Chicken for Dinner

A data mining project on discovering how to win in PUBG

Zijun Xu
University of Colorado Boulder
zixu6769@colorado.edu

Jonathan Young
University of Colorado Boulder
joyo2566@colorado.edu

Jianyi Chen
University of Colorado Boulder
jich2029@colorado.edu

Problem statement

Playerunknown's Battleground (PUBG for short) is a popular multiplayer online battle royale game developed and published by Bluehole, a Korean publisher. There is a maximum of 100 players per game. The game starts by putting all of the players on an airplane, which goes through the map in a random direction. The players are free to choose when to jump out of the plane after the back door opens. Various resources are provided across the map, including weapons (with a large number of possible combinations of parts), armor, medicine (healing), vehicles, etc.

Additionally, after a short while when all players land on the ground, the game forces all the players to move to the “safe zone”. If a player remains outside the safe zone, they will continually lose vitality. The size of this safe zone decreases with time, picking a new circle of the same relative size inside the new circle. This is to force the player’s closer together to instigate struggle (as shown in figure 1). The game ends when there is only one team left.

A team may consists of 1-4 players at the start of the game, but only one must survive till the end for the team to win.

Statistically speaking, at the start of a game a single player has only a 1 percent chance of winning based solely on number of players. In other words, it is hard to win in PUBG! By mining the death statistics data, we aim to create a survival guide for any player who wishes to have a better chance of survival in the game. Our survival guide should be able to answer some interesting questions like: Does skill or luck win more games? Do players that get more kills win more games? Where do the most player deaths occur? Which weapon should they choose to ensure they have the best chance to win a fight? Under what situation should they use a certain type of gun?

Figure 1: A example of the safe zone reduction in playerunknown's battleground

Literature survey

We have found some interesting prior work that is related to our project. Christoph Egerland has done an exploration on where most of the people jump out of the plane in PUBG.[1] By using Jupyter notebook, Christoph scaled both maps' coordinates to fit the map image in order to draw density points later on the image file (shown in figure 2). Since the dataset doesn't contain the positions that players jumped out but only the positions of players' deaths, Christoph restricted his program to only take death positions in the first two minutes of the game - because most of the players that die early do not usually move far away from their jump locations.

Another interesting project was done by Justin Moore and published on August 8, 2017.[2] In his project *To drive...or not to drive*, he analysed vehicle usage in depth. First of all, he showed that the mean of average ride distance grows as there are more people in a team. Justin then drew a scatter plot showing the correlation between average ride distance and wins, as shown in figure 3. The result indicates that players who have the average ride distance between 2000 and 4000 are the players with the most wins. In other words, players die more often if they don't drive at all, or drive too much.

Figure 2: Density graph of where the players jump at the most

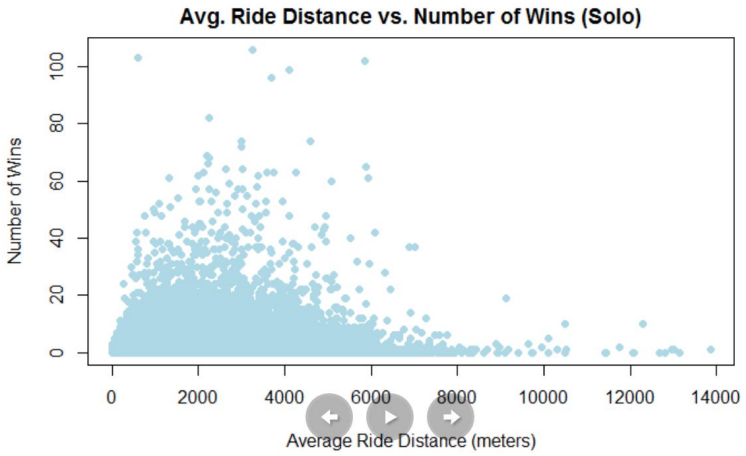

Figure 3: Average ride distance vs # of wins

Proposed Work Edited

Before we process our dataset, we will have to detect and clean any corrupt or inaccurate data to ensure the completeness and accuracy of our dataset. Specifically, we will locate those data that have missing

or inaccurate attributes then remove them from our dataset.

For example, there are some rows in our dataset that are missing the player's coordinates, which is inaccurate and needs to be cleaned. We are likely to remove attributes like victim name and killer name, which are also useless to us. In addition, to create the density graph showing players' coordinates, we will have to normalize the player position attributes in our dataset. The coordinate in our dataset is represented by big numbers in range from 0 to 800000, while our pictures of the maps are of the size 1000 * 1000. We will have to scale the player position by multiplying them with 1000/800000 in order to keep the player position and the picture size in the same page. Another step we will take in the data cleaning is the removal of all deaths that occured before the first safe zone reduction. This is to help with our proposed analysis. In addition, since there are two completely different maps in PUBG, we are likely to divide the data into two small sets based on the map types (MIRAMAR and ERANGEL).

After preprocessing, we will start analysing the data. Our analysis will be whether or not player skill or player luck has a higher precedence for determining a player's final ranking. Luck and skill will be defined by the relations between travel distance and number of kills. The players with lower travel distances had luck on their side because they were closer to the center of the safe zones during the size reductions. And those who have more kills (over some particular threshold) often have a higher level of skill. We will analyze a players luck vs their skill. To do this we will be looking at four different relations: low distance-low kills, low distance-high kills, high distance-low kills, and high distance-high kills.

Besides luck-skill analysis, we also would like to analyse the relationship between the time index of a safe zone reduction and the locations of the most player deaths in this time in order to create some mapping of where to avoid during particular instances.

Proposed Work Updated

We are adding more proposed work in addition to the original proposed work. When we are done with the luck-skill relation analysis, we will be able to answer the question that *whether or not player skill or player luck has a higher precedence for determining a player's final ranking*. However, this result alone is not enough to help a player get better placement in the game since they cannot control their luck. As PUBG is a shooting game at its core and there are various types of weapons; the player's choice of weapon is one of the most important factors when determining how long they can survive. Hence, we would look into the death_match_stats_ data and discover the relationship between the player's choice of weapon and their final placement.

Specifically, we want to find out the best choice of weapon in different states of the game (early, middle, late). As the early fights in the game tend to have shorter range than later games, different kinds of weapons might be prefered. To do this, we will have to divide the data based on the time of each safe zone. We will then look

into the used weapon for each of these time slots and find a relation between these choices of weapons and the player's final placement.

On the other hand, we also want to analyse the shooting range for each weapon and find out the optimal range for the player to use a certain weapon. In detail, we would apply the euclidean distance formula to the victim coordinates and killer coordinates to get a distance for the weapon that the killer used. We will store all the distance data for each weapon and cluster the distances for each type of weapon. By using the cluster technique, we will be able to find the most effective range of a certain type of weapon, hence we will be able to help the players to choose their weapons that works the best given a certain distance.

Data set

URL: https://www.kaggle.com/skihikingkevin/pubg-match-deaths/data
There are two main parts of our dataset. The first part provides detailed information on the match, such as weapon type, player position, player placement (ranking), round time. The second part gives more general information on the match, including distance traveled by foot and vehicles, player placement, player damage dealt, player kill numbers.

Evaluation methods

We will be looking at the four relations described above, blocking all players into these four bins. We hope to be able to use these bins to analyze a players relative luck or skill and which is a bigger factor for survival.

For the deaths during the reductions we plan to group the players who died during particular time indexes. Since the size reductions of the safe zone always occur at the same time intervals during any given game, we can create bins such that the size of the bin is the time between circle reductions, i.e. (0-first),(first-second) and so on. After the bins have been made we will look at the player locations and create a heat map for the given game.

Tools

We will be using MySQL and Jupyter Notebooks for cleaning and analysis. As well as Weka for visualization. For built in library, we used pandas, numpy, and alchemy.

Milestones

- Preprocessing
  - Data cleaning
- Creating the evaluation methods for luck - skill
  - Bin creation/bucketing of players
- Making bins for time indexes
- Mine the data/extract desired information
- Creating visuals

Milestones Completed

- Preprocessing
  - Data cleaning completed

    - Data integration canceled
    - All data into database
- Creating the evaluation methods for luck and skill
    - Started testing various numbers of possible methods

Milestones Todo

- Complete luck-skill analysis
- Prepare for weapon analysis
    - Making bins for time indexes
    - Divide the data
    - Mine the data
    - Analyse choice of weapons
    - Analyse range of weapons
- Create visuals
    - Cluster weapon ranges
    - Density graph for safe zone reduction

Reference

1. Egerland, Christoph. “PUBG Data Analysis.” *PUBG Data Analysis | Kaggle*, Jan. 2018, www.kaggle.com/chegerland/pubg-data-analysis.
2.Moore, Justin. “To Drive... or Not to Drive...” *PUBG Data Analysis*, 8 Aug. 2017, pubganalysis.wordpress.com/2017/08/08/to-drive-or-not-to-drive/

Feedback

We were given the feedback that we had not found any prior work to help us with our mining idea. To fix this we went online and found some prior work which helped us to narrow down our idea.

Milestones Completed (continued)

Since we have too much data for a Pandas data framework to handle storing in active memory, therefore we created a database with clean and desired data. To do this we created a Pandas dataframe to help clean our data. In cleaning we used .dropna() to remove any rows with missing values.

We also removed any rows where the player had no walking distance because this player was inactive and would skew our data. To speed up cleaning we ran it in chunks, cleaned a chunk, and then stored that clean data into a database.

During the copy into the database, we put constraints on which attributes were copied so that our database only contained our desired information. In this step we also combined all five of the files in each set (agg_match_stats_, kill_match_stats_final_). In the kill_match_stats_ files, we also dropped all rows with zeros.

We also added an attribute in the aggregate data to hold the total distance traveled for each player (walk + drive). The final step in our cleaning was to select only the games that were solo matches to be used to determine our luck and skill parameters.

Data integration is canceled. We have no need to integrate because we are

going to individually mine each data set for different information. There was no reason for us to join them into a single table.

Since we decided to do away with the data integration, we had to find a different way to remove the early deaths in a game. To do this we removed the lowest 30% of the final rankings. Since the lowest ranked players died early, and having quite a bit of experience playing the game, we determined that keeping only those ranked 70 or above was a reasonable method for removing early deaths from the agg_match_stats_final_ data sets.

In the kill_match_stats_ data we also removed any row where the player was killed by either being down and out, or where the killed_by data was strange.

Results So Far

For our analysis of the luck and skill parameters, we normalized the total distance traveled and our skill parameter which is equal to player_kills+player_dbno+0.5*player_assists. Then we graphed the relationship between the sum of our normalized parameters and their final placements. We also looked at the distribution of total distance traveled to see if the spread covered an acceptable range.

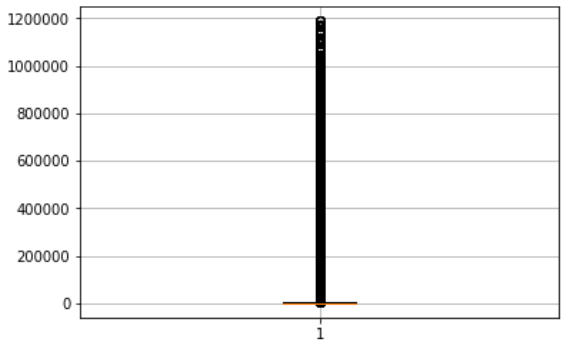

Figure 4: Boxplot of total travel distance before cleaning

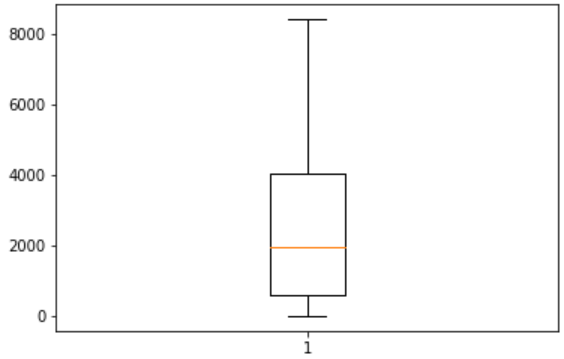

Figure 5: Histogram of total distance traveled to show spread, after cleaning

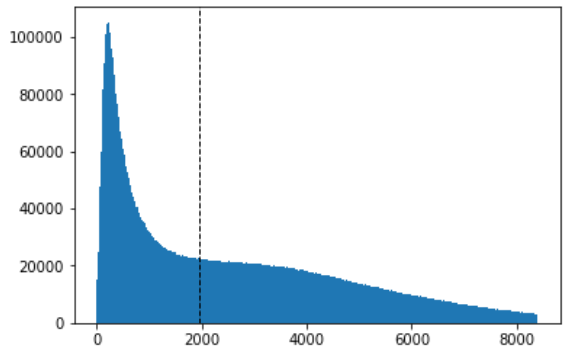

Figure 6: Distribution of total travel distance in meters, after cleaning