The NBA Data Playbook: Unraveling the Secrets of Success

Team Members: Ali Saadeddine, Bennett Blanco, Riris Grace, Saumya Anand, Sulaiman Alhomoud, Ya Chu Hsu

Instantiating Authentification

```
from google.colab import auth
auth.authenticate_user()
```

Table of Contents

Problem Definition

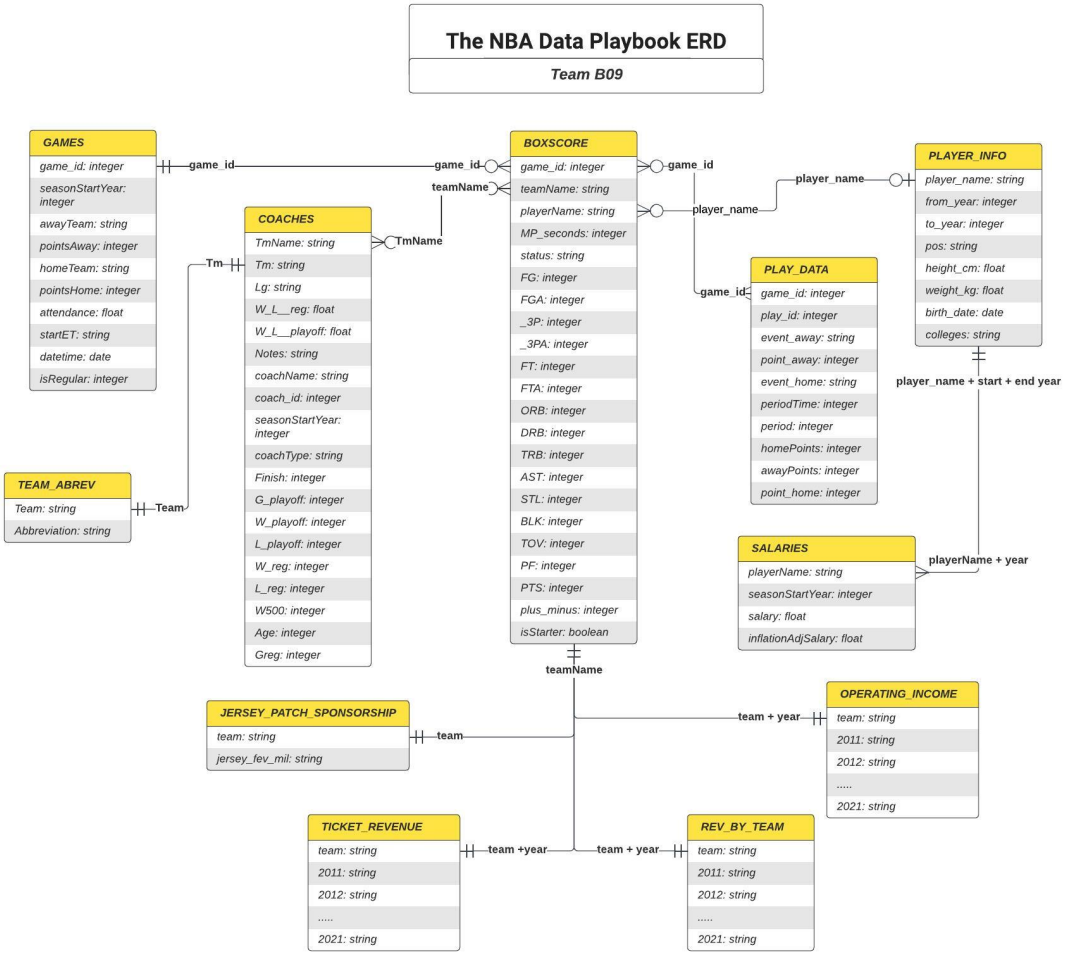
The NBA is the pinnacle of basketball, bringing together the best talent from around the world. It is also notorious for being very player-stats-driven, which often influences the decision-making of coaching and teams. Our goal is to analyze data for players, coaches, and organizations within the NBA to get a data-driven understanding of how all of these actors interact. For the scope of this project, we'll look at what drives the financial performance of organizations, as well as how player and team performance has affected wins/losses throughout the history of the league.

Each actor within the NBA will be able to realize value from this analysis. A better understanding of how each actor interacts with each other can help opponents strategize for upcoming games, guide organizations to make data-driven financial decisions, or influence decision-making in sports betting.

Executive Summary

The comprehensive analysis of NBA player performance, team dynamics, and financial trends reveals several key insights. Kevin Durant stands out as the scoring leader, emphasizing the dominance of top-tier players in the current era. LeBron James, despite lacking formal collegiate education, emerges as the highest-paid player. The correlation between top contributors and high scorers underscores the significance of offensive prowess over assists. Defensive prowess is exemplified by DeAndre Jordan, while successful coaching performances by legends like Steve Kerr and Phil Jackson are highlighted. Team-wise, the Los Angeles Lakers demonstrate consistent excellence, while fan engagement, reflected in attendance, is notably strong for teams like the Chicago Bulls. The timing of games at 8:00 pm correlates with higher scores, indicating more exciting matchups. Additionally, the analysis sheds light on the evolving landscape of player salaries over the years, reflecting the impact of inflation.

Entity Relationship Diagram



The ERD for the NBA Data Playbook have the interconnections among 11 tables, encapsulating various facets of NBA data, including Players, Teams, Games, Scores, and Financials. Games are associated with teams and their respective coaches, while players are connected to their performance data and salary details. Revenue streams are traced to teams across multiple years. Each game record is linked to specific team and coaching information, and player profiles are correlated with individual game statistics and financial data.The relationships between tables vary, encompassing one-to-one, one-to-many, and many-to-many associations. This data amalgamation provides comprehensive insights necessary for analyzing NBA games and unveiling interesting insights as described in Exploratory Data Analysis (EDA).

Introduction, Motivation, and Importance of the Project

Embarking on a data analysis project focused on the NBA is fueled by the vast and intricate web of information surrounding players, coaches, and organizations within the league. The motivation behind delving into this trove of data lies in the profound impact it can have across various domains. By unraveling the interplay between players, coaches, and organizations, a clearer understanding emerges, enabling opponents to craft strategic game plans that cater to specific strengths and weaknesses. Moreover, this wealth of information can serve as a compass for organizations, steering them toward data-driven financial decisions that optimize player acquisitions, contract negotiations, and overall team management. Beyond the court, the insights derived from data analysis can also play a pivotal role in the realm of sports betting, providing enthusiasts with a well-informed basis for decision-making. In essence, the motivation behind this NBA data analysis project lies in its potential to revolutionize how we perceive, strategize, and engage with the dynamic world of professional basketball.

Data Sources

- Web-scraped results of NBA data from https://www.basketball-reference.com/ available on Kaggle https://www.kaggle.com/datasets/patrickhallila1994/nba-data-from-basketball-reference?select=games.csv
- Financial data web-scraped from https://runrepeat.com/nba-revenue-statistics
- so far ununsed --> https://www.spotrac.com/nba/

Data Cleaning and Preparation

Constructing an ELT pipeline for Financial Data

Collecting data from [runrepeat] requires three things.

- We need to parse through the html code with BeautifulSoup and convert the html table content into dataframes.
- We have to create a service account in IAM & Admin, give the appropriate permissions, and then load the JSON key into our web scraping script
- Then we can load the results into the BigQuery project using the google cloud API to load a job.

This will be and ELT process since we're just going to set the entire schema to `String`. Once the data is in, we can make the proper adjustments. The pipeline script can be found here.

Loading and Cleaning Webscraped Data

Each webscraped table contains a column for NBA teams, and then year columns from 2011-2021. However, each year represents the same metric. To tidy this up we need to unpivot the columns. The process is similar for our three tables. Additionally, `operating_income` and `revenue` are measured in millions, so we'll make that adjustment as well.

`operating_income`

```
%%bigquery --project= teamb09
CREATE OR REPLACE TABLE `teamb09.web_scrape.operating_income_cleaned` AS (
  WITH loaded AS (
    SELECT Team AS team, CAST(year AS STRING) AS year, REGEXP_REPLACE(operatin
    FROM teamb09.web_scrape.operating_income
    UNPIVOT (
      operating_income FOR year IN (`2011`, `2012`, `2013`, `2014`, `2015`, `2
    )
  )

  SELECT team, year, CAST((CAST(REPLACE(operating_income, ' ', '') AS INT64) *
  FROM loaded
);
```

Job ID c10fb10e-4259-408b-944b-bc6be888acc0 successfully

executed: 100%

revenue

```
%%bigquery --project= teamb09
CREATE OR REPLACE TABLE `teamb09.web_scrape.revenue_cleaned` AS (
  WITH loaded AS (
    SELECT Team AS team, CAST(year AS STRING) AS year, REGEXP_REPLACE(revenue,
    FROM teamb09.web_scrape.rev_by_team
    UNPIVOT (
      revenue FOR year IN (`2011`, `2012`, `2013`, `2014`, `2015`, `2016`, `20
    )
  )

  SELECT team, year, CAST((CAST(REPLACE(revenue, ' ', '') AS INT64) * 1000000)
  FROM loaded
);
```

Job ID b1a3bdd7-349d-406e-a799-9e82df4bf38b successfully

executed: 100%

ticket_rev

```
%%bigquery --project= teamb09
CREATE OR REPLACE TABLE `teamb09.web_scrape.ticket_rev_cleaned` AS (
  WITH loaded AS (
    SELECT Team AS team, CAST(year AS STRING) AS year, REGEXP_REPLACE(revenue,
    FROM teamb09.web_scrape.rev_by_team
    UNPIVOT (
      revenue FOR year IN (`2011`, `2012`, `2013`, `2014`, `2015`, `2016`, `20
    )
  )

  SELECT team, year, CAST((CAST(REPLACE(revenue, ' ', '') AS INT64) * 1000000)
  FROM loaded
);
```

Job ID 6cb30230-e682-4b3c-b328-30ee3c9fc799 successfully executed: 100%

Cleaning Kaggle Data

`boxscore`

A major issue with the table `boxscore` is that stats columns for players, such as `STL` can contain `String` values if a player didn't play for a certain reason. We can use CASE to build out a `status` column that will map these values to a new column, and then we can leave the stats values as NULL so we don't skew the data when we aggregate these columns.

```
%%bigquery --project= teamb09
CREATE OR REPLACE TABLE teamb09.kaggle.boxscore_cleaned AS
SELECT
  game_id,
  teamName,
  playerName,

  CASE
    WHEN MP = 'Did Not Play' OR MP = 'Did Not Dress' THEN NULL
    WHEN MP = 'Not With Team' OR MP = 'Player Suspended' THEN NULL
    ELSE CAST(SPLIT(MP, ':')[OFFSET(0)] AS INT64) * 60 + CAST(SPLIT(MP, ':')[O
  END AS MP_seconds,

```

```
## Creating a status column

  CASE
    WHEN FG = 'Not With Team' OR FGA = 'Not With Team' OR _3P = 'Not With Team
         _3PA = 'Not With Team' OR FT = 'Not With Team' OR FTA = 'Not With Tea
         ORB = 'Not With Team' OR DRB = 'Not With Team' OR TRB = 'Not With Tea
         AST = 'Not With Team' OR STL = 'Not With Team' OR BLK = 'Not With Tea
         TOV = 'Not With Team' OR PF = 'Not With Team' OR PTS = 'Not With Team

    WHEN FG = 'Did Not Dress' OR FGA = 'Did Not Dress' OR _3P = 'Did Not Dress
         _3PA = 'Did Not Dress' OR FT = 'Did Not Dress' OR FTA = 'Did Not Dres
         ORB = 'Did Not Dress' OR DRB = 'Did Not Dress' OR TRB = 'Did Not Dres
         AST = 'Did Not Dress' OR STL = 'Did Not Dress' OR BLK = 'Did Not Dres
         TOV = 'Did Not Dress' OR PF = 'Did Not Dress' OR PTS = 'Did Not Dress

    WHEN FG = 'Did Not Play' OR FGA = 'Did Not Play' OR _3P = 'Did Not Play' O
         _3PA = 'Did Not Play' OR FT = 'Did Not Play' OR FTA = 'Did Not Play'
         ORB = 'Did Not Play' OR DRB = 'Did Not Play' OR TRB = 'Did Not Play'
         AST = 'Did Not Play' OR STL = 'Did Not Play' OR BLK = 'Did Not Play'
         TOV = 'Did Not Play' OR PF = 'Did Not Play' OR PTS = 'Did Not Play' T

    WHEN FG = 'Player Suspended' OR FGA = 'Player Suspended' OR _3P = 'Player
         _3PA = 'Player Suspended' OR FT = 'Player Suspended' OR FTA = 'Player
         ORB = 'Player Suspended' OR DRB = 'Player Suspended' OR TRB = 'Player
         AST = 'Player Suspended' OR STL = 'Player Suspended' OR BLK = 'Player
         TOV = 'Player Suspended' OR PF = 'Player Suspended' OR PTS = 'Player

    ELSE 'Played'
  END AS status,

-- Clean statistical columns, converting to INT64 and handle non-numeric statu
  CASE WHEN FG IN ('Did Not Play', 'Did Not Dress', 'Not With Team', 'Player S
  CASE WHEN FGA IN ('Did Not Play', 'Did Not Dress', 'Not With Team', 'Player
  CASE WHEN _3P IN ('Did Not Play', 'Did Not Dress', 'Not With Team', 'Player
  CASE WHEN _3PA IN ('Did Not Play', 'Did Not Dress', 'Not With Team', 'Player
  CASE WHEN FT IN ('Did Not Play', 'Did Not Dress', 'Not With Team', 'Player S
  CASE WHEN FTA IN ('Did Not Play', 'Did Not Dress', 'Not With Team', 'Player
  CASE WHEN ORB IN ('Did Not Play', 'Did Not Dress', 'Not With Team', 'Player
  CASE WHEN DRB IN ('Did Not Play', 'Did Not Dress', 'Not With Team', 'Player
  CASE WHEN TRB IN ('Did Not Play', 'Did Not Dress', 'Not With Team', 'Player
  CASE WHEN AST IN ('Did Not Play', 'Did Not Dress', 'Not With Team', 'Player
  CASE WHEN STL IN ('Did Not Play', 'Did Not Dress', 'Not With Team', 'Player
  CASE WHEN BLK IN ('Did Not Play', 'Did Not Dress', 'Not With Team', 'Player
  CASE WHEN TOV IN ('Did Not Play', 'Did Not Dress', 'Not With Team', 'Player
  CASE WHEN PF IN ('Did Not Play', 'Did Not Dress', 'Not With Team', 'Player S
```

```
  CASE WHEN PTS IN ('Did Not Play', 'Did Not Dress', 'Not With Team', 'Player

CAST(
  CASE
    WHEN `___` IN ('Did Not Play', 'Did Not Dress', 'Not With Team', 'Player S
    ELSE NULLIF(REGEXP_REPLACE(`___`, r'[^0-9-+]', ''), '')
  END AS INT64
) AS plus_minus,

  CAST(isStarter AS BOOLEAN) AS isStarter
FROM
  `teamb09.kaggle.boxscore`;
```

Job ID 4cb2c78b-3223-4ba8-9579-81bfcc327937 successfully

executed: 100%

coaches

In the `coaches` table, the following columns were classified as `float`:

- Age
- G_reg
- W_reg
- L_reg
- W_500
- Finish
- G_playoff
- W_playoff
- L-playoff

Their data type should be `int`, as they only contain numeric values.

The data type of each columns were changed accordingly. Next, missing values were handled, as there were nulls in G_playoff, W_playoff, L_playoff, W_L_playoff. We imputed these nulls with 0, as not every team makes the playoffs (resulting in null values)

Additionally, the `notes` column is actually in reference to how far a team made it in the playoffs. Since the only values are conference champions (east or west) and NBA champions, there are a lot of null values because only 2 of 32 teams can obtain this status every season. The nulls can stay, and the name will be switched from `notes` to `accolade`.

```
%%bigquery --project= teamb09
CREATE OR REPLACE TABLE `teamb09.kaggle.coaches_updated` AS (
  SELECT
    coach_id,
    coachName AS name,
    coachType AS role,
    CAST(Age AS INT64) AS age,
    Lg AS league,
    CAST(G_reg AS INT64) AS regular_games_coached,
    CAST(W_reg AS INT64) AS regular_season_wins,
    CAST(L_reg AS INT64) AS regular_season_losses,
    W_L__reg AS reagular_winloss_ratio,
    CAST(W____500 AS INT64) AS games_from_500,
    CAST(Finish AS INT64) AS season_rank,
    CAST(G_playoff AS INT64) AS playoff_games_coached,
    CAST(W_playoff AS INT64) AS playoff_wins,
    CAST(L_playoff AS INT64) AS playoff_losses,
    W_L__playoff AS playoff_winloss_ratio
```

```
    W_L__playoff AS playoff_win_loss_ratio,
    Notes as accolade,
    CASE
        WHEN coach.Tm = 'SEA' THEN 'Seattle Supersonics'
        WHEN coach.Tm = 'KEN' THEN 'Kentucky Colonels'
        WHEN coach.Tm = 'NJN' THEN 'New Jersey Nets'
        WHEN coach.Tm = 'PHO' THEN 'Phoenix Suns'
        WHEN coach.Tm = 'KCK' THEN 'Kansas City Kings'
        WHEN coach.Tm = 'BRK' THEN 'Brooklyn Nets'
        WHEN coach.Tm = 'WSB' THEN 'Washington Bullets'
        WHEN coach.Tm = 'VAN' THEN 'Vancouver Grizzlies'
        WHEN coach.Tm = 'NOH' THEN 'New Orleans Hornets'
        WHEN coach.Tm = 'CHH' THEN 'Charlotte Hornets'
        WHEN coach.Tm = 'SDC' THEN 'San Diego Clippers'
        WHEN coach.Tm = 'NOK' THEN 'New Orleans/Okalhoma City Hornets'
        WHEN coach.Tm = 'CHO' THEN 'Charlotte Hornets'
        WHEN coach.Tm = 'BUF' THEN 'Buffalo Braves'
        WHEN coach.Tm = 'CAR' THEN 'Carolina Cougars'
        WHEN coach.Tm = 'STL' THEN 'St. Louis Hawks'
        WHEN coach.Tm = 'DNA' THEN 'Denver Nuggets'
        ELSE abr.Abbreviation
    END AS Team
  FROM `kaggle.Coaches-Original` as coach
  LEFT JOIN `teamb09.web_scrape.team_abrev` as abr
  ON coach.Tm = abr.Team
);
```

Job ID fed1f2c2-c1b4-4456-a476-89c896072a6a successfully executed: 100%

games

The game table is fairly clean to begin with. The only exception is that the notes column here refers to special cases for a specific game. For example, if a game was played outside of the United States for a special event. This rarely happens, and doesn't affect anything else about how the game is played. We can just drop the column.

```
%%bigquery --project= teamb09
ALTER TABLE `teamb09.kaggle.games`
DROP COLUMN notes;
```

```
Executing query with job ID: 8e6e0d75-ae88-495e-bd09-f04eb634403f
Query executing: 0.56s
ERROR:
 400 Column not found: notes at [2:13]

Location: US
Job ID: 8e6e0d75-ae88-495e-bd09-f04eb634403f
```

`player_info`

A major issue with the `player_info` table is the lack of standardized measurements. Additionally the `birthDate` for each player is formatted as `string`.

Below are the converted fields:

- Height (from feet and inches to cm),
- Weight (from pounds to kg),
- Birthdate (from string to date format).

Additionally, there are instances of duplicate names in the table which actually represent different individuals. Therefore, when using the relationship between tables, we need to include the year to accurately identify players. This covers all the instances, because it is one of two cases:

- a duplicate player is a father and son, who played at different times
- the players actually have the same name, but are on different teams

```
%%bigquery --project= teamb09
--check duplicate player by all column
SELECT playerName, `From`, `To`, Pos, Ht, Wt, birthDate, colleges, count(*) re
GROUP BY playerName, `From`, `To`, Pos, Ht, Wt, birthDate, colleges
HAVING repeat > 1;

--check duplicate player by playerName only
CREATE OR REPLACE TABLE `teamb09.temp.player_info_duplicate`
AS
SELECT playerName, count(*) repeat FROM `teamb09.kaggle.player_info`
GROUP BY playerName
HAVING repeat > 1;

--check duplicate player by playerName and college
CREATE OR REPLACE TABLE `teamb09.temp.player_info_duplicate2`
AS
SELECT playerName,colleges, count(*) repeat FROM `teamb09.kaggle.player_info`
GROUP BY playerName, colleges
HAVING repeat > 1;

--create new table with convertion in height (from feet and inches to cm), wei
CREATE OR REPLACE TABLE `teamb09.kaggle.player_info_cleaned`
AS
SELECT
  playerName AS player_name,
  `From` AS from_year,
  `To` AS to_year,
  Pos AS pos,
  ROUND(((CAST((SPLIT(Ht, '-')[OFFSET(0)]) AS INT64 ) * 12) + CAST((SPLIT(Ht,
  ROUND(Wt*0.45359237,2) AS weight_kg,
  PARSE_DATE('%B %d, %Y', birthDate) as birth_date,
  Colleges as colleges
FROM `teamb09.kaggle.player_info`;
```

Job ID 9f6dda97-47f7-470d-b70f-7cb9990ce053 successfully

executed: 100%

salaries

The only issue with the `salaries` table, is that the data is loaded in as string due to the $ character. With some casting and replacing, this is easily resolved.

```
%%bigquery --project= teamb09
CREATE OR REPLACE TABLE teamb09.kaggle.salaries_cleaned AS
SELECT
  playerName,
  seasonStartYear,
  CAST(REPLACE(REPLACE(CAST(salary AS STRING), '$', ''), ',', '') AS FLOAT64)
  CAST(REPLACE(REPLACE(CAST(inflationAdjSalary AS STRING), '$', ''), ',', '')
FROM
  teamb09.kaggle.salaries;
```

Job ID 3a72ccae-679e-4034-bbdd-599d4c56c953 successfully executed: 100%

Creating a Per Game Stats Table

In general, NBA players are typically evaluated on the per game stats. Since the data we have contains stat lines by specific games, we need to group players game stats by name, team, and season. Then we'll take the average of each group (player in a given season) to get their per game stats. Saving this as a view will be useful for conducting further analyses at this level.

```
%%bigquery --project= teamb09
CREATE OR REPLACE VIEW `teamb09.kaggle.per_game_stats` AS (
  WITH player_stats AS (
    SELECT
      bs.game_id, bs.playerName AS Player, bs.teamName AS Team, -- Info and Pl
      g.seasonStartYear AS Season, g.awayTeam AS Opponent, -- Year and Opposin
      bs.PTS AS Points, bs.AST AS Assists, bs.TRB AS Total_Rebounds, bs.STL AS
      CASE
        WHEN bs.FGA = 0 THEN NULL
        ELSE ROUND(bs.FG / bs.FGA, 2)
        END AS FG_Pct, -- Getting field goal %
      CASE
        WHEN bs.FTA = 0 THEN NULL
        ELSE ROUND(bs.FT / bs.FTA, 2)
        END AS FT_Pct, -- Getting free throw goal %
      CASE
        WHEN bs.ThreeP = 0 THEN NULL
        ELSE ROUND(bs.ThreeP / bs.ThreePA, 2)
        END AS Three_Pct, -- Getting 3ptr %
      CASE
        WHEN bs.MP_seconds = 0 THEN NULL
        ELSE ROUND(bs.MP_seconds / 60, 2)
        END AS MPG,
      bs.plus_minus
    FROM `teamb09.kaggle.boxscore_cleaned` AS bs
    INNER JOIN `teamb09.kaggle.games` AS g
    USING(game_id)
  )
  SELECT Player, Team, Season, ROUND(AVG(Points), 1) AS PPG, ROUND(AVG(Assists
  ROUND(SUM(FG) / NULLIF(SUM(FGA), 0), 2) AS FGP,
  ROUND(SUM(FT) / NULLIF(SUM(FTA), 0), 2) AS FTP,
  ROUND(SUM(ThreeP) / NULLIF(SUM(ThreePA), 0), 2) AS ThreePP
  FROM player_stats
  GROUP BY Player, Team, Season
);
```

Job ID 24241bd2-c1ce-49f1-91e6-f12498e3e643 successfully executed: 100%

No charts were generated by quickchart

Exploratory Data Analysis

Player Analysis

What are each players relative stats?

To see how players are performing relative to the league in a given season, we can compare a players per game stats to the league average in a given season. Players with positive relative stats are "better" than the average player in a given season, and vice versa if they have negative values.

```
%%bigquery --project= teamb09
SELECT
  pgs.Player, pgs.Team, pgs.Season, pi.pos,
  ROUND(pgs.PPG - AVG(pgs.PPG) OVER(PARTITION BY pgs.Season), 1) AS Relative_P
  ROUND(pgs.APG - AVG(pgs.APG) OVER(PARTITION BY pgs.Season), 1) AS Relative_A
  ROUND(pgs.RPG - AVG(pgs.RPG) OVER(PARTITION BY pgs.Season), 1) AS Relative_R
  ROUND(pgs.SPG - AVG(pgs.SPG) OVER(PARTITION BY pgs.Season), 1) AS Relative_S
  ROUND(pgs.BPG - AVG(pgs.BPG) OVER(PARTITION BY pgs.Season), 1) AS Relative_B
  ROUND(pgs.TPG - AVG(pgs.TPG) OVER(PARTITION BY pgs.Season), 1) AS Relative_T
  ROUND(pgs.PPG - AVG(pgs.PPG) OVER(PARTITION BY pgs.Season), 1) AS Relative_P
  ROUND(pgs.FGP - AVG(pgs.FGP) OVER(PARTITION BY pgs.Season), 1) AS Relative_F
  ROUND(pgs.FTP - AVG(pgs.FTP) OVER(PARTITION BY pgs.Season), 1) AS Relative_F
  ROUND(pgs.ThreePP - AVG(pgs.ThreePP) OVER(PARTITION BY pgs.Season), 1) AS Re
FROM `teamb09.kaggle.per_game_stats` AS pgs
LEFT JOIN `teamb09.kaggle.player_info_cleaned` AS pi
ON pgs.Player = pi.player_name;
```

Job ID d9fc3fe7-656e-46be-ad65-78b3dde9cae7 successfully executed: 100%

Downloading: 100%

	Player	Team	Season	pos	Relative_PPG	Relative_APG	Relative_F
0	Larry Hughes	Golden State Warriors	2000	G	9.0	2.7	
1	Rod Strickland	Washington Wizards	2000	G	4.7	5.2	-
2	Toni Kukoč	Atlanta Hawks	2000	None	12.2	4.4	
3	Bob Sura	Golden State Warriors	2000	G	3.6	2.8	
4	Nick Van Exel	Denver Nuggets	2000	G	10.2	6.7	-
...	...	...	...	...	...	...	
10870	Reggie Slater	New Jersey Nets	2001	F	-6.5	-1.8	-
10871	Charlie Bell	Dallas Mavericks	2001	G	-7.8	-1.8	-
10872	Ernest Brown	Miami Heat	2001	C	-6.8	-1.8	-
	Matt	Philadelphia					

Top 5 Players with Highest Average Points Per Game (2010-2022)

This query identifies the players with the highest scoring efficiency over the past decade. The way basketball is played is often changing every several years. So, this helps us see which are the most dominant scorers in the current state of the game.

```
%%bigquery --project=teamb09

SELECT b.playerName, AVG(b.PTS) as avgPoints
FROM teamb09.kaggle.boxscore_cleaned AS b
JOIN teamb09.kaggle.games AS g ON b.game_id = g.game_id
WHERE g.seasonStartYear BETWEEN 2010 AND 2022
GROUP BY b.playerName
ORDER BY avgPoints DESC
LIMIT 5;
```

Job ID 1752173f-1791-4d63-9c52-7582c68dd90d successfully executed: 100%

Downloading: 100%

	playerName	avgPoints
0	Kevin Durant	27.970509
1	LeBron James	26.954442
2	James Harden	26.253488
3	Russell Westbrook	25.066085
4	Stephen Curry	24.607387

The results show that Kevin Durant leads with the highest average points per game at `27.97`, followed by LeBron James, James Harden, Russell Westbrook, and Stephen Curry. This makes sense, given that these players are regarded as the most dominant in the current era.

Which college basketball programs generate the highest average salary in the NBA?

The reasoning behind the question is mainly insighful for upcoming 4 or 5 star recruits that know they'll be one and dones (enter the NBA draft after one season of college ball). These players typically chose from the places they're offered, so it might be helpful to see which generate the highest NBA returns, if money is an issue. Furthermore, we'll see how many alumni are in the NBA for each school.

```
%%bigquery --project=teamb09
-- What university generate the highest salary on average?
  WITH rookies AS (
    SELECT playerName, MIN(seasonStartYear) Rookie_Season
    FROM kaggle.player_info_cleaned as pi
    LEFT JOIN kaggle.salaries_cleaned as sc
    ON pi.player_name = sc.playerName
    WHERE salary IS NOT NULL AND seasonStartYear BETWEEN from_year AND to_year
    GROUP BY playerName
  ),

  rookie_contracts AS (
    SELECT
      DISTINCT r.playerName,
      r.Rookie_Season,
      MIN(sc.salary) AS Starting_Salary,
      MAX(pi.colleges) AS colleges
    FROM
      rookies r
    LEFT JOIN
      `teamb09.kaggle.salaries_cleaned` sc
      USING(playerName)
    LEFT JOIN
      `teamb09.kaggle.player_info_cleaned` pi
    ON
      r.playerName = pi.player_name
    GROUP BY
      r.playerName, r.Rookie_Season
  )

  SELECT colleges, ROUND(AVG(Starting_Salary), 2) avg_starting_salary, COUNT(r
  FROM rookie_contracts
  GROUP BY colleges
  HAVING COUNT(colleges) > 10 -- removes outlies
  ORDER BY avg_starting_salary DESC
```

Job ID 381680ab-a37f-473f-997b-f2cbc7df4919 successfully executed: 100%

Downloading: 100%

	colleges	avg_starting_salary	alumni
0	Duke	1697143.71	38
1	Alabama	1579037.93	15

2	Texas	1566146.47	17
3	Kentucky	1517124.29	51
4	Ohio State	1512539.31	13
5	Arizona	1487520.44	32
6	Illinois	1467066.00	12
7	California	1432654.27	15
8	Kansas	1431402.87	31
9	UNLV	1355984.45	11
10	Indiana	1349964.53	17
11	Washington	1309692.89	18
12	UConn	1169818.79	24
13	Villanova	1152386.83	18
14	Syracuse	1136434.42	26
15	Georgia Tech	1112527.09	23
16	UCLA	1035730.89	38
17	LSU	1008263.70	20
18	Maryland	1000580.22	18
19	NC State	977272.91	11
20	Florida State	957327.13	16
21	Stanford	930863.67	15
22	Louisville	927139.33	15
23	Florida	902627.21	19
24	UNC	891403.90	42
25	Michigan	865953.92	24
26	Memphis	845610.47	15
27	Michigan State	804530.04	24
28	Marquette	788682.27	15

29	USC	728681.73	11
30	Pitt	724897.50	12
31	Arkansas	703807.55	11
32	Georgetown	694303.36	14
33	St. John's	651514.45	11
34	Iowa State	644778.36	11
35	Cincinnati	617836.77	13
36	Notre Dame	516121.69	13

From this we can infer the most reputable college basketball programs. We see that Duke players will generally have the highest salary, which checks out. They generally have the largest amount of high performing players (Kyrie Irving,Carlos Boozer, Shane Battier, JJ Reddick). From the count we'll see that UNC actually has the largest alumni base from our years of observation (42). As a prospect selecting a college, this may help you make a more informed decision. Do you want a higher salary going into the league? If offered a spot, go for the highest average school in your list. Maybe money is less of a concern and you'd rather just get to the league? Maybe the bigger alumni base will be more beneficial to you.

Players with Highest Contribution in Points Scored and Assists (2010-2022)

This query identifies players with the highest overall impact on the game through scoring and playmaking. It sums the total points and assists for each player. This should show similar results to the highest average scorers.

```
%%bigquery --project=teamb09

SELECT b.playerName, SUM(b.PTS) + SUM(b.AST) AS totalContribution
FROM `teamb09.kaggle`.boxscore_cleaned AS b
JOIN `teamb09.kaggle`.games AS g ON b.game_id = g.game_id
WHERE g.seasonStartYear BETWEEN 2010 AND 2022
GROUP BY b.playerName
ORDER BY totalContribution DESC
LIMIT 5;
```

Job ID 31987a9a-278a-4e8f-b8e4-e6931d2f7884 successfully executed: 100%

Downloading: 100%

	playerName	totalContribution
0	LeBron James	30321
1	James Harden	28273
2	Russell Westbrook	27025
3	Kevin Durant	24267
4	Stephen Curry	22841

LeBron James stands out as the top contributor with a total contribution score of `30,321`, followed by James Harden, Russell Westbrook, Kevin Durant, and Stephen Curry. These results are the exact same as the highest average scorers, which may suggest that assists don't carry as much wait when considering a players offensive prowess.

Best Defensive Players (2010-2022)

This query aims to identify the best defensive players by considering steals, blocks, and defensive rebounds. It provides a composite score based on these defensive metrics. With a better understanding of who is defensively the best player, teams with low defensive ratings can look to scout these players, or an upcoming opponent can strategize ahead of time to try and work around this player.

```
%%bigquery --project=teamb09

SELECT b.playerName, SUM(b.STL) + SUM(b.BLK) + SUM(b.DRB) AS totalDefenseScore
FROM `teamb09.kaggle`.boxscore_cleaned AS b
JOIN `teamb09.kaggle`.games AS g ON b.game_id = g.game_id
WHERE g.seasonStartYear BETWEEN 2010 AND 2022
GROUP BY b.playerName
ORDER BY totalDefenseScore DESC
LIMIT 5;
```

Job ID d22ac213-5dec-429e-8d49-4bb211740cd5 successfully executed: 100%

Downloading: 100%

	playerName	totalDefenseScore
0	DeAndre Jordan	8644
1	LeBron James	7915
2	Dwight Howard	7592
3	Andre Drummond	7277
4	Kevin Durant	6775

DeAndre Jordan emerged as the top defensive player with a total defensive score of `8,644`, with LeBron James, Dwight Howard, Andre Drummond, and Kevin Durant completing the top five.

Identifying Players with the Most Triple-Doubles Off the Bench in Regular Games

This query aims to find the players with the most triple-doubles off the bench in regular games, this query allows us to see that there are reserve players with high potential, as evidenced by their ability to score triple-doubles even though they are not the main players.

```
%%bigquery --project=teamb09
SELECT boxscore.*, games.datetime
FROM teamb09.kaggle.boxscore_cleaned AS boxscore
LEFT JOIN teamb09.kaggle.games ON boxscore.game_id = games.game_id
WHERE
    (((boxscore.PTS>=10 AND boxscore.TRB>=10 AND boxscore.AST>=10) OR
    (boxscore.PTS>=10 AND boxscore.TRB>=10 AND boxscore.STL>=10) OR
    (boxscore.PTS>=10 AND boxscore.TRB>=10 AND boxscore.BLK>=10) OR
    (boxscore.PTS>=10 AND boxscore.AST>=10 AND boxscore.STL>=10) OR
    (boxscore.PTS>=10 AND boxscore.AST>=10 AND boxscore.BLK>=10) OR
    (boxscore.PTS>=10 AND boxscore.STL>=10 AND boxscore.BLK>=10) OR
    (boxscore.TRB>=10 AND boxscore.AST>=10 AND boxscore.STL>=10) OR
    (boxscore.TRB>=10 AND boxscore.AST>=10 AND boxscore.BLK>=10) OR
    (boxscore.TRB>=10 AND boxscore.STL>=10 AND boxscore.BLK>=10) OR
    (boxscore.AST>=10 AND boxscore.STL>=10 AND boxscore.BLK>=10)) AND
    boxscore.isStarter = false)
ORDER BY games.datetime DESC;
```

Job ID 8f28a413-8b89-45ce-915f-20ac133489af successfully executed: 100%

Downloading: 100%

	game_id	teamName	playerName	MP_seconds	status	FG	FGA	ThreeP	Three
0	28966	Portland Trail Blazers	Evan Turner	1705	Played	6	7	1	
1	28951	Portland Trail Blazers	Evan Turner	1496	Played	5	5	1	
2	28521	Los Angeles Clippers	Lou Williams	1933	Played	9	19	2	
3	28041	New Orleans Pelicans	Julius Randle	1520	Played	9	13	0	
4	27702	Philadelphia	Markelle	1515	Played	6	13	0	

		76ers	Fultz					
5	27669	Boston Celtics	Greg Monroe	1668	Played	8	10	0
6	27326	Philadelphia 76ers	T.J. McConnell	2230	Played	5	11	0
7	25531	New Orleans Pelicans	Tim Frazier	2276	Played	4	10	0
8	24923	Orlando Magic	Elfrid Payton	2069	Played	9	21	0
9	24614	Miami Heat	Hassan Whiteside	1622	Played	4	8	0
10	24453	Boston Celtics	Marcus Smart	1951	Played	4	12	1
11	23211	Miami Heat	Hassan Whiteside	1477	Played	6	10	0
12	21929	Houston Rockets	Jeremy Lin	1751	Played	6	12	1
13	21607	New Orleans Pelicans	Tyreke Evans	1974	Played	2	10	0
14	20143	Milwaukee Bucks	Larry Sanders	1928	Played	5	7	0
15	19545	Minnesota Timberwolves	J.J. Barea	2821	Played	10	22	2
		New Jersey	Terrence					

The results here are interesting, as this list consists of a lot of 6th MOTY recipients and highly valued role players. If a front office needs extra firepower going into the trade deadline, these are good players to look to trade for or sign if they happen to be free agents.

Finding Players with Triple-Doubles but Fewer than 10 Points in NBA Games

This query allows us to see one of the rarest triple-double achievements in the NBA with fewer than 10 points. A player achieves triple-doubles, but they have less than 10 points in the game.

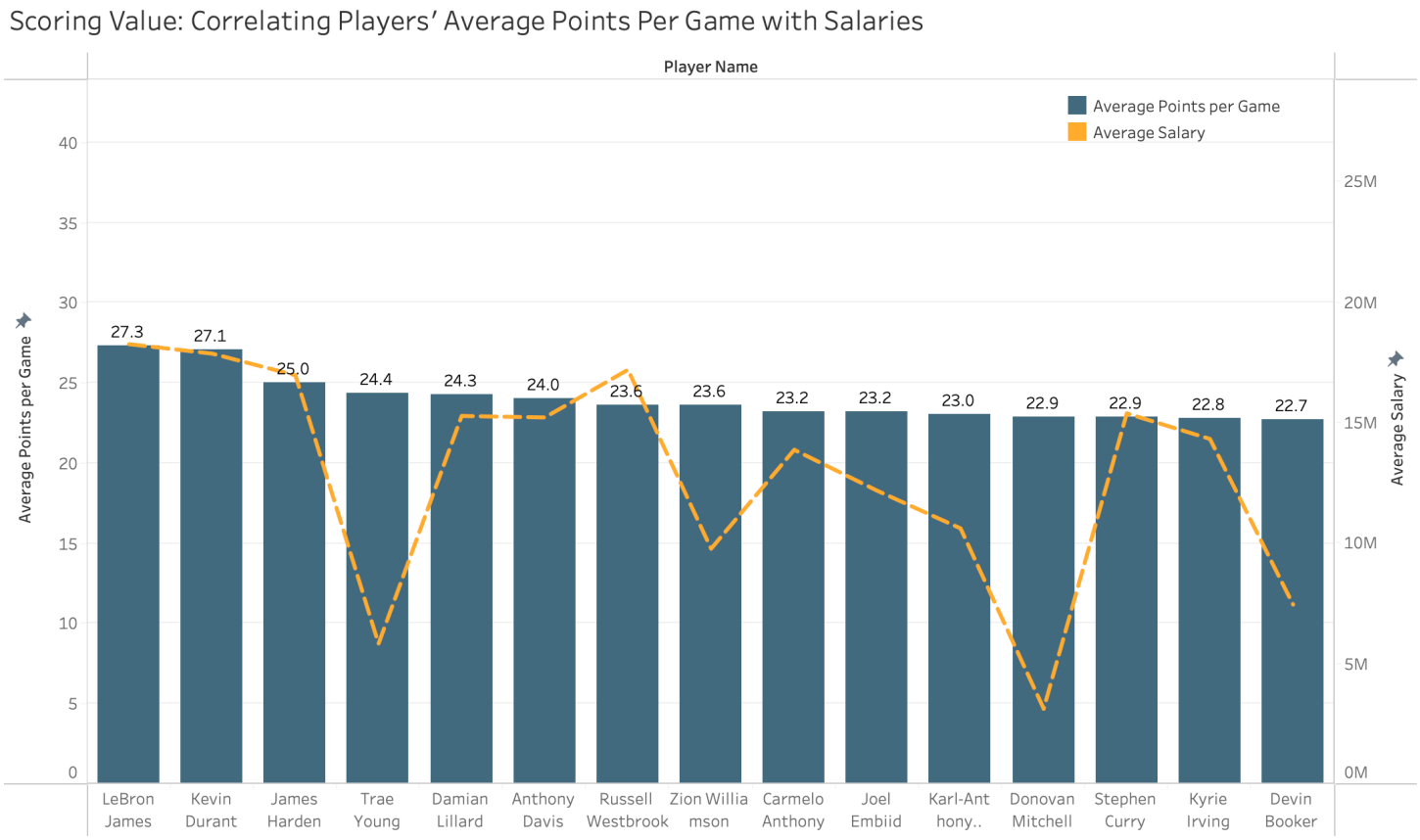
```
%%bigquery --project=teamb09
SELECT boxscore.*, games.datetime
FROM teamb09.kaggle.boxscore_cleaned AS boxscore
LEFT JOIN teamb09.kaggle.games ON boxscore.game_id = games.game_id
WHERE
    ((boxscore.TRB>=10 AND boxscore.AST>=10 AND boxscore.STL>=10) OR
    (boxscore.TRB>=10 AND boxscore.AST>=10 AND boxscore.BLK>=10) OR
    (boxscore.TRB>=10 AND boxscore.STL>=10 AND boxscore.BLK>=10) OR
    (boxscore.AST>=10 AND boxscore.STL>=10 AND boxscore.BLK>=10)) AND
    boxscore.PTS<10
ORDER BY games.datetime DESC;
```

Job ID 34722f3b-8f68-4819-ae57-a12ec73ea033 successfully executed: 100%

Downloading: 100%

	game_id	teamName	playerName	MP_seconds	status	FG	FGA	ThreeP	ThreePA
0	25975	Golden State Warriors	Draymond Green	2264	Played	2	6	0	3

This is a fun insight, it doesn't provide much business value, but it is still an interesting find. The value from this stat line was related to the buzz and publicity that the league and Draymond Green got when this occured.



There's a visible trend where players with higher scoring averages tend to have larger salaries, as seen with stars like LeBron James and Kevin Durant. However, the correlation isn't absolute, with some players like Trae Young and Donovan Mitchel having low salaries compared to their scoring averages. This indicates that while scoring is a significant factor in salary determination, it's not the only one, with a player's brand, defensive skills, or other on-court contributions likely also playing critical roles in their valuation

Team Analysis

How do the various teams perform across the playoff season and the regular season?

This query helps us analyze what is the ratio of win/lose across the two seasons- playoffs and regular. It also might help front offices understand which coaches and players actually have what it takes when the serious games come around.

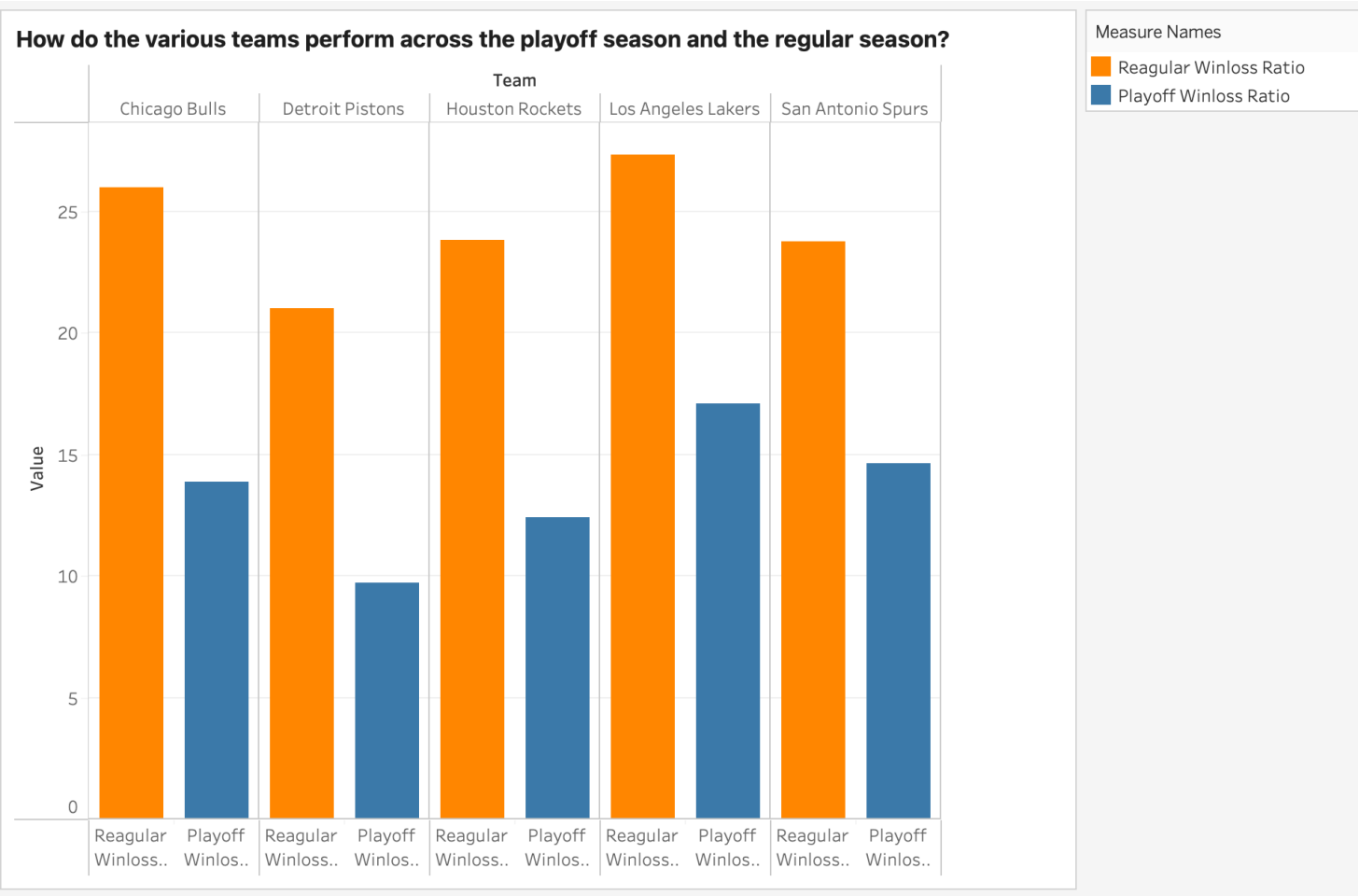
```
%%bigquery --project= teamb09
SELECT
    Team,
    SUM(CASE
        WHEN playoff_winloss_ratio > reagular_winloss_ratio THEN 1
        ELSE 0
    END) AS PlayOffGreaterReg,
    SUM(CASE
        WHEN playoff_winloss_ratio < reagular_winloss_ratio THEN 1
        ELSE 0
    END) AS RegGreaterPlayOff
FROM
    teamb09.kaggle.coaches_updated
GROUP BY
    Team
ORDER BY PlayOffGreaterReg DESC, RegGreaterPlayOff DESC
LIMIT 5;
```

Job ID a1e6deb4-959e-4ee8-843b-54415e3f8703 successfully executed: 100%

Downloading: 100%

	Team	PlayOffGreaterReg	RegGreaterPlayOff
0	Los Angeles Lakers	9	22
1	Detroit Pistons	4	19
2	San Antonio Spurs	3	28
3	Houston Rockets	3	27
4	Chicago Bulls	3	27

It can be seen that the team 'LAL'- Los Angeles Lakers performed the best across both the seasons with a cumulative Win to Lose ratio of 9 in Play off season and 22 in Regular season.



Teams with Highest Average Attendance in Home Games (2010-2022)

This query assesses the popularity and fan support for teams by analyzing average home game attendances. Higher attendance often correlates with team performance and fan engagement.

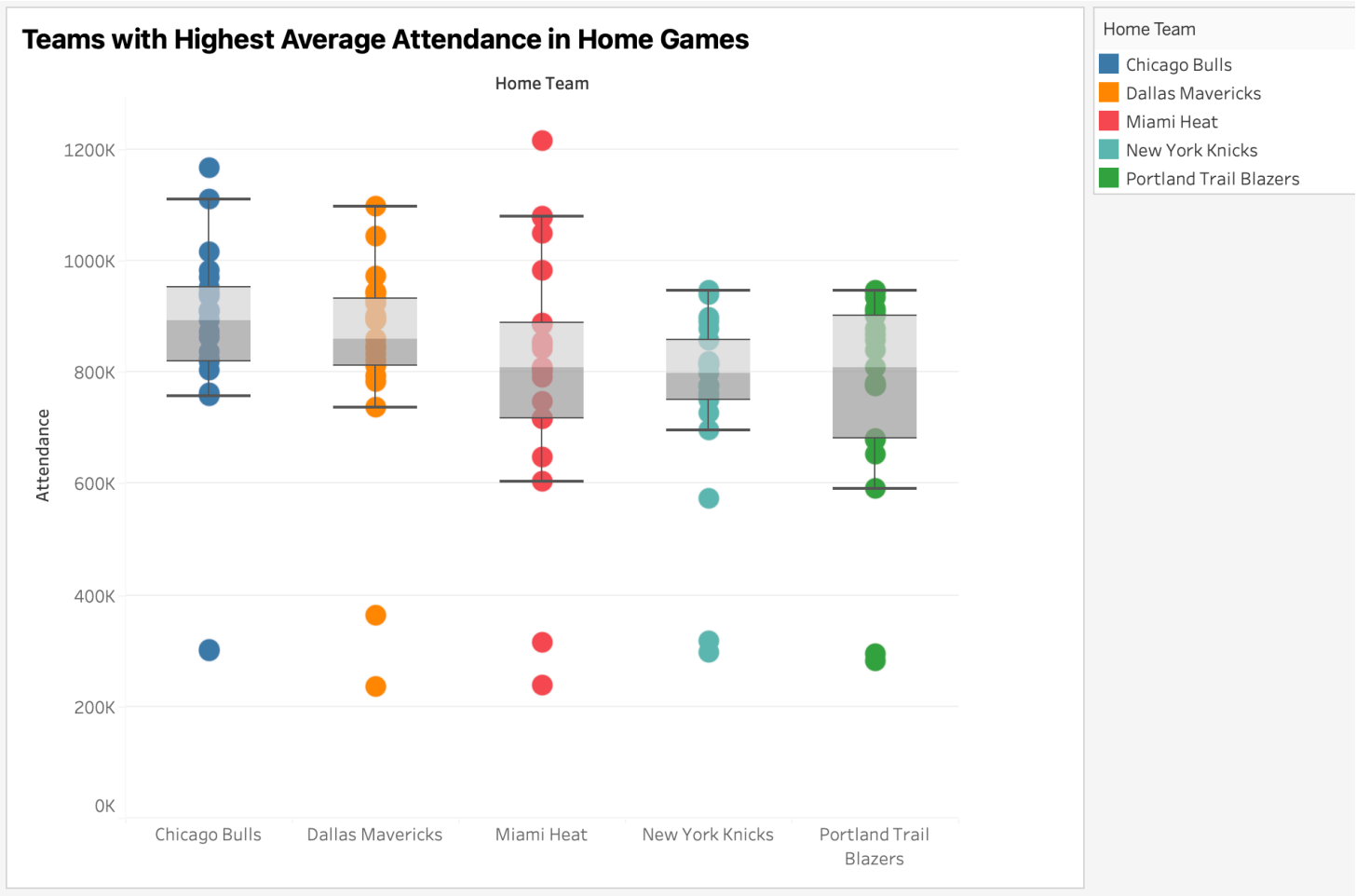
```
%%bigquery --project=teamb09

SELECT homeTeam, AVG(attendance) as avgAttendance
FROM `teamb09.kaggle`.games
WHERE seasonStartYear BETWEEN 2010 AND 2022
GROUP BY homeTeam
ORDER BY avgAttendance DESC
LIMIT 5;
```

Job ID 88a222f3-ecf6-4d31-a7ed-9d46b74c15f7 successfully executed: 100%

Downloading: 100%

	homeTeam	avgAttendance
0	Chicago Bulls	21323.851415
1	Dallas Mavericks	20066.455422
2	Miami Heat	19765.599109
3	Portland Trail Blazers	19735.252381
4	New York Knicks	19491.985149

The Chicago Bulls had the highest average attendance with `21,323`, followed by the Dallas Mavericks, Miami Heat, Portland Trail Blazers, and New York Knicks. This indicates that these teams likely had strong fan engagement during the specified period. These results are also supported by the appearance of teams such as the Bulls, Heat, and Knicks, which have some of the deepest and most dedicated fanbases in the league. Furtheremore, this could be correlated with the size of the arena that the teams play in.



What are the times of the day with the most exciting games?

This SQL query retrieves information about the top 5 times of the day that have the highest-scoring games, based on the sum of points scored by both the away and home teams.

It helps identify and rank the times of the day when games are most exciting or have the highest scores, allowing us to pinpoint specific periods associated with intense or high-scoring basketball matches.

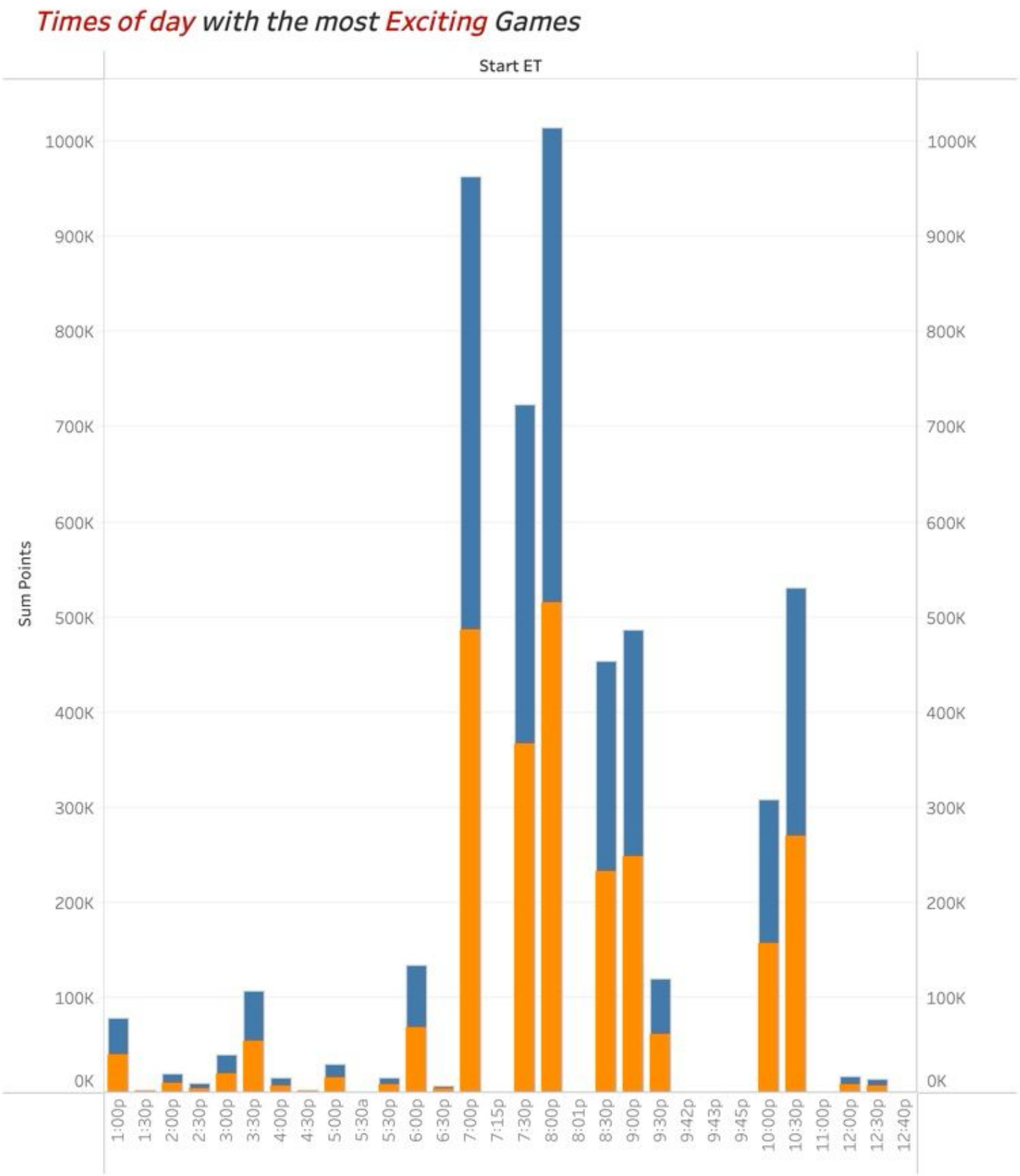
```
%%bigquery --project=teamb09
SELECT startET, SUM(pointsAway + pointsHome) AS totalPoints
FROM `teamb09.kaggle.games`
GROUP BY startET
ORDER BY totalPoints DESC
LIMIT 5;
```

Job ID e91826d6-45ec-4536-9a74-2d4c05e05b8c successfully executed: 100%

Downloading: 100%

	startET	totalPoints
0	8:00p	1015144
1	7:00p	963577
2	7:30p	723439
3	10:30p	530293
4	9:00p	486496

This is interesting, as the 8:00pm slot is usually the NBA primetime game. Usually the higher performing teams that are matched up for the week get this slot, so it would make sense that they're scoring more and that the games are more "exciting".


Times of day with the most Exciting Games
Start ET
Sum Points
1000K
900K
800K
700K
600K
500K
400K
300K
200K
100K
0K
1:00p
1:30p
2:00p
2:30p
3:00p
3:30p
4:00p
4:30p
5:00p
5:30a
5:30p
6:00p
6:30p
7:00p
7:15p
7:30p
8:00p
8:01p
8:30p
9:00p
9:30p
9:42p
9:43p
9:45p
10:00p
10:30p
11:00p
12:00p
12:30p
12:40p

Which coach have the Highest Win Percentage in Regular Games (2010-2022)?

This query determines the most successful coaches based on their win percentage in regular games. It is a measure of a coach's ability to lead their team to victory.

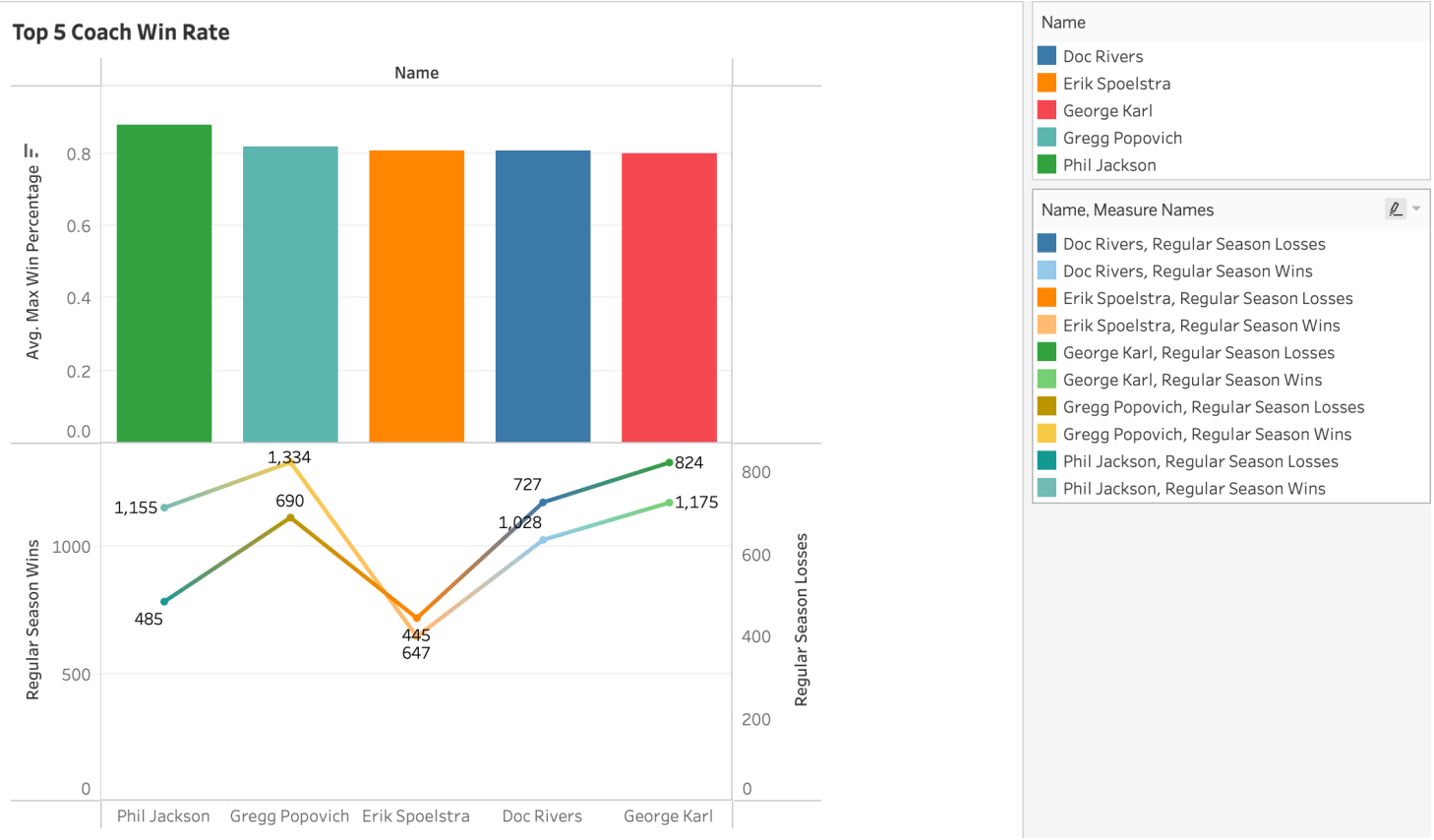
```
%%bigquery --project=teamb09
--new query
SELECT c.Name as Name, count(Name) as season_num, SUM(regular_games_coached) a
FROM `teamb09.kaggle`.coaches_updated AS c
GROUP BY c.Name
HAVING play_num>1000
ORDER BY maxWinPercentage DESC, play_num DESC
LIMIT 5;
```

Job ID 67073ef8-07fe-43d7-9df0-331100d16951 successfully executed: 100%

Downloading: 100%

	Name	season_num	play_num	maxWinPercentage
0	Phil Jackson	20	1640	0.878049
1	Gregg Popovich	26	2024	0.817073
2	Doc Rivers	23	1755	0.804878
3	Erik Spoelstra	14	1092	0.804878
4	George Karl	27	1999	0.800000

From the data, Erik Spoelstra is the youngest coach with the fewest number of games, which is 1092 over 14 seasons, yet he ranks among the top 5 coaches with the highest win percentage. He is ranked fourth among senior coaches who have coached more than 1600 games, namely Phil Jackson, Gregg Popovich, Doc Rivers, and George Karl. Among these coaches, Gregg Popovich is the oldest active basketball coach which also listed as one of Hall of Fame NBA coaches.



Financial Analysis

What are the top five players by salary, per season?

Let's take a quick look at the per game stats for the highest paid players in each season. After we get this information, we'll be able to pair it with relative per game stats to see which players are undervalued or overvalued.

```
%%bigquery --project= teamb09
WITH RankedSalaries AS (
    SELECT
        pgs.*,
        sc.salary,
        ROW_NUMBER() OVER (PARTITION BY pgs.Season ORDER BY sc.salary DESC) AS
    FROM
        `teamb09.kaggle.per_game_stats` AS pgs
    LEFT JOIN
        kaggle.salaries_cleaned AS sc
    ON
        pgs.Player = sc.playerName
)
SELECT *,
FROM RankedSalaries
WHERE SalaryRank <= 5;
```

Job ID 1e1b8c7e-92da-4a25-8504-a6bfd0994f4c successfully executed: 100%

Downloading: 100%

	Player	Team	Season	PPG	APG	RPG	SPG	BPG	TPG	FGP	FTP	Th
0	Chris Paul	New Orleans/Oklahoma City Hornets	2006	17.2	8.9	4.4	1.8	0.0	2.5	0.44	0.82	
1	LeBron James	Cleveland Cavaliers	2006	26.9	6.4	7.0	1.6	0.7	3.2	0.46	0.71	
2	Chris Paul	New Orleans/Oklahoma City Hornets	2006	17.2	8.9	4.4	1.8	0.0	2.5	0.44	0.82	
3	LeBron James	Cleveland Cavaliers	2006	26.9	6.4	7.0	1.6	0.7	3.2	0.46	0.71	
4	Kyle Lowry	Memphis Grizzlies	2006	5.6	3.2	3.1	1.4	0.1	1.2	0.37	0.89	
...	...	...	...	...	...	...	...	...	...	...	...	...
95	Stephen Curry	Golden State Warriors	2010	18.6	5.8	3.9	1.5	0.3	3.1	0.48	0.93	
96	Chris Paul	New Orleans Hornets	2010	16.3	9.9	4.3	2.3	0.1	2.3	0.47	0.87	

It can be seen that Chris Paul has been ranked first, in terms of salary drawn, followed by Russell Westbrook and LeBron James, on the second and third position, respectively, for the '08 season. It has to be noted that these values would differ across different seasons.

What Players are Undervalued or Overvalued?

This will be a super insightful query. By building off the previous query and comparing relative per game stats to relative salary, we can valuate players. For example, a player that has positive relative stats and positive or negative and negative, they are probably fairly valued. However, if a player has negative per game relative stats and a positive relative salary, they might be overvalued. The inverse of this would indicate the play is undervalued.

```
%%bigquery --project= teamb09
-- get relative stats
WITH relative_stats AS (
  SELECT
    pgs.Player, pgs.Team, pgs.Season, pi.pos,
    ROUND(pgs.PPG - AVG(pgs.PPG) OVER(PARTITION BY pgs.Season), 1) AS Relative
    ROUND(pgs.APG - AVG(pgs.APG) OVER(PARTITION BY pgs.Season), 1) AS Relative
    ROUND(pgs.RPG - AVG(pgs.RPG) OVER(PARTITION BY pgs.Season), 1) AS Relative
    ROUND(pgs.SPG - AVG(pgs.SPG) OVER(PARTITION BY pgs.Season), 1) AS Relative
    ROUND(pgs.BPG - AVG(pgs.BPG) OVER(PARTITION BY pgs.Season), 1) AS Relative
    ROUND(pgs.TPG - AVG(pgs.TPG) OVER(PARTITION BY pgs.Season), 1) AS Relative
    ROUND(pgs.PPG - AVG(pgs.PPG) OVER(PARTITION BY pgs.Season), 1) AS Relative
    ROUND(pgs.FGP - AVG(pgs.FGP) OVER(PARTITION BY pgs.Season), 1) AS Relative
    ROUND(pgs.FTP - AVG(pgs.FTP) OVER(PARTITION BY pgs.Season), 1) AS Relative
    ROUND(pgs.ThreePP - AVG(pgs.ThreePP) OVER(PARTITION BY pgs.Season), 1) AS
    ROUND(sc.Salary - AVG(sc.Salary) OVER(PARTITION BY pgs.Season), 2) AS Rela
  FROM `teamb09.kaggle.per_game_stats` AS pgs
  LEFT JOIN `teamb09.kaggle.player_info_cleaned` AS pi
  ON pgs.Player = pi.player_name
  LEFT JOIN `teamb09.kaggle.salaries_cleaned` AS sc
  ON pgs.Player = sc.playerName
),

-- Rank the Salaries
RankedSalaries AS (
    SELECT
        rs.*,
        sc.salary,
```

```
        ROW_NUMBER() OVER(PARTITION BY rs.Season ORDER BY sc.salary DESC) AS S
    FROM
        relative_stats AS rs
    LEFT JOIN
        `teamb09.kaggle.salaries_cleaned` AS sc
    ON
        rs.Player = sc.playerName
)

-- Compare relative stats and relative salary to the average to see if players
SELECT *
FROM RankedSalaries;
```

Job ID c7134acf-fc9f-4b9b-928e-e5357f57a4d0 successfully executed: 100%

Downloading: 100%

	Player	Team	Season	pos	Relative_PPG	Relative_APG	Relative_RPG
0	James Harden	Houston Rockets	2019	G	24.5	5.2	2.3
1	James Harden	Houston Rockets	2019	G	24.5	5.2	2.3
2	James Harden	Houston Rockets	2019	G	24.5	5.2	2.3
3	LeBron James	Los Angeles Lakers	2019	F-G	15.8	8.4	3.8
4	LeBron James	Los Angeles Lakers	2019	F-G	15.8	8.4	3.8
...	...	...	...	...	...	...	...
1003006	Leandro Barbosa	Indiana Pacers	2011	G	-1.3	-0.6	-1.8
1003007	Leandro Barbosa	Indiana Pacers	2011	G	-1.3	-0.6	-1.8
1003008	Keith Bogans	New Jersey Nets	2011	G-F	-4.9	-1.4	-1.8
1003009	Keith Bogans	New Jersey Nets	2011	G-F	-4.9	-1.4	-1.8
1003010	Keith Bogans	New Jersey Nets	2011	G-F	-4.9	-1.4	-1.8

1003011 rows × 17 columns

The results of this table are valuable for front office stakeholders. Depending on the need, you can navigate or filter the table to find overvalued players on your team that you might want to cut salary for or let go when contract negotiations come around. Additionally it's an excellent resource to scout undervalued talent.

How has inflation affected salaries across various years in relation to player popularity?

This bar chart explores the intersection of macroeconomics and professional sports by comparing average NBA player salaries and their inflation-adjusted counterparts from 2011 to 2019. The horizontal axis spans the years, while the vertical axis measures salaries in millions of dollars. The first set of bars reveals the salaries adjusted for inflation, offering a nuanced understanding of how economic factors have influenced player compensation, while the second set showcases raw fluctuations in annual average salaries. Peaks and troughs expose the correlation between economic conditions and player earnings, providing a succinct visual narrative of the intricate relationship between inflation and NBA salaries over the past decade.

This visual representation serves as a powerful lens into the financial dynamics of the NBA, illustrating how macroeconomic trends have shaped the earning potential of players. By juxtaposing average salaries with their inflation-adjusted counterparts, the chart provides a concise yet comprehensive narrative, appealing to sports enthusiasts, economists, and those intrigued by the interplay of finance and athletics.

Burrowing deeper into the analysis pictured through the above plot, the aforementioned query helps analyze how inflation, across different season start year, affects the salary drawn by different players based on their popularity. It provides insights into how macroeconomic factors mingle with extensive fanbase of NBA players.

```
%%bigquery --project=teamb09
SELECT DISTINCT(playerName), seasonStartYear, AVG(inflationAdjSalary) AS AvgIn
FROM teamb09.kaggle.salaries_cleaned
GROUP BY seasonStartYear, playerName
ORDER BY seasonStartYear DESC, AvgInflationSalary DESC;
```

Job ID 4cf65f3a-e2f3-41a7-b3b5-e78ed0b8b94f successfully executed: 100%

Downloading: 100%

	playerName	seasonStartYear	AvgInflationSalary
0	Stephen Curry	2019	42674629.0
1	Chris Paul	2019	40844595.0
2	Russell Westbrook	2019	40844595.0
3	John Wall	2019	40518442.0
4	James Harden	2019	40518442.0
...	...	...	...
11577	Stephen Howard	1996	314002.0
11578	Henry James	1996	314002.0
11579	Aaron Williams	1996	293022.0
11580	Robert Werdann	1996	287820.0
11581	Gaylon Nickerson	1996	265107.0

11582 rows × 3 columns

It can be seen that there is a huge difference between the average salaries drawn by players from 2019 to 1996, exhibiting how starkly the implications of infaltion have changed over the years. There has been a huge jump in the average salaries drawn, exmplafiying how inflation has changed the value of a dollar.

Average Salaries of Top 5 Highest-Paid Players (2010-2022)

This query provides insights into the financial aspects of the league, highlighting the top earners and their average salaries. It reflects the market value of elite players.

```
%%bigquery --project=teamb09

SELECT playerName, ROUND(AVG(salary), 2) as avgSalary
FROM `teamb09.kaggle`.salaries_cleaned
WHERE seasonStartYear BETWEEN 2010 AND 2022
GROUP BY playerName
ORDER BY avgSalary DESC
LIMIT 5;
```

Job ID 0643e815-2c00-4bf1-abff-107c509cfad0 successfully executed: 100%

Downloading: 100%

	playerName	avgSalary
0	Kobe Bryant	26142123.83
1	LeBron James	24809006.70
2	Chris Paul	23058352.20
3	Kevin Durant	21859077.20
4	Chris Bosh	20648922.22

Kobe Bryant was the highest-paid player on average with `26.1 million dollars`, followed by LeBron James, Chris Paul, Kevin Durant, and Chris Bosh.

Analyzing Players with Most Triple-Doubles but Below Average Salaries

This query enables us to identify players who have achieved the most triple doubles while earning a low salary (lower than the average for each season). This information is useful for teams in scouting for talent.

```
%%bigquery --project=teamb09
--query for the player's with triple double
CREATE OR REPLACE TABLE teamb09.temp.triple_double
```

```
AS
SELECT boxscore.*, games.datetime,
FROM teamb09.kaggle.boxscore_cleaned AS boxscore
INNER JOIN teamb09.kaggle.games ON boxscore.game_id = games.game_id
WHERE
    ((boxscore.PTS>=10 AND boxscore.TRB>=10 AND boxscore.AST>=10) OR
    (boxscore.PTS>=10 AND boxscore.TRB>=10 AND boxscore.STL>=10) OR
    (boxscore.PTS>=10 AND boxscore.TRB>=10 AND boxscore.BLK>=10) OR
    (boxscore.PTS>=10 AND boxscore.AST>=10 AND boxscore.STL>=10) OR
    (boxscore.PTS>=10 AND boxscore.AST>=10 AND boxscore.BLK>=10) OR
    (boxscore.PTS>=10 AND boxscore.STL>=10 AND boxscore.BLK>=10) OR
    (boxscore.TRB>=10 AND boxscore.AST>=10 AND boxscore.STL>=10) OR
    (boxscore.TRB>=10 AND boxscore.AST>=10 AND boxscore.BLK>=10) OR
    (boxscore.TRB>=10 AND boxscore.STL>=10 AND boxscore.BLK>=10) OR
    (boxscore.AST>=10 AND boxscore.STL>=10 AND boxscore.BLK>=10))
ORDER BY games.datetime DESC;

--query for the player's salary under average
CREATE OR REPLACE TABLE teamb09.temp.low_salary
AS
SELECT playerName, seasonStartYear, salary, ROUND(avg_salary,2) AS avg_salary,
FROM
  (SELECT *, AVG(salary) OVER(PARTITION BY seasonStartYear) AS avg_salary, sal
  FROM teamb09.kaggle.salaries_cleaned
  ORDER BY playerName, seasonStartYear)
WHERE relative_salary<1;

--query to identify player's with triple double score, but paid under average
SELECT triple_double.*, low_salary.salary, low_salary.avg_salary, low_salary.r
FROM teamb09.temp.triple_double AS triple_double
INNER JOIN teamb09.temp.low_salary AS low_salary
ON triple_double.playerName = low_salary.playerName AND EXTRACT(YEAR FROM trip

```

Job ID 40ac0226-0c79-4388-954b-37e34f4b6614 successfully executed: 100%

Downloading: 100%

	game_id	teamName	playerName	MP_seconds	status	FG	FGA	ThreeP	ThreeI
0	29494	Miami Heat	Bam Adebayo	2249	Played	7	12	0	
1	29460	Miami Heat	Bam Adebayo	2611	Played	13	18	0	
2	29337	Atlanta Hawks	Trae Young	2483	Played	11	23	4	
3	28984	New York Knicks	Mario Hezonja	2523	Played	6	11	0	
4	28782	Atlanta Hawks	Trae Young	1719	Played	4	14	2	
...	...	...	...	...	...	...	...	...	
125	5517	Los Angeles Clippers	Jeff McInnis	2619	Played	8	15	3	
126	5006	Philadelphia 76ers	Aaron McKie	2855	Played	8	16	2	
127	4792	Los Angeles Clippers	Lamar Odom	2492	Played	6	12	1	
		Charlotte							

It can be seen that Bam Adebayo has drawn a relatively lower salary while scoring the most triple doubles.

Evaluating Players with Low Performance (High Fouls and Turnovers) but High Salaries (5 times higher than average salary)

This query enables us to identify the player with the highest total of personal fouls and turnovers (low performance) who also has a high salary.

```
%%bigquery --project=teamb09
--query to create table of players most high number of fouls and turnover
CREATE OR REPLACE TABLE `teamb09.temp.fouls_turnover`
AS
SELECT boxscore.game_id, boxscore.teamName, boxscore.playerName, boxscore.TOV,
FROM teamb09.kaggle.boxscore_cleaned AS boxscore
INNER JOIN teamb09.kaggle.games ON boxscore.game_id = games.game_id
WHERE TOV IS NOT NULL AND PF IS NOT NULL
ORDER BY games.datetime DESC;

--query for the player's salary 5 times higher average
CREATE OR REPLACE TABLE teamb09.temp.high_salary
AS
SELECT playerName, seasonStartYear, salary, ROUND(avg_salary,2) AS avg_salary,
FROM
  (SELECT *, AVG(salary) OVER(PARTITION BY seasonStartYear) AS avg_salary, sal
  FROM teamb09.kaggle.salaries_cleaned
  ORDER BY playerName, seasonStartYear)
WHERE relative_salary>5;

--query to identify player's with low performance, but paid 5 times higher tha
SELECT fouls_turnover.*, high_salary.salary, high_salary.avg_salary, high_sala
FROM teamb09.temp.fouls_turnover AS fouls_turnover
INNER JOIN teamb09.temp.high_salary AS high_salary
ON fouls_turnover.playerName = high_salary.playerName AND fouls_turnover.year
ORDER BY fouls_turnover.TOV desc, fouls_turnover.PF DESC, high_salary.relative
LIMIT 100;
```

Job ID b8c911fc-3a31-4373-902f-1748adf95c7a successfully executed: 100%

Downloading: 100%

	game_id	teamName	playerName	TOV	PF	year	salary	avg_salary	relati
0	27201	Cleveland Cavaliers	LeBron James	11	0	2018	35654150.0	6256553.48	
1	28172	Oklahoma City Thunder	Russell Westbrook	10	5	2018	35665000.0	6256553.48	
2	28621	Oklahoma City Thunder	Russell Westbrook	10	3	2019	38506482.0	6922103.41	
3	28245	Detroit Pistons	Blake Griffin	10	3	2018	31873932.0	6256553.48	
4	20118	Los Angeles Lakers	Kobe Bryant	10	2	2012	27849000.0	4274055.67	
...	...	...	...	...	...	...	...	...	
95	21932	New York Knicks	Carmelo Anthony	7	4	2014	22458401.0	4293242.99	

The results here are interesting because all of these are high performing players. This might be a result of these players being the main ball handlers or main defensive players for their team. Since, they're the focal point in most of the plays during the game, it would make sense that they're recording more turnovers and fouls. A great example of this is Russell Westbrook. Westbrook handled the ball most of the time in OKC, which led to a lot of turnovers. However, he was still an exceptional player.

Preping Data for Tableau

If we look at the query above that creates the per game stats for each NBA player, you'll notice that each stat has its own columns. While this makes sense in tabular form, Tableau will not be able to understand that these are all related and we won't be able to create meaningful visualizations. To get things ready for our dashboard, we actually will need to UNPIVOT the data. This will create the table in such a way that stat values are all in one column, and the stat names are all in one column. Furthermore, we'll attach some additional player information as well as salary so we don't need to make those relationships later in Tableau.

Note that this query below will not run. There are lots of union alls, so to keep things tidy, the entire query is not below (rather a sample). Refer to the BigQuery database for the entire query.

```
%%bigquery --project=teamb09
WITH UnpivotedData AS (
  SELECT
    Player,
    Team,
    Season,
    'PPG' AS stat,
    PPG AS per_game,
    PERCENT_RANK() OVER (PARTITION BY Season ORDER BY PPG) AS percentile
  FROM
    kaggle.pgs

  UNION ALL

  SELECT
    Player,
    Team,
    Season,
    'FTP' AS stat,
    FTP AS per_game,
    PERCENT_RANK() OVER (PARTITION BY Season ORDER BY FTP) AS percentile
  FROM
    kaggle.pgs

  UNION ALL... -- Repeat this step for each per game stat you need inside the

  )

  SELECT
    Player,
    Team,
    Season,
    stat,
    per_game,
    percentile
  FROM
    UnpivotedData
)
```

Next We do a similar process for the average stats per seasons

```
%%bigquery --project=teamb09
SELECT
```

```
    Season,
    REPLACE(stat, 'avg_', '') AS stat,
    average
FROM
    (
        SELECT
            Season,
            avg_PPG,
            avg_APG,
            avg_RPG,
            avg_SPG,
            avg_BPG,
            avg_TPG,
            avg_FGP,
            avg_FTP,
            avg_ThreePP,
            avg_salary
        FROM kaggle.avg_stats_per_szn
    ) src
UNPIVOT
    (
        average FOR stat IN (
            avg_PPG,
            avg_APG,
            avg_RPG,
            avg_SPG,
            avg_BPG,
            avg_TPG,
            avg_FGP,
            avg_FTP,
            avg_ThreePP,
            avg_salary
        )
    ) unpiv
```

Job ID ae0a0fa7-1825-4748-9d97-ada570bd2b77 successfully executed: 100%

Downloading: 100%

	Season	stat	average
0	2002	PPG	7.76
1	2002	APG	1.77
2	2002	RPG	3.54
3	2002	SPG	0.66
4	2002	BPG	0.42
...	...	...	...
195	2012	TPG	1.12
196	2012	FGP	0.43
197	2012	FTP	0.71
198	2012	ThreePP	0.28
199	2012	salary	4274055.67

200 rows × 3 columns

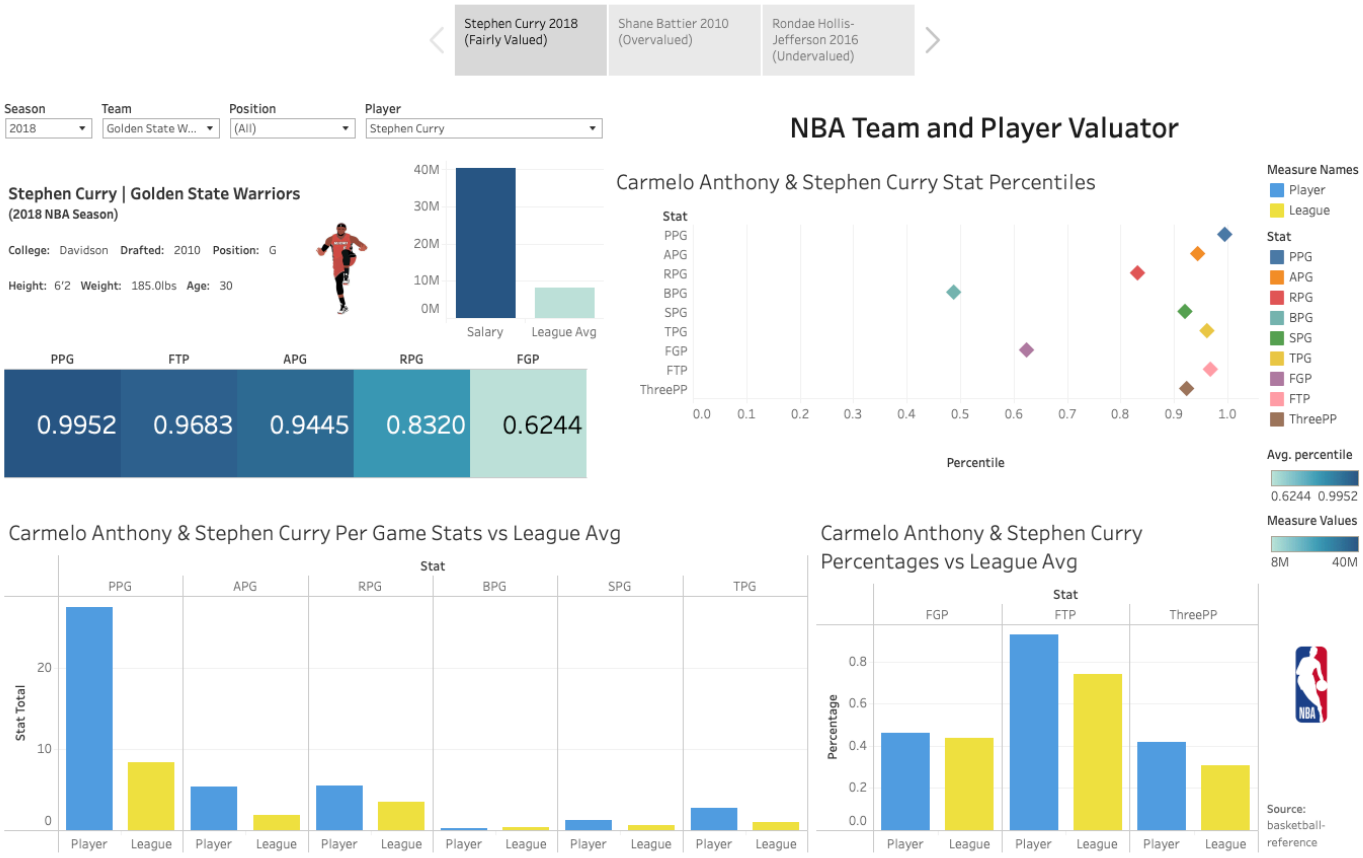
Finally we create our datasource in Tableau as a custom SQL query so that are dashboard is fully functional with all the required data

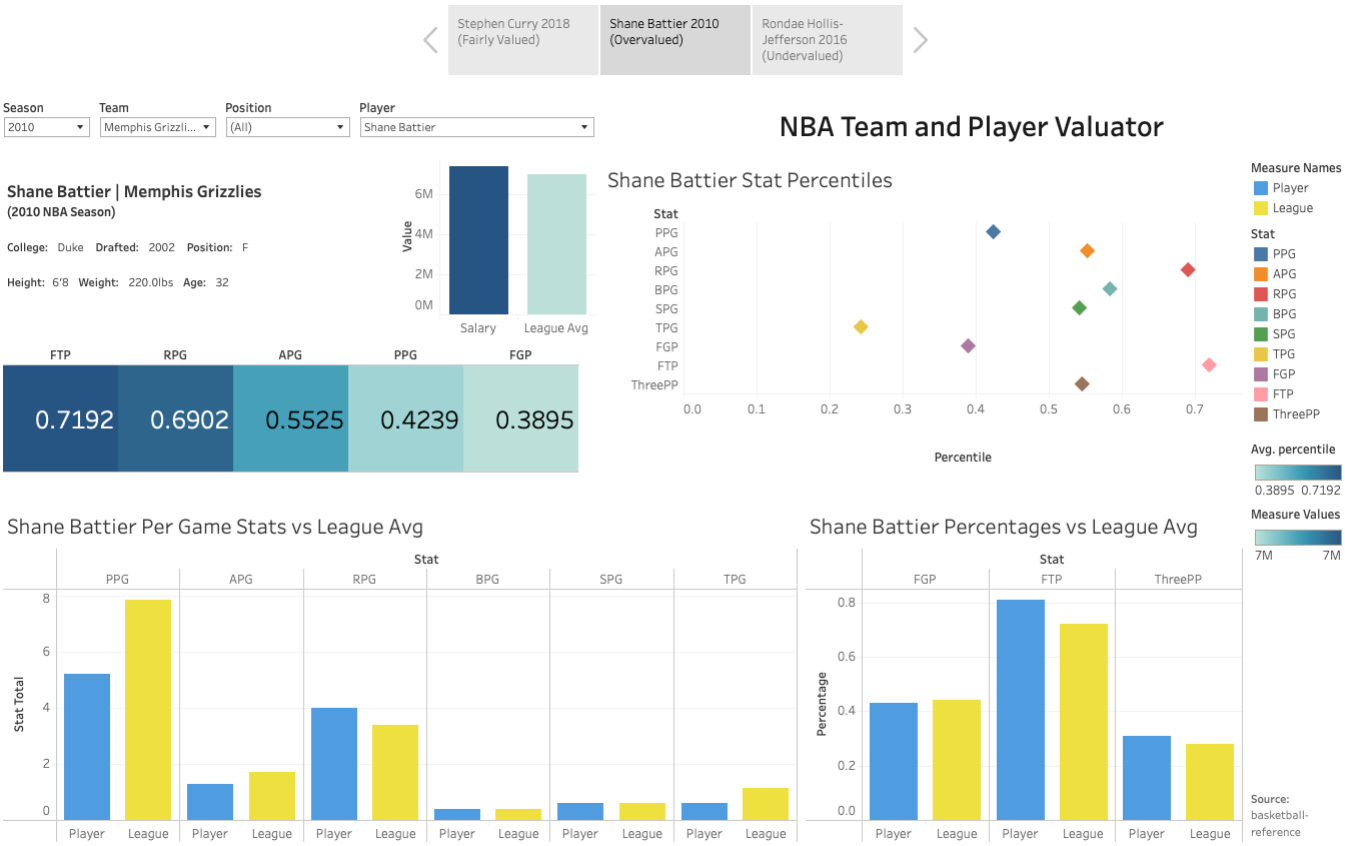
```
%%bigquery --project=teamb09
  SELECT Player, Team, Season, pos, birth_date, colleges, height_cm, weight_kg
  FROM `teamb09.kaggle.pgs_unpivot`
  LEFT JOIN kaggle.unpiv_avg_per_game_szns
  USING(Season, Stat)
  LEFT JOIN kaggle.player_eval
  USING(Player, Team, Season)
```

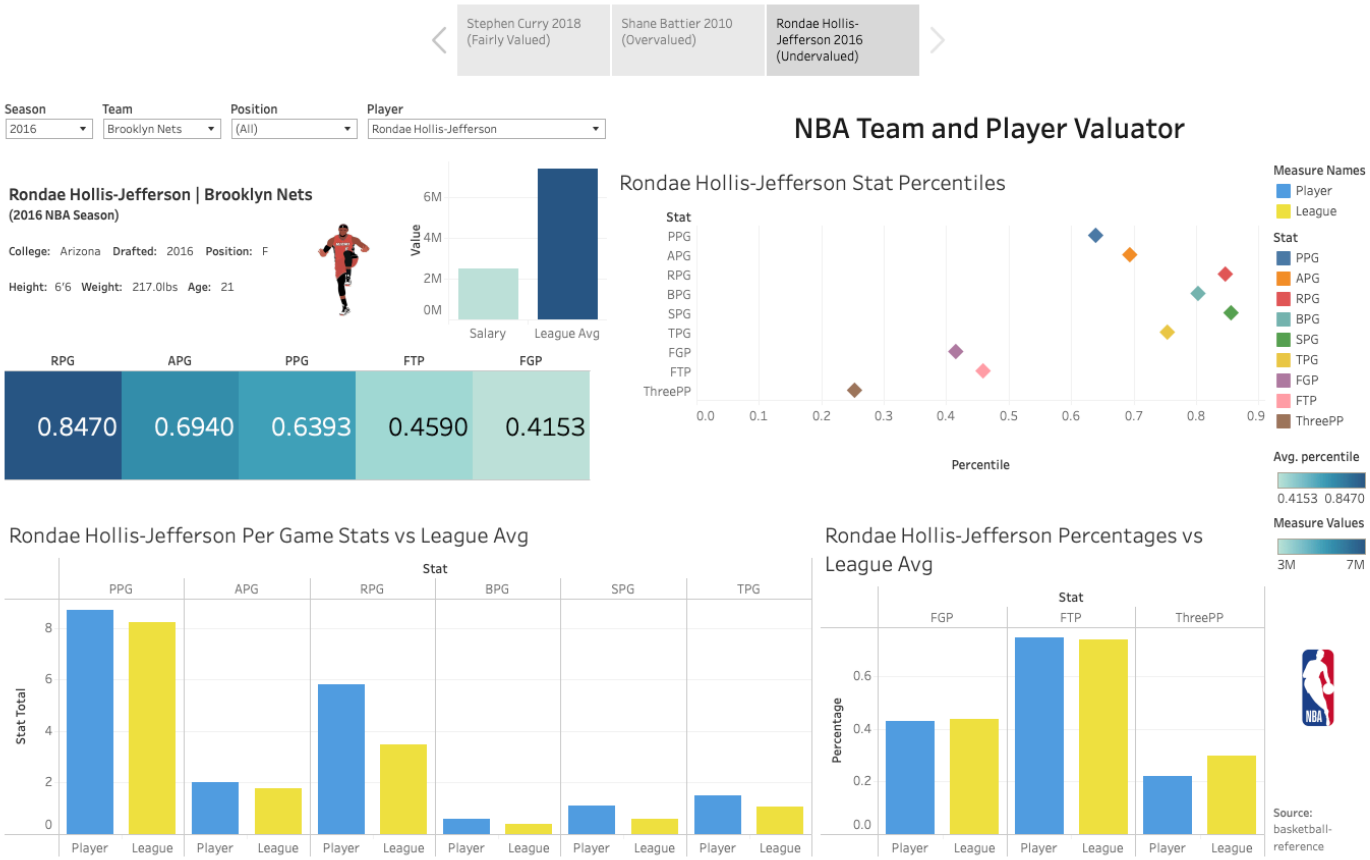
Job ID 6244a3ee-c5c0-4a81-9a9c-7ff109406115 successfully executed: 100%

Downloading: 100%

	Player	Team	Season	pos	birth_date	colleges	height_cm	weig
0	Cezary Trybański	Memphis Grizzlies	2002	None	NaT	None	NaN	
1	D.J. Mbenga	Los Angeles Lakers	2007	None	NaT	None	NaN	
2	Aleksandar Radojević	Denver Nuggets	2000	None	NaT	None	NaN	
3	Zoran Planinić	New Jersey Nets	2002	None	NaT	None	NaN	
4	Levi Randolph	Cleveland Cavaliers	2019	None	NaT	None	NaN	
...	...	...	...	...	...	...	...	
99022	Luther Head	Sacramento Kings	2010	G	1982-11-26	Illinois	190.50	
99023	James Anderson	San Antonio Spurs	2010	G-F	1989-03-25	Oklahoma State	198.12	
99024	Devin Ebanks	Los Angeles Lakers	2010	F	1989-10-28	West Virginia	205.74	


Stephen Curry 2018 (Fairly Valued)
Shane Battier 2010 (Overvalued)
Rondae Hollis-Jefferson 2016 (Undervalued)
Season
2018
Team
Golden State W...
Position
(All)
Player
Stephen Curry
NBA Team and Player Valuator
Stephen Curry | Golden State Warriors
(2018 NBA Season)
College: Davidson Drafted: 2010 Position: G
Height: 6'2 Weight: 185.0lbs Age: 30
40M
30M
20M
10M
0M
Salary
League Avg
PPG
FTP
APG
RPG
FGP
0.9952
0.9683
0.9445
0.8320
0.6244
Carmelo Anthony & Stephen Curry Stat Percentiles
Stat
PPG
APG
RPG
BPG
SPG
TPG
FGP
FTP
ThreePP
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Percentile
Measure Names
Player
League
Stat
PPG
APG
RPG
BPG
SPG
TPG
FGP
FTP
ThreePP
Avg. percentile
0.6244 0.9952
Measure Values
8M 40M
Carmelo Anthony & Stephen Curry Per Game Stats vs League Avg
Stat
PPG
APG
RPG
BPG
SPG
TPG
Stat Total
20
10
0
Player
League
Carmelo Anthony & Stephen Curry Percentages vs League Avg
Stat
FGP
FTP
ThreePP
Percentage
0.8
0.6
0.4
0.2
0.0
Source: basketball-reference


Stephen Curry 2018 (Fairly Valued)
Shane Battier 2010 (Overvalued)
Rondae Hollis-Jefferson 2016 (Undervalued)
Season
2010
Team
Memphis Grizzli...
Position
(All)
Player
Shane Battier
NBA Team and Player Valuator
Shane Battier | Memphis Grizzlies
(2010 NBA Season)
College: Duke Drafted: 2002 Position: F
Height: 6'8 Weight: 220.0lbs Age: 32
Value
6M
4M
2M
0M
Salary
League Avg
FTP
RPG
APG
PPG
FGP
0.7192
0.6902
0.5525
0.4239
0.3895
Shane Battier Stat Percentiles
Stat
PPG
APG
RPG
BPG
SPG
TPG
FGP
FTP
ThreePP
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Percentile
Measure Names
Player
League
Stat
PPG
APG
RPG
BPG
SPG
TPG
FGP
FTP
ThreePP
Avg. percentile
0.3895 0.7192
Measure Values
7M 7M
Shane Battier Per Game Stats vs League Avg
Stat
PPG
APG
RPG
BPG
SPG
TPG
Stat Total
8
6
4
2
0
Player
League
Shane Battier Percentages vs League Avg
Stat
FGP
FTP
ThreePP
Percentage
0.8
0.6
0.4
0.2
0.0
Source: basketball-reference



After we're done, this is what the [dashboard] results to. The base function is to valuate players, but our data actually works out to where we can evaluate on the positional and team level. This means that the dashboard itself can actually be utilized by any actor (player, team, organization) be it for valuating a player, finding team weaknesses or player weaknesses, or defending your own value as a player.

Key Findings and Conclusion

If we take a step back and look at the queries in the EDA section, we managed to better understand players, teams, and financials.

Business Value of player queries:

The player queries can be utilized by opposing teams and front offices. For example, the top defensive players query can help coaches plan their matchups ahead of time so that the team will be able to make the proper adjustments when they have to play that team with the specific player. For a front office or media use case example, the players with the top per game stats query can be utilized to schedule primetime game slots.

Business Value of Team queries:

The team queries can be utilized by the teams and front offices themselves, and the NBA as an organization. For example, a top 5 team in average attendance like the Knicks might want to use this information to continue utilizing their current marketing strategies or pricing in order to maintain that metric.

Business Value of Financial queries:

The Financial queries are very useful for front office operations. The information provided by these can help make data-driven financial decsions. For example, the relative per game stats and relative salary query allows front offices to see what players are underperforming or overperforming based on their current salary. It's a great way to narrow down what players you are looking at in the trade window or in the free agency.

Dashboard Value Considering the capabilities of the dashboard, we have already mentioned that there is use on the Player, Team and Organizaitonal level. Though the dashboard only reaches 2019 (due to the limitations of the data), it serves as an excellent proof of concept to be utilized on current and future data. This will allow data informed decision making at all levels of the NBA.

References

- Kaggle Dataset
- Webscraped Data
- Alter Table Syntax (Snowflake)
- Alter Table Syntax (Clickhouse)
- ERD Tool

Data Dictionaries

`boxscore`

Field Name	Description
game_id (Game ID)	A unique identifier for each game.
teamName (Team Name)	The name of the team that the player is associated with.
playerName (Player Name)	The name of the player.
MP (Minutes Played)	The total minutes the player played in the game.
FG (Field Goals Made)	The number of field goals made by the player.
FGA (Field Goal Attempts)	The number of field goal attempts by the player.
ThreeP (3-Points Made)	The number of 3-point field goals made by the player.
ThreePA (3-Point Attempts)	The number of 3-point field goal attempts by the player.
FT (Free Throws Made)	The number of free throws made by the player.
FTA (Free Throw Attempts)	The number of free throw attempts by the player.
ORB (Offensive Rebounds)	The number of offensive rebounds secured by the player.
DRB (Defensive Rebounds)	The number of defensive rebounds secured by the player.
TRB (Total Rebounds)	The total number of rebounds secured by the player.
AST (Assists)	The number of assists made by the player.
STL (Steals)	The number of times the player stole the ball.
BLK (Blocks)	The number of shots blocked by the player.
TOV (Turnovers)	The number of times the player turned over the ball.
PF (Personal Fouls)	The number of personal fouls committed by the player.
PTS (Points)	The total points scored by the player.
+/- (Plus/Minus)	The player's plus/minus statistic.
isStarter (Is Starter)	Indicates if the player was a starter for the game.

`coaches`

Column Name	Desc
Team	Represents the abbreviation or code for the NBA team.
League	Indicates the league affiliation, which is NBA (National Basketbal
regular_winloss_ratio (Win-Loss Ratio in the Regular Season)	Represents the team's performance in the regular season, indicat
playoff_winloss_ratio (Win-Loss Ratio in the Playoffs)	Represents the team's performance in the playoff season, indicat
Accolade	notes about the team's performance, achievements, or notable e
Name	Specifies the name of the head coach leading the team during a
Coach_id	Represents a unique identifier for the coach.
seasonStartYear	Indicates the starting year of the NBA season.
role	Specifies the role or type of coaching position, such as "Head Co
season_rank	Represents the finishing position or rank of the team in a particul
playoff_games_coached	Indicates the total number of playoff games played under the coa
playoff_wins	Represents the number of playoff games won by the team under
playoff_losses	Represents the number of playoff games lost by the team under
Age	Refers to the age of the head coach during a particular season.
regular_games_coached	Indicates the total number of regular-season games played by th
regular_season_wins	Represents the number of regular-season games won by the tear
regular_season_losses	Represents the number of regular-season games lost by the tear
W500	Indicates whether the team had a win-loss ratio greater than .500

`Player_info`

Column Name	Description
player_name	The name of the player
from_year	The player's first year in the NBA
to_year	The player's last year in the NBA
pos	The player's position that they play
height_cm	The player's height measured in centimeters
weight_kg	The player's weight measured in kilograms
birth_date	The player's date of birth
colleges	Which college the player attended (if null they either came from high school or international play)

`salaries`

Column Name	Description
playerName (Player Name)	The name of the player.
seasonStartYear (Season Year)	The year of the season for which the salary is applicable.
salary (Salary)	The salary earned by the player for the season.
inflationAdjSalary (Inflation Adjusted Salary)	The player's salary adjusted for inflation to reflect current value comparison.

`jersey_patch_sponsorships`

Column Name	Description
Team	Team name
jersey_rev_mil	Estimate of revenue generated by sponsorships (since inception)

`operating_income`

Column Name	Description
Team	Team name
Year	Year of finances
Operating Income	Operating Income ($ Mil)

`rev_by_team`

Column Name	Description
Team	Team name
Year	Year of finances
Revenue	Revenue ($ Mil)

`ticket_revenue`

Column Name	Description
Team	Team name
Year	Year of finances
Ticket Revenue	Average revenue per ticket ($)

`team_abrev`

Column Name	Description
Team	Team name
Abbreviation	Three letter abbreviation for Team