Project – 1 Documentation

**Scraping and Analysis of Stastistics of Sports Players from Real-Time Webpage**

**OVERVIEW :**A project that gives you a better understanding of scraping data from websites and how to analyse them. Usage of various libraries as NumPy, Mat Plot, Pandas.

In the course of completing the project, you use the web scraping function, converting the extracted data into a pandas data Frame, and Storing the analysed data.

**Problem Statement**

Web scrape basketball statistics from Wikipedia of some of the greatest basketball players and export it as a CSV file format.

**Software Requirements**

1. Programming Language : Python

2. Environemnt: Jupyter Notebooks / Google Collab

3. Database: CSV(export type)

4. Operation System: Windows XP or above

5. Librarires Used: Beautiful Soup, requests, Pandas, NumPy, boto3 ,Matplotlib, display

1. **Open a New Notebook and import the required libraires**

|  |  |
| --- | --- |
|  | import bs4  import requests  import pandas as pd  import numpy as np  !pip install boto3  import boto3  import matplotlib.pyplot as plt  from IPython.display import display |

Description: importing the libraries to read the html code, process the data, and to plot the graphs

1. **Reading the webpage**

def get\_basketball\_stats(link='https://en.wikipedia.org/wiki/Michael\_Jordan'):

response = requests.get(link)

soup = bs4.BeautifulSoup(response.text, 'html.parser')

Description: read the web pages one by one from the list

1. **Main Function Process**

|  |  |
| --- | --- |
| table = soup.find(class\_='wikitable sortable')  headers = table.tr  titles = headers.find\_all('abbr')  data = {title['title']: [] for title in titles}  for row in table.find\_all('tr')[1:]:  for key, a in zip(data.keys(),row.find\_all('td')[2:]):  data[key].append(''.join(c for c in a.text if (c.isdigit() or c == '.')))  Min = min([len(x) for x in data.values()])  for key in data.keys():  data[key] = list(map(lambda x: float(x), data[key][:Min]))  return data  Description: here the data cleaning takes places that is removes the unwanted data and restores the data and the data is stored in dictioneries   1. **Declaring links and names of the personals to scrap the data** |  |
| links=['https://en.wikipedia.org/wiki/Michael\_Jordan'\  ,'https://en.wikipedia.org/wiki/Kobe\_Bryant'\  ,'https://en.wikipedia.org/wiki/LeBron\_James'\  ,'https://en.wikipedia.org/wiki/Stephen\_Curry']  names=['Michael Jordan','Kobe Bryant','Lebron James','Stephen Curry'] |  |

michael\_jordan\_dict = get\_basketball\_stats(links[0])

kobe\_bryant\_dict = get\_basketball\_stats(links[1])

lebron\_james\_dict = get\_basketball\_stats(links[2])

stephen\_curry\_dict = get\_basketball\_stats(links[3])

mj\_table = pd.DataFrame(michael\_jordan\_dict)

kb\_table = pd.DataFrame(kobe\_bryant\_dict)

lj\_table = pd.DataFrame(lebron\_james\_dict)

sc\_table = pd.DataFrame(stephen\_curry\_dict)

list\_table =[mj\_table, kb\_table, lj\_table, sc\_table]

i = 0

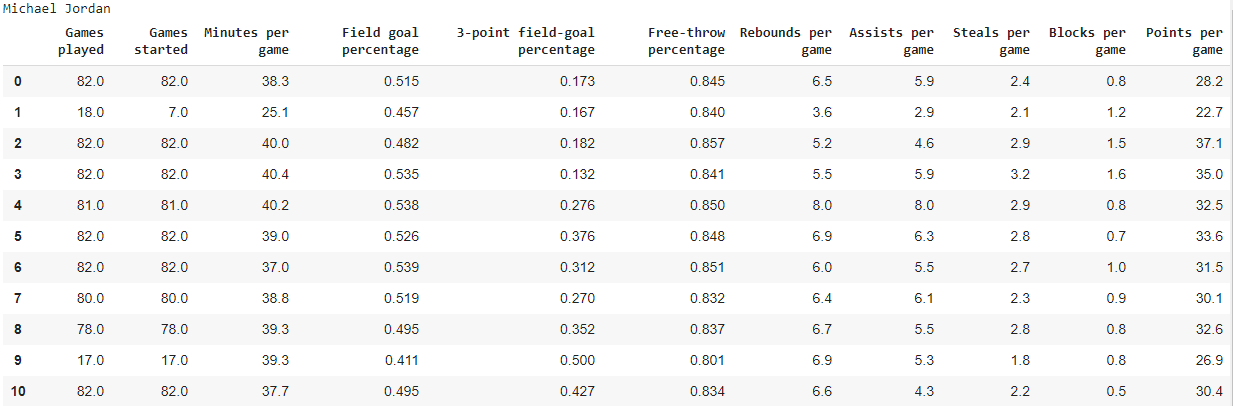
for name in names:

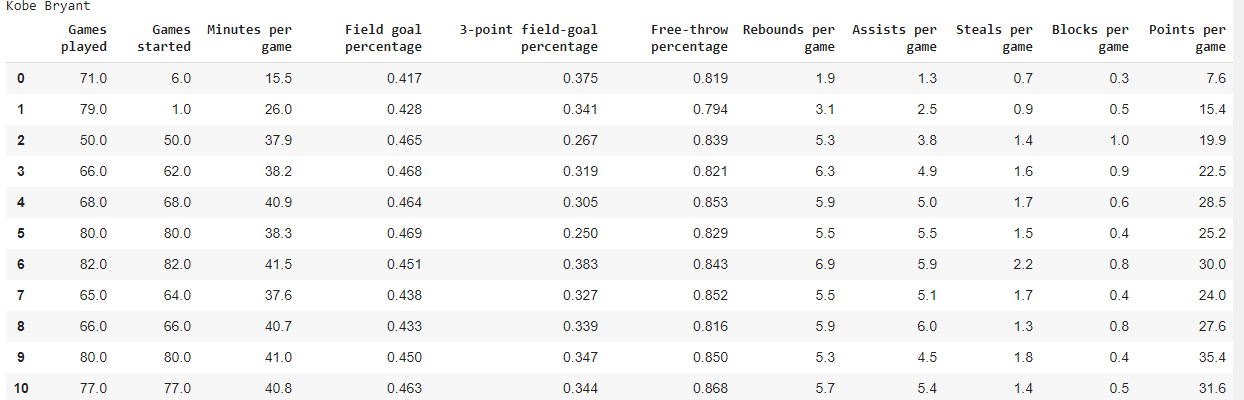
    print(name)

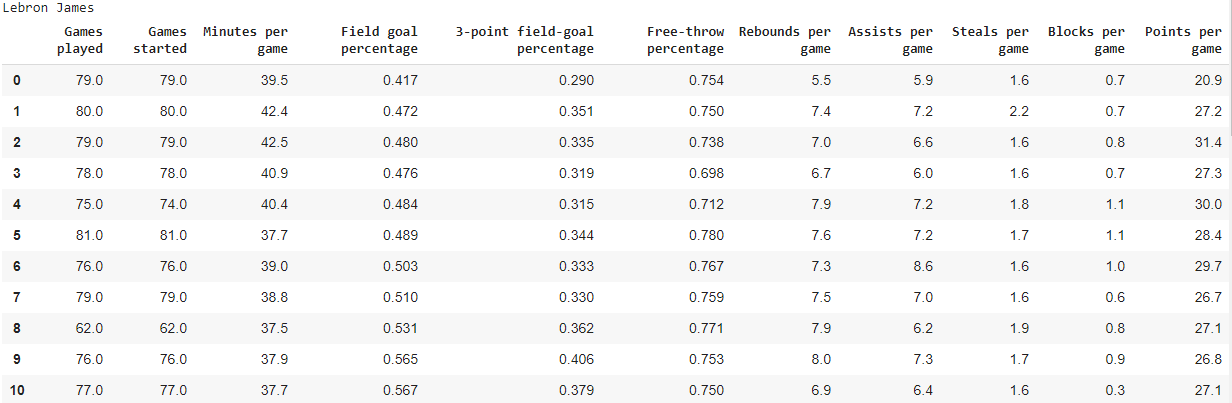
    display(list\_table[i])

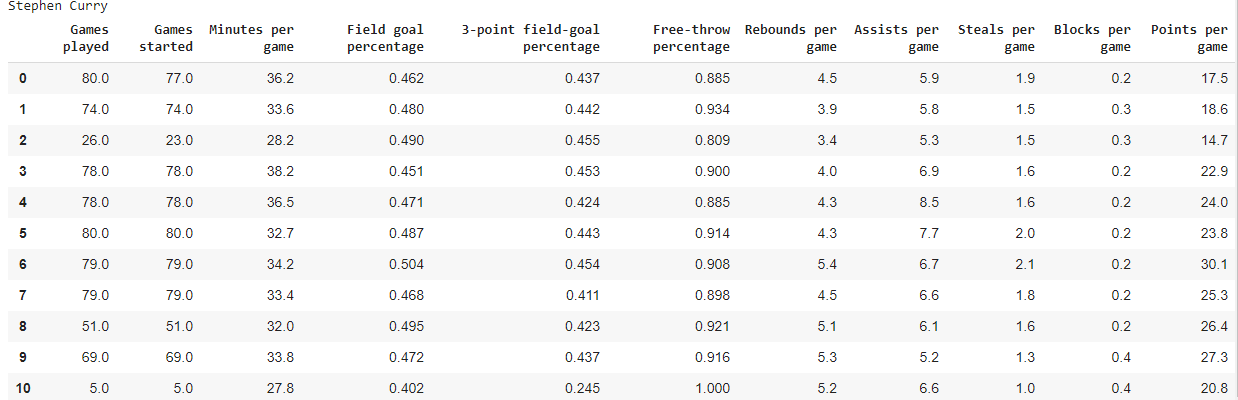
    i += 1

**Output:**









Description: declaring the links in the list, each link is stored in each dictionery, and the data is stored in the dataframes(rows and columns)

**6.Making a plot using matplotlib**

j = 0

for name in names:

    plt.plot(list\_table[j][['Points per game']],label=name)

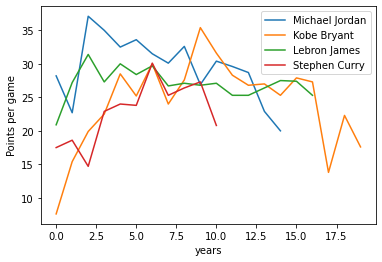
    plt.legend()

    plt.xlabel('years')

    plt.ylabel('Points per game')

    j += 1

**Output:**

****

Description: here all the data frames are compared and visualised graphically.

|  |
| --- |
|  |

**Conclusion**

Few important libraries were imported for data preprocessing and visualisation. later we collected(Web scrapped) and preprocess the data, now we converted the raw data into dataframes, then using matplotlib we visualised the data.

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |