The Best Way to Win a Match of PUBG

JOSHUA RAMOS, University of Colorado at Boulder

TYLER MOORE, University of Colorado at Boulder

1 ABSTRACT

PlayerUnkown's Battlegrounds is a battle royale developed by Brendan Greene. The goal of the game is to be that last man standing. Teams consist of up to 4 people. Winning in a group is very similar to playing a solo match. A team must have at least one teammate alive at the end of the game.

The game has many variables when trying to win. Players begin on a plane, and are allowed to choose where they want to jump. There is loot scattered throughout the game, but it's distributed differently on the map. Some parts of the map are ideal for jumping because they will have more loot but it comes at a cost. More players jump in areas of the map with more loot.

The game has many variables for instance, players can pick up better armor, guns, and vehicles that can make the game different every time you play.

We aimed to discover what playstyle will most likely result in a win. To answer this, we categorized the play styles based on average speed and many kills will increase the chances of a player obtaining a win.

In conclusion, we found that an average speed of 1.7 m/s was a good pace because it placed a player in the top ~16. We also found that the higher number of kills a team or player has a relationship with placement. Data analysis indicated greater movements and more player enemy interactions resulted in a better chance of winning. Therefore, a more aggressive play style usually results in more victories compared to camping and playing very discretionary usually results in a higher chance of losing.

2 INTRODUCTION

Upon starting this project we wanted our questions to revolve around the idea of playstyle It took time to try to figure out how to extrapolate those play styles from datasets of seemingly unrelated information. We came down to two ways to determine playstyles, kills and placement and average speed and placement.

Team kills seemed like an excellent way for us to fit playstyle. Kills somewhat represent player interactions and jumping in populated areas. The map in PUBG is enormous, so it's easy to avoid players if you choose to. We thought team kills vs. placement would be a great place to start.

The main problem with this question is: How do we compare total kills with teams of 4 with solo players? We thought the best way to go about

this was taking entire team kills divided by the number of team members and using that as our comparison data. The creation of this new dataset allows us to mine our data with increased performance.

We also found distance traveled as a team to be relevant because the more a player moves, the more aggressive the player is. Players that tend to be more passive tend to camp and hide in certain spots unless it's essential to travel to a new place. Now the problem we have with this question is how do we turn distance into something useful because we want to know more about movement then displacement.

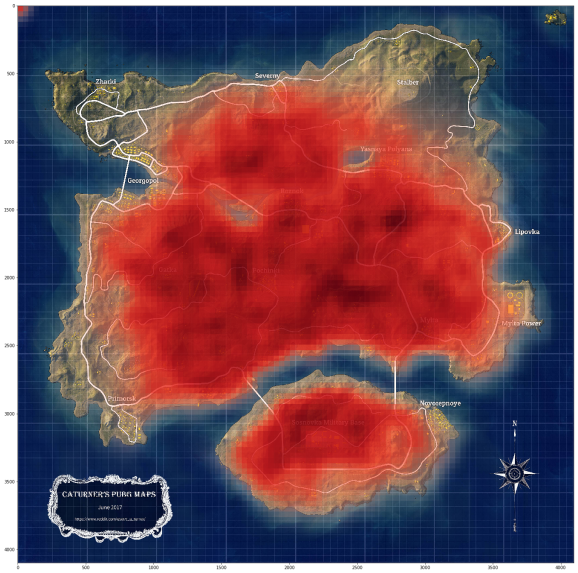
We had to average the team's distance divided by the team's playtime to get a speed in meters per second. Speed is a useful figure because it shows us how much of the actual game time did the player spend moving. Then, we compared that with their placement to get a better idea of what speed will result in a higher win percentage.

---

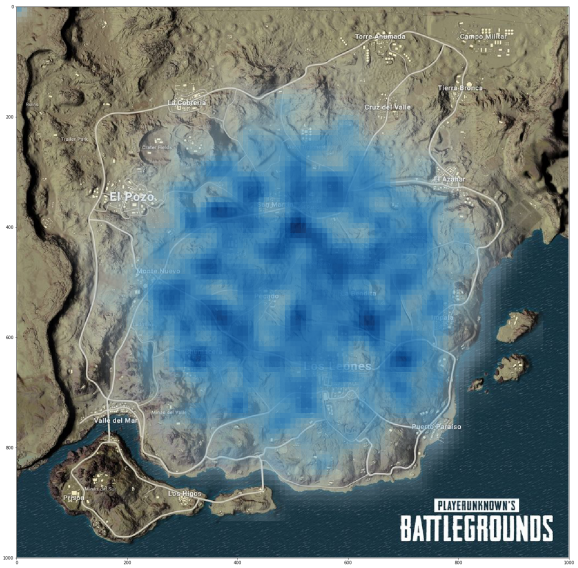
3 RELATED WORK

PlayerUnknown's Battlegrounds is a relatively new game with an initial release in 2017, so there is some research done, but there hasn't been as much work as we would've liked. We found two unique projects from the Kaggle page and found past research on the dataset we are mining on.

The first user we found to do some research didnt list his actual name listed but he a screen name called Skihikingkevin. Skihikingkevin took the data set and created heatmaps of the last circle based on where the final people died and their coordinates. Skihikingkevin used python for his data processing using the libraries as listed matplotlib, numpy, pandas, seaborn and utilized my jupyter notebook to show precisely what he did. In conclusion, he found, "Erangel: Although a lot more spread out, there are a few pockets of land that is more contested for final circle, primarily near very open areas between pochinki and mylta. It is worth noting that a lot of close river areas north are avoided so players based off of seeing the first circle, should be able to make an educated guess where the next circle will be.
Miramar: A lot different and more clustered. Notable clusters are between bendita and Leones, south of San Martin and west of Pecado. It is interesting to see that a large portion on the of the map is often never in the final circle." Using Skihikingkevin's research it can assist a player in jumping in a location where they won't have to move as much to stay in the circle.



Skihijinkevin's Erangel Heatmap



Skihijinkevin's Miramar Heatmap

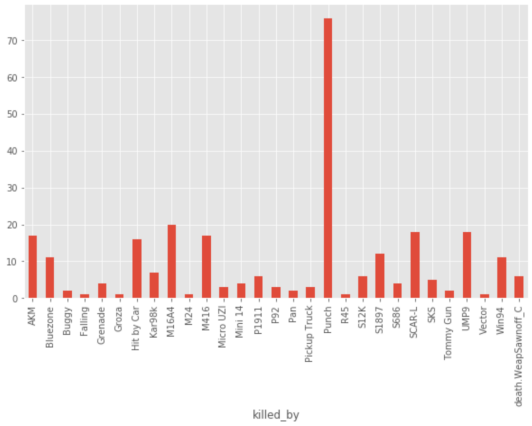
Another user named Christoph Egerland mined the same dataset we will be using. He focused on making maps of where players jumped the most. The tools he used were as follows: python, matplotlib, numpy, pandas, seaborn and utilized my jupyter notebook to show what he was doing. He focused on finding where the first kills of the game were, and used those positions to make a heat map. The heat map is useful because it can tell players where the best place to be in the final circle.

Christoph Egerland Erangel Jump Map

Christoph Egerland Miramar Jump Map

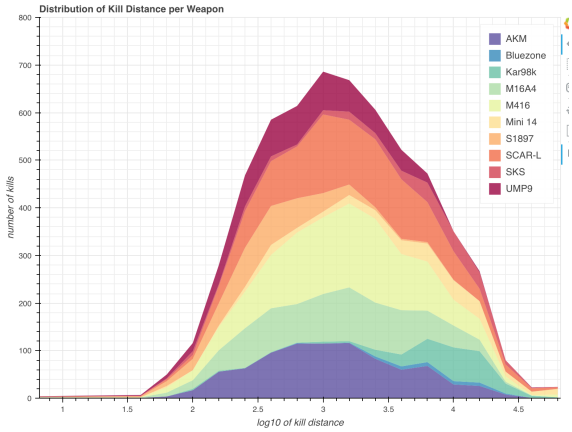
The final research project we found on the dataset based loosely on our project was conducted by a username Mithrillian. His focus was on finding the distance distribution of kills, and he also looked at what weapons were used to kill individual players.

He produced a lot of visuals based on his research that shows some excellent analysis. First, he made a visualization what weapons were used to kill individual players as shown below.



Mithrillan's Visualization of weapons used

Using the weapons used and the distance of the kill he created a distribution of kills and the distances those weapons killed an individual player as shown below.



Mithrillan's Visualization of Distribution of Kills by Weapon

Based on the previous research, we wanted to do something that we had not yet seen. This is a primary reason we went in the direction we did. We wanted to learn what the best strategy to give us the best chance of winning. After all, that is the primary purpose of the game.

4 DATASET

When we downloaded the datasets, we realized there are multiple attributes to look at and evaluate. In one of the datasets, it has all the information on deaths. The dataset involves who the killer was, their position, how much time has gone by, the victims name, the victim's location, etc. This dataset has over 65 million entries.

There is also another dataset based purely on a specific person for an individual match, including the match ID, their team ID, how far they traveled, how many kills they had, the type of game they are in, the amount of time survived, their individual and team placement.

In addition, we had to create a new dataset because we needed more information than the datasets present could give us. This new dataset holds all the team data. This means that we took all the data from the individual dataset and joined their data together to form teams. By creating this new dataset, we were able to decide which attributes we needed. They include: the match ID, their team ID, number of players in a match, the party size (number of players in a team), the average amount of distance traveled by the team using a car, the average amount of distance moved by the team walking, the average survival time by the team, the total number of team kills, and where the team finished the game. All these attributes were used in helping us figure out answers to our original questions. We will talk about how we created this new team dataset in the section "Main Techniques".

Every single attribute in all the tables involves integers, except for the match ID which is just a random string value.

5 MAIN TECHNIQUES APPLIED

The three main techniques we utilized in this project were data integration, data cleaning, and then finding results to help us answer our questions. When we first downloaded our dataset, it did not look like we would need to do that much data cleaning. However, we quickly learned this would become the most time-consuming part of the project. All the data was already in numbers, and there were not any null values which led us to believe it wouldn't be too bad. But after we integrated two tables into one, we realized we needed to clean a large part of it.

Data Integration

The first thing any dataset needs is to integrate the data to create a new dataset which contained all the team information. The new dataset was significant because a lot of our questions involved team data. The techniques we used to do this were the functions Group By and Concat

in the python library. By using these built-in functions, we were able to look at the individual data and group all the attributes together based on similar team ID and game ID. This means that we looked at all the players with the same game ID (to make sure they were playing in the same match) and team ID to collect all the individual player data. For example, for the attribute "team kills," we summed up all the individual player kills who had the same game ID and team ID. For the attribute, "team distance traveled," we took the mean of all the individual player's with the same match ID and team ID and calculated distance traveled. We chose the mean instead of summing them up, like we did for team kills, because a group usually moves in a pack, so it does not make sense to add up all the individual player distances.

Data Cleaning

After we created the new team dataset, we quickly realized some of the information was not correct. For example, in the attribute "party size," just showed what the game mode they were playing. While playing PUBG, a player can enter a solo game, duo game, or squad game. So "party size" showed player teams of either 1, 2, or 4 which represents the game mode they are in respectively. We wanted to determine the exact number of players on each team. For example, if a player signs up for a squad match, there could be a chance to only have three players on the user's team. We wanted "party size" to show there are only three people on this specific team. Therefore, we had to change this attribute so that it added up all the players with the same game ID and team ID to show the number of players on each team.

The next step in the data cleaning process including the "speed" attribute. One of the questions we were trying to figure out did the amount of distance traveled affect the players chance of winning the match. However, we realized that of course people who make it to the end of the game would have traveled farther than someone who dies early on. This would not prove anything. So we created a "speed" attribute which divided the average distance traveled by a team by the survival time. In PUBG, they measure distance in meters and time in seconds, so our "speed" attribute measured in meters / second. This new attribute is what we weighed to determine if there was a certain speed a team could travel to become victorious.

Another essential data cleaning technique we had to go through was getting rid of outliers in the "speed attribute." To visualize the data, we did a k-means clustering algorithm on the "speed" data to see which speed creates the best chance to win. This k-means clustering algorithm is further explained later in this paper. When we ran the data, we saw there were multiple points with a rate of over 400. After doing some research, we found out the highest speed a player could go in this game is around 7 m/s. This means that some players were using cheat codes or something similar to get a speed of much higher than 7. To work around this, we deleted all data that was over 10 m/s. We also eliminated data that had a speed equal to 0 m/s, because this means the teams were not playing at all.

Clustering

Once all our data was clean, the next step was to start finding answers to our questions. As previously stated, we used the k-means algorithm to detect clusters among our speed data. We used python to do this by importing the KMeans library and using the built-in function. We wanted to use six different clusters and were able to color code each group to see the distinctions. We also used X's to show where the centroid was in each cluster. While we did not get as much distinction as we'd like, the results showed a steady increase in team placement as the average speed went up. We will talk more about this is in the results section.

Association

Another question we had was: Does the number of kills a team has to affect their overall placement? We thought the best way to go about answering this question was association because we were able to use conditional probability to figure out the highest chance to win. In our python code, we looked at three different scenarios. In the first scenario, we looked at all the data with less than three kills. This means that we only included teams with less than three kills to see what the average placement was. We then analyzed another scenario which included all teams that had bwtween 3 and 8 kills. Finally, we analyzed data for all teams with 8 or more kills. As predicted, we saw the average placement increase with teams who had killed more players.

We also wanted to see what the winning percentage would be for these three scenarios. Once again, we looked at teams with less than three kills and determined what the winning percentage would be. We then repeated this for the other two scenarios. As predicted, the more kills a team had, the higher chance they had of winning the match.

However, after going through this process, we realized this data might be inaccurate. We were looking at all of the team data which means all solo, duo, and squad games. This strategy is not fair for a match that is played with solos because it is much more difficult for one player to get 8 or more kills than it is for a whole squad. To address this problem, we divided total team kills by the number of players they had instead of looking at total team kills. For example, if a team had eight kills with four players on it, the kills per player would be 2. This makes our data much more accurate because if a solo player had two kills in a match, these scenarios would be compared fairly. After coming to this realization, we changed our conditions because it would be scarce for a team to have eight kills per player. Instead, we tested on teams to have 3 or more kills per player to see what the average placement and winning percentage would be. The winning percentage went down as expected but it was still higher than teams who averaged less than three kills per player.

Combining

After determining which speed and number of kills it takes to create the highest chance to win, we decided to join the requirements together to see if it increases the winning percentage even more. We made another conditional statement that only looked at teams with a certain speed and kills per player. We will go more in-depth on our results in the next section, but surprisingly, the winning percentage went down.

---

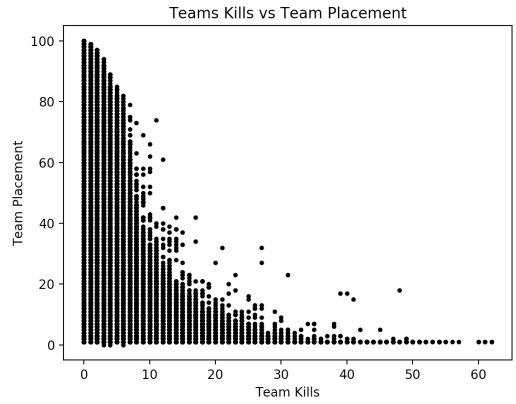
6 KEY RESULTS

The primary results we found were more aggressive playstyles tend to have an advantage against a more passive style. We had to learn what made a player more aggressive or more passive, which will be defined later in this section. The main result we found was aggressive play styles tend to have a higher chance of winning. Coming to this conclusion wasn't easy.

Running code on large datasets can be very time-consuming. Our dataset had roughly 64 million unique data points. Each had about 12 attributes.

Running that much data required a lot of data cleaning and powerful computers. It would sometimes take 15-30 minutes to run our code just to look at the results. We stuck to running our plots on 64,000 unit sets before moving to are much more massive sets, which helped to expedite the process.

The first plot we made was the plot below. We used our team kills data, which was created in the cleaning process by combining player stats from players on the team. After that, we plotted the kills and team placement to see what trends we found.



As you can see from our scatterplot, there's a significant drop off in placement around ten kills. We noticed the teams that had more than ten kills would place somewhere in the top 20. This graph led us to our actual data mining finding confidence in kills and placement.

We used conditional probability to see what was the average placement of the team with a certain amount of kills the intervals we used were less than 1, between 1 and 3 and 3 and above. Our results are listed below.

```
Average Placement for less than 1 kills: 36.67432441322
Average Placement for 1 or more kills and less than 3: 24.806010829791635
Average Placement for 3 or more kills: 12.791688527496298
```

We found that 1 or less team average kills correlates with a placement of 36. Teams of 4 tend to have the highest placement of 25, teams of 2 tend to have the highest placement of 50, and solos have a placement of 100. On average, the placement from 1 to 3 was 24.8 which shows that as you get more kills on average you will place higher. That pattern continues to hold true with 3 or more kills. With 3 or more kills you will find an average placement of 12.79, which is an excellent position to see.

Using data mining we were able to find that accumulation more kills results in a better average placement, and a more aggressive player, based on our definition of accumulating more kills, will place higher. With these results, we can make a better hypothesis that the better way to play is aggressive.

To further our research, we wanted to get to the meat of the topic and see kills versus winning. Since PlayerUnknown's Battlegrounds is a battle royal game, there are a 100 players that will join a game, and only one will prevail. In a statistically perfect world, there is a 1/100 chance that you win making your statistic of willing only 1%. The 1% is the comparison that we will be using the next information we gain.

We did very similar conditional probability finding the change in chances of getting a victory compared to kills. After running our code on the large dataset, we were able to see some fascinating statistics.

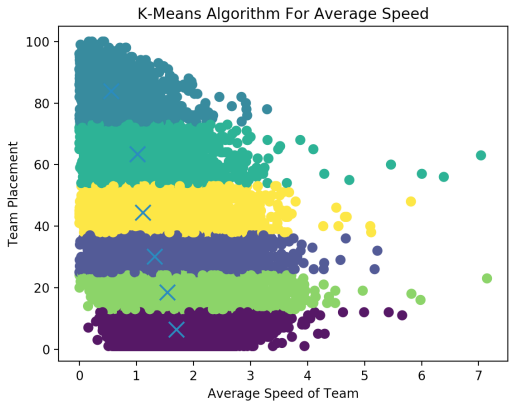
```
Percentage for first place finish with less than 1 kills: 0.08%
Percentage for first place finish with 1 or more kills and less than 3: 2.09%
Percentage for first place finish with more than or equal to 3 kills: 17.39%
```

As we can see if you get 1 or fewer kills per player your chance of winning is well below 1% meaning that if an individual or a team can’t get a kill, that means it's improbable you’ll ever win a game. Which makes perfect sense because you need at least 1 kill to win the game.

The next stat we found was there is a 2.09% chance of winning if you get between 1 to 3 kills per player. That's very close to doubling your initial chance of winning. This, again, shows that more kills/aggressive playstyle increases the chance of winning.

The final statistic shows some gripping stats. The chance of winning with 3 or more kills per player is 17.39%. This makes for a 17x greater chance of getting a victory. These findings better solidify our hypothesis that more aggressive playstyles result in a higher chance of victories. This information, paired with our average speed traveled, will help us get a clear answer in what playstyle will result in the highest chance of winning.

Speed is a great way to show playstyle because it shows distance traveled vs. time. It allowed us to see if a player tends to move around more or kept still and saw what placement an individual placed vs. the distance they traveled. We decided to do this using k-means clustering and plot our data. The graph produced is shown below.



As you can see, the graph produced sections based on placement and it also displayed the centroids.

The centroids are what we used to show the average speed of that cluster. The centroids move to higher average speeds as the placement gets lower and lower. At a placement between 75 and 100, the mean speed is 0.8 m/s. At a placement between 55 and 75, the average speed was 1.0 m/s. The next cluster was between 38 and 55 with an average speed of 1.2 m/s. After that, the cluster was between 22 and 38 with an average speed of 1.35 m/s. The second to last cluster was between 12 and 22 with an average speed of 1.6 m/s. The final cluster was between 12 and 1 with an average speed of 1.8 m/s.

As we can see from the results speed does make a difference. The more you move, the higher chance you have of winning, and with an average speed, there's a great chance of winning. Which once again relates to an aggressive playstyle.

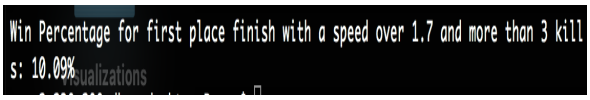
A player with a high average speed makes them place higher, so if a player is more passive and tends to camp more then they will likely place between 50 and 100. Where speeds above 1.0 m/s tend to place in the top 50% and the higher the speed, the better the placement. We found the sweet spot to be around 1.75-1.85 m/s.

As you can see the scatter plot has a lot of variance in speed vs placement. It's not entirely accurate to say if you average 1.75 m/s it doesn't always mean you will win likewise with kills but it does put a player at an advantage and shows what skills the game requires from a player, and the aggressive game style can be seen on a competitive level.

The last thing we wanted to do was combine the stats we found and see if the rates add up and see how much it increases your chance to win. We took the winning cluster from our k-means graph and then compared that with getting 3 or more

kills which we proved to give you the highest chance of winning in our kills analysis.

How we coded it was using the knowledge we've previously mind and combining it, which made it a simple for loop that took found the measurements we were looking for and grabbed that data and compared it to the total data. The information we found is shown below.



Compared to the base statistic we talked about earlier of a 1% chance of winning you are 10x more likely to win if your speed is over 1.7 and you have 3 or more kills. This statistic is lower than just maintaining a speed of 1.75-1.85, and this makes sense because we are putting more requirements on our data, as some players traveling 1.75 didn't always get more than three kills. We could classify this as overfitting, but we thought it would be something useful to include.

Overall, this assignment helped us gain a better grasp on data mining. It taught us what to do as well as what not to do. We learned that preprocessing can sometimes take longer than actually mining that data, as many algorithms are included libraries in python. While the data is always different. Sometimes there are null values, and other times there are things that don't make sense. After the data was clean, it was nice to quickly implement libraries and find our statistics.

7 APPLICATIONS

The knowledge we gained can be applied very quickly because it is a video game and so it is easy to test out this new found strategy. One way to do this is actually to play the game ourselves. We played several games and determined if killing more players or maintaining a certain speed would increase our winning percentage.

We took data of all the games we played to see if our results were accurate. This is a small sample, but we played 50 games. Each time we played, we landed in the same city to keep our results constant. We played duo in all 50 games because we knew how we were supposed to play. If we had other people playing with us, they would not have known our playing style and could have skewed our results. Since we did duo, we were aiming for a total of 6 kills per team, or 3 per person. This is because we found out there was a 17% chance to win the entire match when each player on the team averaged 3 or more kills.

Out of the 50 games we played, there were only ten where we had six kills or more as a team. In those ten games, we won two of them. This means given we had 6 or more kills and our winning percentage was 20%. This is a little higher than what we found in the results, but this is such a small sample it is hard to compare. The important thing is that these percentages are close.

In the other 40 games when our duo team had less than 6 kills, we only won once. This turns into a winning percentage of 2.5%. This is also a little higher than the results we found, which was 2.09%.

From our test experiment, we could see that the results we found on the large dataset are pretty accurate. This means that teams who average 3 or more kills per player in a given match have a much higher chance to win than teams that do not. There are two reasons why this is the case; the first is because when someone gets killed, they drop all the materials and guns they were carrying. This means the player that got the kill now has the opportunity to pick up the dead player's guns depending on if they are better. This allows for a team who gets a lot of kills to usually hold the best stuff in the game because they have the most opportunities to pick them up. The other reason is that a team who gets a lot of kills are usually very skilled and therefore has a better chance to win.

The other thing we tested for in our project was to see if speed had anything to do with a higher winning percentage. This was hard to check for because there is no way to calculate our speed unless we are looking at the data collected during our game. Instead, we just made sure to continually keep moving and see if that affected anything. However, it was pretty easy to understand that the thing was changing our winning percentage the most was trying to get 6 kills or more. So in future cases, the best way to win is to try and get as many kills as possible to get the best guns in the game.

---

8 SOURCES

Egerland, Christoph. "PUBG Data Analysis." PUBG Data Analysis | Kaggle, www.kaggle.com/chegerland/pubg-data-analysis.

KP. *Final Circle Heat Map*. www.kaggle.com/skihikingkevin/final-circle-heatmap.

MITHRILLION. *Kill Distance Analysis*. www.kaggle.com/mithrillion/kill-distance-analysis

URL: https://www.kaggle.com/skihikingkevin/pubg-match-deaths/