PUBG Data Royale

CSCI 4502 Final Project

Matthew Davenport
Federico Aragon
Thomas Alder
Haotian Dong
University of Colorado
Boulder, Colorado, USA

ABSTRACT

With an average concurrent player count of under that 400,000 players PlayerUnknown's Battlegrounds has become one of the leading figures in the popular "battle royale" genre of video games. Naturally, this games player base is competitive and the second they drop out of the plane into the map there is only one thing on their mind, the elusive "Chicken Dinner" or first place whether it be solo, as a duo, or with a team. In this project, we looked to use data to understand the mechanics behind getting first place and what strategies would yield a better chance of winning. In short, how do we use data to try and win? Throughout working with our dataset our findings indicate results that may go against intuition. Intuitively, one may think that the most kills or the most damage would mean that you're the "king of the hill" and that would mean an automatic first place. Though, our findings show that this is not necessarily true. With classification and aggregations of OLAP cubes and a data warehouse we found that players in first place didn't have the highest kills/assists/damage in a game as first place players only had an average of three kills, one assist, and 396 damage. In addition, these players also had kills that were a lot more close combat zones in comparison to kills over a greater distance illustrating an importance in winning battles in close quarters. In essence, these show that a good PUBG strategy is about *survival* rather than *attrition*.

PROBLEM STATEMENT/MOTIVATION

In this project we are looking to analyze data from the results of a player's matches they played in a Battle Royale video game named PlayerUnknown's Battlegrounds. Using this dataset, we hope to answer questions regarding the game's mechanisms through an analysis of of player damage and deaths and how this statistic could be improved. Mainly, we are looking for the the best method and details which will lead to a win. Through analyzing the data of over 720,000 matches, we are going to try to understand the ways in which winners of these matches were able to obtain the fabled "Winner Winner Chicken Dinner". Find the best strategy to use when playing the game in terms of determining the most dangerous places to avoid, places which may be beneficial, and the general strategy of players that tend to win more will help us define a general strategy to try and gains the most wins.

RELATED WORK

1 **Final Circle Heatmap**

Kevin Pei. 2018. Final Circle Heatmap: PUBG Match Deaths and Statistics. https://www.kaggle.com/skihikingkevin/final-circle-heatmap

This kernel explored trends of “final circles” across a number of games played in PlayerUnknown Battlegrounds. It visualized their closing/finishing patterns and reflected on the movement expectancies of circles through analysis of the data. Throughout a match in PUBG players are restricted to a circular playing area which will begin to decrease in size at given time intervals throughout the match. The circle is pivotal to the functionality of the game since it brings surviving players closer together when their numbers grow sparse during the late game. Using python and jupyter notebooks, author Kevin Pei uses the deaths of players who placed first and second and who did not die outside of the circle to estimate the location of each final circle. He then plots the results on a heatmap which shows the distribution of final circle locations, demonstrating areas of congestion in which the final circle ended up in a concentrated area. The maps visualized show that the circle never tends to touch the map limits, and the central areas of the map have coverage of the finishing circles much more frequently than the edges of the map.

2 **Survival Analysis Using Kaplan-Meier Estimator**

Robert Kwiatowski. 2018. PUBG - Survival Analysis (Kaplan-Meier): PUBG Match Deaths and Statistics. https://www.kaggle.com/datark1/pubg-survival-analysis-kaplan-meier

This previous work shows the damage and die rate data for each players outside the playzone before PC 1.0 Patch #15 of PlayerUnknown Battleground’s to visualize the survival situations. The data set also has the data of the party situation including SOLO, DUO, and SQUAD, the use of vehicle for each players, and the numbers of players for each game. He plots the results for different groups with those conditions to illustrate the survival rate for players. Then, Kwiatowski uses a Kaplan-Meier estimator to use non-parametric analysis using the number of events that occur during a time frame and the survived units from the previous time frame. Each time frame represents the amount of time the circle is a stagnant playzone. Using this, he creates three plots which shows the survival probability based on the team members a player has which are solo (no team members), duo (one team member), and a squad (2-3 team members). Here it is seen that the highest chance of survival is for squads. The same process is done for two different scenarios in which a player uses a vehicle versus when they do not which yields results that show using a vehicle has a greater probability of survival. A final stratification is done based on the number of players in a match between small (< 33 players), medium (33-66 players), and large (> 66 players). Here the results show that the smaller number of players creates the larger survival chances.

3 **Data Visualizations**

Peter Ott. 2018. PUBG - You're not as bad as you think: PUBG Match Deaths and Statistics. Various data visualizations - https://www.kaggle.com/ottpeterr/pubg-you-re-not-as-bad-as-you-think

In this previous work, Peter Ott creates a variation of data visualizations mainly histograms which plots different relationships which can be found in the data sets. First, he plots the

relationship between the rank of team placement and player kills before splitting the dataset into quartiles based on the final ranking of teams following a match. Here he finds that making it into the top five isn't a task that requires a lot of kills which many might think with a "King of the Hill" style of game. Through his next visualizations, the same theme occurs. He finds that a lot of kills is necessary to be a winner otherwise known as being part of the "Chicken Dinner Club" with the majority of winners falling into the three to six kill range.

DATA SET

For this project we used the dataset "PUBG Match Deaths and Statistics", which we originally acquired from Kaggle.com[1]. The dataset was compiled by user skihikingkevin on Kaggle, also known as Kevin Pei, and the individual data for the dataset was extracted from pubg.op.gg, a game data tracking site. The data set includes the results from over 720,000 matches, and is comprised of a file for aggregate data and a file for player-death data. The player-death file contained stats for individual deaths and named key details about those deaths, like x_position and y_position at time of death for the killer and for the victim. The aggregate data file contained stats which help extend understanding of the players involved, providing stats for team dynamics, travel distance, damage dealt, and other. The dataset is saved onto all of our machines so that we can all familiarize ourselves with its content. The files in the dataset each contain four csv files. Four csv files cover all of the deaths that occur during the matches, and four csv files cover the aggregate data for each player during the matches, with each file being about 2GB in size. Each csv file has over 13,000,000 rows of data, so working with the whole dataset on tetious computational processes can be quite slow. For this reason smaller sample portions of the data set may be used in testing code. Since the data is organized in no particular order we are going to use the first one or two csv's as a training set for our classifier, which guesses whether a match was won or not. Another of the unused csv files will be used as a testing set for our classification model.

[1] https://www.kaggle.com/skihikingkevin/pubg-match-deaths/home

MAIN TECHNIQUES APPLIED

1 Data Preprocessing

In order to preprocess the data, we chose to insert the csv files into a MySQL database which provided to be preliminary database allow for faster queries than a Panda's Dataframe. The main preprocessing came through organization of values and columns into dimension tables that would make up our data warehouse. With this preprocessing, writing queries for accessing the data allow for the data warehouse process to be simpler than searching the Dataframe with the csvs loaded into it. That way we didn't need to load it every time we ran the script.

2 Data Warehouse/Cube

For our data warehouse we used a MySQL backend which had a star schema. The csvs were broken up into six separate dimensions, match_date_dimension, match_id_dimension, time_in_game_dimension, match_mode_dimension, killer_dimension, victim_dimension which all had foreign key relationships to the fact table named death_fact. For the data warehouse, we decided to treat each transaction as a death. Each death had a

key from each dimension which held the respective information. In order to organize the csvs first each we created a MySQL database which allowed for queries into the relations for quicker separation and organization.

The data warehouse didn't contain all the csv file data due to the time it would take to go through all 700,000+ matches was going to be around a week if not more. We chose only to run the script which organized the preliminary database into the data warehouse for 24 hours and had a total of 720 matches in the data warehouse which is only a marginal amount of the original data set. Though this isn't a substantial amount we felt that performing the aggregations, cuts, and analysis of the data cubes that were based on the data warehouse as a good way to analyze the data in conjunction to also using the original csvs for further exploration and visualizations. While the data warehouse wasn't our main way to understand the data, even the small amount of matches helped us answer the question of the best strategy to get first place better.

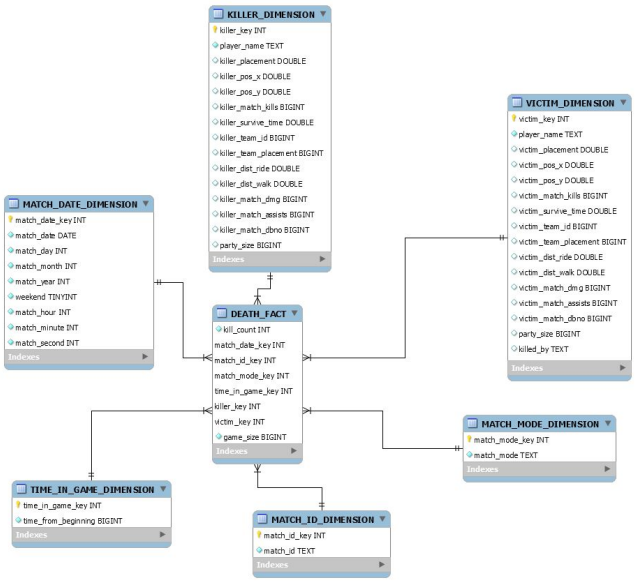


Figure 1: **Star Schema model our Data Warehouse backend was based off of. Both csvs were separated, organized, and inserted with a Python script.**

This MySQL backend enabled for easy integration with our Data Cube tool which was Python Cubes. Through Python functions, we could drilldown, make cuts, and aggregate the data to further understand it. Since each death was represented in the fact table, we were able to drill-down based on the dimensions we wanted to explore. For example, we could drill-down on the victim_dimension and the killer_dimension to get all the deaths which a player killed another. This allowed analysis on the distance of kills and if players that won had kills that were longer or shorter than the average. In addition, we could see the average amount of deaths that occured in all games in the warehouse through the aggregation of Python Cubes.

Through aggregation of the deaths in a game with the cube, we found that a match had an average of 91 deaths. In addition, aggregations of the kill distance found that players in first place had an average kill distance that was shorter in comparison to other players.

3 Data Classification/Clustering

For classification, we used the Python library scikit-learn which provides machine learning algorithms for usage with data sets. We elected to use Naive Bayes Classification for our dataset. Naive Bayes uses Bayes' Theorem with the assumption that there is conditional independence between attributes. Scikit-learn offers multiple different types of Naive Bayes classification methods. These were Gaussian, Multinomial, and Bernoulli Naive Bayes.

In the classification of our data, we chose to solely use the aggregate data since these provided the final outcome of all the players in a match and had attributes like player_dmg, player_kills, player_assists, etc. which we thought would be useful in determining the end placement of a player. In order to prepare our

data we first needed to decide what we were going to use for training and testing datasets. Since the aggregate data came in five separate csvs of size 13,000,000 or so we figured that a single csv would work for both our training data and another would work as our testing data. We opted for agg_match_stats_0 as training and agg_match_stats_3 as testing. Anything larger than these csvs would end up with a MemoryError due to their size.

Further preparation include “cleaning” the sets in which we labeled everybody who got first place as 0 and everyone who placed outside of first as 1. This avoided multi-class classification with increased the model’s prediction accuracy.

Scikit-learn enabled easy classification through first fitting the training data to a model with a .fit() method on the Naive Bayes object provided by the library. Once the model was fitted we could predict using the test data with a .predict() method also on the object. The prediction could then be compared to the test data to find out how many mislabeled points there were.

We found that Bernoulli Naive Bayes provided the highest accuracy out of all three classification methods. With an accuracy of 93%, Bernoulli Naive Bayes mislabeled 962,467 points out of a possible 13,840,680. With these results we were able to find the mean kills, assists, down but no outs, distance walked and rode, damage and survive time of players that were in the training data along with their standard deviations.

Mean of Attributes of First Place from Training Data

Attribute	Mean Value	Standard Deviation
Kills	3.63	3.80
Assists	1.03	1.21
Down but no outs	1.94	2.56
Distance walked	3005.05 m	3495.29
Distance rode (vehicles)	2890.00 m	3495.83
Damage	396.76	1.21
Survive Time	1818.50 seconds	255.54

Figure 2: **Table of the mean attributes of players in first place from the training data.**

A worry we had with these results was the standard deviation of the distance walked and the distance rode attributes. This may suggest that the distance walked and rode attributes had an error during the fitting process so we can’t have complete confidence in the results of these attributes. In comparison, the standard deviations of all the other attributes are more sound and aren’t as outrageous meaning those results are good basis for our findings.

An important note is that, again, this assumes conditional independence of attributes hence “naive” but the method of classification results in the real world have worked quite well meaning there isn’t any reason to not doubt these results. In addition, naive Bayes only requires a small number of training data for its model meaning that our large dataset aids in the model fitting as there are an extensive amount of data points in it.

4 Data Visualizations

For our heat map visualization in both Erangel and Miramar map, we used jupyter

notebook to load our data set from deaths file. In order to get the correct heat map for victims who were killed by actual players, we dropped the data which has NaN killer placement and NaN killer name. And the heat map visualization did not contain all csv files from deaths file, but it should also reflect the accurate situations for both maps. Because each csv files contains a lot of data. We used the scatter from matplotlib to draw the points by using victim_position_x and victim_poistion_y and shaped the points with the value of victim_placements. Also, we drawn different victims density of areas with different colors, which is able to illustrate the difference between each places clearly. And those heat maps also can show the most dangerous area in a game.

Also, we created several histograms from aggregate data to show the frequency for each way of death across 792 matches, which can help us to know which weapon was used more among players. The aggregate data also provides the value for time of death. This allowed for creating a histogram about time of death across 792 matches, we are also able to analyze the dangerous time interval to fight with other team in a game. Based on those visualization, it can provide some strategies about how to get more kills with some specific weapons, and how players avoid fight during a specific time interval to get high a rank. More information and exact findings of these visualizations are found in the visualizations section. These data visualizations were accomplished through queries into our data warehouse as this provided a transaction view of deaths we could see the time in game for each.

Figure 3: **Map of Erangel which will be used with geographic player locations to produce a heatmap of player deaths. This map was produced by Reddit user /u/c_a_turner on the /r/PUBATTLEGROUNDS subreddit. Other maps can be found at https://www.reddit.com/r/PUBATTLEGROUNDS/comments/6g1u2v/alternative_pubg_maps_topographic_realistic_raw/.**

KEY RESULTS

The key findings we have discovered includes what the best weapons to use in PUBG are, and the best locations to drop or navigate to based on previous death data. The 3 weapons that have been used to kill the most players are all assault rifles (as shown in Fig. 3) which suggests that these are not only the most popular weapons in the game, but the most effective as well.

We have also noticed that there are significantly more player kills in the various named locations/towns across the map (as shown in Fig. 4) which suggests that in order to have a better chance of staying alive throughout the game, you should try to avoid these areas.

The time that a player/team stays alive plays a vital role in determining their placement for that match, somewhat regardless of kills, especially since the number of players who are killed in the first 600 seconds of the game is exponentially higher than those killed throughout the rest of the game (as shown in Fig. 5). If a player/team can stay alive for as long as possible, their chances of placing top 10, top 5, or even top 3 all increase drastically.

We found that, on average, the best team size is a 2 man squad. The average number of kills for a winner is between 3 and 4 kills. The average number of knockdowns (but not kills) for a winner is about 2. The average damage a winner deals is about 400. A winner will get at least one assist on average. And finally, the average survival time of a winner is about half an hour (1818 seconds).

APPLICATIONS

The main application of our findings is applying them while playing the game. Since our goal was to find the most effective methods to reach first place, players can apply tactics based on the findings to gain an upper hand on opponents. In addition, its growth as a viable eSport means that teams that compete competitively can too use findings like these to understand the game further and what it takes to win. In general players should be aiming for doing at least as good as the “average” winner, whos results can be seen above. In addition, these players should also keep in mind that assault rifles tend to get the most kills, making them the most effective weapon type in the game, and they should avoid any named locations on the map in order to stay alive as long as possible.

VISUALIZATIONS

Figure 4 (below): This histogram shows the most lethal across 792 games in our data warehouse. As seen, ignoring “Down and Out”, the top three weapons were M416, M16A4, and Scar-L. All three of these weapons use the 5.56mm ammo which indicates that weapons that use this ammo are more lethal. In addition, the Mini 14 uses the same ammo. In comparison, the other types of ammo have little representation as the 7.62mm ammo only has a single weapon on the chart with the AKM as well as 9mm with the UMP9 and 12 Gauge with the S1897. There is a lack of sniper rifles on this histogram and light machine guns further indicating that assault rifles are most commonly used.

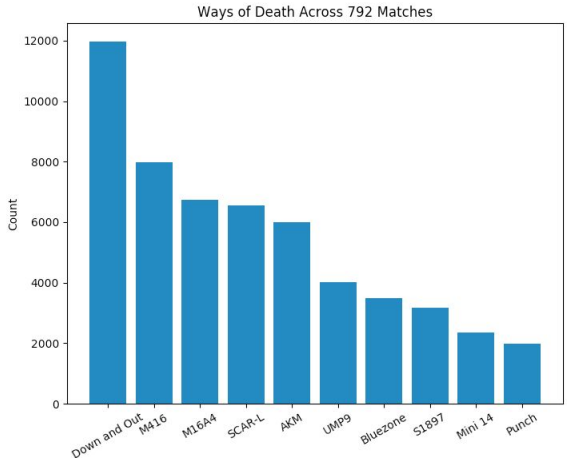


Figure 5 (below): This histogram illustrates the time of deaths across the 792 games in our data warehouse. As seen, the most active time in terms of deaths during the game is early meaning that if players can survive past 300-450 seconds the likelihood of placing in higher spots dramatically increases. This further indicates that the first five minutes during the game are

absolutely crucial since as time goes on the amount of death stays relatively even with a few rises and falls. This also shows the length of games that can occur. Naturally as there are less players there are less deaths, but again this stresses the importance of the early game.

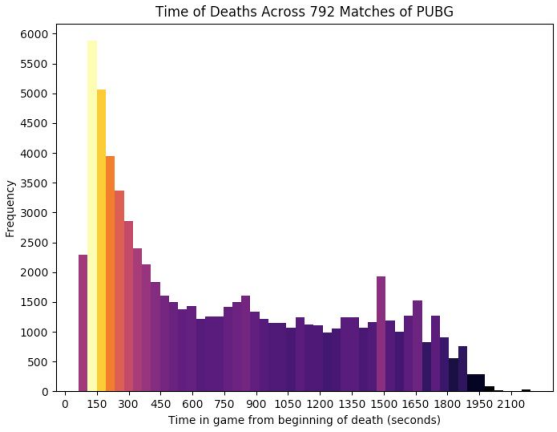


Figure 6 (below):

Visualization of scatter above the Erangel map to show the most dangerous areas of Erangel. As seen the darker blue spots are the area of least victims in the map not including the areas that have little no victims. This exhibits that a lot of the action occurs in areas that have a large amount of buildings which makes sense as these are areas where loot spawn. Compare this with Figure 3. The edges of circles might indicate where players died due to being on the edge of the circle while down. In this case, players are unable to really move meaning that a large amount of deaths occur by knocking someone down and meeting their fate by being in the “Bluezone”. This means that a direct kill from a weapon is not the only strategy when in a gun fight with another player. One, depending on the time of game, could be to knocking them down and fleeing if the circle is close or ammo/resources are limited on the players end.

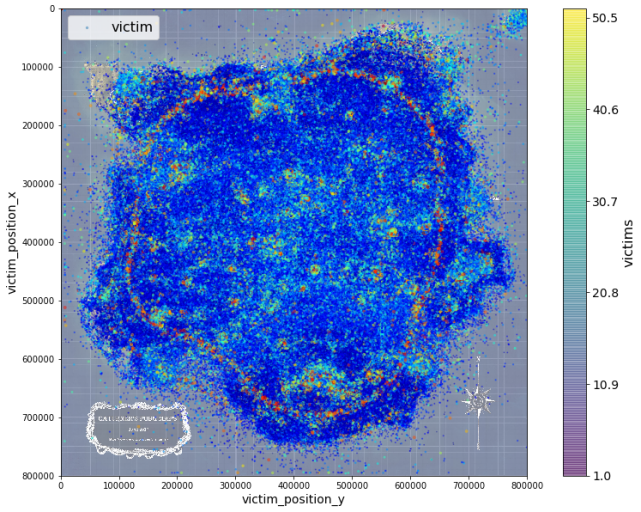


Figure 7 (below): Visualization of scatter above Miramar map to show the most dangerous area in Miramar. The heat map results follow closely with the Erangel map with the lighter blue colors highlighting areas of towns and loot concentration.

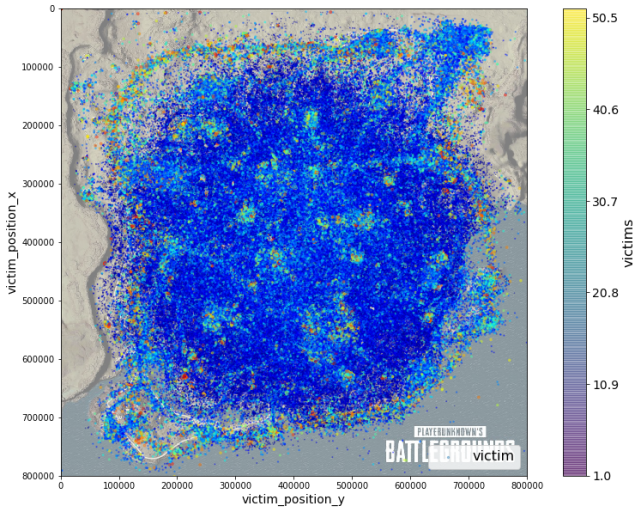


Figure 8 (below): Map of Miramar for comparison with Figure 7. As seen, the areas of light blue, indicating higher number of victims, trend around areas of high concentration of

buildings and towns.

Figure 9 (below)**:** Visualization of the correlation matrix for numerical features of the aggregate data set helped us understand the relationship between certain features of the data set. Being able to see the more closely correlated features provided inspiration for engineered features such as player ranking and match performance grade.

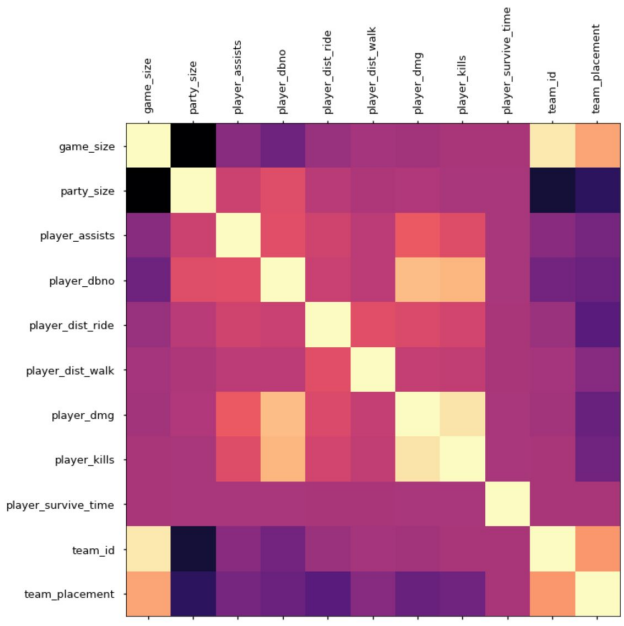


Figure 10 (below) implies that within solo games players who traveled longer distances tended to be players who dealt less damage, while players who traveled short distances were likely to inflict more damage. This makes sense, since players who land in populated areas engage in more fights during early game and tend to survive for less time, but are aggressive and deal more damage before they are killed. A cautious player on the other hand may travel far to avoid conflict, and will only engage when necessary.

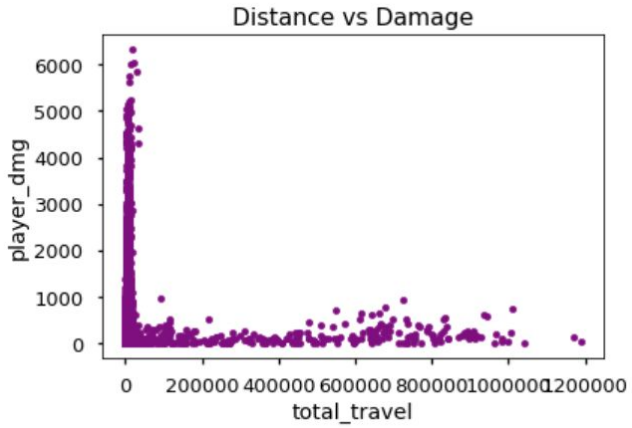
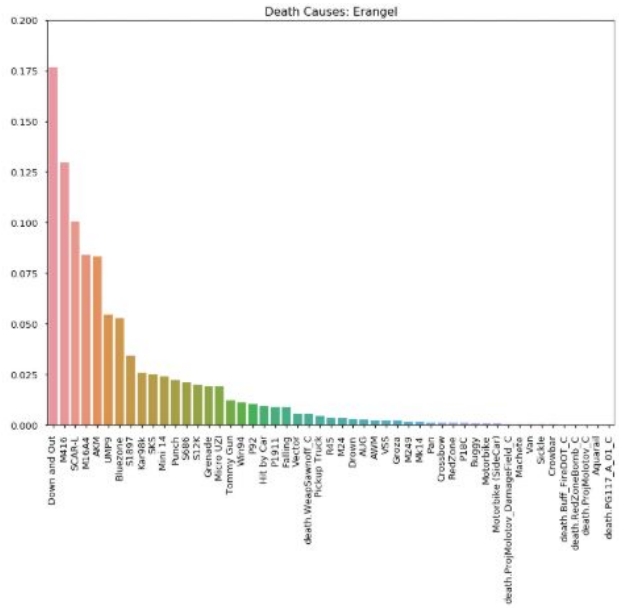


Figure 11 (below): This histogram expands further on Figure 5 using the csvs rather than the data warehouse. Since the warehouse only contained so many matches this takes into account even more matches. As seen, the results follow closely with what we saw in Figure 5 which was that the M416, Scar-L, and M16A4 being most responsible for the causes of death by a weapon excluding “Down and Out”. Again 5.56 mm ammo weapons find themselves with higher causes of death than other weapons in the top 10. This also further solidifies our point that an assault rifle is the most viable weapon in the game and a user should try and find one at the earliest moment.



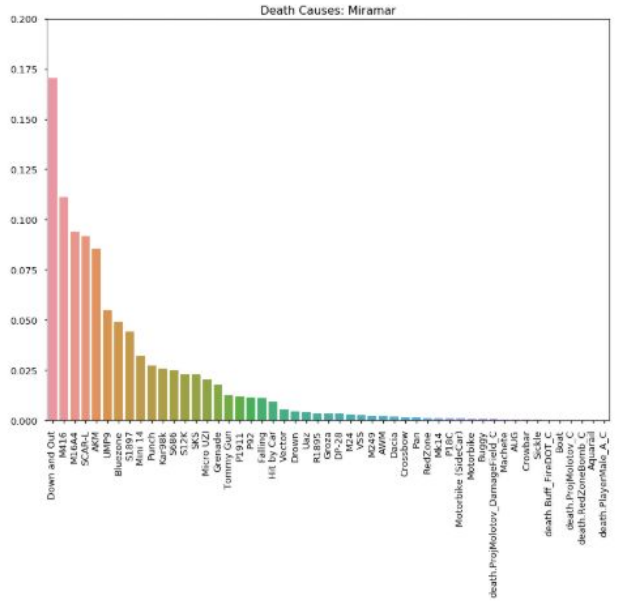


Figure 12 (below): This histogram shows the causes of death on the Miramar map. Like Erangel, the M416, M16A4, and Scar-L find themselves as the most efficient weapon disregarding “Down and Out”. This again adds evidence that these weapons are the most efficient weapons to have if you can find one. This data can also be used in balancing the game as since these are the most lethal the developer could chose to “nerf” these weapons if the mechanics continually favor them. Though in this map, the AKM has a greater presence which shows that the weapon viability may change depending on the map.

Figure 13 (below): This bar chart shows the summation of the ride and walk distances players accumulated during games. This shows that the use of vehicles during a match is high, but players tend to favor walking over using one. This could be due to vehicles attracting more attention due to their noise or players have a tough time find vehicles in general.