ML Final Project: Analysis

Andrew Pagtakhan

January 20, 2020

```
library(tidyverse)
library(dendextend)
library(factoextra)
library(GGally)
library(ggfortify)
library(ggrepel)
library(gridExtra)
library(knitr)
library(mclust)
library(NbClust)
library(rgl)

# set for reproducible results
set.seed(14)

# clear variables
rm(list = ls())

# set working directory

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

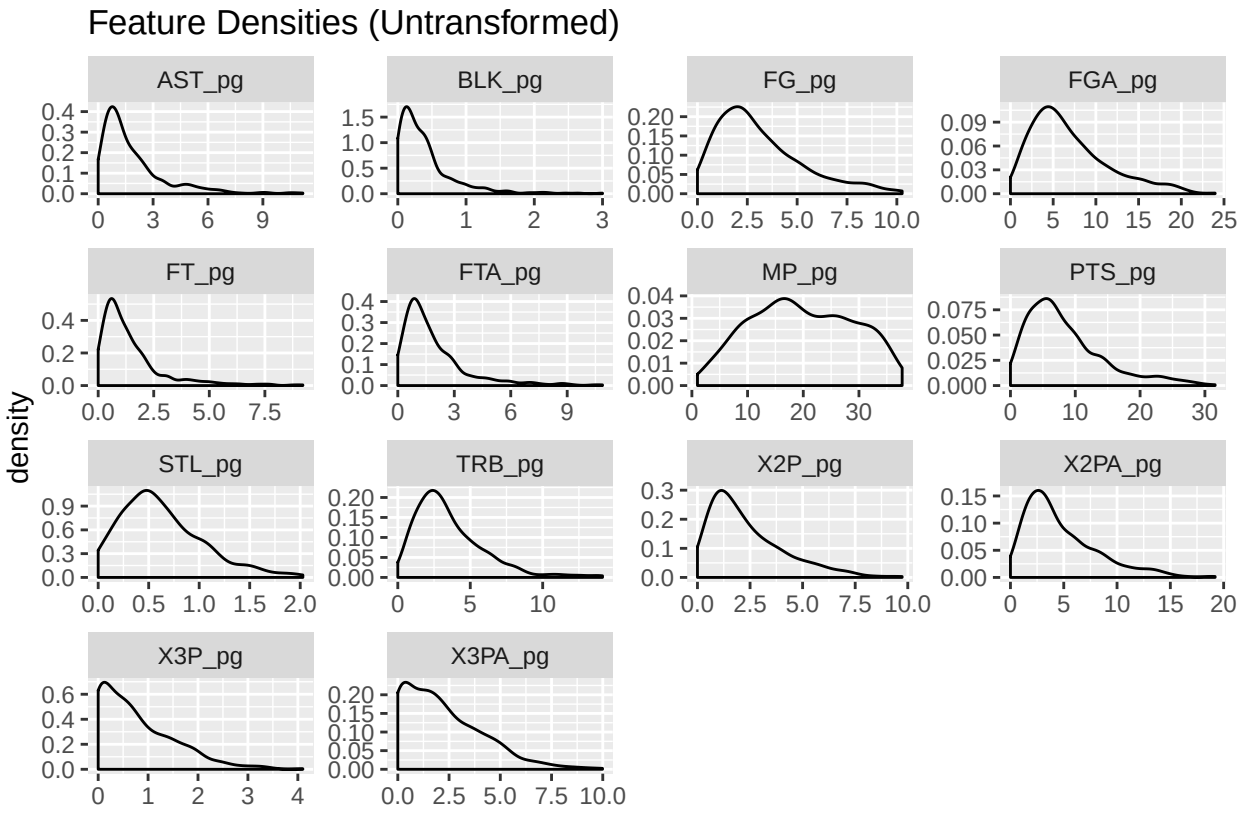
```
# load data
nba <- read.csv(filename)
```

```
# 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]

nba_feat %>%
  pivot_longer(cols = everything()) %>%
  ggplot(aes(x = value)) +
  geom_density() +
  facet_wrap(~name, scales = "free") +
  labs(title = 'Feature Densities (Untransformed)',
       x = '')
```



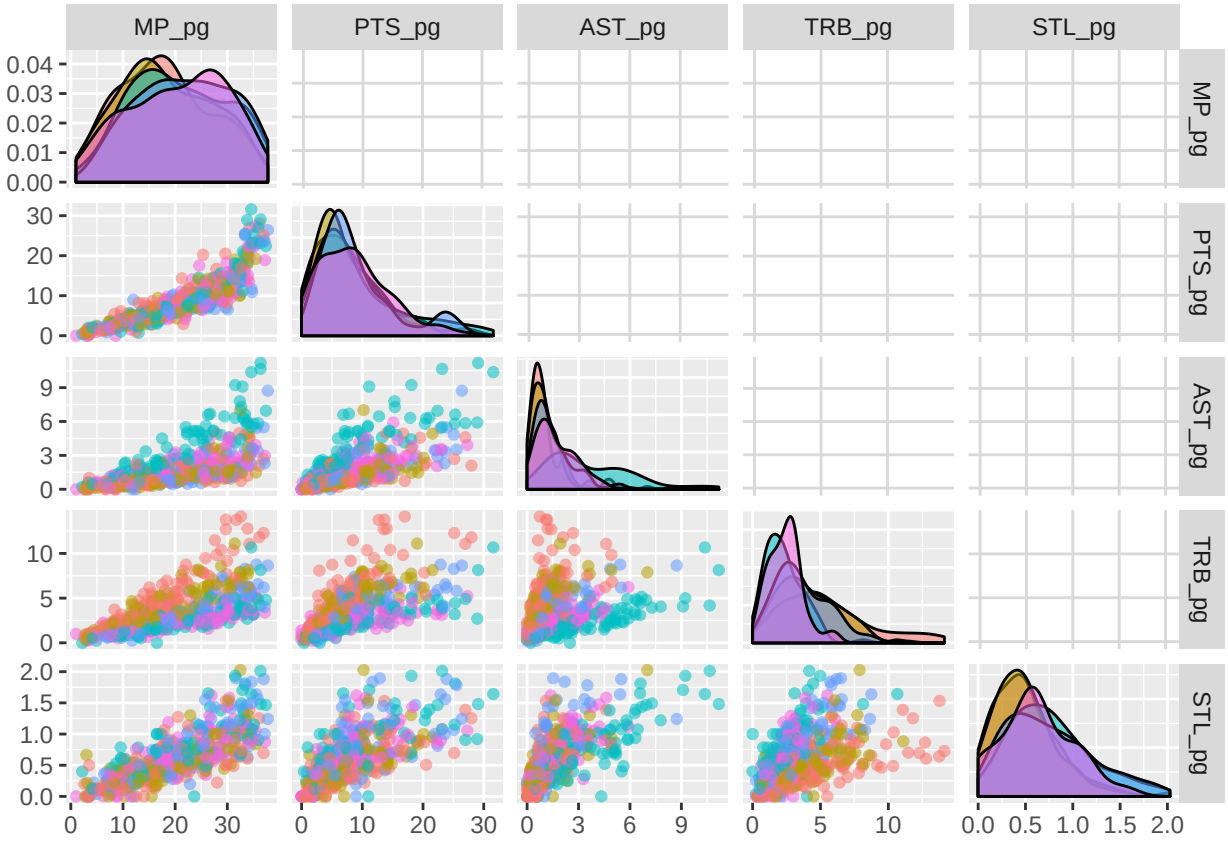
Variance was calculated to determine if we need to scale the data

```
# calculate variance
round(apply(nba_feat, MARGIN = 2, FUN = var), 2)
```

```
##   MP_pg   FG_pg  FGA_pg  X3P_pg X3PA_pg  X2P_pg X2PA_pg   FT_pg  FTA_pg
##   82.07    4.67   20.49    0.57    3.76    3.23   11.89    2.04    2.99
##  TRB_pg  AST_pg  STL_pg  BLK_pg  PTS_pg
##    5.89    3.11    0.17    0.17   36.72
```

```
# look at pairs plots for key stats
feat_plot <- c('MP_pg', 'PTS_pg', 'AST_pg', 'TRB_pg',
```

```
                'STL_pg', 'BLK_pg')
nba_feat_plot_pos <- nba[ , feat_plot]
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.

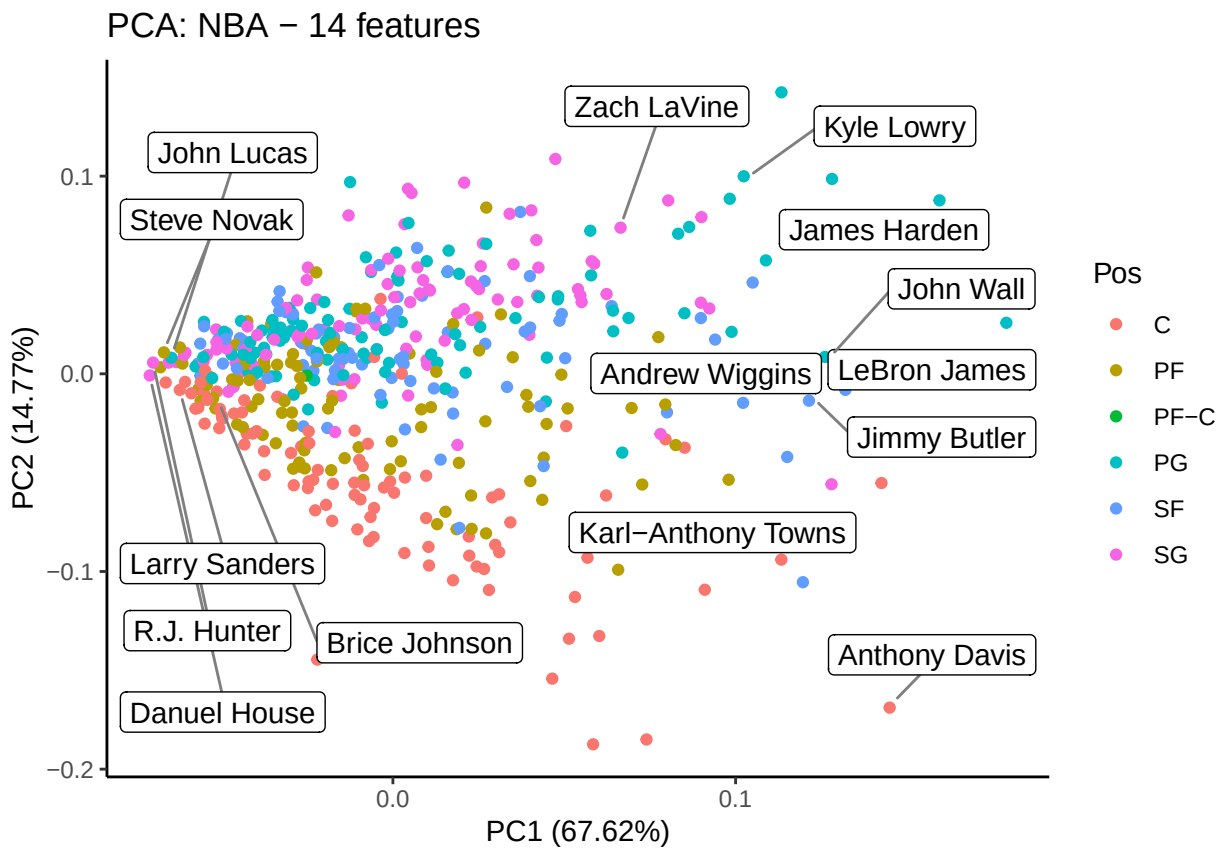
```
nba_feat_sc <- scale(nba_feat)

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



See plotting in 3D

```
pca_plot <- nba_pca$x[ , 1:3]
plot3d(pca_plot)
```

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.

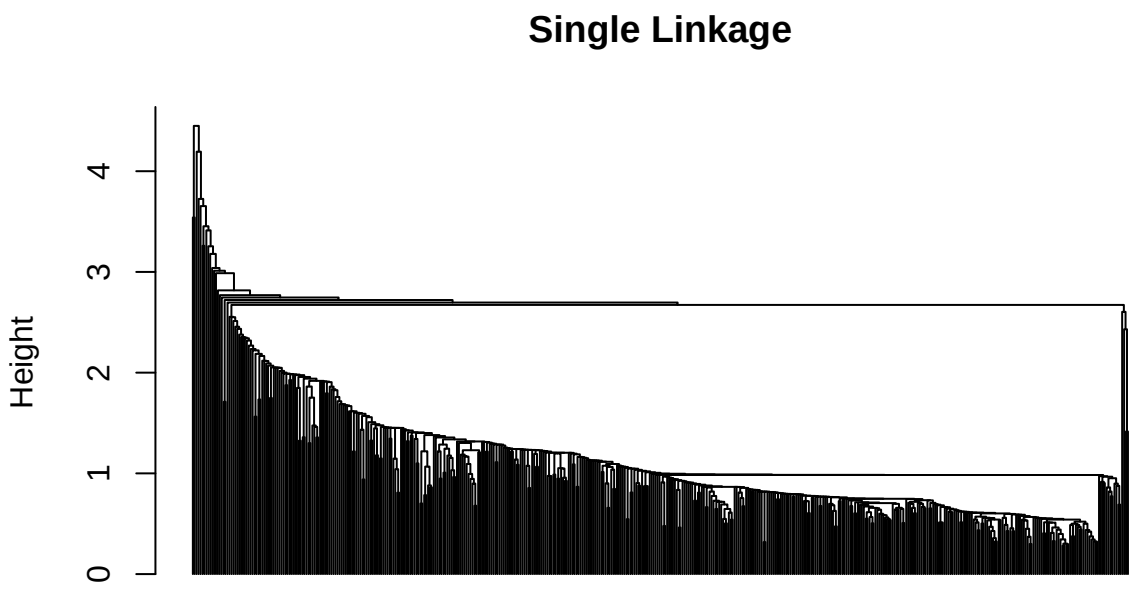
Clustering

Hierarchical clustering

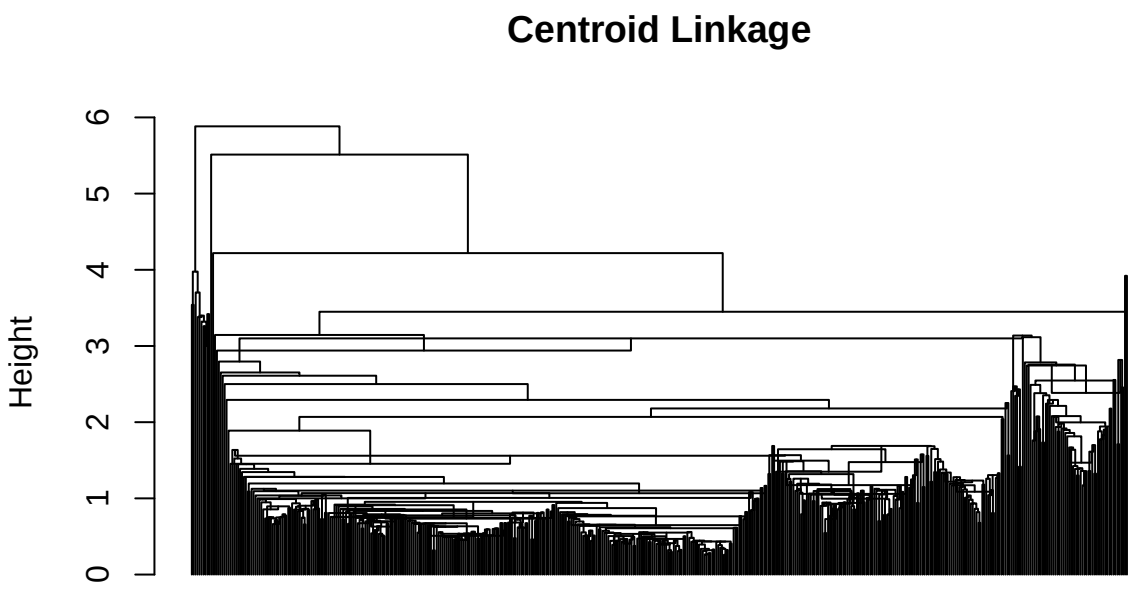
The first method of clustering we will try is hierarchical clustering. The dendrogram can help provide a visual aid in the number of clusters we can start to use.

```
# 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')
```

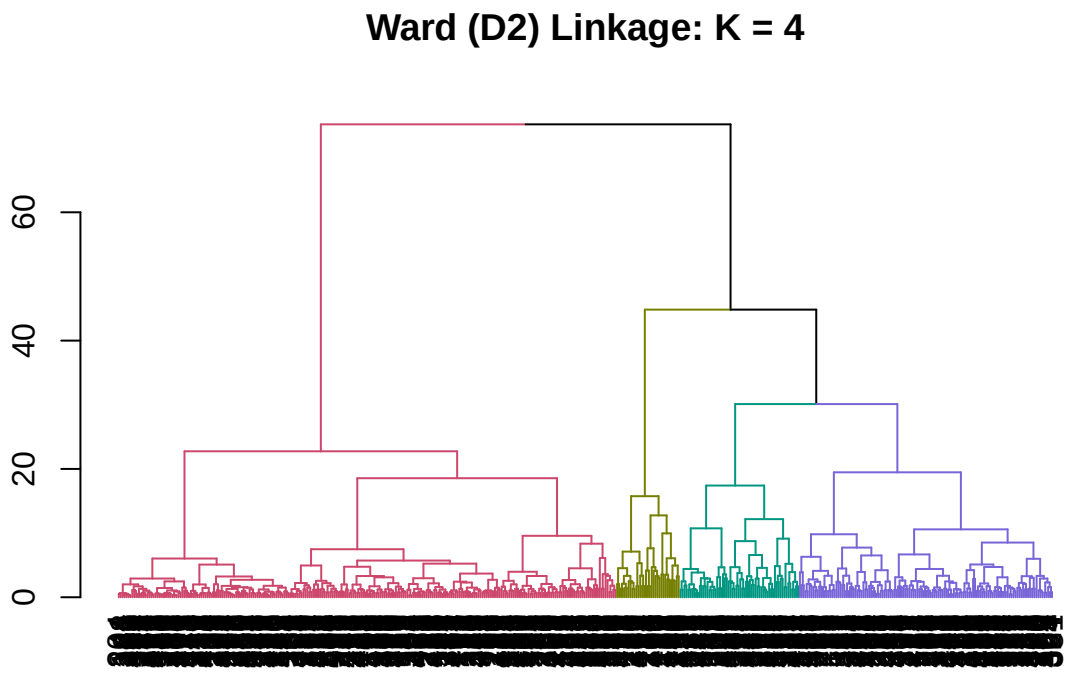
```r
# nearest neighbors method
plot(hcl_single, hang = -1, main = 'Single Linkage',
     labels = FALSE, xlab = '', sub = '')
```




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

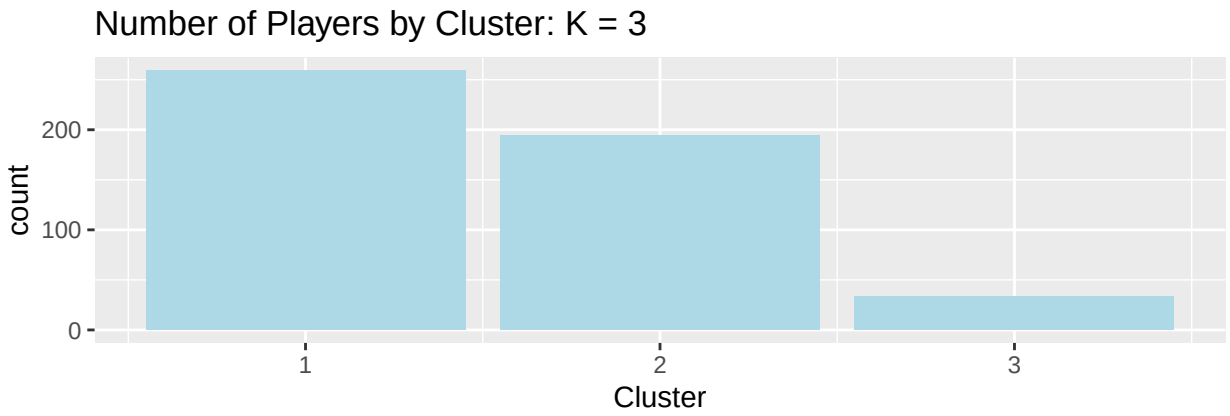


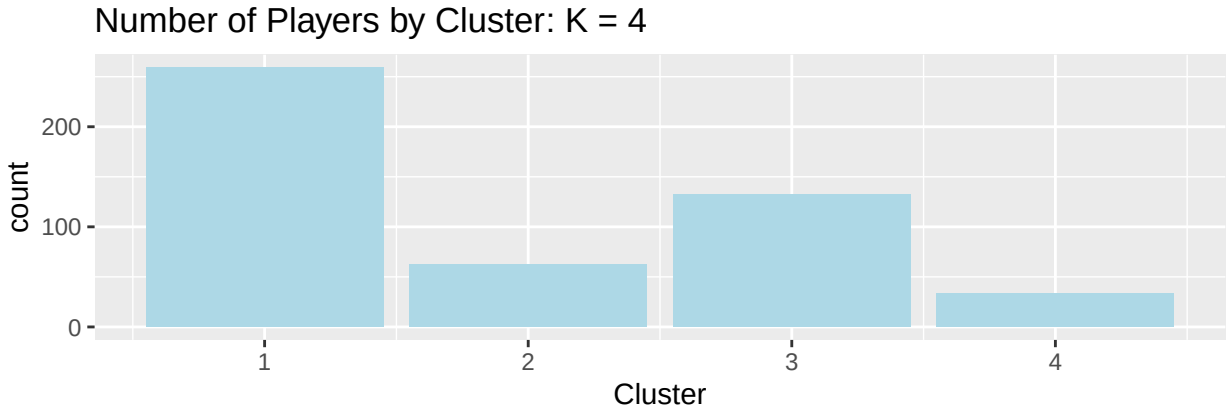

```
# Ward's minimum variance method,
# with dissimilarities are squared before clustering
dend <- as.dendrogram(hcl_ward)
hcl_k <- 4
dend_col <- color_branches(dend, k = hcl_k)
plot(dend_col, main = paste0('Ward (D2) Linkage: K = ', hcl_k))
```



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.

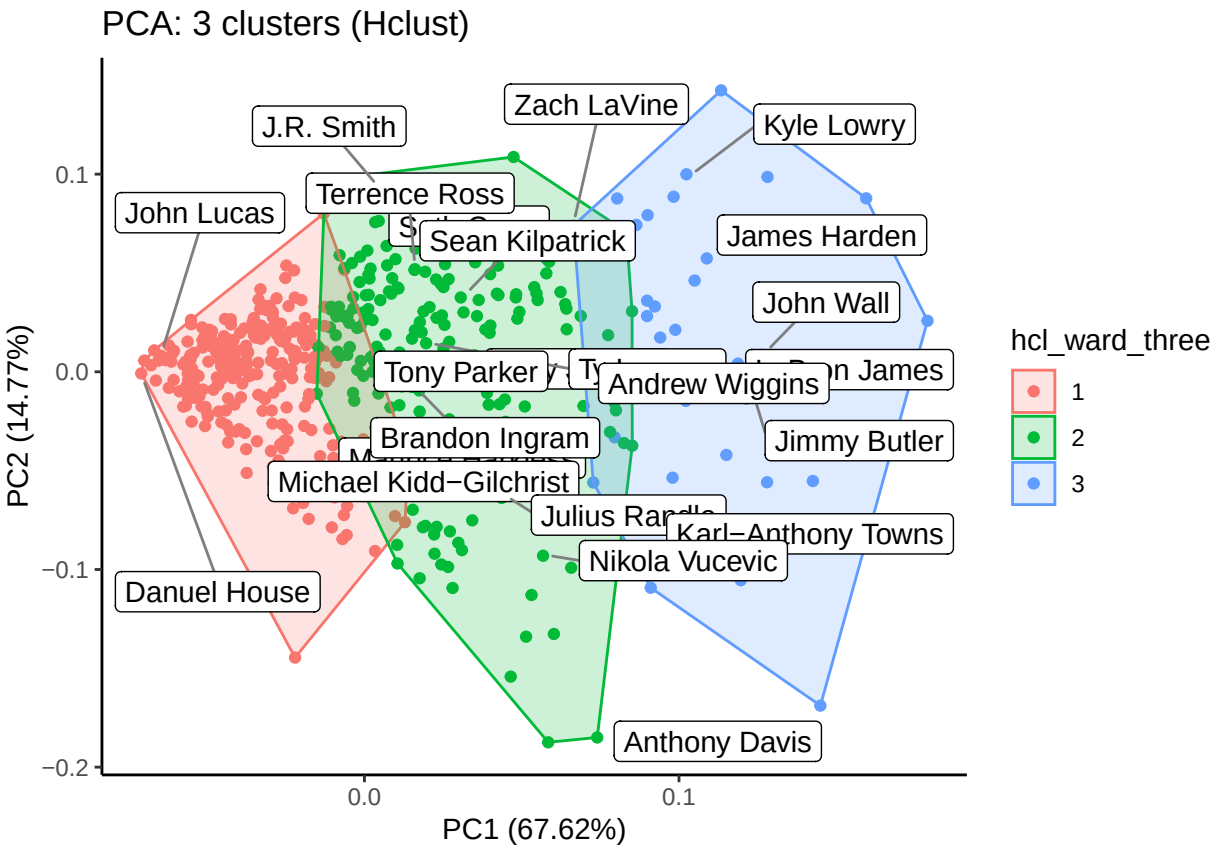
```
# add cluster labels to main data
nba$hcl_ward_labs_three <- cutree(hcl_ward, k = 3)
nba$hcl_ward_labs_four <- cutree(hcl_ward, k = 4)

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

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

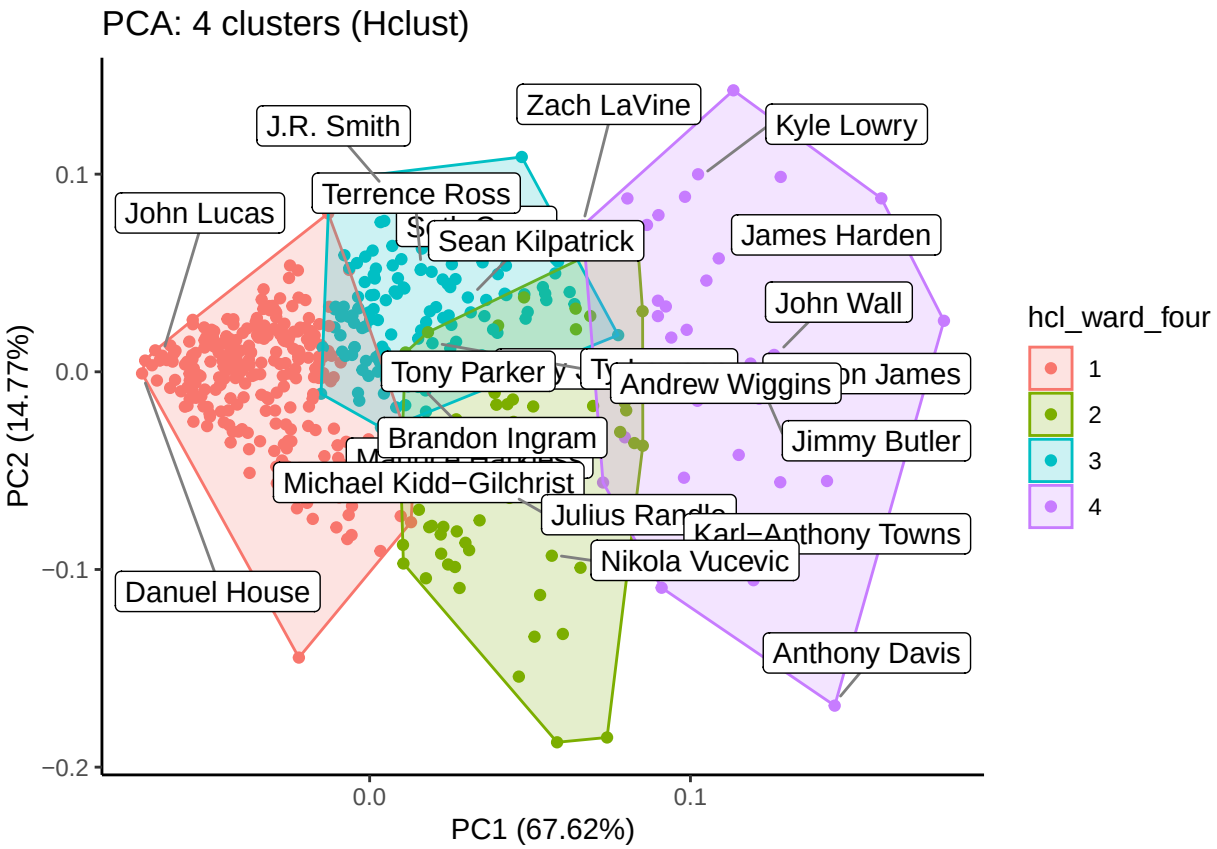




Visualizing clusters in PCA space

```
# 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
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



3d plot

```
hcl_pca <- cbind(nba_pca$x[ , 1:3], enframe(nba$hcl_ward_labs_four))
pca_plot <- hcl_pca
plot3d(pca_plot,
       col = hcl_pca$value, size = 15)
```

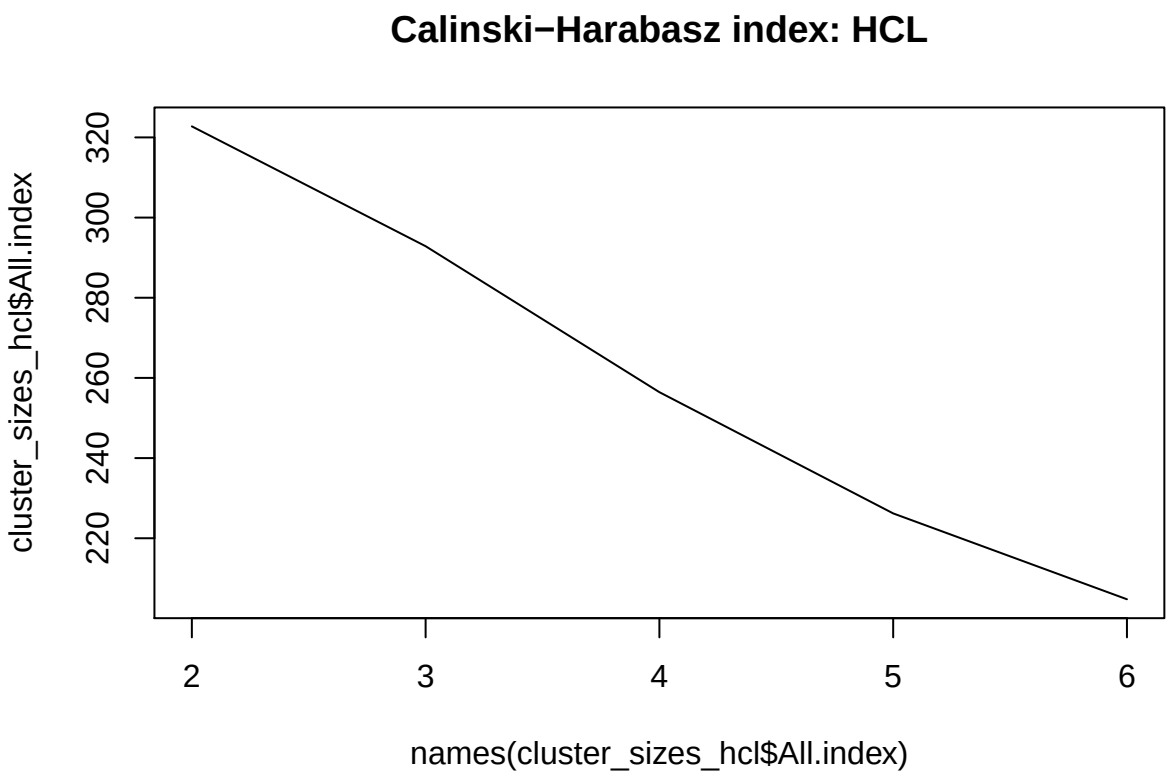
Visually, it looks like the four cluster solution may be able to give us more actionable insights vs the 3-cluster method. The average stats by cluster shows pretty clear separation among the groups. Group 4 are the stars, followed by group 2, 3, and then 4. The main difference between groups 2 and 3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

Optimize number of clusters

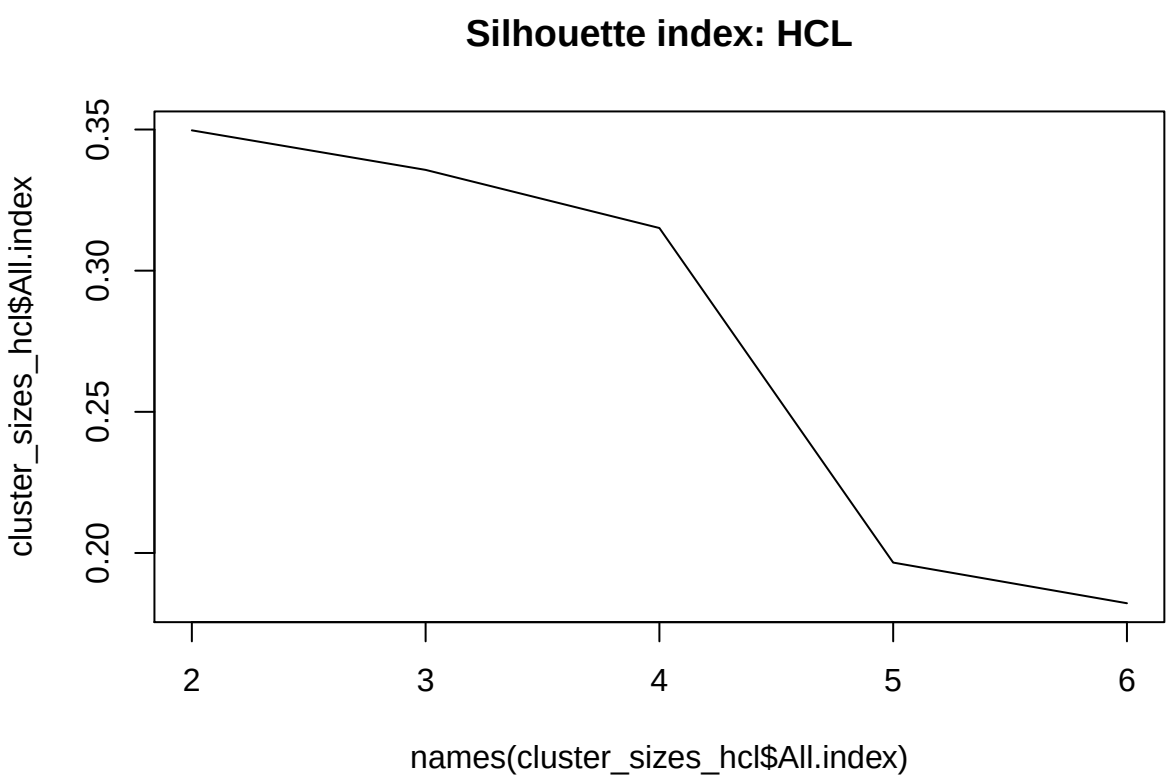
Methods: Calinski-Harabasz index and Scott

```
# 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')
```




```
# 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 = 'silhouette')

# plot C(G)
plot(names(cluster_sizes_hcl$All.index),
     cluster_sizes_hcl$All.index,
     main = 'Silhouette 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.

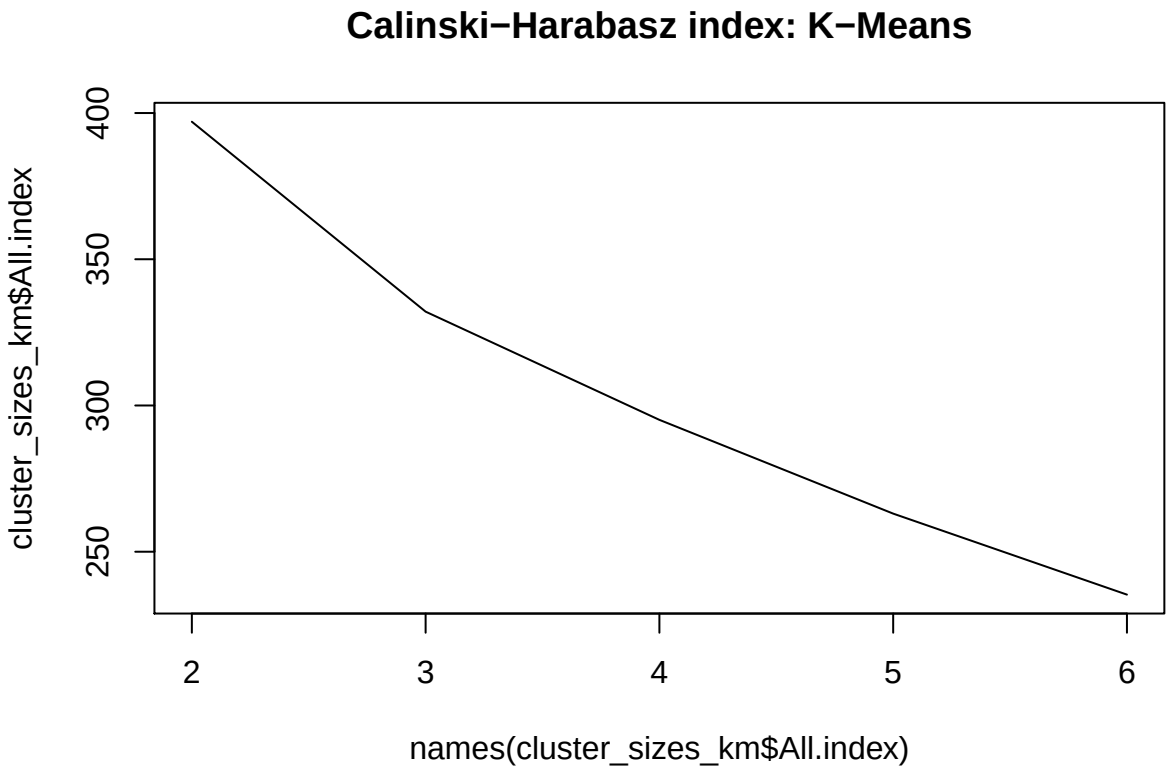
K-Means

Optimize number of clusters

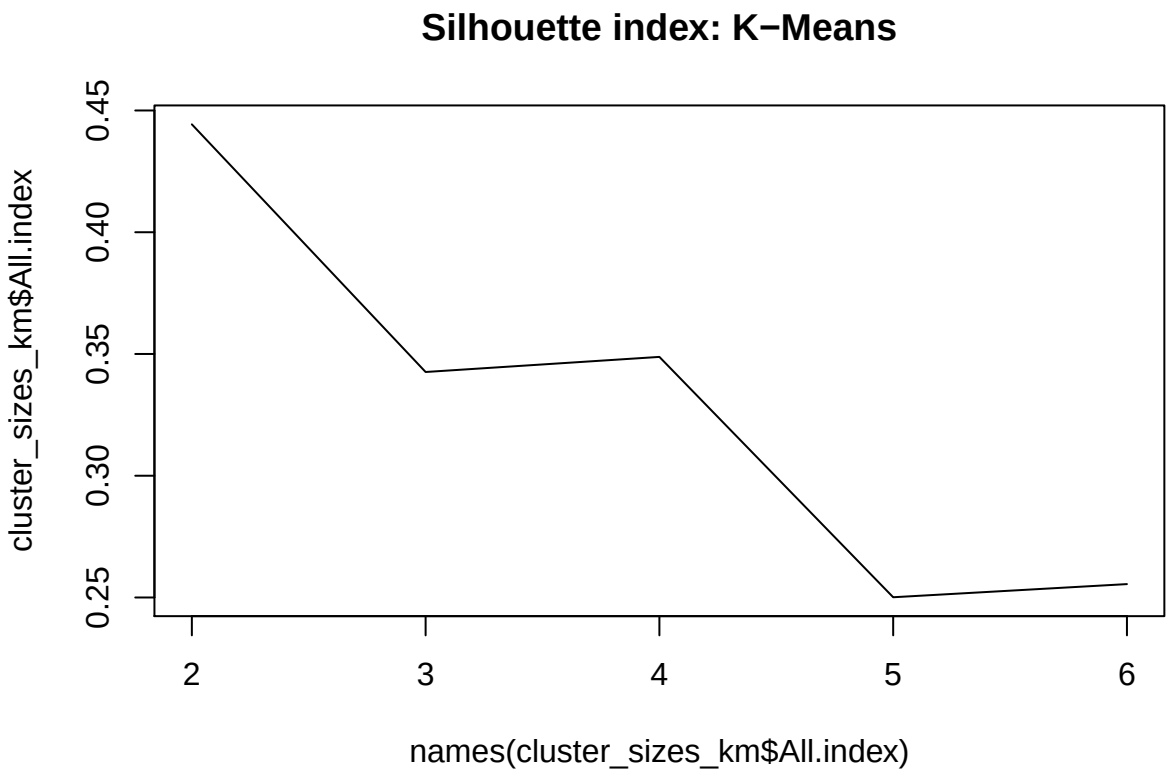
Method: Calinski-Harabasz index

```
# 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 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 = 'silhouette')

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



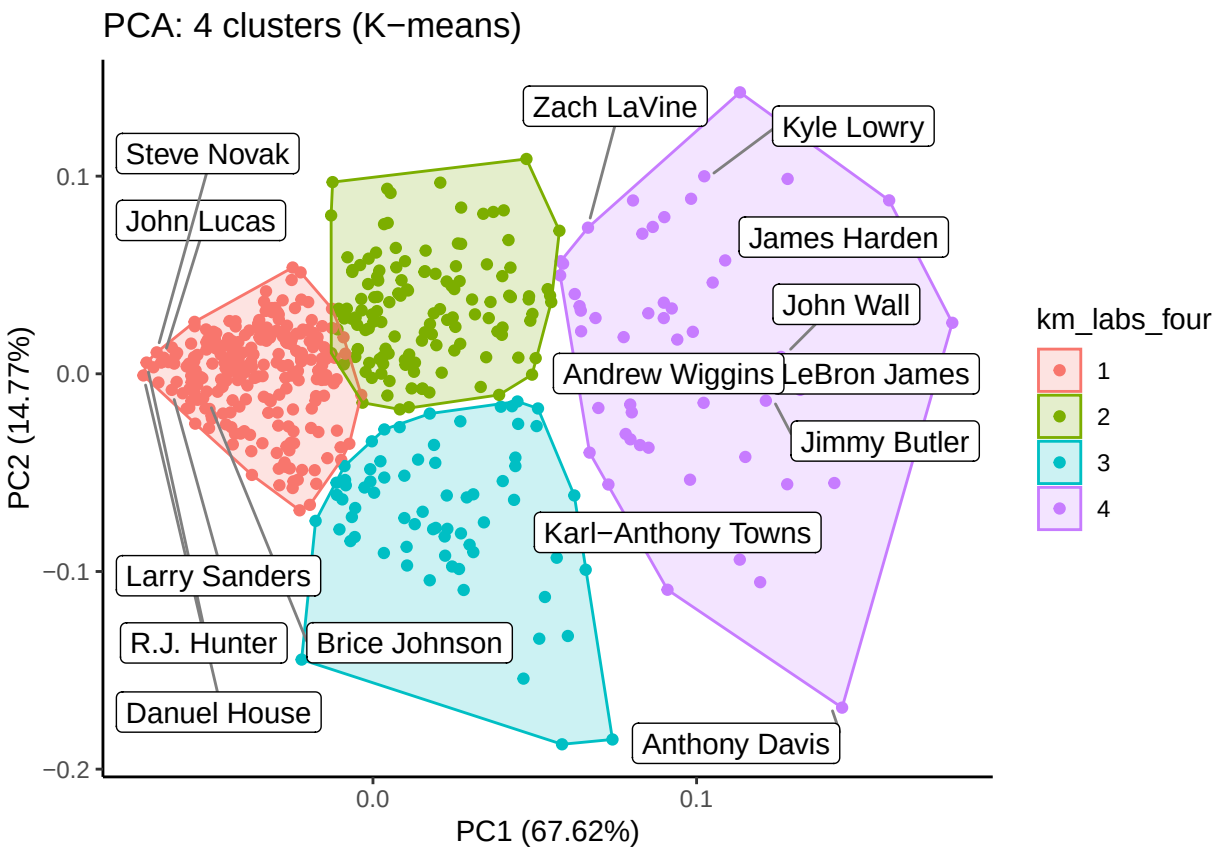
The optimization methods tell us that 2 clusters is best, but we will need more groups for meaningful and interesting separation among players. The silhouette index shows that there is a distinct drop off after 4 clusters.

K-means clustering with 4 groups

```
km_k <- 4
km_four <- kmeans(x = nba_feat_sc,
            centers = km_k,
            nstart = 100,
            algorithm = 'Hartigan-Wong')
```

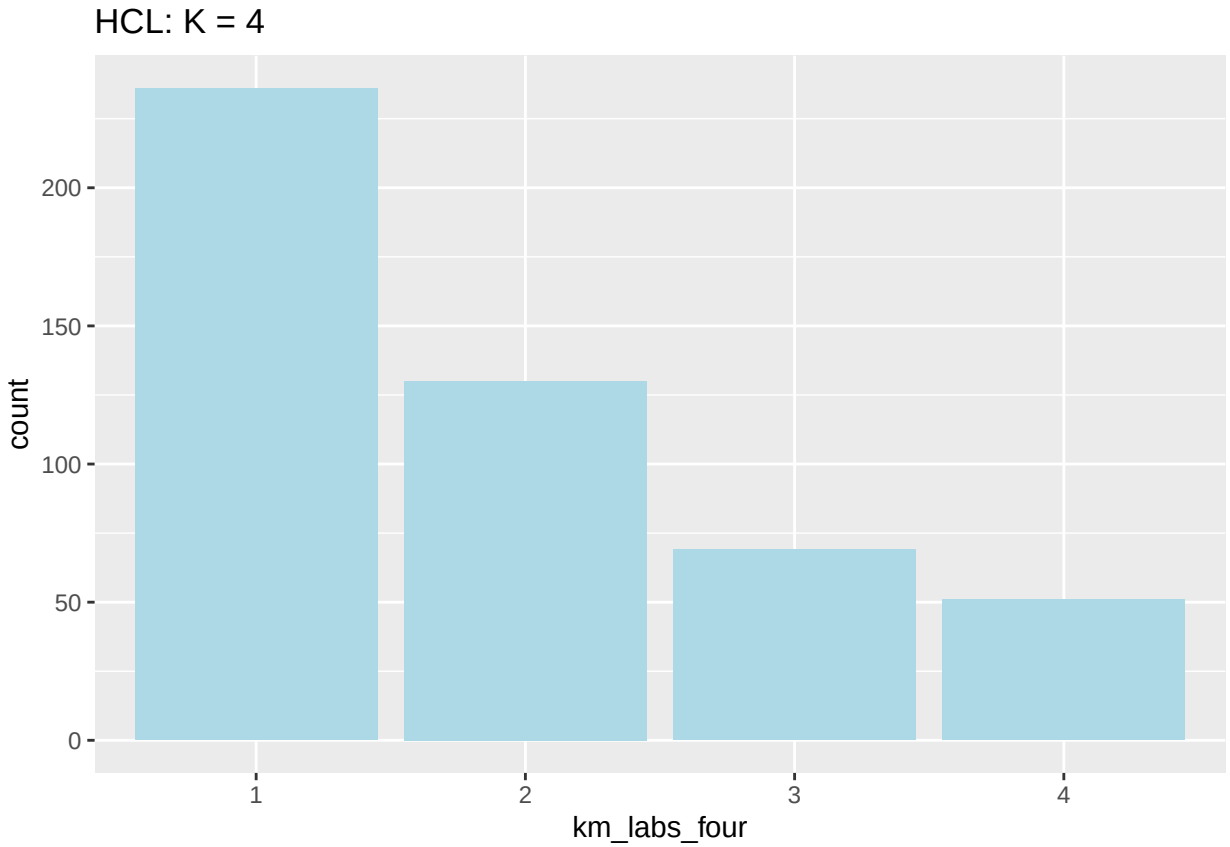
```
nba$km_labs_four <- factor(km_four$cluster)
```

```
# plot k-means clusters in PC space
# 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_four',
          title = paste0('PCA: ', km_k, ' clusters (K-means)'),
          label = km_labels)
```



There is clean separation in the 4-cluster plot. We will now see how these clusters are separated by inspecting features within each group.

```
# get distribution of players in each cluster
ggplot(data = nba,
       aes(x = km_labs_four)) +
       geom_bar(fill = 'lightblue') +
       ggtitle('HCL: K = 4')
```



Describe why distribution is not balanced

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.

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

```
##   km_labs_four    MP_pg   PTS_pg   TRB_pg    AST_pg    BLK_pg    STL_pg
## 1            1 12.24558  3.94709 2.129925 0.8788919 0.2354097 0.3548853
## 2            2 25.61995 10.30825 3.484223 2.5209841 0.2885179 0.8214061
## 3            3 24.96591 10.29999 7.010810 1.6435808 0.9473312 0.7868895
## 4            4 33.85906 21.82156 5.724658 4.7255814 0.6158290 1.1623242
##          VORP      PER        RPM
## 1 -0.06694915 10.18941 -1.9458898
## 2  0.50153846 12.75308 -0.8625385
## 3  1.26811594 17.09710  0.3679710
## 4  3.19215686 21.29020  2.5100000
```

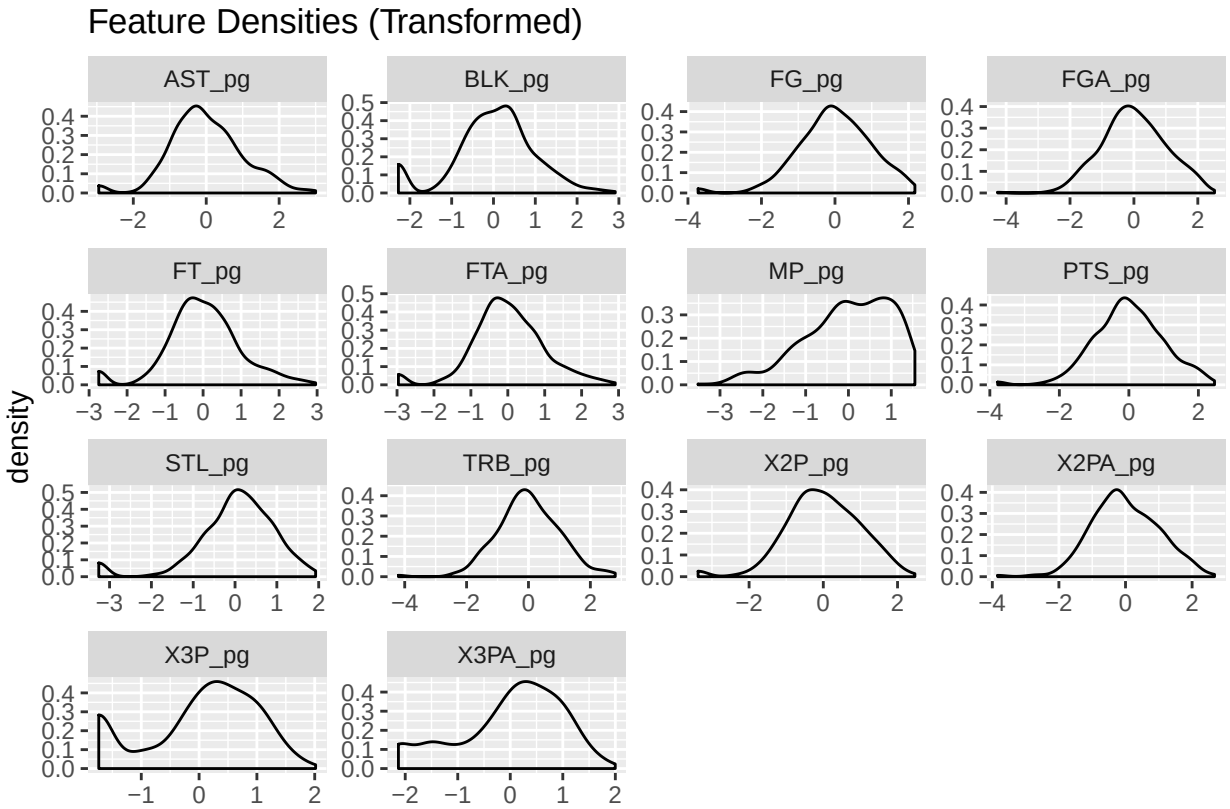
Observations Although the CH index indicates that the optimal number of clusters is 2, this seems too low of a number to meaningfully break out the NBA players into groups. It is also important to note that the CH index is a heuristic method. So although CH is a good approach to look for the number of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

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

```
# transform features
nba_feat_cr <- (nba_feat) ^ (1/3)
# scale transformed features
nba_feat_cr_sc <- scale(nba_feat_cr)
```

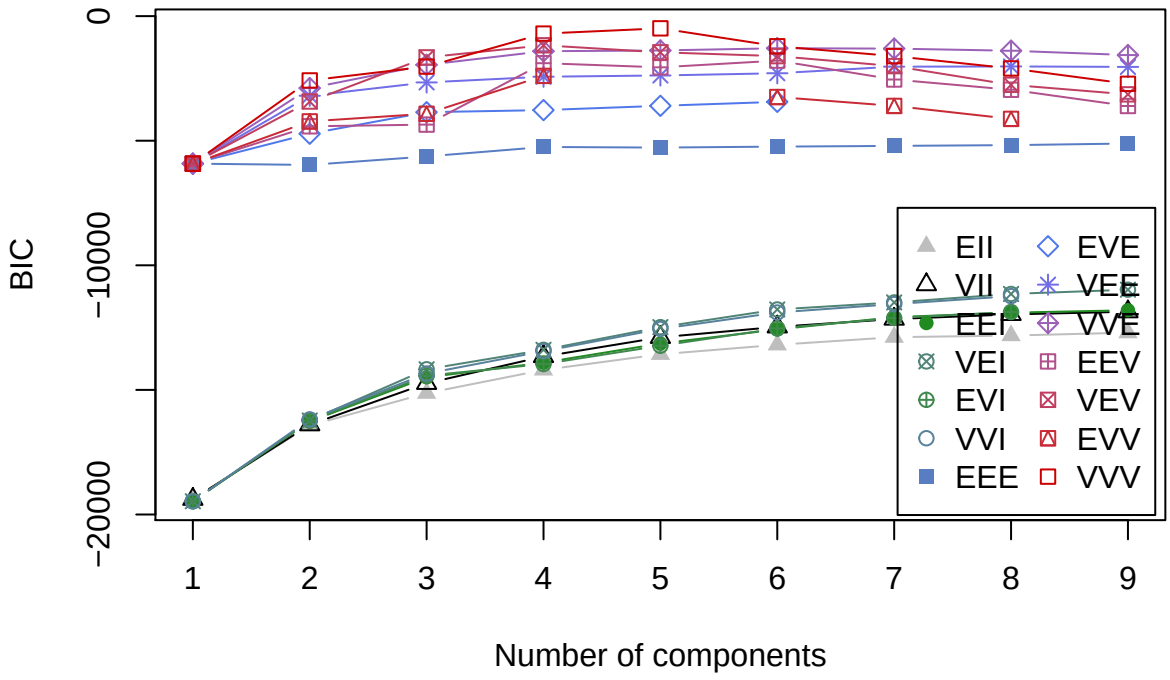
```
# plot transformed variables
nba_feat_cr_sc %>% as_tibble() %>%
  pivot_longer(cols = everything()) %>%
  ggplot(aes(x = value)) +
  geom_density() +
  facet_wrap(~name, scales = "free") +
  labs(title = 'Feature Densities (Transformed)',
       x = '')
```



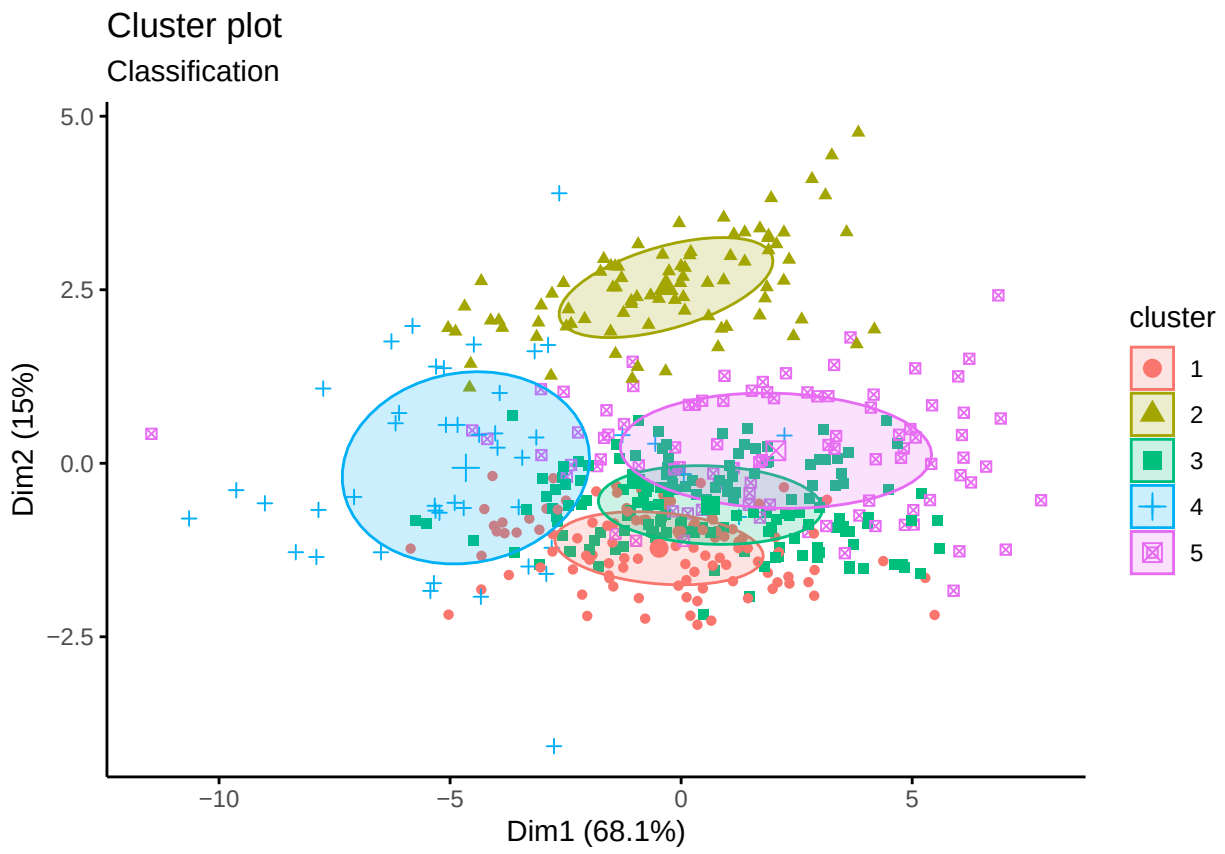

```
# run model
player_clust.mcl <- Mclust(nba_feat_cr_sc)
summary(player_clust.mcl)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVV (ellipsoidal, varying volume, shape, and orientation) model
## with 5 components:
##
##  log-likelihood   n  df       BIC       ICL
##        1607.448 486 599 -490.6436 -505.4606
##
## Clustering table:
##   1   2   3   4   5
## 106  84 160  45  91
```

```
plot(player_clust.mcl, what = "BIC")
```




```
# plot results
fviz_mclust(player_clust.mcl, "classification", geom = "point")
```




```
# add cluster labels to plot
nba$mcl_labs <- player_clust.mcl$classification
```

Compare methods between Clusters

We will now compare crosstab solutions

```
# run crosstabs between cluster methods
xtab_hcl_km <- xtabs(~nba$hcl_ward_four + nba$km_labs_four)
xtab_hcl_mcl <- xtabs(~nba$hcl_ward_labs_four + nba$mcl_labs)
xtab_km_mcl <- xtabs(~nba$km_labs_four + nba$mcl_labs)

xtab_hcl_km
```

```
##                  nba$km_labs_four
## nba$hcl_ward_four   1   2   3   4
##                 1 231   4  24   0
##                 2   0   8  42  12
##                 3   5 118   3   6
##                 4   0   0   0  33
```

```
xtab_hcl_mcl
```

```
##                       nba$mcl_labs
## nba$hcl_ward_labs_four  1  2  3  4  5
##                      1 55 60 75 41 28
##                      2  0 24 12  1 25
##                      3 48  0 63  3 18
```

```
##                       4  3  0 10  0 20
```

```
xtab_km_mcl
```

```
##                  nba$mcl_labs
## nba$km_labs_four  1  2  3  4  5
##                1 52 40 74 41 29
##                2 51  0 63  2 14
##                3  0 44  7  2 16
##                4  3  0 16  0 32
```

Between HCL and KM, the maximum possible agreement between clusters is 87% (424 / 486). Between HCL and MCL, the maximum possible agreement between clusters is 30% (148 / 486). Between KM and MCL, the maximum possible agreement between clusters is 41% (201 / 486).

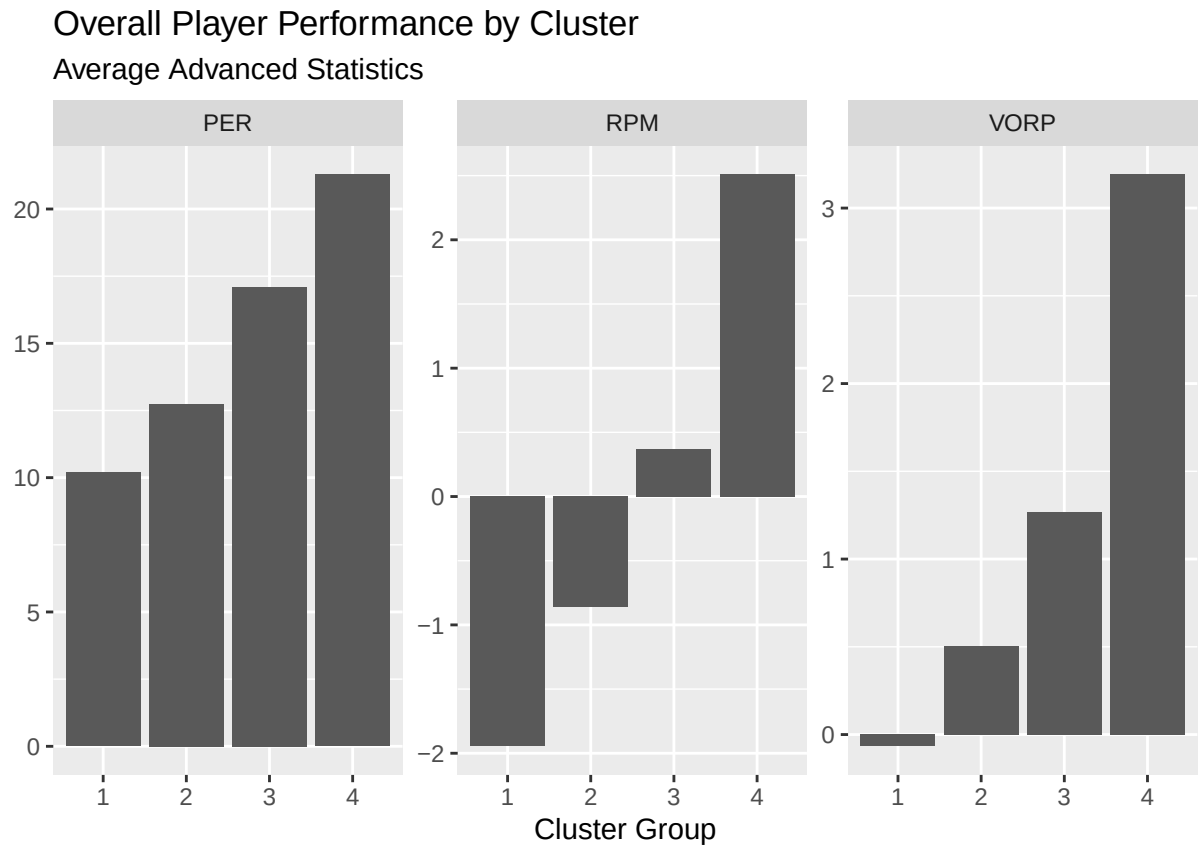
Based on the analysis, MCL tends to group players by position, whereas HCL and KM tend to cluster based on overall player statistics. This conclusion was reached based on inspecting distributions of player positions across the clustering methods. Because we are looking at player statistics and team compensation, the model-based clustering is not an ideal fit for this purpose.

Final Cluster selection

K-Means (4 clusters) was the optimal solution. Comparing the HCL and KM cluster plots (per above) reveals the K-Means produces clearer separation of players based on overall skillsets.

Inspection of clusters suggest that groupings separate players by skillsets. We validated this by comparing the clusters against advanced statistics. PER, VORP, and RPM are advanced statistics commonly used to assess general player performance. None of these statistics were used in the cluster modeling.

```
# plot averages by cluster
nba_km_avg %>%
  select(km_labs_four, VORP, PER, RPM) %>%
  pivot_longer(cols = c("VORP", "PER", "RPM")) %>%
  ggplot(aes(x = km_labs_four, y = value)) +
  geom_col() +
  facet_wrap(~name, scales = 'free') +
  labs(title = 'Overall Player Performance by Cluster',
       subtitle = 'Average Advanced Statistics',
       x = 'Cluster Group',
       y = '')
```

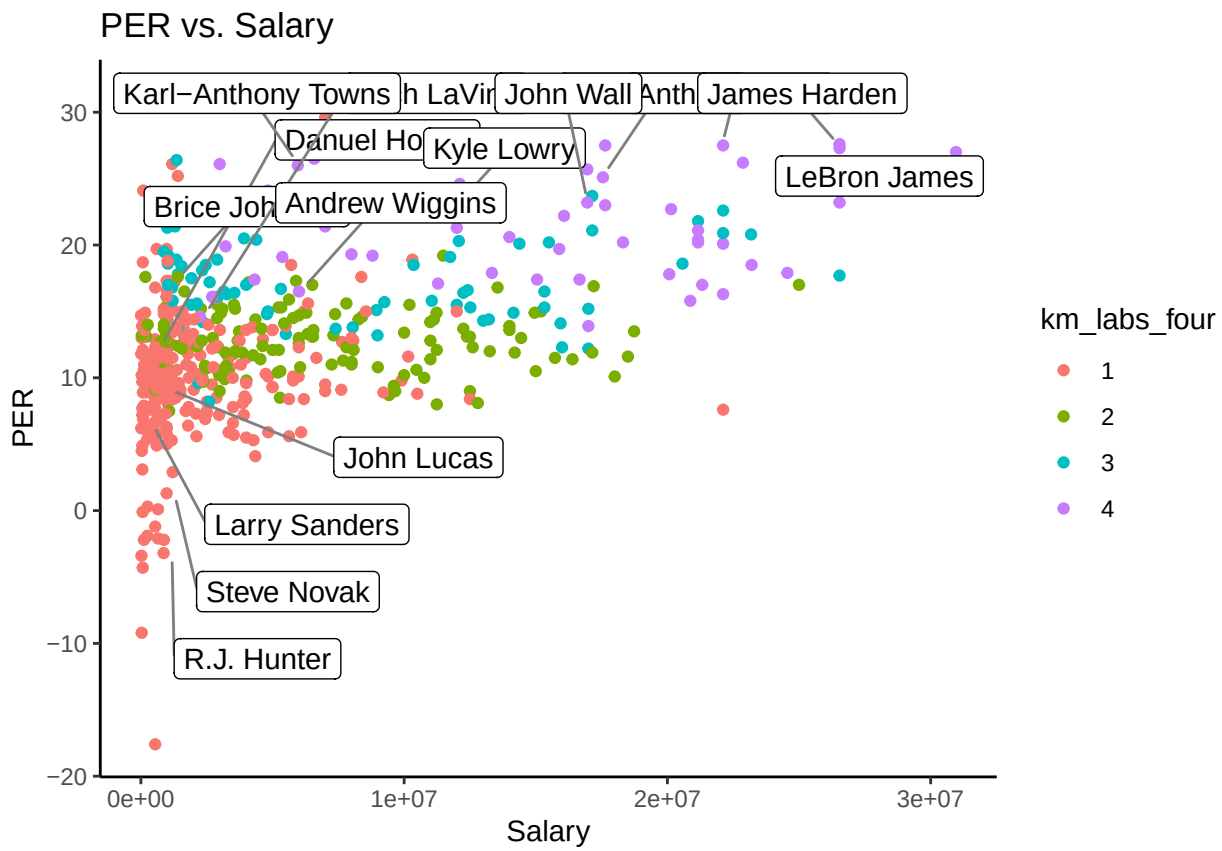


Post-Cluster Analysis

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

Clusters vs. Player Salaries

```
# 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()
```




```
# 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      Chandler Parsons 34 19.852941 MEM 22116750   7.6            1
## 2         Miles Plumlee 45 10.755556 MIL 12500000   8.4            1
## 3          Amir Johnson 80 20.100000 BOS 12000000  15.0            1
## 4       Mirza Teletovic 70 16.185714 MIL 10500000   8.8            1
## 5          Al Jefferson 66 14.106061 IND 10314532  18.9            1
## 6            Alec Burks 42 15.547619 UTA 10154495  11.6            1
## 7             Omer Asik 31 15.548387 NOP  9904494   9.8            1
## 8        Meyers Leonard 74 16.513514 POR  9213483   8.9            1
## 9        Tiago Splitter  8  9.500000 PHI  8550000  15.0            1
## 10        Dwight Powell 77 17.311688 DAL  8375000  17.6            1
## 11       Darrell Arthur 41 15.585366 DEN  8070175  12.8            1
## 12         Tyler Zeller 51 10.294118 BOS  8000000  13.0            1
## 13         Cole Aldrich 62  8.564516 MIN  7643979  12.7            1
## 14         Corey Brewer 82 15.621951 HOU  7612172   9.1            1
## 15           Boris Diaw 73 17.575342 UTA  7000000   9.0            1
## 16    Boban Marjanovic 35  8.371429 DET  7000000  29.6            1
## 17      Rodney Stuckey 39 17.846154 IND  7000000   9.5            1
## 18            Ed Davis 46 17.152174 POR  6666667  11.5            1
```

```
## 19              Aron Baynes 75 15.506667 DET  6500000  13.1           1
## 20            Spencer Hawes 54 14.759259 CHO  6348759  15.6           1
## 21              Tarik Black 67 16.283582 LAL  6191000  15.0           1
## 22             Lance Thomas 46 21.043478 NYK  6191000   8.4           1
## 23         Andrew Nicholson 38  9.000000 WAS  6088993   5.9           1
## 24           Ramon Sessions 50 16.220000 CHO  6000000  12.3           1
## 25             Alex Abrines 68 15.514706 OKC  5994764  10.1           1
## 26         Shaun Livingston 76 17.697368 GSW  5782450  10.1           1
## 27           Josh McRoberts 22 17.318182 MIA  5782450   9.8           1
## 28           Brandan Wright 28 15.964286 MEM  5709880  18.5           1
## 29           Wesley Johnson 68 11.911765 LAC  5628000   8.4           1
## 30          Jared Sullinger 11 10.727273 TOR  5628000   5.6           1
## 31               Luis Scola 36 12.805556 BRK  5500000  13.9           1
## 32             Roy Hibbert 48 14.208333 CHO  5000000  13.6           1
## 33            Jonas Jerebko 78 15.794872 BOS  5000000   9.3           1
## 34             Jason Smith 74 14.432432 WAS  5000000  13.6           1
## 35             C.J. Watson 62 16.322581 ORL  5000000   9.3           1
## 36            Mike Dunleavy 53 15.867925 ATL  4837500  10.1           1
## 37             Kyle Singler 32 12.031250 OKC  4837500   5.9           1
## 38             Jaylen Brown 78 17.192308 BOS  4743000  10.3           1
## 39            Alexis Ajinca 39 14.974359 NOP  4638203  12.9           1
## 40          Greivis Vasquez  3 13.000000 BRK  4347826   4.1           1
## 41            Dragan Bender 43 13.348837 PHO  4276320   5.3           1
## 42             Devin Harris 65 16.723077 DAL  4227996  13.8           1
## 43               Lavoy Allen 61 14.278689 IND  4000000  11.6           1
## 44         Leandro Barbosa 67 14.373134 PHO  4000000  11.5           1
## 45            Udonis Haslem 17  7.647059 MIA  4000000   8.4           1
## 46              Jordan Hill  7  6.714286 MIN  4000000   5.5           1
## 47           Kris Humphries 56 12.303571 ATL  4000000  13.6           1
## 48         Lance Stephenson 18 20.055556 NOP  4000000   9.6           1
## 49                Dante Exum 66 18.606061 UTA  3940320   8.6           1
## 50            Mario Hezonja 65 14.769231 ORL  3909840   7.2           1
## 51                 Kris Dunn 78 17.089744 MIN  3872520   8.1           1
## 52          Nemanja Bjelica 65 18.307692 MIN  3800000  11.0           1
## 53            Nick Collison 20  6.400000 OKC  3750000  12.8           1
## 54              Paul Pierce 25 11.080000 LAC  3527920   5.7           1
## 55              Mike Miller 20  7.550000 DEN  3500000   7.8           1
## 56             Brandon Rush 47 21.914894 MIN  3500000   6.6           1
## 57          Anthony Morrow 49 14.571429 OKC  3488000  10.0           1
## 58               Trey Burke 57 12.333333 WAS  3386598  10.8           1
## 59               Mike Scott 18 10.833333 ATL  3333334   5.9           1
## 60           K.J. McDaniels 49 10.306122 BRK  3333333  11.5           1
## 61          Justin Hamilton 64 18.390625 BRK  3000000  13.6           1
## 62          Stanley Johnson 77 17.805195 DET  2969880   7.2           1
## 63           Dewayne Dedmon 76 17.500000 SAS  2898000  16.0           1
## 64     Mindaugas Kuzminskas 68 14.941176 NYK  2898000  12.4           1
## 65         Tomas Satoransky 57 12.614035 WAS  2870813   8.5           1
## 66               Noah Vonleh 74 17.094595 POR  2751360  10.8           1
## 67              Jakob Poeltl 54 11.592593 TOR  2703960  12.2           1
## 68              Aaron Brooks 65 13.753846 IND  2700000   9.5           1
## 69               Thon Maker 57  9.859649 MIL  2568600  14.0           1
## 70          Malcolm Delaney 73 17.095890 ATL  2500000   7.5           1
## 71               Randy Foye 69 18.608696 BRK  2500000   7.3           1
## 72        Richard Jefferson 79 20.430380 CLE  2500000   8.2           1
```

```
## 73         Domantas Sabonis 81 20.148148 OKC  2440200   6.9            1
## 74               Trey Lyles 71 16.309859 UTA  2340600  10.0            1
## 75    Taurean Waller-Prince 59 16.627119 ATL  2318280   9.8            1
## 76           Reggie Bullock 31 15.064516 DET  2255644  11.7            1
## 77                Luc Mbah 80 22.337500 LAC  2203000  10.3            1
## 78     Georgios Papagiannis 22 16.136364 SAC  2202240  12.7            1
## 79            Cameron Payne 31 14.903226 OKC  2112480   5.6            1
## 80         Denzel Valentine 57 17.122807 CHI  2092200   7.3            1
## 81            Adreian Payne 18  7.500000 MIN  2022240  14.4            1
## 82              Kelly Oubre 79 20.316456 WAS  2006640   9.1            1
## 83         Juan Hernangomez 62 13.580645 DEN  1987440  13.3            1
## 84             Terry Rozier 74 17.067568 BOS  1906440  10.8            1
## 85              James Young 29  7.586207 BOS  1825200  10.0            1
## 86            Rashad Vaughn 41 11.170732 MIL  1811040   7.8            1
## 87           Kevin Seraphin 49 11.408163 IND  1800000  14.4            1
## 88             Wade Baldwin 33 12.272727 MEM  1793760   6.4            1
## 89               Quincy Acy 38 14.684211 BRK  1790092  11.8            1
## 90              Tyler Ennis 53 11.094340 LAL  1733880  11.0            1
## 91               Sam Dekker 77 18.428571 HOU  1720560  13.0            1
## 92        Joffrey Lauvergne 70 14.000000 OKC  1709719  12.6            1
## 93           Henry Ellenson 19  7.684211 DET  1704120   7.5            1
## 94             Jerian Grant 63 16.317460 CHI  1643040  13.1            1
## 95            Malik Beasley 22  7.500000 DEN  1627320  13.7            1
## 96            Bruno Caboclo  9  4.444444 TOR  1589640  14.6            1
## 97             Delon Wright 27 16.518519 TOR  1577280  15.0            1
## 98          Justin Anderson 75 16.373333 DAL  1514160  13.9            1
## 99          Marcelo Huertas 23 10.304348 LAL  1500000   9.1            1
## 100         DeAndre' Bembry 38  9.763158 ATL  1499760   8.8            1
## 101            Bobby Portis 64 15.625000 CHI  1453680  14.9            1
## 102       Demetrius Jackson  5  3.400000 BOS  1450000  30.8            1
## 103      Malachi Richardson 22  9.000000 SAC  1439800   9.6            1
## 104             JaVale McGee 77  9.597403 GSW  1403611  25.2            1
## 105           Deyonta Davis 36  6.611111 MEM  1369299  10.6            1
## 106          Shabazz Napier 53  9.660377 POR  1350120  13.6            1
## 107              Tyus Jones 60 12.900000 MIN  1339680  13.8            1
## 108 Timothe Luwawu-Cabarrot 69 17.246377 PHI  1326960   8.5            1
## 109            Jarell Martin 42 13.285714 MEM  1286160   8.7            1
## 110            Brice Johnson  3  3.000000 LAC  1273920  17.2            1
## 111             Luke Babbitt 68 15.661765 MIA  1227286   8.4            1
## 112            Jordan Mickey 25  5.640000 BOS  1223653   9.8            1
## 113              C.J. Wilcox 22  4.909091 ORL  1209600   2.9            1
## 114            Pascal Siakam 55 15.618182 TOR  1196040  11.5            1
## 115            Kyle Anderson 72 14.166667 SAS  1192080  12.5            1
## 116             Josh Huestis  2 15.500000 OKC  1191480  26.1            1
## 117        Chris McCullough 16  5.000000 BRK  1191480  15.2            1
## 118             Kevon Looney 53  8.433962 GSW  1182840  13.4            1
## 119         Dejounte Murray 38  8.473684 SAS  1180080   9.6            1
## 120            Damian Jones 10  8.500000 GSW  1171560   5.3            1
## 121        Rakeem Christmas 29  7.551724 IND  1052342  10.4            1
## 122              Joe Young 33  4.090909 IND  1052342  11.4            1
## 123            John Jenkins  4  3.250000 PHO  1050961  17.3            1
## 124          Glenn Robinson 69 20.666667 IND  1050500  11.5            1
## 125         Anthony Bennett 23 11.478261 BRK  1015696  14.7            1
## 126            Isaiah Canaan 39 15.179487 CHI  1015696   8.1            1
```

```
## 127         DeAndre Liggins 62 12.532258 CLE  1015696   7.5           1
## 128            Mike Muscala 70 17.671429 ATL  1015696  14.4           1
## 129             Willie Reed 71 14.521127 MIA  1015696  17.1           1
## 130             Jeff Withey 51  8.470588 UTA  1015696  18.8           1
## 131          Chris Andersen 12  9.500000 CLE   980431  11.6           1
## 132           Alan Anderson 30 10.266667 LAC   980431   5.0           1
## 133            Brandon Bass 52 11.096154 LAC   980431  19.7           1
## 134               Ian Clark 77 14.766234 GSW   980431  13.1           1
## 135            Jerami Grant 80 19.137500 OKC   980431  10.1           1
## 136            Gerald Green 47 11.446809 BOS   980431  12.0           1
## 137             James Jones 48  7.937500 CLE   980431  11.3           1
## 138              John Lucas  5  2.200000 MIN   980431   9.1           1
## 139           James Michael 52  8.788462 GSW   980431  13.0           1
## 140            Steve Novak  8  2.750000 MIL   980431   1.3           1
## 141          Arinze Onuaku  8  3.500000 ORL   980431   5.8           1
## 142          Brian Roberts 41 10.146341 CHO   980431   9.8           1
## 143        Thomas Robinson 48 11.666667 LAL   980431  17.3           1
## 144           Damjan Rudez 45  6.977778 ORL   980431   6.3           1
## 145         Jarnell Stokes  2  3.500000 DEN   980431  31.5           1
## 146             Jason Terry 74 18.445946 MIL   980431   9.0           1
## 147        Marcus Thornton 33 17.424242 WAS   980431  10.4           1
## 148              Beno Udrih 39 14.358974 DET   980431  16.1           1
## 149       Anderson Varejao 14  6.571429 GSW   980431   9.4           1
## 150          Sasha Vujacic 42  9.714286 NYK   980431   8.6           1
## 151             David West 68 12.558824 GSW   980431  16.6           1
## 152            Metta World 25  6.400000 LAL   980431   6.2           1
## 153      Stephen Zimmerman 19  5.684211 ORL   950000   7.3           1
## 154        Andrew Harrison 72 20.472222 MEM   945000   8.7           1
## 155              Raul Neto 40  8.650000 UTA   937800  10.7           1
## 156         Reggie Williams  6 13.166667 NOP   895197  11.7           1
## 157         Pat Connaughton 39  8.102564 POR   874636  11.8           1
## 158      Cristiano Felicio 66 15.757576 CHI   874636  15.2           1
## 159          Aaron Harrison  5  3.400000 CHO   874636  -2.2           1
## 160         Darrun Hilliard 39  9.769231 DET   874636   5.9           1
## 161            Jordan McRae 37 10.378378 CLE   874636   9.7           1
## 162             Salah Mejri 73 12.397260 DAL   874636  14.8           1
## 163       Jonathon Simmons 78 17.846154 SAS   874636   9.9           1
## 164          Christian Wood 13  8.230769 CHO   874636  15.1           1
## 165             R.J. Hunter  3  3.000000 CHI   864346  -3.2           1
## 166             Paul Zipser 44 19.159091 CHI   750000   6.9           1
## 167             Bobby Brown 25  4.920000 HOU   680534  10.8           1
## 168         Michael Gbinije  9  3.555556 DET   650000  -2.1           1
## 169            A.J. Hammons 22  7.409091 DAL   650000   8.4           1
## 170          Georges Niang 23  4.043478 IND   650000   0.1           1
## 171          Joel Bolomboy 12  4.416667 UTA   600000  19.7           1
## 172            Jake Layman 35  7.114286 POR   600000   4.9           1
## 173     Donatas Motiejunas 34 14.088235 NOP   576724   9.2           1
## 174              Ron Baker 52 16.480769 NYK   543471   7.5           1
## 175             Ben Bentil  3  3.333333 DAL   543471 -17.6           1
## 176          Davis Bertans 67 12.059701 SAS   543471  12.9           1
## 177       Nicolas Brussino 54  9.648148 DAL   543471  10.7           1
## 178          Semaj Christon 64 15.203125 OKC   543471   5.7           1
## 179          Cheick Diallo 17 11.705882 NOP   543471  16.8           1
## 180             Kay Felder 42  9.190476 CLE   543471  11.2           1
```

```
## 181     Dorian Finney-Smith 81 20.271605 DAL   543471   7.7           1
## 182            Bryn Forbes 36  7.916667 SAS   543471   5.9           1
## 183         Treveon Graham 27  7.000000 CHO   543471  10.6           1
## 184           Danuel House  1  1.000000 WAS   543471  12.2           1
## 185          Derrick Jones 32 17.031250 PHO   543471  12.0           1
## 186    Nicolas Laprovittola 18  9.666667 SAS   543471   8.4           1
## 187          Patrick McCaw 71 15.126761 GSW   543471   8.6           1
## 188       Sheldon McClellan 30  9.566667 WAS   543471  10.1           1
## 189          Maurice Ndour 32 10.343750 NYK   543471  11.3           1
## 190          Daniel Ochefu 19  3.947368 WAS   543471   6.6           1
## 191         Chinanu Onuaku  5 10.400000 HOU   543471  12.3           1
## 192        Marshall Plumlee 21  8.095238 NYK   543471  10.9           1
## 193          Tim Quarterman 16  5.000000 POR   543471  10.2           1
## 194          Diamond Stone  7  3.428571 LAC   543471  -1.2           1
## 195          Fred VanVleet 37  7.945946 TOR   543471  10.5           1
## 196           Kyle Wiltjer 14  3.142857 HOU   543471   6.7           1
## 197        Jonathan Gibson 17 13.588235 DAL   469943   9.5           1
## 198          Toney Douglas 24 16.416667 MEM   379159  10.6           1
## 199           Joel Anthony 19  6.421053 SAS   346034  11.6           1
## 200             Ryan Kelly 16  6.875000 ATL   286785   7.8           1
## 201          Ronnie Price 14  9.571429 PHO   276828   5.9           1
## 202          Isaiah Taylor  4 13.000000 HOU   255000   0.3           1
## 203          Jose Calderon 41 13.146341 LAL   247991   8.9           1
## 204           Norris Cole 13  9.615385 OKC   247991   5.4           1
## 205        Lamar Patterson  5  8.000000 ATL   246956  -1.9           1
## 206           Andrew Bogut 27 21.592593 DAL   242224   9.3           1
## 207           Okaro White 35 13.457143 MIA   210995   7.5           1
## 208          Larry Sanders  5  2.600000 CLE   207722   6.5           1
## 209        Johnny O'Bryant 11  7.272727 DEN   161483  14.9           1
## 210         Chasson Randle 26 11.500000 NYK   143860  13.6           1
## 211           Omri Casspi 36 17.861111 SAC   138414   9.9           1
## 212          Briante Weber 20 10.250000 CHO   128623  11.0           1
## 213           Manny Harris  4  6.250000 DAL   115344  -2.2           1
## 214        Hollis Thompson 40 18.800000 PHI   115344   7.9           1
## 215       Derrick Williams 50 16.080000 CLE   115344  10.6           1
## 216             Shawn Long 18 13.000000 PHI    89513  24.1           1
## 217           Wayne Selden 14 16.857143 MEM    83119   6.9           1
## 218          Troy Williams 30 18.566667 MEM    76725   8.9           1
## 219         Archie Goodwin 15 14.266667 BRK    75000  18.7           1
## 220           David Nwaba 20 19.850000 LAL    73528  12.1           1
## 221             Gary Neal  2  9.000000 ATL    72193  -4.3           1
## 222            Mike Tobey  2 12.500000 CHO    67135  -0.1           1
## 223         Anthony Brown 11 14.454545 NOP    57672   7.2           1
## 224          Jarell Eddie  5 12.400000 PHO    57672   9.7           1
## 225           Alonzo Gee 13  6.846154 DEN    57672   3.1           1
## 226          Justin Harper  3 10.333333 PHI    57672   4.9           1
## 227          Jarrett Jack  2 16.500000 NOP    57672   7.7           1
## 228          Jarrod Uthoff  9 12.777778 DAL    47953  13.9           1
## 229           Gary Payton  6 16.500000 MIL    35116   4.5           1
## 230        Patricio Garino  5  8.600000 ORL    31969  -9.2           1
## 231    Marcus Georges-Hunt  5  9.600000 ORL    31969  10.2           1
## 232         Pierre Jackson  8 10.500000 DAL    31969  13.0           1
## 233         Elijah Millsap  2 11.500000 PHO    23069  -3.4           1
## 234             Quinn Cook 14 13.428571 NOP    15984  11.8           1
```

```
## 235            Axel Toupane  4 11.750000 NOP    15435   6.2            1
## 236          Dahntay Jones  1 12.000000 CLE     5767  14.7            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)
           )
```

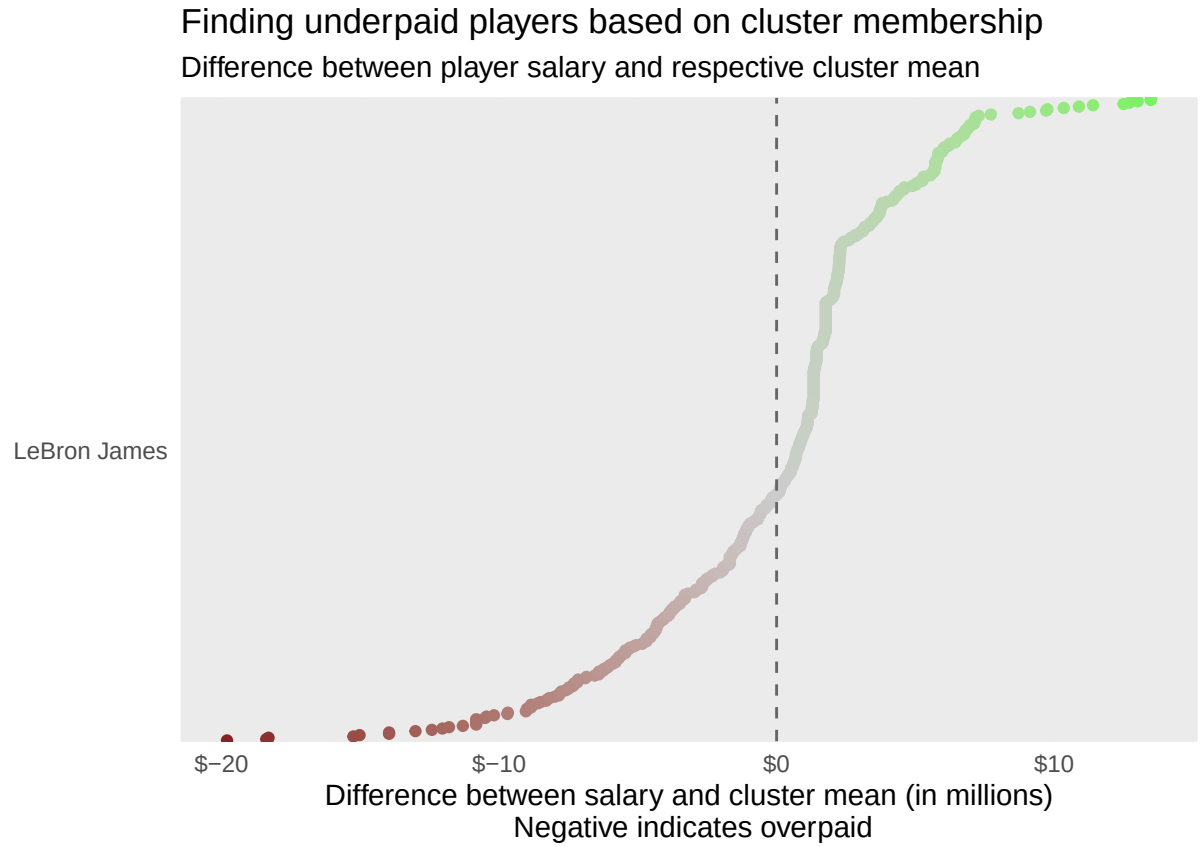
```
##                  Player  G    MP_pg  Tm   Salary  PER km_labs_four
## 1          Devin Booker 78 35.00000 PHO  2223600 14.6            4
## 2           Zach LaVine 47 37.21277 MIN  2240880 14.6            4
## 3       Dennis Schroder 79 31.45570 ATL  2708582 16.1            4
## 4  Giannis Antetokounmpo 80 35.56250 MIL  2995421 26.1            4
## 5         C.J. McCollum 80 34.95000 POR  3219579 19.9            4
## 6    Kristaps Porzingis 66 32.78788 NYK  4317720 17.4            4
## 7           Joel Embiid 31 25.35484 PHI  4826160 24.1            4
## 8         Jabari Parker 51 33.88235 MIL  5374320 19.1            4
## 9    Karl-Anthony Towns 82 36.95122 MIN  5960160 26.0            4
## 10       Andrew Wiggins 82 37.17073 MIN  6006600 16.5            4
## 11        Isaiah Thomas 76 33.80263 BOS  6587132 26.5            4
## 12         Lou Williams 81 24.61728 LAL  7000000 21.4            4
## 13          George Hill 49 31.51020 UTA  8000000 19.3            4
## 14          Jeff Teague 82 32.40244 IND  8800000 19.2            4
## 15         Jrue Holiday 67 32.68657 NOP 11286518 17.1            4
## 16           Kyle Lowry 60 37.40000 TOR 12000000 22.9            4
## 17         Kemba Walker 79 34.67089 CHO 12000000 21.3            4
## 18        Stephen Curry 79 33.39241 GSW 12112359 24.6            4
## 19            Rudy Gay 30 33.76667 SAC 13333333 17.9            4
## 20         Eric Bledsoe 66 32.96970 PHO 14000000 20.6            4
## 21     Danilo Gallinari 63 33.87302 DEN 15050000 17.4            4
## 22         Goran Dragic 73 33.68493 MIA 15891725 19.7            4
## 23       Gordon Hayward 73 34.46575 UTA 16073140 22.2            4
## 24        Klay Thompson 78 33.96154 GSW 16663575 17.4            4
## 25     DeMarcus Cousins 72 34.23611 SAC 16957900 25.7            4
## 26            John Wall 78 36.35897 WAS 16957900 23.2            4
## 27        Evan Fournier 68 32.85294 ORL 17000000 13.9            4
## 28         Jimmy Butler 76 36.96053 CHI 17552209 25.1            4
## 29         Kyrie Irving 72 35.06944 CLE 17638063 23.0            4
## 30        Kawhi Leonard 74 33.43243 SAS 17638063 27.5            4
## 31          Paul George 75 35.85333 IND 18314532 20.2            4
## 32         Paul Millsap 69 33.95652 ATL 20072033 17.8            4
## 33        Blake Griffin 61 34.03279 LAC 20140839 22.7            4
## 34        Nicolas Batum 77 33.98701 CHO 20869566 15.8            4
## 35           Marc Gasol 74 34.20270 MEM 21165675 20.2            4
## 36          Brook Lopez 75 29.62667 BRK 21165675 20.4            4
## 37           Kevin Love 60 31.41667 CLE 21165675 21.1            4
## 38         Derrick Rose 64 32.53125 NYK 21323252 17.0            4
## 39      Harrison Barnes 79 35.48101 DAL 22116750 16.3            4
## 40         Bradley Beal 77 34.85714 WAS 22116750 20.1            4
## 41        Anthony Davis 75 36.10667 NOP 22116750 27.5            4
## 42           Chris Paul 61 31.49180 LAC 22868827 26.2            4
## 43          Dwyane Wade 60 29.86667 CHI 23200000 18.5            4
## 44       Damian Lillard 75 35.92000 POR 24328425 24.1            4
```

```
## 45       Carmelo Anthony 74 34.29730 NYK 24559380 17.9           4
## 46           Mike Conley 69 33.21739 MEM 26540100 23.2           4
## 47         DeMar DeRozan 74 35.40541 TOR 26540100 24.0           4
## 48          Kevin Durant 62 33.38710 GSW 26540100 27.6           4
## 49          James Harden 81 36.38272 HOU 26540100 27.3           4
## 50     Russell Westbrook 81 34.59259 OKC 26540100 30.6           4
## 51          LeBron James 74 37.75676 CLE 30963450 27.0           4
```

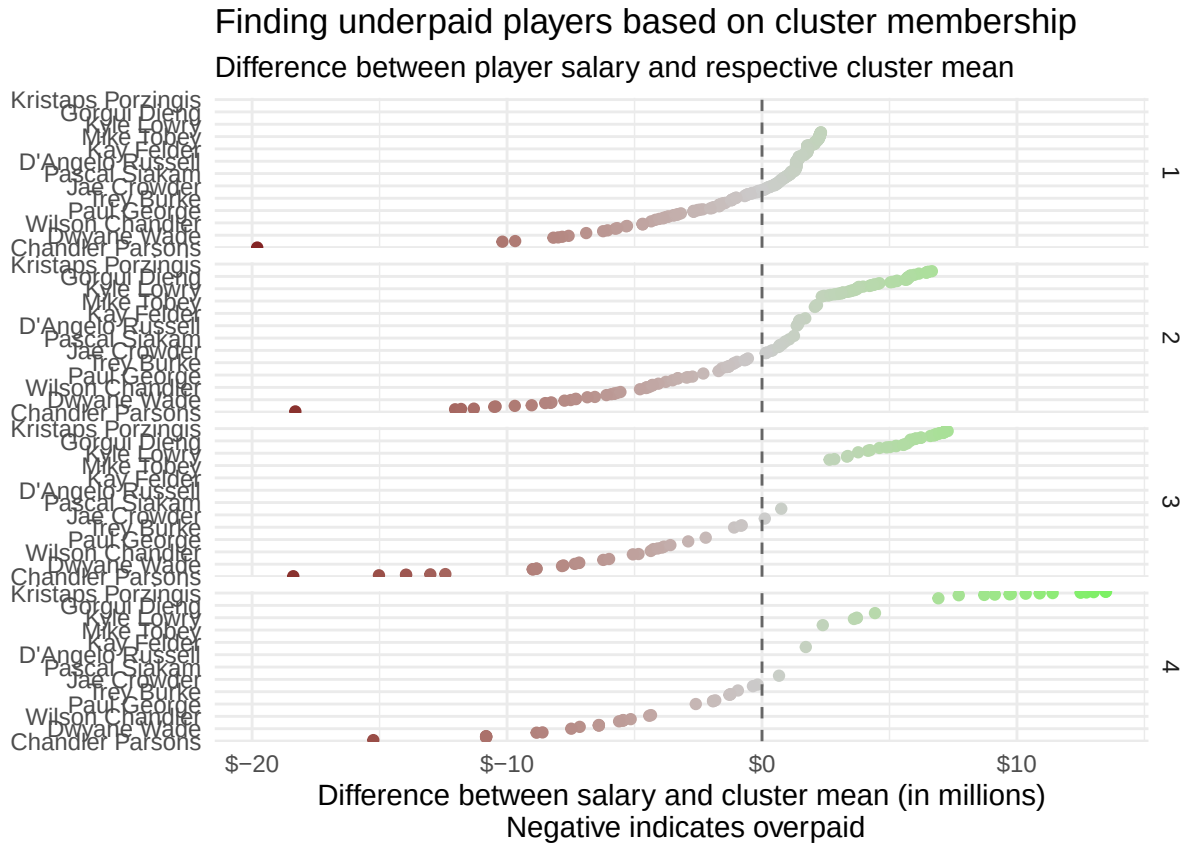
Observations There is potential to update salaries based on player tiers. For example, Chandler Parsons was paid 22M but is considered a low tier player, and is paid more than the high tier players such as Steph Curry (12M) and Kawhi Leonard (17.6M).

George to break out chart by cluster

```
nba %>%
  group_by(km_labs_four) %>%
  mutate(Clust_Salary = mean(Salary)) %>%
  ungroup() %>%
  mutate(Salary_diff = (Clust_Salary - Salary) / 1000000) %>%
  ggplot(aes(x = reorder(Player, Salary_diff), y = Salary_diff, color = Salary_diff)) +
  geom_point() +
  geom_hline(yintercept = 0,
             linetype = "dashed",
             color = "grey40") +
  scale_x_discrete(labels = players.to.show) +
  scale_y_continuous(labels = scales::dollar) +
  scale_color_gradient2(mid = "grey80", high = "green") +
  labs(title = "Finding underpaid players based on cluster membership",
       subtitle = "Difference between player salary and respective cluster mean",
       x = "",
       y = "Difference between salary and cluster mean (in millions)\n Negative indicates overpaid") +
  coord_flip() +
  theme_minimal() +
  theme(legend.position = "none")
```




```
#
nba %>%
  group_by(km_labs_four) %>%
  mutate(Clust_Salary = mean(Salary)) %>%
  ungroup() %>%
  mutate(Salary_diff = (Clust_Salary - Salary) / 1000000) %>%
  ggplot(aes(x = reorder(Player, Salary_diff), y = Salary_diff, color = Salary_diff)) +
  geom_point() +
  geom_hline(yintercept = 0,
             linetype = "dashed",
             color = "grey40") +
  scale_x_discrete(breaks = function(x) x[c(TRUE, rep(FALSE, 40 - 1))]) +
  scale_y_continuous(labels = scales::dollar) +
  scale_color_gradient2(mid = "grey80", high = "green") +
  labs(title = "Finding underpaid players based on cluster membership",
       subtitle = "Difference between player salary and respective cluster mean",
       x = "",
       y = "Difference between salary and cluster mean (in millions)\n Negative indicates overpaid") +
  coord_flip() +
  theme_minimal() +
  theme(legend.position = "none") +
  facet_grid(km_labs_four~.)
```



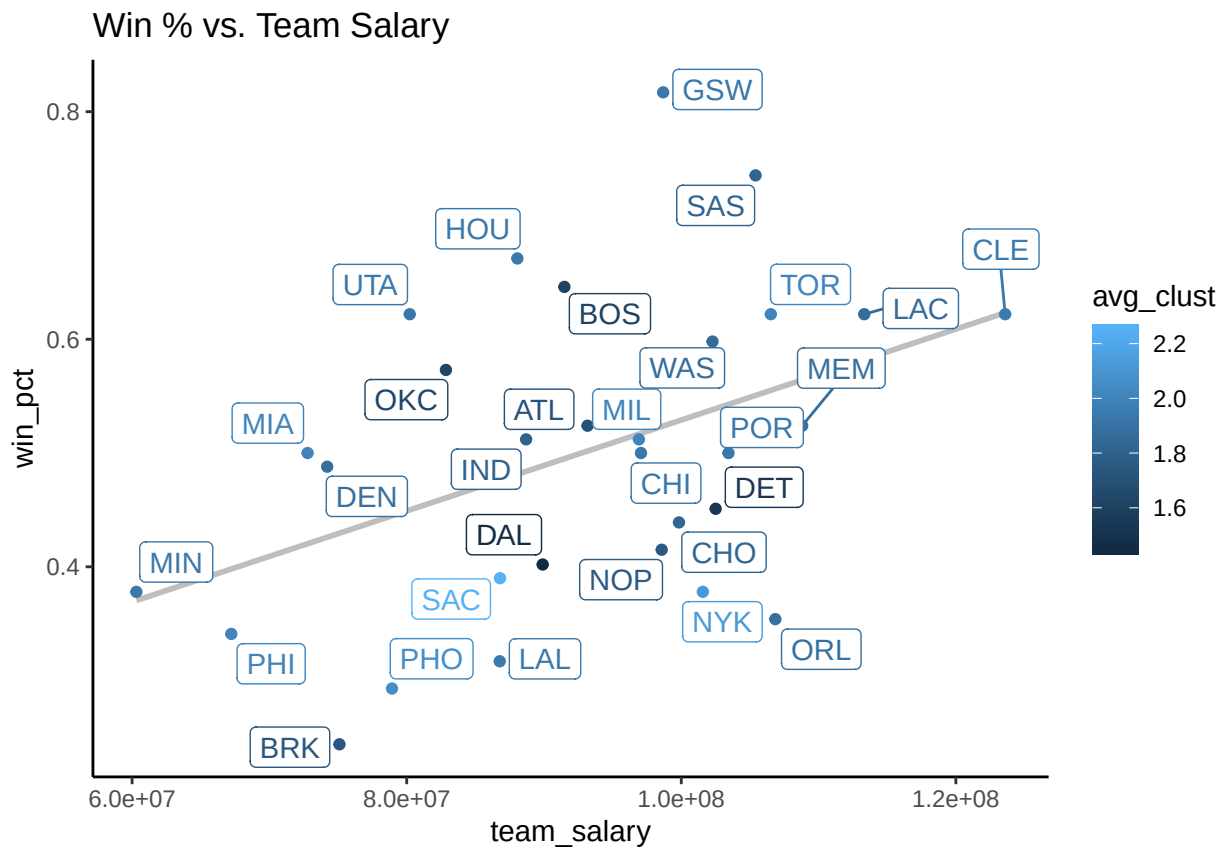
Team Compensation and Performance vs clusters

```
# 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))
                       )

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

```
##     Tm team_salary win_pct avg_clust
## 1  SAC    86799609   0.390  2.250000
## 2  NYK   101570502   0.378  2.125000
## 3  PHO    78930157   0.293  2.055556
## 4  MIA    72782449   0.500  2.000000
## 5  MIL    96913241   0.512  2.000000
## 6  PHI    67225712   0.341  2.000000
## 7  TOR   106521470   0.622  2.000000
## 8  CLE   123591014   0.622  1.941176
## 9  LAL    86775415   0.317  1.941176
## 10 CHI    97064073   0.500  1.933333
```

```
## 11 GSW    98681493   0.817  1.933333
## 12 UTA    80223193   0.622  1.933333
## 13 HOU    88062247   0.671  1.928571
## 14 MIN    60311572   0.378  1.928571
## 15 POR   103439444   0.500  1.928571
## 16 DEN    74208517   0.488  1.882353
## 17 MEM   108808118   0.524  1.882353
## 18 ORL   106849160   0.354  1.882353
## 19 LAC   113327068   0.622  1.866667
## 20 WAS   102276673   0.598  1.866667
## 21 CHO    99830531   0.439  1.823529
## 22 SAS   105410231   0.744  1.812500
## 23 IND    88698690   0.512  1.800000
## 24 NOP    98573436   0.415  1.761905
## 25 BRK    75102568   0.244  1.736842
## 26 ATL    93172774   0.524  1.705882
## 27 OKC    82858524   0.573  1.625000
## 28 BOS    91484921   0.646  1.600000
## 29 DET   102503259   0.451  1.533333
## 30 DAL    89904500   0.402  1.450000
```

```r
# plot
## ANDREW TO FORMAT SALARY #
ggplot(nba_team,
       aes(x = team_salary, y = win_pct, color = avg_clust)) +
  geom_smooth(method = 'lm', formula = y ~ x, se = FALSE, color = 'gray') +
  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            1
## 98      Stephen Curry 24.6            4
## 119      Kevin Durant 27.6            4
## 164   Draymond Green 16.5            3
## 211   Andre Iguodala 14.4            2
## 235     Damian Jones  5.3            1
## 268 Shaun Livingston 10.1            1
## 270     Kevon Looney 13.4            1
## 285    James Michael 13.0            1
## 286    Patrick McCaw  8.6            1
## 293     JaVale McGee 25.2            1
## 344    Zaza Pachulia 16.1            3
## 427    Klay Thompson 17.4            4
```

```
## 443 Anderson Varejao  9.4            1
## 457       David West 16.6            1
```

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 %.