

EDA on Vande Bharat

Performed by Piyush Borhade



Importing required libraries!

In [1]:

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

In [2]:

```
df = pd.read_csv("Vande Bharat.csv")  
df
```

Out[2]:

	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 [3]:

```
df.info()
```

```
<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
```

So our dataset consists of 16 columns in which 14 columns have 'object' as their datatype and the remaining 2 columns have 'int' datatype.

In [4]:

```
df.shape
```

Out[4]:

```
(26, 16)
```

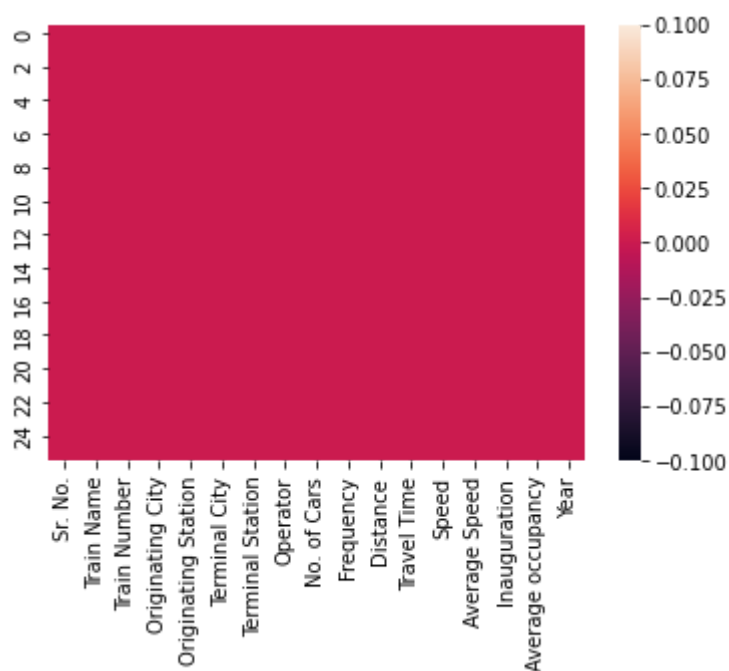
This dataset is from Kaggle and our dataset has 26 rows and 16 columns

In [41]:

```
sns.heatmap(df.isna())
```

Out[41]:

<AxesSubplot:>



In [39]:

```
df.isna().sum()
```

Out[39]:

```
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
Year         0
dtype: int64
```

In [43]:

```
df.duplicated().sum()
```

Out[43]:

0

Conclusion:

We can observe that our dataset does not have duplicate values as well as null value

In [45]:

```
df.describe().round()
```

Out[45]:

	Sr. No.	No. of Cars	Year
count	26.0	26.0	26.0
mean	13.0	13.0	2022.0
std	7.0	4.0	1.0
min	1.0	8.0	2019.0
25%	7.0	8.0	2022.0
50%	14.0	16.0	2023.0
75%	19.0	16.0	2023.0
max	25.0	16.0	2023.0

In [46]:

```
df.corr()
```

Out[46]:

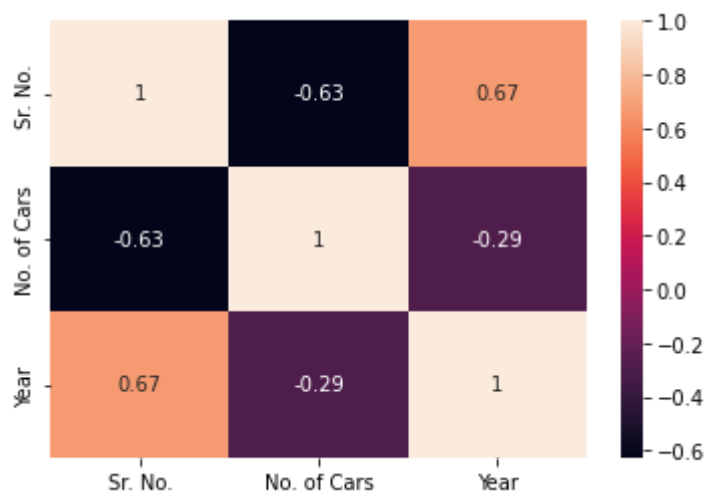
	Sr. No.	No. of Cars	Year
Sr. No.	1.000000	-0.625565	0.669123
No. of Cars	-0.625565	1.000000	-0.291969
Year	0.669123	-0.291969	1.000000

In [47]:

```
cm = df.corr()  
sns.heatmap(cm,annot=True)
```

Out[47]:

<AxesSubplot:>

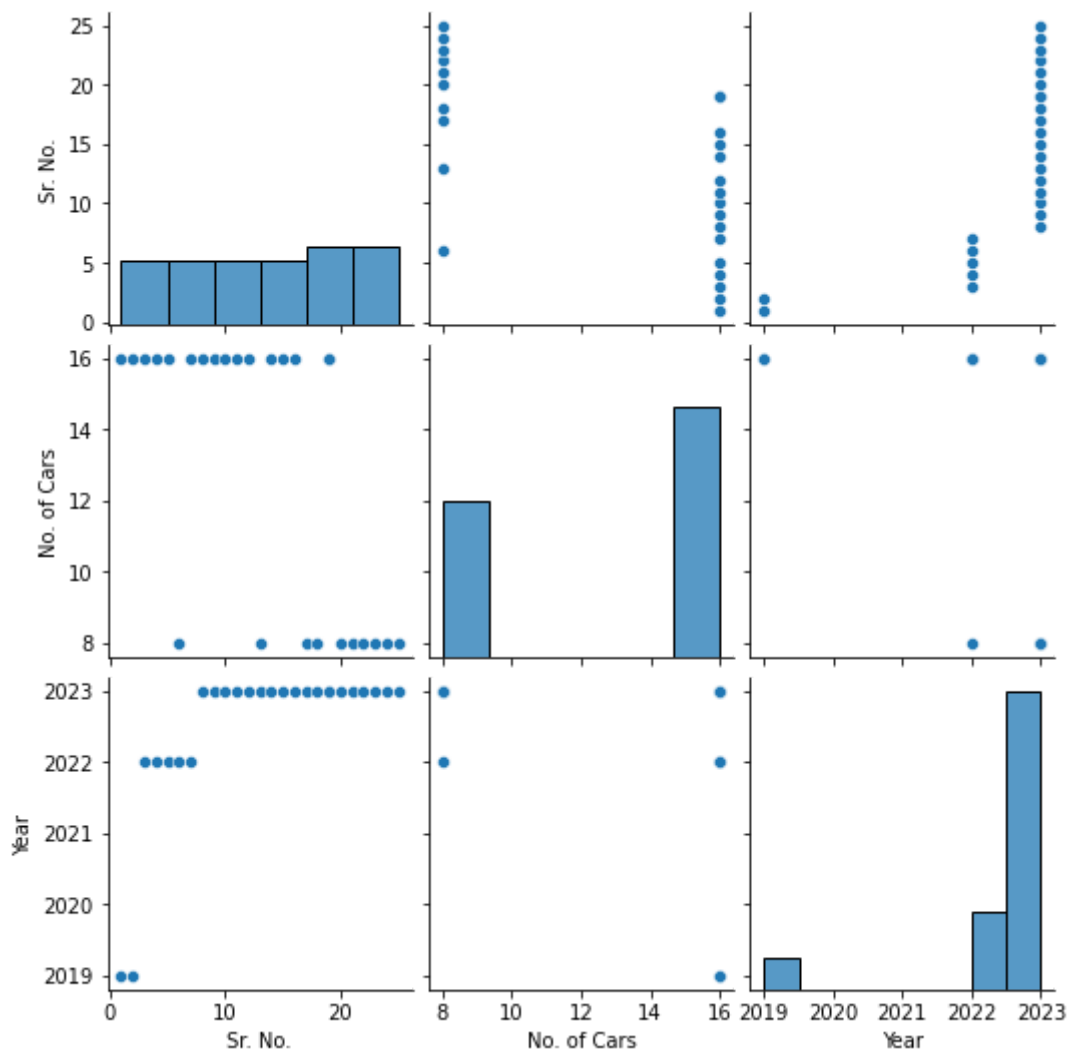


In [50]:

```
sns.pairplot(df)
```

Out[50]:

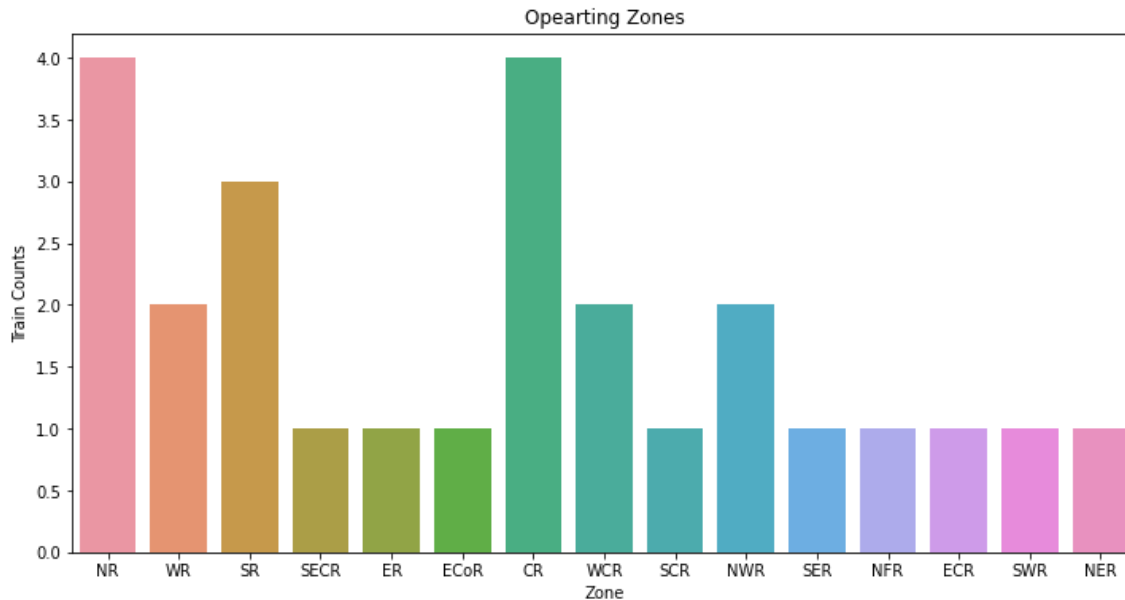
<seaborn.axisgrid.PairGrid at 0x1afbe47a160>



Different Operating Zones

In [11]:

```
plt.figure(figsize=(12,6))
sns.countplot(x='Operator',data=df)
plt.title("Opearting Zones")
plt.xlabel("Zone")
plt.ylabel("Train Counts")
plt.show()
```



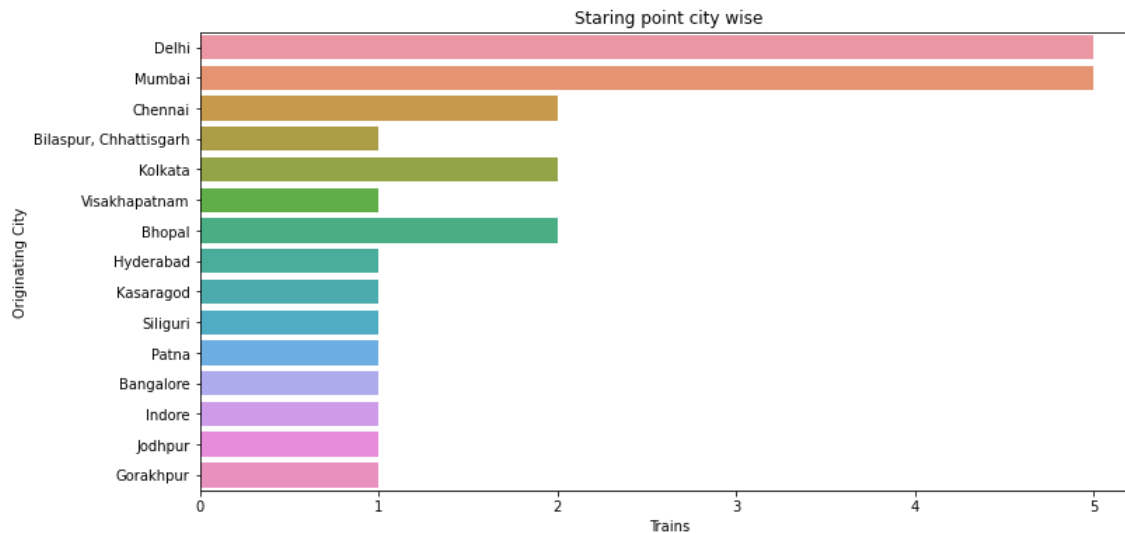
Conclusion:

NR and CR operates highest train whose count is 4 trains
then comes the SR with 3 trains operated
following with WR, WCR and NWR consisting 2 trains
and the reamining operates 1 train.

Vande Bharat Express - Source/Originating Station

In [12]:

```
plt.figure(figsize=(12,6))
sns.countplot(y='Originating City',data=df)
plt.title("Staring point city wise")
plt.ylabel("Originating City")
plt.xlabel("Trains")
plt.show()
```



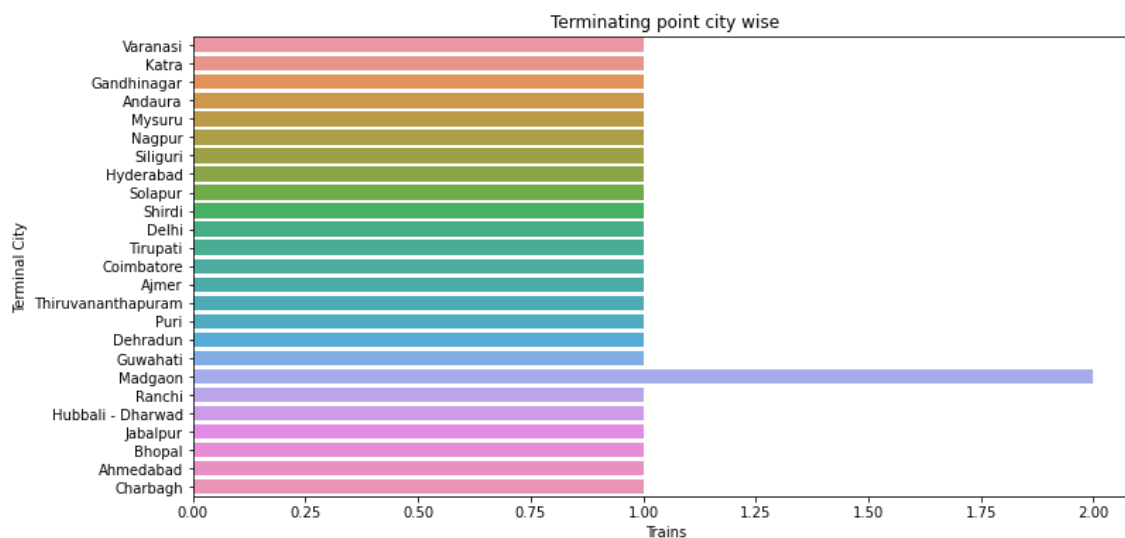
Conclusion:

Delhi,Mumbai has 5 trains originating,
nextly Chennai, Kolkata, Bhopal with 2 trains originating.

Vande Bharat Express - Terminal Cities

In [13]:

```
plt.figure(figsize=(12,6))
sns.countplot(y='Terminal City',data=df)
plt.title("Terminating point city wise")
plt.ylabel("Terminal City")
plt.xlabel("Trains")
plt.show()
```



Conclusion:

Madgaon is the city 2 terminating points which is highest among all the cities

In [14]:

```
df.columns
```

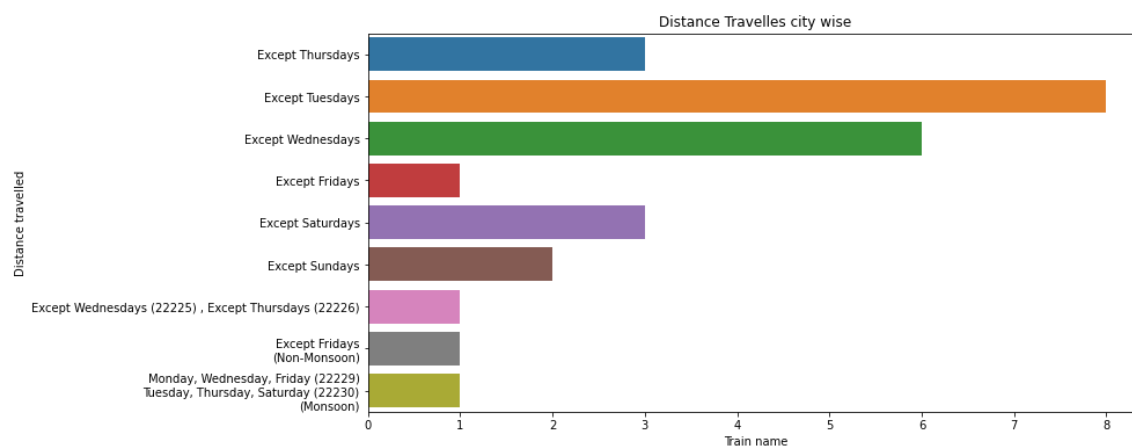
Out[14]:

```
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', 'Year'],
      dtype='object')
```

Frequency of train on days

In [15]:

```
plt.figure(figsize=(12,6))
sns.countplot(y=df['Frequency'])
plt.title("Distance Travelles city wise")
plt.ylabel("Distance travelled")
plt.xlabel("Train name")
plt.show()
```



Conclusion:

Most of the trains are runned 'Except Tuesday' with around 8 trains running then 'Except Wednesday' is running which is around 6

Originating Cities - Zone wise

In [16]:

```
df.columns
```

Out[16]:

```
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', 'Year'],  
      dtype='object')
```

In [20]:

```
df[['Originating City', 'Terminal City', 'Operator', 'Speed']].groupby(['Originating City',
```

Out[20]:

Originating City	Terminal City	Operator	Speed
Bangalore	Hubballi - Dharwad	SWR	130 km/h (81 mph)
1			
Bhopal	Delhi	WCR	160 km/h (99 mph)
1			
	Jabalpur	WCR	110 km/h (68 mph)
1			
Bilaspur, Chhattisgarh	Nagpur	SECR	130 km/h (81 mph)
1			
Chennai	Coimbatore	SR	130 km/h (81 mph)
1			
	Mysuru	SR	130 km/h (81 mph)
1			
Delhi	Ajmer	NWR	110 km/h (68 mph)
1			
	Andaura	NR	130 km/h (81 mph)
1			
	Dehradun	NR	130 km/h (81 mph)
1			
	Katra	NR	130 km/h (81 mph)
1			
	Varanasi	NR	130 km/h (81 mph)
1			
Gorakhpur	Charbagh	NER	110 km/h (68 mph)
1			
Hyderabad	Tirupati	SCR	130 km/h (81 mph)
1			
Indore	Bhopal	WR	110 km/h (68 mph)
1			
Jodhpur	Ahmedabad	NWR	130 km/h (81 mph)
1			
Kasaragod	Thiruvananthapuram	SR	110 km/h (68 mph)
1			
Kolkata	Puri	SER	130 km/h (81 mph)
1			
	Siliguri	ER	130 km/h (81 mph)
1			
Mumbai	Gandhinagar	WR	130 km/h (81 mph)
1			
	Madgaon	CR	120 km/h (75 mph)
2			
	Shirdi	CR	110 km/h (68 mph)
1			
	Solapur	CR	110 km/h (68 mph)
1			
Patna	Ranchi	ECR	130 km/h (81 mph)
1			
Siliguri	Guwahati	NFR	110 km/h (68 mph)
1			
Visakhapatnam	Hyderabad	ECOR	130 km/h (81 mph)
1			

dtype: int64

Conclusion:

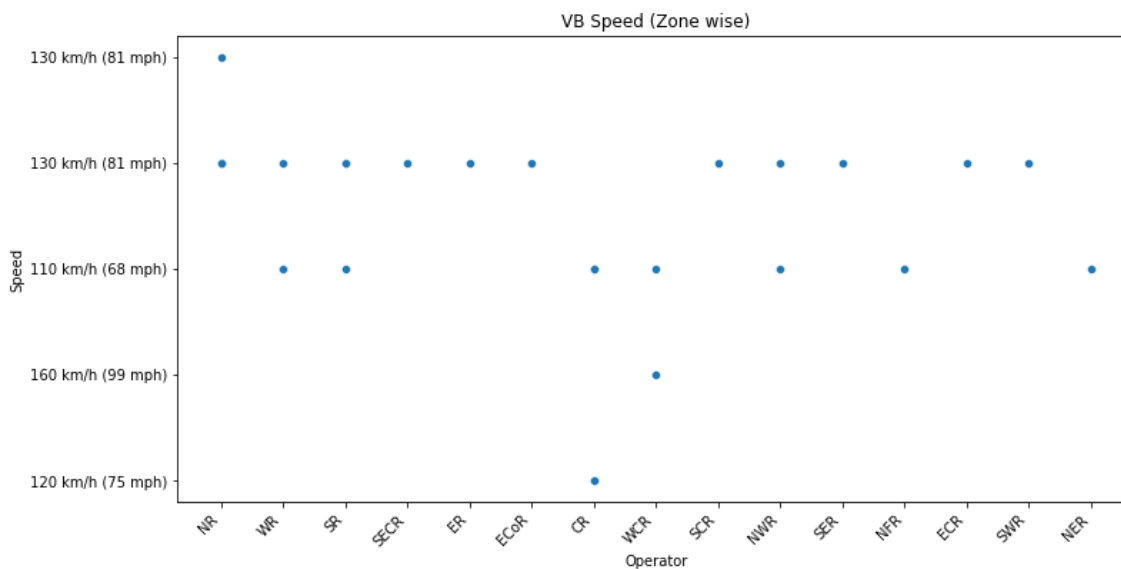
Train originates from Delhi has highest terminal cities (5)

Train originates from Mumbai has Second highest terminal cities (4)

Speed of Vande Bharat with respect to Operator

In [24]:

```
plt.figure(figsize=(12,6))
sns.scatterplot(data=df,x='Operator', y='Speed')
plt.xticks(rotation=45, ha='right')
plt.title('VB Speed (Zone wise)')
plt.show()
```



Conclusion:

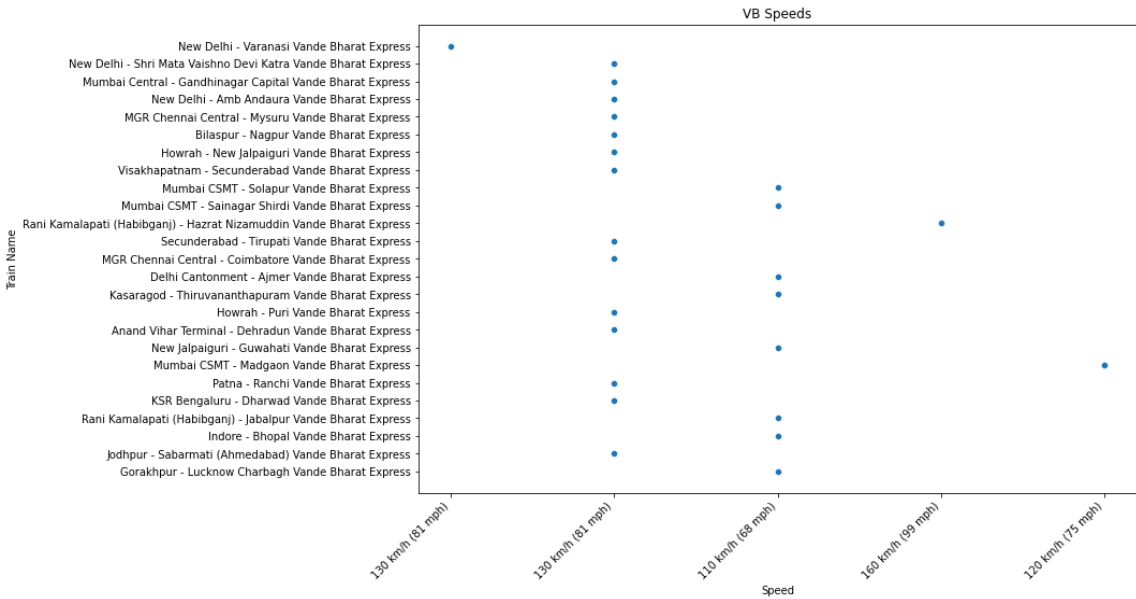
NR operator has most highest speed i.e 130kms/hr

CR operator has least speed among all express i.e 120kms/hr

Vande bharat speed!

In [28]:

```
plt.figure(figsize=(12,8))
sns.scatterplot(data=df,y='Train Name', x='Speed')
plt.xticks(rotation=45, ha='right')
plt.title('VB Speeds')
plt.show()
```



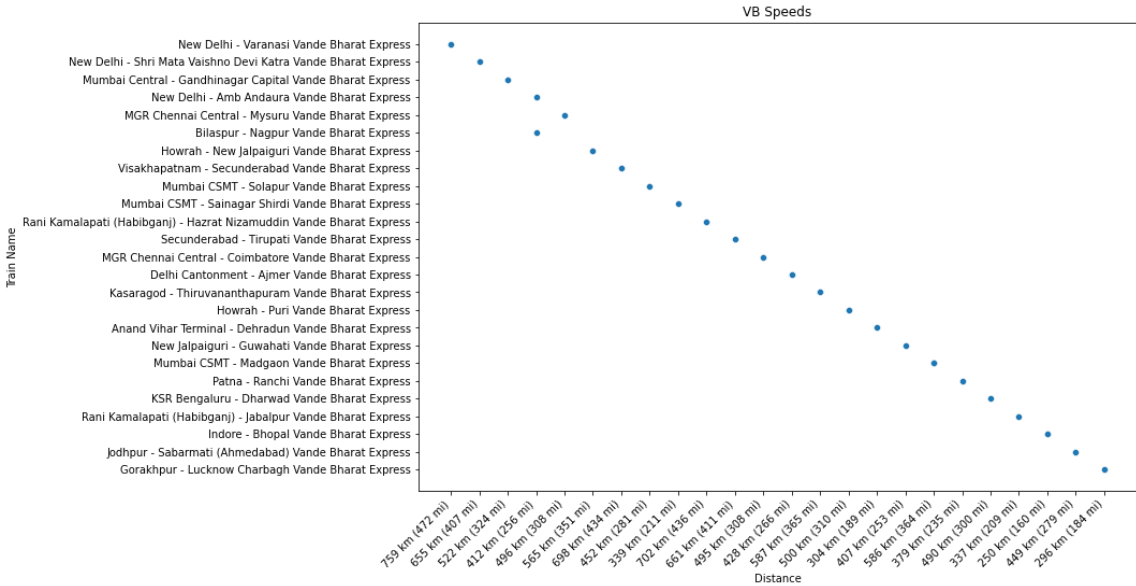
Conclusion:

New Delhi-Varanasi Vande Bharat Express is fastest among other all whose speed is around 130km/hr.

Distance travelled by Vande Bharat

In [30]:

```
plt.figure(figsize=(12,8))
sns.scatterplot(data=df,y='Train Name', x='Distance')
plt.xticks(rotation=45, ha='right')
plt.title('VB Speeds')
plt.show()
```



Conclusion:

New Delhi-Varanasi Vande Bharat Express travells highest distance which is almost 760 kms

The least distance travelled by any Vande Bharat express is around 296 kms and it is Gorakhpur-Lucknow Charbagh Vande Bharat Express

Vande Bharat Express - Inauguration Year

In [32]:

```
vb1 = df[["Train Name", "Originating City", "Terminal City", "Operator", "Speed", "Inauguration Year"]]
vb1.head()
```

Out[32]:

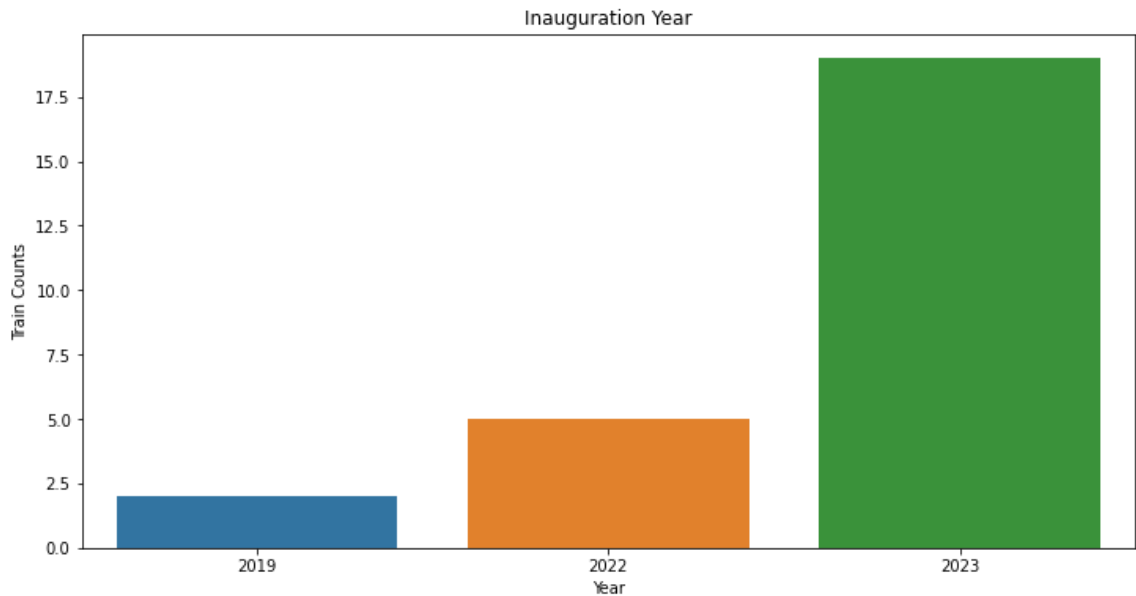
	Train Name	Originating City	Terminal City	Operator	Speed	Inauguration
0	New Delhi - Varanasi Vande Bharat Express	Delhi	Varanasi	NR	130 km/h (81 mph)	2/15/2019
1	New Delhi - Shri Mata Vaishno Devi Katra Vande...	Delhi	Katra	NR	130 km/h (81 mph)	10/3/2019
2	Mumbai Central - Gandhinagar Capital Vande Bha...	Mumbai	Gandhinagar	WR	130 km/h (81 mph)	9/30/2022
3	New Delhi - Amb Andaura Vande Bharat Express	Delhi	Andaura	NR	130 km/h (81 mph)	10/13/2022
4	MGR Chennai Central - Mysuru Vande Bharat Express	Chennai	Mysuru	SR	130 km/h (81 mph)	11/11/2022

In []:

```
df['Year'] = pd.DatetimeIndex(df['Inauguration']).year
df
```

In [31]:

```
plt.figure(figsize=(12,6))
sns.countplot(x='Year',data=df)
plt.title("Inauguration Year")
plt.xlabel("Year")
plt.ylabel("Train Counts")
plt.show()
```



Conclusion:

18 Vande Bharat Express has been inaugurated in the year 2023 (till date)
 5 Vande Bharat Express had been inaugurated in the year 2022
 2 Vande Bharat Express had been inaugurated in the year 2019

In [48]:

```
df.columns
```

Out[48]:

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

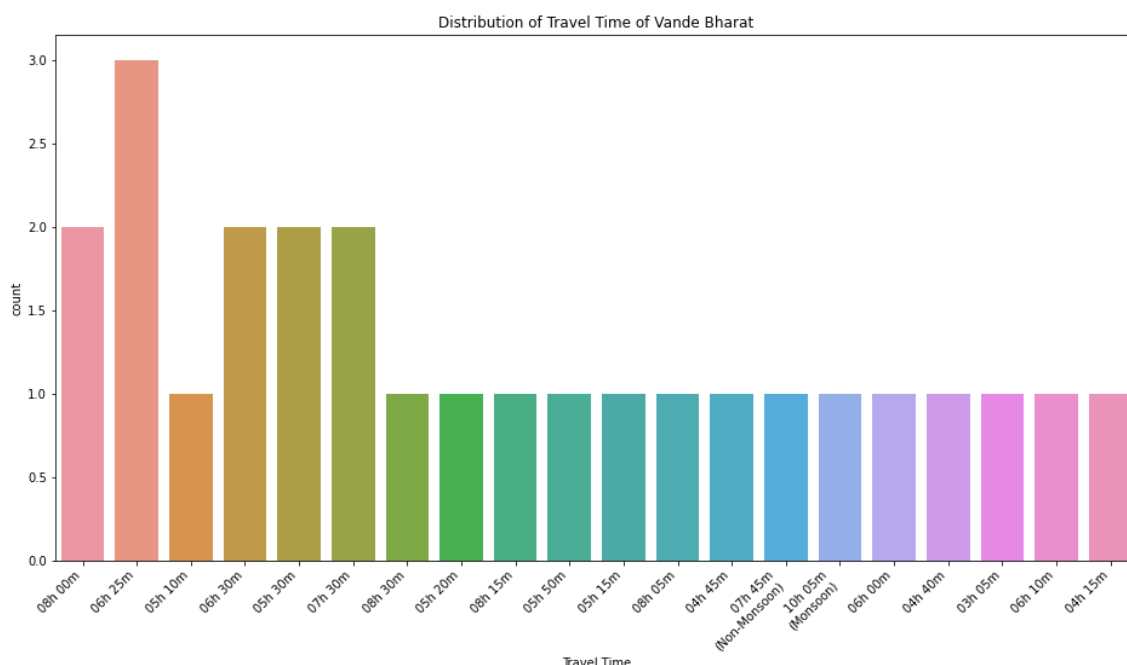
Distribution of Travel Time of Vande Bharat

In [52]:

```
plt.figure(figsize = (16,8))  
sns.countplot(data = df, x = 'Travel Time')  
plt.xlabel('Travel Time')  
plt.xticks(rotation=45, ha='right')  
plt.title('Distribution of Travel Time of Vande Bharat')
```

Out[52]:

```
Text(0.5, 1.0, 'Distribution of Travel Time of Vande Bharat')
```



Don't forget to follow me on [Medium \(https://medium.com/@borhadepiyush\)](https://medium.com/@borhadepiyush) | [GitHub \(https://github.com/PiyushBorhade\)](https://github.com/PiyushBorhade) | [Linkedin \(https://www.linkedin.com/in/piyush-borhade/\)](https://www.linkedin.com/in/piyush-borhade/) | [Kaggle \(https://www.kaggle.com/piyushborhade\)](https://www.kaggle.com/piyushborhade) | [YouTube \(https://www.youtube.com/channel/UCeFRxm0sPqPe3U6zWfN1DKg\)](https://www.youtube.com/channel/UCeFRxm0sPqPe3U6zWfN1DKg) 😊

