Project 2: IPL data Analysis Using Numpy + Matplotlib

IPL- Here we will use IPL dataset players data like salary and how many games they played overall and their complete progess of the players. the data is completly provided by the company so we analyse by using numpy and visulaize the data using matplotlib.

Data analysis- Data analysis is the process of examining data to reach conclusions about a topic or question.

Numpy - NumPy is an open-source Python library designed to handle large, multi-dimensional arrays and matrices of numerical data, as well as perform mathematical operations on these data structures.

Matplotlib- Python library for displaying data and creating static, animated, and interactive plots.

In [2]:
```
import numpy as np
```

In [3]:
```
#Seasons
Seasons = ["2010","2011","2012","2013","2014","2015","2016","2017","2018","2019"]

# we want to print the seasons with index so we are using dict datatype to specify
Sdict = {"2010":0,"2011":1,"2012":2,"2013":3,"2014":4,"2015":5,"2016":6,"2017":7,"2
```

In [4]:
```
Sdict
```

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

In [5]:
```
# Players
Players = ["Sachin","Rahul","Smith","Sami","Pollard","Morris","Samson","Dhoni","Koh

# we want to print the seasons with index so we are using dict datatype to specify
Pdict = {"Sachin":0,"Rahul":1,"Smith":2,"Sami":3,"Pollard":4,"Morris":5,"Samson":6,
```

In [6]:
```
Pdict
```

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

In [7]:
```
#Salaries

Sachin_Salary = [15946875,17718750,19490625,21262500,23034375,24806250,25244493,278
Rahul_Salary = [12000000,12744189,13488377,14232567,14976754,16324500,18038573,1975
Smith_Salary = [4621800,5828090,13041250,14410581,15779912,14500000,16022500,175450
Sami_Salary = [3713640,4694041,13041250,14410581,15779912,17149243,18518574,1945000
Pollard_Salary = [4493160,4806720,6061274,13758000,15202590,16647180,18091770,19536
Morris_Salary = [3348000,4235220,12455000,14410581,15779912,14500000,16022500,17545
Samson_Salary = [3144240,3380160,3615960,4574189,13520500,14940153,16359805,1777945
Dhoni_Salary = [0,0,4171200,4484040,4796880,6053663,15506632,16669630,17832627,1899
Kohli_Salary = [0,0,0,4822800,5184480,5546160,6993708,16402500,17632688,18862875]
Sky_Salary = [3031920,3841443,13041250,14410581,15779912,14200000,15691000,17182000

#Matrix

Salary = np.array([Sachin_Salary, Rahul_Salary, Smith_Salary, Sami_Salary, Pollard_
                   Sky_Salary])
```

In [8]:
```
Salary
```

Out[8]:
```
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 [9]:
```
#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
```

In [10]:
```
Games
```

Out[10]:
```
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 [11]:
```
#Points
Sachin_PTS = [2832,2430,2323,2201,1970,2078,1616,2133,83,782]
Rahul_PTS = [1653,1426,1779,1688,1619,1312,1129,1170,1245,1154]
Smith_PTS = [2478,2132,2250,2304,2258,2111,1683,2036,2089,1743]
Sami_PTS = [2122,1881,1978,1504,1943,1970,1245,1920,2112,966]
Pollard_PTS = [1292,1443,1695,1624,1503,1784,1113,1296,1297,646]
Morris_PTS = [1572,1561,1496,1746,1678,1438,1025,1232,1281,928]
Samson_PTS = [1258,1104,1684,1781,841,1268,1189,1186,1185,1564]
Dhoni_PTS = [903,903,1624,1871,2472,2161,1850,2280,2593,686]
Kohli_PTS = [597,597,597,1361,1619,2026,852,0,159,904]
Sky_PTS = [2040,1397,1254,2386,2045,1941,1082,1463,1028,1331]

#Matrix
Points = np.array([Sachin_PTS, Rahul_PTS, Smith_PTS, Sami_PTS, Pollard_PTS, Morris_
```

In [12]:
```
Points
```

Out[12]:
```
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]])
```

here in the dataset seasons and players are in dict format and slalary and games points this three in array format

silcing and indexing with the data

In [15]:
```
Games
```

Out[15]:
```
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 [16]:
```
Games[5] # displays fifth row in the array
```

Out[16]:
```
array([70, 69, 67, 77, 70, 77, 57, 74, 79, 44])
```

In [17]:
```
Games[0:4]  # displays from 0th row to 3rd row
```

Out[17]:
```
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]])
```

In [18]:
```
Games[0,5] # indexing in 0th row fifth element
```

Out[18]:
```
82
```

In [19]:
```
Games[-3,-5] # reveerse indexing
```

Out[19]:
```
78
```

In [20]:
```
Points
```

Out[20]:
```
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 [21]:
```
Points[1:6] # from 1st row to 5th row
```

Out[21]:
```
array([[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]])
```

In [22]:
```
Points[6,4]
```

Out[22]:
```
841
```

In [23]:
```
Points[-6:-1] #from -6 to -2
```

Out[23]:
```
array([[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]])
```

In [24]:
```
Points[5,7]
```

Out[24]:
```
1232
```

In [25]:
```
Points[:] # complete array will print
```

Out[25]:
```
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 [26]:
```
Pdict
```

Out[26]:
```
{'Sachin': 0,
 'Rahul': 1,
 'Smith': 2,
 'Sami': 3,
 'Pollard': 4,
 'Morris': 5,
 'Samson': 6,
 'Dhoni': 7,
 'Kohli': 8,
 'Sky': 9}
```

In [27]:
```
Pdict['Sachin'] # By giving the keys we can get the value
```

Out[27]:
```
0
```

In [28]:
```
Pdict['Sky']
```

Out[28]:
```
9
```

In [29]:
```
Pdict['Sami']
```

Out[29]:
```
3
```

In [30]:
```
Points
```

Out[30]:
```
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 [31]:
```
Games
```

Out[31]:
```
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 [32]:
```
Salary
```

Out[32]:
```
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]])
```

If we want to now the salary for each game we divide salary by games

In [34]:
```
Salary/Games  # we got the float value of salary
```

```
C:\Users\91630\AppData\Local\Temp\ipykernel_28468\1223354707.py:1: RuntimeWarning: d
ivide by zero encountered in divide
  Salary/Games  # we got the float value of salary
```

Out[34]:
```
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 [35]:
```
Salary//Games
```

```
C:\Users\91630\AppData\Local\Temp\ipykernel_28468\1634212085.py:1: RuntimeWarning: d
ivide by zero encountered in floor_divide
  Salary//Games
```

Out[35]:
```
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]])
```

To avoid the warning like the above code we import warnings

In [37]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [38]:
```
Salary//Games  # no warnings
```

Out[38]:
```
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]])
```

now we will import matplotlib

In [40]:
```
import matplotlib.pyplot as plt
```

In [41]:
```
%matplotlib inline
```

In [42]:
```
Salary # to display the salary matrix
```

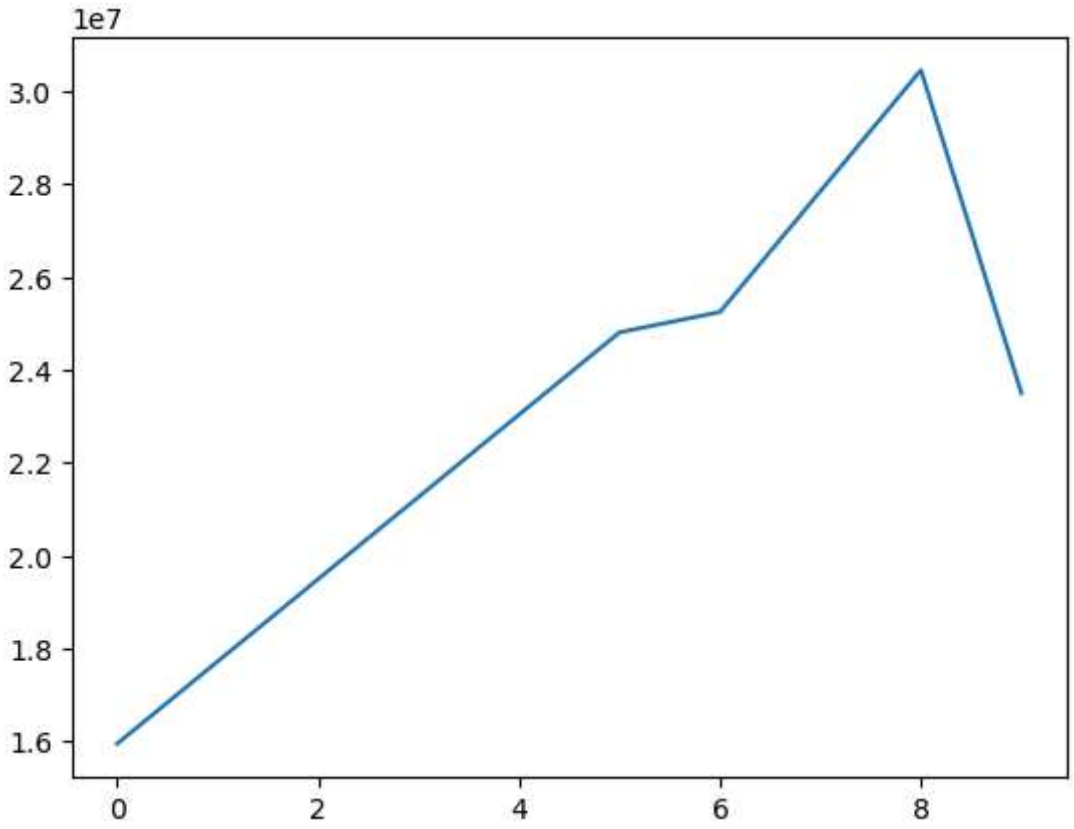
Out[42]:
```
array([[15946875, 17718750, 19490625, 21262500, 23034375, 24806250,
        25244493, 27849149, 30453805, 23500000],
       [12000000, 12744189, 13488377, 14232567, 14976754, 16324500,
        18038573, 19752645, 21466718, 23180790],
       [ 4621800,  5828090, 13041250, 14410581, 15779912, 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 [43]:
```
Pdict
```

Out[43]:
```
{'Sachin': 0,
 'Rahul': 1,
 'Smith': 2,
 'Sami': 3,
 'Pollard': 4,
 'Morris': 5,
 'Samson': 6,
 'Dhoni': 7,
 'Kohli': 8,
 'Sky': 9}
```

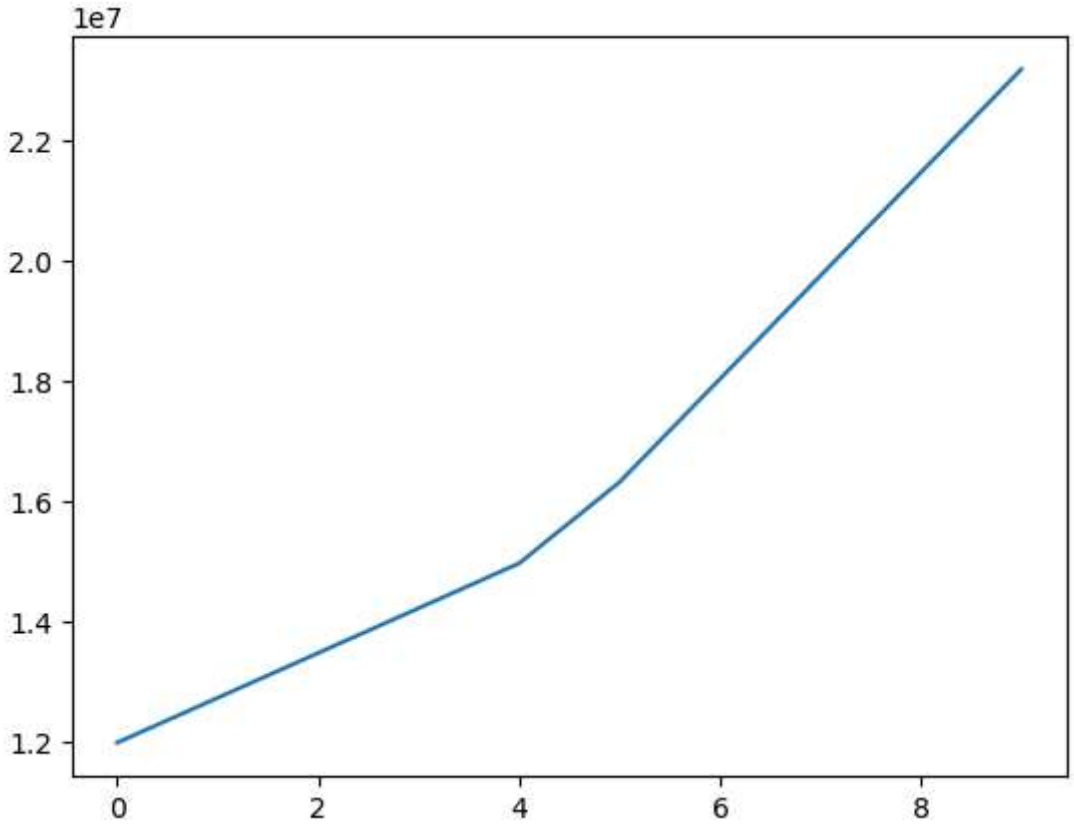
In [44]:
```
plt.plot(Salary[0]) # we got the salaray of 0th player i.e Sachin
```

Out[44]:
```
[<matplotlib.lines.Line2D at 0x1f7fa159550>]
```



In [45]:
```
plt.plot(Salary[1]) # we got the salaray of 1st player i.e rahul
```

Out[45]:
```
[<matplotlib.lines.Line2D at 0x1f7f8417d10>]
```



`we will upgrade the plotting by linestyle`

`Line Styles -ls`

character - description

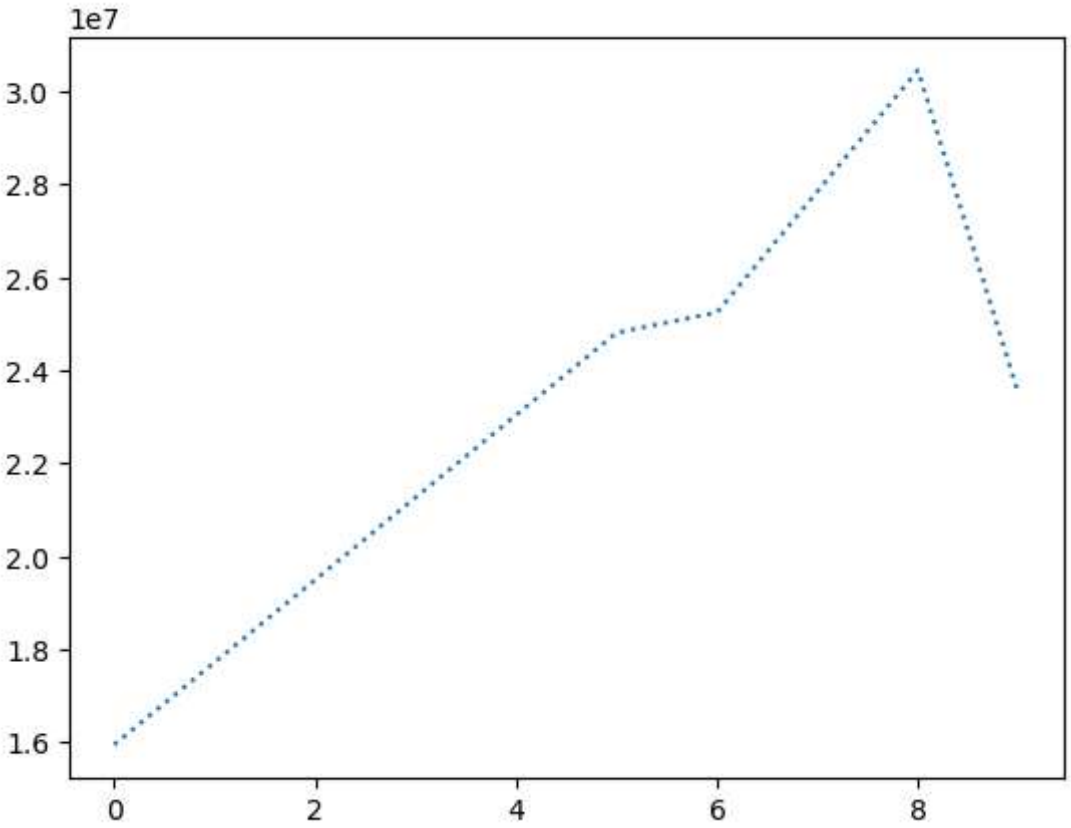
'-' - solid line style

'--' - dashed line style

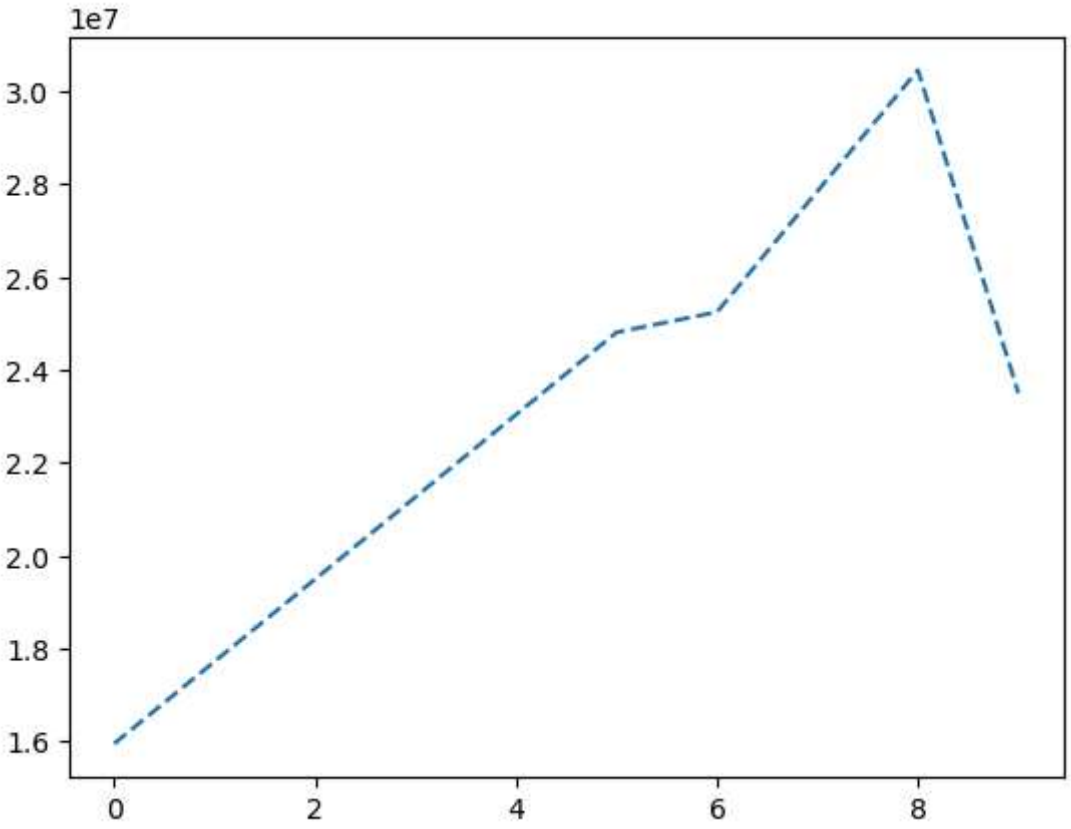
'-.' - dash-dot line style

':' - dotted line style

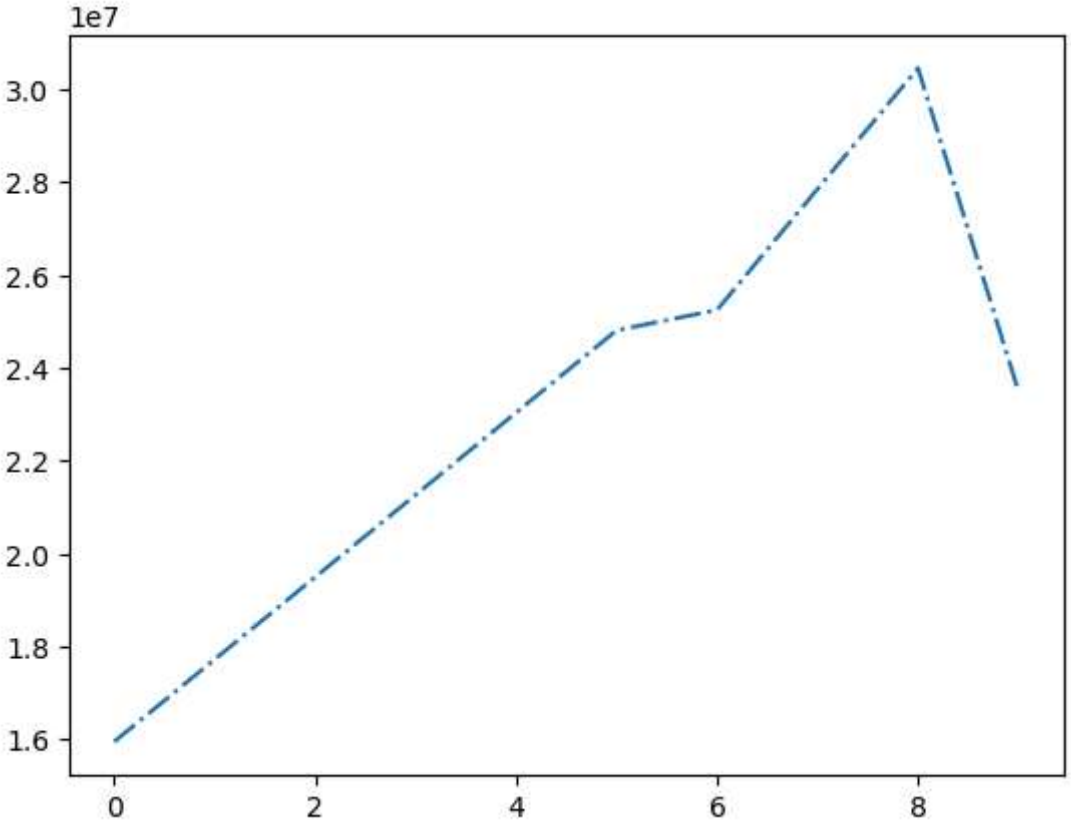
In [47]:
```
plt.plot(Salary[0],ls=':')
plt.show()  # we got the salaray of 0th player i.e Sachin with linestyle-....
```



In [48]:
```
plt.plot(Salary[0],ls='--')
plt.show()  # we got the salaray of 0th player i.e Sachin with linestyle: ----
```



In [49]:
```
plt.plot(Salary[0],ls='-.')
plt.show()  # we got the salaray of 0th player i.e Sachin with linestyle: -.-.-.-
```



we will upgrade the plotting by linestyle, Color

`Colors -c`

The supported color abbreviations are the single letter codes

character - color

'b' - blue

'g' - green

'r' - red

'c' - cyan

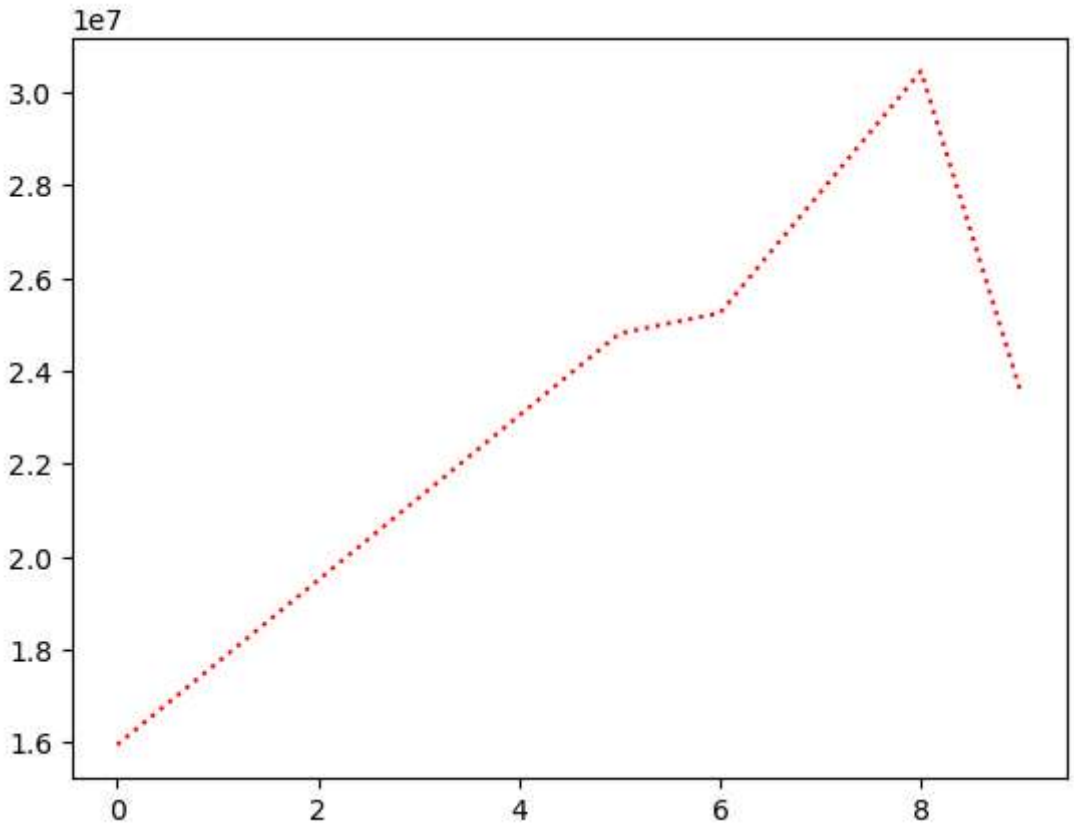
'm' - magenta

'y' - yellow

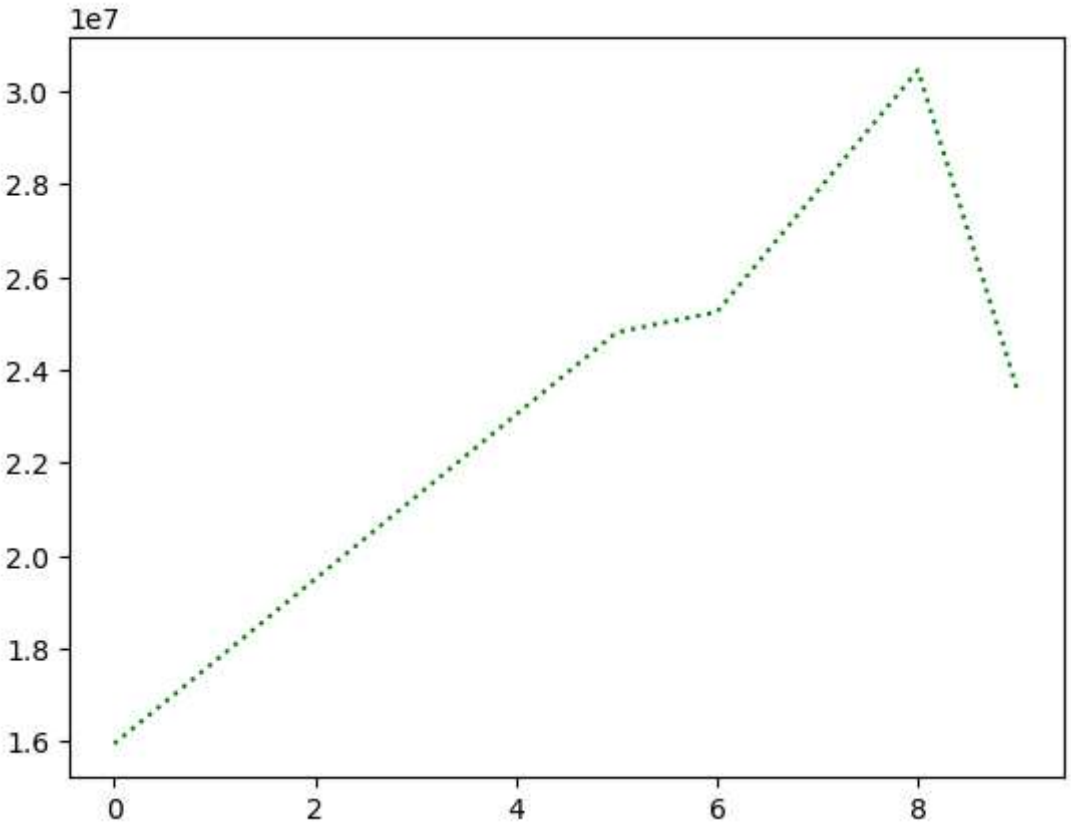
'k' - black

'w' - white white

In [51]:
```
plt.plot(Salary[0],ls=':',c='red')
plt.show()  # we got the salaray of 0th player i.e Sachin with linestyle-.... with
```



In [52]:
```
plt.plot(Salary[0],ls=':',c='Green')
plt.show()  # we got the salaray of 0th player i.e Sachin with linestyle-.... with
```



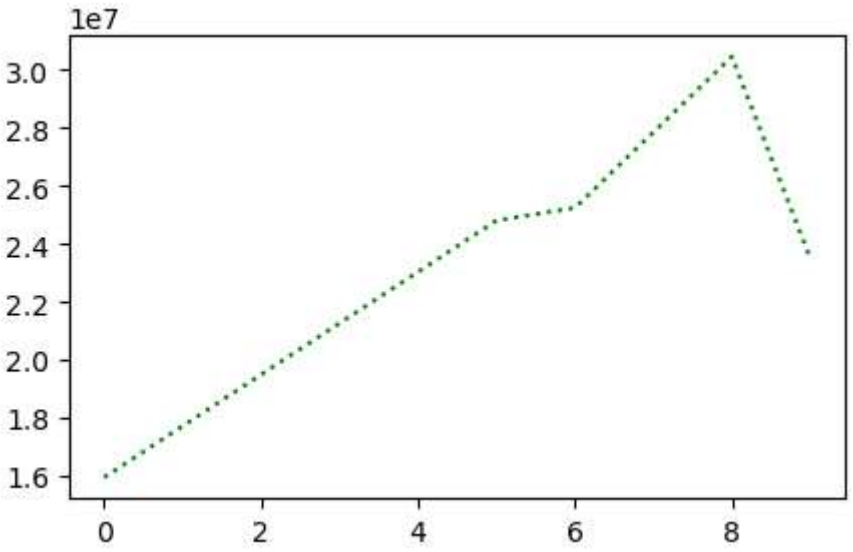
by adding parameters we can change more and more graph style and we can write parameters in short form also.

we will change the size of the graph because it is too big

In [55]:
```
# to reduc ethe size of the graph we will use this commands
```

In [56]:
```
%matplotlib inline
plt.rcParams['figure.figsize']=5,3
```

In [57]:
```
plt.plot(Salary[0],ls=':',c='Green')
plt.show()  # we got the salaray of 0th player i.e Sachin with linestyle-.... with
```



`we will upgrade the plotting by linestyle, Color, marker`

`Markers`

character - description

'.' - point marker

',' - pixel marker

'o' - circle marker

'v' - triangle_down marker

'^' - triangle_up marker

'<' - triangle_left marker

'>' - triangle_right marker

'1' - tri_down marker

'2'- tri_up marker

'3'- tri_left marker

'4' - tri_right marker

'8'- octagon marker

's'- square marker

'p'- pentagon marker

'P' - plus (filled) marker

'*' - star marker

'h'- hexagon1 marker

'H'- hexagon2 marker

'+'- plus marker

'x'- x marker

'X'- x (filled) marker

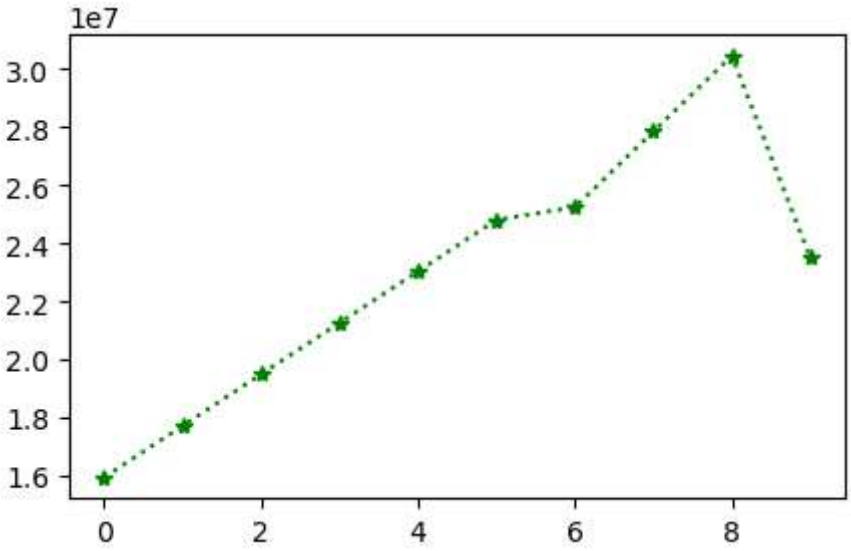
'D'- diamond marker

'd'- thin_diamond marker

'|'- vline marker

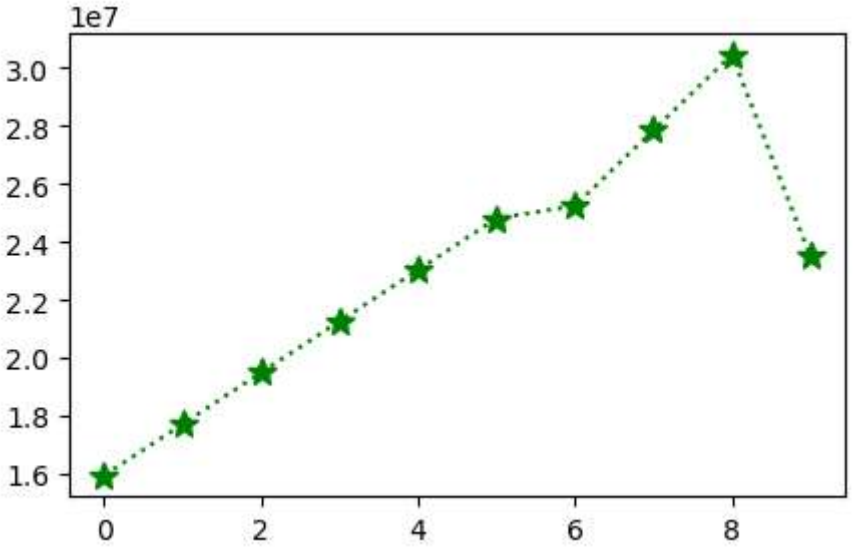
'_'- hline marker

In [59]:
```
plt.plot(Salary[0],ls=':',c='Green',marker='*')
plt.show()  # we got the salaray of 0th player i.e Sachin with linestyle-.... with
```



for better understanding we use other parameter i.e ms-markers style we need to mention in ms

In [61]:
```
# we got the salaray of 0th player i.e Sachin with linestyle-.... with color green
plt.plot(Salary[0],ls=':',c='Green',marker='*',ms=10)
plt.show()
```



In [62]:
```
Pdict
```

Out[62]:
```
{'Sachin': 0,
 'Rahul': 1,
 'Smith': 2,
 'Sami': 3,
 'Pollard': 4,
 'Morris': 5,
 'Samson': 6,
 'Dhoni': 7,
 'Kohli': 8,
 'Sky': 9}
```

In [63]:
```
Sdict
```

Out[63]:
```
{'2010': 0,
 '2011': 1,
 '2012': 2,
 '2013': 3,
 '2014': 4,
 '2015': 5,
 '2016': 6,
 '2017': 7,
 '2018': 8,
 '2019': 9}
```

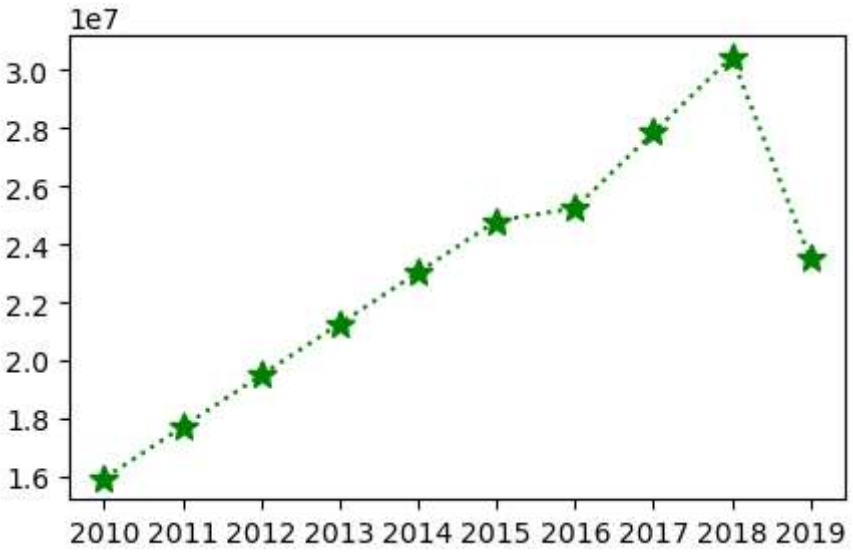
In [64]:
```
list(range(0,10))
```

Out[64]:
```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

x ticks means -x axis , y ticks means y- axis

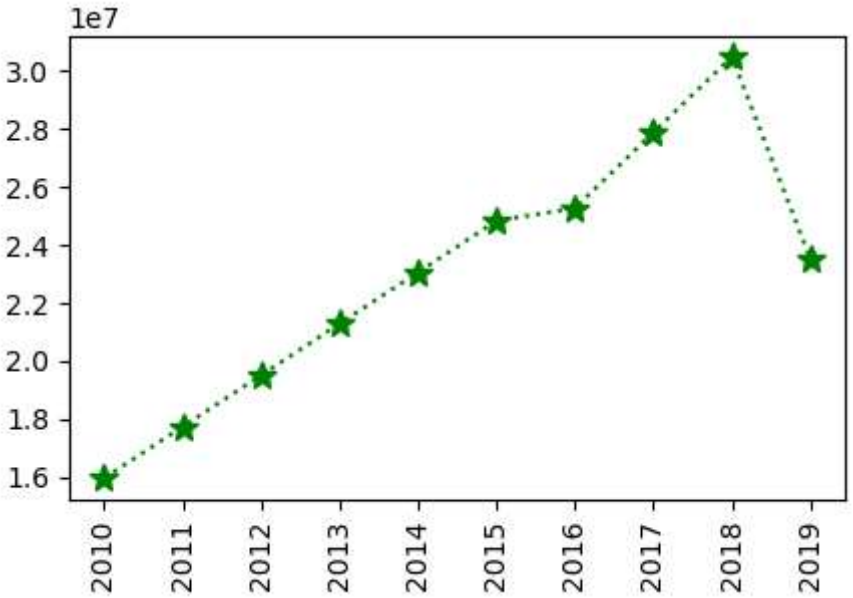
in the graph x axis the number are plotted that was a bit confusing but actually it denotes seasons for the better understanding we changed the x axis with seasons years by using list

In [67]:
```
plt.plot(Salary[0],ls=':',c='Green',marker='*',ms=10)
plt.xticks(list(range(0,10)),Seasons)
plt.show()
```

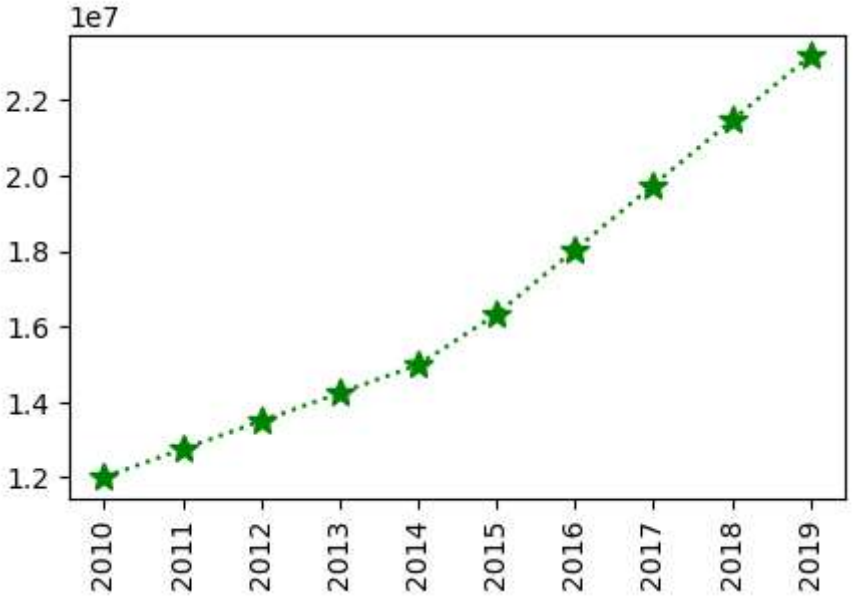


we will change the years i.e seasons to vertical for neat graph plot using rotation command

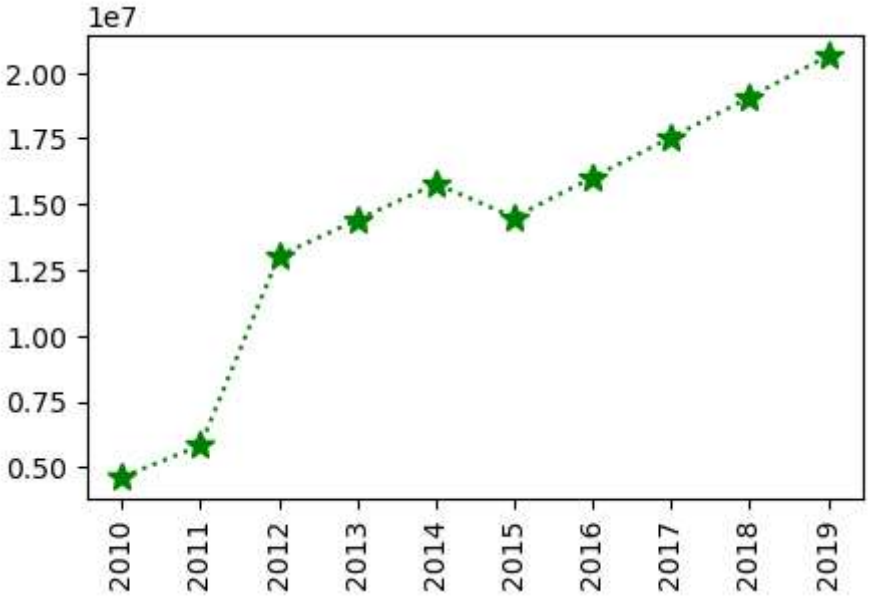
In [69]:
```
plt.plot(Salary[0],ls=':',c='Green',marker='*',ms=10)    # sachin
plt.xticks(list(range(0,10)),Seasons,rotation='vertical')
plt.show()
```



In [70]:
```
plt.plot(Salary[1],ls=':',c='Green',marker='*',ms=10)   # rahul
plt.xticks(list(range(0,10)),Seasons,rotation='vertical')
plt.show()
```

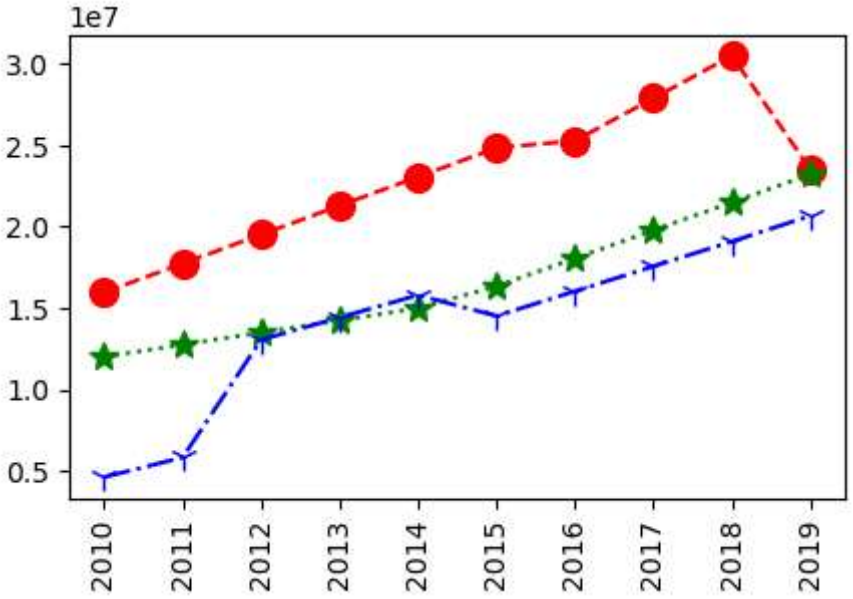


In [71]:
```
plt.plot(Salary[2],ls=':',c='Green',marker='*',ms=10)  # smith
plt.xticks(list(range(0,10)),Seasons,rotation='vertical')
plt.show()
```



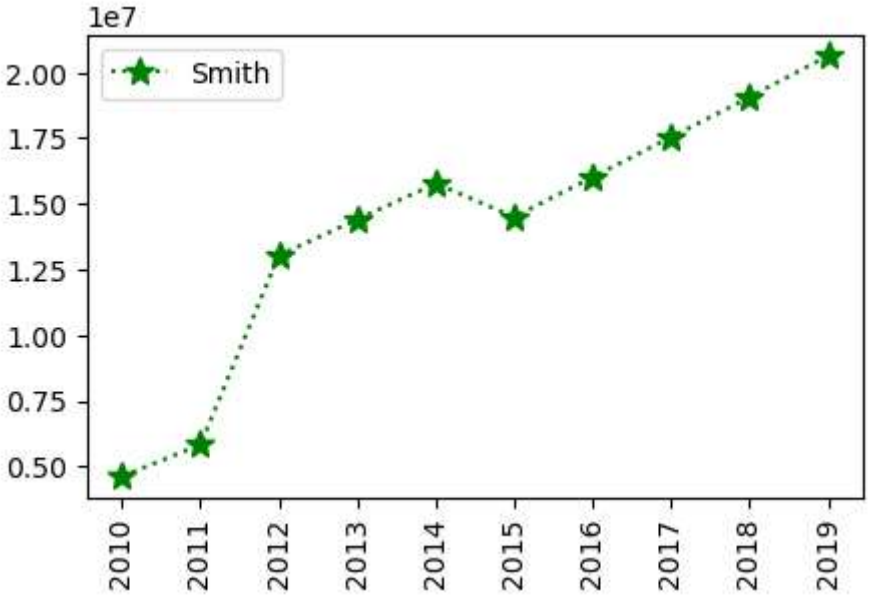
comparing of two players and more than that in a single graph

In [73]:
```
plt.plot(Salary[0],ls='--',c='red',marker='o',ms=10) # sachin
plt.plot(Salary[1],ls=':',c='Green',marker='*',ms=10) # sachin
plt.plot(Salary[2],ls='-.',c='blue',marker='1',ms=10)  # smith
plt.xticks(list(range(0,10)),Seasons,rotation='vertical')
plt.show()
```

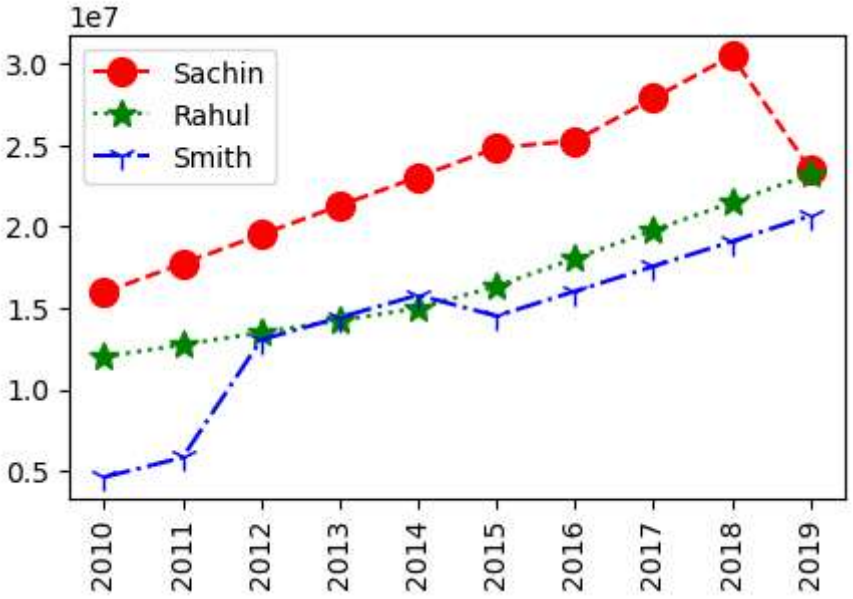


so in the above graph we cannot identify which color is defines which player so we use legend so it will display the player with symbol

In [75]:
```
plt.plot(Salary[2],ls=':',c='Green',marker='*',ms=10, label=Players[2])  # smith
plt.legend()
plt.xticks(list(range(0,10)),Seasons,rotation='vertical')
plt.show()
```

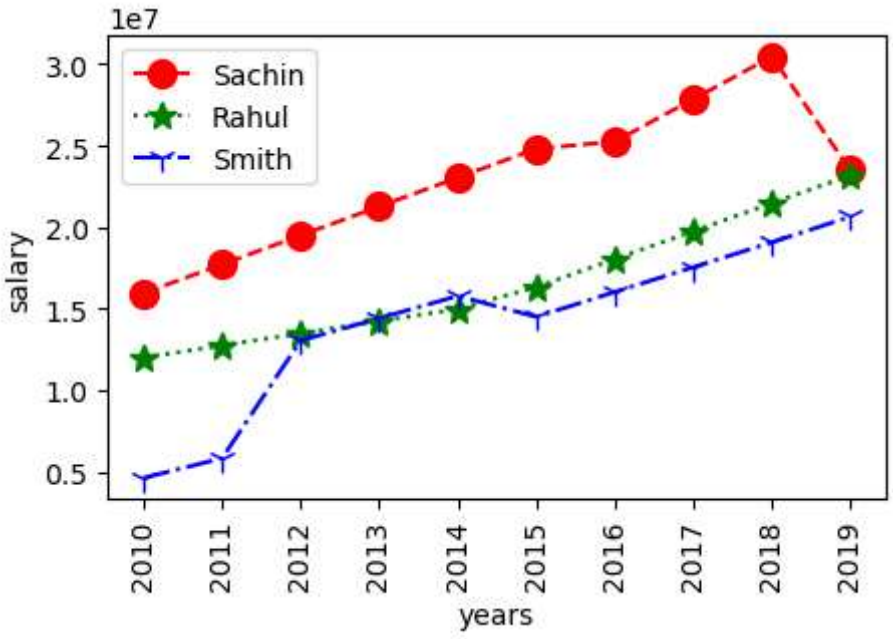


In [76]:
```
plt.plot(Salary[0],ls='--',c='red',marker='o',ms=10,label=Players[0]) # sachin
plt.plot(Salary[1],ls=':',c='Green',marker='*',ms=10,label=Players[1]) # rahul
plt.plot(Salary[2],ls='-.',c='blue',marker='1',ms=10,label=Players[2])  # smith
plt.legend()
plt.xticks(list(range(0,10)),Seasons,rotation='vertical')
plt.show()
```



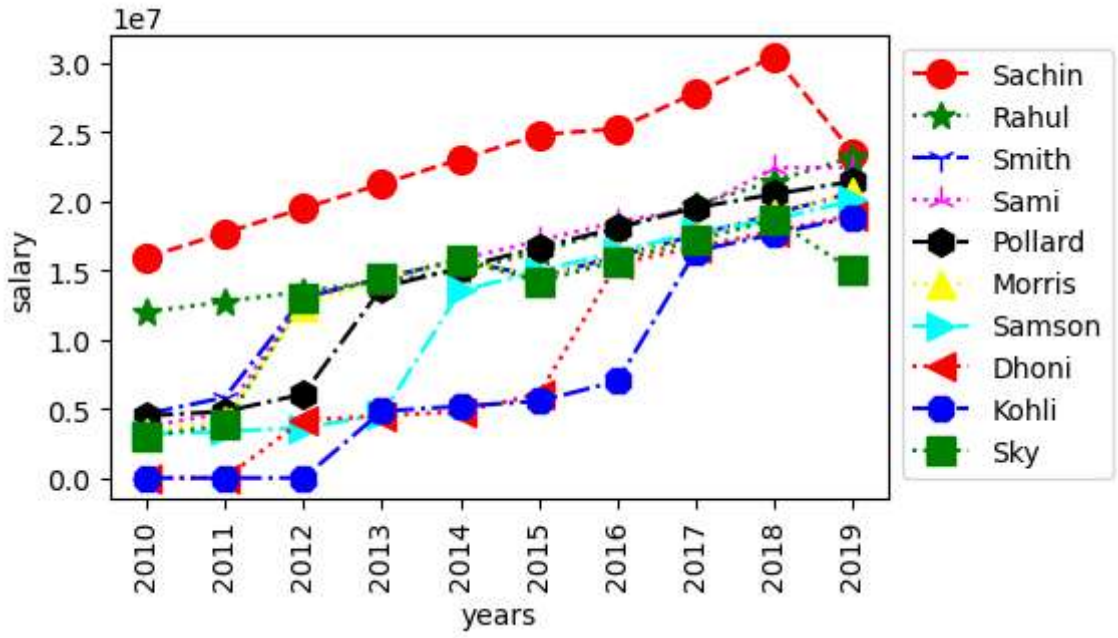
we will label the x axis and y axis

In [78]:
```
plt.plot(Salary[0],ls='--',c='red',marker='o',ms=10,label=Players[0]) # sachin
plt.plot(Salary[1],ls=':',c='Green',marker='*',ms=10,label=Players[1]) # rahul
plt.plot(Salary[2],ls='-.',c='blue',marker='1',ms=10,label=Players[2])  # smith
plt.legend()
plt.xlabel('years')
plt.ylabel('salary')
plt.xticks(list(range(0,10)),Seasons,rotation='vertical')
plt.show()
```



the labels in the graph is a bit clumsy so we will define it outside of the graph

In [80]:
```
plt.plot(Salary[0],ls='--',c='red',marker='o',ms=10,label=Players[0])
plt.plot(Salary[1],ls=':',c='Green',marker='*',ms=10,label=Players[1])
plt.plot(Salary[2],ls='-.',c='blue',marker='1',ms=10,label=Players[2])
plt.plot(Salary[3],ls=':',c='Magenta',marker='2',ms=10,label=Players[3])
plt.plot(Salary[4],ls='-.',c='black',marker='h',ms=10,label=Players[4])
plt.plot(Salary[5],ls=':',c='yellow',marker='^',ms=10,label=Players[5])
plt.plot(Salary[6],ls='-.',c='cyan',marker='>',ms=10,label=Players[6])
plt.plot(Salary[7],ls=':',c='red',marker='<',ms=10,label=Players[7])
plt.plot(Salary[8],ls='-.',c='blue',marker='8',ms=10,label=Players[8])
plt.plot(Salary[9],ls=':',c='Green',marker='s',ms=10,label=Players[9])
plt.legend(loc='upper left',bbox_to_anchor=(1,1))
plt.xlabel('years')
plt.ylabel('salary')
plt.xticks(list(range(0,10)),Seasons,rotation='vertical')
plt.show()
```



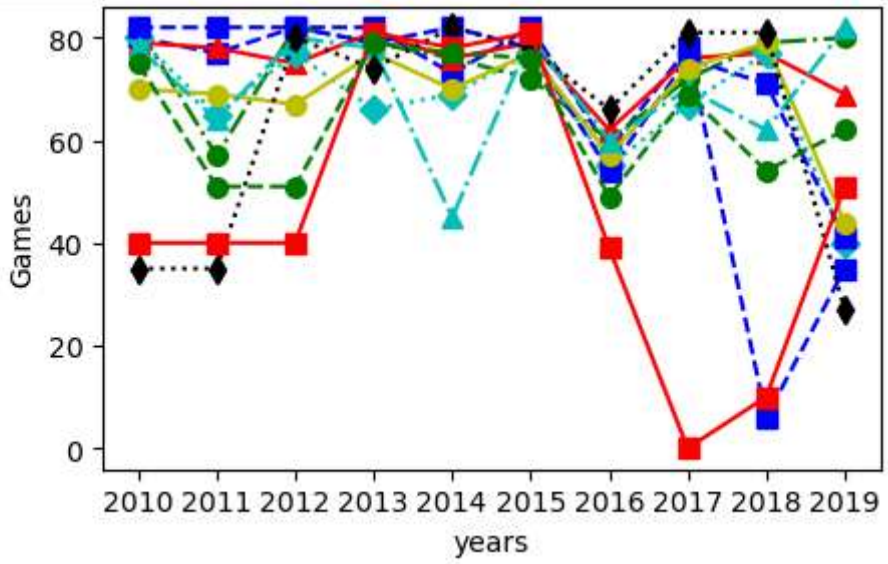
In [81]:
```
Games
```

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],
       [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 [82]:
```
plt.plot(Games[0], c='b', ls = '--', marker = 's', ms = 7, label = Games[0])
plt.plot(Games[1], c='g', ls = '-.', marker = 'o', ms = 7, label = Games[1])
plt.plot(Games[2], c='r', ls = '-', marker = '^', ms = 7, label = Games[2])
plt.plot(Games[3], c='c', ls = ':', marker = 'D', ms = 7, label = Games[3])
plt.plot(Games[4], c='b', ls = '--', marker = 's', ms = 7, label = Games[4])
plt.plot(Games[5], c='y', ls = '-', marker = 'o', ms = 7, label = Games[5])
```

```
plt.plot(Games[6], c='c', ls = '-.', marker = '^', ms = 7, label = Games[6])
plt.plot(Games[7], c='k', ls = ':', marker = 'd', ms = 7, label = Games[7])
plt.plot(Games[8], c='r', ls = '-', marker = 's', ms = 7, label = Games[8])
plt.plot(Games[9], c='g', ls = '--', marker = 'o', ms = 7, label = Games[9])
plt.legend(loc = 'lower right',bbox_to_anchor=(0.5,1) )
plt.xlabel('years')
plt.ylabel('Games')
plt.xticks(list(range(0,10)), Seasons,rotation='horizontal')
plt.show()
```

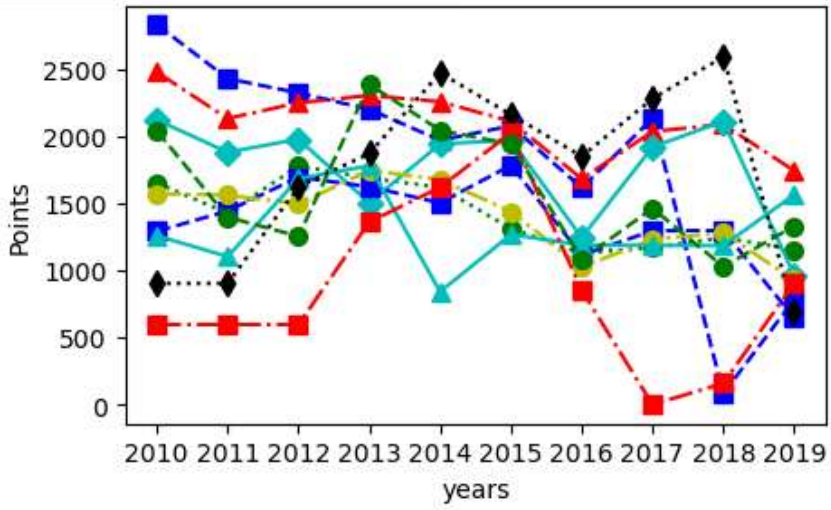
```
[80 77 82 82 73 82 58 78  6 35]
[82 57 82 79 76 72 60 72 79 80]
[79 78 75 81 76 79 62 76 77 69]
[80 65 77 66 69 77 55 67 77 40]
[82 82 82 79 82 78 54 76 71 41]
[70 69 67 77 70 77 57 74 79 44]
[78 64 80 78 45 80 60 70 62 82]
[35 35 80 74 82 78 66 81 81 27]
[40 40 40 81 78 81 39  0 10 51]
[75 51 51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



In [83]:

```
plt.plot(Points[0], c='b', ls = '--', marker = 's', ms = 7, label = Points[0])
plt.plot(Points[1], c='g', ls = ':', marker = 'o', ms = 7, label = Points[1])
plt.plot(Points[2], c='r', ls = '-.', marker = '^', ms = 7, label = Points[2])
plt.plot(Points[3], c='c', ls = '-', marker = 'D', ms = 7, label = Points[3])
plt.plot(Points[4], c='b', ls = '--', marker = 's', ms = 7, label = Points[4])
plt.plot(Points[5], c='y', ls = '-.', marker = 'o', ms = 7, label = Points[5])
plt.plot(Points[6], c='c', ls = '-', marker = '^', ms = 7, label = Points[6])
plt.plot(Points[7], c='k', ls = ':', marker = 'd', ms = 7, label = Points[7])
plt.plot(Points[8], c='r', ls = '-.', marker = 's', ms = 7, label = Points[8])
plt.plot(Points[9], c='g', ls = '--', marker = 'o', ms = 7, label = Points[9])
plt.xlabel('years')
plt.ylabel('Points')
plt.legend(loc = 'lower right',bbox_to_anchor=(0.5,1) )
plt.xticks(list(range(0,10)), Seasons,rotation='horizontal')
plt.show()
```

```
[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]
```



when we merge all the players into the graph it was not clean and look good that's why we to introduced bussiness intelligence

python is need the clean the raw data and feed to the table

for big data python is slow to clean and interactive graphs so we will use tools power bi

`In this project we learned matrics, buliding matrics, Dictionary in python, visualization using pyplot, ipl data analysis`