**NBA SALARY INSIGHTS**

**PROJECT**

**September 2019**

**CLEANUP PROCESS**

[**SALARY CAP DataFrame (by Sai)**](#_p7jw42s4ioq4) **2**

[**PLAYER PERFORMANCE CLEANUP (by Melissa):**](#_2nqwut3lch55) **4**

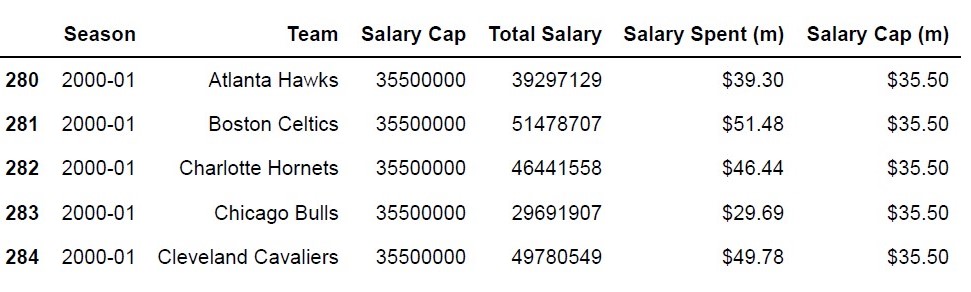
[**Position Vs Salary Dataframe (by Deepen):**](#_ujtaerx7hn8p) **8**

[**AGE vs Salary – DATA CLEANING (by Mary)**](#_tadyfwrma37g) **9**

# SALARY CAP DataFrame (by Sai)

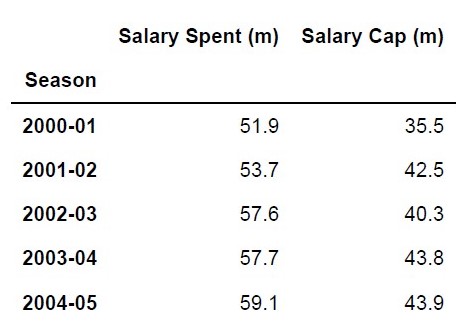
Cleanup Process:

* Dropped NaN values and unnamed columns from the two datasets
* Specified the years using
* Formatted “Salary Cap”, “Salary Spent”, “Salary” to “Salary Spent (m)”, “Salary Cap (m)” and “Avg Player Salary (m) to simplify calculations further down





* Converted Object into Float to do further calculations
* Groupby by seasons and specify the columns using



* Finally Merge the two datasets

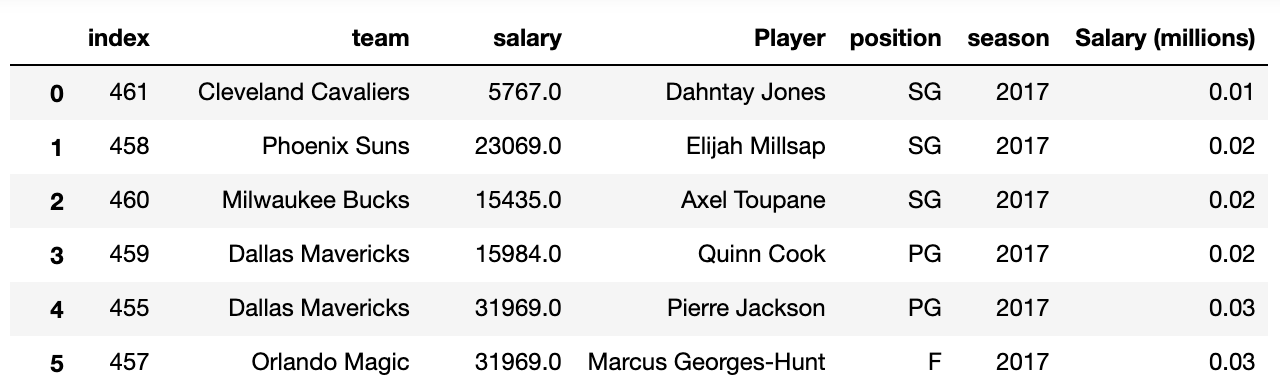


# PLAYER PERFORMANCE CLEANUP (by Melissa):

NBASalaryData03-17.csv. Contains salary data from years 2002-2017.

Cleanup:

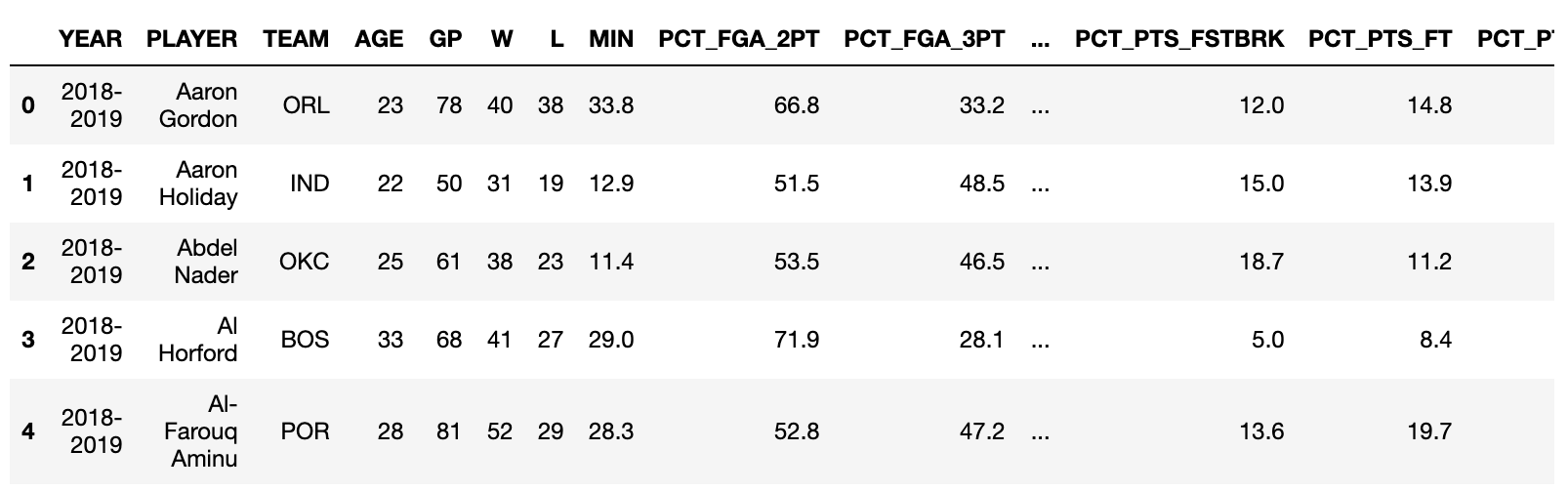
* Dropped NaN values and unnamed columns
* Changed “player” to “Player” to make it easier to merge with other files.
* Chose only the 2016-2017 season, and changed “2016-2017” to “2017” for easier merging
* Converted “salary” to “Salary(millions)” to simplify calculations





Created a new DataFrame to compare players with their salaries.

nba\_shot\_types.csv



Cleanup:

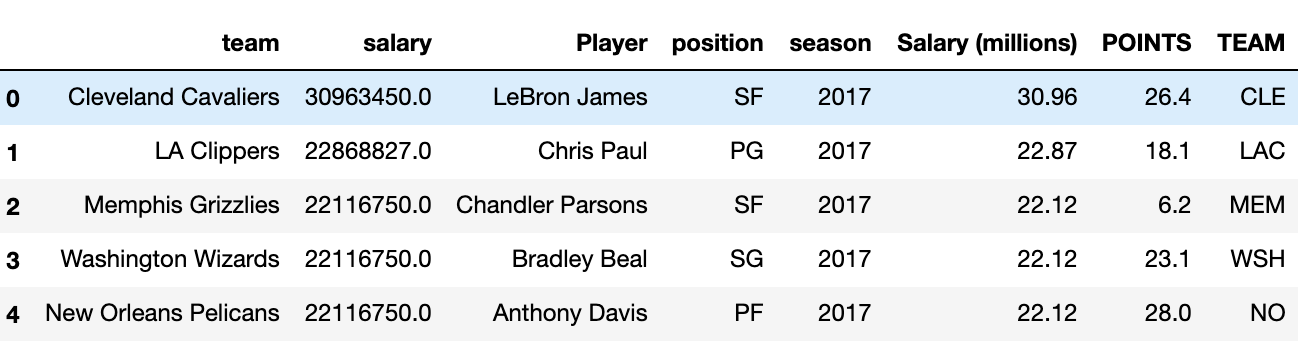
* Dropped NaN values and unnamed columns
* Changed “player” to “Player” to make it easier to merge with other files.
* Chose only the YEAR 2016-2017, and changed “2016-2017” to “2017” for easier merging
* Created a new DataFrame to include only the Player, Year, Team and Min
* Merged the new DataFrame with the salary DataFrame



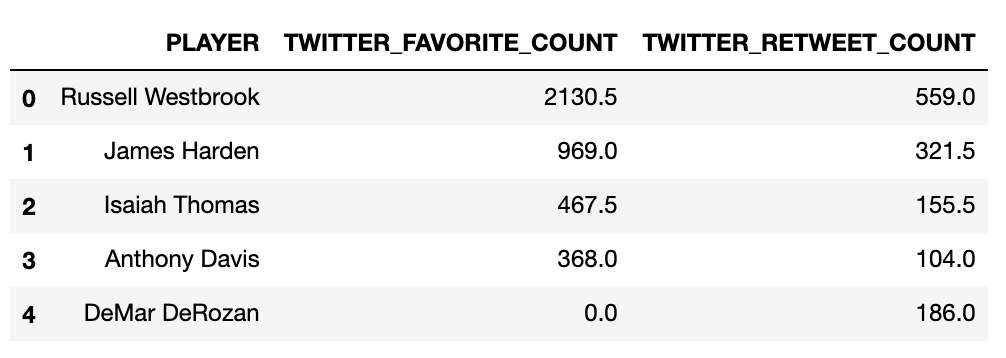
nba\_2017\_nba\_players\_with\_salary.csv

Cleanup:

* Dropped NaN values and unnamed columns
* Changed “player” to “Player” to make it easier to merge with other files
* Created a new DataFrame to include only the Player, Points, and Team
* Merged the new DataFrame with the salary DataFrame



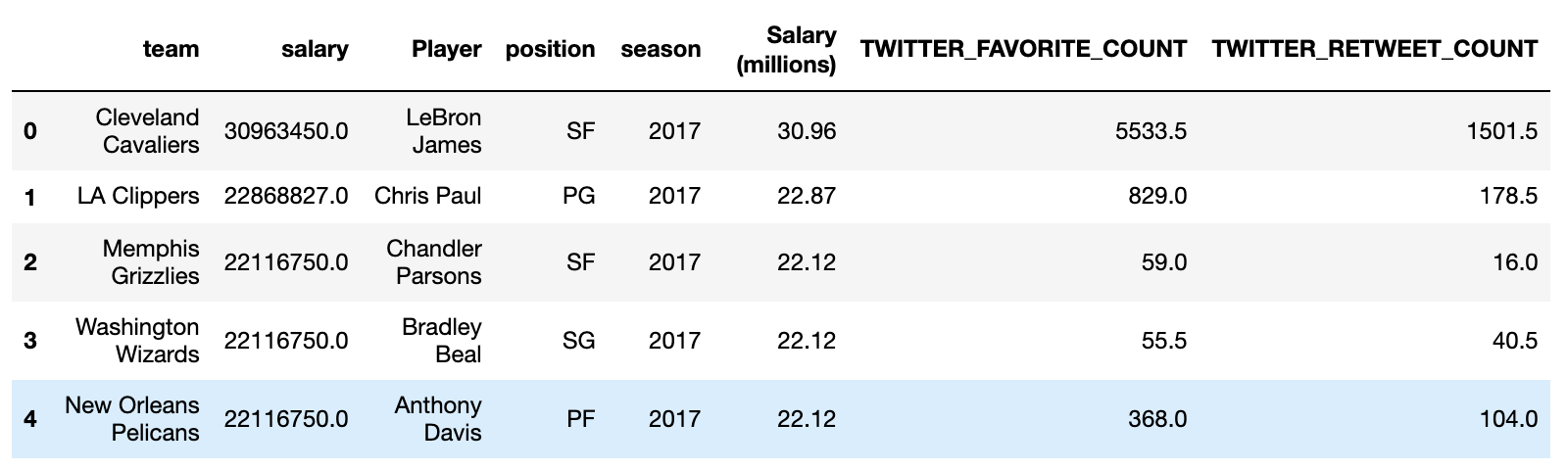
Twitter DataFrame



Twitter data merged with salary data.

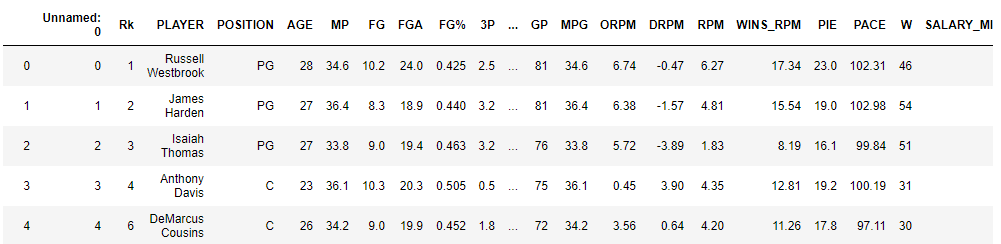
Cleanup:

* Dropped NaN values and unnamed columns
* Changed “player” to “Player” to make it easier to merge with other files
* Merged the DataFrame with the salary DataFrame



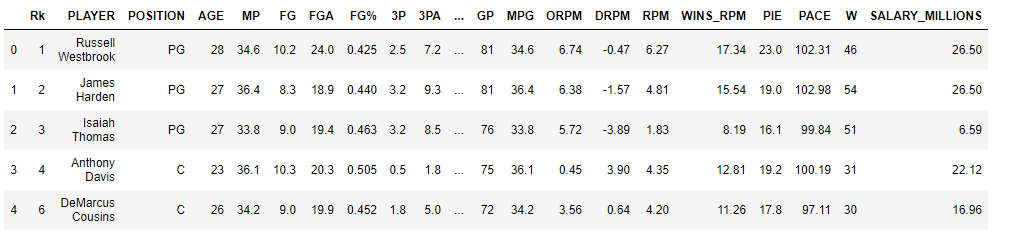
# Position Vs Salary Dataframe (by Deepen):

Nba\_2017\_nba\_players\_with\_salary.csv



Cleanup:

* Dropped NaN values and unnamed columns
* Consolidated position “PF-C” into “PF” for Power Forwards



# AGE vs Salary – DATA CLEANING (by Mary)

#1 Found article on top paid athletes at

<https://nypost.com/2017/06/07/the-top-25-highest-paid-athletes-in-the-world-for-2017-are/>

Format of data not good, so I followed another link to Forbes article

<https://www.forbes.com/sites/kurtbadenhausen/2017/06/07/ronaldo-lebron-top-the-worlds-highest-paid-athletes-of-2017/#bda024d4a1c9>

This provided a link to a table with the World’s Highest Paid Athletes

[**See Highest Paid Athletes List Here**](http://www.forbes.com/athletes/list)

<https://www.forbes.com/athletes/list/>

So I copied and pasted the list from this webpage into EXCEL, added column for the original URL and then saved as CSV (input/TopPaidAthletes2017.csv)

1. Sorted the data by rank

2. Set the index to rank

3. Grouped by sport and country and got the mean

4. Head to get top 5 sports

5. Grouped by sport and count names to get

#2 Found some player data from Keggle (NBA Player Stats since 1950)

<https://www.kaggle.com/drgilermo/nba-players-stats>

player\_data.csv

Calculated years in league (year end MINUS year start)

Then grouped the dataset by years in league

#3 Decided to focus on NBA player data for 2017

Used Kaggle Social Power Performance Effect 2017 NBA

<https://www.kaggle.com/liujiaqi/nba-player-social-power-performance-effect>

1. Got rid of bogus “Unnamed” column

2. Dropped columns that were not related to age and salary

3. Found mean, other statistics and count on the salary in the dataset

4. Renamed some columns

5. Calculated standard error on salary by age

6. Did linear regression and correlation