

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

```
# Loading the dataset
```

```
df = pd.read_csv('D:/ibm/datafile_02.csv')
```

```
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.	\
0	Kolkata	1343	
1	Haldia	4450	
2	Paradeep	7640	
3	Visakhapatnam	8220	
4	Ennore	4700	

	Traffic in Eleventh Plan (MT) (2011-12) Ach.	\
0	1223	
1	3101	
2	5425	
3	6742	
4	1496	

	Traffic in Eleventh Plan (MT) (2011-12) %	\
0	9100	
1	7000	
2	7100	
3	8200	
4	3200	

	Total Capacity in Eleventh Plan (MT) (2011-12) Proj.	\
0	3145	
1	6340	
2	10640	
3	10810	
4	6420	

	Total Capacity in Eleventh Plan (MT) (2011-12) Ach.	\
0	1635	
1	5070	
2	7650	
3	7293	
4	3100	

	Total Capacity in Eleventh Plan (MT) (2011-12) %
0	5100
1	7900
2	7100
3	6700
4	4800

*# Preprocessing the dataset*

*# Renaming the columns*

```
df.rename(columns = {'Traffic in Eleventh Plan (MT) (2011-12)Proj.': 'Traffic_Projected', 'Traffic in Eleventh Plan (MT) (2011-12)Ach.': 'Traffic_Achieved', 'Total Capacity in Eleventh Plan (MT) (2011-12) Proj.': 'Total_Capacity_Projected', 'Total Capacity in Eleventh Plan (MT) (2011-12) Ach.': 'Total_Capacity_Achieved'}, inplace = True)
df
```

	Port	Traffic_Projected	Traffic_Achieved \
0	Kolkata	1343	1223
1	Haldia	4450	3101
2	Paradeep	7640	5425
3	Visakhapatnam	8220	6742
4	Ennore	4700	1496
5	Chennai	5750	5571
6	Tuticorin	3172	2810
7	Cochin	3817	2010
8	NMPT	4881	3294
9	Mormugao	4455	3900
10	Mumbai	7105	5618
11	JNPT	6604	6575
12	Kandla	8672	8250

	Traffic in Eleventh Plan (MT) (2011-12) %
0	9100
3145	
1	7000
6340	
2	7100
10640	
3	8200
10810	
4	3200
6420	
5	9700
7230	
6	8900
6398	

7	5300
5475	
8	6800
6050	
9	8800
6690	
10	7900
9191	
11	10000
9560	
12	9500
12220	

	Total_Capacity_Achieved (2011-12) %	Total Capacity in Eleventh Plan (MT)
0	1635	
5100		
1	5070	
7900		
2	7650	
7100		
3	7293	
6700		
4	3100	
4800		
5	7972	
11000		
6	3334	
5200		
7	4098	
7400		
8	5097	
8400		
9	4190	
6200		
10	4453	
4800		
11	6400	
6600		
12	8691	
7100		

*# 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 in Eleventh Plan (MT) (2011-12)
%':'Total_Percent'}, inplace = True)
df.iloc[:,3:4] = Traffic_Percent
df.iloc[:,6:] = Total_Percent
df

```

	Port	Traffic_Projected	Traffic_Achieved
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			

	Total_Capacity_Projected	Total_Capacity_Achieved	Total_Percent
0	3145	1635	51.99
1	6340	5070	79.97
2	10640	7650	71.90
3	10810	7293	67.47
4	6420	3100	48.29
5	7230	7972	110.26
6	6398	3334	52.11
7	5475	4098	74.85
8	6050	5097	84.25
9	6690	4190	62.63
10	9191	4453	48.45
11	9560	6400	66.95
12	12220	8691	71.12

```
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 \
count	13.000000	13.000000	13.000000
mean	5446.846154	4308.846154	77.887692
std	2133.280019	2212.894855	19.382398
min	1343.000000	1223.000000	31.830000
25%	4450.000000	2810.000000	69.690000
50%	4881.000000	3900.000000	82.020000
75%	7105.000000	5618.000000	91.060000
max	8672.000000	8250.000000	99.560000

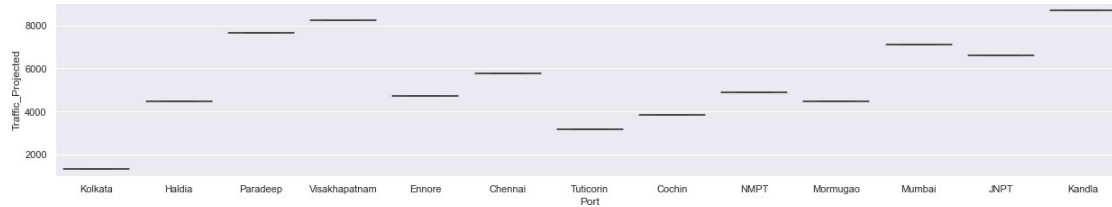
	Total_Capacity_Projected	Total_Capacity_Achieved
Total_Percent		
count	13.000000	13.000000
13.000000		
mean	7705.307692	5306.384615
68.480000		
std	2570.242673	2140.254796
17.252637		
min	3145.000000	1635.000000
48.290000		
25%	6340.000000	4098.000000
52.110000		
50%	6690.000000	5070.000000
67.470000		
75%	9560.000000	7293.000000
74.850000		
max	12220.000000	8691.000000
110.260000		

```
#Finding Outliers anr 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)
```

```
<AxesSubplot:xlabel='Port', ylabel='Traffic_Projected'>
```

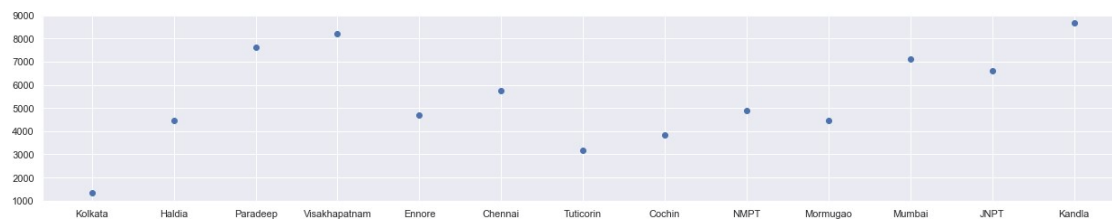


```
# Visualization using various plots
```

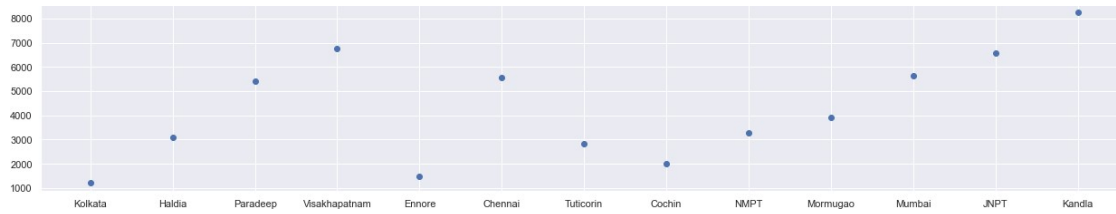
```
import matplotlib.pyplot as plt
import seaborn as sns
```

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

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

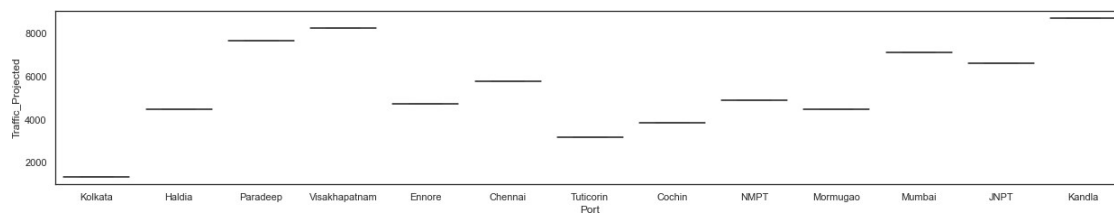


```
plt.scatter(df.Port,df.Traffic_Achieved)
sns.set_style('white')
sns.set_context('notebook')
```



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

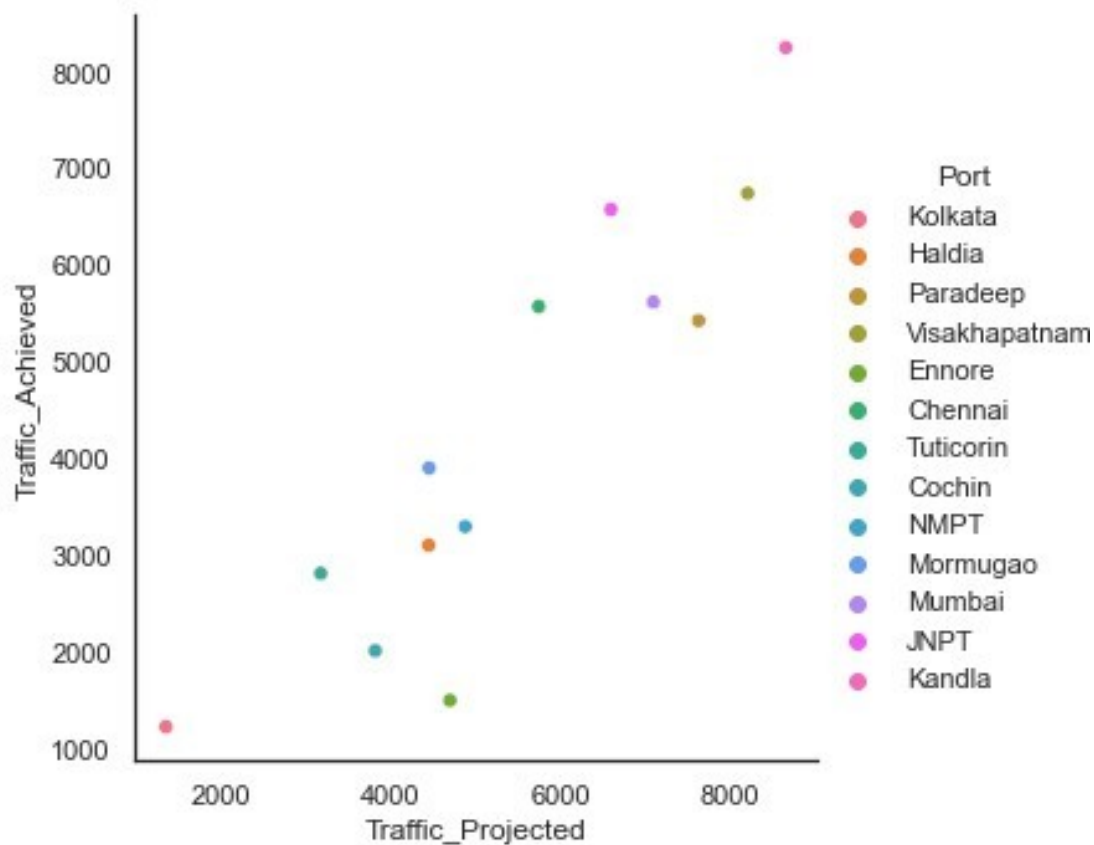
```
<AxesSubplot:xlabel='Port', ylabel='Traffic_Projected'>
```



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

```
<seaborn.axisgrid.FacetGrid at 0x1a33a579940>
```





```

colors=['r','k','g','b','y']
plt.pie(df.Traffic_Projected,labels=df.Port,colors=colors,startangle=1
35)

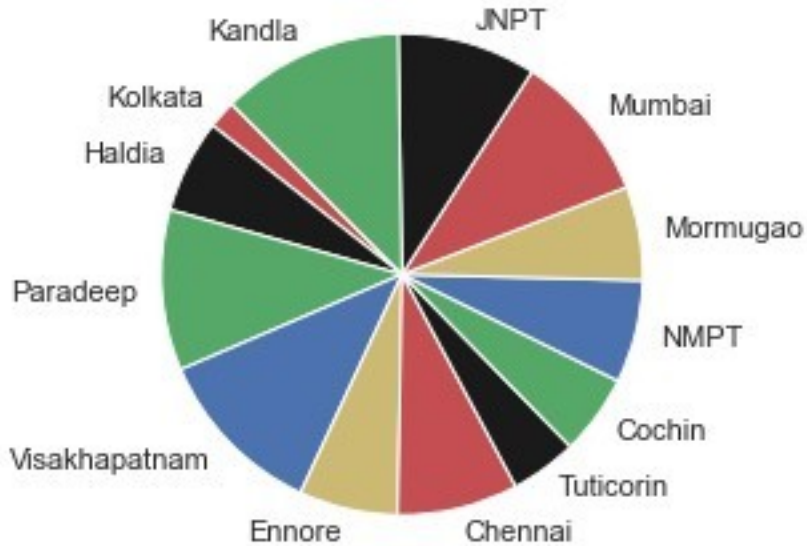
```

```

([<matplotlib.patches.Wedge at 0x1a339ebce80>,
 <matplotlib.patches.Wedge at 0x1a339ebc2b0>,
 <matplotlib.patches.Wedge at 0x1a339eaba30>,
 <matplotlib.patches.Wedge at 0x1a339eab250>,
 <matplotlib.patches.Wedge at 0x1a339eab190>,
 <matplotlib.patches.Wedge at 0x1a339ebcf10>,
 <matplotlib.patches.Wedge at 0x1a339f12d60>,
 <matplotlib.patches.Wedge at 0x1a339f125b0>,
 <matplotlib.patches.Wedge at 0x1a339b77040>,
 <matplotlib.patches.Wedge at 0x1a339e9dd90>,
 <matplotlib.patches.Wedge at 0x1a339e9d490>,
 <matplotlib.patches.Wedge at 0x1a336bf5d90>,
 <matplotlib.patches.Wedge at 0x1a339e7a940>],
 [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, -0.645870667891871, 'Cochin'),
Text(1.068042956118005, -0.2632190036580062, 'NMPT'),
Text(1.083660055487473, 0.18889384357592884, 'Mormugao'),
Text(0.85153993640342, 0.6963330644957625, 'Mumbai'),
Text(0.3009306210259454, 1.058036275998578, 'JNPT'),
Text(-0.4290148932653011, 1.0128900341876024, 'Kandla')])
```



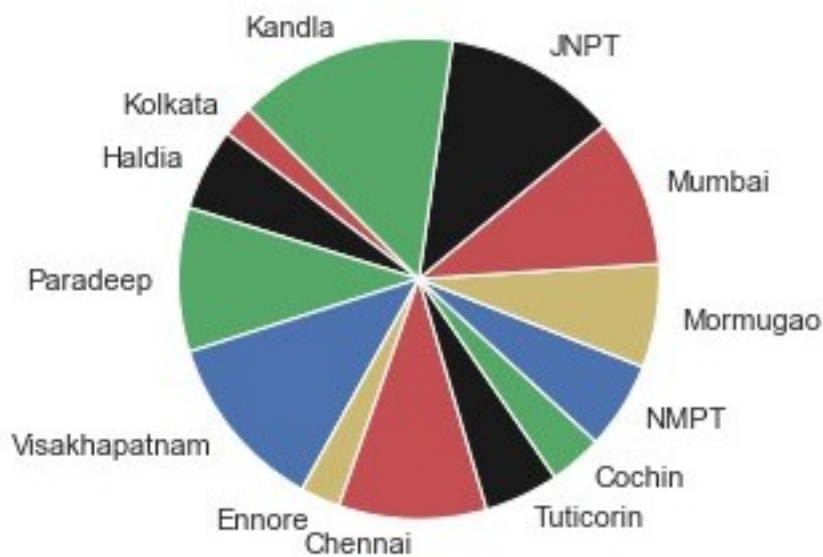
```
plt.pie(df.Traffic_Achieved, labels=df.Port, colors=colors, startangle=135)
```

```
([<matplotlib.patches.Wedge at 0x1a339e83d90>,
<matplotlib.patches.Wedge at 0x1a339e83370>,
<matplotlib.patches.Wedge at 0x1a339cdfd0>,
<matplotlib.patches.Wedge at 0x1a339e891f0>,
<matplotlib.patches.Wedge at 0x1a339e898e0>,
<matplotlib.patches.Wedge at 0x1a339e83460>,
<matplotlib.patches.Wedge at 0x1a339e23e80>,
<matplotlib.patches.Wedge at 0x1a339f9ab80>,
<matplotlib.patches.Wedge at 0x1a339f9a490>,
<matplotlib.patches.Wedge at 0x1a339f9a760>,
<matplotlib.patches.Wedge at 0x1a339e4dc70>,
<matplotlib.patches.Wedge at 0x1a33a21e580>,
<matplotlib.patches.Wedge at 0x1a33a21ee20>],
[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, -1.0032728440992498, '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')]

```

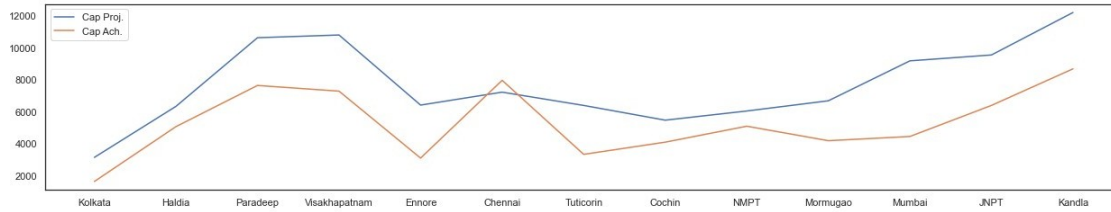


```

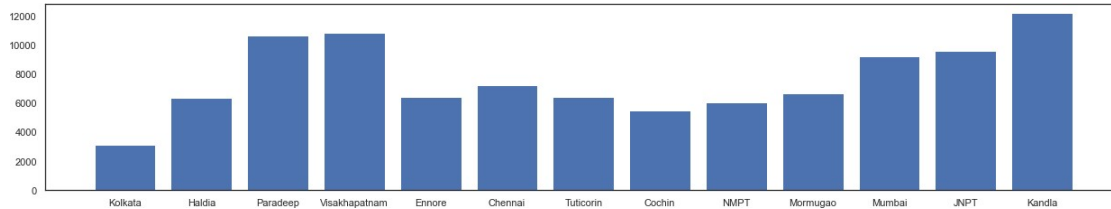
plt.plot(df.Port,df.Total_Capacity_Projected,label='Cap Proj.')
plt.plot(df.Port,df.Total_Capacity_Achieved,label='Cap Ach.')
plt.legend()

```

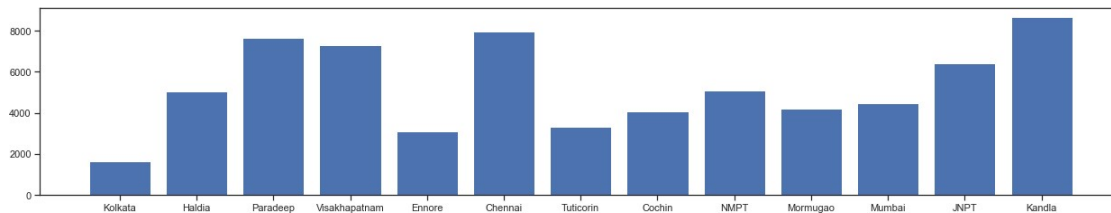
<matplotlib.legend.Legend at 0x1a33a256460>



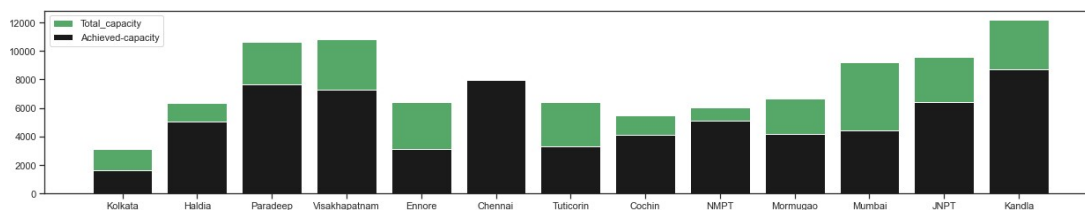
```
plt.bar(df.Port,df.Total_Capacity_Projected)
sns.set_style('ticks')
```



```
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 0x1a33a2bb460>
```

