How to Succeed in E-Sports: A Data Analysis on PlayerUnknown's Battlegrounds

Wei-Hua Hsu (Wafer)

2020/5/15

1. Introduction

PlayerUnknown's Battlegrounds (PUBG) is a multi-player online shooter battle royale game, which is a similar game to Fortnite. In a game match, 100 players are parachuting onto the island. There are solo, duo, and squad game mode. Once the game starts, players scavenge towns and buildings for transports, weapons, and medical supplies. They can either decide to fight or hide, but the blue field will appear and constantly damage players, so as to aggregate players closer and closer. Therefore, players start conflicts until the last player who standing alive in the safe zone emerge.

2. Datasets and Analytics

As expected, the game results in PUBG are varying; thereby, in order to succeed in PUBG, finding the strategies of standing in match is crucial. This project analyzes the relationship between variables within the game data. There are 2 datasets with over 720,000 matches from PUBG, the datasets are downloaded from Kaggle. For the first dataset "agg_match_stats", it illustrates players' statistical performance from each match, including categorical variables such as team ID, player name, team number, game mode, and placement; and quantitative variables of player statistics like number of kill, assist, hit point, or survival time. On the other hand, the second dataset "kill_match_stats" provides killer and victim information. It contains the locations, how long did they survive in a match, victim and killer name, cause of death, and the placement. This project attempts to analyze the data, and then provide strategies for players who are eager to win the first place in PUBG.

3. Initial Hypotheses

To survive until the end, there are 3 main factors for succeeding in PUBG:

1. Player Mobility versus survival possibility

How can we conclude the relationship of distance of walk, distance of ride, and the player's survive time? Does it increase one's chance for survival if they utilize vehicles?

2. Player Performance

Compare with player statistics such as knockdown point (dbno), hit point (dmg), and kills. Does a higher knockdown point or hit points determine number of kills? How about the association between each other? Is that relating to win/lose?

3. Killer and the Victim

Find the possible distribution of killers' position on the map, and explore the frequency of the reason for dying. Additionally, generate the victims' location then visualize the position between killer and victim. Can we find a specific location in which is likely to have more survival chances? Which weapon takes an advantages for players to knockdown others.

4. Exploratory Data Analysis (EDA)

Categorical or Continuous Variable: The game dataset illustrates the players' performance in PUBG competition, while the death dataset gives the data of killers' and victims' position, survival time, and the cause of death. The discription and type of each variable have been noted below. This data are sampled by orginal datasets, please refer to appendix for more details.

```
# load game dataset
game <- read_csv("./data/sub_game.csv")
```

- game

title	description	variable type
date	match date and time	date
game_size	number of team in the match	continuous
match_id	the unique ID of the match	categorical
match_mode	TPP(third-person) or FPP(first-person)	categorical
party_size	number of players per team	categorical

title	description	variable type
player_assists	number of assists the player has scored	continuous
player_dbno	enemies who has been knockdowned but not killed	continuous
player_dist_ride	total distance of travel in a vehicle	continuous
player_dist_walk	total distance of travel on foot	continuous
player_dmg	total hit point that the player has dealt	continuous
player_kills	number of enemy the player has killed	continuous
player_name	name of the player	categorical
player_survive_time	how long did the player alive in the match	continuous
team_id	team ID that the player belonged to	categorical
team_placement	the final rank of the team within the match	categorical

```
# load death dataset
death <- read_csv("./data/sub_death.csv")
```

- death

title	description	variable type
killed_by	the weapons used by the killer	categorical
killer_name	name of the killer	categorical
killer_placement	final rank of the killer in a match	categorical
killer_position_x	(0~800000) x_position by killer	continuous
killer_position_y	(0~800000) y_position by killer	continuous
map	map in the match	categorical
match_id	the unique ID of the match	categorical
time	(second) how long did the victim alive in the match	continuous
victim_name	name of the victim	categorical
victim_placement	final rank of the victim in a match	categorical
victim_position_x	(0~800000) x_position by victim	continuous
victim_position_y	(0~800000) x_position by victim	continuous

Data Manipulation: parsed date, set the variables type, and renamed variables to be factors to factors in both datasets.

```
game1 <- game %>%
  mutate(date = ymd_hms(date),
         party_size = recode(party_size,
                             "1" = "solo", "2" = "duo", "4" = "squad")) %>%
  rename("player_survival_time(sec)" = "player_survive_time",
         "team_number"               = "game_size",
         "game_mode"                 = "party_size") %>%
  mutate(team_id        = as.factor(team_id),
         team_placement = as.factor(team_placement),
         team_number    = as.factor(team_number))
```

```
death1 <- death %>%
  mutate(killer_placement = as.factor(killer_placement),
         victim_placement = as.factor(victim_placement),
         killed_by        = as.factor(killed_by)) %>%
  rename("victim_survival_time(sec)" = "time")
```

Missing Values: In game dataset, 9 missing values gather in the same variable "player_name", it seems like

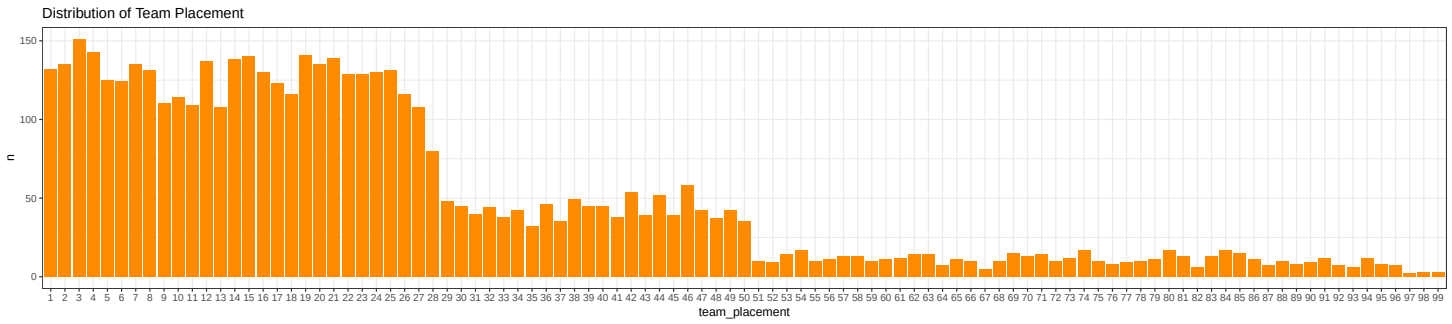
those data is basically correct and completed except for losing the player name. So I changed those NA values to ordered strings instead of removing them. On the other hand, I found some missing values in death dataset but I won't remove them. Please refer to appendix for details of death dataset.

```
# missing values for each column
game1 %>%
  summarize_all(funs(sum(is.na(.))))
# view the missing values
game1 %>%
  filter(is.na(player_name)) %>%
  select(player_name, everything())
# replace NA player names with ordered strings in player_name
game1_NA <- game1 %>%
  filter(is.na(player_name)) %>%
  mutate(tmp = cumsum(is.na(player_name))) %>%
  unite(player_name, tmp, col = "player_name", sep = "")
# drop NA values from player name
game1_dropNA <- game1 %>%
                drop_na(player_name)
# bind NA player names data with non NA player names data
game2 <- rbind(game1_dropNA, game1_NA)
```

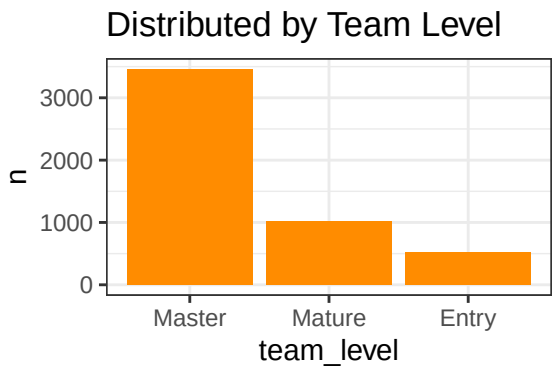
Observe Distributions and Create New Variables: I divided players into used vehicles or walked only, and created a new variable "mobility" to analyze the strengths or weeknesses of vehicle using.

```
game2 <- game2 %>%
  mutate(mobility = ifelse(player_dist_ride > 0, 1, 0),
         mobility = recode(mobility, "1" = "drive & walk", "0" = "walk only"))
```

According to the distributions of team placement below, there are approximately similar number of players in three intervals: placement 1 to 27, rank 28 to 50, and rank 51 to 99. Therefore, a new variable "team_level" has been created.

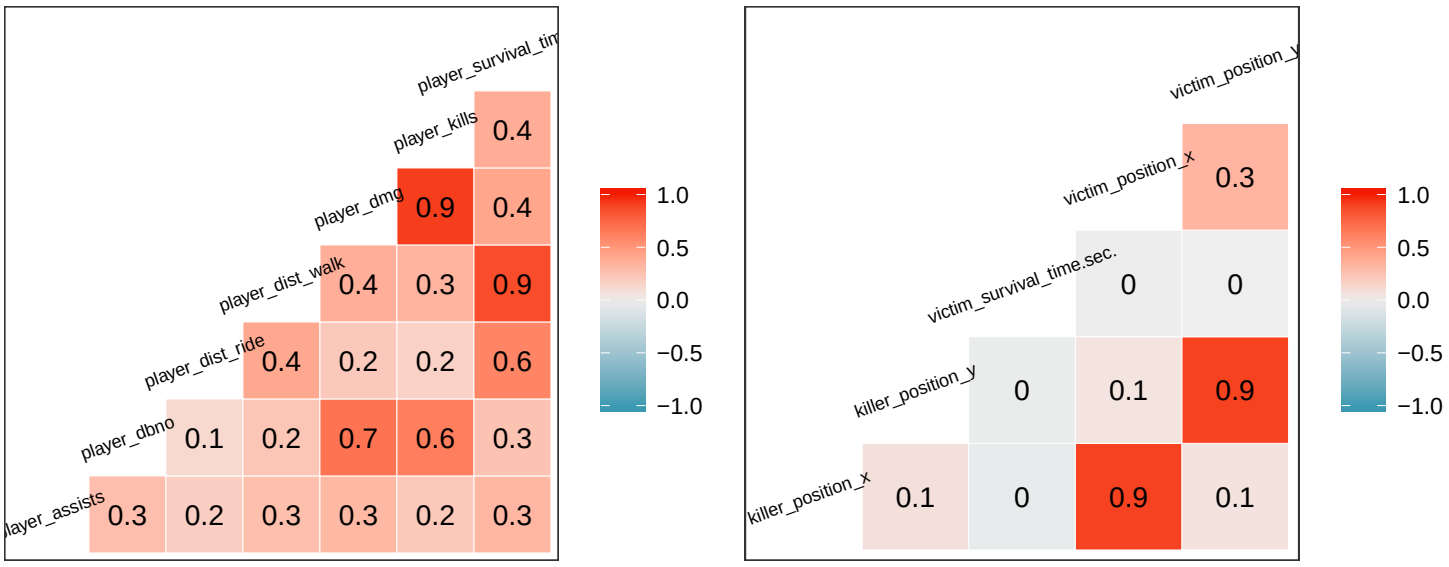



```
game2$team_level <- cut(as.numeric(game2$team_placement),
                        c(-Inf, 27, 50, Inf), c("Master","Mature","Entry"))

game2 %>%
  group_by(team_level) %>%
  count() %>%
  ggplot(mapping = aes(x = team_level, y = n)) +
  geom_col(fill = "dark orange") +
  ggtitle("Distributed by Team Level")
```



Correlation Matrixs: Visualize the correlation matrix by ggcorr(), this is from "GGally" package. Besides, only numeric data can be applied for this function, so I selected the continuous variables only.

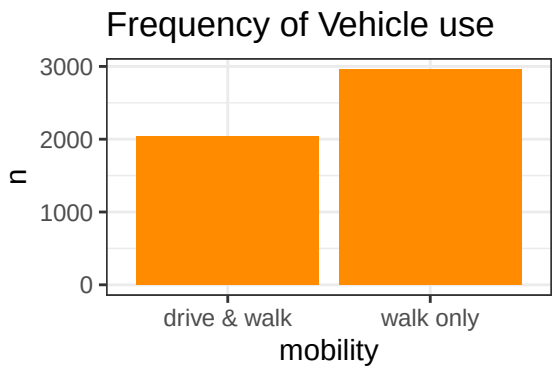
```
gamecorr <- ggcorr(game2[ , c(-1, -2, -3, -4, -5, -12, -14:-17)],
                   label = TRUE, angle = 20, size = 2.5)
deathcorr <- ggcorr(death1[ , c(-1, -2, -3, -6, -7, -9, -10)],
                   label = TRUE, angle = 20, size = 2.5)
```



5. Data-driven Hypotheses

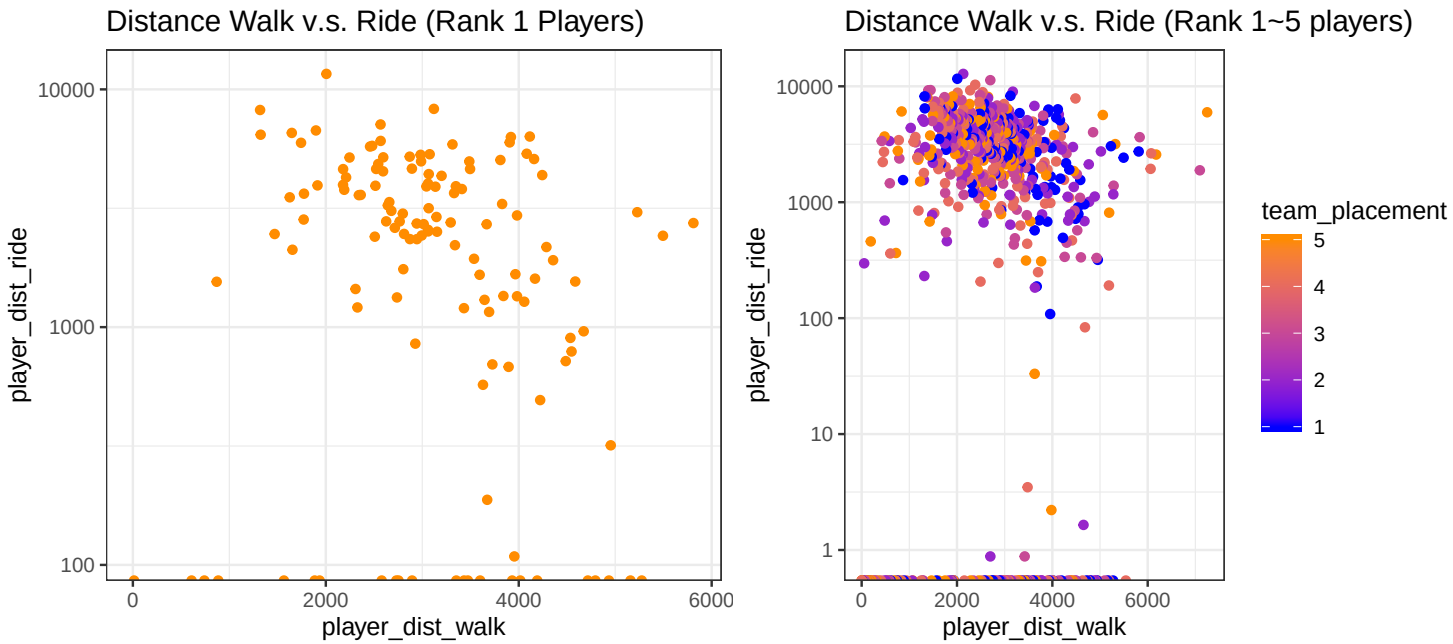
Hypothesis 1: Based on the PUBG game rules, it is difficult to use weapons and drive vehicles simultaneously. Besides, if players work with team, one is responsible for controlling the car, the other in charge of defense and attack, it will increase risks to be spot and killed together. Consequently, we can find the bar chart below, there are almost more than 1000 player who chose to not using vehicles. Next part I will filter outstanding players and discuss whether those players who didn't drive tending to win or not.

```
game2 %>%
  group_by(mobility) %>%
  count() %>%
  ggplot(mapping = aes(x = mobility, y = n)) +
  geom_col(fill = "dark orange") +
  ggtitle("Frequency of Vehicle use")
```



I found a similar dense part on the distance walk and ride for rank 1 players and rank 1~5 players. There are more players who gathered at the similar points by 1500~3500 walk with 3000~9000 ride.

```
dist_rank1 <- game2 %>%
  filter(team_placement == 1) %>%
  ggplot(mapping = aes(x = player_dist_walk, y = player_dist_ride)) +
  geom_point(color = "dark orange") +
  scale_y_log10() +
  ggtitle("Distance Walk v.s. Ride (Rank 1 Players)")

game2$team_placement <- as.integer(game2$team_placement)

dist_rank1to5 <- game2 %>%
  filter((team_placement <= 5)) %>%
  select(team_placement, player_dist_ride, player_dist_walk) %>%
  arrange(desc(player_dist_walk, team_placement)) %>%
  ggplot(mapping = aes(x = player_dist_walk, y = player_dist_ride,
                       color = team_placement)) +
  geom_point() +
  scale_y_log10() +
  scale_color_continuous(low = "blue", high = "dark orange") +
  ggtitle("Distance Walk v.s. Ride (Rank 1~5 players)")
```

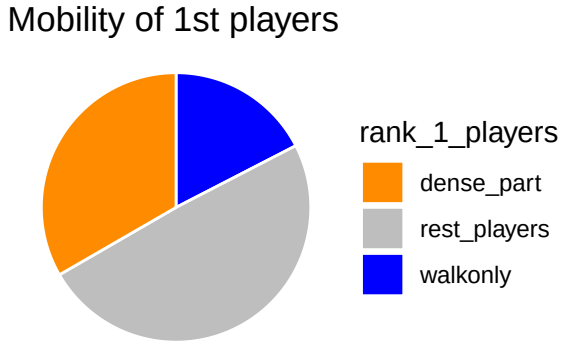


What is the proportion of walk only and walk & ride in the dense part? The only way to stand out from the game

is to be the last one alive in a match, thus filtered rank 1 player only.

```
# filter rank one players, assign as rank1_team
rank1_team <- game2 %>%
  filter(team_placement == 1) %>%
  select(team_placement, mobility, player_dist_walk, player_dist_ride)
# all rank 1 players = 132
all_players <- count(rank1_team) %>%
               rename(all_players = n)
# 1st player who walked only = 23
walkonly <- rank1_team %>%
  filter(mobility == "walk only") %>%
  count() %>%
  rename(walkonly = n)
# filter the density part from upper scatter plot = 37
dense_part <- rank1_team %>%
  filter(player_dist_ride >= 3000 & player_dist_ride <= 9000) %>%
  filter(player_dist_walk >= 1500 & player_dist_walk <= 3500) %>%
  count() %>%
  rename(dense_part = n)
# conbine information, get 1st player with different choice for moving
transports_table <- cbind(all_players, walkonly, dense_part) %>%
  gather("all_players", "walkonly", "dense_part",
         key = "rank_1_players", value = "amount")
# get the rest of players apart from those two group
transports_table[nrow(transports_table) + 1, ] =
  list("rest_players", transports_table[1, 2] -
                       transports_table[2, 2] - transports_table[3, 2])
```

```
# mapping them into a pie chart
transports_table %>% slice(2:4) %>%
  ggplot(aes(x = "", y = amount, fill = rank_1_players)) +
  geom_bar(width = 1, stat = "identity", color = "white") +
  coord_polar("y", start = 0) +
  scale_fill_manual(values = c("dark orange", "grey", "blue")) +
  theme_void() +
  ggtitle("Mobility of 1st players")
```



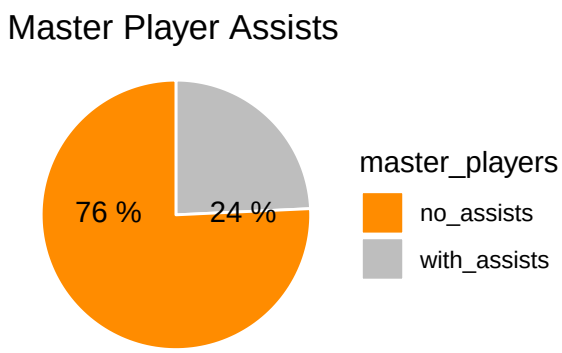
According to the pie chart above, people who walk and ride accounted for 33.3% (orange part), which is more than players who walked only (blue part: 17.4%), the grey area 49.3% is the rest of rank one people. In conclusion, based on the chart and analysis, the figure of walk only is considerable, but it might be advisable to walk 1.5~3.5k and ride vehicles about 3k~9k. (Note: I interpret the percentage in words as I failed to put the percentage on the pie chart.)

Hypothesis 2

Player Assists: I analyzed player assists independently since player assist is slight significant to other player statistics. As the pie chart below, there are about three quarter of players who gave no assist to their teammate and about 24% of players who gave equal or more than one assist. I think the result shows the game tactics, such as one of the members whose role might be scrambling medical packages and helping others all the time.

```
# master players who did/didn't assist others = 792/2667 (duo or squad)
game2 %>%
  filter(game_mode == "duo" | game_mode == "squad") %>%
  filter(player_assists >= 1 & team_level == "Master") %>%
  count() %>%
  rename(with_assists = n)->
  master_asst
game2 %>%
  filter(game_mode == "duo" | game_mode == "squad") %>%
  filter(player_assists == 0 & team_level == "Master") %>%
  count() %>%
  rename(no_assists = n)->
  master_no_asst
```

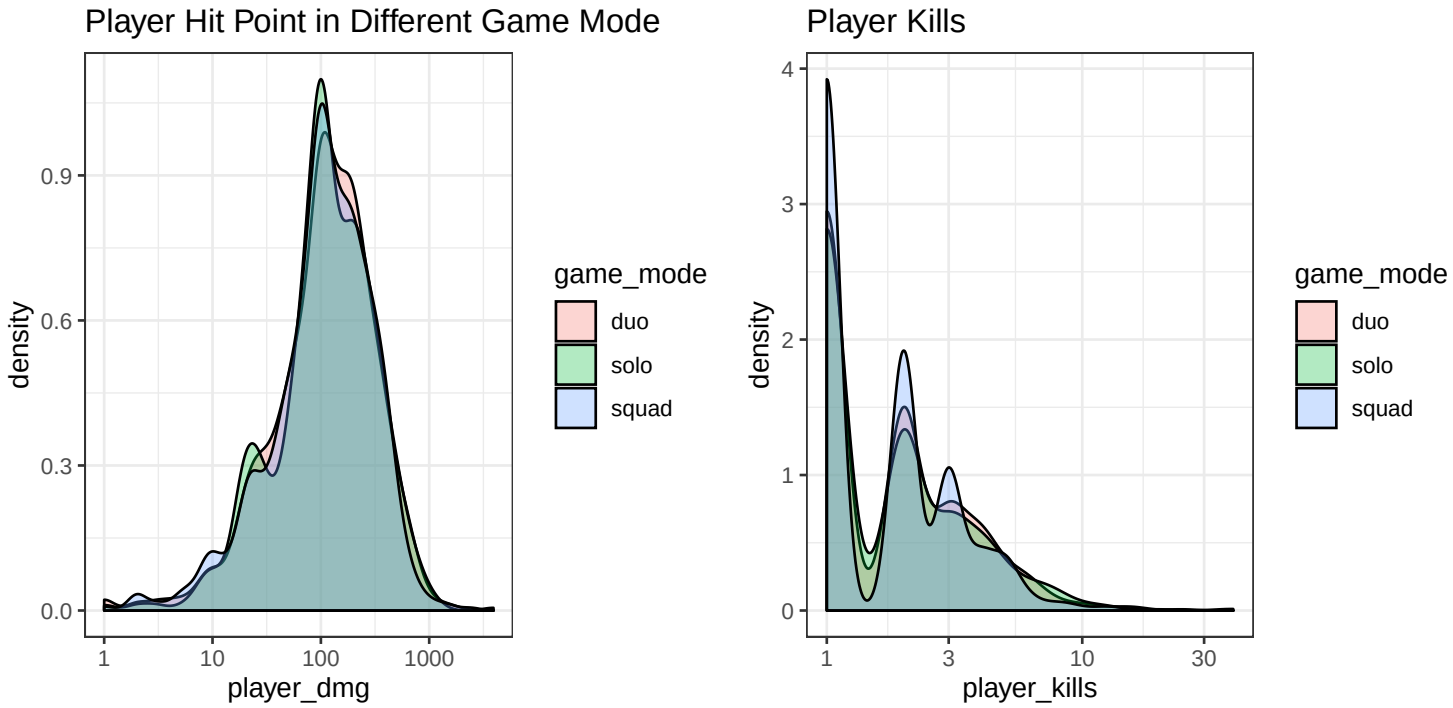
```
master_asst <- cbind(master_asst, master_no_asst) %>%
  gather("with_assists", "no_assists",
         key = "master_players", value = "amount") %>%
  ggplot(aes(x = "", y = amount, fill = master_players)) +
  geom_bar(width = 1, stat = "identity", color = "white") +
  coord_polar("y", start = 0) +
  geom_text(aes(x = 1.0, label = paste(round(amount/sum(amount)*100), "%"))) +
  scale_fill_manual(values = c("dark orange", "grey", "blue")) +
  theme_void() +
  ggtitle("Master Player Assists")
master_asst
```



Player Damages and Kills: For damage points, the pattern looks similar, but we can find the peak at 100 damage, players in solo mode performed the best than duo and squad. It might because players who want survive should defend or attack others in case of being killed. However, players in duo or squad mode normally assign a certain duty for each one. Their prerequisite is to win chicken dinner rather than earning as many points as they can. Conversely, with teammate's assistance, a player whose role is attacking others in squad mode would be able to kill more enemies than duo or solo mode.

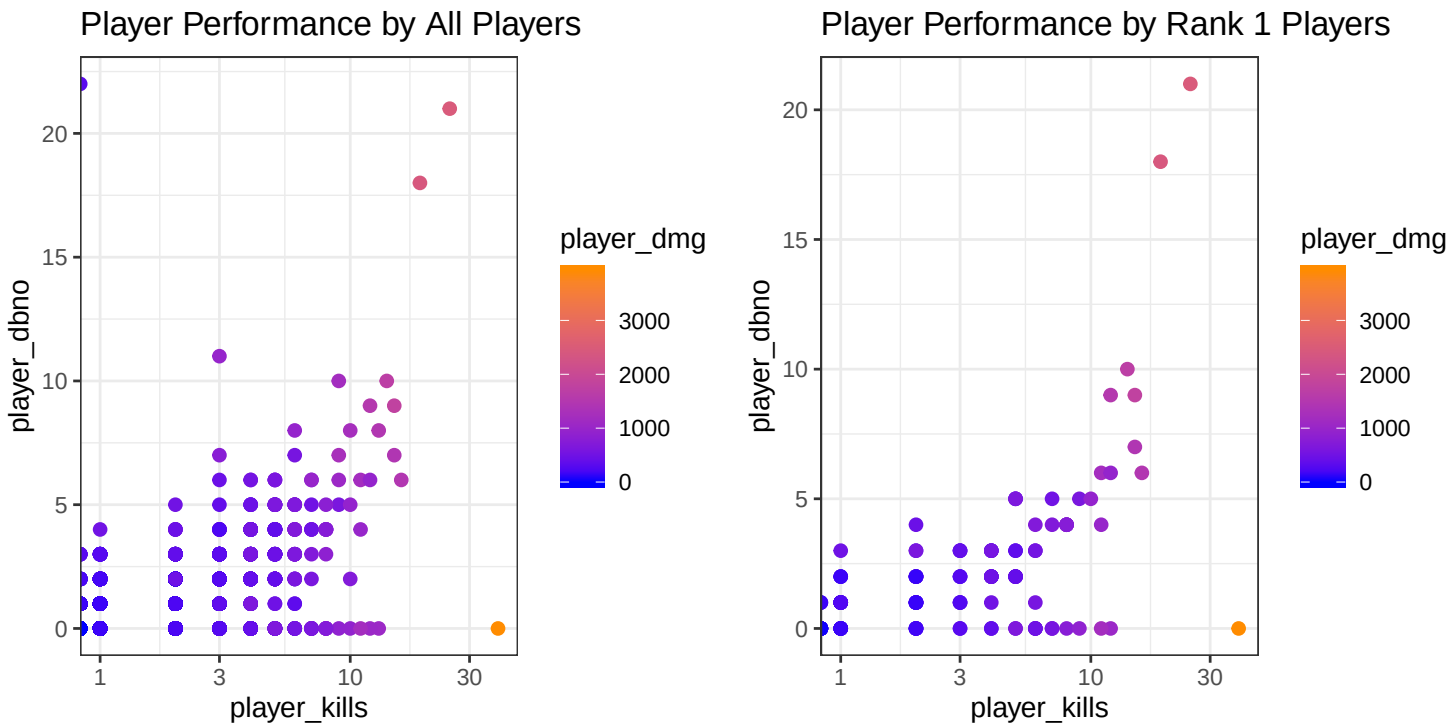
```
density_dmg <- ggplot(data = game2,
                      mapping = aes(x = player_dmg, fill = game_mode)) +
  scale_x_log10() +
  geom_density(alpha = 0.3) +
  ggtitle("Player Hit Point in Different Game Mode")
density_kills <- ggplot(data = game2,
```

```
                    mapping = aes(x = player_kills, fill = game_mode)) +
  scale_x_log10() +
  geom_density(alpha = 0.3) +
  ggtitle("Player Kills")
```



Overall Player Statistics: To conclude the play statistics, I put three main causes (player kills, knockdown points, and hit points) together but filter to two groups: all players and rank one players only. We can see the pattern of these two plots are similar. In other words, win or lose is not necessary related to player statistics.

```
player_stats_all <- game2 %>%
  ggplot(aes(x = player_kills, y = player_dbno, color = player_dmg)) +
  geom_point(size = 2) +
  scale_x_log10() +
  scale_color_continuous(low = "blue", high = "dark orange") +
  ggtitle("Player Performance by All Players")
player_stats_1st <- game2 %>%
  filter(team_placement == 1) %>%
  ggplot(aes(x = player_kills, y = player_dbno, color = player_dmg)) +
  geom_point(size = 2) +
  scale_x_log10() +
  scale_color_continuous(low = "blue", high = "dark orange") +
  ggtitle("Player Performance by Rank 1 Players")
```

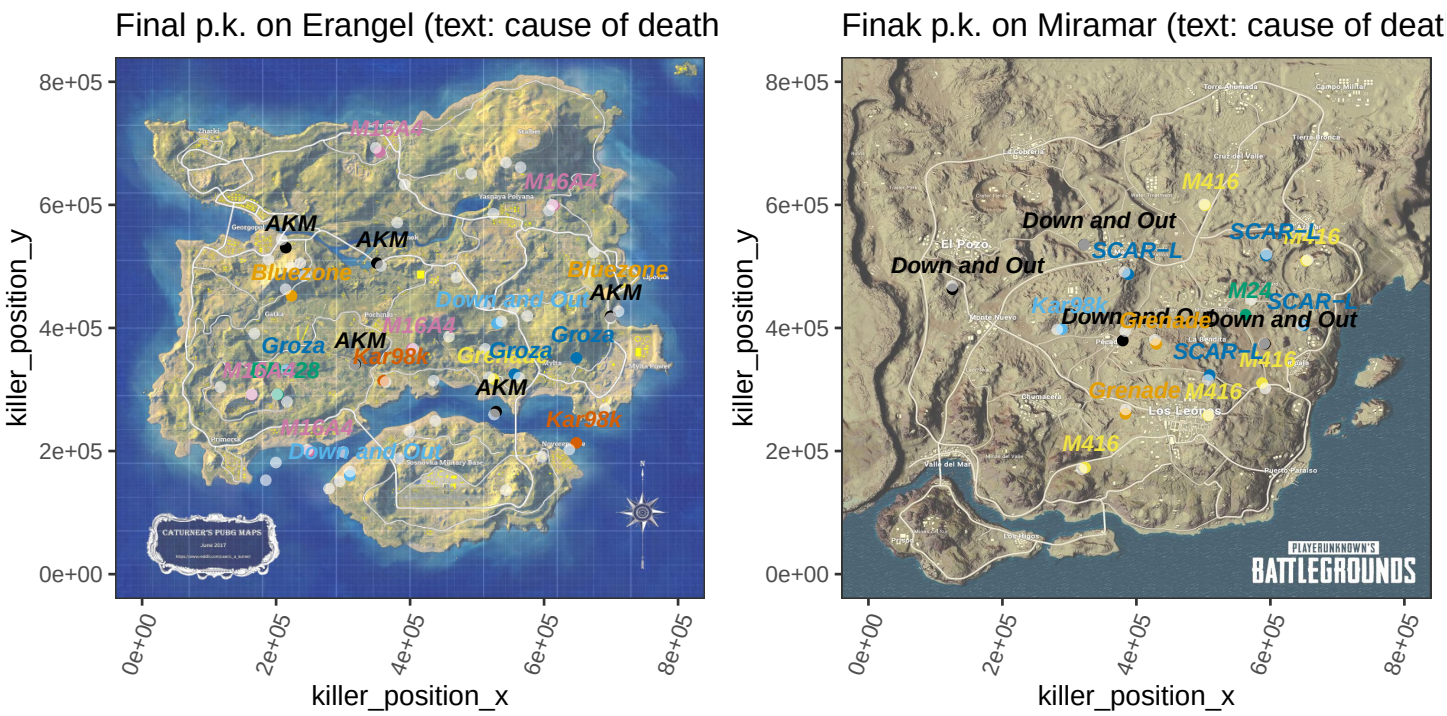


Hypothesis 3: To see the killer and the victim's position on the map, I filter killers whose placement is one and victims whose placement is two. Then add the cause of death on the map. Afterwards We may be able to find out which place accumulate more fights.

```
final_pk_ERAN <- death1 %>%
  filter(killer_placement == 1 & victim_placement == 2) %>%
  filter(map == "ERANGEL")

final_pk_MIRA <- death1 %>%
  filter(killer_placement == 1 & victim_placement == 2) %>%
  filter(map == "MIRAMAR")
```

```
final_eran <- ggplot(final_pk_ERAN, mapping = aes(x = killer_position_x,
                                    y = killer_position_y, color = killed_by)) +
  annotation_raster(erangel, ymin = -Inf, ymax = Inf, xmin = -Inf, xmax = Inf) +
  geom_point() + geom_point(aes(x = victim_position_x, y = victim_position_y),
                                color = "white", alpha = 0.6) +
  scale_x_continuous(limits = c(0, 800000)) +
  scale_y_continuous(limits = c(0, 800000)) +
  theme(axis.text.x = element_text(angle = 70, hjust = 1),
        legend.position = "none") +
  geom_text(aes(label = killed_by, fontface = "bold.italic"),
            hjust = 0.4, vjust = -1, size = 3) +
  scale_color_colorblind() +
  ggtitle("Final p.k. on Erangel (text: cause of death)")

final_mira <- ggplot(final_pk_MIRA, mapping = aes(x = killer_position_x,
                                    y = killer_position_y, color = killed_by)) +
  annotation_raster(miramar, ymin = -Inf, ymax = Inf, xmin = -Inf, xmax = Inf) +
  geom_point() + geom_point(aes(x = victim_position_x, y = victim_position_y),
                                color = "white", alpha = 0.6) +
  scale_x_continuous(limits = c(0, 800000)) +
  scale_y_continuous(limits = c(0, 800000)) +
```

```
  theme(axis.text.x = element_text(angle = 70, hjust = 1),
        legend.position = "none") +
  geom_text(aes(label = killed_by, fontface = "bold.italic"),
            hjust = 0.4, vjust = -1, size = 3) +
  scale_color_colorblind() +
  ggtitle("Finak p.k. on Miramar (text: cause of death)")
```



According to the plots, the distance between killer and victim on Erangel is longer than on Miramar, and the amount of players are more than on Miramar. The possible reason might be it is easier to manipulate vehicles and move on the island than on the desert. Furthermore, it seems like M16A4 and AKM on Erangel or M416 and SCAR-L on Miramar are popular weapons. Lastly, the distributions on Erangel are scattered but always nearby the buildings or villages, while killers are more likely to gather around La Bendita and Los Leones on desert.

6. Discussion

As reported by the data analysis above, we can summarize at least three main directions for players applying to their team tactics. First is vehicle using, it is advisable to walk for about 1.5k to 3.5k as well as ride vehicles for 3k to 9k. However, if you are not good at manipulate vehicles, you may walk only since the survival rate of walking only is pretty good too. Secondly, the player statistics is unnecessary, your placement won't be affected by kills or hit points that much. Moreover, the kills and hit points are very important to team strategies, if you want to achieve PUBG, you should have a good rapport with team. Lastly, in regard with finding a good hiding place. It will give you more strengths if you can be familiar with suburban area on Erangel island, be skilled in fighting on desert terrain and avoid staying around La Bendita and Los Leones on Miramar island. Nevertheless, in addition to choose a safer spot, remember to watch out the continuously shrinking blue zone. For the weapon choosing, there are more players who achieve on Erangel with M16A4 and AKM, besides, M416 and SCAR-L is recommended to use on Miramar. I believe if you practice PUBG with these tips, you may be able to enjoy your chicken dinner everyday!

7. Appendix

Load original datasets and subset data: My original data is too huge to work the research smoothly, therefore I determined my sample size through the sample size calculator, as my two data which has 11,640,855 and 11,993,485

observations separately, my sample size should be at least 1,068 in terms of 95% confidence level and 3% of margin of error. In summary, I am going to take a random sample size 5,000 from each dataset and then export the data. My research of this project will be based on the subset data.

```
game_ori <- read_csv("./data/agg_match_stats_4.csv")
# game dataset contains 11993485 objects and 15 variables
death_ori <- read_csv("./data/kill_match_stats_final_4.csv")
# death dataset contains 11640855 objects and 12 variables
```

```
# sample size 5000
sub_game <- game_ori[sample(1:nrow(game), 5000, replace=FALSE), ]
sub_death <- death_ori[sample(1:nrow(death), 5000, replace=FALSE), ]
# export the subset data
write_csv(sub_game, path = "sub_game_1.csv")
write_csv(sub_death, path = "sub_death_1.csv")
```

Bug Reporting: Are there any parsing fails in my datasets?

```
problems(game)
problems(death)
# both of them return 0 rows, it seems like there is no parsing error
```

Missing Values: There are some missing values in six columns from the death dataset: killer_name, killer_placement, killer_position_x, killer_position_y, map and victim_placement.

```
# find missing values for each column
death1 %>%
  summarize_all(funs(sum(is.na(.))))
```

As the table above, there are four columns (killer_name, killer_placement, killer_position_x, and killer_position_y) without indices in the same 339 rows. We can find that all of them are related to killer's information. Those victims were dead by shield deaths, suicide or accidental death rather than killed by enemies. Therefore I would not remove those NA values since they are reasonable.

```
# filter the data that without killer's information
death1_nokill <- death1 %>%
  filter(is.na(killer_name)) %>%
  filter(is.na(killer_placement)) %>%
  filter(is.na(killer_position_x)) %>%
  filter(is.na(killer_position_y))
# returns 339 rows which gathering in the same rows
```

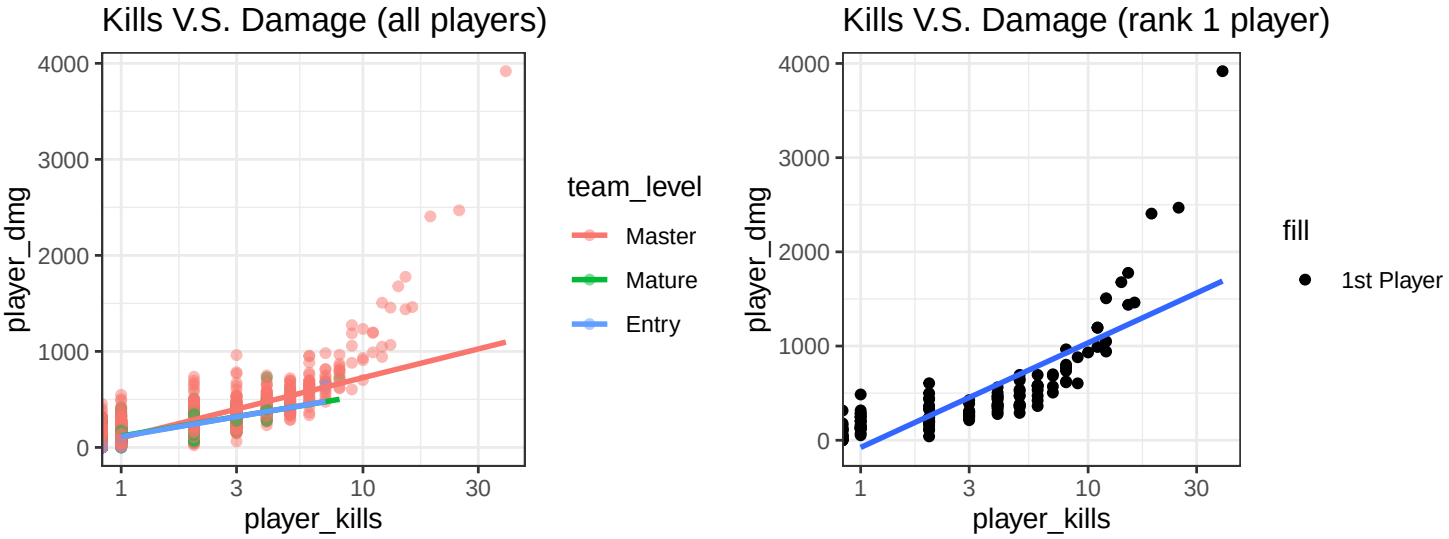
Additionally, there are 47 missing values in map and 102 missing values in victim_placement, but I literally have no idea with them. I might remove them when I need to use these two variables later on.

```
# view NA in map
death1 %>%
  filter(is.na(map)) %>%
  select(map, everything())
# view NA in victim_placement
death1 %>%
  filter(is.na(victim_placement)) %>%
  select(victim_placement, everything())
```

Hypothesis 2: hit points v.s. player kills

```
# kills versus damage (all players)
all_killdmg <- ggplot(data = game2,
```

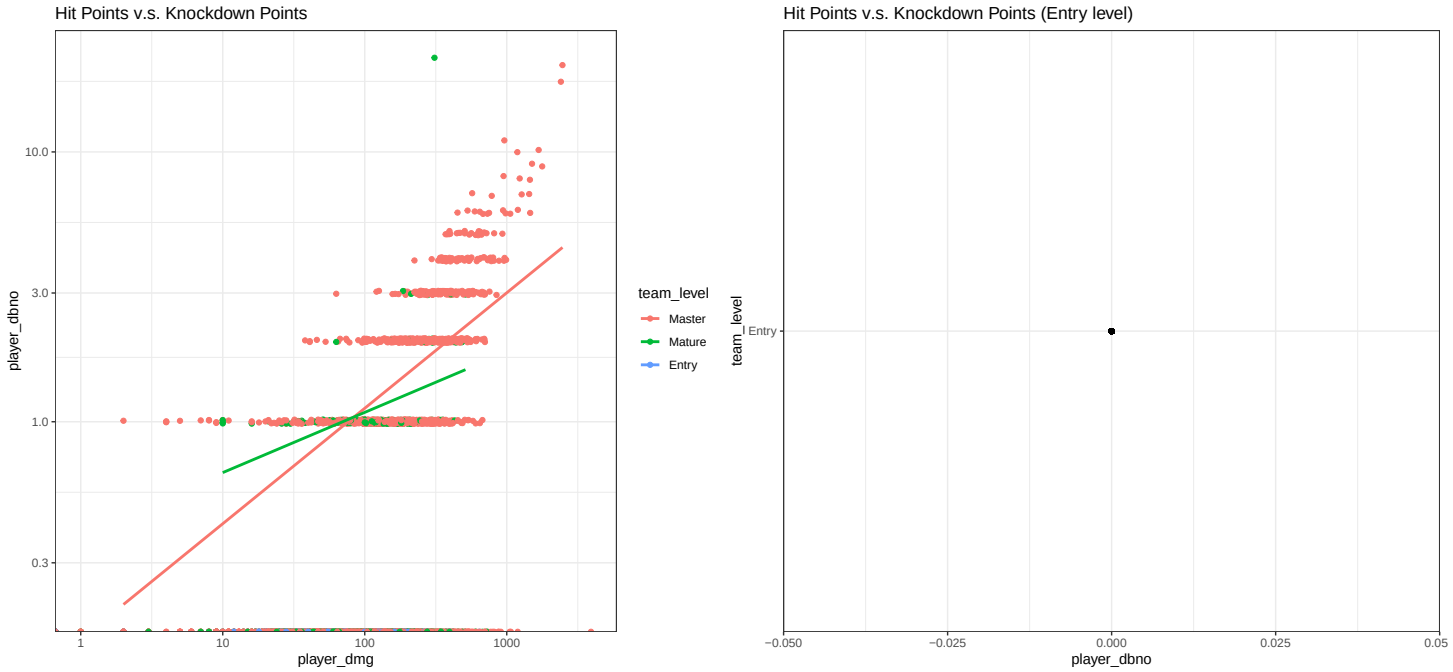
```
                    mapping = aes(x = player_kills, y = player_dmg, color = team_level)) +
  geom_point(alpha = 0.5) +
  scale_x_log10() +
  geom_smooth(se = FALSE, method = lm) +
  ggtitle("Kills V.S. Damage (all players)")

# kills versus damage (rank 1)
rank1_killdmg <- game2 %>%
  filter(as.numeric(team_placement) <= 1) %>%
  ggplot(mapping = aes(x = player_kills, y = player_dmg)) +
  scale_x_log10() +
  geom_point(aes(fill = "1st Player")) +
  geom_smooth(aes(x = player_kills, y = player_dmg), se = FALSE, method = lm) +
  ggtitle("Kills V.S. Damage (rank 1 player)")

ggarrange(all_killdmg, rank1_killdmg)
```



Hypothesis 2: hit points v.s. knockdown points

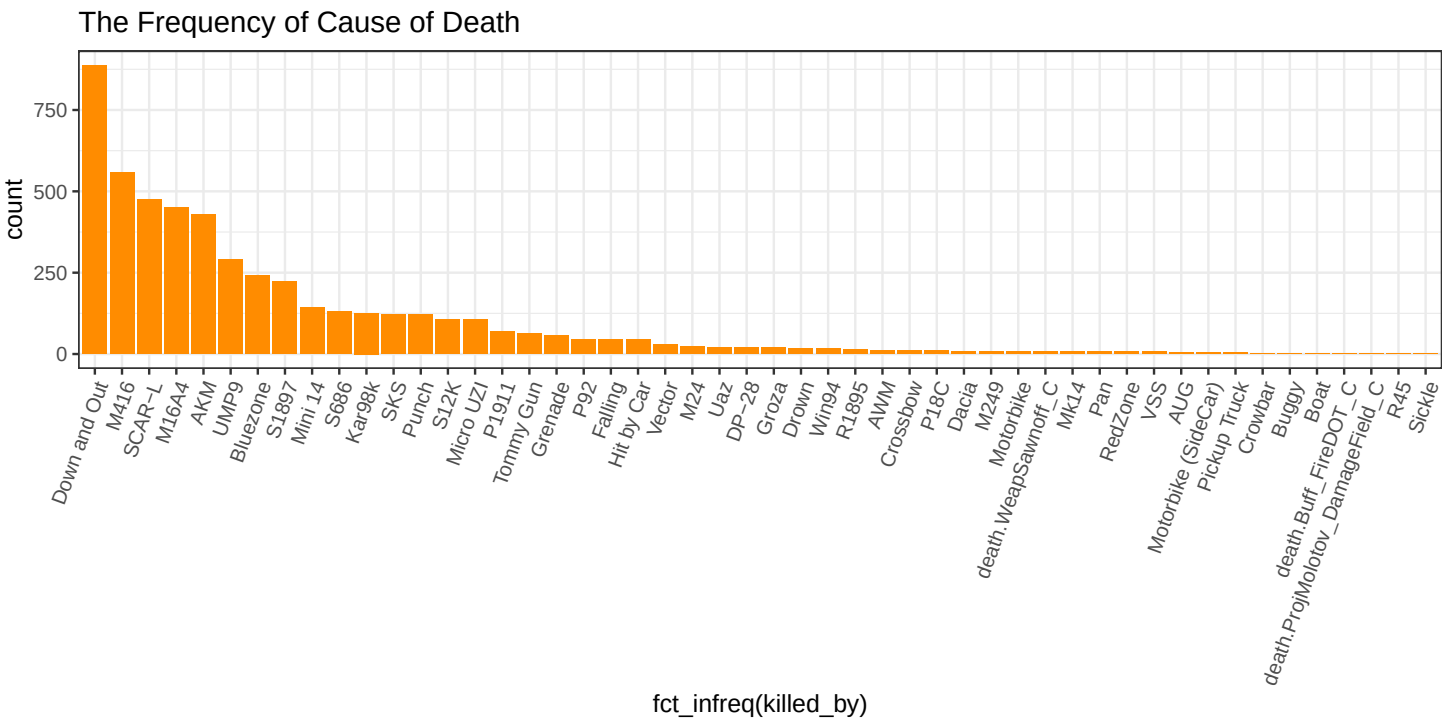
```
# The figure has positive correlation
# Mature level had one outlier but nomarally scored hit points lower than 500
# Master level performed best
all_dmgkd <- ggplot(data = game2, mapping = aes(x = player_dmg,
                                    y = player_dbno, color = team_level)) +
  geom_jitter() +
  scale_x_log10() +
  scale_y_log10() +
  geom_smooth(se = FALSE, method = lm) +
  ggtitle("Hit Points v.s. Knockdown Points")

# Entry level players did't score any player_dbno points
entry_dmgkd <- game2 %>%
  filter(team_level == "Entry") %>%
  ggplot(aes(x = player_dbno, y = team_level)) +
  geom_point() +
  ggtitle("Hit Points v.s. Knockdown Points (Entry level)")
```

```r
# arrange these two plots
ggarrange(all_dmgkd, entry_dmgkd)
```



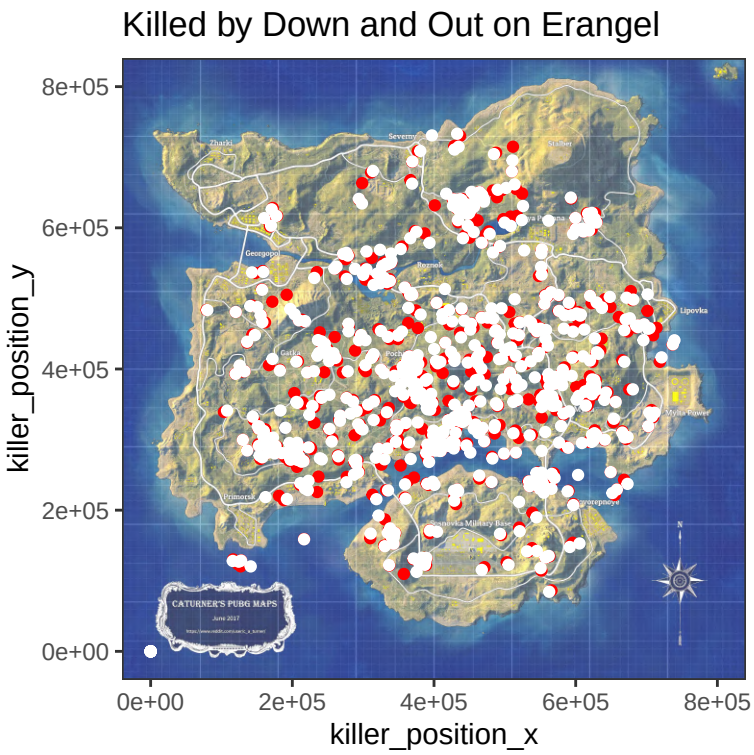
Hypothesis 3: plot the distribution map of killer and victim's position who were killed by down and out

```r
player_killedby <- death1 %>%
  group_by(killed_by) %>%
  count() %>%
  arrange(desc(n))
# killed by down and out accounted for the vest majority of death
# draw the frequency of cause of death with bar chart
ggplot(death1, mapping = aes(x = fct_infreq(killed_by))) +
  geom_bar(fill = "dark orange") +
  theme(axis.text.x = element_text(angle = 70, hjust = 1)) +
  ggtitle("The Frequency of Cause of Death")
```



It is clear to see there are some place such as the settlement of Pecado in the center gathered more victims, players who want to achieve the game should avoid closing those part too early, or it may increase the risk of being killed.

```
# filter kiiled by down and out in each map
killbydo_ERAN <- death1 %>%
  filter(killed_by == "Down and Out") %>%
  filter(map == "ERANGEL")
killbydo_MIRA <- death1 %>%
  filter(killed_by == "Down and Out") %>%
  filter(map == "MIRAMAR")

# draw a scatter plot and insert the map image at the back

### kdbydo_er
ggplot(killbydo_ERAN, aes(x = killer_position_x, y = killer_position_y)) +
  annotation_raster(erangel, ymin = -Inf,ymax= Inf,xmin = -Inf,xmax = Inf) +
  geom_point(colour="red") +
  geom_point(aes(x = victim_position_x, y= victim_position_y), colour="white") +
  scale_x_continuous(limits = c(0, 800000)) +
  scale_y_continuous(limits = c(0, 800000)) +
  ggtitle("Killed by Down and Out on Erangel")
```




```
###kdbydo_mir
ggplot(killbydo_MIRA, aes(x = killer_position_x, y = killer_position_y)) +
  annotation_raster(miramar, ymin = -Inf,ymax= Inf,xmin = -Inf,xmax = Inf) +
  geom_point(colour="red") +
  geom_point(aes(x = victim_position_x, y= victim_position_y), colour="white") +
  scale_x_continuous(limits = c(0, 800000)) +
  scale_y_continuous(limits = c(0, 800000)) +
  ggtitle("Killed by Down and Out on Miramar")
```