ML Final Project: Analysis

Andrew Pagtakhan

January 20, 2020

```
library(dplyr)
library(dendextend)
library(GGally)
library(ggfortify)
library(ggplot2)
library(ggrepel)
library(gridExtra)
library(knitr)
library(mclust)
library(NbClust)

# set for reproducible results
set.seed(14)

# clear variables
rm(list = ls())

# set working directory
# wd <- 'C:/Users/apagta950/Documents/NYU/Courses/Spring 2020/ML/Projects/ML-NBA/Data/'
opts_knit$set(root.dir = normalizePath('..'))

# define file name for analysis
filename <- 'Data/season_stats_clean.csv'
```

Research Question: Are there underlying patterns of groupings between NBA team compensation vs. overall team skillsets?

Data Cleaning

The data cleaning was done in separate R scripts. https://github.com/joemarlo/ML-NBA Steps: * Filtered data to the 2016 - 2017 season * Assigned player to one team based on the most minutes he played for, including stats across all teams played for * Scraped player salaries and RPM data from ESPN website, using fuzzy matching on player names to join to the main dataset * Scaled/Transformed data using cube root (shown below)

Exploratory Data analysis

```
# load data
nba <- read.csv(filename)
str(nba)
```

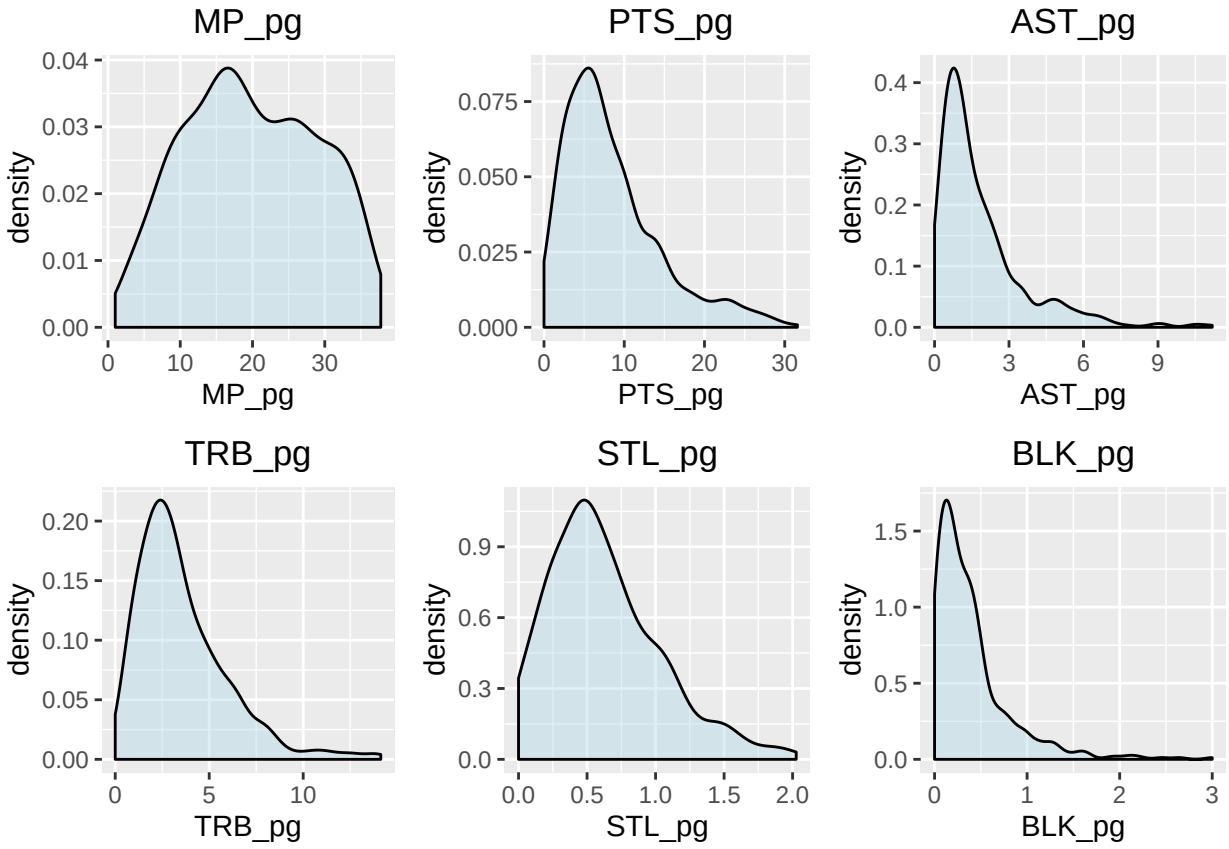
```
## 'data.frame':    486 obs. of  41 variables:
##  $ Year        : int  2017 2017 2017 2017 2017 2017 2017 2017 2017 2017 ...
##  $ Player      : Factor w/ 486 levels "A.J. Hammons",..: 12 377 429 35 15 81 286 292 451 8 ...
##  $ Tm          : Factor w/ 30 levels "ATL","BOS","BRK",..: 21 3 21 26 19 18 27 12 15 25 ...
##  $ Pos         : Factor w/ 6 levels "C","PF","PF-C",..: 6 2 1 6 1 1 2 2 6 5 ...
##  $ Age         : int  23 26 23 31 28 28 31 27 35 26 ...
##  $ G           : int  68 38 80 61 39 62 72 61 71 61 ...
##  $ MP          : int  1055 558 2389 1580 584 531 2335 871 1914 1773 ...
##  $ FG          : int  134 70 374 185 89 45 500 77 274 183 ...
##  $ FGA         : int  341 170 655 420 178 86 1049 168 595 466 ...
##  $ X3P         : int  94 37 0 62 0 0 23 0 15 70 ...
##  $ X3PA        : int  247 90 1 151 4 0 56 1 54 212 ...
##  $ X2P         : int  40 33 374 123 89 45 477 77 259 113 ...
##  $ X2PA        : int  94 80 654 269 174 86 993 167 541 254 ...
##  $ FT          : int  44 45 157 83 29 15 220 23 80 96 ...
##  $ FTA         : int  49 60 257 93 40 22 271 33 130 136 ...
##  $ TRB         : int  86 115 615 125 177 158 524 220 391 451 ...
##  $ AST         : int  40 18 86 78 12 25 139 57 98 99 ...
##  $ STL         : int  37 14 88 21 20 25 46 18 115 60 ...
##  $ BLK         : int  8 15 78 7 22 23 89 24 29 44 ...
##  $ PTS         : int  406 222 905 515 207 105 1243 177 643 532 ...
##  $ VORP        : num  -0.1 -0.1 1.5 -0.6 -0.3 0.4 1.8 0.4 1.3 0.4 ...
##  $ PER         : num  10.1 11.8 16.5 9 12.9 12.7 18.6 11.6 13.3 11.3 ...
##  $ MP_pg       : num  15.5 14.7 29.9 25.9 15 ...
##  $ FG_pg       : num  1.97 1.84 4.67 3.03 2.28 ...
##  $ FGA_pg      : num  5.01 4.47 8.19 6.89 4.56 ...
##  $ X3P_pg      : num  1.382 0.974 0 1.016 0 ...
##  $ X3PA_pg     : num  3.6324 2.3684 0.0125 2.4754 0.1026 ...
##  $ X2P_pg      : num  0.588 0.868 4.675 2.016 2.282 ...
##  $ X2PA_pg     : num  1.38 2.11 8.18 4.41 4.46 ...
##  $ FT_pg       : num  0.647 1.184 1.962 1.361 0.744 ...
##  $ FTA_pg      : num  0.721 1.579 3.212 1.525 1.026 ...
##  $ TRB_pg      : num  1.26 3.03 7.69 2.05 4.54 ...
##  $ AST_pg      : num  0.588 0.474 1.075 1.279 0.308 ...
##  $ STL_pg      : num  0.544 0.368 1.1 0.344 0.513 ...
##  $ BLK_pg      : num  0.118 0.395 0.975 0.115 0.564 ...
##  $ PTS_pg      : num  5.97 5.84 11.31 8.44 5.31 ...
##  $ team_salary: int  96276478 83392086 96276478 95667033 101408319 83666820 108372142 94002004 110960
##  $ win_pct     : num  0.573 0.244 0.573 0.39 0.415 0.378 0.744 0.512 0.524 0.5 ...
##  $ team_rank   : int  10 30 10 23 21 24 2 13 11 15 ...
##  $ Salary      : num  5994764 1790092 3140517 12500000 4638203 ...
##  $ RPM         : num  -2.12 -2.59 1.38 -4.28 -1.53 0.01 0.96 -3.47 0.12 1.27 ...
```

```
# replace NA values in RPM column with 0s
nba$RPM[is.na(nba$RPM)] <- 0
```

We decided to use the 14 features in our analysis (on a per game basis) because they provide a good balance of offensive (e.g Pts, Ast) and defensive stats (e.g. Reb, Blk). Additionally, advanced statistics such as VORP and PER utilize a combination of these 'base' statistics in their calculations. So, by using these base statistics between offensive and defensive metrics, this is more likely to provide more balanced groupings between those who are more offensive and those who are better at defense.

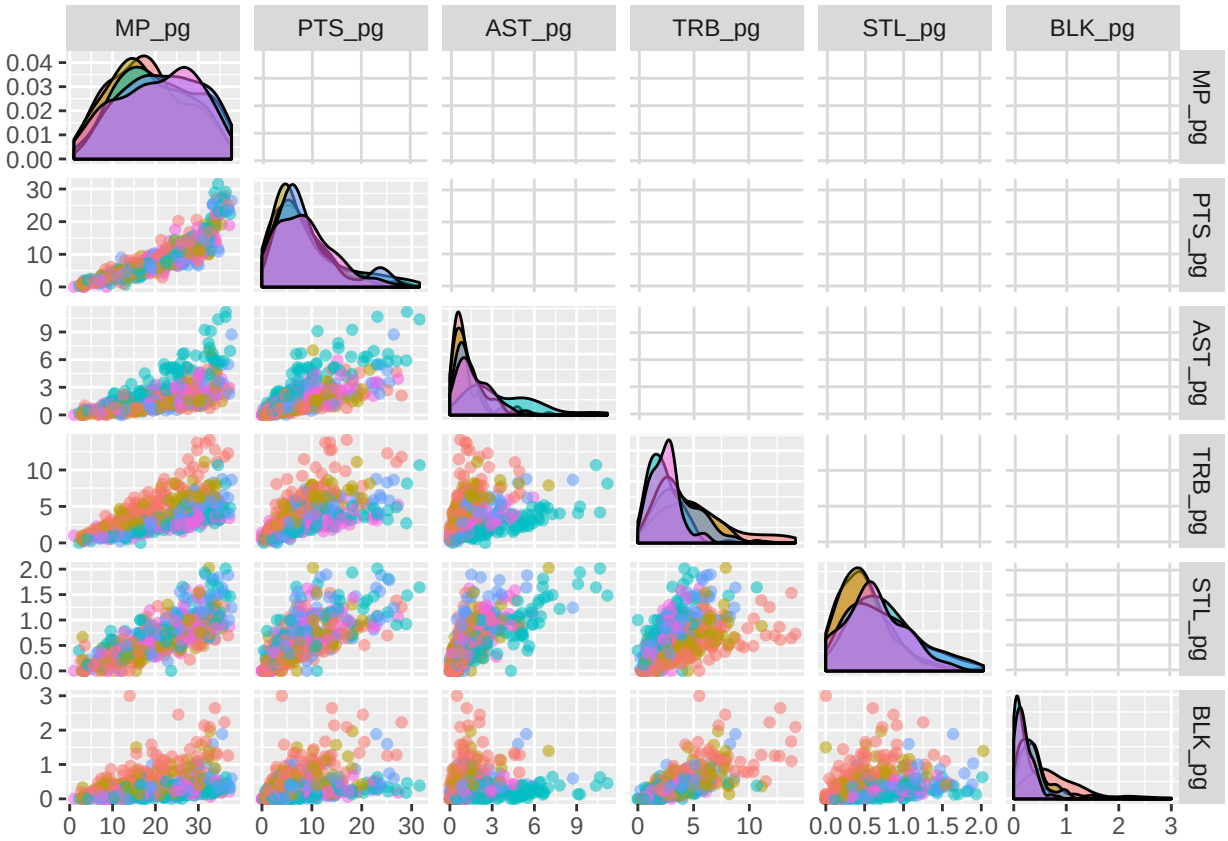
```
# plot distributions of key per game stats
features_pg <- grep('_pg', names(nba), value = TRUE)
nba_feat <- nba[ , features_pg]
```

```r
# further subset for plotting purposes
feat_plot <- c('MP_pg', 'PTS_pg', 'AST_pg', 'TRB_pg',
               'STL_pg', 'BLK_pg')

# dataframe for plotting
nba_feat_plot <- nba[ , feat_plot]

# plot key stats
plots_stats <- list(rep(NA, length = length(feat_plot)))
for (i in seq_along(feat_plot)) {
  p <- ggplot(nba_feat_plot, aes_string(x = feat_plot[i])) +
        geom_density(fill = 'lightblue', alpha = 0.4) +
        ggtitle(feat_plot[i]) +
        # center title
        theme(plot.title = element_text(hjust = 0.5))
  plots_stats[[i]] <- p
}

# multiple plotting
cowplot::plot_grid(plotlist = plots_stats, nrow = 2)
```




```r
# look at pairs plots for key stats
nba_feat_plot_pos <- cbind(nba_feat_plot, nba$Pos)
ggpairs(nba_feat_plot_pos,
        columns = 1:(length(names(nba_feat_plot_pos)) - 1),
        progress = FALSE,
```

```
        mapping = ggplot2::aes(colour = nba$Pos, alpha = 0.4),
        upper = list(continuous = wrap('cor', size = 0))
)
```



Most of the features such as points and assists look like they exhibit a negative binomial distribution. If we look at the pairs plot, it looks like there are clear groupings by position.

Run PCA to inspect if there are any groupings

Before running any clustering algorithms, we will perform Principal Component Analysis to determine if there are any potential clusters. We first scaled the data because the ranges of the data can be different. A good example is minutes and blocks per game - Most players will have more minutes per game than blocks per game.

```
# scale/transform data
methods <- c('scale', 'log', 'cube_root')
# choose method - change index
method <- methods[1]
nba_feat_sc <- if (method == methods[1]) {
                scale(nba_feat)
               } else if (method == methods[2]) {
               # replace 0s with small number
               nba_feat[nba_feat == 0] <- .0001
               log(nba_feat)
               } else {
               (nba_feat)^(1/3)
               }
```

```r
# sample size when sampling players in each cluster
# when taking the log, there may not be enough players to sample from
sample_size <- if (method == methods[2]) 2 else 10

# run PCA
# princomp() uses spectral decomposition
# prcomp() uses singular value decomposition
nba_pca <- prcomp(nba_feat_sc)
summary(nba_pca)
```

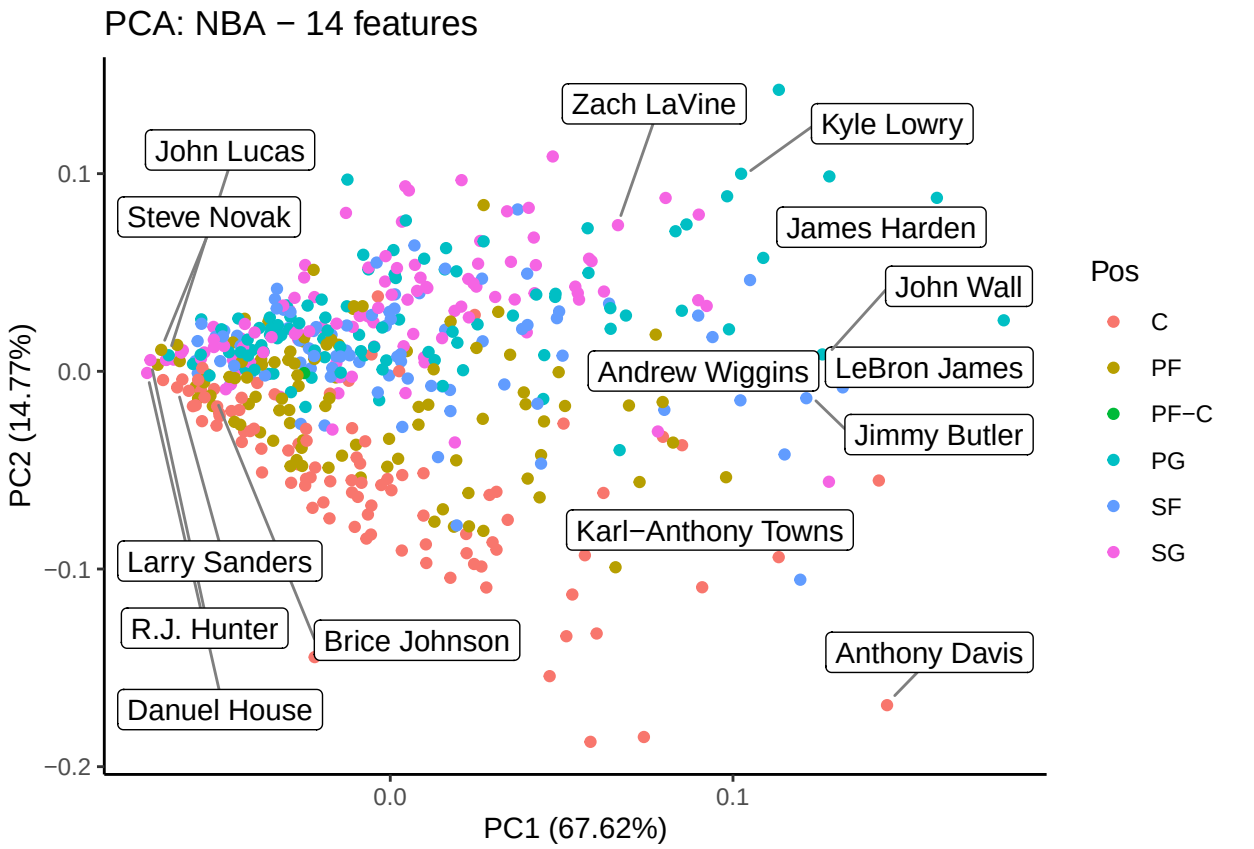
```
## Importance of components:
##                            PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     3.0767 1.4379 0.88672 0.80819 0.63364 0.52061 0.46651
## Proportion of Variance 0.6762 0.1477 0.05616 0.04666 0.02868 0.01936 0.01555
## Cumulative Proportion  0.6762 0.8239 0.88002 0.92667 0.95535 0.97471 0.99026
##                            PC8     PC9    PC10    PC11      PC12      PC13
## Standard deviation     0.29741 0.16765 0.10779 0.09065 2.849e-15 2.194e-15
## Proportion of Variance 0.00632 0.00201 0.00083 0.00059 0.000e+00 0.000e+00
## Cumulative Proportion  0.99658 0.99858 0.99941 1.00000 1.000e+00 1.000e+00
##                            PC14
## Standard deviation     1.597e-15
## Proportion of Variance 0.000e+00
## Cumulative Proportion  1.000e+00
```

```r
# create plot PCA data function
plot_pca <- function(object, frame = FALSE, x = 1, y = 2,
                     data, colour, title, label) {
  # plots data in PCA space
  # object = PCA or K-Means object
  # x = which PC for x-axis (1, 2, ,3, etc..)
  # y = which PC for y-axis (1, 2, 3, etc..)
  # object: PCA or K-means object
  # data = underlying data
  p <- autoplot(nba_pca, x = x, y = y, data = nba, colour = colour, frame = frame) +
        ggtitle(title) +
        # center title
        theme(plot.title = element_text(hjust = 0.5)) +
        geom_label_repel(aes(label = label),
                         box.padding   = 0.35,
                         point.padding = 0.5,
                         segment.color = 'grey50') +
        theme_classic()
  return(p)
}
```

Since 2 components make up 83% of the cumulative variance, we will plot these

```r
# Labels: Players who played more than 36 min per game or less than 3 min per game
labels_pca <- ifelse(nba$MP_pg >= 36 | nba$MP_pg <= 3,
                     as.character(nba$Player), '')
title_pca <- paste0('PCA: NBA - ', ncol(nba_feat) ,' features')

# Plot first two components with positions
plot_pca(nba_pca, data = nba, colour = 'Pos',
         label = labels_pca, title = title_pca
```

```
)
```



Observations Based on the PCA plot, it looks like there are natural grouping based on position from the colors coding. For example, the centers are on the bottom diagonal, point guards are near the top, and SF/PFs are in the middle. It is also noticable that the stars and superstars are on the right-side of the cluster. Since this looks like a fan, we can also say that the stars are placed more towards the 'tips' of the fan. So, we will hypothesize that there may also be clusters from left to right, where the right-most are the top players, and the left side are the lower performing players.

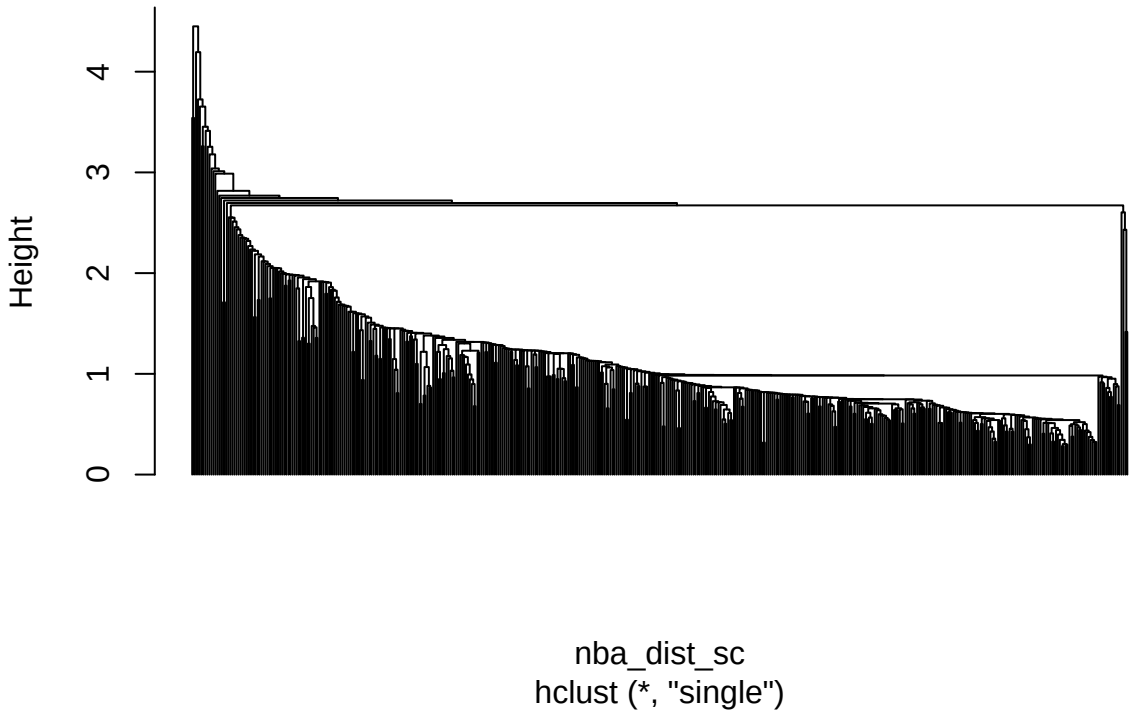
Clustering

Hierarchical clustering

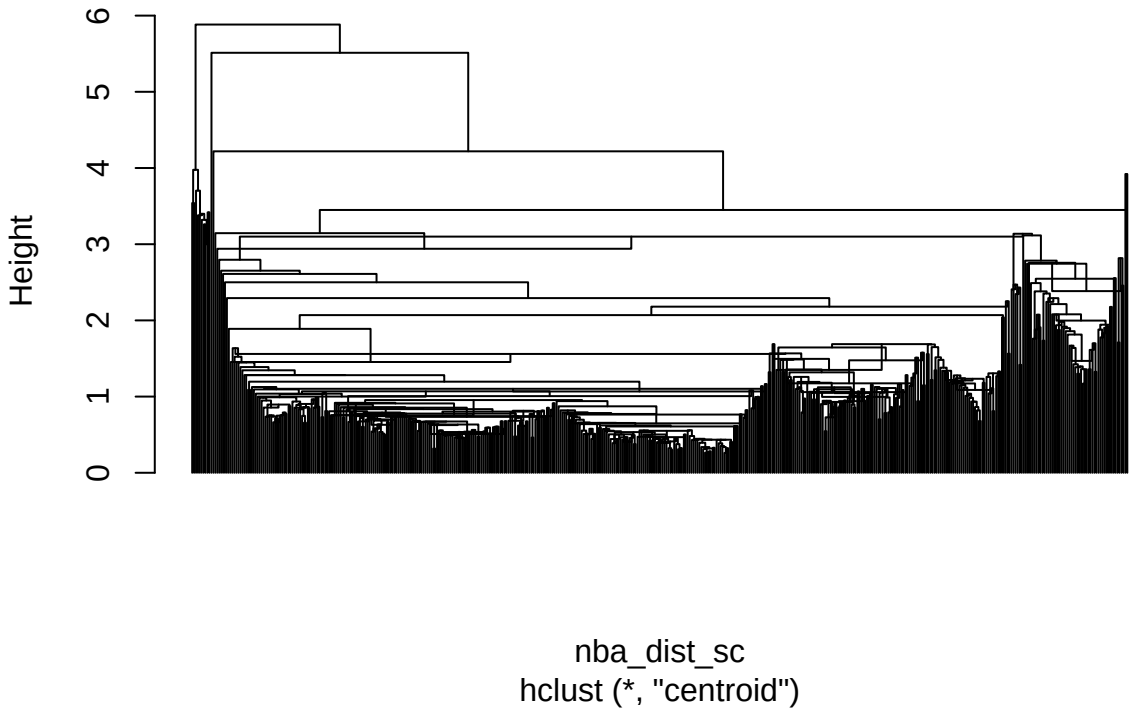
The first method of clustering we will try is hierarchical clustering. The dendrogram can help provide a visual aid in the number of clusters we can start to use.

```
# distance matrix for features
nba_dist_sc <- dist(nba_feat_sc, method = 'euclidean')

# try single, centroid, and ward (D2) linkage hier clustering
hcl_single <- hclust(d = nba_dist_sc, method = 'single')
hcl_centroid <- hclust(d = nba_dist_sc, method = 'centroid')
hcl_ward <-  hclust(d = nba_dist_sc, method = 'ward.D2')
```

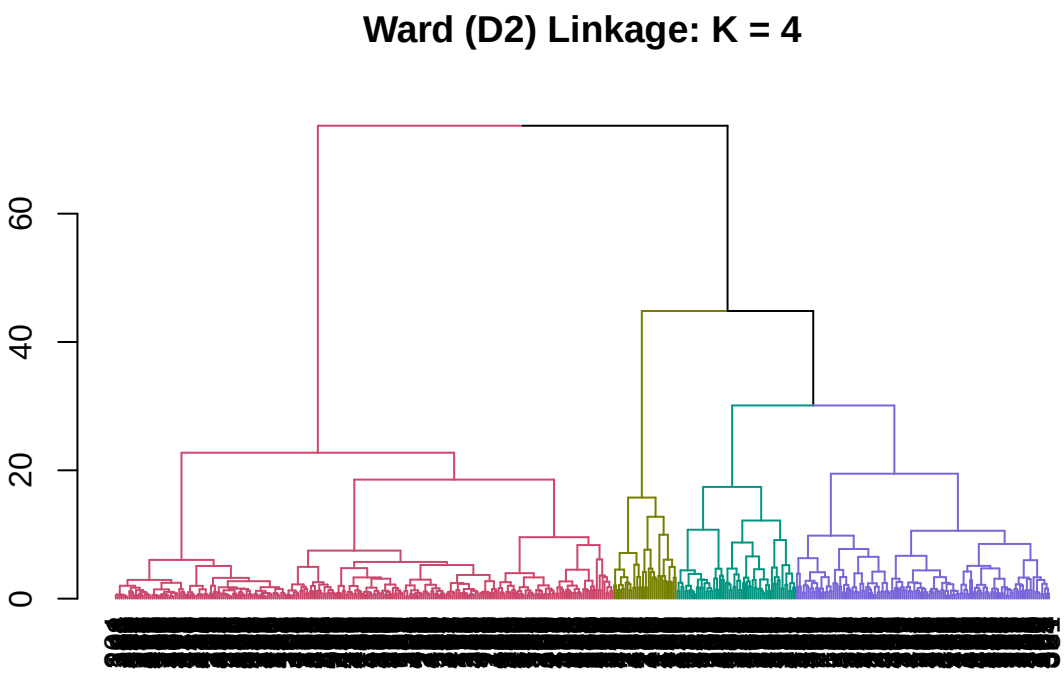
```
# nearest neighbors method
plot(hcl_single, hang = -1, main = 'Single Linkage', labels = FALSE)
```




```
# groups centroid
plot(hcl_centroid, hang = -1, main = 'Centroid Linkage', labels = FALSE)
```



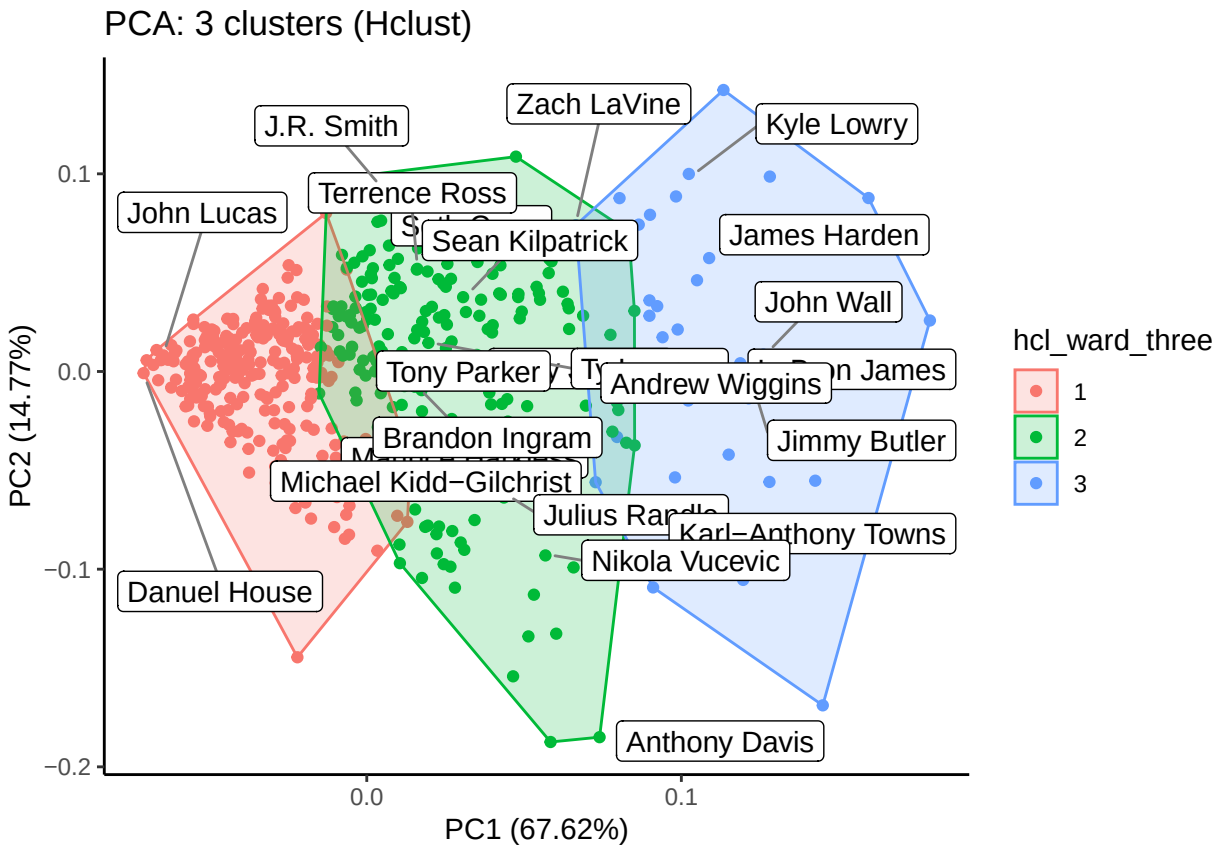
```
# Ward's minimum variance method,
# with dissimilarities are squared before clustering
dend <- as.dendrogram(hcl_ward)
hcl_k <- 4
dend_col <- color_branches(dend, k = hcl_k)
plot(dend_col, main = paste0('Ward (D2) Linkage: K = ', hcl_k))
```

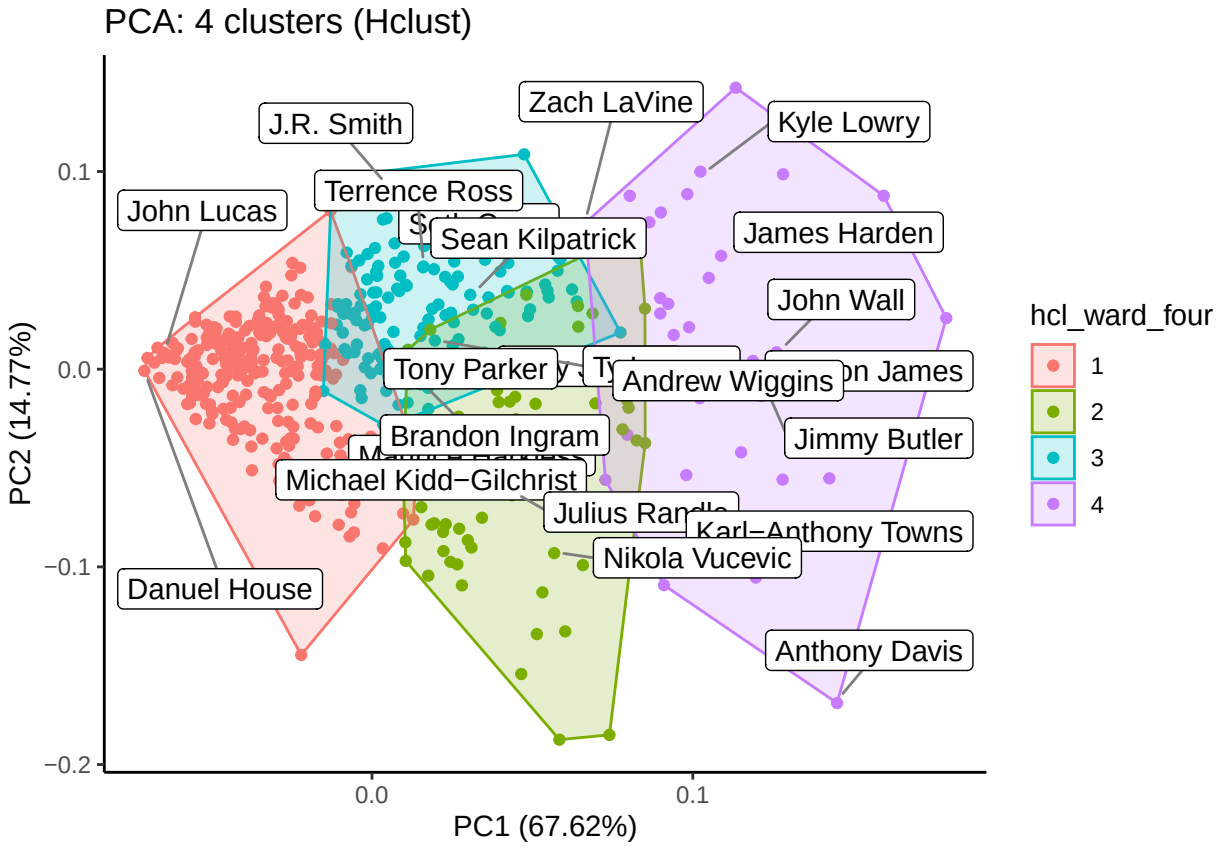


Since the Ward dendrogram seems to be the best among the three, we will look at its distribution for 3 and 4 clusters. We chose these initial groupings because this is a good number of initial clusters to group NBA players.

```
# add cluster labels to main data
nba$hcl_ward_labs_three <- cutree(hcl_ward, k = 3)
nba$hcl_ward_labs_four <- cutree(hcl_ward, k = 4)

# plot frequencies
p_one <- ggplot(data = nba, aes(x = hcl_ward_labs_three)) +
          geom_bar(fill = 'lightblue') + ggtitle('HCL: K = 3')
p_two <- ggplot(data = nba, aes(x = hcl_ward_labs_four)) +
          geom_bar(fill = 'lightblue') + ggtitle('HCL: K = 4')

# combine
cowplot::plot_grid(p_one, p_two, nrow = 2)
```


```r
# add labels to data
nba$hcl_ward_three <- factor(cutree(hcl_ward, k = 3))
nba$hcl_ward_four <- factor(cutree(hcl_ward, k = 4))

# player names to include in plot
hcl_labels <- ifelse(nba$MP_pg >= 36 | nba$MP_pg <= 2.5 |
                     (nba$MP_pg >= 28.8 & nba$MP_pg <= 29) |
                     (nba$MP_pg >= 25 & nba$MP_pg <= 25.2),
                     as.character(nba$Player), '' )

# elements to loop over
hcl_labs <- names(nba %>% select(tail(names(.), 2)))
hcl_ks <- c(3, 4)

# plot hclust labels superimposed over PCA
for (i in seq_along(hcl_labs)) {
  p <- plot_pca(nba_pca, frame = TRUE,
                data = nba, colour = hcl_labs[i],
                title = paste0('PCA: ', hcl_ks[i], ' clusters (Hclust)'),
                label = hcl_labels
  )

  print(p)
}
```


PCA: 3 clusters (Hclust)
J.R. Smith
Zach LaVine
Kyle Lowry
Terrence Ross
John Lucas
Sean Kilpatrick
James Harden
John Wall
Tony Parker
Andrew Wiggins
Brandon Ingram
Jimmy Butler
Michael Kidd−Gilchrist
Karl−Anthony Towns
Nikola Vucevic
Danuel House
Anthony Davis
0.1
0.0
−0.1
−0.2
PC2 (14.77%)
0.0
0.1
PC1 (67.62%)
hcl_ward_three
1
2
3


PCA: 4 clusters (Hclust)
J.R. Smith
Zach LaVine
Kyle Lowry
Terrence Ross
John Lucas
Sean Kilpatrick
James Harden
John Wall
Tony Parker
Andrew Wiggins
Brandon Ingram
Jimmy Butler
Michael Kidd−Gilchrist
Karl−Anthony Towns
Nikola Vucevic
Danuel House
Anthony Davis
0.1
0.0
−0.1
−0.2
PC2 (14.77%)
0.0
0.1
PC1 (67.62%)
hcl_ward_four
1
2
3
4

Visually, it looks like the four cluster solution may be able to give us more actionable insights vs the 3-cluster method. The average stats by cluster shows pretty clear separation among the groups. Group 4 are the stars, followed by group 2, 3, and then 4. The main difference between groups 2 and 3 is that group 2 looks to contain more players who tend to have more rebound and blocks per game.

```
# averages by cluster
nba_hclust_avg <- data.frame(nba
                             %>% select(hcl_ward_four, MP_pg, PTS_pg, TRB_pg,
                                        AST_pg, BLK_pg, STL_pg, VORP, PER, RPM)
                             %>% group_by(hcl_ward_four)
                             %>% summarise_all(list(mean))
                             )
nba_hclust_avg
```

```
##   hcl_ward_four    MP_pg    PTS_pg   TRB_pg    AST_pg    BLK_pg    STL_pg
## 1             1 12.93324  4.248338 2.460736 0.8806744 0.2966152 0.3820685
## 2             2 29.23959 13.347648 7.003145 2.8336944 0.8213332 0.9513499
## 3             3 25.48711 10.433832 3.477880 2.4597941 0.2909106 0.8101552
## 4             4 34.64079 23.940773 6.082089 4.8812610 0.7179077 1.1887785
##          VORP      PER        RPM
## 1 0.004247104 10.76216 -1.8063707
## 2 1.600000000 17.48065  0.7817742
## 3 0.521969697 12.91742 -0.7471212
## 4 3.954545455 22.78182  3.0315152
```

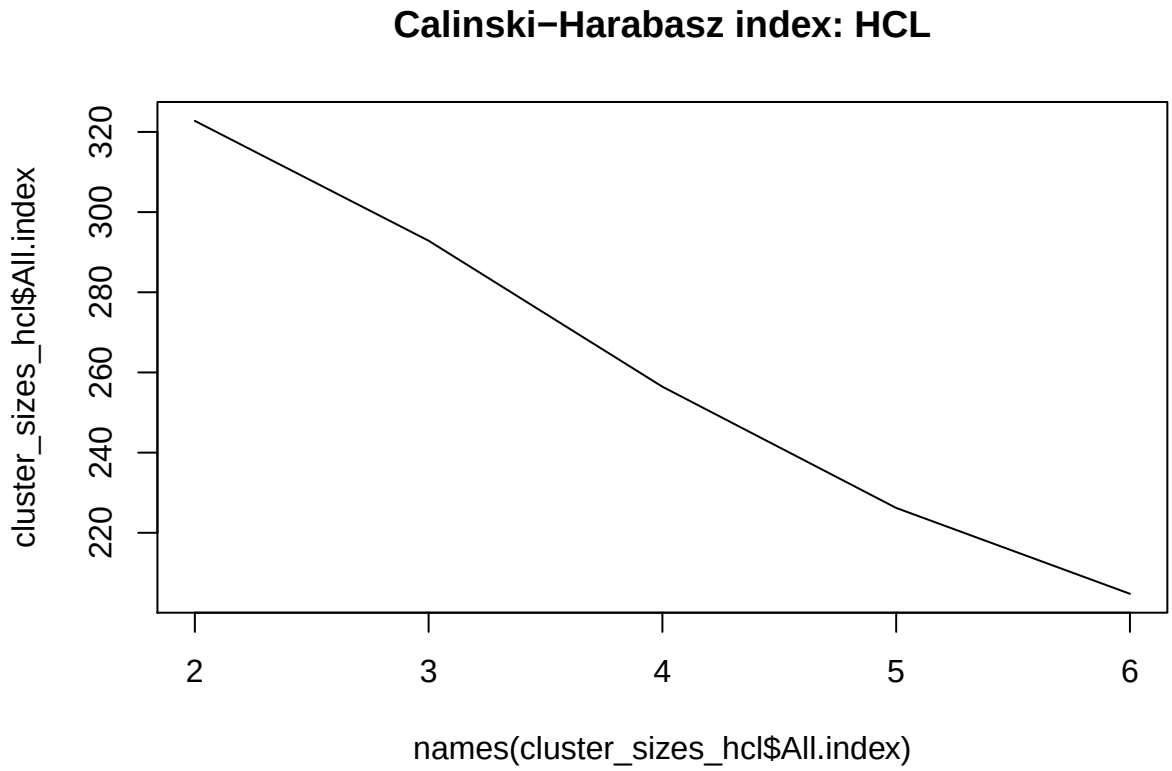
```
# sample players from each cluster
for (k in 1:hcl_ks[2]) {
  print(paste0('Cluster ', k))
  print(sample(subset(nba, hcl_ward_four == k)$Player, size = sample_size))
}
```

```
## [1] "Cluster 1"
##  [1] Sasha Vujacic   Chris Andersen  Justin Anderson Mike Scott
##  [5] Nene Hilario    Willie Reed     Mike Dunleavy   Shawn Long
##  [9] Kyle Wiltjer    Kevon Looney
## 486 Levels: A.J. Hammons Aaron Brooks Aaron Gordon ... Zaza Pachulia
## [1] "Cluster 2"
##  [1] Tobias Harris  Pau Gasol      Dwyane Wade    Dirk Nowitzki  Tony Allen
##  [6] Jahlil Okafor  Nikola Vucevic Jrue Holiday   Zach Randolph  Jeff Teague
## 486 Levels: A.J. Hammons Aaron Brooks Aaron Gordon ... Zaza Pachulia
## [1] "Cluster 3"
##  [1] Raymond Felton   Jerryd Bayless   DeMarre Carroll  Brandon Knight
##  [5] Maurice Harkless Terrence Ross    Kent Bazemore    Andrew Harrison
##  [9] Otto Porter      Jameer Nelson
## 486 Levels: A.J. Hammons Aaron Brooks Aaron Gordon ... Zaza Pachulia
## [1] "Cluster 4"
##  [1] Russell Westbrook Damian Lillard    Brook Lopez       Devin Booker
##  [5] LeBron James      Klay Thompson     Gordon Hayward    C.J. McCollum
##  [9] Kawhi Leonard     John Wall
## 486 Levels: A.J. Hammons Aaron Brooks Aaron Gordon ... Zaza Pachulia
```

Optimize number of clusters

Method: Calinski-Harabasz index

```
# get optimal cluster sizes
cluster_sizes_hcl <- NbClust(data = nba_feat_sc,
                             # it will likely be harder to interpret clusters
                             # past this amount
                             max.nc = 6,
                             method = 'ward.D2',
                             index = 'ch')

# plot C(G)
plot(names(cluster_sizes_hcl$All.index),
     cluster_sizes_hcl$All.index,
     main = 'Calinski-Harabasz index: HCL',
     type = 'l')
```



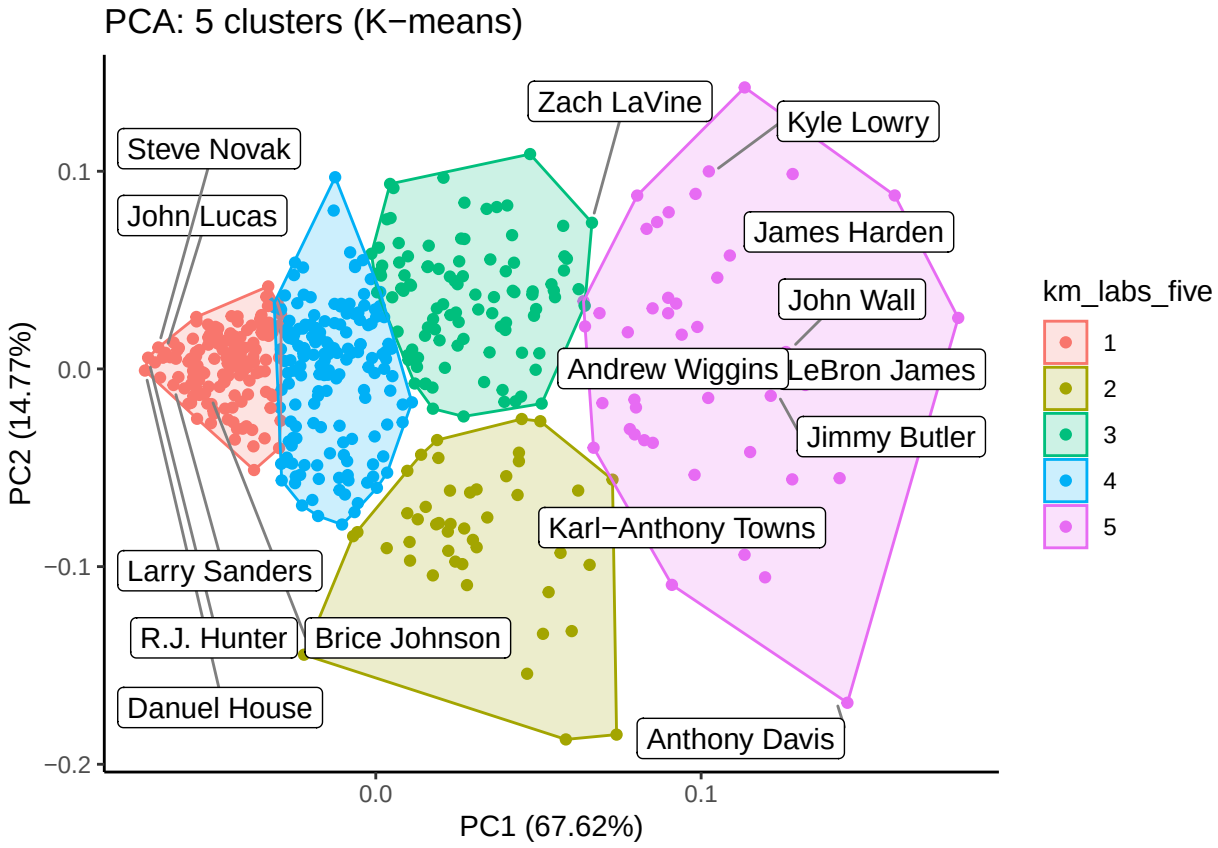
Among the different hierarchical clustering methods, the Ward method seems to be the best. The dendogram looks the most structured and the distribution of players in each cluster is more balanced. Hierarchical clustering could seem like a potential fit if we want the better players to be in a more 'select' group. Although the CH index indicates 2 clusters is optimal, we need to look at the practicality as well. 3 clusters may hav differences between groups of players. But, it is possible that NBA front offices will likely need more differentation when grouping player performance. Looking at the 4 cluster solutions and stats, the blue cluster tends to have more players who rebound, block more shots and tend to be more efficient (based on PER). So, these clusters seem to have decent separation from each other. We will now try K-Means clustering too see if that works better.

K-Means

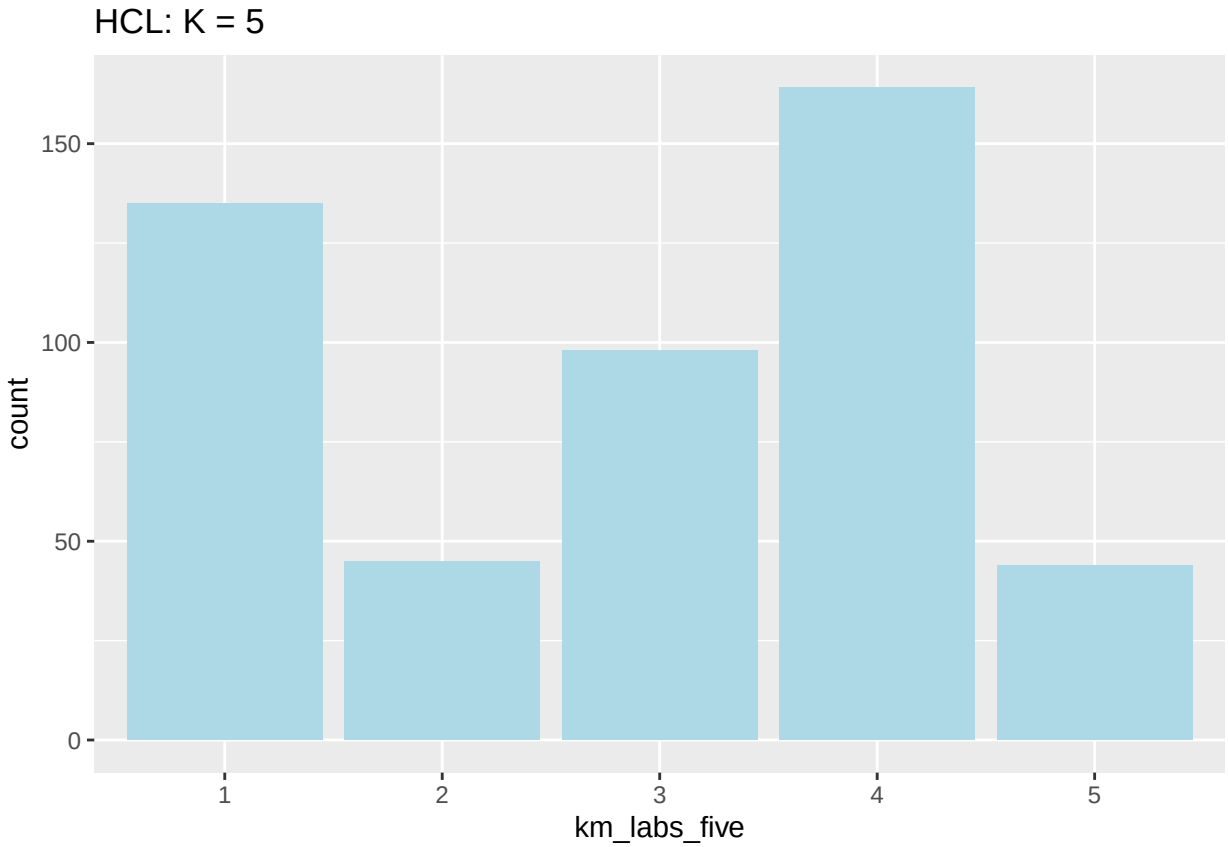
```
# try K-means clustering
# try arbitrary number of clusters first
km_k <- 5
```

```
km_five <- kmeans(x = nba_feat_sc,
             centers = km_k,
             nstart = 100,
             algorithm = 'Hartigan-Wong')
```

```
# plot k-means clusters in PC space
nba$km_labs_five <- factor(km_five$cluster)
# Labels: Players who played more than 36 min per game or less than 3 min per game
km_labels <- ifelse(nba$MP_pg >= 36 | nba$MP_pg <= 3,
                        as.character(nba$Player), '' )

plot_pca(km_five, data = nba, frame = TRUE, colour = 'km_labs_five',
          title = paste0('PCA: ', km_k, ' clusters (K-means)'),
          label = km_labels)
```




```
# get distribution of players in each cluster
ggplot(data = nba,
       aes(x = km_labs_five)) +
       geom_bar(fill = 'lightblue') +
       ggtitle('HCL: K = 5')
```



It looks like the clusters are somewhat interpretable. Clusters to the right seem to indicate star players, while clusters to the left indicate lower performing players. However, the question becomes if 5 clusters is meaningful. Based on the averages for each cluster, there is not much difference between clusters 2 and 4. Additionally, it looks like there is room to better balance the number of players in each cluster and create more separation between clusters. We will now optimize the number of clusters utilizing the Calinski-Harabasz index.

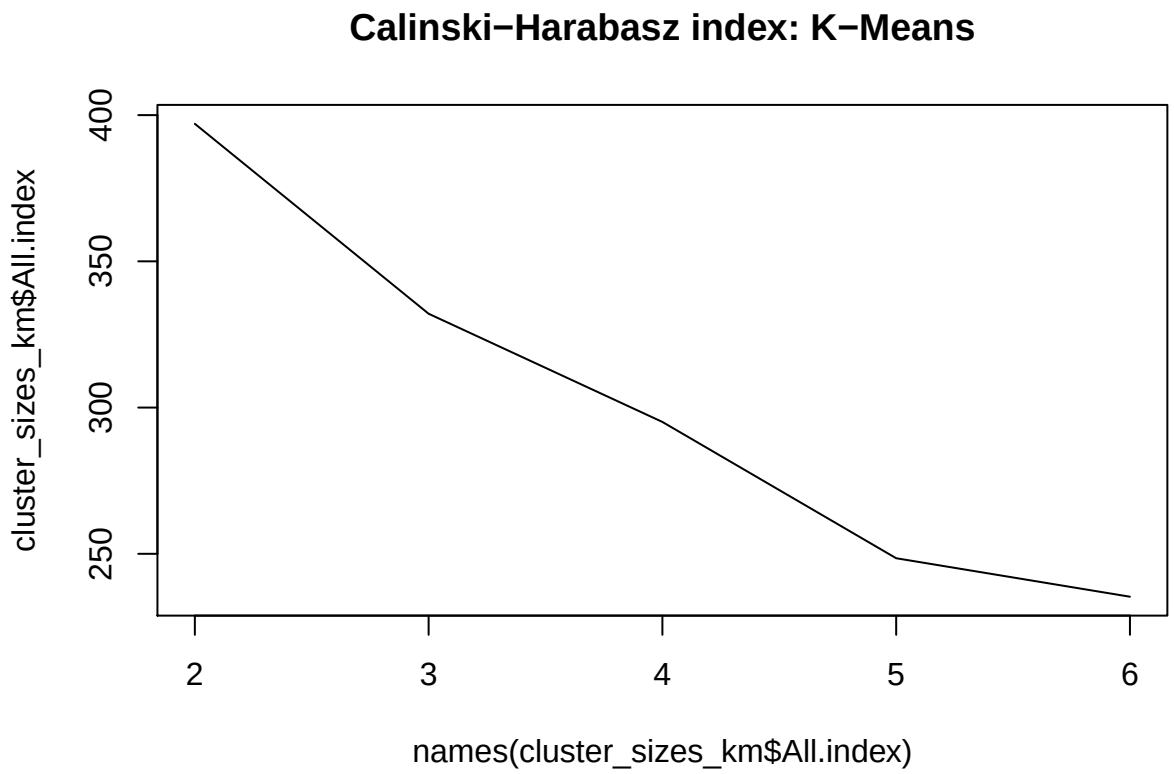
```
# averages by cluster
nba_km_five_avg <- data.frame(nba
                              %>% select(km_labs_five, MP_pg, PTS_pg, TRB_pg,
                                         AST_pg, BLK_pg, STL_pg, VORP, PER, RPM)
                              %>% group_by(km_labs_five)
                              %>% summarise_all(list(mean))
                              )
nba_km_five_avg
```

```
##   km_labs_five     MP_pg    PTS_pg   TRB_pg    AST_pg    BLK_pg    STL_pg
## 1            1  9.005583  2.676594 1.534546 0.6595486 0.1629359 0.2634598
## 2            2 26.672311 11.603424 7.912464 1.5598756 1.0899728 0.7592654
## 3            3 27.845685 11.871305 3.717884 3.0390405 0.3230775 0.9282767
## 4            4 18.438826  6.430108 3.301966 1.3364509 0.3624506 0.5566981
## 5            5 34.117211 22.585572 5.959433 4.8481993 0.6292845 1.1842273
##          VORP       PER        RPM
## 1 -0.08962963  9.084444 -1.9914074
## 2  1.37555556 17.795556  0.4668889
## 3  0.72755102 13.577551 -0.4542857
## 4  0.15792683 12.219512 -1.5098171
## 5  3.47045455 21.961364  2.7727273
```

Optimize number of clusters

Method: Calinski-Harabasz index

```
# get optimal cluster sizes
cluster_sizes_km <- NbClust(data = nba_feat_sc,
                            # it will likely be harder to interpret clusters
                            # past this amount
                            max.nc = 6,
                            method = 'kmeans',
                            index = 'ch')

# plot C(G)
plot(names(cluster_sizes_km$All.index),
     cluster_sizes_km$All.index,
     main = 'Calinski-Harabasz index: K-Means',
     type = 'l')
```




```
# get best number of clusters
cluster_sizes_km$Best.nc
```

```
## Number_clusters     Value_Index
##          2.0000        397.0208
```

```
# show all indices
cluster_sizes_km$All.index
```

```
##        2        3        4        5        6
## 397.0208 332.0847 295.0862 248.4888 235.3342
```

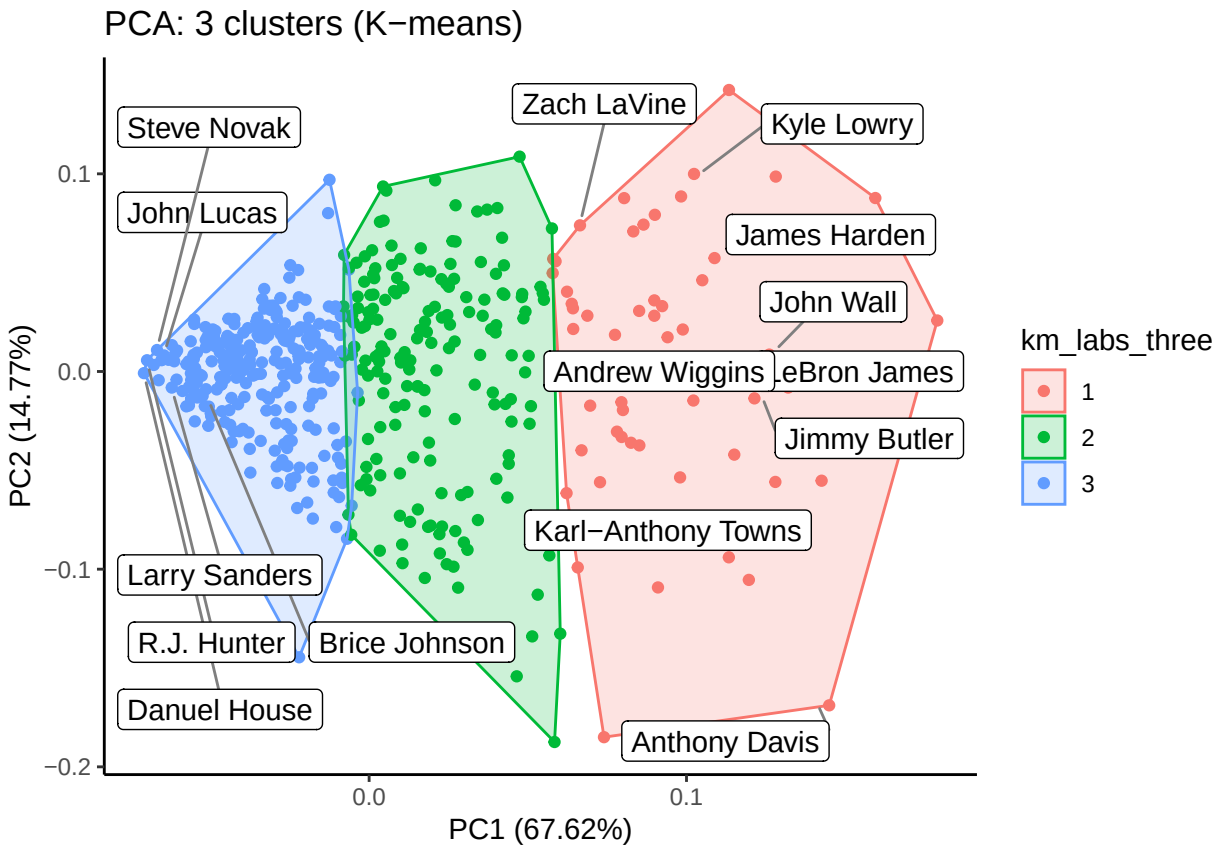
Observations Although the CH index indicates that the optimal number of clusters is 2, this seems too low of a number to meaningfully break out the NBA players into groups. It is also important to note that the CH index is a heuristic method. So although CH is a good approach to look for the number of clusters, it is

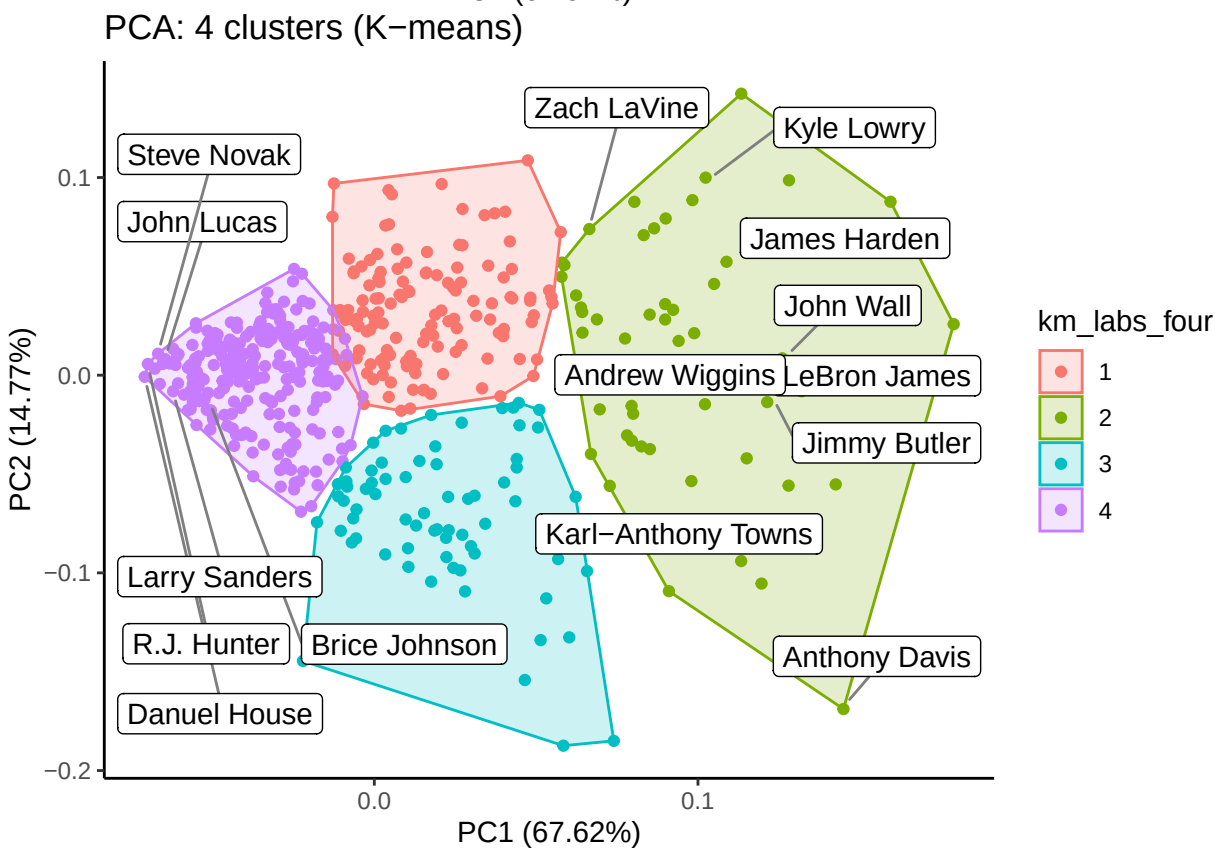
important to combine this with our practical goal of looking for underlying patterns in the players. Thus, I think a more reasonable number to understand the data is with 3 - 4 clusters, which show the second and third best partitions based on the CH index. We will look at both and determine which one is a better fit for our goal.

Try 3 and 4 clusters (K-Means)

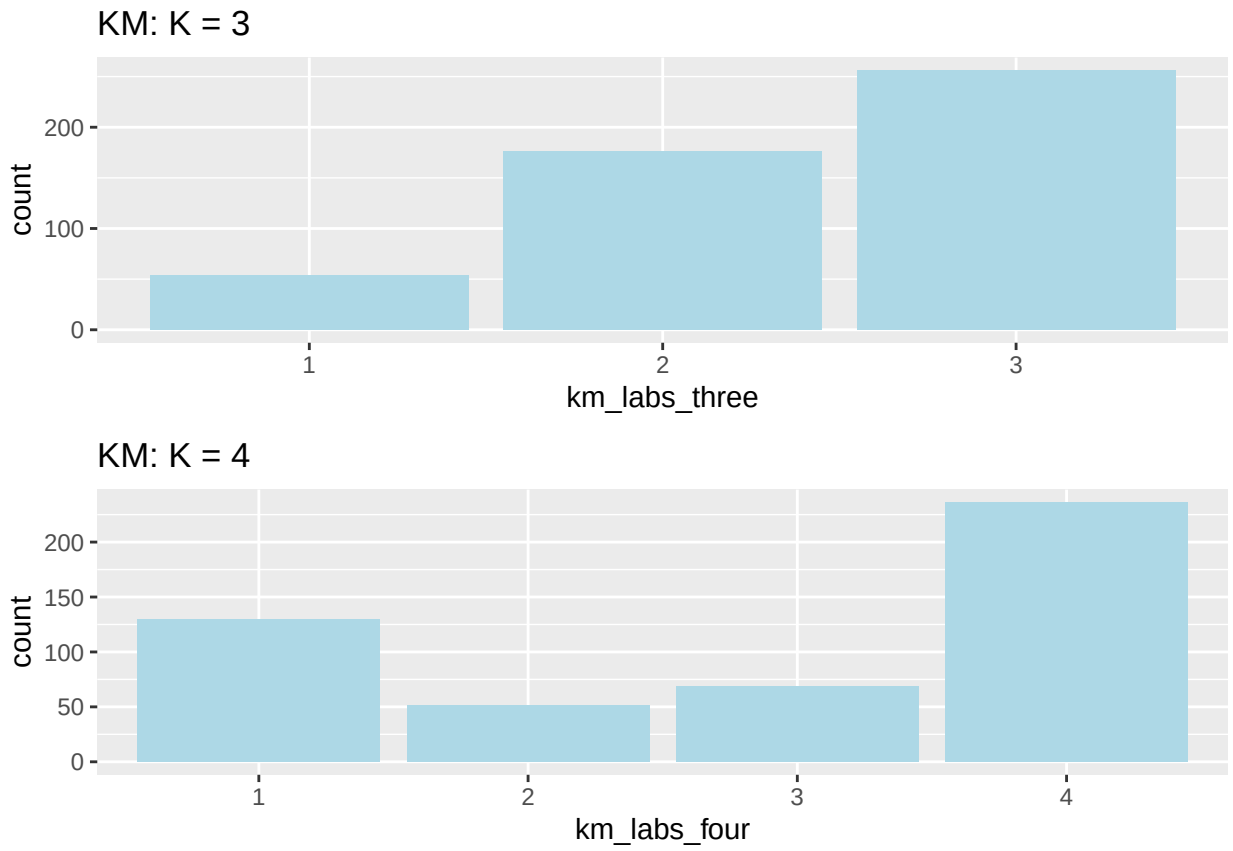
```
# number of clusters to try
num_clust <- c(3, 4)

# store cluster results
clust_list <- list(rep(NA, length = length(num_clust)))
for (i in seq_along(num_clust)) {
  clust_list[[i]] <- kmeans(x = nba_feat_sc,
                       centers = num_clust[i],
                       nstart = 100,
                       algorithm = 'Hartigan-Wong')

  }
```

```
# add cluster labels to main data
nba$km_labs_three <- factor(clust_list[[1]]$cluster)
nba$km_labs_four <- factor(clust_list[[2]]$cluster)
# elements to loop over
km_labs <- names(nba %>% select(tail(names(.), 2)))
km_ks <- c(3, 4)

# plot k-means clusters in PC space
for (i in seq_along(clust_list)) {
  p <- plot_pca(clust_list[[i]], data = nba, frame = TRUE, colour = km_labs[i],
          title = paste0('PCA: ', km_ks[i], ' clusters (K-means)'),
          label = km_labels)
  print(p)
}
```


PCA: 3 clusters (K−means)
Steve Novak
Zach LaVine
Kyle Lowry
John Lucas
James Harden
John Wall
Andrew Wiggins
LeBron James
Jimmy Butler
Karl−Anthony Towns
Larry Sanders
R.J. Hunter
Brice Johnson
Danuel House
Anthony Davis
PC2 (14.77%)
PC1 (67.62%)
0.1
0.0
−0.1
−0.2
km_labs_three
1
2
3


PCA: 4 clusters (K−means)
Steve Novak
Zach LaVine
Kyle Lowry
John Lucas
James Harden
John Wall
Andrew Wiggins
LeBron James
Jimmy Butler
Karl−Anthony Towns
Larry Sanders
R.J. Hunter
Brice Johnson
Danuel House
Anthony Davis
PC2 (14.77%)
PC1 (67.62%)
0.1
0.0
−0.1
−0.2
km_labs_four
1
2
3
4

```
# number of players in each cluster
p_three <- ggplot(data = nba, aes(x = km_labs_three)) +
          geom_bar(fill = 'lightblue') + ggtitle('KM: K = 3')
p_four <- ggplot(data = nba, aes(x = km_labs_four)) +
          geom_bar(fill = 'lightblue') + ggtitle('KM: K = 4')

# combine
cowplot::plot_grid(p_three, p_four, nrow = 2)
```



We may be able to get more insight with four clusters instead of three. We will sample players from each cluster and look at cluster averages.

```
# averages by cluster
nba_clust_avg <- data.frame(nba
                            %>% select(km_labs_four, MP_pg, PTS_pg, TRB_pg,
                                       AST_pg, BLK_pg, STL_pg)
                            %>% group_by(km_labs_four)
                            %>% summarise_all(list(mean))
                            )
nba_clust_avg
```

```
##   km_labs_four    MP_pg   PTS_pg   TRB_pg    AST_pg    BLK_pg    STL_pg
## 1            1 25.61995 10.30825 3.484223 2.5209841 0.2885179 0.8214061
## 2            2 33.85906 21.82156 5.724658 4.7255814 0.6158290 1.1623242
## 3            3 24.96591 10.29999 7.010810 1.6435808 0.9473312 0.7868895
## 4            4 12.24558  3.94709 2.129925 0.8788919 0.2354097 0.3548853
```

```r
# sample players in each cluster
for (k in 1:4) {
  print(paste0('Clusters: ', k))
  print(sample(subset(nba, km_labs_four == k)$Player, size = sample_size))
}
```

```
## [1] "Clusters: 1"
##  [1] Jeff Green      Brandon Knight  Khris Middleton Jeremy Lamb
##  [5] Dirk Nowitzki   J.J. Redick     Gary Harris     Victor Oladipo
##  [9] Danny Green     Jae Crowder
## 486 Levels: A.J. Hammons Aaron Brooks Aaron Gordon ... Zaza Pachulia
## [1] "Clusters: 2"
##  [1] Anthony Davis         Kawhi Leonard         C.J. McCollum
##  [4] Jrue Holiday          Harrison Barnes       Kevin Durant
##  [7] Kristaps Porzingis    James Harden          George Hill
## [10] Giannis Antetokounmpo
## 486 Levels: A.J. Hammons Aaron Brooks Aaron Gordon ... Zaza Pachulia
## [1] "Clusters: 3"
##  [1] Tyson Chandler          Pau Gasol               Timofey Mozgov
##  [4] Lucas Nogueira          Alan Williams           Kenneth Faried
##  [7] Dwight Howard           Hassan Whiteside        Rondae Hollis-Jefferson
## [10] Kyle O'Quinn
## 486 Levels: A.J. Hammons Aaron Brooks Aaron Gordon ... Zaza Pachulia
## [1] "Clusters: 4"
##  [1] Roy Hibbert          Justin Anderson      Cheick Diallo
##  [4] Georgios Papagiannis Michael Gbinije      Mike Miller
##  [7] Aaron Brooks         Kyle Anderson        Tiago Splitter
## [10] Josh McRoberts
## 486 Levels: A.J. Hammons Aaron Brooks Aaron Gordon ... Zaza Pachulia
```

Based on the plot, cluster distributions, and group averages, it looks like 4 clusters is optimal. One main reason is that there is more separation vs 4 clusters, which can provide more value when bucketing players by overall skillsets. Across the statistics, it looks like the clusters are broken out into the following: Best players (2), Good players with more assists, i.e. guards (1), good players who rebound and block more, .e.g forwards (4), and Low-performing players (3). We will now try model-based clustering as a third method.

Model-Based Clustering

```r
# run model-based clustering
# nba_mcl <- Mclust(nba_feat_sc)
# summary(nba_mcl)
```

PITFALL: more minutes per game tends towards higher stats per game.. redo metric to per 36 min instead? However, this has its fallbacks too.. Ex: player who played for 2 min with 2 pts and 2 reb would transalte to 18 pts and 18 reb per 36 min..

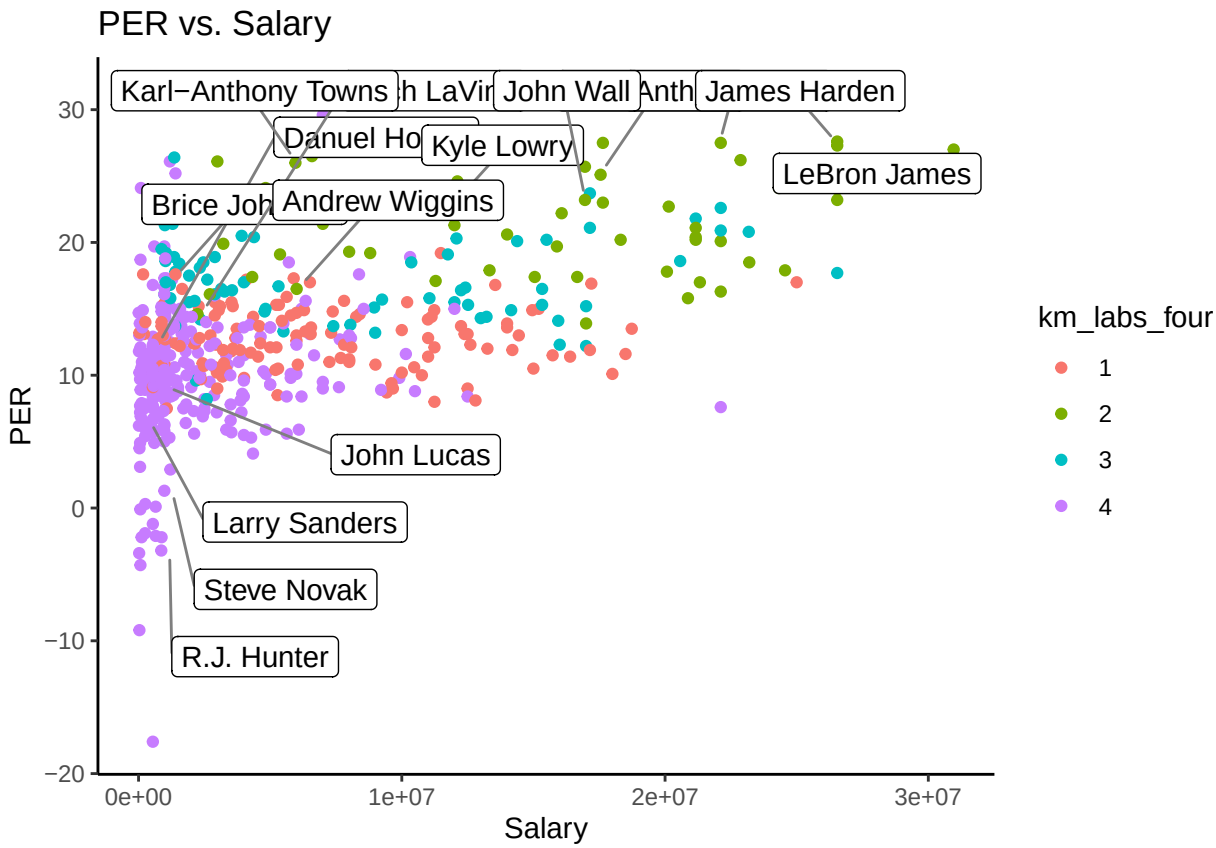
Post-Cluster Analysis

We will now look at different statistics and demographics to see how the clustering lines up

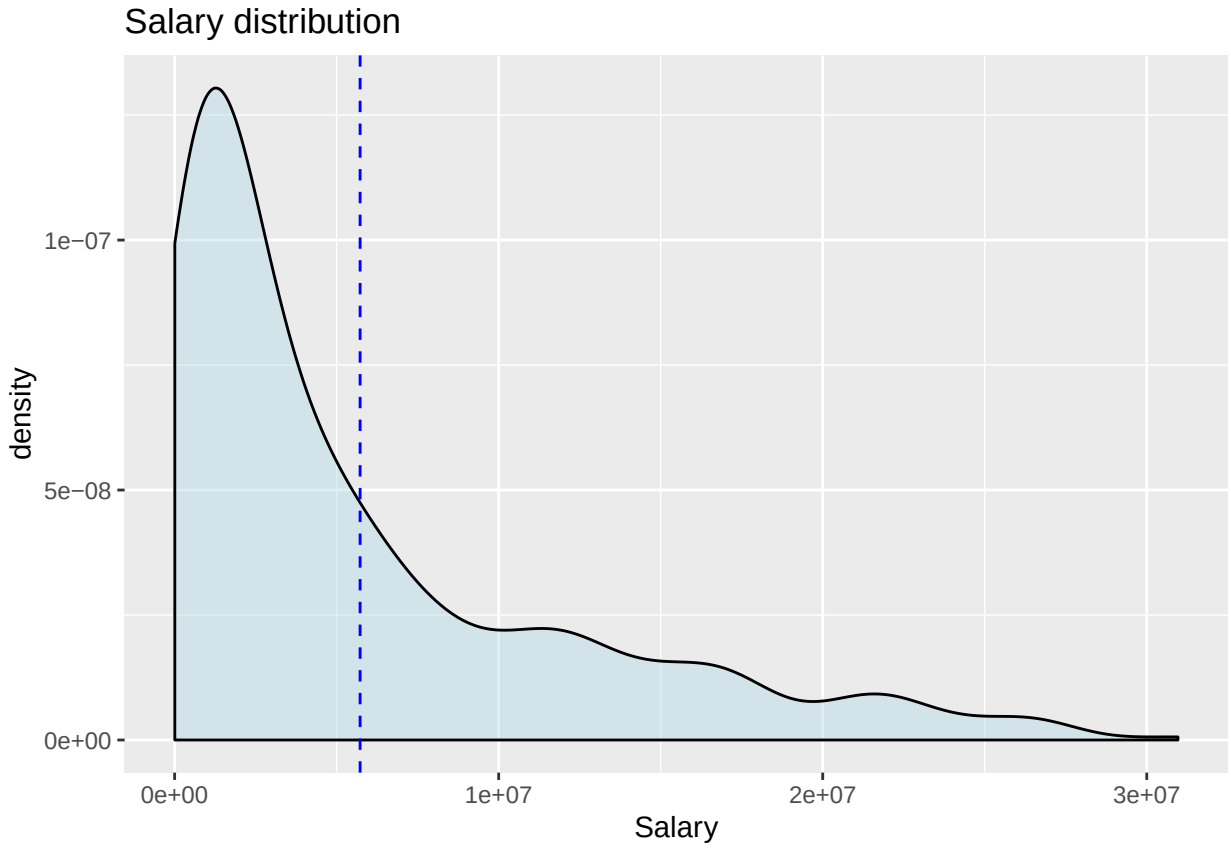
Clusters vs. Player Salaries

Try: K-means, 4 clusters

```
# salary vs. advanced stats, overlayed with clusters

# cluster label to use
cl_label <- "km_labs_four"

# plot PER vs. salary
ggplot(data = nba, aes(x = Salary, y = PER)) +
  geom_point(aes_string(color = cl_label)) +
  geom_label_repel(aes(label = labels_pca),
                   box.padding   = 0.35,
                   point.padding = 0.5,
                   segment.color = 'grey50') +
  ggtitle('PER vs. Salary') +
  theme_classic()
```




```
# plot salary distribution
ggplot(nba, aes(x = Salary)) +
  geom_density(fill = 'lightblue', alpha = 0.4) +
  ggtitle('Salary distribution') +
  geom_vline(xintercept = mean(nba$Salary),
             color = "blue",
             linetype = 'dashed')
```




```
# salary statistics
summary(nba$Salary)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##     5767   980431  2994230  5724075  8064256 30963450
```

```
# Highest Paid players in Lowest Tier
data.frame(nba
           %>% select(Player, G, MP_pg, Tm,
                      Salary, PER, cl_label)
           %>% filter(km_labs_four == 1)
           %>% arrange(desc(Salary))
           )
```

```
##                    Player  G    MP_pg  Tm   Salary  PER km_labs_four
## 1             Dirk Nowitzki 54 26.37037 DAL 25000000 17.0            1
## 2             Ryan Anderson 72 29.38889 HOU 18735364 13.5            1
## 3              Allen Crabbe 79 28.53165 POR 18500000 11.6            1
## 4                 Luol Deng 56 26.53571 LAL 18000000 10.1            1
## 5             Tobias Harris 82 31.30488 DET 17200000 16.9            1
## 6           Wesley Matthews 73 34.17808 DAL 17145838 11.9            1
## 7              Evan Turner 65 25.50769 POR 16393443 11.4            1
## 8             Kent Bazemore 73 26.89041 ATL 15730338 11.5            1
## 9           Khris Middleton 29 30.65517 MIL 15200000 15.0            1
## 10               Jeff Green 69 22.23188 ORL 15000000 10.5            1
## 11            Reggie Jackson 52 27.38462 DET 14956522 14.9            1
## 12               Tony Parker 63 25.19048 SAS 14445313 13.0            1
## 13          DeMarre Carroll 72 26.13889 TOR 14200000 11.9            1
```

```
## 14             Manu Ginobili 69 18.71014 SAS 14000000 13.9           1
## 15               Rajon Rondo 69 26.71014 CHI 14000000 13.6           1
## 16               Ricky Rubio 75 32.92000 MIN 13550000 16.8           1
## 17             Jamal Crawford 82 26.30488 LAC 13253012 12.0           1
## 18                 J.R. Smith 41 28.95122 CLE 12800000  8.1           1
## 19            Brandon Knight 54 21.11111 PHO 12606250 12.3           1
## 20             Arron Afflalo 61 25.90164 SAC 12500000  9.0           1
## 21           Jordan Clarkson 82 29.23171 LAL 12500000 13.1           1
## 22              Eric Gordon 75 30.97333 HOU 12385364 13.1           1
## 23           Marvin Williams 76 30.19737 CHO 12250000 13.7           1
## 24                Jeremy Lin 36 24.52778 BRK 11483254 19.2           1
## 25              Courtney Lee 77 31.93506 NYK 11242000 12.1           1
## 26              Solomon Hill 80 29.67500 NOP 11241218  8.0           1
## 27            Wilson Chandler 71 30.94366 DEN 11233146 14.9           1
## 28            Andre Iguodala 76 26.28947 GSW 11131368 14.4           1
## 29               Joe Johnson 78 23.62821 UTA 11000000 12.8           1
## 30             Austin Rivers 74 27.75676 LAC 11000000 11.4           1
## 31                 Jon Leuer 75 25.92000 DET 10991957 14.2           1
## 32               Monta Ellis 74 27.00000 IND 10763500 10.0           1
## 33              Jared Dudley 64 21.28125 PHO 10470000 10.6           1
## 34              Tyreke Evans 40 19.70000 NOP 10203755 15.5           1
## 35               Danny Green 68 26.57353 SAS 10000000 10.2           1
## 36             Terrence Ross 78 25.06410 TOR 10000000 13.4           1
## 37             Iman Shumpert 76 25.48684 CLE  9662922  9.0           1
## 38       Matthew Dellavedova 76 26.13158 MIL  9607500  9.4           1
## 39            Jerryd Bayless  3 23.66667 PHI  9424084  8.7           1
## 40          Gerald Henderson 72 23.15278 PHI  9000000 10.8           1
## 41            Ersan Ilyasova 82 26.12195 PHI  8400000 14.6           1
## 42             Avery Bradley 55 33.36364 BOS  8269663 14.4           1
## 43             E'Twaun Moore 73 24.93151 NOP  8081363 12.1           1
## 44          Sergio Rodriguez 68 22.32353 PHI  8000000 11.0           1
## 45            Garrett Temple 65 26.58462 SAC  8000000 11.2           1
## 46          Anthony Tolliver 65 22.72308 SAC  8000000 11.1           1
## 47              Trevor Ariza 80 34.66250 HOU  7806971 12.3           1
## 48             Channing Frye 74 18.89189 CLE  7806971 15.6           1
## 49           Al-Farouq Aminu 61 29.06557 POR  7680965 11.3           1
## 50               J.J. Redick 78 28.17949 LAC  7377500 14.8           1
## 51               Cory Joseph 80 25.03750 TOR  7315000 13.2           1
## 52             D.J. Augustin 78 19.71795 ORL  7250000 11.0           1
## 53            Victor Oladipo 67 33.16418 OKC  6552961 13.6           1
## 54               Jodie Meeks 36 20.50000 ORL  6540000 13.1           1
## 55               Jeremy Lamb 62 18.43548 CHO  6511628 17.0           1
## 56            Marco Belinelli 74 24.02703 CHO  6333333 13.3           1
## 57               Jae Crowder 72 32.43056 BOS  6286408 14.9           1
## 58          Patrick Patterson 65 24.60000 TOR  6050000 10.8           1
## 59           Patrick Beverley 67 30.71642 HOU  6000000 13.0           1
## 60            Wayne Ellington 62 24.19355 MIA  6000000 12.6           1
## 61                  Ish Smith 81 24.13580 DET  6000000 14.7           1
## 62               Otto Porter 80 32.56250 WAS  5893981 17.3           1
## 63             Nikola Mirotic 70 23.98571 CHI  5782450 14.5           1
## 64              Tyler Johnson 73 29.83562 MIA  5628000 15.9           1
## 65                 Nick Young 60 25.93333 LAL  5443918 14.1           1
## 66           D'Angelo Russell 63 28.74603 LAL  5332800 15.3           1
## 67                P.J. Tucker 81 27.60494 PHO  5300000 10.5           1
```

```
## 68            Brandon Ingram 79 28.84810 LAL  5281680  8.5           1
## 69               Kyle Korver 67 26.16418 ATL  5239437 12.1           1
## 70           Darren Collison 68 30.33824 SAC  5229454 15.3           1
## 71         Langston Galloway 74 20.20270 NOP  5200000 10.4           1
## 72          Brandon Jennings 81 22.24691 NYK  5000000 12.1           1
## 73             Marcus Morris 79 32.46835 DET  4625000 12.4           1
## 74                C.J. Miles 76 23.36842 IND  4583450 13.7           1
## 75             Jameer Nelson 75 27.26667 DEN  4540525 11.4           1
## 76              Aaron Gordon 80 28.72500 ORL  4351320 14.4           1
## 77              Vince Carter 73 24.64384 MEM  4264057 11.7           1
## 78                J.J. Barea 35 22.02857 DAL  4096950 17.2           1
## 79              Ben McLemore 61 19.27869 SAC  4008882  9.8           1
## 80           Thabo Sefolosha 62 25.74194 ATL  3850000 11.9           1
## 81          Bojan Bogdanovic 81 25.71605 BRK  3730653 13.5           1
## 82  Kentavious Caldwell-Pope 76 33.27632 DET  3678319 12.8           1
## 83                Patty Mills 80 21.92500 SAS  3578948 15.2           1
## 84              Marcus Smart 79 30.36709 BOS  3578880 12.0           1
## 85               Will Barton 60 28.41667 DEN  3533333 15.5           1
## 86               Buddy Hield 82 23.02439 NOP  3517200 11.8           1
## 87              Troy Daniels 67 17.65672 MEM  3332940 10.4           1
## 88           Emmanuel Mudiay 55 25.56364 DEN  3241800 10.9           1
## 89              Jamal Murray 82 21.51220 DEN  3210840 11.9           1
## 90   Michael Carter-Williams 45 18.80000 CHI  3183526  9.9           1
## 91              Kelly Olynyk 75 20.50667 BOS  3094014 15.2           1
## 92          Shabazz Muhammad 78 19.43590 MIN  3046299 14.9           1
## 93              Nik Stauskas 80 27.35000 PHI  2993040  9.0           1
## 94          Dante Cunningham 66 24.98485 NOP  2978250 10.2           1
## 95           Marquese Chriss 82 21.25610 PHO  2941440 12.3           1
## 96                Seth Curry 70 28.98571 DAL  2898000 15.5           1
## 97               James Ennis 64 23.45312 MEM  2898000 10.6           1
## 98              Dion Waiters 46 30.08696 MIA  2898000 14.5           1
## 99             Frank Kaminsky 75 26.05333 CHO  2730000 13.0           1
## 100           Doug McDermott 66 22.84848 CHI  2483040 10.7           1
## 101             Shelvin Mack 55 21.90909 UTA  2433334 10.9           1
## 102               Tony Snell 80 29.20000 MIL  2368327  9.7           1
## 103              Dario Saric 81 26.28395 PHI  2318280 12.8           1
## 104             Tim Hardaway 79 27.26582 ATL  2281605 15.2           1
## 105               Joe Ingles 82 24.04878 UTA  2150000 12.4           1
## 106              Tim Frazier 65 23.46154 NOP  2090000 12.4           1
## 107              Gary Harris 57 31.26316 DEN  1655880 16.5           1
## 108             Caris LeVert 57 21.70175 BRK  1562280 12.2           1
## 109              Rodney Hood 59 27.00000 UTA  1406520 12.4           1
## 110        Marreese Speights 82 15.68293 LAC  1403611 17.6           1
## 111         Isaiah Whitehead 73 22.50685 BRK  1074145  7.5           1
## 112         Robert Covington 67 31.62687 PHI  1015696 13.2           1
## 113           Justin Holiday 82 19.98780 NYK  1015696 12.7           1
## 114            Jordan Farmar  2 17.50000 SAC   980431 14.4           1
## 115           Raymond Felton 80 21.25000 LAC   980431 10.9           1
## 116               Joe Harris 52 21.88462 BRK   980431  9.0           1
## 117          Sean Kilpatrick 70 25.05714 BRK   980431 13.1           1
## 118                Ty Lawson 69 25.10145 SAC   980431 15.4           1
## 119          Malcolm Brogdon 75 26.42667 MIL   925000 14.9           1
## 120               Tyler Ulis 61 18.40984 PHO   918369 13.0           1
## 121           T.J. McConnell 81 26.33333 PHI   874636 13.7           1
```

```
## 122            Norman Powell 76 18.00000 TOR   874636 14.0            1
## 123          Josh Richardson 53 30.45283 MIA   874636 10.7            1
## 124        Spencer Dinwiddie 59 22.61017 BRK   726672 12.7            1
## 125          Rodney McGruder 78 25.20513 MIA   543471  9.1            1
## 126          Deron Williams 64 25.89062 DAL   259526 14.0            1
## 127             Matt Barnes 74 24.01351 SAC   242224 10.3            1
## 128            Yogi Ferrell 46 26.02174 DAL   207798 13.1            1
## 129          Jordan Crawford 19 23.26316 NOP   173099 17.6            1
## 130           Alex Poythress  6 26.16667 PHI    35166 13.2            1
```

```
# Lowest Paid players in highest tier
data.frame(nba
           %>% select(Player, G, MP_pg, Tm, Salary, PER, cl_label)
           %>% filter(km_labs_four == 4)
           %>% arrange(Salary)
           )
```

```
##                    Player  G     MP_pg  Tm   Salary   PER km_labs_four
## 1            Dahntay Jones  1 12.000000 CLE     5767  14.7            4
## 2             Axel Toupane  4 11.750000 NOP    15435   6.2            4
## 3               Quinn Cook 14 13.428571 NOP    15984  11.8            4
## 4            Elijah Millsap  2 11.500000 PHO    23069  -3.4            4
## 5          Patricio Garino  5  8.600000 ORL    31969  -9.2            4
## 6       Marcus Georges-Hunt  5  9.600000 ORL    31969  10.2            4
## 7            Pierre Jackson  8 10.500000 DAL    31969  13.0            4
## 8              Gary Payton  6 16.500000 MIL    35116   4.5            4
## 9            Jarrod Uthoff  9 12.777778 DAL    47953  13.9            4
## 10           Anthony Brown 11 14.454545 NOP    57672   7.2            4
## 11            Jarell Eddie  5 12.400000 PHO    57672   9.7            4
## 12              Alonzo Gee 13  6.846154 DEN    57672   3.1            4
## 13           Justin Harper  3 10.333333 PHI    57672   4.9            4
## 14            Jarrett Jack  2 16.500000 NOP    57672   7.7            4
## 15              Mike Tobey  2 12.500000 CHO    67135  -0.1            4
## 16               Gary Neal  2  9.000000 ATL    72193  -4.3            4
## 17             David Nwaba 20 19.850000 LAL    73528  12.1            4
## 18          Archie Goodwin 15 14.266667 BRK    75000  18.7            4
## 19           Troy Williams 30 18.566667 MEM    76725   8.9            4
## 20            Wayne Selden 14 16.857143 MEM    83119   6.9            4
## 21              Shawn Long 18 13.000000 PHI    89513  24.1            4
## 22            Manny Harris  4  6.250000 DAL   115344  -2.2            4
## 23         Hollis Thompson 40 18.800000 PHI   115344   7.9            4
## 24        Derrick Williams 50 16.080000 CLE   115344  10.6            4
## 25           Briante Weber 20 10.250000 CHO   128623  11.0            4
## 26              Omri Casspi 36 17.861111 SAC   138414   9.9            4
## 27          Chasson Randle 26 11.500000 NYK   143860  13.6            4
## 28         Johnny O'Bryant 11  7.272727 DEN   161483  14.9            4
## 29            Larry Sanders  5  2.600000 CLE   207722   6.5            4
## 30              Okaro White 35 13.457143 MIA   210995   7.5            4
## 31            Andrew Bogut 27 21.592593 DAL   242224   9.3            4
## 32         Lamar Patterson  5  8.000000 ATL   246956  -1.9            4
## 33           Jose Calderon 41 13.146341 LAL   247991   8.9            4
## 34             Norris Cole 13  9.615385 OKC   247991   5.4            4
## 35           Isaiah Taylor  4 13.000000 HOU   255000   0.3            4
## 36            Ronnie Price 14  9.571429 PHO   276828   5.9            4
## 37              Ryan Kelly 16  6.875000 ATL   286785   7.8            4
```

```
## 38            Joel Anthony 19  6.421053 SAS   346034  11.6           4
## 39           Toney Douglas 24 16.416667 MEM   379159  10.6           4
## 40         Jonathan Gibson 17 13.588235 DAL   469943   9.5           4
## 41               Ron Baker 52 16.480769 NYK   543471   7.5           4
## 42              Ben Bentil  3  3.333333 DAL   543471 -17.6           4
## 43           Davis Bertans 67 12.059701 SAS   543471  12.9           4
## 44         Nicolas Brussino 54  9.648148 DAL   543471  10.7           4
## 45           Semaj Christon 64 15.203125 OKC   543471   5.7           4
## 46           Cheick Diallo 17 11.705882 NOP   543471  16.8           4
## 47              Kay Felder 42  9.190476 CLE   543471  11.2           4
## 48     Dorian Finney-Smith 81 20.271605 DAL   543471   7.7           4
## 49             Bryn Forbes 36  7.916667 SAS   543471   5.9           4
## 50          Treveon Graham 27  7.000000 CHO   543471  10.6           4
## 51            Danuel House  1  1.000000 WAS   543471  12.2           4
## 52           Derrick Jones 32 17.031250 PHO   543471  12.0           4
## 53    Nicolas Laprovittola 18  9.666667 SAS   543471   8.4           4
## 54           Patrick McCaw 71 15.126761 GSW   543471   8.6           4
## 55       Sheldon McClellan 30  9.566667 WAS   543471  10.1           4
## 56           Maurice Ndour 32 10.343750 NYK   543471  11.3           4
## 57           Daniel Ochefu 19  3.947368 WAS   543471   6.6           4
## 58          Chinanu Onuaku  5 10.400000 HOU   543471  12.3           4
## 59        Marshall Plumlee 21  8.095238 NYK   543471  10.9           4
## 60          Tim Quarterman 16  5.000000 POR   543471  10.2           4
## 61           Diamond Stone  7  3.428571 LAC   543471  -1.2           4
## 62           Fred VanVleet 37  7.945946 TOR   543471  10.5           4
## 63            Kyle Wiltjer 14  3.142857 HOU   543471   6.7           4
## 64      Donatas Motiejunas 34 14.088235 NOP   576724   9.2           4
## 65           Joel Bolomboy 12  4.416667 UTA   600000  19.7           4
## 66            Jake Layman 35  7.114286 POR   600000   4.9           4
## 67         Michael Gbinije  9  3.555556 DET   650000  -2.1           4
## 68            A.J. Hammons 22  7.409091 DAL   650000   8.4           4
## 69           Georges Niang 23  4.043478 IND   650000   0.1           4
## 70             Bobby Brown 25  4.920000 HOU   680534  10.8           4
## 71             Paul Zipser 44 19.159091 CHI   750000   6.9           4
## 72             R.J. Hunter  3  3.000000 CHI   864346  -3.2           4
## 73        Pat Connaughton 39  8.102564 POR   874636  11.8           4
## 74       Cristiano Felicio 66 15.757576 CHI   874636  15.2           4
## 75          Aaron Harrison  5  3.400000 CHO   874636  -2.2           4
## 76         Darrun Hilliard 39  9.769231 DET   874636   5.9           4
## 77            Jordan McRae 37 10.378378 CLE   874636   9.7           4
## 78             Salah Mejri 73 12.397260 DAL   874636  14.8           4
## 79        Jonathon Simmons 78 17.846154 SAS   874636   9.9           4
## 80          Christian Wood 13  8.230769 CHO   874636  15.1           4
## 81         Reggie Williams  6 13.166667 NOP   895197  11.7           4
## 82               Raul Neto 40  8.650000 UTA   937800  10.7           4
## 83         Andrew Harrison 72 20.472222 MEM   945000   8.7           4
## 84       Stephen Zimmerman 19  5.684211 ORL   950000   7.3           4
## 85          Chris Andersen 12  9.500000 CLE   980431  11.6           4
## 86           Alan Anderson 30 10.266667 LAC   980431   5.0           4
## 87            Brandon Bass 52 11.096154 LAC   980431  19.7           4
## 88               Ian Clark 77 14.766234 GSW   980431  13.1           4
## 89            Jerami Grant 80 19.137500 OKC   980431  10.1           4
## 90            Gerald Green 47 11.446809 BOS   980431  12.0           4
## 91             James Jones 48  7.937500 CLE   980431  11.3           4
```

```
## 92              John Lucas  5  2.200000 MIN   980431   9.1           4
## 93           James Michael 52  8.788462 GSW   980431  13.0           4
## 94             Steve Novak  8  2.750000 MIL   980431   1.3           4
## 95           Arinze Onuaku  8  3.500000 ORL   980431   5.8           4
## 96           Brian Roberts 41 10.146341 CHO   980431   9.8           4
## 97         Thomas Robinson 48 11.666667 LAL   980431  17.3           4
## 98           Damjan Rudez 45  6.977778 ORL   980431   6.3           4
## 99          Jarnell Stokes  2  3.500000 DEN   980431  31.5           4
## 100            Jason Terry 74 18.445946 MIL   980431   9.0           4
## 101        Marcus Thornton 33 17.424242 WAS   980431  10.4           4
## 102             Beno Udrih 39 14.358974 DET   980431  16.1           4
## 103       Anderson Varejao 14  6.571429 GSW   980431   9.4           4
## 104          Sasha Vujacic 42  9.714286 NYK   980431   8.6           4
## 105             David West 68 12.558824 GSW   980431  16.6           4
## 106            Metta World 25  6.400000 LAL   980431   6.2           4
## 107        Anthony Bennett 23 11.478261 BRK  1015696  14.7           4
## 108          Isaiah Canaan 39 15.179487 CHI  1015696   8.1           4
## 109        DeAndre Liggins 62 12.532258 CLE  1015696   7.5           4
## 110           Mike Muscala 70 17.671429 ATL  1015696  14.4           4
## 111            Willie Reed 71 14.521127 MIA  1015696  17.1           4
## 112            Jeff Withey 51  8.470588 UTA  1015696  18.8           4
## 113         Glenn Robinson 69 20.666667 IND  1050500  11.5           4
## 114           John Jenkins  4  3.250000 PHO  1050961  17.3           4
## 115       Rakeem Christmas 29  7.551724 IND  1052342  10.4           4
## 116              Joe Young 33  4.090909 IND  1052342  11.4           4
## 117           Damian Jones 10  8.500000 GSW  1171560   5.3           4
## 118        Dejounte Murray 38  8.473684 SAS  1180080   9.6           4
## 119           Kevon Looney 53  8.433962 GSW  1182840  13.4           4
## 120           Josh Huestis  2 15.500000 OKC  1191480  26.1           4
## 121       Chris McCullough 16  5.000000 BRK  1191480  15.2           4
## 122          Kyle Anderson 72 14.166667 SAS  1192080  12.5           4
## 123          Pascal Siakam 55 15.618182 TOR  1196040  11.5           4
## 124            C.J. Wilcox 22  4.909091 ORL  1209600   2.9           4
## 125          Jordan Mickey 25  5.640000 BOS  1223653   9.8           4
## 126           Luke Babbitt 68 15.661765 MIA  1227286   8.4           4
## 127          Brice Johnson  3  3.000000 LAC  1273920  17.2           4
## 128          Jarell Martin 42 13.285714 MEM  1286160   8.7           4
## 129 Timothe Luwawu-Cabarrot 69 17.246377 PHI  1326960   8.5           4
## 130              Tyus Jones 60 12.900000 MIN  1339680  13.8           4
## 131         Shabazz Napier 53  9.660377 POR  1350120  13.6           4
## 132          Deyonta Davis 36  6.611111 MEM  1369299  10.6           4
## 133           JaVale McGee 77  9.597403 GSW  1403611  25.2           4
## 134     Malachi Richardson 22  9.000000 SAC  1439800   9.6           4
## 135      Demetrius Jackson  5  3.400000 BOS  1450000  30.8           4
## 136           Bobby Portis 64 15.625000 CHI  1453680  14.9           4
## 137       DeAndre' Bembry 38  9.763158 ATL  1499760   8.8           4
## 138        Marcelo Huertas 23 10.304348 LAL  1500000   9.1           4
## 139        Justin Anderson 75 16.373333 DAL  1514160  13.9           4
## 140          Delon Wright 27 16.518519 TOR  1577280  15.0           4
## 141          Bruno Caboclo  9  4.444444 TOR  1589640  14.6           4
## 142          Malik Beasley 22  7.500000 DEN  1627320  13.7           4
## 143           Jerian Grant 63 16.317460 CHI  1643040  13.1           4
## 144         Henry Ellenson 19  7.684211 DET  1704120   7.5           4
## 145      Joffrey Lauvergne 70 14.000000 OKC  1709719  12.6           4
```

```
## 146              Sam Dekker 77 18.428571 HOU  1720560  13.0           4
## 147            Tyler Ennis 53 11.094340 LAL  1733880  11.0           4
## 148             Quincy Acy 38 14.684211 BRK  1790092  11.8           4
## 149           Wade Baldwin 33 12.272727 MEM  1793760   6.4           4
## 150         Kevin Seraphin 49 11.408163 IND  1800000  14.4           4
## 151          Rashad Vaughn 41 11.170732 MIL  1811040   7.8           4
## 152            James Young 29  7.586207 BOS  1825200  10.0           4
## 153           Terry Rozier 74 17.067568 BOS  1906440  10.8           4
## 154       Juan Hernangomez 62 13.580645 DEN  1987440  13.3           4
## 155            Kelly Oubre 79 20.316456 WAS  2006640   9.1           4
## 156          Adreian Payne 18  7.500000 MIN  2022240  14.4           4
## 157       Denzel Valentine 57 17.122807 CHI  2092200   7.3           4
## 158          Cameron Payne 31 14.903226 OKC  2112480   5.6           4
## 159   Georgios Papagiannis 22 16.136364 SAC  2202240  12.7           4
## 160               Luc Mbah 80 22.337500 LAC  2203000  10.3           4
## 161         Reggie Bullock 31 15.064516 DET  2255644  11.7           4
## 162  Taurean Waller-Prince 59 16.627119 ATL  2318280   9.8           4
## 163             Trey Lyles 71 16.309859 UTA  2340600  10.0           4
## 164       Domantas Sabonis 81 20.148148 OKC  2440200   6.9           4
## 165        Malcolm Delaney 73 17.095890 ATL  2500000   7.5           4
## 166             Randy Foye 69 18.608696 BRK  2500000   7.3           4
## 167      Richard Jefferson 79 20.430380 CLE  2500000   8.2           4
## 168             Thon Maker 57  9.859649 MIL  2568600  14.0           4
## 169           Aaron Brooks 65 13.753846 IND  2700000   9.5           4
## 170           Jakob Poeltl 54 11.592593 TOR  2703960  12.2           4
## 171            Noah Vonleh 74 17.094595 POR  2751360  10.8           4
## 172       Tomas Satoransky 57 12.614035 WAS  2870813   8.5           4
## 173         Dewayne Dedmon 76 17.500000 SAS  2898000  16.0           4
## 174   Mindaugas Kuzminskas 68 14.941176 NYK  2898000  12.4           4
## 175        Stanley Johnson 77 17.805195 DET  2969880   7.2           4
## 176        Justin Hamilton 64 18.390625 BRK  3000000  13.6           4
## 177         K.J. McDaniels 49 10.306122 BRK  3333333  11.5           4
## 178             Mike Scott 18 10.833333 ATL  3333334   5.9           4
## 179             Trey Burke 57 12.333333 WAS  3386598  10.8           4
## 180         Anthony Morrow 49 14.571429 OKC  3488000  10.0           4
## 181            Mike Miller 20  7.550000 DEN  3500000   7.8           4
## 182           Brandon Rush 47 21.914894 MIN  3500000   6.6           4
## 183            Paul Pierce 25 11.080000 LAC  3527920   5.7           4
## 184          Nick Collison 20  6.400000 OKC  3750000  12.8           4
## 185        Nemanja Bjelica 65 18.307692 MIN  3800000  11.0           4
## 186              Kris Dunn 78 17.089744 MIN  3872520   8.1           4
## 187          Mario Hezonja 65 14.769231 ORL  3909840   7.2           4
## 188             Dante Exum 66 18.606061 UTA  3940320   8.6           4
## 189            Lavoy Allen 61 14.278689 IND  4000000  11.6           4
## 190        Leandro Barbosa 67 14.373134 PHO  4000000  11.5           4
## 191          Udonis Haslem 17  7.647059 MIA  4000000   8.4           4
## 192            Jordan Hill  7  6.714286 MIN  4000000   5.5           4
## 193         Kris Humphries 56 12.303571 ATL  4000000  13.6           4
## 194       Lance Stephenson 18 20.055556 NOP  4000000   9.6           4
## 195           Devin Harris 65 16.723077 DAL  4227996  13.8           4
## 196          Dragan Bender 43 13.348837 PHO  4276320   5.3           4
## 197        Greivis Vasquez  3 13.000000 BRK  4347826   4.1           4
## 198          Alexis Ajinca 39 14.974359 NOP  4638203  12.9           4
## 199           Jaylen Brown 78 17.192308 BOS  4743000  10.3           4
```

```
## 200           Mike Dunleavy 53 15.867925 ATL  4837500  10.1           4
## 201            Kyle Singler 32 12.031250 OKC  4837500   5.9           4
## 202             Roy Hibbert 48 14.208333 CHO  5000000  13.6           4
## 203           Jonas Jerebko 78 15.794872 BOS  5000000   9.3           4
## 204             Jason Smith 74 14.432432 WAS  5000000  13.6           4
## 205             C.J. Watson 62 16.322581 ORL  5000000   9.3           4
## 206              Luis Scola 36 12.805556 BRK  5500000  13.9           4
## 207          Wesley Johnson 68 11.911765 LAC  5628000   8.4           4
## 208         Jared Sullinger 11 10.727273 TOR  5628000   5.6           4
## 209          Brandan Wright 28 15.964286 MEM  5709880  18.5           4
## 210        Shaun Livingston 76 17.697368 GSW  5782450  10.1           4
## 211          Josh McRoberts 22 17.318182 MIA  5782450   9.8           4
## 212            Alex Abrines 68 15.514706 OKC  5994764  10.1           4
## 213          Ramon Sessions 50 16.220000 CHO  6000000  12.3           4
## 214        Andrew Nicholson 38  9.000000 WAS  6088993   5.9           4
## 215             Tarik Black 67 16.283582 LAL  6191000  15.0           4
## 216           Lance Thomas 46 21.043478 NYK  6191000   8.4           4
## 217          Spencer Hawes 54 14.759259 CHO  6348759  15.6           4
## 218            Aron Baynes 75 15.506667 DET  6500000  13.1           4
## 219               Ed Davis 46 17.152174 POR  6666667  11.5           4
## 220             Boris Diaw 73 17.575342 UTA  7000000   9.0           4
## 221        Boban Marjanovic 35  8.371429 DET  7000000  29.6           4
## 222          Rodney Stuckey 39 17.846154 IND  7000000   9.5           4
## 223            Corey Brewer 82 15.621951 HOU  7612172   9.1           4
## 224            Cole Aldrich 62  8.564516 MIN  7643979  12.7           4
## 225            Tyler Zeller 51 10.294118 BOS  8000000  13.0           4
## 226          Darrell Arthur 41 15.585366 DEN  8070175  12.8           4
## 227           Dwight Powell 77 17.311688 DAL  8375000  17.6           4
## 228          Tiago Splitter  8  9.500000 PHI  8550000  15.0           4
## 229          Meyers Leonard 74 16.513514 POR  9213483   8.9           4
## 230               Omer Asik 31 15.548387 NOP  9904494   9.8           4
## 231              Alec Burks 42 15.547619 UTA 10154495  11.6           4
## 232            Al Jefferson 66 14.106061 IND 10314532  18.9           4
## 233         Mirza Teletovic 70 16.185714 MIL 10500000   8.8           4
## 234            Amir Johnson 80 20.100000 BOS 12000000  15.0           4
## 235           Miles Plumlee 45 10.755556 MIL 12500000   8.4           4
## 236        Chandler Parsons 34 19.852941 MEM 22116750   7.6           4
```

Observations Based on the plot and tables, it looks like there is potential to update salaries based on player tiers. For example, Chandler Parsons was paid 22M but is considered a low tier player, and is paid more than the high tier players such as Steph Curry (12M) and Kawhi Leonard (17.6M).

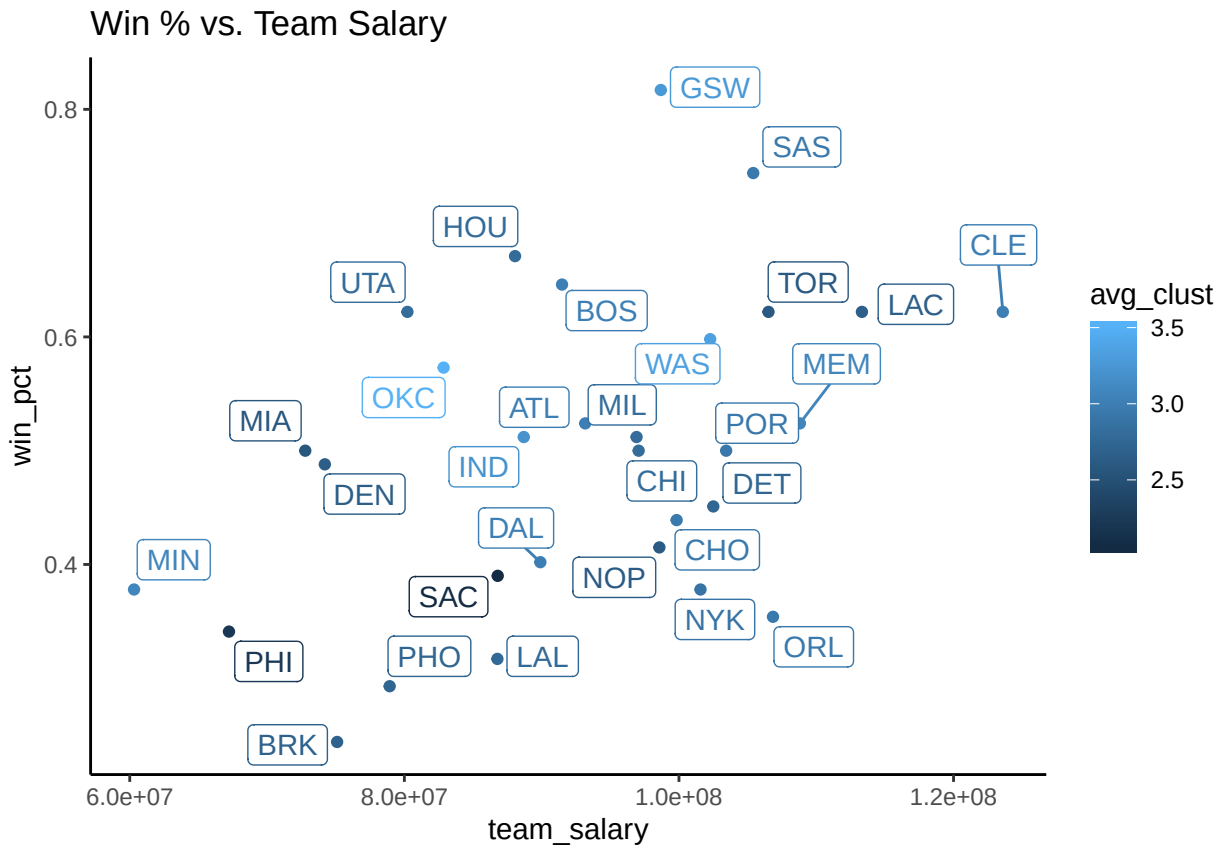
Team Compensation and Performance vs clusters

```
# convert labels to numeric
nba$cl_lab_numeric <- as.numeric(nba[ , cl_label])
nba_team <- data.frame(nba
                       %>% select(Tm, Salary, win_pct, cl_lab_numeric)
                       %>% group_by(Tm)
                       %>% summarise(team_salary = sum(Salary),
                                     win_pct = mean(win_pct),
                                     avg_clust = mean(cl_lab_numeric))
                       )
```

```r
# order by descnding average cluster label
arrange(nba_team, desc(avg_clust))
```

```
##     Tm team_salary win_pct avg_clust
## 1  OKC    82858524   0.573  3.500000
## 2  WAS   102276673   0.598  3.333333
## 3  GSW    98681493   0.817  3.266667
## 4  IND    88698690   0.512  3.200000
## 5  MIN    60311572   0.378  3.071429
## 6  MEM   108808118   0.524  3.058824
## 7  ATL    93172774   0.524  3.000000
## 8  BOS    91484921   0.646  3.000000
## 9  CLE   123591014   0.622  3.000000
## 10 DAL    89904500   0.402  3.000000
## 11 CHO    99830531   0.439  2.941176
## 12 ORL   106849160   0.354  2.941176
## 13 SAS   105410231   0.744  2.937500
## 14 POR   103439444   0.500  2.928571
## 15 NYK   101570502   0.378  2.875000
## 16 MIL    96913241   0.512  2.812500
## 17 CHI    97064073   0.500  2.800000
## 18 UTA    80223193   0.622  2.800000
## 19 HOU    88062247   0.671  2.785714
## 20 LAL    86775415   0.317  2.764706
## 21 DET   102503259   0.451  2.733333
## 22 PHO    78930157   0.293  2.722222
## 23 BRK    75102568   0.244  2.684211
## 24 LAC   113327068   0.622  2.666667
## 25 DEN    74208517   0.488  2.647059
## 26 NOP    98573436   0.415  2.619048
## 27 TOR   106521470   0.622  2.600000
## 28 MIA    72782449   0.500  2.571429
## 29 PHI    67225712   0.341  2.222222
## 30 SAC    86799609   0.390  2.062500
```

```r
# plot
ggplot(nba_team,
       aes(x = team_salary, y = win_pct, color = avg_clust)) +
  geom_point() +
  geom_label_repel(label = nba_team$Tm) +
  ggtitle('Win % vs. Team Salary') +
  theme_classic()
```




```
# Inspect some teams
teams_sample <- c('NYK', 'GSW', 'BOS')

teams_sample_list <- list(rep(NA, length = length(teams_sample)))
for (i in seq_along(teams_sample)) {
  teams_sample_list[[i]] <- nba[nba$Tm == teams_sample[i],
                                c('Player', 'PER', cl_label)]
}

# change index to see different teams
teams_sample_list[[2]]
```

```
##               Player  PER km_labs_four
## 82         Ian Clark 13.1            4
## 98     Stephen Curry 24.6            2
## 119     Kevin Durant 27.6            2
## 164   Draymond Green 16.5            3
## 211   Andre Iguodala 14.4            1
## 235     Damian Jones  5.3            4
## 268 Shaun Livingston 10.1            4
## 270     Kevon Looney 13.4            4
## 285    James Michael 13.0            4
## 286    Patrick McCaw  8.6            4
## 293     JaVale McGee 25.2            4
## 344    Zaza Pachulia 16.1            3
## 427    Klay Thompson 17.4            2
## 443 Anderson Varejao  9.4            4
```

```
## 457       David West 16.6            4
```

Observations Although a team can have better players on average clusters, there are many variables at play here. A team can be better on average but poor management or coaching can affect a team's overall performance, e.g. NYK. Interestingly, GSW did not have the highest average cluster rating, because their bench is not very strong. This speaks to the strong influence that starter players can have on team performance. Another interesting note is that teams can play well even if they do not have many all-stars or a strong overall team, e.g BOS. This could be driven by great coaching and team chemistry. It is important to note that items such as injuries could greatly influence win %, even if players have high ratings.

Although there is a correlation between overall team salary and win %, it is interesting that average player rating does not necessarily align with overall win %.