ML 2011 Final Project

Joe Marlo, Andrew Pagtakhan, George Perrett, Bilal Waheed

In Loving Memory of Kobe Bryant: 1978 - 2020

Contents

R version 3.5.1 (2018-07-02) Platform: x86_64-w64-mingw32/x64 (64-bit) Running under: Windows >= 8 x64 (build 9200)

1 Resources for reproducing this project

The repository containing the raw and cleaned data along with all code required to run the analyses can be found at https://github.com/joemarlo/ML-NBA.

The original source of the data can be found at these links:

- https://www.kaggle.com/drgilermo/nba-players-stats#player_data.cs
- http://www.espn.com/nba/salaries/_/year/2017
- http://www.espn.com/nba/statistics/rpm/_/year/2017

2 Introduction

2.1 Research question

Are there underlying patterns of groupings between NBA team compensation vs. overall team skill-sets?

2.2 Group member roles

We all collaboratively contributed to the project in all aspects. Specifically, Joe contributed to scraping the player salary, additional player features, and visualizations. Andrew contributed to cleaning the data, hierarchical/K-means models, and post-cluster analysis. Bilal contributed to the hierarchical/K-Means modeling and the write-up. George contributed to the model-based clustering and data transformation.

2.3 Data description

The project is an unsupervised approach to discover underlying patterns or groupings between NBA compensation vs. overall team skill-sets. It uses K-means, hierarchical, and model-based clustering along with other techniques and tools such as principal component analysis, standardizing, scaling, and web-scraping.

The data includes NBA statistics for over 3,000 players, 60+ seasons, and over 50 features per players. The measurement scales are numerical totals and percentages for features such as: points, assists, rebounds, player attributes (height, weight, college attended, etc.). The full dataset represents a time period of 1950-2017, however we are selecting data for the 2016-2017 season.

Key feature set:

- MP_pg: Minutes played per game
- FG_pg: Field goals made per game
- FGA_pg: Field goal attempts per game
- 3P_pg: 3-point shots made per game
- 3PA_pg 3-points shot attempts per game
- 2P_pg: 2-point shots made per game
- 2PA_pg: 2-point shots attempted per game
- FT_pg: Free throw shots made per game
- FTA_pg: Free throw shots attempted per game
- TRB_pg: Total rebounds (offensive + defensive) per game
- AST_pg: Total assists per game
- STL_pg: Total steals per game
- BLK_pg: Total blocks per game
- PTS_pg: Total points made per game

Additional features:

- win_pct: team winning percentage during the regular season (Total wins / total games played)
- Player: First and last name of NBA player
- Team: NBA team player played on. For multiple teams, this is the team the player played on for the most minutes
- Position: NBA position, e.g. C = Center, PF = Power Forward, SG = Shooting Guard, PG = Point Guard
- Salary: Player salary (USD)
- RPM: Real Plus/Minus. Player's average impact in terms of net point differential per 100 offensive and defensive possessions
- VORP: Value Over Replacement Player. Measure to estimate each player's overall contribution to the team
- PER: Player Efficiency Rating. Measures player's overall contributions across different statistics.

2.4 Analysis methods

For this analysis, we applied the following modeling techniques to analyse NBA player data:

- Principal Component Analysis
- Hierarchical Clustering
- K-Means
- Model-Based clustering

2.5 Data exploration and transformation

The data cleaning was done in separate R scripts which can be found within the repo/Data 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 and standardizing. For model-based clustering, various transformations such as Log + 1, square root, cube root, and Box-Cox were applied. The cube root yielded the best transformation to normalize the data. This was validated using QQ plots. We scaled the data for Hierarchical and K-Means to put all features on an equally weighted basis. For example, minutes played per game vs. blocks per game are on different ranges, so scaling these puts them on a comparable basis.

3 Analysis

3.1 Exploratory data analysis

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

# set for reproducible results
set.seed(14)
# set theme for ggplot
theme_set(theme_minimal())
# clear variables
rm(list = ls())
```

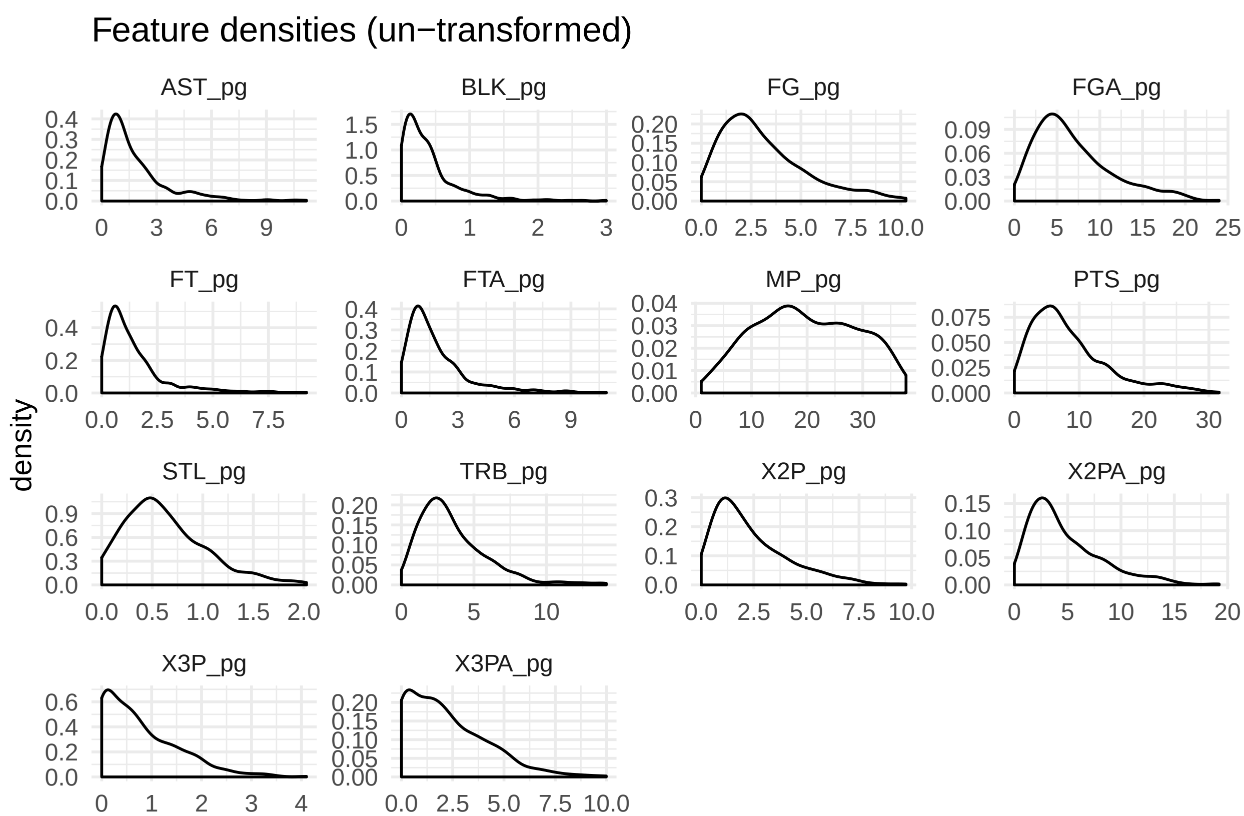
We decided to use the 14 features in our analysis (on a per game basis) because they provide a good balance of offensive (e.g Pts, Ast) and defensive stats (e.g. Reb, Blk). This is more likely to provide more balanced groupings between those who are more offensive and those who are better at defense.

With these features, we will explore the distribution of each future by plot the distributions of key per game stats.

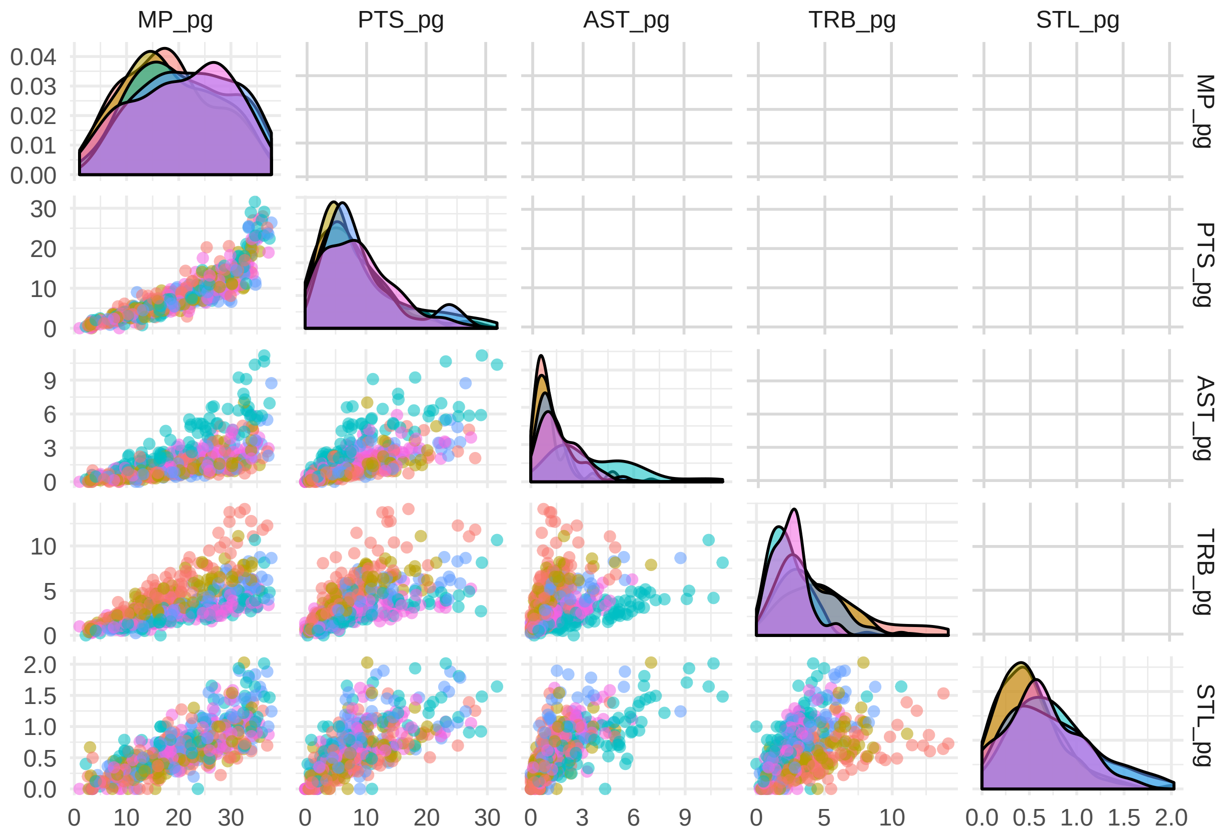
Table 1: Feature Variance

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

```r
# load data
nba <- read.csv('Data/season_stats_clean.csv')
# replace NA values in RPM column with 0s
nba$RPM[is.na(nba$RPM)] <- 0
```



Variance was examined to provide a numerical estimate of the spread of each feature to determine if scaling is necessary. As an exploratory analysis, bi-variate comparisons were made between all features to determine if there are any groupings between variables. A subset of the features is presented below.



Due to the discrepancy in spread between features we decided to scale the data. Scaling each feature ensures that no single feature will dominate subsequent analyses as a result of the way the feature was measured. A good example is minutes and blocks per game - most players will have more minutes per game than blocks per game.

```
# scale the data
nba_feat_sc <- scale(nba_feat)
```

3.2 Principal component analysis

Before running any clustering algorithms, we will perform principal component analysis to determine if there are any inherent groupings among players.

The first two principal components explain the majority of the variance in the feature set. We will plot the data in these two dimensions to better assess player similarities.

```
# run PCA
nba_pca <- prcomp(nba_feat_sc)
summary(nba_pca)
```

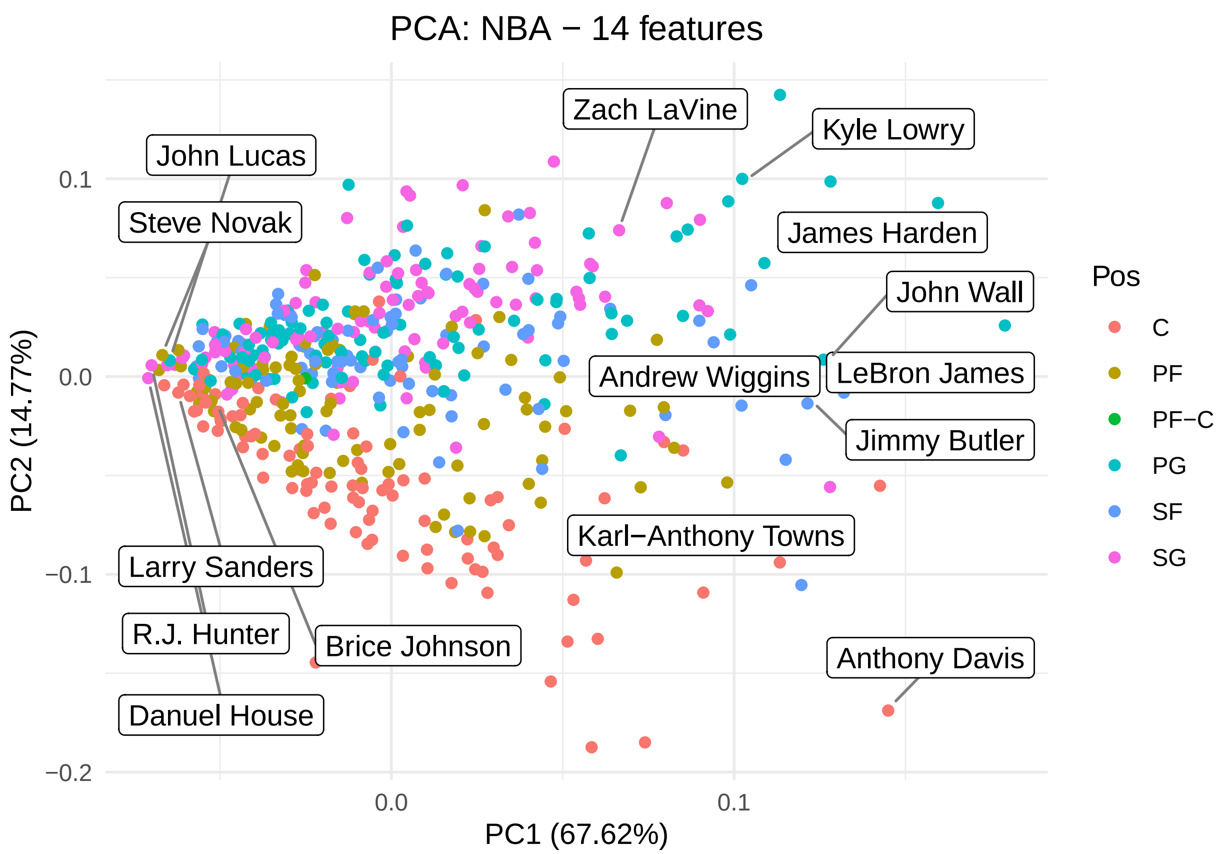
```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6
## Standard deviation     3.0767 1.4379 0.88672 0.80819 0.63364 0.52061
## Proportion of Variance 0.6762 0.1477 0.05616 0.04666 0.02868 0.01936
## Cumulative Proportion  0.6762 0.8239 0.88002 0.92667 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
```

```
# create a function that will plot PCA
plot_pca <- function(object, frame = FALSE, x = 1, y = 2,
                     data, colour, title, label) {#, leg_title) {
  # 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')# +
    ## This is supposed to override the autoplot legend title.
    ## Only works when plotting PCA directly. Does not work for HCL and KM objects
    #labs(colour = leg_title)
  return(p)
}
```

At first glance, there are differences between players based on overall statistics. For example, LeBron James and James Harden are near each other, indicating star players may be grouped together. There are also similarities based on player position. For example, Centers/Power Forwards such as Anthony Davis and Karl-Anthony Towns are in the bottom of the chart.

Examining the PCA in three-dimensional space shows a 'cone' shape spread of the data similar to the two-dimensional 'fan' shape.



3.3 Clustering

3.3.1 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.

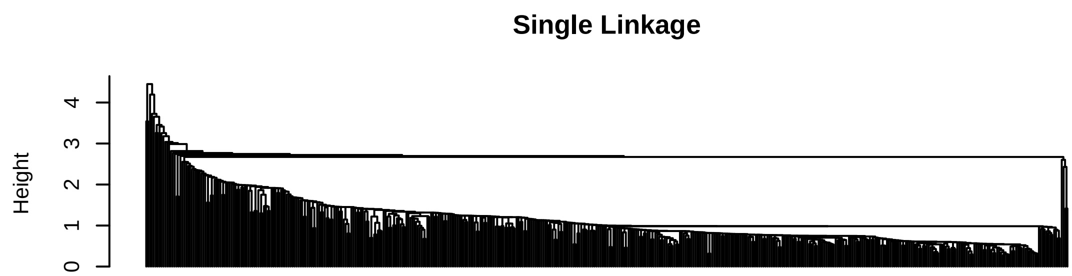
```
# distance matrix for features
nba_dist_sc <- dist(nba_feat_sc, method = 'euclidean')

# 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')
```

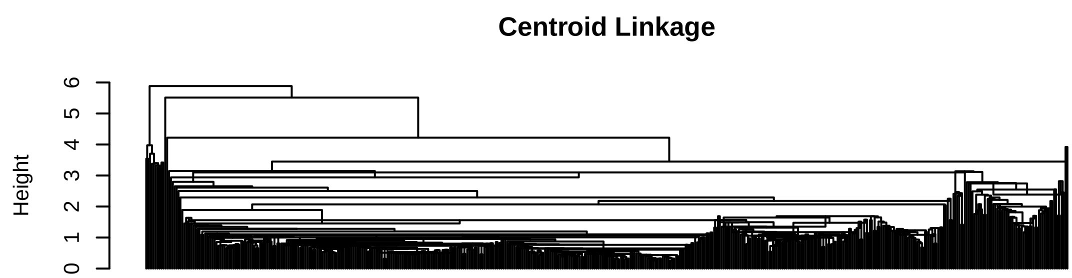
Of the three methods, the Ward dendrogram appears to be the best because there is clear separation between the clusters. It is difficult to determine a clean cutpoint to separate clusters within the single- and centroid-linkage methods. Moreover, the single- and centroid-linkage approaches produce many clusters with few or single members.

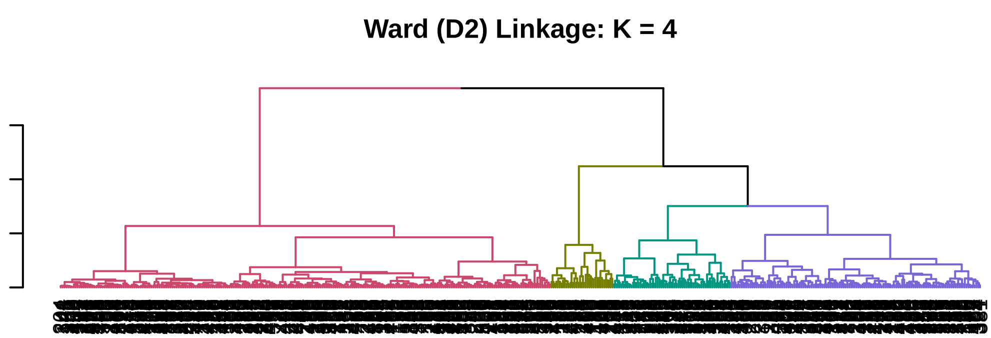
Since the Ward dendrogram seems to be the best among the three, we will look at its distribution for three and four clusters. We chose these initial groupings because this provides a good start to separate and distinguish between player groups.

```
par(mfrow = c(3, 1))
# 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),
     labels = FALSE)
```

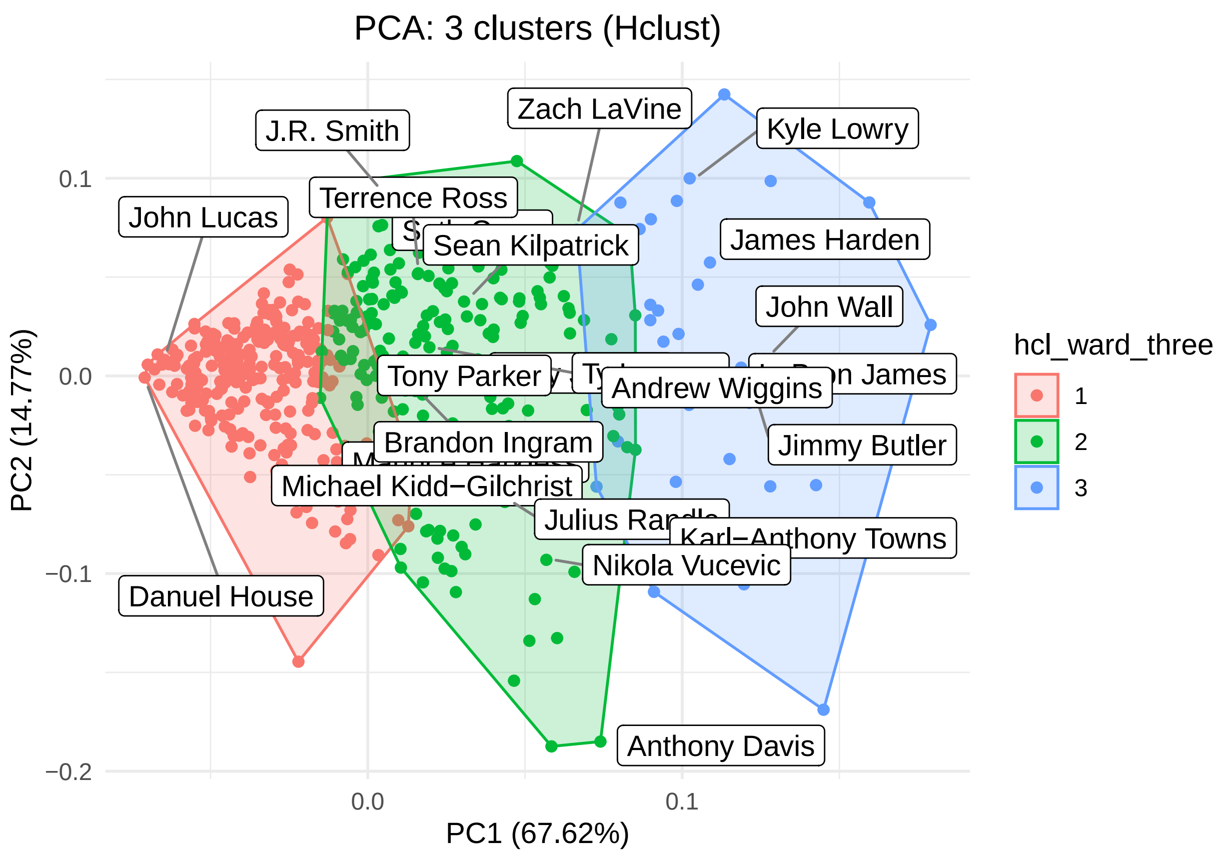


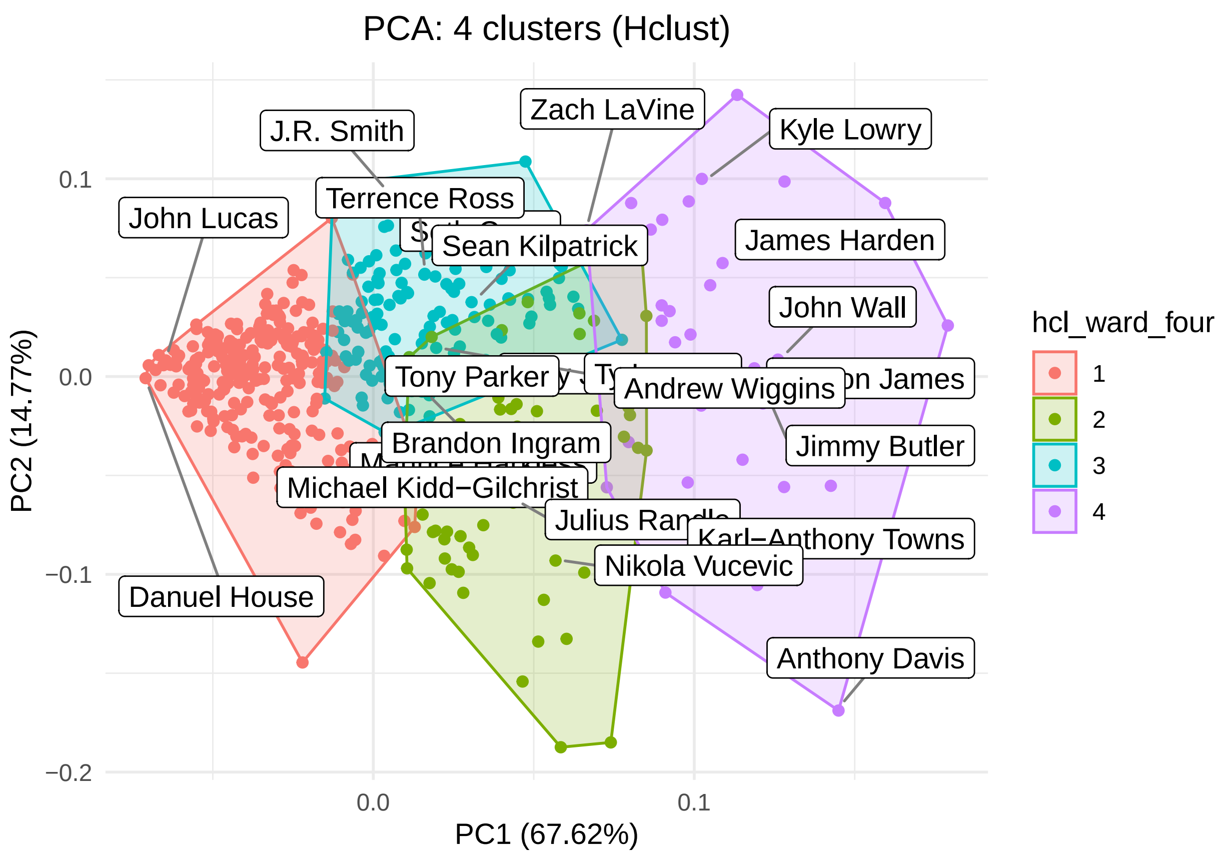




Clusters	k = 3	k = 4
1	259	259
2	194	62
3	33	132
4	NA	33

Visualizing the clusters in PC space, we see clear separation in the four-cluster solution vs. the three-cluster method based on apparent skillsets. For example, LeBron James and James Harden (star players) are in one cluster, whereas John Lucas and Danuel House (lower performers) are in the left-most cluster, opposite to the 'star player' cluster on the right. These will be explored further when we look at player statistics by cluster.





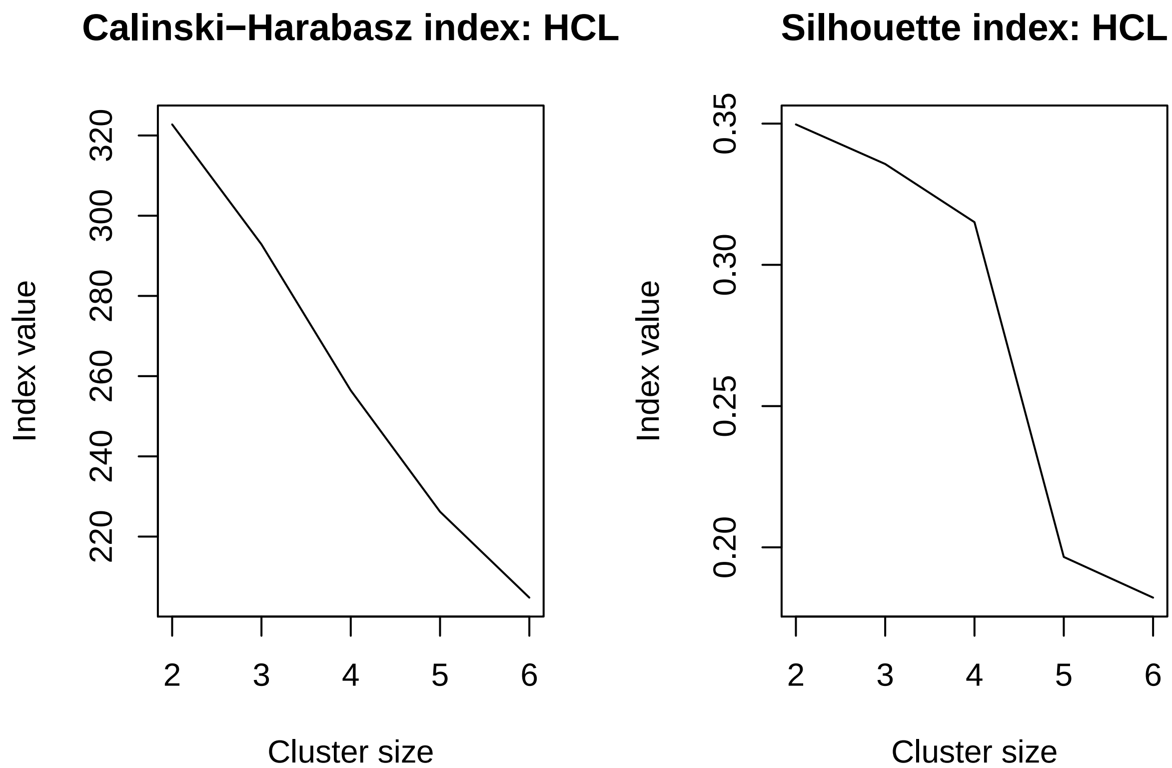
3.3.1.1 Optimize number of clusters

To optimize the number of clusters, we used two index methods: Calinski-Harabasz and Silhouette. These methods measure how well separated the clusters are, and how homogeneous data is within each cluster. Both indices yield 2 as the optimal number.

Although the indices say 2 is optimal, these does not provide enough distinction or separation among players. However, the Silhouette index provides 4 as a local maximum. This provides a more practical distinction among players.

Methods: Calinski-Harabasz index and Silhouette

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

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

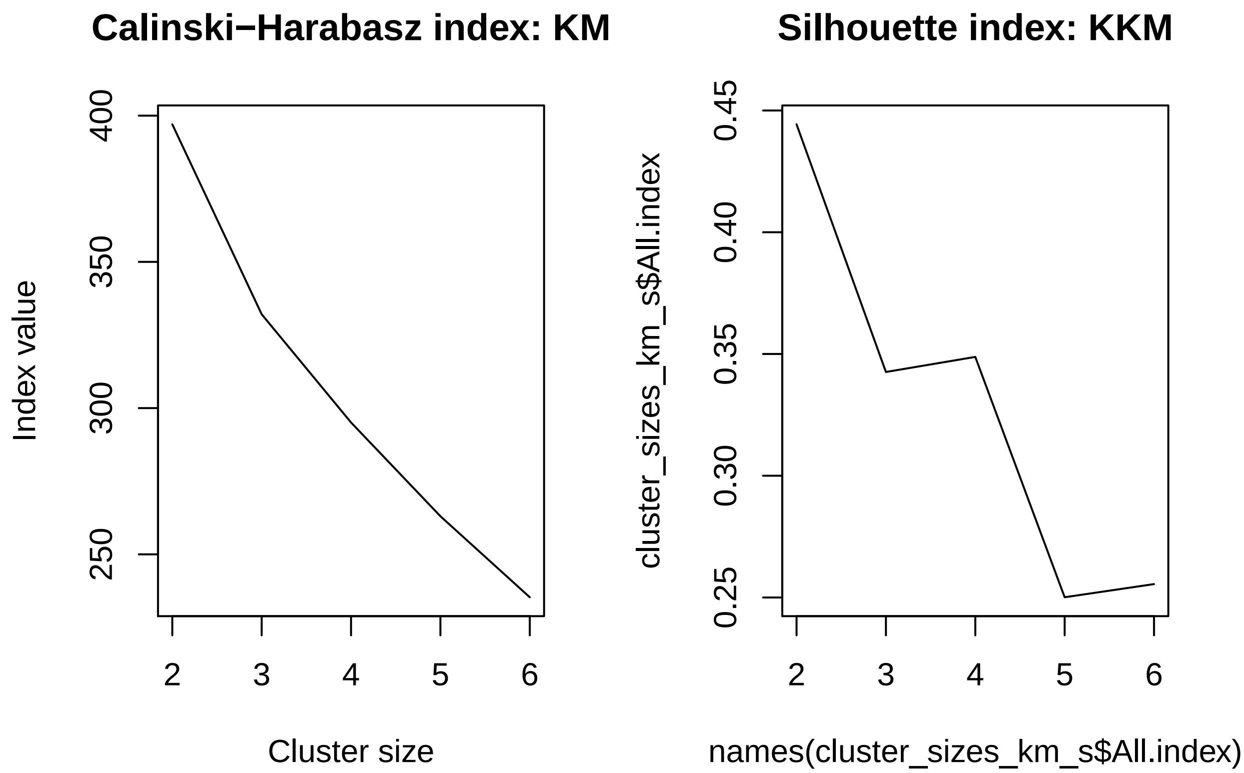


3.3.2 K-Means

3.3.2.1 Optimize number of clusters

Similar to the optimization for hierarchical, we found similar results for K-Means. We decided with the 4-cluster solution based on the Silhouette index (local maximum).

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

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



3.3.2.2 K-means clustering with four groups

Compared to the hierarchical solution presented earlier, there is cleaner separation in the K-Means plot. We will see how these clusters are separated by inspecting features within each group.

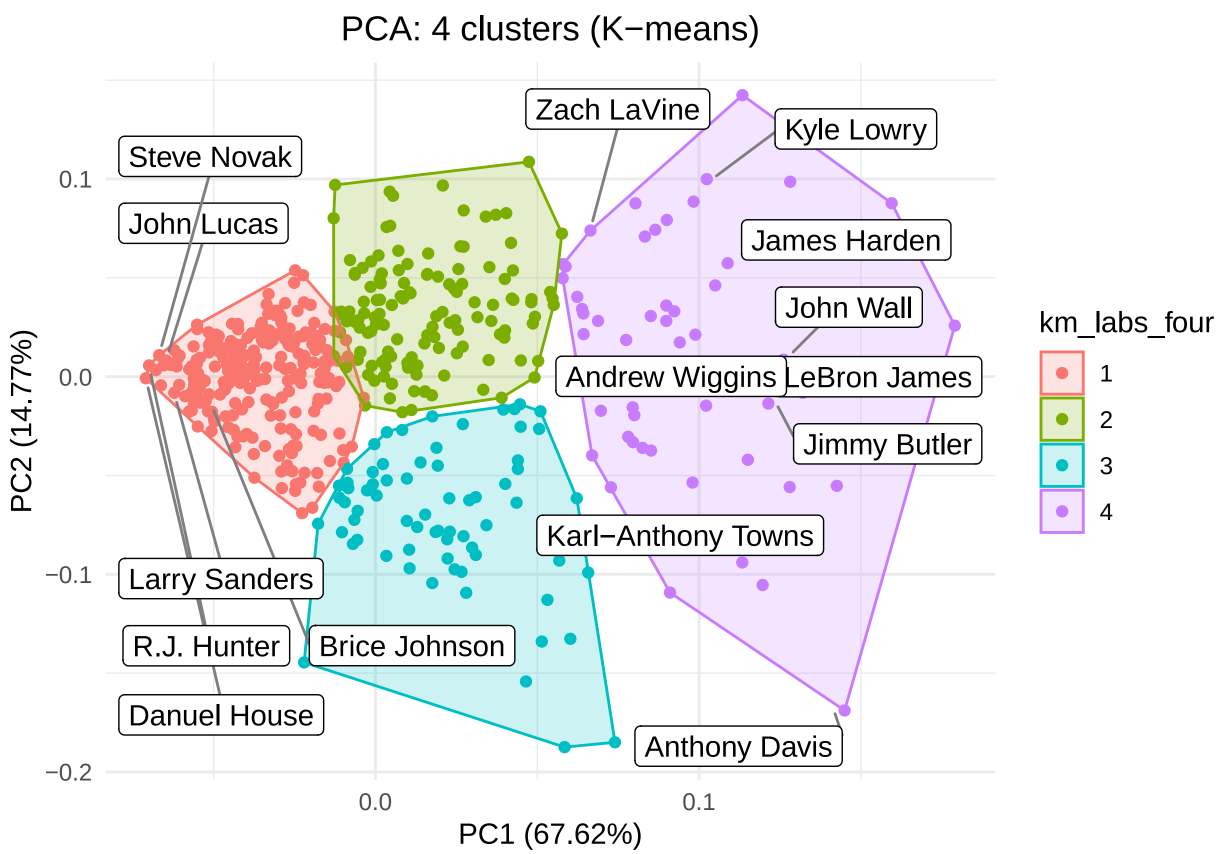
```
km_k <- 4
km_four <- kmeans(x = nba_feat_sc,
            centers = km_k,
            nstart = 100,
            algorithm = 'Hartigan-Wong')

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

Table 3: K-Means Player Distribution

Clusters	Count
1	236
2	130
3	69
4	51

Table 4: Average Stats by Cluster

km_labs_four	MP_pg	PTS_pg	TRB_pg	AST_pg	BLK_pg	STL_pg	VORP	PER	RPM
1	12.25	3.95	2.13	0.88	0.24	0.35	-0.07	10.19	-1.95
2	25.62	10.31	3.48	2.52	0.29	0.82	0.50	12.75	-0.86
3	24.97	10.30	7.01	1.64	0.95	0.79	1.27	17.10	0.37
4	33.86	21.82	5.72	4.73	0.62	1.16	3.19	21.29	2.51

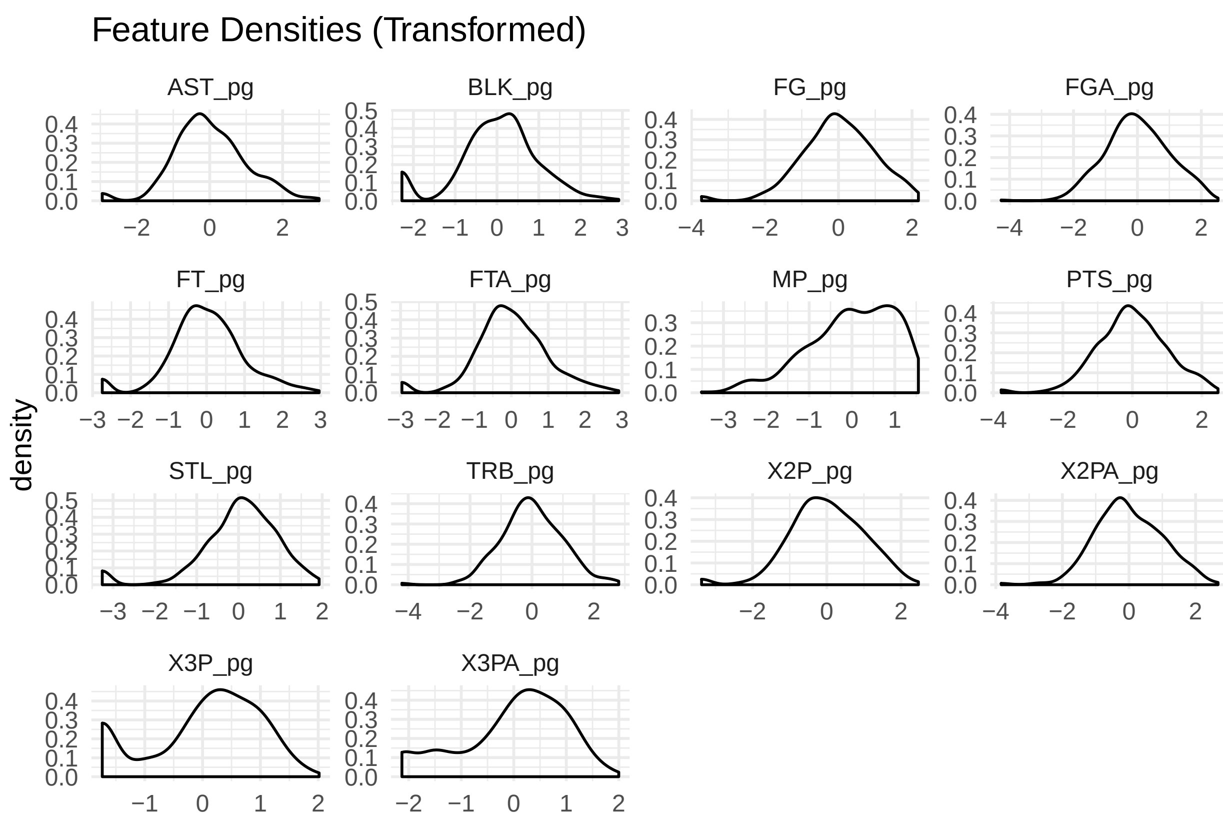


One explanation for the imbalanced number of players across clusters is that player skillset level is inherently imbalanced. This imbalance is reflected in the univariate plots, where most densities were positively skewed. This suggests that there a few players whose statistics significantly exceed those of the average player. This is also reflected in the average statistics by cluster shown below.

3.3.3 Model-based clustering

To test alternative approaches to Hierarchical and K-Means clustering, we will perform model-based clustering. Prior to modeling, we will transform the data using the cube root. This normalizes the data, which is a key assumption in model-based clustering. We tested other transformations such as the square root, Box-Cox, and log + 1. Of these, the cube root yielded the best transformation for model-based clustering. Similar to the scaling performed for the previous clustering methods, the features should be on the same scale.

```
# transform features via cube-root
nba_feat_cr <- (nba_feat) ^ (1/3)

# scale transformed features
nba_feat_cr_sc <- scale(nba_feat_cr)
```

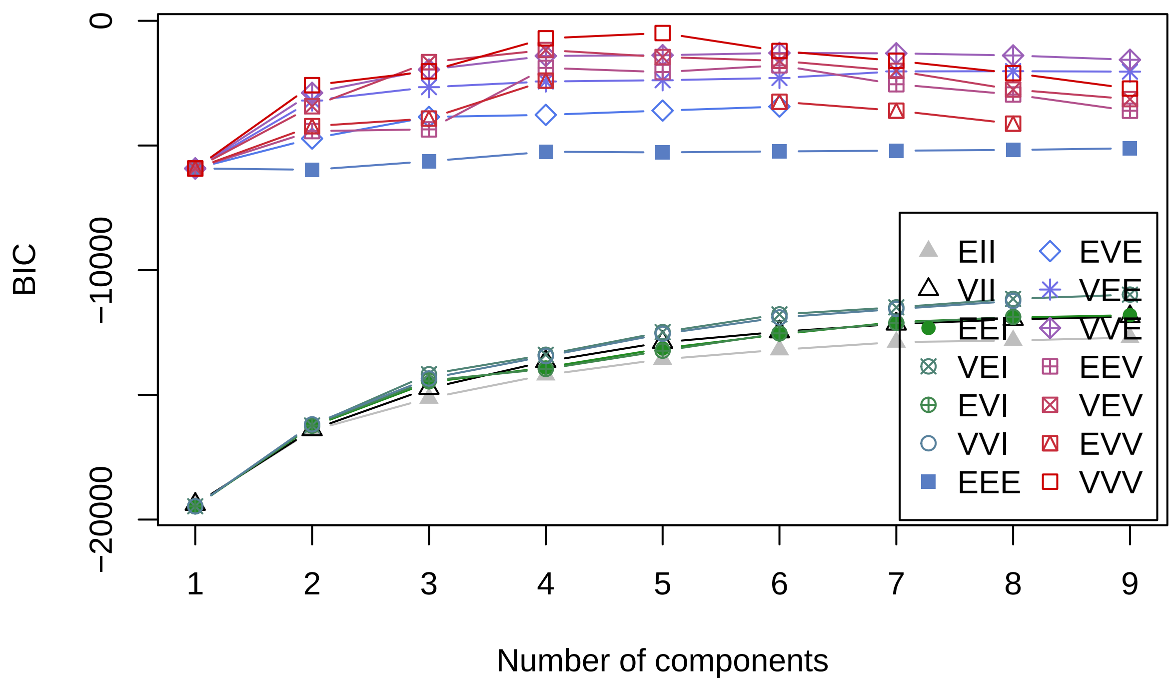


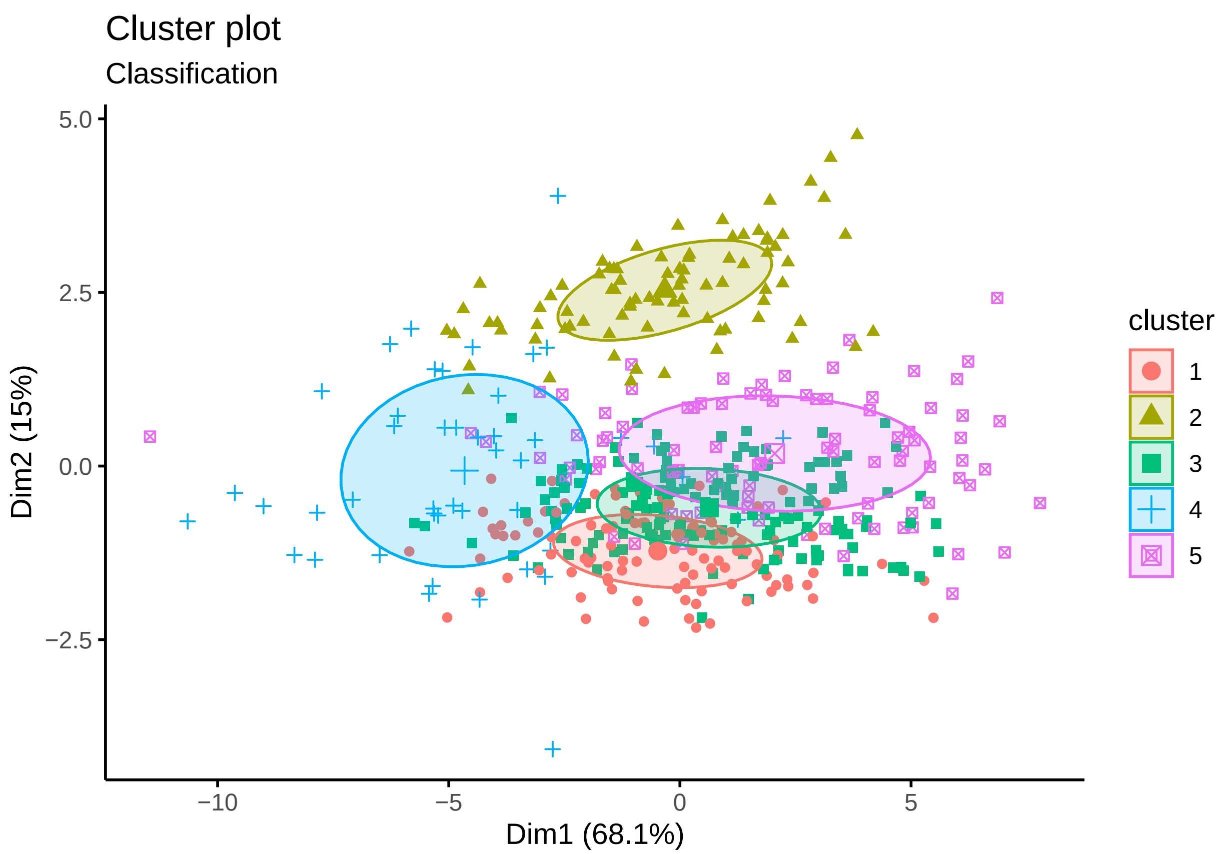
The model-based clustering produced a five-cluster solution based on the BIC. Compared to the previous methods, model-based clustering tended to group players by style of play. On the other hand, K-Means and Hierarchical appear to cluster on overall skillsets across statistics. The clusters also align more with the original PCA plot by position. This type of clustering suggests emphasis on style of play, e.g. better rebounders/blockers, better passers, scorers, etc.

```
# 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
```





3.3.4 Comparing methodologies

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.

```
# 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
```

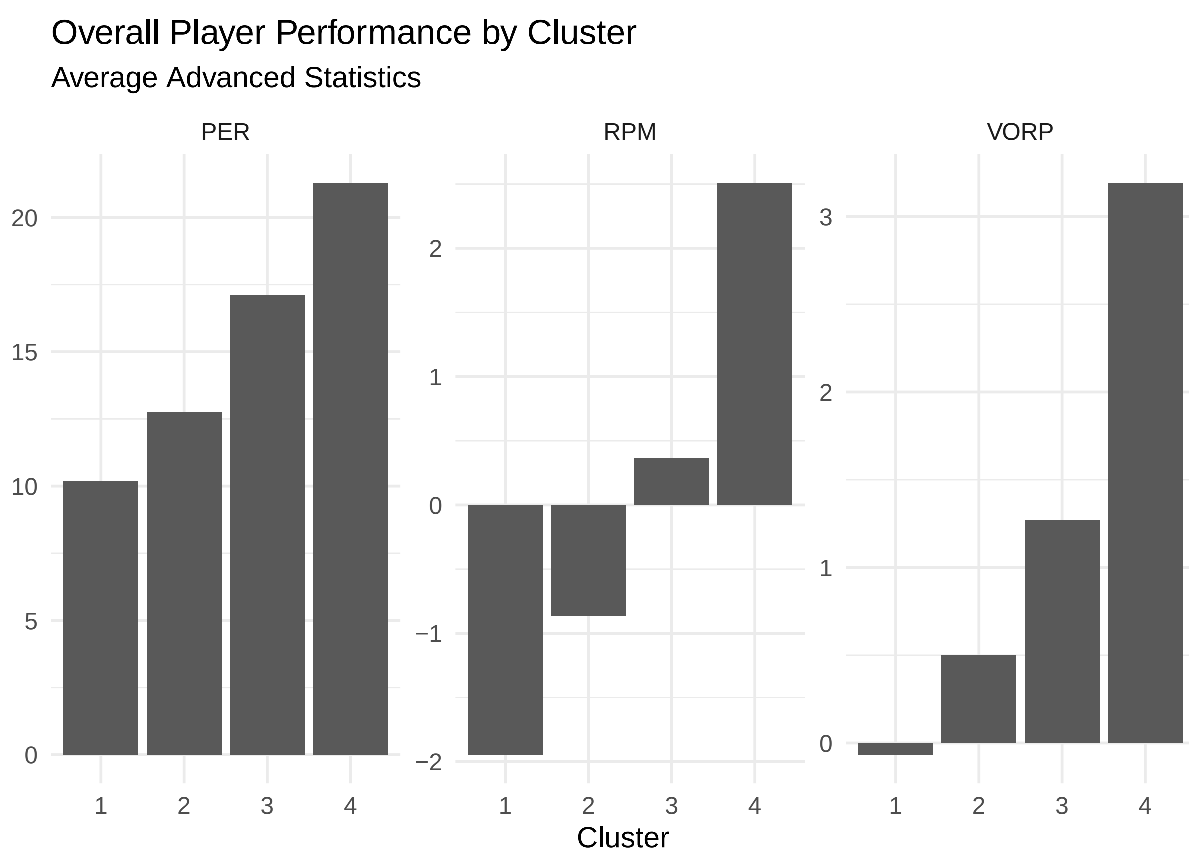
```
xtab_km_mcl
```

```
##                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
```

3.4 Final cluster selection

K-Means (four 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.

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.



4 Conclusion

4.1 Post-cluster analysis

We will now look at different statistics and demographics to assess cluster membership and draw insights.

There is potential to update salaries based on player tiers. For example, Chandler Parsons was paid 22M but is considered a low performer, and is paid more than star players such as Steph Curry (12M) and Kawhi Leonard (17.6M). This shows that salary is not necessarily strongly correlated with player performance.

4.1.1 Clusters vs. player salaries

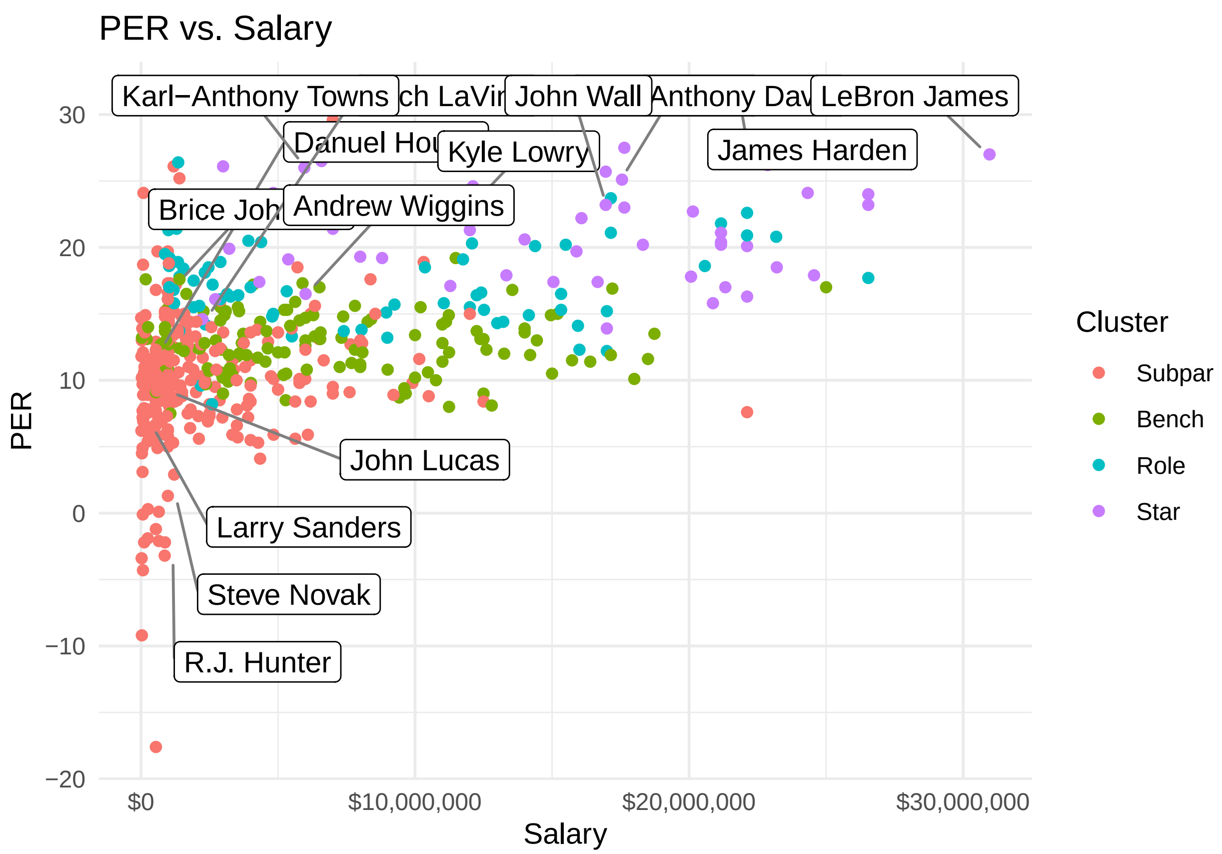


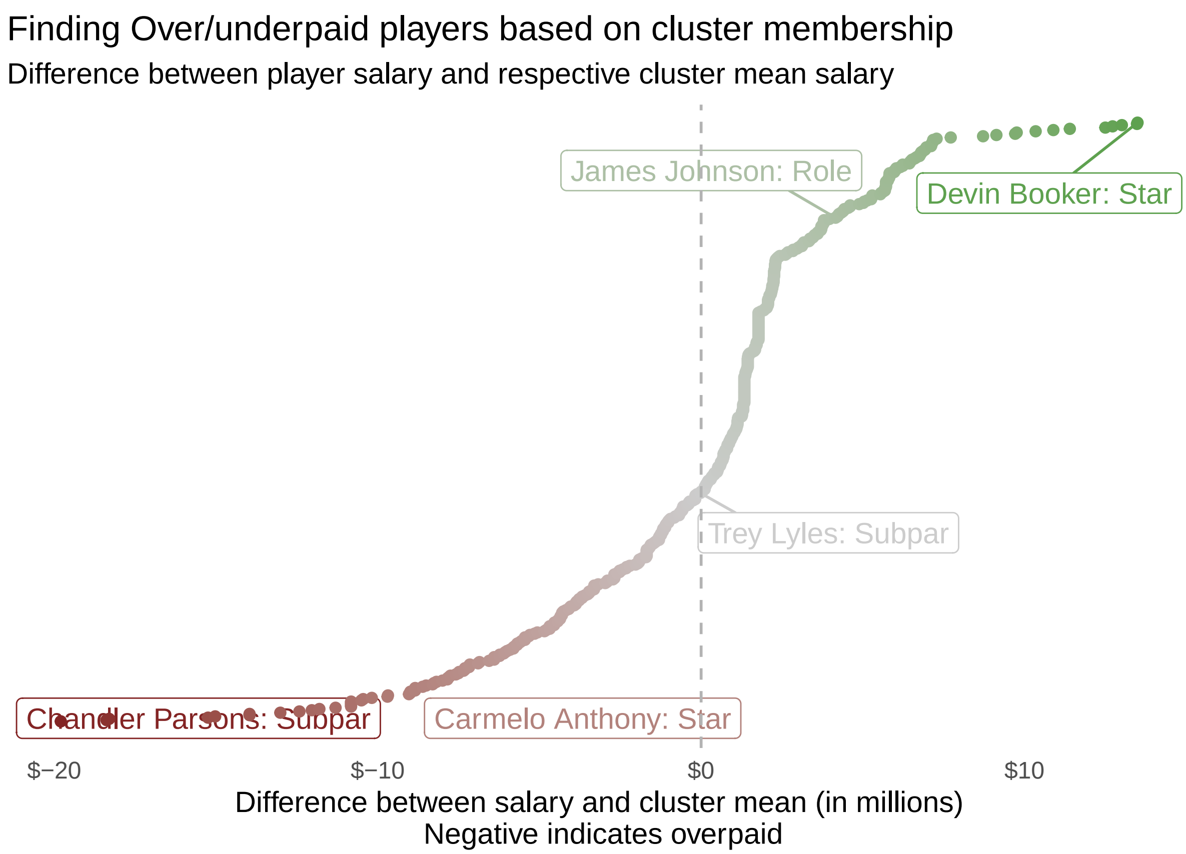
Table 5: Highest Paid Players: Subpar Players

Player	G	MP_pg	Tm	Salary	PER	km_labs_opt_names
Chandler Parsons	34	19.85	MEM	22,116,750	7.6	Subpar
Miles Plumlee	45	10.76	MIL	12,500,000	8.4	Subpar
Amir Johnson	80	20.10	BOS	12,000,000	15.0	Subpar
Mirza Teletovic	70	16.19	MIL	10,500,000	8.8	Subpar
Al Jefferson	66	14.11	IND	10,314,532	18.9	Subpar
Alec Burks	42	15.55	UTA	10,154,495	11.6	Subpar

Table 6: Lowest Paid Players: Star Players

Player	G	MP_pg	Tm	Salary	PER	km_labs_opt_names
Devin Booker	78	35.00	PHO	2,223,600	14.6	Star
Zach LaVine	47	37.21	MIN	2,240,880	14.6	Star
Dennis Schroder	79	31.46	ATL	2,708,582	16.1	Star
Giannis Antetokounmpo	80	35.56	MIL	2,995,421	26.1	Star
C.J. McCollum	80	34.95	POR	3,219,579	19.9	Star
Kristaps Porzingis	66	32.79	NYK	4,317,720	17.4	Star

The chart below indicates how much a player is overpaid or underpaid with respect to their cluster average salary. For example, Chandler Parsons was paid 22M vs. the cluster salary average (~2M), indicating he was overpaid. It is important to note that there are players who may have been injured, so their statistics may not be commensurate to their salaries.



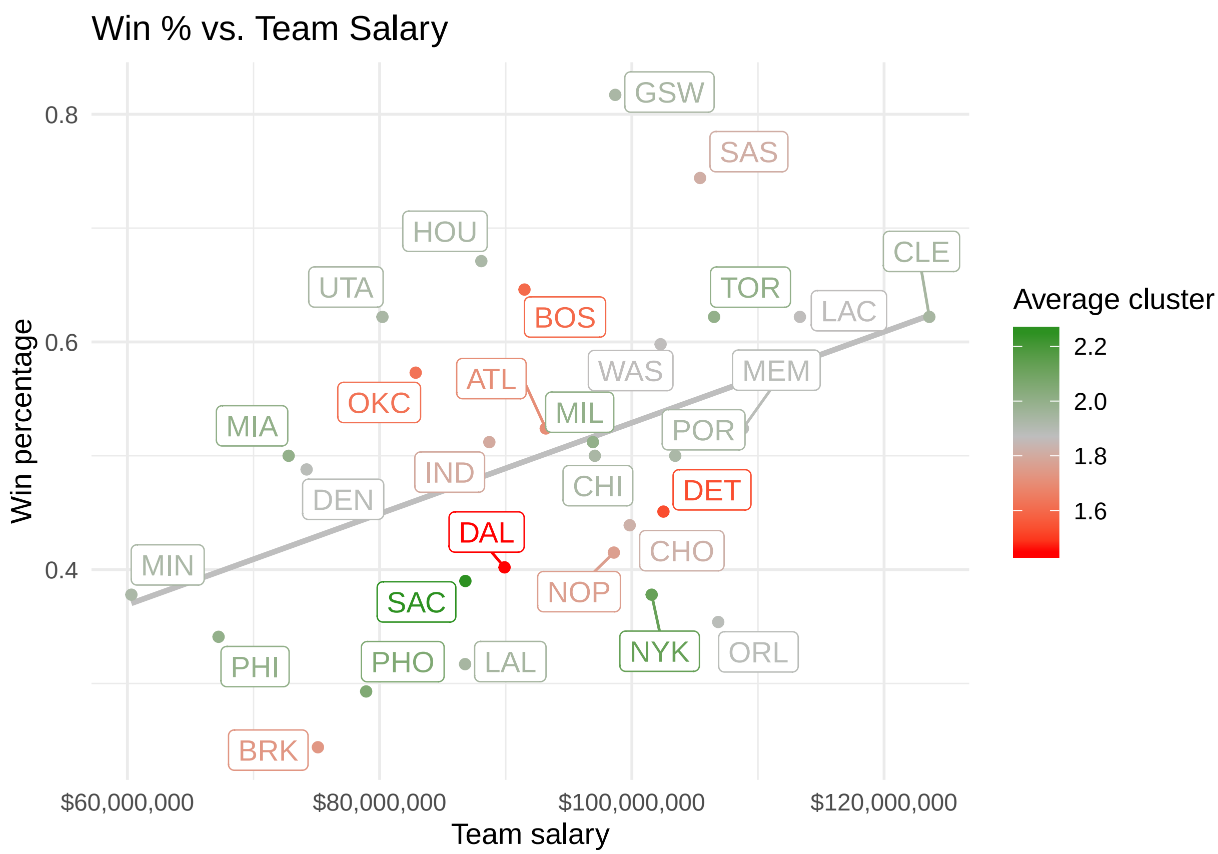
4.1.2 Team compensation and performance vs. clusters

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. Paying more does not necessarily translate into wining more games. For example, the Utah Jazz (UTA) have the same winning percentage as the Los Angeles Clippers (LAC) despite a substantial difference in player payroll spending. This could be driven factors outside of the dataset such as good 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 %.

Tm	team_salary	win_pct	avg_clust
SAC	86,799,609	0.39	2.25
NYK	101,570,502	0.38	2.12
PHO	78,930,157	0.29	2.06
MIA	72,782,449	0.50	2.00
MIL	96,913,241	0.51	2.00
PHI	67,225,712	0.34	2.00
TOR	106,521,470	0.62	2.00
CLE	123,591,014	0.62	1.94
LAL	86,775,415	0.32	1.94
CHI	97,064,073	0.50	1.93
GSW	98,681,493	0.82	1.93
UTA	80,223,193	0.62	1.93
HOU	88,062,247	0.67	1.93
MIN	60,311,572	0.38	1.93
POR	103,439,444	0.50	1.93
DEN	74,208,517	0.49	1.88
MEM	108,808,118	0.52	1.88
ORL	106,849,160	0.35	1.88
LAC	113,327,068	0.62	1.87
WAS	102,276,673	0.60	1.87
CHO	99,830,531	0.44	1.82
SAS	105,410,231	0.74	1.81
IND	88,698,690	0.51	1.80
NOP	98,573,436	0.42	1.76
BRK	75,102,568	0.24	1.74
ATL	93,172,774	0.52	1.71
OKC	82,858,524	0.57	1.62
BOS	91,484,921	0.65	1.60
DET	102,503,259	0.45	1.53
DAL	89,904,500	0.40	1.45

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 %.



4.2 Final thoughts

The K-means four cluster solution was able to accurately group players by general skill set and productivity. Tables 7-9 show cluster assignment of players on 3 selected teams and provide an illustrated example of the accuracy of the clustering . Across all three of the selected teams, cluster assignment accurately reflects the contributions of each player. This is also reflected empirically when statistics such as VORP RPM and PER are explored at the cluster level (refer to tables 5 & 6).

Our analysis provides interesting insight into the similarities and differences between NBA players in a given season. In general, teams look to have a diverse roster of players with varying skillsets and styles of play. As shown in the PER v. Salary plot, some players are high producers and are effective scorers, but are not highly compensated. This posits that some team managers are prudent at selecting good talent early, when they are under low salary contracts, and developing their players into future stars.

On the other hand, in the Win % vs. Salary plot, it was clear that high team spending on higher-caliber players does not necessarily translate into higher winning percentages. For example, a team like Boston consisted of mostly average players and spent a relatively lower amount of money than most teams. However, their winning percentage was approximately 65%.

Our unsupervised clustering methodology was effective in grouping and assessing NBA players based on similar characteristics, skill sets, and play styles. This information is useful for team managers and coaches to build cost-effective, winning franchises.

Table 7: NYK: Players

Player	PER	km_labs_opt_names
Carmelo Anthony	17.9	Star
Kristaps Porzingis	17.4	Star
Derrick Rose	17.0	Star
Willy Hernangomez	18.9	Role
Joakim Noah	15.2	Role
Kyle O'Quinn	20.5	Role
Justin Holiday	12.7	Bench
Brandon Jennings	12.1	Bench
Courtney Lee	12.1	Bench
Ron Baker	7.5	Subpar
Mindaugas Kuzminskas	12.4	Subpar
Maurice Ndour	11.3	Subpar
Marshall Plumlee	10.9	Subpar
Chasson Randle	13.6	Subpar
Lance Thomas	8.4	Subpar
Sasha Vujacic	8.6	Subpar

Table 8: NYK: Players

Player	PER	km_labs_opt_names
Stephen Curry	24.6	Star
Kevin Durant	27.6	Star
Klay Thompson	17.4	Star
Draymond Green	16.5	Role
Zaza Pachulia	16.1	Role
Andre Iguodala	14.4	Bench
Ian Clark	13.1	Subpar
Damian Jones	5.3	Subpar
Shaun Livingston	10.1	Subpar
Kevon Looney	13.4	Subpar
James Michael	13.0	Subpar
Patrick McCaw	8.6	Subpar
JaVale McGee	25.2	Subpar
Anderson Varejao	9.4	Subpar
David West	16.6	Subpar

Table 9: NYK: Players

Player	PER	km_labs_opt_names
Isaiah Thomas	26.5	Star
Al Horford	17.7	Role
Avery Bradley	14.4	Bench
Jae Crowder	14.9	Bench
Kelly Olynyk	15.2	Bench
Marcus Smart	12.0	Bench
Jaylen Brown	10.3	Subpar
Gerald Green	12.0	Subpar
Demetrius Jackson	30.8	Subpar
Jonas Jerebko	9.3	Subpar
Amir Johnson	15.0	Subpar
Jordan Mickey	9.8	Subpar
Terry Rozier	10.8	Subpar
James Young	10.0	Subpar
Tyler Zeller	13.0	Subpar