Shane Fuller

Professor Read

CS376B: Applied Data Science

April 10th, 2019

Data Quality Report

Overview Section

As the world becomes more digitalized, and interests move further towards expression over technology, this trend is seen especially in the growing field of Esports. First established in the early 2000's, the competitive interests in video games has seen a steady growth, however recently with the development of Twitch, eLeagues, and corporate sponsors, the viewership and participants have seen an exponential growth in the past few years.

With that said, the tools that the players can use in their respective fields is still a largely untapped market. Where you still massive databases in professional sports industries, which help create tools for spectators to view the game differently, and for the participants to learn from their faults, these data analysis tools are absent in the world of Esports.

This absence is the result of multiple issues. First, simply in the environment that players are competing. When spectating a tennis match, it is clear to see body movement, or shot placement percentages, and clear to recognize the small victories with each successive point. In the eSports industry, the competitive measures the lead to success are mostly mental, and do not rely as heavily on physical movements. What is physical, however, is a collection of very quick, sharp, calculated movements that is not physically viewed by the spectator. These are movements like mouse clicks, controller stick movements, and well-timed keyboard touches. When going about calculating these data, this presents a massive challenge for people interested in data analysis.

The next issue comes in defining a clear success in most of the games. While competing, it is true that one team will result in a win, and the other a loss. However, the thousands of interactions that lead up to the result do not have the same clear successor. Often, interactions between two players does not result in a clear victory, and results in a simple advantage state, or one player improving a position over another. Defining these interactions can be hard to define in most games, and therefore it is hard to get value out of the data collected.

Lastly, and perhaps the biggest issue, is a lack of understanding from the public of how to approach the field of eSports. This issue is one that is changing for the better, however society is still at a point where many cannot view eSports in the same way that traditional sport culture is viewed. Unfortunately, this concept extends to the developers of the games, who do not include tools for competitive players to learn, grow, and adapt with the current meta. This is a result of

the developers not intending to market their games to only competitive players, and these leaves the development of new tools to the communities themselves. To compare to traditional sports, this would be the equivalent of NBA players to create the tools to record shot percentage, rebounds, assists, and matchup trends rather than an association that would hire specialists in data collection and science to record and maintain this information. Therefore, the process is long, strenuous, and relies on the passion of the community rather than any nominal gain.

Summary Section

This led to the work down by Fizzi in the creation of Project Slippi, the main tool used for data collection in this analysis. Project Slippi is described as a Melee Data Framework, and essentially is a tool that offers every player the tools available to begin data analysis on the matches they play. This framework was implemented at a tournament in Philadelphia under the name, The Game Steals the Script. The software was installed on 25 consoles, and every game played on the setup over the course of the data exported a .slp file to the online database, which then became available to anyone who wished to view the file. This form of data collection, because it was implemented with code, came out to be extremely objective, as every file was consistent, and could not be tampered with by any individuals. Amongst the data, there were no missing columns, and only had NA values for when ratios were calculated with 0 in the denominator.

Therefore, the quality of the data is not measured in terms of how smoothly gathering the data goes or by any bias, but rather how smoothly the tournament ran, and the players that were in attendance. For example, if the tournament had multiple games where players were often not playing games competitively to completion, this has the ability to skew the data to show results that do not lead to a true depiction of killCount. Another measure of quality comes from the skill level of the players attending. If the tournament has a majority of players who often perform well, this will produce more consistent results with proper depiction of killCount than one that has a majority of players who often do not perform well. Lastly, another relevant measure of quality comes from the character representation throughout the tournament. Although there are 26 characters, only about 10 are often used competitively. However, it is important that these characters all have proper representation, as the features that used for the analysis are highly dependent on the character that is being played.

Having addressed how quality can be measured, this tournament had fairly good data quality. Of the 204 entries in the bracket, about 15 of those were in the top 100 best players for the year, which is good representation for a tournament. In addition, about 100 of the other players fit the tier below top level, and the rest fill the tiers of players who are not often successful in a tournament setting. In regard to the character diversity, this also appeared to be successful. The results show that the top 10 viable characters had representation of between 2000-2800 games each and was evenly distributed. The remaining characters ranged from 99 games to 1500 games, which is a fair representation for how often players are chosen in competitive play. The last measure of quality is in the number of games that were played to a

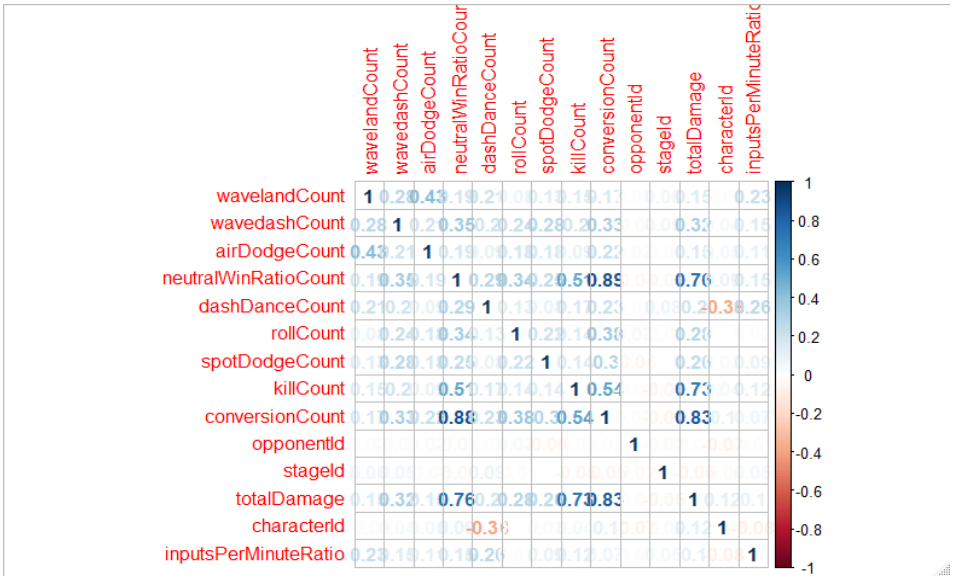
competitive level to its completion, however this is simply a variable that cannot be calculated by the data received from the .slp file. Rather this is a subjective measurement that can only be gathered by in person spectators, who can properly witness multiple the interactions between players at the tournament. Therefore, this study cannot speak on the quality of all games that were played, however it is safe to assume most were competitive, as a large majority of the games recorded were tournament sets, which are by nature treated with seriousness between both competitors.

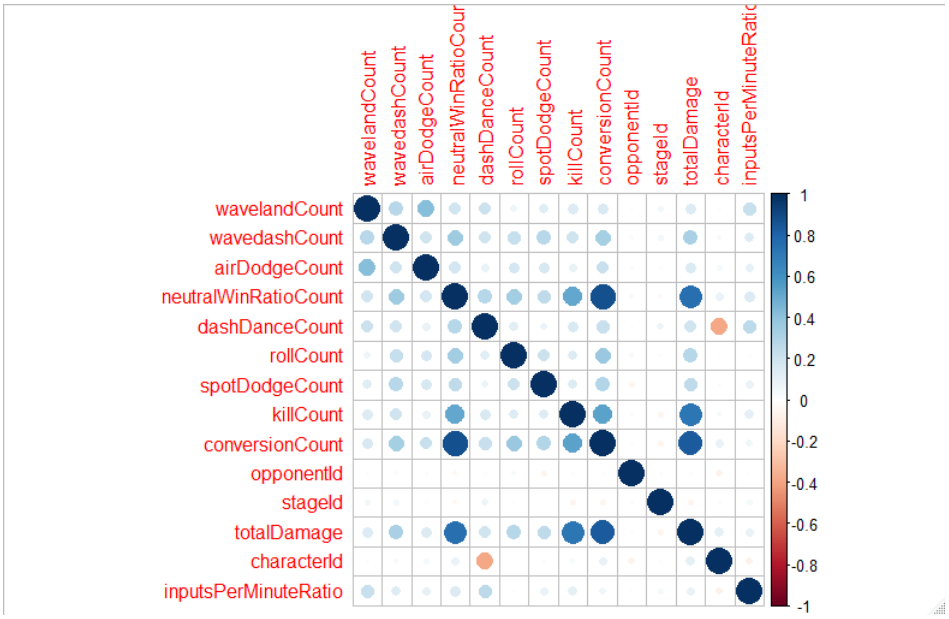
In conclusion, the quality of this data is unbiased, consistent, and does not content very many missing data values. By tournament standards, The Gang Steals the Script showed proper player and character representation, with a good mixture of top and mid-level talent, as well as a proper character distribution that would be expected in a competitive melee tournament.

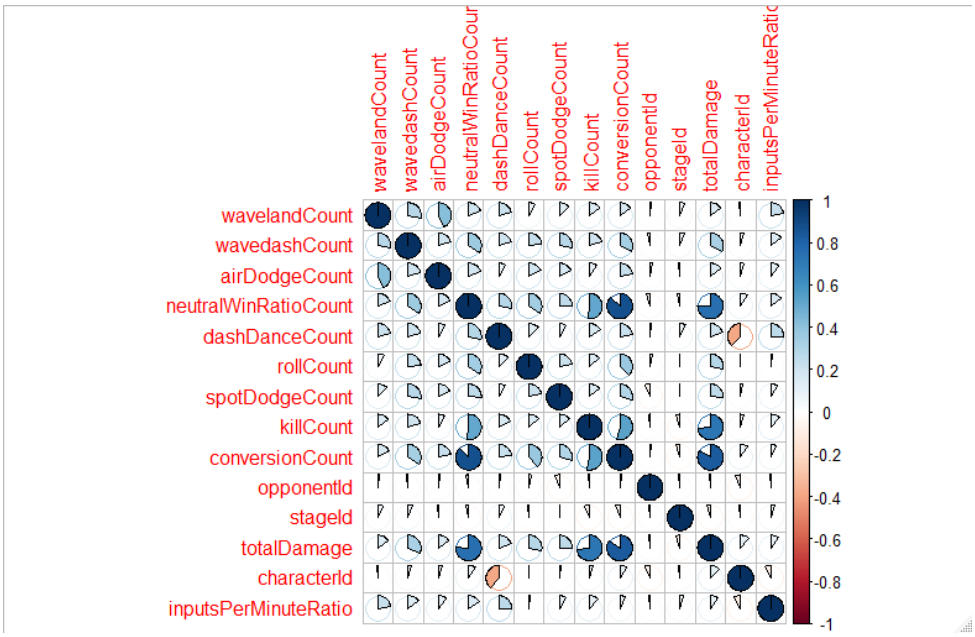
Details Section

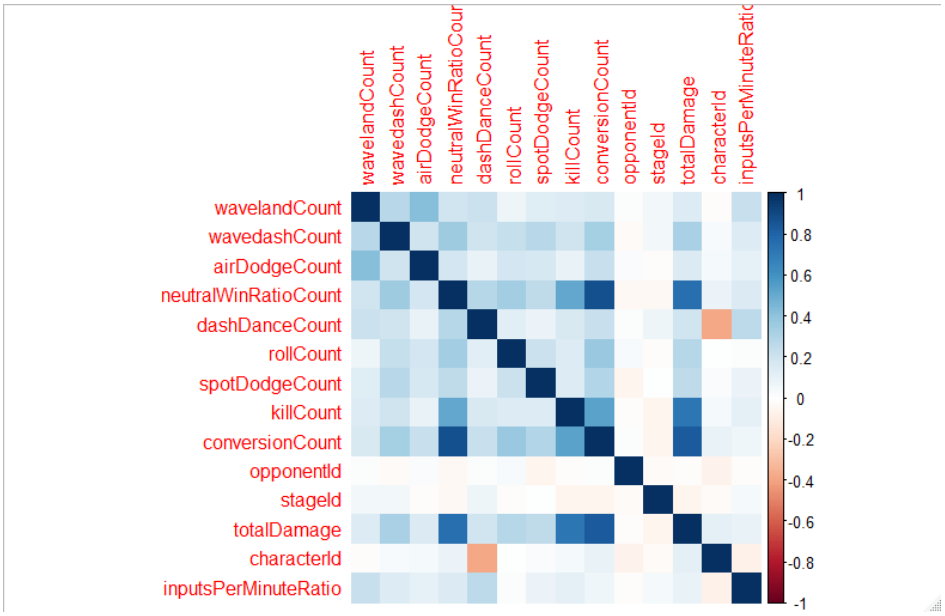
Moving forward, the following section shows the data exploration done in R, specifically reviewing the correlation in the data, as well as the summary statistics and for the features that were included in the final analysis. This section will include multiple plots, as well as a brief description of what each feature, and why it is important to the model.

Correlation Plots






wavelandCount
wavedashCount
airDodgeCount
neutralWinRatioCount
dashDanceCount
rollCount
spotDodgeCount
killCount
conversionCount
opponentId
stageId
totalDamage
characterId
inputsPerMinuteRatio
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1


wavelandCount
wavedashCount
airDodgeCount
neutralWinRatioCount
dashDanceCount
rollCount
spotDodgeCount
killCount
conversionCount
opponentId
stageId
totalDamage
characterId
inputsPerMinuteRatio
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

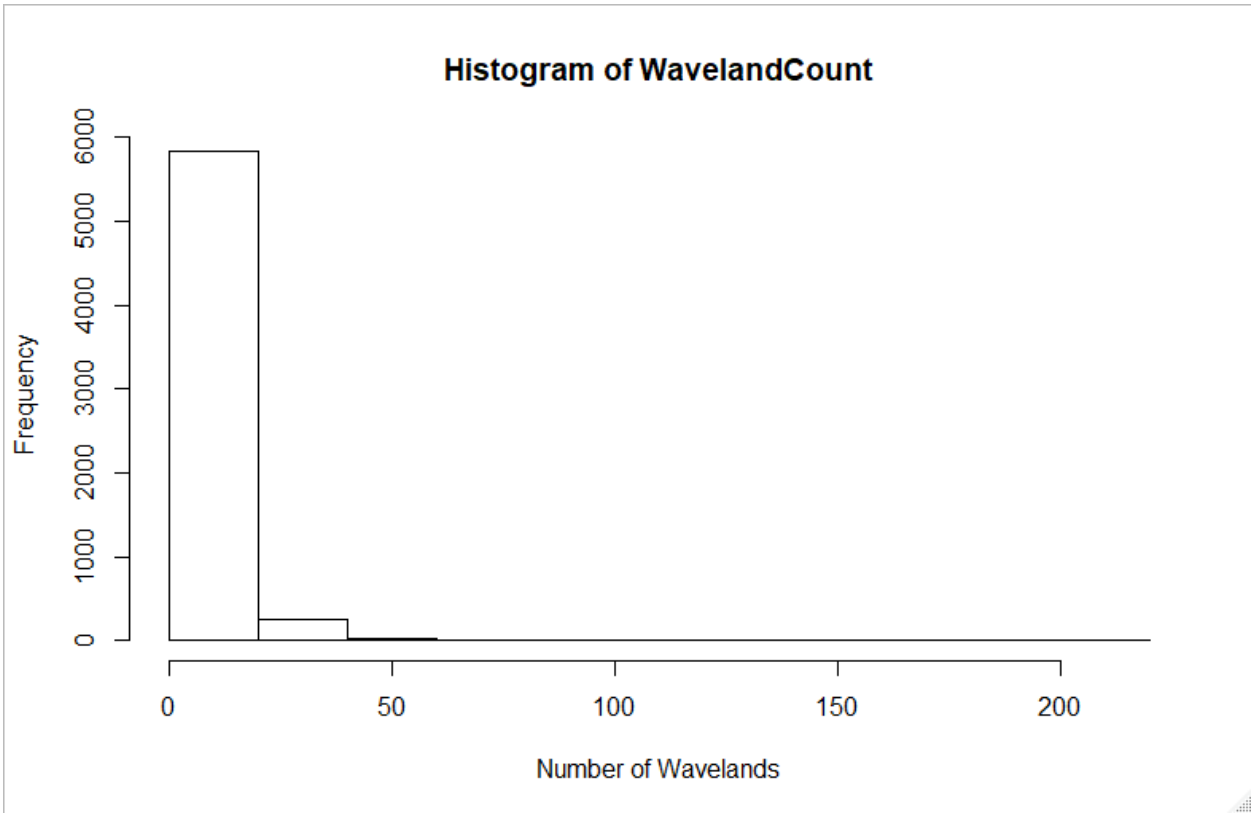
wavelandCount

Description: the number of wavelands performed by the player in a game

Summary Statistics:

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   2.000   5.000   7.074  10.000 218.000
```

Histogram: The histogram below shows a strong right skew.



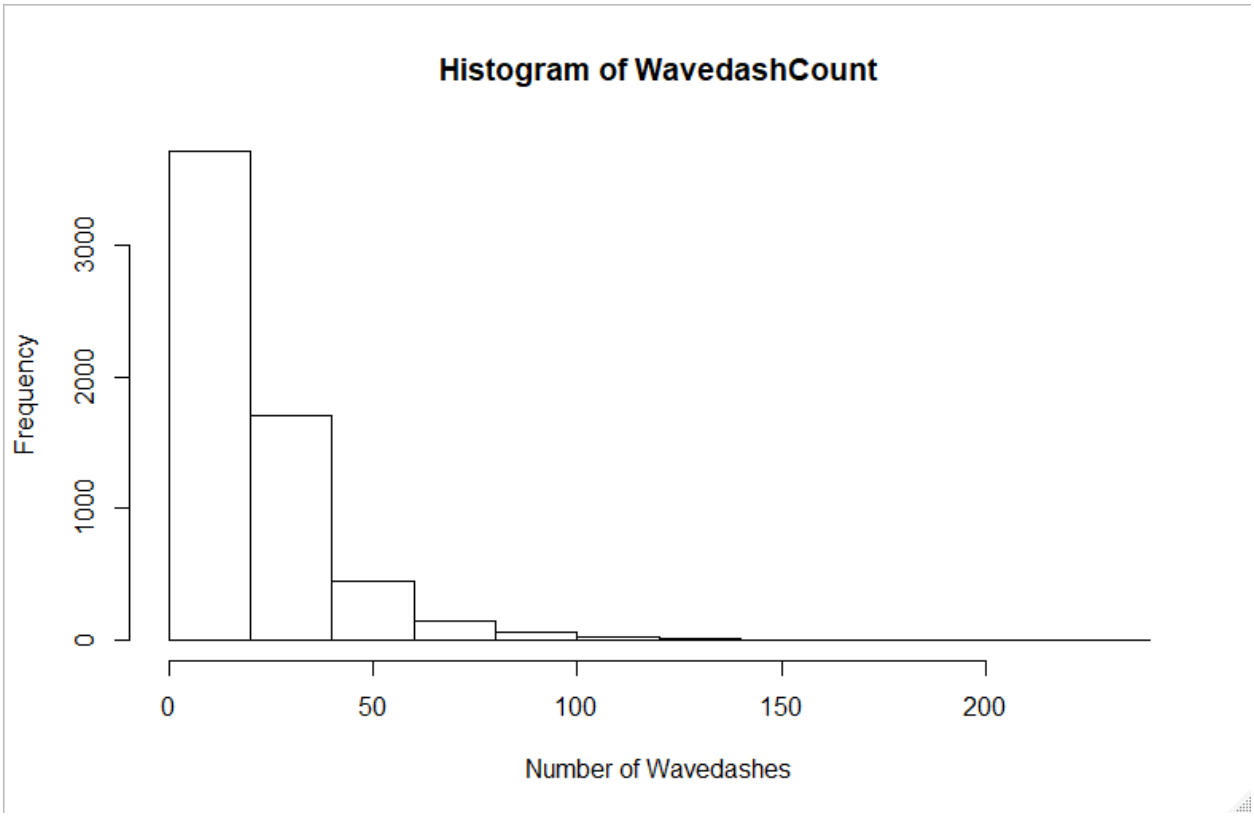
wavedashCount

Description: the number of wavedashes performed by the player in a game

Summary Statistics:

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    9.00   17.00   21.15   28.00  223.00
```

Histogram: The histogram below shows a strong right skew.



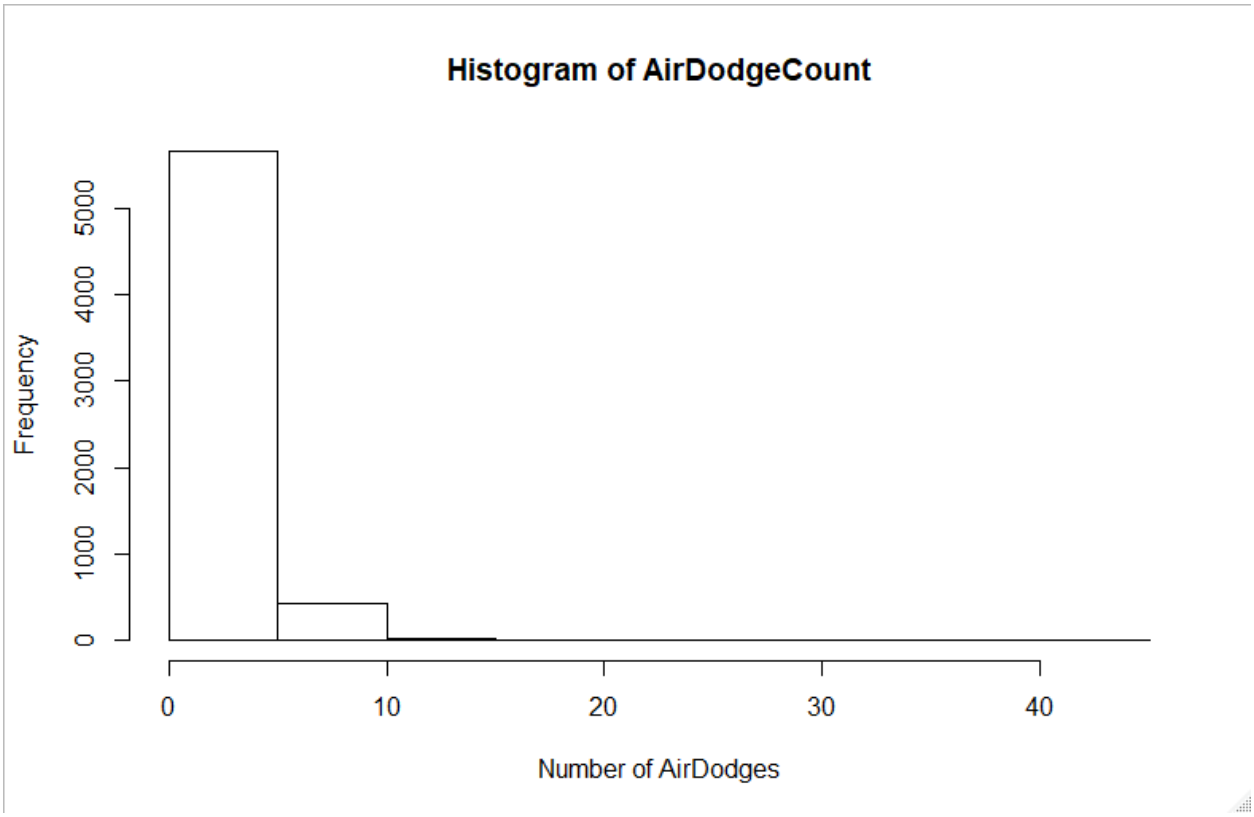
airDodgeCount

Description: the number of airdodges performed by the player in a game

Summary Statistics:

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   1.000   2.000   2.323   3.000  41.000
```

Histogram: The histogram below shows a strong right skew.



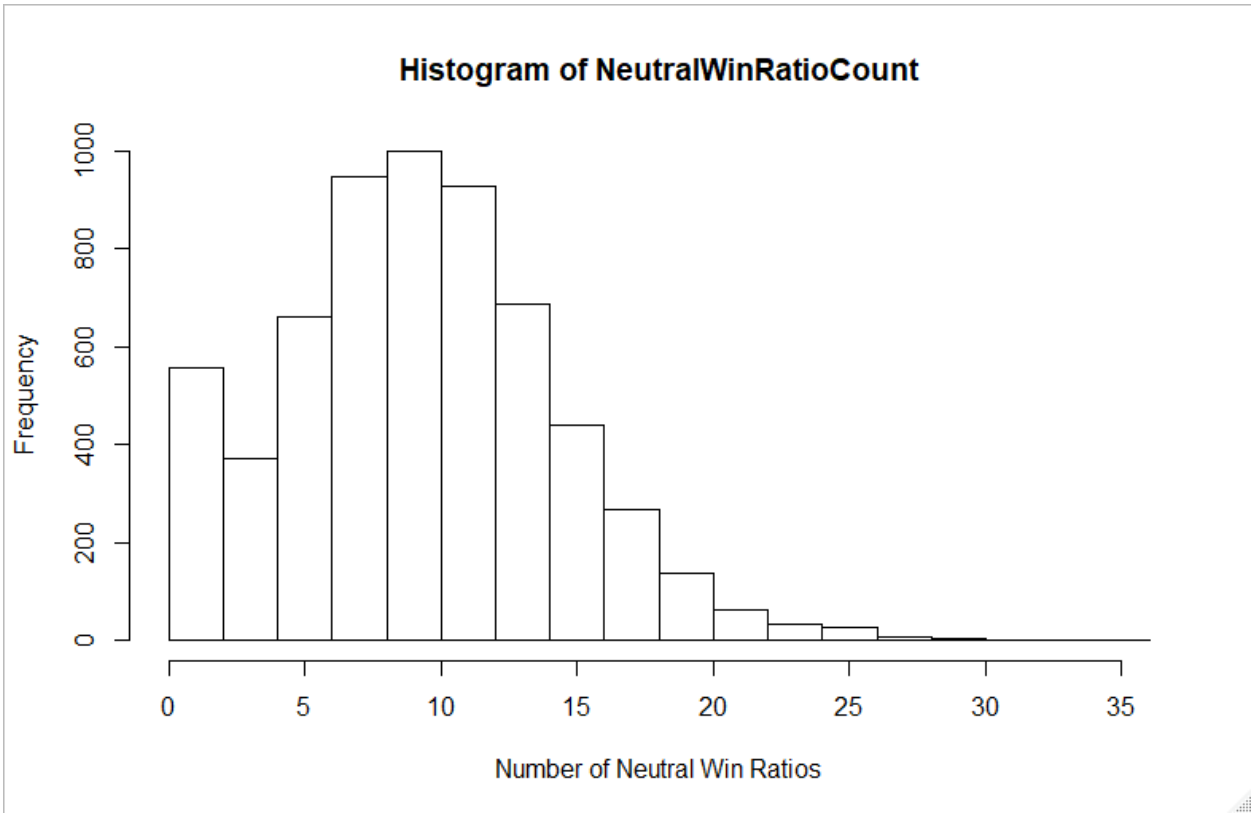
neutralWinRatioCount

Description: the number of times a player gains advantage state from a neutral setting

Summary Statistics:

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   6.000   9.000   9.662  13.000  36.000
```

Histogram: The histogram below shows a relatively normal distribution.



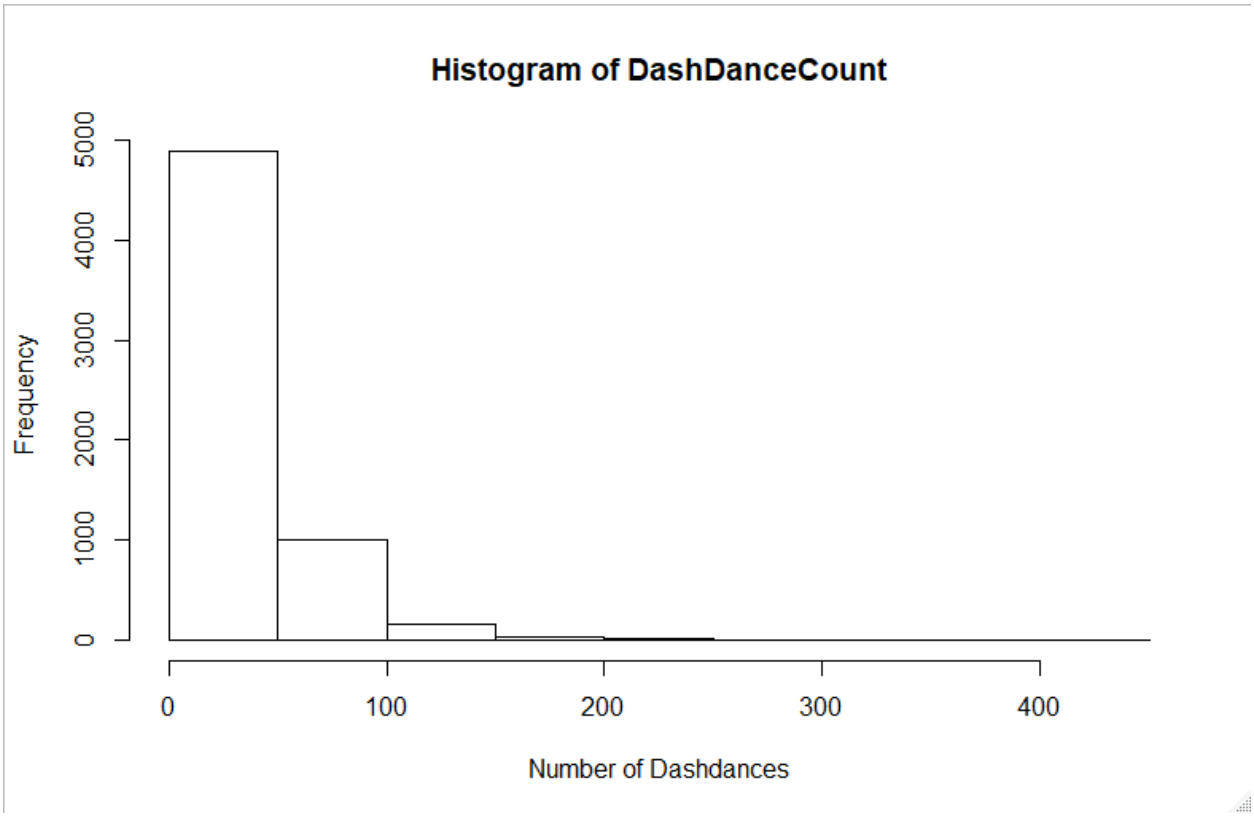
dashDanceCount

Description: the number of dashdances performed by the player in a game

Summary Statistics:

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    9.00   23.00   31.64   44.00  447.00
```

Histogram: The histogram below shows a strong right skew.



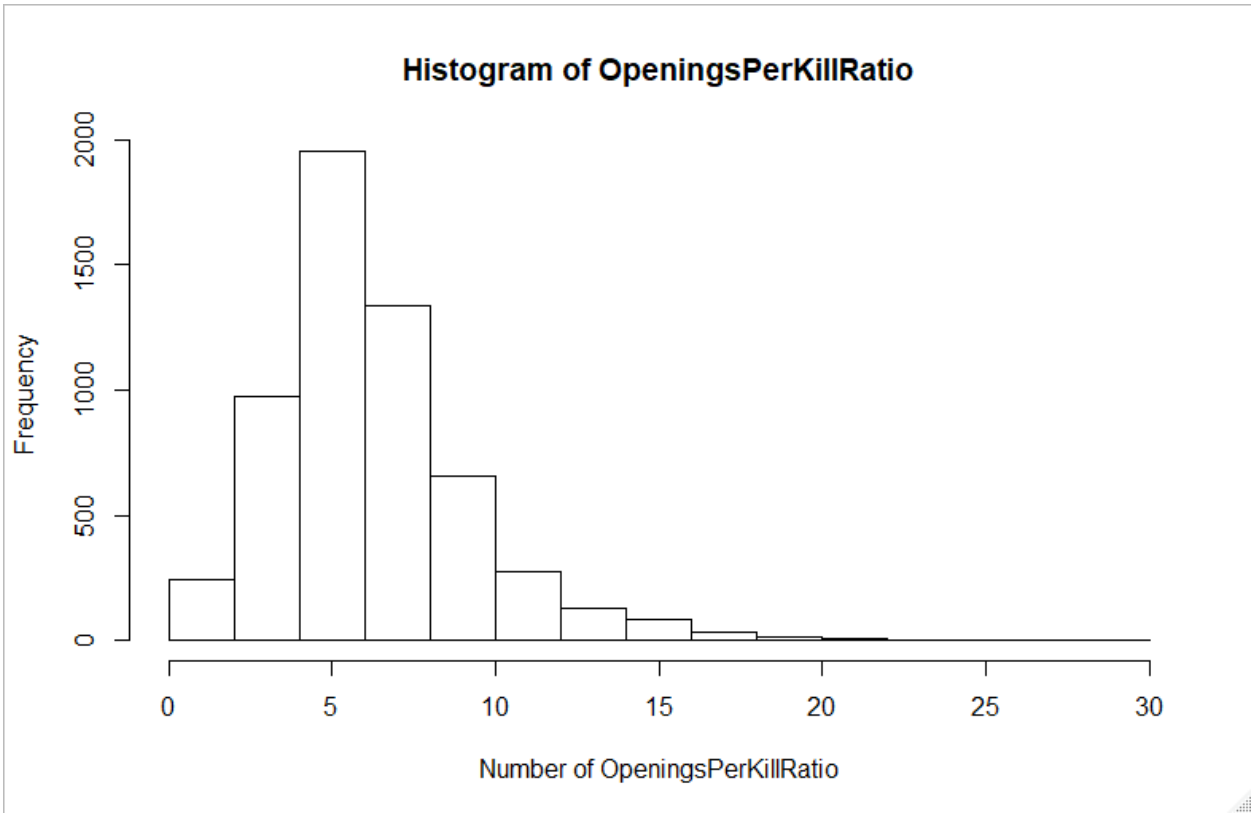
openingsPerKillRatio

Description: the average number of openings it took for the player to take a stock

Summary Statistics:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0.000	4.500	5.750	6.357	8.000	29.000	409

Histogram: The histogram below shows a relatively normal distribution.



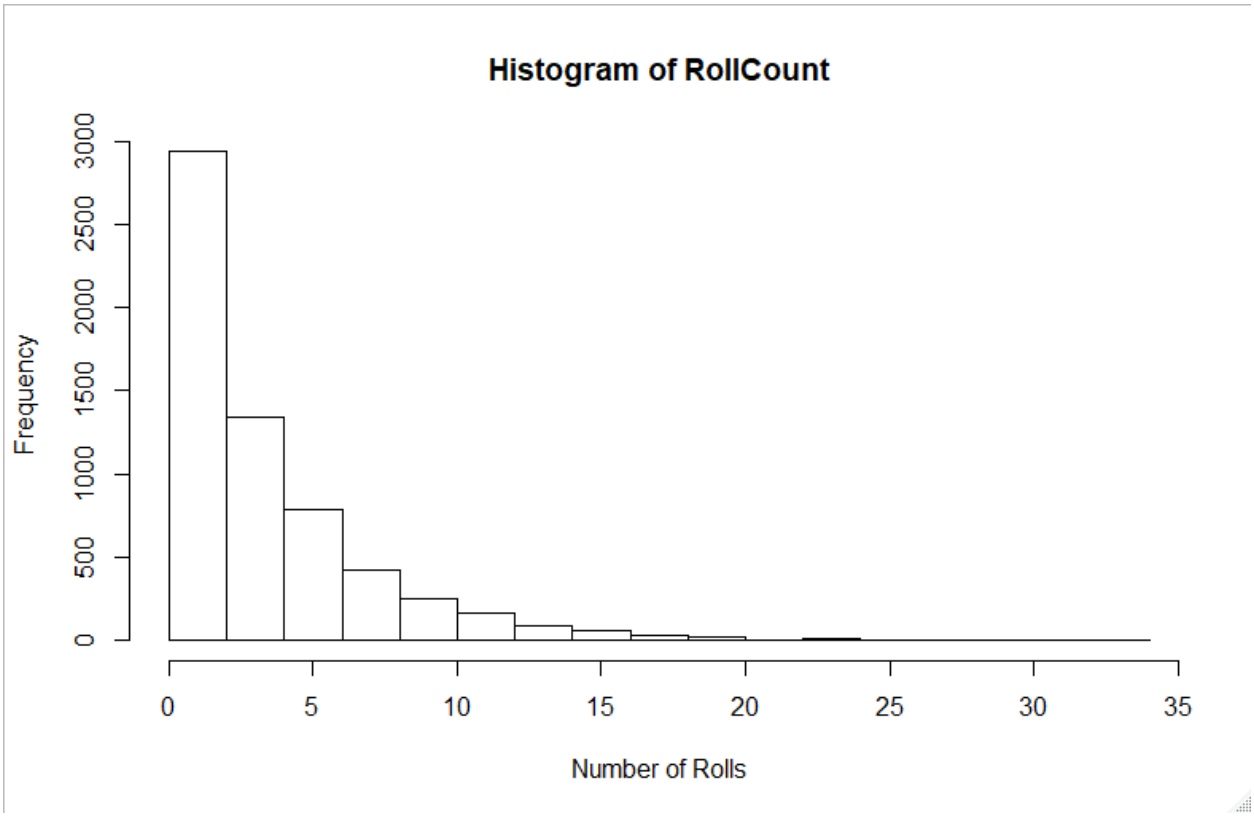
rollCount

Description: the number of rolls performed by the player in a game

Summary Statistics:

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   1.000   3.000   3.687   5.000  33.000
```

Histogram: The histogram below shows a strong right skew.



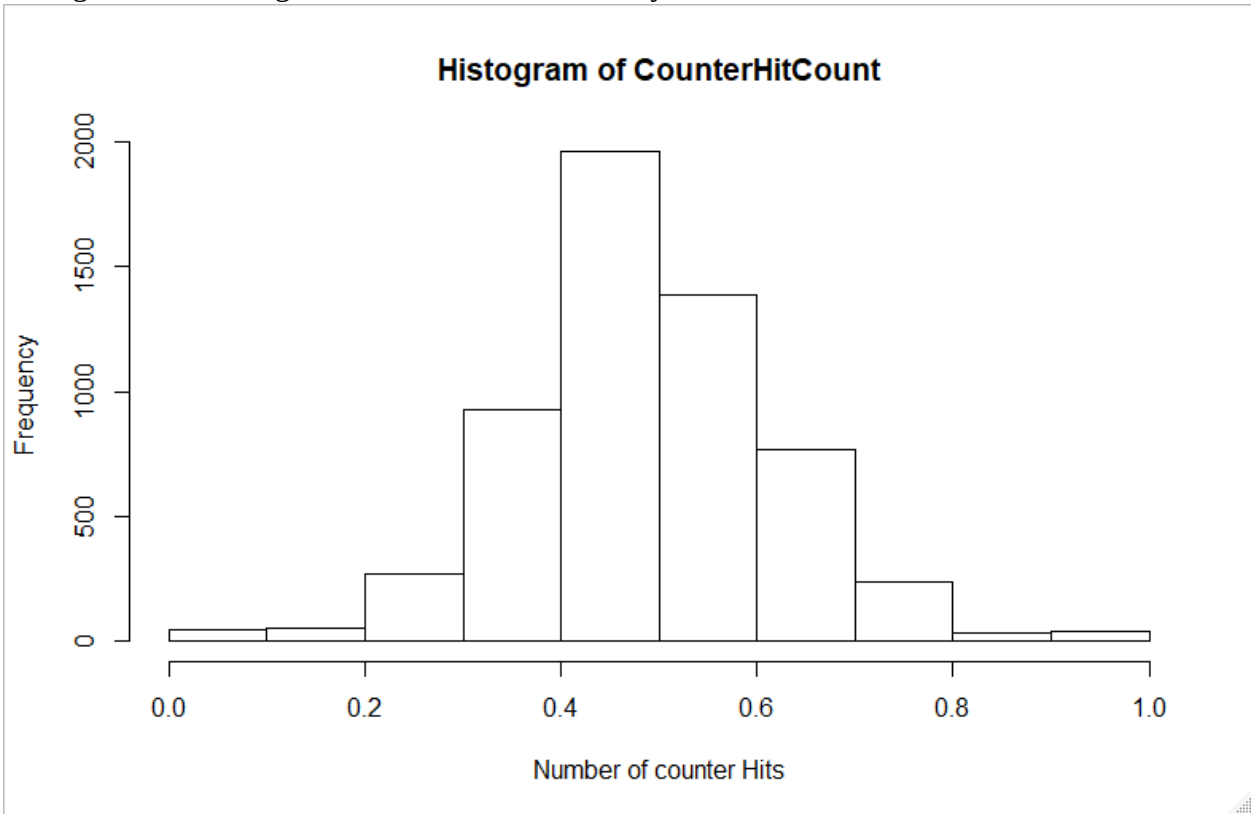
counterHitCount

Description: the number of hits a player landed from a disadvantage state

Summary Statistics:

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
 0.0000  0.4167  0.5000  0.5000  0.5833  1.0000     402
```

Histogram: The histogram below shows a relatively normal distribution.



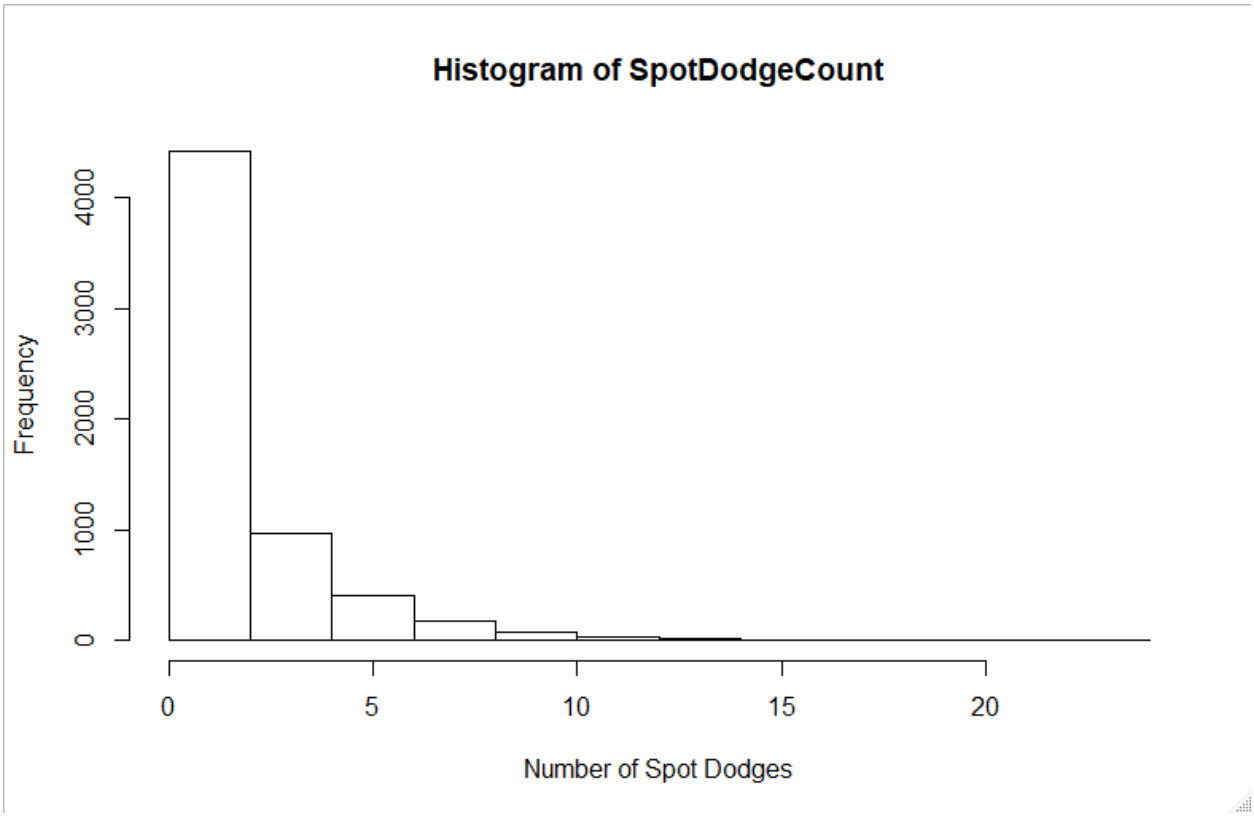
spotDodgeCount

Description: the number of spotdodges performed by the player in a game

Summary Statistics:

```
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   0.000   1.000   1.939   3.000  24.000
```

Histogram: The histogram below shows a strong right skew.



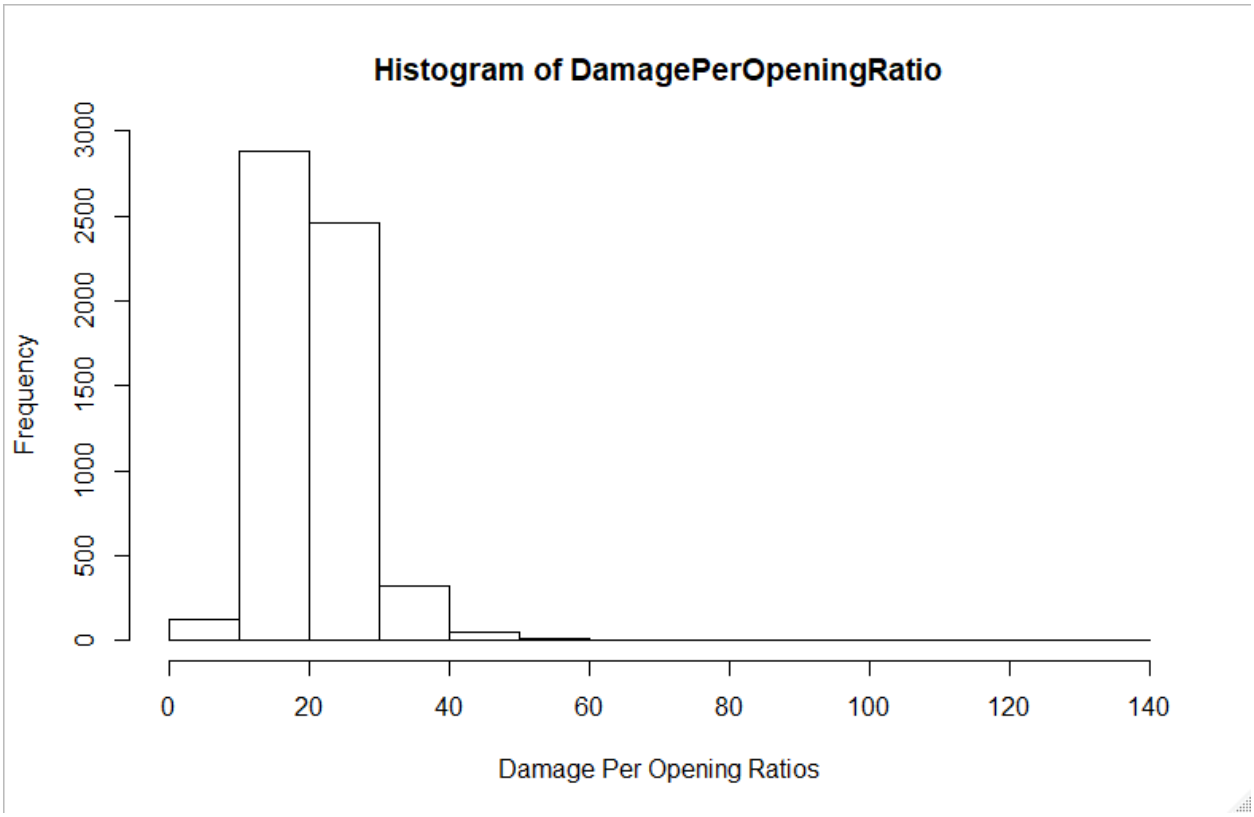
damagePerOpeningRatio

Description: the average damage done for each advantage state gained from neutral

Summary Statistics:

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
   2.00   16.60   19.82   20.68   23.73  137.29     270
```

Histogram: The histogram below shows a strong right skew.



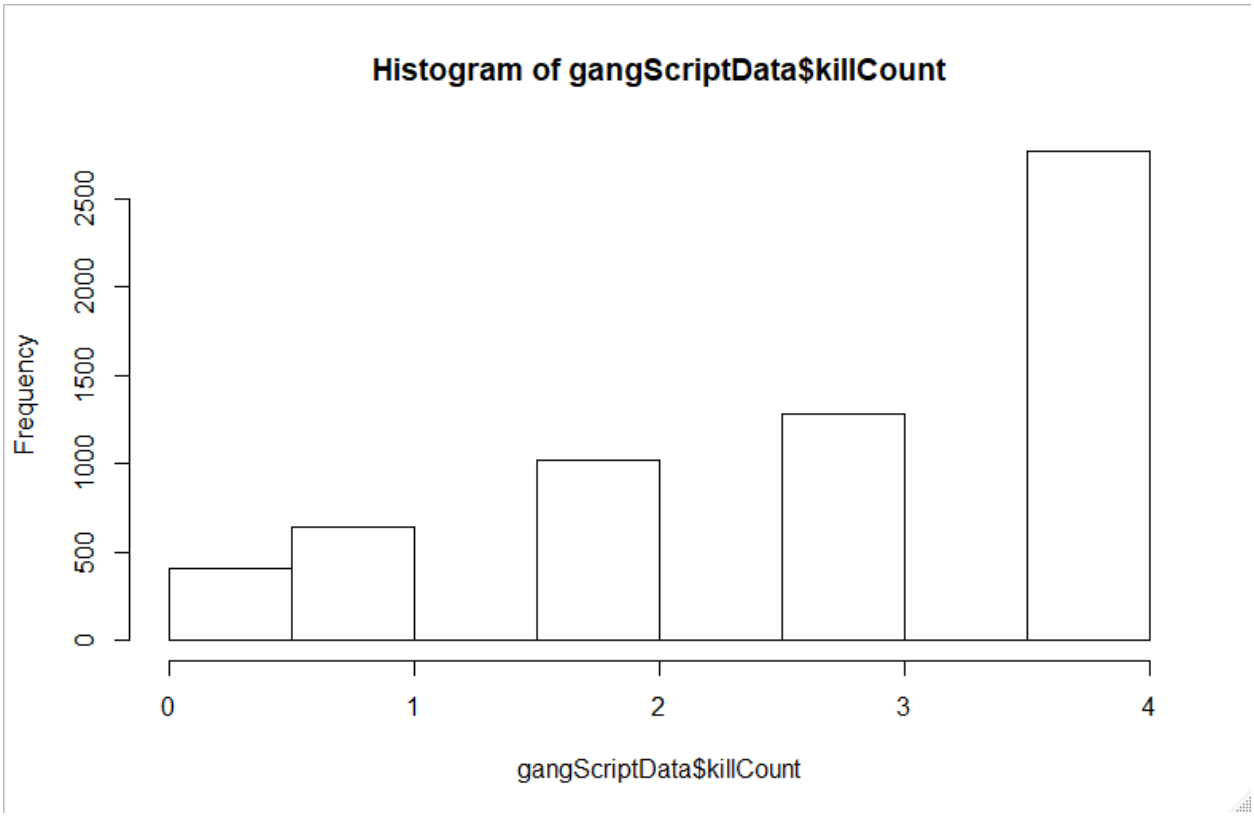
killCount

Description: the number of stocks taken per game (the target feature)

Summary Statistics:

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   2.000   3.000   2.874   4.000   4.000
```

Histogram: The histogram below shows a strong left skew.



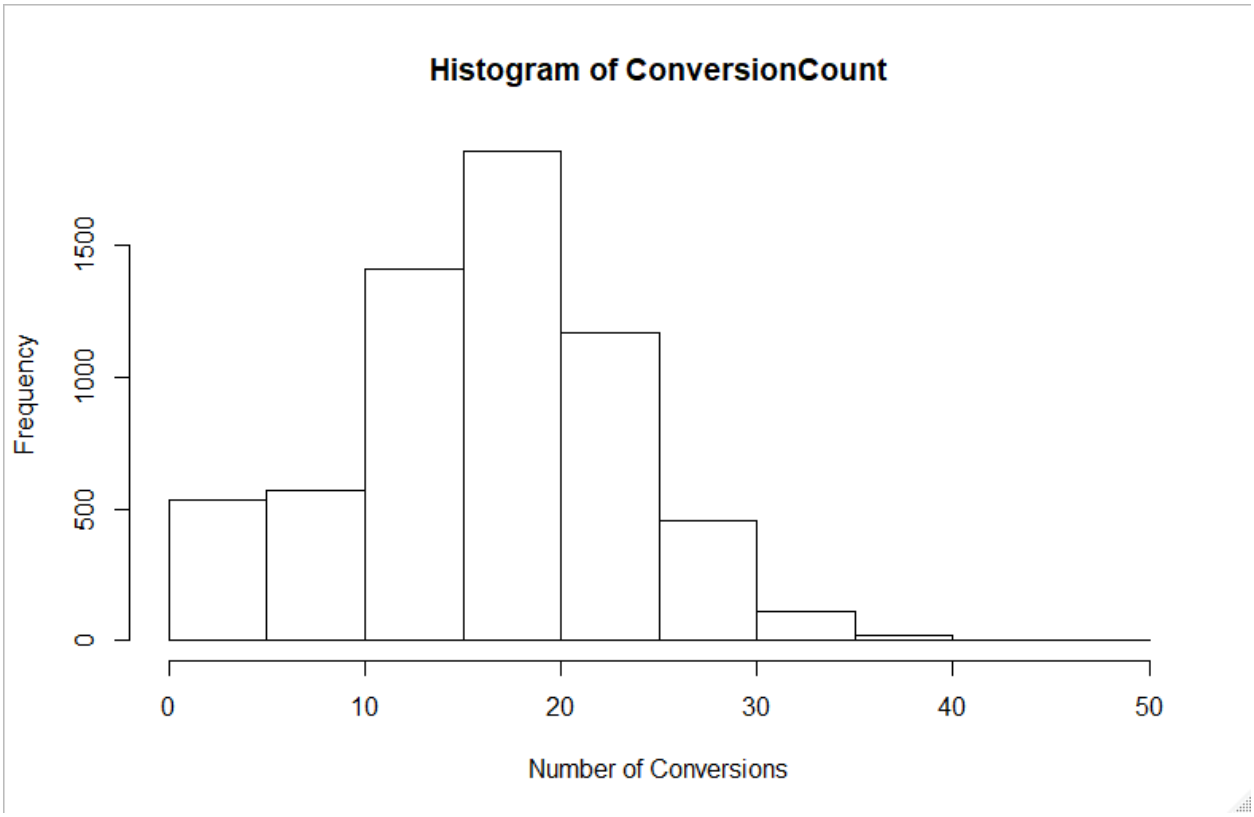
conversionCount

Description: the number of conversions performed by the player in a game

Summary Statistics:

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00   13.00   17.00   16.56   21.00   47.00
```

Histogram: The histogram below shows a relatively normal distribution.



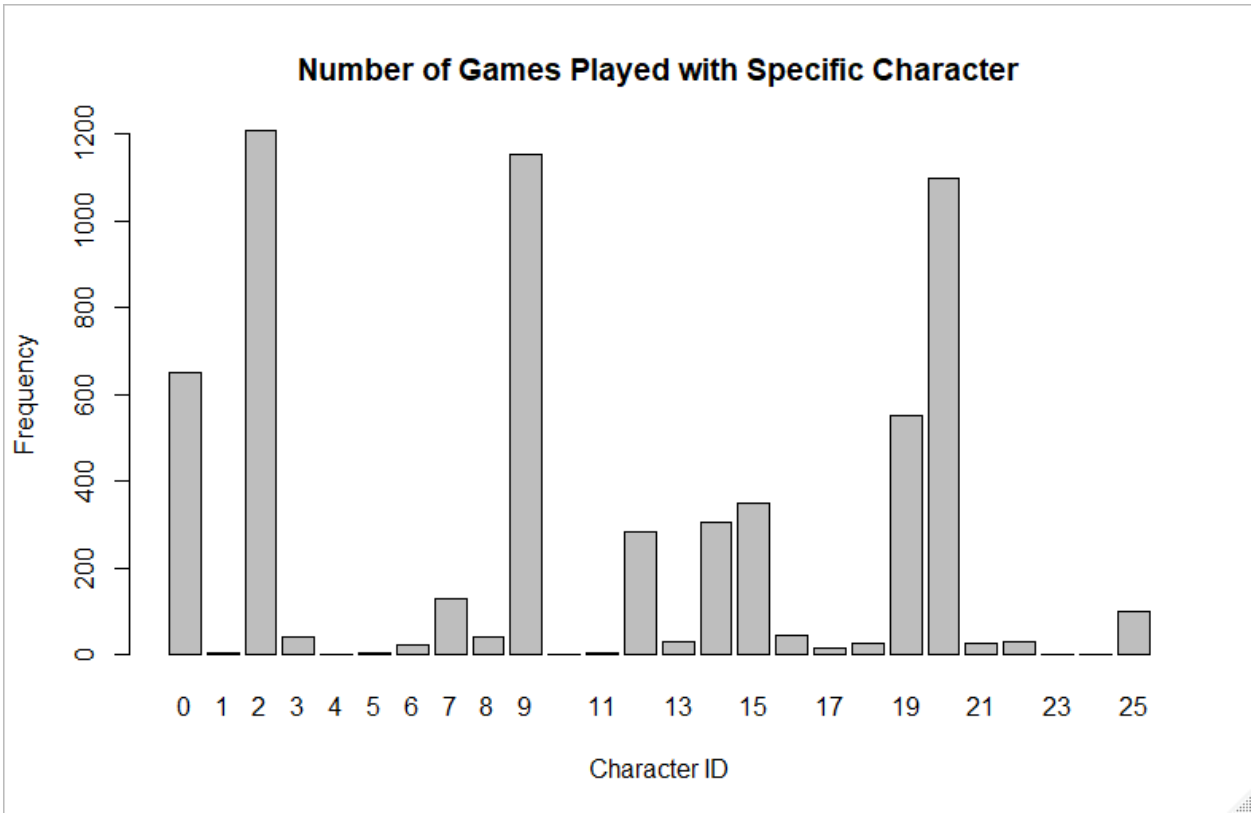
opponentId

Description: the character the player was playing against in a given game

Summary Statistics:

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    2.00    9.00   10.67   19.00   25.00
```

Barplot: The barplot shows the characters that were chosen most frequently.



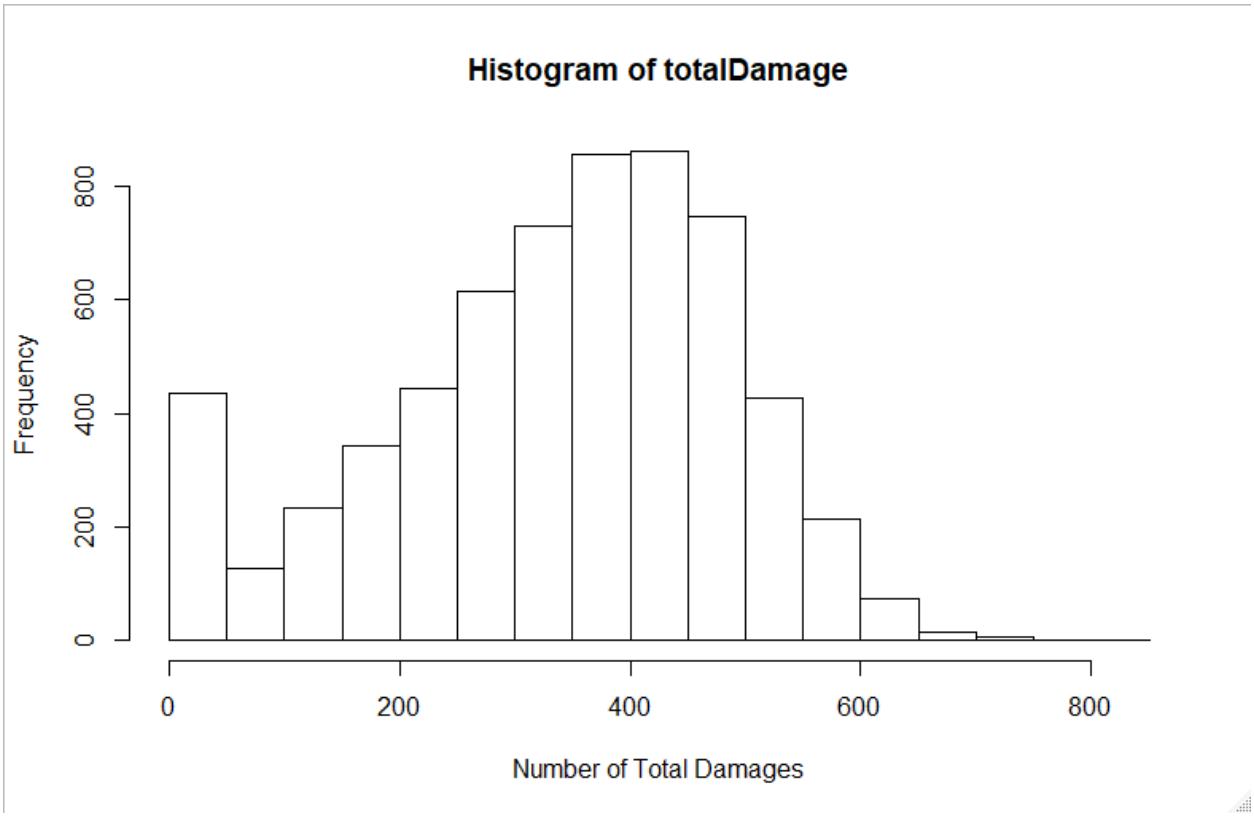
totalDamage

Description: the total percentage dealt to the opposing player over the course of a game

Summary Statistics:

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.0   244.3   357.8   335.6   447.1   820.7
```

Histogram: The histogram below shows a relatively normal distribution.



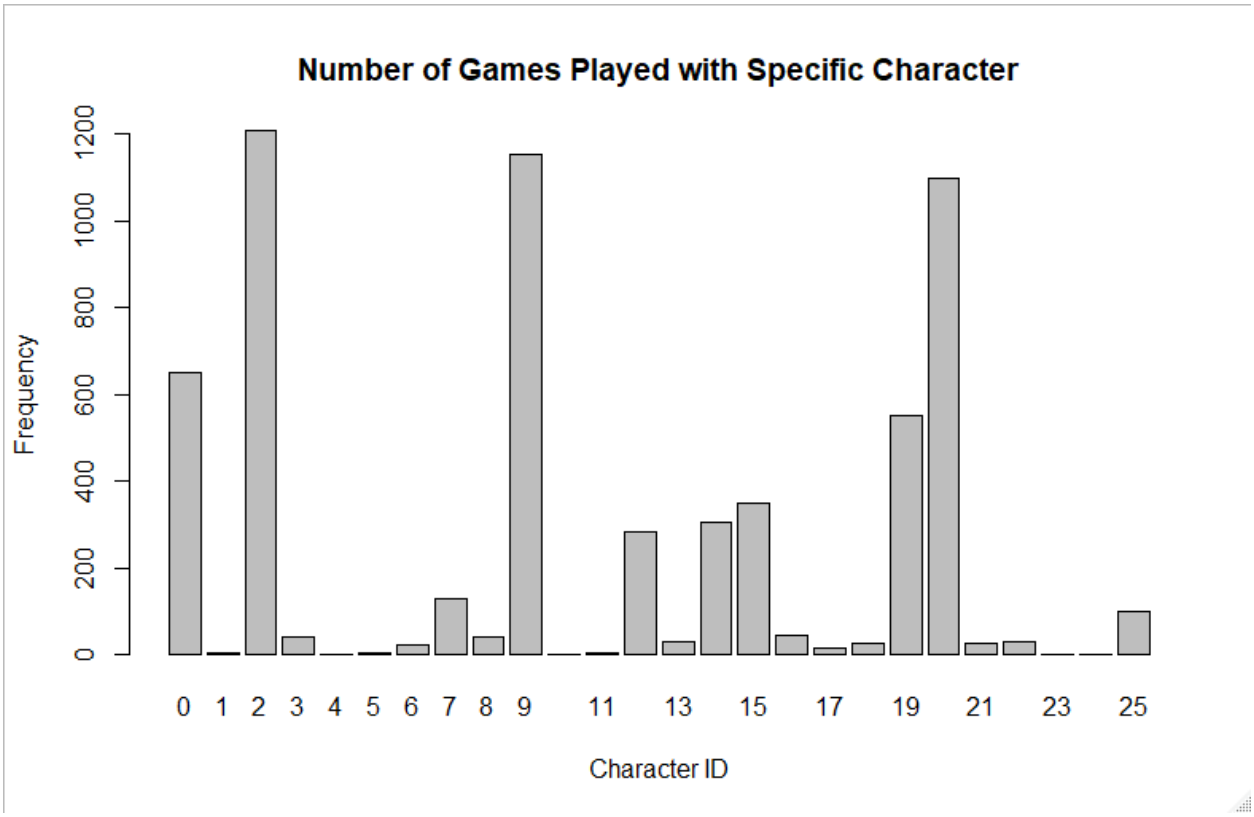
characterId

Description: the character the player was playing in a given game

Summary Statistics:

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    2.00    9.00   10.67   19.00   25.00
```

Barplot: The barplot shows the characters that were chosen most frequently.



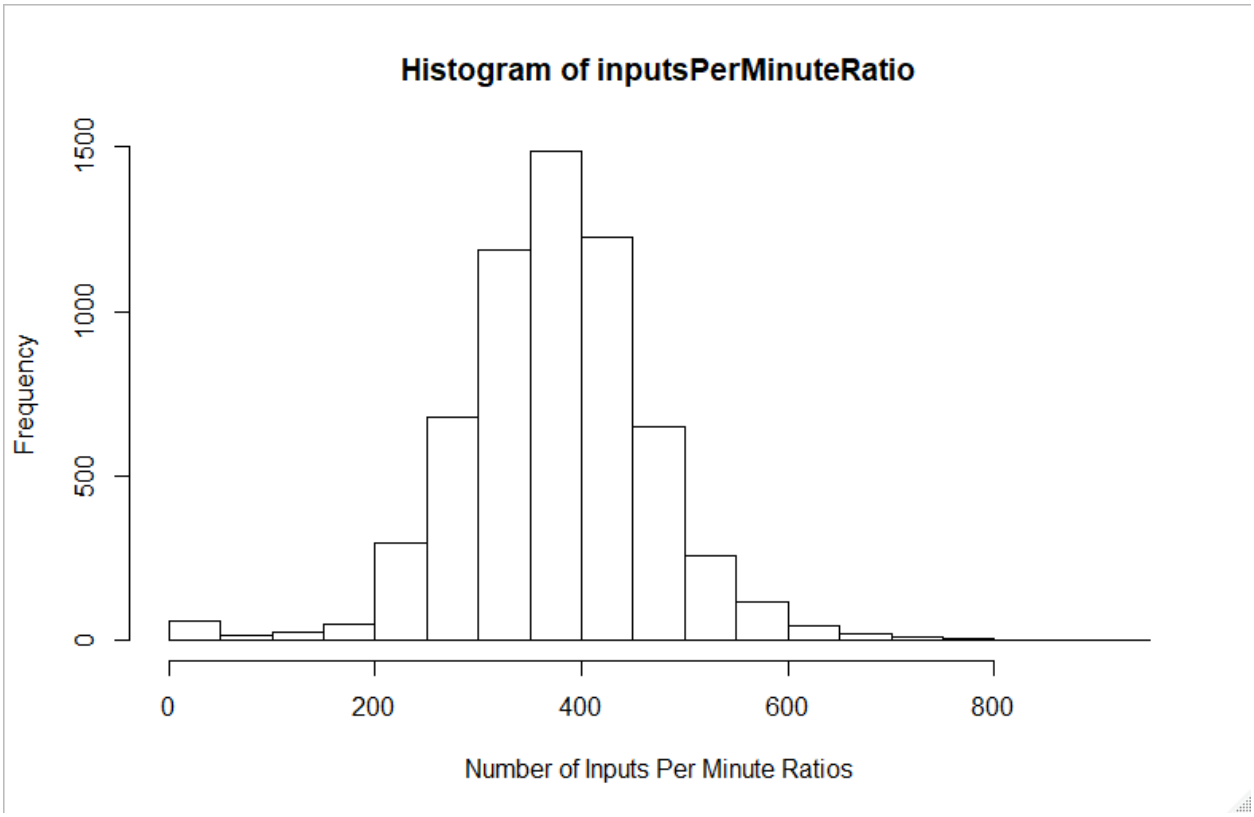
inputsPerMinuteRatio

Description: the average actions per minute performed by the player in a minute

Summary Statistics:

```
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.0   320.3   376.8   374.6   429.6   923.0
```

Histogram: The histogram below shows a relatively normal distribution.



Deleted Features

When going through all of the data that is supplied with a single .slp file, each JSON document contains a very large and detailed number of attributes from each game. Not only is the framework of Project Slippi meant to aid in data analysis, but it is also a platform that is meant to allow for players to review entire games, as well as having the ability to use the platform to stream games without proper recording equipment. This is relevant because every .slp file holds the data for not only the data features that are useful for this analysis, but also a memory of every input from the game for both players. Therefore, the initial retrieval of the data included plenty of feature deletion when first parsed into the CSV file. Entire documents were

excluded from the analysis, including all information on specific inputs, combo data, conversion specific data, and meta data involving time of each occurrence of the desired variables. In a more detailed analysis, the use of combo data would be helpful, however this data would have been too complicated to include in this iteration of the data project.

From this step, this left a table including 3066 games, each flat file holding the data involved for both players. After the data manipulation step, the information for both players was separated, and this left 6132 rows of game information for individual players. At this point, there 22 features for each player, and this is where the feature deletion process really began. To start, there were a few feature selection strategies to consider, with the goal to identify what features were no longer necessary. First, the process began with running multiple correlation plots of the data. The features damagePerOpeningRatio, openingsPerKillRatio, and counterHitRatio had to be removed, as they all included "NA" values that were not suitable for the correlation model to be run. The correlation plots did not show any two variables that were too strongly related. Among the stronger relationships, there was conversionCount and neutralWinRatioCount, totalDamage and neutralWinRatioCount, totalDamage and conversionCount, and totalDamage and the target, killCount. However, although these variables were decently correlated, each data point is important to the analysis, so they were all kept in for this study. Next, a random forest model was run to see the attribute importance. The most important far and away was openingsPerKillRatio, and the least important tended to show stageId, opponentId, and spotDodgeCount to be among the least important in distinguishing killCount. However, for the final model, these variables were still considered. Continuing forward, feature selection next focused on entropy values to remove features. However, all features were somewhere in between an entropy of 8.0-8.7, therefore this process also did not remove any features. After this, prediction tree models, forward greedy with a tree evaluation, and forward greedy with a linear evaluation we're all performed. However, these results were indecisive, and did not lead to the removal of any features.

This concludes the feature selection process, and the last step towards going through the data is simply removing features that intuitively seem unhelpful as a competitive player towards determining a killCount. From here an additional 6 features were removed to create the final ABT. First, inputsPerMinuteTotal was removed, as the values were confusing and misleading, with especially unclear units. Next, id was removed, as the values were not helpful, and they were causing issues with the model creation. Last, the remaining four variables that were deleted were total counts, which proved to be unnecessary due to the existence of their ratios, which is a more helpful unit of measure for this type of information. These four variables were inputCount, openingPerKillCount, inputesPerMinuteCount, and beneficalTradeRatioCount. With these features removed, this resulted in the final ABT to be used for model creation.