Day15_Matplotlib_using_IPL_Data_1

June 3, 2025

1 Day 15 – Introduction to Matplotlib & Data Visualization (IPL Dataset)

Today, I was introduced to **Matplotlib**, a powerful Python library used for **data visualization**. I began learning how to create simple graphs using the **IPL dataset** to uncover patterns and trends in cricket data.

1.1 What is Matplotlib?

Matplotlib is a plotting library for Python that helps you create: - Line plots - Bar charts - Histograms - Pie charts - Scatter plots - Heatmaps - and more!

1.1.1 Where is Matplotlib Used?

- Exploratory Data Analysis (EDA)
- Visualizing patterns and trends in data
- Creating dashboards and reports
- Academic and business presentations
- Working with datasets like IPL matches, stock prices, weather, etc.

1.1.2 Advantages of Matplotlib

- Easy to learn and use for beginners
- Highly customizable (colors, sizes, fonts, line styles, etc.)
- Integrates well with NumPy and Pandas
- Supports a wide range of plot types
- Perfect for Jupyter Notebooks and interactive environments
- Good for creating publication-quality visualizations

1.1.3 Disadvantages of Matplotlib

- Syntax can become complex for advanced plots
- Limited interactivity (compared to libraries like Plotly or Bokeh)

- Default styling is basic (can be improved with additional settings or Seaborn)
- Steeper learning curve when customizing subplots, multiple axes, legends, etc.

[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])
```

[ ]:
```
# Explore variables / data set
```

[2]:
```
Salary
```

[2]:
```
array([[15946875, 17718750, 19490625, 21262500, 23034375, 24806250,
        25244493, 27849149, 30453805, 23500000],
       [12000000, 12744189, 13488377, 14232567, 14976754, 16324500,
        18038573, 19752645, 21466718, 23180790],
       [ 4621800,  5828090, 13041250, 14410581, 15779912, 14500000,
        16022500, 17545000, 19067500, 20644400],
       [ 3713640,  4694041, 13041250, 14410581, 15779912, 17149243,
        18518574, 19450000, 22407474, 22458000],
       [ 4493160,  4806720,  6061274, 13758000, 15202590, 16647180,
        18091770, 19536360, 20513178, 21436271],
       [ 3348000,  4235220, 12455000, 14410581, 15779912, 14500000,
        16022500, 17545000, 19067500, 20644400],
       [ 3144240,  3380160,  3615960,  4574189, 13520500, 14940153,
        16359805, 17779458, 18668431, 20068563],
       [       0,        0,  4171200,  4484040,  4796880,  6053663,
```

```
       15506632, 16669630, 17832627, 18995624],
      [       0,        0,        0,  4822800,  5184480,  5546160,
        6993708, 16402500, 17632688, 18862875],
      [ 3031920,  3841443, 13041250, 14410581, 15779912, 14200000,
       15691000, 17182000, 18673000, 15000000]])
```

[3]:
```
Games
```

[3]:
```
array([[80, 77, 82, 82, 73, 82, 58, 78,  6, 35],
       [82, 57, 82, 79, 76, 72, 60, 72, 79, 80],
       [79, 78, 75, 81, 76, 79, 62, 76, 77, 69],
       [80, 65, 77, 66, 69, 77, 55, 67, 77, 40],
       [82, 82, 82, 79, 82, 78, 54, 76, 71, 41],
       [70, 69, 67, 77, 70, 77, 57, 74, 79, 44],
       [78, 64, 80, 78, 45, 80, 60, 70, 62, 82],
       [35, 35, 80, 74, 82, 78, 66, 81, 81, 27],
       [40, 40, 40, 81, 78, 81, 39,  0, 10, 51],
       [75, 51, 51, 79, 77, 76, 49, 69, 54, 62]])
```

[4]:
```
Points
```

[4]:
```
array([[2832, 2430, 2323, 2201, 1970, 2078, 1616, 2133,   83,  782],
       [1653, 1426, 1779, 1688, 1619, 1312, 1129, 1170, 1245, 1154],
       [2478, 2132, 2250, 2304, 2258, 2111, 1683, 2036, 2089, 1743],
       [2122, 1881, 1978, 1504, 1943, 1970, 1245, 1920, 2112,  966],
       [1292, 1443, 1695, 1624, 1503, 1784, 1113, 1296, 1297,  646],
       [1572, 1561, 1496, 1746, 1678, 1438, 1025, 1232, 1281,  928],
       [1258, 1104, 1684, 1781,  841, 1268, 1189, 1186, 1185, 1564],
       [ 903,  903, 1624, 1871, 2472, 2161, 1850, 2280, 2593,  686],
       [ 597,  597,  597, 1361, 1619, 2026,  852,    0,  159,  904],
       [2040, 1397, 1254, 2386, 2045, 1941, 1082, 1463, 1028, 1331]])
```

[5]:
```
Salary[0] # Featch 1st players salary to see
```

[5]:
```
array([15946875, 17718750, 19490625, 21262500, 23034375, 24806250,
       25244493, 27849149, 30453805, 23500000])
```

[6]:
```
Games
```

[6]:
```
array([[80, 77, 82, 82, 73, 82, 58, 78,  6, 35],
       [82, 57, 82, 79, 76, 72, 60, 72, 79, 80],
       [79, 78, 75, 81, 76, 79, 62, 76, 77, 69],
       [80, 65, 77, 66, 69, 77, 55, 67, 77, 40],
       [82, 82, 82, 79, 82, 78, 54, 76, 71, 41],
       [70, 69, 67, 77, 70, 77, 57, 74, 79, 44],
       [78, 64, 80, 78, 45, 80, 60, 70, 62, 82],
       [35, 35, 80, 74, 82, 78, 66, 81, 81, 27],
       [40, 40, 40, 81, 78, 81, 39,  0, 10, 51],
```

```
        [75, 51, 51, 79, 77, 76, 49, 69, 54, 62]])
```

[7]:
```
Games[1:5]
```

[7]:
```
array([[82, 57, 82, 79, 76, 72, 60, 72, 79, 80],
       [79, 78, 75, 81, 76, 79, 62, 76, 77, 69],
       [80, 65, 77, 66, 69, 77, 55, 67, 77, 40],
       [82, 82, 82, 79, 82, 78, 54, 76, 71, 41]])
```

[8]:
```
Games[1,5]
```

[8]:
```
72
```

[9]:
```
Pdict
```

[9]:
```
{'Sachin': 0,
 'Rahul': 1,
 'Smith': 2,
 'Sami': 3,
 'Pollard': 4,
 'Morris': 5,
 'Samson': 6,
 'Dhoni': 7,
 'Kohli': 8,
 'Sky': 9}
```

[10]:
```
Salary/Games
```

```
C:\Users\aksha\AppData\Local\Temp\ipykernel_17360\3709746658.py:1:
RuntimeWarning: divide by zero encountered in divide
  Salary/Games
```

[10]:
```
array([[ 199335.9375    ,  230113.63636364,  237690.54878049,
         259298.7804878 ,  315539.38356164,  302515.24390244,
         435249.87931034,  357040.37179487, 5075634.16666667,
         671428.57142857],
       [ 146341.46341463,  223582.26315789,  164492.40243902,
         180159.07594937,  197062.55263158,  226729.16666667,
         300642.88333333,  274342.29166667,  271730.60759494,
         289759.875     ],
       [  58503.79746835,   74719.1025641 ,  173883.33333333,
         177908.40740741,  207630.42105263,  183544.30379747,
         258427.41935484,  230855.26315789,  247629.87012987,
         299194.20289855],
       [  46420.5       ,   72216.01538462,  169366.88311688,
         218342.13636364,  228694.37681159,  222717.44155844,
         336701.34545455,  290298.50746269,  291006.15584416,
         561450.        ],
```

```
       [  54794.63414634,   58618.53658537,   73917.97560976,
         174151.89873418,  185397.43902439,  213425.38461538,
         335032.77777778,  257057.36842105,  288918.        ,
         522835.87804878],
       [  47828.57142857,   61380.        ,  185895.52238806,
         187150.4025974 ,  225427.31428571,  188311.68831169,
         281096.49122807,  237094.59459459,  241360.75949367,
         469190.90909091],
       [  40310.76923077,   52815.        ,   45199.5       ,
          58643.44871795,  300455.55555556,  186751.9125    ,
         272663.41666667,  253992.25714286,  301103.72580645,
         244738.57317073],
       [      0.        ,       0.        ,   52140.        ,
          60595.13513514,   58498.53658537,   77611.06410256,
         234948.96969697,  205797.90123457,  220155.88888889,
         703541.62962963],
       [      0.        ,       0.        ,       0.        ,
          59540.74074074,   66467.69230769,   68471.11111111,
         179325.84615385,             inf, 1763268.8       ,
         369860.29411765],
       [  40425.6       ,   75322.41176471,  255710.78431373,
         182412.41772152,  204933.92207792,  186842.10526316,
         320224.48979592,  249014.49275362,  345796.2962963 ,
         241935.48387097]])
```

[11]:
```
np.round(Salary//Games)
```

```
C:\Users\aksha\AppData\Local\Temp\ipykernel_17360\3663165759.py:1:
RuntimeWarning: divide by zero encountered in floor_divide
  np.round(Salary//Games)
```

[11]:
```
array([[ 199335,  230113,  237690,  259298,  315539,  302515,  435249,
          357040, 5075634,  671428],
       [ 146341,  223582,  164492,  180159,  197062,  226729,  300642,
         274342,  271730,  289759],
       [  58503,   74719,  173883,  177908,  207630,  183544,  258427,
         230855,  247629,  299194],
       [  46420,   72216,  169366,  218342,  228694,  222717,  336701,
         290298,  291006,  561450],
       [  54794,   58618,   73917,  174151,  185397,  213425,  335032,
         257057,  288918,  522835],
       [  47828,   61380,  185895,  187150,  225427,  188311,  281096,
         237094,  241360,  469190],
       [  40310,   52815,   45199,   58643,  300455,  186751,  272663,
         253992,  301103,  244738],
       [      0,       0,   52140,   60595,   58498,   77611,  234948,
         205797,  220155,  703541],
       [      0,       0,       0,   59540,   66467,   68471,  179325,
```

```
       0, 1763268,  369860],
[  40425,   75322,  255710,  182412,  204933,  186842,  320224,
  249014,  345796,  241935]])
```

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

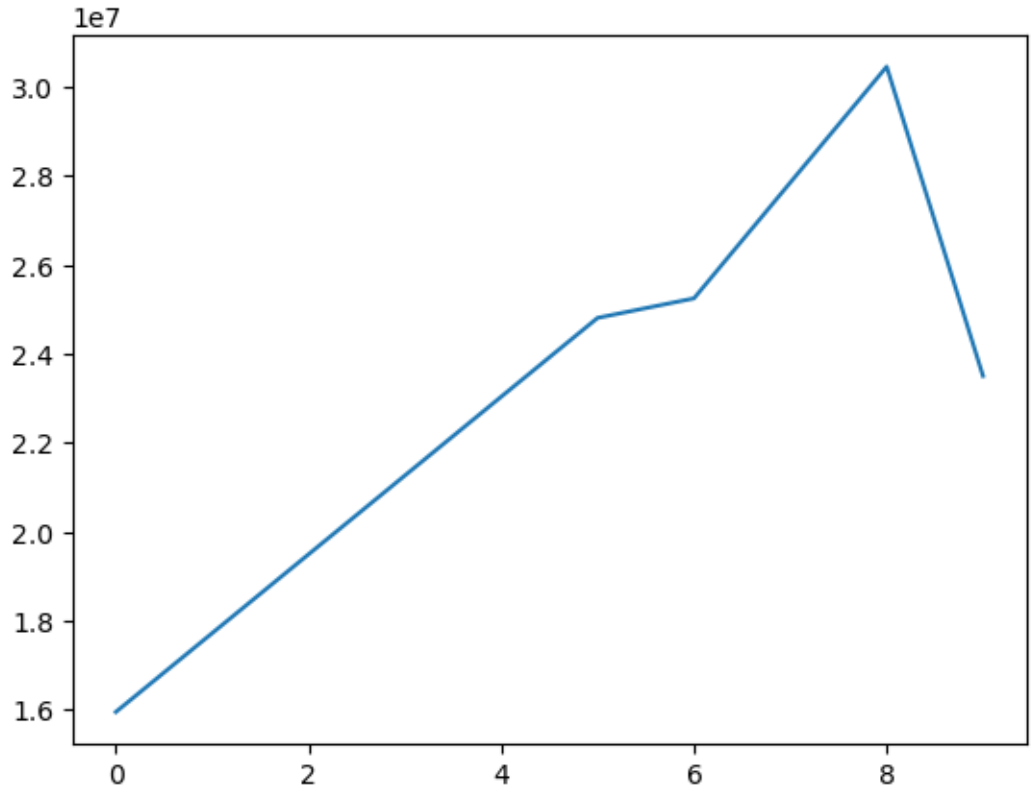
[13]:
```
import matplotlib.pyplot as plt
```

[14]:
```
Salary[0]
```

[14]:
```
array([15946875, 17718750, 19490625, 21262500, 23034375, 24806250,
       25244493, 27849149, 30453805, 23500000])
```

[15]:
```
# Plot only variable
plt.plot(Salary[0])
```

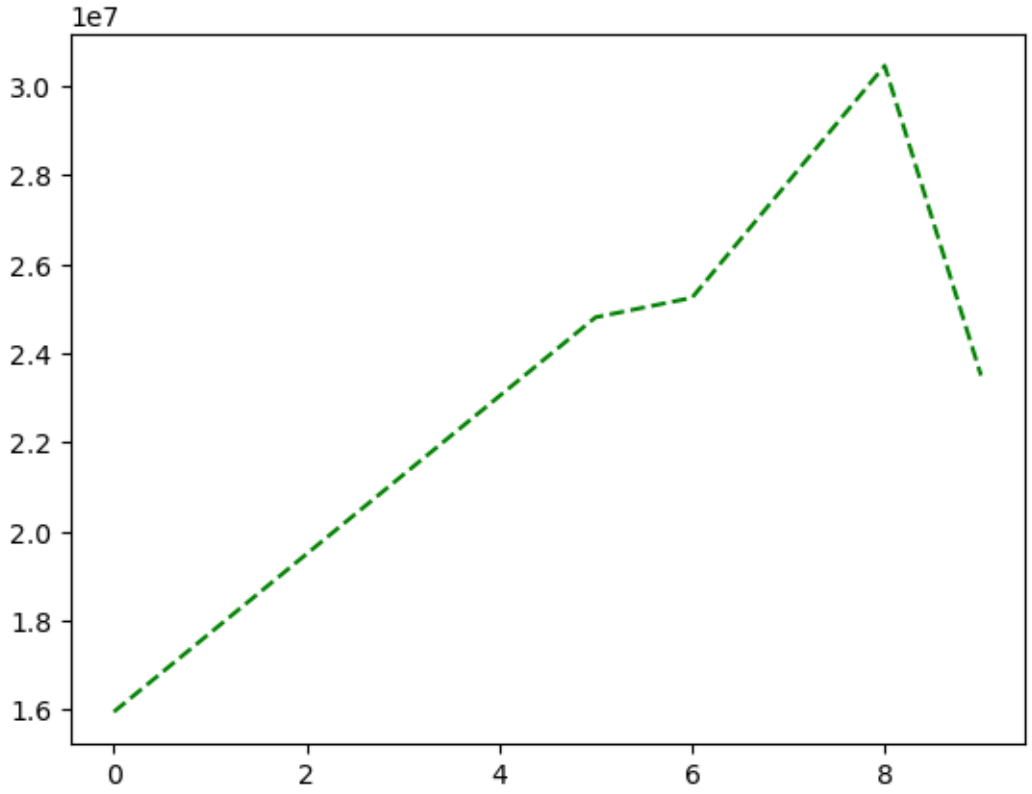
[15]:
```
[<matplotlib.lines.Line2D at 0x1a893c290a0>]
```



Color & line Style

[16]:
```
# Change color and line style
plt.plot(Salary[0], color = 'green',ls='--')
```

[16]:
```
[<matplotlib.lines.Line2D at 0x1a893cca2a0>]
```



Matplotlib Colors & Line Styles (Reference)

Colors

The supported color abbreviations in Matplotlib are:

Character	Color
`'b'`	blue
`'g'`	green
`'r'`	red
`'c'`	cyan
`'m'`	magenta
`'y'`	yellow
`'k'`	black
`'w'`	white

You can also use `'C0'`, `'C1'`, ..., `'C9'` to get colors from the **default property cycle** used by

Matplotlib.

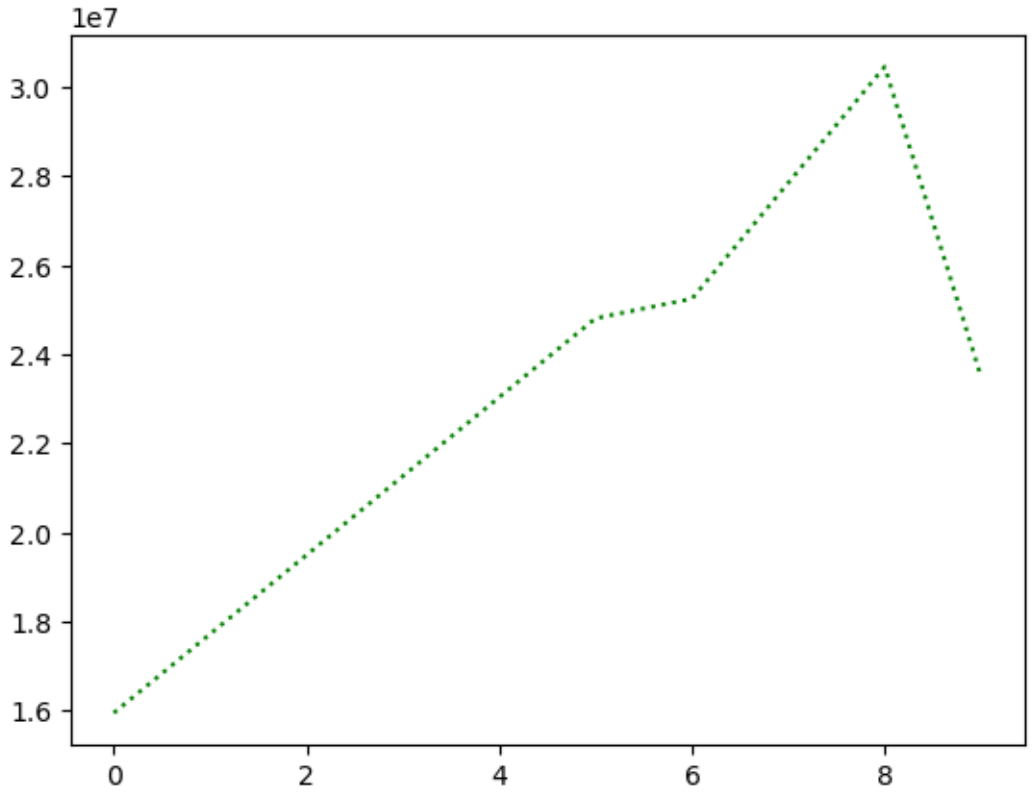
Line Styles

Here are the line style options supported in Matplotlib:

Character	Description
'-'	solid line
'--'	dashed line
'-.'	dash-dot line
':'	dotted line

[17]:
```
plt.plot(Salary[0], c = 'g',ls=':') # Line style dotted line
```

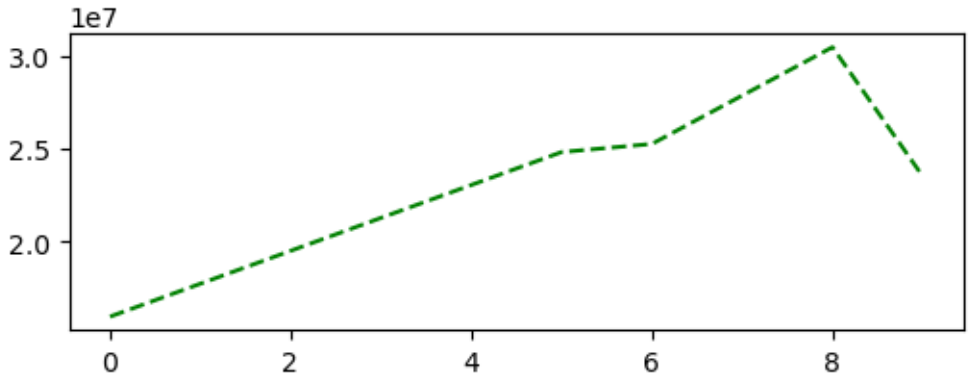
[17]:
```
[<matplotlib.lines.Line2D at 0x1a895544470>]
```



Plot Figure Functions

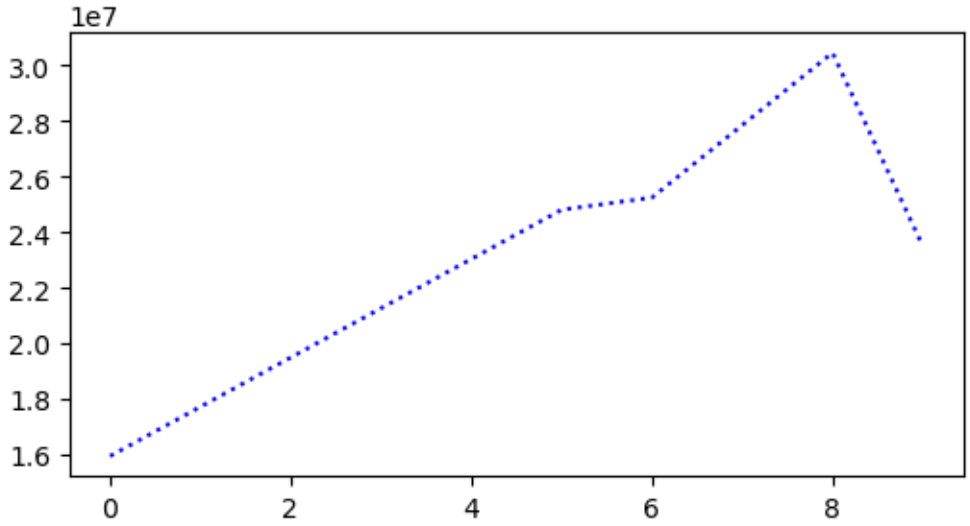
[18]:
```
%matplotlib inline
plt.rcParams['figure.figsize'] = 6,2
```

```
# or plt.figure(figsize=(width, height))
plt.plot(Salary[0], c = 'g',ls='--')
plt.show()
```



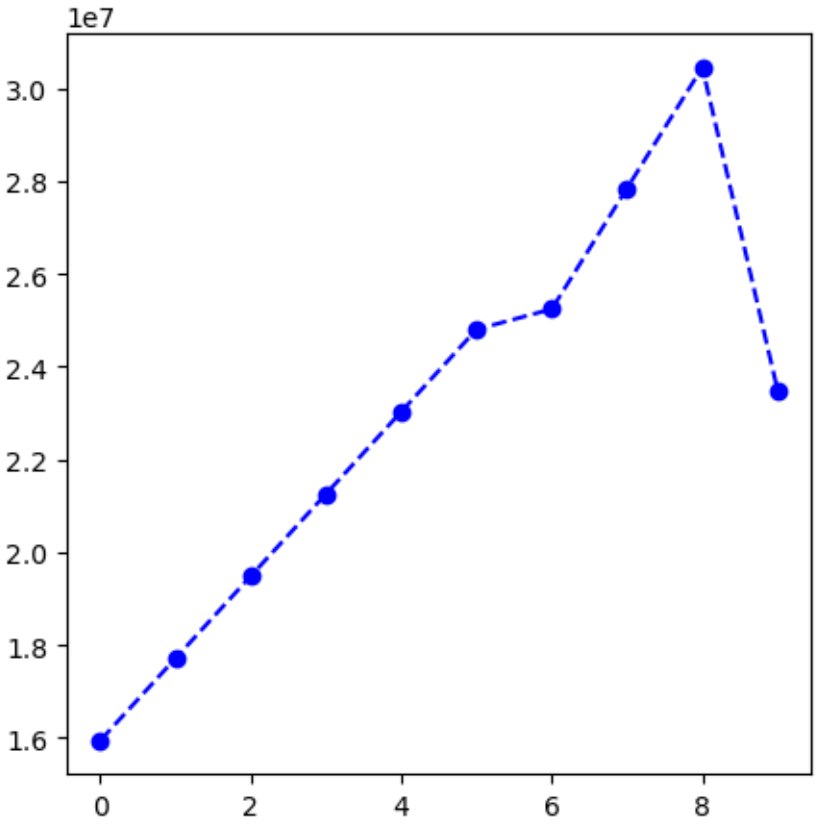
Our Use This Figure Function

[20]:
```
plt.figure(figsize=(6,3))
plt.plot(Salary[0], c = 'b',ls=':')
plt.show()
```



Marker

[21]:
```
plt.figure(figsize=(5,5))
plt.plot(Salary[0], c = 'b',ls='--', marker = 'o')
plt.show()
```



Matplotlib Marker Styles Reference

Use these markers in your plots to style the points in `plt.plot(x, y, marker='...')`.

Character	Marker Type
`'.'`	point marker
`','`	pixel marker
`'o'`	circle marker
`'v'`	triangle_down marker
`'^'`	triangle_up marker
`'<'`	triangle_left marker
`'>'`	triangle_right marker
`'1'`	tri_down marker
`'2'`	tri_up marker
`'3'`	tri_left marker
`'4'`	tri_right marker
`'8'`	octagon marker
`'s'`	square marker
`'p'`	pentagon marker
`'P'`	plus (filled) marker
`'*'`	star marker

Character	Marker Type
'h'	hexagon1 marker
'H'	hexagon2 marker
'+'	plus marker
'x'	x marker
'X'	x (filled) marker
'D'	diamond marker
'd'	thin_diamond marker
'|'	vertical line marker
'_'	horizontal line marker

Example Usage: “‘python

plt.plot(x, y, marker=‘o’) # Circle marker

plt.plot(x, y, marker=’*’) # Star marker

[22]:
```
Sdict
```

[22]:
```
{'2015': 0,
 '2016': 1,
 '2017': 2,
 '2018': 3,
 '2019': 4,
 '2020': 5,
 '2021': 6,
 '2022': 7,
 '2023': 8,
 '2024': 9}
```

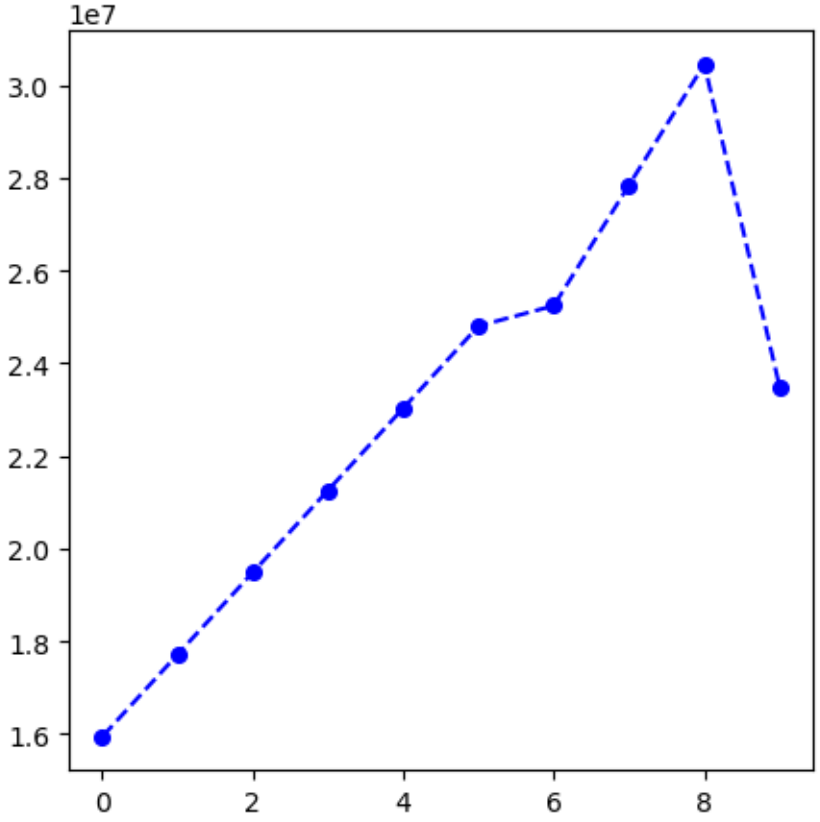
[23]:
```
Pdict
```

[23]:
```
{'Sachin': 0,
 'Rahul': 1,
 'Smith': 2,
 'Sami': 3,
 'Pollard': 4,
 'Morris': 5,
 'Samson': 6,
 'Dhoni': 7,
 'Kohli': 8,
 'Sky': 9}
```

Marker Size

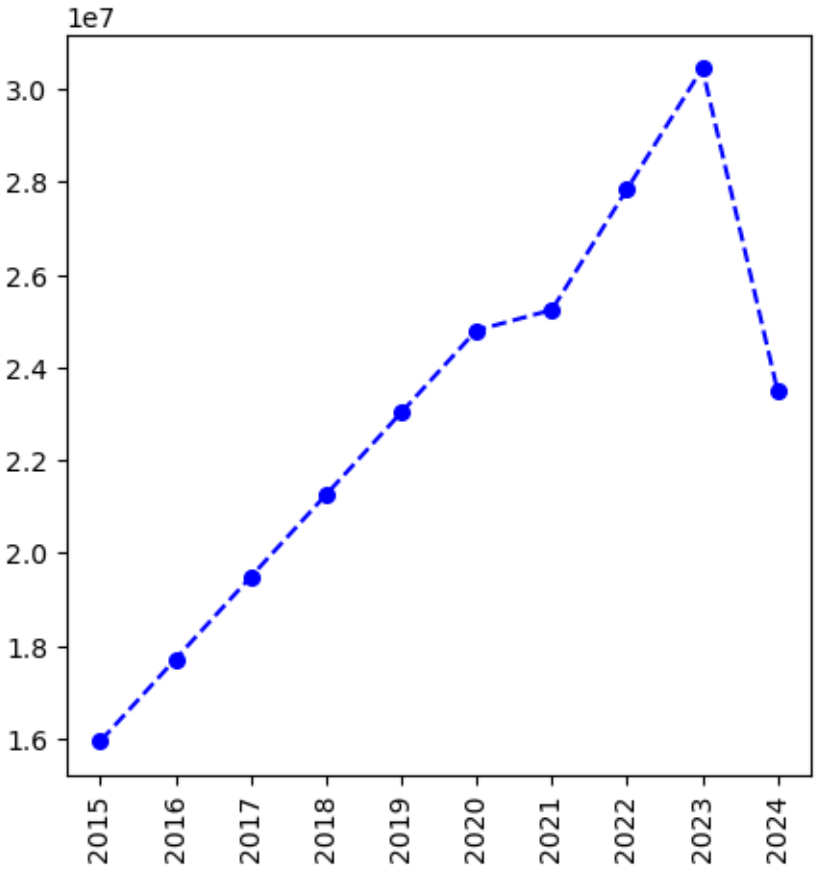
[24]:
```
plt.figure(figsize=(5,5))
plt.plot(Salary[0], c = 'b',ls='--', marker = 'o',ms = 5.5)
```

```
plt.show()
```

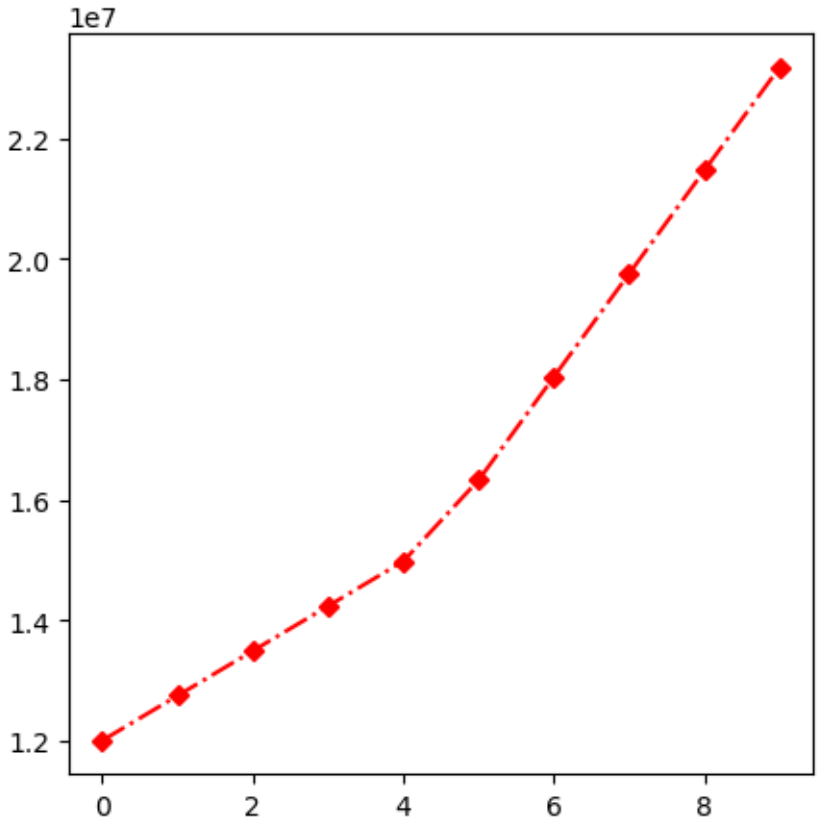


Get values on x-axis

[25]:
```
plt.figure(figsize=(5,5))
plt.plot(Salary[0], c = 'b',ls='--', marker = 'o',ms = 5.5)
plt.xticks(list(range(0,10)),Seasons,rotation = 'vertical')
plt.show()
```

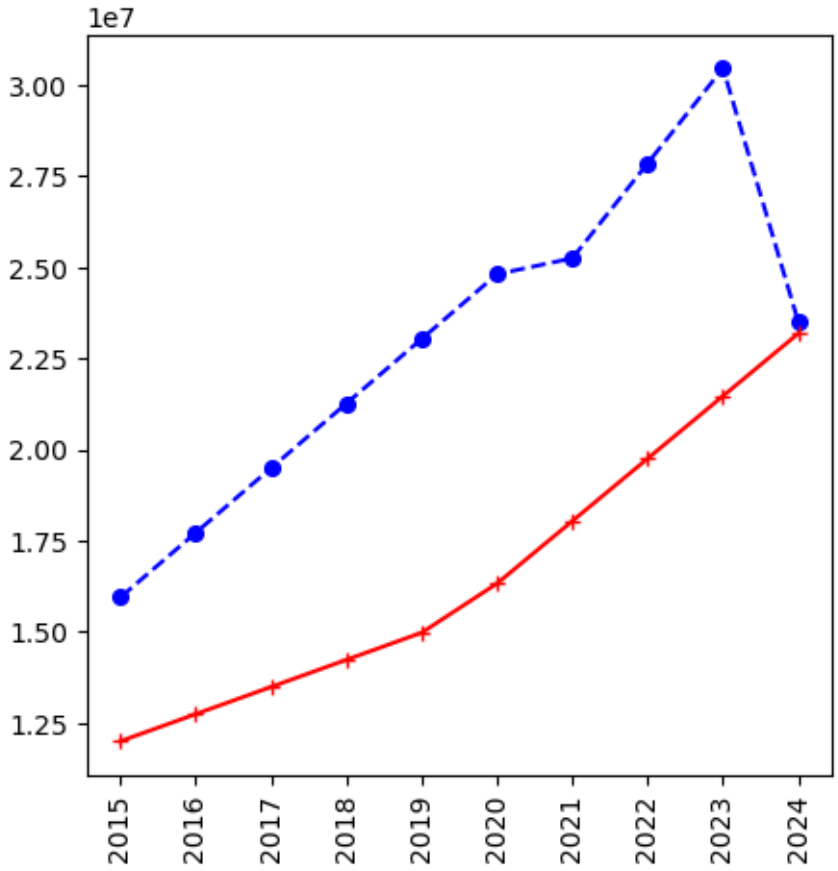


[29]:
```
# Plot for 2nd player
plt.figure(figsize=(5,5))
plt.plot(Salary[1], c = 'r',ls='-.', marker = 'D',ms = 5.5)
plt.show()
```



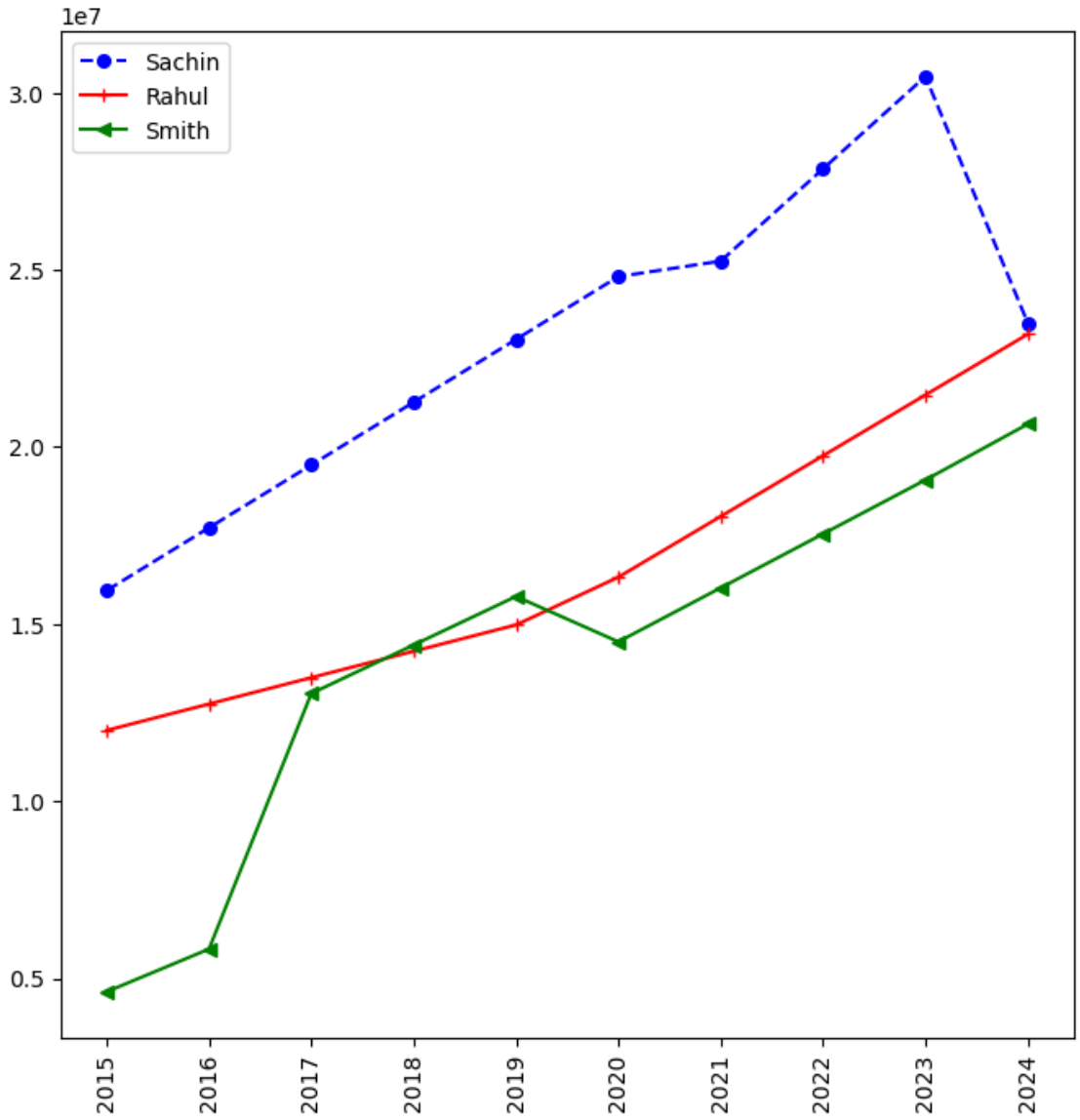
Plot 2 players salary in one

[31]:
```
plt.figure(figsize=(5,5))
plt.plot(Salary[0], c = 'b',ls='--', marker = 'o',ms = 5.5)
plt.plot(Salary[1], c = 'r',ls='-', marker = '+',ms = 5.5)
plt.xticks(list(range(0,10)),Seasons,rotation = 'vertical')
plt.show()
```



3 Players now

[32]:
```
plt.figure(figsize=(8,8))
plt.plot(Salary[0], c = 'b',ls='--', marker = 'o',ms = 5.5, label = Players[0])
plt.plot(Salary[1], c = 'r',ls='-', marker = '+',ms = 5.5, label = Players[1])
plt.plot(Salary[2], c = 'g',ls='-', marker = '<',ms = 5.5, label = Players[2])
plt.xticks(list(range(0,10)),Seasons,rotation = 'vertical')
plt.legend()
plt.show()
```

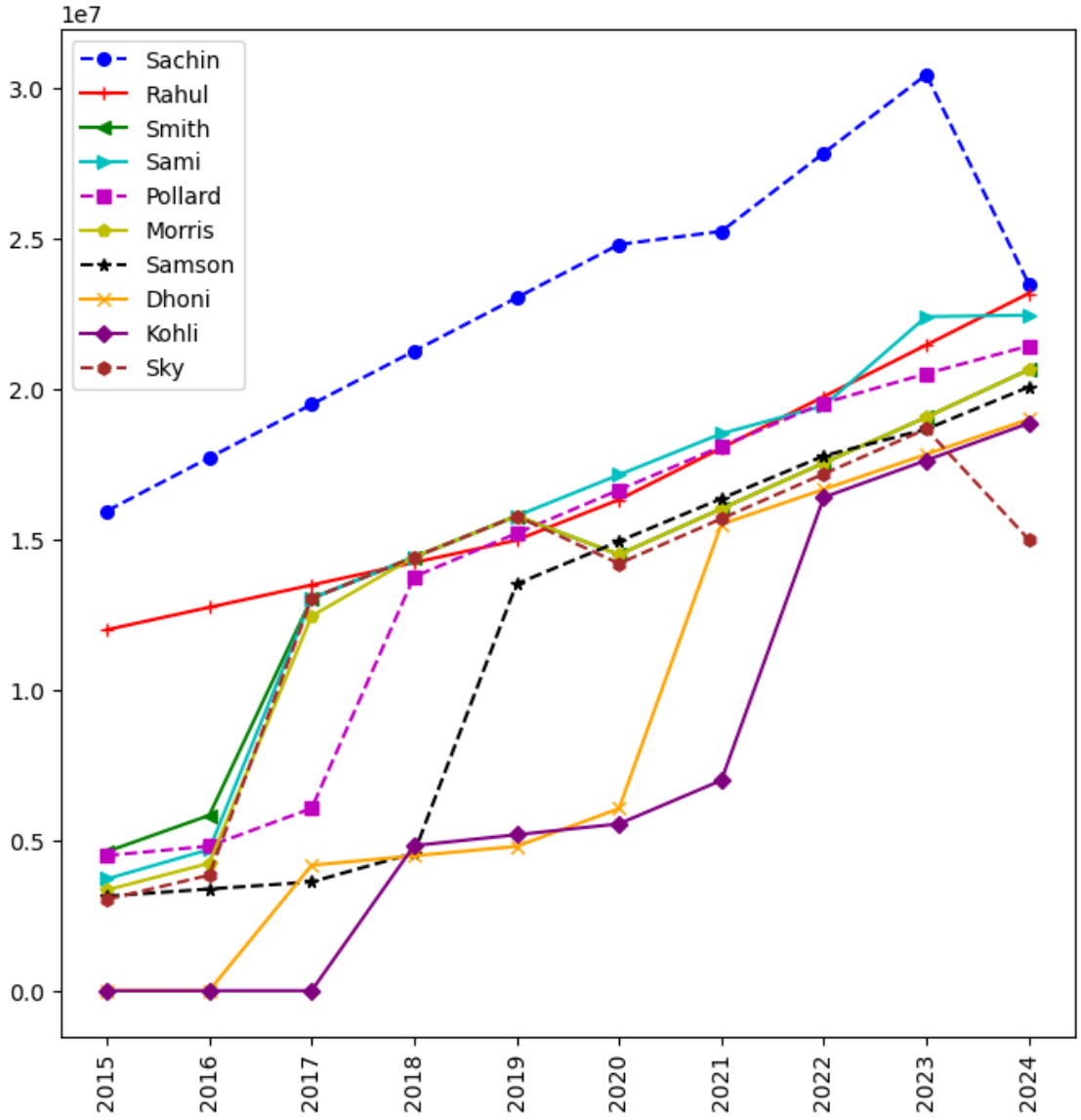


For **small to medium-sized datasets**, libraries like **Matplotlib** in Python are ideal for data visualization. They offer flexibility, customization, and full control over the plotting process.

However, when working with **large datasets**, Matplotlib may become less efficient and slower. In such cases, it's better to use tools like **Power BI** or **Tableau**, which are designed for handling and visualizing **larger volumes of data interactively** and efficiently.

[33]:
```
# So messy
plt.figure(figsize=(8,8))
plt.plot(Salary[0], c='b', ls='--', marker='o', ms=5.5, label=Players[0])
plt.plot(Salary[1], c='r', ls='-', marker='+', ms=5.5, label=Players[1])
plt.plot(Salary[2], c='g', ls='-', marker='<', ms=5.5, label=Players[2])
```

```
plt.plot(Salary[3], c='c', ls='-', marker='>', ms=5.5, label=Players[3])
plt.plot(Salary[4], c='m', ls='--', marker='s', ms=5.5, label=Players[4])
plt.plot(Salary[5], c='y', ls='-', marker='p', ms=5.5, label=Players[5])
plt.plot(Salary[6], c='k', ls='--', marker='*', ms=5.5, label=Players[6])
plt.plot(Salary[7], c='orange', ls='-', marker='x', ms=5.5, label=Players[7])
plt.plot(Salary[8], c='purple', ls='-', marker='D', ms=5.5, label=Players[8])
plt.plot(Salary[9], c='brown', ls='--', marker='h', ms=5.5, label=Players[9])
plt.xticks(list(range(0,10)), Seasons, rotation='vertical')
plt.legend()
plt.show()
```



[ ]: