Analysis of first 10 minutes of a *League of Legends* Game

Allen Lang

STAT 1601

Professor Afriyie

Spring 2021

Abstract

A dataset of diamond ranked games from video game *League of Legends* is analyzed. The objective is to discover factors that contribute to winning or gaining advantages in a single game. Analysis is done on dataset that provides statistics of the game state at the 10-minute mark. Variable pairs describing each team were combined into a singular difference variable that measured advantages that one team had over the other. Insights were found using data wrangling, data visualization, linear regression, multivariate regression, logistic regression, hypotheses test, and confidence intervals all performed on R. It was discovered that heralds are not as popular to take as dragons, and that red team takes dragons more often while blue team takes heralds more often. Finding measures of variation and center was not as interesting for the difference variables because the dataset was very balanced. Almost all of the averages were very close to 0. Logistic regression model was built to predict who would win the game. Gold, experience, jungle minion, towers, and lane minion differences matter the most for winning the game. While it made sense that gold, experience, and jungle minion advantages add chances to winning, it is surprising that lane minion and tower advantages decrease it. To test the balance of the dataset, a two-tailed hypothesis test showed inconclusive evidence that one team won more games than the other. This should be the case because players have no choice of which side they want to play on, so having the side impact the game would be unfair. Recommendations for future projects are to look at methods to incorporate categorical variables into the logistic regression, and combine insights from this dataset that focuses on the first 10 minutes to common metrics in professional games that focuses on the first 15 minutes.

Introduction

I analyze a dataset from the popular video game *League of Legends*. This report summarizes a semester-long project in which I apply concepts learned regarding R programming language and statistics. In this paper, the motivation for dataset selection will be outlined. A brief background into how *League of Legends* is played is then given. The introduction finishes with a description of the dataset and relevant variables as well as a few general research questions. The content portion will cover insights and observations obtained from data wrangling, data visualization, regression analyses, and statistical inferences. The conclusion section will wrap up the report, summarizing main takeaways, noting limitations in the study, and offering future extensions. References will contain the link to the dataset and the appendix will contain graphs and figures that complement the report.

Motivation

Last semester in CS 4774 Machine Learning, I made a time-series model to forecast US COVID-19 cases. This semester, I wanted to do analyze a dataset about something focused on leisure. I play video games in my free time, so I decided to find datasets about League of Legends, a game I have played sporadically for 10 years now. I play games like League of Legends because of its competitive nature, the goal is to win. From reading guides to watching professional players, I have spent a lot of time thinking about how to win a game of League of Legends. This is a chance to think about winning a League of Legends game in an alternative, quantitatively-focused way.

Background

The dataset I have chosen for the final project is from Kaggle and is named: “League of Legends Diamond Ranked Games (10 min). League of Legends is a MOBA (multiplayer online battle arena) game. Each League of Legends match consists of two teams (blue and red) that have 5 players each. Games typically last 30 minutes. Players on both teams select characters (champions) with unique abilities, and players control those champions in an effort to destroy the opposing team’s Nexus (base). Three lanes connect the two bases, and minions (lane minions) periodically spawn from each base and travel along these lanes; in between, there exists a jungle filled with hostile neutral monsters (jungle minions). During a match, players can kill enemy champions, enemy minions, enemy structures, and neutral monsters to gain experience and gold. Experience level up a player’s champion and makes them stronger. Gold can be used to purchase items, which augment players’ champions. The team with the best coordination and stronger players will lead to stronger champions and will most likely win in a fight against the other team. A single match concludes when one team destroys the other team’s Nexus, or one team surrenders.

Dataset Description

This dataset is a collection of a little around 9800 matches (observations) from Diamond tier. League of Legends has a ranked system, and Diamond tier is one of the most prestigious tiers; as of 2020, Diamond players represent the top 2% of all players in the North America server. For each match, there contain 19 statistics for each team for the first 10 minutes of each game, so 38 totals. Additionally, a unique match ID is given to each observation, and the 40th data column indicates which team ended up winning the game.

Variable Description

For Table I, italicized variables are the converted variables. With the exception of “TeamWin” variable, all other variables in Table I and Table II refer to the game state at 10 minutes.

Table I

Original and Converted Categorical Variables

Name	**Description**	**Motivation**
blueWins, *TeamWin*	Only variable that describes the whole match instead of first 10 minutes blueWins: 1 if blue team won the game, 0 if red team won TeamWin: shows which team won using characters: “blue” or “red”	Only variable that describes the entire match, and is important because it indicates who won.
blueFirstBlood, redFirstBlood, *FirstBlood*	First blood is the team that gets the first kill on an enemy champion blueFirstBlood: 1 if blue team got first blood, 0 if not redFirstBlood: 1 if red team got first blood, 0 if not FirstBlood: combined two above, shows which team got first blood using characters: “blue”, “red”, or “neither”	First blood kills are worth a little bit more than normal kills, so they can be important. Additionally, first bloods can be a confidence booster for the team that gets it and vice versa.
blueDragons, redDragons, *Dragon*	Dragons are neutral objectives with benefits to the team that kills it. Due to spawn and respawn rules, it is impossible for a team to kill two dragons before 10 minutes. Thus, a new column is made to describe who got *the* dragon before 10 minutes. blueDragons: 1 if blue team got it, 0 if not redDragons: 1 if red team got it, 0 if not Dragon: combined two above, shows which team got dragon using characters: “blue”, “red”, or “neither”	Killing dragons gives a team benefits, while killing multiple dragons will yield long-term benefits the longer a match goes. Being the first to kill dragon may mean a team is already stronger than the other.
blueHeralds, redHeralds	Rift heralds are neutral objectives with benefits to the team that kills it. Due to spawn and respawn rules, it is impossible for a team to kill two heralds before 10 minutes. Thus, a new column is made to describe who got *the* herald before 10 minutes. blueHeralds: 1 if blue team got it, 0 if not redHeralds: 1 if red team got it, 0 if not Herald: combined two above, shows which team got herald using characters: “blue”, “red”, or “neither”	Same rational as killing dragons. Rift herald’s reward helps destroy enemy structures instead of permanently strengthening a team, so it is seen as a short-term reward while dragon is long-term.

Table II features the original variables, as well as the difference variables after wrangling.

Table II

Original and Converted Numerical Variables

Name	Description	Motivation
blueTotalGold, redTotalGold, GoldDiff	Gold is currency used to purchase items that empower champions. Sources: Enemy champions, lane minions, jungle minions, enemy wards, enemy towers blueTotalGold: blue team total gold at 10 minutes redTotalGold: red team total gold at 10 minutes GoldDiff: blue team minus red team total gold	Gold is used to buy items, which makes players stronger. Gold difference is a very common metric used to quantify leads.
blueTotalExperience, redTotalExperience, ExpDiff	Experience points gain champions levels, making them stronger. Sources: Enemy champions, lane minions, jungle minions, enemy wards, enemy towers blueTotalExperience: blue team total experience at 10 min redTotalExperience: red team total experience at 10 min ExpDiff: blue team minus red team total experience	Experience gain can level up champions, which makes them stronger. In a way, every level can be also thought of as worth gold because of the boost in strength.
blueTotalMinions, redTotalMinions, MinionsDiff	Lane minions are a count of number of enemy minions killed. These minions travel down each of the 3 lanes. blueTotalMinions: blue team lane minions at 10 min redTotalMinions: red team lane minions at 10 min MinionsDiff: blue team minus red team lane minions	Lane minions are constantly coming into lanes, meaning they are a constant stream of income and experience. Teams with lots lane minions generally are strong.
blueJungleMin, redJungleMin, JungleMinDiff	Jungle minions are a count of number of neutral minions killed. These minions are found in the jungle between the 3 lanes. blueJungleMin: blue team jungle minions at 10 min redJungleMin: red team jungle minions at 10 min	The jungler takes the jungle minions and gets stronger. Since jungler can impact all 3 lanes, this could be indicative of team strength.

	JungleMinDiff: blue team minus red team jungle minions	
blueTowers, redTowers, TowersDiff	Towers are very strong structures that reward gold and experience. Early on in the game, they deal a lot of damage and serve as deterrent/guardian. blueTowers: blue team towers at 10 min redTowers: red team towers at 10 min TowersDiff: blue team minus red team towers	Towers grant vision, and can be thought of as an outpost. Destroying towers means gaining that territory as your own. Towers are durable, so any team that has managed to destroy a tower in the first 10 minutes must have had a very sizable advantage.
blueWardsPlaced, redWardsPlaced, WardsPlacedDiff	Wards are small items that grant vision on the map. Fog of war normally obscures vision on areas of the map without a team member there. blueWardsPlaced: blue team wards placed at 10 min redWardsPlaced: red team wards placed at 10 min WardsPlacedDiff: blue team minus red team wards placed	Wards grant vision, making people feel safer. It alerts the team of enemy team movements, reducing likelihood of death.
blueWardsDestroyed, redWardsDestroyed, WardsDestDiff	There are ways to deny vision of other team by destroying the other team's wards. blueWardsDestroyed: blue team wards destroyed at 10 min redWardsDestroyed: red team wards destroyed at 10 min WardsDestDiff: blue team minus red team wards placed	If wards grant vision and make people feel safer, then destroying enemy wards will make the enemy team uncomfortable and left doubting the movements of your team.
blueKills, redKills, KillDiff, blueAssists, redAssists, AssistDiff	Kill refers to defeating enemy champions, while assists go to team members who contribute but do not deal the killing blow. Kills offer the most gold and experience, while assists offer reduced amounts. blueKills: blue team champion kills at 10 min redKills: red team champion kills at 10 min KillDiff: blue minus red blueAssists: blue team champion assists at 10 min redAssists: red team champion assists at 10 min AssistDiff: blue minus red	Kills offer the most gold and experience, but are hardest to obtain. Many players tunnel vision on kill statistics and think that is most indicative of winning a game.

Research Questions

Some general questions that highlight the direction I want to take the analysis:

- What factors are most important for winning? What factors matter less?
- Is it easier to take dragon or herald? Does it depend on the team?
- Is spending time getting dragon or herald more valuable than getting kills and towers in lane?
- Is a team that is ahead leading in all other categories? What do they sacrifice?

Content

Data Wrangling

Data Cleaning

The reason why I chose not to convert any data types was because they had already imported in as integers or doubles since all of them were numerical but had no order (factor). I mentioned in the dataset description deliverable that I wanted to convert columns that had numerical quantities into categorical columns with character vector. This is because it more intuitive to see a column that says which team got an objective or won rather than seeing a 0 or 1. In the case of first blood, dragon, and herald that could have neither team get it, the viewer would need to look at two columns. I decided to remove unneeded columns such as gameID because are just unique identifiers for a game. Columns like redGoldDiff are just the inverse of blueGoldDiff. blueCSPerMin is just blueTotalMinions divided by 10 since every game only goes to 10 minutes. Finally, I decided to combine pairs of columns like blue kills and red kills into a difference because it is easier to work with. They are all consistent: blue statistic minus the red statistic. Positive number means blue team has more of it, and vice versa.

I did not find many interesting findings, except for the fact that there were no missing values in any of the columns. I also realized that all 10000 games had first blood occur before 10 minutes, there was not a single game in which all 10 players went deathless for 10 minutes.

Summary Measures of Key Variables

For key variables, I made frequency tables and two-way tables for all four categorical variables (Table III – VIII). I put dragon and herald together because they are both neutral objectives: one is on the top half and other is on the bottom half of the map. Both are frequently fought over in the first 10 minutes. For numerical variables, I felt that the most important variables were gold difference and experience difference because these are statistics that are hard for players to gauge during a game (Table IX). I also wanted to find out about minions killed because these are countable and are often used in game to gauge leads (Table X). I grouped these key numerical variables by each of the categorical variables.

I found that blue team and red team won similar number of games, which shows that blue or red (something out of control of player) has no effect on winning games (Table III). I found that teams that got first blood won their games 3 out of 5 times, or 3 to 2 (Table IV). Teams seem to like getting dragon more than herald, as the number of “neither” was larger for herald (Table VII). Red team seemed to get dragon more, but blue team seemed to get herald more (Table VI, Table VII). All the key numerical variables were quite even, there were marginal differences. When grouped, all key numerical variables favored the team that ended up winning/got the objective (Table XI – XIII). For teams that won, the average gold and experience they got at 10 minutes was 17000 and 18000, respectively (Table XIV).

Filtering

For filtering, I decided to see the impact of towers on the game because towers lead to shifts in territory. I also wanted to see the impact of the 17000 gold and 18000 experience threshold I found from

the previous section. Since Smite is used to safely secure neutral objectives like dragon and herald and higher levels means smite deals more damage, I wanted to see if teams with level advantages took dragons more than heralds because their smite could deal more damage than the other team's.
I found that, if a team had just a one tower advantage over the other team, then said team would have a very large difference in gold, experience, and other key metrics (Table XV). While the ratio for first bloods resulting in a win was 3 to 2, this ratio was 3 to 1, showing taking towers is much more important than a single kill (Table XVI). For the threshold, teams usually have a 227 minion kills, or a 20-minion advantage over their opponent (Table XVII). The ratio of winning is 10 to 3 (Table XVIII). For level difference, the advantage was more apparent in dragons than in heralds, as a lot of teams still decided to ignore the herald in first 10 minutes (Table XX – XXI). The biggest difference lay in the jungle minion difference, as teams with higher smite power can more easily farm the jungle (Table XIX).

Data Visualization

Single Categorical or Numerical Variables

For first bloods, shows the blue team to have a slightly higher percentage than that of red team. The difference is small, but blue team may get first bloods easier (Figure 1). For dragon, both teams want to take it before 10 minutes (3/4 of the games), with red team taking more than the blue team (40% to 36%) (Figure 2). For herald, blue team gets rift herald slightly more often than red team, but over 60% of the games have neither team taking it (Figure 3). This could suggest that both teams value rift herald less than dragon, or there is not enough time because herald spawns at 8 minutes. The Gold differential and the experience differential histograms show similar distributions (Figure 4 – 5). Both are symmetric in shape with little outliers. The mean, median, and mode appear to be around 0, so neither team having an advantage. A minor difference is spread. For gold, the spread ranges from -10000 to around 10000, but for experience it ranges from -10000 to 8000. The red team appears to have more games in which they are very far ahead (-8000 to -1000) than the blue team as the left side of the graphs have little bit more bars than the right side.

Categorical/Numerical or Numerical/Numerical

For the team that got first blood, the gold diff and exp diff would favor them at 10 minutes. However, the median difference from 0 for gold diff was larger than the median difference from 0 for exp diff (Figure 6 -7). This suggests that first blood might impact the gold gains more than the experience gain. For teams that got dragon or herald first, they also get a gold advantage at first 10 minutes (Figure 8-9). For games where neither team took dragon or herald, the median gold difference was actually 0. This makes sense as an even game makes it hard to take objectives. However, if teams get a neutral objective, then they have a gold difference. It is not clear about causation, so it cannot be said whether gold differences make dragons/heralds easier to take or vice versa. A numerical variable that seems to have a very even spread is the wards placed variable (Figure 10). The interquartile range appears to be very small as both teams seem to place equal number of wards. There are a lot of outliers. In gold/experience difference and total minion/gold difference scatter plots, there is a strong positive correlation between the two (Figure 11 - 12).

Three or more variables

When plotting gold and experience differences with the team that won as differentiating color, teams with a lead at 10 minutes generally went on to win the game (Figure 13). More red dots were in the bottom left (red team exp and gold advantage) and more blue dots were in the top right (blue team advantage). A similar but less clear trend was shown when differentiating based on who got herald or dragon (Figure 14- 15). Some games saw teams not take the neutral objective despite one team having a big advantage. Very few teams with massive disadvantages secured the dragon or herald. The team that took dragon and herald had a slight lead in jungle minions taken, although the correlation is not as clear for either graphs (Figure 16 – 17). Teams with gold and experience advantages also had noticeable minion kill advantages (Figure 18). When plotting total minions and jungle minion differences, the correlation is also not strong. The wards placed and destroyed differences plot has an odd shape (Figure 19). As the games with little to no ward placed differences had the greatest number of wards destroyed difference (Figure 20). This means teams placed equal amounts of wards, but somehow one team was sweeping and destroying them while the other did not. There seems to be no correlation with either and the team that won the game in the end.

Special plots

Pie charts provided clearer information than bar charts in section one on dragons and heralds (Figure 21 – 22). When looking at a heatmap and towers destroyed, the largest tower difference seemed to be when neither dragon nor herald was taken (Figure 23 - 24). It also makes sense that the dark blue would favor the left (red team destroy more towers) and vice versa. The shades of blue also seem more accentuated at either end for the herald graph compared to the dragon graph. Herald does help take towers.

Regression

Linear Regression

With the correlation coefficient to be almost 0.9, the relationship between GoldDiff and ExpDiff variables is implied to be a strong positive linear relationship. I make a simple linear regression model, with GoldExp as dependent variable (y) and ExpDiff as independent variable (x).

$$GoldDiff = 1.143 * ExpDiff + 52.843877$$

There is a positive slope, meaning gold difference goes up as experience difference goes up and vice versa (Figure 25). The R squared value is about 0.8, which means about 80% of the variability in GoldDiff is explained by the ExpDiff variable. Based on the graph, I expected the R squared value to be higher, but this still shows a strong model. I made up experience differences of 2000, 3000, and -5000 and got gold predictions of 2338, 3482, and -5662 (Figure 26). Notably, the prediction interval is quite large, with lower and upper bounds having a difference of 2000 from the fit value. This suggests volatility in the model. When I made a model between GoldDiff and MinionsDiff, the model was worse, getting R squared of 0.4.

Multivariate Regression

Since kills (k), assists (a), towers (T), minions (M), jungle minions (J), wards (w), etc. all give gold to teams, I built a model, and AIC decided to keep all 6.

$$\Delta\ g = 1097.82\Delta T + 5.5746\Delta w + 374.29\Delta k + 52.85\Delta a + 22.14\Delta M + 23.08\Delta J + 21.1.$$

All coefficients are positive, having more increases gold difference. Towers (whole team) and kills (single player) give a lot of gold, so it makes sense that they have the highest coefficients. Assists give reduced gold. Jungle and normal minions are steady stream of gold income, but worth less than kills and towers. It makes sense that wards have smallest coefficient because they give less gold than minions and they are destroyed less frequently. The R squared value of 0.9511 means that 95% of the total variance in gold difference is explained by these variables. The adjusted R square value, 0.9511, means that the model will produce predictions that are good 95% of the time (Figure 27). Two predictions were made. One game is when blue team had more kills but red team had more towers. Blue team has a slight lead (688), but the lower bound is -378 while the upper bound is 1755. Another game is when blue team has more kills while red team has farmed more minions. Despite 100 more minions, red team just has a small lead (-455) while the upper bound (600) shows the blue team still having a lead regardless (Figure 28).

Logistic Regression

I made two models. One was predicting who would win. I used differences in gold, experience, towers, minions, jungle minions, and wards placed, wards destroyed, kills, assists and the AIC model removed the last four (Figure 29). I made another model to scale the variables to make the exponentiated coefficients clearer. Gold (4.8% per 100 gold), experience (2.2% per 100 gold), and jungle minion (4.4% per 10 minion) advantages increase chances of winning (Figure 30). Surprisingly, minions (4% per 10 minions) and tower advantages (22% decrease) actually decrease chances to win. My first prediction gave blue team a 61% chance to win when they had advantages in all 5 variables. The second prediction has blue team with pretty good tower, minion, and jungle minion advantages, but deficits in gold and experience give blue team a 16% chance. The third game is actually from a professional game between FPX and IG. IG, the blue team, trailed at 10 minutes, but still won despite the model giving them a 22% chance to win. The second model was to predict which team got the first dragon. I used same 9 variables, but AIC kept all of them this time (Figure 31). Every variable contributed positively towards a team's chances of getting the dragon before 10 minutes except for gold and experience but I did not scale them. Notable percent increases were towers (44%) and kills (43%) (Figure 32). First game, blue team leads in every category and has a 97% to take dragon. Interestingly, the second game has blue team having deficits in all variables except for jungle minions and minions. The model still favors blue team to take the dragon. The third game is still referring to the FPX vs. IG game. I mentioned IG was down by 10 minutes, and had only a 31% chance to get dragon. Right after the 10-minute mark, FPX managed to take the dragon.

Inference

Proportions

Team side should not matter because players do not get to choose this before a game. Ideally, removing all other factors, someone should have an equal chance to win (50%) regardless of blue or red. I made a hypothesis test, with a significance level of 0.05. The null hypothesis is the win rate for blue team being equal to 50% (fair 50/50 chance). The alternative hypothesis is the win rate for blue team not equal to 50% (one side gives benefits). The p-value is 0.8563, which is larger than the significance level. I fail to reject the null hypothesis. We do not have statistical evidence to conclude that the chance to win a game as the blue team is not 50%. There is a chance that the null is true, that the win rate is 50%. This makes sense since a fair game would not let blue or red side choice influence the outcome. I made a confidence interval for dragons to estimate the games in which blue, red, or neither team got dragon. We are 95% confident that the true percentage/proportion of all diamond league games in which the blue team took dragon before 10 minutes is between 0.3525 and 0.3716. For red team, we are 95% confident it is between 0.4034 and 0.4229. For neither team, it is between 0.2167 and 0.2333.

Means

I found confidence intervals for red and blue team specific statistics, not differences. Since they were very similar, I will only discuss blue team’s, although both are on the HTML. Gaining level is through accruing experience points. However, levels are easier to interpret because they are integers: levels go from 1 to 18. I found confidence intervals for blue and red teams' average levels. We are 95% confident that the true mean of blue team average levels at 10 minutes of all diamond league games is between 6.909986 and 6.922022. The average level is almost level 7. The upper and lower bounds for the two confidence levels are very small. For gold, we are 95% confident that the true mean of blue team total gold at 10 minutes of all diamond league games is between 16473.17 and 16533.74. Dividing it by 5 players, this means the average gold per player is a little over 3000. For comparison, a complete, end-game item is generally around 3500 gold in price. For lane minions, we are 95% confident that the true mean of blue team total lane minions at 10 minutes of all diamond league games is between 216.2685 and 217.1307. Dividing the total by 3 (two players are in roles that do not get lane minions), the average per player estimate of minion count at 10 minutes is a little more than 70. Perfect play means a rate of 10 minions per minute, or 100 by 10 minutes. Since diamond plays are very good, I expected it to be a bit closer to 100, but there are incentives for not getting minions (helping take dragon or herald, helping other lanes).

Conclusion/Discussion/Limitations

In this analysis, I looked at how different factors in the game at 10 minutes influenced the outcome of the game through wrangling, visualization, regression, and statistical inference. Teams took dragon more frequently than herald. Red team took more dragons while blue team took more heralds. While kills are often looked at as important objectives, teams that led in towers destroyed at 10 minutes had a higher win percentage than teams that took first blood. The averages and standard deviation of the difference variables were all very close to 0. This means that the dataset of games was very even because it had games where the blue team was winning and games where the red team was winning. The largest tower difference came when neither herald nor dragon was taken, which implies that dedicating resources to taking towers means sacrificing time that could be used to take herald or dragon. The gold and experience difference had one of the highest correlation coefficients, and a linear regression model produced a model with an R^2 value of 0.8. A multivariate regression model was built using towers, wards destroyed, kills, assists, lane, and jungle minions to predict the gold difference. It produced an adjusted R squared value of 0.9511 because these variables all give gold to a team. Logistic regression models were made to predict winning the game. Gold, experience, jungle minion, towers, and lane minion differences matter the most for winning the game. While it made sense that gold, experience, and jungle minion advantages add chances to winning, it is surprising that lane minion and tower advantages decrease it. A two-tailed hypothesis test showed inconclusive evidence that one team won more games than the other. The average level of diamond games at 10 minutes falls close to level 7.

Limitations to the study involves looking at the differences variable. Future extensions could look at how to use the individual statistics (blueTotalGold and redTotalGold) together for a suitable analytical framework. Another limitation is that most analytical league data sites for professional games have statistics at the 15-minute mark, instead of the 10-minute mark. This means it is difficult to combine the results of this study with other sites because the game state can always change between 10 and 15 minutes. Another limitation is the logistic regression model only took numerical variables, and did not take into account factors like first blood, dragon, or herald. Extensions would be to learn about logistic regression models or preprocessing that can also accept categorical variables. A final limitation is that the champion selection phase before the game is also important. Some champions are strong early and weak later on while others are weak early and strong later on. Game advantages in the first 10 minutes might not matter if the enemy team is much stronger later into the game. Possible extensions to this are difficult, as the champion selection phase adds more complexity to the analysis. Like a game in any other sport, predicting and analyzing *League of Legends* games is difficult and must factor in many factors. Quantitative analysis is just one factor to take into account.

References

Dataset: https://www.kaggle.com/bobbyscience/league-of-legends-diamond-ranked-games-10-min

Spawn timers for Dragon: https://leagueoflegends.fandom.com/wiki/Dragon_pit_(League_of_Legends)

Spawn timers for Herald: https://leagueoflegends.fandom.com/wiki/Rift_Herald_(League_of_Legends)

Invictus Gaming vs. FPX Game 2 Semifinals Worlds 2019: https://gol.gg/game/stats/20485/page-game/

Professor Afriyie's Lecture Videos

R Studio documentation: https://docs.rstudio.com/

Appendix

Table III

Team that won the game

```
## Blue  Red
## 4930 4949
```

Table IV

Team that got First Blood

```
## Blue  Red
## 4987 4892
```

Table V

Two way table for first blood and winning game

```
##        FirstBlood
## TeamWin Blue  Red
##    Blue 2987 1943
##    Red  2000 2949
```

Table VI

Team that got dragon

```
##    Blue Neither     Red
##    3576    2222    4081
```

Table VII

Team that got herald

```
##    Blue Neither     Red
##    1857    6441    1581
```

Table VIII

Two way table for dragon and herald

```
##           Herald
## Dragon     Blue Neither  Red
##   Blue      710    2323  543
##   Neither  338    1576  308
##   Red       809    2542  730
```

Table IX

Mean and standard deviation of gold and experience differences

```
##   mean(GoldDiff) sd(GoldDiff) mean(ExpDiff) sd(ExpDiff)
## 1       14.41411     2453.349     -33.62031     1920.37
```

Table X

Mean and standard deviation of lane minion and jungle minion differences

```
##   mean(MinionsDiff) sd(MinionsDiff) mean(JungleMinDiff) sd(JungleMinDiff)
## 1        -0.6496609        30.94267          -0.8034214          14.27473
```

Table XI

Team metrics based on who won the game

```
##   TeamWin `mean(GoldDiff)` `mean(MinionsDiff~ `mean(ExpDiff)` `mean(WardsP
laced~
## * <chr>              <dbl>              <dbl>           <dbl>
<dbl>
## 1 Blue               1271.               8.93            908.
0.360
## 2 Red               -1237.             -10.2            -972.
-0.517
```

Table XII

Team metrics based on who got the dragon

```
##   Dragon  `mean(GoldDiff)` `mean(MinionsDif~ `mean(ExpDiff)` `mean(WardsPl
acedD~

## * <chr>              <dbl>              <dbl>          <dbl>
<dbl>

## 1 Blue               776.              3.61           506.
1.01

## 2 Neither             48.3             0.0320          18.7
-0.279

## 3 Red               -671.             -4.75          -535.
-0.924
```

Table XIII

Team metrics based on who got the herald

```
##   Herald  `mean(GoldDiff)` `mean(MinionsDif~ `mean(ExpDiff)` `mean(WardsPl
acedD~

## * <chr>              <dbl>              <dbl>          <dbl>
<dbl>

## 1 Blue              845.               5.25           615.
0.0819

## 2 Neither             3.35            -0.748          -41.2
0.221

## 3 Red              -916.              -7.18          -765.           -
1.49
```

Table XIV

Team total gold and experience at 10 minutes when they won

```
##   TeamWin `mean(blueTotalG~ `mean(blueTotalE~ `mean(redTotalG~ `mean(redTo
talEx~

## * <chr>               <dbl>              <dbl>            <dbl>
<dbl>

## 1 Blue               17145.             18405.           15875.
17496.
```

```
## 2 Red                15864.            17453.           17101.
18425.
```

Table XV

Key metrics when blue team has more towers destroyed

```
##   mean(GoldDiff) mean(MinionsDiff) mean(ExpDiff) mean(WardsPlacedDiff)
## 1         3363.2          26.70975      1844.909            -0.5941043
```

Table XVI

Who won the game when blue team had tower advantage at 10 minutes

```
## Blue  Red
##  335  106
```

Table XVII

Minion metrics when blue team met the threshold

```
##   mean(blueJungleMin) mean(blueMinions) mean(redJungleMin) mean(redMinions
)
## 1            52.95157          226.3784           49.53451         207.114
9
```

Table XVIII

Who won the game when blue team met the threshold

```
## Blue  Red
## 2108  659
```

Table XIX

Jungle minion difference when blue team has higher level

```
##   mean(blueJungleMin) mean(redJungleMin)
## 1            54.48432           44.00697
```

Table XX

Dragons taken when blue team has higher level

```
##    Blue Neither     Red
##     167      71      49
```

Table XXI

Heralds taken when blue team has higher level

```
##    Blue Neither     Red
##      77     188      22
```

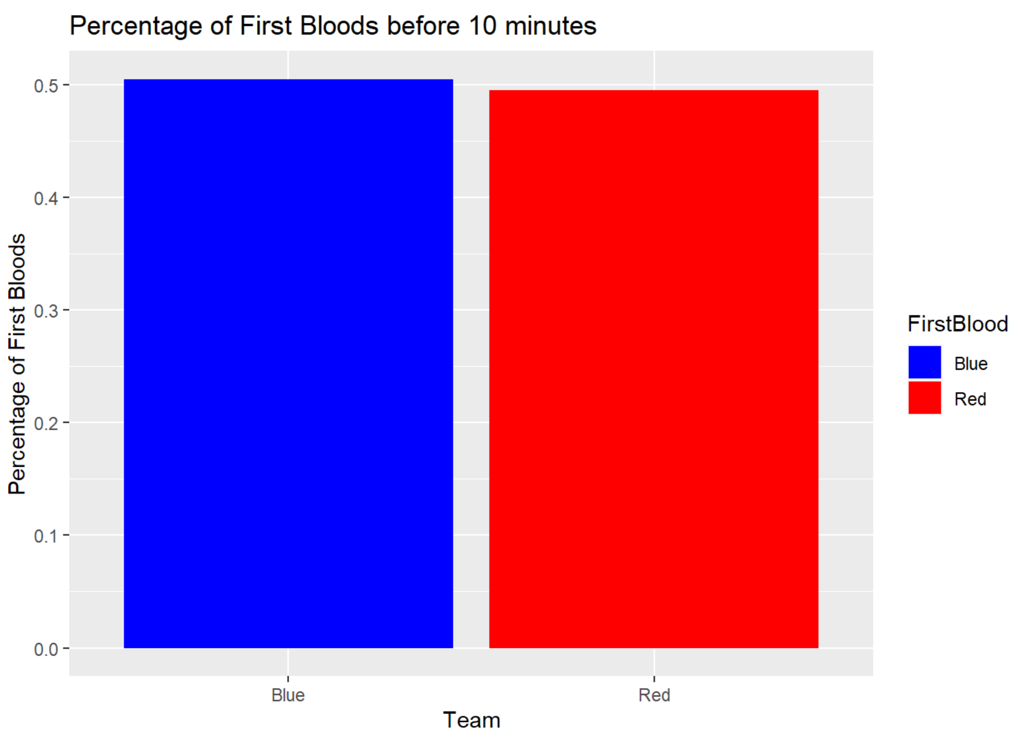

Figure 1. Percentage of first bloods to each team.

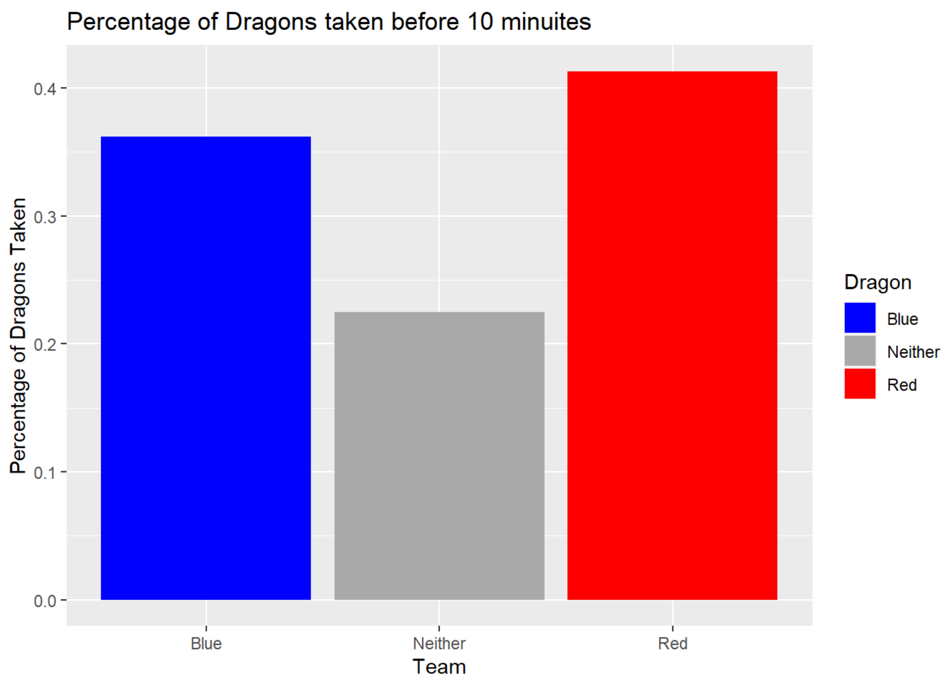

Figure 2. Percentage of dragons to each team.

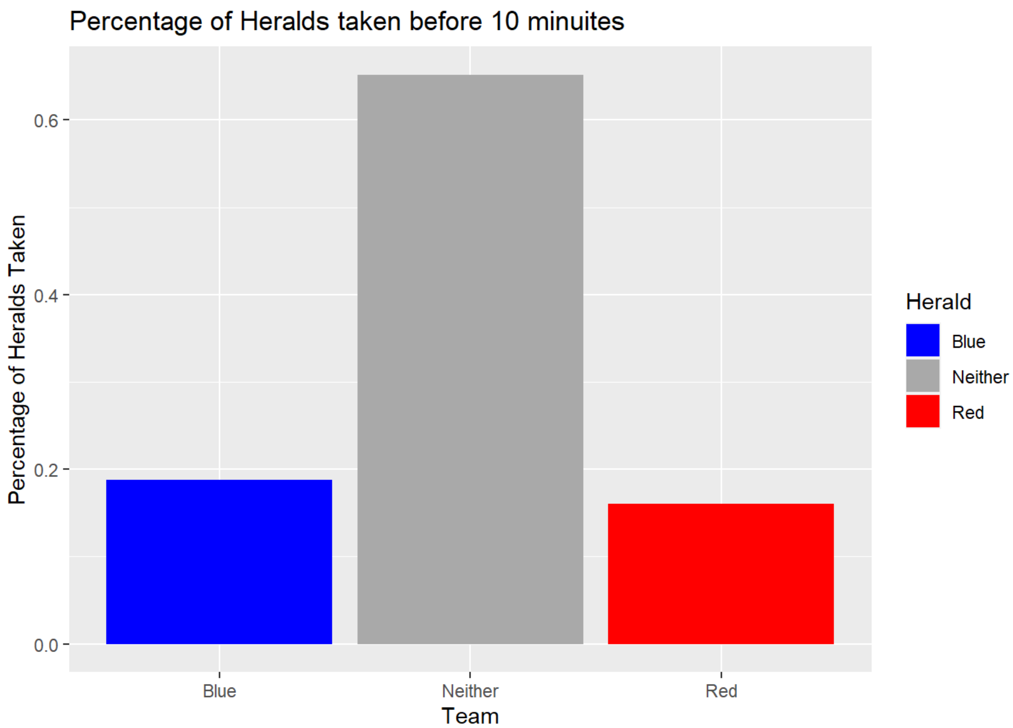

Figure 3. Percentage of heralds to each team.

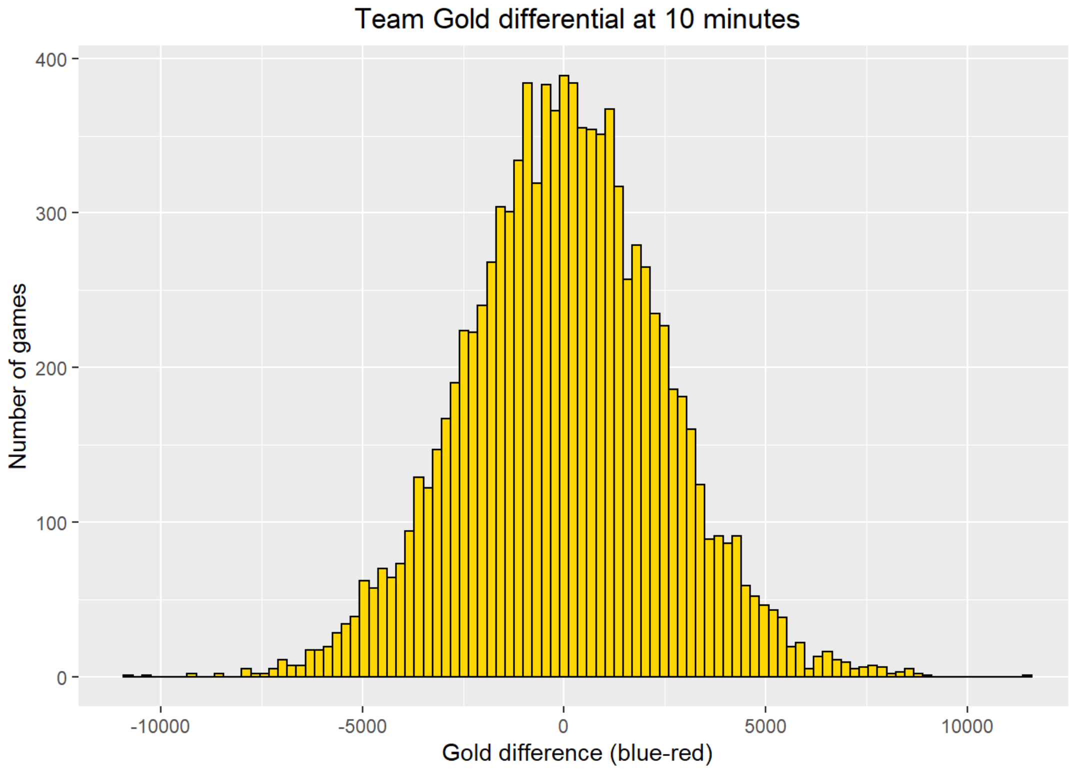

Figure 4. Gold difference distribution

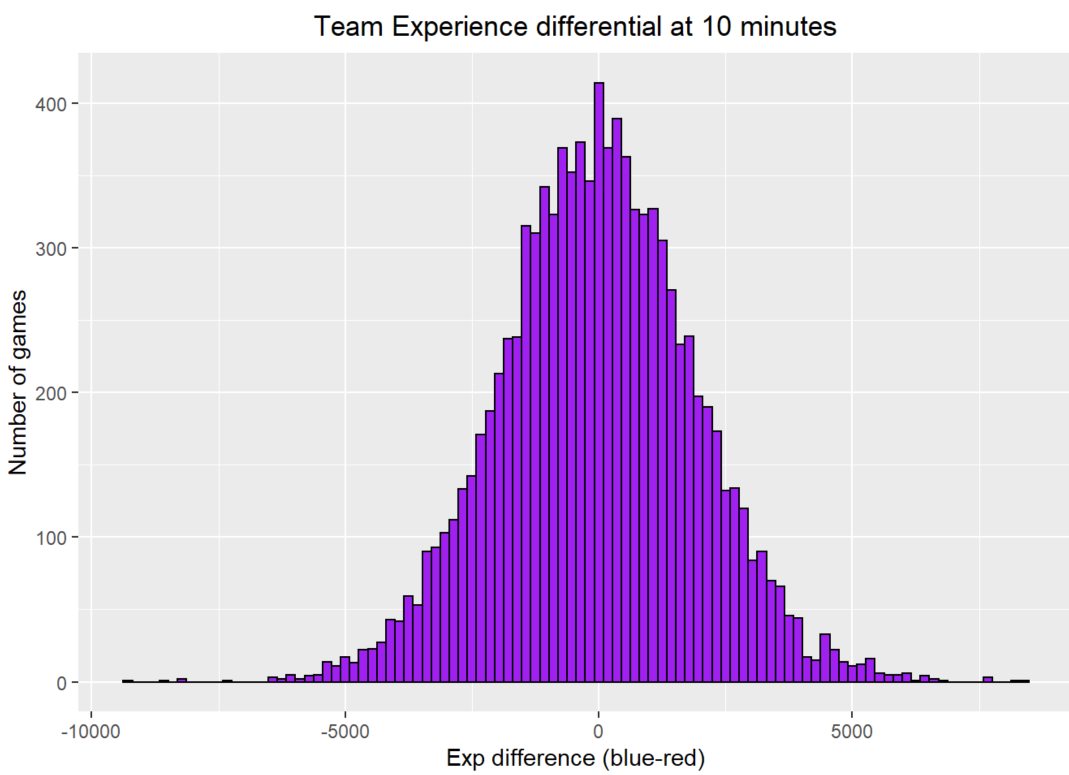

Figure 5. Experience difference distribution.

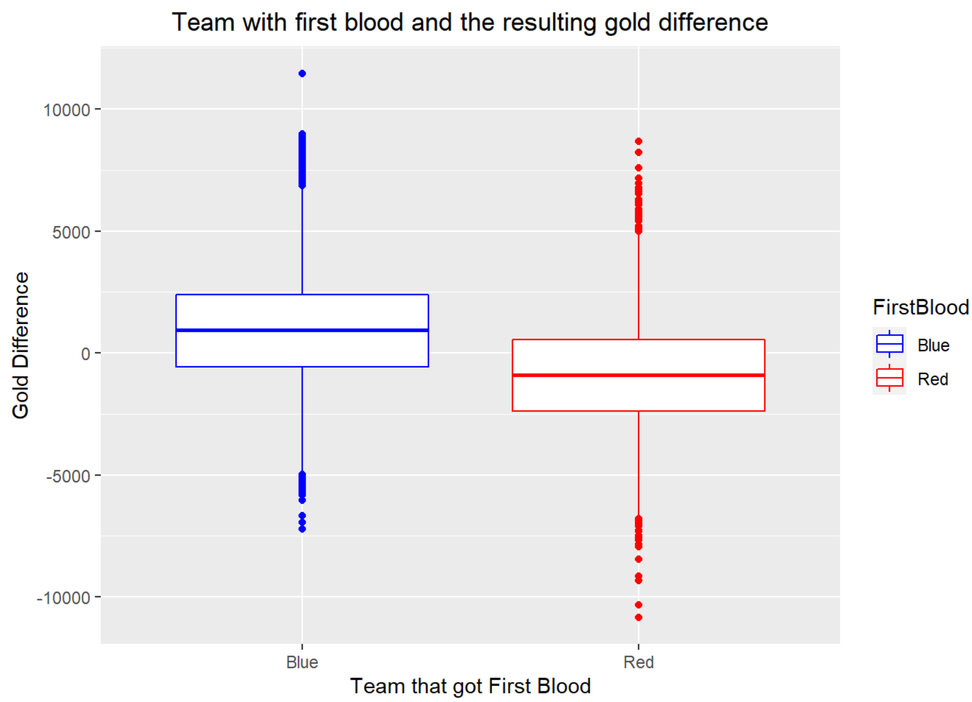

Figure 6. Gold difference when teams get first blood.

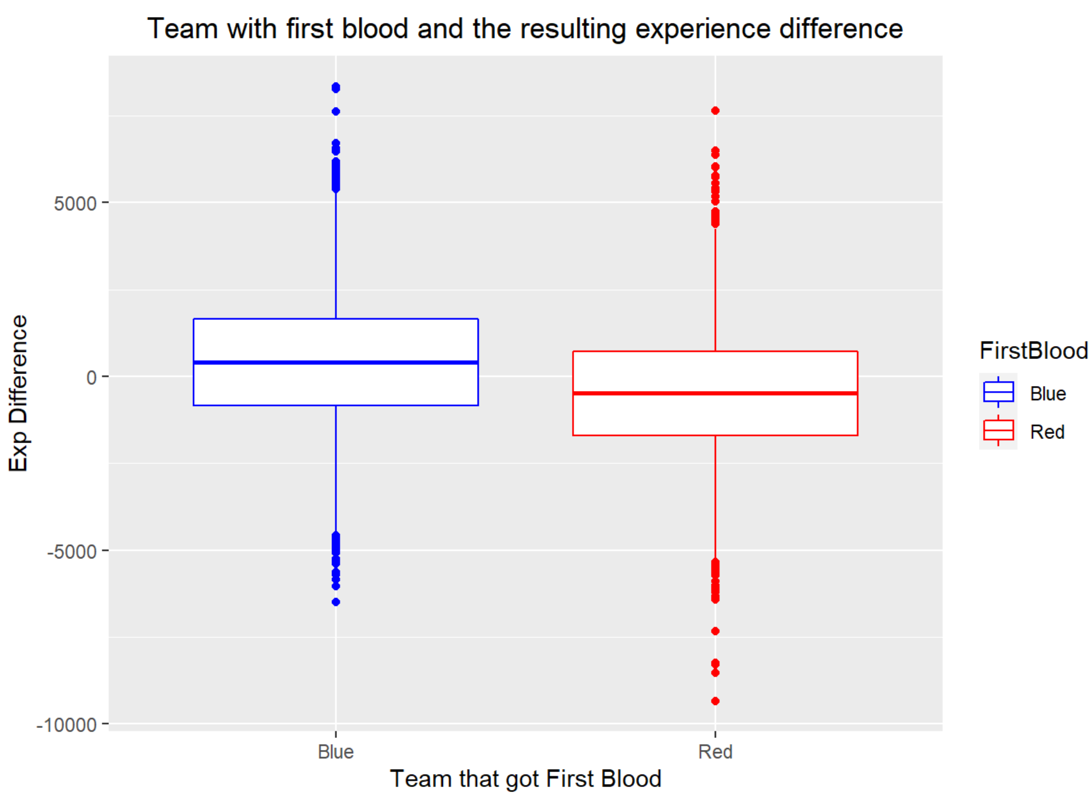

Figure 7. Experience difference when teams get first blood.

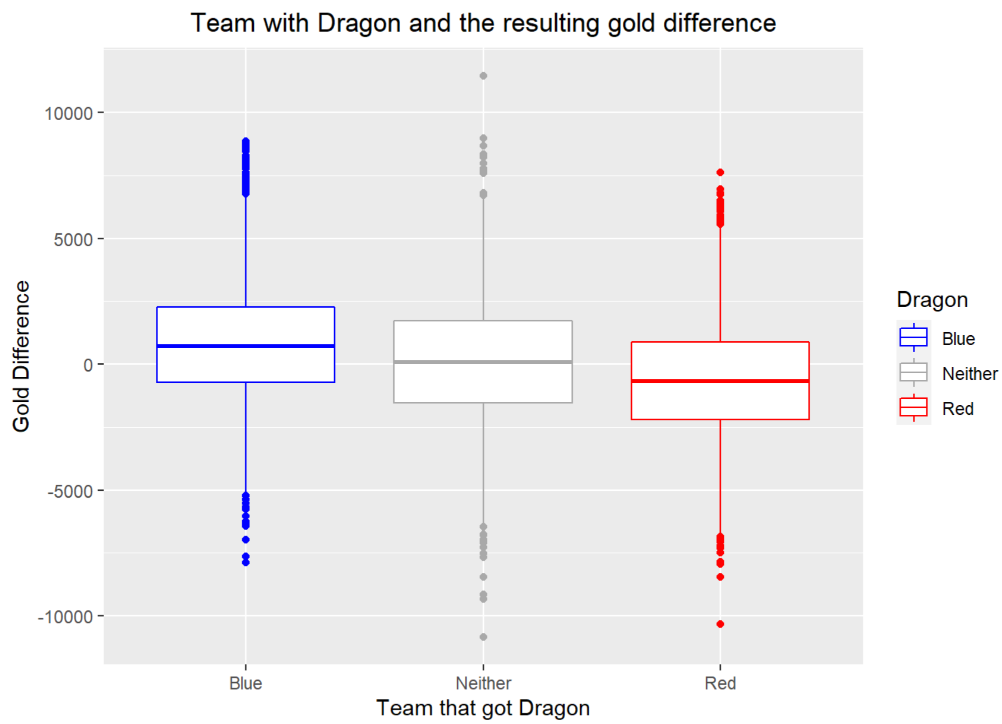

Figure 8. Gold difference when teams get dragon

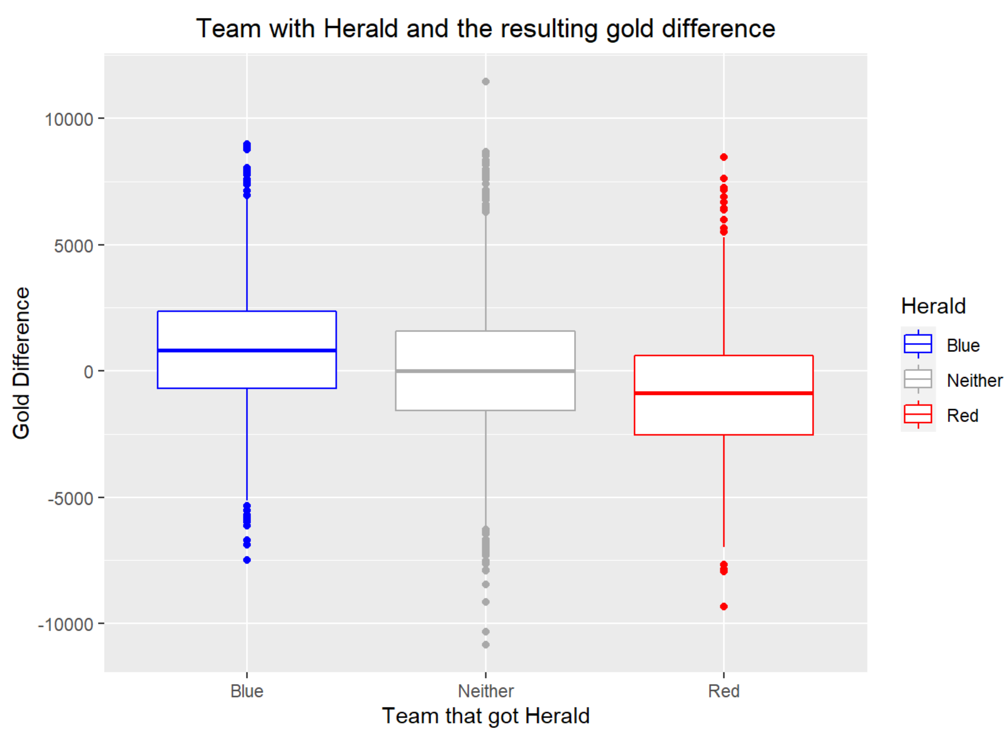

Figure 9. Gold difference when teams get herald

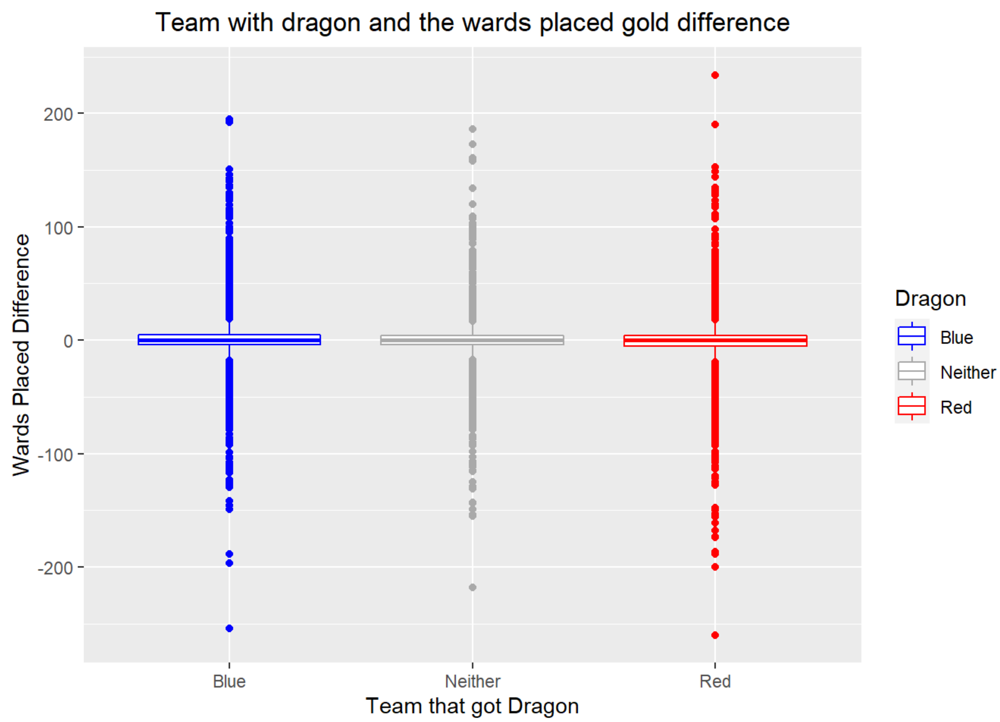

Figure 10. Wards placed and teams that got dragon

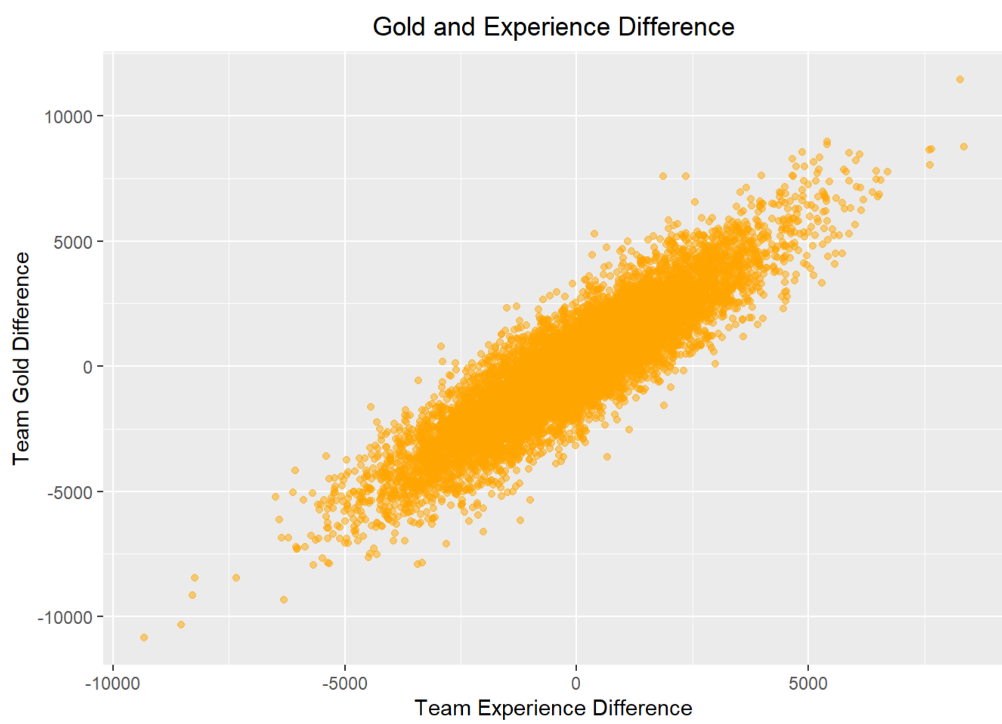

Figure 11. Gold and experience difference

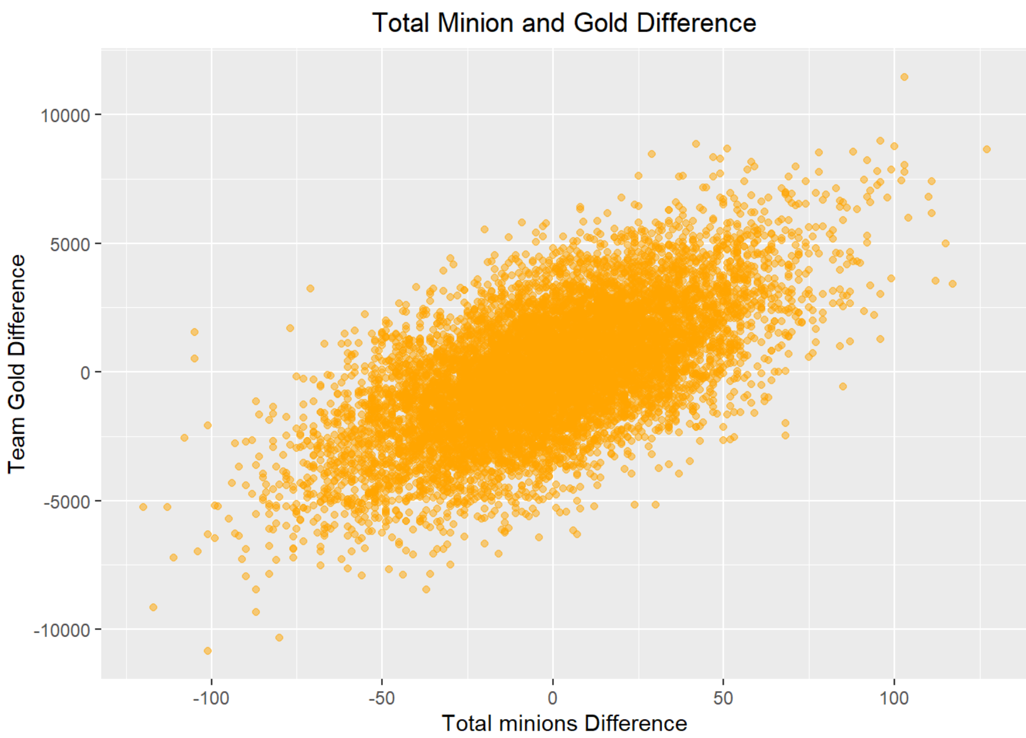

Figure 12. Lane minion and gold difference

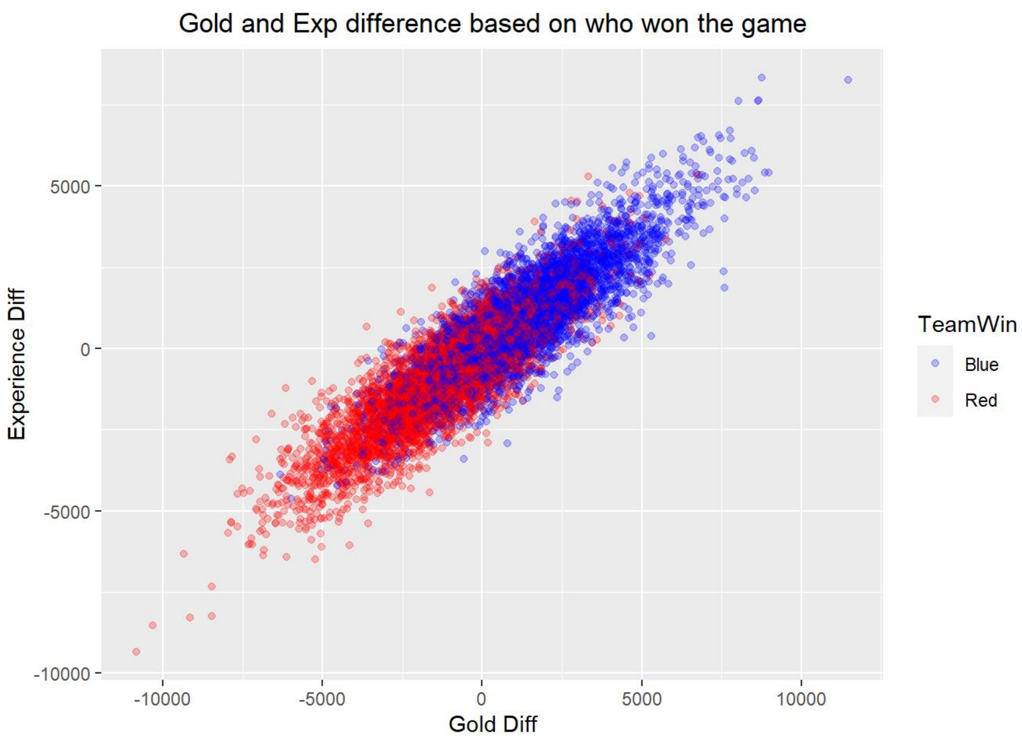

Figure 13. Gold and experience difference and who won the game

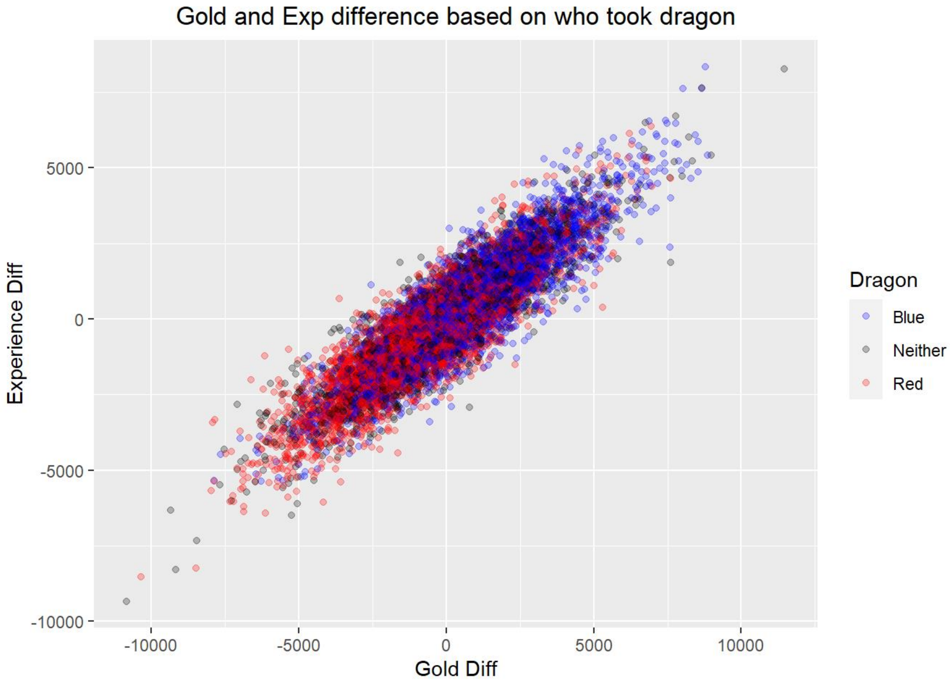

Figure 14. Gold and experience difference and who took dragon

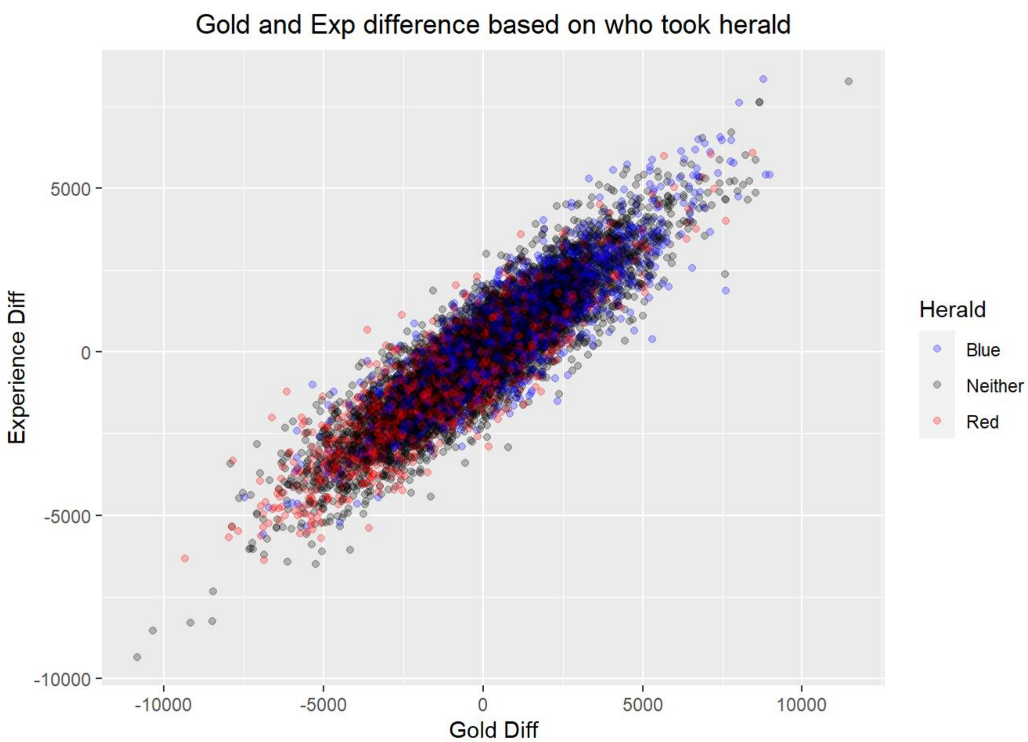

Figure 15. Gold and experience difference and who took herald.

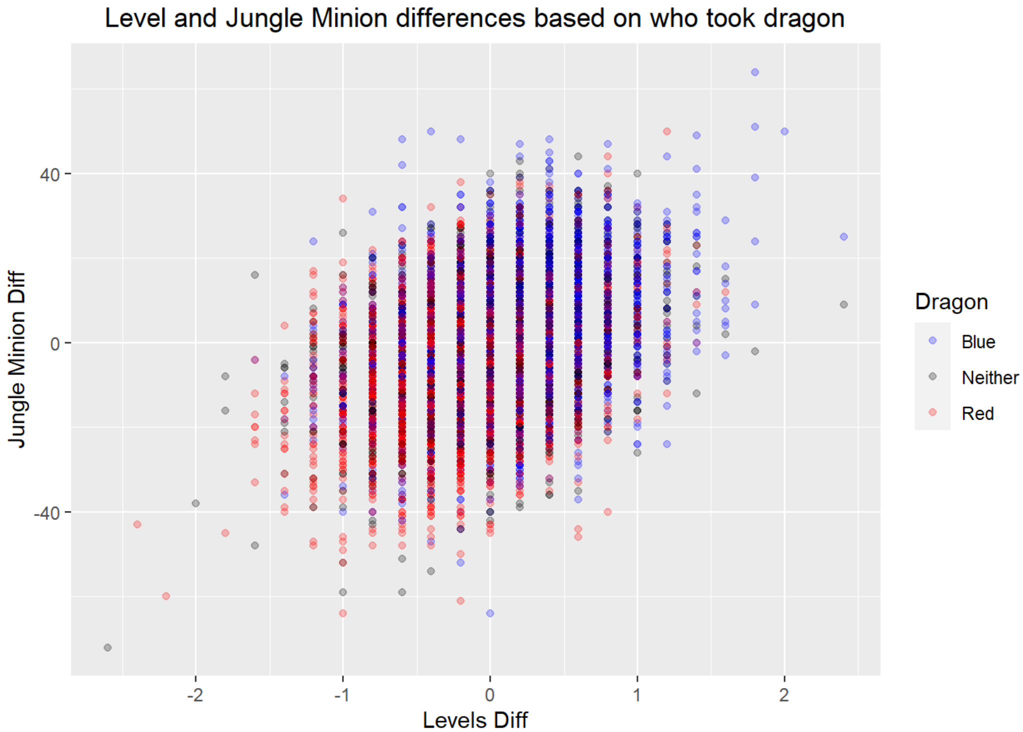

Figure 16. Level and jungle minion difference and who took dragon.

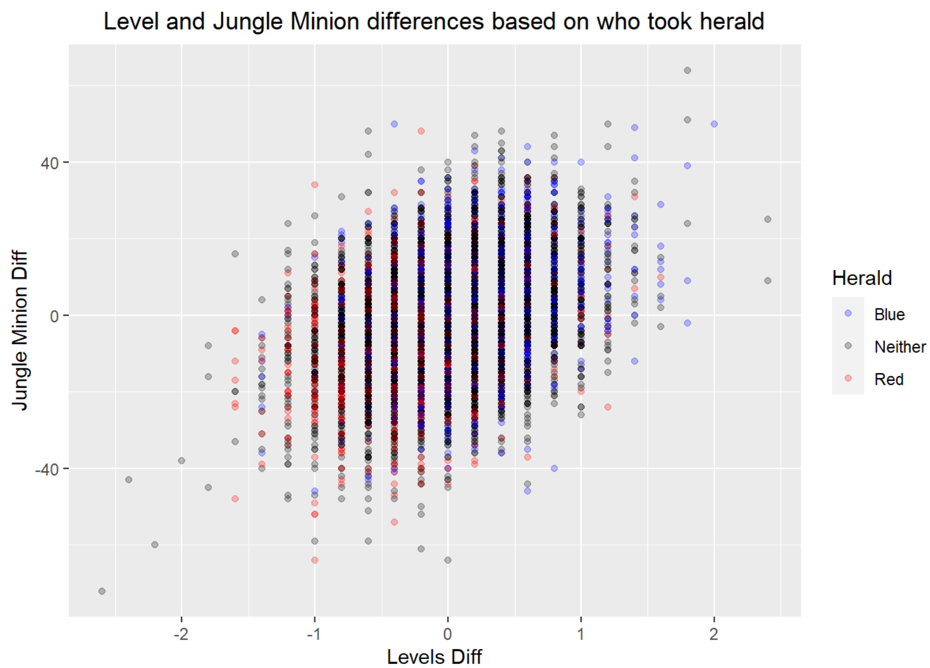

Figure 17. Level and jungle minion difference and who took herald.

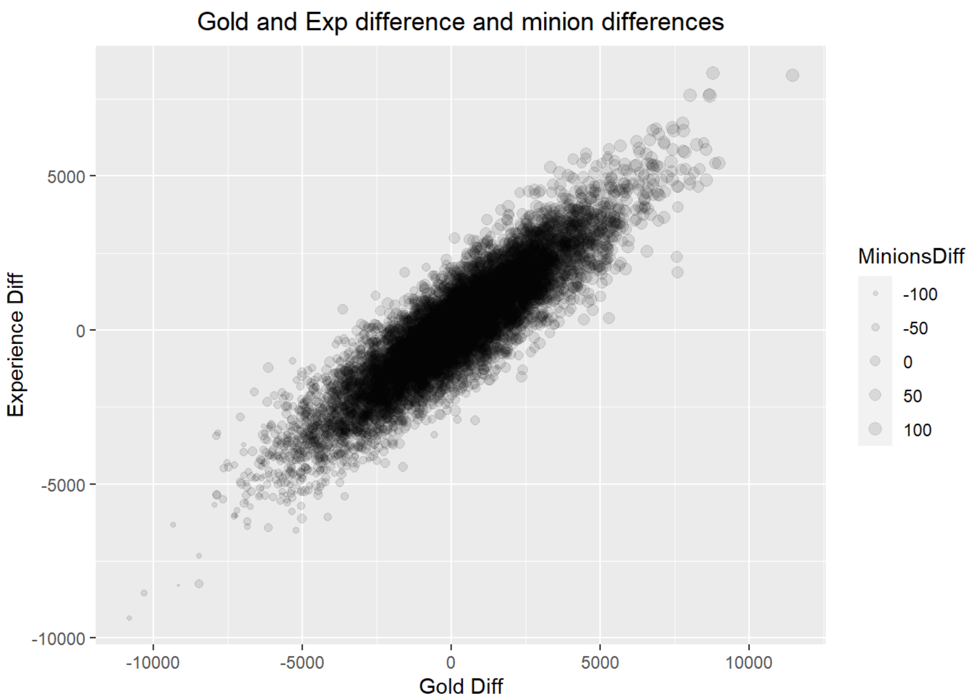

Figure 18. Experience and gold difference based on the difference in lane minions

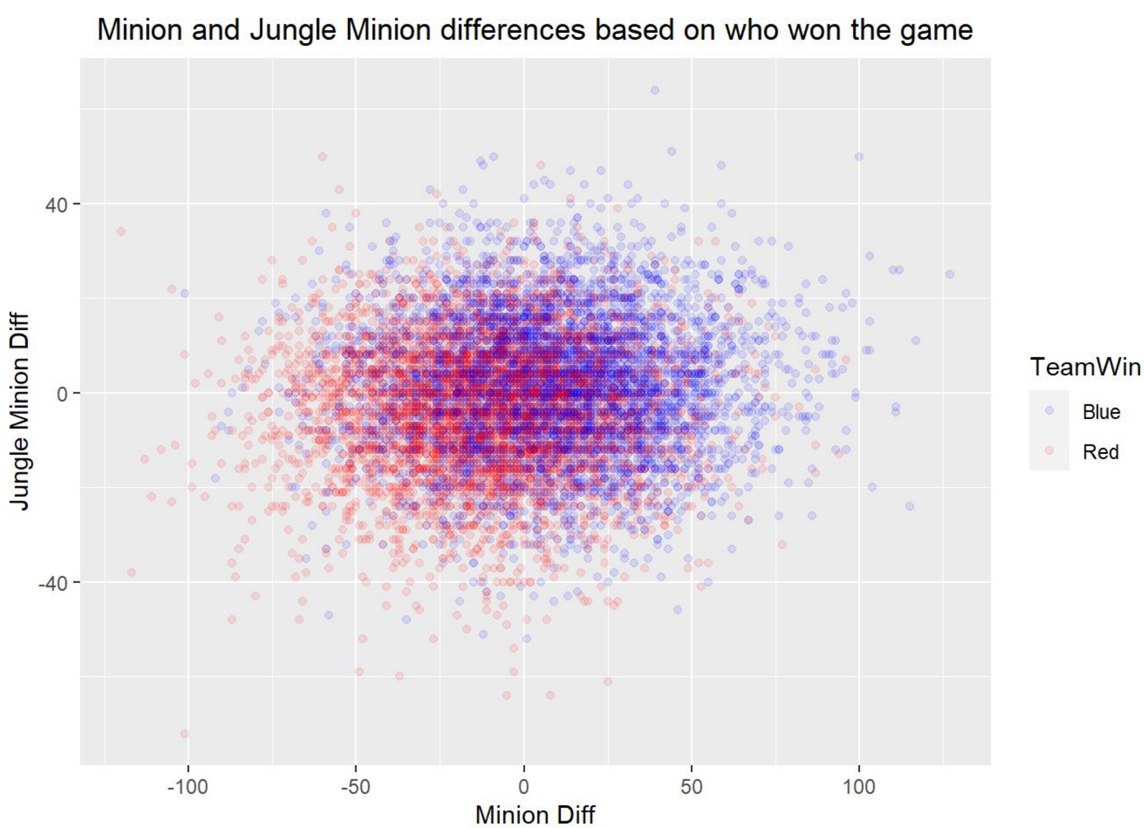

Figure 19. Lane and jungle minion difference based on who won the game

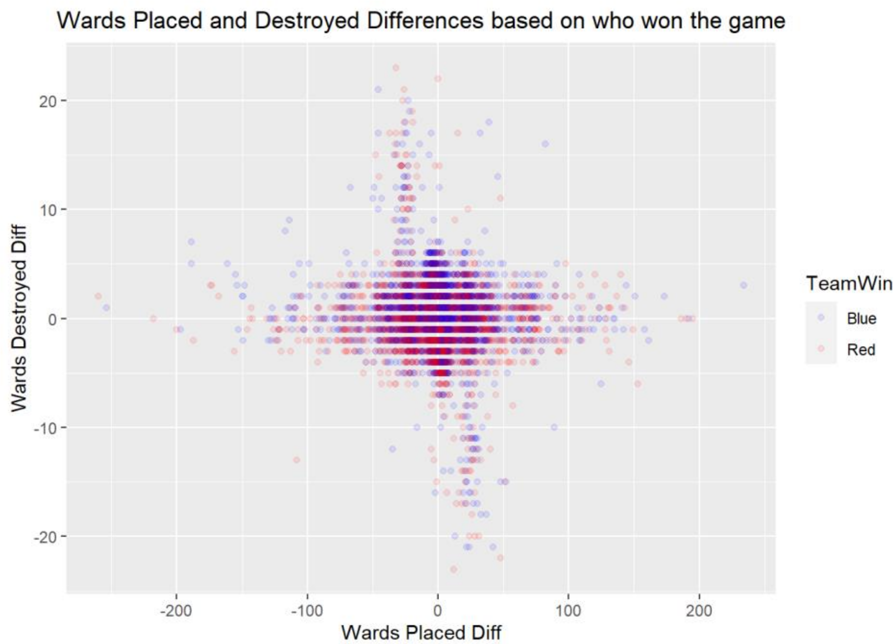

Figure 20. Wards destroyed and placed based on who the game

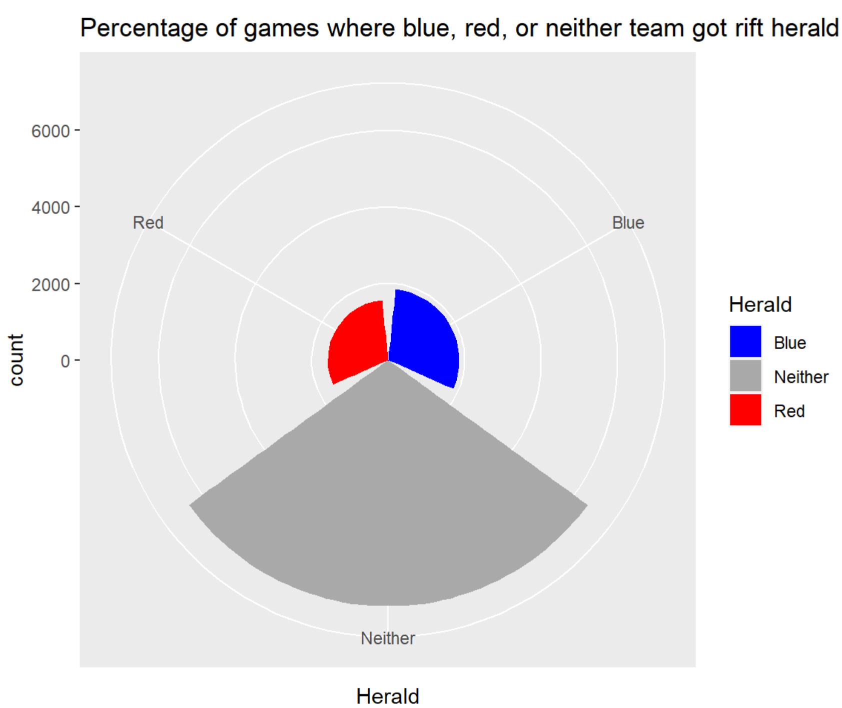

Figure 21. Pie chart of teams that took herald.

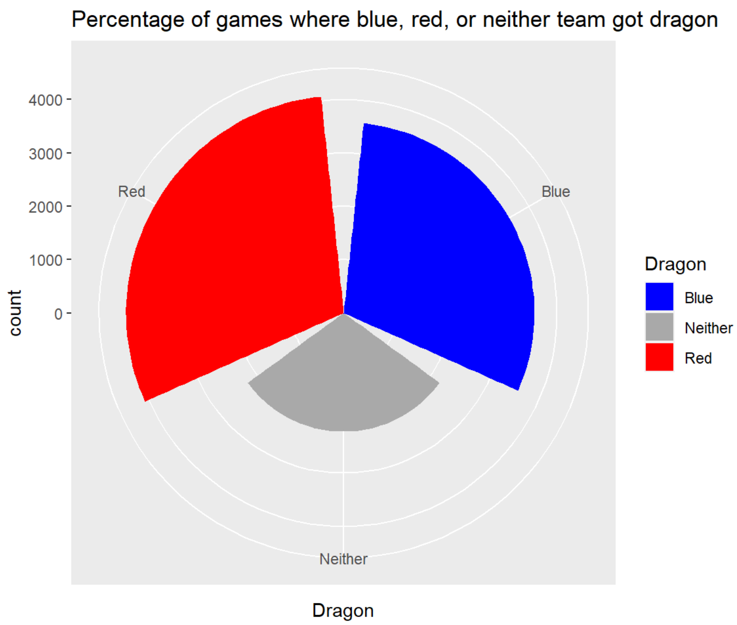

Figure 22. Pie chart of teams that took dragon.

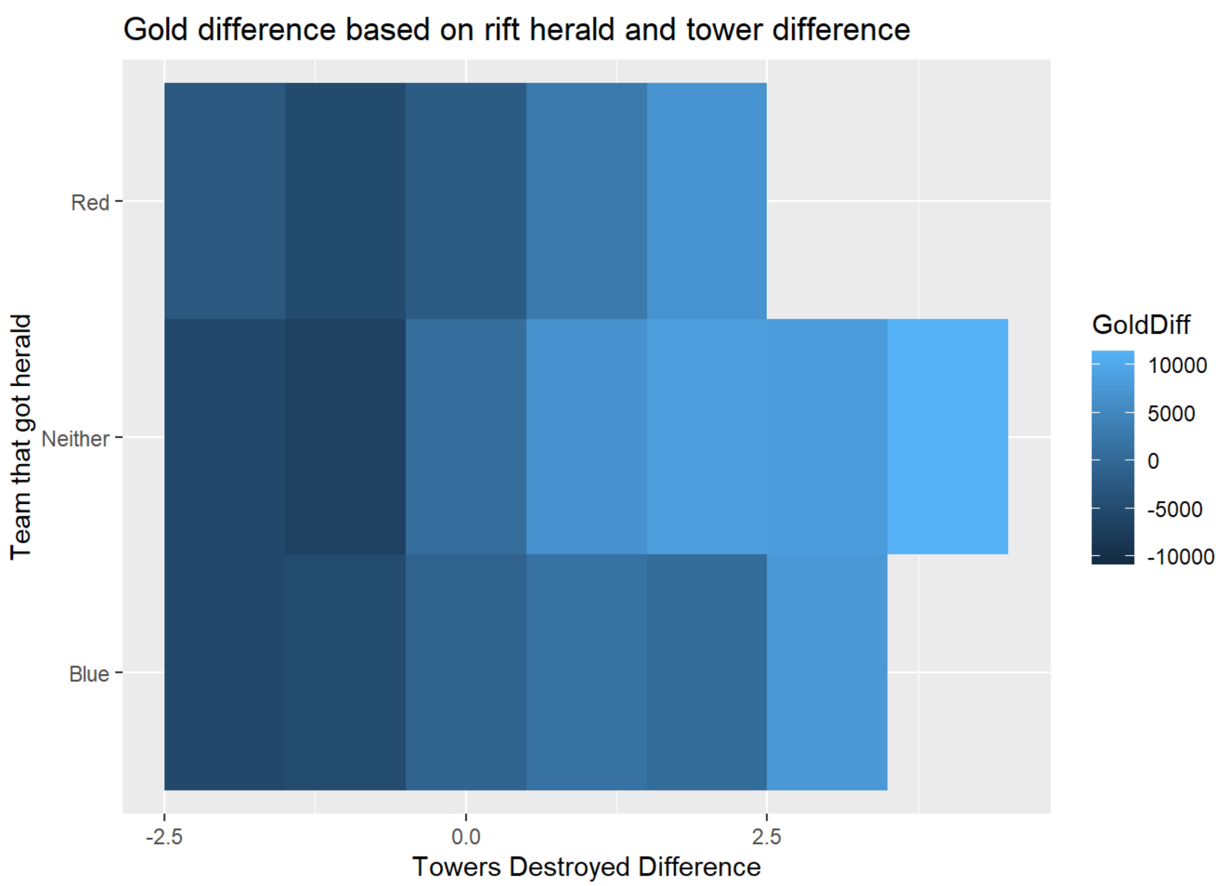

Figure 23. Gold difference based on towers destroyed difference and getting herald.

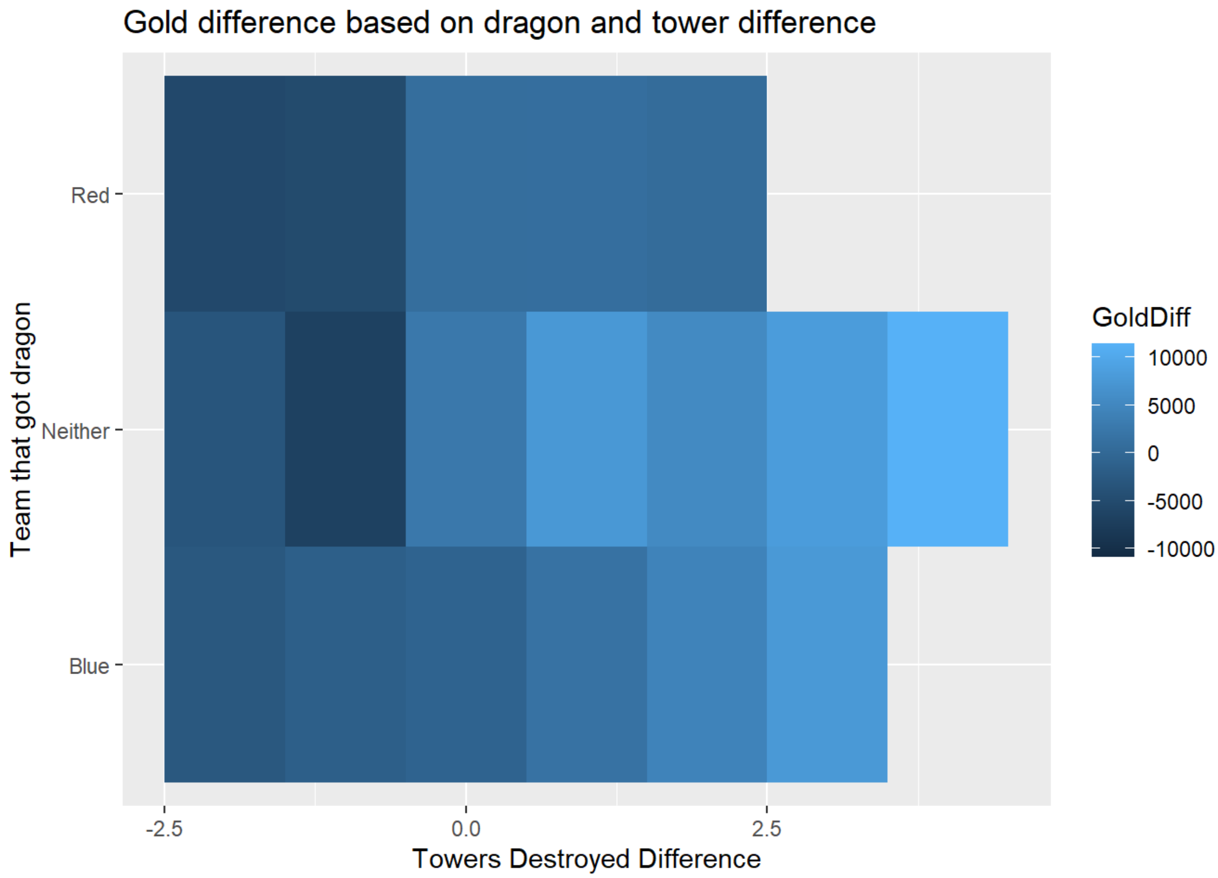

Figure 24. Gold difference based on towers destroyed difference and getting dragon.

```
## Call:
## lm(formula = GoldDiff ~ ExpDiff, data = league_csv)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4811.0  -717.9    -2.3   720.6  5415.4
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 52.843877  11.026012   4.793 1.67e-06 ***
## ExpDiff      1.143052   0.005741 199.103  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1096 on 9877 degrees of freedom
## Multiple R-squared:  0.8005, Adjusted R-squared:  0.8005
## F-statistic: 3.964e+04 on 1 and 9877 DF,  p-value: < 2.2e-16
```

Figure 25. Linear regression model for gold and experience difference.

```
##         fit        lwr       upr
## 1  2338.948   190.8393  4487.057
## 2  3482.001  1333.7422  5630.259
## 3 -5662.417 -7811.1316 -3513.703
```

Figure 26. Prediction intervals for three experience differences (2000, 3000, -5000).

```
##
## Call:
## lm(formula = GoldDiff ~ TowersDiff + WardsDestDiff + KillDiff +
##     AssistDiff + MinionsDiff + JungleMinDiff, data = multivar_golddiff)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2665.29  -359.45    -7.61   361.28  2309.04
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     21.1367     5.4816   3.856 0.000116 ***
## TowersDiff    1097.8202    17.6477  62.208  < 2e-16 ***
## WardsDestDiff    5.5746     1.9318   2.886 0.003914 **
## KillDiff       374.2895     2.5977 144.085  < 2e-16 ***
## AssistDiff      52.8515     1.7470  30.253  < 2e-16 ***
## MinionsDiff     22.1380     0.2027 109.195  < 2e-16 ***
## JungleMinDiff   23.0848     0.3875  59.567  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 542.8 on 9872 degrees of freedom
## Multiple R-squared:  0.9511, Adjusted R-squared:  0.9511
## F-statistic: 3.199e+04 on 6 and 9872 DF,  p-value: < 2.2e-16
```

Figure 27. Multivariate Regression for predicting gold difference

```
##        fit       lwr      upr
## 1 688.0573 -378.9583 1755.073
##        fit       lwr      upr
## 1 -455.934 -1521.637 609.7689
```

Figure 28. Prediction intervals for two sets of parameters into the multivariate model.

```
## ## Call:
## glm(formula = TeamWin ~ GoldDiff + ExpDiff + TowersDiff + MinionsDiff +
##     JungleMinDiff, family = "binomial", data = logistic_csv)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8050  -0.8921  -0.1481   0.8888   2.6586
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -4.693e-04  2.380e-02  -0.020  0.98427
## GoldDiff       4.704e-04  2.505e-05  18.779  < 2e-16 ***
## ExpDiff        2.168e-04  3.148e-05   6.886 5.72e-12 ***
## TowersDiff    -2.483e-01  9.415e-02  -2.637  0.00837 **
## MinionsDiff   -3.132e-03  1.031e-03  -3.038  0.00238 **
## JungleMinDiff  4.275e-03  1.904e-03   2.245  0.02478 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13695  on 9878  degrees of freedom
## Residual deviance: 10555  on 9873  degrees of freedom
## AIC: 10567
##
## Number of Fisher Scoring iterations: 4
```

Figure 29. Logistic regression model to predict who won the game.

```
##   (Intercept)      GoldDiff       ExpDiff    TowersDiff   MinionsDiff
##     0.9995308     1.0004705     1.0002168     0.7801625     0.9968729
## JungleMinDiff
##     1.0042839
```

Figure 30. Exponentiated coefficients for model in figure 29.

```
## glm(formula = Dragon ~ GoldDiff + ExpDiff + TowersDiff + WardsDestDiff +
##     WardsPlacedDiff + KillDiff + AssistDiff + MinionsDiff + JungleMinDiff,
##     family = "binomial", data = drag_logistic_csv)
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6553  -0.9914  -0.5200   1.0376   2.5672
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.221e-01  2.525e-02  -4.835 1.33e-06 ***
## GoldDiff        -4.082e-04  4.957e-05  -8.234  < 2e-16 ***
## ExpDiff         -2.715e-04  3.544e-05  -7.661 1.85e-14 ***
## TowersDiff       3.710e-01  9.864e-02   3.761 0.000169 ***
## WardsDestDiff    2.322e-02  8.876e-03   2.616 0.008888 **
## WardsPlacedDiff  3.278e-03  9.852e-04   3.327 0.000878 ***
## KillDiff         3.566e-01  2.327e-02  15.323  < 2e-16 ***
## AssistDiff       6.331e-02  8.848e-03   7.155 8.37e-13 ***
## MinionsDiff      1.501e-02  1.478e-03  10.153  < 2e-16 ***
## JungleMinDiff    5.379e-02  2.480e-03  21.690  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 10581.5  on 7656  degrees of freedom
## Residual deviance:  9182.6  on 7647  degrees of freedom
## AIC: 9202.6
## Number of Fisher Scoring iterations: 3
```

Figure 31. Logistic regression model to predict who took the dragon.

```
##     (Intercept)        GoldDiff         ExpDiff      TowersDiff   WardsDes
tDiff
##       0.8850667       0.9995919       0.9997285       1.4491487        1.02
34934
## WardsPlacedDiff        KillDiff      AssistDiff     MinionsDiff   JungleMi
nDiff
##       1.0032830       1.4284462       1.0653550       1.0151188        1.05
52635
```

Figure 32. Exponentiated coefficients for model in figure 31.