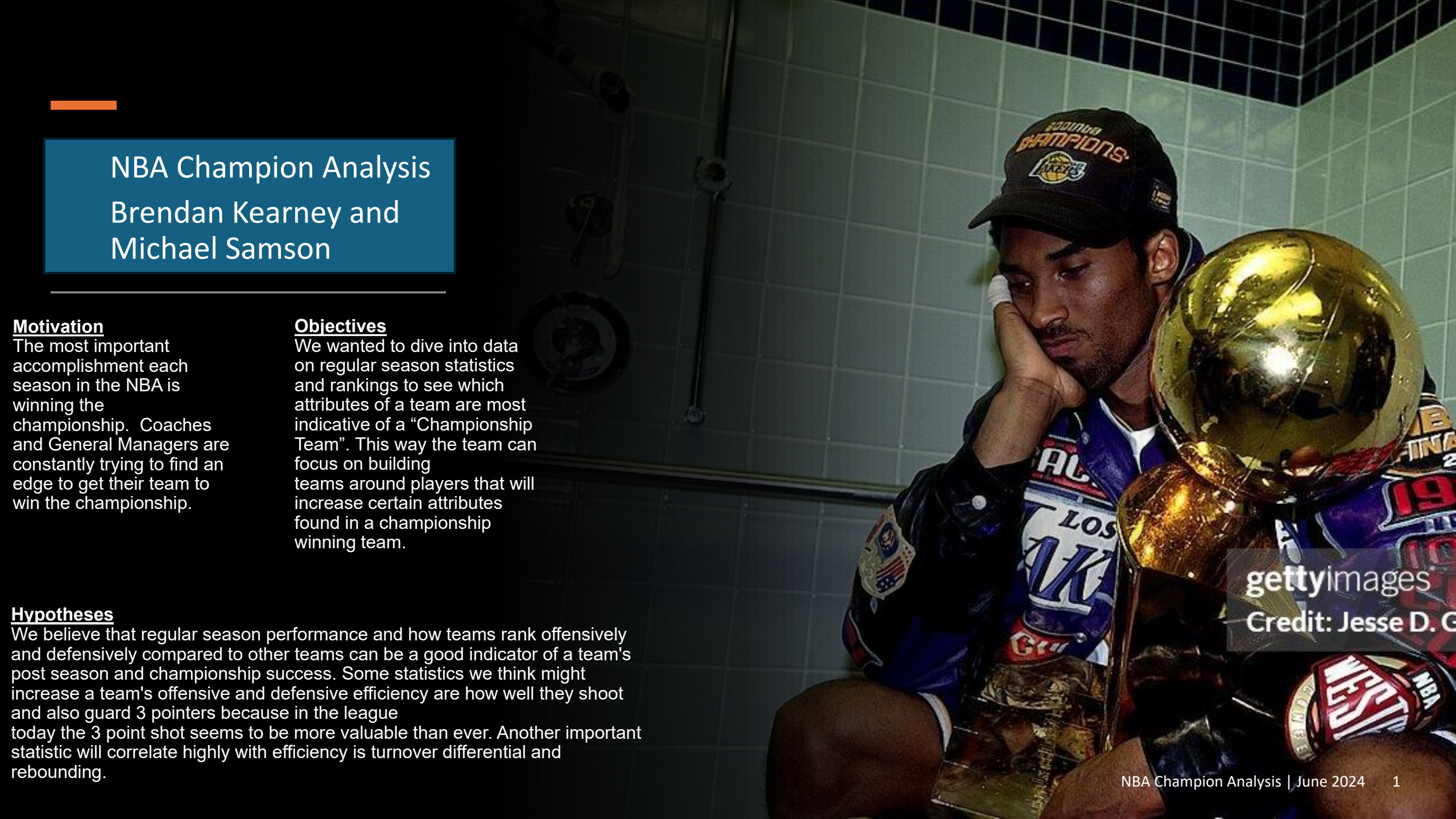
NBA Champion Analysis

Brendan Kearney and
Michael Samson

Motivation
The most important accomplishment each season in the NBA is winning the championship. Coaches and General Managers are constantly trying to find an edge to get their team to win the championship.

Objectives
We wanted to dive into data on regular season statistics and rankings to see which attributes of a team are most indicative of a “Championship Team”. This way the team can focus on building
teams around players that will increase certain attributes found in a championship winning team.

Hypotheses
We believe that regular season performance and how teams rank offensively and defensively compared to other teams can be a good indicator of a team's post season and championship success. Some statistics we think might increase a team's offensive and defensive efficiency are how well they shoot and also guard 3 pointers because in the league
today the 3 point shot seems to be more valuable than ever. Another important statistic will correlate highly with efficiency is turnover differential and rebounding.



Data Source	Description	Key Variables and Size	Link
NBA_API Python Package	This package is an API client for nba.com that can be directly imported into Jupyter Notebooks. The table we focused on was the team estimated metrics. For the purposes of this project, we pulled data going back to the 2003-2004 NBA season, but more can be pulled if desired	*Key Variables:* • Offensive/Defensive/Net Ranking Final Size: • 629 rows x 31 cols	• NBA API Documentaton • TeamEstimatedMetrics
Basketball Reference	The primary dataset that we pulled from basketball reference was the offensive and defensive statistics of each team in the NBA. We pulled data going back to the 2003-2004 NBA season, but more can be pulled if desired as well	*Key Variables:* • Team/Opponent Three Point and Field Goal Percentage • Team Turnovers and Opponents Blocking Final Size: • 629 rows x 49 cols	2023-24 NBA Season Summary | Basketball-Reference.com
	Basketball Reference also had two separate tables that showed historical data going back to 1947. For the purposes of this project, we only worked with the past 20 years as well.	Key Variable: • Championship Team Final Size: • 88 rows x 3 cols	• Historical NBA MVP Data • NBA & ABA Champions | Basketball-Reference.com

Data Manipulation Methods: NBA API

Importing Packages

This package can be directly imported into the Jupyter notebook using the line `!pip install nba_api` and then importing then the class needed to access the data with `from nba_api.stats.endpoints import TeamEstimatedMetrics.`

Team Estimated Metrics Object

We then instantiate a TeamEstimatedMetrics object that takes in 3 variables:

- **LeagueID:** Defaults to *"00"* for the NBA. But can also retrieve WNBA, ABA, and G-league data.
- **Season:** Takes in the Year of the season formatted like *"2023-24"*.
- **SeasonType:** *["Regular Season" (default), "All Star", "Playoffs", "Pre Season"].*

Accessing Data

A Pandas DataFrame can be created by using the `get_data_frames()` method. This returns a list that contains the DataFrame based on the parameters that we passed. To get the DataFrame from the returned list we just added a `[0]` following the `get_data_frames method call.`

Historical Data

Once we created a function that could successfully pull a TeamEstimatedMetrics DataFrame for a given year, we created a function that would pull for multiple years and append them together. To do this we created a loop that calls back to our function that pulls the DataFrame for the given year. We added a season column each time to identify each and then appended them on to each other using the pandas `concat function.`

Obstacles

We were lucky that this dataset was very well kept so we did not have to deal with missing values. The sole errant value we had was that the format of the Clippers team name was LA Clippers, which we had to replace to Los Angeles Clippers. The biggest issue we encountered was when we were pulling the historical data. We had to figure out a way to manipulate the years into strings to pass them into the function to pull the TeamEstimatedMetrics data.

Data Manipulation Methods: Basketball Reference

Parsing HTML content

The first step was to send an HTTP request to the basketball reference url: `requests.get(url).` Next, we parse the HTML of the website returned from our HTTP request using: `BeautifulSoup(response.content, "html.parser").`

Identify Tables

By looking at the source code for the website, we were able to identify the table names was "per_game-team" and "per_game-opponent". We were able to pass this table name to *soup.find("table", {"id": table_name}) to get a BeautifulSoup Tag Element.*

Extracting Column Names

Using list comprehension, we extracted all the column names, denoted by the HTML tag of "th". Once we pulled those, we found that there were unnecessary trailing headers that we had to trim excess header using list slicing.

Extracting Data

To get the data we had to loop through cell by cell by first finding all the rows labeled "tr" and identifying the cells for each row labeled "td". If there were cells in the rows, we would get the text from each cell in the row and append that returned list to a team_stats list.

Creating DataFrame

Finally, we were able to create a DataFrame using the team_stats list of rows as the data and the trimmed_headers as the columns. Upon creation of the DataFrame we found that each team name had a trailing * which we got rid of by using str.replace("*", "").

Joining Tables

We then joined the team data with the opponent data to get one full DataFrame with both the team statistics and opponent statistics for the given year. We did this using an inner join on the "Team" column and added suffixes on the end of each column to differentiate.

Historical Data

Once we created a function that could successfully pull a merged team and opponent dataset for a given year, we created a function that would pull for multiple years and append them together. To do this we created a loop that calls back to our functions. We added a season column each time to identify each and then appended them on to each other using the pandas `concat function.`

Obstacles

Our biggest obstacle was encountered when pulling historical data. This is because we were doing a lot of pull requests to the basketball reference API when looping through. We would eventually run into an issue where we would get a "Too Many Requests" error and then we would no longer be able to make any more requests from the notebook. To solve this issue, we added in a line that would cause the loop to pause each time before making the next request using: time.sleep(5).

Data Manipulation Methods: Merging to Final Dataset

Merging Main Datasets

We performed an inner join on our full basketball reference dataset and our nba_api dataset, using the team names and season columns as our merge keys.

Champion Dataset

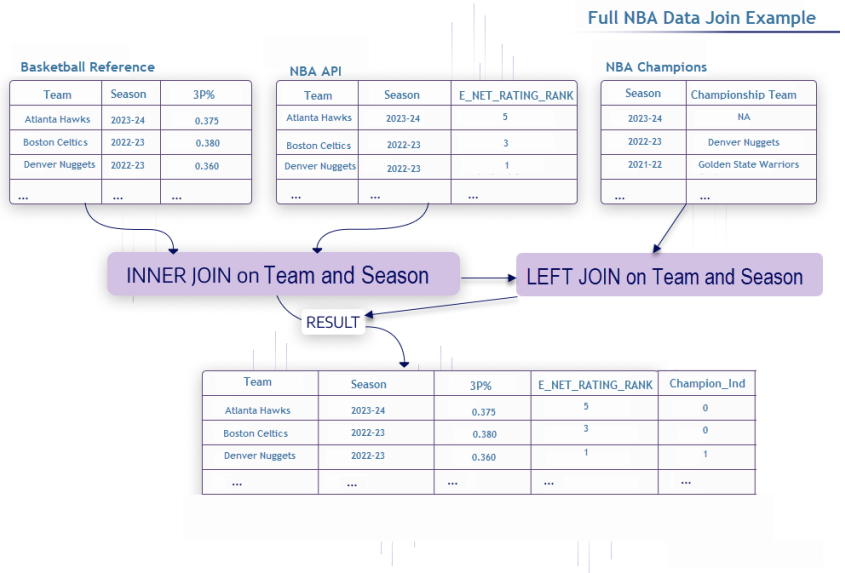
A key variable that we wanted to investigate when comparing our historical data that we did not have in either of our datasets is who won the championship in the given year. To solve for this, we downloaded a dataset on historical data that we would later join to our datasets.

Adding in Champion Indicator

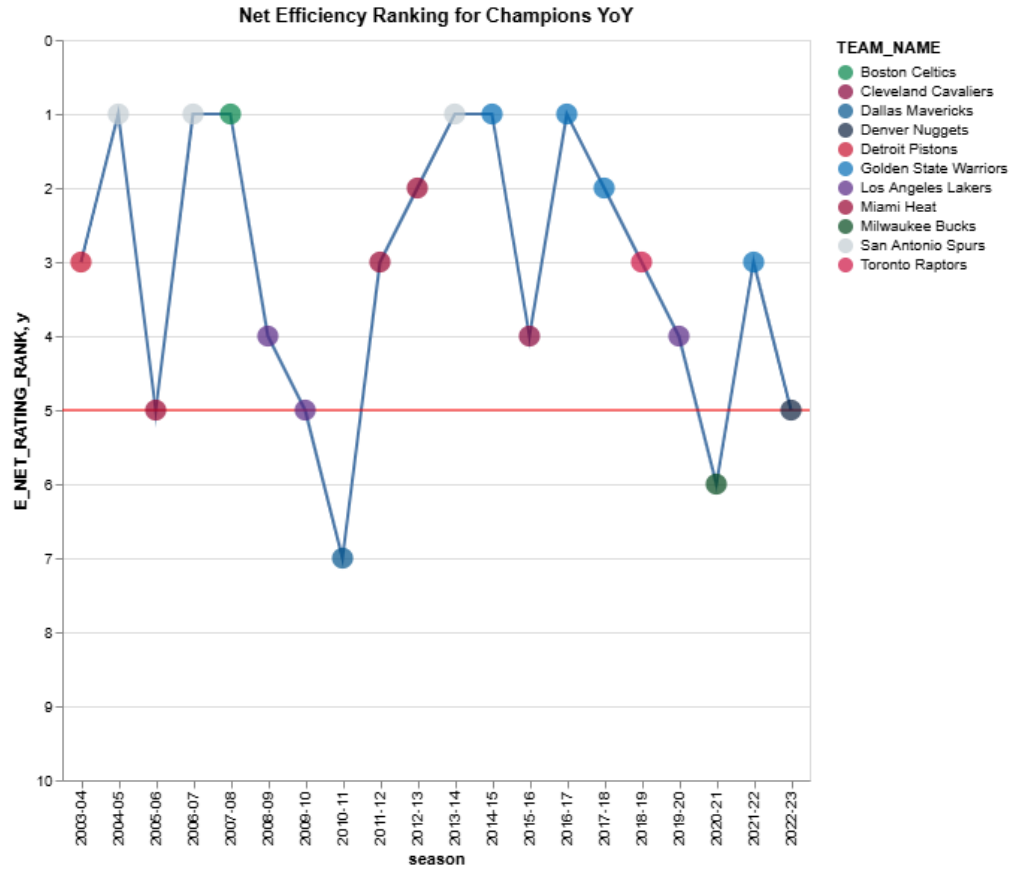
To get a champion indicator reference our champion dataset based on Team name and season to the main datasets. For all the rows that were not null we created an indicator of 1 that signified the team was a champion that year.

Data Quality Checks

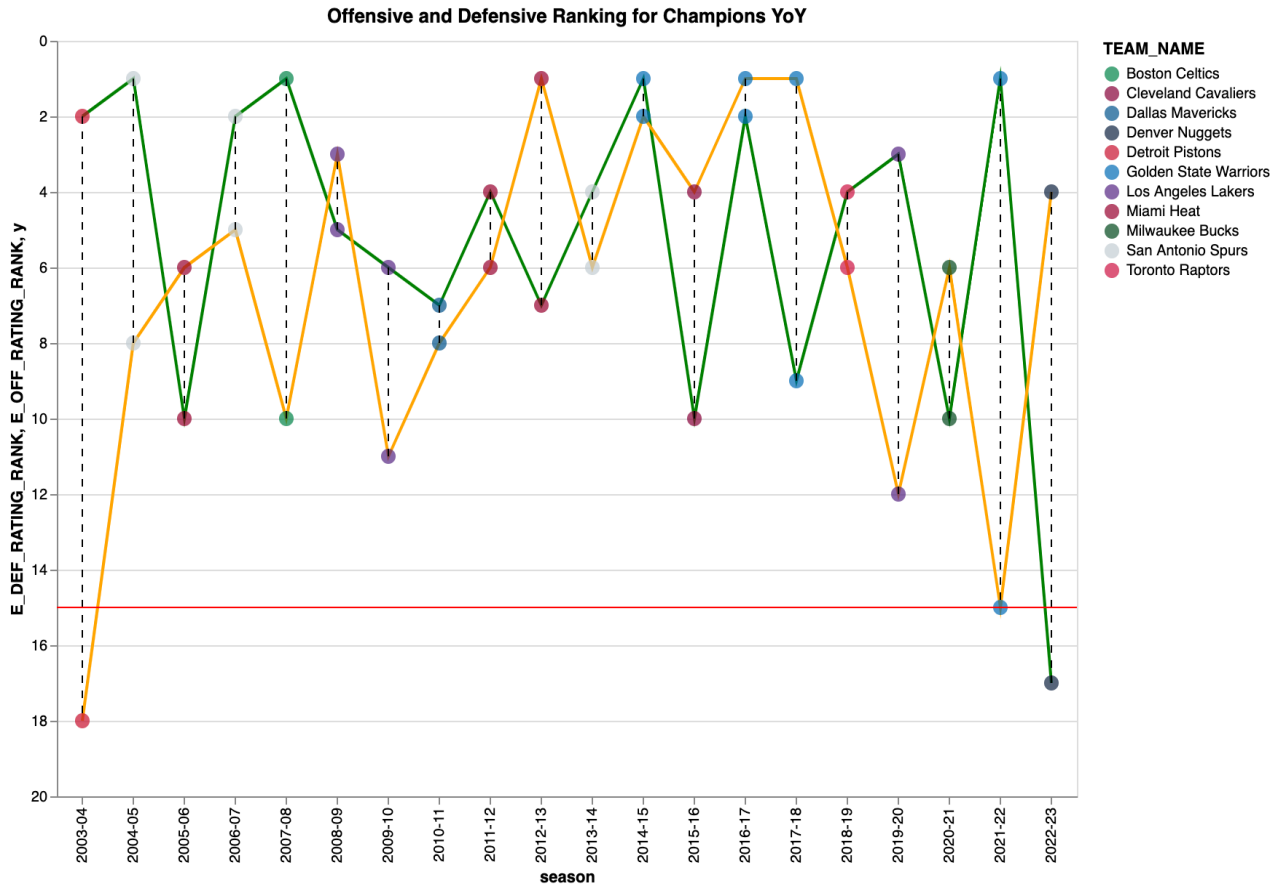
To confirm that we had successfully merged our data accurately, we added in some print statements in our function to print the number of rows of the DataFrame *(629 for 20 years of historical data, since the 2003-2004 season only had 29 teams)* and that we had the right number of champions*(20).*



Analysis/Visualizations: NBA API Efficiency Rankings of Past Champions

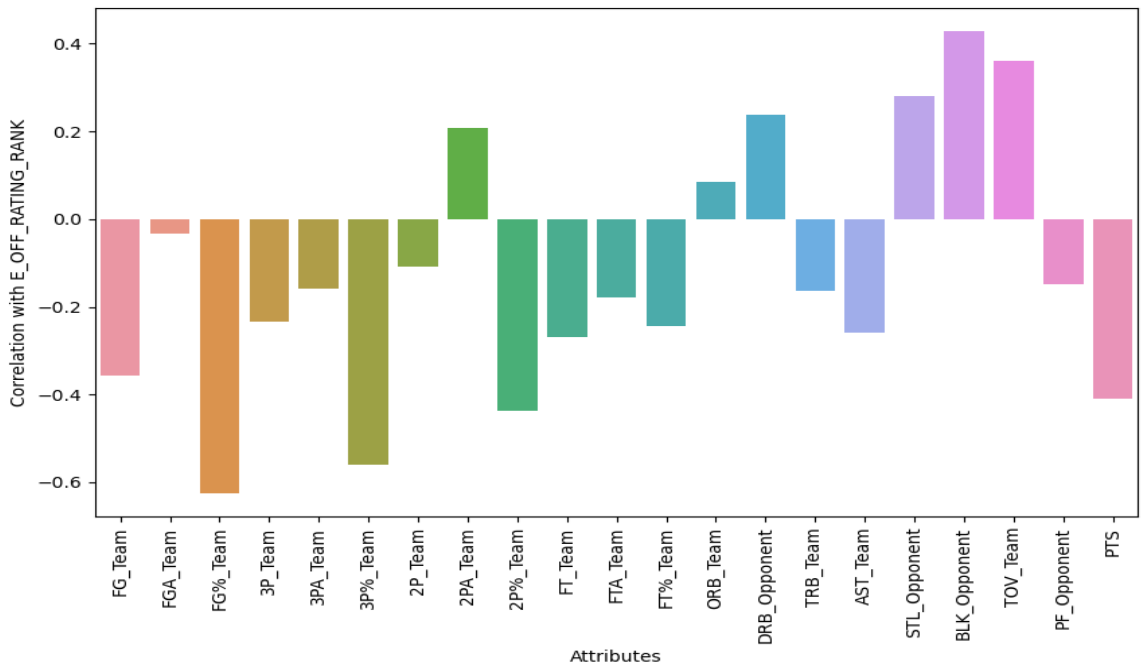


In a YoY champion graph we see that regular season Net Efficiency rating team rank is very telling of post season championship success. In 20 years, only 2 teams won without being in the top 5.

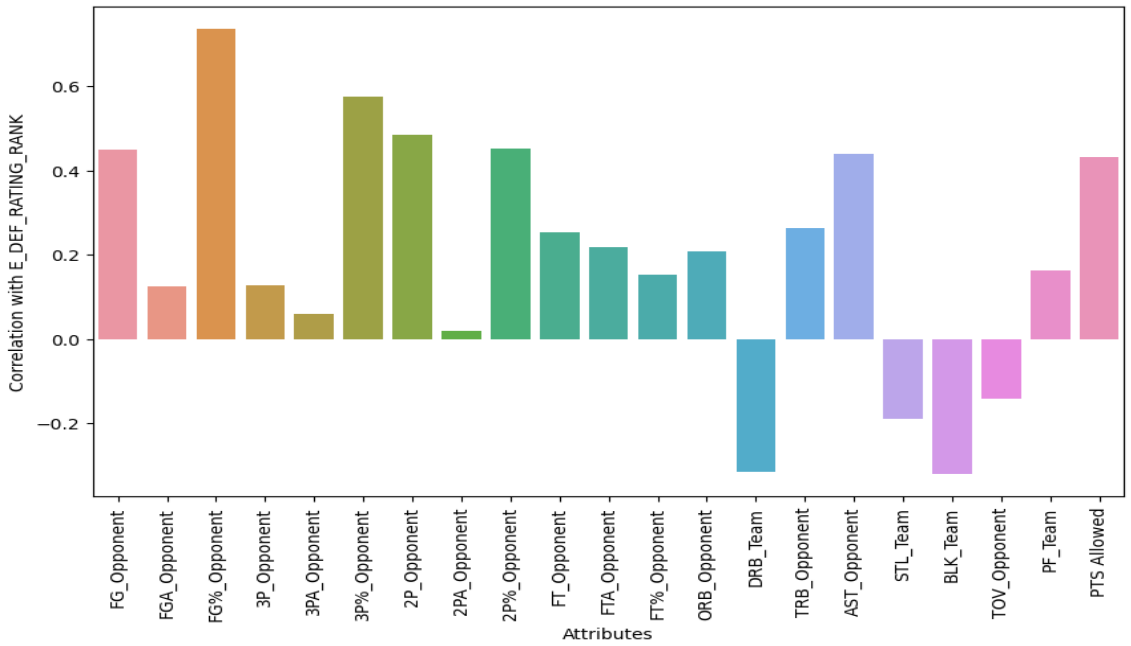


When looking at offensive (green) and defensive (yellow/orange) regular season rankings of champs, we find champ's often have both categories in the top ten and never both rankings outside the top ten. We put a horizontal line in at 15 to show rankings that were in the bottom half of the league.

Analysis/Visualizations: Basketball Reference Attributes impact on Offense and Defense Rank



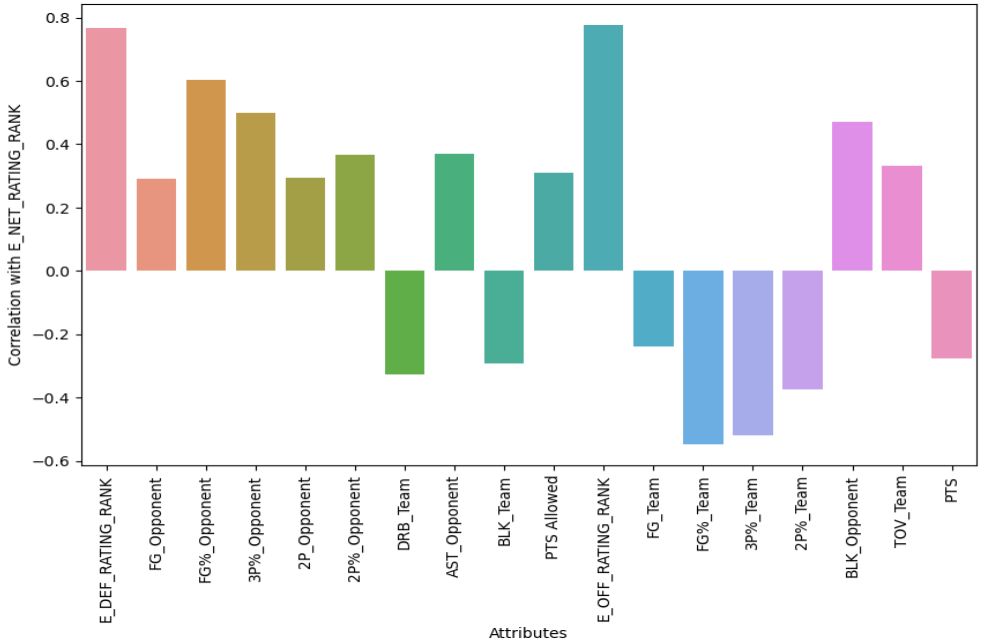
As shown in the chart above the largest indicators of E_OFF_Rating_Rank are the scoring percentages (FG%, 3P%, 2P%, PTS). We see that higher 3 point and field goal percentages mean a lower rank (this is better as it is closer to 1 as it is to 30). Similarly, we see that more team turnovers and blocks against mean a higher rank (this is worse because if you are ranked 30th overall you are last in the NBA).



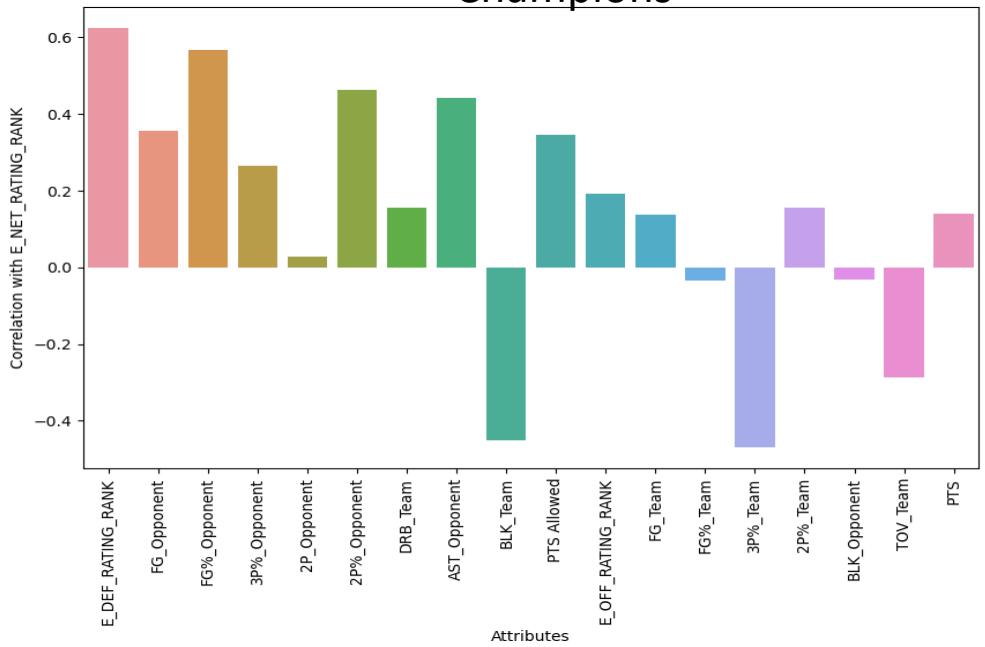
As shown in the chart above the largest indicators of E_DEF_Rating_Rank are the opponent scoring percentages (FG%, 3P%, 2P%, PTS). Here we see the higher the opponent's field goal percentages and points allowed the higher the rank (as in rank closer to 30). Defensive rebounds, steals, blocks and forcing opponent turnovers are highly correlated with a team having a rank closer to 1.

Analysis/Visualizations: Basketball Reference Attributes on Net Rank

Overall



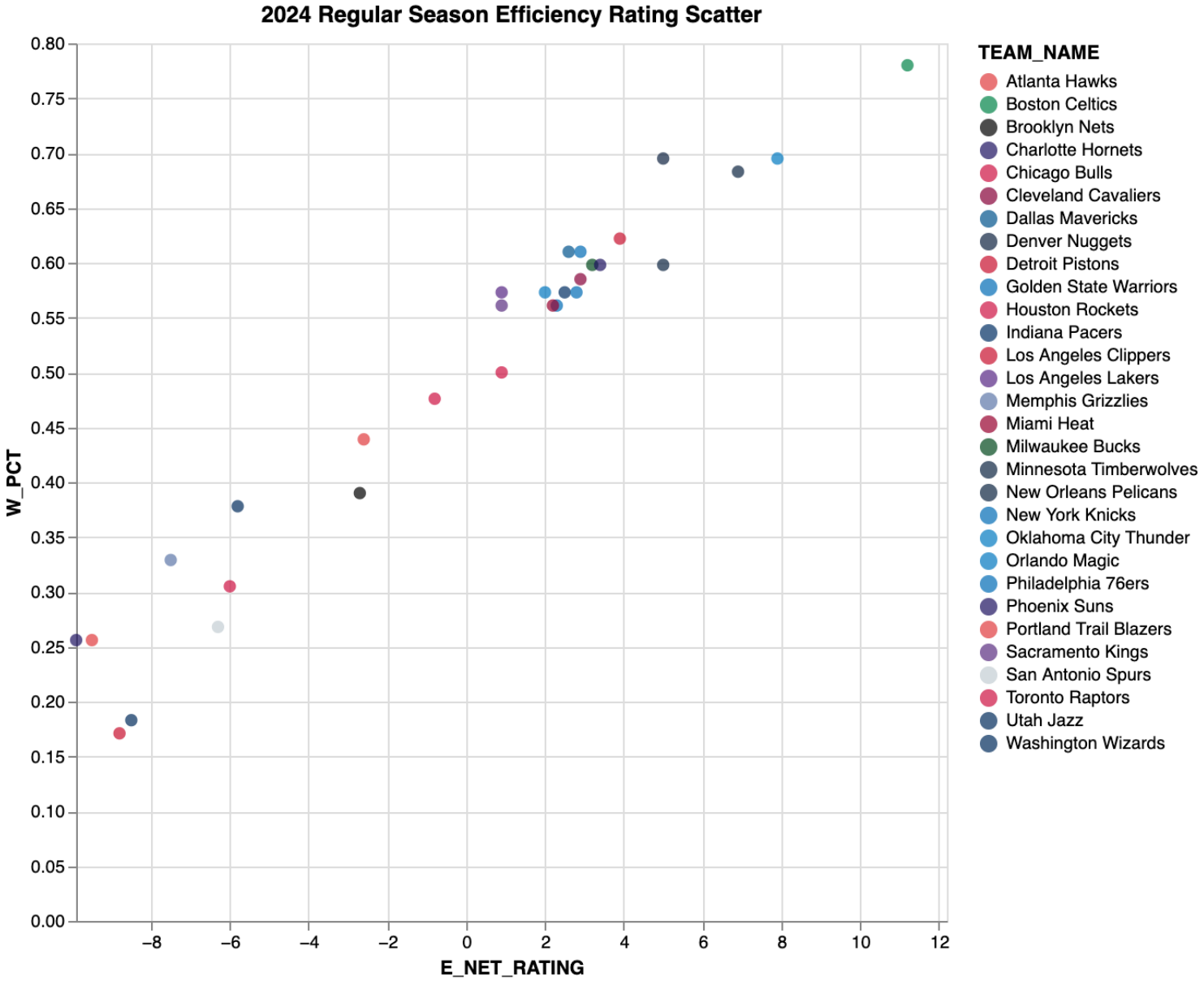
Champions



This chart shows how the most significant indicators from the offensive and defensive rankings correlated with net rank. We see similar trends here with the Team and Opponent FG%/3PT% being the most significant to a lower/better ranking.

When looking at just the champions it is interesting we see that the amount a team blocks and three-point percentage becomes the clear most important indicators in having a lower rank.

Analysis/Visualizations: 2024 Regular Season Efficiency Ranking Scatter Plot



Here we made a scatter plot to show the relationship between Net Efficiency Rating fr om NBA API and a team's win percentage for the 2024 season. We can see from this scatter plot that the relationship is almost perfectly linear which demonstrates that win percentage and net efficiency rating are very positively correlated. Based on the regular season statistics for this season, the Boston Celtics look like they are posed to have a good shot at winning the championship.

Conclusion

We found that Regular season net ranking translated very well to a team winning a championship. The most important factors to building a championship team are as we hypothesized the Three Point Percentage of a Team and limiting the opponents Field Goal Percentage.

Limitations

Boiling teams down to solely stats cannot account for intangibles of having different kinds of players on teams. While it would be nice to be as simple as having the best three-point shooters on a team, this does not factor in having a leader when things are not going right or other things like injuries.

Next Steps

Given what we know about the historical data of champions based on their regular season statistics. We could think about making a machine learning model to help make predictions and forecasts about next season's playoffs and champion once the regular season is complete

Statement of Work

Pulling NBA_API Dat
Brendan found the NBA_API dataset to use and was responsible for figuring out how to pull from the API to get the data in a pandas DataFrame.

Pulling Basketball Reference Data
Michael found the basketball reference dataset to use and worked with BeautifulSoup to figure out how to scrape the data into a pandas DataFrame.

Merging Historical Datasets
Brendan created the historical datasets using the functions established for single seasons. He then merged these along with the champion indicator to get our full dataset. Michael helped to troubleshoot the issues that were encountered with the error of too many requests to the basketball reference websites

NBA_API Visualizations
Michael did exploratory data analysis on the NBA_API dataset to identify that the most correlated variables with winning a championship were net/off/def ranking. He then created the visualizations to show how past champions ranked in those categories.

Basketball Reference Visualizations
Brendan investigated how the basketball reference columns correlated with the three most important variables that Michael identified. He created the correlation charts to show which basketball reference columns were most important in these rankings.

Designing Report
Brendan made the skeleton of the report and then we both took turns filling in slides and respective visualizations. We met over 2 different calls to go over the outline and the designs together and came to agreements on which layouts looked best and how text and graphics should be formatted

Formatting Notebook
Brendan originally put the notebook into Google co-labs and Michael would update him with code changes in slack to avoid overwriting changes the other one is making at the same time. Originally, we started out with all the code somewhat unorganized, but we decided it was best if we wrapped most of it in functions. This was particularly helpful when we had to merge datasets.

Collaboration
We collaborated very well in that we met every week at least twice and then once with our mentor, Kira, each week. In our meetings we did good to help each other whenever one of us was stuck on something and brainstorm how we could fix it. The main issue that we faced was establishing a good working environment that we could use to share code with eachother. One thing that we could do to improve future collaboration would to be to utilize a version control methodology using github.