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

```
# Loading the dataset
df = pd.read_csv('/Railway Traffic in all regions.new.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. | 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
df.rename(columns = {'Traffic in Eleventh Plan (MT) (2011-12)Proj.':'Traffic_Projected','
df
```

| | Port | Traffic_Projected | Traffic_Achieved | Traffic in Eleventh Plan (MT) (2011-12) % | Total_Capacity_Pro |
|---|---|---|---|---|---|
| **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
df.iloc[:,3:4] = Traffic_Percent
df.iloc[:,6:] = Total_Percent
df
```

| | Port | Traffic_Projected | Traffic_Achieved | Traffic_Percent | Total_Capac |
|---|---|---|---|---|---|
| **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 Capacity in Eleventh Plan (MT) (2011-12) %  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 Capacity in Eleventh Plan (MT) (2011-12) %  13 non-null     float64
dtypes: float64(2), int64(4), object(1)
memory usage: 856.0+ bytes
```

```
df.describe()
```

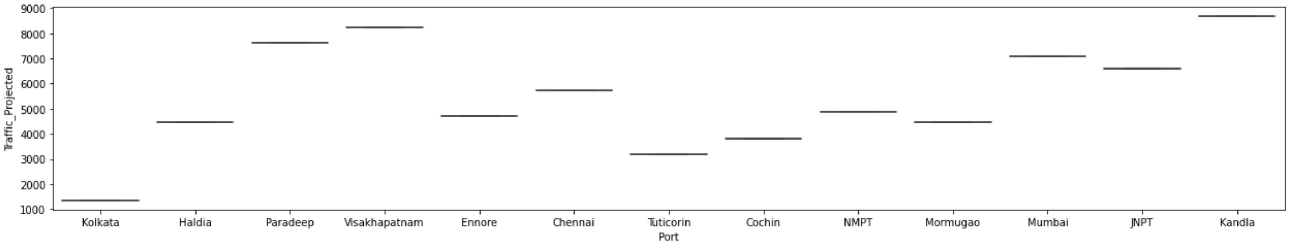
| | Traffic_Projected | Traffic_Achieved | Traffic_Percent | Total_Capacity_Project |
|---|---|---|---|---|
| **count** | 13.000000 | 13.000000 | 13.000000 | 13.00000 |
| **mean** | 5446.846154 | 4308.846154 | 77.887692 | 7705.3076 |
| **std** | 2133.280019 | 2212.894855 | 19.382398 | 2570.2426 |
| **min** | 1343.000000 | 1223.000000 | 31.830000 | 3145.0000 |
| **25%** | 4450.000000 | 2810.000000 | 69.690000 | 6340.0000 |
| **50%** | 4881.000000 | 3900.000000 | 82.020000 | 6690.0000 |
| **75%** | 7105.000000 | 5618.000000 | 91.060000 | 9560.0000 |
| **max** | 8672.000000 | 8250.000000 | 99.560000 | 12220.0000 |

```
#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)
```

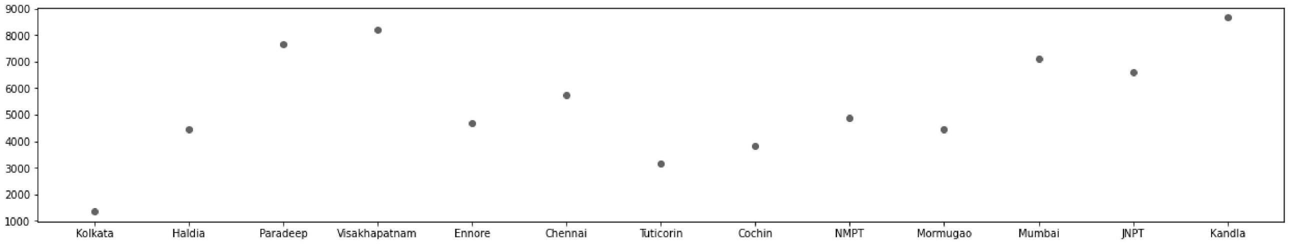
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3a8a1dbe90>
```



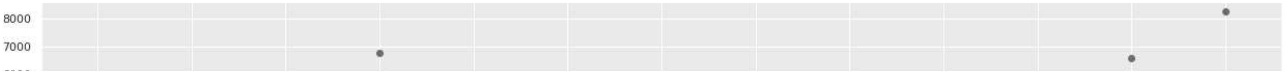
```
# Visualization using various plots
import matplotlib.pyplot as plt
import seaborn as sns
```

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

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

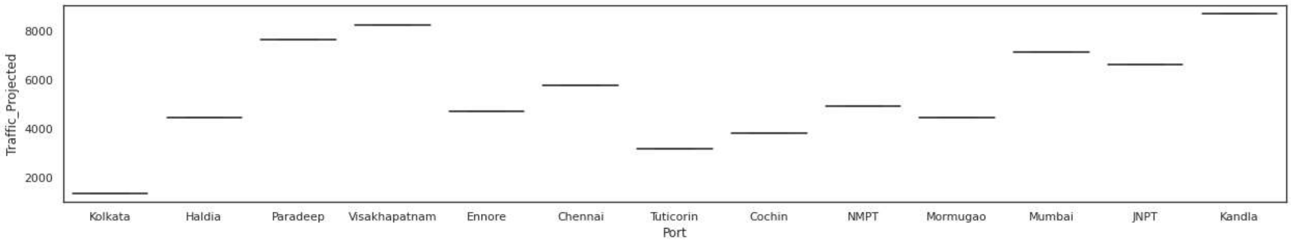


```
plt.scatter(df.Port,df.Traffic_Achieved)
sns.set_style('white')
sns.set_context('notebook')
```



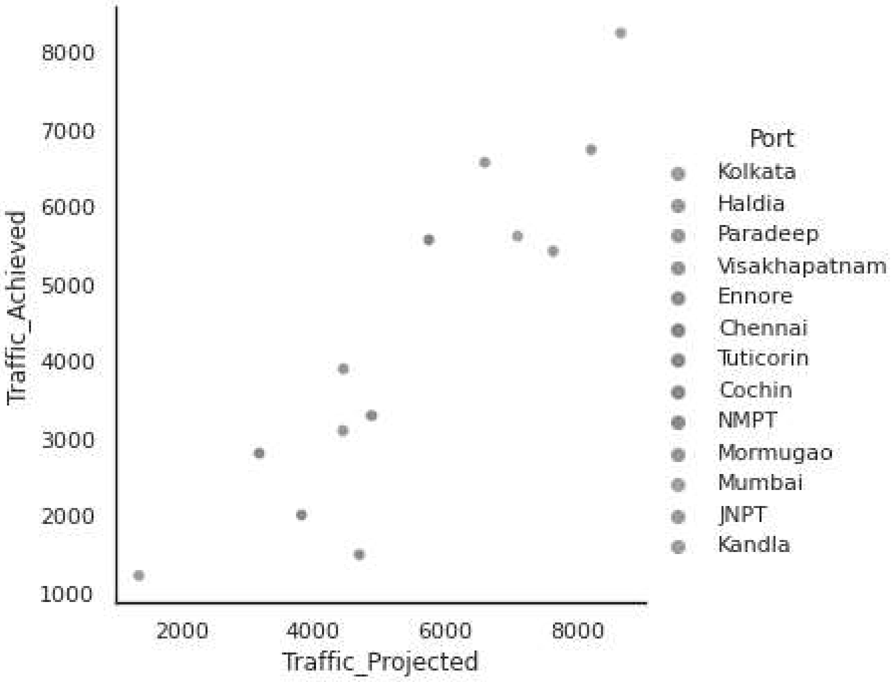
```
sns.boxplot(x='Port',y='Traffic_Projected',data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3a87236190>
```



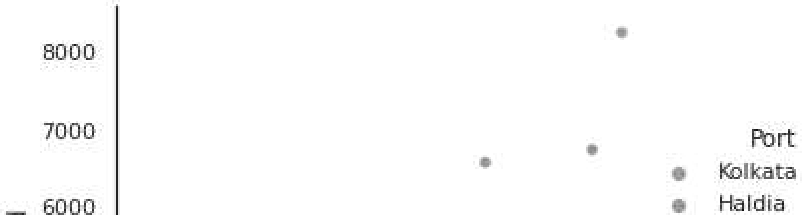
```
sns.relplot(data=df,x="Traffic_Projected",y='Traffic_Achieved',hue='Port')
```

```
<seaborn.axisgrid.FacetGrid at 0x7f3a870bf690>
```



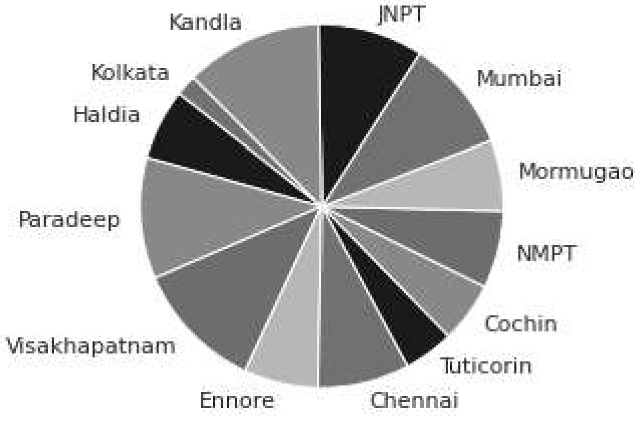
```
sns.relplot(data=df,x="Traffic_Projected",y='Traffic_Achieved',hue='Port')
```

```
<seaborn.axisgrid.FacetGrid at 0x7f3a870c8850>
```



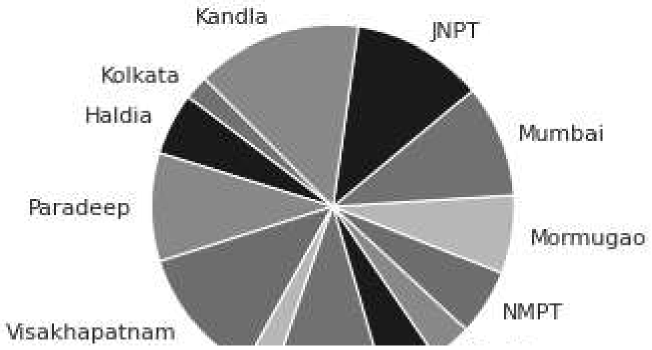
```
colors=['r','k','g','b','y']
plt.pie(df.Traffic_Projected,labels=df.Port,colors=colors,startangle=135)
```

```
([<matplotlib.patches.Wedge at 0x7f3aa175e990>,
  <matplotlib.patches.Wedge at 0x7f3a8724b890>,
  <matplotlib.patches.Wedge at 0x7f3a8724b910>,
  <matplotlib.patches.Wedge at 0x7f3a87207790>,
  <matplotlib.patches.Wedge at 0x7f3a8713fc10>,
  <matplotlib.patches.Wedge at 0x7f3a877c5d90>,
  <matplotlib.patches.Wedge at 0x7f3a8711fcd0>,
  <matplotlib.patches.Wedge at 0x7f3a87125390>,
  <matplotlib.patches.Wedge at 0x7f3a8712a0d0>,
  <matplotlib.patches.Wedge at 0x7f3a870f5ed0>,
  <matplotlib.patches.Wedge at 0x7f3a8a036e50>,
  <matplotlib.patches.Wedge at 0x7f3a8a036f90>,
  <matplotlib.patches.Wedge at 0x7f3a8a036c10>],
 [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')])
```



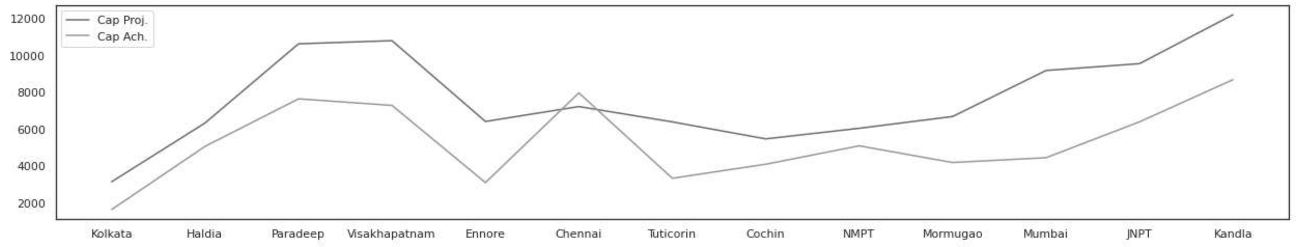
```
plt.pie(df.Traffic_Achieved,labels=df.Port,colors=colors,startangle=135)
```

```
([<matplotlib.patches.Wedge at 0x7f3a8a164f10>,
  <matplotlib.patches.Wedge at 0x7f3a872a12d0>,
  <matplotlib.patches.Wedge at 0x7f3a872a1fd0>,
  <matplotlib.patches.Wedge at 0x7f3a872a1d50>,
  <matplotlib.patches.Wedge at 0x7f3a872990d0>,
  <matplotlib.patches.Wedge at 0x7f3a870bbc50>,
  <matplotlib.patches.Wedge at 0x7f3a8727e150>,
  <matplotlib.patches.Wedge at 0x7f3a8727ef50>,
  <matplotlib.patches.Wedge at 0x7f3a8728f790>,
  <matplotlib.patches.Wedge at 0x7f3a87299290>,
  <matplotlib.patches.Wedge at 0x7f3a8728f990>,
  <matplotlib.patches.Wedge at 0x7f3a87245050>,
  <matplotlib.patches.Wedge at 0x7f3a87284f10>],
 [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')])
```

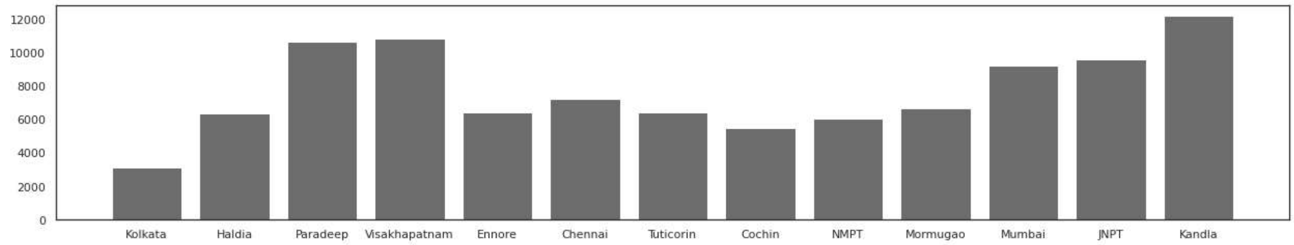


```
plt.plot(df.Port,df.Total_Capacity_Projected,label='Cap Proj.')
plt.plot(df.Port,df.Total_Capacity_Achieved,label='Cap Ach.')
plt.legend()
```

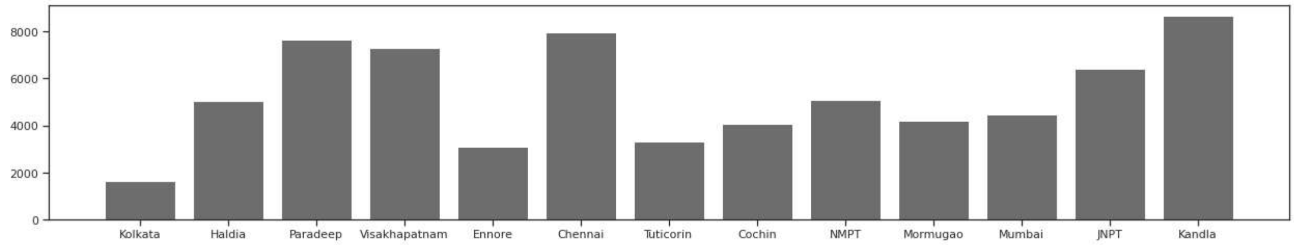
```
<matplotlib.legend.Legend at 0x7f3a872b81d0>
```



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

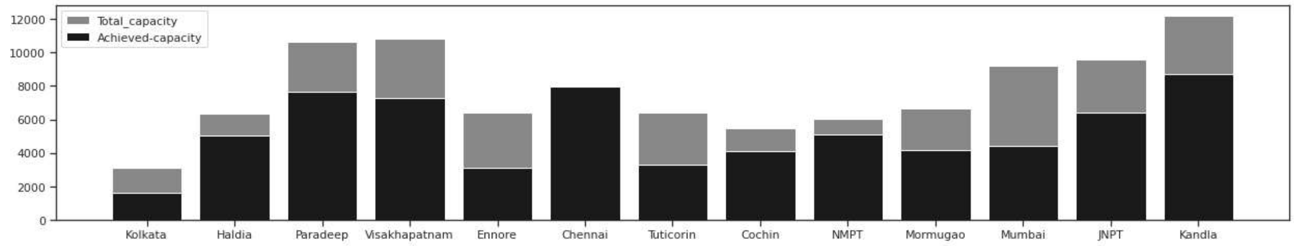


```
plt.bar(df.Port,df.Total_Capacity_Achieved)
sns.set_style('ticks')
```



```
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 0x7f3a86ddb5d0>
```



✓ 0s completed at 3:12 PM ● ✕