```
import numpy as num
import pandas as pa
```

```
# Loading the dataset
data = pa.read_csv('/datafile_02.csv')
print(data.columns)
data.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
4	Ennore	4700	1496	3200	6420	3100	4800

```
# Preprocessing the dataset

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

	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	
7	Cochin	3817	2010	5300	

```
# Perparing the Calculations:

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

10	Mumbai	7105	5618	7900

```
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( (data.Total_Capacity_Achieved/data.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
data.rename(columns = {'Traffic in Eleventh Plan (MT) (2011-12) %':'Traffic_Percent','Tota
data.iloc[:,3:4] = Traffic_Percent
data.iloc[:,6:] = Total_Percent
data
```

	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	

```
data.shape
```

```
(13, 7)
```

```
# Checking for null values
```

```
data.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
data.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
```

```
data.describe()
```

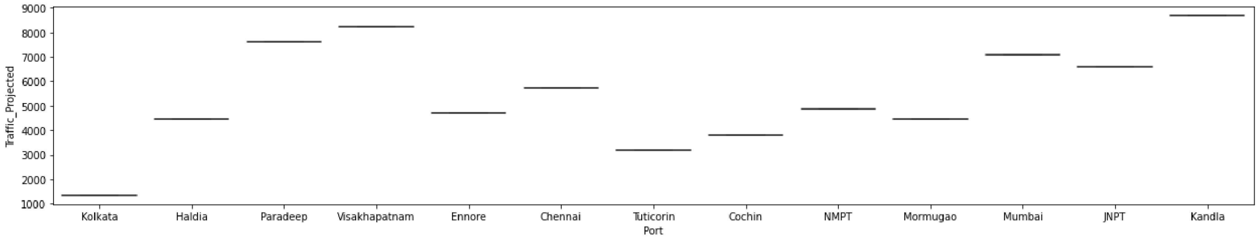
	Traffic_Projected	Traffic_Achieved	Traffic_Percent	Total_Capacity_Proje
count	13.000000	13.000000	13.000000	13.00
mean	5446.846154	4308.846154	77.887692	7705.30
std	2133.280019	2212.894855	19.382398	2570.24
min	1343.000000	1223.000000	31.830000	3145.00
25%	4450.000000	2810.000000	69.690000	6340.00
50%	4881.000000	3900.000000	82.020000	6690.00
75%	7105.000000	5618.000000	91.060000	9560.00
max	8672.000000	8250.000000	99.560000	12220.00

```
#Finding Outliers anr replacing the outliers

import seaborn as sea
import matplotlib.pyplot as plot
plot.rcParams["figure.figsize"] = [17.50, 3.50]
plot.rcParams["figure.autolayout"] = True

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

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe4083de050>
```

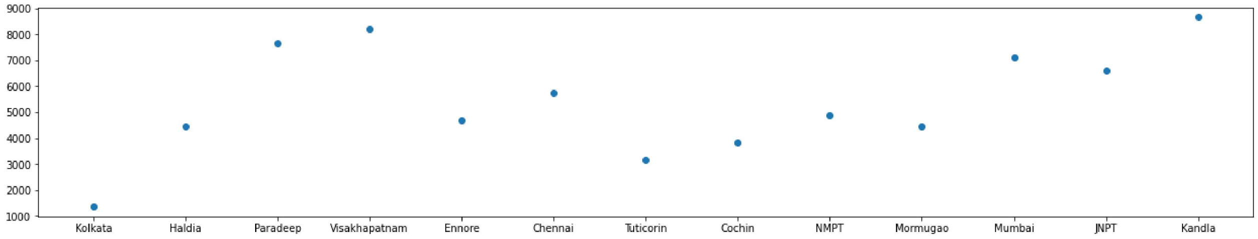


Visualization using various plots

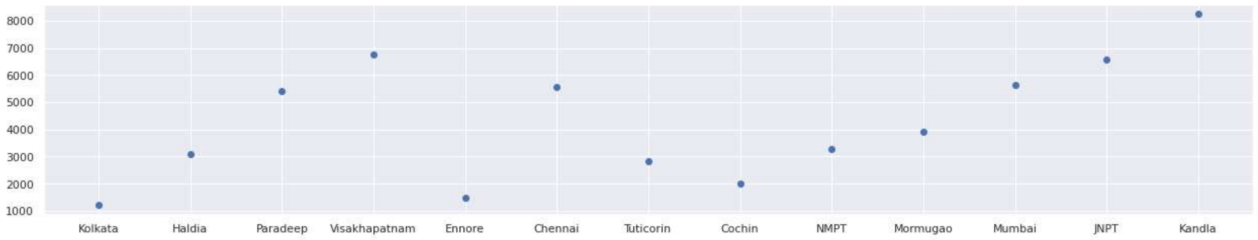
```
import matplotlib.pyplot as plot
import seaborn as sea
```

```
import matplotlib.pyplot as plot
import seaborn as sea

plot.scatter(data.Port,data.Traffic_Projected)
sea.set()
```

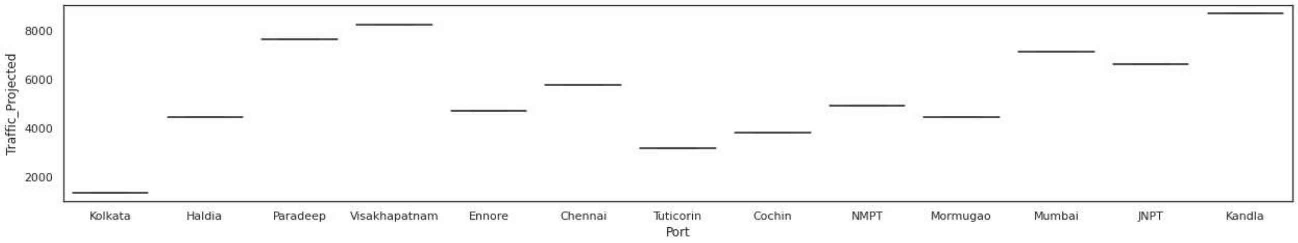


```
plot.scatter(data.Port,data.Traffic_Achieved)
sea.set_style('white')
sea.set_context('notebook')
```



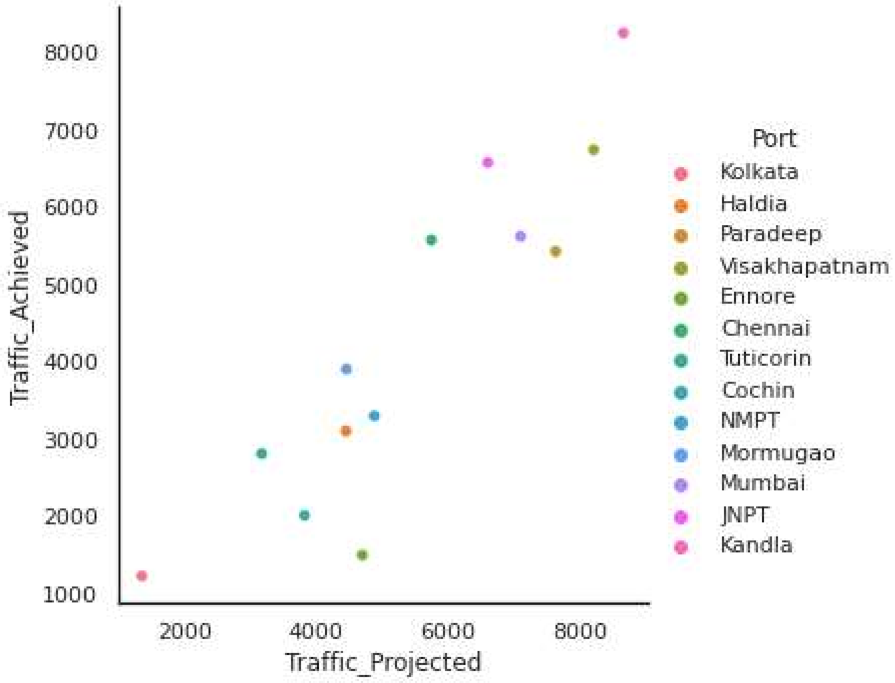
```
sea.boxplot(x='Port',y='Traffic_Projected',data=data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe3f9074310>
```



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

```
<seaborn.axisgrid.FacetGrid at 0x7fe4082dd050>
```



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

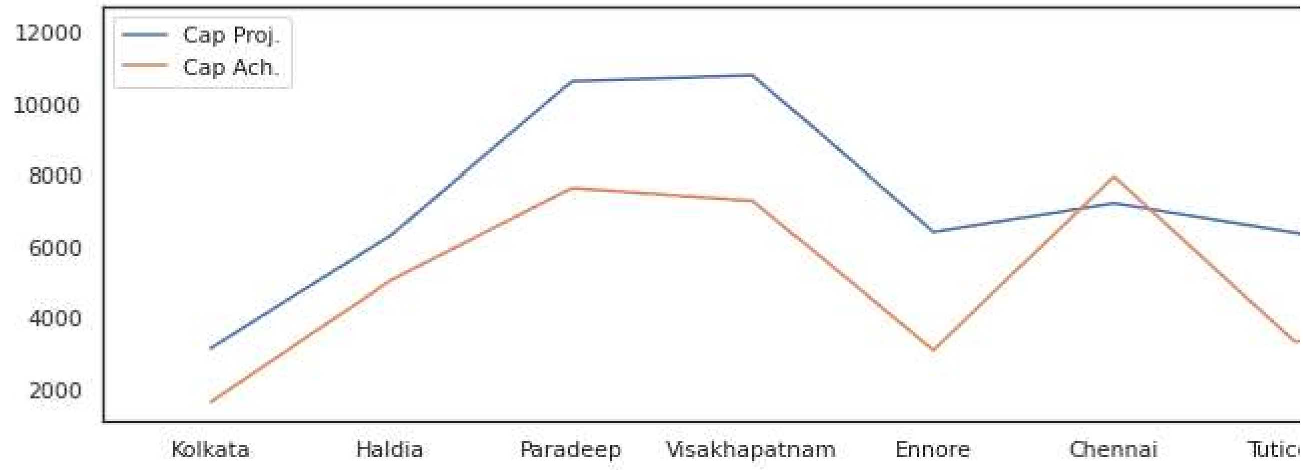
```
([<matplotlib.patches.Wedge at 0x7fe3f8d951d0>,
  <matplotlib.patches.Wedge at 0x7fe3f8d95590>,
  <matplotlib.patches.Wedge at 0x7fe3f8d95550>,
  <matplotlib.patches.Wedge at 0x7fe3f8d9e190>,
  <matplotlib.patches.Wedge at 0x7fe3f8d9ead0>,
  <matplotlib.patches.Wedge at 0x7fe3f8d95050>,
  <matplotlib.patches.Wedge at 0x7fe3f8da83d0>,
  <matplotlib.patches.Wedge at 0x7fe3f8da8410>,
  <matplotlib.patches.Wedge at 0x7fe3f8da8950>,
  <matplotlib.patches.Wedge at 0x7fe3f8da8e90>,
  <matplotlib.patches.Wedge at 0x7fe3f8d353d0>,
  <matplotlib.patches.Wedge at 0x7fe3f8d35850>,
  <matplotlib.patches.Wedge at 0x7fe3f8d432d0>],
 [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')])
```

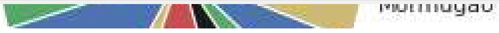
```
plot.pie(data.Traffic_Achieved,labels=data.Port,colors=colors,startangle=135)
```

```
([<matplotlib.patches.Wedge at 0x7fe3f8d0a690>,
  <matplotlib.patches.Wedge at 0x7fe3f8d0aa50>,
  <matplotlib.patches.Wedge at 0x7fe3f8d0aa10>,
  <matplotlib.patches.Wedge at 0x7fe3f8d170d0>,
  <matplotlib.patches.Wedge at 0x7fe3f8d178d0>,
  <matplotlib.patches.Wedge at 0x7fe3f8d0a4d0>,
  <matplotlib.patches.Wedge at 0x7fe3f8d211d0>,
  <matplotlib.patches.Wedge at 0x7fe3f8d21790>,
  <matplotlib.patches.Wedge at 0x7fe3f8d2d210>,
  <matplotlib.patches.Wedge at 0x7fe3f8d21210>,
  <matplotlib.patches.Wedge at 0x7fe3f8d2d750>,
  <matplotlib.patches.Wedge at 0x7fe3f8cbb110>,
```

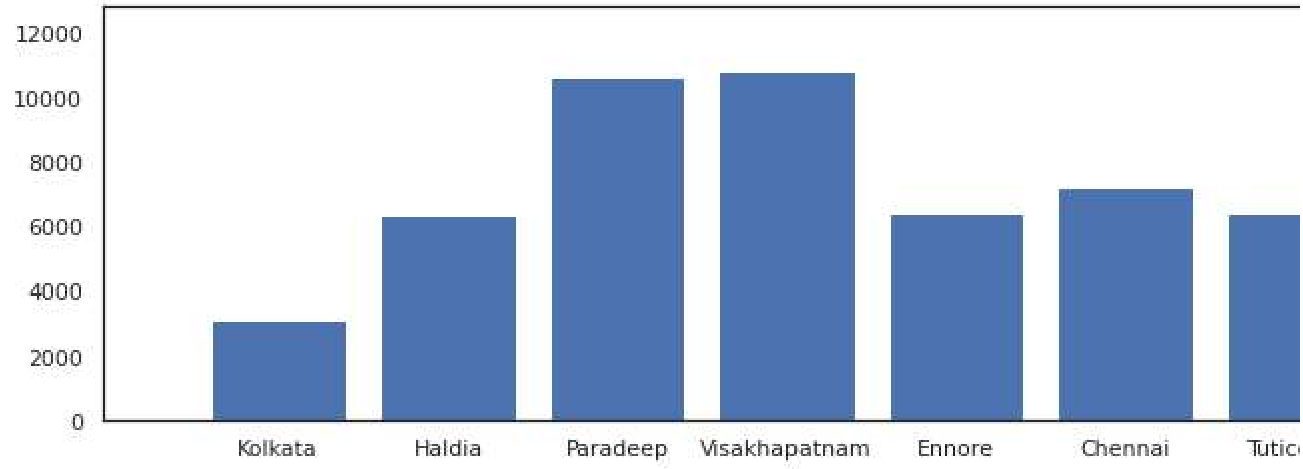
```
plot.plot(data.Port,data.Total_Capacity_Projected,label='Cap Proj.')
plot.plot(data.Port,data.Total_Capacity_Achieved,label='Cap Ach.')
plot.legend()
```

```
<matplotlib.legend.Legend at 0x7fe3f9074b50>
```

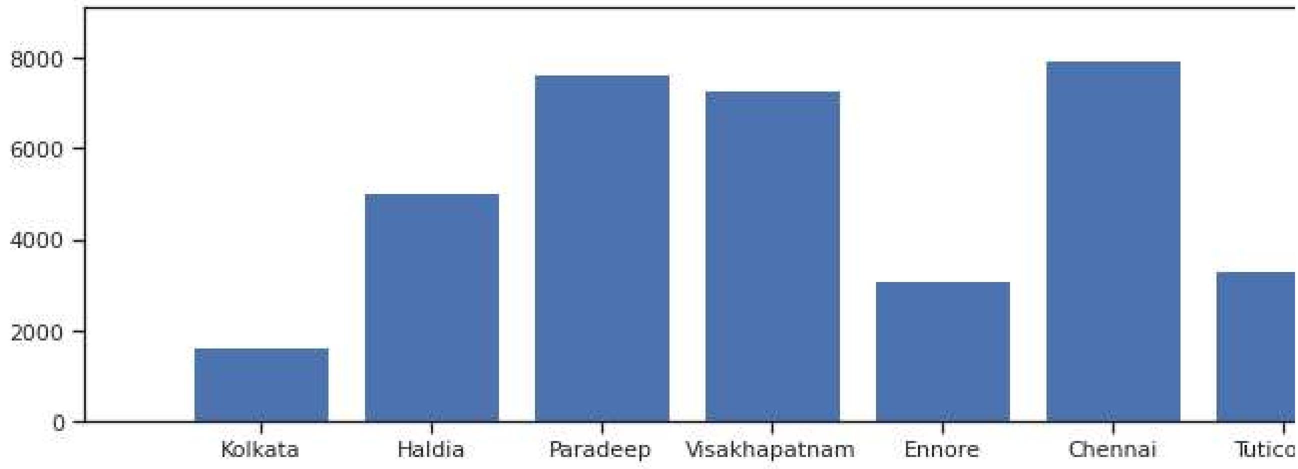




```
plot.bar(data.Port,data.Total_Capacity_Projected)
sea.set_style('ticks')
```

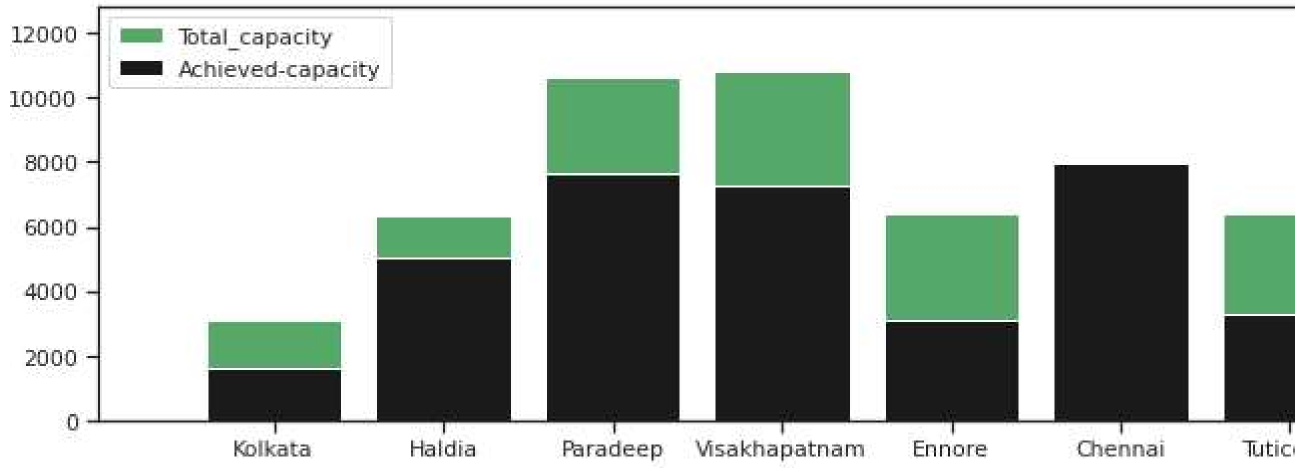


```
plot.bar(data.Port,data.Total_Capacity_Achieved)
sea.set_style('ticks')
```



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

```
<matplotlib.legend.Legend at 0x7fe3f8bc3050>
```



Colab paid products - Cancel contracts here

✓ 0s completed at 12:14 ● ×