Exploratory Data Analysis - BC Ferries Route between Vancouver and Sunshine Coast

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt2
import statsmodels.api as sm
from statsmodels.tsa.api import VAR
```

```
task2_df = pd.read_excel('/content/drive/MyDrive/Colab_Data/bc_ferries/BC Ferries - Senior Da
```

```
total_cols_t2 = task2_df.columns
num_col_t2 = task2_df._get_numeric_data().columns
datetime_cols_t2 = task2_df.select_dtypes(include=['datetime64']).columns
cat_col_t2 = list(set(total_cols_t2)-set(num_col_t2)-set(datetime_cols_t2))
```

```
print('Total Columns', total_cols_t2)
print('Numeric Columns', num_col_t2)
print('Datetime Columns', datetime_cols_t2)
print('Categorical Columns', cat_col_t2)
```

```
Total Columns Index(['Vessel', 'Departure Terminal', 'Arrival Terminal', 'Sched Dept Ts
       'Actual Dept Ts', 'Sched Arr Ts', 'Actual Arr Ts', 'Bus', 'Semi',
       'Commercial', 'Over Height Private Vehicle',
       'Under Height Private Vehicle ', 'Foot Passengers',
       'Vehicle Passengers', 'Bus Overloads', 'Semi Overloads',
       'Commercial Overloads', 'Over Height Private Vehicle Overloads',
       'Under Height Private Vehicle  Overloads'],
      dtype='object')
Numeric Columns Index(['Semi', 'Commercial', 'Over Height Private Vehicle',
       'Under Height Private Vehicle ', 'Foot Passengers',
       'Vehicle Passengers', 'Semi Overloads', 'Commercial Overloads',
       'Over Height Private Vehicle Overloads',
       'Under Height Private Vehicle  Overloads'],
      dtype='object')
Datetime Columns Index(['Sched Dept Ts', 'Actual Dept Ts', 'Sched Arr Ts', 'Actual Arr
Categorical Columns ['Departure Terminal', 'Vessel', 'Bus', 'Arrival Terminal', 'Bus Ov
```

Features Description

```
task2_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6460 entries, 0 to 6459
Data columns (total 19 columns):
 #   Column                                    Non-Null Count  Dtype
```

```
 ---  ------                                  --------------  -----
  0   Vessel                                  6459 non-null   object
  1   Departure Terminal                      6459 non-null   object
  2   Arrival Terminal                        6459 non-null   object
  3   Sched Dept Ts                           6459 non-null   datetime64[ns]
  4   Actual Dept Ts                          6459 non-null   datetime64[ns]
  5   Sched Arr Ts                            6459 non-null   datetime64[ns]
  6   Actual Arr Ts                           6459 non-null   datetime64[ns]
  7   Bus                                     6460 non-null   object
  8   Semi                                    6459 non-null   float64
  9   Commercial                              6459 non-null   float64
  10  Over Height Private Vehicle             6459 non-null   float64
  11  Under Height Private Vehicle            6459 non-null   float64
  12  Foot Passengers                         6459 non-null   float64
  13  Vehicle Passengers                      6459 non-null   float64
  14  Bus Overloads                           6460 non-null   object
  15  Semi Overloads                          6459 non-null   float64
  16  Commercial Overloads                    6459 non-null   float64
  17  Over Height Private Vehicle Overloads   6459 non-null   float64
  18  Under Height Private Vehicle  Overloads 6459 non-null   float64
dtypes: datetime64[ns](4), float64(10), object(5)
memory usage: 959.0+ KB
```

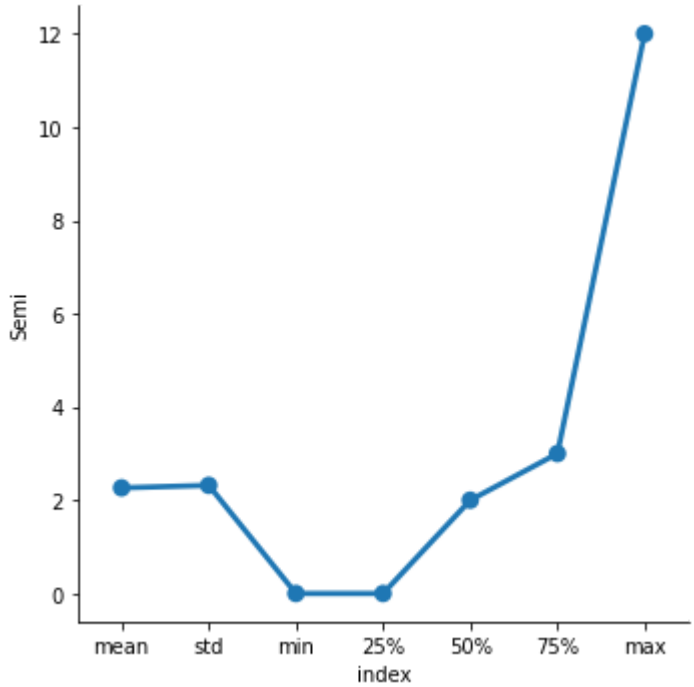
```
task2_df.describe()
```

	Semi	Commercial	Over Height Private Vehicle	Under Height Private Vehicle	Foot Passengers	Vehicle Passengers	Ove
count	6459.000000	6459.000000	6459.000000	6459.000000	6459.000000	6459.000000	6459
mean	2.264747	3.589875	10.805697	163.806626	88.444341	329.529803	0
std	2.318723	3.480703	6.395637	79.394819	64.023657	179.807107	0
min	0.000000	0.000000	0.000000	-2.000000	0.000000	-2.000000	0
25%	0.000000	1.000000	6.000000	102.000000	40.000000	188.000000	0
50%	2.000000	3.000000	10.000000	167.000000	78.000000	324.000000	0
75%	3.000000	5.000000	15.000000	230.000000	123.000000	457.000000	0
max	12.000000	20.000000	37.000000	890.000000	641.000000	1026.000000	8

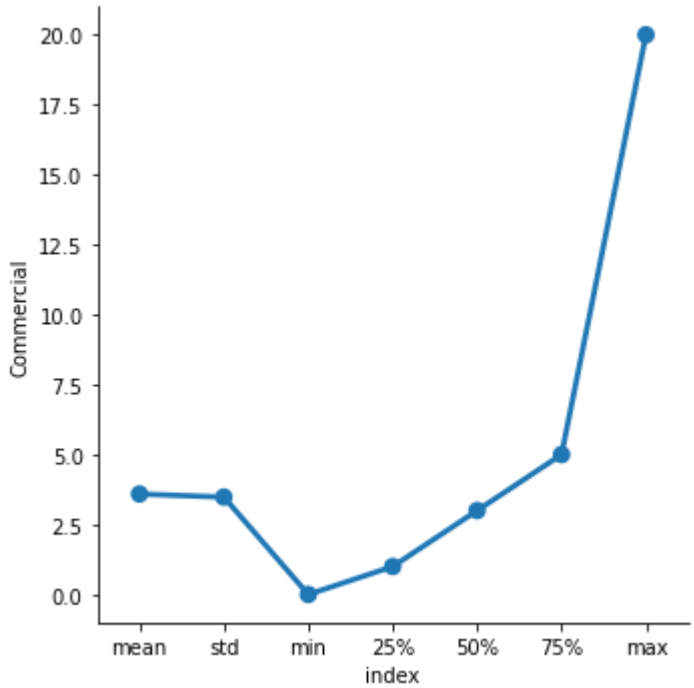
```
describe_num_df = task2_df.describe(include=['int64','float64'])
describe_num_df.reset_index(inplace=True)
describe_num_df = describe_num_df[describe_num_df['index'] != 'count']
for i in num_col_t2:
  if i in ['index']:
    continue
```

```
sns.factorplot(x='index', y=i, data=describe_num_df)
plt2.show()
```

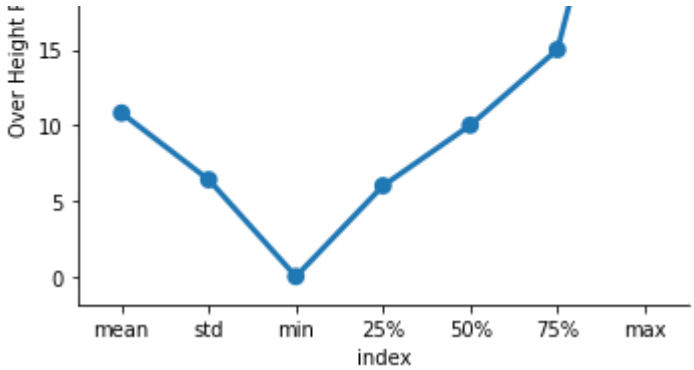
```
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:3717: UserWarning: The `f
  warnings.warn(msg)
```



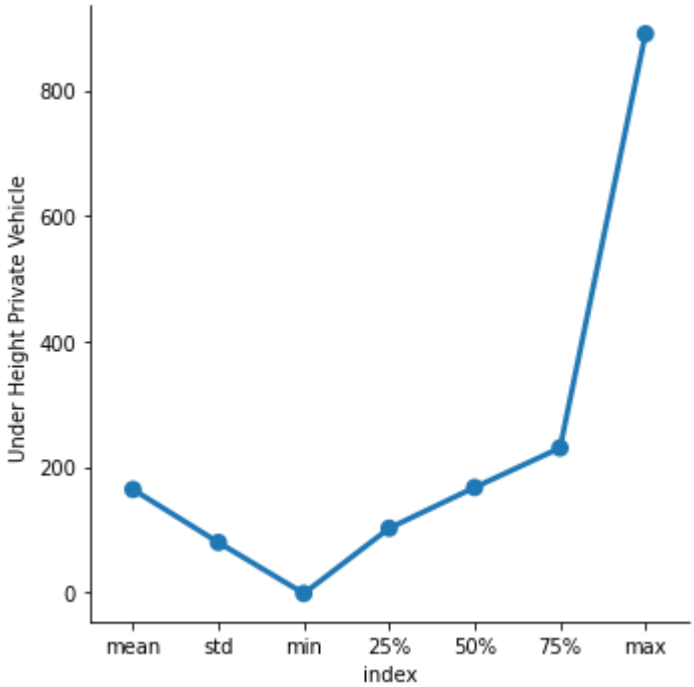

```
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:3717: UserWarning: The `f
  warnings.warn(msg)
```



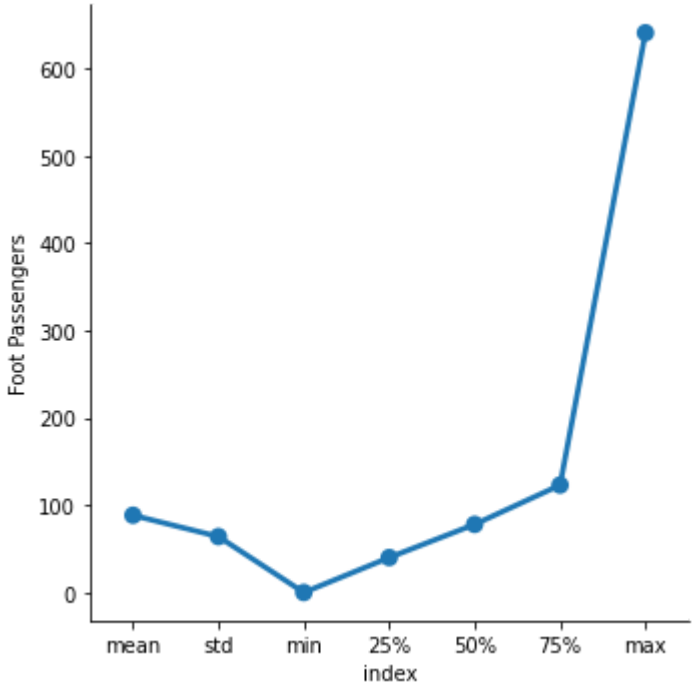

```
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:3717: UserWarning: The `f
  warnings.warn(msg)
```




```
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:3717: UserWarning: The `f
  warnings.warn(msg)
```

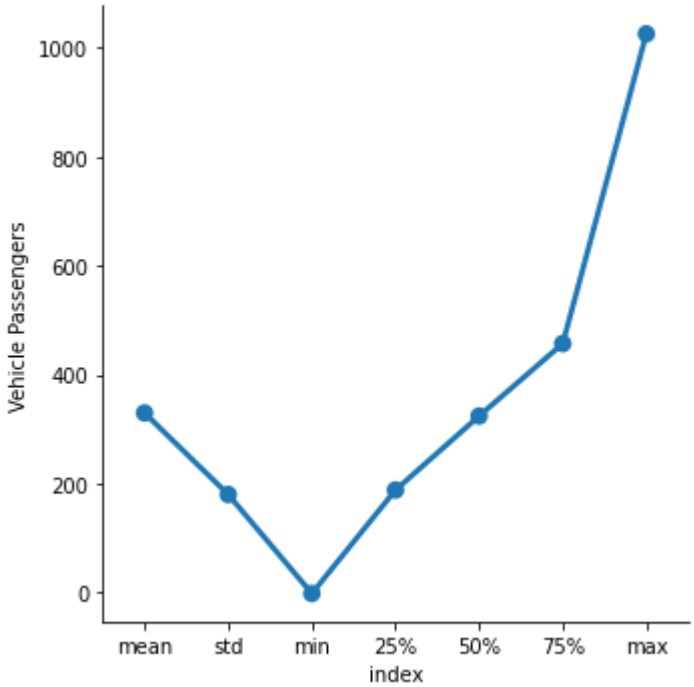



```
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:3717: UserWarning: The `f
  warnings.warn(msg)
```

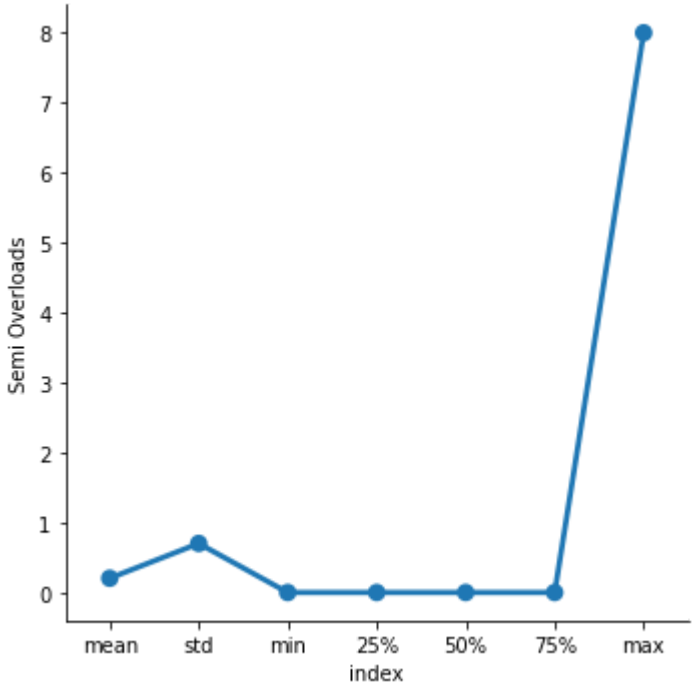



```
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:3717: UserWarning: The `f
```

```
  warnings.warn(msg)
```




```
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:3717: UserWarning: The `f
  warnings.warn(msg)
```



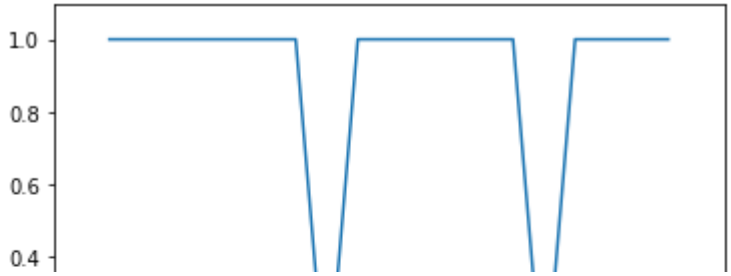

```
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:3717: UserWarning: The `f
  warnings.warn(msg)
```

Plotting Null Values

```
null_df_t2 = task2_df.apply(lambda x: sum(x.isnull())).to_frame(name='null_count')
print(null_df_t2)
```

```
                                           null_count
Vessel                                              1
Departure Terminal                                  1
Arrival Terminal                                    1
Sched Dept Ts                                       1
Actual Dept Ts                                      1
Sched Arr Ts                                        1
Actual Arr Ts                                       1
Bus                                                 0
Semi                                                1
Commercial                                          1
Over Height Private Vehicle                         1
Under Height Private Vehicle                        1
Foot Passengers                                     1
Vehicle Passengers                                  1
Bus Overloads                                       0
Semi Overloads                                      1
Commercial Overloads                                1
Over Height Private Vehicle Overloads               1
Under Height Private Vehicle  Overloads             1
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:3717: UserWarning: The `f
```

```
plt2.plot(null_df_t2.index, null_df_t2['null_count'])
plt2.xticks(null_df_t2.index, null_df_t2.index, rotation=45,
horizontalalignment='right')
plt2.xlabel('column names')
plt2.margins(0.1)
plt2.show()
```




```
task2_df[task2_df.isna().any(axis=1)]
```

	Vessel	Departure Terminal	Arrival Terminal	Sched Dept Ts	Actual Dept Ts	Sched Arr Ts	Actual Arr Ts	Bus	Semi	Commercial
6459	NaN	NaN	NaN	NaT	NaT	NaT	NaT	- 1 -	NaN	NaN

```
# Since the entire row has more than 85% NaNs, we are going to drop the row
perc = 85.0 # Here N is 85
min_count =  int(((100-perc)/100)*task2_df.shape[1] + 1)
task2_df = task2_df.dropna(axis = 0, thresh=min_count)
```

```
min_count
```

```
3
```

```
task2_df[task2_df.isna().any(axis=1)]
```

	Vessel	Departure Terminal	Arrival Terminal	Sched Dept Ts	Actual Dept Ts	Sched Arr Ts	Actual Arr Ts	Bus	Semi	Commercial	He Pri Veh

```
task2_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6459 entries, 0 to 6458
Data columns (total 19 columns):
 #   Column                                  Non-Null Count  Dtype
```

```
 ---  ------                                  --------------  -----
  0   Vessel                                  6459 non-null   object
  1   Departure Terminal                      6459 non-null   object
  2   Arrival Terminal                        6459 non-null   object
  3   Sched Dept Ts                           6459 non-null   datetime64[ns]
  4   Actual Dept Ts                          6459 non-null   datetime64[ns]
  5   Sched Arr Ts                            6459 non-null   datetime64[ns]
  6   Actual Arr Ts                           6459 non-null   datetime64[ns]
  7   Bus                                     6459 non-null   object
  8   Semi                                    6459 non-null   float64
  9   Commercial                              6459 non-null   float64
  10  Over Height Private Vehicle             6459 non-null   float64
  11  Under Height Private Vehicle            6459 non-null   float64
  12  Foot Passengers                         6459 non-null   float64
  13  Vehicle Passengers                      6459 non-null   float64
  14  Bus Overloads                           6459 non-null   object
  15  Semi Overloads                          6459 non-null   float64
  16  Commercial Overloads                    6459 non-null   float64
  17  Over Height Private Vehicle Overloads   6459 non-null   float64
  18  Under Height Private Vehicle  Overloads 6459 non-null   float64
dtypes: datetime64[ns](4), float64(10), object(5)
memory usage: 1009.2+ KB
```

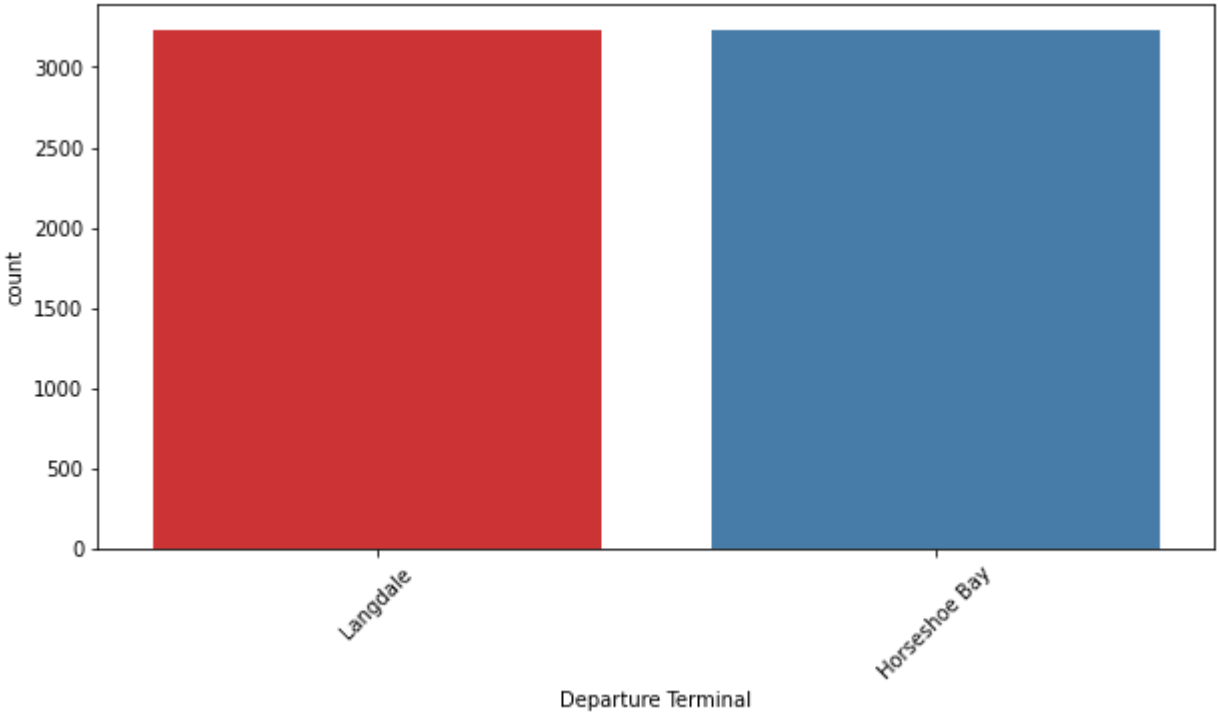
```
# Check if there's any null values left
null_df_t2 = task2_df.apply(lambda x: sum(x.isnull())).to_frame(name='null_count')
print(null_df_t2)
```

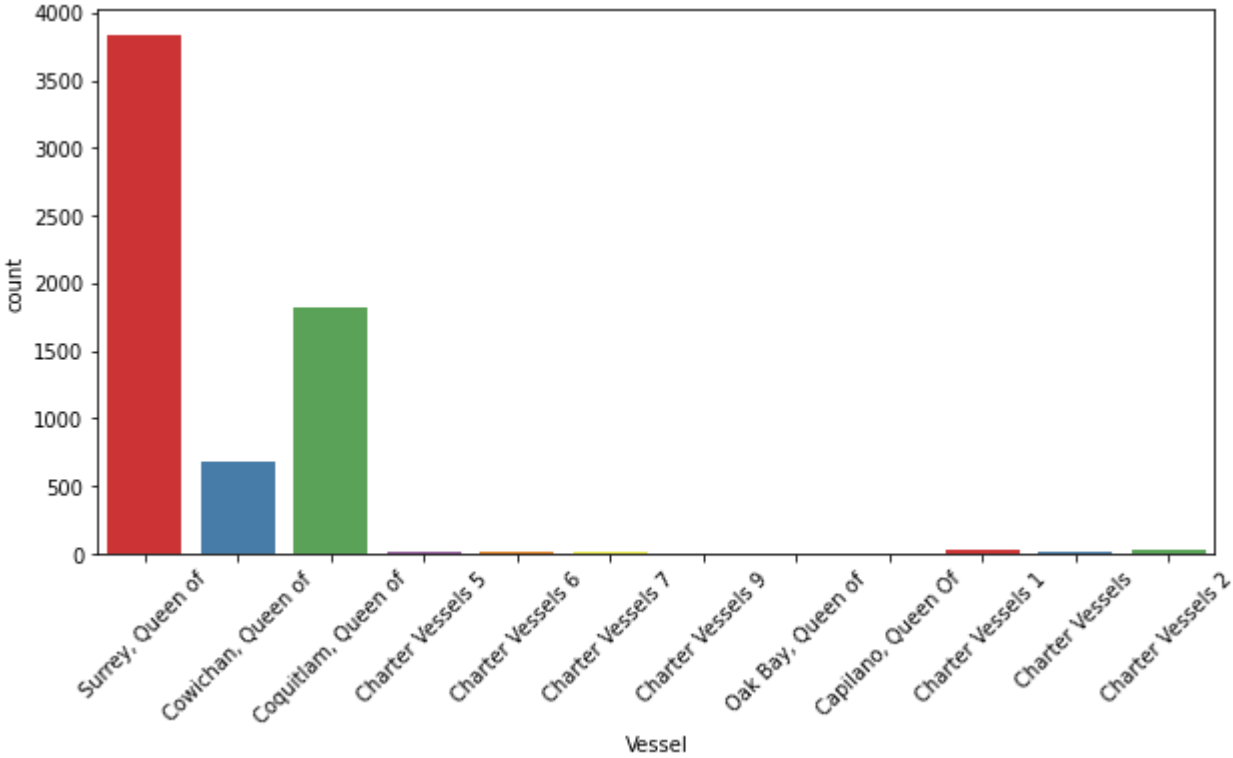
```
                                          null_count
Vessel                                             0
Departure Terminal                                 0
Arrival Terminal                                   0
Sched Dept Ts                                      0
Actual Dept Ts                                     0
Sched Arr Ts                                       0
Actual Arr Ts                                      0
Bus                                                0
Semi                                               0
Commercial                                         0
Over Height Private Vehicle                        0
Under Height Private Vehicle                       0
Foot Passengers                                    0
Vehicle Passengers                                 0
Bus Overloads                                      0
Semi Overloads                                     0
Commercial Overloads                               0
Over Height Private Vehicle Overloads              0
Under Height Private Vehicle  Overloads            0
```

Plotting value_counts() function for categorical variables.

```
for i in cat_col_t2:
   if i in ['source']:
```

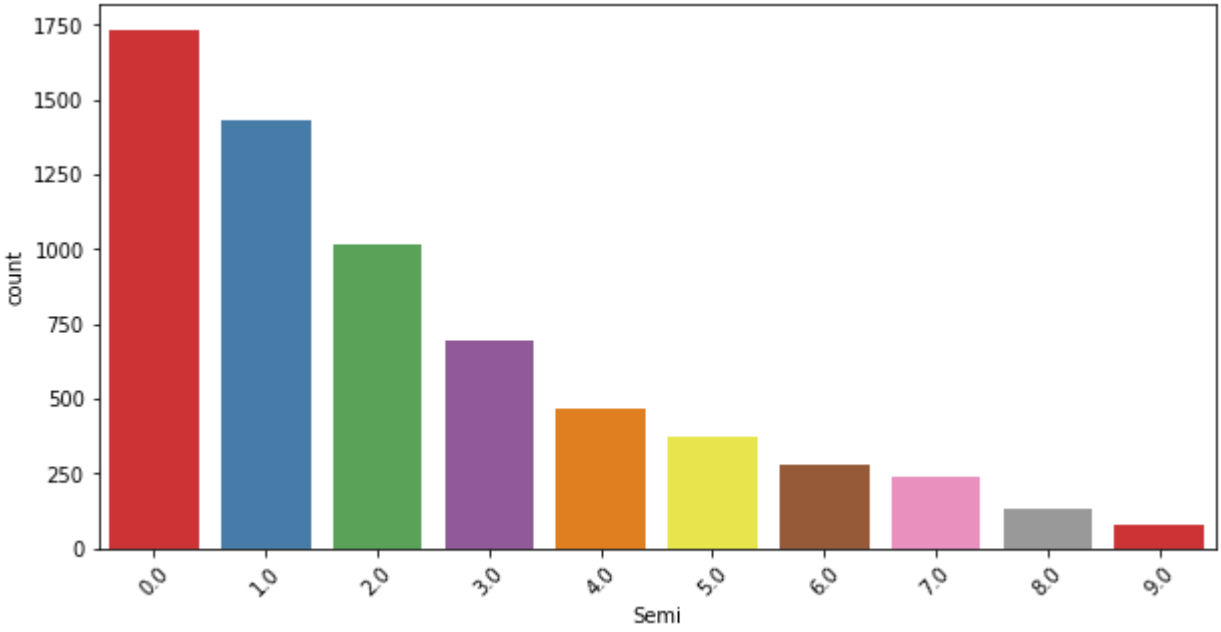
```
        continue
    plt2.figure(figsize=(10, 5))
    chart = sns.countplot(
    data=task2_df,
    x=i,
    palette='Set1'
  )
    chart.set_xticklabels(chart.get_xticklabels(), rotation=45)
    plt2.show()
```

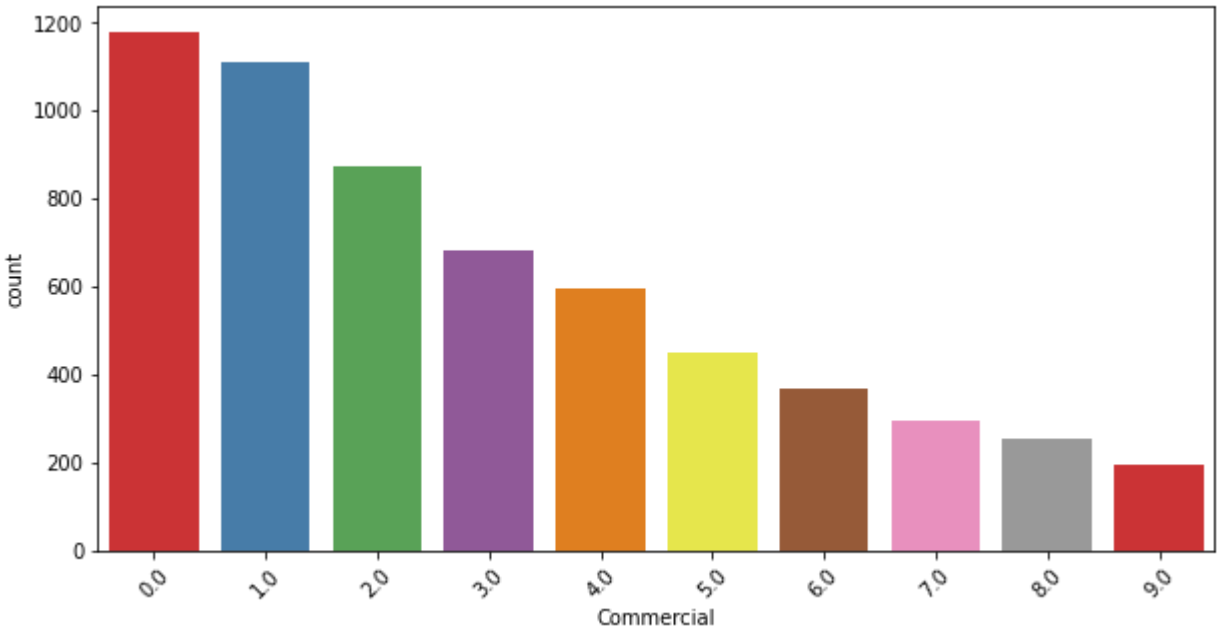

count
3000
2500
2000
1500
1000
500
0
Langdale
Horseshoe Bay
Departure Terminal

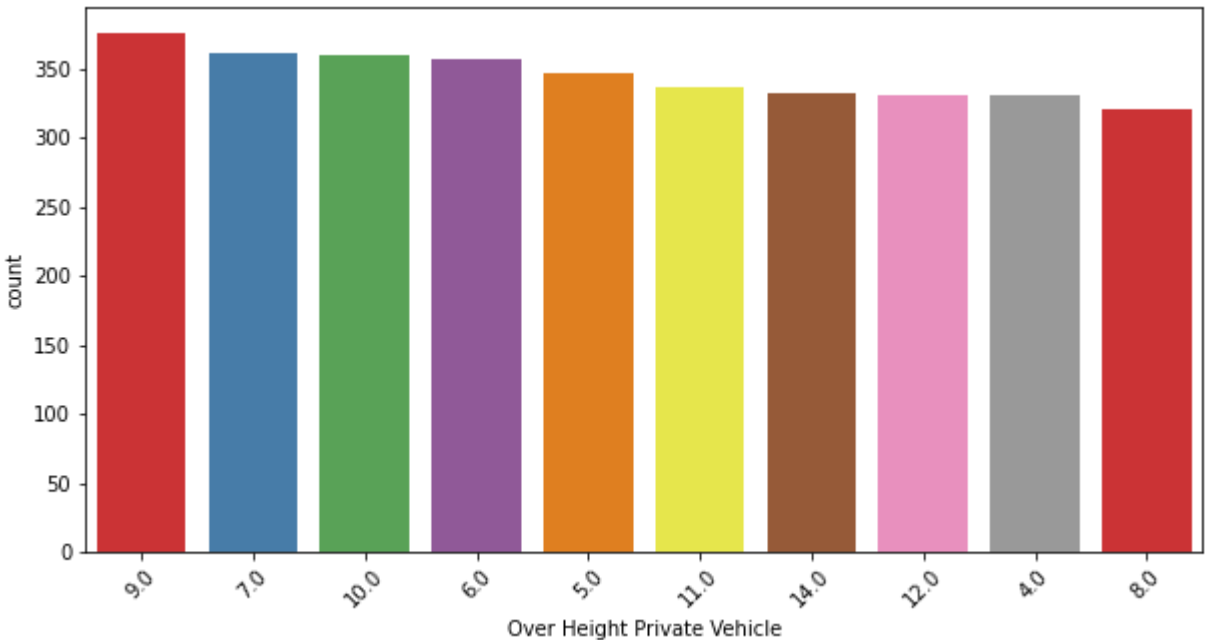

count
4000
3500
3000
2500
2000
1500
1000
500
0
Surrey, Queen of
Cowichan, Queen of
Coquitlam, Queen of
Charter Vessels 5
Charter Vessels 6
Charter Vessels 7
Charter Vessels 9
Oak Bay, Queen of
Capilano, Queen Of
Charter Vessels 1
Charter Vessels
Charter Vessels 2
Vessel

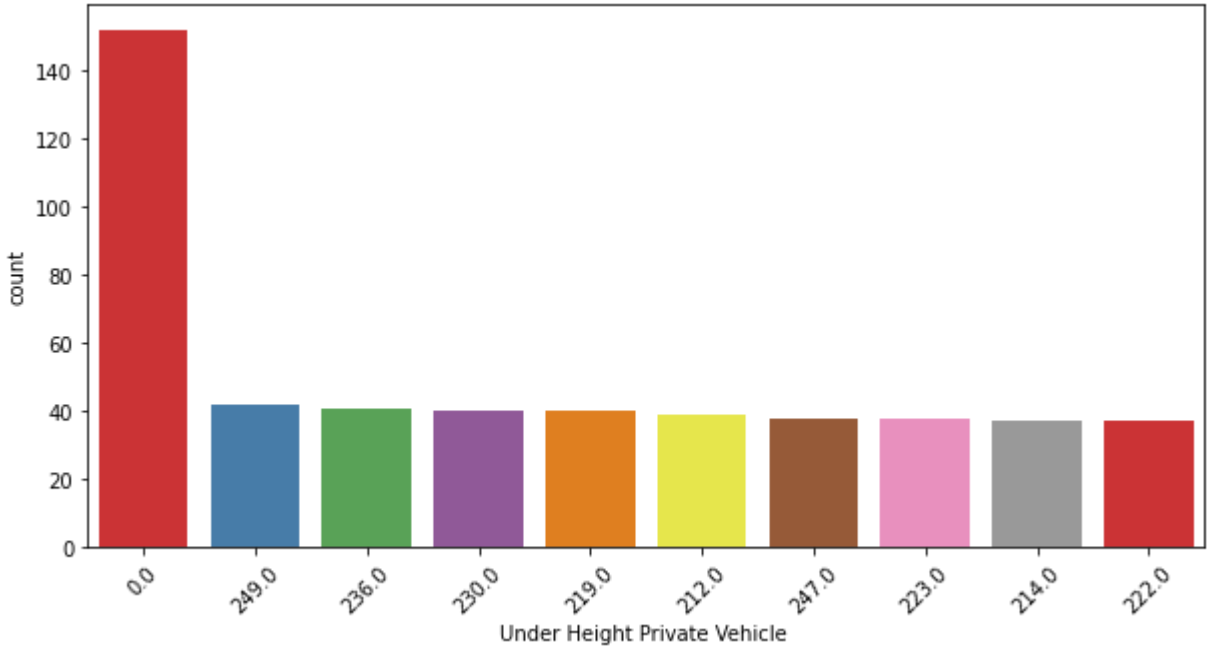

count
5000
4000
3000
2000

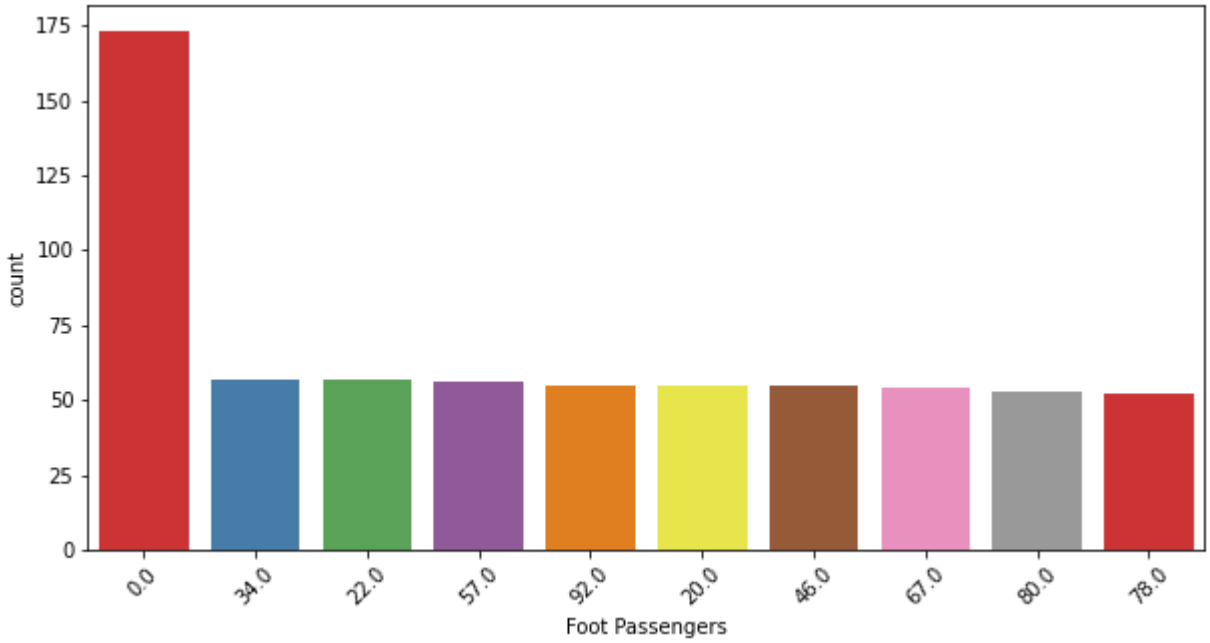
```
for i in num_col_t2:
   if i in ['source']:
     continue
   plt2.figure(figsize=(10, 5))
   chart = sns.countplot(
             data=task2_df,
             x=i,
             palette='Set1',
# This option plot top category of numerical values.
             order=pd.value_counts(task2_df[i]).iloc[:10].index
           )
   chart.set_xticklabels(chart.get_xticklabels(), rotation=45)
   plt2.show()
```

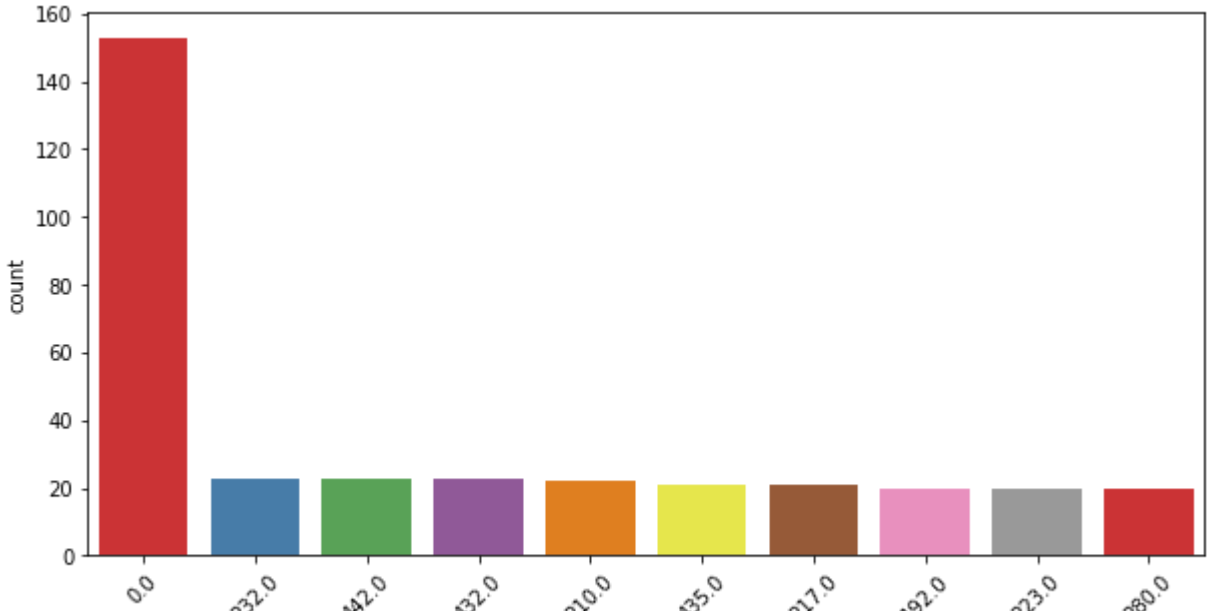

count
1750
1500
1250
1000
750
500
250
0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
Semi

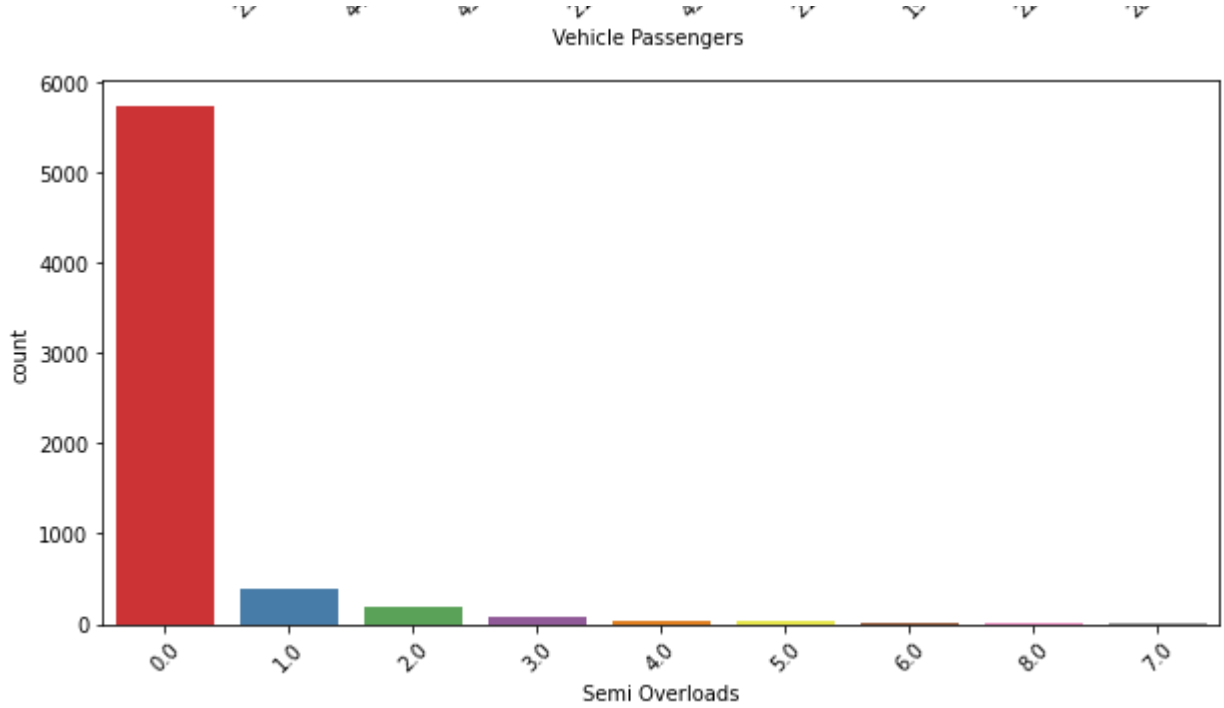

count
1200
1000
800
600
400
200
0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
Commercial

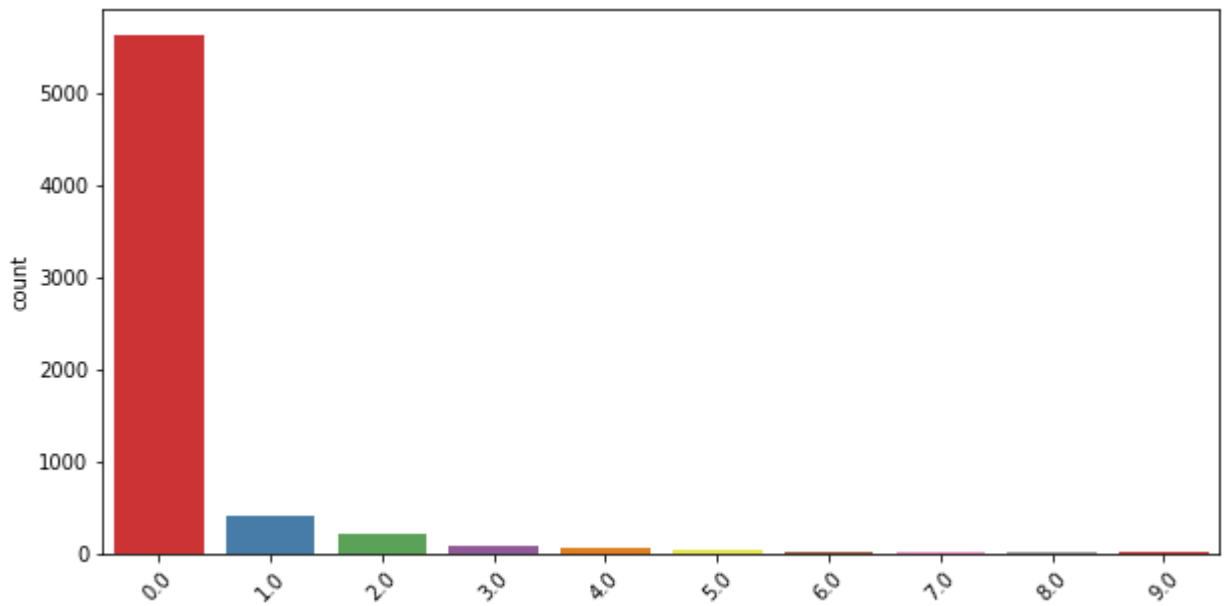

count
350
300
250
200
150
100
50
0
9.0
7.0
10.0
6.0
5.0
11.0
14.0
12.0
4.0
8.0
Over Height Private Vehicle


count
0
20
40
60
80
100
120
140
0.0
249.0
236.0
230.0
219.0
212.0
247.0
223.0
214.0
222.0
Under Height Private Vehicle


count
0
25
50
75
100
125
150
175
0.0
34.0
22.0
57.0
92.0
20.0
46.0
67.0
80.0
78.0
Foot Passengers


count
0
20
40
60
80
100
120
140
160
0.0



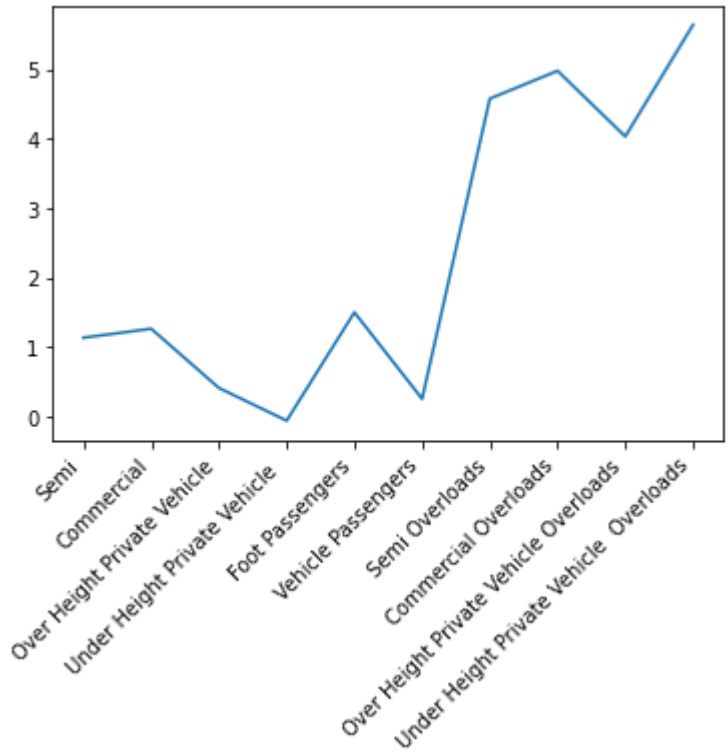


Skew and Kurtosis


```
skew = {}
kurt = {}
for i in num_col_t2:
# to skip columns for plotting
   if i in ['num_orders']:
       continue
   skew[i] = task2_df[i].skew()
   kurt[i] = task2_df[i].kurt()
print(skew)
```

```
{'Semi': 1.132460760809597, 'Commercial': 1.2621581380505422, 'Over Height Private Vehi
```

```
plt2.plot(list(skew.keys()),list(skew.values()))
plt2.xticks(rotation=45, horizontalalignment='right')
plt2.show()
```




```
print(kurt)
```

```
{'Semi': 0.6518036420087965, 'Commercial': 1.451984292684847, 'Over Height Private Vehi
```

```
corrmat = task2_df.corr()
print(corrmat)
```

```
                                          Semi  ...  Under Height Private Vehicle  O
Semi                                  1.000000  ...
Commercial                            0.530789  ...
Over Height Private Vehicle           0.360041  ...
Under Height Private Vehicle          0.140040  ...
Foot Passengers                      -0.068828  ...
Vehicle Passengers                    0.064648  ...
Semi Overloads                        0.373686  ...
Commercial Overloads                  0.317494  ...
Over Height Private Vehicle Overloads 0.181254  ...
Under Height Private Vehicle  Overloads 0.000741  ...

[10 rows x 10 columns]
```

```
plt2.figure(figsize=(18, 10))
```

```
sns.heatmap(corrmat, vmax=1, annot=True, linewidths=.5)
plt2.xticks(rotation=30, horizontalalignment='right')
plt2.show()
```


```
datetime_cols_t2
```

```
Index(['Sched Dept Ts', 'Actual Dept Ts', 'Sched Arr Ts', 'Actual Arr Ts'], dtype='obje
```

```
task2_df.dtypes
```

```
Vessel                                          object
Departure Terminal                              object
Arrival Terminal                                object
Sched Dept Ts                             datetime64[ns]
```

```
Actual Dept Ts                            datetime64[ns]
Sched Arr Ts                              datetime64[ns]
Actual Arr Ts                             datetime64[ns]
Bus                                                object
Semi                                              float64
Commercial                                        float64
Over Height Private Vehicle                       float64
Under Height Private Vehicle                      float64
Foot Passengers                                   float64
Vehicle Passengers                                float64
Bus Overloads                                      object
Semi Overloads                                    float64
Commercial Overloads                              float64
Over Height Private Vehicle Overloads             float64
Under Height Private Vehicle  Overloads           float64
dtype: object
```

```
columns_to_be_converted = ['Semi', 'Commercial', 'Over Height Private Vehicle', 'Under Height
task2_df[columns_to_be_converted] = task2_df[columns_to_be_converted].astype(int)
```

```
task2_df.head(2)
```

	Vessel	Departure Terminal	Arrival Terminal	Sched Dept Ts	Actual Dept Ts	Sched Arr Ts	Actual Arr Ts	Bus	Semi
0	Surrey, Queen of	Langdale	Horseshoe Bay	2019-01-01 08:40:00	2019-01-01 08:40:00	2019-01-01 09:19:59.999	2019-01-01 09:19:59.999	0	0
1	Surrey, Queen of	Horseshoe Bay	Langdale	2019-01-01 09:45:00	2019-01-01 09:46:00	2019-01-01 10:24:59.999	2019-01-01 10:26:59.999	0	1

```
task2_df['departure_delay'] = task2_df['Actual Dept Ts'] - task2_df['Sched Dept Ts']
```

```
task2_df['arrival_delay'] = task2_df['Actual Arr Ts'] - task2_df['Sched Arr Ts']
```

```
task2_df.head()
```

	Vessel	Departure Terminal	Arrival Terminal	Sched Dept Ts	Actual Dept Ts	Sched Arr Ts	Actual Arr Ts	Bus
0	Surrey, Queen of	Langdale	Horseshoe Bay	2019-01-01 08:40:00.000	2019-01-01 08:40:00.000	2019-01-01 09:19:59.999	2019-01-01 09:19:59.999	0
1	Surrey, Queen of	Horseshoe Bay	Langdale	2019-01-01 09:45:00.000	2019-01-01 09:46:00.000	2019-01-01 10:24:59.999	2019-01-01 10:26:59.999	0
2	Surrey, Queen of	Langdale	Horseshoe Bay	2019-01-01 10:49:59.999	2019-01-01 10:49:59.999	2019-01-01 11:29:59.999	2019-01-01 11:34:59.999	0
3	Surrey, Queen of	Horseshoe Bay	Langdale	2019-01-01 11:54:59.999	2019-01-01 11:58:00.000	2019-01-01 12:35:00.000	2019-01-01 12:37:59.999	0
4	Surrey, Queen	Langdale	Horseshoe	2019-01-01	2019-01-01	2019-01-01	2019-01-01	0

```
task2_df['total_delay'] = task2_df['departure_delay'] + task2_df['arrival_delay']
```

```
task2_df.head(2)
```

	Vessel	Departure Terminal	Arrival Terminal	Sched Dept Ts	Actual Dept Ts	Sched Arr Ts	Actual Arr Ts	Bus	Semi
0	Surrey, Queen of	Langdale	Horseshoe Bay	2019-01-01 08:40:00	2019-01-01 08:40:00	2019-01-01 09:19:59.999	2019-01-01 09:19:59.999	0	0
1	Surrey, Queen of	Horseshoe Bay	Langdale	2019-01-01 09:45:00	2019-01-01 09:46:00	2019-01-01 10:24:59.999	2019-01-01 10:26:59.999	0	1

```
task2_df['total_overload'] = task2_df['Bus Overloads'] + task2_df['Semi Overloads'] + task2_c
```

Exploring time series analysis for percentage overload Vs Month of the Year

```
y_overload = task2_df
```

```
y_delay = task2_df[['total_delay', 'Bus Overloads', 'Semi Overloads', 'Commercial Overloads',
```

```
y_delay.head()
```

	total_delay	Bus Overloads	Semi Overloads	Commercial Overloads	Over Height Private Vehi Overlo
0	0 days 00:00:00	0	0	0	
1	0 days 00:03:00	0	0	0	
2	0 days 00:05:00	0	0	0	
3	0 days 00:06:00	0	0	0	
4	0 days 00:03:00.002000	0	0	0	

```
y_delay['total_overload'] = y_delay['Bus Overloads'] + y_delay['Semi Overloads'] + y_delay['C
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user
  """Entry point for launching an IPython kernel.
```

```
y_delay.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6459 entries, 0 to 6458
Data columns (total 7 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   total_delay                             6459 non-null   timedelta64[ns]
 1   Bus Overloads                           6459 non-null   int64
 2   Semi Overloads                          6459 non-null   int64
 3   Commercial Overloads                    6459 non-null   int64
 4   Over Height Private Vehicle Overloads   6459 non-null   int64
 5   Under Height Private Vehicle  Overloads 6459 non-null   int64
 6   total_overload                          6459 non-null   int64
dtypes: int64(6), timedelta64[ns](1)
memory usage: 723.7 KB
```

```
data = y_delay.iloc[:, 0:6200]
model = VAR(data)
model_fit = model.fit()
```