SI206 Final Project Report
Udoka Nwansi
Zoe Zhang
https://github.com/udokanwa/206FinalProject/tree/main

Project Goals

It's the NBA playoff season. Us the data scrapers, Udoka and Zoe, also cheered along with fans all over the world for their favorite teams. More importantly, we wondered if there was a way to predict a team's success in a season; we wanted to know how the number of All Stars on an NBA team affect the team's regular season performance. In this project, we will be scraping the wikipedia page for NBA all-star to return a list of all-star players. Using the NBA stats API's standings endpoint, we will determine if there is a correlation between the number of NBA all stars in a team and their success in the past four seasons.

Achieved Goals

After a rigorous process of web-scraping, API requests, database creation, CSV curation, database manipulation, and matplotlib, we came to three conclusions and achieved our goal. We were able to create 2 databases with overlapping columns, a CSV of conference wins and losses, two scatter plots of conference win and loss percentages, as well as two histograms of league wins and losses. We concluded through examining our four plots that, one, there is no significant relationship between the percentage of conference or league wins and the number of all star players present in a team. Two, There is a moderate negative relationship between the percentage of conference or league losses and the number of all star players present in a team. Therefore, The number of all star players on an NBA team is largely ineffective at predicting team performance, which answered the initial question that prompted us to take on this project.

Problems Encountered

The first issue we faced was finding an API that provided adequate information that allowed for careful analysis and a variety of graphs. We had several ideas, including comparing flight prices and comparing fast fashion prices, but couldn't find a usable api for these two ideas.

Another issue we had was with the database loading in data in increments of 25 rows at a time. Since we are loading in four seasons' worth of data, we had a hard time figuring out how to request for times with a different query string in a specific increment.

We also encountered the issue of extracting quantitative data to determine which team has had the strongest season. The API is quite comprehensive with many endpoints. After some research and playing around with the code, we decided to use the standings endpoint to determine a team's success. Because conference wins and losses are not percentages like league wins and losses, we had to painstakingly convert them both into percentages.

When it came time to create the graphs, we did have a hard time ideating which graph we wanted to attribute to each element we had retrieved from the api, but we came to a conclusion soon enough to use scatter plots and boxplots. Before long and after some back-and-forth debugging, our project was complete.

CSV File

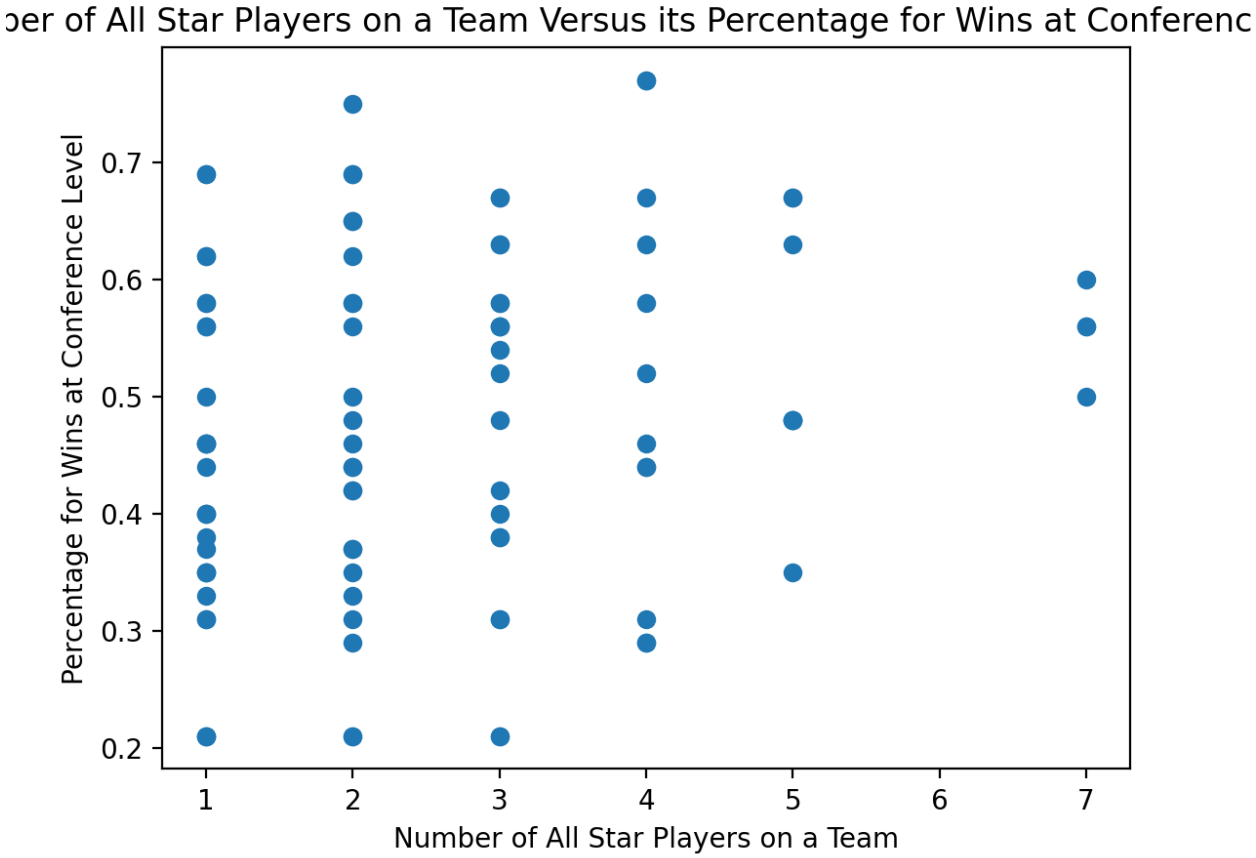
[This is a screenshot of the .csv file in which we put our calculations. It is also attached in the .zip file]

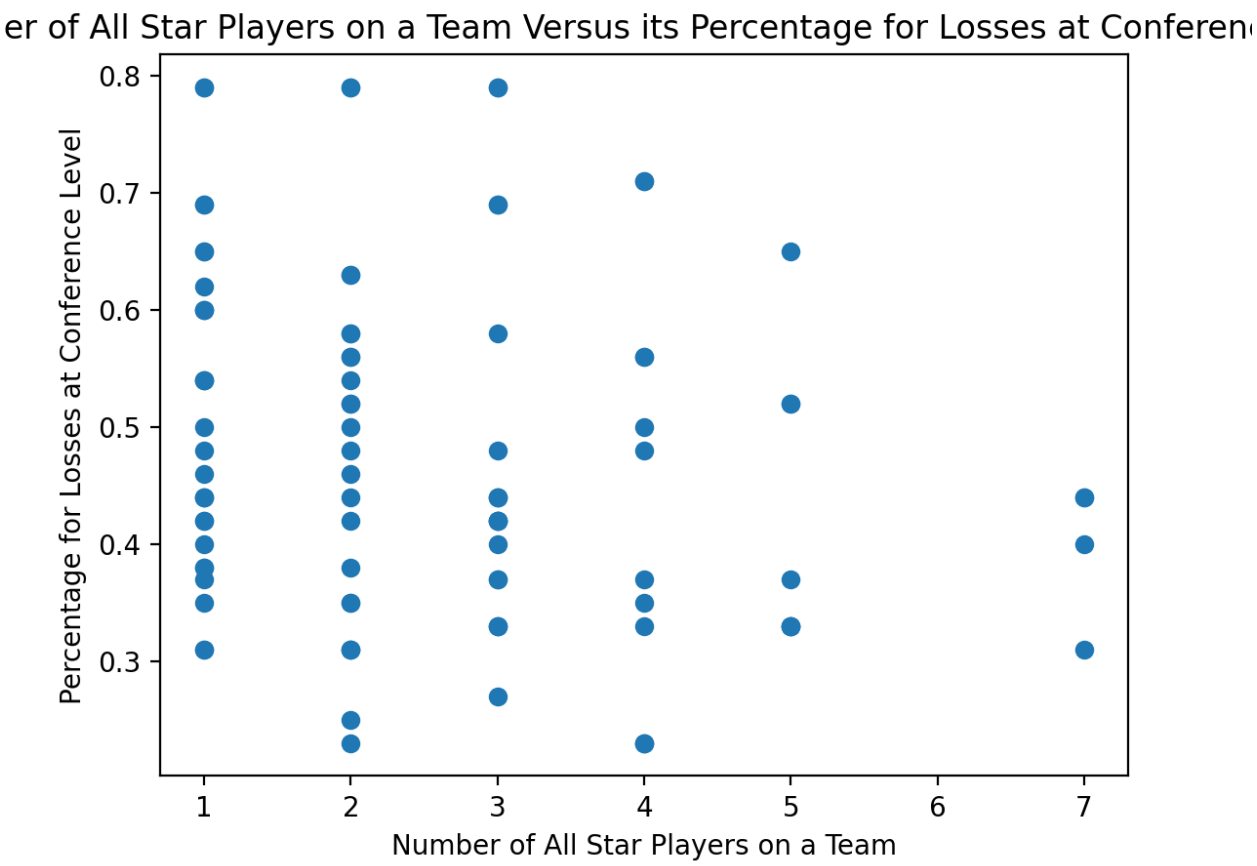
winLossRatio

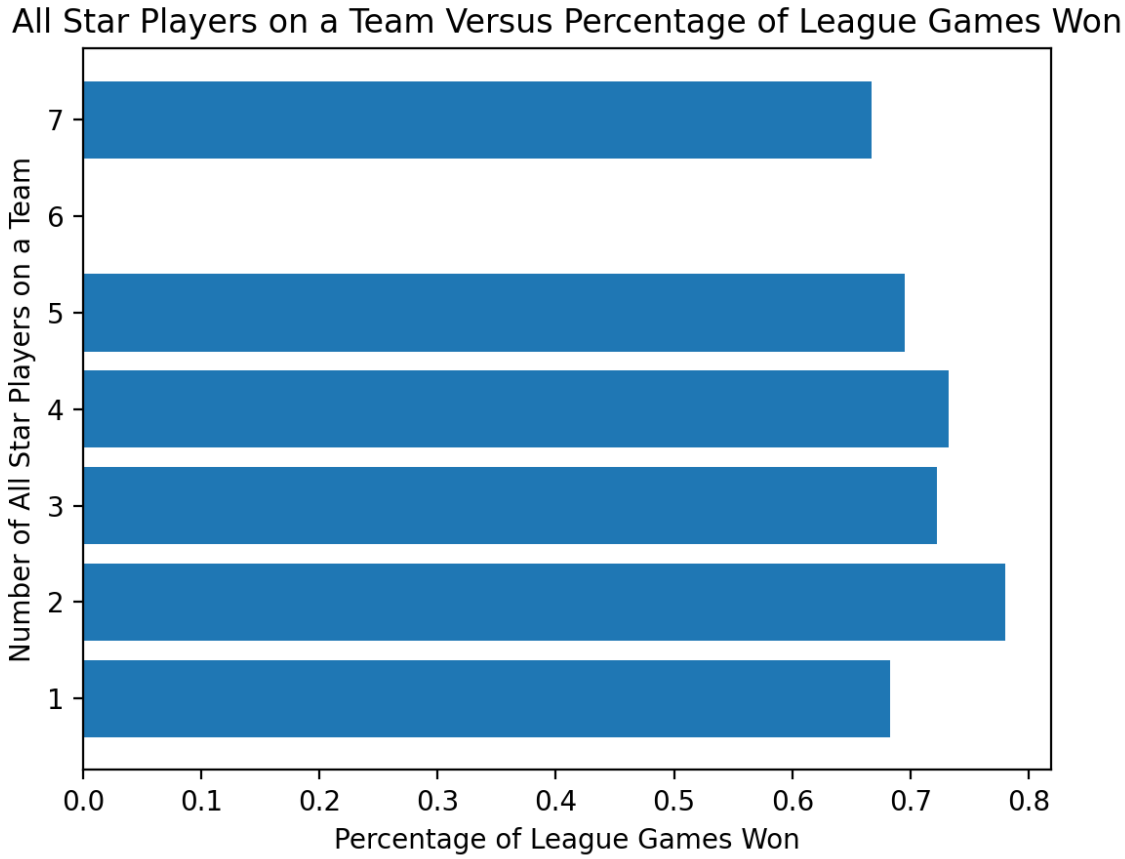
Team Name	Number of All Star Players	Percentage of Conference Wins	Percentage of Conference Losses	Season
Washington Wizards	2	0.46	0.54	2021
Charlotte Hornets	3	0.52	0.48	2021
Miami Heat	4	0.67	0.33	2021
Atlanta Hawks	1	0.5	0.5	2021
New York Knicks	3	0.42	0.58	2021
Brooklyn Nets	7	0.6	0.4	2021
Toronto Raptors	2	0.58	0.42	2021
Boston Celtics	3	0.63	0.37	2021
Chicago Bulls	3	0.56	0.44	2021
Milwaukee Bucks	4	0.63	0.37	2021
Cleveland Cavaliers	4	0.52	0.48	2021
Memphis Grizzlies	1	0.69	0.31	2021
Houston Rockets	1	0.21	0.79	2021
Dallas Mavericks	1	0.69	0.31	2021
San Antonio Spurs	1	0.46	0.54	2021
New Orleans Pelicans	2	0.48	0.52	2021
Golden State Warriors	5	0.63	0.37	2021
Sacramento Kings	1	0.38	0.62	2021
Phoenix Suns	2	0.75	0.25	2021
Los Angeles Lakers	5	0.35	0.65	2021
Utah Jazz	3	0.63	0.37	2021
Minnesota Timberwolves	2	0.62	0.38	2021
Denver Nuggets	2	0.56	0.44	2021
Portland Trail Blazers	1	0.21	0.79	2021
Miami Heat	4	0.46	0.35	2020
Atlanta Hawks	1	0.46	0.35	2020
Charlotte Hornets	3	0.38	0.42	2020
Washington Wizards	2	0.31	0.5	2020
Brooklyn Nets	7	0.5	0.31	2020
Boston Celtics	3	0.38	0.42	2020
New York Knicks	3	0.48	0.33	2020
Toronto Raptors	2	0.33	0.48	2020
Chicago Bulls	3	0.4	0.4	2020
Milwaukee Bucks	4	0.58	0.23	2020

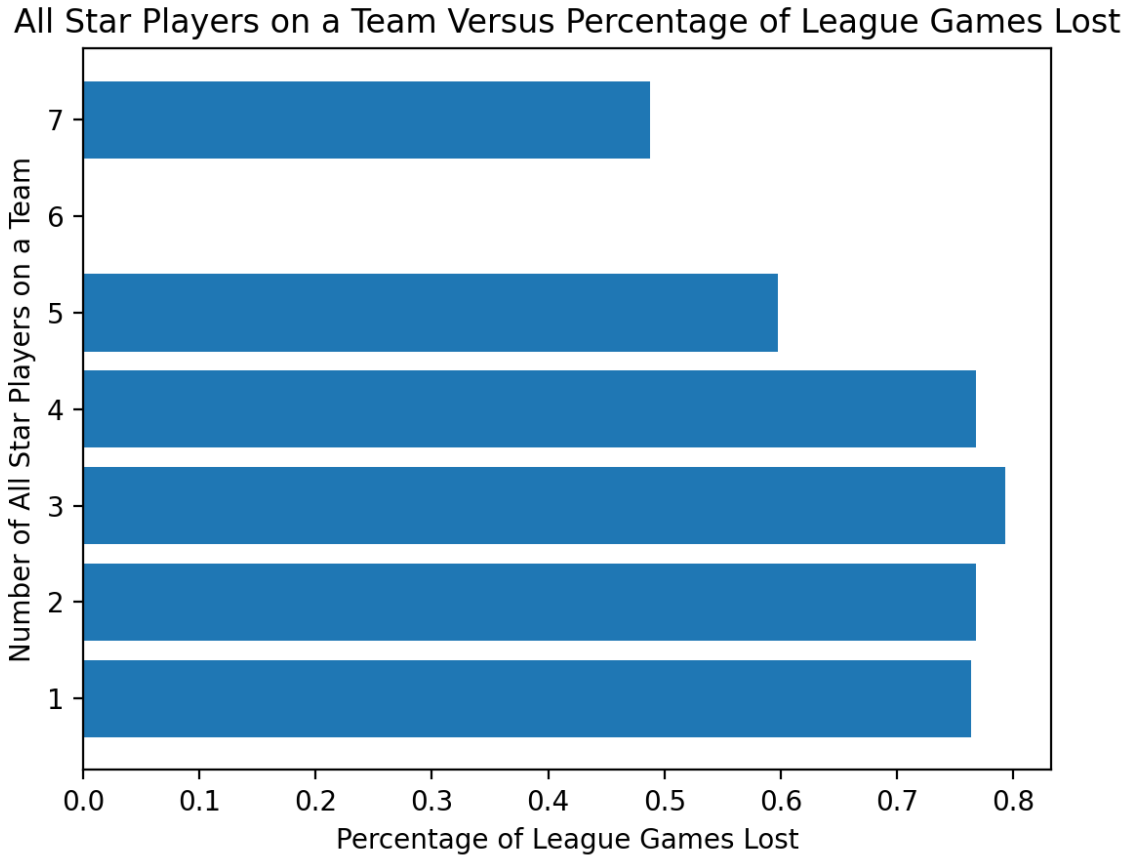
Graphs

[These are all the visualizations that we made for this project]




er of All Star Players on a Team Versus its Percentage for Losses at Conferen
Percentage for Losses at Conference Level
0.8
0.7
0.6
0.5
0.4
0.3
1
2
3
4
5
6
7
Number of All Star Players on a Team


All Star Players on a Team Versus Percentage of League Games Won
Number of All Star Players on a Team
7
6
5
4
3
2
1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Percentage of League Games Won



Instructions for Running the code

Simply run NBA_ALLSTARS.py on your code editor for the entirety of the project. Run the code 6 times for scrape_api_create_database to load all 120 rows of data from the last 4 seasons into the database. The function will use the INSERT OR IGNORE operation in SQL to add 20 items to the database each time the code is run.

Documentation for each function

Function Name	**Description**	**Input**	**Output**
find_all_AllStars()	This function will find all of the current active NBA All Stars in the NBA.	None	player_list: list of 2021-2022's 69 players

count_AllStarTeams(player_list)	This function returns a dictionary with the number of All Stars on each NBA team (if there's more than 0).	player_list	team_dict: key is team name, value is number of all star players on that team
scrape_api_create_database(db_filename, cur, conn)	This function will gather data about each team from the api and create a database from it.	db_filename ,, cur, conn	Doesn't return anything, but creates the Teams database.
create_database(cur, conn)	This function will create a database from the web scraping results for all star players	cur, conn	Doesn't return anything, but creates the allStars database.
calcuate_WLRatio(cur,conn):	This function uses JOIN to select the conference win/loss counts from each team, then calculates them as a percentage	cur,conn	new: a list of tuples. Those items include the percentages of conference wins and losses.
create_CSV(csv_filename, new)	This function takes the list of tuples of W/L percentages and writes the calculated data into a CSV file	csv_filename, new	Doesn't return anything, but creates the csv file.
findLeagueInfo(conn, cur)	This function selects the percentage of league wins and losses from the Teams table.	conn,cur	info: a list of tuples. Those items include the percentages of league wins and losses.
scatter_w(new)	This function will create a scatterplot of percentage conference wins for teams with different numbers of all star players.	new	Doesn't return anything, but creates a scatterplot through matplotlib.
scatter_l(new)	This function will create a scatterplot of percentage conference losses for teams with different numbers of all star players.	new	Doesn't return anything, but creates a scatterplot through matplotlib.
histogram_w(info)	This function will create a histogram of percentage league wins for teams with different numbers of all star players	info	Doesn't return anything, but creates a histogram through matplotlib.

histogram_l(info)	This function will create a histogram of percentage league losses for teams with different numbers of all star players	info	Doesn't return anything, but creates a histogram through matplotlib.
main()	This function calls all the aforementioned ones	none	none

Resources

Date	**Issue Description**	**Location of Resource**	**Result (did it solve the issue?**
April 11th	Difficult to figure out how to scrape the data of **only** the active AllStar players, as opposed to all of the AllStar players in NBA history.	https://en.wikipedia.org/wiki/List_of_NBA_All-Stars	The Wikipedia AllStar table color-coded active players with a blue highlight, so we used tags.find() to choose blue-colored table squares.
April 11th	Trying to find a site with the the entire NBA roster and full team names.	https://basketball.realgm.com/nba/players	This site has the full team name instead of abbreviations, which is how the NBA site is formatted (i.e Portland Trailblazers instead of POR)
April 12th	A lot of different end points to maneuver and win/loss data are not all in percentage form.	https://rapidapi.com/api-sports/api/api-nba	Carefully selected the best endpoint and calculated W/L data into percentages.