In [355]:

```
# IMPORT LIBRARIES
import numpy  as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [356]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\23_Vande Bharat - 23_Vande Bharat.csv")
a
```

Out[356]:

	Sr. No.	Train Name	Train Number	Originating City	Originating Station	Terminal City
0	1	New Delhi - Varanasi Vande Bharat Express	22435/22436	Delhi	New Delhi	Varanasi
1	2	New Delhi - Shri Mata Vaishno Devi Katra Vande...	22439/22440	Delhi	New Delhi	Katra
2	3	Mumbai Central - Gandhinagar Capital Vande Bha...	20901/20902	Mumbai	Mumbai Central	Gandhinagar
3	4	New Delhi - Amb Andaura Vande Bharat Express	22447/22448	Delhi	New Delhi	Andaura
4	5	MGR Chennai Central - Mysuru Vande Bharat Express	20607/20608	Chennai	Chennai Central	Mysuru
5	6	Bilaspur - Nagpur Vande Bharat Express	20825/20826	Bilaspur, Chhattisgarh	Bilaspur Junction	Nagpur
6	7	Howrah - New Jalpaiguri Vande Bharat Express	22301/22302	Kolkata	Howrah Junction	Siliguri
7	8	Visakhapatnam - Secunderabad Vande Bharat Express	20833/20834	Visakhapatnam	Visakhapatnam Junction	Hyderabad
8	9	Mumbai CSMT - Solapur Vande Bharat Express	22225/22226	Mumbai	Chhatrapati Shivaji Terminus	Solapur
9	10	Mumbai CSMT - Sainagar Shirdi Vande Bharat Exp...	22223/22224	Mumbai	Chhatrapati Shivaji Terminus	Shirdi
10	11	Rani Kamalapati (Habibganj) - Hazrat Nizamuddi...	20171/20172	Bhopal	Habibganj (Rani Kamalapati)	Delhi
11	12	Secunderabad - Tirupati Vande Bharat Express	20701/20702	Hyderabad	Secunderabad Junction	Tirupati
12	13	MGR Chennai Central - Coimbatore Vande Bharat ...	20643/20644	Chennai	Chennai Central	Coimbatore
13	14	Delhi Cantonment - Ajmer Vande Bharat Express	20977/20978	Delhi	Delhi Cantonment	Ajmer
14	15	Kasaragod - Thiruvananthapuram Vande Bharat Ex...	20633/20634	Kasaragod	Kasaragod	Thiruvananthapuram
15	16	Howrah - Puri Vande Bharat Express	22895/22896	Kolkata	Howrah Junction	Puri

	Sr. No.	Train Name	Train Number	Originating City	Originating Station	Terminal City
16	17	Anand Vihar Terminal - Dehradun Vande Bharat E...	22457/22458	Delhi	Anand Vihar Terminal	Dehradun
17	18	New Jalpaiguri - Guwahati Vande Bharat Express	22227/22228	Siliguri	New Jalpaiguri Junction	Guwahati
18	19	Mumbai CSMT - Madgaon Vande Bharat Express	22229/22230	Mumbai	Chhatrapati Shivaji Terminus	Madgaon
19	19	Mumbai CSMT - Madgaon Vande Bharat Express	22229/22230	Mumbai	Chhatrapati Shivaji Terminus	Madgaon
20	20	Patna - Ranchi Vande Bharat Express	22349/22350	Patna	Patna Junction	Ranchi
21	21	KSR Bengaluru - Dharwad Vande Bharat Express	20661/20662	Bangalore	Bangalore City	Hubbali - Dharwad
22	22	Rani Kamalapati (Habibganj) - Jabalpur Vande B...	20173/20174	Bhopal	Habibganj (Rani Kamalapati)	Jabalpur
23	23	Indore - Bhopal Vande Bharat Express	20911/20912	Indore	Indore Junction	Bhopal
24	24	Jodhpur - Sabarmati (Ahmedabad) Vande Bharat E...	12461/12462	Jodhpur	Jodhpur Junction	Ahmedabad
25	25	Gorakhpur - Lucknow Charbagh Vande Bharat Express	22549/22550	Gorakhpur	Gorakhpur Junction	Charbagh

In [357]:

```
a=a.head(10)
a
```

Out[357]:

	Sr. No.	Train Name	Train Number	Originating City	Originating Station	Terminal City	Terminal Station
0	1	New Delhi - Varanasi Vande Bharat Express	22435/22436	Delhi	New Delhi	Varanasi	Varanasi Junction
1	2	New Delhi - Shri Mata Vaishno Devi Katra Vande...	22439/22440	Delhi	New Delhi	Katra	Shri Mata Vaishno Devi Katra
2	3	Mumbai Central - Gandhinagar Capital Vande Bha...	20901/20902	Mumbai	Mumbai Central	Gandhinagar	Gandhinagar Capital
3	4	New Delhi - Amb Andaura Vande Bharat Express	22447/22448	Delhi	New Delhi	Andaura	Amb Andaura
4	5	MGR Chennai Central - Mysuru Vande Bharat Express	20607/20608	Chennai	Chennai Central	Mysuru	Mysore Junction
5	6	Bilaspur - Nagpur Vande Bharat Express	20825/20826	Bilaspur, Chhattisgarh	Bilaspur Junction	Nagpur	Nagpur Junction
6	7	Howrah - New Jalpaiguri Vande Bharat Express	22301/22302	Kolkata	Howrah Junction	Siliguri	New Jalpaiguri Junction
7	8	Visakhapatnam - Secunderabad Vande Bharat Express	20833/20834	Visakhapatnam	Visakhapatnam Junction	Hyderabad	Secunderabad Junction
8	9	Mumbai CSMT - Solapur Vande Bharat Express	22225/22226	Mumbai	Chhatrapati Shivaji Terminus	Solapur	Solapur
9	10	Mumbai CSMT - Sainagar Shirdi Vande Bharat Exp...	22223/22224	Mumbai	Chhatrapati Shivaji Terminus	Shirdi	Sainagar Shirdi

In [358]:

```
# to find
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Sr. No.              10 non-null     int64
 1   Train Name           10 non-null     object
 2   Train Number         10 non-null     object
 3   Originating City     10 non-null     object
 4   Originating Station  10 non-null     object
 5   Terminal City        10 non-null     object
 6   Terminal Station     10 non-null     object
 7   Operator             10 non-null     object
 8   No. of Cars          10 non-null     int64
 9   Frequency            10 non-null     object
 10  Distance             10 non-null     object
 11  Travel Time          10 non-null     object
 12  Speed                10 non-null     object
 13  Average Speed        10 non-null     object
 14  Inauguration         10 non-null     object
 15  Average occupancy    10 non-null     object
dtypes: int64(2), object(14)
memory usage: 1.4+ KB
```

In [359]:

```
# to display summary of statastic
a.describe()
```

Out[359]:

	Sr. No.	No. of Cars
count	10.00000	10.000000
mean	5.50000	15.200000
std	3.02765	2.529822
min	1.00000	8.000000
25%	3.25000	16.000000
50%	5.50000	16.000000
75%	7.75000	16.000000
max	10.00000	16.000000

In [360]:

```
# to display colum heading
a.columns
```

Out[360]:

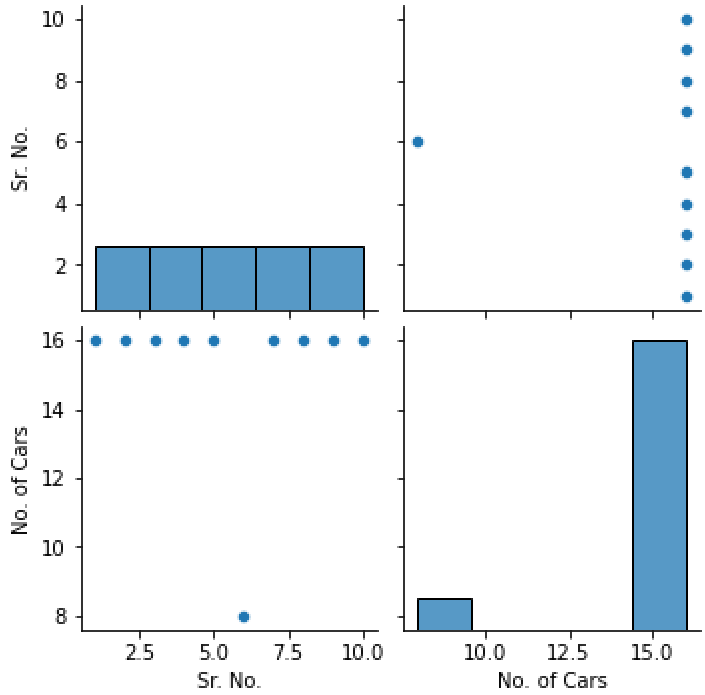
```
Index(['Sr. No.', 'Train Name', 'Train Number', 'Originating City',
       'Originating Station', 'Terminal City', 'Terminal Station', 'Operat
or',
       'No. of Cars', 'Frequency', 'Distance', 'Travel Time', 'Speed',
       'Average Speed', 'Inauguration', 'Average occupancy'],
      dtype='object')
```

In [361]:

```
sns.pairplot(a)
```

Out[361]:

```
<seaborn.axisgrid.PairGrid at 0x243fd936490>
```

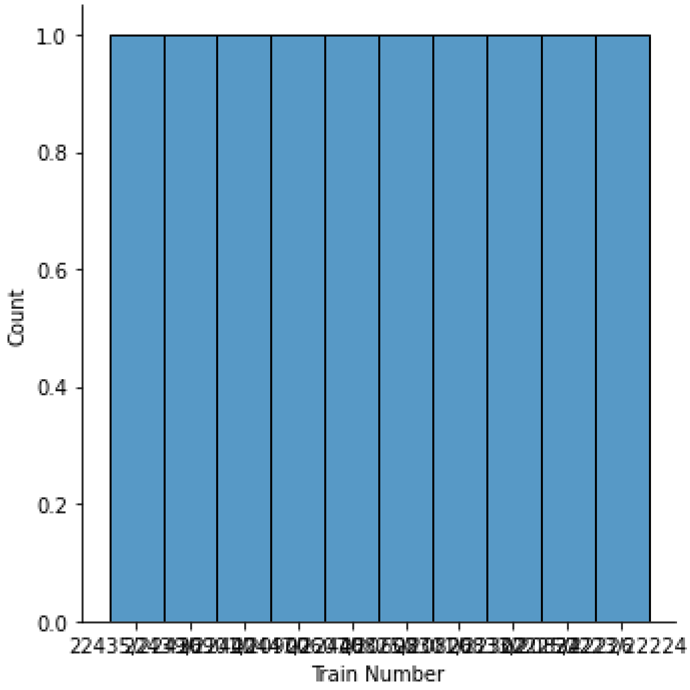


In [363]:

```
sns.displot(a["Train Number"])
```

Out[363]:

```
<seaborn.axisgrid.FacetGrid at 0x243fdabd550>
```



In [364]:

```
b=a[['Sr. No.', 'Train Name', 'Train Number', 'Originating City',
     'Originating Station', 'Terminal City', 'Terminal Station', 'Operator',
     'No. of Cars']]
b
```

Out[364]:

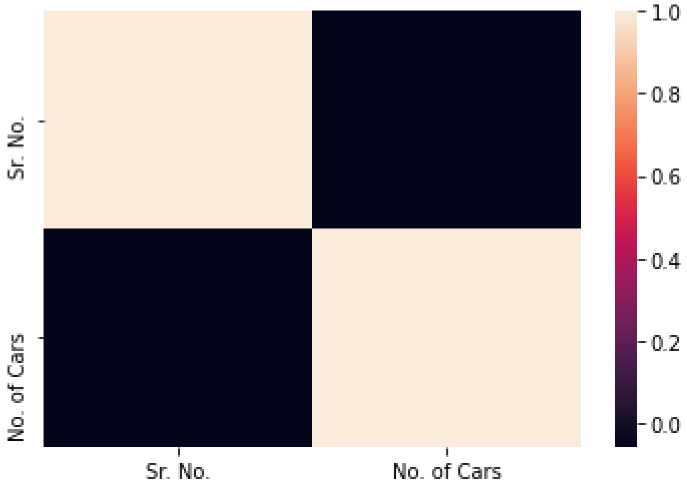
	Sr. No.	Train Name	Train Number	Originating City	Originating Station	Terminal City	Terminal Station
0	1	New Delhi - Varanasi Vande Bharat Express	22435/22436	Delhi	New Delhi	Varanasi	Varanasi Junction
1	2	New Delhi - Shri Mata Vaishno Devi Katra Vande...	22439/22440	Delhi	New Delhi	Katra	Shri Mata Vaishno Devi Katra
2	3	Mumbai Central - Gandhinagar Capital Vande Bha...	20901/20902	Mumbai	Mumbai Central	Gandhinagar	Gandhinagar Capital
3	4	New Delhi - Amb Andaura Vande Bharat Express	22447/22448	Delhi	New Delhi	Andaura	Amb Andaura
4	5	MGR Chennai Central - Mysuru Vande Bharat Express	20607/20608	Chennai	Chennai Central	Mysuru	Mysore Junction
5	6	Bilaspur - Nagpur Vande Bharat Express	20825/20826	Bilaspur, Chhattisgarh	Bilaspur Junction	Nagpur	Nagpur Junction
6	7	Howrah - New Jalpaiguri Vande Bharat Express	22301/22302	Kolkata	Howrah Junction	Siliguri	New Jalpaiguri Junction
7	8	Visakhapatnam - Secunderabad Vande Bharat Express	20833/20834	Visakhapatnam	Visakhapatnam Junction	Hyderabad	Secunderabad Junction
8	9	Mumbai CSMT - Solapur Vande Bharat Express	22225/22226	Mumbai	Chhatrapati Shivaji Terminus	Solapur	Solapur
9	10	Mumbai CSMT - Sainagar Shirdi Vande Bharat Exp...	22223/22224	Mumbai	Chhatrapati Shivaji Terminus	Shirdi	Sainagar Shirdi

In [365]:

```
sns.heatmap(b.corr())
```

Out[365]:

```
<AxesSubplot:>
```



In [367]:

```
x=a[['Sr. No.','No. of Cars']]
y=a['No. of Cars']
```

In [368]:

```
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3)
```

In [369]:

```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[369]:

```
LinearRegression()
```

In [370]:

```
lr.intercept_
```

Out[370]:

```
16.0
```

In [371]:

```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[371]:

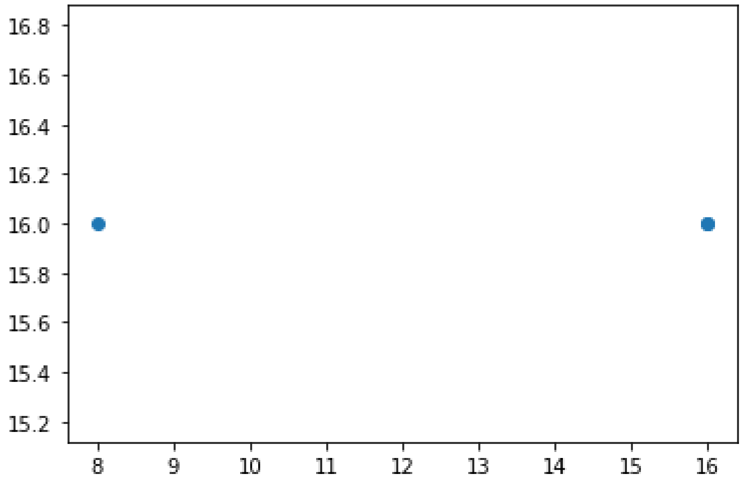
	Co-efficient
Sr. No.	0.0
No. of Cars	0.0

In [372]:

```
prediction = lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[372]:

```
<matplotlib.collections.PathCollection at 0x243fec9cd60>
```



In [373]:

```
lr.score(x_test,y_test)
```

Out[373]:

```
-0.5
```

In [374]:

```
lr.score(x_train,y_train)
```

Out[374]:

```
1.0
```

In [375]:

```
from sklearn.linear_model import Ridge,Lasso
```

In [376]:

```
rr=Ridge(alpha=10)
rr.fit(x_test,y_test)
```

Out[376]:

```
Ridge(alpha=10)
```

In [377]:

```
rr.score(x_test,y_test)
```

Out[377]:

```
0.9633881330309901
```

In [378]:

```
la=Lasso(alpha=10)
la.fit(x_test,y_test)
```

Out[378]:

```
Lasso(alpha=10)
```

In [379]:

```
la.score(x_test,y_test)
```

Out[379]:

```
0.5056152343749997
```

In [380]:

```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_coordinat
e_descent.py:530: ConvergenceWarning: Objective did not converge. You migh
t want to increase the number of iterations. Duality gap: 0.0, tolerance:
0.0
  model = cd_fast.enet_coordinate_descent(
```

Out[380]:

```
ElasticNet()
```

In [381]:

```
en.coef_
```

Out[381]:

```
array([0., 0.])
```

In [382]:

```
en.intercept_
```

Out[382]:

```
16.0
```

In [383]:

```
prediction=en.predict(x_test)
prediction
```

Out[383]:

```
array([16., 16., 16.])
```

In [384]:

```
en.score(x_test,y_test)
```

Out[384]:

```
-0.5
```

EVALUATION METRICS

In [385]:

```
from sklearn import metrics
```

In [386]:

```
print("Mean Absolute Error:",metrics.mean_absolute_error(y_test,prediction))
```

```
Mean Absolute Error: 2.6666666666666665
```

In [387]:

```
print("Mean Squared Error",metrics.mean_squared_error(y_test,prediction))
```

```
Mean Squared Error 21.333333333333332
```

In [388]:

```
print("Root Mean Squared Error",np.sqrt(metrics.mean_squared_error(y_test,prediction)))
```

```
Root Mean Squared Error 4.618802153517006
```

MODEL SAVING

In [389]:

```
import pickle
```

In [390]:

```
filename='prediction'
pickle.dump(lr,open(filename,'wb'))
```

In [391]:

```
import pandas as pd
import pickle
```

In [392]:

```
filename='prediction'
model=pickle.load(open(filename,'rb'))
```

In [395]:

```
real=[[10,20],[13,23]]
result=model.predict(real)
result
```

Out[395]:

```
array([16., 16.])
```

In [ ]: