IPL DATA ANALAYSIS USING MATPLOTLIB LIBRARY

In [ ]:

```
#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,"2

#Players
Players = ["Sachin","Rahul","Smith","Sami","Pollard","Morris","Samson","Dhoni","Koh
Pdict = {"Sachin":0,"Rahul":1,"Smith":2,"Sami":3,"Pollard":4,"Morris":5,"Samson":6,

#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_

#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

#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 [147…
```
Salary
```

Out[147…
```
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 [149…
```
Games
```

Out[149…
```
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 [151…
```
Points
```

Out[151…
```
array([[2832, 2430, 2323, 2201, 1970, 2078, 1616, 2133,   83,  782],
       [1653, 1426, 1779, 1688, 1619, 1312, 1129, 1170, 1245, 1154],
       [2478, 2132, 2250, 2304, 2258, 2111, 1683, 2036, 2089, 1743],
       [2122, 1881, 1978, 1504, 1943, 1970, 1245, 1920, 2112,  966],
       [1292, 1443, 1695, 1624, 1503, 1784, 1113, 1296, 1297,  646],
       [1572, 1561, 1496, 1746, 1678, 1438, 1025, 1232, 1281,  928],
       [1258, 1104, 1684, 1781,  841, 1268, 1189, 1186, 1185, 1564],
       [ 903,  903, 1624, 1871, 2472, 2161, 1850, 2280, 2593,  686],
       [ 597,  597,  597, 1361, 1619, 2026,  852,    0,  159,  904],
       [2040, 1397, 1254, 2386, 2045, 1941, 1082, 1463, 1028, 1331]])
```

In [153…
```
Games[5]
```

Out[153… array([70, 69, 67, 77, 70, 77, 57, 74, 79, 44])

In [155…
```
Games[0:5]
```

Out[155…
```
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 [157…
```
Games[0,5]
```

Out[157… 82

In [159…
```
Salary
```

Out[159…
```
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 [69]:
```
Games
```

Out[69]:
```
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 [71]:
```
Salary/Games
```

Out[71]:
```
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 [73]:
```
np.round(Salary/Games)
```

Out[73]:
```
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.]])
```

Warnings ignore regarding project

In [75]:
```
import warnings
warnings.filterwarnings("ignore")
```

Assigned the matplotlib library as plt for data visuvalizatins

In [77]:
```
import matplotlib.pyplot as plt
plt
```

Out[77]:
```
<module 'matplotlib.pyplot' from 'C:\\Users\\SAIF SHAIK\\anaconda3\\Lib\\site-packages\\matplotlib\\pyplot.py'>
```

In [79]:
```
Salary
```

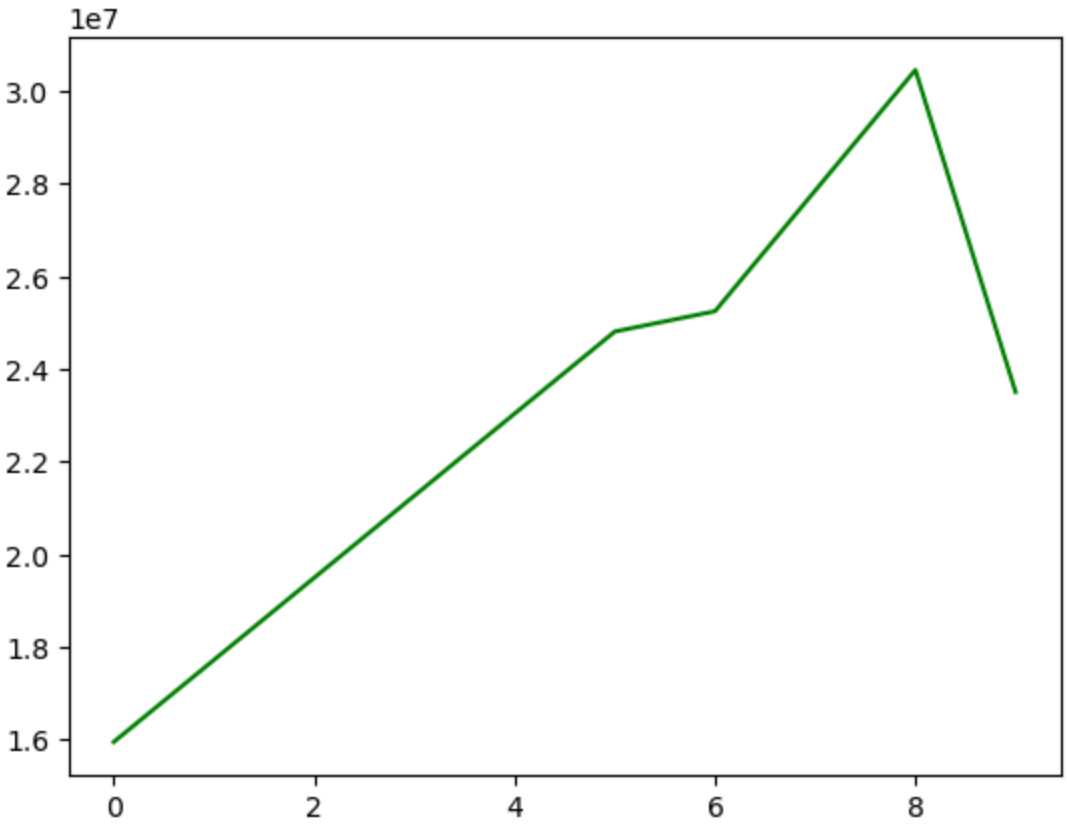
Out[79]:
```
array([[15946875, 17718750, 19490625, 21262500, 23034375, 24806250,
        25244493, 27849149, 30453805, 23500000],
       [12000000, 12744189, 13488377, 14232567, 14976754, 16324500,
        18038573, 19752645, 21466718, 23180790],
       [ 4621800,  5828090, 13041250, 14410581, 15779912, 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 [81]:
```
Salary[0]
```

Out[81]:
```
array([15946875, 17718750, 19490625, 21262500, 23034375, 24806250,
       25244493, 27849149, 30453805, 23500000])
```

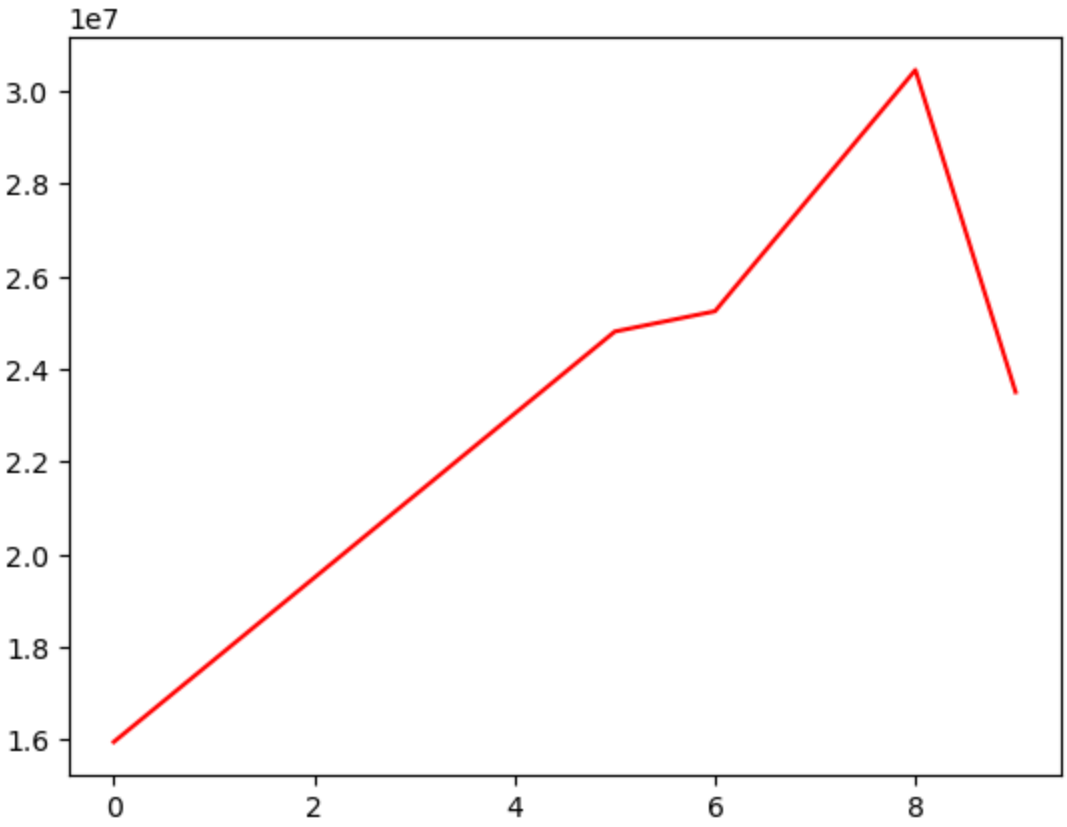
Plot()functions in plt

In [83]:
```
plt.plot(Salary[0], c = 'g')
plt.show()
```



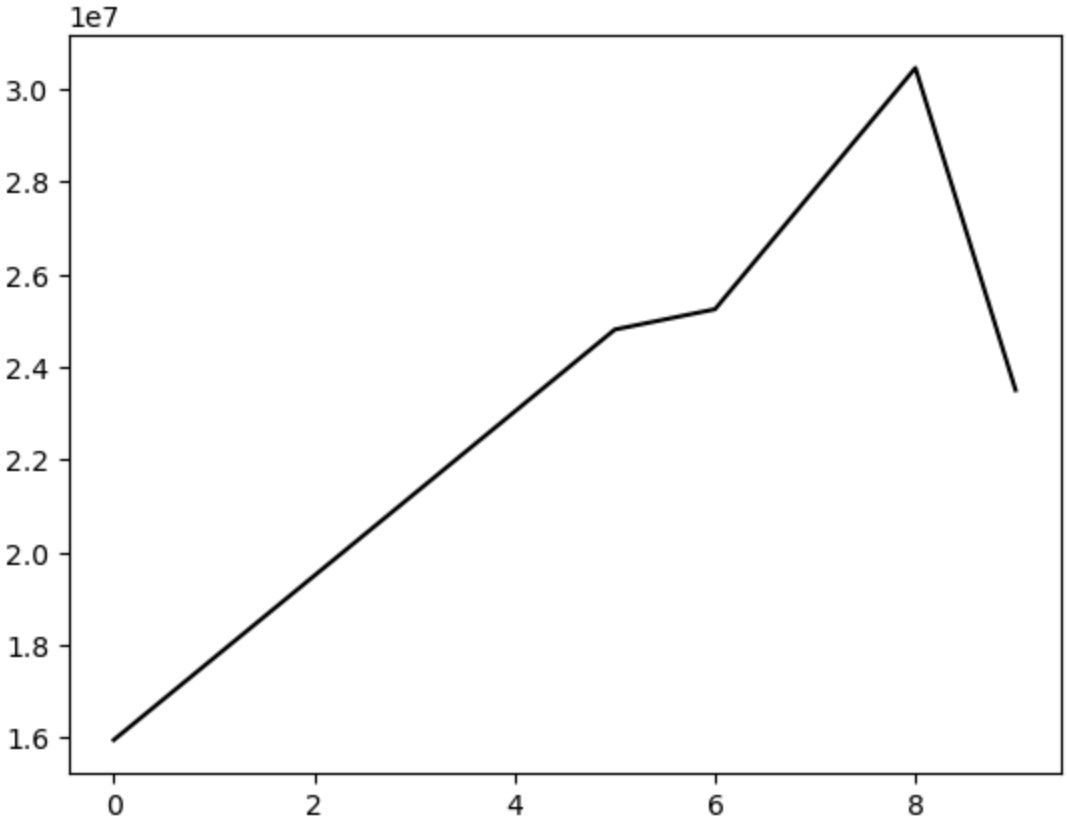
c is color function to plot a graph

In [85]:
```
plt.plot(Salary[0],c='r')
plt.show()
```



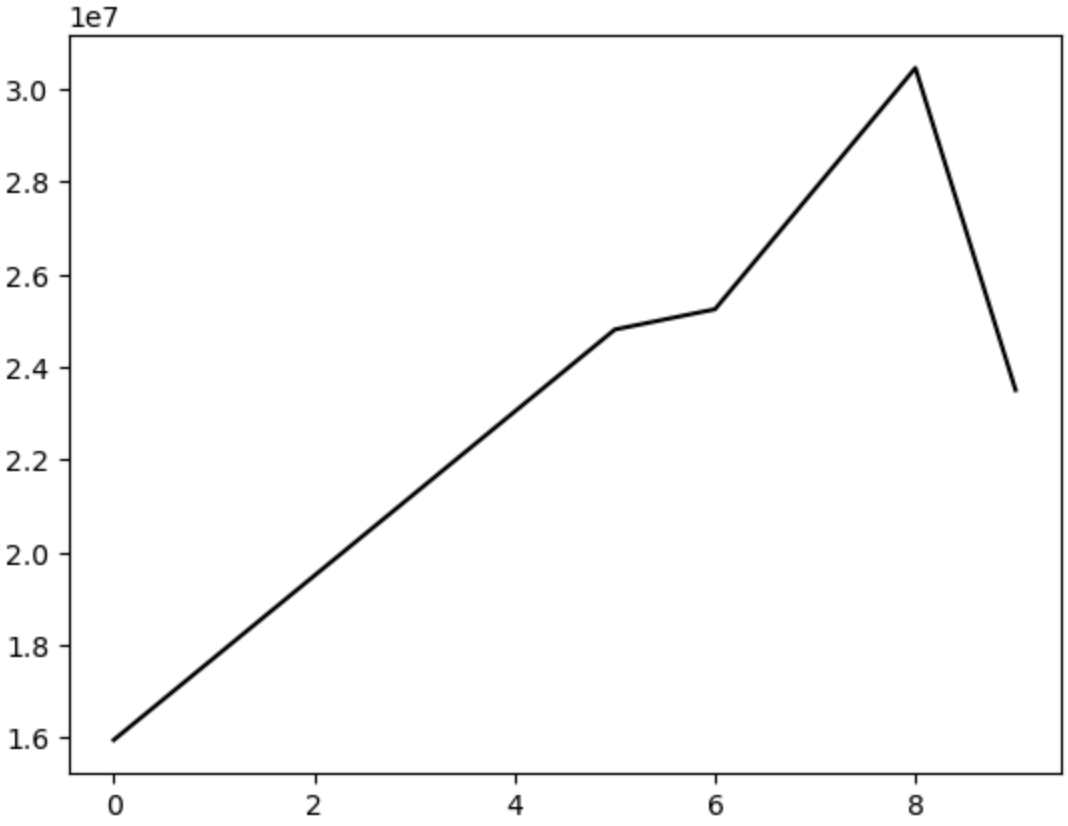
In [87]:
```
plt.plot(Salary[0],color ='black')
```

Out[87]:
```
[<matplotlib.lines.Line2D at 0x205e5be3440>]
```



In [89]:
```
plt.plot(Salary[0],c='k')
```

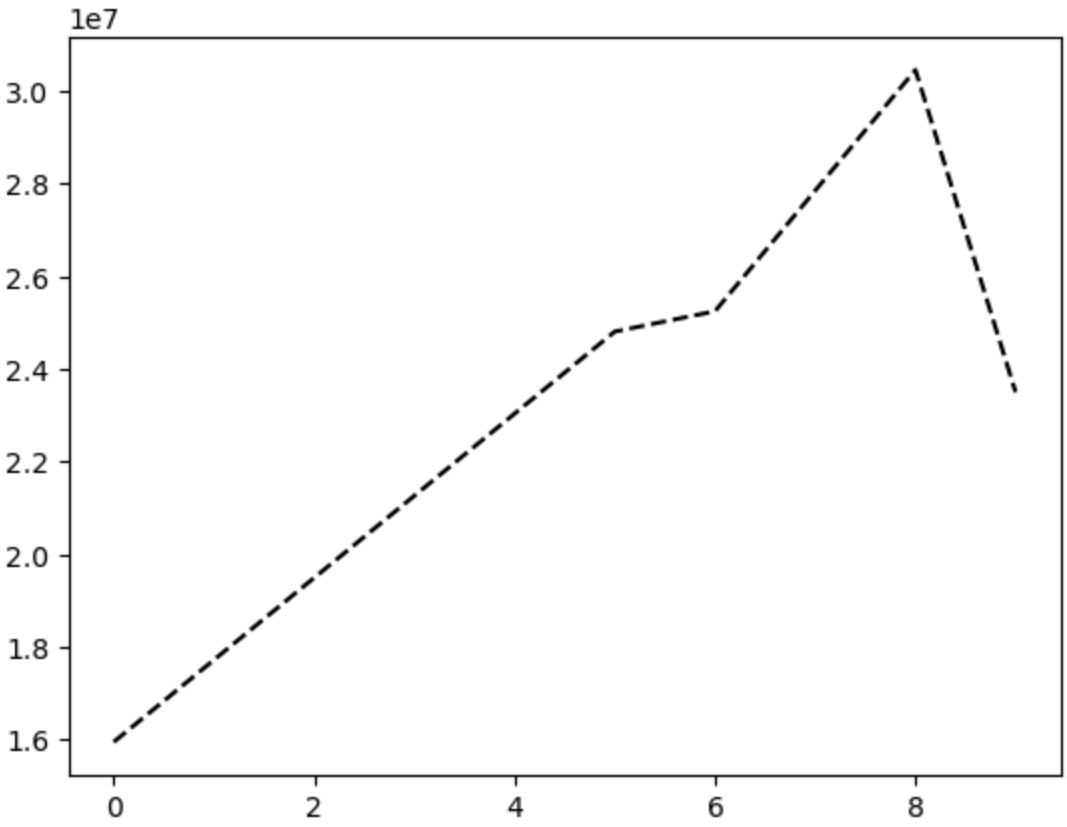
Out[89]:
```
[<matplotlib.lines.Line2D at 0x205e6c9a120>]
```



ls means line space in graph

In [91]:
```
plt.plot(Salary[0],c='k',ls= '--')
```

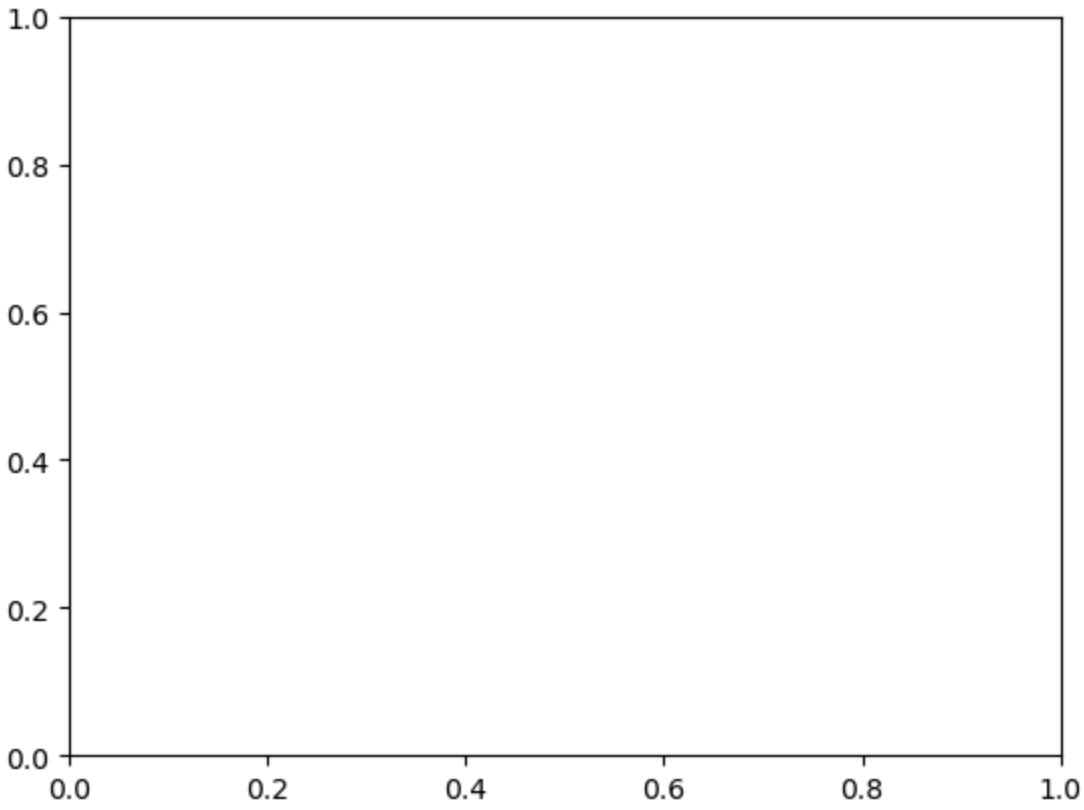
Out[91]:
```
[<matplotlib.lines.Line2D at 0x205e6d1ff80>]
```



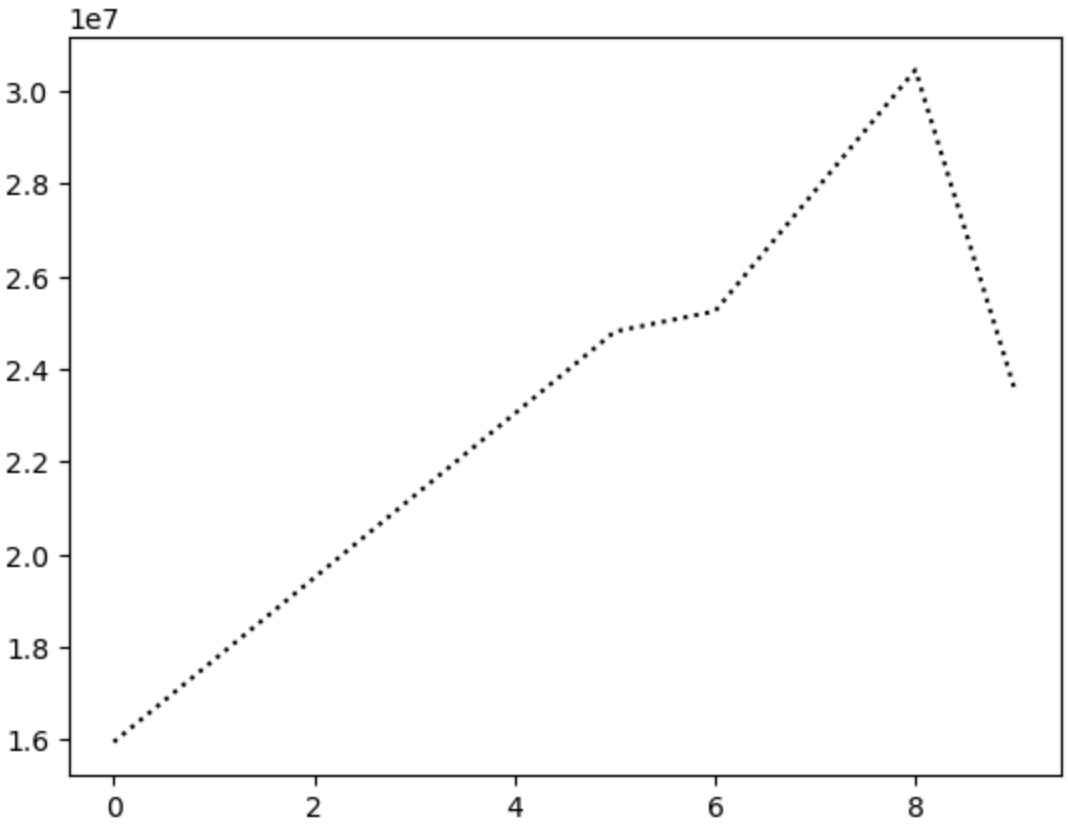
In [93]:
```
plt.plot(Salary[0],c='r',ls = '*')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[93], line 1
----> 1 plt.plot(Salary[0],c='r',ls = '*')

File ~\anaconda3\Lib\site-packages\matplotlib\pyplot.py:3794, in plot(scalex, scale
y, data, *args, **kwargs)
   3786 @_copy_docstring_and_deprecators(Axes.plot)
   3787 def plot(
   3788     *args: float | ArrayLike | str,
   (...)
   3792     **kwargs,
   3793 ) -> list[Line2D]:
-> 3794     return gca().plot(
   3795         *args,
   3796         scalex=scalex,
   3797         scaley=scaley,
   3798         **({"data": data} if data is not None else {}),
   3799         **kwargs,
   3800     )

File ~\anaconda3\Lib\site-packages\matplotlib\axes\_axes.py:1779, in Axes.plot(self,
scalex, scaley, data, *args, **kwargs)
   1536 """
   1537 Plot y versus x as lines and/or markers.
   1538
   (...)
   1776 (``'green'``) or hex strings (``'#008000'``).
   1777 """
   1778 kwargs = cbook.normalize_kwargs(kwargs, mlines.Line2D)
-> 1779 lines = [*self._get_lines(self, *args, data=data, **kwargs)]
   1780 for line in lines:
   1781     self.add_line(line)

File ~\anaconda3\Lib\site-packages\matplotlib\axes\_base.py:296, in _process_plot_va
r_args.__call__(self, axes, data, *args, **kwargs)
    294     this += args[0],
    295     args = args[1:]
--> 296 yield from self._plot_args(
    297     axes, this, kwargs, ambiguous_fmt_datakey=ambiguous_fmt_datakey)

File ~\anaconda3\Lib\site-packages\matplotlib\axes\_base.py:534, in _process_plot_va
r_args._plot_args(self, axes, tup, kwargs, return_kwargs, ambiguous_fmt_datakey)
    532     return list(result)
    533 else:
--> 534     return [l[0] for l in result]

File ~\anaconda3\Lib\site-packages\matplotlib\axes\_base.py:527, in <genexpr>(.0)
    522 else:
    523     raise ValueError(
    524         f"label must be scalar or have the same length as the input "
    525         f"data, but found {len(label)} for {n_datasets} datasets.")
--> 527 result = (make_artist(axes, x[:, j % ncx], y[:, j % ncy], kw,
    528                       {**kwargs, 'label': label})
    529           for j, label in enumerate(labels))
    531 if return_kwargs:
```

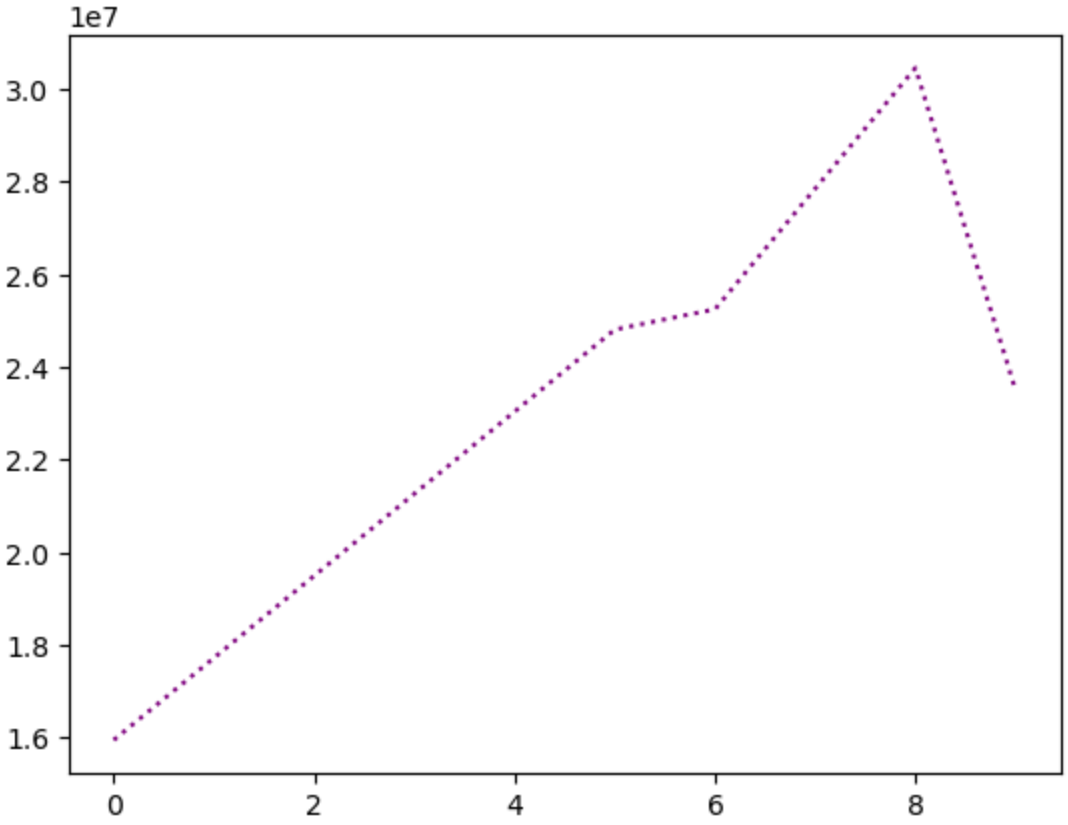
```
    532     return list(result)

File ~\anaconda3\Lib\site-packages\matplotlib\axes\_base.py:335, in _process_plot_va
r_args._makeline(self, axes, x, y, kw, kwargs)
    333 kw = {**kw, **kwargs}  # Don't modify the original kw.
    334 self._setdefaults(self._getdefaults(kw), kw)
--> 335 seg = mlines.Line2D(x, y, **kw)
    336 return seg, kw

File ~\anaconda3\Lib\site-packages\matplotlib\lines.py:372, in Line2D.__init__(self,
xdata, ydata, linewidth, linestyle, color, gapcolor, marker, markersize, markeredgew
idth, markeredgecolor, markerfacecolor, markerfacecoloralt, fillstyle, antialiased,
dash_capstyle, solid_capstyle, dash_joinstyle, solid_joinstyle, pickradius, drawstyl
e, markevery, **kwargs)
    369 self._dash_pattern = (0, None)  # offset, dash (scaled by linewidth)
    371 self.set_linewidth(linewidth)
--> 372 self.set_linestyle(linestyle)
    373 self.set_drawstyle(drawstyle)
    375 self._color = None

File ~\anaconda3\Lib\site-packages\matplotlib\lines.py:1177, in Line2D.set_linestyle
(self, ls)
   1175 if ls in [' ', '', 'none']:
   1176     ls = 'None'
-> 1177 _api.check_in_list([*self._lineStyles, *ls_mapper_r], ls=ls)
   1178 if ls not in self._lineStyles:
   1179     ls = ls_mapper_r[ls]

File ~\anaconda3\Lib\site-packages\matplotlib\_api\__init__.py:129, in check_in_list
(values, _print_supported_values, **kwargs)
    127 if _print_supported_values:
    128     msg += f"; supported values are {', '.join(map(repr, values))}"
--> 129 raise ValueError(msg)

ValueError: '*' is not a valid value for ls; supported values are '-', '--', '-.',
':', 'None', ' ', '', 'solid', 'dashed', 'dashdot', 'dotted'
```



In [95]:
```
plt.plot(Salary[0],c='k',ls=':')
plt.show()
```



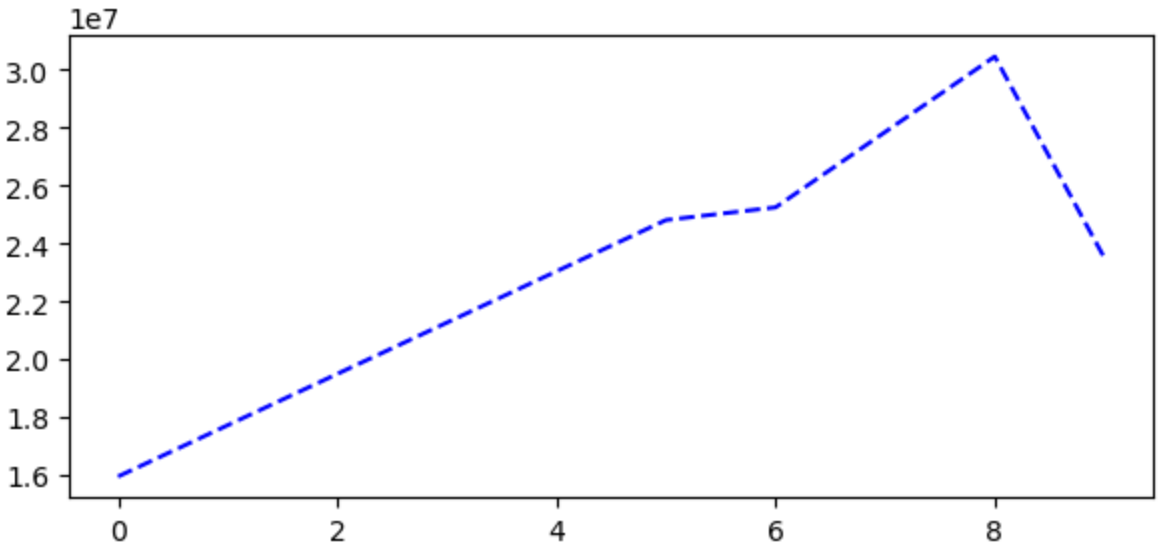
In [97]:
```
plt.plot(Salary[0],c='purple',ls='dotted')
plt.show()
```



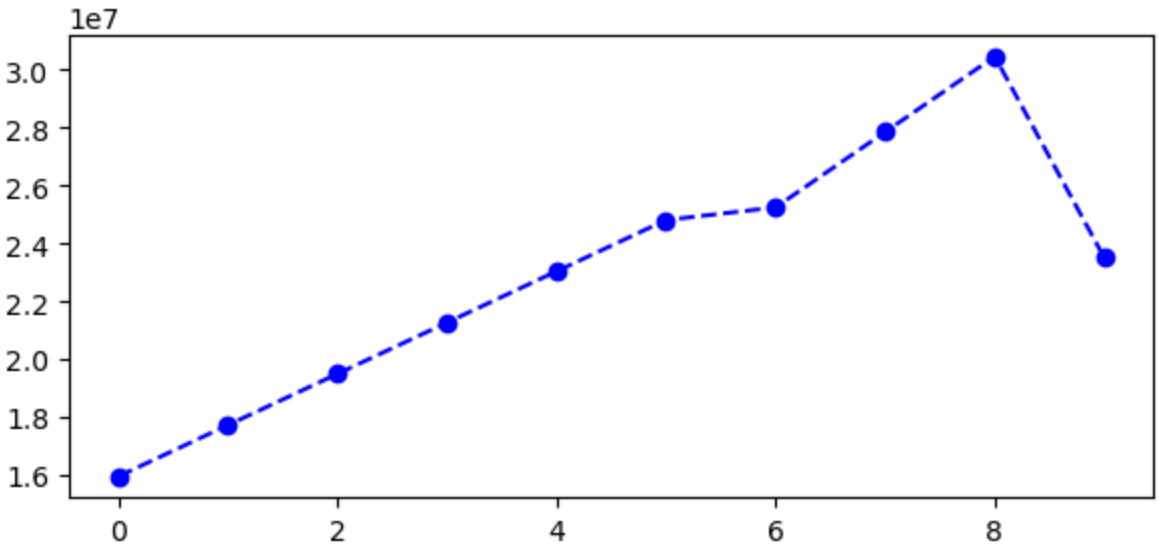
rcParams() function is for the figuresize to plot the width and height

In [99]:
```
plt.rcParams['figure.figsize']= 7,3
```

In [101…
```
plt.plot(Salary[0],c='Blue',ls='--')
plt.show()
```



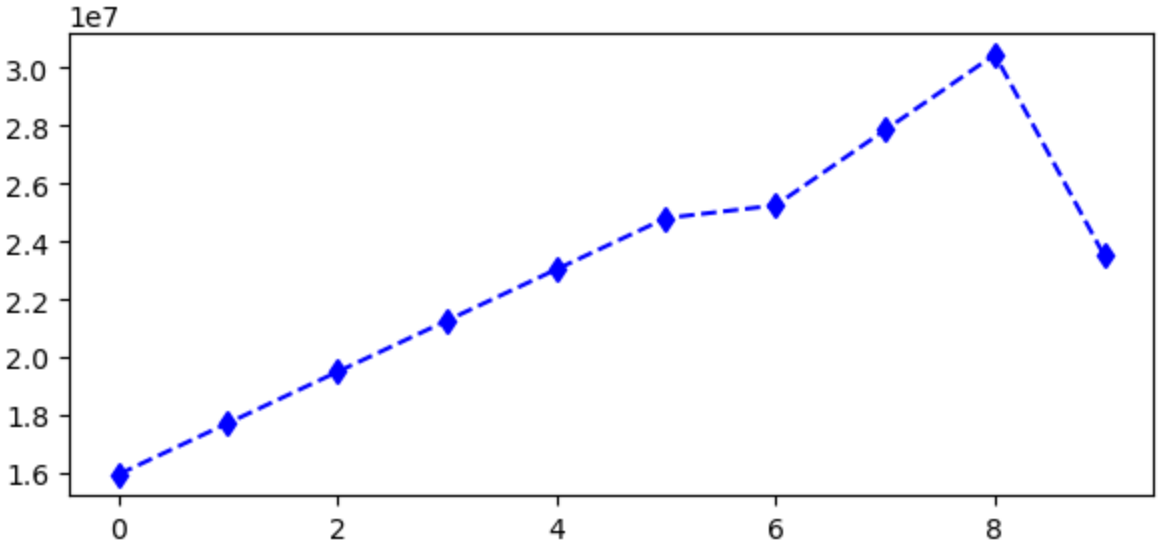
In [102…
```
plt.plot(Salary[0],c='Blue',ls='--',marker='o')
plt.show()
```



marker function is for the markering the points in the graphs

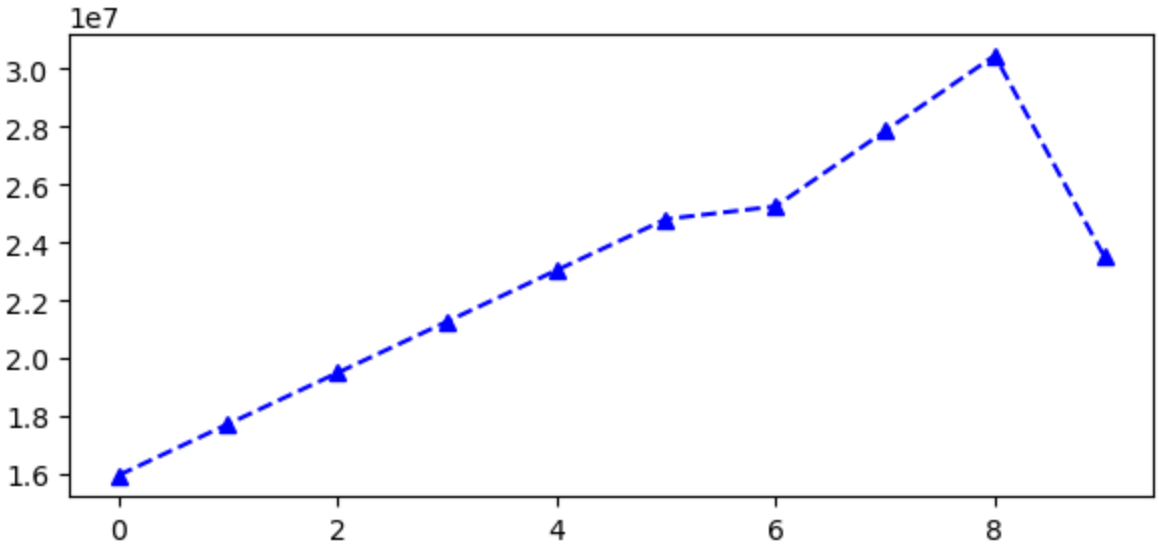
In [104…
```
plt.plot(Salary[0],c='Blue',ls='--',marker='d')
```

Out[104…
```
[<matplotlib.lines.Line2D at 0x205e6ee5400>]
```



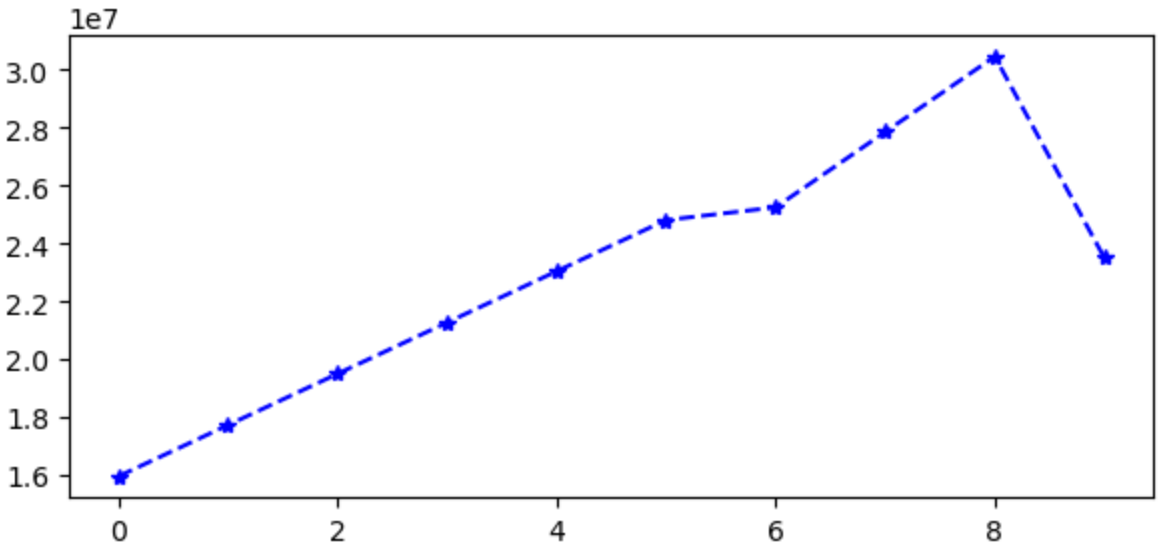
In [107…
```
plt.plot(Salary[0],c='Blue',ls='--',marker='^')
```

Out[107…
```
[<matplotlib.lines.Line2D at 0x205e5b43b60>]
```



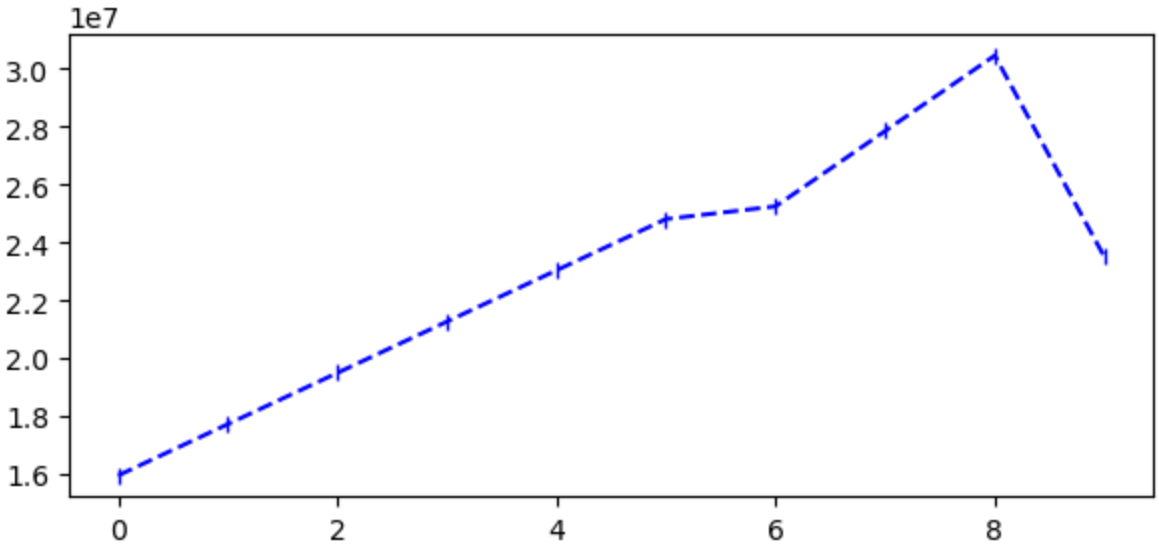
In [109…
```
plt.plot(Salary[0],c='Blue',ls='--',marker='*')
```

Out[109…
```
[<matplotlib.lines.Line2D at 0x205e6f4d340>]
```



In [111…
```
plt.plot(Salary[0],c='Blue',ls='--',marker='|')
```

Out[111…
```
[<matplotlib.lines.Line2D at 0x205e6f7b350>]
```



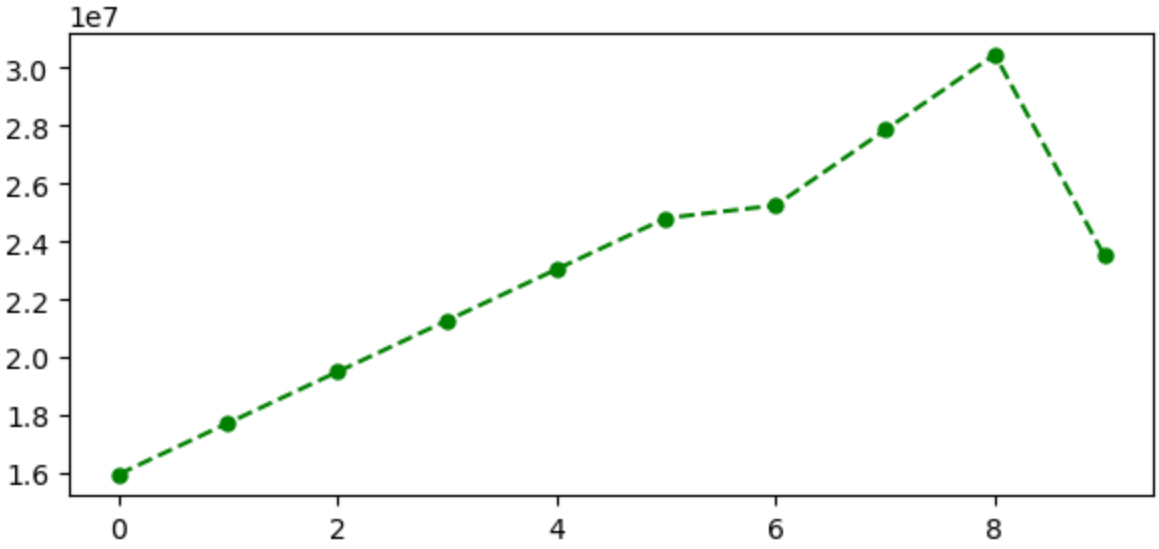
In [113…
```
Games
```

Out[113…
```
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]])
```

ms means marker size

In [115…
```
plt.plot(Salary[0],c='Green',ls='--',marker='o',ms= 5)
```

Out[115…
```
[<matplotlib.lines.Line2D at 0x205e6fffa40>]
```



In [117…
```
list(range(0,10))
```

Out[117…
```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Salary dict prints salarys of the players

In [119…
```
Sdict
```

Out[119…
```
{'2015': 0,
 '2016': 1,
 '2017': 2,
 '2018': 3,
 '2019': 4,
 '2020': 5,
 '2021': 6,
 '2022': 7,
 '2023': 8,
 '2024': 9}
```

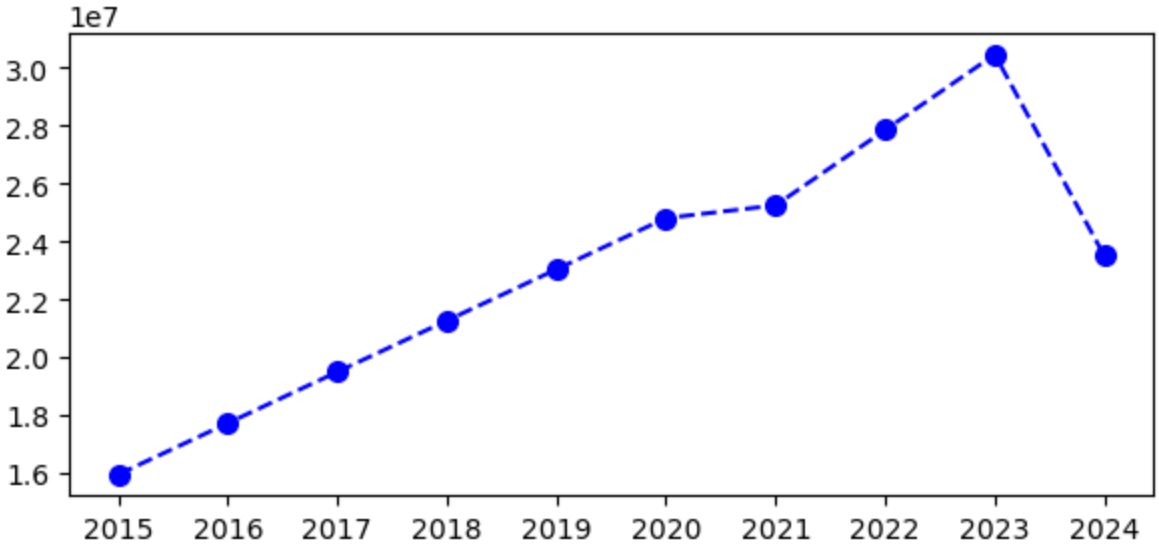
prints players name and values in the dict

In [121…
```
Pdict
```

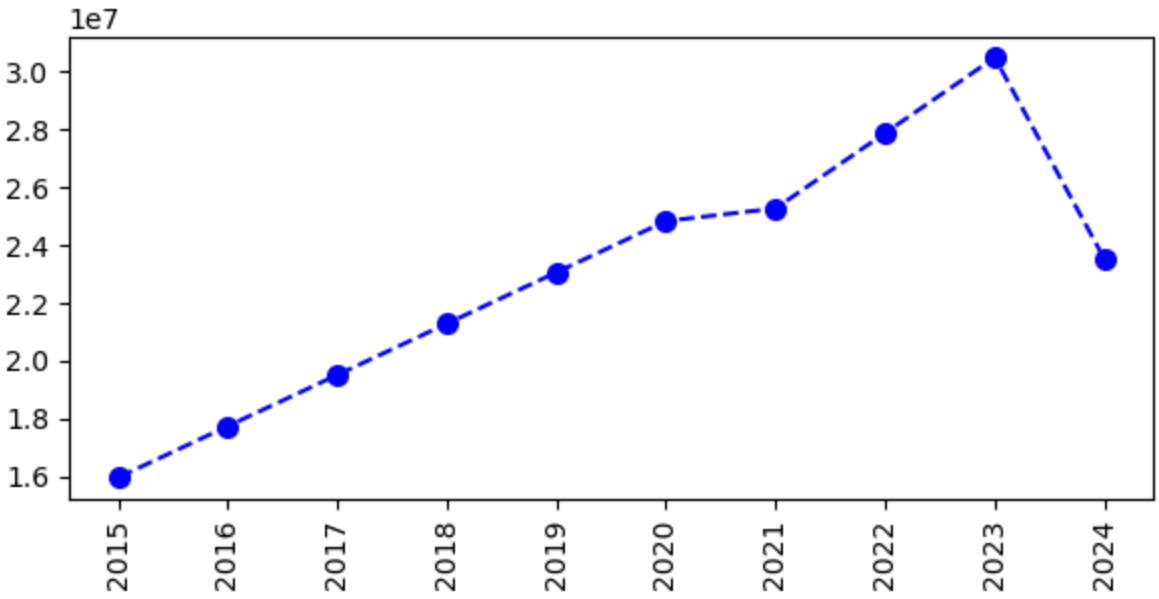
Out[121…
```
{'Sachin': 0,
 'Rahul': 1,
 'Smith': 2,
 'Sami': 3,
 'Pollard': 4,
 'Morris': 5,
 'Samson': 6,
 'Dhoni': 7,
 'Kohli': 8,
 'Sky': 9}
```

ploting the xaxis and yaxis is like are xticks() and yticks()

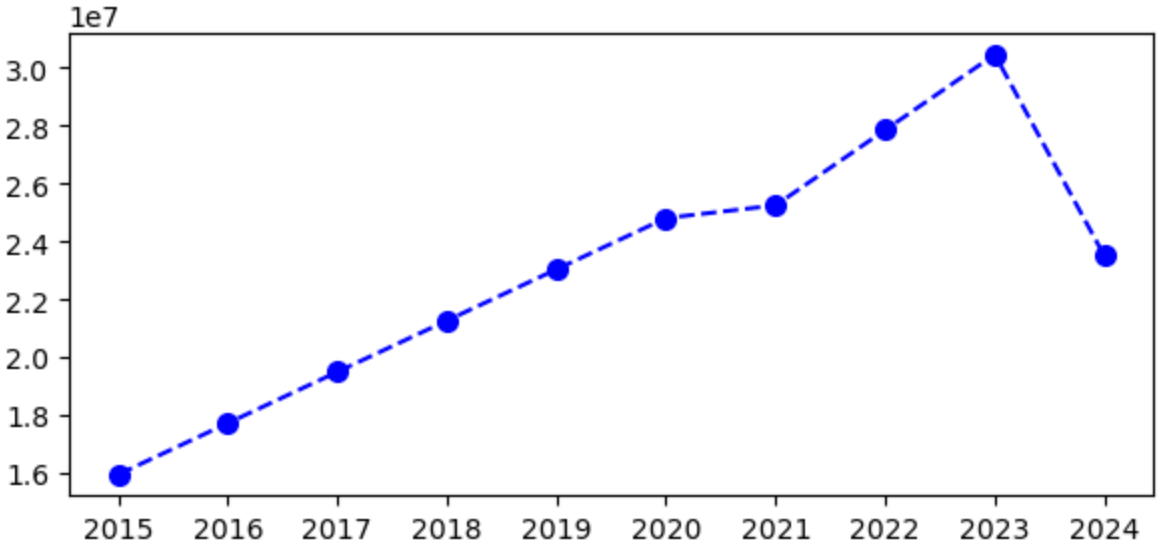
In [123…
```
plt.plot(Salary[0],c='Blue',ls='--',marker='o',ms=7)
plt.xticks(list(range(0,10)),Seasons)
plt.show()
```



In [125…
```
plt.plot(Salary[0],c='Blue',ls='--',marker='o',ms=7)
plt.xticks(list(range(0,10)),Seasons,rotation='vertical')
plt.show()
```



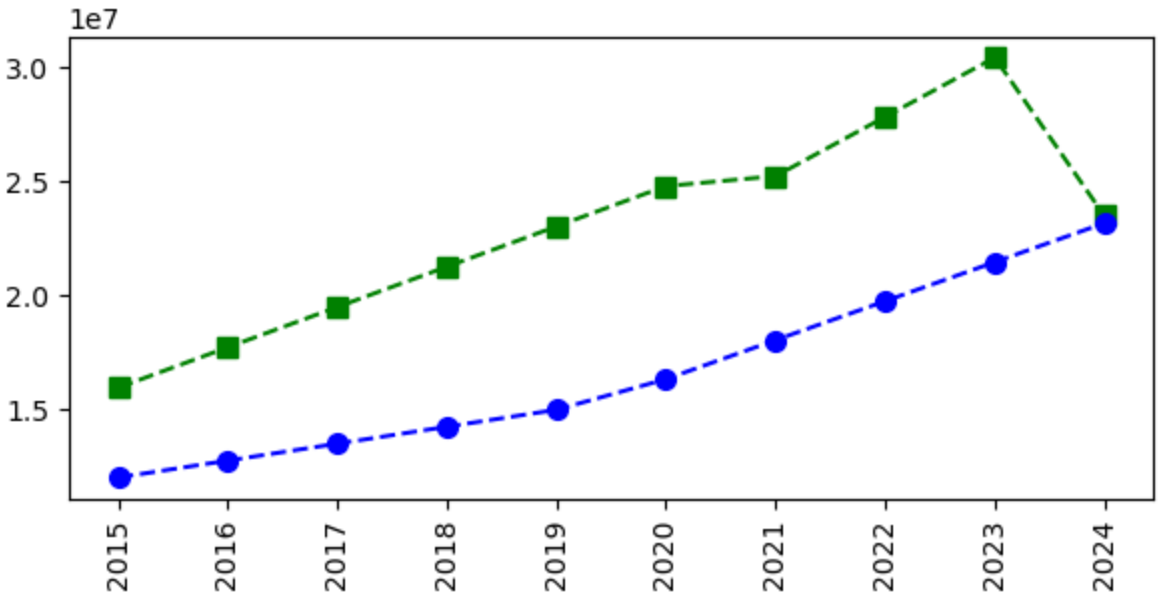
In [127…
```
plt.plot(Salary[0],c='Blue',ls='--',marker='o',ms=7)
plt.xticks(list(range(0,10)),Seasons,rotation='horizontal')
plt.show()
```



In [134…
```
Salary[1]
```

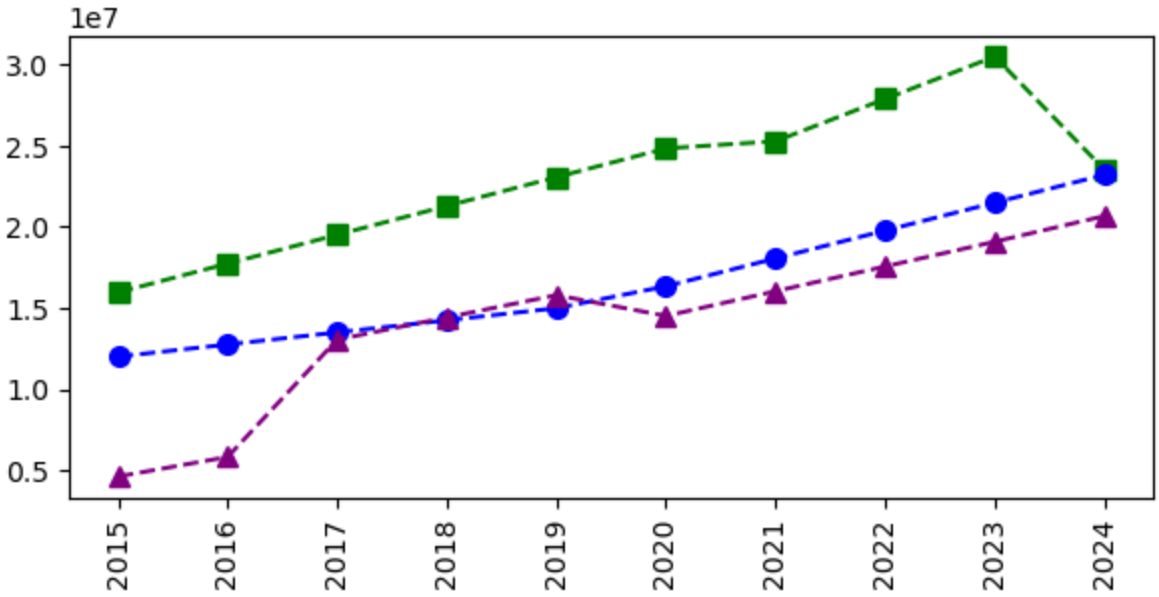
Out[134…
```
array([12000000, 12744189, 13488377, 14232567, 14976754, 16324500,
       18038573, 19752645, 21466718, 23180790])
```

In [138…
```
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.xticks(list(range(0,10)),Seasons,rotation='vertical')
plt.show()
```



In [140…

```
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='purple',ls='--',marker='^',ms=7,label =Players[2])

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

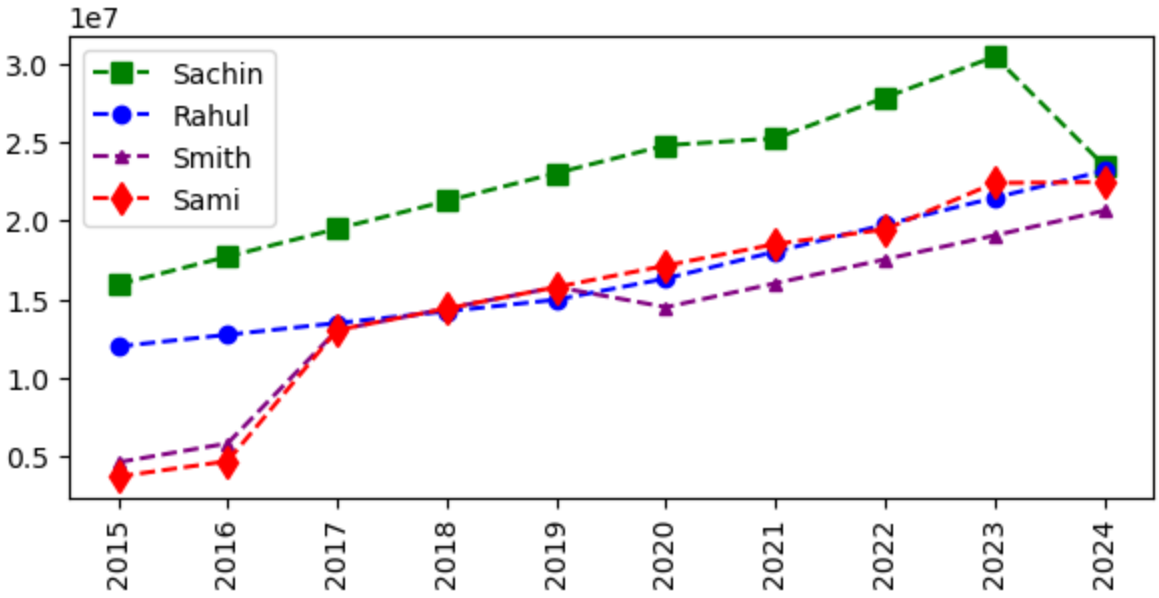


legend() function for using the names and identifing the markers assigned to which players inside the plot graphs

In [142…

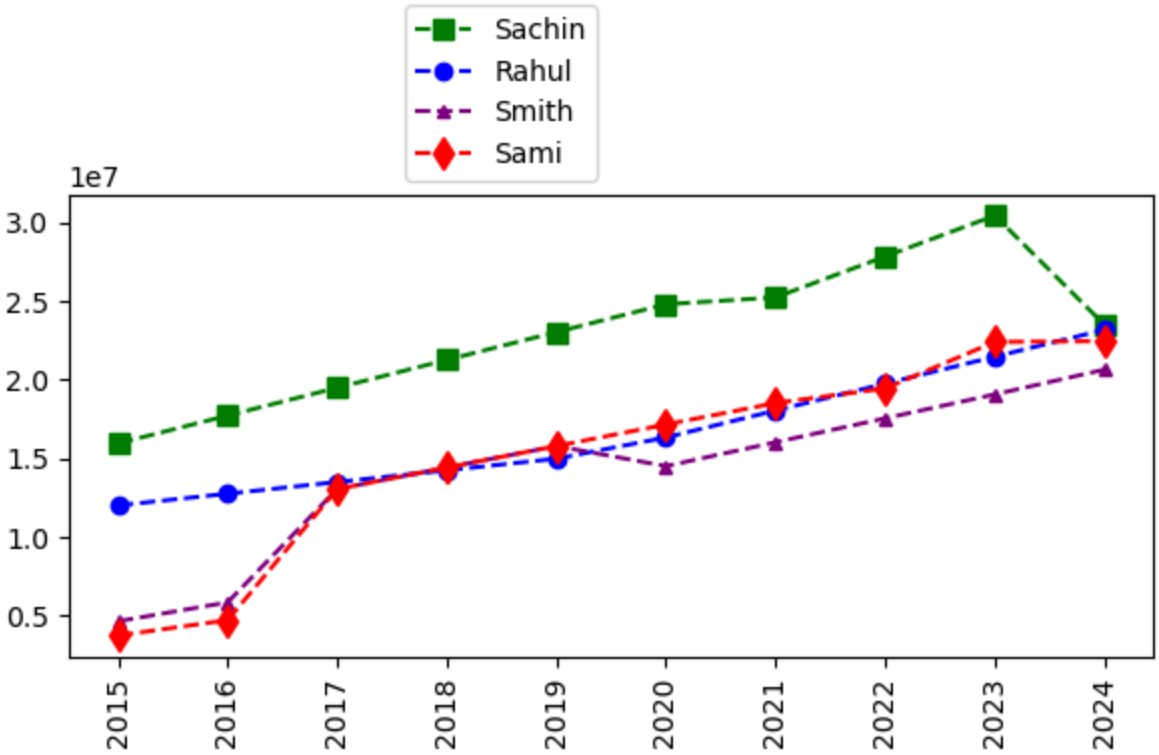
```
plt.plot(Salary[0],c='green',ls='--',marker='s',ms=7,label=Players[0])
plt.plot(Salary[1],c='Blue',ls='--',marker='o',ms=6,label =Players[1])
plt.plot(Salary[2],c='purple',ls='--',marker='^',ms=5,label =Players[2])
```

```
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 [ ]:
```
# bbox_to_anchor() function is used for the out of the plot g
```

In [144…
```
plt.plot(Salary[0],c='green',ls='--',marker='s',ms=7,label=Players[0])
plt.plot(Salary[1],c='Blue',ls='--',marker='o',ms=6,label =Players[1])
plt.plot(Salary[2],c='purple',ls='--',marker='^',ms=5,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()
```


Sachin
Rahul
Smith
Sami
1e7
3.0
2.5
2.0
1.5
1.0
0.5
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024

In [ ]: