In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
df = pd.read_csv('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')
```

Out[2]:

	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

In [3]:
```
# Renaming the columns
df.rename(columns = {'Traffic in Eleventh Plan (MT) (2011-12)Proj.':'Traffic_Proj
df
```

Out[3]:

	Port	Traffic_Projected	Traffic_Achieved	Traffic in Eleventh Plan (MT) (2011-12) %	Total_Capacity_Projected	Total_Cap
0	Kolkata	1343	1223	9100	3145	
1	Haldia	4450	3101	7000	6340	
2	Paradeep	7640	5425	7100	10640	
3	Visakhapatnam	8220	6742	8200	10810	
4	Ennore	4700	1496	3200	6420	
5	Chennai	5750	5571	9700	7230	
6	Tuticorin	3172	2810	8900	6398	
7	Cochin	3817	2010	5300	5475	
8	NMPT	4881	3294	6800	6050	
9	Mormugao	4455	3900	8800	6690	
10	Mumbai	7105	5618	7900	9191	
11	JNPT	6604	6575	10000	9560	
12	Kandla	8672	8250	9500	12220	

In [4]:
```
# Perparing the Calculations:

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

Out[4]:
```
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
```

In [5]:
```
Total_Percent = round( (df.Total_Capacity_Achieved/df.Total_Capacity_Projected)*1
Total_Percent
```

Out[5]:
```
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
```

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

Out[6]:

	Port	Traffic_Projected	Traffic_Achieved	Traffic_Percent%	Total_Capacity_Projected	T
0	Kolkata	1343	1223	91.06	3145	
1	Haldia	4450	3101	69.69	6340	
2	Paradeep	7640	5425	71.01	10640	
3	Visakhapatnam	8220	6742	82.02	10810	
4	Ennore	4700	1496	31.83	6420	
5	Chennai	5750	5571	96.89	7230	
6	Tuticorin	3172	2810	88.59	6398	
7	Cochin	3817	2010	52.66	5475	
8	NMPT	4881	3294	67.49	6050	
9	Mormugao	4455	3900	87.54	6690	
10	Mumbai	7105	5618	79.07	9191	
11	JNPT	6604	6575	99.56	9560	
12	Kandla	8672	8250	95.13	12220	

In [7]:
```
df.shape
```

Out[7]:
```
(13, 7)
```

In [8]:
```
# Checking for null values

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

In [9]:
```
# Checking for null values

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

Out[9]:
```
Port                        0
Traffic_Projected           0
Traffic_Achieved            0
Traffic_Percent%            0
Total_Capacity_Projected    0
Total_Capacity_Achieved     0
Total_Percent%              0
dtype: int64
```

In [10]:
```
df.describe()
```

Out[10]:

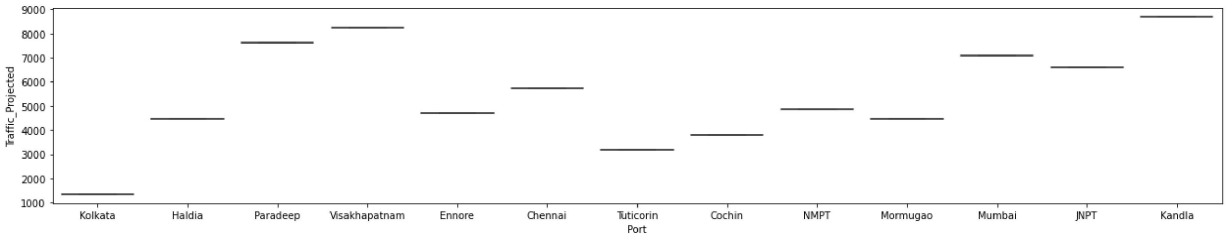
	Traffic_Projected	Traffic_Achieved	Traffic_Percent%	Total_Capacity_Projected	Total_Capacity
count	13.000000	13.000000	13.000000	13.000000	
mean	5446.846154	4308.846154	77.887692	7705.307692	5
std	2133.280019	2212.894855	19.382398	2570.242673	2
min	1343.000000	1223.000000	31.830000	3145.000000	1
25%	4450.000000	2810.000000	69.690000	6340.000000	4
50%	4881.000000	3900.000000	82.020000	6690.000000	5
75%	7105.000000	5618.000000	91.060000	9560.000000	7
max	8672.000000	8250.000000	99.560000	12220.000000	8

In [11]:
```
#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)
```

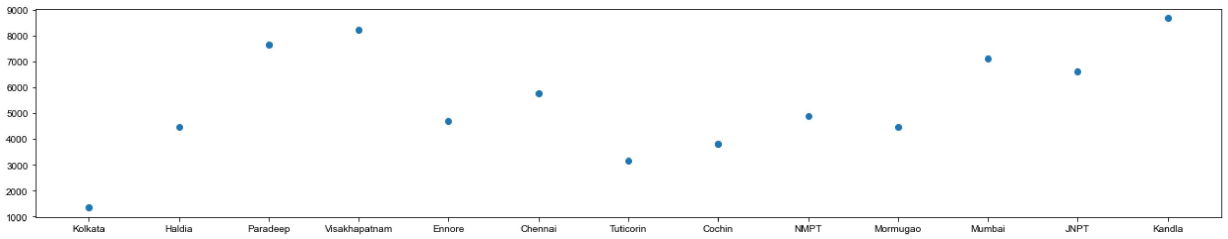
Out[11]: <AxesSubplot:xlabel='Port', ylabel='Traffic_Projected'>



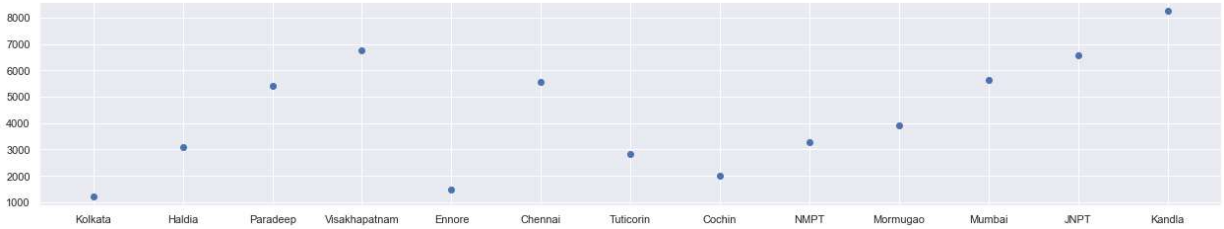
In [12]:
```
# Visualization using various plots
import matplotlib.pyplot as plt
import seaborn as sns


import matplotlib.pyplot as plt
import seaborn as sns

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

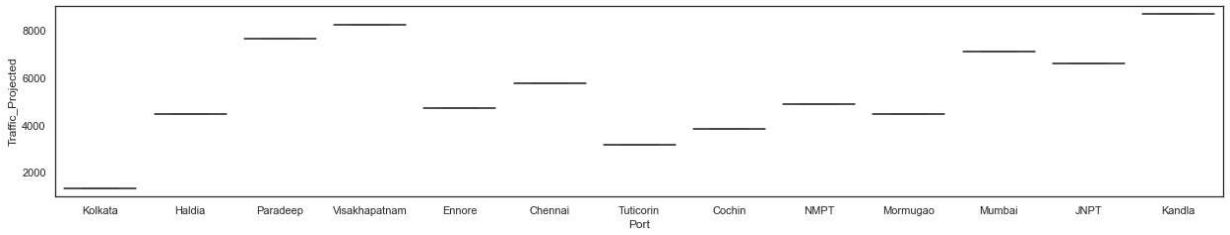


In [13]:
```
plt.scatter(df.Port,df.Traffic_Achieved)
sns.set_style('white')
sns.set_context('notebook')
```



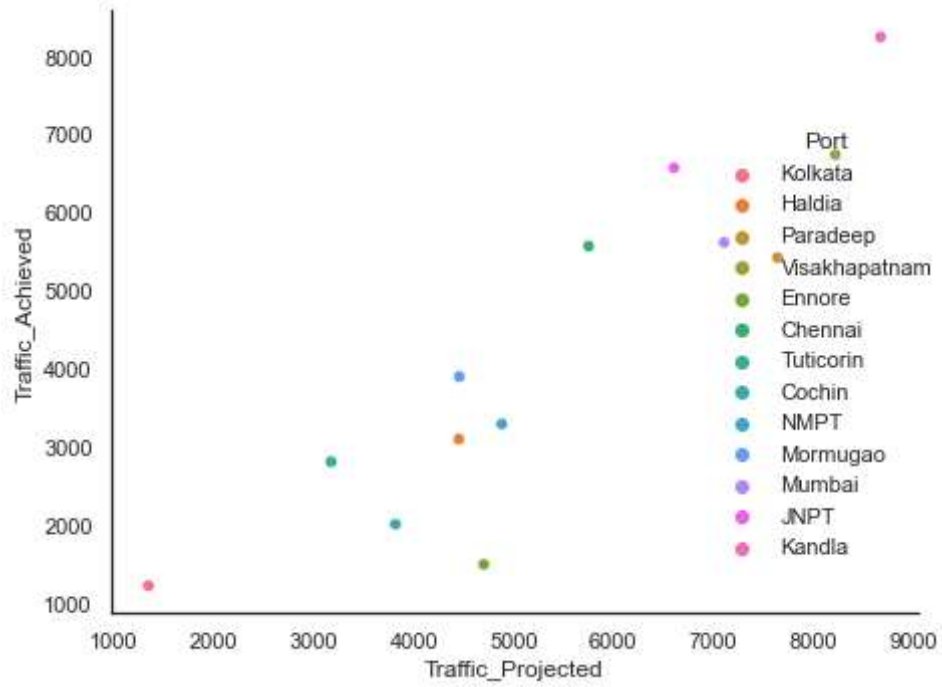
In [14]:
```
sns.boxplot(x='Port',y='Traffic_Projected',data=df)
```

Out[14]: <AxesSubplot:xlabel='Port', ylabel='Traffic_Projected'>



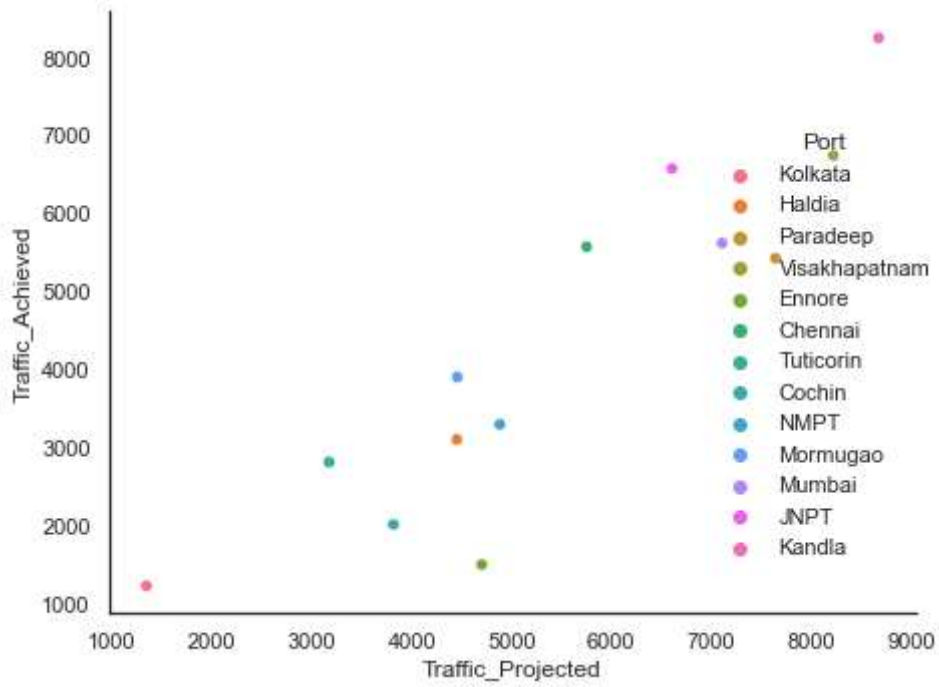
In [15]:
```
sns.relplot(data=df,x="Traffic_Projected",y='Traffic_Achieved',hue='Port')
```

Out[15]: <seaborn.axisgrid.FacetGrid at 0x2101ad12fd0>



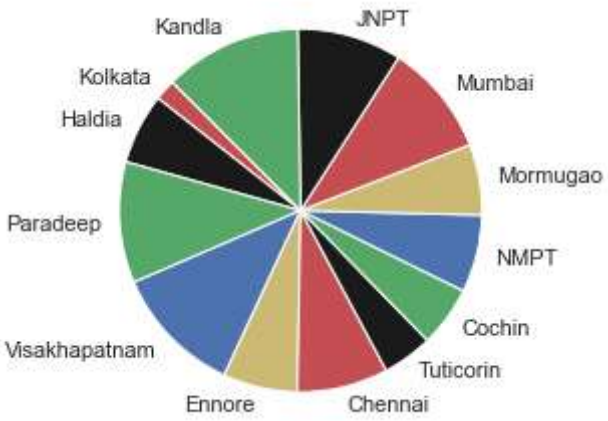
In [16]:
```
sns.relplot(data=df,x="Traffic_Projected",y='Traffic_Achieved',hue='Port')
```

Out[16]:
```
<seaborn.axisgrid.FacetGrid at 0x2101c0b47c0>
```



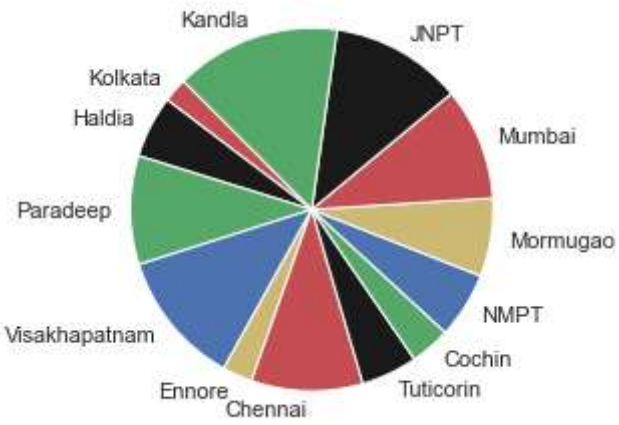
In [17]:
```
colors=['r','k','g','b','y']
plt.pie(df.Traffic_Projected,labels=df.Port,colors=colors,startangle=135)
```

Out[17]:
```
([<matplotlib.patches.Wedge at 0x2101c185eb0>,
  <matplotlib.patches.Wedge at 0x2101c192400>,
  <matplotlib.patches.Wedge at 0x2101c192880>,
  <matplotlib.patches.Wedge at 0x2101c192d00>,
  <matplotlib.patches.Wedge at 0x2101c19f1c0>,
  <matplotlib.patches.Wedge at 0x2101c185040>,
  <matplotlib.patches.Wedge at 0x2101c19fa90>,
  <matplotlib.patches.Wedge at 0x2101c19ff10>,
  <matplotlib.patches.Wedge at 0x2101c1ae3d0>,
  <matplotlib.patches.Wedge at 0x2101c1ae850>,
  <matplotlib.patches.Wedge at 0x2101c1aecd0>,
  <matplotlib.patches.Wedge at 0x2101c30c190>,
  <matplotlib.patches.Wedge at 0x2101c30c610>],
 [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')])
```



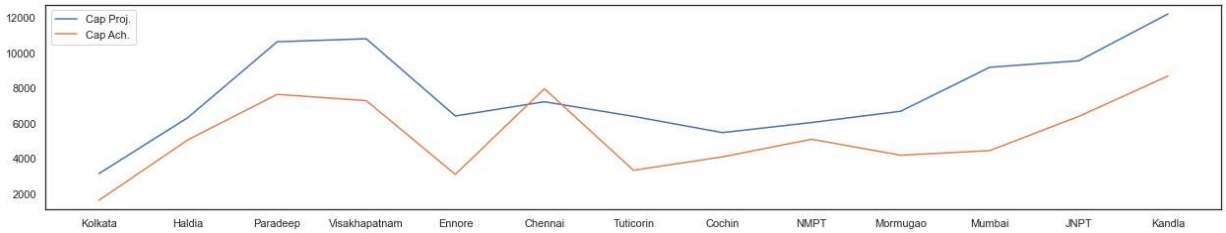
In [18]:
```
plt.pie(df.Traffic_Achieved,labels=df.Port,colors=colors,startangle=135)
```

Out[18]:
```
([<matplotlib.patches.Wedge at 0x2101c346be0>,
  <matplotlib.patches.Wedge at 0x2101c354130>,
  <matplotlib.patches.Wedge at 0x2101c3545b0>,
  <matplotlib.patches.Wedge at 0x2101c3549a0>,
  <matplotlib.patches.Wedge at 0x2101c354dc0>,
  <matplotlib.patches.Wedge at 0x2101c346af0>,
  <matplotlib.patches.Wedge at 0x2101c363610>,
  <matplotlib.patches.Wedge at 0x2101c363a30>,
  <matplotlib.patches.Wedge at 0x2101c363e50>,
  <matplotlib.patches.Wedge at 0x2101c3712b0>,
  <matplotlib.patches.Wedge at 0x2101c3716d0>,
  <matplotlib.patches.Wedge at 0x2101c371af0>,
  <matplotlib.patches.Wedge at 0x2101c371f10>],
 [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')])
```

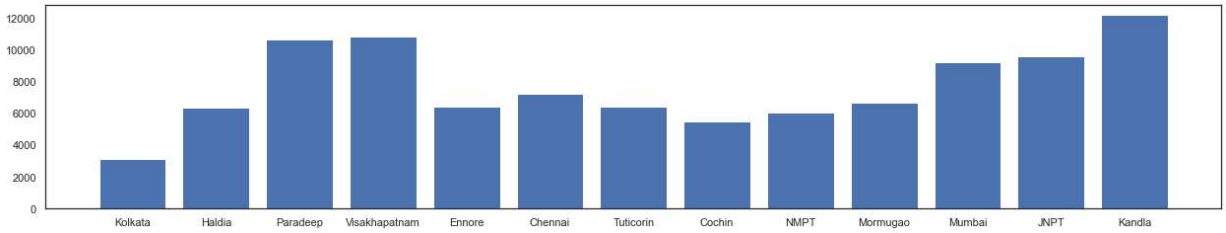


In [19]:
```
plt.plot(df.Port,df.Total_Capacity_Projected,label='Cap Proj.')
plt.plot(df.Port,df.Total_Capacity_Achieved,label='Cap Ach.')
plt.legend()
```

Out[19]: <matplotlib.legend.Legend at 0x2101aafd160>

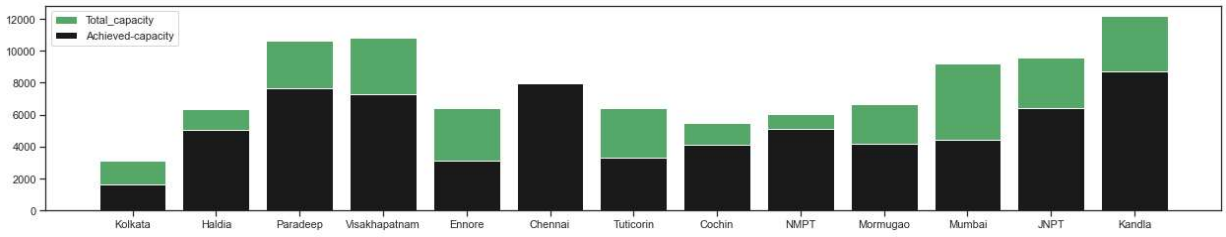


In [20]:
```
plt.bar(df.Port,df.Total_Capacity_Projected)
sns.set_style('ticks')
```



In [21]:
```
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()
```

Out[21]: <matplotlib.legend.Legend at 0x2101633e8e0>



In [ ]: