

# 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
- Indexation is very important to plot the datapoints
- we will see that & 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 the past 10 yrs
- Our main goal is to find trends, patterns & their performance for the past 10 yrs

In [109...]

```
#Import numpy
import numpy as np

#Seasons
seasons = ["2012", "2013", "2014", "2015", "2016", "2017", "2018", "2019", "2020", "2021"]
sdict = {"2012":0, "2013":1, "2014":2, "2015":3, "2016":4, "2017":5, "2018":6, "2019":7, "2020":8, "2021":9}

#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}

#Salaries
Sachin_Salary = [15946875, 17718750, 19490625, 21262500, 23034375, 24806250, 25244493, 278150000, 300000000, 330000000, 360000000, 390000000, 420000000, 450000000, 480000000, 510000000, 540000000]
Rahul_Salary = [12000000, 12744189, 13488377, 14232567, 14976754, 16324500, 18038573, 197520000, 22744189, 24488377, 26232567, 27976754, 29324500, 31038573, 327520000]
Smith_Salary = [4621800, 5828090, 13041250, 14410581, 15779912, 14500000, 16022500, 17545021000, 23041250, 34410581, 47779912, 6149243, 78518574, 9450000]
Sami_Salary = [3713640, 4694041, 13041250, 14410581, 15779912, 17149243, 18518574, 1945000022000, 31041250, 40410581, 53779912, 66149243, 81518574, 9845000]
Pollard_Salary = [4493160, 4806720, 6061274, 13758000, 15202590, 16647180, 18091770, 1953623000, 4806720, 5202590, 6647180, 8091770, 9536]
Morris_Salary = [3348000, 4235220, 12455000, 14410581, 15779912, 14500000, 16022500, 1754524000, 5135220, 6550000, 84410581, 97779912, 114500000, 13022500, 147545]
Samson_Salary = [3144240, 3380160, 3615960, 4574189, 13520500, 14940153, 16359805, 177794525000, 5184480, 5546160, 6993708, 16402500, 17632688, 18862875]
Dhoni_Salary = [0, 0, 4171200, 4484040, 4796880, 6053663, 15506632, 16669630, 17832627, 189928000, 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]

#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

#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_
```

In [110...]: Salary # matrix format

```
Out[110...]: 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 [111...]: # Building your first matrix -  
Games

```
Out[111...]: 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 [112... Points

```
Out[112... 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 [113... 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 [114... np.reshape(mydata,(4,5)) # 5 rows &amp; 4 columns

```
Out[114... array([[ 0,  1,  2,  3,  4],
   [ 5,  6,  7,  8,  9],
   [10, 11, 12, 13, 14],
   [15, 16, 17, 18, 19]])
```

In [115... mydata

```
Out[115... array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
   17, 18, 19])
```

```
In [116... #np.reshape(mydata,(5,4), order = 'c')
# 'c' means to read / write the elements using C-Like index order
MATR1 = np.reshape(mydata, (5,4), order = 'c')
MATR1
```

```
Out[116... array([[ 0,  1,  2,  3],
   [ 4,  5,  6,  7],
   [ 8,  9, 10, 11],
   [12, 13, 14, 15],
   [16, 17, 18, 19]])
```

In [117... MATR1

```
Out[117... array([[ 0,  1,  2,  3],
   [ 4,  5,  6,  7],
   [ 8,  9, 10, 11],
   [12, 13, 14, 15],
   [16, 17, 18, 19]])
```

```
In [118... # If i want to get only no.3
MATR1[4,3]
```

```
Out[118... 19
```

In [119... MATR1[3,3]

```
Out[119... 15
```

```
In [120... MATR1[4,2]
```

```
Out[120... 18
```

```
In [121... MATR1
```

```
Out[121... array([[ 0,  1,  2,  3],
                  [ 4,  5,  6,  7],
                  [ 8,  9, 10, 11],
                  [12, 13, 14, 15],
                  [16, 17, 18, 19]])
```

```
In [122... mydata
```

```
Out[122... array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
                  17, 18, 19])
```

```
In [123... MATR2 = np.reshape(mydata, (5,4), order = 'F') # reshape behaviour are - 'C', 'F', '
MATR2
```

```
Out[123... array([[ 0,  5, 10, 15],
                  [ 1,  6, 11, 16],
                  [ 2,  7, 12, 17],
                  [ 3,  8, 13, 18],
                  [ 4,  9, 14, 19]])
```

```
In [124... MATR2[3,3]
```

```
Out[124... 18
```

```
In [125... MATR2[4,3]
```

```
Out[125... 19
```

```
In [126... MATR2[0:2]
```

```
Out[126... array([[ 0,  5, 10, 15],
                  [ 1,  6, 11, 16]])
```

```
In [127... MATR2[2:4]
```

```
Out[127... array([[ 2,  7, 12, 17],
                  [ 3,  8, 13, 18]])
```

```
In [128... MATR2[4:4]
```

```
Out[128... array([], shape=(0, 4), dtype=int32)
```

```
In [129... MATR2[1:2]
```

```
Out[129... array([[ 1,  6, 11, 16]])
```

```
In [130... MATR2[1,2]
```

```
Out[130... 11
```

```
In [131... MATR2
```

```
Out[131... array([[ 0,  5, 10, 15],
                  [ 1,  6, 11, 16],
                  [ 2,  7, 12, 17],
                  [ 3,  8, 13, 18],
                  [ 4,  9, 14, 19]])
```

```
In [132... MATR2[-2,-1]
```

```
Out[132... 18
```

```
In [133... MATR2[-4,-1]
```

```
Out[133... 16
```

```
In [134... MATR2[-3,-3]
```

```
Out[134... 7
```

```
In [135... MATR2[-2,3]
```

```
Out[135... 18
```

```
In [136... MATR2
```

```
Out[136... array([[ 0,  5, 10, 15],
                  [ 1,  6, 11, 16],
                  [ 2,  7, 12, 17],
                  [ 3,  8, 13, 18],
                  [ 4,  9, 14, 19]])
```

```
In [137... MATR2[0:2]
```

```
Out[137... array([[ 0,  5, 10, 15],
                  [ 1,  6, 11, 16]])
```

```
In [138... mydata
```

```
Out[138... array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
                  17, 18, 19])
```

```
In [139... MATR3 = np.reshape(mydata, (5,4), order = 'A')
MATR3
```

```
Out[139... array([[ 0,  1,  2,  3],
                  [ 4,  5,  6,  7],
                  [ 8,  9, 10, 11],
                  [12, 13, 14, 15],
                  [16, 17, 18, 19]])
```

In [140... MATR2 ## F shaped

```
Out[140... array([[ 0,  5, 10, 15],
                  [ 1,  6, 11, 16],
                  [ 2,  7, 12, 17],
                  [ 3,  8, 13, 18],
                  [ 4,  9, 14, 19]])
```

In [141... MATR1 # c shaped

```
Out[141... array([[ 0,  1,  2,  3],
                  [ 4,  5,  6,  7],
                  [ 8,  9, 10, 11],
                  [12, 13, 14, 15],
                  [16, 17, 18, 19]])
```

In [142... a1 = ['welcome', 'to', 'datascience']
a2 = ['requiried', 'hard', 'work']
a3 = [1,2,3]

In [143... [a1,a2,a3] # List same datatype

```
Out[143... [['welcome', 'to', 'datascience'], ['requiried', 'hard', 'work'], [1, 2, 3]]
```

In [144... np.array([a1,a2,a3]) # u11 - unicode 11 character : 3\*3 matrix

```
Out[144... array([['welcome', 'to', 'datascience'],
                  ['requiried', 'hard', 'work'],
                  ['1', '2', '3']], dtype='<U11')
```

In [145... Games

```
Out[145... 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 [146... Games

```
Out[146... 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 [147... Games[0]
```

```
Out[147... array([80, 77, 82, 82, 73, 82, 58, 78, 6, 35])
```

```
In [148... Games[5]
```

```
Out[148... array([70, 69, 67, 77, 70, 77, 57, 74, 79, 44])
```

```
In [149... Games[0:5]
```

```
Out[149... 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 [150... Games[0,5]
```

```
Out[150... 82
```

```
In [151... Games[0,2]
```

```
Out[151... 82
```

```
In [152... Games
```

```
Out[152... 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 [153... Games[0:2]
```

```
Out[153... array([[80, 77, 82, 82, 73, 82, 58, 78, 6, 35],  
[82, 57, 82, 79, 76, 72, 60, 72, 79, 80]])
```

```
In [154... Games
```

```
Out[154... 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 [155... Games[1:2]
```

```
Out[155... array([[82, 57, 82, 79, 76, 72, 60, 72, 79, 80]])
```

```
In [156... Games[2]
```

```
Out[156... array([79, 78, 75, 81, 76, 79, 62, 76, 77, 69])
```

```
In [157... Games
```

```
Out[157... 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 [158... Games[2,8]
```

```
Out[158... 77
```

```
In [159... Games
```

```
Out[159... 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 [160... Games[-3:-1]
```

```
Out[160... array([[35, 35, 80, 74, 82, 78, 66, 81, 81, 27],  
[40, 40, 40, 81, 78, 81, 39, 0, 10, 51]])
```

```
In [161... Games[-3,-1]
```

```
Out[161... 27
```

```
In [162... Points
```

```
Out[162... 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 [163... Points[0]
```

```
Out[163... array([2832, 2430, 2323, 2201, 1970, 2078, 1616, 2133, 83, 782])
```

```
In [164... Points
```

```
Out[164... 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 [165... Points[6,1]
```

```
Out[165... 1104
```

```
In [166... Points[3:6]
```

```
Out[166... 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 [167... Points
```

```
Out[167... 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 [168... Points[-6,-1]
```

```
Out[168... 646
```

```
In [169...]: ====== DICTIONARY =====#
# dict does not maintain the order
dict1 = {'keys': 'val1', 'key2': 'val2', 'key3': 'val3'}
```

```
In [170...]: dict1
```

```
Out[170...]: {'keys': 'val1', 'key2': 'val2', 'key3': 'val3'}
```

```
In [171...]: dict1['key2']
```

```
Out[171...]: 'val2'
```

```
In [172...]: dict2 = {'bang': 2, 'hyd': 'we are hear', 'pune': True}
```

```
In [173...]: dict2
```

```
Out[173...]: {'bang': 2, 'hyd': 'we are hear', 'pune': True}
```

```
In [174...]: dict3 = {'Germany': 'I have been here', 'France': 2, 'Spain': True}
```

```
In [175...]: dict3
```

```
Out[175...]: {'Germany': 'I have been here', 'France': 2, 'Spain': True}
```

```
In [176...]: Games
```

```
Out[176...]: 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 [177...]: pdict['Rahul']
```

```
Out[177...]: 1
```

```
In [178...]: Games[1]
```

```
Out[178...]: array([82, 57, 82, 79, 76, 72, 60, 72, 79, 80])
```

## Games

```
In [179...]: Games[pdict['Rahul']]
```

```
Out[179... array([82, 57, 82, 79, 76, 72, 60, 72, 79, 80])
```

```
In [180... Points
```

```
Out[180... 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 [181... Salary
```

```
Out[181... 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,  

```

```
In [182... Salary[2,4]
```

```
Out[182... 15779912
```

```
In [183... Salary
```

```
Out[183... 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, 4822800, 5184480, 5546160,
   6993708, 16402500, 17632688, 18862875],
   [ 3031920, 3841443, 13041250, 14410581, 15779912, 14200000,
   15691000, 17182000, 18673000, 15000000]])
```

In [184... Salary[pdict['Sky']][sdict['2019']]

Out[184... 17182000

In [185... Salary

```
Out[185... 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, 4822800, 5184480, 5546160,
   6993708, 16402500, 17632688, 18862875],
   [ 3031920, 3841443, 13041250, 14410581, 15779912, 14200000,
   15691000, 17182000, 18673000, 15000000]])
```

In [186... Games

```
Out[186... 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 [187... Salary/Games

```
Out[187... 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 [188... np.round(Salary/Games)

```
Out[188]: 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 [189]: import warnings
warnings.filterwarnings('ignore')
#np.round(FieldGoals/Games)
#FieldGoals/Games # this matrix is lot of decimal points yo can not round
#round()
```

```
In [190]: ## --- First visualization ----##
```

```
In [191]: import numpy as np
import matplotlib.pyplot as plt
```

```
In [192]: %matplotlib inline # keep the plot inside jupyter notes insted of getting in other
UsageError: unrecognized arguments: # keep the plot inside jupyter notes insted of g
etting in other screen
```

```
In [193]: Salary
```

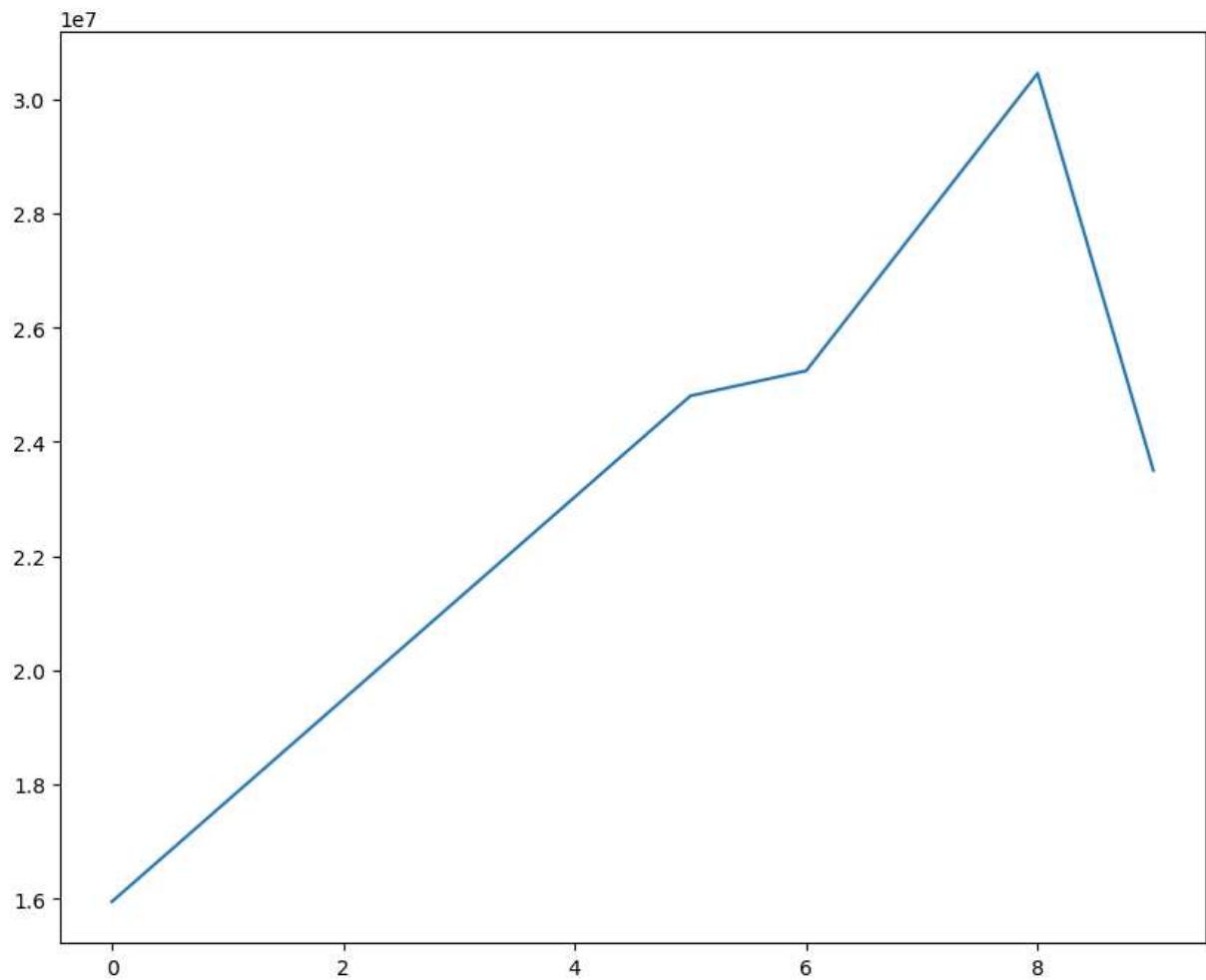
```
Out[193... array([[15946875, 17718750, 19490625, 21262500, 23034375, 24806250,
   25244493, 27849149, 30453805, 23500000],
   [12000000, 12744189, 13488377, 14232567, 14976754, 16324500,
   18038573, 19752645, 21466718, 23180790],
   [ 4621800, 5828090, 13041250, 14410581, 15779912, 14500000,
   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 [194... Salary[0]
```

```
Out[194... array([15946875, 17718750, 19490625, 21262500, 23034375, 24806250,
   25244493, 27849149, 30453805, 23500000])
```

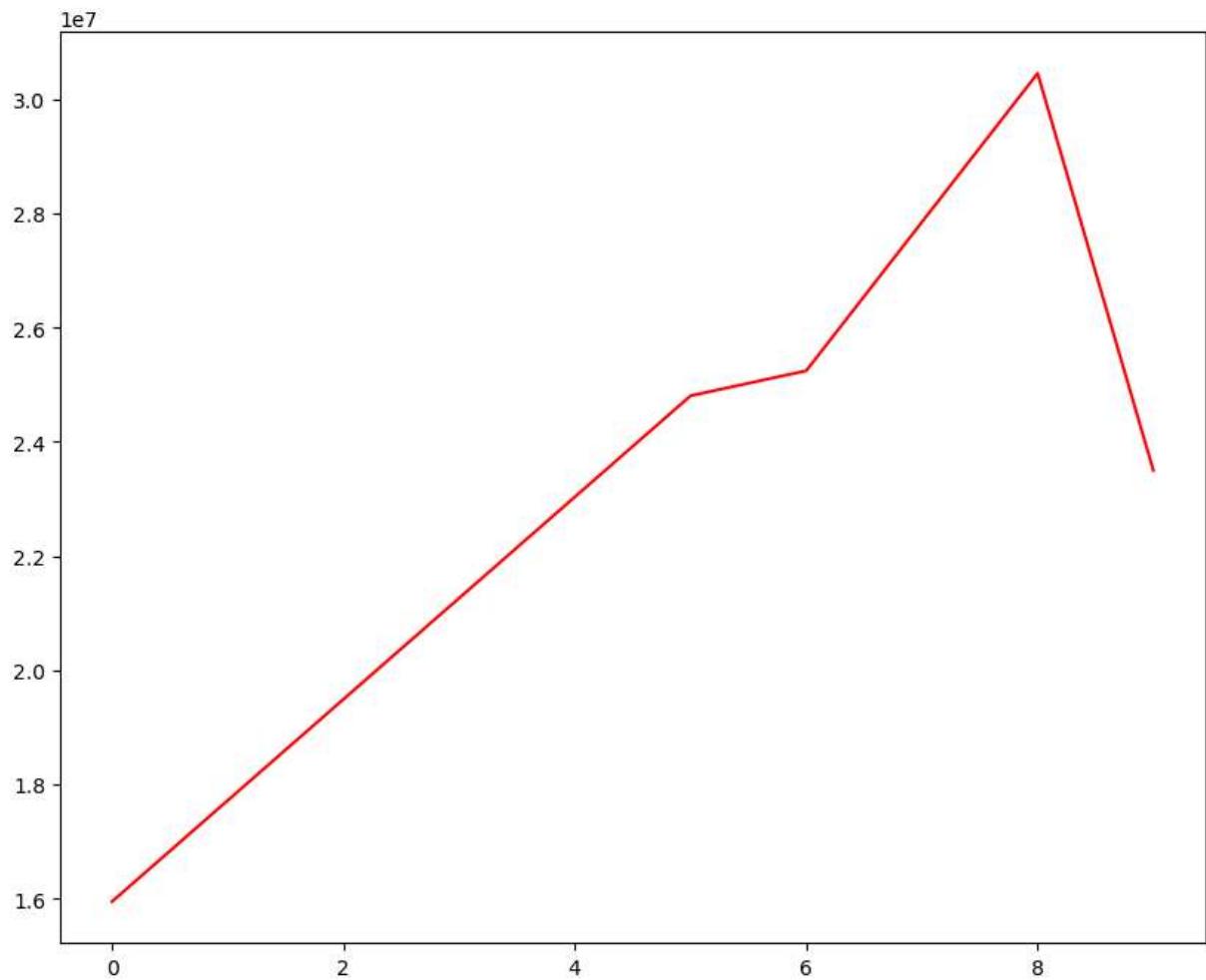
```
In [195... plt.plot(Salary[0])
```

```
Out[195... [matplotlib.lines.Line2D at 0x29a01c08bf0]]
```



```
In [196...]: plt.plot(Salary[0], c='red')
```

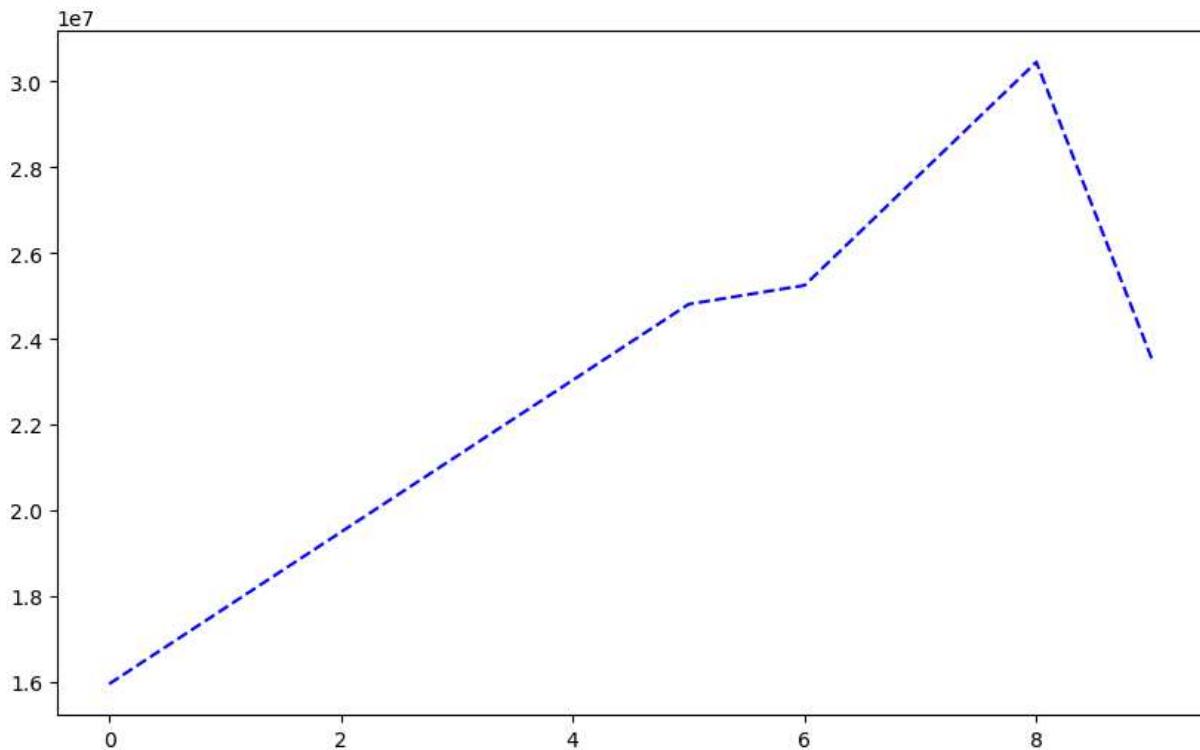
```
Out[196...]: <matplotlib.lines.Line2D at 0x29a02120fb0>
```



```
In [197...]: %matplotlib inline  
plt.rcParams['figure.figsize'] = 10,6
```

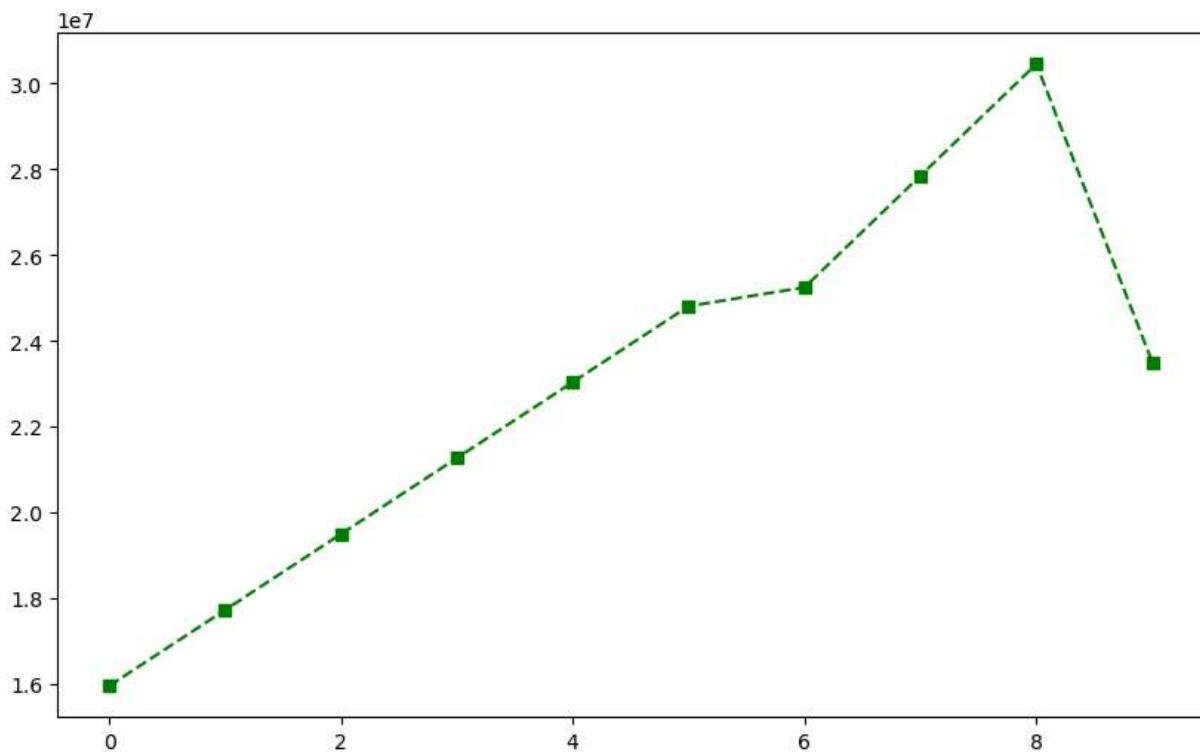
```
In [198...]: plt.plot(Salary[0], c='Blue', ls = 'dashed')
```

```
Out[198...]: <matplotlib.lines.Line2D at 0x29a02189310>
```



```
In [199... plt.plot(Salary[0], c='Green', ls = '--', marker = 's') # s - squares
```

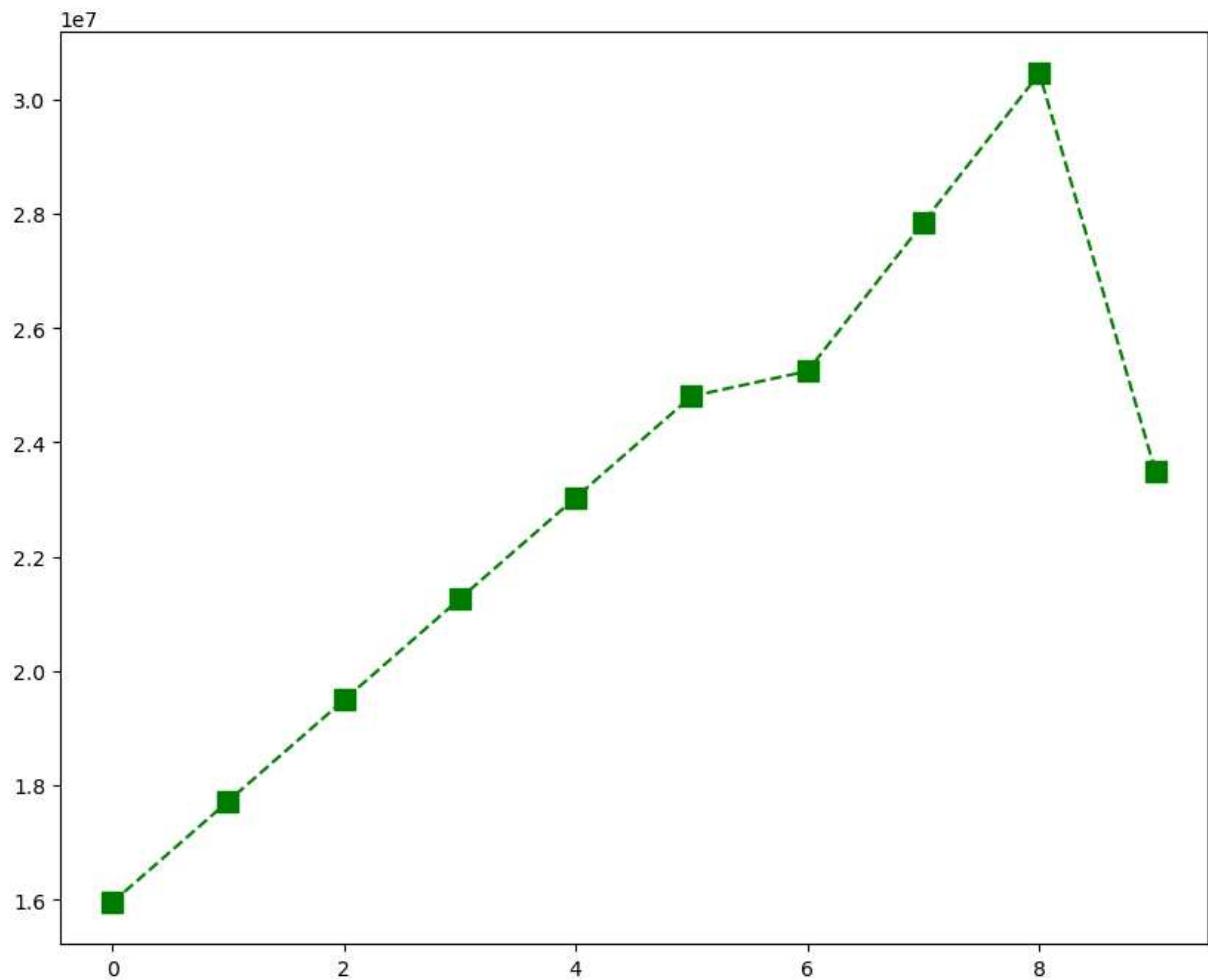
```
Out[199... <matplotlib.lines.Line2D at 0x29a02642990>]
```



```
In [200... %matplotlib inline  
plt.rcParams['figure.figsize'] = 10,8 #runtime configuration parameter
```

```
In [201... plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 10)
```

```
plt.show()
```



```
In [202...]: list(range(0,10))
```

```
Out[202...]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
In [203...]: sdict
```

```
Out[203...]: {'2012': 0,
 '2013': 1,
 '2014': 2,
 '2015': 3,
 '2016': 4,
 '2017': 5,
 '2018': 6,
 '2019': 7,
 '2020': 8,
 '2021': 9}
```

```
In [204...]: pdict
```

```
Out[204... {'Sachin': 0,  
 'Rahul': 1,  
 'Smith': 2,  
 'Sami': 3,  
 'Pollard': 4,  
 'Morris': 5,  
 'Samson': 6,  
 'Dhoni': 7,  
 'Kohli': 8,  
 'Sky': 9}
```

```
In [205... plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7)  
plt.xticks(list(range(0,10)), Seasons)  
plt.show()
```

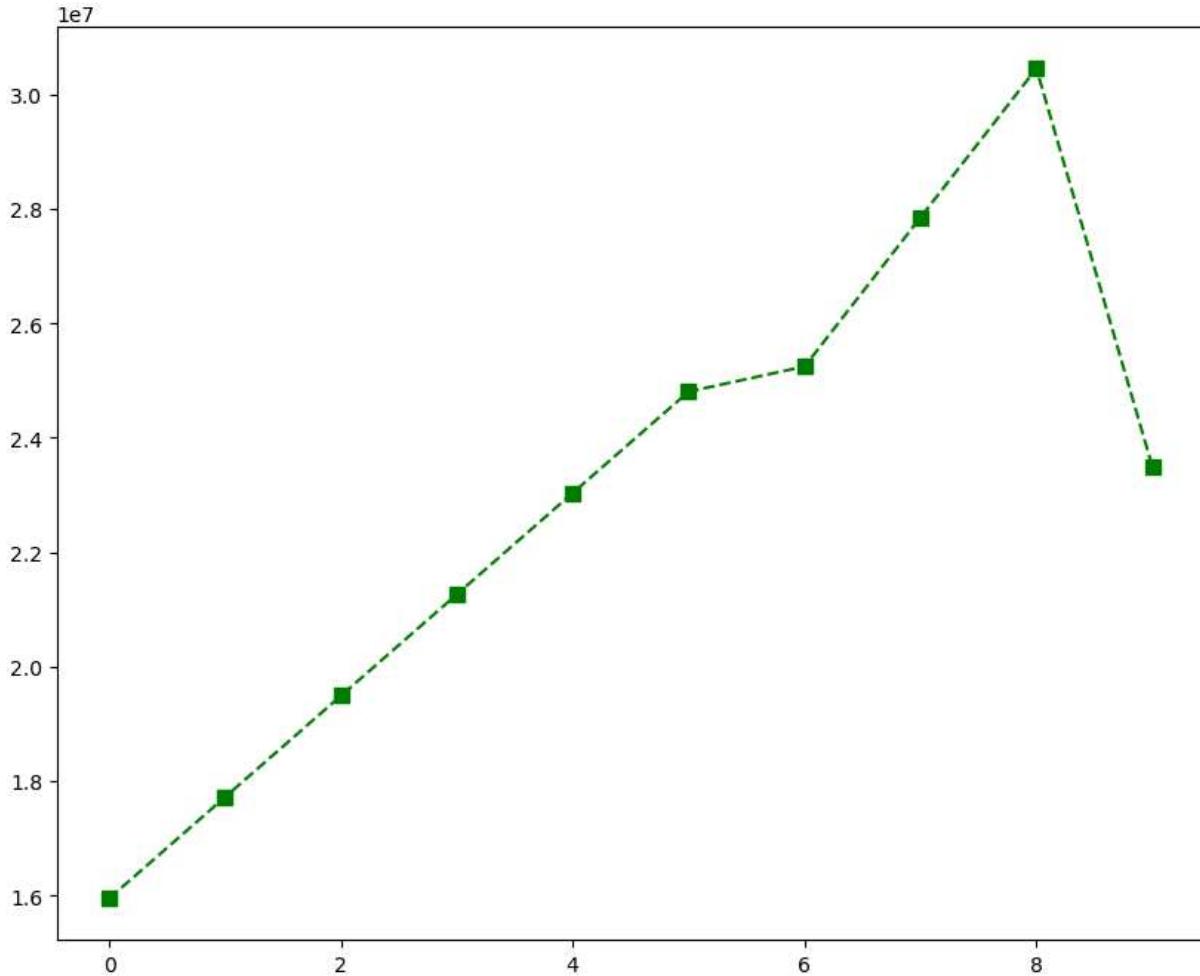
NameError

Traceback (most recent call last)

Cell In[205], line 2

```
1 plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7)  
----> 2 plt.xticks(list(range(0,10)), Seasons)  
3 plt.show()
```

NameError: name 'Seasons' is not defined

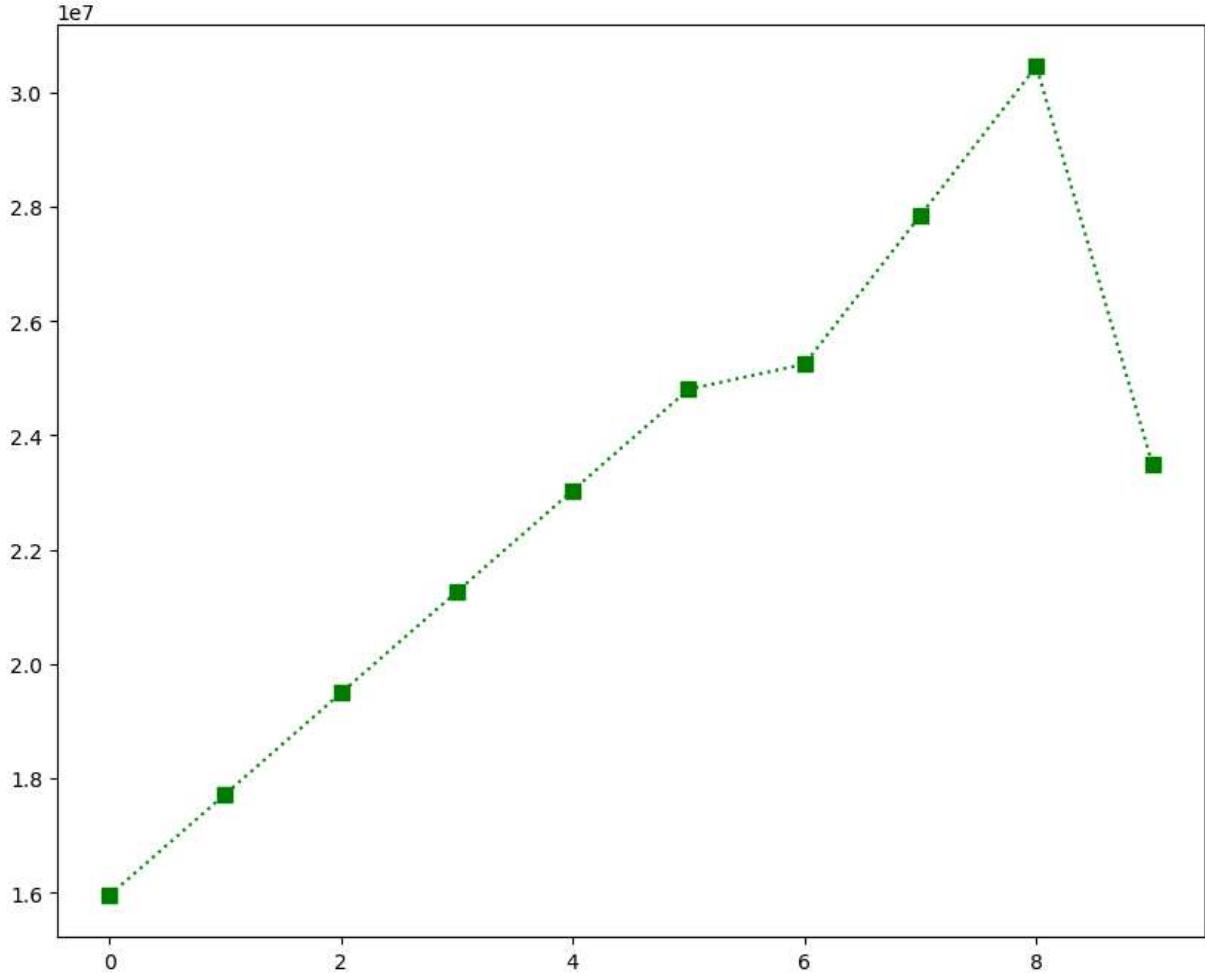


```
In [206... plt.plot(Salary[0], c='Green', ls = ':', marker = 's', ms = 7, label = Players[0])  
plt.xticks(list(range(0,10)), Seasons, rotation='vertical')
```

```
plt.show()
```

```
NameError Traceback (most recent call last)
Cell In[206], line 2
      1 plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Players[0])
      2 plt.xticks(list(range(0,10)), Seasons, rotation='vertical')
      3 plt.show()

NameError: name 'Seasons' is not defined
```



In [207... Games

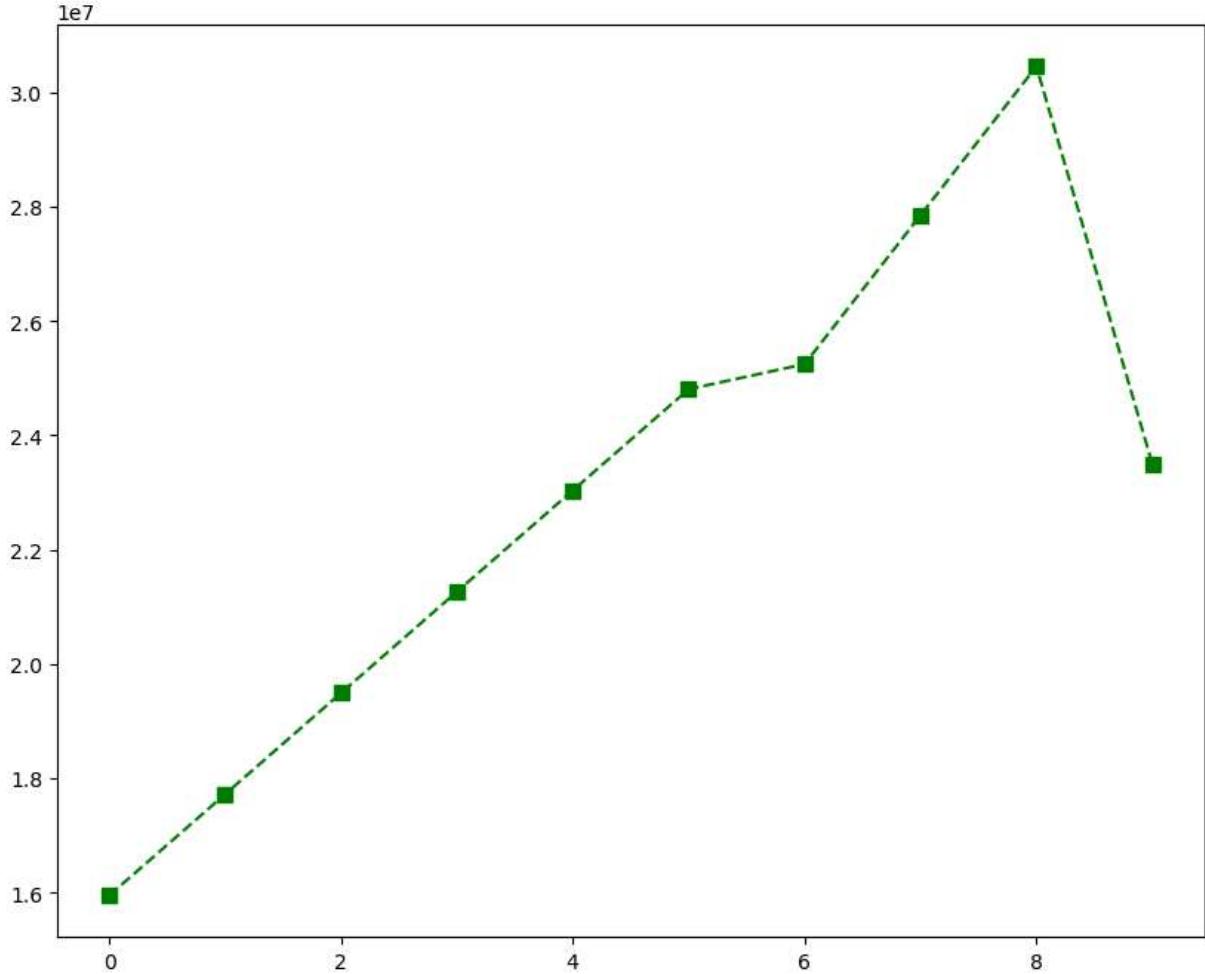
```
Out[207... array([[80, 77, 82, 82, 73, 82, 58, 78, 6, 35],
       [82, 57, 82, 79, 76, 72, 60, 72, 79, 80],
       [79, 78, 75, 81, 76, 79, 62, 76, 77, 69],
       [80, 65, 77, 66, 69, 77, 55, 67, 77, 40],
       [82, 82, 82, 79, 82, 78, 54, 76, 71, 41],
       [70, 69, 67, 77, 70, 77, 57, 74, 79, 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 [208... plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Players[0])
plt.xticks(list(range(0,10)), Seasons, rotation='horizontal')
```

```
plt.show()
```

```
NameError                                                 Traceback (most recent call last)
Cell In[208], line 2
      1 plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Players[0])
      2 plt.xticks(list(range(0,10)), Seasons, rotation='horizontal')
      3 plt.show()

NameError: name 'Seasons' is not defined
```



```
In [209... Salary[0]
```

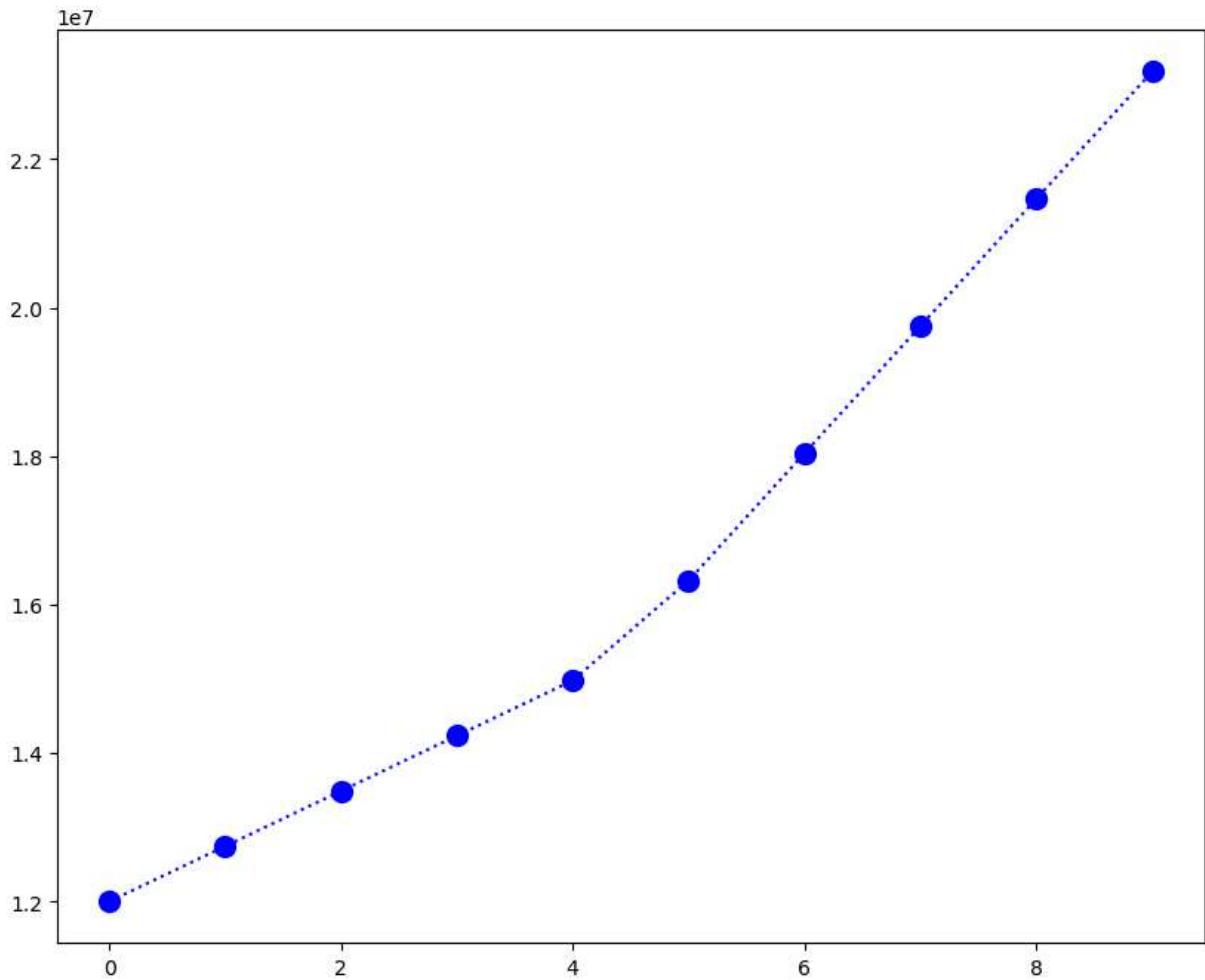
```
Out[209... array([15946875, 17718750, 19490625, 21262500, 23034375, 24806250,
   25244493, 27849149, 30453805, 23500000])
```

```
In [210... Salary[1]
```

```
Out[210... array([12000000, 12744189, 13488377, 14232567, 14976754, 16324500,
   18038573, 19752645, 21466718, 23180790])
```

```
In [211... plt.plot(Salary[1], c='Blue', ls = ':', marker = 'o', ms = 10, label = Players[1])
```

```
Out[211... [<matplotlib.lines.Line2D at 0x29a016ed940>]
```



```
In [212]: # More visualization
```

```
In [214]: plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 10, label = Players[0])
plt.plot(Salary[1], c='Blue', ls = ':', marker = 'o', ms = 10, label = Players[1])

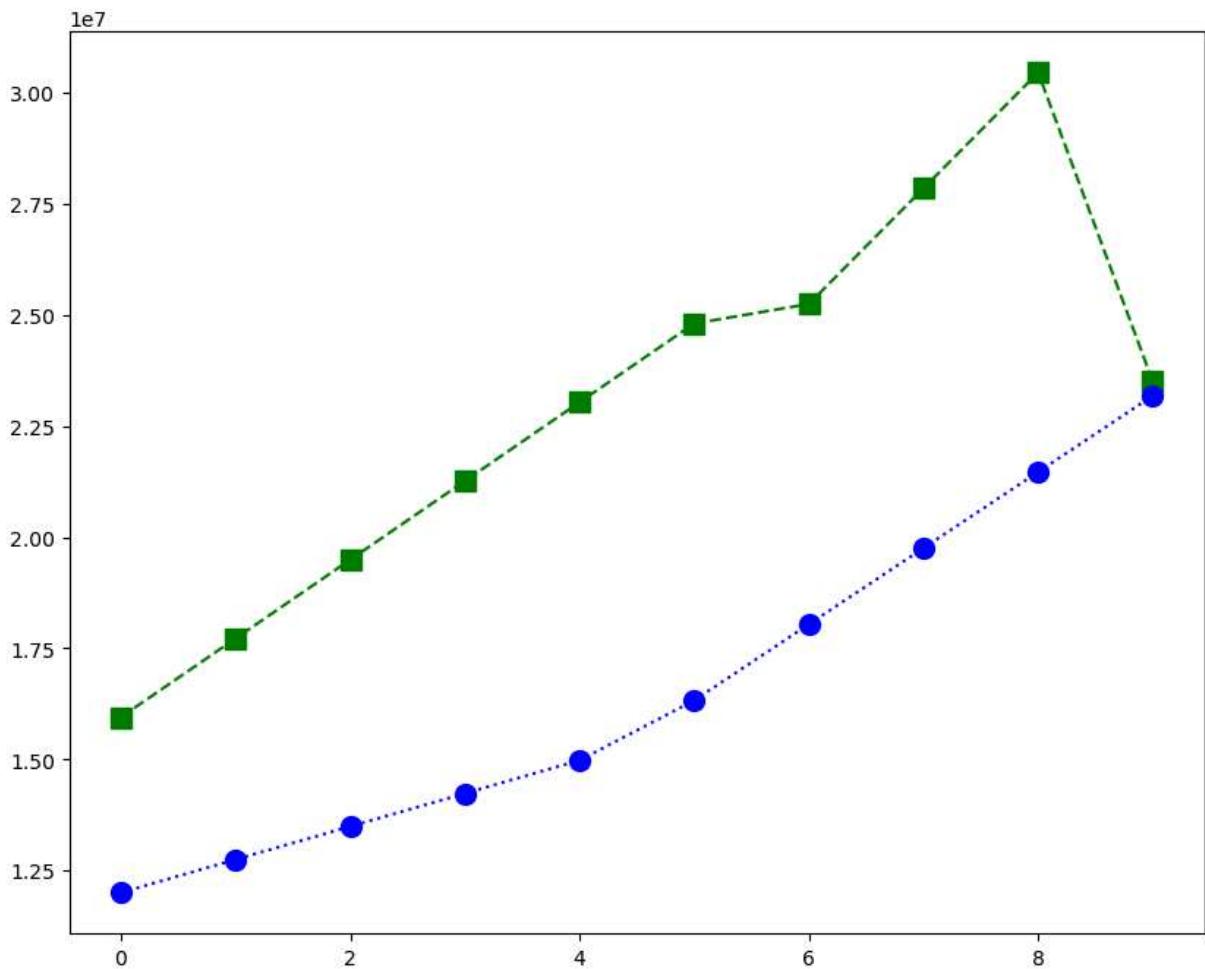
plt.xticks(list(range(0,10)), Seasons, rotation='vertical')

plt.show()
```

-----

**NameError** Traceback (most recent call last)  
Cell In[214], line 4  
1 plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 10, label = Players[0])  
2 plt.plot(Salary[1], c='Blue', ls = ':', marker = 'o', ms = 10, label = Players[1])  
----> 4 plt.xticks(list(range(0,10)), Seasons, rotation='vertical')  
6 plt.show()

**NameError:** name 'Seasons' is not defined



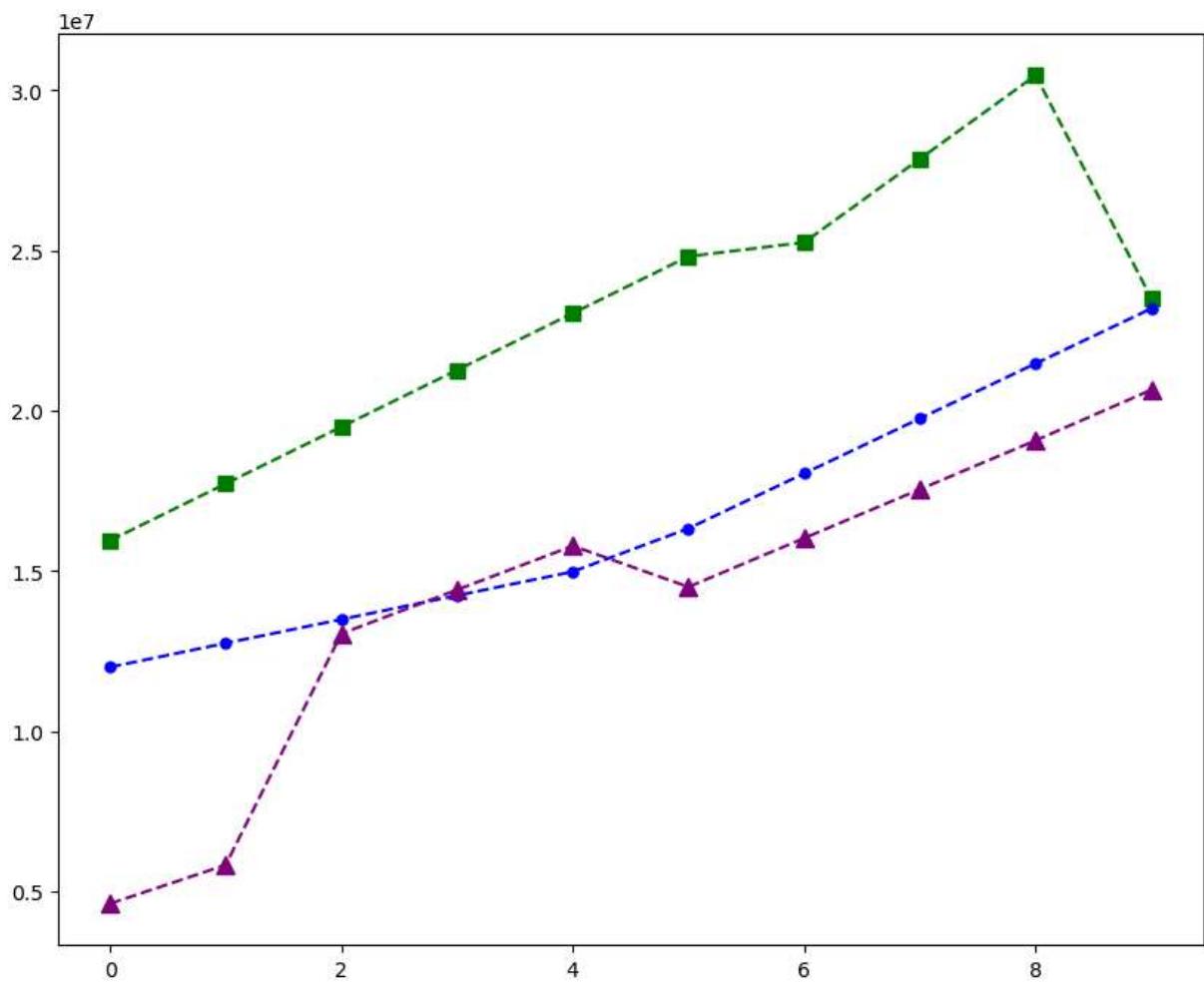
```
In [215]: plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Players[0])
plt.plot(Salary[1], c='Blue', ls = '--', marker = 'o', ms = 5, label = Players[1])
plt.plot(Salary[2], c='purple', ls = '--', marker = '^', ms = 8, label = Players[2])

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

plt.show()
```

```
NameError                                                 Traceback (most recent call last)
Cell In[215], line 6
      2 plt.plot(Salary[1], c='Blue', ls = '--', marker = 'o', ms = 5, label = Playe
      3 rs[1])
      4 plt.plot(Salary[2], c='purple', ls = '--', marker = '^', ms = 8, label = Pla
      5 yers[2])
--> 6 plt.xticks(list(range(0,10)), Seasons, rotation='vertical')
      8 plt.show()

NameError: name 'Seasons' is not defined
```



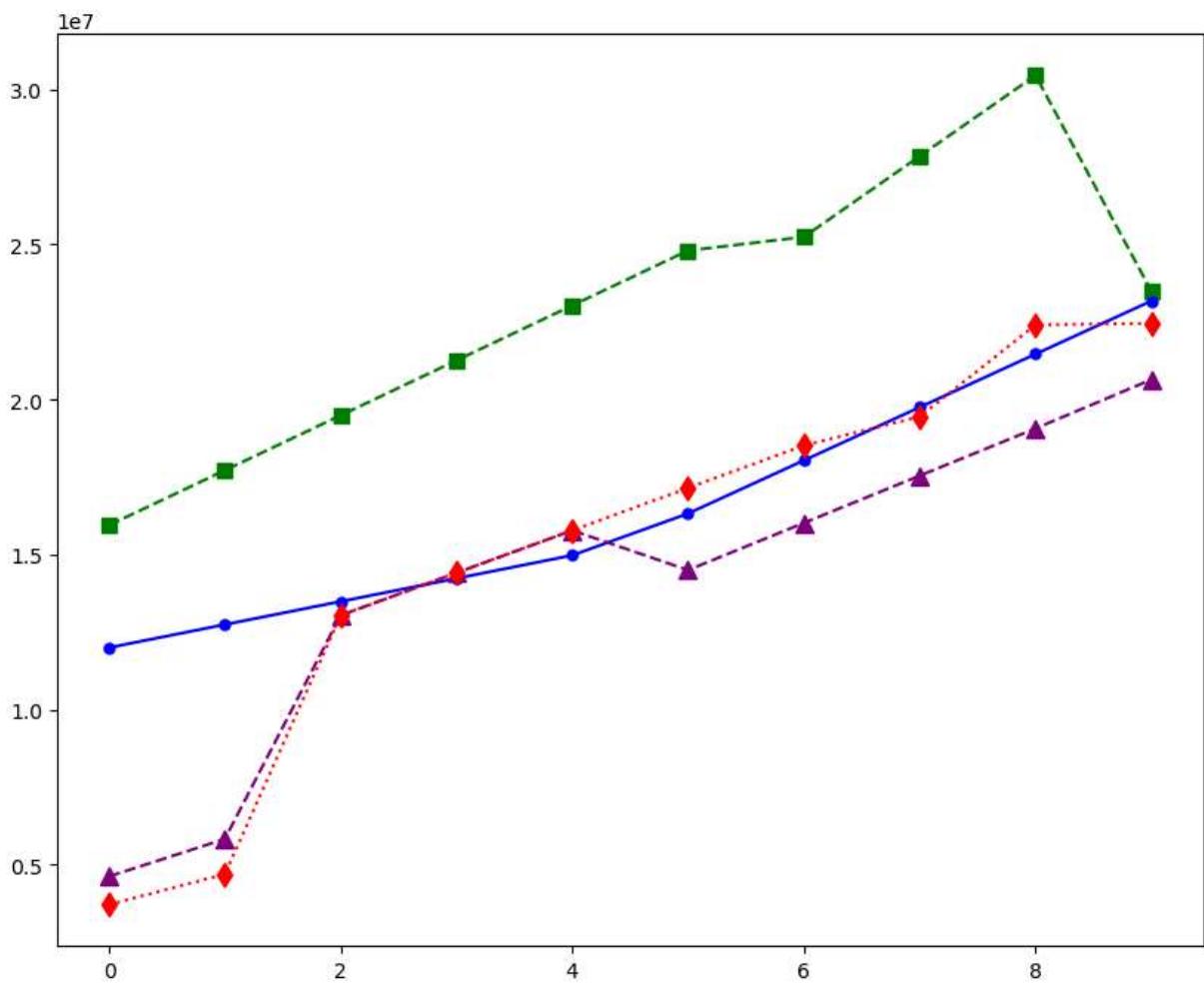
```
In [216]: plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Players[0])
plt.plot(Salary[1], c='Blue', ls = '--', marker = 'o', ms = 5, label = Players[1])
plt.plot(Salary[2], c='purple', ls = '--', marker = '^', ms = 8, label = Players[2])
plt.plot(Salary[3], c='Red', ls = ':', marker = 'd', ms = 8, label = Players[3])

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

plt.show()
```

NameError Traceback (most recent call last)  
Cell In[216], line 6  
3 plt.plot(Salary[2], c='purple', ls = '--', marker = '^', ms = 8, label = Players[2])  
4 plt.plot(Salary[3], c='Red', ls = ':', marker = 'd', ms = 8, label = Players[3])  
--> 6 plt.xticks(list(range(0,10)), Seasons, rotation='vertical')  
8 plt.show()

NameError: name 'Seasons' is not defined



In [217...]

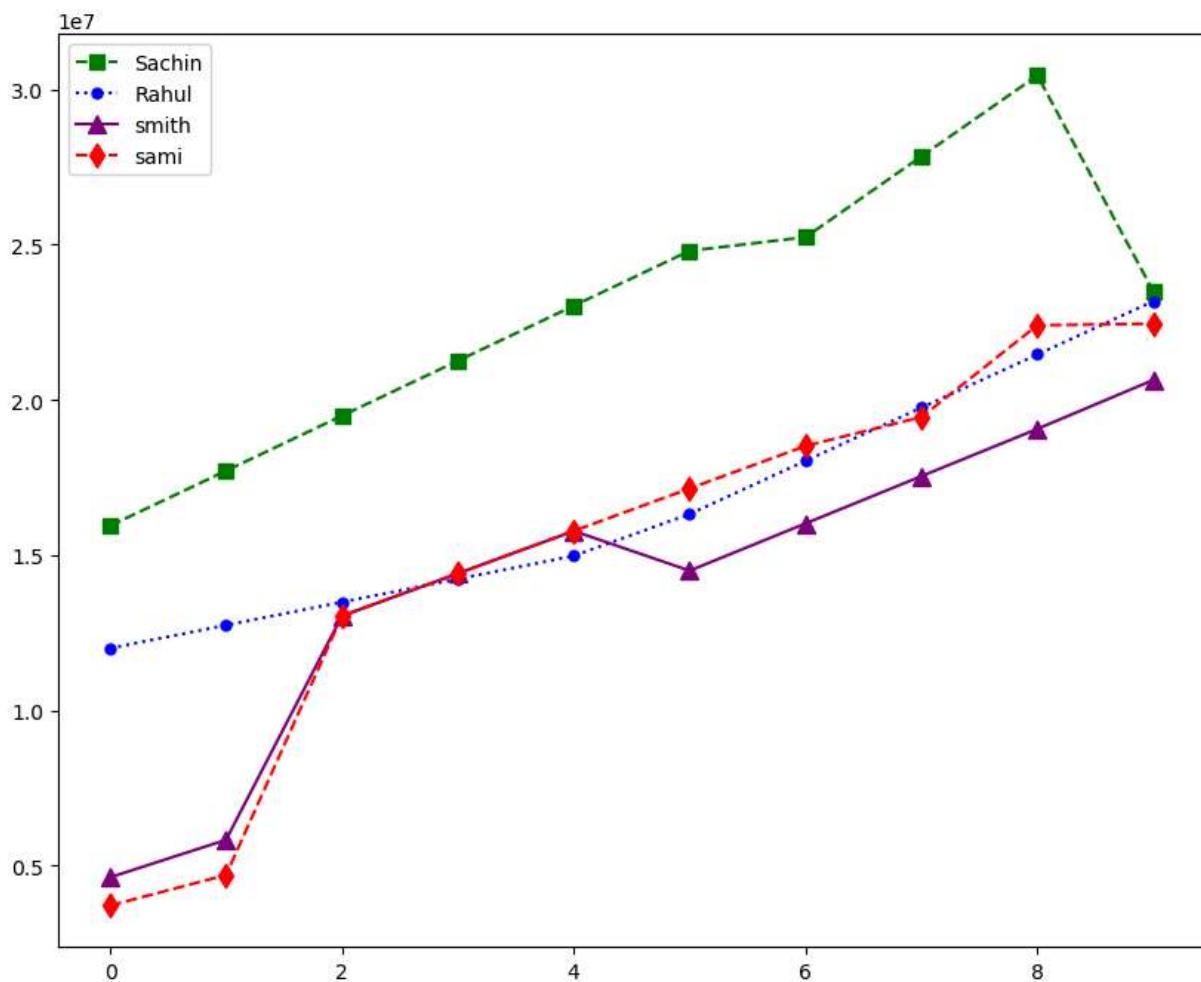
```
# how to add Legned in visualisation

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

plt.show()
```

NameError Traceback (most recent call last)  
Cell In[217], line 8  
6 plt.plot(Salary[3], c='Red', ls = '--', marker = 'd', ms = 8, label = Player  
s[3])  
7 plt.legend()  
----> 8 plt.xticks(list(range(0,10)), Seasons, rotation='vertical')  
10 plt.show()

NameError: name 'Seasons' is not defined



```
In [218]: plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Players[0])
plt.plot(Salary[1], c='Blue', ls = '--', marker = 'o', ms = 5, label = Players[1])
plt.plot(Salary[2], c='purple', ls = '--', marker = '^', ms = 8, label = Players[2])
plt.plot(Salary[3], c='Red', ls = '--', marker = 'd', ms = 8, label = Players[3])
plt.legend(loc = 'upper left',bbox_to_anchor=(0,0) )
plt.xticks(list(range(0,10)), Seasons,rotation='vertical')

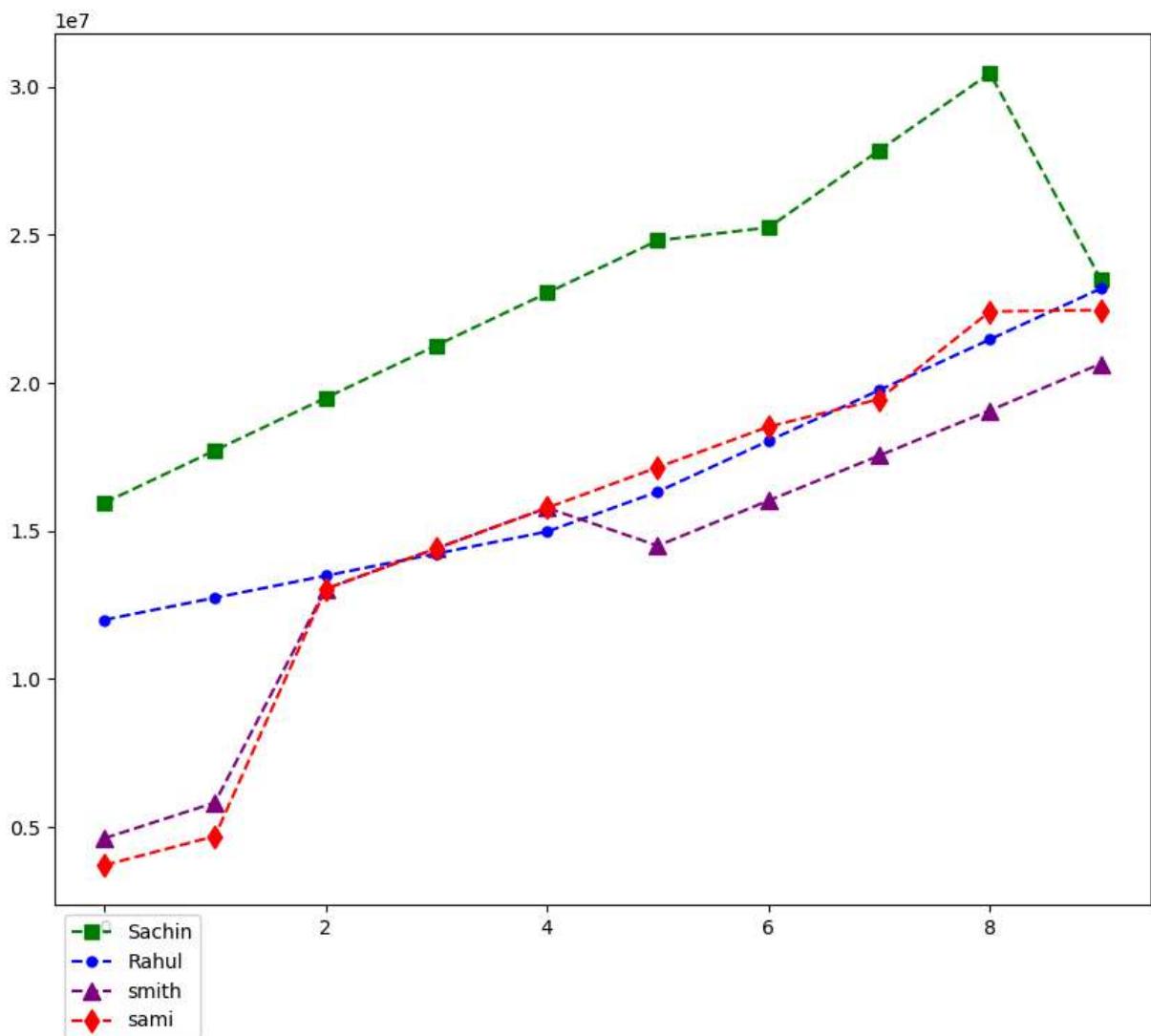
plt.show()
```

NameError Traceback (most recent call last)

Cell In[218], line 6

```
4 plt.plot(Salary[3], c='Red', ls = '--', marker = 'd', ms = 8, label = Player
s[3])
5 plt.legend(loc = 'upper left',bbox_to_anchor=(0,0) )
----> 6 plt.xticks(list(range(0,10)), Seasons,rotation='vertical')
8 plt.show()
```

NameError: name 'Seasons' is not defined



In [219]:

```

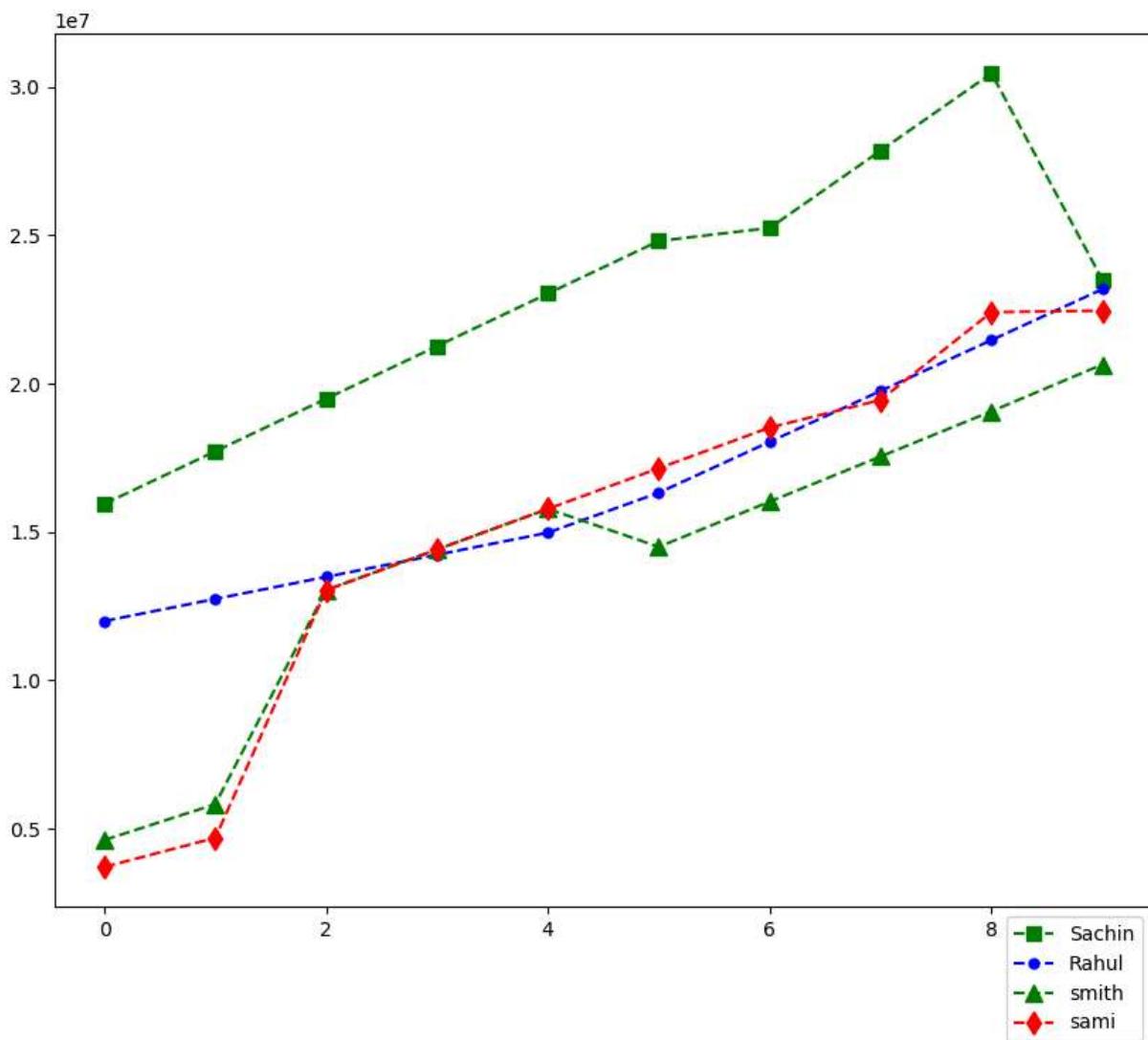
plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Players[0])
plt.plot(Salary[1], c='Blue', ls = '--', marker = 'o', ms = 5, label = Players[1])
plt.plot(Salary[2], c='Green', ls = '--', marker = '^', ms = 8, label = Players[2])
plt.plot(Salary[3], c='Red', ls = '--', marker = 'd',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

plt.show()

```

NameError Traceback (most recent call last)  
Cell In[219], line 6  
4 plt.plot(Salary[3], c='Red', ls = '--', marker = 'd', ms = 8, label = Player  
s[3])  
5 plt.legend(loc = 'upper right',bbox\_to\_anchor=(1,0) )  
----> 6 plt.xticks(list(range(0,10)), Seasons, rotation='vertical')  
8 plt.show()

NameError: name 'Seasons' is not defined



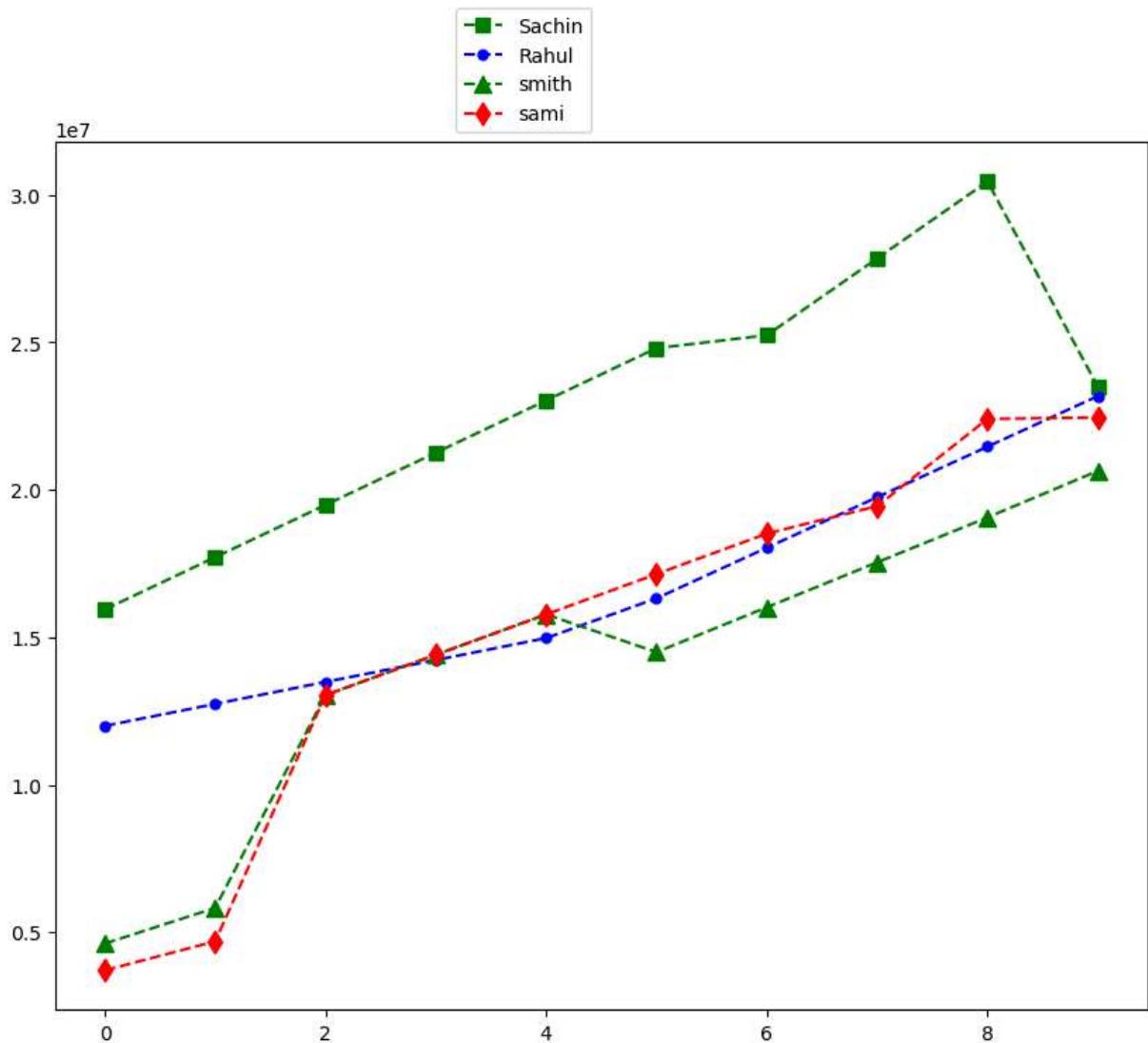
```
In [220]: plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Players[0])
plt.plot(Salary[1], c='Blue', ls = '--', marker = 'o', ms = 5, label = Players[1])
plt.plot(Salary[2], c='Green', ls = '--', marker = '^', ms = 8, label = Players[2])
plt.plot(Salary[3], c='Red', ls = '--', marker = 'd', ms = 8, label = Players[3])
plt.legend(loc = 'lower right',bbox_to_anchor=(0.5,1) )
plt.xticks(list(range(0,10)), Seasons, rotation='vertical')

plt.show()
```

NameError Traceback (most recent call last)

```
Cell In[220], line 6
      4 plt.plot(Salary[3], c='Red', ls = '--', marker = 'd', ms = 8, label = Player
      5 s[3])
      6 plt.legend(loc = 'lower right',bbox_to_anchor=(0.5,1) )
--> 7 plt.xticks(list(range(0,10)), Seasons, rotation='vertical')
      8 plt.show()

NameError: name 'Seasons' is not defined
```



In [221]:

```

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[221], line 12
    9 plt.plot(Salary[8], c='Red', ls = '--', marker = 's', ms = 7, label = Player
   10 s[8])
   11 plt.plot(Salary[9], c='Red', ls = '--', marker = 'o', ms = 7, label = Player
   12 s[9])
--> 12 plt.legend(loc = 'lower right',bbox_to_anchor=(0.5,1) )
   13 plt.xticks(list(range(0,10)), Seasons, rotation='vertical')
   14 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, reverse, scatterpoints, scatteryoffsets, prop, fontsize, labelcolor, borderpad, labels_pacing, handlelength, handleheight, handletextpad, borderaxespad, columnspacing, nocols, mode, fancybox, shadow, title, title_fontsize, framealpha, edgecolor, facecolor, bbox_to_anchor, bbox_transform, frameon, handler_map, title_fontproperties, alignment, 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]

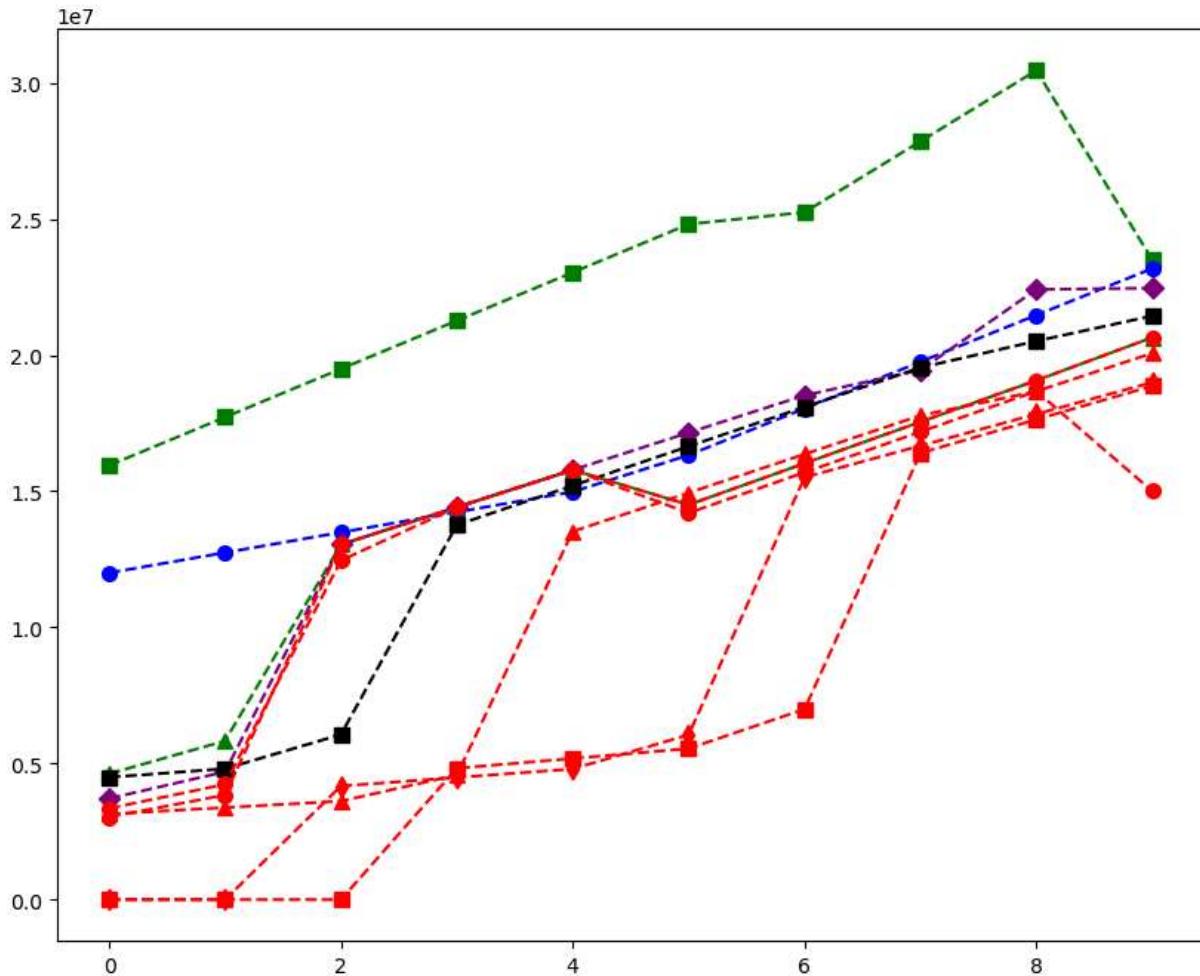
```

```

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**: 'lower 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 [222]...

```
# 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])

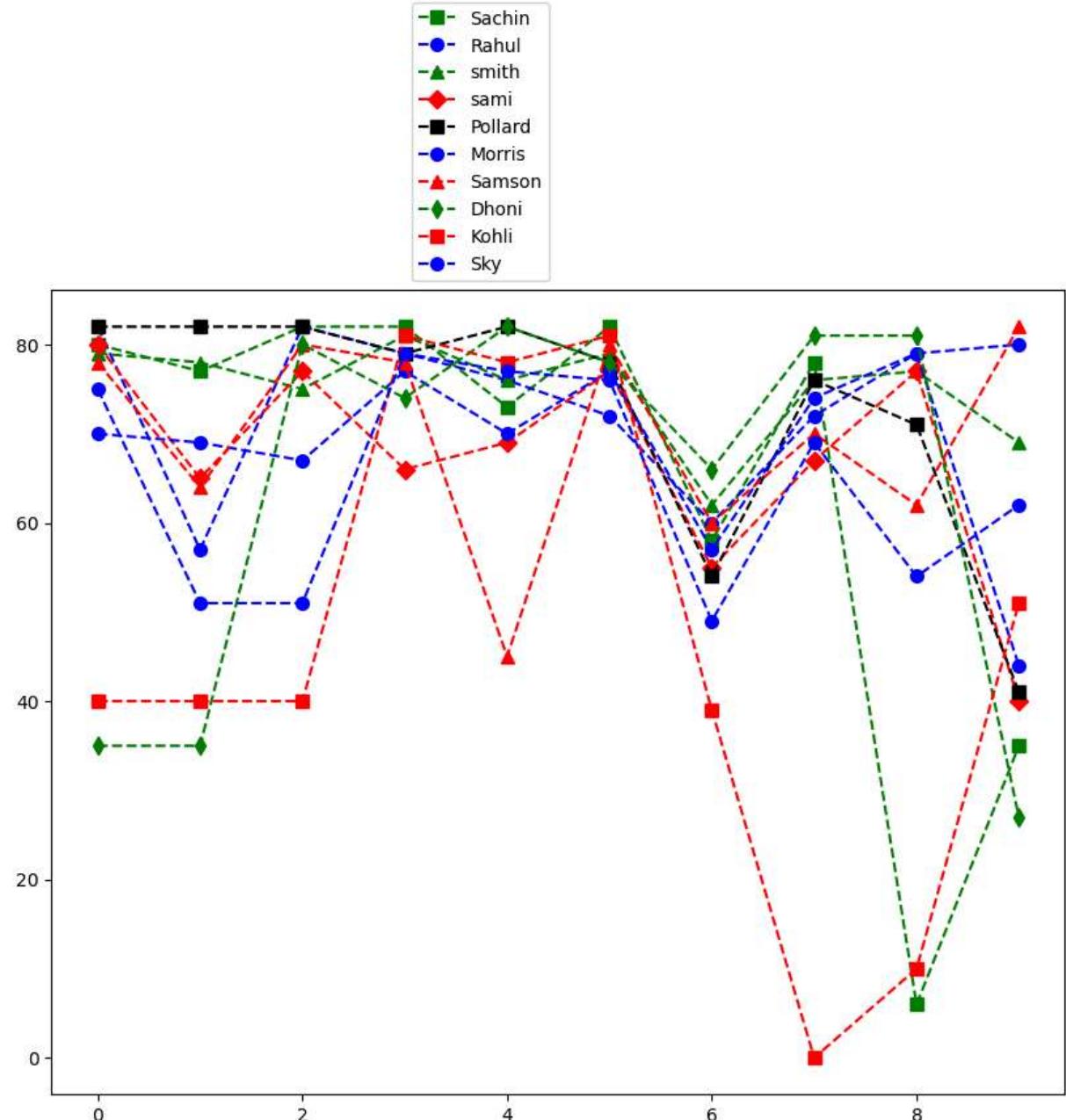
plt.legend(loc = 'lower right', bbox_to_anchor=(0.5,1))
plt.xticks(list(range(0,10)), Seasons, rotation='vertical')

plt.show()

```

```
NameError
Cell In[222], line 15
    12 plt.plot(Games[9], c='Blue', ls = '--', marker = 'o', ms = 7, label = Players[9])
    13 plt.legend(loc = 'lower right',bbox_to_anchor=(0.5,1) )
--> 14 plt.xticks(list(range(0,10)), Seasons,rotation='vertical')
    15 plt.show()
```

NameError: name 'Seasons' is not defined



- In this section we learned - 1>Matrices 2>Building matrices -

np.reshape 3>Dictionaried in python(order doesnot mater)(keys & values)4>visualizaing using pyplot 5>Basket ball analysis

In [ ]:

In [ ]: