Redefining Position Roles in the NBA

Final Project | BUSN 41201 - Big Data

The University of Chicago Booth School of Business

Josh Eckmann | March 13, 2025

Abstract

In team sports - the compatability of individual play style between teammates often recieves high criticism in the development of rosters, and throughout the process of stated or organic role definition. Historically, the National Basketball Association (NBA) has been home to relatively homogenous teams with roles defined primarily by player position - but significant rule changes, unique star players, and the globalization of the game have made the position-defined role obsolete. This work uses box score information from every NBA game since 2010 to redefine the "positions" in a way that reflects the attributes and realistic role of the individual player. Through principal component analysis and clustering, this work leverages machine learning tools to identify and define roles in the new NBA - the findings (particularily when mapped to historically successful teams) create great value for front offices as they look to compile rosters that work well together and, hopefully, win games.

Contents

Part 0 | Introduction

The Changing NBA Landscape

Anyone familiar with the NBA would tell you that it has changed drastically over the last 30 years. With 7-footers regularly bringing the ball up the court and player still defined as Centers shooting threes, the game looks nothing like it did some 30 years ago. Some attribute these changes to the players, and some attribute them to the introduction and acceptance of analytics in basketball - but one thing is true, the way the game is played has changed and it does not appear to be going back anytime soon.

Whether a result of new shot-first players, like Steph Curry and James Harden, or the analytics based advantage of increased three point shooting - one of the biggest statistically identifiable shift in game play has been the rise in popularity of the three pointer and the related bump in game scores. The plot below shows the average number of three pointers attempted in each NBA game in each season since the three point line was introduced in 1979.

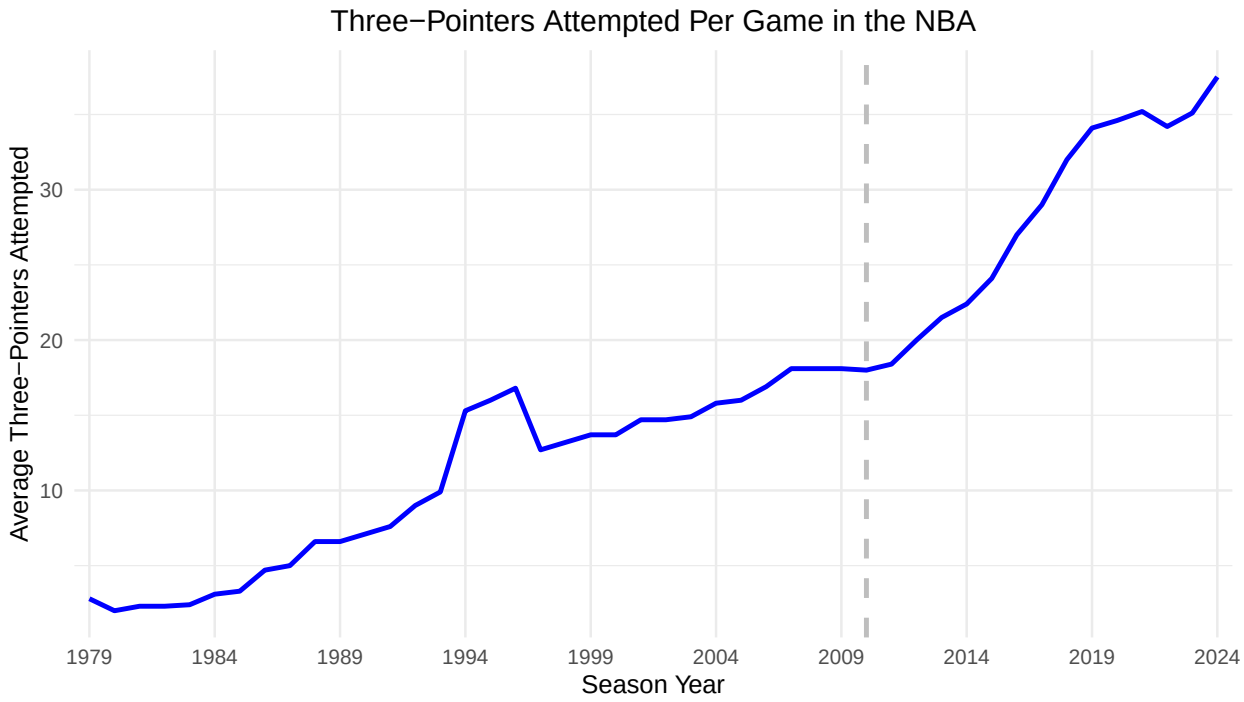

Figure 1: Three pointers attempted per team per game in each NBA season since 1979

Using three pointers as a general proxy for the style of the game, we see a relatively sharp shift towards score-centric game play. As the plot shows, the 2010 season appears to be a notable cutoff between the old and new NBA - this notion is backed up by the general basketball community consensus on the different eras of the NBA.

The Problem

The Traditional Basketball Positions

Basketball players were historically classified by on-court position - with actions following the prescribed position. Let's take a quick look at the five positions:

- **Point Guard (PG):** Primary ball handler, responsible for directing the offense and distributing the ball.
- **Shooting Guard (SG):** Typically a strong scorer, expected to shoot the three well and handle the ball some.
- **Small Forward (SF):** Versatile player - typically playing a key offensive and defensive role with great athleticism.
- **Power Forward (PF):** Physical player with good defensive ability and strong scoring at close range.
- **Center (C):** Usually the tallest player, focused on rebounding and defending around the rim.

To look at how these roles translate into play style - we first look at scoring and shot-taking by each group over time. The plot below depicts the average point and three point attempts by each group - scaled to be comparable as the amount per 48 minutes of playing time (the length of an NBA game).

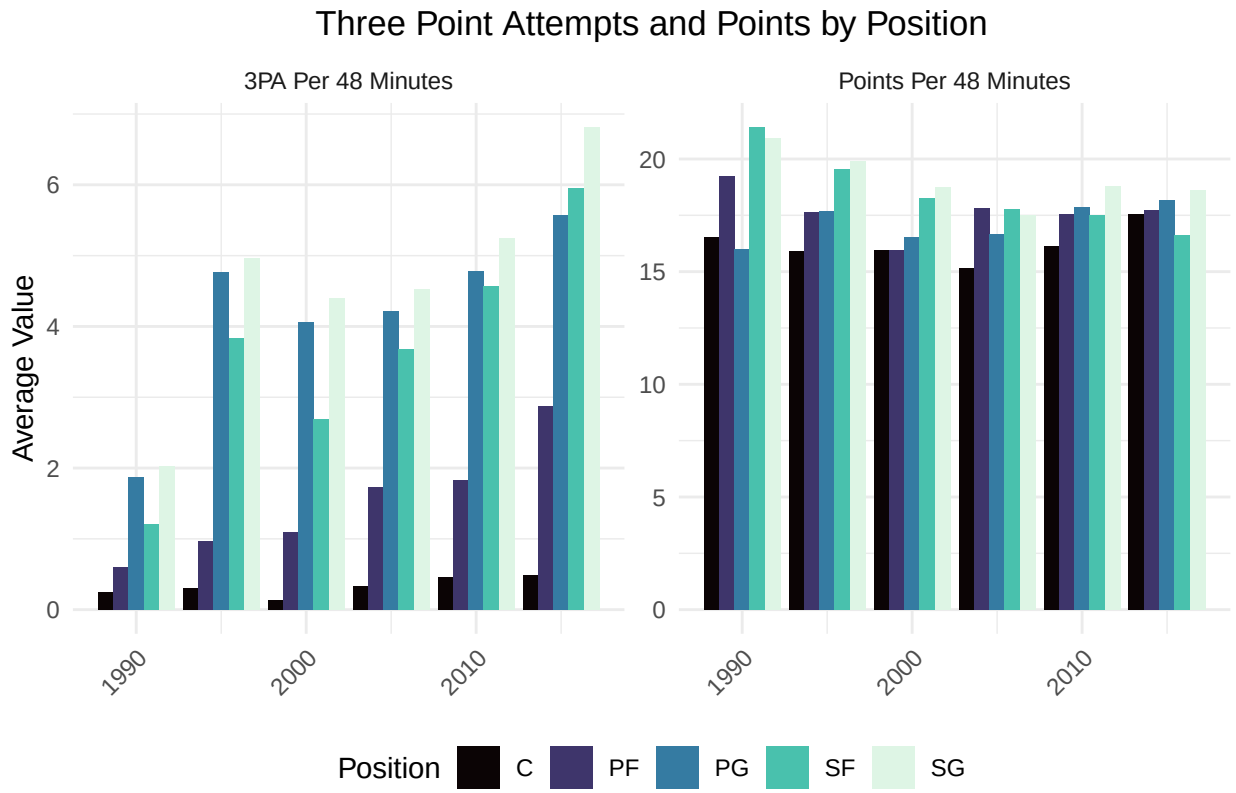

Figure 2: The obsolete position label – shooting and scoring activity by position group over time

The plots above perfectly illustrate the shift in position-defined roles in the NBA. We see that in 1990, there is a great discrepancy in the scoring load with Small Forwards and Shooting Guards scoring far more than Point Guards and Centers - and the prototypical perimeter players (PGs and SGs) shooting significantly more threes. Compare that to 2015 and not only do we see a universal increase in the number of threes per

48, but we see a near perfect uniformity in the distribution of scoring burden. If the assigned positions were still reflective of in-game roles we would not see this shift.

The problem then - is to redefine the NBA players based on the way they contribute to the game. Using individual statistics we seek to define new position labels that better reflect what each player brings to the table.

Proposed Solution

To add value to the field our goal is to define and then clearly label new role-based position groups. The first step involves applying Principal Component Analysis (PCA) to reduce the dimensionality of the player statistics, highlighting the most important features that contribute to player identity. Next, we use clustering techniques to group players based on these key features, uncovering play style based clusters that differ from traditional position designations. Finally, we will employ a Random Forest model to refine these groupings and identify the factors that most significantly influence player positioning. This comprehensive approach will allow us to create more nuanced and data-driven position categories, potentially leading to new insights into player roles and team strategies - allowing all parties to predict position assignment of current players as I do or of potential players/matchups.

Part 1 | Principal Component Analysis (PCA)

Why Use PCA

In the context of this work, it is crucial to use PCA for dimension reduction. The data we use contains the yearly stats for every NBA player from 2010-2024 - with many dimensions. Our data set contains 35 statistical variables for each player, and many of them may be correlated - meaning dimension reduction will be useful and helpful as a prerequisite for clustering.

Preparing and Plotting PC1 and PC2

After scaling the numeric data in our data set, I remove rows with no values (indicating seasons where a player did not play did not play - likely due to injury as they would not be in the data set unless the previous and next season included recorded statistics) and rows where players played in less than 5 games (indicating a player may have faced injuries, was a very deep bench player, or was on a two-way contract and only player due to extreme circumstances). Also, because we are primarily worried about identifying characteristics that define on court play and are using both raw and rate statistics - we do not want to include players who play very little. This would likely lead to a significant cluster of bench players. This is valuable but can easily be observed by looking at playing time statistics with no insight into the contribution of a given player. To do this, we remove player seasons from the data where players saw the court less than 10 minutes per game. This cutoff, while somewhat random, is informed by the fact that the bottom quartile of the players play less than 13 minutes per game - we leave some small role players but avoid confusing the analysis with information skewed by lack of playing time.

With the data cleaned up a bit more I used prcomp to estimate principal components for the data using 31 dimensions (each a different statistic for the players season). The plot below shows where a random sample of 20 player seasons fall on the first two principal components - allowing for some initial analysis of what the PCs are rotating on.

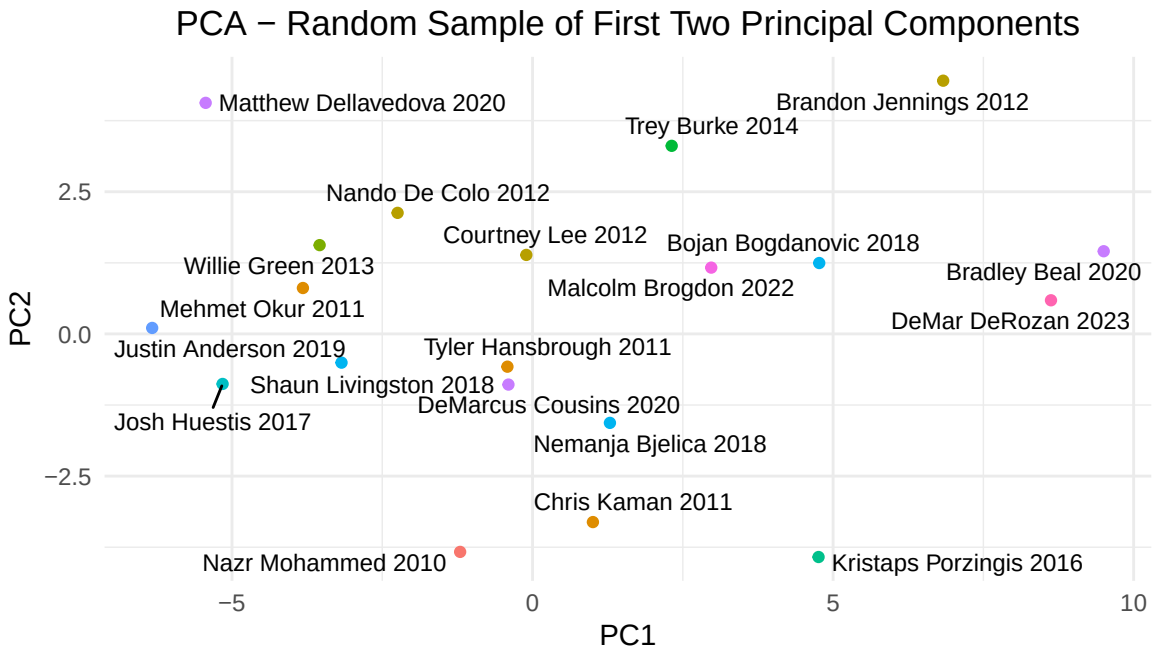

Figure 3: Sample of player seasons plotted on principal components 1 and 2

This plot shows some early signs of clear distinction within our player seasons based on the principal component rotations. Players that took on larger roles appear to be more positive on PC1 - including all-stars like DeMar DeRozan in 2023. Principal component 2 appears to capture more on court distinctions with long defensive and rebounding players like Kristaps Porzingis scoring very negatively and more ball-handling shooters like Brandon Jennings showing up as very positive.

We will look closer into the effective explanation of variance of our PCA as well as the direction and magnitude of rotation on variables of the first few PCs below.

SCREE Plot

The SCREE plot below shows the variance explained by each of the rotations/ principal components, which we will use (in part) to determine how many of these PCs to use in our analysis moving forward. It is hard to tell by simply eye-balling the "elbow" of the plot but it appears that we will end up using 6 or 7 PCs.

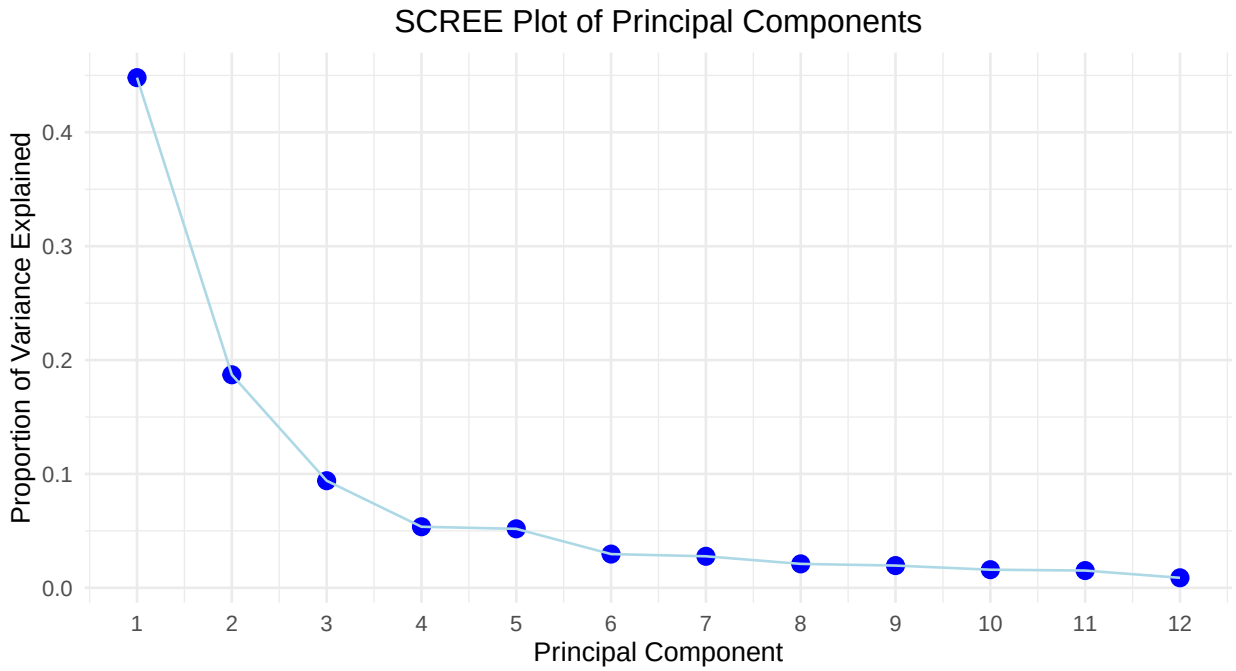

Figure 4: SCREE plot of explained variance for the first 12 PCs

The SCREE plot above shows how well the first few PCs do at explaining variance, with almost 75% of the variance being explained by PC1, PC2, and PC3. Knowing that the first PCs are very important, we wanted to look at the key rotation loadings with the greatest magnitude for each of the first principal components to understand what significant basketball factors may be captured by them.

Top 5 Rotations by Magnitude for Principal Component 1-4

The tables below show the five loadings with the greatest magnitude (positive or negative) to give us a look at what factors may be leading the variance that PCA is accounting for.

Principal Component 1

Variable	Rotation_Value
PTS	0.2556929
USG	0.2540252
FGM	0.2526473
FGA	0.2502313
PTS_PG	0.2452987

Principal Component 1, where points and Usage (USG - an advanced statistic estimated the number of a teams offensive possessions a player uses while on the floor) have large rotation values, may be identifying players that are more star-like; dominating the shot attempts while on the court and scoring at a higher than average level.

Principal Component 2

Variable	Rotation_Value
REB_PM	-0.3421792
BLK_PM	-0.3349846
BLK_PG	-0.3270140
BLK	-0.3062370
FG_PCT	-0.2917259

Principal Component 2 looks like it focuses on defensive and rebounding oriented players - possibly big men who spend a great deal of time around the rim.

Principal Component 3

Variable	Rotation_Value
ORtg	0.4199327
TOV_PM	-0.3169074
TS	0.3163672
3PM	0.2693990
3P_PCT	0.2489355

Principal Component 3, based on the loadings, looks like it identifies efficient players. Offensive rating (ORtg) and True Shooting Percentage (TS), which both have high positive rotations, are both advanced efficiency statistics - meaning players high on PC3 turn the ball over infrequently and make good use of their time on the court to score and help the team.

Principal Component 4

Variable	Rotation_Value
STL	-0.3892142
STL_PM	-0.3725635
TOV_PM	0.3580780
STL_PG	-0.3310324
PTS_PM	0.2913939

Principal Component 4 seems to identify efficient defense-first scorers (think Jrue Holiday) - where those who have a big negative value get lots of steals and score well on a per minute rate.

Using LASSO to Select PCs

Although we could use the elbow method - selecting the number of PCs based on the amount of variance each one explains, we choose to leverage generalized linear models (glm) and LASSO regression to aid in the selection of relevant principal components for clustering.

In order to run glm and LASSO regression for PCA selection though, we must select a target variable. For this we have chosen the advanced statistic Usage Rate (defined above) - as we believe it is the single statistic most representative of on-court roles in the NBA. After defining the target variable and running a 10 fold cross validated LASSO regression of the predicted values from our PCA onto Usage Rate. We get the LASSO Path plot below and the respective AICc minimizing set of PCs to use.

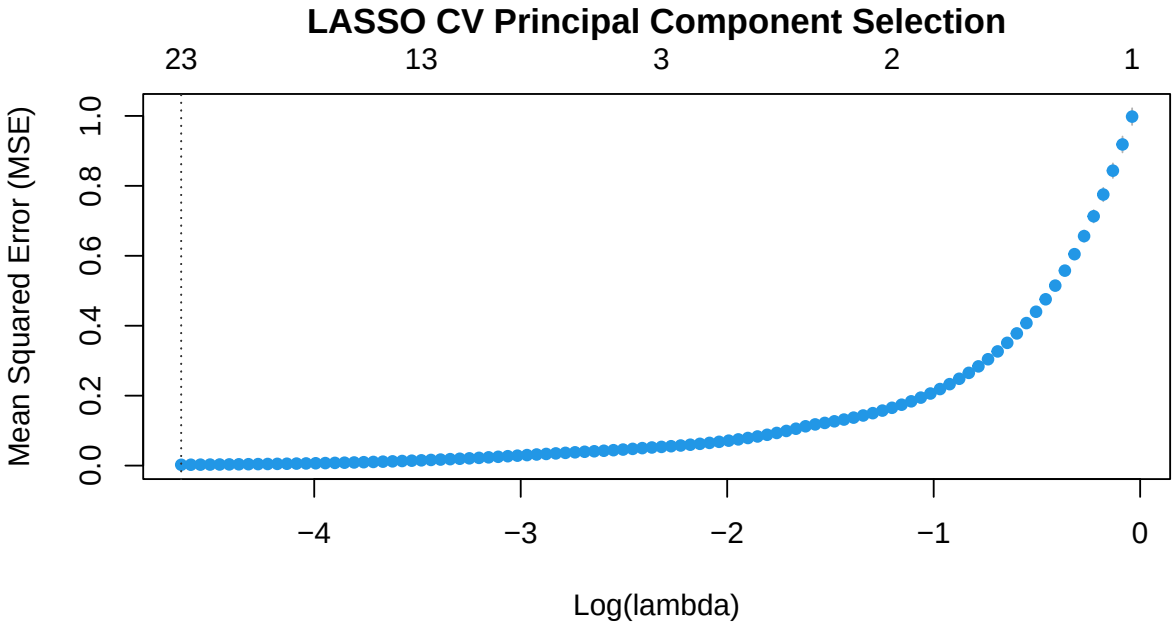

Figure 5: Cross validated LASSO principal component selection

The cross-validated LASSO model, as shown above, selected 23 PCs (but we note that this LASSO may be bias as the data used for the PCs included our target USG) as the best for averaging out of sample deviance

when predicting Usage Rate (if we decrease the lambda minimum ratio it selects more PCs until we run out of variables). Because usage rate is just a proxy for our consideration of how large a role a player has and does not necessarily capture the elements of the new positions we want to define - we take these results as somewhat tangential to the primary focus of this work. Performing the LASSO does validate to us that no advanced statistic like usage can be used to capture the entirety of depth to player roles - and because Usage is contained within the data used for PCA, we perform another PCA as an aside where USG is not included (note at this point that usage can generally be calculated using other dimensions in the system but we will remove anyway and see if we get a different result or if LASSO wants us to again use as many PCs as we can.

LASSO for PCs (Accounting for Complexity of USG)

As mentioned above, USG can be estimated (but not perfectly calculated) within system - so even when we remove it there are many close ties between **every** principal component and USG. To address this we use a higher LASSO penalty to identify the most important principal components.

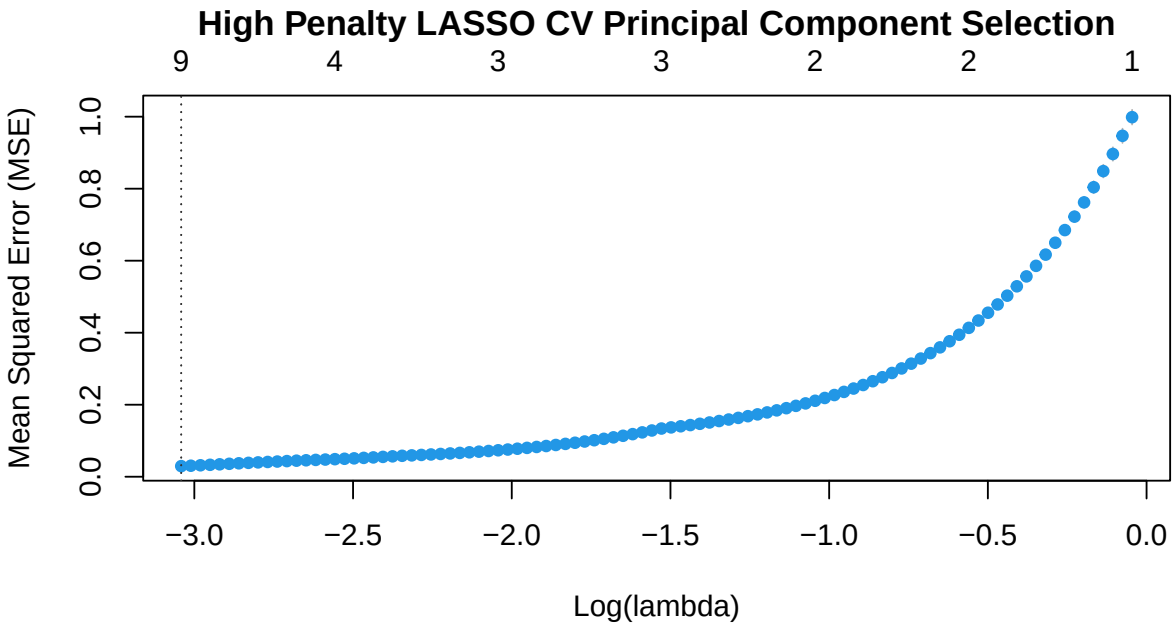

Figure 6: Withough USG cross−validated LASSO principal component selection

The plot above, which is mapping slightly different PCs than those in our primary problem, shows the selection of 9 with a moderately high lambda penalty - which is still valuable for us as we consider the number of PCs it takes to capture variance generally in the data.

We will be using the "explanation of variance" criteria for selecting principal components for clustering in the next section. For this we select the first 7 principal components.

Part 2 | Clustering

Why use Clustering?

Now that we have completed PCA - we can use the findings to reduce our dimensionality, making it easier to measure distance and see which players fall close to each other on the dimensions that we have determined explain a great deal of variance between on-court contributions. We are able to capture groups on 7 dimensions instead of 34, very helpful in the interpretation. The goal of this project is to define new NBA archetypes based on role and play style, and clustering will do exactly that by using the principal components that capture the major differences to group players into their new "position" groups.

Clustering - in short, is the way we move this project from analysis to actionable insight.

K-Means Clustering using 7 Principal Components

Now - the problem becomes the selection of K, the number of clusters (AKA positions) we want to use. While the conventional expectation would be to automatically use 5 - because there are 5 players for each team on the court at a time, but remember we are attempting to capture more holistically what a player brings to the team. We may be willing to use more clusters because we believe there are more role-centered player types in the NBA than just 5 on-court "positions".

First, we use k-means and visualize the within cluster sum of squares to see where additional clusters do not decrease error - indicating potentially "role-capturing" groups of player seasons.

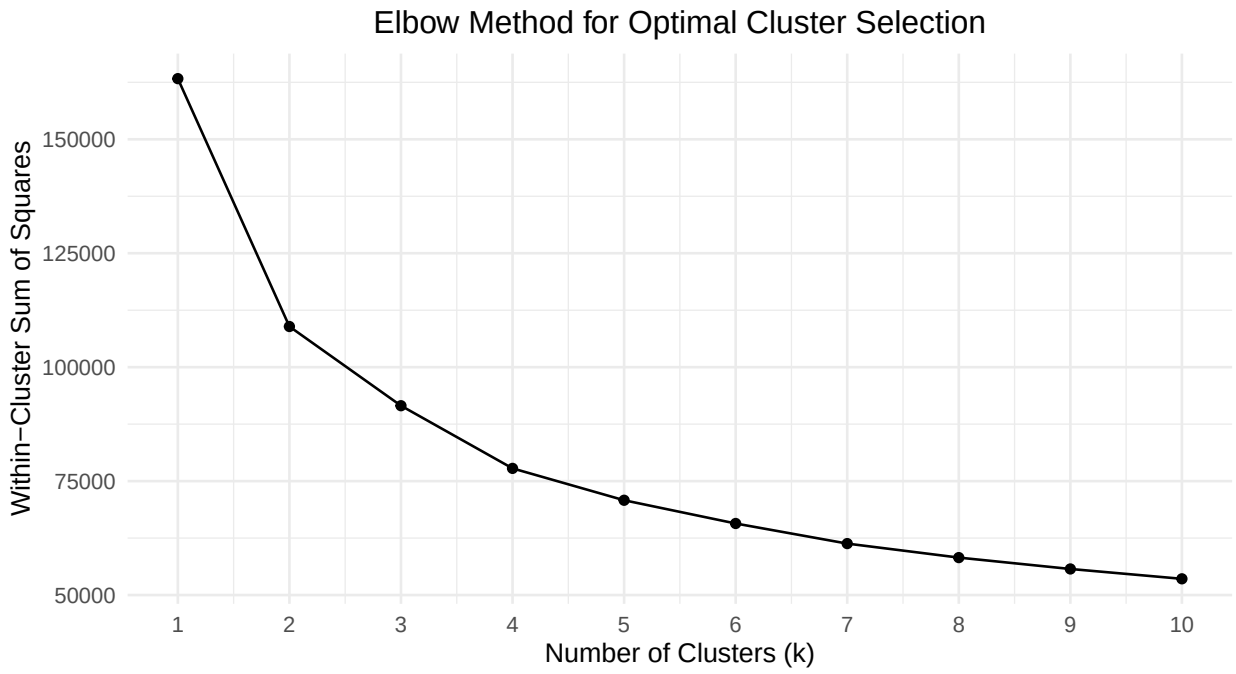

Figure 7: K−Means Clustering and WSS for K Selection

Hierarchical Clustering

As we use principal components to perform our clustering - we believe it is valuable to consider a hierarchical clustering method as certain PCs may be more determining in the assignment of groups than others (this

idea is pursued with the knowledge that each principal component accounted for a different level of variance degradation in the original PCA)

We perform agglomerative (bottom-up) clustering and the tree plot below shows the clusters formed by this method.

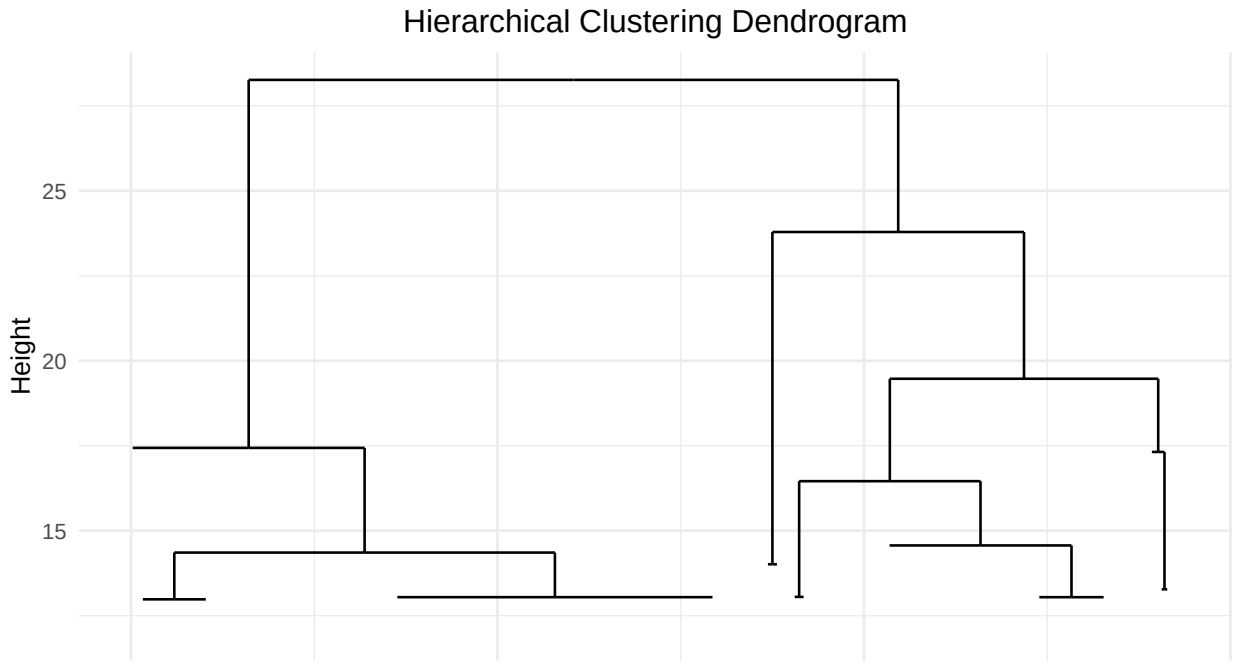

Figure 8: Dendrogram for agglomerative clustering − cutting only the top clusters

Assigning Cluster Labels

We have clustered using both K-Means and Agglomerative Clustering - and based on the resulting within-cluster sum of squares we have determined that 7 clusters best captures the differences in players. We now assign the labels (from both K-means and Hierarchical Clustering) to the original data - in part to compare the discrepancy in assignment between the two methods but primarily to use it in the next section where we measure factor importance in group assignment with the hope to best define the new positions.

The two clustering techniques assign different groups - each method with its own unique benefits. For this work, heirarchical clustering is nice because of the nested structure - for example we may have a subgroup we'd call ball-dominant scorers; agglomerative clustering splits this into three point shooters and aggressive score-at-the-basket types - maintaining that they fall under a similar result category. This means that when we cut clusters off at a certain level (K=7) some clusters are considerable larger than others. K-Means is also useful here though as the clusters are relatively definitive - there is 'distance' between them and after deciding on K, assignment tends to be more rigid and a bit more uniform in count, allowing us to potentially identify the truly unique role-based positions. We will keep both clustering techniques in mind moving forward because some player groups should be smaller (There aren't that many true stars in the NBA but that may well be a role) - below are the counts for each cluster for each of the techniques.

Player Cluster Counts

Cluster	Agglomerative Count	K-Means Count
1	3442	862
2	1766	964
3	86	452
4	103	622
5	151	1015
6	144	1333
7	31	475

Revisiting Key Statistics

As a intermediary check - we revisit figure 1, where we had plotted trends in three point attempts and points per 48 by conventional position group; now though - we plot the same metrics but use our newly-defined clusters as role "positions"

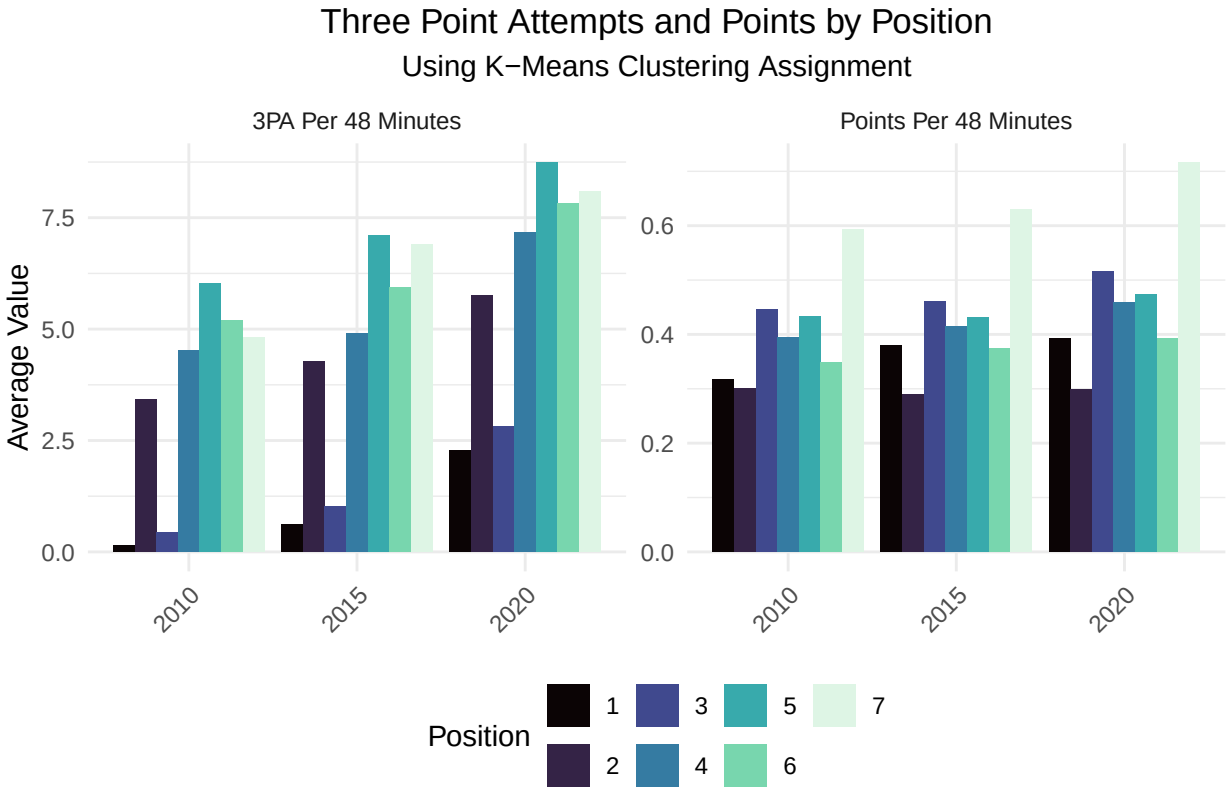

Figure 9: Cluster assignments and key on−court metrics from figure 2

The plots above show very promising results for the splitting of "positions" based on roles - with the plot on the right showing large variance in the scoring by each group, and even those group who score a similar amount (like clusters 3 and 5) have a very different way of doing so as seen through the average number of threes that each of those groups shoot.

Before we move to measuring variable importance in the determination of cluster assignment - we take a quick look at a randomly selected group of players in each of our newly assigned clusters.

Randomly Selected Players from Each Cluster

1	2	3
Richaun Holmes 2019	Iman Shumpert 2017	Tyson Chandler 2010
Damian Jones 2021	Wayne Selden 2016	Karl-Anthony Towns 2015
Chris Andersen 2015	T.J. Warren 2023	Kristaps Porzingis 2021
Precious Achiuwa 2022	Jordan McLaughlin 2020	Pau Gasol 2015
Jason Smith 2010	Brandon Goodwin 2019	Andre Drummond 2014

4	5	6	7
Rajon Rondo 2018	Thaddeus Young 2017	David Nwaba 2018	Paul George 2015
Evan Turner 2011	JR Smith 2015	Torrey Craig 2021	John Wall 2016
Jordan Farmar 2013	Ray Allen 2013	Doug McDermott 2023	Spencer Dinwiddie 2022
Tony Allen 2014	E'Twaun Moore 2016	Raymond Felton 2017	Isaiah Thomas 2015
Marcus Smart 2022	Carl Landry 2012	James Ennis III 2017	Giannis Antetokounmpo 2017

Position and Role Definition

Knowing that we have effectively split and clustered players is interesting - but the real value of this work comes from being able to identify and label each of the newly created archetypes. In the next section - we employ machine learning tools like random forest to understand the assignment (based on statistical on-court contributions) of each cluster group.

Part 3 | Decision Trees and Random Forest

Why Use Trees?

For this work, we want to understand what makes any given player a cluster 1 player so that we can define that group in a more understandable way for coaches and GMs trying to make decisions for their team - and so that we, as fans, can understand what makes good team composition.

We start with Classification and Regression Trees (CART) to capture and visualize the pertinent decision rules and then employ random forest - leveraging the nonparametric nature of these methods to, finally, define our new NBA roles.

CART

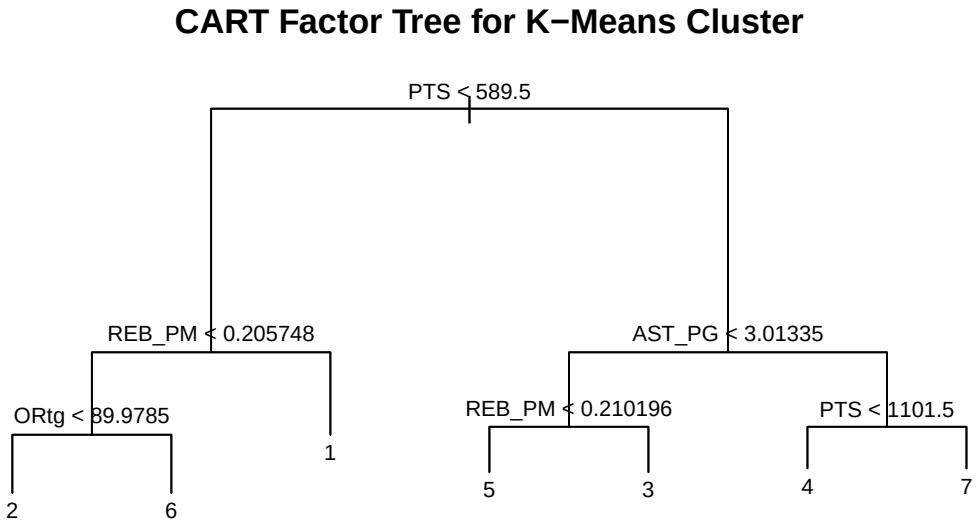

Figure 9: Decision tree for k−means clusters assignment based on stats

The tree above provides some very interesting insight - splitting first on total points for the season separating clusters 5,3,4, and 7 as higher scoring. Looking deeper into the "high scoring" branch we see that the first break comes from assists indicating another role-centric rank. After completing random forest below we will use those results in conjecture with this decision tree to define the roles more completely.

Random Forest

Below we use the non-parametric random forest approach to calculate variable importance in the assignment of our clusters.

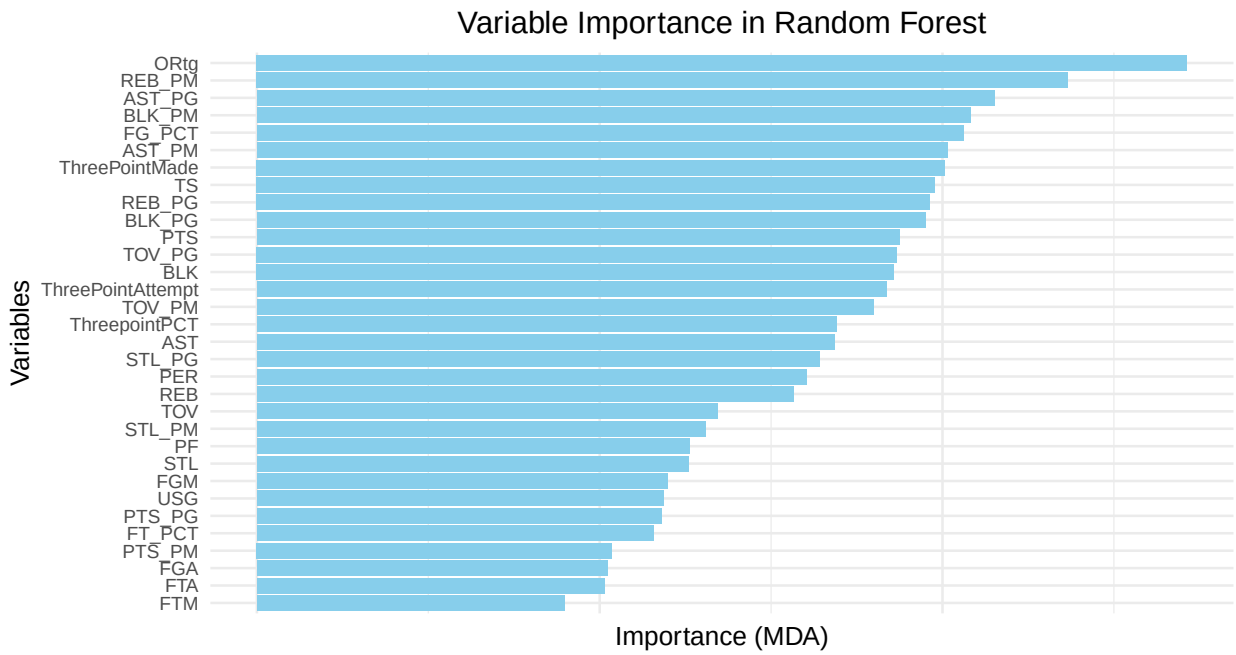

Figure 10: Random Forest variable importance in cluster assignment

Re-mapping With Key Statistics

To increase readability of rate/raw statistics we compose our trees again:

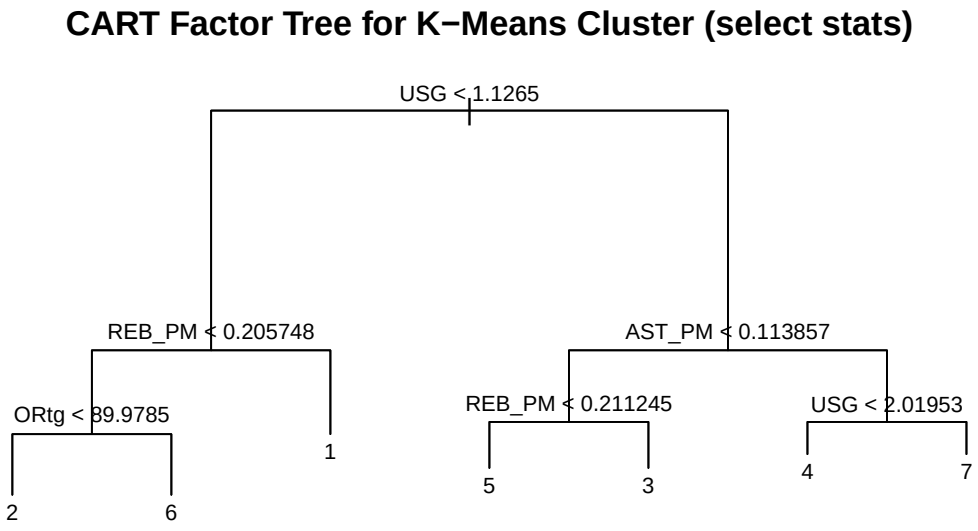

Figure 11: Additional decision tree for k−means clusters assignment

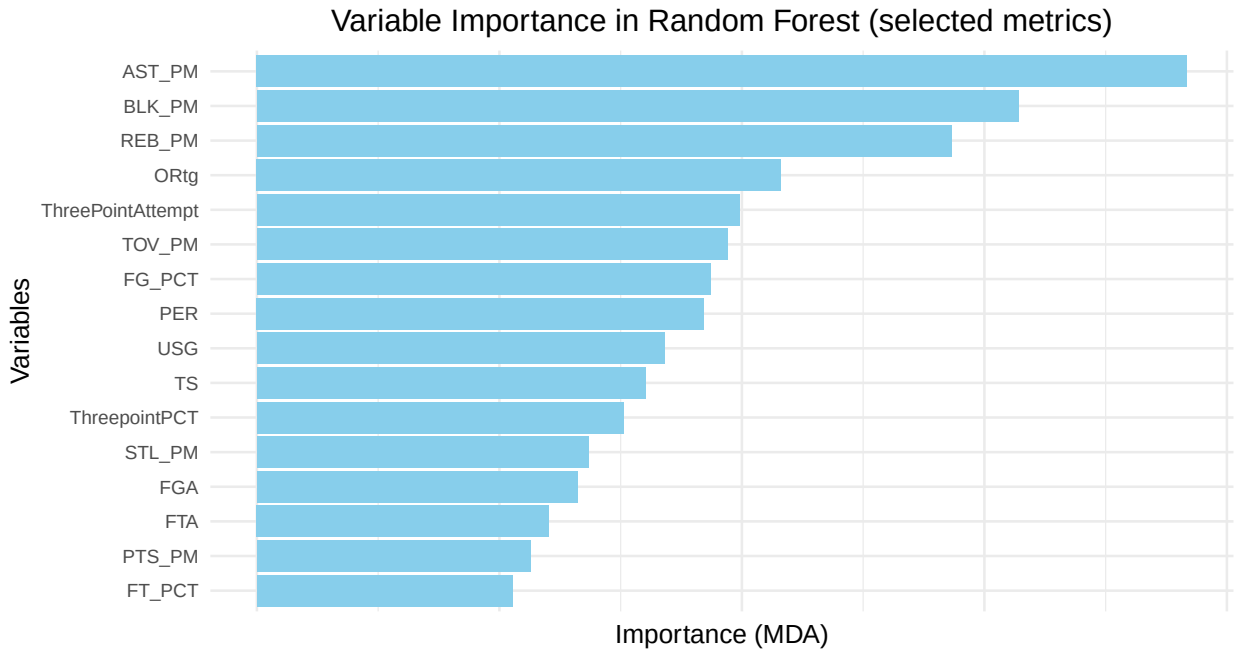

Figure 12: Random Forrest on selected rate−based metric

As we saw in the first two plots, the inclusion of both raw and rate (having points, points per game, and points per minute) complicates our variable importance interpretation slightly. They were included as they are crucial in understanding player role in absolute terms and relative to the opportunities they are given (players with lots of points but few points per minute are different from those with high points per minute - whether because of opportunities given or efficiency). Above we reform our tree using shooting statistics, per minute data, and the advanced statistics like usage and offensive rating - to make the visualization of our clusters easier.

Average Statistics by Cluster

Using the trees above - we find out a significant deal about the assignment of clusters and the overall importance of factors in our position definition problem. Before we name and define the 7 positions, lets look at some averages of the important statistics for each of the groups we have defined.

Cluster Summary of NBA Player Statistics

	Offensive Metrics				Shooting Metrics		
Cluster	Points Per Game	Assists Per Game	Usage %	Offensive Rating	Field Goal %	3-Point %	True Shooting %
1	6.767667	0.9507609	0.6830756	98.94726	0.5384086	0.1433567	0.5753550
2	4.487092	1.4182537	0.3802531	79.36876	0.3846154	0.2567573	0.4655727
3	14.303176	1.9276340	1.7937807	101.63415	0.5369339	0.1918976	0.5811942
4	11.492701	4.6795236	1.3801517	88.87085	0.4325162	0.3352381	0.5279572
5	12.910587	2.1666727	1.6407293	100.42151	0.4447245	0.3680185	0.5604524
6	6.934813	1.3402650	0.7085844	98.56128	0.4386088	0.3628343	0.5535687
7	22.449807	5.6829327	2.7925501	100.48168	0.4678567	0.3539726	0.5761280

Cluster Summary of NBA Player Statistics Cont.

	Defensive Metrics			Efficiency Metrics
Cluster	Rebounds Per Game	Blocks Per Game	Steals Per Game	Player Efficiency Rating
1	5.119749	0.7522187	0.5002297	5.030724
2	2.264636	0.2310390	0.5056233	2.272910
3	8.903624	1.3424469	0.7824876	10.043686
4	3.449519	0.2788878	1.0719725	6.073926
5	4.258525	0.3948135	0.8667121	7.774504
6	2.699624	0.2537837	0.5481973	4.224113
7	5.787063	0.5042584	1.2448098	13.250428

Using the information from this table - and the findings of our tree models above, we feel comfortable assigning labels to the newly created NBA roles.

Predicting Roles in the 2024-25 NBA Season

Thanks to the work done above - we can use new data to predict/assign cluster labels to the players in the current NBA Season.

We could predict the probability that each individual falls into each of the possible clusters but here we will choose to set our predict model to classify a cluster for each player. The table below shows a random selection of this years NBA players with their conventional position assignment grouped into their role-based position groups.

Randomly Selected Players from Each Cluster (2024 Season)

1	2	3
Jonas Valanciunas C	Kelly Olynyk C	Jakob Poeltl C
Kel'el Ware C	Jusuf Nurkic C	Onyeka Okongwu C
Richaun Holmes PF	Reggie Jackson PG	Jarrett Allen C
Kevon Looney C	Wendell Carter Jr. C	Anthony Davis C
Mo Bamba C	Jake LaRavia PF	Amen Thompson SF

4	5	6	7
Julius Randle PF	Nikola Vucevic C	Tre Mann PG	Tyrese Maxey PG
Cole Anthony PG	Dillon Brooks SF	Josh Green SG	Darius Garland PG
Tyus Jones PG	Corey Kispert SF	Jared Butler PG	Jayson Tatum PF
Tyrese Haliburton PG	Lauri Markkanen PF	Vit Krejci PG	Giannis Antetokounmpo PF
Jalen Johnson SF	Pascal Siakam PF	Caris LeVert SF	Jalen Green SG

This demonstrates how we can easily assign role based positions - and just based on the random selection we ended up with a lot of centers (C) in our sample, which shows how even though they have historically been defined as centers their roles are not identical. The ability to map player performance to our position groups is extremely valuable, and in the next section we define what these positions bring to the court.

Each of our newly defined position groups include members from multiple historical position assignments - that is because size no longer mirrors role. Historically a guy the size of Nikola Vucevic would have been expected to spend time close to the rim and defend - but here he is with a role that more closely reflects a perimeter player like Dillon Brooks. This is the benefit of this work - particularly when considering the names and details about the clusters from Part 4 - we can quickly understand the way an individual plays, and what that means for their team.

Example of Player Classification Using Decision Tree:

To put into practice how this classification model depicts more relevant role oriented roles - lets look at 2018 Malik Monk. Following the decision tree in Figure 11 - we compare his usage of 1.33 to the decision criteria and follow the right branch to the assists per minute break (with a value of 0.11), Monk's 0.092 do not make the cut and we go left - the final branch point is rebounds per minute with a threshold of 0.211; Monk once again falls below that and is assigned to cluster 5.

On its own it is not super interesting - but consider the other players with the same on court characteristics (those in cluster 5). Also in this cluster is 2022 Al Horford, a basketball player that conventional wisdom would tell you is vastly different from Malik Monk. Horford is 6'9" tall and typically labeled as a center of power forward, whereas Monk is 6'3" tall and classified in the obsolete position assignment as a shooting guard. By leveraging the tools in this work - we can see that these two players really bring the same thing to the table on a team. Defining groups by their roles and contribution captures much more than any obsolete physical-size based position assignment ever could.

Part 4 | Conclusion

The New NBA Positions

Using statistics about each group - particularly those that random forest found were most important in the determination of factor assignment - we describe the 7 new roles below, adding detail and examples to paint a clear picture of how each contributes to a team.

Cluster 1: The "Defense First Role-Player"

Examples: Taj Gibson, Tyson Chandler, Robin Lopez

- **Points Per Game:** Low (6.77), possible secondary scoring role.
- **Rebounds Per Game:** High (5.12), strong rebounding ability.
- **Blocks Per Game:** High (0.75), solid defensive contributions.
- **Usage:** Moderate/Low (0.68), smaller role - focused on other areas.

The Defense First Role-Player doesn't need the ball in their hands to make a difference - the benefit they provide comes from defense, rebounding, and decision making. With a high offensive rating, these guys don't make many mistakes with the opportunities they get. This group does not get the media fame but ask a coach and they will tell you how important guys who can defend and rebound without needing to be high usage are to winning teams.

Cluster 2: The "Benchwarmer"

Examples: Cody Martin, Ronnie Price

- **Points Per Game:** Very low (4.49), minimal scoring role.
- **Offensive Rating:** Very low (79.4), poor offensive players.
- **Rebounds and Assists Per Game:** Low (2.26 and 1.42), makes very few contributions.
- **Usage:** Low (0.38), indicating a very low level of involvement.

The name says it all - the Benchwarmer is the low contributing cluster. While every individual in the data plays more than 10 minutes per game - the benchwarmers do not do very well in that time. Assignment to this on-court role is never a good sign as it probably means the person should not be seeing the court - they take a high number of shots while on the court but do not seem to score well. All NBA players are top 0.001% athletes in the world, but one cluster has to capture those that aren't necessarily leading teams to victory - and the Benchwarmer role is the. This role also captures past-their-prime stars - who may add some locker room value but are not the players they once were.

Cluster 3: The "Two-Way Playmaker"

Examples: Karl-Anthony Towns, Anthony Davis, Tim Duncan

- **Points Per Game:** High (14.3), solid scoring ability.
- **Offensive Rating:** Very High (101.6), efficient and effective offensive player.
- **Rebounds Per Game:** Very high (8.9), excels at rebounding - athletic player.
- **BLK_PG:** High (1.34), solid rim protection.

- **PER:** High (10.04), efficient scorer - not a perimeter shooter.
- **Usage:** High (1.79), plays an important team role.

The Two-Way Playmaker is a primary option for most teams and a big athlete, while they do not shoot very many 3s - they score very efficiently close to the rim and are typically longer or more athletic players with the ability to impact the game offensively and defensively. Conventional big men and slashing athletes with the ability to score and lead teams on both ends of the court.

Cluster 4: The "Floor General"

Examples: Rajon Rondo, Chris Paul, Ricky Rubio

- **Points Per Game:** Moderate (11.49), solid scorer but not very efficient.
- **Rebounds Per Game:** Low (3.45), limited rebounding - perimeter player .
- **Assists Per Game:** Very High (4.68), great distributor and offense conductor.
- **Steals Per Game:** High (1.07), strong on-ball defender.
- **Usage:** Moderate (1.38), balanced offensive role - lacks some efficiency.

The Floor General runs the offense as the primary ball-handler - distributing to primary scoring options while racking up decent scoring statistics in a relatively inefficient manner. Likely tasked with guarding the other team's floor general they have decently strong defensive ability and although they may lack the size to rebound, this role is crucial for winning teams.

Cluster 5: The "Shooter"

Examples: JJ Reddick, Kyle Korver, Klay Thompson, Ray Allen

- **Points Per Game:** High (12.91), primary scorer.
- **Offensive Rating:** Very High (100.4), efficient and effective offensive player.
- **Assists Per Game:** Moderate (2.17), handles the ball a decent amount.
- **Usage:** High (1.64), heavily involved in offense.
- **3PT Percentage:** High (36.8%), strong perimeter shooters.

The shooter is the newest crucial archetype in the NBA - guys that can shoot the ball consistently. They do not dominate the ball on offense but they are heavily involved in the scheme, attempting the most threes and making them at a high rate. While capable on defense and in rebounding - this player brings one main thing to the table; their ability to shoot the increasing popular three.

Cluster 6: The "Offensive Glue"

Examples: Doug McDermott, Austin Rivers, Tony Snell

- **Points Per Game:** Moderate (6.93), secondary or supporting scorer.
- **Rebounds Per Game:** Low (2.70), not an elite rebounded.
- **Offensive Rating:** Moderate/High (98.5), strong in their moments.

- **Usage:** Low (0.71), low-usage offensive option.

The Offensive Glue is an offense-first role player. Though not a first or second option on the court, they do their job, shoot the ball well, and have good offensive rating/PER indicating efficiency and strong team play. The Offensive glue tends to be a less prevalent version of the shooter - good from three but with fewer touches and lower usage. They tend to play selflessly and are involved in offensive sets that create points for others.

Cluster 7: The "Superstar"

Examples: LeBron James, Kevin Durant, Steph Curry

- **Points Per Game:** Very High (22.45), dominant pure scorer.
- **Offensive Rating:** Very High (100.4), efficient and effective offensive player.
- **Assists Per Game:** High (5.68), playmaking ability.
- **PER:** High (13.25), very effective - elite player.
- **Usage:** Very High (2.79), central offensive figure.

The Superstar is the reason people watch the NBA - these are the high scoring, dominating players. Of different sizes and play-styles this role is very ball dominant and includes only those who can do it well. This player is the center of most great teams and nearly everything runs through them. We all know a superstar when we watch - and defining the positions in this way allows us to fit the other 6 around a superstar.

Conclusion and Predictive Power

This work, driven by a problem with the current assignment of positions unrelated to roles in the NBA, uses many tools from the course to effectively deconstruct the principal components of on-court play, cluster players in through multiple methods, and then use CART and Random Forest to understand between-group differences and inform the new naming and defining of our role-based positions.

As a predictive tool - this project uses tools with which parties can take any player (not even just NBA) and use my model (the CART model particularly) to predict what position group they fall in. While we do not explicitly mention the way this work can be used for prediction, the importance lies in being able to tell what a players on court contribution is in a simple manner. Whether considering trades or draft picks - we can now identify the contribution-focused role that a given player plays through prediction on these models. There is an endless number of applications in the basketball world and future work could benefit from team deconstruction to understand the interaction between player types.

Above, I use the key statistics from our decision trees and random forest outputs (like Offensive Rating, Usage, Points, Assists, and Rebounds) to distinguish in less statistical terms what these clusters mean - transitioning analytics into actionable insight. It is my hope that, in the future, basketball executives will leverage tools like this in their formation of teams.

Part 5 | Appendix

This appendix includes the selected R code for each of the sections:

Data Source

The data used for this work is originally from basketball_reference.com and was in part accessed through *NocturneBear*'s github repository.

Part 0 | Introduction

```
nbadata <- nbadata %>%
   mutate(season_year = substr(Season, 1, 4))

threeplot <- nbadata %>%
  filter(season_year >= 1979)

# ggplot(threeplot, aes(x = season_year, y = X3PA, group = 1)) +
#  geom_line(color = "blue", size = 1) +
#  scale_x_discrete(
#    breaks = seq(min(threeplot$season_year, na.rm = TRUE),
#                 max(threeplot$season_year, na.rm = TRUE), by = 5)
#  ) +
#  labs(title = "Three-Pointers Attempted Per Game in the NBA",
#       x = "Season Year",
#       y = "Average Three-Pointers Attempted",
#       caption = "Figure 1: Three pointers attempted per team per game in
# each NBA season since 1979") +
#  geom_vline(xintercept = "2010", linetype = "dashed", color = "gray", size = 1)+
#
#  theme_minimal()+
#    theme(plot.caption = element_text(hjust = 0.5, size = 10))
```

```
statsplot <- stats %>%
  filter(Year >=1980) # this dataset has the end year of the season as the year

statsplot$Pos <- sub("-.*", "", statsplot$Pos)

statsplot[is.na(statsplot)] <- 0
statsplot <- statsplot %>%
  mutate(points = FT*1 + X3P*3 + X2P*2)

statsplot1 <- statsplot %>%
  group_by(Year, Pos) %>%
  summarise(
    avg_points_per48 = mean(PTS/(MP/48), na.rm=T),
    avg_3PA = mean(X3PA/(MP/48), na.rm =TRUE),
    .groups = 'keep'
  )

statsplot2 <- statsplot1 %>%
  filter(Year %in% c(1990,1995,2000,2005,2010,2015))

data_long <- statsplot2 %>%
  gather(key = "stat_type", value = "value", avg_points_per48, avg_3PA)
```

```r
stat_labels <- c("avg_points_per48" = "Points Per 48 Minutes",
                 "avg_3PA" = "3PA Per 48 Minutes")


# Create the side-by-side bar plot
# ggplot(data_long, aes(x = Year, y = value, fill = Pos)) +
#  geom_bar(stat = "identity", position = "dodge") +
#  facet_wrap(~stat_type, scales = "free_y", labeller = labeller(stat_type = stat_labels)) +
#  scale_fill_viridis_d(option = "G") +
#  labs(x = NULL, y = "Average Value", title = "Three Point Attempts and Points
# by Position", fill = "Position", caption = "Figure 2: # The obsolete
# position label - shooting
# and scoring activity by position group over time") +
#  theme_minimal() +
#  theme(axis.text.x = element_text(angle = 45, hjust = 1),
#        legend.position = "bottom",
#        plot.caption = element_text(hjust = 0.5, size = 10),
#        plot.title = element_text(hjust = .5))
```

Part 1 | Principal Component Analysis

```r
players <- read.csv("Players.csv")
stats <- read.csv("Seasons_Stats.csv")
player_details <- read.csv("player_data.csv")
stats1 <- read.csv("regular_season_box_scores_2010_2024_part_1.csv")
stats2 <- read.csv("regular_season_box_scores_2010_2024_part_2.csv")
stats3 <- read.csv("regular_season_box_scores_2010_2024_part_3.csv")
boxscore <- rbind(stats1, stats2, stats3)

convert_to_minutes <- function(time_str) {
  if (is.na(time_str) || time_str == "") return(0)  # Handle empty cases
  parts <- strsplit(time_str, ":")[[1]]  # Split string at ":"
  as.numeric(parts[1]) + as.numeric(parts[2]) / 60  # Convert to total minutes
}
boxscore <- boxscore %>%
  mutate(minutes_numeric = sapply(minutes, convert_to_minutes))

season_stats <- boxscore %>%
  group_by(season_year, personName) %>%
  summarise(
    GP = sum(!is.na(minutes) & minutes != ""),    # Count games played
    MP = sum(as.numeric(minutes_numeric), na.rm = TRUE), # Sum minutes played
    FGM = sum(fieldGoalsMade, na.rm = TRUE),
    FGA = sum(fieldGoalsAttempted, na.rm = TRUE),
    "3PM" = sum(threePointersMade, na.rm = TRUE),
    "3PA" = sum(threePointersAttempted, na.rm = TRUE),
    FTM = sum(freeThrowsMade, na.rm = TRUE),
    FTA = sum(freeThrowsAttempted, na.rm = TRUE),
    REB = sum(reboundsTotal, na.rm = TRUE),
    AST = sum(assists, na.rm = TRUE),
    STL = sum(steals, na.rm = TRUE),
    BLK = sum(blocks, na.rm = TRUE),
    TOV = sum(turnovers, na.rm = TRUE),
```

```
    PF  = sum(foulsPersonal, na.rm = TRUE),
    PTS = sum(points, na.rm = TRUE),
    FG_PCT = ifelse(FGA > 0, FGM / FGA, 0),
    "3P_PCT" = ifelse(`3PA` > 0, `3PM` / `3PA`, 0),
    FT_PCT = ifelse(FTA > 0, FTM / FTA, 0),
    REB_PG = REB / GP,
    AST_PG = AST / GP,
    STL_PG = STL / GP,
    BLK_PG = BLK / GP,
    TOV_PG = TOV / GP,
    PTS_PG = PTS / GP,
    REB_PM = REB / MP,
    AST_PM = AST / MP,
    STL_PM = STL / MP,
    BLK_PM = BLK / MP,
    TOV_PM = TOV / MP,
    PTS_PM = PTS / MP,
    .groups = "keep"
  ) %>%
  mutate(season_start_year = substr(season_year, 1, 4))

# Simplified PER
season_stats <- season_stats %>%
  mutate( PER = (PTS + (0.4 * FGM) - (0.7 * FGA) - (0.4 * (FTA - FTM)) +
              (0.4 * STL) + (0.7 * BLK) - (0.7 * TOV) + (0.2 * PF) +
              (0.1 * REB)) / GP)

# True Shooting Percentage (TS%)
season_stats <- season_stats %>%
  mutate( TS = PTS / (2 * (FGA + 0.44 * FTA)))
# Usage
season_stats <- season_stats %>%
  mutate( USG = (FGA * (MP / 5) + (0.44 * FTA) * (MP / 5) + (TOV * 100)) / (MP * 100))

# Calculate Offensive Rating
season_stats <- season_stats %>%
  mutate( ORtg = (PTS * 100) / (FGA + 0.44 * FTA + TOV))
season_stats$season_year <- season_stats$season_start_year
```

```
numeric_data <- season_stats %>%
  select(season_year, personName, GP, MP, FGM, FGA,
         `3PM`, `3PA`, FTM, FTA, REB, AST, STL, BLK, TOV,
         PF, PTS, FG_PCT, `3P_PCT`, FT_PCT, REB_PG, AST_PG,
         STL_PG, BLK_PG, TOV_PG, PTS_PG, PER, TS, USG, ORtg,
         REB_PM, AST_PM, STL_PM, BLK_PM, TOV_PM, PTS_PM)

numeric_data <- na.omit(numeric_data) # Removing na rows - player didn't record stats
numeric_data <- numeric_data %>%
  filter(GP >= 6)
numeric_data <- numeric_data %>%
  filter(MP/GP > 10)

numeric_data <- numeric_data %>% select(season_year, personName, FGM, FGA,
```

```
                                        `3PM`, `3PA`, FTM, FTA,
                                        REB, AST, STL, BLK, TOV,
                                        PF, PTS, FG_PCT, `3P_PCT`,
                                        FT_PCT, REB_PG, AST_PG,
                                        STL_PG, BLK_PG, TOV_PG,
                                        PTS_PG, PER, TS, USG,
                                        ORtg, REB_PM, AST_PM, STL_PM,
                                        BLK_PM, TOV_PM, PTS_PM)

# Scale the numeric data (excluding year and name)
scaled_data <- scale(numeric_data[, -c(1, 2)])

# PCA
pca_result <- prcomp(scaled_data, center = TRUE, scale. = TRUE)

pca_scores <- as.data.frame(pca_result$x)

pca_scores$season_year <- numeric_data$season_year
pca_scores$personName <- numeric_data$personName
pca_scores$Name_yr <- paste(pca_scores$personName, pca_scores$season_year)
```

SCREE PLOT

```
variance_explained <- pca_result$sdev^2 / sum(pca_result$sdev^2)

# Create a data frame for plotting
scree_data <- data.frame(PC = seq_along(variance_explained), Variance = variance_explained)
scree_data2 <- scree_data %>%
  filter(PC <= 12)

# ggplot(scree_data2, aes(x = PC, y = Variance)) +
#  geom_point(size = 3, color = "blue") +
#  geom_line(group = 1, color = "lightblue") +
#  labs(title = "SCREE Plot of Principal Components", x = "Principal Component",
#       y = "Proportion of Variance Explained",
#       caption = "Figure 4: SCREE plot of explained variance for the first 12 PCs") +
#  scale_x_continuous(breaks = 1:12) +
#  theme_minimal()+
#      theme(plot.caption = element_text(hjust = 0.5, size = 10),
#         plot.title = element_text(hjust = .5))
```

```
rotation_vals <- pca_result$rotation[, 1:5]

# Function to get top 5 loadings (including sign)
top_signed_rotations <- lapply(1:5, function(pc) {
  pc_name <- colnames(rotation_vals)[pc]  # Get PC name (PC1, PC2, etc.)
  sorted_vars <- sort(abs(rotation_vals[, pc]), decreasing = TRUE)  # Sort by absolute value
  top_vars <- names(sorted_vars[1:5])  # Get variable names of top 5
  top_values <- rotation_vals[top_vars, pc]
  # Create a data frame
  data.frame(
    Principal_Component = pc_name,
    Variable = top_vars,
```

```
    Rotation_Value = top_values
  )
})

# Combine into one data frame
top_signed_rotations_df <- do.call(rbind, top_signed_rotations)

PC1 <- top_signed_rotations_df %>%
  filter(Principal_Component == "PC1") %>%
  select(Variable, Rotation_Value)
PC2 <- top_signed_rotations_df %>%
  filter(Principal_Component == "PC2") %>%
  select(Variable, Rotation_Value)
PC3 <- top_signed_rotations_df %>%
  filter(Principal_Component == "PC3") %>%
  select(Variable, Rotation_Value)
PC4 <- top_signed_rotations_df %>%
  filter(Principal_Component == "PC4") %>%
  select(Variable, Rotation_Value)
PC5 <- top_signed_rotations_df %>%
  filter(Principal_Component == "PC5") %>%
  select(Variable,Rotation_Value)
```

Top 5 Rotations by Magnitude for Principal Component 1-4

```
# PC1 %>%
#  gt() %>%
#  tab_header(
#    title = "Principal Component 1")

# PC2 %>%
#  gt() %>%
#  tab_header(
#    title = "Principal Component 2")

# PC3 %>%
#  gt() %>%
#  tab_header(
#    title = "Principal Component 3")

# PC4 %>%
#  gt() %>%
#  tab_header(
#    title = "Principal Component 4")
```

Using LASSO to Select PCs

```
# Predict
zPCA <- predict(pca_result)

USG <- numeric_data$USG
USG <- scale(USG)
USG <- as.vector(USG)
```

```r
lassoPCR <- cv.gamlr(x=zPCA, y=USG, nfold=20)

## Plot

# plot(lassoPCR, main = "LASSO CV Principal Component Selection",
#     xlab = "Log(lambda)",
#     ylab = "Mean Squared Error (MSE)",
#     cex = 1.2)
# mtext("Figure 5: Cross validated LASSO principal component selection",
# side = 1, line = 4,cex = .75)
```

LASSO for PCs (Accounting for Complexity of USG)

```r
numeric_data2 <- numeric_data %>%
  select(!USG)

scaled_data2 <- scale(numeric_data2[, -c(1, 2)])

pca_result2 <- prcomp(scaled_data2, center = TRUE, scale. = TRUE)

# Predict
zPCA <- predict(pca_result2)

USG <- numeric_data$USG
USG <- scale(USG)

lassoPCR <- cv.gamlr(x=zPCA, y=USG, nfold=20, lambda.min.ratio = .05)

## Plot

# plot(lassoPCR, main = "High Penalty LASSO CV Principal Component Selection",
#     xlab = "Log(lambda)",
#     ylab = "Mean Squared Error (MSE)",
#     cex = 1.2)
# mtext("Figure 6: Withough USG cross-validated LASSO principal
# component selection", side = 1, line = 4,cex = .75)
```

Part 2 | Clustering

K-Means Clustering using 7 Principal Components

```r
# our 7

pca_clustering <- cbind(numeric_data[, c("personName", "season_year")], pca_result$x[, 1:7])

wss <- numeric(10)  # Store within-cluster sum of squares
for (k in 1:10) {
  km <- kmeans(pca_clustering[, -c(1,2)], centers = k, nstart = 25)
  wss[k] <- km$tot.withinss
}

# ggplot(data.frame(k = 1:10, wss = wss), aes(x = k, y = wss)) +
#  geom_point() + geom_line() +
```

```
#  labs(title = "Elbow Method for Optimal Cluster Selection", x = "Number of Clusters (k)",
# y = "Within-Cluster Sum of Squares",
#       caption = "Figure 7: K-Means Clustering and WSS for K Selection"
#       ) +
#    scale_x_continuous(breaks = 1:10) +
#  theme_minimal()+
#        theme(plot.caption = element_text(hjust = 0.5, size = 10),
#         plot.title = element_text(hjust = .5))
```

Hierarchical Clustering

```
player_info <- pca_clustering[, c("personName", "season_year")]

# Remove non-numeric columns before clustering
invisible({
  pca_numeric_clust <- pca_clustering %>%
    ungroup() %>%
  select(-season_year, -personName)
})

dist_matrix <- dist(pca_numeric_clust, method = "euclidean")

hc <- hclust(dist_matrix, method = "complete")

dendro <- as.dendrogram(hc)

dendro_data <- ggdendrogram(hc, rotate = FALSE, size = 2)
dendroplot <-suppressWarnings({
  ggplot(dendro_data$plot_env$data$segments, aes(x = x, y = y, xend = xend, yend = yend)) +
    geom_segment() +
    theme_minimal() +
   scale_y_continuous(limits = c(15, NA)) +
    labs(title = "Hierarchical Clustering Dendrogram",
       x = NULL,
       y = "Height",
       caption = "Figure 8: Dendrogram for
       agglomerative clustering - cutting only the top clusters") +
    theme(plot.caption = element_text(hjust = 0.5, size = 10),
          plot.title = element_text(hjust = .5),
          axis.text.x = element_blank())
})
```

Assigning Cluster Labels

```
set.seed(123)  # Set seed for reproducibility
kmeans_result <- kmeans(pca_clustering[, -c(1, 2)], centers = 7, nstart = 25)
numeric_data$K_cluster <- kmeans_result$cluster

#Hierarchical

cluster_labels <- cutree(hc, k = 7)

numeric_data$H_cluster <- cluster_labels
```

```r
numeric_data <- numeric_data %>%
  mutate(same_cluster = (K_cluster == H_cluster))

clustercountsK <- numeric_data %>%
  group_by(K_cluster) %>%
  summarise(n_people = n())
```

```r
clustercountsH <- numeric_data %>%
  group_by(H_cluster) %>%
  summarise(n_people3 = n())

clustercountsH$npeople9 <- clustercountsK$n_people

# clustercountsH %>%
#  gt() %>%
#  cols_label(
#    H_cluster = "Cluster",
#    n_people3 = "Agglomerative Count",
#    npeople9 = "K-Means Count"
#  ) %>%
#  tab_header(
#    title = "Player Cluster Counts"
#  )
```

Revisiting Key Statistics

```r
numeric_data66 <- season_stats

numeric_data66 <- na.omit(numeric_data66) # Removing na rows - player didn't record stats
numeric_data66 <- numeric_data66 %>%
  filter(GP >= 6)
numeric_data66 <- numeric_data66 %>%
  filter(MP/GP > 10)
statsplot5 <- numeric_data66

statsplot5$K_cluster <- numeric_data$K_cluster

ploty <- statsplot5 %>%
  group_by(season_year, K_cluster) %>%
  summarise(
    avg_points_per48 = mean(PTS_PM, na.rm=T),
    avg_3PA = mean(`3PA`/(MP/48), na.rm =TRUE),
    .groups = 'keep'
  )

statsplot8 <- ploty %>%
  filter(season_year %in% c(2010,2015,2020))

data_long2 <- statsplot8 %>%
  gather(key = "stat_type", value = "value", avg_points_per48, avg_3PA)

stat_labels <- c("avg_points_per48" = "Points Per 48 Minutes", "avg_3PA" = "3PA Per 48 Minutes")
```

```r
# Create the side-by-side bar plot
# ggplot(data_long2, aes(x = season_year, y = value, fill = as.factor(K_cluster))) +
#  geom_bar(stat = "identity", position = "dodge") +
#  facet_wrap(~stat_type, scales = "free_y", labeller = labeller(stat_type = stat_labels)) +
#  scale_fill_viridis_d(option = "G") +
#  labs(x = NULL, y = "Average Value", title = "Three Point Attempts and
# Points by Position",
# fill = "Position", caption = "Figure 9: Cluster assignments and
# key on-court metrics from figure 2",
#       subtitle = "Using K-Means Clustering Assignment") +
#  theme_minimal() +
#  theme(axis.text.x = element_text(angle = 45, hjust = 1),
#        legend.position = "bottom",
#        plot.caption = element_text(hjust = 0.5, size = 10),
#        plot.title = element_text(hjust = .5),
#        plot.subtitle = element_text(hjust = .5))
```

```r
selected_players <- numeric_data  %>%
  group_by(K_cluster) %>%
  slice_sample(n = 5) %>%
  ungroup()

selected_players$NameYear <- paste(selected_players$personName, selected_players$season_year)
players_wide <- selected_players %>%
  select(K_cluster, NameYear) %>%
  mutate(row_id = row_number()) %>%  # Create an index for reshaping
  tidyr::pivot_wider(names_from = K_cluster, values_from = NameYear) %>%
  select(-row_id)

fortable <- read.csv("selected_players.csv")

# fortable %>%
#  select(`X1`, `X2`, `X3`) %>%
#  gt() %>%
#  tab_header(title = "Randomly Selected Players from Each Cluster") %>%
#    cols_label(
#    `X1` = "1",
#    `X2` = "2",
#    `X3` = "3",
#  ) %>%
#    tab_style(
#    style = cell_text(size = px(8)),
#    locations = cells_body()
#  ) %>%
# tab_options(
#    table.font.size = 11)
```

Part 3 | Decision Trees and Random Forest

CART

```r
xforcart <- numeric_data %>%
  ungroup() %>%
  select(-season_year, -personName, -H_cluster, -same_cluster) %>%
  mutate(K_cluster = as.factor(K_cluster)) %>%  # Ensure K_cluster is a factor
  na.omit()  # Remove missing values

# Ensure only existing columns are selected (including K_cluster)
valid_columns <- intersect(names(xforcart),
                           c("K_cluster", "FGM", "FGA", "3PM", "3PA",
                             "FTM", "FTA", "REB", "AST",
                             "STL", "BLK", "TOV", "PF", "PTS",
                             "FG_PCT", "3P_PCT", "FT_PCT",
                             "REB_PG", "AST_PG", "STL_PG",
                             "BLK_PG", "TOV_PG", "PTS_PG", "PER",
                             "TS", "USG", "ORtg", "REB_PM",
                             "AST_PM", "STL_PM", "BLK_PM",
                             "TOV_PM", "PTS_PM"))

xforcart <- xforcart %>%
  select(all_of(valid_columns)) %>%
  mutate(across(where(is.integer), as.numeric))  # Ensure integers are numeric
cluster <- xforcart$K_cluster
xforcart <- xforcart %>%
  rename("ThreePointMade" = "3PM",
         "ThreePointAttempt" = "3PA",
         "ThreepointPCT" = "3P_PCT")

# Fit decision tree model using K_cluster as response
ktree <- tree(K_cluster ~. , data = xforcart)
ktreepruned <- prune.tree(ktree, best = 7)
# Plot tree
# plot(ktreepruned)
# title("CART Factor Tree for K-Means Cluster")
# mtext("Figure 9: Decision tree for
# k-means clusters assignment based on stats",
# side = 1, line = 4,cex = .75)
# text(ktreepruned, pretty = 0, cex = .75)
```

Random Forest

```r
rfkmeans <-  randomForest(K_cluster ~ ., data= xforcart, importance=TRUE)

# varImpPlot(rfkmeans,type=1)

importance_data <- randomForest::importance(rfkmeans, type = 1)

# Convert to a data frame for ggplot
importance_df <- data.frame(Variable = rownames(importance_data),
                            Importance = importance_data[,1])

# ggplot(importance_df, aes(x = reorder(Variable, Importance), y = Importance)) +
#  geom_bar(stat = "identity", fill = "skyblue") +
#  coord_flip() +  # Flip coordinates to make it easier to read
```

```
#  labs(title = "Variable Importance in Random Forest",
#       x = "Variables",
#       y = "Importance (MDA)",
#       caption = "Figure 10: Random Forest variable importance in cluster assignment") +
#  theme_minimal() +
#  theme(axis.text = element_text(size = 8),
#        plot.title = element_text(hjust = .5),
#        axis.text.x = element_blank(),
#        plot.caption = element_text(hjust = 0.5, size = 10)
#        )
```

Re-mapping With Key Statistics

```
xforcart44 <- numeric_data %>%
  ungroup() %>%
  select(-season_year, -personName, -H_cluster, -same_cluster) %>%
  mutate(K_cluster = as.factor(K_cluster)) %>%  # Ensure K_cluster is a factor
  na.omit()  # Remove missing values

# Ensure only existing columns are selected (including K_cluster)
valid_columns44 <- intersect(names(xforcart44),
                           c("K_cluster", "FGA",
                             "3PA", "FTA", "FG_PCT",
                             "3P_PCT", "FT_PCT", "PER",
                             "TS", "USG", "ORtg", "REB_PM",
                             "AST_PM", "STL_PM", "BLK_PM",
                             "TOV_PM", "PTS_PM"))

xforcart44 <- xforcart44 %>%
  select(all_of(valid_columns44)) %>%
  mutate(across(where(is.integer), as.numeric))  # Ensure integers are numeric

xforcart44 <- xforcart44 %>%
  rename("ThreePointAttempt" = "3PA",
         "ThreepointPCT" = "3P_PCT")

# Fit decision tree model using K_cluster as response
ktree44 <- tree(K_cluster ~. , data = xforcart44)
ktreepruned44 <- prune.tree(ktree44, best = 7)
# Plot tree
#plot(ktreepruned44)
#title("CART Factor Tree for K-Means Cluster (select stats)")
#mtext("Figure 11: Additional decision tree
# for k-means clusters assignment", side = 1, line = 4,cex = .75)
#text(ktreepruned44, pretty = 0, cex = .75)
```

```
rfkmeans44 <-  randomForest(K_cluster ~ ., data= xforcart44, importance=TRUE)

# varImpPlot(rfkmeans,type=1)

importance_data44 <- randomForest::importance(rfkmeans44, type = 1)

# Convert to a data frame for ggplot
```

```r
importance_df44 <- data.frame(Variable = rownames(importance_data44),
                              Importance = importance_data44[,1])

# ggplot(importance_df44, aes(x = reorder(Variable, Importance), y = Importance)) +
#  geom_bar(stat = "identity", fill = "skyblue") +
#  coord_flip() +  # Flip coordinates to make it easier to read
#  labs(title = "Variable Importance in Random Forest (selected metrics)",
#       x = "Variables",
#       y = "Importance (MDA)",
#       caption = "Figure 12: Random Forrest on selected rate-based metric") +
#  theme_minimal() +
#  theme(axis.text = element_text(size = 8),
#        plot.title = element_text(hjust = .5),
#        axis.text.x = element_blank(),
#        plot.caption = element_text(hjust = 0.5, size = 10)
#        )
```

Average Statistics by Cluster

```r
# averages %>%
#  select(K_cluster, REB_PG, BLK_PG, STL_PG, PER)%>%
#  gt() %>%  tab_header(
#    title = "Cluster Summary of NBA Player Statistics Cont."
#  ) %>%
#  cols_label(
#    K_cluster = "Cluster",
#    REB_PG = "Rebounds Per Game",
#    BLK_PG = "Blocks Per Game",
#    STL_PG = "Steals Per Game",
#    PER = "Player Efficiency Rating"
#  ) %>%
#  tab_spanner(label = "",
#              columns = c(K_cluster)) %>%
#  tab_spanner(
#    label = "Defensive Metrics",
#    columns = c(REB_PG, BLK_PG, STL_PG)
#  ) %>%
#  tab_spanner(
#    label = "Efficiency Metrics",
#    columns = c(PER)
#  ) %>%
#  tab_style(
#   style = list(
#      cell_fill(color = "#DCE6F1"),
#      cell_text(weight = "bold")
#    ),
#    locations = cells_column_labels()
#  ) %>%
#  tab_style(
#    style = cell_text(size = px(8)),
#    locations = cells_body()
#  ) %>%
#  tab_options(
```

```
#    table.font.size = 10,
#    table.border.top.style = "solid",
#    table.border.top.color = "black",
#    table.border.bottom.style = "solid",
#    table.border.bottom.color = "black"
#  )
```

```
thisyear <- read_excel("thisyear.xlsx")

forpredict <- thisyear %>%
  mutate(
    REB_PM = REB / MP,
    AST_PM = AST / MP,
    STL_PM = STL / MP,
    BLK_PM = BLK / MP,
    TOV_PM = TOV / MP,
    PTS_PM = PTS / MP
  ) %>%
  mutate(season_start_year = 2024)
forpredict <- na.omit(forpredict)

forpredict <- forpredict %>%
  filter(MP/G >=10)

# Simplified PER
season_statspred <- forpredict %>%
  mutate( PER = (PTS + (0.4 * FGM) - (0.7 * FGA) - (0.4 * (FTA - FTM)) +
              (0.4 * STL) + (0.7 * BLK) - (0.7 * TOV) + (0.2 * PF) +
              (0.1 * REB)) / G)

# True Shooting Percentage (TS%)
season_statspred <- season_statspred %>%
  mutate( TS = PTS / (2 * (FGA + 0.44 * FTA)))
# Usage
season_statspred <- season_statspred %>%
  mutate( USG = (FGA * (MP / 5) + (0.44 * FTA) * (MP / 5) + (TOV * 100)) / (MP * 100))

# Calculate Offensive Rating
season_statspred <- season_statspred %>%
  mutate( ORtg = (PTS * 100) / (FGA + 0.44 * FTA + TOV))


## Predict

predictingnumers <- season_statspred %>%
  select("FGA", "3PA", "FTA", "FG_PCT", "3P_PCT",
                              "FT%", "PER",  "TS", "USG", "ORtg", "REB_PM",
                              "AST_PM", "STL_PM", "BLK_PM", "TOV_PM", "PTS_PM")

predictingnumers <- predictingnumers %>%
  rename("ThreePointAttempt" = "3PA",
         "ThreepointPCT" = "3P_PCT",
         "FT_PCT" = "FT%")
```

```
forpredict$Cluster <- predict(ktree44, predictingnumers, type="class")

set.seed(59)
selected_players2024 <- forpredict  %>%
  group_by(Cluster) %>%
  slice_sample(n = 5) %>%
  ungroup()
selected_players2024$Name_Position <- paste(selected_players2024$Player, selected_players2024$Pos)

players_wide2024 <- selected_players2024 %>%
  select(Cluster, Name_Position) %>%
  mutate(row_id = row_number()) %>%  # Create an index for reshaping
  tidyr::pivot_wider(names_from = Cluster, values_from = Name_Position) %>%
  select(-row_id)


selected_playersnewest <- read.csv("selectpredict.csv")


# selected_playersnewest %>%
#  select(`X1`, `X2`, `X3`) %>%
##  gt() %>%
#  tab_header(title = "Randomly Selected Players from Each Cluster (2024 Season)") %>%
#    cols_label(
#   `X1` = "1",
#    `X2` = "2",
##    `X3` = "3",
#  ) %>%
#    tab_style(
#   style = cell_text(size = px(8)),
#    locations = cells_body()
#  ) %>%
#  tab_options(
#    table.font.size = 11)#
```

```
#selected_playersnewest %>%
#  select(`X4`, `X5`, `X6`, `X7`) %>%
#  gt() %>%
#    cols_label(
#    `X4` = "4",
#    `X5` = "5",
#    `X6` = "6",
#    `X7` = "7"
#  ) %>%
#    tab_style(
#    style = cell_text(size = px(8)),
#    locations = cells_body()
#  ) %>%
#  tab_options(
#    table.font.size = 11)#
```