

IPL data analysis project with NUMPY + MATPLOTLIB (Matrix Visualization)

MATRICES / NUMPY ----- Matrix is the tabular representation of the data

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

```
In [2]: # Seasons
Seasons = ["2010", "2011", "2012", "2013", "2014", "2015", "2016", "2017", "2018", "2019"]
Sdict = {"2010":0, "2011":1, "2012":2, "2013":3, "2014":4, "2015":5, "2016":6, "2017":7, "2018":8, "2019":9}
```

```
In [3]: # 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}
```

```
In [4]: # Salaries
Sachin_Salary = [15946875, 17718750, 19490625, 21262500, 23034375, 24806250, 25244490, 26000000, 27000000, 28000000]
Rahul_Salary = [12000000, 12744189, 13488377, 14232567, 14976754, 16324500, 18038573, 19000000, 20000000, 21000000]
Smith_Salary = [4621800, 5828090, 13041250, 14410581, 15779912, 14500000, 16022500, 17000000, 18000000, 19000000]
Sami_Salary = [3713640, 4694041, 13041250, 14410581, 15779912, 17149243, 18518574, 19000000, 20000000, 21000000]
Pollard_Salary = [4493160, 4806720, 6061274, 13758000, 15202590, 16647180, 18091770, 19000000, 20000000, 21000000]
Morris_Salary = [3348000, 4235220, 12455000, 14410581, 15779912, 14500000, 16022500, 17000000, 18000000, 19000000]
Samson_Salary = [3144240, 3380160, 3615960, 4574189, 13520500, 14940153, 16359805, 17000000, 18000000, 19000000]
Dhoni_Salary = [0, 0, 4171200, 4484040, 4796880, 6053663, 15506632, 16669630, 17832627, 19000000]
Kohli_Salary = [0, 0, 0, 4822800, 5184480, 5546160, 6993708, 16402500, 17632688, 18862800]
Sky_Salary = [3031920, 3841443, 13041250, 14410581, 15779912, 14200000, 15691000, 17100000, 18000000, 19000000]
```

```
In [5]: #Matrix
Salary = np.array([Sachin_Salary, Rahul_Salary, Smith_Salary, Sami_Salary, Pollard_Salary, Morris_Salary, Samson_Salary, Dhoni_Salary, Kohli_Salary, Sky_Salary])
```

```
Out[5]: 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 [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, Dhoni_G, Kohli_G, Sky_G])
```

```
Out[7]: 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 [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, Mc
Points
```

```
Out[9]: 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 [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
```

```
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, 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 [21]: mydata = np.arange(0, 20)
         print(mydata)
```

```
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19]
```

```
In [22]: mydata
```

```
Out[22]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
                17, 18, 19])
```

```
In [23]: np.reshape(mydata) # when we will use reshape for our data we will have to de
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[23], line 1
----> 1 np.reshape(mydata)

File <__array_function__ internals>:179, in reshape(*args, **kwargs)

TypeError: _reshape_dispatcher() missing 1 required positional argument: 'newshape'
```

```
In [24]: np.reshape(mydata, (2, 10))
```

```
Out[24]: array([[ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9],
                [10, 11, 12, 13, 14, 15, 16, 17, 18, 19]])
```

```
In [25]: np.reshape(mydata, (4, 5))
```

```
Out[25]: array([[ 0,  1,  2,  3,  4],
               [ 5,  6,  7,  8,  9],
               [10, 11, 12, 13, 14],
               [15, 16, 17, 18, 19]])
```

```
In [26]: mydata
```

```
Out[26]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
               17, 18, 19])
```

```
In [27]: # np.reshape(mydata,(5,4), order = 'c')
         # 'C' means to read / write the elements using C-like index order
         mat1 = np.reshape(mydata, (4, 5), order='c')
         mat1
```

```
Out[27]: array([[ 0,  1,  2,  3,  4],
               [ 5,  6,  7,  8,  9],
               [10, 11, 12, 13, 14],
               [15, 16, 17, 18, 19]])
```

```
In [28]: mat1 = np.reshape(mydata, (4, 5), order='a') # c and a both are same
         mat1
```

```
Out[28]: array([[ 0,  1,  2,  3,  4],
               [ 5,  6,  7,  8,  9],
               [10, 11, 12, 13, 14],
               [15, 16, 17, 18, 19]])
```

```
In [29]: # if I want to get a specific number (3) in our matrix
         mat1[0, 3]
```

```
Out[29]: 3
```

```
In [30]: # if I want to get number 9 with positive indexing
         mat1[1, 4]
```

```
Out[30]: 9
```

```
In [31]: # if I want to get number 9 with negative indexing
         mat1[-3, -1]
```

```
Out[31]: 9
```

```
In [32]: mat1[0:3]
```

```
Out[32]: array([[ 0,  1,  2,  3,  4],
               [ 5,  6,  7,  8,  9],
               [10, 11, 12, 13, 14]])
```

```
In [33]: mat2 = np.reshape(mydata, (4, 5), order='f')
mat2
```

```
Out[33]: array([[ 0,  4,  8, 12, 16],
                [ 1,  5,  9, 13, 17],
                [ 2,  6, 10, 14, 18],
                [ 3,  7, 11, 15, 19]])
```

```
In [34]: # if I want to get a specific number (3) in our matrix
mat2[3, 0]
```

```
Out[34]: 3
```

```
In [35]: # if I want to get number 9 with positive indexing
mat2[1, 2]
```

```
Out[35]: 9
```

```
In [36]: # if I want to get number 9 with positive indexing
mat2[-3, -3]
```

```
Out[36]: 9
```

```
In [37]: mat2[0:3]
```

```
Out[37]: array([[ 0,  4,  8, 12, 16],
                [ 1,  5,  9, 13, 17],
                [ 2,  6, 10, 14, 18]])
```

```
In [38]: a1 = ['welcome', 'to', 'datascience']
a2 = ['required', 'data', 'science']
a3= [1, 2, 3]
[a1, a2, a3]
```

```
Out[38]: [['welcome', 'to', 'datascience'], ['required', 'data', 'science'], [1, 2,
3]]
```

```
In [39]: np.array([a1, a2, a3])
```

```
Out[39]: array(['welcome', 'to', 'datascience'],
                ['required', 'data', 'science'],
                ['1', '2', '3']], dtype='<U11')
```

```
In [40]: Games
```

```
Out[40]: 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 [41]: Games[0]
```

```
Out[41]: array([80, 77, 82, 82, 73, 82, 58, 78,  6, 35])
```

```
In [42]: Games[5]
```

```
Out[42]: array([70, 69, 67, 77, 70, 77, 57, 74, 79, 44])
```

```
In [43]: Games[0:5]
```

```
Out[43]: 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 [44]: Games[0, 5]
```

```
Out[44]: 82
```

```
In [45]: Games[2]
```

```
Out[45]: array([79, 78, 75, 81, 76, 79, 62, 76, 77, 69])
```

```
In [46]: Games[-5:-2]
```

```
Out[46]: array([[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]])
```

```
In [47]: Points
```

```
Out[47]: 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 [48]: Points[0]
```

```
Out[48]: array([2832, 2430, 2323, 2201, 1970, 2078, 1616, 2133,  83, 782])
```

```
In [49]: Points[6, 1]
```

```
Out[49]: 1104
```

```
In [50]: Points[6, 1]
```

```
Out[50]: 1104
```

```
In [51]: Points[3:6]
```

```
Out[51]: array([[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]])
```

```
In [52]: Points[3:6, 5:7]
```

```
Out[52]: array([[1970, 1245],
               [1784, 1113],
               [1438, 1025]])
```

```
In [53]: Points[-6, -1]
```

```
Out[53]: 646
```

```
In [54]: # Dictionary
dict1 = {'key1': 'val1', 'key2': 'val2', 'key3': 'val3'}
dict1
```

```
Out[54]: {'key1': 'val1', 'key2': 'val2', 'key3': 'val3'}
```

main advantage of the dictionary is we don't required to count which no. of row which players are sitting

```
In [55]: dict1['key2']
```

```
Out[55]: 'val2'
```

```
In [56]: Games
```

```
Out[56]: 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 [57]: Pdict
```

```
Out[57]: {'Sachin': 0,
          'Rahul': 1,
          'Smith': 2,
          'Sami': 3,
          'Pollard': 4,
          'Morris': 5,
          'Samson': 6,
          'Dhoni': 7,
          'Kohli': 8,
          'Sky': 9}
```



```
In [58]: Sdict
```

```
Out[58]: {'2010': 0,  
          '2011': 1,  
          '2012': 2,  
          '2013': 3,  
          '2014': 4,  
          '2015': 5,  
          '2016': 6,  
          '2017': 7,  
          '2018': 8,  
          '2019': 9}
```

```
In [59]: Games[1]
```

```
Out[59]: array([82, 57, 82, 79, 76, 72, 60, 72, 79, 80])
```

```
In [60]: Pdict['Rahul']
```

```
Out[60]: 1
```

```
In [61]: """  
          Pdict['Rahul'] = 1  
          Games[1] = array([82, 57, 82, 79, 76, 72, 60, 72, 79, 80])  
          """  
          Games[Pdict['Rahul']]
```

```
Out[61]: array([82, 57, 82, 79, 76, 72, 60, 72, 79, 80])
```

```
In [62]: Games[Sdict['2011']]
```

```
Out[62]: array([82, 57, 82, 79, 76, 72, 60, 72, 79, 80])
```

```
In [63]: Games[0]
```

```
Out[63]: array([80, 77, 82, 82, 73, 82, 58, 78, 6, 35])
```

```
In [64]: Games[Pdict['Sachin']]
```

```
Out[64]: array([80, 77, 82, 82, 73, 82, 58, 78, 6, 35])
```

```
In [65]: Games[Sdict['2010']]
```

```
Out[65]: array([80, 77, 82, 82, 73, 82, 58, 78, 6, 35])
```

```
In [66]: Salary
```

```
Out[66]: 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 [67]: Salary[2, 4]
```

```
Out[67]: 15779912
```

```
In [68]: # Pdict['Sky'] == 1
         # Salary[1]
         Salary[Pdict['Sky']]
```

```
Out[68]: array([ 3031920,  3841443, 13041250, 14410581, 15779912, 14200000,
                15691000, 17182000, 18673000, 15000000])
```

```
In [69]: # Sdict['2019'] == 1
         # Salary[1]
         Salary[Sdict['2019']]
```

```
Out[69]: array([ 3031920,  3841443, 13041250, 14410581, 15779912, 14200000,
                15691000, 17182000, 18673000, 15000000])
```

```
In [70]: Salary[Pdict['Sky']][Sdict['2019']] # nested list
         # Salary[9, 9] and Salary[Pdict['Sky']][Sdict['2019']] both are same thing
```

```
Out[70]: 15000000
```

```
In [71]: Salary[Pdict['Sky']][8]
```

```
Out[71]: 18673000
```

```
In [72]: Salary[9][9]
```

```
Out[72]: 15000000
```

```
In [73]: Pdict['Sky']
```

```
Out[73]: 9
```

```
In [74]: Sdict['2019']
```

```
Out[74]: 9
```

```
In [75]: Salary[9,9]
```

```
Out[75]: 15000000
```

```
In [76]: Salary[Pdict['Sami']][Sdict['2013']]
```

```
Out[76]: 14410581
```

```
In [77]: Salary
```

```
Out[77]: 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 [78]: Games
```

```
Out[78]: 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 [79]: Salary/Games

C:\Users\Hp\AppData\Local\Temp\ipykernel_15272\3709746658.py:1: RuntimeWarning: divide by zero encountered in divide
Salary/Games

```
Out[79]: 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 [80]: np.round(Salary/Games)
```

```
C:\Users\Hp\AppData\Local\Temp\ipykernel_15272\3232172828.py:1: RuntimeWarning: divide by zero encountered in divide
  np.round(Salary/Games)
```

```
Out[80]: 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 [81]: import warnings
warnings.filterwarnings('ignore')
```

```
In [82]: np.round(Salary/Games) # after importing warnings we are not getting warning
```

```
Out[82]: 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 [83]: import numpy as np
import matplotlib.pyplot as plt
```

```
In [84]: # %matplotlib inline this will keep the plot inside jupyter notes insted of g
%matplotlib inline
Salary
```

```
Out[84]: 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 [85]: Salary[0]
```

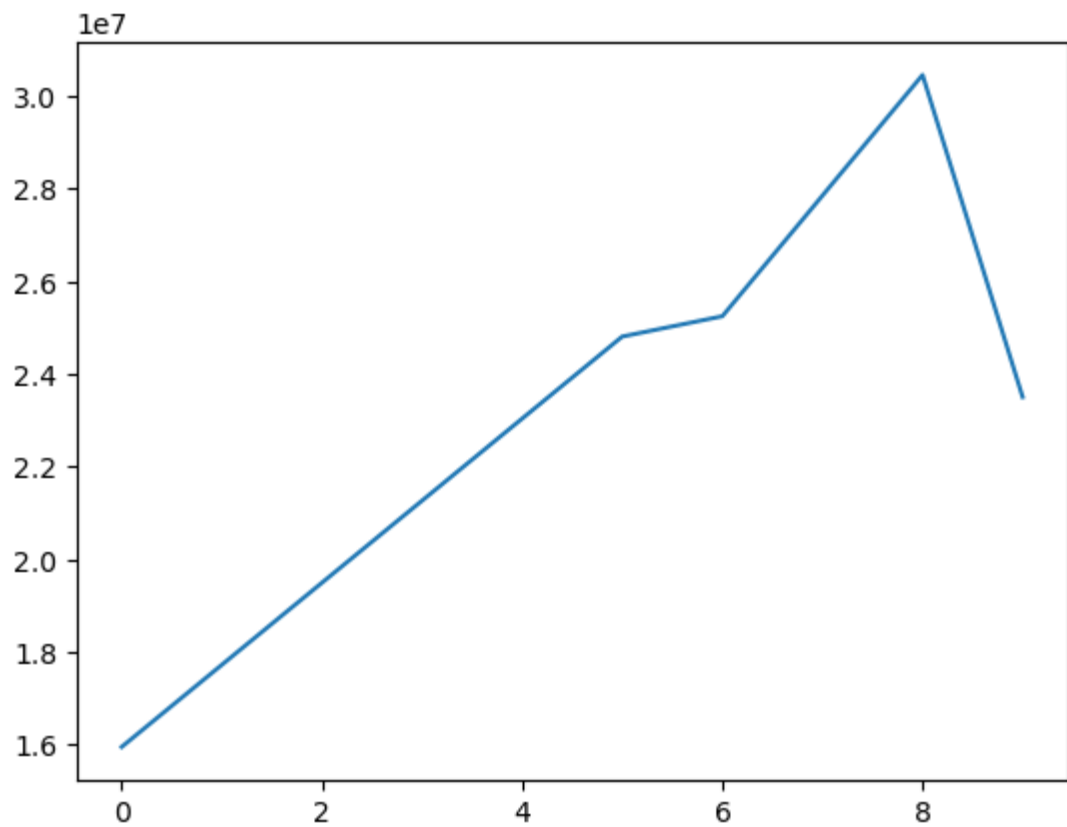
```
Out[85]: array([15946875, 17718750, 19490625, 21262500, 23034375, 24806250,
                25244493, 27849149, 30453805, 23500000])
```

here x axis is season and y axis is salary

- 1.6 means 160 or 15946875, 1.8 means 180 or 17718750, 2.0 means 200 or 19490625, and so on.....
- 15946875, 17718750, 19490625, 21262500, 23034375, 24806250, 25244493, 27849149, 30453805, 23500000.....

```
In [ ]: plt.plot(Salary[0])
```

```
Out[86]: [<matplotlib.lines.Line2D at 0x2540c6f1390>]
```

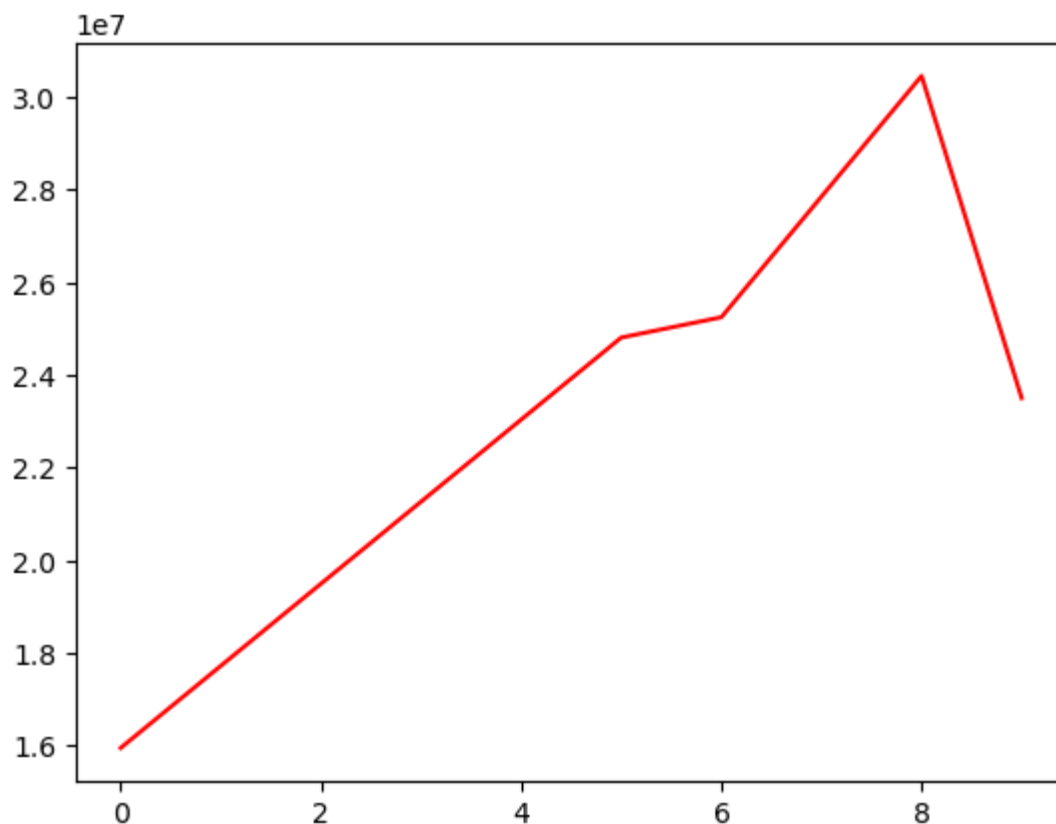


Interpretation Insight

- it keep on increasing till 7th seasons and then it is decline
- salary delcined after 7th season
- salary keep on increasing till 7th and then it increasæd.
- the player is consistence performance
- max salary is at 7

```
In [92]: plt.plot(Salary[0], c='Red')  
# plt.plot(Salary[0], c='g')  
# plt.plot(Salary[0], c='m')
```

```
Out[92]: [<matplotlib.lines.Line2D at 0x2540e584d00>]
```



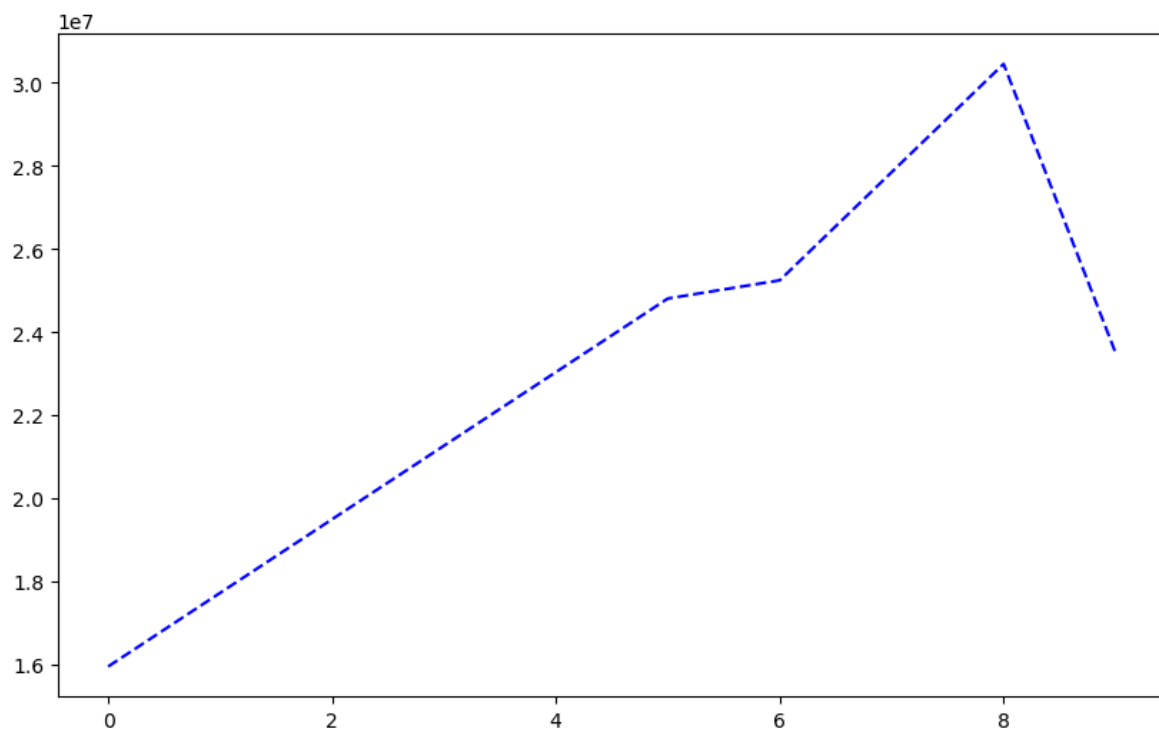
In Matplotlib, rcParams (short for "runtime configuration parameters") is a dictionary-like object that holds various configuration settings for customizing the appearance of plots and figures created using Matplotlib. These settings include things like line styles, colors, fonts, and other visual properties.

```
In [100]: %matplotlib inline  
plt.rcParams['figure.figsize'] = 10, 6
```



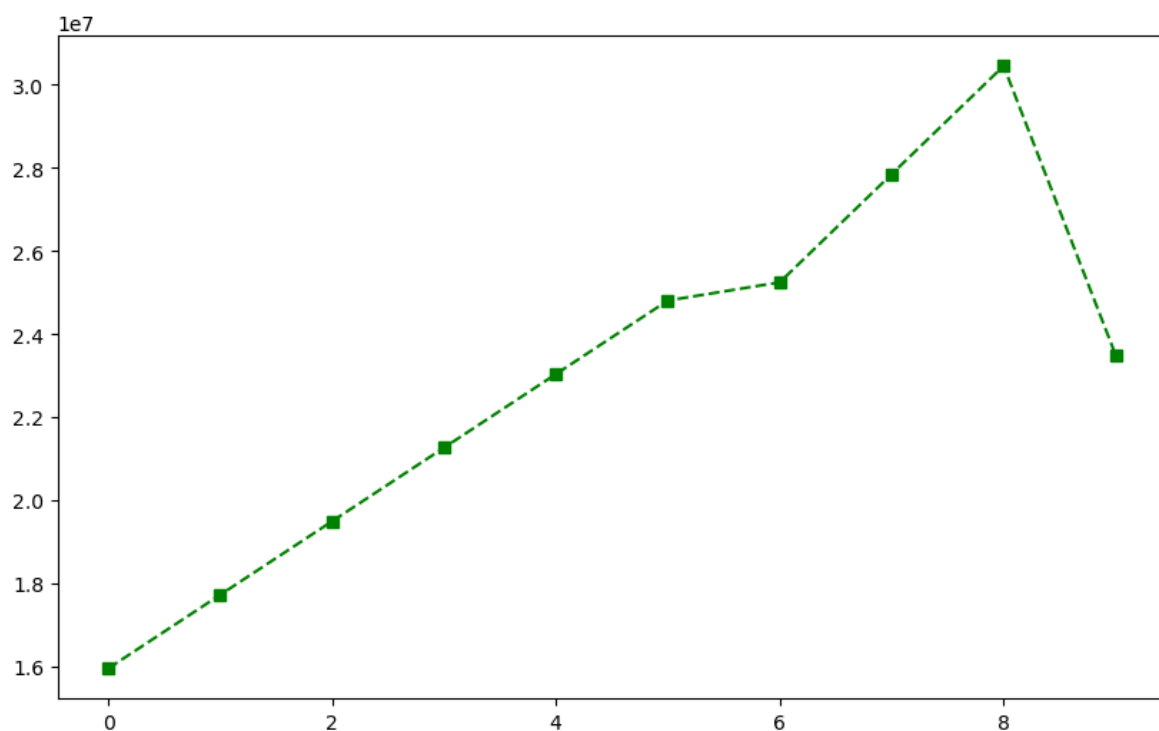
```
In [102]: plt.plot(Salary[0], c='b', ls='--')  
# plt.plot(Salary[0], c='Blue', ls='dashed')
```

Out[102]: [



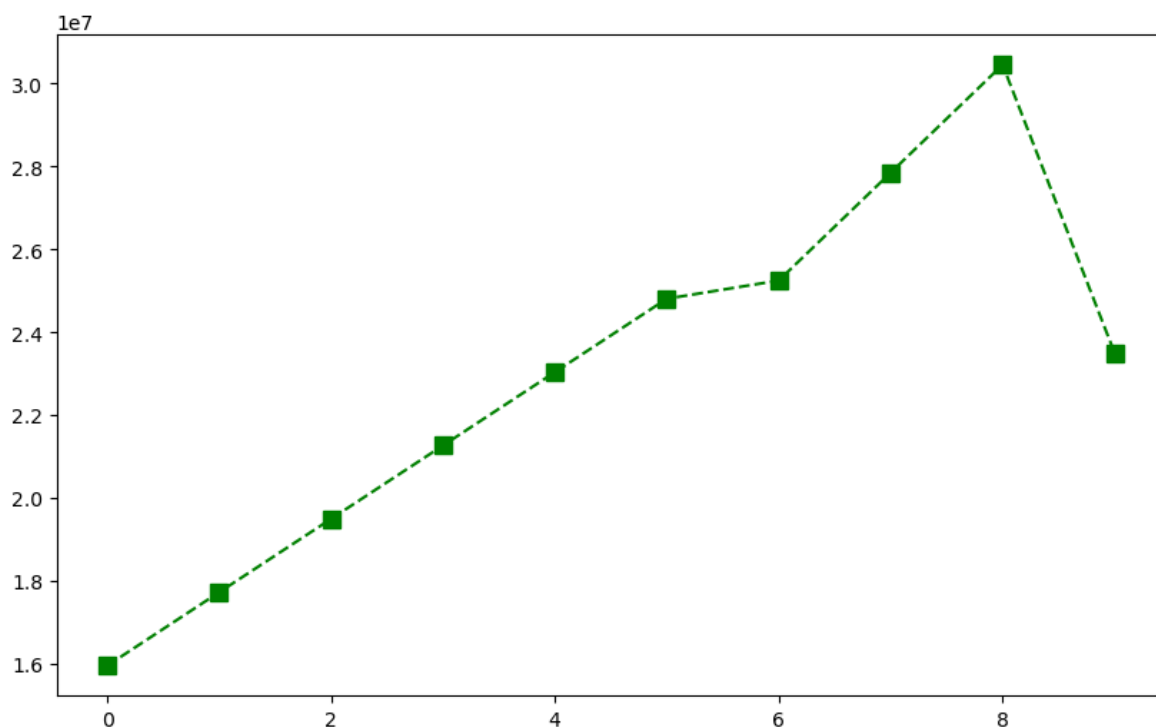
```
In [103]: plt.plot(Salary[0], c = 'g', ls= '--', marker='s') # s= square marker
```

Out[103]: [



```
In [111]: %matplotlib inline  
plt.rcParams['figure.figsize'] = 10, 6
```

```
In [113]: plt.plot(Salary[0], c = 'green', ls = '--', marker = 's', ms = 8) # ms = micro
plt.show()
```



```
In [106]: list(range(0, 10))
```

```
Out[106]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

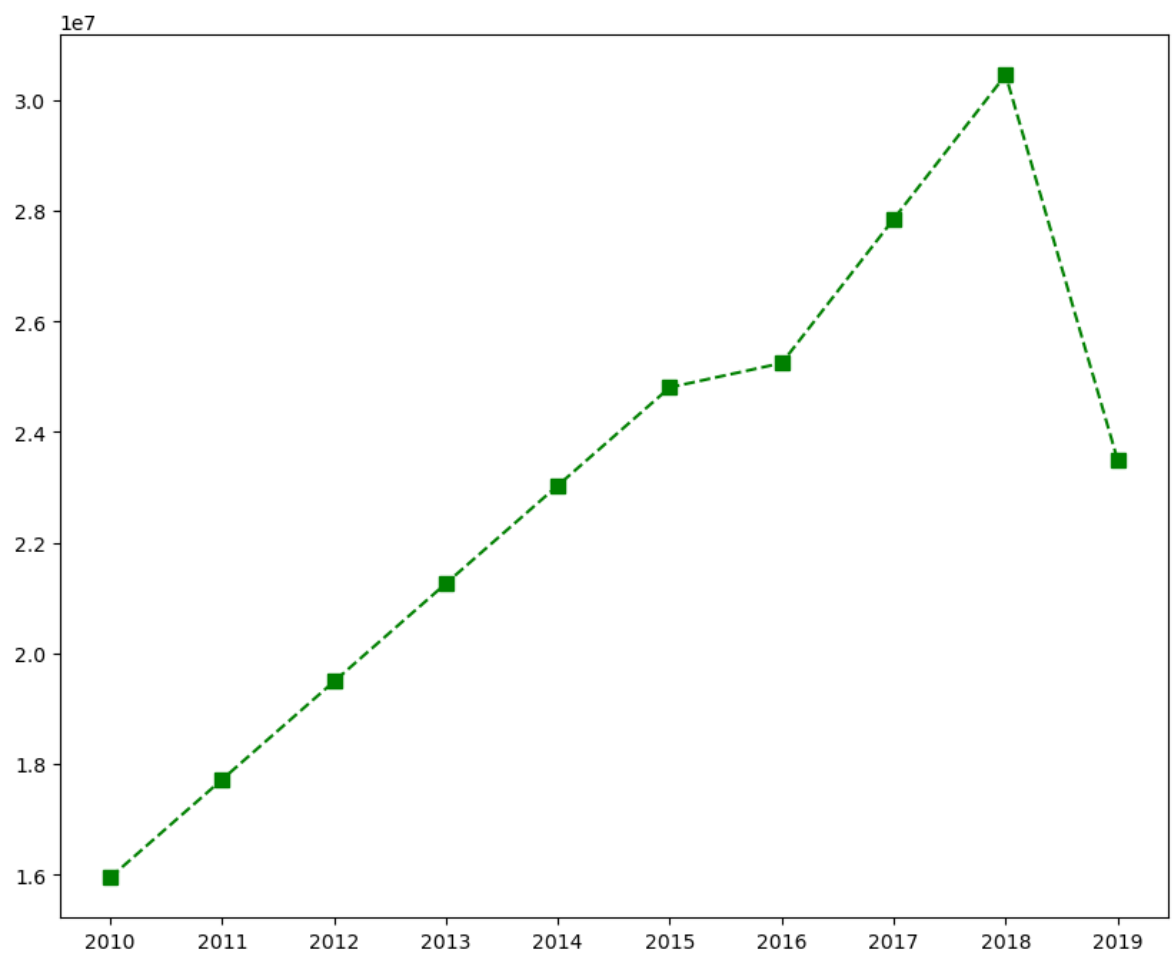
```
In [107]: Sdict
```

```
Out[107]: {'2010': 0,
            '2011': 1,
            '2012': 2,
            '2013': 3,
            '2014': 4,
            '2015': 5,
            '2016': 6,
            '2017': 7,
            '2018': 8,
            '2019': 9}
```

```
In [108]: Pdict
```

```
Out[108]: {'Sachin': 0,
            'Rahul': 1,
            'Smith': 2,
            'Sami': 3,
            'Pollard': 4,
            'Morris': 5,
            'Samson': 6,
            'Dhoni': 7,
            'Kohli': 8,
            'Sky': 9}
```

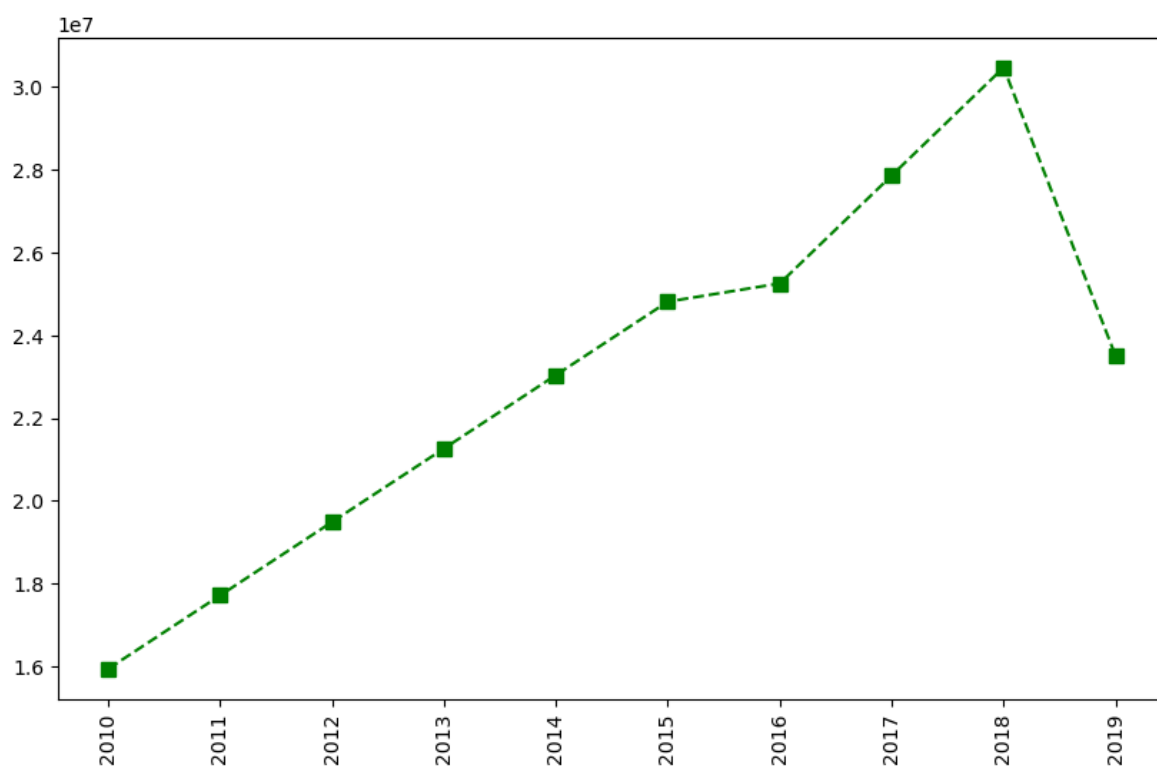
```
In [110]: plt.plot(Salary[0], c = 'green', ls = '--', marker = 's', ms = 7)
plt.xticks(list(range(0, 10)), Seasons)
plt.show()
```



Interpretation Insight

- salary of players keep increasing till 2018 and after that salary decrease at 2019

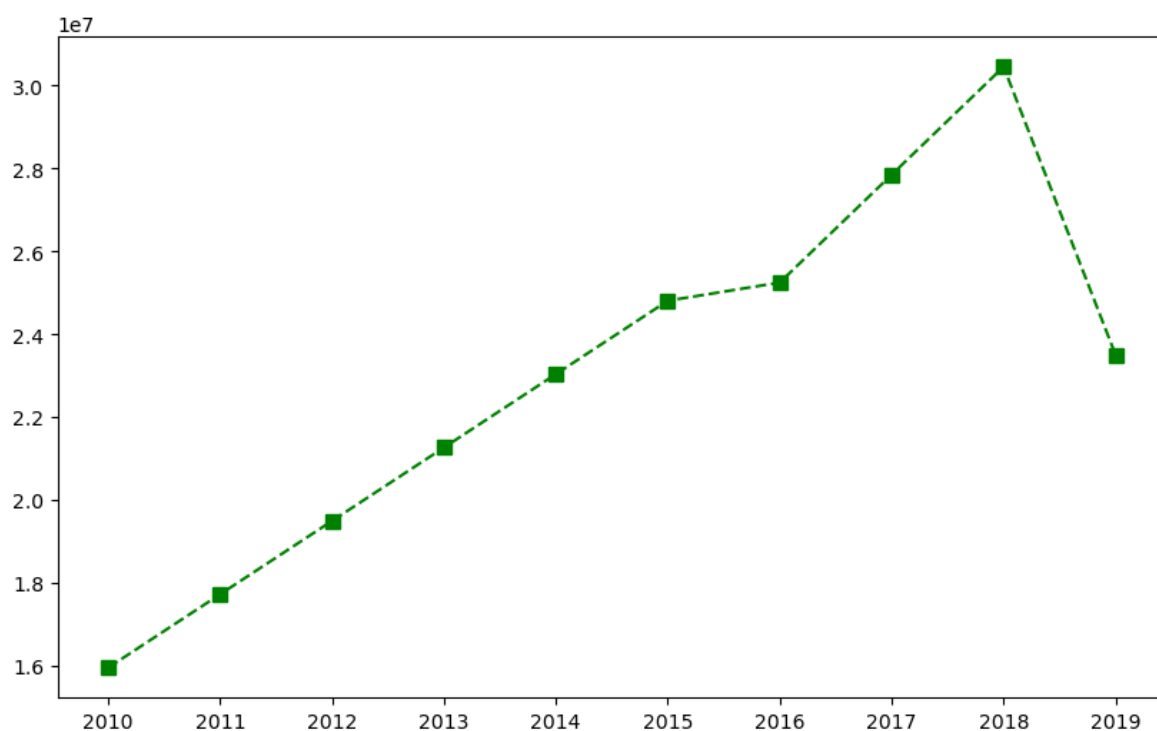
```
In [114]: plt.plot(Salary[0], c = 'green', ls = '--', marker = 's', ms = 7)
plt.xticks(list(range(0, 10)), Seasons, rotation = 'vertical')
plt.show()
```



```
In [115]: Games
```

```
Out[115]: 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 [116]: plt.plot(Salary[0], c = 'green', ls = '--', marker = 's', ms = 7, label = Play  
plt.xticks(list(range(0, 10)), Seasons, rotation = 'horizontal')  
plt.show()
```

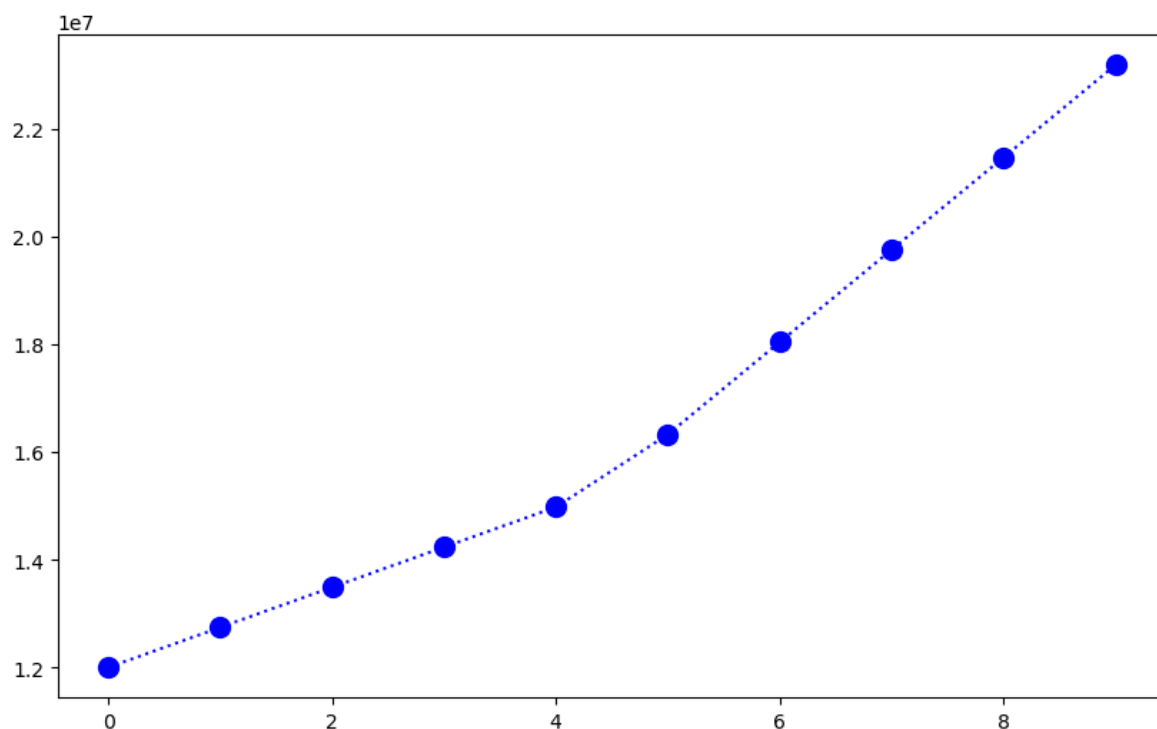


```
In [117]: Salary[0]
```

```
Out[117]: array([15946875, 17718750, 19490625, 21262500, 23034375, 24806250,  
                25244493, 27849149, 30453805, 23500000])
```

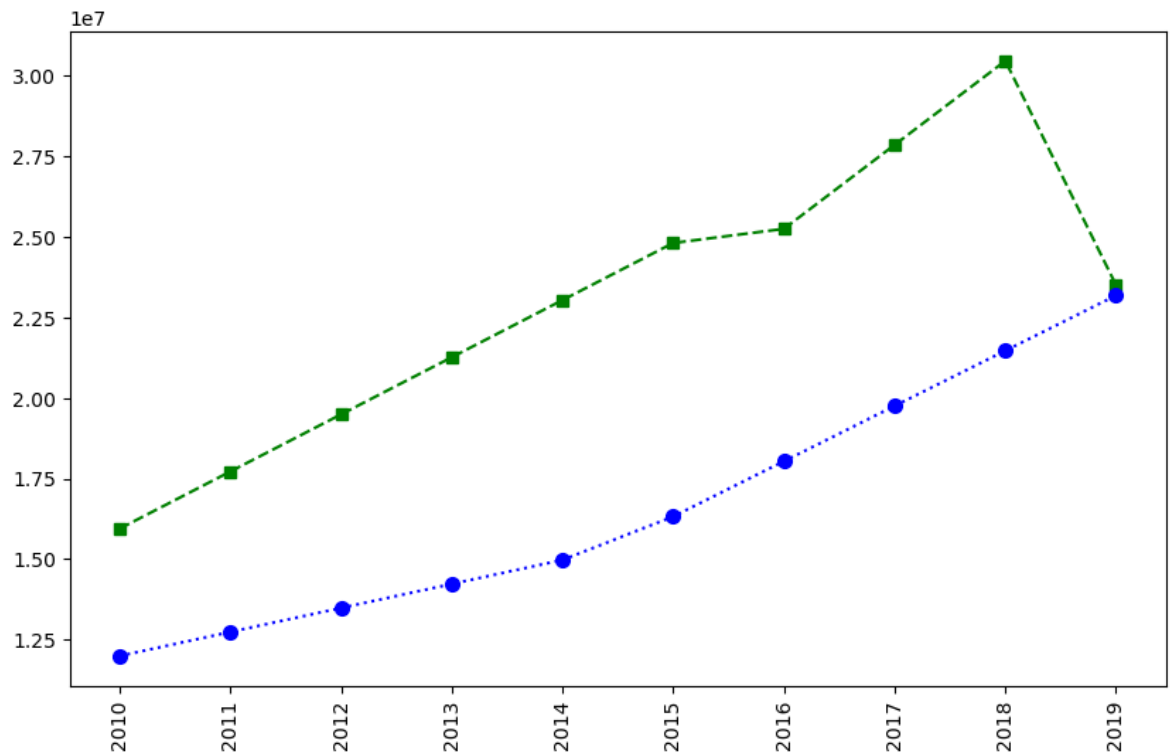
```
In [118]: plt.plot(Salary[1], c = 'Blue', ls = ':', marker = 'o', ms = 10, label = Player
```

```
Out[118]: [<matplotlib.lines.Line2D at 0x2540f5390c0>]
```



```
In [119]: # More Visualization
```

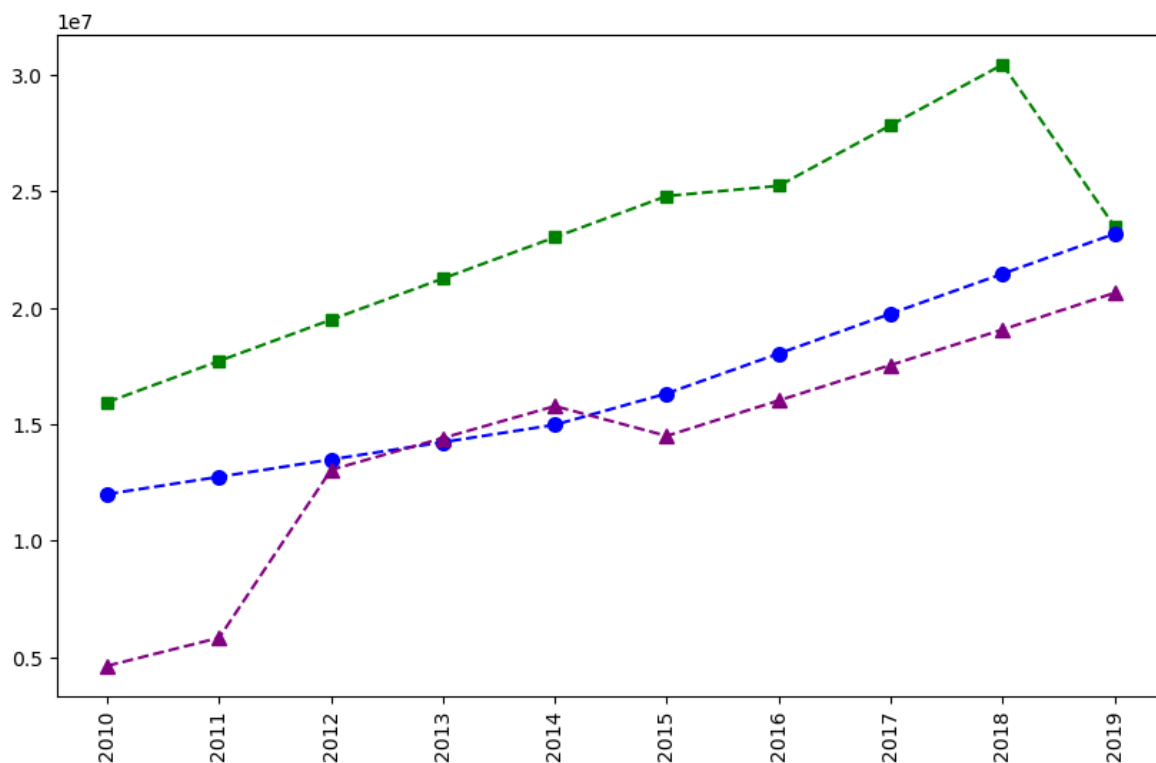
```
In [125]: plt.plot(Salary[0], c = 'green', ls = '--', marker = 's', ms = 6, label = Player1)
plt.plot(Salary[1], c = 'Blue', ls = ':', marker = 'o', ms = 7, label = Player2)
plt.xticks(list(range(0,10)), Seasons,rotation='vertical')
plt.show()
```



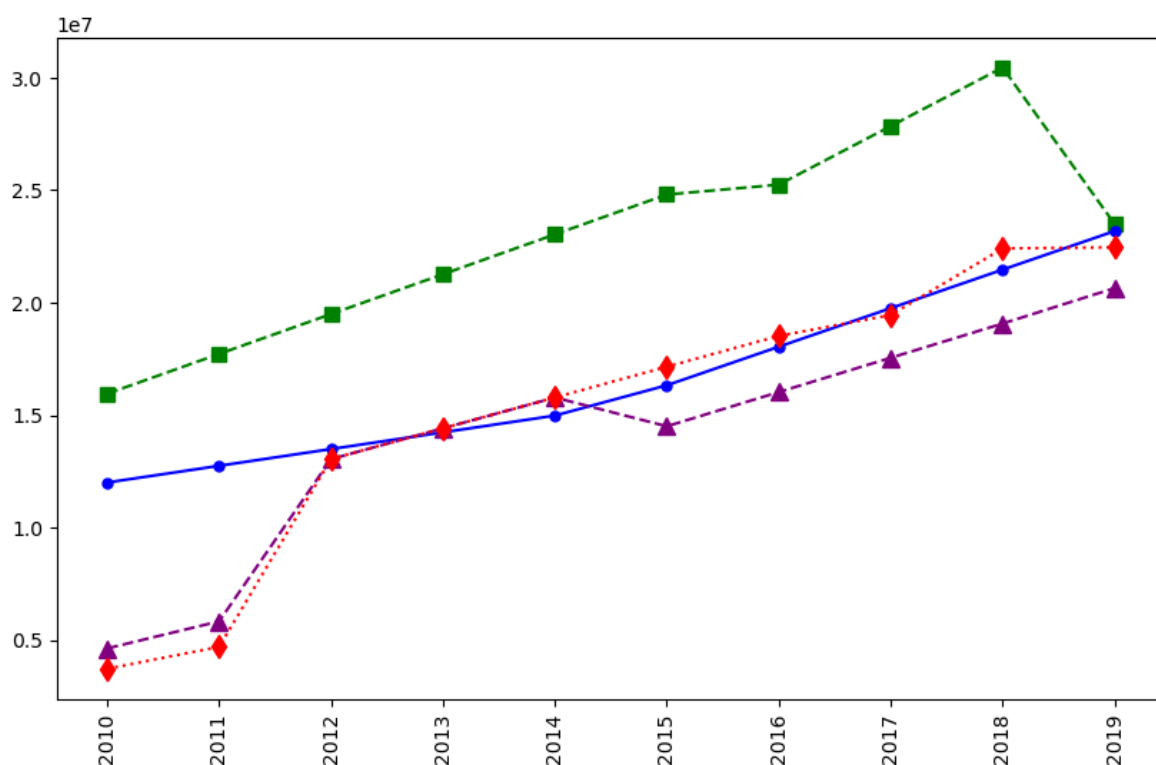
Interpretation Insight

- first player salary is higher compare to second player
- second player has consistent performance but salary of first player higher than second player

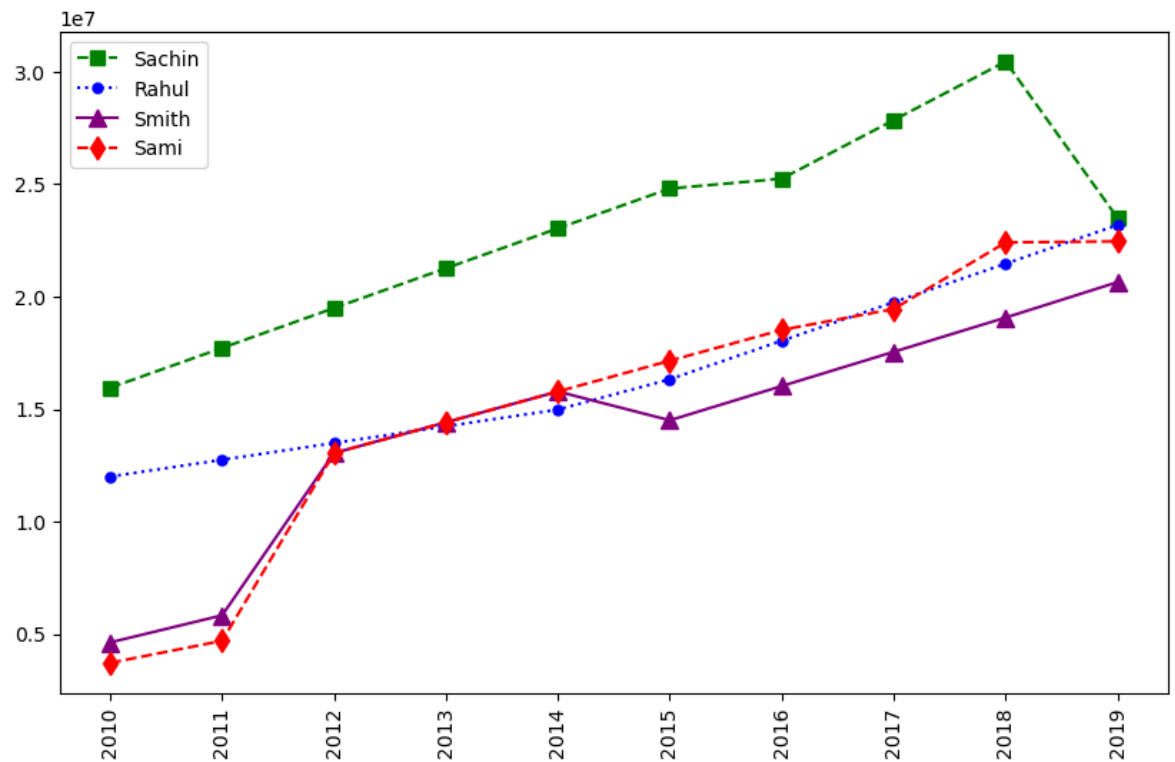
```
In [126]: plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 6, label = Player
plt.plot(Salary[1], c='Blue', ls = '--', marker = 'o', ms = 7, label = Players
plt.plot(Salary[2], c='purple', ls = '--', marker = '^', ms = 7, label = Playe
plt.xticks(list(range(0,10)), Seasons,rotation='vertical')
plt.show()
```



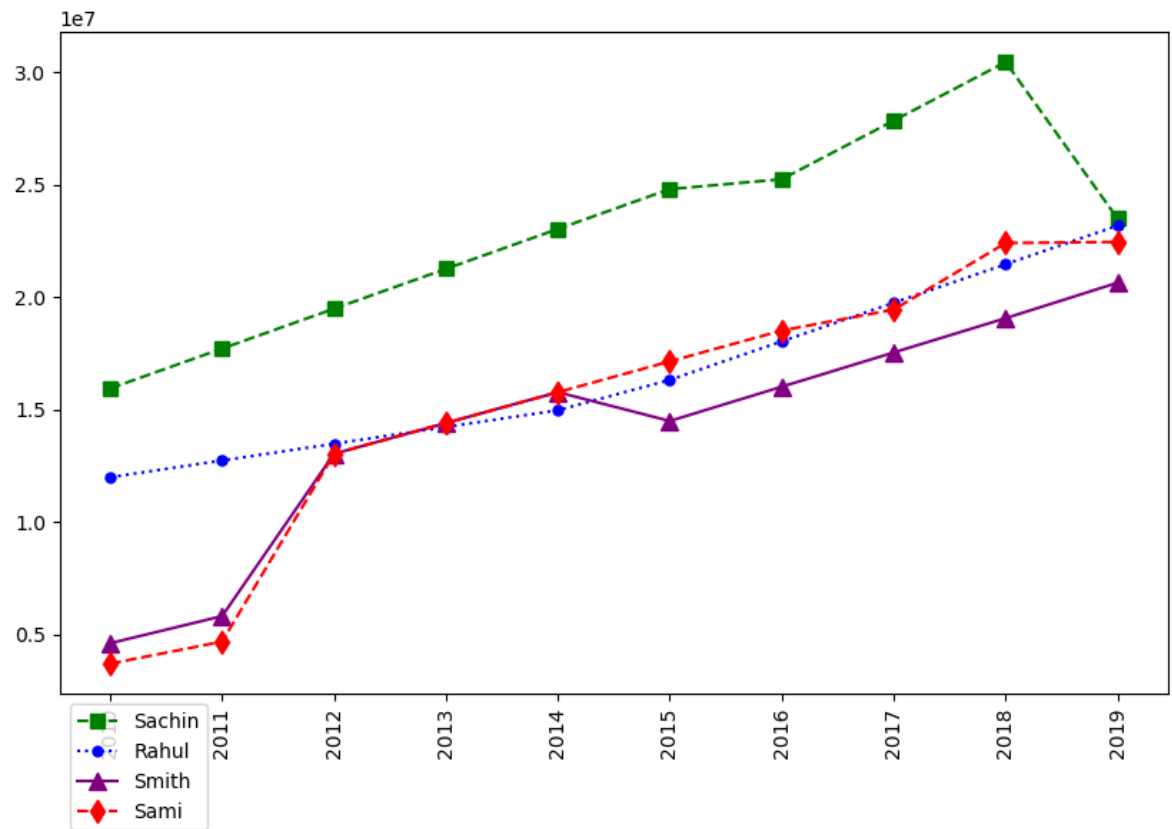
```
In [127]: plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Player
plt.plot(Salary[1], c='Blue', ls = '-', marker = 'o', ms = 5, label = Players[
plt.plot(Salary[2], c='purple', ls = '--', marker = '^', ms = 8, label = Playe
plt.plot(Salary[3], c='Red', ls = ':', marker = 'd', ms = 8, label = Players[3
plt.xticks(list(range(0,10)), Seasons,rotation='vertical')
plt.show()
```



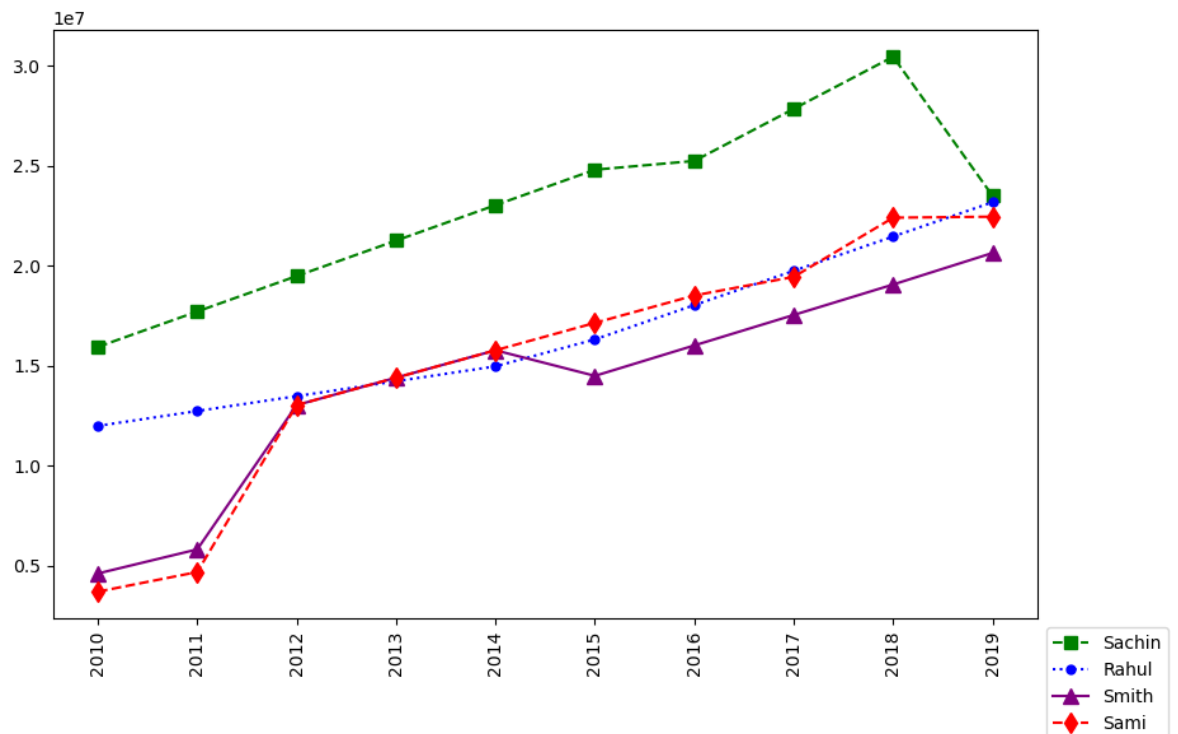
```
In [128]: plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Player
plt.plot(Salary[1], c='Blue', ls = ':', marker = 'o', ms = 5, label = Players[
plt.plot(Salary[2], c='purple', ls = '-', marker = '^', ms = 8, label = Player
plt.plot(Salary[3], c='Red', ls = '--', marker = 'd', ms = 8, label = Players[
plt.legend()
plt.xticks(list(range(0,10)), Seasons,rotation='vertical')
plt.show()
```



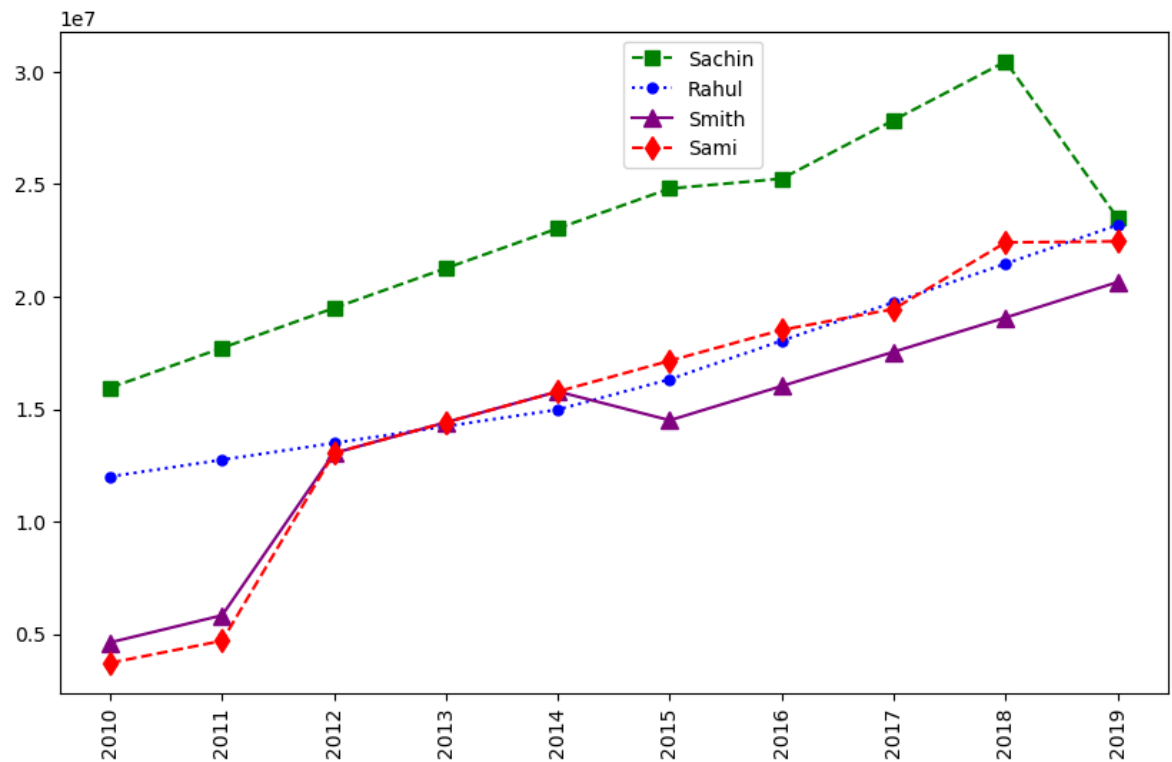

```
In [129]: plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Player
plt.plot(Salary[1], c='Blue', ls = ':', marker = 'o', ms = 5, label = Players[
plt.plot(Salary[2], c='purple', ls = '-', marker = '^', ms = 8, label = Player
plt.plot(Salary[3], c='Red', ls = '--', marker = 'd', ms = 8, label = Players[
plt.legend(loc = 'upper left', bbox_to_anchor=(0, 0))
plt.xticks(list(range(0,10)), Seasons,rotation='vertical')
plt.show()
```



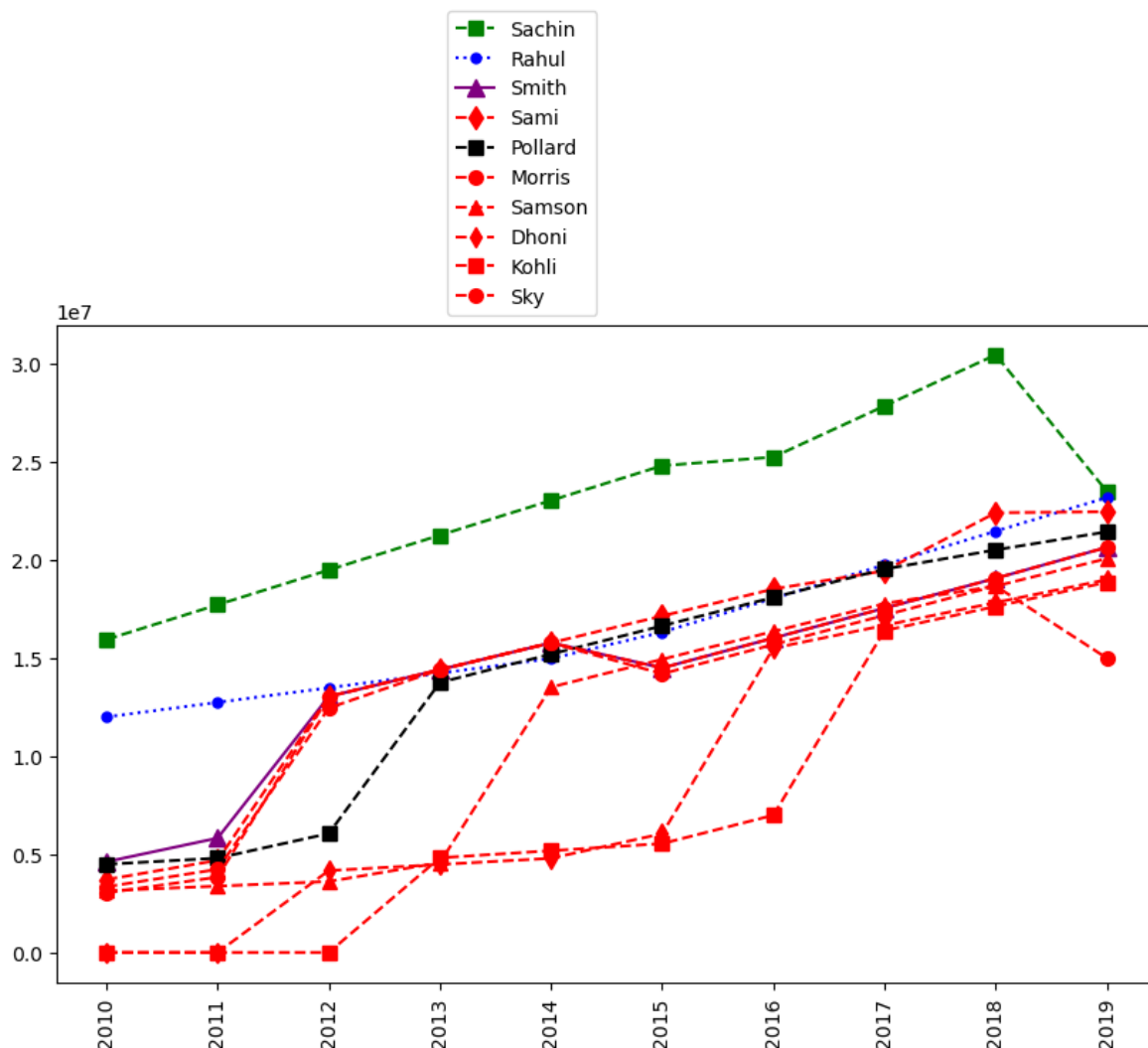
```
In [130]: plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Player
plt.plot(Salary[1], c='Blue', ls = ':', marker = 'o', ms = 5, label = Players[
plt.plot(Salary[2], c='purple', ls = '-', marker = '^', ms = 8, label = Player
plt.plot(Salary[3], c='Red', ls = '--', marker = 'd', ms = 8, label = Players[
plt.legend(loc = 'upper left', bbox_to_anchor=(1, 0))
plt.xticks(list(range(0,10)), Seasons,rotation='vertical')
plt.show()
```



```
In [131]: plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Player
plt.plot(Salary[1], c='Blue', ls = ':', marker = 'o', ms = 5, label = Players[
plt.plot(Salary[2], c='purple', ls = '-', marker = '^', ms = 8, label = Player
plt.plot(Salary[3], c='Red', ls = '--', marker = 'd', ms = 8, label = Players[
plt.legend(loc = 'upper left', bbox_to_anchor=(0.5, 1))
plt.xticks(list(range(0,10)), Seasons,rotation='vertical')
plt.show()
```



```
In [132]: plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Player
plt.plot(Salary[1], c='Blue', ls = ':', marker = 'o', ms = 5, label = Players[
plt.plot(Salary[2], c='purple', ls = '-', marker = '^', ms = 8, label = Player
plt.plot(Salary[3], c='Red', ls = '--', marker = 'd', ms = 8, label = Players[
plt.plot(Salary[4], c='Black', ls = '--', marker = 's', ms = 7, label = Player
plt.plot(Salary[5], c='Red', ls = '--', marker = 'o', ms = 7, label = Players[
plt.plot(Salary[6], c='Red', ls = '--', marker = '^', ms = 7, label = Players[
plt.plot(Salary[7], c='Red', ls = '--', marker = 'd', ms = 7, label = Players[
plt.plot(Salary[8], c='Red', ls = '--', marker = 's', ms = 7, label = Players[
plt.plot(Salary[9], c='Red', ls = '--', marker = 'o', ms = 7, label = Players[
plt.legend(loc = 'lower right', bbox_to_anchor=(0.5, 1))
plt.xticks(list(range(0,10)), Seasons,rotation='vertical')
plt.show()
```



```
In [133]: # 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()
```

