Performance of a Casual Osu! Player

Yanyu Yang yy8439

3/22/2021

Introduction

My project will be focused on Osu!, a free PC rhythm game by Dean "peppy" Herbert. The main goal is to click circles in time to songs where users can create their own "map" of patterns.[1] Although I mainly play the game in single player mode, there is a built in ranking system. Performance points, officially abbreviated as "pp", are awarded at the end of every map. Rankings are calculated based on each player's total pp.

At the time of data collection on March 17, 2021, I was ranked 535,723rd out of the entire player base with 884pp. As one can deduct from the title, I am a fairly casual player. The following sections will analyze my Osu! performance.

Dataset 1: My Osu! Profile[3]

```r
library(readxl)
# import dataset 1
slendermaniac <- read_excel("slendermaniac.xlsx")
head(slendermaniac)
```

```
## # A tibble: 6 x 4
##   song                           difficulty_name   accuracy    pp
##   <chr>                          <chr>                <dbl> <dbl>
## 1 Koto no Ha (TV Size)           Gu's Insane           96.9    54
## 2 &Z (TV size)                   Insane                97.0    47
## 3 unravel                        Hard                  91.7    45
## 4 BRAVE JEWEL (TV Size)          Hyper                 95.0    44
## 5 IGNITE (TV size ver.)          Hard                  99.1    44
## 6 Dancing stars on me! (TV Size) Chaozomi's Insane     87.8    44
```

The first data set was obtained straight from my Osu! profile (https://osu.ppy.sh/users/4655284). I entered the top 50 maps from the "Best Performance" section into an excel spreadsheet with the following variables:

- **song**: the title of the song used for the map
- **difficulty_name**: the difficulty category, usually "easy", "hard", "insane", or "expert". difficulties listed as [name]'s [difficulty] indicate a guest mapper.
- **accuracy**: hits can get a score of 300, 100, 50, or miss. accuracy is calculated as (100*# of 300s) + (67.77*# of 100s) + (33.33* # of 50s) + (0* # of misses)
- **pp**: performance points awarded

Dataset 2: Pps-by-grumd[2]

```r
# import dataset 2
map_data <- read_excel("map_data.xlsx")
head(map_data)
```

```
## # A tibble: 6 x 7
##   song               difficulty_name    pp length   bpm difficulty overweightness
##   <chr>              <chr>           <dbl>  <dbl> <dbl>      <dbl>          <dbl>
## 1 Koto no Ha (TV S~ Gu's Insane        52     88   145       3.91              0
## 2 &Z (TV size)       Insane             92     88   158       4.2             169
## 3 unravel            Hard               44    203   135       3.52              0
## 4 BRAVE JEWEL (TV ~ Hyper              72     89   192       3.98              3
## 5 IGNITE (TV size ~ Hard               52     87   171       3.43             12
## 6 Dancing stars on~ Chaozomi's Ins~    51     97   137       3.67              0
```

The second data set was collected from https://osu-pps.com/#/osu/maps, a fanmade site which scrapes and compiles official data for each osu map. I looked up the top 50 songs from my profile in pps by grumd and entered the results into a spreadsheet with the following variables:

- ***song***: same variable as in the first data set
- ***difficulty_name***: same variable as in the first data set
- ***pp***: average performance points awarded to the top 11,000 Osu! players. this more or represents the top pp ceiling, since even top players don't make perfect pp plays with 100% accuracy
- ***length***: the length of the map in seconds
- ***bpm*** beats per minute
- ***difficulty***: the current difficulty system using stars. can range from 0 stars to 7+ stars
- ***overweightness***: the number of top 11,000 players who have this map in their top 50 plays. higher overweightness means that map grants higher than average pp

In both data sets, each row represents a different map. I expect higher pp to be associated with maps that have higher difficulty, higher bpm, and are longer.

Tidy

Since the data sets were made by entering data into an excel sheet myself, I tried my best to make them as tidy as possible. However, there is one thing I would like to change. There is a "pp" variable in each dataset but they mean different things. I will rename the "pp" variable in map_data to "top_pp" in order to show the difference between variables.

```r
library(tidyverse)
# rename column "pp" to "top_pp"
map_data <- map_data %>% rename(top_pp = pp)
```

Now I have distinct variable names for the pp I gained and the average pp gained by the most competitive players.

Join

```
# join both datasets to form a new master dataset
osu <- inner_join(slendermaniac, map_data, by = c('song', 'difficulty_name'))
```

I left joined using both the "song" and "difficulty_name" variables. There were many maps with the same song name and were differentiated by different difficulty names, so certain maps were counted multiple times when I joined only by "song".

The joined dataset has 50 observations and 9 variables.

Summary Statistics

```
# make a new difficulty category variable
osu <- osu %>%
  mutate(difficulty_category = case_when(
    difficulty <= 1.99 ~ 'Easy',
    difficulty >= 2.0 & difficulty <= 2.69 ~ 'Normal',
    difficulty >= 2.7 & difficulty <= 3.99 ~ 'Hard',
    difficulty >= 4.0 & difficulty <= 5.29 ~ 'Insane',
    difficulty >= 5.3 & difficulty <= 6.49 ~ 'Expert',
    difficulty >= 6.5 ~ 'Expert +'
  ))
```

I added a new variable named "difficulty_category" and grouped the maps according to the difficulty category to star ratings conversion in the Osu! Knowledge Base, which functions as an official encyclopedia for Osu! terminology.[1] At a quick glance, I noticed that certain maps had different difficulty names from the difficulty category.

```
# filter by difficulty name
osu %>% filter(difficulty_name == 'Crazy Diamond')
```

```
## # A tibble: 1 x 10
##   song  difficulty_name accuracy    pp top_pp length   bpm difficulty
##   <chr> <chr>              <dbl> <dbl>  <dbl>  <dbl> <dbl>      <dbl>
## 1 I Wa~ Crazy Diamond       99.0    30     41     64   107       3.09
## # ... with 2 more variables: overweightness <dbl>, difficulty_category <chr>
```

An example is the map for Savage Garden's song "I Want You", which was used as an ending song for JoJo's Bizarre Adventure: Diamond Is Unbreakable. "Crazy Diamond" is the name of the main character's power, and Osu! mappers often enjoy adding these references to anime franchises when using custom difficulty names.

Another source of naming differences may be that in the early days, there were no strict cutoffs for difficulty names. Therefore, a map with a difficulty of 3.99 may be named "Hard" *or* "Insane" at the discretion of the mapper.

```
# ascending difficulty
osu %>% select(song, difficulty) %>%
  arrange(difficulty)
```

```
## # A tibble: 50 x 2
##    song                          difficulty
##    <chr>                              <dbl>
##   1 I Want You                         3.09
##   2 A Sweet Smile                      3.11
##   3 Bonetrousle                        3.18
##   4 &Z (TV size)                       3.24
##   5 All In All                         3.24
##   6 The Bad Thing                      3.34
##   7 MONSTER                            3.39
##   8 Donna Toki mo Zutto (TV Size)      3.42
##   9 IGNITE (TV size ver.)              3.43
## 10 MOBILE SUIT (W-REC MIX)             3.43
## # ... with 40 more rows
```

```
# descending difficulty
osu %>% select(song, difficulty) %>%
  arrange(desc(difficulty))
```

```
## # A tibble: 50 x 2
##    song                               difficulty
##    <chr>                                   <dbl>
##   1 Yuuki no Reason                         4.74
##   2 MOBILE SUIT (W-REC MIX)                 4.64
##   3 BRAVE JEWEL (TV Size)                   4.4
##   4 Paintings? Oh, yeah                     4.26
##   5 Bull's eye                              4.23
##   6 &Z (TV size)                            4.2
##   7 Sentimental Love (TV Size)              4.14
##   8 Sentimental Love (TV Size)              4.14
##   9 Shinkai Shoujo                          4.13
## 10 Shingeki st-hrn-egt20130629 Kyojin       4.08
## # ... with 40 more rows
```

Overall, the easiest map I played was "I Want You" with 3.09 stars and the hardest map I played was "Yuuki no Reason" with 4.74 stars.

```
# mean difficulty of overweight maps
overweight <- osu %>% filter(overweightness > 0) %>%
  summarize(mean(difficulty))
overweight
```

```
## # A tibble: 1 x 1
##   `mean(difficulty)`
##                <dbl>
## 1               3.91
```

```
# mean difficulty of no weight maps
no_weight <- osu %>% filter(overweightness == 0) %>%
  summarize(mean(difficulty))
no_weight
```

```
## # A tibble: 1 x 1
##   `mean(difficulty)`
##                <dbl>
## 1               3.57
```

```
# difference of means
overweight - no_weight
```

```
##   mean(difficulty)
## 1        0.3441883
```

When filtering maps with an overweightness above 0, which means that at least 1 player ranked in the top 11,000 have the map in their top plays, the average difficulty is 3.910 stars. Maps with an overweightness of 0 had an average difficulty of 3.565 stars. The difference in average difficulty between overweight and no weight maps is 0.344 stars.

Summary Statistics for Numeric Variables

```
# summary statistics of numeric variables
table1 <- osu %>%
  select(accuracy, pp, top_pp, length, bpm, difficulty, overweightness) %>%
  summarize_all(funs(mean, min, max, sd, var)) %>%
  pivot_longer(cols=(1:35))
```

The following table shows the mean, minimum, maximum, standard deviation, and variance for each numeric variable in the dataset.

Variable	Accuracy	pp	top pp	Length	BPM	Difficulty	Overweightness
Mean	94.671	34.84	59.06	111.18	163.84	3.717	9.58
Minimum	86.47	28	33	56	79	3.09	0
Maximum	100	54	121	242	222	4.74	169
Standard Deviation	3.204	5.776	17.944	44.73	28.167	0.374	31.021
Variance	10.268	33.361	322.017	2000.762	793.402	0.14	962.33

Summary Statistics by Difficulty Category

```
# summary statistics grouped by difficulty category
table2 <- osu %>% group_by(difficulty_category) %>%
  summarize(mean(difficulty),
           min(difficulty),
           max(difficulty),
           sd(difficulty),
           n_distinct(song),
           mean(accuracy),
           mean(pp),
           mean(top_pp),
           mean(length),
           mean(bpm))
```

The following table includes the summary statistics by difficulty category. The summary statistics I used include mean, minimum, max, standard deviation of difficulty, the number of maps, the number of distinct songs (not maps), the mean and standard deviation of length, and the mean and standard deviation of bpm.

Category	Hard	Insane
Mean Difficulty	3.572	4.296
Minimum Difficulty	3.09	4.08
Maximum Difficulty	3.99	4.74
Standard Deviation	0.238	0.227
Distinct Songs	39	9
Mean Accuracy	95.081	93.032
Mean pp	35.325	32.9
Mean top pp	54.125	78.8
Mean Length (s)	112	107.9
Mean BPM	163.6	164.8

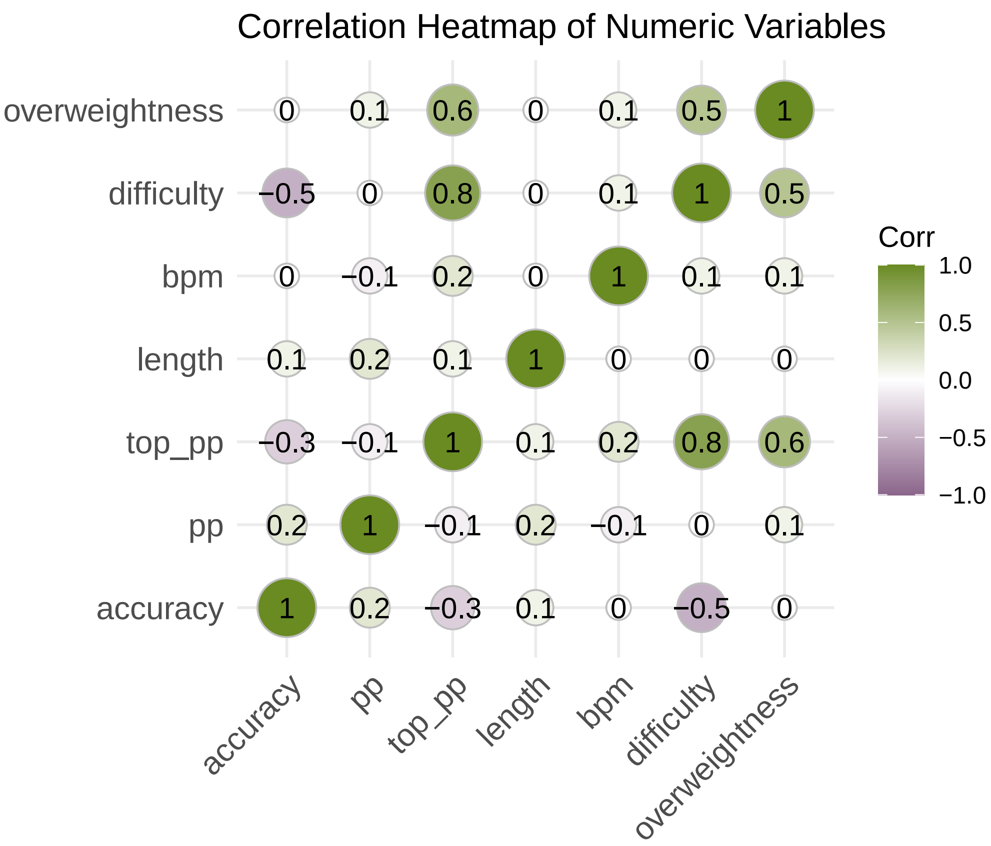
Visualizations

Correlation Heatmap

```
library(ggcorrplot)
# make a dataset with only numeric variables
numeric <- osu %>%
  select_if(is.numeric)

# correlation matrix
corr <- round(cor(numeric), 1)

# correlation heat map
ggcorrplot(corr, method='circle',
           lab = TRUE,
           title = 'Correlation Heatmap of Numeric Variables',
           colors = c('plum4', 'white', 'olivedrab4'))
```

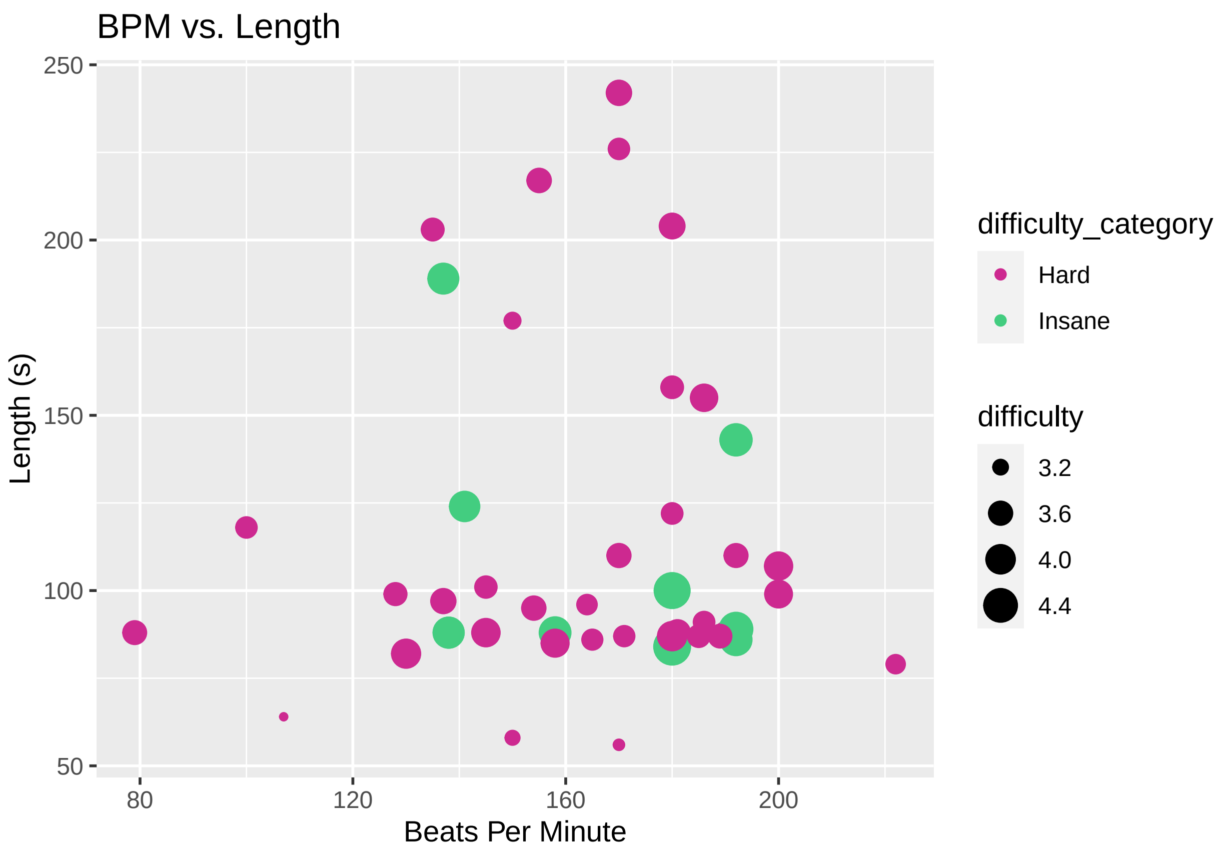


Considering Osu! is a circle clicking game, I made a correlation heatmap using circles. It seems that map difficulty ("difficulty") and average pp set by top 11,000 players ("top_pp") have the strongest positive correlation. On the other hand, map difficulty ("difficulty") and map accuracy set by me ("accuracy") have the strongest negative correlation.

This makes sense in context. Harder maps tend to give the highest pp, especially when played by the most competitive players. It is also pretty expected of me to have lowered accuracy when playing higher difficulty maps due to a lack of skill.

Scatterplot

```
library(ggplot2)
# scatterplot of bpm vs. length
osu %>% ggplot(aes(x = bpm, y = length)) +
  geom_point(aes(size = difficulty, col = difficulty_category)) +
  scale_color_manual(values = c('maroon3', 'seagreen3')) +
  labs(title = 'BPM vs. Length',
       x = 'Beats Per Minute',
       y = 'Length (s)')
```

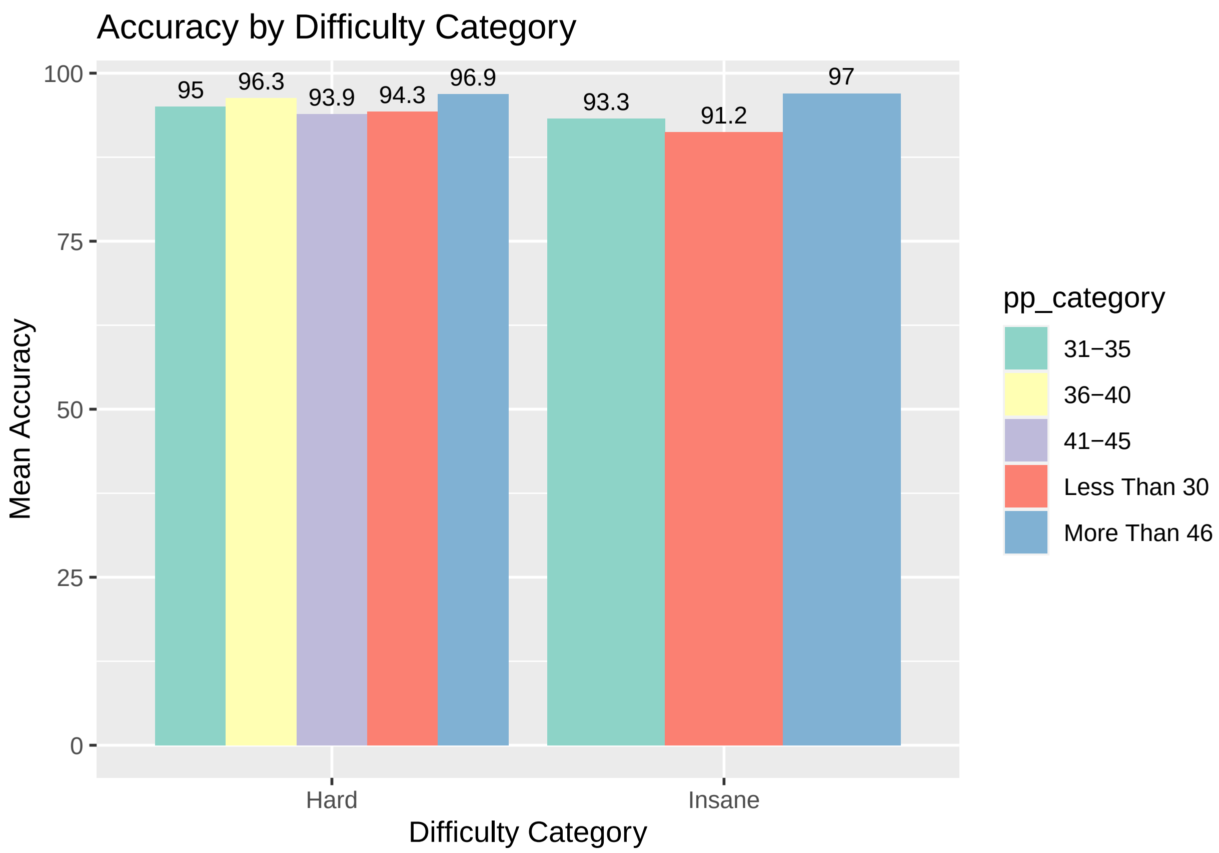


The scatter plot above plots the length in seconds of each map against beats per minute. The points are grouped based on difficulty category, with Maroon 3 representing "Hard" and Sea Green 3 representing "Insane" maps. The size of points also changes depending on difficulty (star rating), with larger points representing higher difficulty and lower points representing lower difficulty,

Bar Plot

```
# make new variable of pp categories
osu <- osu %>%
  mutate(pp_category = case_when(
    pp <= 30 ~ 'Less Than 30',
    pp >= 30 & pp <= 35 ~ '31-35',
    pp >= 36 & pp <= 40 ~ '36-40',
    pp >= 41 & pp <= 45 ~ '41-45',
    pp >= 46 ~ 'More Than 46'
  ))

# bat plot of difficulty category vs. mean accuracy
osu %>% ggplot(aes(x = difficulty_category, y = accuracy, fill = pp_category)) +
  geom_bar(position = 'dodge', stat = 'summary') +
  stat_summary(aes(label = round(..y..,0.5)), fun.y=mean, geom='text', size = 3,
             vjust = -0.5, position = position_dodge(width =0.9) ) +
  scale_fill_brewer(palette = 'Set3') +
  labs(title = 'Accuracy by Difficulty Category',
       x = 'Difficulty Category',
       y = 'Mean Accuracy')
```



The bar plot above displays the mean accuracy of not only difficulty category, but pp category within each difficulty as well, In order to achieve this, I mutated the “osu” data set to add a new categorical variable that separates pp awarded to me into 5 broad brackets: “less than 30”, “31-35”, “36-40”, “41-45”, and “More than 46.”

It is evident upon first glance that the Hard category also contains all 5 pp categories. However, the Insane category only contains maps awarded me ’Less Than 30“,”31-35“, and”More Than 46" pp. The maps I had the lowest average accuracy on were Insane maps that awarded me less than 30 pp, while the maps I had the highest average accuracy on were Insane maps that gave me more than 46 pp.

Principal Component Analysis

```
# standardize numeric variables
osu_scale <- numeric %>%
  scale(.) %>%
  as.data.frame()
```

First, I standardized my dataset containing only numeric variables from “osu”.

```
# run pca
osu_pca <- osu_scale %>%
  prcomp()

# variance
percent <- 100*(osu_pca$sdev^2/sum(osu_pca$sdev^2))
percent
```

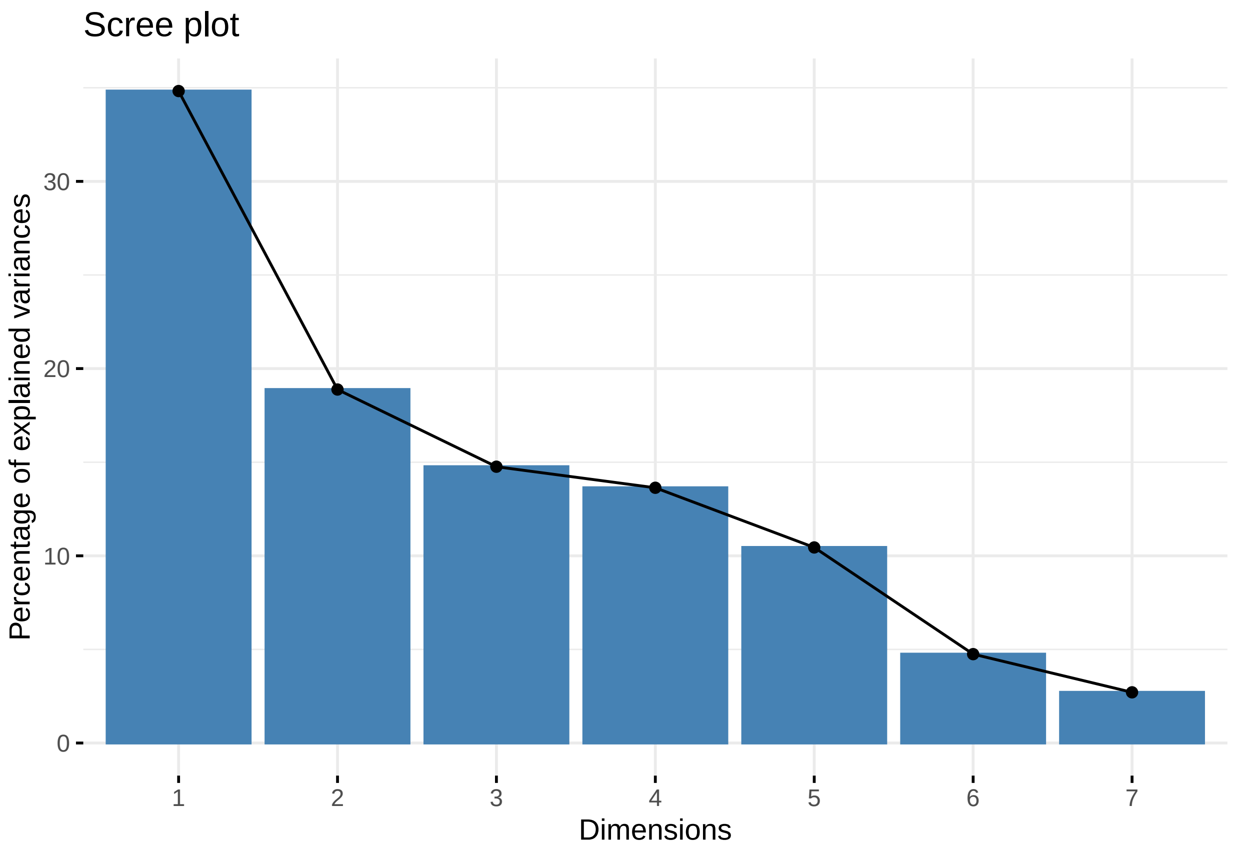
```
## [1] 34.824037 18.881331 14.758784 13.633786 10.446743  4.747483  2.707836
```

```
percent[1] + percent[2]
```

```
## [1] 53.70537
```

53.71% of variance can be explained by the first two principal components.

```
library(factoextra)
# scree plot
fviz_screeplot(osu_pca)
```

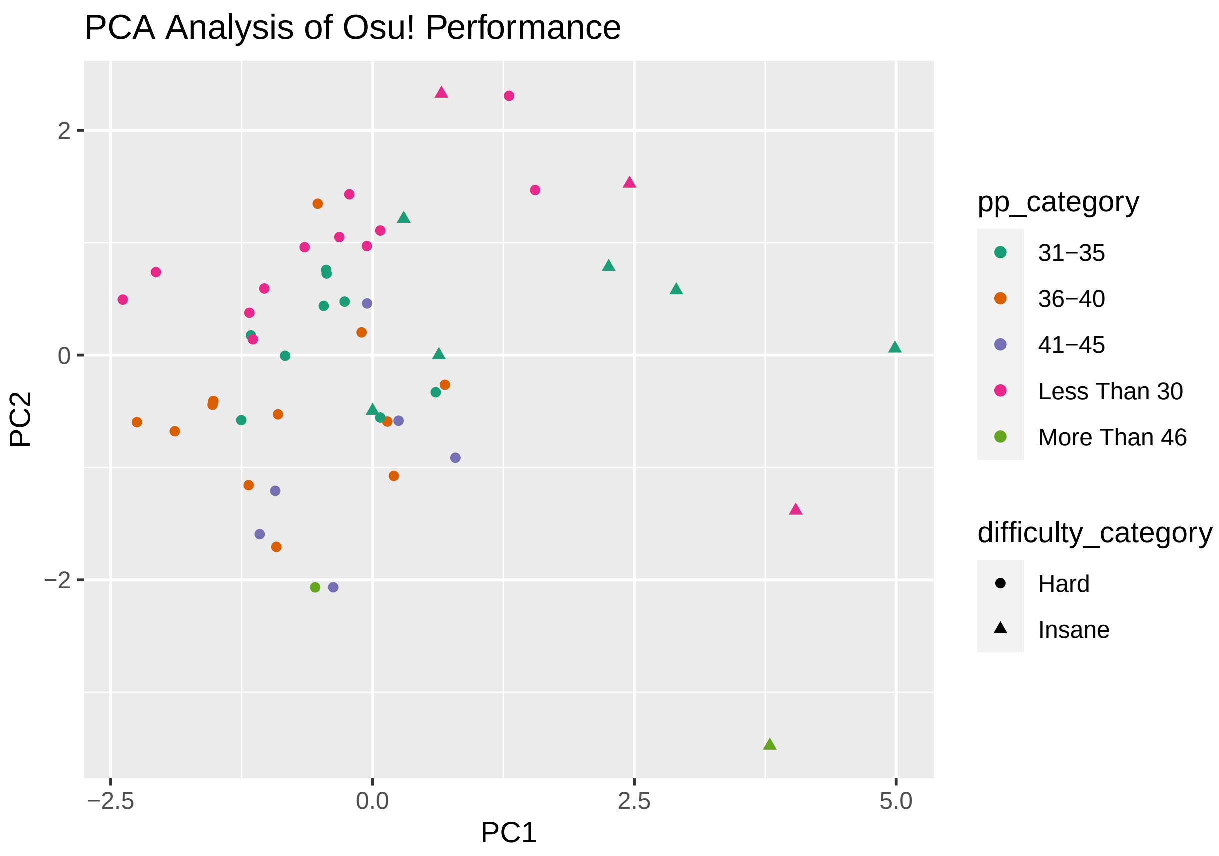


The scree plot has an "elbow" at roughly PC2. Using this data as well as the cumulative explained variance found in the previous section, I will keep the first two principal components.

```
# matrix of pca data
x <- eigen(cov(osu_scale))$vectors[,1:2]
# save as data frame
PC1 <- as.vector(as.matrix(osu_scale) %*% x[,1])
PC2 <- as.vector(as.matrix(osu_scale) %*% x[,2])
# add to original data set
osu_pca2 <- osu %>%
  mutate(PC1 = PC1, PC2 = PC2)
```

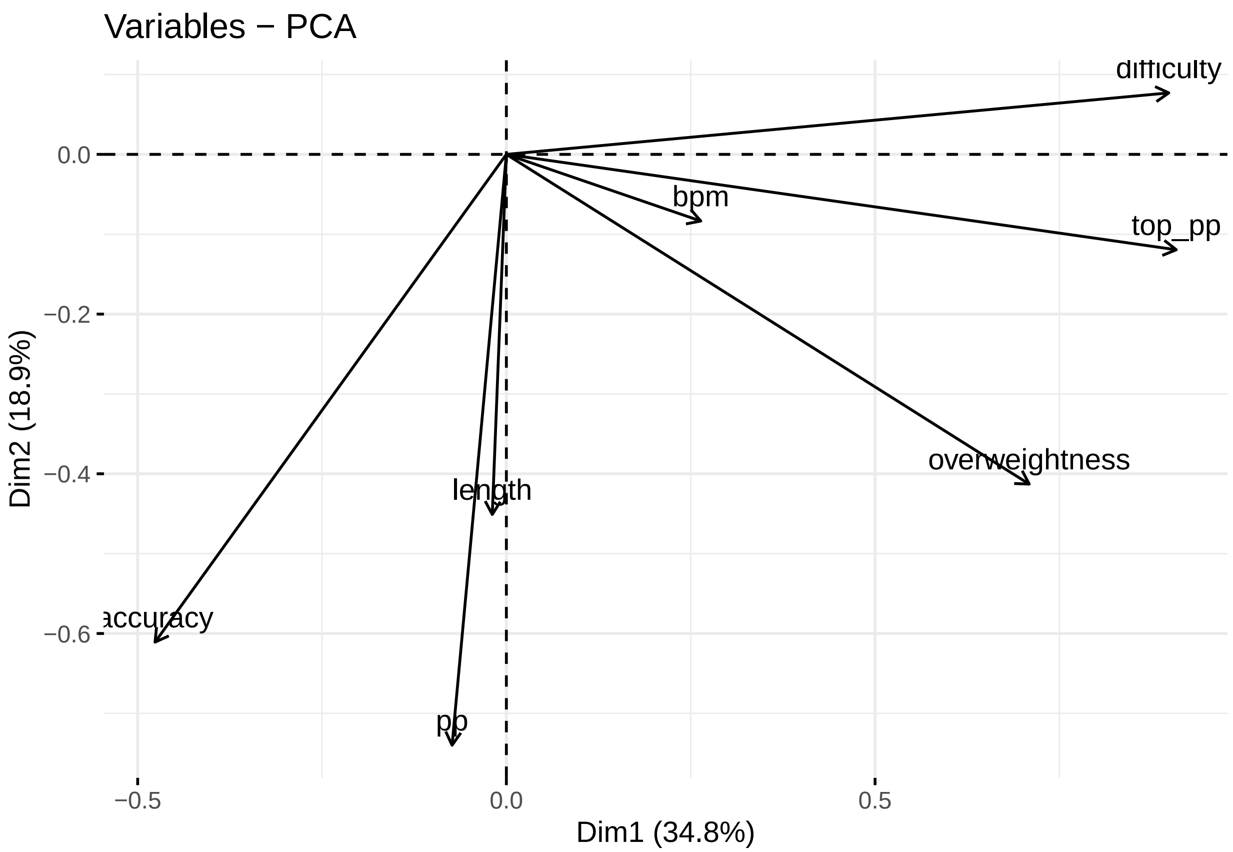
I added the first two principal components into the original dataset.

```
# plot of pca
ggplot(osu_pca2, aes(x = PC1, y = PC2)) +
  geom_point(aes(colour = pp_category, shape = difficulty_category)) +
  scale_color_brewer(palette = 'Dark2') +
  ggtitle('PCA Analysis of Osu! Performance')
```



Rotation Matrix

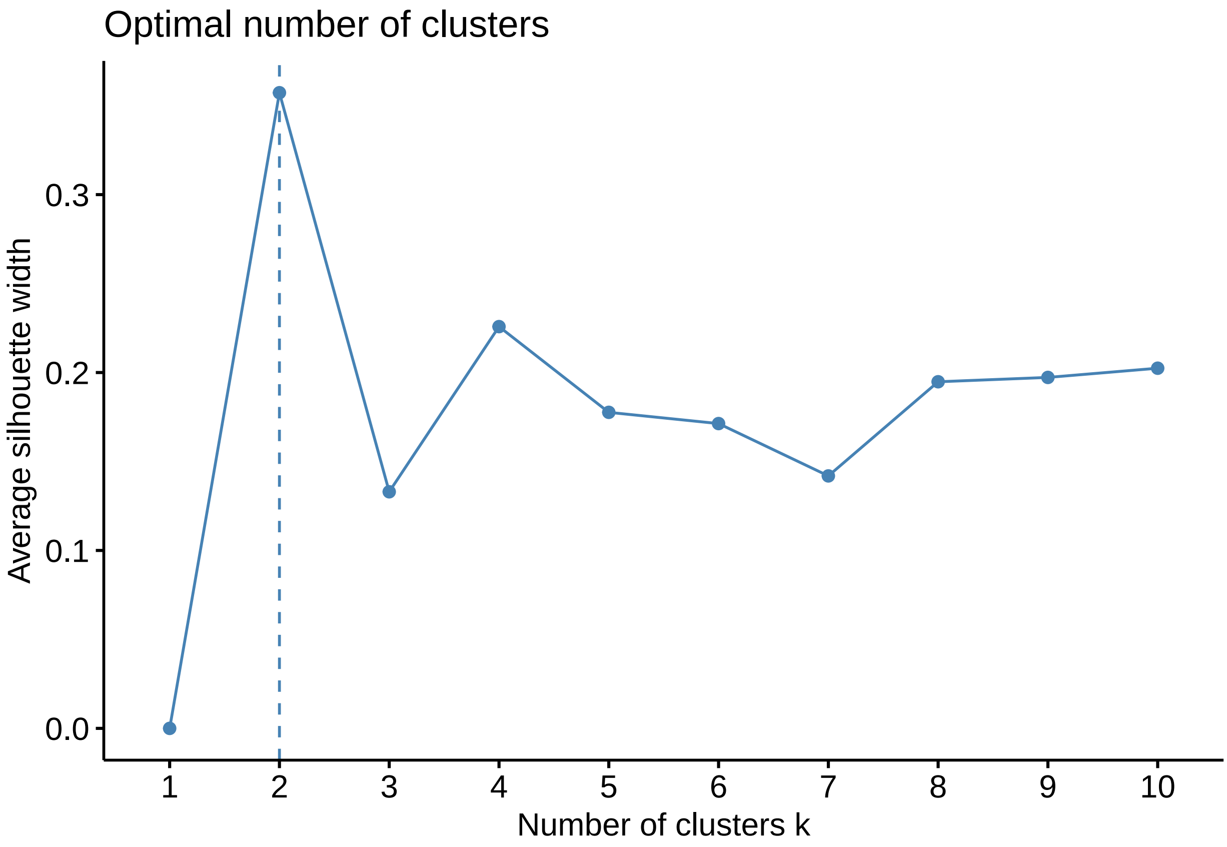
```
# rotation matrix of pca
fviz_pca_var(osu_pca, col.var = "black")
```



The Rotational Matrix shows that higher accuracy, pp, and length contribute to lower PC1 and PC2. Higher overweightness, top pp, and bpm contribute to higher PC1 but lower PC2. Higher difficulty contributes to both higher PC1 and PC2.

PAM Clustering

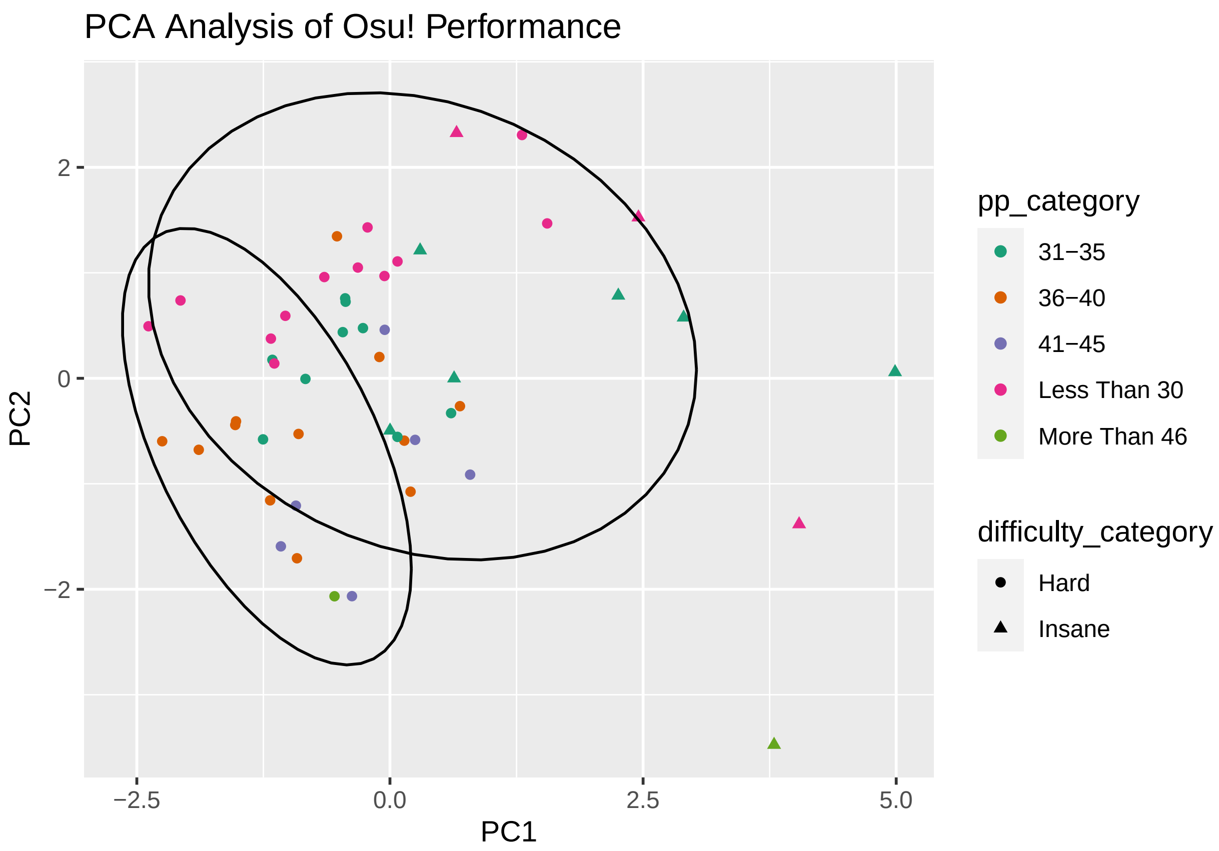
```
library(cluster)
# find optimal number of clusters
fviz_nbclust(osu_scale, FUNcluster = pam, method = "s")
```



```
# run PAM algorithm
osu_pam1 <- osu_pca2 %>%
  select(PC1, PC2) %>%
  pam(2)
# add cluster variable to original data set
osu_pam2 <- cbind(osu_pca2, cluster = osu_pam1$clustering)
```

According to this graph, the optimal number of clusters I should use is 2. I added the cluster variable to the original data set.

```
# plot of pam clusters
ggplot(osu_pam2, aes(x = PC1, y = PC2)) +
  geom_point(aes(colour = pp_category, shape = difficulty_category)) +
  scale_color_brewer(palette = 'Dark2') +
  stat_ellipse(aes(group = cluster)) +
  ggtitle('PCA Analysis of Osu! Performance')
```



Cluster 1 has a lower value for both PC1 and PC2. According to the rotational matrix above, this is likely contributed by higher accuracy, length, and pp. Cluster 2 has a higher value for PC1 and an equal or higher value for PC2 compared to Cluster 1. This means that Cluster 2 likely has higher difficulty, overweightness, top pp, and bpm.

In context, Cluster 1 represents the maps that I did better on. This explains the higher accuracy and pp awarded to me, as well as the fact that most of the maps are in the Hard category, It also means that I don't struggle as much with longer maps.

Cluster 2 represents harder maps that I did not do as well on. This explains the higher average pp of top 11,000 players which goes hand in hand with maps being considered overweighted. These maps also tend to have higher difficulty and a faster bpm, two things I know I struggle with purely based on personal experience.

References

ˆ1 FAQ. (n.d.). Retrieved from https://osu.ppy.sh/wiki/en/FAQ ˆ2 Pps by grumd - osu! farm maps. (n.d.). Retrieved from https://osu-pps.com/#/osu/maps ˆ3 lendermaniac · player info: Osu! (n.d.). Retrieved from https://osu.ppy.sh/users/4655284