In [83]:

```
#Import numpy
import numpy as np

#Seasons
Seasons = ["2010","2011","2012","2013","2014","2015","2016","2017","2018","2
Sdict = {"2010":0,"2011":1,"2012":2,"2013":3,"2014":4,"2015":5,"2016":6,"201

#Players
Players = ["Sachin","Rahul","Smith","Sami","Pollard","Morris","Samson","Dhor
Pdict = {"Sachin":0,"Rahul":1,"Smith":2,"Sami":3,"Pollard":4,"Morris":5,"Sam

#Salaries
Sachin_Salary = [15946875,17718750,19490625,21262500,23034375,24806250,25244
Rahul_Salary = [12000000,12744189,13488377,14232567,14976754,16324500,180385
Smith_Salary = [4621800,5828090,13041250,14410581,15779912,14500000,16022500
Sami_Salary = [3713640,4694041,13041250,14410581,15779912,17149243,18518574,
Pollard_Salary = [4493160,4806720,6061274,13758000,15202590,16647180,1809177
Morris_Salary = [3348000,4235220,12455000,14410581,15779912,14500000,160225
Samson_Salary = [3144240,3380160,3615960,4574189,13520500,14940153,16359805,
Dhoni_Salary = [0,0,4171200,4484040,4796880,6053663,15506632,16669630,178326
Kohli_Salary = [0,0,0,4822800,5184480,5546160,6993708,16402500,17632688,1886
Sky_Salary = [3031920,3841443,13041250,14410581,15779912,14200000,15691000,1
#Matrix
Salary = np.array([Sachin_Salary, Rahul_Salary, Smith_Salary, Sami_Salary, F

#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, S

#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,
```

In [84]:
```
Salary # matrix format
```

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]])
```

Building your first matrix

Games

In [85]:
```
Games
```

Out[85]:
```
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 [86]:
```
Points
```

Out[86]:
```
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 [87]:
```
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 [88]:
```
np.reshape(mydata,(4,5))
```

Out[88]:
```
array([[ 0,  1,  2,  3,  4],
       [ 5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14],
       [15, 16, 17, 18, 19]])
```

In [89]:
```
mydata
```

Out[89]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19])
```

np.reshape(mydata,(5,4),order = 'c')

In [90]:
```
MATRIX1= np.reshape(mydata,(5,4), order = 'c')
MATRIX1
```

Out[90]:
```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11],
       [12, 13, 14, 15],
       [16, 17, 18, 19]])
```

In [91]:
```
MATRIX1.T
```

Out[91]:
```
array([[ 0,  4,  8, 12, 16],
       [ 1,  5,  9, 13, 17],
       [ 2,  6, 10, 14, 18],
       [ 3,  7, 11, 15, 19]])
```

In [92]:
```
MATRIX2= np.reshape(mydata, (5,4), order = 'F')
MATRIX2
```

Out[92]:
```
array([[ 0,  5, 10, 15],
       [ 1,  6, 11, 16],
       [ 2,  7, 12, 17],
       [ 3,  8, 13, 18],
       [ 4,  9, 14, 19]])
```

In [93]:
```
MATRIX3 = np.reshape(mydata,(5,4), order = 'A')
MATRIX3
```

Out[93]:
```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11],
       [12, 13, 14, 15],
       [16, 17, 18, 19]])
```

If i want to get only no.3

In [94]:
```
MATRIX1[4,3]
```

Out[94]:
```
19
```

```
In [95]: MATRIX1[4,2]
```

```
Out[95]: 18
```

```
In [96]: MATRIX3[4,3]
```

```
Out[96]: 19
```

```
In [97]: MATRIX2[4,2]
```

```
Out[97]: 14
```

```
In [98]: MATRIX2[4,1]
```

```
Out[98]: 9
```

```
In [99]: MATRIX1
```

```
Out[99]: array([[ 0,  1,  2,  3],
                [ 4,  5,  6,  7],
                [ 8,  9, 10, 11],
                [12, 13, 14, 15],
                [16, 17, 18, 19]])
```

```
In [100]: MATRIX1[-3,-2]
```

```
Out[100]: 10
```

```
In [101]: MATRIX1[-3,-1]
```

```
Out[101]: 11
```

```
In [102]: MATRIX1[-1,-1]
```

```
Out[102]: 19
```

```
In [103]: MATRIX1[-5,-1]
```

```
Out[103]: 3
```

```
In [104]: MATRIX1[-5,0]
```

```
Out[104]: 0
```

```
In [105]: MATRIX1[-5,-4]
```

```
Out[105]: 0
```

```
In [106]: MATRIX1[-3,-4]
```

```
Out[106]: 8
```

```
In [107]: MATRIX1[-2,-1]
```

```
Out[107]: 15
```

```
In [108]: MATRIX1[-2,-3]

Out[108]: 13

In [109]: MATRIX1[-2,-4]

Out[109]: 12

In [110]: mydata

Out[110]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
                 17, 18, 19])

In [111]: MATRIX2[0:2]

Out[111]: array([[ 0,  5, 10, 15],
                 [ 1,  6, 11, 16]])

In [112]: MATRIX2

Out[112]: array([[ 0,  5, 10, 15],
                 [ 1,  6, 11, 16],
                 [ 2,  7, 12, 17],
                 [ 3,  8, 13, 18],
                 [ 4,  9, 14, 19]])

In [113]: MATRIX2[1:2]

Out[113]: array([[ 1,  6, 11, 16]])

In [114]: MATRIX2[-2,-1]

Out[114]: 18

In [115]: MATRIX2[1:3]

Out[115]: array([[ 1,  6, 11, 16],
                 [ 2,  7, 12, 17]])

In [116]: MATRIX2[1:2,1:3]

Out[116]: array([[ 6, 11]])

In [117]: MATRIX2[1:2,]

Out[117]: array([[ 1,  6, 11, 16]])

In [118]: MATRIX2[-3,3]

Out[118]: 17

In [119]: MATRIX2[-3,-3]

Out[119]: 7
```

In [120]:
```
MATRIX2
```

Out[120]:
```
array([[ 0,  5, 10, 15],
       [ 1,  6, 11, 16],
       [ 2,  7, 12, 17],
       [ 3,  8, 13, 18],
       [ 4,  9, 14, 19]])
```

In [121]:
```
MATRIX2[0:2]
```

Out[121]:
```
array([[ 0,  5, 10, 15],
       [ 1,  6, 11, 16]])
```

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]:
```
MATRIX2.shape
```

Out[123]:
```
(5, 4)
```

In [124]:
```
MATRIX3
```

Out[124]:
```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11],
       [12, 13, 14, 15],
       [16, 17, 18, 19]])
```

In [125]:
```
MATRIX2
```

Out[125]:
```
array([[ 0,  5, 10, 15],
       [ 1,  6, 11, 16],
       [ 2,  7, 12, 17],
       [ 3,  8, 13, 18],
       [ 4,  9, 14, 19]])
```

In [126]:
```
MATRIX1
```

Out[126]:
```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11],
       [12, 13, 14, 15],
       [16, 17, 18, 19]])
```

In [127]:
```
a1 = ['welcome', 'to', 'datascience']
```

In [128]:
```
a2 = ['required','hard','work']
```

In [129]:
```
a3 = [1,2,3]
```

In [130]:
```
[a1,a2,a3]
```

Out[130]:
```
[['welcome', 'to', 'datascience'], ['required', 'hard', 'work'], [1, 2,
3]]
```

In [131]:
```
np.array([a1,a2,a3])
```

Out[131]:
```
array([['welcome', 'to', 'datascience'],
       ['required', 'hard', 'work'],
       ['1', '2', '3']], dtype='<U11')
```

In [132]:
```
Games
```

Out[132]:
```
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 [133]:
```
Games[0]
```

Out[133]:
```
array([80, 77, 82, 82, 73, 82, 58, 78,  6, 35])
```

In [134]:
```
Games[5]
```

Out[134]:
```
array([70, 69, 67, 77, 70, 77, 57, 74, 79, 44])
```

In [135]:
```
Games[0:5]
```

Out[135]:
```
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 [136]:
```
Games[0,5]
```

Out[136]:
```
82
```

In [137]:
```
Games[0,2]
```

Out[137]:
```
82
```

In [138]:
```
Games
```

Out[138]:
```
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 [139]:
```
Games[0:2]
```

Out[139]:
```
array([[80, 77, 82, 82, 73, 82, 58, 78,  6, 35],
       [82, 57, 82, 79, 76, 72, 60, 72, 79, 80]])
```

In [140]:
```
Games
```

Out[140]:
```
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 [141]:
```
Games[1:2]
```

Out[141]:
```
array([[82, 57, 82, 79, 76, 72, 60, 72, 79, 80]])
```

In [142]:
```
Games[2]
```

Out[142]:
```
array([79, 78, 75, 81, 76, 79, 62, 76, 77, 69])
```

In [143]:
```
Games
```

Out[143]:
```
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 [144]:
```
Games
```

Out[144]:
```
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 [145]:
```
Games[-3,-1]
```

Out[145]:
```
27
```

In [146]:
```
Games[-3:-1]
```

Out[146]:
```
array([[35, 35, 80, 74, 82, 78, 66, 81, 81, 27],
       [40, 40, 40, 81, 78, 81, 39,  0, 10, 51]])
```

In [147]:
```
Games[3,-1]
```

Out[147]:
```
40
```

In [148]:
```
Games[-3,-1]
```

Out[148]:
```
27
```

In [149]:
```
Points
```

Out[149]:
```
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 [150]:
```
Points[0]
```

Out[150]:
```
array([2832, 2430, 2323, 2201, 1970, 2078, 1616, 2133,   83,  782])
```

In [151]:
```
Points
```

Out[151]:
```
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 [152]:
```
Points[6,1]
```

Out[152]:
```
1104
```

In [153]:
```
Points[3:6]
```

Out[153]:
```
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 [154]:
```
Points
```

Out[154]:
```
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 [155]:
```
Points[-6,-1]
```

Out[155]:
```
646
```

In [156]:
```
# =====DICTIONARY=====#

 # dict does not maintain the order
```

In [157]:
```
dict1 = {'key1' : 'val1', 'key2':'val2','key3':'val3'}
```

In [158]:
```
dict1
```

Out[158]:
```
{'key1': 'val1', 'key2': 'val2', 'key3': 'val3'}
```

In [159]:
```
dict1['key2']
```

Out[159]:
```
'val2'
```

In [160]:
```
dict2 = {'bang':2, 'hyd':'we are hear','pune':True}
dict2
```

Out[160]:
```
{'bang': 2, 'hyd': 'we are hear', 'pune': True}
```

In [161]:
```
dict3 = {'Germany':'I have been here', 'France':2,'Spain':True}
```

In [162]:
```
dict3
```

Out[162]:
```
{'Germany': 'I have been here', 'France': 2, 'Spain': True}
```

In [163]:
```
dict3['Germany']
```

Out[163]:
```
'I have been here'
```

In [164]:
```
# if you check theat dataset seasons & players are dictionary type of data
# if you look at the pdict players names are key part:nos are the values
# dictionary can guide us which player at which level and which row
# main advantage of the dictionary is we dont required to count which
 no row which players are sitting
```

```
  Cell In[164], line 5
    no row which players are sitting
    ^
IndentationError: unexpected indent
```

In [165]:
```
Games
```

Out[165]:
```
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 [166]:
```
Pdict
```

Out[166]:
```
{'Sachin': 0,
 'Rahul': 1,
 'Smith': 2,
 'Sami': 3,
 'Pollard': 4,
 'Morris': 5,
 'Samson': 6,
 'Dhoni': 7,
 'Kohli': 8,
 'Sky': 9}
```

In [167]:
```
# how do i know player kobebryant is at
```

In [168]:
```
Pdict['Sachin']
```

Out[168]:
```
0
```

In [169]:
```
Games
```

Out[169]:
```
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 [170]:
```
Pdict['Rahul']
```

Out[170]:
```
1
```

In [171]:
```
Games[1]
```

Out[171]:
```
array([82, 57, 82, 79, 76, 72, 60, 72, 79, 80])
```

Games

In [172]:
```
Points
```

Out[172]:
```
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 [173]:
```
Salary
```

Out[173]:
```
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 [174]:
```
Salary[2,4]
```

Out[174]:
```
15779912
```

In [175]:
```
Salary
```

Out[175]:
```
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 [176]:
```
Salary[Pdict['Sky']][Sdict['2019']]
```

Out[176]:
```
15000000
```

In [177]:
```
Salary
```

Out[177]:
```
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 [178]:
```
Games
```

Out[178]:
```
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 [179]:
```
Salary/Games
```

```
C:\Users\RUPA\AppData\Local\Temp\ipykernel_7484\3709746658.py:1: RuntimeWa
rning: divide by zero encountered in divide
  Salary/Games
```

Out[179]:
```
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 [180]:
```
np.round(Salary/Games)
```

```
C:\Users\RUPA\AppData\Local\Temp\ipykernel_7484\3232172828.py:1: RuntimeWa
rning: divide by zero encountered in divide
  np.round(Salary/Games)
```

Out[180]:
```
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 [181]:
```
import warnings
warnings.filterwarnings('ignore')
#np.round(FieldGoals/Games)
#FieldGoals/Games  # this matrix is lot of decimal points yo can not round
#round()
```

In [182]:
```
## --- First visualization ----##
```

In [183]:
```
import numpy as np
import matplotlib.pyplot as plt
```

In [184]:
```
# matplotlib inline # keep the plot inside jupyter nots insted of getting in
```

In [185]:
```
Salary
```

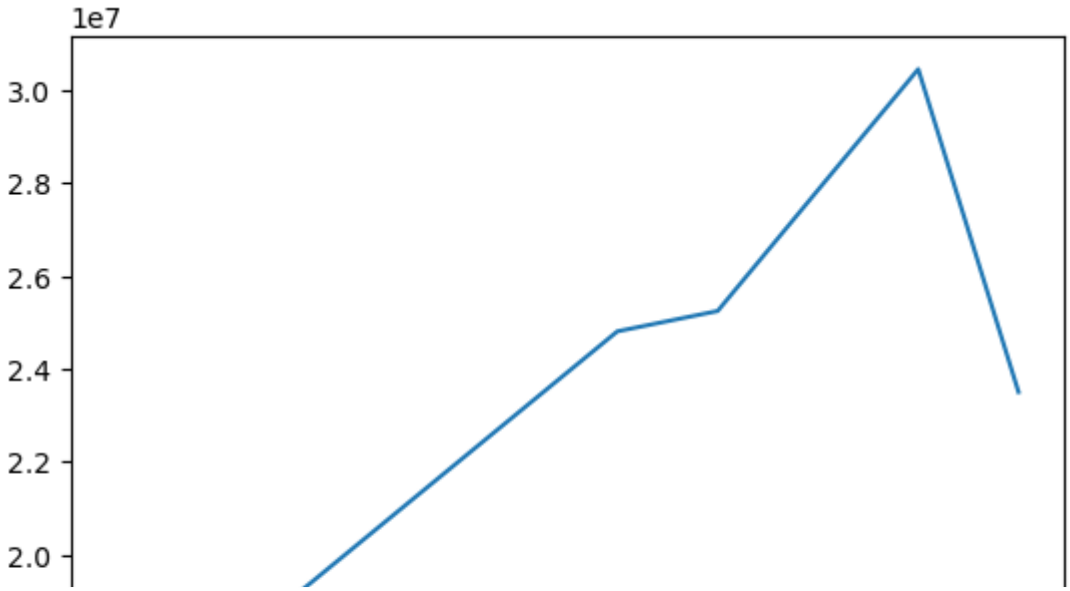
Out[185]:
```
array([[15946875, 17718750, 19490625, 21262500, 23034375, 24806250,
        25244493, 27849149, 30453805, 23500000],
       [12000000, 12744189, 13488377, 14232567, 14976754, 16324500,
        18038573, 19752645, 21466718, 23180790],
       [ 4621800,  5828090, 13041250, 14410581, 15779912, 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,        0,  4822800,  5184480,  5546160,
         6993708, 16402500, 17632688, 18862875],
       [ 3031920,  3841443, 13041250, 14410581, 15779912, 14200000,
        15691000, 17182000, 18673000, 15000000]])
```

In [186]:
```
Salary[0]
```

Out[186]:
```
array([15946875, 17718750, 19490625, 21262500, 23034375, 24806250,
       25244493, 27849149, 30453805, 23500000])
```

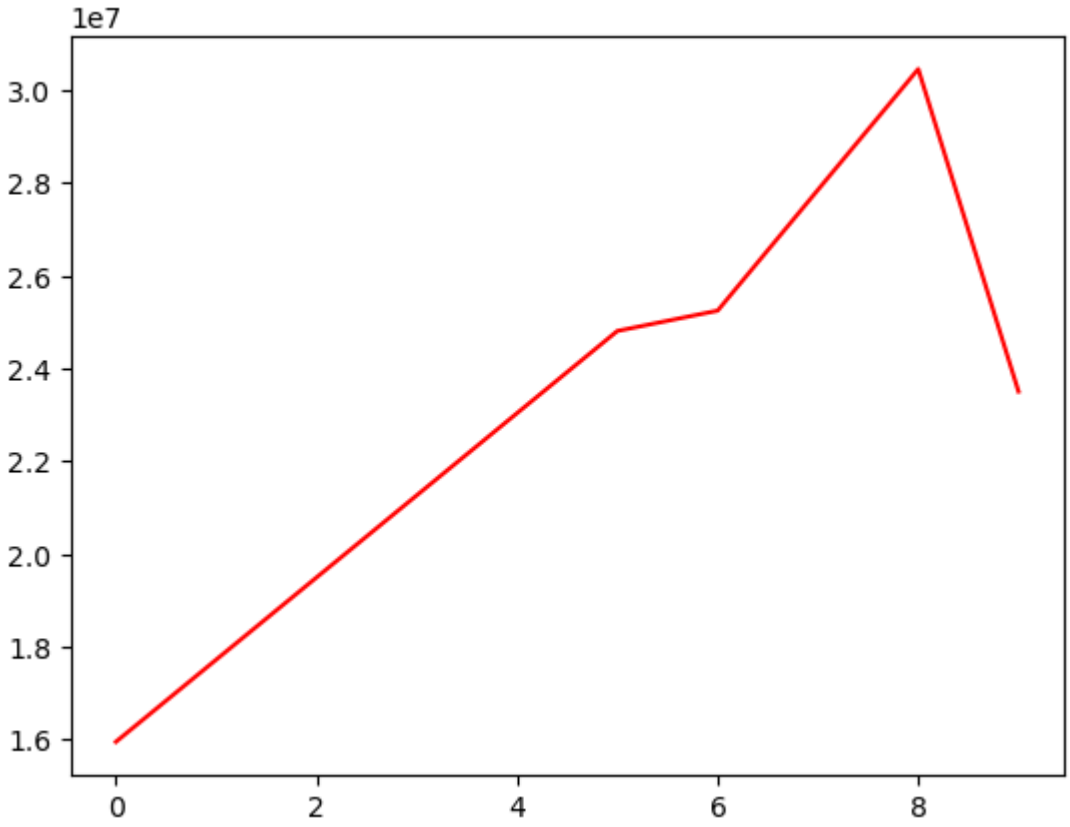
In [187]:
```
plt.plot(Salary[0])
```

Out[187]:
```
[<matplotlib.lines.Line2D at 0x28dc9b44550>]
```



In [188]:
```
plt.plot(Salary[0], c='red')
```

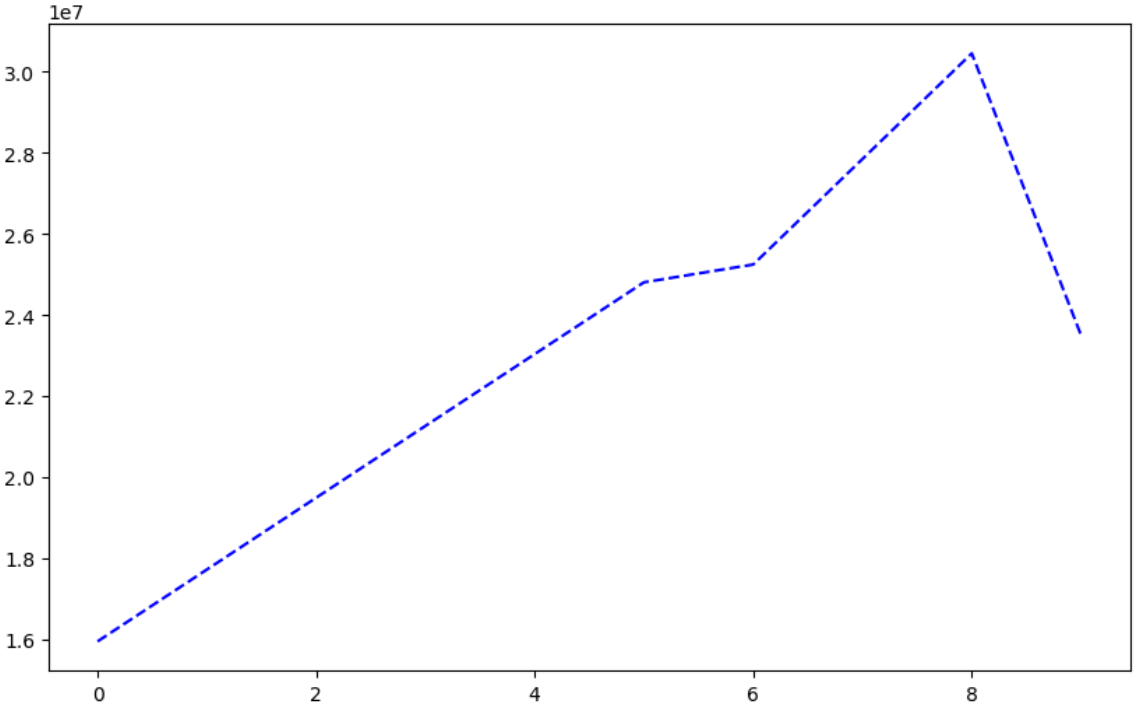
Out[188]:
```
[<matplotlib.lines.Line2D at 0x28dca371e50>]
```



In [189]:
```
%matplotlib inline
plt.rcParams['figure.figsize'] = 10,6
```

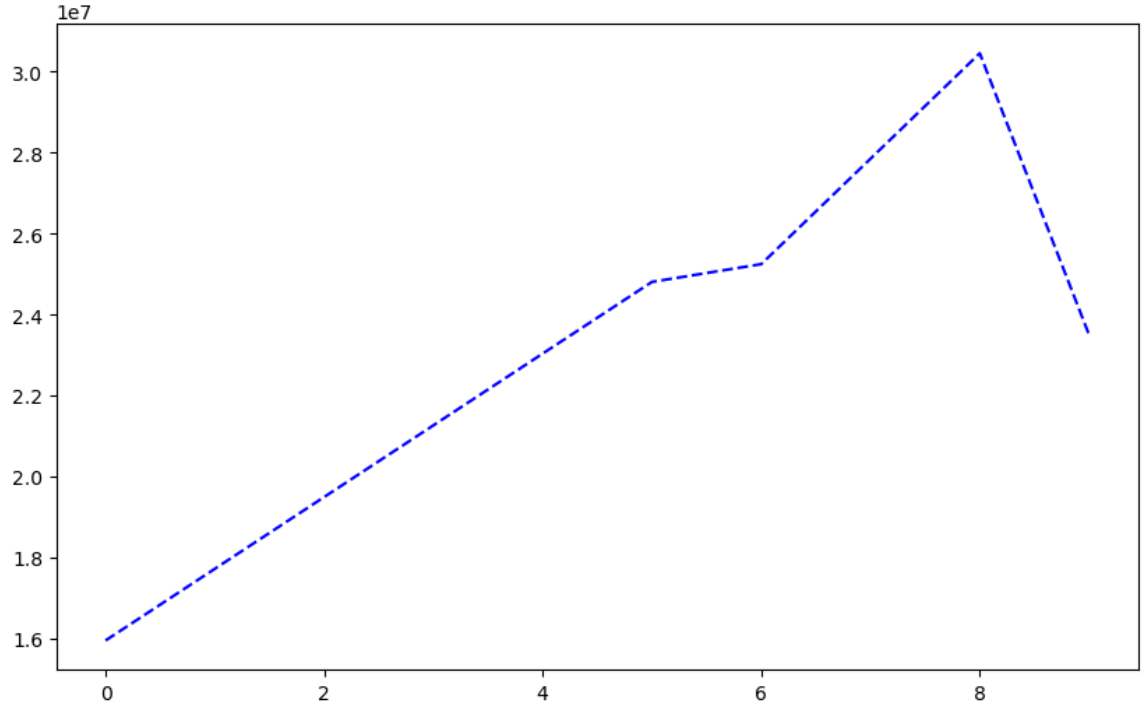
In [190]:
```
plt.plot(Salary[0], c='Blue', ls = 'dashed')
```

Out[190]:
```
[<matplotlib.lines.Line2D at 0x28dc9b21b90>]
```



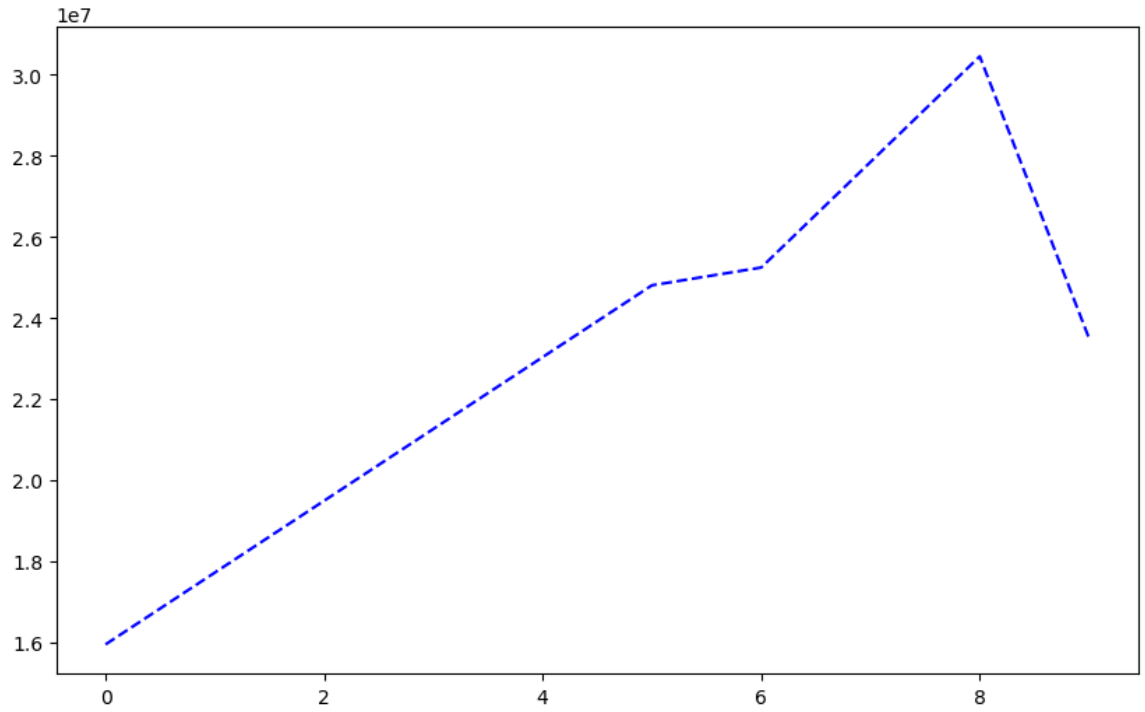
In [191]:
```
plt.plot(Salary[0], c='Blue', ls = '--')
```

Out[191]:
```
[<matplotlib.lines.Line2D at 0x28dc9c28ed0>]
```



In [192]:
```
plt.plot(Salary[0], c='Blue', ls = '--')
```

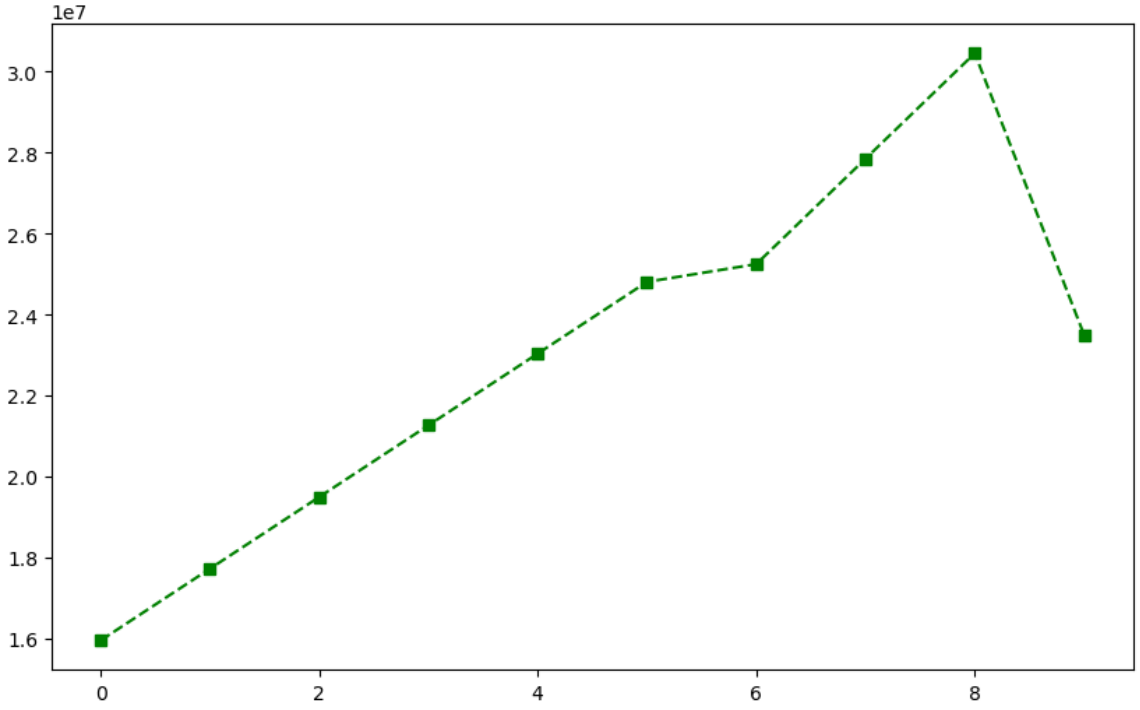
Out[192]:
```
[<matplotlib.lines.Line2D at 0x28dc9c4eed0>]
```



here only single and double dashed is allow

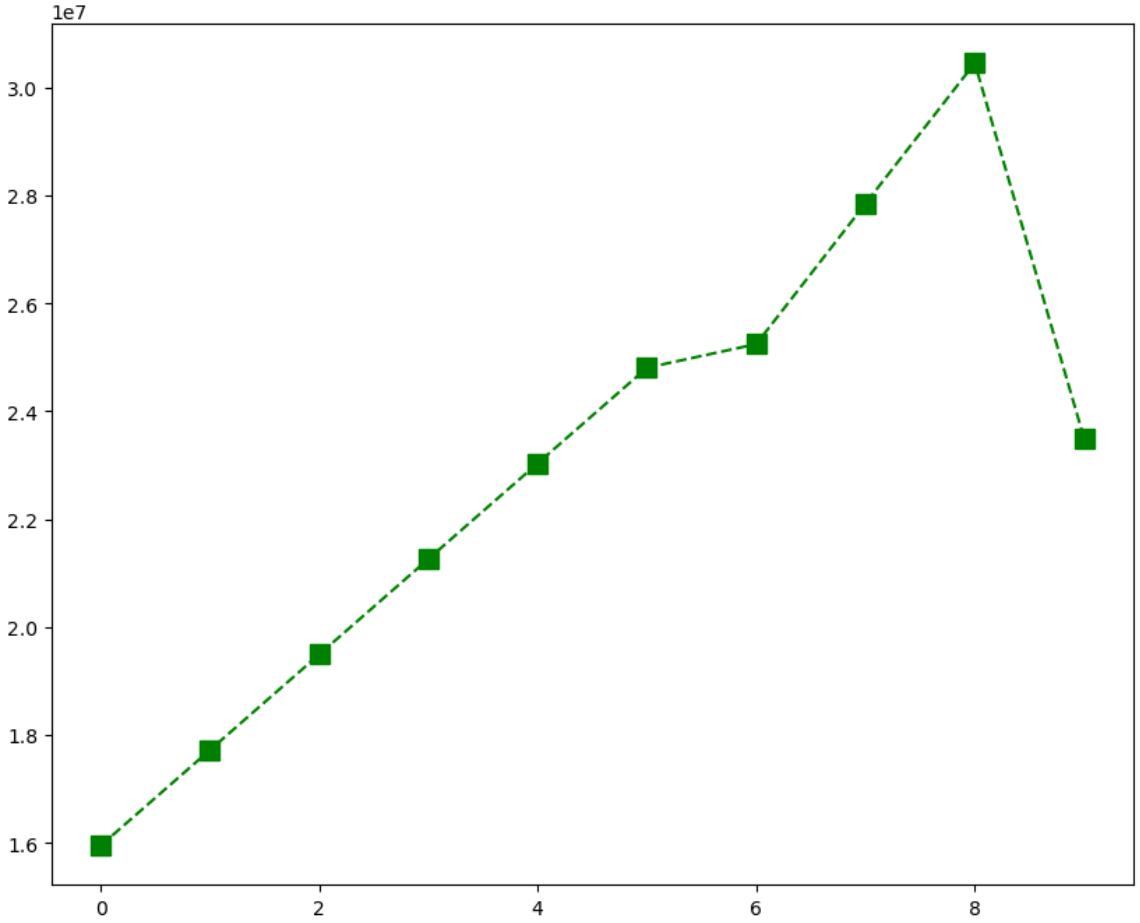
In [193]:
```
plt.plot(Salary[0], c='Green', ls = '--', marker = 's') # s - squares
```

Out[193]:
```
[<matplotlib.lines.Line2D at 0x28dcb471d90>]
```

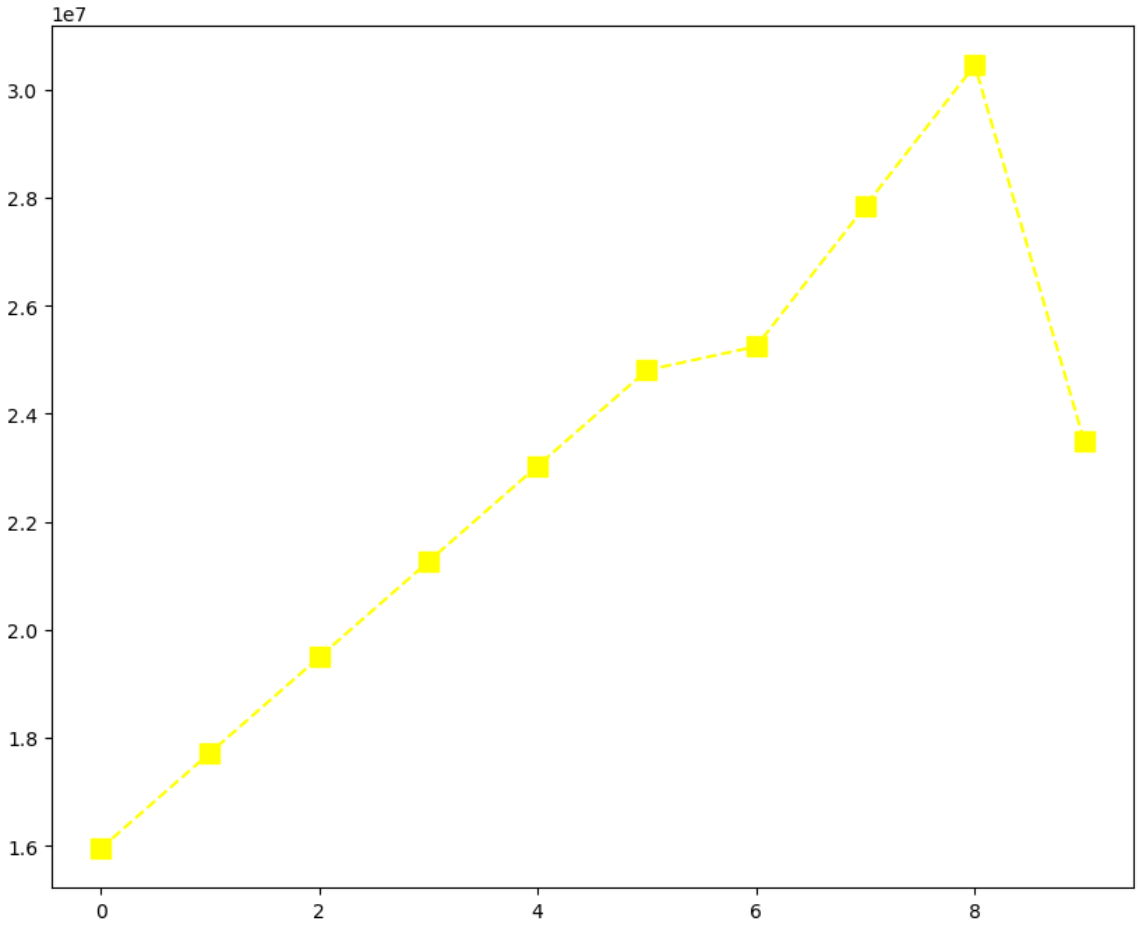


In [194]:
```
%matplotlib inline
plt.rcParams['figure.figsize'] = 10,8 #runtime configuration parameter
```

In [195]:
```
plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 10)
plt.show()
```

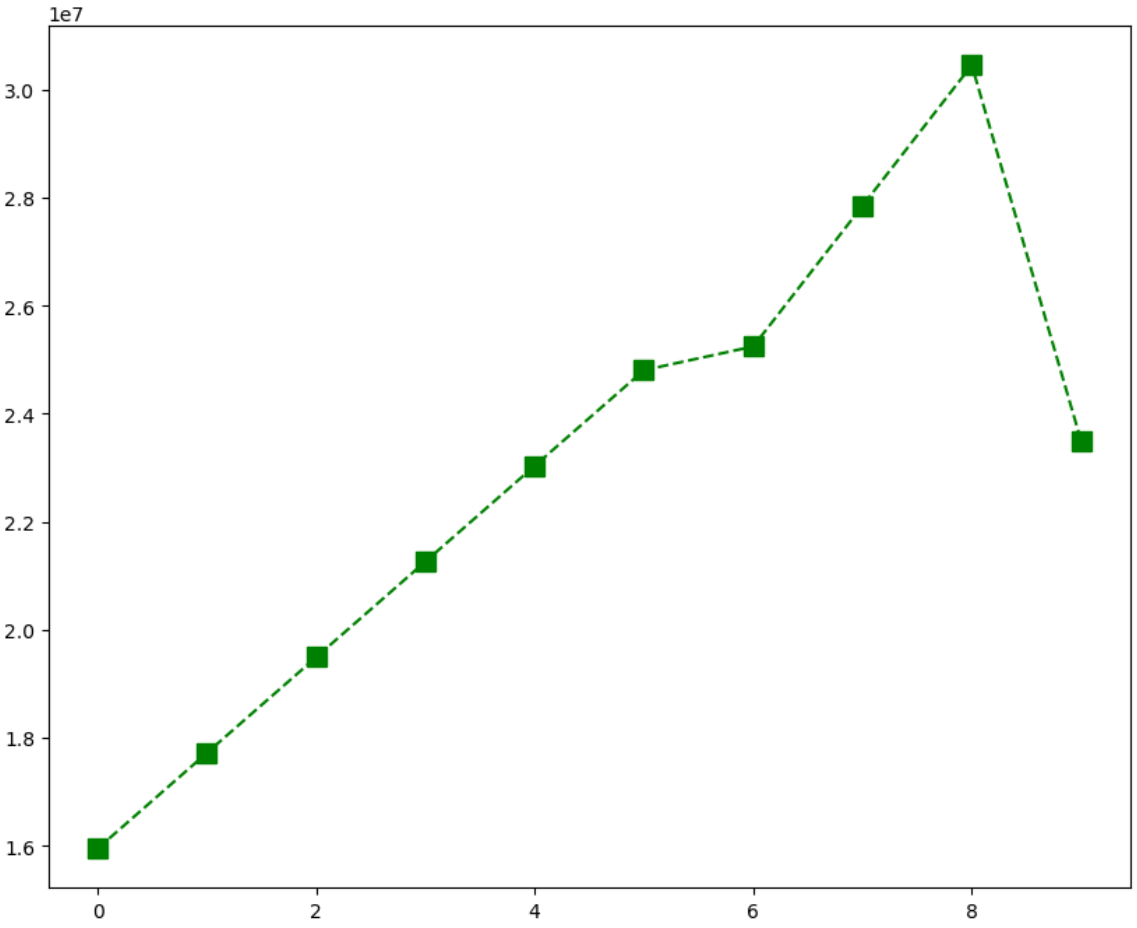


In [196]:
```
plt.plot(Salary[0], c='yellow', ls = '--', marker = 's', ms = 10)
plt.show()
```



In [197]:
```
%matplotlib inline
plt.rcParams['figure.figsize'] = 10,8 #runtime configuration parameter
```

In [198]:
```
plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 10)
plt.show()
```



In [199]:
```
list(range(0,10))
```

Out[199]:
```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

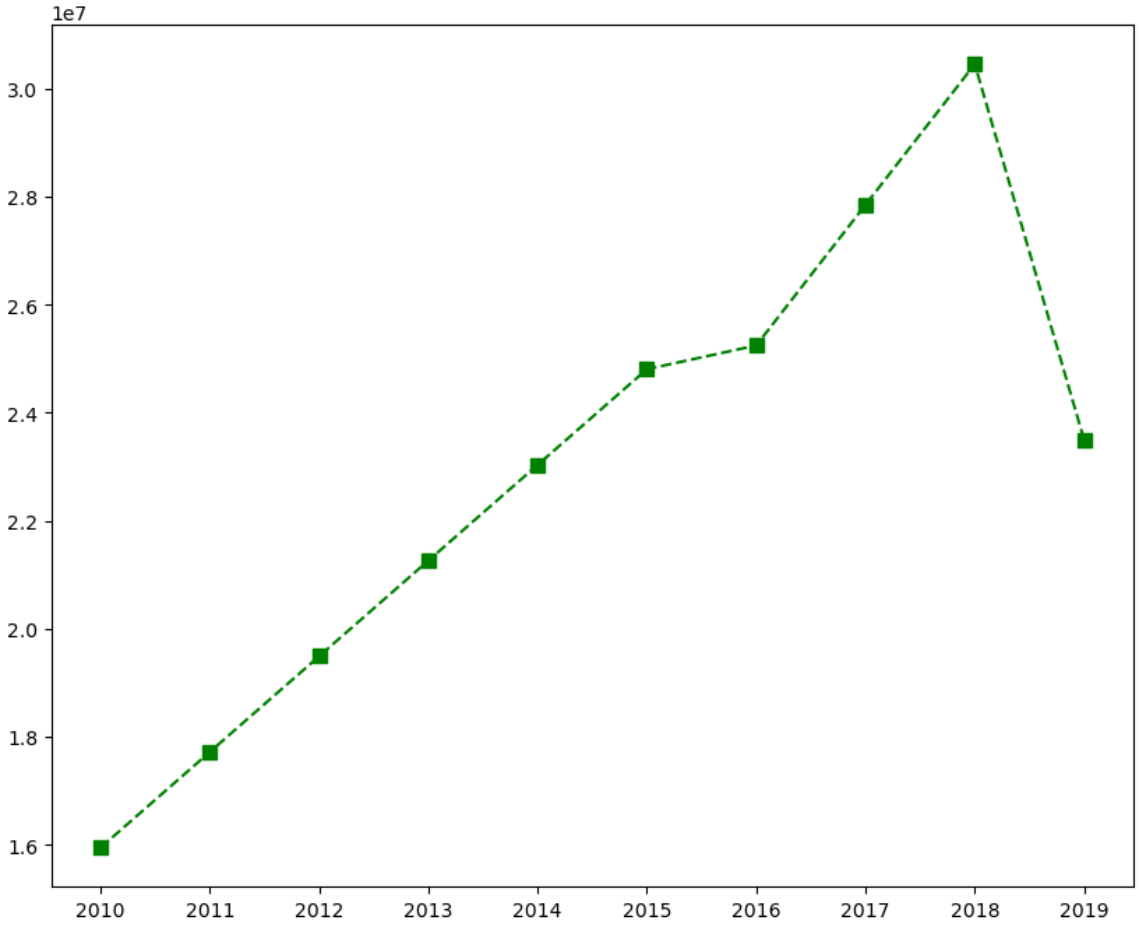
In [200]:
```
Sdict
```

Out[200]:
```
{'2010': 0,
 '2011': 1,
 '2012': 2,
 '2013': 3,
 '2014': 4,
 '2015': 5,
 '2016': 6,
 '2017': 7,
 '2018': 8,
 '2019': 9}
```

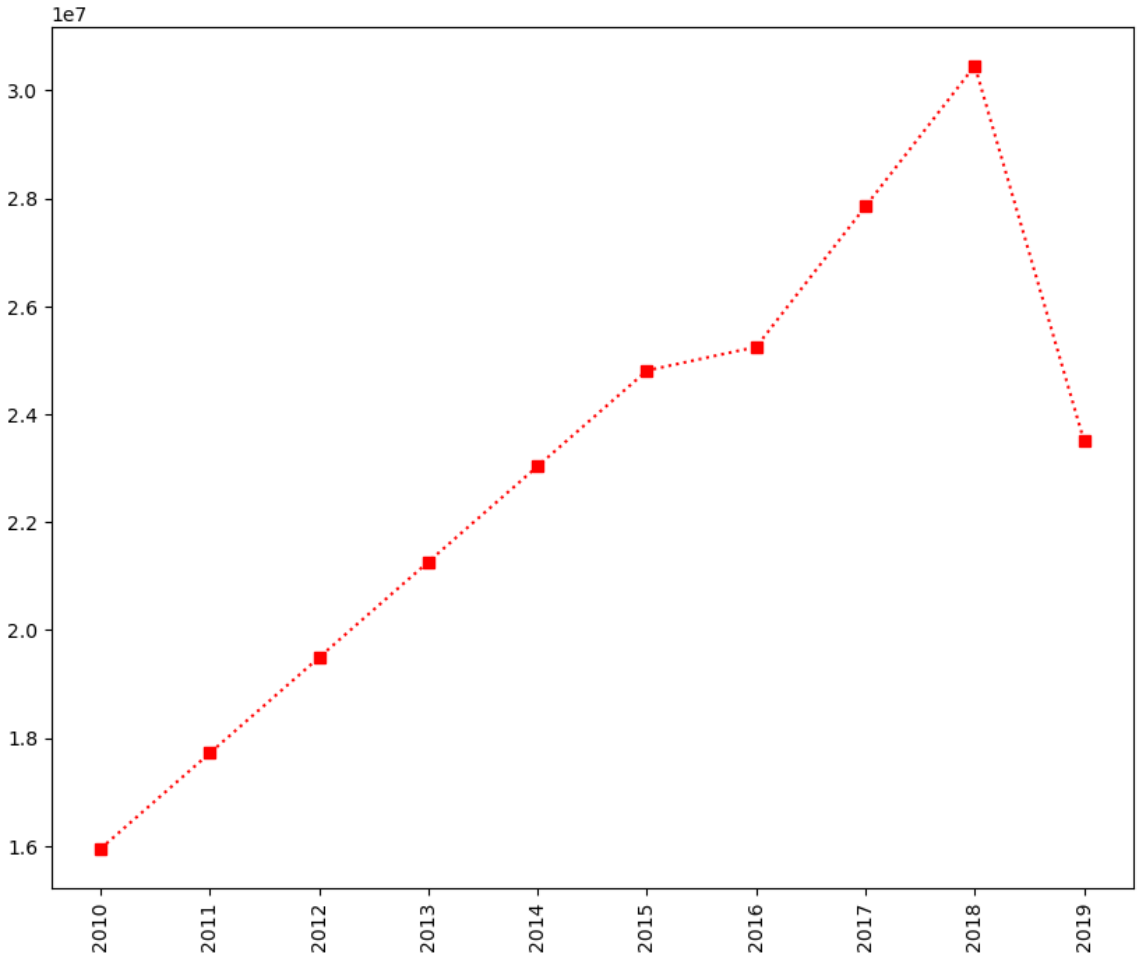
In [201]:
```
Pdict
```

Out[201]:
```
{'Sachin': 0,
 'Rahul': 1,
 'Smith': 2,
 'Sami': 3,
 'Pollard': 4,
 'Morris': 5,
 'Samson': 6,
 'Dhoni': 7,
 'Kohli': 8,
 'Sky': 9}
```

In [202]:
```
plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7)
plt.xticks(list(range(0,10)), Seasons)
plt.show()
```



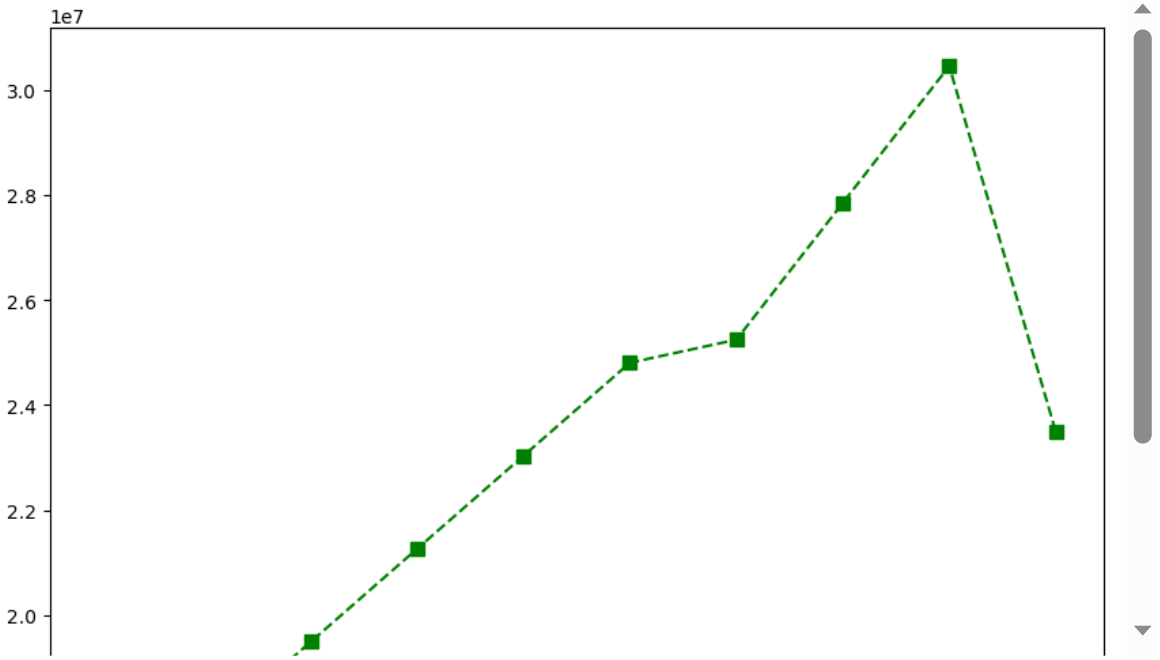
In [203]:
```
plt.plot(Salary[0], c='Red', ls = ':', marker = 's', ms = 6, label = Players
plt.xticks(list(range(0,10)), Seasons,rotation='vertical')
plt.show()
```



In [204]:
```
Games
```

Out[204]:
```
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 [205]:
```
plt.plot(Salary[0], c = 'Green',ls = '--', marker = 's', ms = 7, label = Pla
plt.xticks(list(range(0,10)),Seasons,rotation ='horizontal')
plt.show()
```



In [206]:
```
Salary[0]
```

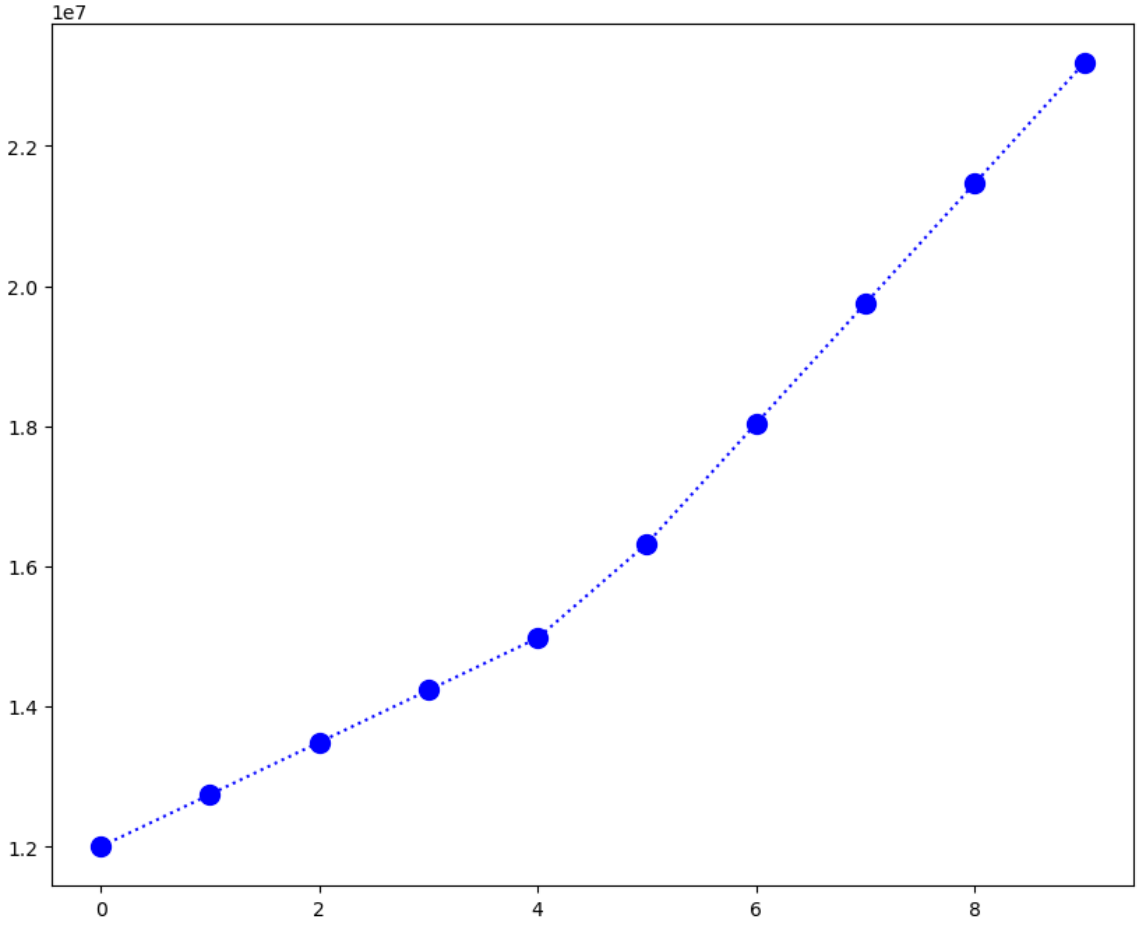
Out[206]:
```
array([15946875, 17718750, 19490625, 21262500, 23034375, 24806250,
       25244493, 27849149, 30453805, 23500000])
```

In [207]:
```
Salary[1]
```

Out[207]:
```
array([12000000, 12744189, 13488377, 14232567, 14976754, 16324500,
       18038573, 19752645, 21466718, 23180790])
```

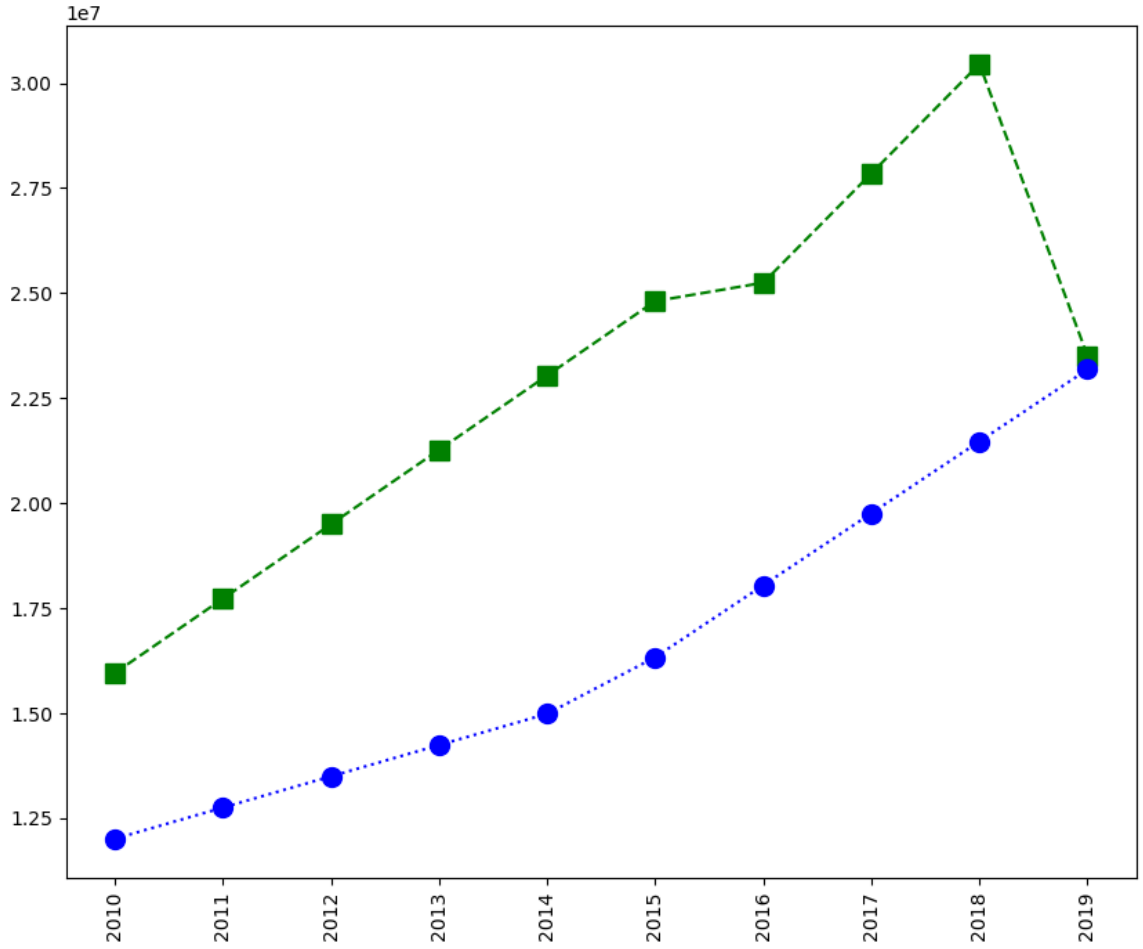
In [208]:
```
plt.plot(Salary[1], c='Blue', ls = ':', marker = 'o', ms = 10, label = Playe
```

Out[208]:
```
[<matplotlib.lines.Line2D at 0x28dcb5c4550>]
```

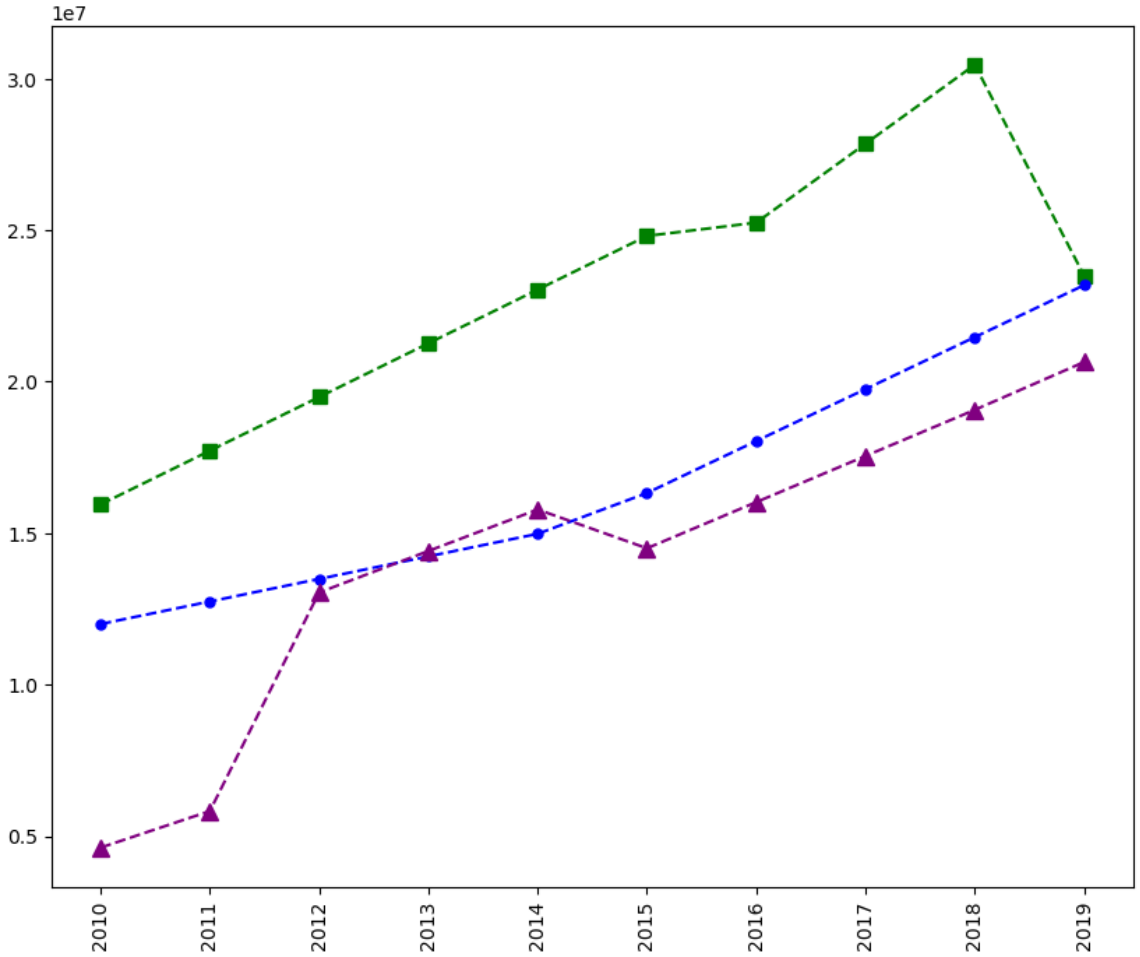


In [209]:
```
# More visualization
```

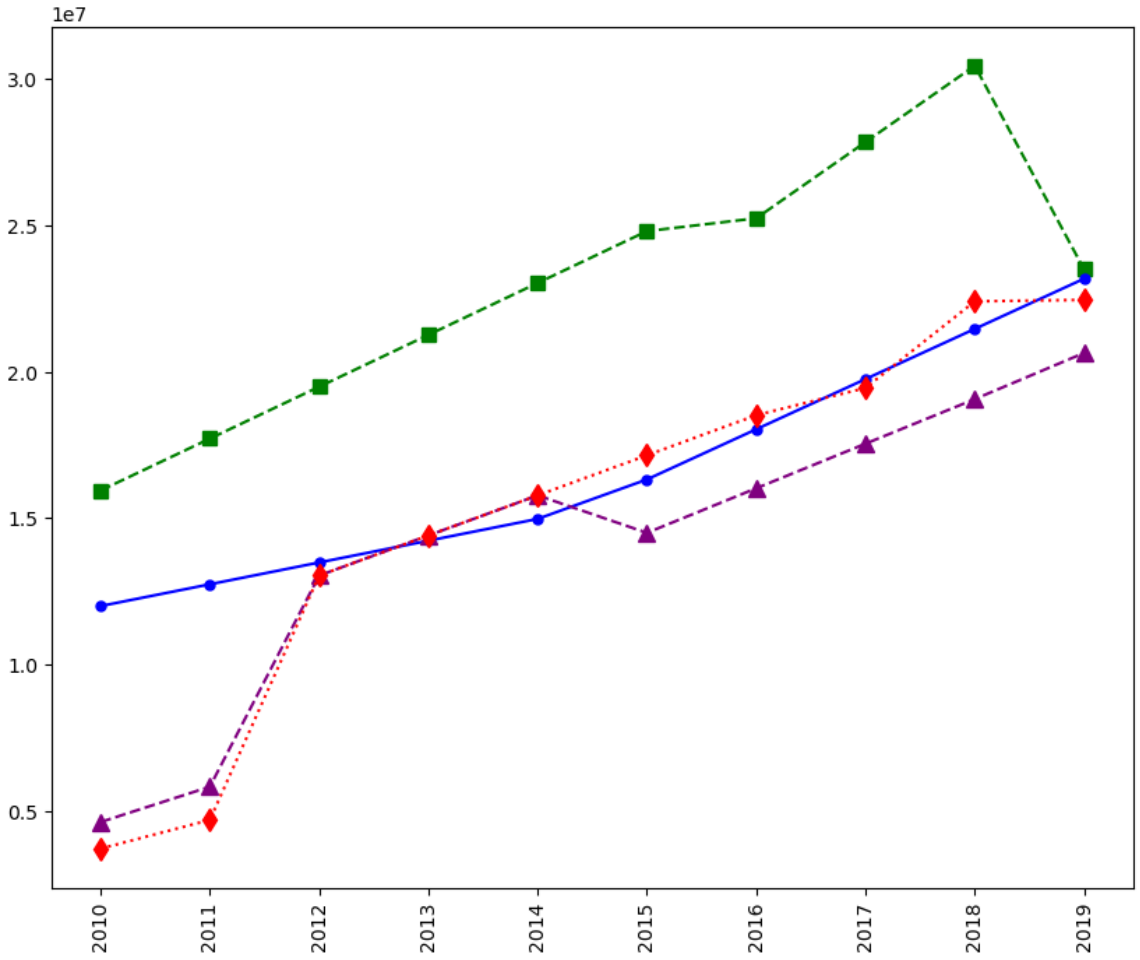
In [210]:
```
plt.plot(Salary[0], c = 'Green', ls = '--', marker = 's', ms = 10, label = P
plt.plot(Salary[1], c = 'Blue', ls = ':', marker = 'o',ms = 10, label = Play

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



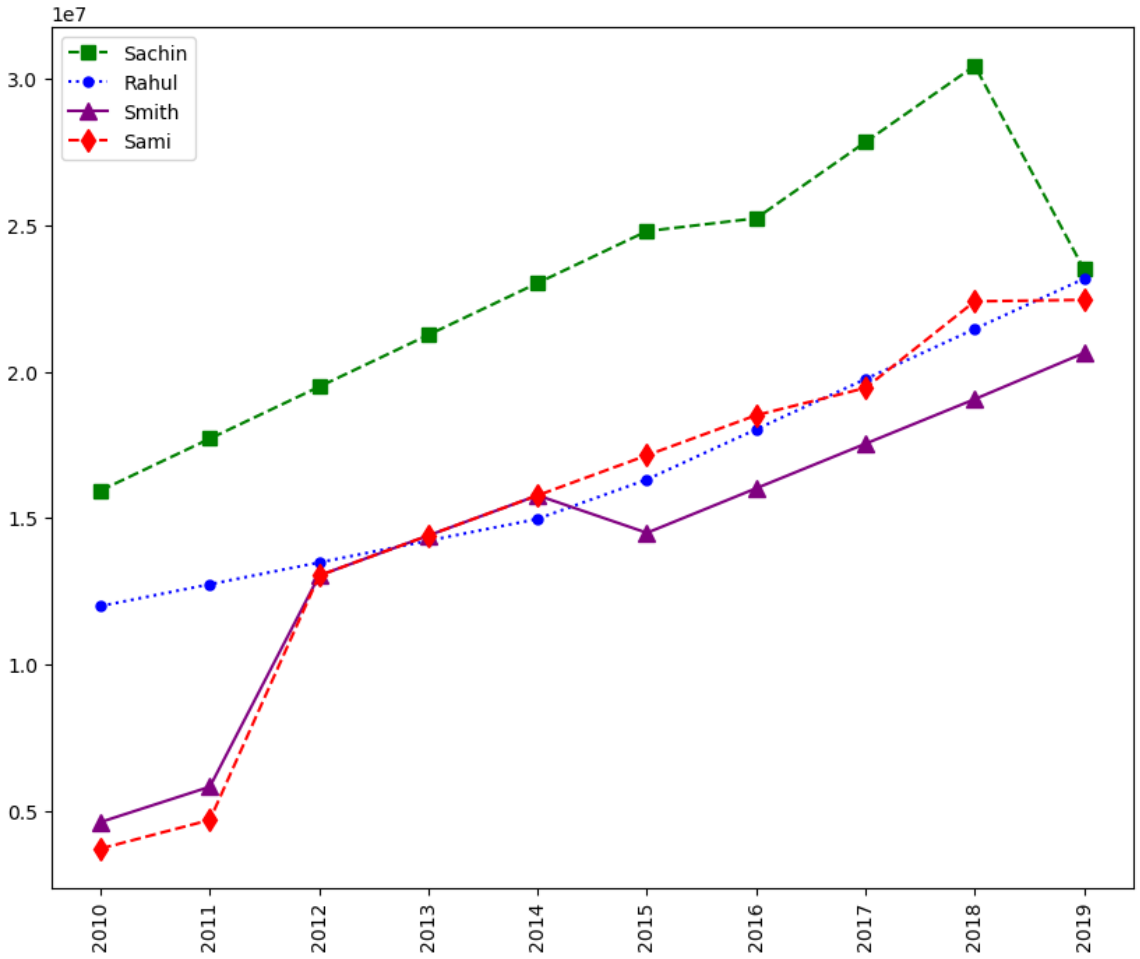
In [211]:
```
plt.plot(Salary[0], c = 'Green',ls = '--', marker = 's', ms = 7,label = Play
plt.plot(Salary[1], c ='Blue' , ls = '--', marker = 'o', ms = 5 ,label =Play
plt.plot(Salary[2], c = 'Purple',ls ='--' ,marker = '^', ms = 8 ,label =Play

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



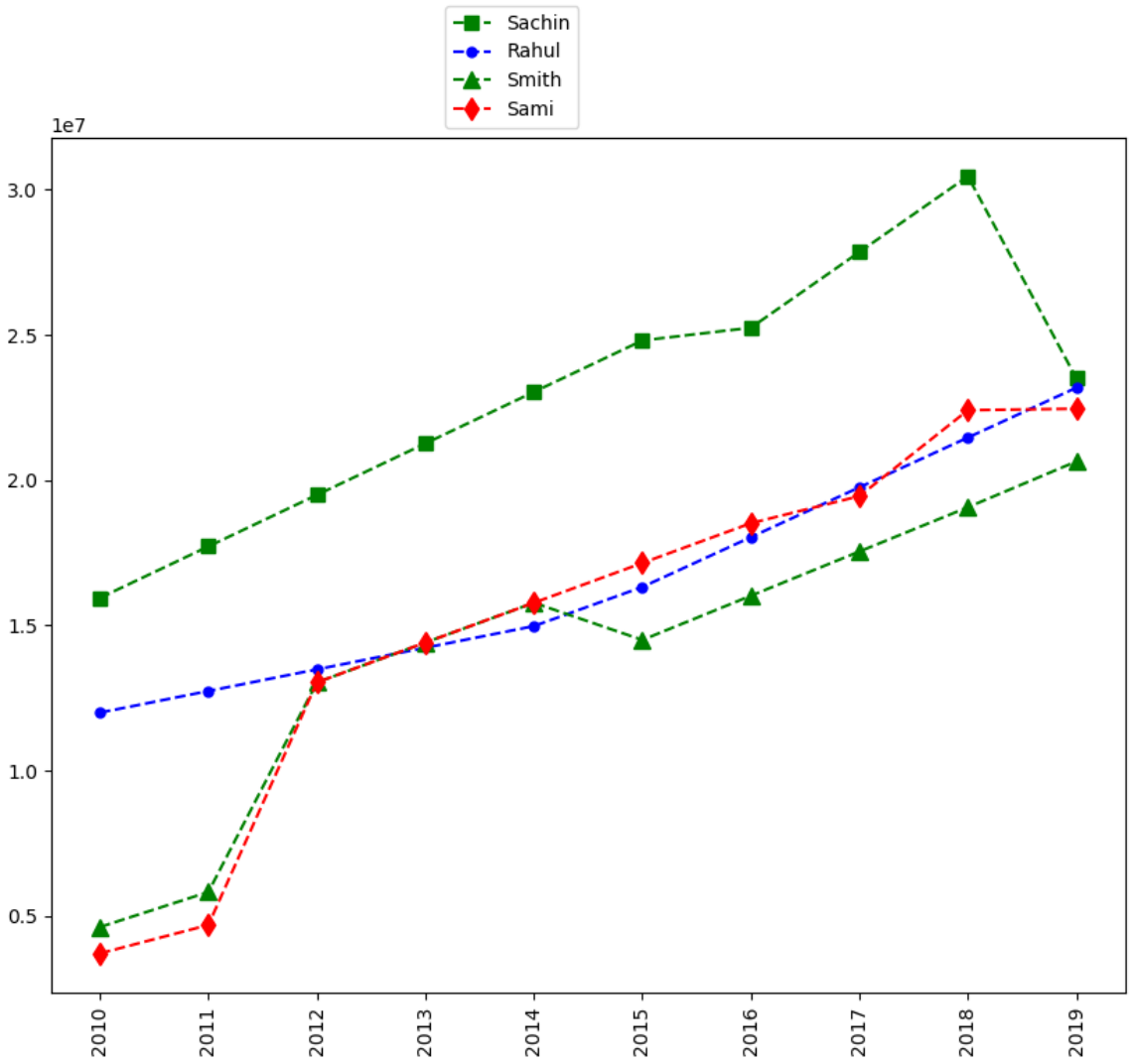
In [212]:
```
plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Play
plt.plot(Salary[1], c='Blue', ls = '-', marker = 'o', ms = 5, label = Player
plt.plot(Salary[2], c='purple', ls = '--', marker = '^', ms = 8, label = Pla
plt.plot(Salary[3], c='Red', ls = ':', marker = 'd', ms = 8, label = Players

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

plt.show()
```



In [213]:
```
# how to add legned in visualisation

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

plt.show()
```



In [216]:
```
plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Play
plt.plot(Salary[1], c='Blue', ls = '--', marker = 'o', ms = 5, label = Playe
plt.plot(Salary[2], c='Green', ls = '--', marker = '^', ms = 8, label = Play
plt.plot(Salary[3], c='Red', ls = '--', marker = 'd', ms = 8, label = Player
plt.legend(loc = 'lower right',bbox_to_anchor=(0.5,1) )
plt.xticks(list(range(0,10)), Seasons,rotation='vertical')

plt.show()
```

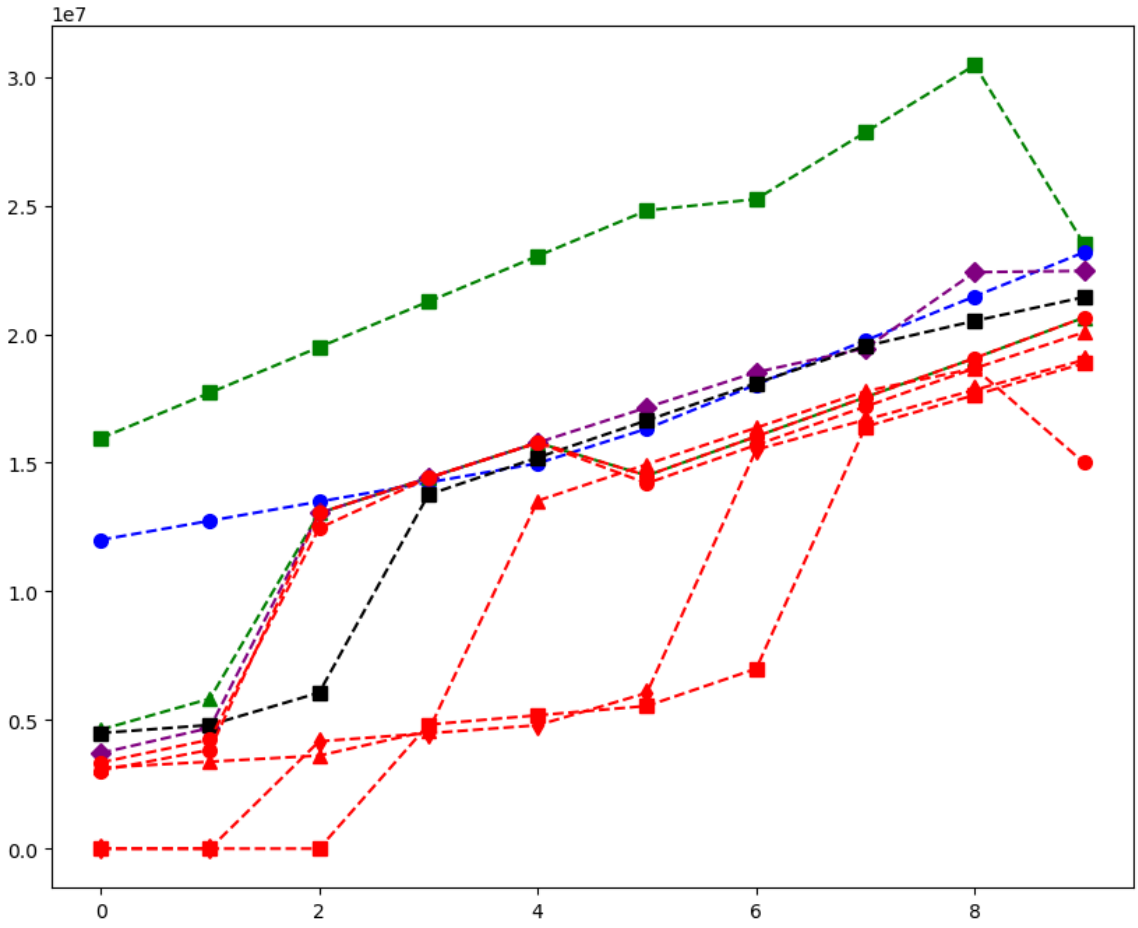


In [219]:

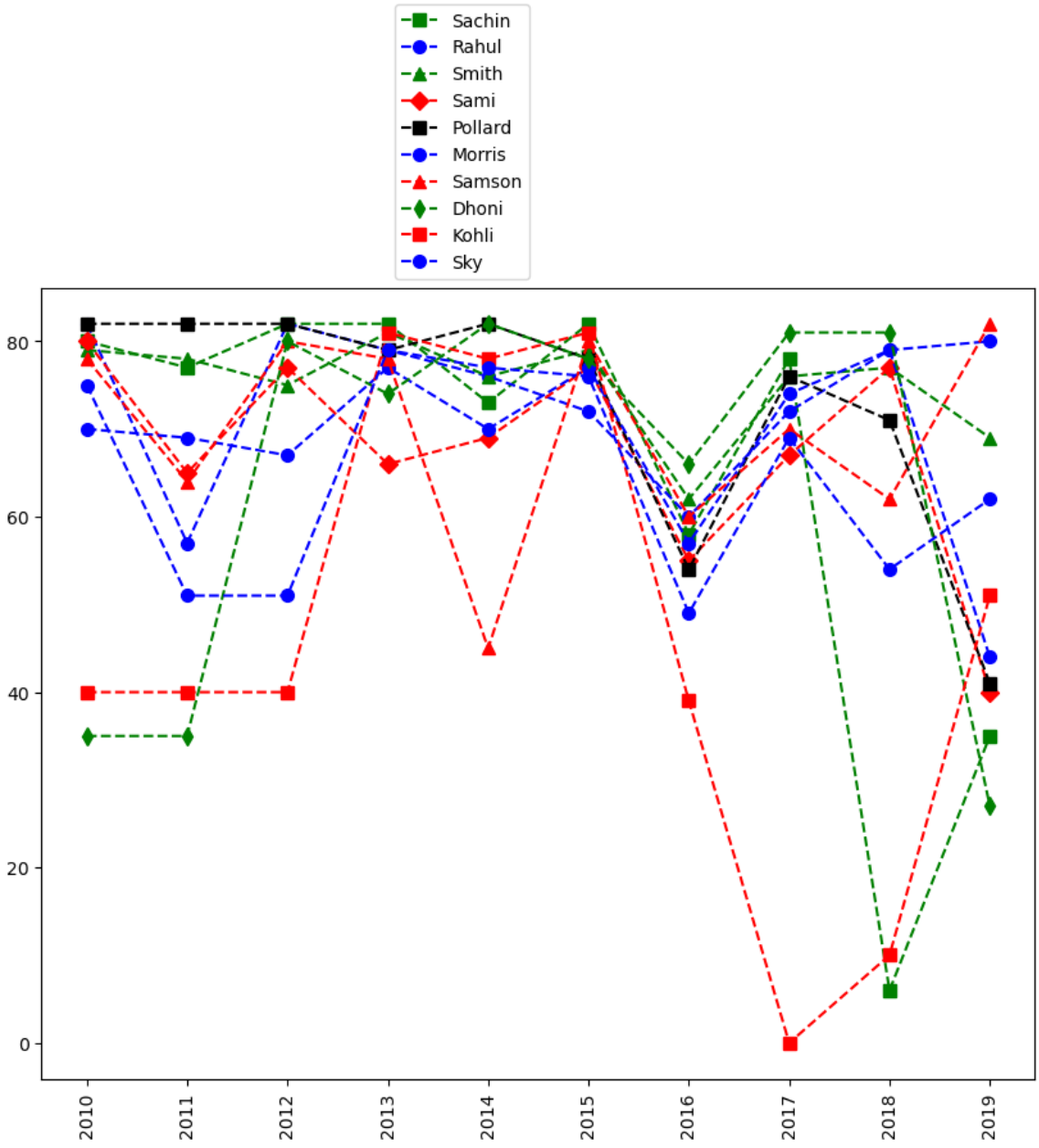
```
plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Play
plt.plot(Salary[1], c='Blue', ls = '--', marker = 'o', ms = 7, label = Playe
plt.plot(Salary[2], c='Green', ls = '--', marker = '^', ms = 7, label = Play
plt.plot(Salary[3], c='Purple', ls = '--', marker = 'D', ms = 7, label = Pla
plt.plot(Salary[4], c='Black', ls = '--', marker = 's', ms = 7, label = Play
plt.plot(Salary[5], c='Red', ls = '--', marker = 'o', ms = 7, label = Player
plt.plot(Salary[6], c='Red', ls = '--', marker = '^', ms = 7, label = Player
plt.plot(Salary[7], c='Red', ls = '--', marker = 'd', ms = 7, label = Player
plt.plot(Salary[8], c='Red', ls = '--', marker = 's', ms = 7, label = Player
plt.plot(Salary[9], c='Red', ls = '--', marker = 'o', ms = 7, label = Player

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

plt.show()
```

```
---------------------------------------------------------------------------
-
ValueError                                Traceback (most recent call las
t)
Cell In[219], line 12
      9 plt.plot(Salary[8], c='Red', ls = '--', marker = 's', ms = 7, labe
l = Players[8])
     10 plt.plot(Salary[9], c='Red', ls = '--', marker = 'o', ms = 7, labe
l = Players[9])
---> 12 plt.legend(loc = 'lover right',bbox_to_anchor=(0.5,1) )
     13 plt.xticks(list(range(0,10)), Seasons,rotation='vertical')
     15 plt.show()

File ~\anaconda3\Lib\site-packages\matplotlib\pyplot.py:2710, in legend(*a
rgs, **kwargs)
   2708 @_copy_docstring_and_deprecators(Axes.legend)
   2709 def legend(*args, **kwargs):
-> 2710     return gca().legend(*args, **kwargs)

File ~\anaconda3\Lib\site-packages\matplotlib\axes\_axes.py:318, in Axes.l
egend(self, *args, **kwargs)
    316 if len(extra_args):
    317     raise TypeError('legend only accepts two non-keyword argument
s')
--> 318 self.legend_ = mlegend.Legend(self, handles, labels, **kwargs)
    319 self.legend_._remove_method = self._remove_legend
    320 return self.legend_

File ~\anaconda3\Lib\site-packages\matplotlib\_api\deprecation.py:454, in
make_keyword_only.<locals>.wrapper(*args, **kwargs)
    448 if len(args) > name_idx:
    449     warn_deprecated(
    450         since, message="Passing the %(name)s %(obj_type)s "
    451         "positionally is deprecated since Matplotlib %(since)s; th
e "
    452         "parameter will become keyword-only %(removal)s.",
    453         name=name, obj_type=f"parameter of {func.__name__}()")
--> 454 return func(*args, **kwargs)

File ~\anaconda3\Lib\site-packages\matplotlib\legend.py:530, in Legend.__i
nit__(self, parent, handles, labels, loc, numpoints, markerscale, markerfi
rst, reverse, scatterpoints, scatteryoffsets, prop, fontsize, labelcolor,
borderpad, labelspacing, handlelength, handleheight, handletextpad, border
axespad, columnspacing, ncols, mode, fancybox, shadow, title, title_fontsi
ze, framealpha, edgecolor, facecolor, bbox_to_anchor, bbox_transform, fram
eon, handler_map, title_fontproperties, alignment, ncol, draggable)
    528             loc = locs[0] + ' ' + locs[1]
    529     # check that loc is in acceptable strings
--> 530     loc = _api.check_getitem(self.codes, loc=loc)
    532 if self.isaxes and self._outside_loc:
    533     raise ValueError(
    534         f"'outside' option for loc='{loc0}' keyword argument only
"
    535         "works for figure legends")

File ~\anaconda3\Lib\site-packages\matplotlib\_api\__init__.py:192, in che
ck_getitem(_mapping, **kwargs)
    190     return mapping[v]
    191 except KeyError:
--> 192     raise ValueError(
    193         "{!r} is not a valid value for {}; supported values are
```

```
{}"
    194         .format(v, k, ', '.join(map(repr, mapping)))) from None

ValueError: 'lover right' is not a valid value for loc; supported values a
re 'best', 'upper right', 'upper left', 'lower left', 'lower right', 'righ
t', 'center left', 'center right', 'lower center', 'upper center', 'cente
r'
```



In [220]:
```
# we can visualize the how many games played by a player

plt.plot(Games[0], c='Green', ls = '--', marker = 's', ms = 7, label = Playe
plt.plot(Games[1], c='Blue', ls = '--', marker = 'o', ms = 7, label = Player
plt.plot(Games[2], c='Green', ls = '--', marker = '^', ms = 7, label = Playe
plt.plot(Games[3], c='Red', ls = '--', marker = 'D', ms = 7, label = Players
plt.plot(Games[4], c='Black', ls = '--', marker = 's', ms = 7, label = Playe
plt.plot(Games[5], c='Blue', ls = '--', marker = 'o', ms = 7, label = Player
plt.plot(Games[6], c='red', ls = '--', marker = '^', ms = 7, label = Players
plt.plot(Games[7], c='Green', ls = '--', marker = 'd', ms = 7, label = Playe
plt.plot(Games[8], c='Red', ls = '--', marker = 's', ms = 7, label = Players
plt.plot(Games[9], c='Blue', ls = '--', marker = 'o', ms = 7, label = Player

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

plt.show()
```



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