Vande Bharat Express: A Study of 25 Active Trains in Indian Railways - July 2023

#ABOUT DATASET Dataset contains information about Train 18(Official known as Vande Bharat), its services, top speed, avg speed, Railway Zone etc.You will also find Duration, Date of Inauguration, Train Numbers and more to help with data. Note: - CSMT - Madgaon Vande Bharats are essentialy one train, with 2 different timing depending on the season i.e Monsoon or Non-Monsoon

Importing libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.graph_objs as go
```

importing dataset

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

In [3]:
```
df.head()
```

Out[3]:

	Sr. No.	Train Name	Train Number	Originating City	Originating Station	Terminal City	Terminal Station	Operator	No. of Cars	Frequenc
0	1	New Delhi - Varanasi Vande Bharat Express	22435/22436	Delhi	New Delhi	Varanasi	Varanasi Junction	NR	16	Exce Thursda
1	2	New Delhi - Shri Mata Vaishno Devi Katra Vande...	22439/22440	Delhi	New Delhi	Katra	Shri Mata Vaishno Devi Katra	NR	16	Exce Tuesda
2	3	Mumbai Central - Gandhinagar Capital Vande Bha...	20901/20902	Mumbai	Mumbai Central	Gandhinagar	Gandhinagar Capital	WR	16	Exce Wednesda
3	4	New Delhi - Amb Andaura Vande Bharat Express	22447/22448	Delhi	New Delhi	Andaura	Amb Andaura	NR	16	Exce Frida
4	5	MGR Chennai Central - Mysuru Vande Bharat Express	20607/20608	Chennai	Chennai Central	Mysuru	Mysore Junction	SR	16	Exce Wednesda

In [4]:
```
df.columns
```

Out[4]:
```
Index(['Sr. No.', 'Train Name', 'Train Number', 'Originating City',
       'Originating Station', 'Terminal City', 'Terminal Station', 'Operator',
       'No. of Cars', 'Frequency', 'Distance', 'Travel Time', 'Speed',
       'Average Speed', 'Inauguration', 'Average occupancy'],
      dtype='object')
```

In [5]:
```
df.shape
```

Out[5]:
```
(26, 16)
```

In [6]:
```
df.info()
```

Loading [MathJax]/extensions/Safe.js

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26 entries, 0 to 25
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   Sr. No.              26 non-null     int64 
 1   Train Name           26 non-null     object
 2   Train Number         26 non-null     object
 3   Originating City     26 non-null     object
 4   Originating Station  26 non-null     object
 5   Terminal City        26 non-null     object
 6   Terminal Station     26 non-null     object
 7   Operator             26 non-null     object
 8   No. of Cars          26 non-null     int64 
 9   Frequency            26 non-null     object
 10  Distance             26 non-null     object
 11  Travel Time          26 non-null     object
 12  Speed                26 non-null     object
 13  Average Speed        26 non-null     object
 14  Inauguration         26 non-null     object
 15  Average occupancy    26 non-null     object
dtypes: int64(2), object(14)
memory usage: 3.4+ KB
```

Checking for duplicated values

In [7]:
```
df.duplicated().sum()
```

Out[7]:
```
0
```

Checking for missing values

In [8]:
```
df.isna().sum()
```

Out[8]:
```
Sr. No.                0
Train Name             0
Train Number           0
Originating City       0
Originating Station    0
Terminal City          0
Terminal Station       0
Operator               0
No. of Cars            0
Frequency              0
Distance               0
Travel Time            0
Speed                  0
Average Speed          0
Inauguration           0
Average occupancy      0
dtype: int64
```

No missing values

In [9]:
```
df.drop('Sr. No.',axis=1,inplace=True)
```

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

Loading [MathJax]/extensions/Safe.js

Out[10]:

```
Train Name             25
Train Number           25
Originating City       15
Originating Station    18
Terminal City          25
Terminal Station       25
Operator               15
No. of Cars             2
Frequency               9
Distance               24
Travel Time            20
Speed                   5
Average Speed          17
Inauguration           18
Average occupancy      22
dtype: int64
```

In [11]:

```
for i in df:
  print(i)
  print('\n')
  print(df[i].value_counts())
  print('-'*100)
```

Loading [MathJax]/extensions/Safe.js

```
Train Name


Mumbai CSMT - Madgaon Vande Bharat Express                              2
New Delhi - Varanasi Vande Bharat Express                               1
Delhi Cantonment - Ajmer Vande Bharat Express                           1
Jodhpur - Sabarmati (Ahmedabad) Vande Bharat Express                    1
Indore - Bhopal Vande Bharat Express                                    1
Rani Kamalapati (Habibganj) - Jabalpur Vande Bharat Express             1
KSR Bengaluru - Dharwad Vande Bharat Express                            1
Patna - Ranchi Vande Bharat Express                                     1
New Jalpaiguri - Guwahati Vande Bharat Express                          1
Anand Vihar Terminal - Dehradun Vande Bharat Express                    1
Howrah - Puri Vande Bharat Express                                      1
Kasaragod - Thiruvananthapuram Vande Bharat Express                     1
MGR Chennai Central - Coimbatore Vande Bharat Express                   1
New Delhi - Shri Mata Vaishno Devi Katra Vande Bharat Express           1
Secunderabad - Tirupati Vande Bharat Express                            1
Rani Kamalapati (Habibganj) - Hazrat Nizamuddin Vande Bharat Express    1
Mumbai CSMT - Sainagar Shirdi Vande Bharat Express                      1
Mumbai CSMT - Solapur Vande Bharat Express                              1
Visakhapatnam - Secunderabad Vande Bharat Express                       1
Howrah - New Jalpaiguri Vande Bharat Express                            1
Bilaspur - Nagpur Vande Bharat Express                                  1
MGR Chennai Central - Mysuru Vande Bharat Express                       1
New Delhi - Amb Andaura Vande Bharat Express                            1
Mumbai Central - Gandhinagar Capital Vande Bharat Express               1
Gorakhpur - Lucknow Charbagh Vande Bharat Express                       1
Name: Train Name, dtype: int64
----------------------------------------------------------------------------------------
------------
Train Number


22229/22230    2
22435/22436    1
20977/20978    1
12461/12462    1
20911/20912    1
20173/20174    1
20661/20662    1
22349/22350    1
22227/22228    1
22457/22458    1
22895/22896    1
20633/20634    1
20643/20644    1
22439/22440    1
20701/20702    1
20171/20172    1
22223/22224    1
22225/22226    1
20833/20834    1
22301/22302    1
20825/20826    1
20607/20608    1
22447/22448    1
20901/20902    1
22549/22550    1
Name: Train Number, dtype: int64
----------------------------------------------------------------------------------------
------------
Originating City
```

```
Delhi                     5
Mumbai                    5
Chennai                   2
Kolkata                   2
Bhopal                    2
Bilaspur, Chhattisgarh    1
Visakhapatnam             1
Hyderabad                 1
Kasaragod                 1
Siliguri                  1
Patna                     1
Bangalore                 1
Indore                    1
Jodhpur                   1
Gorakhpur                 1
Name: Originating City, dtype: int64
----------------------------------------------------------------------------------------
------------
Originating Station


Chhatrapati Shivaji Terminus    4
New Delhi                       3
Chennai Central                 2
Howrah Junction                 2
Habibganj (Rani Kamalapati)     2
Anand Vihar Terminal            1
Jodhpur Junction                1
Indore Junction                 1
Bangalore City                  1
Patna Junction                  1
New Jalpaiguri Junction         1
Delhi Cantonment                1
Kasaragod                       1
Mumbai Central                  1
Secunderabad Junction           1
Visakhapatnam Junction          1
Bilaspur Junction               1
Gorakhpur Junction              1
Name: Originating Station, dtype: int64
----------------------------------------------------------------------------------------
------------
Terminal City


Madgaon               2
Varanasi              1
Ajmer                 1
Ahmedabad             1
Bhopal                1
Jabalpur              1
Hubbali - Dharwad     1
Ranchi                1
Guwahati              1
Dehradun              1
Puri                  1
Thiruvananthapuram    1
Coimbatore            1
Katra                 1
Tirupati              1
Delhi                 1
Shirdi                1
Solapur               1
```

```
Hyderabad             1
Siliguri              1
Nagpur                1
Mysuru                1
Andaura               1
Gandhinagar           1
Charbagh              1
Name: Terminal City, dtype: int64
----------------------------------------------------------------------------------------
------------
Terminal Station


Madgaon Junction                2
Varanasi Junction               1
Ajmer Junction                  1
Sabarmati Junction              1
Bhopal Junction                 1
Jabalpur Junction               1
Dharwad                         1
Ranchi Junction                 1
Guwahati                        1
Dehradun Terminal               1
Puri                            1
Thiruvananthapuram Central      1
Coimbatore Junction             1
Shri Mata Vaishno Devi Katra    1
Tirupati                        1
Hazrat Nizamuddin               1
Sainagar Shirdi                 1
Solapur                         1
Secunderabad Junction           1
New Jalpaiguri Junction         1
Nagpur Junction                 1
Mysore Junction                 1
Amb Andaura                     1
Gandhinagar Capital             1
Lucknow Charbagh                1
Name: Terminal Station, dtype: int64
----------------------------------------------------------------------------------------
------------
Operator


NR      4
CR      4
SR      3
WR      2
WCR     2
NWR     2
SECR    1
ER      1
ECoR    1
SCR     1
SER     1
NFR     1
ECR     1
SWR     1
NER     1
Name: Operator, dtype: int64
----------------------------------------------------------------------------------------
------------
No. of Cars
```

Loading [MathJax]/extensions/Safe.js

```
16    16
8     10
Name: No. of Cars, dtype: int64
----------------------------------------------------------------------------------------
------------
Frequency


Except Tuesdays                                                                      8
Except Wednesdays                                                                    6
Except Thursdays                                                                     3
Except Saturdays                                                                     3
Except Sundays                                                                       2
Except Fridays                                                                       1
Except Wednesdays (22225) , Except Thursdays (22226)                                 1
Except Fridays\n(Non-Monsoon)                                                        1
Monday, Wednesday, Friday (22229)\nTuesday, Thursday, Saturday (22230)\n(Monsoon)    1
Name: Frequency, dtype: int64
----------------------------------------------------------------------------------------
------------
Distance


412 km (256 mi)    2
586 km (364 mi)    2
759 km (472 mi)    1
587 km (365 mi)    1
449 km (279 mi)    1
250 km (160 mi)    1
337 km (209 mi)    1
490 km (300 mi)    1
379 km (235 mi)    1
407 km (253 mi)    1
304 km (189 mi)    1
500 km (310 mi)    1
428 km (266 mi)    1
655 km (407 mi)    1
495 km (308 mi)    1
661 km (411 mi)    1
702 km (436 mi)    1
339 km (211 mi)    1
452 km (281 mi)    1
698 km (434 mi)    1
565 km (351 mi)    1
496 km (308 mi)    1
522 km (324 mi)    1
296 km (184 mi)    1
Name: Distance, dtype: int64
----------------------------------------------------------------------------------------
------------
Travel Time


06h 25m                   3
08h 00m                   2
06h 30m                   2
05h 30m                   2
07h 30m                   2
04h 45m                   1
06h 10m                   1
03h 05m                   1
04h 40m                   1
06h 00m                   1
```

```
10h 05m\n(Monsoon)        1
07h 45m\n(Non-Monsoon)    1
05h 15m                   1
08h 05m                   1
05h 50m                   1
08h 15m                   1
05h 20m                   1
08h 30m                   1
05h 10m                   1
04h 15m                   1
Name: Travel Time, dtype: int64
----------------------------------------------------------------------------------------
------------
Speed


130 km/h (81 mph)    14
110 km/h (68 mph)     8
120 km/h (75 mph)     2
130 km/h (81 mph)     1
160 km/h (99 mph)     1
Name: Speed, dtype: int64
----------------------------------------------------------------------------------------
------------
Average Speed


82 km/h (51 mph)                   5
73 km/h (45 mph)                   3
79 km/h (49 mph)                   2
75 km/h (47 mph)                   2
64 km/h (40 mph)                   2
95 km/h (59 mph)                   1
74 km/h (46 mph)                   1
76 km/h (47 mph)                   1
63 km/h (39 mph)                   1
57 km/h (35 mph)\n(Monsoon)        1
75 km/h (47 mph)\n(Non-Monsoon)    1
85 km/h (53 mph)                   1
78 km/h (48 mph)                   1
80 km/h (50 mph)                   1
94 km/h (58 mph)                   1
70 km/h (43 mph)                   1
71 km/h (44 mph)                   1
Name: Average Speed, dtype: int64
----------------------------------------------------------------------------------------
------------
Inauguration


6/27/2023     6
7/7/2023      2
4/8/2023      2
2/10/2023     2
5/29/2023     1
5/25/2023     1
5/18/2023     1
4/25/2023     1
4/12/2023     1
2/15/2019     1
10/3/2019     1
1/15/2023     1
12/30/2022    1
12/11/2022    1
```

```
11/11/2022    1
10/13/2022    1
9/30/2022     1
4/1/2023      1
Name: Inauguration, dtype: int64
------------------------------------------------------------------------------------------
------------
Average occupancy


100%    3
70%     2
94%     2
126%    1
177%    1
53%     1
37%     1
44%     1
72%     1
118%    1
91%     1
99%     1
150%    1
114%    1
110%    1
90%     1
93%     1
120%    1
96%     1
75%     1
132%    1
77%     1
Name: Average occupancy, dtype: int64
------------------------------------------------------------------------------------------
------------
```

In [12]:

```
df.columns
```

Out[12]:

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

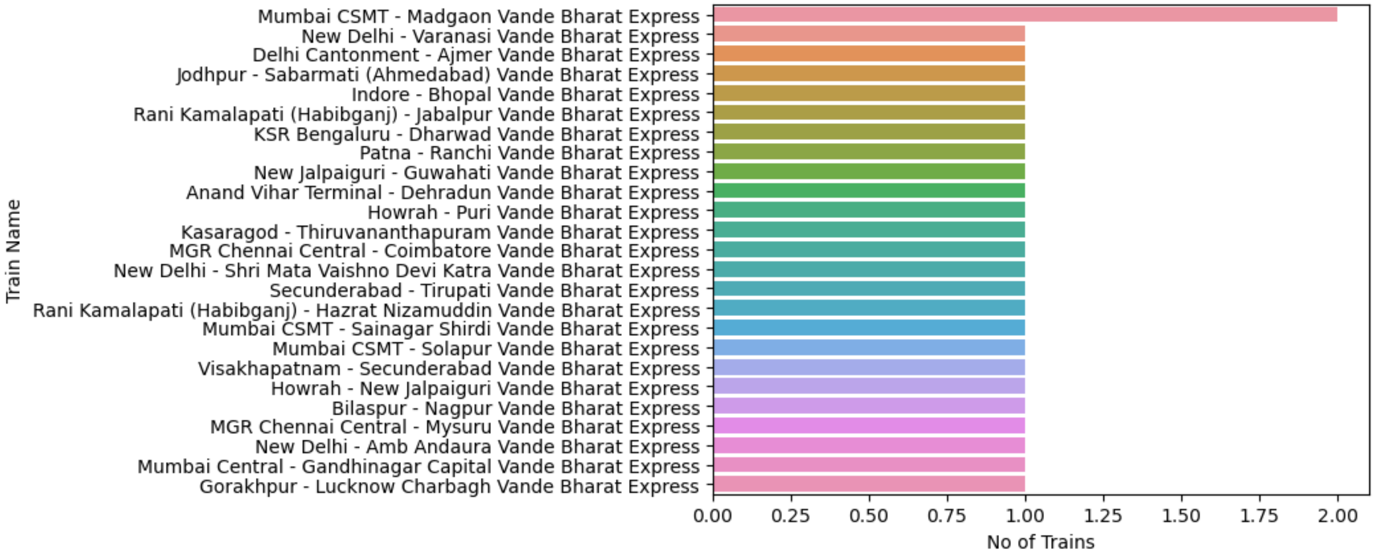
In [13]:

```
sns.barplot(y=df['Train Name'].value_counts().keys(),x=df['Train Name'].value_counts(),d
plt.title("",fontsize=20,fontweight="bold")
plt.ylabel('Train Name')
plt.xlabel('No of Trains')
```

Out[13]:

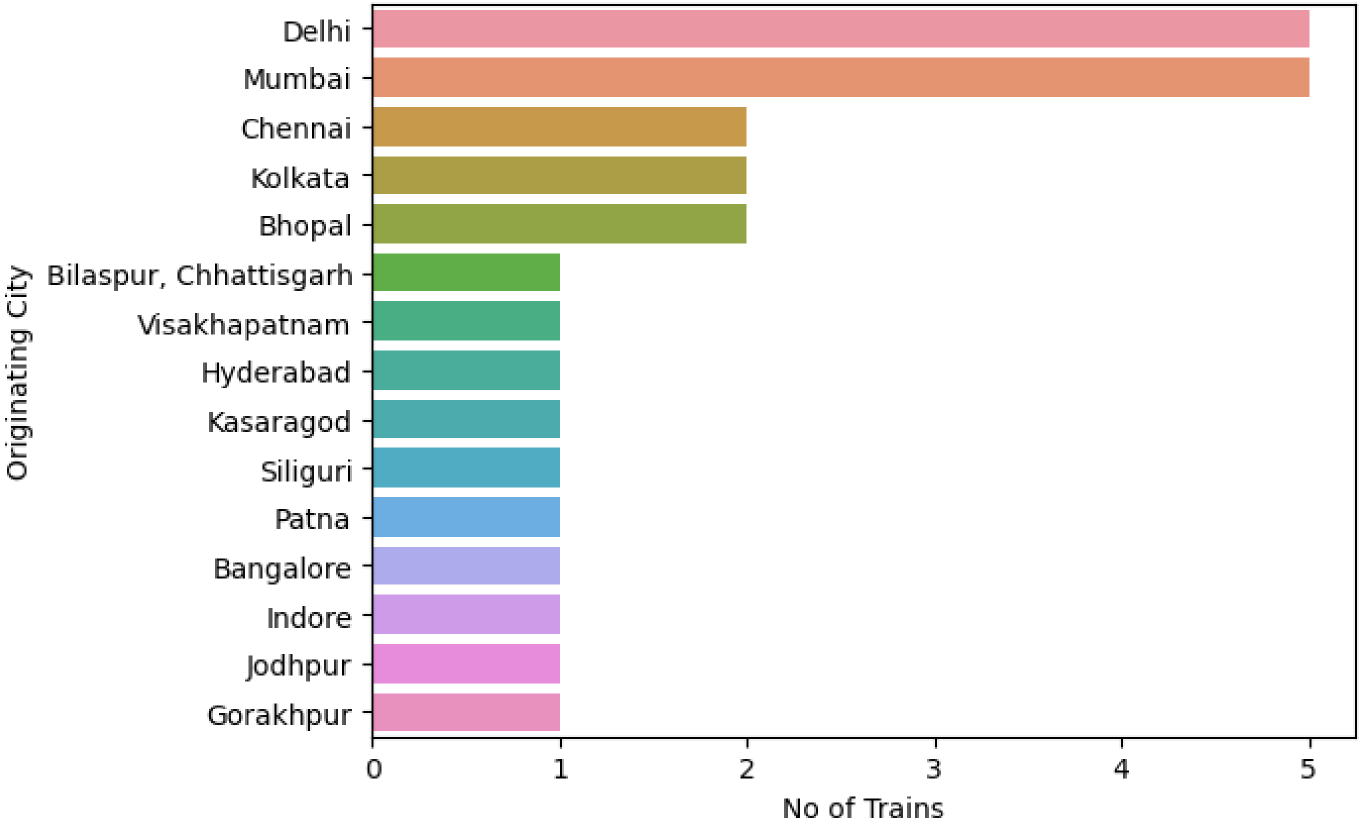
```
Text(0.5, 0, 'No of Trains')
```

Loading [MathJax]/extensions/Safe.js



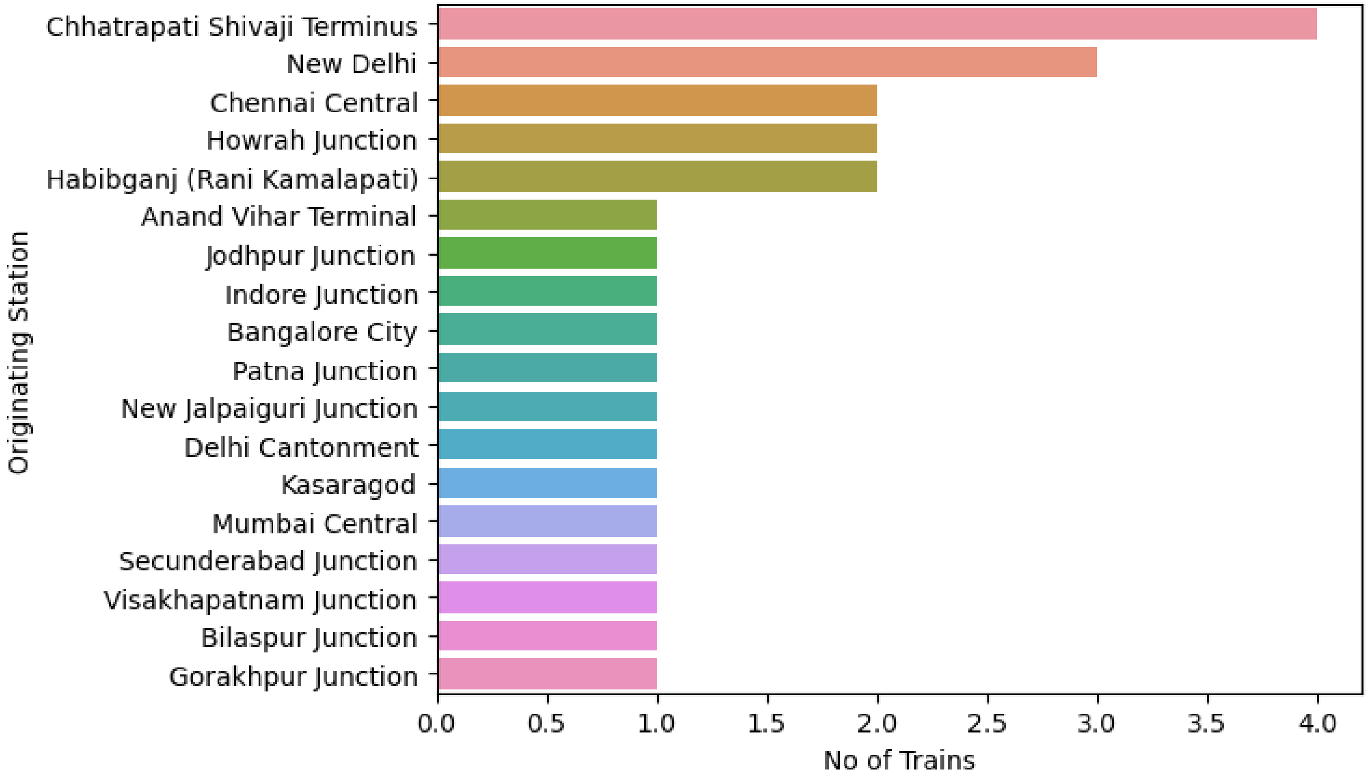
In [14]:
```
sns.barplot(y=df['Originating City'].value_counts().keys(),x=df['Originating City'].valu
plt.ylabel('Originating City')
plt.xlabel('No of Trains')
```

Out[14]:
```
Text(0.5, 0, 'No of Trains')
```



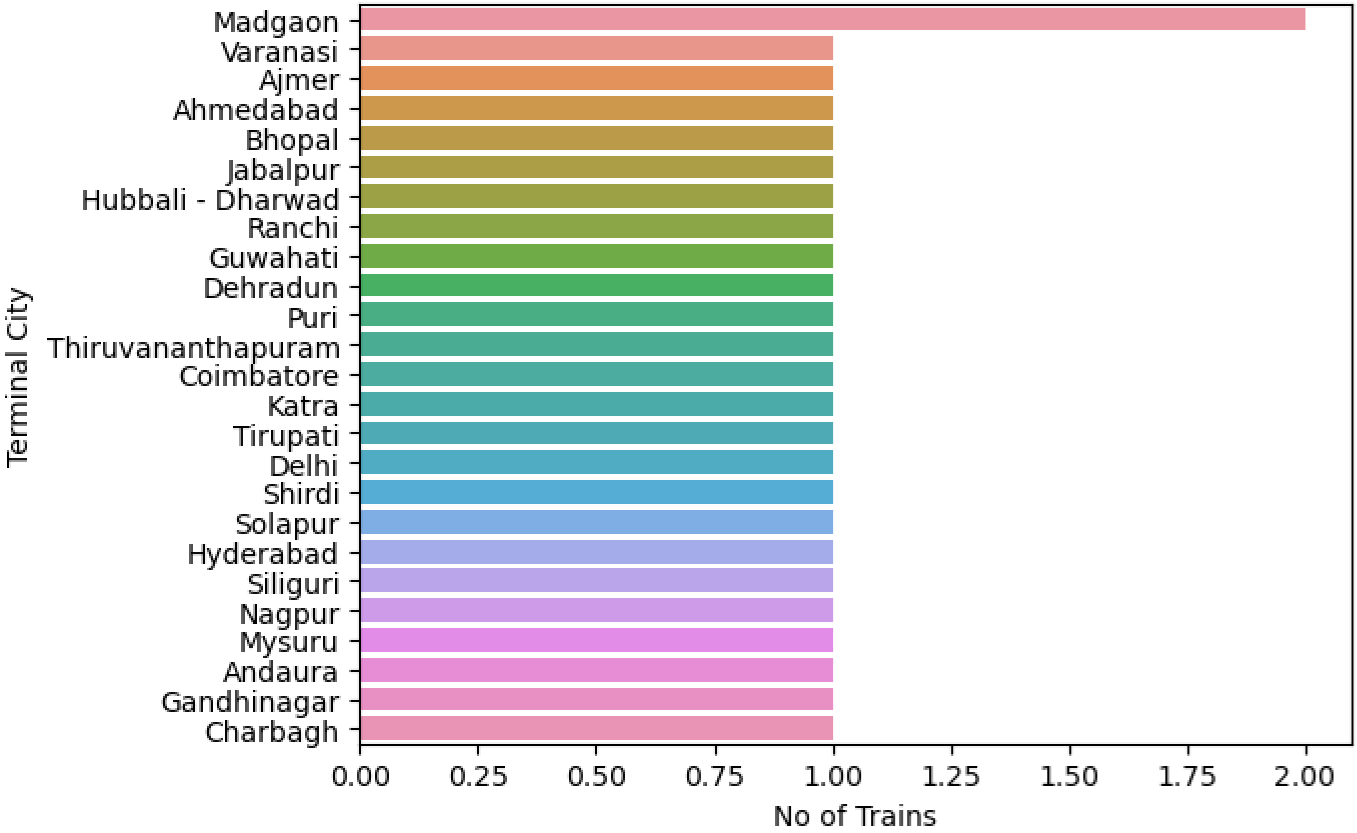
In [15]:
```
sns.barplot(y=df['Originating Station'].value_counts().keys(),x=df['Originating Station'
plt.ylabel('Originating Station')
plt.xlabel('No of Trains')
```

Out[15]:
```
Text(0.5, 0, 'No of Trains')
```



In [16]:
```
sns.barplot(y=df['Terminal City'].value_counts().keys(),x=df['Terminal City'].value_coun
plt.ylabel('Terminal City')
plt.xlabel('No of Trains')
```

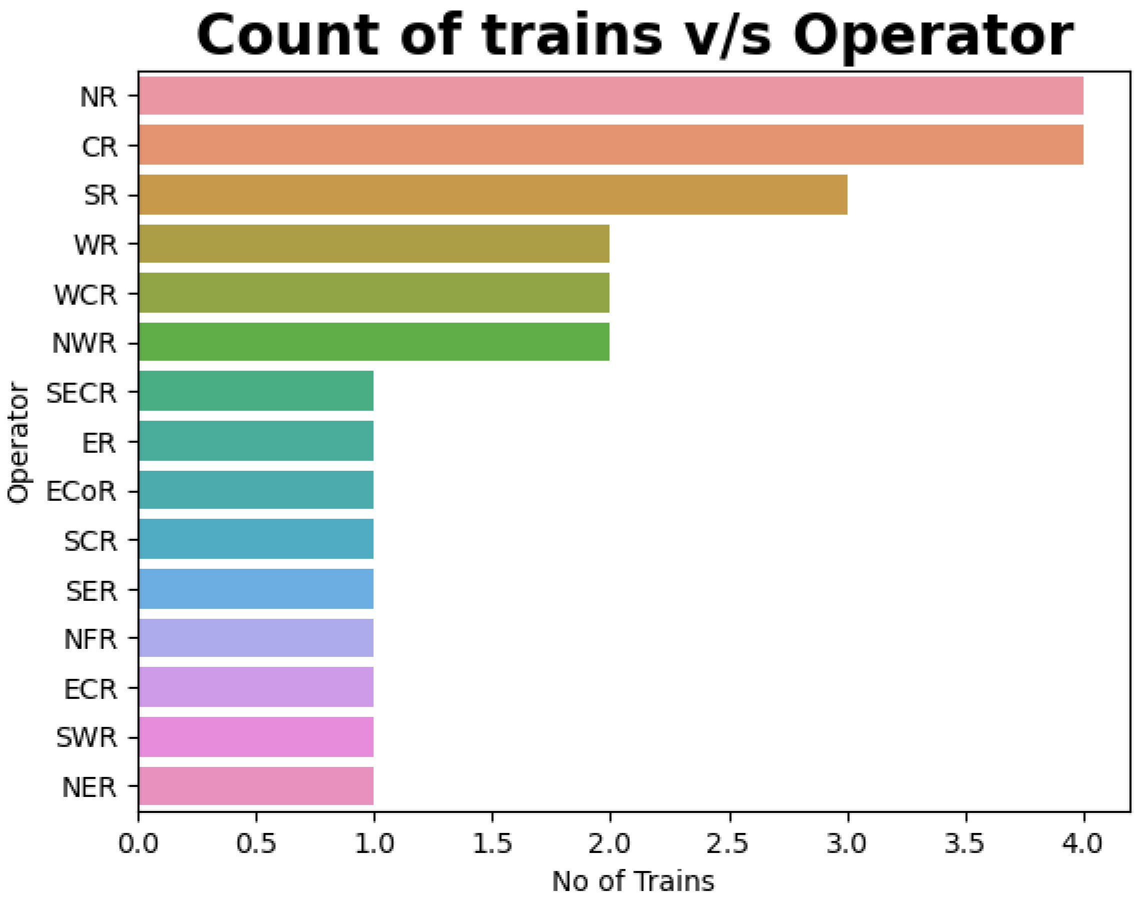
Out[16]:
```
Text(0.5, 0, 'No of Trains')
```



In [17]:
```
sns.barplot(y=df['Operator'].value_counts().keys(),x=df['Operator'].value_counts(),data=
plt.title("Count of trains v/s Operator",fontsize=20,fontweight="bold")
plt.ylabel('Operator')
plt.xlabel('No of Trains')
```

Out[17]:

```
Text(0.5, 0, 'No of Trains')
```



In [18]:

```
df['No. of Cars'].value_counts()
```

Out[18]:

```
16    16
8     10
Name: No. of Cars, dtype: int64
```

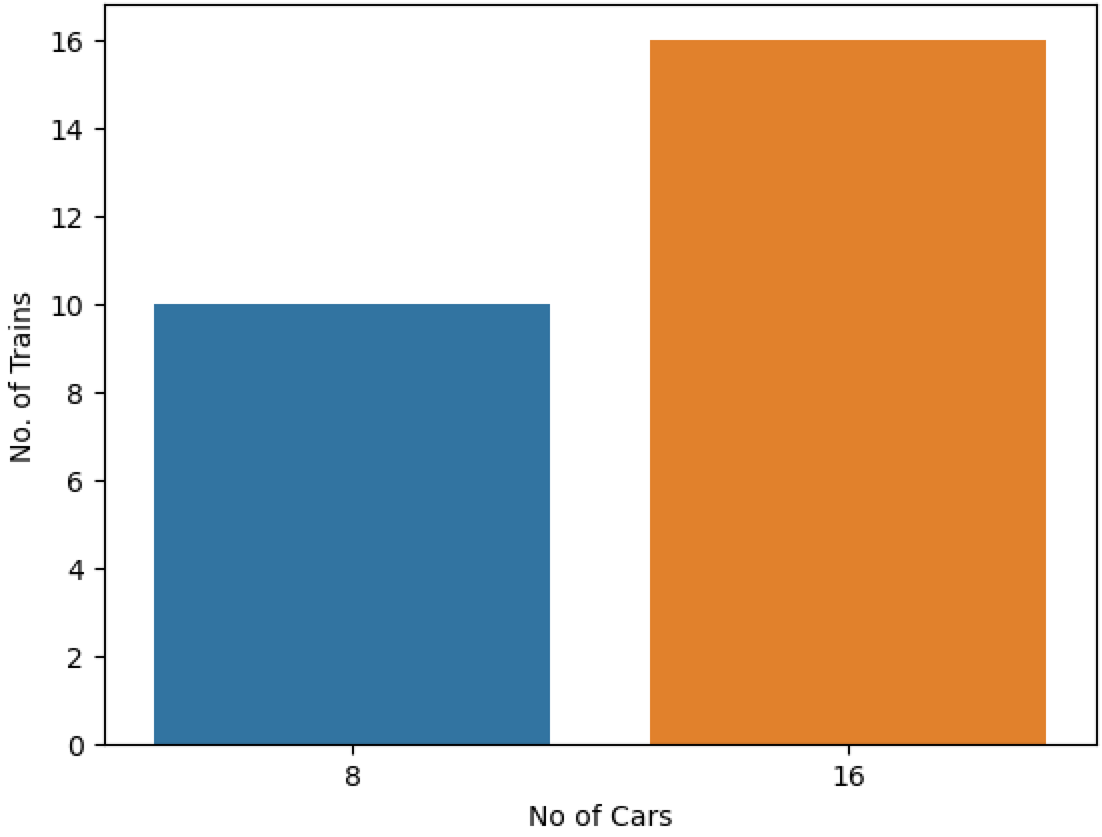
In [19]:

```
sns.barplot(x=df['No. of Cars'].value_counts().keys(),y=df['No. of Cars'].value_counts()
plt.ylabel('No. of Trains')
plt.xlabel('No of Cars')
```

Out[19]:

```
Text(0.5, 0, 'No of Cars')
```

Loading [MathJax]/extensions/Safe.js

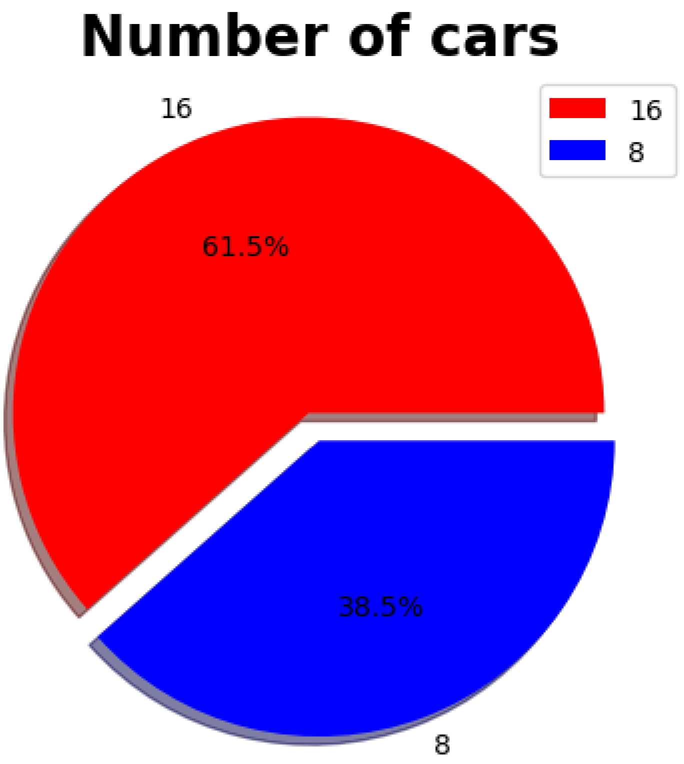


In [20]:

```
plt.pie(df['No. of Cars'].value_counts(),labels=df["No. of Cars"].value_counts().index,e
        autopct='%1.1f%%',colors=["r","b"],shadow=True)
plt.title("Number of cars",fontsize=20,fontweight="bold")
plt.legend()
```

Out[20]:

```
<matplotlib.legend.Legend at 0x1abe8a55750>
```

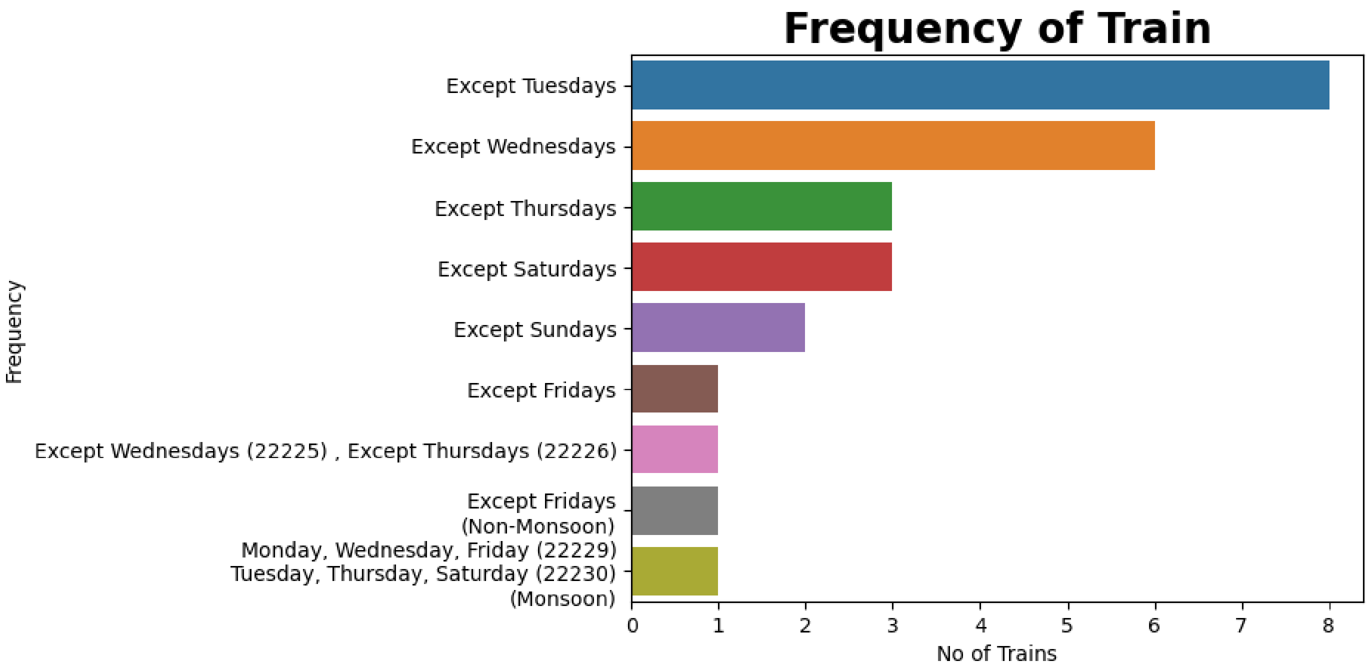


In [21]:

```
sns.barplot(y=df['Frequency'].value_counts().keys(),x=df['Frequency'].value_counts(),dat
plt.title("Frequency of Train",fontsize=20,fontweight="bold")
plt.ylabel('Frequency')
plt.xlabel('No of Trains')
```

Out[21]:

```
Text(0.5, 0, 'No of Trains')
```



In [22]:

```
df['Average occupancy'] = df['Average occupancy'].str.rstrip('%').astype(float)
```

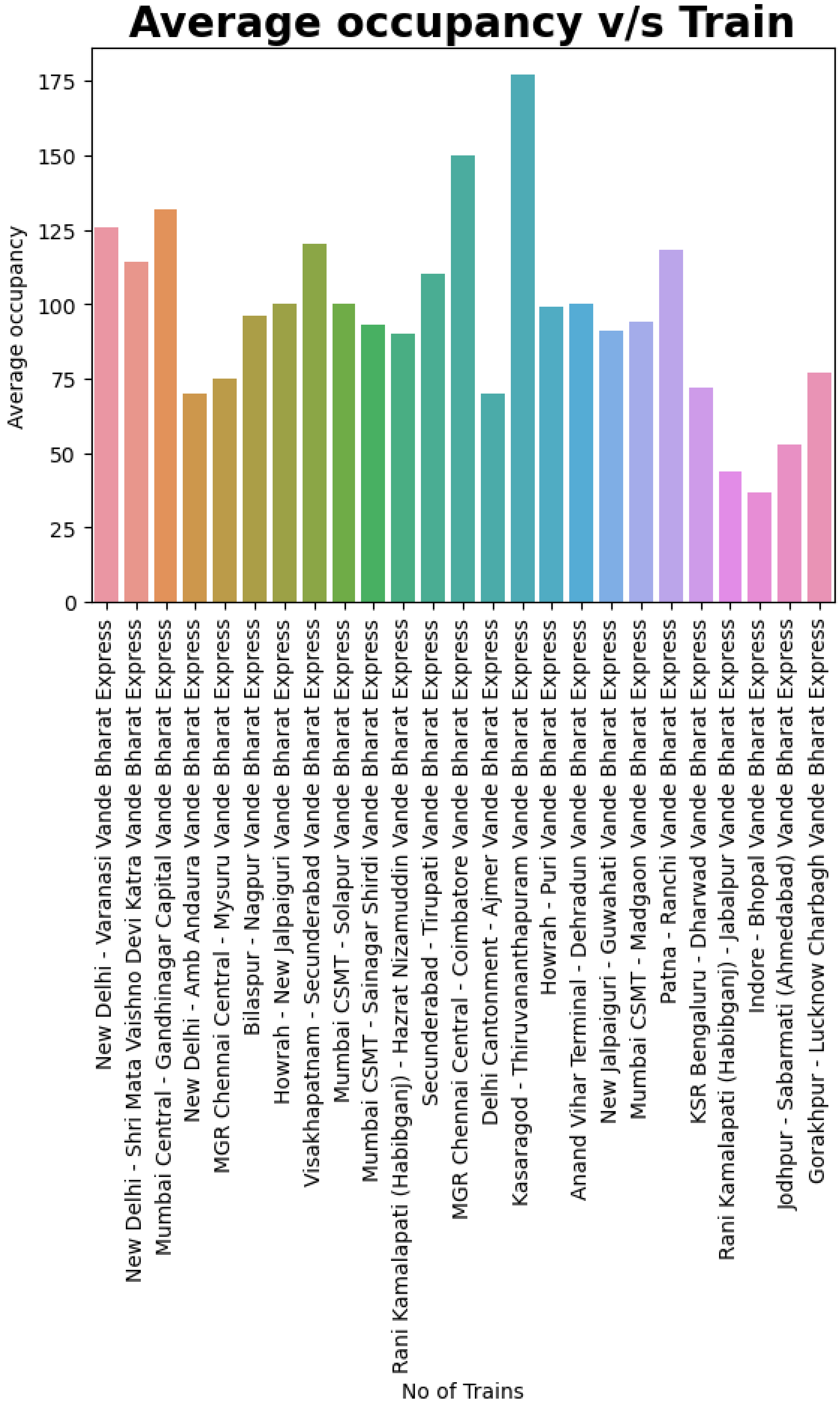
In [23]:

```
sns.barplot(y='Average occupancy',x='Train Name',data=df)
plt.title("Average occupancy v/s Train",fontsize=20,fontweight="bold")
plt.xticks(rotation='vertical')
plt.ylabel('Average occupancy')
plt.xlabel('No of Trains')
```

Out[23]:

```
Text(0.5, 0, 'No of Trains')
```

Loading [MathJax]/extensions/Safe.js



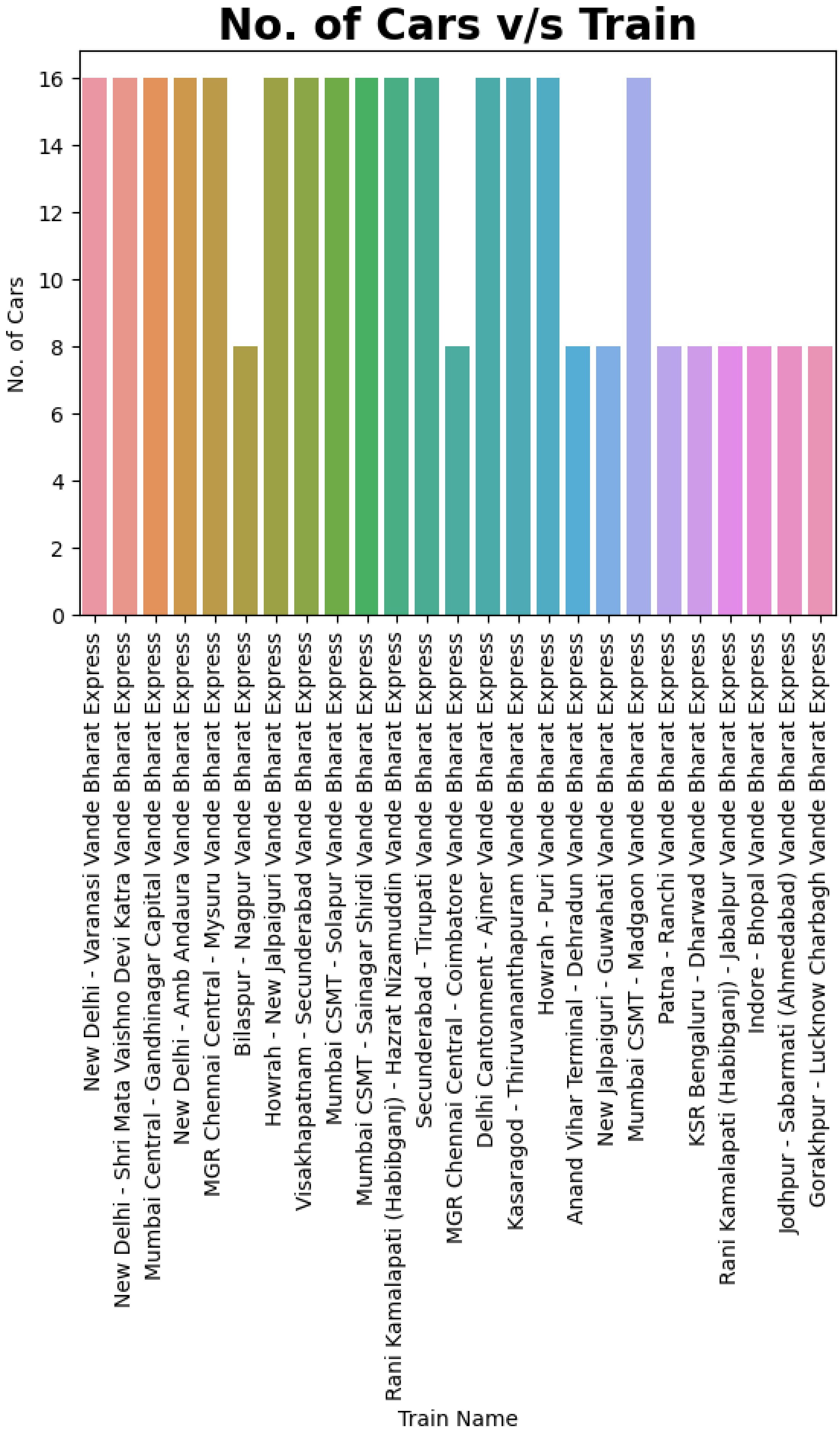
In [24]:

```
sns.barplot(y='No. of Cars',x='Train Name',data=df)
plt.title("No. of Cars v/s Train",fontsize=20,fontweight="bold")
plt.xticks(rotation='vertical')
plt.ylabel('No. of Cars')
                         rain Name')
```

Loading [MathJax]/extensions/Safe.js

Out[24]:

```
Text(0.5, 0, 'Train Name')
```



In [25]:

```
def speed_to_kmph(speed):
    if 'km/h' in speed:
```

Loading [MathJax]/extensions/Safe.js

```
        return float(speed.split()[0])
    elif 'mph' in speed:
        mph_value = float(speed.split()[0])
        return mph_value * 1.60934
    else:
        return None
df['Speed'] = df['Speed'].apply(speed_to_kmph)
df['Average Speed'] = df['Average Speed'].apply(speed_to_kmph)
```

In [26]:

```
df['Inauguration'] = pd.to_datetime(df['Inauguration'])
```

In [27]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26 entries, 0 to 25
Data columns (total 15 columns):
 #   Column               Non-Null Count  Dtype         
---  ------               --------------  -----         
 0   Train Name           26 non-null     object        
 1   Train Number         26 non-null     object        
 2   Originating City     26 non-null     object        
 3   Originating Station  26 non-null     object        
 4   Terminal City        26 non-null     object        
 5   Terminal Station     26 non-null     object        
 6   Operator             26 non-null     object        
 7   No. of Cars          26 non-null     int64         
 8   Frequency            26 non-null     object        
 9   Distance             26 non-null     object        
 10  Travel Time          26 non-null     object        
 11  Speed                26 non-null     float64       
 12  Average Speed        26 non-null     float64       
 13  Inauguration         26 non-null     datetime64[ns]
 14  Average occupancy    26 non-null     float64       
dtypes: datetime64[ns](1), float64(3), int64(1), object(10)
memory usage: 3.2+ KB
```

In [28]:

```
df.head(3)
```

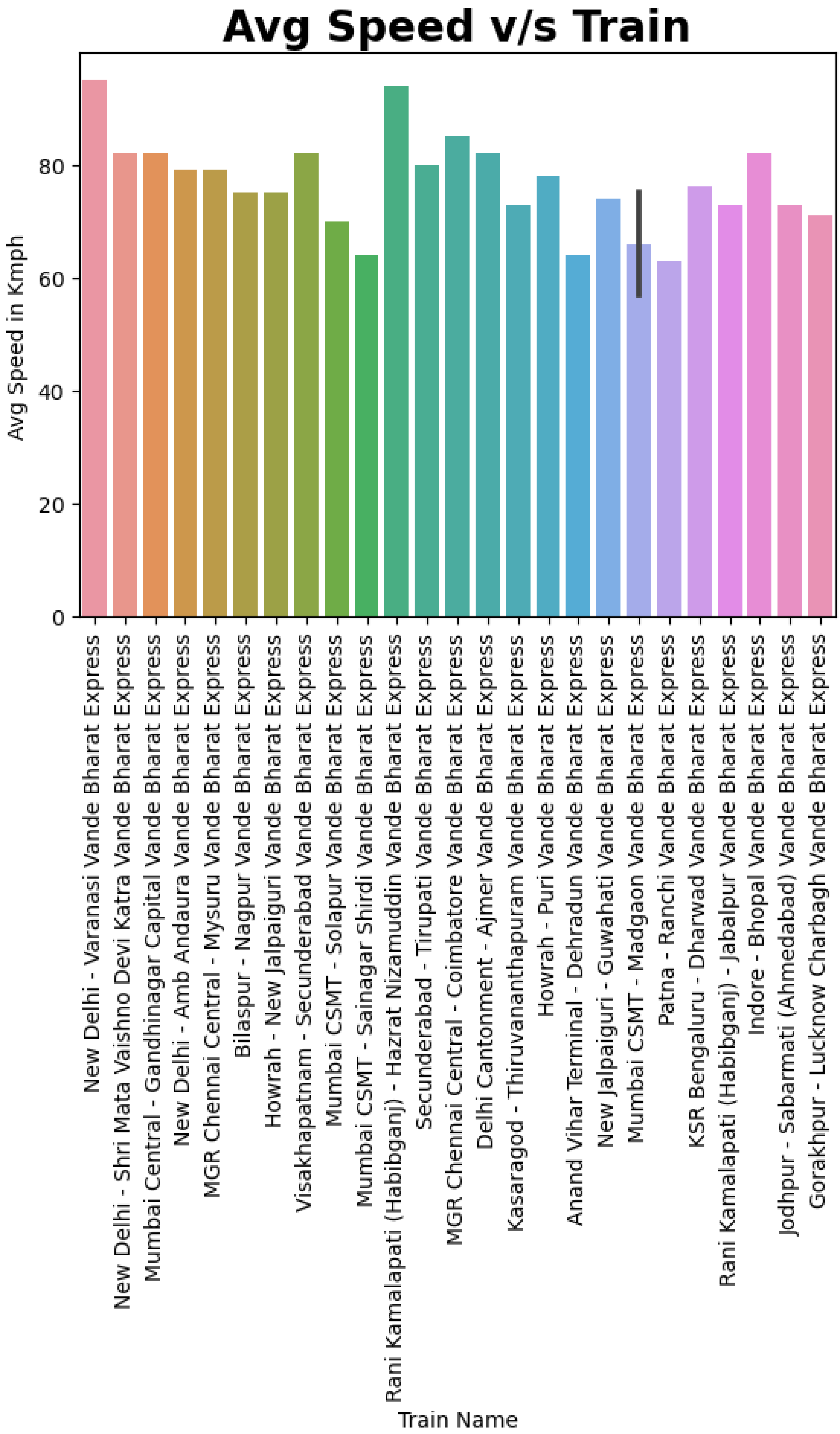
Out[28]:

	Train Name	Train Number	Originating City	Originating Station	Terminal City	Terminal Station	Operator	No. of Cars	Frequency	D
0	New Delhi - Varanasi Vande Bharat Express	22435/22436	Delhi	New Delhi	Varanasi	Varanasi Junction	NR	16	Except Thursdays	(
1	New Delhi - Shri Mata Vaishno Devi Katra Vande...	22439/22440	Delhi	New Delhi	Katra	Shri Mata Vaishno Devi Katra	NR	16	Except Tuesdays	(
2	Mumbai Central - Gandhinagar Capital Vande Bha...	20901/20902	Mumbai	Mumbai Central	Gandhinagar	Gandhinagar Capital	WR	16	Except Wednesdays	(

In [29]:

```
sns.barplot(x='Train Name',y='Average Speed',data=df)
plt.title("Avg Speed v/s Train",fontsize=20,fontweight="bold")
plt.xticks(rotation='vertical')
plt.ylabel('Avg Speed in Kmph')
plt.xlabel('Train Name')
```

```
Out[29]: Text(0.5, 0, 'Train Name')
```



```
In [30]: df['Distance'] = df['Distance'].str.extract(r'(\d+)').astype(float)
```

Loading [MathJax]/extensions/Safe.js

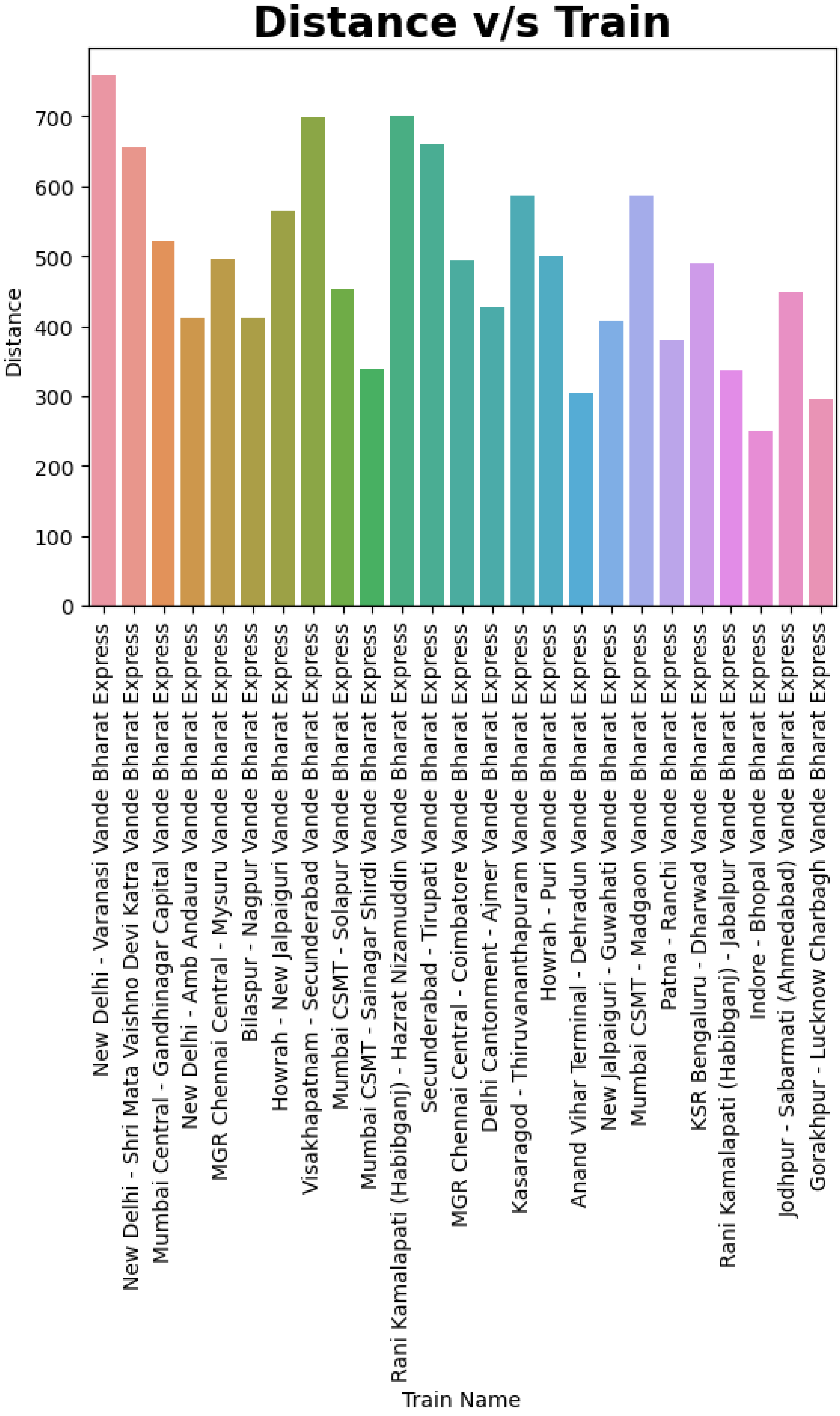
In [31]:

```
sns.barplot(x='Train Name',y='Distance',data=df)
plt.title("Distance v/s Train",fontsize=20,fontweight="bold")
plt.xticks(rotation='vertical')
plt.ylabel('Distance')
plt.xlabel('Train Name')
```

Out[31]:

```
Text(0.5, 0, 'Train Name')
```

Loading [MathJax]/extensions/Safe.js



LATEST TRAIN INAUGURATION

In [32]:

```
df_sorted = df.sort_values(by='Inauguration', ascending=False)
print(df_sorted[['Train Name','Inauguration']])
```

```
                                           Train Name Inauguration
25  Gorakhpur - Lucknow Charbagh Vande Bharat Express   2023-07-07
24  Jodhpur - Sabarmati (Ahmedabad) Vande Bharat E...   2023-07-07
23               Indore - Bhopal Vande Bharat Express   2023-06-27
22  Rani Kamalapati (Habibganj) - Jabalpur Vande B...   2023-06-27
21       KSR Bengaluru - Dharwad Vande Bharat Express   2023-06-27
20                Patna - Ranchi Vande Bharat Express   2023-06-27
19         Mumbai CSMT - Madgaon Vande Bharat Express   2023-06-27
18         Mumbai CSMT - Madgaon Vande Bharat Express   2023-06-27
17     New Jalpaiguri - Guwahati Vande Bharat Express   2023-05-29
16  Anand Vihar Terminal - Dehradun Vande Bharat E...   2023-05-25
15                 Howrah - Puri Vande Bharat Express   2023-05-18
14  Kasaragod - Thiruvananthapuram Vande Bharat Ex...   2023-04-25
13      Delhi Cantonment - Ajmer Vande Bharat Express   2023-04-12
12  MGR Chennai Central - Coimbatore Vande Bharat ...   2023-04-08
11       Secunderabad - Tirupati Vande Bharat Express   2023-04-08
10  Rani Kamalapati (Habibganj) - Hazrat Nizamuddi...   2023-04-01
9   Mumbai CSMT - Sainagar Shirdi Vande Bharat Exp...   2023-02-10
8          Mumbai CSMT - Solapur Vande Bharat Express   2023-02-10
7   Visakhapatnam - Secunderabad Vande Bharat Express   2023-01-15
6        Howrah - New Jalpaiguri Vande Bharat Express   2022-12-30
5              Bilaspur - Nagpur Vande Bharat Express   2022-12-11
4   MGR Chennai Central - Mysuru Vande Bharat Express   2022-11-11
3        New Delhi - Amb Andaura Vande Bharat Express   2022-10-13
2   Mumbai Central - Gandhinagar Capital Vande Bha...   2022-09-30
1   New Delhi - Shri Mata Vaishno Devi Katra Vande...   2019-10-03
0           New Delhi - Varanasi Vande Bharat Express   2019-02-15
```

Loading [MathJax]/extensions/Safe.js