Gambling the Game of Probabilities that may Provide Insight into Non Probablitistic Behavior

Final Project

Gabrielle Young

2024-03-22

Recently, I have noticed a spike in gambling, specifically online gambling, in friend groups I associate with. This could be because I am from Las Vegas, or maybe because the majority of my friends have now turned 21 and can legally gamble, or maybe for some other reason I have not considered. Perhaps it is because gambling is addictive. From my understanding, gambling can be just as addictive, if not more addictive than actual substances.

I will be exploring this dataset I found on Kaggle: https://www.kaggle.com/datasets/kingabzpro/gambling-behavior-bustabit/data

This dataset contains behavior of gamblers using the online platform called Bustabit. This data was collected from 10/31/2016 to 12/10/2016.

The rules of the game are that you bet money in Bits and you must cash out before the game "busts." Wins are calculated using the amount of the bet multiplied by the amount cashed out. For example, you bet 10 and you cash out at 3.5, so your win would be 35 minus what you put in, so 25. Bonuses are also added and must be multiplied by the bet. On Bustabit, the house also has a slight advantage, where for every 1 out of 100 games, all players bust.

The dataset at a glance:

- 50001 rows; 43% NAs, meaning the player has busted before they were able to cash out. They chose a cash out value to play until, but busted before it was reached.
- 9 variables: ID, GameID, Username, Bet, CashedOut, Bonus, Profit, BustedAt, PlayDate

Key variables of interest:

- **GameID:** an identifier randomized to match the exact time of play for each player login that has a set BustedAt multiplyer value
- **Username:** contains the unique nametag of each player
- **Bet:** the amount of bitcoin the player pays
- **CashedOut:** the multiplier value the player ends the game with that can be used to find player's Total Win
- **Bonus:** the percentage (as a decimal) for each game the player is rewarded with
- **Profit:** amount of bitcoin the player walks away with (calculated as (Bet x CashedOut) + (Bet + Bonus) - Bet); Profit of NA is 0, Profit of 0 is a loss or a "bust."
- **BustedAt:** where the multiplier has been randomly set to "bust"
- **PlayDate:** Year, month, day and time (Hours, minutes, seconds) of play

Peep the Dataset

```
## Rows: 50,000
## Columns: 9
## $ Id        <int> 14196549, 10676217, 15577107, 25732127, 17995432, 14147823, ~
## $ GameID    <int> 3366002, 3343882, 3374646, 3429241, 3389174, 3365723, 334455~
## $ Username  <chr> "papai", "znay22", "rrrrrrrr", "sanya1206", "ADM", "afrod", ~
## $ Bet       <int> 5, 3, 4, 10, 50, 2, 1, 8, 2000, 1000, 6, 1, 24, 584, 15, 15,~
## $ CashedOut <dbl> 1.20, NA, 1.33, NA, 1.50, NA, 1.05, NA, 1.20, 1.05, NA, NA, ~
## $ Bonus     <dbl> 0.00, NA, 3.00, NA, 1.40, NA, 4.00, NA, 0.00, 0.00, NA, NA, ~
## $ Profit    <dbl> 1.00, NA, 1.44, NA, 25.70, NA, 0.09, NA, 400.00, 50.00, NA, ~
## $ BustedAt  <dbl> 8.24, 1.40, 3.15, 1.63, 2.29, 1.04, 1.05, 1.13, 9.14, 10.09,~
## $ PlayDate  <chr> "2016-11-20T19:44:19Z", "2016-11-14T14:21:50Z", "2016-11-23T~
```

New Variables:

- **Time** (separated from PlayDate variable)
- **Date** (separated from PlayDate variable)
- **FrequentPlayer** (binary variable I will calculate from Username)
- **NumberOfPlays** (calculated from Username)
- **Win** (binary variable where 1 = Profit greater than 0, 0 = Profit equal to 0)
- **ReturningPlayer** (calculated by Username where at player has played at LEAST one time in the past prior to the current date of play)
- **AverageBet** (calculated using Username and Bet)
- **AverageCashOut** (calculated using Username and CashedOut)
- **PlayerNumber** (Player 1, Player 2, etc.; derived from GameID and Username)

New Dataset:

```
## Rows: 50,000
## Columns: 17
## $ Id              <int> 14196549, 10676217, 15577107, 25732127, 17995432, 1414~
## $ GameID          <int> 3366002, 3343882, 3374646, 3429241, 3389174, 3365723, ~
## $ Username        <chr> "papai", "znay22", "rrrrrrrr", "sanya1206", "ADM", "af~
## $ Bet             <int> 5, 3, 4, 10, 50, 2, 1, 8, 2000, 1000, 6, 1, 24, 584, 1~
## $ CashedOut       <dbl> 1.20, 0.00, 1.33, 0.00, 1.50, 0.00, 1.05, 0.00, 1.20, ~
## $ Bonus           <dbl> 0.00, 0.00, 3.00, 0.00, 1.40, 0.00, 4.00, 0.00, 0.00, ~
## $ Profit          <dbl> 1.00, 0.00, 1.44, 0.00, 25.70, 0.00, 0.09, 0.00, 400.0~
## $ BustedAt        <dbl> 8.24, 1.40, 3.15, 1.63, 2.29, 1.04, 1.05, 1.13, 9.14, ~
## $ PlayDate        <chr> "2016-11-20T19:44:19Z", "2016-11-14T14:21:50Z", "2016-~
## $ NumberOfPlays   <int> 14, 150, 18, 3, 16, 100, 62, 72, 58, 2, 112, 40, 5, 8,~
## $ ReturningPlayer <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, ~
## $ AverageBet      <dbl> 16.214286, 22.613333, 136.555556, 60.000000, 910.93750~
## $ AverageCashOut  <dbl> 0.9285714, 0.8738000, 1.1216667, 0.9066667, 1.0312500,~
## $ Date            <date> 2016-11-20, 2016-11-14, 2016-11-23, 2016-12-08, 2016-~
## $ Time            <time> 19:44:19, 14:21:50, 06:39:15, 18:13:55, 08:14:48, 17:~
## $ Win             <dbl> 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, ~
## $ FrequentPlayer  <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, ~
```

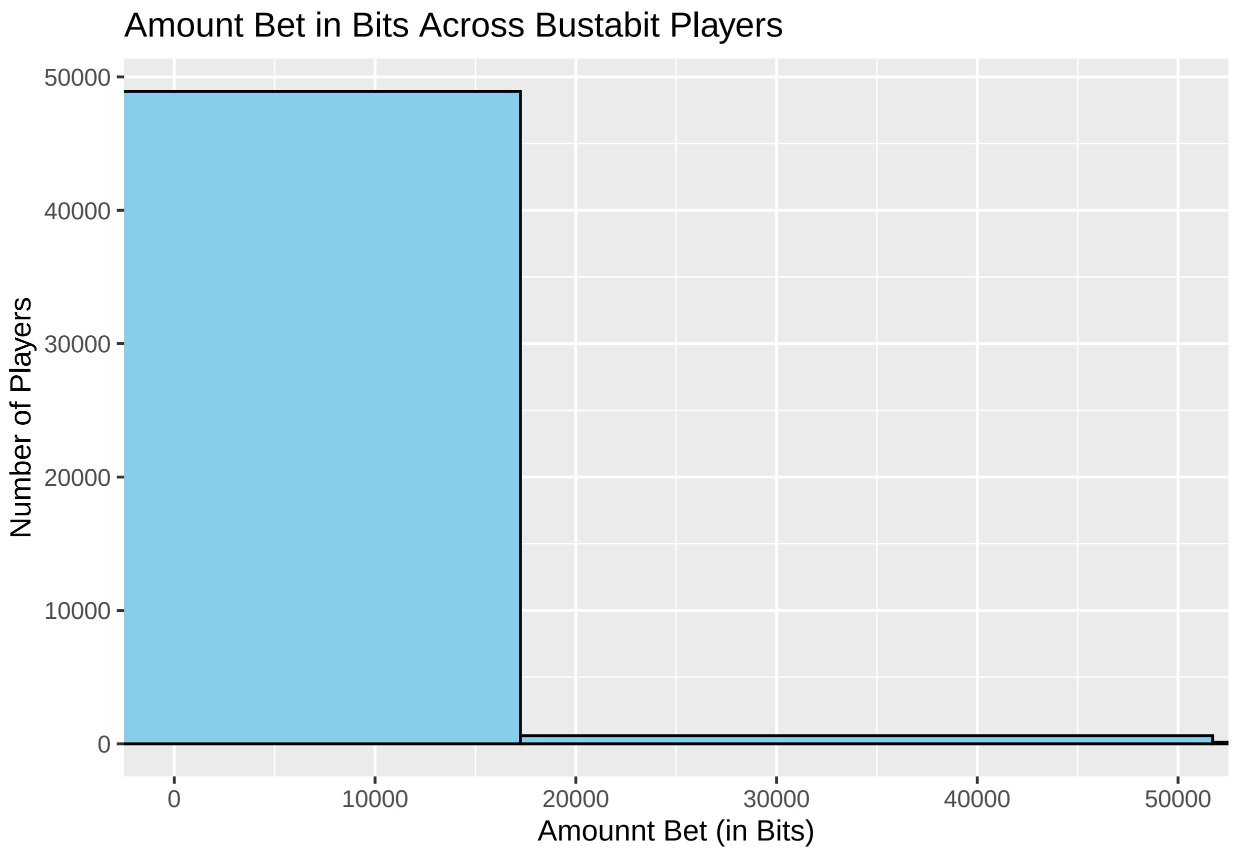
This preprocessing involved: - Coding NAs as 0 because these values were not invaluable. Within the data, each NA represented an instance where a bet was made, but the player busted, and lost. Meaning, they made a profit, cash out, and bonus of 0. - Creating the new variables to represent qualities of players that originally were not uncovered - Taking a 20% proportion of the data to work with because this data was originally 50,000 rows

Aims

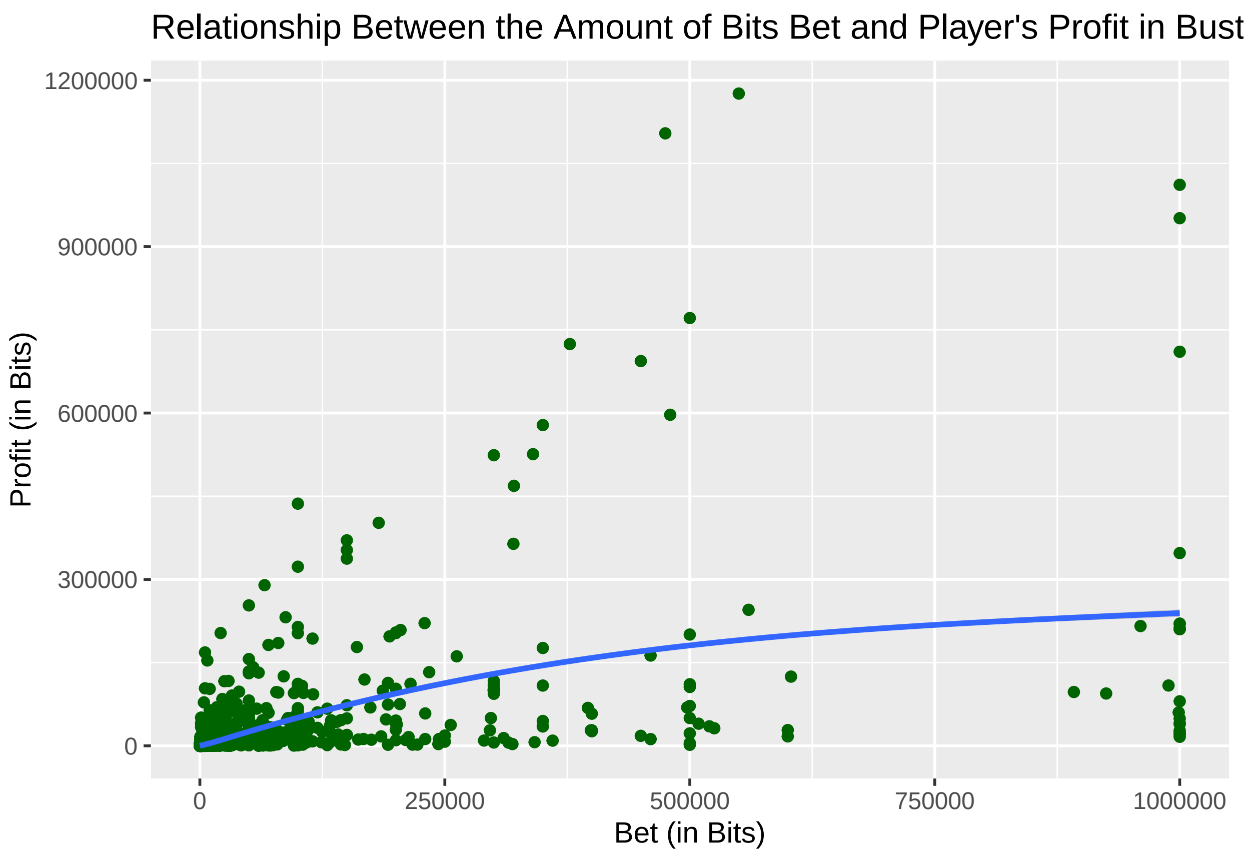
- **Aim 1:** Predict wins and loses of gambling.
- **Aim 2:** Predict the "types of gamblers" players may be.
- **Aim 3:** Explore the win or loss outcome two players playing the same game.

Exploratory Data Analysis

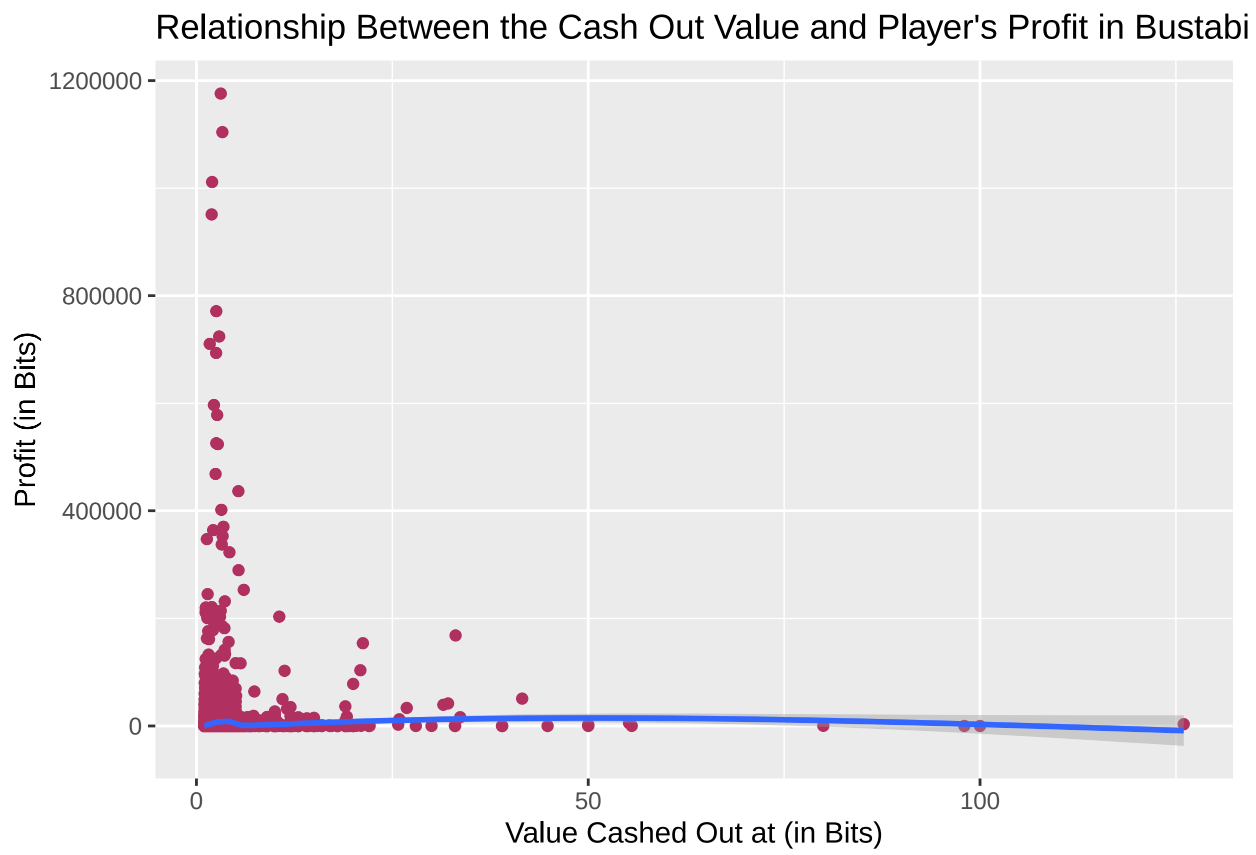
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



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

## Warning: Removed 21266 rows containing non-finite outside the scale range
## (`stat_smooth()`).

## Warning: Removed 21266 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



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

## Warning: Removed 21266 rows containing non-finite outside the scale range
## (`stat_smooth()`).
## Removed 21266 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

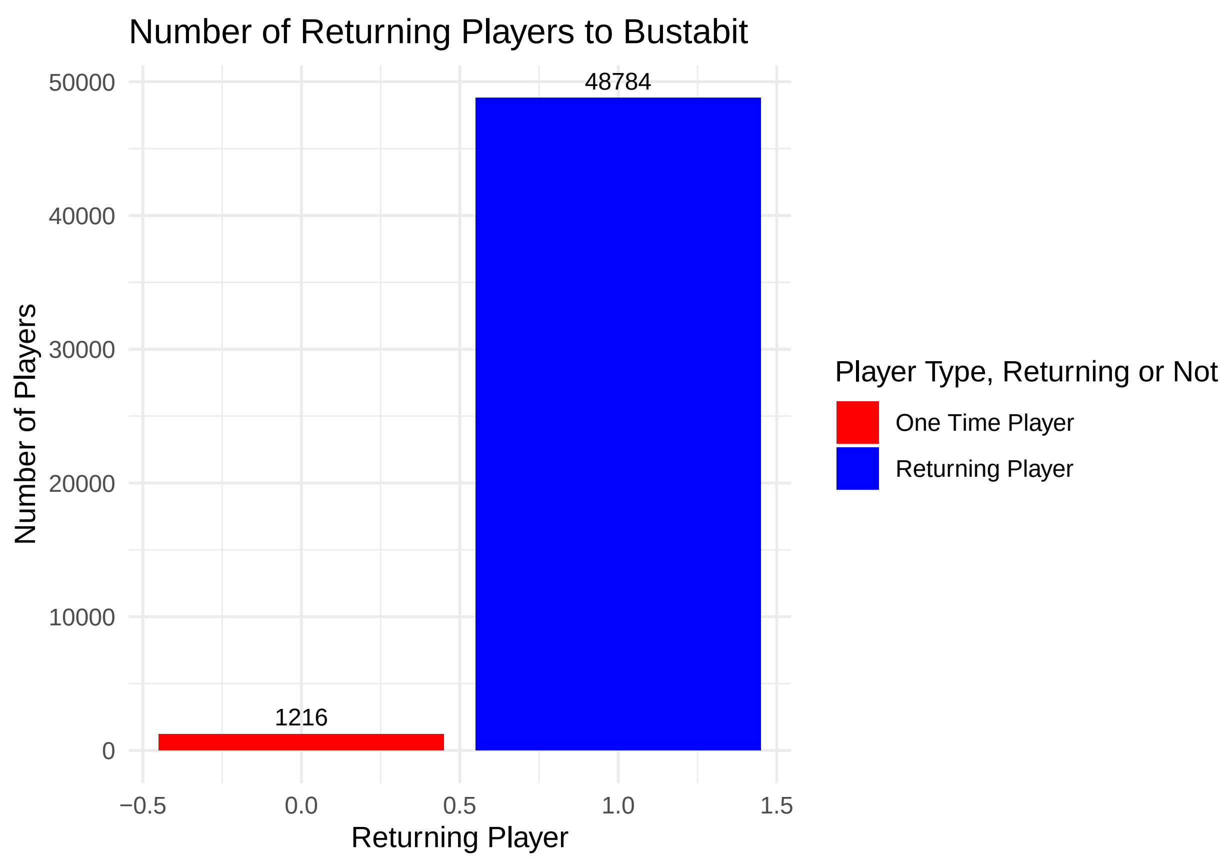


- Are wins and losses based on the amount bet?

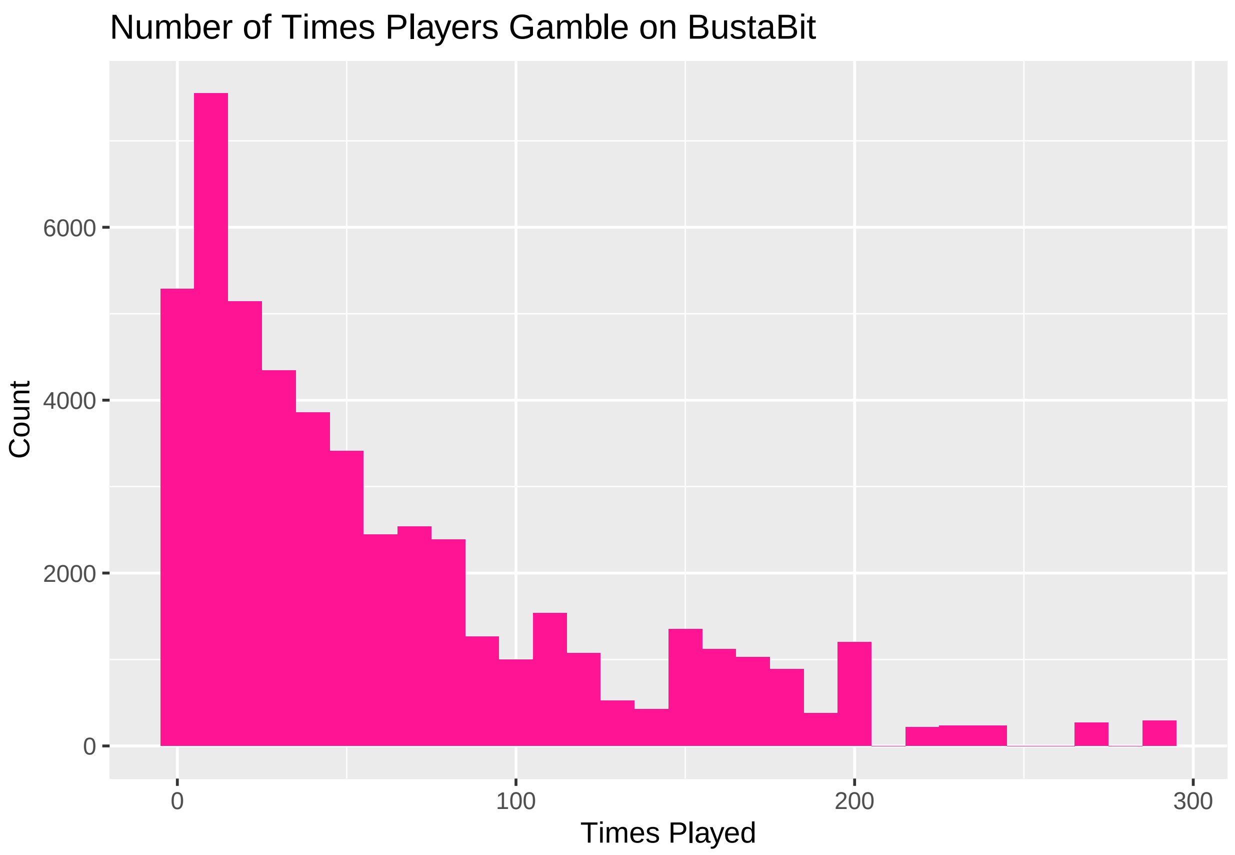
Trends: - majority of players bet smaller amounts - the more players bet, the higher their profits are

Are there Types of Players?

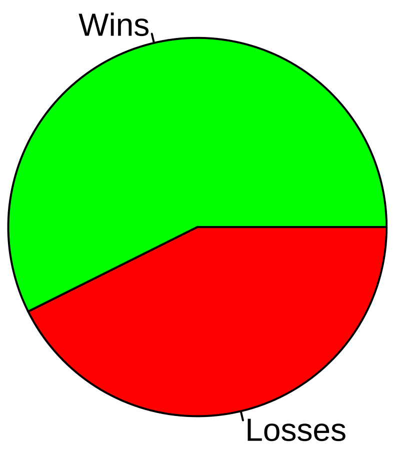
```
## Warning: `stat(count)` was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(count)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



There are many different habits across players. It would be interesting to see if there are groups of players with similar gambling habits.



Overall, we seem to have more Wins than Losses when looking at all games within the dataset. However, will we be able to predict these Wins and Losses?

Aim 1: Predict wins and loses of gambling

Predictors:

- Bet
- Bet, PlayDate, Bet, Bonus, Frequent plalyer, Number of plays, Average bet, Average Cash out

Outcome Variable of interest:

- Profit (in bits)

Modeling with Random Forest and Boosted Tree Models

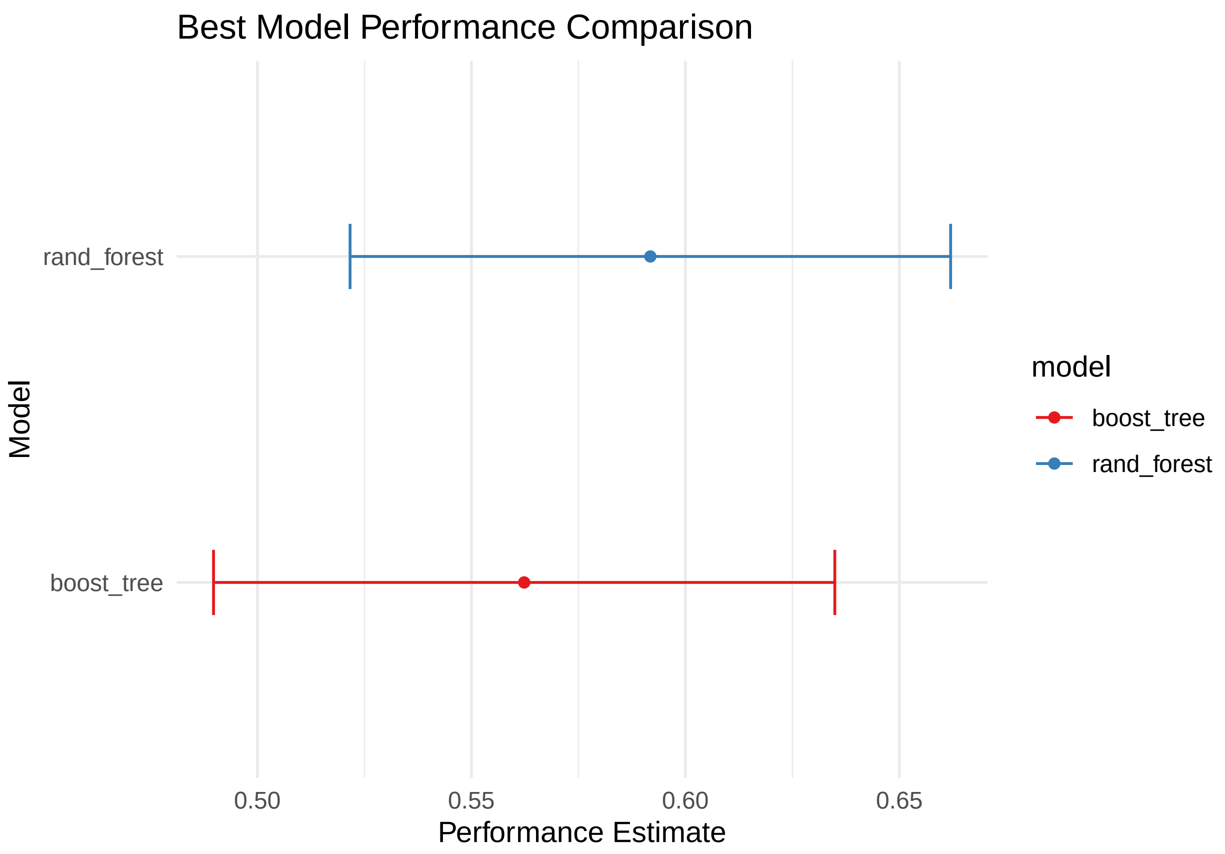
- perform well on high-dimensional data
- we want to make a comparision between the performance of the two models

```
## # A tibble: 4 x 9
##   wflow_id        .config preproc model .metric .estimator    mean     n std_err
##   <chr>           <chr>   <chr>   <chr> <chr>   <chr>        <dbl> <int>   <dbl>
## 1 base_random_fo~ Prepro~ recipe  rand~ rmse    standard   1.42e+4    10 1.70e+3
## 2 base_random_fo~ Prepro~ recipe  rand~ rsq     standard   2.46e-1    10 7.26e-2
## 3 base_boosted_t~ Prepro~ recipe  boos~ rmse    standard   1.44e+4    10 1.62e+3
## 4 base_boosted_t~ Prepro~ recipe  boos~ rsq     standard   2.50e-1    10 8.06e-2
```

```
## # A tibble: 4 x 9
##   wflow_id        .config preproc model .metric .estimator    mean     n std_err
##   <chr>           <chr>   <chr>   <chr> <chr>   <chr>        <dbl> <int>   <dbl>
## 1 base_random_fo~ Prepro~ recipe  rand~ rmse    standard   1.09e+4    10 1.28e+3
## 2 base_random_fo~ Prepro~ recipe  rand~ rsq     standard   5.81e-1    10 6.41e-2
## 3 base_boosted_t~ Prepro~ recipe  boos~ rmse    standard   1.14e+4    10 1.61e+3
## 4 base_boosted_t~ Prepro~ recipe  boos~ rsq     standard   6.06e-1    10 7.83e-2
```

```
## # A tibble: 2 x 7
##   wflow_id         .config              preproc model      .metric  mean std_err
##   <chr>            <chr>                <chr>   <chr>      <chr>   <dbl>   <dbl>
## 1 fe_random_forest Preprocessor1_Model1 recipe  rand_fore~ rsq     0.592  0.0702
## 2 fe_boosted_tree  Preprocessor1_Model3 recipe  boost_tree rsq     0.562  0.0726
```

Comparing the Models



Random Forest model had better overall performance, so we will use this model on the test set

Performance on the Test Data

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard   11558.
## 2 rsq     standard       0.419
## 3 mae     standard     904.
```

- This model has high error and is making predictions that are not precise
- High r-mean squared error, low r-squared value, and high mean absolute error
- It is difficult, if not impossible, to make accurate predictions of profit with any variables, which stands to prove the idea that gambling is purely probabilistic

Aim 2: Predict the "types of players" individuals may be

Predictors:

- Bet, PlayDate, Bet, Bonus, Frequent plalyer, Number of plays, Average bet, Average Cash out

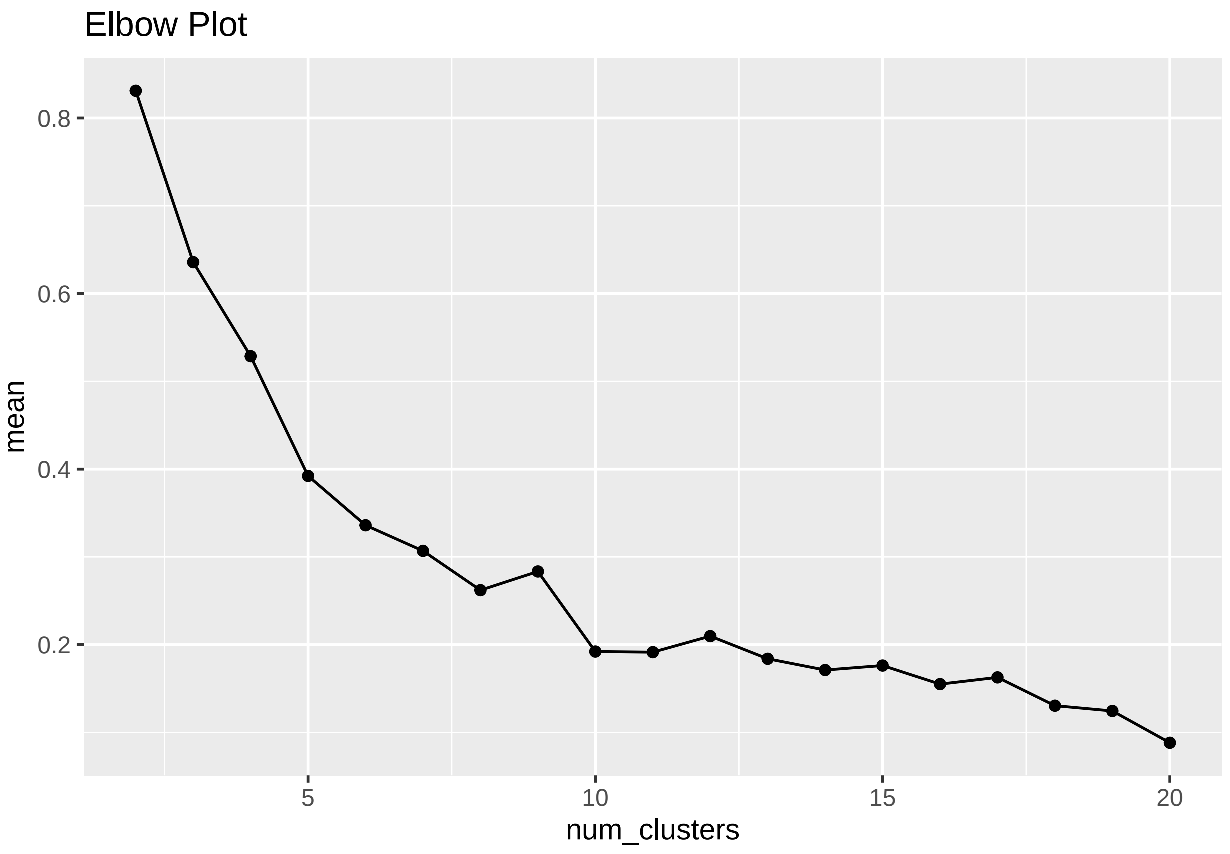
Outcome Variables:

- Clusters that may illustrate different "Types of Players"

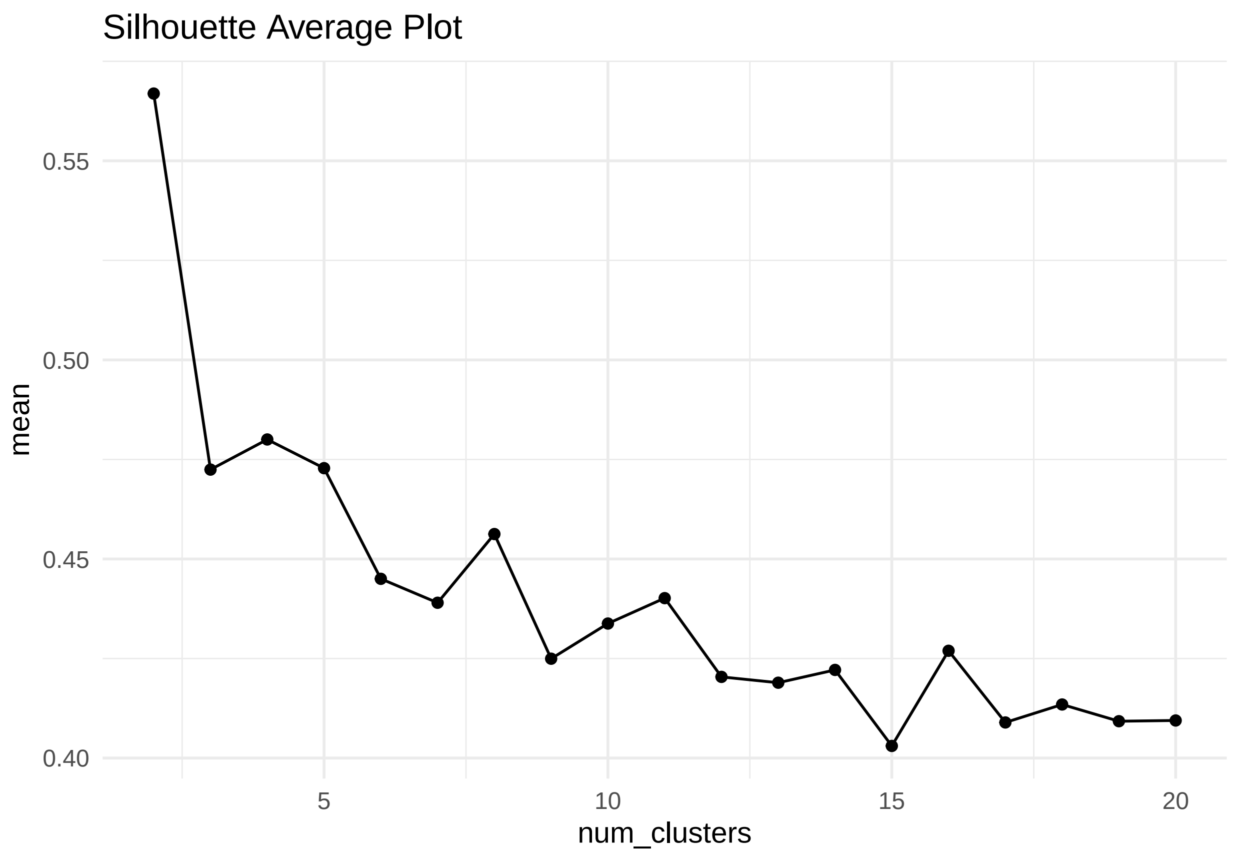
Modeling with K-Means Clustering Analysis

- We want to visualize the trends within high-dimensional data, where we may see patterns emerge that may represent types of players

```
## # A tibble: 38 x 7
##    num_clusters .metric         .estimator  mean     n std_err .config
##           <int> <chr>           <chr>      <dbl> <int>   <dbl> <chr>
##  1            2 silhouette_avg standard   0.567     6  0.0987 Preprocessor1_Mod~
##  2            2 sse_ratio      standard   0.831     6  0.0213 Preprocessor1_Mod~
##  3            3 silhouette_avg standard   0.472     6  0.0587 Preprocessor1_Mod~
##  4            3 sse_ratio      standard   0.636     6  0.0273 Preprocessor1_Mod~
##  5            4 silhouette_avg standard   0.480     6  0.0115 Preprocessor1_Mod~
##  6            4 sse_ratio      standard   0.529     6  0.0101 Preprocessor1_Mod~
##  7            5 silhouette_avg standard   0.473     6  0.0374 Preprocessor1_Mod~
##  8            5 sse_ratio      standard   0.392     6  0.0338 Preprocessor1_Mod~
##  9            6 silhouette_avg standard   0.445     6  0.0299 Preprocessor1_Mod~
## 10            6 sse_ratio      standard   0.336     6  0.0304 Preprocessor1_Mod~
## # i 28 more rows
```



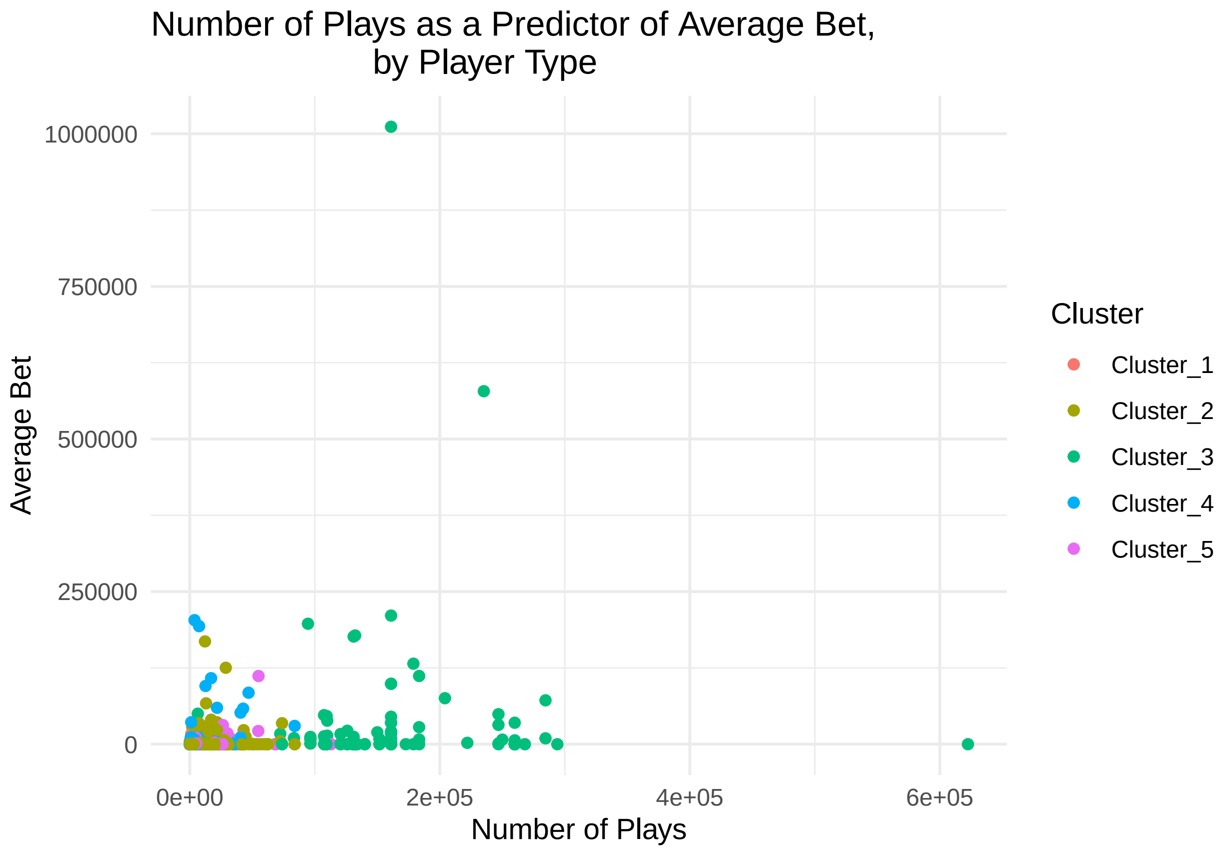
About 5 clusters seem optimal based on the curve of the plot



-

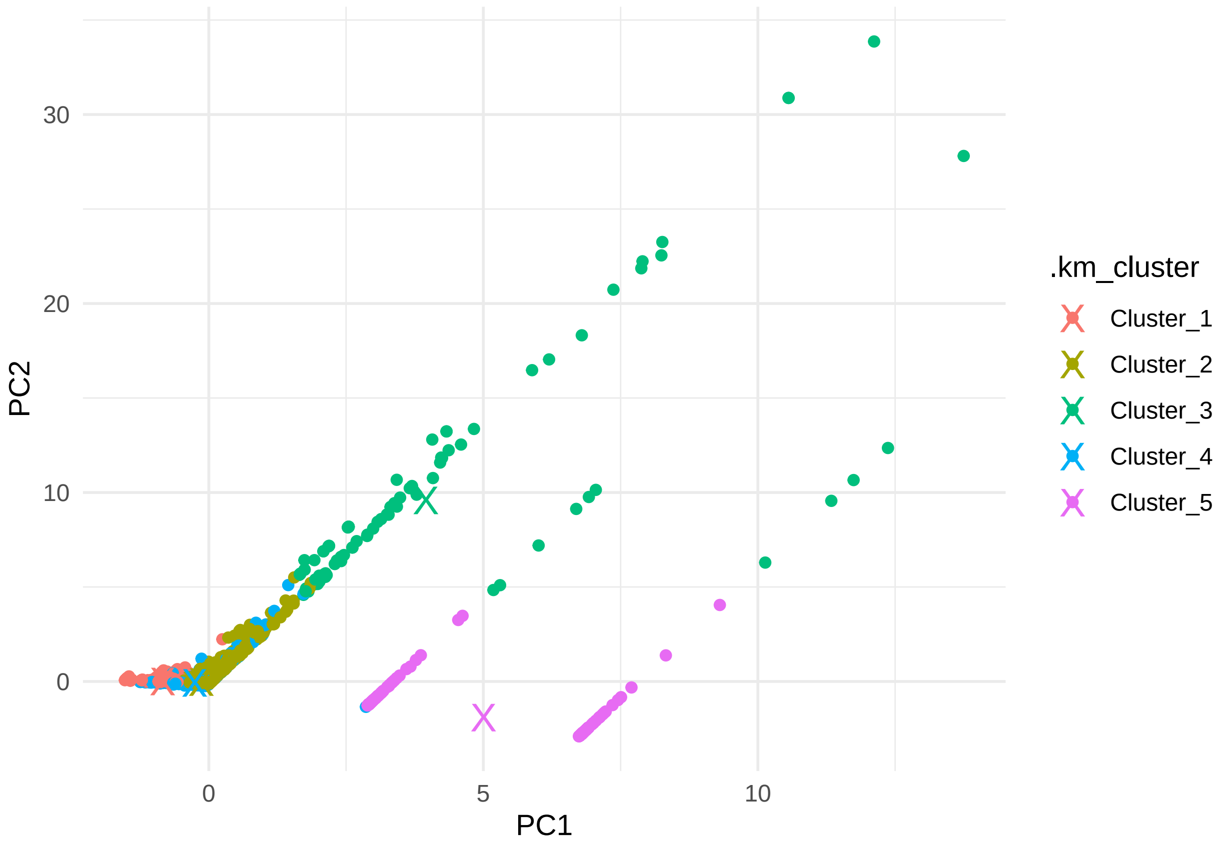
About 5 clusters seem optimal, as this is where we begin to see a steep fall

Visualizing High-Dimensional Data



Here, we cannot see the trends between players very well. So, we are going to use dimension reduction techniques to capture the first two Principle components of the data.

Pulling Principle Components



Aim 3: Explore the win or loss outcome two players playing the same game

Predictor Variables:

- Bet, PlayDate, Bet, Bonus, Frequent plalyer, Number of plays, Average bet, Average Cash out

Outcome Variable:

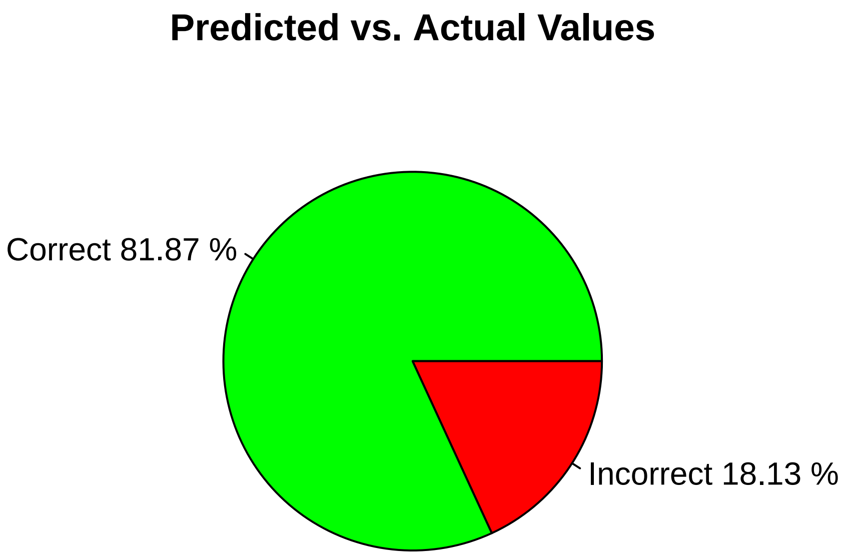
- Win (binary variable, where 0 = loss, 1 = win)

Modeling with K-Nearest-Neighbors (kNN)

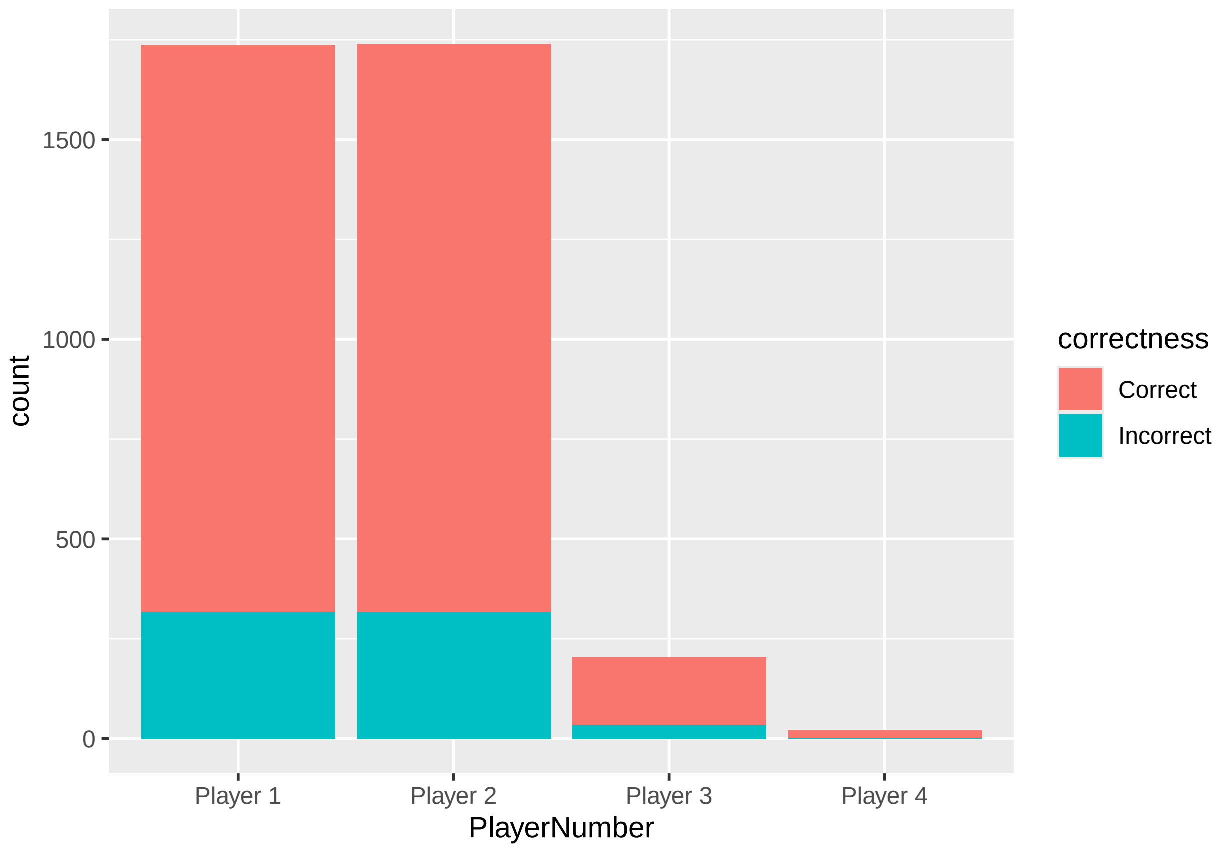
- Used to make predictions of a binary outcome (Win)

```
## # A tibble: 36 x 7
##    neighbors .metric  .estimator  mean     n std_err .config
##        <dbl> <chr>    <chr>      <dbl> <int>   <dbl> <chr>
##  1         3 accuracy binary     0.813     5 0.00512 Preprocessor1_Model01
##  2         3 kap      binary     0.619     5 0.0107  Preprocessor1_Model01
##  3         4 accuracy binary     0.813     5 0.00416 Preprocessor1_Model02
##  4         4 kap      binary     0.618     5 0.00878 Preprocessor1_Model02
##  5         5 accuracy binary     0.828     5 0.00381 Preprocessor1_Model03
##  6         5 kap      binary     0.650     5 0.00781 Preprocessor1_Model03
##  7         6 accuracy binary     0.828     5 0.00410 Preprocessor1_Model04
##  8         6 kap      binary     0.650     5 0.00852 Preprocessor1_Model04
##  9         7 accuracy binary     0.829     5 0.00333 Preprocessor1_Model05
## 10         7 kap      binary     0.652     5 0.00694 Preprocessor1_Model05
## # i 26 more rows

## # A tibble: 2 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.819
## 2 kap      binary         0.635
```

- The model was able to predict Win around 81.9% of the time, with these constraints.



We will now visualize if these predictions differ by Player number, as we stratified by this value.



- Illustrates the predicted correctness of wins for each Player Number
- It does not seem that there are any predicted differences between players 1 and 2 when they have the same set initial conditions determined by Game ID.

Conclusions

- These models have the potential to be applicable to other gambling domains, besides predicting exact profits. -With more information on players, trends between gamblers may become more apparent.
- It may be possible that certain qualities of gamblers can be used to predict their wins and losses, which would have major implications in real-life
- For example, knowing one's strengths and weaknesses as a more "frequent ßplayer," may alter how they gamble in future occurrences.