
# NBA Stats!

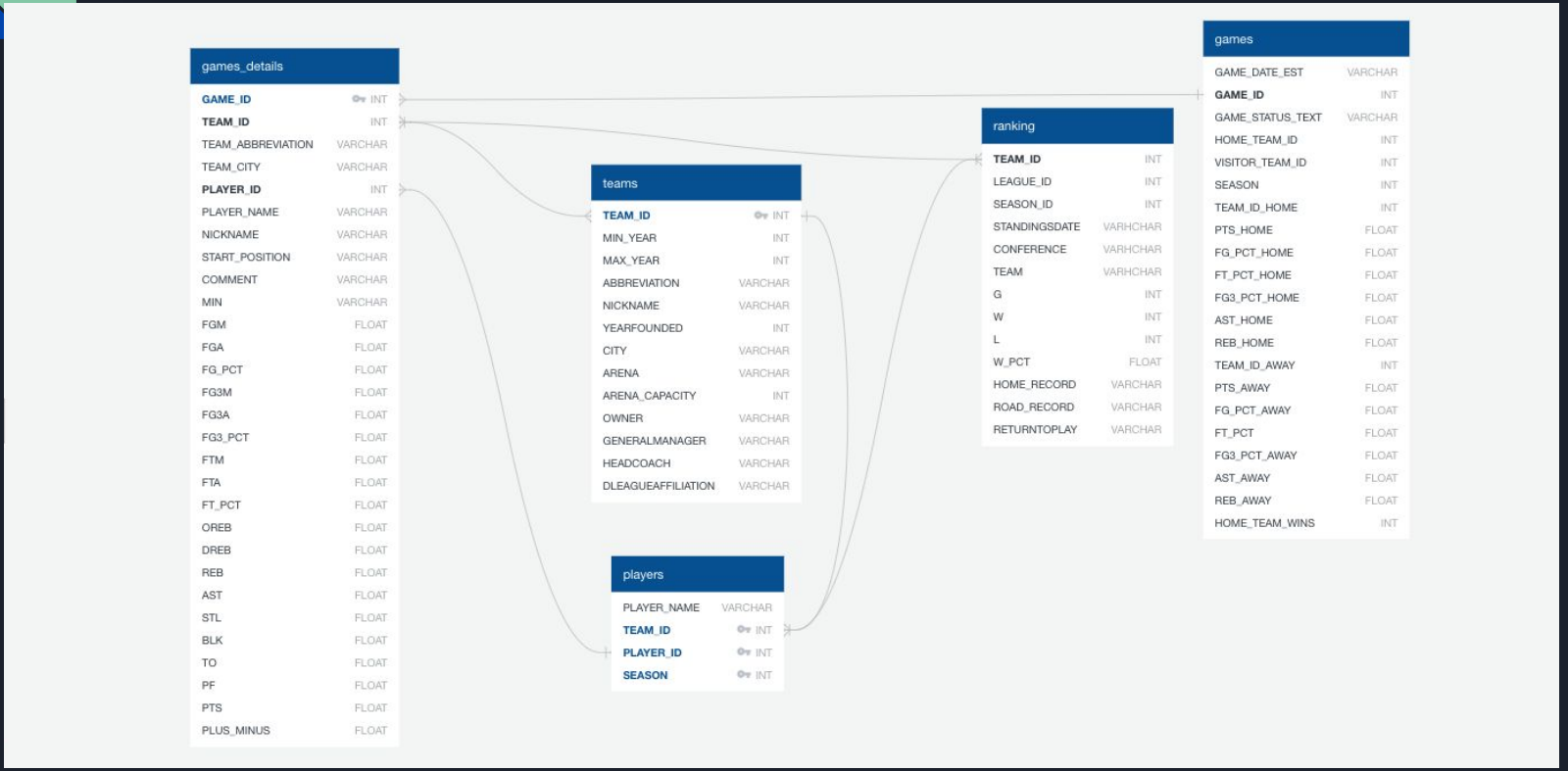
By: Julia, Liz, Brent, and Jerrica

# Challenge Set Up

- Took the Data Engineering Route of Project 3 Challenge
- Discussed what would be interesting and cool to play with
    - Flirted with Liz's Peloton usage data. Found we were missing a lot of data due to added features and metrics over time.
    - Started looking for larger datasets.
- Found a robust data set through Kaggle
- Decided to take on the following challenge:

**Our Goal: Can we programmatically create the ULTIMATE NBA fantasy team?**

# Data Schema


games_details
GAME_ID INT
TEAM_ID INT
TEAM_ABBREVIATION VARCHAR
TEAM_CITY VARCHAR
PLAYER_ID INT
PLAYER_NAME VARCHAR
NICKNAME VARCHAR
START_POSITION VARCHAR
COMMENT VARCHAR
MIN VARCHAR
FGM FLOAT
FGA FLOAT
FG_PCT FLOAT
FG3M FLOAT
FG3A FLOAT
FG3_PCT FLOAT
FTM FLOAT
FTA FLOAT
FT_PCT FLOAT
OREB FLOAT
DREB FLOAT
REB FLOAT
AST FLOAT
STL FLOAT
BLK FLOAT
TO FLOAT
PF FLOAT
PTS FLOAT
PLUS_MINUS FLOAT
teams
TEAM_ID INT
MIN_YEAR INT
MAX_YEAR INT
ABBREVIATION VARCHAR
NICKNAME VARCHAR
YEARFOUNDED INT
CITY VARCHAR
ARENA VARCHAR
ARENA_CAPACITY INT
OWNER VARCHAR
GENERALMANAGER VARCHAR
HEADCOACH VARCHAR
DLEAGUEAFFILIATION VARCHAR
players
PLAYER_NAME VARCHAR
TEAM_ID INT
PLAYER_ID INT
SEASON INT
ranking
TEAM_ID INT
LEAGUE_ID INT
SEASON_ID INT
STANDINGSDATE VARHCHAR
CONFERENCE VARHCHAR
TEAM VARHCHAR
G INT
W INT
L INT
W_PCT FLOAT
HOME_RECORD VARCHAR
ROAD_RECORD VARCHAR
RETURNTOPLAY VARCHAR
games
GAME_DATE_EST VARCHAR
GAME_ID INT
GAME_STATUS_TEXT VARCHAR
HOME_TEAM_ID INT
VISITOR_TEAM_ID INT
SEASON INT
TEAM_ID_HOME INT
PTS_HOME FLOAT
FG_PCT_HOME FLOAT
FT_PCT_HOME FLOAT
FG3_PCT_HOME FLOAT
AST_HOME FLOAT
REB_HOME FLOAT
TEAM_ID_AWAY INT
PTS_AWAY FLOAT
FG_PCT_AWAY FLOAT
FT_PCT FLOAT
FG3_PCT_AWAY FLOAT
AST_AWAY FLOAT
REB_AWAY FLOAT
HOME_TEAM_WINS INT

# Data Cleaning

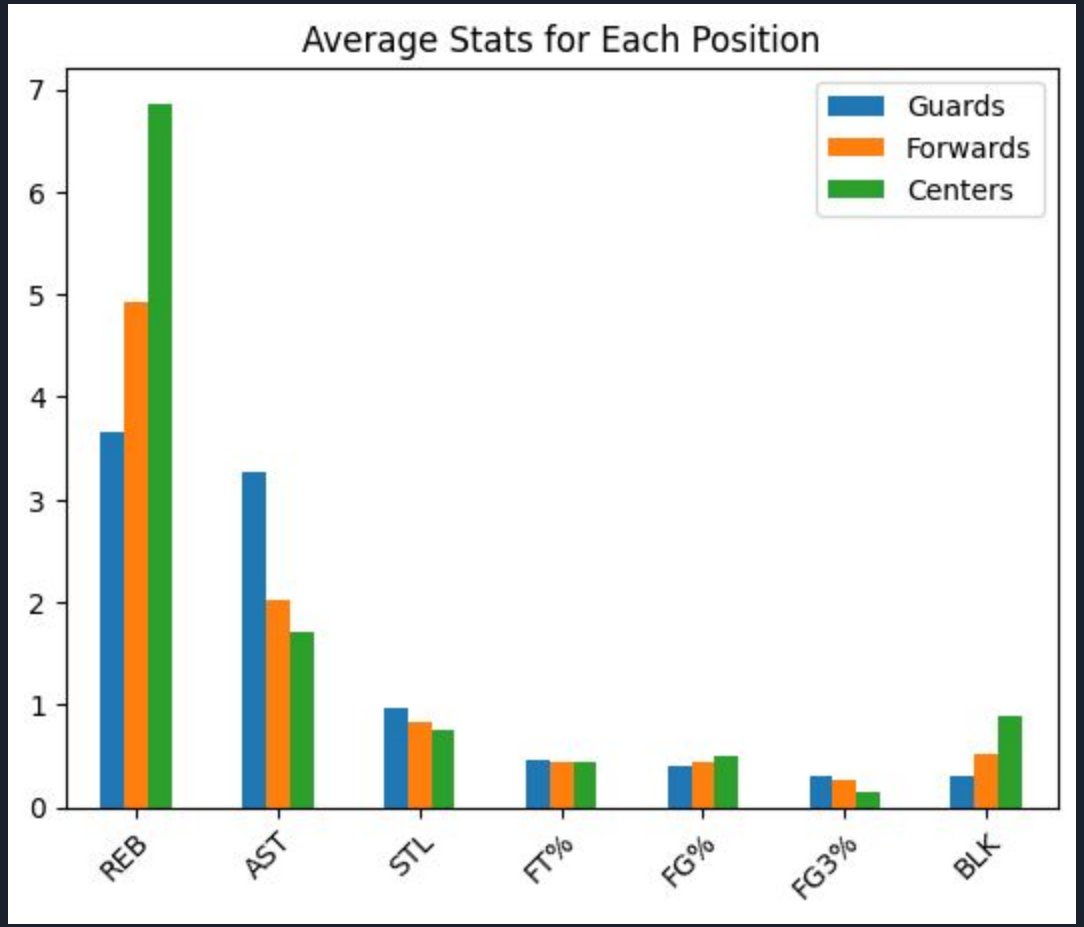
```python
# Pandas
import pandas as pd

#Install psycopg2
import psycopg2

# SQL Alchemy
from sqlalchemy import create_engine
engine = create_engine('postgresql://postgres:postgres@localhost:5433/project_3_nba_stats')
```

✓ 0.0s Python

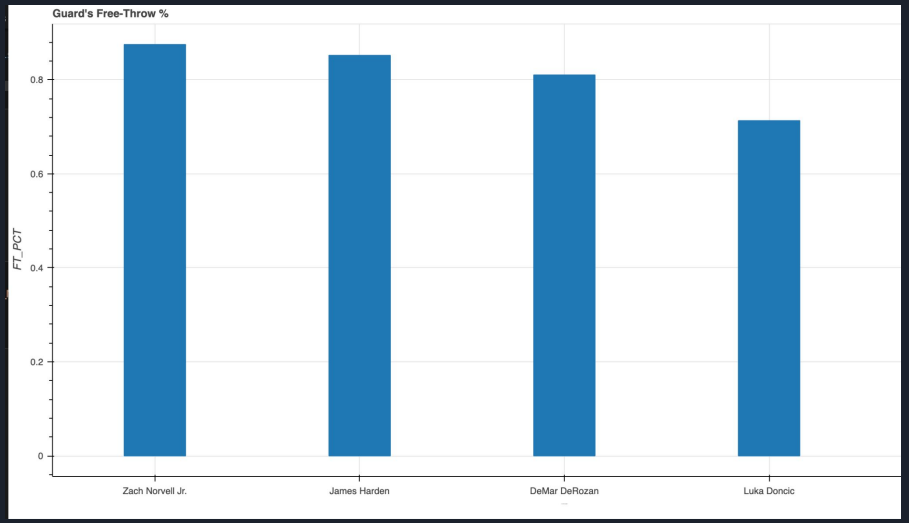
```python
# Dependencies
import pandas as pd
from pathlib import Path

# CSV paths
games_details_path = 'SELECT * FROM games_details'
games_path = 'SELECT * FROM games'
teams_path = 'SELECT * FROM teams'
players_path = 'SELECT * FROM players'
ranking_path = 'SELECT * FROM ranking'

# Creating dataframes
games_details = pd.read_sql_query(games_details_path, con=engine)
games = pd.read_sql_query(games_path, con=engine)
teams = pd.read_sql_query(teams_path, con=engine)
players = pd.read_sql_query(players_path, con=engine)
ranking = pd.read_sql_query(ranking_path, con=engine)
```

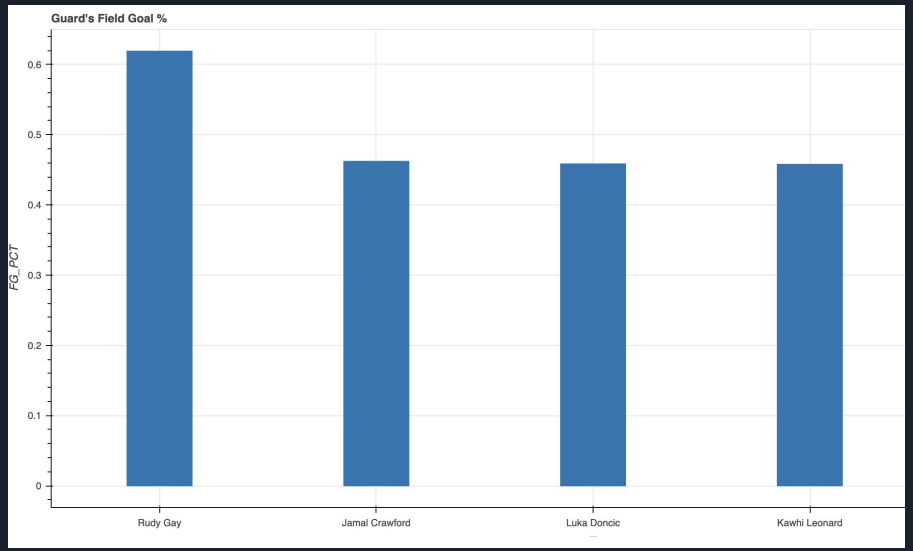
✓ 8.0s

Connected PostgreSQL to VSCode. Exported Cleaned files as new CSVs to further manipulate.

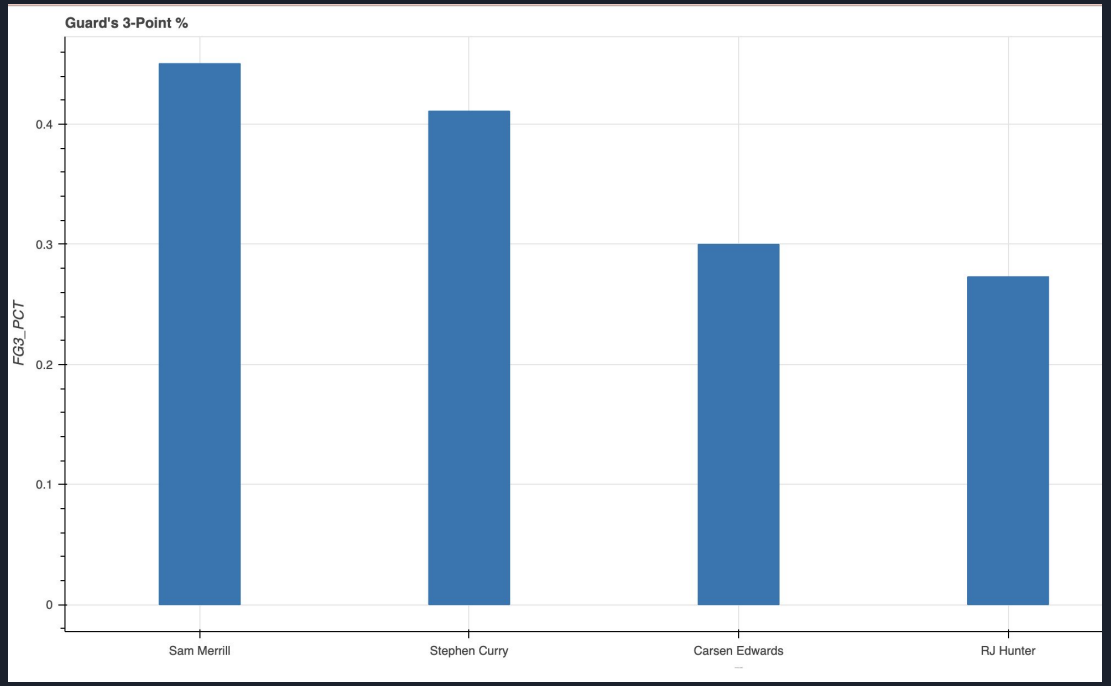

Average Stats for Each Position
Guards
Forwards
Centers
7
6
5
4
3
2
1
0
REB
AST
STL
FT%
FG%
FG3%
BLK

# Guards: Who are the best shooters?

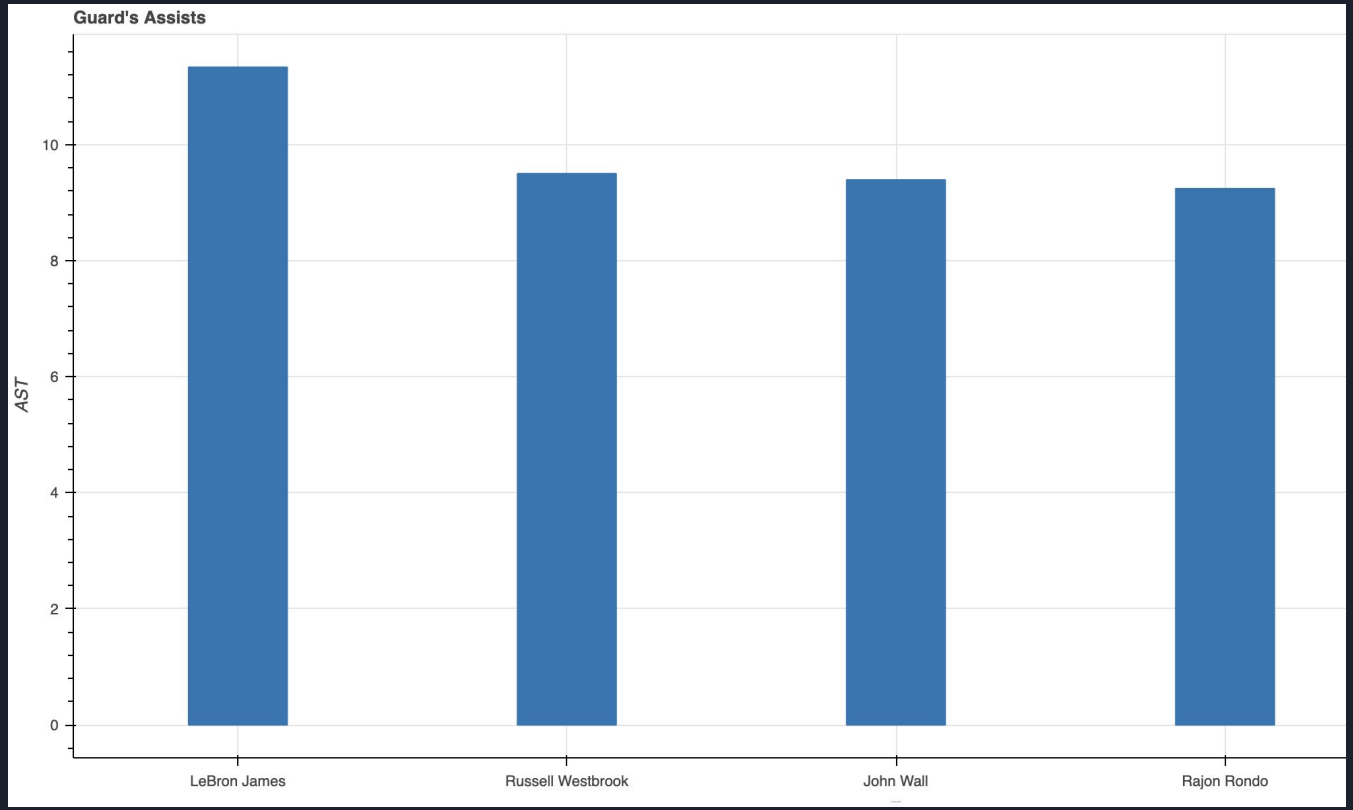


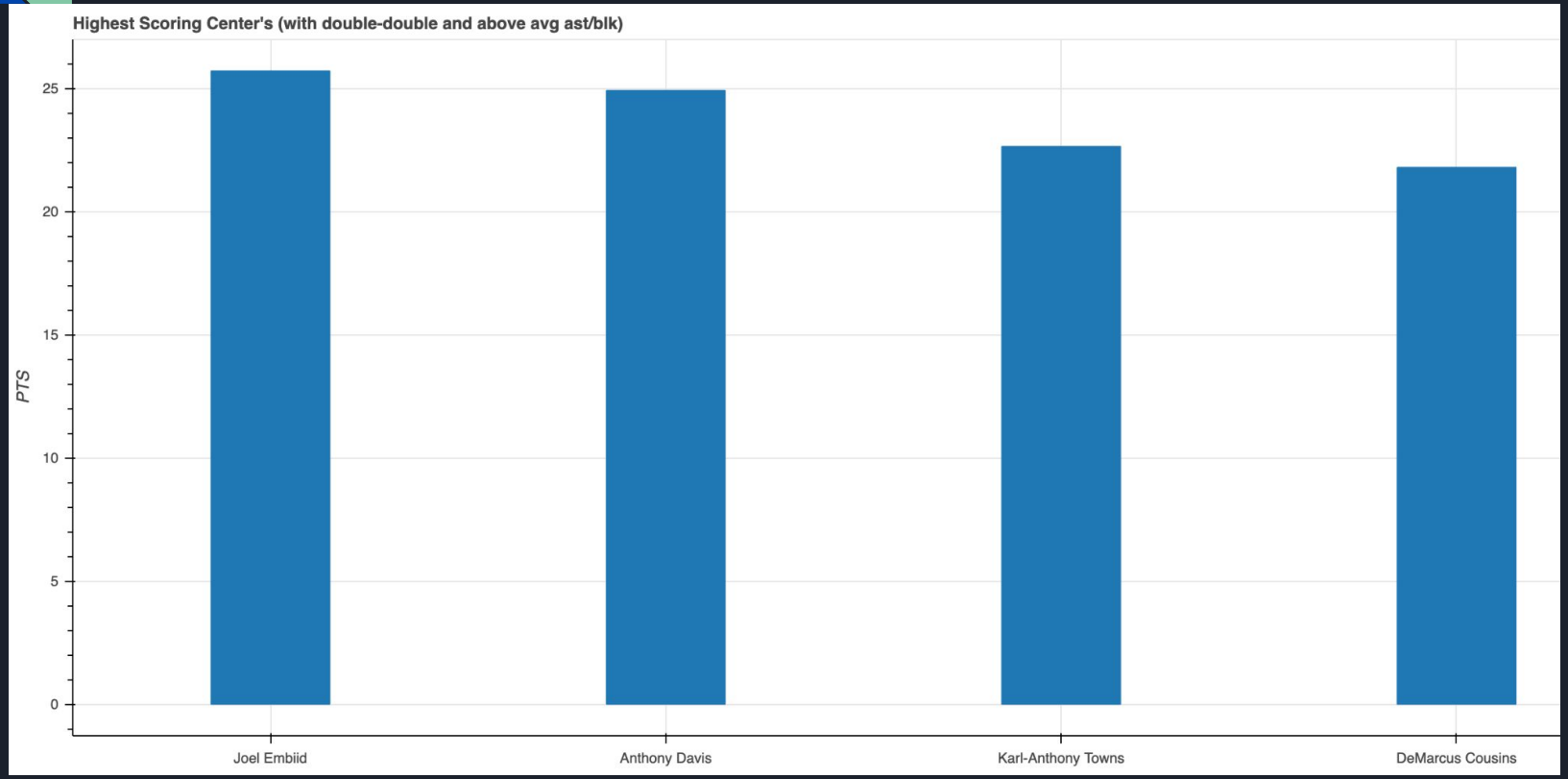
# Guards: Who's the best 3-point shooter?



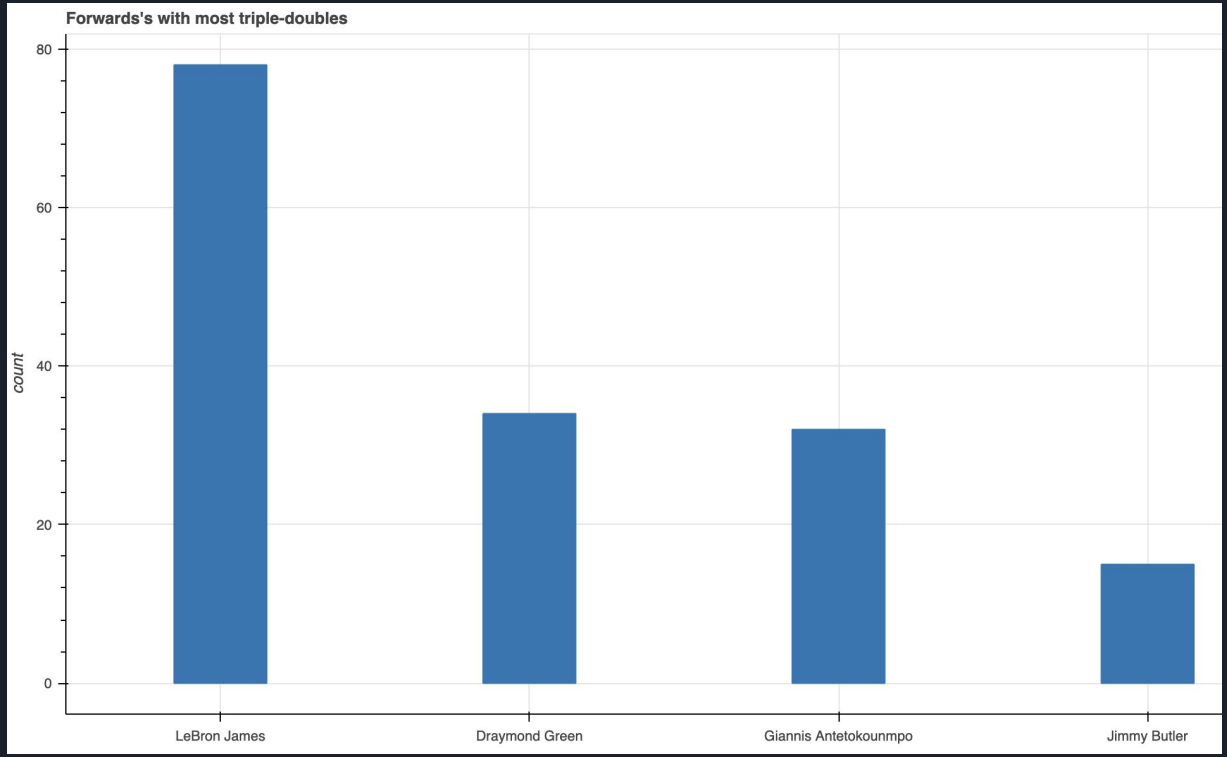
# Guards: Who's a selfless player?



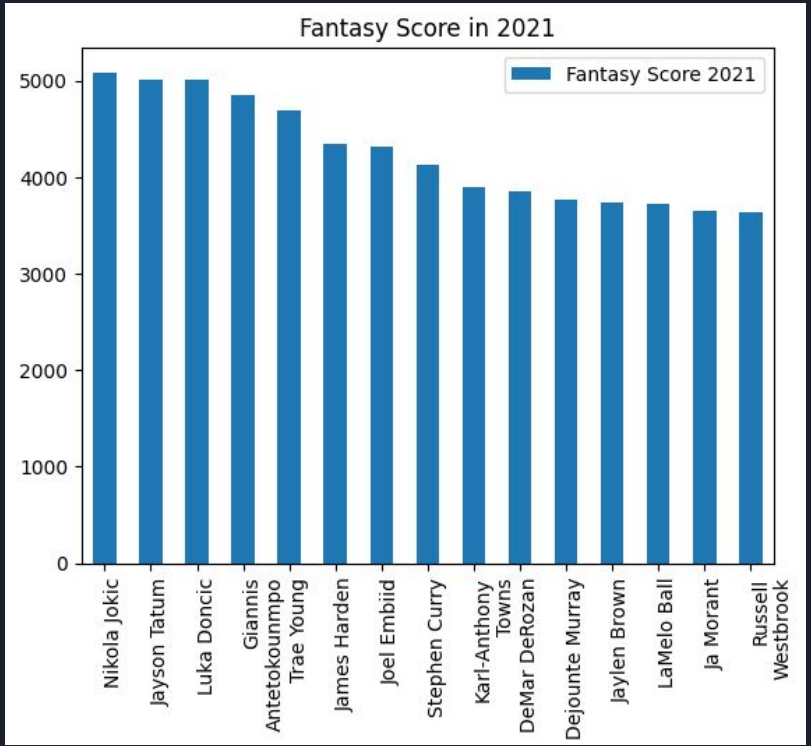
# Centers: Who gets the most double-doubles?



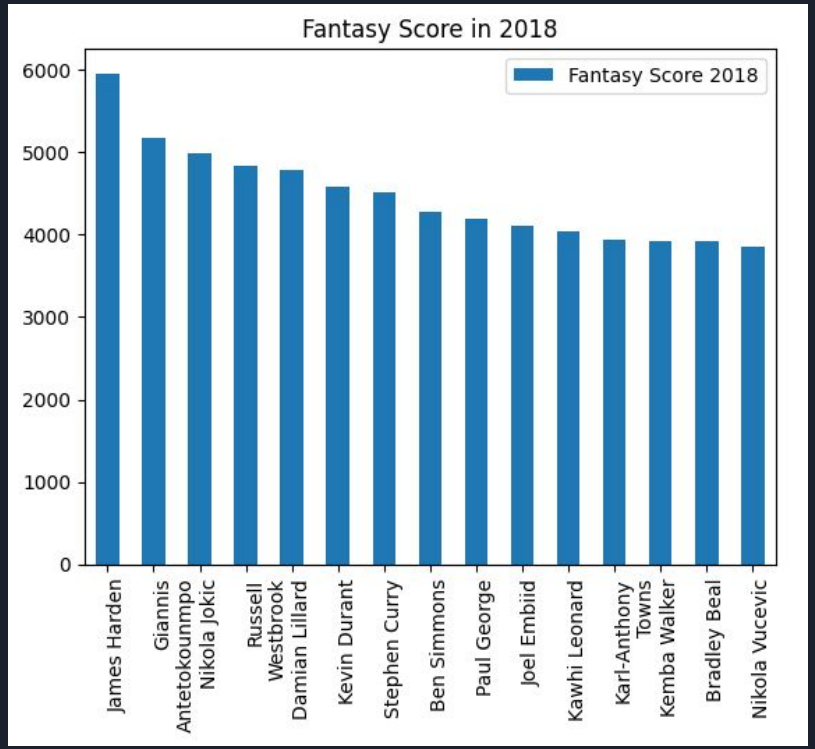
# Forwards: Who gets the most triple-doubles?



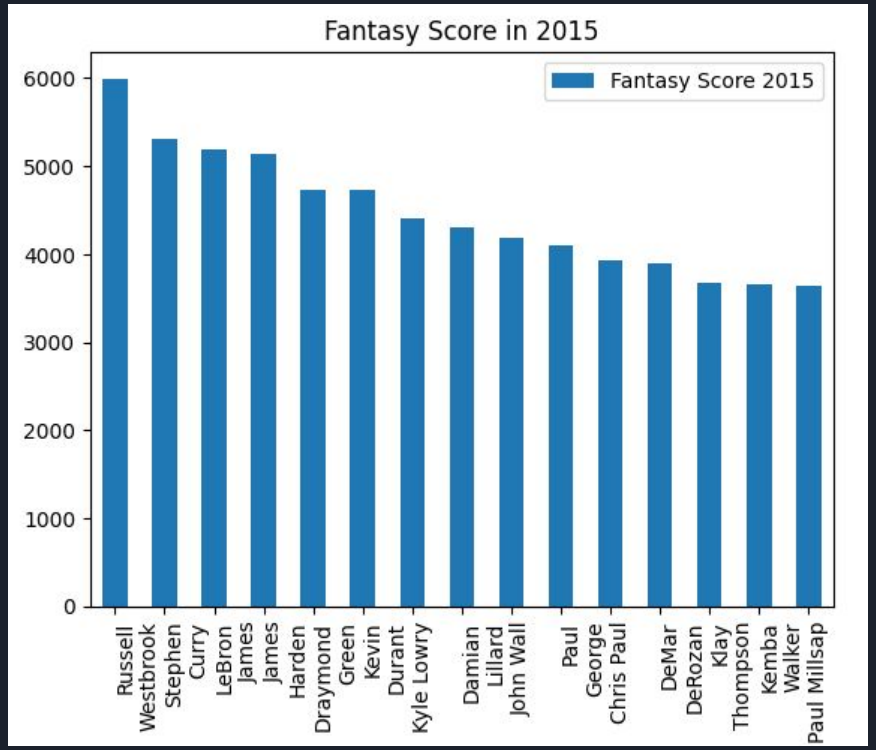
# Which players get the most fantasy points?



# Which players get the most fantasy points?



# Which players get the most fantasy points?



# ULTIMATE Fantasy Team

- Limited to the data set to 2015-2021 seasons
- Calculating the Fantasy
  - Pulled stats of 10 players
    - 4 guards
    - 4 forwards
    - 2 centers

# Our Fantasy Picks - Guards

James Harden

Luka Doncic

Stephen Curry

Russell Westbrook




# Our Fantasy Picks - Forwards

LeBron James

Draymond Green

Giannis Antetokounmpo

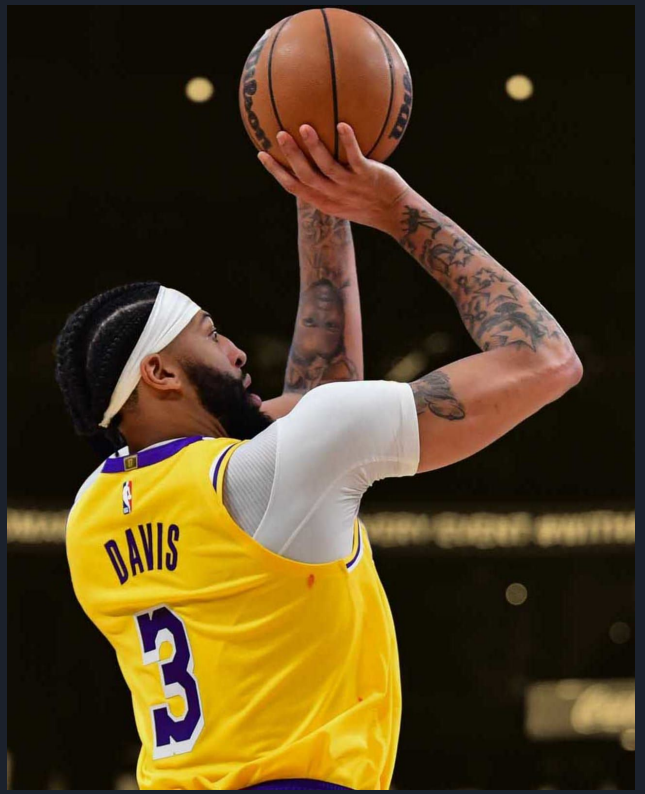
Jimmy Butler




# Our Fantasy Picks - Centers

Joel Embiid

Anthony Davis

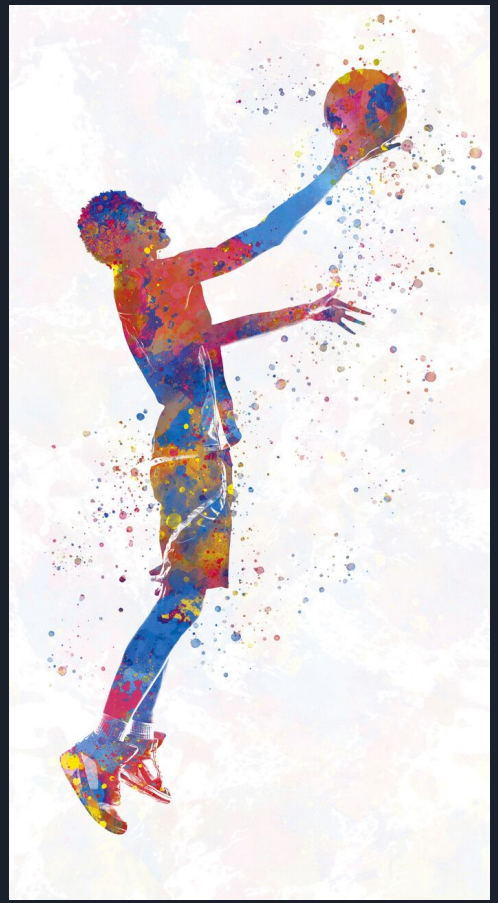



# Dream Team *Performance*

Going head to head with a randomly generated dream team!

How does our fantasy team perform against a randomly generated team?

***WHO WILL WIN?***

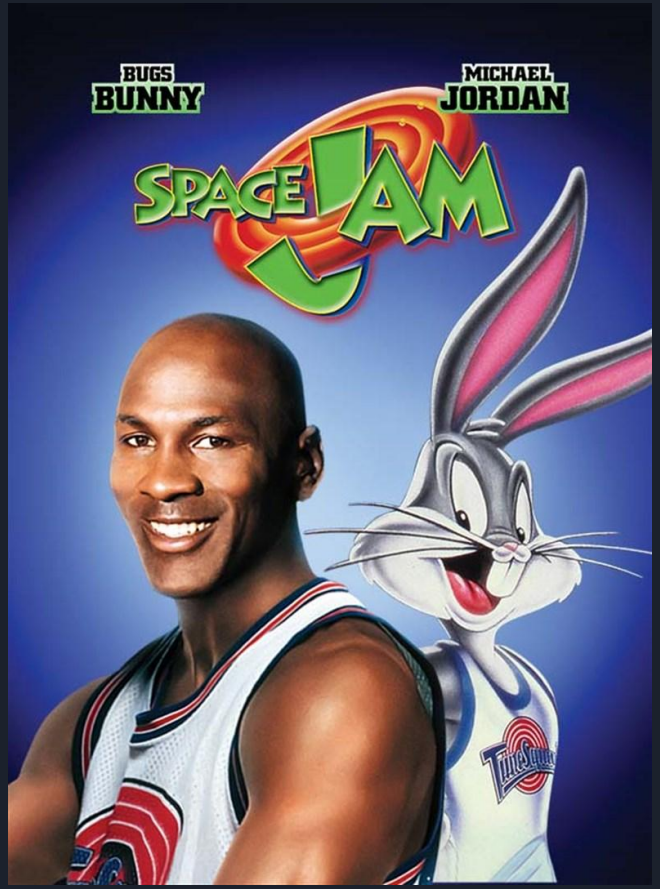


# More Info

- What went right
  - The data was large and readily available. It was really easy to get in and work with.
  - Data cleaning went very fast!
  - Simulation of games illustrated our dream team's performance
- What went wrong
  - Bokeh took a while to figure out - new libraries, but Brent persevered
  - Figuring out SQL (primary and foreign keys, table setup)
  - Putting together HTML to showcase our win/lose betting strategy
- What we'd do with more time
  - Develop fantasy teams for each of us to pit against each other and battle it out!
  - We wanted to do a teams analysis to see if any teams had particularly strong players, and how that figures into the teams scores. We also wanted to see if champion teams would win in a fantasy scenario

# That’s All Folks!

Please note: Michael Jordan was *not* included in our data set!
#oldman



# Reference Sources


NBA games data (kaggle.com)

3.12.3 Documentation (python.org)

SQLAlchemy Documentation — SQLAlchemy 2.0 Documentation


**Shout out to Thomas and Henry for all of your help!!!**