Hackathon

Kevin Collins, Ryan Quinlan, Benjamin Kwin, Dominic Ventura

2024-02-13

Read in Data

```
# Previous Season Data
previous_season = read.csv("Reduced.Prev.Season (1).csv", stringsAsFactors = T)

# Remove Observations without Premium Revenue Earnings
previous_season = previous_season[7:154, ]

# Current Season Schedule
schedule = read.csv("schedule (1).csv", stringsAsFactors = T)

# Create Dummy Variable for Home Opener
previous_season$Opening.Day = ifelse(previous_season$HomeGameNumber == 1, 1, 0)
schedule$Opening.Day = ifelse(schedule$HomeGameNumber == 1, 1, 0)
```

PREMIUM REVENUE

TOTAL REVENUE
Stepwise selection Fit on 2018-2022 for Total Revenue Train: 2018-2022; Test: None
“‘r library(tidyverse) library(caret) library(leaps) library(MASS)
# Fit the full model on all data (2018 - 2022) 	
total.full.model.18.22 <- lm(Total.Revenue ~ Opening.Day + Year + Days.from.Beg + Is.Holiday + Opp.Pred.Wpct + Opp.Prev.ASG + Opp.Prev.Conf.Finals + Opp.Prev.NBA.Finals + Opp.Prev.NBA.Champs, data = previous_season)
# Stepwise regression model on ’18 - ’22 total.revenue.model <- stepAIC(total.full.model.18.22, direction = “both”, trace = FALSE) # Summary ’18 - ’22 	
summary(total.revenue.model) “‘

PREMIUM REVENUE

```
## ## Call: ## lm(formula = Total.Revenue ~
Opening.Day + Year + Days.from.Beg + ##
Opp.Pred.Wpct + Opp.Prev.ASG +
Opp.Prev.NBA.Finals, data = previous_season) ## ##
Residuals: ##     Min      1Q  Median      3Q
Max ## -341074  -95222    2054   83934  600931 ##
## Coefficients: ##                        Estimate
Std. Error t value Pr(>|t|) ## (Intercept)
-294447473   14822271 -19.865  < 2e-16 *** ##
Opening.Day             161561      85061   1.899
0.05956 . ## Year                     146419
7338  19.954  < 2e-16 *** ## Days.from.Beg
1016        251   4.050 8.41e-05 *** ##
Opp.Pred.Wpct           306094     124997   2.449
0.01556 * ## Opp.Prev.ASG              49249
16296   3.022  0.00298 ** ## Opp.Prev.NBA.Finals
84103      53014   1.586  0.11488 ## --- ##
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05
'.' 0.1 ' ' 1 ## ## Residual standard error:
141100 on 141 degrees of freedom ## Multiple
R-squared:  0.7757, Adjusted R-squared:  0.7661 ##
F-statistic: 81.27 on 6 and 141 DF,  p-value: <
2.2e-16
```

PCA and K-Means Clustering

Load packages and read in data

```
require(stats)
require(vegan)
```

```
## Loading required package: vegan

## Loading required package: permute

## This is vegan 2.6-4

##
## Attaching package: 'vegan'

## The following object is masked from 'package:caret':
##
##     tolerance
```

```
require(tidyverse)
require(ggbiplot)
```

```
## Loading required package: ggbiplot

## Warning: package 'ggbiplot' was built under R version 4.2.3
```

```r
require(factoextra)
```

```
## Loading required package: factoextra
```

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

```r
require(ggplot2)

# Remove: MasterEventName, MasterEventDate, and BowlTier from data
schedule = schedule[, -c(2,3,5)]

# Create Predictions for Premium and Total Revenue
schedule$Prem.Rev.Pred = predict(premium.revenue.model, newdata = schedule)
schedule$Tot.Rev.Pred = predict(total.revenue.model, newdata = schedule)

# Remove Year and Previous Win Percentage
schedule = schedule[, -c(1, 7)]
```

Prep data

```r
scaled_data = scale(schedule)
```

Covariance matrix

```r
S = round(cov(scaled_data), 2)
print(S)
```

```
##                      HomeGameNumber Is.Holiday Days.from.Beg Opp.Pred.Wins
## HomeGameNumber                 1.00       0.09          1.00          0.29
## Is.Holiday                     0.09       1.00          0.10          0.10
## Days.from.Beg                  1.00       0.10          1.00          0.30
## Opp.Pred.Wins                  0.29       0.10          0.30          1.00
## Opp.Pred.Wpct                  0.29       0.10          0.30          1.00
## Opp.Prev.ASG                   0.11       0.19          0.10          0.67
## Opp.Prev.NBA.Finals            0.09      -0.04          0.08          0.22
## Opp.Prev.Conf.Finals           0.17      -0.09          0.16          0.45
## Opening.Day                   -0.27      -0.04         -0.25          0.14
## Prem.Rev.Pred                  0.76       0.11          0.76          0.72
## Tot.Rev.Pred                   0.65       0.15          0.65          0.83
##                      Opp.Pred.Wpct Opp.Prev.ASG Opp.Prev.NBA.Finals
## HomeGameNumber                0.29         0.11                 0.09
## Is.Holiday                    0.10         0.19                -0.04
## Days.from.Beg                 0.30         0.10                 0.08
## Opp.Pred.Wins                 1.00         0.67                 0.22
## Opp.Pred.Wpct                 1.00         0.67                 0.22
## Opp.Prev.ASG                  0.67         1.00                 0.03
## Opp.Prev.NBA.Finals           0.22         0.03                 1.00
## Opp.Prev.Conf.Finals          0.45         0.43                 0.38
## Opening.Day                   0.14         0.03                -0.03
## Prem.Rev.Pred                 0.72         0.63                 0.19
## Tot.Rev.Pred                  0.83         0.69                 0.26
```

```
##                       Opp.Prev.Conf.Finals Opening.Day Prem.Rev.Pred
## HomeGameNumber                        0.17       -0.27          0.76
## Is.Holiday                           -0.09       -0.04          0.11
## Days.from.Beg                         0.16       -0.25          0.76
## Opp.Pred.Wins                         0.45        0.14          0.72
## Opp.Pred.Wpct                         0.45        0.14          0.72
## Opp.Prev.ASG                          0.43        0.03          0.63
## Opp.Prev.NBA.Finals                   0.38       -0.03          0.19
## Opp.Prev.Conf.Finals                  1.00       -0.07          0.58
## Opening.Day                          -0.07        1.00          0.04
## Prem.Rev.Pred                         0.58        0.04          1.00
## Tot.Rev.Pred                          0.44        0.18          0.95
##                       Tot.Rev.Pred
## HomeGameNumber                0.65
## Is.Holiday                    0.15
## Days.from.Beg                 0.65
## Opp.Pred.Wins                 0.83
## Opp.Pred.Wpct                 0.83
## Opp.Prev.ASG                  0.69
## Opp.Prev.NBA.Finals           0.26
## Opp.Prev.Conf.Finals          0.44
## Opening.Day                   0.18
## Prem.Rev.Pred                 0.95
## Tot.Rev.Pred                  1.00
```

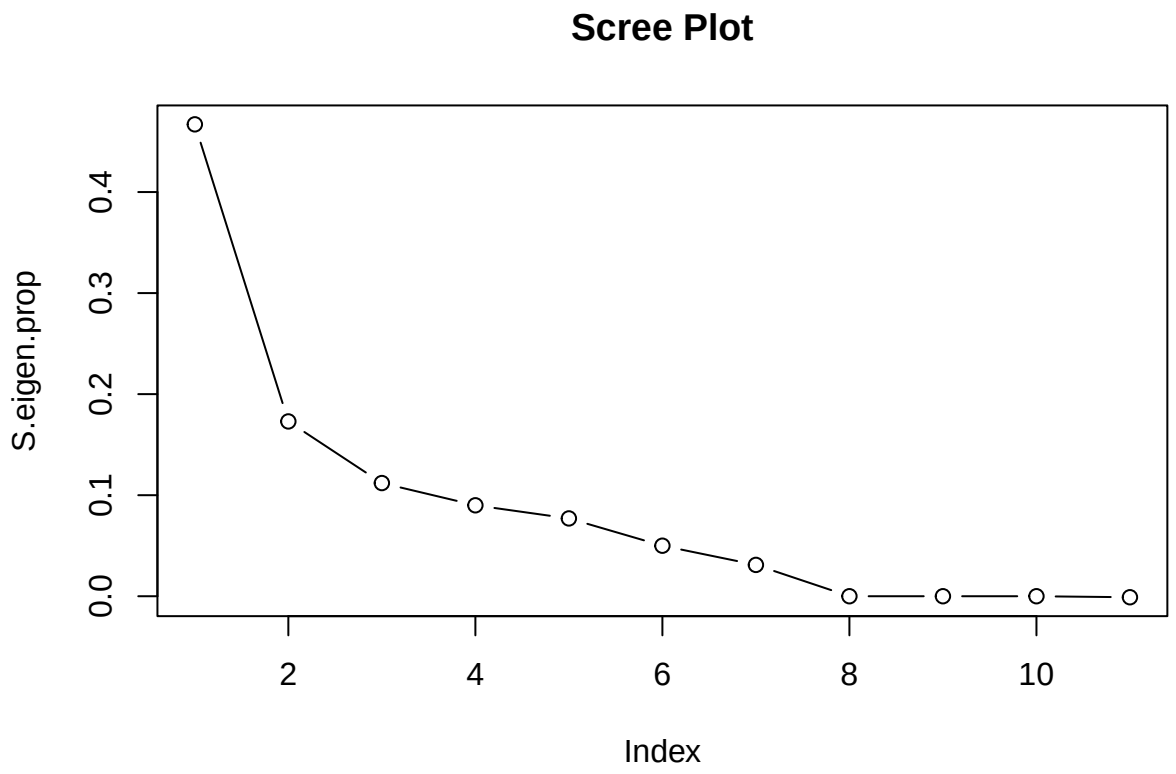
Eigenvalues

```
S.eigen = eigen(S)  # Eigenvalues always sum to total number of variables
S.eigen.prop = round(S.eigen$values / sum(S.eigen$values), 3)  # Calculate proportion of total amount o
print(S.eigen.prop)
```

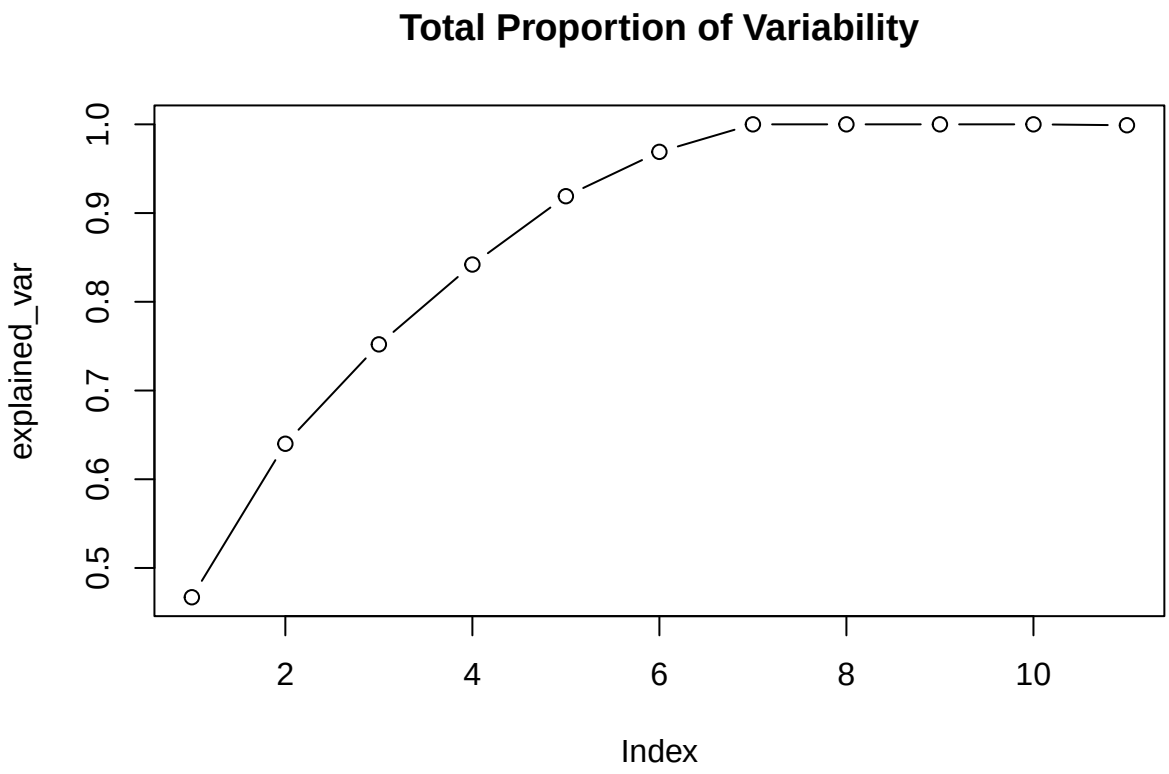
```
##  [1]  0.467  0.173  0.112  0.090  0.077  0.050  0.031  0.000  0.000  0.000
## [11] -0.001
```

Scree plots

```
# Calculate cumulative explained variability
explained_var = NULL

for (i in 1:length(S.eigen.prop)) {
    explained_var = c(explained_var, sum(S.eigen.prop[c(1:i)]))
}

plot(S.eigen.prop, type = "b", main = "Scree Plot")
```




```
plot(explained_var, type = "b", main = "Total Proportion of Variability")
```



Interpret eigenvalues

```
components = 5
eigenvectors = round(S.eigen$vectors, 2)

loadings = eigenvectors[ , 1:components]
rownames(loadings) = colnames(schedule)
colnames(loadings) = c("PC1", "PC2", "PC3", "PC4", "PC5")
print(loadings)
```

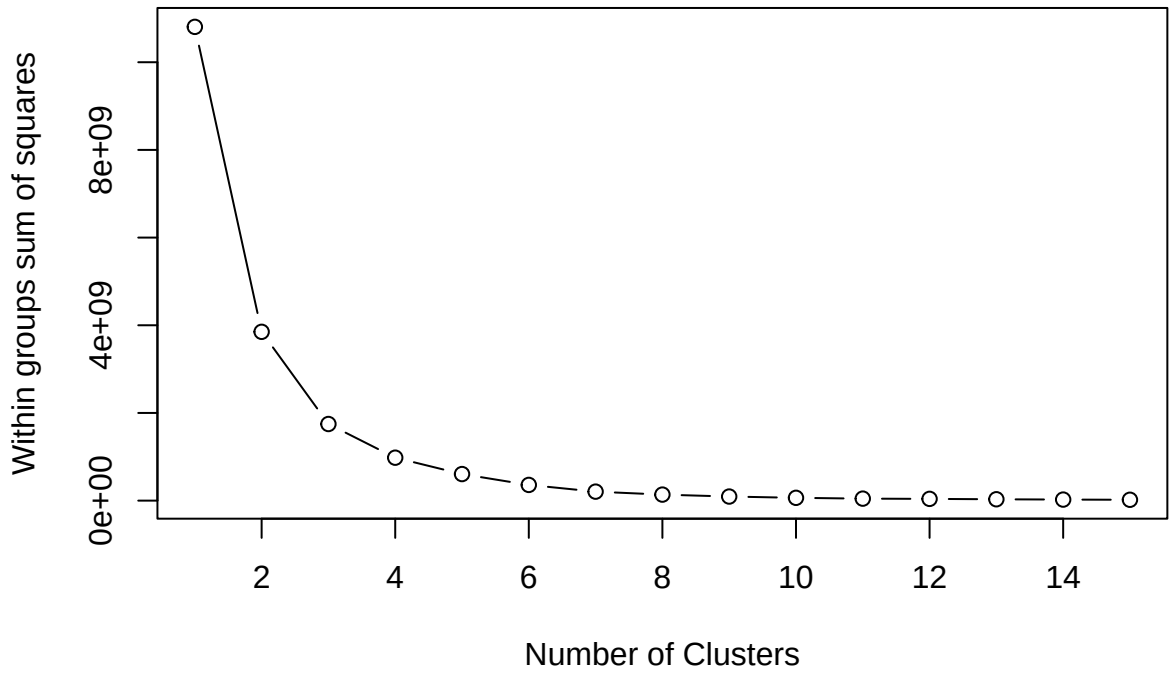
```
##                          PC1   PC2   PC3   PC4   PC5
## HomeGameNumber         -0.29 -0.54  0.02 -0.13 -0.05
## Is.Holiday             -0.07 -0.04  0.53  0.61 -0.51
## Days.from.Beg          -0.29 -0.53  0.04 -0.15 -0.06
## Opp.Pred.Wins          -0.38  0.27  0.06 -0.01  0.06
## Opp.Pred.Wpct          -0.38  0.27  0.06 -0.01  0.06
## Opp.Prev.ASG           -0.30  0.30  0.18  0.26  0.31
## Opp.Prev.NBA.Finals    -0.12  0.08 -0.62  0.16 -0.64
## Opp.Prev.Conf.Finals   -0.25  0.17 -0.48  0.20  0.15
## Opening.Day            -0.01  0.39  0.24 -0.65 -0.43
## Prem.Rev.Pred          -0.43 -0.10  0.01 -0.09  0.03
## Tot.Rev.Pred           -0.43  0.02  0.09 -0.13 -0.10
```

Reduce dimensionality of dataset

```r
# Select desired columns from PCA
df = schedule[, c(10, 5, 8, 9, 7)]
```

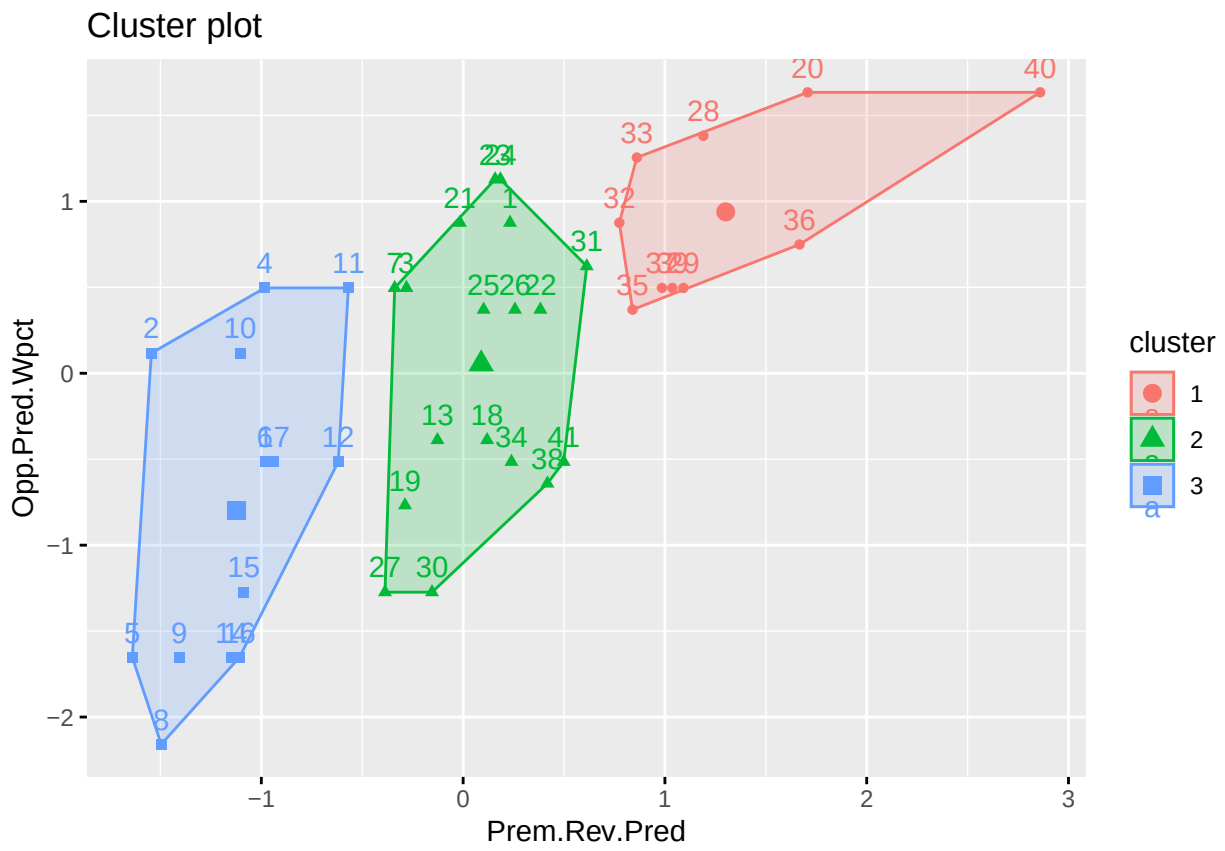
Determine optimal number of clusters

```r
wss = 0

for (i in 1:15) {
  km.out <- kmeans(df, centers = i, nstart = 20)
  # Save total within sum of squares to wss variable
  wss[i] <- km.out$tot.withinss

}

# Plot total within sum of squares vs. number of clusters
plot(1:15, wss, type = "b", xlab = "Number of Clusters", ylab = "Within groups sum of squares")
```

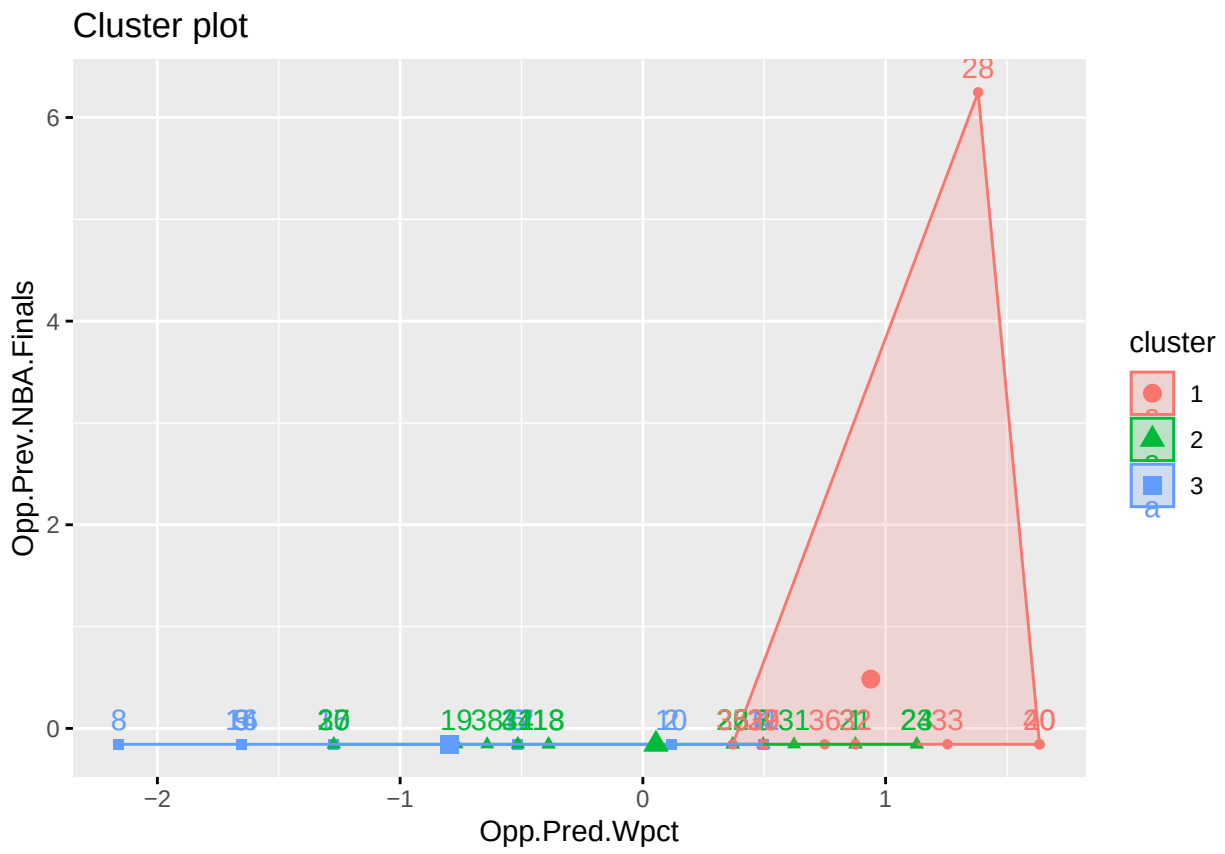


k-Means Clustering

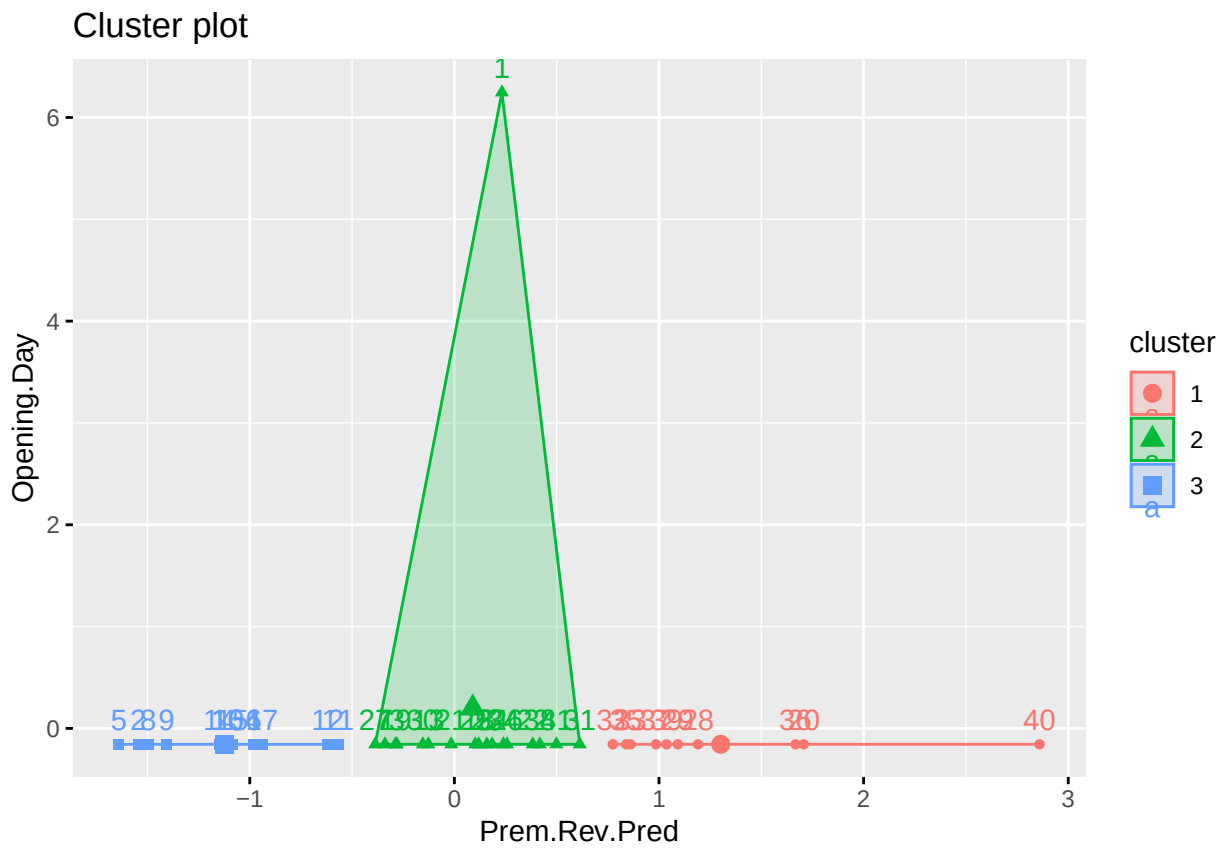
```r
#plot(df$height, df$tube.diam, pch = 19, xlab = "Height", ylab = "Tube Diameter")
#plot(df$height, df$wing1.length, pch = 19, xlab = "Height", ylab = "Wing Length")
#plot(df$tube.diam, df$wing1.length, pch = 19, xlab = "Tube Diameter", ylab = "Wing Length")

clusters = kmeans(df, centers = 3)
df$cluster_number = clusters$cluster
fviz_cluster(clusters, df[c(1, 2)])
```



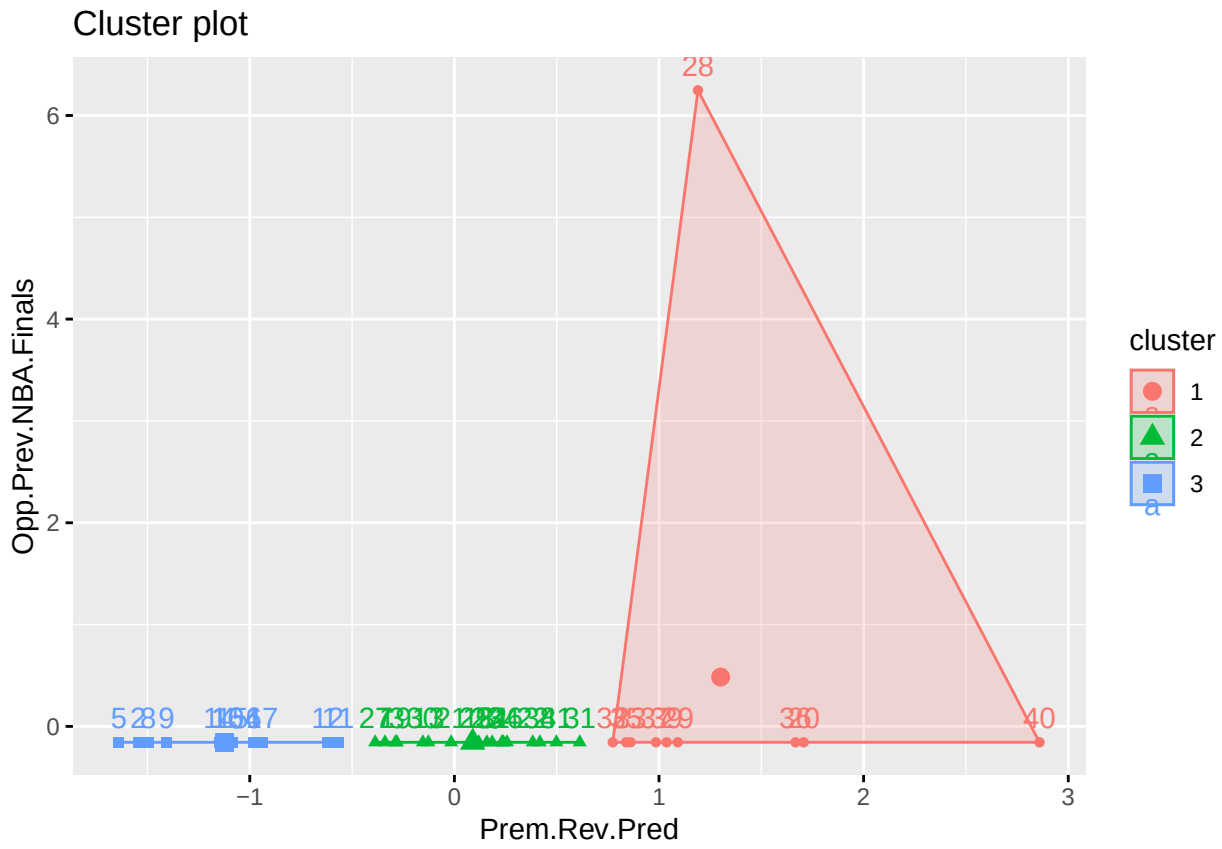

```
fviz_cluster(clusters, df[c(2, 5)])
```




```
fviz_cluster(clusters, df[c(1, 4)])
```




```
fviz_cluster(clusters, df[c(1, 5)])
```


Cluster plot
Opp.Prev.NBA.Finals
Prem.Rev.Pred
cluster
1
2
3
6
4
2
0
−1
0
1
2
3
28
40