# Project 1

Kali, Blake, Daniel, and Tony

10/23/20

# Introduction

```
For the basis of our first project, we decided to use a data set containing key data points coll
ected from various matches in the game League of Legends.League of Legends is a multiplayer onli
ne Battle Arena (MOBA)in which two teams of five players each compete to destroy the opposing te
am’s base, called the Nexus.

The nexus of each team spawns small, NPC characters called minions (sometimes creeps) that follo
w one of three lanes and support the player(s) that has chosen that lane to capture objectives a
nd move closer towards the enemy nexus. Each lane is guarded by three towers, each at different
locations. These towers also support the player by defeating minions and enemy champions who hap
pen to be in the tower’s range.Once the three towers have been destroyed, the team can move on t
o destroy the enemy’s inhibitor. This inhibitor prevents the opposing team from summoning an upg
raded version of these minions, who deal increased damage and have significantly more health.Eac
h team has three inhibitors, each one corresponding to a lane. Only one needs to be destroyed be
fore the next stage can be completed. The final stage consists of two towers protecting the nexu
s. Both towers and at least one inhibitor must be destroyed in order to begin damaging the Nexu
s. The team who destroys their opponent’s Nexus first wins the game.


The teams are divided into sub roles that players can choose, which dictates the lane that their
champion plays in.Each player must choose a champion character to control for the course of the
game, and each champion has unique abilities and play styles to fill the roles that each team mu
st have. Some champions excel full frontal encounters, while others are more suited towards stea
lth and quick strikes, while few can heal and protect their team.Players kill both enemy minion
s, elite neutral monsters,and enemy players to increase the power of their champion and buy item
s from the gold that these kills produce.Items increase the stats of the player and allow for th
e player to “build”into a sort of play style that is suited for the champion they chose.
```

# Import + Cleaning

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.3
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
df <- read.csv("https://raw.githubusercontent.com/kalihale/miscfiles/main/high_diamond_ranked_10
min.csv")
names(df)[names(df) == "blueWins"] <- "Wins"
df$Wins <- as.factor(df$Wins)
levels(df$Wins) <- c("red", "blue")
df$blueFirstBlood <- as.factor(df$blueFirstBlood)
levels(df$blueFirstBlood) <- c("FALSE", "TRUE")
df$redFirstBlood <- as.factor(df$redFirstBlood)
levels(df$redFirstBlood) <- c("FALSE", "TRUE")

dfBlueWins <- df[df$Wins == "blue", ]
dfRedWins <- df[df$Wins == "red", ]
```

# Data Dictionary

```
This data set is a collection of many of the metrics listed above, and will be used in our analy
sis to see which of these metrics seems to contribute the most to a victory for a given team.For
the sake of our analysis, we have set up a dictionary to make our data more understandable. The
following is a list of terms and how they are to be understood within the Data Set.
```

- “Red” – The Color of the Opposing Team
- “Blue” – The Color of the Current Team
- gameId – The Riot ID assigned to the match
- Champion – Refers to the playable character that the player has chosen to compete with.
- Ward – Device used to grant players vision of areas that they are not currently in range to see.
- First Blood - Refers to the first player kill of the game.
- Kill – Refers to the act of a player defeating another player through combat.
- Death – Refers to the act of being defeated in both Player vs. Player and environmental combat.
- Assists – Refers to score points given to teammates who help another player in killing an enemy.
- Lane – Refers to one of the three standard areas in which players compete (Top, Middle, Bottom). They dictate where champions are best suited as well as where minions can be killed.
- Jungle - Refers to the area outside of the lanes where additional monsters exist that can be slain, including large and elite monsters.
- Minions / Creeps – Small non-player characters who roam the lanes of the game, attempting to help players reach the goal of destroying the enemy Nexus.
- Monster – Small non-player characters who reside in the Jungle and are used by champions to gather resources and experience.

- Elite / Elite Monsters – Large,non-player characters who are substantially tougher than the regular monsters and tend to provide some bonus to the player after being slain.
- Dragon – Large monster that resides in the bottom half of the jungle,that comes in four variants. Each variant corresponds to an element, being Earth, Fire, Water, and Air. Dragons provide a bonus depending on the element when killed. When a team kill four dragons, they receive a large bonus.
- Herald – Large monster that resides in the top half of the jungle, when killed drops and eye which can be used by the player to summon a friendly version of the herald that will attack enemy towers and deal massive amounts of damage.
- Baron / Baron Nashor – Large Elite Monster that resides in the top half of the jungle. Typically require a full team effort to kill. When defeated, Baron drops a buff to the team which killed him that increases the effectiveness that minions have.
- Gold - Currency collected in the game used to buy equipment for champions.
- Experience - Points used to gauge progress towards champion levels, awarded by killing players, monsters, and minions.
- Level – Refers to the current tier that the chosen champion currently resides as. Each level comes with increased value of that champion.
- CS / Creep Score / MS / Minion Score – Score that each player has which is a sum tally of how many non-playable enemies that the player has killed.
- Diff – Refers to the difference in between two measurements.

# Exploratory Data Analysis

## Dragon, herald

```
In this section, we are going to explore if the elite monsters on the map will affect the winrat
e. Since the dragon spawn at 5:00 and respawn every 5 mintes after killed,the Herald spawn at 8:
00 and respawn once 6 minutes after killed, there could only be 1 being capture for the first 10
minutes of a game.
```

```
#Make df$Wins numerical to calculate the winrate
levels(df$Wins) <- c("0", "1")
df$Wins <- as.numeric(df$Wins)
##Select Data
Blue_dragon <- df[df$blueDragons==1,]
Red_dragon <- df[df$redDragons==1,]

Blue_Herald <- df[df$blueHeralds==1,]
Red_Herald <- df[df$redHeralds==1,]

Blue_both <- df[df$blueEliteMonsters==2,]
Red_both <- df[df$redEliteMonsters==2,]

nrow(Blue_dragon)/nrow(df)
```

```
## [1] 0.36198
```

```
nrow(Red_dragon)/nrow(df)
```

```
## [1] 0.4130985
```

```
##Red side have slightly more chance to get a dragon
nrow(Blue_Herald)/nrow(df)
```

```
## [1] 0.1879745
```

```
nrow(Red_Herald)/nrow(df)
```

```
## [1] 0.1600364
```

```
##Blue side have slightly more chance to get a herald
nrow(Blue_both)/nrow(df)
```

```
## [1] 0.07186962
```

```
nrow(Red_both)/nrow(df)
```

```
## [1] 0.07389412
```

```
##About the same
```

Does dragon make a difference in winrate?

```
H0: No. The mean difference of winrate is less than or equals to 0.
Ha: Yes. The mean difference of winrate is greater than 0.
```

```
t.test(df$Wins,Blue_dragon$Wins)$p.value
```

```
## [1] 6.181448e-50
```

```
Very small number. We can reject the null hypothesis. We can conclude that getting dragon can de
finitly increase the winrate.
```

Does Herald make a difference in winrate?

```
H0: No. The mean difference of winrate is less than or equals to 0.
Ha: Yes. The mean difference of winrate is greater than 0.
```

```
t.test(df$Wins,Blue_Herald$Wins)$p.value
```

```
## [1] 1.797869e-14
```

```
Very small number. We can reject the null hypothesis. We can conclude that getting herald can de
finitly increase the winrate.
```

How about both?

```
H0: No. The mean difference of winrate is less than or equals to 0.
Ha: Yes. The mean difference of winrate is greater than 0.
```

```
t.test(df$Wins,Blue_both$Wins)$p.value
```

```
## [1] 2.009182e-38
```

```
Very small number. We can reject the null hypothesis. We can conclude that getting both dragon a
nd herald can definitly increase the winrate.
```

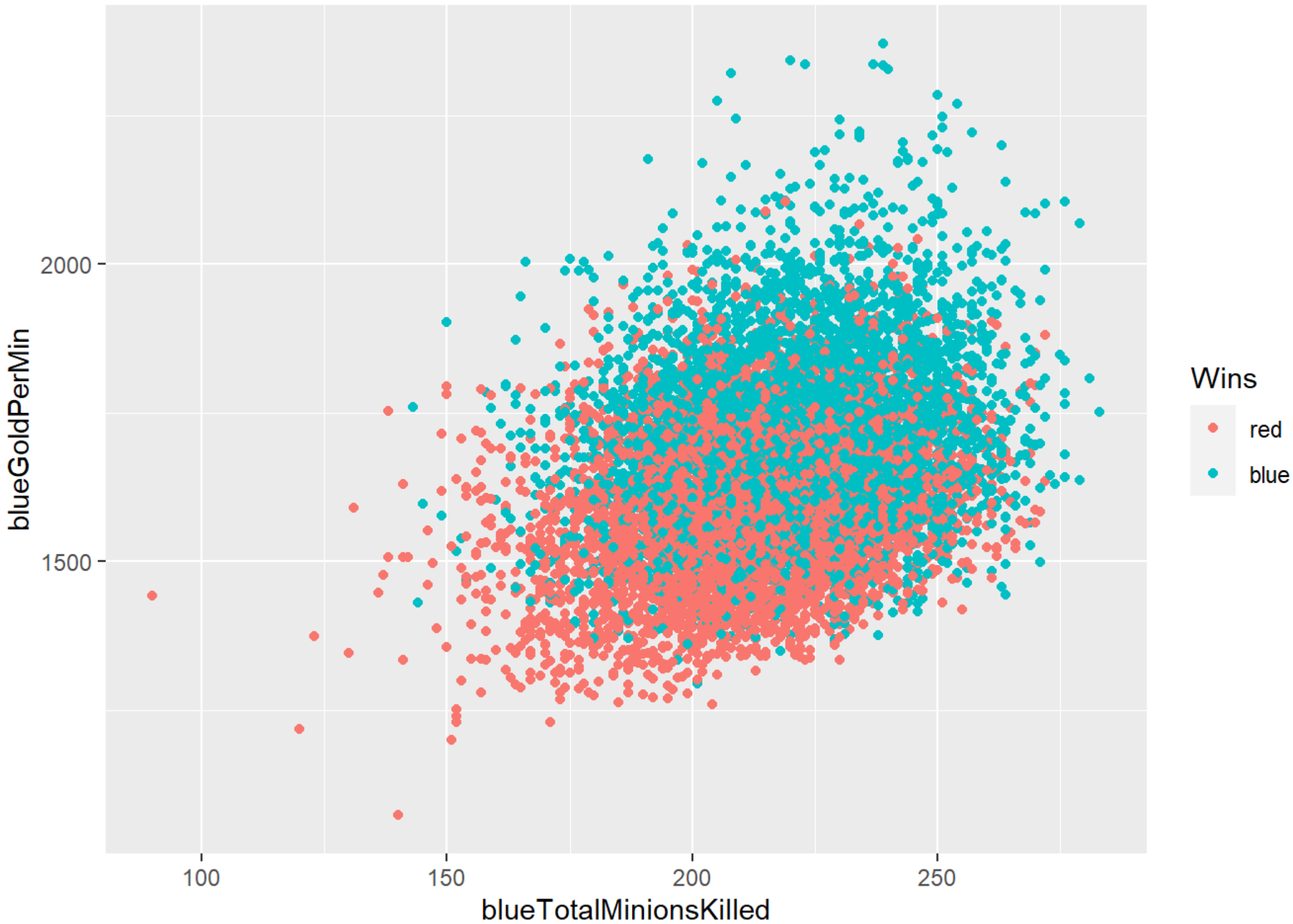
Change df$Wins back.

```
df$Wins <- as.factor(df$Wins)
levels(df$Wins) <- c("red", "blue")
```

# Gold diff

```
The creep score of a team vs its gold gain rate per minute are positively related. The more mini
ons a team kills, the quicker and more gold their tend to gain.
```

```
ggplot(data = df, aes(x = blueTotalMinionsKilled, y = blueGoldPerMin, color = Wins)) + geom_poin
t()
```



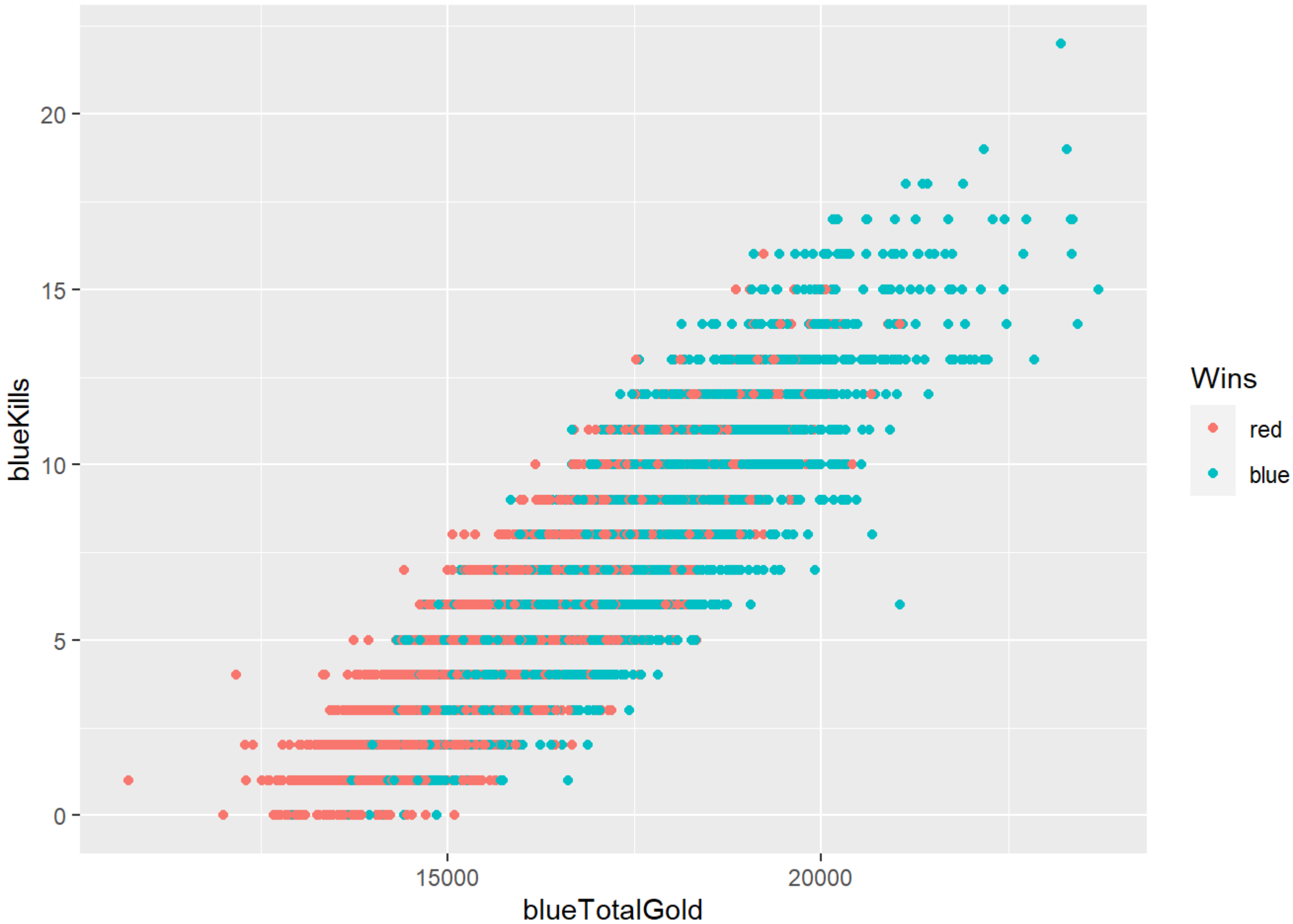

```
ggplot(data = df, aes(x = blueGoldPerMin, y = redGoldPerMin, color = Wins)) + geom_point()
```


```
cor(df$blueGoldPerMin, df$redGoldPerMin)
```

```
## [1] -0.3142125
```

```
Kills also play a role in this calculation as kills do provide gold. Again, another positve rela
tionship is exhibited as the more kills a team has, the better financially they tend to do.
```

```
ggplot(data = df, aes(x = blueTotalGold, y = blueKills, color = Wins)) + geom_point()
```



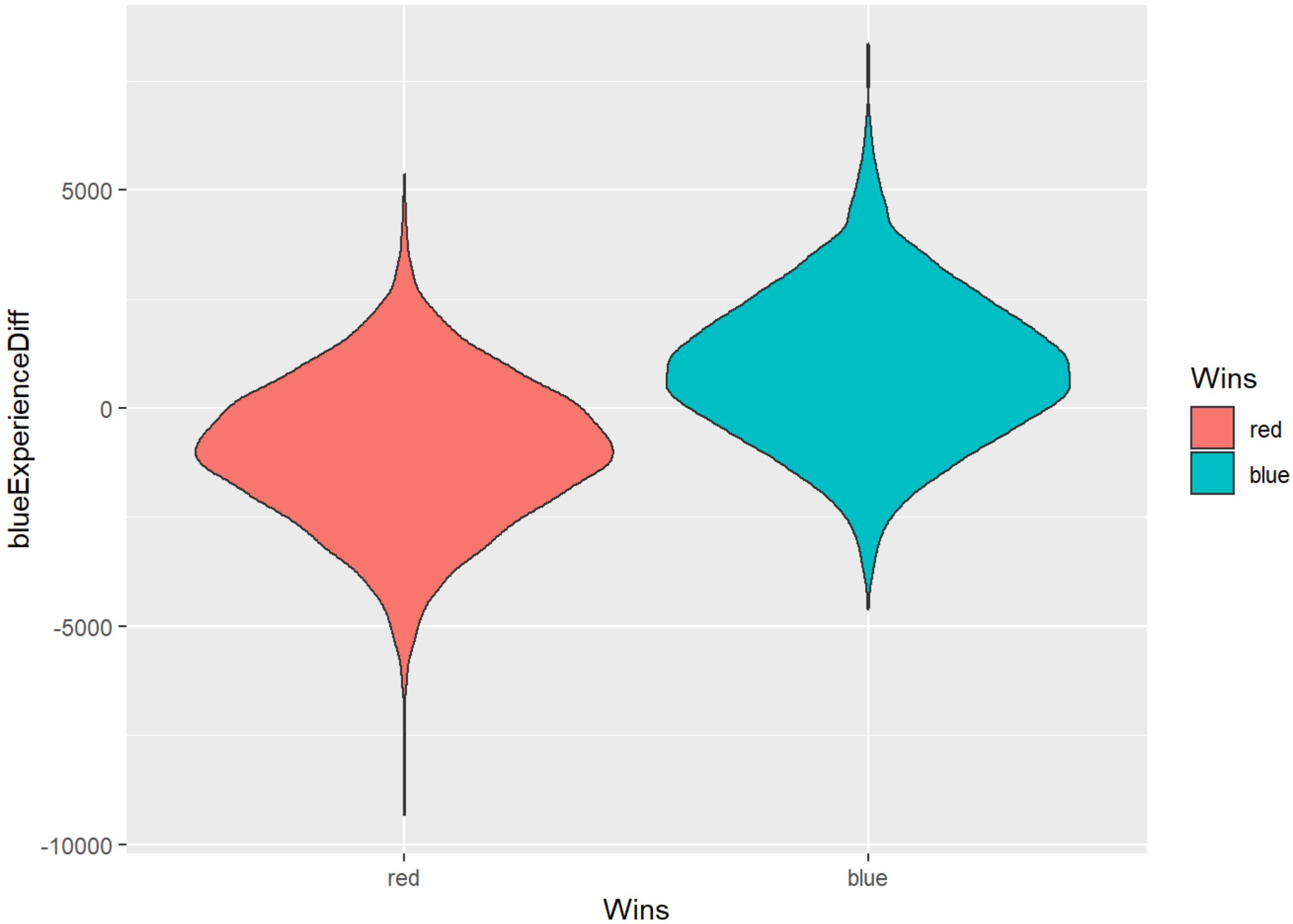

```
cor(df$blueTotalGold, df$blueKills)
```

```
## [1] 0.8887509
```

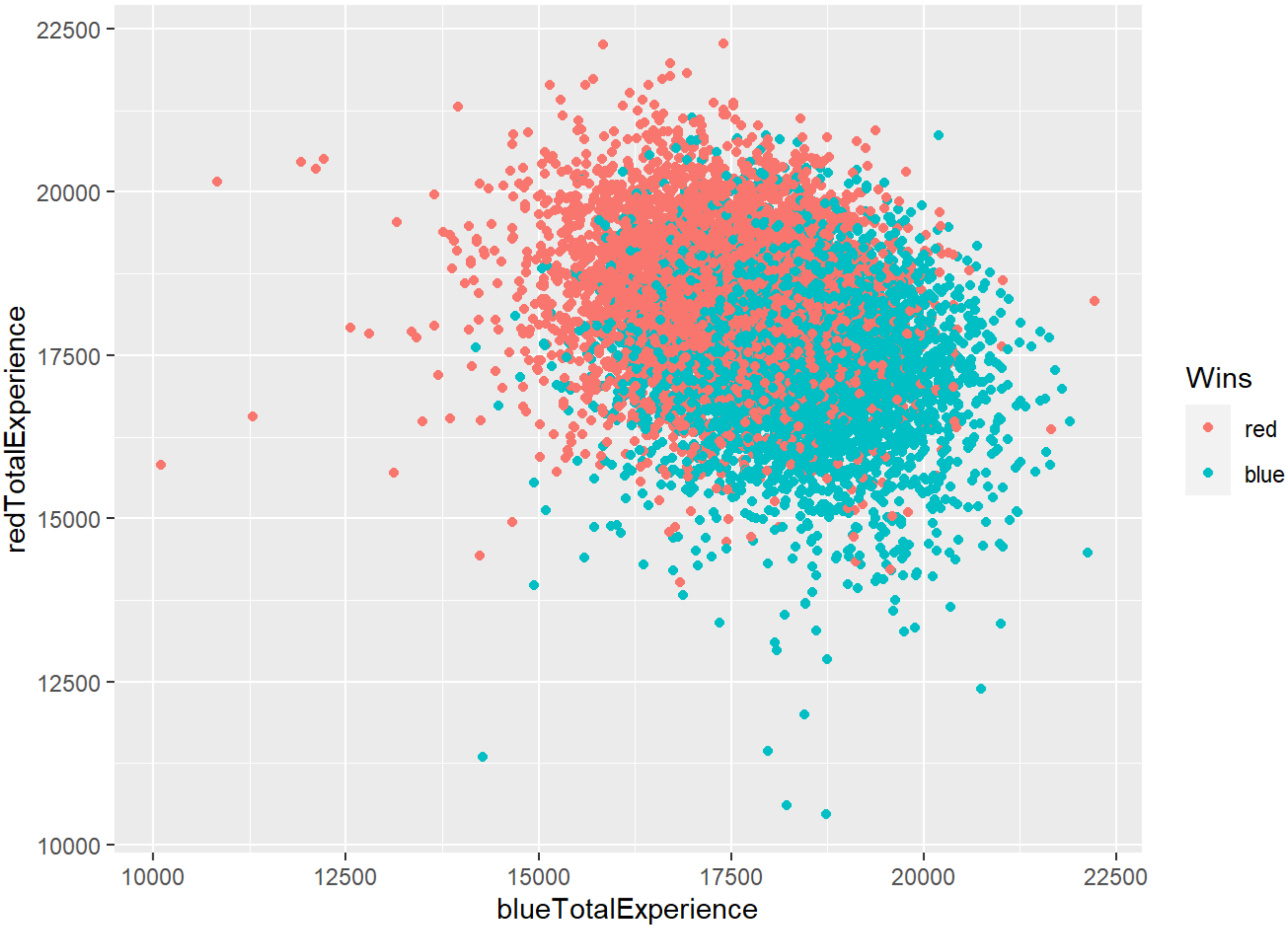
# Experience diff

```
The experience difference a team has is positively related to the their performance in game. If
a team has a positive experience diff in comparison to the other team (current - opposing), they
exhibit greater frequency of wins.
```

```
ggplot(data = df, aes(x = Wins, y = blueExperienceDiff, fill = Wins)) + geom_violin()
```




```
ggplot(data = df, aes(x = blueTotalExperience, y = redTotalExperience, color = Wins)) + geom_poi
nt()
```




```
t.test(dfBlueWins$blueTotalExperience, dfBlueWins$redTotalExperience)
```

```
##
##  Welch Two Sample t-test
##
## data:  dfBlueWins$blueTotalExperience and dfBlueWins$redTotalExperience
## t = 40.784, df = 9829.1, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  864.6201 951.9288
## sample estimates:
## mean of x mean of y
##  18404.58  17496.30
```

```
t.test(dfRedWins$blueTotalExperience, dfRedWins$redTotalExperience)
```

```
##
##  Welch Two Sample t-test
##
## data:  dfRedWins$blueTotalExperience and dfRedWins$redTotalExperience
## t = -43.886, df = 9871.6, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1015.3092  -928.4887
## sample estimates:
## mean of x mean of y
##  17453.47  18425.37
```

```
The Converse also proved to be false.
```

# Wards Placed

```
library(tidyverse)

meanPerGame <- function(default = 1){

  return( (df$blueWardsPlaced[default] + df$redWardsPlaced[default]) )

}
means <- sapply(1:nrow(df), meanPerGame)

mean(means)
```

```
## [1] 44.65624
```

```
max(dfBlueWins$blueWardsPlaced)
```

```
## [1] 250
```

```
max(dfBlueWins$redWardsPlaced)
```

```
## [1] 268
```

```
max(dfRedWins$redWardsPlaced)
```

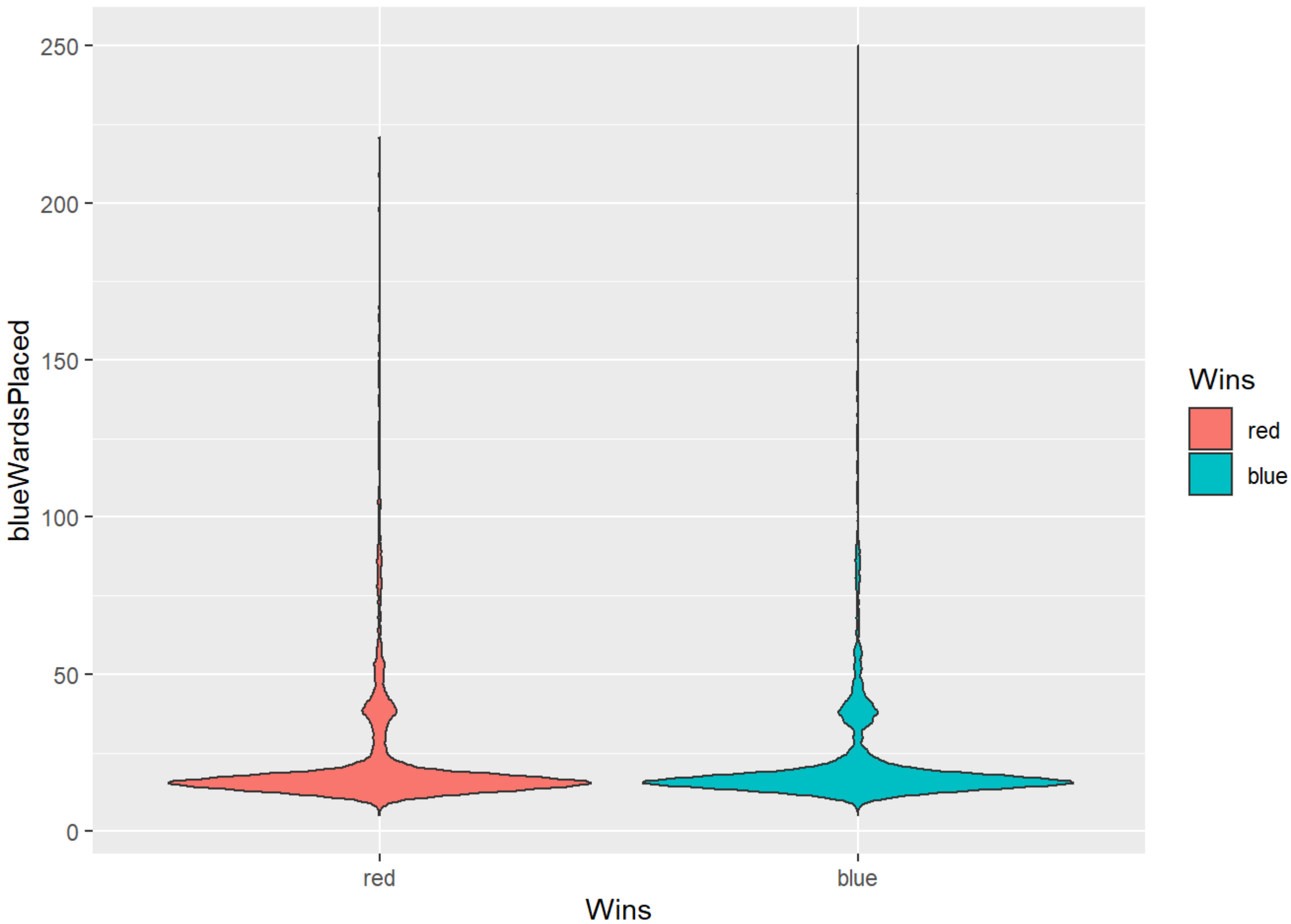
```
## [1] 276
```

```
max(dfRedWins$blueWardsPlaced)
```

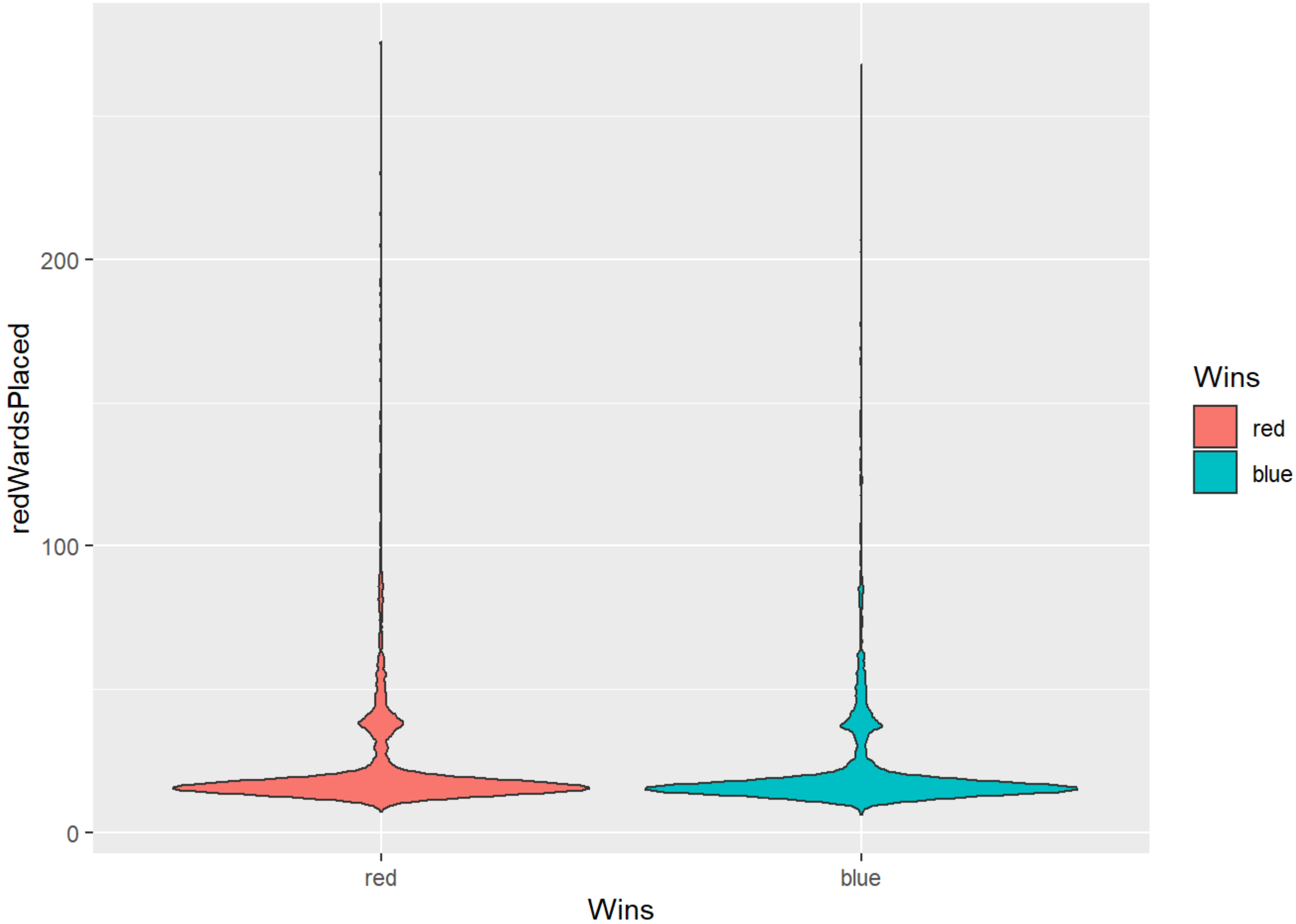
```
## [1] 221
```

```
Plots of the Distribution of Wards based upon wins-- despite winning or losing, neither team sho
ws any inherent advantage based upon their wards.
```

```
ggplot(data = df, aes(x = Wins, y = blueWardsPlaced, fill = Wins)) + geom_violin()
```




```
ggplot(data = df, aes(x = Wins, y = redWardsPlaced, fill = Wins)) + geom_violin()
```




```
length(which(df$Wins == "blue"))
```

```
## [1] 4930
```

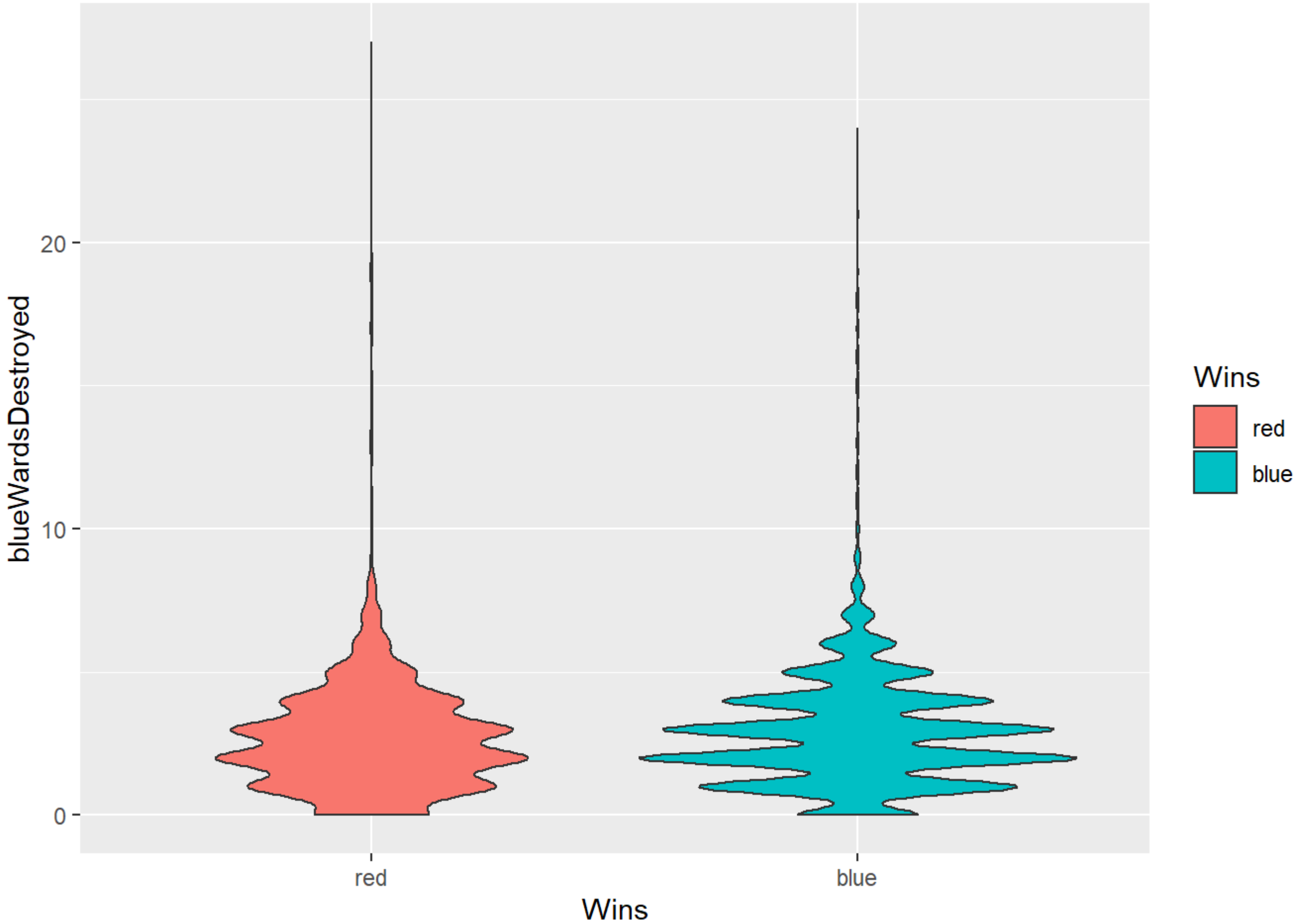
```
length(which(df$Wins == "red"))
```

```
## [1] 4949
```

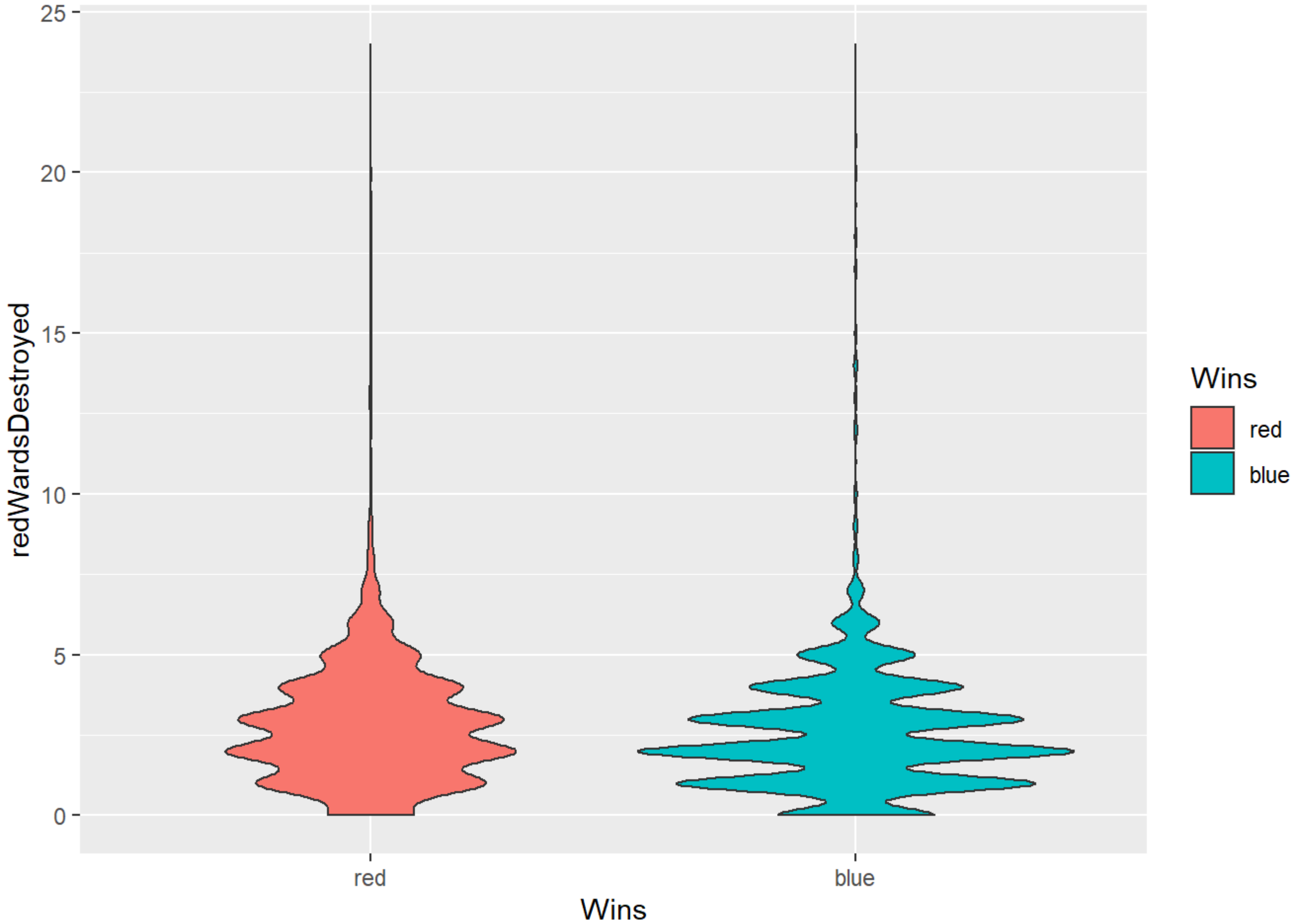
```
ggplot(data = df, aes(blueWardsPlaced, redWardsPlaced, color = Wins)) + geom_count() + geom_smoo
th()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```


```
ggplot(data = df, aes(x = Wins, y = blueWardsDestroyed, fill = Wins)) + geom_violin()
```

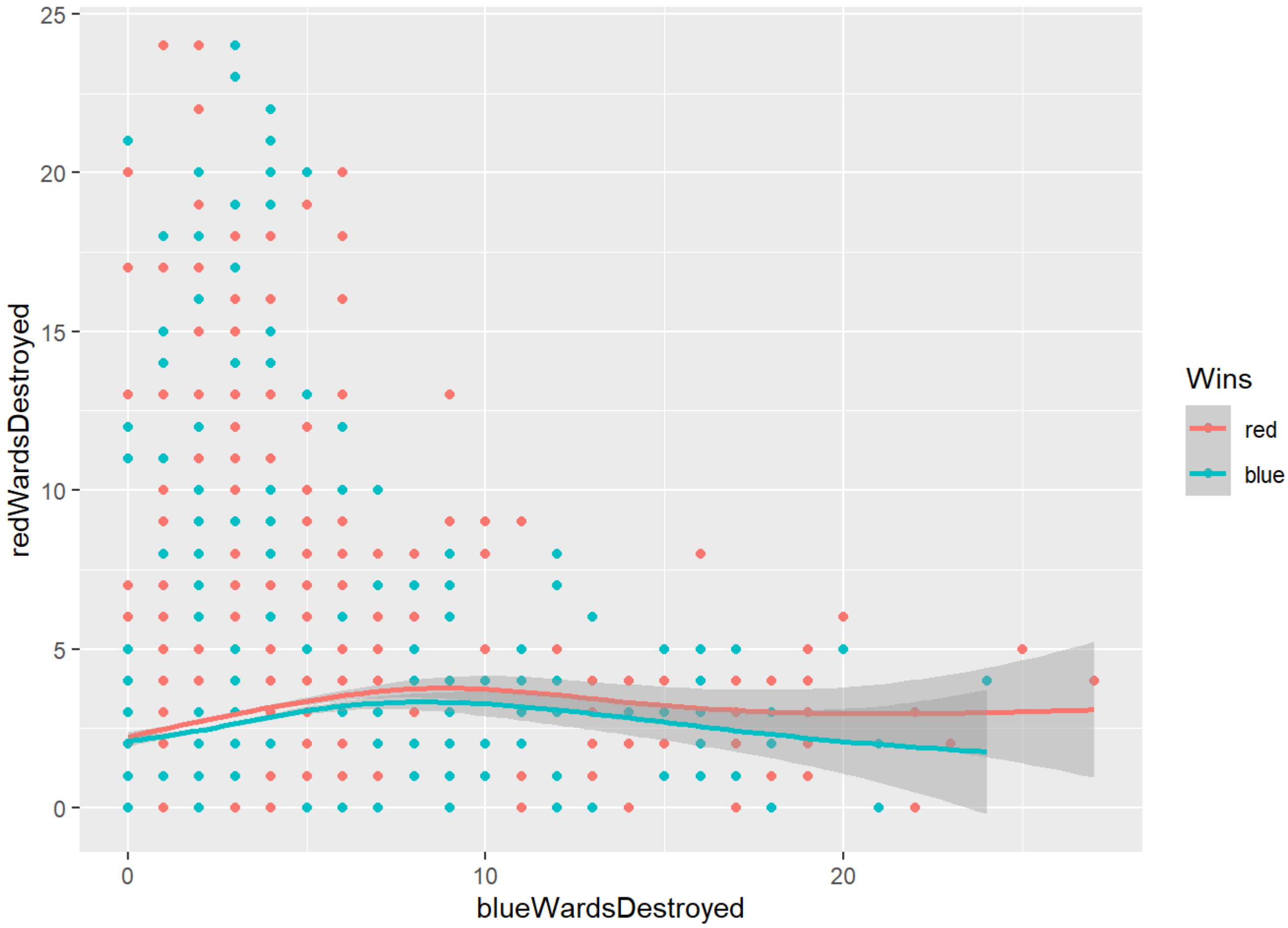



```
ggplot(data = df, aes(x = Wins, y = redWardsDestroyed, fill = Wins)) + geom_violin()
```




```
ggplot(data = df, aes(blueWardsDestroyed, redWardsDestroyed, color = Wins)) + geom_point() + geo
m_smooth()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```




```
Boostrap assumptions & t Test for the mean of blue wins, all three seem to show little correlati
on between performance improvement and placing & destroying wards.
```

```
x <- function(default=1){
  resample <- sample(dfBlueWins$blueWardsPlaced, length(dfBlueWins), replace = T)
  return(mean(resample))
}
data <- sapply(1:10000, x)

sum(data > mean(dfRedWins$blueWardsPlaced)) / length(data)
```

```
## [1] 0.4518
```

```
y <- function(default=1){
  resample <- sample(dfBlueWins$blueWardsDestroyed, length(dfBlueWins), replace = T)
  return(mean(resample))
}
data <- sapply(1:10000, y)
sum(data > mean(dfRedWins$blueWardsDestroyed)) / length(data)
```

```
## [1] 0.6989
```

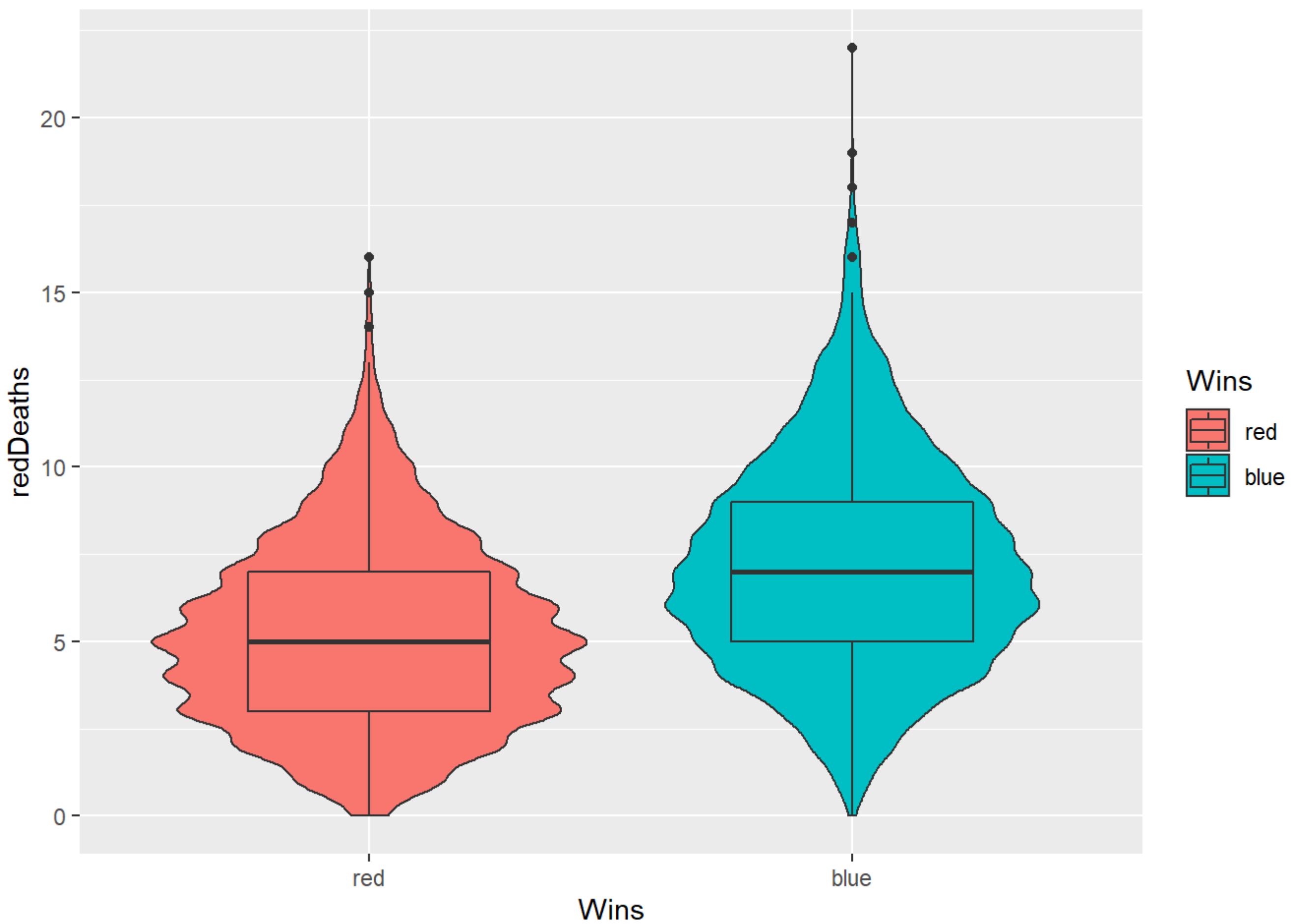
```
t.test(dfBlueWins$blueWardsPlaced,dfRedWins$blueWardsPlaced)$p.value
```

```
## [1] 0.9931048
```

# Kills & First Blood

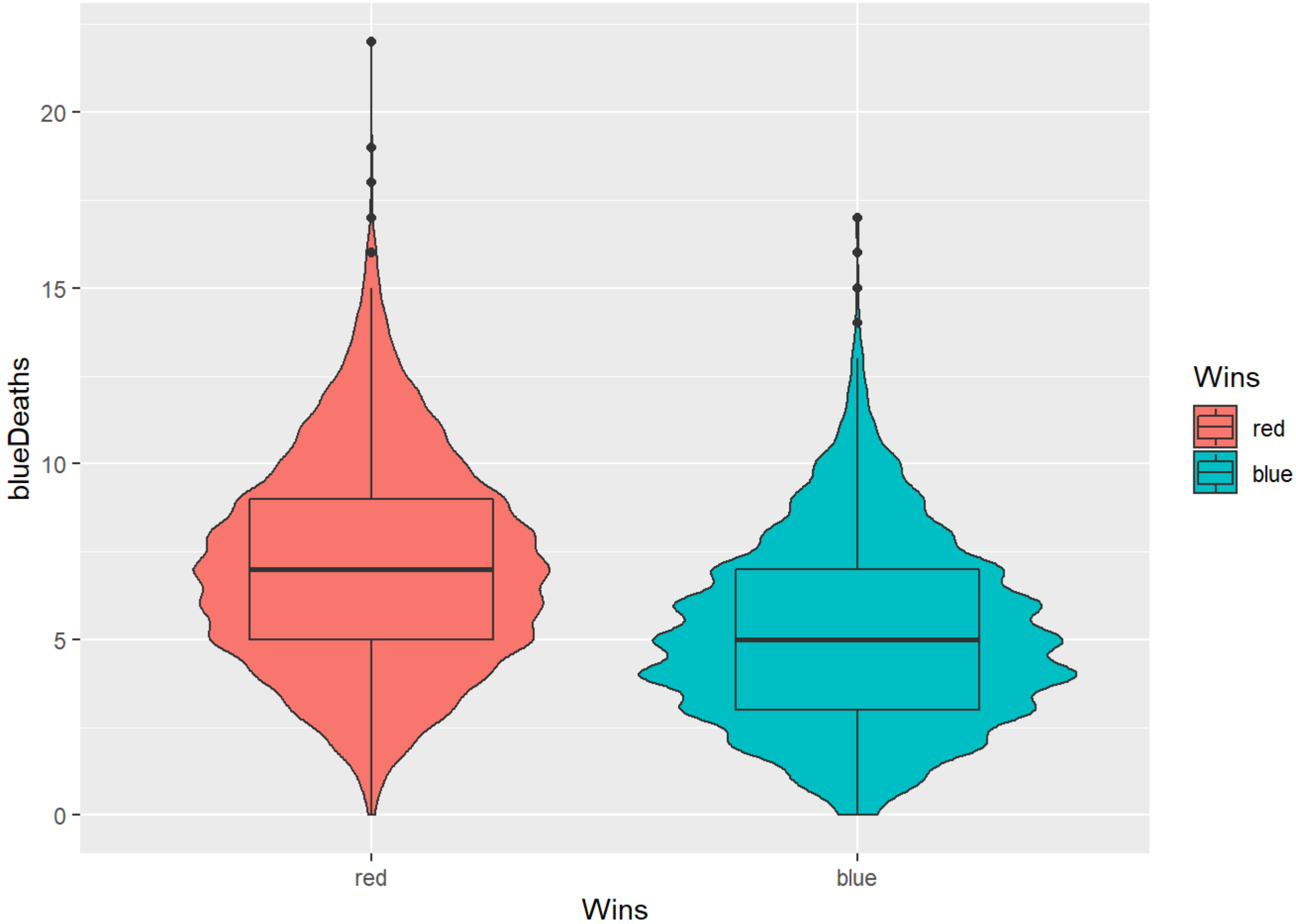
```
Based upon these violin / box plots, we can assume an inverse relationship between red deaths an
d red performance in game. The more deaths red has, the less their frequency of winning.
```

```
ggplot(data = df, aes(x = Wins, y = redDeaths, fill = Wins)) + geom_violin() + geom_boxplot(widt
h = 0.5)
```

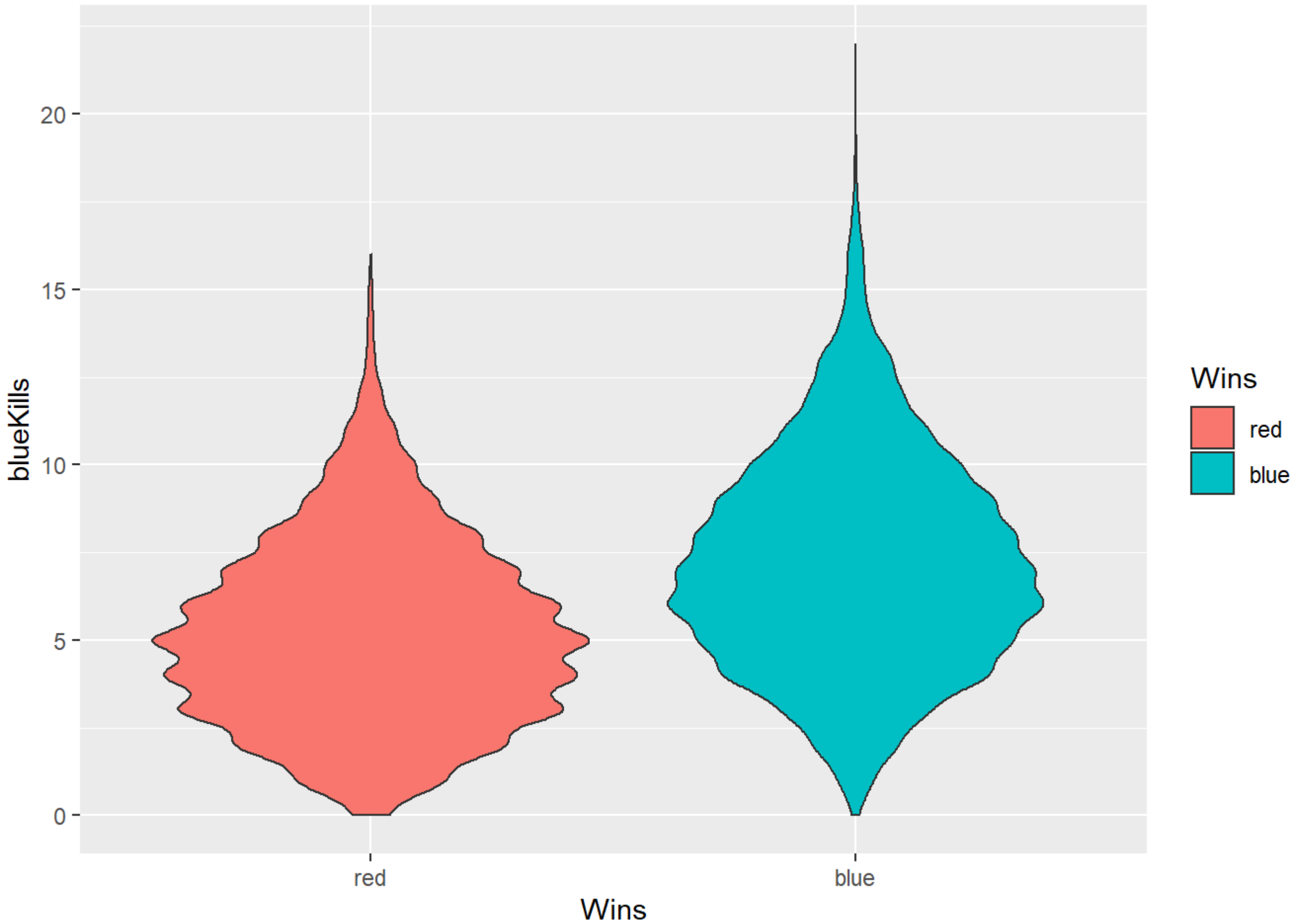



```
Deaths appear to have a more significant effect on the outcome of the game.The same assumption a
pplies to blue.
```

```
ggplot(data = df, aes(x = Wins, y = blueDeaths, fill = Wins)) + geom_violin() + geom_boxplot(wid
th = 0.5)
```




```
ggplot(data = df, aes(x = Wins, y = blueKills, fill = Wins)) + geom_violin()
```




# Conclusions & Further Questions

```
From our analysis, we can conclude that a number of factors influence a team's performance durin
g the game of league of legends, these factors acting as sort of predictors of a successful stra
tegy. Teams that successfully accomplished the main mechanics of the game such as creep score fa
rming, killing opposing players, and slaying elites and monsters tended to have a higher win rat
e than those who performed much lower in those aspects.
We also included factors such as warding that we initially thought would have a signficant impac
t on the direction of the game. From our analysis, however, we determined that these factors wer
e ultimately inconsequential to a team's performance, despite our initial beliefs.

Furthermore, we were left with some questions to reflect upon in our analysis. Our data set was
a conglomeration of a team, and did not detail the individual performance of each player. As suc
h, we could not get an accurate breakdown of how a team truly performed based upon the overview
given. For example, did a certain team member contribute more to the total kills of a team than
other players did? In relation to warding, the question also comes up as to the efficacy of a wa
rd. The data only counted the number of wards placed and broken. How many enemies did a ward spo
t? Could our data be slightly skewed if a player misplaced a ward? The dragon collection also do
es not specify which type of dragon was defeated, which subtly influences the benefits a team re
ceives. The data set also does not include the champions a team chose, which drastically influce
s gameplay style as some champions contain abilities that more effectively counter an enemy cham
pion.  In addition to this, our analysis only covered the in game factors of a team's performanc
e, but does not account for any additional human interaction as to why a team performed as did.
Since humans come from a variety of different statuses, a question to note would be how the play
er's status influenced their overall gameplay. How tired were they during their game? What foods
did they have? are they vision-impaired? The variety of factors and difficult in obtaining data
such as this is likely why it is not included.

Overall, the breakdown of our analysis boiled in on finding factors which contributed most to a
successful game in order to establish a review of the trends of successful, high-level, League o
f Legends gameplay.
```