League of Legends

Thanh

2024-06-01

Introduction

League of Legends

League of Legends is a team-based strategy game where two teams of five powerful champions face off to destroy the other's base. Choose from over 160 champions to make epic plays, secure kills, and take down towers as you battle your way to victory.

Glossary

- Warding totem: An item that can be used to provide visions for map/objectives control.
- Minions: Valuable resources that can be killed to gain gold.
- Jungle monsters: Valuable resources that can be killed to gain gold and buffs.
- Elite monsters: Valuable resources that can be killed to gain massive advantages.
- Dragons: Elite monster which gives team bonuses when killed. The 4th dragon killed by a team gives a massive stats bonus. The 5th dragon (Elder Dragon) offers a huge advantage to the team.

- Herald: Elite monster which gives stats bonus when killed. It helps pushing a lane and destroying structures.
- Towers: Structures you have to destroy to reach the enemy Nexus.
- Level: Champion level. Start at 1. Max is 18.

Dataset

This dataset contains the first 10min. stats of approx. 10k ranked games (SOLO QUEUE) from a high ELO (DIAMOND I to MASTER). Players have roughly the same level.

Each game is unique. The gameId can help you to fetch more attributes from the Riot API.

There are 19 features per team (38 in total) collected after 10min in-game. This includes kills, deaths, gold, experience, level.

```
## Rows: 9,879
## Columns: 40
## $ gameId                       <dbl> 4519157822, 4523371949, 4521474530, 45243~
## $ blueWins                     <int> 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0,~
## $ blueWardsPlaced              <int> 28, 12, 15, 43, 75, 18, 18, 16, 16, 13, 2~
## $ blueWardsDestroyed           <int> 2, 1, 0, 1, 4, 0, 3, 2, 3, 1, 3, 2, 1, 3,~
## $ blueFirstBlood               <int> 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0,~
## $ blueKills                    <int> 9, 5, 7, 4, 6, 5, 7, 5, 7, 4, 4, 11, 7, 4~
## $ blueDeaths                   <int> 6, 5, 11, 5, 6, 3, 6, 13, 7, 5, 4, 11, 1,~
## $ blueAssists                  <int> 11, 5, 4, 5, 6, 6, 7, 3, 8, 5, 6, 7, 11, ~
## $ blueEliteMonsters            <int> 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1,~
## $ blueDragons                  <int> 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0,~
## $ blueHeralds                  <int> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1,~
## $ blueTowersDestroyed          <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ blueTotalGold                <int> 17210, 14712, 16113, 15157, 16400, 15899,~
## $ blueAvgLevel                 <dbl> 6.6, 6.6, 6.4, 7.0, 7.0, 7.0, 6.8, 6.4, 7~
## $ blueTotalExperience          <int> 17039, 16265, 16221, 17954, 18543, 18161,~
## $ blueTotalMinionsKilled       <int> 195, 174, 186, 201, 210, 225, 225, 209, 1~
## $ blueTotalJungleMinionsKilled <int> 36, 43, 46, 55, 57, 42, 53, 48, 61, 39, 2~
## $ blueGoldDiff                 <int> 643, -2908, -1172, -1321, -1004, 698, 241~
## $ blueExperienceDiff           <int> -8, -1173, -1033, -7, 230, 101, 1563, -80~
## $ blueCSPerMin                 <dbl> 19.5, 17.4, 18.6, 20.1, 21.0, 22.5, 22.5,~
## $ blueGoldPerMin               <dbl> 1721.0, 1471.2, 1611.3, 1515.7, 1640.0, 1~
## $ redWardsPlaced               <int> 15, 12, 15, 15, 17, 36, 57, 15, 15, 16, 1~
## $ redWardsDestroyed            <int> 6, 1, 3, 2, 2, 5, 1, 0, 2, 2, 2, 1, 1, 3,~
## $ redFirstBlood                <int> 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1,~
## $ redKills                     <int> 6, 5, 11, 5, 6, 3, 6, 13, 7, 5, 4, 11, 1,~
## $ redDeaths                    <int> 9, 5, 7, 4, 6, 5, 7, 5, 7, 4, 4, 11, 7, 4~
## $ redAssists                   <int> 8, 2, 14, 10, 7, 2, 9, 11, 5, 4, 5, 9, 1,~
## $ redEliteMonsters             <int> 0, 2, 0, 0, 1, 0, 0, 1, 2, 0, 1, 0, 0, 0,~
## $ redDragons                   <int> 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0,~
## $ redHeralds                   <int> 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,~
## $ redTowersDestroyed           <int> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ redTotalGold                 <int> 16567, 17620, 17285, 16478, 17404, 15201,~
## $ redAvgLevel                  <dbl> 6.8, 6.8, 6.8, 7.0, 7.0, 7.0, 6.4, 6.6, 7~
## $ redTotalExperience           <int> 17047, 17438, 17254, 17961, 18313, 18060,~
## $ redTotalMinionsKilled        <int> 197, 240, 203, 235, 225, 221, 164, 157, 2~
## $ redTotalJungleMinionsKilled  <int> 55, 52, 28, 47, 67, 59, 35, 54, 53, 43, 5~
## $ redGoldDiff                  <int> -643, 2908, 1172, 1321, 1004, -698, -2411~
## $ redExperienceDiff            <int> 8, 1173, 1033, 7, -230, -101, -1563, 800,~
## $ redCSPerMin                  <dbl> 19.7, 24.0, 20.3, 23.5, 22.5, 22.1, 16.4,~
## $ redGoldPerMin                <dbl> 1656.7, 1762.0, 1728.5, 1647.8, 1740.4, 1~
```

Let's check if there are any missing values.

```
##                       gameId                     blueWins
##                            0                            0
##              blueWardsPlaced           blueWardsDestroyed
##                            0                            0
##               blueFirstBlood                    blueKills
##                            0                            0
##                   blueDeaths                  blueAssists
##                            0                            0
##            blueEliteMonsters                  blueDragons
##                            0                            0
##                  blueHeralds          blueTowersDestroyed
##                            0                            0
##                blueTotalGold                 blueAvgLevel
##                            0                            0
##          blueTotalExperience       blueTotalMinionsKilled
##                            0                            0
## blueTotalJungleMinionsKilled                 blueGoldDiff
##                            0                            0
##           blueExperienceDiff                 blueCSPerMin
##                            0                            0
##               blueGoldPerMin               redWardsPlaced
##                            0                            0
##            redWardsDestroyed                redFirstBlood
##                            0                            0
##                     redKills                    redDeaths
##                            0                            0
##                   redAssists             redEliteMonsters
##                            0                            0
##                   redDragons                   redHeralds
##                            0                            0
##           redTowersDestroyed                 redTotalGold
##                            0                            0
##                  redAvgLevel           redTotalExperience
##                            0                            0
##        redTotalMinionsKilled  redTotalJungleMinionsKilled
##                            0                            0
##                  redGoldDiff            redExperienceDiff
##                            0                            0
##                  redCSPerMin                redGoldPerMin
##                            0                            0
```

Let's check whether each attribute has the correct type

```
##                       gameId                     blueWins
##                    "numeric"                    "integer"
##              blueWardsPlaced           blueWardsDestroyed
##                    "integer"                    "integer"
##                blueFirstBlood                     blueKills 
##                     "integer"                     "integer" 
##                    blueDeaths                   blueAssists 
##                     "integer"                     "integer" 
##             blueEliteMonsters                   blueDragons 
##                     "integer"                     "integer" 
##                   blueHeralds           blueTowersDestroyed 
##                     "integer"                     "integer" 
##                 blueTotalGold                  blueAvgLevel 
##                     "integer"                     "numeric" 
##           blueTotalExperience        blueTotalMinionsKilled 
##                     "integer"                     "integer" 
## blueTotalJungleMinionsKilled                  blueGoldDiff 
##                     "integer"                     "integer" 
##            blueExperienceDiff                  blueCSPerMin 
##                     "integer"                     "numeric" 
##                blueGoldPerMin                redWardsPlaced 
##                     "numeric"                     "integer" 
##             redWardsDestroyed                 redFirstBlood 
##                     "integer"                     "integer" 
##                      redKills                     redDeaths 
##                     "integer"                     "integer" 
##                    redAssists              redEliteMonsters 
##                     "integer"                     "integer" 
##                    redDragons                    redHeralds 
##                     "integer"                     "integer" 
##            redTowersDestroyed                  redTotalGold 
##                     "integer"                     "integer" 
##                   redAvgLevel            redTotalExperience 
##                     "numeric"                     "integer" 
##         redTotalMinionsKilled  redTotalJungleMinionsKilled 
##                     "integer"                     "integer" 
##                   redGoldDiff             redExperienceDiff 
##                     "integer"                     "integer" 
##                   redCSPerMin                 redGoldPerMin 
##                     "numeric"                     "numeric" 
```

For easy intepretation, I will create a new column "Winner"

The distribution between Red and Blue wins

How does gold advantage affect one team's winrate ?

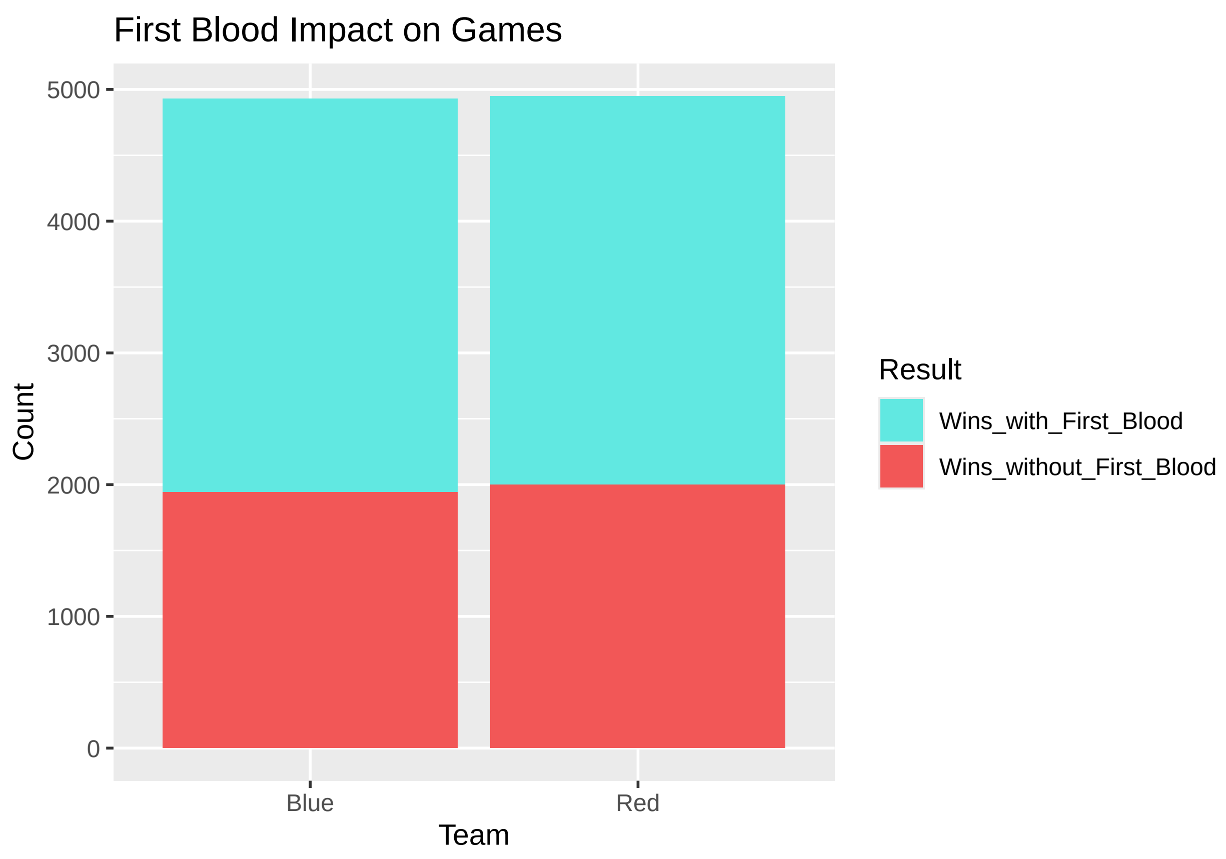
Average gold when winning a game for blue team: 1270.7180527

Win rate for blue team when exceeding the average gold: 82.15%

Average gold when winning a game for red team: 1237.0666801

Win rate for red team when exceeding the average gold: 81.53%

First Blood



Wards

Wards Placed

Let's see the wards placed by each team

We can see that both team has a similar distribution of wards placed and also outliers. The outliers are likely due to the fact that in League of Legends, there are players that have bad attitudes and refuse to cooperate with their team, they just sit in the base and place wards to troll their team.

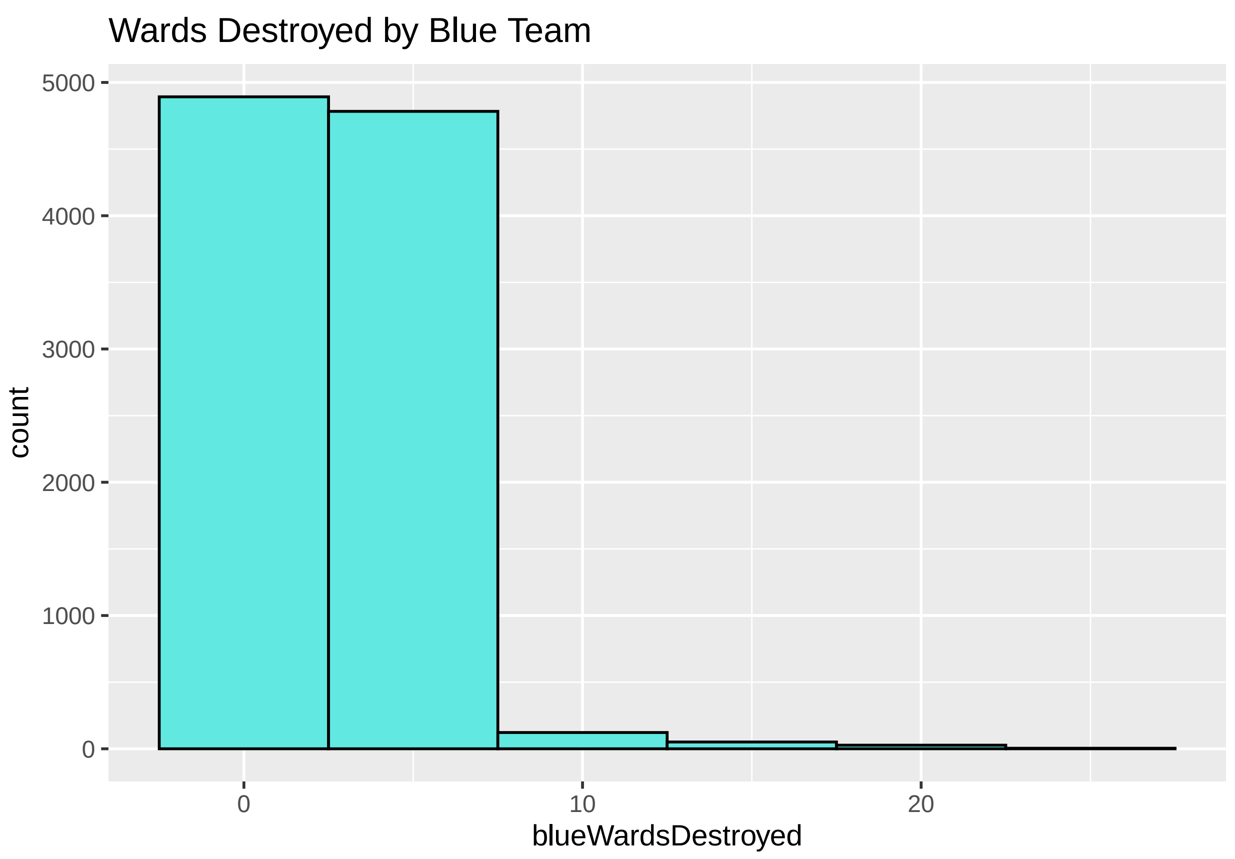
Let's see if placing more wards can help a team win the game.

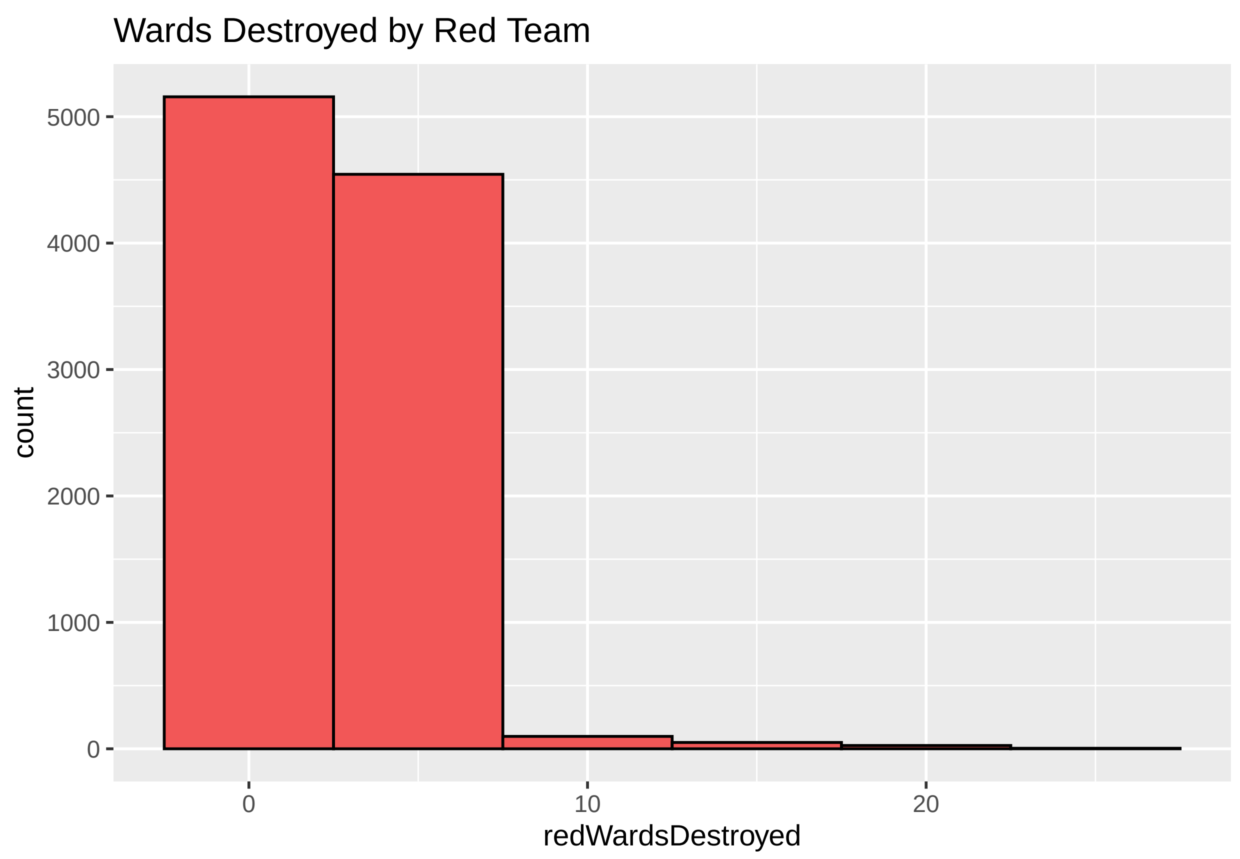
Blue's team winrate when placing more wards: 52.9858849%

Red's team winrate when placing more wards: 53.3759929%

Wards Destroyed

Let's see the wards destroyed by each team


Wards Destroyed by Blue Team
5000
4000
3000
2000
1000
0
count
0
10
20
blueWardsDestroyed



Now, let's see if destroying a proportion of the enemy's wards can help a team win the game. Let's say 25%.

Blue's team winrate when destroying 25% of the enemy's wards: 56.1697376%

Red's team winrate when destroying 25% of the enemy's wards: 57.4392413%

Minions

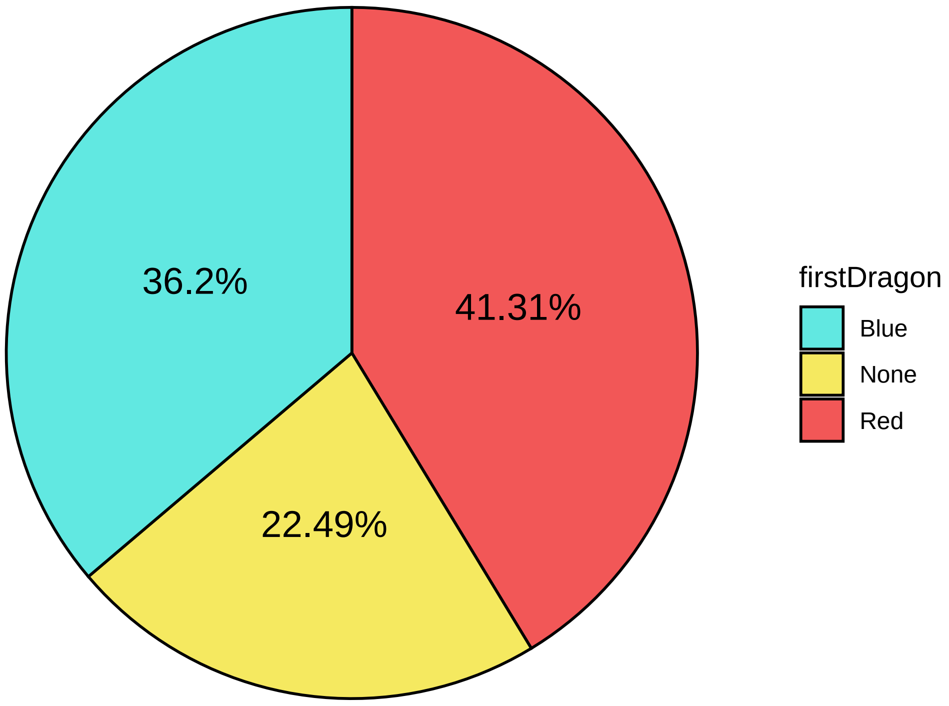
Let's see if having more minions killed can help a team win the game.

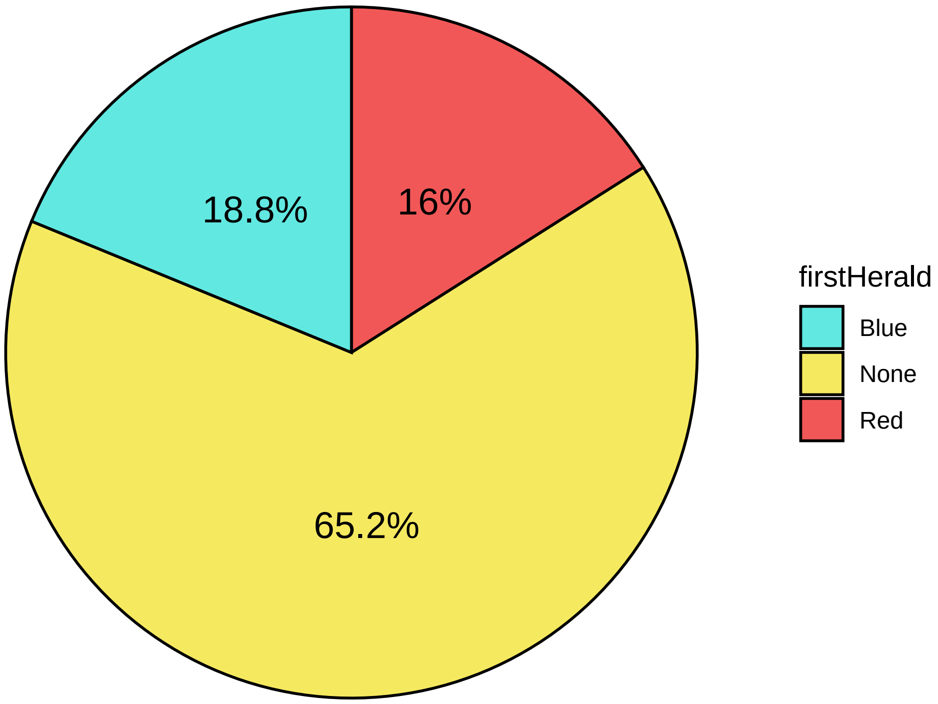
Blue's team winrate when killing more minions: 62.9683902%

Red's team winrate when killing more minions: 62.6583551%

Neutral objectives

LEAGUE OF LEGENDS
Krugs
Baron Nashor
Red Brambleback
Razorbeaks
Gromp
Blue Sentinel
Wolves
Wolves
Blue Sentinel
Gromp
Razorbeaks
Dragon
Red Brambleback
Krugs

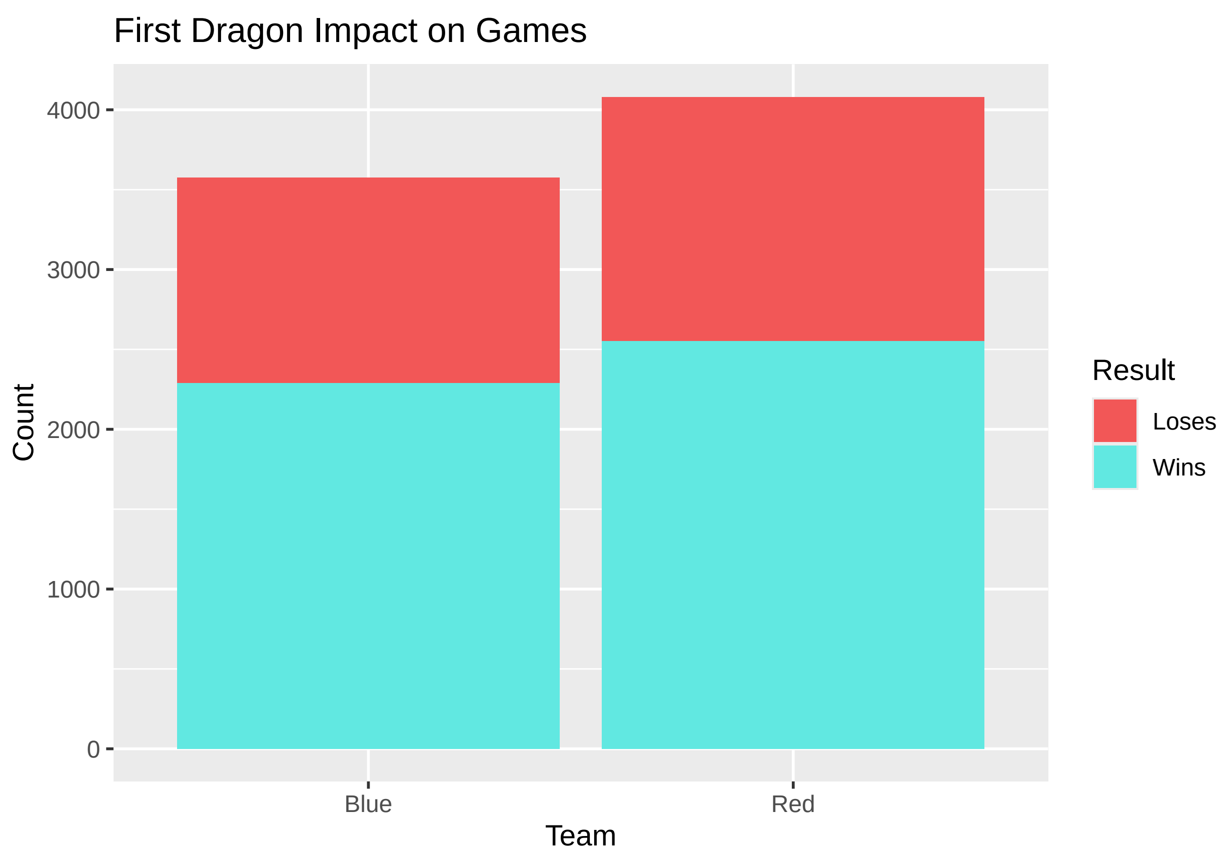


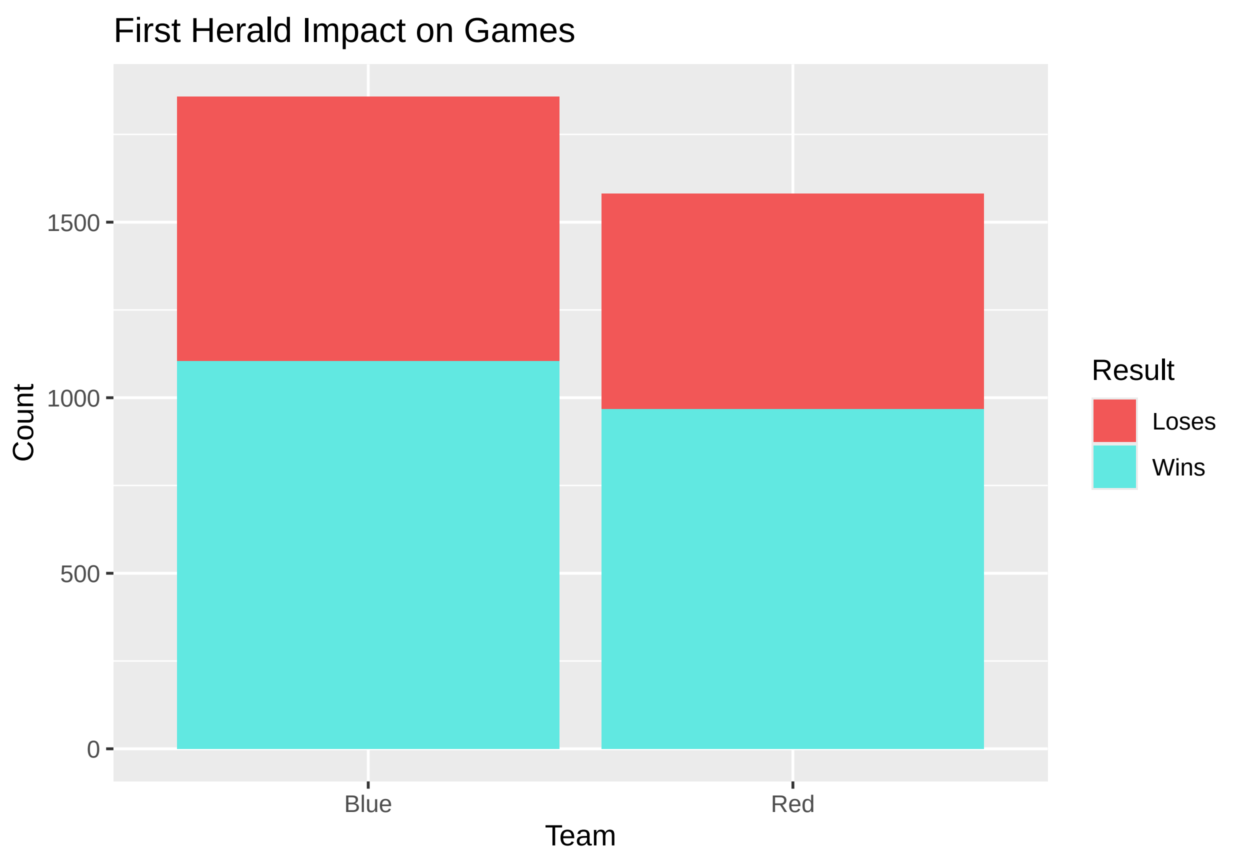


Based on the map and two pie charts above, we can see why there is a huge difference between the probability of having first dragon/herald among 2 teams.

- Firstly, the path to the dragon pit is more accessible for the red team, making them the better team to control and secure the dragon while the blue team has a better chance to secure the herald.
- Secondly, since this data is just for the first 10 minutes, the Herald does not make much of an impact on the game. The dragon, on the other hand, can provide a huge advantage to the team that secures it first.
- Out of 9879 games, there are 6441 games where the first herald is not secured while there are 2222 games where the first dragon is not secured. We can see that dragons are much more preferred than heralds.

Let's see how many games the team that secures the first dragon/herald wins.


First Dragon Impact on Games
Count
4000
3000
2000
1000
0
Blue
Red
Team
Result
Loses
Wins



What about the winrate when securing both the first dragon and herald ?

For blue team, the winrate when securing both the first dragon and herald: 73.5211268%

For red team, the winrate when securing both the first dragon and herald: 71.369863%

KDA (Kill, Death, Assist)

KDA is a metric that measures the performance of a player in a game. It is calculated by the formula (Kills + Assists) / Deaths. Should the Deaths be 0, the denominator is set to 1. The higher the KDA, the better the player is performing.

Let's create new columns "KDA" for both teams.

Blue's team winrate when having a higher KDA: 70.0919305%

Red's team winrate when having a higher KDA: 70.5226193%

We see that KDA plays a significant role in determining the outcome of the game.

Jungle

As a seasoned player, I would say that the jungle role is the most important role in the game. The jungler is responsible for securing objectives, providing vision, and helping the team in laning phase, small fights. Let's see how the jungle role affects the game.

To determine the impact of the jungle role, we will look at the total jungle monsters killed by each team and the number of elite monsters killed by each team. I would say that a better jungler would have a higher number of jungle monsters killed and elite monsters killed.

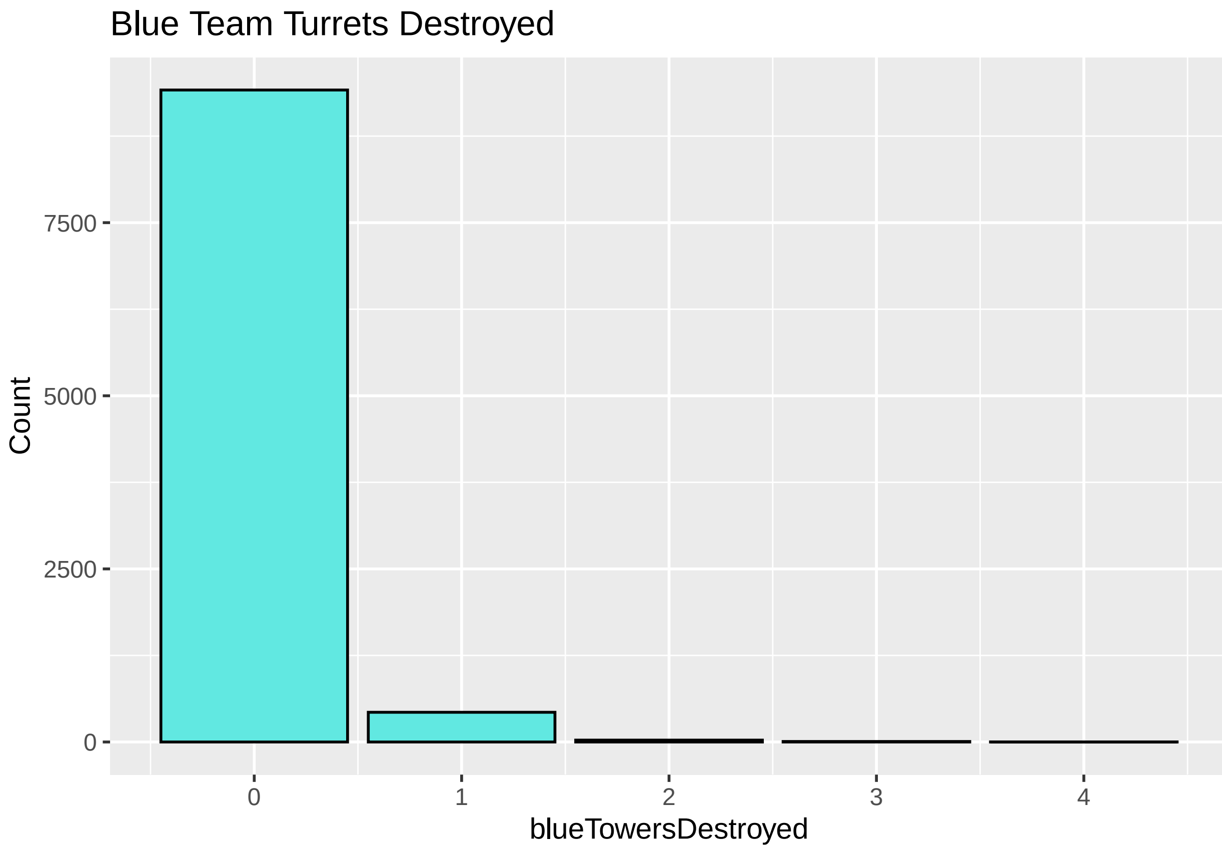
Blue's team winrate when having a better jungler: 68.7759336%

Red's team winrate when having a better jungler: 67.1341748%

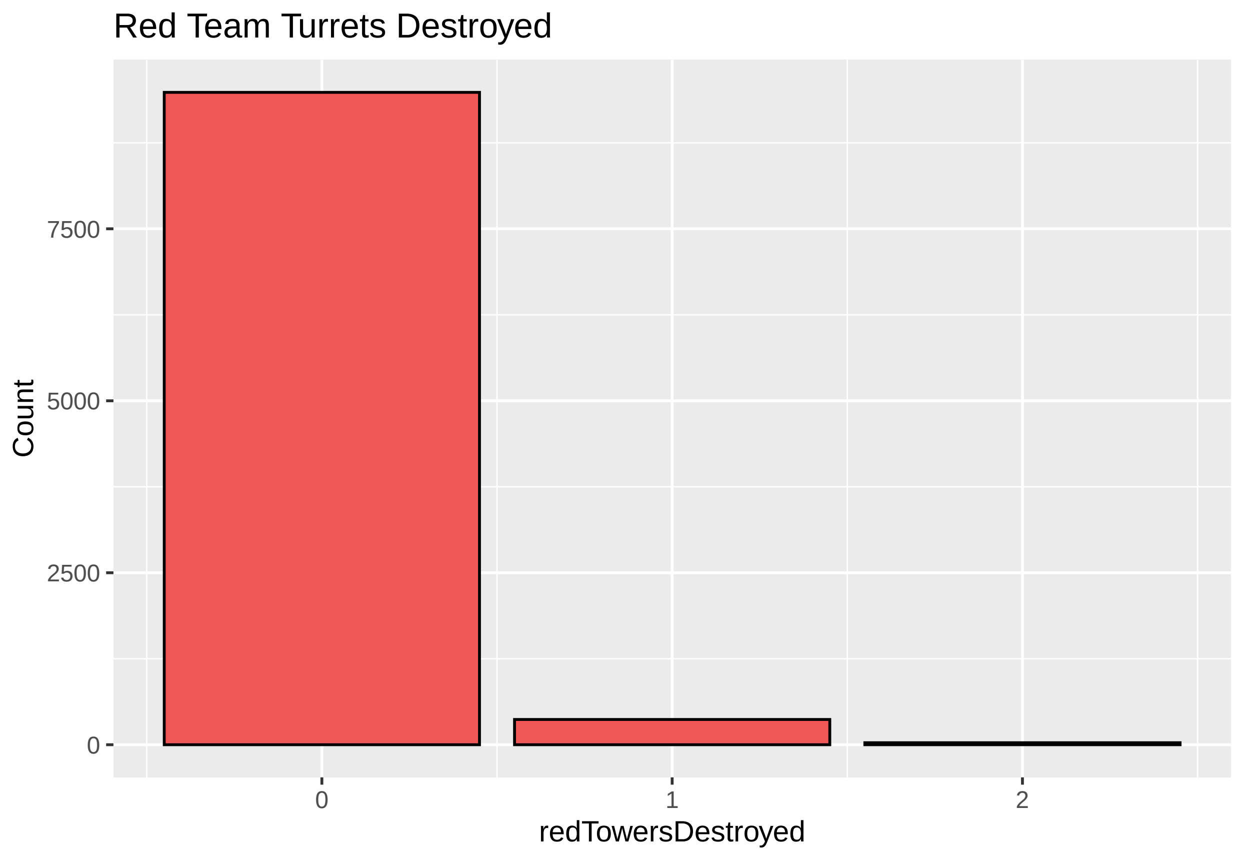
So if possible, you should help your jungler in any way you can. It can be as simple as providing vision, helping him secure objectives, defending his camps from the enemy jungler, it all helps.

Turrets

Turret takedowns are crucial in League of Legends because it allows players to roam freely, control vision,



setup team fights.



We see that there are some games where the blue team destroyed 3, 4 turrets while red team couldn't. This could be due to the fact that the blue team had access to the Herald.

In 8 games where the blue team destroyed 3 or more turrets, there are 5 games where the blue team secured the Herald.

Let's consider the winrate for both team regarding destroying 0, 1 or 2 turrets.

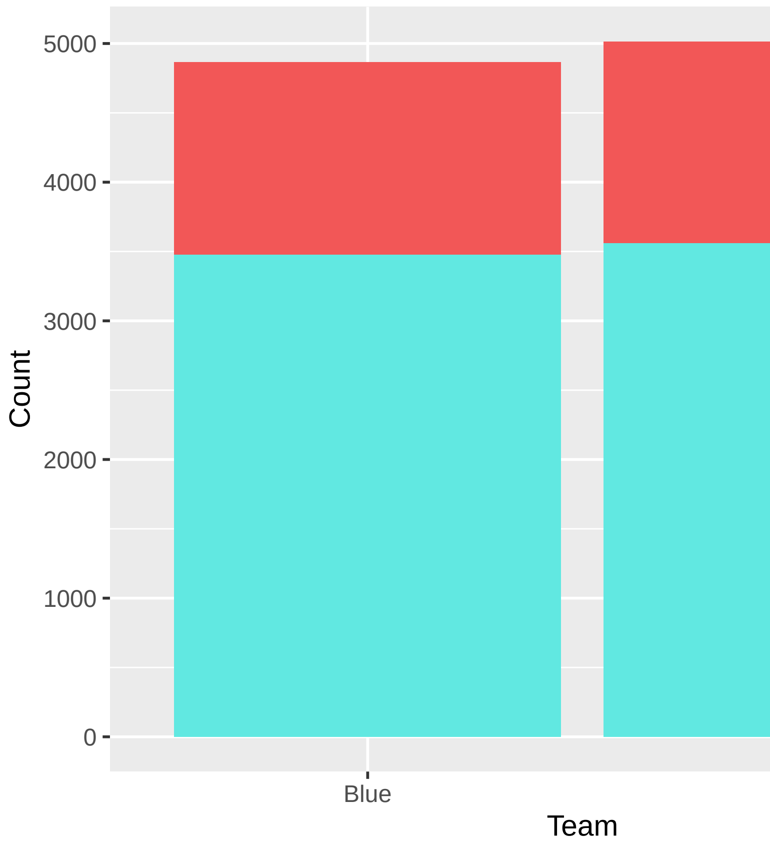
Team	turrets_0	turrets_1	turrets_2
Blue	48.64578	73.65967	96.29630
Red	49.00348	76.29428	75.86207

Seems weird that with taking down 2 turrets, the winrate for blue team is much higher than red team. Unfortunately, since the dataset is limited and the gameID cannot be used to fetch more statistics from the Riot API anymore, we cannot determine the exact reason for this.

Experience

Experience or XP is game mechanic that allows champions to level up after reaching certain amounts of XP. Leveling up unlocks new abilities or higher ranks of existing abilities. Many base stats and some items, runes, buffs, as well as certain abilities, scale with champion level. Experience normally isn't gained passively over time, and must instead be earned through various different means, such as killing minions, monsters, champions, or by being near a minion or monster when it dies.

In League of Legends, there is a term called "denying minions". Basically, it means that you are trying to prevent the enemy from getting near the minions to gain experience. Since this data is about Diamond elo, a little lead can make a huge impact. For example, reaching level 2 or 6 can mean a kill or at least a summoner spell for these players.



Let's see how having more experience can affect the game.

Logistic Regression

Variables selection

Based on the analysis above, these are the variables that will be dropped:

- gameId
- Winner
- blue/redAvgLevel
- blue/redExperienceDiff
- blue/redCSPerMin
- blue/redGoldPerMin
- firstDragon
- firstHerald

Splitting the data

We are using 80% of the data for training and 20% for testing.

```
set.seed(420)
split <- initial_split(global, prop = 0.8, strata = blueWins)
```

```
train <- split %>%
         training()
test <- split %>%
        testing()
```

Training the model

```
model <- logistic_reg(mixture=0, penalty=double(1)) %>%
  set_engine("glmnet") %>%
  set_mode("classification") %>%
  fit(blueWins ~ ., data=train)
```

Making predictions

```
predictions <- predict(model, test, type="class")
```

Evaluating the model

Confusion matrix:

```
##             Truth
## Prediction   0   1
##            0 730 273
##            1 260 713
```

The accuracy of the model is 73.03%