

```
import numpy as np
import pandas as pd
```

```
from google.colab import files
uploaded = files.upload()
```

datafile_02.csv

- **datafile_02.csv**(text/csv) - 988 bytes, last modified: 11/9/2022 - 100% done
Saving datafile_02.csv to datafile_02.csv

```
import io
df = pd.read_csv(io.BytesIO(uploaded['datafile_02.csv']))
print(df)
```

```
0      9100
1      7000
2      7100
3      8200
4      3200
5      9700
6      8900
7      5300
8      6800
9      8800

10     7900
11    10000
12     9500
```

```
Total Capacity in Eleventh Plan (MT) (2011-12) Proj. \
0      3145
1      6340
2     10640
3     10810
4      6420
5      7230
6      6398
7      5475
8      6050
9      6690
10     9191
11     9560
12    12220
```

```
Total Capacity in Eleventh Plan (MT) (2011-12) Ach. \
0      1635
1      5070
2      7650
3      7293
4      3100
5      7972
6      3334
7      4098
8      5097
9      4190
10     4453
11     5400
```

```

11
12
Total Capacity in Eleventh Plan (MT) (2011-12) %
0
1
2
3
4
5
6
7
8
9
10
11
12

```

```

6400
8691
5100
7900
7100
6700
4800
11000
5200
7400
8400
6200
4800
6600
7100

```

```

print(df.columns)
df.head()

```

```

Index(['Port', 'Traffic in Eleventh Plan (MT) (2011-12)Proj.',
      'Traffic in Eleventh Plan (MT) (2011-12) Ach.',
      'Traffic in Eleventh Plan (MT) (2011-12) %',
      'Total Capacity in Eleventh Plan (MT) (2011-12) Proj.',
      'Total Capacity in Eleventh Plan (MT) (2011-12) Ach.',
      'Total Capacity in Eleventh Plan (MT) (2011-12) %'],
      dtype='object')

```

	Port	Traffic in Eleventh Plan (MT) (2011- 12)Proj.	Traffic in Eleventh Plan (MT) (2011- 12) Ach.	Traffic in Eleventh Plan (MT) (2011- 12) %	Total Capacity in Eleventh Plan (MT) (2011-12) Proj.	Total Capacity in Eleventh Plan (MT) (2011-12) Ach.	Total Capacity in Eleventh Plan (MT) (2011-12) %
0	Kolkata	1343	1223	9100	3145	1635	5100
1	Haldia	4450	3101	7000	6340	5070	7900
2	Paradeep	7640	5425	7100	10640	7650	7100
3	Visakhapatnam	8220	6742	8200	10810	7293	6700

```
# Preprocessing the dataset
```

```
# Renaming the columns
```

```
df.rename(columns = {'Traffic in Eleventh Plan (MT) (2011-12)Proj.': 'Traffic_Projected', 'Traffic in  
df
```

	Port	Traffic_Projected	Traffic_Achieved	Traffic in Eleventh Plan (MT) (2011- 12) %	Total_Capacity_P
0	Kolkata	1343	1223	9100	
1	Haldia	4450	3101	7000	
2	Paradeep	7640	5425	7100	
3	Visakhapatnam	8220	6742	8200	
4	Ennore	4700	1496	3200	
5	Chennai	5750	5571	9700	
6	Tuticorin	3172	2810	8900	

Perparing the Calculations:

```
Traffic_Percent = round((df.Traffic_Achieved/df.Traffic_Projected)*100,2)
```

```
~      ~~~~~~      ~~~~~~      ~~~~~~      ~~~~~~
```

Traffic_Percent

```
0      91.06
1      69.69
2      71.01
3      82.02
4      31.83
5      96.89
6      88.59
7      52.66
8      67.49
9      87.54
10     79.07
11     99.56
12     95.13
dtype: float64
```

```
Total_Percent = round( (df.Total_Capacity_Achieved/df.Total_Capacity_Projected)*100,2)
```

Total_Percent

```
0      51.99
1      79.97
2      71.90
3      67.47
4      48.29
5     110.26
6      52.11
7      74.85
8      84.25
9      62.63
10     48.45
```

```

11      66.95
12      71.12
dtype: float64

```

```

# Replacing the existing columns with newly created columns
df.rename(columns = {'Traffic in Eleventh Plan (MT) (2011-12) %': 'Traffic_Percent', 'Total Capacity
df.iloc[:,3:4] = Traffic_Percent
df.iloc[:,6:] = Total_Percent
df

```

	Port	Traffic_Projected	Traffic_Achieved	Traffic_Percent	Total_Cap
0	Kolkata	1343	1223	91.06	
1	Haldia	4450	3101	69.69	
2	Paradeep	7640	5425	71.01	
3	Visakhapatnam	8220	6742	82.02	
4	Ennore	4700	1496	31.83	
5	Chennai	5750	5571	96.89	
6	Tuticorin	3172	2810	88.59	
7	Cochin	3817	2010	52.66	
8	NMPT	4881	3294	67.49	
9	Mormugao	4455	3900	87.54	
10	Mumbai	7105	5618	79.07	
11	JNPT	6604	6575	99.56	
12	Kandla	8672	8250	95.13	

```
df.shape
```

```
(13, 7)
```

```
# Checking for null values
```

```
df.isnull().sum()
```

```

Port                0
Traffic_Projected    0
Traffic_Achieved     0
Traffic_Percent      0
Total_Capacity_Projected  0
Total_Capacity_Achieved  0
Total_Percent        0
dtype: int64

```

```
# Summary of Dataset
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13 entries, 0 to 12
Data columns (total 7 columns):
#   Column                                Non-Null Count  Dtype
---  ---                                -
0   Port                                13 non-null    object
1   Traffic_Projected                  13 non-null    int64
2   Traffic_Achieved                   13 non-null    int64
3   Traffic_Percent                    13 non-null    float64
4   Total_Capacity_Projected           13 non-null    int64
5   Total_Capacity_Achieved            13 non-null    int64
6   Total_Percent                      13 non-null    float64
dtypes: float64(2), int64(4), object(1)
memory usage: 856.0+ bytes
```

```
df.describe()
```

	Traffic_Projected	Traffic_Achieved	Traffic_Percent	Total_Capacity_Projected
count	13.000000	13.000000	13.000000	13.000000
mean	5446.846154	4308.846154	77.887692	7705.307692
std	2133.280019	2212.894855	19.382398	2570.246914
min	1343.000000	1223.000000	31.830000	3145.000000
25%	4450.000000	2810.000000	69.690000	6340.000000
50%	4881.000000	3900.000000	82.020000	6690.000000
75%	7105.000000	5618.000000	91.060000	9560.000000
max	8672.000000	8250.000000	99.560000	12220.000000

```
#Finding Outliers and replacing the outliers
```

```
import seaborn as sns
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = [17.50, 3.50]
plt.rcParams["figure.autolayout"] = True

sns.boxplot(x='Port', y='Traffic_Projected', data=df)
```

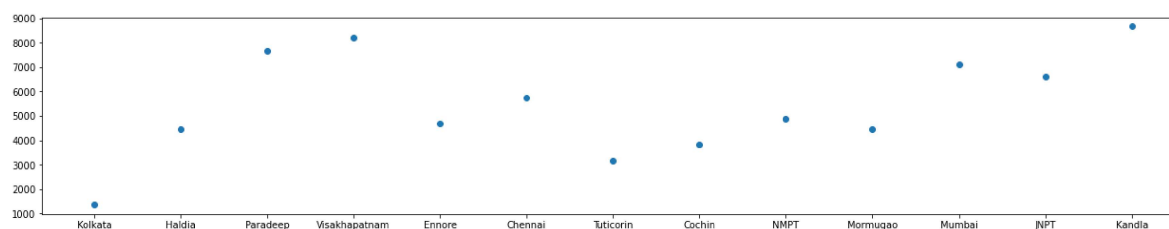
<matplotlib.axes._subplots.AxesSubplot at 0x7f7073b01bd0>

9000
8000
7000

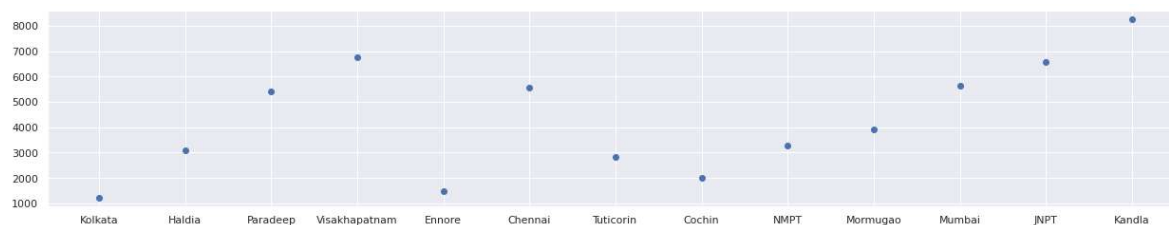
```
# Visualization using various plots
import matplotlib.pyplot as plt
import seaborn as sns
```

```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
plt.scatter(df.Port,df.Traffic_Projected)
sns.set()
```

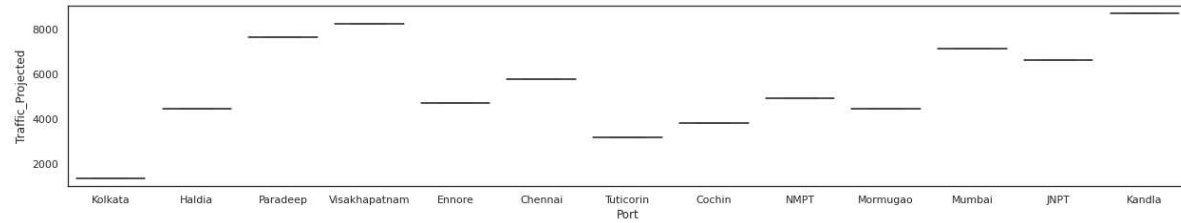


```
plt.scatter(df.Port,df.Traffic_Achieved)
sns.set_style('white')
sns.set_context('notebook')
```



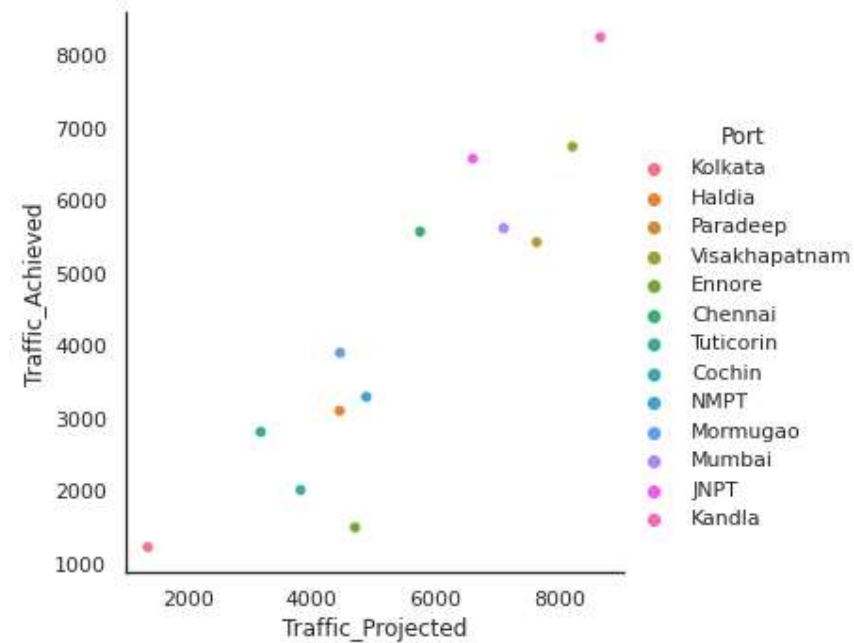
```
sns.boxplot(x='Port',y='Traffic_Projected',data=df)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f7070b68e90>



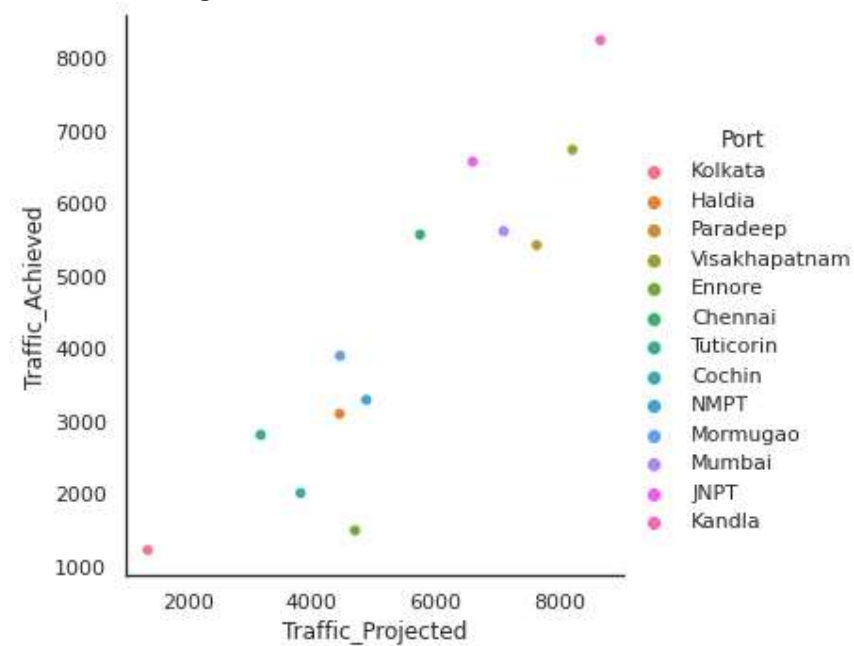
```
sns.relplot(data=df,x="Traffic_Projected",y='Traffic_Achieved',hue='Port')
```

<seaborn.axisgrid.FacetGrid at 0x7f7074358e90>



```
sns.relplot(data=df,x="Traffic_Projected",y='Traffic_Achieved',hue='Port')
```

<seaborn.axisgrid.FacetGrid at 0x7f70708e3050>



```

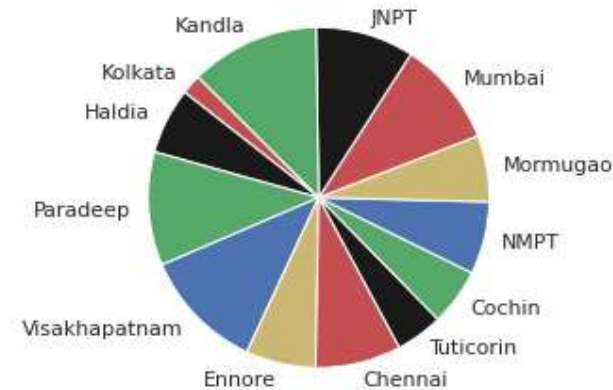
colors=['r','k','g','b','y']
plt.pie(df.Traffic_Projected,labels=df.Port,colors=colors,startangle=135)

```

```

([<matplotlib.patches.Wedge at 0x7f70707fc910>,
 <matplotlib.patches.Wedge at 0x7f7070842350>,
 <matplotlib.patches.Wedge at 0x7f7070805310>,
 <matplotlib.patches.Wedge at 0x7f70708053d0>,
 <matplotlib.patches.Wedge at 0x7f7070792090>,
 <matplotlib.patches.Wedge at 0x7f70707f46d0>,
 <matplotlib.patches.Wedge at 0x7f70707924d0>,
 <matplotlib.patches.Wedge at 0x7f7070792a90>,
 <matplotlib.patches.Wedge at 0x7f707079f510>,
 <matplotlib.patches.Wedge at 0x7f7070792510>,
 <matplotlib.patches.Wedge at 0x7f707079fa50>,
 <matplotlib.patches.Wedge at 0x7f70707a9410>,
 <matplotlib.patches.Wedge at 0x7f70707a9450>],
 [Text(-0.8227559810574577, 0.7301182066173808, 'Kolkata'),
 Text(-0.9813250715975654, 0.49699205612769437, 'Haldia'),
 Text(-1.0974872598930092, -0.07430823892769065, 'Paradeep'),
 Text(-0.7887307236408752, -0.7667488803936667, 'Visakhapatnam'),
 Text(-0.24681684658019215, -1.0719521650914328, 'Ennore'),
 Text(0.2586200998460918, -1.0691658636318304, 'Chennai'),
 Text(0.6508775584042567, -0.8867685176897709, 'Tuticorin'),
 Text(0.8904218552781084, -0.6458706678918711, 'Cochin'),
 Text(1.068042956118005, -0.2632190036580062, 'NMPT'),
 Text(1.083660055487473, 0.18889384357592884, 'Mormugao'),
 Text(0.8515399364034199, 0.6963330644957625, 'Mumbai'),
 Text(0.3009306210259454, 1.058036275998578, 'JNPT'),
 Text(-0.4290148932653011, 1.0128900341876024, 'Kandla')])

```



```

plt.pie(df.Traffic_Achieved,labels=df.Port,colors=colors,startangle=135)

```

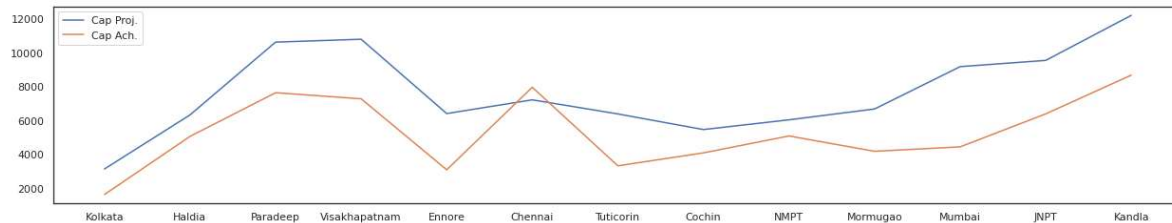


```
([<matplotlib.patches.Wedge at 0x7f7070b6c4d0>,
 <matplotlib.patches.Wedge at 0x7f7070b6c2d0>,
 <matplotlib.patches.Wedge at 0x7f7070b6ca50>,
 <matplotlib.patches.Wedge at 0x7f7070acca10>,
 <matplotlib.patches.Wedge at 0x7f7070a45210>,
 <matplotlib.patches.Wedge at 0x7f70710e2cd0>,
 <matplotlib.patches.Wedge at 0x7f70709c97d0>,
 <matplotlib.patches.Wedge at 0x7f70709c9110>,
 <matplotlib.patches.Wedge at 0x7f7070a412d0>,
 <matplotlib.patches.Wedge at 0x7f7070a45bd0>,
 <matplotlib.patches.Wedge at 0x7f7070947c90>,
 <matplotlib.patches.Wedge at 0x7f7070947f50>,
 <matplotlib.patches.Wedge at 0x7f70709413d0>],
 [Text(-0.8292984732192152, 0.7226783809664425, 'Kolkata'),
 Text(-0.9785761523202932, 0.5023830352529937, 'Haldia'),
 Text(-1.0999917035067, -0.004272261278115301, 'Paradeep'),
 Text(-0.8509776584186843, -0.6970201036356506, 'Visakhapatnam'),
 Text(-0.4510472262335761, -1.0032728440992495, 'Ennore'),
 Text(-0.02876119418292548, -1.0996239328557615, 'Chennai'),
 Text(0.472408825560159, -0.9933931253702492, 'Tuticorin'),
 Text(0.720536507722156, -0.831160117570351, 'Cochin'),
 Text(0.932508592180857, -0.5834618458038848, 'NMPT'),
 Text(1.086706953339725, -0.1704933944852213, 'Mormugao'),
 Text(1.022266221308968, 0.406167173428238, 'Mumbai'),
 Text(0.535803745383295, 0.9606843115369551, 'JNPT'),
 Text(-0.3488393338981779, 1.0432215100952797, 'Kandla')])
```

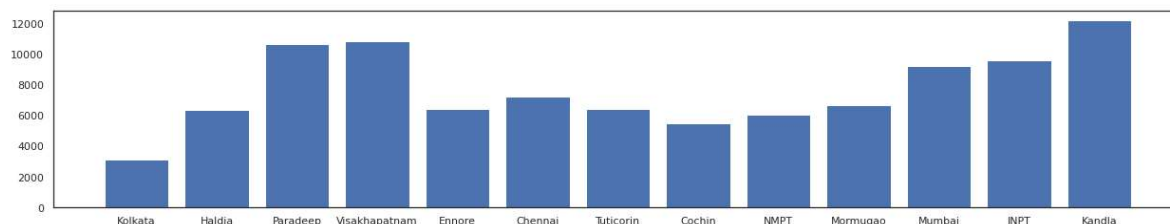


```
plt.plot(df.Port,df.Total_Capacity_Projected,label='Cap Proj.')
plt.plot(df.Port,df.Total_Capacity_Achieved,label='Cap Ach.')
plt.legend()
```

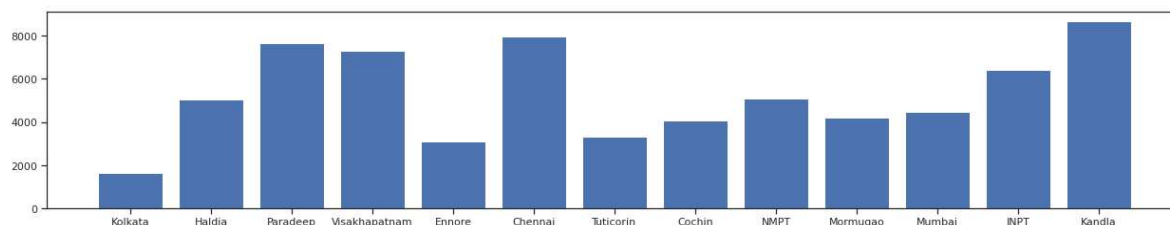
<matplotlib.legend.Legend at 0x7f70710dcb10>



```
plt.bar(df.Port,df.Total_Capacity_Projected)
sns.set_style('ticks')
```

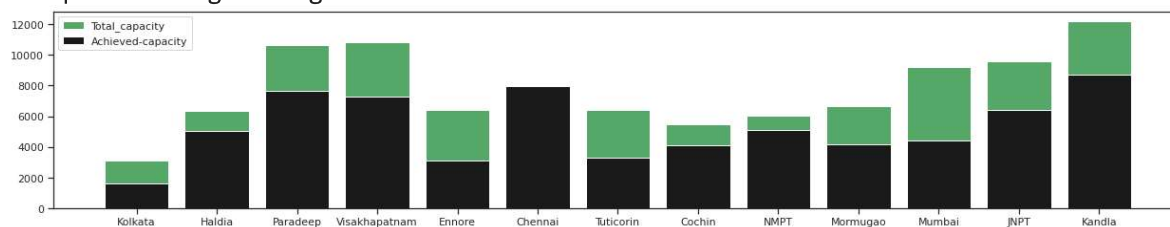


```
plt.bar(df.Port,df.Total_Capacity_Achieved)
sns.set_style('ticks')
```



```
plt.bar(df.Port,df.Total_Capacity_Projected,label='Total_capacity',color='g')
plt.bar(df.Port,df.Total_Capacity_Achieved,label='Achieved-capacity',color='k')
plt.legend()
```

<matplotlib.legend.Legend at 0x7f70706832d0>



[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 7:32 PM

