

IPL match data analysis using numpy and matplotlib

In [2]:

```
#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}

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

#Salaries
Sachin_Salary = [15946875,17718750,19490625,21262500,23034375,24806250,25244493,26015000,27785000]
Rahul_Salary = [12000000,12744189,13488377,14232567,14976754,16324500,18038573,18800000,20570000]
Smith_Salary = [4621800,5828090,13041250,14410581,15779912,14500000,16022500,17500000,19270000]
Sami_Salary = [3713640,4694041,13041250,14410581,15779912,17149243,18518574,19450000,21220000,23000000]
Pollard_Salary = [4493160,4806720,6061274,13758000,15202590,16647180,18091770,19270000,21040000,22810000]
Morris_Salary = [3348000,4235220,12455000,14410581,15779912,14500000,16022500,17500000,19270000,21040000]
Samson_Salary = [3144240,3380160,3615960,4574189,13520500,14940153,16359805,17770000,19540000,21310000]
Dhoni_Salary = [0,0,4171200,4484040,4796880,6053663,15506632,16669630,17832627,18632688,20402500]
Kohli_Salary = [0,0,0,4822800,5184480,5546160,6993708,16402500,17632688,18862875,20632688]
Sky_Salary = [3031920,3841443,13041250,14410581,15779912,14200000,15691000,17182000,19052000,20822000]
#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])
```

In [3]:

Salary

```
Out[3]: 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 [4]: Games

```
Out[4]: 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 [5]: Points

```
Out[5]: 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]]))
```

In [6]: Sdict

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

In [7]: Pdict

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

In [8]: Games

```
Out[8]: 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 [9]: Games[5]

```
Out[9]: array([70, 69, 67, 77, 70, 77, 57, 74, 79, 44])
```

In [10]: Games[5,3]

```
Out[10]: np.int64(77)
```

In [11]: Salary

```
Out[11]: 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 [12]: Salary[0]

```
Out[12]: array([15946875, 17718750, 19490625, 21262500, 23034375, 24806250,  
                25244493, 27849149, 30453805, 23500000])
```

```
In [13]: Games[0]
```

```
Out[13]: array([80, 77, 82, 82, 73, 82, 58, 78, 6, 35])
```

```
In [14]: Salary[0]/Games[0]
```

```
Out[14]: array([ 199335.9375 , 230113.63636364, 237690.54878049,  
                259298.7804878 , 315539.38356164, 302515.24390244,  
                435249.87931034, 357040.37179487, 5075634.16666667,  
                671428.57142857])
```

```
In [15]: np.round(Salary[0]/Games[0])
```

```
Out[15]: array([ 199336., 230114., 237691., 259299., 315539., 302515.,  
                435250., 357040., 5075634., 671429.])
```

lets visualize the data

```
In [16]: import warnings  
warnings.filterwarnings('ignore')
```

```
#to ignore unwanted errors write the code as ignore all
```

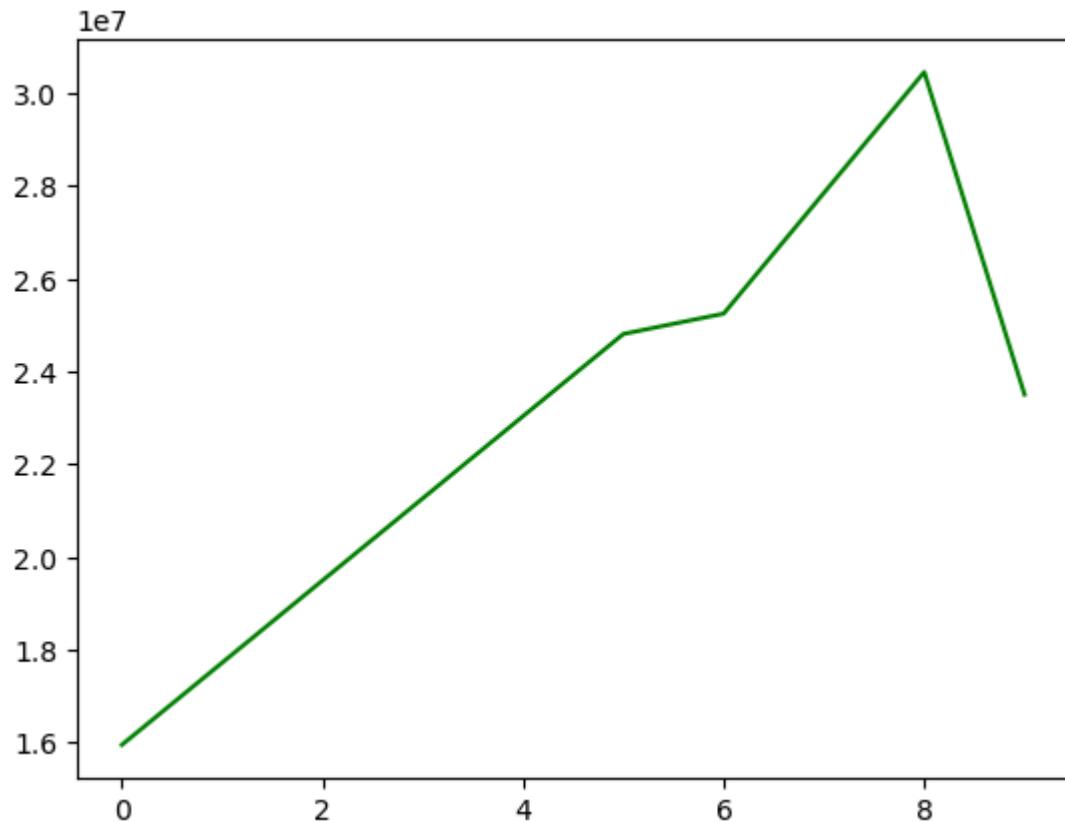
```
In [17]: import matplotlib.pyplot as plt
```

```
In [18]: Salary[0]
```

```
Out[18]: array([15946875, 17718750, 19490625, 21262500, 23034375, 24806250,  
                25244493, 27849149, 30453805, 23500000])
```

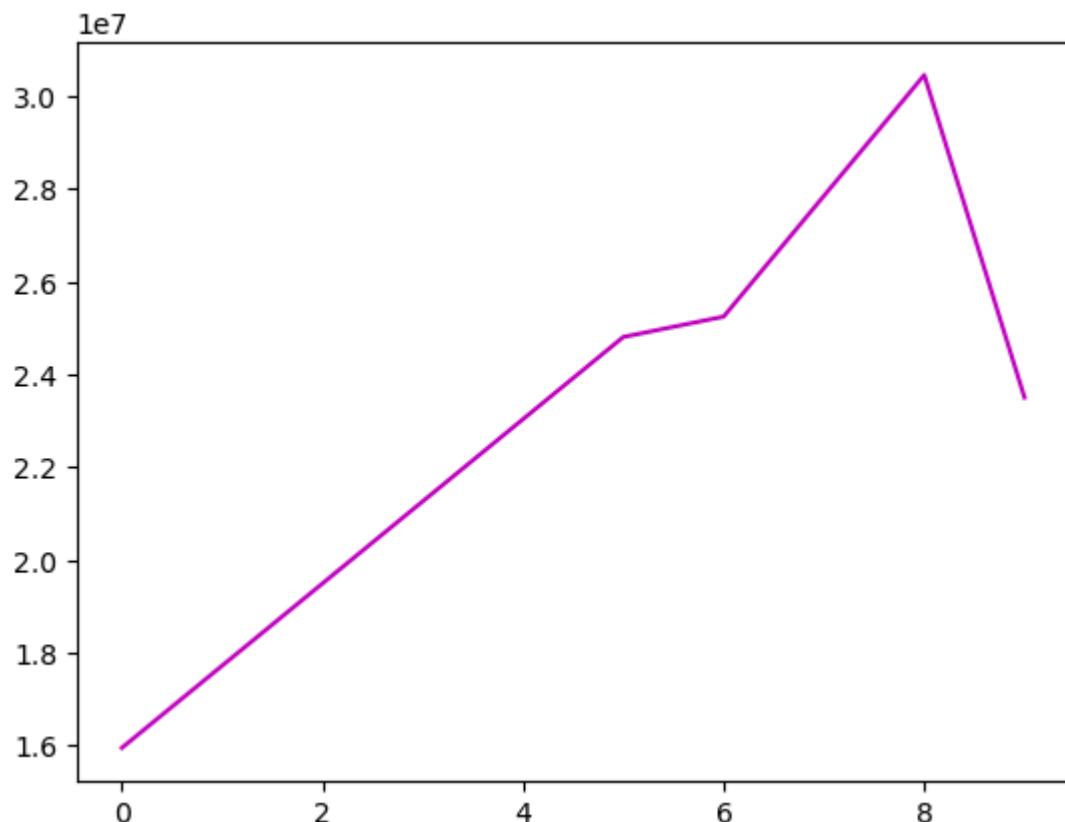
```
In [19]: plt.plot(Salary[0],c= 'green')
```

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



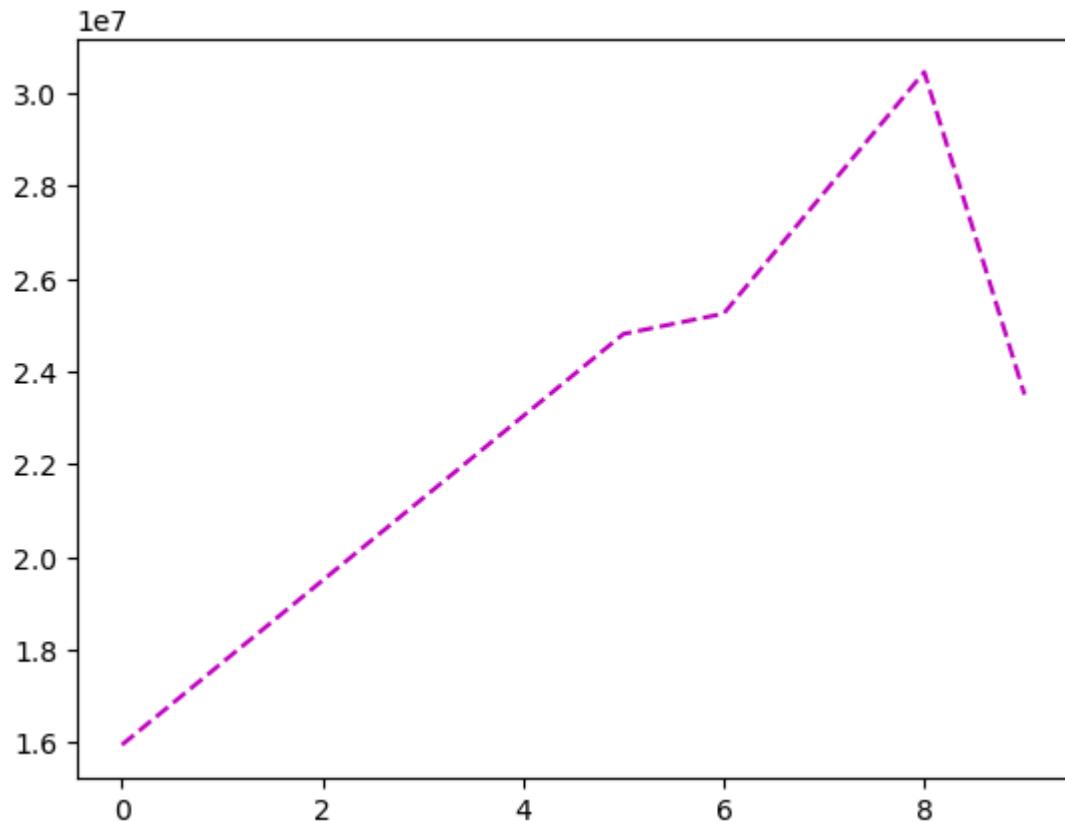
```
In [20]: plt.plot(Salary[0], c= 'm')
```

```
Out[20]: [
```



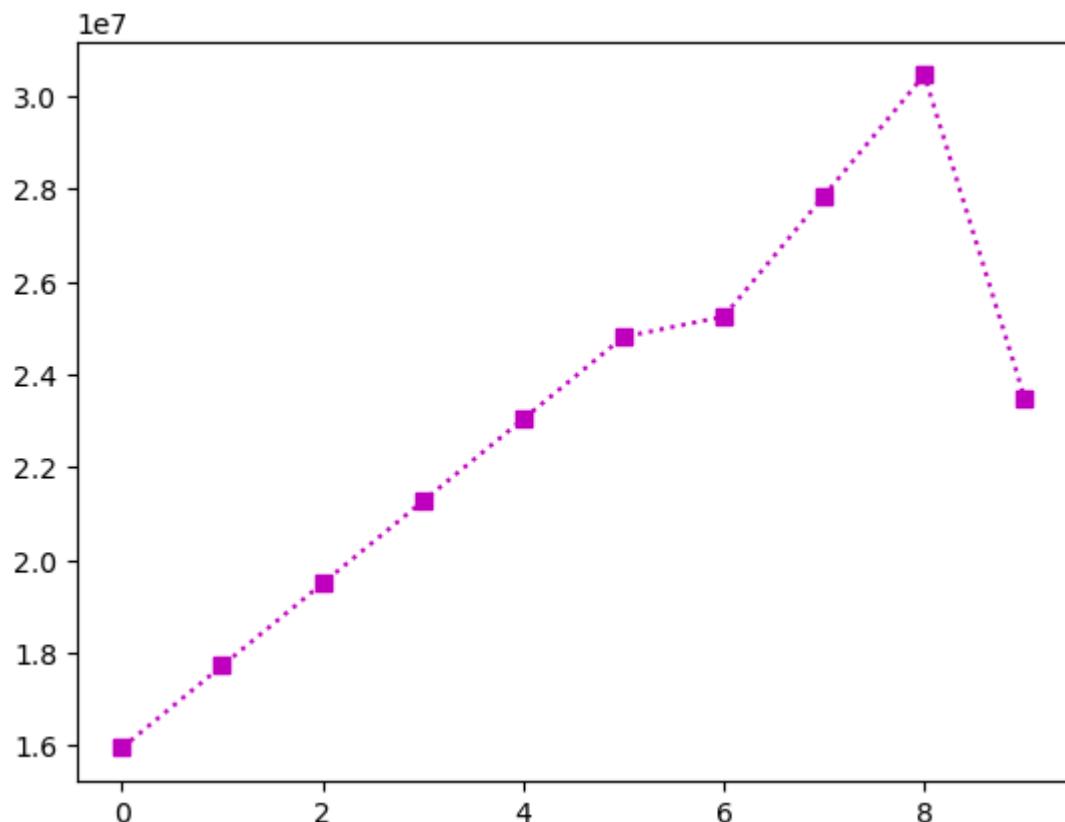
```
In [21]: plt.plot(Salary[0], c= 'm', ls = '--')
```

```
Out[21]: [
```



```
In [22]: plt.plot(Salary[0],c= 'm', ls = ':', marker = 's')
```

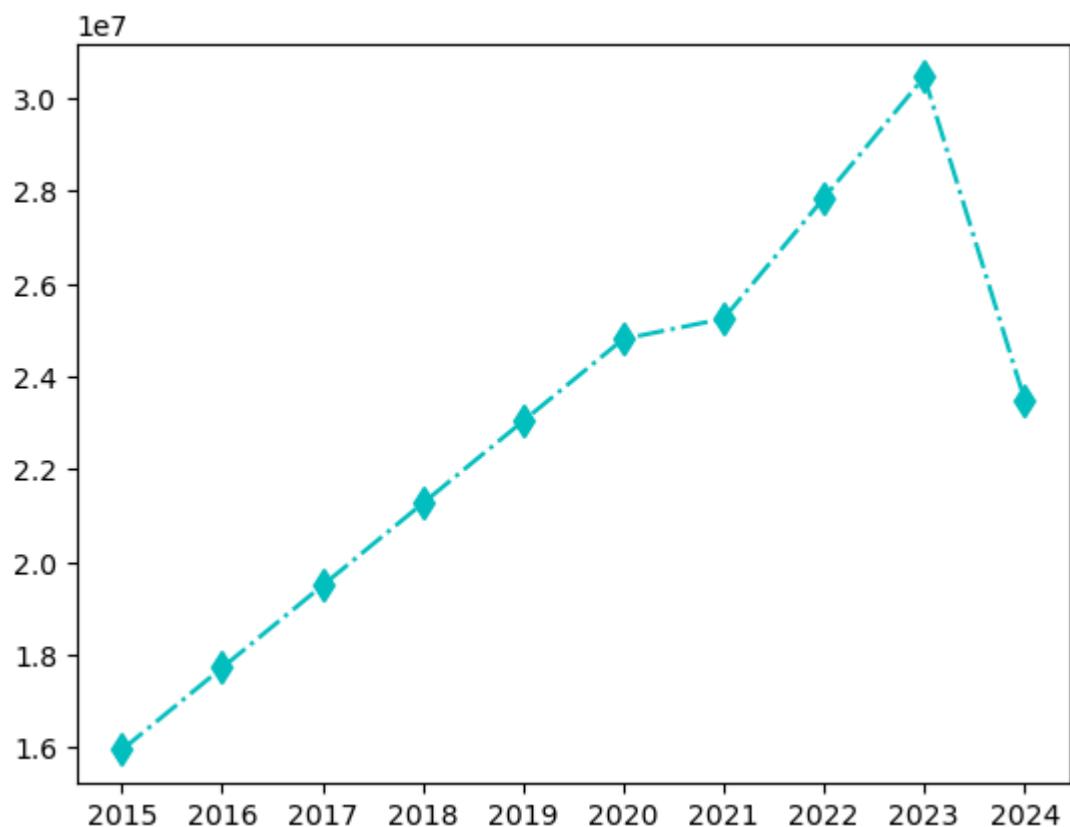
```
Out[22]: [<matplotlib.lines.Line2D at 0x25f6df91450>]
```



```
In [23]: Games
```

```
Out[23]: 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 [24]: plt.plot(Salary[0], c='c', ls='-.', marker='d', ms=8, label= Players[0])
plt.xticks(list(range(0,10)),Seasons, rotation= 'horizontal')
plt.show()
```



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

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

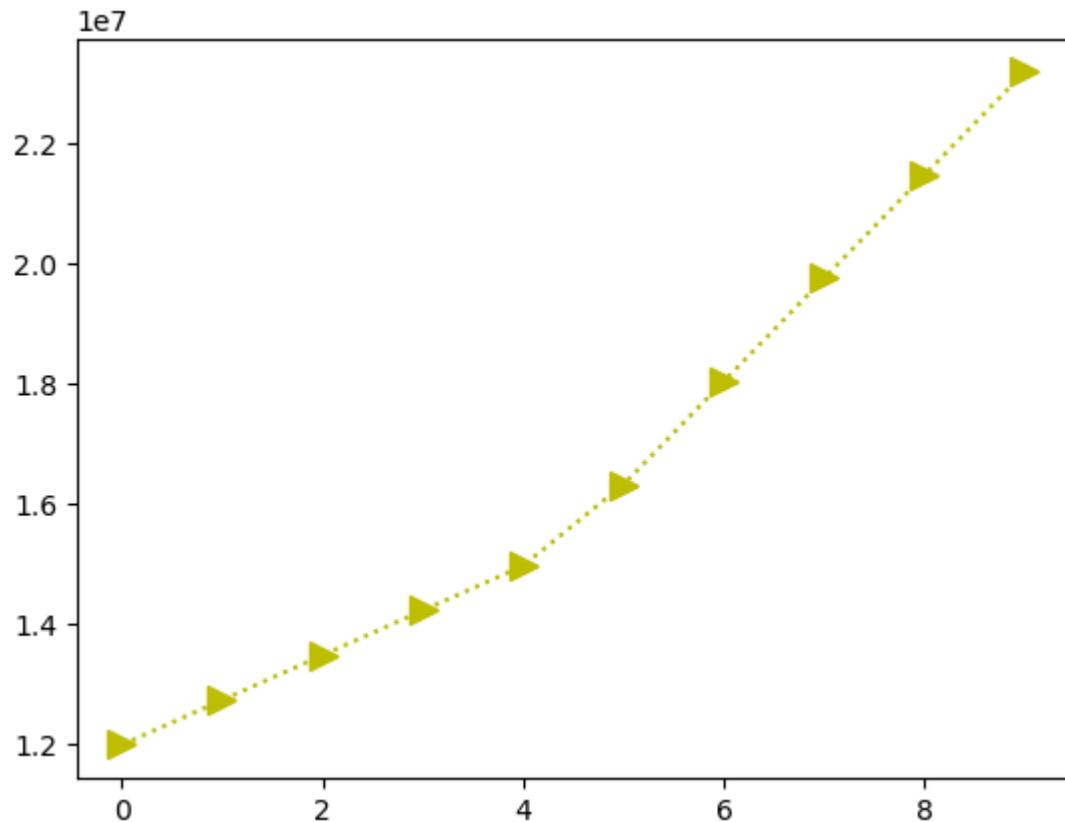
```
In [26]: Salary[1]
```

```
Out[26]: array([12000000, 12744189, 13488377, 14232567, 14976754, 16324500,
 18038573, 19752645, 21466718, 23180790])
```

```
In [27]: plt.plot(Salary[1], c='y', ls=':', marker='>', ms=10, label=Players[1])
```

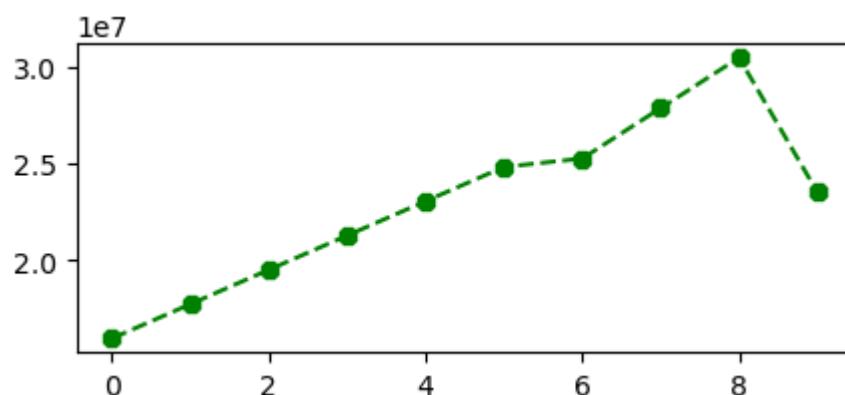
```
Out[27]: [

```

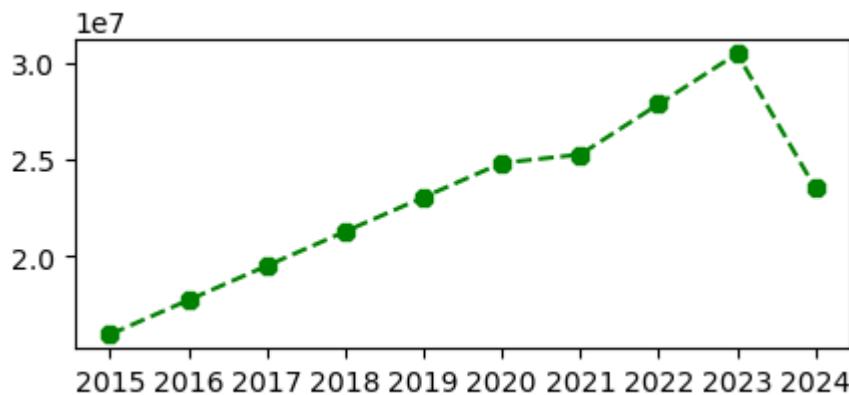


```
In [28]: %matplotlib inline  
plt.rcParams['figure.figsize'] = 5,2
```

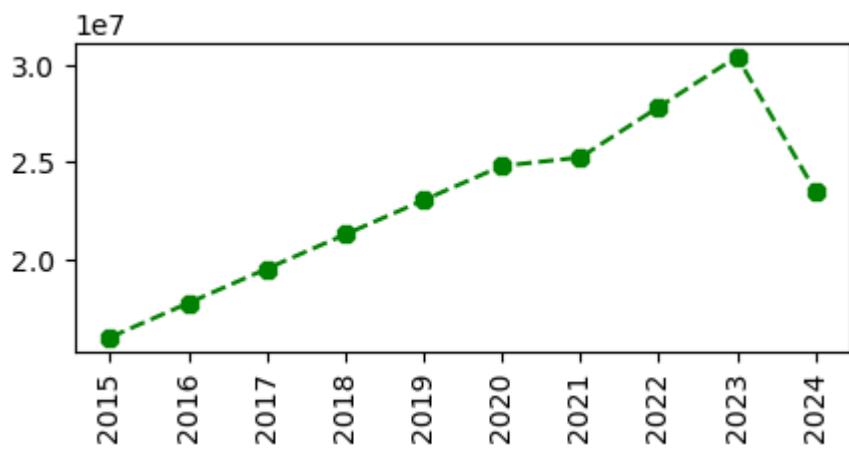
```
In [29]: plt.plot(Salary[0], ls='--', c='g', marker='8')  
plt.show()
```



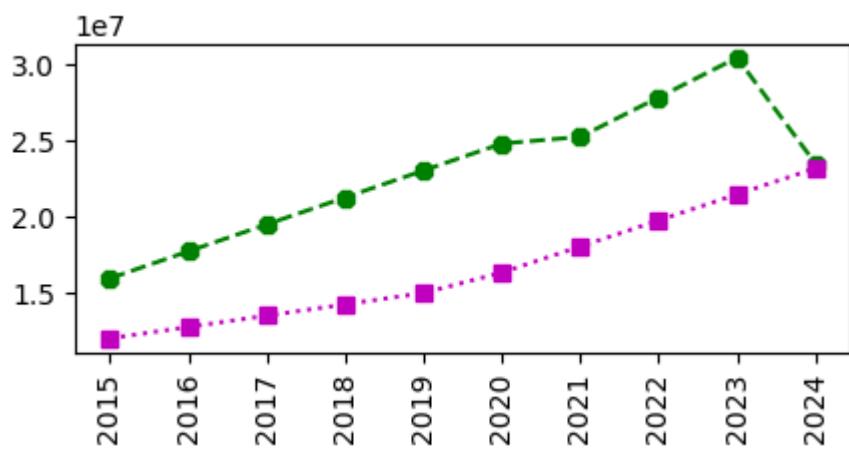
```
In [30]: %matplotlib inline  
plt.plot(Salary[0], ls='--', c='g', marker='8')  
plt.xticks(list(range(0,10)), Seasons)  
plt.show()
```



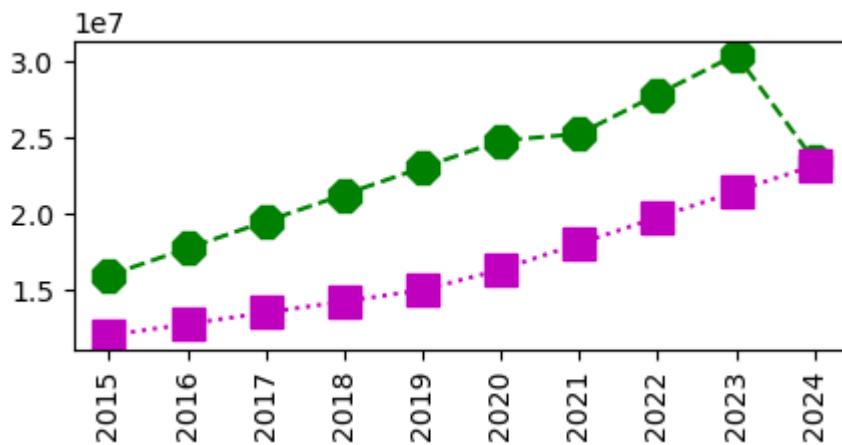
```
In [31]: %matplotlib inline
plt.plot(Salary[0], ls='--', c='g', marker='8')
plt.xticks(list(range(0,10)), Seasons, rotation = 'vertical')
plt.show()
```



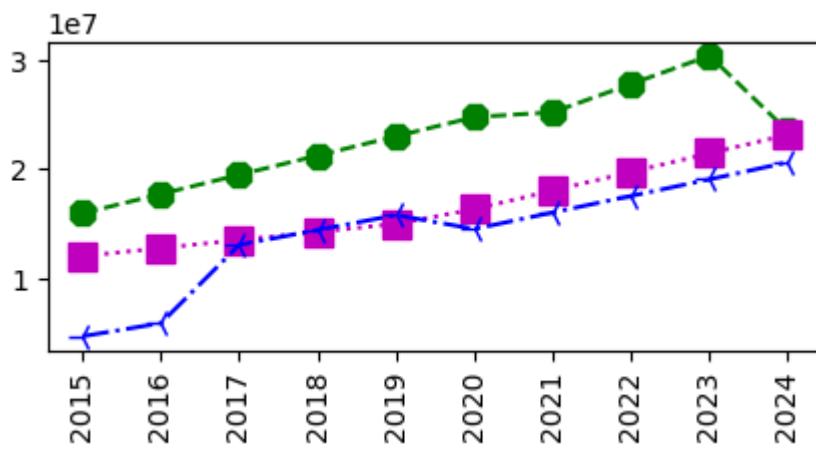
```
In [32]: plt.plot(Salary[0], ls= '--', c='g',marker='8')
plt.plot(Salary[1], ls=':', c='m',marker='s')
plt.xticks(list(range(0,10)), Seasons, rotation='vertical')
plt.show()
```



```
In [33]: plt.plot(Salary[0], ls= '--', c='g',marker='8', ms='12',label = Players[0])
plt.plot(Salary[1], ls=':', c='m',marker='s', ms='12', label = Players[2])
plt.xticks(list(range(0,10)), Seasons, rotation='vertical')
plt.show()
```

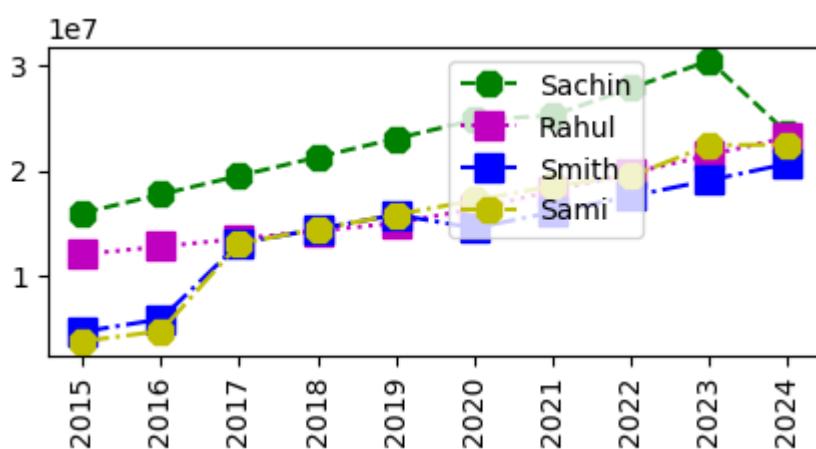


```
In [34]: plt.plot(Salary[0], ls='--', c='g', marker='8', ms='10', label = Players[0])
plt.plot(Salary[1], ls=':', c='m', marker='s', ms='10', label = Players[1])
plt.plot(Salary[2], ls='-.', c = 'b', marker='3', ms='10', label = Players[2])
plt.xticks(list(range(0,10)),Seasons,rotation='vertical')
plt.show()
```



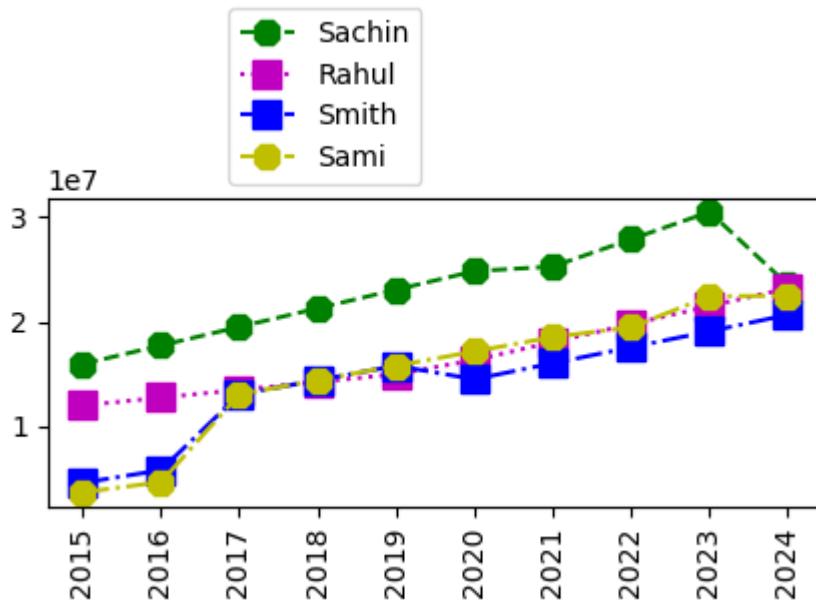
```
In [39]: plt.plot(Salary[0], ls='--', c='g', marker='8', ms='10', label = Players[0])
plt.plot(Salary[1], ls=':', c='m', marker='s', ms='10', label = Players[1])
plt.plot(Salary[2], ls='-.', c = 'b', marker='3', ms='10', label = Players[2])
plt.plot(Salary[3], ls='-.', c = 'y', marker='8', ms='10', label = Players[3])
plt.legend(loc = 'upper left',bbox_to_anchor=(0.5,1))

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



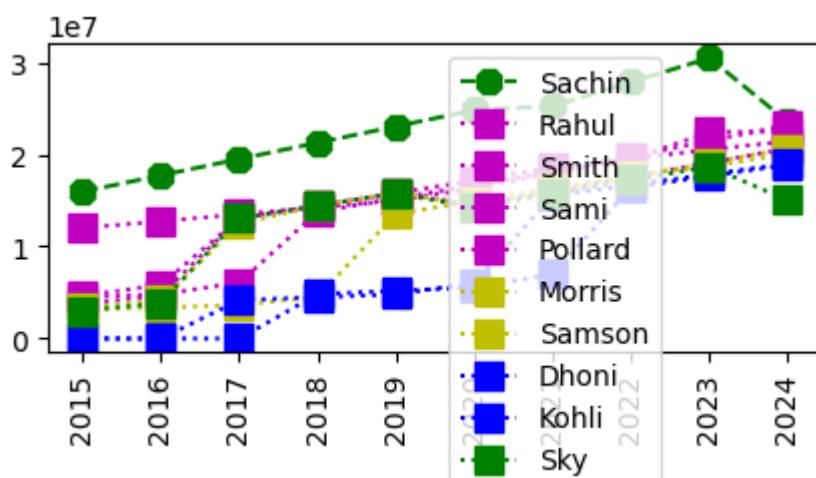
```
In [40]: plt.plot(Salary[0], ls= '--', c='g', marker='8', ms='10', label = Players[0])
plt.plot(Salary[1], ls=':', c='m', marker='s', ms='10', label = Players[1])
plt.plot(Salary[2], ls='-.', c = 'b', marker='s', ms='10', label = Players[2])
plt.plot(Salary[3], ls='-.', c = 'y', marker='8', ms='10', 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()
```



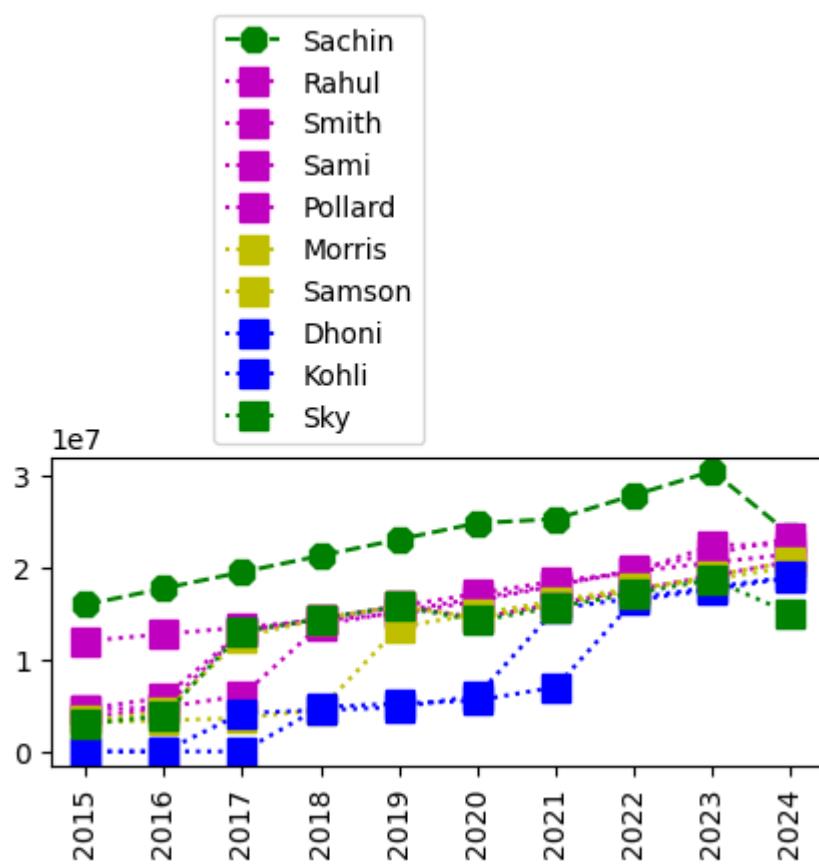
```
In [51]: plt.plot(Salary[0], ls= '--', c='g', marker='8', ms='10', label = Players[0])
plt.plot(Salary[1], ls=':', c='m', marker='s', ms='10', label = Players[1])
plt.plot(Salary[2], ls=':', c='m', marker='s', ms='10', label = Players[2])
plt.plot(Salary[3], ls=':', c='m', marker='s', ms='10', label = Players[3])
plt.plot(Salary[4], ls=':', c='m', marker='s', ms='10', label = Players[4])
plt.plot(Salary[5], ls=':', c='y', marker='s', ms='10', label = Players[5])
plt.plot(Salary[6], ls=':', c='y', marker='s', ms='10', label = Players[6])
plt.plot(Salary[7], ls=':', c='b', marker='s', ms='10', label = Players[7])
plt.plot(Salary[8], ls=':', c='b', marker='s', ms='10', label = Players[8])
plt.plot(Salary[9], ls=':', c='g', marker='s', ms='10', label = Players[9])
plt.legend(loc = 'upper left',bbox_to_anchor=(0.5,1))

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



```
In [47]: plt.plot(Salary[0], ls='--', c='g', marker='8', ms=10, label = Players[0])
plt.plot(Salary[1], ls=':', c='m', marker='s', ms=10, label = Players[1])
plt.plot(Salary[2], ls=':', c='m', marker='s', ms=10, label = Players[2])
plt.plot(Salary[3], ls=':', c='m', marker='s', ms=10, label = Players[3])
plt.plot(Salary[4], ls=':', c='m', marker='s', ms=10, label = Players[4])
plt.plot(Salary[5], ls=':', c='y', marker='s', ms=10, label = Players[5])
plt.plot(Salary[6], ls=':', c='y', marker='s', ms=10, label = Players[6])
plt.plot(Salary[7], ls=':', c='b', marker='s', ms=10, label = Players[7])
plt.plot(Salary[8], ls=':', c='b', marker='s', ms=10, label = Players[8])
plt.plot(Salary[9], ls=':', c='g', marker='s', ms=10, 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 []:

In []:

In []:

In []: