FINAL REPORT 604

NBA Analysis

Marcelino Rodriguez
Matias Totz
Rajvir Kaur
Nyadual Makuach

Contents

Table of Figures

1 Introduction

When analyzing sports, particularly basketball, there are numerous factors that can contribute to a team's success. These factors are the key motivation behind this analysis. This raises the question: What are those factors (if any)? Are these factors variable? Are they measurable? Can they be predictive? Can they grow in relevance to directly impact a team's victory, or do they only influence specific aspects of a match? These are some of the questions that can be explored through data analysis. With sufficient data, it may even be possible to draw meaningful conclusions and gain a deeper understanding of the variables that affect the success of a basketball team.

In this project, we aim to answer one central question: What makes a basketball team successful? This question is intentionally broad to allow for various interpretations, as success in basketball is subjective and can mean different things depending on whom you ask. For some, success may be winning championships, while for others, it may be financial stability or media popularity.

To approach this question, this analysis will use and study four datasets that cover different aspects of basketball-related data. These datasets encompass team performance, player demographics, coach relevance, and basketball economics. The combination of these data sets aims to provide insights into the various variables that could influence team success. The objective of this project is to analyze the data and uncover patterns that provide a more comprehensive understanding of what drives success in basketball.

Looking into team performance, we hope to answer questions such as how does field goal percentage impact win rate. Do teams with higher three-point shooting efficiency win more games. How does the average performance of players impact the overall team’s success.

For our coach analysis, we are interested in exploring if coaches truly make an impact on their teams. We will explore if coaches with more years of experience outperform less experienced coaches. Do coaches with previous NBA experience have an advantage. Do team budgets influence what type of coach they choose. We will also investigate if there is a relationship between the country of origin and a coach’s performance.

When analyzing the demographics of players, we are hoping to see how height impacts a player. How does the weight of a team impact their success. How do teams perform when they have more international players than domestic players.

Finally, when looking into team budgets and salaries we want to figure out the relationship between players and their team budgets. Do players who get paid more perform better. Do teams with higher budgets have more success. Are players with better statistics paid higher than their teammates.

2 Data Sets

In this project, we are using five distinct datasets to analyze various aspects of basketball, including team performance, player demographics, and financial information. Each dataset comes from different sources and has unique features that contribute to our overall analysis. Below is an overview of these datasets:

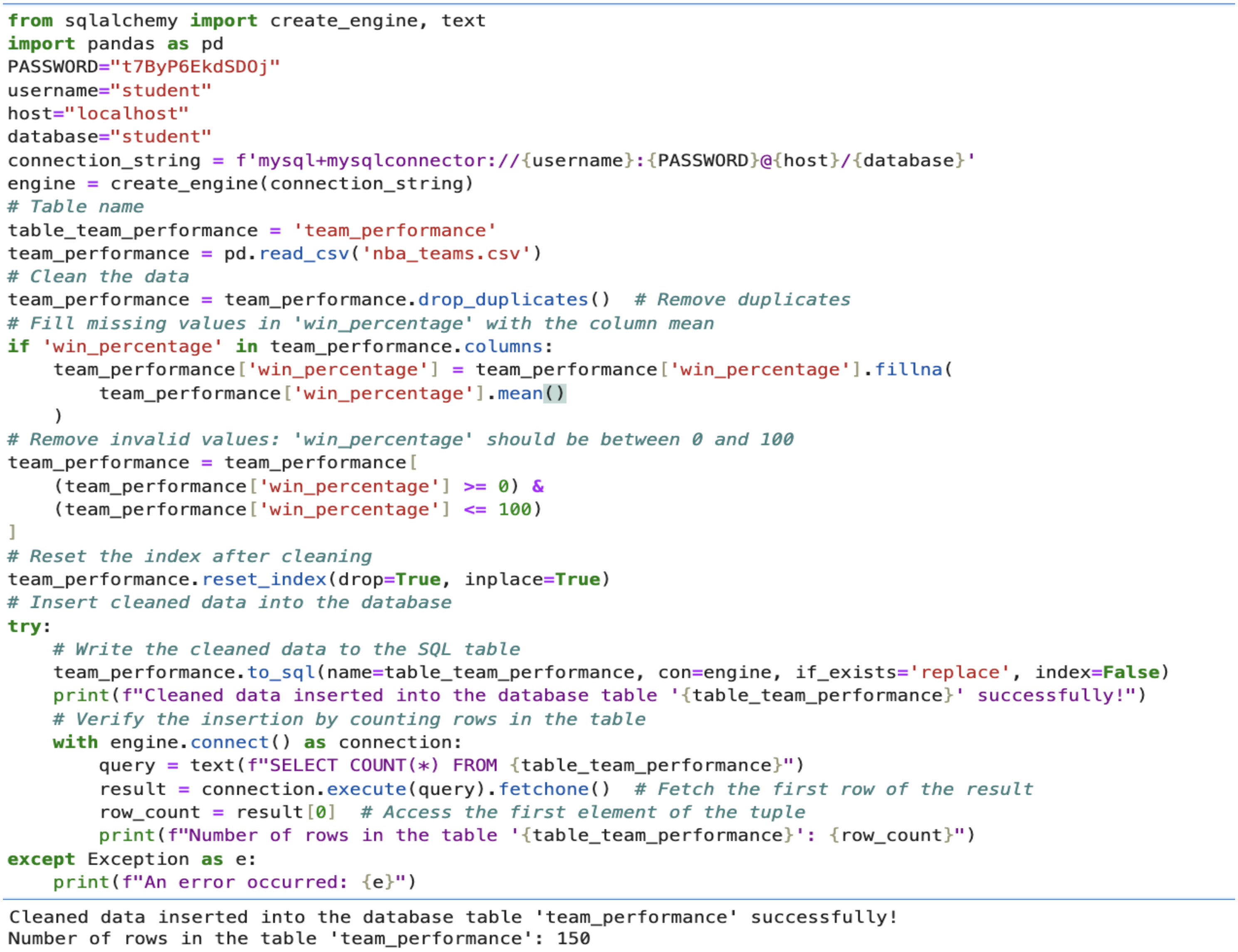
2.1 Teams Performance – Rajvir Kaur

This dataset includes metrics that track team performance across different seasons. It captures variables such as games played, wins, losses, playoff appearances, and championships won. This data will help us understand how performance metrics correlate with team success. This dataset represents comprehensive statistical data on NBA team performance across multiple seasons. Each row corresponds to a specific team's performance in a particular season, capturing metrics that describe both offensive and defensive contributions, shooting efficiency, and overall gameplay. Key metrics include the number of games played, wins, losses, and win percentage, alongside detailed statistics such as points scored, rebounds, assists, and shooting percentages (field goal, three-point, and free throw). This dataset serves as a valuable resource for understanding the factors that drive team success in the NBA. The dataset, downloaded from Kaggle, contains performance data for NBA teams and includes 29 columns and 700 rows from 2000 – 2023 year. As stated on the Kaggle dataset, all data was collected through NBA sources which are public and allow for personal usage.

Cleaning of Data:

This project focuses on cleaning a dataset and inserting the cleaned data into a MySQL database using SQLAlchemy. It begins by establishing a secure connection to the MySQL database through a connection string. The dataset is loaded from a CSV file into a Pandas DataFrame, where initial cleaning is performed. Duplicate rows are removed to ensure data consistency, and missing values in the win_percentage column are replaced with the column's mean. Invalid rows where win_percentage falls outside the range of 0 to 100 are filtered out, and numeric columns like games_played, wins, and losses are checked to ensure they contain non-negative values. After cleaning, the DataFrame index is reset to maintain consistency.

Once the cleaning process is complete, the script writes the cleaned data into a MySQL table named team_performance, replacing any existing table with the same name. It then verifies the successful insertion by executing a SQL query to count the rows in the table, confirming that the data has been stored correctly. Throughout the process, robust error handling is implemented to catch and log any issues that arise, ensuring the script's reliability and ease of debugging. This workflow provides a clean and validated dataset ready for further analysis in the database. After cleaning the dataset, we have 25 columns and 150 rows from year 2018-2023.

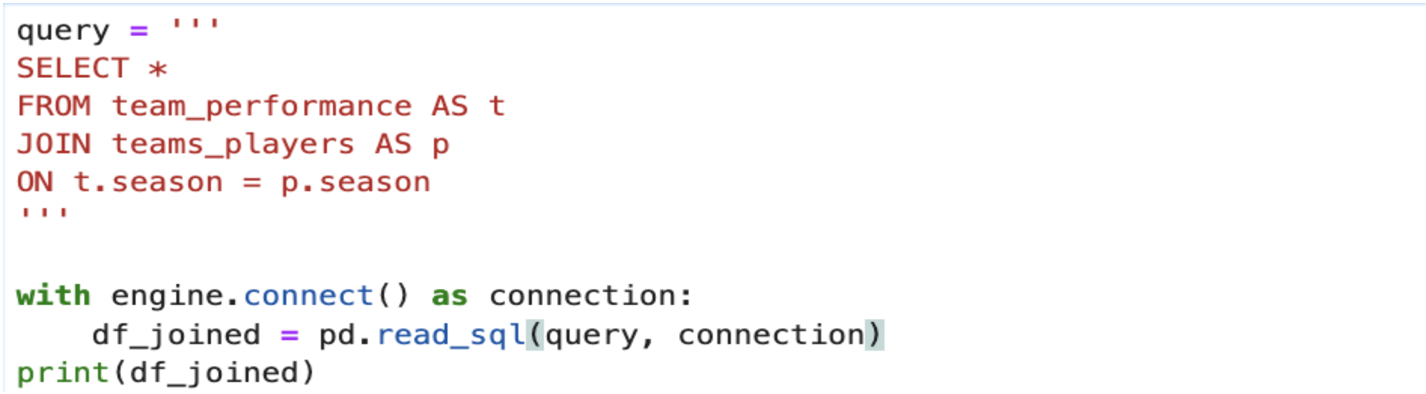


```python
from sqlalchemy import create_engine, text
import pandas as pd
PASSWORD="t7ByP6EkdSDOj"
username="student"
host="localhost"
database="student"
connection_string = f'mysql+mysqlconnector://{username}:{PASSWORD}@{host}/{database}'
engine = create_engine(connection_string)
# Table name
table_team_performance = 'team_performance'
team_performance = pd.read_csv('nba_teams.csv')
# Clean the data
team_performance = team_performance.drop_duplicates()  # Remove duplicates
# Fill missing values in 'win_percentage' with the column mean
if 'win_percentage' in team_performance.columns:
    team_performance['win_percentage'] = team_performance['win_percentage'].fillna(
        team_performance['win_percentage'].mean()
    )
# Remove invalid values: 'win_percentage' should be between 0 and 100
team_performance = team_performance[
    (team_performance['win_percentage'] >= 0) &
    (team_performance['win_percentage'] <= 100)
]
# Reset the index after cleaning
team_performance.reset_index(drop=True, inplace=True)
# Insert cleaned data into the database
try:
    # Write the cleaned data to the SQL table
    team_performance.to_sql(name=table_team_performance, con=engine, if_exists='replace', index=False)
    print(f"Cleaned data inserted into the database table '{table_team_performance}' successfully!")
    # Verify the insertion by counting rows in the table
    with engine.connect() as connection:
        query = text(f"SELECT COUNT(*) FROM {table_team_performance}")
        result = connection.execute(query).fetchone()  # Fetch the first row of the result
        row_count = result[0]  # Access the first element of the tuple
        print(f"Number of rows in the table '{table_team_performance}': {row_count}")
except Exception as e:
    print(f"An error occurred: {e}")
```

```
Cleaned data inserted into the database table 'team_performance' successfully!
Number of rows in the table 'team_performance': 150
```

Joining of Dataset with another Dataset:

To join the team_performance table with the player_demographics table using an INNER JOIN and the season column as a common key as shown in Figure 2.1, the query combines rows where both the Team and season columns match. The result includes team-level metrics such as games_played, wins, and win_percentage from the team_performance table, alongside player-specific attributes like Player_Name, Age, Position, and Nationality from the player_demographics table. This ensures that each row represents a player's demographic details paired with their team's performance data for a specific season. The join condition ensures both tables align accurately, duplicating team statistics for all players on the same team. Additional filters can be applied using a WHERE clause to focus on specific teams, seasons, or player roles. It is crucial to ensure the season column is consistently formatted across both datasets. This approach provides a comprehensive view of team performance and player demographics, enabling detailed analysis. In the below Figure 2.1, the team_players is a player_demographics data. Based on these datasets, we have answered the research questions.



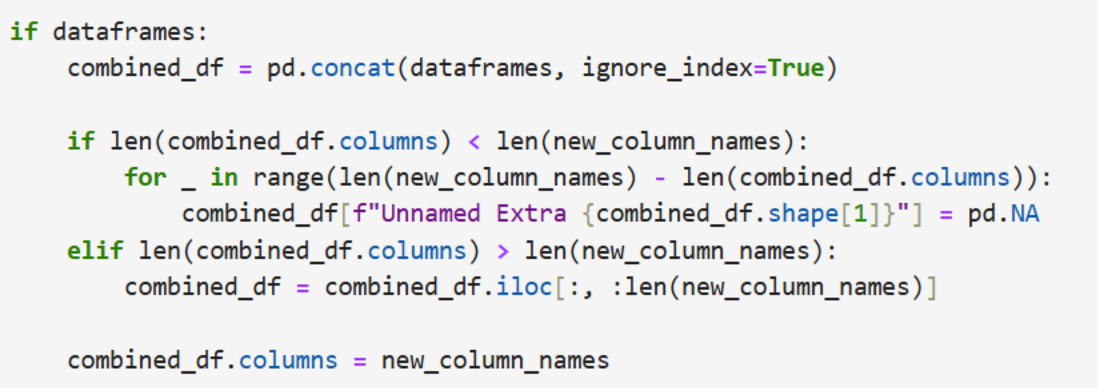
```
query = '''
SELECT *
FROM team_performance AS t
JOIN teams_players AS p
ON t.season = p.season
'''

with engine.connect() as connection:
    df_joined = pd.read_sql(query, connection)
print(df_joined)
```

Figure 2.1: Joining of Datasets

2.2 Coach Dataset – Matias Totz

Going into this project, we wanted to learn more about the leadership of NBA teams and how they impact success. The impacts of leadership can be found in many different areas such as parents, teachers, team captains, or even informal leaders at work. By analyzing the impacts of coaches on their NBA teams, we are hoping to find answers to how much of an impact leadership can have and what aspects of leaders produce the most significant results. We are most interested in exploring a few questions when looking at coaches within the NBA. We are interested in exploring if coaches get better with more experience, if coaches perform better when they have previously played in the NBA, if teams with high budgets prefer specific coaches, and finally if certain countries produce better coaches than others. This dataset is unique since it will be a combination of data found from multiple sources. The main source of data will be coming from basketball-reference.com where you can find data tables of each team's coach, how many seasons the coach has coached, the performance of the team during the regular season and during the playoffs. We will also be adding a few extra columns where we will capture where the coach studied, salary, and if the coach played in the NBA. All data sources use data that is public and both the NBA and basketball-refernce.com allow personal usage of their data. One key concern is that coach salaries are not made public, however you can find estimates online that are usually very accurate. By far the most interesting question we want to explore with this dataset will be if coaches who played in the NBA when they were younger perform better than non-NBA coaches. We also are interested in seeing how much of an influence education has on coaches since we would expect that education has less of an impact on players but more of an impact on coaches since they need to think strategically and communicate with their players.

We specifically chose this dataset because it gave us the career win rate, the years of experience, and the coach's name. Based on this, we were able to combine it with many other sources to create the perfect dataset. To begin the data cleaning, we had to complete a ton of work on the coach data since we were using multiple sources, and the main source was very difficult to work with without any preparation. Using five different years, we combined each year into a cleaned data frame. The most difficult aspects of this step were that numerous columns had very little data for specific years, and we were cleaning each dataset one by one. This caused issues since we would be combining a cleaned dataset with an unclean dataset and so the columns would not match up. The biggest challenge was writing code to combine the top few rows into one descriptive title.

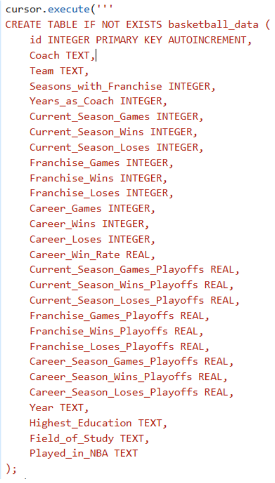


```
if dataframes:
    combined_df = pd.concat(dataframes, ignore_index=True)

    if len(combined_df.columns) < len(new_column_names):
        for _ in range(len(new_column_names) - len(combined_df.columns)):
            combined_df[f"Unnamed Extra {combined_df.shape[1]}"] = pd.NA
    elif len(combined_df.columns) > len(new_column_names):
        combined_df = combined_df.iloc[:, :len(new_column_names)]

    combined_df.columns = new_column_names
```

2.2 Figure 1: Coach Dataset: Combining Datasets with Different Columns

Once we had the cleaned basketball dataset, we needed to add in a few extra columns that would play a roll in our future analysis. These columns included the coaches highest education, the field of study, and if they played in the NBA. All of this data came from online sources and Wikipedia so we relied heavily on the accuracy of this data being correct. We then were able to merge the new dataset with the cleaned dataset with a left join using the coaches name as the on field.

Our next step was creating a basic SQL database to store our newly created data so we could create queries.



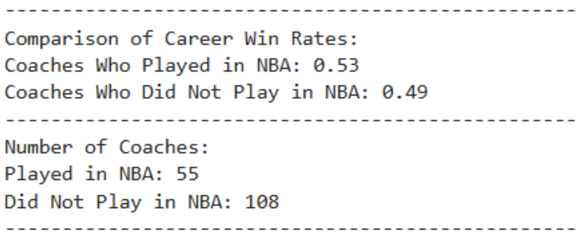
```
cursor.execute('''
CREATE TABLE IF NOT EXISTS basketball_data (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    Coach TEXT,
    Team TEXT,
    Seasons_with_Franchise INTEGER,
    Years_as_Coach INTEGER,
    Current_Season_Games INTEGER,
    Current_Season_Wins INTEGER,
    Current_Season_Loses INTEGER,
    Franchise_Games INTEGER,
    Franchise_Wins INTEGER,
    Franchise_Loses INTEGER,
    Career_Games INTEGER,
    Career_Wins INTEGER,
    Career_Loses INTEGER,
    Career_Win_Rate REAL,
    Current_Season_Games_Playoffs REAL,
    Current_Season_Wins_Playoffs REAL,
    Current_Season_Loses_Playoffs REAL,
    Franchise_Games_Playoffs REAL,
    Franchise_Wins_Playoffs REAL,
    Franchise_Loses_Playoffs REAL,
    Career_Season_Games_Playoffs REAL,
    Career_Season_Wins_Playoffs REAL,
    Career_Season_Loses_Playoffs REAL,
    Year TEXT,
    Highest_Education TEXT,
    Field_of_Study TEXT,
    Played_in_NBA TEXT
);
```

2.1 Figure 2: Basic SQL Database

Once we finished this step, we were able to complete the individual analysis specifically on the coach dataset. Overall we learned that the data would show promising results such as coaches who played in the NBA having a 4% higher win rate than coaches without NBA experience and coaches with relevant education having a 3% higher win rate than coaches without relevant education but neither were statistically significant. We also looked into the impact of having more years of experience as a coach on their win rate and our results were promising but not statistically significant. We ran into major issues with our coach dataset since we were unable to find any areas that were statistically significant. This may be the result of poor-quality data and a lack of data points for specific statistical testing. As stated earlier, a big part of the dataset was built on various articles from the internet which may have been incorrect. We also only used four seasons of data since the cleaning process was very difficult and time consuming.

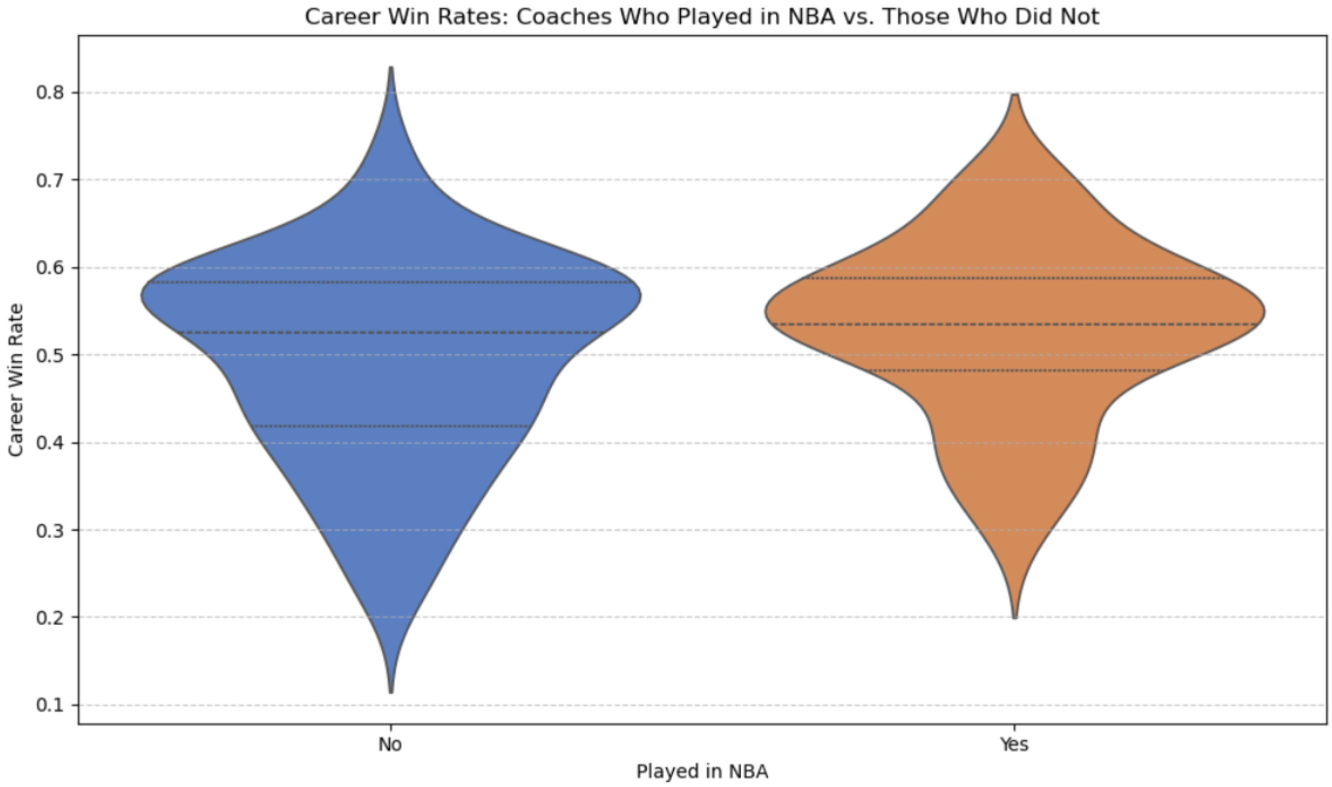
Individual Analysis Key Findings:

When comparing the average win rates of coaches, there does not appear to be a relationship between having played in the NBA and having a better win rate. Although the average does seem higher for coaches with NBA experience, our graph says otherwise.



```
--------------------------------------------------
Comparison of Career Win Rates:
Coaches Who Played in NBA: 0.53
Coaches Who Did Not Play in NBA: 0.49
--------------------------------------------------
Number of Coaches:
Played in NBA: 55
Did Not Play in NBA: 108
--------------------------------------------------
```

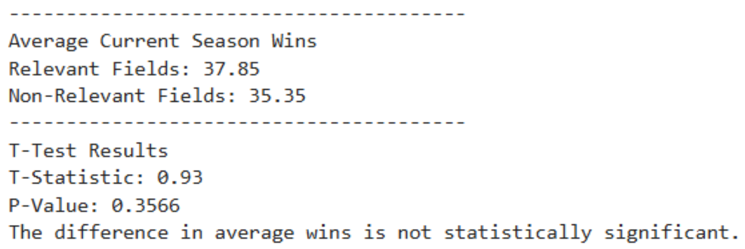
2.2 Figure 3: Win Rates of Coaches with NBA Experience vs Coaches without



2.2 Figure 4: Win Rates of Coaches with Different NBA Experience

As you can see in the graph above, although coaches without previous NBA experience tend to have more extreme outliers with a higher maximum win rate and a lower minimum win rate, both experiences tend to have most coaches in the 50-60% win rate section.

Next, we looked at NBA coaches with relevant education such as physical education or business and compared them to coaches with nonrelevant education such as biology or American history.

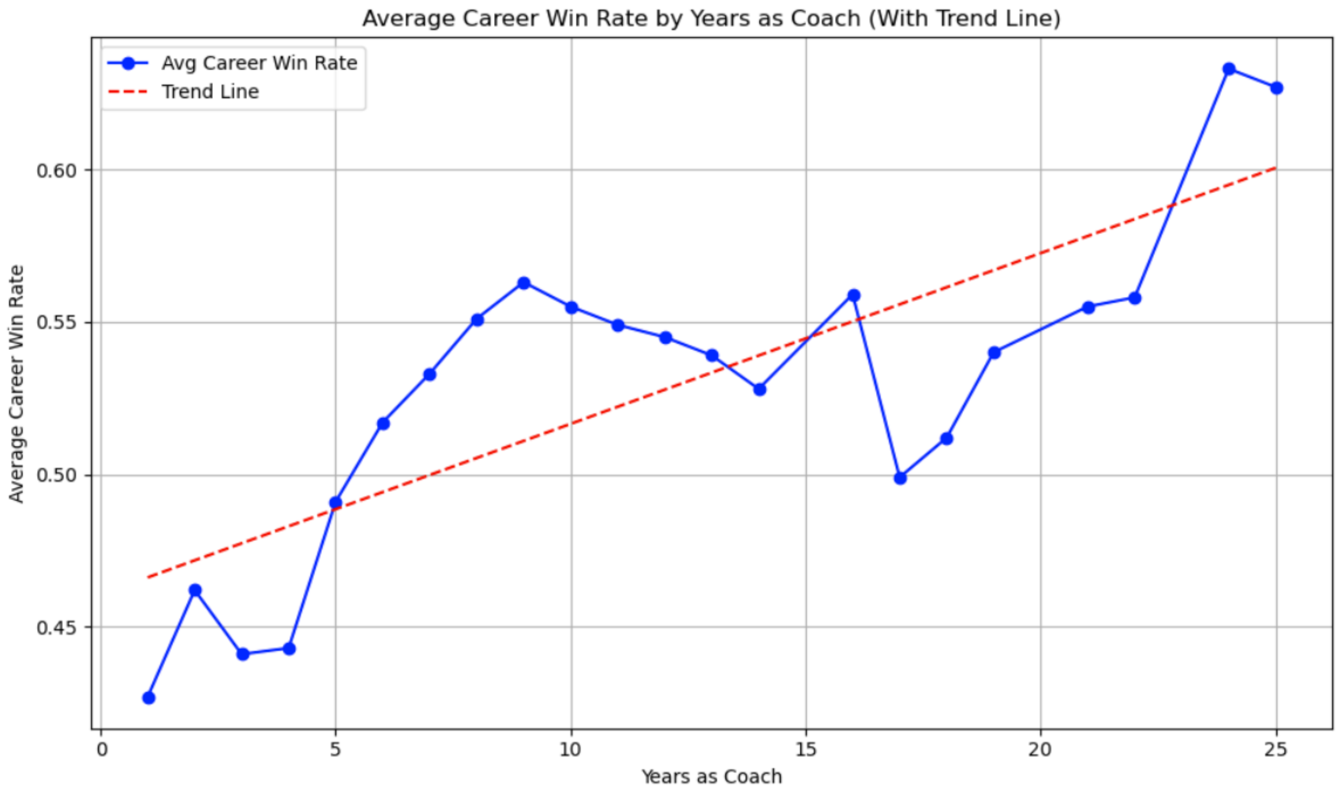


```
----------------------------------------
Average Current Season Wins
Relevant Fields: 37.85
Non-Relevant Fields: 35.35
----------------------------------------
T-Test Results
T-Statistic: 0.93
P-Value: 0.3566
The difference in average wins is not statistically significant.
```

2.2 Figure 5: Does Education of Coaches Matter

Although it shows that the average win rate of coaches with relevant education do have higher win rates, it simply is not statistically significant. In the future, it would be nice to have had more accurate data and chosen questions that we could have used on the entire dataset. Many of our questions that we explore tend to filter the data down to just a few datapoints that we can not even run statistical analysis on properly.

We then looked at the impact of years of experience on the average win rate of coaches.



2.2 Figure 6: Years of Experience and Win Rate

As you can see from our graph, there does seem to be an upwards trend where coaches with more years of experience have better win rates. However, once coaches get to year 15, there seems to be a significant drop-off and many coaches with this range of experience tend to underperform. It is only until coaches get closer to 25 years of experience that they jump back up in performance.

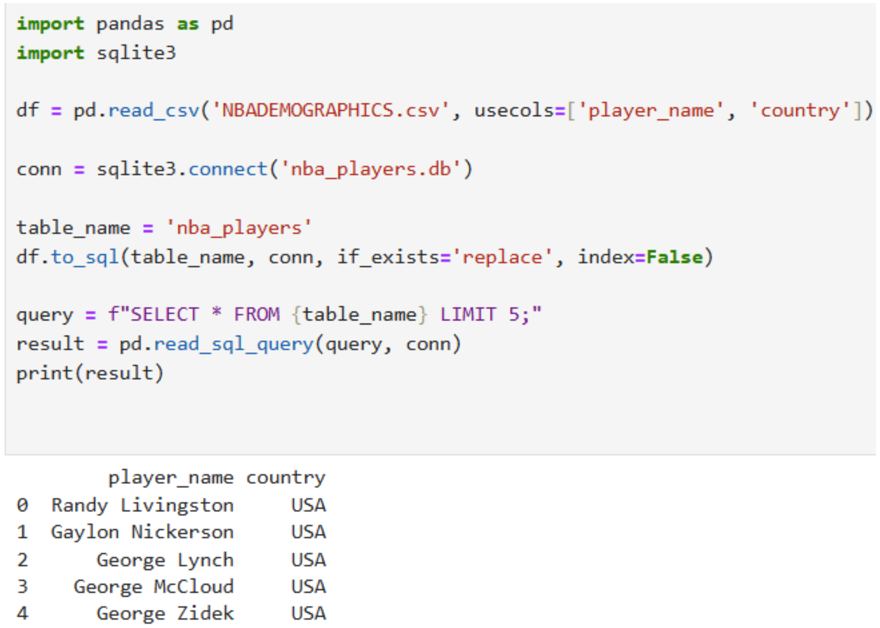
After conducting analysis on the coach dataset, we decided to combine it with our other datasets. We first combined the dataset with the budget dataset to see if there was any relationship between the team budget and which coach they chose. However, this resulted in non-statistical relationships.

After exploring a few other questions, we decided to combine the coach dataset with the demographic dataset and team performance dataset. Combining the datasets proved to be the most difficult aspect of the coach analysis since so many aspects of the data did not match and needed to be cleaned. We started off by loading the entire demographic dataset into a SQL database and pulled both the player's name and the country they are from. The coolest aspect of this analysis was that we were able to find a unique way of combining the coach dataset with other datasets by realizing that many of the coaches were at some point a player and could be found on our other datasets.

After pulling the player names and countries for over twenty different seasons hoping to capture old enough seasons where some of the coaches might have played in, we then loaded the team performance dataset. We decided to pull the team's name, win percentage, and season. This was to answer a unique question brought up during our presentation where someone asked to explore coaches who played in the NBA and played on great teams. The hypothesis would be that a coach who previously played in the NBA and on a great team may be a better coach than a coach who played in the NBA but on a bad team.

Once we had both datasets, we joined the original data using the coaches name as the primary key with both of them to produce a final dataset that contained the coaches name, the team they played for when they were in the NBA, that team's win percentage, their country, and their win rate as a coach. This was an incredible challenge to build, however we managed to find a way to combine three of our datasets together.

Unfortunately, we were unable to find any statistical significance between coaches NBA win percentage and their coaching win percentage. Our biggest challenge was finding enough data to build our models on since there were only a few coaches who actually played in the NBA and most countries were USA. However, the biggest focus of this project was on finding a unique way to combine datasets that are not necessarily meant to be combined. Although we did not find any statistical significance of NBA coaches, we were able to find a unique way of combining datasets.

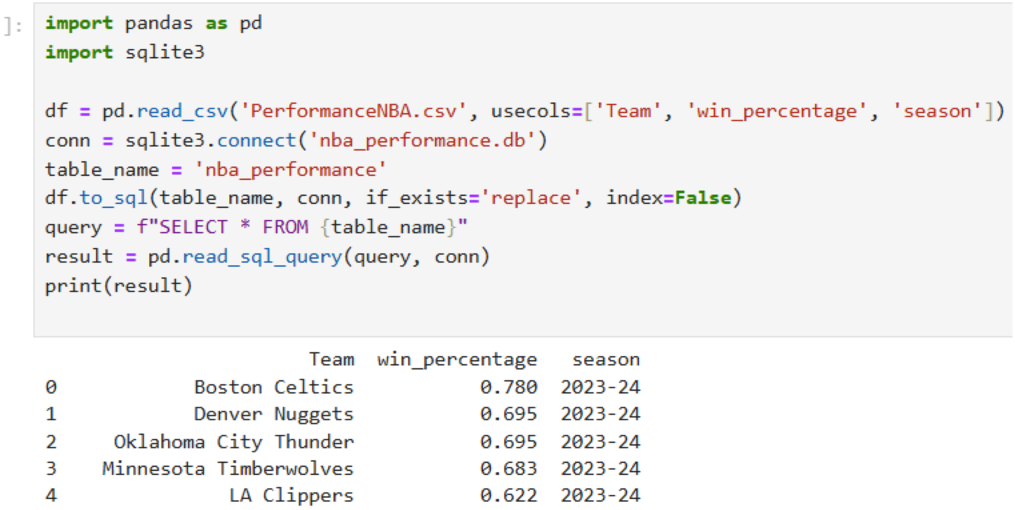


```
import pandas as pd
import sqlite3

df = pd.read_csv('NBADEMOGRAPHICS.csv', usecols=['player_name', 'country'])

conn = sqlite3.connect('nba_players.db')

table_name = 'nba_players'
df.to_sql(table_name, conn, if_exists='replace', index=False)

query = f"SELECT * FROM {table_name} LIMIT 5;"
result = pd.read_sql_query(query, conn)
print(result)
```

```
        player_name country
0  Randy Livingston     USA
1  Gaylon Nickerson     USA
2      George Lynch     USA
3    George McCloud     USA
4      George Zidek     USA
```

Combining Data with Coaches Step 1

The NBADEMOGRAPHICS.csv is the player demographic dataset however we pull every single season up until 2000-01. We are specifically interested in identifying any trends with the country of origin and the impact on coaching ability. Unfortunately, looking back at this analysis, we ran into many problems because most coaches who previously played in the NBA who we were able to pick out from the player demographic dataset were mostly USA players. It was very uncommon for any players to be drafted from outside North America in the early 2000s and unfortunately, many of the coaches who previously played in the NBA came from these times. However, it was fascinating to find a way to pull coaches who were previous players from our dataset by utilizing the older data stored.



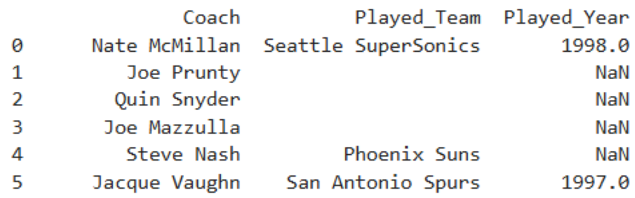
```
import pandas as pd
import sqlite3

df = pd.read_csv('PerformanceNBA.csv', usecols=['Team', 'win_percentage', 'season'])
conn = sqlite3.connect('nba_performance.db')
table_name = 'nba_performance'
df.to_sql(table_name, conn, if_exists='replace', index=False)
query = f"SELECT * FROM {table_name}"
result = pd.read_sql_query(query, conn)
print(result)
```

```
                     Team  win_percentage   season
0          Boston Celtics           0.780  2023-24
1          Denver Nuggets           0.695  2023-24
2   Oklahoma City Thunder           0.695  2023-24
3  Minnesota Timberwolves           0.683  2023-24
4             LA Clippers           0.622  2023-24
```

Combining Data with Coaches Step 2

Our next step in combining our data with our coach dataset was similar to step one but with the team performance dataset. Since we were not able to pull the coaches name from this dataset, we decided to look up the last team the coach played for if they played in the NBA. Then we pulled their teams' win rate from that season. So instead of finding out how good the coach was when they played in the NBA, we were able to track how good the team they played for was.

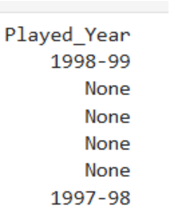
If a coach from the 2022-23 season previously played in the NBA, we needed to figure out what team they played for and what season. We did some searching and found data for some of the coaches which we pulled into a list and combined with our original dataset. This new dataset stored the coaches' name, the team they played for and the season. We used the most recent season they played in the NBA.



```
            Coach          Played_Team  Played_Year
0    Nate McMillan  Seattle SuperSonics       1998.0
1       Joe Prunty                               NaN
2      Quin Snyder                               NaN
3     Joe Mazzulla                               NaN
4       Steve Nash         Phoenix Suns          NaN
5    Jacque Vaughn    San Antonio Spurs       1997.0
```

Combining Data with Coaches Step 3

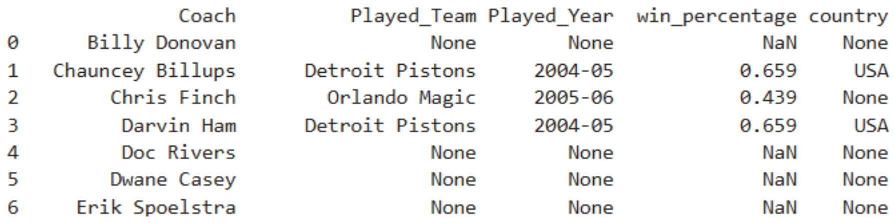
This is where we ran into a big problem. The datasets that we were using had different formats of the specific season. In one dataset, the season was written as 1998-99 and in our current dataset it was written as 1998.0. Therefore, we created a new list and replaced the old years with the correct format.



```
Played_Year
    1998-99
       None
       None
       None
       None
    1997-98
```

Combining Data with Coaches Step 4

Finally, we were able to combine all of the datasets by either joining on the coach's name, or by joining on the coach's team and season year. We also included the coach career win rate from our original dataset in the very last column that was not captured in the screenshot.



```
             Coach        Played_Team Played_Year  win_percentage country
0     Billy Donovan              None        None             NaN    None
1  Chauncey Billups    Detroit Pistons     2004-05           0.659     USA
2       Chris Finch      Orlando Magic     2005-06           0.439    None
3        Darvin Ham    Detroit Pistons     2004-05           0.659     USA
4        Doc Rivers              None        None             NaN    None
5       Dwane Casey              None        None             NaN    None
6    Erik Spoelstra              None        None             NaN    None
```

Combining Data with Coaches Step 5

2.3 Players Demographics – Nyadual Makuach

The dataset used in this analysis, NBA Players Data, was sourced from Kaggle and is licensed under Kaggle's data licensing terms under the CC BY-SA 4.0 license. This licensing allows the dataset to be used for personal and educational purposes. The dataset spans NBA seasons from 1996-1997 to 2022-2023, providing detailed information about players, including attributes such as age, height, weight, country, name, seasons, and teams. These attributes are essential for analyzing how player demographics influence team success in the NBA.

The dataset has a straightforward structure, containing 12,844 rows and 22 columns. Key attributes include player names, team abbreviations, age, height, weight, college attended, country of origin, and the season associated with each record. Covering data for 36 teams and 2,551 unique players, the dataset is provided in CSV format, making it easy to analyze using tools like Python and SQL.

Several challenges were encountered during the data cleaning process. Given the extensive timeframe of the dataset, many players had switched teams' multiple times. This made it challenging to track individual player contributions consistently. To address this issue, I grouped data by player ID to enable cumulative analysis. Additionally, I focused the analysis on a narrower timeframe, specifically the 2018-2019 to 2022-2023 seasons, to ensure the results were both accurate and relevant.

Inconsistencies in the dataset also required attention. For example, height and weight were recorded in mixed units, with some entries in centimeters and others in feet/inches. To resolve this, I standardized all measurements to centimeters. Furthermore, some team names varied across seasons, and outdated team names were present in the dataset. Aligning and standardizing team names, as well as removing irrelevant entries, was essential to maintain consistency and reliability in the analysis. These preprocessing steps were critical to preparing the dataset for analysis, ensuring the insights derived from the data were robust and meaningful. This groundwork provides a solid foundation for exploring the relationship between player demographics and team success in the NBA.

2.4 Players.’ Salaries and Teams Budget – Marcelino Rodriguez Anglada

These datasets have been created based on information from the website Hoopshype[4]. This website provides a list of players and teams by season, along with their respective salaries and budgets. The website only provides the data in list/table format. The format of how Hoopshype provides the data is shown in Table 2.4.1 (players' salaries) and Table 2.4.2 (teams' budgets).

Name of Players	**Current Season**	**Next Season**

Table 2.4.1 Players Salary Hoopshype Format

Name of Players	**Current Season**	**Next Season**

Table 2.4.2 Teams Budget Hoopshype Format

It is important to note that Hoopshype generates a new list for players' salaries each season, reflecting changes in player rosters. Conversely, the budget information remains consistent in terms of row count, as the NBA comprises

30 teams, a number that has remained unchanged for the years analyzed in this project. As stated on the Hoopshype website, all data is collected from publicly available NBA sources and we are allowed to use for personal usage.

Data Acquisition Challenges

One significant challenge during data acquisition was the absence of a downloadable format for the data. Hoopshype does not offer options for exporting its data; therefore, it had to be manually copied and pasted into Excel files. This process introduced initial complications, especially as the project aimed to analyze data spanning 10 years. Consequently, 10 individual Excel files were created, each containing player salary information for a single season.

To address this, Python and the panda's library were used to combine the data into a single dataset.

Player's Salaries Dataset

After cleaning and restructuring, the final players' salaries dataset includes 1,400 rows, each representing a unique player, and 11 columns, including player names (stored as strings) and salary data for 10 seasons (stored as numeric values). The final structure of the dataset is illustrated in Table 2.4.3.

Player Name	Season	Season	Season	Season	Season	Season	Season	Season	Season	Season

Table 2.4.3 Player's Salaries Table Final Structure

Teams' Budgets Dataset

The teams' budgets dataset underwent a similar acquisition and processing procedure as the players' salaries dataset. However, due to the fixed number of NBA teams, this dataset contains only 30 rows, each representing a different team, and 11 columns, including team names (strings) and budget data for 10 years (numeric values). The final structure is shown in Table 2.4.4.

Player Name	Season	Season	Season	Season	Season	Season	Season	Season	Season	Season

Table 2.4.4 Team's Budget Table Final Structure

Both the players' salaries and teams' budgets datasets share a consistent structure, with identification columns (e.g., player or team names) and corresponding values for each year. This standardized format facilitates efficient data visualization and analysis. These datasets form the foundation for the subsequent analyses conducted in this project.

Players – Teams

In addition to the datasets previously mentioned, a third dataset was required to link players to their respective teams, enabling the analysis to directly examine the relationship between these two attributes. This dataset, obtained from Basketball-Reference[5], includes players' names and the teams they played for during specific seasons. Additionally, it contains player characteristics, such as height, which could be used for further analysis to compare salaries against physical attributes.

The data collection process for this dataset was similar to the previous ones; however, in this case, the website provided a downloadable link for the data. The dataset was organized by basketball teams and seasons, detailing player information for each team during the respective seasons. While the data collection process was simpler in some respects, it introduced complexity due to changes in the analysis scope. Originally planned for 10 years, the analysis was adjusted to 5 years to account for the data collection challenges. This adjustment required analyzing 30 teams over 5 seasons, resulting in a total of 150 datasets (5 per team for the targeted seasons).

After collecting these 150 files, the next step was to merge them into a single dataset. This was accomplished using Python and the Pandas library, following the same approach as with the previous datasets.

The team member in charge of this datasets and further analysis was Marcelino Rodriguez Anglada.

3 Data Exploration

The analysis was organized into sections, with contributions from a team of four members working collaboratively on five distinct datasets. Each team member independently analyzed one dataset, with the exception of the Salaries/Budget dataset, which was treated as a single combined dataset and analyzed jointly. Each member conducted their examination independently, contributing their insights and findings to the final project. This collaborative approach ensured a comprehensive understanding of the data and led to well-rounded conclusions.

This section offers a detailed overview of the analysis conducted by each team member, emphasizing the specific datasets they examined and the insights they derived. It is important to note that this analysis required the application of tools and techniques learned in the 604 class. These included using Python and SQL queries to retrieve, link, and analyze the data.

Establishing a connection to a database was also essential for this project. The database utilized was provided by the institution and the 604 class. While login credentials varied between users, the connection process followed a consistent approach across all team members.

3.1 Coach Analysis

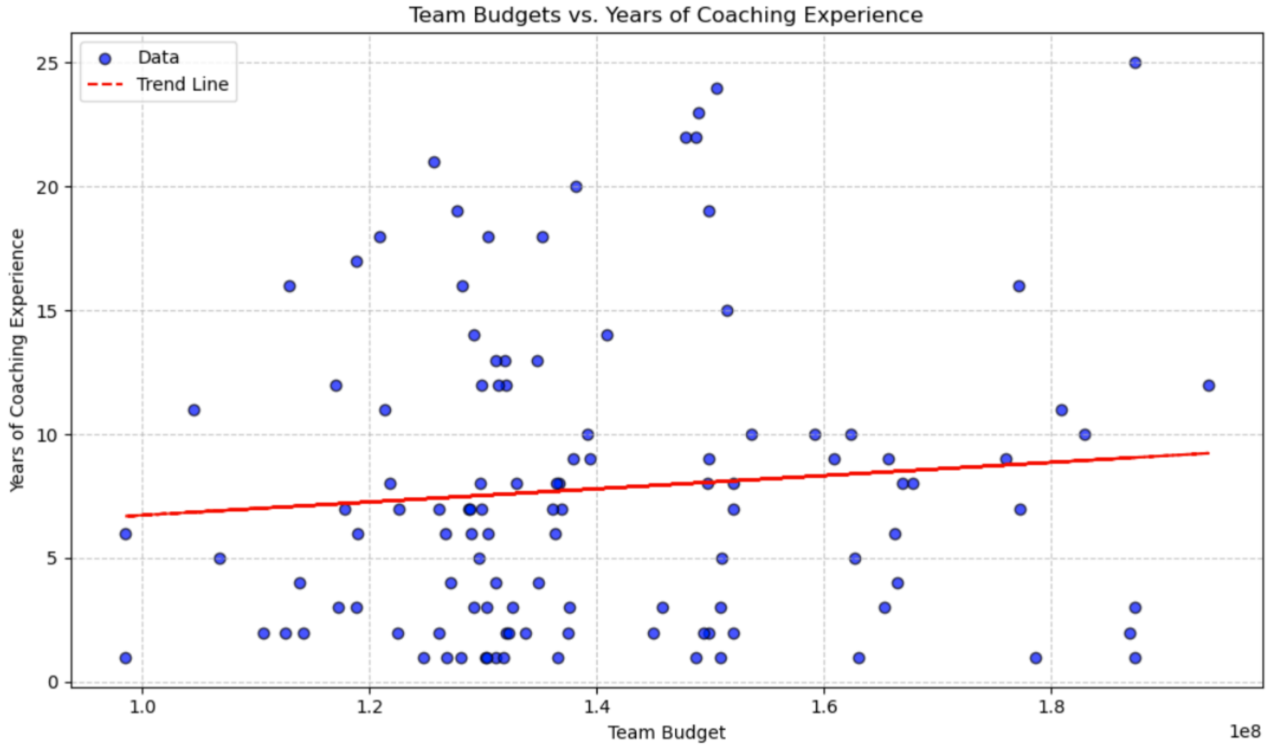
After completing the individual analysis on the coach dataset, we have found that there are no independent variables that are statistically significant for coaches. This does not specifically mean that coaches do not impact their team's overall success, however we have been unable to find any variables that associate NBA teams' success with a specific attribute of their coach. The biggest challenge with this dataset is that there are so few datapoints. There are only 30 coaches for each season where most of the coaches coach for many years before retiring or getting fired. On top of that, there are very few coaches who previously played in the NBA making it very difficult to find a way to connect our coach dataset with our other datasets.

However, we will be starting our analysis by combining the coach dataset with our budget dataset. Using the teams_all_years.csv file and combining it with the final_basketball.xlsx dataset, we are able to combine datasets to create a new column called budget where we join datasets on the team name and season year.

Utilizing an inner join, we can use *merged_df = pd.merge(coaches_df, teams_long, on=['Team', 'Year'], how='inner')* to combine the datasets. Going into this analysis, we were most interested in seeing if there was any relationship between how big of a budget a team has and the specific type of coach they choose. For example, a team with a huge budget may be able to pay higher salaries and attract the best coaches. However, as we found out in our individual analysis, it doesn't actually seem like coaches get better with a specific attribute that we explored. This was a foreshadow into the analysis between coaches and team budgets since we were unable to find any meaningful results.

A major problem we faced with combining the coach dataset with budgets and salaries was the fact that coach salaries are private. Coach salaries do not have to be reported publicly so we were unable to find any relationship between coaches' salaries and their performance since we did not have that data. It would have been exciting to investigate the impact of coach salaries on their team performance however we would need to scrape the web and utilize many different articles that state coach salaries but could be completely wrong. However, we did investigate the relationship between team budgets and the coach they select.

The first question we wanted to explore was if teams with a higher budget select coaches with more years of experience. Most people would assume that as you gain more years of experience in a field, you become better and therefore are able to demand a higher salary. However, we ran into some major issues with this analysis. It seems like teams have no preference for coaches. Both teams with high and low budgets seem to select coaches with no experience and with over 10 years of experience at the same rate. What is most interesting, is that if you remove the highest years of experience, we can see that teams with a lower budget select coaches with more years of experience than teams with a higher budget. At first this really confused our group however after some research, we found that teams with a higher budget tend to take more risks and can afford switching out their players more often. As teams switch out their key players more often, they also tend to explore different team chemistries with different coaches. That may be why you see a few of the teams with the largest budget selecting coaches with only 1-3 years of experience.

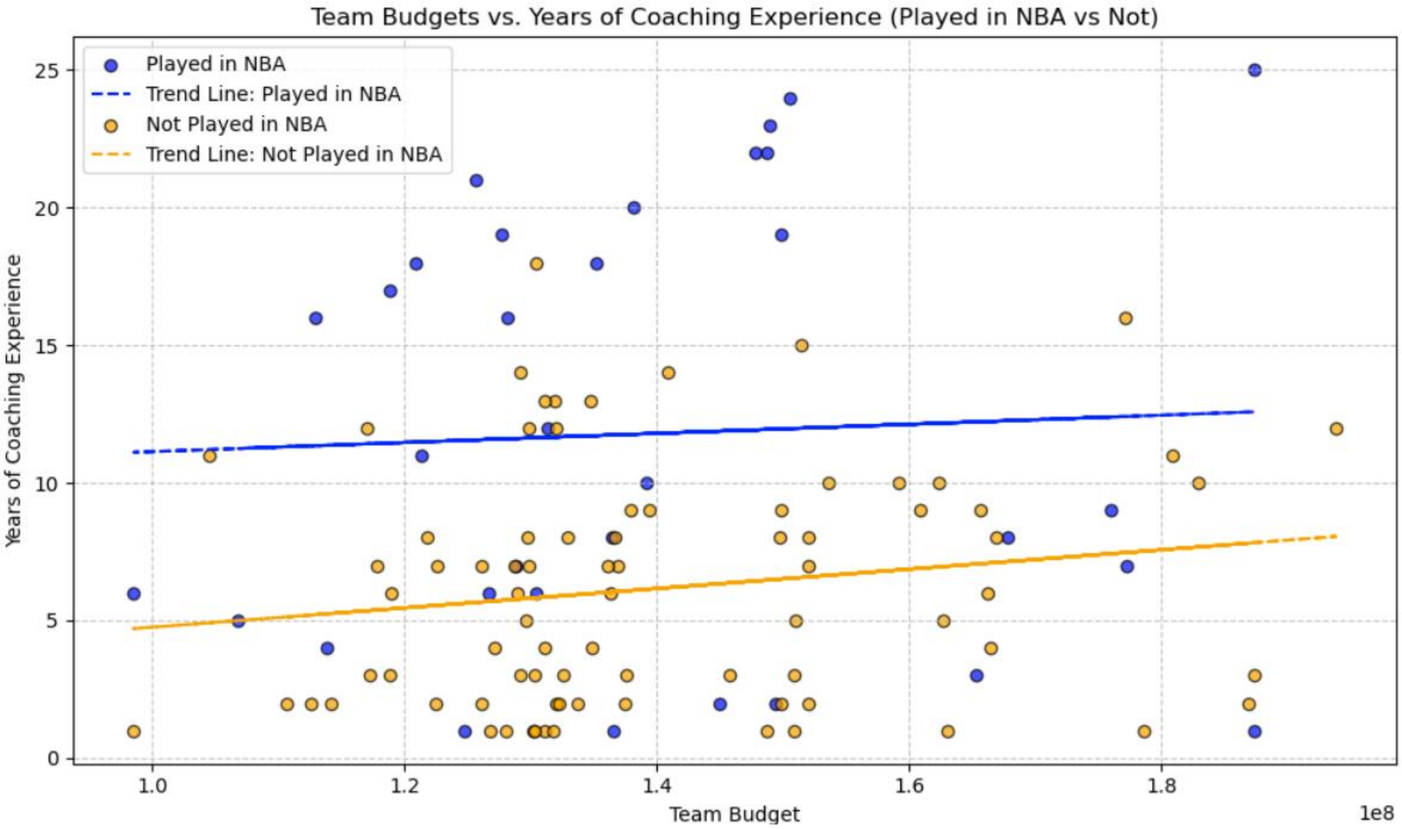


Coach Analysis: Team Budget vs Years of Experience

For our next analysis, we wanted to explore if teams preferred coaches with or without prior NBA experience. Going into this analysis, we expected that players would have a preference to be coached by players they looked up to when they were younger however they do not get to decide who coaches them. In our presentation, numerous students wanted us to explore the relationship between coaches and their players (team chemistry) however this was too difficult to track. There may be certain players who have a good relationship with their coach, but others may have a poor relationship. It also is almost impossible to track into a dataset since feelings can not be captured accurately.

To start our analysis on team preferences toward coaches with NBA experience, we had to combine the same two datasets together however split the coaches into two different groups. One for coaches with prior NBA experience and one without. After running the correlation and t-tests, we found that it had a 0.05 correlation which is incredibly low and a p-value of 0.8 which is significantly high.

We then decided to combine both years of experience and NBA experience all together to see if there were any interesting relationships.



Coach Analysis: Team Budget vs NBA Experience

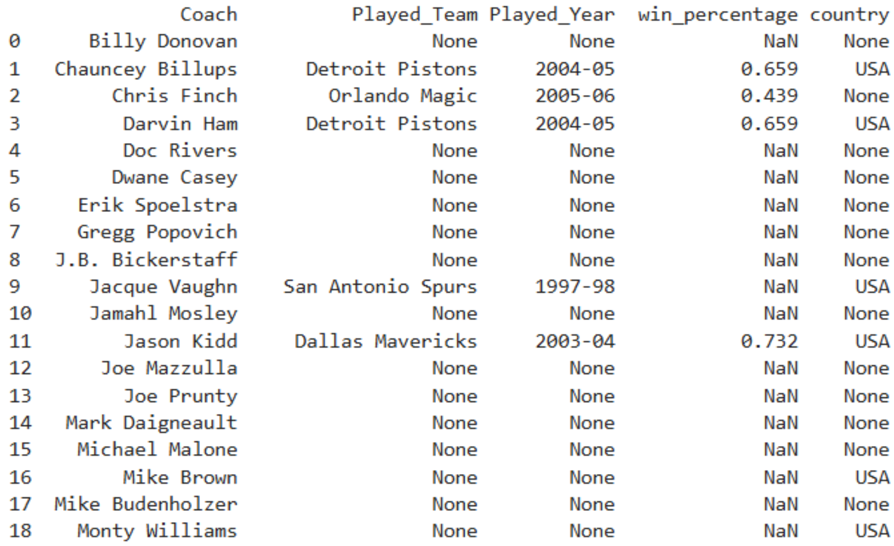
After combining all three variables, we found a very interesting relationship. Although it did not look like teams with a higher budget preferred a specific type of coach, we did find a completely unrelated relationship. It seems like coaches who previously played in the NBA, tend to stick around as coaches for longer. When you look at the difference between the yellow and blue line, it seems that on average, coaches with NBA experience have about five extra years of experience.

Let's explore this relationship a bit more. Coaches who previously played in the NBA may have already built connections with the NBA and with the media which makes it easier for them to get hired. There also may be a bias towards coaches with NBA experience and teams giving them more opportunities than coaches without NBA experience. Building off of this new relationship, we decided to combine our datasets to analyze coaches who played in the NBA.

One of the most challenging aspects of this specific analysis, was that our coach dataset did not have any obvious ways to relate to our other datasets. However, there was one specific way we could combine all of our datasets which was coaches with previous NBA experience. We have already explored the relationship between coaches and team budgets; however it is almost impossible to connect coaches with team performance and team demographics. This is mainly because coaches do not play and therefore do not contribute to any of the data captured in our other datasets. Our coach dataset already has team performance built into it; therefore, we would need to find a unique way to combine all data.

This is where a comment left on our video presentation led to a breakthrough. A student had mentioned that it would be interesting to see if coaches with previous NBA experience different success as coaches would have based on how well they performed as players. After exploring the player dataset, we were surprised to find that since our dataset captured so many seasons, there were in fact a few coaches who were also in this dataset. Therefore, we could do some analysis on those coaches who played in seasons that we have in our dataset. For the player demographic dataset, we were able to find some of the coaches on there. Unfortunately, we could not specifically relate a coach to the team performance dataset since our datasets only captured teams. This means that we had to figure out a way to capture the team that the coach played for when they were in the NBA. By doing so, we would be able to pull their old team's performance.

After many days of trying to combine our datasets together, we finally were able to produce a dataset from the coaches in the 2022-23 season. This is where we ran into our biggest problem yet. Since there is no other way to connect coaches with our other two datasets other than coaches who previously played in the NBA, we could not use the majority of our data.



```
                Coach          Played_Team Played_Year  win_percentage country
0       Billy Donovan                 None        None             NaN    None
1    Chauncey Billups      Detroit Pistons     2004-05           0.659     USA
2         Chris Finch        Orlando Magic     2005-06           0.439    None
3          Darvin Ham      Detroit Pistons     2004-05           0.659     USA
4          Doc Rivers                 None        None             NaN    None
5         Dwane Casey                 None        None             NaN    None
6      Erik Spoelstra                 None        None             NaN    None
7      Gregg Popovich                 None        None             NaN    None
8    J.B. Bickerstaff                 None        None             NaN    None
9       Jacque Vaughn    San Antonio Spurs     1997-98             NaN     USA
10      Jamahl Mosley                 None        None             NaN    None
11         Jason Kidd     Dallas Mavericks     2003-04           0.732     USA
12       Joe Mazzulla                 None        None             NaN    None
13         Joe Prunty                 None        None             NaN    None
14    Mark Daigneault                 None        None             NaN    None
15     Michael Malone                 None        None             NaN    None
16         Mike Brown                 None        None             NaN     USA
17   Mike Budenholzer                 None        None             NaN    None
18     Monty Williams                 None        None             NaN     USA
```

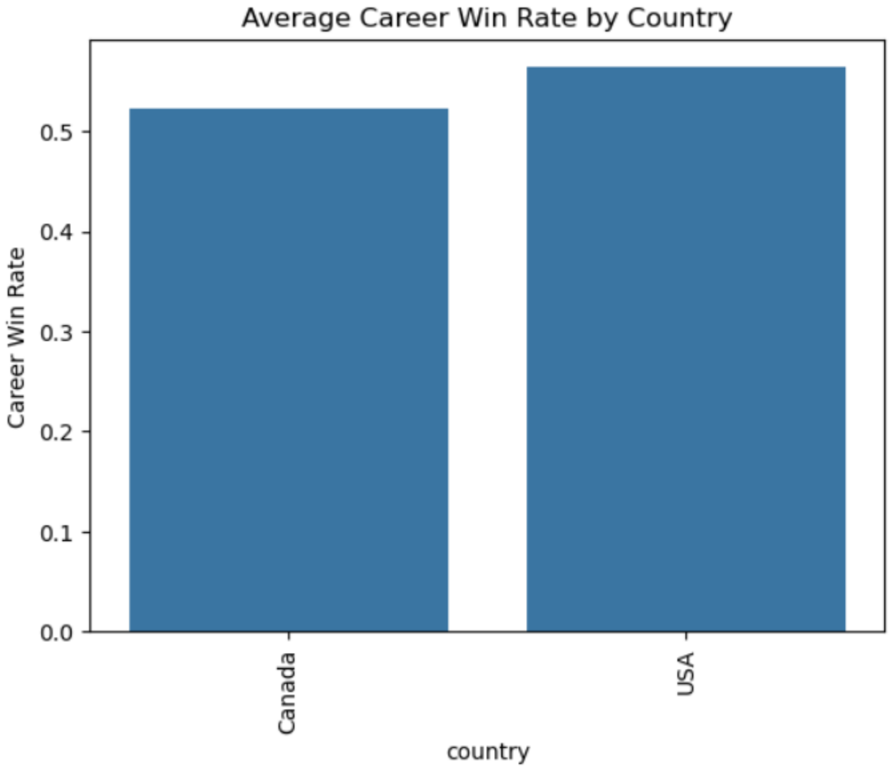
Coach Analysis: Data Issues Arise

As you can see from the final combination of data, most coaches don't actually have any data that connects them to our other datasets. This is mainly due to the fact that very few coaches have actually played in the NBA when they were younger. Out of the 32 coaches listed in our 2022-23 season, only seven coaches had previously played in the NBA with two of them playing in seasons not captured in our data. Therefore, we were only able to capture five datapoints for an entire season. On top of this, we purposely selected the 2022-23 season because it had the most datapoints out of the other seasons.

Another big issue that you can see from our dataset is that many NBA players in the 2000s came from North America. However, the majority of coaches for the most recent NBA seasons who did play in the NBA, came from these years where the NBA had so few international players. Therefore, we ran into major issues because we simply did not have enough datapoints.

Once we managed to combine all of the data together, we looked into the impact of country of origin on coach performance, and the impact of NBA experience on coach win rate. As stated before, there were so few coaches from outside North America that were coaches and played in the NBA, that we had to compare Canada to the USA.

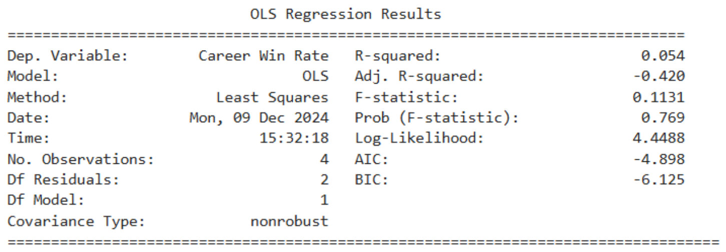
For the data that we managed to combine with all datasets, there was actually only one coach from Canada and the rest were from the USA. On top of this, the coach who came from Canada, did not even have a win percentage pulled from our other dataset.



Coach Analysis: Country of Origin vs NBA Win Rate

For this specific analysis, we only used data that we were able to combine from our other data sources. We most likely could have manually looked up each coaches country of origin, however that would defeat the hard work of finding a unique solution to combine our datasets in the first place. As you can see from the graph, it does seem like coaches from the USA have a slight advantage over Canadian coaches, however this was based on one Canadian coach.

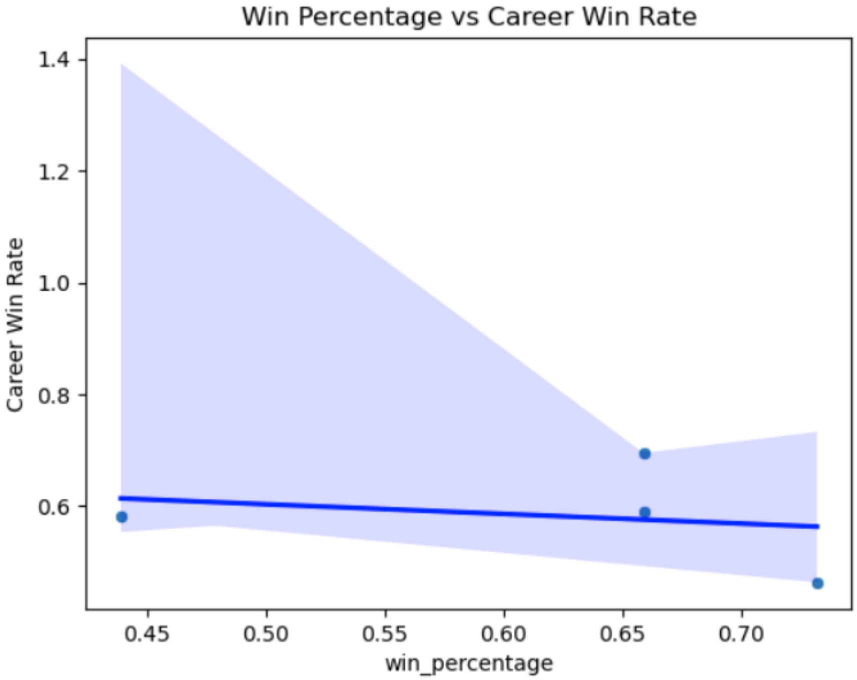
For our final analysis, we looked at coaches with NBA experience. Specifically, we wanted to figure out if coaches with NBA experience who played for good teams have a better performance as a coach than coaches with NBA experience who played for bad teams. In order to do this analysis, we needed to compare the team performance of the coaches NBA teams. For example, Billy Donovan played for the Detroit Pistons in the 2004-05 season. Therefore, we found a way to pull out the win percentage of the Detroit Pistons for the 2004-05 season and attach it to Billy Donovan. Then we ran some statistics.



```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Career Win Rate   R-squared:                       0.054
Model:                            OLS   Adj. R-squared:                 -0.420
Method:                 Least Squares   F-statistic:                    0.1131
Date:                Mon, 09 Dec 2024   Prob (F-statistic):              0.769
Time:                        15:32:18   Log-Likelihood:                 4.4488
No. Observations:                   4   AIC:                            -4.898
Df Residuals:                       2   BIC:                            -6.125
Df Model:                           1
Covariance Type:            nonrobust
==================================================================================
```

Coach Analysis: Previous NBA Experience

Unfortunately, it did not seem to matter. Coaches who played for really good teams compared to coaches who played for bad teams, had roughly the same win rates as a coach. For example, we had a coach who played for a team which had a 73% win rate which is incredible. However, as a coach he only had a 46% win rate which is very low. When looking at other coaches, we had one coach who played for a team with a 44% win rate but as a coach he had a 58% win rate.



Coach Analysis: Coaches who Played for Good Teams vs Bad Teams

As you can see from our graph, there does not seem to be any relationship. In fact, there actually seems to be a very slight negative relationship. However, due to the limitations of our data, we were not able to determine if the coaches played a key roll on the team or sat on the bench.

This was an incredibly difficult challenge. We ran into so many issues with lack of data especially since it was so difficult to find any ways to connect coaches with players. The coach dataset and the three other team datasets were so uniquely different that it was almost impossible to find any ways to connect the data. In the future, we would have preferred to pick a fourth dataset that could be more easily combined with our other datasets. However, we did find it incredibly fun to combine such different data. Despite the challenges, we uncovered a few interesting insights, such as the longer tenure of coaches with NBA experience and the limited but nuanced impact of team performance during their playing careers on their subsequent coaching success. Additionally, this analysis highlighted the significant role data limitations can play in shaping the scope and direction of an analysis. The experience reinforced how crucial it is to not only find connections within the data but also critically evaluate the quality of those connections.

Looking back, we would probably choose different datasets to work with. While it was fun and creative to try to connect such unrelated data, the limits of our datasets (especially how hard it was to link coaches to teams and players) made it tough to draw bigger conclusions. If we did this again, we would focus on datasets that are easier to combine or have more natural overlaps. We also might think about manually adding data, like tagging specific coaches, even though that wasn't part of our original plan to stick with pre-existing data. This analysis gave us lots of ideas for what to do next. For example, it would be interesting to dig into other factors that could affect coaching success, like leadership style or coaching philosophy. We could also look at assistant coaches or college coaches who moved to the NBA to get more data points. Another idea would be to study aspects outside the court, like how team ownership impacts coaching careers. On top of that, we could use machine learning to spot patterns in the data that we might not see otherwise.

3.2 NBA Demographics Analysis

The primary objective of this analysis is to examine how various demographic attributes contribute to the success of NBA teams. In today's workplace environment, including professional sports, demographics play a significant role in shaping outcomes and fostering success. The insights gained from this analysis could help teams refine their drafting strategies and build more competitive rosters.

3.2.1 Set Up Environment

For this analysis, we utilize two key datasets: the NBA Players Data and the Teams Performance dataset. These datasets are interconnected to explore the relationship between player demographics and team success.

To connect to the database provided in the DATA 604 class, the following credentials were used:

- Password: "91a4cTGQsTCIR"
- Username: "student"
- Host: "localhost"
- Database: "student"

3.2.2 Research Questions

By addressing the below questions, the analysis aims to uncover actionable insights into the impact of player demographics on team success, ultimately helping teams make data-driven decisions to improve performance.

- How does a team's average height and weight influence their scoring performance?

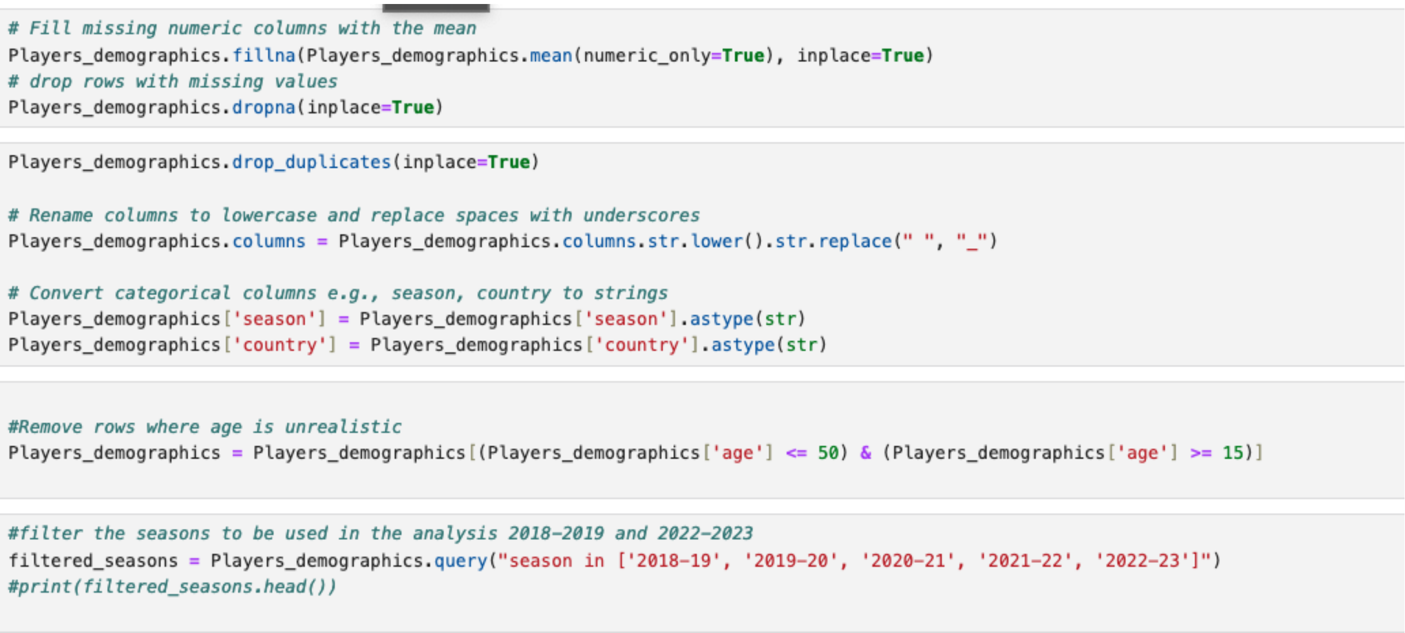
- Do teams with a higher proportion of international players achieve different performance outcomes compared to teams with predominantly domestic players?
- Do younger players perform better compared to older players?

The data cleaning process for this analysis involved several crucial steps to ensure the datasets were prepared for meaningful exploration and analysis. The primary dataset, NBA Players Data, contained demographic details of players, while the second dataset, NBA Teams Performance, provided team-level performance metrics. Cleaning these datasets required addressing inconsistencies, standardizing formats, and integrating them effectively to enable analysis.

Initially, datasets were inspected for missing, duplicate, or inconsistent data. For example, the NBA Players Data contained height and weight measurements in mixed units (e.g., feet/inches and centimeters). These were standardized to centimeters and kilograms, respectively, to ensure uniformity. Additionally, some player entries were repeated across seasons or teams due to transfers. To handle this, the data was grouped by player ID, and cumulative statistics were calculated where necessary.

The NBA Teams Performance dataset required alignment of team names and seasons with the NBA Players Data to facilitate a seamless connection between the two datasets. Outdated or misaligned team names were corrected, and irrelevant entries were removed to maintain consistency.

SQL queries played a pivotal role in joining the datasets and preparing them for analysis. The datasets were loaded into a database using the credentials provided in the DATA 604 course environment as shown above. Using SQL, inner joins were performed to link the datasets based on shared attribute such teams and season. This allowed the integration of player demographic data with team-level performance metrics. For example, a query combining the average height and weight of players on each team with their total scoring for the season enabled the exploration of the impact of physical attributes on team success.



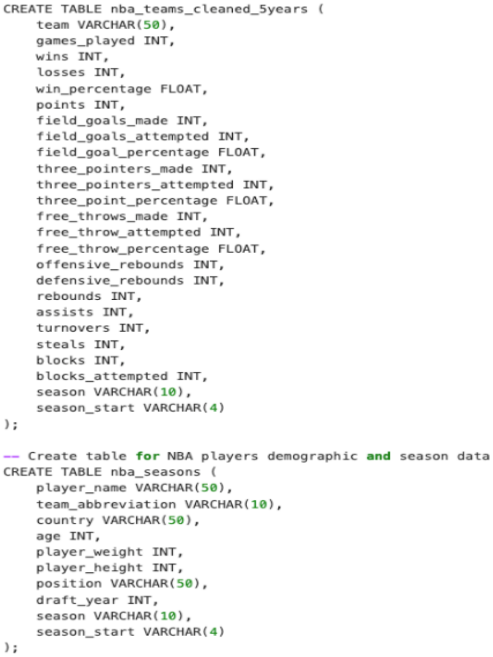
```
# Fill missing numeric columns with the mean
Players_demographics.fillna(Players_demographics.mean(numeric_only=True), inplace=True)
# drop rows with missing values
Players_demographics.dropna(inplace=True)
```

```
Players_demographics.drop_duplicates(inplace=True)

# Rename columns to lowercase and replace spaces with underscores
Players_demographics.columns = Players_demographics.columns.str.lower().str.replace(" ", "_")

# Convert categorical columns e.g., season, country to strings
Players_demographics['season'] = Players_demographics['season'].astype(str)
Players_demographics['country'] = Players_demographics['country'].astype(str)
```

```
#Remove rows where age is unrealistic
Players_demographics = Players_demographics[(Players_demographics['age'] <= 50) & (Players_demographics['age'] >= 15)]
```

```
#filter the seasons to be used in the analysis 2018-2019 and 2022-2023
filtered_seasons = Players_demographics.query("season in ['2018-19', '2019-20', '2020-21', '2021-22', '2022-23']")
#print(filtered_seasons.head())
```

Figure 3.2.1: shows data cleaning process.



```
CREATE TABLE nba_teams_cleaned_5years (
    team VARCHAR(50),
    games_played INT,
    wins INT,
    losses INT,
    win_percentage FLOAT,
    points INT,
    field_goals_made INT,
    field_goals_attempted INT,
    field_goal_percentage FLOAT,
    three_pointers_made INT,
    three_pointers_attempted INT,
    three_point_percentage FLOAT,
    free_throws_made INT,
    free_throw_attempted INT,
    free_throw_percentage FLOAT,
    offensive_rebounds INT,
    defensive_rebounds INT,
    rebounds INT,
    assists INT,
    turnovers INT,
    steals INT,
    blocks INT,
    blocks_attempted INT,
    season VARCHAR(10),
    season_start VARCHAR(4)
);

-- Create table for NBA players demographic and season data
CREATE TABLE nba_seasons (
    player_name VARCHAR(50),
    team_abbreviation VARCHAR(10),
    country VARCHAR(50),
    age INT,
    player_weight INT,
    player_height INT,
    position VARCHAR(50),
    draft_year INT,
    season VARCHAR(10),
    season_start VARCHAR(4)
);
```

Figure 3.2.2. shows SQL query for creating tables to join the two datasets

3.2.3 Insights from the NBA demographics explorations

For the first research question, we explored whether there is a correlation between a team's average player height and their total points scored. Analyzing this relationship across the top five teams from the 2018-2019 to 2022-2023 seasons, we found no significant correlation between a player's height and team performance in terms of points scored. The p-value of 0.081 suggests that the relationship between team's average height and points is not statistically significant at the typical significance level of 0.05. Additionally, the slope of the regression line indicates a weak relationship between these two variables, reinforcing the lack of a meaningful connection. In contrast, when examining the relationship between average player_weight and points scored, we found a highly significant result. The p-value of 0.000 indicates that the relationship between weight and points is statistically significant. However, the slope of the regression line reveals a negative relationship: as the average weight of players increases, the total points scored per team tends to decrease. This suggests that heavier players may not necessarily contribute as much to scoring, or that teams with higher average player weight may play a different style of game that does not prioritize scoring. See the figure below.

- Height vs. Points: p-value = 0.081, Slope = -0.04
- Weight vs. Points: p-value = 0.000, Slope = -0.05

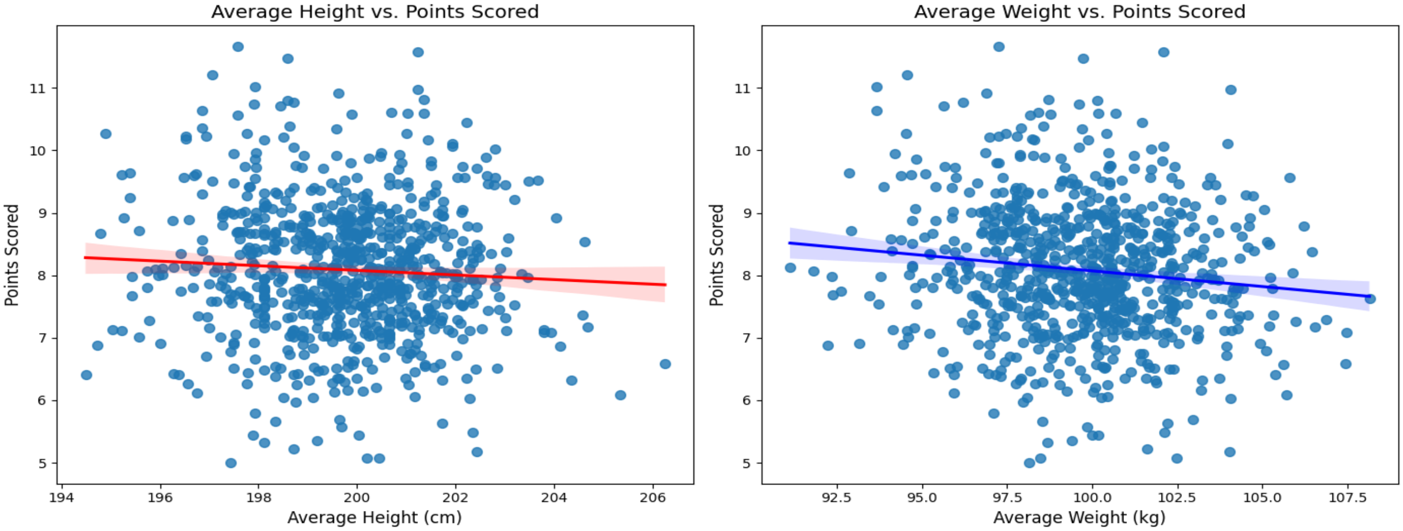


Figure 3.2.3 shows the analysis for average height and average weight vs points score.

The second question we explored was whether teams with a higher proportion of international players perform differently from those with low proportion of international. To investigate this, we combined the NBA Teams Performance dataset to get the win percentages with the NBA Demographics dataset and used player's country to categorize them as either international or domestic (U.S. players). After conducting the analysis, we found that teams with a higher proportion of international player specifically those with more than 50% international players tended to have a higher win percentage. This suggests that international diversity may have a positive impact on a team's overall performance. The following figure illustrates this finding.

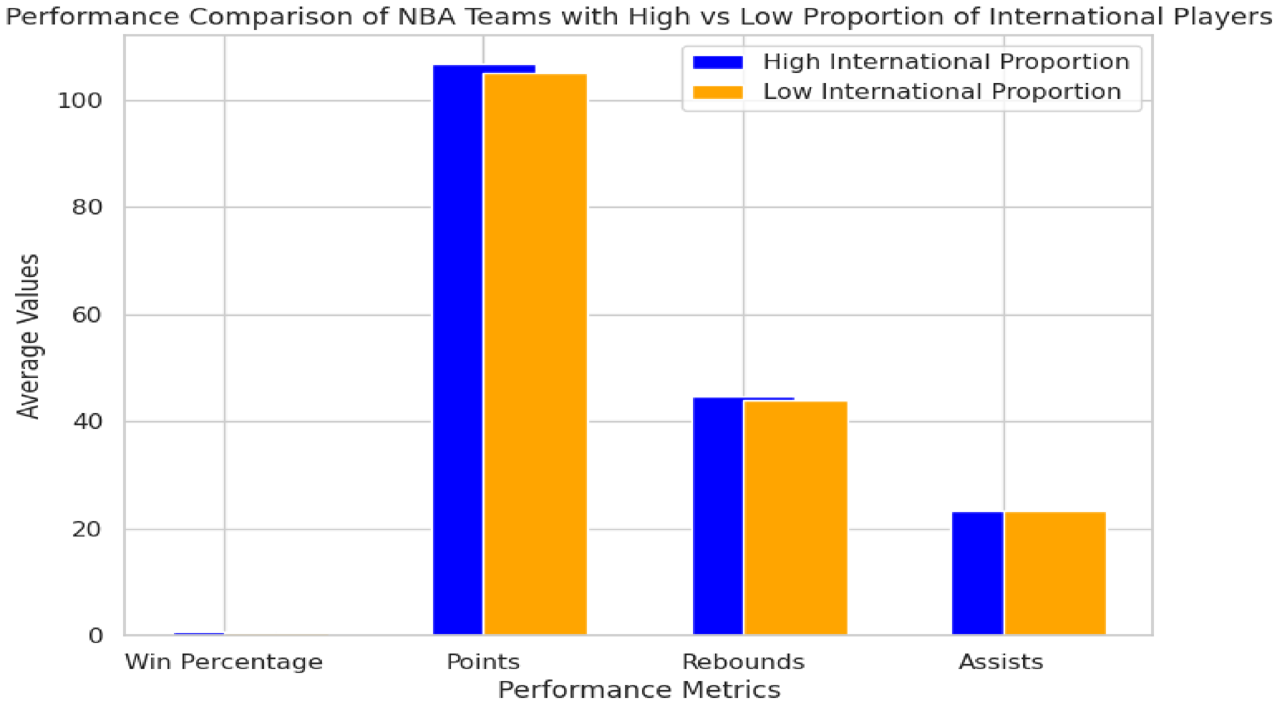


Figure 3.2.4 illustrates High vs low proportions of international players in the NBA teams.

The final research question explored in the NBA demographics analysis focused on age: Do older players perform better compared to younger players? To address this, we defined "younger" and "older" players by comparing their ages to the median age of the dataset. We then calculated the average performance metrics, including points, rebounds, and assists, for each group. A t-test was conducted to assess statistical significance. The results revealed a significant difference in performance between the two age groups, with a p-value of 0.000 for all three metrics—points, rebounds, and assists. This indicates that the performance differences between younger and older players are highly statistically significant. The visualization below illustrates these findings.

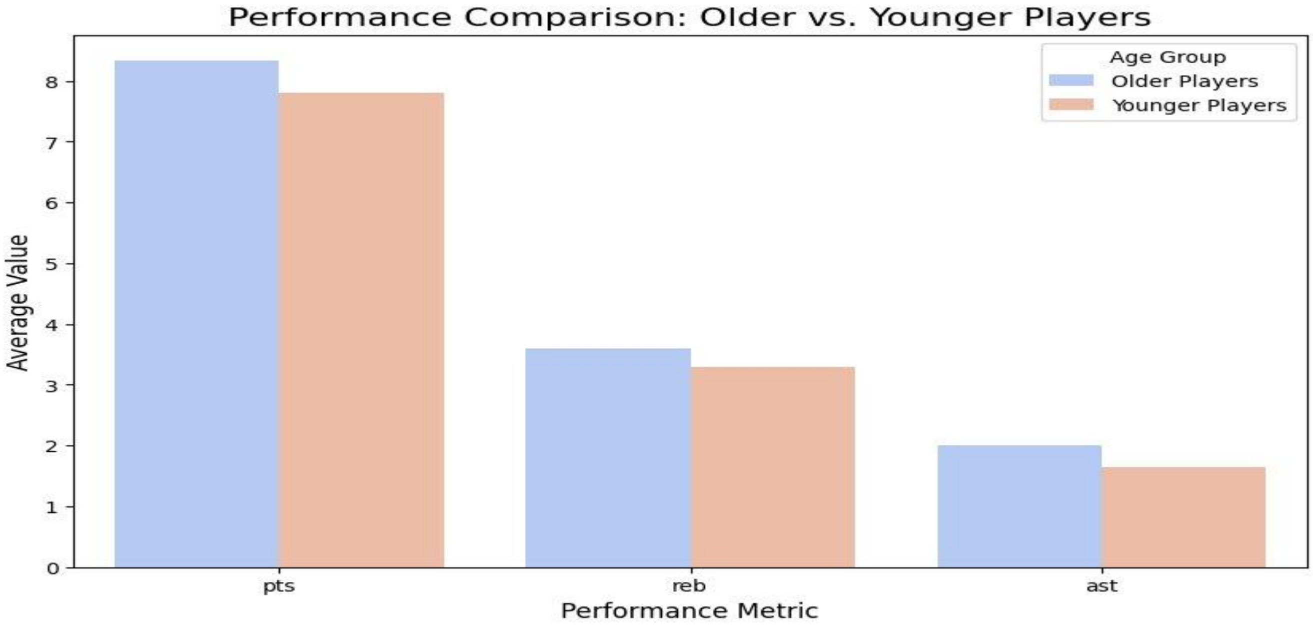
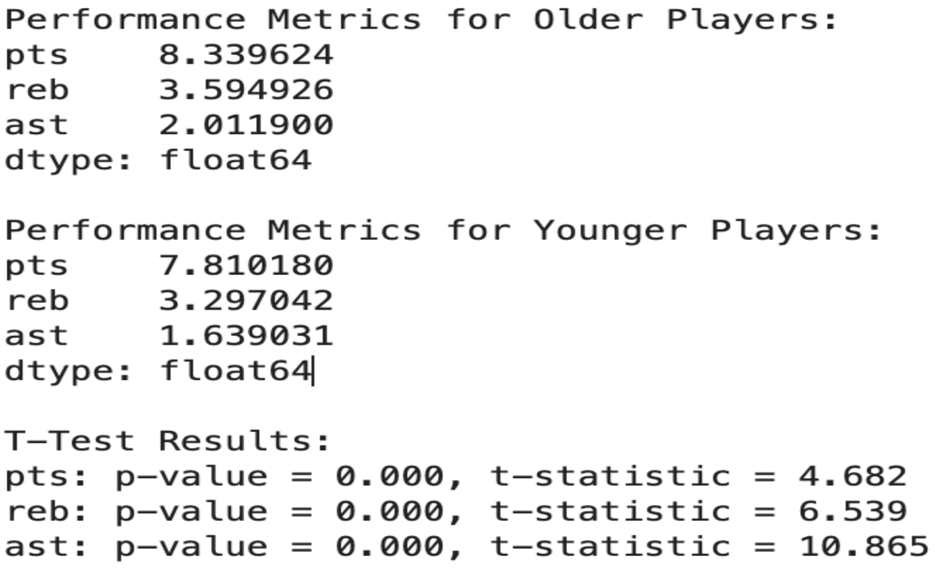


Figure 3.2.5 shows performance comparison between younger players vs older players.



```
Performance Metrics for Older Players:
pts    8.339624
reb    3.594926
ast    2.011900
dtype: float64

Performance Metrics for Younger Players:
pts    7.810180
reb    3.297042
ast    1.639031
dtype: float64

T-Test Results:
pts: p-value = 0.000, t-statistic = 4.682
reb: p-value = 0.000, t-statistic = 6.539
ast: p-value = 0.000, t-statistic = 10.865
```

Figure 3.2.6 shows performance metrics for older vs younger

3.2 Teams Performance Analysis

The objective is to analyze how game performance metrics influence team success in the NBA. This involves examining relationships between key performance indicators, such as points scored, field goal percentages,

turnovers, rebounds, assists, and team outcomes like win percentage and playoff qualifications. By identifying patterns and correlations, the analysis aims to uncover which factors most significantly contribute to a team's success. Insights from this study can help teams optimize strategies, enhance player performance, and improve overall outcomes in future seasons.

3.2.1 Set up environment

The first step in this analysis is to prepare the environment by importing the necessary libraries and loading the datasets. The datasets used in this analysis are:

- Team_Performnace
- Player_Demographics

To connect to the database provided in the 604 class, the following credentials were required:

- Password: " t7ByP6EkdSDOj"
- Username: "student"
- Host: "localhost"
- Database: "student"

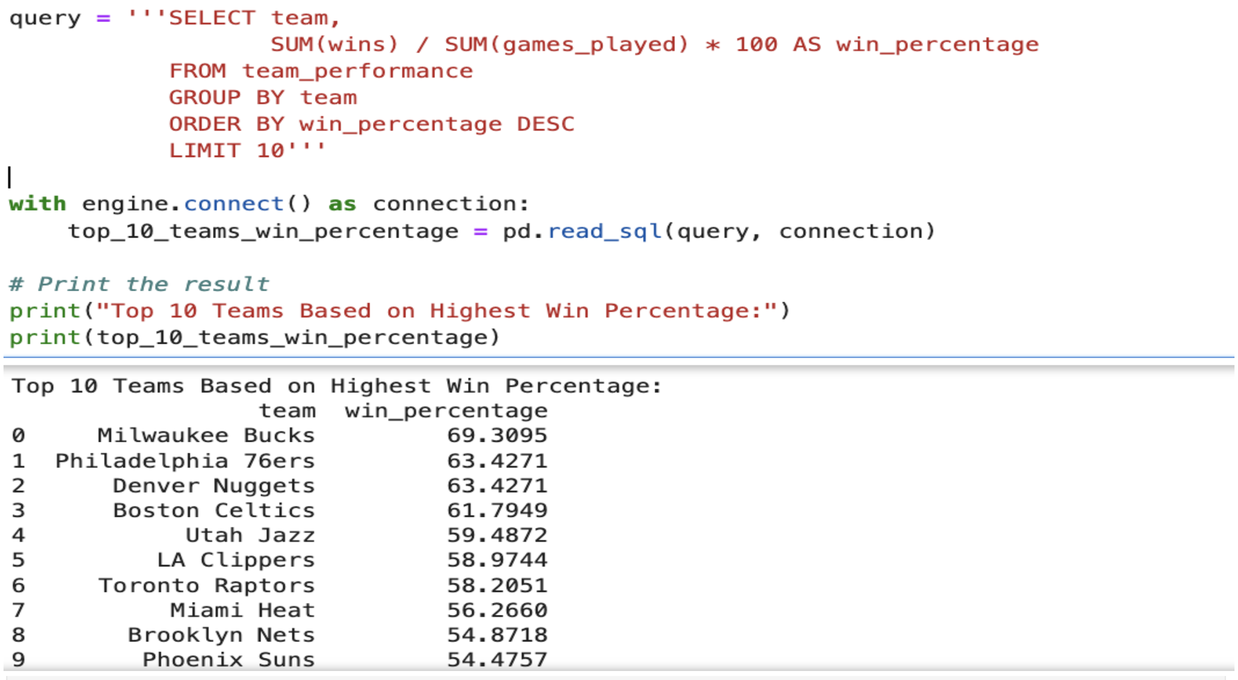
Once the environment was set up, these the above datasets were utilized to perform the analysis effectively and to answer the below research questions.

3.2.2 Research Questions

These research questions aim to uncover critical factors that drive success in the NBA and offer actionable insights for teams and players.

- Which team has the highest overall shooting efficiency based on field goal percentage, three-point percentage, and free throw percentage?
- How does a team’s win percentage correlate with its field goal percentage and three-point shooting efficiency?
- How does team performance correlate with individual player statistics?

There are 30 teams in our dataset. Finding the Top 10 teams based on their win percentage is essential to identify high performers and understand the factors driving success. It highlights teams that excel consistently, offering insights into key performance metrics like shooting accuracy or defensive strength. These teams serve as benchmarks, setting standards for others and enabling comparisons across seasons to detect trends in success. The analysis engages fans and stakeholders by showcasing dominant teams and provides strategic insights for improving team performance. It also helps management assess the effectiveness of investments in players, coaching, or facilities. Additionally, understanding the top team’s aids in contextualizing the league’s competitive landscape and planning match strategies effectively.



```
query = '''SELECT team,
                  SUM(wins) / SUM(games_played) * 100 AS win_percentage
           FROM team_performance
           GROUP BY team
           ORDER BY win_percentage DESC
           LIMIT 10'''

with engine.connect() as connection:
    top_10_teams_win_percentage = pd.read_sql(query, connection)

# Print the result
print("Top 10 Teams Based on Highest Win Percentage:")
print(top_10_teams_win_percentage)
```

```
Top 10 Teams Based on Highest Win Percentage:
                 team  win_percentage
0     Milwaukee Bucks         69.3095
1  Philadelphia 76ers         63.4271
2      Denver Nuggets         63.4271
3      Boston Celtics         61.7949
4           Utah Jazz         59.4872
5         LA Clippers         58.9744
6     Toronto Raptors         58.2051
7          Miami Heat         56.2660
8       Brooklyn Nets         54.8718
9        Phoenix Suns         54.4757
```

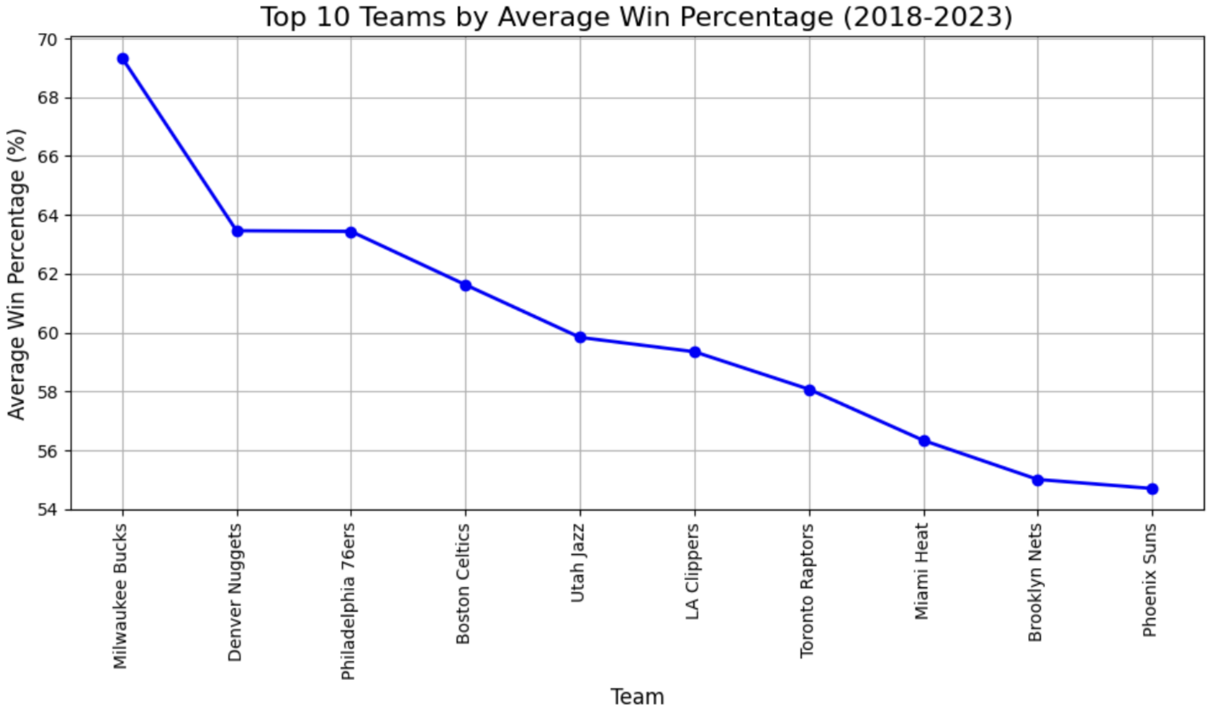


Figure 3.3.1: Top 10 NBA teams by average win percentage.

The above plot Figure 3.3.1 shows the Top 10 NBA teams by average win percentage from 2018 to 2023. The Milwaukee Bucks lead with nearly 70%, reflecting consistent dominance, followed by the Denver Nuggets with over 65%. The 76ers, Celtics, and Jazz maintain competitive performances around 62%-64%. Teams like the LA Clippers and Raptors perform steadily above 60%, while the Heat, Nets, and Suns complete the list with percentages between 54%-58%. The gradual decline highlights a clear performance gap between the top and bottom of the list.

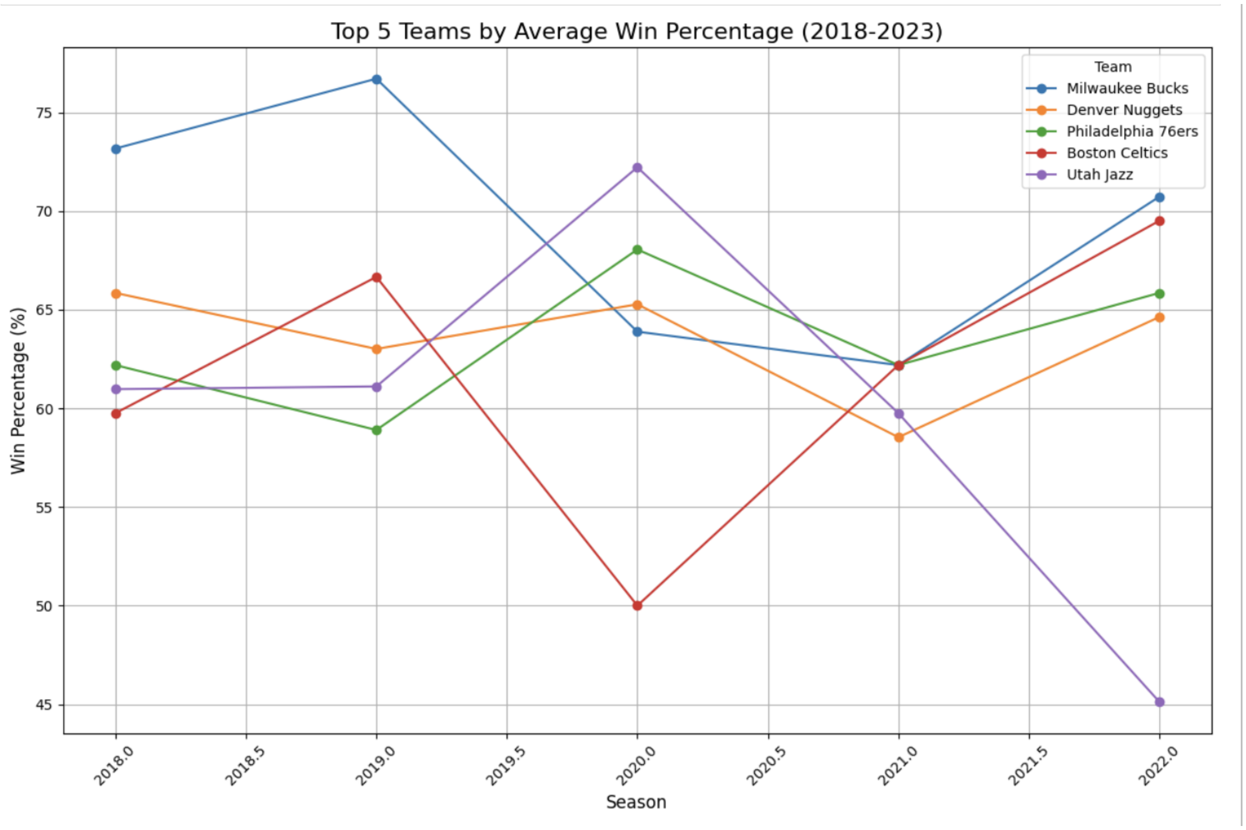
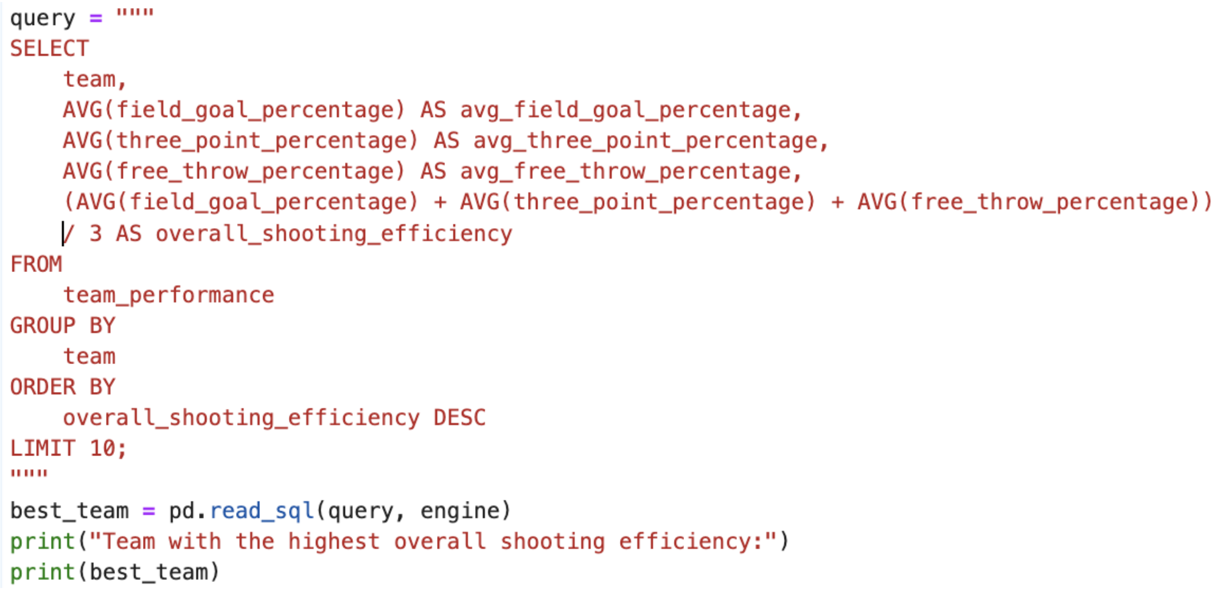


Figure 3.3.2: win percentage trends for the Top 5 NBA teams.

The above plot Figure 3.3.2 shows win percentage trends for the Top 5 NBA teams (2018–2023). The Milwaukee Bucks consistently performed well, peaking around 2020, while the Denver Nuggets improved steadily toward the end of the timeline. The Philadelphia 76ers showed stable performance with minor fluctuations. The Boston Celtics experienced significant variability, including a sharp drop in 2020, followed by recovery. Utah Jazz performed strongly initially but declined sharply after 2021. The 2020 season stands out for variability across teams, possibly due to the pandemic's impact on the league.

1. **Which team has the highest overall shooting efficiency based on field goal percentage, three-point percentage, and free throw percentage?**

Analyzing which team has the highest overall shooting efficiency based on field goal percentage, three-point percentage, and free throw percentage is essential for evaluating offensive strength. Shooting efficiency directly impacts a team's ability to score and win games, making it a critical performance metric. By including all aspects of shooting, it provides a holistic view of a team's scoring capabilities. This analysis helps teams refine offensive strategies, such as focusing on three-point accuracy or improving free throw performance. It also offers insights into player development, as high efficiency often reflects well-trained players. Benchmarking shooting efficiency across teams highlights leaders and inspires improvement. Additionally, it engages fans and stakeholders by showcasing which teams excel offensively. Overall, it provides valuable insights into how scoring efficiency influences team success.



```
query = """
SELECT
    team,
    AVG(field_goal_percentage) AS avg_field_goal_percentage,
    AVG(three_point_percentage) AS avg_three_point_percentage,
    AVG(free_throw_percentage) AS avg_free_throw_percentage,
    (AVG(field_goal_percentage) + AVG(three_point_percentage) + AVG(free_throw_percentage))
    / 3 AS overall_shooting_efficiency
FROM
    team_performance
GROUP BY
    team
ORDER BY
    overall_shooting_efficiency DESC
LIMIT 10;
"""
best_team = pd.read_sql(query, engine)
print("Team with the highest overall shooting efficiency:")
print(best_team)
```

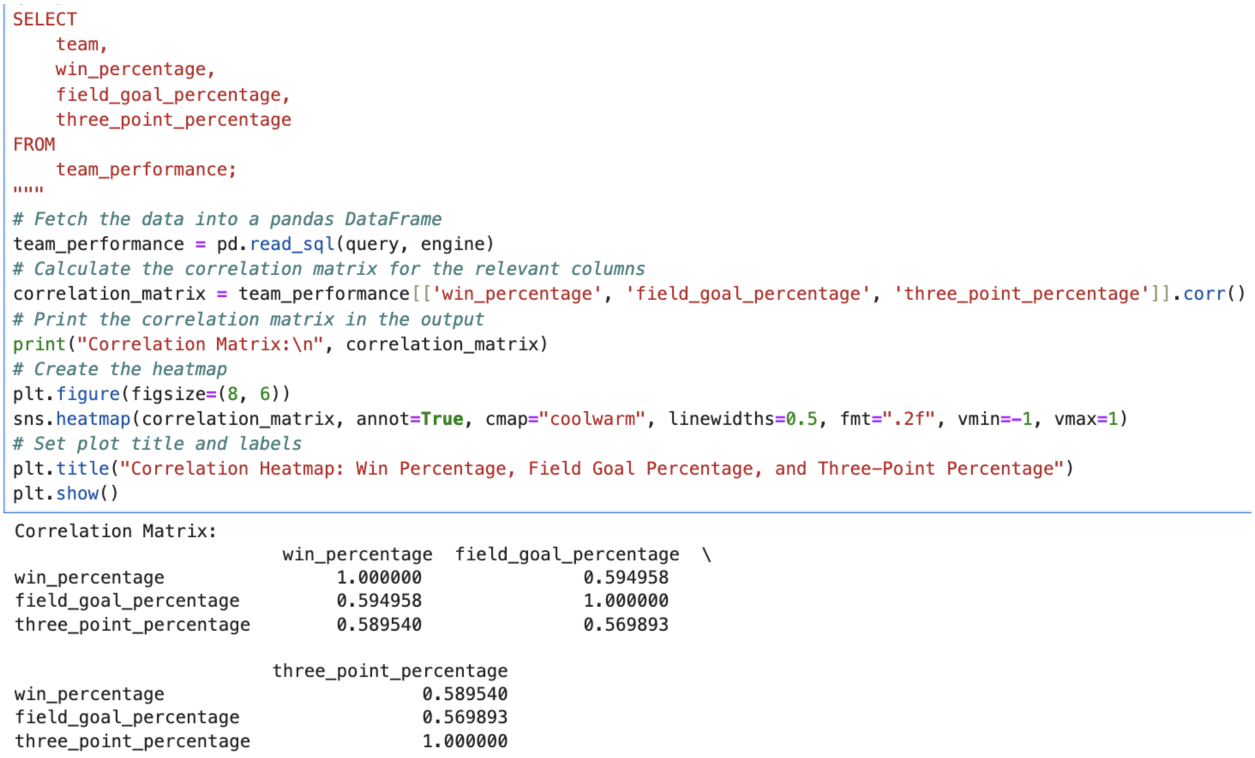
Table 3.3.1: NBA teams' overall shooting efficiency

Teams	Avg Field Goal Percentage	Avg Three-Point Percentage	Avg Free Throw Percentage	Overall Shooting Efficiency
LA Clippers	47.08	38.5	79.92	55.17
Phoenix Suns	47.38	36.06	80.74	54.73
Boston Celtics	46.66	36.72	80.12	54.5
Philadelphia 76ers	47.36	37.04	78.98	54.46
Golden State Warriors	46.9	36.88	79.04	54.27
Portland Trail Blazers	45.98	36.64	79.94	54.19
Denver Nuggets	48.22	36.38	77.62	54.07
Chicago Bulls	46.92	35.98	79.02	53.97
Atlanta Hawks	46.42	35.68	79.68	53.93
Brooklyn Nets	47.06	36.54	77.98	53.86

From the above Table 3.3.1, The **LA Clippers** lead with the highest overall shooting efficiency (55.17%), reflecting balanced shooting performance. The **Denver Nuggets** have the highest field goal percentage (48.22%) but rank lower overall due to weaker three-point and free throw percentages. Teams like the **Boston Celtics** and **Phoenix Suns** maintain balanced scoring efficiency across metrics. The **Brooklyn Nets** rank lowest in overall efficiency (53.86%), despite a strong free throw percentage. This data highlights team scoring effectiveness and provides a basis for performance comparisons.

2. **How does a team's win percentage correlate with its field goal percentage and three-point shooting efficiency?**

Analyzing the correlation between a team's win percentage and its field goal and three-point shooting efficiency highlights how scoring accuracy impacts success. It helps teams identify key metrics, refine offensive strategies, and benchmark their performance against high-performing teams. Understanding these relationships enables targeted improvements, such as focusing on three-point accuracy or shooting consistency, to maximize wins. This analysis also provides long-term insights into the evolving importance of shooting efficiency in basketball and helps allocate resources effectively for player development and recruitment.



```
SELECT
    team,
    win_percentage,
    field_goal_percentage,
    three_point_percentage
FROM
    team_performance;
"""
# Fetch the data into a pandas DataFrame
team_performance = pd.read_sql(query, engine)
# Calculate the correlation matrix for the relevant columns
correlation_matrix = team_performance[['win_percentage', 'field_goal_percentage', 'three_point_percentage']].corr()
# Print the correlation matrix in the output
print("Correlation Matrix:\n", correlation_matrix)
# Create the heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm", linewidths=0.5, fmt=".2f", vmin=-1, vmax=1)
# Set plot title and labels
plt.title("Correlation Heatmap: Win Percentage, Field Goal Percentage, and Three-Point Percentage")
plt.show()
```

```
Correlation Matrix:
                         win_percentage  field_goal_percentage  \
win_percentage                1.000000               0.594958
field_goal_percentage         0.594958               1.000000
three_point_percentage        0.589540               0.569893

                        three_point_percentage
win_percentage                        0.589540
field_goal_percentage                 0.569893
three_point_percentage                1.000000
```

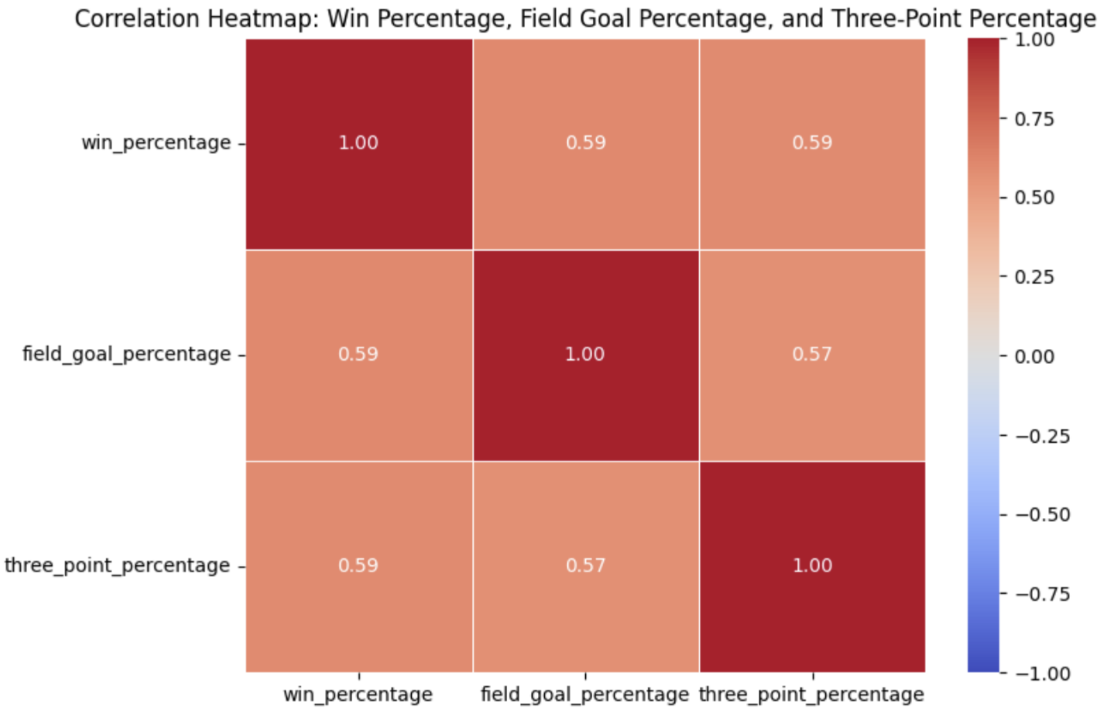


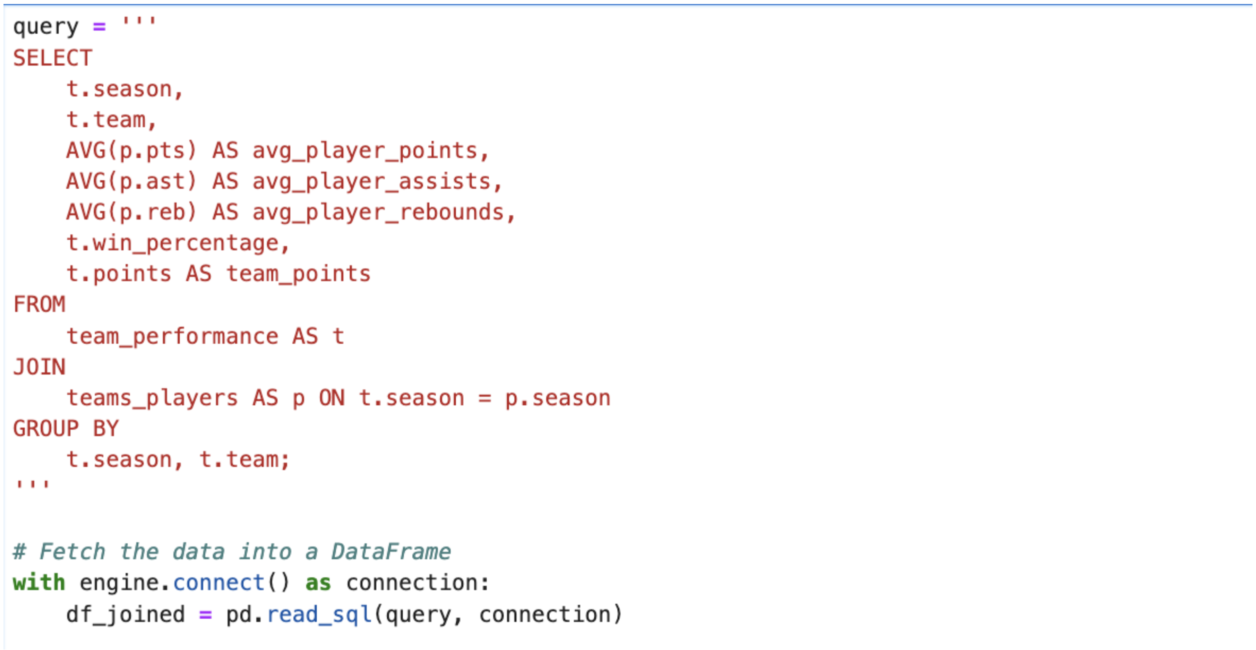
Figure 3.3.3 Co-relation Matrix of Team's win percentage with field goal percentage and three-point shooting efficiency

This heatmap Figure 3.3.3 shows the correlation between win percentage, field goal percentage, and three-point percentage for NBA teams. The correlation between **win percentage and field goal percentage** is **0.59**, indicating a moderate positive relationship, meaning better overall shooting accuracy leads to more wins. Similarly, the

correlation between win percentage and three-point percentage is also **0.59**, showing three-point efficiency significantly impacts success. The **correlation between field goal percentage and three-point percentage** is slightly lower at **0.57**, reflecting their partial dependence. These insights highlight that shooting efficiency is critical for winning games, though other factors like defense and turnovers also play a role.

3. How does team performance correlate with individual player statistics?

Analyzing correlations between team performance metrics (e.g., win percentage) and player stats (e.g., points, rebounds, assists) helps identify key drivers of success. It enables teams to optimize strategies, assign roles, and build balanced rosters by focusing on impactful contributions. The insights guide resource allocation and recruiting efforts, ensuring investments align with performance goals. This analysis also quantifies individual player impact on team outcomes and supports long-term planning for sustained success.



```
query = '''
SELECT
    t.season,
    t.team,
    AVG(p.pts) AS avg_player_points,
    AVG(p.ast) AS avg_player_assists,
    AVG(p.reb) AS avg_player_rebounds,
    t.win_percentage,
    t.points AS team_points
FROM
    team_performance AS t
JOIN
    teams_players AS p ON t.season = p.season
GROUP BY
    t.season, t.team;
'''

# Fetch the data into a DataFrame
with engine.connect() as connection:
    df_joined = pd.read_sql(query, connection)
```

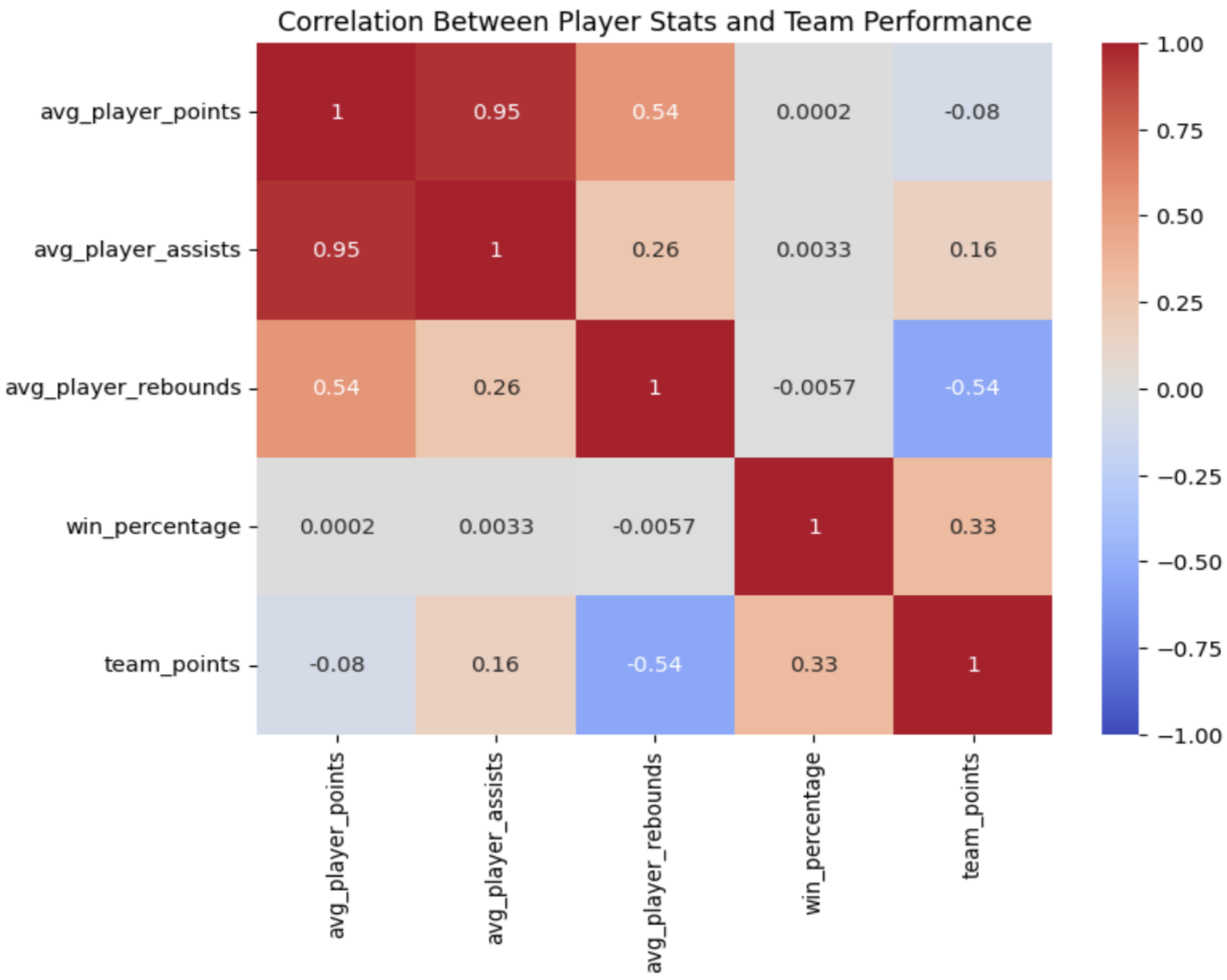


Figure 3.3.4: Correlation matrix between player statistics and team performance metrics.

This heatmap Figure 3.3.4 shows the correlations between player statistics (average points, assists, rebounds), team performance metrics (win percentage), and team points. There is a very strong positive correlation (**0.95**) between player points and assists, indicating balanced offensive capabilities. Player rebounds have moderate correlations with points (**0.54**) and assists (**0.26**), suggesting some overlap in roles. Win percentage shows very weak correlations with individual player stats, such as points (**0.0002**), assists (**0.0033**), and rebounds (**-0.0057**), indicating team success depends more on collective efforts. Team points have a moderate positive correlation with win percentage (**0.33**), suggesting higher-scoring teams tend to win more, though other factors like defense matter. Interestingly, rebounds have a negative correlation with team points (**-0.54**), possibly reflecting missed shots rather than scoring efficiency. Overall, the analysis highlights the limited direct impact of individual stats on win percentage, emphasizing teamwork and strategy.

3.3 Salary – Budget Analysis

The objective of this analysis is to examine whether there is a relationship between player salaries and team budgets. Specifically, it seeks to determine if a team's budget influences the hiring of players. The scope of this analysis focuses on the top 5 player salaries and the top 5 team budgets over a period of 5 years or 5 seasons.

3.3.1 Set up environment.

The first step in this analysis is to prepare the environment by importing the necessary libraries and loading the datasets. The datasets used in this analysis are:

- players_all_years: Contains player salaries for the required years.
- teams_all_years: Contains team budgets for the required years.
- Players_Teams_Data: Establishes the relationship between teams and players.

The key libraries imported for this analysis include:

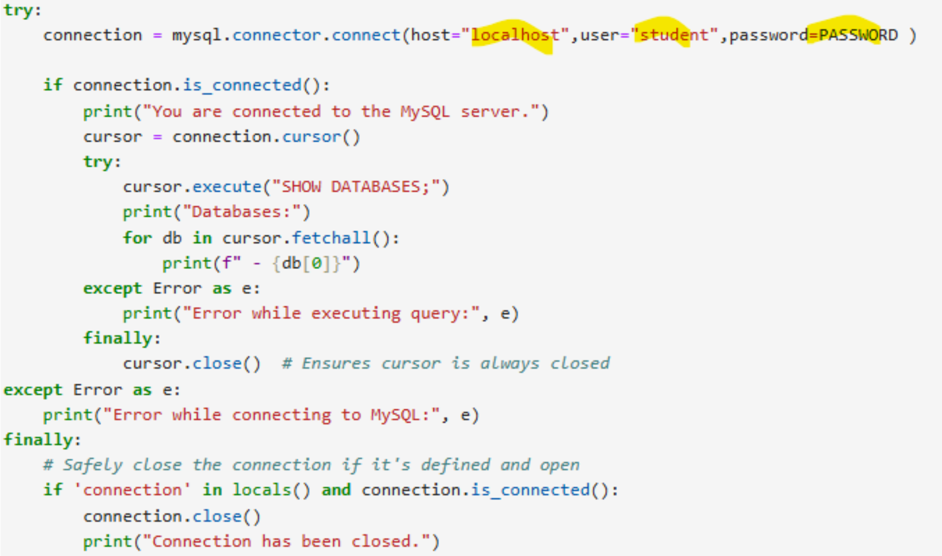
- pandas: For data manipulation and analysis in Python.
- csv: For handling CSV file formats.
- mysql.connector: For establishing database connections.
- sqlalchemy: For performing database operations.
- matplotlib.pyplot: For creating visualizations and graphs.

To connect to the database provided in the 604 class, the following credentials were required:

- Password: "vxMTcnlHATeIL"
- Username: "student"
- Host: "localhost"
- Database: "student"

Each team member was assigned a unique password to ensure secure access. Once the environment was set up, these tools and datasets were utilized to perform the analysis effectively.

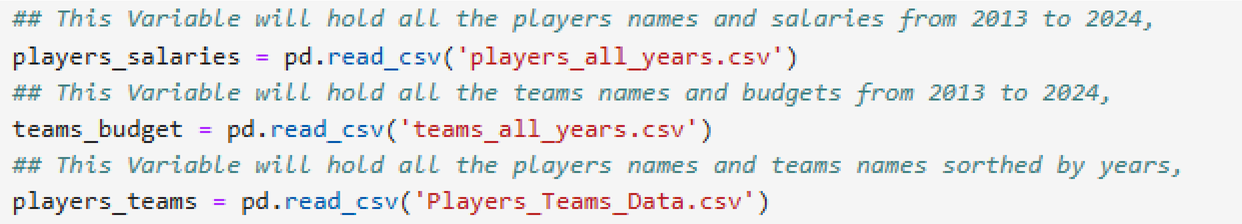
Because the analysis required working with a data base it is necessary to create or stablish the connection to it, a successful connection to the data base indicates that it is possible to run SQL queries to further explore the data. Figure 3.4.1 shows a code the snip code used to connect to the data base in addition to that shows hot to close the connection to the database.



```
try:
    connection = mysql.connector.connect(host="localhost",user="student",password=PASSWORD )

    if connection.is_connected():
        print("You are connected to the MySQL server.")
        cursor = connection.cursor()
        try:
            cursor.execute("SHOW DATABASES;")
            print("Databases:")
            for db in cursor.fetchall():
                print(f" - {db[0]}")
        except Error as e:
            print("Error while executing query:", e)
        finally:
            cursor.close()  # Ensures cursor is always closed
except Error as e:
    print("Error while connecting to MySQL:", e)
finally:
    # Safely close the connection if it's defined and open
    if 'connection' in locals() and connection.is_connected():
        connection.close()
        print("Connection has been closed.")
```

Figure 3.4.1 Database Connection

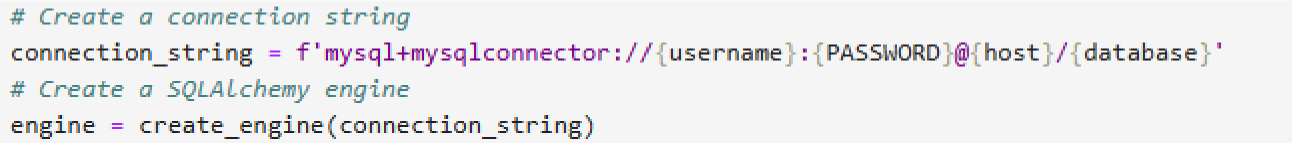
After confirming that the database connection was working properly, the load of the dataset was the next step. Figure 3.4.2 shows the procedure how to load the csv files into the python environment and saved into variables for easy data manipulation.



```
## This Variable will hold all the players names and salaries from 2013 to 2024,
players_salaries = pd.read_csv('players_all_years.csv')
## This Variable will hold all the teams names and budgets from 2013 to 2024,
teams_budget = pd.read_csv('teams_all_years.csv')
## This Variable will hold all the players names and teams names sorthed by years,
players_teams = pd.read_csv('Players_Teams_Data.csv')
```

Figure 3.4.2 Load CSV datasets

The connection to the database has been confirmed from the previous code. However, there are multiple ways to connect to the database. Figure 3.4.3 shows a different approach to create a connection with the database. This is the procedure that its going to be used in this analysis the reason is the simplicity of the code.



```
# Create a connection string
connection_string = f'mysql+mysqlconnector://{username}:{PASSWORD}@{host}/{database}'
# Create a SQLAlchemy engine
engine = create_engine(connection_string)
```

Figure 3.4.3 Database Connection Second Method

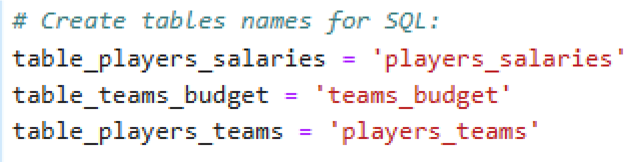
With the connection created and the dataset loaded to python environment, the next step is to start importing the datasets into the database. This means creating the respective tables that will hold the data. There are different approaches to achieve this. For this section, the first step is to set the names of the tables to build. The names of the tables and their respective datasets are listed in the table below:

Name of Data set	**Name of the Table in the database**	**Name of the variable in python**
players_all_years	'players_salaries'	table_players_salaries
teams_all_years	'teams_budget'	table_teams_budget

Players_Teams_Data	'players_teams'	table_players_teams

Table 3.4.1 Database Table Names Reference

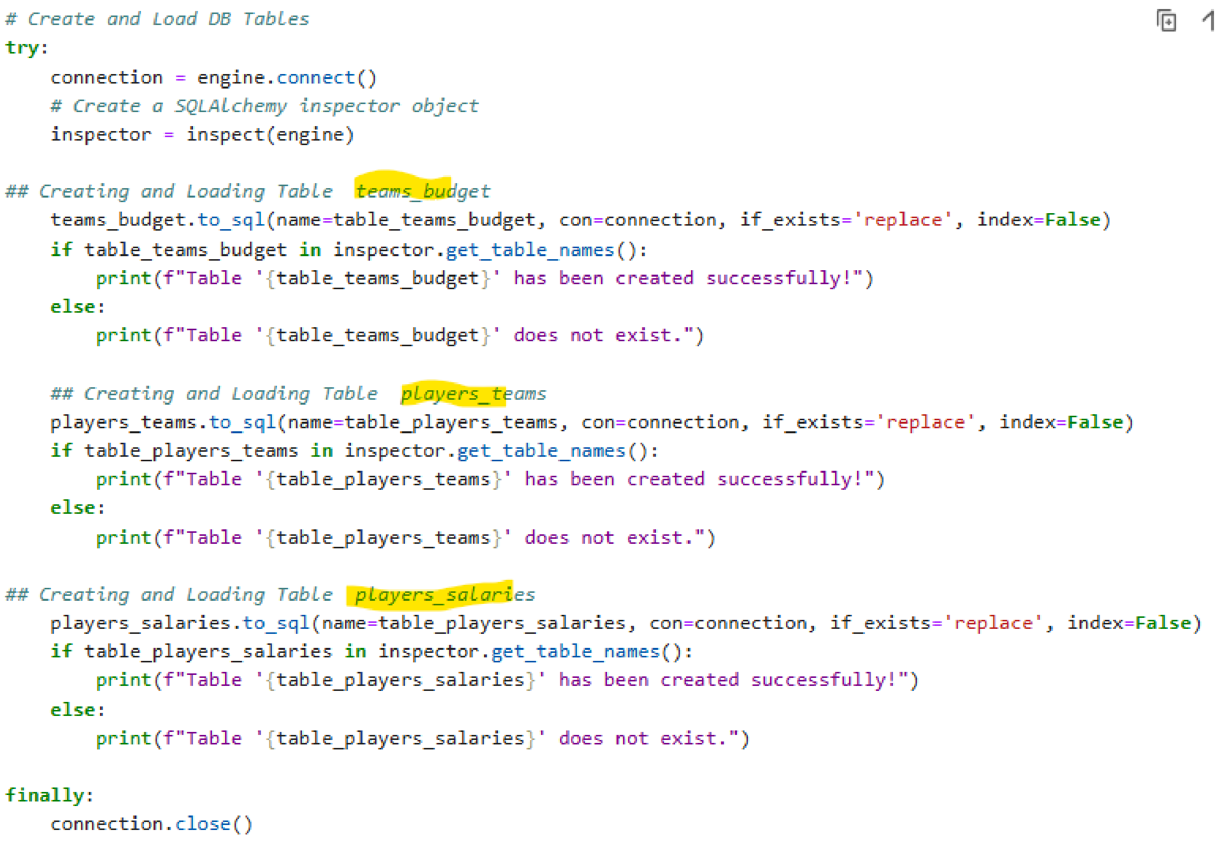
The table 3.4.1 shows the relationship between the name of the dataset used and the corresponding table name in the database. Additionally, the name or string of each table in the database is saved in a variable in the Python environment to allow for better and easier access from within Python. Figure 3.4.4 shows the creation of the names of the SQL tables and being saved in their respective python variables.



```
# Create tables names for SQL:
table_players_salaries = 'players_salaries'
table_teams_budget = 'teams_budget'
table_players_teams = 'players_teams'
```

Figure 3.4 4 SQL Table Names

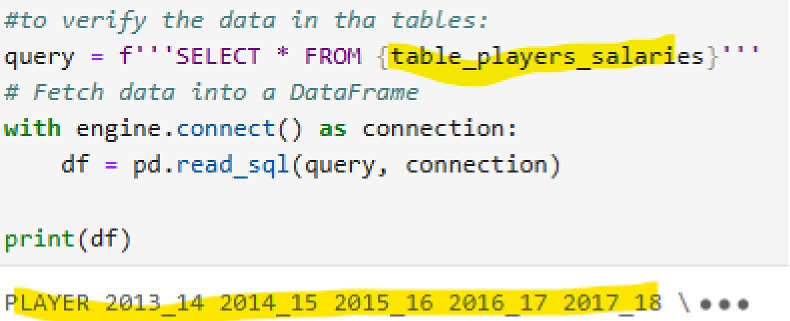
From that it is possible to start creating the tables and load the datasets to those tables following the format/ structure that the datasets hold. Figure 3.4.5 shows the procedure.



```
# Create and Load DB Tables
try:
    connection = engine.connect()
    # Create a SQLAlchemy inspector object
    inspector = inspect(engine)

## Creating and Loading Table  teams_budget
    teams_budget.to_sql(name=table_teams_budget, con=connection, if_exists='replace', index=False)
    if table_teams_budget in inspector.get_table_names():
        print(f"Table '{table_teams_budget}' has been created successfully!")
    else:
        print(f"Table '{table_teams_budget}' does not exist.")

    ## Creating and Loading Table  players_teams
    players_teams.to_sql(name=table_players_teams, con=connection, if_exists='replace', index=False)
    if table_players_teams in inspector.get_table_names():
        print(f"Table '{table_players_teams}' has been created successfully!")
    else:
        print(f"Table '{table_players_teams}' does not exist.")

## Creating and Loading Table  players_salaries
    players_salaries.to_sql(name=table_players_salaries, con=connection, if_exists='replace', index=False)
    if table_players_salaries in inspector.get_table_names():
        print(f"Table '{table_players_salaries}' has been created successfully!")
    else:
        print(f"Table '{table_players_salaries}' does not exist.")

finally:
    connection.close()
```

Figure 3.4.5 Create and Load DB Tables

To verify that the data has been load successfully and verify the creation of the tables it is possible to run a simple query using ‘Select *’ statement. Figure 3.4.6 shows the procedure:



```
#to verify the data in tha tables:
query = f'''SELECT * FROM {table_players_salaries}'''
# Fetch data into a DataFrame
with engine.connect() as connection:
    df = pd.read_sql(query, connection)

print(df)
```

```
PLAYER 2013_14 2014_15 2015_16 2016_17 2017_18 \ •••
```

Figure 3.4. 6 Verify Loaded data in DB Tables

The previous figure illustrates the code utilized to verify that the data has been successfully loaded into the database. The code depicted in the image demonstrates the retrieval of data from the table named table_player_salaries in the Python environment (refer to Table 3.4.1 for the corresponding table name in the database). To verify the data from other tables in the database, the table name should be replaced with the appropriate one intended for exploration.

This data verification concludes the environment setup. A quick recap of the setup is as follows: the database connection has been established. The tables 'players_salaries', 'teams_budget', and 'players_teams' have been created and loaded to proceed with the proposed analysis.

3.3.2 Players Salaries Analysis

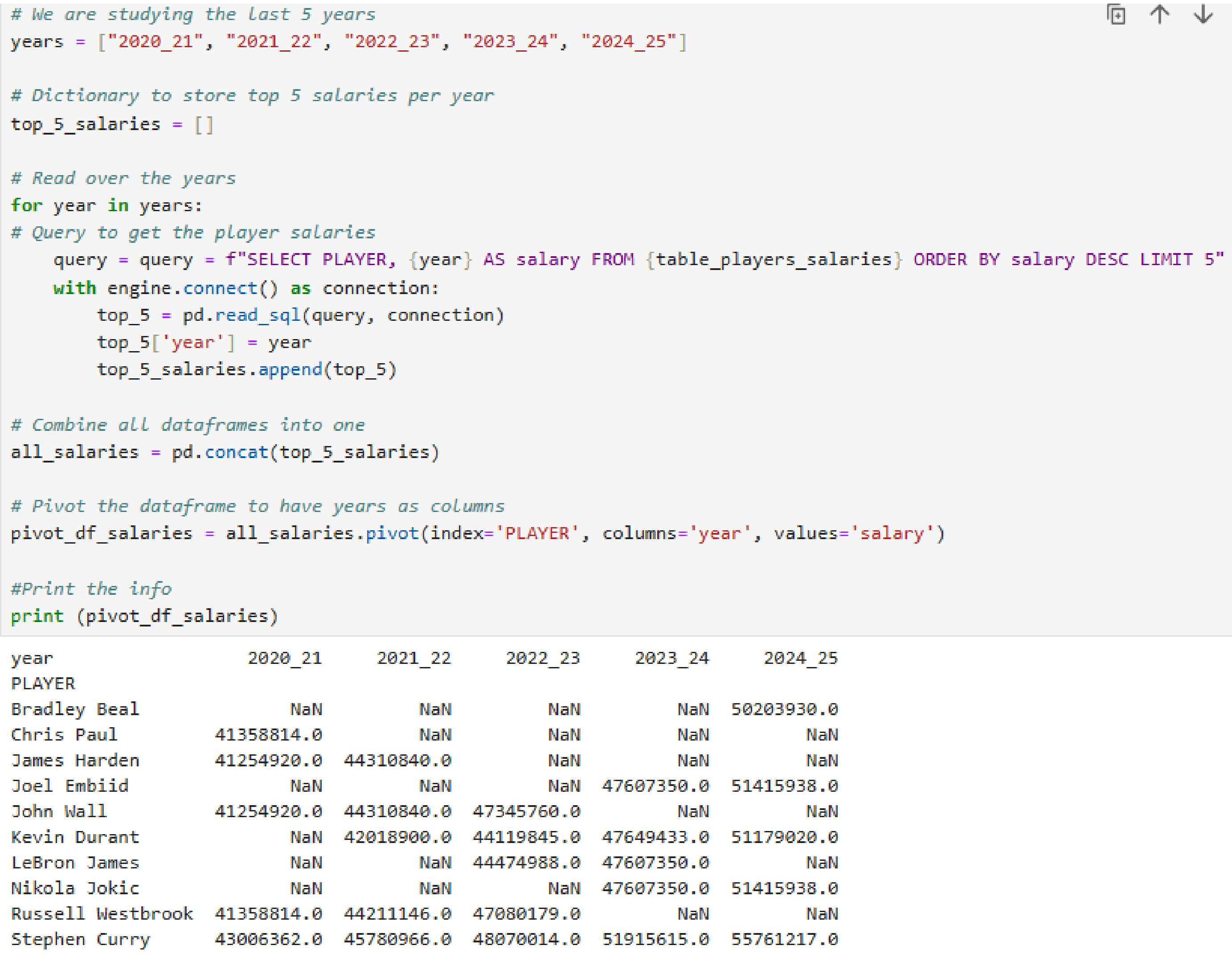
To begin the analysis, the initial step is to extract the top 5 players' salaries for the past 5 years (2020 to 2024). This will help identify how salaries are distributed among players and uncover any patterns or trends in salary allocation over this period.

The expected outcome will be presented in a table format, displaying:

- Player names
- Salaries
- Year

The data will be sorted by year, ensuring that the top salaries for each year are highlighted clearly. This step lays the foundation for further analysis of salary patterns and their relationship to other variables.

Figure 3.4.7 shows the code used to display the top five players salaries in addition to that it shows the output of running the code displaying the names of players, the years and its respective salary per year.



```python
# We are studying the last 5 years
years = ["2020_21", "2021_22", "2022_23", "2023_24", "2024_25"]

# Dictionary to store top 5 salaries per year
top_5_salaries = []

# Read over the years
for year in years:
# Query to get the player salaries
    query = query = f"SELECT PLAYER, {year} AS salary FROM {table_players_salaries} ORDER BY salary DESC LIMIT 5"
    with engine.connect() as connection:
        top_5 = pd.read_sql(query, connection)
        top_5['year'] = year
        top_5_salaries.append(top_5)

# Combine all dataframes into one
all_salaries = pd.concat(top_5_salaries)

# Pivot the dataframe to have years as columns
pivot_df_salaries = all_salaries.pivot(index='PLAYER', columns='year', values='salary')

#Print the info
print (pivot_df_salaries)
```

```
year                  2020_21     2021_22     2022_23     2023_24     2024_25
PLAYER
Bradley Beal              NaN         NaN         NaN         NaN  50203930.0
Chris Paul         41358814.0         NaN         NaN         NaN         NaN
James Harden       41254920.0  44310840.0         NaN         NaN         NaN
Joel Embiid               NaN         NaN         NaN  47607350.0  51415938.0
John Wall          41254920.0  44310840.0  47345760.0         NaN         NaN
Kevin Durant              NaN  42018900.0  44119845.0  47649433.0  51179020.0
LeBron James              NaN         NaN  44474988.0  47607350.0         NaN
Nikola Jokic              NaN         NaN         NaN  47607350.0  51415938.0
Russell Westbrook  41358814.0  44211146.0  47080179.0         NaN         NaN
Stephen Curry      43006362.0  45780966.0  48070014.0  51915615.0  55761217.0
```

Figure 3.4. 7 Top 5 Player's Salaries

From the output of the code in Figure 3.4.7, multiple NaN values are observed, representing missing data for specific fields. In this case, NaN indicates that a player did not appear among the top 5 highest-paid players for a given year.

This occurs because the analysis filters only the top 5 player salaries for each year, resulting in a different set of player names annually. When these yearly results are combined into a single table, Python assigns NaN values for years where a player did not make the top 5 list. While the player may have been active during those years, their salary data is excluded because it falls outside the scope of the top-earner analysis.

As a result, each year in the table includes only 5 values corresponding to the top 5 salaries for that year. The NaN values represent players who were not in the top 5 for that specific year but appear because they were among the top earners in another year. These values are irrelevant for analyzing a given year's top salaries. However, they become relevant in another year, and they are included in the table to provided consistency across all analyzed years.

Another notable observation is that, across the five years analyzed, only 10 players consistently appear in the top 5 highest-paid players' list. This suggests a possible relationship between salaries and team contracts. It is common for athlete contracts to span multiple years, locking players into high-paying agreements. This would explain their recurring presence in the table, as their contracts likely maintained them among the highest earners over several seasons. Another interesting behavior is the constant increment in salaries over the years.

To better illustrate this trend, figure 3.4.8, shows a plot that displays the players and their respective salaries over the years.

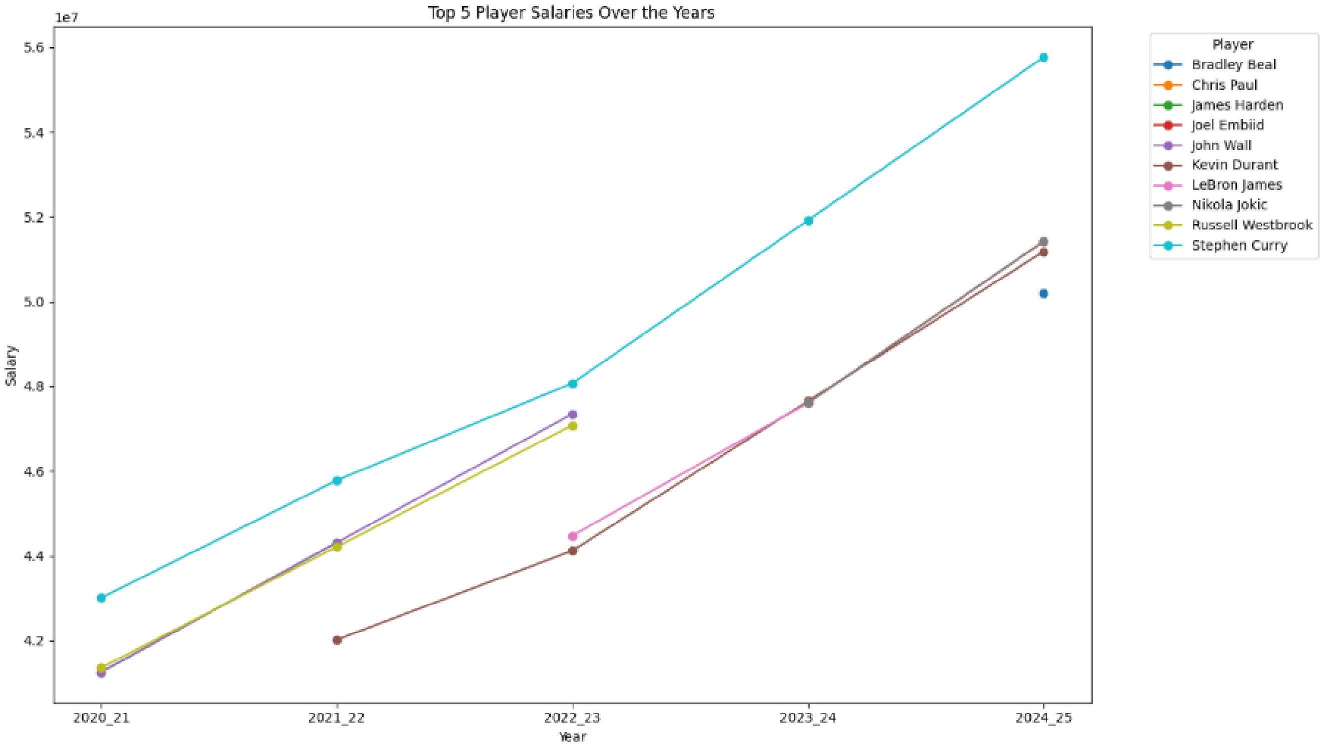


Figure 3.4. 8 Plot Top 5 Player's Salaries over the Years

The plot above illustrates trends in player salaries over the analyzed five-year period, highlighting several key findings:

1. A small group of players consistently ranked among the top five highest-paid athletes, with ten players dominating this category throughout the period. This suggests that once a player secures a top salary, they tend to retain this status for the duration of their contracts.
2. Some salaries are identical, resulting in overlapping lines on the plot, which could indicate standardized salary structures for certain players or contract types.
3. The top five salaries not only repeat among the same players over the years but also exhibit a consistent upward trend, with salaries steadily increasing year after year. While the rate of increase varies among players, the overall incremental pattern is evident across the board.

These findings provide insights into the structure and progression of top player salaries in professional sports, highlighting both stability in salary leadership and ongoing growth in earnings.

3.3.3 Team’s Budgets Analysis

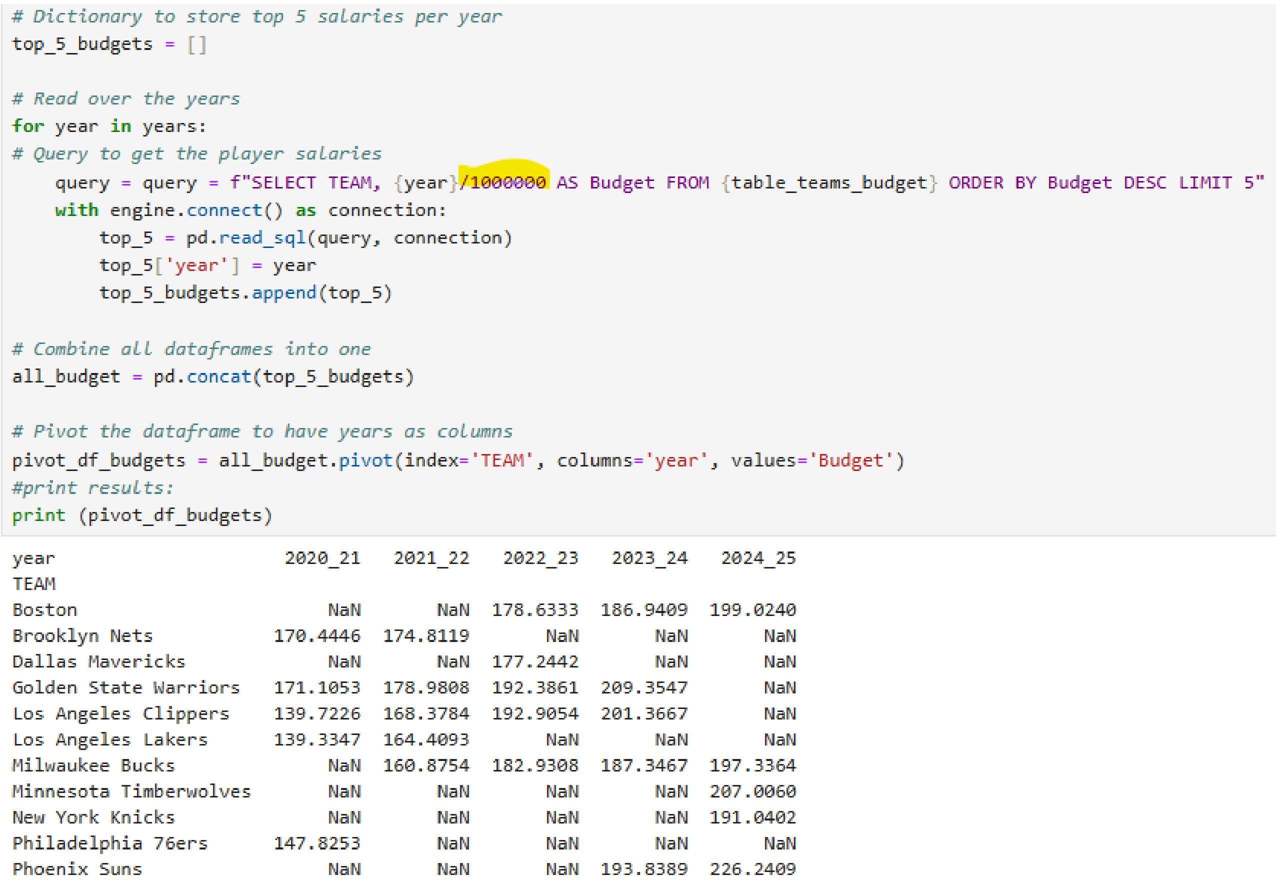
To begin the analysis, the initial step is to extract the top 5 team’s budgets for the past 5 years (2020 to 2024). This will help identify how budgets are distributed among teams and uncover any patterns or trends in budgets allocation over this period.

The expected outcome will be presented in a table format, displaying:

- Team’s names
- Budget
- Year

The data will be sorted by year, ensuring that the top budgets for each year are highlighted clearly. This step lays the foundation for further analysis of salary patterns and their relationship to other variables.

Figure 3.4.9 shows the code used to display the top 5 team’s budgets in addition to that it shows the output of running the code displaying the names of teams, the years and its respective budget per year. This analysis is very similar to the one in the Player’s Salaries section.



```
# Dictionary to store top 5 salaries per year
top_5_budgets = []

# Read over the years
for year in years:
# Query to get the player salaries
    query = query = f"SELECT TEAM, {year}/1000000 AS Budget FROM {table_teams_budget} ORDER BY Budget DESC LIMIT 5"
    with engine.connect() as connection:
        top_5 = pd.read_sql(query, connection)
        top_5['year'] = year
        top_5_budgets.append(top_5)

# Combine all dataframes into one
all_budget = pd.concat(top_5_budgets)

# Pivot the dataframe to have years as columns
pivot_df_budgets = all_budget.pivot(index='TEAM', columns='year', values='Budget')
#print results:
print (pivot_df_budgets)
```

```
year                    2020_21   2021_22   2022_23   2023_24   2024_25
TEAM
Boston                      NaN       NaN  178.6333  186.9409  199.0240
Brooklyn Nets          170.4446  174.8119       NaN       NaN       NaN
Dallas Mavericks            NaN       NaN  177.2442       NaN       NaN
Golden State Warriors  171.1053  178.9808  192.3861  209.3547       NaN
Los Angeles Clippers   139.7226  168.3784  192.9054  201.3667       NaN
Los Angeles Lakers     139.3347  164.4093       NaN       NaN       NaN
Milwaukee Bucks             NaN  160.8754  182.9308  187.3467  197.3364
Minnesota Timberwolves      NaN       NaN       NaN       NaN  207.0060
New York Knicks             NaN       NaN       NaN       NaN  191.0402
Philadelphia 76ers     147.8253       NaN       NaN       NaN       NaN
Phoenix Suns                NaN       NaN       NaN  193.8389  226.2409
```

Figure 3.4.9 Top 5 Teams Budget

From the output of the code in Figure 3.4.9, multiple NaN values are observed, representing missing data for specific fields. In this case, NaN indicates that a team did not appear among the top 5 highest-budget teams for a given year.

This occurs because the analysis filters only the top 5 team budgets for each year, resulting in a different set of teams each year. When these yearly results are combined into a single table, Python assigns NaN values for years where a team did not make the top 5 list. While the team was active during those years, their budget data is excluded because it falls outside the scope of the top-budget analysis.

As a result, each year in the table includes only 5 values corresponding to the top 5 budgets for that year. The NaN values represent teams that were not in the top 5 for that specific year but appear because they were among the top spenders in another year. These values are irrelevant for analyzing a given year's top budgets. However, they become relevant in another year, and they are included in the table to provide consistency across all analyzed years.

Another notable observation is that, across the five years analyzed, only 11 teams out of the 30 consistently appear in the top 5 highest-budget teams' list. This raises the question: why does this happen? Is it related to winning championships, popularity, or the factors that contribute to their budgets? Understanding these dynamics could provide insight into the financial strategies of these teams and how they manage their resources to maintain their position among the highest spenders. In addition to that, another interesting behavior is the constant increase in team budgets over the years.

To better illustrate this trend, Figure 3.4.10 shows a plot that displays the teams and their respective budgets over the years.

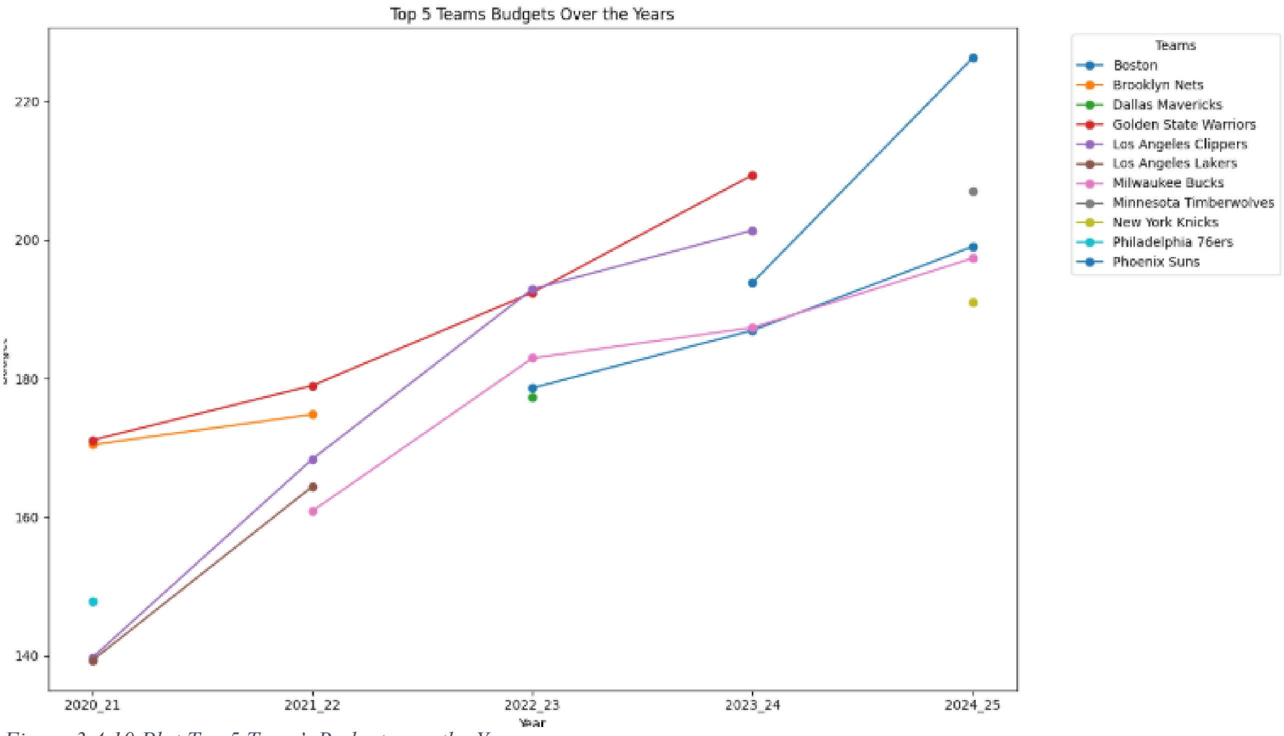


Figure 3.4.10 Plot Top 5 Team's Budget over the Years

The plot above illustrates trends in team budgets over the analyzed five-year period, highlighting several key findings:

- Fewer than half of the teams are included in the top five highest-budget teams.
- Some teams consistently hold the top budgets for more than one year.

The factors influencing this behavior are not immediately clear from the data. The plot reveals that while some teams consistently appear at the top over several years, others are only present for one or two years. A notable observation is the overall increase in team budgets throughout the period analyzed.

3.3.4 Relationship between the Top players salaries and the Top Teams budgets

In this section, the analysis uses the three datasets mentioned in the introductory part of the analysis. The objective is to explore whether there is a relationship between the top 5 player salaries and the top 5 team budgets. In other words, do teams with the highest budgets also have the highest-paid players?

The first step in the analysis is to link players to their respective teams for each season or year. It's important to note that players may change teams throughout their careers, which can lead to fluctuations in their salaries. Therefore, linking players to teams during each season is crucial. This step could help in drawing conclusions such as:

- Higher-budget teams tend to have more top-paid players.
- The top-budget teams always have at least one top-paid player.

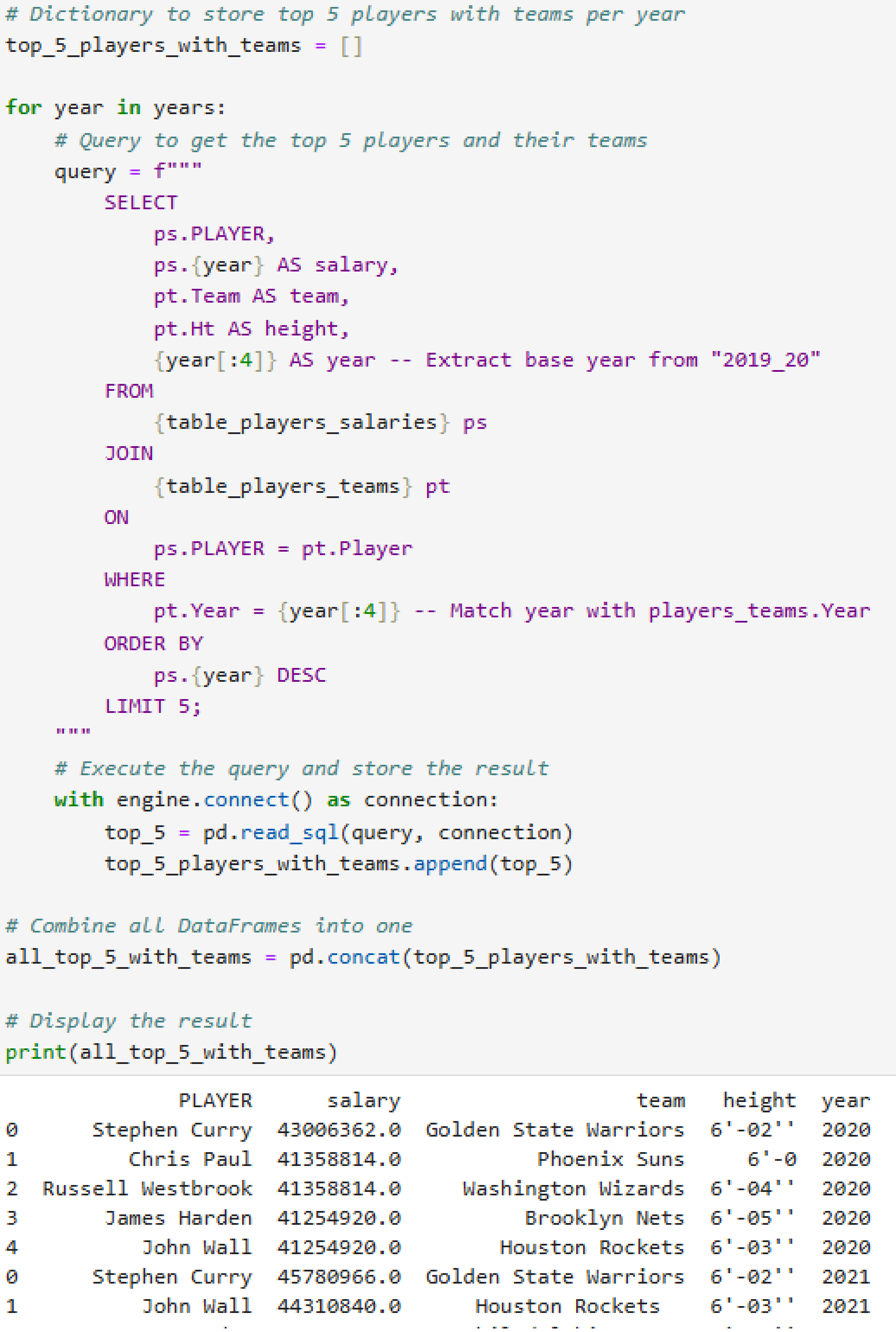
These conclusions may raise further questions, such as whether a team's budget is related to its performance. For example, do teams with higher budgets tend to win more, and does the budget influence performance? However, due to time limitations and the scope of this analysis, we are focusing specifically on examining the relationship between top players and high team budgets.

That being said the first step in this analysis is getting the top 5 players list, link to its respective team, this has been done by joining the tables "players_salaries" and "players_teams" from the DB. Getting a new data frame with the structure shown in table 3.4.2

Player Name	Salary	Team Name	Height	Year

Table 3.4.2 Joined Dataframe

The newly created data frame will contain all the players ranked in the top 5 highest salaries, ordered by year. This means that some player names may appear repeatedly across different years. From the previous analysis, we concluded that only 10 players consistently appear in the top 5 rankings. While it's possible to filter the list to only include unique player names, doing so would not be appropriate for this analysis, as it needs to be performed on a year-by-year basis. In other words, the analysis aims to determine if, in a given year, a top player belonged to or played for a top-budget team. Figure 3.4.11 shows the code used to generate Table 3.4.2



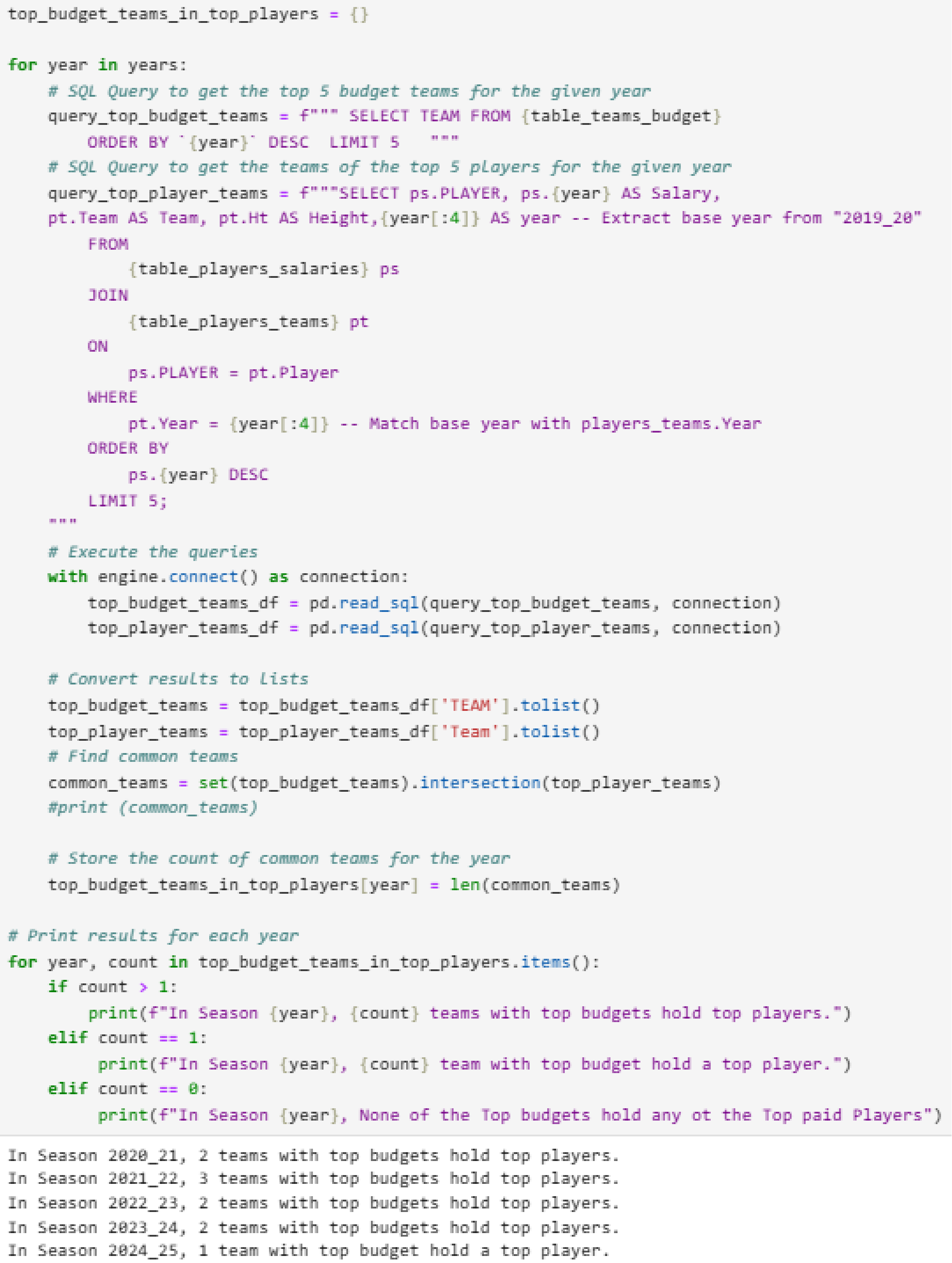
```
# Dictionary to store top 5 players with teams per year
top_5_players_with_teams = []

for year in years:
    # Query to get the top 5 players and their teams
    query = f"""
        SELECT
            ps.PLAYER,
            ps.{year} AS salary,
            pt.Team AS team,
            pt.Ht AS height,
            {year[:4]} AS year -- Extract base year from "2019_20"
        FROM
            {table_players_salaries} ps
        JOIN
            {table_players_teams} pt
        ON
            ps.PLAYER = pt.Player
        WHERE
            pt.Year = {year[:4]} -- Match year with players_teams.Year
        ORDER BY
            ps.{year} DESC
        LIMIT 5;
    """
    # Execute the query and store the result
    with engine.connect() as connection:
        top_5 = pd.read_sql(query, connection)
        top_5_players_with_teams.append(top_5)

# Combine all DataFrames into one
all_top_5_with_teams = pd.concat(top_5_players_with_teams)

# Display the result
print(all_top_5_with_teams)
```

```
              PLAYER      salary                   team  height  year
0      Stephen Curry  43006362.0  Golden State Warriors  6'-02''  2020
1         Chris Paul  41358814.0           Phoenix Suns     6'-0  2020
2  Russell Westbrook  41358814.0     Washington Wizards  6'-04''  2020
3       James Harden  41254920.0          Brooklyn Nets  6'-05''  2020
4          John Wall  41254920.0        Houston Rockets  6'-03''  2020
0      Stephen Curry  45780966.0  Golden State Warriors  6'-02''  2021
1          John Wall  44310840.0        Houston Rockets   6'-03''  2021
```

Figure 3.4.11 Top Players in Top Teams

With the table and information now prepared, we can relate the top 5 players' salaries to their respective teams and compare this to the list of top-budget teams previously created. If a player belongs to a top-budget team during the analyzed years, it becomes possible to conclude that there is a relationship between the top-paid players and the top-budget teams.

Figure 3.4.12 displays the code used to compare the top 5 players against the top 5 budgets for the given years. The analysis employs SQL queries to extract relevant information from the database tables, joining them to facilitate comparison and analysis. Additionally, Python code is used to compute the intersections of these SQL query results and to create new DataFrames for further comparison.



```
top_budget_teams_in_top_players = {}

for year in years:
    # SQL Query to get the top 5 budget teams for the given year
    query_top_budget_teams = f""" SELECT TEAM FROM {table_teams_budget}
        ORDER BY `{year}` DESC  LIMIT 5   """
    # SQL Query to get the teams of the top 5 players for the given year
    query_top_player_teams = f"""SELECT ps.PLAYER, ps.{year} AS Salary,
    pt.Team AS Team, pt.Ht AS Height,{year[:4]} AS year -- Extract base year from "2019_20"
        FROM
            {table_players_salaries} ps
        JOIN
            {table_players_teams} pt
        ON
            ps.PLAYER = pt.Player
        WHERE
            pt.Year = {year[:4]} -- Match base year with players_teams.Year
        ORDER BY
            ps.{year} DESC
        LIMIT 5;
    """
    # Execute the queries
    with engine.connect() as connection:
        top_budget_teams_df = pd.read_sql(query_top_budget_teams, connection)
        top_player_teams_df = pd.read_sql(query_top_player_teams, connection)

    # Convert results to lists
    top_budget_teams = top_budget_teams_df['TEAM'].tolist()
    top_player_teams = top_player_teams_df['Team'].tolist()
    # Find common teams
    common_teams = set(top_budget_teams).intersection(top_player_teams)
    #print (common_teams)

    # Store the count of common teams for the year
    top_budget_teams_in_top_players[year] = len(common_teams)

# Print results for each year
for year, count in top_budget_teams_in_top_players.items():
    if count > 1:
        print(f"In Season {year}, {count} teams with top budgets hold top players.")
    elif count == 1:
         print(f"In Season {year}, {count} team with top budget hold a top player.")
    elif count == 0:
         print(f"In Season {year}, None of the Top budgets hold any ot the Top paid Players")
```

```
In Season 2020_21, 2 teams with top budgets hold top players.
In Season 2021_22, 3 teams with top budgets hold top players.
In Season 2022_23, 2 teams with top budgets hold top players.
In Season 2023_24, 2 teams with top budgets hold top players.
In Season 2024_25, 1 team with top budget hold a top player.
```

Figure 3.4.12 Top Players in Top Teams

The output of the code shows a count of the top-paid players associated with top-budget teams in each year. If no players from the top salary list are associated with top-budget teams, the output will explicitly indicate that no relationship exists for that year. This approach enables a year-by-year evaluation of the potential connection between player salaries and team budgets.

The result for this analysis suggests that:

- During season 2020_21, 2 top budget teams hold top paid players.
- During season 2021_22, 3 top budget teams hold top paid players.
- During season 2022_23, 2 top budget teams hold top paid players.
- During season 2023_24, 2 top budget teams hold top paid players.
- During season 2024_25, 1 top budget team hold a top paid player.

Based on the output, it is evident that a relationship exists between top-paid players and top-budget teams. While the exact strength of this relationship remains unknown and would require deeper analysis to quantify, this preliminary evaluation confirms a connection between these two factors. This observation underscores the potential influence of team budgets on their ability to attract and retain high-salary players, suggesting a dynamic worth exploring further in subsequent analyses.

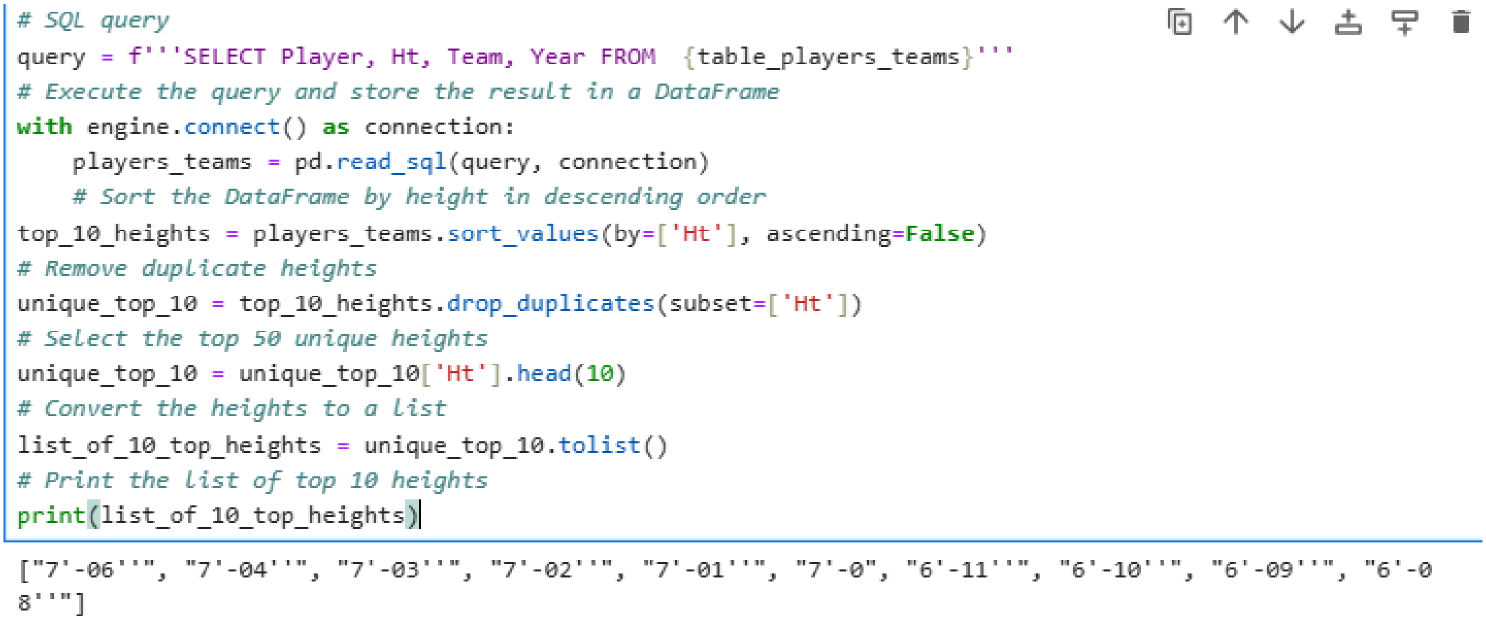
3.4 Relationship between the Top players salaries and the Player's Height

From the previous analysis, a new DataFrame was created that includes the names of players, their salaries, teams, years, and heights (Table 3.4.2). The focus of this section is to investigate whether there is a relationship between high salaries and player height.

The analysis involves extracting a list of all recorded heights from the "players_teams" table and comparing it against the heights of players listed in Table 3.4.2. Using the code shown in Figure 3.4.11, the table was constructed to include player names for the five analyzed years. In that context, repetitive values of player names were retained because they were relevant for year-based analysis.

For this height-based analysis, however, the repetitive values are not needed, as a player’s height does not vary across the analyzed years. Therefore, duplicate player names have been removed from the DataFrame to focus on unique players and their respective heights. This adjustment ensures a more accurate and streamlined comparison between height and salary.

To begin this analysis, the first step is to retrieve all player heights from the NBA players list contained in the "players_teams" table. Figure 3.4.13 demonstrates the procedure for accomplishing this. The figure also illustrates the subsequent filtering process, which reduces the list to include only the top 10 heights from all the players.



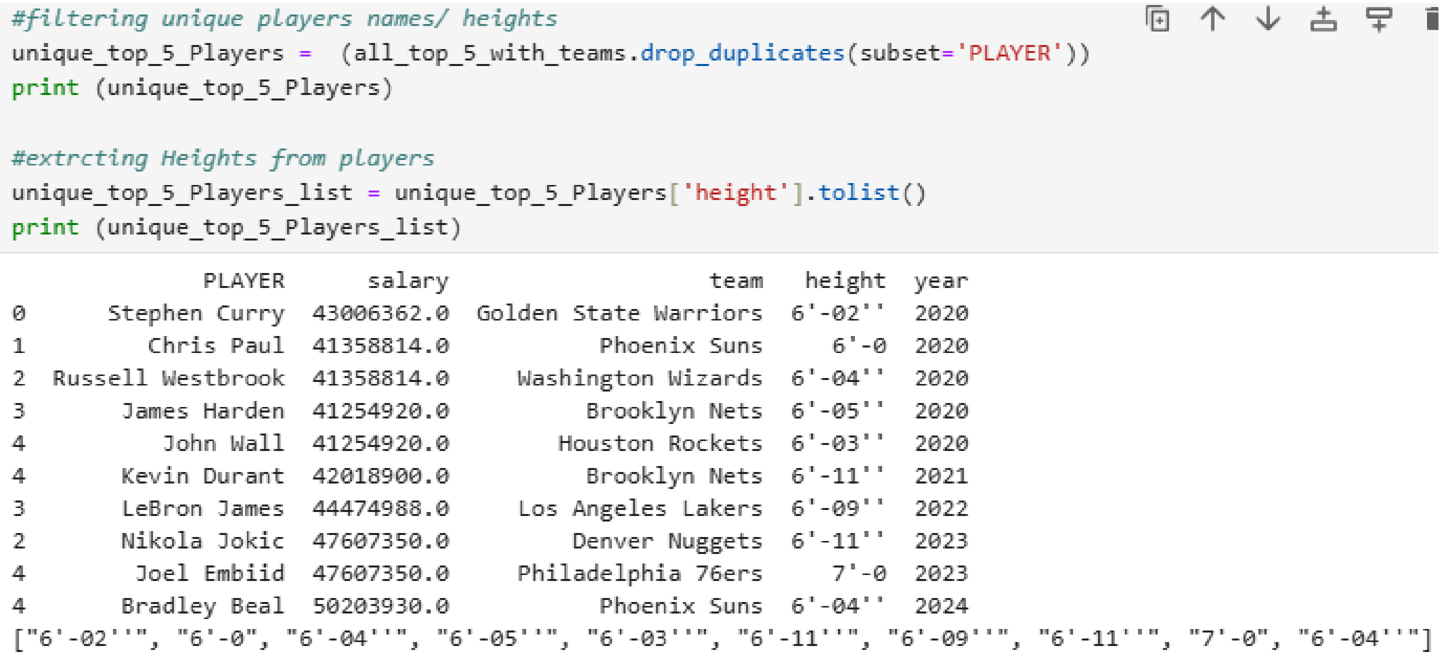
```
# SQL query
query = f'''SELECT Player, Ht, Team, Year FROM  {table_players_teams}'''
# Execute the query and store the result in a DataFrame
with engine.connect() as connection:
    players_teams = pd.read_sql(query, connection)
    # Sort the DataFrame by height in descending order
top_10_heights = players_teams.sort_values(by=['Ht'], ascending=False)
# Remove duplicate heights
unique_top_10 = top_10_heights.drop_duplicates(subset=['Ht'])
# Select the top 50 unique heights
unique_top_10 = unique_top_10['Ht'].head(10)
# Convert the heights to a list
list_of_10_top_heights = unique_top_10.tolist()
# Print the list of top 10 heights
print(list_of_10_top_heights)
```

```
["7'-06''", "7'-04''", "7'-03''", "7'-02''", "7'-01''", "7'-0", "6'-11''", "6'-10''", "6'-09''", "6'-0
8''"]
```

Figure 3.4.13 Top 10 Heights from NBA Players

With the filtered list of the top 10 heights obtained, the next step is to compare it against the list of top-paid players derived from the previous analysis. Figure 3.4.14 demonstrates the process of filtering the top-paid players, ensuring that only unique values are retained by removing repetitive entries. This results in a refined dataset of 10 players, representing the top earners over the five analyzed years, as previously discussed.

The figure also illustrates the extraction of heights corresponding to these 10 players. These extracted heights will serve as the basis for comparison with the top 10 heights identified in Figure 3.4.13, enabling further analysis of potential relationships between player height and high salaries.



```
#filtering unique players names/ heights
unique_top_5_Players =  (all_top_5_with_teams.drop_duplicates(subset='PLAYER'))
print (unique_top_5_Players)

#extrcting Heights from players
unique_top_5_Players_list = unique_top_5_Players['height'].tolist()
print (unique_top_5_Players_list)
```

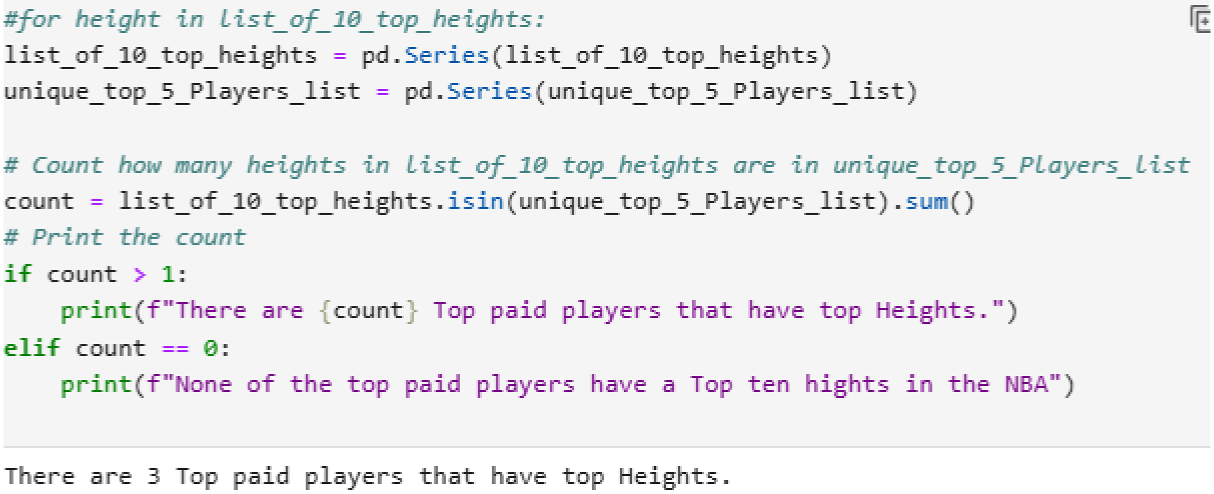
```
            PLAYER      salary                   team   height  year
0      Stephen Curry  43006362.0  Golden State Warriors  6'-02''  2020
1         Chris Paul  41358814.0           Phoenix Suns     6'-0  2020
2  Russell Westbrook  41358814.0     Washington Wizards  6'-04''  2020
3       James Harden  41254920.0          Brooklyn Nets  6'-05''  2020
4          John Wall  41254920.0        Houston Rockets  6'-03''  2020
4       Kevin Durant  42018900.0          Brooklyn Nets  6'-11''  2021
3       LeBron James  44474988.0     Los Angeles Lakers  6'-09''  2022
2       Nikola Jokic  47607350.0         Denver Nuggets  6'-11''  2023
4        Joel Embiid  47607350.0     Philadelphia 76ers     7'-0  2023
4       Bradley Beal  50203930.0           Phoenix Suns  6'-04''  2024
["6'-02''", "6'-0", "6'-04''", "6'-05''", "6'-03''", "6'-11''", "6'-09''", "6'-11''", "7'-0", "6'-04''"]
```

Figure 3.4.14 Unique top !0 player Heights over 5 years

Having this data, the final step involves comparing the two lists: the heights of the top-paid players and the top 10 heights from all NBA players. The comparison will count the occurrences where a height from the top-paid players matches one of the top 10 heights from the overall NBA dataset.

This count provides an indication of the relationship between player height and salary, offering insights into how relevant or significant height is among top earners in the NBA. A high number of matches could suggest that taller players are more likely to be among the highest-paid, while fewer matches might indicate that height is less of a factor in determining salary.

Figure 3.415 shows the code that makes this comparison and the final result.



```
#for height in list_of_10_top_heights:
list_of_10_top_heights = pd.Series(list_of_10_top_heights)
unique_top_5_Players_list = pd.Series(unique_top_5_Players_list)

# Count how many heights in list_of_10_top_heights are in unique_top_5_Players_list
count = list_of_10_top_heights.isin(unique_top_5_Players_list).sum()
# Print the count
if count > 1:
    print(f"There are {count} Top paid players that have top Heights.")
elif count == 0:
    print(f"None of the top paid players have a Top ten hights in the NBA")
```

```
There are 3 Top paid players that have top Heights.
```

Figure 3.4. 15 Top players Height vs Top 10 Heights in NBA

From the output shown in Figure 3.4.15, it can be concluded that while there is some correlation between height and salary, the relationship is not particularly strong. Specifically, only 3 out of the 10 players with the highest salaries are among the top 10 tallest players in the dataset, whose heights range from '7-6' to '6-6'.

This finding is particularly shocking because the most common height in the dataset is 6-5, slightly shorter than the height range analyzed. This suggests that factors other than height, such as skill level, team contribution, or marketability, could play a more significant role in determining player salaries in the NBA.

..

4 Conclusion

The project will be considered successful:

- If it identifies significant correlations between demographic patterns and team success, providing insights that can help NBA leaders improve their overall performance.
- If uncover trends and insights that will help explain how salaries and budgets impact overall success in basketball.

Conclusion From Teams - Coach Analysis

Overall, we concluded that coaches are not actually all that important in the NBA. This is most likely why their salaries are so low when compared to their team's players. Surprisingly, coaches with previous NBA experience do not have an upper hand over coaches without NBA experience. This may be because those coaches without NBA experience have better education and more experience coaching universities and colleges to make up for the lack of experience. We also found that coaches with relevant education do not have a statistical difference than coaches without relevant education. For both research questions, we require significantly more data since it is difficult to find data on coaches since there are only 30 for each season and many coaches are employed for multiple years. Surprisingly more years as a coach does not equal a higher win rate. Although the data looked promising, we could not find statistical significance. This may be due to a few coaches having many years of coaching experience but terrible performances later in their careers. We also found out that teams with a higher budget do not prefer the coach they choose. Surprisingly, teams also did not care if their coaches played in the NBA which shows how little influence the media has on which coach is chosen. After combining all of the data together, we found that due to so few international players being selected to play in the NBA during the early 2000s and then becoming coaches, there was no relationship between your country of origin and how good of a coach you were. This came down to not having nearly enough data. The most challenging aspect of this project was finding a unique way to combine data on coaches with players since they are two completely different jobs. Luckily, there are a few coaches with previous NBA experience but not nearly enough to find statistical significance. We also found that coaches who played in the NBA for good teams performed at the same level in coaching as coaches who played in the NBA for bad teams. Again, we need significantly more data to continue this analysis.

The dataset revealed that coaches, on average, do not significantly influence win rates based on the variables analyzed, such as NBA playing experience or relevant education. However, we found a trend where coaches with NBA experience tend to have longer careers. In the future, we would prioritize collecting additional data on coaches, such as leadership styles, player feedback, and team chemistry, to better understand their impact. Additionally, automating the data collection process could reduce reliance on manual efforts and improve data accuracy.

Conclusion from Teams –Demographics Analysis.

In conclusion, the analysis of NBA player demographics provided valuable insights into how various factors such as height, weight, international diversity, and age influence team performance. The first research question, which examined the correlation between average player height and team scoring, revealed no significant relationship, as indicated by the p-value of 0.081. However, the analysis of player weight showed a statistically significant negative relationship with scoring, suggesting that teams with heavier players may score fewer points on average.The second research question explored the impact of international players on team performance. The findings suggested that teams with a higher proportion of international players (over 50%) tend to have a higher win percentage, indicating that international diversity could positively influence a team's overall performance.

Finally, the analysis of age revealed significant performance differences between younger and older players. With a p-value of 0.000 for points, rebounds, and assists, the results showed that older players generally perform better than younger players across all measured metrics. Overall, the analysis highlights the importance of demographic factors in shaping team success. While physical attributes like weight and age appear to influence performance, the proportion of international players also seems to have a positive effect on team success. These findings offer valuable insights that can help teams refine their drafting strategies and build more effective, well-rounded rosters.

Conclusion From Teams - Players Performance Analysis

Based on the above, the analysis reveals that NBA team success, measured by win percentage, depends on collective performance, teamwork, and shooting efficiency rather than individual player statistics alone. Shooting efficiency metrics, like field goal and three-point percentages, moderately correlate with win percentage, highlighting their importance in driving success. Individual player stats, such as points, assists, and rebounds, show weak correlations with win percentage, emphasizing the value of balanced team contributions. Team points moderately correlate with success, but other factors like defense and turnovers also play key roles. Strong correlations between player points and assists demonstrate the importance of team-oriented play, while rebounds show mixed relationships. Long-term success is achieved by balancing offensive efficiency, defensive strength, and teamwork. Overall, sustained success requires optimizing both individual and collective efforts alongside strategic scoring and defense.

Conclusion From Salary – Budget Analysis

Based on the analysis of the top five highest-paid players and the top-budget NBA teams from 2020 to 2024, a clear relationship is evident between top-paid players and teams with the largest budgets. While the precise strength of this relationship remains unclear and would require further analysis to quantify, this preliminary evaluation confirms a connection between these two factors.

The findings underscore the potential influence of team budgets on their ability to attract and retain high-salary players, suggesting a dynamic worth exploring in future studies. Speaking broadly, the impact of economics on team success lacks standardization. However, the analysis indicates that a team's economic success is largely determined by its budget. Among the top 10 teams consistently appearing in the analysis, the Los Angeles Clippers, Golden State Warriors, and Milwaukee Bucks stand out as the most successful economically, with regular appearances on the list in at least three of the five years analyzed.

A similar conclusion applies to the top players. The five highest-paid players enjoy significant financial success, with Stephen Curry, Kevin Durant, and John Wall frequently appearing on the list. This positions them as the most successful basketball players financially during the period analyzed.

Nevertheless, the relationship between financial success (e.g., salaries and budgets) and performance success (e.g., championships, win rates) remains inconclusive and requires further investigation to deeply understand the relationship.

One key lesson to change for next time is that we should aim to include data on salary caps, luxury taxes, and endorsements to provide a more comprehensive view of financial decision-making in the NBA. Additionally, a broader scope, including mid-range and lower-tier salaries, could offer insights into overall salary distribution and team dynamics. Any future analysis should be focused on analyzing the impact of financial constraints, like salary caps, on team performance and exploring the relationship between budget allocation and long-term success could provide valuable insights for team management.

Conclusion From Salary – Height Analysis

Based on the executed analysis it can be concluded that while there is some correlation between height and salary, the relationship is not particularly strong. Specifically, only 3 out of the 10 players with the highest salaries are among the top 10 tallest players in the dataset. This suggests that factors other than height, such as skill level, team contribution, or marketability, could play a more significant role in determining player salaries in the NBA.

Conclusion From Team Performance

This dataset provided insights into how various performance metrics influence team success. Metrics like shooting efficiency showed moderate correlations with win percentage, emphasizing their importance in driving success. However, individual statistics like points, assists, and rebounds had weak correlations, highlighting the importance of teamwork over individual contributions. We learned some key lessons and should do a few aspects differently. Future analyses could focus on integrating defensive metrics and turnovers, which are critical but were less emphasized. Additionally, using more granular data, such as in-game situations or player combinations, could offer richer insights. Also expanding the analysis to include advanced performance metrics (e.g., player efficiency ratings) and investigating trends over a longer timeline could help identify evolving strategies and their impact on team success.

References

Mark Sanchez. (2022) Steve Kerr admits 'I'm hungover right now' day after Warriors' championship derived from *https://nypost.com/2022/06/18/steve-kerr-admits-im-hungover-right-now-after-warriors-title/*

NBA Coaches derived from: https://www.basketball-reference.com/leagues/NBA_2023_coaches.html

NBA Teams Stats *https://www.kaggle.com/datasets/mharvnek/nba-team-stats-00-to-18/code*

NBA Players derived from: *https://www.kaggle.com/datasets/justinas/nba-players-data/data*
Team dunkers (2023) How Much Is a NBA Championship Ring Worth derived from: *https://www.dunkest.com/en/nba/news/127624/how-much-nba-championship-ring-worth*

[4] NBA hoopsnype derived from *https://hoopshype.com/salaries/*

[5] Basketball-Reference derived from https://www.basketball-reference.com/