Day15_Matplotlib_using_IPL_Data_2

June 4, 2025

[1]:

```
#Import numpy
import numpy as np

#Seasons
Seasons =␣
 ↪["2015","2016","2017","2018","2019","2020","2021","2022","2023","2024"]
Sdict = {"2015":0,"2016":1,"2017":2,"2018":3,"2019":4,"2020":5,"2021":6,"2022":
 ↪7,"2023":8,"2024":9}

#Players
Players =␣
 ↪["Sachin","Rahul","Smith","Sami","Pollard","Morris","Samson","Dhoni","Kohli","Sky"]
Pdict = {"Sachin":0,"Rahul":1,"Smith":2,"Sami":3,"Pollard":4,"Morris":
 ↪5,"Samson":6,"Dhoni":7,"Kohli":8,"Sky":9}

#Salaries
Sachin_Salary =␣
 ↪[15946875,17718750,19490625,21262500,23034375,24806250,25244493,27849149,30453805,23500000]
Rahul_Salary =␣
 ↪[12000000,12744189,13488377,14232567,14976754,16324500,18038573,19752645,21466718,23180790]
Smith_Salary =␣
 ↪[4621800,5828090,13041250,14410581,15779912,14500000,16022500,17545000,19067500,20644400]
Sami_Salary =␣
 ↪[3713640,4694041,13041250,14410581,15779912,17149243,18518574,19450000,22407474,22458000]
Pollard_Salary =␣
 ↪[4493160,4806720,6061274,13758000,15202590,16647180,18091770,19536360,20513178,21436271]
Morris_Salary =␣
 ↪[3348000,4235220,12455000,14410581,15779912,14500000,16022500,17545000,19067500,20644400]
Samson_Salary =␣
 ↪[3144240,3380160,3615960,4574189,13520500,14940153,16359805,17779458,18668431,20068563]
Dhoni_Salary =␣
 ↪[0,0,4171200,4484040,4796880,6053663,15506632,16669630,17832627,18995624]
Kohli_Salary =␣
 ↪[0,0,0,4822800,5184480,5546160,6993708,16402500,17632688,18862875]
Sky_Salary =␣
 ↪[3031920,3841443,13041250,14410581,15779912,14200000,15691000,17182000,18673000,15000000]
```

```
#Matrix
Salary = np.array([Sachin_Salary, Rahul_Salary, Smith_Salary, Sami_Salary,␣
 ↪Pollard_Salary, Morris_Salary, Samson_Salary, Dhoni_Salary, Kohli_Salary,␣
 ↪Sky_Salary])

#Games
Sachin_G = [80,77,82,82,73,82,58,78,6,35]
Rahul_G = [82,57,82,79,76,72,60,72,79,80]
Smith_G = [79,78,75,81,76,79,62,76,77,69]
Sami_G = [80,65,77,66,69,77,55,67,77,40]
Pollard_G = [82,82,82,79,82,78,54,76,71,41]
Morris_G = [70,69,67,77,70,77,57,74,79,44]
Samson_G = [78,64,80,78,45,80,60,70,62,82]
Dhoni_G = [35,35,80,74,82,78,66,81,81,27]
Kohli_G = [40,40,40,81,78,81,39,0,10,51]
Sky_G = [75,51,51,79,77,76,49,69,54,62]
#Matrix
Games = np.array([Sachin_G, Rahul_G, Smith_G, Sami_G, Pollard_G, Morris_G,␣
 ↪Samson_G, Dhoni_G, Kohli_G, Sky_G])

#Points
Sachin_PTS = [2832,2430,2323,2201,1970,2078,1616,2133,83,782]
Rahul_PTS = [1653,1426,1779,1688,1619,1312,1129,1170,1245,1154]
Smith_PTS = [2478,2132,2250,2304,2258,2111,1683,2036,2089,1743]
Sami_PTS = [2122,1881,1978,1504,1943,1970,1245,1920,2112,966]
Pollard_PTS = [1292,1443,1695,1624,1503,1784,1113,1296,1297,646]
Morris_PTS = [1572,1561,1496,1746,1678,1438,1025,1232,1281,928]
Samson_PTS = [1258,1104,1684,1781,841,1268,1189,1186,1185,1564]
Dhoni_PTS = [903,903,1624,1871,2472,2161,1850,2280,2593,686]
Kohli_PTS = [597,597,597,1361,1619,2026,852,0,159,904]
Sky_PTS = [2040,1397,1254,2386,2045,1941,1082,1463,1028,1331]
#Matrix
Points = np.array([Sachin_PTS, Rahul_PTS, Smith_PTS, Sami_PTS, Pollard_PTS,␣
 ↪Morris_PTS, Samson_PTS, Dhoni_PTS, Kohli_PTS, Sky_PTS])
```

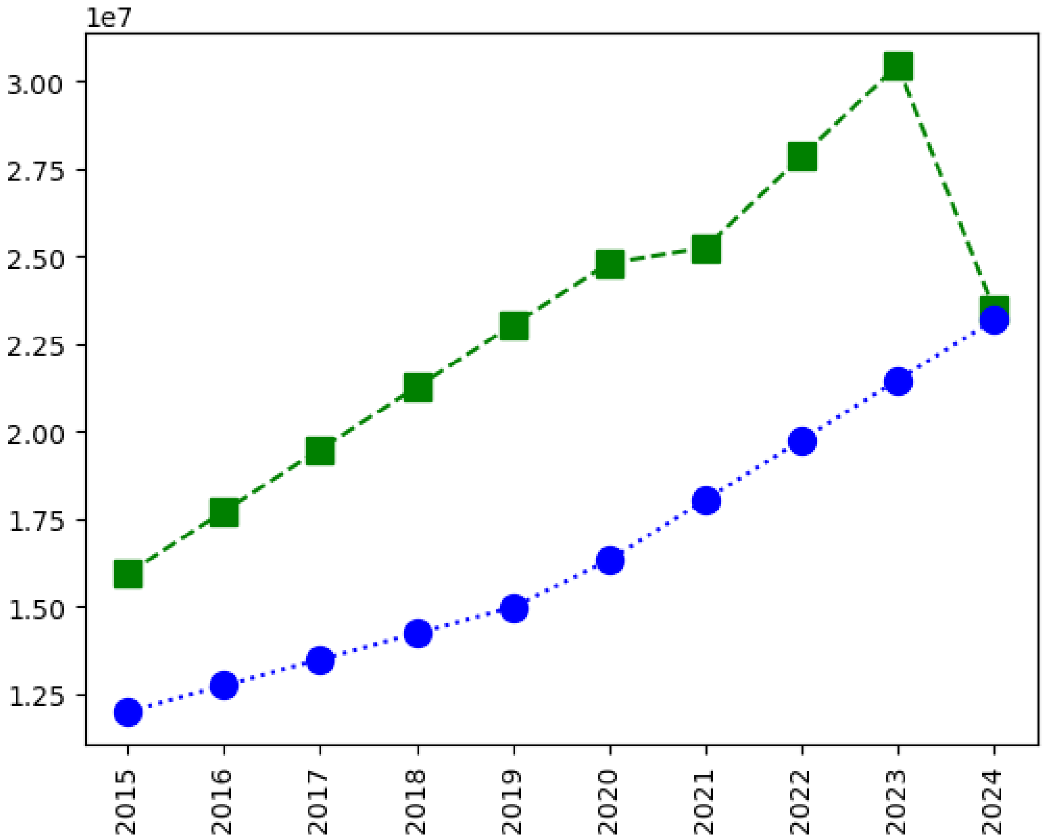
[2]:

```
import matplotlib.pyplot as plt
```

More visualization

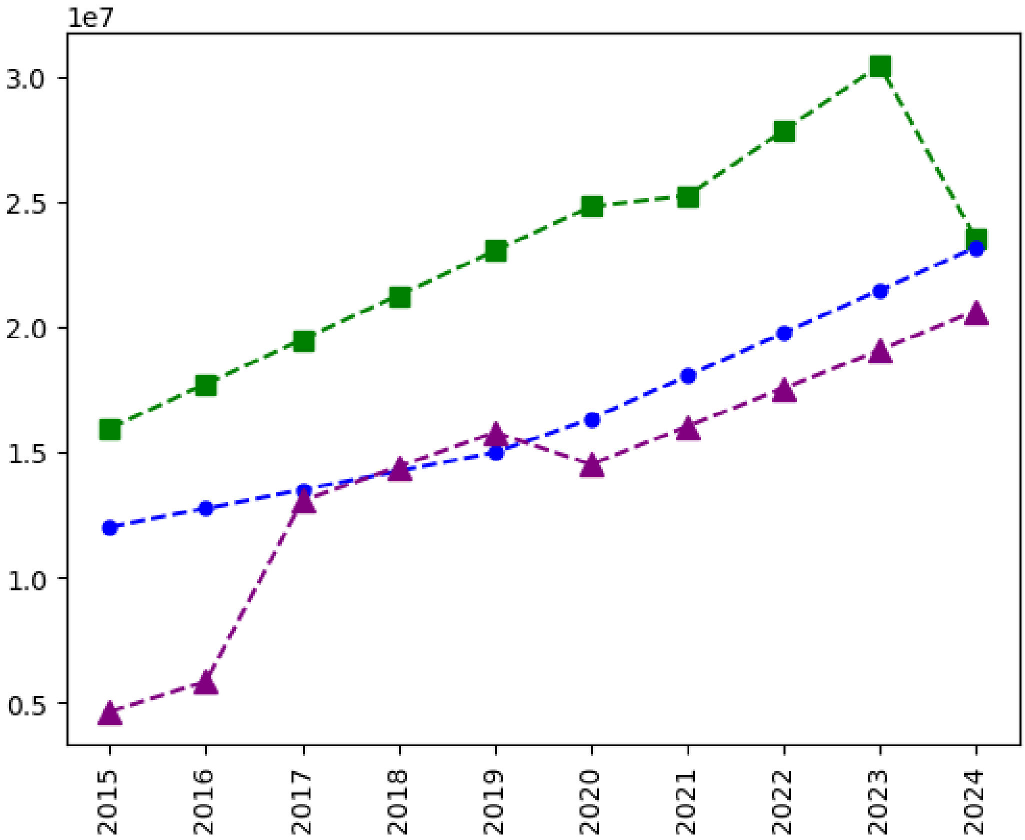
[3]:

```
plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 10, label =␣
 ↪Players[0])
plt.plot(Salary[1], c='Blue', ls = ':', marker = 'o', ms = 10, label =␣
 ↪Players[1])

plt.xticks(list(range(0,10)), Seasons,rotation='vertical')

plt.show()
```



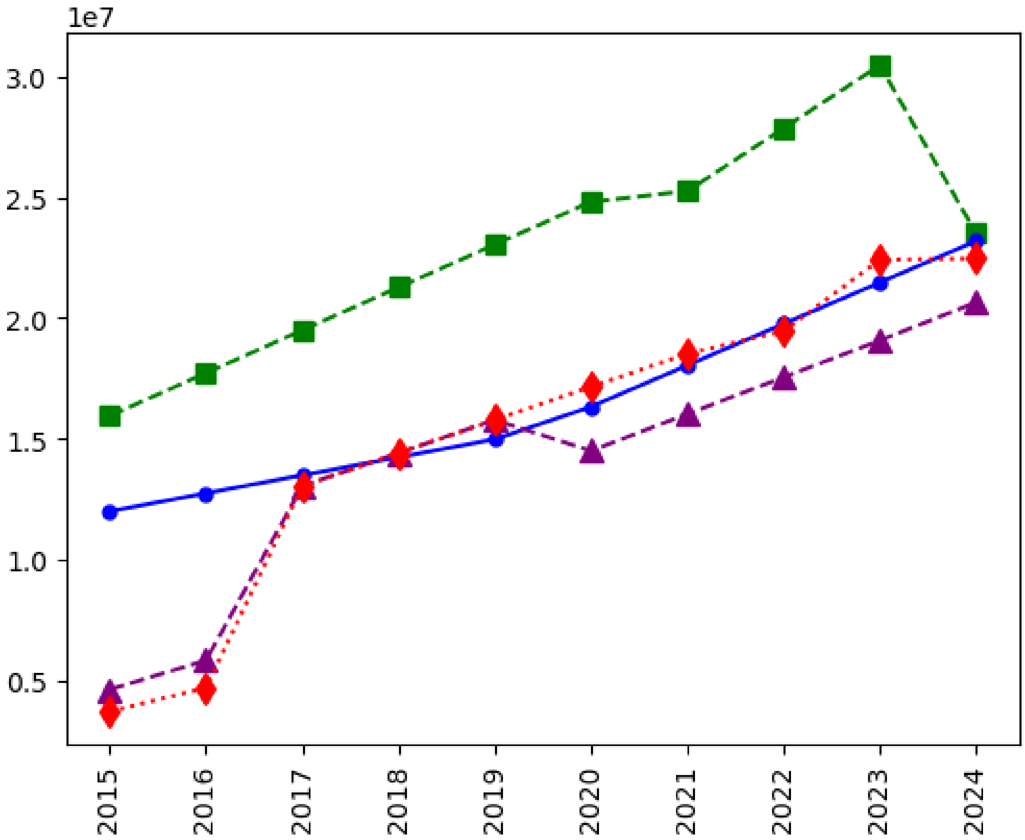
[4]:

```
plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label =␣
  ↪Players[0])
plt.plot(Salary[1], c='Blue', ls = '--', marker = 'o', ms = 5, label =␣
  ↪Players[1])
plt.plot(Salary[2], c='purple', ls = '--', marker = '^', ms = 8, label =␣
  ↪Players[2])


plt.xticks(list(range(0,10)), Seasons,rotation='vertical')

plt.show()
```

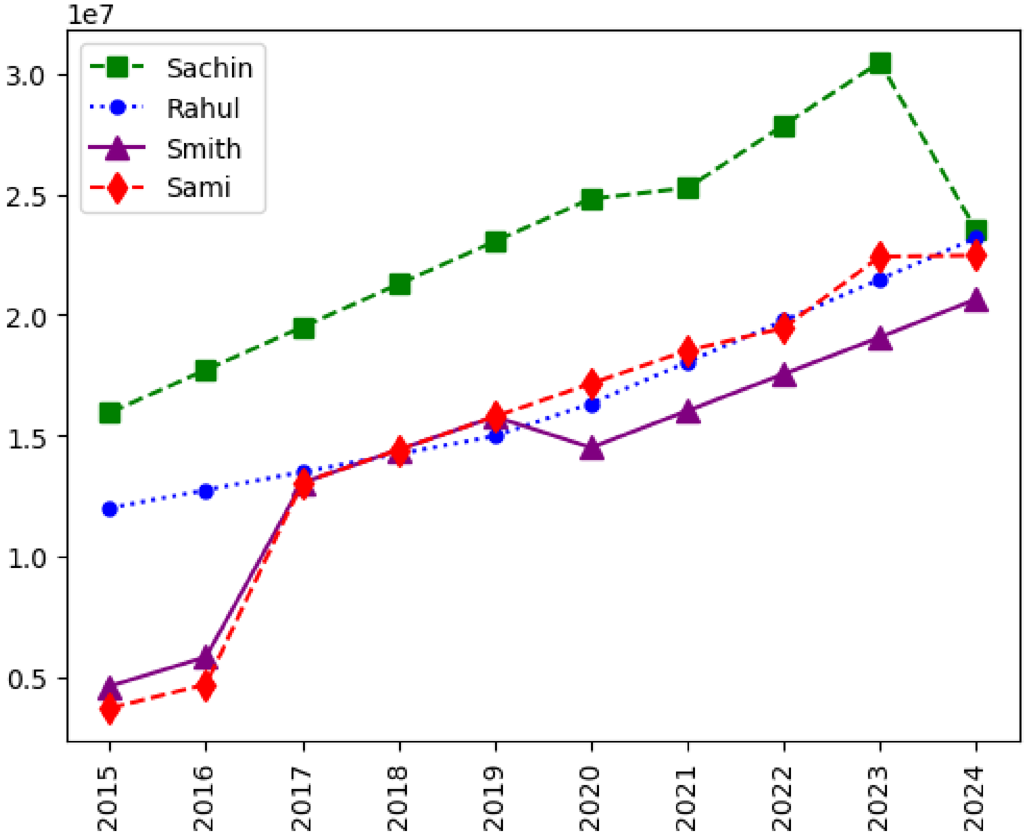


[5]:

```
plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label =␣
  ↪Players[0])
plt.plot(Salary[1], c='Blue', ls = '-', marker = 'o', ms = 5, label =␣
  ↪Players[1])
plt.plot(Salary[2], c='purple', ls = '--', marker = '^', ms = 8, label =␣
  ↪Players[2])
plt.plot(Salary[3], c='Red', ls = ':', marker = 'd', ms = 8, label = Players[3])

plt.xticks(list(range(0,10)), Seasons,rotation='vertical')

plt.show()
```

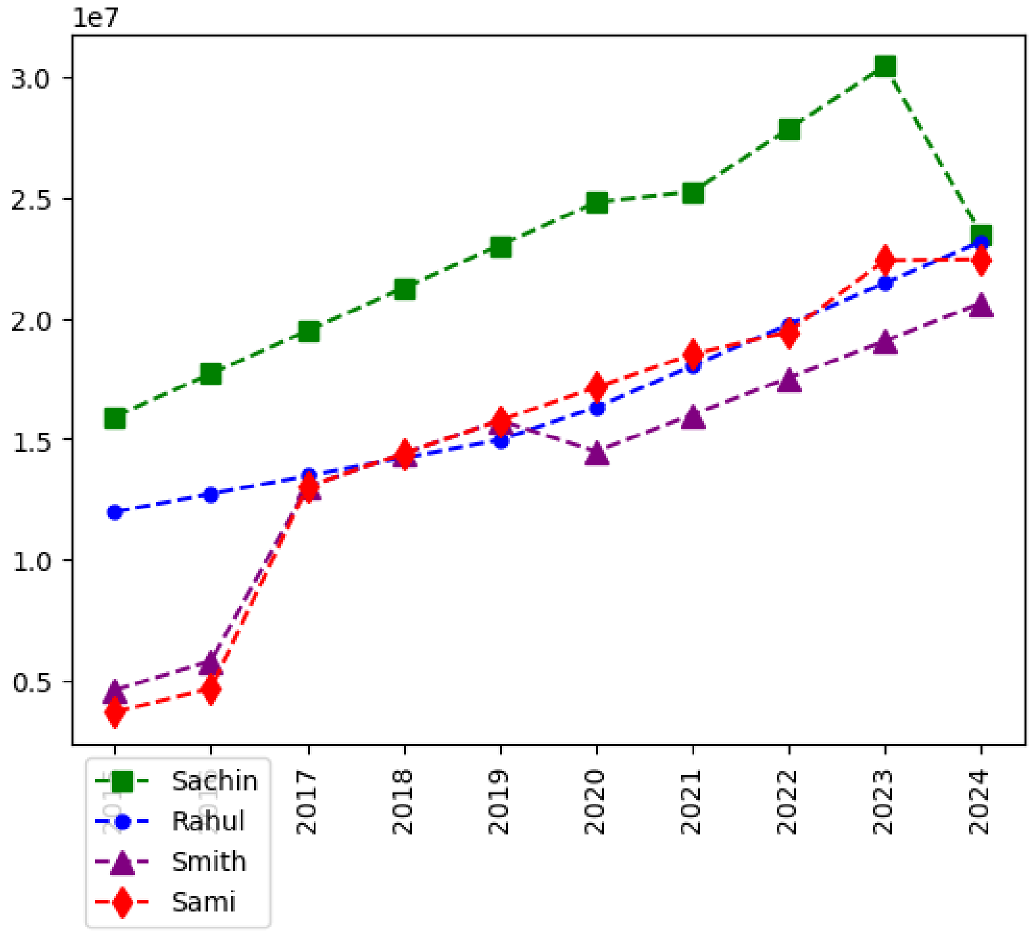


```
[6]: # how to add legned in visualisation

     plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label =␣
      ↪Players[0])
     plt.plot(Salary[1], c='Blue', ls = ':', marker = 'o', ms = 5, label =␣
      ↪Players[1])
     plt.plot(Salary[2], c='purple', ls = '-', marker = '^', ms = 8, label =␣
      ↪Players[2])
     plt.plot(Salary[3], c='Red', ls = '--', marker = 'd', ms = 8, label =␣
      ↪Players[3])
     plt.legend()
     plt.xticks(list(range(0,10)), Seasons,rotation='vertical')

     plt.show()
```

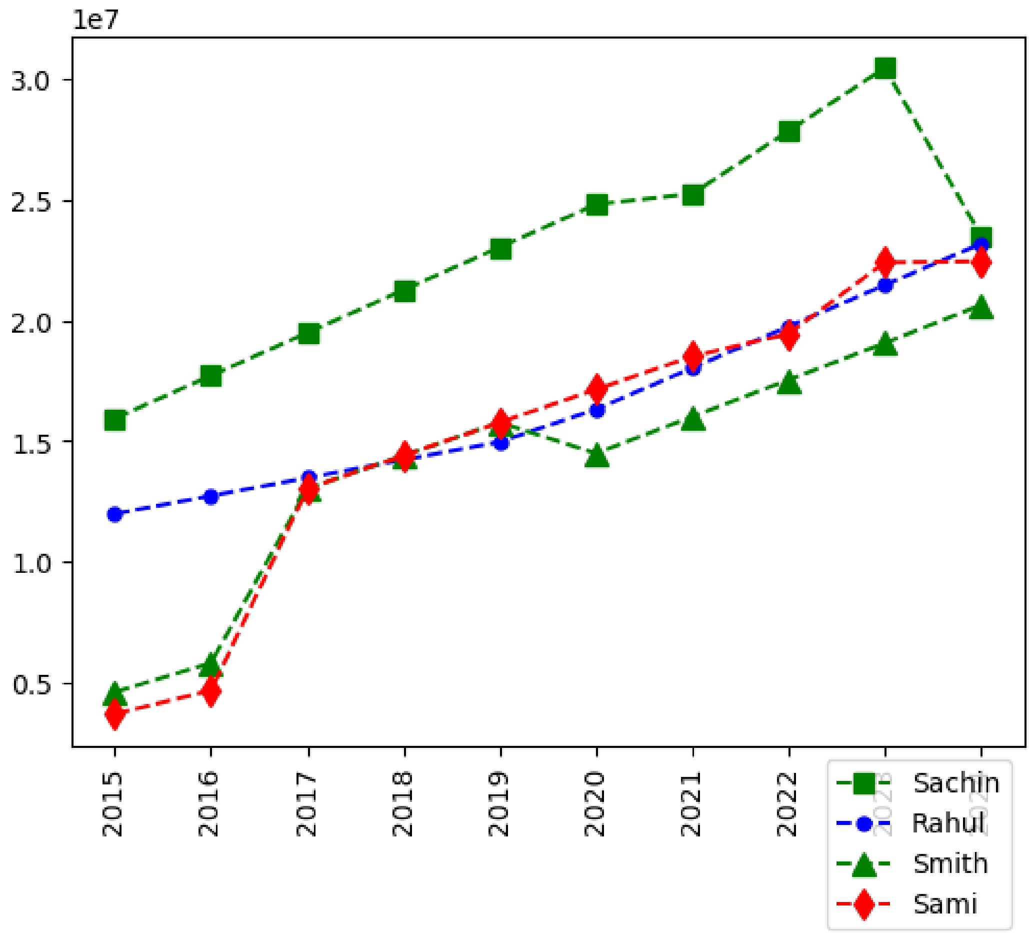


[7]:

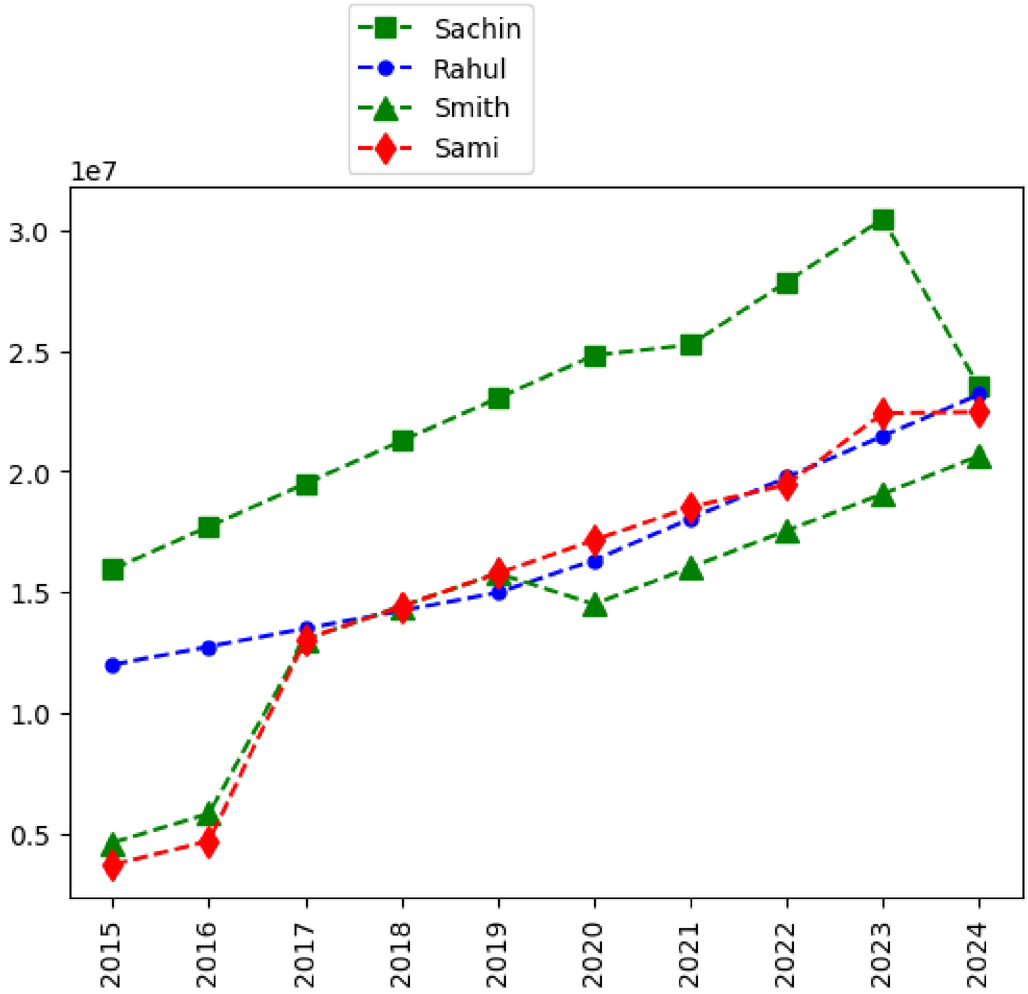
```
plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Players[0])
plt.plot(Salary[1], c='Blue', ls = '--', marker = 'o', ms = 5, label = Players[1])
plt.plot(Salary[2], c='purple', ls = '--', marker = '^', ms = 8, label = Players[2])
plt.plot(Salary[3], c='Red', ls = '--', marker = 'd', ms = 8, label = Players[3])
plt.legend(loc = 'upper left',bbox_to_anchor=(0,0) )
plt.xticks(list(range(0,10)), Seasons,rotation='vertical')

plt.show()
```



```
[8]: plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label =␣
       ↪Players[0])
     plt.plot(Salary[1], c='Blue', ls = '--', marker = 'o', ms = 5, label =␣
       ↪Players[1])
     plt.plot(Salary[2], c='Green', ls = '--', marker = '^', ms = 8, label =␣
       ↪Players[2])
     plt.plot(Salary[3], c='Red', ls = '--', marker = 'd', ms = 8, label =␣
       ↪Players[3])
     plt.legend(loc = 'upper right',bbox_to_anchor=(1,0) )
     plt.xticks(list(range(0,10)), Seasons,rotation='vertical')

     plt.show()
```

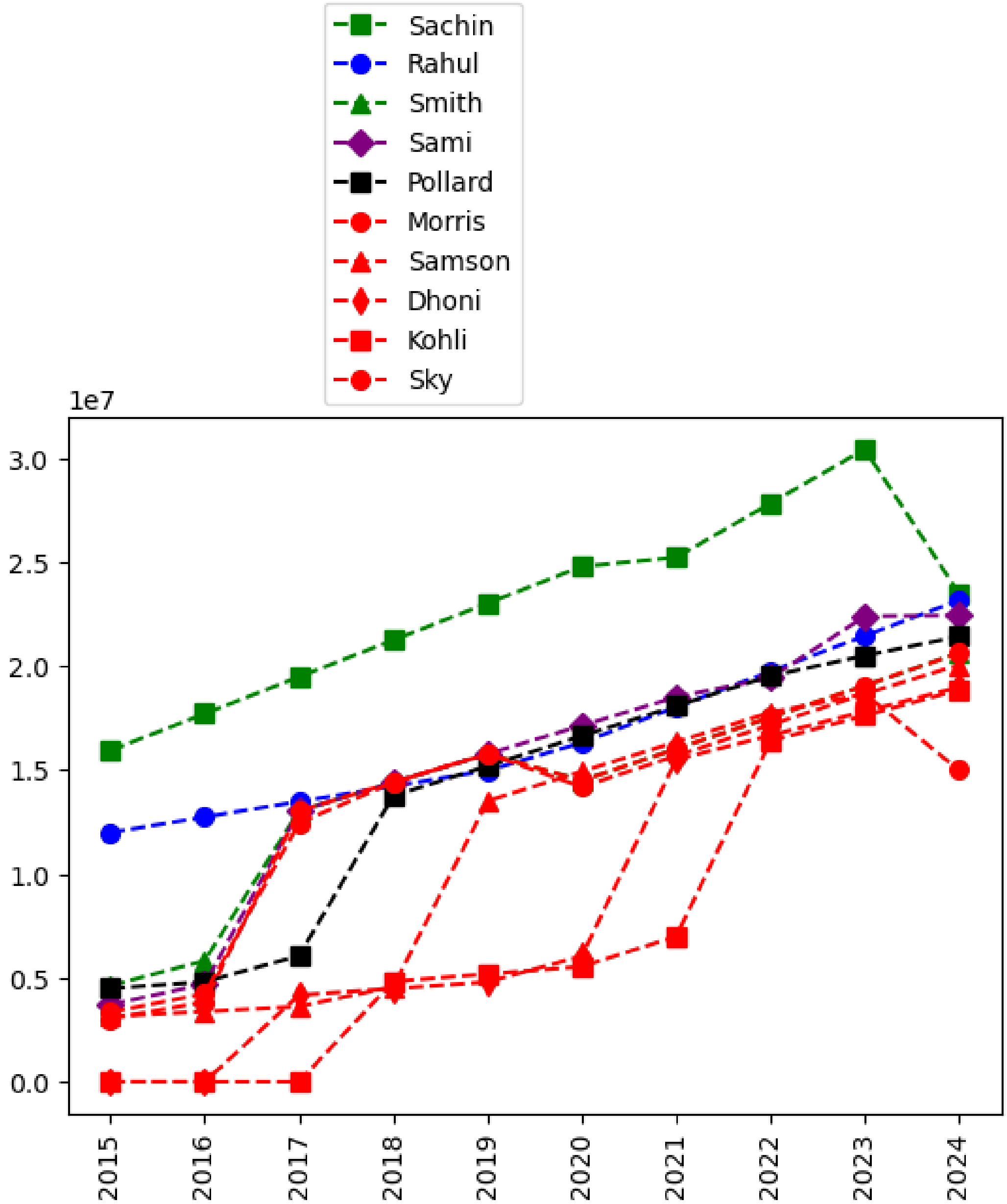


[9]:
```
plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label =␣
  ↪Players[0])
plt.plot(Salary[1], c='Blue', ls = '--', marker = 'o', ms = 5, label =␣
  ↪Players[1])
plt.plot(Salary[2], c='Green', ls = '--', marker = '^', ms = 8, label =␣
  ↪Players[2])
plt.plot(Salary[3], c='Red', ls = '--', marker = 'd', ms = 8, label =␣
  ↪Players[3])
plt.legend(loc = 'lower right',bbox_to_anchor=(0.5,1) )
plt.xticks(list(range(0,10)), Seasons,rotation='vertical')

plt.show()
```



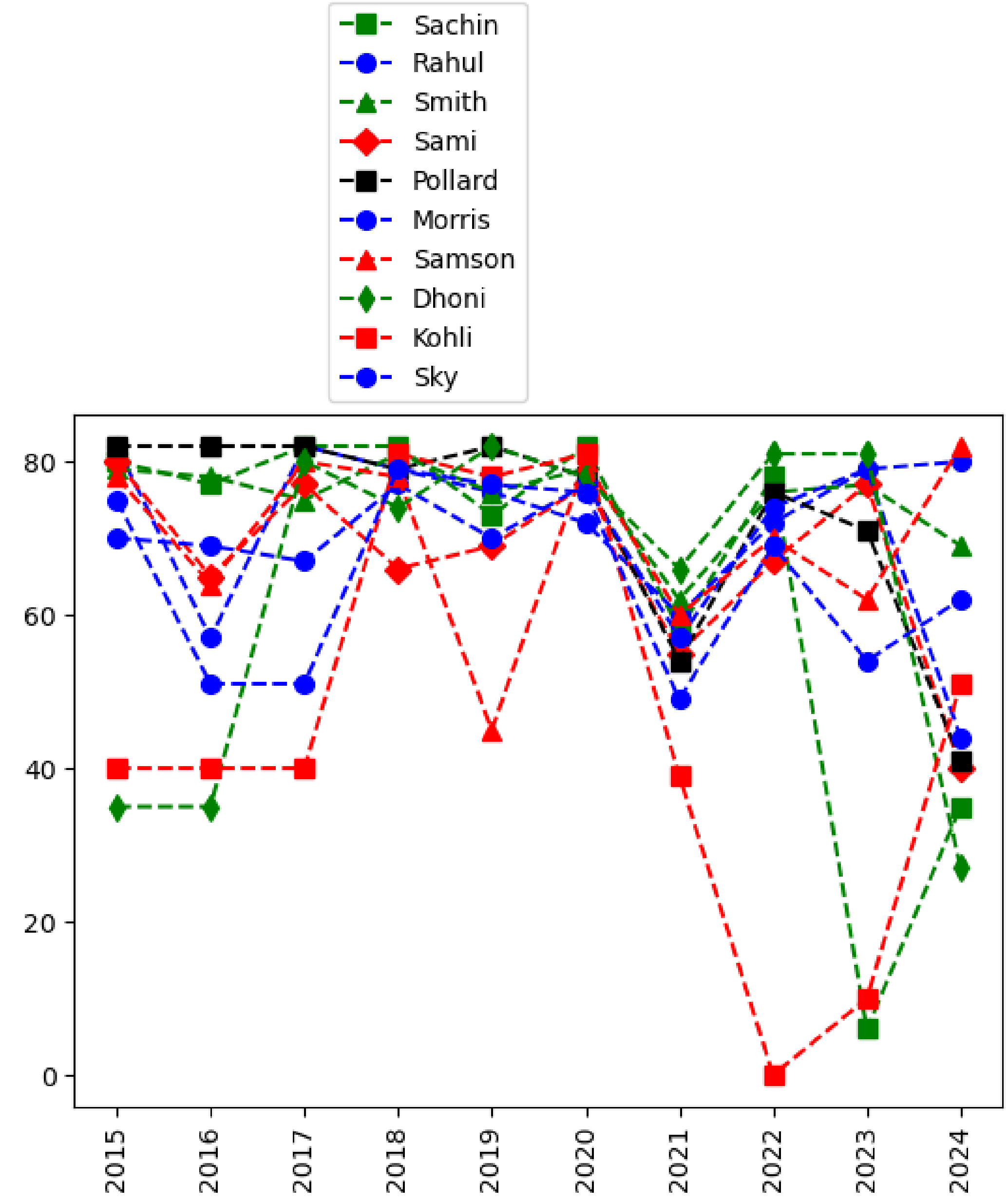
```
[12]: import warnings
      warnings.filterwarnings('ignore')
```

```
[14]: plt.plot(Salary[0], c='Green', ls='--', marker='s', ms=7, label=Players[0])
      plt.plot(Salary[1], c='Blue', ls='--', marker='o', ms=7, label=Players[1])
      plt.plot(Salary[2], c='Green', ls='--', marker='^', ms=7, label=Players[2])
      plt.plot(Salary[3], c='Purple', ls='--', marker='D', ms=7, label=Players[3])
      plt.plot(Salary[4], c='Black', ls='--', marker='s', ms=7, label=Players[4])
      plt.plot(Salary[5], c='Red', ls='--', marker='o', ms=7, label=Players[5])
      plt.plot(Salary[6], c='Red', ls='--', marker='^', ms=7, label=Players[6])
      plt.plot(Salary[7], c='Red', ls='--', marker='d', ms=7, label=Players[7])
      plt.plot(Salary[8], c='Red', ls='--', marker='s', ms=7, label=Players[8])
      plt.plot(Salary[9], c='Red', ls='--', marker='o', ms=7, label=Players[9])

      plt.legend(loc='lower right', bbox_to_anchor=(0.5, 1))
      plt.xticks(list(range(0, 10)), Seasons, rotation='vertical')
plt.show()
```



[11]:
```
# we can visualize the how many games played by a player

plt.plot(Games[0], c='Green', ls = '--', marker = 's', ms = 7, label =␣
  ↪Players[0])
plt.plot(Games[1], c='Blue', ls = '--', marker = 'o', ms = 7, label =␣
  ↪Players[1])
```

```
plt.plot(Games[2], c='Green', ls = '--', marker = '^', ms = 7, label = Players[2])
plt.plot(Games[3], c='Red', ls = '--', marker = 'D', ms = 7, label = Players[3])
plt.plot(Games[4], c='Black', ls = '--', marker = 's', ms = 7, label = Players[4])
plt.plot(Games[5], c='Blue', ls = '--', marker = 'o', ms = 7, label = Players[5])
plt.plot(Games[6], c='red', ls = '--', marker = '^', ms = 7, label = Players[6])
plt.plot(Games[7], c='Green', ls = '--', marker = 'd', ms = 7, label = Players[7])
plt.plot(Games[8], c='Red', ls = '--', marker = 's', ms = 7, label = Players[8])
plt.plot(Games[9], c='Blue', ls = '--', marker = 'o', ms = 7, label = Players[9])

plt.legend(loc = 'lower right',bbox_to_anchor=(0.5,1) )
plt.xticks(list(range(0,10)), Seasons,rotation='vertical')

plt.show()
```



This was a little introduction to Matplotlib using the IPL dataset. As part of my learning journey, I was asked to explore Matplotlib — and here, I've just covered one type of graph to get started.

More exmaple of advance visualizations (yet to learn)

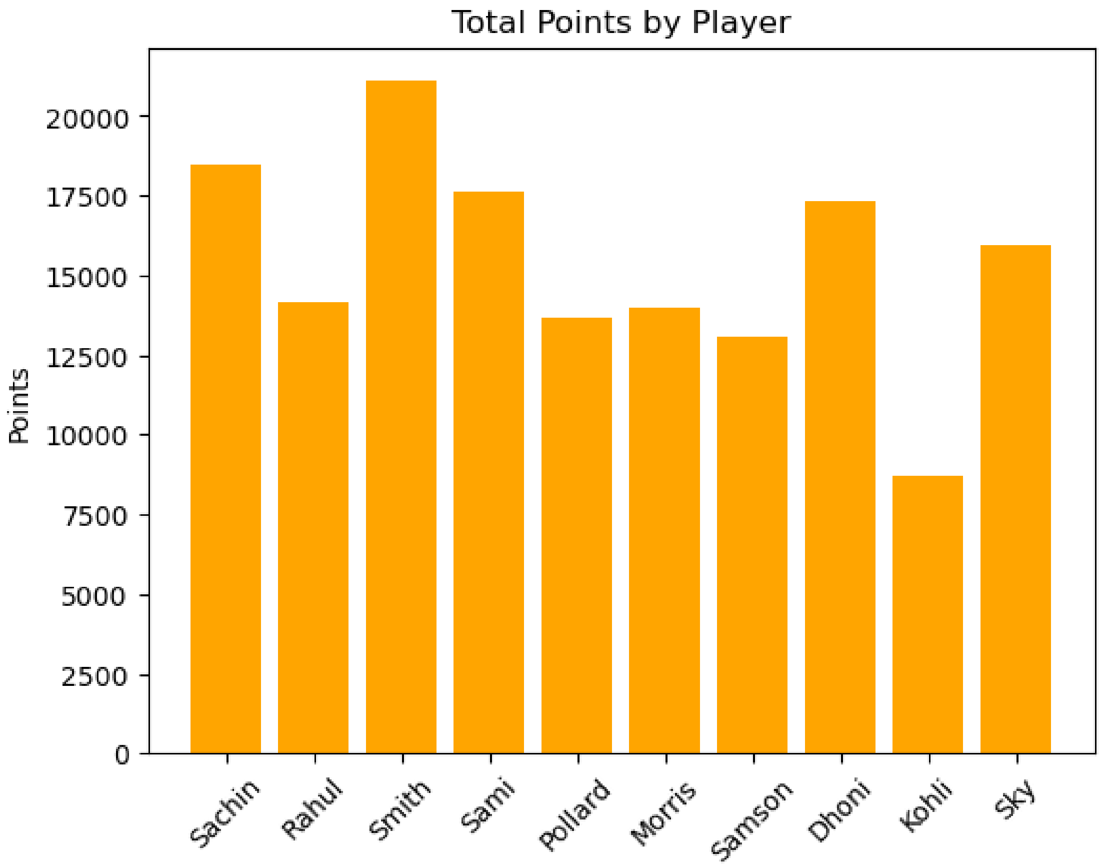
Bar Chart: Total Points by Player

What it shows: Cumulative points across all seasons.

Why it's useful: Quick comparison of overall contribution.

[15]:

```
total_points = Points.sum(axis=1)
plt.bar(Players, total_points, color='orange')
plt.title("Total Points by Player")
plt.ylabel("Points")
plt.xticks(rotation=45)
plt.show()
```



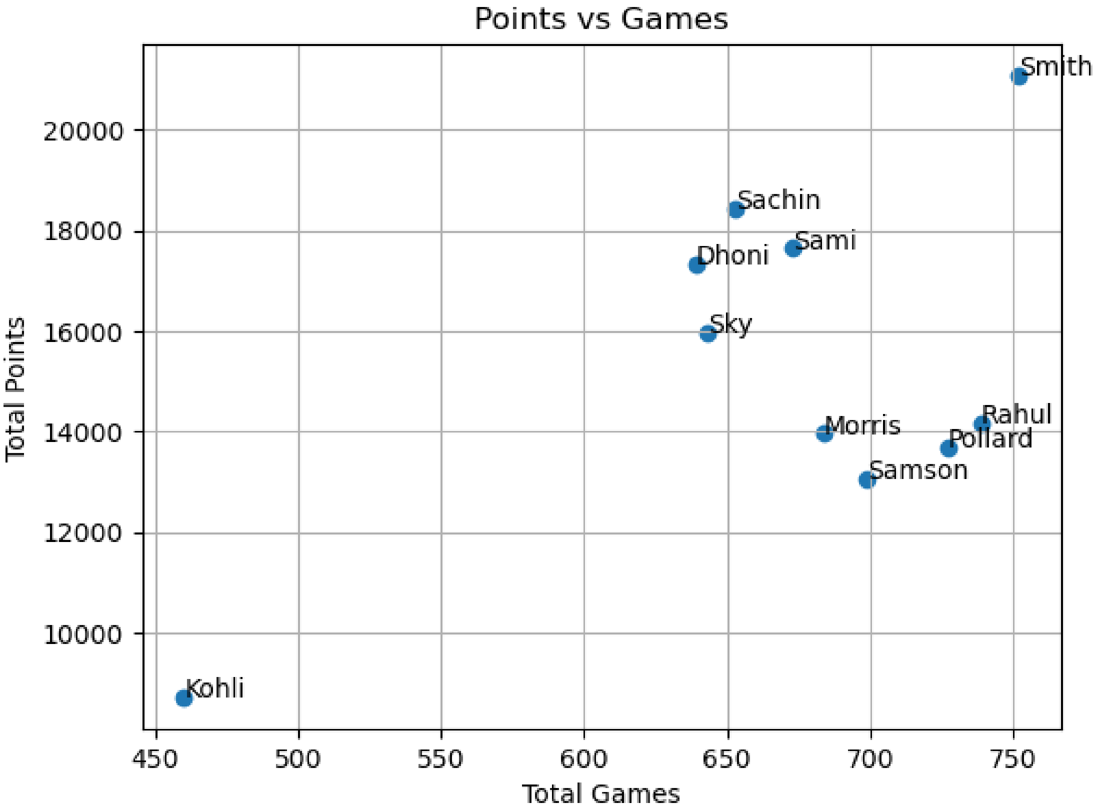
Scatter Plot: Points vs Games

What it shows: For all players, how their points relate to the number of games played.

Why it's useful: Reveals efficiency — e.g., high points in fewer games.

[16]:

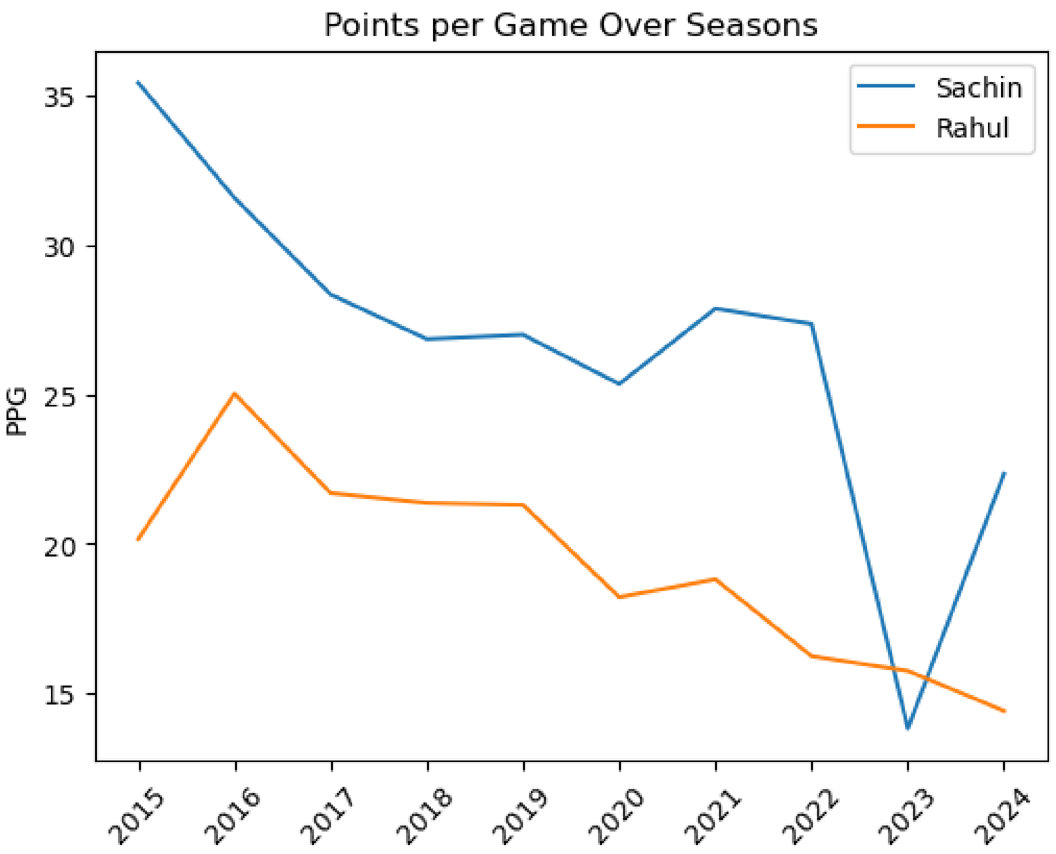
```
total_games = Games.sum(axis=1)
plt.scatter(total_games, total_points)
for i, name in enumerate(Players):
    plt.text(total_games[i], total_points[i], name)
plt.xlabel("Total Games")
plt.ylabel("Total Points")
plt.title("Points vs Games")
plt.grid(True)
plt.show()
```



Line Plot: Points Per Game (Efficiency) Over Seasons

What it shows: Seasonal efficiency instead of raw totals.

[17]:
```
ppg = Points / Games  # Avoid divide-by-zero if any games are 0
plt.plot(ppg[0], label=Players[0])
plt.plot(ppg[1], label=Players[1])
# Add more players if needed
plt.legend()
plt.title("Points per Game Over Seasons")
plt.xticks(list(range(10)), Seasons, rotation=45)
plt.ylabel("PPG")
plt.show()
```



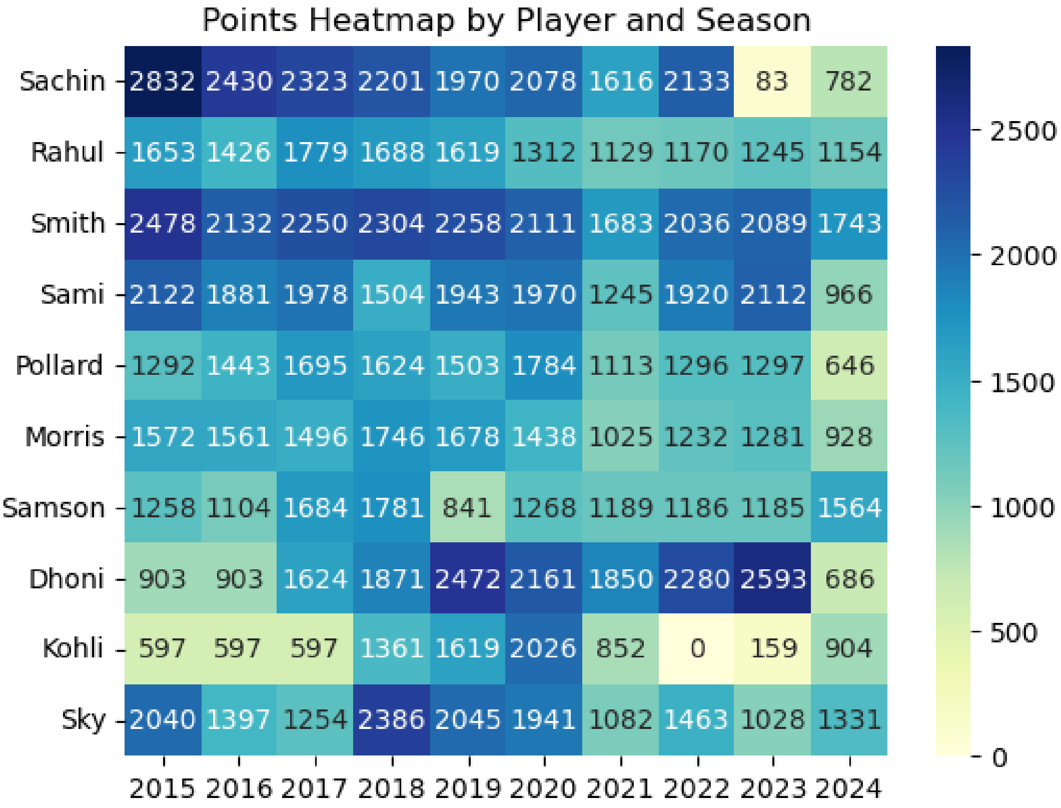
Heatmap: Player Salaries or Points Over Seasons

What it shows: A grid view of who earned/performed the most in which years.

Why it's useful: Instantly shows patterns.

[18]:

```
import seaborn as sns
import pandas as pd

df = pd.DataFrame(Points, index=Players, columns=Seasons)
sns.heatmap(df, annot=True, fmt='d', cmap='YlGnBu')
plt.title("Points Heatmap by Player and Season")
plt.show()
```



	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024
Sachin	2832	2430	2323	2201	1970	2078	1616	2133	83	782
Rahul	1653	1426	1779	1688	1619	1312	1129	1170	1245	1154
Smith	2478	2132	2250	2304	2258	2111	1683	2036	2089	1743
Sami	2122	1881	1978	1504	1943	1970	1245	1920	2112	966
Pollard	1292	1443	1695	1624	1503	1784	1113	1296	1297	646
Morris	1572	1561	1496	1746	1678	1438	1025	1232	1281	928
Samson	1258	1104	1684	1781	841	1268	1189	1186	1185	1564
Dhoni	903	903	1624	1871	2472	2161	1850	2280	2593	686
Kohli	597	597	597	1361	1619	2026	852	0	159	904
Sky	2040	1397	1254	2386	2045	1941	1082	1463	1028	1331

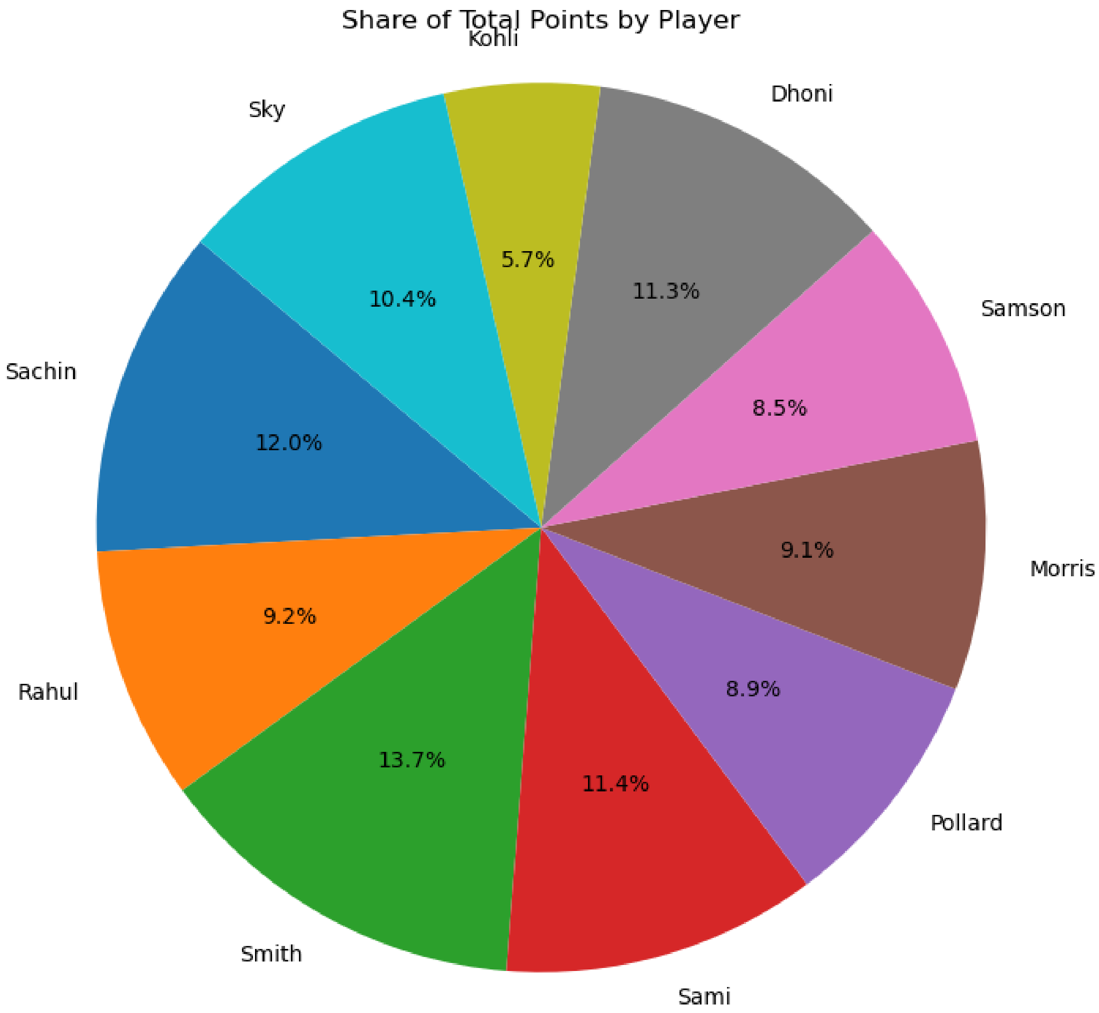
Pie Chart: Share of Total Salary or Points

What it shows: Proportion of total contribution or earning.

Why it's useful: Shows dominance or balance among players.

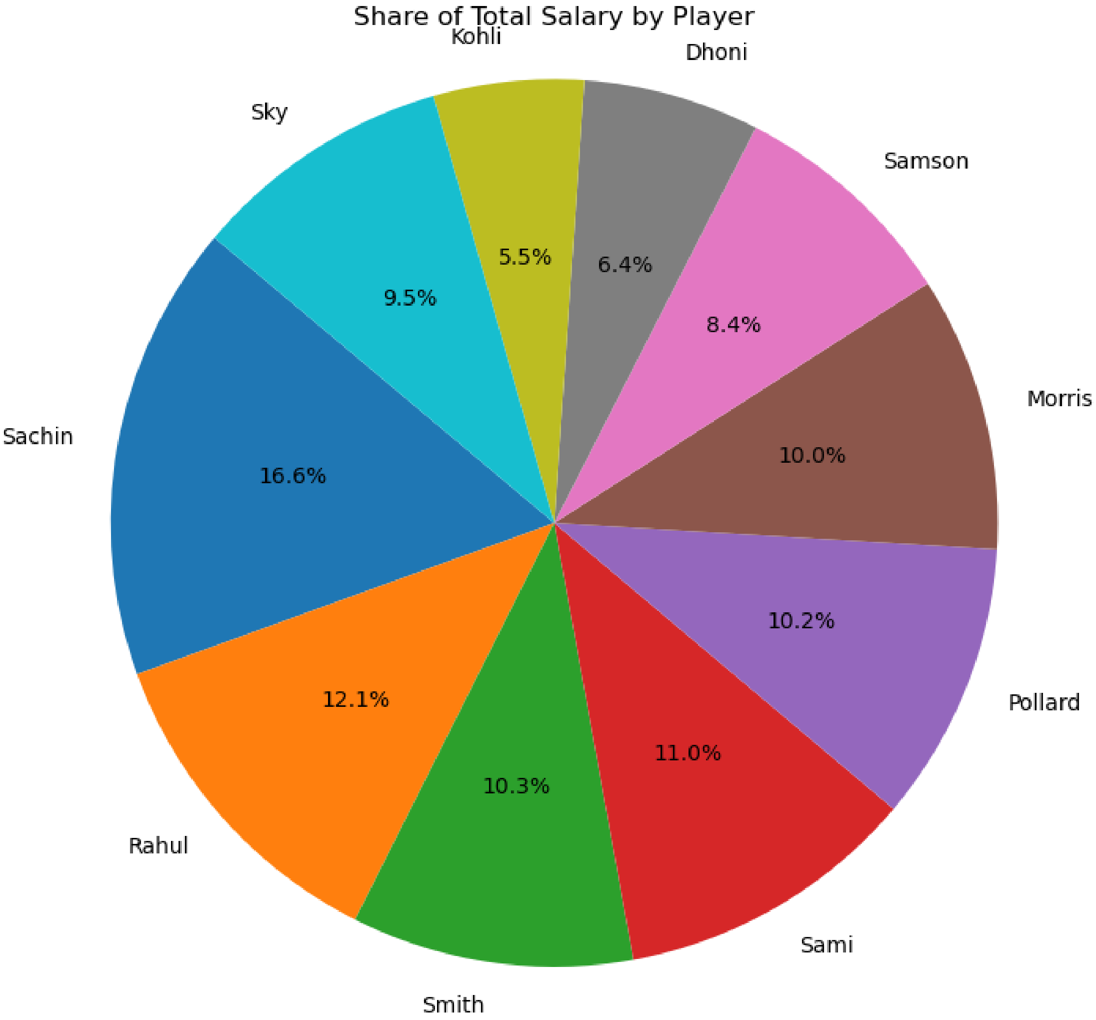
[19]:

```
import matplotlib.pyplot as plt

# Total Points
total_points = Points.sum(axis=1)

# Pie Chart
plt.figure(figsize=(8,8))
plt.pie(total_points, labels=Players, autopct='%1.1f%%', startangle=140)
plt.title("Share of Total Points by Player")
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()
```



Pie Chart: Share of Total Salary by Player

```
[20]: # Total Salary
total_salary = Salary.sum(axis=1)

# Pie Chart
plt.figure(figsize=(8,8))
plt.pie(total_salary, labels=Players, autopct='%1.1f%%', startangle=140,␣
  ↪colors=plt.cm.tab10.colors)
plt.title("Share of Total Salary by Player")
plt.axis('equal')
plt.show()
```


Share of Total Salary by Player
Kohli
5.5%
Dhoni
6.4%
Samson
8.4%
Morris
10.0%
Pollard
10.2%
Sami
11.0%
Smith
10.3%
Rahul
12.1%
Sachin
16.6%
Sky
9.5%