LoL - Predictions

Lucas Welch

2023-01-28

//Observation: Problem:

//Observation: Take this data set and find which factor (blueWardsPlaced, blueWardsDestroyed, blueFirstBlood, etc) has the largest correlation to winning the game (blueWins) and explain your process briefly. Create a data visualization of your choice using this data set.

Input the following libraries (Not all will be used)

```r
library("corrplot")
```

```
## corrplot 0.92 loaded
```

```r
library("factoextra")
```

```
## Loading required package: ggplot2
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
library("ggplot2")
library("randomForest")
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
library("igraph")
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

```r
library("neuralnet")
library("tidyverse")
```

```
## ── Attaching packages
## ───────────────────────────────────────
## tidyverse 1.3.2 ──
## ✔ tibble  3.1.8      ✔ dplyr   1.0.10
## ✔ tidyr   1.3.0      ✔ stringr 1.5.0
## ✔ readr   2.1.2      ✔ forcats 0.5.2
## ✔ purrr   1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::as_data_frame() masks tibble::as_data_frame(), igraph::as_data_frame()
## ✖ dplyr::combine()       masks randomForest::combine()
## ✖ purrr::compose()       masks igraph::compose()
## ✖ dplyr::compute()       masks neuralnet::compute()
## ✖ tidyr::crossing()      masks igraph::crossing()
## ✖ dplyr::filter()        masks stats::filter()
## ✖ dplyr::groups()        masks igraph::groups()
## ✖ dplyr::lag()           masks stats::lag()
## ✖ randomForest::margin() masks ggplot2::margin()
## ✖ purrr::simplify()      masks igraph::simplify()
```

```
library("skimr")
library("caret")
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library("neuralnet")
```

Load in the dataset and look at the following variables

```
league_data <- read.csv("high_diamond_ranked_10min.csv")
```

```
colnames(league_data)
```

```
##  [1] "gameId"                       "blueWins"
##  [3] "blueWardsPlaced"              "blueWardsDestroyed"
##  [5] "blueFirstBlood"               "blueKills"
##  [7] "blueDeaths"                   "blueAssists"
##  [9] "blueEliteMonsters"            "blueDragons"
## [11] "blueHeralds"                  "blueTowersDestroyed"
## [13] "blueTotalGold"                "blueAvgLevel"
## [15] "blueTotalExperience"          "blueTotalMinionsKilled"
## [17] "blueTotalJungleMinionsKilled" "blueGoldDiff"
## [19] "blueExperienceDiff"           "blueCSPerMin"
## [21] "blueGoldPerMin"               "redWardsPlaced"
## [23] "redWardsDestroyed"            "redFirstBlood"
## [25] "redKills"                     "redDeaths"
## [27] "redAssists"                   "redEliteMonsters"
## [29] "redDragons"                   "redHeralds"
## [31] "redTowersDestroyed"           "redTotalGold"
## [33] "redAvgLevel"                  "redTotalExperience"
## [35] "redTotalMinionsKilled"        "redTotalJungleMinionsKilled"
## [37] "redGoldDiff"                  "redExperienceDiff"
## [39] "redCSPerMin"                  "redGoldPerMin"
```

```
skim_without_charts(league_data)
```

Data summary

Name	league_data
Number of rows	9879
Number of columns	40
________________________	
Column type frequency:	
numeric	40

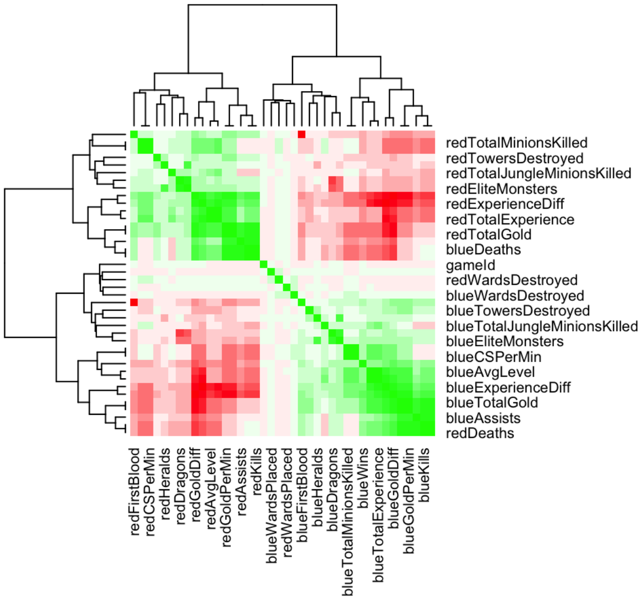
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	
gameId	0	1	4500084044.85	27573278.49	4295358071.0	4483301169.00	4510920346.0	4.521733e+09	452
blueWins	0	1	0.50	0.50	0.0	0.00	0.0	1.000000e+00	
blueWardsPlaced	0	1	22.29	18.02	5.0	14.00	16.0	2.000000e+01	
blueWardsDestroyed	0	1	2.82	2.17	0.0	1.00	3.0	4.000000e+00	
blueFirstBlood	0	1	0.50	0.50	0.0	0.00	1.0	1.000000e+00	
blueKills	0	1	6.18	3.01	0.0	4.00	6.0	8.000000e+00	
blueDeaths	0	1	6.14	2.93	0.0	4.00	6.0	8.000000e+00	
blueAssists	0	1	6.65	4.06	0.0	4.00	6.0	9.000000e+00	
blueEliteMonsters	0	1	0.55	0.63	0.0	0.00	0.0	1.000000e+00	
blueDragons	0	1	0.36	0.48	0.0	0.00	0.0	1.000000e+00	
blueHeralds	0	1	0.19	0.39	0.0	0.00	0.0	0.000000e+00	
blueTowersDestroyed	0	1	0.05	0.24	0.0	0.00	0.0	0.000000e+00	
blueTotalGold	0	1	16503.46	1535.45	10730.0	15415.50	16398.0	1.745900e+04	
blueAvgLevel	0	1	6.92	0.31	4.6	6.80	7.0	7.200000e+00	
blueTotalExperience	0	1	17928.11	1200.52	10098.0	17168.00	17951.0	1.872400e+04	
blueTotalMinionsKilled	0	1	216.70	21.86	90.0	202.00	218.0	2.320000e+02	
blueTotalJungleMinionsKilled	0	1	50.51	9.90	0.0	44.00	50.0	5.600000e+01	
blueGoldDiff	0	1	14.41	2453.35	-10830.0	-1585.50	14.0	1.596000e+03	
blueExperienceDiff	0	1	-33.62	1920.37	-9333.0	-1290.50	-28.0	1.212000e+03	
blueCSPerMin	0	1	21.67	2.19	9.0	20.20	21.8	2.320000e+01	
blueGoldPerMin	0	1	1650.35	153.54	1073.0	1541.55	1639.8	1.745900e+03	
redWardsPlaced	0	1	22.37	18.46	6.0	14.00	16.0	2.000000e+01	
redWardsDestroyed	0	1	2.72	2.14	0.0	1.00	2.0	4.000000e+00	
redFirstBlood	0	1	0.50	0.50	0.0	0.00	0.0	1.000000e+00	
redKills	0	1	6.14	2.93	0.0	4.00	6.0	8.000000e+00	
redDeaths	0	1	6.18	3.01	0.0	4.00	6.0	8.000000e+00	
redAssists	0	1	6.66	4.06	0.0	4.00	6.0	9.000000e+00	
redEliteMonsters	0	1	0.57	0.63	0.0	0.00	0.0	1.000000e+00	
redDragons	0	1	0.41	0.49	0.0	0.00	0.0	1.000000e+00	
redHeralds	0	1	0.16	0.37	0.0	0.00	0.0	0.000000e+00	
redTowersDestroyed	0	1	0.04	0.22	0.0	0.00	0.0	0.000000e+00	
redTotalGold	0	1	16489.04	1490.89	11212.0	15427.50	16378.0	1.741850e+04	
redAvgLevel	0	1	6.93	0.31	4.8	6.80	7.0	7.200000e+00	
redTotalExperience	0	1	17961.73	1198.58	10465.0	17209.50	17974.0	1.876450e+04	
redTotalMinionsKilled	0	1	217.35	21.91	107.0	203.00	218.0	2.330000e+02	
redTotalJungleMinionsKilled	0	1	51.31	10.03	4.0	44.00	51.0	5.700000e+01	
redGoldDiff	0	1	-14.41	2453.35	-11467.0	-1596.00	-14.0	1.585500e+03	
redExperienceDiff	0	1	33.62	1920.37	-8348.0	-1212.00	28.0	1.290500e+03	
redCSPerMin	0	1	21.73	2.19	10.7	20.30	21.8	2.330000e+01	
redGoldPerMin	0	1	1648.90	149.09	1121.2	1542.75	1637.8	1.741850e+03	

Use correlation matrix using heatmap to analyze variables associated with red team, blue team, and no correlation

```
res = cor(league_data)
heatmap(res, col = colorRampPalette(c("red", "white", "green"))(20), symm=TRUE, margins = c(10,10))
```



//Observation: Gives me a visual of the correlation between variables

Finding correlation that causes the winning rate for team blue

Create a new data set with selected columns

```
blue_team <- league_data |>
  select(blueKills, blueTotalGold, blueWins, blueTowersDestroyed, blueTotalMinionsKilled, blueHeralds)
```

Create a new column indicating win/loss status

```
blue_team2 <- blue_team |>
  mutate(BWins = ifelse(blueWins > 0, "Win", "Loss"))
```

[Test: x > 18,000 Gold]

Filter data to only include games with more than 18000 gold

Analyze the effect of having more than 18000 gold on your chances of winning

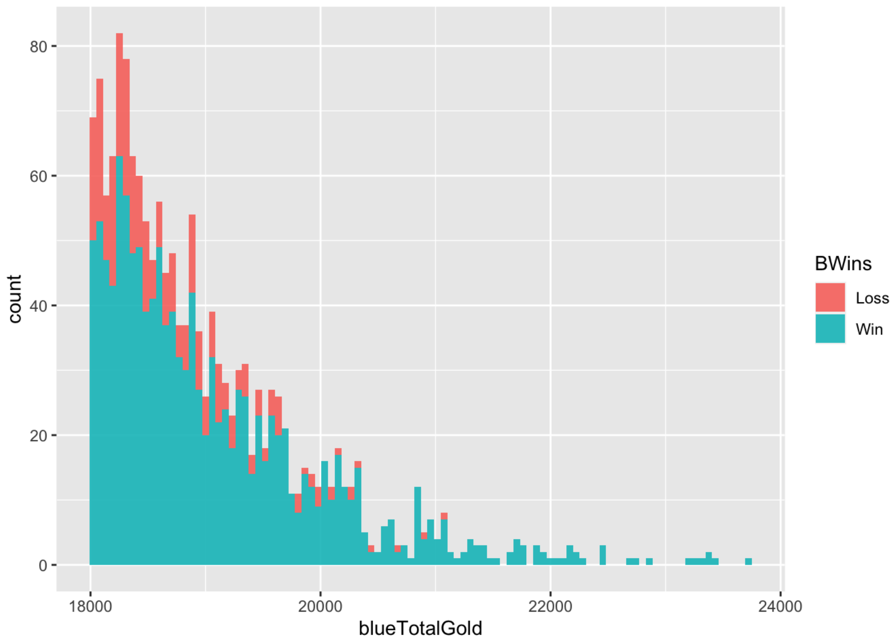
```
blue_team_wins <- blue_team2 |>
  filter(blueTotalGold > 18000)
```

Summarize win/loss rate for games with more than 18000 gold

```
more_than_18000 <- data.frame(winning_chance = sum(blue_team_wins$BWins == "Win") / nrow(blue_team_wins) * 100,
                              losing_chance = sum(blue_team_wins$BWins == "Loss") / nrow(blue_team_wins) * 100)
```

Plot histogram of blueTotalGold for games with more than 18000 gold

```
ggplot(blue_team_wins) +
  geom_histogram(mapping = aes(x = blueTotalGold, fill = BWins), bins = 100, alpha = 0.9)
```



//Observation: Having over 18000 gold gives you an 81% likelihood of winning the game and a 18% likelihood of losing it.

[Test: x < 16,000 Gold]

Filter data to only include games with less than 16000 gold

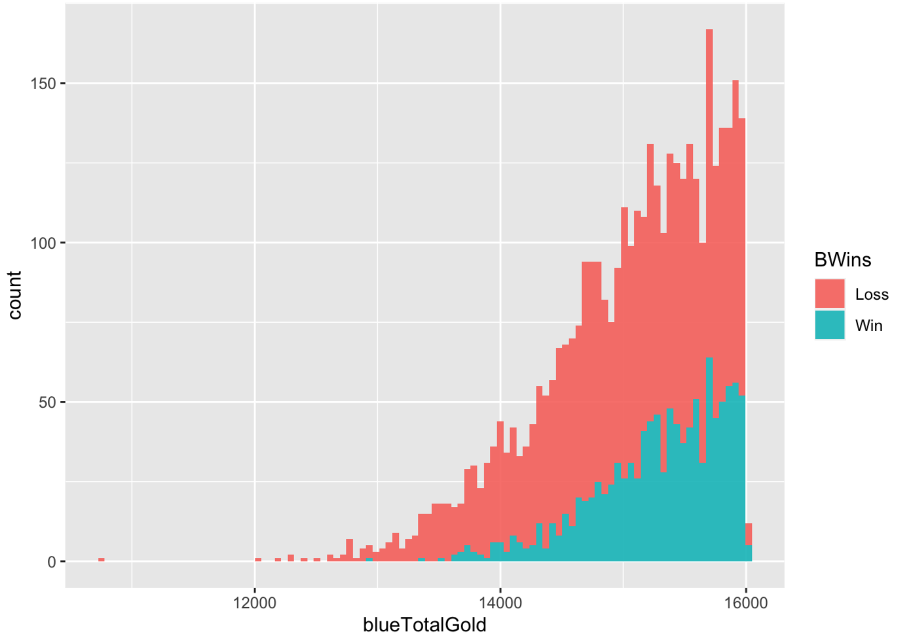
```
blue_team_downward_linear <- blue_team2 |>
  filter(blueTotalGold < 16000)
```

Summarize win/loss rate for games with less than 16000 gold

```
less_than_16000 <- data.frame(losing_chance = sum(blue_team_downward_linear$BWins == "Loss") / nrow(blue_team_dow
nward_linear) * 100,
                              winning_chance = sum(blue_team_downward_linear$BWins == "Win") / nrow(blue_team_dow
nward_linear) * 100)
```

Plot histogram of blueTotalGold for games with less than 16000 gold

```
ggplot(blue_team_downward_linear) +
  geom_histogram(mapping = aes(x = blueTotalGold, fill = BWins), bins = 100, alpha = 0.9)
```



//Observation: The analysis shows that if your gold is below 16000, there is a 71% likelihood of losing the game and a 38% chance of winning it.

[Test: Minion and Gold Correlation]

MORE gold from minions analysis

Create new data set with selected columns

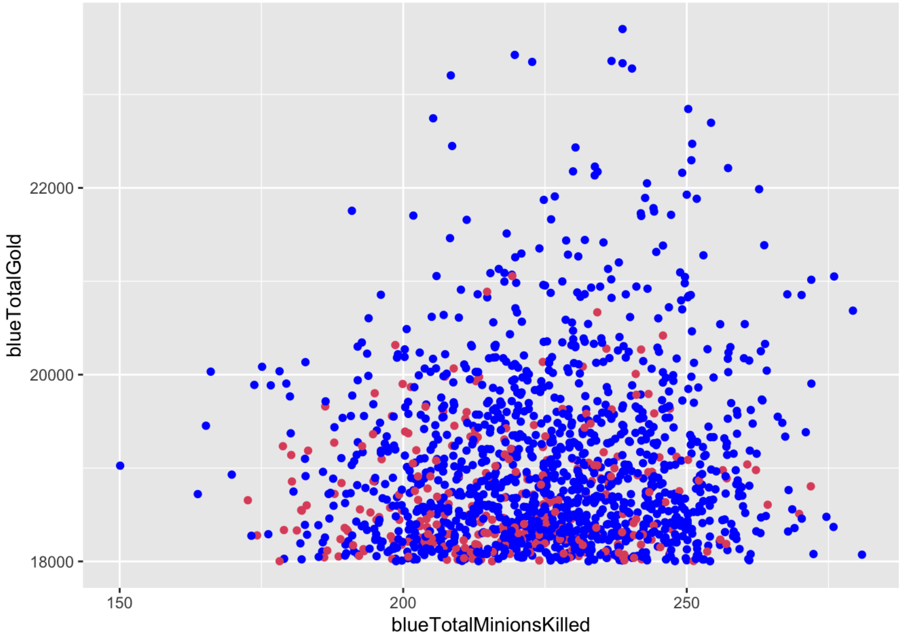
```
minions_gold <- league_data |>
  select(blueTotalGold, blueTotalMinionsKilled, blueWins)
```

Filter data to only include games with more than 18000 gold

```
minions_gold_v2 <- minions_gold |>
  filter(blueTotalGold > 18000)
```

Plot results

```
ggplot(data = minions_gold_v2) + geom_jitter(aes(x = blueTotalMinionsKilled, y = blueTotalGold, color = blueWin
s), show.legend = FALSE) + scale_color_gradient("Wins", low = "10", high = "blue")
```



//Observation: There is a positive slope for having more minions killed and gold acquired that lead to a win. Therefore, killing more minions results in more gold which allows players surpass 18,000 gold making their chances of winning higher.

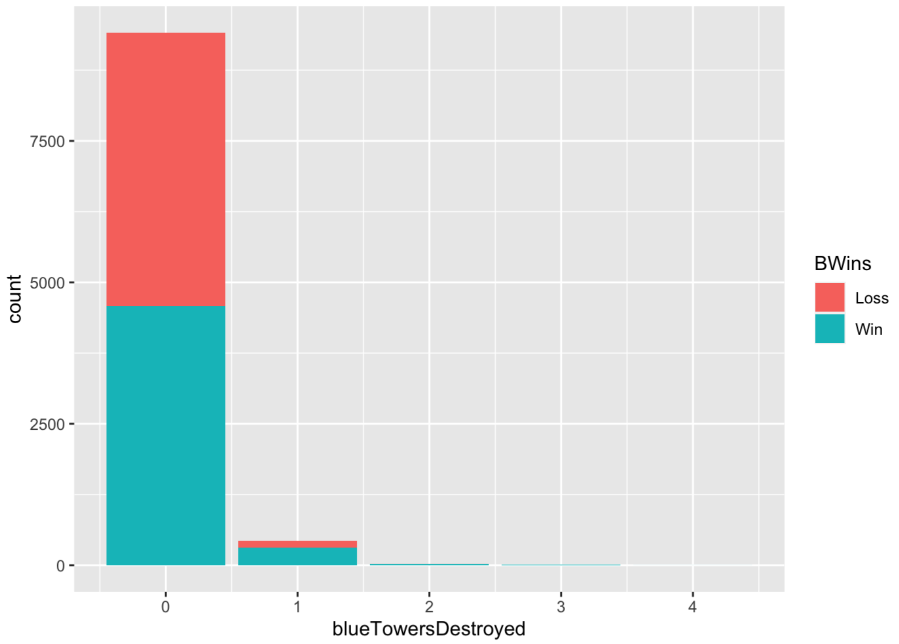
[Test: tower's Destroyed 0]

Destroying blue tower winning probability

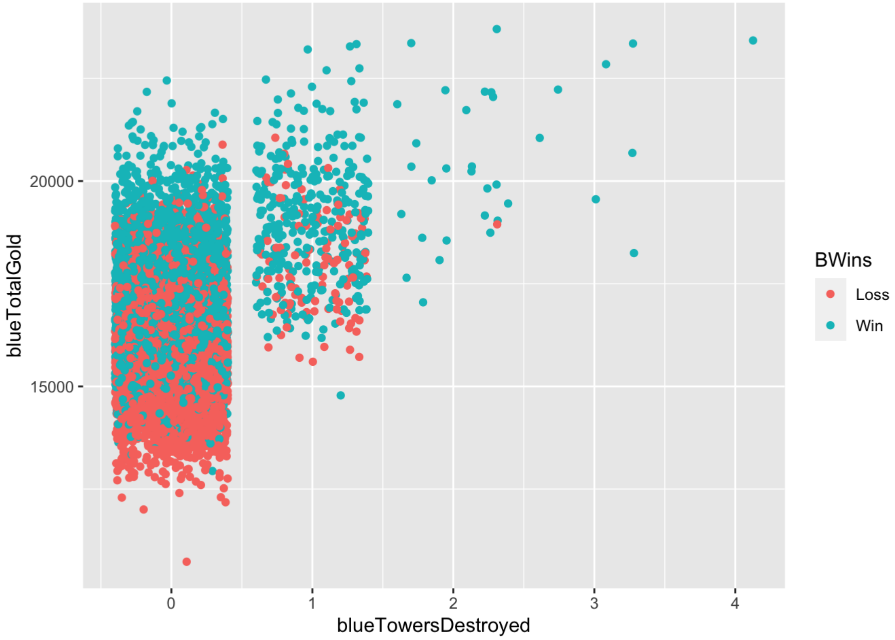
```
tower <- league_data |>
  select(blueTowersDestroyed, blueTotalGold, blueWins) |>
  mutate(BWins = ifelse(blueWins > 0, "Win", "Loss"))
```

```
tower_v1 <- tower |>
  filter(blueWins == 1 & blueTowersDestroyed == 0) |>
  summarize(count = n())

winning_chance <- 4580/9876 * 100
losing_chance <- 5296/9876 * 100
tower_0 <- data.frame(winning_chance, losing_chance)
```

```
ggplot(tower) +
  geom_bar(mapping = aes(x = blueTowersDestroyed, fill = BWins))
```



```
ggplot(tower) +
  geom_jitter(mapping = aes(x = blueTowersDestroyed, y = blueTotalGold, color = BWins))
```



//Observation: The probability of winning is estimated to be 46%, whereas the probability of losing stands at 54%. This significant imbalance can greatly impact someone’s probability of success and should therefore use other methods to increase probability.

[Test: towers destroyed 2+]

Destroy more than 1 tower probability

Filter data to only include cases where 2 or more blue towers were destroyed

Count the number of wins where 2 or more blue towers were destroyed

Calculate winning and losing chances

```
tower2 <- tower |>
  filter(blueTowersDestroyed >= 2)
str(tower2)
```

```
## 'data.frame':    35 obs. of  4 variables:
##  $ blueTowersDestroyed: int  2 2 2 2 2 3 3 2 2 2 ...
##  $ blueTotalGold      : int  19036 20352 18553 19453 19197 21051 22228 20018 21728 19915 ...
##  $ blueWins           : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ BWins              : chr  "Win" "Win" "Win" "Win" ...
```

```
win_count <- tower2 |>
filter(blueWins == 1) |>
summarise(count = n())

winning_chance <- 34/35 * 100
losing_chance <- 1/35 * 100
```

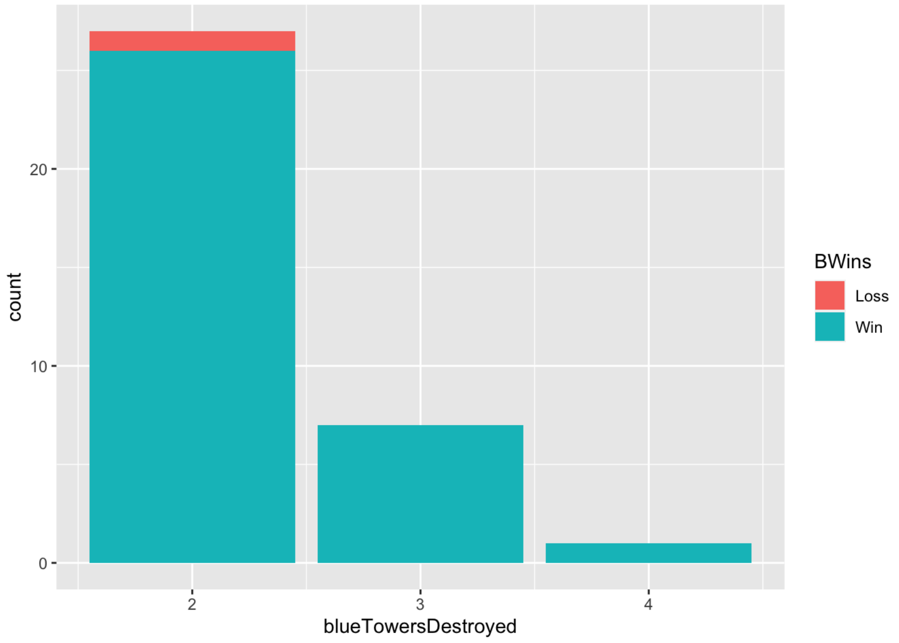
Store the results in a data frame and print

```
tower_more_than_or_equal_to_2 <- data.frame(winning_chance = winning_chance, losing_chance = losing_chance)
print(tower_more_than_or_equal_to_2)
```

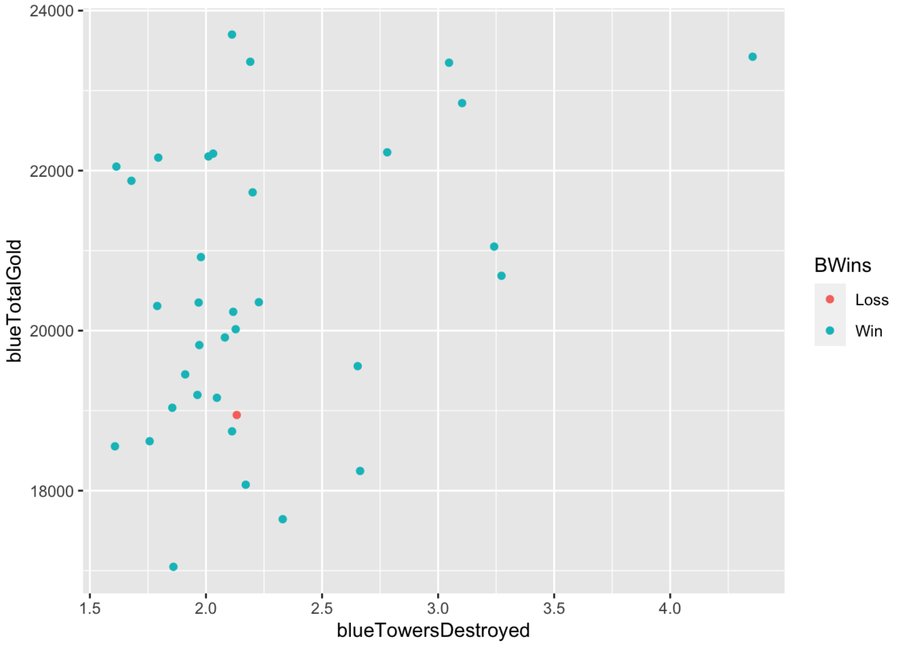
```
##   winning_chance losing_chance
## 1       97.14286      2.857143
```

Plottower destruction vs. win/loss results

```
ggplot(data = tower2) +
geom_bar(mapping = aes(x = blueTowersDestroyed, fill = BWins))
```



```
ggplot(tower2) +
geom_jitter(mapping = aes(x =blueTowersDestroyed, y = blueTotalGold, color = BWins))
```

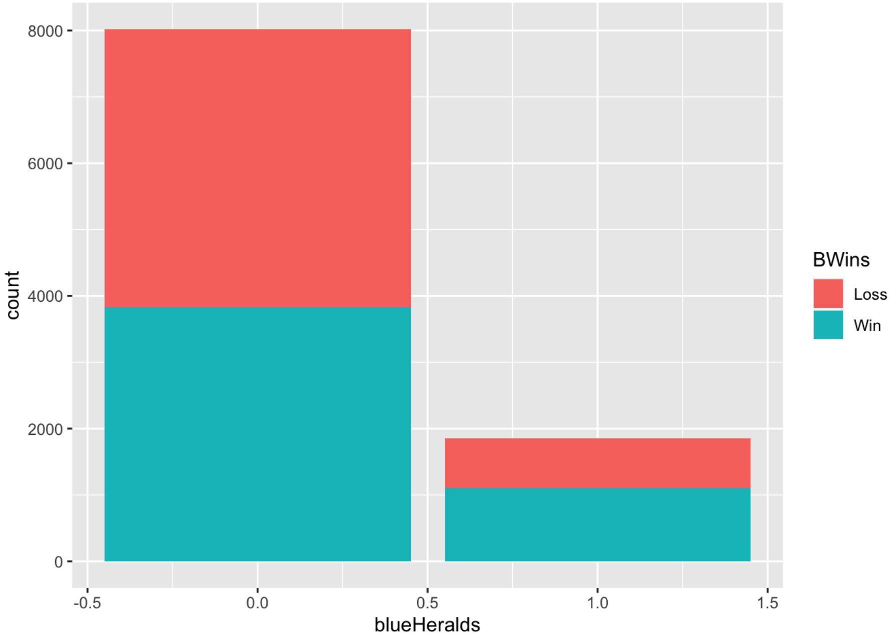


//Observation: The probability of winning is estimated to be 97%, whereas the probability of losing stands at 3%. This significant imbalance can greatly impact someone's probability of success.

[Test: Acquiring Herald] (Ran into some issues with the plots)

Heralds and win rate

```
ggplot(blue_team2, aes(x = blueHeralds, fill = BWins)) +
geom_bar()
```



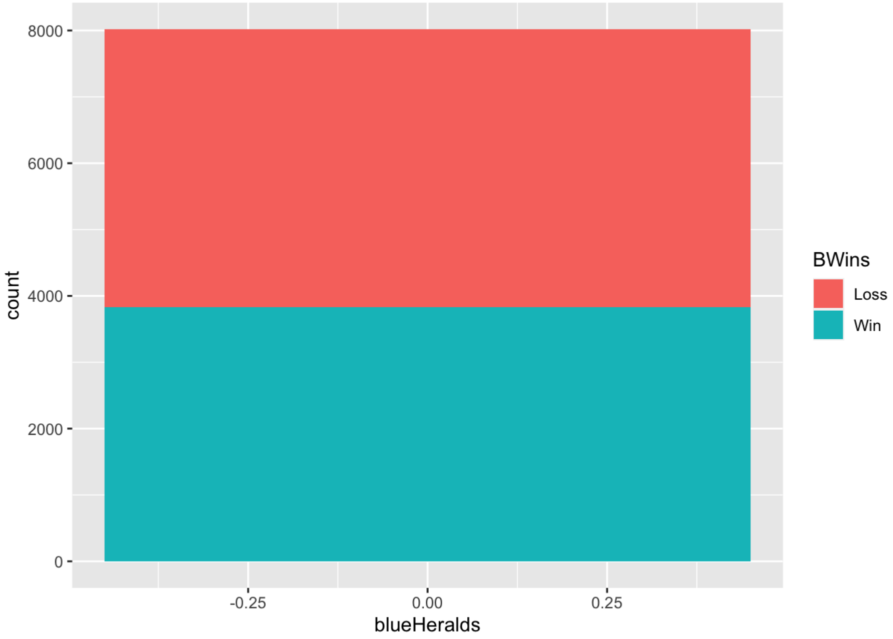
Players that did not get herald

```
no_herald_results <- blue_team2 |>
filter(blueHeralds == 0 & BWins == "Win") |>
summarise(no_herald_wins = n(), winning_chance = 100 * no_herald_wins / nrow(blue_team2))
print(no_herald_results)
```

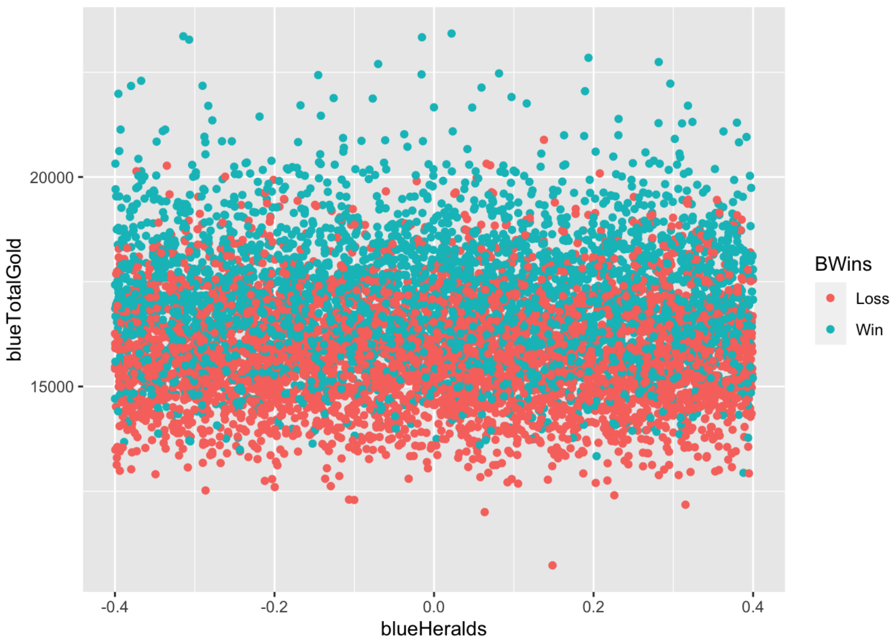
```
##   no_herald_wins winning_chance
## 1           3825       38.71849
```

```
herald <- blue_team2 |>
filter(blueHeralds == 0)
```

```
ggplot(herald, aes(x = blueHeralds, fill = BWins)) +
geom_bar()
```



```
ggplot(herald, aes(x = blueHeralds, y = blueTotalGold, color = BWins)) +
geom_jitter()
```



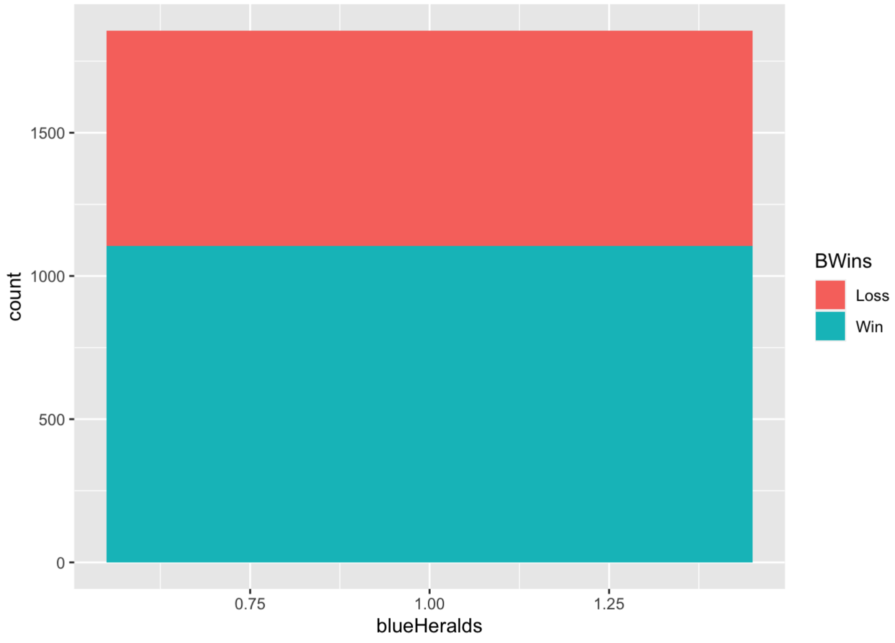
```
winning_chance <- 3825/8022

losing_chance <- 4197/8022

no_herald <- data.frame(winning_chance = 100 * winning_chance, losing_chance = 100 * losing_chance)
print(no_herald)
```

```
##   winning_chance losing_chance
## 1       47.68138      52.31862
```

Players that did get herald analysis

```
herald_df <- blue_team2 |>
  filter(blueHeralds == 1) |>
  select(BWins, blueHeralds)
```

```
ggplot(herald_df) +
  geom_bar(mapping = aes(x = blueHeralds, fill = BWins))
```



```
herald_win_rate <- data.frame(Win = 1105/1857 * 100, Lose = 752/1857 * 100)
print(herald_win_rate)
```

```
##        Win     Lose
## 1 59.50458 40.49542
```

//Observation: Players securing Herald increases your chances of winning a game by 59%, whereas the likelihood of losing is 40%.
//Observation: Players that did not get herald have a 48% of winning the game and a 52% of losing the game.

//Observation: In conclusion, having more towers, gold, minion death, and objective lead contributes to higher odds of winning. Acquiring Herald strengthens your chances of winning by adding to your existing advantage of towers, gold, and objective lead.