Final Project

Vamshee Deepak Goud Katta

12/8/2021

```
# Reading the input data
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.0     v forcats 0.5.1
```

```
## Warning: package ’tibble’ was built under R version 4.1.2
```

```
## Warning: package ’readr’ was built under R version 4.1.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(ISLR)
library(flexclust)
```

```
## Loading required package: grid
```

```
## Loading required package: lattice
```

```
## Loading required package: modeltools
```

```
## Loading required package: stats4
```

```
library(cowplot)
library(GGally)
```

```
## Warning: package ’GGally’ was built under R version 4.1.2
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
set.seed(123)

# Arranging the data in descending order of BustedAt value
Gamble <- read.csv('bustabit.csv')
head(Gamble)
Gamble %>%
    arrange(desc(BustedAt))
head(Gamble)
```

```
# Deriving relevant features for clustering
bustabit <- Gamble %>%
  mutate(CashedOut = ifelse(is.na(CashedOut), BustedAt + .01, CashedOut),
         Profit = ifelse(is.na(Profit), 0, Profit),
         Losses = ifelse(Profit == 0, -1*Bet, 0),
         GameWon = ifelse(Profit == 0, 0, 1),
         GameLost = ifelse(Profit == 0, 1, 0))

# Look at the first five rows of the features data
head(bustabit)
```

```
##         Id  GameID  Username Bet CashedOut Bonus Profit BustedAt
## 1 14196549 3366002     papai   5      1.20   0.0   1.00     8.24
## 2 10676217 3343882    znay22   3      1.41    NA   0.00     1.40
## 3 15577107 3374646  rrrrrrrr   4      1.33   3.0   1.44     3.15
## 4 25732127 3429241 sanya1206  10      1.64    NA   0.00     1.63
## 5 17995432 3389174       ADM  50      1.50   1.4  25.70     2.29
## 6 14147823 3365723     afrod   2      1.05    NA   0.00     1.04
##               PlayDate Losses GameWon GameLost
## 1 2016-11-20T19:44:19Z      0       1        0
## 2 2016-11-14T14:21:50Z     -3       0        1
## 3 2016-11-23T06:39:15Z      0       1        0
## 4 2016-12-08T18:13:55Z    -10       0        1
## 5 2016-11-27T08:14:48Z      0       1        0
## 6 2016-11-20T17:50:55Z     -2       0        1
```

```
# Creating per-player statistics
player <- bustabit %>%
  group_by(Username) %>%
  summarize(AverageCashedOut = mean(CashedOut),
            AverageBet = mean(Bet),
            TotalProfit = sum(Profit),
            TotalLosses = sum(Losses),
            GamesWon = sum(GameWon),
            GamesLost = sum(GameLost))
# Displaying the cleaned data
head(player)
```

```
## # A tibble: 6 x 7
##   Username         AverageCashedOut AverageBet TotalProfit TotalLosses GamesWon
##   <chr>                       <dbl>      <dbl>       <dbl>       <dbl>    <dbl>
## 1 ----------------             1.04       10.3        0.7           -8        2
## 2 --dilib--                    1.50      211.       371.         -1239        2
## 3 -_-TUYUL-_-                  2.65       30.4       48.4         -140        1
## 4 -__---                       1.33    21776.    183322.       -116046       19
## 5 -31337-                      1.22       32.5       21.5          -55        3
## 6 -i_                          1.14        3          0.96           0        2
## # ... with 1 more variable: GamesLost <dbl>
```

```
# Standardizing the data
standard <- function(x)
  {z=(x-mean(x))/sd(x)}

# Apply the function to each numeric variable in the clustering set
standardized <- player %>%
    mutate_if(is.numeric, standard)

# Summarize our standardized data
summary(standardized)
```

```
##    Username         AverageCashedOut     AverageBet       TotalProfit
##  Length:4149        Min.   :-0.76289   Min.   :-0.1773   Min.   :-0.09052
##  Class :character   1st Qu.:-0.28157   1st Qu.:-0.1765   1st Qu.:-0.09050
##  Mode  :character   Median :-0.18056   Median :-0.1711   Median :-0.08974
##                     Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.00000
##                     3rd Qu.: 0.02752   3rd Qu.:-0.1384   3rd Qu.:-0.08183
##                     Max.   :41.72651   Max.   :24.9971   Max.   :40.73652
##   TotalLosses           GamesWon          GamesLost
##  Min.   :-41.84541   Min.   :-0.4320   Min.   :-0.41356
##  1st Qu.:  0.09837   1st Qu.:-0.3696   1st Qu.:-0.41356
##  Median :  0.10847   Median :-0.3071   Median :-0.33306
##  Mean   :  0.00000   Mean   : 0.0000   Mean   : 0.00000
##  3rd Qu.:  0.10916   3rd Qu.:-0.1196   3rd Qu.:-0.09156
##  Max.   :  0.10916   Max.   :13.2534   Max.   :19.30911
```

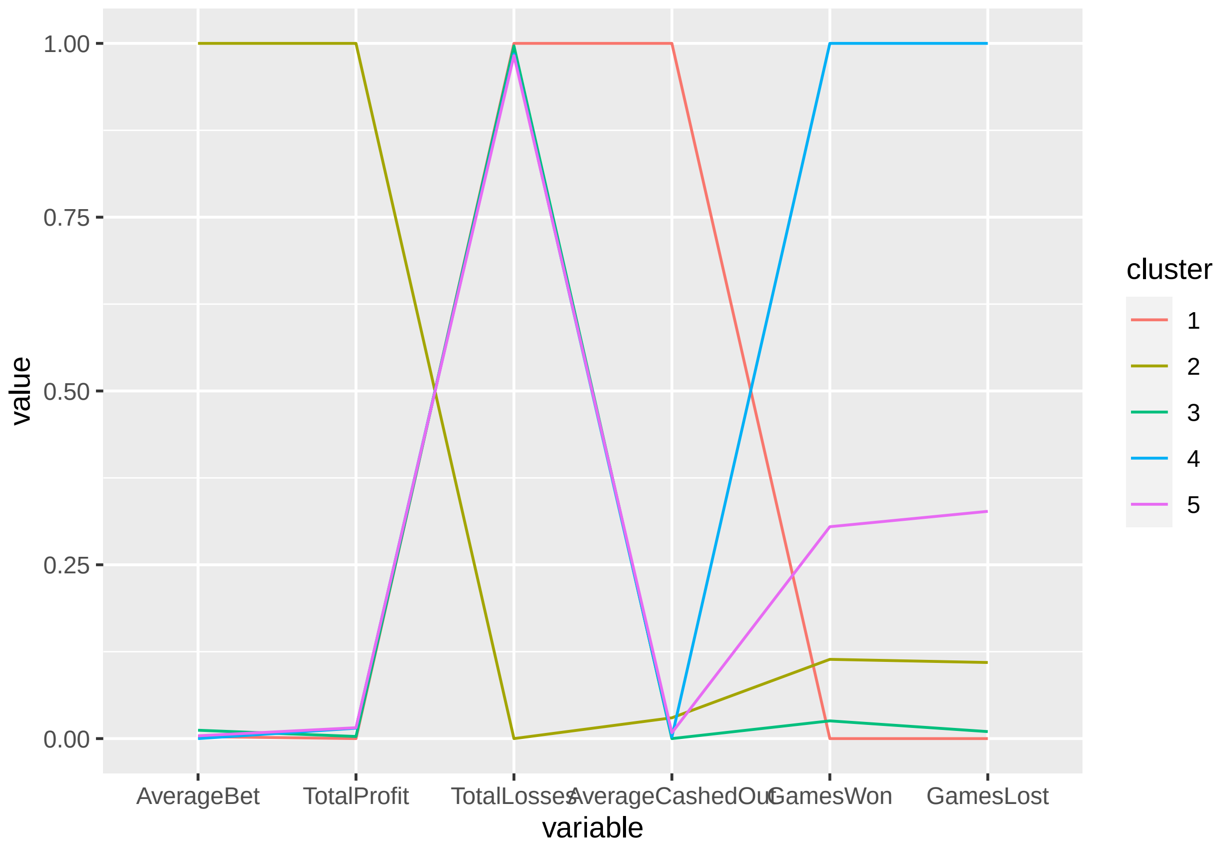
```
set.seed(2021)

# Cluster the players using k-means with five clusters
cluster <- select(standardized,-Username)%>%
                                          kmeans( centers = 5)

# Store the cluster assignments back into the clustering data frame object
player$cluster <- factor(cluster$cluster)

# Look at the distribution of cluster assignments
table(player$cluster)
```

```
##
##    1    2    3    4    5
##   17   16 3626   78  412
```

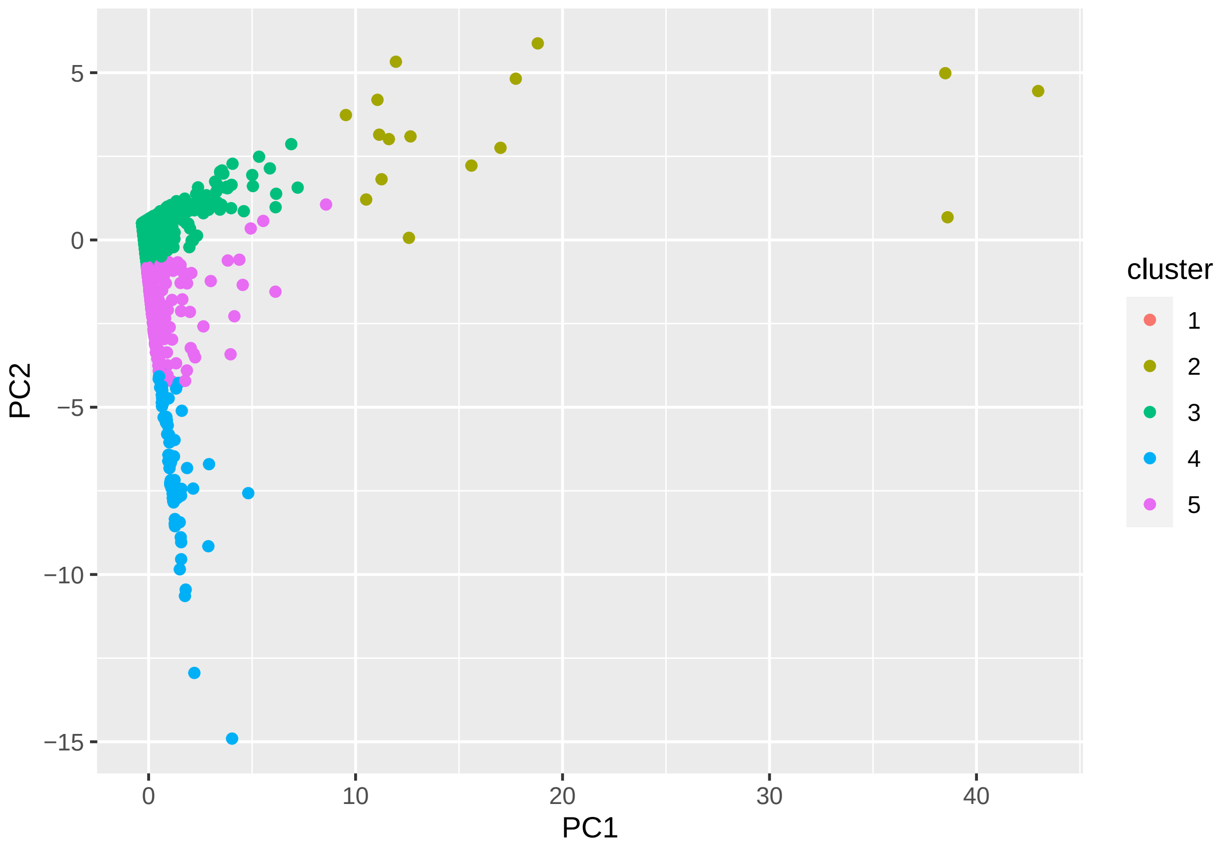
```r
# Group by the cluster assignment and calculate averages
cluster_avg <- player %>%
    group_by(cluster) %>%
    summarize_if(is.numeric,mean)

# View the resulting table
cluster_avg
```

```
## # A tibble: 5 x 7
##   cluster AverageCashedOut AverageBet TotalProfit TotalLosses GamesWon GamesLost
##   <fct>              <dbl>      <dbl>       <dbl>       <dbl>    <dbl>     <dbl>
## 1 1                  27.4       1278.        619.       -581.    0.706      1.53
## 2 2                   2.47    298946.    1198191.   -1056062.   10.6        8.06
## 3 3                   1.70      4024.       4273.      -4366.    2.91       2.13
## 4 4                   1.76       432.      18568.     -16724.   87.2       61.2
## 5 5                   1.92      1633.      19363.     -19205.   27.1       21.0
```

```r
# Create the min-max scaling function
deviation <- function(x) {
    z=(x-min(x))/(max(x)-min(x))
}

# Apply this function to each numeric variable in the bustabit_clus_avg object
bustabit_avg <- cluster_avg %>%
    mutate_if(is.numeric, deviation)

# Create a parallel coordinate plot of the values
ggparcoord(bustabit_avg, columns = c(2,3,4,5,6,7),
           groupColumn = "cluster", scale = "globalminmax", order = "skewness")
```



```
# Principal components
components <- as.data.frame(prcomp(standardized[2:7])$x)

# Store the cluster assignments in the new data frame
components$cluster <- player$cluster

# Use ggplot() to plot PC1 vs PC2, and color by the cluster assignment
principle <- ggplot(components,aes(PC1,PC2,color=cluster))+
      geom_point()

# View the resulting plot
principle
```



```
# Forming clusters dataframe with cluster names
clusters <- c(
    "Risky Commoners",
    "High Rollers",
    "Risk Takers",
    "Cautious Commoners",
    "Strategic Addicts"
)

# Append the cluster names to the cluster means table
Named_clusters <- cluster_avg %>%
    cbind(Name = clusters)

# View the cluster means table with your appended cluster names
Named_clusters
```

```
##   cluster AverageCashedOut  AverageBet  TotalProfit   TotalLosses   GamesWon
## 1       1        27.448235   1278.2574     619.4041     -581.2941  0.7058824
## 2       2         2.470024 298945.6618 1198191.1631 -1056062.1875 10.5625000
## 3       3         1.699993   4024.1102    4272.6656    -4365.7788  2.9109211
## 4       4         1.758407    432.1163   18568.1141   -16724.0641 87.1794872
## 5       5         1.915776   1633.2292   19362.9909   -19205.1165 27.0606796
##   GamesLost               Name
## 1  1.529412    Risky Commoners
## 2  8.062500       High Rollers
## 3  2.128792        Risk Takers
## 4 61.205128 Cautious Commoners
## 5 21.036408  Strategic Addicts
```