

```

In [9]: #Import numpy
import numpy as np

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

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

#Salaries
Sachin_Salary = [15946875, 17718750, 19490625, 21262500, 23034375, 24806250, 25244493,
Rahul_Salary = [12000000, 12744189, 13488377, 14232567, 14976754, 16324500, 18038573, 1
Smith_Salary = [4621800, 5828090, 13041250, 14410581, 15779912, 14500000, 16022500, 175
Sami_Salary = [3713640, 4694041, 13041250, 14410581, 15779912, 17149243, 18518574, 1945
Pollard_Salary = [4493160, 4806720, 6061274, 13758000, 15202590, 16647180, 18091770, 19
Morris_Salary = [3348000, 4235220, 12455000, 14410581, 15779912, 14500000, 16022500, 17
Samson_Salary = [3144240, 3380160, 3615960, 4574189, 13520500, 14940153, 16359805, 1777
Dhoni_Salary = [0, 0, 4171200, 4484040, 4796880, 6053663, 15506632, 16669630, 17832627, 1
Kohli_Salary = [0, 0, 0, 4822800, 5184480, 5546160, 6993708, 16402500, 17632688, 18862875
Sky_Salary = [3031920, 3841443, 13041250, 14410581, 15779912, 14200000, 15691000, 17182

#Matrix
Salary = np.array([Sachin_Salary, Rahul_Salary, Smith_Salary, Sami_Salary, Polla

#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, Samso

#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, Morr

```

```

In [11]: Salary # martrix format

```

```
Out[11]: array([[15946875, 17718750, 19490625, 21262500, 23034375, 24806250,
                25244493, 27849149, 30453805, 23500000],
               [12000000, 12744189, 13488377, 14232567, 14976754, 16324500,
                18038573, 19752645, 21466718, 23180790],
               [ 4621800,  5828090, 13041250, 14410581, 15779912, 14500000,
                16022500, 17545000, 19067500, 20644400],
               [ 3713640,  4694041, 13041250, 14410581, 15779912, 17149243,
                18518574, 19450000, 22407474, 22458000],
               [ 4493160,  4806720,  6061274, 13758000, 15202590, 16647180,
                18091770, 19536360, 20513178, 21436271],
               [ 3348000,  4235220, 12455000, 14410581, 15779912, 14500000,
                16022500, 17545000, 19067500, 20644400],
               [ 3144240,  3380160,  3615960,  4574189, 13520500, 14940153,
                16359805, 17779458, 18668431, 20068563],
               [      0,      0,  4171200,  4484040,  4796880,  6053663,
                15506632, 16669630, 17832627, 18995624],
               [      0,      0,      0,  4822800,  5184480,  5546160,
                6993708, 16402500, 17632688, 18862875],
               [ 3031920,  3841443, 13041250, 14410581, 15779912, 14200000,
                15691000, 17182000, 18673000, 15000000]])
```

In [11]: Games

```
Out[11]: array([[80, 77, 82, 82, 73, 82, 58, 78,  6, 35],
               [82, 57, 82, 79, 76, 72, 60, 72, 79, 80],
               [79, 78, 75, 81, 76, 79, 62, 76, 77, 69],
               [80, 65, 77, 66, 69, 77, 55, 67, 77, 40],
               [82, 82, 82, 79, 82, 78, 54, 76, 71, 41],
               [70, 69, 67, 77, 70, 77, 57, 74, 79, 44],
               [78, 64, 80, 78, 45, 80, 60, 70, 62, 82],
               [35, 35, 80, 74, 82, 78, 66, 81, 81, 27],
               [40, 40, 40, 81, 78, 81, 39,  0, 10, 51],
               [75, 51, 51, 79, 77, 76, 49, 69, 54, 62]])
```

In [13]: Points

```
Out[13]: 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 [15]: 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 [17]: np.reshape(mydata,(4,5)) # 5 rows & 4 columns
```

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

In [19]: mydata

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

```
In [21]: MATR1 = np.reshape(mydata, (5,4), order = 'c')
MATR1
```

```
Out[21]: array([[ 0,  1,  2,  3],
               [ 4,  5,  6,  7],
               [ 8,  9, 10, 11],
               [12, 13, 14, 15],
               [16, 17, 18, 19]])
```

```
In [23]: MATR1
```

```
Out[23]: array([[ 0,  1,  2,  3],
               [ 4,  5,  6,  7],
               [ 8,  9, 10, 11],
               [12, 13, 14, 15],
               [16, 17, 18, 19]])
```

```
In [25]: MATR1[4,3]
```

```
Out[25]: 19
```

```
In [27]: MATR1[3,3]
```

```
Out[27]: 15
```

```
In [29]: MATR1
```

```
Out[29]: array([[ 0,  1,  2,  3],
               [ 4,  5,  6,  7],
               [ 8,  9, 10, 11],
               [12, 13, 14, 15],
               [16, 17, 18, 19]])
```

```
In [31]: MATR1[-3,-1]
```

```
Out[31]: 11
```

```
In [33]: MATR1
```

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

```
In [35]: mydata
```

```
Out[35]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
               17, 18, 19])
```

```
In [37]: MATR2 = np.reshape(mydata, (5,4), order = 'F') # reshape behaviour are - 'C', 'F'
MATR2
```

```
Out[37]: array([[ 0,  5, 10, 15],
               [ 1,  6, 11, 16],
               [ 2,  7, 12, 17],
               [ 3,  8, 13, 18],
               [ 4,  9, 14, 19]])
```

```
In [39]: MATR2[4,3]
```

```
Out[39]: 19
```

```
In [41]: MATR2[0,2]
```

```
Out[41]: 10
```

```
In [43]: MATR2[0:2]
```

```
Out[43]: array([[ 0,  5, 10, 15],
               [ 1,  6, 11, 16]])
```

```
In [45]: MATR2
```

```
Out[45]: array([[ 0,  5, 10, 15],
               [ 1,  6, 11, 16],
               [ 2,  7, 12, 17],
               [ 3,  8, 13, 18],
               [ 4,  9, 14, 19]])
```

```
In [47]: MATR2[1:2]
```

```
Out[47]: array([[ 1,  6, 11, 16]])
```

```
In [49]: MATR2[1,2]
```

```
Out[49]: 11
```

```
In [51]: MATR2
```

```
Out[51]: array([[ 0,  5, 10, 15],
               [ 1,  6, 11, 16],
               [ 2,  7, 12, 17],
               [ 3,  8, 13, 18],
               [ 4,  9, 14, 19]])
```

```
In [53]: MATR2[-2,-1]
```

```
Out[53]: 18
```

```
In [55]: MATR2[-3,-3]
```

```
Out[55]: 7
```

```
In [57]: MATR2
```

```
Out[57]: array([[ 0,  5, 10, 15],
               [ 1,  6, 11, 16],
               [ 2,  7, 12, 17],
               [ 3,  8, 13, 18],
               [ 4,  9, 14, 19]])
```

```
In [59]: MATR2[0:2]
```

```
Out[59]: array([[ 0,  5, 10, 15],
                [ 1,  6, 11, 16]])
```

```
In [61]: mydata
```

```
Out[61]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
                17, 18, 19])
```

```
In [63]: MATR3 = np.reshape(mydata, (5,4), order = 'A')
MATR3
```

```
Out[63]: array([[ 0,  1,  2,  3],
                [ 4,  5,  6,  7],
                [ 8,  9, 10, 11],
                [12, 13, 14, 15],
                [16, 17, 18, 19]])
```

```
In [65]: MATR2
```

```
Out[65]: array([[ 0,  5, 10, 15],
                [ 1,  6, 11, 16],
                [ 2,  7, 12, 17],
                [ 3,  8, 13, 18],
                [ 4,  9, 14, 19]])
```

```
In [67]: MATR1
```

```
Out[67]: array([[ 0,  1,  2,  3],
                [ 4,  5,  6,  7],
                [ 8,  9, 10, 11],
                [12, 13, 14, 15],
                [16, 17, 18, 19]])
```

```
In [69]: a1 = ['welcome', 'to', 'datascience']
a2 = ['required', 'hard', 'work' ]
a3 = [1,2,3]
```

```
In [71]: [a1,a2,a3]
```

```
Out[71]: [['welcome', 'to', 'datascience'], ['required', 'hard', 'work'], [1, 2, 3]]
```

```
In [73]: np.array([a1,a2,a3])
```

```
Out[73]: array(['welcome', 'to', 'datascience'],
                ['required', 'hard', 'work'],
                ['1', '2', '3']], dtype='<U11')
```

```
In [75]: Games
```

```
Out[75]: 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 [77]: Games[0]
```

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

```
In [79]: Games[5]
```

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

```
In [81]: Games[0:5]
```

```
Out[81]: 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 [83]: Games[0,5]
```

```
Out[83]: 82
```

```
In [85]: Games[0,2]
```

```
Out[85]: 82
```

```
In [87]: Games
```

```
Out[87]: 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 [89]: Games[0:2]
```

```
Out[89]: array([[80, 77, 82, 82, 73, 82, 58, 78, 6, 35],
               [82, 57, 82, 79, 76, 72, 60, 72, 79, 80]])
```

```
In [91]: Games
```

```
Out[91]: 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 [93]: Games[1:2]
```

```
Out[93]: array([[82, 57, 82, 79, 76, 72, 60, 72, 79, 80]])
```

```
In [95]: Games[2]
```

```
Out[95]: array([79, 78, 75, 81, 76, 79, 62, 76, 77, 69])
```

```
In [97]: Games
```

```
Out[97]: 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 [99]: Games[2,8]
```

```
Out[99]: 77
```

```
In [101... Games
```

```
Out[101... 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 [103... Games[-3:-1]
```

```
Out[103... array([[35, 35, 80, 74, 82, 78, 66, 81, 81, 27],
                 [40, 40, 40, 81, 78, 81, 39, 0, 10, 51]])
```

```
In [105... Games[-3,-1]
```

```
Out[105... 27
```

In [107...

Points

Out[107...

```
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 [109...

Points[0]

Out[109...

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

In [111...

Points

Out[111...

```
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...

Points[6,1]

Out[113...

```
1104
```

In [115...

Points[3:6]

Out[115...

```
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 [117...

Points

Out[117...

```
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 [119...

Points[0]

Out[119...

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

In [121...

Points



```
Out[121...] 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 [123...] Points[6,1]
```

```
Out[123...] 1104
```

```
In [125...] Points[3:6]
```

```
Out[125...] 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 [127...] Points
```

```
Out[127...] 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 [129...] Points[-6,-1]
```

```
Out[129...] 646
```

```
In [131...] #===== DICTIONARY =====#

# dict does not maintain the order

dict1 = {'key1':'val1', 'key2':'val2', 'key3':'val3'}
```

```
In [135...] dict1
```

```
Out[135...] {'key1': 'val1', 'key2': 'val2', 'key3': 'val3'}
```

```
In [137...] dict1['key2']
```

```
Out[137...] 'val2'
```

```
In [139...] dict2 = {'bang':2,'hyd':'we are hear', 'pune':True}
```

```
In [141...] dict2
```

```
Out[141...] {'bang': 2, 'hyd': 'we are hear', 'pune': True}
```

```
In [143... dict3 = {'Germany': 'I have been here', 'France': 2, 'Spain': True}
```

```
In [145... dict3
```

```
Out[145... {'Germany': 'I have been here', 'France': 2, 'Spain': True}
```

```
In [147... dict3['Germany']
```

```
Out[147... 'I have been here'
```

if you check the 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 don't required to count which no row which players are sitting

```
In [150... Games
```

```
Out[150... 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 [152... Pdict
```

```
Out[152... {'Sachin': 0,
            'Rahul': 1,
            'Smith': 2,
            'Sami': 3,
            'Pollard': 4,
            'Morris': 5,
            'Samson': 6,
            'Dhoni': 7,
            'Kohli': 8,
            'Sky': 9}
```

```
In [154... # how do i know player kobe Bryant is at

Pdict['Sachin']
```

```
Out[154... 0
```

```
In [156... Games[0]
```

```
Out[156... array([80, 77, 82, 82, 73, 82, 58, 78, 6, 35])
```

```
In [158... Games
```

```
Out[158... 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... Pdict['Rahul']
```

```
Out[160... 1
```

```
In [162... Games[1]
```

```
Out[162... array([82, 57, 82, 79, 76, 72, 60, 72, 79, 80])
```

## Games

```
In [165... Games[Pdict['Rahul']]
```

```
Out[165... array([82, 57, 82, 79, 76, 72, 60, 72, 79, 80])
```

```
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 [169...

Salary

```
Out[169...] 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 [171...

Salary[2,4]

Out[171...

15779912

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 [175...] Salary[Pdict['Sky']][Sdict['2019']]
```

```
Out[175...] 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 [179...] Games
```

```
Out[179... 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 [181... Salary/Games
```

C:\Users\samik\AppData\Local\Temp\ipykernel\_17884\3709746658.py:1: RuntimeWarning: divide by zero encountered in divide  
Salary/Games

```
Out[181... 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 [183... np.round(Salary/Games)
```

C:\Users\samik\AppData\Local\Temp\ipykernel\_17884\3232172828.py:1: RuntimeWarning: divide by zero encountered in divide  
 np.round(Salary/Games)

```
Out[183... 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 [185... import warnings
warnings.filterwarnings('ignore')
#np.round(FieldGoals/Games)
#FieldGoals/Games # this matrix is lot of decimal points yo can not round
#round()
```

```
In [187... ## --- First visualization ----##
```

```
In [3]: import numpy as np
import matplotlib.pyplot as plt
```

```
In [5]: %matplotlib inline # keep the plot inside jupyter nots insted of getting in othe
```

UsageError: unrecognized arguments: # keep the plot inside jupyter nots insted of getting in other screen

```
In [193... Salary
```

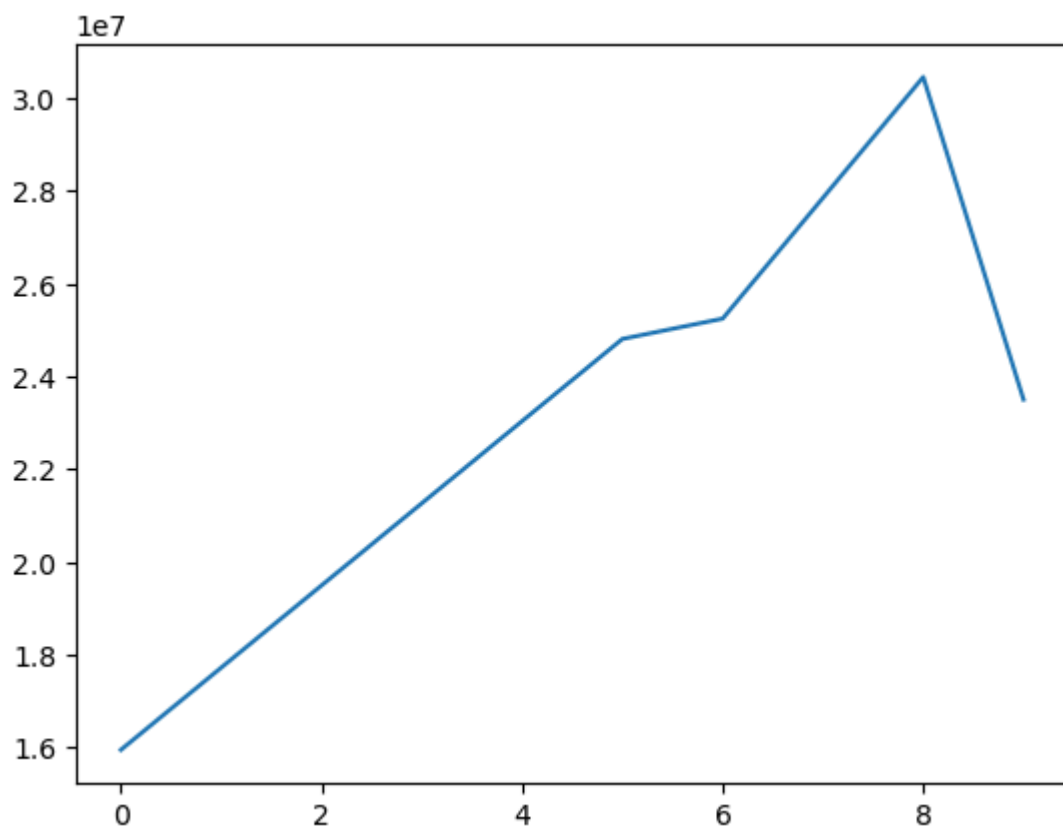
```
Out[193...] array([[15946875, 17718750, 19490625, 21262500, 23034375, 24806250,
        25244493, 27849149, 30453805, 23500000],
        [12000000, 12744189, 13488377, 14232567, 14976754, 16324500,
        18038573, 19752645, 21466718, 23180790],
        [ 4621800,  5828090, 13041250, 14410581, 15779912, 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 [195...] Salary[0]
```

```
Out[195...] array([15946875, 17718750, 19490625, 21262500, 23034375, 24806250,
        25244493, 27849149, 30453805, 23500000])
```

```
In [197...] plt.plot(Salary[0])
```

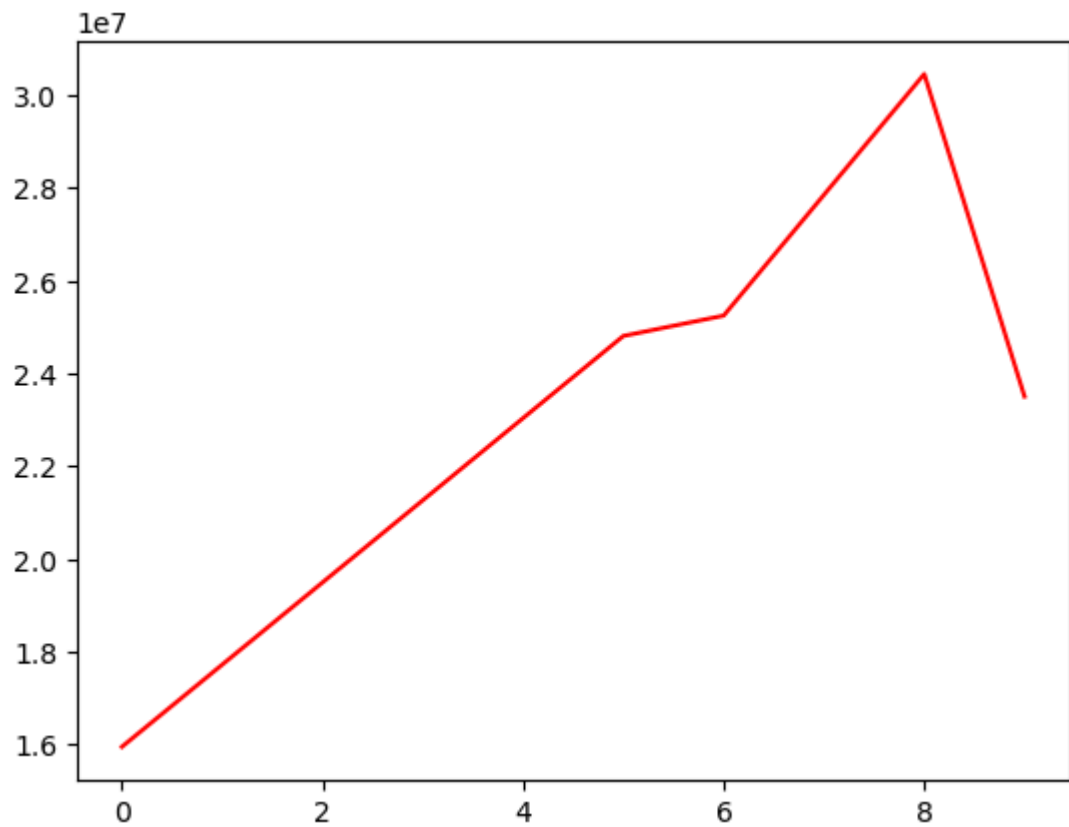
```
Out[197...] [<matplotlib.lines.Line2D at 0x29cfa7df830>]
```



```
In [199...] plt.plot(Salary[0], c='red')
```

```
Out[199...] [<matplotlib.lines.Line2D at 0x29cfa8914c0>]
```

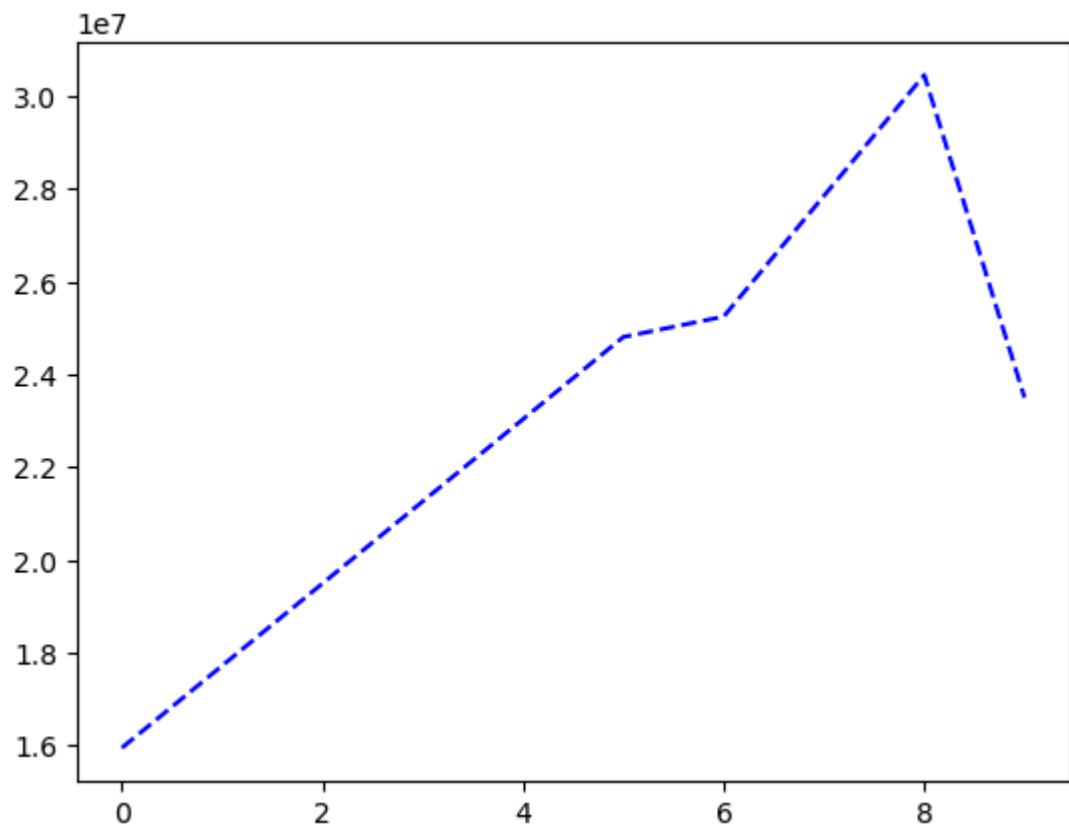




```
In [ ]: matplotlib inline  
plt.rcParams['figure.figsize'] = 10,6
```

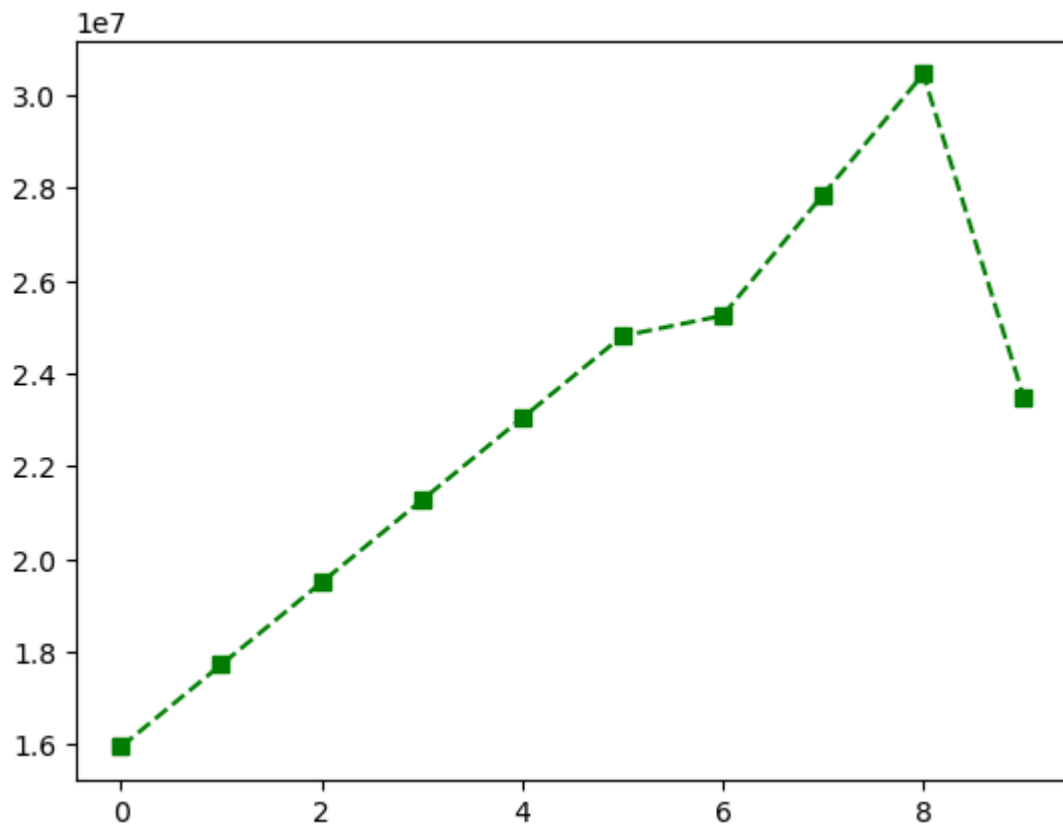
```
In [203... plt.plot(Salary[0], c='Blue', ls = 'dashed')
```

```
Out[203... [<matplotlib.lines.Line2D at 0x29cfa90ba70>]
```



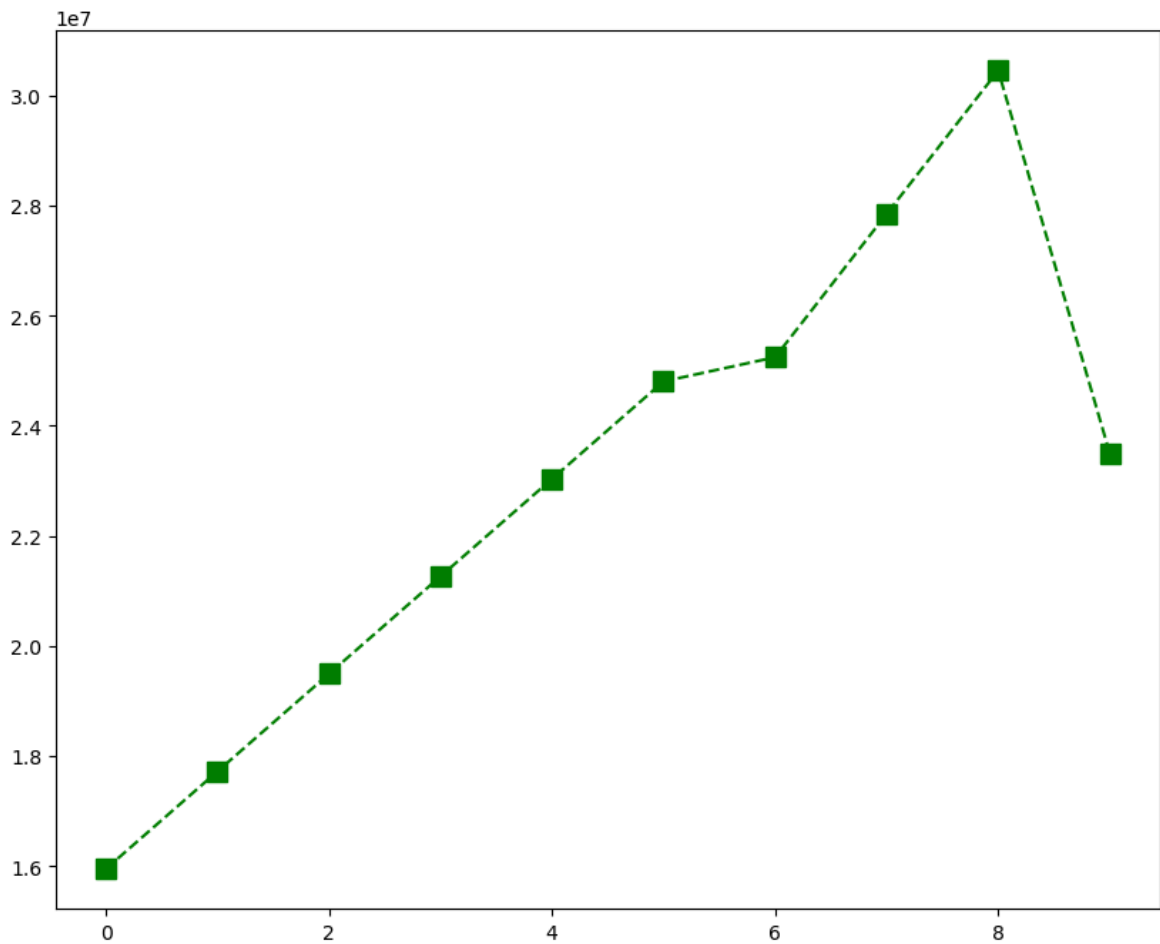
```
In [205... plt.plot(Salary[0], c='Green', ls = '--', marker = 's') # s - squares
```

```
Out[205... [<matplotlib.lines.Line2D at 0x29cfb1aec00>]
```



```
In [207... %matplotlib inline  
plt.rcParams['figure.figsize'] = 10,8 #runtime configuration parameter
```

```
In [209... plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 10)  
plt.show()
```



```
In [211...] list(range(0,10))
```

```
Out[211...] [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

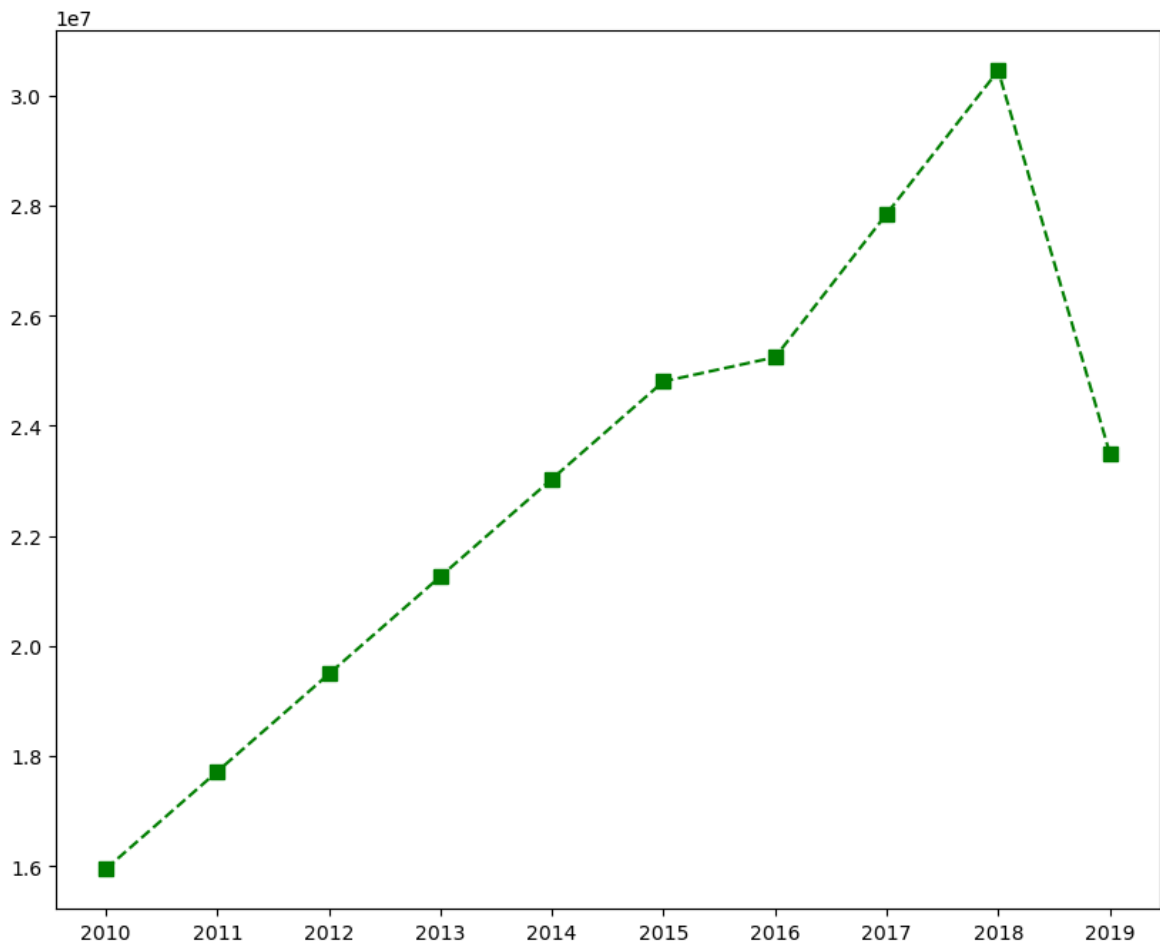
```
In [213...] Sdict
```

```
Out[213...] {'2010': 0,  
             '2011': 1,  
             '2012': 2,  
             '2013': 3,  
             '2014': 4,  
             '2015': 5,  
             '2016': 6,  
             '2017': 7,  
             '2018': 8,  
             '2019': 9}
```

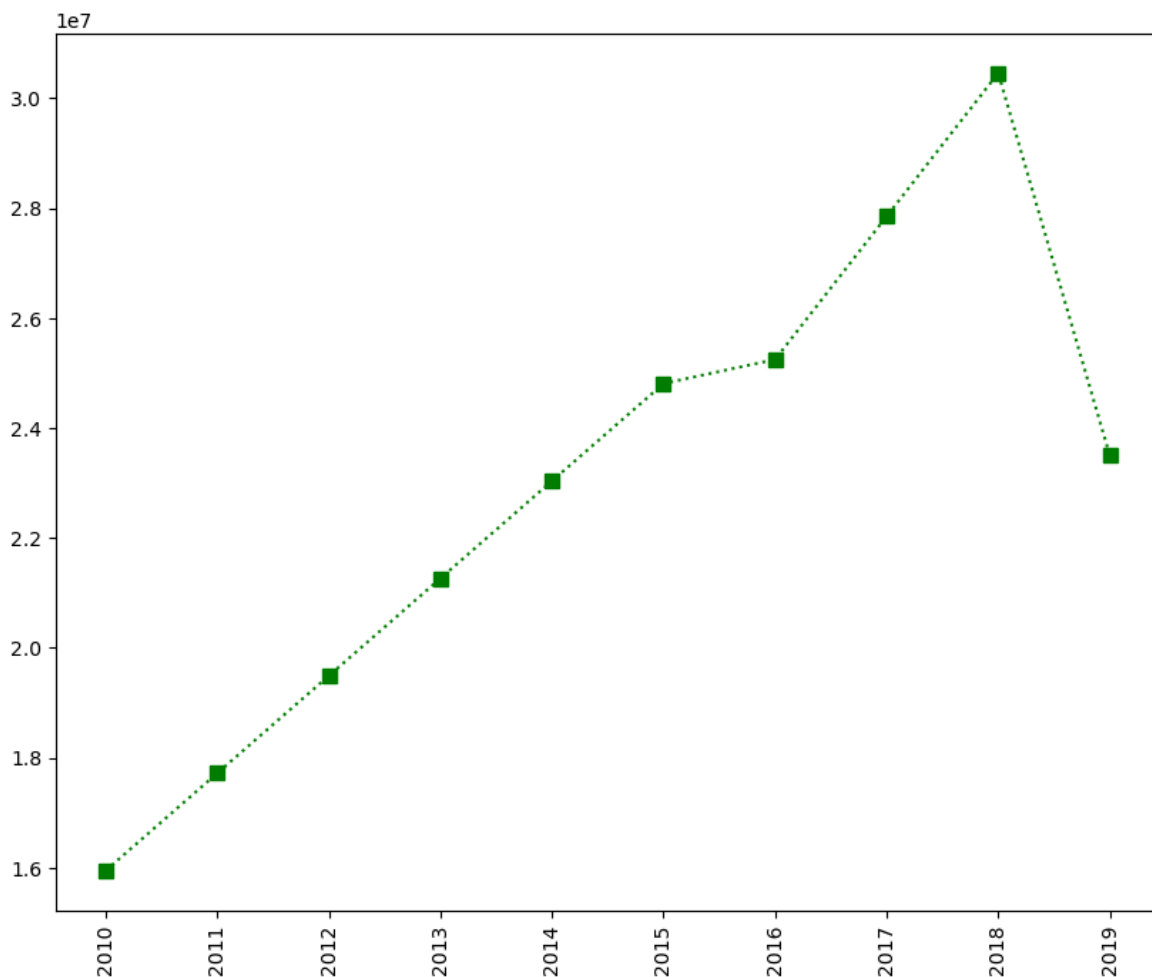
```
In [215...] Pdict
```

```
Out[215...] {'Sachin': 0,  
             'Rahul': 1,  
             'Smith': 2,  
             'Sami': 3,  
             'Pollard': 4,  
             'Morris': 5,  
             'Samson': 6,  
             'Dhoni': 7,  
             'Kohli': 8,  
             'Sky': 9}
```

```
In [217... plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7)
plt.xticks(list(range(0,10)), Seasons)
plt.show()
```



```
In [219... 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()
```



In [221...

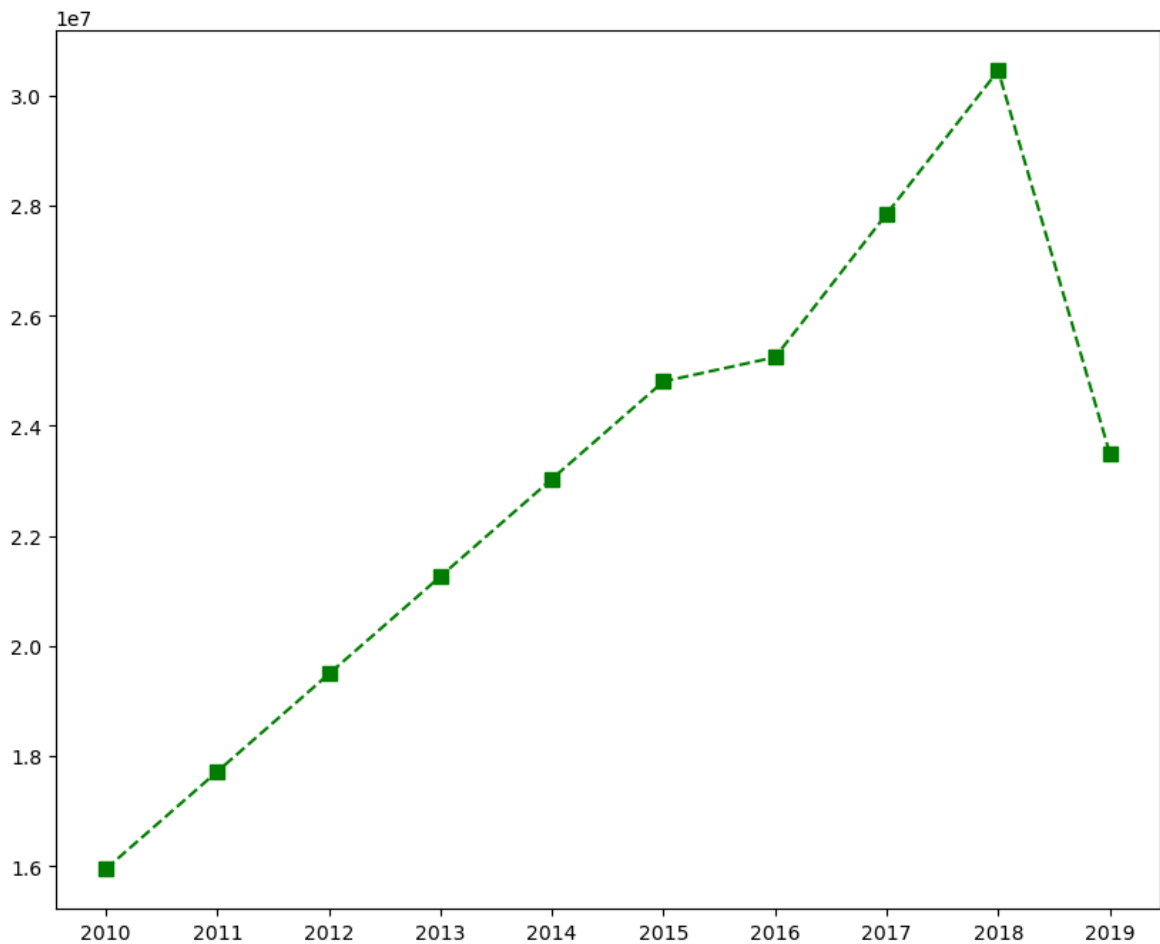
Games

Out[221...

```
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 [223...

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



In [225...] `Salary[0]`

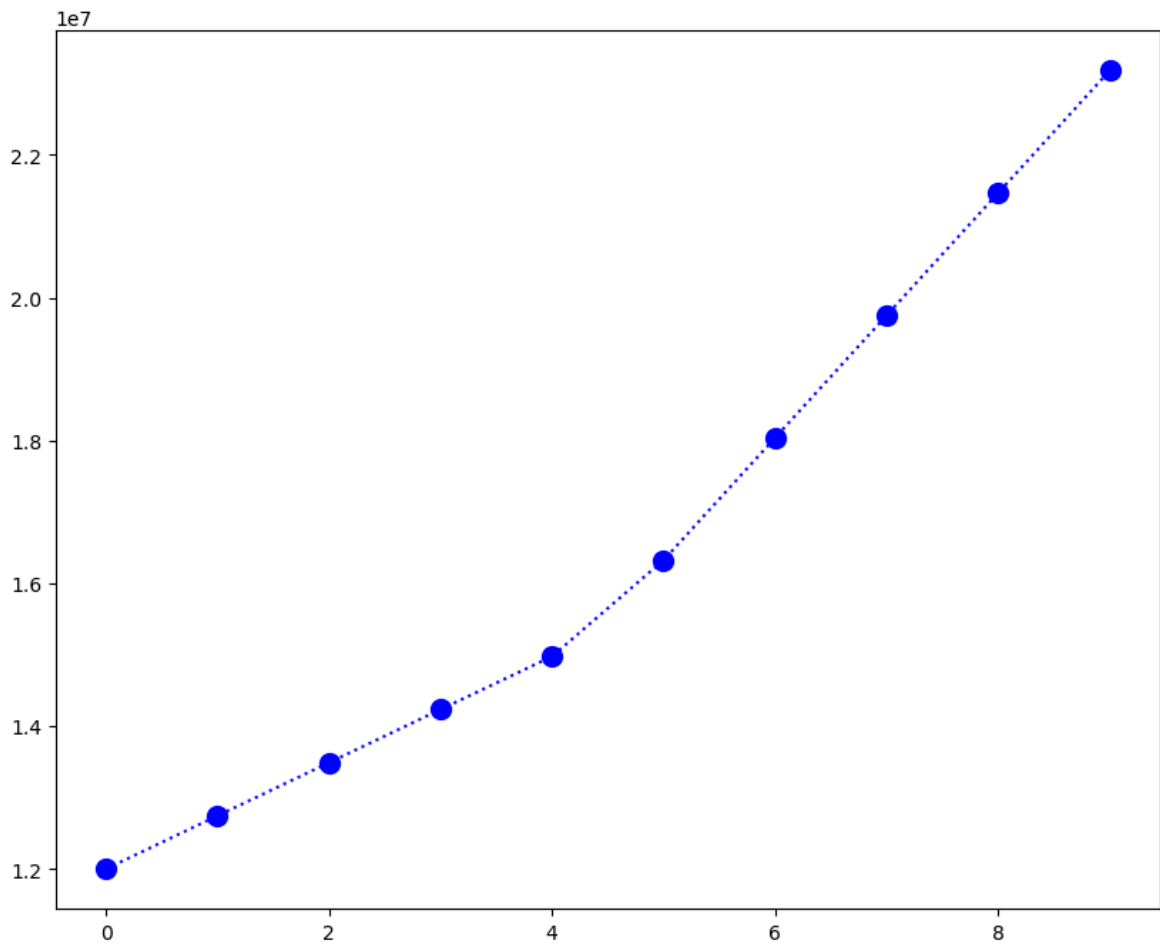
Out[225...] `array([15946875, 17718750, 19490625, 21262500, 23034375, 24806250, 25244493, 27849149, 30453805, 23500000])`

In [227...] `Salary[1]`

Out[227...] `array([12000000, 12744189, 13488377, 14232567, 14976754, 16324500, 18038573, 19752645, 21466718, 23180790])`

In [229...] `plt.plot(Salary[1], c='Blue', ls = ':', marker = 'o', ms = 10, label = Players[1`

Out[229...] `[<matplotlib.lines.Line2D at 0x29cfa97f7d0>]`

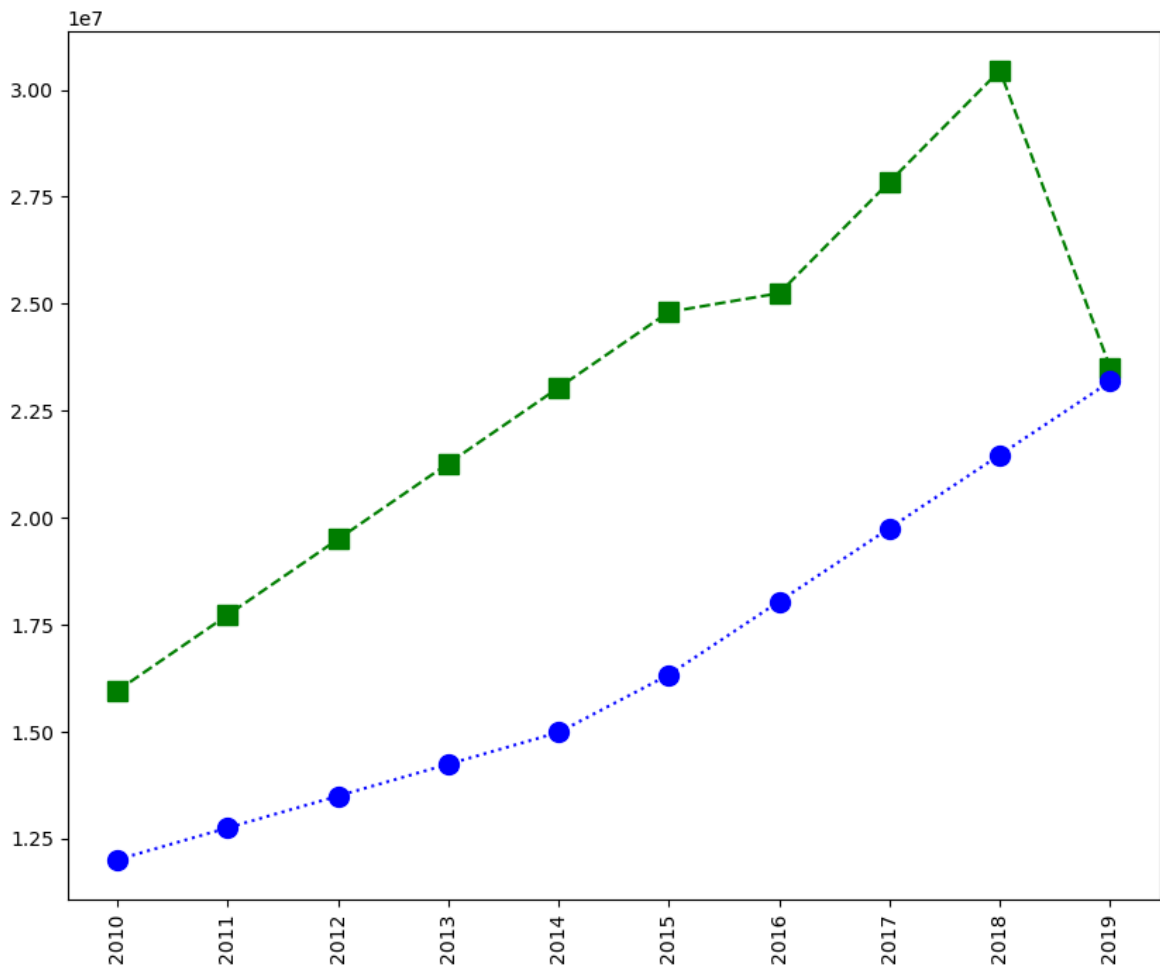


```
In [ ]: # More visualization
```

```
In [231... plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 10, label = Players
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()
```



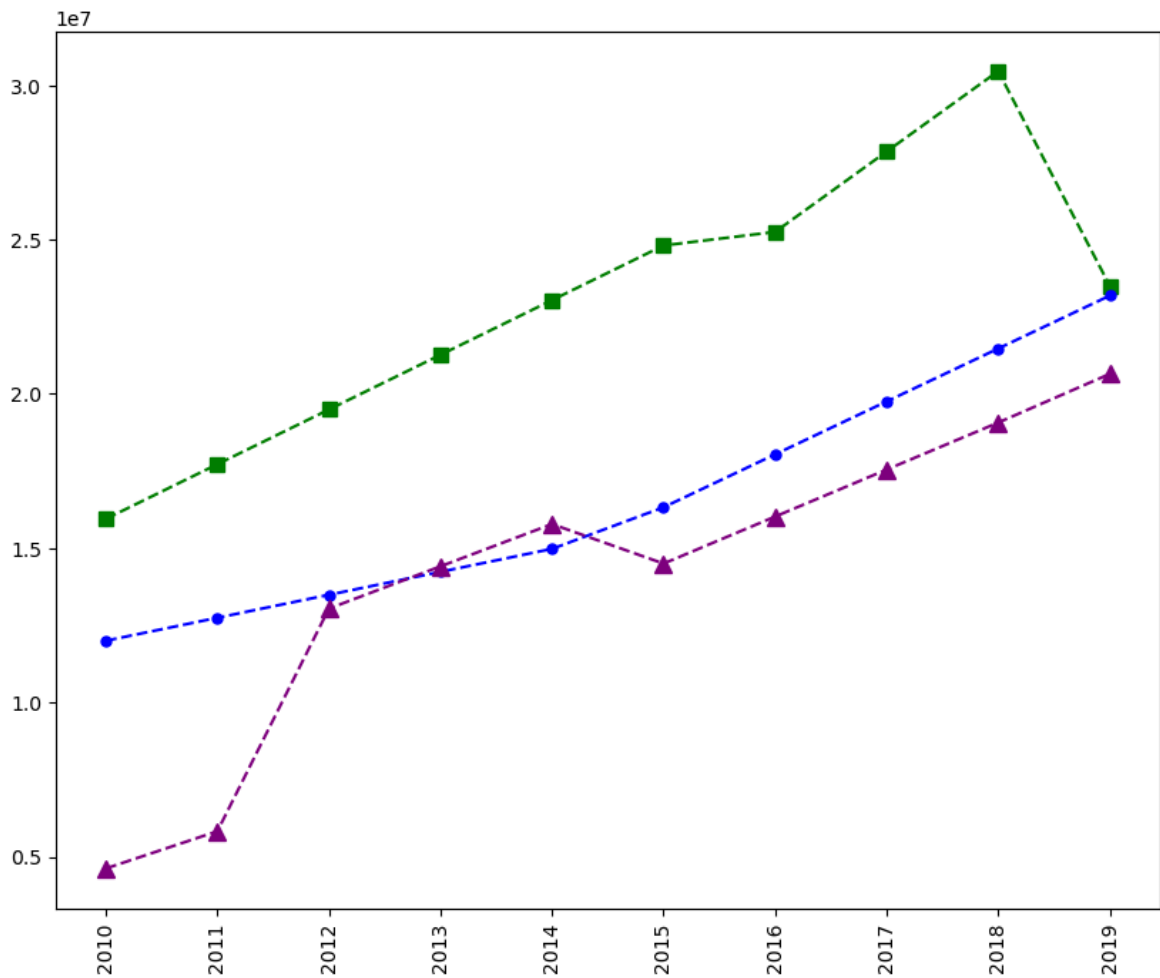
In [233...

```
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()
```

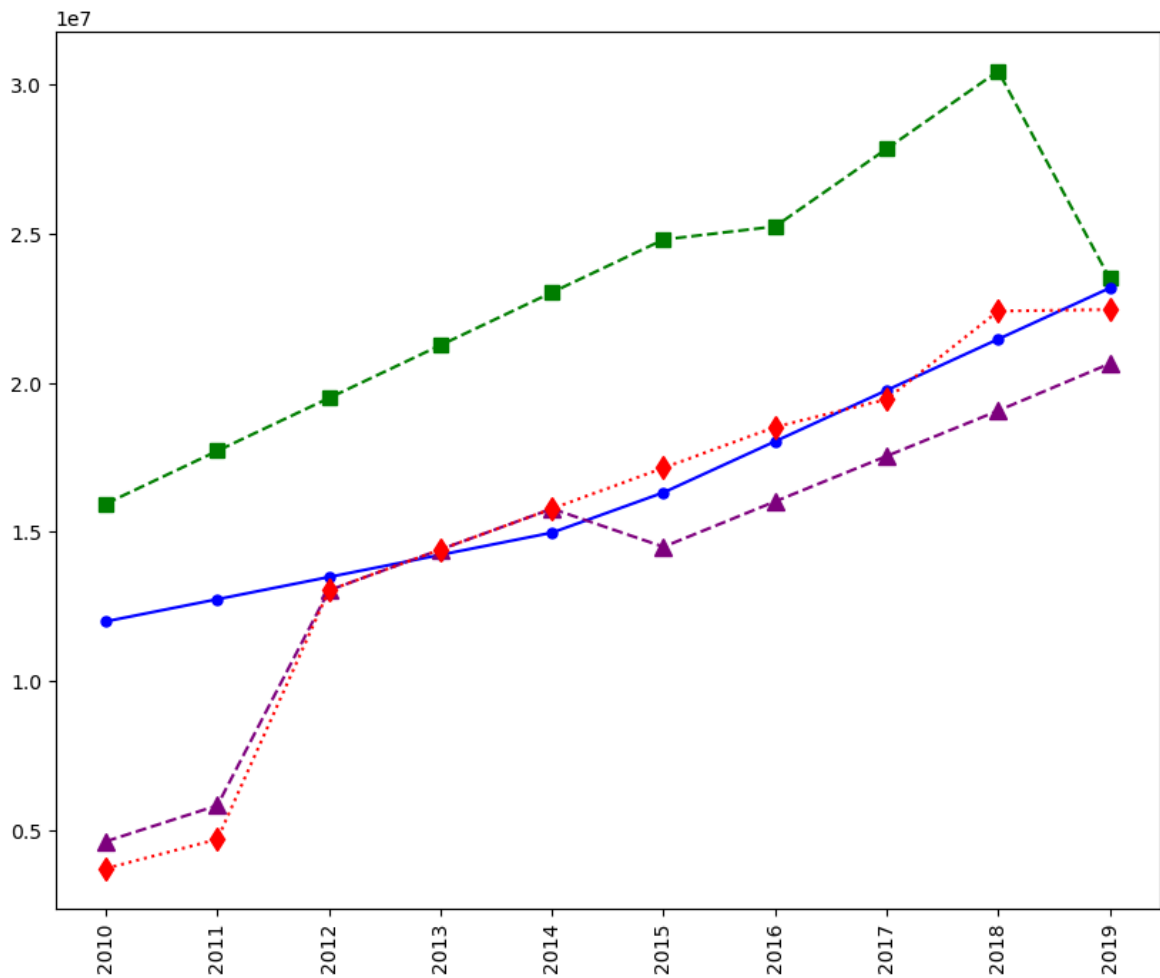




```
In [235... plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Players[0])
plt.plot(Salary[1], c='Blue', ls = '-', marker = 'o', ms = 5, label = Players[1])
plt.plot(Salary[2], c='purple', ls = '--', marker = '^', ms = 8, label = Players[2])
plt.plot(Salary[3], c='Red', ls = ':', marker = 'd', ms = 8, label = Players[3])

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

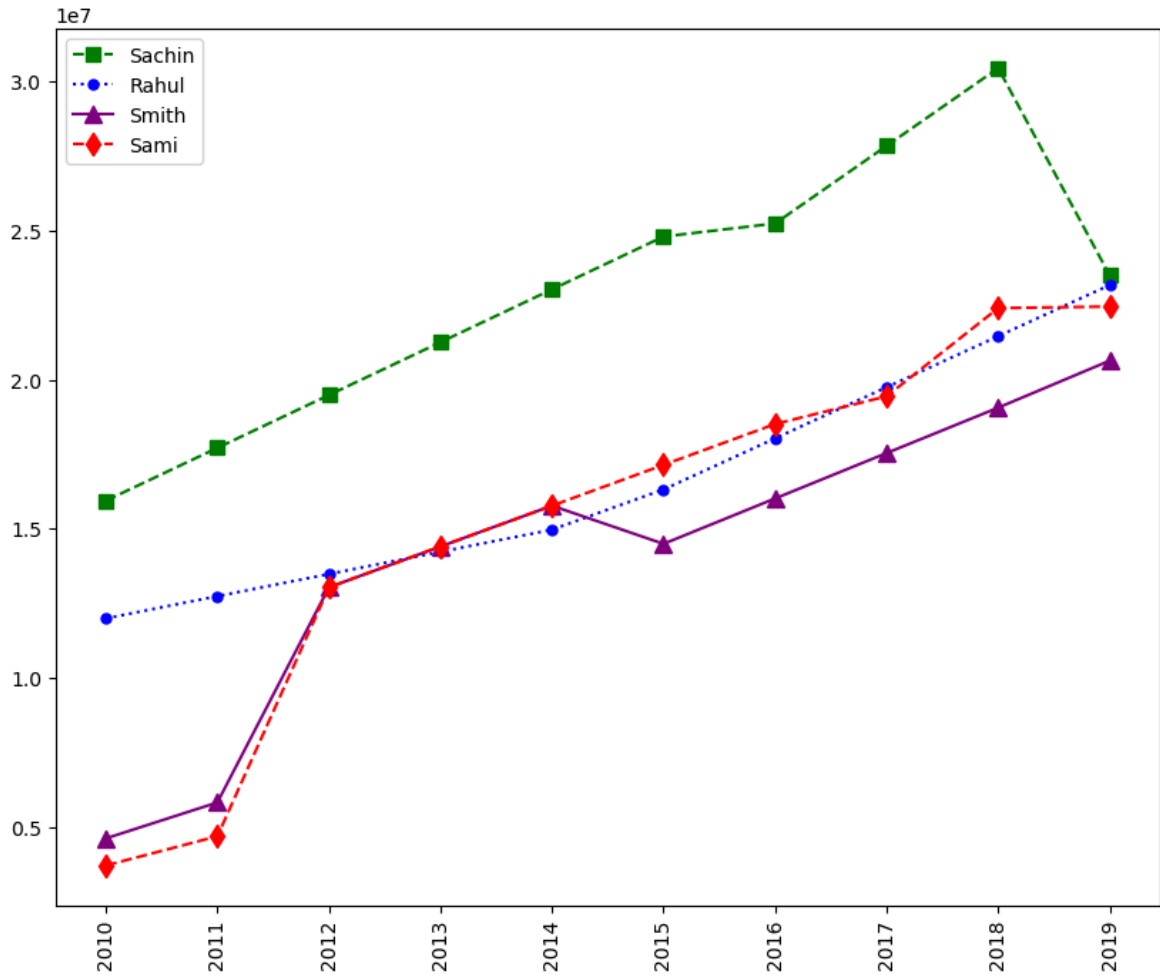
plt.show()
```



```
In [237... # how to add legend 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')

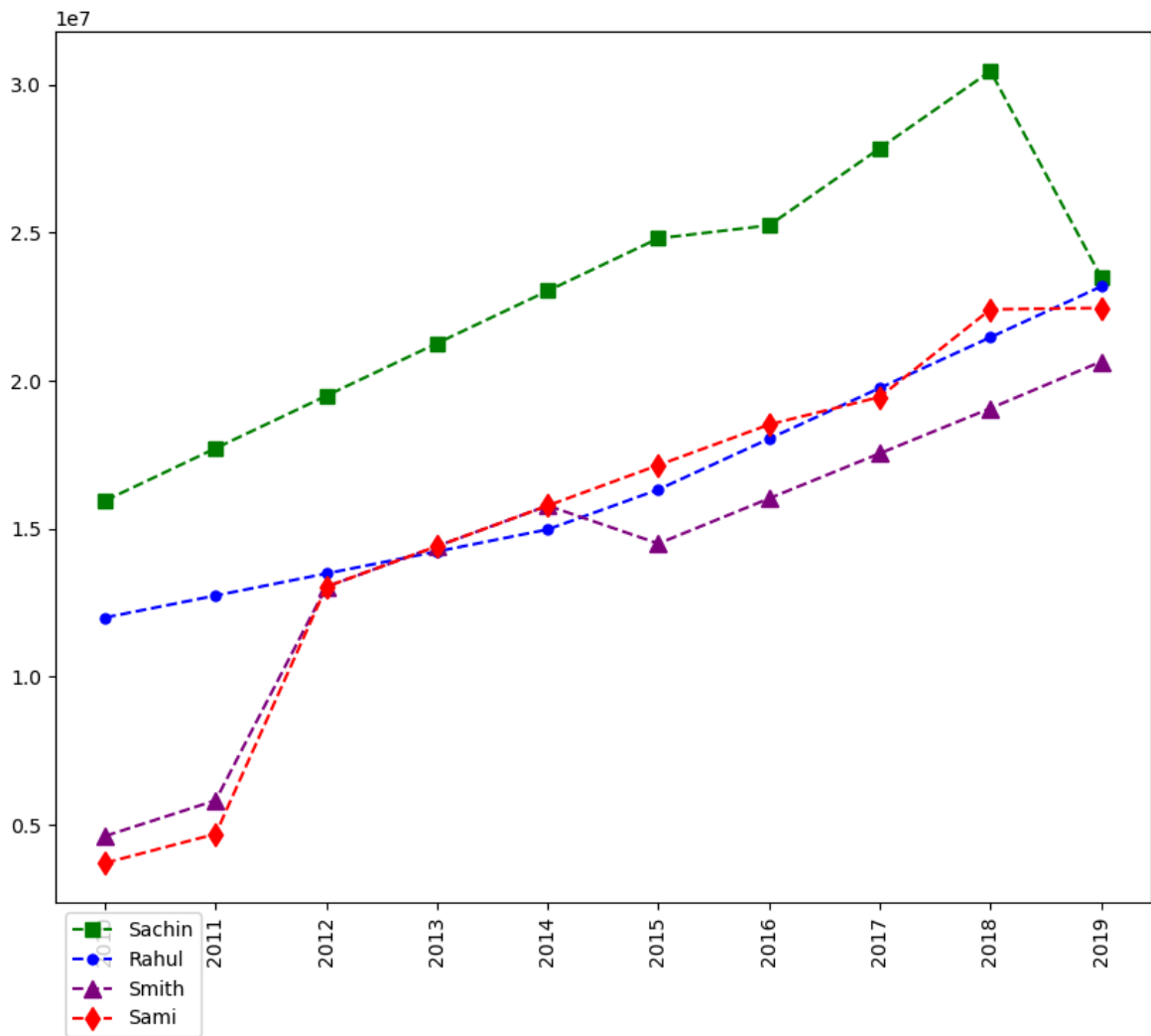
plt.show()
```



In [239...

```
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')

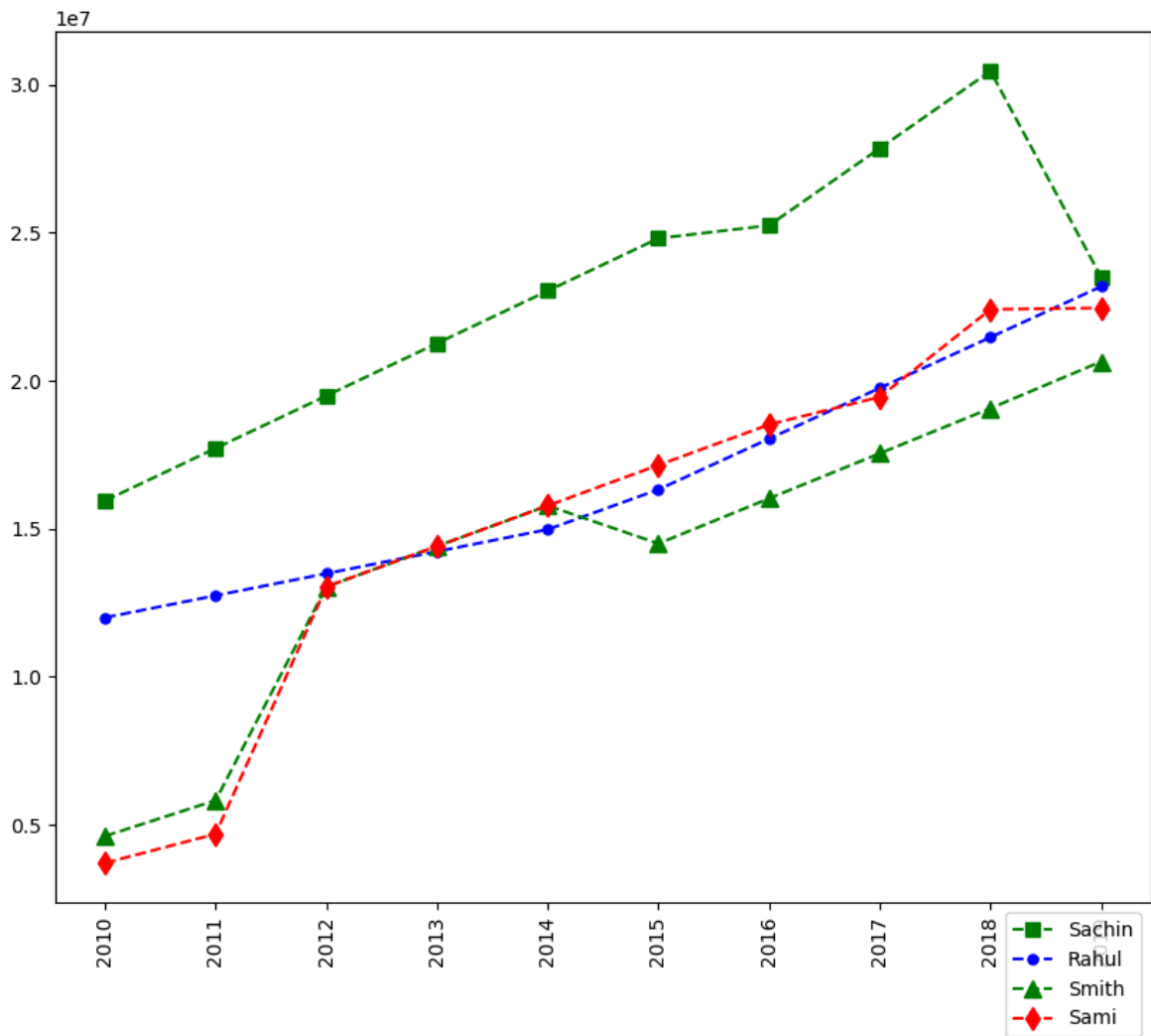
plt.show()
```



In [241...

```
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')

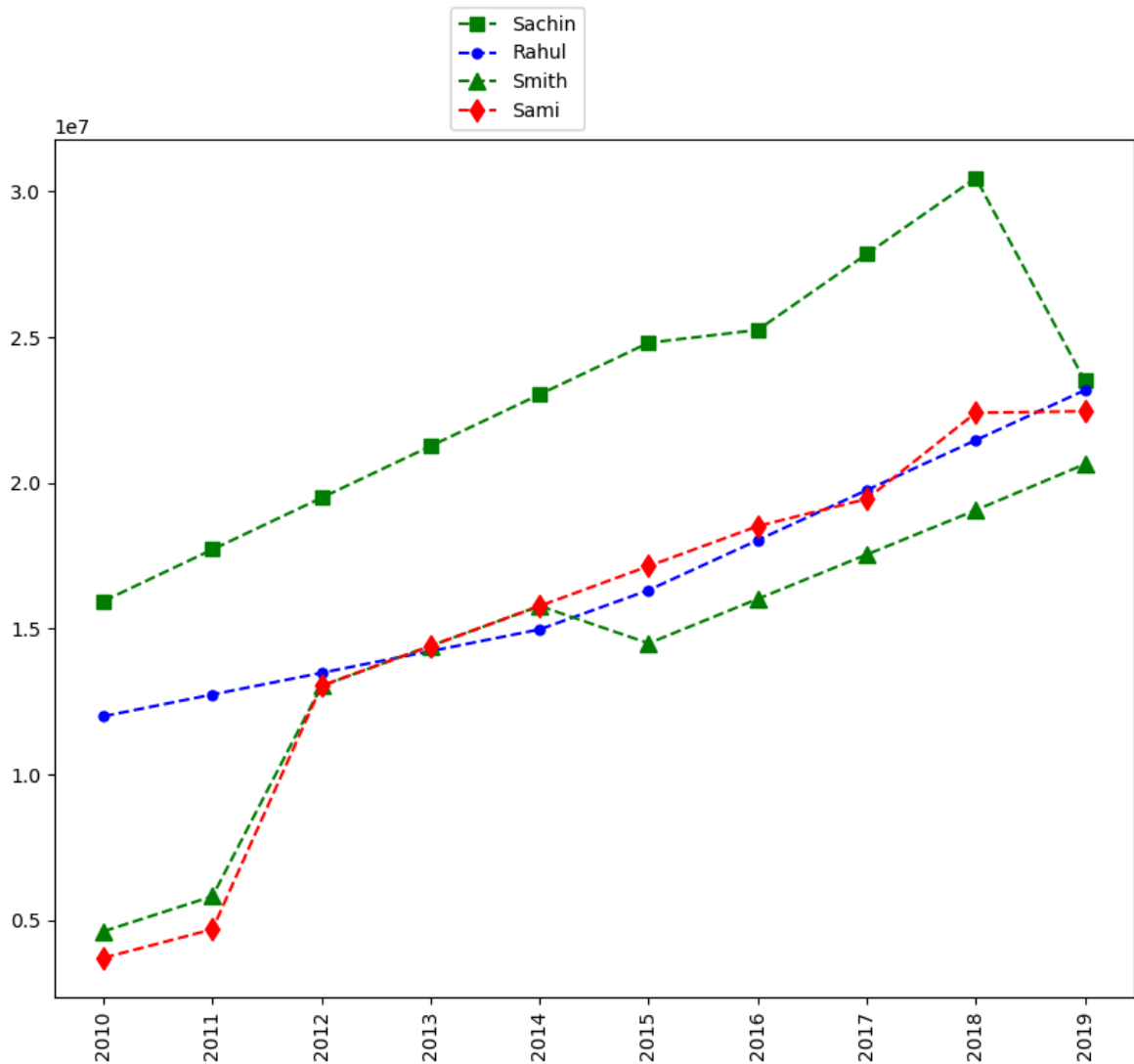
plt.show()
```



In [243...

```
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')

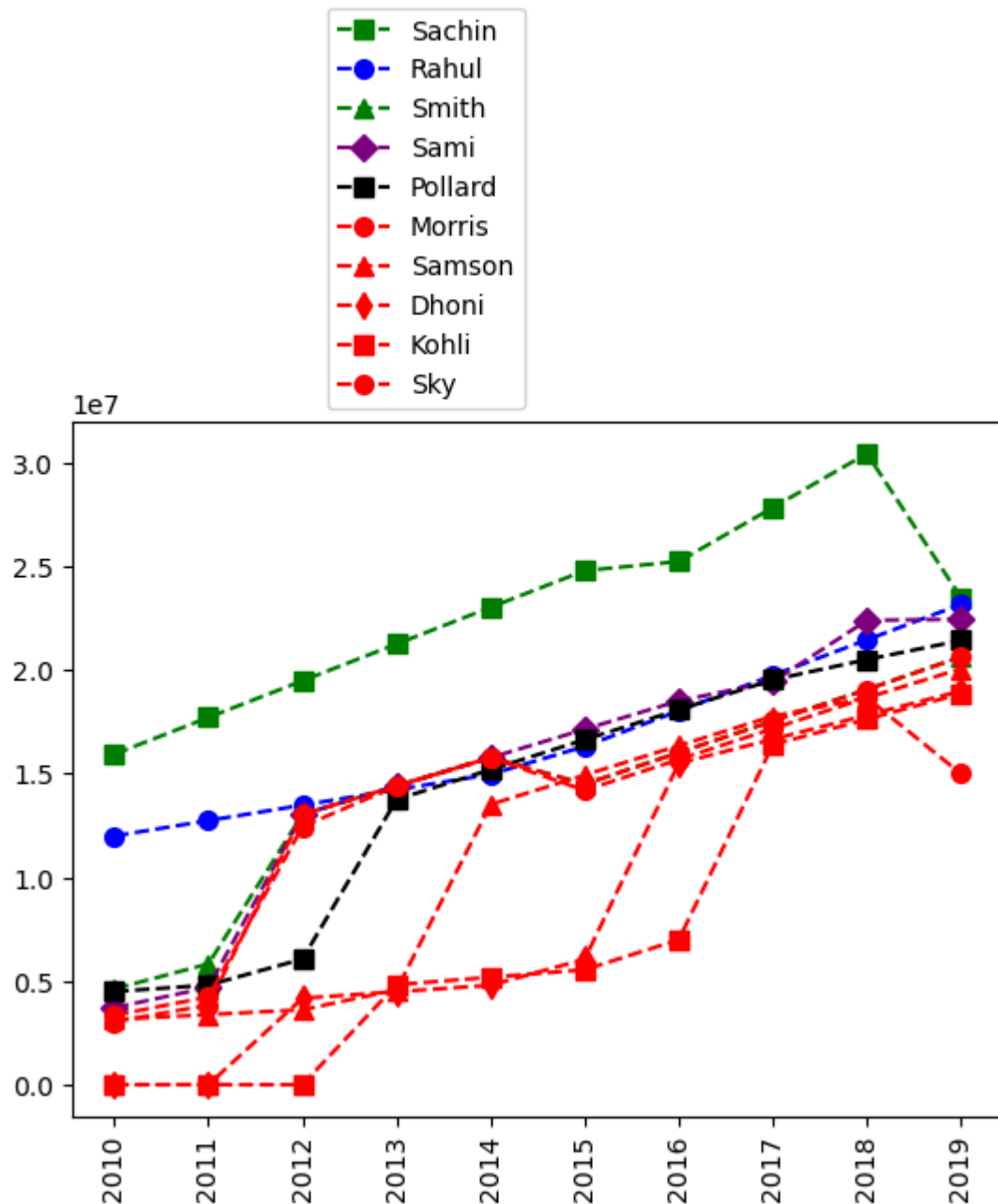
plt.show()
```



```
In [13]: 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()
```

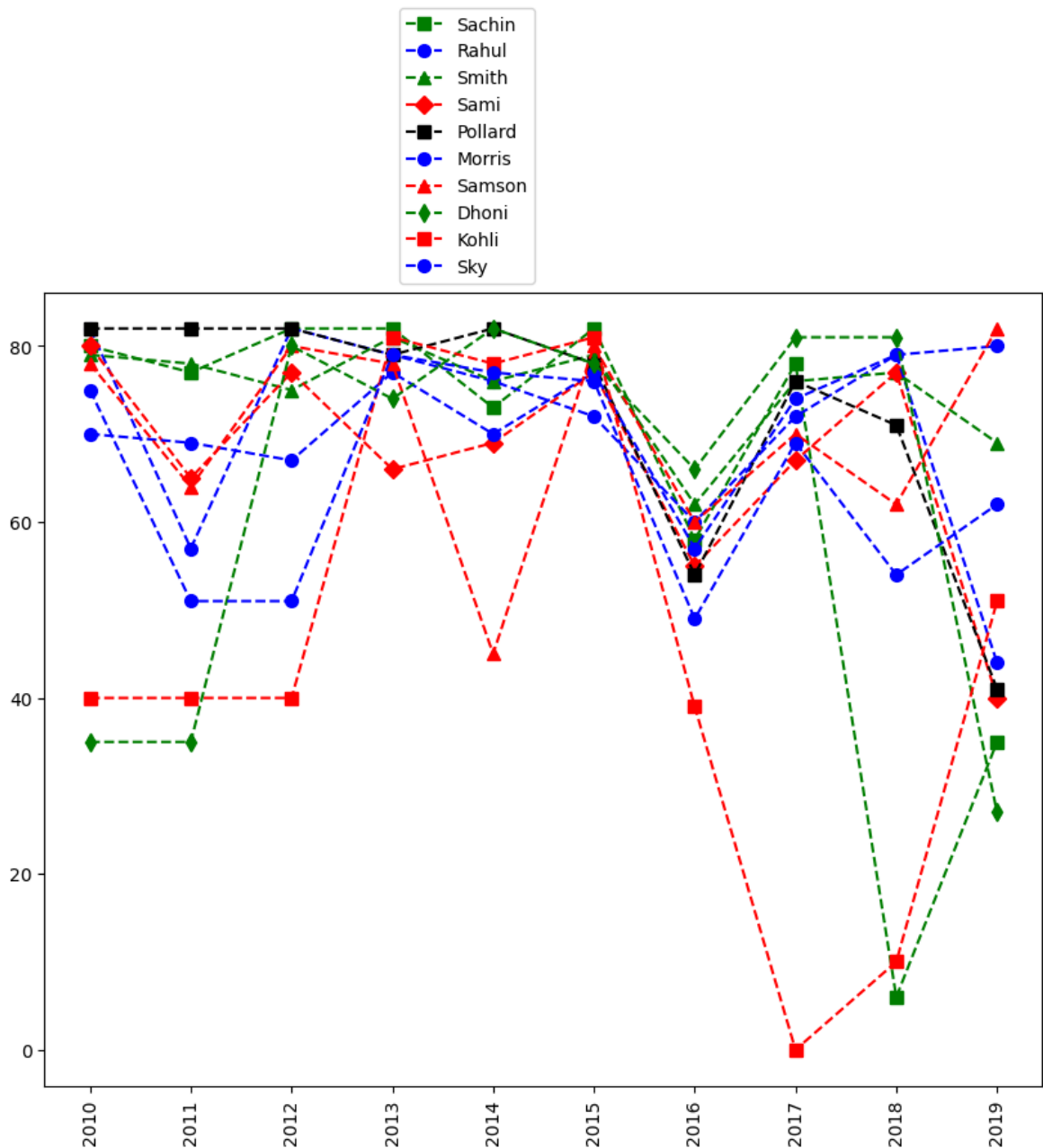


In [247... *# we can visualize the how many games played by a player*

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

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

plt.show()
```



\* In 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 [ ]: