

ML Final Project: Analysis

Andrew Pagtakhan, Joe Marlo, Bilal Waheed, George Perrett

January 20, 2020

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)

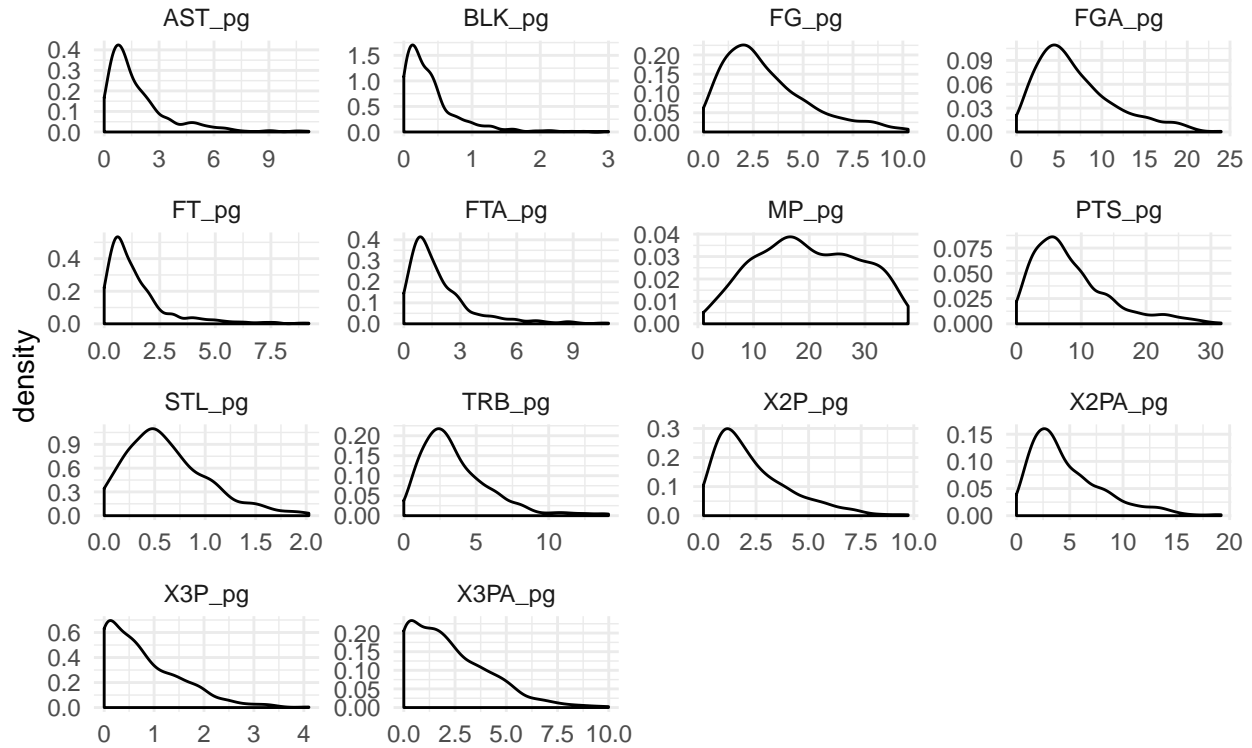
Exploratory Data analysis

```
# 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.

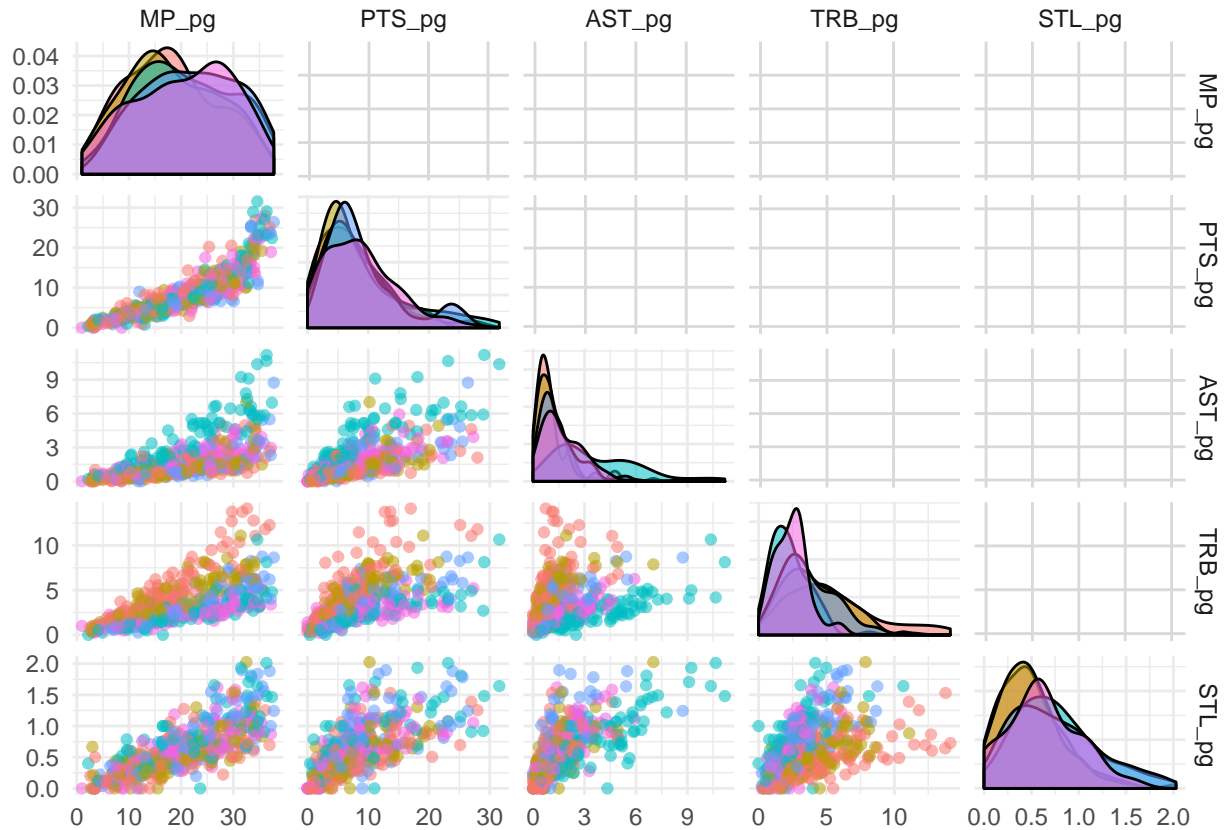
Feature Densities (Untransformed)



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  TRB_pg
##  82.07   4.67  20.49   0.57   3.76   3.23  11.89   2.04   2.99   5.89
##  AST_pg  STL_pg  BLK_pg  PTS_pg
##   3.11   0.17   0.17  36.72
```



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      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
```

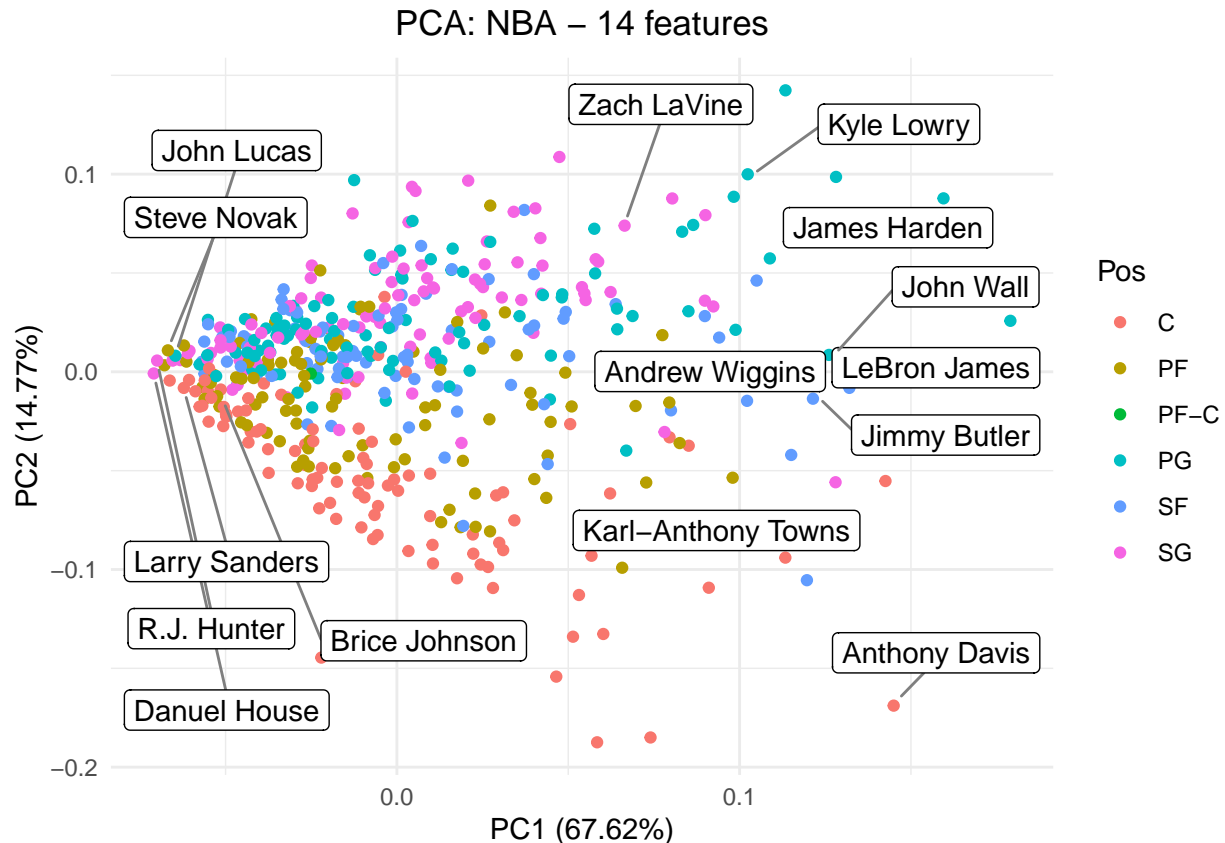
```
## Cumulative Proportion  0.99658 0.99858 0.99941 1.00000 1.000e+00 1.000e+00
##                               PC14
## Standard deviation      1.597e-15
## Proportion of Variance  0.000e+00
## Cumulative Proportion  1.000e+00

# 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

```
# 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 noticeable 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

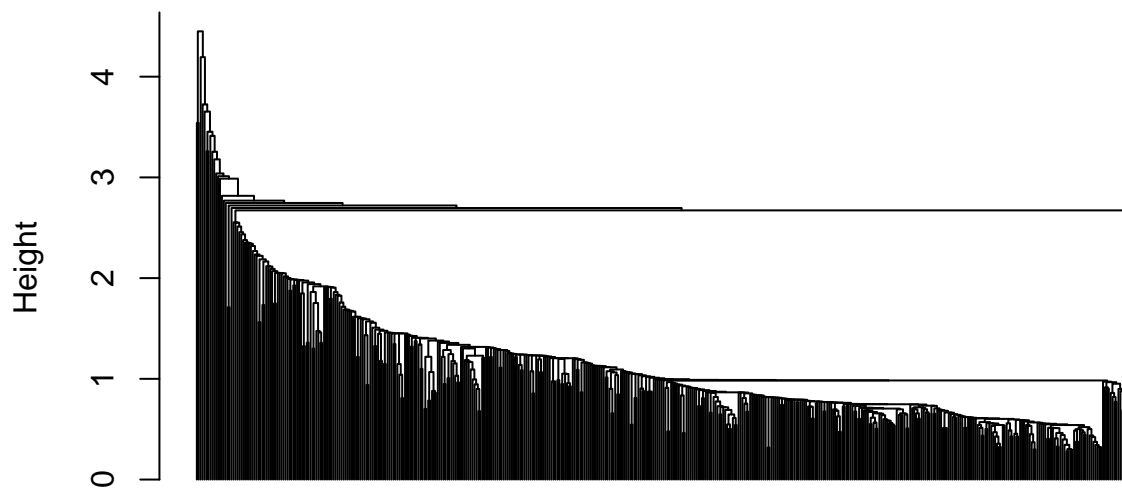
The first method of clustering we will try is hierarchical clustering. The dendrogram can help provide a visual aid in the number of clusters we can start to use.

```
# 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')

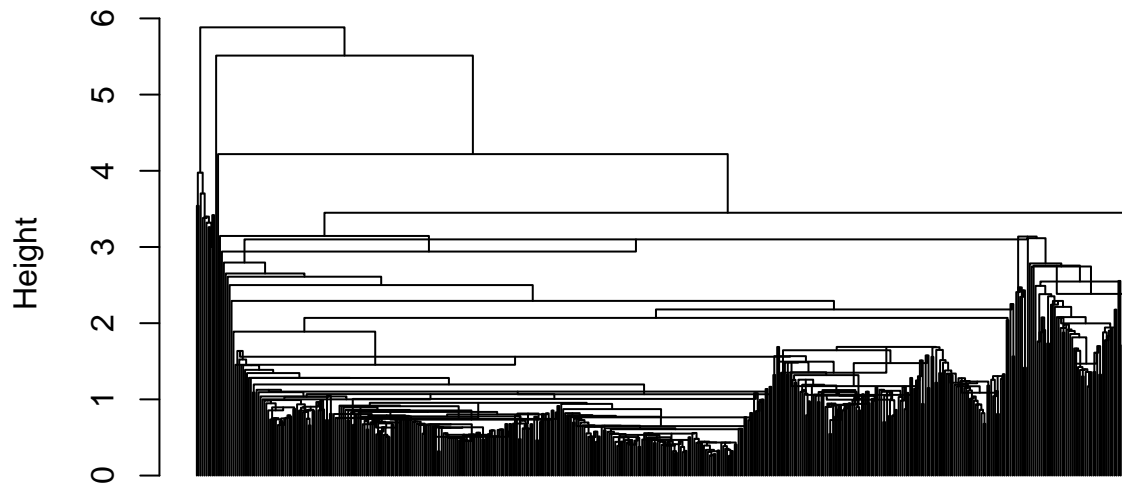
# nearest neighbors method
plot(hcl_single, hang = -1, main = 'Single Linkage',
     labels = FALSE, xlab = '', sub = '')
```

Single Linkage



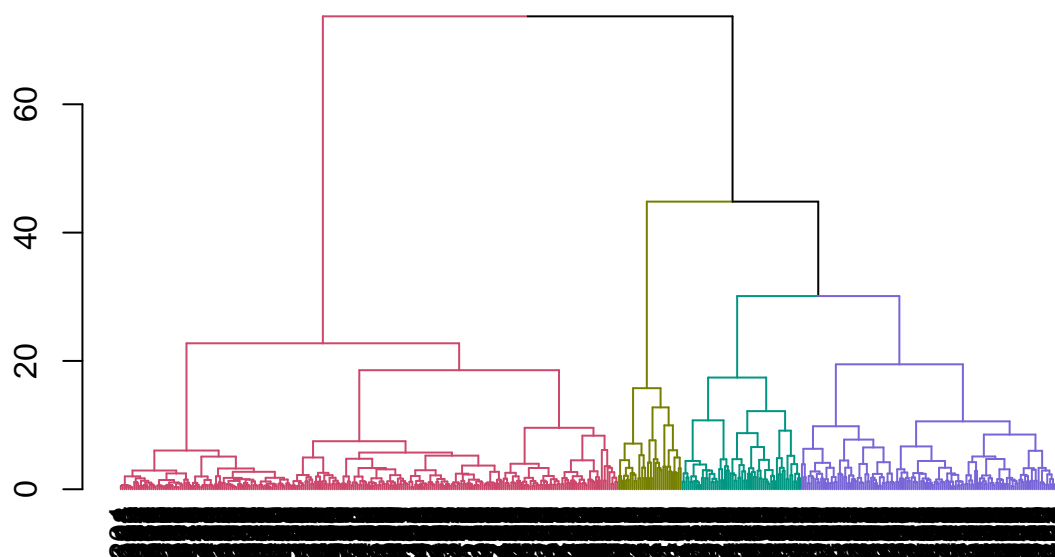
```
# groups centroid  
plot(hcl_centroid, hang = -1, main = 'Centroid Linkage',  
     labels = FALSE, xlab = '', sub = '')
```

Centroid Linkage



```
# 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))
```

Ward (D2) Linkage: K = 4



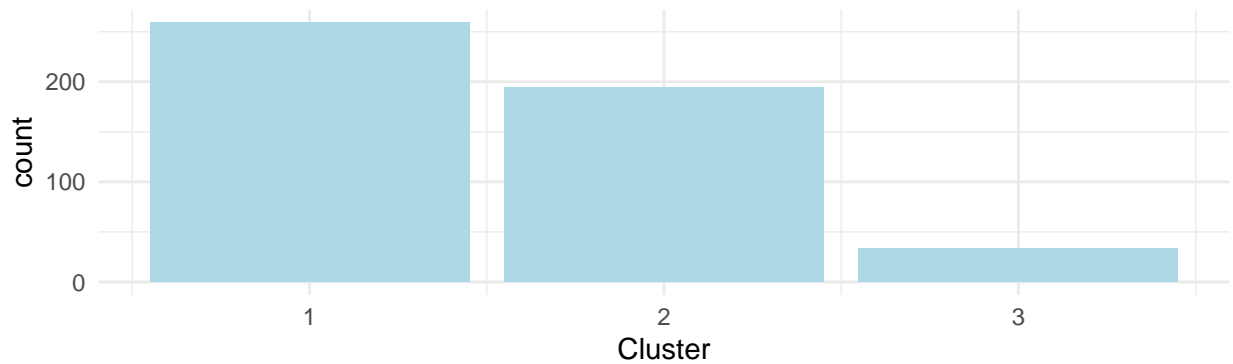
Since the Ward dendrogram seems to be the best among the three, we will look at its distribution for 3 and 4 clusters. We chose these initial groupings because this is a good number of initial clusters to group NBA players.

```
# 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)

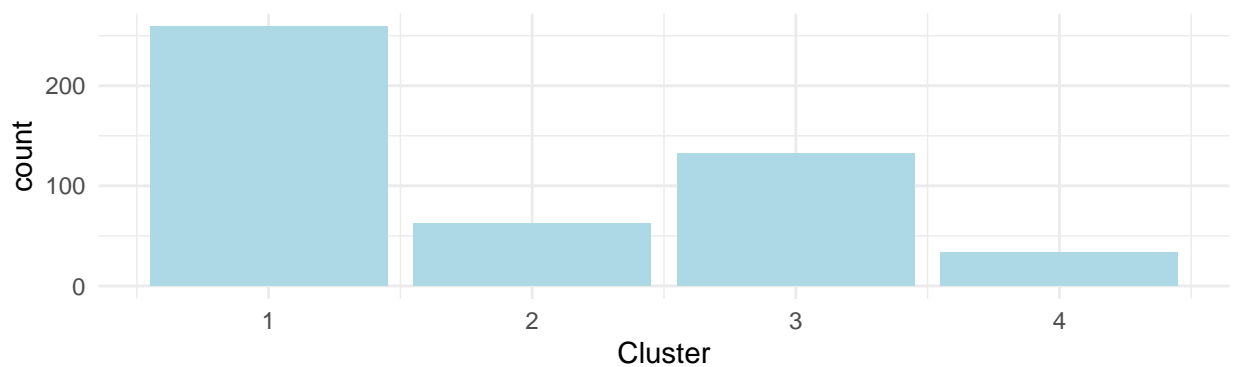
# plot frequencies
p_one <- ggplot(data = nba, aes(x = hcl_ward_labs_three)) +
  geom_bar(fill = 'lightblue') +
  ggtitle('Number of Players by Cluster: K = 3') +
  xlab('Cluster')
p_two <- ggplot(data = nba, aes(x = hcl_ward_labs_four)) +
  geom_bar(fill = 'lightblue') +
  ggtitle('Number of Players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

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


Number of Players by Cluster: K = 3



Number of Players by Cluster: K = 4



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))

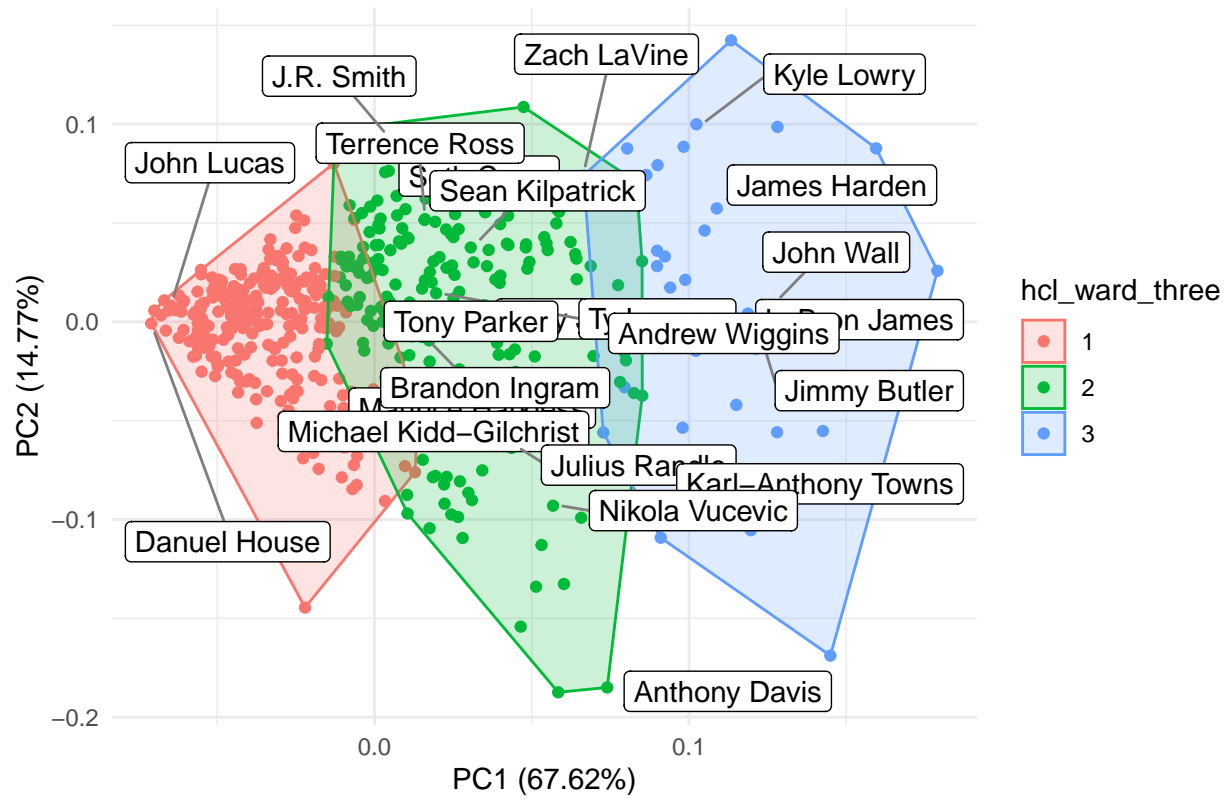
# player names to include in plot
hcl_labels <- ifelse(nba$MP_pg >= 36 | nba$MP_pg <= 2.5 |
  (nba$MP_pg >= 28.8 & nba$MP_pg <= 29) |
  (nba$MP_pg >= 25 & nba$MP_pg <= 25.2),
  as.character(nba$Player), '' )

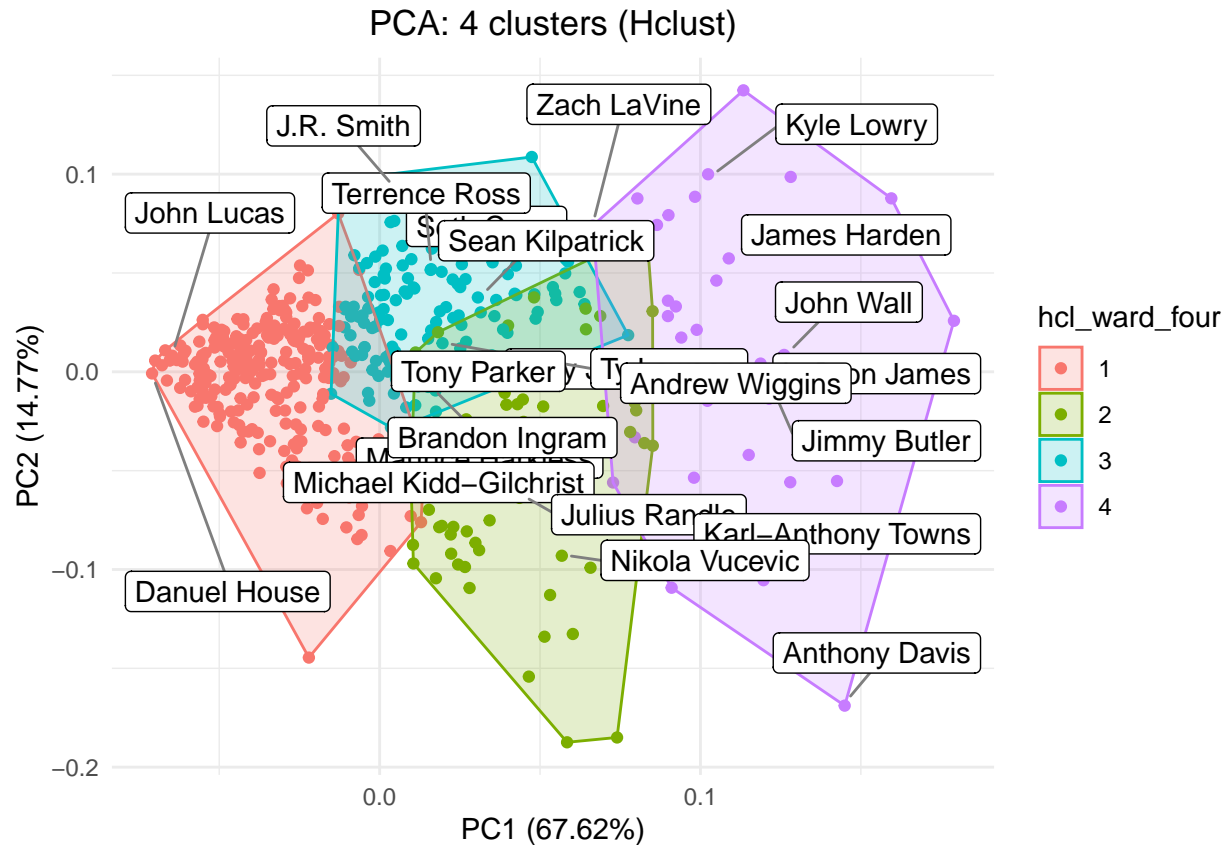
# elements to loop over
hcl_labs <- names(nba %>% select(tail(names(.), 2)))
hcl_ks <- c(3, 4)

# 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)





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.

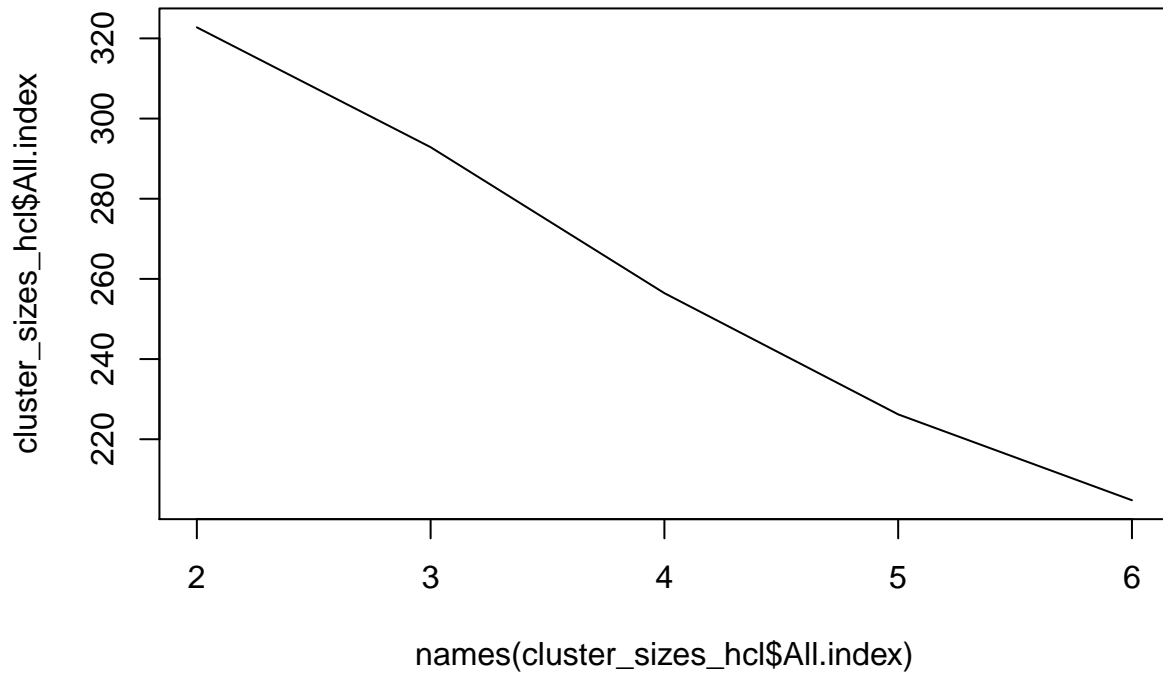
Optimize number of clusters

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')

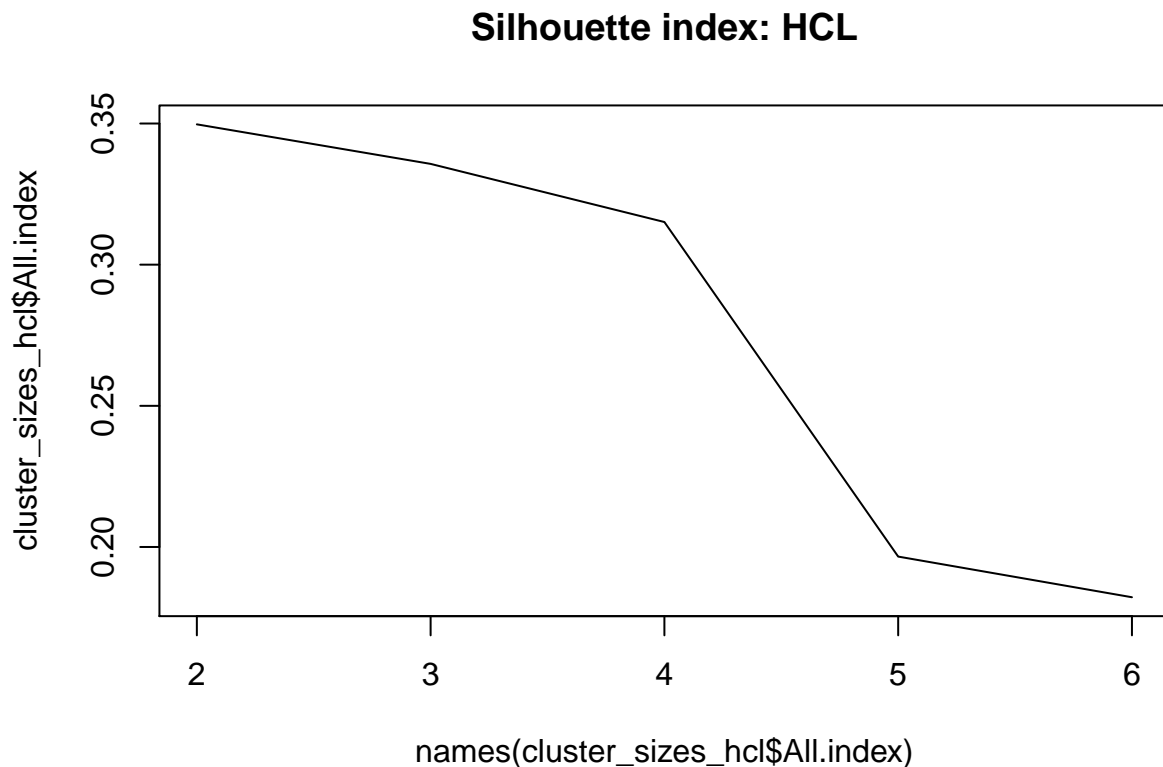
# plot C(G)
plot(names(cluster_sizes_hcl$All.index),
     cluster_sizes_hcl$All.index,
     main = 'Calinski-Harabasz index: HCL',
     type = 'l')
```

Calinski–Harabasz index: HCL



```
# 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 dendrogram 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 have differences between groups of players. But, it is possible that NBA front offices will likely need more differentiation 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

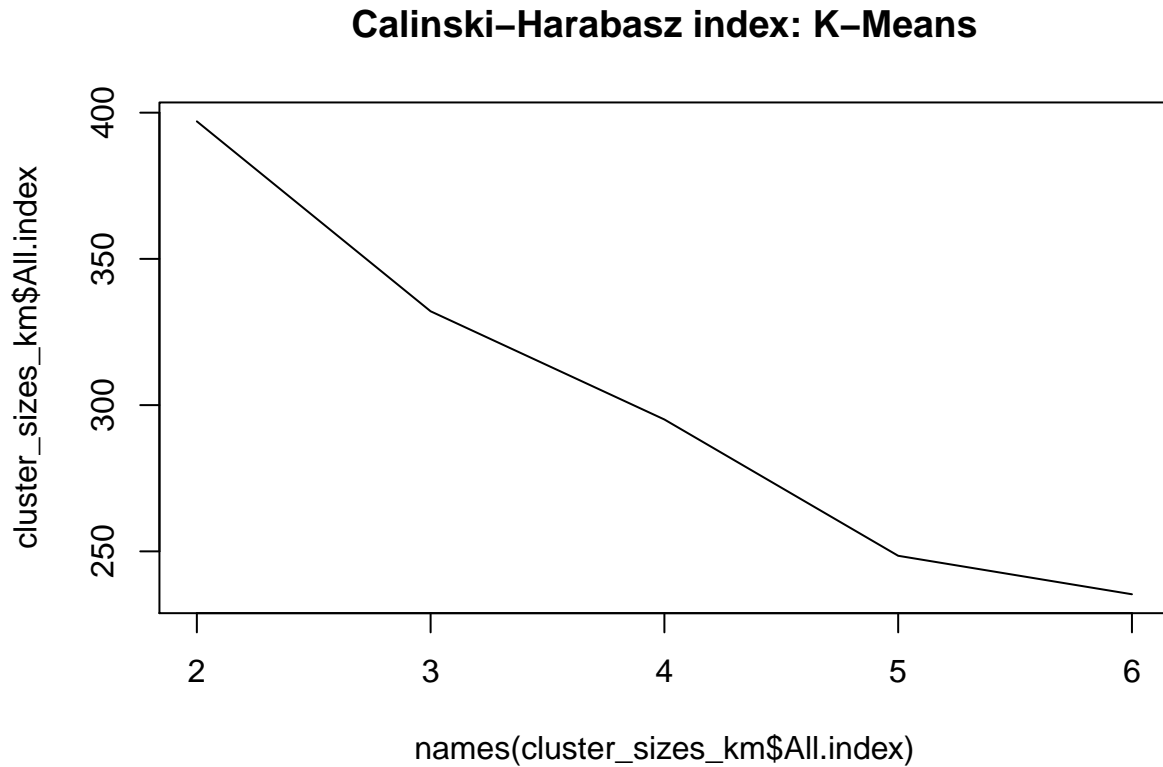
Optimize number of clusters

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')

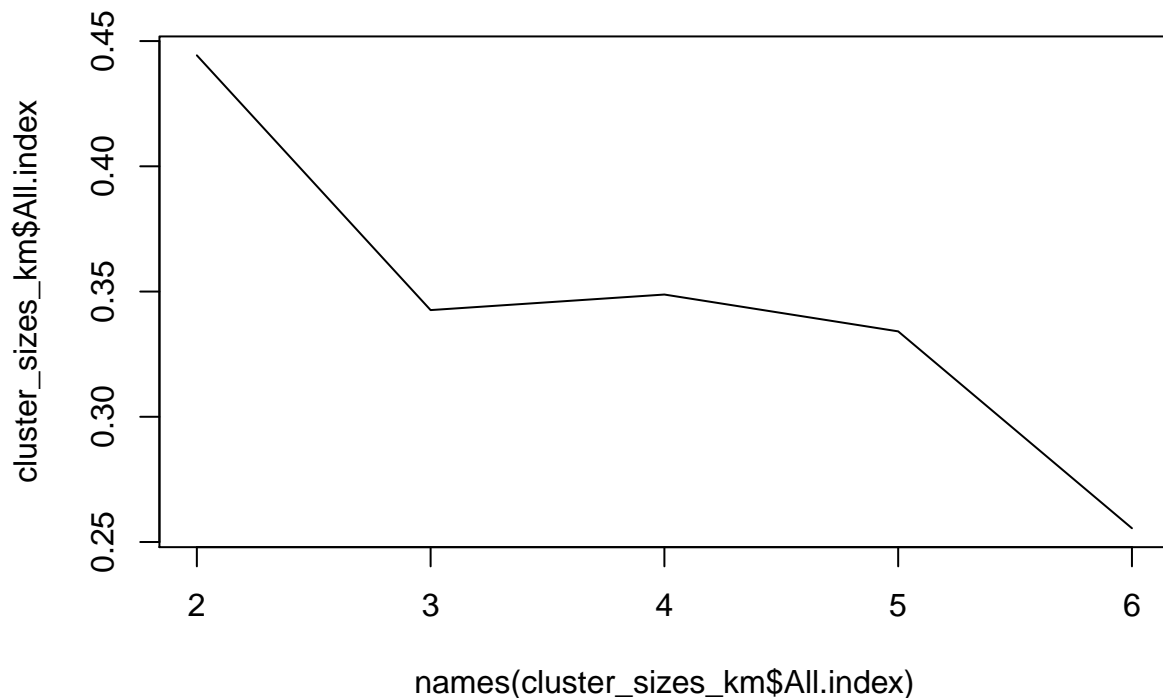
# plot C(G)
plot(names(cluster_sizes_km$All.index),
```

```
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')
```

Silhouette index: K-Means



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.

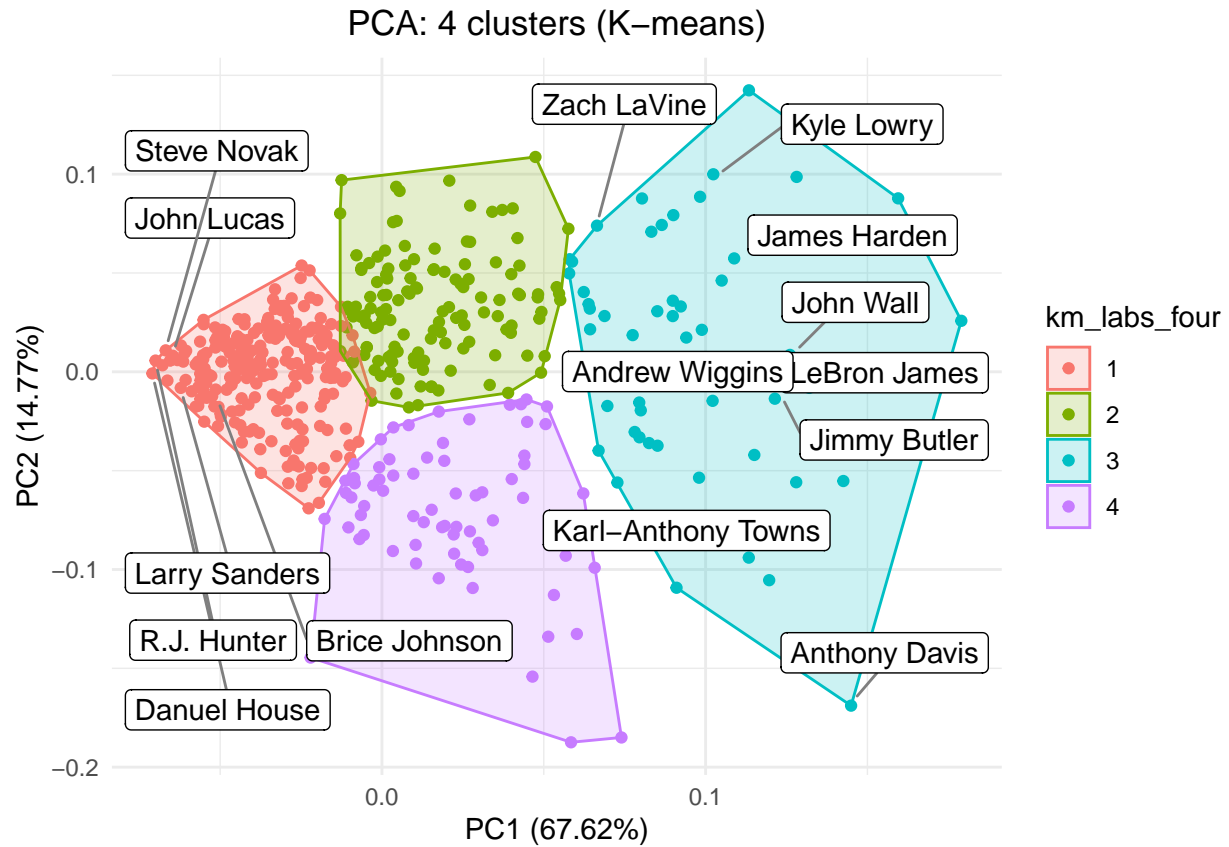
K-means clustering with 4 groups

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

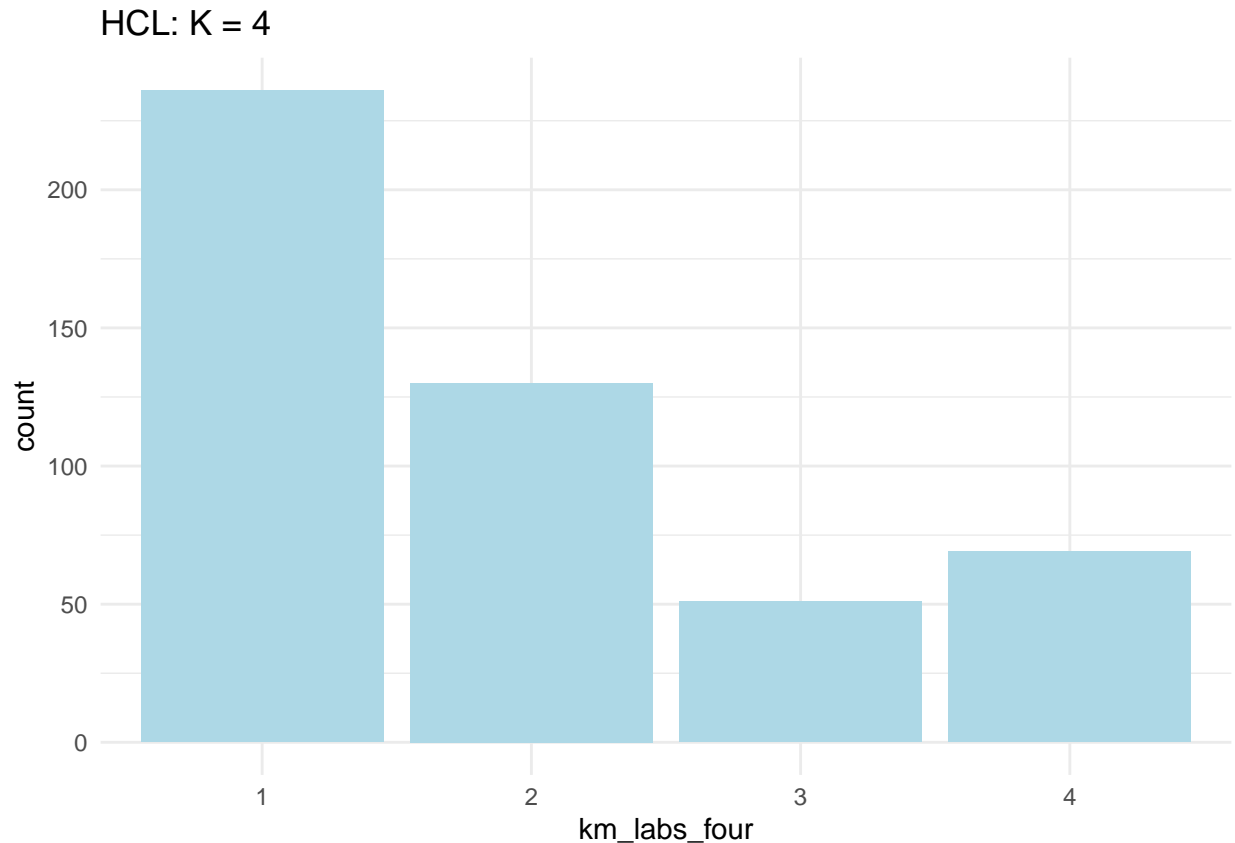
nba$km_labs_four <- factor(km_four$cluster)

# 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.



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	33.85906	21.82156	5.724658	4.7255814	0.6158290	1.1623242
## 4	4	24.96591	10.29999	7.010810	1.6435808	0.9473312	0.7868895
##		VORP	PER	RPM			
## 1		-0.06694915	10.18941	-1.9458898			
## 2		0.50153846	12.75308	-0.8625385			
## 3		3.19215686	21.29020	2.5100000			

```
## 4 1.26811594 17.09710 0.3679710
```

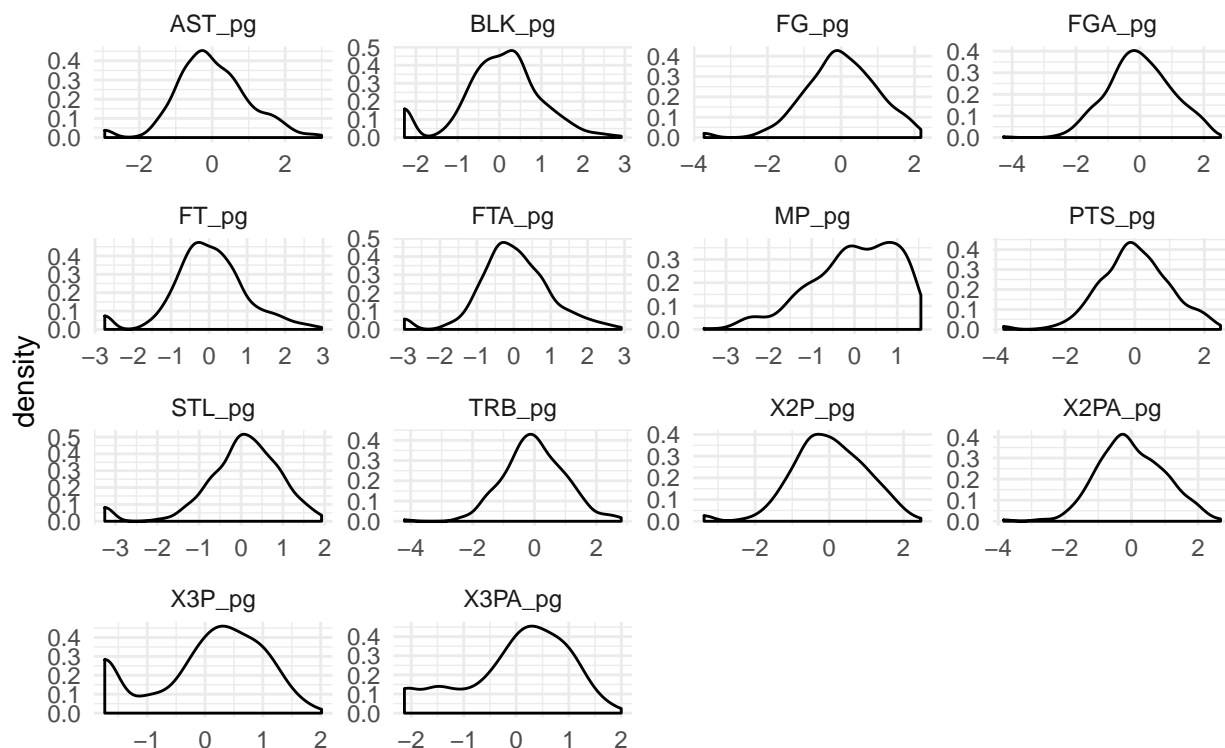
Observations Although the CH index indicates that the optimal number of clusters is 2, this seems too low of a number to meaningfully break out the NBA players into groups. It is also important to note that the CH index is a heuristic method. So although CH is a good approach to look for the number of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

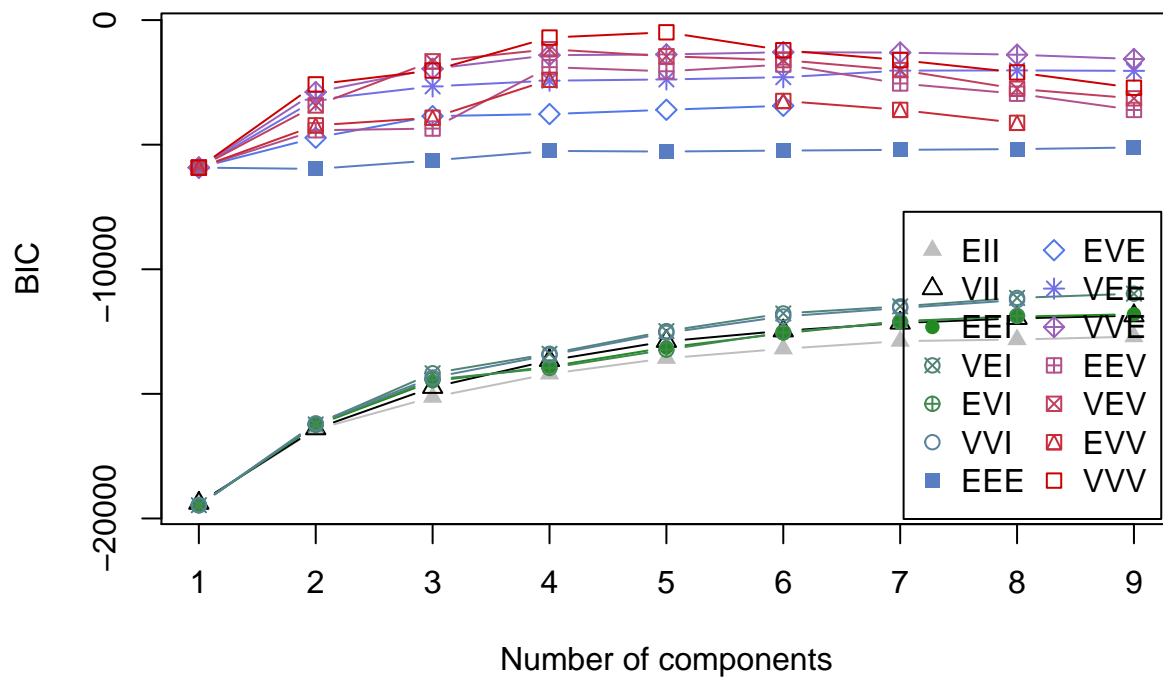
Feature Densities (Transformed)



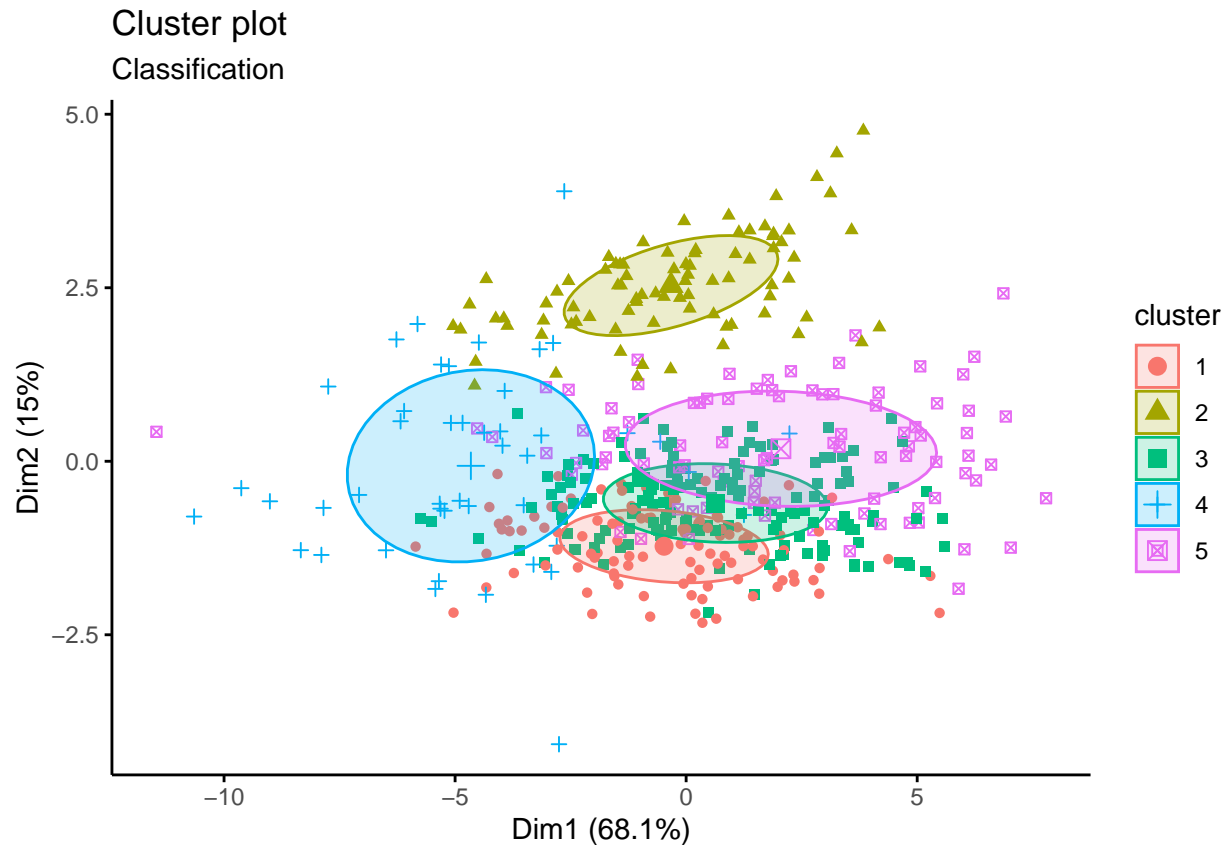
```
# run model
player_clust.mcl <- Mclust(nba_feat_cr_sc)
summary(player_clust.mcl)
```

```
## -----
## 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    0   24
##                2   0    8   12   42
##                3   5   118    6    3
##                4   0    0   33    0
```

```
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  3  0 16  0 32
##          4  0 44  7  2 16
```

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.

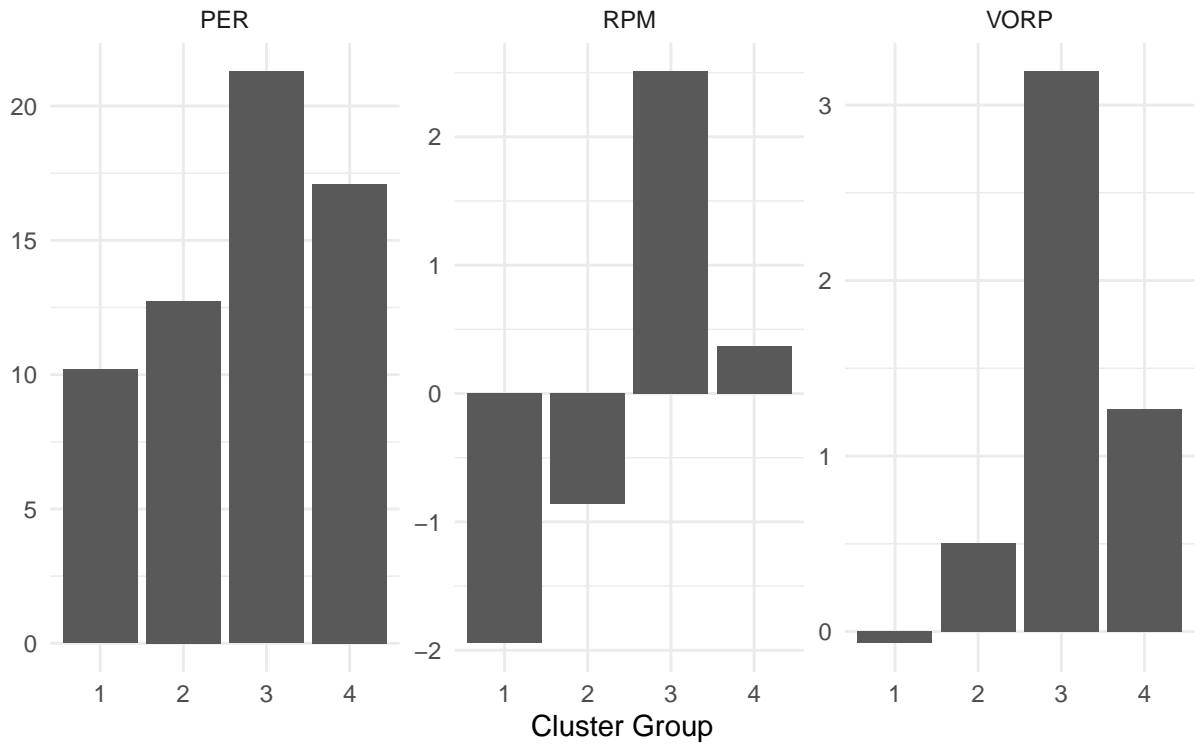
Final Cluster selection

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.

Overall Player Performance by Cluster

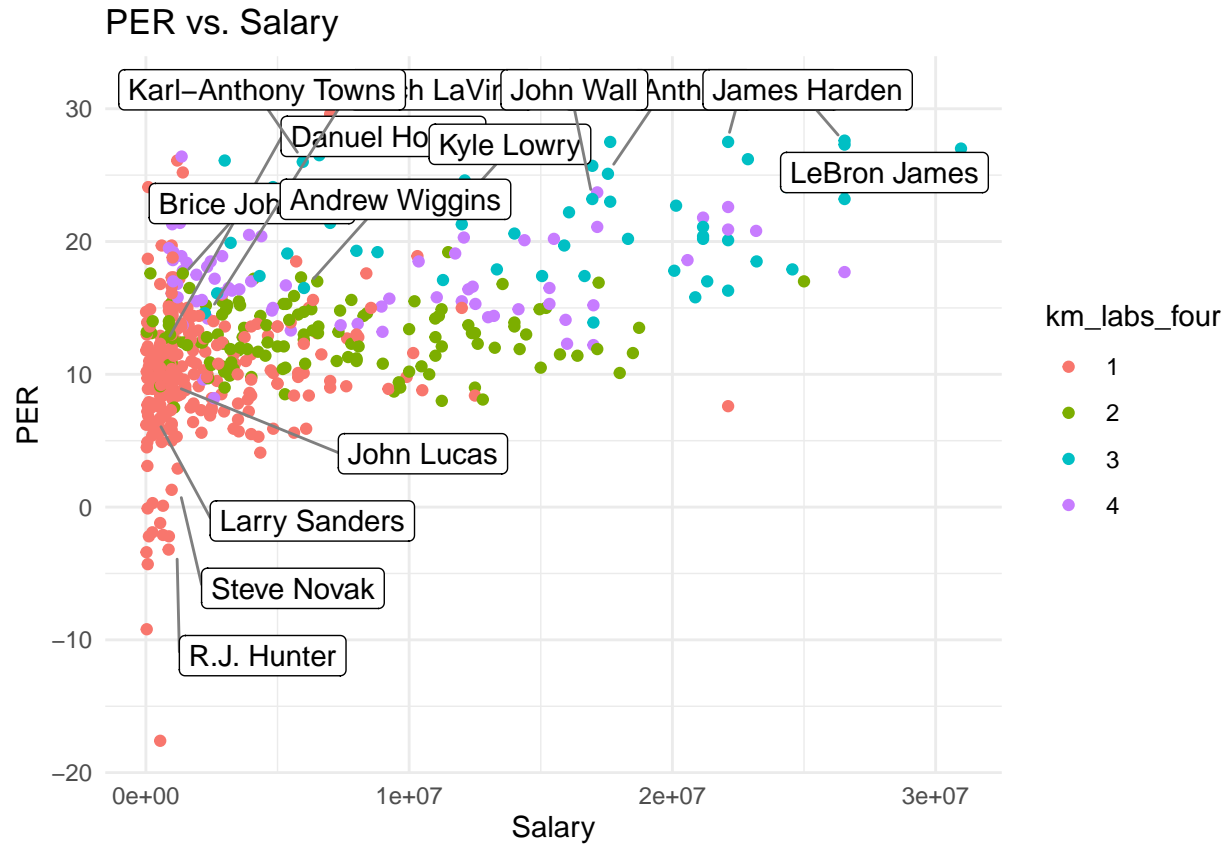
Average Advanced Statistics



Post-Cluster Analysis

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

Clusters vs. Player Salaries



Highest Paid players in Lowest Tier

```
head(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 19.85294 MEM 22116750 7.6          1
## 2 Miles Plumlee 45 10.75556 MIL 12500000 8.4          1
## 3 Amir Johnson 80 20.10000 BOS 12000000 15.0         1
## 4 Mirza Teletovic 70 16.18571 MIL 10500000 8.8          1
## 5 Al Jefferson 66 14.10606 IND 10314532 18.9         1
## 6 Alec Burks 42 15.54762 UTA 10154495 11.6         1
```

Lowest Paid players in highest tier

```
head(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 Alan Williams 47 15.06383 PHO 874636 19.5          4
```

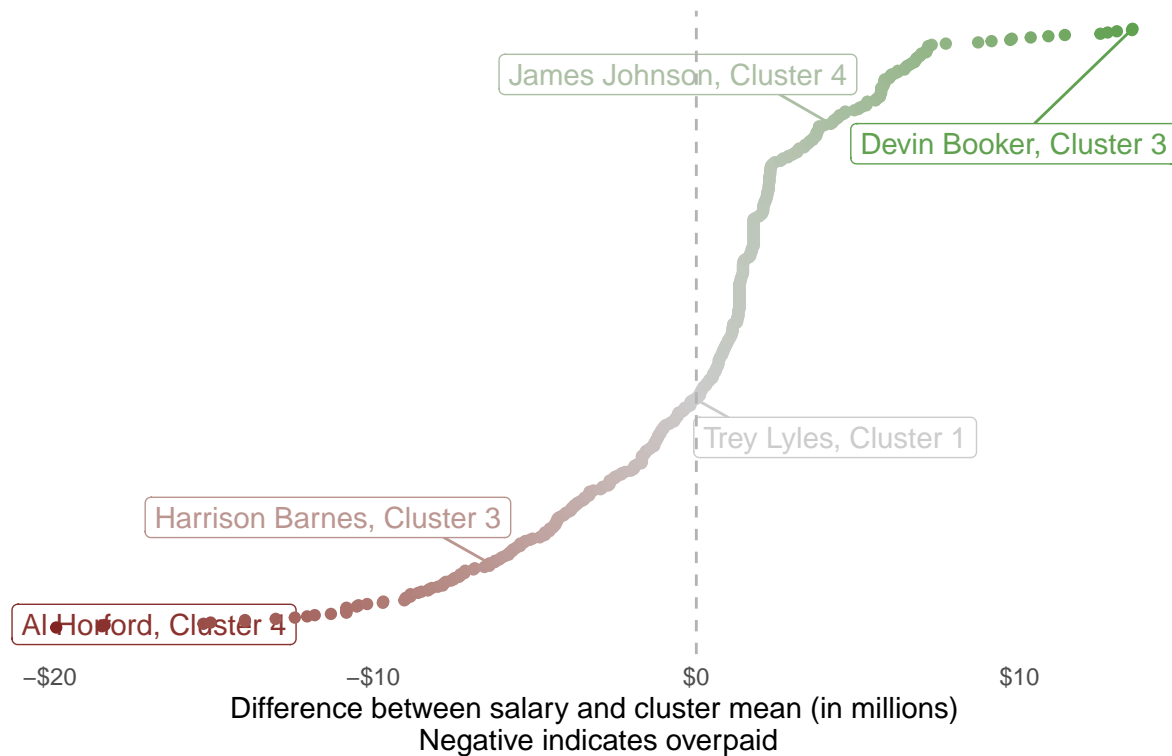
```
## 2 JaMychal Green 77 27.28571 MEM 980431 13.5 4
## 3 Terrence Jones 54 23.51852 NOP 980431 16.1 4
## 4 Edy Tavares 2 14.00000 CLE 1000000 21.3 4
## 5 Richaun Holmes 57 20.92982 PHI 1025831 18.6 4
## 6 Ivica Zubac 38 16.02632 LAL 1034956 17.0 4
```

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

Finding underpaid players based on cluster membership

Difference between player salary and respective cluster mean salary



Team Compensation and Performance vs clusters

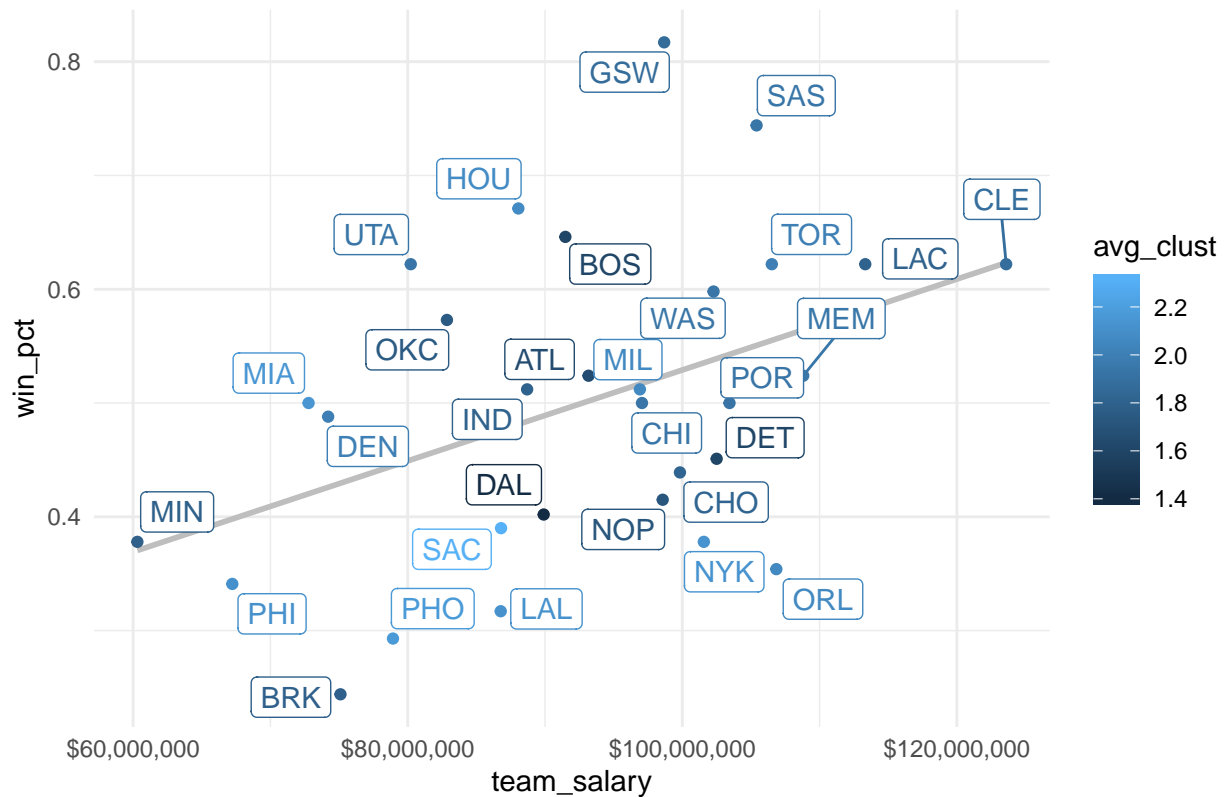
```
# convert labels to numeric
nba$cl_lab_numeric <- as.numeric(nba[, cl_label])
nba_team <- data.frame(nba
  %>% select(Tm, Salary, win_pct, cl_lab_numeric)
  %>% group_by(Tm)
  %>% summarise(team_salary = sum(Salary),
    win_pct = mean(win_pct),
    avg_clust = mean(cl_lab_numeric))
)
```



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

##	Tm	team_salary	win_pct	avg_clust
## 1	SAC	86799609	0.390	2.312500
## 2	PHO	78930157	0.293	2.166667
## 3	MIA	72782449	0.500	2.142857
## 4	NYK	101570502	0.378	2.125000
## 5	LAL	86775415	0.317	2.117647
## 6	PHI	67225712	0.341	2.111111
## 7	HOU	88062247	0.671	2.071429
## 8	MIL	96913241	0.512	2.062500
## 9	ORL	106849160	0.354	2.058824
## 10	DEN	74208517	0.488	2.000000
## 11	TOR	106521470	0.622	2.000000
## 12	MEM	108808118	0.524	1.941176
## 13	SAS	105410231	0.744	1.937500
## 14	CHI	97064073	0.500	1.933333
## 15	UTA	80223193	0.622	1.933333
## 16	WAS	102276673	0.598	1.933333
## 17	POR	103439444	0.500	1.928571
## 18	CLE	123591014	0.622	1.882353
## 19	GSW	98681493	0.817	1.866667
## 20	CHO	99830531	0.439	1.823529
## 21	IND	88698690	0.512	1.800000
## 22	LAC	113327068	0.622	1.800000
## 23	BRK	75102568	0.244	1.789474
## 24	MIN	60311572	0.378	1.785714
## 25	OKC	82858524	0.573	1.750000
## 26	NOP	98573436	0.415	1.714286
## 27	ATL	93172774	0.524	1.647059
## 28	BOS	91484921	0.646	1.600000
## 29	DET	102503259	0.451	1.600000
## 30	DAL	89904500	0.402	1.400000

Win % vs. Team Salary



```
# 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           3
## 119   Kevin Durant 27.6           3
## 164   Draymond Green 16.5          4
## 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          4
## 427    Klay Thompson 17.4          3
```

## 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 %.