Should NBA Teams Attempt More 3-Point Shots?

Data 7001: Group J
Keith Curtis
Yuxia Fu
Tse-Wei Lee
Sanjay Pitty
Zongjian Wu

We give consent to use this report as a teaching resource.

Executive Summary

This report discusses Team J's research on whether National Basketball Association (NBA) teams should attempt more 3-Point shots. Specifically, is there an optimum mix of 2-Point and 3-Point shots that teams should target? The impetus for this research is the sharp increase in 3-Point shot attempts over the last several seasons. Key stakeholders include individuals with direct involvement in the NBA, including owners, front office executives, coaches, and the players who will actually attempt the 3-Point shots. Sport fans and those who participate in sport gambling are also stakeholders.

Section 1 provides an overview of the NBA, and also presents the research question and key stakeholders. The data source is introduced in Section 2, and four variables that were calculated from the original data are presented in Section 3. Section 4 discusses technical details of the two linear regression models and K-means clustering analysis that addressed the research question, while graphical storytelling that enhances these technical details are presented in Section 5. Conclusions and recommendations that resulted from the analysis are provided in Section 6.

Based on the research, there is not an optimum mix of 2-Point and 3-Point shot attempts. Teams should prioritize improving their accuracy with each type of shot, instead of changing their mix of shots. There is not an optimum playing style, as illustrated by the varied styles of the last three NBA Champions. One limitation of the research is that aggregate data was obtained, and so the optimum mix of 2-Point and 3-Point shots in specific game situations could not be determined.

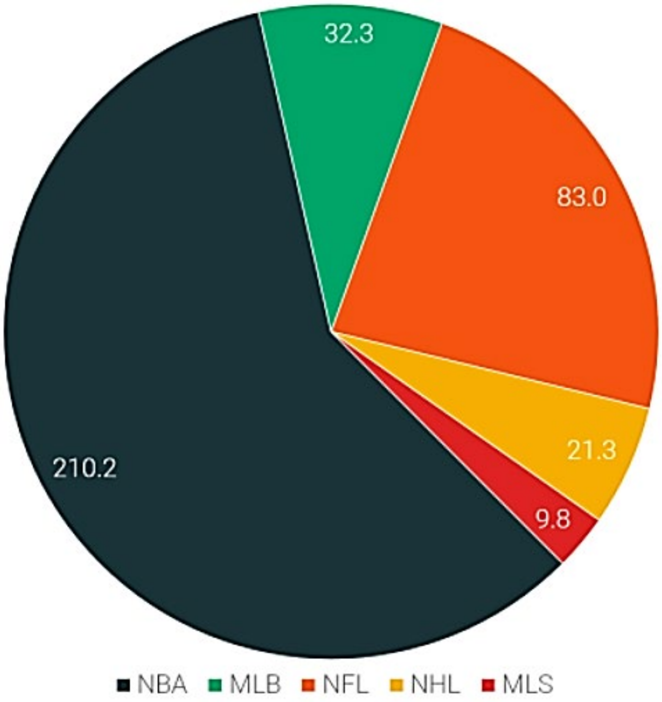
Table of Contents

1 Problem Solving with Data

1.1 Introduction

The National Basketball Association (NBA) is a professional basketball league in North America. They began play in 1946, and there are currently 30 teams. All but one is team is based in the United States. The lone exception is the Toronto Raptors in Canada. Each team is valued at approximately $2.48 Billion in U.S. Dollars[1], and the annual industry revenue is about $10 Billion U.S. Dollars[2]. Furthermore, indirect industries such as sports betting and media invest heavily in NBA-related industry and businesses. The NBA is the most popular sports league on social media[3]. The figure below is a pie chart that was obtained from DTGB. It shows that among the top five sports leagues in North America, the NBA has a 58.9% share of social media followers:

Figure 1: Popularity of North American Sports Leagues on Social Media (in Millions)



In recent years, the outstanding performance and popularity of Stephen Curry have revolutionised the basketball industry. The general public considered him the pioneer who started the new chapter of the 3 points era of the NBA. Starting from the legendary season of 2015-2016, Stephen Curry changed the role of 3-Point shots in basketball. Although some fans criticised that 3-Pointers are harming the diversity of basketball games, there is no doubt that every team realised how critical 3 points could be within basketball strategy[4].

1.2 2-Point v. 3-Point Shots

In basketball, all attempted shots besides free throws are called field goals. Prior to 1979, all successful field goals in the NBA were worth 2 points. During the 1979 – 1980 NBA Season, a 3-

Point field goal was added. This shot is worth more because of its difficulty, since it is either longer than a 2-Point shot, or it is attempted from an area where the backboard cannot be utilized. On each side of a basketball court, two parallel lines and an arc are painted to separate the areas where a successful field goal is worth 2 or 3 points. The current NBA 3-Point shot distance varies from 22 Feet (6.7 Meters) in the corner to 23 Feet 9 Inches (7.2 Meters) from the arc.

The figures below are diagrams of an NBA basketball court[5]. In both diagrams, the 2-Point area is shaded in yellow, while the 3-Point area is shaded in green. Figure 2 applies to teams that are shooting at the basket on the left, while teams are shooting at the right basket in Figure 3:

Figure 2: NBA Court Diagram (Team is Shooting at the Left Basket)

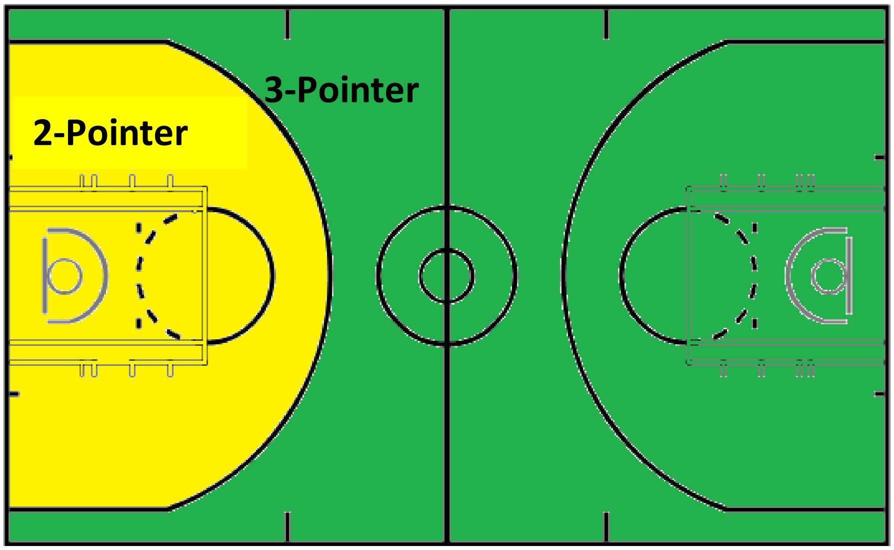
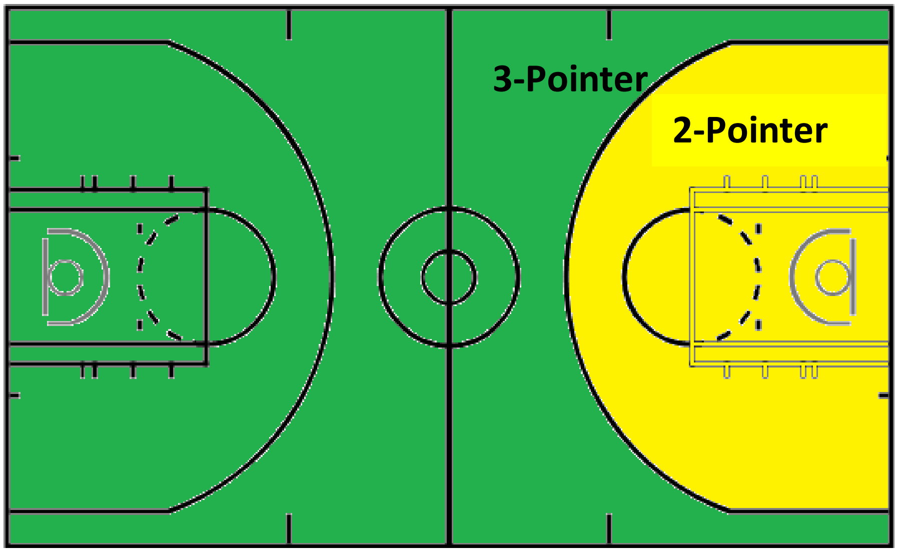
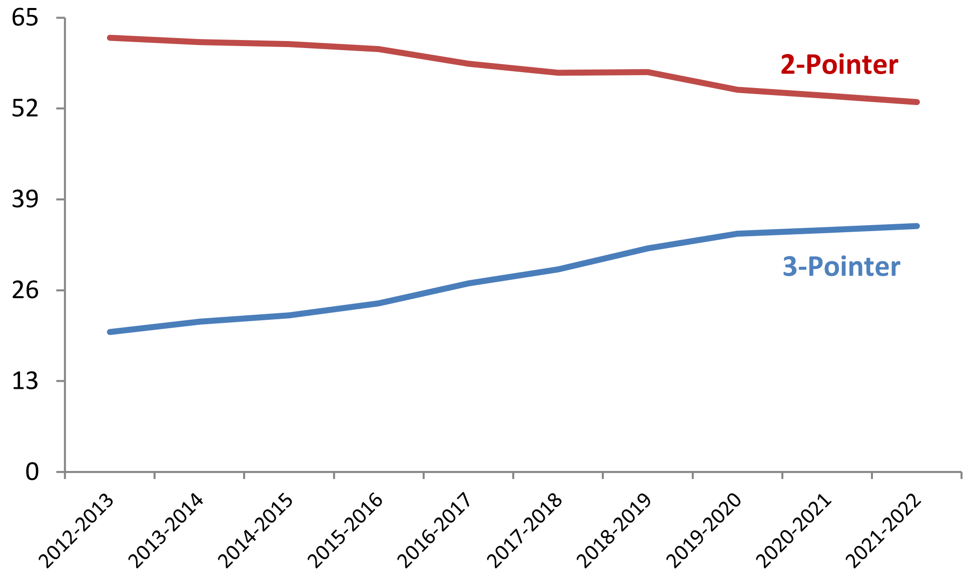


Figure 3: NBA Court Diagram (Team is Shooting at the Right Basket)



During the last several seasons, the number of 3-Point shot attempts has increased dramatically. The figure below shows the 10-year trend in the average number of combined 2-Point and 3-Point shot attempts that were attempted by both teams in a game. The average number of 2-Point attempts has decreased from 62.1 to 52.9 (17.4%) since the 2012-2013 Season, while 3-Point attempts have increased from 20.0 to 35.2 (76.0%) during the same time period:

Figure 4: 10-Year Trend of Per-game Field Goal Attempts by Type



1.3 Research Question

The main question that was explored is whether there is an optimum mix of 2-Point and 3-Point shot attempts. In other words, how does the additional reward of earning one more point for a successful 3-Point shot justify the risk of missing more shots?

1.4 Data Science Process

The Data Science Process steps that were outlined during the class were followed for this project, and they are listed below:

1. Problem Solving with Data
2. Getting the Data I Need
3. Is My Data Fit for Use
4. Making the Data Confess
5. Storytelling with Data

1.5 Key Stakeholders

The key stakeholders for this project are outlined below:

- NBA Team Owners and Management Teams: This report can impact how much 3-Point shooting ability should influence team roster choices. One motivation for wise roster choices is the enormous financial investment in players, as indicated by the $109.14 salary cap for each team during the 2020-2021 Season[6].

- Professional Sports Coaches: This report can help professional sports coaches to organize a competitive in-game strategy. For example, whether the team should focus on the 3-Point attempts in the second half of the game.

- Sports Gambling: The sports gaming stakeholder can use this report to calculate the winning odds. If 3 point shots can outperform 2-Point attempts, then sports gambling stakeholders can adjust the odds of a team with better 3-Point players.

- Basketball Players: Basketball players can use the result from this report for more statistically efficient training. The knowledge can be valuable in decision-making during the game.

- The population of sports fans: The general interest in 3-Point shots has increased since Stephen Curry started the 3-Point revolution in the NBA. The report can provide a data-science-based explanation of this 3-Point trend.

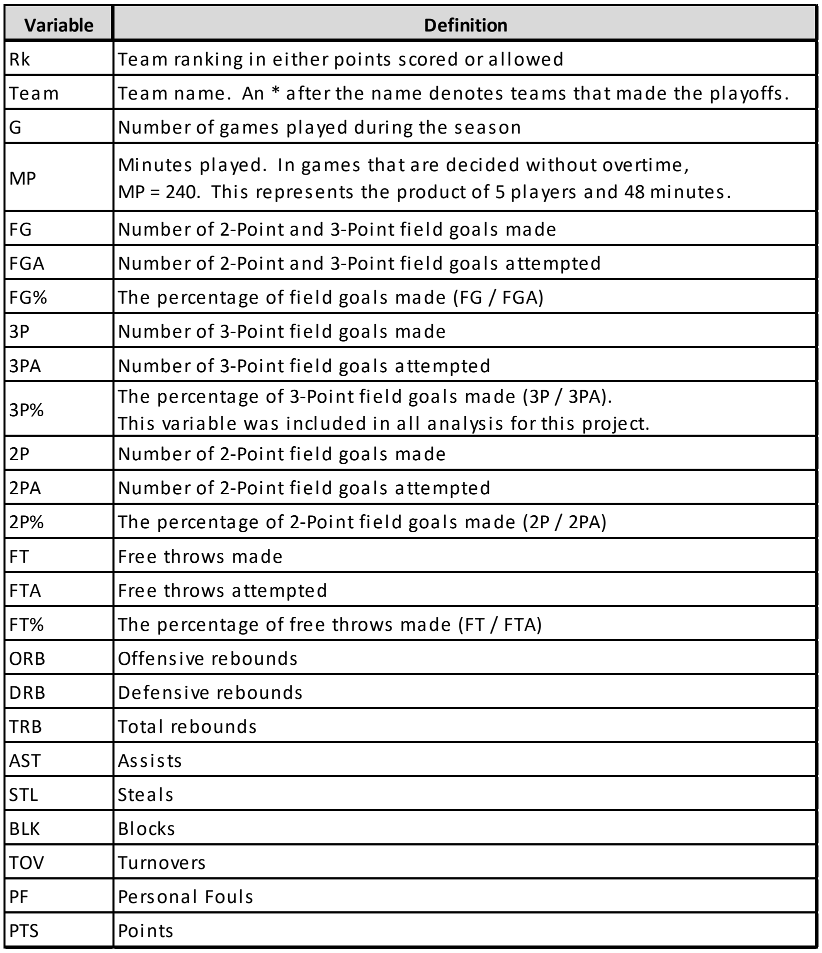
2 Getting the Data I Need

All data for this project was obtained from Basketball Reference[7]. It is a comprehensive data source, as it contains player, team and season statistics from every NBA season. Box scores from individual games are also available from Basketball Reference.

3 Is My Data Fit For Use?

Statistics from the last ten NBA seasons were obtained from Basketball Reference. This data is contained in comma-separated value (.csv) files, and each season is represented by two files. One file contains the team's statistics, while the other contains the aggregate statistics that opponents achieved against the team. All 20 files contain 31 rows. The first 30 rows represent each NBA team, while the last row contains league-wide averages for the season. All quantitative variables within the files are per-game averages, except for the team rank and number of games. Team statistics from the most recent season (2021 – 2022) are found in Appendix A. A glossary of the columns within each file is provided in the table below:

Table 1: Glossary of the Variables from the Basketball Reference Files



Variable	Definition
Rk	Team ranking in either points scored or allowed
Team	Team name. An * after the name denotes teams that made the playoffs.
G	Number of games played during the season
MP	Minutes played. In games that are decided without overtime, MP = 240. This represents the product of 5 players and 48 minutes.
FG	Number of 2-Point and 3-Point field goals made
FGA	Number of 2-Point and 3-Point field goals attempted
FG%	The percentage of field goals made (FG / FGA)
3P	Number of 3-Point field goals made
3PA	Number of 3-Point field goals attempted
3P%	The percentage of 3-Point field goals made (3P / 3PA). This variable was included in all analysis for this project.
2P	Number of 2-Point field goals made
2PA	Number of 2-Point field goals attempted
2P%	The percentage of 2-Point field goals made (2P / 2PA)
FT	Free throws made
FTA	Free throws attempted
FT%	The percentage of free throws made (FT / FTA)
ORB	Offensive rebounds
DRB	Defensive rebounds
TRB	Total rebounds
AST	Assists
STL	Steals
BLK	Blocks
TOV	Turnovers
PF	Personal Fouls
PTS	Points

The data within all 20 files is complete, and therefore no missing value imputation was needed. However, other pre-processing was necessary before further analysis could be performed. The pre-processing steps are listed below, and they were performed in R:

1. Append the 10 .csv files that contain team statistics and league averages.

2. Add a column that contains the season. This is a value between 2012_2013 and 2021_2022. NBA seasons are denoted by the year that they start and finish, and they typically run from October to June.

3. Output the 300 team/season combinations and 10 league average rows into separate .csv files. Note that the 10 league average rows were only used to create the line graphs in Figure 4.

4. Append the 10 .csv files that contain opponent statistics and league averages.

5. Add a column that contains the season.

6. Output the 300 opponent/season combinations into a .csv file. The 10 league average rows can be discarded, since the results are identical to the 10 rows from the team files.

7. Merge the 10 season team and opponent datasets by team name (Team) and season (Season). Since the variable names are identical in the team and opponent files, a suffix of _T or _O is added to each variable name besides Team and Season that is obtained from the team or opponent file, respectively. A suffix is not added to Team and Season because those two variables are used to merge the team and opponent datasets.

8. Calculate four new variables in the merged dataset. These variables are described below:

   - 3-Point Attempt %: The percentage of a team's field goal attempts that are 3-Point shots. The formula is shown below:

     $$\text{3-Point Attempt \%} = \frac{Per\ Game\ Average\ 3\ Point\ Attempts}{Per\ Game\ Average\ 2\ Point\ Attempts + Per\ Game\ Average\ 3\ Point\ Attempts}$$

     This variable is a feature in both regression models, as well as the K-means cluster analysis. 3-Point Attempt % provides a more fair comparison among teams than per-game average 3-Point attempts, since better teams tend to have more opportunity to attempt shots.

   - Point Differential: Average per-game point differential between a team and their opponent. Positive values denote that a team outscores their opponents on average, while a team is being outscored on average has negative values. Point Differential is the response variable for the two regression models that were fit.

   - Playoffs: A Yes/No indicator of whether the team made the playoffs during that season.

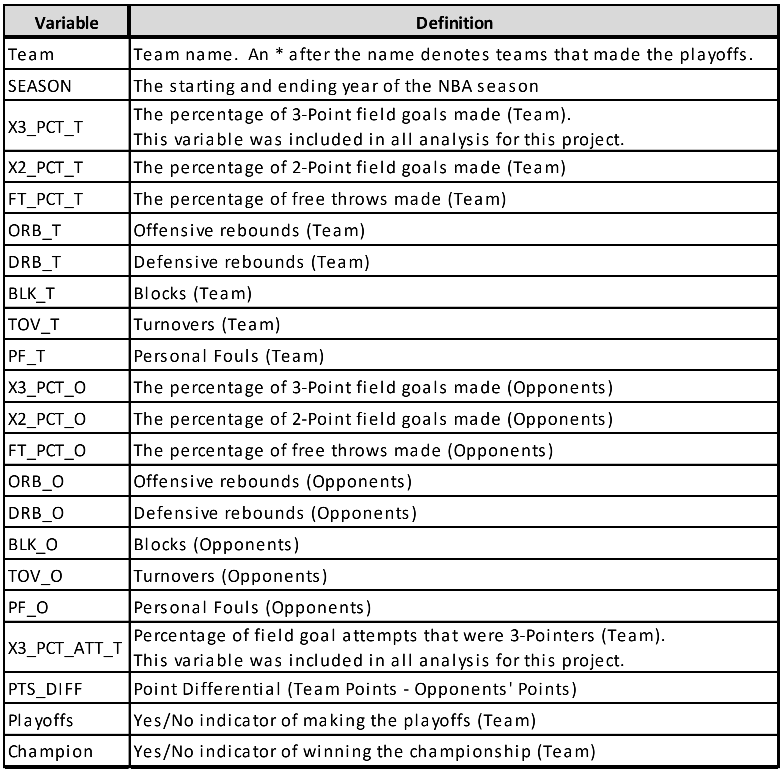
   - Champion: A Yes/No indicator of whether the team was NBA Champion during the season.

9. Remove several variables from the merged dataset. Variables that are irrelevant to answering the research question were removed, including the number of games (G), minutes played (MP) and assists (AST). Total rebounds (TRB) were removed because they are the sum of offensive (ORB) and defensive rebounds (DRB), and different conclusions can be drawn for the separate rebound types. The overall field goal percentage variable (FG%) was removed for a similar reason, because the accuracy of the individual field goal types, namely 2-Point and 3-Point field goals, provides more information. The eight variables that pertain to field goal and free throw counts also were removed (FG, FGA, 3P, 3PA, 2P, 2P, FT, FTA) because it was preferable to analyze

percent accuracy over the number of shots. For the removed variables, both the team and opponent versions were discarded.

The table below shows a glossary of the remaining variables in the final dataset:

Table 2: Glossary of the Variables from the Final Dataset



Variable	Definition
Team	Team name. An * after the name denotes teams that made the playoffs.
SEASON	The starting and ending year of the NBA season
X3_PCT_T	The percentage of 3-Point field goals made (Team). This variable was included in all analysis for this project.
X2_PCT_T	The percentage of 2-Point field goals made (Team)
FT_PCT_T	The percentage of free throws made (Team)
ORB_T	Offensive rebounds (Team)
DRB_T	Defensive rebounds (Team)
BLK_T	Blocks (Team)
TOV_T	Turnovers (Team)
PF_T	Personal Fouls (Team)
X3_PCT_O	The percentage of 3-Point field goals made (Opponents)
X2_PCT_O	The percentage of 2-Point field goals made (Opponents)
FT_PCT_O	The percentage of free throws made (Opponents)
ORB_O	Offensive rebounds (Opponents)
DRB_O	Defensive rebounds (Opponents)
BLK_O	Blocks (Opponents)
TOV_O	Turnovers (Opponents)
PF_O	Personal Fouls (Opponents)
X3_PCT_ATT_T	Percentage of field goal attempts that were 3-Pointers (Team). This variable was included in all analysis for this project.
PTS_DIFF	Point Differential (Team Points - Opponents' Points)
Playoffs	Yes/No indicator of making the playoffs (Team)
Champion	Yes/No indicator of winning the championship (Team)

A sample of the final dataset from the most recent season (2021 – 2022) is shown in Appendix B. Appendix C contains the R code that performed the above process steps.

4 Making the Data Confess

4.1 Overview

The research question was investigated through three separate analyses. Two linear regression models were fit, and one cluster analysis was performed. The scope of each analysis is specified in the table below:

Table 3: Comparison of the Three Analysis Efforts

Technique	Candidate Features	Response Variable
Linear Regression (2 Features)	3-Point Accuracy (Team), 3-Point Attempt % (Team)	Point Differential
K-Means Clustering (2 Features)		N/A
Linear Regression (Stepwise Selection)	17 Initial Candidates, 13 Chosen	Point Differential

All the above analysis was performed on just the last three seasons of NBA data. This decision was based on the 10-Year trend that is shown in Figure 4. The blue trend line shows that 3-Point attempt counts have been more stable the last three seasons.

4.2 Linear Regression (2 Features)

4.2.1 Purpose

The goal of this model is to determine if team success varies by 3-Point Attempt %, when controlling for their 3-Point Accuracy. Team success is measured by Point Differential. The linear regression model therefore has Point Differential as the response variable, and the two features are 3-Point Accuracy and 3-Point Attempt %.

The theoretical formula for the regression model is shown below:

$$\text{Point Differential} = \beta_0 + \beta_1 * \text{3-Point Accuracy} + \beta_2 * \text{3-Point Attempt \%} + \in,$$

where β_0 = Unknown intercept parameter,
β_1 and β_2 = Unknown slope parameters, and
$\in$ = Error or residual term

The formula for the estimated regression model is shown below. The parameter estimates are least squares estimates, and so the sum of the squared distance between the 90 observed and estimated Point Differentials is minimized:

$$\text{Point Differential} = \hat{\beta}_0 + \hat{\beta}_1 * \text{3-Point Accuracy} + \hat{\beta}_2 * \text{3-Point Attempt \%},$$

where $\hat{\beta}_0$ = Estimated Intercept Coefficient, and
$\hat{\beta}_1$ and $\hat{\beta}_2$ = Estimated slope coefficients

The parameter estimate symbols are replaced with the estimated values in the formula below:

$$\text{Predicted Point Differential} = -70.40 + (183.56 * 3\ \text{Point Accuracy}) + (11.30 * 3\ \text{Point Attempt}\,\%)$$

4.2.2 Model Summary

Partial t-tests were used to test whether the true slope parameters (β_1 and β_2) differ from 0. In a model with two features, a Partial t-test measures the impact of a feature to a model that already contains the other feature. The t-test for 3-Point Accuracy is highly significant, based on its p-value ≈ 0. Since the estimated slope value ($\hat{\beta}_1 = 183.56$) is positive, we can conclude that 3-Point Accuracy has a positive association with Point Differential, when controlling for 3-Point Attempt %. The partial t-test for 3-Point Attempt % (β_2) yields a high p-value of 0.256. Since we cannot conclude that the true slope parameter for 3-Point Attempt % (β_2) differs significantly from zero, we also cannot state that 3-Point Attempt % has an association with Point Differential, when controlling for 3-Point Accuracy.

This fitted model yields a Coefficient of Determination (R^2) value of 0.4077. Therefore, 40.77% of the variability in Point Differential can be explained by two features. Almost all the contribution to R^2 is from 3-Point Accuracy, however, since a simple linear regression model with only 3-Point Accuracy achieves an R^2 value of 0.3988.

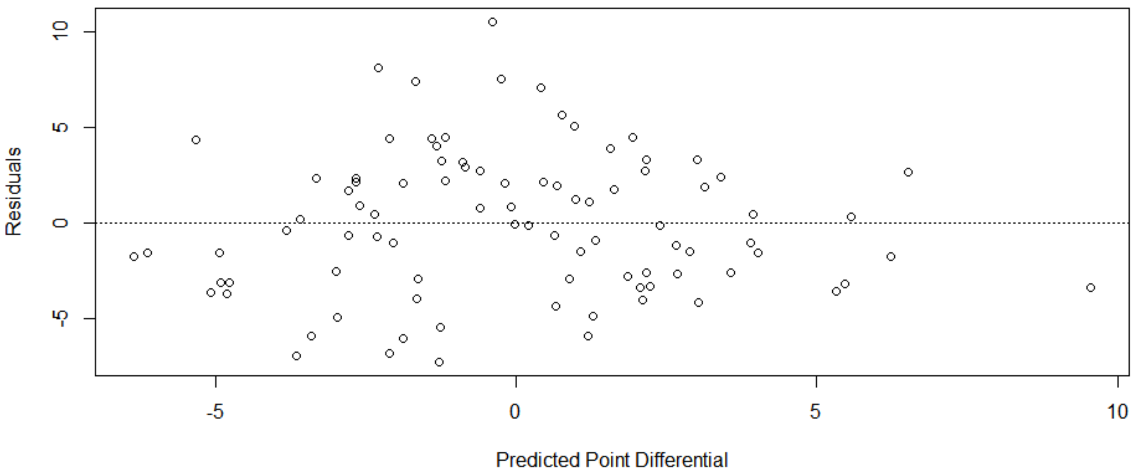
The 2 feature model was fit in R, and the summary output is found in Appendix D. Appendix E contains the code that fit the model.

4.2.3 Model Diagnostics: Residuals

In a linear regression model, a residual is the difference between the observed and predicted values of the response variable. Inferences regarding a linear regression model, such as interpretation of the p-values from the Partial t-tests, are based on the assumption that residuals are Normally and independently distributed with a mean of zero and a constant variance, or $\in_i \sim NID(0,\sigma^2)$.

The equal variance assumption was assessed using the figure below. It is a scatterplot of the residuals versus the predicted Point Differential values:

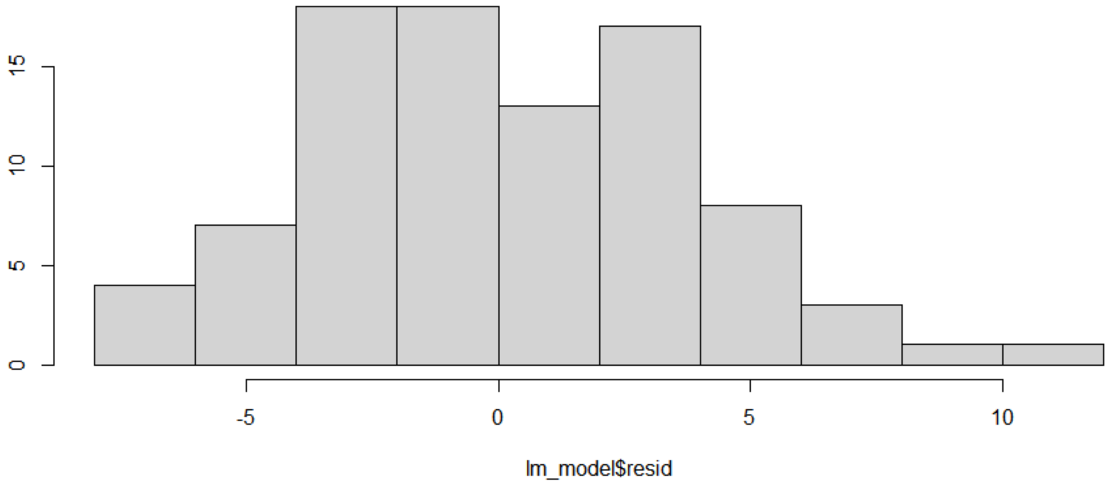
Figure 5: Residuals by Predicted Point Differential (2 Feature Model)



The above plot does not show a strong violation of the equal variance assumption. Although the range of residuals is narrow on the ends of the graph for very low and very high differentials, the number of team/season combinations is also very sparse. The residuals also do not exhibit a pattern as predicted Point Differentials change.

The assumption of Normally distributed residuals was assessed with two graphs and one statistical test. The first graph is the histogram in the figure below:

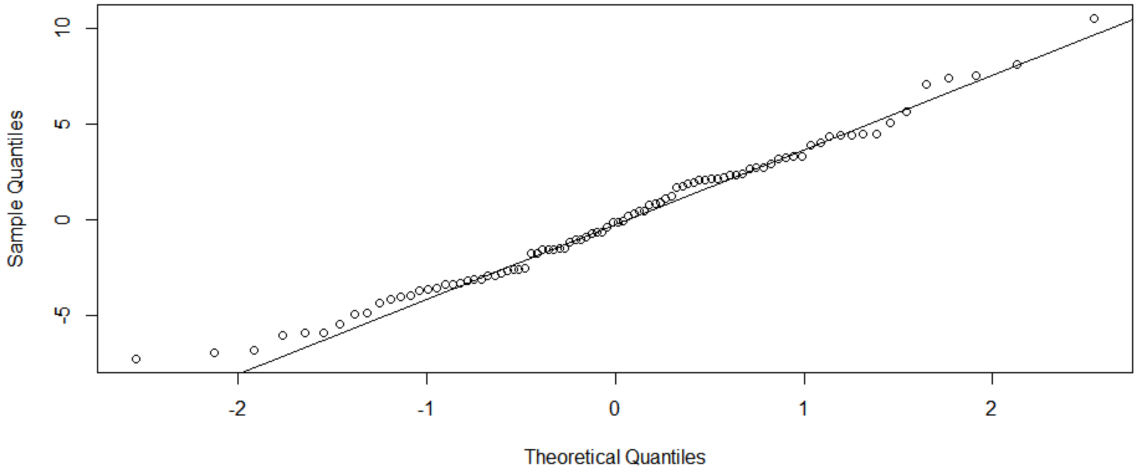
Figure 6: Histogram of the Residuals (2 Feature Model)



The above histogram resembles a Normal Distribution in that the largest bars are the two above and two below zero. However, the distribution is right-skewed, since the largest residuals in absolute value are positive residuals.

The second graph is a Q-Q Plot. The plot shows the sample quantiles of the residuals against the theoretical quantiles from a Normal distribution. If the residuals follow a Normal distribution, then they will be distributed near the line. The Q-Q Plot is shown in the figure below:

Figure 7: Q-Q Plot of the Residuals (2 Feature Model)



Since the residuals are right-skewed, and there is less dispersion in the negative residual values, the lowest negative residuals do not fall on the line. However, this does not appear to be a severe violation of the normality assumption.

The Jarque-Bera Test[8] simultaneously tests that the data follow the skewness and kurtosis of a Normal Distribution. Skewness is a measure of how much a distribution is asymmetric, which is contrary to the symmetry of a Normal Distribution around its mean. Kurtosis is a measure of the peak of a distribution, which is a value of 3 for a Normal Distribution. In the Jarque-Bera Test, the Null Hypothesis is that the skewness and kurtosis of the data both follow a Normal Distribution. Since the p-value for the Jarque-Bera test on the residuals is 0.4916, we do not have sufficient evidence that the residuals do not follow a Normal Distribution.

Based on these diagnostics, the assumptions of Normally distributed residuals with equal variance appear to hold for the 2-feature model. Independence of the residuals cannot be assessed for this model, since there is no time ordering of the team/season combinations.

4.2.4. Model Diagnostics: Multicollinearity

Multicollinearity is the empirical correlation among the values of the features. The calculation of the least squares estimates of the intercept and slope coefficients require inversion of a matrix that contains the feature values. If two or more features are highly correlated, then the inverted matrix is less stable, and interpretation of the regression parameters may defy our intuition. For example, with severe multicollinearity, the sign (positive or negative) of a slope parameter for a feature may be opposite of what would result in a simple linear regression that contains only that feature.

Multicollinearity was diagnosed for this model using variance inflation factors (VIF). A VIF value is calculated for each feature, and it measures the proportion increase in the variance of the feature's estimated slope that is caused by multicollinearity. A value of 1 indicates that no multicollinearity exists. Severe multicollinearity is present when any VIF value is 5 or higher[9].

In a model with two features, the VIF values are identical for both. In this model, both 3-Point Accuracy and 3-Point Attempt % have VIF values of 1.01. Therefore, since this value is slightly higher than 1, the model has negligible multicollinearity.

The R code that performed the residual and multicollinearity diagnostics is found in Appendix F.

4.3 K-means Clustering (2 Features)

4.3.1 Purpose

This analysis clustered team/season combinations that have similar 3-Point Accuracy and 3-Point Attempt % values. The resulting clusters were then compared and contrasted. The K-means clustering algorithm seeks to minimize the variability among observations within clusters, and also to maximize the variability among clusters. It is unsupervised learning, and so there is no response variable. However, one variable on which the clusters were compared in Section 5.2 is Point Differential, which is the response variable from the two regression models. The clusters were also compared on their rates of playoff appearances and NBA Championships in Section 5.2.

4.3.2 Feature Standardization

One necessary pre-processing step was to standardize the values of the two features. Unlike the linear regression models in this project, K-means clustering is sensitive to the scales of the features. Therefore, to ensure that both features can contribute to the cluster assignments, both 3-Point Accuracy and 3-Point Attempt % were converted to their corresponding Z-Scores, or Standard Normal scores. Z-Score standardization is performed as follows:

1. Calculate the mean and standard deviation for each feature. For example, the mean and standard deviation for 3-Point Attempt % are 39.18% and 4.06%, respectively.

2. For each feature observation, subtract its corresponding mean, and then divide the difference by its corresponding standard deviation. For example, for a 3-Point Attempt % of 40%, the Z-Score is (40% - 39.18%) / 4.06%, or 0.202.

After standardization, the new observations for both features have a mean of 0 and standard deviation of 1.

Although both 3-Point Accuracy and 3-Point Attempt % values theoretically range from 0 – 100%, the observed distributions of the features differed enough to warrant standardization. The mean and standard deviation of 3-Point Accuracy are 35.91% and 1.63%, respectively, while the 3-Point Attempt % mean and standard deviation are 39.18% and 4.06%, respectively. The standard deviation for 3-Point Attempt % is therefore over twice as large as the 3-Point Accuracy standard deviation.

4.3.3 Methodology

The K-means clustering algorithm is executed in the following manner:

1. Choose a value for the number of clusters (K).

2. K cluster centroids are randomly generated.

3. For each observation, calculate the Euclidean Distance to each centroid. For this analysis, an observation is one of the 90 team/season combinations from the last three seasons.

4. Form clusters by assigning each observation to the centroid that is the closest in Euclidean Distance.

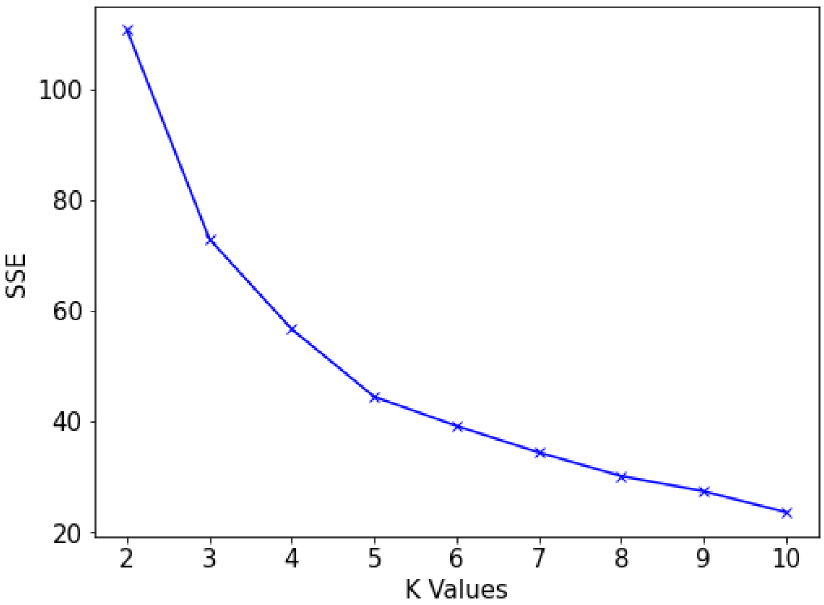
5. Within each cluster, calculate new centroid values based on the feature means for the observations that are assigned to the cluster. For this analysis, the pair of mean values for 3-Point Accuracy and 3-Point Attempt % is calculated within each cluster.

The above five steps are repeated until team/season combinations no longer migrate to new clusters, and therefore the cluster centroid values also remain unchanged.

4.3.4 Optimum Number of Clusters (K)

Two metrics were considered for determining the optimum number of clusters. They are the Elbow Method and Silhouette Coefficient. The Elbow Method is based on a graph of Mean Sum of Squares Error (SSE) by the number of clusters (K). Within each cluster, SSE is the sum of the squared Euclidean Distances between each observation and the cluster centroid. The Mean SSE is the sum of the SSE values for each cluster divided by the number of clusters. Mean SSE values decrease as K increases, and they converge to zero when the number of clusters equals the number of observations. However, it is more difficult do draw useful conclusions from the data when there are too many clusters. The Elbow Method balances the competing goals of obtaining low variability within clusters and making useful inferences from them. The figure below shows a plot of Mean SSE values for cluster counts from 2 – 10:

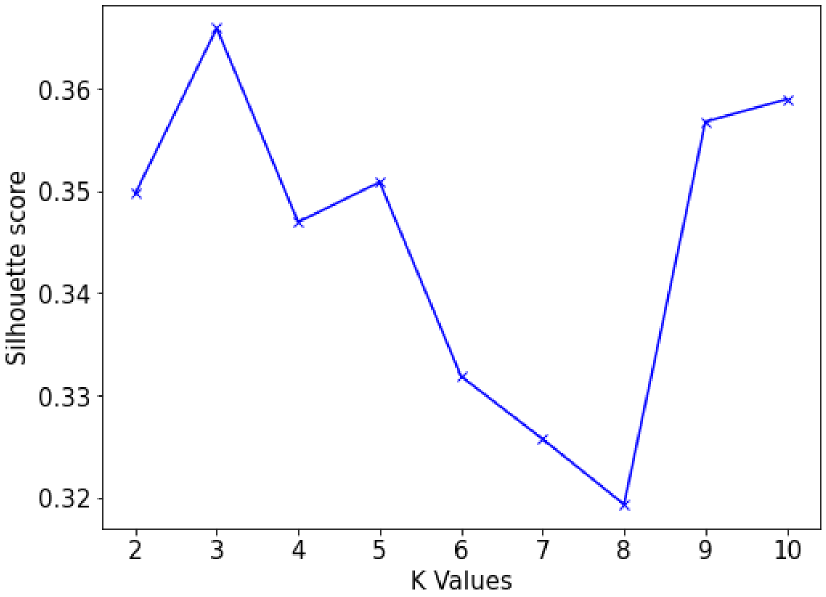
Figure 8: Line Graph of Mean SSE by the Number of Clusters (K)



If the above line is interpreted like an arm, the elbow is where the arm bends. The elbow reflects the transition from sharp decreases in Mean SSE as K is increased to small Mean SSE decreases as K is increased. Based on the Elbow Method, K values of 3, 4, and 5 were all considered viable.

The figure below shows a plot of Silhouette Coefficient values for cluster counts from 2 – 10:

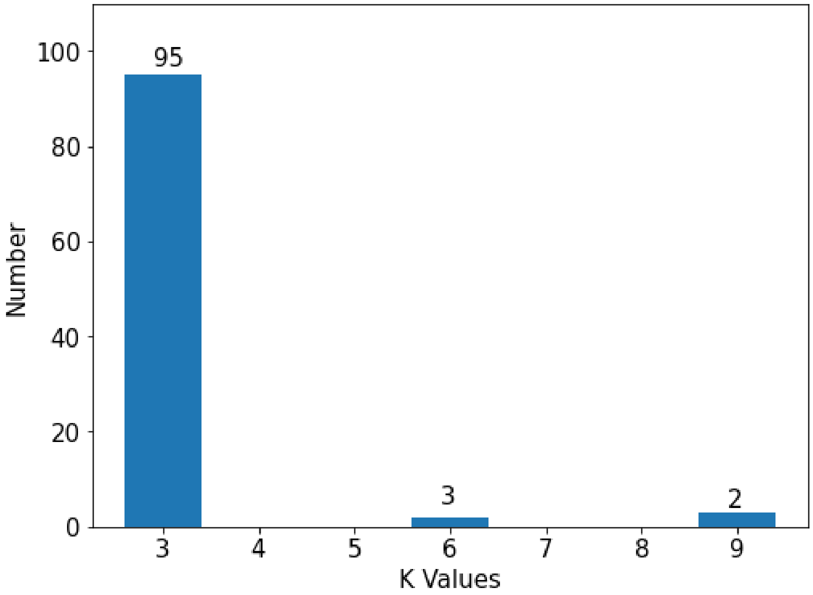
Figure 9: Line Graph of Silhouette Coefficients by the Number of Clusters (K)



Based on the above graph, 3 clusters were chosen, since it produced the largest Silhouette Coefficient. This choice is also consistent with the conclusion that was drawn from the Elbow Method.

Cluster assignments are sensitive to the random choice of the initial cluster centroids. Therefore, the choice of K = 3 was validated by running multiple simulations. Within each simulation round, the clustering algorithm was run for nine K values from 2 – 10. The value of K that produced the highest Silhouette Coefficient was then recorded. 100 simulation rounds were performed, and the results are summarized in the figure below. This bar chart shows that the choice of 3 clusters was verified, since it was the optimum value in 95 of the 100 simulations:

Figure 10: Bar Graph of K Values that Produced the Highest Silhouette Coefficient



One challenge with K-means clustering is that large disparities in observation counts can occur among the clusters. This did not occur in the final cluster assignments, as Clusters 1, 2 and 3 contain 26, 29 and 35 team/season combinations, respectively.

The Python code that performed this analysis if found in Appendix G.

4.4 Linear Regression (Stepwise Selection)

4.4.1 Purpose

Like the model with 2 features, the goal of this linear regression model is to determine if team success varies by 3-Point Attempt %. This model also has the same response variable, namely Point Differential. However, the impact of 3-Point Attempt % is being measured while controlling on more features. Specifically, 17 features including 3-Point Attempt % were initial candidates, and 13 features were chosen for the final model. The 13 chosen features are listed below:

1. 3-Point Accuracy (Team)
2. 2-Point Accuracy (Team)
3. Free Throw Accuracy (Team)
4. Offensive Rebounds (Team)
5. Turnovers (Team)
6. Personal Fouls (Team)
7. Three Point Attempt % (Team)
8. 3-Point Accuracy (Opponents)
9. 2-Point Accuracy (Opponents)
10. Free Throw Accuracy (Opponents)
11. Offensive Rebounds (Opponents)
12. Turnovers (Opponents)
13. Personal Fouls (Opponents)

4.4.2 Stepwise Selection Methodology

Stepwise selection was performed by the Step function in R. This default function uses the Akaike Information Criterion (AIC) to compare candidate models. The formula for AIC is 2*p – 2*LL, where p is the number of features in the model, and LL is the log-likelihood of the model. Since lower values of AIC are better, the 2*p portion of the formula provides a penalty as features are added to the model, which helps to prevent overfitting. The process that was followed by the Step function is listed below:

1. Fit a model with all features.

2. Calculate the AIC value for this model.

3. Sort the features in ascending order by the AIC value that would be achieved if the feature was removed.

4. Remove the feature with the lowest AIC value from the model, provided that the AIC value with the feature removed is better (lower) than the current model's AIC.

The above four steps are repeated until the following two conditions are met:

1. AIC cannot be improved by removing any features.

2. AIC cannot be improved by returning a previously removed feature to the model.

Four iterations of stepwise selection were performed for this model. Four features were removed and were never returned to the model. The order in which the four features were removed is listed below:

1. Blocks (Team)
2. Blocks (Opponents)
3. Defensive Rebounds (Team)
4. Defensive Rebounds (Opponents)

The Step function has a parameter k, which adjusts the default penalty in the AIC formula. The default value of k is 2. When k is increased, fewer features get selected for the final model. Different k values were attempted with the goal of obtaining a model that both includes 3-Point Attempt %, but does not include other features with a high p-value. These conflicting goals were achieved with a value of k = 3.

4.4.3 Model Summary

Like the 2 feature regression model, Partial t-tests were used to test whether the true slope coefficients differ from 0. For all 12 features besides 3-Point Attempt %, the Partial t-test p-values ≈ 0. Therefore, we can conclude that the true slope coefficients for these 12 features are non-zero, and therefore they have an association with Point Differential. This fitted model yields a Coefficient of Determination (R^2) value of 0.99. Therefore, all but 1% of the variability in Point Differential can be explained by only 13 features.

The p-value for the 3-Point Attempt % Partial t-test is 0.0538. Although this is stronger than the 0.256 p-value that was achieved in the 2 feature model, 3-Point Attempt % still does not have a strong association with Point Differential. This conclusion is based on the following three reasons:

- The magnitude of the estimated 3-Point Attempt % slope coefficient is smaller in the stepwise model than the 2 feature model. The slope value is only 3.0 in the stepwise model, as compared to 11.30 in the 2 feature model. The interpretation of the 3.0 value is that a 1% increase in 3-Point Attempt % would produce only a negligible 0.03 increase in Point Differential.

- The better overall quality of fit in the stepwise model will produce more statistical significance than the 2 feature model, even for variables where practical significance does not occur. The Root Mean Squared Error (RMSE) is only 0.52 in the stepwise model, which is 86.2% lower than the 3.76 value from the 2 feature model.

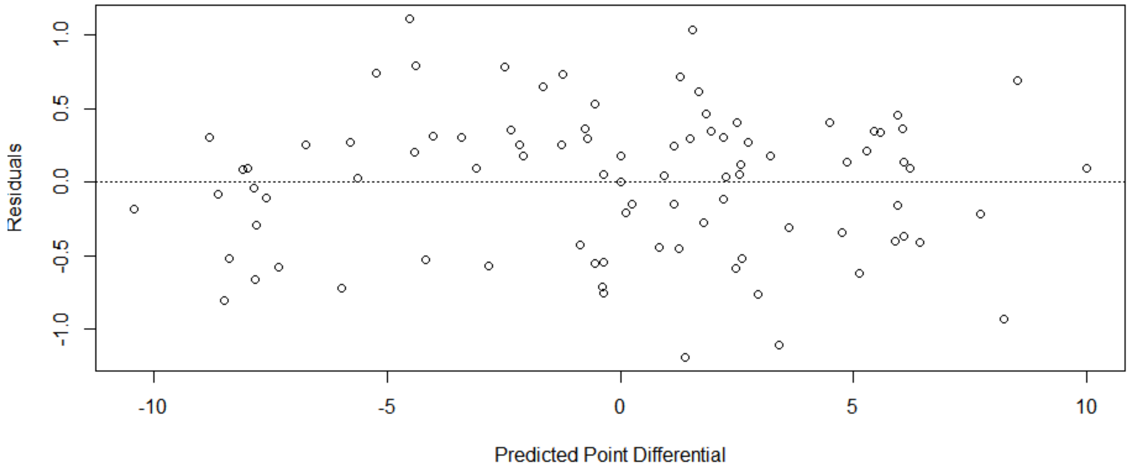
- If 3-Point Attempt % is removed from the stepwise model, R^2 decreases negligibly from 0.9900 to 0.9895.

This stepwise model was fit in R, and the summary output is found in Appendix H. Appendix I contains the code that fit the model.

4.4.4 Model Diagnostics: Residuals

The analysis of residuals followed the same approach that was used for the 2-feature model. The equal variance assumption was assessed using the figure below. It is a scatterplot of the residuals versus the predicted Point Differential values:

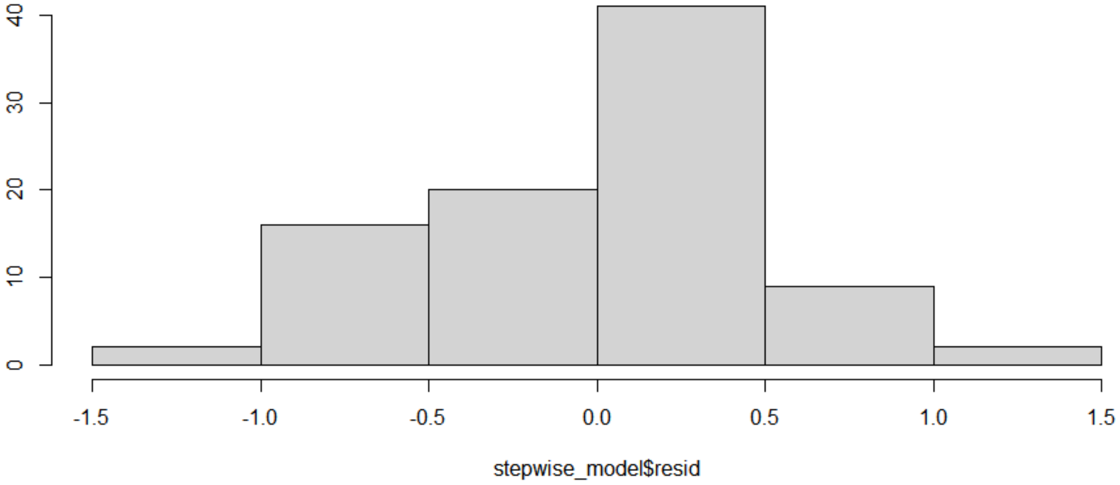
Figure 11: Residuals by Predicted Point Differential (Stepwise Model)



The spread of the residuals is generally consistent with the equal variance assumption. The small exception is for predicted Point Differential values near -5. The range of residuals is narrower in this region, since the residuals tend to be positive, and therefore the model is tending to underestimate Point Differential for these team/season combinations.

The assumption of Normally distributed residuals was assessed with two graphs and one statistical test. The first graph is the histogram in the figure below:

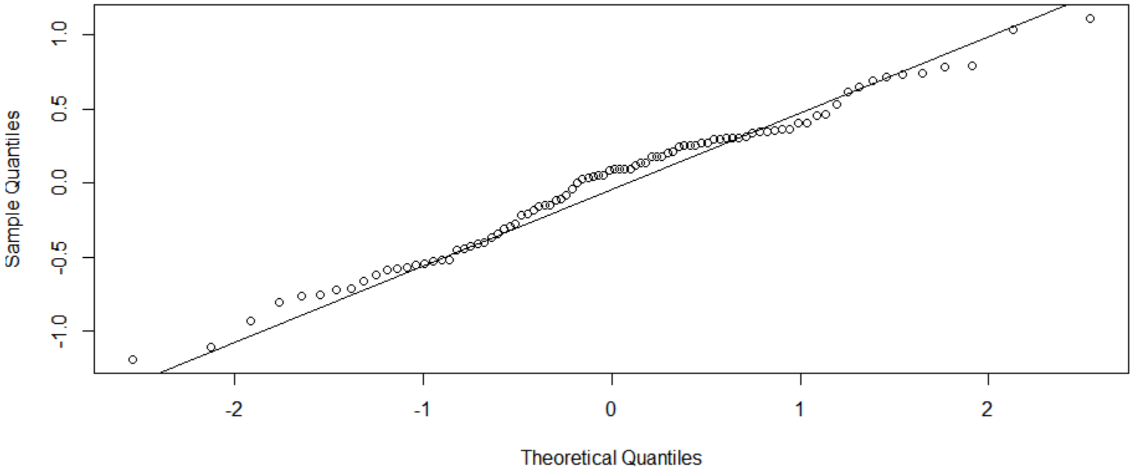
Figure 12: Histogram of the Residuals (Stepwise Model)



The above histogram resembles a Normal Distribution, since there is a single peak towards the middle of the distribution, and the bars become smaller as the absolute value of the residuals increase. The only caveat is the high frequency of small positive residuals between 0 – 0.5, as compared to the frequency of small negative residuals between -0.5 – 0.

The second graph is a Q-Q Plot. The plot shows the sample quantiles of the residuals against the theoretical quantiles from a Normal Distribution. If the residuals follow a Normal Distribution, then they will be distributed near the line. The Q-Q Plot is shown in the figure below:

Figure 13: Q-Q Plot of the Residuals (Stepwise Model)



Since the residuals tend to reside near the line, they appear to be consistent with a Normal Distribution. The separation from the line towards the middle of the plot is due to the high concentration of low positive residuals that was shown in the histogram.

The Jarque-Bera Test was used to simultaneously test that the data follow the skewness and kurtosis of a Normal Distribution. Since the p-value for the test is 0.5933, we do not have sufficient evidence that the residuals do not follow a Normal Distribution.

Based on these diagnostics, the assumptions of Normally distributed residuals with equal variance appear to hold for the stepwise model.

4.4.5 Model Diagnostics: Multicollinearity

VIF values were used to evaluate multicollinearity in the same manner as they were used in the 2-feature model. The VIF values for each slope parameter are shown in the table below:

Table 4: Variance Inflation Factors (VIF) for the Stepwise Model

Feature	VIF
Free Throw Accuracy (Team)	2.08
Personal Fouls (Team)	1.86
Turnovers (Team)	1.82
3-Point Accuracy (Team)	1.55
Turnovers (Opponent)	1.54
Personal Fouls (Opponent)	1.44
Offensive Rebounds (Team)	1.39
2-Point Accuracy (Team)	1.38
3-Point Accuracy (Opponent)	1.36
3-Point Attempt % (Team)	1.27
2-Point Accuracy (Opponent)	1.26
Free Throw Accuracy (Opponent)	1.23
Offensive Rebounds (Opponent)	1.12

The largest VIF is 2.08, and it corresponds to the slope parameter for Free Throw Accuracy (Team). Since this value is below 5, strong multicollinearity does not exist for the Stepwise Model.

The R code that performed the residual and multicollinearity diagnostics is found in Appendix J.

5 Storytelling with Data

5.1 Linear Regression (2 Features)

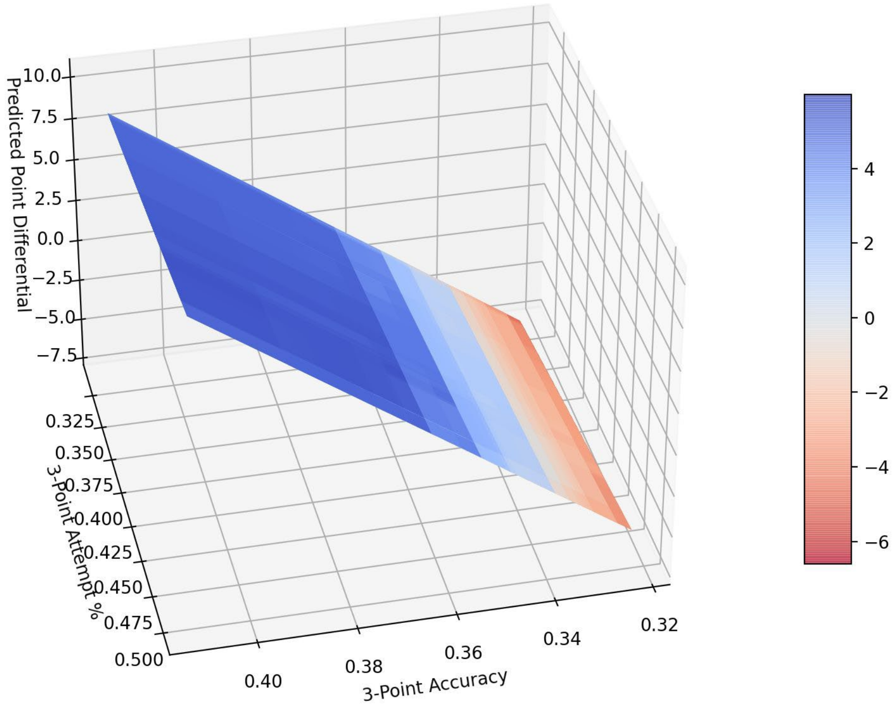
This section contains three graphical representations of the linear regression model with 2 features. The first visual was added based on peer review feedback of the trial presentation video that was shown in class. In the video, only the second and third visuals were presented, and there was feedback that more context about the model equation should have first been provided. The equation that was estimated by least squares is shown below:

$$\text{Predicted Point Differential} = -70.40 + (183.56 * \text{3 Point Accuracy}) + (11.30 * \text{3 Point Attempt \%})$$

The equation shows that a 1% (0.01) change in 3-Point Accuracy produces a predicted increase in Point Differential of 1.8356. However, the Predicted Point differential increases by only 0.113 when 3-Point Attempt % increases by 1% (0.01). The ratio of the 3-Point Accuracy and 3-Point Attempt % slope coefficients is 16.24.

The following 3-Dimensional graph is a representation of the above regression equation. The two features are 3-Point Accuracy and 3-Point Attempt %, and they are displayed on the x-axis and y-axis, respectively. The predicted Point Differential values that correspond to each pair of feature values are represented by a plane on the z-axis:

Figure 14: 3-Dimensional Representation of the 2 Feature Regression Model

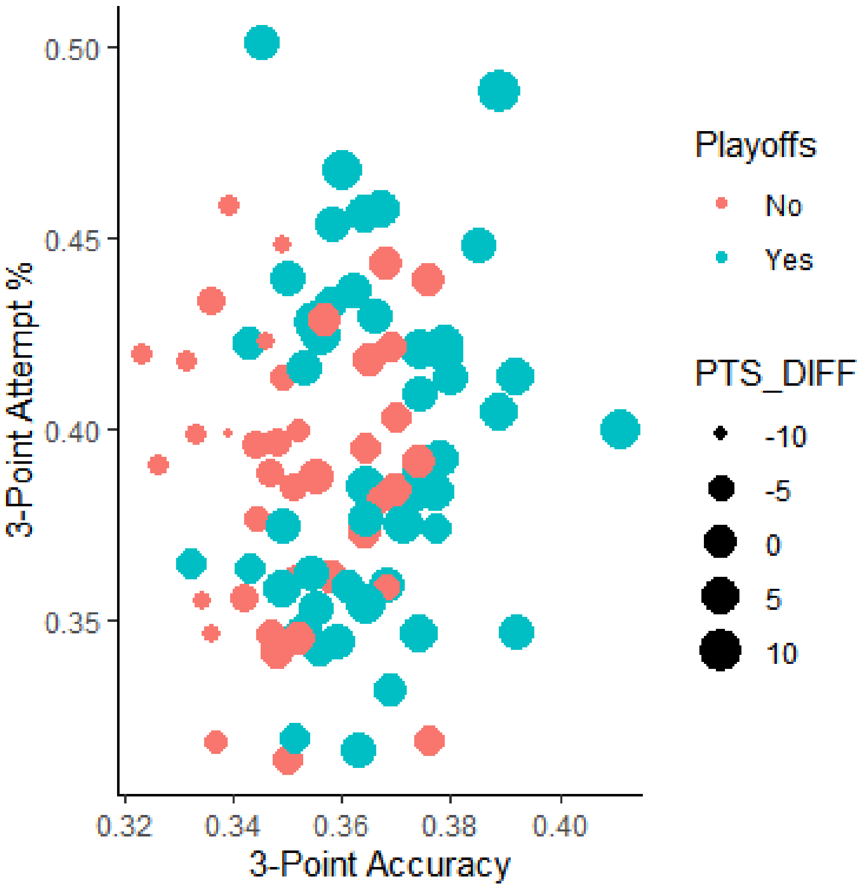


The graph confirms that 3-Point Accuracy is the dominant feature in the regression equation. The plane is steep when moving along the 3-Point Accuracy axis (x-axis), and shallow when moving along the 3-Point Attempt % axis (y-axis). The color changes also illustrate the dominance of 3-Point Accuracy. Red bands represent negative predicted Point Differentials, while positive differentials are represented by the blue bands. The color changes are pronounced when moving along the 3-Point Accuracy axis, but are negligible when moving along the 3-Point Attempt % axis.

The Python code that created the 3D plot is found in Appendix K.

The scatterplot below is also a representation of the linear regression model with 2 features. The two features are a team's 3-Point Accuracy and 3-Point Attempt %, and they are represented on the x-axis and y-axis, respectively. The size of the circles represents Point Differential. Larger circles are better, since a higher Point Differential means that a team outscores their opponent by a larger margin:

Figure 15: Scatterplot Representation of the 2-Feature Regression Model (Playoff Teams are Shaded in Blue)



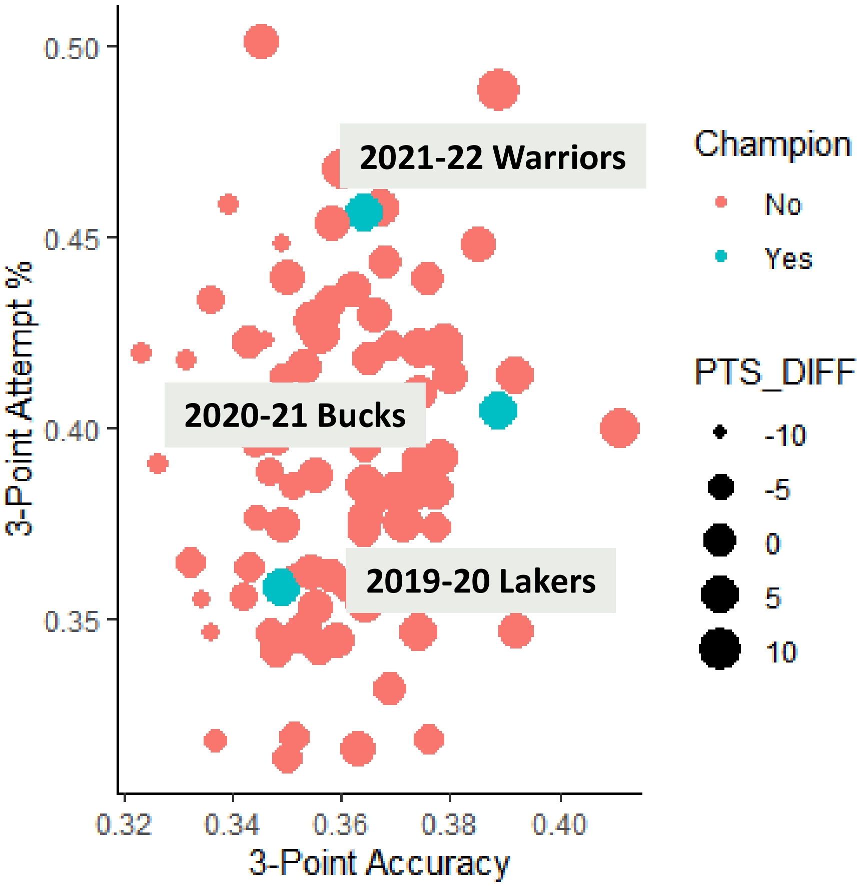
The above graph shows that higher Point Differential values are associated more with 3-Point Accuracy than 3-Point Attempt %. 3-Point Accuracy improves when moving from left to right along the x-axis. When moving in this direction, Point Differentials also improve, as denoted by the larger circles. However, when moving up and down along the y-axis, there is a mixture of large and small circles. This illustrates that 3-Point Attempt % is not strongly associated with Point Differential.

The graph interpretation is consistent with the conclusions that were drawn from the partial t-tests in the "Making the Data Confess" section. The partial t-test for 3-Point Accuracy was highly significant with a p-value near 0, whereas the t-test for the 3-Point Attempt % coefficient yielded a high p-value of 0.256.

The graph also shows that playoff appearances are more associated with 3-Point Accuracy than 3-Point Attempt %. Since 16 teams make the playoffs each season, there were 48 playoff teams over the three analyzed seasons. These 48 teams are shaded in blue. When moving along the 3-Point Accuracy axis from left to right, there is an increase in the proportion of playoff teams. However, there is a mixture of playoff and non-playoff teams when moving up and down the y-axis.

The figure below is identical to the Figure 15 scatterplot, except that the NBA Champions from the last three seasons are highlighted in blue:

Figure 16: Scatterplot Representation of the 2-Feature Regression Model (NBA Champions are Shaded in Blue)



The graph shows that the attempting a higher percentage of 3-Point shots is not an indicator of winning championships. The last three NBA Champions have very different playing styles, as their 3-Point Attempt % values span a wide range. These values are reported in the table below, along with their corresponding percentiles:

Table 5: 3-Point Attempt % Values for the Last 3 NBA Champions

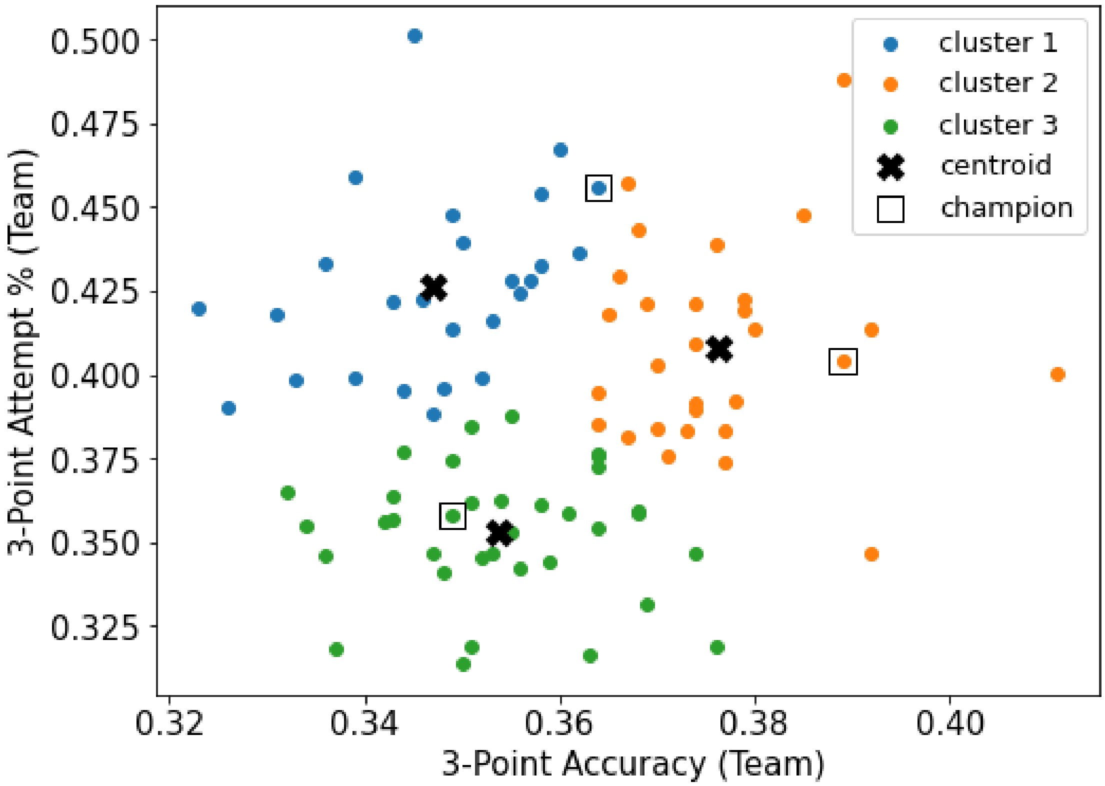
Champion	3-Point Attempt %	Percentile
2019-20 Lakers	35.8%	23.3%
2020-21 Bucks	40.4%	63.3%
2021-22 Warriors	45.6%	94.4%

The percentile range in 3-Point Attempt % is 71.1% among the last three NBA Champions. The lowest percentile of 23.3% was achieved by the 2019-2020 Los Angeles Lakers, while a percentile of 94.4% was achieved by the 2021-2022 Golden State Warriors. The Warriors are the team for whom Stephen Curry plays.

5.2 K-means Clustering (2 Features)

The cluster assignments of the 90 team/season combinations are color-coded in the scatterplot below:

Figure 17: Scatterplot of the 2-Feature Cluster Assignments



The three clusters are also profiled in the table below:

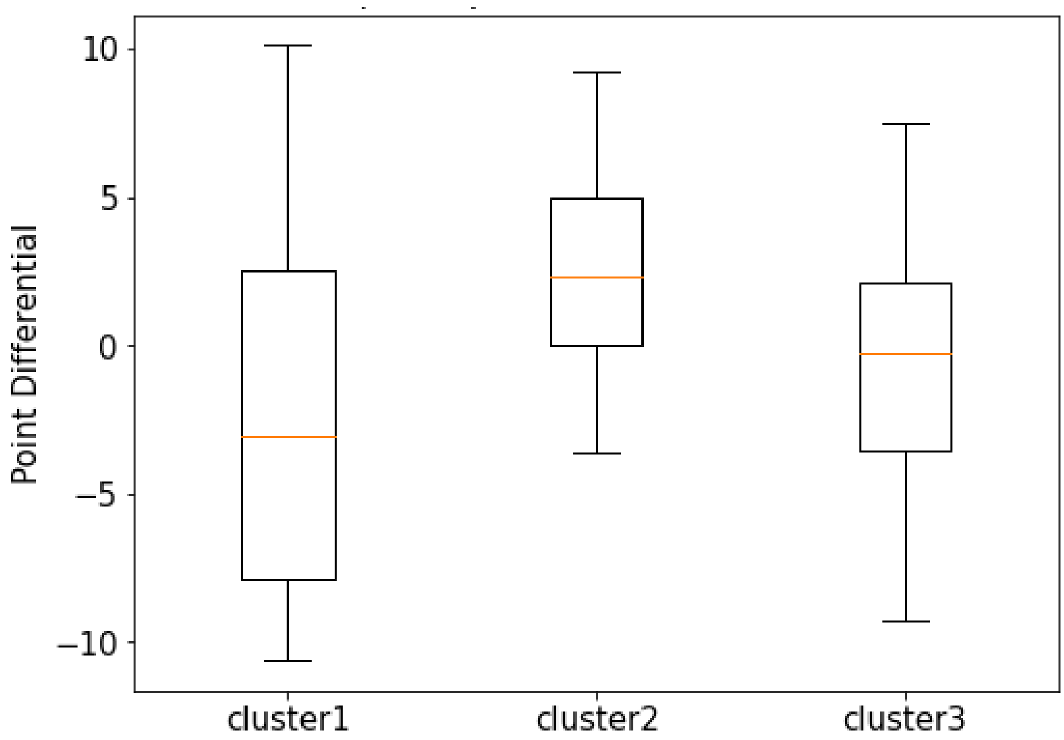
Table 6: Summary of the Three Clusters

Cluster	Team/Season Count	3-Point Accuracy	3-Point Attempt %	Playoff %	NBA Champion
1	26	Lowest	Highest	23%	1
2	29	Highest	Medium	42%	1
3	35	Medium	Lowest	35%	1

Cluster 1 attempts the highest percentage of 3-Point shots, but they are also the least accurate. Their percentage of teams that make the playoffs is only 23%, which is the lowest among the three clusters. Cluster 2 achieves the highest accuracy with medium rates of 3-Point shot attempts. They also achieve the highest rate of playoff teams (42%). Cluster 3 is the most conservative in attempting 3-Point shots, and they achieve medium accuracy. Their playoff percentage of 35% is between the rates that were achieved by Clusters 1 and 3. Each cluster also contains one of the last three NBA Champions. This supports the prior conclusion that the recent champions have different playing styles.

The distributions of Point Differential values by cluster are compared below with boxplots:

Figure 18: Comparative Boxplots of Point Differential Values by Cluster

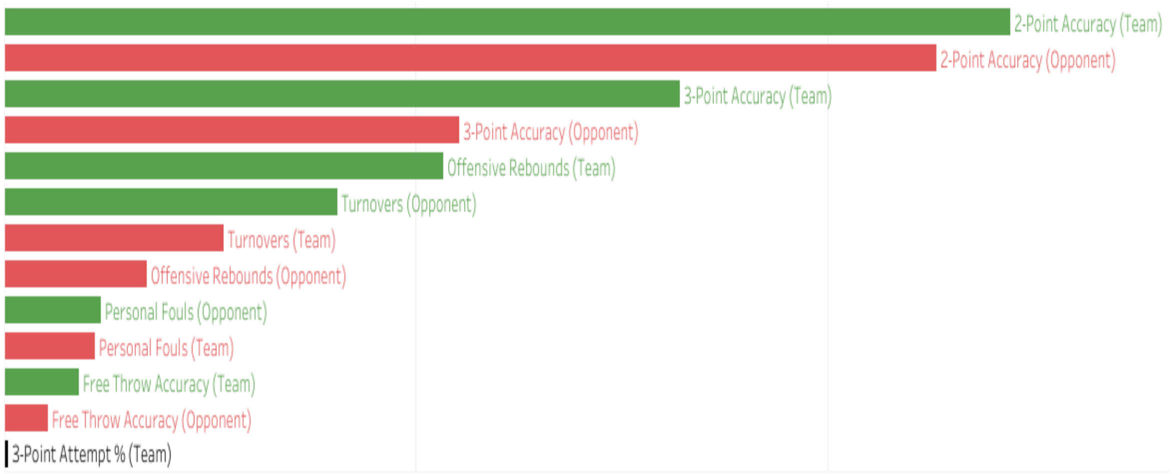


Cluster 1 attempts the highest rate of 3-Point shots with the lowest accuracy. The team/season combinations within Cluster 1 achieve the worst point average Point Differential values. This cluster also has the highest spread of Point Differential values, which indicates that their strategy is high risk with varied rewards. Cluster 2 achieves the best 3-Point Accuracy with a medium rate of attempts. This high accuracy produces the best average Point Differential values, and also the least variation. On average, Cluster 3 achieves a neutral Point Differential that is close to 0. They are taking less risk in attempting 3-Point shots than the other two clusters.

5.3 Linear Regression (Stepwise Selection)

The bar graph below was created in Tableau, and it rank-orders the 13 features from best to worst. The ranking is based on the partial sum of squares that the feature achieved. In the context of this model, a feature's partial sum of squares value measures its contribution to a model that already contains the other 12 features:

Figure 19: Bar Graph of Partial Sums of Squares from the Stepwise Regression Model



In the above graph, positive qualities for a team are shaded in green, while red bars denote negative qualities. 3-Point Attempt % is shaded in a neutral color of black. The size of the 3-Point Attempt % bar shows that it has a negligible contribution to a model that already contains the other 12 features. Therefore, 3-Point Attempt % does not show a strong association with Point Differential.

The graph interpretation is consistent with the conclusions that were drawn from the partial t-tests in the "Making the Data Confess" section. The partial t-tests for the 12 features besides 3-Point Attempt % are highly significant with p-values near 0. However, the t-test for the 3-Point Attempt % coefficient yielded a moderately high p-value of 0.0538.

6 Conclusions and Recommendations

Based on the analysis, the following three conclusions were drawn:

- There is not an optimum mix of 2-Point and 3-Point shot attempts.
- The mix of shot attempts varied greatly among the last three NBA Champions.
- In general, teams should prioritize improving the accuracy of their shooting over shot selection.

One limitation of the data is that specific game situations could not be studied. A potential research project is to assess the optimum mix of shots by game situation. For example, if a team is trailing by 20 points with 5 minutes left in a game, they would likely increase their rate of 3-Point shooting, due to the limited time to achieve a comeback victory. However, some 2-Point attempts may still be justified, since the opponent will be expecting more 3-Point shots and adjusting their defense accordingly.

7 References

1 NBA Valuations:
https://www.forbes.com/sites/mikeozanian/2021/10/18/the-nbas-most-valuable-teams-2021-22-new-york-knicks-lead-a-trio-now-worth-over-5-billion-each/?sh=72d8ea2cd32c

2 NBA Revenue:
https://www.cnbc.com/2021/10/18/nba-2021-2022-season-10-billion-revenue-tv-viewership-rebound.html

3 NBA Popularity:
https://runrepeat.com/nba-popularity

4 Stephen Curry:
https://www.sportskeeda.com/basketball/stephen-curry-nba-s-3-Point-revolution

5 NBA Court Diagram:
The diagram was obtained from www.conceptdraw.com, and the green and yellow colors were added using Paint software.

6 NBA Salary Cap:
https://www.investopedia.com/articles/investing/070715/nbas-business-model.asp

7 Basketball Reference:
https://www.basketball-reference.com

8 Jarque-Bera Test:
https://www.statology.org/how-to-conduct-a-jarque-bera-test-in-r/

9 Variance Inflation Factors (VIF):
https://www.statology.org/multicollinearity-regression/

Appendix A: NBA Team Statistics for the 2021 – 2022 Season

Rk	Team	G	MP	FG	FGA	FG%	3P	3PA	3P%	2P	2PA	2P%	FT	FTA	FT%	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS
1	Minnesota Timberwolves*	82	241.2	41.6	91	0.457	14.8	41.3	0.358	26.8	49.7	0.54	18	23.1	0.778	11.2	32.9	44.2	25.7	8.8	5.6	14.3	21.8	115.9
2	Memphis Grizzlies*	82	241.2	43.5	94.4	0.461	11.5	32.7	0.353	32	61.7	0.519	17	23.1	0.734	14.1	35	49.2	26	9.8	6.5	13.2	19.8	115.6
3	Milwaukee Bucks*	82	240.9	41.8	89.4	0.468	14.1	38.4	0.366	27.8	51	0.544	17.8	22.9	0.776	10.2	36.5	46.7	23.9	7.6	4	13.4	18.2	115.5
4	Charlotte Hornets	82	242.4	42.8	91.4	0.468	13.9	38.2	0.365	28.8	53.3	0.542	15.8	21.4	0.74	10.8	33.7	44.6	28.1	8.6	4.9	13.3	19.9	115.3
5	Phoenix Suns*	82	240.6	43.7	90.1	0.485	11.6	31.9	0.364	32.1	58.2	0.551	15.9	19.9	0.797	9.8	35.5	45.3	27.4	8.6	4.4	12.9	19.9	114.8
6	Atlanta Hawks*	82	240.3	41.5	88.3	0.47	12.9	34.4	0.374	28.6	53.9	0.531	18.1	22.3	0.812	10	33.9	44	24.6	7.2	4.2	11.9	18.7	113.9
7	Utah Jazz*	82	240.6	40.6	86.2	0.471	14.5	40.3	0.36	26	45.8	0.568	17.9	23.4	0.767	10.8	35.6	46.3	22.4	7.2	4.9	14	18.9	113.6
8	San Antonio Spurs	82	241.5	43.2	92.7	0.467	11.3	32	0.352	32	60.7	0.527	15.4	20.4	0.754	11	34.3	45.3	27.9	7.6	4.9	12.7	18.1	113.2
9	Brooklyn Nets*	82	240.9	42	88.4	0.475	11.5	31.7	0.361	30.5	56.7	0.538	17.5	21.7	0.805	10.3	34.1	44.4	25.3	7.1	5.5	14.1	20.4	112.9
10	Denver Nuggets*	82	241.5	41.7	86.3	0.483	12.7	35.9	0.353	29	50.4	0.575	16.7	21	0.795	9.2	34.9	44.1	27.8	7.2	3.7	14.5	20	112.7
11	Los Angeles Lakers	82	243.7	41.6	88.8	0.469	12	34.5	0.347	29.7	54.3	0.546	16.8	23	0.732	9.5	34.5	44	24	7.6	5.2	14.5	20.2	112.1
12	Boston Celtics*	82	242.7	40.7	87.4	0.466	13.2	37.1	0.356	27.5	50.3	0.547	17	20.9	0.816	10.5	35.5	46.1	24.8	7.2	5.8	13.6	18.5	111.8
13	Chicago Bulls*	82	240.6	41.7	86.9	0.48	10.6	28.8	0.369	31.1	58.1	0.535	17.5	21.5	0.813	8.7	33.7	42.3	23.9	7.1	4.1	12.8	18.8	111.6
14	Indiana Pacers	82	242.4	41.4	89.5	0.463	12.2	35.4	0.344	29.3	54.1	0.541	16.4	21.4	0.768	11.3	32.6	43.9	25.4	7.1	5.6	14.4	20.4	111.5
15	Golden State Warriors*	82	240.6	40.5	86.4	0.469	14.3	39.4	0.364	26.2	47	0.557	15.6	20.3	0.769	9.8	35.7	45.5	27.1	8.8	4.5	14.9	21	111
16	Sacramento Kings	82	241.5	40.5	88.1	0.46	11.4	33.2	0.344	29.1	54.9	0.53	17.9	23.3	0.768	9.6	33.4	42.9	23.7	7.2	4.5	14.1	18.9	110.3
17	Miami Heat*	82	242.1	39.6	84.8	0.467	13.6	35.8	0.379	26	49	0.531	17.3	21.4	0.808	9.8	33.9	43.7	25.5	7.4	3.2	14.6	20.5	110
18	Philadelphia 76ers*	82	241.5	39.4	84.5	0.466	11.6	31.8	0.364	27.8	52.7	0.527	19.6	23.8	0.821	8.5	33.8	42.3	23.7	7.7	5.3	12.5	19.4	109.9
19	Houston Rockets	82	240.9	39.4	86.4	0.456	13.5	38.7	0.349	25.9	47.7	0.543	17.5	24.5	0.713	9.6	32.4	42	23.6	7.3	4.7	16.5	20.6	109.7
20	Toronto Raptors*	82	242.1	40.6	91.3	0.445	11.9	34.2	0.349	28.7	57.1	0.503	16.2	21.3	0.759	13.4	32	45.3	22.1	9	4.6	12.5	19.6	109.4
21	New Orleans Pelicans*	82	240.9	40.2	88	0.457	10.6	32.1	0.332	29.5	55.9	0.528	18.3	23.2	0.789	12	33.2	45.2	25	8.3	4	14.1	19.7	109.3
22	Washington Wizards	82	241.8	40.6	86	0.472	10.5	30.6	0.342	30.1	55.4	0.543	17	21.7	0.783	9	34.1	43.1	25	6.4	5	13.1	18.8	108.6
23	Los Angeles Clippers	82	241.2	40.1	87.4	0.458	12.8	34.2	0.374	27.3	53.3	0.512	15.5	19.6	0.793	9.1	34.9	44	24	7.4	5	13.7	18.6	108.4
24	Dallas Mavericks*	82	240.9	39.3	85.1	0.461	13.1	37.4	0.35	26.2	47.8	0.548	16.4	21.2	0.771	9.3	33.8	43	23.4	6.7	4	12.5	19.7	108
25	Cleveland Cavaliers	82	240.6	39.7	84.6	0.469	11.6	32.8	0.355	28.1	51.9	0.541	16.8	22.1	0.76	10.2	34	44.2	25.2	7.1	4.2	14.4	17.5	107.8
26	New York Knicks	82	241.2	37.7	86.2	0.437	13.2	36.9	0.357	24.5	49.3	0.497	18	24.1	0.744	11.5	34.6	46.1	21.9	7	4.9	13.3	20.4	106.5
27	Portland Trail Blazers	82	240.6	38.5	87.1	0.442	12.7	36.8	0.346	25.8	50.3	0.513	16.4	21.6	0.76	10.4	32.5	42.9	22.9	8	4.5	14.5	21.1	106.2
28	Detroit Pistons	82	241.2	38.2	88.6	0.431	11.3	34.6	0.326	26.9	54	0.498	17.2	22	0.782	11	32	43	23.5	7.7	4.8	14.2	21.9	104.8
29	Orlando Magic	82	241.2	38.3	88.3	0.434	12.2	36.9	0.331	26.1	51.4	0.507	15.5	19.7	0.787	9.1	35.2	44.3	23.7	6.8	4.5	14.5	19.7	104.2
30	Oklahoma City Thunder	82	241.5	38.3	89.1	0.43	12.1	37.4	0.323	26.2	51.8	0.507	15	19.9	0.756	10.4	35.2	45.6	22.2	7.6	4.6	14	18.3	103.7
	League Average	82	241.4	40.6	88.1	0.461	12.4	35.2	0.354	28.2	52.9	0.533	16.9	21.9	0.775	10.3	34.1	44.5	24.6	7.6	4.7	13.8	19.6	110.6

Appendix B: Final Dataset Sample from the 2021 – 2022 Season

Team	SEASON	X3_PCT_T	X2_PCT_T	FT_PCT_T	ORB_T	DRB_T	BLK_T	TOV_T	PF_T	X3_PCT_O	X2_PCT_O	FT_PCT_O	ORB_O	DRB_O	BLK_O	TOV_O	PF_O	X3_PCT_ATT_T	PTS_DIFF	Playoffs	Champion
Atlanta Hawks*	2021_2022	0.374	0.531	0.812	10	33.9	4.2	11.9	18.7	0.364	0.541	0.792	10.2	33.6	4.5	12.8	20.3	0.3896	1.5	Yes	No
Boston Celtics*	2021_2022	0.356	0.547	0.816	10.5	35.5	5.8	13.6	18.5	0.339	0.497	0.783	10.5	33.3	4.6	13.8	19.4	0.4245	7.3	Yes	No
Brooklyn Nets*	2021_2022	0.361	0.538	0.805	10.3	34.1	5.5	14.1	20.4	0.345	0.523	0.788	11.3	32.8	4.9	13.3	19.7	0.3586	0.8	Yes	No
Charlotte Hornets	2021_2022	0.365	0.542	0.74	10.8	33.7	4.9	13.3	19.9	0.362	0.544	0.763	11.4	35.6	4.6	15	19.6	0.4179	0.4	No	No
Chicago Bulls*	2021_2022	0.369	0.535	0.813	8.7	33.7	4.1	12.8	18.8	0.366	0.536	0.795	9.3	33.9	5	13.1	18.2	0.3314	-0.4	Yes	No
Cleveland Cavaliers	2021_2022	0.355	0.541	0.76	10.2	34	4.2	14.4	17.5	0.353	0.514	0.774	10.5	32.4	4.6	13.4	20	0.3877	2.1	No	No
Dallas Mavericks*	2021_2022	0.35	0.548	0.771	9.3	33.8	4	12.5	19.7	0.34	0.527	0.767	9.5	34.1	3.7	13.1	20.1	0.4395	3.3	Yes	No
Denver Nuggets*	2021_2022	0.353	0.575	0.795	9.2	34.9	3.7	14.5	20	0.346	0.549	0.757	9.7	32.7	4.8	12.8	19.9	0.416	2.3	Yes	No
Detroit Pistons	2021_2022	0.326	0.498	0.782	11	32	4.8	14.2	21.9	0.362	0.54	0.777	10.3	35.9	5.2	14.6	19.8	0.3905	-7.7	No	No
Golden State Warriors*	2021_2022	0.364	0.557	0.769	9.8	35.7	4.5	14.9	21	0.339	0.509	0.759	9.7	33	3.9	14.5	18	0.456	5.5	Yes	Yes
Houston Rockets	2021_2022	0.349	0.543	0.713	9.6	32.4	4.7	16.5	20.6	0.353	0.57	0.782	11.2	34.5	5.8	14.1	22.1	0.4479	-8.5	No	No
Indiana Pacers	2021_2022	0.344	0.541	0.768	11.3	32.6	5.6	14.4	20.4	0.372	0.547	0.797	10.2	33	4.9	13.3	19.2	0.3955	-3.4	No	No
Los Angeles Clippers	2021_2022	0.374	0.512	0.793	9.1	34.9	5	13.7	18.6	0.345	0.513	0.775	12	35.1	4.1	13.6	18.5	0.3913	0	No	No
Los Angeles Lakers	2021_2022	0.347	0.546	0.732	9.5	34.5	5.2	14.5	20.2	0.352	0.548	0.747	11	35.6	4.1	14	20.1	0.3885	-3	No	No
Memphis Grizzlies*	2021_2022	0.353	0.519	0.734	14.1	35	6.5	13.2	19.8	0.349	0.522	0.771	10	33	6	15.1	19.8	0.3464	5.7	Yes	No
Miami Heat*	2021_2022	0.379	0.531	0.808	9.8	33.9	3.2	14.6	20.5	0.339	0.537	0.779	9.5	32.1	4	15	20.6	0.4222	4.4	Yes	No
Milwaukee Bucks*	2021_2022	0.366	0.544	0.776	10.2	36.5	4	13.4	18.2	0.356	0.537	0.751	9.9	34.2	4.4	13	19.7	0.4295	3.4	Yes	No
Minnesota Timberwolves*	2021_2022	0.358	0.54	0.778	11.2	32.9	5.6	14.3	21.8	0.35	0.543	0.78	11	34.9	4.8	16.3	20.9	0.4538	2.6	Yes	No
New Orleans Pelicans*	2021_2022	0.332	0.528	0.789	12	33.2	4	14.1	19.7	0.365	0.546	0.786	9.3	32.5	4.8	14.2	20.5	0.3648	-1	Yes	No
New York Knicks	2021_2022	0.357	0.497	0.744	11.5	34.6	4.9	13.3	20.4	0.342	0.527	0.764	9.3	34.3	4.6	12.7	20.4	0.4281	-0.1	No	No
Oklahoma City Thunder	2021_2022	0.323	0.507	0.756	10.4	35.2	4.6	14	18.3	0.353	0.536	0.771	11	37.3	6	13.2	17.8	0.4198	-8.1	No	No
Orlando Magic	2021_2022	0.331	0.507	0.787	9.1	35.2	4.5	14.5	19.7	0.363	0.524	0.772	10.4	36.8	5.2	13.2	18.3	0.4179	-8	No	No
Philadelphia 76ers*	2021_2022	0.364	0.527	0.821	8.5	33.8	5.3	12.5	19.4	0.346	0.527	0.769	10.2	34	4.6	13.3	19.4	0.3763	2.6	Yes	No
Phoenix Suns*	2021_2022	0.364	0.551	0.797	9.8	35.5	4.4	12.9	19.9	0.34	0.51	0.761	10.5	33.9	4	14.7	18.9	0.3541	7.5	Yes	No
Portland Trail Blazers	2021_2022	0.346	0.513	0.76	10.4	32.5	4.5	14.5	21.1	0.371	0.561	0.777	9.8	35	5	14	19.4	0.4225	-8.9	No	No
Sacramento Kings	2021_2022	0.344	0.53	0.768	9.6	33.4	4.5	14.1	18.9	0.372	0.547	0.776	10.6	35.3	4.8	13.5	20.3	0.3768	-5.5	No	No
San Antonio Spurs	2021_2022	0.352	0.527	0.754	11	34.3	4.9	12.7	18.1	0.36	0.528	0.767	11.4	35.4	4.9	13.8	18.9	0.3452	0.2	No	No
Toronto Raptors*	2021_2022	0.349	0.503	0.759	13.4	32	4.6	12.5	19.6	0.354	0.538	0.789	10.3	33.7	5.1	15.8	19.1	0.3746	2.3	Yes	No
Utah Jazz*	2021_2022	0.36	0.568	0.767	10.8	35.6	4.9	14	18.9	0.35	0.518	0.758	9.9	31.7	4.3	12	20.3	0.4675	6	Yes	No
Washington Wizards	2021_2022	0.342	0.543	0.783	9	34.1	5	13.1	18.8	0.361	0.522	0.807	10.2	34	4.2	11.9	19.9	0.3558	-3.4	No	No

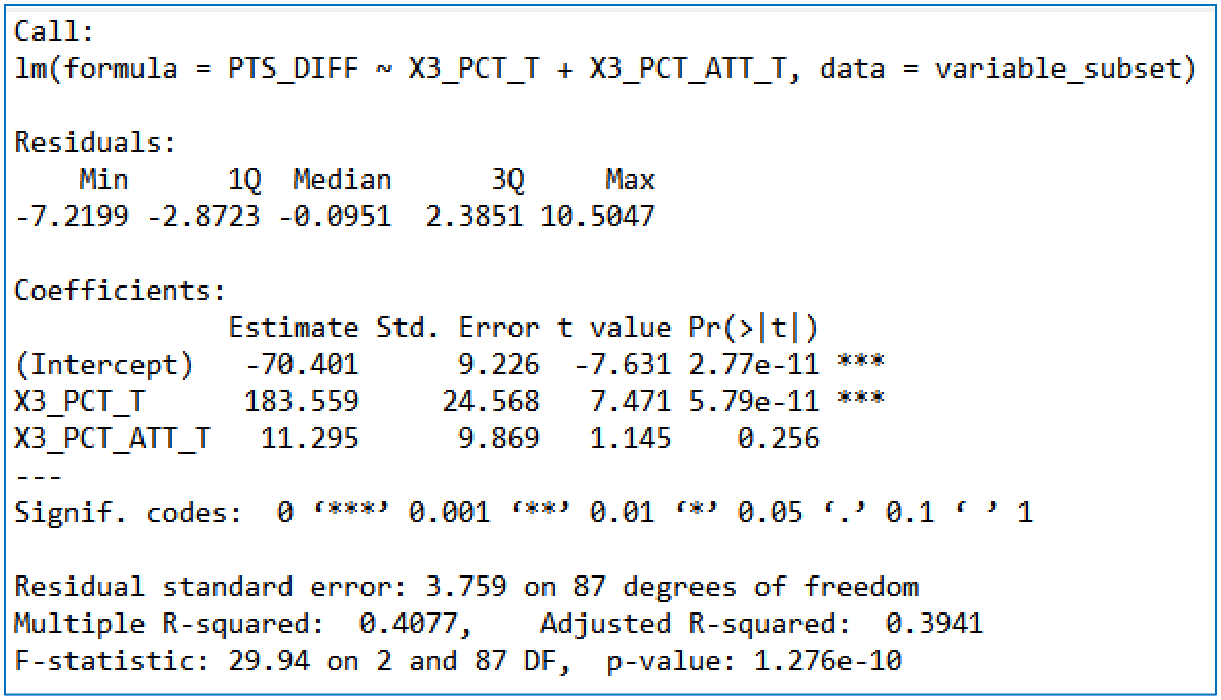
Appendix C: R Code for Creating the Final Dataset

```
### Part 1: Stack the 10 Team .csv files.

# Set the working directory
setwd('C:/fun/Australia/2022 Semester 2/DATA7001/Group Project/NBA Data/csv')

# Read in the Team .csv dataset from 10 years ago, and assign it to the first_file dataframe.
first_file <- read.csv('nba_2012_2013_team.csv')

# Add the source dataset name to the dataset column in first_file.
first_file$dataset <- 'nba_2012_2013_team.csv'

# List the last 9 seasons of Team datasets in the file_list array.
file_list <- c('nba_2013_2014_team.csv','nba_2014_2015_team.csv',
            'nba_2015_2016_team.csv','nba_2016_2017_team.csv',
            'nba_2017_2018_team.csv','nba_2018_2019_team.csv',
            'nba_2019_2020_team.csv','nba_2020_2021_team.csv',
            'nba_2021_2022_team.csv')

# Initialize the cumulative_seasons dataframe with first_file.
cumulative_seasons <- first_file

# Append the last nine seasons to the cumulative_seasons dataset.
for (f in file_list)
{fn <- read.csv(f)
fn$dataset <- f
cumulative_seasons <- rbind(cumulative_seasons,fn)}

# Validate the cumulative_seasons dataset.  It has 310 rows, or 300 for individual team/season
# combinations, and 10 rows with the league averages for a season.
table(cumulative_seasons$dataset)
dim(cumulative_seasons)

# Output the 300 team/season and 10 league season average rows into separate dataframe.
cumulative_seasons_league <- cumulative_seasons[cumulative_seasons$"Team" == "League Average",]
cumulative_seasons_team <- cumulative_seasons[cumulative_seasons$"Team" != "League Average",]

# Output the two separate dataframes into .csv files.
write.csv(cumulative_seasons_league,'all_seasons_league.csv')
write.csv(cumulative_seasons_team,'all_seasons_team.csv')


### Part 2: Stack the 10 Opponent .csv files.

# Read in the Opponent .csv dataset from 10 years ago, and assign it to the first_file dataframe.
```

```
first_file <- read.csv('nba_2012_2013_opponent.csv')
# Add the source dataset name to the dataset column in first_file.
first_file$dataset <- 'nba_2012_2013_opponent.csv'

# List the last 9 seasons of Opponent datasets in the file_list array.
file_list <- c('nba_2013_2014_opponent.csv','nba_2014_2015_opponent.csv',
               'nba_2015_2016_opponent.csv','nba_2016_2017_opponent.csv',
               'nba_2017_2018_opponent.csv','nba_2018_2019_opponent.csv',
               'nba_2019_2020_opponent.csv','nba_2020_2021_opponent.csv',
               'nba_2021_2022_opponent.csv')

# Initialize the cumulative_seasons dataframe with first_file.
cumulative_seasons <- first_file

# Append the last nine seasons to the cumulative_seasons dataset.
for (f in file_list)
{fn <- read.csv(f)
fn$dataset <- f
cumulative_seasons <- rbind(cumulative_seasons,fn)}

# Validate the cumulative_seasons dataset.  It has 310 rows, or 300 for individual opponent/season
# combinations, and 10 rows with the league averages for a season.
table(cumulative_seasons$dataset)
dim(cumulative_seasons)

# Output only the 300 opponent/season rows into a dataframe.
cumulative_seasons_opponent <- cumulative_seasons[cumulative_seasons$"Opponent" != "League
Average",]

# Output the team/season dataframe into a .csv file.
write.csv(cumulative_seasons_opponent,'all_seasons_opponent.csv')


### Part 3: Merge the Team and Opponent Datasets

# Read in the file containing 10 seasons of Team data.
team_pre <- read.csv('all_seasons_team.csv')
# Drop the Rk variable.
team <- team_pre[ ,2:26]
# Use the substring function to create a Season variable from the filename.
team$Season <- substring(team$dataset,5,13)
# Drop the filename variable.
team <- team[,-25]
# Add _T to the name of each variable except for Team and SEASON. Team and SEASON are
# used to merge the Team and Opponent datasets.
names(team) <-
c("Team","G_T","MP_T","FG_T","FGA_T","FG_PCT_T","X3P_T","X3PA_T","X3_PCT_T","X2P_T",
           "X2PA_T","X2_PCT_T","FT_T","FTA_T","FT_PCT_T","ORB_T","DRB_T","TRB_T","AST_T",
            "STL_T","BLK_T","TOV_T","PF_T","PTS_T","SEASON")

# Read in the file containing 10 seasons of Opponent data.
opponent_pre <- read.csv('all_seasons_opponent.csv')
# Drop the Rk variable.
opponent <- opponent_pre[ ,2:26]
# Use the substring function to create a Season variable from the filename.
opponent$Season <- substring(opponent$dataset,5,13)
# Drop the filename variable.
opponent <- opponent[,-25]
# Add _O to the name of each variable except for Team and SEASON. Team and SEASON are
# used to merge the Team and Opponent datasets.
names(opponent) <-
c("Team","G_O","MP_O","FG_O","FGA_O","FG_PCT_O","X3P_O","X3PA_O","X3_PCT_O","X2P_O",
              "X2PA_O","X2_PCT_O","FT_O","FTA_O","FT_PCT_O","ORB_O","DRB_O","TRB_O","AST_O",
              "STL_O","BLK_O","TOV_O","PF_O","PTS_O","SEASON")

# Merge the Team and Opponent data by Team and SEASON.
team_opponent_pre <- merge(team,opponent,by = intersect(names(team),names(opponent)))
# Sort the merged dataset by SEASON and Team.
team_opponent <-
team_opponent_pre[order(team_opponent_pre$SEASON,team_opponent_pre$Team),]

# Check for redundant columns
cor(team_opponent$G_T,team_opponent$G_O)
cor(team_opponent$MP_T,team_opponent$MP_O)

# Drop two redundant columns
team_opponent <- within(team_opponent, rm(G_O,MP_O))

# Keep only the last three seasons of data.
team_opponent_last_three <- team_opponent[team_opponent[,2] == '2019_2020' |
                              team_opponent[,2] == '2020_2021' |
                              team_opponent[,2] == '2021_2022',]

# Calculate a Team's 3-Point Attempt %.
team_opponent_last_three$X3_PCT_ATT_T <- round(team_opponent_last_three$X3PA_T /
team_opponent_last_three$FGA_T,4)

# Calculate the Point Differential between a Team and its Opponents.
team_opponent_last_three$PTS_DIFF <- team_opponent$PTS_T - team_opponent$PTS_O

# Calculate a Yes/No indicator of making the playoffs for the Team/Season combination.
team_opponent_last_three$Playoffs <-
ifelse(substring(team_opponent$Team,nchar(team_opponent$Team),nchar(team_opponent$Team))=='
*','Yes','No')

# Calculate a Yes/No indicator for the team that win the championship during the season.
```

```
team_opponent_last_three$Champion <- ifelse((team_opponent$Team == 'Los Angeles Lakers*' &
team_opponent$SEASON == '2019_2020') |
                    (team_opponent$Team == 'Milwaukee Bucks*' & team_opponent$SEASON ==
'2020_2021') |
                    (team_opponent$Team == 'Golden State Warriors*' & team_opponent$SEASON ==
'2021_2022')
                    ,'Yes','No')

# Remove several variables that will not be used in analysis.
team_opponent_last_three_reduced <-
team_opponent_last_three[,c(1:2,10,13,16:18,22:24,31,34,37:39,43:45,47:50)]

# Output the merged dataset with four new variables to the last_three_seasons_team_opponent.csv
file.
write.csv(team_opponent_last_three_reduced,"last_three_seasons_team_opponent.csv")
```

Appendix D: Summary of the 2 Feature Linear Regression Model



```
Call:
lm(formula = PTS_DIFF ~ X3_PCT_T + X3_PCT_ATT_T, data = variable_subset)

Residuals:
    Min      1Q  Median      3Q     Max 
-7.2199 -2.8723 -0.0951  2.3851 10.5047 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)   -70.401      9.226  -7.631 2.77e-11 ***
X3_PCT_T      183.559     24.568   7.471 5.79e-11 ***
X3_PCT_ATT_T   11.295      9.869   1.145    0.256    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.759 on 87 degrees of freedom
Multiple R-squared:  0.4077,	Adjusted R-squared:  0.3941 
F-statistic: 29.94 on 2 and 87 DF,  p-value: 1.276e-10
```

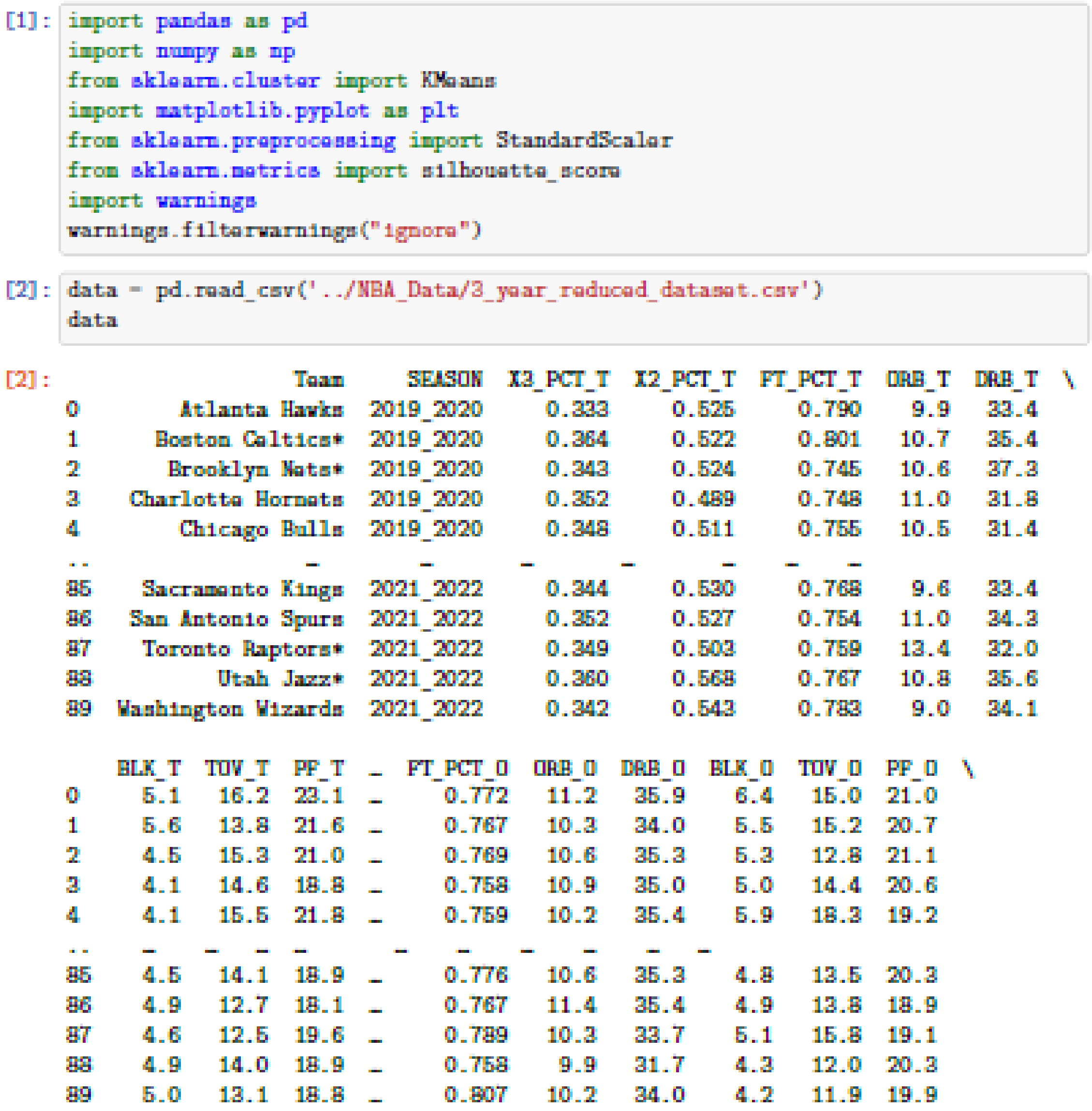
Appendix E: R Code for the 2 Feature Linear Regression Model

```
library(corrplot)
setwd('C:/fun/Australia/2022 Semester 2/DATA7001/Group Project/NBA Data/csv')

# Read in the 10-Year dataset.
teams_opponents <- read.csv("all_seasons_team_opponent.csv")

# Obtain only the last three seasons.
last_three_seasons <- teams_opponents[teams_opponents$SEASON == '2019_2020' |
                                      teams_opponents$SEASON == '2020_2021' |
                                      teams_opponents$SEASON == '2021_2022',]

# Variables for the question of whether 3-Point accuracy contributes to the optimum mix of
# 2-Point v. 3-Point shot attempts.
variables_to_keep <- c('PTS_DIFF','X3_PCT_ATT_T','X3_PCT_T','Playoffs','Champion')
variable_subset <- last_three_seasons[,(names(last_three_seasons) %in% (variables_to_keep))]

# Fit a linear regression model where point differential is predicted by 3-Point accuracy
# and the % of shots that are 3-Point attempts.
lm_model <- lm(PTS_DIFF ~ X3_PCT_T + X3_PCT_ATT_T, data = variable_subset)
summary(lm_model)

# Import the ggplot2 library.
library(ggplot2)

# The graph shows that 3-Point accuracy shows that point differential doesn't vary much
# by the mix of 3-Point v. 2-Point attempts.
# The last three NBA Champions are spread throughout the scatterplot.
ggplot(variable_subset,aes(x = X3_PCT_T, y = X3_PCT_ATT_T, color = Champion, size =
PTS_DIFF)) +
  geom_point(alpha = 1) + theme_classic() + xlab("3-Point Accuracy") + ylab("3-Point Attempt
%")

# Teams have success with a wide mix of 3-Point v. 2-Point shot attempts.
ggplot(variable_subset,aes(x = X3_PCT_T, y = X3_PCT_ATT_T, color = Playoffs, size = PTS_DIFF))
+
  geom_point(alpha = 1) + theme_classic() + xlab("3-Point Accuracy") + ylab("3-Point Attempt
%")
```

Appendix F: R Code for the 2 Feature Linear Model Diagnostics

```
setwd('C:/fun/Australia/2022 Semester 2/DATA7001/Group Project/NBA Data/csv')

# Read in the dataset with the last three seasons of NBA data.
last_three_seasons <- read.csv("3_year_reduced_dataset.csv")

# Variables for the question of whether 3-Point accuracy contributes to the optimum mix of
variables_to_keep <- c('PTS_DIFF','X3_PCT_ATT_T','X3_PCT_T','Playoffs','Champion')
variable_subset <- last_three_seasons[,(names(last_three_seasons) %in% (variables_to_keep))]
print(variable_subset)

# Fit a linear regression model where point differential is predicted by 3-Point accuracy
# and the % of shots that are 3-Point attempts.
lm_model <- lm(PTS_DIFF ~ X3_PCT_T + X3_PCT_ATT_T, data = variable_subset)
summary(lm_model)
pred <- predicted(lm_model)

# Obtain the Variance Inflaction Factors from the 2 feature regression model.
vif(lm_model)

###To analyzing residuals
## -The mean of zeros is zero (and the sum of the errors is zero)
## -The distribution of the erros are normal
## -All of the errors are independent
## -Variance of errors is constant (Homoscedastic)

layout(matrix(c(1,1,2,3),2,2,byrow=T))
#Linear Regression x Residuals Plot
windows(10,5)
plot(lm_model$resid~predict(lm_model), main="Linear Regression x Residuals Plot",
    xlab="Predicted Point Differential",ylab="Residuals")
abline(h=0,lty=3)

#Histogram of Residuals
hist(lm_model$resid, main="Histogram of Residuals",     ylab="Residuals")
#Q-Q Plot -> close to the line is better
qqnorm(lm_model$resid)
qqline(lm_model$resid)

###The Jarque-Bera test (fBasics library)
jarqueberaTest(lm_model$resid)
```

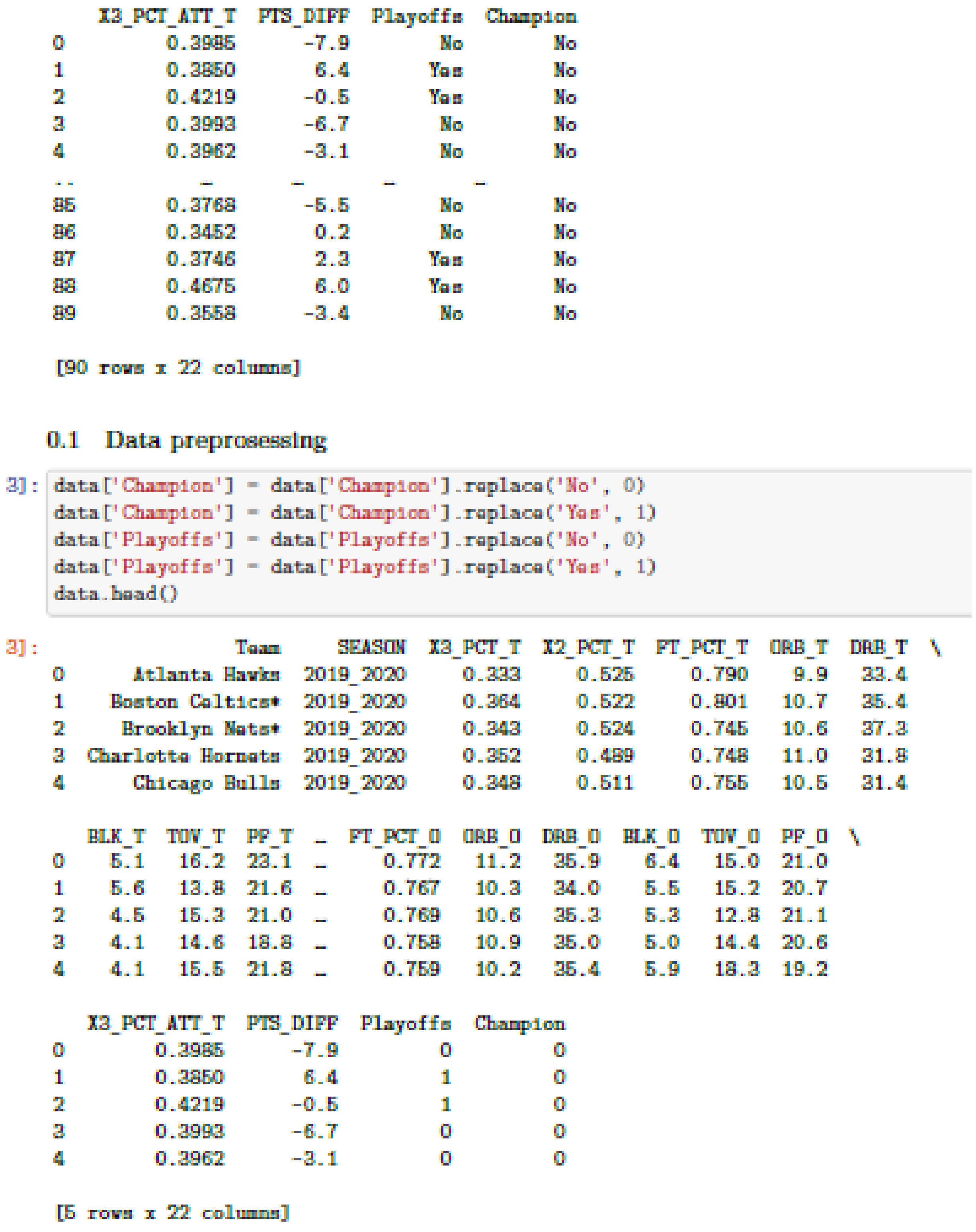
Appendix G: Python Code for the K-Means Clustering



```
[1]: import pandas as pd
     import numpy as np
     from sklearn.cluster import KMeans
     import matplotlib.pyplot as plt
     from sklearn.preprocessing import StandardScaler
     from sklearn.metrics import silhouette_score
     import warnings
     warnings.filterwarnings("ignore")
```

```
[2]: data = pd.read_csv('../NBA_Data/3_year_reduced_dataset.csv')
     data
```

```
[2]:                  Team     SEASON  X3_PCT_T  X2_PCT_T  FT_PCT_T  ORB_T  DRB_T  \
     0       Atlanta Hawks  2019_2020     0.333     0.525     0.790    9.9   33.4
     1     Boston Celtics*  2019_2020     0.364     0.522     0.801   10.7   35.4
     2       Brooklyn Nets*  2019_2020     0.343     0.524     0.745   10.6   37.3
     3   Charlotte Hornets  2019_2020     0.352     0.489     0.748   11.0   31.8
     4       Chicago Bulls  2019_2020     0.348     0.511     0.755   10.5   31.4
     ..                ...        ...       ...       ...       ...    ...    ...
     85    Sacramento Kings  2021_2022     0.344     0.530     0.768    9.6   33.4
     86   San Antonio Spurs  2021_2022     0.352     0.527     0.754   11.0   34.3
     87    Toronto Raptors*  2021_2022     0.349     0.503     0.759   13.4   32.0
     88          Utah Jazz*  2021_2022     0.360     0.568     0.767   10.8   35.6
     89  Washington Wizards  2021_2022     0.342     0.543     0.783    9.0   34.1

         BLK_T  TOV_T  PF_T  ...  FT_PCT_O  ORB_O  DRB_O  BLK_O  TOV_O  PF_O  \
     0     5.1   16.2  23.1  ...     0.772   11.2   35.9    6.4   15.0  21.0
     1     5.6   13.8  21.6  ...     0.767   10.3   34.0    5.5   15.2  20.7
     2     4.5   15.3  21.0  ...     0.769   10.6   35.3    5.3   12.8  21.1
     3     4.1   14.6  18.8  ...     0.758   10.9   35.0    5.0   14.4  20.6
     4     4.1   15.5  21.8  ...     0.759   10.2   35.4    5.9   18.3  19.2
     ..    ...    ...   ...  ...       ...    ...    ...    ...    ...   ...
     85    4.5   14.1  18.9  ...     0.776   10.6   35.3    4.8   13.5  20.3
     86    4.9   12.7  18.1  ...     0.767   11.4   35.4    4.9   13.8  18.9
     87    4.6   12.5  19.6  ...     0.789   10.3   33.7    5.1   15.8  19.1
     88    4.9   14.0  18.9  ...     0.758    9.9   31.7    4.3   12.0  20.3
     89    5.0   13.1  18.8  ...     0.807   10.2   34.0    4.2   11.9  19.9
```

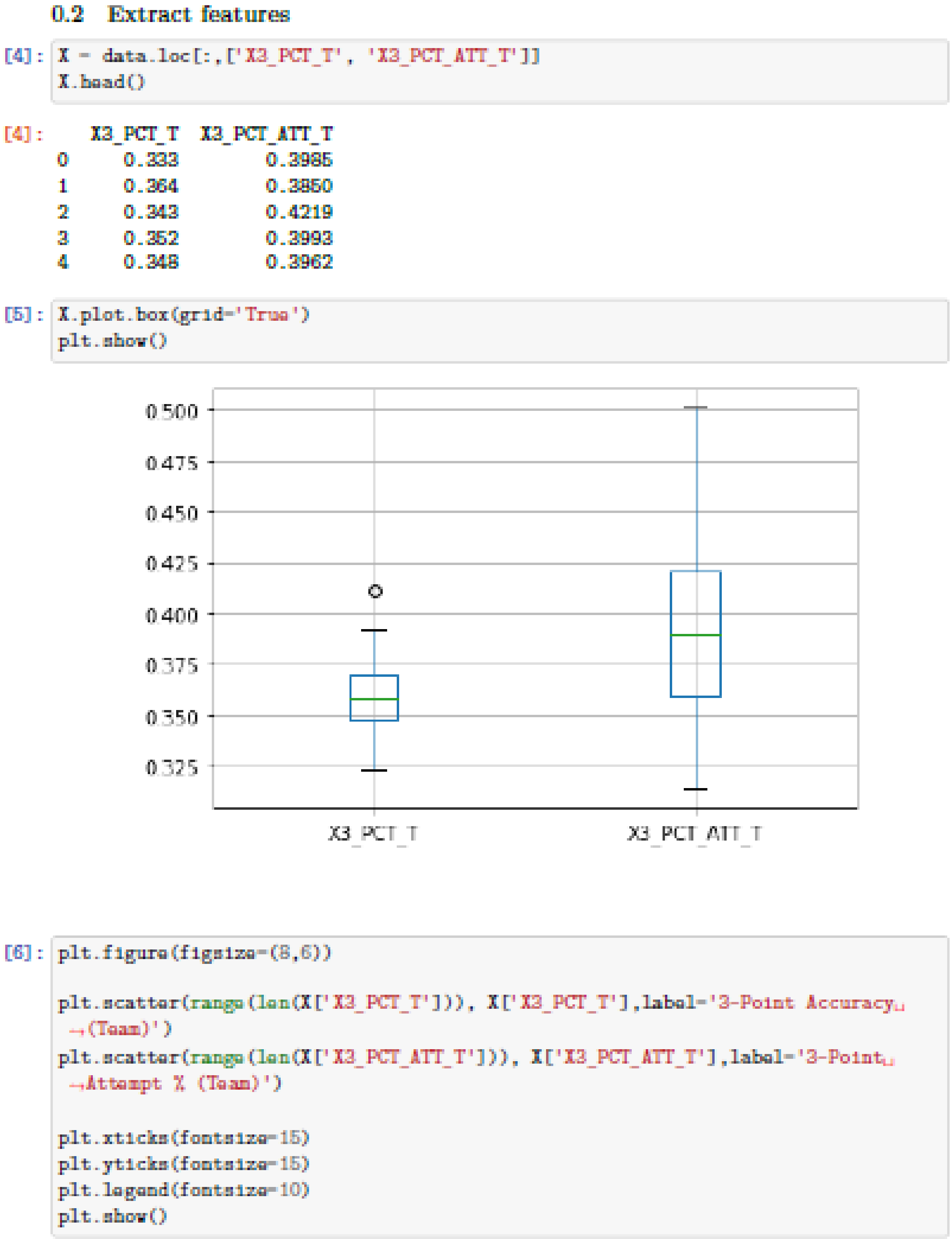


```
    X3_PCT_ATT_T  PTS_DIFF  Playoffs  Champion
0         0.3985      -7.9        No        No
1         0.3850       6.4       Yes        No
2         0.4219      -0.5       Yes        No
3         0.3993      -6.7        No        No
4         0.3962      -3.1        No        No
..           ...       ...       ...       ...
85        0.3768      -5.5        No        No
86        0.3452       0.2        No        No
87        0.3746       2.3       Yes        No
88        0.4675       6.0       Yes        No
89        0.3558      -3.4        No        No

[90 rows x 22 columns]
```

0.1 Data preprosessing

```
3]: data['Champion'] = data['Champion'].replace('No', 0)
    data['Champion'] = data['Champion'].replace('Yes', 1)
    data['Playoffs'] = data['Playoffs'].replace('No', 0)
    data['Playoffs'] = data['Playoffs'].replace('Yes', 1)
    data.head()
```

```
3]:                 Team     SEASON  X3_PCT_T  X2_PCT_T  FT_PCT_T  ORB_T  DRB_T  \
    0      Atlanta Hawks  2019_2020     0.333     0.525     0.790    9.9   33.4
    1    Boston Celtics*  2019_2020     0.364     0.522     0.801   10.7   35.4
    2     Brooklyn Nets*  2019_2020     0.343     0.524     0.745   10.6   37.3
    3  Charlotte Hornets  2019_2020     0.352     0.489     0.748   11.0   31.8
    4      Chicago Bulls  2019_2020     0.348     0.511     0.755   10.5   31.4

       BLK_T  TOV_T  PF_T  ...  FT_PCT_O  ORB_O  DRB_O  BLK_O  TOV_O  PF_O  \
    0    5.1   16.2  23.1  ...     0.772   11.2   35.9    6.4   15.0  21.0
    1    5.6   13.8  21.6  ...     0.767   10.3   34.0    5.5   15.2  20.7
    2    4.5   15.3  21.0  ...     0.769   10.6   35.3    5.3   12.8  21.1
    3    4.1   14.6  18.8  ...     0.758   10.9   35.0    5.0   14.4  20.6
    4    4.1   15.5  21.8  ...     0.759   10.2   35.4    5.9   18.3  19.2

       X3_PCT_ATT_T  PTS_DIFF  Playoffs  Champion
    0        0.3985      -7.9         0         0
    1        0.3850       6.4         1         0
    2        0.4219      -0.5         1         0
    3        0.3993      -6.7         0         0
    4        0.3962      -3.1         0         0

    [5 rows x 22 columns]
```



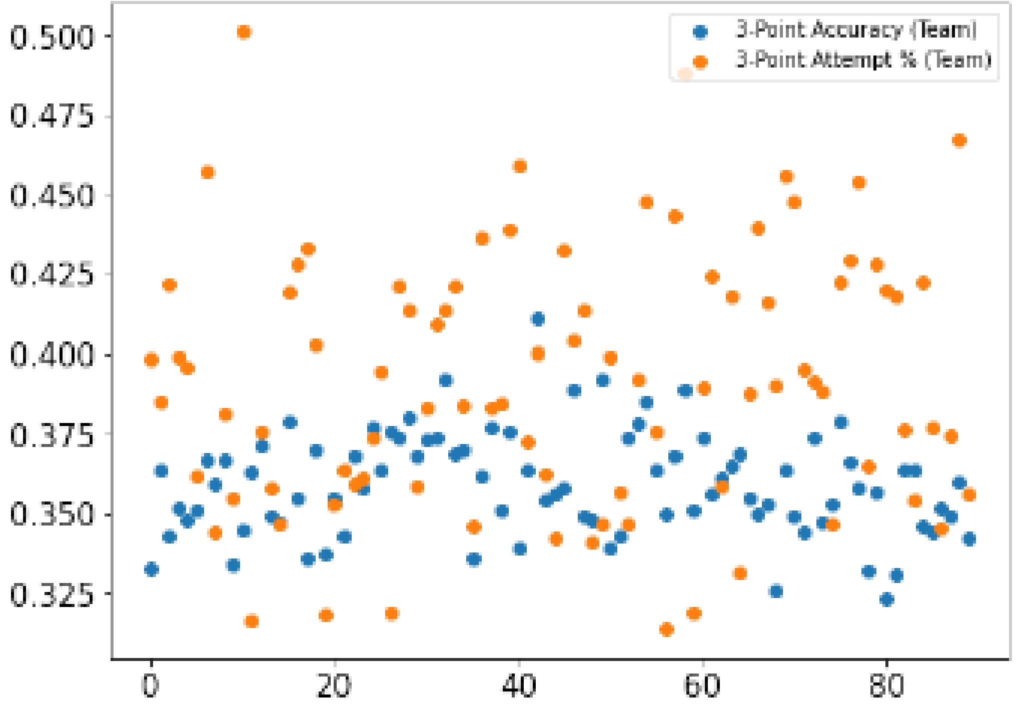
0.2 Extract features

```
[4]: X = data.loc[:,['X3_PCT_T', 'X3_PCT_ATT_T']]
     X.head()
```

```
[4]:    X3_PCT_T  X3_PCT_ATT_T
     0     0.333        0.3985
     1     0.364        0.3850
     2     0.343        0.4219
     3     0.352        0.3993
     4     0.348        0.3962
```

```
[5]: X.plot.box(grid='True')
     plt.show()
```

```
[6]: plt.figure(figsize=(8,6))

     plt.scatter(range(len(X['X3_PCT_T'])), X['X3_PCT_T'],label='3-Point Accuracy␣
      ↪(Team)')
     plt.scatter(range(len(X['X3_PCT_ATT_T'])), X['X3_PCT_ATT_T'],label='3-Point␣
      ↪Attempt % (Team)')

     plt.xticks(fontsize=15)
     plt.yticks(fontsize=15)
     plt.legend(fontsize=10)
     plt.show()
```

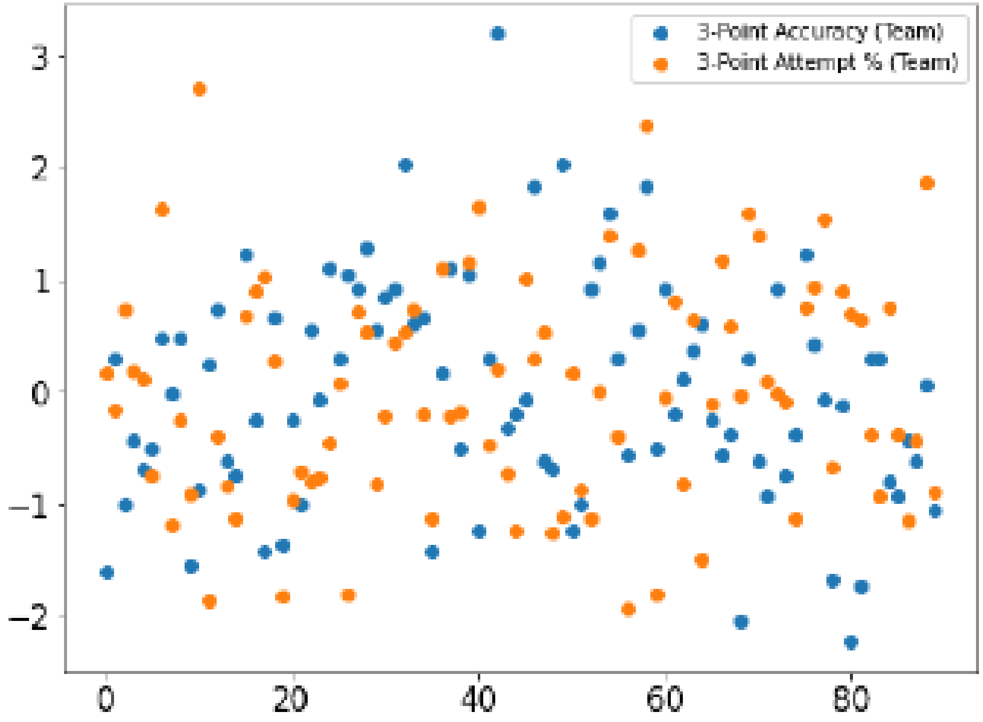


0.3 Standardization

```
[7]: X_scaled = StandardScaler().fit_transform(X)
     pd.DataFrame(StandardScaler().fit_transform(X), columns =␣
      ↪['X3_PCT_T','X3_PCT_ATT_T']).head()
```

```
[7]:    X3_PCT_T  X3_PCT_ATT_T
     0 -1.609146      0.165376
     1  0.300474     -0.168675
     2 -0.993140      0.744298
     3 -0.438734      0.185171
     4 -0.685136      0.108463
```

```
[8]: plt.figure(figsize=(8,6))

     plt.scatter(range(len(X_scaled[:,0])), X_scaled[:,0],label='3-Point Accuracy␣
      ↪(Team)')
     plt.scatter(range(len(X_scaled[:,1])), X_scaled[:,1],label='3-Point Attempt %␣
      ↪(Team)')

     plt.xticks(fontsize=15)
     plt.yticks(fontsize=15)
plt.legend(fontsize=10)
plt.show()
```

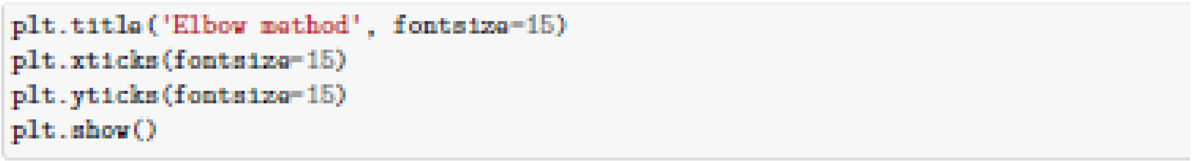


0.4 Determine the optimal number of clusters

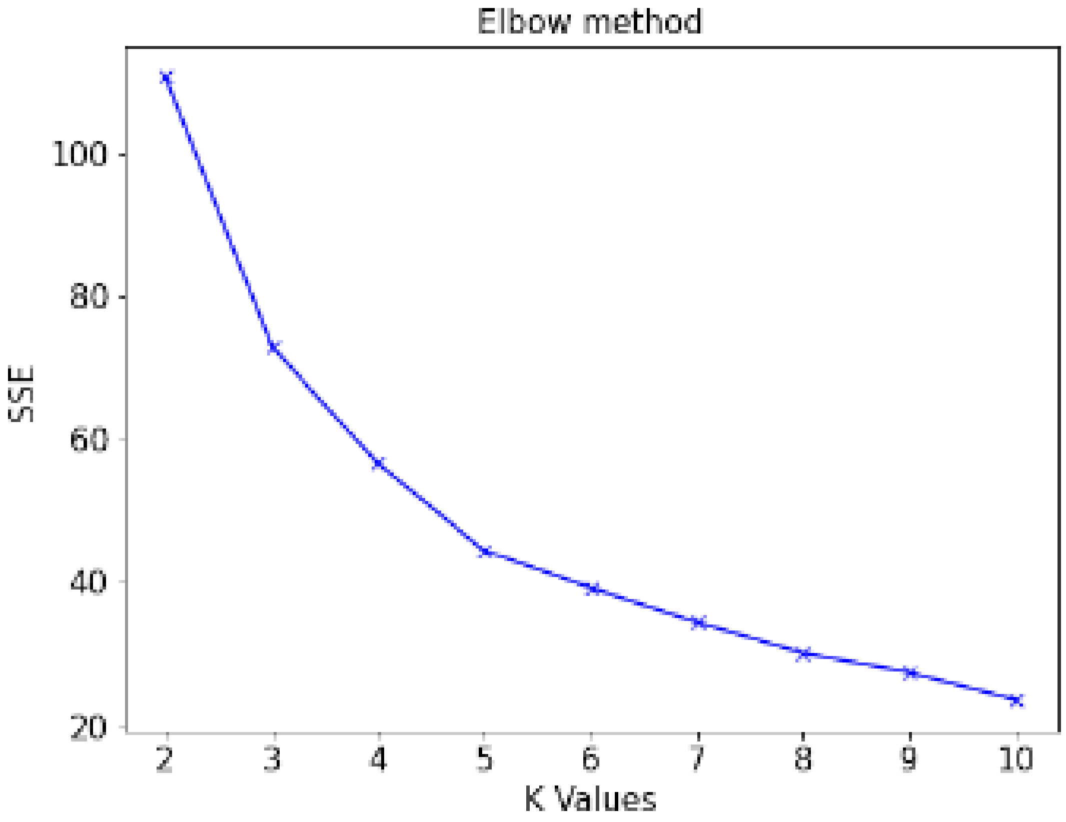
```
[353]: SSE = []
       silhouette_scores = []
       for k in range(2,11):
           kmeans = KMeans(n_clusters=k)
           kmeans.fit(X_scaled)
           SSE.append(kmeans.inertia_)
           silhouette_scores.append(silhouette_score(X_scaled, kmeans.labels_))
```

0.5 Elbow Method

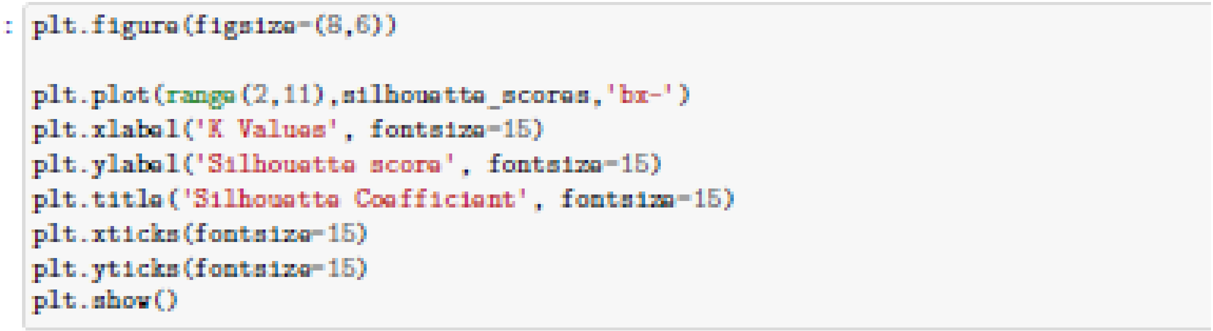
```
[403]: plt.figure(figsize=(8,6))

       plt.plot(range(2,11),SSE,'bx-')
       plt.xlabel('K Values', fontsize=15)
       plt.ylabel('SSE', fontsize=15) # sum of square error
```



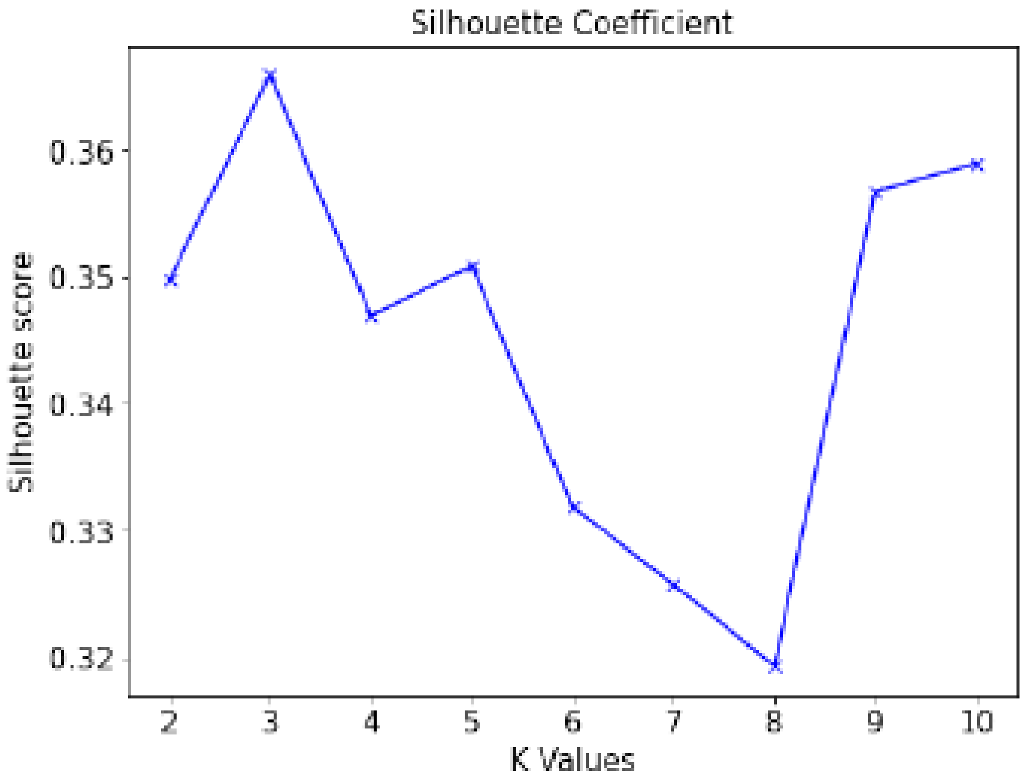
```
plt.title('Elbow method', fontsize=15)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
```

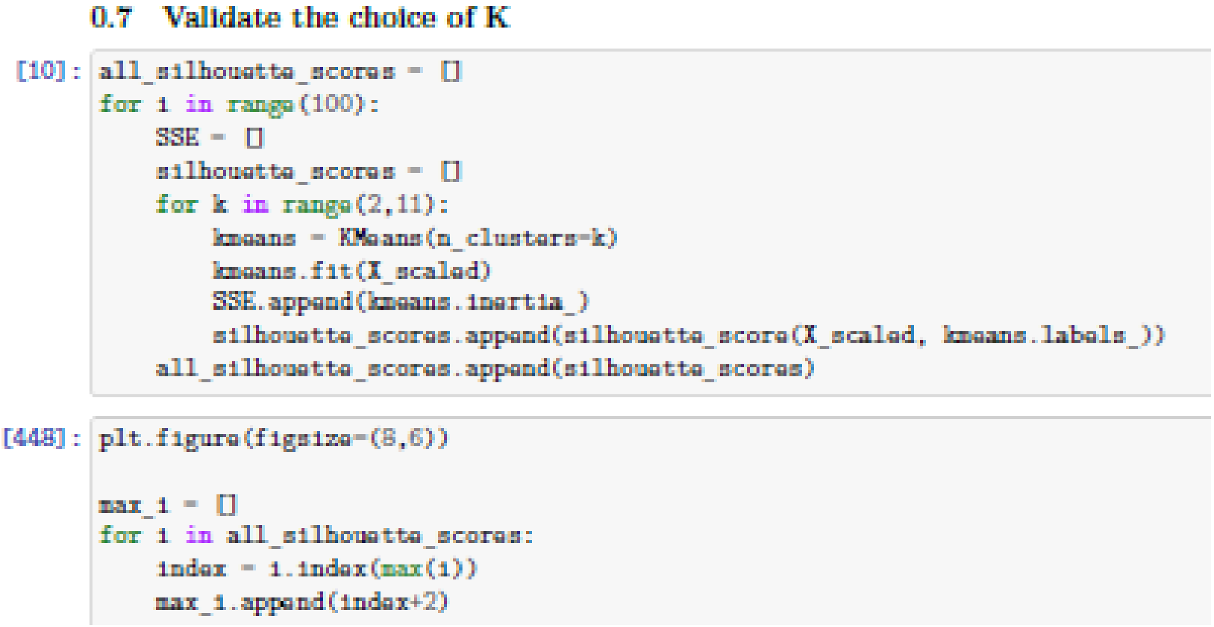


0.6 Average Silhouette Method



```
: plt.figure(figsize=(8,6))

  plt.plot(range(2,11),silhouette_scores,'bx-')
  plt.xlabel('K Values', fontsize=15)
  plt.ylabel('Silhouette score', fontsize=15)
  plt.title('Silhouette Coefficient', fontsize=15)
  plt.xticks(fontsize=15)
  plt.yticks(fontsize=15)
  plt.show()
```

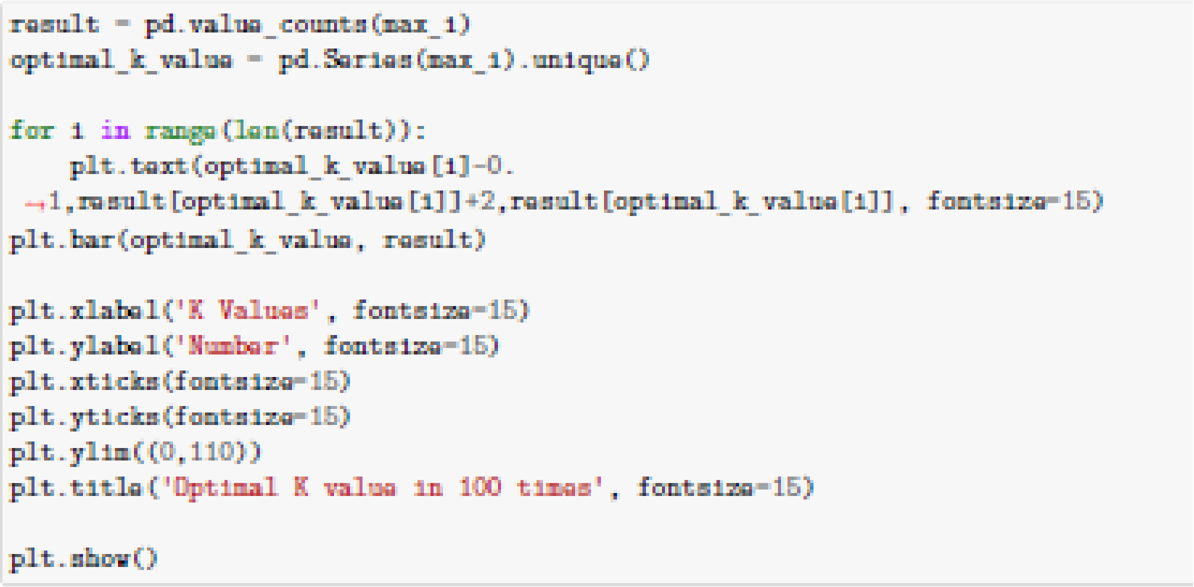




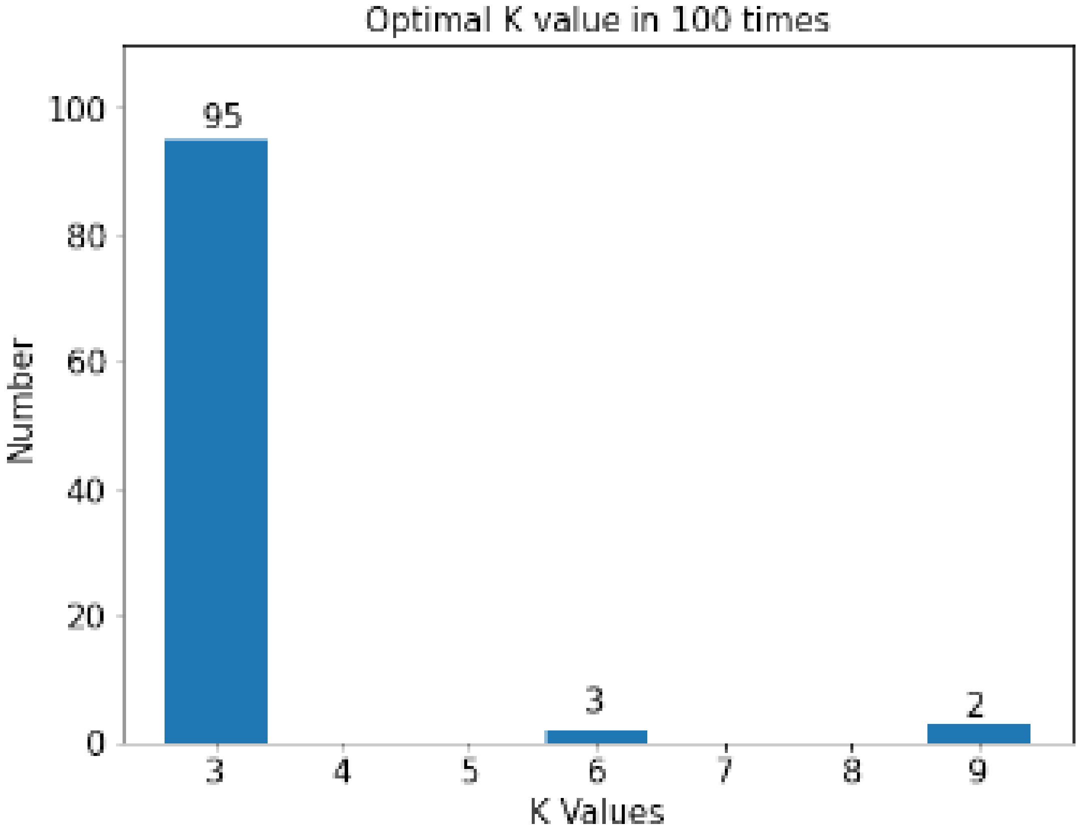
0.7 Validate the choice of K

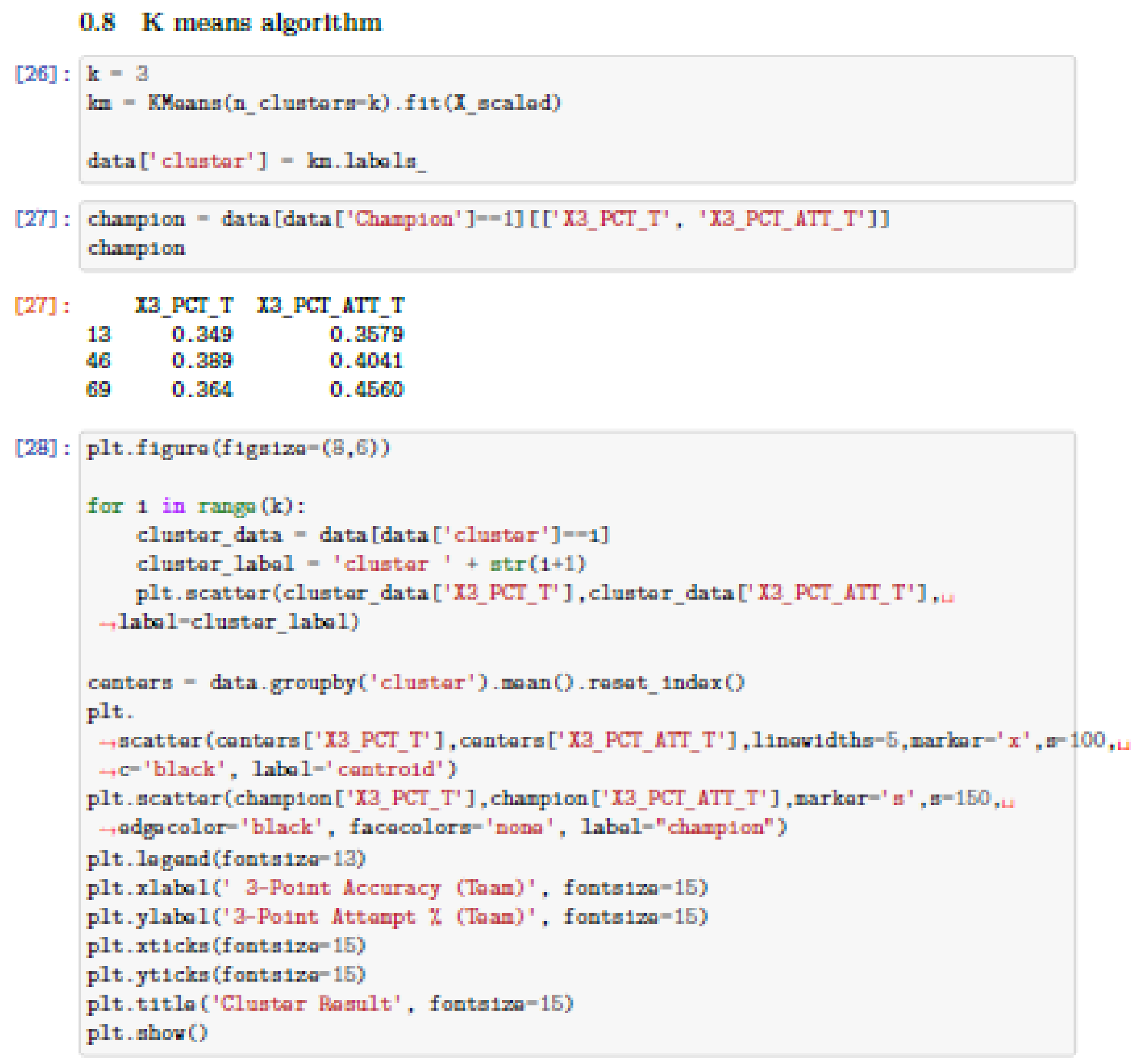
```
[10]: all_silhouette_scores = []
      for i in range(100):
          SSE = []
          silhouette_scores = []
          for k in range(2,11):
              kmeans = KMeans(n_clusters=k)
              kmeans.fit(X_scaled)
              SSE.append(kmeans.inertia_)
              silhouette_scores.append(silhouette_score(X_scaled, kmeans.labels_))
          all_silhouette_scores.append(silhouette_scores)
```

```
[448]: plt.figure(figsize=(8,6))

       max_i = []
       for i in all_silhouette_scores:
           index = i.index(max(i))
           max_i.append(index+2)
```



```
result = pd.value_counts(max_i)
optimal_k_value = pd.Series(max_i).unique()

for i in range(len(result)):
    plt.text(optimal_k_value[i]-0.
 →1,result[optimal_k_value[i]]+2,result[optimal_k_value[i]], fontsize=15)
plt.bar(optimal_k_value, result)

plt.xlabel('K Values', fontsize=15)
plt.ylabel('Number', fontsize=15)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.ylim((0,110))
plt.title('Optimal K value in 100 times', fontsize=15)

plt.show()
```





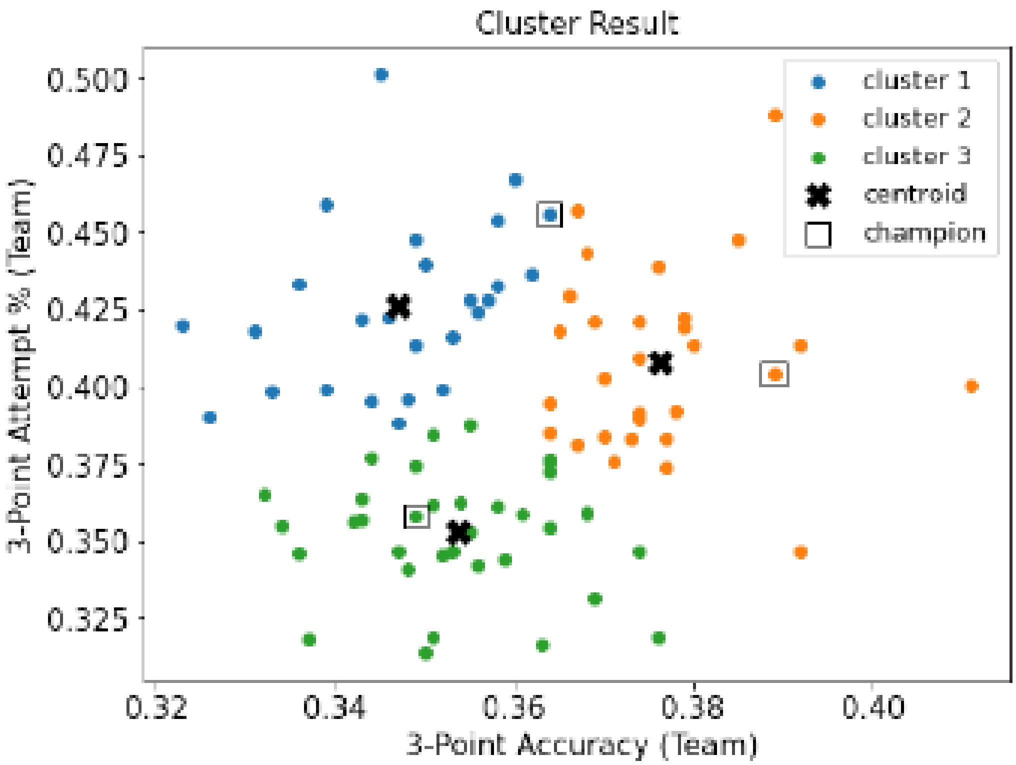
0.8 K means algorithm

[26]:
```
k = 3
km = KMeans(n_clusters=k).fit(X_scaled)

data['cluster'] = km.labels_
```

[27]:
```
champion = data[data['Champion']==1][['X3_PCT_T', 'X3_PCT_ATT_T']]
champion
```

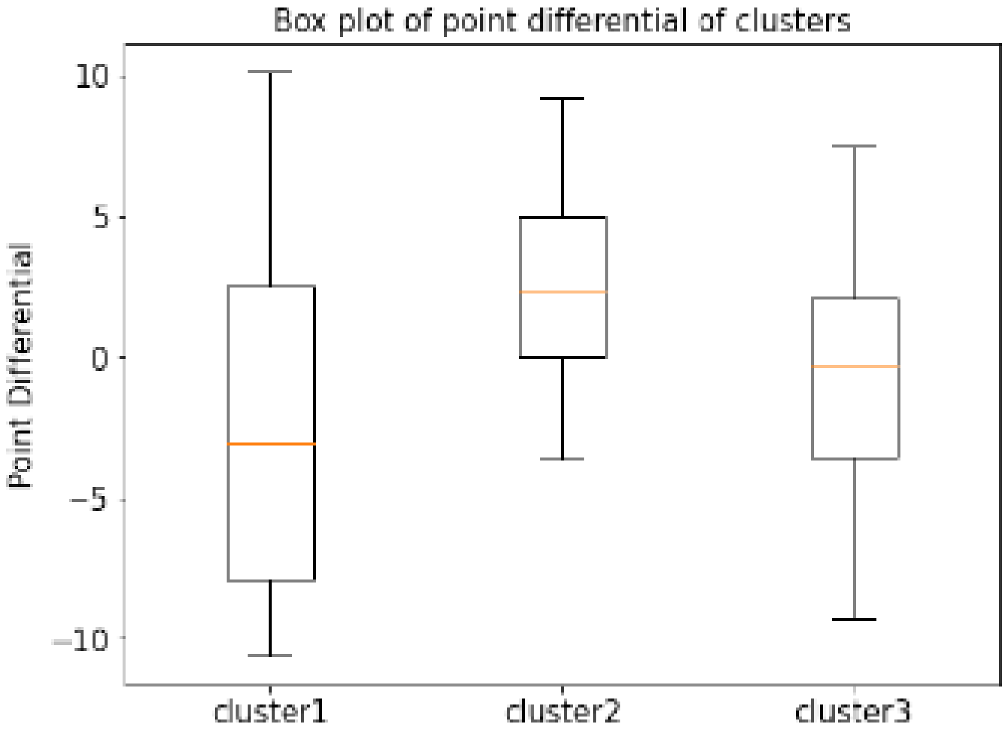
[27]:
```
    X3_PCT_T  X3_PCT_ATT_T
13     0.349        0.3579
46     0.389        0.4041
69     0.364        0.4560
```

[28]:
```
plt.figure(figsize=(8,6))

for i in range(k):
    cluster_data = data[data['cluster']==i]
    cluster_label = 'cluster ' + str(i+1)
    plt.scatter(cluster_data['X3_PCT_T'],cluster_data['X3_PCT_ATT_T'],␣
 ↪label=cluster_label)

centers = data.groupby('cluster').mean().reset_index()
plt.
 ↪scatter(centers['X3_PCT_T'],centers['X3_PCT_ATT_T'],linewidths=5,marker='x',s=100,␣
 ↪c='black', label='centroid')
plt.scatter(champion['X3_PCT_T'],champion['X3_PCT_ATT_T'],marker='s',s=150,␣
 ↪edgecolor='black', facecolors='none', label="champion")
plt.legend(fontsize=13)
plt.xlabel(' 3-Point Accuracy (Team)', fontsize=15)
plt.ylabel('3-Point Attempt % (Team)', fontsize=15)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.title('Cluster Result', fontsize=15)
plt.show()
```

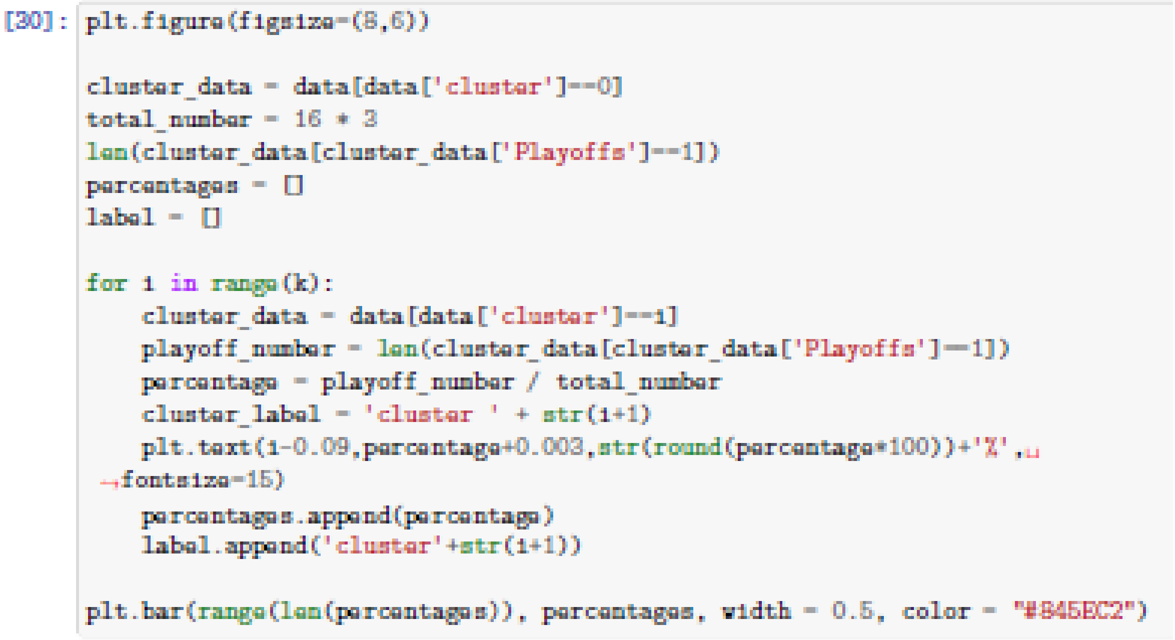


0.9 Box plot of the clustering result

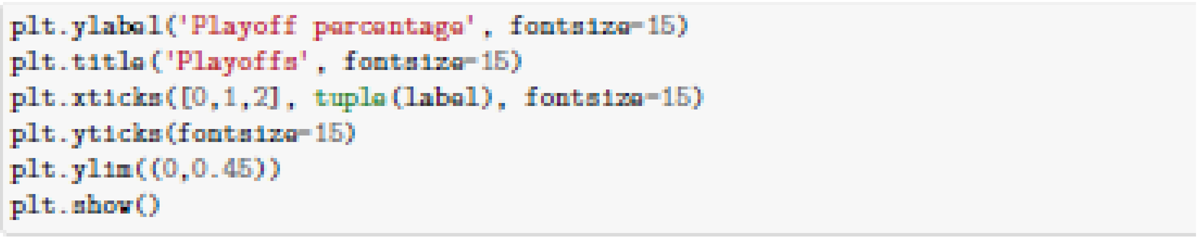
```
[29]: plt.figure(figsize=(8,6))
      boxes = []
      label = []
      for i in range(k):
          box = data.loc[data['cluster']==i, 'PTS_DIFF']
          boxes.append(box)
          label.append('cluster'+str(i+1))
      plt.ylabel('Point Differential', fontsize=15)
      plt.boxplot(boxes, labels=tuple(label))
      plt.xticks(fontsize=15)
      plt.yticks(fontsize=15)
      plt.title('Box plot of point differential of clusters', fontsize=15)
      plt.show()
```



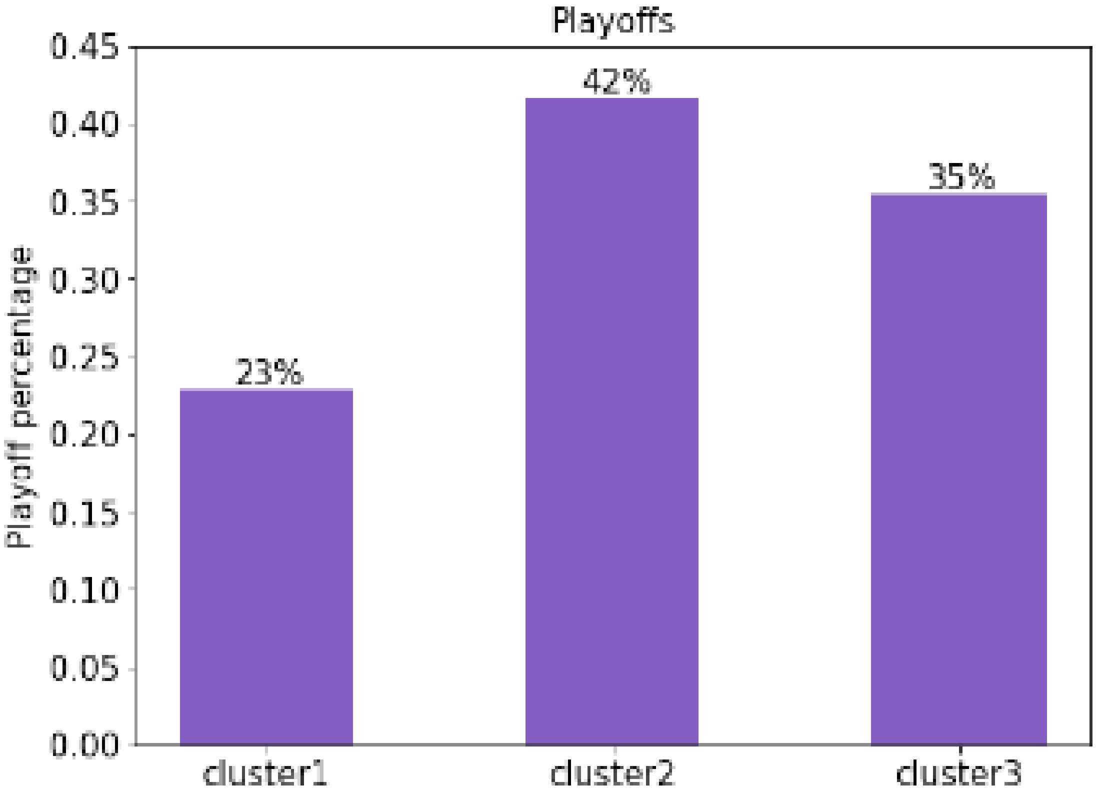
0.10 Percentage



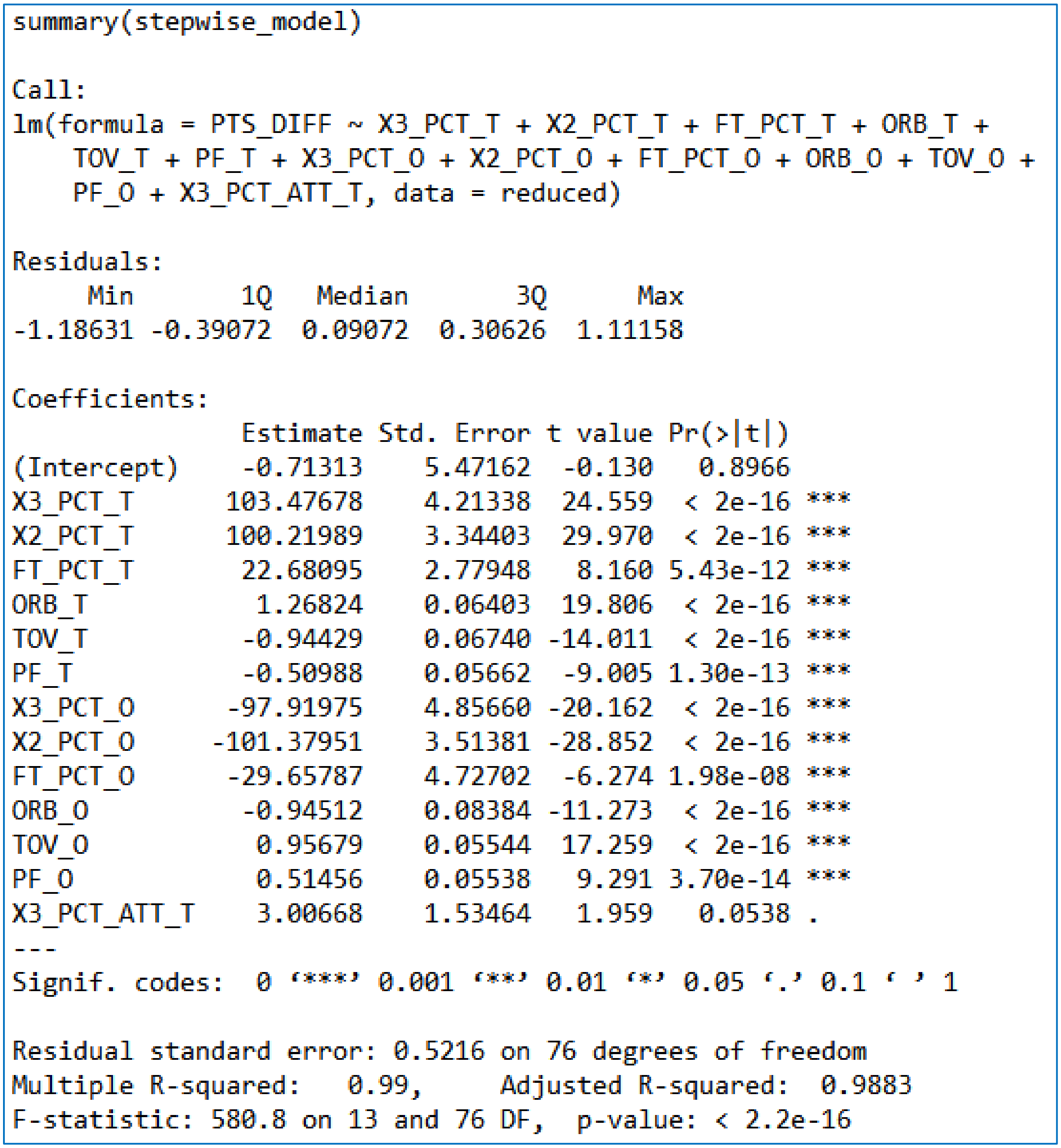
```
[30]: plt.figure(figsize=(8,6))

      cluster_data = data[data['cluster']==0]
      total_number = 16 * 3
      len(cluster_data[cluster_data['Playoffs']==1])
      percentages = []
      label = []

      for i in range(k):
          cluster_data = data[data['cluster']==i]
          playoff_number = len(cluster_data[cluster_data['Playoffs']==1])
          percentage = playoff_number / total_number
          cluster_label = 'cluster ' + str(i+1)
          plt.text(i-0.09,percentage+0.003,str(round(percentage*100))+'%',␣
       ↪fontsize=15)
          percentages.append(percentage)
          label.append('cluster'+str(i+1))

      plt.bar(range(len(percentages)), percentages, width = 0.5, color = "#845EC2")
```



```
plt.ylabel('Playoff percentage', fontsize=15)
plt.title('Playoffs', fontsize=15)
plt.xticks([0,1,2], tuple(label), fontsize=15)
plt.yticks(fontsize=15)
plt.ylim((0,0.45))
plt.show()
```



Appendix H: Summary of the Stepwise Selection Model



```
summary(stepwise_model)

Call:
lm(formula = PTS_DIFF ~ X3_PCT_T + X2_PCT_T + FT_PCT_T + ORB_T +
    TOV_T + PF_T + X3_PCT_O + X2_PCT_O + FT_PCT_O + ORB_O + TOV_O +
    PF_O + X3_PCT_ATT_T, data = reduced)

Residuals:
     Min       1Q   Median       3Q      Max
-1.18631 -0.39072  0.09072  0.30626  1.11158

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   -0.71313    5.47162  -0.130   0.8966
X3_PCT_T     103.47678    4.21338  24.559  < 2e-16 ***
X2_PCT_T     100.21989    3.34403  29.970  < 2e-16 ***
FT_PCT_T      22.68095    2.77948   8.160 5.43e-12 ***
ORB_T          1.26824    0.06403  19.806  < 2e-16 ***
TOV_T         -0.94429    0.06740 -14.011  < 2e-16 ***
PF_T          -0.50988    0.05662  -9.005 1.30e-13 ***
X3_PCT_O     -97.91975    4.85660 -20.162  < 2e-16 ***
X2_PCT_O    -101.37951    3.51381 -28.852  < 2e-16 ***
FT_PCT_O     -29.65787    4.72702  -6.274 1.98e-08 ***
ORB_O         -0.94512    0.08384 -11.273  < 2e-16 ***
TOV_O          0.95679    0.05544  17.259  < 2e-16 ***
PF_O           0.51456    0.05538   9.291 3.70e-14 ***
X3_PCT_ATT_T   3.00668    1.53464   1.959   0.0538 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5216 on 76 degrees of freedom
Multiple R-squared:   0.99,     Adjusted R-squared:  0.9883
F-statistic: 580.8 on 13 and 76 DF,  p-value: < 2.2e-16
```

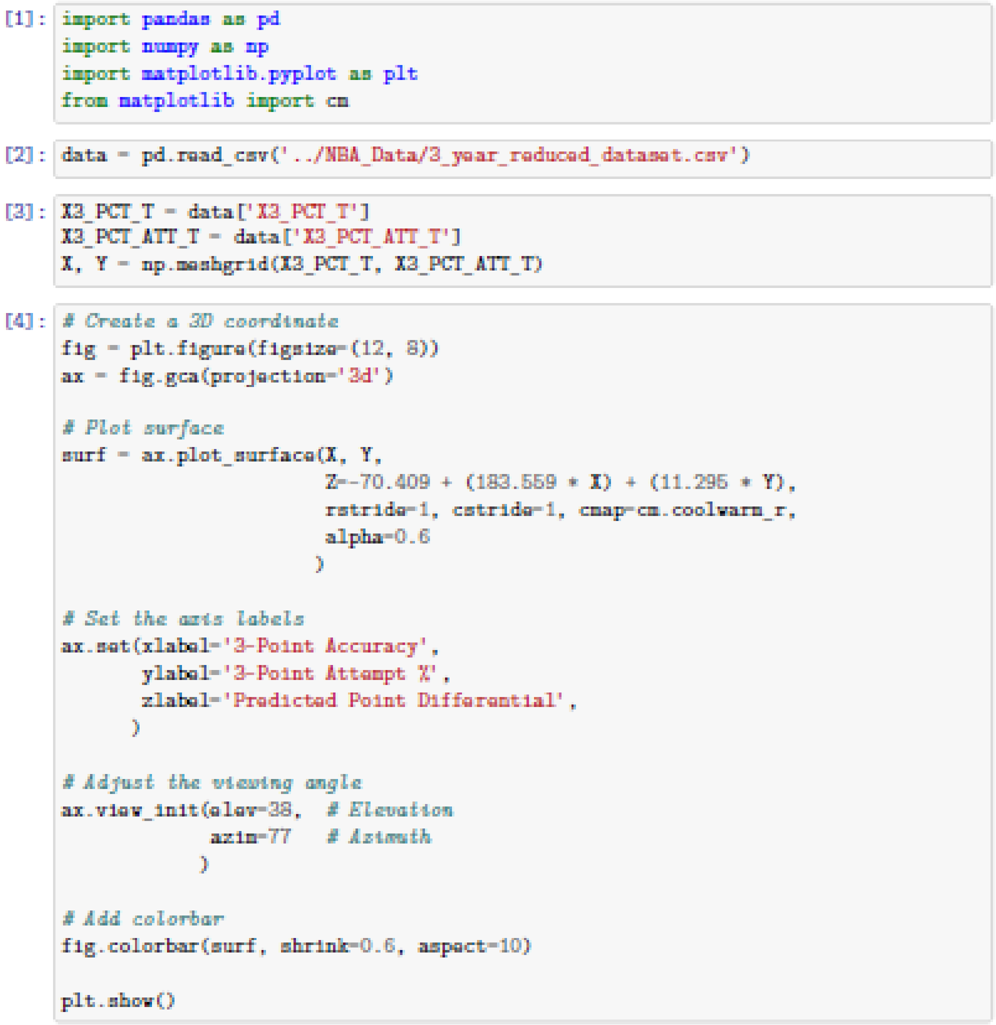
Appendix I: R Code for the Stepwise Selection Model

```
library(car)
setwd('C:/fun/Australia/2022 Semester 2/DATA7001/Group Project/NBA Data/csv')

# Read in the last three seasons of data.
teams_opponents <- read.csv("3_year_reduced_dataset.csv")

# Select only the 17 candidate features and response variable.
reduced <- teams_opponents[,3:20]

# Fit the full model that contains all 17 candidate features.
full_model <- lm(data=reduced,PTS_DIFF ~ .)

# Perform stewise selection on the full model.
stepwise_model <- step(full_model,direction = 'both',k = 3)

# Print a summary of the stepwise selection model with 13 features.
summary(stepwise_model)

# Calculate the Type III Sums of Squares for the bar graph in Tableau.
Anova(stepwise_model,data = reduced,type = "III")
```

Appendix J: R Code for the Stepwise Model Diagnostics

```
setwd('C:/fun/Australia/2022 Semester 2/DATA7001/Group Project/NBA Data/csv')

# Read in the dataset with the last three seasons of NBA data.
last_three_seasons <- read.csv("3_year_reduced_dataset.csv")

variables_to_keep_13 <-
c('PTS_DIFF','X3_PCT_T','X2_PCT_T','FT_PCT_T','ORB_T','TOV_T','PF_T','X3_PCT_O','X2_PCT_O','
FT_PCT_O','ORB_O','TOV_O','PF_O','X3_PCT_ATT_T')
variable_subset_13 <- last_three_seasons[,(names(last_three_seasons) %in%
(variables_to_keep_13))]
print(variable_subset_13)
# Fit a linear regression model
stepwise_model<- lm(formula = PTS_DIFF ~ X3_PCT_T + X2_PCT_T + FT_PCT_T + ORB_T +TOV_T
+ PF_T + X3_PCT_O + X2_PCT_O + FT_PCT_O + ORB_O + TOV_O + PF_O + X3_PCT_ATT_T, data =
variable_subset_13)
summary(stepwise_model)

# Obtain the Variance Inflaction Factors from the stepwise regression model.
vif(stepwise_model)

###To analyzing residuals
## -The mean of zeros is zero (and the sum of the errors is zero)
## -The distribution of the erros are normal
## -All of the errors are independent
## -Variance of errors is constant (Homoscedastic)

layout(matrix(c(1,1,2,3),2,2,byrow=T))
#Linear Regression x Residuals Plot
windows(10,5)
plot(stepwise_model$resid~predict(stepwise_model),
    main="Linear Regression x Residuals Plot",
    xlab="Predicted Point Differential",ylab="Residuals")
abline(h=0,lty=3)

#Histogram of Residuals
hist(stepwise_model$resid, main="Histogram of Residuals",     ylab="Residuals")
#Q-Q Plot -> close to the line is better
qqnorm(stepwise_model$resid)
qqline(stepwise_model$resid)

###The Jarque-Bera test (fBasics library)
jarqueberaTest(stepwise_model$resid)
```

Appendix K: Python Code for the 2 Feature Regression Model 3D Plot



[1]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm
```

[2]:

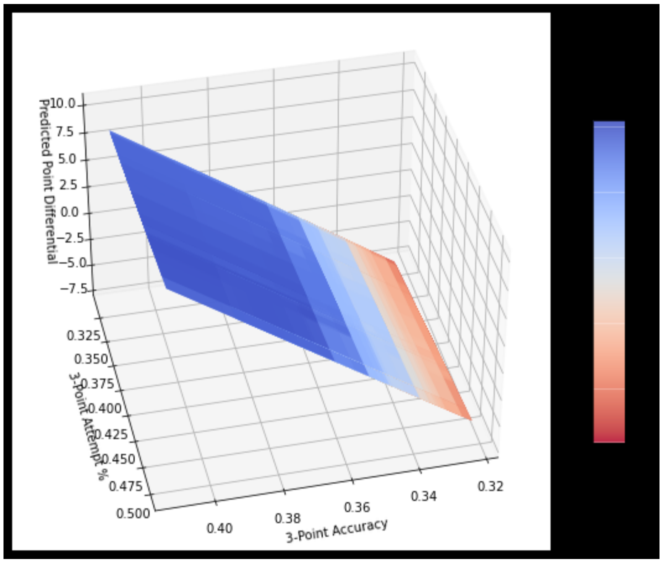
```python
data = pd.read_csv('../NBA_Data/3_year_reduced_dataset.csv')
```

[3]:

```python
X3_PCT_T = data['X3_PCT_T']
X3_PCT_ATT_T = data['X3_PCT_ATT_T']
X, Y = np.meshgrid(X3_PCT_T, X3_PCT_ATT_T)
```

[4]:

```python
# Create a 3D coordinate
fig = plt.figure(figsize=(12, 8))
ax = fig.gca(projection='3d')

# Plot surface
surf = ax.plot_surface(X, Y,
                       Z=-70.409 + (183.559 * X) + (11.295 * Y),
                       rstride=1, cstride=1, cmap=cm.coolwarm_r,
                       alpha=0.6
                      )

# Set the axis labels
ax.set(xlabel='3-Point Accuracy',
       ylabel='3-Point Attempt %',
       zlabel='Predicted Point Differential',
      )

# Adjust the viewing angle
ax.view_init(elev=38,  # Elevation
             azim=77   # Azimuth
            )

# Add colorbar
fig.colorbar(surf, shrink=0.6, aspect=10)

plt.show()
```


Predicted Point Differential
3-Point Attempt %
3-Point Accuracy