

In [2]:

```
#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","Kohli"]
Pdict = {"Sachin":0,"Rahul":1,"Smith":2,"Sami":3,"Pollard":4,"Morris":5,"Samson":6,"Kohli":7}

#Salaries
Sachin_Salary = [15946875,17718750,19490625,21262500,23034375,24806250,25244493,26015425,26786250]
Rahul_Salary = [12000000,12744189,13488377,14232567,14976754,16324500,18038573,188091770,196022500]
Smith_Salary = [4621800,5828090,13041250,14410581,15779912,14500000,16022500,17500000,18326275]
Sami_Salary = [3713640,4694041,13041250,14410581,15779912,17149243,18518574,19450000,20326275,21202500]
Pollard_Salary = [4493160,4806720,6061274,13758000,15202590,16647180,18091770,19000000,20026275,21022500]
Morris_Salary = [3348000,4235220,12455000,14410581,15779912,14500000,16022500,17000000,18026275,19022500]
Samson_Salary = [3144240,3380160,3615960,4574189,13520500,14940153,16359805,17770000,18726275,19722500]
Dhoni_Salary = [0,0,4171200,4484040,4796880,6053663,15506632,16669630,17832627,18862875]
Kohli_Salary = [0,0,0,4822800,5184480,5546160,6993708,16402500,17632688,18862875]
Sky_Salary = [3031920,3841443,13041250,14410581,15779912,14200000,15691000,17182000,18026275,19022500]
#Matrix
Salary = np.array([Sachin_Salary, Rahul_Salary, Smith_Salary, Sami_Salary, Pollard_Salary, Morris_Salary, Samson_Salary, Dhoni_Salary, Kohli_Salary, Sky_Salary])

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

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

In [3]: Salary

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

In [4]: Games

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

In [5]: Salary/Games

```
C:\Users\Ravi\AppData\Local\Temp\ipykernel_31620\3709746658.py:1: RuntimeWarning:
divide by zero encountered in divide
Salary/Games
```

```
Out[5]: 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.        ,      0.        ,
  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 [6]: `np.round(Salary/Games)`

```
C:\Users\Ravi\AppData\Local\Temp\ipykernel_31620\3232172828.py:1: RuntimeWarning:
divide by zero encountered in divide
np.round(Salary/Games)
```

```
Out[6]: array([[ 199336.,  230114.,  237691.,  259299.,  315539.,  302515.,
   435250.,  357040.,  5075634.,  671429.],
   [ 146341.,  223582.,  164492.,  180159.,  197063.,  226729.,
   300643.,  274342.,  271731.,  289760.],
   [ 58504.,  74719.,  173883.,  177908.,  207630.,  183544.,
   258427.,  230855.,  247630.,  299194.],
   [ 46420.,  72216.,  169367.,  218342.,  228694.,  222717.,
   336701.,  290299.,  291006.,  561450.],
   [ 54795.,  58619.,  73918.,  174152.,  185397.,  213425.,
   335033.,  257057.,  288918.,  522836.],
   [ 47829.,  61380.,  185896.,  187150.,  225427.,  188312.,
   281096.,  237095.,  241361.,  469191.],
   [ 40311.,  52815.,  45200.,  58643.,  300456.,  186752.,
   272663.,  253992.,  301104.,  244739.],
   [ 0.,  0.,  52140.,  60595.,  58499.,  77611.,
   234949.,  205798.,  220156.,  703542.],
   [ 0.,  0.,  0.,  59541.,  66468.,  68471.,
   179326.,  inf,  1763269.,  369860.],
   [ 40426.,  75322.,  255711.,  182412.,  204934.,  186842.,
   320224.,  249014.,  345796.,  241935.]])
```

In [7]: Games[5]

```
Out[7]: array([70, 69, 67, 77, 70, 77, 57, 74, 79, 44])
```

In [8]: Pdict

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

In [9]: Sdict

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

In [10]: `import warnings
warnings.filterwarnings('ignore')`

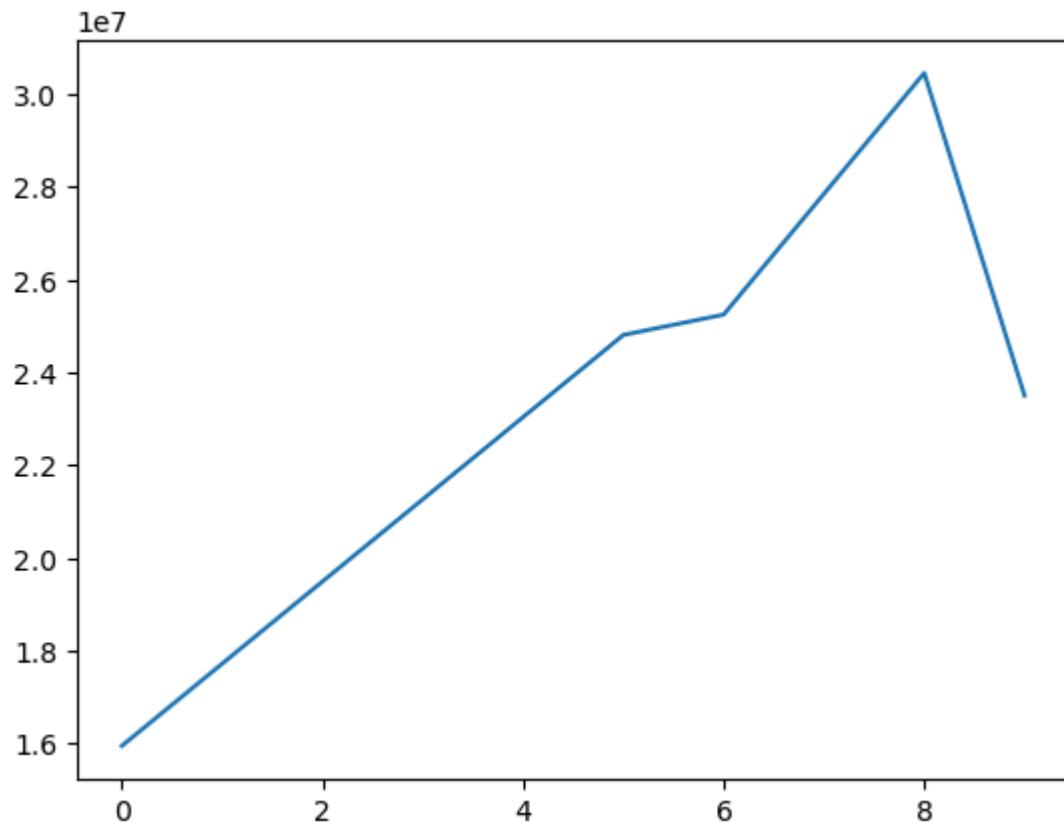
In [11]: `import matplotlib.pyplot as plt`

In [12]: Salary[0]

```
Out[12]: array([15946875, 17718750, 19490625, 21262500, 23034375, 24806250,  
   25244493, 27849149, 30453805, 23500000])
```

```
In [13]: plt.plot(Salary[0])
```

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

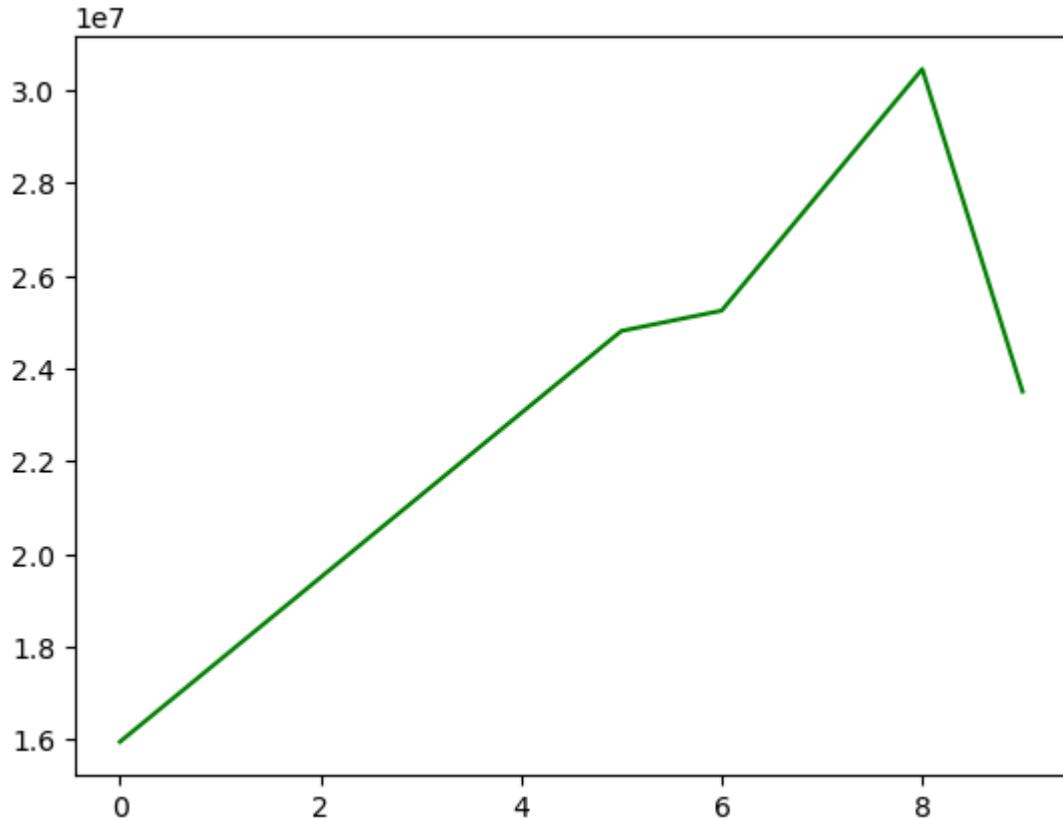


```
In [14]: Games[0]
```

```
Out[14]: array([80, 77, 82, 82, 73, 82, 58, 78, 6, 35])
```

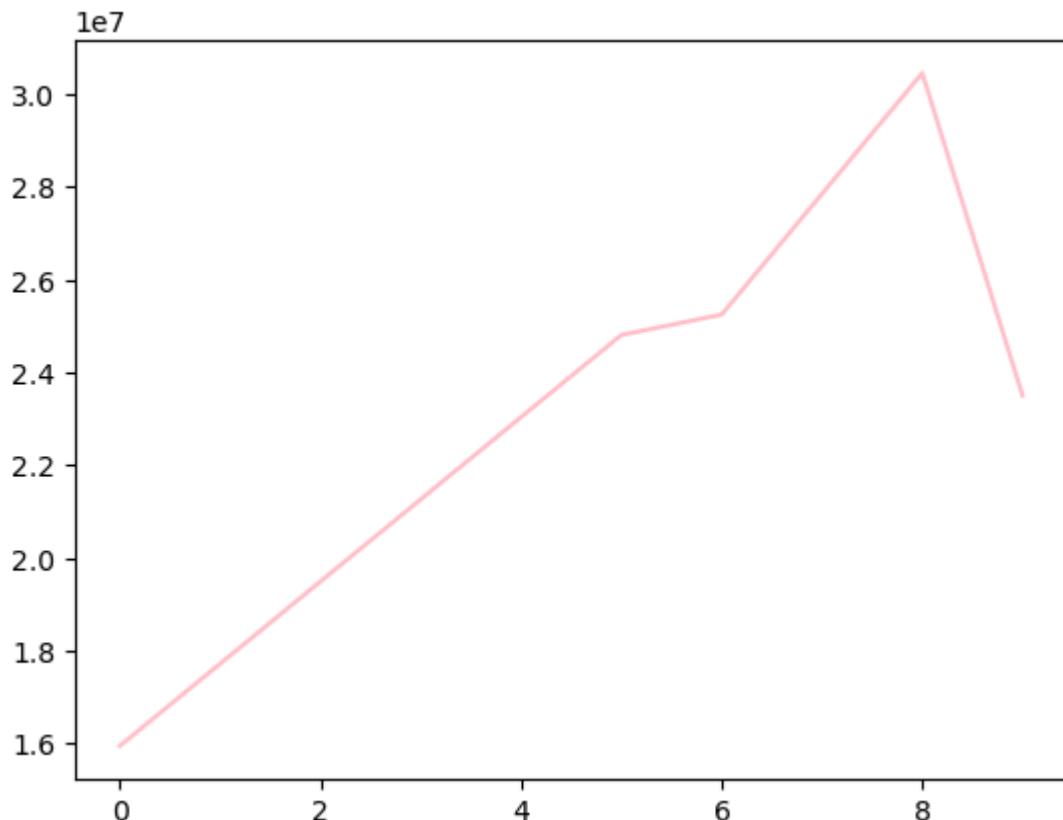
```
In [16]: plt.plot(Salary[0], color='green')
```

```
Out[16]: [
```



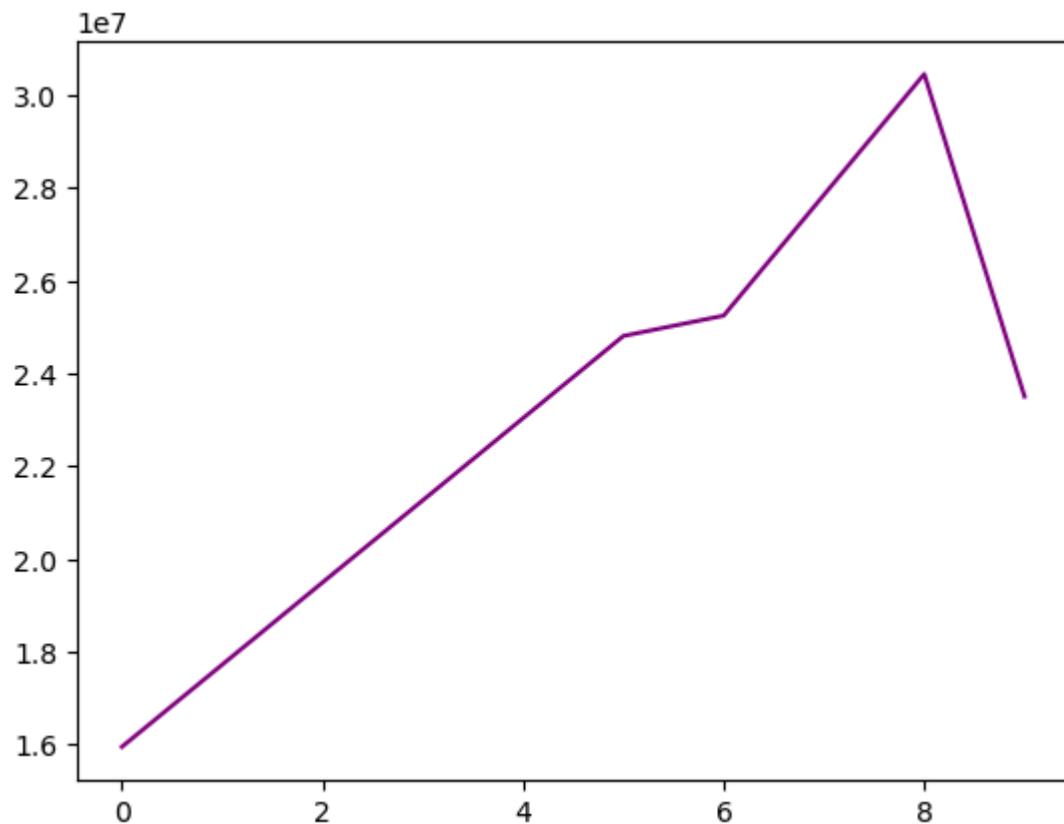
```
In [17]: plt.plot(Salary[0],color='pink')
```

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



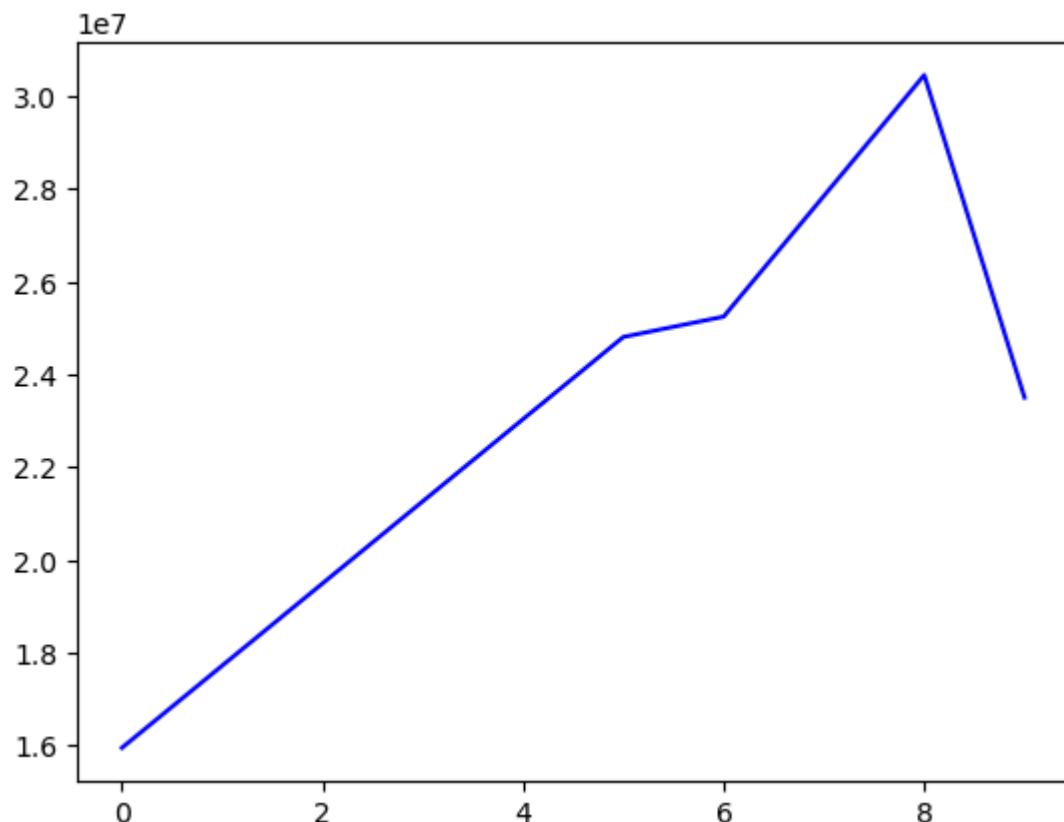
```
In [18]: plt.plot(Salary[0],color='purple')
```

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



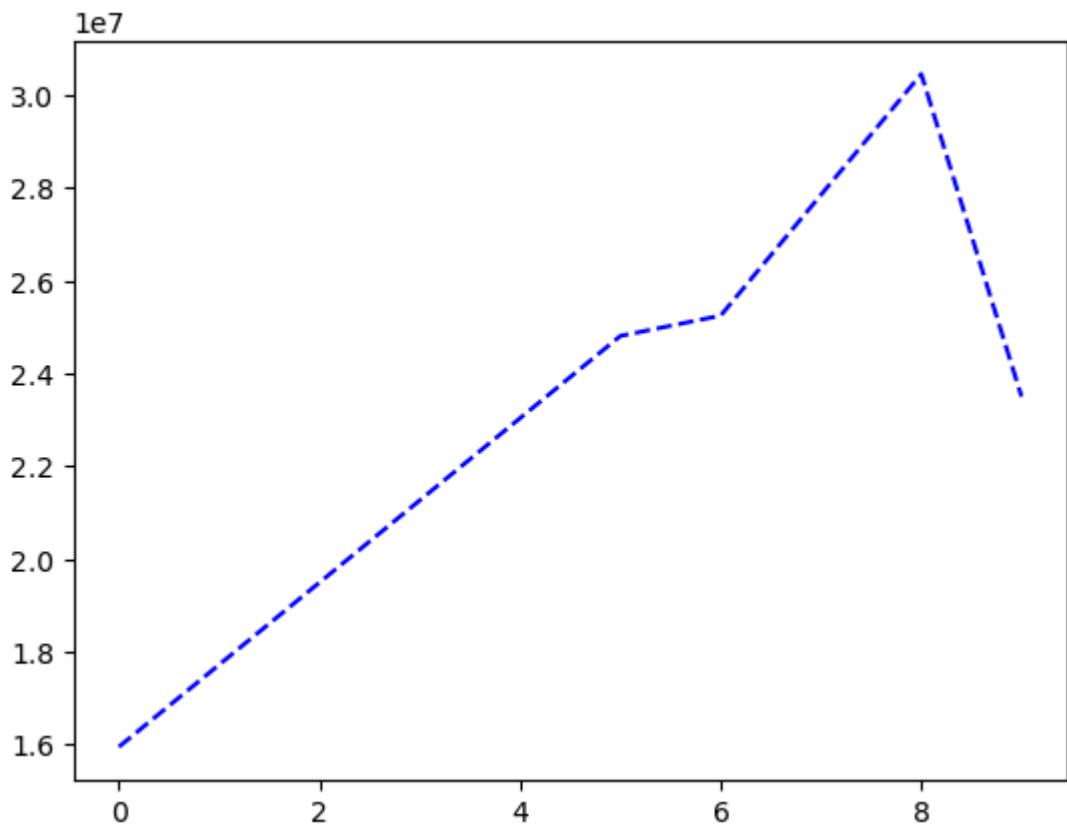
```
In [20]: plt.plot(Salary[0], c='b')
```

```
Out[20]: [
```



```
In [21]: plt.plot(Salary[0], c='b', ls = '--')
```

```
Out[21]: [
```



```
In [23]: plt.plot(Salary[0],c='b', ls =';') #valid value for ls; supported values are '-'
```

```

-----
ValueError                                Traceback (most recent call last)
Cell In[23], line 1
----> 1 plt.plot(Salary[0],c=    , ls =   ) #valid value for ls; supported values
      are '-', '--', '-.', ':', 'None', ' ', '', 'solid', 'dashed', 'dashdot', 'dotted'

File C:\ProgramData\anaconda3\Lib\site-packages\matplotlib\pyplot.py:3838, in plo
t(scalex, scaley, data, *args, **kwargs)
    3830 @_copy_docstring_and_deprecators(Axes.plot)
    3831 def plot(
    3832     *args: float | ArrayLike | str,
    (...) 3836     **kwargs,
    3837 ) -> list[Line2D]:
-> 3838     return gca().plot(
    3839         *args,
    3840         scalex=scalex,
    3841         scaley=scaley,
    3842         **({      : data} if data is not None else {}),
    3843         **kwargs,
    3844     )

File C:\ProgramData\anaconda3\Lib\site-packages\matplotlib\axes\_axes.py:1777, in
Axes.plot(self, scalex, scaley, data, *args, **kwargs)
1534 """
1535 Plot y versus x as lines and/or markers.
1536
(...). 1774 (``'green'``) or hex strings (``'#008000'``).
1775 """
1776 kwargs = cbook.normalize_kwargs(kwargs, mlines.Line2D)
-> 1777 lines = [*self._get_lines(self, *args, data=data, **kwargs)]
1778 for line in lines:
1779     self.add_line(line)

File C:\ProgramData\anaconda3\Lib\site-packages\matplotlib\axes\_base.py:297, in
_process_plot_var_args.__call__(self, axes, data, return_kw_args, *args, **kwargs)
295     this += args[0],
296     args = args[1:]
--> 297 yield from self._plot_args(
    298     axes, this, kwargs, ambiguous_fmt_datakey=ambiguous_fmt_datakey,
    299     return_kw_args=return_kw_args
300 )

File C:\ProgramData\anaconda3\Lib\site-packages\matplotlib\axes\_base.py:546, in
_process_plot_var_args._plot_args(self, axes, tup, kwargs, return_kw_args, ambiguou
us_fmt_datakey)
544     return list(result)
545 else:
--> 546     return [l[0] for l in result]

File C:\ProgramData\anaconda3\Lib\site-packages\matplotlib\axes\_base.py:539, in
<genexpr>(.0)
534 else:
535     raise ValueError(
536         f"label must be scalar or have the same length as the input "
537         f"data, but found {len(label)} for {n_datasets} datasets.")
--> 539 result = (make_artist(axes, x[:, j % ncx], y[:, j % ncy], kw,
    540         **kwargs,          : label))
    541         for j, label in enumerate(labels))
543 if return_kw_args:
544     return list(result)

```

```

File C:\ProgramData\anaconda3\Lib\site-packages\matplotlib\axes\_base.py:338, in
_process_plot_var_args._make_line(self, axes, x, y, kw, kwargs)
 336 kw = {**kw, **kwargs} # Don't modify the original kw.
 337 self._setdefaults(self._getdefaults(kw), kw)
--> 338 seg = mlines.Line2D(x, y, **kw)
 339 return seg, kw

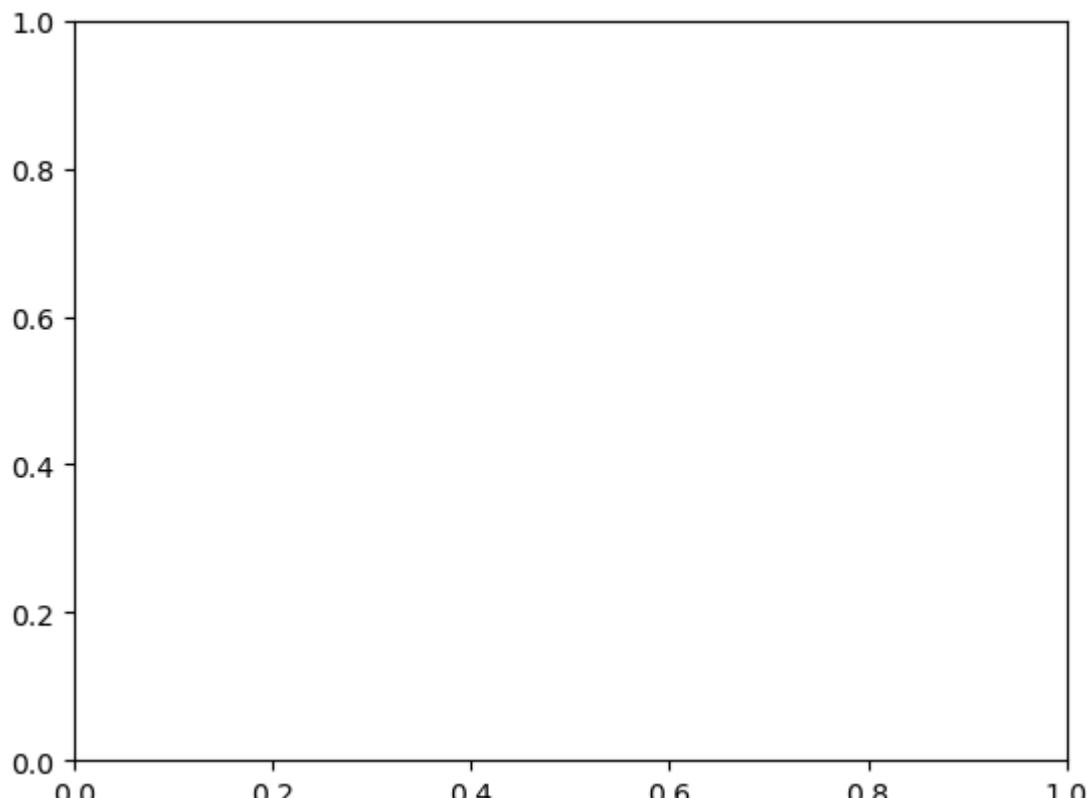
File C:\ProgramData\anaconda3\Lib\site-packages\matplotlib\lines.py:386, in Line2D.__init__(self, xdata, ydata, linewidth, linestyle, color, gapcolor, marker, markersize, markeredgewidth, markeredgecolor, markerfacecolor, markerfacecoloralt, fillstyle, antialiased, dash_capstyle, solid_capstyle, dash_joinstyle, solid_joinsstyle, pickradius, drawstyle, markevery, **kwargs)
 383 self._dash_pattern = (0, None) # offset, dash (scaled by linewidth)
 385 self.set_linewidth(linewidth)
--> 386 self.set_linestyle(linestyle)
 387 self.set_drawstyle(drawstyle)
 389 self._color = None

File C:\ProgramData\anaconda3\Lib\site-packages\matplotlib\lines.py:1192, in Line2D.set_linestyle(self, ls)
 1190 if ls in [' ', '', 'none']:
 1191     ls = 'None'
-> 1192 _api.check_in_list([*self._lineStyles, *ls_mapper_r], ls=ls)
 1193 if ls not in self._lineStyles:
 1194     ls = ls_mapper_r[ls]

File C:\ProgramData\anaconda3\Lib\site-packages\matplotlib\_api\__init__.py:130,
in check_in_list(values, _print_supported_values, **kwargs)
 128 if _print_supported_values:
 129     msg += f"; supported values are {', '.join(map(repr, values))}"
--> 130 raise ValueError(msg)

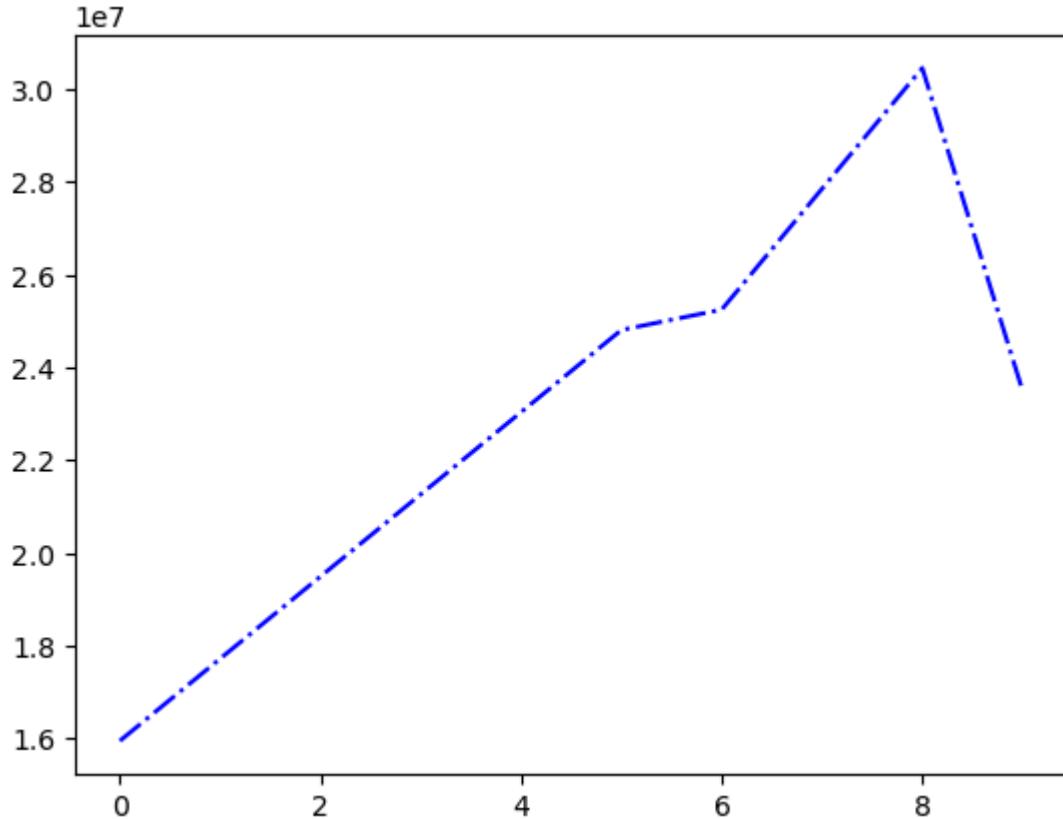
ValueError: ';' is not a valid value for ls; supported values are '-', '--',
'-.', ':', 'None', ' ', '', 'solid', 'dashed', 'dashdot', 'dotted'

```



```
In [24]: plt.plot(Salary[0],c='b', ls ='-.' )
```

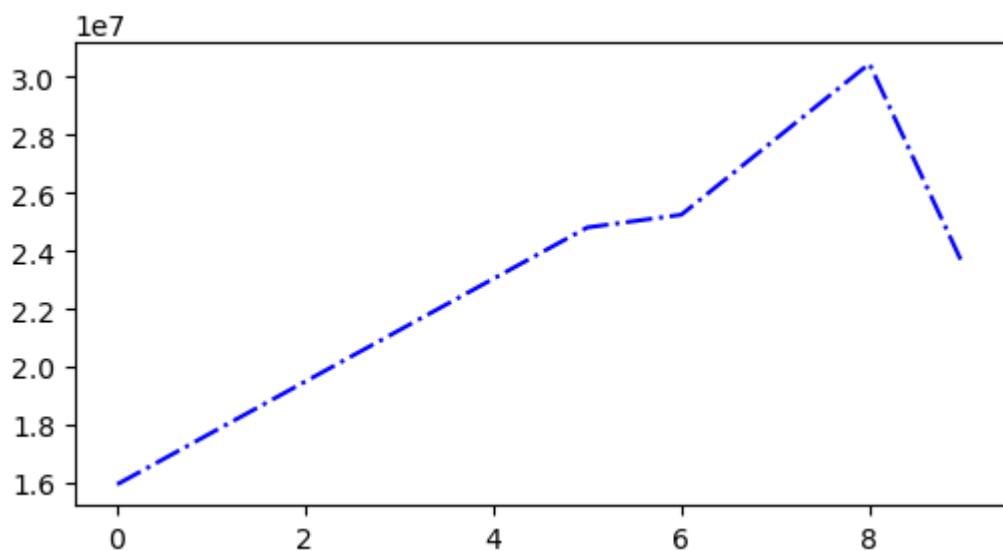
```
Out[24]: [
```



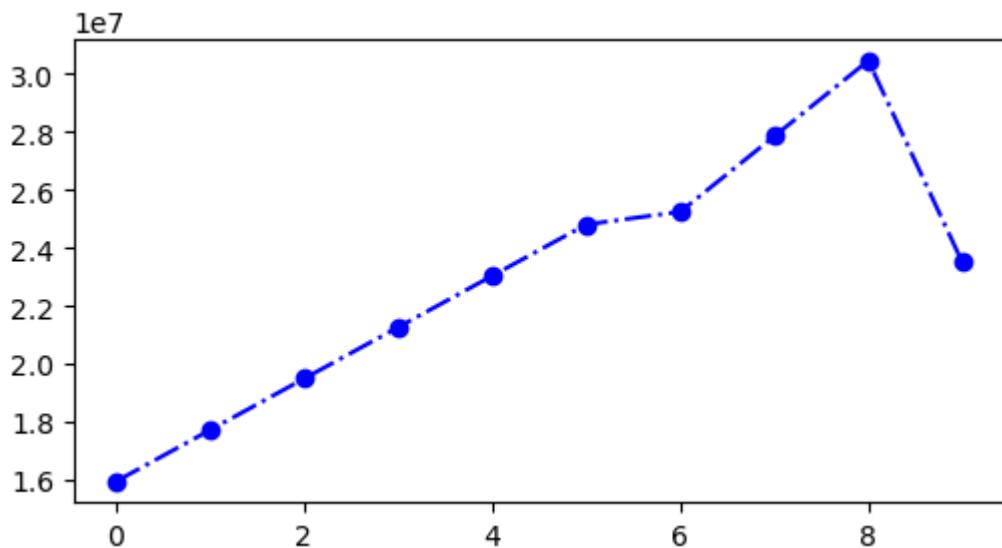
```
In [29]: %matplotlib inline  
plt.rcParams['figure.figsize']= 6,3
```

```
In [ ]:
```

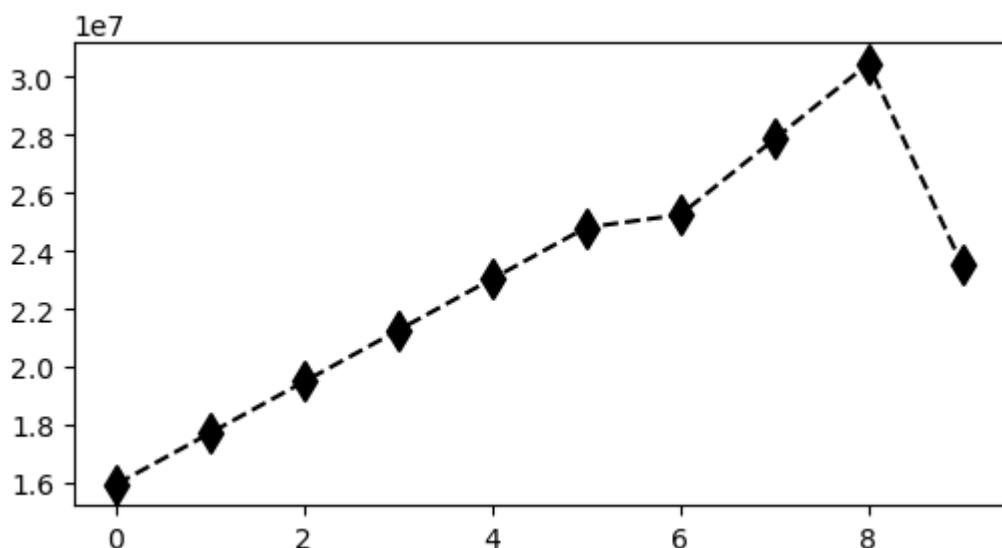
```
In [30]: plt.plot(Salary[0],c='b', ls ='-.' )  
plt.show()
```



```
In [32]: plt.plot(Salary[0],c='b', ls ='-.', marker ='o')  
plt.show()
```



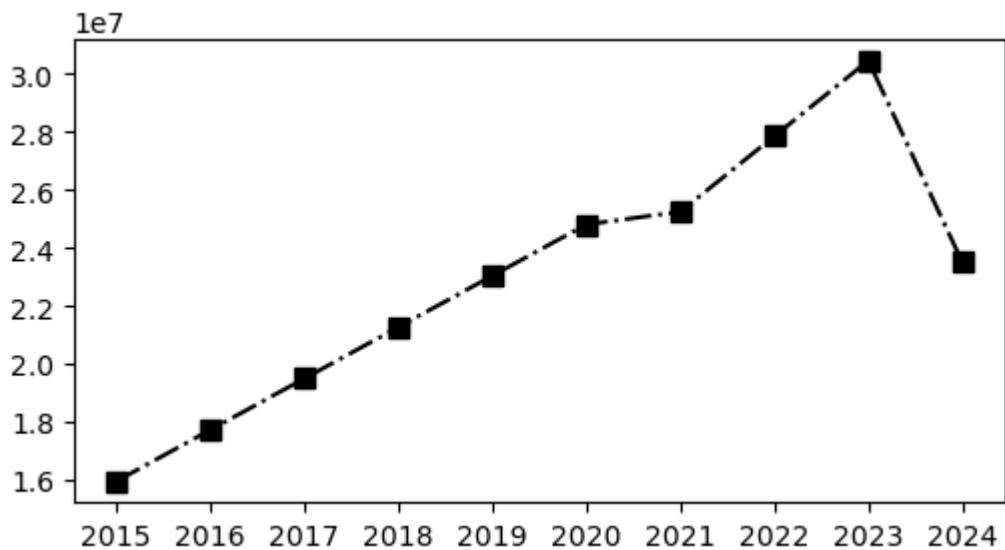
```
In [33]: plt.plot(Salary[0], c='k', ls ='-.', marker ='s', ms=10)
plt.show()
```



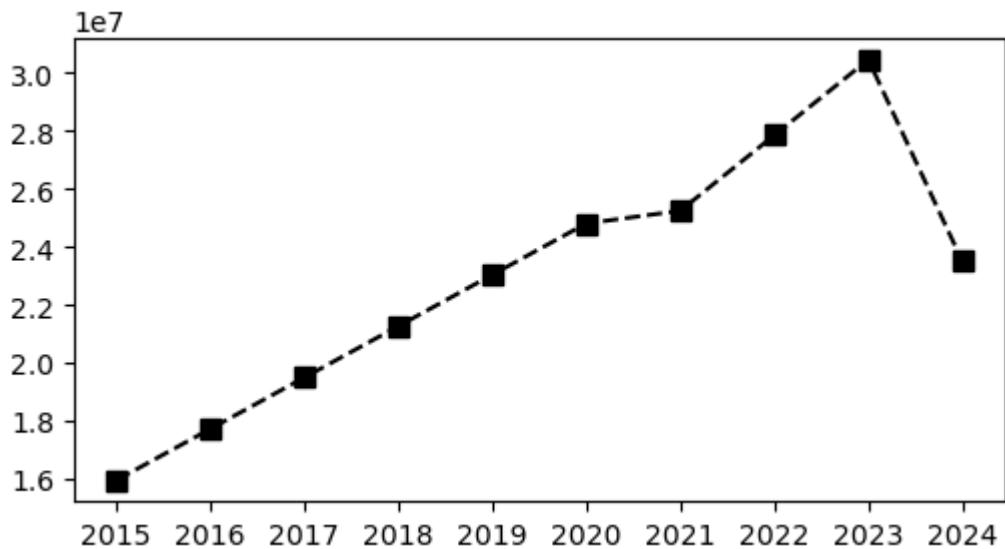
```
In [34]: Sdict
```

```
Out[34]: {'2015': 0,
          '2016': 1,
          '2017': 2,
          '2018': 3,
          '2019': 4,
          '2020': 5,
          '2021': 6,
          '2022': 7,
          '2023': 8,
          '2024': 9}
```

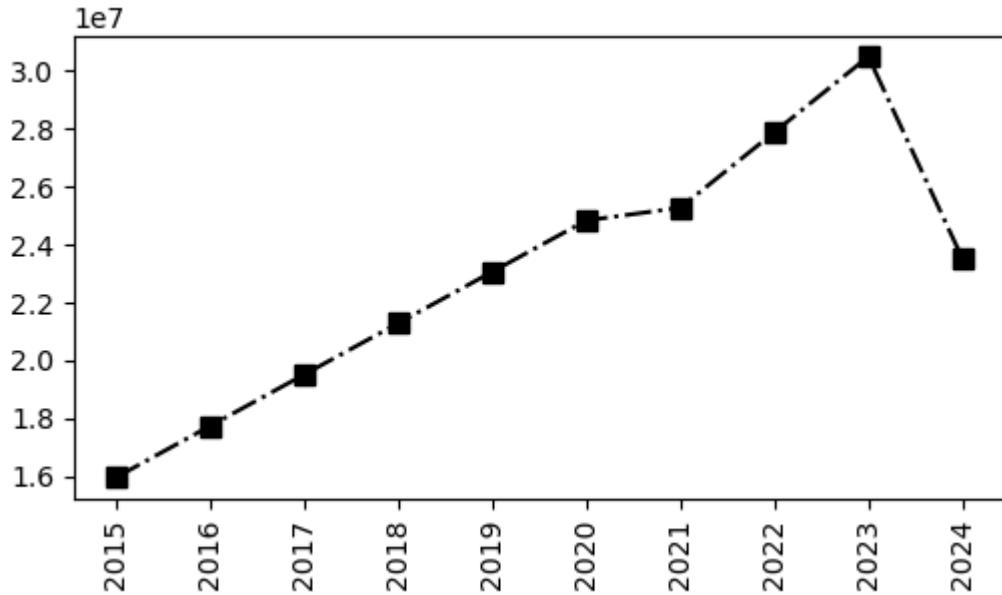
```
In [36]: plt.plot(Salary[0], c='k', ls ='-.', marker ='s', ms=7)
plt.xticks(list(range(0,10)),Seasons)
plt.show()
```



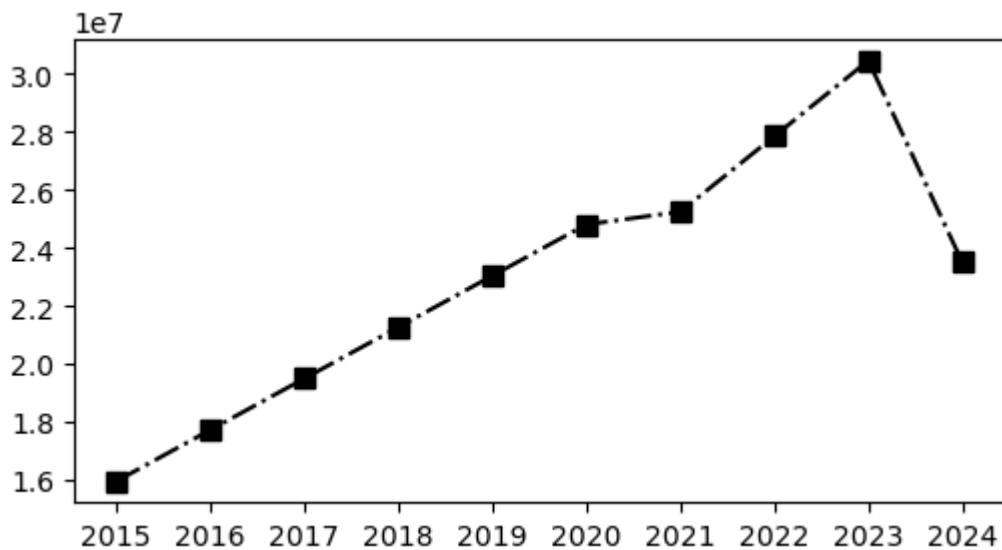
```
In [37]: plt.plot(Salary[0],c='k', ls ='-.',marker ='s',ms=7)  
plt.xticks(list(range(0,10)),Seasons)  
plt.show()
```



```
In [38]: plt.plot(Salary[0],c='k', ls ='-.',marker ='s',ms=7)  
plt.xticks(list(range(0,10)),Seasons,rotation='vertical')  
plt.show()
```



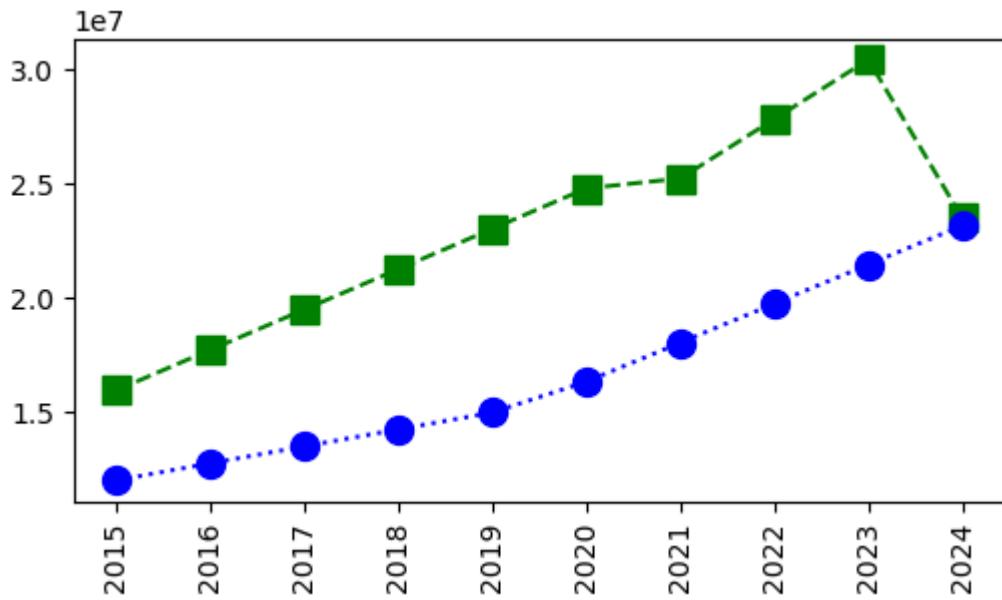
```
In [44]: plt.plot(Salary[0], c='k', ls ='-.', marker = 's', ms=7)
plt.xticks(list(range(0,10)), Seasons, rotation='horizontal')
plt.show()
```



```
In [45]: plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 10, label = Players[0])
plt.plot(Salary[1], c='Blue', ls = ':', marker = 'o', ms = 10, label = Players[1])

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

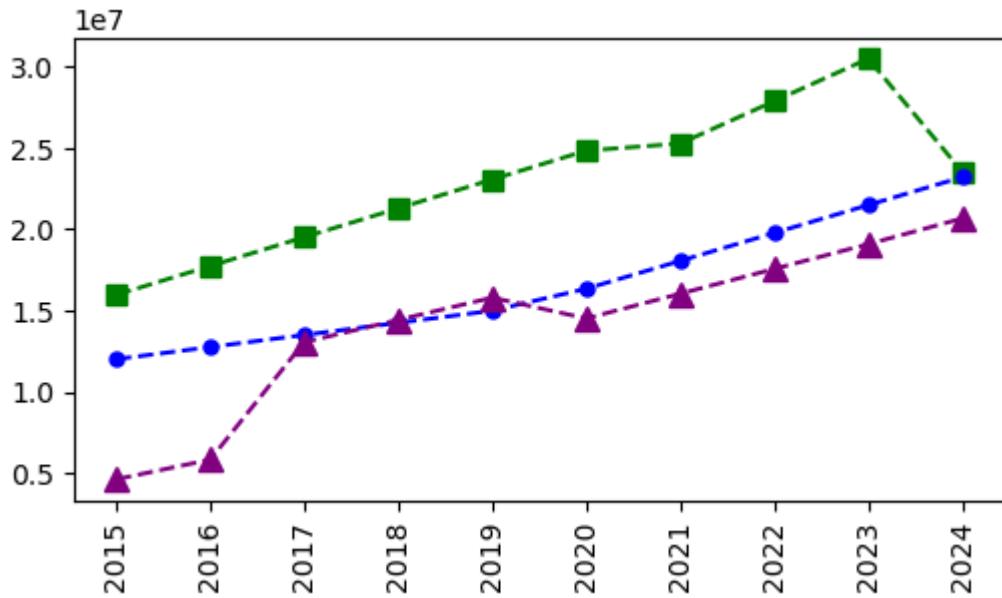
plt.show()
```



```
In [46]: plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Players[0])
plt.plot(Salary[1], c='Blue', ls = '--', marker = 'o', ms = 5, label = Players[1])
plt.plot(Salary[2], c='purple', ls = '--', marker = '^', ms = 8, label = Players[2])

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

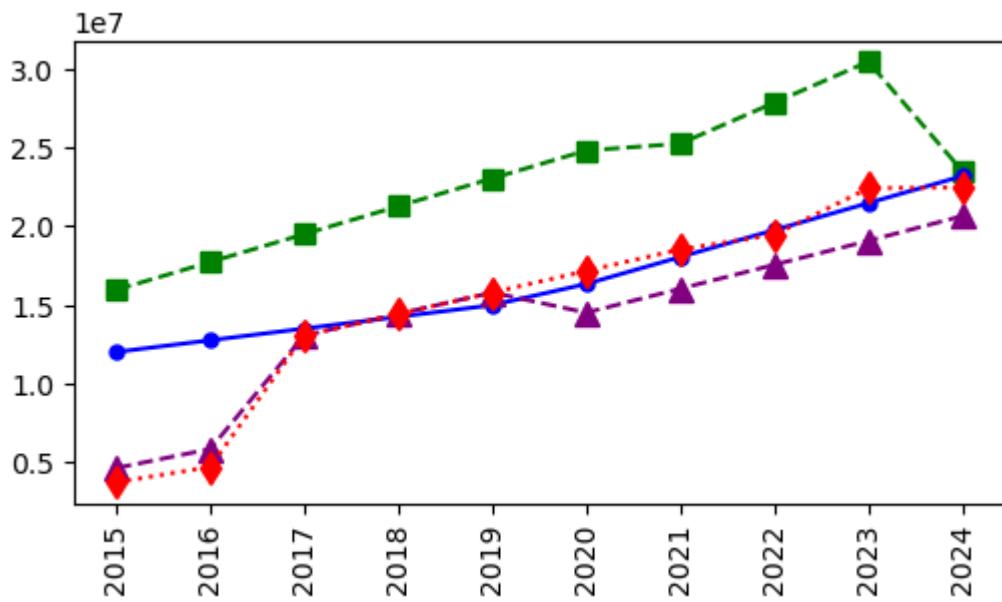
plt.show()
```



```
In [47]: plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Players[0])
plt.plot(Salary[1], c='Blue', ls = '--', marker = 'o', ms = 5, label = Players[1])
plt.plot(Salary[2], c='purple', ls = '--', marker = '^', ms = 8, label = Players[2])
plt.plot(Salary[3], c='Red', ls = ':', marker = 'd', ms = 8, label = Players[3])

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

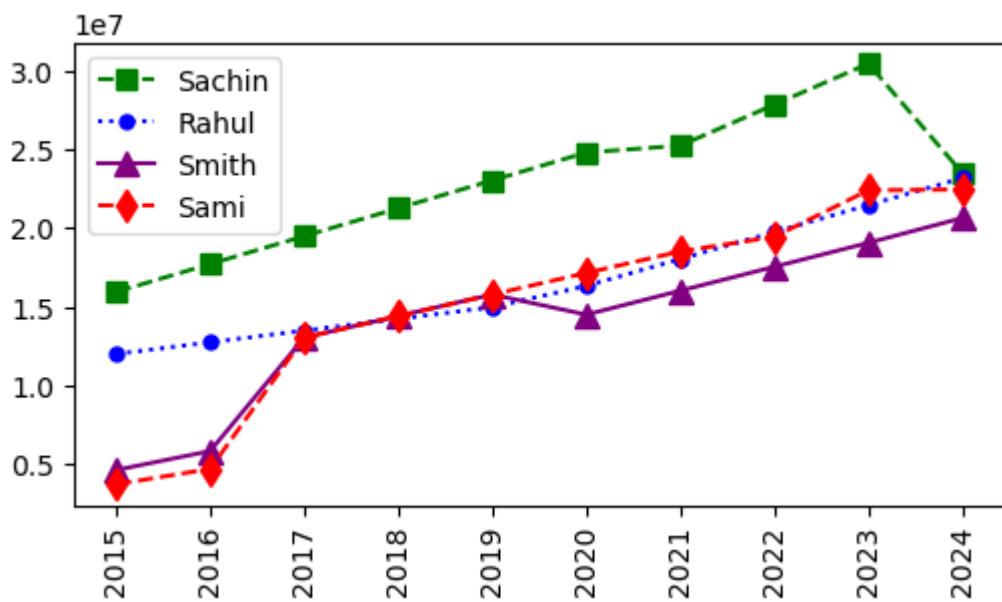
plt.show()
```



```
In [48]: # how to add Legend in visualisation
```

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

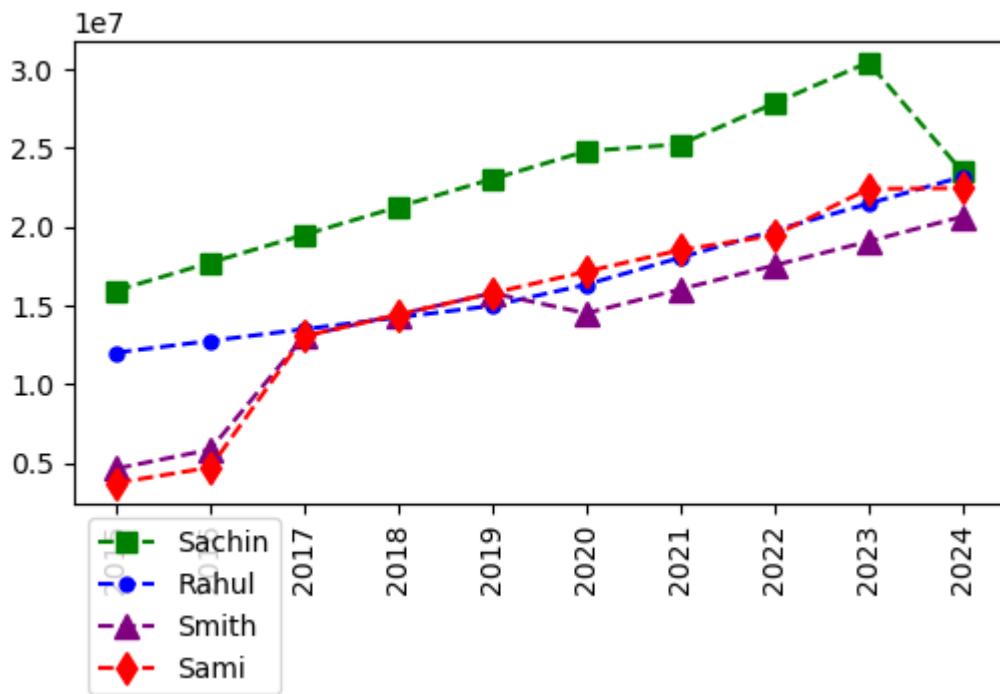
plt.show()
```



```
In [49]:
```

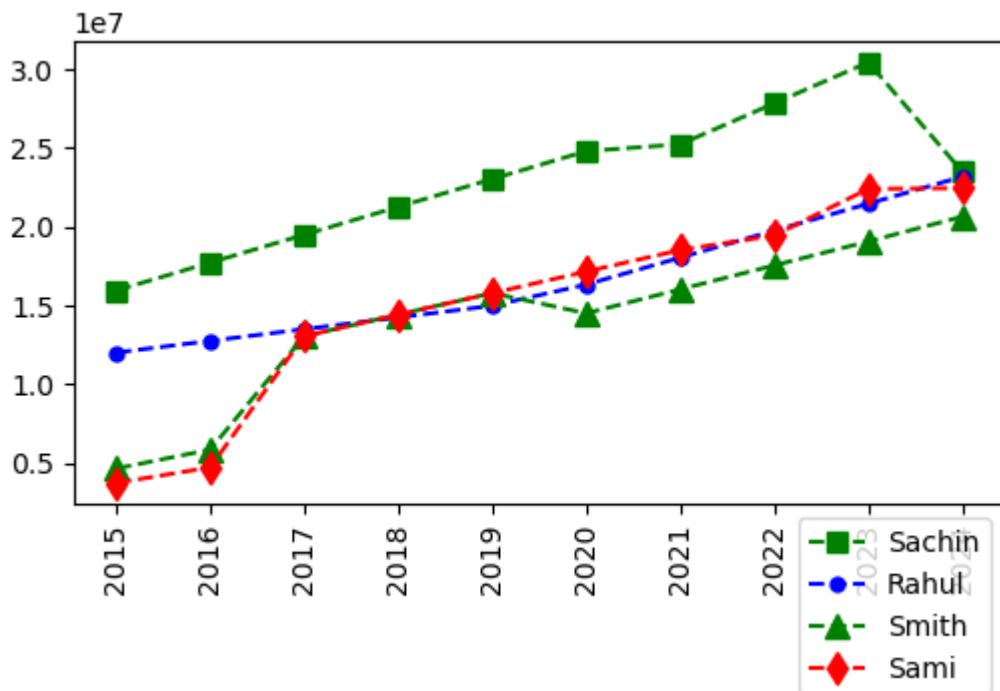
```
plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Players[0])
plt.plot(Salary[1], c='Blue', ls = '--', marker = 'o', ms = 5, label = Players[1])
plt.plot(Salary[2], c='purple', ls = '-.', marker = '^', ms = 8, label = Players[2])
plt.plot(Salary[3], c='Red', ls = '---', marker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

plt.show()
```



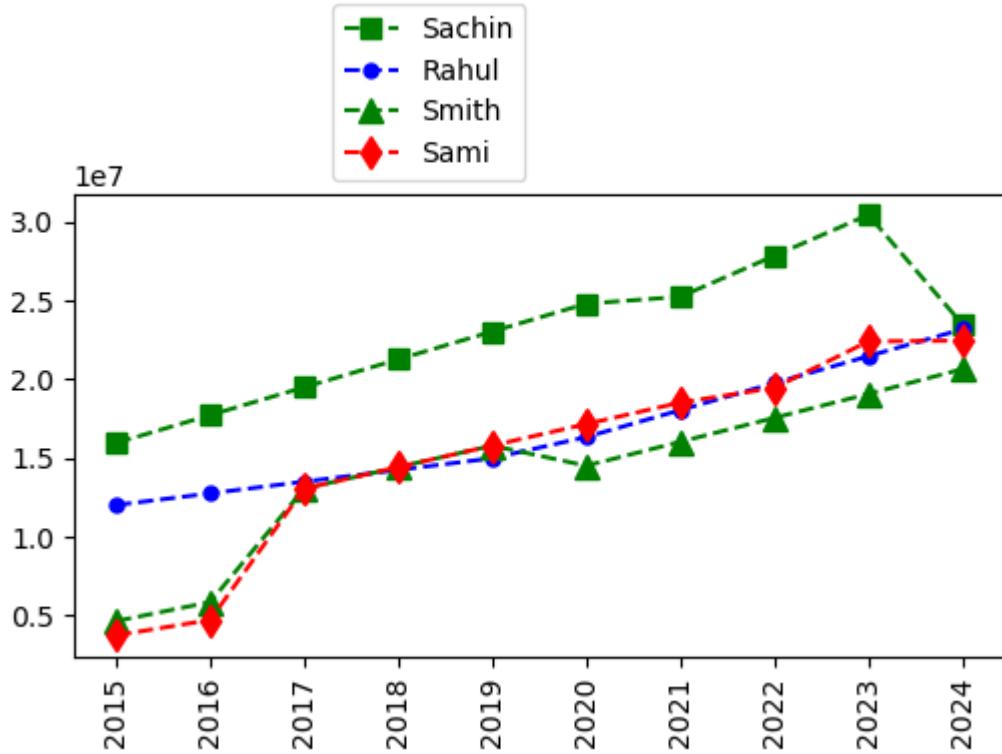
```
In [50]: plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Players[0])
plt.plot(Salary[1], c='Blue', ls = '--', marker = 'o', ms = 5, label = Players[1]
plt.plot(Salary[2], c='Green', ls = '--', marker = '^', ms = 8, label = Players[2]
plt.plot(Salary[3], c='Red', ls = '--', marker = 'd', ms = 8, label = Players[3]
plt.legend(loc = 'upper right',bbox_to_anchor=(1,0) )
plt.xticks(list(range(0,10)), Seasons,rotation='vertical')

plt.show()
```



```
In [51]: plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Players[0])
plt.plot(Salary[1], c='Blue', ls = '--', marker = 'o', ms = 5, label = Players[1]
plt.plot(Salary[2], c='Green', ls = '--', marker = '^', ms = 8, label = Players[2]
plt.plot(Salary[3], c='Red', ls = '--', marker = 'd', ms = 8, label = Players[3]
plt.legend(loc = 'lower right',bbox_to_anchor=(0.5,1) )
plt.xticks(list(range(0,10)), Seasons,rotation='vertical')
```

```
plt.show()
```



```
In [53]: plt.plot(Salary[0], c='Green', ls = '--', marker = 's', ms = 7, label = Players[0]
plt.plot(Salary[1], c='Blue', ls = '--', marker = 'o', ms = 7, label = Players[1]
plt.plot(Salary[2], c='Green', ls = '--', marker = '^', ms = 7, label = Players[2]
plt.plot(Salary[3], c='Purple', ls = '--', marker = 'D', ms = 7, label = Players[3]
plt.plot(Salary[4], c='Black', ls = '--', marker = 's', ms = 7, label = Players[4]
plt.plot(Salary[5], c='Red', ls = '--', marker = 'o', ms = 7, label = Players[5]
plt.plot(Salary[6], c='Red', ls = '--', marker = '^', ms = 7, label = Players[6]
plt.plot(Salary[7], c='Red', ls = '--', marker = 'd', ms = 7, label = Players[7]
plt.plot(Salary[8], c='Red', ls = '--', marker = 's', ms = 7, label = Players[8]
plt.plot(Salary[9], c='Red', ls = '--', marker = 'o', ms = 7, label = Players[9]

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

plt.show()
```

```

-----
ValueError                                Traceback (most recent call last)
Cell In[53], line 12
    9 plt.plot(Salary[8], c='Red', ls = '--', marker = 's', ms = 7, label = Players[8])
   10 plt.plot(Salary[9], c='Red', ls = '--', marker = 'o', ms = 7, label = Players[9])
--> 12 plt.legend(loc = ,bbox_to_anchor=(0.5,1))
   13 plt.xticks(list(range(0,10)), Seasons, rotation='vertical')
   15 plt.show()

File C:\ProgramData\anaconda3\Lib\site-packages\matplotlib\pyplot.py:3628, in legend(*args, **kwargs)
3626 @_copy_docstring_and_deprecators(Axes.legend)
3627 def legend(*args, **kwargs) -> Legend:
-> 3628     return gca().legend(*args, **kwargs)

File C:\ProgramData\anaconda3\Lib\site-packages\matplotlib\axes\_axes.py:337, in Axes.legend(self, *args, **kwargs)
220 """
221 Place a legend on the Axes.
222
(...). 334 .. plot:: gallery/text_labels_and_annotations/legend.py
335 """
336 handles, labels, kwargs = mlegend._parse_legend_args([self], *args, **kwargs)
--> 337 self.legend_ = mlegend.Legend(self, handles, labels, **kwargs)
  338 self.legend_.remove_method = self._remove_legend
  339 return self.legend_

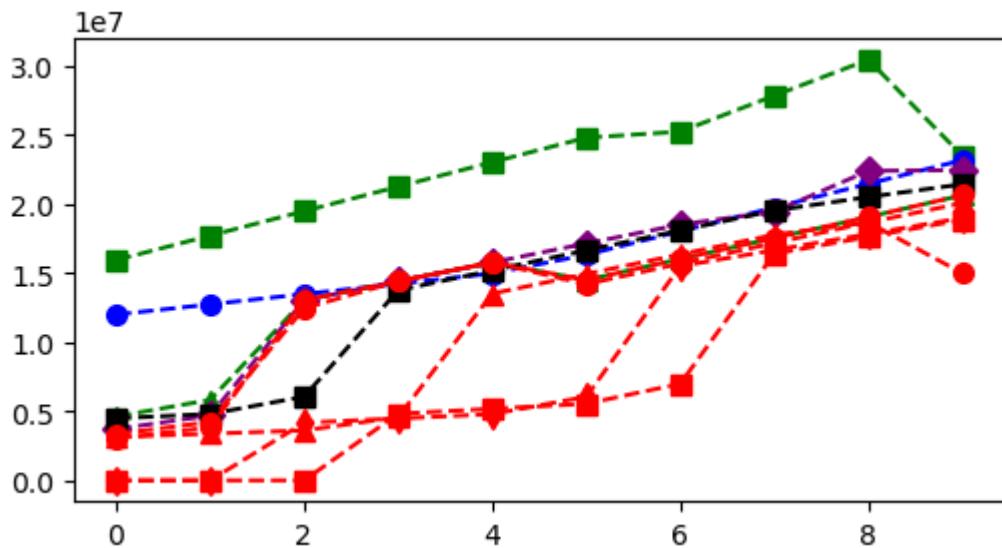
File C:\ProgramData\anaconda3\Lib\site-packages\matplotlib\legend.py:553, in Legend.__init__(self, parent, handles, labels, loc, numpoints, markerscale, markerfirst, reverse, scatterpoints, scatteryoffsets, prop, fontsize, labelcolor, borderpad, labelspacing, handlelength, handleheight, handletextpad, borderaxespad, columnspacing, ncols, mode, fancybox, shadow, title, title_fontsize, framealpha, edgecolor, facecolor, bbox_to_anchor, bbox_transform, frameon, handler_map, title_fontproperties, alignment, ncol, draggable)
550 self._init_legend_box(handles, labels, markerfirst)
552 # Set legend location
--> 553 self.set_loc(loc)
555 # figure out title font properties:
556 if title_fontsize is not None and title_fontproperties is not None:

File C:\ProgramData\anaconda3\Lib\site-packages\matplotlib\legend.py:671, in Legend.set_loc(self, loc)
669         loc = locs[0] + ' ' + locs[1]
670     # check that loc is in acceptable strings
--> 671     loc = _api.check_getitem(self.codes, loc=loc)
672 elif np.iterable(loc):
673     # coerce iterable into tuple
674     loc = tuple(loc)

File C:\ProgramData\anaconda3\Lib\site-packages\matplotlib\_api\__init__.py:184,
in check_getitem(mapping, **kwargs)
182     return mapping[v]
183 except KeyError:
--> 184     raise ValueError(
  185         f"{v}!r is not a valid value for {k}; supported values are "
  186         f"{''.join(map(repr, mapping))}" from None

```

```
ValueError: 'lower right' is not a valid value for loc; supported values are 'best', 'upper right', 'upper left', 'lower left', 'lower right', 'right', 'center left', 'center right', 'lower center', 'upper center', 'center'
```

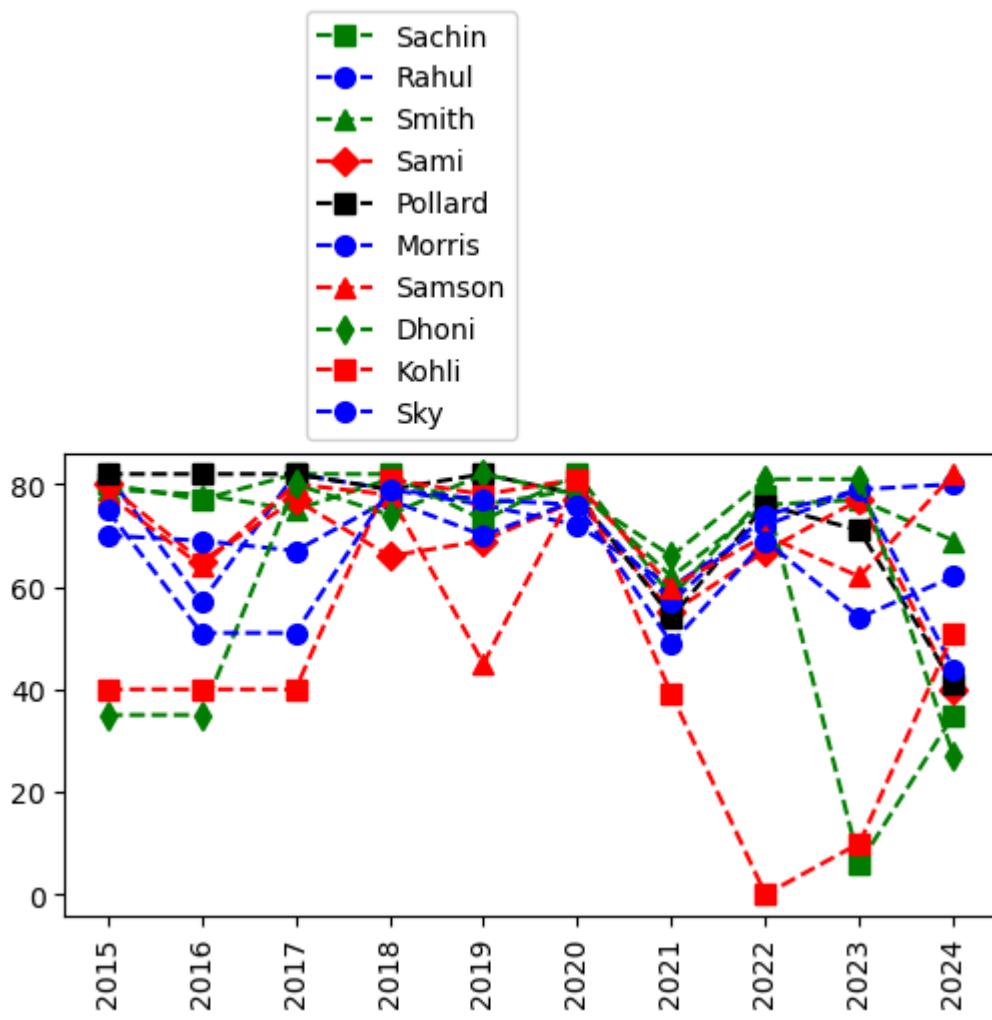


In [54]: # we can visualize the how many games played by a player

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

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

plt.show()
```



In []:

In []: