Creating a Data Pipeline

# Project context

* This report describes the extraction, transformation and loading of tabular statistical data from Basketball Reference
* The data in question is per-game statistics from the national Basketball Association (NBA)
* The NBA is the top professional basketball league in North America. It was founded in 1947 and is currently home to the best basketball players in the world.
* Basketball Reference is an online depository of basketball statistics, game logs etc for different basketball leagues.

# Python Modules

* Pandas
* Bs4 (BeautifulSoup)
* sqlalchemy
* requests
* numpy

# Skills Showcased:

* Ability to implement the Object-Oriented Programming paradigm and utilise inheritance
* Webscraping using BeautifulSoup
* Establish a connection between Python and a local MySQL server using sqlalchemy
* transfer pandas dataframes to tables in a SQL database
* read SQL tables as pandas dataframe
* Query SQL tables using SQL language in Python
* Query pandas dataframe using indexing, filtering etc

# Project Outcome

In undergoing this project, a per-game stats extractor tool was created (see file: ” data\_pipeline\_bball\_reference\_to\_SQL\_or\_EXCEL.py”. it scrapes the tabular data from basketball reference and stores in either an EXCEL workbook or a table in the local SQL server. A flowchart has been made explaining how to use the tool (see page 3)

Note:

* This report is meant to showcase Data Engineering skills hence the following pages will highlight the data extraction, cleaning and loading undergone using python.
* It is part of a larger ongoing data science project which will use the extracted data with linear regression modelling to establish correlation between different player statistics and per-game scoring

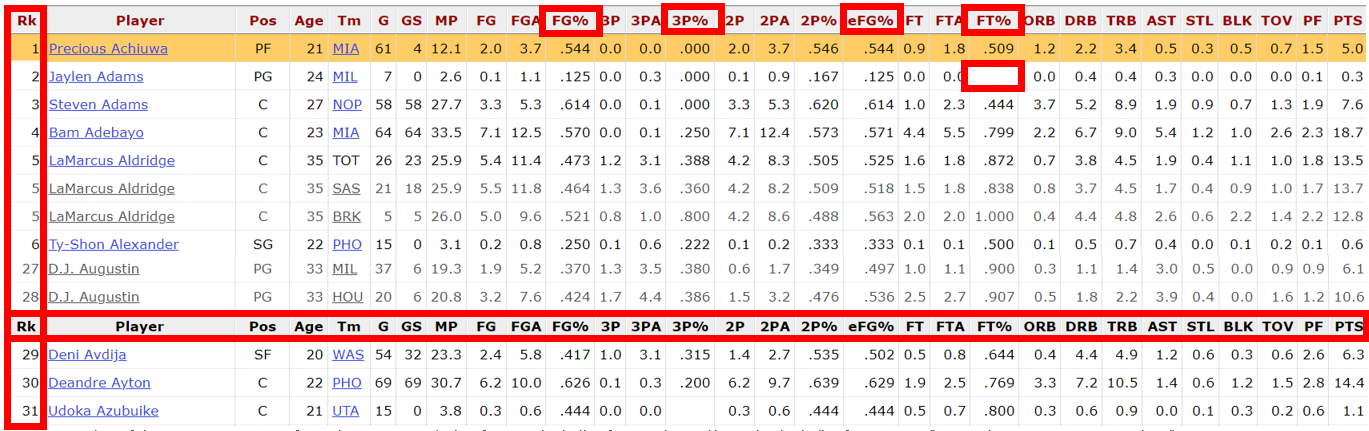
# Project Steps

1. Creating a class that scrapes and cleans points per-game data from basketball reference using Beautiful Soup (file : “NBA\_webscraping\_OOP.py”)
2. Inheriting from this class and establishing loading processes to a local SQL server and/or to an Excel workbook called “Data\_store.xlsx” (file: ” data\_pipeline\_bball\_reference\_to\_SQL\_or\_EXCEL.py”)

## Step A: Extraction and cleaning

Highlighted in figure 1 are some of the features of the extracted table that had to be cleaned. In addition, to the cleaning process a year column was added to keep track of the year where the data was extracted.

Figure 1: A slice of the per-game statistics the 2020-2021 season (2020-21 NBA Player Stats: Per Game | Basketball-Reference.com, 2021)



Percent symbol replaced with ‘\_per’ to make headings SQL friendly

Empty field replaced with ‘nan’ as SQL did not accept empty fields in FLOAT columns

Rank Column removed

Row of repeated columns headings removed

Rows where the Team column (‘Tm’) = ‘TOT’ aggregate the statistics of players who have played for multiple teams in a given season. These rows were removed in the transformation process

Year column added to show what year the data is taken from

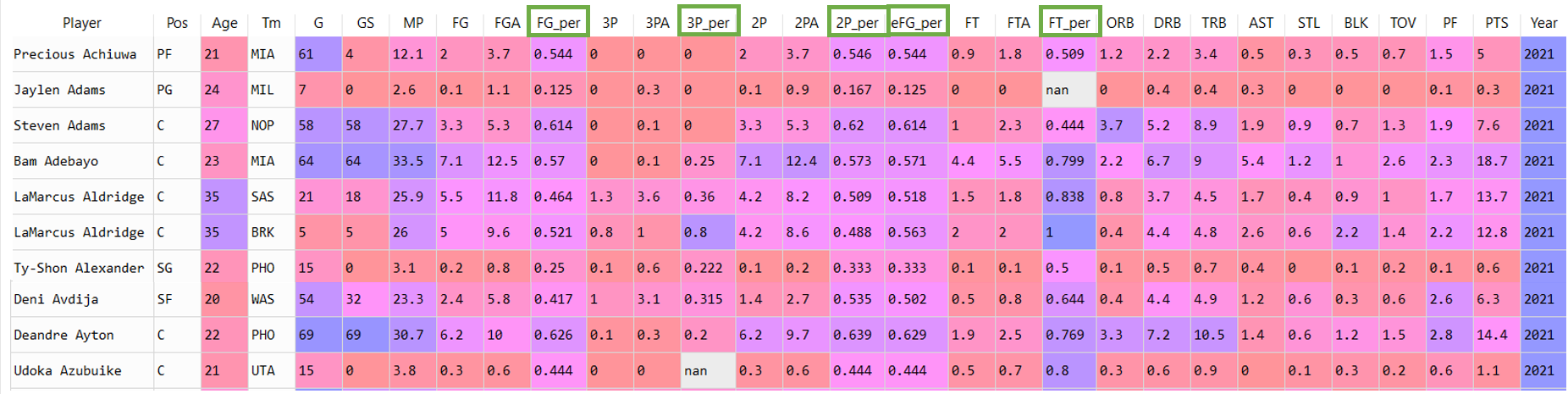


Figure 2: Dataframe of the transformed data shown in figure 1

**Heading abbreviations Explained:**

* Pos = Position; Tm = Team; MP = Minutes Played
* G = Games Played; GS = Games Started
* FG = Field Goal(s) (made); FGA = ‘’ Attempted
* FG\_per = ‘’ percentage; eFG\_per = Effective ‘’
* 3P = Three pointer(s) (made); 3PA = ‘’ Attempted
* 3P\_per = ‘’ percentage
* 2P = Two Pointer(s) (made); 2PA = …; 2P\_per = …
* FT = Free Throws; FTA = … ; FT\_per = …
* ORB = Offensive Rebound; DRB = Defensive ‘’
* TBR = Total ‘’; AST = Assists; BLK = Blocks
* TOV = Turnovers; PF = personal fouls; PTS = points

## Step B: Loading

To finish the pipeline certain actions were put in place in SQL and Excel to ensure smooth transfer of data from dataframe to the storage location:

OPEN FILE: ” data\_pipeline\_bball\_reference\_to\_SQL\_or\_EXCEL.py”

Scroll down to Line 66: Testing

Where would you like the data to be stored?

Specify the year range you would like to extract

Specifying one year

Specifying Multiple years

Write the year in a list object:

I.e. [Year1]

e.g. <year\_range = [2019]>

This extracts data for 2019

Write the range of years in a list object:

i.e. [first-year, last\_year ]

e.g. <year\_range = [2018,2020] >

This extracts data for 2018, 2019 and 2020

EXCEL

SQL

Both

Write:

<save\_to = "Excel">

Write:

<save\_to = "SQL">

Write:

<save\_to = "Both">

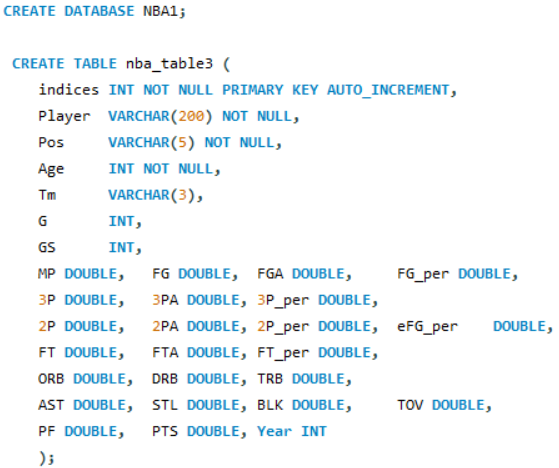
* <host = "…" > #defines the location of the MySQL server
* <username ="…" > # the username for MySQL is ‘root’ by default
* <password = "…" >
* <dbname = "…"> #name of database created in substep B
* <table\_name = "nba\_table3"> #name of table created ‘’

**Run Script**

1. For loading into SQL server:

* A database was created to store the transformed data (‘nba1’)
* A table was created with each column assigned an appropriate datatype (‘nba\_table3’)

Figure 3: The SQL table created to store the extracted and transformed data (available in the file: 'Headings\_for\_per\_game\_stats\_data.sql'



1. For loading into Excel

* The required columns were simply written into an Excel File (“Data\_Store.xlsx’)



Figure 4: Some of the headings of the Excel FIle 'Data\_Store.xlsx'

## How the Data Pipeline tool was used in project

The data pipeline tool was used to extract data from 1990-2021 from Basketball Reference. To not overload the servers the extraction was done over two days with tables funnelled in at 5-year chunks every few hours. In total over 16,000 rows of data was extracted. (16221)

## How to use the Data Pipeline Tool

# References

Basketball-Reference.com. 2021. *2020-21 NBA Player Stats: Per Game | Basketball-Reference.com*. [online] Available at: <https://www.basketball-reference.com/leagues/NBA\_2021\_per\_game.html> [Accessed 10 September 2021].