8-2022

Redefining NBA Basketball Positions Through Visualization and Mega-Cluster Analysis

Alexander L. Hedquist
Utah State University

REDEFINING NBA BASKETBALL POSITIONS THROUGH VISUALIZATION AND MEGA-CLUSTER ANALYSIS

by

Alexander L. Hedquist

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Statistics

Approved:

Jürgen Symanzik, Ph.D. Major Professor	Brennan Bean, Ph.D. Committee Member
Kevin Moon, Ph.D. Committee Member	D. Richard Cutler, Ph.D. Vice Provost for Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2022

ABSTRACT

Redefining NBA Basketball Positions Through Visualization and Mega-Cluster Analysis

by

Alexander L. Hedquist, Master of Science

Utah State University, 2022

Major Professor: Jürgen Symanzik, Ph.D.
Department: Mathematics and Statistics

In basketball, player positions constitute the simplest and most widely-used tool to characterize members of a team. While the standard five positions, including Point Guard, Shooting Guard, Small Forward, Power Forward, and Center provide general categories for certain major types of players, these vague position titles limit players to a pre-defined role, and limit coaches' and managers' ability to recruit, draft, and utilize players in an effective manner. This MS thesis proposes a method for expanding the current basketball positions to define players based on their abilities and performance rather than based on height, weight, or perceived role. We analyze players from the past 20 seasons of the National Basketball Association (NBA) to determine updated and meaningful player positions. We utilize a collection of indices in R to select nine as an optimal number of player clusters. We perform hierarchical cluster analysis to regroup players into nine meaningful and specific categories. Using R and Python, we explore the differences between these player clusters through visualization techniques, such as dendograms and histograms, and dimensionality reduction methods, including Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (tSNE), and Potential of Heat-Diffusion for Affinity-Based Trajectory Embedding (PHATE). We also use a grand tour software feature to explore these updated player clusters in a more dynamic and interactive fashion. Finally, we introduce a new method called *mega-clustering* that allows us to partition each NBA season's player clusters into combined clusters for an overall analysis and discussion of each position's unique attributes. In addition, we assemble all player data and clustering results into a single GitHub repository for easy access and further analysis.

(134 pages)

PUBLIC ABSTRACT

Redefining NBA Basketball Positions Through Visualization and Mega-Cluster Analysis

Alexander L. Hedquist

Basketball players have historically been classified based on one of five positions, namely Point Guards, Shooting Guards, Small Forwards, and Centers. While grouping players into these five categories may provide general descriptions of their perceived role, these standard positions fall short of describing players based on their true abilities and performance. This MS thesis proposes a method to group players of the National Basketball Association (NBA) from the past 20 seasons into more meaningful and specific player positions. We systematically group these players into nine distinct categories, and we draw from a vast array of visualization tools, techniques, and software to view and analyze these new player positions and compare them to the standard roles currently used by the basketball community. These visualization tools and methods allow us to view highly complex data with many variables in low-dimensional plots that are both meaningful and interpretable. Each season's nine player positions are then grouped into nine overall positions across the 20-year span and their unique attributes and behaviors will be explored in depth. All of the player tables, the individual player position assignments, and many other relevant data tables are assembled and included on a single online repository for public access and use.

ACKNOWLEDGMENTS

I express gratitude to the Utah State University Mathematics & Statistics Department for their exceptional financial and emotional support. An incredible staff has made my experience at Utah State impactful and enjoyable.

I am grateful to my committee members, Dr. Brennan Bean and Dr. Kevin Moon, for their continued support and willingness to participate and provide feedback. Their graduate courses in which I participated provided inspiration for some ideas presented in this thesis.

I acknowledge the immeasurable impact of my advisor, Dr. Jürgen Symanzik. His incredible patience and tireless attention to detail taught me profound life lessons about the importance of precision and thoroughness. His patience with my errors and sometimes slow progress, as well as his continued interest in the contents and subject matter of this thesis are appreciated beyond measure. Jürgen has been one of my favorite professors in all my years of education, and I am grateful to have him as a friend.

Finally, I would like to acknowledge and thank my wife, Aimee, for her continued patience and support through this arduous process. The events of the past three years, including a car accident, moving to a new city, and the birth of our first child, Jack, have impacted us dramatically, but her love and support have remained constant.

CONTENTS

List of Tables

List of Figures

CHAPTER 1

Introduction

1.1 Background

Basketball is one of the most popular sports in the world. In 2021, the International Basketball Federation (FIBA) estimated that 450 million people play basketball at some level worldwide (FIBA, 2021). The pinnacle of the basketball world is certainly the National Basketball Association (NBA). In 2021, Game 6 of the NBA Finals between the Milwaukee Bucks and the Phoenix Suns peaked at 16.54 million viewers worldwide (NBA, 2021). With the NBA's popularity growing globally, the way the game is played is changing rapidly. Players, coaches, and managers are discovering new and innovative ways to play the game, and players' roles and abilities are adapting to these new approaches. The conventional method to create lineups is by selecting one player from each of the standard five positions of basketball, but with the constant evolution of the game, coaches and managers must become more precise in categorizing players if they want to achieve the highest possible performance out of their lineups.

1.1.1 NBA Basketball Standard Positions

The game of basketball is contested between two teams with five players from each team on the floor at a time. Historically, these players have been assigned a position and a number based on their role on the court. These positions are: Point Guard (one), Shooting Guard (two), Small Forward (three), Power Forward (four), and Center (five). Teams and coaches may choose to play multiple players from the same position on the court at once (e.g., Two Power Forwards and no Center), but the standard lineup structure contains one player from each of the five positions with a dynamic and flexible set of roles.

The Point Guard is the player who generally brings the ball up the court and runs the offense. This type of player will frequently call out plays and sets to get certain players good shots. Point Guards are usually very fast and can score from the outside and inside. They generally have more assists (passes immediately preceding a made basket) than other players and are strong ball-handlers and dribblers. Some famous NBA Point Guards include John Stockton and Stephen Curry.

The Shooting Guard is a player who shares many of the same characteristics of a Point Guard, but their primary goal is to shoot on offense. These players often move without the ball and get open for quick shots. They often play on the 'wings', while the point guard plays more in the middle of the court. Some famous Shooting Guards include Michael Jordan and Kobe Bryant.

The Small Forward is a very versatile player. These players have a wide range of roles depending on the team and the game situation. They are generally very strong defenders, and tend to be a bit larger and taller than the 'guard' players. These players can be great outside shooters like the Shooting Guards, but may also be adept at finishing around the basket and rebounding. Some famous Small Forwards include LeBron James and Kevin Durant.

The Power Forward is often a larger version of the Small Forward. These players frequently play around the low 'blocks' or the 'post' by the basket, and are proficient mid-range scorers. These players are strong and can guard big players under the basket. Some famous Power Forwards include Tim Duncan and Karl Malone.

The Center is usually one of the tallest players on the team. Centers are strong defenders and shot-blockers, and deter smaller players from driving to the basket. Centers generally score most of their points in the painted area. Centers also have high rebound totals and set lots of screens for ball-handlers. Some famous Centers include Kareem Abdul-Jabbar and Shaquille O'Neal.

1.1.2 The Limits of Standard Positions

While these standard positions have been the most common approach to creating lineups and classifying players, these categories do not paint the whole picture of player abilities or their true role on the floor. For example, John Stockton and Stephen Curry are both classified as Point Guards. However, their roles on the court are incredibly different, and these players would normally not be talked about in the same sentence. John Stockton fits more of the standard definition for a Point Guard. He is the NBA's all-time leader in assists and steals. While Stockton was a threat to score, he generally had very modest scoring averages, especially when compared to Stephen Curry's scoring potential. Curry is widely considered as the greatest three-point shooter of all time, and recently surpassed Ray Allen, a well-known Hall of Fame Shooting Guard, for the most three-pointers made all time in December of 2021. John Stockton and Stephen Curry can and should be classified into different positions since John Stockton was a pass-first Point Guard, while Stephen Curry was a score-first Point Guard.

Similar inconsistencies can be found across all seasons and players. We can also find examples of players who play multiple positions on the floor. LeBron James and Kevin Durant are technically classified as Small Forwards, but these two players have been known to play all five positions throughout their career. Kevin Durant stands at almost seven feet tall, giving him the height of a Center. He has the mid-range shooting and post-up ability of a Power Forward, the length and quickness on defense of a Small Forward, and the shooting ability of some of the best Shooting Guards. Durant and James also frequently bring the ball up the floor and are considered the 'floor generals', similar to the role of a Point Guard.

In the last decade, the NBA has experienced a major surge in three-point shooting, largely due to the increased understanding and use of basketball analytics (Schuhmann, 2021). With more threes being attempted, and with the mid-range jump shot on the decline, players who can stretch the floor and play inside and outside are highly sought after. Teams and coaches are adjusting their strategy on both defense and offense in response to the increased three-point shooting across all NBA teams.

Forcing a taller player to play the traditional Power Forward or Center position when he or she is a great ball-handler and outside shooter will diminish that player's ability to impact the game. How can we classify these players more accurately to paint a correct picture of player roles and lineup compositions? How can we determine which players truly have similar roles and which players have very different roles? The answer is through cluster analysis.

1.1.3 Previous Research Into 'Updated' Player Positions

While the mainstream classification of basketball players still revolves around the standard five positions, research into 'updated' or 'advanced' player positions has been conducted in the past decade.

Alagappan (2012) used a method called topological data analysis based on player shot charts to classify players from the 2010-2011 NBA season into new positions that were more indicative of their roles on the court. He proposed stretching "from 5 to 13" positions and described how these updated positions can improve team building, player management, and recruiting.

Kalman and Bosch (2020) used data from the 2009-2010 NBA season to the 2018-2019 NBA season to perform model-based clustering. Nine clusters were chosen to restructure player positions. Specific players were tracked to see how their roles evolved over the course of their career. These new positions were compared to the standard positions and regression and random forest models were constructed to predict lineup performances based on their compositions.

Finally, Jyad (2020) analyzed the 2018-19 NBA season using Principal Component Analysis (PCA) to explore the characteristics that appear to distinguish players the most. Hierarchical cluster analysis was performed to group players into nine different positions. These new player positions were analyzed in detail to determine their unique attributes.

1.1.4 Motivation

The current framework for classifying basketball players does not allow for confident decision-making when building lineups and teams. Players are short-changed when their array of unique skills and abilities are mischaracterized to fit an extremely broad and vague definition. The assumption that one player from each of these standard positions must be present on the floor limits a team's ability to respond to game flow and unique matchups. The advent of so-called 'small lineups' in today's game provides an example of how teams are departing from the standard lineup composition to try and make the opposing team uncomfortable.

Creating updated player positions will open the door to more advanced and focused lineups and will allow players, coaches, and managers to create the optimal lineups for specific matchups and game situations. These new-and-improved player categories will allow players to play their true role on the court, rather than forcing them into a standard role that does not match their abilities.

This MS thesis aims to expand on previous research by analyzing and visualizing player clusters in more detail. While certain visualization techniques are introduced by previous authors, we will give considerable attention to a wide array of static and dynamic visualization techniques for exploring player cluster distinctions. While Alagappan (2012) and Jyad (2020) explored player characteristics in a single season and Kalman and Bosch (2020) analyzed a span of ten seasons, this research will consider twenty NBA seasons of player statistics to determine the dominant player characteristics that have prevailed over time. This MS thesis will differ from previous work by clustering based on easily trackable game statistics rather than more advanced and less intuitive metrics like assist rates, efficiency, and shot locations.

1.2 Overview

We will begin in Chapter 2 by discussing the 20 seasons of NBA player data and variables that will be used for the cluster analysis and exploration of the new player positions. Attention will be given to the data cleaning and processing performed to enhance the proceeding methods. The GitHub page (`https://github.com/ahed1194/MS_Thesis`) with all relevant data tables as well as all R (R Core Team, 2021) and Python (Van Rossum and Drake, 2009) code used to retrieve and prepare this data will be provided and summarized. Zuccolotto et al. (2021) introduced how to summarize basketball data in R through visualizations of player statistics and game statistics, and they introduced a `BasketballAnalyzeR`

R package that provides helpful tools and functions for analyzing basketball data.

Following the presentation of the data, Chapter 3 will explore the various methods used to analyze the NBA player data. An overview of some major types of clustering will be provided, followed by a discussion of ways to select the optimal number of clusters for a particular data set. An overview of various validity checks and dimensionality reduction methods will be presented. Finally, this chapter will provide descriptions and links to documentation for all R packages, Python packages, and other software employed to carry out these analyses.

In Chapter 4, we will determine the optimal number of player positions for an individual season and for all 20 seasons combined. Justification for this choice will be provided through various dimensionality reduction visualizations in R, Python, and GGobi (Cook and Swayne, 2007).

Chapter 5 will begin by providing the results of the consistency measures of our clustering algorithm from season to season. Next, we will present the clustering results and key characteristics of these clusters for players in the 2000-2001 NBA season. We will then discuss a new technique called *mega-clustering*. A description of this method will be provided. This will be followed by an in-depth visual and numerical analysis of these 'mega-clusters' and their distinguishing features. Finally, we will compare the *mega-clustering* results to those obtained through clustering all 20 seasons combined.

The results obtained through Chapters 4 and 5 will be discussed in detail in Chapter 6. Reasoning behind the specific number of clusters chosen will be provided. We will specifically compare and contrast the various visualization techniques based on how they contribute to the analysis and effective visualization of cluster differences. The positions defined through the single season and combined season clusterings will then be compared and analyzed.

We will conclude in Chapter 7 with a glimpse into the wide range of applications of this analysis, followed by some proposed future improvements and complementary research that can be performed.

Appendices A, B, C, D, and E can be consulted for additional discussions and insights.

In Appendix A, we will discuss variations in the lower cutoff for minutes played. In Appendix B, we will provide the details of the indices used to choose the optimal cluster number. In Appendix C, we will analyze and discuss how variations in the lower and upper limit parameters affect the optimal cluster selections. In Appendix D, we will view the simulations of the consistency measures for cluster differences from season to season. Finally, in Appendix E, we will briefly discuss the results of clustering players from the 2000-2001 NBA season into only three clusters instead of nine.

All relevant data and code can be found at the following GitHub link:

https://github.com/ahed1194/MS_Thesis

Specifically, the reader may access the following tables in the following sub-folders of the above URL:

- `Player_Data`: All the individual player tables with their career statistics
- `Lineup_Data`: The 20 lineup tables with the five-man lineup combinations and their statistics
- `R_Code`: The R code used to scrape and analyze the NBA player data
- `Python_Code`: The Python code used for visualizing the player clusters
- `Player_Cluster`: The scaled player data with their cluster assignments by season
- `Mega_Cluster`: The 'mega-cluster' assignments for each season's clusters

CHAPTER 2

Data Overview

In this chapter, we will provide the source for the individual NBA player data as well as the lineup data used in this MS thesis. We will discuss the many variables included in our data sets and how they were manipulated and normalized to prepare for further analysis. Finally, we will provide information on how to access these scraped NBA player and lineup tables as well as the code used to generate all figures and tables that will be presented.

2.1 Accessing Individual and Team Data

As the original goal of this research was to cross-analyze individual player statistics with lineup performances comprised of five players, data was compiled for all active NBA players from the 2000-2001 season to the 2019-2020 season. All lineup combinations during the same time-frame were also extracted. Basketball Reference (Basketball Reference, 2022) provides incredibly thorough records of all games and statistics recorded since well before the merger of the NBA and ABA (the two major American basketball leagues) back in 1976. This website allows users to extract most data free of charge, and even has the option to convert most tables to .csv files or other easy-to-use formats.

Data was extracted over a 20-year span by using the `read_html` function in the `rvest` R package (Wickham, 2020b). While the individual player data is still available on Basketball Reference, the lineup data has since been moved to a subscription-only section called Stathead (`https://stathead.com/basketball/`).

Once the individual and team lineup tables for all 20 years were scraped, it became necessary to match the five names from each lineup row with each player's unique identifier. Every player who has ever appeared on an NBA roster possesses a unique identifier which consists of the first five letters of his last name, the first two letters of his first name, and then '01' if the player is the first with the given first and last name. For example, Stephen

Curry's unique page for his career statistics and history is found at

https://www.basketball-reference.com/players/c/curryst01.html.

For duplicate appearances of a player reference, a '02' is added if he is the second, a '03' if he is the third, and so on. For example, three people with the name 'George Johnson' have played in the NBA. The third player to appear in the NBA named George Johnson has this unique webpage:

https://www.basketball-reference.com/players/j/johnsge03.html

Data for all players who recorded data from the 2000-2001 NBA season to the 2019-2020 NBA season was collected and saved as individual .csv files on the GitHub page (https://github.com/ahed1194/MS_Thesis). The individual player data can be found by accessing the `Player_Data` sub-directory. The team lineup data by season can be found by accessing the `Lineup_Data` sub-directory.

2.2 Data Description

There are two major data pools discussed in this research. The individual player data is used extensively in this research, while the team lineup data is merely mentioned as a counterpart for potential future applications. Each data pool is discussed in detail below.

2.2.1 Player Data

The player data is comprised of all NBA players who were active on any of the 30 franchises between the 2000-2001 season and the 2019-2020 season. It is important to note that not all of these players were included in the subsequent clustering algorithms and classifications since some players did not record enough minutes to be considered for the cluster analysis (see Section 2.3.1). We should also note that only regular season statistics will be used for this analysis. Playoff minutes vary more widely between players since some players played much of their career on teams in the upper half of the rankings, while other players may have appeared very little in the playoffs due to being on a poor team. Playoff games are starkly different from regular season games since a team plays an opponent up to seven times to determine who advances to the next round. This leads to different strategies and player uses, as well as fewer bench players being utilized due to varying strategies. There is also little reason to rest certain players and go deeper into the bench since it is the end of the season.

The variables included in each one of these player tables are shown in Table 2.1 along with their descriptions. Table 2.2 shows Stephen Curry's career table, and Table 2.3 shows a closeup of some of Curry's statistics by season. While more advanced statistics were available on other pages on Basketball Reference, the focus of this analysis is on classifying players based on data that can be viewed or easily computed from a box score. Statistics like rebounds, steals, assists, turnovers, and points may not provide a comprehensive display of a player's true value, but they shed light on a player's role on a given team, regardless of their value.

It is also important to note the format of the variables in each player table (see Table 2.1 for details and brief explanations of all variables in the player tables). If we move top to bottom on Table 2.1, we can see `Season`, `Age`, and `Team (Tm)`, which are used to identify a player. We will treat each year and each team as if it is a completely new player. With players constantly changing teams mid-season and during the off-season, players' roles change, and they frequently change positions. Stephen Curry may be classified as a traditional Point Guard for the first three seasons, and then transition to a Shooting Guard, for example. In Table 2.4, we can see the last five rows of Dwyane Wade's career. In the 2016-2017 season, he played for the Chicago Bulls (CHI). In the 2017-2018 season, he began with the Cleveland Cavaliers (CLE), but then was traded mid-season to the Miami Heat (MIA). We can also see a total (TOT) row listed above the two different team rows. This total row was removed from the clustering analysis, and Dwyane Wade's two rows for the 2017-2018 season are treated as different players with a unique identifier including his team abbreviation.

Table 2.1: Label explanations for individual NBA player tables (labels are precisely as seen on Basketball Reference (2022))

LABEL	NAME	EXPLANATION
Season	Season	NBA Season listed from fall of the first year to spring of the next year (final two numbers of second year) (Ex: 2019-20)
Age	Age	Players age at the start of the season
Tm	Team	One of 30 teams, listed as three-letter abbreviation (Ex: GSW = Golden State Warriors)
Lg	League	League that the player participates in. For this analysis, all listings are 'NBA'.
Pos	Position	One of five standard player positions: 'PG' = Point Guard, 'SG' = Shooting Guard, 'SF' = Small Forward, 'PF' = Power Forward, 'C' = Center
G	Games Played	Games played in given season. Max games in a season is 82.
GS	Games Started	Games for which the given player was one of five starters.
MP	Minutes Played	AVERAGE minutes played per 36 minutes
FG	Field Goals Made	AVERAGE field goals made per 36 minutes
FGA	Field Goals Attempted	AVERAGE field goals attempted per 36 minutes
FG%	Field Goal Percentage	Field goal percentage for the entire season
3P	Three-Pointers Made	AVERAGE three-pointers made per 36 minutes
3PA	Three-Pointers Attempted	AVERAGE three-pointers attempted per 36 minutes
3P%	Three-Pointer Percentage	Three-point percentage for the entire season
2P	Two-Pointers Made	AVERAGE two-pointers made per 36 minutes
2PA	Two-Pointers Attempted	AVERAGE two-pointers attempted per 36 minutes
2P%	Two-Pointer Percentage	Two-point percentage for the entire season
FT	Free Throws Made	AVERAGE free throws made per 36 minutes
FTA	Free Throws Attempted	AVERAGE free throws attempted per 36 minutes
FT%	Free Throw Percentage	Free throw percentage for the entire season
ORB	Offensive Rebounds	AVERAGE offensive rebounds per 36 minutes
DRB	Defensive Rebounds	AVERAGE defensive rebounds per 36 minutes
TRB	Total Rebounds	AVERAGE total rebounds per 36 minutes
AST	Assists	AVERAGE assists per 36 minutes
STL	Steals	AVERAGE steals per 36 minutes
BLK	Blocks	AVERAGE blocks per 36 minutes
TOV	Turnovers	AVERAGE turnovers per 36 minutes
PF	Personal Fouls	AVERAGE personal fouls per 36 minutes
PTS	Points	AVERAGE points per 36 minutes

Continuing top to bottom on Table 2.1, `Games Played (G)`, `Games Started (GS)`, and `Minutes Played (MP)` are displayed as totals for a given season, while all variables from `Field Goals Made (FG)` to `Points (PTS)` are listed as per 36 minutes average. This deci-

Table 2.2: Stephen Curry career statistics obtained from https://www.basketball-reference.com/players/c/curryst01.html

	Season	Age	Tm	Lg	Pos	G	GS	MP	FG	FGA	FG%	3P	3PA	3P%	2P	2PA	2P%	FT	FTA	FT%	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS
1	2009-10	21	GSW	NBA	PG	80	77	2896	6.6	14.2	0.462	2.1	4.7	0.437	4.5	9.5	0.474	2.2	2.5	0.885	0.6	3.8	4.4	5.9	1.9	0.2	3	3.1	17.4
2	2010-11	22	GSW	NBA	PG	74	74	2489	7.3	15.2	0.48	2.2	4.9	0.442	5.1	10.3	0.498	3.1	3.3	0.934	0.8	3.4	4.1	6.2	1.6	0.3	3.3	3.4	19.9
3	2011-12	23	GSW	NBA	PG	26	23	732	7.1	14.6	0.49	2.7	6	0.455	4.4	8.6	0.514	1.9	2.3	0.809	0.7	3.6	4.3	6.8	1.9	0.4	3.2	3	18.8
4	2012-13	24	GSW	NBA	PG	78	78	2983	7.6	16.8	0.451	3.3	7.2	0.453	4.3	9.5	0.449	3.2	3.5	0.9	0.7	3.1	3.8	6.5	1.5	0.1	2.9	2.4	21.6
5	2013-14	25	GSW	NBA	PG	78	78	2846	8.2	17.5	0.471	3.3	7.8	0.424	4.9	9.7	0.509	3.9	4.4	0.885	0.6	3.6	4.2	8.4	1.6	0.2	3.7	2.5	23.7
6	2014-15	26	GSW	NBA	PG	80	80	2613	9	18.5	0.487	3.9	8.9	0.443	5.1	9.6	0.528	4.2	4.6	0.914	0.8	3.9	4.7	8.5	2.2	0.2	3.4	2.2	26.2
7	2015-16	27	GSW	NBA	PG	79	79	2700	10.7	21.3	0.504	5.4	11.8	0.454	5.4	9.5	0.566	4.8	5.3	0.908	0.9	4.8	5.7	7	2.3	0.2	3.5	2.1	31.7
8	2016-17	28	GSW	NBA	PG	79	79	2638	9.2	19.7	0.468	4.4	10.8	0.411	4.8	8.9	0.537	4.4	4.9	0.898	0.8	4	4.8	7.2	1.9	0.2	3.3	2.5	27.3
9	2017-18	29	GSW	NBA	PG	51	51	1631	9.4	19.1	0.495	4.7	11.1	0.423	4.8	8	0.595	6.1	6.7	0.921	0.8	5	5.8	6.8	1.8	0.2	3.4	2.5	29.7
10	2018-19	30	GSW	NBA	PG	69	69	2331	9.8	20.7	0.472	5.5	12.5	0.437	4.3	8.2	0.525	4.1	4.4	0.916	0.7	5	5.7	5.6	1.4	0.4	3	2.6	29.1
11	2019-20	31	GSW	NBA	PG	5	5	139	8.5	21.2	0.402	3.1	12.7	0.245	5.4	8.5	0.636	6.7	6.7	1	1	5.7	6.7	8.5	1.3	0.5	4.1	2.8	26.9
12	Career	NA		NBA		699	693	23998	8.5	17.9	0.476	3.7	8.6	0.435	4.8	9.3	0.515	3.8	4.2	0.906	0.7	4	4.7	6.9	1.8	0.2	3.3	2.6	24.6

Table 2.3: Stephen Curry career statistics - Showing the first 6 rows and first 13 variables

	Season	Age	Tm	Lg	Pos	G	GS	MP	FG	FGA	FG%	3P	3PA
1	2009-10	21	GSW	NBA	PG	80	77	2896	6.6	14.2	0.462	2.1	4.7
2	2010-11	22	GSW	NBA	PG	74	74	2489	7.3	15.2	0.48	2.2	4.9
3	2011-12	23	GSW	NBA	PG	26	23	732	7.1	14.6	0.49	2.7	6
4	2012-13	24	GSW	NBA	PG	78	78	2983	7.6	16.8	0.451	3.3	7.2
5	2013-14	25	GSW	NBA	PG	78	78	2846	8.2	17.5	0.471	3.3	7.8
6	2014-15	26	GSW	NBA	PG	80	80	2613	9	18.5	0.487	3.9	8.9

Table 2.4: Dwyane Wade's final five rows and first 13 variables obtained from https://www.basketball-reference.com/players/w/wadedw01.html

	Season	Age	Tm	Lg	22Pos	G	GS	MP	FG	FGA	FG%	3P	3PA
...	...	...	...	...	...	...	...	...	...	...	...	...	...
14	2016-17	35	CHI	NBA	SG	60	59	1792	8.3	19.2	0.434	0.9	2.9
15	2017-18	36	TOT	NBA	SG	67	3	1536	7	16	0.438	0.7	2.6
16	2017-18	36	CLE	NBA	SG	46	3	1069	6.7	14.6	0.455	0.8	2.4
17	2017-18	36	MIA	NBA	SG	21	0	467	7.8	19	0.409	0.7	3.2
18	2018-19	37	MIA	NBA	SG	72	2	1885	7.9	18.3	0.433	1.6	5

sion was made due to the volatility of games played. If Player A plays half the season, and is injured for the second half, his total points, rebounds, assists, etc. would appear much lower than another Player B who played a full season, even if Player A's game-to-game output was higher.

2.2.2 Lineup Data

While the team lineup data was not used in the research presented in this MS thesis, we should still discuss its variables and dimensions, since its contents will be extremely useful for further research. Tables 2.5 and 2.6 display the first rows of the 2019-2020 season lineup combinations. Like the individual player data, these tables contain only regular season data.

Lineup table observations consist of five unique players, the team, and the season.

Table 2.5: First five rows and first twelve variables of 2019-2020 season lineups (Ordered by `Minutes Played`). Obtained from https://www.basketball-reference.com/play-index/lineup_finder in May 2019. Note that this link is no longer valid. See Section 2.1

ranker	lineup	team_id	season	g	mp	poss	opp_poss	pace	fg	fga	fg_pct
1	W. Barton | G. Harris | N. Jokic | P. Millsap | J. Murray	DEN	2019-20	38	735.3	1474	1453	95.5	42.3	89.3	0.474
2	B. Bogdanovic | R. Gobert | J. Ingles | D. Mitchell | R. O'Neale	UTA	2019-20	47	570.5	1164	1155	97.6	41.4	80.8	0.513
3	J. Allen | S. Dinwiddie | J. Harris | T. Waller-Prince | G. Temple	BRK	2019-20	43	490.9	1003	999	97.9	39.8	89.1	0.447
4	B. Adebayo | J. Butler | M. Leonard | K. Nunn | D. Robinson	MIA	2019-20	39	487.4	956	955	94.1	41.8	82.7	0.505
5	D. Brooks | J. Crowder | J. Jackson | J. Morant | J. Valanciunas	MEM	2019-20	36	413.7	868	861	100.3	44.2	91.8	0.482

Table 2.6: First five rows and final five variables of 2019-2020 season lineups. Obtained from https://www.basketball-reference.com/play-index/lineup_finder in May 2019. Note that this link is no longer valid. See Section 2.1

X1	X2	X3	X4	X5
/players/b/bartowi01.html	/players/h/harriga01.html	/players/j/jokicni01.html	/players/m/millspa01.html	/players/m/murraja01.html
/players/b/bogdabo02.html	/players/g/goberru01.html	/players/i/inglejo01.html	/players/m/mitchdo01.html	/players/o/onealro01.html
/players/a/allenja01.html	/players/d/dinwisp01.html	/players/h/harrijo01.html	/players/p/princta02.html	/players/t/templga01.html
/players/a/adebaba01.html	/players/b/butleji01.html	/players/l/leoname01.html	/players/n/nunnke01.html	/players/r/robindu01.html
/players/b/brookdi01.html	/players/c/crowdja01.html	/players/j/jacksja02.html	/players/m/moranja01.html	/players/v/valanjo01.html

Other identifiers were added to the end of the tables, including each player's unique reference linked to the Basketball Reference website. These identifiers could be useful for matching individual players with their lineup combinations and performances. Please note that variable abbreviations are lower case for the lineup tables, while they are upper case for the individual player tables.

`Games (g)` and `Minutes Played (mp)` variables were again listed as totals for the entire season. `Possessions (poss)` and `Opponent Possessions (opp_poss)` were listed as totals as well. The rest of the data from `Pace` all the way to `Point Differential (diff_pts)` are listed as averages per 48 minutes playing time to control for lineups that played very little time together.

2.3 Data Manipulation

Once the data was downloaded and stored by player references and season references, the process of setting up the data for analysis began. In this section we will note important changes and filters placed on the data in an attempt to make the subsequent analysis more meaningful.

2.3.1 Lower Limit for Minutes

It is important that we try to include only meaningful minutes played in each game, and eliminate the 'garbage time'. 'Garbage time' is the time in the game when one team is blowing out the other and the outcome has already been decided. At this point, coaches usually pull their star players and put reserves in who don't play many minutes. Garbage time plays more like an exhibition match and should be excluded from further analysis as much as possible, without eliminating any meaningful playing time.

While a cutoff of 24 minutes played (two quarters) per player for each season, higher cutoffs were tested to ensure no major differences in the optimal number of clusters selection. The reader is invited to consult Appendix A for further details and discussion on this topic.

2.3.2 Missing Values

Missing values were located on many player tables in percentage categories. For example, Shaquille O'Neal played most of his seasons without attempting a three-pointer (See `https://www.basketball-reference.com/players/o/onealsh01.html`). This resulted in 0's on `3FG` and `3FGA`, but resulted in NA's on `3FG_pct`. These NA values were set to -0.1. This allowed for players who never attempted a three-point shot to be included in the analysis, but also allowed for a distinction between players who attempted no three-pointers and players who attempted one or more three-pointers and missed all of them.

2.3.3 Normalizing the Data

Once these necessary manipulations were performed, the numerical columns beginning with `FG` all the way to `PTS` were normalized using the `scale` function from base R.

Normalizing the data before analysis has many benefits. It allows for all input variables to be equally treated in models. A player's points-per-game average is likely to be notably higher than their blocks-per-game average, but it can be argued that one block is far more valuable than one point in a game. Many models are based on Euclidean distances between points in determining loss and other important statistics, and we do not want these values to be skewed by differing variable ranges.

Another benefit of normalizing data applies to machine learning. Many machine learning algorithms require the data to be properly normalized or scaled in order to converge to some output (Baijayanta, 2020). There are many potential applications of this research in the field of machine learning and regression that will be discussed in Section 7.2.

2.4 Public Availability

A major contribution of this research to the sports community is the acquisition and cleaning of the player and lineup data. The potential applications and uses of these player tables are limitless. The fact that the lineup data is no longer freely acquired makes this web scraping work even more valuable. A GitHub repository has been made available to house all relevant tables, code, and results (`https://github.com/ahed1194/MS_Thesis`). The individual player tables are found in the `Player_Data` sub-folder and team lineup tables from the 2000-2001 NBA season to the 2019-2020 NBA season are found in the `Team_Data` sub-folder. The reader may also access all player and season clustering results in the `Player_Cluster` and `Mega_Cluster` sub-folders, respectively, to find where any specific players have been classified that were not mentioned in this MS thesis. Finally, all relevant R code and Python code used to acquire and analyze the data are available for public use in the `R_Code` and the `Python_Code` sub-directories. Assuming no changes occur in the Basketball Reference interface, interested individuals and parties may run the R code and scrape to-date player tables to a local drive.

CHAPTER 3

Methods

In this chapter, we will discuss the essentials of data clustering and the various algorithms that can be performed. We will also look into other methods used to verify and enhance our analysis of clustering player positions, such as the Adjusted Rand Index, Principal Component Analysis, tSNE, and PHATE. Finally, we will outline the R packages and other software approaches that were used in this MS thesis.

3.1 What is Clustering?

This section provides a brief informative discussion of different types of clustering while outlining the particular algorithms and methods relevant to this data analysis. Statistical clustering, or cluster analysis, refers to placing observations into meaningful groups (Kaufman and Rousseeuw, 1990, pp. 1-67). Clustering groups similar data points and seeks to exclude points that are beyond some similarity threshold. Clustering data can be done using one of many algorithms depending on the type of data and the end goal. In this research setting, we will focus primarily on the difference between two highly popular clustering methods: hierarchical clustering and k-means clustering.

3.1.1 Hierarchical vs k-means Clustering

Hierarchical clustering and k-means clustering are two fundamentally different ways to approach classifying data points into groups. The former focuses on matching pairs of points that are close together or 'similar' to one another, while the latter focuses on the proximity of individual data points to a cluster's centroid, or local optima (Kaushik and Mathur, 2014). As one would imagine, there are advantages and limits to both of these methods.

k-means clustering is a type of 'centroid' clustering where the number of clusters is pre-determined and the data points are classified based on their proximity to a particular centroid. This type of partitioning works well for large data sets, but requires advanced knowledge of how many ways to divide the data in order to achieve meaningful separations. Convergence is guaranteed in this scenario since a data point will always have a 'closest neighbor', but the interpretability of cluster separations may prove difficult to impossible (Kaushik and Mathur, 2014).

Hierarchical clustering allows the user to stop at any step in the division or agglomeration process. The agglomerative algorithms generally begin with each data point as its own cluster. The most similar clusters are then combined, and this process is iterated until all data points are part of one big cluster. The data points can be combined until variability has reached a certain point or has leveled off (Larose and Larose, 2014). Choosing the number of partitions can be somewhat arbitrary, but hierarchical clustering does have the advantage of interpretability. If the goal is to produce a natural hierarchy of elements, hierarchical clustering methods provide a step-by-step process of inclusion. For NBA player positions, for example, we may observe scoring point guards and passing point guards be combined into a 'point guard' cluster. We could further see an inclusion of shooting guards and point guards into a 'guard' cluster.

A well-known disadvantage of hierarchical clustering is the potential computational cost due to large data sets. This method requires the storage of dissimilarity matrices for each element, and can greatly increase processing time (Kaufman and Rousseeuw, 1990).

Given the benefits and drawbacks, as well as the example of player position data given above, it became evident that hierarchical clustering would be the better choice to proceed with the NBA player data. We already have some prior knowledge about player characteristics, and we would like to have some interpretability for the cluster separations. We will now look at the different types of hierarchical clustering, including the method selected for partitioning the data in this MS thesis.

3.1.2 Different Types of Hierarchical Clustering

The hierarchical clustering algorithms considered for the cluster analysis of the NBA data are: Single Linkage, Complete Linkage, Average Linkage, and Ward's Distance-Squared method (Ferreira and Hitchcock, 2009; Murtagh and Contreras, 2017). While these are not the most complex clustering algorithms available, they comprise some of the most frequently used methods across all fields of study (Rokach and Maimon, 2005). These methods are all known as Agglomerative Nesting (AGNES) methods, or Hierarchical Agglomerative Clustering (HAC) methods (Kaufman and Rousseeuw, 1990). We will begin with a brief description of each clustering method, followed by a comparison. For further comparisons and specific calculations for these clustering methods and related methods, the reader is invited to consult the works of Ferreira and Hitchcock (2009) and Murtagh and Contreras (2017).

Single Linkage: This method combines points into clusters one by one based on a minimum distance between two points in two clusters. It is one of the oldest methods of agglomerative hierarchical clustering. It has the disadvantage of combining groups that may have one pair of points with a small distance, but the overall group is highly distinct, or *dissimilar*.

Complete Linkage: This method combines clusters together based on the maximum distance between points in different clusters. This method carries a similar disadvantage to single linkage since it can be heavily influenced by outliers and will often place points with relatively small distances between each other into different clusters.

Average Linkage: This method combines clusters together iteratively by measuring the average distance between all points in one group and all points in another group. Groups are combined that have the smallest average distance from each other. This method could be seen as an improvement on the limitations mentioned with single and complete linkage.

Ward's Distance-Squared (Ward D2): This method differs from the previously mentioned linkage methods in that it does not group by some distance measure, but rather a within-cluster sum of squares. For this reason, this method is sometimes referred to as the *Ward minimum variance* method. At each step in the agglomeration process, a new cluster is made that minimizes this within sum of squares measure. An important distinction must be made between Ward's method and Ward's Distance-Squared method. They are often used synonymously and perform similarly, but the distance-squared method is more frequently used since it highlights the distances between objects and makes them easier to distinguish and partition. Murtagh and Legendre (2014) discussed the differences between Ward's method and Ward's Distance-Squared method and clarified some overgeneralizations and misunderstandings about them.

We can use dendograms to visually compare the four methods mentioned above (Ferreira and Hitchcock, 2009). Dendograms are plots that show the hierarchical relationship between observations. For this introductory example, we will use the `mtcars` dataset from base R. Table 3.1 displays the 32 rows and 11 columns of this data set.

Table 3.1: mtcars data set (rounded to 1 decimal place)

model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6.0	160.0	110.0	3.9	2.6	16.5	0.0	1.0	4.0	4.0
Mazda RX4 Wag	21.0	6.0	160.0	110.0	3.9	2.9	17.0	0.0	1.0	4.0	4.0
Datsun 710	22.8	4.0	108.0	93.0	3.9	2.3	18.6	1.0	1.0	4.0	1.0
Hornet 4 Drive	21.4	6.0	258.0	110.0	3.1	3.2	19.4	1.0	0.0	3.0	1.0
Hornet Sportabout	18.7	8.0	360.0	175.0	3.2	3.4	17.0	0.0	0.0	3.0	2.0
Valiant	18.1	6.0	225.0	105.0	2.8	3.5	20.2	1.0	0.0	3.0	1.0
Duster 360	14.3	8.0	360.0	245.0	3.2	3.6	15.8	0.0	0.0	3.0	4.0
Merc 240D	24.4	4.0	146.7	62.0	3.7	3.2	20.0	1.0	0.0	4.0	2.0
Merc 230	22.8	4.0	140.8	95.0	3.9	3.2	22.9	1.0	0.0	4.0	2.0
Merc 280	19.2	6.0	167.6	123.0	3.9	3.4	18.3	1.0	0.0	4.0	4.0
Merc 280C	17.8	6.0	167.6	123.0	3.9	3.4	18.9	1.0	0.0	4.0	4.0
Merc 450SE	16.4	8.0	275.8	180.0	3.1	4.1	17.4	0.0	0.0	3.0	3.0
Merc 450SL	17.3	8.0	275.8	180.0	3.1	3.7	17.6	0.0	0.0	3.0	3.0
Merc 450SLC	15.2	8.0	275.8	180.0	3.1	3.8	18.0	0.0	0.0	3.0	3.0
Cadillac Fleetwood	10.4	8.0	472.0	205.0	2.9	5.3	18.0	0.0	0.0	3.0	4.0
Lincoln Continental	10.4	8.0	460.0	215.0	3.0	5.4	17.8	0.0	0.0	3.0	4.0
Chrysler Imperial	14.7	8.0	440.0	230.0	3.2	5.3	17.4	0.0	0.0	3.0	4.0
Fiat 128	32.4	4.0	78.7	66.0	4.1	2.2	19.5	1.0	1.0	4.0	1.0
Honda Civic	30.4	4.0	75.7	52.0	4.9	1.6	18.5	1.0	1.0	4.0	2.0
Toyota Corolla	33.9	4.0	71.1	65.0	4.2	1.8	19.9	1.0	1.0	4.0	1.0
Toyota Corona	21.5	4.0	120.1	97.0	3.7	2.5	20.0	1.0	0.0	3.0	1.0
Dodge Challenger	15.5	8.0	318.0	150.0	2.8	3.5	16.9	0.0	0.0	3.0	2.0
AMC Javelin	15.2	8.0	304.0	150.0	3.2	3.4	17.3	0.0	0.0	3.0	2.0
Camaro Z28	13.3	8.0	350.0	245.0	3.7	3.8	15.4	0.0	0.0	3.0	4.0
Pontiac Firebird	19.2	8.0	400.0	175.0	3.1	3.8	17.1	0.0	0.0	3.0	2.0
Fiat X1-9	27.3	4.0	79.0	66.0	4.1	1.9	18.9	1.0	1.0	4.0	1.0
Porsche 914-2	26.0	4.0	120.3	91.0	4.4	2.1	16.7	0.0	1.0	5.0	2.0
Lotus Europa	30.4	4.0	95.1	113.0	3.8	1.5	16.9	1.0	1.0	5.0	2.0
Ford Pantera L	15.8	8.0	351.0	264.0	4.2	3.2	14.5	0.0	1.0	5.0	4.0
Ferrari Dino	19.7	6.0	145.0	175.0	3.6	2.8	15.5	0.0	1.0	5.0	6.0
Maserati Bora	15.0	8.0	301.0	335.0	3.5	3.6	14.6	0.0	1.0	5.0	8.0
Volvo 142E	21.4	4.0	121.0	109.0	4.1	2.8	18.6	1.0	1.0	4.0	2.0

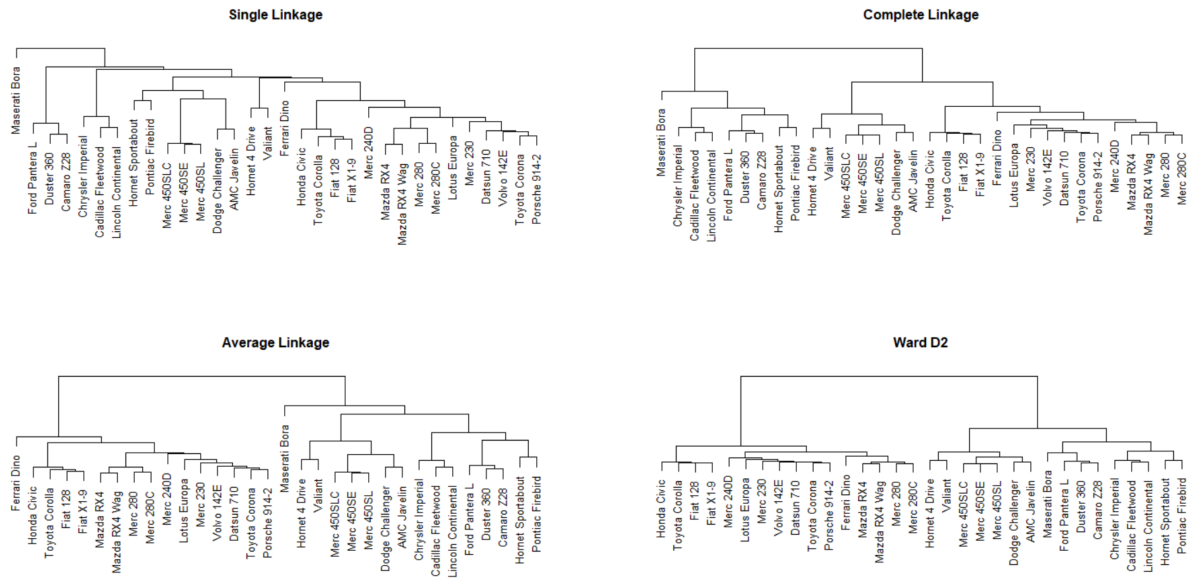


Fig. 3.1: Dendograms displaying differing hierarchical methods. The lower the connection occurs in the dendogram, the earlier these two clusters were combined together. For example, in the Single Linkage method in the top left, the Maserati Bora is linked very last to the rest of the cars.

In Figure 3.1, we can view how each of the four hierarchical methods we have discussed classifies the 32 different cars. While information is lost in dendograms, such as the true proximity of points, it is still informative to determine when certain points, or cars in this case, are combined in the iterative process. The lower the connection occurs on the dendogram, the earlier these two clusters were combined together. For example, we can see with the Single Linkage method in the top left of Figure 3.1 that the Maserati Bora is linked very last. This means that before the final agglomeration to one big cluster, this car was its own cluster, and the 31 other cars formed the other cluster.

When we move to the top right to look at the Complete Linkage method, we can see that the Maserati Bora was combined later in the agglomeration process, but it was linked to a cluster of 8 other cars before the final linkage.

Continuing with the same example, the Average Linkage method shows the Maserati Bora gets linked to a cluster of 15 cars, and then the following and final agglomeration combines two clusters of 16 cars each into one cluster of all 32 cars.

Finally, the Ward D2 method appears to have the most smooth and uniform agglomeration process, where all cars get clustered more evenly as we move through the iterative process. We don't see any later combinations of single cars as we do in the single, complete, or average linkage methods. The Maserati Bora gets linked to a group of just three other cars, rather than being grouped with a very large subset of the entire data set.

For further information about these clustering methods and others, the reader is invited to consider the articles by Borgatti (1994) and Saraçli et al. (2013).

Within R, there exists a `cluster` package that can provide additional help in determining the optimal clustering method between a number of different hierarchical methods, including the four methods listed above (Maechler et al., 2019). One of the functions in this package allows the user to compute an agglomerative clustering coefficient given the method. The details of the calculation are described by Maechler et al. (2019), but will not be discussed in detail in this MS thesis. This clustering coefficient measures the amount of clustering structure found in the data, with values closer to one indicating a stronger structure. When comparing the methods side-by-side, the user can make an educated assumption about the optimal method for clustering the given data set. With the NBA player data, the highest computed coefficient resulted from using the Ward D2 method (see Table 4.1).

3.1.3 Selecting the Optimal Number of Clusters

One potential drawback of hierarchical clustering mentioned previously involves the ambiguous number of clusters needed for further analysis. Agglomerative clustering combines the data from each individual point until all observations are in one cluster. The user has to determine the optimal number of clusters in order to proceed.

Many different measures have been constructed to determine the optimal number of clusters for hierarchical data over the years (Charrad et al., 2014; Martín-Fernández et al., 2020). Many of these measures are related, but all commonly-used methods are calculated in their own unique way. Within the `NbClust` R package (Charrad et al., 2014), 30 different indices can be computed and their choices can be viewed simultaneously to gain a consensus decision of the optimal number of clusters for a given data set (see Section 3.3.7).

In Appendix B, the reader may view the list of all 30 indices used by the `NbClust` R package along with their formula and a brief description. While 30 methods are available, only 26 of these were used for the NBA player data to reduce the computational time. The reasoning for this is based on the comments made by Charrad et al. (2014) in the official `NbClust` article:

"Clustering with index argument set to "alllong" requires more time, as the run of some measures, such as Gamma, Tau, Gap and Gplus, is computationally very expensive, especially when the number of clusters and objects in the data set grows very large. The user can avoid running these four indices by setting the argument index to "all". In this case, only 26 indices are computed."

While it is possible to further limit the types of indices used in the computation, journal articles by Cai et al. (2019), Reimann-Philip et al. (2019), and Sai Krishna et al. (2018) all appear to agree on the use of at least 26 indices in the decision-making process.

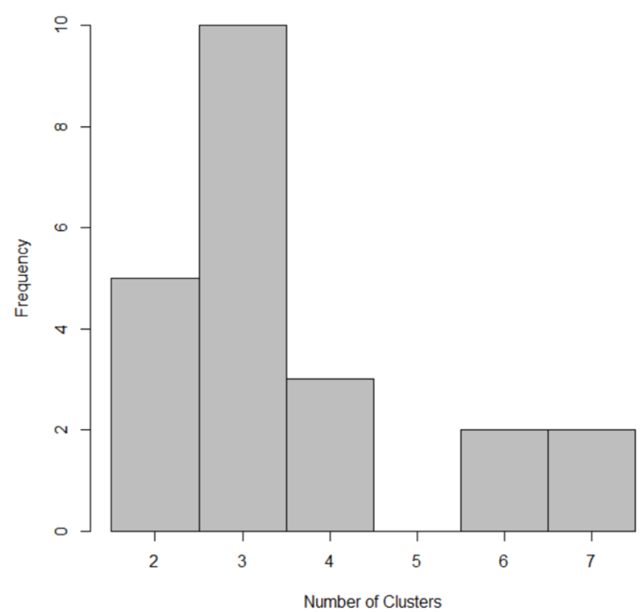


Fig. 3.2: Optimal number of clusters for the `mtcars` data set based on 26 indices. Ten of the 26 indices chose ‘three’ as the optimal cluster number for the cars.

Using the `NbClust` R package on our NBA player data, we can view the optimal number of clusters from year to year based on each criterion, and combine them into one image (Charrad et al., 2014). Figure 3.2 gives an example of the optimal number of clusters as chosen by the 26 criteria for the `mtcars` data set from base R. In this example, 10 of the 26 indices chose *three* as the optimal number of clusters for the `mtcars` data.

3.2 Other Methods

Additional methods are used in this research to either verify or enhance the player data analysis. It is important to verify that our clustering results are both meaningful and consistent. We will begin by discussing the usefulness of the within sum of squares and Adjusted Rand Index calculations, followed by a discussion of dimensionality reduction methods, including PCA, tSNE, and GGobi's grand tour feature.

3.2.1 Within Sum of Squares

A useful method to determine the optimal cluster number involves taking a sum of squares measure within each cluster, known as the within sum of squares (WSS). A sum of squares measure is computed by measuring the distance between each data point and the mean, or in this case, the centroid. We can compute the WSS as we increase or decrease the number of partitions to see how the number of partitions affects the overall clustering variation. The WSS will decrease as we add more partitions to the data since the data points will become increasingly closer to the center of their cluster. Generally, this decrease in WSS will begin very rapidly as we increase the number of clusters, and will level off as we get closer to each data point as its own cluster, or a WSS value of 0.

Figure 3.3 provides an example WSS plot (Galili, 2013) using the `mtcars` data set from base R introduced in Section 3.1.2. We can see that the WSS appears to level off after the third separation, so a possible cluster number for the mtcars data would be *three*. We can also see another drop-off between the sixth and seventh cluster separation that is steeper relative to the previous three separations. This line of logic can help us make a more meaningful decision regarding the best number of clusters.

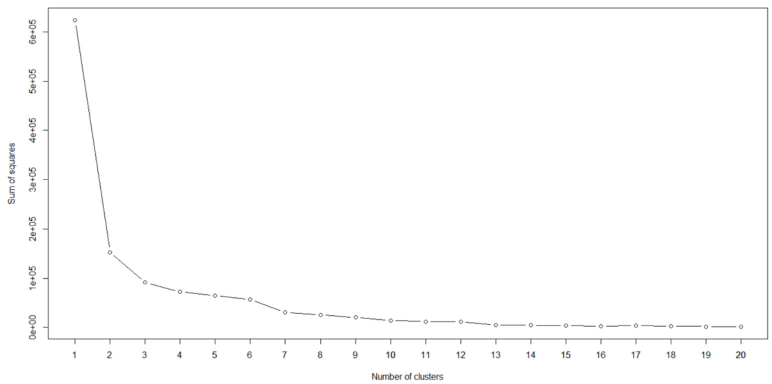


Fig. 3.3: Example within sum of squares plot by cluster number - `mtcars` data set. We can see a significant leveling off, or 'elbow' in the plot around three clusters, making this a reasonable selection.

3.2.2 Adjusted Rand Index

Since we are looking at NBA player data from 2000 to 2020, it became important to verify that the optimal number of clusters from year to year are consistent. Due to the continuing evolution of the game of basketball, and especially with a heavier emphasis on drafting and starting players who can perform a number of different roles on the court rather than specializing, we must verify that it is logical to use the same number of clusters (in this case, nine clusters were selected for the combined data) in each year, or if there has been an evolution in player roles.

While it would be interesting to carry out a more in-depth exploration of the evolution of player usage over a 20+ year span, the pertinent question to this cluster analysis was simply whether or not the clustering algorithm is relatively consistent from year to year.

The Rand Index (RI) provides a similarity measure between two different data clusterings (Rand, 1971). To calculate the Rand Index, we must count up all the 'agreements' between two data clusterings and then divide the resulting number by the total number of unordered pairs. Equation 3.1 displays the full calculation, where a refers to the number of unordered pairs placed in the same cluster in both clustering methods, and b refers to the number of unordered pairs placed in different clusters in both clustering methods. We can count up the total number of unordered pairs by computing $\binom{n}{2} = n(n-1)/2$.

$$RI = \frac{a+b}{\binom{n}{2}} \tag{3.1}$$

Suppose we have a set of 6 elements {A,B,C,D,E,F}, and we cluster them into one of three groups using two different clustering methods. The first clustering, called X, is shown below, where A was clustered into group 1, B into group 2, and so on:

$$1 \quad 2 \quad 3 \quad 1 \quad 2 \quad 3$$

The second clustering we will call Y and is shown below:

$$1 \quad 1 \quad 3 \quad 1 \quad 1 \quad 2$$

Due to the small set of six elements, we only have $\binom{6}{2} = 6(6-1)/2 = 15$ unordered pairs, and we can list them out as follows: {A,B}, {A,C}, {A,D}, {A,E}, {A,F}, {B,C}, {B,D}, {B,E}, {B,F}, {C,D}, {C,E}, {C,F}, {D,E}, {D,F}, {E,F}.

We can calculate a from 3.1 by finding the total number of unordered pairs that are placed in the same cluster in both X and Y. In this case, {A,D} and {B,E} qualify, since A and D are both placed in cluster 1 in X, and A and D are placed in cluster 1 in Y. Similarly, B and E are both placed in cluster 2 in X, and B and E are both placed in cluster 1 in Y. This means that $a = 2$. Next, we can calculate b by finding the total number of unordered pairs that are placed in different clusters in both X and Y. In this case, we have {A,C}, {A,F}, {B,C}, {B,F}, {C,D}, {C,E}, {D,F}, {E,F}, so $b = 8$. When we plug these values into the Rand Index formula, we get: $RI = (2+8)/15 = 10/15 = 0.667$. Notice that 5 pairs were not included in the numerator of the RI calculation. This is because these pairs are clustered into different groups in X and the same group in Y, or vice versa. For example, the ordered pair {A,B} is placed into different clusters (1 and 2) in X, but {A,B} are placed in the same cluster in Y (1 and 1).

Intuitively, the Rand Index is valued between 0 and 1, where a 0 would denote no agreement between the two clusterings, and a 1 indicating that the two data clusterings are exactly the same.

The Adjusted Rand Index (ARI) further adds on the Rand Index by accounting for chance in clustering (Hubert and Arabie, 1985). The random chance is based off a contingency table created between matches of X and Y, where N_{ij} refers to the number of objects in common between X and Y and i and j corresponding to a row and column for each object, respectively. More simply put, the Adjusted Rand Index accounts for chance by computing the Rand Index, the expected value of the Rand Index, and the maximum of the Rand Index. Equation 3.2 illustrates how these computations are implemented.

$$ARI = \frac{RI - Expected(RI)}{Max(RI) - Expected(RI)} \tag{3.2}$$

To illustrate an example of how the Adjusted Rand Index is calculated, we will use the same data and partitions as the previous example. Table 3.2 shows the contingency table for our clustering methods X and Y, where N_{ij} corresponds to the number of times an element is clustered into group i in X and group j in Y. For example, the third row and second column contains a value of 1, since F is the only object placed into cluster 3 in X and cluster 2 in Y.

Table 3.2: Contingency table displaying N_{ij} for clustering methods X and Y

	Y1	Y2	Y3	Row Sums
X1	2	0	0	2
X2	2	0	0	2
X3	0	1	1	2
Col Sums	4	1	1	N=6

Equation 3.3 shows the full calculation for the Adjusted Rand Index, where i refers to the row number, j refers to the column number, and P and $P*$ refer to two different partitions of the same data.

$$ARI(P, P*) = \frac{\sum_{ij}\binom{N_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{N}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{N}{2}} \tag{3.3}$$

We will start by calculating $\sum_{ij}\binom{N_{ij}}{2} = \binom{2}{2} + \binom{0}{2} + \binom{0}{2} + \binom{2}{2} + \binom{0}{2} + \binom{0}{2} + \binom{0}{2} + \binom{1}{2} + \binom{1}{2} = 1 + 0 + 0 + 1 + 0 + 0 + 0 + 0 + 0 = 2$. Since a_i corresponds to the row sums, we can calculate

$\sum_i \binom{a_i}{2} = \binom{2}{2} + \binom{2}{2} + \binom{2}{2} = 1 + 1 + 1 = 3$. Finally, since b_j corresponds to the column sums, we can calculate $\sum_j \binom{b_j}{2} = \binom{4}{2} + \binom{1}{2} + \binom{1}{2} = 6 + 0 + 0 = 6$. Plugging these values into Equation 3.3, we obtain the following:

$$ARI = \frac{2 - [3 * 6]/\binom{6}{2}}{\frac{1}{2}[3 + 6] - [3 * 6]/\binom{6}{2}} = \frac{2 - 1.2}{4.5 - 1.2} = \frac{0.8}{3.3} = 0.242$$

As in the case of the Rand Index, the output of the Adjusted Rand Index will tend towards 1 as the two data clusterings become more similar, and will tend towards 0 for two clusterings that highly disagree.

For the purposes of this research, we want to verify that there is consistency from year to year in the way the hierarchical clustering algorithm partitions the NBA players. Obviously, players who change teams, roles, or positions, or players who simply develop and enhance their skills in the offseason will have a high chance of being placed in a different cluster, but this is to be expected. We simply want to verify that the clustering is worth more than random partitioning.

The Adjusted Rand Index was calculated comparing each season to the season immediately preceding it and immediately following it. Rows were removed from the two seasons in question if the player did not participate in both seasons. The total number of 19 different ARI measures were taken starting with the 2000-2001 season compared to the 2001-2002 season, and ending with the 2018-2019 season compared to the 2019-2020 season. The results were then compared to a random baseline where the same players were randomly assigned to one of nine clusters and simulated 9,999 times to compare the ARI results to our achieved outcomes.

3.2.3 Principal Component Analysis

Principal Component Analysis (PCA) endeavors to reduce data down to a few principal components in order to more easily visualize it in two-dimensional space (Pearson, 1901). The first principal component is the one that maximizes the variance of the projection of data. While PCA was used only as a baseline for more sophisticated dimensionality reduction methods, it is important to set the stage with the most frequently used and understood method available.

For the purposes of this research, PCA was used as a dimensionality reduction method for exploratory data analysis. This method allows us to display the nine NBA player clusters in two dimensional space while still being able to view some separation between the clusters. Jolliffe (1986) provided further information on the calculation, history, and scope of PCA.

3.2.4 tSNE

As an alternative to PCA, t-Distributed Stochastic Neighbor Embedding (tSNE) offers a more robust technique to visualize high-dimensional data in two-dimensional space. While PCA is the more frequently used method, tSNE is a more advanced technique that can analyze much more complicated data sets (van der Maaten and Hinton, 2008).

While PCA provides a linear dimensionality reduction, i.e., placing dissimilar points farther away from each other in the two-dimensional plane, tSNE can evaluate non-linear and non-parametric relationships to provide a better two-dimensional interpretation of the data. The details of the tSNE implementation will not be presented in this research, but the reader may consult the work of Hinton and Roweis (2002) for further information about tSNE and its uses.

3.2.5 PHATE

Another dimensionality reduction method known as Potential for Heat-diffusion Affinity-based Trajectory Embedding (PHATE) (Moon et al., 2019) was employed in this MS thesis using Python. Like tSNE, PHATE is capable of handling non-linear data, as well as data sets with lots of noise. Examples of the output in two and three dimensions were compared to the other representations of the NBA player data to show how the nine player clusters can be distinguished using different methods and projections.

The details of the PHATE implementation and logic will not be presented in this MS thesis, but the official documentation, introductory code, and examples provided by

Moon et al. (2019) are available on Github via the following link: `https://github.com/KrishnaswamyLab/PHATE`.

3.3 R Packages

In this section, we will explore briefly the different R (R Core Team, 2021) packages used for this research. While we will not describe all functionalities of the various R packages, helpful links to documentation and examples will be given for further study.

3.3.1 tidyverse

The `tidyverse` is actually a collection of data manipulation R packages (Wickham et al., 2019). Loading `tidyverse` allows the user to access many functions and tools, including those found within the `rvest`, `purrr`, and `dplyr` R packages described in more detail in the next sections.

3.3.2 rvest

The `rvest` R package provides a simple and compact way to scrape data from the web (Wickham, 2020b). `rvest` is found within the `tidyverse` R package, therefore it can be loaded either by loading `tidyverse` or by installing and loading `rvest` directly.

In this research, `rvest` was used to extract all lineup tables by referencing the specific HTML nodes (kjytay, 2018). For further information about rvest, please visit the following help page: `https://rvest.tidyverse.org/`

3.3.3 purrr

The `purrr` R package is a data manipulation tool that enhances R programming by providing tools to work with vectors and functions (Henry and Wickham, 2020). The `purrr` R package can be installed and loaded directly, or can be loaded by simply loading the `tidyverse` R package.

The `map_dbl` function was used in this research to synthesize and display the results of the `agnes` function from the `cluster` R package (see Section 3.3.8). The `map_dbl` function, as well as the `map_chr`, `map_lgl`, and `map_int` functions return an atomic vector of the indicated type. Boehmke (2020) provides a complete example using the `agnes` R function as well as the `map_dbl` function for displaying the results of the different algorithms.

The reader is invited to visit the `purrr` help page found at the following link: https://purrr.tidyverse.org/

3.3.4 dplyr

The `dplyr` R package is a data storage, manipulation, and transformation tool (Wickham et al., 2020). The functions associated with `dplyr` allow the user to more compactly organize and summarize data by using fewer steps than base R.

Various functions from `dplyr` were used throughout the data preparation, including using the piping process to web scrape, subset, re-order, and add new columns to the data. For more information on the wide array of uses of the `dplyr` R package, please visit the following source: https://dplyr.tidyverse.org/

3.3.5 XML

The `XML` R package provides many helpful tools for parsing, generating, and reading XML and HTML documents through R (Temple Lang, 2020).

In this research, the `XML` R package was used to scrape all individual player tables from the year 2000 to the year 2020 (Frey, 2019). For further information about `XML`, please visit the help page: https://cran.r-project.org/web/packages/XML/XML.pdf

3.3.6 httr

The `httr` R package is designed to work with the most frequent HTTP verbs, like GET(), POST(), HEAD(), etc. The package is designed to allow the user to easily access content such as status codes and cookies (Wickham, 2020a).

The `GET()` function in R was used to access the URLs and parse specific information from the HTML tables. For further information on the usage of the `httr` R package, please visit the help page: https://cran.r-project.org/web/packages/httr/httr.pdf

3.3.7 NbClust

The `NbClust` R package was briefly discussed in Section 3.1.3. The 30 different indices found in Appendix B provide the user with a proposal of the optimal number of clusters. The user can select all or any subset of the indices in the selection process (Charrad et al., 2014).

The indices mentioned were used to determine the optimal number of clusters for the player data by year, and then aggregated to determine the optimal number of clusters for all 20 years combined. Histograms were used to display the frequency of selections for each number between as low as two clusters and as high as 20 clusters. It was determined that the start and end points always show an increase in number of selections over their neighbors. See Appendix C for work and visuals related to the analysis of the effect of different start and end points on the optimal cluster selection.

The reader is also invited to view the help page for the `NbClust` R package found here: https://cran.r-project.org/web/packages/NbClust/NbClust.pdf

3.3.8 cluster

The `cluster` R package provides an array of clustering algorithims and tools for analyzing and plotting clustering results (Maechler et al., 2019).

For this analysis, the `agnes` function was used to compute the agglomerative nesting coefficient. This measures the amount of clustering structure found, with values closer to 1 indicating a stronger structure. Different types of hierarchical clustering algorithms (including "ward", "ward D2", "single", "complete", and "average") were computed and the coefficients were compared to determine the most useful method. See the following help page for additional information and usage for the `cluster` package as well as many helpful examples: https://cran.r-project.org/web/packages/cluster/cluster.pdf

3.3.9 factoextra

The `factoextra` R package provides an efficient and effective way to extract and visualize multivariate data using methods such as PCA, Correspondence Analysis (CA), Multiple

Correspondence Analysis (MCA), and others (Kassambara and Mundt, 2020).

The `fviz_cluster` R function in `factoextra` was used to display the results of the nine clusters by year, as well as the *mega-clustering* analysis results. This was used as a base visualization for comparison to more advanced methods such as tSNE. For further information on the functionalities of `factoextra`, please visit the following source: https://cran.r-project.org/web/packages/factoextra/factoextra.pdf

3.3.10 mclust

The `mclust` R package provides "Gaussian mixture modeling for model-based clustering, classification, and density estimation" (Scrucca et al., 2016). Within the `mclust` library there exists an `AdjustedRandIndex` function for comparing two classifications.

In this research, the `AdjustedRandIndex` function was used to verify similarities between data clusterings from season to season. This process is described in Section 3.2.2. For further information on the use cases of the `AdjustedRandIndex` R function, as well as the other functions within the `mclust` R package, the reader is invited to visit the following source: https://cran.r-project.org/web/packages/mclust/mclust.pdf

3.3.11 Rtsne

The `Rtsne` R package allows the user to implement the tSNE dimensionality reduction method (Krijthe, 2015). This method was discussed in Section 3.2.4. The function takes as input a matrix where the rows are observations and the columns are variables or dimensions.

In this research, the different player statistics constituted the columns and each individual player in a given season made up a row. The new values resulting from the `Rtsne` procedure were then displayed using baseR to show the separations of the nine player clusters. For further information on the `Rtsne` R package, the reader is invited to visit the help page: https://cran.r-project.org/web/packages/Rtsne/Rtsne.pdf

3.4 Python Packages

This section provides a brief explanation of the various Python (Van Rossum and Drake,

2009) packages that were used in this MS thesis. The use of Python provided additional methods to explore and visualize our NBA player data and clustering results.

3.4.1 pandas

The `pandas` Python package is an open source data manipulation and analysis tool (McKinney, 2010). This package allows the user to easily perform functions including the following: reading in data, adding rows and columns to a data frame, slicing data, and merging and reshaping data frames.

In this MS thesis, the `pandas` Python package was used to read in the NBA player .csv files. `pandas` was also used to further prepare and modify the data to be run through the PHATE modeling process. For further information about the uses of `pandas`, the reader is invited to consult the following web page: https://pandas.pydata.org/docs/getting_started/index.html#getting-started

3.4.2 matplotlib

The `matplotlib` Python package contains a wide variety of plotting functions, from static graphs to dynamic and interactive visualizations (Hunter, 2007). This package works well with data in many different formats, including the resulting objects from the `phate` procedure.

The `matplotlib` Python package was used in this research to visualize the NBA player clusters using PHATE. Two-dimensional scatter plots were created and displayed in a single image using the `subplot` function within `matplotlib`. For further information on the applications of `matplotlib`, please visit the following user guide: https://matplotlib.org/stable/users/index.html.

3.4.3 scprep

The `scprep` Python package is a framework for loading, preprocessing, and plotting matrices (https://github.com/KrishnaswamyLab/scprep). The `scprep` package allows the user to work with many of the open-source Python packages, including `scipy` (https://www.scipy.org/), `pandas`, `numpy` (https://numpy.org/), and `matplotlib`.

For the purposes of this research, the `scatter2d` function within the `scprep` Python package was used to display the results of the PHATE procedure. This package was used in conjunction with the `matplotlib` package to create the PHATE visualizations. The following URL may be consulted for further examples of scatterplots using `scprep`: https://scprep.readthedocs.io/en/stable/examples/scatter.html

3.4.4 phate

The `phate` Python package (Moon et al., 2019) was created to implement the PHATE procedure, as discussed in Section 3.2.5. Results from this alternative dimensionality reduction method can then be displayed using a variety of plotting packages, including the `matplotlib` family of plotting functions mentioned in Section 3.4.2.

In this research, the `phate` Python package was used to provide an additional visualization for the NBA player cluster data to compare to PCA and tSNE. The reader may learn more about the `phate` package and see examples at the following link: https://dburkhardt.github.io/tutorial/visualizing_phate/

3.5 GGobi

GGobi is an interactive platform constructed to provide insights into multi-dimensional data (Cook and Swayne, 2007). From the official GGobi website found at http://ggobi.org/, the description reads as follows: "GGobi is an open source visualization program for exploring high-dimensional data. It provides highly dynamic and interactive graphics such as tours, as well as familiar graphics such as the scatterplot, barchart and parallel coordinates plots. Plots are interactive and linked with brushing and identification."

GGobi was used in this research to explore and validate the nine player clusters using all their individual statistics. The grand tour feature was used extensively to visualize the nine NBA player clusters for each year, as well as to analyze and brush the nine player clusters across all 20 NBA seasons. The grand tour is a procedure that allows the user to view scatterplots from all possible projections for high-dimensional data (Asimov, 1985; Cook et al., 1995). The user may pause at any point and 'brush' clusters of points to watch how they behave across different projections.

Lee et al. (2020) provided an in-depth comparison between grand tours and other 'embedding' reduction methods, including tSNE. The authors of this research introduce the `liminal` R package as a link between tours and embedding methods in order to bridge the gaps in understanding between the two approaches (Lee, 2021).

Examples of the grand tour and 'brushing' features will be shown and discussed in the next chapter.

CHAPTER 4

Selecting the Optimal Method & Number of Clusters

In this chapter, we will go further into detail on the selection and justification of the best clustering method and the optimal number of clusters for each of the 20 NBA seasons. This chapter will cover the application of the `NbClust` R package discussed in Section 3.3.7, as well as the various visualizations used to explore the clusters, including histograms, scatter plots, and snapshots of the grand tour feature of GGobi.

4.1 Selecting a Clustering Method

Before we can decide on the optimal number of clusters for the NBA players, we need to decide which algorithm will perform the best in making meaningful separations. As previously mentioned, a hierarchical approach was ultimately chosen in place of a k-means algorithm, largely due to its interpretability (see Section 3.1.1).

Many hierarchical methods exist, and the most frequently encountered methods were cross-analyzed using the `agnes` function in the `cluster` R package (Maechler et al., 2019) (see Section 3.3.8). For each NBA season, the AGNES coefficient was computed to measure the amount of clustering structure, where higher coefficients (closer to 1) indicate a stronger clustering structure. Table 4.1 displays the average AGNES coefficient for the major hierarchical clustering methods. Ward's (Distance-Squared) method showed the highest amount of structure at 0.959, therefore the decision was made to proceed with this method for the rest of the analysis (see Section 3.1.2).

Table 4.1: AGNES coefficient comparison between different hierarchical methods. Ward's Distance-Squared method shows the highest amount of clustering structure at 0.959.

	Average	Single	Complete	Ward (Dist-Squared)	Weighted
Coefficient	0.793	0.716	0.863	0.959	0.818

4.2 Application of NbClust

Selecting the optimal number of clusters comprises a large portion of this research. The task of assigning the appropriate number of clusters, especially in hierarchical clustering methods, can be somewhat subjective. With this in mind, we will attempt, through a wide array of visualizations and calculations, to justify the selection of nine clusters for any given year of player data, as well as for the combined clustering over a 20 year span of NBA players.

The `NbClust` R package was introduced in Section 3.3.7 as an effective way to select the optimal number of clusters by using 26 different indices and tallying their 'votes'. We can use these index results to see which cluster amount tends to get selected the most and to see the trends as we stretch from two partitions all the way to twenty different partitions.

4.2.1 Determining Start/End Points

The `NbClust` function in R contains arguments for start and end points. Initially, a starting value of five was chosen since the sports community already categorizes players into one of five positions. However, we must also consider the possibility that we may be able to cluster players into *fewer* than five positions.

Starting points from two to five were chosen, and end points from fifteen to twenty to see how the decisions by the indices would be affected. In Appendix B, the reader may view a more in-depth discussion of how varying the start and end points affects the resulting decisions for best cluster number. Three clusters are overwhelmingly chosen as the optimal cluster number. This cluster configuration tends to separate high scoring forwards and traditional 'big men' from the rest of the players. Additional details and commentary on the three-cluster configuration can be found in Appendix E. With this in mind, one of the primary motives of this research is to describe players in more detail and explore subtle differences between players through visualization. For this reason, we focus on cluster sizes of five or more for the remainder of this thesis.

Ultimately it was determined that varying start and end points did not influence our final decision for the optimal number of clusters.

4.2.2 NbClust Results

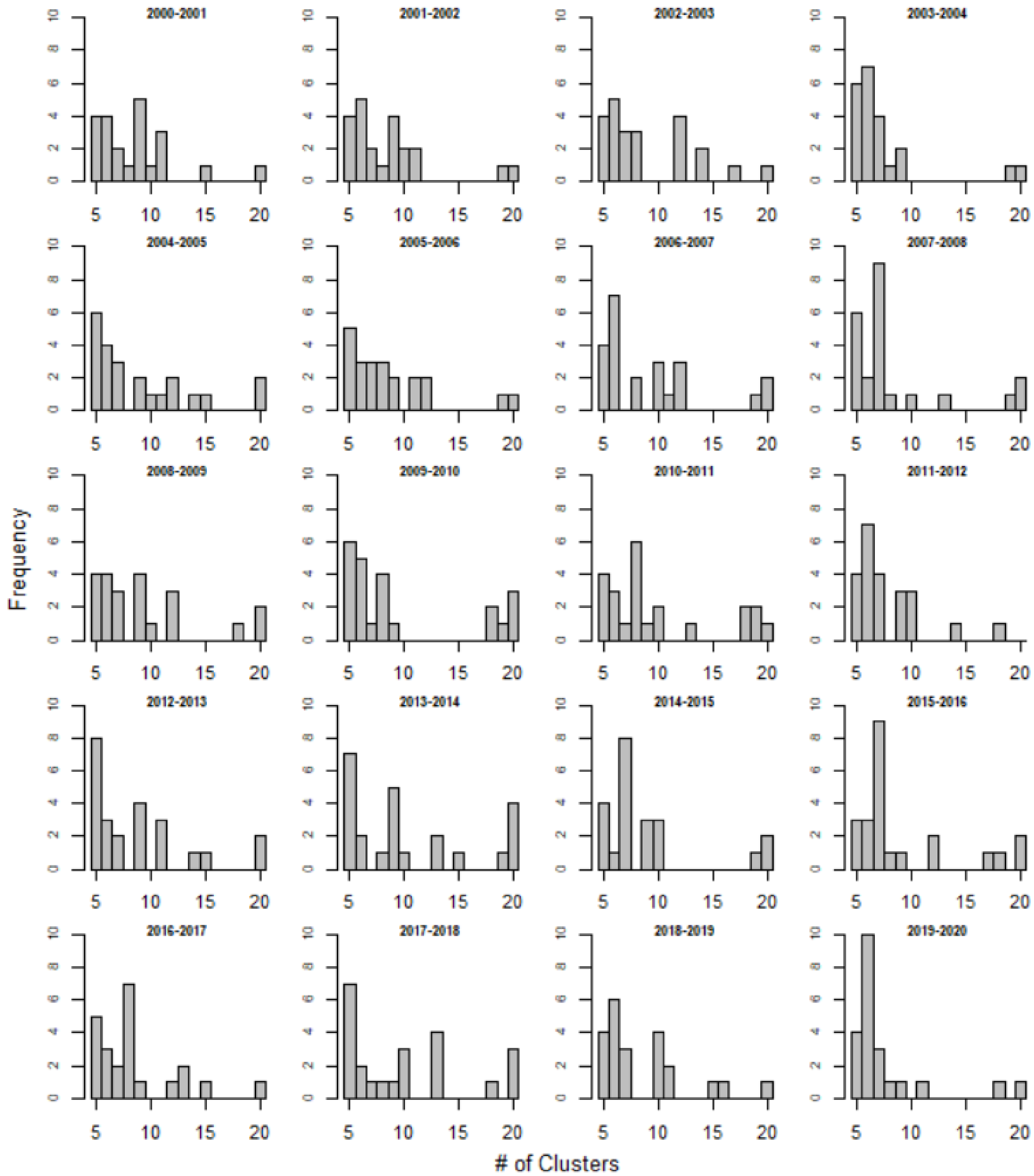


Fig. 4.1: Optimal number of clusters per year for NBA player data based on 26 indices by season. Most years' cluster selections decrease from 5 to 15 clusters, followed by an increase in selections from 15 to 20 clusters. Individual seasons such as the 2000-2001 season and the 2008-2009 season show nine clusters as the optimal selection.

We can individually assess the optimal cluster selections for each NBA season in question. The histogram matrix in Figure 4.1 displays the optimal number of clusters chosen by the 26 indices from the 2000-2001 season to the 2019-2020 season. Most histograms are skewed to the right with many showing local maximums at around eight or nine clusters.

In Figure 4.2, we can view the results of the `NbClust` analysis for all years combined. If we look beyond the highest frequency at six clusters, we can see a strong local maximum at nine clusters, as well as a local maximum at twelve, fifteen, and twenty clusters.

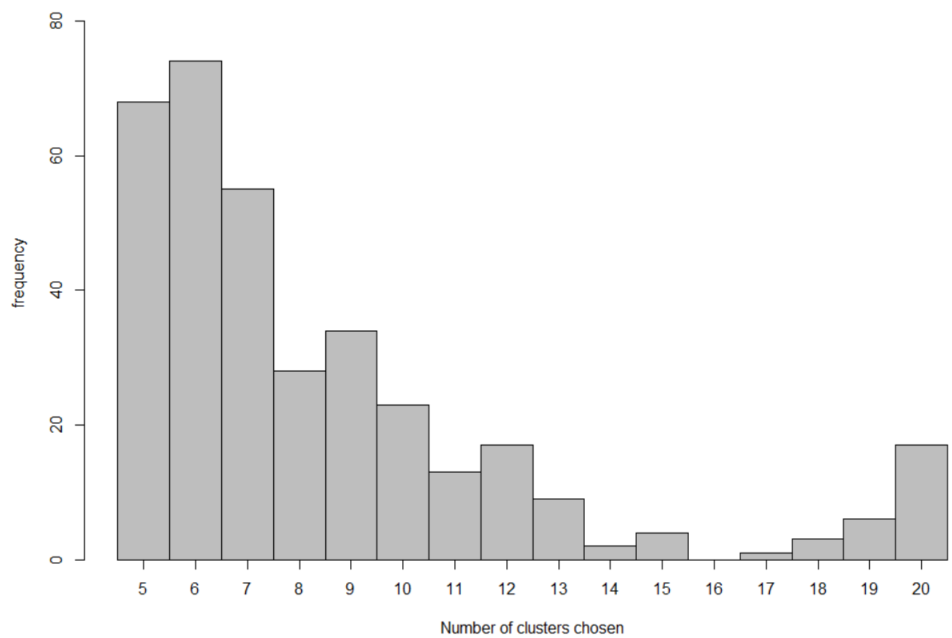


Fig. 4.2: Optimal number of clusters for NBA player data based on 26 indices from the 2000-2001 season to the 2019-2020 season using Ward D2. We see the most frequently selected cluster number is six, with local maximums occurring at nine, twelve, fifteen, and twenty clusters.

4.3 Clusterplots/Dimensionality Reduction

We will justify the choice of nine clusters through various visualizations and dimensionality reduction methods. In this section, we will view the results of PCA (see Section 3.2.3), tSNE (see Section 3.2.4), and PHATE (see Section 3.2.5). Examples of each will be given, as well as several side-by-side comparisons of the methods. Please be advised that the cluster numbers are not consistent from season to season. For example, Cluster 1 from the 2000-2001 NBA season PCA clusters will likely not be the same player position as Cluster 1 from the 2001-2002 PCA clusters. This same inconsistency will also apply to the visualizations using the other dimensionality reduction methods from season to season. One must look in to the underlying data points to determine the mapping of clusters from one year to the next.

4.3.1 PCA

We will visualize the results of PCA partitioning through base R and through the `factoextra` R package (see Section 3.3.9). The results of PCA in base R by year for each NBA season available are shown in Figure 4.3.

We can see from Figure 4.3 that for each year some clusters have clear separations, while others appear to have considerable overlap. This does not mean that the overlapping clusters are not distinct. This likely means that the visualization does not capture the correct dimensions to accurately depict the distinction.

We can take a closer look at the first season's PCA clustering in Figure 4.4. We can see that Cluster 9 in the top right of the scatter plot has clear separation from the rest of the data. We can also see that Cluster 7 on the bottom middle of the scatter plot shows very little overlap with the rest of the data. Other clusters like 1, 3 and 5 show compactness, but the limitations of two dimensions make it difficult to determine clear separation.

We can view a similar PCA output using the `factoextra` R package. Figure 4.5 shows the same 2000-2001 NBA season's players. We can see that this figure is the mirrored version of Figure 4.4 with the well-separated cluster appearing in the top left as opposed to the top right. This additional plot draws a shape around the clusters and also labels the center of each cluster with a different symbol. The `factoextra` plot gives the appearances of potentially unique 'planes' on which the points may appear in higher dimensions.

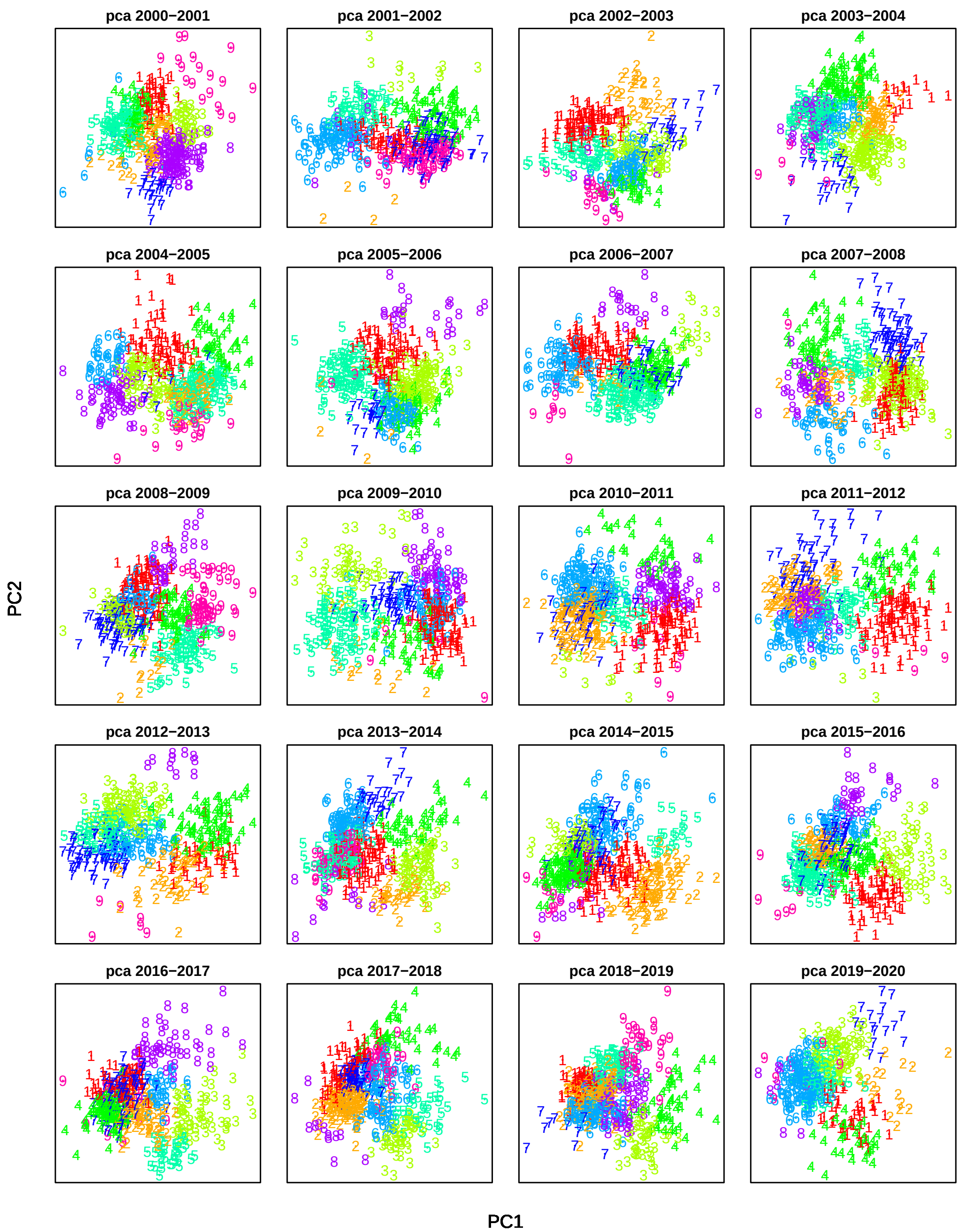


Fig. 4.3: PCA plots for NBA seasons 2000-2001 to 2019-2020 - separated into nine clusters. Note that the cluster numbers are not consistent from season to season. For example, Cluster 9 in the 2000-2001 season corresponds to the `Superstar` players, while in the 2001-2002 season, the `Superstar` players correspond to Cluster 3.

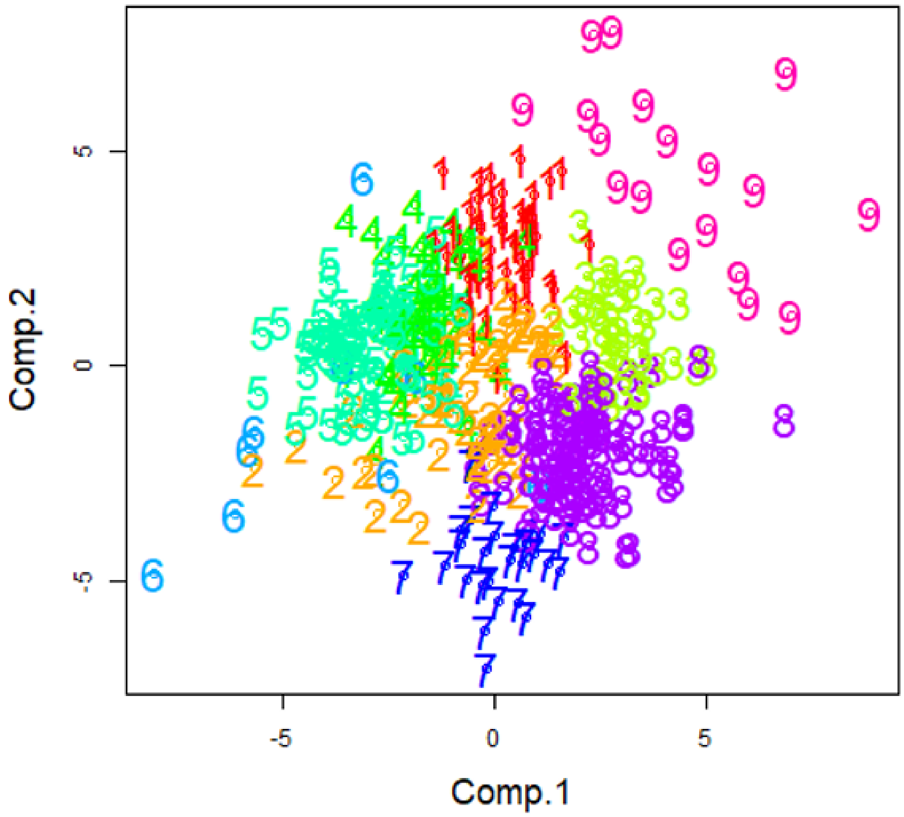


Fig. 4.4: PCA plot using base R for players in the 2000-2001 NBA season - separated into nine clusters. While this technique does not display all player clusters as being highly distinct, we can see certain clusters that show relative separation. We can see that Cluster 9 in the top right of the scatter plot has clear separation from the rest of the data.

Figure 4.5 shows that the first principal component ('Dim1' on the x-axis) accounts for 29.8% of the total variation, while the second principal component ('Dim2' on the y-axis) accounts for 26.6%. Between these first two principal components, we have only accounted for roughly 56% of the total variation in the player clusters. This further illustrates the need for more advanced dimensionality reduction techniques to visualize the cluster separations.

The reader may also consult Appendix E for a brief discussion and visualization of only three player clusters instead of nine.

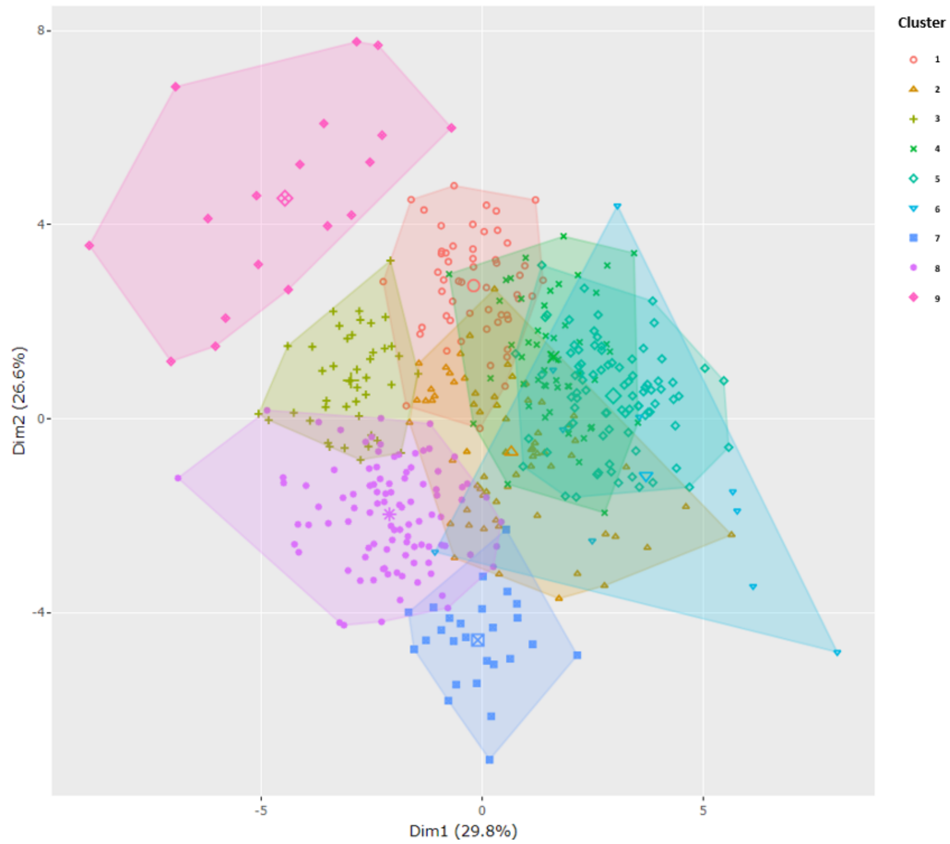


Fig. 4.5: PCA plot using `factoextra` R package for players in the 2000-2001 NBA season - separated into nine clusters. While this technique does not display all player clusters as being highly distinct, we can see certain clusters that show relative separation. We can see that Cluster 9 in the top left of the scatter plot has clear separation from the rest of the data.

4.3.2 tSNE

The tSNE method is discussed in Section 3.2.4. Figure 4.6 gives a two-dimensional representation of each NBA season using tSNE. We can compare these results to the same data using PCA found in Figure 4.3. In general, there tends to be greater distinctions and spacing between clusters using tSNE. This outcome was to be expected based on the robustness of tSNE with more complex and high-dimensional data.

Figure 4.7 displays a close-up view of the 2000-2001 season using tSNE, while Figure 4.8 shows a side-by-side comparison of tSNE and PCA. When compared with PCA, we can see considerably fewer overlaps between clusters. Clusters 1, 3, 4, 5, and 7 show almost no

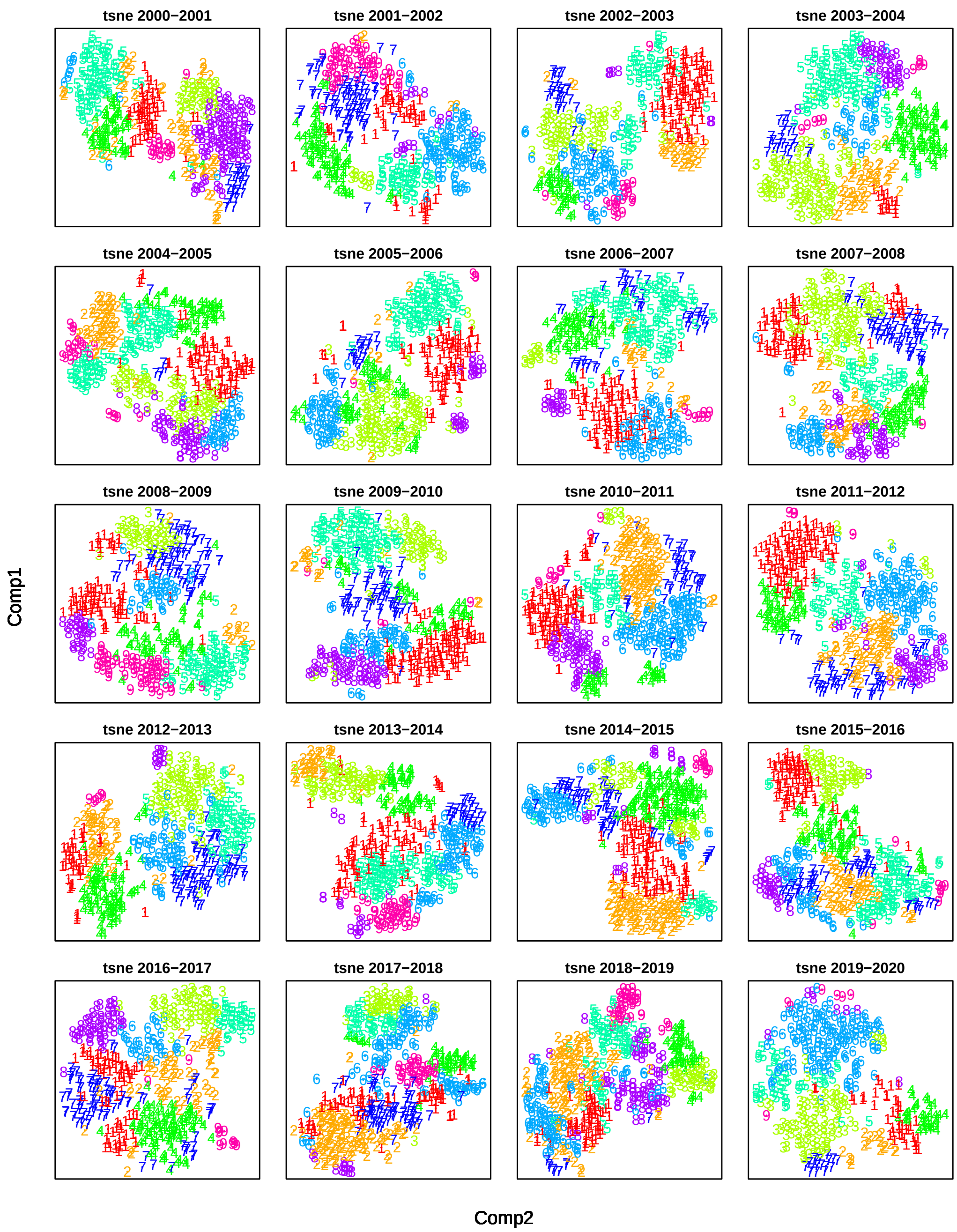


Fig. 4.6: tSNE plots for NBA seasons 2000-2001 to 2019-2020 - separated into nine clusters. Please note that cluster numbers are not consistent from season to season. For example, Cluster 9 in the 2000-2001 season corresponds to the `Superstar` players, while in the 2001-2002 season, Cluster 3 corresponds to the `Superstar` players.

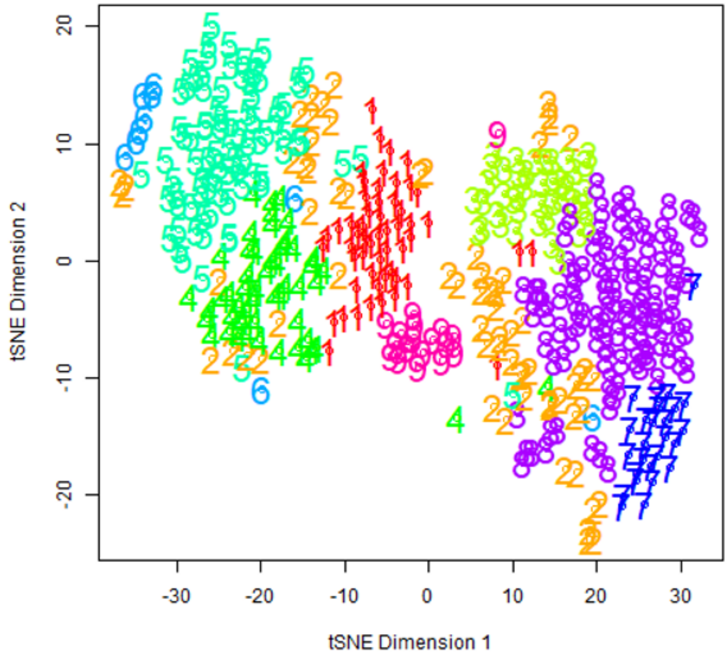


Fig. 4.7: tSNE plot for players in the 2000-2001 NBA season - separated into nine clusters.

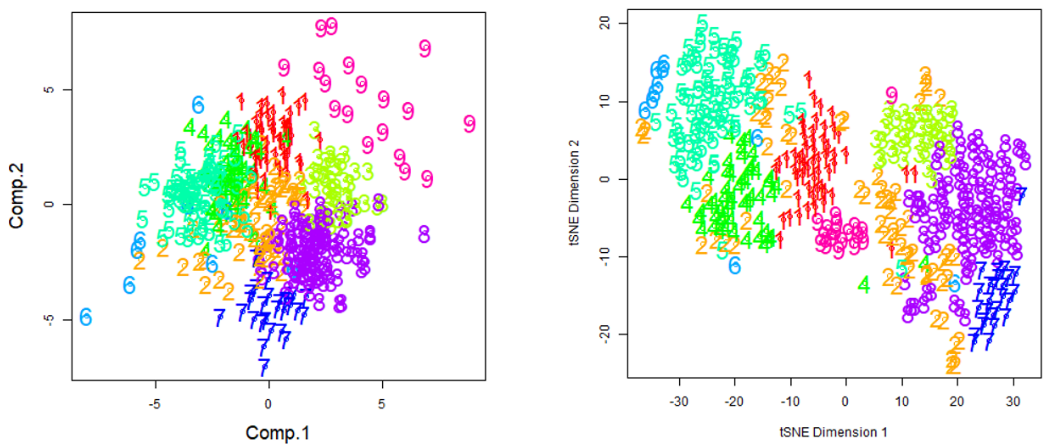


Fig. 4.8: Visualizing the 2000-2001 NBA season clusters using PCA (left) and tSNE (right) - separated into nine clusters. In general, tSNE does a better job of showing the distinction between clusters than PCA. We can see that most clusters in the tSNE plot, with the exception of Clusters 2 and 4, show relatively strong distinction from the rest of the data.

overlap with other data points. Clusters 6, 8, and 9 still show high distinction from the other data points, but may have just a few points which appear misplaced in this limited two-dimensional view. Cluster 2 is the only cluster that appears very spread out across the entire plot. We will see in Section 5.2.1 that this cluster corresponds to the **Bench Role Players** position, which is a smaller cluster with more miscellaneous players. This is overall an encouraging sight as we attempt to justify the meaningfulness of using nine partitions for the NBA player data.

4.3.3 PHATE

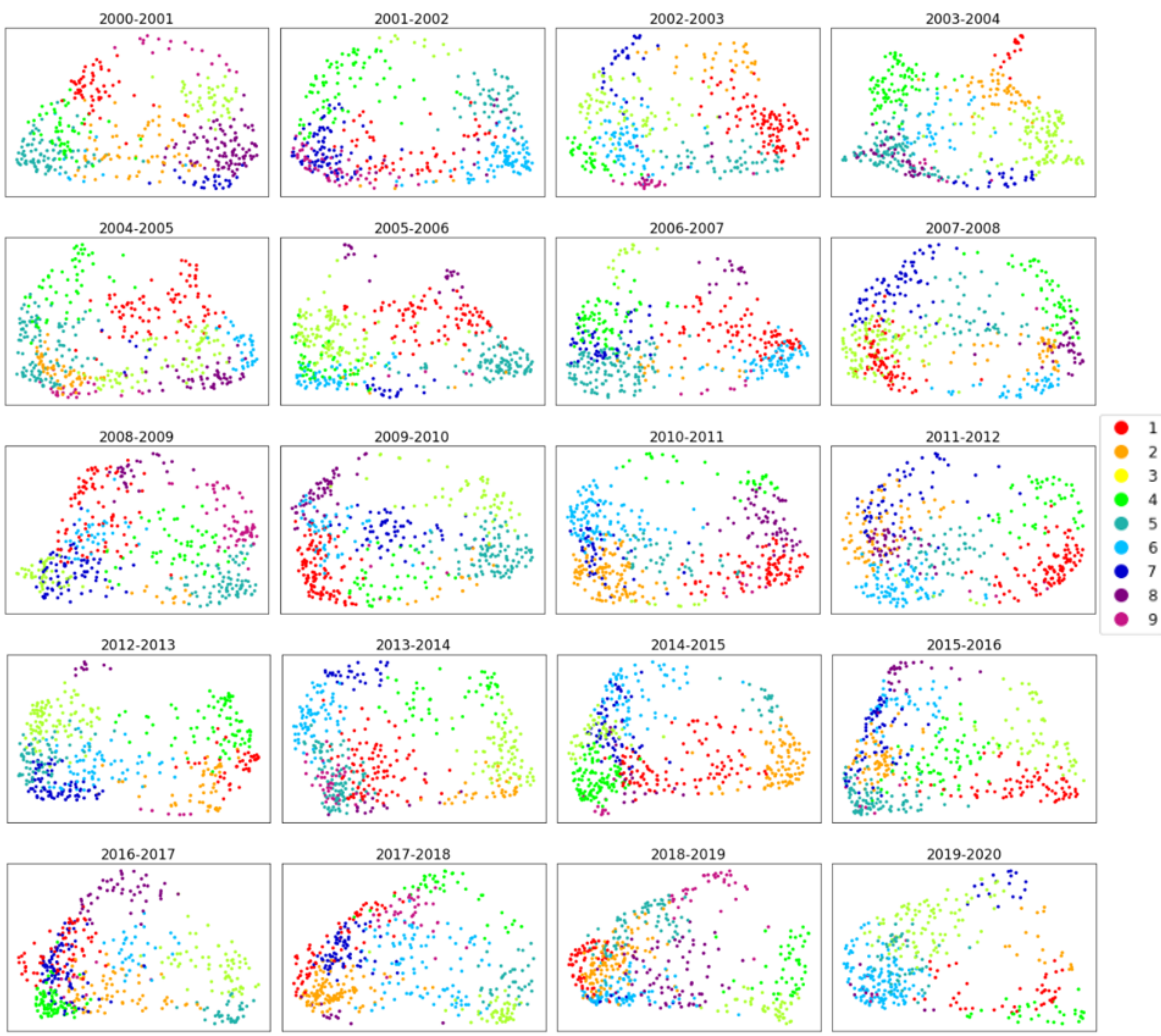


Fig. 4.9: PHATE plots for NBA seasons 2000-2001 to 2019-2020 - separated into nine clusters. Please note that cluster numbers are not consistent from season to season. For example, Cluster 9 in the 2000-2001 season corresponds to the `Superstar` players, while in the 2001-2002 season, Cluster 3 corresponds to the `Superstar` players.

The PHATE procedure (see Section 3.2.5) was also used to view the clustering results for all 20 NBA seasons. Figure 4.9 displays the results of dimensionality reduction in Python using PHATE for each of the 20 NBA seasons. These plots display the nine distinct clusters with considerably less overlap than the PCA plots. Note that the colors and cluster numbers mirror those of the PCA and tSNE plots.

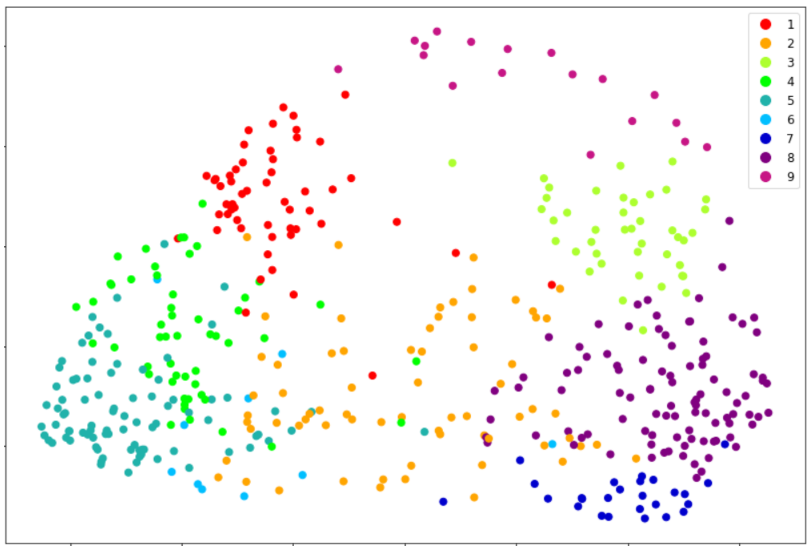


Fig. 4.10: PHATE plot for NBA players in the 2000-2001 season - separated into nine clusters. PHATE does an excellent job of displaying the uniqueness of many of the nine player clusters in two dimensions. Clusters 1 and 3 in the top right show particularly strong separation from the rest of the players.

We can take a closer look at the 2000-2001 NBA season in Figure 4.10 to view the unique clusters. We can see that Clusters 1, 3, 5, 7, 8, and 9 are mostly distinct from the rest of the data. Note that the cluster colors and numbers match those of Figure 4.4 and 4.7. For this particular season, Clusters 2, 4, and 6 appear to be spread across many other clusters with very little distinction. However, it is overall encouraging to see mostly clear distinctions between the clusters across all seasons.

4.4 GGobi

While `tSNE` and `PHATE` provide a certain level of clarity and justification for our use of nine different player clusters, we want to examine these clusters in more detail in a much more interactive fashion. GGobi provides a way to visualize and identify cluster separations across all dimensions available (see Section 3.5).

In this section, we will view snapshots of different projections created using the GGobi interface.

4.4.1 Grand Tour/Brushing Results

Based on the selection of nine clusters across the 20 years of data, we can use the *Brush* feature in GGobi to customize the color and shape of the points according to their partition. Table 4.2 shows how these cluster symbols line up with the colors and numbers in the PCA, tSNE, and PHATE plots.

Table 4.2: GGobi cluster colors and symbols compared to PCA, tSNE, and PHATE clusters

Cluster	PCA/tSNE/PHATE Color	GGobi Symbol	GGobi Color
1	Red	Large +	Purple
2	Orange	Large X	Pink
3	Yellow	Large ◯	Red
4	Lime Green	Large □	Blue
5	Sea Green	Small +	Green
6	Light Blue	Small x	Orange
7	Royal Blue	Small ∘	White
8	Purple	Small □	Gray
9	Magenta	Large +	Yellow

Once the data is appropriately brushed, we can view the data in many static plots, including scatter plots, histograms, and parallel coordinate plots. Since R provides plenty of options for static visualization, our primary focus with GGobi was the dynamic/interactive features provided by the grand tour functionality (see Section 3.5).

Figure 4.11 shows an example of one projection of the data from the 2000-2001 NBA season. We can see that many of the clusters show strong distinction in this two-dimensional view, including the small orange x's (top left), the small yellow +'s (top right), the small gray □'s (right), the large red ◯'s (lower middle), and the large yellow +'s (bottom). We will discuss to which player positions these various colors and shapes correspond in Section 6.2.1. The user may also note in the bottom left of Figure 4.11 the variables that contribute most to this projection. A longer bar indicates a larger impact, while a smaller bar indicates a low impact of a variable on the projection. It appears that the top four variables are 'X2', which corresponds to two-point shots, 'X3', which corresponds to three-point shots, 'PT', which corresponds to points, and 'FT', which corresponds to free-throws. Note that only the first two letters of each variable are displayed in the axes.

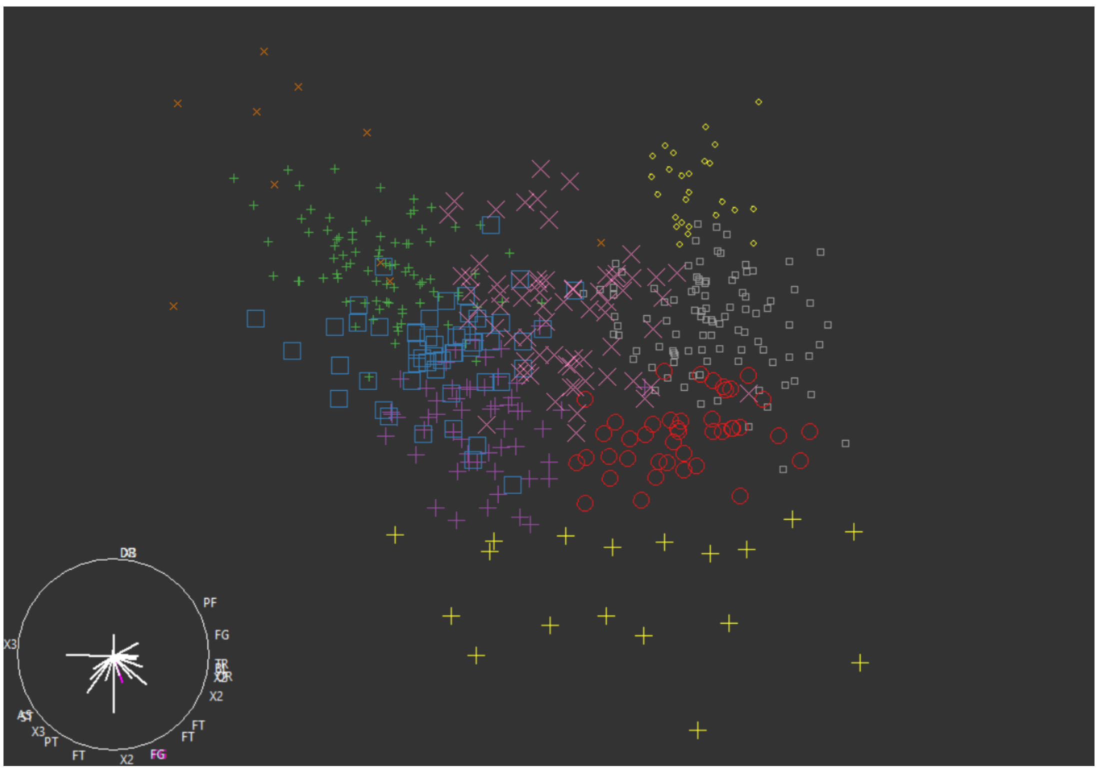


Fig. 4.11: Projection of nine clusters in GGobi. This projection shows several clusters clearly distinct from the rest of the players. Cluster 9 (`Superstars`; large yellow +'s) on the bottom is a notable example.

We can use the grand tour feature to seek low-dimensional representations of our data that show clear separations of each of our clusters at different points in the tour. We will briefly view a projection for each of the nine clusters at a point in the tour where they show distinction. While these static views of the dynamic tour will not completely capture every cluster's uniqueness, they provide insight into the process by which one can view each cluster's movement across all projections. The reader may also note that the projection map at the bottom left of each figure can aid in viewing which variables carry the most weight in the projection.

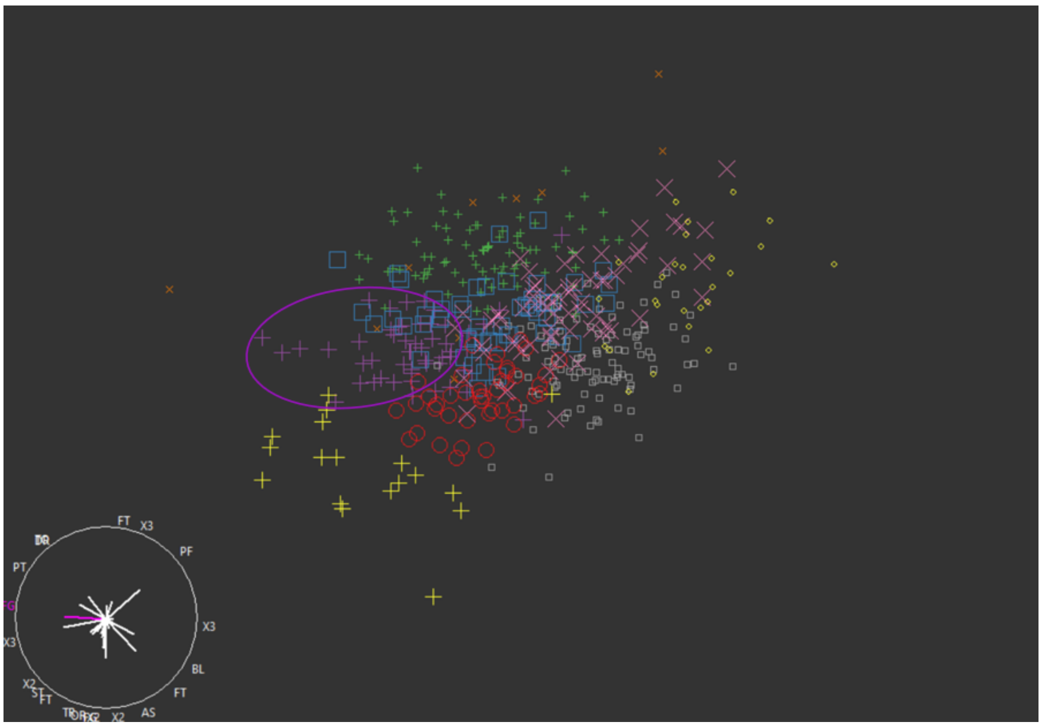


Fig. 4.12: Projection showing separation of Cluster 1 (Large Purple +'s) in GGobi

Figure 4.12 shows Cluster 1 with some minor separation from the rest. This cluster overlaps heavily with Cluster 4 in Figure 4.11, but this projection shows some clear distinction. Personal fouls ('PF') and free throws ('FT') appear to have a larger impact on this projection than other variables.

Figure 4.13 shows Cluster 2 with considerably less overlap than in most other projections viewed. We recall from our PCA, tSNE, and PHATE plots (see Section 4.3) that Cluster 2 represents one of the more ambiguous positions with players who appear to not stand out in any single area. It is encouraging to see some distinction in this figure, albeit with some

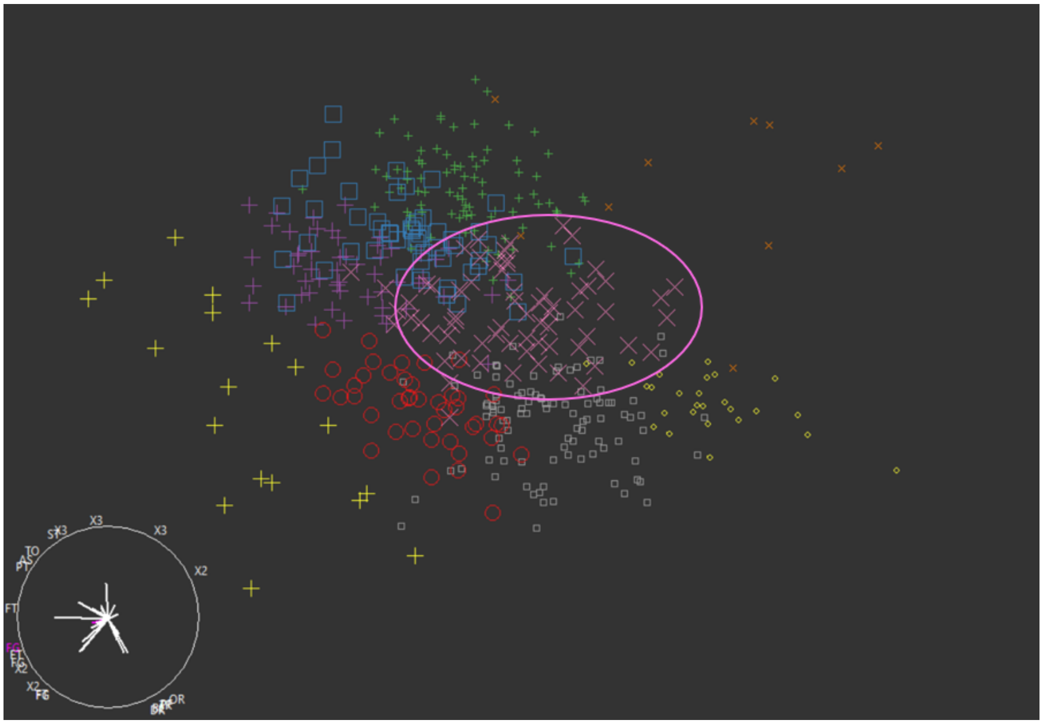


Fig. 4.13: Projection showing separation of Cluster 2 (Large Pink X's) in GGobi

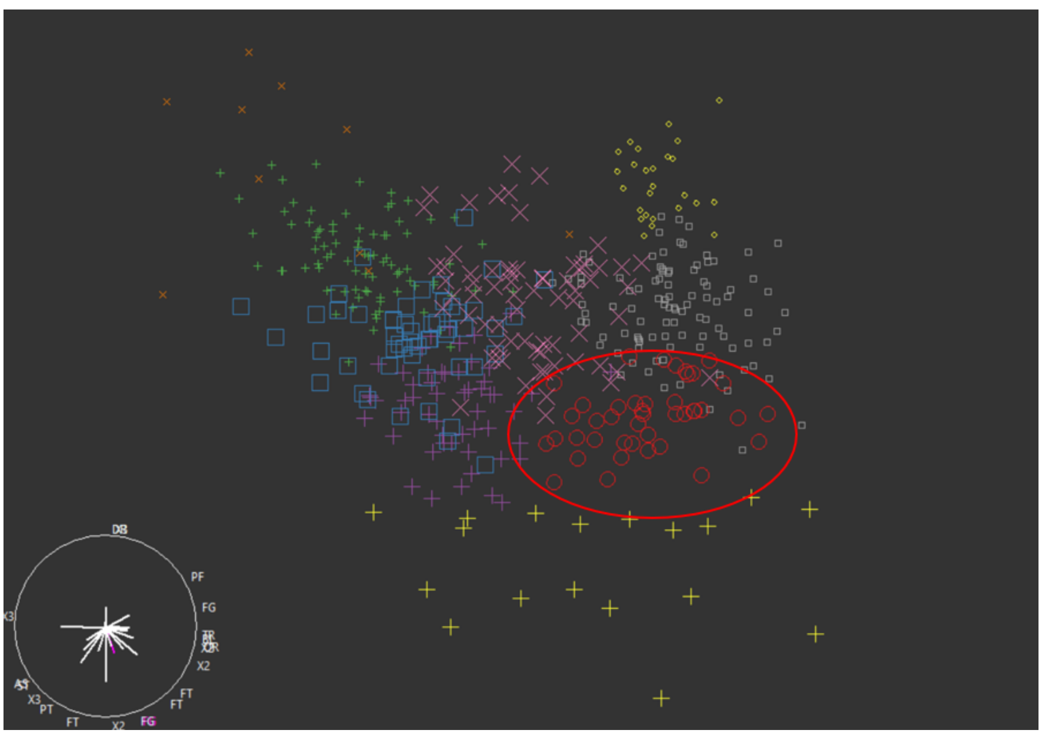


Fig. 4.14: Projection showing separation of Cluster 3 (Large Red ◯'s) in GGobi

overlap with several other clusters. The variables that contribute the most to this projection include free throws ('FT'), field goals ('FG'), and defensive rebounds ('DR').

Figure 4.14 displays the uniqueness of Cluster 3. This particular projection shows the large red ◯'s with very little overlap with any other cluster. Two-point shots ('X2'), three-point shots ('X3'), and points ('PT') appear to have the largest impact on this projection.

Figure 4.15 shows Cluster 4. The process of finding a projection to illustrate the uniqueness of this particular cluster proved extremely difficult. Like the players in Cluster 2, these players in Cluster 4 are usually found in the middle of all the projections. These players do not stand out in one particular area, so it is difficult to find a projection that distinguishes them clearly like some of the other positions. The variables that carry the most weight in this projection include offensive rebounds ('OR'), defensive rebounds ('DR'), and two-pointers ('X2').

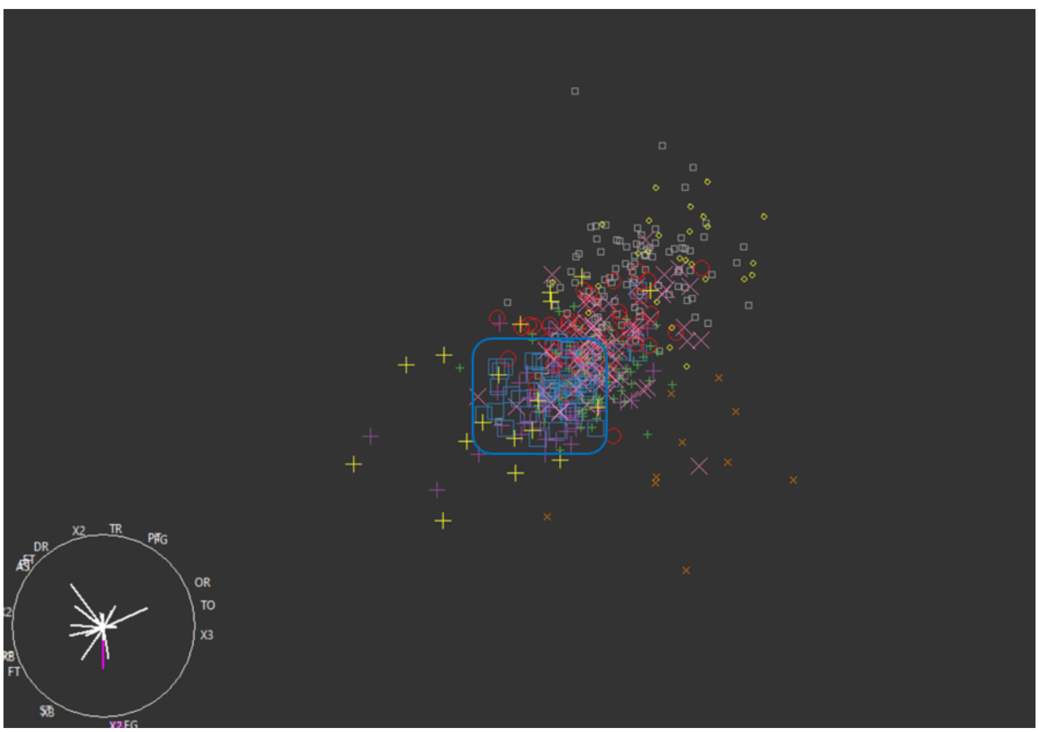


Fig. 4.15: Projection showing separation of Cluster 4 (Large Blue □'s) in GGobi

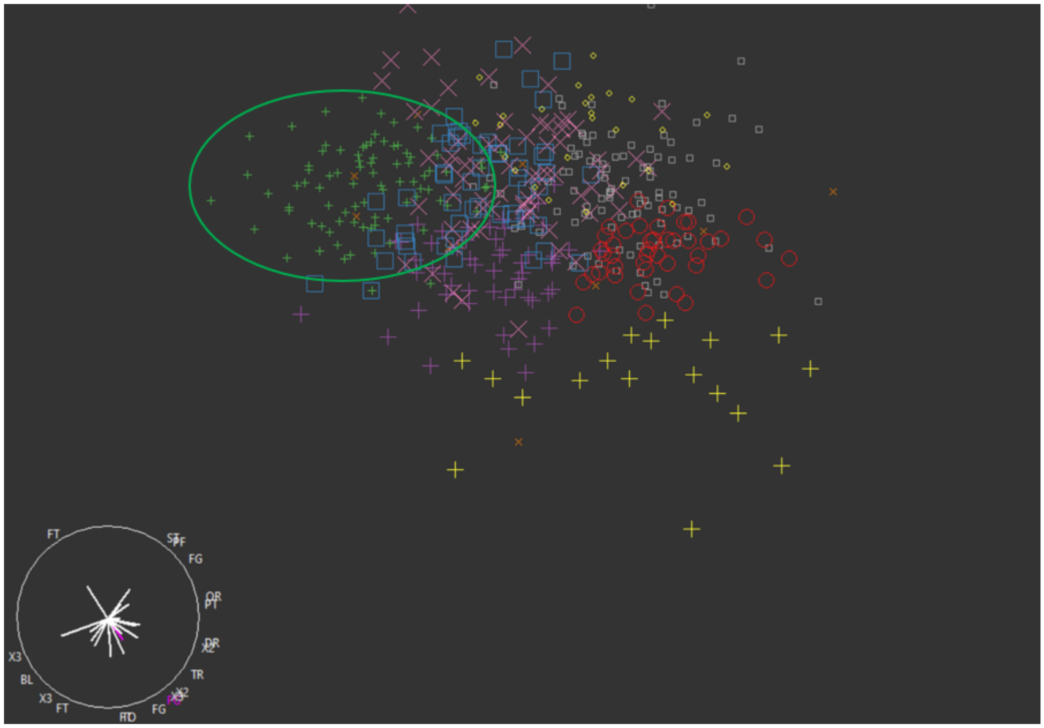


Fig. 4.16: Projection showing separation of Cluster 5 (Small Green +'s) in GGobi

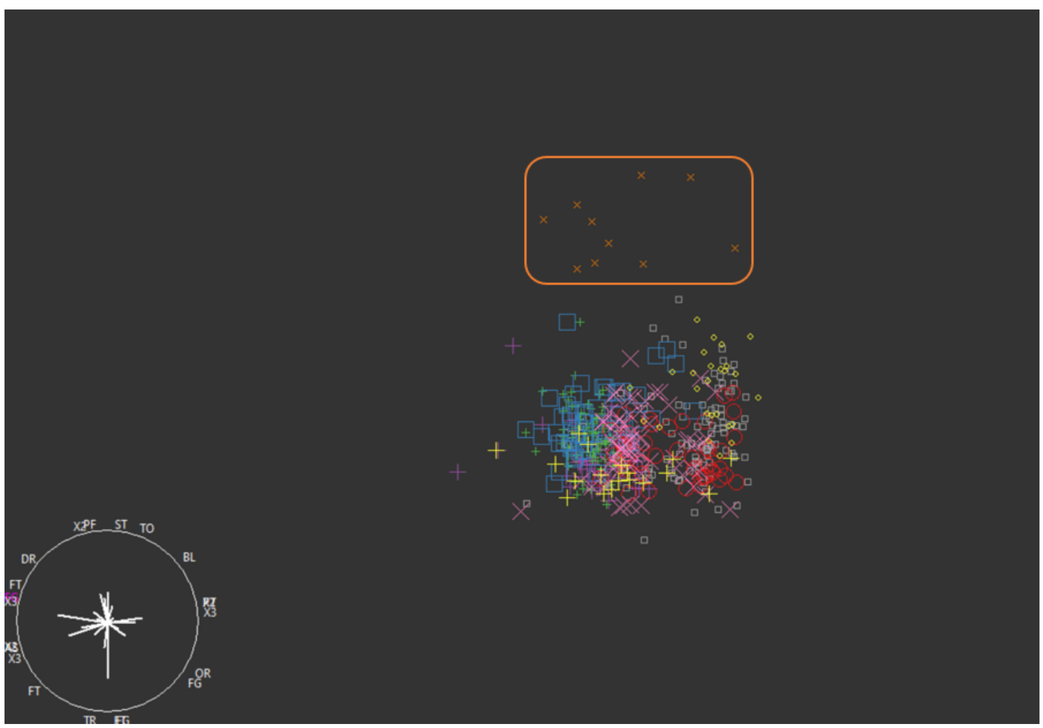


Fig. 4.17: Projection showing separation of Cluster 6 (Small Orange X's) in GGobi

Figure 4.16 shows a pause in the grand tour where Cluster 5 stands out on the upper left portion of the plot. Some variables that appear to have a bigger impact in this projection include free throws (‘FT’), three-pointers (‘X3’), and steals (‘ST’).

Figure 4.17 shows a very clear distinction for Cluster 6. This particular projection is extremely insightful since it appears that all the other players are packed together, while Cluster 6 stands out up above. Field goals (‘FG’) and three-pointers (‘X3’) appear to have the largest impact on this projection.

Figure 4.18 shows Cluster 7 with almost no overlaps. As was the case with several other clusters, it was relatively easy to find a projection where Cluster 7 stood out from the rest. In particular, when these data points were observed during the grand tour, they frequently did not follow the flow of the rest of the data and would move in opposite directions of the rest of the points. We will see in Section 5.2.1 that these players are part of the **Defensive Big Men** cluster. The most influential variables in this projection include two-pointers (‘X2’), three-pointers (‘X3’), and points (‘PT’).

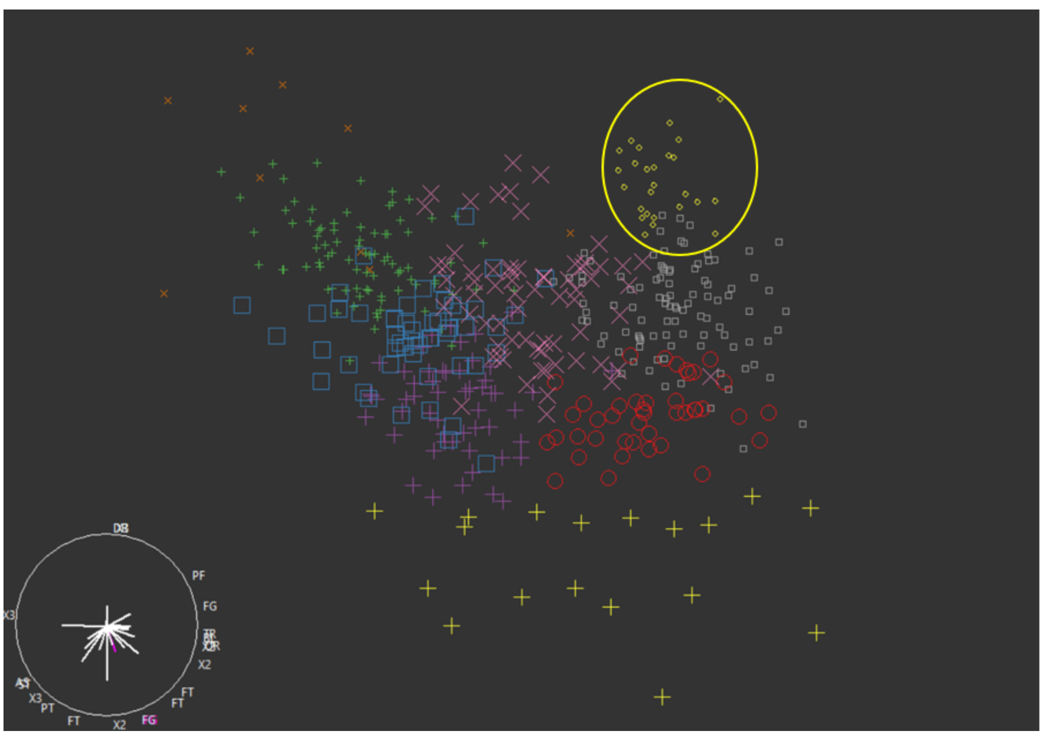


Fig. 4.18: Projection showing separation of Cluster 7 (Small Yellow ∘’s) in GGobi)

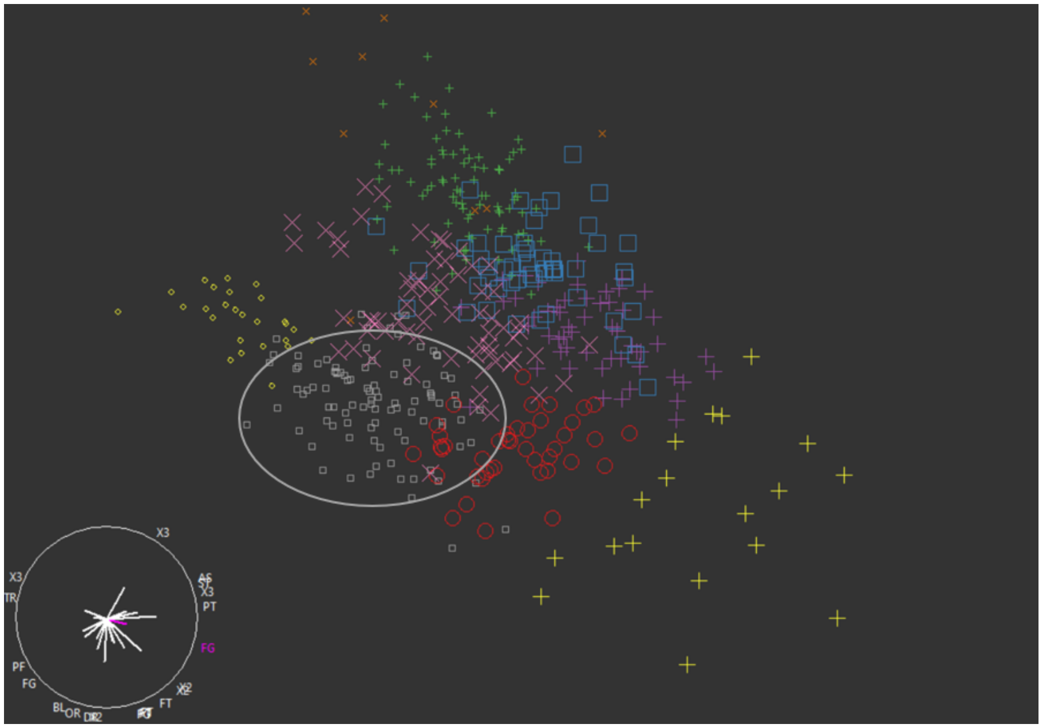


Fig. 4.19: Projection showing separation of Cluster 8 (Small Gray □'s) in GGobi

Figure 4.19 provides a moment in the grand tour where Cluster 8 overlaps very little with other clusters. It is interesting to note that this cluster's proximity to Cluster 7 to the upper left, Cluster 2 to the upper right, and Cluster 3 to the lower right. This same relationship can be seen in the introductory figure (Figure 4.11), meaning that these positions are likely related to one another. The most important variables in this projection include three-pointers ('X3'), points ('PT'), and two-pointers ('X2').

Figure 4.20 shows a very clear distinction of Cluster 9 from the rest of the players. Much like Cluster 3, this cluster is clearly distinct across many projections. We will see in Section 5.3.2 that Clusters 3 and 9 constitute the final agglomeration in the hierarchical clustering process, suggesting that they are the most distinct of the new player positions. In this projection, the variables that are most influential include include three-pointers ('X3'), points ('PT'), and two-pointers ('X2').

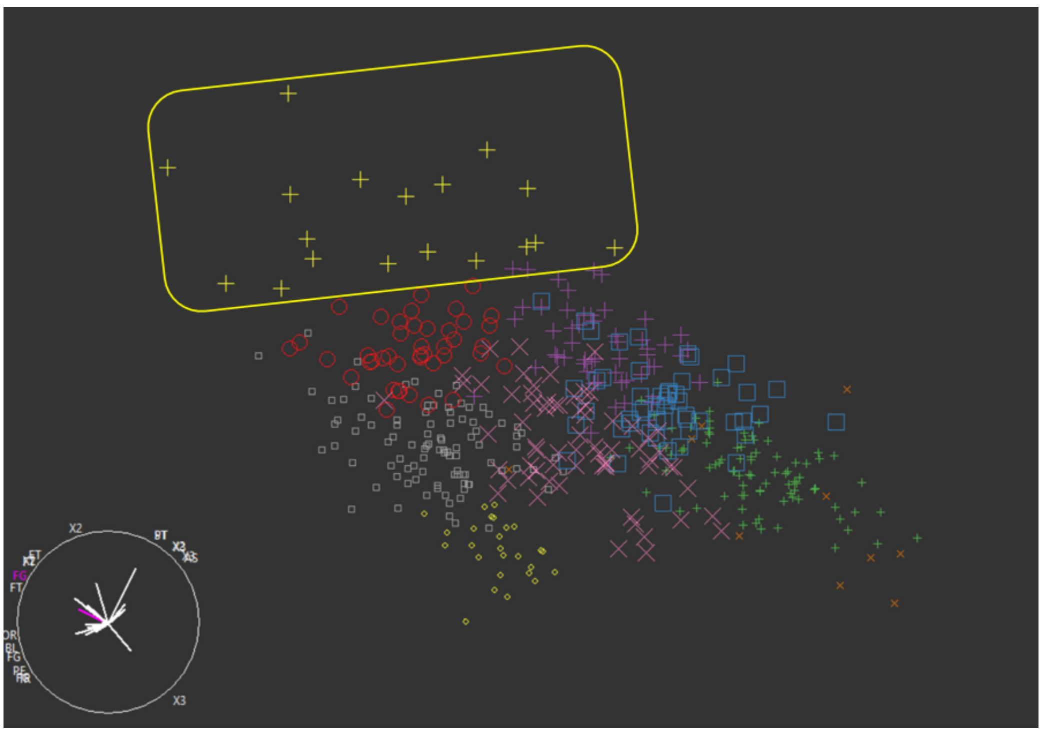


Fig. 4.20: Projection showing separation of Cluster 9 (Large Yellow +'s) in GGobi

CHAPTER 5

Clustering Results

Now that we have determined the validity and uniqueness of the nine clusters, we will explore the details and characteristics of the player clusters. We will verify the consistency of the clustering algorithm using the Adjusted Rand Index, and then explore and discuss the implementation of mega-clustering of the players across all 20 seasons.

5.1 Clustering by Year

Before we look at the characteristics of the nine NBA player clusters for the different seasons, we can use the Adjusted Rand Index (see Section 3.2.2) to confirm that the Ward D2 algorithm is somewhat consistent from year to year.

5.1.1 Adjusted Rand Index Results

We can see from the Table 5.1 that the ARI falls between 0.18 and 0.32. These results were compared to a random benchmark where all players were clustered randomly for each year. The amount of players randomly placed in a given cluster was fixed to the amount of players placed in that cluster by Ward's D2 method. After randomly assigning all players to one of the nine clusters, the Adjusted Rand Index was calculated comparing the two seasons. This process was simulated 9,999 times for each pair of seasons. Figure 5.1 displays a histogram of all simulations comparing the 2017-2018 NBA season to the 2018-2019 NBA season, while Figure 5.2 compares the simulations to the actual ARI for the two seasons based on Ward's D2 Method. A complete display of the ARI simulations for all pairs of adjacent NBA seasons can be found in Appendix D.

Table 5.1: Adjusted Rand Index comparing adjacent seasons. An ARI value of 0 would indicate no consistency in clustering from season to season, while a value of 1 would indicate identical clustering between two seasons. The lowest ARI result (0.182) occurs when comparing the 2018-2019 season to the 2019-2020 season, while the highest ARI result (0.348) results from comparing the 2009-2010 season to the 2010-2011 season.

Seasons	ARI
2000-2001 vs 2001-2002	0.193
2001-2002 vs 2002-2003	0.227
2002-2003 vs 2003-2004	0.210
2003-2004 vs 2004-2005	0.254
2004-2005 vs 2005-2006	0.208
2005-2006 vs 2006-2007	0.248
2006-2007 vs 2007-2008	0.227
2007-2008 vs 2008-2009	0.271
2008-2009 vs 2009-2010	0.277
2009-2010 vs 2010-2011	0.348
2010-2011 vs 2011-2012	0.316
2011-2012 vs 2012-2013	0.279
2012-2013 vs 2013-2014	0.242
2013-2014 vs 2014-2015	0.250
2014-2015 vs 2015-2016	0.248
2015-2016 vs 2016-2017	0.277
2016-2017 vs 2017-2018	0.247
2017-2018 vs 2018-2019	0.285
2018-2019 vs 2019-2020	0.182

We can see that the ARI does well in detecting true random cluster assignments as nearly all of the simulated scores fall between -0.02 and 0.02. We expect many players to evolve due to being traded to a new team, changing roles, or simply improving their skills, but the fact that we still see a significant link between any two adjacent seasons is very encouraging.

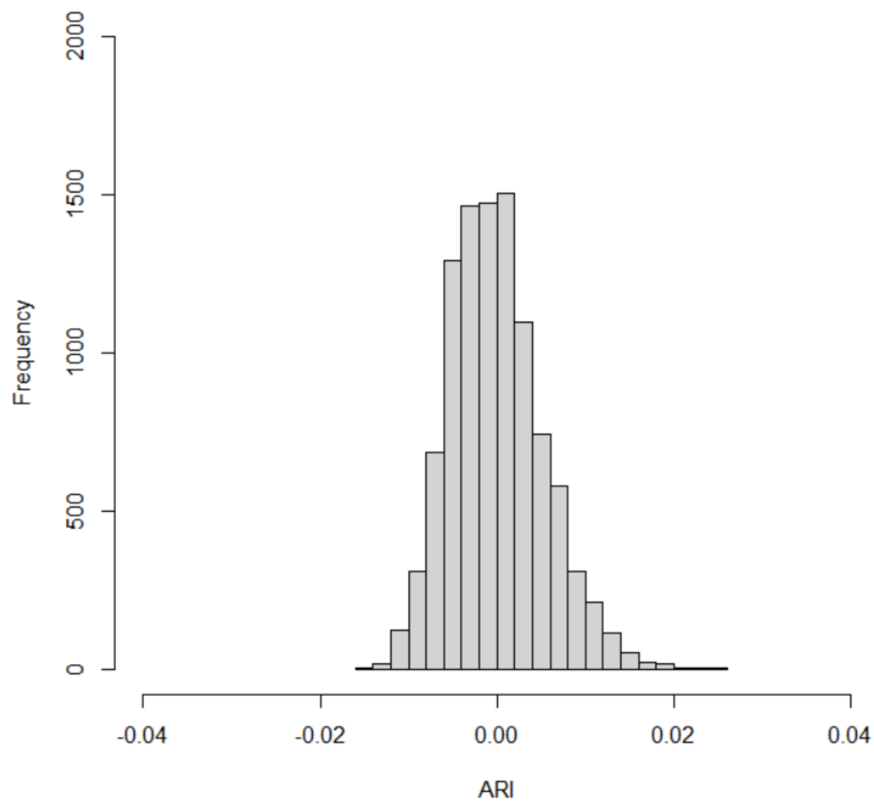


Fig. 5.1: ARI calculation for 9,999 simulations of random cluster assignment for the 2017-2018 and 2018-2019 NBA seasons. In a random simulation of clustering two seasons, we would expect most ARI values to fall around 0, meaning there was no consistency in the two seasons' clusterings of the same players. We can see that nearly all ARI values in the simulations fall between -0.02 and 0.02.

5.2 Exploring Clustering Characteristics for a Single Season

We will now explore the nine clusters for an individual season. We will choose the first season in the 20-year span (2000-2001) for this in-depth exploration to remain consistent with the close-up views from Chapter 4.

5.2.1 Single Season Cluster Characteristics

One way we can explore the characteristics of the different clusters is to look at their averages in each of the 21 statistical categories. For each cluster we will classify the statistical

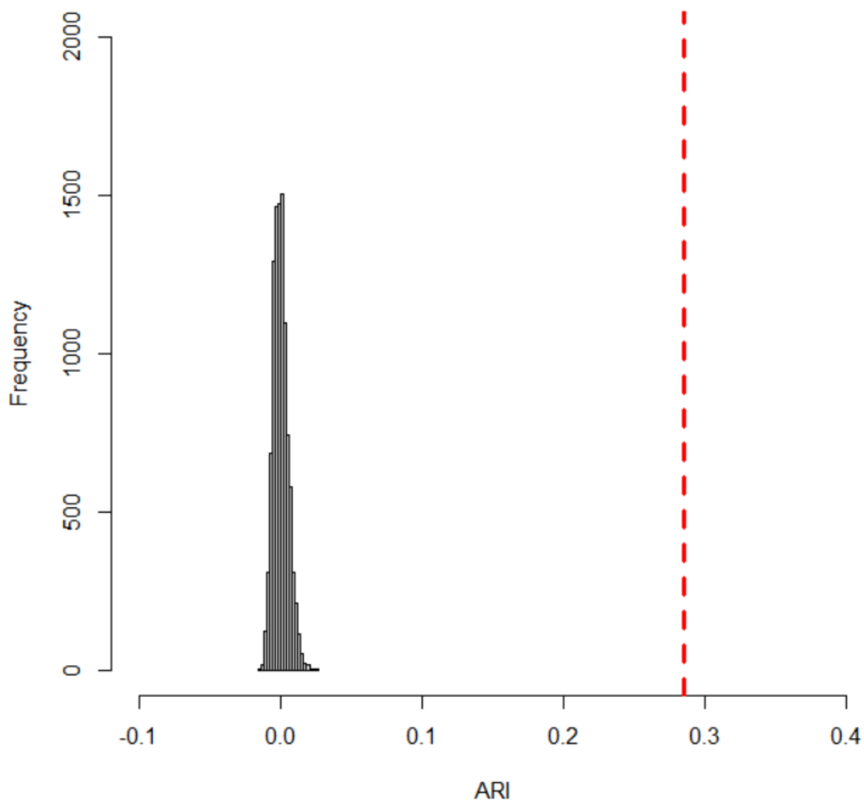


Fig. 5.2: ARI calculation for 9,999 simulations of random cluster assignment for the 2017-2018 and 2018-2019 NBA seasons compared to true ARI from hierarchical clustering. It is clear from these random clustering simulations that the true clustering results were somewhat consistent from season to season.

categories as ‘high’ if the average for the given cluster is above the 75th percentile for all players in the 2000-2001 NBA season. Table 5.2 displays the results of this method for all nine clusters across all 21 statistical categories.

Table 5.2: Cluster characteristics by statistical category for the 2000-2001 NBA season. 'high' values indicate that players in this cluster are, on average, above the 75th percentile for all players in the given season. 'low' values indicate that players in this cluster are, on average, below the 25th percentile for all players in the given season.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
FG	high		high				low		high
FGA	high		high				low		high
FG%		low			low	low		high	
3P	high				high	high			
3PA					high	high	low		
3P%	high				high		low		
2P	high		high		low	low	low		high
2PA	high		high		low	low	low		high
2P%		low				low	low	high	
FT	high		high			low	low		high
FTA			high		low	low			high
FT%						low	low		
ORB				low	low	low	high	high	
DRB				low			high	high	
TRB				low			high	high	
AST				high			low	low	
STL				high		high			
BLK						low	high	high	high
TOV				high		high			high
PF							high	high	
PTS	high		high			low	low		high

From Table 5.2, we can pick out some unique characteristics of the different clusters, and perhaps make some preliminary assumptions about the types of players who were likely classified in this particular group. We can pair these results with Table 5.3, which displays 10 players selected from each cluster. Generally, more well-known players were selected as examples as this will make it easier to analyze player roles based on the players' perceived impacts on the court. The table also includes the total number of players found in each cluster. If there were less than 10 players found in a cluster, all players from that cluster are included in the table. The full list of players in each cluster for the 2000-2001 NBA season, as well as clustering assignments by player for the other 19 NBA seasons, can be found in the `Mega_Cluster` sub-folder of the GitHub repository.

Using Table 5.2 and Table 5.3, we will characterize these new player positions based on

Table 5.3: Notable players in each cluster for the 2000-2001 NBA season

#1 Score-First Guards	**#2 Bench Role Players**	**#3 Scoring Big Men**
Total Players: 54	Total Players: 68	Total Players: 42
Ray Allen - MIL	John Amaechi - ORL	Vin Baker - SEA
Michael Finley - DAL	Shandon Anderson - HOU	Vlade Divac - SAC
Steve Francis - HOU	Isaac Austin - VAN	Juwan Howard - WAS
Allan Houston - NYK	David Benoit - UTA	Juwan Howard - DAL
Rashard Lewis - SEA	PJ Brown - CHA	Zydrunas Ilgauskas - CLE
Dirk Nowitzki - DAL	Desmond Mason - SEA	Shawn Kemp - POR
Gary Payton - SEA	Greg Foster - LAL	Hakeem Olajuwon - HOU
Latrell Sprewell - NYK	Devean George - LAL	Rasheed Wallace - POR
Peja Stojakovic - SAC	AC Green - MIA	Donyell Marshall - UTA
Grant Hill - ORL	Ron Harper - LAL	Elton Brand - CHI

#4 Pass-First Guards	**#5 Two-Way Players/ Primary Defenders**	**#6 Bench Perimeter Scorers**
Total Players: 52	Total Players: 86	Total Players: 10
Stacey Augmon - POR	Brent Barry - SEA	Nick Anderson - SAC
Mookie Blaylock - GSW	Bruce Bowen - MIA	Eric Barkley - POR
Baron Davis - CHA	Jamal Crawford - CHI	Raja Bell - PHI
Anfernee Hardaway - PHX	Dell Curry - TOR	Muggsy Bogues - TOR
Tim Hardaway - MIA	Derek Fisher - LAL	Scott Burrell - CHA
Bobby Jackson - SAC	Hersey Hawkins - CHA	Kornel David - DET
Jason Kidd - PHX	Robert Horry - LAL	Michael Hawkins - CLE
Steve Nash - DAL	Steve Kerr - SAS	Jaren Jackson - SAS
Scotti Pippen - POR	Hedo Turkoglu - SAC	Terry Mills - IND
John Stockton - UTA	Bryon Russell - UTA	Elliot Perry - ORL

#7 Defensive Big Men	**#8 Interior Big Men**	**#9 Superstars**
Total Players: 26	Total Players: 98	Total Players: 18
Luc Longley - NYK	Marcus Camby - NYK	Kobe Bryant - LAL
Ben Wallace - DET	Erick Dampier - GSW	Vince Carter - TOR
Otis Thorpe - CHA	Patrick Ewing - SEA	Tim Duncan - SAS
Eric Montross - DET	Kenyon Martin - NJN	Kevin Garnett - MIN
Eric Montross - TOR	Dikembe Mutombo - ATL	Allen Iverson - PHI
Jeff Foster - IND	Dikembe Mutombo - PHI	Karl Malone - UTA
Duane Causwell - MIA	Jermaine O'Neal - IND	Tracy McGrady - ORL
Adonal Foyle - GSW	Greg Ostertag - UTA	Shaquille O'Neal - LAL
Michael Ruffin - CHI	Shawn Bradley - DAL	Paul Pierce - BOS
Joel Pryzbilla - MIL	Jamaal Magloire - CHA	Chris Webber - SAC

their 'highs' and 'lows' and the key players that were placed in these clusters. For example, Cluster 1 shows 'high' values in most of the scoring categories, including field goals made and attempted, three-pointers made, free-throws made, and total points. We can also see that Cluster 1 includes players such as Ray Allen, Michael Finley, and Steve Francis. These players' main role on the court was to score from the outside and inside, rather than ball facilitating/distributing. We will call this position **Score-First Guards**.

Cluster 2 contains players with low field goal percentage and two-point percentage, and with average marks in every other category. Most of these players came off the bench and were relied on for defense and hustle, rather than scoring and distributing. These players didn't tend to make a splash on the box score and they very likely had inconsistent minutes, so we will call these players **Bench Role Players**.

Cluster 3 shows similar ‘high’ values to Cluster 1. The key difference here is that Cluster 3 does not show ‘high’ values in three-pointers made or three-point percentage like Cluster 1. We can also see players like Vlade Divac, Shawn Kemp, and Hakeem Olajuwon. These players were well-known big men who scored a lot from the interior, and went to the free-throw line at high rates. We will call these players **Scoring Big Men**.

Cluster 4 is highlighted by lower-than-average rebounding numbers, and higher-than-average assists, steals, and turnovers. This information coupled with some key players like Jason Kidd, Steve Nash, and John Stockton indicate that this position is clearly for guards whose primary role is ball handling and passing. These players have the ball in their hands a lot, so they get credit for a lot more assists, but also a lot more turnovers. We will call this position **Pass-First Guards**.

Cluster 5 contains players with ‘high’ values for all three-point shooting categories, and ‘low’ values for two-pointers made and attempted. We can see players like Brent Barry, Dell Curry, and Steve Kerr, who are known as some of the best three-point shooters in the NBA. We can also see players like Bruce Bowen, Robert Horry, and Bryon Russell. These players could shoot the three, but also frequently took the most difficult defensive assignment. We will call this position **Two-Way Players/Primary Defenders**.

Cluster 6 shows players with higher three-point making and shooting averages, but low points per game averages. This may indicate that these players were scorers, but they didn’t get as many minutes. Players like Nick Anderson, Raja Bell, and Jaren Jackson played fewer games and came off the bench. These players were good scorers, but may have been inconsistent with minutes throughout the season. We will call these players **Bench Perimeter Scorers**.

Cluster 7 shows players who take very few shots from any distance, and are low on points. These players have high rebounding and blocking totals. This information combined with player names like Luc Longley and Ben Wallace indicates that these players are **Defensive Big Men**. These players’ primary role on the floor is to defend the paint and contest shots from close range.

Cluster 8 contains similar ‘highs’ and ‘lows’ to Cluster 7, but these players don’t show ‘low’ values for shots attempted. Players like Marcus Camby, Patrick Ewing, and Kenyon Martin were strong paint defenders, but were effective post-up players who could score around the basket at high percentages. We will call these players **Interior Big Men**.

Cluster 9 is the easiest group to distinguish. These players have ‘high’ values in a wide range of statistics, including field goals attempted, free throws attempted, turnovers, and points. Players like Kobe Bryant, Vince Carter, and Allen Iverson were among the elite superstars of the league. These players have the ball in their hands on most offensive possessions, and they take all of the big shots. We will call these players the **Superstars**.

It is important to note at this point that these new clusters each contain many players from different traditional positions (see Section 1.1.1). For example, the **Score-First Guards** includes Shooting Guards like Ray Allen and Michael Finley, Point Guards like Steve Francis and Gary Payton, small forwards like Peja Stojakovic and Grant Hill, and even power forwards like Dirk Nowitzki. The **Superstars** cluster provides another example of the variety of standard positions that can be found within these new clusters. Allen Iverson (Point Guard), Kobe Bryant (Shooting Guard), Vince Carter (Small Forward), Karl Malone (Power Forward), and Shaquille O’Neal (Center) are all traditionally classified as different positions, but are placed in the same cluster here due to their actual performance.

This type of analysis can be conducted for each of the nine clusters across all 20 seasons to determine what unique roles are found on the court that are not currently classified by any player position. We will take a closer look at each individual cluster for the players across all seasons combined in the next sections.

5.3 Mega-Clustering

In this section we will explore a new method developed to cluster across all 20 NBA seasons. We will refer to this method as *mega-clustering*, since it involves applying the same hierarchical clustering method, but instead of to individual players, the clustering applies to the nine clusters for each year. In total we will have 9 x 20 = 180 individual ‘objects’, and we will be clustering each one into one of nine 'mega-clusters'. We will then view some of the characteristics of each 'mega-cluster' to determine our new player positions. We will also provide an example of how an individual player's position can evolve over the course of his career. Finally, we will display the results of combined clustering where all individual players from each season will be clustered together.

5.3.1 Methodology

Before we conduct our analysis of the nine 'mega-clusters', we will calculate the 'highs' and 'lows' for every cluster across every season. This will result in each season having a table having nine rows, one for each of the nine clusters, and 21 columns, one for each of the statistical categories. We then can then add an identifier column that lists the year that this season ended.

In order to *mega-cluster* all of the season clusters, we must append each year's table so that we have 180 rows, one for each cluster, with each row containing their 'highs' and 'lows'. Finally, we can convert the 'high' values to 1's, the 'low' values to -1, and the blanks to 0's. Table 5.4 displays the first 20 rows of this new table for all clusters across the 20 NBA seasons. The full table can be found at the following link: `https://github.com/ahed1194/MS_Thesis/blob/main/Mega_Cluster/megaclusters.csv`.

Now that we have the 180 rows that show each cluster's characteristics across the 20 seasons, we can apply Ward's D2 method to 'cluster the clusters' into one of nine groups. The purpose of this *mega-clustering* approach is to to link each player position from year to year. For example, if Stephen Curry is classified into Cluster 7 in the 2018-2019 season, we want to see which cluster he is in for the 2019-2020 season. We would expect him to be grouped with similar players in both years, assuming his skills and his role on the team didn't change. We also want to see which players most consistently appear in the same cluster. As a reminder, the cluster numbers vary from year to year, so we will need to use these 'mega-clusters' to see which player appears in that same group the most.

Obviously, we would hope for a one-to-one matching from year to year. This way, each 'mega-cluster' would have 20 observations: one cluster from each season. It was highly

Table 5.4: Cluster characteristics for NBA seasons 2000-2001 to 2019-2020 – first 20 rows. A value of '1' for a given player cluster indicates that these players, on average, are higher than the 75th percentile of all players for the given season and the given statistic. A value of '-1' for a given player cluster indicates that these players, on average, are below the 25th percentile of all players for the given season and the given statistic. A value of '0' is given for all players in between.

CLUSTER	YEAR	FG	FGA	FG.	X3P	X3PA	X3P.	X2P	X2PA	X2P.	FT	FTA	FT.	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS
1	2001	1	1	0	1	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1
2	2001	0	0	-1	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0
3	2001	1	1	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	1
4	2001	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	-1	1	1	0	1	0	0
5	2001	0	0	-1	1	1	1	-1	-1	0	0	-1	0	-1	0	0	0	0	0	0	0	0
6	2001	0	0	-1	1	1	0	-1	-1	-1	-1	-1	-1	-1	0	0	0	1	-1	1	0	-1
7	2001	-1	-1	0	0	-1	-1	-1	-1	-1	-1	0	-1	1	1	1	-1	0	1	0	1	-1
8	2001	0	0	1	0	0	0	0	0	1	0	0	0	1	1	1	-1	0	1	0	1	0
9	2001	1	1	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0	1	1	0	1
1	2002	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	2002	-1	-1	-1	0	0	0	-1	-1	-1	-1	-1	-1	0	0	0	0	0	0	0	0	-1
3	2002	1	1	1	0	0	0	1	1	0	1	1	0	0	1	0	1	0	0	1	0	1
4	2002	1	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1
5	2002	0	0	1	0	0	0	1	1	1	1	1	0	1	1	1	0	0	1	0	0	0
6	2002	-1	-1	0	0	-1	-1	0	0	0	0	0	-1	1	1	1	-1	0	1	0	1	-1
7	2002	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	-1	1	1	0	1	0	0
8	2002	-1	0	-1	0	0	-1	0	0	-1	1	1	-1	0	0	0	0	0	0	0	1	0
9	2002	0	0	0	1	1	0	-1	-1	0	0	-1	0	0	0	0	0	0	0	0	0	0
1	2003	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	-1	0	1	0	1	0
2	2003	1	1	1	0	0	0	1	1	1	1	1	0	0	1	1	0	0	1	0	0	1

likely that this wouldn't match up perfectly, but we hope that we will at least see close to 20 observations in each.

5.3.2 Mega-Clustering Visualization

Once Ward's D2 method was performed on our new data, we can apply the same visualizations and analyses that were conducted on the individual seasons.

We will begin by viewing the hierarchical clustering process to see when and how the nine 'mega-clusters' were formed. Figure 5.3 shows the order in which each of the player positions were split from the complete data. We can recall from Section 3.1.2 that we can consider the dendogram from top-to-bottom or bottom-to-top. In this case it is informative to discuss the clustering in terms of 'splits' from the top down.

We can see that the first split (labeled '1' at the top of the plot) separates Clusters 9 and 3 from the rest of the data. This means that these two positions were clearly the most distinct from the rest of the players. We will see in Section 5.3.3 that these two positions correspond to the **Superstars** (Cluster 3) and the **Scoring Big Men** (Cluster 9). These are generally the most dominant players on the floor.

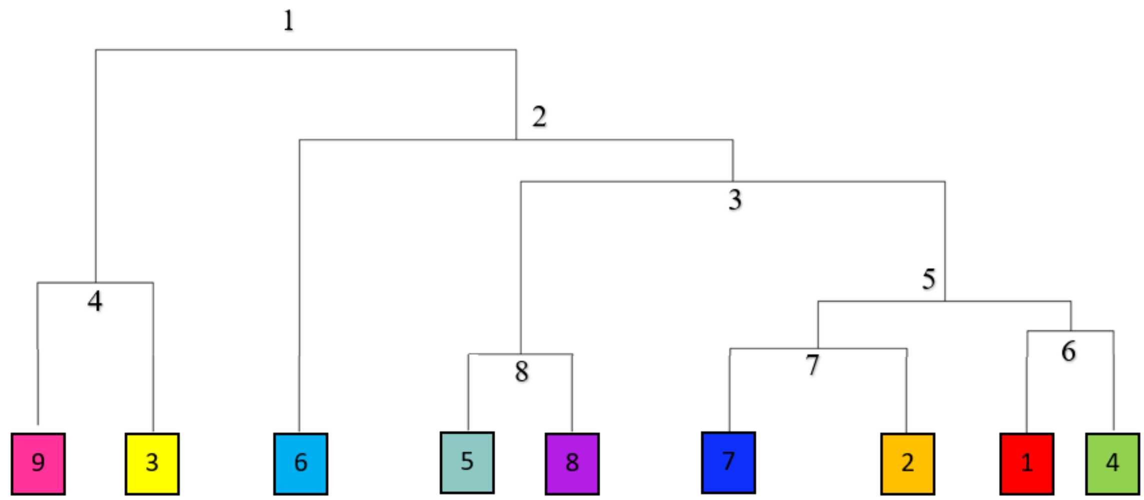


Fig. 5.3: Dendogram displaying hierarchical clustering of the nine 'mega-clusters'. The higher the combination of two clusters occurs, the more distinct these clusters are. We can see that the final connection brings Clusters 3 and 9 (`Superstars` and `Scoring Big Men` together with the other seven clusters.

We can also see that the final 'split' occurs as Clusters 5 and 8 are separated. Apparently, these two 'mega-clusters' are the most similar of the nine. We will also see in Section 5.3.3 that these two clusters correspond to the **Miscellaneous/Transient Players** (Cluster 5) and the **Bench Role Players** (Cluster 8). These two positions are quite similar in that they include players who have fairly small impacts on the floor and who tend to play relatively few minutes per game. This dendogram will be discussed in more detail in Section 6.2.2.

We can now begin to explore the various dimensionality reduction methods discussed in Section 3.2. We will begin with the visualization of the nine 'mega-clusters' using PCA via the `factoextra` R package (see Section 3.2.3 and Section 3.3.9). Figure 5.4 displays the results of PCA on the new 'mega-cluster' data.

In Figure 5.4, we can see that four of the 'mega-clusters' are highly distinct even in this two-dimensional view. We also can see individual data points highlighted. The observation labels consist of a starting number from 1 to 9 corresponding to one of the nine clusters for a given season, and the last four digits correspond to the ending year of that NBA season.

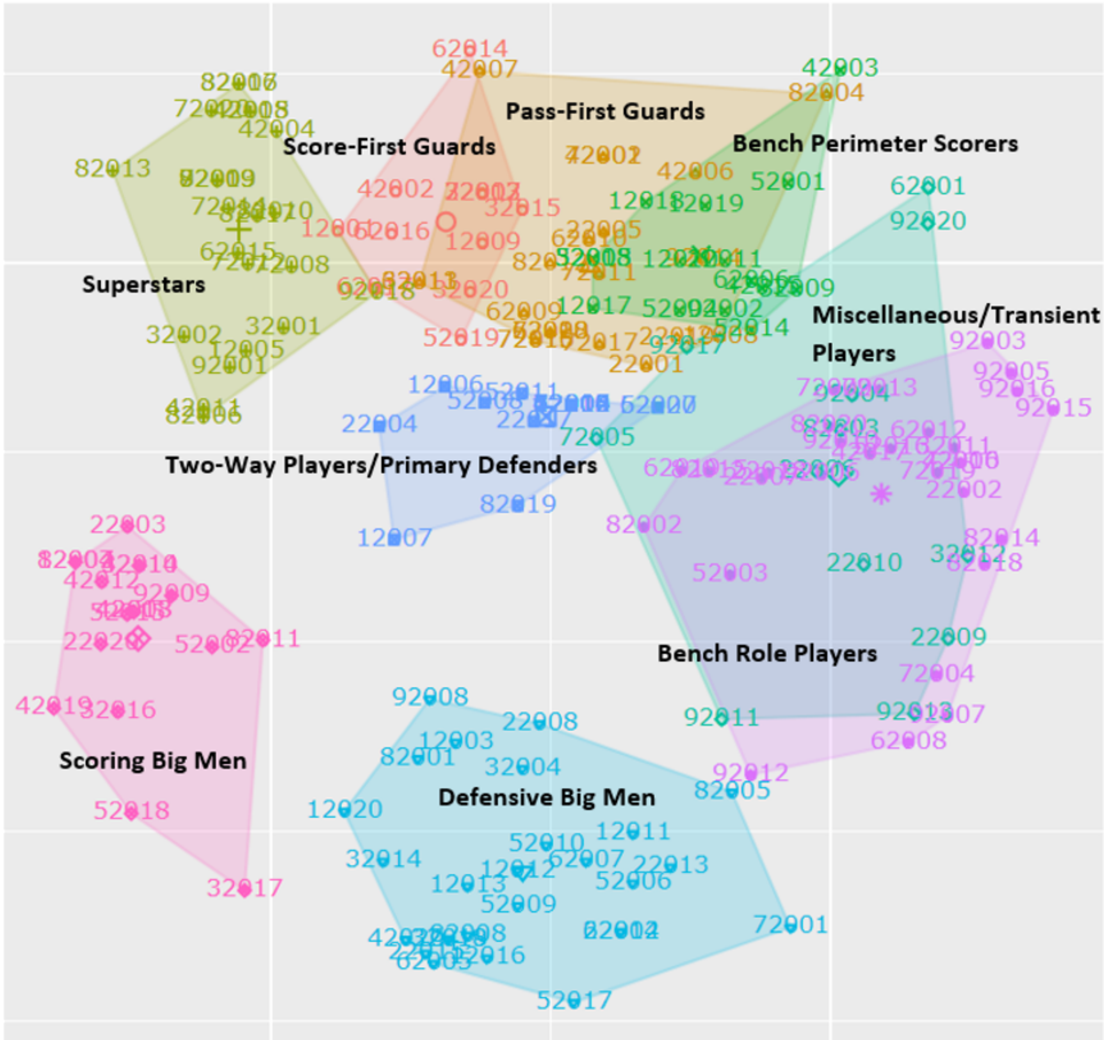


Fig. 5.4: 'mega-clusters' using PCA from the `factoextra` R package. The `Score-First Guards` and the `Pass-First Guards` appear to overlap, likely due to many similar aspects of their positions, while the `Defensive Big Men` and the `Scoring Big Men` appear well-separated from the rest of players, likely due to their highly distinctive roles.

For example, '1.2007' refers to the first cluster from the 2006-2007 NBA season.

While this initial PCA visualization shows distinctness for several 'mega-clusters' we would like to further visualize these clusters using tSNE (see Section 3.2.4 and Section 3.3.11). Figure 5.5 displays the same 'mega-clusters' using tSNE in R.

The same labeling method of cluster number and year was applied to the tSNE figure as to the PCA figure. We can see that all of these 'mega-clusters' are highly distinct through this visualization. We can also see that most of the 'mega-clusters' appear to have a similar

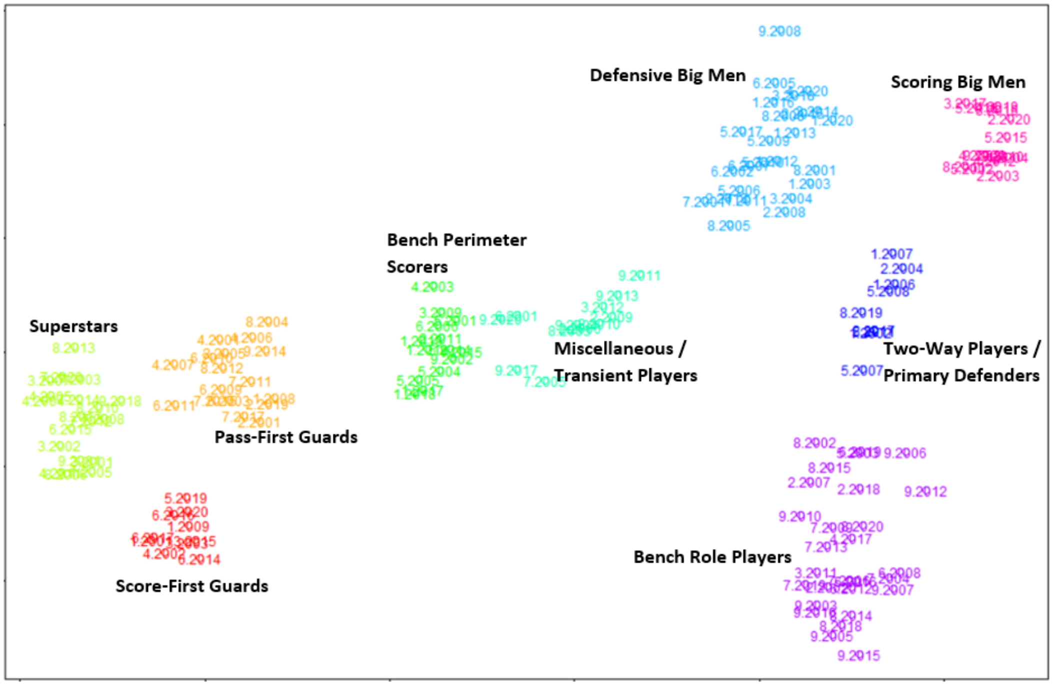


Fig. 5.5: 'mega-clusters' using tSNE from the `Rtsne` R package. This visualization technique displays clear separation for all nine player clusters.

amount of data points. For the most part, we don't see many repeat years in the same cluster, although we can see an example in the purple cluster on the bottom right that we have cluster 8 and cluster 9 from the 2014-2015 season.

5.3.3 Mega-Clustering Results

Table 5.5 provides the distribution of yearly clusters in each of the nine 'mega-clusters'. We can see that Clusters 1 and 5 have only 12 year clusters, while Clusters 6 and 8 have 27 and 29, respectively. Five of the nine 'mega-clusters' have more than 20 observations, while the other four have fewer than 20 observations.

We can view which players appear the most frequently in each 'mega-cluster'. Table 5.6 shows the top 10 players in each 'mega-cluster' based on number of appearances, while Table 5.7 shows the top 10 players in each 'mega-cluster' based on the percentage of their career spent in that particular cluster. This second top 10 list was implemented to capture players that either had shorter careers, or whose entire playing career is not captured in the

Table 5.5: Number of season clusters in each ‘mega-cluster’ – filled red for ‘mega-clusters’ with less than 20 season clusters and green for ‘mega-clusters’ with more than 20 season clusters

mega-cluster	Count
1	12
2	22
3	23
4	16
5	12
6	27
7	22
8	29
9	17

20-year span being analyzed. A full list of the number of appearances by player in each of these ‘mega-clusters’ can be found in the GitHub repository by accessing the following link: (https://github.com/ahed1194/MS_Thesis/tree/main/Mega_Cluster).

Based on these lists of key players in each ‘mega-cluster’, we can begin to characterize these different ‘mega-clusters’ into new player positions.

5.3.4 Cluster 1: Score-First Guards

Cluster 1 (red cluster in top middle of Figure 5.4 and red cluster in bottom left of Figure 5.5) contains players such as Ray Allen, Jamaal Crawford, JJ Redick, and CJ McCollum. While these players are usually not the top scorer on the team, they are known as great scorers. Taking a look at Jamal Crawford’s career statistics at https://www.basketball-reference.com/players/c/crawfja01.html, we can see that from the 2009-2010 season to the 2019-2020 season, he started in only 40 games (less than four starts per season), and he still managed to average over 14 points per game seven times. When these players are on the floor, their primary goal is to find ways to score. We will call this position **Score-First Guards**.

5.3.5 Cluster 2: Pass-First Guards

Cluster 2 (orange cluster in top middle of Figure 5.4 and orange cluster in middle

Table 5.6: Most frequently occurring players in each 'mega-cluster'

Cluster 1 Score-First Guards	# of Apps	Cluster 2 Pass-First Guards	# of Apps	Cluster 3 Superstars	# of Apps
Jason Richardson	7	Andre Miller	15	LeBron James	17
Ray Allen	6	Raymond Felton	13	Dwyane Wade	17
Jamaal Crawford	6	Rajon Rondo	13	Kobe Bryant	16
JJ Redick	6	Beno Udrih	13	Carmelo Anthony	13
Al Harrington	6	Jose Calderon	12	Allen Iverson	12
Tim Thomas	6	Earl Watson	11	Tony Parker	12
Leandro Barbosa	5	Ish Smith	11	Russell Westbrook	12
Trey Burke	5	Jarrett Jack	10	Kevin Durant	11
Alec Burks	5	Jeff Teague	10	Paul Pierce	10
Brevin Knight	5	Steve Nash	9	Derrick Rose	10

Cluster 4 Bench Perimeter Scorers	# of Apps	Cluster 5 Miscellaneous/Transient Players	# of Apps	Cluster 6 Defensive Big Men	# of Apps
Kyle Korver	14	Earl Barron	2	Tyson Chandler	19
Marco Belinelli	10	Jarron Collins	2	Reggie Evans	14
Rasual Butler	10	Jason Collins	2	Brendan Haywood	14
Wayne Ellington	10	Justin Harper	2	Kendrick Perkins	14
James Posey	9	Solomon Jones	2	Marcus Camby	13
Derek Fisher	8	Art Long	2	Zaza Pachulia	13
Damon Jones	8	Primoz Brezec	2	Anderson Varejao	13
James Jones	8	Dominic McGuire	2	Samuel Dalembert	11
Wes Matthews	8	Byron Mullens	2	Ben Wallace	11
Brent Barry	7	Jannero Pargo	2	Eric Dampier	11

Cluster 7 Two-Way Players/Primary Defenders	# of Apps	Cluster 8 Bench Role Players	# of Apps	Cluster 9 Scoring Big Men	# of Apps
Marvin Williams	11	Jason Collins	7	Pau Gasol	15
Erson Ilyasova	10	Thabo Sefolosha	7	Zach Randolph	14
Shawn Marion	10	Anthony Tolliver	7	Dwight Howard	12
Jeff Green	9	Jared Dudley	6	Tim Duncan	11
Andrei Kirilenko	9	Gary Temple	6	Al Jefferson	11
Markieff Morris	9	Derek Fisher	5	David Lee	10
Tayshaun Prince	9	Richard Jefferson	5	Javale McGee	10
Joe Smith	9	Wesley Johnson	5	Greg Monroe	10
Gerald Wallace	9	DeShawn Stevenson	5	Kevin Garnett	9
Tony Allen	8	Solomon Hill	5	LaMarcus Aldridge	9

left of Figure 5.5) shows players like Rajon Rondo, Ricky Rubio, and Steve Nash. These players are well-known as 'pass-first' guards. They are generally high in assists, steals, and turnovers, but are not commonly the leading scorer on their team. Andre Miller (`https://www.basketball-reference.com/players/m/millean02.html`) spent a total of 15 seasons in this 'mega-cluster'. He led the entire league in assists in the 2001-2002 season with 10.9. We will call this position **Pass-First Guards**.

5.3.6 Cluster 3: Superstars

Cluster 3 (light green cluster in top left of Figure 5.4 and yellow-green cluster on far left of Figure 5.5) contains players like LeBron James, Kobe Bryant, and Kevin Durant, so these are clearly the 'superstar' players. These players have the ball in their hands very frequently when they are on the court, and they are high scorers from inside and

Table 5.7: Players with highest percentage of career in each 'mega-cluster'

Cluster 1 Score-First Guards	% of Career	Cluster 2 Pass-First Guards	% of Career	Cluster 3 Superstars	% of Career
Courtney Alexander	75%	Tim Frazier	100%	Lebron James	100%
Jordan McRae	71%	Travis Best	100%	Dwyane Wade	100%
Jimmer Fredette	67%	Tyler Ennis	100%	Kobe Bryant	100%
Trey Burke	63%	Avery Johnson	100%	Russell Westbrook	100%
Allan houston	60%	Robert Pack	100%	Allen Iverson	92%
Latrell Sprewell	60%	Fred Vanvleet	100%	Kevin Durant	91%
CJ McCollum	57%	Jerian Grant	100%	Kyrie Irving	89%
Andrew Wiggins	57%	Andrew Harrison	100%	Derek Rose	83%
Rodney Rogers	57%	Ricky Rubio	89%	James Harden	82%
Tim Hardaway Jr.	50%	Rajon Rondo	87%	Devin Booker	80%
Cluster 4 Bench Perimeter Scorers	**% of Career**	**Cluster 5 Miscellaneous/Transient Players**	**% of Career**	**Cluster 6 Defensive Big Men**	**% of Career**
Jon Barry	100%	Byron Mullens	33%	Shawn Bradley	100%
Glen Rice	100%	Dominic McGuire	25%	Dikembe Mutombo	100%
Walter McCarty	100%	Yakhouba Diawara	25%	Greg Ostertag	100%
Chris Whitney	100%	Bryce Drew	25%	Bismack Biyombo	100%
Rick Fox	100%	Henry Ellenson	25%	Brendan Haywood	100%
Damon Jones	89%	Tremaine Fowlkes	25%	Brian Skinner	100%
Wesley Person	88%	Pops Mensah-Bonsu	25%	Miles Plumlee	100%
Daequan Cook	86%	Adam Morrison	25%	Etan Thomas	100%
Pat Garrity	86%	Randolph Morris	25%	Steven Hunter	100%
Mirza Teletovic	83%	Jeremy Pargo	25%	Bo Outlaw	100%
Cluster 7 Two-Way Players/Primary Defenders	**% of Career**	**Cluster 8 Bench Role Players**	**% of Career**	**Cluster 9 Scoring Big Men**	**% of Career**
Derrick Brown	100%	Mardy Collins	80%	Boban Marjanovic	100%
Landry Fields	100%	Patric McCaw	80%	Julius Randle	100%
KJ McDaniels	100%	Dorian Finney-Smith	75%	Karl Anthony Towns	100%
Andrew Nicholson	83%	Raymond Livingston	75%	Anthony Davis	88%
Alonzo Gee	80%	Jake Layman	75%	Greg Monroe	83%
Justin Anderson	80%	Marquis Teague	75%	Jusuf Nurkic	83%
Joffrey Lauvergne	80%	Rashad Vaughn	75%	Maurice Speights	82%
Trey Lyles	80%	James Anderson	67%	Montrezl Harrell	80%
Donatas Motiejunas	80%	Solomon Hill	63%	Willy Hernangomez	80%
Maurice Harkless	78%	Quinton Ross	63%	Al Jefferson	79%

outside. If we look at the points leaders throughout the 20 NBA seasons (https://www.basketball-reference.com/leaders/ptsyearly.html), we see players like LeBron James, Kobe Bryant, Allen Iverson, and Kevin Durant. They appeared in this 'mega-cluster' 17, 16, 12, and 11 times, respectively. We will call this position the **Superstars**.

5.3.7 Cluster 4: Bench Perimeter Scorers

Cluster 4 (green cluster in top right of Figure 5.4 and bright green cluster in middle of Figure 5.5) includes key players such as Kyle Korver, Derek Fisher, James Jones, and Brent Barry. These players shoot a high percentage from the three-point line and usually come off the bench. Their primary role is to provide a spark off the bench with three-pointers and defensive hustle. Damon Jones (https://www.basketball-reference.com/players/j/jonesda01.html) appeared 8 times in this cluster, and spent 89% of his career in this cluster. We can also see that for his most of his career he averaged more three-point attempts per game than two-point attempts. We will call this position **Bench Perimeter Scorers**.

5.3.8 Cluster 5: Miscellaneous/Transient Players

Cluster 5 (forest green cluster in right middle of Figure 5.4 and teal cluster in middle of Figure 5.5) appears to be the cluster with the least clarity. No player in the 20-year span appears in this cluster more than twice. This grouping appears to comprise of miscellaneous players. They may have played inconsistent minutes throughout the season, or barely breached the minimum threshold for minutes played to be included. Taking a look at Earl Barron's career statistics at `https://www.basketball-reference.com/players/b/barroea01.html`, we can see that he played for seven different teams in his eight seasons, and he even played overseas during the 2008-2009 season. The most games he played in a season was 46 in the 2007-2008 season, which is exactly half of all possible games for that season. We can also look at a player like Jason Collins (`https://www.basketball-reference.com/players/c/collija04.html`), who spent his first eight seasons or so with the New Jersey Nets. He played in the majority of the games during that span, so he likely appeared in a different 'mega-cluster' during that time, but his last six seasons he spent with five different teams. This adds to our assertion that this cluster is for transient players who bounce around from team to team and show very little consistency in their performances. We will call this position **Miscellaneous/Transient Players**.

5.3.9 Cluster 6: Defensive Big Men

Cluster 6 (light blue cluster in bottom middle of Figure 5.4 and light blue cluster in top right of Figure 5.5) contains players like Ben Wallace, Shawn Bradley, and Dikembe Mutombo. These players' primary role is to play defense and rebound the ball, and their only field goals will be high-percentage shots at or near the rim. Many of the players in this category were/are well-known for their rebounding and interior defense. Ben Wallace (`https://www.basketball-reference.com/players/w/wallabe01.html`) led the entire league in total rebounds twice and blocks once in his career, while Dikembe Mutombo (`https://www.basketball-reference.com/players/m/mutomdi01.html`) led the league in total rebounds twice and blocks three times throughout his career. We will call this position **Defensive Big Men**.

5.3.10 Cluster 7: Two-Way Playeres/Primary Defenders

Cluster 7 (royal blue cluster in middle of Figure 5.4 and royal blue cluster in middle right of Figure 5.5) has many notable wing defenders, such as Shawn Marion, Andrei Kirilenko, and Tayshaun Prince. These players frequently play the traditional 'small forward' position and often take the most difficult defensive assignment. Andrei Kirilenko averaged more than one block and more than one steal for almost every season of his career, and led the league in blocks in the 2004-2005 season. These players are known for their quickness and length, and they don't normally take a high volume of shots. We will call this position **Two-Way Players/Primary Defenders**.

5.3.11 Cluster 8: Bench Role Players

Cluster 8 (purple cluster in right middle of Figure 5.4 and purple cluster in bottom right of Figure 5.5) is similar to Cluster 5 in that the same players don't consistently get classified in this group. Three players spent 7 of the possible 20 seasons in this 'mega-cluster', namely Jason Collins, Thabo Sefolosha, and Anthony Tolliver. These players' points per game averages for their entire career were 3.6, 5.7, and 6.1, respectively. These are very low averages, so these players were not counted on for scoring. They mostly came off the bench and likely played a very minor role on the team when they were in this 'mega-cluster.' We will call this position **Bench Role Players**.

5.3.12 Cluster 9: Scoring Big Men

Finally, Cluster 9 (pink cluster in bottom left of Figure 5.4 and pink cluster in top right of Figure 5.5) contains players like Tim Duncan, Kevin Garnett, and Anthony Davis. These are well-known 'scoring big men'. Each of these players were around seven feet tall, and average more than 20 points per game any given season. It is also notable that Tim Duncan won the Most Valuable Player award in the 2001-2002 and 2002-2003 seasons, and Kevin Garnett won the award in the 2003-2004 season. These players were the focal points of their teams and the entire offense generally ran through them. We will call this position **Scoring Big Men**.

This type of analysis and characterization can be conducted more thoroughly than the top 10 player lists to determine the uniqueness of each of the nine 'mega-clusters'.

5.3.13 Individual Player Tracking

In addition to viewing the most frequently occurring players in each 'mega-cluster', we can track an individual player's position evolution from season to season. For this example, we will examine Stephen Curry's career from his rookie season in 2009-2010 to the 2019-2020 season.

Table 5.8: Stephen Curry's 'mega-cluster' position by season

Season	mega-cluster
2009-2010	Superstars
2010-2011	Pass-First Guards
2011-2012	Superstars
2012-2013	Pass-First Guards
2013-2014	Score-First Guards
2014-2015	Superstars
2015-2016	Superstars
2016-2017	Superstars
2017-2018	Superstars
2018-2019	Superstars
2019-2020	Superstars

Table 5.8 shows Stephen Curry's 'mega-cluster' for each of the 11 seasons since his rookie year. We can see that in his first five seasons he alternated between the **Superstar**, **Pass-First Guard**, and **Score-First Guard** positions. In these first seasons preceding Curry's first of two MVP awards in 2014-2015, he was not as dominant of a player and he likely shared a lot of characteristics of the **Pass-First Guards** and the **Score-First Guards**. He had the ability to score in bunches at times as an elite outside shooter, but he also played the traditional Point Guard role running the offense and distributing to other

players. It is possible that Curry hovered between the overlaps of the three clusters in the top of Figure 5.4. We would expect there to be fringe players in each of these ‘mega-clusters’ as they are transitioning roles or developing their skills, especially in the early years of their career.

This method of examining individual players’ positions over the course of their career can provide insights into their development and evolution.

5.4 Clustering All Years Combined

The last method we want to consider involves clustering all players combined over the 20 NBA seasons. While *mega-clustering* is the focus of the results, it can still be informative to compare our ‘mega-clusters’ to the nine clusters obtained from performing the same hierarchical clustering method on all seasons combined. Like the *mega-clustering* method, each unique combination of player, season, and team is considered a unique data point to be partitioned. The major difference between these two approaches is that we are partitioning players instead of partitioning each season’s nine clusters. With *mega-clustering*, we partitioned each season’s nine clusters based on their ‘highs’ and ‘lows’, whereas with this combined clustering method, we are clustering all players based on their scaled statistics.

Table 5.9: Combined clustering notable players

CLUSTER 1	Seasons in Cluster	Total Seasons	PCT
Mo Williams	15	15	100%
JJ Barea	14	14	100%
Steve Nash	14	14	100%
Tyreke Evans	11	11	100%
Ben Gordon	11	11	100%
Darren Collison	10	10	100%
Sam Cassell	9	9	100%
Kemba Walker	9	9	100%
Trey Burke	8	8	100%
Jordan Clarkson	8	8	100%

CLUSTER 2	Seasons in Cluster	Total Seasons	PCT
Rajon Rondo	15	15	100%
Eric Maynor	8	8	100%
Cameron Payne	6	6	100%
Lorenzo Brown	5	5	100%
Mardy Collins	5	5	100%
TJ McConnel	5	5	100%
Keith McLeod	5	5	100%
Pablo Prigioni	5	5	100%
Chris Childs	4	4	100%
Shane Larkin	4	4	100%

CLUSTER 3	Seasons in Cluster	Total Seasons	PCT
Zach Randolph	18	18	100%
Al Jefferson	14	14	100%
DeMarcus Cousins	10	10	100%
Kenneth Faried	9	9	100%
Nikola Vucevic	9	9	100%
Anthony Davis	8	8	100%
Ike Diogu	7	7	100%
Boban Marjanovic	7	7	100%
Jusuf Nurkic	6	6	100%
Willy Hernangomez	5	5	100%

CLUSTER 4	Seasons in Cluster	Total Seasons	PCT
Maurice Harkless	9	9	100%
Kelly Olynyk	7	7	100%
Stacey Augmon	6	6	100%
Nemanja Bjelica	5	5	100%
Landry Fields	5	5	100%
Yi Jianlian	5	5	100%
Terrence Jones	5	5	100%
Thon Maker	5	5	100%
KJ McDaniels	5	5	100%
Derrick Brown	4	4	100%

CLUSTER 5	Seasons in Cluster	Total Seasons	PCT
Jamison Brewer	2	4	50%
Elliot Williams	2	5	40%
Bruno Caboclo	2	6	33%
Linton Johnson	2	6	33%
DeAndre Liggins	2	6	33%
Dominic McGuire	2	8	25%
Michael Ruffin	2	8	25%
Tariq Abdul-Rahad	1	4	25%
Ron Baker	1	4	25%
Yakhouba Diawara	1	4	25%

CLUSTER 6	Seasons in Cluster	Total Seasons	PCT
Marco Belinelli	14	14	100%
Wayne Ellington	13	13	100%
Troy Daniels	9	9	100%
Wesley Person	8	8	100%
Eric Piatkowski	8	8	100%
Joe Harris	5	5	100%
Davis Bertans	4	4	100%
Seth Curry	4	4	100%
Rudy Fernandez	4	4	100%
Tim Hardaway	4	4	100%

CLUSTER 7	Seasons in Cluster	Total Seasons	PCT
Bruce Bowen	9	9	100%
Tony Snell	6	7	86%
Chris Johnson	5	6	83%
Hubert Davis	4	5	80%
Bryon Russell	4	5	80%
Dorian Finney-Smith	3	4	75%
Rashad Vaughn	3	4	75%
James Young	3	4	75%
Iman Shumpert	8	11	73%
Alan Anderson	5	8	63%

CLUSTER 8	Seasons in Cluster	Total Seasons	PCT
Brendan Haywood	14	14	100%
Ryan Hollins	13	13	100%
DeAndre Jordan	13	13	100%
Ben Wallace	13	13	100%
Joel Anthony	11	11	100%
Ed Davis	11	11	100%
Andris Biedrins	10	10	100%
Dikembe Mutombo	10	10	100%
Ronny Turiaf	10	10	100%
Adonal Foyle	9	9	100%

CLUSTER 9	Seasons in Cluster	Total Seasons	PCT
Kevin Durant	12	12	100%
LeBron James	16	17	94%
Kobe Bryant	14	16	88%
Carmelo Anthony	15	18	83%
Russell Westbrook	10	12	83%
James Harden	8	11	73%
Allen Iverson	9	13	69%
Correy Maggete	9	13	69%
Dirk Nowitzki	13	19	68%
Dwyane Wade	11	17	65%

While the same in-depth exploration won't be performed on this clustering method as with the 'mega-clusters', we still want to verify that we achieve similar results. Table 5.9 displays the most frequently occurring players in each of the nine clusters for the 20 NBA seasons. For example, Mo Williams appears in Cluster 1 for all 15 seasons in which he played during this 20-season span.

Beginning with Cluster 1, we can see players like Mo Williams, Steve Nash, Kemba Walker, and Jordan Clarkson. These players are scoring guards, which is similar to the **Score-First Guards** 'mega-cluster'.

With Cluster 2, we can see players like Rajon Rondo, Cameron Payne, and TJ McConnel. These players are 'pass-first' guards who will have high assist counts. This is similar to the **Pass-First Guards** 'mega-cluster'.

Cluster 3 shows Zach Randolph, DeMarcus Cousins, Nikola Vucevic, and Anthony Davis. These players are 'scoring big men' who often play in the post, but also have the ability to shoot from the outside. This is similar to the **Scoring Big Men** 'mega-cluster'.

Cluster 4 shows players like Maurice Harkless, Kelly Olynyk, and Stacey Augmon. These players stretch the floor with defense, but generally do not score at a high volume. They appear to be similar to the **Two-Way Players/Primary Defenders** 'mega-cluster'.

Cluster 5 appears to have a lot of less-recognized players and no player spends more than two seasons in this cluster. This is strikingly similar to the **Miscellaneous/Transient Players** 'mega-cluster'.

Cluster 6 shows players like Marco Belinelli, Wesley Person, Joe Harris, and Seth Curry. These players can score in bunches, but generally come off the bench. This cluster appears to mirror the **Bench Perimeter Scorers** 'mega-cluster'.

Cluster 7 is another ambiguous cluster that appears to capture a lot of bench role players who don't score at a high volume. This is similar to the **Bench Role Players** 'mega-cluster'.

Cluster 8 shows players like Ben Wallace and Dikembe Mutombo. These players were discussed along with the **Defensive Big Men** 'mega-cluster', and the other top players appear to be consistent with this description of players with high block and rebound totals.

Cluster 9 is clearly the superstar cluster with players like Kevin Durant, LeBron James, and Kobe Bryant. This bears a strong resemblance to the **Superstars** 'mega-cluster'.

CHAPTER 6

Discussion

This MS Thesis is comprised of two major components: (1) The selection of **nine** as the preferred cluster number for NBA player positions, and (2) the visualization and analysis of these new player positions. This chapter discusses the results from Chapters 4 and 5 while comparing the results with those achieved through previous research.

6.1 Number of Clusters Selection

The decision to proceed with nine player clusters weighed heavily on two pillars: (1) the `NbClust` index selections in R, and (2) the influences of previous work. This choice of nine separate groups was further validated through various visualizations and dimensionality reduction techniques. We will discuss these two pillars and their importance, followed by a summary of the visualization results in the next section.

The `NbClust` index selections displayed in Figure 4.2 show a jump at six, nine, twelve, and fifteen. As discussed in Appendix B, if we move the starting point between two and five clusters and the end point between twelve and twenty, the histogram trends downward before climbing back up at the end. With this caveat in mind, this MS thesis aims to strike a balance between describing players' abilities in further detail without creating clusters with little to no meaning. Nine clusters provides this 'happy medium'.

In addition, we can see the work of Kalman and Bosch (2020) and Jyad (2020) both selecting nine as the optimal number of clusters through differing methods. Kalman and Bosch (2020) analyzed 10 NBA seasons, beginning with the 2009-2010 season and concluding with the 2017-2018 season, and arrived at the decision of nine clusters through the `mclust` R package (see Section 3.3.10). Jyad (2020) used hierarchical clustering on the 2018-2019 NBA season and used an 'elbow plot' that tracks the amount of variance explained by the number of clusters. Similar to the WSS plot discussed in Section 3.2.1, the goal of an elbow plot is to find the elbow of the curve where variation starts to level off. Jyad (2020) observed the elbow effect at two, six, and nine clusters, and arrived at the conclusion that nine clusters made the most sense as the goal is to be able to describe players with increasing precision and detail.

Alagappan (2012), on the other hand, used a visual technique called topological data analysis to observe groups in the 2010-2011 NBA season. This method involves normalizing data points and displaying results in a type of map, where the user can determine what branches constitute separate and distinct partitions. Similar to the other methods discussed in this MS thesis, the selection of the optimal number of clusters in this case is largely up to user preference and individual interpretation. Alagappan (2012) selected thirteen as the optimal number of clusters. This selection of a large number allows for more in-depth discussion of single player differences, especially since the author only considered a single season.

6.2 Comparison of Visualization Techniques

In this section, we will compare the various visualization methods and how they provide clarity on the player clusters. We will specifically explore how each of these methods displays the *distinctiveness* of the clusters.

6.2.1 Single Season Visualization

When clustering an individual NBA season, the application of multiple dimensionality reduction methods was extremely useful and insightful. In Figure 4.8, we can see that the tSNE method shows cluster distinction that PCA fails to capture. For example, the PCA plot shows the **Bench Perimeter Scorers** cluster (Cluster 6) on the left-hand side being quite spread out, while the tSNE plot shows this cluster seemingly tightly packed in the top left corner, with the exception of three points, located at approximately (-20, -10), (-15, 5), and (20, 15). It is unclear immediately whether or not these 'stray' points are related to the 6's in the PCA plot around (-3, -3) and (-4, 5), but this could be investigated further.

While the PCA and tSNE projections display only a two-dimensional view of very high-dimensional data, the relative compactness of the points is encouraging. In the PCA plot, the **Score-First Guards** cluster (Cluster 1) overlaps heavily with the **Bench Role Players**, **Pass-First Guards**, and **Two-Way Players/Primary Defenders** clusters (Cluster 2, 4, and 5, respectively). When we look at the **Score-First Guards** cluster (Cluster 1) in the tSNE plot, we can see very little overlap on the edges with Clusters 2, 4, and 5.

The **Bench Role Players** (Cluster 2) provide an interesting exception in that they appear more compact in the PCA plot than the tSNE plot. The tSNE plot seems to show these players on the outer boundaries of many other clusters, while the PCA plot shows the **Bench Role Players** right in the middle of the other players.

Finally, the **Superstars** cluster (Cluster 9) in the PCA plot shows good separation from the rest of the data points, but the points are quite spread out. When we look at this same cluster in the tSNE plot, we can see that the data points appear tightly packed together, with the exception of one point located at around (10, 10). In general, it appears that the tSNE method does a better job of displaying the *distinctiveness* of the nine player clusters.

The PHATE method also shows good distinctiveness of data points in their respective clusters. We notice in Figure 4.10 that the **Superstars** cluster (Cluster 9) stands out in the top right of the plot. This is similar to the PCA and tSNE plots. We can also see from the PHATE plot that the **Defensive Big Men** and **Interior Big Men** clusters (Cluster 7 and 8) are located in the bottom right corner of the plot with a small amount of overlap. In this case, the PHATE plot performs similarly to the tSNE and PHATE plots, since both of these visualizations show Cluster 2 and Cluster 4 with heavy overlapping with each other and other positions. It is very possible that these overlaps constitute players who are ‘fence-sitters’, meaning that they were very close to being placed in a different cluster.

The implementation of GGobi for viewing the nine player clusters allows for in-depth exploration of all possible projections. The full list of players in each cluster for the 2000-2001 NBA season can be found by accessing the following link within the GitHub repository:

https://github.com/ahed1194/MS_Thesis/blob/main/Player_Cluster/player_cluster0001_scaled.csv. The process of running and pausing the grand tour provided additional insights and views not available through static clustering. Not every projection can capture every cluster as distinct and compact simultaneously, therefore it became very useful to pause the tour and view projections that capture one or several clusters in 'a good light'.

Figure 4.11 shows a particular point in the projection where we see clear distinctions of certain clusters. We can see that the **Superstars** cluster (Cluster 9 - Large Yellow +) is well-separated across the bottom of the plot. This cluster is easily distinguished in the other dimensionality reduction methods employed. Another cluster that shows clear distinction in this figure is the **Defensive Big Men** (Cluster 7 - Small Yellow Circles). It appears that many variable categories carry a similar amount of weight in this projection, including two-point shots ('X2'), three-point shots ('X3'), free-throws ('FT'), and points ('PT').

We can also see some clusters in this projection that overlap heavily with one or more other clusters. For example, the **Score-First Guards** cluster (Cluster 1 - Large Purple +'s) and the **Pass-First Guards** (Cluster 4 - Large Blue Squares) show considerable overlap. One must examine other projections or observe the points actively moving throughout the tour in order to see their distinction. Figure 4.12 gives an example where the **Score-First Guards** (Cluster 1 - Large Purple +'s) and the **Pass-First Guards** (Cluster 4 - Large Blue Squares) show some distinction.

This high-dimensional visualization technique is extremely effective for complex data. Alagappan (2012) utilized a visual technique called topological data analysis that provides robustness to noisy data. This technique provides insights beyond the capabilities of static 2D plots. When analyzing NBA player data with a wide range of measurements and statistics, the ability to customize the visualization to capture the unique behaviors of the different clusters is highly insightful.

6.2.2 Mega-Clustering Visualization

Taking a look at the dendogram in Figure 5.3, we can see which 'mega-clusters' are considered the most distinct and which clusters could have potentially been combined. The first 'split' (or, equivalently, the final 'combination') distinguishes the **Superstars** (Cluster 3) and the **Scoring Big Men** (Cluster 9) from the rest of the players. These players have the biggest impact on the floor, especially from a scoring and ball-handling standpoint. The offense runs through these players, and they both tend to have the ball in their hands more than any other players.

The second 'split' occurs as the **Defensive Big Men** (Cluster 6) are separated from the larger group of players. This also makes sense since these players are quite unique in that they are defined by their defensive role, whereas most other positions are defined by offensive metrics such as shooting and assists.

The third 'split' separates the **Miscellaneous/Transient Players** (Cluster 5) and the **Bench Role Players** (Cluster 8) from the remaining four positions. These two positions are very similar in that their roles are not clearly defined and the players tend to have low and inconsistent minutes.

The fourth 'split' places the **Superstars** (Cluster 3) and the **Scoring Big Men** (Cluster 9) into their own cluster. These two positions were the first to be separated, and they were also the first to become their own clusters. This reiterates their importance and their perceived impact.

The fifth 'split' separates the **Two-Way Players/Primary Defenders** (Cluster 7) and the **Pass-First Guards** (Cluster 2) from the **Score-First Guards** (Cluster 1) and the **Bench Perimeter Scorers** (Cluster 4). This division seems to be based on scoring ability, since the **Score-First Guards** and the **Bench Perimeter Scorers** both have scoring as their primary role and point of impact, while the other two positions are more about adding spacing, defense, and passing to the team.

The sixth 'split' separates the **Score-First Guards** (Cluster 1) and the **Bench Perimeter Scorers** (Cluster 4). This separation is likely due to the latter generally playing less minutes and scoring less points per game than the **Score-First Guards**, who are mostly starters.

The seventh 'split' places the **Two-Way Players/Primary Defenders** (Cluster 7) and the **Pass-First Guards** (Cluster 2) into their own position. This division likely occurs due to the higher assist and turnover totals for the **Pass-First Guards**.

The eight and final 'split' divides the **Miscellaneous/Transient Players** (Cluster 5) from the **Bench Role Players** (Cluster 8). These two positions are clearly the most similar of the nine, as we have mentioned.

Next we will discuss the PCA plot in Figure 5.4, which displays the nine *mega-clusters* across the 20 NBA seasons. We can see four distinct clusters in this projection: the **Superstars** (dark yellow) on the top left, the **Scoring Big Men** (pink) on the bottom left, the **Defensive Big Men** (light blue) on the bottom middle, and the **Two-Way Players/Primary Defenders** (royal blue) in the middle.

We can see that the **Defensive Big Men** and the **Scoring Big Men** on the bottom middle and left of Figure 5.4 appear the most distinct and separate from the other clusters. In the visualization of the clusters generated by Alagappan (2012), the **Paint Protector** and **Scoring Rebounder** players were spread out far to the right, while the ball-handling positions were clustered to the left side. The **Paint Protector** and **Scoring Rebounder** positions from the work of Alagappan (2012) line up well with the **Defensive Big Men** and **Scoring Big Men** positions defined here. The **Superstars** (dark yellow) and the **Score-First Guards** (red) appear to touch in the top left of Figure 5.4. The visualization of Alagappan (2012) showed three positions in close proximity that are similar to these two: **Offensive Ball-Handler**, **Shooting Ball-Handler**, and **Combo Ball-Handler**.

In the top of Figure 5.4, we can see the **Pass-First Guards** (orange) cluster overlapping heavily on its left side with the **Score-First Guards** (red). We can also see the **Pass-First Guards** (orange) position overlapping on its right side with the **Bench Perimeter Scorers** (lime green). The fact that the first two principal components show some heavy overlap of these three clusters is not surprising. Scoring guards and passing guards may be distinguished by certain statistics, but are likely quite similar in other areas. For example, they likely get similar rebounding, stealing, and blocking totals. Similarly, passing guards likely overlap with bench scorers due to lower rebounding totals and blocks.

The remaining two clusters on the right side of Figure 5.4 are the **Bench Role Players** (purple) and the **Miscellaneous/Transient Players** (teal). Again, it is not surprising that these two clusters would heavily overlap in this projection. Both of these positions likely contain players who play significantly lower minutes, and record average to below-average numbers in most statistical categories. Many of these players were likely subject to mid-season trades and were not one of the main rotation players for every game.

For the cases where the 'mega-clusters' overlap in Figure 5.4, we can visually analyze the tSNE method in Figure 5.5 for further clarity and separation. We can see that the four 'mega-clusters' that are well-separated and distinct in Figure 5.4, namely the **Superstars** (dark yellow, top left), **Scoring Big Men** (pink, bottom left), **Defensive Big Men** (light blue, bottom middle), and **Two-Way Players/Primary Defenders** (royal blue, middle), are also distinct in Figure 5.5. For the other five 'mega-clusters', we can see that tSNE does a good job of displaying a view where each cluster is separate and compact. The **Pass-First Guards** (orange), **Score-First Guards** (red), and the **Bench Perimeter Scorers** (lime green) are well-separated on the left side of the tSNE projection. The **Bench Role Players** (purple) and the **Miscellaneous/Transient Players** (teal) are also very far apart in the tSNE projection.

Combining the results of the PCA and tSNE methods is sufficient to view the distinctions of the 'mega-clusters'. Now we will discuss the labeling of these groups in more detail and compare them to other research.

6.3 Mega-Cluster Characterization

We will now explore in more detail the nine 'mega-cluster' positions chosen through this analysis, and compare our positions with those defined in previous research.

While the naming of these updated positions may provide some reference to the traditional player positions, it is important to note that most of these new positions contain players from many different standard positions (see Section 1.1.1). The **Score-First Guards** and **Pass-First Guards** appear to contain mostly Shooting Guards and Point Guards, respectively. The **Superstars** cluster contains many standard positions, such as Point Guards

(Tony Parker & Allen Iverson), Shooting Guards (Kobe Bryant & Dwyane Wade), and Small Forwards (LeBron James & Kevin Durant). The **Defensive Big Men** and **Scoring Big Men** 'mega-clusters' contain a mix of Power Forwards and Centers. The **Miscellaneous & Transient Players** and the **Bench Role Players** each appear to contain a fairly even mix of all five standard positions. This analysis highlights the ambiguity of standard position classification and the need for updated positions.

Referring back to our introductory examples in Section 1.1.2, we listed several players and their traditional positions. Stephen Curry and John Stockton, who are both classified traditionally as Point Guards, are placed into different 'mega-clusters' in all seasons for which we have conducted this analysis. We can see from Table 5.8 that Stephen Curry is mostly classified as a **Superstar**. While we only have the final three years of John Stockton's career in our 20-year span, he is classified as a **Bench Role Player** in all three years. We can be highly confident, however, that had we clustered Stockton during his prime playing years, he would have been classified as a **Pass-First Guard**. Even with the years for which we have data for these players, it is clear that they play very different roles on the court.

Michael Jordan and Kobe Bryant are both traditionally classified as Shooting Guards. Through the *mega-clustering* analysis, Kobe Bryant spent all 17 seasons of the 20-year timespan in the **Superstars** 'mega-cluster' (see Tables 5.6 and 5.7). Michael Jordan only played two seasons in our time window, and he was classified as a **Superstar** in 2001-2002 and a **Scoring Big Man** in 2002-2003. While the latter classification may seem incorrect, we must consider that Jordan averaged less than one three-point attempt per game in his final two seasons. Most of his scoring came from post-up and turnaround shots, similar to what we would expect from **Scoring Big Men** players. We can be confident that in Michael Jordan's prime years in the 1980's and 1990's, he would be classified as a **Superstar**, as he is considered by many to be the greatest basketball player of all time.

Another example from Section 1.1.2 illustrated the dynamic roles of two players classified as Small Forwards: LeBron James and Kevin Durant. We see in Tables 5.6 and 5.7 that LeBron James spent 17 seasons in the **Superstar** 'mega-cluster' (100% of his career), and

Kevin Durant spent 11 seasons (91% of his career) in the **Superstar** 'mega-cluster'. This is a fine example of how the updated position classification gives more value and context to these key players.

Two well-known Power Forwards mentioned in Section 1.1.2 are Tim Duncan and Karl Malone. Tim Duncan spent a total of 11 seasons in our 20-season span in the **Scoring Big Men** 'mega-cluster'. This is not surprising since Duncan is regarded as one of the great mid-range and post-up scorers in NBA history. Karl Malone was past the prime of his career by the debut of the 2000-2001 NBA season, and he retired after the 2003-2004 season, but his final four season's 'mega-clusters' were **Superstar**, **Superstar**, **Scoring Big Man**, and **Two-Way Player/Primary Defender**. Note that his final season was spent with a new team, the Los Angeles Lakers, where Shaquille O'Neal was established as the primary interior scorer. This provides an example of how player roles and positions can change when they join a new team. Karl Malone is currently the third all-time leading scorer in NBA history (as of April 2022), so it would make sense that in his prime playing years he could be classified as either a **Superstar** or a **Scoring Big Man**.

Finally, Kareem Abdul-Jabbar and Shaquille O'Neal were presented in Section 1.1.2 as examples of Centers. Since the 2000-2001 NBA season, O'Neal was classified as a **Scoring Big Man** seven times and a **Superstar** four times. While we do not have clustering results for Kareem Abdul-Jabbar, we can look at his career statistics at `https://www.basketball-reference.com/players/a/abdulka01.html` and see that he led the league in points per game twice in his career, and regularly averaged more than 25 points per game. This would place him with the elite superstars of the league in scoring averages every year. We can also see his dominant presence inside through his high rebounding and blocking totals. It is very likely, given this information, that he would have been classified similarly to O'Neal for most of his career, either as a **Scoring Big Man** or a **Superstar**.

Table 6.1 displays how our nine 'mega-clusters' overlap with the clusters defined by previous authors. The clusters characterized by the other researchers are linked with the 'mega-cluster' that best matched their description and example players.

Table 6.1: Comparing 'mega-clusters' to previous work

Hedquist	Jyad (2020)	Kalman and Bosch (2020)	Alagappan (2012)
2000-2001 to 2019-2020 NBA Season Hierarchical Clustering	2018-2019 NBA Season Hierarchical Clustering	2019-2010 to 2018-2019 NBA Season Model-Based Clustering	2010-2011 NBA Season Topological Data Analysis
Score-First Guards	Elite 3 Point Shooters 3 Level Shooters	High Usage Guard Three Point Shooting Guard	Combo Ball-Handler Shooting Ball-Handler
Defensive Big Men	Traditional Big Men	Traditional Center	Paint Protector
Bench Role Players	Role Players	Versatile Role Player	Role Player
Superstars	Elite All-Stars	Ball-Dominant Scorer	One-of-a-Kind NBA 1st Team
Pass-First Guards	Decent Ball Handlers	Floor General	Offensive Ball-Handler Defensive Ball-Handler
Bench Perimeter Scorers	3 and D Players	Stretch Forward	3-Point Rebounder
Miscellaneous/Transient Players			Role-Playing Ball-Handler
Two-Way Players/Primary Defenders	Two-Way Perimeter Players		NBA 2nd Team
Scoring Big Men	Elite Modern Big Men	Skilled Forward Mid-Range Big	Scoring Rebounder Scoring Paint Protector

We can see that the **Score-First Guards** 'mega-cluster' is broken up into two clusters by the other three groups of researchers. Jyad (2020) and Kalman and Bosch (2020) separated them by their three-point shooting abilities, while Alagappan (2012) separated them based on their defensive abilities.

All of the researchers have a clearly-defined **Defensive Big Men** cluster for tall players who protect the paint and are not high scorers. Another clearly-defined cluster is the **Bench Role Players** cluster. Every researcher distinguishes a cluster for players who displayed average marks in most statistical categories. These players were generally lower scorers coming off the bench.

The **Superstars** 'mega-cluster' is clearly defined in our analysis, and we can see that Jyad (2020) and Kalman and Bosch (2020) both had this 'elite' cluster in their analysis. Alagappan (2012) determined that there were two clusters within this superstar category: **One-of-a-kind** and **NBA 1st Team**. It is possible that if we had increased our number of clusters from nine to thirteen, we may have seen a similar split within the **Superstars** cluster. However, it is also quite plausible that we would see splits in the **Miscellaneous/-Transient Players** cluster (Cluster 5; teal cluster), the **Defensive Big Men** cluster (Cluster 6; light blue cluster) or the **Bench Role Players** cluster (Cluster 8; purple cluster), since they appear to be starting to separate in Figure 5.5.

It is overall highly encouraging to see considerable overlapping of player positions considering that all of these methods used different season ranges, clustering algorithims, and/or optimal cluster numbers. This MS thesis employs a 20-year span, so it is very likely that some of these player positions have evolved and adapted over the past two decades. When we combine two decades of player data, we achieve a more holistic view of the important distinctions between players. As we can see with the single-season view in Section 5.2.1, there are slightly different position characteristics when compared to the *mega-clustering* over 20 years combined. Looking specifically at the 2000-2001 discussion in Section 5.2.1, we can see that there is no **Miscellaneous/Transient Players** cluster, but there is an additional position for big men, namely the **Interior Big Men** cluster. It is possible that there was a greater and more diverse usage for big men on the court, especially since the three-pointer wasn't shot at the volume that it is today.

In general we would expect each season to play out differently and for different players rise to the top. This inevitably causes teams and players to react accordingly and adjust strategies. Recent years have seen an incredible surge in three-point shot attempts per game, so we would expect some player positions to evolve accordingly.

Performing hierarchical clustering on a single season or a span of several seasons provides key insights into player behaviors and usage. It can also display the evolution of player knowledge and abilities. Understanding how to categorize players more accurately opens up a wide range of applications and possibilities that will be discussed in the following chapter.

CHAPTER 7

Conclusion and Future Work

In this MS thesis, we have responded to the research question related to how players can be classified to give more meaning to their positions. We have outlined new methods for assigning and analyzing player positions based on their abilities and performances, rather than an assumed role based on their height or weight. In this concluding chapter, we will briefly discuss the implications for practice, and we will outline some possible directions for future research that can be conducted.

7.1 Implications

The first major implication of our study stems from the selection of the optimal number of player positions as well as the practical reasons for choosing a higher number of clusters than five. The decision of a precise number of player positions is critical to understanding player roles at the right level of detail. We have introduced several methods to numerically and visually justify the higher cluster selection.

This research highlights the value and utility of using hierarchical clustering to partition players. The tiered nature of this clustering algorithm in a basketball context allows for in-depth interpretation of player similarities and differences. The 'splitting' and 'combining' of player positions as we move up and down the hierarchy are easily interpretable for basketball minds as to how players can adapt to different roles or evolve and improve their skills.

Another important contribution of our research surrounds the many visualization options available for viewing and comparing player positions. While this MS thesis does not cover many of the more complicated and intensive visualization methods, we have still provided valuable insights into the vast possibilities for analyzing NBA players. The NBA player clustering is meant to be understood and applied specifically in a sports context, and the visualizations implemented are invaluable tools for sports researchers and fanatics alike.

Another major implication of this study is the introduction of *mega-clustering* across many NBA seasons. While there are certainly applications for this method outside of the sports community, we are primarily interested in how we are able to track individual players over the course of their career. Being able to track player evolution and even position evolution through *mega-clustering* is an incredibly powerful tool for managers and coaches as they plan and adjust their strategies for drafting and fielding lineups.

A final contribution of this MS thesis is found in the NBA individual player data as well as the seasonal lineup data scraped from https://www.basketball-reference.com/. Users wishing to analyze this data will see the utility in the pre-scraped NBA tables included on the Github repository (https://github.com/ahed1194/MS_Thesis), particularly those individuals without a technical background or those without a subscription to the *Stathead* (https://stathead.com/basketball/) portion of the website (see Section 2.4). These tables coupled with the analysis conducted in this research provide a solid foundation for many different paths of exploration and discovery.

7.2 Future Work

This MS thesis creates many opportunities for future research and analysis. The contribution of the player and lineup tables will allow for many types of analysis related to team performance and potential.

The player clusters can be used to perform various predictive regression and machine learning models to determine the optimal lineups for a given team. The exploration of historical lineups and the composition of teams based on these new positions can help to better predict performances in the future. The lineup tables included within the GitHub repository (https://github.com/ahed1194/MS_Thesis/tree/main/Lineup_Data) will prove extremely useful for this type of predictive analysis. One can find many examples of lineups that underperformed despite being filled with many well-known and talented players. One can also find historical examples of lineups filled with more underrated players who exceeded expectations. These new player positions can be factored in to the modeling process to determine which player positions have the largest impact on game outcomes. The reader may consult the work of Holland (2020), Ahmadalinezhad et al. (2019), Pelechrinis (2019), and Perera et al. (2016) for examples of how to analyze lineup data and predict performances.

This research also encourages further analysis of the player positions from year to year. As we discovered in our comparison of the single-season clusters and the 'mega-clusters', the player clusters for a given season will vary from the combined clustering. An analysis of how positions are evolving will help coaches and managers to determine which positions are becoming obsolete or are beginning to merge with others. Each year may also require a different number of player clusters to avoid overgeneralizing certain positions. This type of analysis over the course of many seasons can also be applied to individual players to view their position changes over the course of their career.

Finally, this MS thesis offers the opportunity to expand the clustering parameters to more complex statistics, including efficient field goal percentage, pace, and win shares. While this research focuses on standard player data, more complex statistics may yield slightly different player positions.

References

Ahmadalinezhad, M., M. Makrehchi, and N. Seward (2019). Basketball Lineup Performance Prediction Using Network Analysis. In *2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 519–524. https://doi.org/10.1145/3341161.3342932.

Alagappan, M. (2012). From 5 to 13: Redefining the Positions of Basketball – Sloan Sports Conference. https://web.math.utk.edu/~fernando/Students/GregClark/pdf/Alagappan-Muthu-EOSMarch2012PPT.pdf.

Asimov, D. (1985). The Grand Tour: A Tool for Viewing Multidimensional Data. *Siam Journal on Scientific and Statistical Computing 6*(1), 128–143. https://doi.org/10.1137/0906011.

Baijayanta, R. (2020). All About Feature Scaling. *Towards Data Science*. https://towardsdatascience.com/all-about-feature-scaling-bcc0ad75cb35.

Basketball Reference (2000-2022). https://www.basketball-reference.com. Last Accessed: October 12, 2020.

Boehmke, B. (2020). Hierarchical Cluster Analysis. *UC Business Analytics*. https://uc-r.github.io/hc_clustering#:~:text=0.3988593%200.8608085%201.864967207-,Hierarchical%20Clustering%20with%20R,cluster%20package%5D%20for%20divisive%20HC.

Borgatti, S. P. (1994). How to Explain Hierarchical Clustering. *Connections 17*(2), 78–80. http://www.analytictech.com/networks/hiclus.htm.

Cai, Y., Y. Chang, and Y. Liu (2019). Multi-omics Profiling Reveals Distinct Microenvironment Characterization of Endometrial Cancer. *Biomedicine & Pharmacotherapy 118*, 109244. https://doi.org/10.1016/j.biopha.2019.109244.

Charrad, M., N. Ghazzali, V. Boiteau, and A. Niknafs (2014). NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set. *Journal of Statistical Software 61*(6), 1–36. https://doi.org/10.18637/jss.v061.i06.

Cook, D., A. Buja, J. Cabrera, and C. Hurley (1995). Grand Tour and Projection Pursuit. *Journal of Computational and Graphical Statistics 4*(3), 155–172. https://doi.org/10.2307/1390844.

Cook, D. and D. F. Swayne (2007). *Interactive and Dynamic Graphics for Data Analysis: With R and GGobi*. Springer-Verlag; New York, NY.

Ferreira, L. and D. B. Hitchcock (2009). A Comparison of Hierarchical Methods for Clustering Functional Data. *Communications in Statistics - Simulation and Computation 38*(9), 1925–1949. https://doi.org/10.1080/03610910903168603.

FIBA (2021). International Basketball Migration Report 2021. https://www.fiba.basketball/documents/ibmr2021.pdf.

Frey, R. (December 15, 2019). Web Scraping Basketball Reference Using R – stackoverflow.com. https://stackoverflow.com/questions/48778493/web-scraping-basketball-reference-using-r.

Galili, T. (2013). K-Means Clustering. *R in Action*. https://www.r-statistics.com/2013/08/k-means-clustering-from-r-in-action/.

Henry, L. and H. Wickham (2020). *purrr: Functional Programming Tools*. R package version 0.3.4. https://CRAN.R-project.org/package=purrr.

Hinton, G. E. and S. Roweis (2002). Stochastic Neighbor Embedding. In *Proceedings of the 15th International Conference on Neural Information Processing Systems*, Volume 15, pp. 857–864. MIT Press; Cambridge, MA. https://dl.acm.org/doi/10.5555/2968618.2968725.

Holland, W. (February 5, 2020). Hacking the NBA, Maximizing DFS Lineups With Machine Learning. *Medium.com*. https://wilsonholland.medium.com/hacking-the-nba-maximizing-dfs-lineups-with-machine-learning-4ce9728712c9.

Hubert, L. and P. Arabie (1985). Comparing Clusters. *Journal of Classification 2*(1), 193–218. https://doi.org/10.1007/BF01908075.

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering 9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55.

Jolliffe, I. T. (1986). *Principal Component Analysis*. Springer, New York, NY. https://doi.org/10.1007/978-1-4757-1904-8.

Jyad, A. (November 16, 2020). Redefining NBA Player Classifications Using Clustering. *Towards Data Science*. https://towardsdatascience.com/redefining-nba-player-classifications-using-clustering-36a348fa54a8.

Kalman, S. and J. Bosch (2020). NBA Lineup Analysis on Clustered Player Tendencies: A New Approach to the Positions of Basketball & Modeling Lineup Efficiency. *MIT Sloan Sports Conference*. https://www.sloansportsconference.com/research-papers/nba-lineup-analysis-on-clustered-player-tendencies-a-new-approach-to-the-positions-of-basketball-modeling-lineup-efficiency.

Kassambara, A. and F. Mundt (2020). *factoextra: Extract and Visualize the Results of Multivariate Data Analyses*. https://CRAN.R-project.org/package=factoextra.

Kaufman, L. and P. J. Rousseeuw (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley and Sons, Inc.; New York, NY.

Kaushik, M. and B. Mathur (2014). Comparative Study of K-Means and Hierarchical Clustering Techniques. *International Journal of Software and Hardware Research in Engineering 2*(6), 93–98. https://ijournals.in/wp-content/uploads/2017/07/IJSHRE-2653.compressed.pdf.

kjytay (2018). Scraping NBA Game Data From basketball-reference.com. *R-bloggers*. https://www.r-bloggers.com/2018/12/scraping-nba-game-data-from-basketball-reference-com/.

Krijthe, J. H. (2015). *Rtsne: T-Distributed Stochastic Neighbor Embedding Using Barnes-Hut Implementation*. https://github.com/jkrijthe/Rtsne.

Larose, D. T. and C. D. Larose (2014). Hierarchical and k-Means Clustering. In *Discovering Knowledge in Data: An Introduction to Data Mining, Second Edition*, pp. 209–227. John Wiley & Sons, Inc. https://doi.org/10.1002/9781118874059.ch10.

Lee, S. (2021). *liminal: Multivariate Data Visualization with Tours and Embeddings*. https://CRAN.R-project.org/package=liminal.

Lee, S., U. Laa, and D. Cook (2020). Casting Multiple Shadows: High-Dimensional Interactive Data Visualisation with Tours and Embeddings. *arXiv*. https://doi.org/10.48550/arXiv.2012.06077.

Maechler, M., P. Rousseeuw, A. Struyf, M. Hubert, and K. Hornik (2019). *cluster: Cluster Analysis Basics and Extensions*. R package version 2.1.0. https://CRAN.R-project.org/package=cluster.

Martín-Fernández, J. D., J. M. Luna-Romera, B. Pontes, and J. C. Riquelme-Santos (2020). Indexes to Find the Optimal Number of Clusters in a Hierarchical Clustering. In F. Martínez Álvarez, A. Troncoso Lora, J. Sáez Muñoz, H. Quintián, and E. Corchado (Eds.), *14th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2019)*, pp. 3–13. Springer International Publishing; Cham, Switzerland. https://doi.org/10.1007/978-3-030-20055-8_1.

McKinney, W. (2010). Data Structures for Statistical Computing in Python. In S. van der Walt and J. Millman (Eds.), *Proceedings of the 9th Python in Science Conference*, pp. 56–61. https://doi.org/10.25080/Majora-92bf1922-00a.

Moon, K. R., D. van Dijk, Z. Wang, S. Gigante, D. B. Burkhardt, W. S. Chen, K. Yim, A. van den Elzen, M. J. Hirn, R. R. Coifman, N. B. Ivanova, G. Wolf, and S. Krishnaswamy (2019). Visualizing Structure and Transitions in High-Dimensional Biological Data. *Nature Biotechnology 37*, 1482–1492. https://doi.org/10.1038/s41587-019-0336-3.

Murtagh, F. and P. Contreras (2017). Algorithms for Hierarchical Clustering: An Overview, II. *WIREs Data Mining and Knowledge Discovery 7*(6), e1219. https://doi.org/10.1002/widm.1219.

Murtagh, F. and P. Legendre (2014). Ward's Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion? *Journal of Classification 31*(3), 274–295. https://doi.org/10.1007/s00357-014-9161-z.

NBA (July 21, 2021). NBA Finals Finishes Up 32 Percent in Viewership vs. 2020 NBA Finals. https://www.nba.com/news/2021-nba-finals-finishes-up-32-percent-in-viewership5.

Pearson, K. (1901). LIII. On Lines and Planes of Closest Fit to Systems of Points in Space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2*(11), 559–572. https://doi.org/10.1080/14786440109462720.

Pelechrinis, K. (2019). LinNet: Probabilistic Lineup Evaluation Through Network Embedding. In U. Brefeld, E. Curry, E. Daly, B. MacNamee, A. Marascu, F. Pinelli, M. Berlingerio, N. Hurley (Eds.), *Lecture Notes on Computer Science 11053,* 20–36. Springer, Cham, Switzerland. https://doi.org/10.1007/978-3-030-10997-4_2.

Perera, H., J. Davis, and T. B. Swartz (2016). Optimal Lineups in Twenty20 Cricket. *Journal of Statistical Computation and Simulation 86*(14), 2888–2900. https://doi.org/10.1080/00949655.2015.1136629.

R Core Team (2021). *R: A Language and Environment for Statistical Computing.* Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Rand, W. M. (1971). Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association 66*(336), 846–850. https://doi.org/10.1080/01621459.1971.10482356.

Reimann-Philip, U., M. Speck, C. Orser, S. Johnson, A. Hilyard, H. Turner, A. J. Stokes, and A. L. Small-Howard (2019). Cannabis Chemovar Nomenclature Misrepresents Chemical and Genetic Diversity; Survey of Variations in Chemical Profiles and Genetic Markers in Nevada Medical Cannabis Samples. *Cannabis and Cannabinoid Research 5*(3), 215–230. http://doi.org/10.1089/can.2018.0063.

Rokach, L. and O. Maimon (2005). Clustering Methods. In O. Maimon and L. Rokach (Eds.), *Data Mining and Knowledge Discovery Handbook*, pp. 321–352. Springer; Boston, MA. https://doi.org/10.1007/0-387-25465-X_15.

Sai Krishna, T. V., A. Yesu Babu, and R. Kiran Kumar (2018). Determination of Optimal Clusters for a Non-hierarchical Clustering Paradigm K-Means Algorithm. In N. Chaki, A. Cortesi, and N. Devarakonda (Eds.), *Proceedings of International Conference on Computational Intelligence and Data Engineering*, pp. 301–316. Springer; Singapore. https://doi.org/10.1007/978-981-10-6319-0_26.

Saraçli, S., N. Doğan, and I. Doğan (2013). Comparison of Hierarchical Cluster Analysis Methods by Cophenetic Correlation. *Journal of Inequities and Applications*, 203. https://doi.org/10.1186/1029-242X-2013-203.

Schuhmann, J. (October 14, 2021). NBA's 3-point Revolution: How 1 Shot is Changing the Game. https://www.nba.com/news/3-point-era-nba-75.

Scrucca, L., M. Fop, T. B. Murphy, and A. E. Raftery (2016). mclust 5: Clustering, Classification and Density Estimation Using Gaussian Finite Mixture Models. *The R Journal 8*(1), 289–317. https://doi.org/10.32614/RJ-2016-021.

Temple Lang, D. (2020). *XML: Tools for Parsing and Generating XML Within R and S-Plus.* R package version 3.99-0.5. https://CRAN.R-project.org/package=XML.

van der Maaten, L. and G. Hinton (2008). Visualizing Data Using t-SNE. *Journal of Machine Learning Research 9*(86), 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html.

Van Rossum, G. and F. L. Drake (2009). *Python 3 Reference Manual.* Scotts Valley, CA: CreateSpace. https://dl.acm.org/doi/book/10.5555/1593511.

Wickham, H. (2020a). *httr: Tools for Working with URLs and HTTP.* R package version 1.4.2. https://CRAN.R-project.org/package=httr.

Wickham, H. (2020b). *rvest: Easily Harvest (Scrape) Web Pages.* R package version 0.3.6. https://CRAN.R-project.org/package=rvest.

Wickham, H., M. Averick, J. Bryan, W. Chang, L. D. McGowan, R. François, G. Grolemund, A. Hayes, L. Henry, J. Hester, M. Kuhn, T. L. Pedersen, E. Miller, S. M. Bache, K. Müller, J. Ooms, D. Robinson, D. P. Seidel, V. Spinu, K. Takahashi, D. Vaughan, C. Wilke, K. Woo, and H. Yutani (2019). Welcome to the tidyverse. *Journal of Open Source Software 4*(43), 1686. https://doi.org/10.21105/joss.01686.

Wickham, H., R. François, L. Henry, and K. Müller (2020). *dplyr: A Grammar of Data Manipulation.* R package version 1.0.1. https://CRAN.R-project.org/package=dplyr.

Zuccolotto, P., M. Manisera, and M. Sandri (2021). Alley-oop! Basketball Analytics in R. *Significance 8*(2), 26–31. https://doi.org/10.1111/1740-9713.01507.

APPENDICES

APPENDIX A

Lower Limit for Minutes Played

It is important to verify that selecting a `Minutes Played` cutoff of only 24 minutes (two quarters) will not drastically change the results for the optimal cluster selection. Higher cutoffs of 48 minutes (one game) and 240 minutes (five games) played were tested using the `NbClust` function in the `NbClust` R package. Figures A.1 and A.2 display the combined cluster selection histograms across the 20 NBA seasons, similar to the histogram obtained in Figure 4.2.

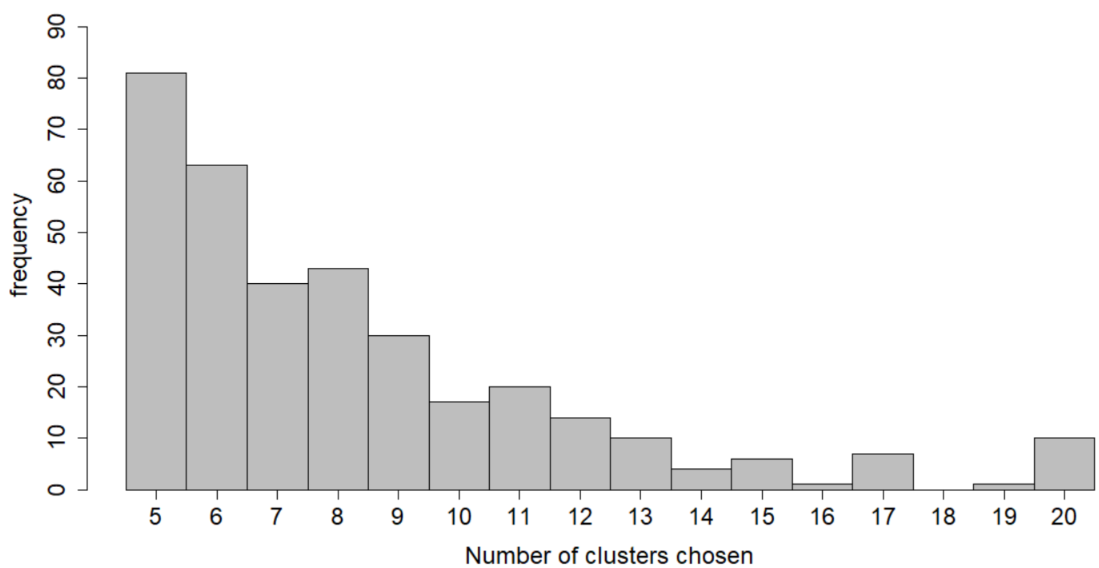


Fig. A.1: Optimal number of clusters for NBA player data based on 26 indices from the 2000-2001 season to the 2019-2020 season using Ward D2 – Using 48 `Minutes Played` minimum cutoff. We can see from these figures that there is still a declining trend as we increase from five clusters to around fourteen or fifteen clusters, followed by a slight incline as we approach twenty clusters.

We can see from these figures that there is still a declining trend as we increase from five clusters to around fourteen or fifteen clusters, followed by a slight incline as we approach twenty clusters. We can can also see that both of these figures show minor ‘jumps’, or local

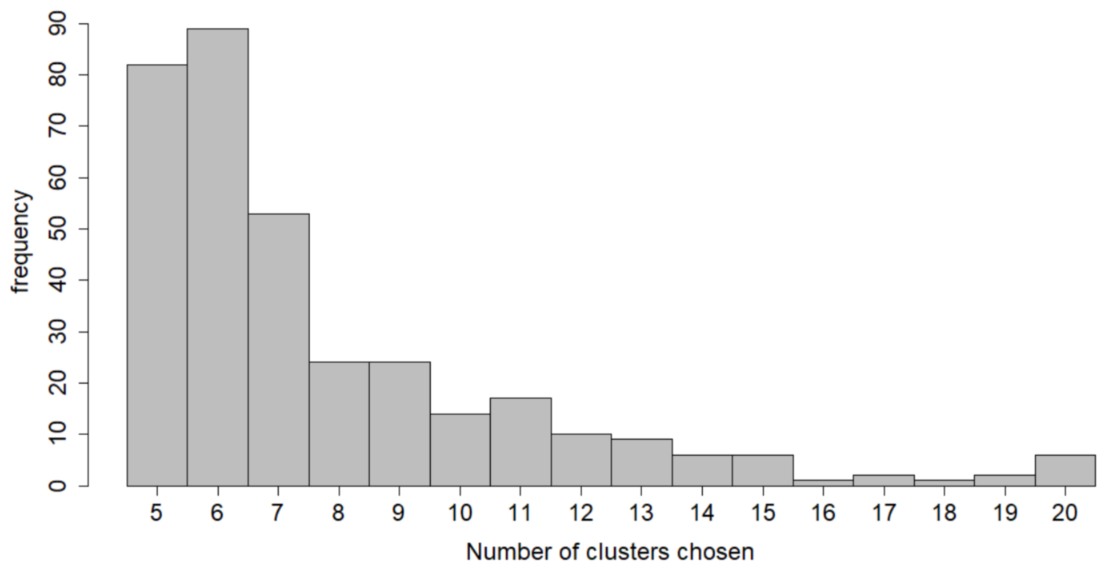


Fig. A.2: Optimal number of clusters for NBA player data based on 26 indices from the 2000-2001 season to the 2019-2020 season using Ward D2 – Using 240 `Minutes Played` minimum cutoff. We can see from these figures that there is still a declining trend as we increase from five clusters to around fourteen or fifteen clusters, followed by a slight incline as we approach twenty clusters.

maximas, at regular intervals, very similar to those observed in Figure 4.2. This observation leads to the assumption that the players who are removed as we increase the cutoffs are not having a major impact on the selections by the `NbClust` R function.

We can also consider the number of players being removed with each of these cutoff levels. Tables A.1, A.2, and A.3 display the number of player rows removed by year using the 24 minute cutoff, 48 minute cutoff, and the 240 minute cutoff. We can observe that the 24 minute cutoff leaves between 95% and 98% of all possible players for the analysis. The 48 minute cutoff leaves between 80% and 89% of all possible players, and the 240 minute cutoff leaves between 67% and 79% of all possible players.

While the argument could be made that setting a minimum of 24 minutes played over the course of an entire season may likely still include many 'garbage time' players, the decision was made to err on the side of *over-inclusion* rather than removing players who could meaningfully contribute to the cluster algorithm and subsequent position characteristics analysis. We would be removing at least 10% of all players with a cutoff of 48 minutes or more, and the research of Alagappan (2012), Jyad (2020), and Kalman and Bosch (2020) makes no mention of the exclusion of any players during the seasons they analyzed. Taking this conservative approach of removing a small amount of players who are almost guaranteed to be 'garbage time' players was determined to be the best course of action.

Table A.1: Number of rows removed by year from player tables using 24 `Minutes Played` cutoff. At least 95% of all possible players are used in each season after applying the cutoff.

Season	Player rows before 24 min cutoff	Player rows after 24 min cutoff	Garbage time players removed	Pct of total data used
2000-2001	510	496	14	**97%**
2001-2002	483	474	9	**98%**
2002-2003	471	455	16	**97%**
2003-2004	570	541	29	**95%**
2004-2005	562	551	11	**98%**
2005-2006	539	519	20	**96%**
2006-2007	489	481	8	**98%**
2007-2008	571	550	21	**96%**
2008-2009	563	542	21	**96%**
2009-2010	572	555	17	**97%**
2010-2011	613	594	19	**97%**
2011-2012	526	511	15	**97%**
2012-2013	553	538	15	**97%**
2013-2014	583	562	21	**96%**
2014-2015	625	608	17	**97%**
2015-2016	561	543	18	**97%**
2016-2017	577	559	18	**97%**
2017-2018	609	579	30	**95%**
2018-2019	639	624	15	**98%**
2019-2020	465	453	12	**97%**

Table A.2: Number of rows removed by year from player tables using 48 `Minutes Played` cutoff. Applying this cutoff eliminates between 10% and 20% of all players for a given season.

Season	Player rows before 48 min cutoff	Player rows after 48 min cutoff	Garbage time players removed	Pct of total data used
2000-2001	510	443	67	**87%**
2001-2002	483	431	52	**89%**
2002-2003	471	419	52	**89%**
2003-2004	570	457	113	**80%**
2004-2005	562	481	81	**86%**
2005-2006	539	455	84	**84%**
2006-2007	489	445	44	**91%**
2007-2008	571	474	97	**83%**
2008-2009	563	465	98	**83%**
2009-2010	572	471	101	**82%**
2010-2011	613	496	117	**81%**
2011-2012	526	473	53	**90%**
2012-2013	553	477	76	**86%**
2013-2014	583	493	90	**85%**
2014-2015	625	524	101	**84%**
2015-2016	561	483	78	**86%**
2016-2017	577	489	88	**85%**
2017-2018	609	505	104	**83%**
2018-2019	639	526	113	**82%**
2019-2020	465	398	67	**86%**

Table A.3: Number of rows removed by year from player tables using 240 `Minutes Played` cutoff. Applying this cutoff eliminates between 20% and 35% of all players for a given season.

Season	Player rows before 240 min cutoff	Player rows after 240 min cutoff	Garbage time players removed	Pct of total data used
2000-2001	510	381	129	**75%**
2001-2002	483	371	112	**77%**
2002-2003	471	366	105	**78%**
2003-2004	570	402	168	**71%**
2004-2005	562	411	151	**73%**
2005-2006	539	392	147	**73%**
2006-2007	489	386	103	**79%**
2007-2008	571	400	171	**70%**
2008-2009	563	395	168	**70%**
2009-2010	572	400	172	**70%**
2010-2011	613	412	201	**67%**
2011-2012	526	399	127	**76%**
2012-2013	553	417	136	**75%**
2013-2014	583	414	169	**71%**
2014-2015	625	443	182	**71%**
2015-2016	561	420	141	**75%**
2016-2017	577	420	157	**73%**
2017-2018	609	426	183	**70%**
2018-2019	639	455	184	**71%**
2019-2020	465	351	114	**75%**

APPENDIX B

NbClust Indices

In this appendix, we include a brief summary of the 30 indices used in the `NbClust` R package (Section 3.1.3) to select the optimal number of clusters. Tables B.1 and B.2 outline the 30 different indices along with their formulas, the logic employed to select the optimal cluster number, and a brief description of its application. Following these tables, the reader may view a glossary of terms and symbols that are used in these equations where applicable.

B.1 NbClust Indices

Table B.1: NbClust Indices 1-15

	INDEX	FORMULA	OPTIMAL # OF CLUSTERS	DESCRIPTION
1	CH	$CH(q) = \frac{trace(B_q)/(q-1)}{trace(W_q)/(n-q)}$	Maximum value of the index	Based on average between and within cluster sum of squares
2	Duda	$Duda >= 1 - \frac{2}{\pi p} - z\sqrt{\frac{2(1-\frac{8}{\pi^2 p})}{n_m p}} = critVal_{Duda}$	Smallest number of clusters such that index $>$ criticalValue	Ratio criterion using sum of squares within clusters
3	Pseudot2	$Pseudot2 <= (\frac{1 - critVal_{Duda}}{critVal_{Duda}}) * (n_k + n_l - 2)$	Smallest number of clusters such that index $>$ criticalValue	*Only applicable for hierarchical Within group dispersion ratio
4	Cindex	$Cindex = \frac{S_w - S_{min}}{S_{max} - S_{min}}, S_{min} \neq S_{max}, Cindex \in (0,1)$	Maximum value of the index	Uses smallest differences between pairs vs largest differences
5	Gamma *not used	$Gamma = \frac{s(+) - s(-)}{s(+) + s(-)}$	Maximum value of the index	Determines if within-cluster dissimilarity is less than between-cluster dissimilarity
6	Beale	$Beale = F \equiv \frac{(\frac{V_{kl}}{W_k + W_l})}{((\frac{n_m - 1}{n_m - 2})2^{\frac{2}{p}} - 1)}$	Number of clusters such that critical value $>=$ alpha	Uses F-ratio to test hypothesis of the existence of q1 vs q2 clusters of data
7	CCC	$CCC = ln[\frac{1 - E(R^2)}{1 - R^2}]\frac{\sqrt{\frac{np^*}{2}}}{(0.0001 + E(R^2))^{1.2}}$	Maximum value of the index	SAS software method: compare R^2 to R^2 obtained from uniformly distributed data
8	Ptbiserial	$Ptbiserial = \frac{[\bar{S}_b - \bar{S}_w][N_w N_b / N_t^2]^{\frac{1}{2}}}{s_d}$	Maximum value of the index	A point biserial correlation between the raw input dissimilarity matrix and a corresponding matrix of 0's and 1's
9	Gplus *not used	$Gplus = \frac{2s(-)}{N_t(N_t - 1)}$	Maximum value of the index	Uses proportion of discordant pairs among all pairs of distinct points
10	DB	$DB(q) = \frac{1}{q}\sum_{k=1}^{q} \max_{k \neq l}(\frac{\delta_k + \delta_l}{d_{kl}})$	Maximum value of the index	Sum ratio of within-cluster scatter to between-cluster separation
11	Frey	$Frey = \frac{\bar{S}_{b_{j+1}} - \bar{S}_{b_j}}{\bar{S}_{w_{j+1}} - \bar{S}_{w_j}}$	Cluster level before index val <100	* Only applicable for hierarchical Ratio of difference scores from two successive levels in the hierarchy
12	Hartigan	$Hartigan = (\frac{trace(W_{\tilde{q}})}{trace(W_{\tilde{q}+1})} - 1)(n - \tilde{q} - 1)$	Maximum difference between hierarchy levels of the index	Based on Euclidean within-cluster sum of squares
13	Tau *not used	$Tau = \frac{s(+) - s(-)}{[(N_t(N_t - 1)/2 - t)(N_t(N_t - 1)/2)]^{1/2}}$	Maximum value of the index	Computed between corresponding entries within two matrices: first shows distance between items, and the second 0/1 indicates if pairs are within same cluster
14	Ratkowsky	$Ratkowsky = \frac{\bar{S}}{q^{1/2}}$	Maximum value of the index	Based on the sum of squares between clusters for each variable and the total sum of squares for each variable
15	Scott	$Scott = n \log \frac{\det(T)}{\det(W_q)}$	Maximum difference between hierarchy levels of the index	Log of determinants of within sum of squares and total sum of squares

Table B.2: NbClust Indices 16-30

	INDEX	FORMULA	OPTIMAL # OF CLUSTERS	DESCRIPTION
16	Marriot	$Marriot = q^2 \det(W_q)$	Max. value of second differences between levels of the index	Uses determinant of within sum of squares
17	Ball	$Ball = \frac{W_q}{q}$	Maximum difference between hierarchy levels of the index	Based on the average distance of items to their cluster centroids
18	Trcovw	$Trcovw = trace(COV(W_q))$	Maximum difference between hierarchy levels of the index	Trace of within clusters pooled covariance matrix
19	Tracew	$Tracew = trace(W_q)$	Max. value of second differences between levels	*One of most popular Uses trace of within cluster sum of squares
20	Friedman	$Friedman = trace(W_q^{-1} B_q)$	Maximum difference between hierarchy levels of the index	*Used for non-hierarchical clustering Uses the trace of the inverse within sum of squares matrix and the between sum of squares matrix
21	McClain	$McClain = \frac{\bar{S}_w}{\bar{S}_b} = \frac{S_w/N_w}{S_b/N_b}$	Minimum value of the index	Ratio using the average within cluster distance and the average between cluster distance compared to the number of total distances
22	Rubin	$Rubin = \frac{\det(T)}{\det(W_q)}$	Minimum value of second differences between levels	Based on the ratio of the determinant of the total sum of squares and cross products matrix to the determinant of the pooled within cluster matrix
23	KL	$KL(q) = \lvert\frac{DIFF_q}{DIFF_{q+1}}\rvert$	Maximum value of the index	Uses the trace of within sum of squares
24	Silhouette	$Silhouette = \frac{\sum_{i=1}^{n} S(i)}{n}, Silhouette \in [-1, 1]$	Maximum value of the index	Uses the mean distance of a point to the points in the cluster to which it belongs vs the mean distance to the points not in its cluster
25	Gap *not used	$Gap(q) = \frac{1}{B}\sum_{b=1}^{B} \log W_{q^b} - \log W_q$ $Gap(q) \geq Gap(q+1) - s_q + 1, (q = 1, ..., n-2)$	Smallest number of clusters such that criticalVal >= 0	The gap statistic compares the total within intra-cluster variation for different values of k within their expected values under null reference distribution of the data
26	Dindex	$Gain = w(P^{q-1}) - w(P^q)$	Graphical method	Based on clustering gain on intra-cluster inertia
27	Dunn	$Dunn = \frac{\min_{1 \leq i < j \leq q} d(C_i, C_j)}{\max_{1 \leq k \leq q} diam(C_k)}$	Maximum value of the index	Uses the ratio between the minimal intercluster distance to maximal intracluster distance
28	Hubert	$\Gamma(P, Q) = \frac{1}{N_t} \sum_{i=1, i<j}^{n-1} P_{ij} Q_{ij}$	Graphical method	Uses a point-serial correlation coefficient between any two matrices
29	SDindex	$SDindex(q) = \alpha Scat(q) + Dis(q)$	Minimum value of the index	Based on the concepts of average scattering for clusters and total separation between clusters
30	SDbw	$SDbw(q) = Scat(q) + Density.bw(q)$	Minimum value of the index	Based on the criteria of compactness and separation between clusters

B.2 Glossary of Terms

General Terms

$n =$ number of observations,

$p =$ number of variables,

$q =$ number of clusters,

$X = \{x_{ij}\}, i = 1, 2, ..., n, j = 1, 2, ..., p,$

$= n \times p$ data matrix of p variables measured on n independent observations,

$\bar{X} = q \times p$ matrix of cluster means,

$\bar{x} =$ centroid of data matrix X_i,

$k, l = 1, ..., q =$ cluster number,

$C_k =$ a given cluster in the data, where $k = 1, ..., q$,

$n_k =$ number of objects in cluster C_k, $k = 1, ..., q$,

$c_k =$ centroid of cluster C_k,

$x_i = p$-dimensional vector of observations of the ith object in the cluster C_k,

$||x|| = (x^T x)^{1/2}$,

$W_q = \sum_{k=1}^{q} \sum_{i \in C_k} (x_i - c_k)(x_i - c_k)^T$ is the within-group dispersion matrix for the data clustered into q clusters,

$B_q = \sum_{k=1}^{q} n_k (c_k - \bar{x})(c_k - \bar{x})^T$ is the between-group dispersion matrix for the data clustered into q clusters,

$N_t =$ total number of pairs of observations in the data set: $N_t = \frac{n(n-1)}{2}$,

$N_w =$ total number of pairs of observations belonging to the same cluster: $N_w = \sum_{k=1}^{q} \frac{n_k(n_k-1)}{2}$,

$N_b =$ total number of pairs of observations belonging to different clusters: $N_b = N_t - N_w$,

$d(x, y) = \sqrt{\sum_{j=1}^{p} (x_j - y_j)^2} =$ Euclidean distance between two vectors x and y,

$S_w =$ sum of the within-cluster differences: $S_w = \sum_{k=1}^{q} \sum_{i,j \in C_k, i<j} d(x_i, x_j)$,

$S_b =$ sum of the between-cluster differences: $S_b = \sum_{k=1}^{q-1} \sum_{l=k+1}^{q} \sum_{i \in C_k, j \in C_l} d(x_i, x_j)$

Duda

$Duda = \frac{Je(2)}{Je(1)} = \frac{W_k+W_l}{W_m}$, where $C_m = C_k \cup C_l$

and where W_k and W_l are the within-group dispersions for clusters C_k and C_l, and W_m is the within-group disperion for cluster C_m.

Pseudot2

$Pseudot2 = \frac{V_{kl}}{\frac{W_k+W_l}{n_k+n_l-2}}$

where $V_{kl} = W_m - W_k - W_l$, and $C_m = C_k \cup C_l$

and where W_k and W_l are the within-group dispersions for clusters C_k and C_l, and W_m is the within-group disperion for cluster C_m.

Cindex

S_{min} = the sum of the l_w smallest distances between all the pairs of points in the entire data set (there are l_t such pairs);

S_{max} = the sum of the l_w largest distances between all the pairs of points in the entire data set.

Gamma

$s(+)$ = number of concordant comparisons,

$s(-)$ = number of discordant comparisons.

Beale

$V_{kl} = W_m - W_k - W_l$.

CCC

$R^2 = 1 - \frac{trace(X^TX - \bar{X}^TZ^TZ\bar{X})}{trace(X^TX)}$

$X^T X$ = total-sample sum-of-squares and crossproducts (SSCP) matrix ($P \times P$),

$\bar{X} = (Z^T Z)^{-1} Z^T X$,

Z is a cluster indicator matrix $n \times q$ with element $z_{ik} = 1$ if the ith observation belongs to the kth cluster and $z_{ik} = 0$ otherwise,

$E(R^2) = 1 - [\frac{\sum_{j=1}^{p*} \frac{1}{n+u_j} + \sum_{j=p*+1}^{p} \frac{u_j^2}{n+u_j}}{\sum_{j=1}^{p} u_j^2}][\frac{(n-q)^2}{n}][1 + \frac{4}{n}]$,

$u_j = \frac{s_j}{c}$, $j = 1, .., p$,

s_j = square root of the jth eigenvalue of $X^T X/(n-1)$, $j = 1, .., p$,

$c = (\frac{v^*}{q})^{(\frac{1}{p^*})}$,

$v^* = \prod_{j=1}^{p^*} s_j$,

p^* is chosen to be the largest integer less than q such that u_{p^*} is not less than one.

Ptbiserial

$\bar{S}_w = S_w/N_w$,

$\bar{S}_b = S_b/N_b$,

s_d = standard deviation of all distances.

Gplus

$s(\text{-})$ = the number of discordant comparisons.

DB

$d_{kl} = \sqrt[v]{\sum_{k=1}^{p} |c_{k_j} - c_{l_j}|^v}$ = distance between centroids of clusters C_k and C_l (for v=2, d_{kl} is the Euclidean distance),

$\delta_k = \sqrt[u]{\frac{1}{n_k} \sum_{i \in C_k} \sum_{j=1}^{p} |x_{ij} - c_{kj}|^u}$ = dispersion measure of a cluster C_k to the centroid of this cluster).

Frey

$\bar{S}_b = S_b/N_b =$ mean between-cluster distance,

$\bar{S}_w = S_w/N_w =$ mean within-cluster distance.

Hartigan

$\tilde{q} \in \{1, ..., n-2\}$.

Tau

$s(+)$ represents the number of times where two points not clustered together had a larger distance than two points which were in the same cluster, i.e., $s(+)$ is the number of concordant comparisons,

s(-) represents the reverse outcome, i.e., s(-) is the number of discordant responses,

N_t is the total number of distances and t is the number of comparisons of two pairs of points where both pairs represent within cluster comparisons or both pairs are between cluster comparisons.

Ratkowsky

$\bar{S}^2 = \frac{1}{p}\sum_{j=1}^{p}\frac{BGSS_j}{TSS_j}$,

$BGSS_j = \sum_{k=1}^{q} n_k(c_{kj} - \bar{x}_j)^2$.

KL

$DIFF_q = (q-1)^{2/p}trace(W_{q-1}) - q^{2/p}trace(W_q)$

Silhouette

$S(i) = \frac{b(i)-a(i)}{max\{a(i);b(i)\}}$,

$a(i) = \frac{\sum_{j\in\{C_r/i\}} d_{ij}}{n_r - 1}$ is the average dissimilarity of the ith object to all other objects of cluster

C_r,

$b(i) = \frac{\sum_{j \in C_s} d_{ij}}{n_s}$ is the average dissimilarity of the ith object to all objects of cluster C_s.

Gap

$B =$ the number of reference data sets generated using uniform prescription,

$W_{qb} =$ within-dispersion matrix as defined in Hartigan Index,

$s_q = sd_q \sqrt{1 + 1/B}$,

sd_q is the standard deviation of $\{\log W_{qb}\}, b = 1, ..., B : sd_q = \sqrt{\frac{1}{B} \sum_{b=1}^{B} (\log W_{qb} - \bar{l})^2}$,

$\bar{l} = \frac{1}{B} \sum_{b=1}^{B} \log W_{qb}$.

Dindex

$w(P^q) = \frac{1}{q} \sum_{k=1}^{q} \frac{1}{n_k} \sum_{x_i \in C_k} d(x_i, c_k)$

Dunn

$d(C_i, C_j) = \min_{x \in C_i; y \in C_j} d(x, y)$

$diam(C) = \max_{x, y \in C} d(x, y)$

Hubert

$P =$ the proximity matrix of the data set,

$Q =$ an $n \times n$ matrix whose (i, j) element is equal to the distance between the representative points (v_{c_i}, v_{c_j}) of the clusters where the objects x_i and x_j belong.

$\bar{\Gamma} = \frac{\sum_{i=1, i<j}^{n-1} (P_{ij} - \mu_P)(Q_{ij} - \mu_Q)}{\sigma_P \sigma_Q}$

SDindex

$Scat(q) = \frac{\frac{1}{q}\sum_{k=1}^{q}||\sigma^{(k)}||}{||\sigma||}$,

$\sigma = (VAR(V_1), VAR(V_2), ..., VAR(V_p))$; i.e. the vector of variances for each variable in the data set,

$\sigma^{(k)} = (VAR(V_1^{(k)}), VAR(V_2^{(k)}), ..., VAR(V_p^{(k)}))$; i.e. the variance vector for each cluster C_k,

$Dis(q) = \frac{D_{max}}{D_{min}} \sum_{k=1}^{q}(\sum_{z=1}^{q}||c_k - c_z||)^{-1}$,

$D_{max} = \max(||c_k - c_z||) \forall k, z \in \{1, 2, 3, ..., q\}$ is the maximum distance between cluster centers,

$D_{min} = \min(||c_k - c_z||) \forall k, z \in \{1, 2, 3, ..., q\}$ is the minimum distance between cluster centers.

SDbw

$Density.bw(q) = \frac{1}{q(q-1)} \sum_{i=1}^{q}(\sum_{j=1, i\neq j} \frac{density(u_{ij})}{\max(density(c_i), density(c_j))})$,

u_{ij} = the middle point of the line segment defined by the clusters centroids c_i and c_j,

$density(u_{ij}) = \sum_{l=1}^{n_{ij}} f(x_l, u_{ij})$,

n_{ij} = the number of tuples that belong to the clusters C_i and C_j,

$f(x_l, u_{ij})$ = equal to 0 if $d(x, u_{ij}) > Stdev$ and 1 otherwise,

$Stdev = \frac{1}{q}\sqrt{\sum_{k=1}^{q}||\sigma^{(k)}||}$,

$\sigma^{(k)}$ = variance vector for each cluster C_k as described in SDindex.

APPENDIX C

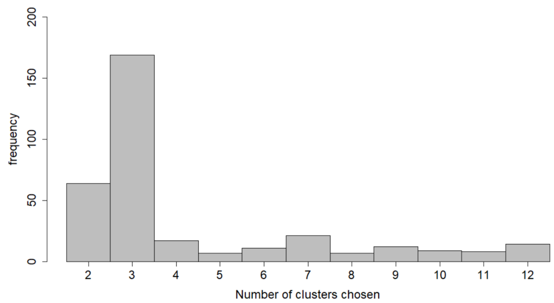
NbClust Start/End Points

In this appendix, we provide an overview of the effect of varying start and end points on the optimal cluster selection conducted by the `NbClust` R package. Since one of the original goals of this MS thesis is to expand current player positions and definitions beyond the traditional five, a starting point of five for the `NbClust` procedure seemed logical. However, we must also consider that there may potentially be fewer than five meaningful positions. The argument could be made that there are only two types of player: Ball-Handlers and Non-Ball-Handlers. We must also consider what happens to the selections as we vary the upper limit of the procedure. Figure C.1 provides the `NbClust` selection histograms with combinations of two starting points (two & five) and three ending points (twelve, fifteen, & twenty).

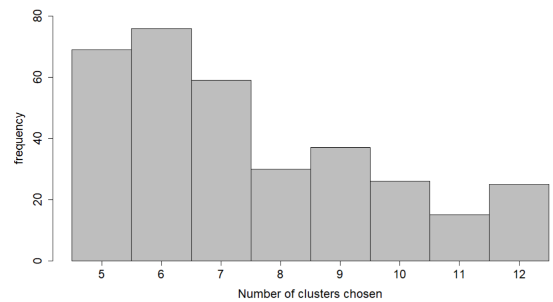
As the starting point moves from two clusters to five clusters, we notice that the highest frequency occurs with three clusters and six clusters, respectively. The three histograms on the left side of Figure C.1 with 'Start=2' show the consensus falling heavily in favor of three clusters. While proceeding with three clusters would have been a legitimate option, the purpose of this research is to describe players in more detail and explore more subtle differences between players through visualization. Describing players in more than five ways will also allow for more precise lineup creation and performance prediction.

The reader will also note that most of the figures show slight increases on the right limit of the histogram, whether the end point is twelve, fifteen, or twenty. This is likely a result of the limits capturing the maximum value plus every choice that would have fallen beyond that maximum value. For this reason, it made sense to ignore the local maximas that occur at both end points because they are capturing all cluster number selections as extreme or more extreme than that value.

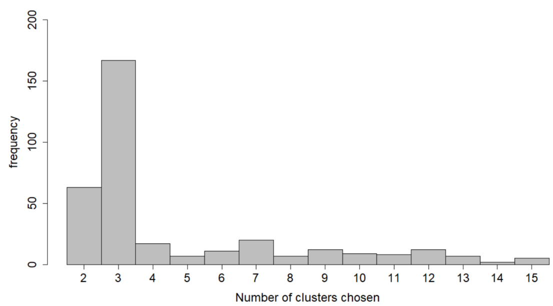
While each of the six histograms presented may appear to tell a slightly different story, the key feature to be considered for this analysis is the ‘jumps’ at nine clusters in the three histograms on the right column of Figure C.1 where ‘Start=5’. When we calculate the optimal cluster number under the assumption that there are more than five meaningful player positions, it is logical to consider nine or even twelve clusters for subsequent analysis.



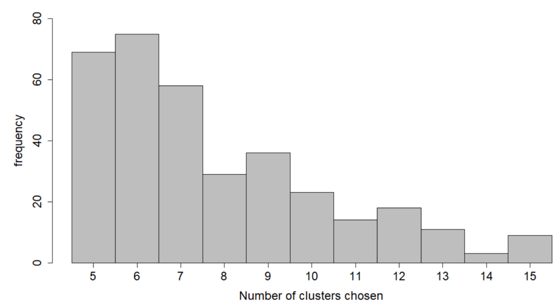
(a) Start=2, End=12



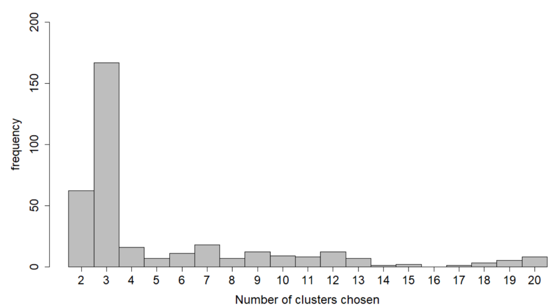
(b) Start=5, End=12



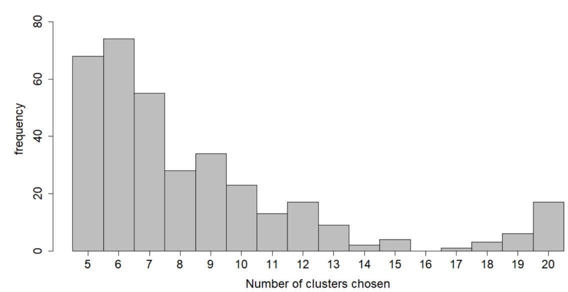
(a) Start=2, End=15



(b) Start=5, End=15



(c) Start=2, End=20



(d) Start=5, End=20

Fig. C.1: Optimal number of clusters selected for the 2000-2001 NBA season with varying start and end points. The three histograms on the left side of the figure with ‘Start=2’ show the consensus falling heavily in favor of three clusters, while the three figures on the right with ‘Start=5’ choose six clusters as the optimal number.

APPENDIX D

Adjusted Rand Index Simulations

This appendix contains the ARI scores calculated for each pair of adjacent NBA seasons (see Table 5.1) can be compared to baseline ARI calculations to confirm that the NBA player clustering similarities from year to year were not observed by chance.

Each NBA season's players were randomly assigned to one of nine clusters, and the amount of players randomly placed in a given cluster was fixed to the amount of players placed in that cluster by Ward's D2 method. 9,999 such clusterings were performed for each NBA season and compared to the following season. The results for each comparison can be viewed in Figure D.1.

We can see that most ARI scores that were calculated from the simulated player clusterings lie between -0.025 and 0.025. When we consider the actual ARI results found in Table 5.1, we can confirm that there is consistency in player clusterings from season to season. The lowest ARI score observed between two adjacent NBA seasons occurred between the 2018-2019 season and the 2019-2020 season (0.182). We would overall expect there to be many players placed in the same clusters from year to year, but we would also expect many players' roles and positions to change over time for many reasons, including age, injuries, or skill development.

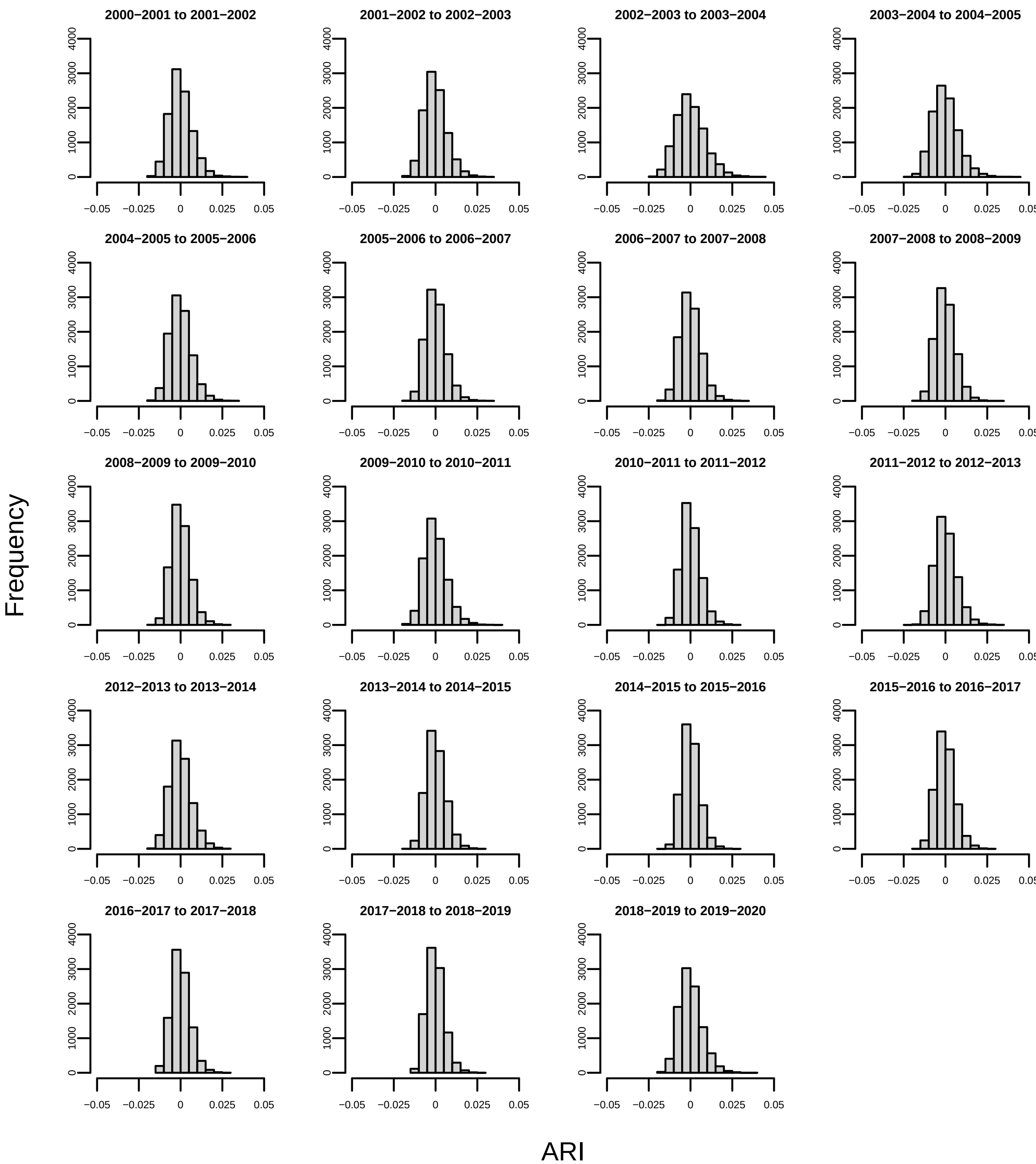


Fig. D.1: 9999 random ARI simulations for each pair of adjacent NBA seasons. Nearly all ARI simulations across the 20 seasons fall between -0.025 and 0.025.

APPENDIX E

Visualizing Three Clusters

This appendix includes a brief discussion and analysis of what would happen if we had only chosen three player clusters as the `NbClust` output suggests. While this outcome does not align with our purpose of analyzing player differences in great detail, this approach must be considered due to being the overwhelming choice by the 26 indices.

Figure E.1 displays the the PCA plot for the players in the 2000-2001 NBA season using the `factoextra` R package. The reader may compare this plot to Figure 4.5, which shows the same players clustered into nine different positions. If we consider the same players listed in Table 5.3 in the main text, we find that the `Defensive Big Men` and `Interior Big Men` (Clusters 7 & 8) are combined into a Traditional Big Men (Cluster 3) in this example. The `Scoring Big Men` and `Superstars` (Clusters 3 & 9) are combined into `Ball-Dominant Scorers` (Cluster 2) in this figure. All other players in Clusters 1, 2, 4, 5, and 6 are found in Cluster 1.

While these results are not discussed in the main text, it is informative to observe how players may be more similar than different in many ways. Even though the focus of this research centers on detailed player differences, the broad view that there may be only two or three different types of players, especially in lower league levels, is certainly noteworthy.

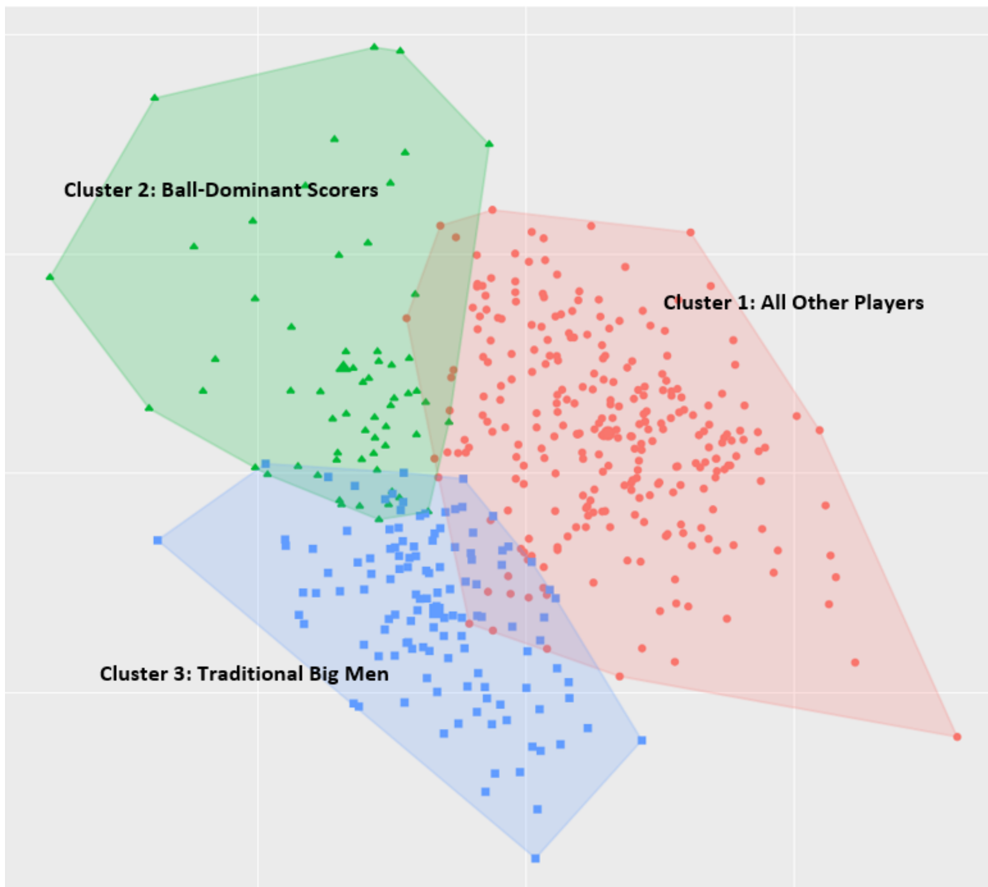


Fig. E.1: PCA plot using `factoextra` R package for players in the 2000-2001 NBA season - separated into three clusters