

```
In [2]: #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,"2018":8,"2019":9}

#Players
Players = ["Sachin","Rahul","Smith","Sami","Pollard","Morris","Samson","Dhoni","Kohli","Sky"]
Pdict = {"Sachin":0,"Rahul":1,"Smith":2,"Sami":3,"Pollard":4,"Morris":5,"Samson":6,"Dhoni":7,"Kohli":8,"Sky":9}

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

#Games
Sachin_G = [80,77,82,82,73,82,58,78,6,35]
Rahul_G = [82,57,82,79,76,72,60,72,79,80]
Smith_G = [79,78,75,81,76,79,62,76,77,69]
Sami_G = [80,65,77,66,69,77,55,67,77,40]
Pollard_G = [82,82,82,79,82,78,54,76,71,41]
Morris_G = [70,69,67,77,70,77,57,74,79,44]
Samson_G = [78,64,80,78,45,80,60,70,62,82]
Dhoni_G = [35,35,80,74,82,78,66,81,81,27]
Kohli_G = [40,40,40,81,78,81,39,0,10,51]
Sky_G = [75,51,51,79,77,76,49,69,54,62]
#Matrix
Games = np.array([Sachin_G, Rahul_G, Smith_G, Sami_G, Pollard_G, Morris_G, Samson_G, Dhoni_G, Kohli_G, Sky_G])

#Points
Sachin PTS = [2832,2430,2323,2201,1970,2078,1616,2133,83,782]
Rahul PTS = [1653,1426,1779,1688,1619,1312,1129,1170,1245,1154]
Smith PTS = [2478,2132,2250,2304,2258,2111,1683,2036,2089,1743]
Sami PTS = [2122,1881,1978,1504,1943,1970,1245,1920,2112,966]
Pollard PTS = [1292,1443,1695,1624,1503,1784,1113,1296,1297,646]
Morris PTS = [1572,1561,1496,1746,1678,1438,1025,1232,1281,928]
Samson PTS = [1258,1104,1684,1781,841,1268,1189,1186,1185,1564]
Dhoni PTS = [903,903,1624,1871,2472,2161,1850,2280,2593,686]
Kohli PTS = [597,597,597,1361,1619,2026,852,0,159,904]
Sky PTS = [2040,1397,1254,2386,2045,1941,1082,1463,1028,1331]
#Matrix
Points = np.array([Sachin PTS, Rahul PTS, Smith PTS, Sami PTS, Pollard PTS, Morris PTS, Samson PTS, Dhoni PTS, Kohli PTS, Sky PTS])
```

In [3]: Salary

```
Out[3]: array([[15946875, 17718750, 19490625, 21262500, 23034375, 24806250,
   25244493, 27849149, 30453805, 23500000],
   [12000000, 12744189, 13488377, 14232567, 14976754, 16324500,
   18038573, 19752645, 21466718, 23180790],
   [ 4621800,  5828090, 13041250, 14410581, 15779912, 14500000,
  16022500, 17545000, 19067500, 20644400],
   [ 3713640,  4694041, 13041250, 14410581, 15779912, 17149243,
  18518574, 19450000, 22407474, 22458000],
   [ 4493160,  4806720, 6061274, 13758000, 15202590, 16647180,
  18091770, 19536360, 20513178, 21436271],
   [ 3348000,  4235220, 12455000, 14410581, 15779912, 14500000,
  16022500, 17545000, 19067500, 20644400],
   [ 3144240,  3380160, 3615960, 4574189, 13520500, 14940153,
  16359805, 17779458, 18668431, 20068563],
   [     0,         0, 4171200, 4484040, 4796880, 6053663,
  15506632, 16669630, 17832627, 18995624],
   [     0,         0,         0, 4822800, 5184480, 5546160,
  6993708, 16402500, 17632688, 18862875],
   [ 3031920,  3841443, 13041250, 14410581, 15779912, 14200000,
  15691000, 17182000, 18673000, 15000000]])
```

In [4]: Games

```
Out[4]: array([[80, 77, 82, 82, 73, 82, 58, 78, 6, 35],
   [82, 57, 82, 79, 76, 72, 60, 72, 79, 80],
   [79, 78, 75, 81, 76, 79, 62, 76, 77, 69],
   [80, 65, 77, 66, 69, 77, 55, 67, 77, 40],
   [82, 82, 82, 79, 82, 78, 54, 76, 71, 41],
   [70, 69, 67, 77, 70, 77, 57, 74, 79, 44],
   [78, 64, 80, 78, 45, 80, 60, 70, 62, 82],
   [35, 35, 80, 74, 82, 78, 66, 81, 81, 27],
   [40, 40, 40, 81, 78, 81, 39, 0, 10, 51],
   [75, 51, 51, 79, 77, 76, 49, 69, 54, 62]]))
```

In [5]: Points

```
Out[5]: array([[2832, 2430, 2323, 2201, 1970, 2078, 1616, 2133, 83, 782],
   [1653, 1426, 1779, 1688, 1619, 1312, 1129, 1170, 1245, 1154],
   [2478, 2132, 2250, 2304, 2258, 2111, 1683, 2036, 2089, 1743],
   [2122, 1881, 1978, 1504, 1943, 1970, 1245, 1920, 2112, 966],
   [1292, 1443, 1695, 1624, 1503, 1784, 1113, 1296, 1297, 646],
   [1572, 1561, 1496, 1746, 1678, 1438, 1025, 1232, 1281, 928],
   [1258, 1104, 1684, 1781, 841, 1268, 1189, 1186, 1185, 1564],
   [ 903,  903, 1624, 1871, 2472, 2161, 1850, 2280, 2593, 686],
   [ 597,  597,  597, 1361, 1619, 2026, 852, 0, 159, 904],
   [2040, 1397, 1254, 2386, 2045, 1941, 1082, 1463, 1028, 1331]]))
```

In [7]: 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 [8]: MATR1 = np.reshape(mydata, (5,4), order = 'c')  
MATR1

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

```
In [9]: MATR1
```

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

```
In [10]: MATR1[4,3]
```

```
Out[10]: 19
```

```
In [11]: MATR1[3,3]
```

```
Out[11]: 15
```

```
In [12]: MATR1
```

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

```
In [13]: MATR1[-3,-1]
```

```
Out[13]: 11
```

```
In [14]: MATR1
```

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

```
In [15]: mydata
```

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

```
In [16]: MATR2 = np.reshape(mydata, (5,4), order = 'F') # reshape behaviour are - 'C', 'F'
MATR2
```

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

```
In [17]: MATR2[4,3]
```

```
Out[17]: 19
```

```
In [18]: MATR2[0,2]
```

```
Out[18]: 10
```

```
In [19]: Games
```

```
Out[19]: 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 [20]: Games[0]
```

```
Out[20]: array([80, 77, 82, 82, 73, 82, 58, 78, 6, 35])
```

```
In [21]: Games[5]
```

```
Out[21]: array([70, 69, 67, 77, 70, 77, 57, 74, 79, 44])
```

```
In [22]: Games[0:5]
```

```
Out[22]: 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 [23]: Games[0,5]
```

```
Out[23]: 82
```

```
In [24]: Games[0,2]
```

```
Out[24]: 82
```

```
In [25]: Games
```

```
Out[25]: 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 [26]: Games[1:2]
```

```
Out[26]: array([[82, 57, 82, 79, 76, 72, 60, 72, 79, 80]])
```

```
In [27]: Games[2]
```

```
Out[27]: array([79, 78, 75, 81, 76, 79, 62, 76, 77, 69])
```

In [28]: Games

```
Out[28]: 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 [29]: Games[-3:-1]

```
Out[29]: array([[35, 35, 80, 74, 82, 78, 66, 81, 81, 27],  
[40, 40, 40, 81, 78, 81, 39, 0, 10, 51]])
```

In [30]: Games[-3,-1]

```
Out[30]: 27
```

In [31]: Points

```
Out[31]: 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 [32]: Points[0]

```
Out[32]: array([2832, 2430, 2323, 2201, 1970, 2078, 1616, 2133, 83, 782])
```

In [33]: Points

```
Out[33]: 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 [34]: Points[6,1]

```
Out[34]: 1104
```

In [35]: Points

```
Out[35]: 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 [36]: Points[-6,-1]
```

```
Out[36]: 646
```

```
In [37]: Pdict
```

```
Out[37]: {'Sachin': 0,  
 'Rahul': 1,  
 'Smith': 2,  
 'Sami': 3,  
 'Pollard': 4,  
 'Morris': 5,  
 'Samson': 6,  
 'Dhoni': 7,  
 'Kohli': 8,  
 'Sky': 9}
```

```
In [38]: Pdict['Sachin']
```

```
Out[38]: 0
```

```
In [39]: Games[0]
```

```
Out[39]: array([80, 77, 82, 82, 73, 82, 58, 78, 6, 35])
```

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

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

```
In [41]: Pdict['Rahul']
```

```
Out[41]: 1
```

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

```
Out[42]: 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 [43]: Games[Pdict['Rahul']]

```
Out[43]: array([82, 57, 82, 79, 76, 72, 60, 72, 79, 80])
```

In [44]: Points

```
Out[44]: 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 [45]: Salary

```
Out[45]: 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 [46]: Salary[2,4]

```
Out[46]: 15779912
```

In [47]: Salary

```
Out[47]: 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 [48]: Salary[Pdict['Sky']][Sdict['2019']]
```

```
Out[48]: 15000000
```

```
In [49]: Salary
```

```
Out[49]: 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 [50]: Games
```

```
Out[50]: 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 [51]: `Salary/Games`

```
C:\Users\harisai\AppData\Local\Temp\ipykernel_23168\3709746658.py:1: RuntimeWarning: divide by zero encountered in divide
    Salary/Games
```

```
Out[51]: 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 [52]: `np.round(Salary/Games)`

```
C:\Users\harisai\AppData\Local\Temp\ipykernel_23168\2909567671.py:1: RuntimeWarning:  
ng: divide by zero encountered in divide  
    np.round(Salary/Games)
```

```
Out[52]: 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.]])
```

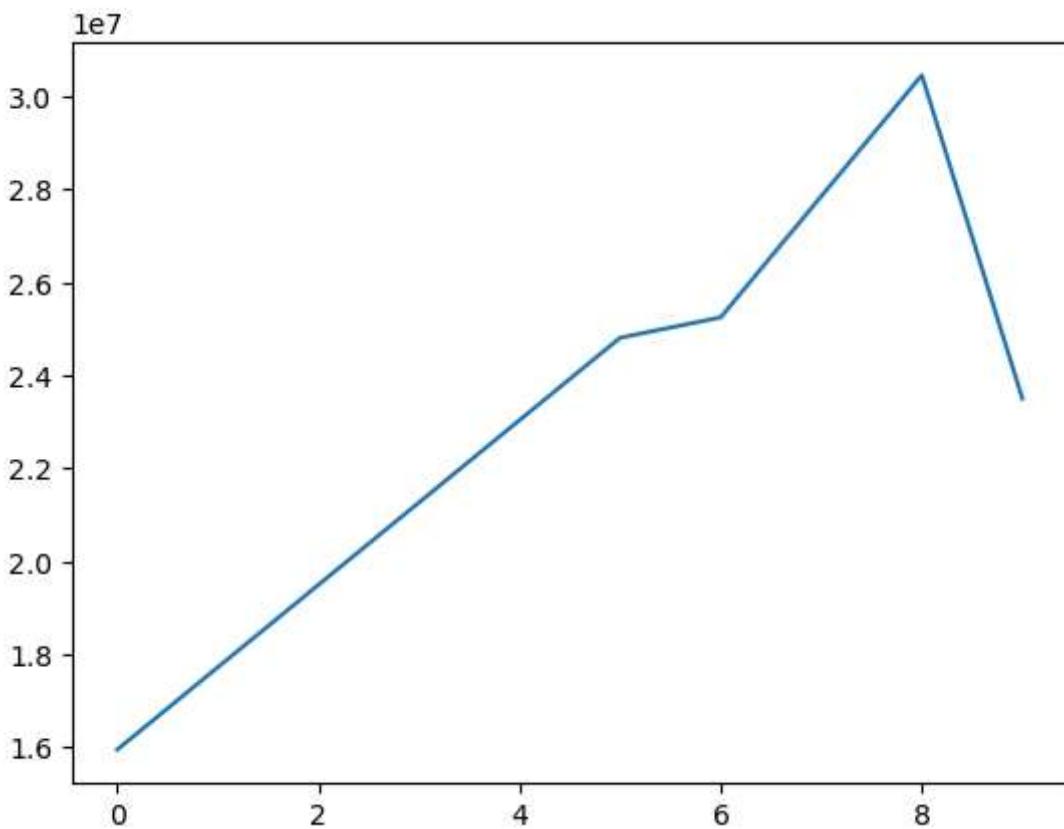
```
In [53]: import warnings  
warnings.filterwarnings('ignore')
```

```
In [58]: import numpy as np  
import matplotlib.pyplot as plt
```

```
In [59]: %matplotlib inline
```

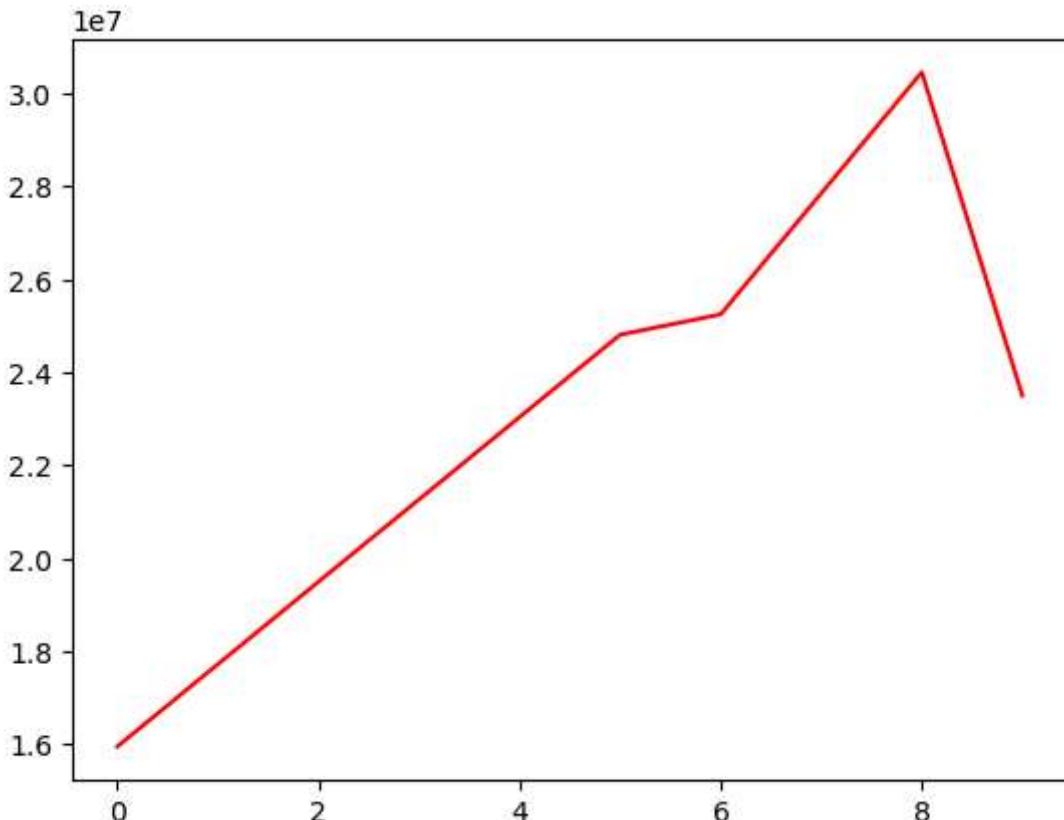
```
In [60]: plt.plot(Salary[0])
```

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



```
In [61]: plt.plot(Salary[0], c='red')
```

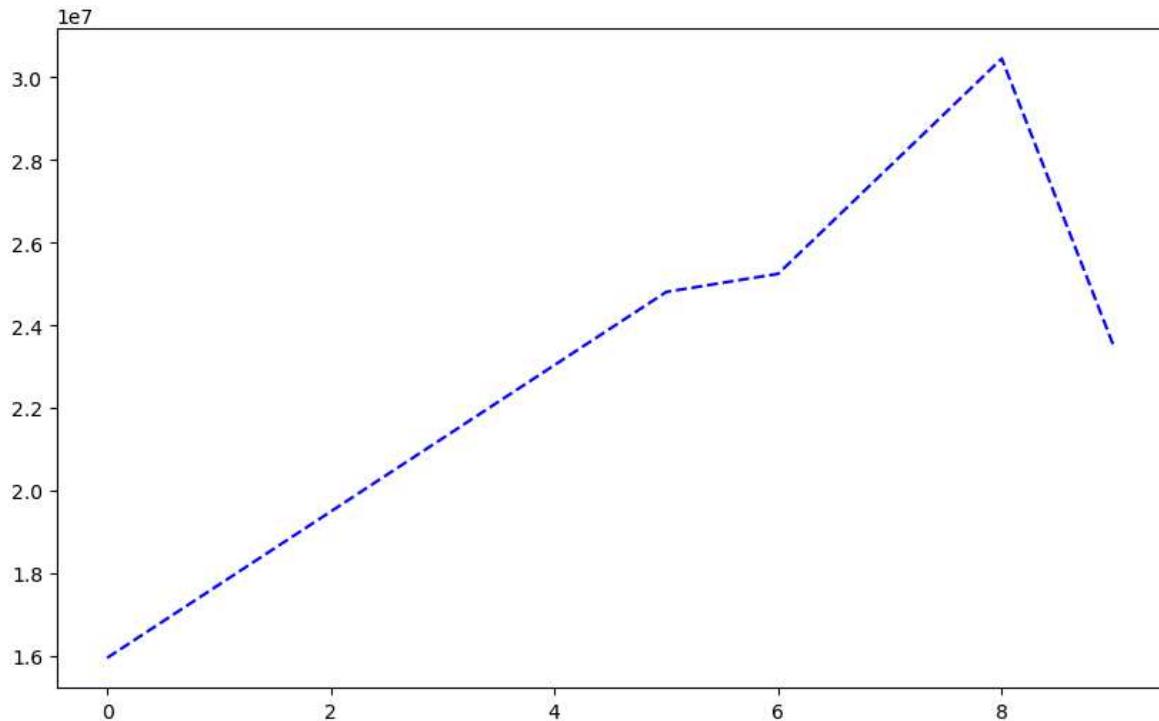
```
Out[61]: [
```



```
In [62]: %matplotlib inline  
plt.rcParams['figure.figsize'] = 10,6
```

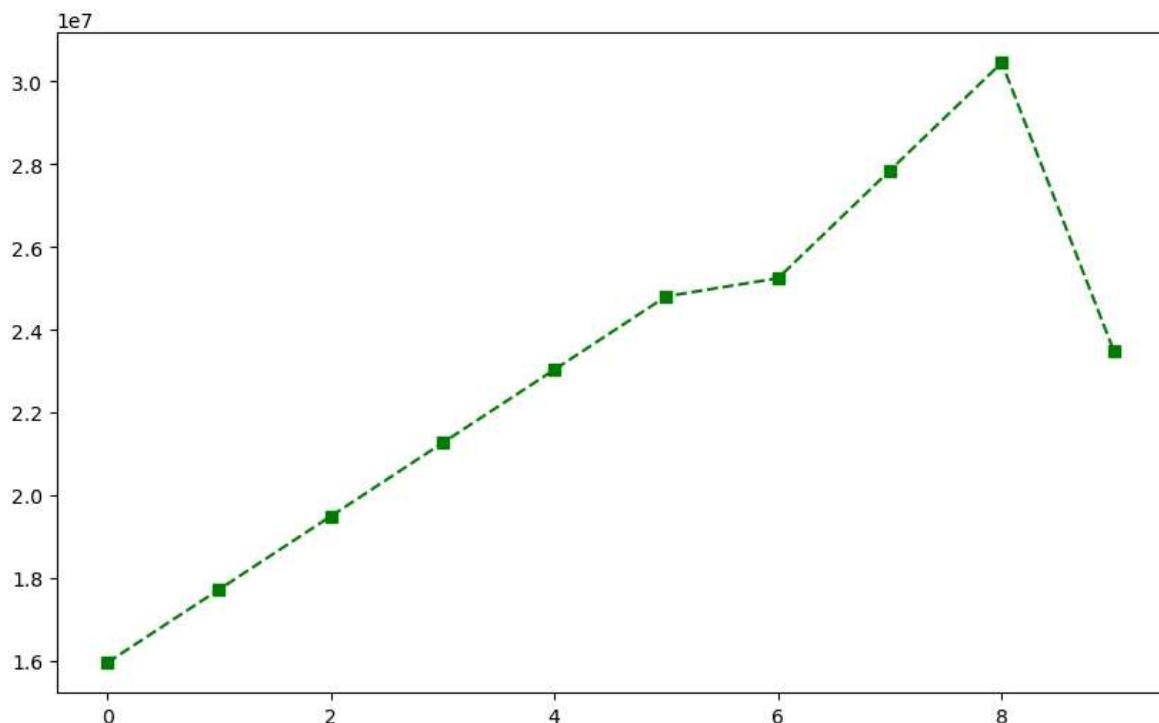
```
In [63]: plt.plot(Salary[0], c='Blue', ls = 'dashed')
```

```
Out[63]: <matplotlib.lines.Line2D at 0x1933288f230>
```



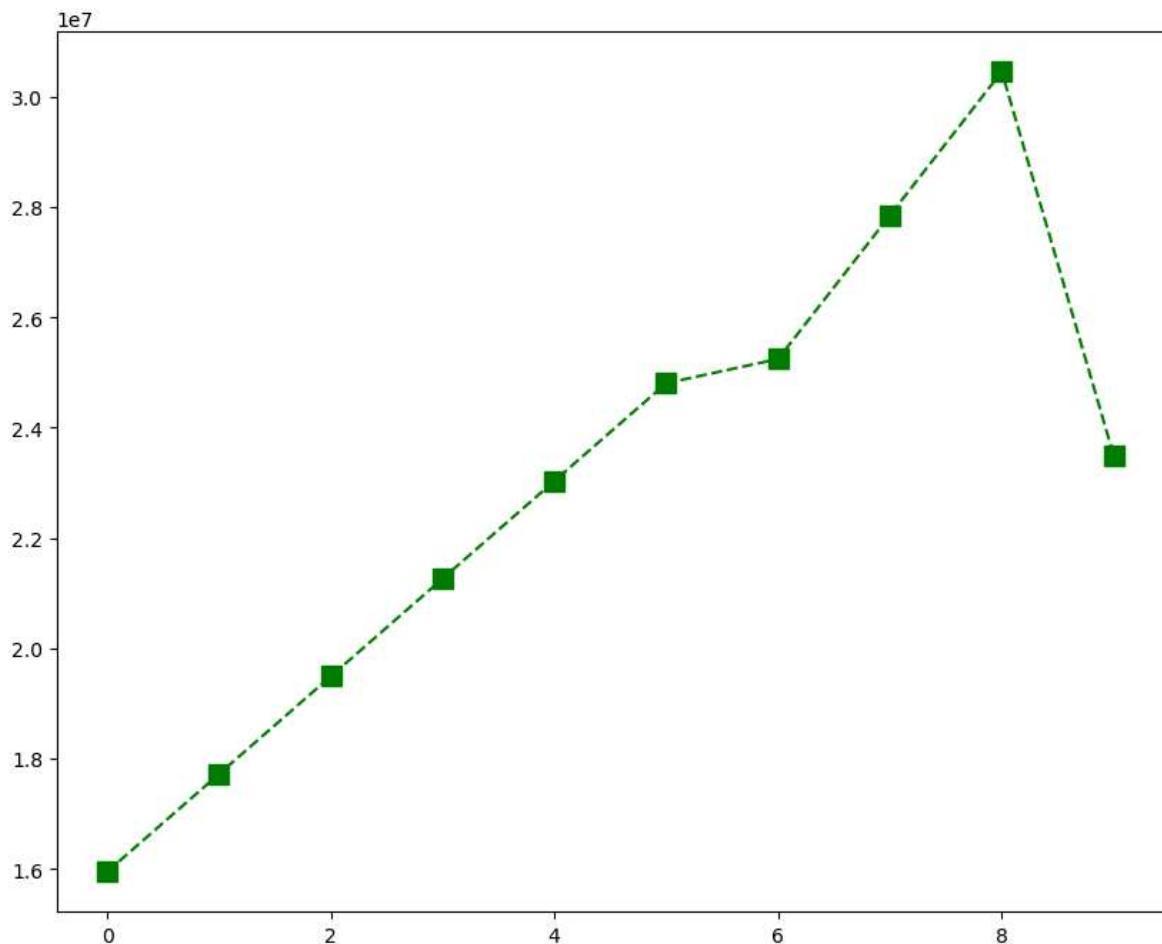
```
In [64]: plt.plot(Salary[0], c='Green', ls = '--', marker = 's')
```

```
Out[64]: <matplotlib.lines.Line2D at 0x1933291ce30>
```



```
In [65]: %matplotlib inline  
plt.rcParams['figure.figsize'] = 10,8
```

```
In [66]: plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 10)  
plt.show()
```



```
In [67]: list(range(0,10))
```

```
Out[67]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

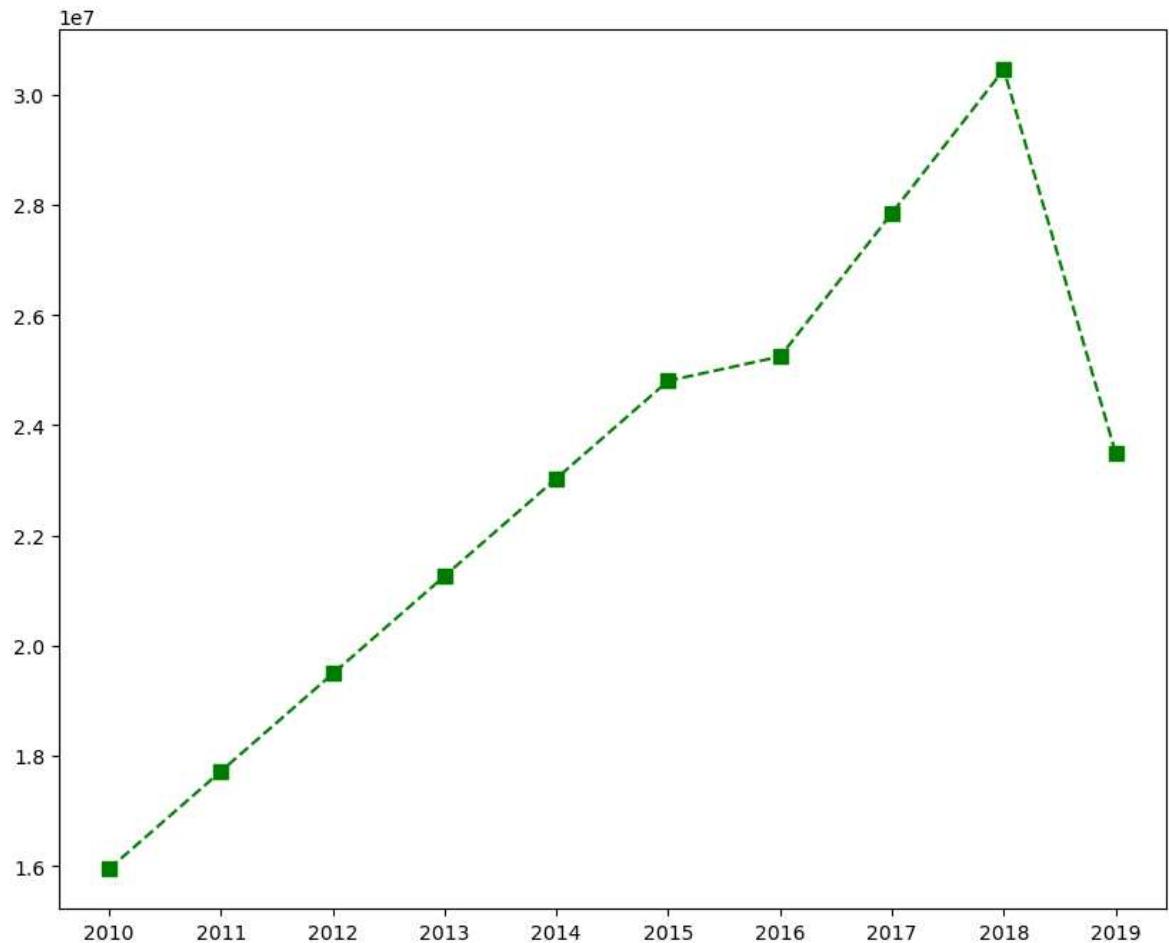
```
In [68]: Sdict
```

```
Out[68]: {'2010': 0,
          '2011': 1,
          '2012': 2,
          '2013': 3,
          '2014': 4,
          '2015': 5,
          '2016': 6,
          '2017': 7,
          '2018': 8,
          '2019': 9}
```

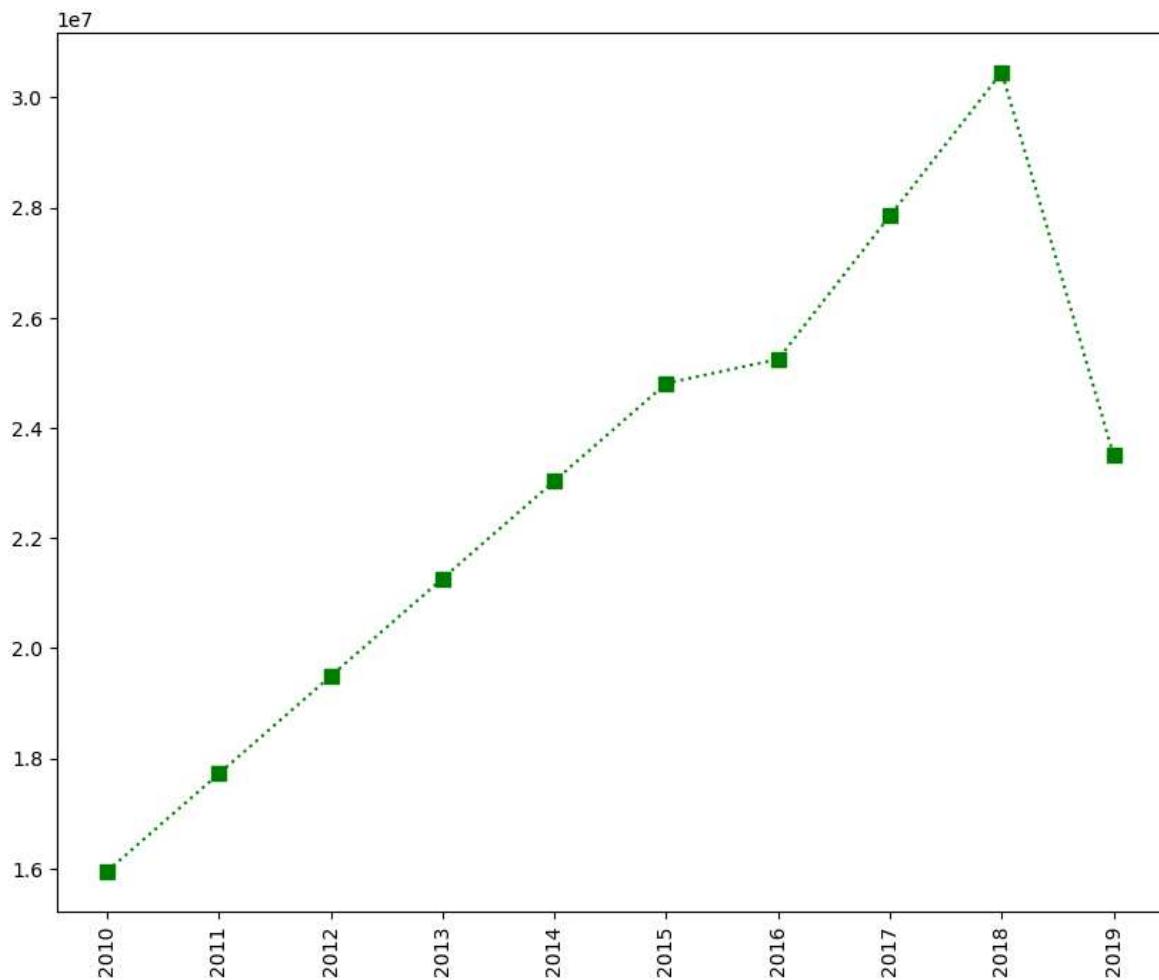
```
In [69]: Pdict
```

```
Out[69]: {'Sachin': 0,
          'Rahul': 1,
          'Smith': 2,
          'Sami': 3,
          'Pollard': 4,
          'Morris': 5,
          'Samson': 6,
          'Dhoni': 7,
          'Kohli': 8,
          'Sky': 9}
```

```
In [70]: plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7)
plt.xticks(list(range(0,10)), Seasons)
plt.show()
```



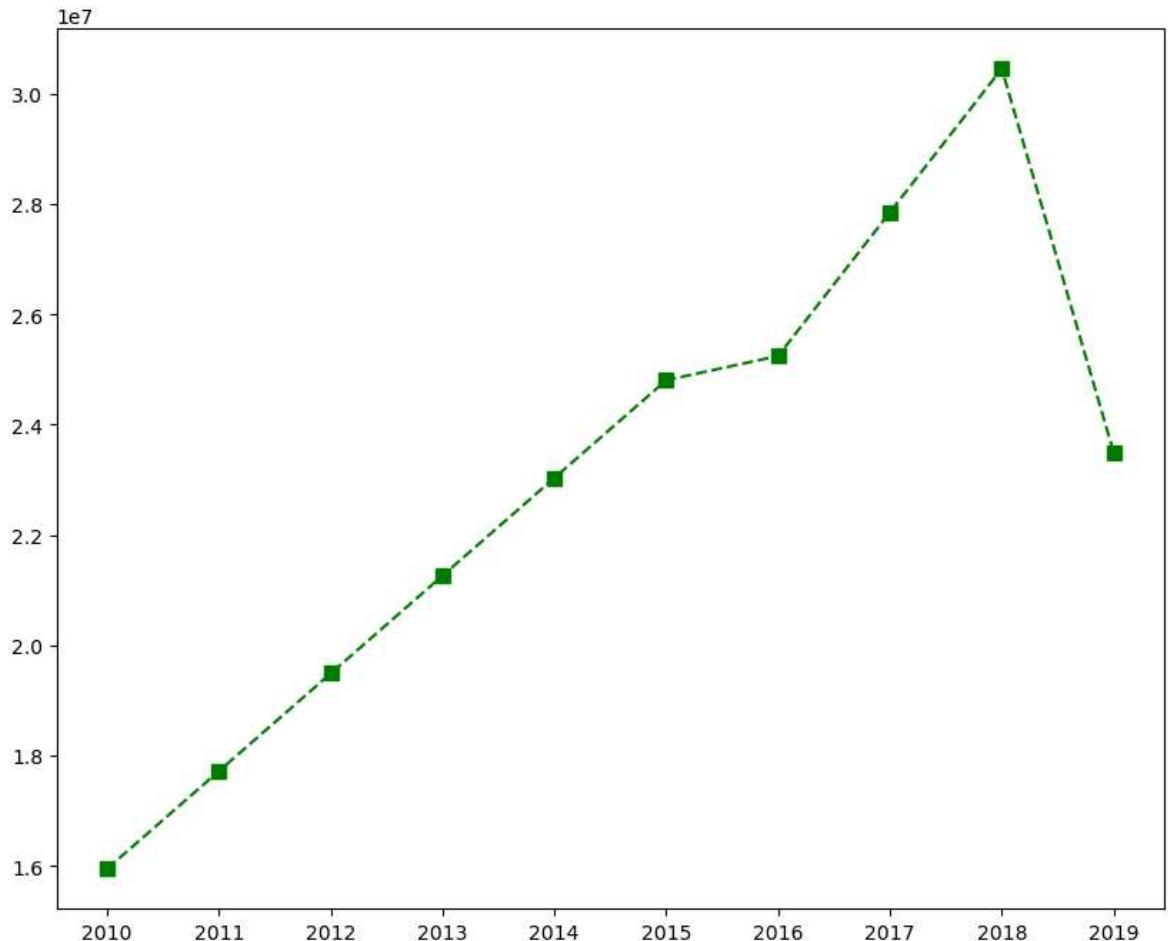
```
In [71]: plt.plot(Salary[0], c='Green', ls = ':', marker = 's', ms = 7, label = Players[0])
plt.xticks(list(range(0,10)), Seasons, rotation='vertical')
plt.show()
```



```
In [72]: Games
```

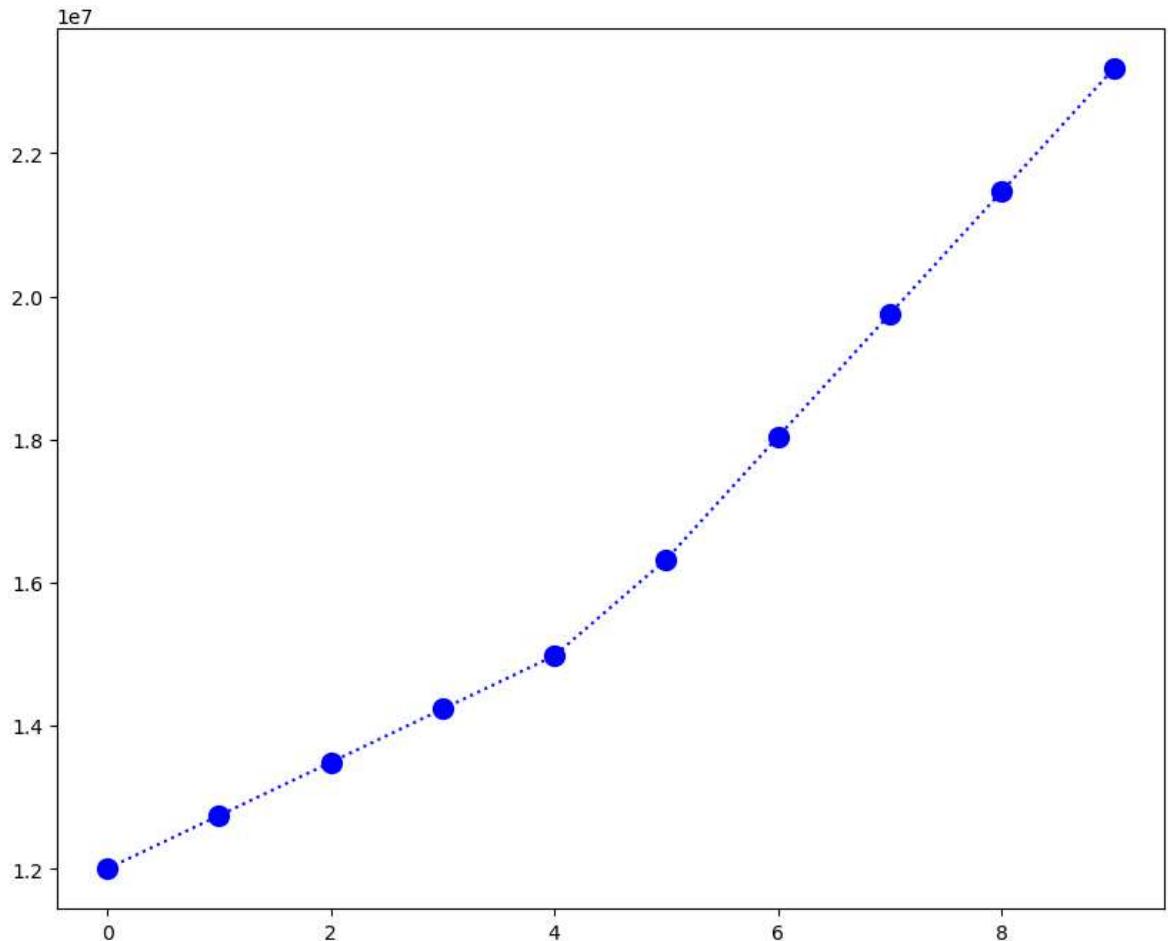
```
Out[72]: array([[80, 77, 82, 82, 73, 82, 58, 78, 6, 35],  
 [82, 57, 82, 79, 76, 72, 60, 72, 79, 80],  
 [79, 78, 75, 81, 76, 79, 62, 76, 77, 69],  
 [80, 65, 77, 66, 69, 77, 55, 67, 77, 40],  
 [82, 82, 82, 79, 82, 78, 54, 76, 71, 41],  
 [70, 69, 67, 77, 70, 77, 57, 74, 79, 44],  
 [78, 64, 80,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



```
In [74]: plt.plot(Salary[1], c='Blue', ls = ':', marker = 'o', ms = 10, label = Players[1]
```

```
Out[74]: [<matplotlib.lines.Line2D at 0x19331104080>]
```

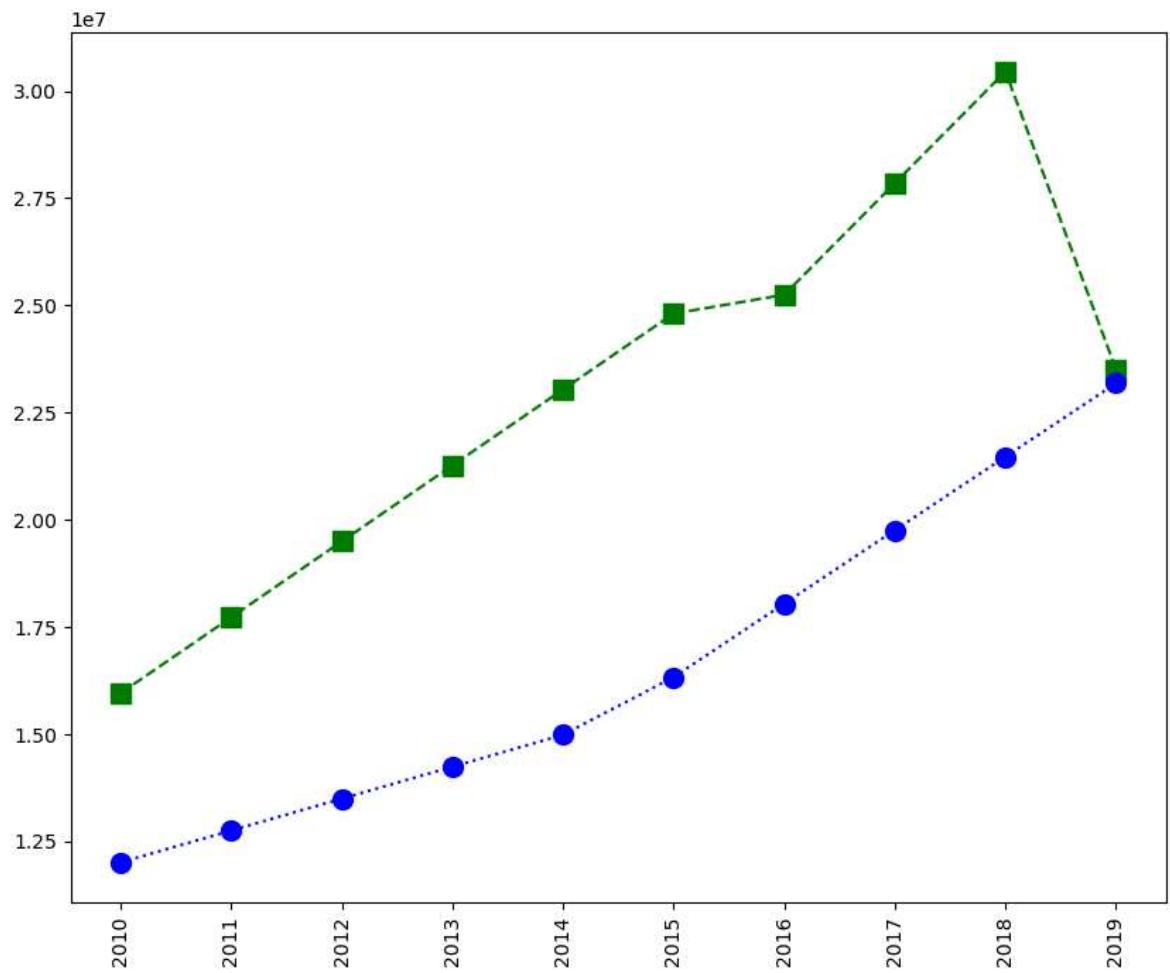


```
In [ ]: # More visualization
```

```
In [75]: 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')

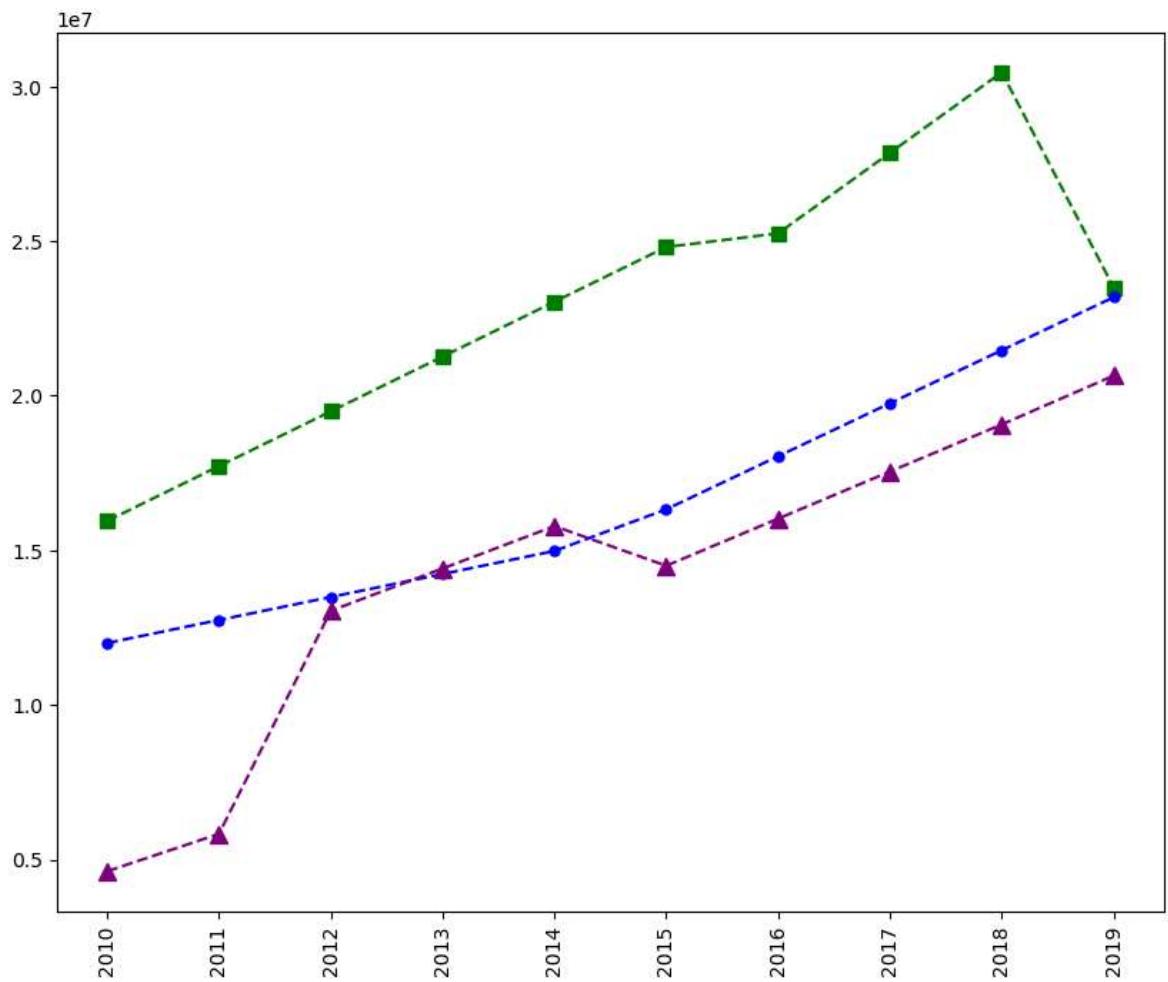
plt.show()
```



```
In [76]: 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')

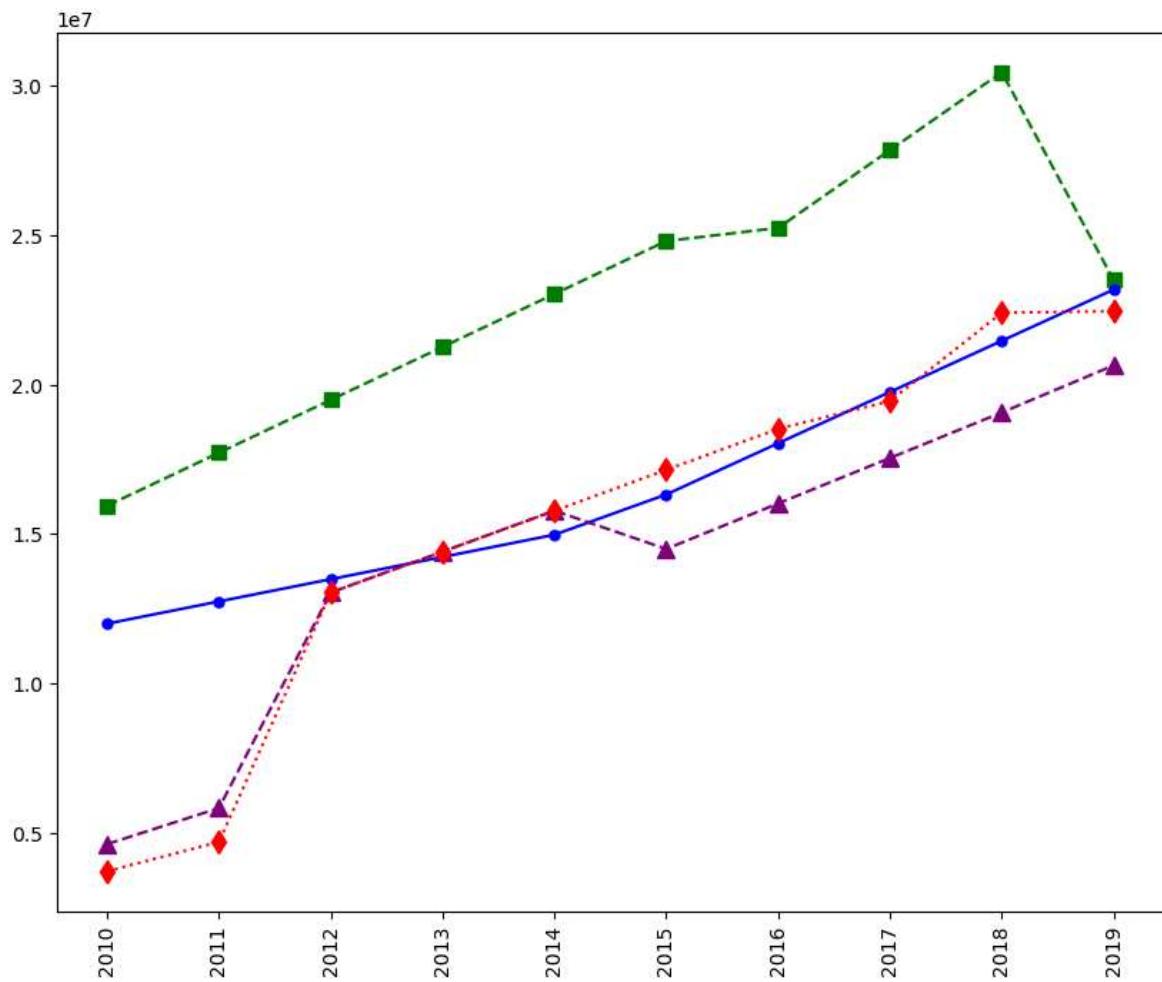
plt.show()
```



```
In [77]: 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])

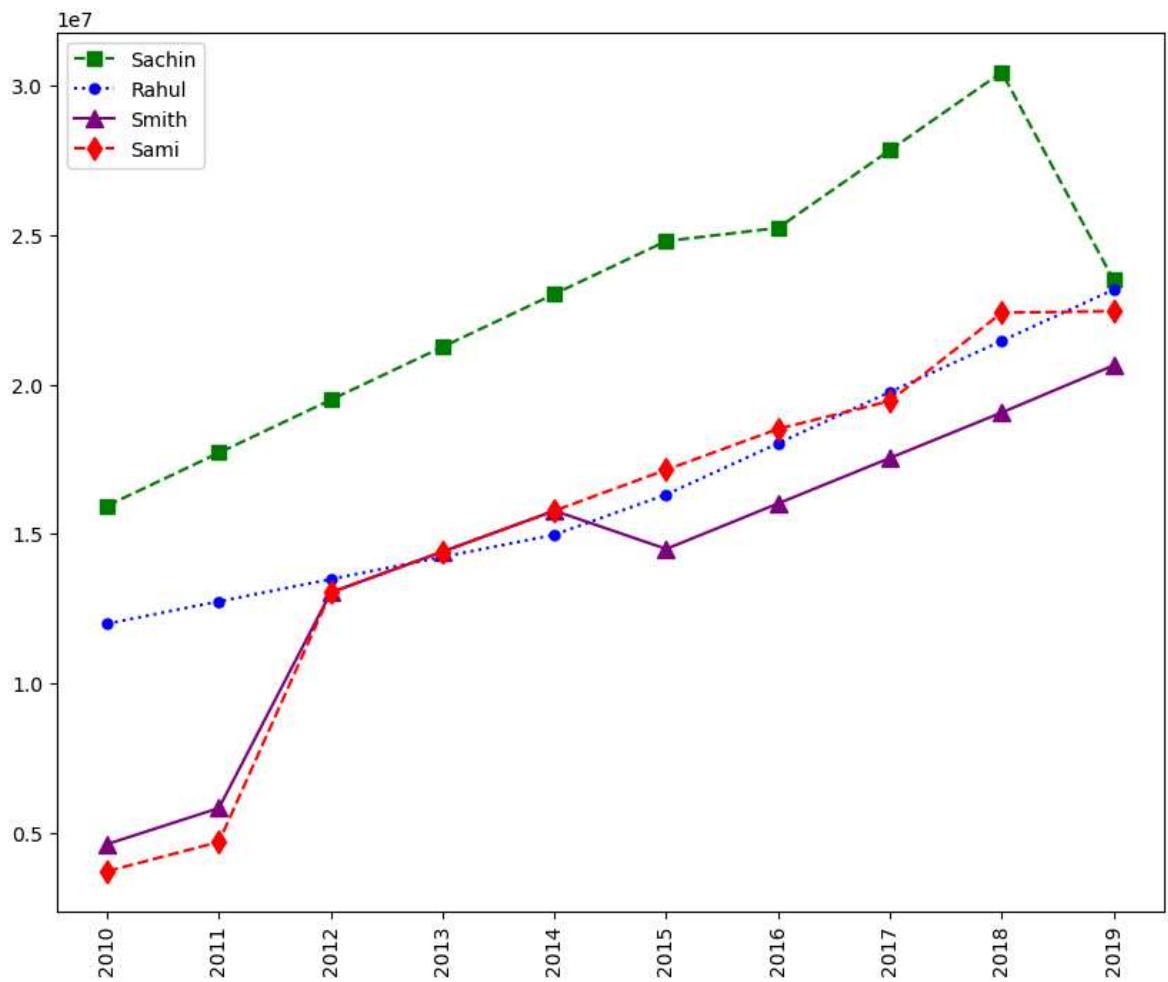
plt.xticks(list(range(0,10)), Seasons, rotation='vertical')

plt.show()
```



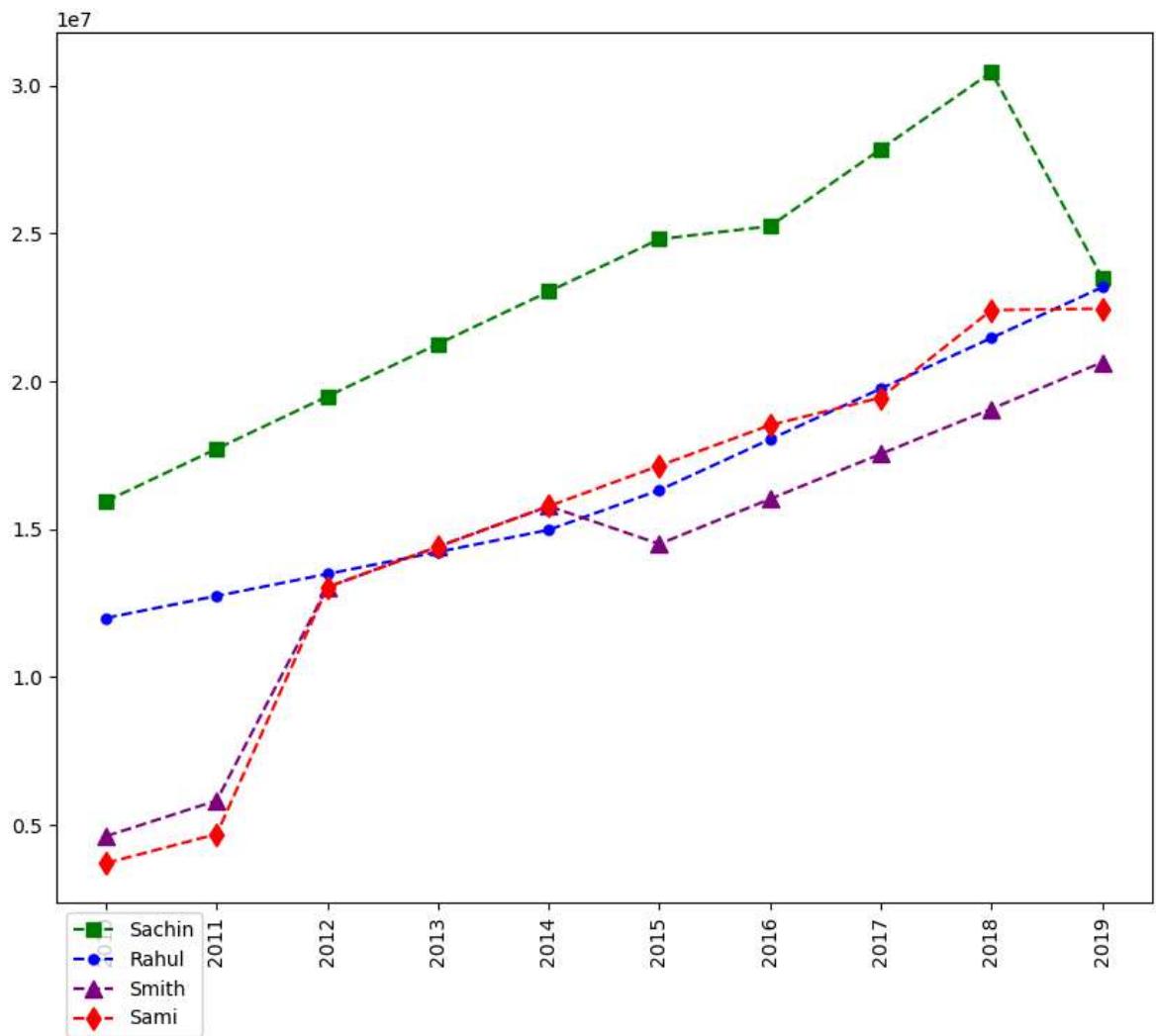
```
In [78]: plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Players[0])
plt.plot(Salary[1], c='Blue', ls = ':', marker = 'o', ms = 5, label = Players[1])
plt.plot(Salary[2], c='purple', ls = '-.', marker = '^', ms = 8, label = Players[2])
plt.plot(Salary[3], c='Red', ls = '---', marker = 'd', ms = 8, label = Players[3])
plt.legend()
plt.xticks(list(range(0,10)), Seasons, rotation='vertical')

plt.show()
```



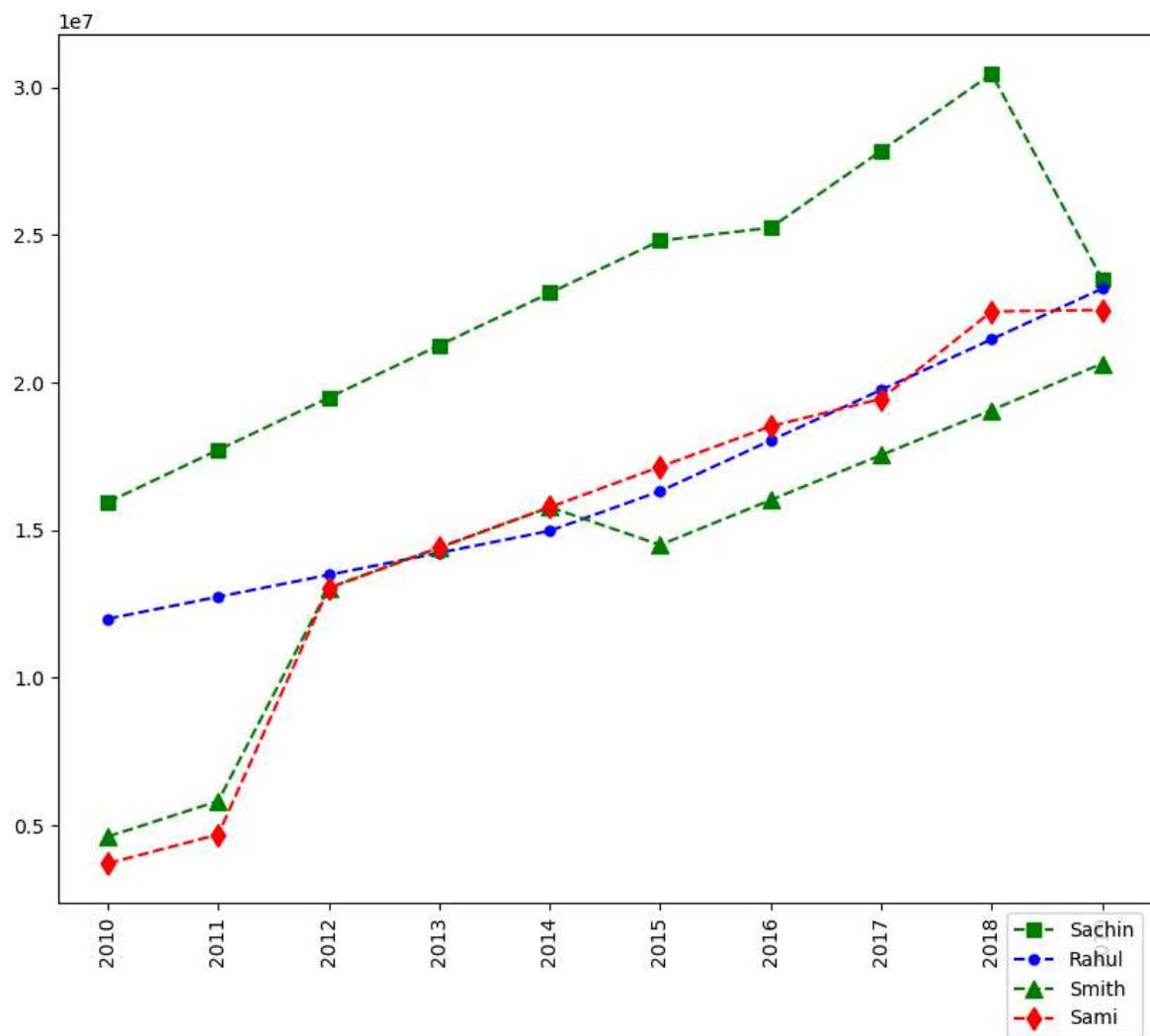
```
In [79]: plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Players[0])
plt.plot(Salary[1], c='Blue', ls = '--', marker = 'o', ms = 5, label = Players[1]
plt.plot(Salary[2], c='purple', ls = '--', marker = '^', ms = 8, label = Players[2]
plt.plot(Salary[3], c='Red', ls = '--', 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()
```



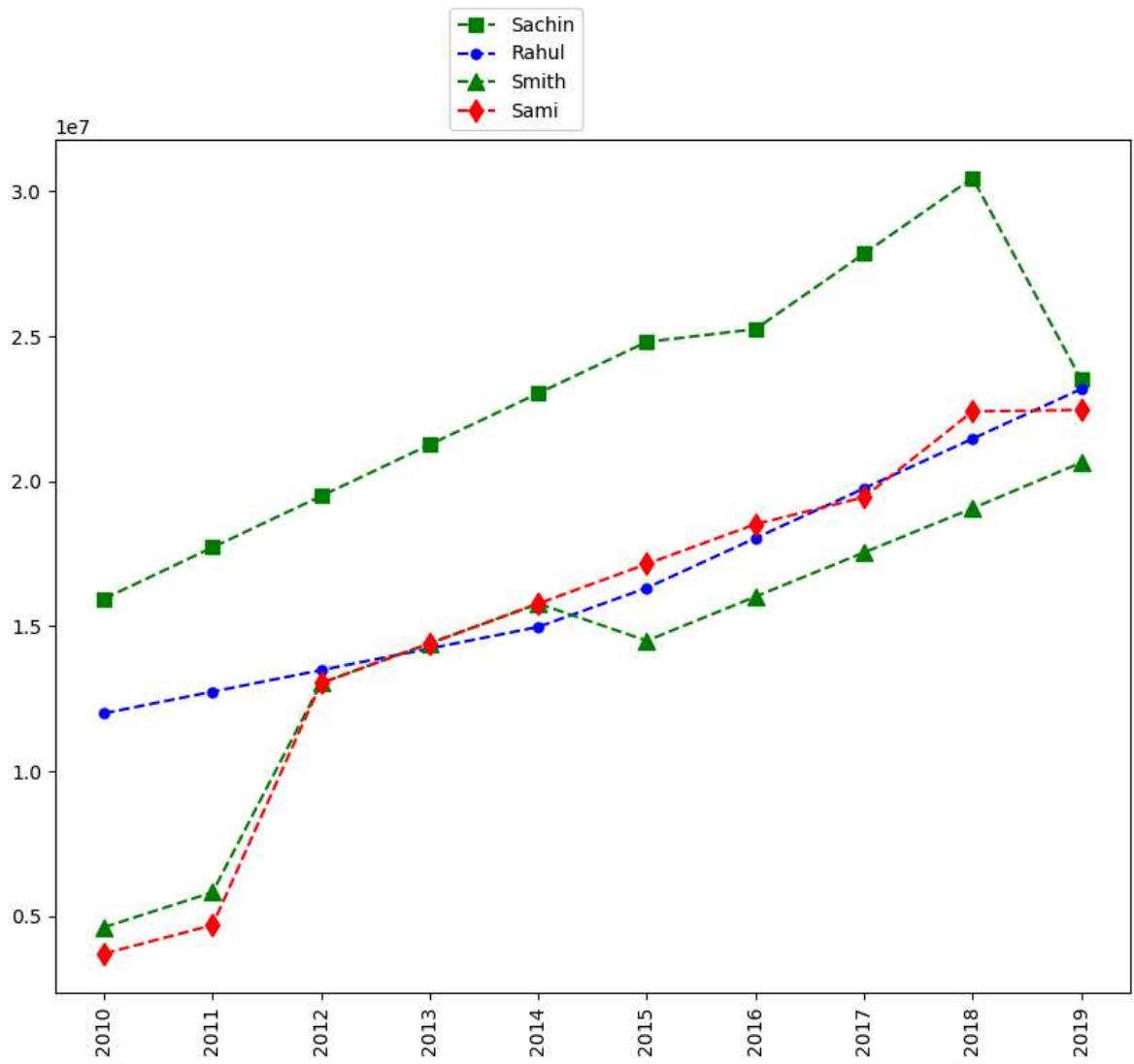
```
In [80]: plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Players[0])
plt.plot(Salary[1], c='Blue', ls = '--', marker = 'o', ms = 5, label = Players[1]
plt.plot(Salary[2], c='Green', ls = '--', marker = '^', ms = 8, label = Players[2]
plt.plot(Salary[3], c='Red', ls = '--', 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 [81]: 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 [85]: 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[85], line 12
    9 plt.plot(Salary[8], c='Red', ls = '--', marker = 's', ms = 7, label = Players[8])
   10 plt.plot(Salary[9], c='Red', ls = '--', marker = 'o', ms = 7, label = Players[9])
--> 12 plt.legend(loc = 'lower 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: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 ~\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_

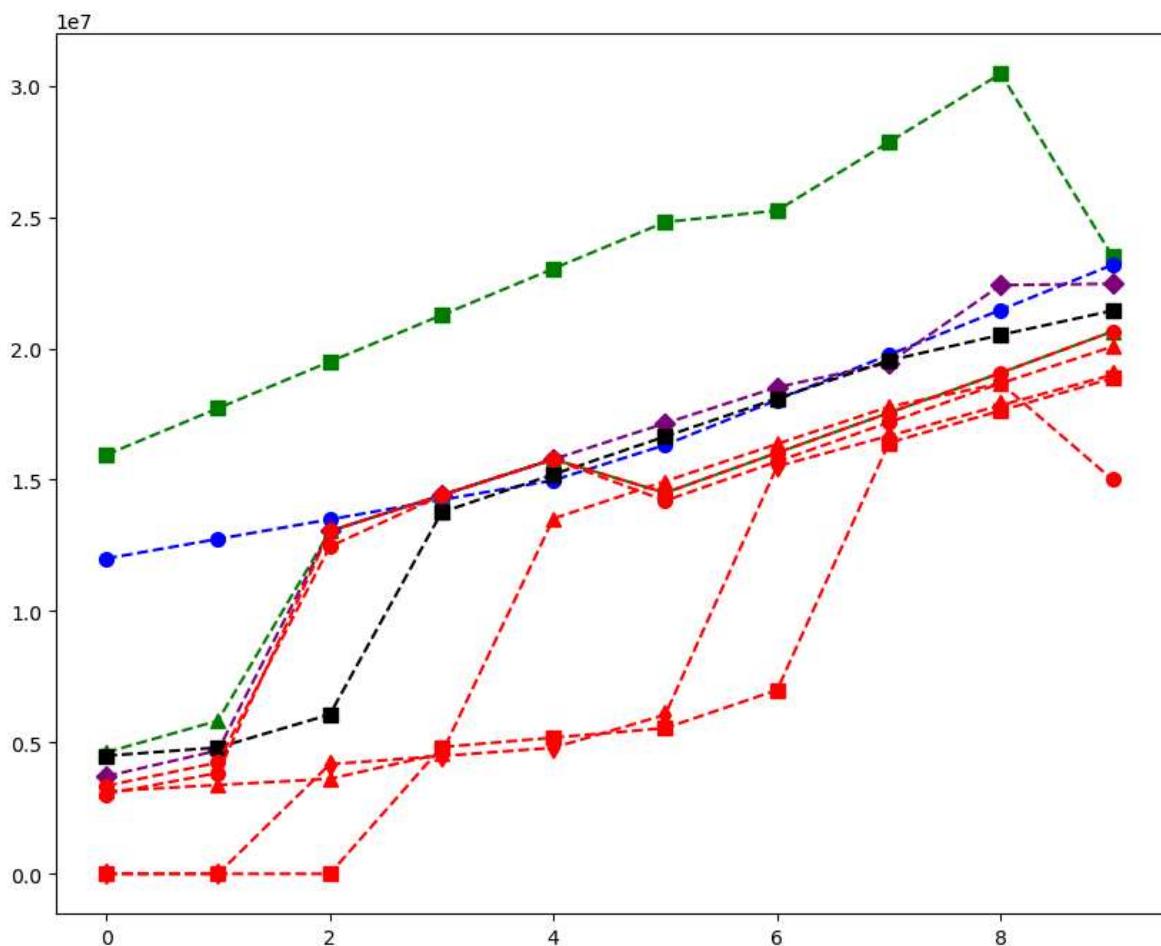
File ~\anaconda3\Lib\site-packages\matplotlib\legend.py:566, in Legend.__init__(self, parent, handles, labels, loc, numpoints, markerscale, markerfirst, reverse, scatterpoints, scatteryoffsets, prop, fontsize, labelcolor, borderpad, labelspacing, handlelength, handleheight, handletextpad, borderaxespad, columnspacing, ncols, 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 ~\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 ~\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'

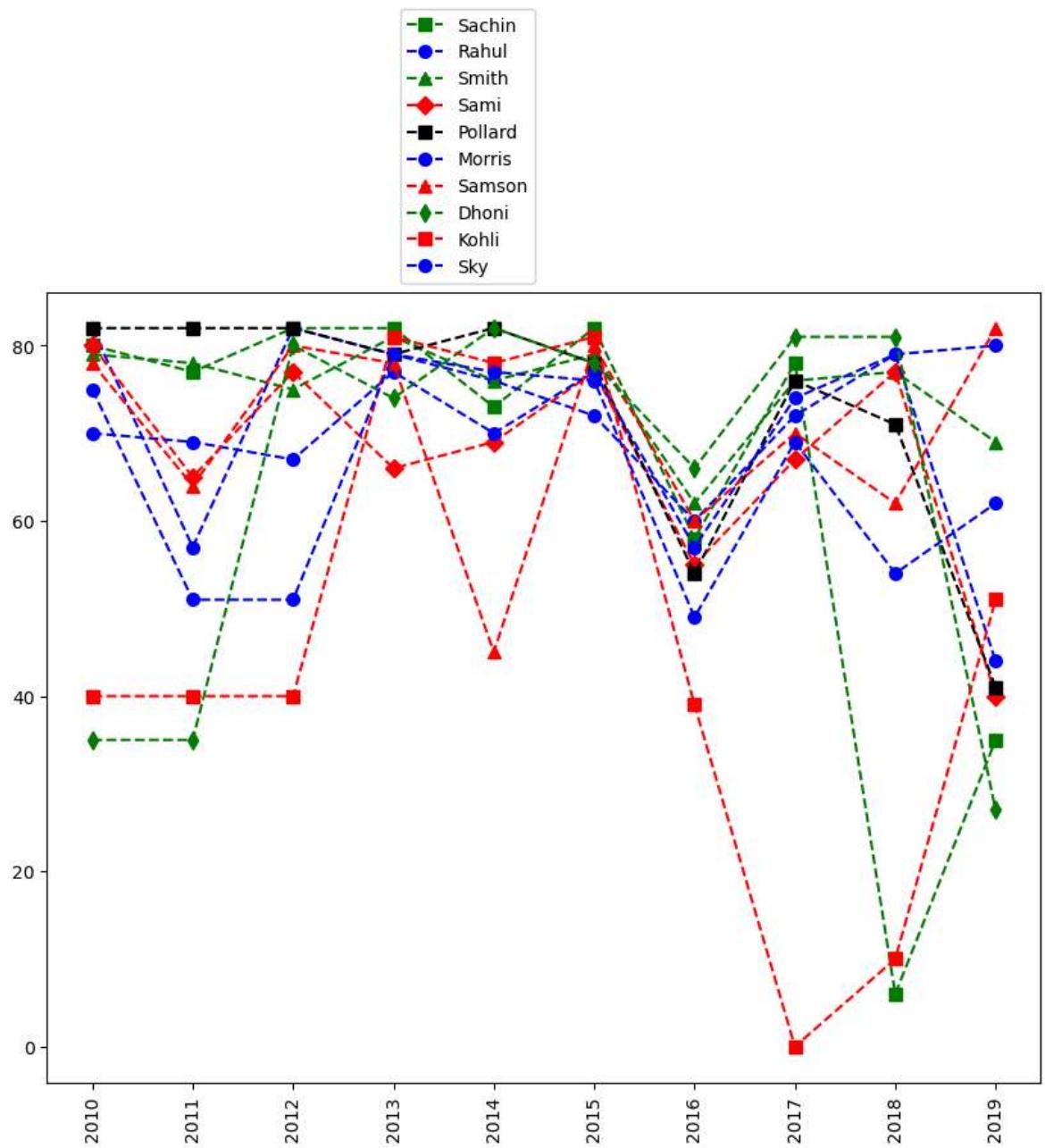


In [86]: # we can visualize the how many games played by a player

```
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 [ ]:

In [ ]: