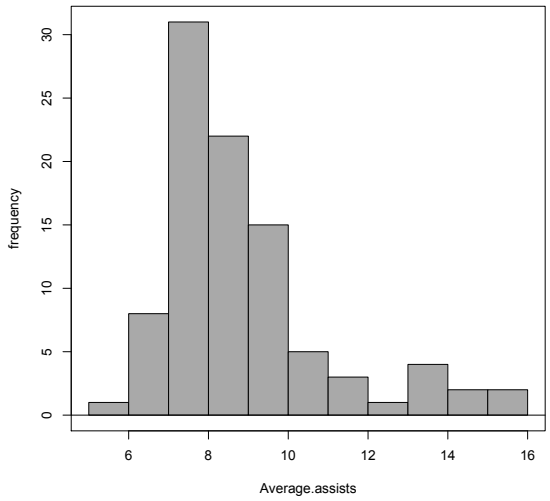
AVERAGE DEATHS	
MEAN	4.890910
STANDARD DEVIATION	0.762970
MINIMUM	3.202492
MAXIMUM	6.684444



- Average assists – This is the average amount of assists a summoner gets in a single game.

AVERAGE ASSISTS

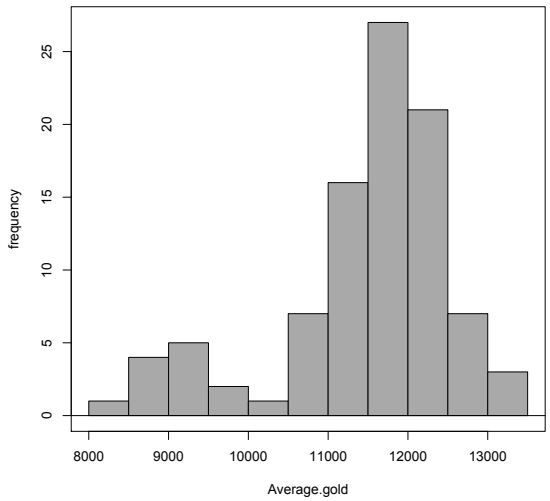
MEAN	8.863070
STANDARD DEVIATION	2.116506
MINIMUM	5.172358
MAXIMUM	15.84694



- Average gold – This is the average amount of gold earned by a summoner in a single game.

AVERAGE GOLD

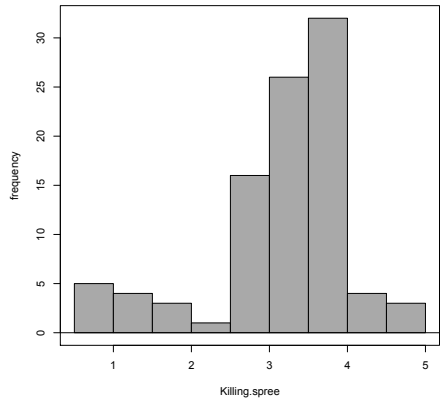
MEAN	11459.26
STANDARD DEVIATION	1066.949
MINIMUM	8483.951
MAXIMUM	13389.81



- Killing spree – This is the average killing spree of a summoner in a single game. Kills are considered to be a grouped in a killing spree if they are within 10 seconds of one another.

KILLING SPREE

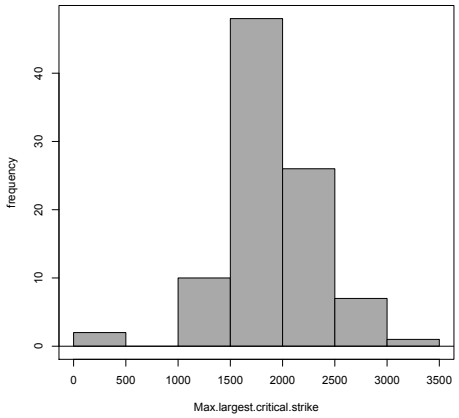
MEAN	3.169347
STANDARD DEVIATION	0.896438
MINIMUM	0.724537
MAXIMUM	4.882212



- Max largest critical strike – Each game, a summoner has a largest critical strike, representing the single highest damage dealing attack that they performed in that game. This value represents the maximum value of all the averages of highest critical strikes per champion. In other words, it is the average largest critical strike amount of a summoner in a single game.

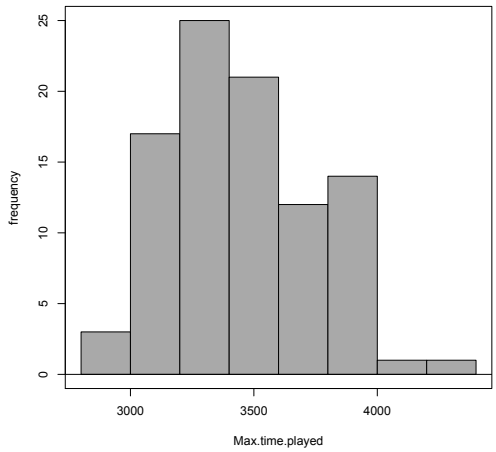
MAX LARGEST CRITICAL SPREE

MEAN	1893.606
STANDARD DEVIATION	45.44055
MINIMUM	331.0000
MAXIMUM	3350.000



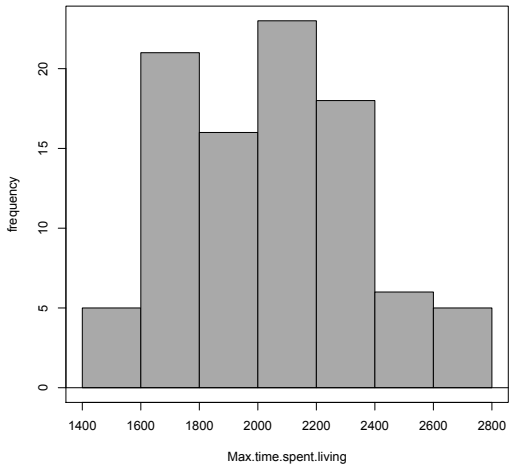
- Max time played – This is the average amount of time (in seconds) in a single game played by a summoner.

MAX TIME PLAYED	
MEAN	3445.830
STANDARD DEVIATION	289.0861
MINIMUM	2958.000
MAXIMUM	4318.000



- Max time spent living – This is the average amount of time (in seconds) that a summoner spends alive during a single game.

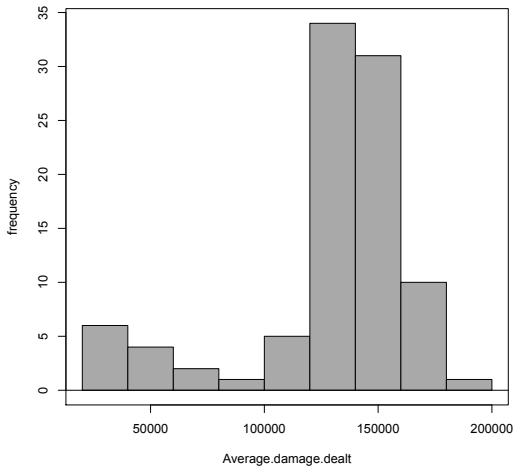
MAX TIME SPENT LIVING	
MEAN	2042.638
STANDARD DEVIATION	303.8686
MINIMUM	1416.000
MAXIMUM	2696.000



- Average damage dealt – This is the average amount of damage dealt by a summoner in a single game.

AVERAGE DAMAGE DEALT

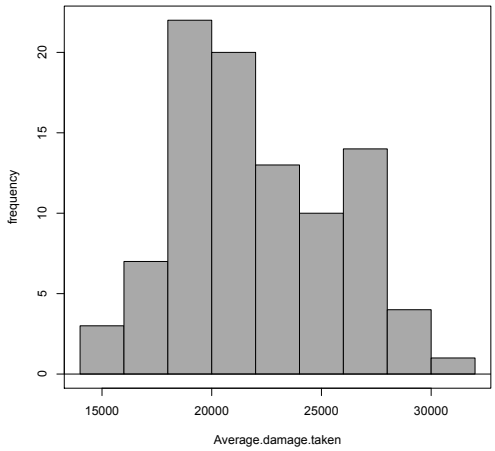
MEAN	129171.3
STANDARD DEVIATION	36293.86
MINIMUM	29829.14
MAXIMUM	187360.6



- Average damage taken - This is the average amount of damage taken by a summoner in a single game.

AVERAGE DAMAGE TAKEN

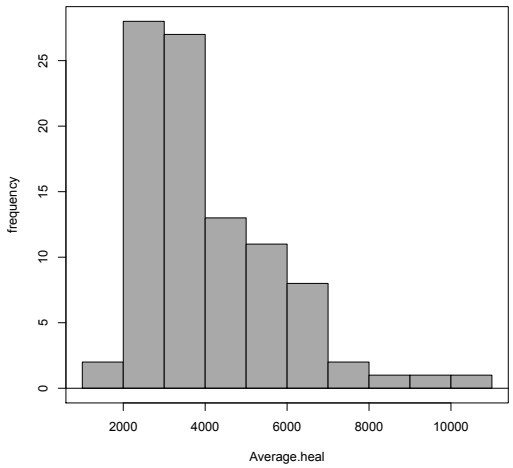
MEAN	22045.57
STANDARD DEVIATION	3701.767
MINIMUM	14346.52
MAXIMUM	30267.73



- Average heal – This is the average amount of health that a summoner heals during a single game.

AVERAGE HEAL

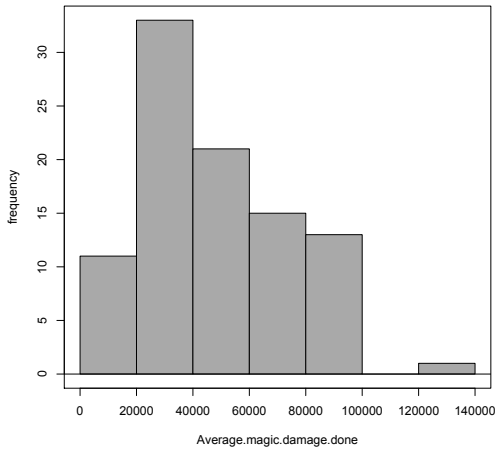
MEAN	4127.912
STANDARD DEVIATION	1614.532
MINIMUM	1790.485
MAXIMUM	13389.81



- Average magic damage done – This is the average amount of magic damage dealt by a summoner in a single game.

AVERAGE MAGIC DAMAGE DONE

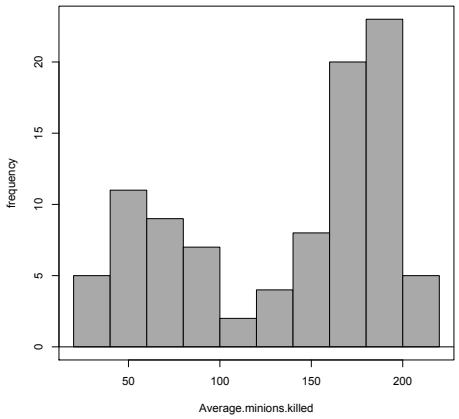
MEAN	47012.13
STANDARD DEVIATION	25381.53
MINIMUM	5668.685
MAXIMUM	125577.8



- Average minions killed – This is the average amount of minions killed by a summoner in a single game. Dealing the final blow to kill a minion rewards the killer with a small amount of gold.

AVERAGE MINIONS KILLED

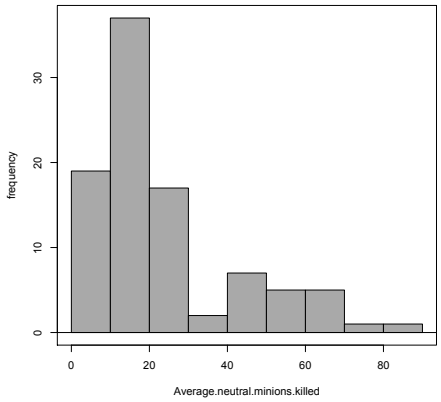
MEAN	135.4705
STANDARD DEVIATION	57.43030
MINIMUM	28.50155
MAXIMUM	217.4756



- Average neutral minions killed – This is the average amount of neutral minions killed by a summoner in a single game. Neutral minions are much more difficult to kill than minions. Dealing the final blow to kill a neutral minion rewards the killer with a large amount of gold, and in some cases: a buff (temporary increase in strength).

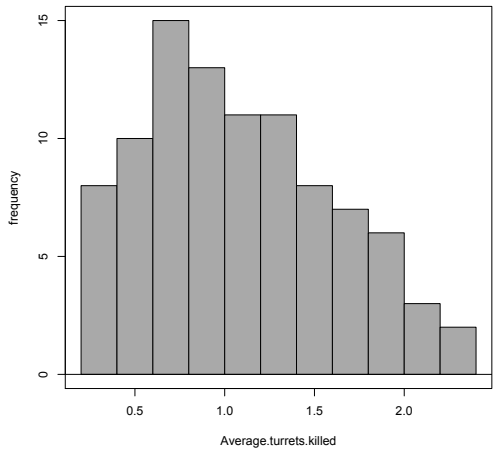
AVERAGE NEUTRAL MINIONS KILLED

MEAN	23.56984
STANDARD DEVIATION	18.91840
MINIMUM	1.783611
MAXIMUM	83.84768



- Average turrets killed – This is the average amount of turrets killed by a summoner's team in a single game.

AVERAGE TURRETS KILLED	
MEAN	1.083962
STANDARD DEVIATION	0.516726
MINIMUM	0.211838
MAXIMUM	2.280206



I proceeded to find the min and max values of within my dataset, and highlighted where they occurred in hopes of finding some meaningful interactions between certain predictors. What I found was quite useful. It appears that there are several times when multiple maximum/minimum values are found represented by the same summoner. I wanted to record all these interactions to test if they are significant later when we fit our model.

SUMMONER	**INTERACTION VARIABLES**
ADRIANMA	Max.largest.critical.strike, Average.minions.killed, Average.turrets.killed
C9SNEAKY	Average.minions.killed, Average.turrets.killed
DIAMOND1KHAZIX	Average.kills, Killing.spree, Average.magic.damage.done
LIQUIDPAINLESS	Average.kills, Average.gold, Average.damage.dealt, Average.damage.taken
IMAQTPIE	Average.deaths, Average.gold
MAMMAMIA	Total.games, Total.wins
BOXERPETE	Average.assists, Average.damage.dealt, Average.damage.taken
ANNIEBOT	Total.games, Total.wins

3 Methods

To answer my question of finding the best strategy for winning games, I decided to fit a linear model that would predict Win.percentage from any of the given variables I had available in the data set. I then wanted to refine my model to be as accurate as possible while also being concise and easy to understand.

To begin, I created a scatterplot matrix of all the variables. What this told me is that all predictor variable, except for Champion.ID (which has all "0" values), should be considered in our linear model. I then fit the model below predicting Win.percentage from all other variables:

$$Win.percentage \sim Average.assists + Average.damage.dealt + Average.damage.taken + Average.deaths + Average.gold + Average.heal + Average.kills + Average.magic.damage.done + Average.minions.killed + Average.neutral.minions.killed + Average.turrets.killed + Killing.spree + Max.largest.critical.strike + Max.time.played + Max.time.spent.living + Summoner.ID + Total.games + Total.wins$$

LINEAR.MODEL.1

R-SQUARED	0.7768
ADJUSTED R-SQUARED	0.7232
F-STATISTIC	14.5
P-VALUE	< 2.2e-16

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	8.040e+01	1.286e+01	6.252	2.26e-08	***
Average.assists	8.357e-03	6.841e-01	0.012	0.990285	
Average.damage.dealt	4.762e-05	7.797e-05	0.611	0.543191	
Average.damage.taken	1.312e-04	2.144e-04	0.612	0.542279	
Average.deaths	-1.790e+00	1.170e+00	-1.530	0.130279	
Average.gold	-2.747e-04	2.226e-03	-0.123	0.902115	
Average.heal	1.728e-04	3.708e-04	0.466	0.642510	
Average.kills	-5.482e-01	1.683e+00	-0.326	0.745475	
Average.magic.damage.done	5.216e-05	2.307e-05	2.261	0.026646	*
Average.minions.killed	-1.096e-01	6.794e-02	-1.613	0.110987	
Average.neutral.minions.killed	-2.405e-01	1.394e-01	-1.724	0.088752	.
Average.turrets.killed	3.786e+00	1.934e+00	1.958	0.053986	.
Killing.spree	4.495e+00	2.923e+00	1.538	0.128275	
Max.largest.critical.strike	-1.122e-03	8.841e-04	-1.269	0.208397	
Max.time.played	-2.122e-03	1.283e-03	-1.654	0.102240	
Max.time.spent.living	-5.540e-03	1.413e-03	-3.922	0.000193	***
Summoner.ID	2.270e-08	1.960e-08	1.158	0.250347	
Total.games	-6.225e-02	1.112e-02	-5.598	3.38e-07	***
Total.wins	1.187e-01	2.208e-02	5.376	8.28e-07	***

This equation actually performs pretty well. Our R^2 and R^2_{Adj} values were 0.7768 and 0.7232 respectively, and through an F-test, we find 14.5 as our F-statistic and less than 2.2e-16 for our p-value; making this a very good model. At the same time, this model is very complex/hard to understand, and has many variables (Average.assists, Average.damage.dealt, Average.damage.taken, Average.deaths, etc.) that are not considered significant by this model. We can also see that since our R^2 value is significantly greater than our R^2_{Adj} value, there may be a few worthless predictors in the model.

To make our model a bit more concise, we need to remove a few of these insignificant variables from our model. Before we do this, however, it is important to add in any important interactions that we may believe to be important in our model. Looking back to our exploratory data, we found interactions that already exist in the dataset, so I decided to add these interactions into our model. Notice some of the interactions (Total.games:Total.wins, Average.damage.dealt, Average.damage.taken, etc.) occur twice in the dataset, making these especially important interactions to examine.

I then fit the model Linear.model.2 below, an enlargement of Linear.model.1 obtained by adding the interactions we discussed at the end of Part 2.

$$
\begin{aligned}
Win.percentage \sim\ & Average.assists + Average.damage.dealt + \\
& Average.damage.taken + Average.deaths + Average.gold + Average.heal + \\
& Average.kills + Average.magic.damage.done + Average.minions.killed + \\
& Average.neutral.minions.killed + Average.turrets.killed + Killing.spree + \\
& Max.largest.critical.strike + Max.time.played + Max.time.spent.living + \\
& Summoner.ID + Total.games + Total.wins + \\
& Max.largest.critical.strike{:}Average.minions.killed{:}Average.turrets.killed + \\
& Average.minions.killed{:}Average.turrets.killed + \\
& Average.kills{:}Killing.spree{:}Average.magic.damage.done + \\
& Average.kills{:}Average.gold + Average.damage.dealt{:}Average.damage.taken + \\
& Average.deaths{:}Average.gold + Total.games{:}Total.wins + \\
& Average.assists{:}Average.damage.dealt{:}Average.damage.taken
\end{aligned}
$$

LINEAR.MODEL.2

R-SQUARED	0.8546
ADJUSTED R-SQUARED	0.7982
F-STATISTIC	15.15
P-VALUE	< 2.2e-16

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	7.158e+01	3.078e+01	2.326	0.02308	*
Average.assists	1.441e+00	1.092e+00	1.320	0.19138	
Average.damage.dealt	1.464e-04	1.502e-04	0.975	0.33310	
Average.damage.taken	4.843e-04	7.795e-04	0.621	0.53655	
Average.deaths	-5.611e+00	5.904e+00	-0.951	0.34527	
Average.gold	-2.797e-03	4.294e-03	-0.651	0.51706	
Average.heal	2.966e-04	3.461e-04	0.857	0.39452	
Average.kills	4.219e+00	4.564e+00	0.925	0.35854	
Average.magic.damage.done	-4.257e-05	5.429e-05	-0.784	0.43568	
Average.minions.killed	-1.834e-02	8.338e-02	-0.220	0.82658	
Average.neutral.minions.killed	-3.178e-02	1.606e-01	-0.198	0.84376	
Average.turrets.killed	2.941e+00	5.333e+00	0.551	0.58321	
Killing.spree	1.615e+00	3.706e+00	0.436	0.66437	
Max.largest.critical.strike	-7.456e-04	1.178e-03	-0.633	0.52909	
Max.time.played	-6.625e-04	1.196e-03	-0.554	0.58159	
Max.time.spent.living	-4.141e-03	1.347e-03	-3.075	0.00305	**
Summoner.ID	3.152e-09	1.850e-08	0.170	0.86526	
Total.games	-7.686e-02	1.146e-02	-6.705	5.10e-09	***
Total.wins	1.265e-01	2.164e-02	5.845	1.65e-07	***
Average.minions.killed:Average.turrets.killed	-7.082e-03	3.154e-02	-0.225	0.82303	
Average.gold:Average.kills	-3.547e-04	3.766e-04	-0.942	0.34957	
Average.damage.dealt:Average.damage.taken	-1.104e-09	5.547e-09	-0.199	0.84282	
Average.deaths:Average.gold	3.874e-04	5.180e-04	0.748	0.45720	
Total.games:Total.wins	7.260e-06	1.417e-06	5.124	2.74e-06	***
Average.minions.killed:Average.turrets.killed:Max.largest.critical.strike	1.740e-06	5.074e-06	0.343	0.73277	
Average.kills:Average.magic.damage.done:Killing.spree	3.021e-06	2.109e-06	1.432	0.15668	
Average.assists:Average.damage.dealt:Average.damage.taken	-3.505e-10	2.508e-10	-1.397	0.16697	

Once again, our model performs pretty well. Our R^2 and R^2_{Adj} values were 0.8546 and 0.7982, and through an F-test, we find 15.15 as our F-statistic and less than 2.2e-16 for our p-value; making this a very good model. This model, however, is *still* very complex/hard to understand, and has many more variables that are not considered significant by this model. We can also see that since our R^2 value is significantly greater than our R^2_{Adj} value, there may be a few worthless predictors in the model.

Upon fist inspection, there is something logically wrong about our model. We are predicting Win.percentage based on many variables, including Total.games and Total.wins, the same predictors used to calculate Win.percentage. It becomes quite obvious that it is foolish to include these predictors in our model.

Below is Linear.model.3, found by removing all instances of the predictors Total.games and Total.wins from Linear.model.2:

$$Win.percentage \sim Average.assists + Average.damage.dealt + Average.damage.taken + Average.deaths + Average.gold + Average.heal + Average.kills + Average.magic.damage.done + Average.minions.killed + Average.neutral.minions.killed + Average.turrets.killed + Killing.spree + Max.largest.critical.strike + Max.time.played + Max.time.spent.living + Summoner.ID + Max.largest.critical.strike:Average.minions.killed:Average.turrets.killed + Average.minions.killed:Average.turrets.killed + Average.kills:Killing.spree:Average.magic.damage.done + Average.kills:Average.gold + Average.damage.dealt:Average.damage.taken + Average.deaths:Average.gold + Average.assists:Average.damage.dealt:Average.damage.taken$$

LINEAR.MODEL.3

R-SQUARED	0.7273
ADJUSTED R-SQUARED	0.6377
F-STATISTIC	8.118
P-VALUE	4.612e-12

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	1.347e+01	3.867e+01	0.348	0.72856	
Average.assists	1.534e-01	1.423e+00	0.108	0.91443	
Average.damage.dealt	3.116e-04	1.781e-04	1.749	0.08465	.
Average.damage.taken	1.935e-03	8.827e-04	2.192	0.03168	*
Average.deaths	-2.034e+00	7.595e+00	-0.268	0.78966	
Average.gold	5.662e-03	5.088e-03	1.113	0.26959	
Average.heal	5.787e-04	4.520e-04	1.280	0.20467	
Average.kills	6.988e+00	6.023e+00	1.160	0.24992	
Average.magic.damage.done	-1.214e-04	6.638e-05	-1.828	0.07175	.
Average.minions.killed	-1.447e-01	1.073e-01	-1.349	0.18176	
Average.neutral.minions.killed	-3.247e-01	2.064e-01	-1.573	0.12025	
Average.turrets.killed	6.825e+00	7.051e+00	0.968	0.33640	
Killing.spree	-5.416e+00	4.640e+00	-1.167	0.24706	
Max.largest.critical.strike	-1.165e-03	1.550e-03	-0.751	0.45504	
Max.time.played	-3.383e-03	1.471e-03	-2.300	0.02444	*
Max.time.spent.living	-4.678e-03	1.623e-03	-2.881	0.00525	**
Summoner.ID	3.502e-08	2.345e-08	1.494	0.13975	
Average.minions.killed:Average.turrets.killed	-9.978e-03	4.212e-02	-0.237	0.81345	
Average.gold:Average.kills	-3.721e-04	4.914e-04	-0.757	0.45143	
Average.damage.dealt:Average.damage.taken	-1.147e-08	6.466e-09	-1.774	0.08043	.
Average.deaths:Average.gold	-2.832e-04	6.679e-04	-0.424	0.67282	
Average.minions.killed:Average.turrets.killed:Max.largest.critical.strike	7.053e-07	6.733e-06	0.105	0.91687	
Average.kills:Average.magic.damage.done:Killing.spree	7.159e-06	2.592e-06	2.762	0.00733	**
Average.assists:Average.damage.dealt:Average.damage.taken	-2.015e-10	3.301e-10	-0.610	0.54354	

As should be expected, this model performs a bit more poorly than our previous two. This is because we removed two predictors that were obviously having a direct correlation to Win.percentage. However, performing an F-test (F-statistic: 8.118, p-value: 4.612e-12) shows that our model is still effective, but there still seems to be some erroneous predictors in our model.

To combat this, I first checked for multicollinearity by computing variance inflation factors for each of the remaining predictors in our model. This was done with the hopes of being able to identify data that is redundant, allowing us to 'clean-up' our model a bit.

```
> vif(LinearModel.3)
                                                          Average.assists
                                                                77.002511
                                                     Average.damage.taken
                                                                90.642658
                                                             Average.gold
                                                               250.155565
                                                            Average.kills
                                                               820.580239
                                                   Average.minions.killed
                                                               322.409703
                                                   Average.turrets.killed
                                                               112.710144
                                             Max.largest.critical.strike
                                                                 4.213312
                                                     Max.time.spent.living
                                                                 2.066189
                           Average.minions.killed:Average.turrets.killed
                                                               229.974994
                               Average.damage.dealt:Average.damage.taken
                                                               355.255690
Average.minions.killed:Average.turrets.killed:Max.largest.critical.strike
                                                                31.517605
               Average.assists:Average.damage.dealt:Average.damage.taken
                                                                57.792344
                                                     Average.damage.dealt
                                                               354.872125
                                                           Average.deaths
                                                               285.057467
                                                             Average.heal
                                                                 4.521614
                                                Average.magic.damage.done
                                                                24.103122
                                             Average.neutral.minions.killed
                                                               129.479701
                                                            Killing.spree
                                                               146.904104
                                                          Max.time.played
                                                                 1.535445
                                                              Summoner.ID
                                                                 1.456185
                                               Average.gold:Average.kills
                                                              1053.374516
                                              Average.deaths:Average.gold
                                                               485.720476
                     Average.kills:Average.magic.damage.done:Killing.spree
                                                                28.021958
```

Examining the variance inflation factors, it seems that there is very much multicollinearity occurring in this particular model. This is not a bad thing, however, as it makes sense that these

factors should all be related. For example, obtaining a kill, destroying a turret, dealing the killing strike to a minion, and many other actions award a summoner with gold, so it is expected that many of our other predictors are also related in a similar fashion. With this in mind, we should only eliminate/combine predictors that exhibit extreme multicollinearity, with an especially focused eye on our interactions, since those interactions may already be accounted for by the individual factors.

For this reason, I decided to remove Average.gold:Average.kills, Average.deaths:Average.gold, and Average.kills from the model. I decided to remove Average.kills in our model because kills in game are a product of being stronger that your opponents (as silly as that may sound), and almost all factors contribute to getting more kills in some way or another. I created the linear model predicting Average.kills from the other applicable variables to illustrate this more clearly. The model, Linear.model.4, is shown below:

$$\begin{aligned} Average.kills \sim\ & Average.assists + Average.damage.dealt + Average.damage.taken \\ & + Average.deaths + Average.gold + Average.heal \\ & + Average.magic.damage.done + Average.minions.killed \\ & + Average.neutral.minions.killed + Average.turrets.killed \\ & + Max.largest.critical.strike + Max.time.played \\ & + Max.time.spent.living \end{aligned}$$

LINEAR.MODEL.4

R-SQUARED	0.9288
ADJUSTED R-SQUARED	0.9172
F-STATISTIC	80.26
P-VALUE	< 2.2e-16

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	-4.794e+00	1.832e+00	-2.616	0.010628	*
Average.assists	-6.587e-01	7.660e-02	-8.599	5.39e-13	***
Average.damage.dealt	-1.174e-05	1.244e-05	-0.944	0.348064	
Average.damage.taken	-2.054e-05	3.361e-05	-0.611	0.542834	
Average.deaths	-2.856e-02	1.004e-01	-0.284	0.776767	
Average.gold	2.399e-03	2.286e-04	10.493	< 2e-16	***
Average.heal	-1.398e-04	5.648e-05	-2.475	0.015423	*
Average.magic.damage.done	4.712e-06	3.450e-06	1.366	0.175853	
Average.minions.killed	-3.881e-02	9.931e-03	-3.908	0.000194	***
Average.neutral.minions.killed	-5.435e-02	2.138e-02	-2.543	0.012933	*
Average.turrets.killed	9.313e-02	3.033e-01	0.307	0.759637	
Max.largest.critical.strike	-7.861e-05	1.408e-04	-0.558	0.578264	
Max.time.played	-3.219e-04	1.895e-04	-1.698	0.093314	.
Max.time.spent.living	-2.467e-04	2.055e-04	-1.201	0.233365	

This model is excellent. It has R^2 and R^2_{Adj} values of 0.9288 and 0.9172 respectively, and through an F-test, we find 80.26 as our F-statistic and less than 2.2e-16 for our p-value; making this an exceptional model. This tells us that while Average.kills may still be a significant predictor of Win.percentage, its significance is very well conserved throughout the other predictors in the

model. After removing Average.gold:Average.kills, Average.deaths:Average.gold, and Average.kills from Linear.model.3, I fit Linear.model.5, shown below.

$$
\begin{aligned}
&Win.percentage \sim Average.assists + Average.damage.dealt \\
&+ Average.damage.taken + Average.deaths + Average.gold + Average.heal \\
&+ Average.magic.damage.done + Average.minions.killed \\
&+ Average.neutral.minions.killed + Average.turrets.killed + Killing.spree \\
&+ Max.largest.critical.strike + Max.time.played + Max.time.spent.living \\
&+ Summoner.ID \\
&+ Max.largest.critical.strike{:}Average.minions.killed{:}Average.turrets.killed \\
&+ Average.minions.killed{:}Average.turrets.killed \\
&+ Average.kills{:}Killing.spree{:}Average.magic.damage.done \\
&+ Average.damage.dealt{:}Average.damage.taken \\
&+ Average.assists{:}Average.damage.dealt{:}Average.damage.taken
\end{aligned}
$$

LINEAR.MODEL.5

R-SQUARED	0.7177
ADJUSTED R-SQUARED	0.6403
F-STATISTIC	9.278
P-VALUE	4.942e-13

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	4.736e+01	2.544e+01	1.862	0.066643	.
Average.assists	3.428e-02	1.318e+00	0.026	0.979316	
Average.damage.dealt	3.301e-04	1.617e-04	2.041	0.044898	*
Average.damage.taken	2.080e-03	7.131e-04	2.917	0.004691	**
Average.deaths	-3.827e+00	1.037e+00	-3.691	0.000427	***
Average.gold	1.412e-03	2.912e-03	0.485	0.629170	
Average.heal	6.143e-04	4.380e-04	1.403	0.164968	
Average.magic.damage.done	-1.246e-04	6.431e-05	-1.938	0.056462	.
Average.minions.killed	-1.204e-01	1.024e-01	-1.175	0.243833	
Average.neutral.minions.killed	-3.128e-01	2.015e-01	-1.553	0.124834	
Average.turrets.killed	1.105e+01	6.193e+00	1.784	0.078513	.
Killing.spree	-4.267e-01	2.613e+00	-0.163	0.870740	
Max.largest.critical.strike	-9.674e-04	1.531e-03	-0.632	0.529524	
Max.time.played	-3.139e-03	1.452e-03	-2.161	0.033951	*
Max.time.spent.living	-4.814e-03	1.604e-03	-3.001	0.003682	**
Summoner.ID	3.449e-08	2.309e-08	1.494	0.139597	
Average.minions.killed:Average.turrets.killed	-3.081e-02	3.886e-02	-0.793	0.430451	
Average.damage.dealt:Average.damage.taken	-1.234e-08	5.594e-09	-2.207	0.030479	*
Average.minions.killed:Average.turrets.killed:Max.largest.critical.strike	3.113e-07	6.692e-06	0.047	0.963024	
Average.magic.damage.done:Killing.spree:Average.kills	7.128e-06	2.542e-06	2.805	0.006453	**
Average.assists:Average.damage.dealt:Average.damage.taken	-1.692e-10	2.915e-10	-0.581	0.563350	

I then performed a forward/backward and backward/forward stepwise regression from this point using the BIC method.

```
> stepwise(LinearModel.5, direction='forward/backward', criterion='BIC')

Direction:  forward/backward
Criterion:  BIC

Start:  AIC=323.99
Win.percentage ~ 1

                               Df Sum of Sq    RSS    AIC
+ Max.time.played               1    610.22 2201.8 305.54
+ Average.deaths                1    386.48 2425.5 314.63
+ Max.largest.critical.strike   1    215.24 2596.7 321.05
+ Max.time.spent.living         1    207.84 2604.2 321.31
+ Summoner.ID                   1    202.76 2609.2 321.50
+ Killing.spree                 1    186.53 2625.4 322.08
<none>                                      2812.0 323.99
+ Average.damage.taken          1     83.47 2728.5 325.70
+ Average.turrets.killed        1     76.40 2735.6 325.94
+ Average.neutral.minions.killed 1     6.57 2805.4 328.31
+ Average.assists               1     6.08 2805.9 328.33
+ Average.gold                  1     4.20 2807.8 328.39
+ Average.minions.killed        1     1.64 2810.3 328.48
+ Average.damage.dealt          1     0.91 2811.1 328.50
+ Average.magic.damage.done     1     0.21 2811.8 328.52
+ Average.heal                  1     0.02 2812.0 328.53
Step:  AIC=305.54
Win.percentage ~ Max.time.played

                               Df Sum of Sq    RSS    AIC
+ Average.deaths                1    318.34 1883.4 295.40
+ Max.largest.critical.strike   1    165.58 2036.2 302.73
<none>                                      2201.8 305.54
+ Summoner.ID                   1     93.49 2108.3 306.00
+ Killing.spree                 1     92.32 2109.4 306.05
+ Average.damage.taken          1     82.40 2119.4 306.49
+ Max.time.spent.living         1     80.52 2121.2 306.58
+ Average.turrets.killed        1     17.86 2183.9 309.31
+ Average.neutral.minions.killed 1    13.42 2188.3 309.51
+ Average.damage.dealt          1      9.80 2192.0 309.66
+ Average.assists               1      2.86 2198.9 309.96
+ Average.gold                  1      1.95 2199.8 310.00
+ Average.magic.damage.done     1      1.88 2199.9 310.00
+ Average.heal                  1      0.71 2201.1 310.05
+ Average.minions.killed        1      0.04 2201.7 310.08
- Max.time.played               1    610.22 2812.0 323.99


Step:  AIC=295.4
Win.percentage ~ Max.time.played + Average.deaths

                               Df Sum of Sq    RSS    AIC
+ Max.time.spent.living         1    423.41 1460.0 276.01
+ Killing.spree                 1    115.00 1768.4 294.02
<none>                                      1883.4 295.40
+ Summoner.ID                   1     86.38 1797.0 295.53
+ Max.largest.critical.strike   1     76.00 1807.4 296.07
+ Average.turrets.killed        1     59.47 1824.0 296.93
+ Average.minions.killed        1     28.30 1855.1 298.52
+ Average.neutral.minions.killed 1    25.95 1857.5 298.64
+ Average.damage.taken          1     20.05 1863.4 298.94
+ Average.heal                  1     19.41 1864.0 298.97
+ Average.gold                  1     16.99 1866.4 299.09
+ Average.assists               1     12.40 1871.0 299.32
+ Average.damage.dealt          1      3.59 1879.8 299.76
+ Average.magic.damage.done     1      0.05 1883.4 299.94
- Average.deaths                1    318.34 2201.8 305.54
- Max.time.played               1    542.07 2425.5 314.63


Step:  AIC=276.01
Win.percentage ~ Max.time.played + Average.deaths + Max.time.spent.living

                               Df Sum of Sq    RSS    AIC
+ Killing.spree                 1    125.94 1334.1 272.07
+ Summoner.ID                   1     76.85 1383.2 275.47
<none>                                      1460.0 276.01
+ Average.turrets.killed        1     44.28 1415.7 277.65
+ Average.gold                  1     33.44 1426.6 278.37
+ Average.minions.killed        1     22.93 1437.1 279.06
+ Average.heal                  1     16.83 1443.2 279.46
+ Average.assists               1     15.20 1444.8 279.57
+ Average.damage.dealt          1     14.61 1445.4 279.61
+ Max.largest.critical.strike   1     10.78 1449.2 279.85
+ Average.magic.damage.done     1      9.16 1450.9 279.96
+ Average.damage.taken          1      7.11 1452.9 280.09
+ Average.neutral.minions.killed 1     3.79 1456.2 280.31
- Max.time.played               1    256.65 1716.7 286.69
- Max.time.spent.living         1    423.41 1883.4 295.40
- Average.deaths                1    661.22 2121.2 306.58
```

```
Step:  AIC=272.07
Win.percentage ~ Max.time.played + Average.deaths + Max.time.spent.living +
    Killing.spree

                                Df Sum of Sq    RSS    AIC
+ Average.damage.dealt           1    154.50 1179.6 265.04
+ Average.assists                1    106.27 1227.8 268.81
+ Average.gold                   1     90.60 1243.5 270.00
+ Summoner.ID                    1     64.70 1269.4 271.94
<none>                                      1334.1 272.07
+ Max.largest.critical.strike    1     54.48 1279.6 272.69
+ Average.damage.taken           1     29.40 1304.7 274.52
+ Average.neutral.minions.killed 1     29.05 1305.0 274.55
- Killing.spree                  1    125.94 1460.0 276.01
+ Average.minions.killed         1      7.06 1327.0 276.12
+ Average.heal                   1      4.79 1329.3 276.28
+ Average.magic.damage.done      1      0.09 1334.0 276.61
+ Average.turrets.killed         1      0.01 1334.1 276.61
- Max.time.played                1    191.47 1525.5 280.13
- Max.time.spent.living          1    434.35 1768.4 294.02
- Average.deaths                 1    696.51 2030.6 307.01


Step:  AIC=265.04
Win.percentage ~ Max.time.played + Average.deaths + Max.time.spent.living +
    Killing.spree + Average.damage.dealt

                                Df Sum of Sq    RSS    AIC
<none>                                      1179.6 265.04
+ Average.turrets.killed         1     28.67 1150.9 267.27
+ Average.assists                1     22.33 1157.3 267.79
+ Summoner.ID                    1     21.18 1158.4 267.88
+ Max.largest.critical.strike    1     17.57 1162.0 268.18
+ Average.neutral.minions.killed 1     10.55 1169.0 268.74
+ Average.minions.killed         1      4.85 1174.7 269.20
+ Average.heal                   1      2.45 1177.1 269.39
+ Average.gold                   1      2.28 1177.3 269.40
+ Average.damage.taken           1      1.75 1177.8 269.45
+ Average.magic.damage.done      1      1.59 1178.0 269.46
- Average.damage.dealt           1    154.50 1334.1 272.07
- Max.time.played                1    179.23 1358.8 273.80
- Killing.spree                  1    265.83 1445.4 279.61
- Average.deaths                 1    327.37 1507.0 283.52
- Max.time.spent.living          1    352.40 1532.0 285.07

Call:
lm(formula = Win.percentage ~ Max.time.played + Average.deaths +
    Max.time.spent.living + Killing.spree + Average.damage.dealt,
    data = Summ)
```

Coefficients

(Intercept)	Max.time.played	Average.deaths	Max.time.spent.living	Killing.spree	Average.damage.dealt
1.044e+02	-5.130e-03	-3.182e+00	-7.675e-03	3.971e+00	-7.716e-05

```
> stepwise(LinearModel.5, direction='backward/forward', criterion='BIC')

Direction:  backward/forward
Criterion:  BIC

Start:  AIC=295.98
Win.percentage ~ Average.assists + Average.damage.dealt + Average.damage.taken +
    Average.deaths + Average.gold + Average.heal + Average.magic.damage.done +
    Average.minions.killed + Average.neutral.minions.killed +
    Average.turrets.killed + Killing.spree + Max.largest.critical.strike +
    Max.time.played + Max.time.spent.living + Summoner.ID + Max.largest.critical.strike:Average.minions.killed:Average.turrets.killed +
    Average.minions.killed:Average.turrets.killed + Average.kills:Killing.spree:Average.magic.damage.done +
    Average.damage.dealt:Average.damage.taken + Average.assists:Average.damage.dealt:Average.damage.taken

                                                                    Df Sum of Sq    RSS    AIC
- Average.minions.killed:Average.turrets.killed:Max.largest.critical.strike  1     0.024 793.95 291.44
- Average.gold                                                       1     2.558 796.49 291.74
- Average.assists:Average.damage.dealt:Average.damage.taken          1     3.665 797.59 291.87
- Average.heal                                                       1    21.396 815.32 293.93
- Summoner.ID                                                        1    24.261 818.19 294.26
- Average.neutral.minions.killed                                     1    26.218 820.15 294.49
<none>                                                                          793.93 295.98
- Max.time.played                                                    1    50.804 844.73 297.26
- Average.magic.damage.done:Killing.spree:Average.kills              1    85.542 879.47 301.05
- Max.time.spent.living                                              1    97.940 891.87 302.37
- Average.deaths                                                     1   148.199 942.13 307.52
Step:  AIC=291.44
Win.percentage ~ Average.assists + Average.damage.dealt + Average.damage.taken +
    Average.deaths + Average.gold + Average.heal + Average.magic.damage.done +
    Average.minions.killed + Average.neutral.minions.killed +
    Average.turrets.killed + Killing.spree + Max.largest.critical.strike +
    Max.time.played + Max.time.spent.living + Summoner.ID + Average.minions.killed:Average.turrets.killed +
    Average.damage.dealt:Average.damage.taken + Average.magic.damage.done:Killing.spree:Average.kills +
    Average.assists:Average.damage.dealt:Average.damage.taken

                                                                  Df Sum of Sq    RSS    AIC
- Average.gold                                                     1     2.733 796.68 287.22
- Average.assists:Average.damage.dealt:Average.damage.taken        1     3.665 797.62 287.33
- Max.largest.critical.strike                                      1     8.237 802.19 287.86
- Average.minions.killed:Average.turrets.killed                    1    11.103 805.05 288.20
- Average.heal                                                     1    21.852 815.80 289.44
- Summoner.ID                                                      1    24.266 818.22 289.72
- Average.neutral.minions.killed                                   1    28.336 822.29 290.19
<none>                                                                      793.95 291.44
- Max.time.played                                                  1    50.791 844.74 292.72
+ Average.minions.killed:Average.turrets.killed:Max.largest.critical.strike  1     0.024 793.93 295.98
- Average.magic.damage.done:Killing.spree:Average.kills            1    90.128 884.08 297.00
- Max.time.spent.living                                            1    99.155 893.11 297.95
- Average.deaths                                                   1   149.212 943.16 303.08
```

{Some steps removed from here for brevity}

```
Step:  AIC=269.08
Win.percentage ~ Average.damage.dealt + Average.damage.taken +
    Average.deaths + Average.magic.damage.done + Average.minions.killed +
    Average.neutral.minions.killed + Average.turrets.killed +
    Killing.spree + Max.time.played + Max.time.spent.living +
    Summoner.ID + Average.damage.dealt:Average.damage.taken +
    Average.magic.damage.done:Killing.spree:Average.kills

                                                 Df Sum of Sq     RSS    AIC
- Summoner.ID                                     1     25.08  861.57 267.32
<none>                                                        836.49 269.08
- Average.neutral.minions.killed                  1     43.11  879.60 269.26
- Average.minions.killed                          1     44.82  881.31 269.44
- Max.time.played                                 1     53.93  890.41 270.41
- Average.damage.dealt:Average.damage.taken       1     59.85  896.33 271.03
+ Average.heal                                    1     19.33  817.15 271.43
+ Max.largest.critical.strike                     1      6.30  830.18 272.92
+ Average.minions.killed:Average.turrets.killed   1      4.47  832.01 273.12
+ Average.assists                                 1      0.88  835.61 273.53
+ Average.gold                                    1      0.01  836.47 273.62
- Average.turrets.killed                          1     93.57  930.06 274.51
- Average.magic.damage.done:Killing.spree:Average.kills  1    123.49  959.97 277.48
- Max.time.spent.living                           1    186.04 1022.53 283.42
- Average.deaths                                  1    355.54 1192.02 297.83
Step:  AIC=267.32
Win.percentage ~ Average.damage.dealt + Average.damage.taken +
    Average.deaths + Average.magic.damage.done + Average.minions.killed +
    Average.neutral.minions.killed + Average.turrets.killed +
    Killing.spree + Max.time.played + Max.time.spent.living +
    Average.damage.dealt:Average.damage.taken + Average.magic.damage.done:Killing.spree:Average.kills

                                                 Df Sum of Sq     RSS    AIC
<none>                                                        861.57 267.32
- Average.neutral.minions.killed                  1     42.78  904.35 267.33
- Average.minions.killed                          1     47.74  909.31 267.84
- Average.damage.dealt:Average.damage.taken       1     49.88  911.45 268.06
+ Summoner.ID                                     1     25.08  836.49 269.08
- Max.time.played                                 1     64.38  925.95 269.55
+ Average.heal                                    1     14.89  846.68 270.22
+ Max.largest.critical.strike                     1      8.18  853.39 270.96
+ Average.minions.killed:Average.turrets.killed   1      4.20  857.37 271.40
+ Average.assists                                 1      1.71  859.86 271.67
+ Average.gold                                    1      0.74  860.83 271.78
- Average.turrets.killed                          1     98.79  960.36 272.98
- Average.magic.damage.done:Killing.spree:Average.kills  1    122.47  984.04 275.27
- Max.time.spent.living                           1    194.17 1055.74 281.88
- Average.deaths                                  1    332.53 1194.10 293.45
```

```
Call:
lm(formula = Win.percentage ~ Average.damage.dealt + Average.damage.taken +
    Average.deaths + Average.magic.damage.done + Average.minions.killed +
    Average.neutral.minions.killed + Average.turrets.killed +
    Killing.spree + Max.time.played + Max.time.spent.living +
    Average.damage.dealt:Average.damage.taken + Average.magic.damage.done:Killing.spree:Average.kills,
    data = Summ)
```

Coefficients

(Intercept)	Average.damage.dealt	Average.damage.taken	Average.deaths	Average.magic.damage.done	Average.minions.killed
7.439e+01	2.072e-04	1.613e-03	-3.814e+00	-1.113e-04	-1.211e-01
Average.neutral.minions.killed	Average.turrets.killed	Killing.spree	Max.time.played	Max.time.spent.living	Average.damage.dealt : Average.damage.taken
-2.739e-01	6.229e+00	7.940e-01	-3.316e-03	-6.037e-03	-8.560e-09
Average.magic.damage.done : Killing.spree : Average.kills					
7.258e-06					

The forward/backward stepwise regression using BIC returned Linear.model.6 (BIC=265.04) ,

$$Win.percentage \sim Max.time.played + Average.deaths + Max.time.spent.living + Killing.spree + Average.damage.dealt$$

while the backward/forward stepwise regression using BIC returned Linear.model.7 (BIC=267.32),

$$Win.percentage \sim Average.damage.dealt + Average.damage.taken + Average.deaths + Average.magic.damage.done + Average.minions.killed + Average.neutral.minions.killed + Average.turrets.killed + Killing.spree + Max.time.played + Max.time.spent.living + Average.damage.dealt{:}Average.damage.taken + Average.magic.damage.done{:}Killing.spree{:}Average.kills$$

Since we are using BIC, we prefer models with lower BIC values. In this case, not only is Linear.model.6 more concise and understandable than Linear.model.7, but it is also a better model according to BIC.

Just to confirm, we should also compare these models using a few different models. First, we will check that these are both effective models.

LINEAR.MODEL.6

R-SQUARED	0.5805
ADJUSTED R-SQUARED	0.5567
F-STATISTIC	24.36
P-VALUE	2.603e-15

```
> summary(LinearModel.6)

Call:
lm(formula = Win.percentage ~ Max.time.played + Average.deaths +
    Max.time.spent.living + Killing.spree + Average.damage.dealt,
    data = Summ)

Residuals:
    Min      1Q  Median      3Q     Max
-7.6557 -2.4370 -0.1094  2.4908 10.7804

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)             1.044e+02  6.341e+00  16.461  < 2e-16 ***
Max.time.played        -5.130e-03  1.403e-03  -3.657 0.000434 ***
Average.deaths         -3.182e+00  6.439e-01  -4.942 3.66e-06 ***
Max.time.spent.living -7.675e-03  1.497e-03  -5.127 1.73e-06 ***
Killing.spree          3.971e+00  8.916e-01   4.453 2.47e-05 ***
Average.damage.dealt  -7.716e-05  2.273e-05  -3.395 0.001031 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

LINEAR.MODEL.7

R-SQUARED	0.6936
ADJUSTED R-SQUARED	0.6482
F-STATISTIC	15.28
P-VALUE	3.431e-16

```
> summary(LinearModel.7)

Call:
lm(formula = Win.percentage ~ Average.damage.dealt + Average.damage.taken +
    Average.deaths + Average.magic.damage.done + Average.minions.killed +
    Average.neutral.minions.killed + Average.turrets.killed +
    Killing.spree + Max.time.played + Max.time.spent.living +
    Average.damage.dealt:Average.damage.taken + Average.magic.damage.done:Killing.spree:Average.kills,
    data = Summ)

Residuals:
    Min      1Q  Median      3Q     Max
-6.4882 -1.5987 -0.0202  1.3688  9.0404

Coefficients:
                                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                                     7.439e+01  1.060e+01   7.020 6.18e-10 ***
Average.damage.dealt                            2.072e-04  1.327e-04   1.562  0.12230
Average.damage.taken                            1.613e-03  5.651e-04   2.855  0.00547 **
Average.deaths                                 -3.814e+00  6.821e-01  -5.591 2.95e-07 ***
Average.magic.damage.done                      -1.113e-04  5.487e-05  -2.029  0.04574 *
Average.minions.killed                         -1.211e-01  5.714e-02  -2.119  0.03719 *
Average.neutral.minions.killed                 -2.739e-01  1.366e-01  -2.005  0.04825 *
Average.turrets.killed                          6.229e+00  2.044e+00   3.048  0.00311 **
Killing.spree                                   7.940e-01  1.446e+00   0.549  0.58448
Max.time.played                                -3.316e-03  1.348e-03  -2.460  0.01601 *
Max.time.spent.living                          -6.037e-03  1.413e-03  -4.273 5.23e-05 ***
Average.damage.dealt:Average.damage.taken      -8.560e-09  3.953e-09  -2.166  0.03328 *
Average.magic.damage.done:Killing.spree:Average.kills  7.258e-06  2.139e-06   3.393  0.00107 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

By conducting an F-test on Linear.model.6 (F-statistic: 24.36; p-value: 2.603e-15) and Linear.model.7 (F-statistic: 15.28; p-value: 3.431e-16) we can see that both of these models are very effective models, but Linear.model.7 seems to explain just under 70% of Win.percentage's variability, while Linear.model.6 seems to falls a bit short, explaining just under 60% o of Win.percentage's variability.

Since, as it turns out, Linear.model.6 is a subset of Liner.model.7, I then compared the two models by building an Analysis of Variance (ANOVA) Table, and then conducted an F-test to see if the extra predictors found in Linear.model.7 are truly necessary.

```
> anova(LinearModel.6, LinearModel.7)
Analysis of Variance Table

Model 1: Win.percentage ~ Max.time.played + Average.deaths + Max.time.spent.living +
    Killing.spree + Average.damage.dealt
Model 2: Win.percentage ~ Average.damage.dealt + Average.damage.taken +
    Average.deaths + Average.magic.damage.done + Average.minions.killed +
    Average.neutral.minions.killed + Average.turrets.killed +
    Killing.spree + Max.time.played + Max.time.spent.living +
    Average.damage.dealt:Average.damage.taken + Average.magic.damage.done:Killing.spree:Average.kills
  Res.Df     RSS Df Sum of Sq      F    Pr(>F)
1     88 1179.58
2     81  861.57  7    318.01 4.2711 0.0004629 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

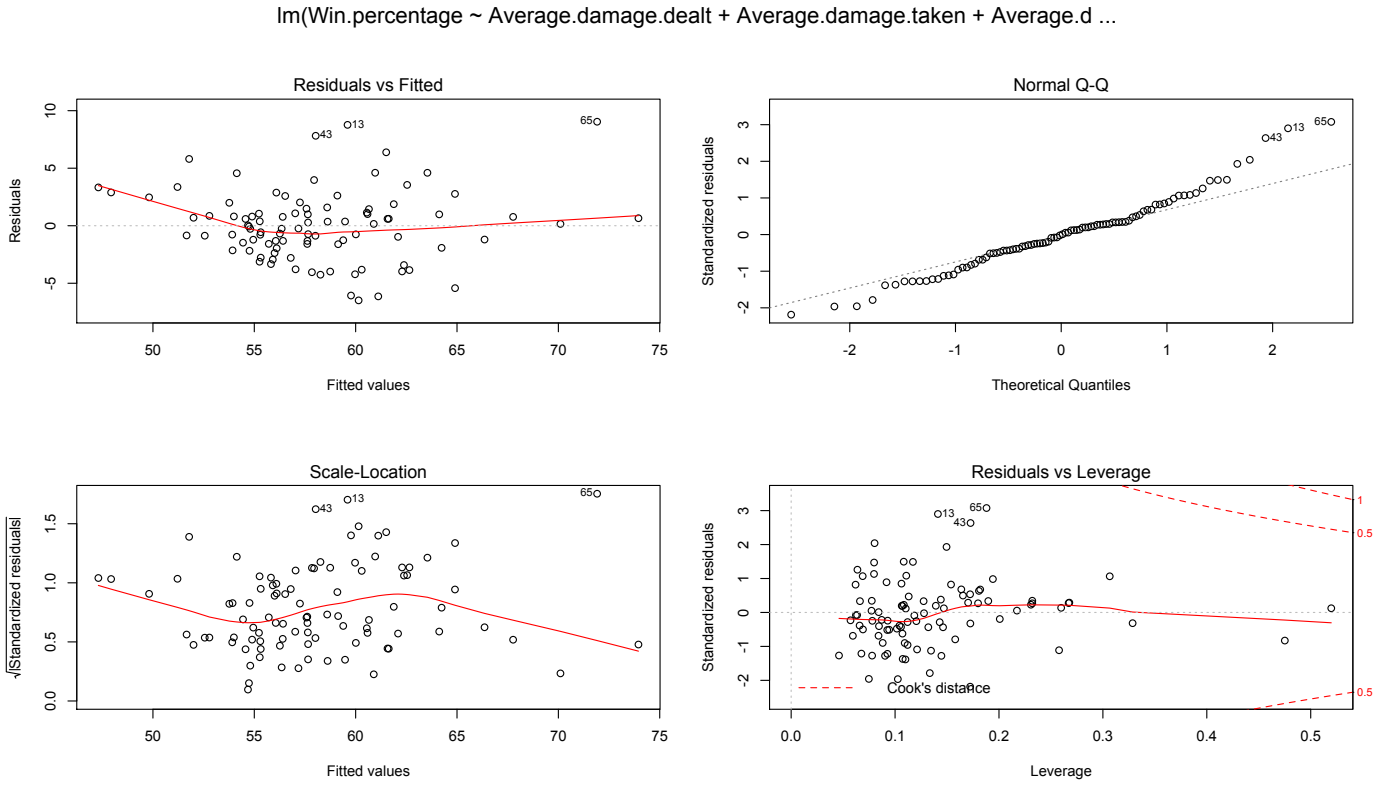
After conducting the F-test (F-statistic: 4.27; p-value: 0.0004629), it appears that the extra predictors and interactions are important, and are helpful in accurately predicting Win.percentage.

4 Results

Since we found both Linear.model.6 and Linear.model.7 to be effective predictors of Win.percentage, we turned to other methods to determine which model was *more* effective. While the BIC value was slightly better for Linear.model.6 and was a simpler model, Linear.model.7 proved to contain valuable predictors which Linear.model.6 does not include. Linear.model.7 is also able to explain about 10% more of Win.percentage's variability. For these reasons, I believe the best linear model for Win.percentage to be Linear.model.7, shown below.

$$
\begin{aligned}
Win.percentage \sim\ & 74.39 + 0.0002072(Average.damage.dealt) \\
& + 0.001613(Average.damage.taken) - 3.81(Average.deaths) \\
& - 0.000113(Average.magic.damage.done) \\
& - 0.1211(Average.minions.killed) \\
& - 0.2739(Average.neutral.minions.killed) \\
& + 6.229(Average.turrets.killed) + 0.794(Killing.spree) \\
& - 0.003316(Max.time.played) - 0.006037(Max.time.spent.living) \\
& - 0.000000008(Average.damage.dealt : Average.damage.taken) \\
& + 0.00007258(Average.magic.damage.done : Killing.spree : Average.kills)
\end{aligned}
$$

And from basic diagnostic plots, we can see our model assumptions are met.



5 Discussion

Since our final model is quite complicated, it may be useful to look at Linear.model.6 (a subset of Linear.model.7) to help identify the most important goals to pursue when playing a game.

After examining both models, it appears to me that the most significant objectives to pursue when in a game are damage dealt, deaths, and time spent living. The more damage you deal, the fewer deaths you have, and the longer you live, the more likely you and your team are to win the game. Looking deeper into our final model, it is also very important to focus on damage taken. This makes sense, because more damage taken usually results in more deaths, and we have already identified that factor as having a negative influence on win percentage.

The only issue with our model is that it is a bit complicated. In cases where a more concise model is needed, we can always use Linear.model.6, which tend to explain a bit less of Win.percentage's variability, but it is still effective and is a much simpler model that our final model. Regardless, both are very effective models that provide valuable insight.