

Matrix Visualization

MATRICES / NUMPY -----

- Matrix is the tabular representation of the data
- Lot of datas are stored in table format,that is why Matrices is very very important topic in python
- as we working on dataframe so matrices are played a major rule
- List is one dimension & matrix is multidimension
- indexation is very important to plot the datapoints
- we will see tht & we gonna analyze the NBA players
- hear i have taken top 10 highest paid player in 2015-2016 season
- we will analyze how 10 players have been playing over the past 10 years & we had the data for past 10yrs yrs
- our main goal is to find trends,patterns & their performance for the past 10 yrs
- ultimately they haven't always been top 10 player & lets see how they improving, what actually secretes or patterns
- dont worry guys if you dont know anything about basket ball NBA
- I will explain indepth of everything
- lets analyze the statistics of the basket ball player
- gp - total games played,mpg - minutes per game,field goal(accuracy), ppg (points per game) -- this is no of point player has scores in that season
- guys slowly i am bringing you into data analytics, jump into datavisualization using python
- i will give you the this code can everybody copy and paste your jupyter notebook
- Now i will explain with matrices

```
In [1]: #Import numpy
import numpy as np
```

```
In [2]: #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}
```

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

```
In [4]: #Salaries
Sachin_Salary = [15946875, 17718750, 19490625, 21262500, 23034375, 24806250, 25244493, 27812000, 29600000, 31400000]
Rahul_Salary = [12000000, 12744189, 13488377, 14232567, 14976754, 16324500, 18038573, 19750000, 21500000, 23250000]
Smith_Salary = [4621800, 5828090, 13041250, 14410581, 15779912, 14500000, 16022500, 17545000, 19000000, 20500000]
Sami_Salary = [3713640, 4694041, 13041250, 14410581, 15779912, 17149243, 18518574, 19450000, 20300000, 21150000]
```

```

Pollard_Salary = [4493160,4806720,6061274,13758000,15202590,16647180,18091770,19536
Morris_Salary = [3348000,4235220,12455000,14410581,15779912,14500000,16022500,17545
Samson_Salary = [3144240,3380160,3615960,4574189,13520500,14940153,16359805,1777945
Dhoni_Salary = [0,0,4171200,4484040,4796880,6053663,15506632,16669630,17832627,1899
Kohli_Salary = [0,0,0,4822800,5184480,5546160,6993708,16402500,17632688,18862875]
Sky_Salary = [3031920,3841443,13041250,14410581,15779912,14200000,15691000,17182000

```

```

In [5]: #Matrix
Salary = np.array([Sachin_Salary, Rahul_Salary, Smith_Salary, Sami_Salary, Pollard_

```

```

In [6]: #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]

```

```

In [7]: #Matrix
Games = np.array([Sachin_G, Rahul_G, Smith_G, Sami_G, Pollard_G, Morris_G, Samson_G

```

```

In [8]: #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]

```

```

In [9]: #Matrix
Points = np.array([Sachin_PTS, Rahul_PTS, Smith_PTS, Sami_PTS, Pollard_PTS, Morris_

```

```

In [10]: Salary          # Matrix format

```

```
Out[10]: 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]])
```

In [11]: Games *# Building Matrix*

```
Out[11]: 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]])
```

In [12]: Points

```
Out[12]: 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,  96],
               [1292, 1443, 1695, 1624, 1503, 1784, 1113, 1296, 1297,  64],
               [1572, 1561, 1496, 1746, 1678, 1438, 1025, 1232, 1281,  92],
               [1258, 1104, 1684, 1781,  841, 1268, 1189, 1186, 1185, 156],
               [ 903,  903, 1624, 1871, 2472, 2161, 1850, 2280, 2593,  68],
               [ 597,  597,  597, 1361, 1619, 2026,  852,  0, 159,  90],
               [2040, 1397, 1254, 2386, 2045, 1941, 1082, 1463, 1028, 1331]])
```

In [13]: Games

```
Out[13]: 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]])
```

```
In [14]: Games[5]
```

```
Out[14]: array([70, 69, 67, 77, 70, 77, 57, 74, 79, 44])
```

```
In [15]: Games [0:5]
```

```
Out[15]: 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]])
```

```
In [16]: Games [0,2]
```

```
Out[16]: 82
```

```
In [17]: Salary
```

```
Out[17]: 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]])
```

```
In [18]: Games
```

```
Out[18]: 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]])
```

```
In [19]: Salary/Games
```

```
C:\Users\Windows10 Pro\AppData\Local\Temp\ipykernel_15504\3709746658.py:1: RuntimeWarning: divide by zero encountered in divide
Salary/Games
```

```
Out[19]: 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]])
```

```
In [20]: np.round(Salary/Games)
```

C:\Users\Windows10 Pro\AppData\Local\Temp\ipykernel_15504\3232172828.py:1: RuntimeWarning: divide by zero encountered in divide
 np.round(Salary/Games)

```
Out[20]: array([[ 199336.,  230114.,  237691.,  259299.,  315539.,  302515.,
                  435250.,  357040.,  5075634.,  671429.],
                [ 146341.,  223582.,  164492.,  180159.,  197063.,  226729.,
                  300643.,  274342.,  271731.,  289760.],
                [  58504.,   74719.,  173883.,  177908.,  207630.,  183544.,
                  258427.,  230855.,  247630.,  299194.],
                [  46420.,   72216.,  169367.,  218342.,  228694.,  222717.,
                  336701.,  290299.,  291006.,  561450.],
                [  54795.,   58619.,   73918.,  174152.,  185397.,  213425.,
                  335033.,  257057.,  288918.,  522836.],
                [  47829.,   61380.,  185896.,  187150.,  225427.,  188312.,
                  281096.,  237095.,  241361.,  469191.],
                [  40311.,   52815.,   45200.,   58643.,  300456.,  186752.,
                  272663.,  253992.,  301104.,  244739.],
                [    0.,    0.,   52140.,   60595.,   58499.,   77611.,
                  234949.,  205798.,  220156.,  703542.],
                [    0.,    0.,    0.,   59541.,   66468.,   68471.,
                  179326.,   inf,  1763269.,  369860.],
                [  40426.,   75322.,  255711.,  182412.,  204934.,  186842.,
                  320224.,  249014.,  345796.,  241935.]])
```

```
In [21]: np.round(Salary//Games)
```

C:\Users\Windows10 Pro\AppData\Local\Temp\ipykernel_15504\3663165759.py:1: RuntimeWarning: divide by zero encountered in floor_divide
 np.round(Salary//Games)

```
Out[21]: 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.]])
```

```
In [22]: import warnings
         warnings.filterwarnings('ignore')
```

```
In [23]: import matplotlib.pyplot as plt
```

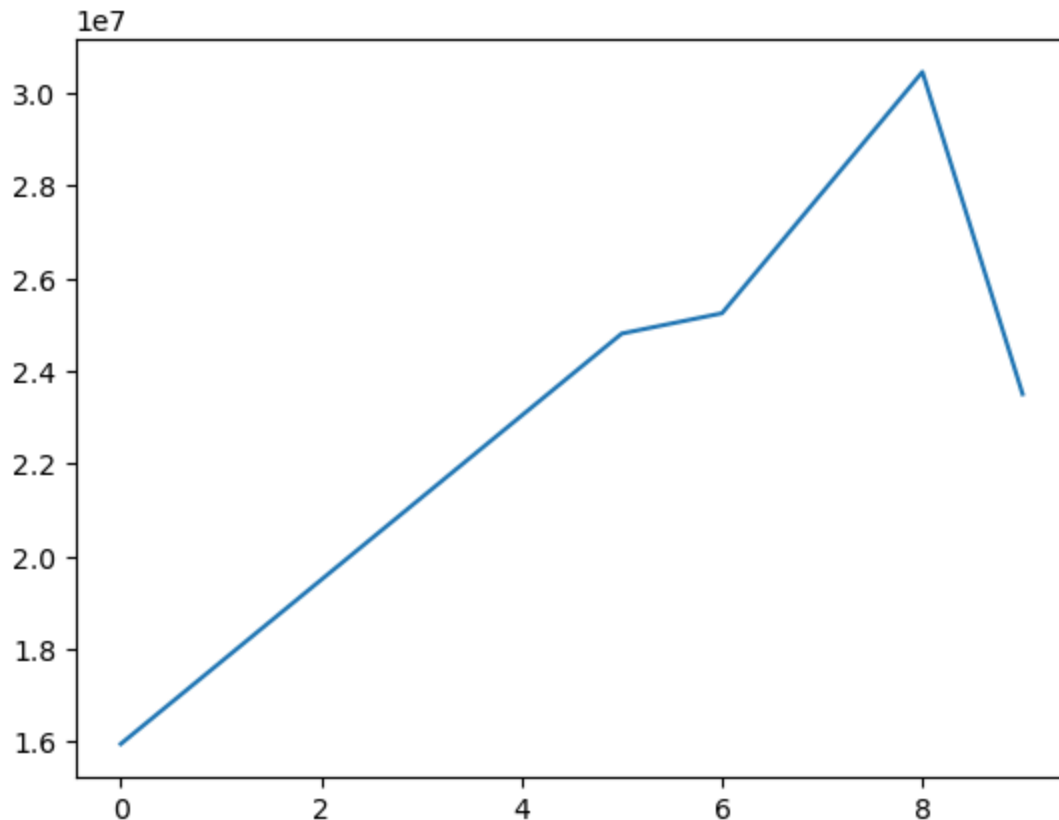
```
In [24]: Salary
```

```
Out[24]: 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]])
```

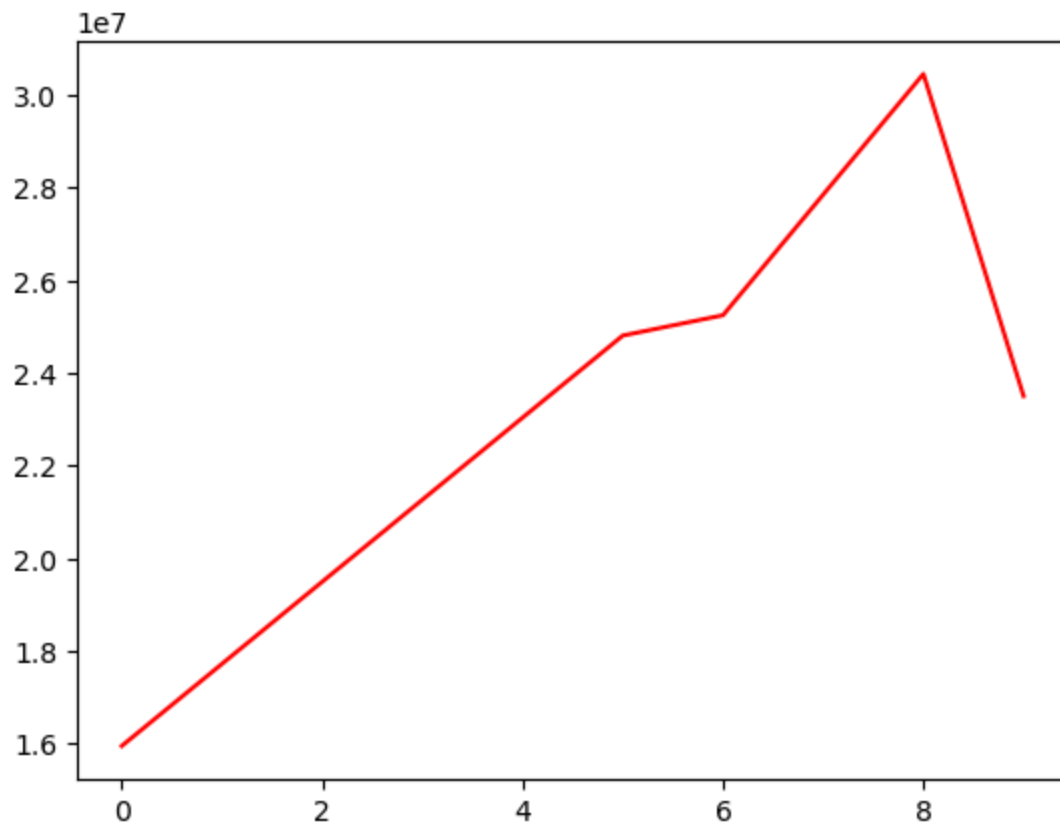
```
In [25]: Salary[0]
```

```
Out[25]: array([15946875, 17718750, 19490625, 21262500, 23034375, 24806250,
                25244493, 27849149, 30453805, 23500000])
```

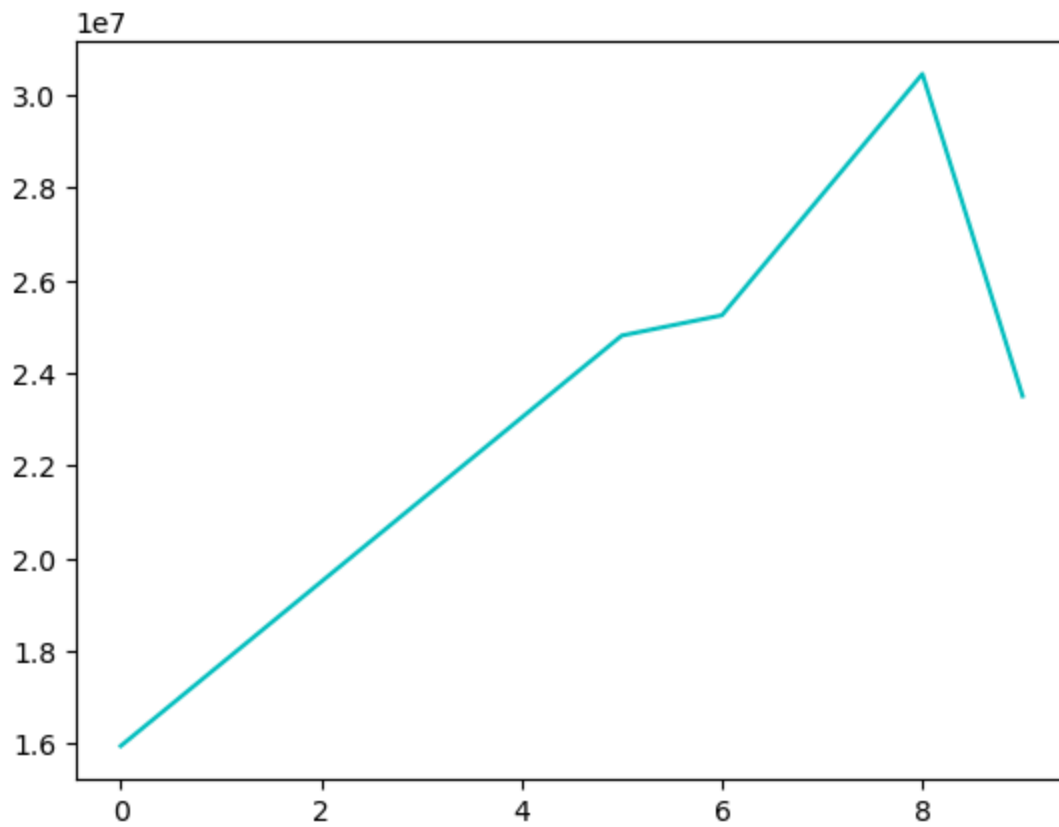
```
In [26]: plt.plot(Salary[0]) # shift+tab to show library info
plt.show()
```



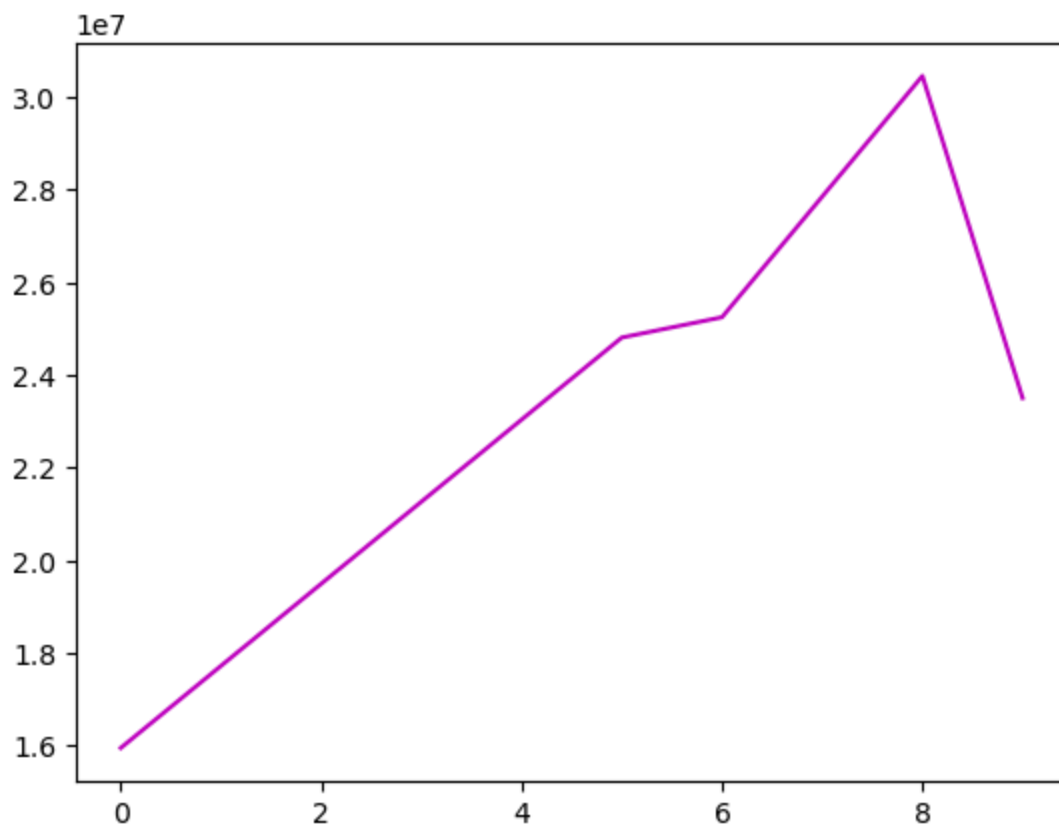

```
In [27]: plt.plot(Salary[0], c='r')  
plt.show()
```



```
In [28]: plt.plot(Salary[0], c='c')  
plt.show()
```



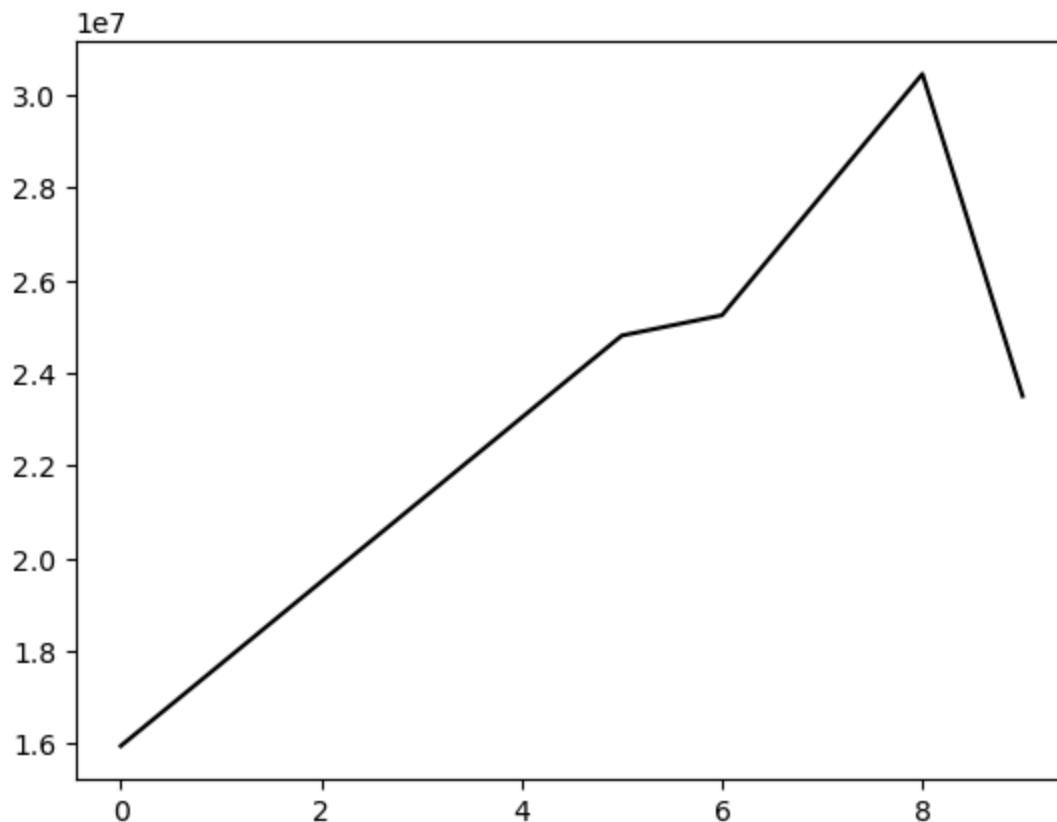
```
In [29]: plt.plot(Salary[0], c='m')  
plt.show()
```



05th May 2025

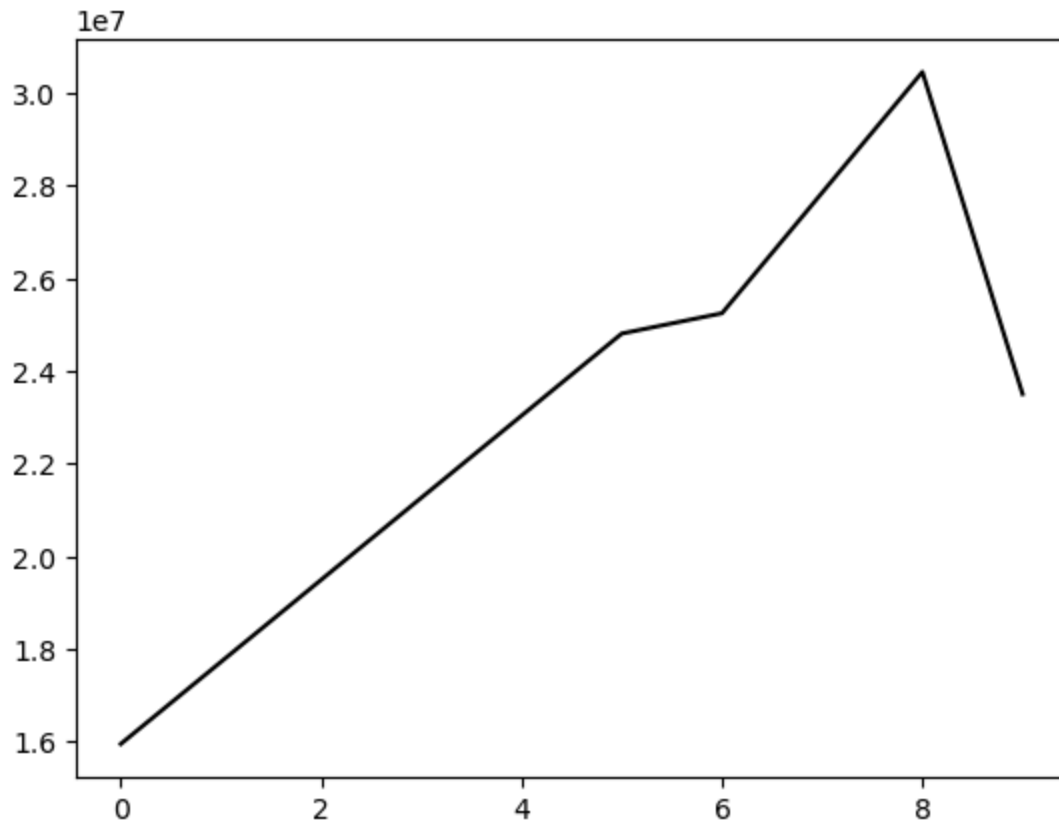
```
In [30]: plt.plot(Salary[0], color = 'black')
```

```
Out[30]: [<matplotlib.lines.Line2D at 0x1be69d35970>]
```



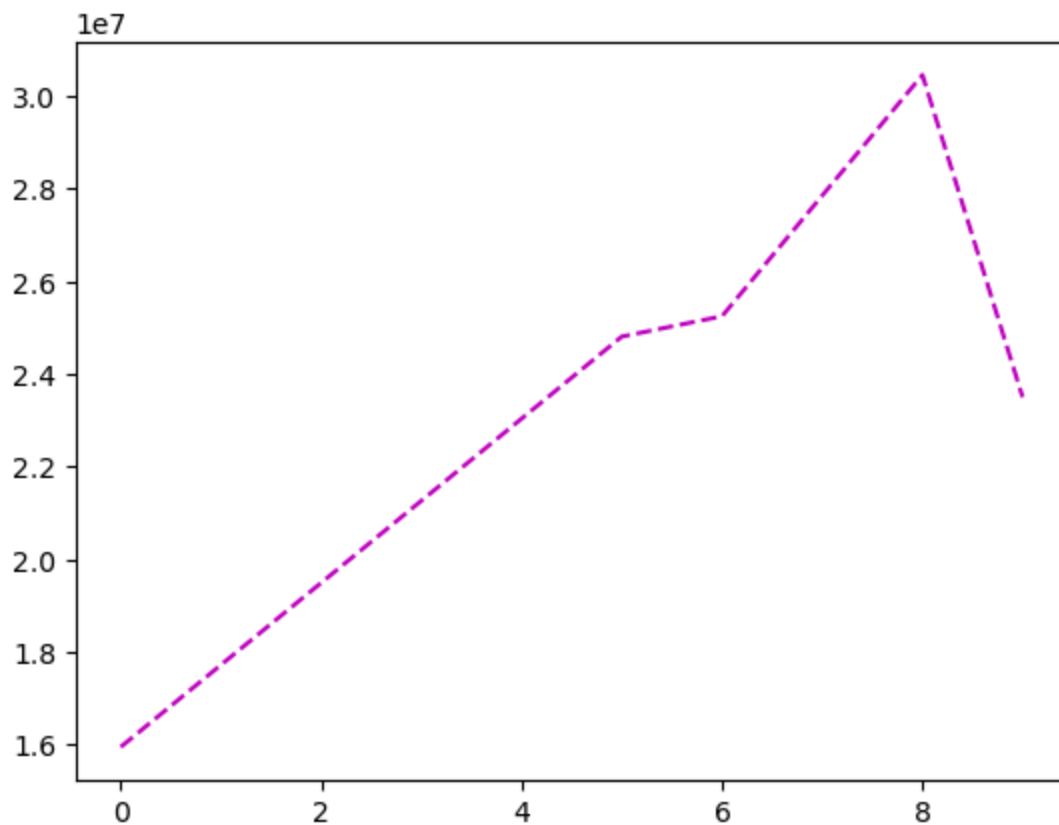
```
In [31]: plt.plot(Salary[0], c = 'k') # short forms
```

```
Out[31]: [<matplotlib.lines.Line2D at 0x1be69de97c0>]
```



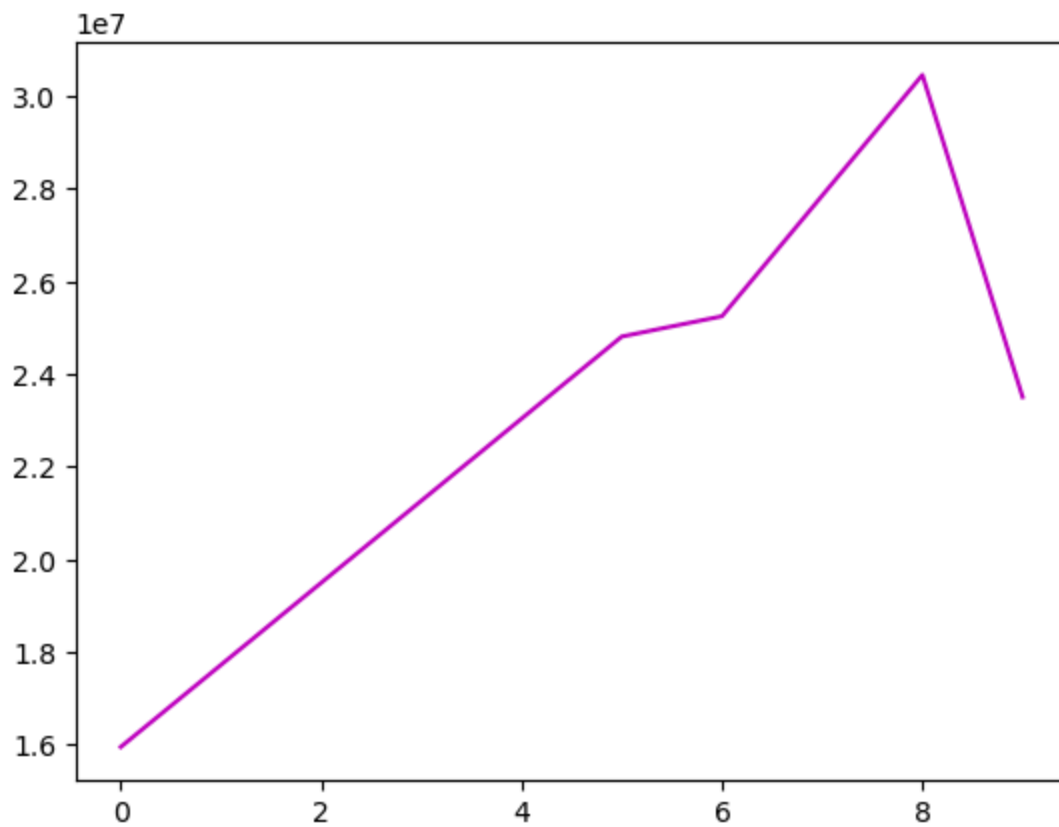
```
In [32]: plt.plot(Salary[0], c = 'm', ls = '--') # short forms
```

```
Out[32]: [matplotlib.lines.Line2D at 0x1be69e3cfb0>]
```



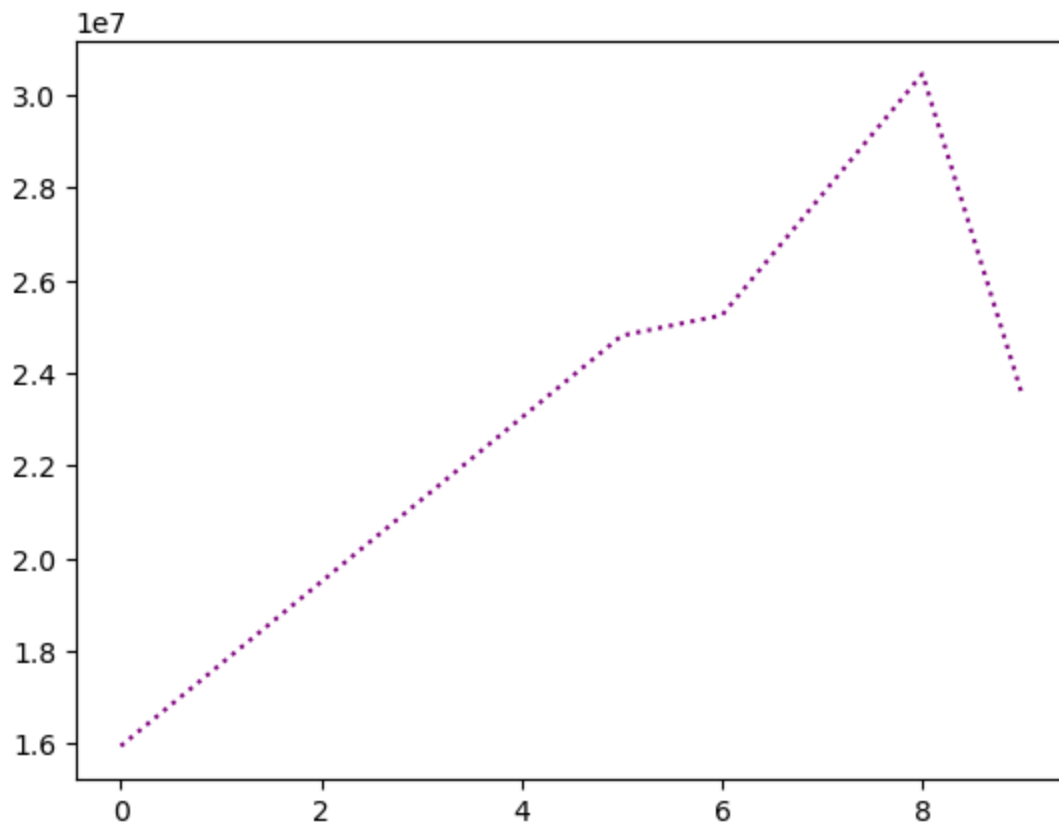
```
In [33]: plt.plot(Salary[0], c = 'm', ls = '-') # short forms
```

```
Out[33]: [<matplotlib.lines.Line2D at 0x1be6cf14860>]
```

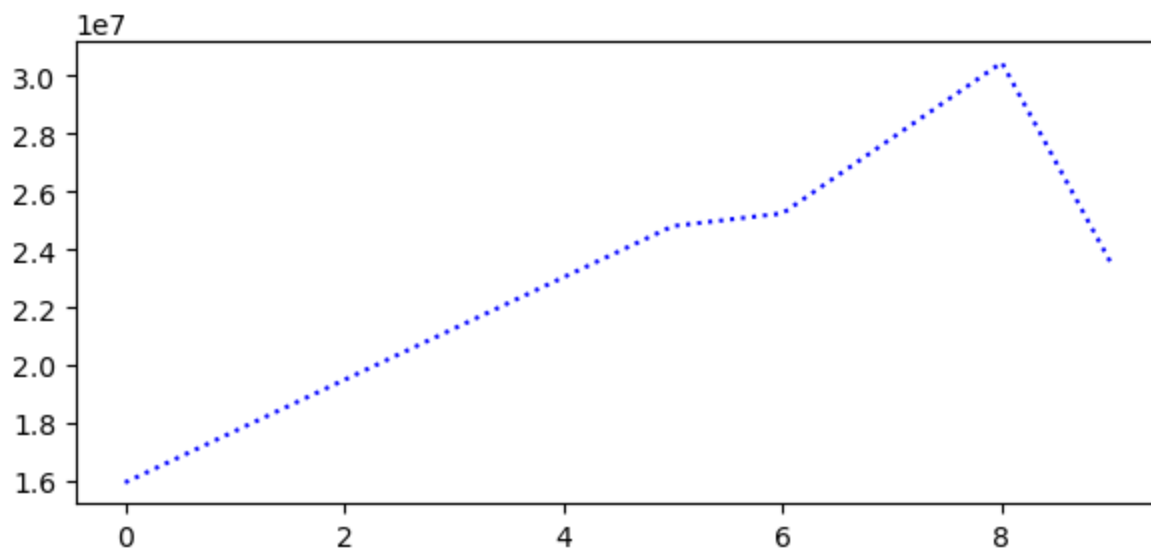


```
In [34]: plt.plot(Salary[0], c = 'purple', ls = 'dotted')
```

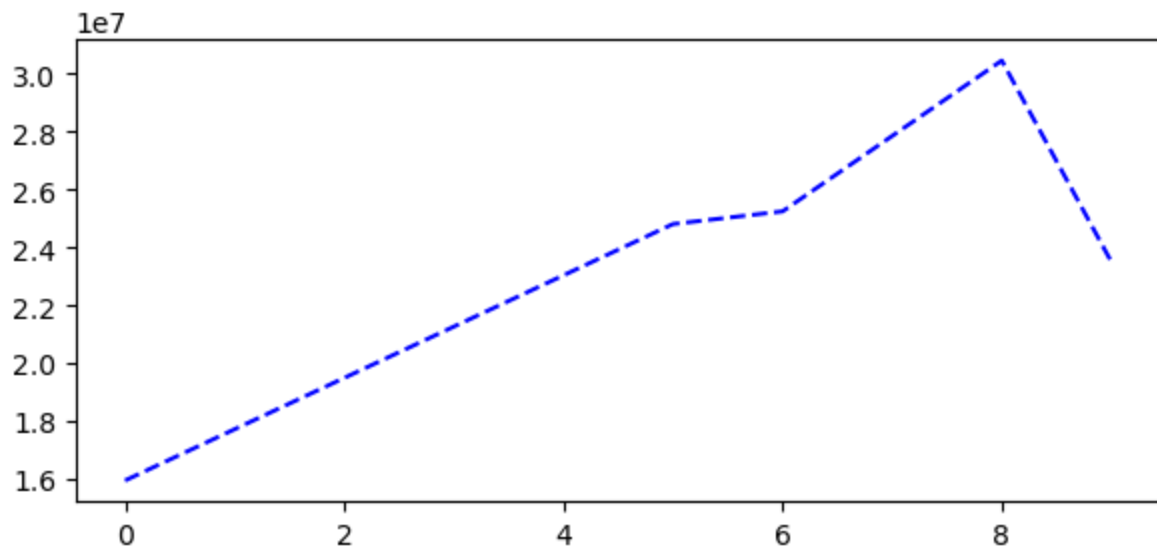
```
Out[34]: [<matplotlib.lines.Line2D at 0x1be6cf7cfe0>]
```



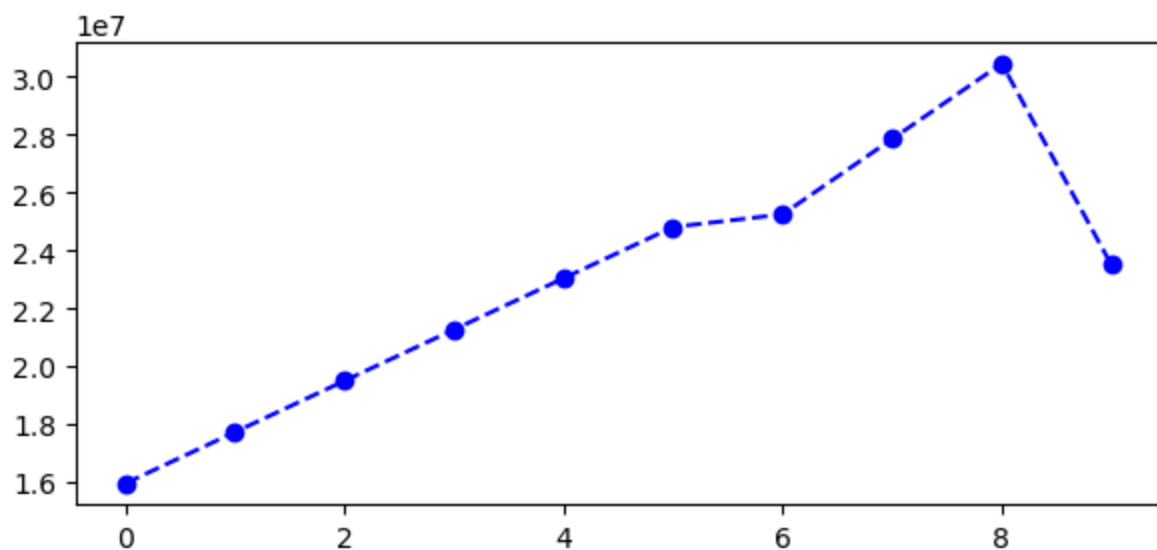
```
In [35]: plt.rcParams['figure.figsize'] = 7,3      # 7 is Width & 3 is Height  
plt.plot(Salary[0], c = 'Blue', ls = 'dotted')  
plt.show()
```



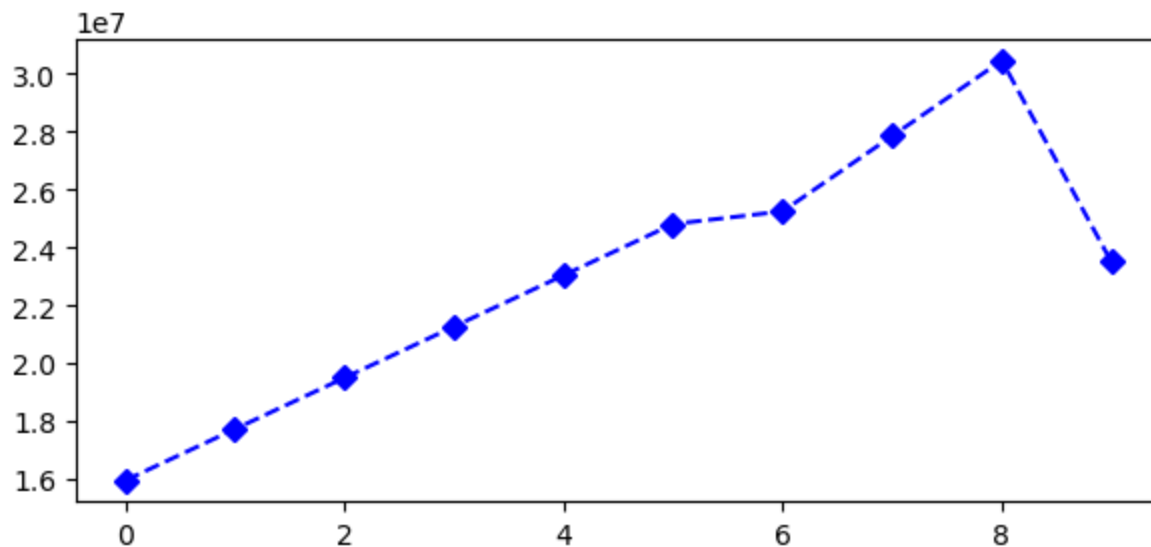
```
In [36]: plt.rcParams['figure.figsize'] = 7,3      # 7 is Width & 3 is Height  
plt.plot(Salary[0], c = 'Blue', ls = '--')  
plt.show()
```



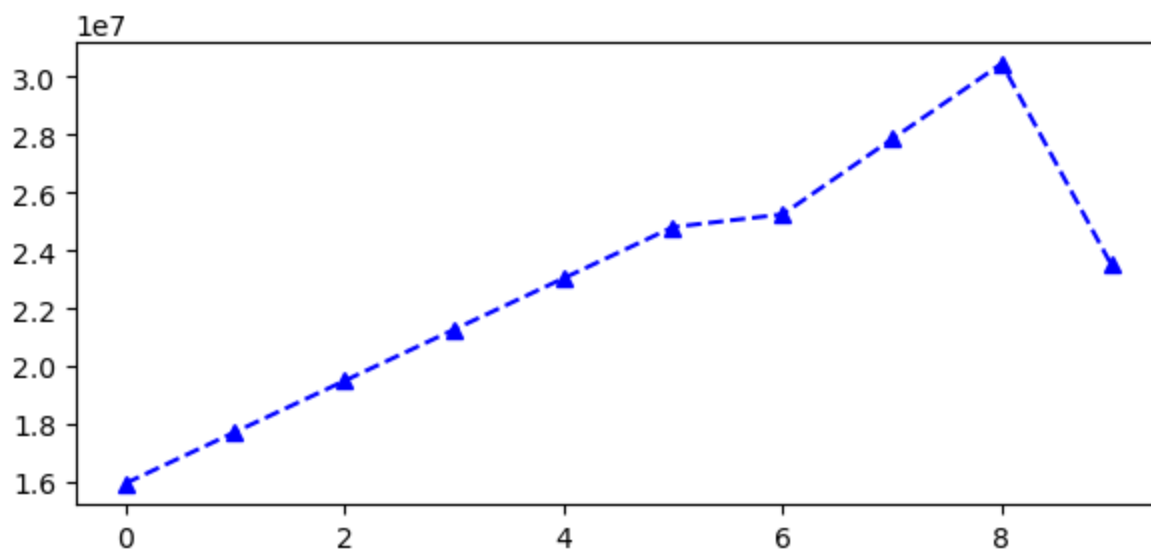
```
In [37]: plt.rcParams['figure.figsize'] = 7,3      # 7 is Width & 3 is Height  
plt.plot(Salary[0], c = 'Blue', ls = '--', marker = 'o')  
plt.show()
```



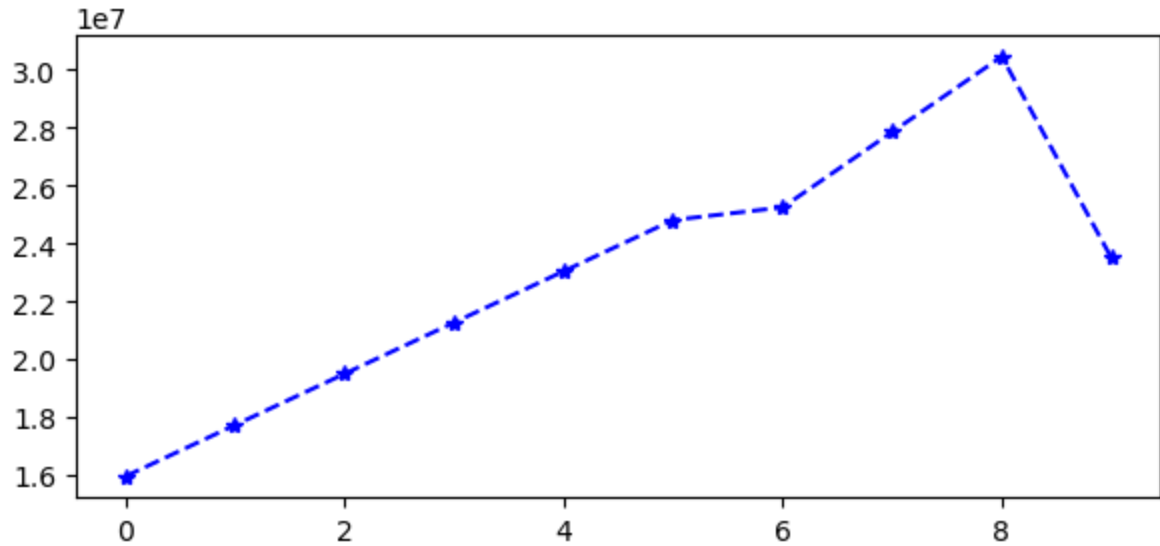
```
In [38]: plt.rcParams['figure.figsize'] = 7,3      # 7 is Width & 3 is Height  
plt.plot(Salary[0], c = 'Blue', ls = '--', marker = 'D')  
plt.show()
```



```
In [39]: plt.rcParams['figure.figsize'] = 7,3      # 7 is Width & 3 is Height
plt.plot(Salary[0], c = 'Blue', ls = '--', marker = '^')
plt.show()
```



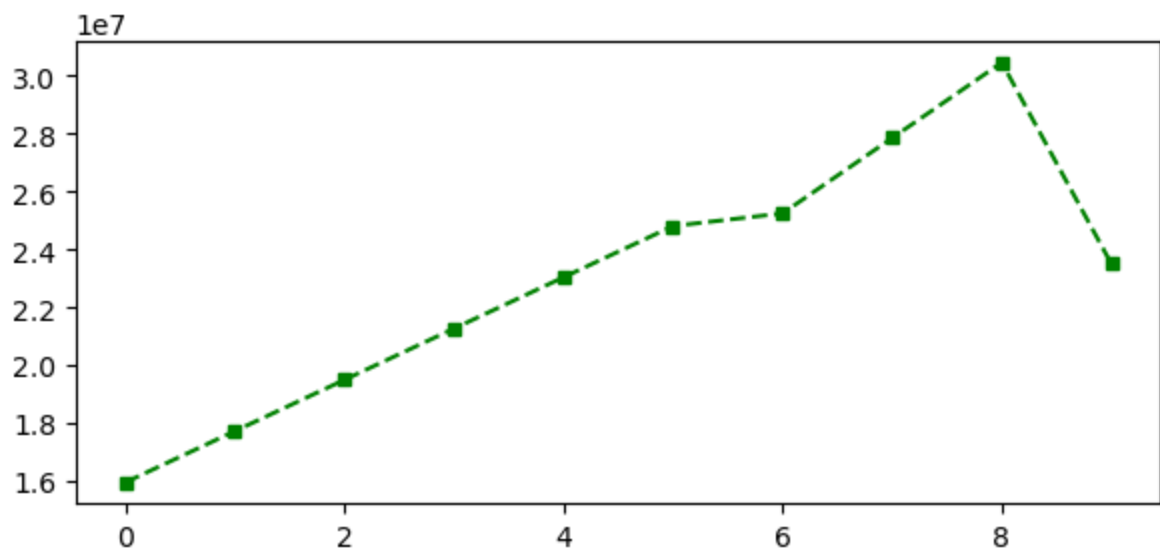
```
In [40]: plt.rcParams['figure.figsize'] = 7,3      # 7 is Width & 3 is Height
plt.plot(Salary[0], c = 'Blue', ls = '--', marker = '*')      # shift+tab
plt.show()
```

In [41]: Games

```
Out[41]: 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]])
```

```
In [42]: plt.rcParams['figure.figsize'] = 7,3      # 7 is Width & 3 is Height
plt.plot(Salary[0], c = 'Green', ls = '--', marker = 's', ms=5)  # ms = marker size
plt.show()
```



In [43]: list(range(0,10))

```
Out[43]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

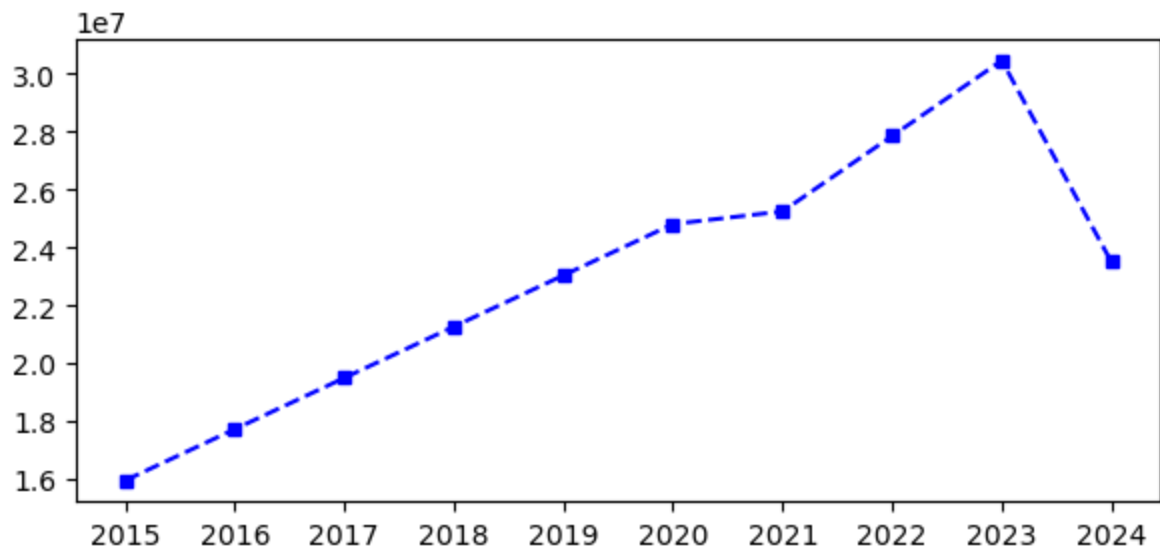
```
In [44]: Sdict
```

```
Out[44]: {'2015': 0,
          '2016': 1,
          '2017': 2,
          '2018': 3,
          '2019': 4,
          '2020': 5,
          '2021': 6,
          '2022': 7,
          '2023': 8,
          '2024': 9}
```

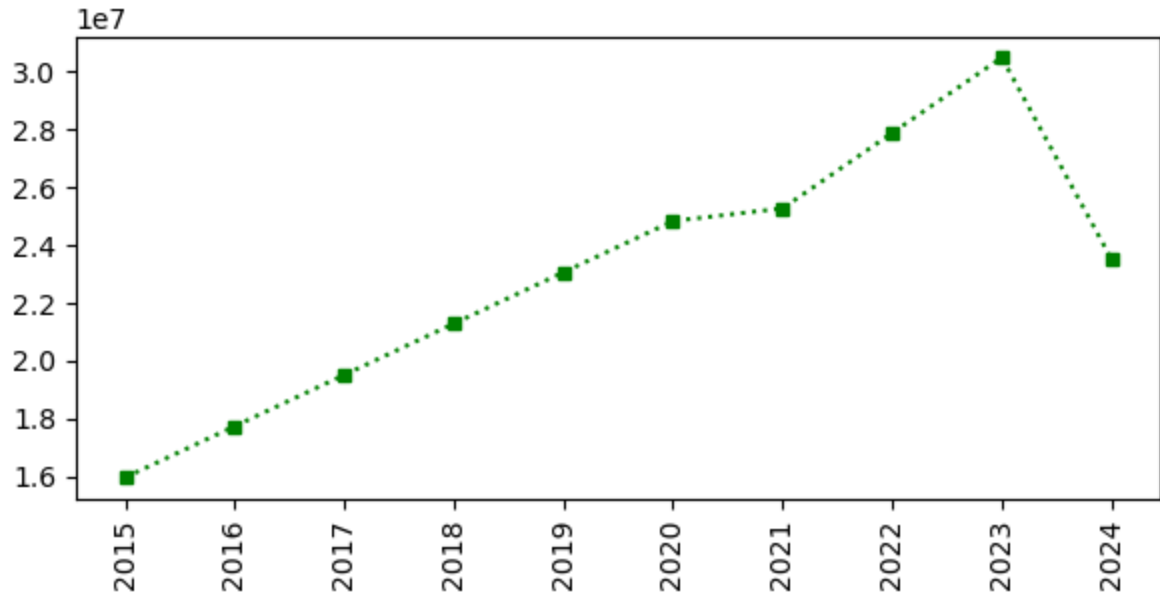
```
In [45]: Pdict
```

```
Out[45]: {'Sachin': 0,
          'Rahul': 1,
          'Smith': 2,
          'Sami': 3,
          'Pollard': 4,
          'Morris': 5,
          'Samson': 6,
          'Dhoni': 7,
          'Kohli': 8,
          'Sky': 9}
```

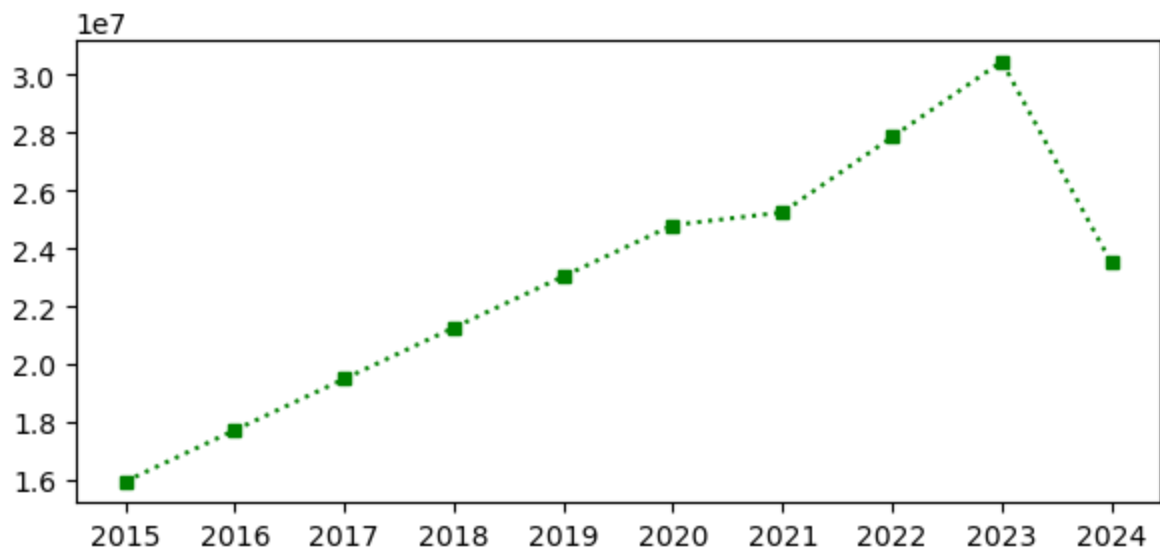
```
In [46]: plt.rcParams['figure.figsize'] = 7,3
plt.plot(Salary[0], c = 'blue', ls = '--', marker = 's', ms=5)
plt.xticks(list(range(0,10)), Seasons)
plt.show()
```



```
In [47]: plt.rcParams['figure.figsize'] = 7,3
plt.plot(Salary[0], c = 'Green', ls = ':', marker = 's', ms=5)
plt.xticks(list(range(0,10)), Seasons, rotation = 'vertical')
plt.show()
```



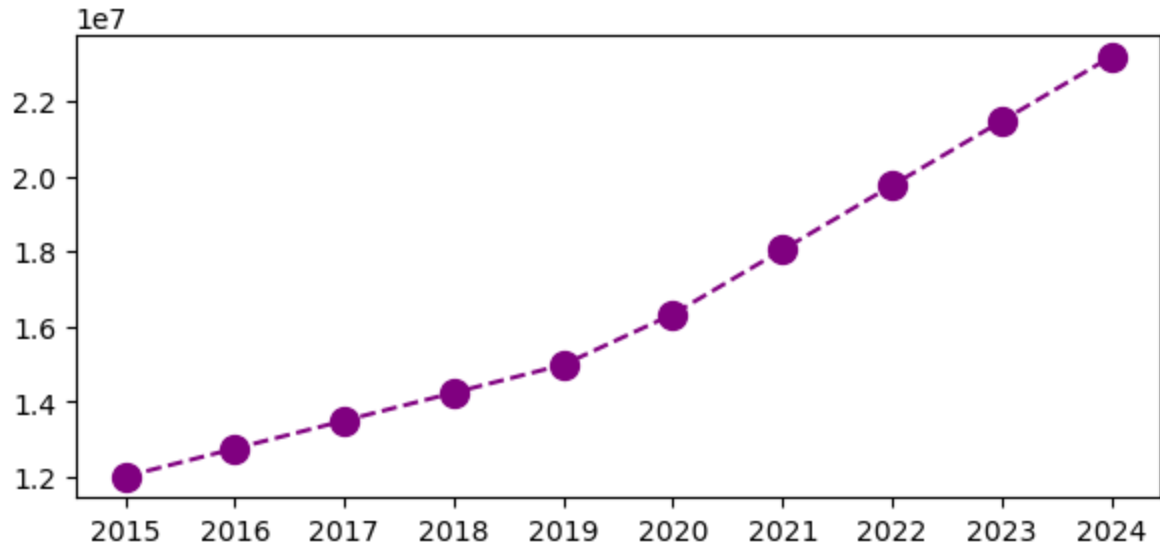
```
In [48]: plt.rcParams['figure.figsize'] = 7,3
plt.plot(Salary[0], c = 'Green', ls = 'dotted', marker = 's', ms=5)
plt.xticks(list(range(0,10)), Seasons, rotation = 'horizontal')
plt.show()
```



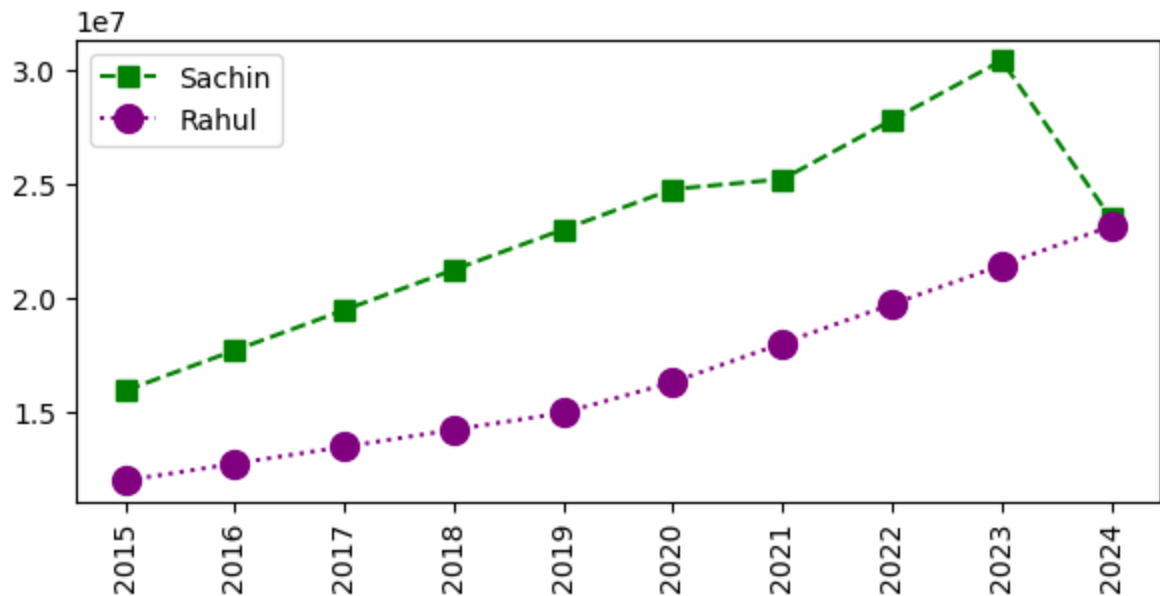
```
In [49]: Salary[1]
```

```
Out[49]: array([12000000, 12744189, 13488377, 14232567, 14976754, 16324500,
18038573, 19752645, 21466718, 23180790])
```

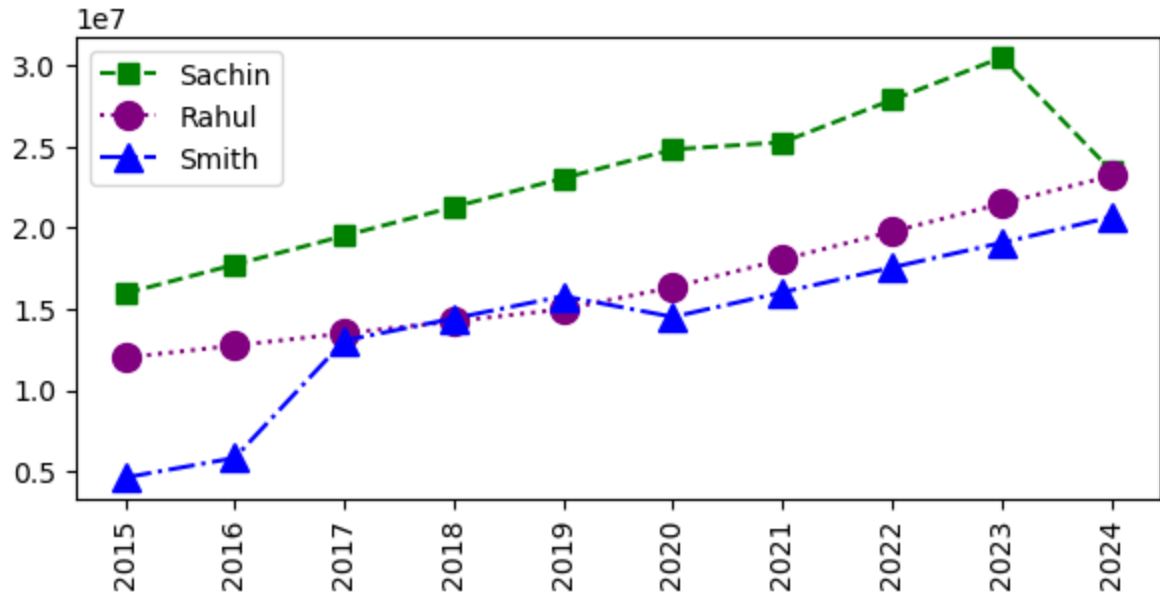
```
In [50]: plt.rcParams['figure.figsize'] = 7,3
plt.plot(Salary[1], c = 'purple', ls = '--', marker = 'o', ms=10, label = Players[1])
plt.xticks(list(range(0,10)), Seasons, rotation = 'horizontal')
plt.show()
```



```
In [51]: plt.rcParams['figure.figsize'] = 7, 3
plt.plot(Salary[0], c='Green', ls='--', marker='s', ms=7, label=Players[0])
plt.plot(Salary[1], c='purple', ls=':', marker='o', ms=10, label=Players[1])
plt.xticks(list(range(0, 10)), Seasons, rotation='vertical')
plt.legend()
plt.show()
```



```
In [52]: plt.rcParams['figure.figsize'] = 7, 3
plt.plot(Salary[0], c='Green', ls='--', marker='s', ms=7, label=Players[0])
plt.plot(Salary[1], c='purple', ls=':', marker='o', ms=10, label=Players[1])
plt.plot(Salary[2], c='Blue', ls='-.', marker='^', ms=10, label=Players[2])
plt.xticks(list(range(0, 10)), Seasons, rotation='vertical')
plt.legend()
plt.show()
```



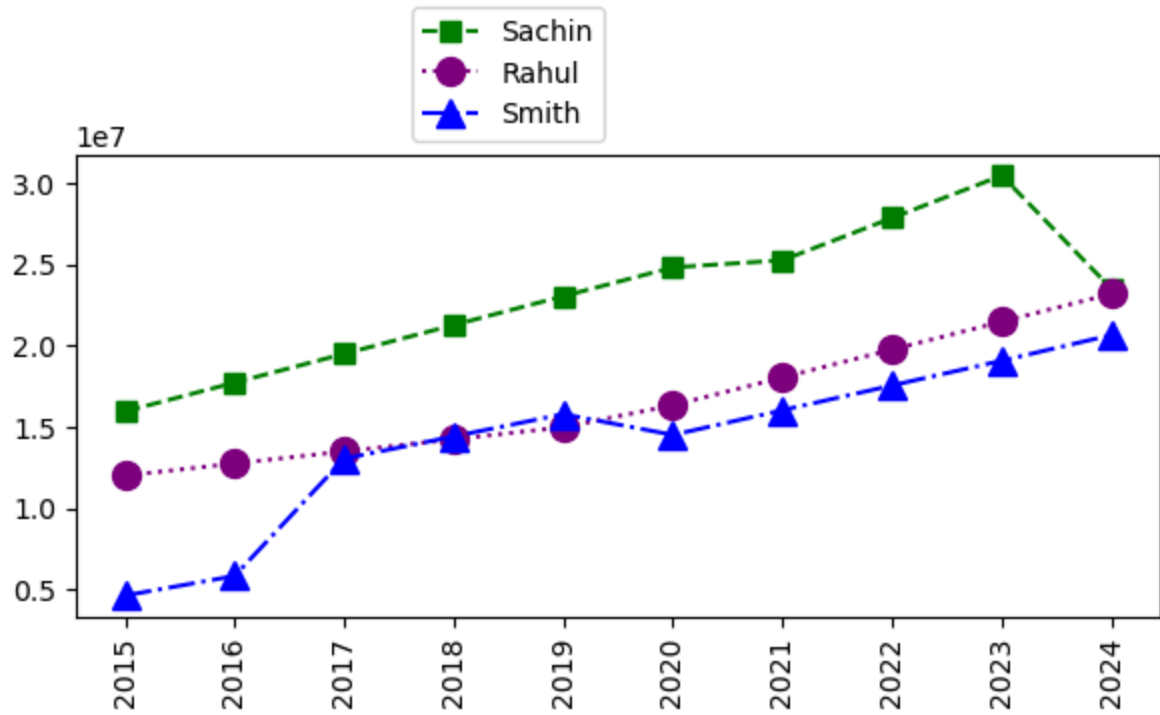
In [53]: Games

```
Out[53]: 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]])
```

```
In [54]: plt.rcParams['figure.figsize'] = 7, 3
plt.plot(Salary[0], c='Green', ls='--', marker='s', ms=7, label=Players[0])
plt.plot(Salary[1], c='purple', ls=':', marker='o', ms=10, label=Players[1])
plt.plot(Salary[2], c='Blue', ls='-.', marker='^', ms=10, label=Players[2])

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

plt.show()
```



```
In [55]: 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()
```

```

-----
ValueError                                Traceback (most recent call last)
Cell In[55], line 12
      9 plt.plot(Salary[8], c='Red', ls = '--', marker = 's', ms = 7, label = Player
s[8])
     10 plt.plot(Salary[9], c='Red', ls = '--', marker = 'o', ms = 7, label = Player
s[9])
--> 12 plt.legend(loc = 'lower right',bbox_to_anchor=(0.5,1) )
     13 plt.xticks(list(range(0,10)), Seasons,rotation='vertical')
     15 plt.show()

File C:\ProgramData\anaconda3\Lib\site-packages\matplotlib\pyplot.py:3384, in legend
(*args, **kwargs)
     3382 @_copy_docstring_and_deprecators(Axes.legend)
     3383 def legend(*args, **kwargs) -> Legend:
-> 3384     return gca().legend(*args, **kwargs)

File C:\ProgramData\anaconda3\Lib\site-packages\matplotlib\axes\_axes.py:323, in Axes
.legend(self, *args, **kwargs)
     206 """
     207 Place a legend on the Axes.
     208
     (...)
     320 .. plot:: gallery/text_labels_and_annotations/legend.py
     321 """
     322 handles, labels, kwargs = mlegend._parse_legend_args([self], *args, **kwargs)
-> 323 self.legend_ = mlegend.Legend(self, handles, labels, **kwargs)
     324 self.legend_.remove_method = self._remove_legend
     325 return self.legend_

File C:\ProgramData\anaconda3\Lib\site-packages\matplotlib\legend.py:566, in Legend.
__init__(self, parent, handles, labels, loc, numpoints, markerscale, markerfirst, re
verse, scatterpoints, scatteryoffsets, prop, fontsize, labelcolor, borderpad, labels
pacing, handlelength, handleheight, handletextpad, borderaxespad, columnspacing, nc
ols, mode, fancybox, shadow, title, title_fontsize, framealpha, edgecolor, facecolor,
bbox_to_anchor, bbox_transform, frameon, handler_map, title_fontproperties, alignmen
t, ncol, draggable)
     563 self._init_legend_box(handles, labels, markerfirst)
     565 # Set legend location
-> 566 self.set_loc(loc)
     568 # figure out title font properties:
     569 if title_fontsize is not None and title_fontproperties is not None:

File C:\ProgramData\anaconda3\Lib\site-packages\matplotlib\legend.py:687, in Legend.
set_loc(self, loc)
     685         loc = locs[0] + ' ' + locs[1]
     686         # check that loc is in acceptable strings
-> 687         loc = _api.check_getitem(self.codes, loc=loc)
     688 elif np.iterable(loc):
     689         # coerce iterable into tuple
     690         loc = tuple(loc)

File C:\ProgramData\anaconda3\Lib\site-packages\matplotlib\_api\_init_.py:183, in
check_getitem(mapping, **kwargs)
     181     return mapping[v]

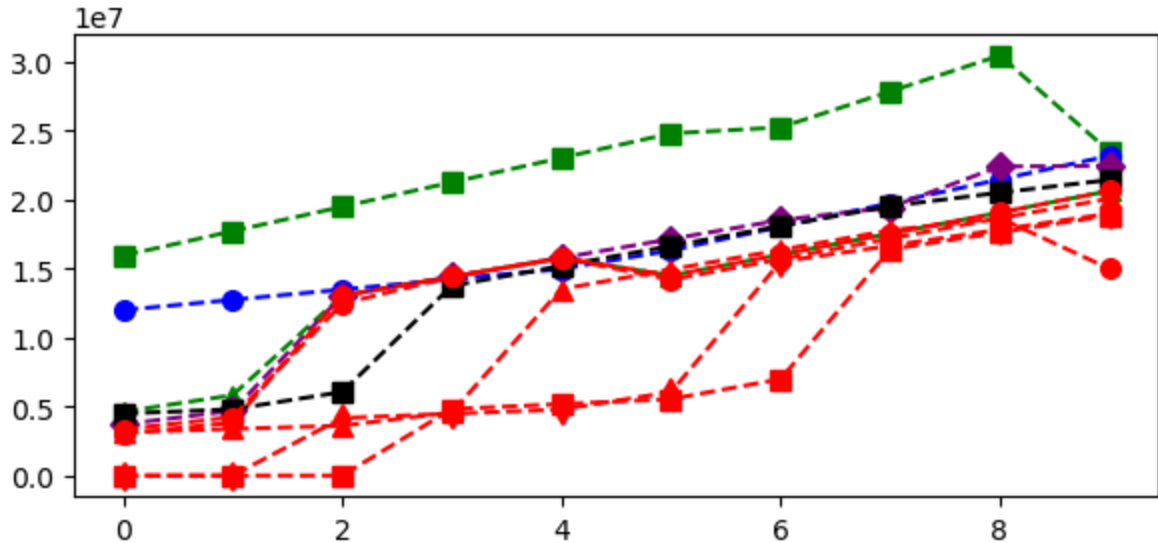
```

```

182 except KeyError:
--> 183     raise ValueError(
184         f"{v!r} is not a valid value for {k}; supported values are "
185         f"{', '.join(map(repr, mapping))}") from None

```

ValueError: 'lover right' is not a valid value for loc; supported values are 'best', 'upper right', 'upper left', 'lower left', 'lower right', 'right', 'center left', 'center right', 'lower center', 'upper center', 'center'



In [56]: # we can visualize the how many games played by a player

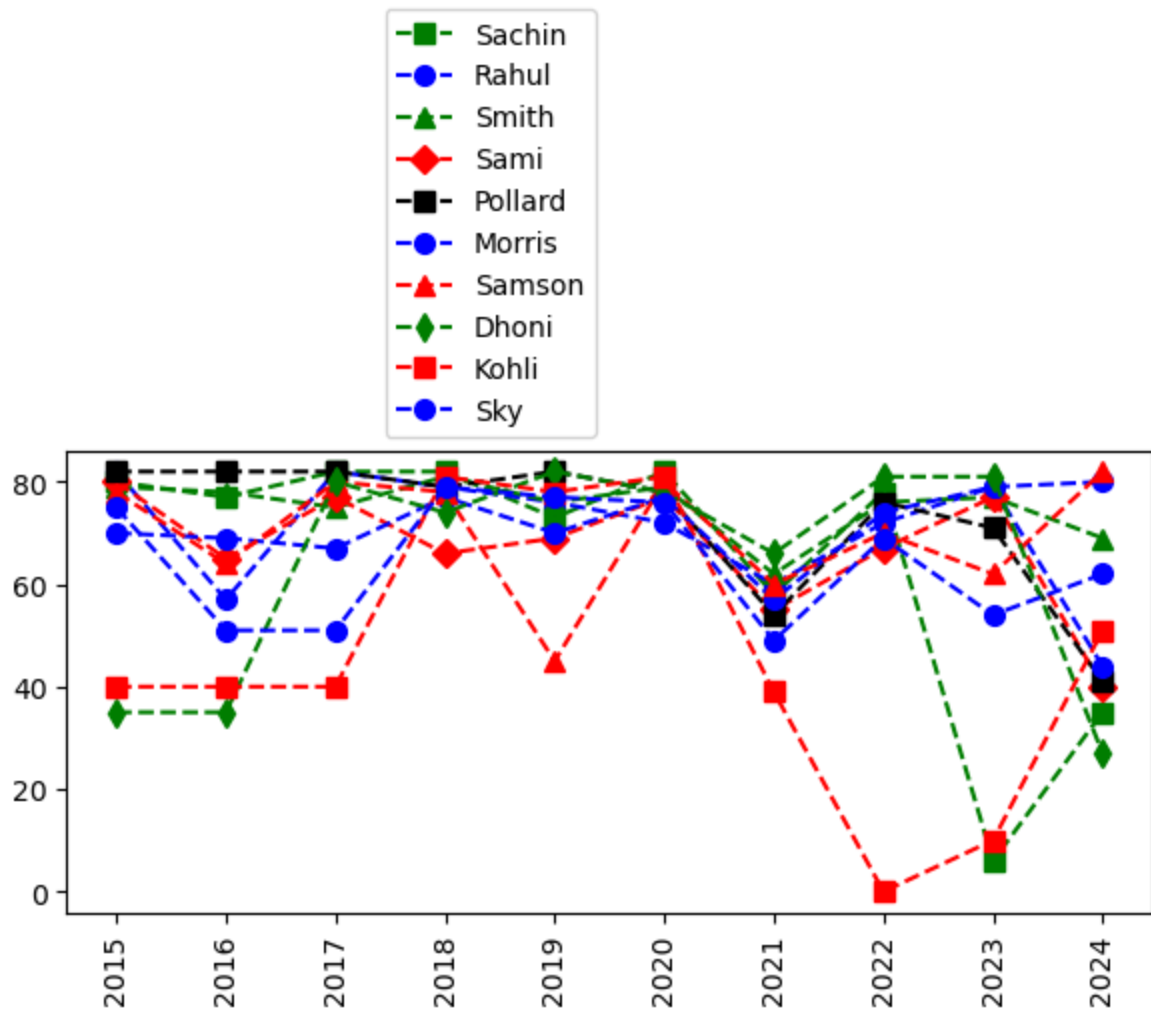
```

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()

```

In []: