NBA Analytics

A data-driven approach to exploring basketball trends and outcomes

Authors:

Mridul Gupta (30180121)

Vardaan Bhatia (30181243)

Gavin Lau (10151742)

Prem Patel (30192278)

Borys Protensenko (30168793)

Introduction

Contextualizing Sports and Data

Data and statistics is ubiquitous with sports. Whether it is tracking "Power Play Goals" in Hockey, "First Serve Points Won" in Tennis, or "Height and Reach" in Boxing, they all speak to how integral we believe these metrics to be to sports performances and outcomes. Despite the universality of data in sports in this decade, sports data and the question of how to use this data to inform decisions in sports is a fairly recent development.

The formation of the Society for American Baseball Research (SABR) in 1971 was the first major sports-related institution that recognized the value of statistics in sports (DSG, 2021), but the concept of leveraging these statistics to make decisions around baseball (aptly named SABR-metrics) did not have any public or commercial recognition until the publication of *Moneyball* (Lewis, 2004) and the release of the film with the same name in 2011.

Basketball's own Association for Professional Basketball Research was established in 1997, and subsequent APBR-metrics (a much less phonetically pleasing name) only made its way into the sport in 2004, with Dean Oliver being hired by the Seattle Supersonics, making him the first statistician hired in the NBA. The current "Statistical Player Value" is derived from "Minutes, Points, Rebounds, Assists, Steals, Blocks, and Turnovers" (*Statistical Player Value, SPV - NBAstuffer*, 2022). Other metrics such as "Offensive Rating", "Defensive Rating", and "Player Winning Percentage" (a combination of Offensive and Defensive Rating), have been developed to try to more accurately represent the game of basketball statistically.

In the modern NBA, the use case of statistics is still on the rise and is at its all-time peak in terms of adoption and implementation. This statistics heavy approach to the sport revealed to the 2000's San Antonio Spurs the potential of playing with pace; moving the ball and getting their teammates open for shots. Following their success, other teams tried to implement similar data-driven strategies in their organizations.

More instances of data-informed strategy come up in the sport everyday. Sports analytics in the past few NBA seasons have emphasized the importance of shot selection, that is teams need to take more shots in the restricted area and beyond the three-point line in an attempt to make the mid range shot obsolete. This meta-game strategy ("meta": a term referring to an emergent strategy of the entire sport or game) was fully adopted by Daryl Morey of the Houston Rockets in the 2018 Season when he traded every "Big Man" (physically larger players, traditionally in Center positions) he had for a shorter player who could shoot the 3 pointer.

Being avid fans of the game, getting an opportunity to incorporate our love for the sport with this project made this project engaging and a joy to challenge ourselves to work through problems and surpass our own expectations.Within the scope of this engagement, each of us has explored basketball from an analytical point of view in order to understand how statistics reflect the functioning of a team, the metrics required for winning games and general player related data sources.

Our analysis gives insights into team and player behaviors and some important considerations for a NBA General Manager trying to build a Championship team.

Objectives

The intent of this project is to get better familiarized with sports analytics, being able to correlate basic and advanced NBA statistics with linear regression modeling and building a real time application using our models. Sports organizations are continuously on the lookout for strategies to gain an advantage over their competitors. Nowadays teams that are not utilizing analytics in their organizations workflow are considered outdated and lagging behind.

Within the scope of this project we are trying to better understand the trends in the current NBA season and getting closer to understanding real life applications of our theoretical statistics to realize our goals of being effective Data Scientists in the Sports Field.

Goals & Research Questions

Based on our understanding of the game, we have listed a set of analyses that we want our focus upon in order to analyze and provide suggestions/recommendations for improving the game. To provide direction to our analysis, we have categorized our questions into the three aforementioned categories.

- **Player Analysis** (that is, statistics pertaining to an individual player over their entire career: skills, injuries, personal achievements)
    - How injuries affect player contracts/salaries?
    - Does a player's net statistics affect their in game performance? Does this performance translate to team success?

- **Team Analysis** (composition, strategies, and historic performance)
    - How does playing defensively or offensively impact a team's overall season?
    - Do individual accomplishments accumulate towards winning the NBA season?

- **Meta Analysis** (using the knowledge acquired from our analyses to build an interactive simulation based on linear regression models).
    - What are some key trends we can see in the sport of basketball?
    - Can we use the data we've acquired to predict basketball outcomes (game outcomes, individual outcomes, etc.)?

Finally, using the results obtained in the above three categories, we created a **NBA Fantasy Game Simulator**, which uses the statistical models and insights we obtained in conjunction with the per game statistics for players competing in the 2022-2023 season, to create a fun interactive tool.

1. How do injuries affect player contracts?

Injuries and harsh NBA schedules go hand and hand, this has been the case since the 82 game season was first set in motion. Many players have been affected by the "injury bug" due to this schedule and have lost a lot of money over the years. Players like Derrick Rose, Greg Oden, Brandon Roy and Gordon Hayward have become the posterchildren for injured players in the NBA. Coaches, players and fans have called for the NBA to shorten the season over the past 10 years but to no avail.

Here we are trying to analyze the average amount of money a player loses due to injuries and if the NBPA (National Basketball Player Association) has legitimate evidence to support their plea to shorten the season.

2. Does a player's net statistics affect their in-game performance? Does this performance translate to team success ?

There is a consistent flow of good and bad teams and players each season. While it seems obvious that a player with better net statistics will contribute more to their team's success, this is not necessarily true.

One common trend is that good players in bad teams often "Stat Pad," that is score, pass, rebound, etc. more to make their statistics inflated. Teams often get trapped into thinking a player on a bad team is a star based on these inflated numbers. The purpose of the following is to analyze whether their net statistics is a good determinant of the player's in game statistics.

3. How does playing defensively or offensively impact a team's overall season?

There is a common understanding in the NBA that defense has a bigger impact on the overall team success compared to offense. On multiple occasions commentators and NBA legends like Bill Russel have been quoted saying "Offense wins games, Defense wins Championships."

Due to the rise in three-point shot usage in the league and the development of high paced offense-focused teams, the reliability of the previous quote seems dubious. We want to see if there is still any weight in the late NBA legend's saying or has the three-point, pass-heavy and team-focused game style totally overpowered the modern NBA defense.

4. Do individual accomplishments accumulate towards winning the NBA season?

Continuing with the trend of team success and factors that affect that, we are looking into the individual accomplishments of players who have been awarded one or more of the 5 core NBA awards. These awards include Most Valuable Player (MVP), Defensive Player of the Year, Most Improved Player, Six Man of the Year, and Rookie of the Year.

This question will help us understand the importance of having award winning players on your team and whether these players can lead a team to a successful record or even championships. Intuitively, one might think having a top 5 player or players with many accolades on your team might lead you deep in the NBA playoffs.

This is certainly the case for Milwaukee Bucks star Giannis Antetoukoumpo, who has been to the all star game 6 times, has won three of 5 core awards - MVP(x2), MIP, DPOY, and was the star player of the Bucks when they won the championship in 2021.

Is this trend consistent throughout the NBA or was this an isolated incident?

Datasets

As a team, our goal was to find datasets that contained relevant information to answer our guiding questions. The sources we are utilizing for our datasets are:

- Basketball-Reference - Basketball Statistics and History.
- NBA.com - Official NBA Stats

Data on these websites are provided by 'SPORTS RADAR', (A Sports Technology Company), the official stats partner for the National Basketball Association (NBA).

We are permitted to use the stats on the NBA website under its copyright terms which states that "the NBA Statistics may only be used, displayed or published for legitimate news reporting or private, non-commercial purposes" (National Basketball Association).

Summary of the datasets:

Below is a brief description of each dataset.

Note that columns that have already been explained will not have an explanation in subsequent descriptions of the datasets.

Player Stats Datasets

For player analysis, our dataset includes a player's information such as age, team, game_played, 2-point field goals, 3-point field goals, points scored, blocks, assists, injuries (body part)

playerstats_advanced.csv:

Columns:

- Season (year)
- Player name
- Pos ((Position on team)
- WS (Win Share is calculated to represent the 'contribution' an individual player has to a team's win; on Basketball-Reference, this is approximately 1:1 with the number of team wins.)
- OWS (Offensive Win Share: calculated from points and possessions when a team was on the offense)
- DWS (Defensive Win Share: calculated from the points allowed per defensive possession)
- Age
- Tm (the player's team)
- FG (number of field goals scored; field goals are two-pointers + three-pointers)
- FGA (number of field goals attempted)
- X3P (number of three-pointers scored)
- X3PA (number of three-pointers attempted)
- X2P (number of two-pointers)
- X2PA (number of two-pointers attempted)
- FT (number of free-throws scored)
- FTA (number of free-throws attempted)
- ORB (number of offensive rebounds)
- DRB (number of defensive rebounds)
- TRB (calculated as ORB+DRB)

- AST (assists in scoring Field Goals)
- STL (number of steals)
- BLK (number of blocks)
- TOV (number of turnovers)
- PF (personal fouls)
- PTS (points score by player; the formula is: X2P*2+X3P*3+FT)

Response Variables:

We looked at **Points Scored** (or its constituents, since this is a calculated statistic) as the focal response variable within this dataset, and also combined this dataset with other data to explore additional trends and responses.

---

injuries_stats_contracts.csv

Columns:

- Season
- Player
- Pos
- Age
- Salary (numeric)
- times_injured (a count of injuries)

Response Variables:

We looked at **Salary** as the focal response variable within this dataset, and also combined this dataset with other data to explore additional trends and responses.

Team Stats Datasets

For team analysis, our dataset includes information on a specific NBA team's Wins, Losses, make_Playoffs, Championships won, games_played, Points Per Game, Opponent Points Per Game

win_awards_data.csv

Columns:

- Season (year)
- Team
- playoffs
- W (Wins)
- SRS (Simple Rating System is a player's own stats vs. the counterpart player on the other team while he is on the court.)
- Pace
- Rel.Pace (Relative Pace)
- ORtg (Offensive Rating)
- Rel.ORtg (Relative Offensive Rating)
- DRtg (Defensive Rating)
- Rel.DRtg (Relative Defensive Rating)

- No_of_Awards (Count of Awards won by players on a Team)

---

teamstatscleaned.csv

Columns:

- Season
- Lg
- Team
- W
- L
- WinLossPercentage (Computation of Wins/Loss)
- Finish
- SRS
- Pace
- Rel.Pace
- ORtg
- Rel.ORtg
- DRtg
- Rel.DRtg
- Playoffs

Meta Stats Datasets

For Meta-Game Analysis our datasets includes the following variables that will help us analyze how basketball has changed over the years: season, games, field goal, points_per_game, offensive/defensive rebounds

metadata_join.csv

Columns :

- Season
- DRtg
- ORtg
- Rel_DRtg
- Rel_ORtg
- Pace
- G
- MP
- FG
- FGA
- X3P
- X3PA
- ORB
- DRB

- TRB
- AST
- PTS
- FG.
- X3P.
- ORtg

---

cummalativedata.csv

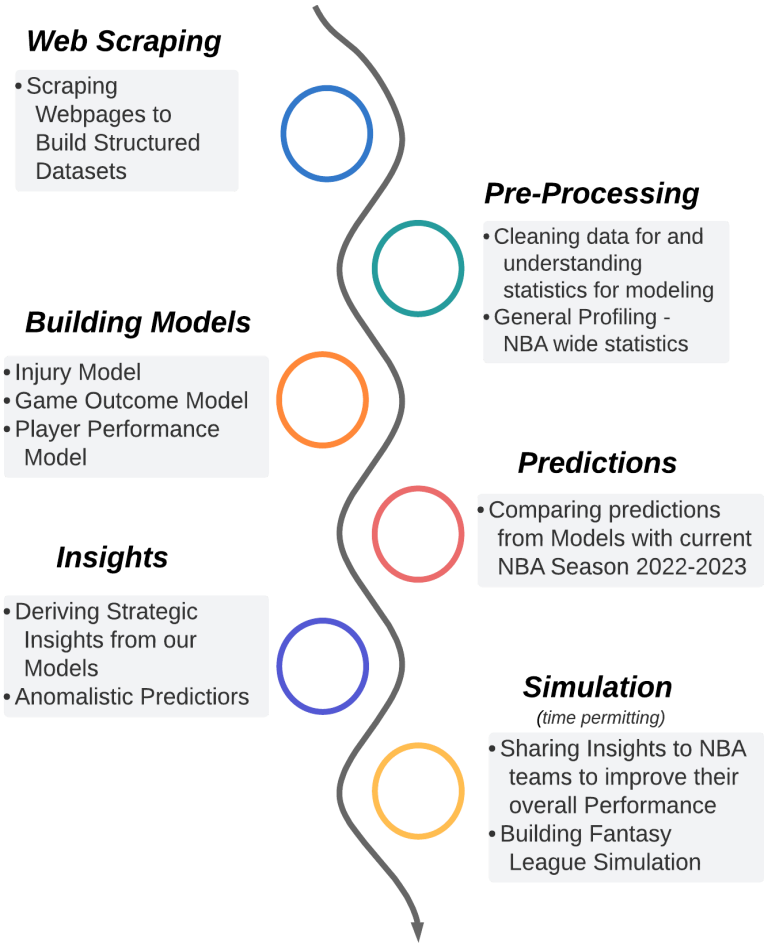
- Date
- Age
- Game.Result
- GS
- MP
- FG
- FGA
- X3P
- X3PA
- FT
- FTA
- ORB
- DRB
- TRB
- ST
- STL
- BLK
- TOV
- PF
- PTS
- GmSc
- PointDifference
- Player
- X2P
- X2PA

The data that is being used for the above three topics has been collected using web scraping. The web scraper in python has been built using two main libraries that are BeautifulSoup4 and urllib. The data available on these websites are mainly built with CSS and HTML, this allows us to parse through the web data in html format for each component required from the webpage.

More information about the web scraping process can be found in the following **Methodology** section.

Methodology

Overview:



1. **Web Scraping**
    a. The data is scraped from the methods defined in **Data Collection.**
    b. Web scraping was a bit challenging to begin with due to the efforts needed to clean the dataset before utilizing it in R or Python. This added to the overall complexity and time required us to get to a point of viable dataset.
2. **Pre-Processing**
    a. Joining the tables to build suitable connections between different datasets
    b. Cleaning the data from Null values and properly using special characters
    c. Removing unnecessary columns before the data file is imported into R for model building
3. **Building Models**
    a. This process includes the determination of models for each of our specified **Guiding Questions**
    b. The process for the model follows this pipeline:
    Multicollinearity Check-> First Order Models-> Interaction Model-> Second Order Model.
    Specific steps of the Statistical workflow can be found in the **Modeling and Predictions** section.
4. **Predictions**
    a. For each of our linear regression models built, we made predictions using data from the 2022-2023 season (the current ongoing NBA season)

   b. This is in the attempt to leverage past seasons' statistics to make informed predictions for this season.
5. **Insights and Visualizations**
   a. An important aspect of data analysis is to be able to understand and present our inference in a feasible and simplistic manner. This part of our process flow will be covered in parts visualizations as well as summarized results based on our inferences..
6. **Simulation**
   a. The final part of our project is the culmination of our work from both DATA603 and DATA604, as a game simulation model which will be hosted (at www.datanerds.lol) for our peers to build their own NBA teams and make real time game simulations against other teams.
   b. The app's backend uses the multiple linear regression models and sampling to predict statistics for each player in the drafted fantasy teams.

Data collection:

All our data was collected via web-scraping scripts written in Python.

Contracts Data for Players playing in 2022-2023 Season:

Webpage Scraped: 2022-23 NBA Player Contracts | Basketball-Reference.com

The following example is one of the simpler scripts that we have built. The code allows us to extract the sub header for contracts by finding all the "tr" attributes for the table-player-contracts. One of our questions deals with player injuries and if there is a relationship between player injuries and the contracts that the said players are currently employed on.

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import pandas as pd

url = "https://www.basketball-reference.com/leagues/NBA_stats_per_game.html"
html = urlopen(url)
soup = BeautifulSoup(html, features="html.parser")
data = soup.find_all('table', id="stats")[0].findAll('tr')
overall= soup.find_all('table', id="stats")[0].findAll('tr')
overall_stats = [[td.getText() for td in data[i].findAll('td')] for i in range(0,len(data))]
overall_stats = overall_stats[2:]

overall_header = [[th.getText() for th in data[i].findAll('th')] for i in range(0,len(data))]
overall_stat = pd.DataFrame(overall_stats,columns = ['Season', 'Lg', 'Age', 'Ht', 'Wt', 'G', 'MP', 'FG', 'FGA', '3P', '3PA', 'FT', 'FTA', 'ORB', 'DRB', 'TRB', 'AST',
'STL', 'BLK', 'TOV', 'PF', 'PTS', 'FG%', '3P%', 'FT%', 'Pace', 'eFG%', 'TOV%', 'ORB%', 'FT/FGA', 'ORtg'])
overall_stat.dropna(inplace = True)
overall_stat.to_csv("C:\Code\stat_overall.csv", index=False)
```

In our SQL joining scripts, this data is joined with injury related data to analyze the trends of how injuries affect a player getting a contract, the contract price, how relevant this observation is and whether there are any outliers to this.

Parsing Per Game Statistics Per Player in the 2022-23 Season:

Webpage Scraped:

a) 2022-23 Boston Celtics Roster and Stats | Basketball-Reference.com

b) Jayson Tatum Stats, Height, Weight, Position, Draft Status and more | Basketball-Reference.com
c) Jayson Tatum 2017-18 Game Log | Basketball-Reference.com

Algorithm

The script does the following:

1. For each team in the NBA, set abbreviation and build the URL. The output looks like (a)
2. Open web page using `urllib` and beautifulsoup, find HREF for the roster div- Getting all players for a team
3. Loop through each player on each team
4. List all seasons the player has played in. For example Jayson Tatum has played in 6 NBA seasons.
5. Build the URL using each seasons code, The output looks like (b)
6. For each season, go to the per game stats and pull the data. The output looks like (c)
7. Store data in csv file in append mode
8. Exit loop

The data is then cleaned, type corrected and split to the most recent season that is 2022-2023. This data is used by us to predict the score per game per player for our simulations. These scores aggregate to a total value which when compared to an opposing team allows us to predict an outcome for a head to head matchup. The data collected from the script below becomes the fundamental basis of the "Game Simulator."

Models similar to the ones we have built can be seen on betting sites like *Fanduels*, *PointBet* and some Fantasy Leagues where each statistic(points, assists, rebounds, steals or blocks) a player gets amounts to some fantasy score. The player with the highest fantasy score after the season ends is declared the winner.

```python
from urllib.request import urlopen
from bs4 import BeautifulSoup
import pandas as pd
import re
import time

teams=["BOS","TOR","PHI","BRK","NYK","MIL","CLE","IND","CHI","DET","ATL","WAS","MIA","ORL","CHO","UTA","DEN","POR","MIN","OKC","PHO","SAC","LAC","GSW","LAL",
       "MEM","NOP","DAL","SAS","HOU"]

for team in teams:
    print(str(team)+" starting...")
    url = "https://www.basketball-reference.com/teams/"+str(team)+"/2023.html"
    html = urlopen(url)
    soup = BeautifulSoup(html, features="html.parser")
    data = soup.find_all('table', id="roster")[0].findAll("td")
    for player in data:
        for a in player.findAll("a",href=True):
            time.sleep(1)
            if "players" in a["href"]:
                url1="https://www.basketball-reference.com"+str(a["href"])
                html = urlopen(url1)
                soup = BeautifulSoup(html, features="html.parser")
                playername=soup.findAll('div',id="meta")[0].findAll("span")[0].text.strip()
                data1 = soup.find_all('div', id="bottom_nav_container")[0]
                print("Scraping "+str(playername)+" data")
                for data in data1.findAll("a"):
                    time.sleep(1)
                    if "gamelog-playoffs" in data.attrs["href"]:
                        continue
                    elif "gamelog" in data.attrs["href"]:
                        url2="https://www.basketball-reference.com"+str(data.attrs["href"])
                        html = urlopen(url2)
                        soup = BeautifulSoup(html, features="html.parser")
                        pergamestats = soup.find_all('table', id="pgl_basic")[0].findAll("tr")
                        pergamestatsdata = [[td.getText() for td in pergamestats[i].findAll('td')] for i in range(0,len(pergamestats))]
                        pergamestatsheader = [[th.getText() for th in pergamestats[i].findAll('th')] for i in range(0,len(pergamestats))]
                        pergamestatsdata = pergamestatsdata[1:]
                        pergamestat = pd.DataFrame(pergamestatsdata, columns = ['G', 'Date', 'Age', 'Tm', '\xa0', 'Opp', '\xa0', 'GS', 'MP', 'FG', 'FGA', 'FG%', '3P',
                        '3PA', '3P%', 'FT', 'FTA', 'FT%', 'ORB', 'DRB', 'TRB', 'AST', 'STL', 'BLK', 'TOV', 'PF', 'PTS', 'GmSc', '+/-'])
                        pergamestat["Player"]=playername
                        pergamestat.to_csv("C:\Code\cummalativePerGame.csv", mode='a', index=False)
                    else:
                        continue
            else:
                continue
    print(str(team)," done...")
```

Parsing individual accomplishment's data since the inception of awards:

Webpage Scraped: [NBA MVP & ABA Most Valuable Player Award Winners | Basketball-Reference.com]

The purpose of getting this data is to understand how often the teams with the so-called "Best Players" actually end up having championship success.

For example, the Boston Celtics went up against the Golden State warriors in the 2022 NBA Finals. The Celtics had the reigning Defensive Player of the Year. However, this award meant little, when Steph Curry from the Warriors averaged 31.1 points across the six championship games and won the title. How relevant these award winners are is clearly relevant from a team building point of view.

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import pandas as pd

awards = ['mvp','roy','dpoy','smoy','mip']
executive_awards=['coy','eoy']

for award in awards:
    url = "https://www.basketball-reference.com/awards/"+str(award)+".html"
    html = urlopen(url)
    soup = BeautifulSoup(html, features="html.parser")
    id = str(award)+"_NBA"
    data = soup.find_all('table', id=id)[0].findAll('tr')
    award_stats = [[td.getText() for td in data[i].findAll('td')] for i in range(0,len(data
    head = [[th.getText() for th in data[i].findAll('th')] for i in range(0,len(data))]
    award_stats=award_stats[2:]
    h = head[1]
    header = h[1:]
    award_stats = pd.DataFrame(award_stats, columns = header)
    award_stats['Award'] = award
    award_stats.to_csv("C:\Code\statsaward.csv", mode='a', index=False)
    print(str(award)," award done...")
```

Parsing advanced statistics from 1980 to 2023:

Webpage Scraped:[2021-22 NBA Player Stats: Advanced | Basketball-Reference.com]

Ever since sports analytics became a popular profession, multiple teams started hiring and tracking advanced statistics for player and team metrics. This includes statistics like how often a player turns over the ball or how much of the team's possessions goes through the said player. Moreover we can look into Player Efficiency Ratings, win share percentages and true shooting percentages as a metric to evaluate the overall confidence in a players skills.

```python
from urllib.request import urlopen
from bs4 import BeautifulSoup
import pandas as pd
import time

years = []

for year in range(1980,2024):
    years.append(year)

for season in years:
    time.sleep(2)
    url = "https://www.basketball-reference.com/leagues/NBA_"+str(season)+"_advanced.html#advanced_stats::per"
    html = urlopen(url)
    soup = BeautifulSoup(html, features="html.parser")
    data = soup.find_all('table', id="advanced_stats")[0].findAll('tr')

    player_stats = [[td.getText() for td in data[i].findAll('td')] for i in range(0,len(data))]
    player_header = [[th.getText() for th in data[i].findAll('th')] for i in range(0,len(data))]
    player_stats = player_stats[1:]
    player_stat = pd.DataFrame(player_stats,columns = ['Player', 'Pos', 'Age', 'Tm', 'G', 'MP', 'PER', 'TS%', '3PAr', 'FTr', 'ORB%', 'DRB%', 'TRB%', 'AST%',
                                                       'STL%', 'BLK%', 'TOV%', 'USG%', '\xa0', 'OWS', 'DWS', 'WS', 'WS/48', '\xa0', 'OBPM', 'DBPM', 'BPM', 'VORP'])
    player_stat.dropna(inplace = True)
    player_stat['Season'] = season
    player_stat.to_csv("C:\Code\playerstat.csv", mode='a', index=False)
    print(str(season)," season done...")
```

Modeling and Predictions

1. After importing the CSV data into Rstudio, for each guiding question we first plotted the distributions of all all our potential predictors and the respective response variables for each guiding question to check if they follow assumptions of normality.

2. If we suspected any co-linearity between our predictors, we tested for multicollinearity using the Multicollinearity tests learned in class.

3. Then we took the valid predictors and built our first-order model.

4. After building the model we checked the results of the individual t-tests and found predictors that were significant. We used 0.05 as the α level, with the following Hypothesis:

   NULL Hypothesis
   H0:β1=β2=0

   Alternative Hypothesis
   Ha: at least one βi is not zero

5. We then reduced the model to only contain significant first-order predictors.

6. Our next step was to check if there were any interaction terms in our model. If we found any significant interaction terms, again using alpha of 0.05, then we went ahead and added those to create a new reduced interaction model.

7. Finally, we compared our model's adjusted R^2 value to our base model.

8. Then we interpreted those improvements in our results.

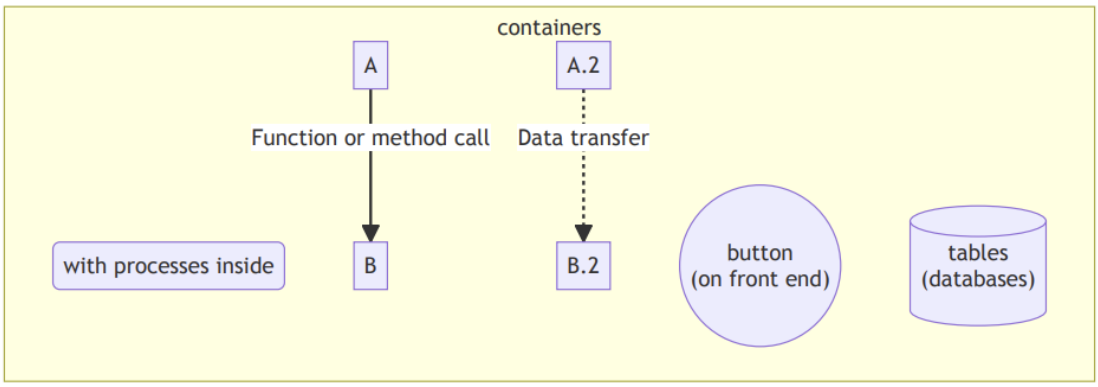
Game Simulator

Overview

The collaboration of the entire group from data collection to model building to the Web Application can be seen in this final component of our project.

Our game simulator is built as an aesthetically simple UI which allows users to build their own "Dream Team", on the similar accords of the Dream Team built by the USA in the 1992 Olympics. The game allows 2 or more users to pick a team from a selection of current players with the 2022-2023 salary cap in mind. The teams built by the players can go head-to-head against each other through the Play button on the UI. The results of the head-to-head matchup is computed through the models built through Linear Regression, providing us with an aggregated value of the net +/- statistic which represents the contribution of each player while they are on court, this value is computed with deviation to introduce randomness into our games similar to that of a real NBA game.

LEGEND for our Web App Diagram



The diagram below signifies the general application flow of our web application for the game simulation. The workflow of our application is divided into three major components.

1. Modeling

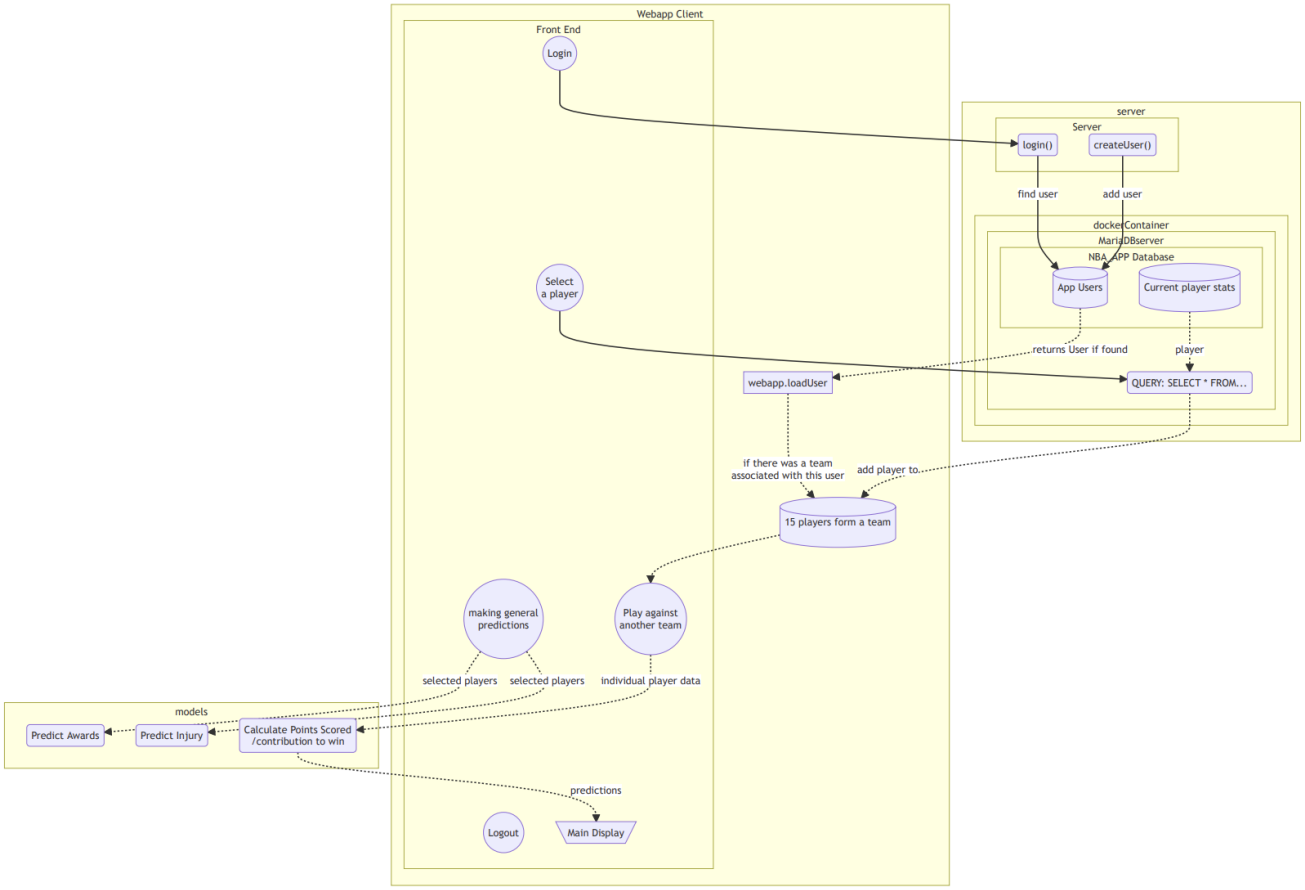
The model is used for the real-time predictions for each of the player statistics that are being computed in the head-to-head model. The model computes two advanced statistics which are the offensive rating per player and the defensive rating per player. This allows us to get a weighted net rating which is also the basis of the head-to-head matchup.

2. Webapp Client

The web app client built with react works as the front end for the user to interact with and to make real-time simulations. The web app has components like player selection for teams, financials, and on-demand game simulation.

3. Server

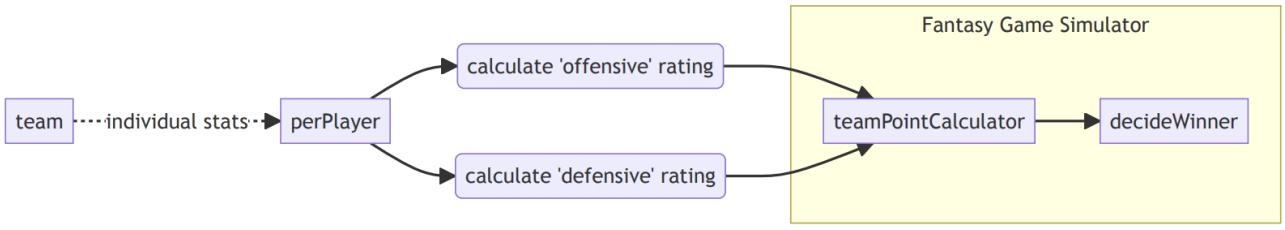
The server is the location where the user login credentials and the details to create a new user and the database is hosted where querying is done on demand based on the specific player selected and the requirement of the current player statistics.



Statistical Model used for predictions in the App:

The model is the component of the web application which is responsible for the processing of the player statistics. The data used in this model can be found under the header "Parsing per game statistics per player for each team," itself has been compiled per game statistics for all current NBA players for the season 2022-2023.

One key aspect of the data is the column "+/-" which is the weighted value of a player's contribution for the overall score. The value becomes the response variable which is used in the model to define offense-related and defense-related prediction models. The value of each player's offense and defense +/- is computed and a total score for the entire team is generated.



Workload Distribution

The workload distribution for the project was equally distributed based on the team members skill set and the skills they wanted to develop, the following were the individual work-load for each of us.

The web scraping scripts were written by Vardaan, and this data was processed by Mridul, Boris and Prem. Boris, Mridul and Prem took the lead on creating the statistical models, with contributions on certain models from Gavin and Vardaan. Gavin and Vardaan took the lead on the development of the web app.

All members contributed to the writing of this report and the final presentation.

Results

How does playing defensively or offensively impact a team's overall season?

This question aims at identifying the correlation between a team's playstyle and the team's success. For this, we start with building models for Defensive and Offensive ratings and then arrive at the final model that involves Wins as the response variables and Offensive and Defensive ratings as the predictor variables.

Sample of Dataset for this question:

Season	DRtg	ORtg	Rel_DRtg	Rel_ORtg	Pace	G	MP	FG	FGA	X3P	X3PA	ORB	DRB	TRB	AST	PTS	FG.	X3P.	ORtg.1
2022-23	111.97667	112.22333	-0.12333333333333328096781	0.12333333333333350356753	99.69333	171	242.5	41.1	88.6	12.2	34.2	10.6	33.3	44.0	25.0	112.6	0.465	0.356	112.0
2021-22	111.95000	111.97333	-0.05000000000000000971445	-0.02666666666666706542177	98.22333	1230	241.4	40.6	88.1	12.4	35.2	10.3	34.1	44.5	24.6	110.6	0.461	0.354	112.0
2020-21	112.33000	112.35333	0.03000000000000001276756	0.05333333333333378389884	99.18000	1080	241.4	41.2	88.4	12.7	34.6	9.8	34.5	44.3	24.8	112.1	0.466	0.367	112.3
2019-20	110.68000	110.50000	0.08000000000000002942091	-0.10000000000000033306691	100.31000	1059	241.8	40.9	88.8	12.2	34.1	10.1	34.8	44.8	24.4	111.8	0.460	0.358	110.6
2018-19	110.40333	110.40000	0.00333333333333335089452	0.00000000000000000000000	100.03667	1230	241.6	41.1	89.2	11.4	32.0	10.3	34.8	45.2	24.6	111.2	0.461	0.355	110.4

Defensive Rating:

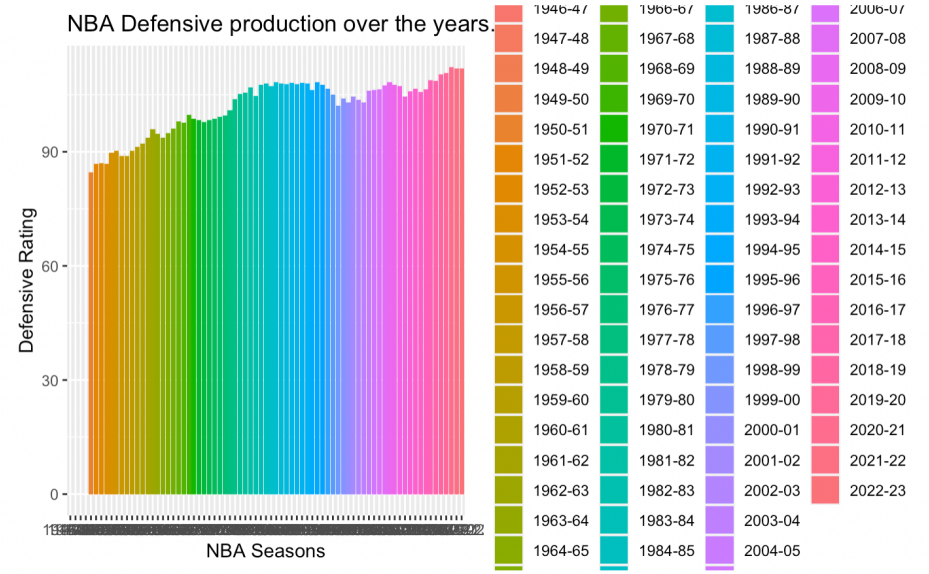
We are going to first predict defensive rating based on the below mentioned predictors as mentioned in **Summary of the datasets**:

- Pace
- FG (Field Goal Percentage)
- FGA (Field Goals Attempted)
- X3PA (Three point shots attempted)
- ORB (Offensive rebounds)
- DRB (Defensive rebounds)
- TRB (Total rebounds)
- AST (Number of assists)

Starting with the initial full model, in the process of improving the R-Squared value, we arrive at the final model which is stated as below:

$$\hat{DefensiveRating} = 82.49960 + 1.06636 * PACE + 1.00602 * FG - 1.12817 * FGA +$$
$$0.59180*\text{X3PA} + 3.34951*\text{ORB} - 0.27441*\text{DRB} - 0.44821*\text{TRB} - 0.63374*\text{AST} - 0.07938*\text{ORB:DRB}$$

The graph depicts an Overview of how the Defensive ratings have changed over the years



Predictions:

```r
defensedata = data.frame(Pace=99.6933, FG=41.1, FGA=88.6, X3PA=34.2, ORB=10.6, DRB=33.3, TRB=44, AST=25)
predict(nba5,defensedata,interval="predict")
```

```
       fit      lwr      upr
1 113.2214 109.7062 116.7366
```

To assess the accuracy of our model, we tried to predict the data for the defensive rating for the Season 2022-23 and obtained a value of 113.22 compared to the original value of 111.98 which gives us a 1% margin of error.

Offensive Rating

Next up, we try to predict the Offensive Ratings based on the same predictors as the **Defensive Rating.**

We started with the initial full model, and in the process of improving the R-Squared value, we arrive at the final model which is stated as below:

$\hat{Offensive Rating} = 119.438663 + 2.073357 * PACE + 1.374430 * FG - 2.233373 * FGA + 0.610792 * X3PA +$
1.395365*ORB-0.606648*DRB-4.691143*AST-0.042626*Pace:FG+0.027061*FG:FGA+0.104209*FG:AST

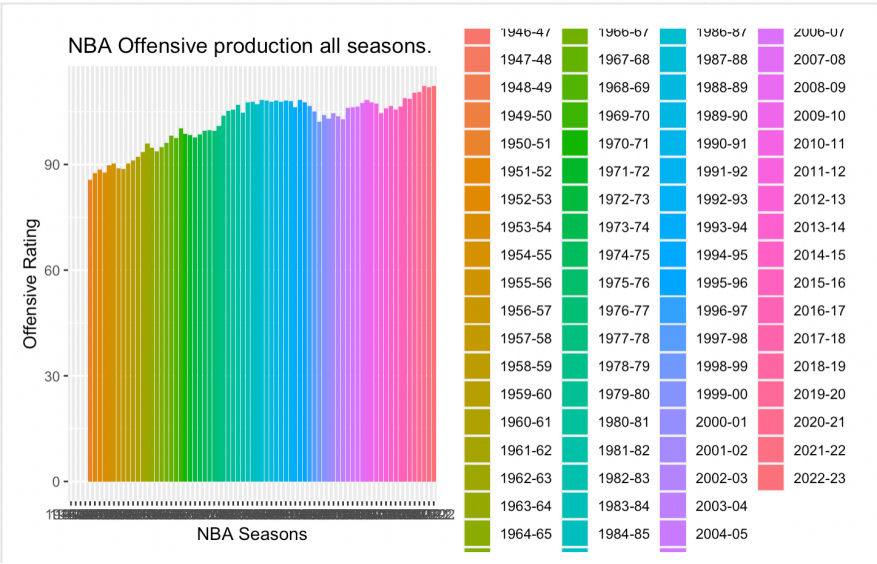
Predictions:

```{r}
offensedata = data.frame(Pace=99.6933, FG=41.1, FGA=88.6, X3PA=34.2, ORB=10.6, DRB=33.3, TRB=44, AST=25)
predict(nba04,offensedata,interval="predict")
```

```
       fit     lwr      upr
1 113.9127 111.213 116.6125
```

To assess the accuracy of our model, we tried to predict the data for the offensive rating for the season 2022-23 and obtained a value of 113.91 compared to the original value of 112.22333 which gives us a 1% margin of error.

The graph depicts an overview of how the Offensive ratings have changed over the years

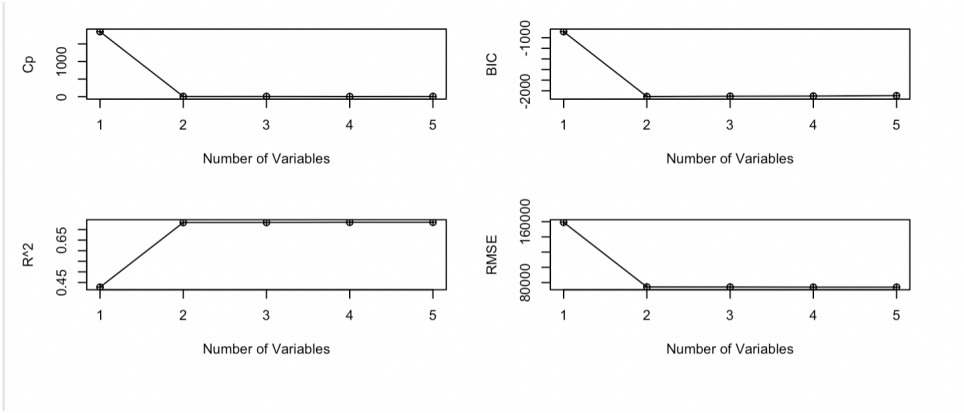


Predict Wins of a Team

Finally, we predict the number of wins of a team in the entire season based on their offensive and defensive ratings.

```{r}
sum= summary(winloss.subset)
rsquare<-c(sum$rsq)
cp<-c(sum$cp)
AdjustedR<-c(sum$adjr2)
RMSE<-c(sum$rss)
BIC<-c(sum$bic)
cbind(rsquare,cp,BIC,RMSE,AdjustedR)
```

```
        rsquare          cp       BIC      RMSE AdjustedR
[1,] 0.4278452 1848.522505  -885.840 159131.60 0.4274901
[2,] 0.7330991    5.878082 -2108.418  74232.31 0.7327675
[3,] 0.7333538    6.339127 -2102.572  74161.48 0.7328566
[4,] 0.7340716    4.001061 -2099.535  73961.82 0.7334101
[5,] 0.7340718    6.000000 -2092.150  73961.77 0.7332444
```



From the plots above we can infer a few things :

- The CP value is exponentially high for the model with a single i.e., the model with only Relative Offensive rating, Also, the CP value is the lowest for the model with 2 variables i.e., model with Offensive and Defensive ratings.
- The RMSE value is at peak at the one variable model keeps decreasing until the model with the maximum variables, this might signify towards actual improvement in the model with 7 variables but that could also be due to 'junk' variables included in the model (since there is inverse relationship with the number of variables)
- From the Adjusted R^2 value we can infer that the value peaks with the model consisting of 4 variables although there is not a huge difference of Adj,R^2 after the chosen model with 2 variables.

From the above we should pick the model with 2 variables since it has the lowest CP and BIC value, additionally there is no significant increase in the Adjusted R2 value after that model.

After finalizing the predictors to be kept to obtain the best possible model, we arrive at the model below:

$$\hat{Wins} = -309.0317 + 5.966942ORtg + 0.9995ORtg + -0.034670(ORtg * DRtg)$$

Predictions:

Results:

```{r}
overalldata = data.frame(ORtg=120.7,DRtg=112.6)
predict(winloss.interactmodel,overalldata,interval="predict")
```

```
       fit      lwr      upr
1 52.53772 39.56189 65.51355
```

The above prediction data is the data available for the current season for the team Boston Celtics. Currently the Celtics are 18-5 making them the best team in the league. Based on the prediction above, we are getting the fitted value of 52.53, while the lower and upper values are 39.56 and 65.51. Based on the season statistics from last season Phoenix Suns had a record of 64-18, based on this inference, the Celtics have the potential to pass that record by 1 win based on our model.

Do individual accomplishments accumulate towards winning the NBA season?

The aim of this question is to determine how individual accomplishments i.e. individual awards accumulate/correlate towards the team winning the NBA season. The response variable for this analysis is the total number of wins which

will be predicted based on the predictor variables.

Dataset for Question 2:

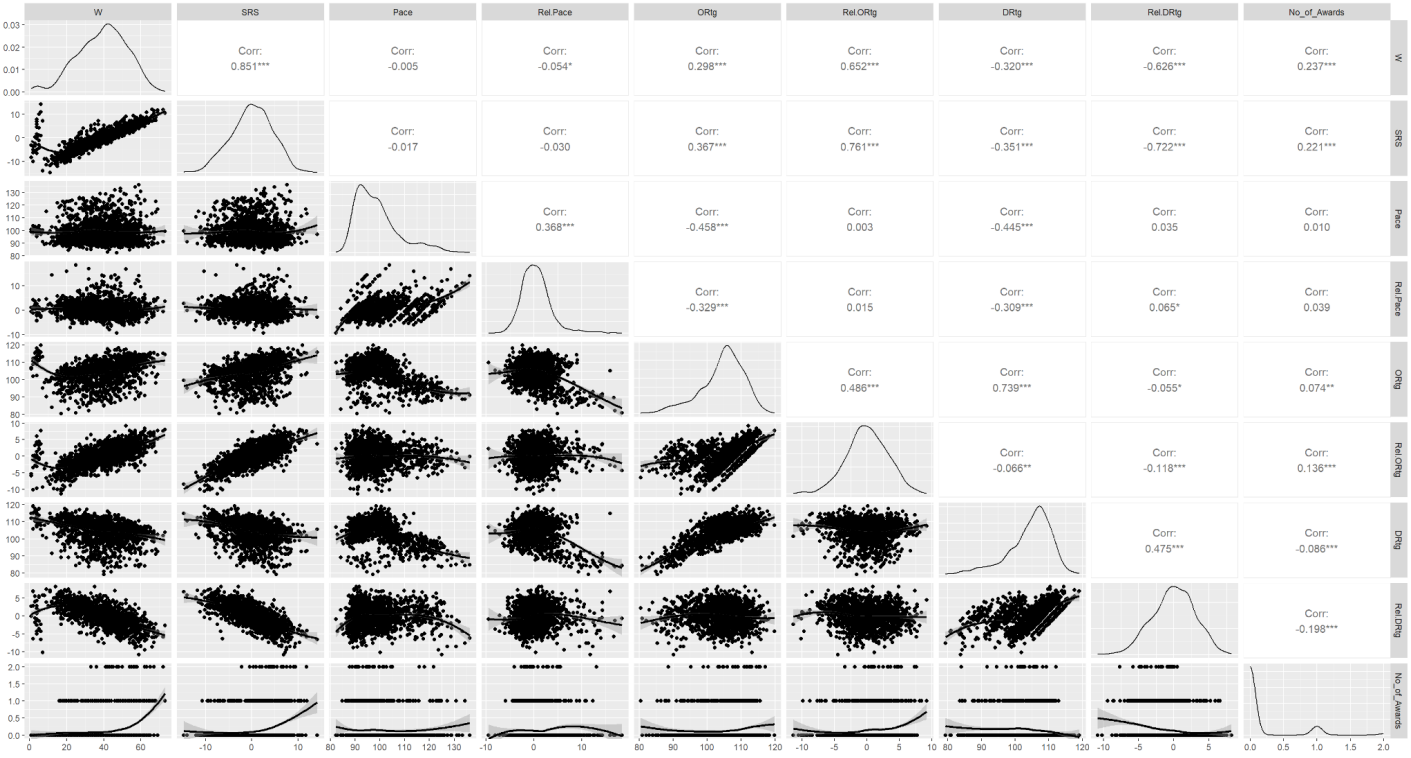
	season	Cleaned_Team	playoffs	W	SRS	Pace	Rel.Pace	ORtg	Rel.ORtg	DRtg	Rel.DRtg	No_of_Awards
1	1951-52	Milwaukee Hawks		17	-7.04	90.7	4.9	80.4	-6.5	89.1	2.2	1
2	1952-53	Fort Wayne Pistons	Lost W. Div. Finals	36	0.17	91.1	10.7	87.6	-0.4	87.7	0.0	1
3	1954-55	Milwaukee Hawks		26	-2.66	100.5	2.4	86.4	-3.4	89.4	-0.4	1
4	1955-56	Rochester Royals		31	-2.61	111.7	8.9	85.3	-5.0	87.9	-2.4	1
5	1955-56	St. Louis Hawks	Lost W. Div. Finals	33	-1.42	109.4	6.6	87.6	-2.7	88.9	-1.4	1
6	1956-57	Boston Celtics	Won Finals	44	4.78	118.0	13.1	88.5	-0.4	84.0	-4.9	2
7	1957-58	Boston Celtics	Lost Finals	49	5.02	124.8	7.8	88.0	-0.8	83.6	-5.2	1

The predictor variables for our model are the team statistics i.e.,

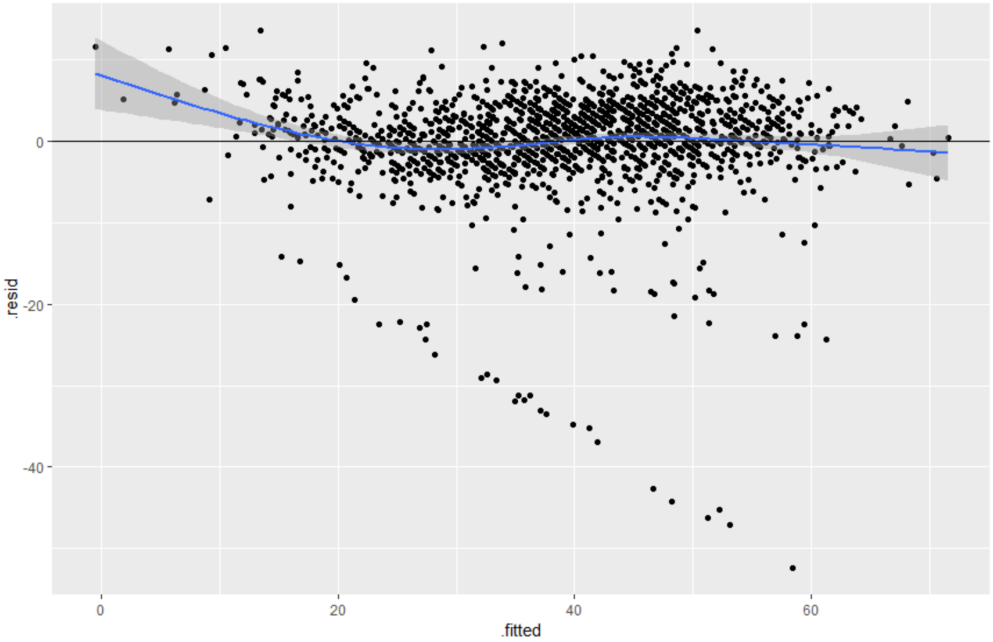
- SRS (Simple Rating System): Team rating statistic that takes into account average point differential and strength of schedule. The rating is denominated in points above/below average, where zero is average.
- Pace: It is an estimate of the number of possessions per 48 minutes by a team.
- Rel.Pace : Relative Pace
- ORtg: Offensive Rating
- Rel.ORtg : Relative Offensive Rating
- DRtg : Defensive Rating
- Rel.DRtg : Relative Defensive Rating
- No_of_Awards : Awards won by that Team

We start with a full model based on the dataset and the predictor and response variables as mentioned above and move towards the final model as stated below:

$$\hat{Wins} = -265.4 + 3.2SRS - 0.85Rel.\,Pace + 4.75ORtg + 1.29DRtg$$
$$+1.39\ \text{No\_of\_Awards} + 3.227\text{SRS:Rel.Pace} - 0.02\text{ORtg:SRS} - 0.02\ \text{ORtg:DRtg}$$
$$+0.0360\text{Rel.Pace:DRtg} - 0.0350\text{Rel.Pace:0Rtg}$$



Residual vs Fitted plot



Since, the residual vs fitted plot indicates that there is a wider spread as we move along the x-axis, this seems to be a case of heteroscedasticity.

We also conducted the multicollinearity test and found out that SRS + ORtg + Rel.ORtg + DRtg + Rel.DRtg + ORtg:Rel.ORtg + ORtg:DRtg + ORtg:Rel.DRtg + Rel.ORtg:DRtg + DRtg:Rel.DRtg are collinear to the response variable i.e., wins.

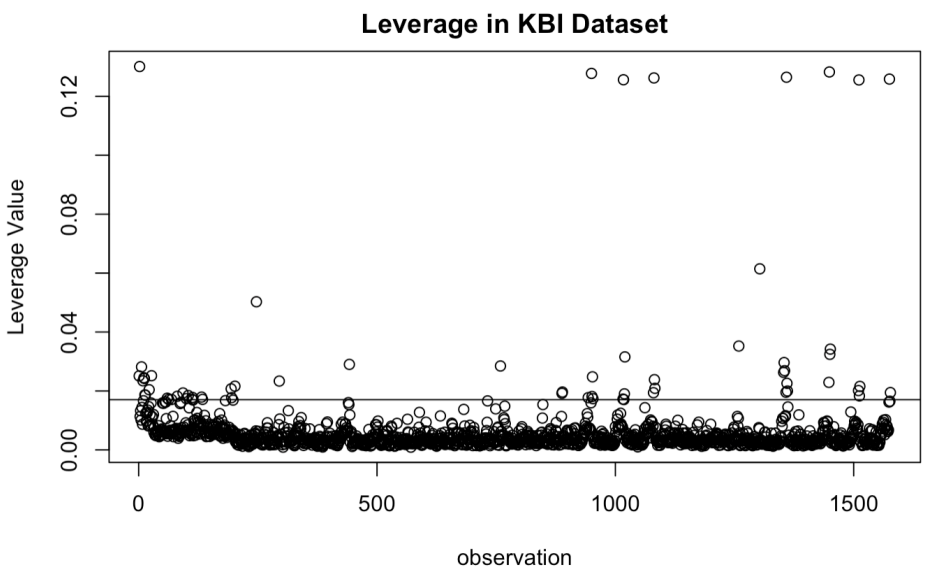
```
 VIF Multicollinearity Diagnostics

                        VIF detection
SRS                   70.3610         1
Rel.Pace               1.4879         0
ORtg              104112.8328         1
Rel.ORtg           26373.3887         1
DRtg               99532.1019         1
Rel.DRtg           22737.0445         1
No_of_Awards           1.0940         0
SRS:Rel.Pace           1.2098         0
ORtg:Rel.ORtg        551.2515         1
ORtg:DRtg           1051.0084         1
ORtg:Rel.DRtg       1276.7998         1
Rel.ORtg:DRtg       1087.4629         1
Rel.ORtg:Rel.DRtg      2.9808         0
DRtg:Rel.DRtg        682.3743         1

Multicollinearity may be due to SRS ORtg Rel.ORtg DRtg Rel.DRtg ORtg:Rel.ORtg ORtg:DRtg ORtg:Rel.DRtg Rel.ORtg:DRtg DRtg:Rel.
DRtg regressors

1 --> COLLINEARITY is detected by the test
0 --> COLLINEARITY is not detected by the test
```

```
[1] "h_I>3p/n, outliers are"
         1          2          6          9         11         12         13         14         22         27         60
0.02509809 0.13010799 0.02817215 0.02344154 0.02447208 0.02423538 0.01868236 0.01855992 0.02042029 0.02517221 0.01754188
        63         69         81         93        100        104        112        113        132        134        194
0.01710039 0.01709579 0.01819561 0.01931585 0.01739858 0.01844647 0.01718234 0.01779275 0.01791181 0.01713555 0.02072408
       195        202        247        295        442        760        889        890        944        951        952
0.01773739 0.02161817 0.05026651 0.02335466 0.02901213 0.02847270 0.01921277 0.01963464 0.01772075 0.12777311 0.01811266
       953        955       1018       1019       1020       1021       1023       1083       1084       1085       1086
0.02479951 0.01737575 0.01723864 0.12561969 0.01707353 0.01901539 0.03155907 0.01940168 0.12623422 0.02383813 0.02089342
      1263       1307       1357       1358       1359       1362       1363       1364       1365       1453       1454
0.03523254 0.06144875 0.02629269 0.02963627 0.02687324 0.01955009 0.12648000 0.02259107 0.01997750 0.02291721 0.12826415
      1455       1457       1516       1517       1518       1519       1582       1584
0.03242616 0.03420158 0.02013215 0.12555800 0.01832482 0.02155558 0.12584506 0.01945336
```



Above is the leverage plot that shows us any possible outliers. Then we removed the outlier using the leverage 'h_I>3p/n'. And made a new datasets with updated values. This resulted in Our model improving from having Adjusted R-squared of 0.73 to 0.77 in our new model. And removing those the two rows with outliers in our dataset helped increase the accuracy of the model. F-statistic has also seen an improvement from 547 to 728 In conclusion we will select our new model in order to have best predictions for our response variable.

Based on this result, we follow the path where we initially build our FIRST-ORDER Model and move towards the

Second Order model and then determine the Final Model chosen.

Predictions:

To test the accuracy of our Final model to predict the stats, we tried to predict the number of games won by the Golden State Warriors in the season 2021-22 where we arrived at a prediction of 52 games won and the actual wins by the team in the season was 53. This gives a 2% margin of error for our model and we can train this model with additional data to improve the accuracy

```
## Prediction
We will be predicting the Games won by Golden State Warriors
```{r}
overalldata = data.frame(SRS=5.52,Rel.Pace=0.2,ORtg=112.5,DRt
predict(win_interaction_reduced_model,overalldata,interval="p
```
```

```
        fit      lwr      upr
1 51.91854 40.23117 63.60591
```

Does a player's net statistics affect their in-game performance? Does this performance translate to team success

The aim of this question is to analyze the player stats, providing insights regarding the player's contribution towards team success? The response variable for this analysis is win shares which will be predicted based on the player's in-game statistics.

Dataset for Question 3

Player	Pos	WS	OWS	DWS	Age	Tm	FG	FGA	X3P	X3PA	X2P	X2PA	FT	FTA	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS
Kareem Abdul-Jabbar	C	14.8	9.5	5.3	32	LAL	10.2	16.9	0.0	0.0	10.2	16.9	4.4	5.8	2.3	8.5	10.8	4.5	1.0	3.4	3.6	2.6	24.8
Tom Abernethy	PF	2.0	1.2	0.8	25	GSW	2.3	4.7	0.0	0.0	2.3	4.7	0.8	1.2	0.9	1.9	2.9	1.3	0.5	0.2	0.6	1.8	5.4
Alvan Adams	C	7.0	3.1	3.9	25	PHO	6.2	11.7	0.0	0.0	6.2	11.6	2.5	3.1	2.1	6.0	8.1	4.3	1.4	0.7	2.9	3.2	14.9
Tiny Archibald	PG	8.9	5.9	2.9	31	BOS	4.8	9.9	0.1	0.2	4.7	9.7	4.5	5.4	0.7	1.7	2.5	8.4	1.3	0.1	3.0	2.7	14.1
Dennis Awtrey	C	0.6	0.1	0.5	31	CHI	1.0	2.3	0.0	0.0	1.0	2.3	1.2	1.9	1.1	3.3	4.4	1.5	0.5	0.6	1.0	2.5	3.3
Gus Bailey	SG	0.2	0.0	0.2	28	WSB	0.8	1.8	0.1	0.1	0.8	1.7	0.3	0.7	0.3	1.1	1.4	1.3	0.4	0.2	0.6	0.9	1.9
James Bailey	PF	1.0	-0.4	1.4	22	SEA	1.8	4.0	0.0	0.0	1.8	4.0	1.0	1.5	1.1	1.9	2.9	0.4	0.3	0.8	1.2	1.7	4.7
Greg Ballard	SF	6.9	4.1	2.8	25	WSB	6.6	13.4	0.2	0.6	6.5	12.9	2.1	2.8	2.9	4.9	7.8	1.9	1.1	0.4	1.6	2.4	15.6

The predictor variables consist of the player statistics which include:

Age : Age of the Player
X3PA : 3-Point Attempts
X3P : 3-point Scored
FGA : Field-Goal Attempts
FG : Field-Goal Scored
ORB : Offensive Rebounds
DRB : Defensive Rebounds
STL : Steals
AST : Assists
BLK : Block
TOV : Turnovers
PF : Personal Fouls
PTS : Points scored
FT : Free throws scored
FTA : Free throws Attempts

Final Model

Our full model has an Adj R^2 value of 0.66 and hence we move forward via an interaction model resulting in an Adj R^2 value of 0.71 based on our final model. The final model chosen has a better Adj R^2 value and hence has a better accuracy of predicting the win shares.

$$\hat{WinShares} = -0.305648 + 0.146916X3PA - 0.665229X3P - 0.906726FG - 0.140737FGA + 0.256993ORB$$
0.153601STL+0.171354AST+0.199997 BLK-0.261396TOV+0.082807PF+0.690488PTS-0.210380FT -0.238074FTA
-0.278108(X3PA*DRB)+0.173305(X3PA*STL) +0.049326 (X3PA*FGA)-0.191485(X3PA*FG) -0.037248(X3PA*AST)+0.465516(X3PA*BLK)
+0.119385(X3PA*TOV)+0.710569(X3P*DRB)+0.027525(X3P*PTS)-1.662771(X3P*BLK)-0.069407(FGA*ORB) -0.234242(FGA*STL)
-0.041602(FGA*AST) -0.084326(FGA*TOV)-0.014324(FGA*FTA) +0.312490(FG*STL) +0.160305(FG*FT) +0.081529(ORB*PTS)
+0.138937 (DRB*STL) +0.027160(DRB*AST)+0.101621(DRB*BLK) -0.098729(DRB*TOV) +0.033443(DRB*PF) 0.051573(STL*AST)-0.254538(STL*BLK)
-0.195162 (STL*PF) + 0.117311(STL*PTS)+ 0.053680(AST*PTS)-0.166985(BLK*TOV) +0.150976(BLK*FT) -0.069980(FT*FTA)

Predictions:

```
#Will Barton 2022
newdata = data.frame(X3P=2.2,X3PA=6.1,FG=5.5,FGA=12.6,ORB=0.6,DRB=4.2,STL=0.8,AST=3.9,BLK= 0.4,TOV=1.8,PF=1.6,PTS=14.7,FT=1.4,FTA=1.8)
predict(reduc_interac_pstatmodel,newdata,interval="predict")
```

```
       fit        lwr      upr
1 2.453918 -0.1959756 5.103811
```

Finally, we used the model chosen to predict the win share value for Will Barton for the season 2022-23. As per the actual stats for 2022-23 season, the win share is 3.2 and the predicted value as per the model comes out to be 2.45. The model predicts with a 22% margin of error and there is scope of improvement in the model as and when there is more data available.

How injuries affect player contracts?

The aim of this question is to understand how past injuries and other player statistics affects the future contracts of the players. For the scope of this question, we try to build a model to predict the player's contracts based on their game statistics and to assess the accuracy of the model, predict contracts for 2022-23 season.

Dataset for Question 4:

Season	Player	Salary	times_inju	WS	Pos	OWS	DWS	Age	FG	FGA	X3P	X3PA	X2P	X2PA	FT	FTA	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS
2022	Aaron Gor	19690909	26	5.2	PF	3.2	2	26	5.8	11.1	1.2	3.5	4.6	7.7	2.3	3.1	1.7	4.2	5.9	2.5	0.6	0.6	1.8	2	15
2022	Al Horforc	26500000	47	7.6	C	3.7	3.8	35	3.9	8.2	1.3	3.8	2.6	4.4	1.2	1.4	1.6	6.1	7.7	3.4	0.7	1.3	0.9	1.9	10.2
2022	Alec Burks	10012800	19	6.1	SG	3.2	2.9	30	3.5	9	1.9	4.8	1.6	4.2	2.7	3.3	0.6	4.3	4.9	3	1	0.3	1.1	2.7	11.7
2022	Alex Carus	9030000	11	2	SG	0.7	1.2	27	2.5	6.2	1	3.1	1.5	3.2	1.4	1.8	0.8	2.8	3.6	4	1.7	0.4	1.4	2.6	7.4
2022	Alex Len	3918600	36	0.9	C	0.4	0.5	28	2.4	4.5	0.2	0.5	2.3	4	1.1	1.6	1.3	2.8	4.1	1.2	0.3	0.6	1.1	2.6	6
2022	Amir Coffe	3395062	1	3.8	SG	2.3	1.5	24	3.1	6.8	1.4	3.7	1.7	3.1	1.5	1.7	0.4	2.5	2.9	1.8	0.6	0.2	0.7	1.3	9
2022	Andre Dru	3200000	12	5.1	C	2	3.1	28	3.4	5.9	0	0	3.4	5.9	1.2	2.2	3.1	6.2	9.3	1.8	1.1	0.9	1.6	2.6	7.9
2022	Andre Igu	1836090	58	1.6	SF	0.5	1.1	38	1.5	3.9	0.5	2.4	0.9	1.5	0.5	0.6	0.7	2.5	3.2	3.7	0.9	0.7	0.9	1.1	4
2022	Andrew W	33616770	12	5.1	SF	1.8	3.4	26	6.5	14	2.2	5.5	4.4	8.5	2	3.2	1.2	3.3	4.5	2.2	1	0.7	1.5	2.2	17.2
2022	Anfernee	22321429	2	1.9	SG	2	-0.1	22	6.2	14	3.1	7.8	3.1	6.2	1.8	2	0.5	2.2	2.6	3.9	0.5	0.1	2	1.9	17.3
2022	Anthony D	37980720	71	4.5	C	2.6	2	28	9.3	17.4	0.3	1.8	8.9	15.6	4.4	6.1	2.7	7.2	9.9	3.1	1.2	2.3	2.1	2.4	23.2

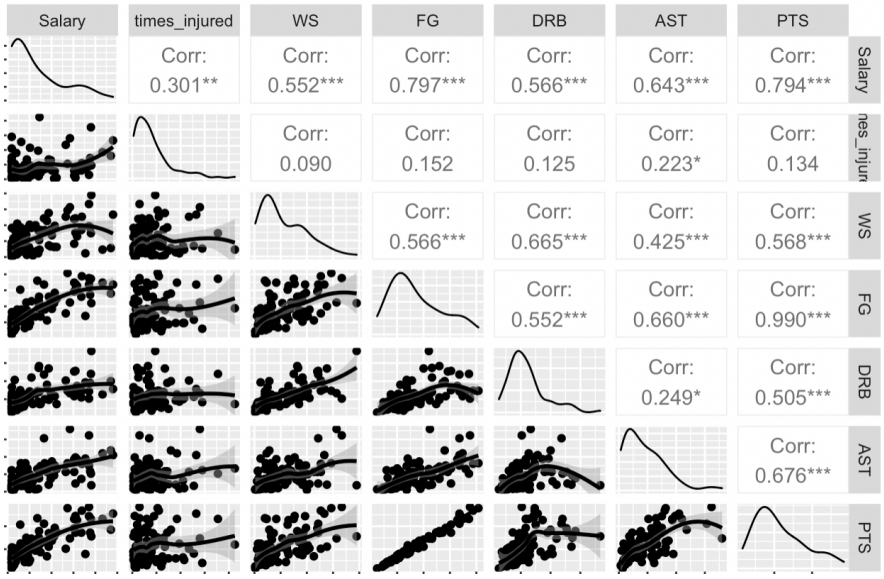
The response variable for our model will be the player salary/contract price information.

The predictor variables are the player stats mentioned below:

- X3PA : 3-Point Attempts
- X3P : 3-point Scored
- FGA : Field-Goal Attempts
- FG : Field-Goal Scored
- ORB : Offensive Rebounds
- DRB : Defensive Rebounds
- STL : Steals
- AST : Assists
- BLK : Block
- TOV : Turnovers
- PF : Personal Fouls
- PTS : Points scored
- FT : Free throws scored
- FTA : Free throws Attempts

The first step of the methodology used involves the use of the "ggpairs" function to determine the co-linearity of the data in detail. The plot determines the spread of residuals values along the range of predictors. We also check the

assumption of equal variance i.e., homoscedasticity and as a result, observed, that some of the residuals are equally spread out in a linear fashion.



Post conducting the multicollinearity test, we found out that X3P+FGA+FG+X2P+FTA+PTS are collinear to the response variable i.e., salary.

```
 VIF Multicollinearity Diagnostics

                    VIF detection
X3P            425.8810         1
X2P           2148.7195         1
WS               3.5654         0
times_injured    1.3858         0
FGA             94.9782         1
FG            3480.9137         1
DRB              3.9572         0
STL              2.3160         0
AST              3.9660         0
BLK              2.8959         0
PTS           1255.9259         1
FTA             57.9323         1
```

Based on this result, we follow the path where we initially build our FIRST-ORDER Model and move towards the Second Order model and then determine the Final Model chosen.

Predictions:

```
Coefficients:
              Estimate Std. Error t value     Pr(>|t|)
(Intercept)   -8526118    1706022  -4.998 0.0000027961 ***
times_injured   114660      44943   2.551      0.01240 *
DRB            1380419     455527   3.030      0.00318 **
AST            1267534     449287   2.821      0.00587 **
FG             2669326     456762   5.844 0.0000000784 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6828000 on 91 degrees of freedom
Multiple R-squared:  0.7137,    Adjusted R-squared:  0.7011
F-statistic: 56.72 on 4 and 91 DF,  p-value: < 0.00000000000000022
```

$$\hat{Salary} = -8526118 + 114660 timesInjured + 1380419 DRB + 1267534 AST + 2669326 FG$$

Game Simulator Model Results:

Final Model

We have to build multiple models for our simulator below is a snippet of a model we used to predict the 2-pointer of an individual player in a simulated game.

$$\begin{aligned}\hat{X2P} = 0.1034 &\text{-}0.0059159\text{Age}\text{-}0.0318536\text{X3PA} + 0.4525856\text{X2PA} + 0.0369033\text{ORB} + 0.0209633\text{DRB} \\ &+ 0.0219890\text{FTA}\text{-}0.0068073\text{AST} + 0.0807074\text{BLK}\text{-}0.0328199\text{TOV} + 0.0426290\text{PF} \\ &\text{-}0.0269191\text{AST:BLK}\text{-}0.0125643\text{FTA:TOV} + 0.0212064\text{ORB:TOV}\text{-}0.0169061\text{ORB:FTA} \\ &\text{-}0.0043944\text{X2PA:AST} + 0.0036997\text{X2PA:FTA} + 0.0040251\text{X2PA:DRB}\text{-}0.0192497\text{X3PA:ORB} \\ &+ 0.0025766\text{Age:X2PA} + 0.017092\text{FTA:BLK} + 0.0070497\text{X3PA:TOV} + 0.0117393\text{AST:TOV}\end{aligned}$$

Conclusion and Discussion

Approach

The approach taken for the course of this project is fairly automated where it can be replicated on various datasets/analysis as the data is acquired by web-scraping, which makes it possible to extract data from any possible website/source with some adjustments. Next up, we have done the pre-processing based on standard practices wherein we clean and process the data using SQL and python (part of DATA 604) and then feed those datasets to the linear regression models on R and compare them against the current NBA season statistics to assess the accuracy and preciseness of the model.

Future Work

The models pertaining to the scope of this project are completely based on linear regression which is kind of the basics of statistical modeling. In terms of the future scope of the project, the first approach will be to apply advanced statistical models and try and predict the response variables with better accuracy and have higher R-squared values for the models. There is also scope of using machine learning models to predict parameters such as predicting the season winner or predicting the award winner players etc.

When talking about predicting season winner, a possible approach can be towards the use of logistic or multinomial regression models as both of them are designed specifically for evaluating categorical and binary answer variables. Logistic regression models can be employed in place of normal linear regression models when the response variable is binary or categorical.

In context to the real time game simulation model as a part of the project, we can also enhance the algorithm at the backend based on advanced modeling and machine learning algorithms. There is also scope to incorporate Fantasy League games as a next step for the simulator.

References

[1] Association for Professional Basketball Research [WWW Document], n.d. URL https://www.apbr.org/ (accessed 11.2.22).

[2] Basketball Statistics & History of Every Team & NBA and WNBA Players [WWW Document], n.d. . Basketball-Reference.com. URL https://www.basketball-reference.com (accessed 11.5.22).

[3] DSG, 2021. The Rise Of Sports Analytics [WWW Document]. URL https://datasportsgroup.com/news-article/74282/the-rise-of-sports-analytics/ (accessed 11.2.22).

[4] Lewis, M. (2004) Moneyball: The Art of Winning an Unfair Game. 1st edition. New York, NY: W. W. Norton & Company.

[5] Official NBA Stats | Stats | NBA.com [WWW Document], n.d. URL https://www.nba.com/stats (accessed 11.5.22).

[6] The Role of Data Science in Sports, 2020. . CORP-MIDS1 (MDS). URL https://www.mastersindatascience.org/resources/big-data-in-sports/ (accessed 11.5.22).

[7] Statistical Player Value, SPV - NBAstuffer (2022). Available at: https://www.nbastuffer.com/analytics101/statistical-player-value-spv/ (accessed: 7 November 2022).

[8] www.Researchgate.Net [Online]. Available at: https://www.researchgate.net/publication/332406802_A_systematic_review_of_sports_analytics (Accessed: 7 November 2022).

[9]]Dartmouth Sports Analytics. (2022). *NBA Shot Selection: How Have Players Changed Their Shooting in the Last Twenty-Five Years?* [online] Available at: https://sites.dartmouth.edu/sportsanalytics/2022/01/25/nba-shot-selection-how-have-players-changed-their-shooting-in-the-last-twenty-five-years/#:~:text=For%20a%20player%20deciding%20between%20the%20mid-range%20and [Accessed 5 Dec. 2022].

[10] Serrano, S. (2018). *Daryl Morey's 'Small Ball' Musical Is Very Good and Very Weird*. [online] The Ringer. Available at: https://www.theringer.com/nba/2018/4/10/17217304/small-ball-musical-daryl-morey-theater-review-houston-rockets [Accessed 5 Dec. 2022].

[11] D'souza, G. (2020). Web Scraping — Python (Requests and BeautifulSoup). [online] The Startup. Available at: https://medium.com/swlh/web-scraping-python-requests-and-beautifulsoup-45d5f48f5a1 [Accessed 30 Nov. 2022].

[12] Stack Overflow. (n.d.). python - scrape through website with href references. [online] Available at: https://stackoverflow.com/questions/19429126/scrape-through-website-with-href-references#:~:text=If%20you%20need%20to%20scrape%20data%20out%20a [Accessed 30 Nov. 2022].

[13] sports.sites.yale.edu. (n.d.). NBA Model Math | Yale Undergraduate Sports Analytics Group. [online] Available at: https://sports.sites.yale.edu/nba-model-math#:~:text=Model%201%3A%20NBA%20Power%20Rankings%20%28Linear%20Regression%29%20Our [Accessed 30 Nov. 2022].

[14] Annieshieh (2020). NBA Player Salaries Prediction with Linear Regression. [online] Medium. Available at: https://medium.com/analytics-vidhya/nba-player-salaries-prediction-with-linear-regression-2b9028

End of Final Project Report