IPL Data Analysis

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

#Seasons
Seasons = ["2015","2016","2017","2018","2019","2020","2021","2022","2023","2024"
Sdict = {"2015":0,"2016":1,"2017":2,"2018":3,"2019":4,"2020":5,"2021":6,"2022":7

#Players
Players = ["Sachin","Rahul","Smith","Sami","Pollard","Morris","Samson","Dhoni","
Pdict = {"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, Samsc

#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 [2]:
```
Salary   # matrix format
```

Out[2]:
```
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 [3]:
```
# Building your first matrix
Games
```

Out[3]:
```
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 [4]:
```
Points
```

Out[4]:
```
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 [5]:
```
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 [6]:
```
np.reshape(mydata,(4,5))  #5 rows & 4 columns
```

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

In [7]:
```
mydata
```

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

In [8]:
```
# np.reshape(mydata,(5,4), order = 'c')  #'c' means to read / write the element

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]:
```
# If I want to get only no.3
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]:
```
MATR2[0:2]
```

Out[19]:
```
array([[ 0,  5, 10, 15],
       [ 1,  6, 11, 16]])
```

In [20]:
```
MATR2
```

Out[20]:
```
array([[ 0,  5, 10, 15],
       [ 1,  6, 11, 16],
       [ 2,  7, 12, 17],
       [ 3,  8, 13, 18],
       [ 4,  9, 14, 19]])
```

In [21]:
```
MATR2[1:2]
```

Out[21]:
```
array([[ 1,  6, 11, 16]])
```

In [22]:
```
MATR2[1,2]
```

Out[22]:
```
11
```

In [23]:
```
MATR2
```

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

In [24]:
```
MATR2[-2,-1]
```

Out[24]:
```
18
```

In [25]:
```
MATR2[-3,-3]
```

Out[25]:
```
7
```

In [26]:
```
MATR2
```

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

In [27]:
```
MATR2[0:2]
```

Out[27]:
```
array([[ 0,  5, 10, 15],
       [ 1,  6, 11, 16]])
```

In [28]:
```
mydata
```

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

In [29]:
```
MATR3 = np.reshape(mydata, (5,4), order='A')
MATR3
```

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

In [30]:
```
MATR2   ##F shaped
```

Out[30]:
```
array([[ 0,  5, 10, 15],
       [ 1,  6, 11, 16],
       [ 2,  7, 12, 17],
       [ 3,  8, 13, 18],
       [ 4,  9, 14, 19]])
```

In [31]:
```
MATR1  #C Shaped
```

Out[31]:
```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11],
       [12, 13, 14, 15],
       [16, 17, 18, 19]])
```

In [32]:
```
a1 = ['welcome', 'to','datascience']
a2 = ['required', 'hard', 'work']
a3= [1,2,3]
```

In [33]:
```
[a1,a2,a3]  #list same datatype
```

Out[33]:
```
[['welcome', 'to', 'datascience'], ['required', 'hard', 'work'], [1, 2, 3]]
```

In [34]:
```
np.array([a1,a2,a3])  #u11 = unicode 11 character : 3*3 matrix
```

Out[34]:
```
array([['welcome', 'to', 'datascience'],
       ['required', 'hard', 'work'],
       ['1', '2', '3']], dtype='<U11')
```

In [35]:
```
Games
```

Out[35]:
```
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 [36]:
```
Games[0]
```

Out[36]:
```
array([80, 77, 82, 82, 73, 82, 58, 78,  6, 35])
```

In [37]:
```
Games[5]
```

Out[37]:
```
array([70, 69, 67, 77, 70, 77, 57, 74, 79, 44])
```

In [38]:
```
Games[0:5]
```

Out[38]:
```
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 [39]:
```
Games[0,5]
```

Out[39]:
```
82
```

In [40]:
```
Games[0,2]
```

Out[40]:
```
82
```

In [41]:
```
Games[0,6]
```

Out[41]:
```
58
```

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[0:2]
```

Out[43]:
```
array([[80, 77, 82, 82, 73, 82, 58, 78,  6, 35],
       [82, 57, 82, 79, 76, 72, 60, 72, 79, 80]])
```

In [44]:
```
Games[1:2]
```

Out[44]:
```
array([[82, 57, 82, 79, 76, 72, 60, 72, 79, 80]])
```

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

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

In [46]:
```
Games[2,8]
```

Out[46]:
```
77
```

In [47]:
```
Games[-3:-1]
```

Out[47]:
```
array([[35, 35, 80, 74, 82, 78, 66, 81, 81, 27],
       [40, 40, 40, 81, 78, 81, 39,  0, 10, 51]])
```

In [48]:
```
Games[-3,-1]
```

Out[48]:
```
27
```

In [49]:
```
Points
```

Out[49]:
```
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 [50]:
```
Points[0]
```

Out[50]:
```
array([2832, 2430, 2323, 2201, 1970, 2078, 1616, 2133,   83,  782])
```

In [51]:
```
Points[6,1]
```

Out[51]:
```
1104
```

In [52]:
```
Points[3:6]
```

Out[52]:
```
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 [53]:
```
Points[-6,-1]
```

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

In [54]:
```
Games
```

Out[54]:
```
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 [55]:
```
Pdict
```

Out[55]:
```
{'Sachin': 0,
 'Rahul': 1,
 'Smith': 2,
 'Sami': 3,
 'Pollard': 4,
 'Morris': 5,
 'Samson': 6,
 'Dhoni': 7,
 'Kohli': 8,
 'Sky': 9}
```

In [56]:
```
Pdict['Sachin']
```

Out[56]:
```
0
```

In [57]:
```
Games[0]
```

Out[57]:
```
array([80, 77, 82, 82, 73, 82, 58, 78,  6, 35])
```

In [58]:
```
Games
```

Out[58]:
```
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 [59]:
```
Pdict['Rahul']
```

Out[59]:
```
1
```

In [60]:
```
Games[1]
```

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

In [61]:
```
Games[Pdict['Rahul']]
```

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

In [62]:
```
Points
```

Out[62]:
```
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 [63]:
```
Salary
```

Out[63]:
```
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 [64]:
```
Salary[2,4]
```

Out[64]:
```
15779912
```

In [65]:
```
Salary
```

Out[65]:
```
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 [66]:
```
Salary[Pdict['Sky']][Sdict['2019']]
```

Out[66]:
```
15779912
```

In [67]:
```
Salary
```

Out[67]:
```
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 [68]:
```
Games
```

Out[68]:
```
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 [69]:
```
Salary / Games
```

```
C:\Users\Prachi\AppData\Local\Temp\ipykernel_15004\1572766764.py:1: RuntimeWarnin
g: divide by zero encountered in divide
  Salary / Games
```

Out[69]:
```
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 [70]:
```
np.round(Salary//Games)
```

```
C:\Users\Prachi\AppData\Local\Temp\ipykernel_15004\3663165759.py:1: RuntimeWarnin
g: divide by zero encountered in floor_divide
  np.round(Salary//Games)
```

Out[70]:
```
array([[ 199335,  230113,  237690,  259298,  315539,  302515,  435249,
          357040, 5075634,  671428],
       [ 146341,  223582,  164492,  180159,  197062,  226729,  300642,
          274342,  271730,  289759],
       [  58503,   74719,  173883,  177908,  207630,  183544,  258427,
          230855,  247629,  299194],
       [  46420,   72216,  169366,  218342,  228694,  222717,  336701,
          290298,  291006,  561450],
       [  54794,   58618,   73917,  174151,  185397,  213425,  335032,
          257057,  288918,  522835],
       [  47828,   61380,  185895,  187150,  225427,  188311,  281096,
          237094,  241360,  469190],
       [  40310,   52815,   45199,   58643,  300455,  186751,  272663,
          253992,  301103,  244738],
       [      0,       0,   52140,   60595,   58498,   77611,  234948,
          205797,  220155,  703541],
       [      0,       0,       0,   59540,   66467,   68471,  179325,
               0, 1763268,  369860],
       [  40425,   75322,  255710,  182412,  204933,  186842,  320224,
          249014,  345796,  241935]])
```

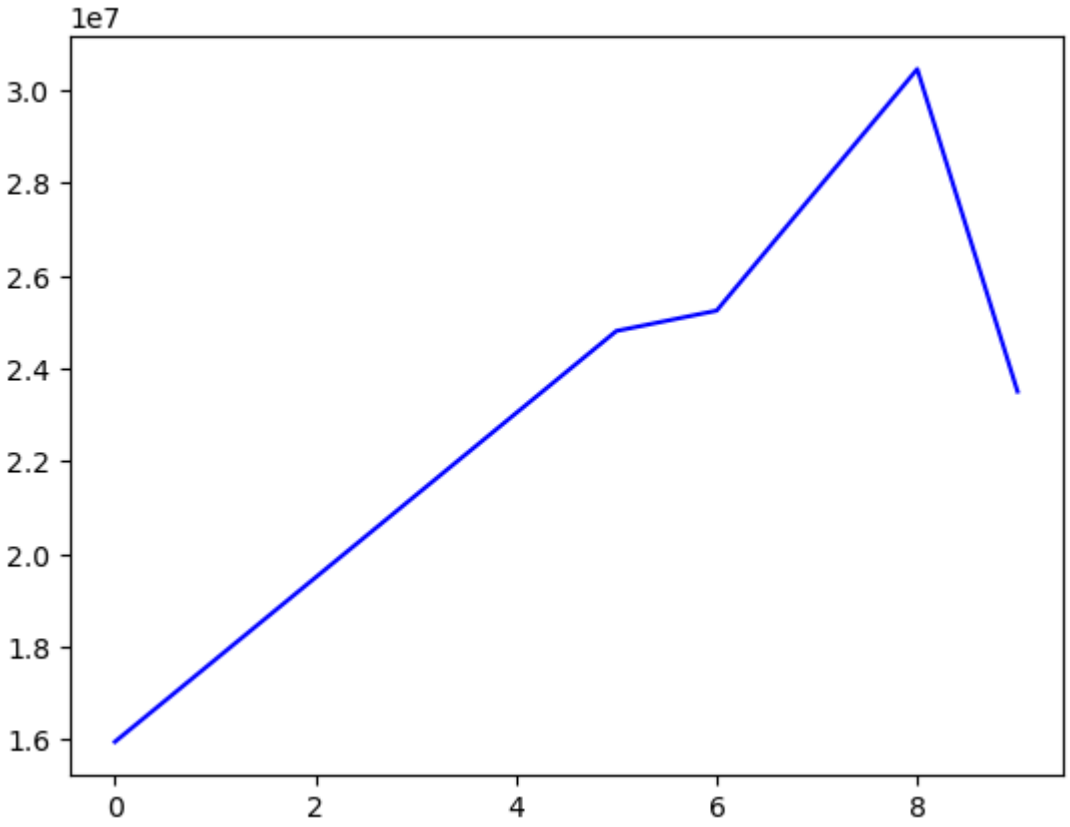
In [71]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [72]:
```
import matplotlib.pyplot as plt
import numpy as np
```

In [73]:
```
Salary[0]
```

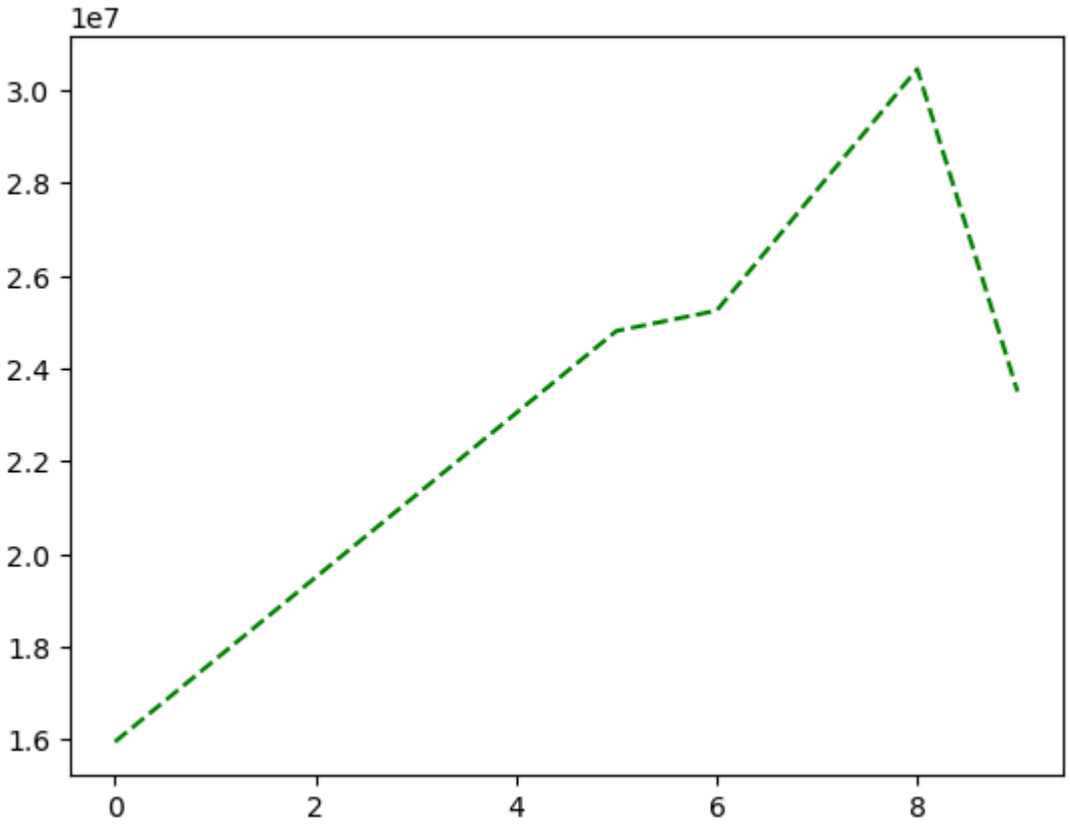
Out[73]:
```
array([15946875, 17718750, 19490625, 21262500, 23034375, 24806250,
       25244493, 27849149, 30453805, 23500000])
```

In [74]:
```
plt.plot(Salary[0], color='Blue', )
plt.show()
```



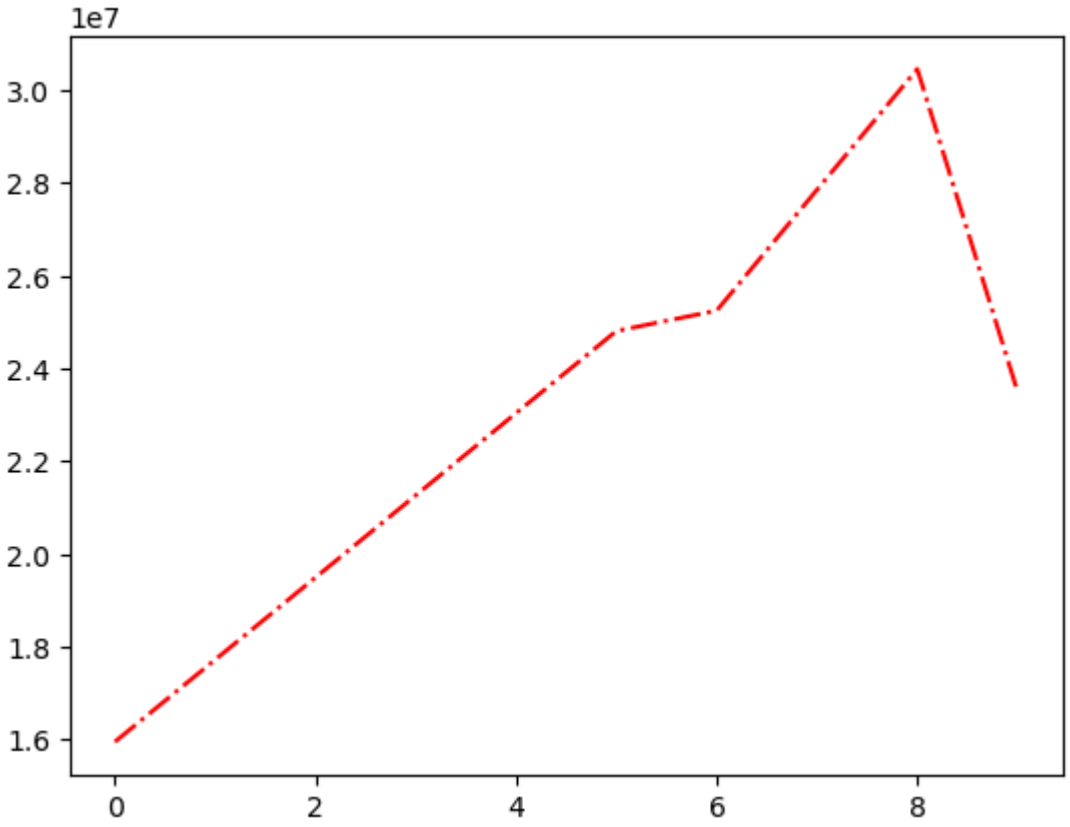
In [75]:
```
plt.plot(Salary[0], ls='--', color='Green' )
```

Out[75]:
```
[<matplotlib.lines.Line2D at 0x220d5847590>]
```



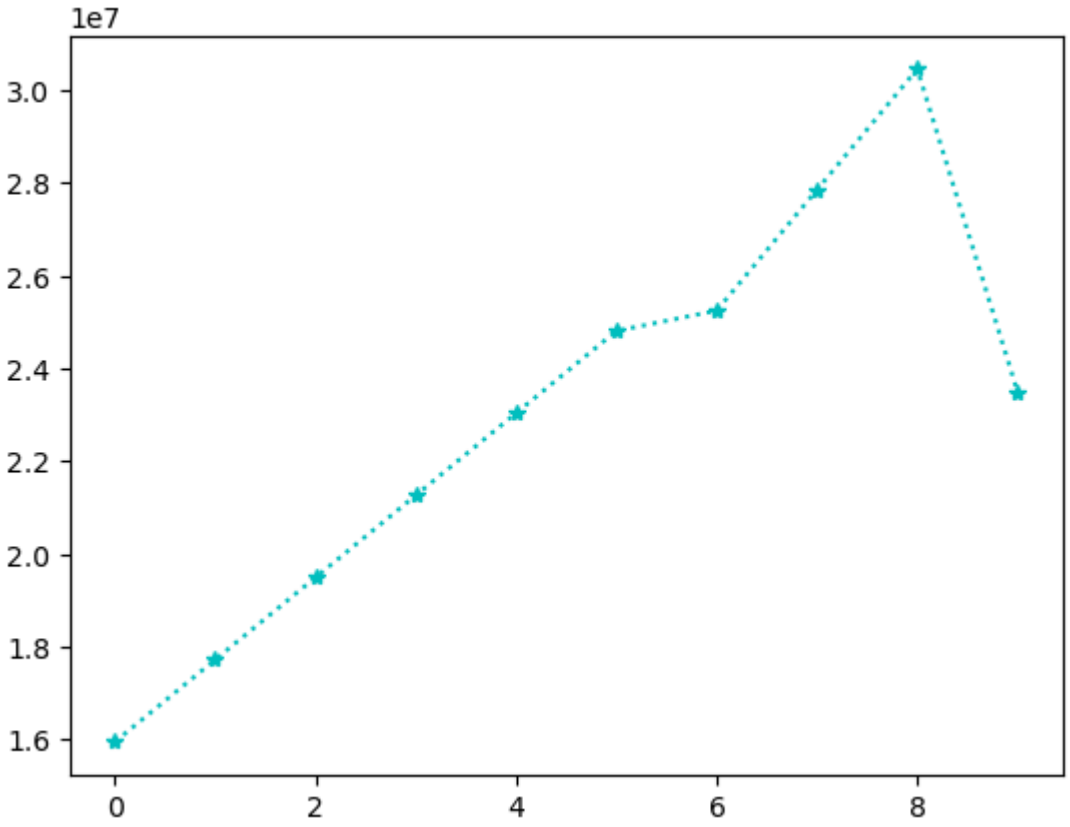
In [76]:
```
plt.plot(Salary[0], ls='-.' ,color ='r')
```

Out[76]:
```
[<matplotlib.lines.Line2D at 0x220d5a7d990>]
```



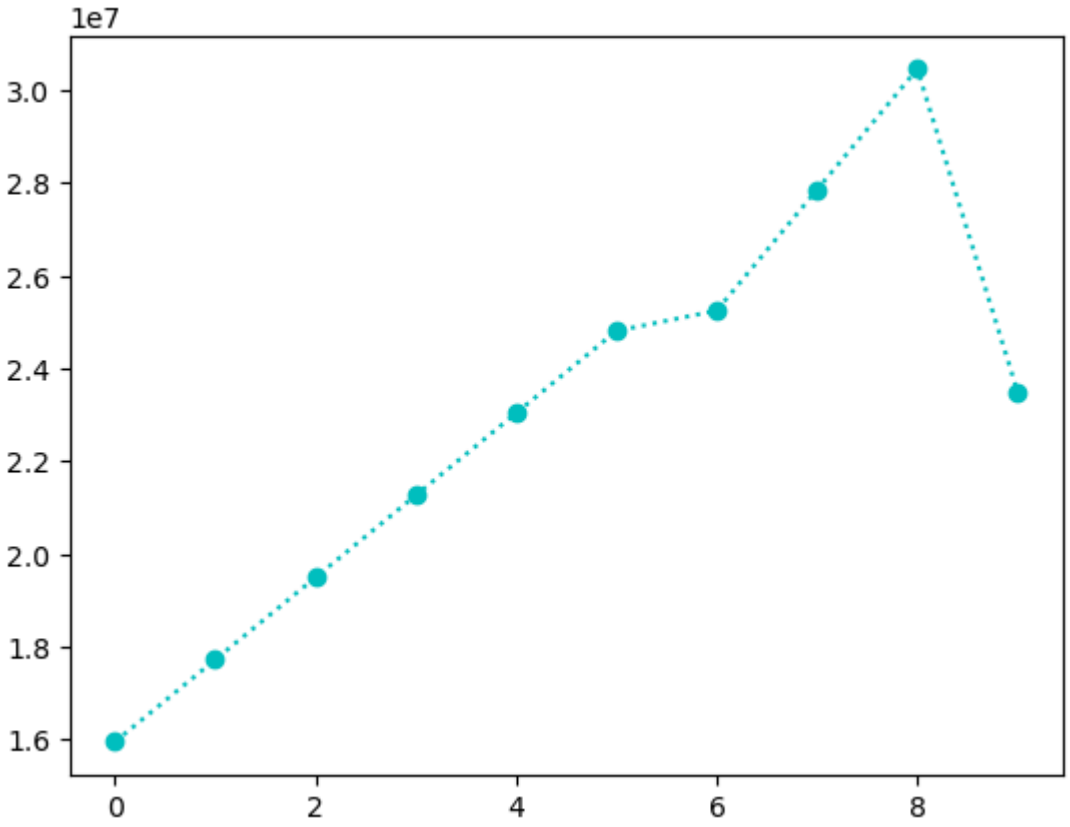
In [77]:
```
plt.plot(Salary[0], ls=':', color='c', marker='*' )
```

Out[77]:
```
[<matplotlib.lines.Line2D at 0x220d5af8e10>]
```



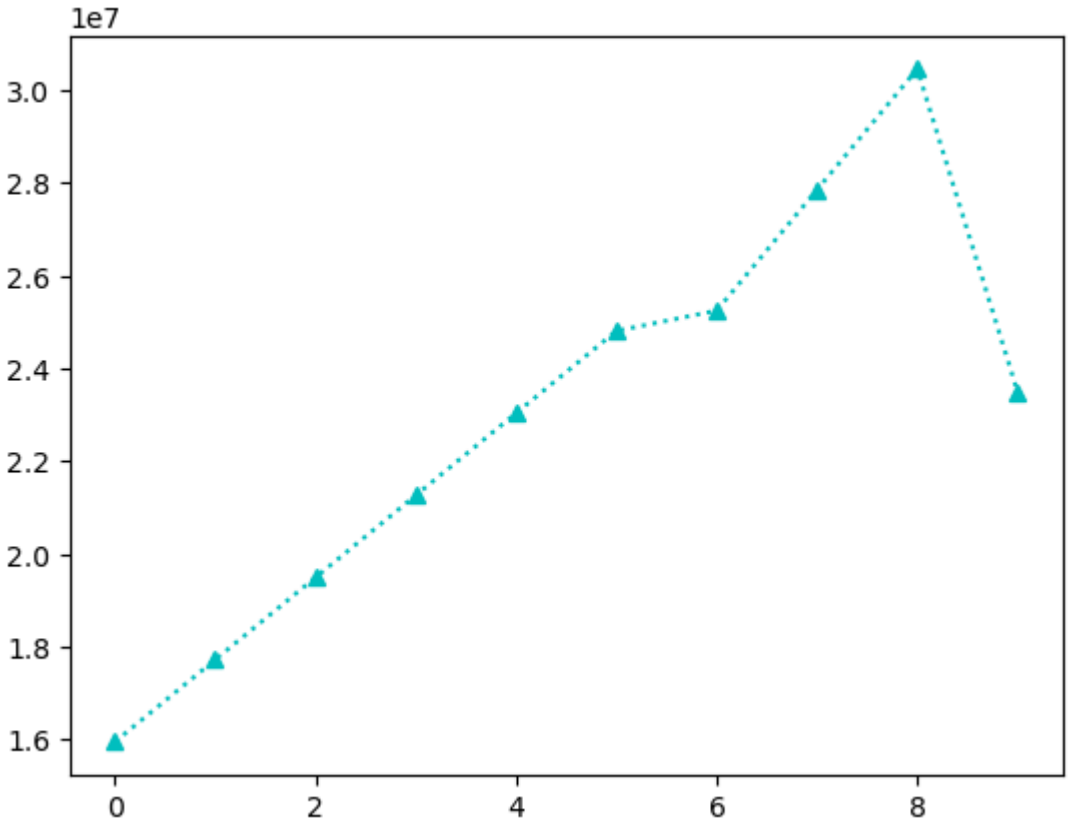
In [78]:
```
plt.plot(Salary[0], ls=':', color='c', marker = 'o' )
```

Out[78]:
```
[<matplotlib.lines.Line2D at 0x220d5b30810>]
```



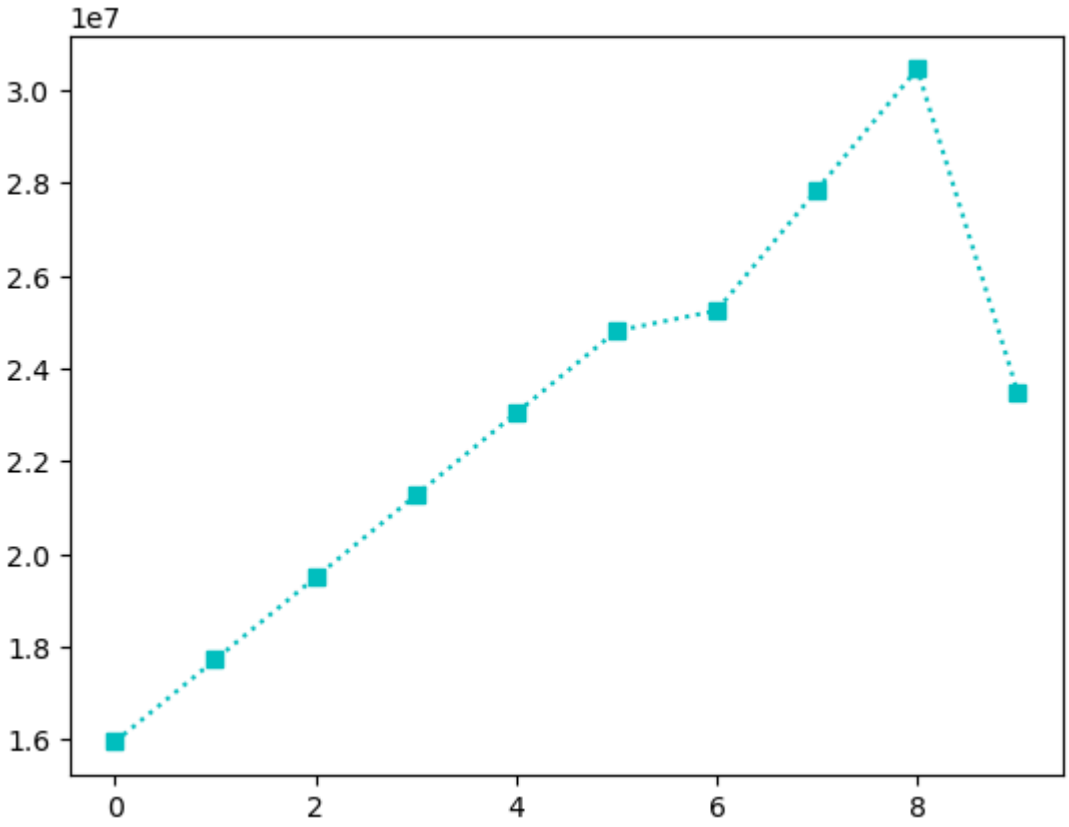
In [79]:
```
plt.plot(Salary[0], ls=':', color='c' , marker='^')
```

Out[79]:
```
[<matplotlib.lines.Line2D at 0x220d5bdd190>]
```



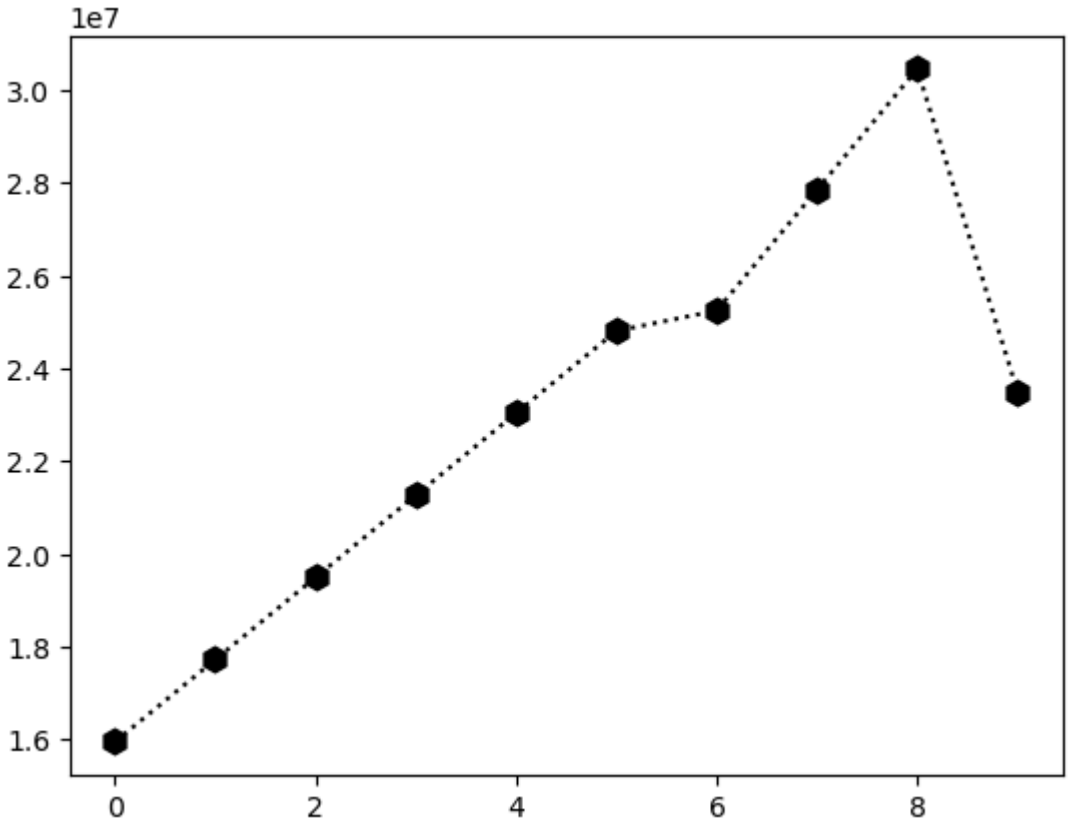
In [80]:
```
plt.plot(Salary[0], ls=':', color='c' , marker='s')
```

Out[80]:
```
[<matplotlib.lines.Line2D at 0x220d5c5f590>]
```



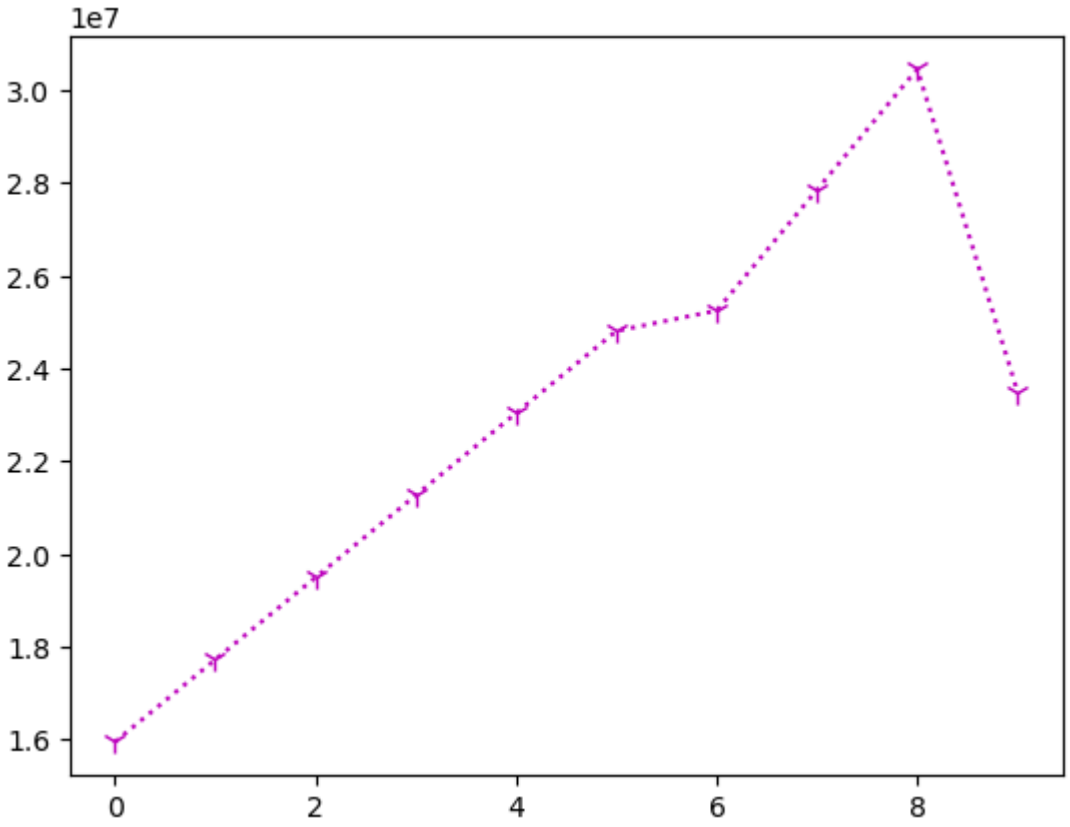
In [81]:
```
plt.plot(Salary[0], ls=':', color='Black', marker='h' , markersize = 9)
```

Out[81]:
```
[<matplotlib.lines.Line2D at 0x220d5965050>]
```



In [82]:
```
plt.plot(Salary[0], ls=':', color='m', marker='1' , markersize = 9)
```

Out[82]:
```
[<matplotlib.lines.Line2D at 0x220d5a1c710>]
```



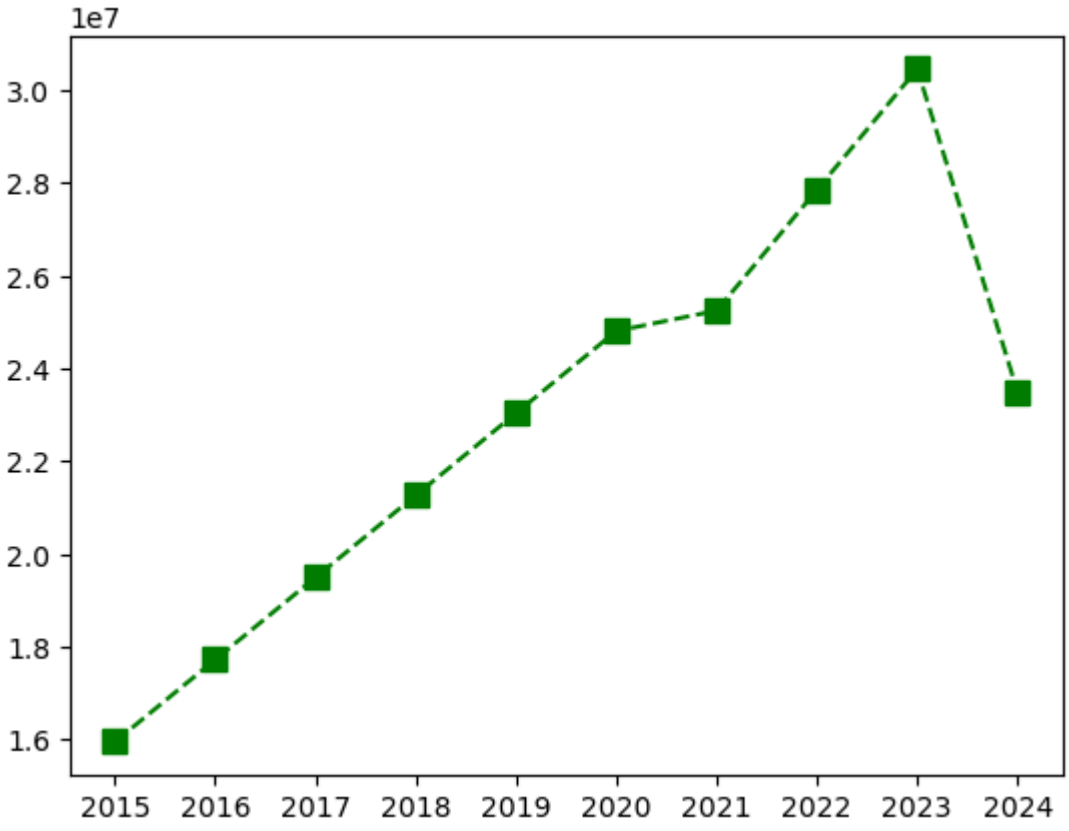
In [83]:
```
Sdict
```

Out[83]:
```
{'2015': 0,
 '2016': 1,
 '2017': 2,
 '2018': 3,
 '2019': 4,
 '2020': 5,
 '2021': 6,
 '2022': 7,
 '2023': 8,
 '2024': 9}
```

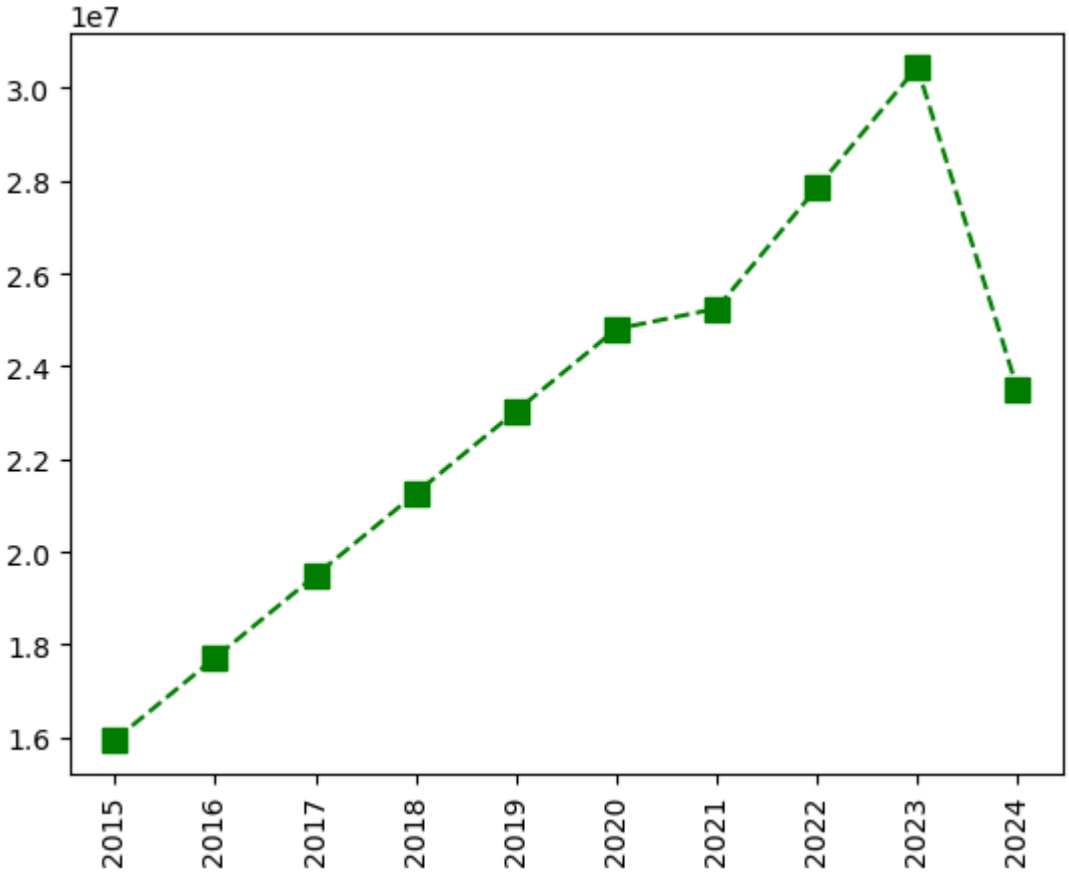
In [84]:
```
Pdict
```

Out[84]:
```
{'Sachin': 0,
 'Rahul': 1,
 'Smith': 2,
 'Sami': 3,
 'Pollard': 4,
 'Morris': 5,
 'Samson': 6,
 'Dhoni': 7,
 'Kohli': 8,
 'Sky': 9}
```

In [85]:
```
plt.plot(Salary[0], c='green', ls = '--', marker ='s', ms = 8)
plt.xticks(list(range(0,10)), Seasons)
plt.show()
```



In [86]:
```
plt.plot(Salary[0], c='green', ls = '--', marker ='s', ms = 8)
plt.xticks(list(range(0,10)), Seasons, rotation='vertical')
plt.show()
```

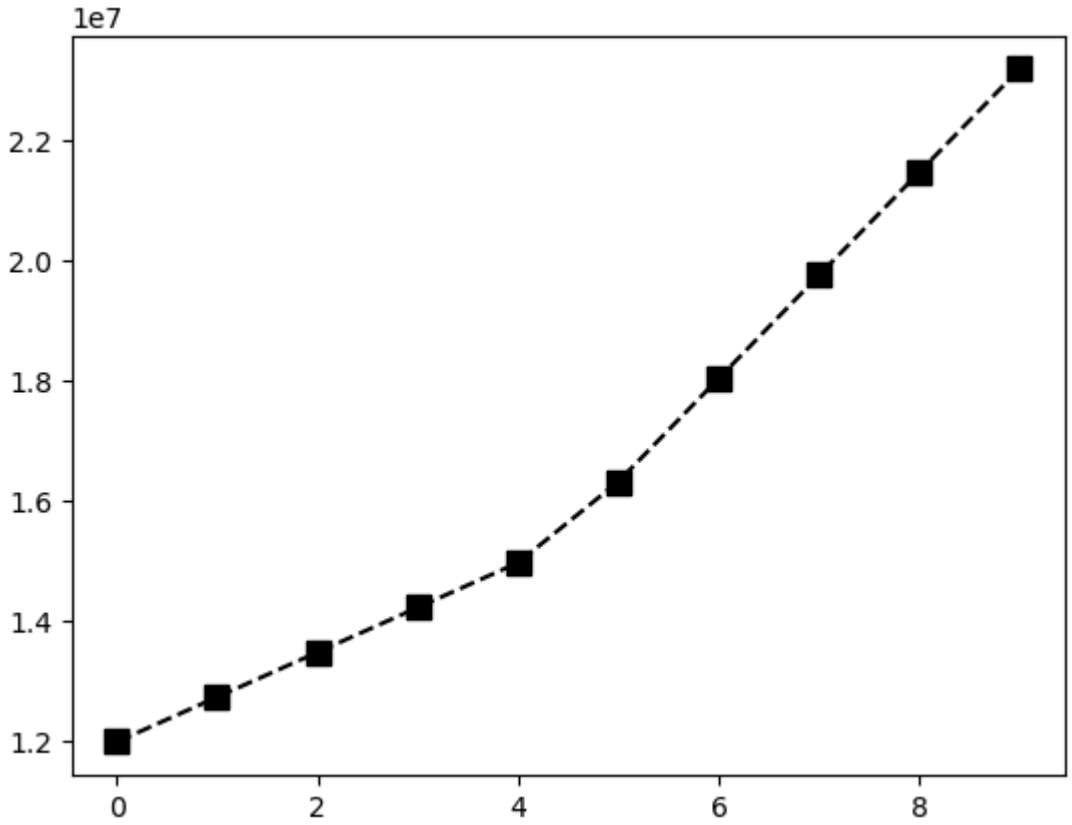


In [87]:
```
Salary[1]
```

Out[87]:
```
array([12000000, 12744189, 13488377, 14232567, 14976754, 16324500,
       18038573, 19752645, 21466718, 23180790])
```

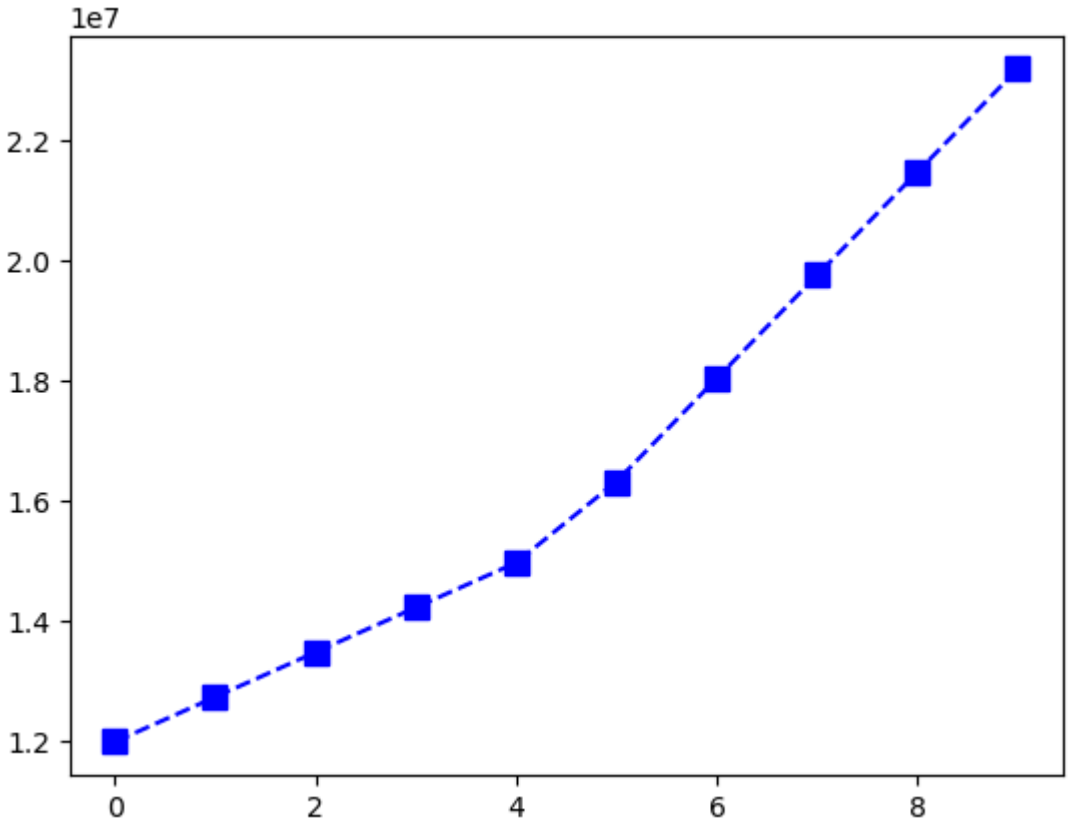
In [88]:
```
plt.plot(Salary[1], c='black', ls = '--', marker ='s', ms = 8)
```

Out[88]:
```
[<matplotlib.lines.Line2D at 0x220d6d7c9d0>]
```

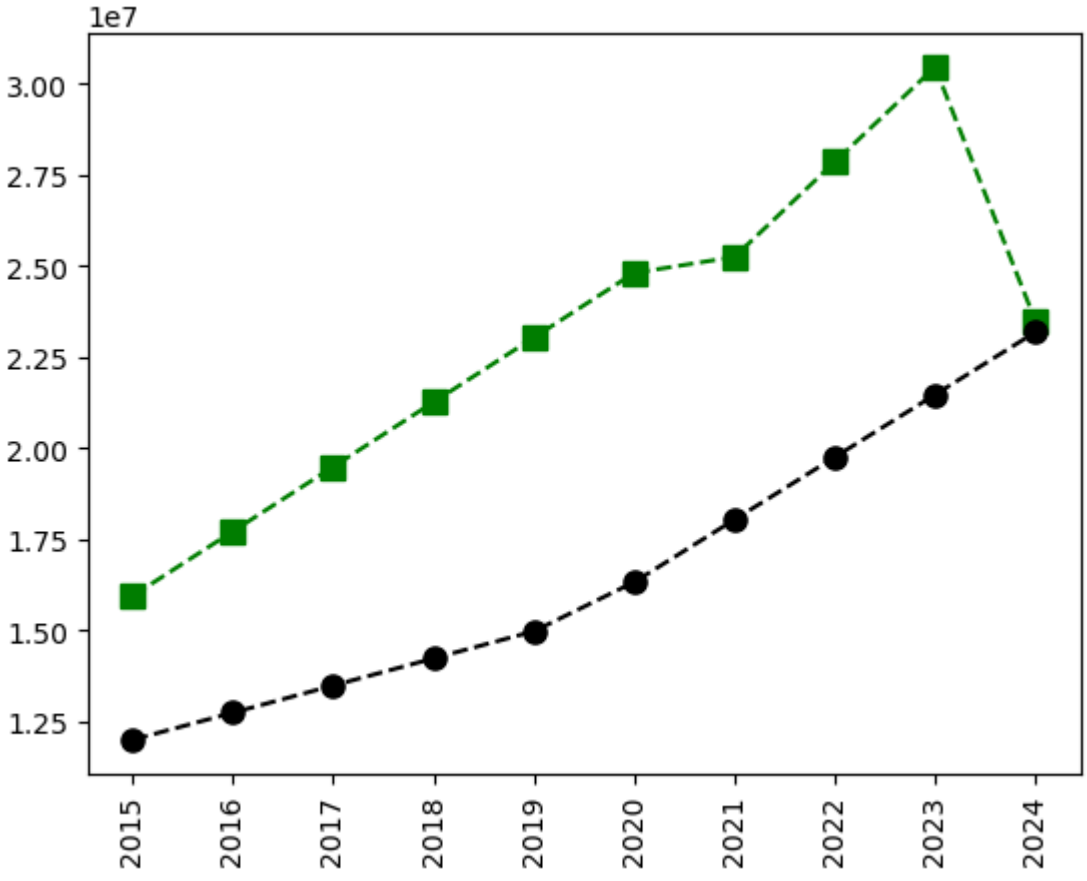


In [89]:
```
plt.plot(Salary[1], c='blue', ls = '--', marker ='s', ms = 8, label = Players[1]
```

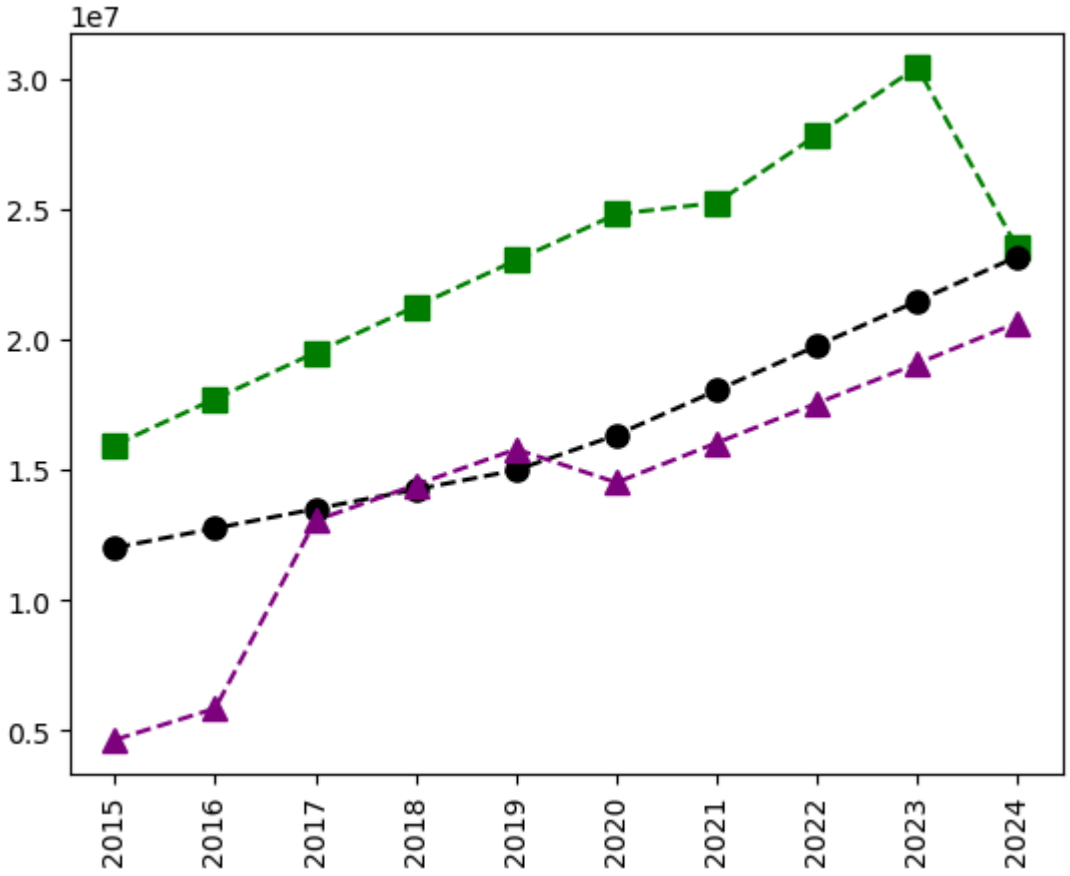
Out[89]:
```
[<matplotlib.lines.Line2D at 0x220d6d770d0>]
```



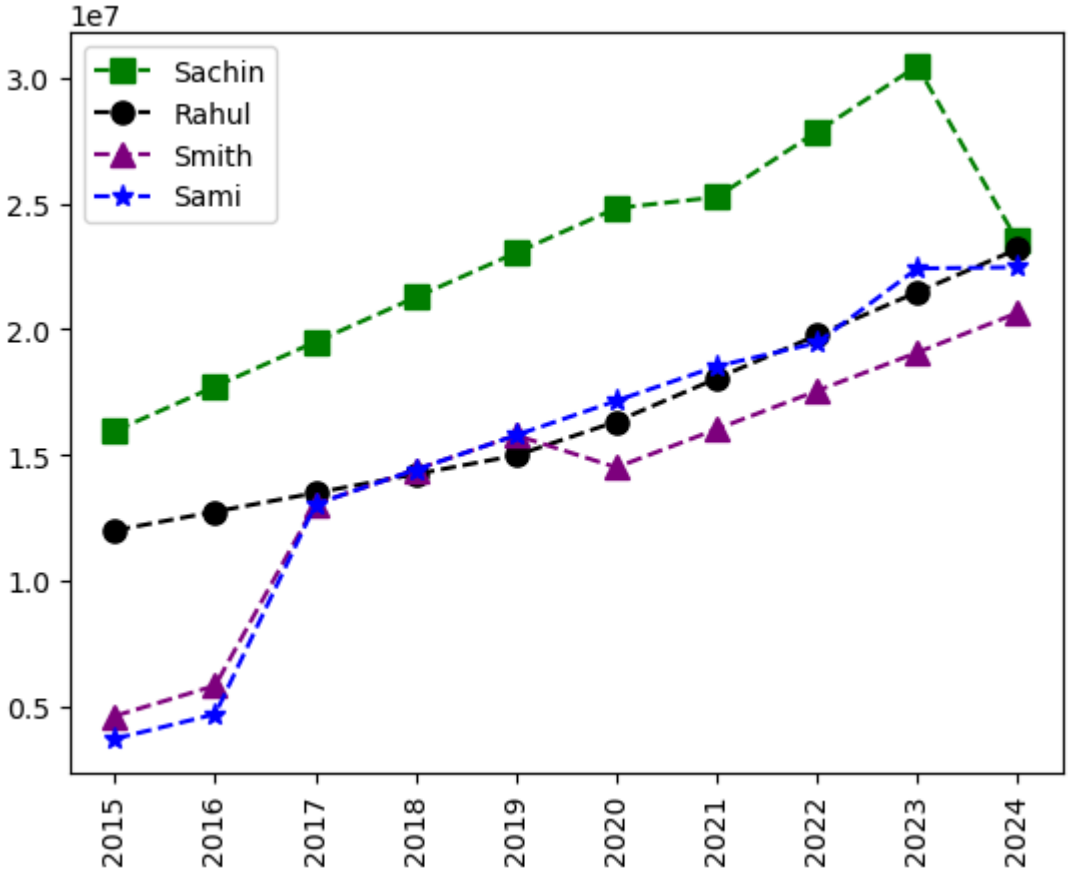
In [90]:
```
plt.plot(Salary[0], c='green', ls = '--', marker ='s', ms = 8, label = Players[0
plt.plot(Salary[1], c='black', ls = '--', marker ='o', ms = 8, label = Players[1

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



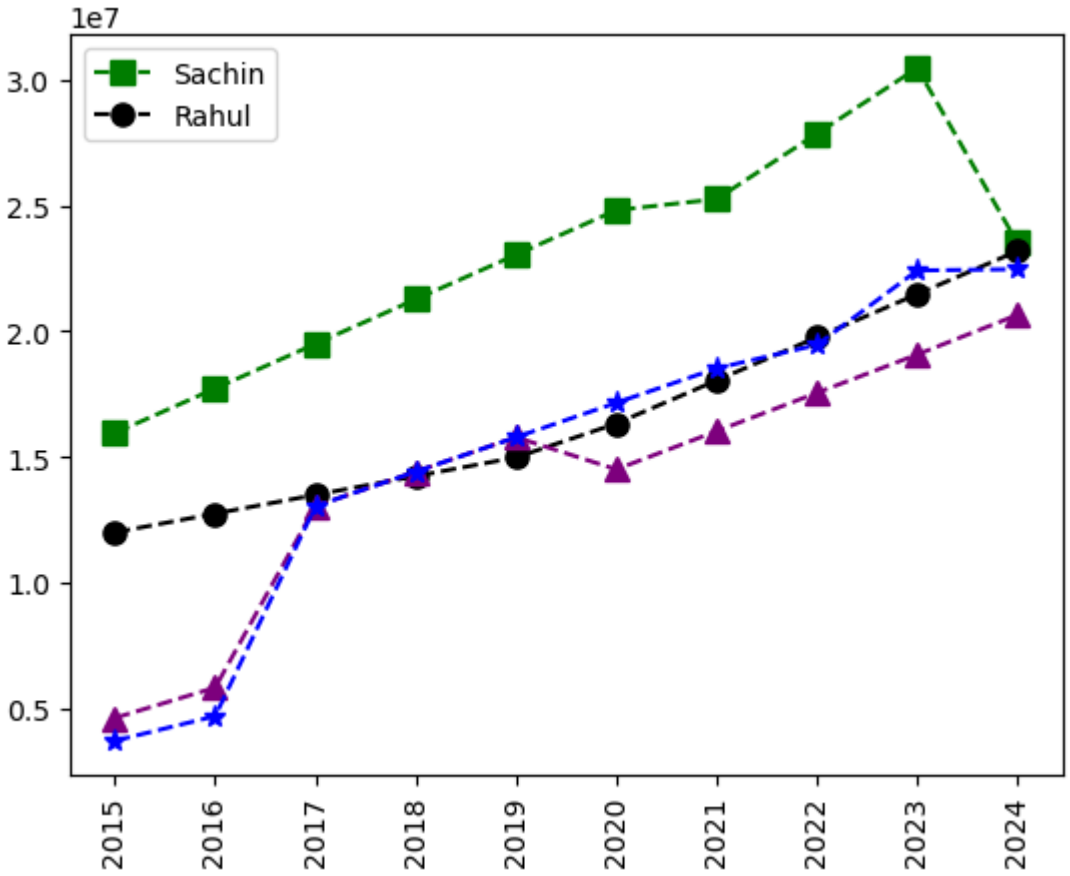
In [91]:
```
plt.plot(Salary[0], c='green', ls = '--', marker ='s', ms = 8, label = Players[0
plt.plot(Salary[1], c='black', ls = '--', marker ='o', ms = 8, label = Players[1
plt.plot(Salary[2], c='purple', ls = '--', marker ='^', ms = 8, label = Players[


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



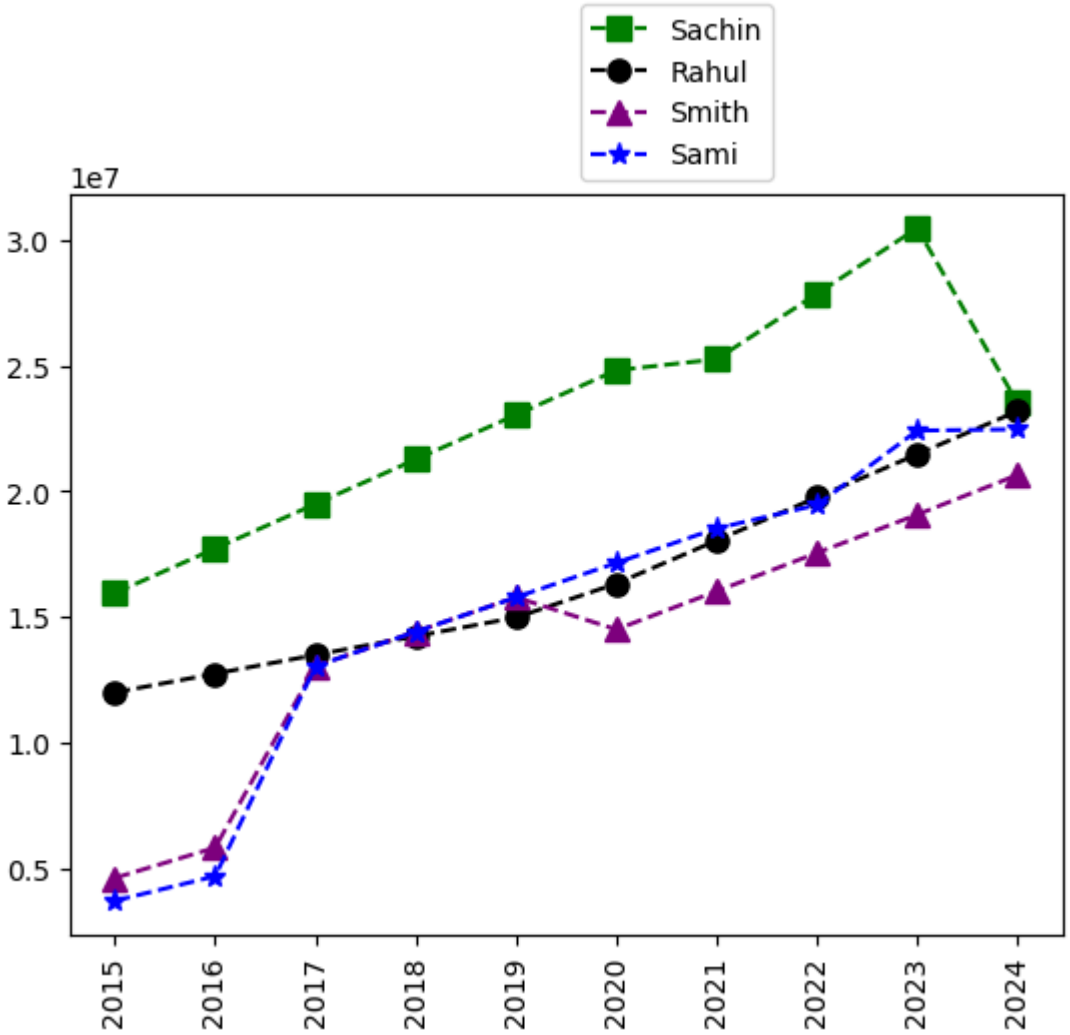
In [92]:
```
plt.plot(Salary[0], c='green', ls = '--', marker ='s', ms = 8, label = Players[0
plt.plot(Salary[1], c='black', ls = '--', marker ='o', ms = 8, label = Players[1
plt.plot(Salary[2], c='purple', ls = '--', marker ='^', ms = 8, label = Players[
plt.plot(Salary[3], c='blue', ls = '--', marker ='*', ms = 8, label = Players[3]


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



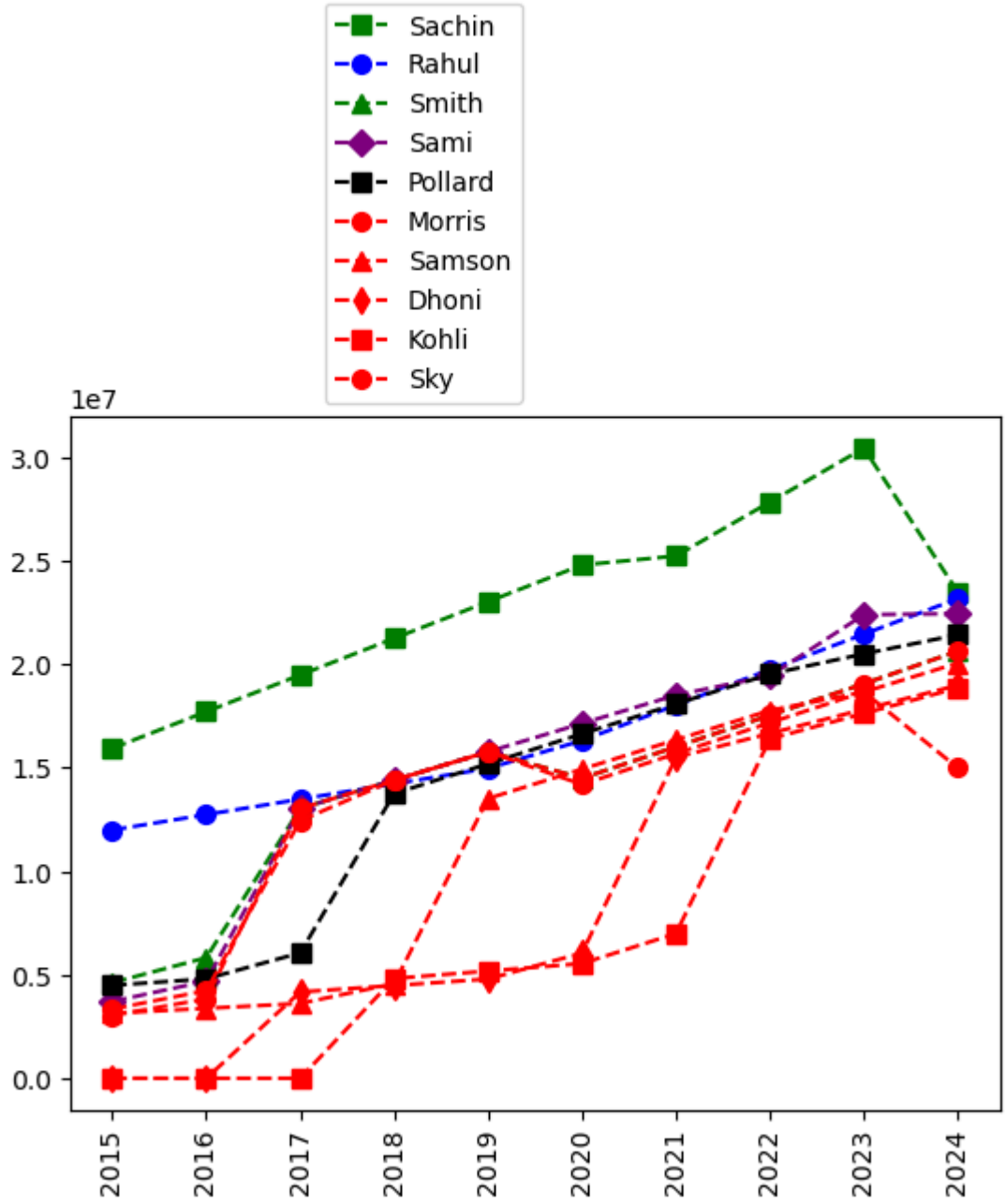
In [93]:
```
plt.plot(Salary[0], c='green', ls = '--', marker ='s', ms = 8, label = Players[0
plt.plot(Salary[1], c='black', ls = '--', marker ='o', ms = 8, label = Players[1
plt.plot(Salary[2], c='purple', ls = '--', marker ='^', ms = 8)
plt.plot(Salary[3], c='blue', ls = '--', marker ='*', ms = 8)


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



In [94]:
```
plt.plot(Salary[0], c='green', ls = '--', marker ='s', ms = 8, label = Players[0
plt.plot(Salary[1], c='black', ls = '--', marker ='o', ms = 8, label = Players[1
plt.plot(Salary[2], c='purple', ls = '--', marker ='^', ms = 8, label = Players[
plt.plot(Salary[3], c='blue', ls = '--', marker ='*', ms = 8, label = Players[3]


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



In [95]:
```
plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Players[
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[
plt.plot(Salary[3], c='Purple', ls = '--', marker = 'D', ms = 7, label = Players
plt.plot(Salary[4], c='Black', ls = '--', marker = 's', ms = 7, label = Players[
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()
```


Sachin
Rahul
Smith
Sami
Pollard
Morris
Samson
Dhoni
Kohli
Sky
1e7
3.0
2.5
2.0
1.5
1.0
0.5
0.0
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024