Cluster - Grouping Best Players

```
library(ggdendro)
library(dplyr)
```

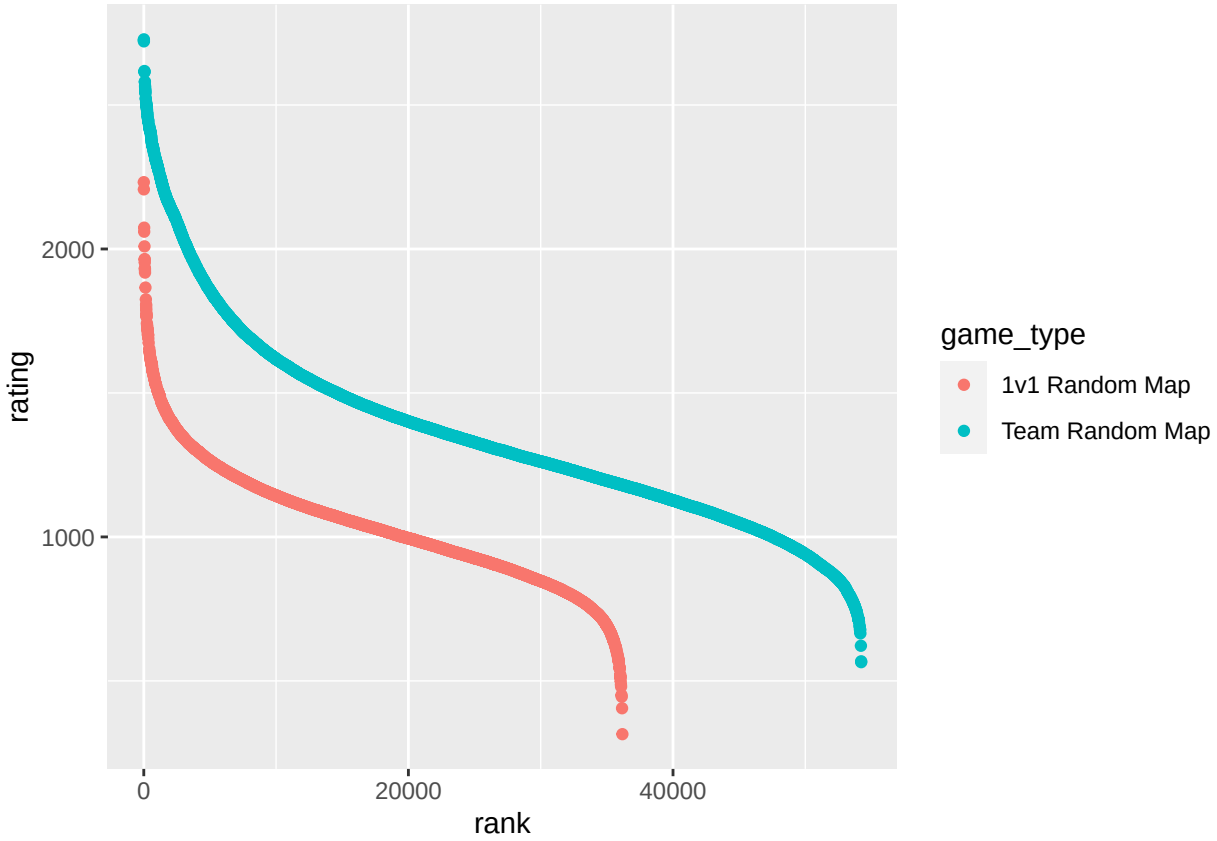
```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(ggdendro)
library(ggthemes)
library(tidyr)
aoe2 <- read.csv("../Data/aoe2_leaderboard_sample.csv")
```

After completing hypothesis 3, and seeing different groups of players with their wins and losses balance, we decided to look at what the game defines as the best players and compare that to a more consistent metric across sports, percentage of wins.

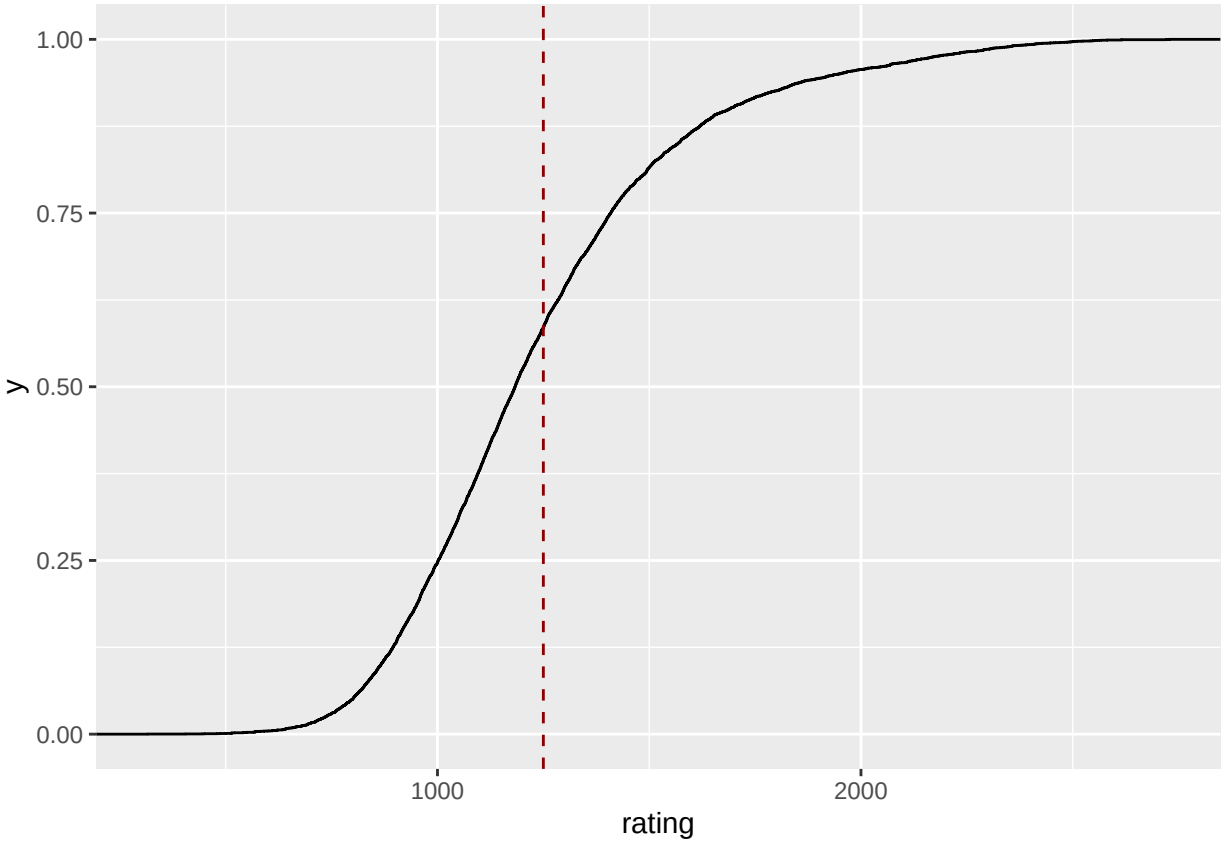
Let's first acknowledge that rank and rating are exact measurements of each other: If a player plays more and wins more their rating will rise and their rank will go up via the leaderboard.

```
ggplot(aoe2) +
  geom_point(aes(rank, rating, color = game_type))
```



So now, we need to decide how to subset the data with the best players, by rank or by rating. After some tests with graphs (discarded ones in scratch), the one that subsetted our clusters the most logically and the data as a whole seemed to be rating at around its inflection point:

```
aoe2 %>%
  ggplot(aes(x = rating)) +
  stat_ecdf() +
  geom_vline(xintercept = 1250,
             linetype = "dashed",
             color = "darkred")
```



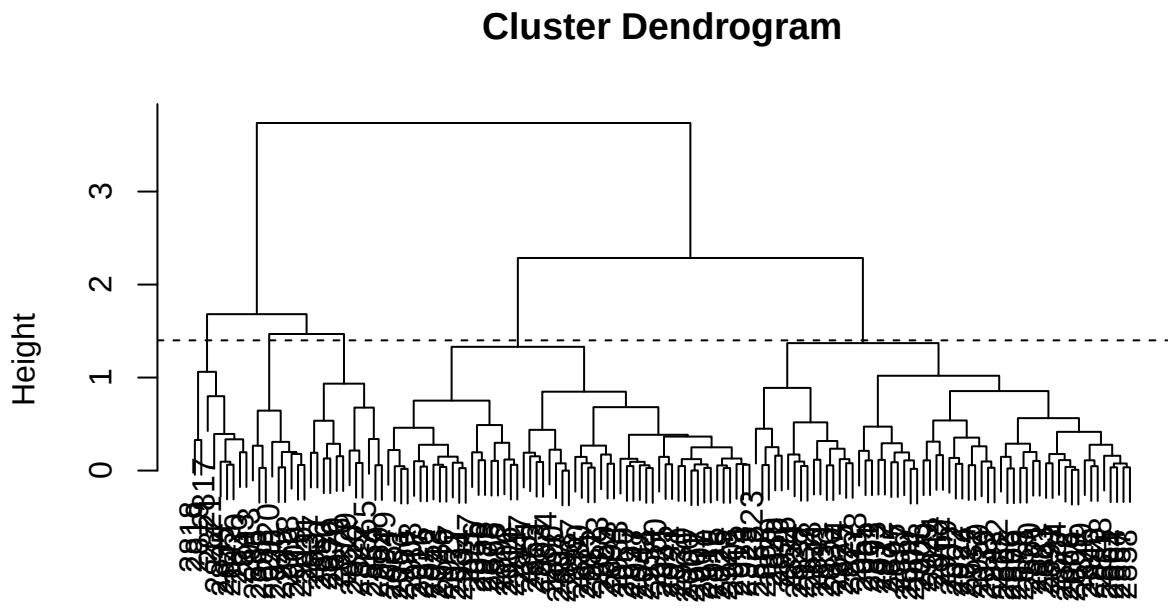
So we will use only players who have at least a 1250 rating. We also found that only a small amount of players who play individual have a rating of 1250, so we decided to also only use the team random map players.

```
aoe2$perc_wins <- aoe2$wins / aoe2$games
aoe2$scalerating <- scale(aoe2$rating)
aoe2$scalepercwins <- scale(aoe2$perc_wins)

filtered_aoe2 <- aoe2[aoe2$rating > 2200 & aoe2$game_type == "Team Random Map", ]
head(filtered_aoe2)
```

```
##      profile_id           name rank rating country games wins losses drops
## 2817    1195260         Kellar    8   2728      NO   364  319     45     2
## 2818     312774    Sun | Keno_    9   2725      MX   300  238     62     4
## 2819     560474        Sunzets   10   2721      MX   253  191     62     4
## 2820     199419      gkt_cloud   49   2618      TW   749  536    213     8
## 2821     332603 teutonic_tanks   54   2616      AT   280  237     43     9
## 2822    1892228            KaN   57   2614      AR   211  153     58     2
##            game_type perc_wins scalerating scalepercwins
## 2817 Team Random Map 0.8763736    4.272475      3.144900
## 2818 Team Random Map 0.7933333    4.263845      2.435177
## 2819 Team Random Map 0.7549407    4.252339      2.107046
## 2820 Team Random Map 0.7156208    3.956060      1.770989
## 2821 Team Random Map 0.8464286    3.950307      2.888968
## 2822 Team Random Map 0.7251185    3.944554      1.852163
```

So we now need to decide which clustering method to use. Since we don't know how many groups of players we want to use, we will utilize hierarchical modeling and the dendrogram to determine how many clusters we will use.

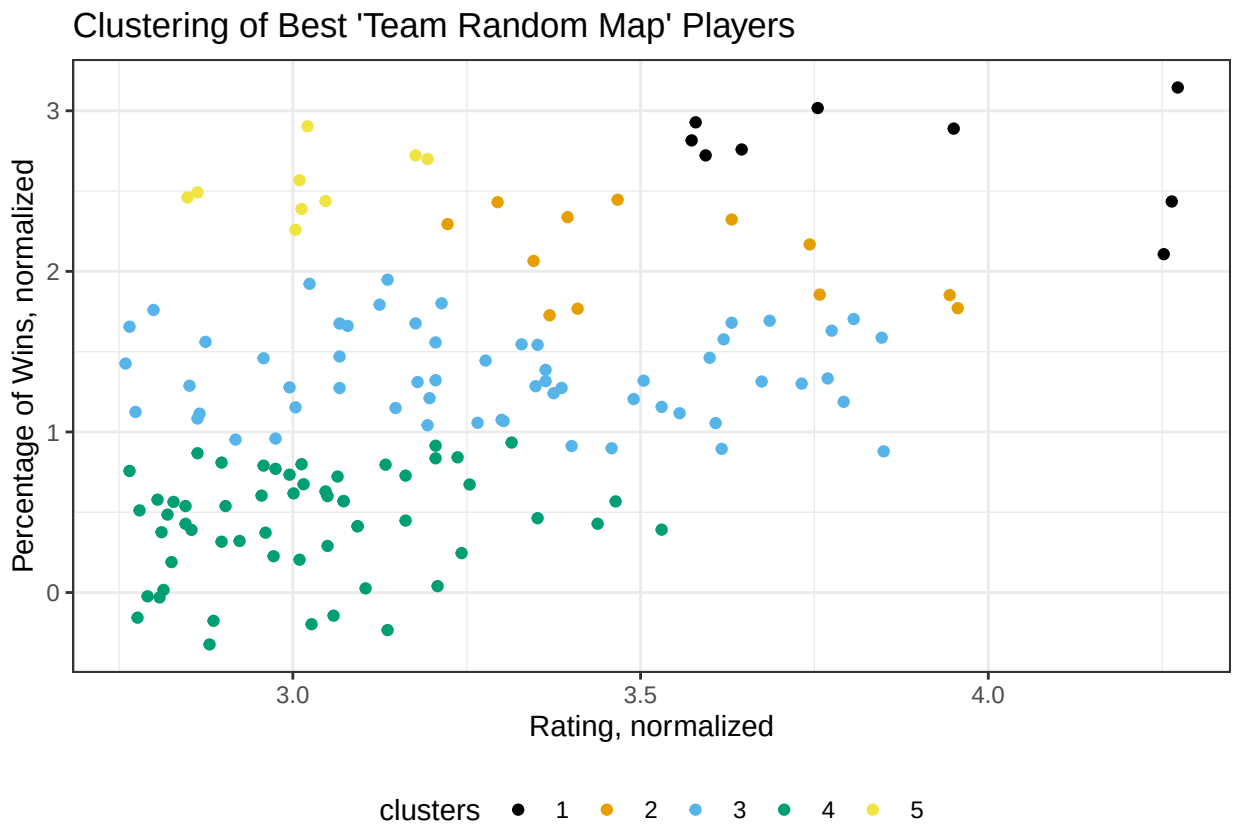
```
aoe2_hclust <-
  hclust(dist(
    dplyr::select(filtered_aoe2,
                  scalerating, scalepercwins)),
         method = "complete")

plot(aoe2_hclust) +
  abline(h = 1.4, lty = 2)
```




```
## integer(0)
```

```
aoe2_player_clusters <-
  cutree(aoe2_hclust,
         k = 5)

filtered_aoe2 <- filtered_aoe2 %>%
  mutate(clusters =
           as.factor(aoe2_player_clusters))

filtered_aoe2 %>%
  ggplot(aes(x = scalerating, y = scalepercwins,
```

```
            color = clusters, label = name)) +
  geom_point() +
  ggthemes::scale_color_colorblind() +
  theme_bw() +
  ggtitle("Clustering of Best 'Team Random Map' Players") +
  xlab("Rating, normalized") +
  ylab("Percentage of Wins, normalized") +
  theme(legend.position = "bottom")
```



Via the dendrogram, we see a gap in the tree with 5 clusters. While splitting into 7 clusters may have a bigger gap, having 7 different types of players seemed excessive for this analysis and spread our data out too thin.

In order to understand these clusters in our graph, let's get the mean and sd of the percentage of wins to cluster them quantitatively.

```
mean(aoe2$perc_wins)
```

```
## [1] 0.5084085
```

```
sd(aoe2$perc_wins)
```

```
## [1] 0.1170038
```

With this, we split our data into 5 player archetypes: 1. Group 1 (Black Dots): The Greats - They have the most games played, the highest rating/rank, and the highest win percentage at 80% + 2. Group 2 (Orage Dots): All Stars - The group with the 2nd highest average rating, but a lower win percentage at around 60-70% 3. Group 3 (Blue Dots): Competitive Players - The highest spread of rating (and therefore games played), but a consistent win percentage of around 55-60%. These players create the biggest cluster in our set. 4. Group 4 (Green Dots): The Casuals - These players, in relative to the rest of this selected data, have the lowest win % at around 50% therefore not too high of rating. 5. Group 5 (Yellow Dots): The Underrated / Unproven - This group defined the purpose of our clustering, as they had 2nd highest win percentage as a cluster at about 75-80%, but do not have the number of games to back up their greatness. This puts them lower in the rating and the rank, to be put with the average "competitive players" and "casuals".

An important note to make is that with more games, Group 5 can fall into the Greats if their win percentage stays consistent, but we predict they would most likely fall into the orange dots when they play enough games.

Lastly, let's quantify each cluster to see if any last trends can be seen.

```
filtered_aoe2 %>%
  group_by(clusters) %>%
  summarise(avgrank = mean(rank), avgrating = mean(rating), avggames = mean(games), avgwins = mean(wins
```

```
## # A tibble: 5 x 7
##   clusters avgrank avgrating avggames avgwins avglosses avgpercwins
##   <fct>      <dbl>     <dbl>    <dbl>   <dbl>     <dbl>       <dbl>
## 1 1           118.     2590.    276.    230.       45.9       0.831
## 2 2           319.     2475     295.    218.       76.8       0.753
## 3 3           645.     2386.    270.    178.       91.7       0.667
## 4 4          1024.     2293.    395.    220.      175.        0.559
## 5 5          1013.     2292.     94.4    76.4      18         0.807
```

An interesting note here is the average number of games column. While we knew cluster 5 would have a very low average number of games compared to the rest of the clusters. We did not expect group 4 to have such a higher amount of average games compared to the rest of the clusters. This does support their name as the "casuals", but brings up an interesting question to be explored in another ESports report:

Does playing a high number of games inevitably converge your winning percentage down? Is there a tactic in not playing too many games in a row / a lot over the same period of time?