Road Accident Casualities Analysis

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

In [2]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [3]:
```
df = pd.read_csv("accident_data.csv")
```

In [4]:
```
df.head()
```

Out[4]:

	Index	Accident_Severity	Accident Date	Latitude	Light_Conditions	District Area	Longitude	Number_of_Casualties	Number_of_Vehicle
0	200701BS64157	Serious	05-06-2019	51.506187	Darkness - lights lit	Kensington and Chelsea	-0.209082	1	
1	200701BS65737	Serious	02-07-2019	51.495029	Daylight	Kensington and Chelsea	-0.173647	1	
2	200701BS66127	Serious	26-08-2019	51.517715	Darkness - lighting unknown	Kensington and Chelsea	-0.210215	1	
3	200701BS66128	Serious	16-08-2019	51.495478	Daylight	Kensington and Chelsea	-0.202731	1	
4	200701BS66837	Slight	03-09-2019	51.488576	Darkness - lights lit	Kensington and Chelsea	-0.192487	1	

In [5]:
```
df.tail()
```

Out[5]:

	Index	Accident_Severity	Accident Date	Latitude	Light_Conditions	District Area	Longitude	Number_of_Casualties	Number_of_Ve
660674	201091NM01760	Slight	18-02-2022	57.374005	Daylight	Highland	-3.467828	2	
660675	201091NM01881	Slight	21-02-2022	57.232273	Darkness - no lighting	Highland	-3.809281	1	
660676	201091NM01935	Slight	23-02-2022	57.585044	Daylight	Highland	-3.862727	1	
660677	201091NM01964	Serious	23-02-2022	57.214898	Darkness - no lighting	Highland	-3.823997	1	
660678	201091NM02142	Serious	28-02-2022	57.575210	Daylight	Highland	-3.895673	1	

In [6]:
```
df.shape
```

Out[6]:
```
(660679, 14)
```

In [7]:
```
df.columns
```

Out[7]:
```
Index(['Index', 'Accident_Severity', 'Accident Date', 'Latitude',
       'Light_Conditions', 'District Area', 'Longitude',
       'Number_of_Casualties', 'Number_of_Vehicles', 'Road_Surface_Conditions',
       'Road_Type', 'Urban_or_Rural_Area', 'Weather_Conditions',
       'Vehicle_Type'],
      dtype='object')
```

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

Out[8]:
```
19
```

In [9]:
```
df = df.drop_duplicates()
```

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
Index                        0
Accident_Severity            0
Accident Date                0
Latitude                    25
Light_Conditions             0
District Area                0
Longitude                   26
Number_of_Casualties         0
Number_of_Vehicles           0
Road_Surface_Conditions    726
Road_Type                 4520
Urban_or_Rural_Area         15
Weather_Conditions       14127
Vehicle_Type                 0
dtype: int64
```

In [11]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 660660 entries, 0 to 660678
Data columns (total 14 columns):
 #   Column                   Non-Null Count   Dtype  
---  ------                   --------------   -----  
 0   Index                    660660 non-null  object 
 1   Accident_Severity        660660 non-null  object 
 2   Accident Date            660660 non-null  object 
 3   Latitude                 660635 non-null  float64
 4   Light_Conditions         660660 non-null  object 
 5   District Area            660660 non-null  object 
 6   Longitude                660634 non-null  float64
 7   Number_of_Casualties     660660 non-null  int64  
 8   Number_of_Vehicles       660660 non-null  int64  
 9   Road_Surface_Conditions  659934 non-null  object 
 10  Road_Type                656140 non-null  object 
 11  Urban_or_Rural_Area      660645 non-null  object 
 12  Weather_Conditions       646533 non-null  object 
 13  Vehicle_Type             660660 non-null  object 
dtypes: float64(2), int64(2), object(10)
memory usage: 75.6+ MB
```

In [12]:
```
df.describe()
```

Out[12]:

	Latitude	Longitude	Number_of_Casualties	Number_of_Vehicles
count	660635.000000	660634.000000	660660.000000	660660.000000
mean	52.553857	-1.431216	1.357047	1.831255
std	1.406909	1.383322	0.824856	0.715272
min	49.914430	-7.516225	1.000000	1.000000
25%	51.490692	-2.332278	1.000000	1.000000
50%	52.315628	-1.411666	1.000000	2.000000
75%	53.453451	-0.232870	1.000000	2.000000
max	60.757544	1.762010	68.000000	32.000000

In [13]:
```
df.nunique()
```

Out[13]:
```
Index                      421020
Accident_Severity               3
Accident Date                1461
Latitude                   511618
Light_Conditions                5
District Area                 422
Longitude                  529766
Number_of_Casualties           36
Number_of_Vehicles             19
Road_Surface_Conditions         5
Road_Type                       5
Urban_or_Rural_Area             3
Weather_Conditions              8
Vehicle_Type                   16
dtype: int64
```

In [14]:
```
object_columns = df.select_dtypes(include=['object', 'bool']).columns
print("Object type columns:")
print(object_columns)

numerical_columns = df.select_dtypes(include=['int64', 'float64']).columns
print("\nNumerical type columns:")
print(numerical_columns)
```

```
Object type columns:
Index(['Index', 'Accident_Severity', 'Accident Date', 'Light_Conditions',
       'District Area', 'Road_Surface_Conditions', 'Road_Type',
       'Urban_or_Rural_Area', 'Weather_Conditions', 'Vehicle_Type'],
      dtype='object')

Numerical type columns:
Index(['Latitude', 'Longitude', 'Number_of_Casualties', 'Number_of_Vehicles'], dtype='object')
```

In [15]:
```
def classify_features(df):
    categorical_features = []
    non_categorical_features = []
    discrete_features = []
    continuous_features = []

    for column in df.columns:
        if df[column].dtype in ['object', 'bool']:
            if df[column].nunique() < 15:
                categorical_features.append(column)
            else:
                non_categorical_features.append(column)
        elif df[column].dtype in ['int64', 'float64']:
            if df[column].nunique() < 10:
                discrete_features.append(column)
            else:
                continuous_features.append(column)

    return categorical_features, non_categorical_features, discrete_features, continuous_features
```

In [16]:
```
categorical, non_categorical, discrete, continuous = classify_features(df)
```

In [17]:
```
print("Categorical Features:", categorical)
print("Non-Categorical Features:", non_categorical)
print("Discrete Features:", discrete)
print("Continuous Features:", continuous)
```

```
Categorical Features: ['Accident_Severity', 'Light_Conditions', 'Road_Surface_Conditions', 'Road_Type', 'Urban_
or_Rural_Area', 'Weather_Conditions']
Non-Categorical Features: ['Index', 'Accident Date', 'District Area', 'Vehicle_Type']
Discrete Features: []
Continuous Features: ['Latitude', 'Longitude', 'Number_of_Casualties', 'Number_of_Vehicles']
```

In [18]:
```
null_counts = df.isnull().sum()
null_columns = null_counts[null_counts > 0].index.tolist()
```

In [19]:
```
total_rows = len(df)
null_percentage = (null_counts / total_rows) * 100
```

In [20]:
```
null_df = pd.DataFrame({
    'Column': null_counts.index,
    'Null Count': null_counts.values,
    'Null Percentage': null_percentage.values
})
```

In [21]:
```
null_df = null_df.sort_values(by='Null Count', ascending=False)
```

In [22]:
```
null_df
```

Out[22]:

	Column	Null Count	Null Percentage
12	Weather_Conditions	14127	2.138316
10	Road_Type	4520	0.684164
9	Road_Surface_Conditions	726	0.109890
6	Longitude	26	0.003935
3	Latitude	25	0.003784
11	Urban_or_Rural_Area	15	0.002270
0	Index	0	0.000000
1	Accident_Severity	0	0.000000
2	Accident Date	0	0.000000
4	Light_Conditions	0	0.000000
5	District Area	0	0.000000
7	Number_of_Casualties	0	0.000000
8	Number_of_Vehicles	0	0.000000
13	Vehicle_Type	0	0.000000

In [23]:
```
mode_value = df['Weather_Conditions'].mode()[0]
df['Weather_Conditions'].fillna(mode_value, inplace=True)
```

In [24]:
```
mode_value = df['Road_Type'].mode()[0]
df['Road_Type'].fillna(mode_value, inplace=True)
```

In [25]:
```
mode_value = df['Road_Surface_Conditions'].mode()[0]
df['Road_Surface_Conditions'].fillna(mode_value, inplace=True)
```

In [26]:
```
mode_value = df['Urban_or_Rural_Area'].mode()[0]
df['Urban_or_Rural_Area'].fillna(mode_value, inplace=True)
```

In [27]:
```
mean_value = df['Latitude'].mean()
df['Latitude'].fillna(mean_value, inplace=True)
```

In [28]:
```
mean_value = df['Longitude'].mean()
df['Longitude'].fillna(mean_value, inplace=True)
```

In [29]:
```
df.isnull().sum()
```

Out[29]:
```
Index                       0
Accident_Severity           0
Accident Date               0
Latitude                    0
Light_Conditions            0
District Area               0
Longitude                   0
Number_of_Casualties        0
Number_of_Vehicles          0
Road_Surface_Conditions     0
Road_Type                   0
Urban_or_Rural_Area         0
Weather_Conditions          0
Vehicle_Type                0
dtype: int64
```

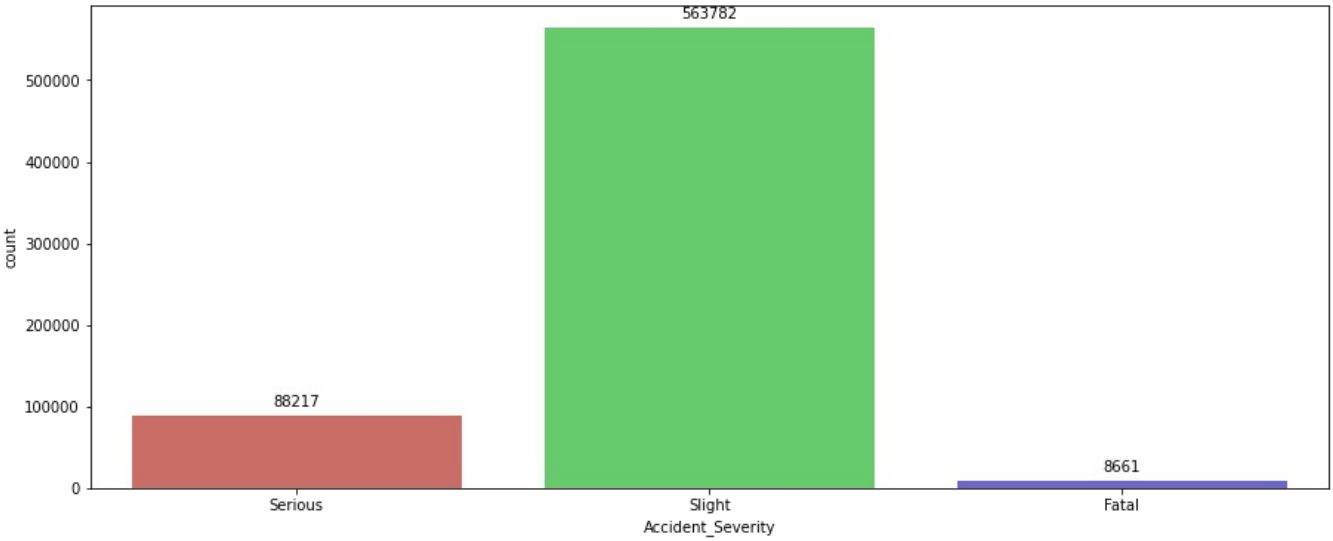
In [30]:
```
for i in categorical:
    print(i, ':')
    print(df[i].unique())
    print()
```

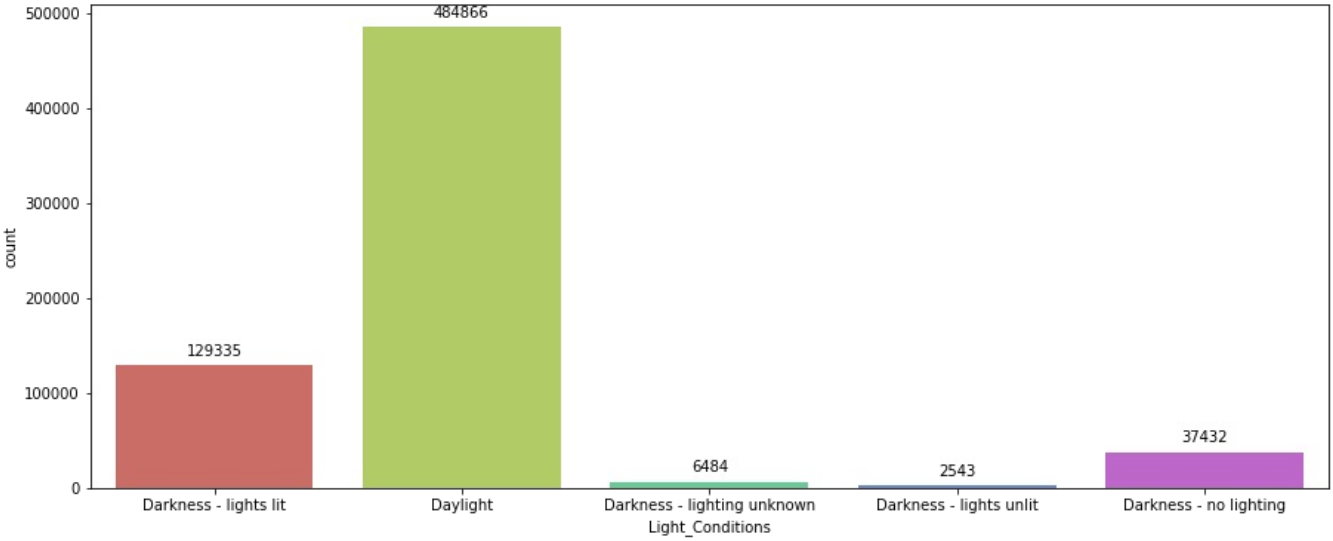
```
Accident_Severity :
['Serious' 'Slight' 'Fatal']

Light_Conditions :
['Darkness - lights lit' 'Daylight' 'Darkness - lighting unknown'
 'Darkness - lights unlit' 'Darkness - no lighting']

Road_Surface_Conditions :
['Dry' 'Wet or damp' 'Snow' 'Frost or ice' 'Flood over 3cm. deep']

Road_Type :
['Single carriageway' 'Dual carriageway' 'One way street' 'Roundabout'
 'Slip road']

Urban_or_Rural_Area :
['Urban' 'Rural' 'Unallocated']

Weather_Conditions :
['Fine no high winds' 'Raining no high winds' 'Other' 'Fine + high winds'
 'Raining + high winds' 'Snowing no high winds' 'Fog or mist'
 'Snowing + high winds']
```

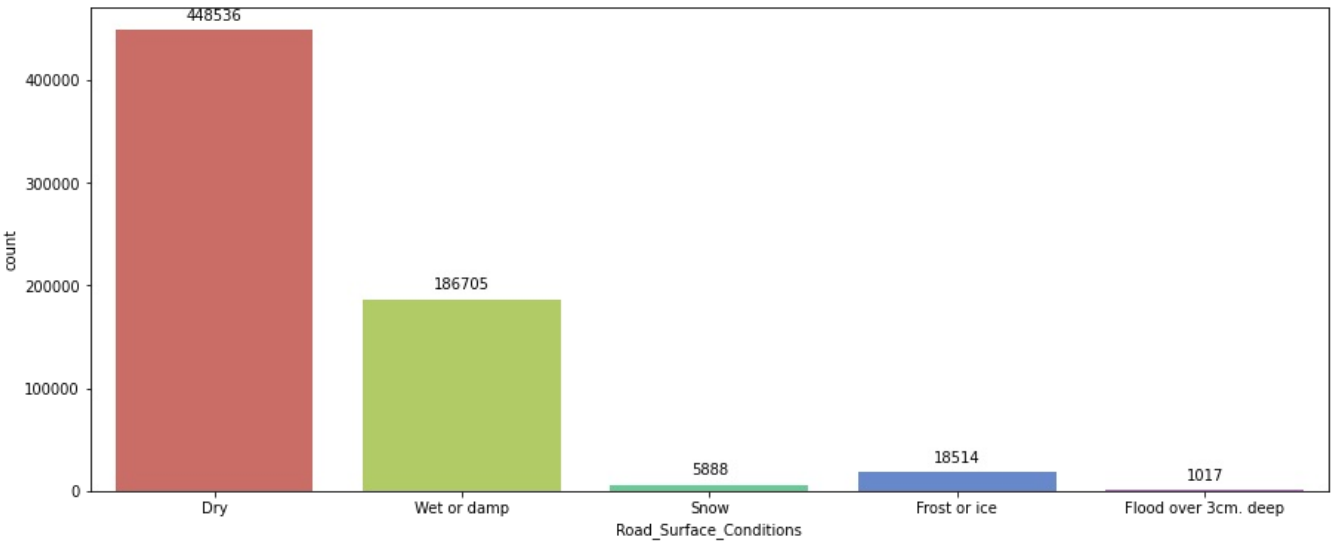
In [31]:
```
for i in categorical:
    print(i, ':')
    print(df[i].value_counts())
    print()
```

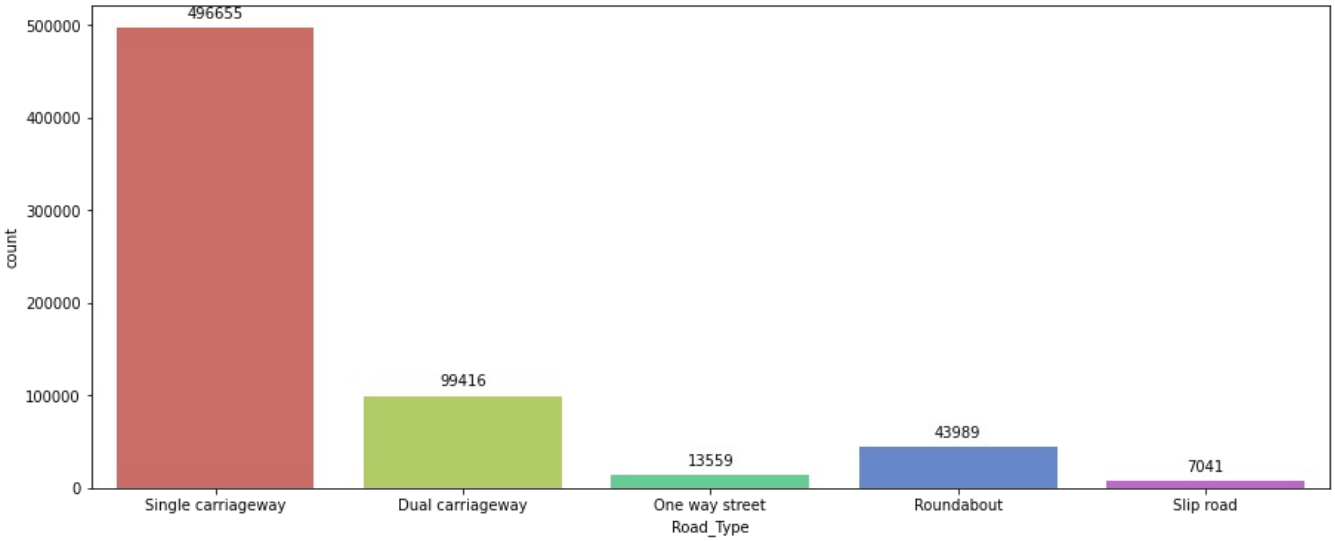
```
Accident_Severity :
Slight     563782
Serious     88217
Fatal        8661
Name: Accident_Severity, dtype: int64

Light_Conditions :
Daylight                      484866
Darkness - lights lit         129335
Darkness - no lighting         37432
Darkness - lighting unknown     6484
Darkness - lights unlit         2543
Name: Light_Conditions, dtype: int64

Road_Surface_Conditions :
Dry                     448536
Wet or damp             186705
Frost or ice             18514
Snow                      5888
Flood over 3cm. deep      1017
Name: Road_Surface_Conditions, dtype: int64

Road_Type :
Single carriageway    496655
Dual carriageway       99416
Roundabout             43989
One way street         13559
Slip road               7041
Name: Road_Type, dtype: int64

Urban_or_Rural_Area :
Urban          421675
Rural          238974
Unallocated        11
Name: Urban_or_Rural_Area, dtype: int64

Weather_Conditions :
Fine no high winds       534999
Raining no high winds     79696
Other                     17148
Raining + high winds       9615
Fine + high winds          8553
Snowing no high winds      6237
Fog or mist                3528
Snowing + high winds        884
Name: Weather_Conditions, dtype: int64
```

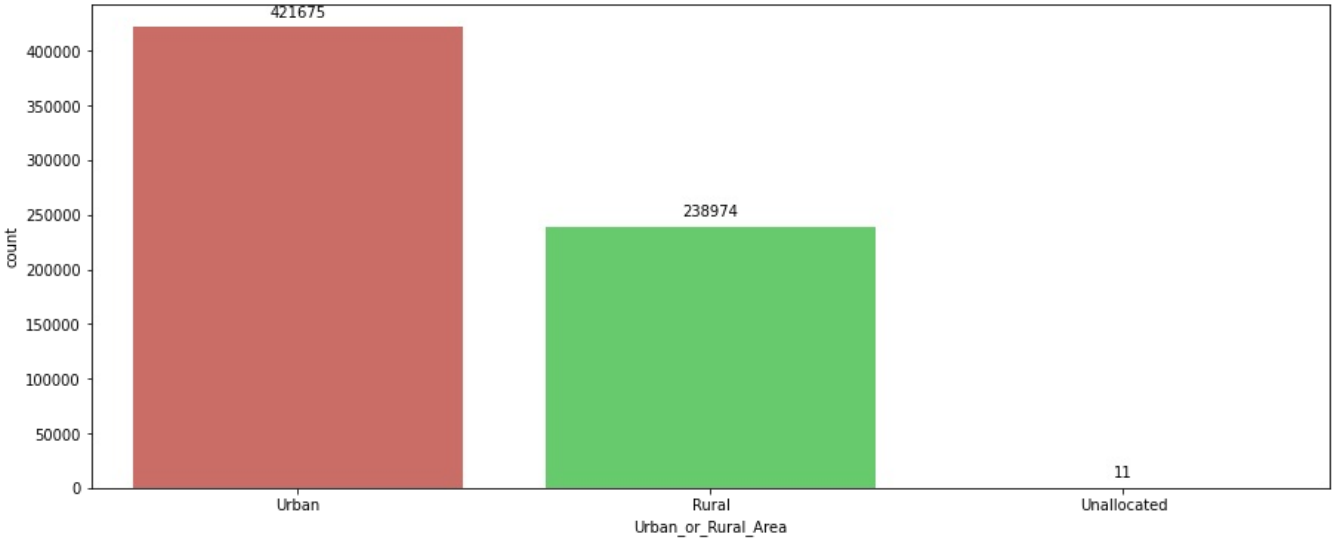
In [32]:
```
for i in categorical:
    plt.figure(figsize=(15, 6))
    ax = sns.countplot(x=i, data=df, palette='hls')
    for p in ax.patches:
        ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2., p.get_height()),
                    ha='center', va='center', xytext=(0, 10), textcoords='offset points')
    plt.show()
```

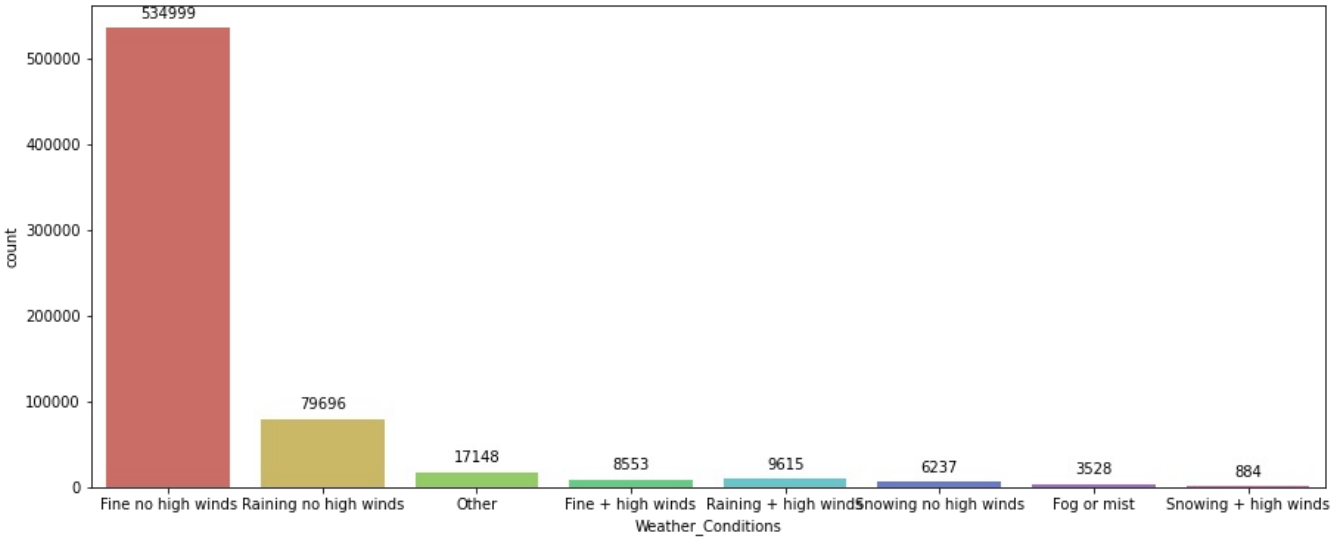

563782
88217
8661
count
Serious
Slight
Fatal
Accident_Severity


484866
129335
6484
2543
37432
count
Darkness - lights lit
Daylight
Darkness - lighting unknown
Darkness - lights unlit
Darkness - no lighting
Light_Conditions


448536
186705
5888
18514
1017
count
Dry
Wet or damp
Snow
Frost or ice
Flood over 3cm. deep
Road_Surface_Conditions

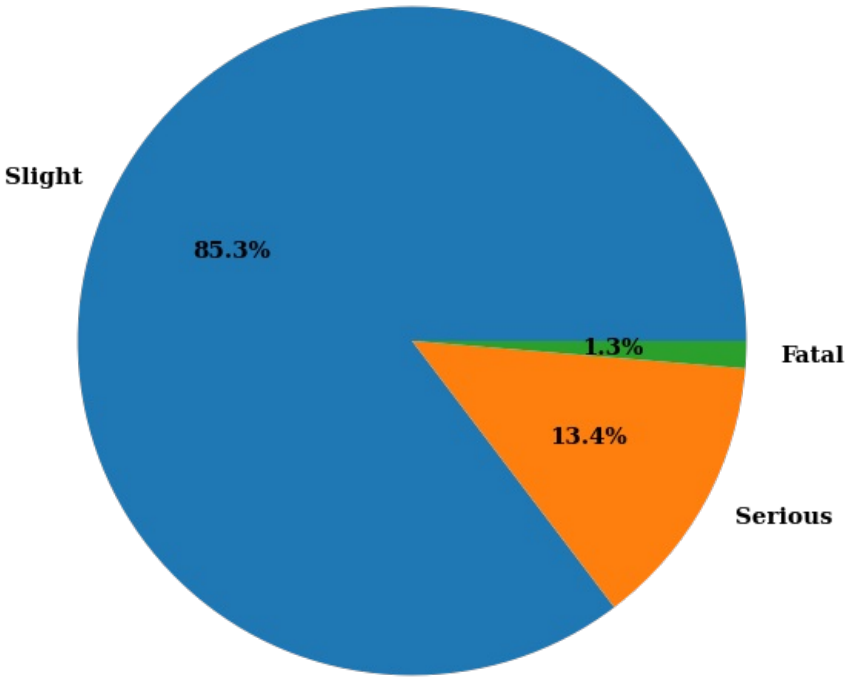




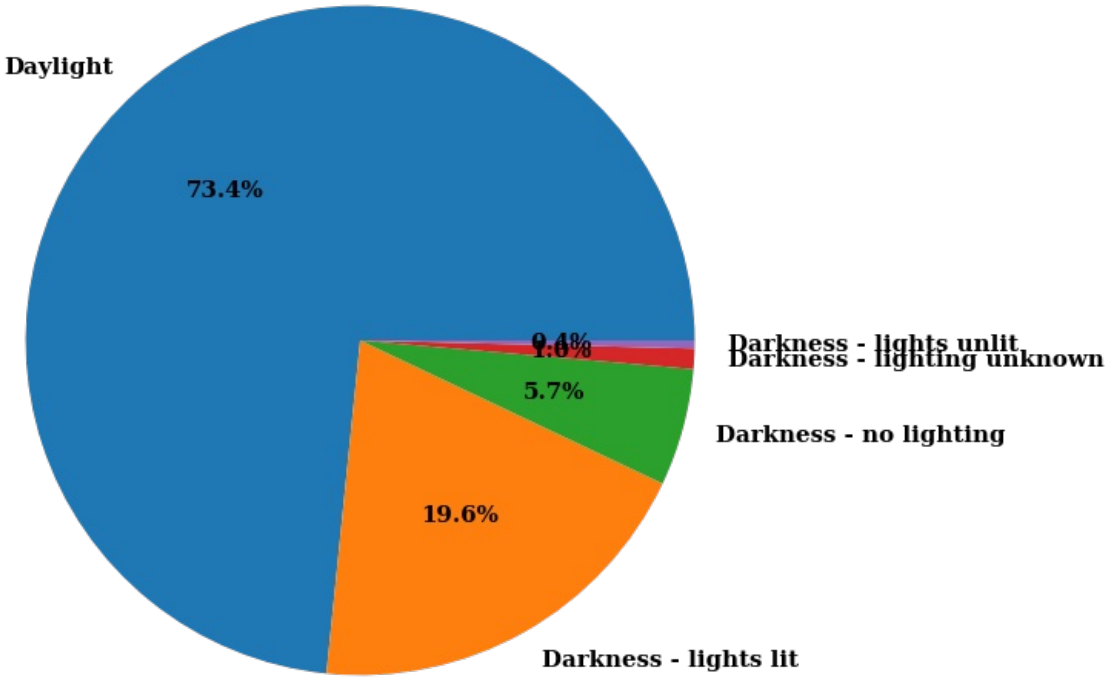


In [33]:
```
for i in categorical:
    plt.figure(figsize=(20,10))
    plt.pie(df[i].value_counts(), labels=df[i].value_counts().index, autopct='%1.1f%%', textprops={'fontsize':
                                        'color': 'black',
                                        'weight': 'bold',
                                        'family': 'serif' })
    hfont = {'fontname':'serif', 'weight': 'bold'}
    plt.title(i, size=20, **hfont)
    plt.show()
```

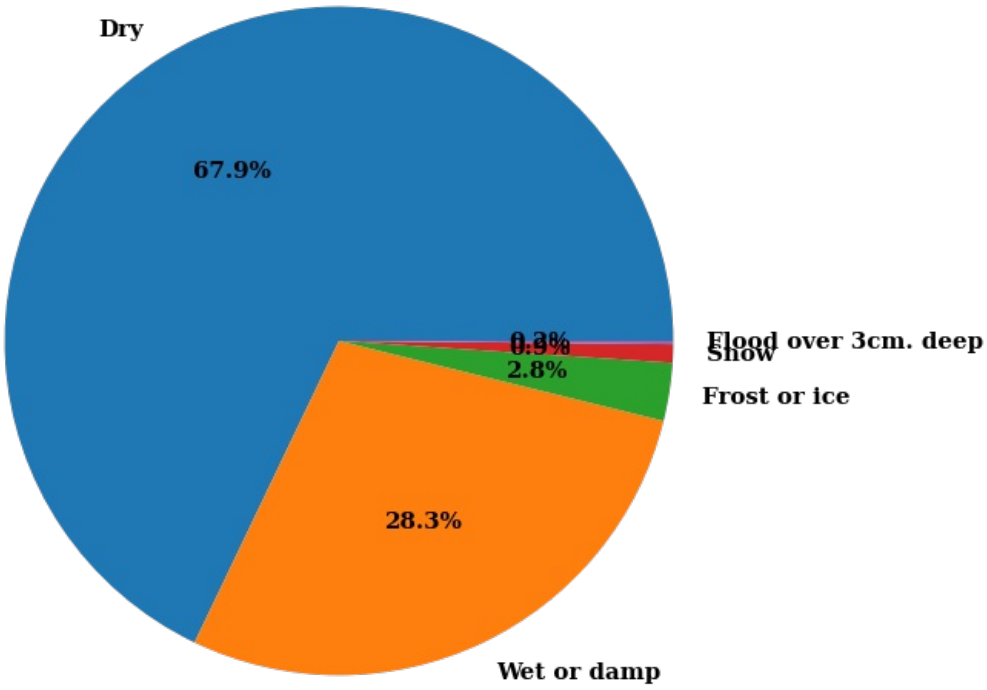
Accident_Severity



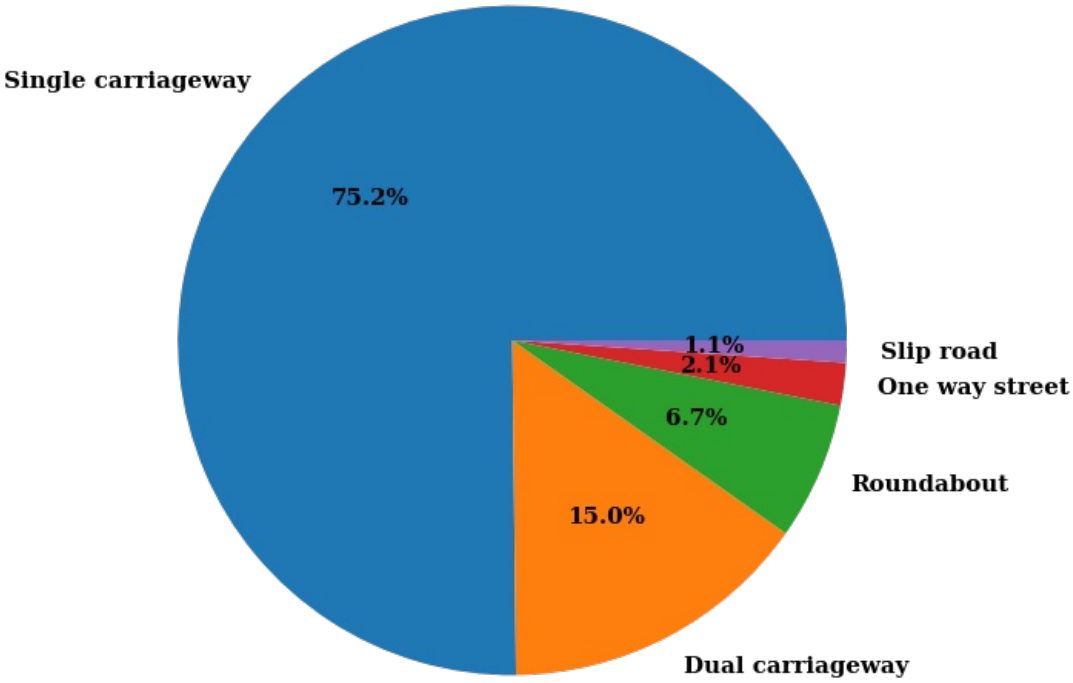
Light_Conditions

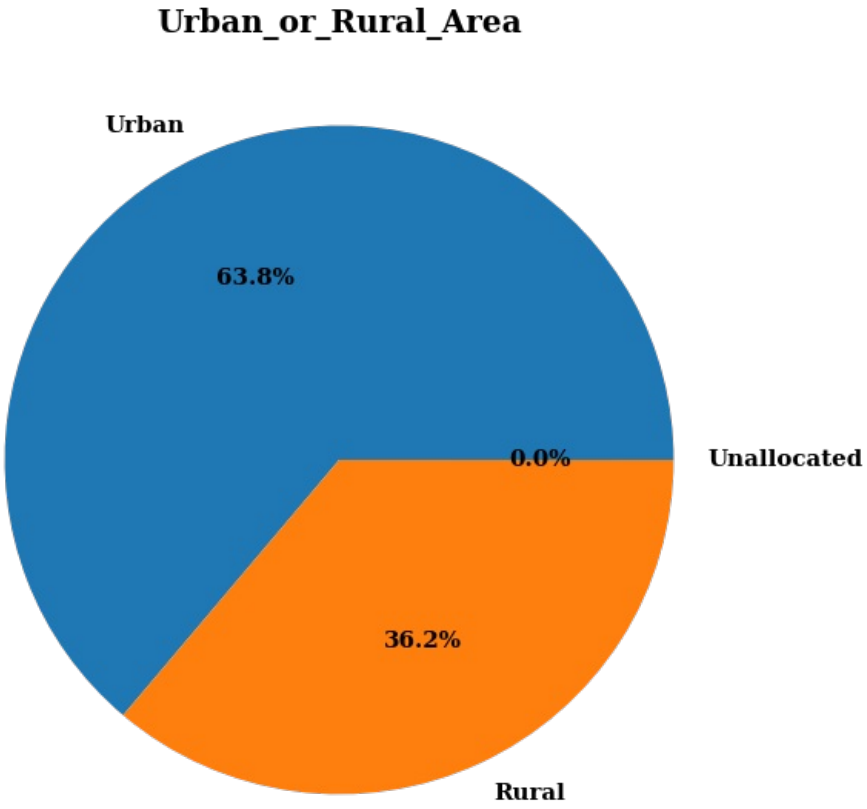


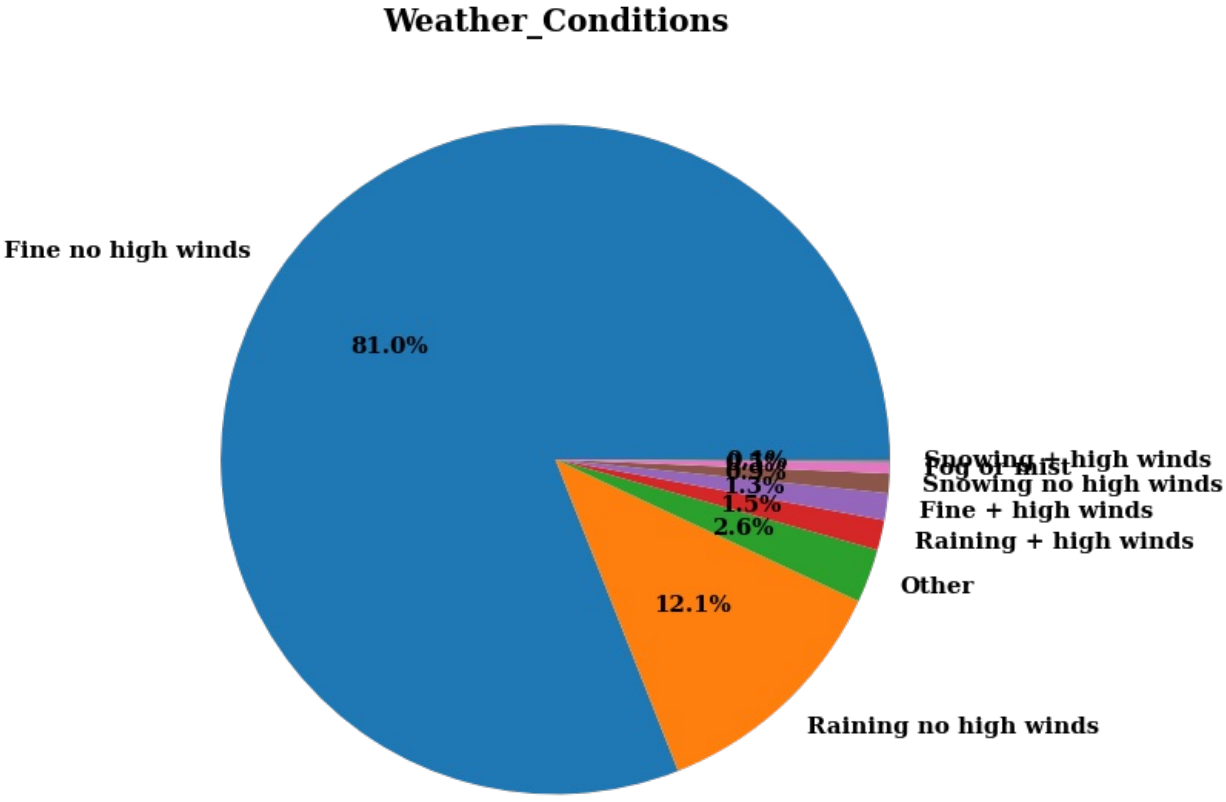
Road_Surface_Conditions



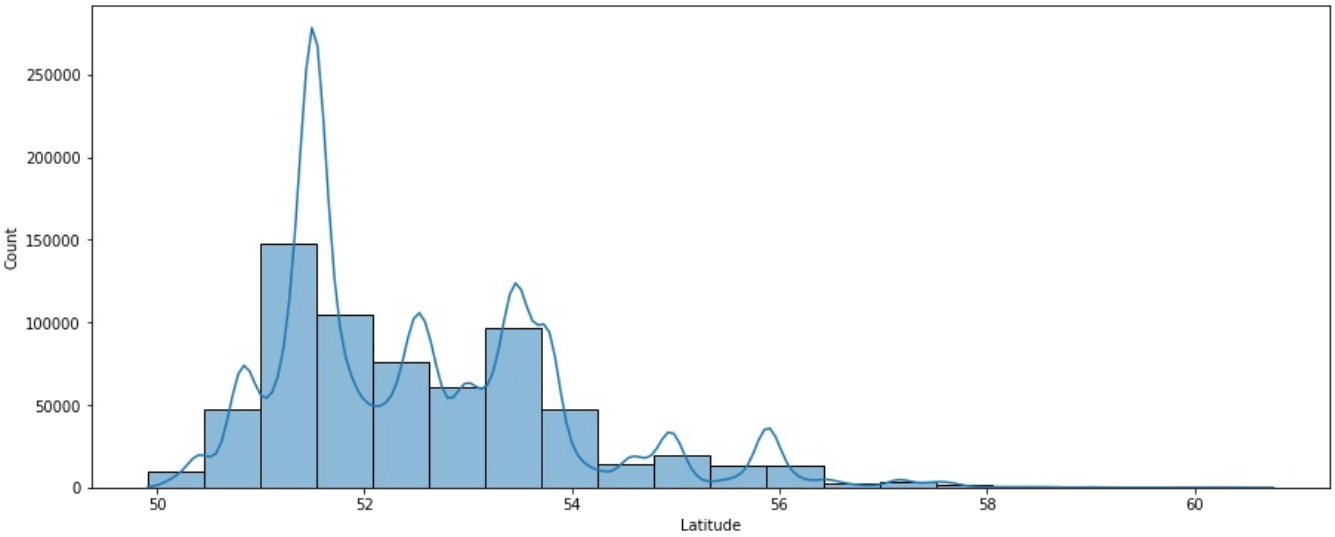
Road_Type

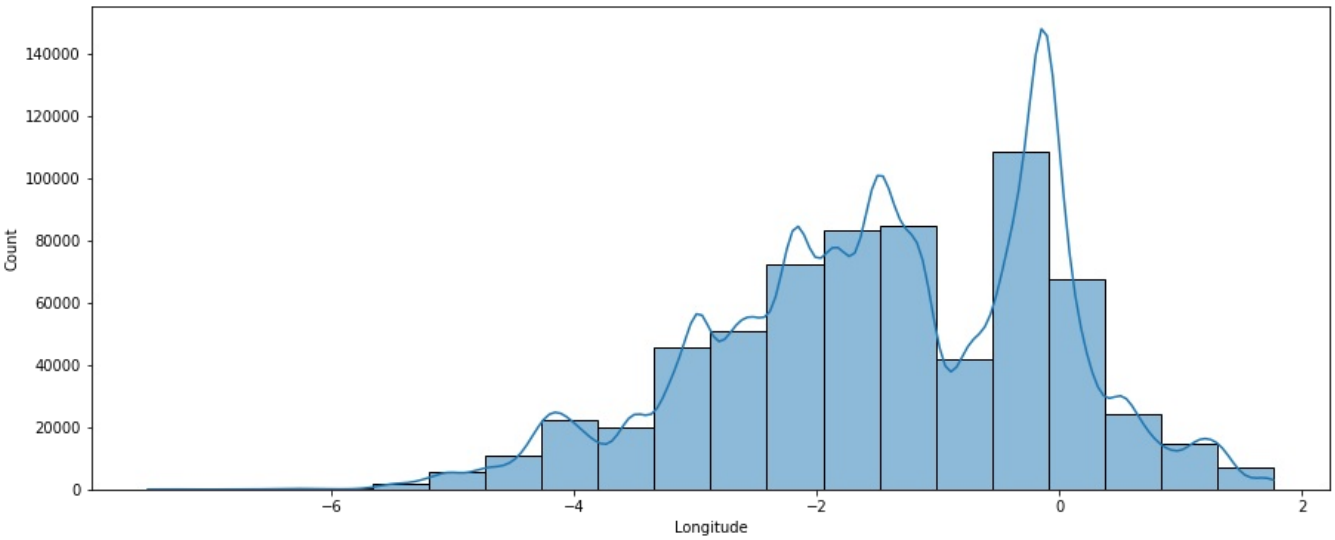


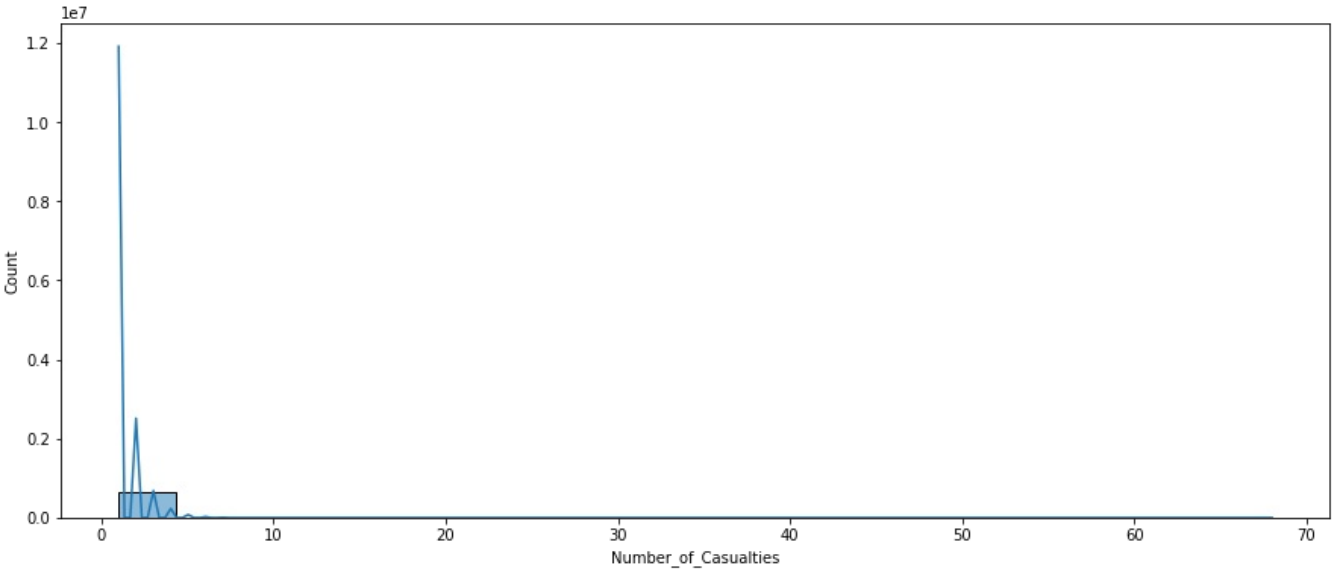


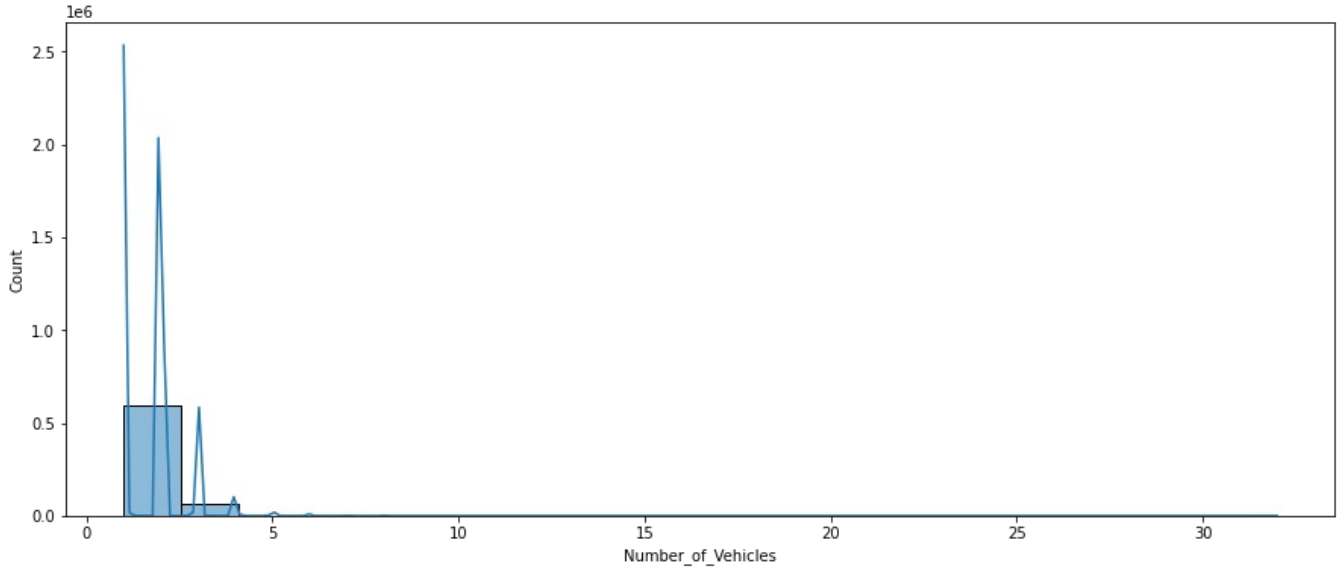


In [34]:
```
for i in continuous:
    plt.figure(figsize=(15,6))
    sns.histplot(df[i], bins = 20, kde = True, palette='hls')
    plt.show()
```

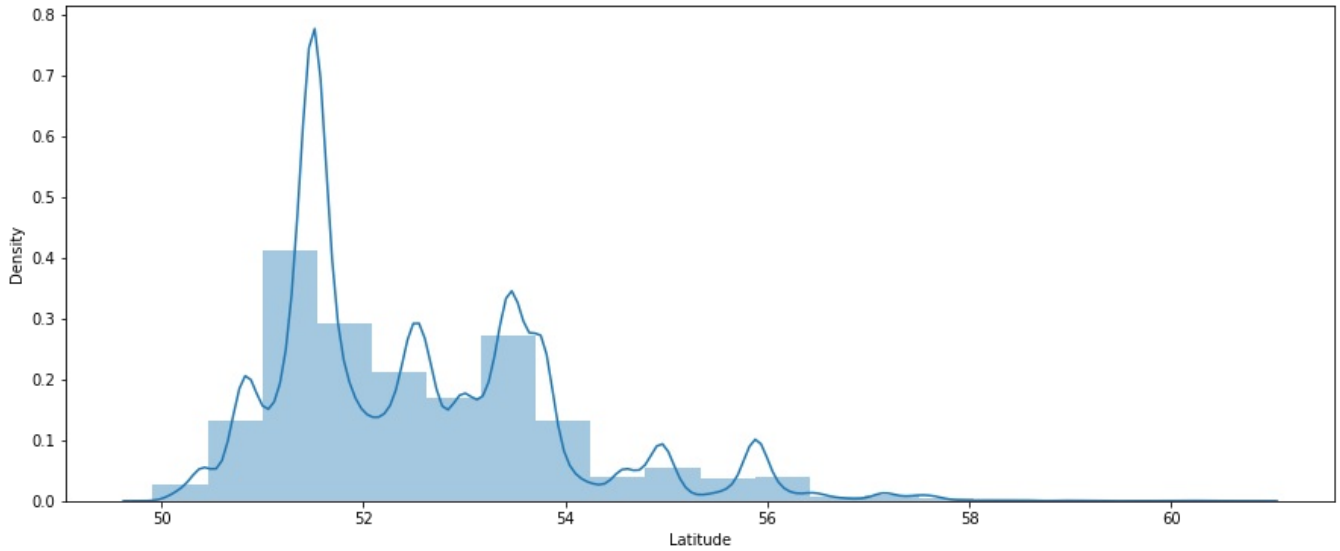

Count
Latitude
250000
200000
150000
100000
50000
0
50
52
54
56
58
60

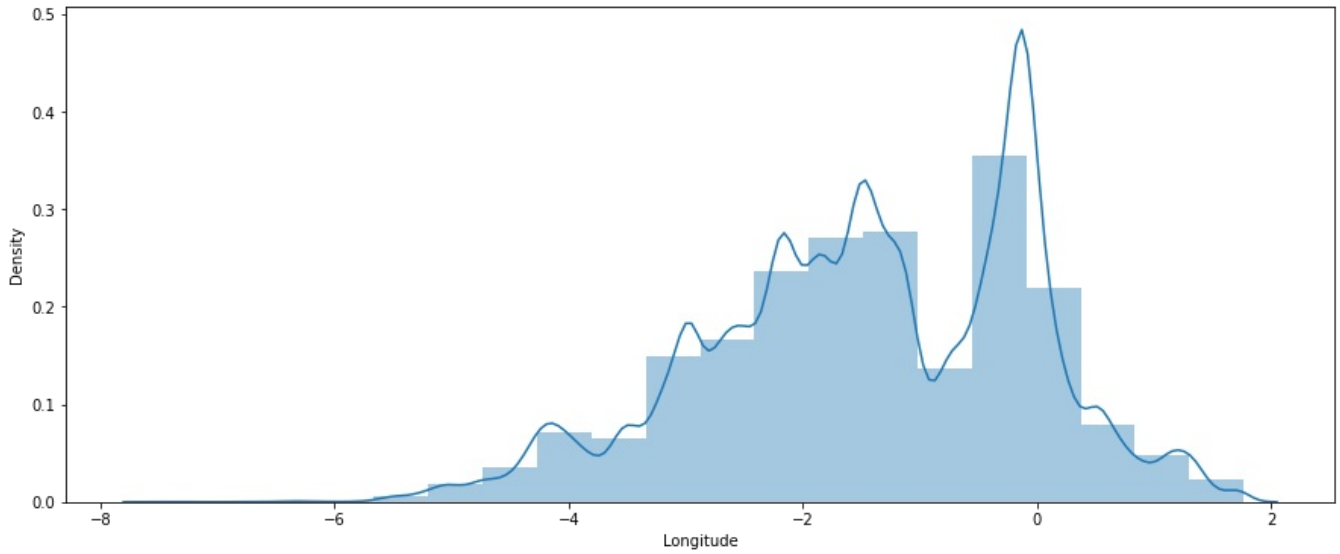

Count
Longitude
140000
120000
100000
80000
60000
40000
20000
0
−6
−4
−2
0
2

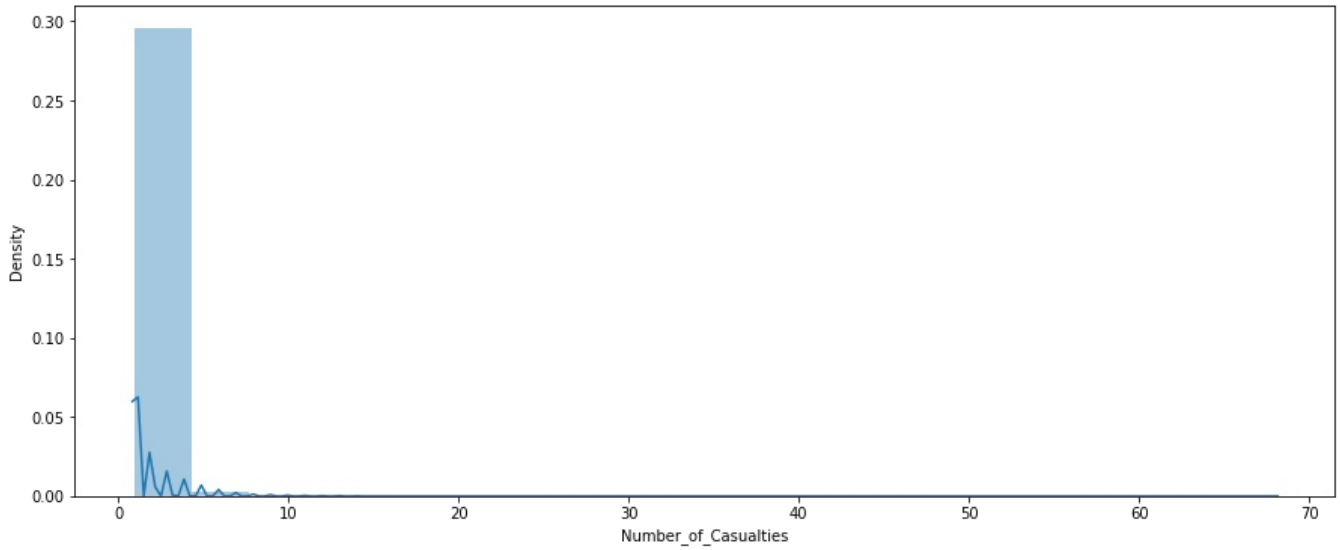

1e7
Count
Number_of_Casualties
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50
60
70

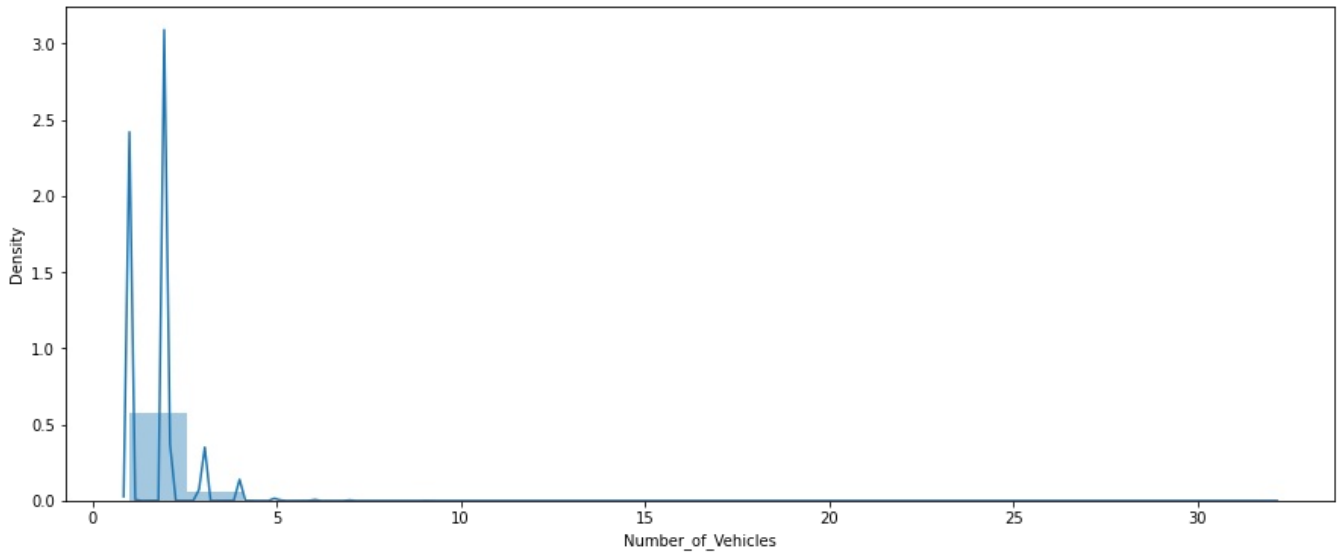


In [35]:
```
for i in continuous:
    plt.figure(figsize=(15,6))
    sns.distplot(df[i], bins = 20, kde = True)
    plt.show()
```

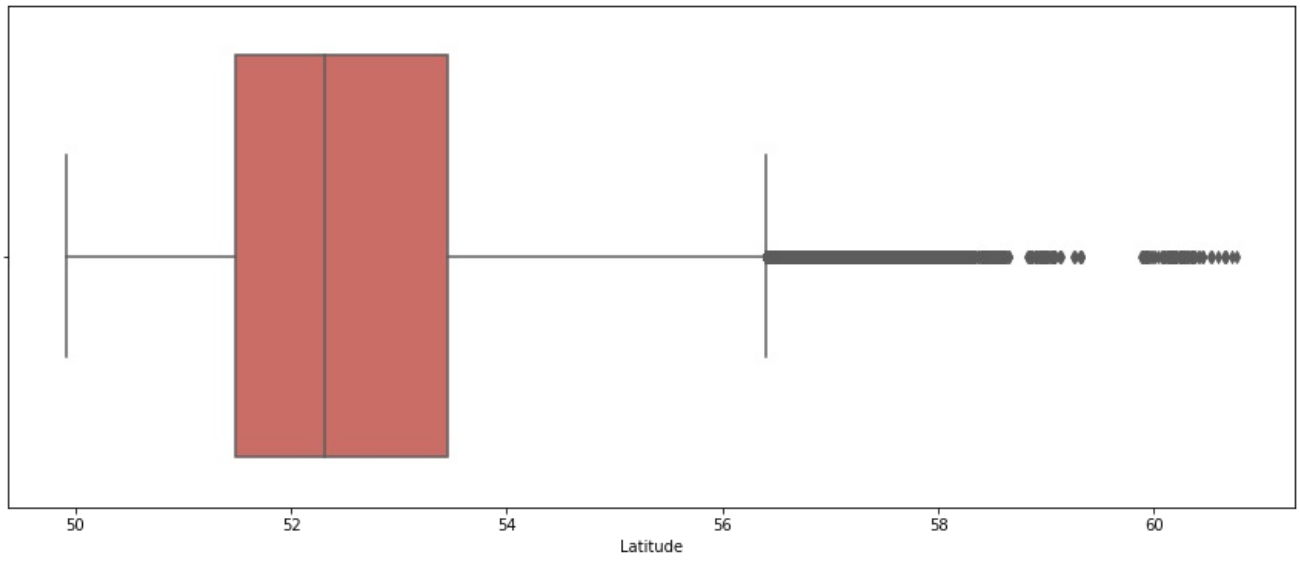


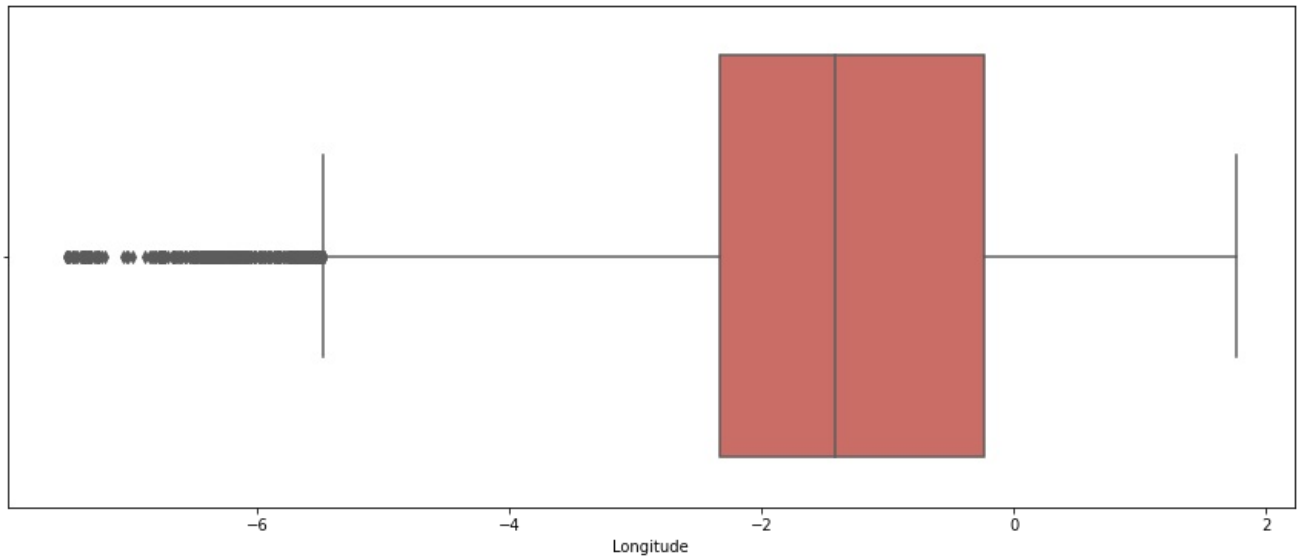


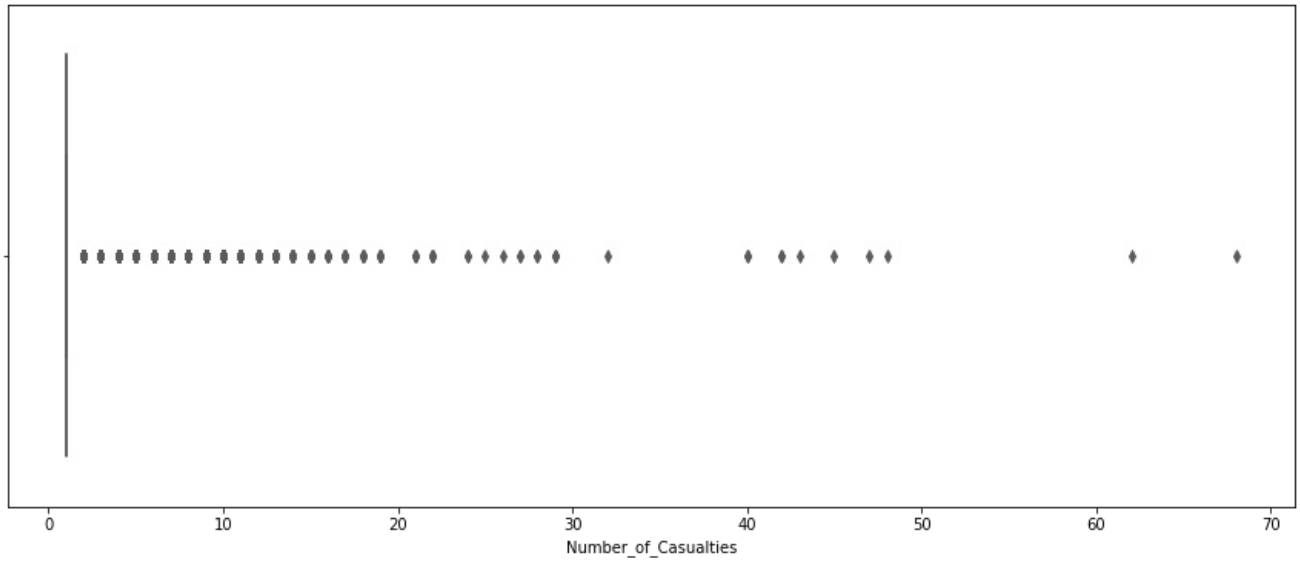


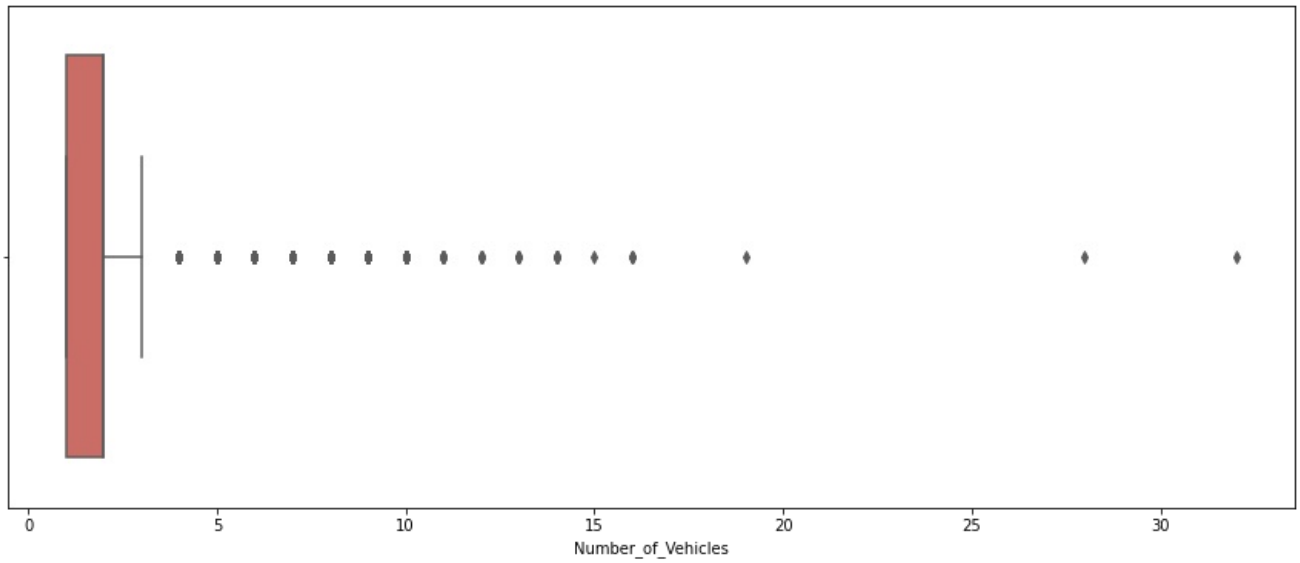


In [36]:
```
for i in continuous:
    plt.figure(figsize=(15,6))
    sns.boxplot(i, data = df, palette='hls')
    plt.show()
```

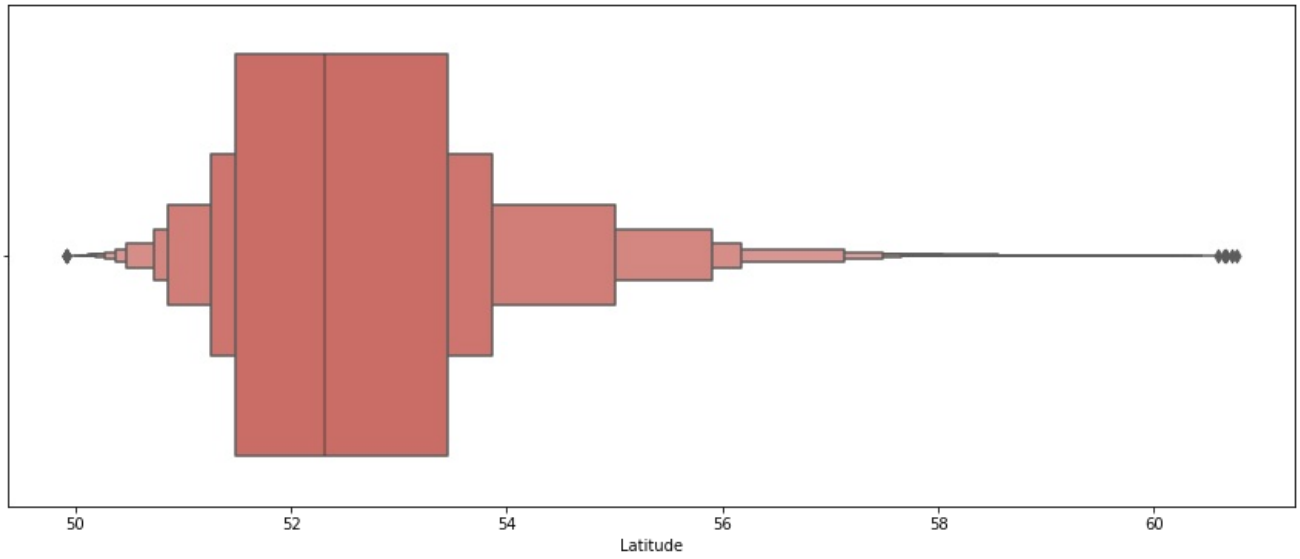


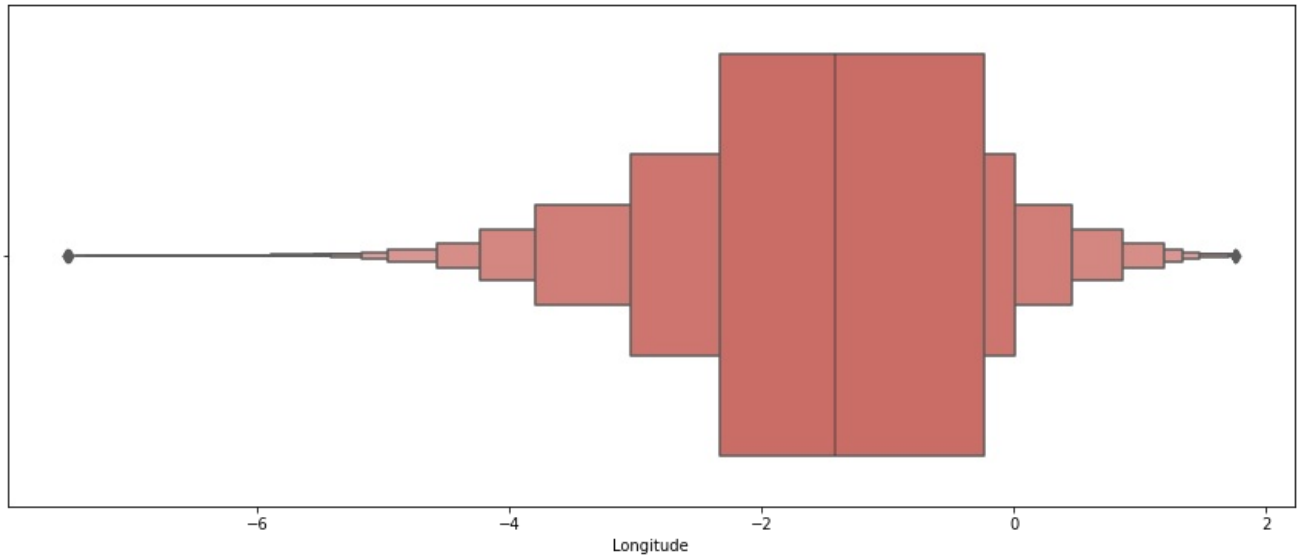


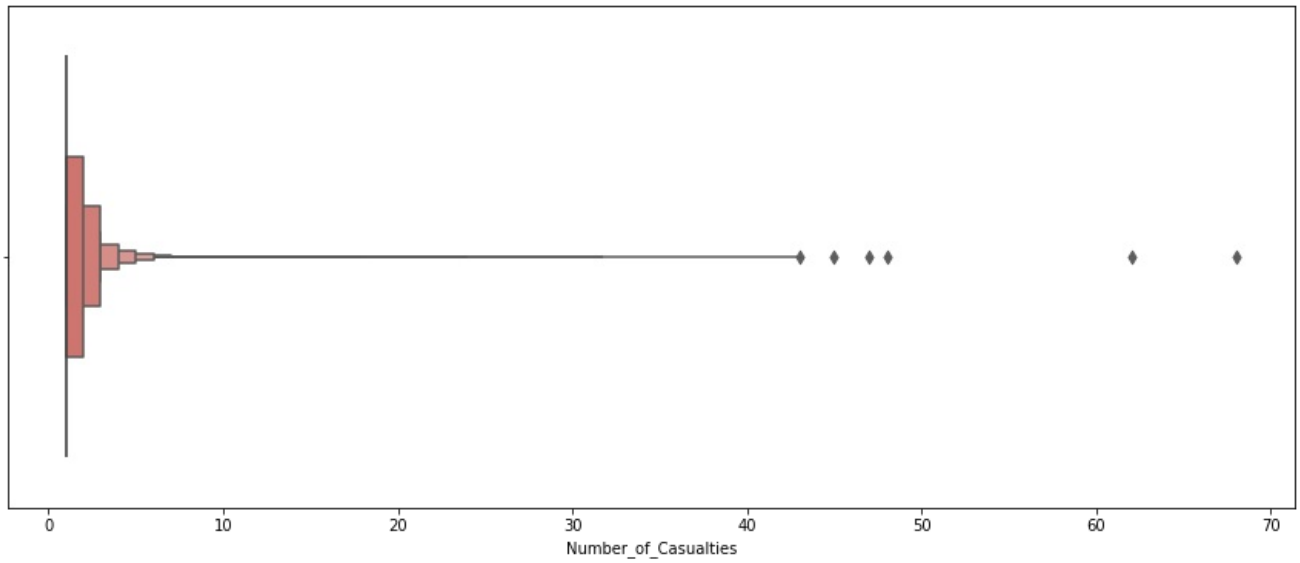


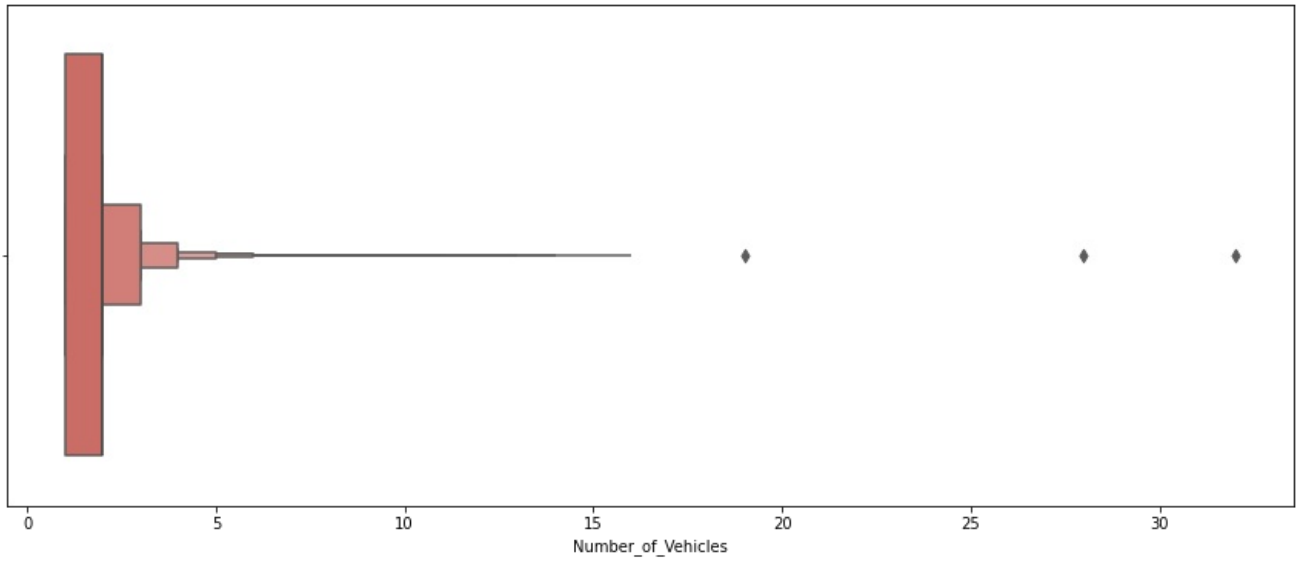


In [37]:
```
for i in continuous:
    plt.figure(figsize=(15,6))
    sns.boxenplot(i, data = df, palette='hls')
    plt.show()
```

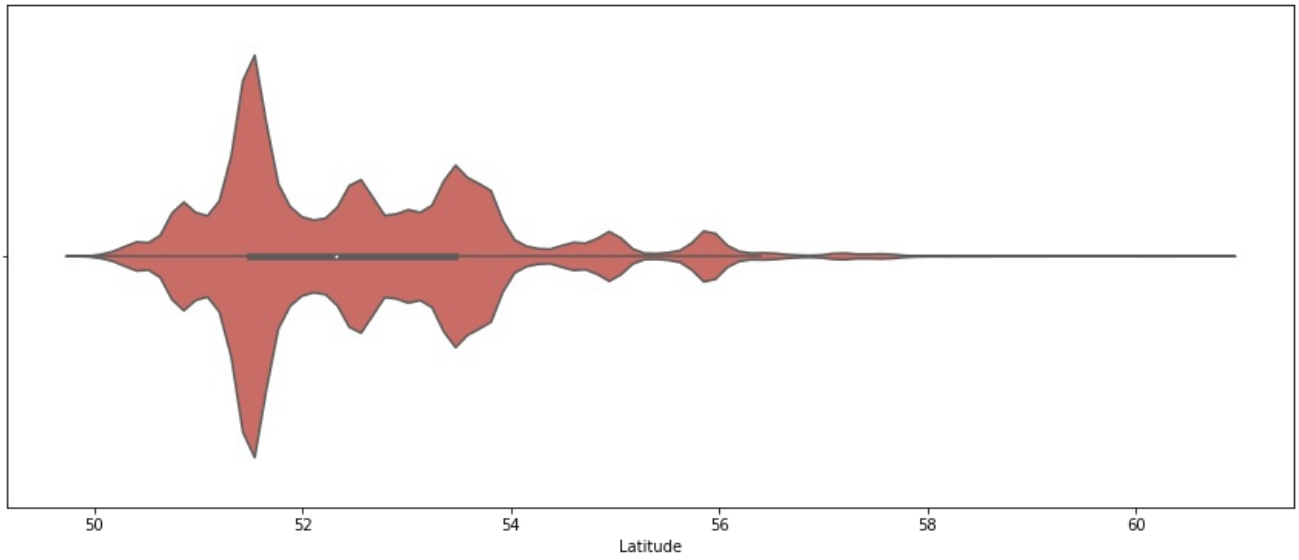

50
52
54
56
58
60
Latitude

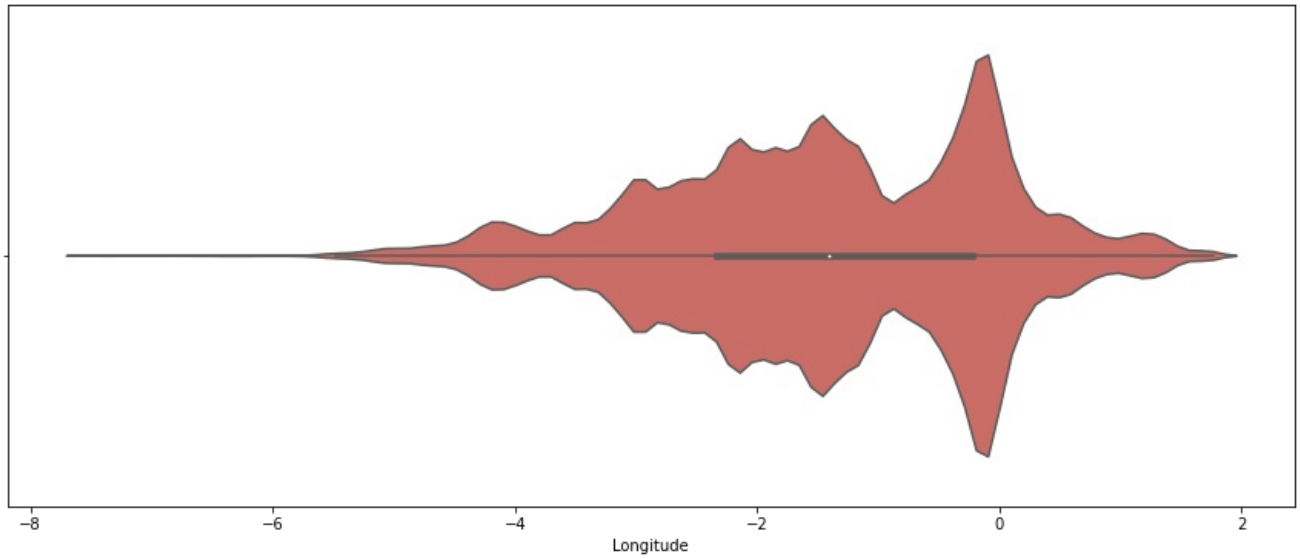

−6
−4
−2
0
2
Longitude

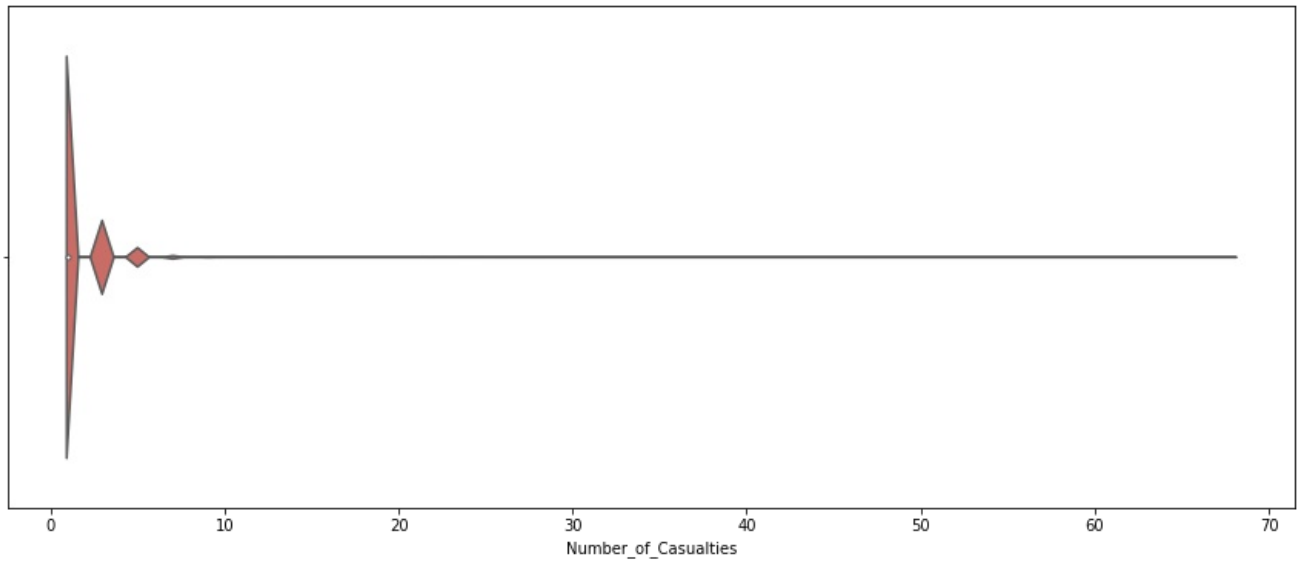

0
10
20
30
40
50
60
70
Number_of_Casualties

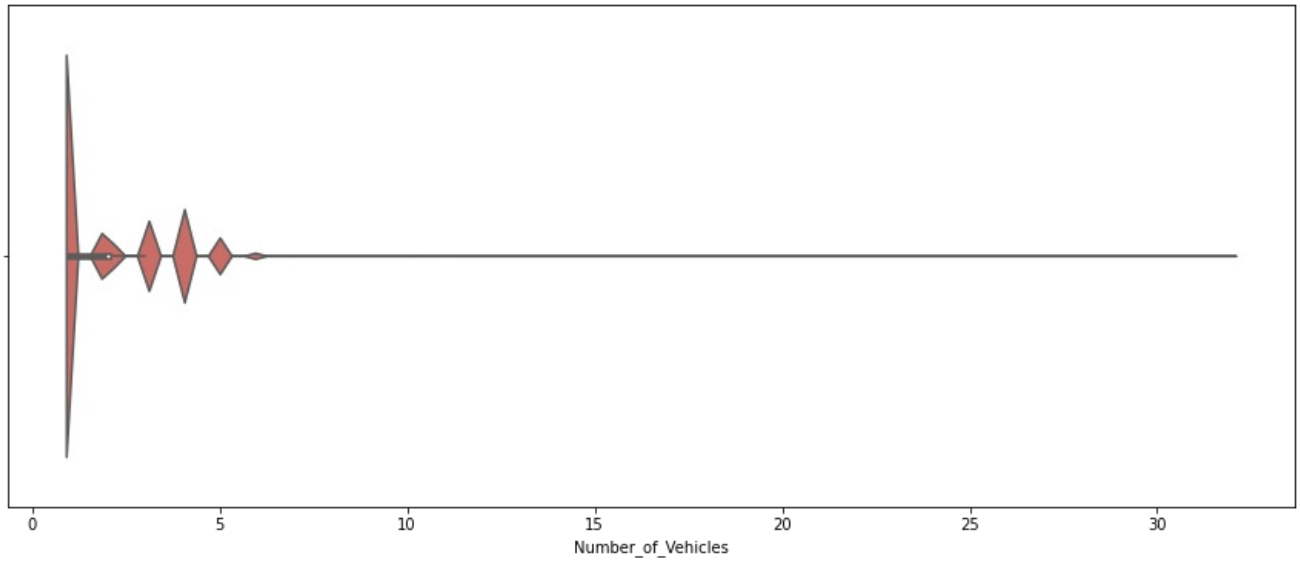


In [38]:
```
for i in continuous:
    plt.figure(figsize=(15,6))
    sns.violinplot(i, data = df, palette='hls')
    plt.show()
```

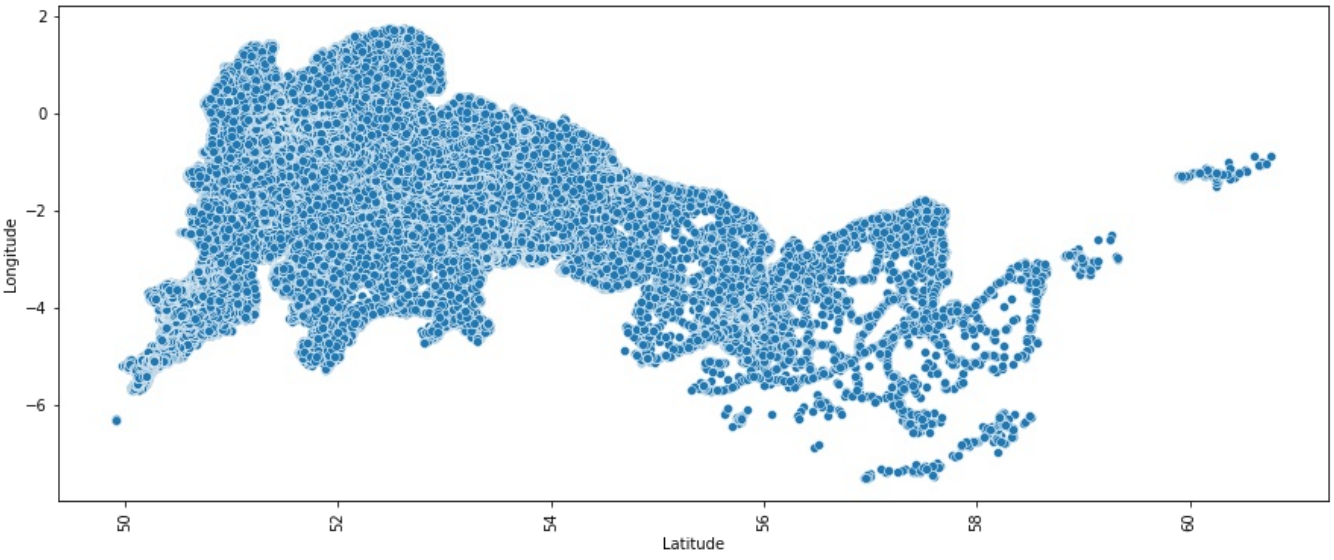


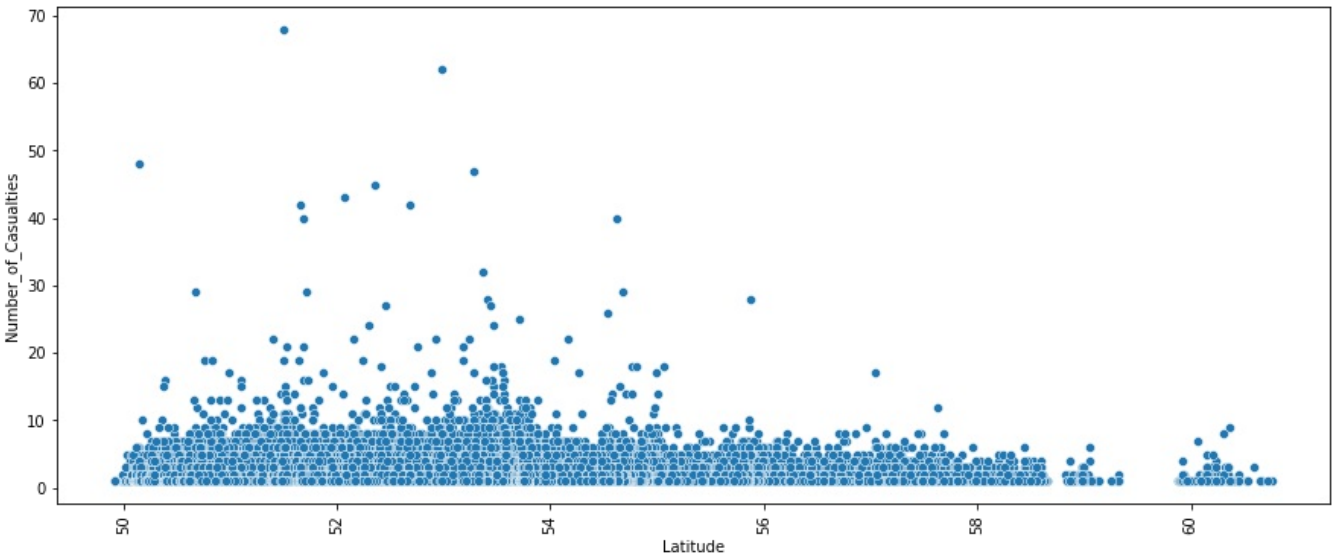


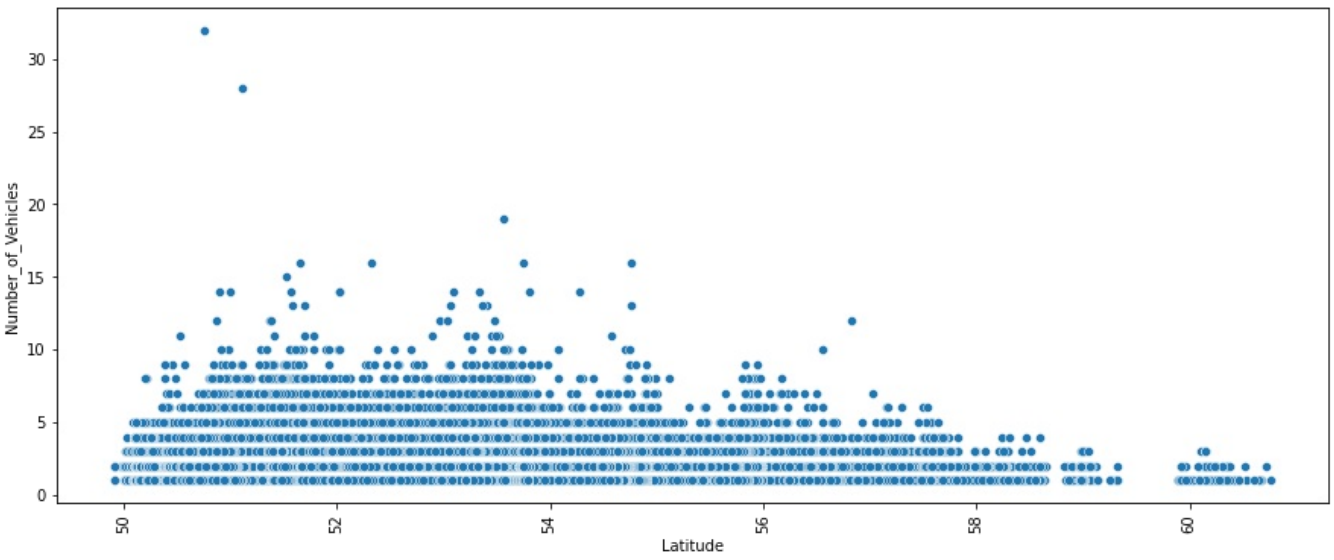


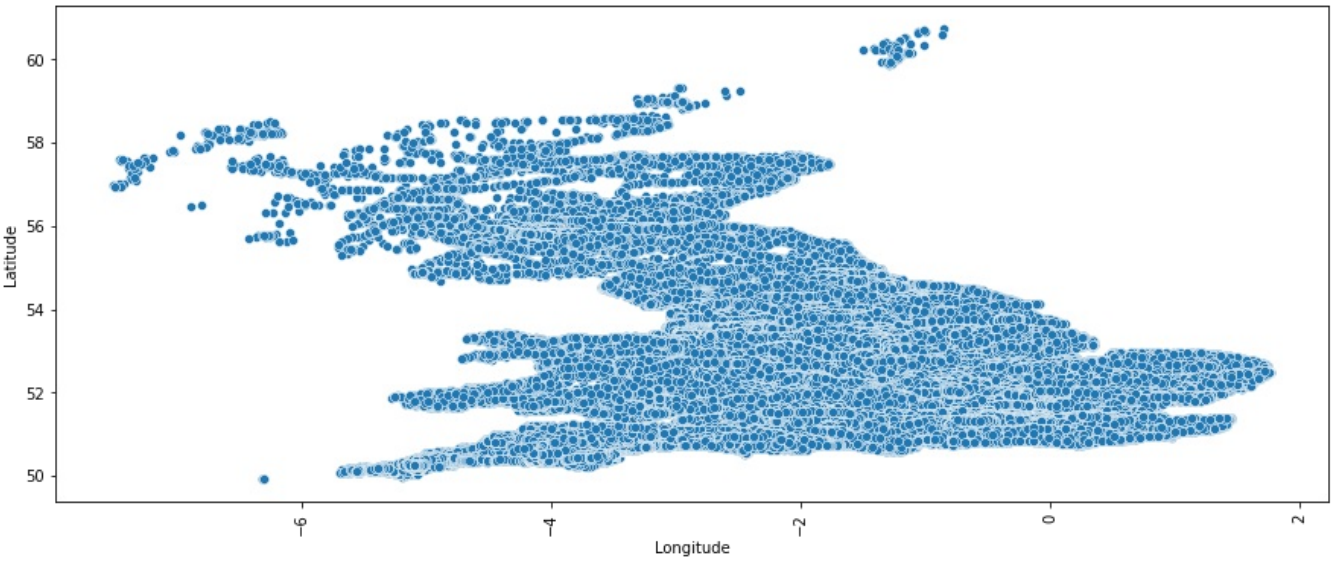


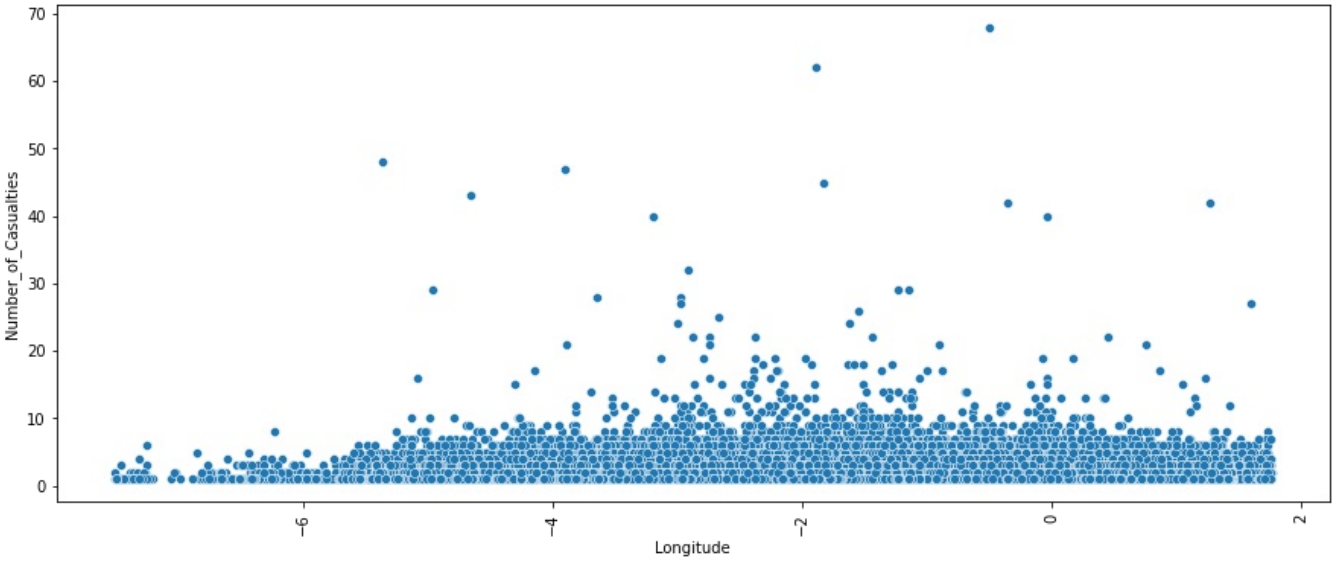
In [39]:
```
for i in continuous:
    for j in continuous:
        if i != j:
            plt.figure(figsize=(15,6))
            sns.scatterplot(x = i, y = j, data = df, ci = None, palette='hls')
            plt.xticks(rotation = 90)
            plt.show()
```

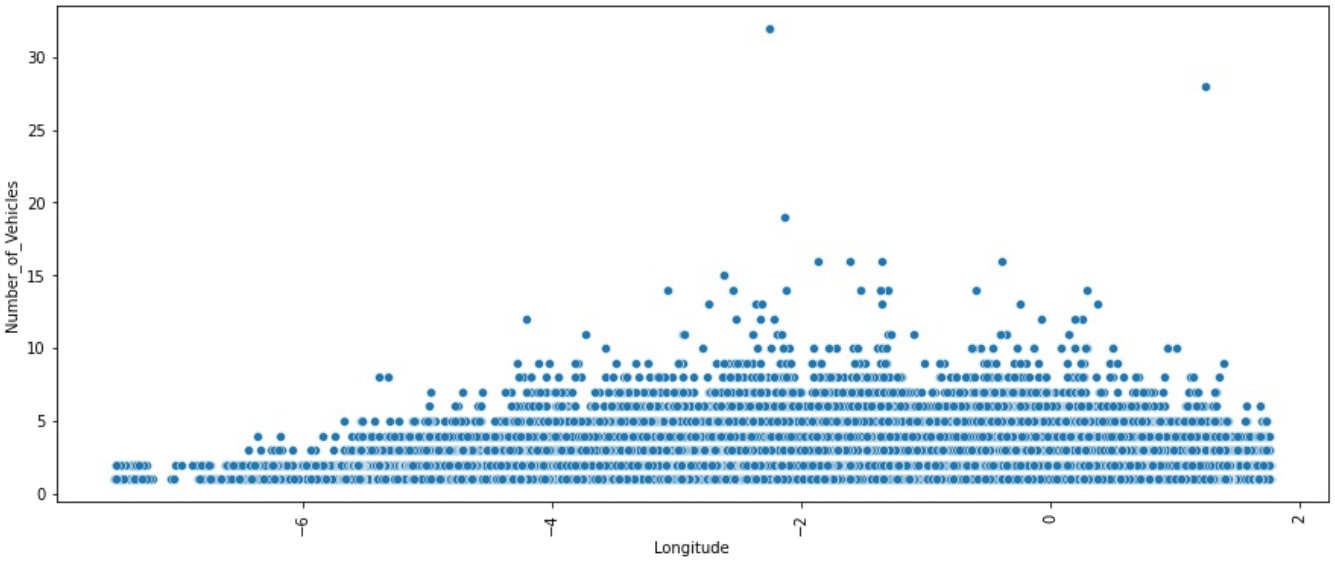

Longitude
Latitude
2
0
−2
−4
−6
50
52
54
56
58
60

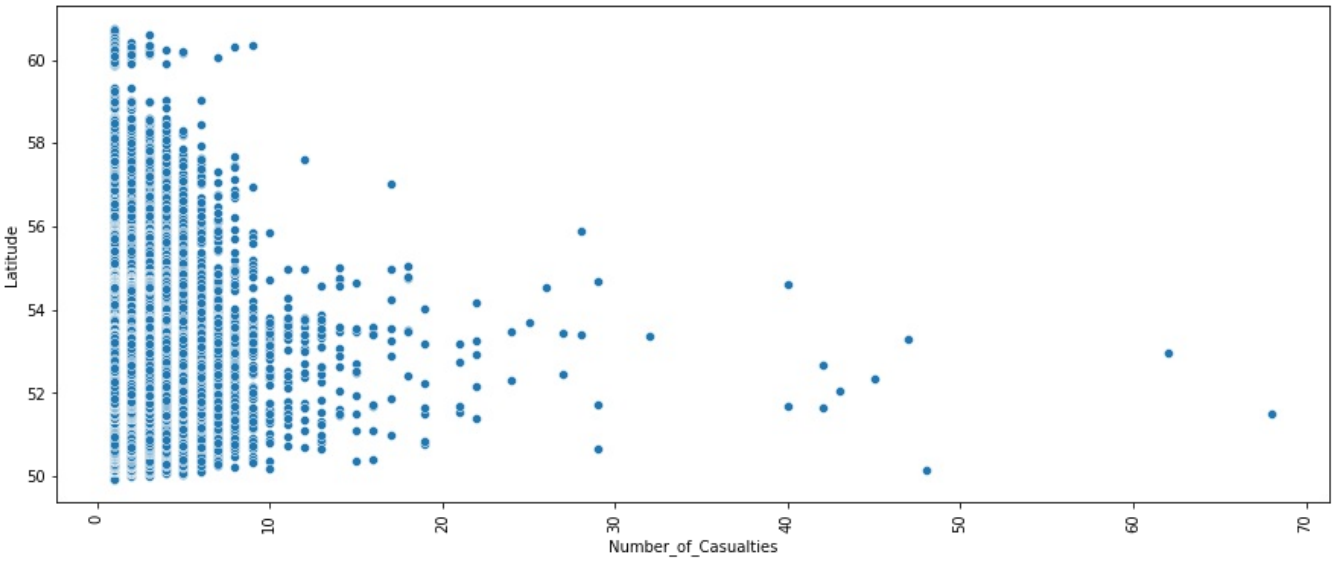

Number_of_Casualties
Latitude
70
60
50
40
30
20
10
0
50
52
54
56
58
60

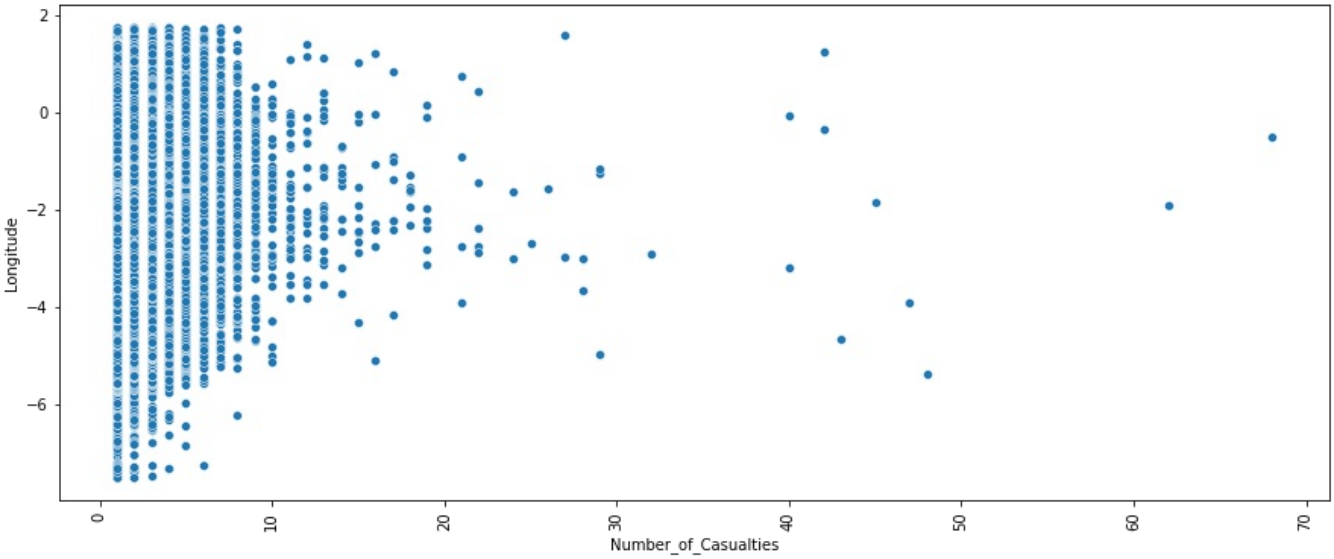

Number_of_Vehicles
Latitude
30
25
20
15
10
5
0
50
52
54
56
58
60

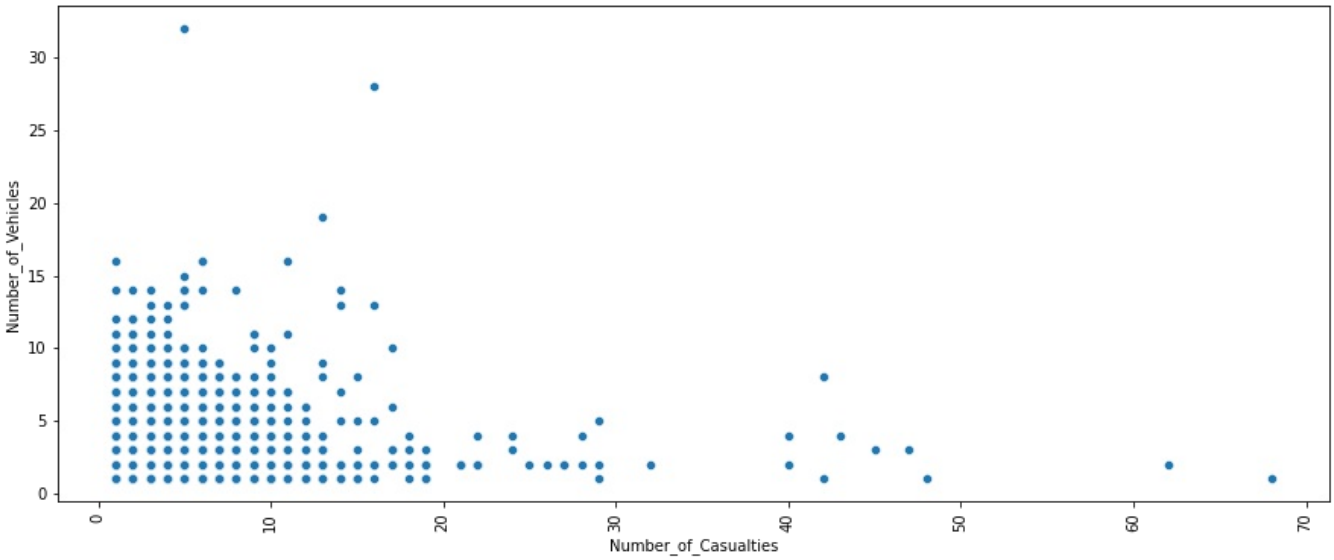

Latitude
60
58
56
54
52
50
−6
−4
−2
0
2
Longitude

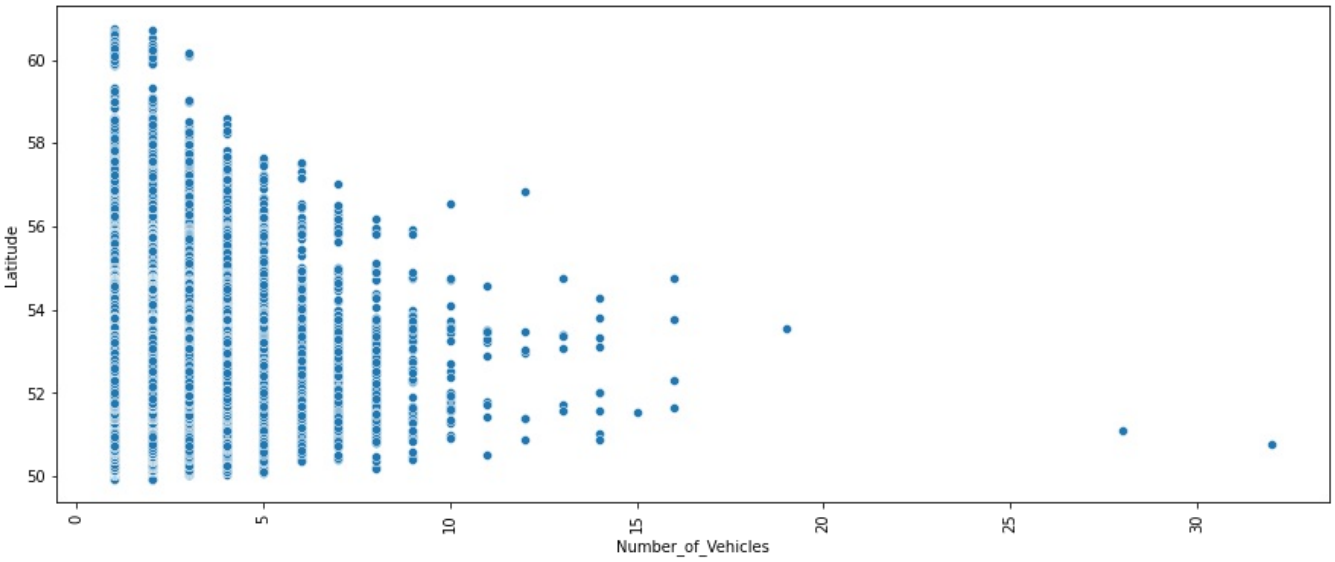

Number_of_Casualties
70
60
50
40
30
20
10
0
−6
−4
−2
0
2
Longitude

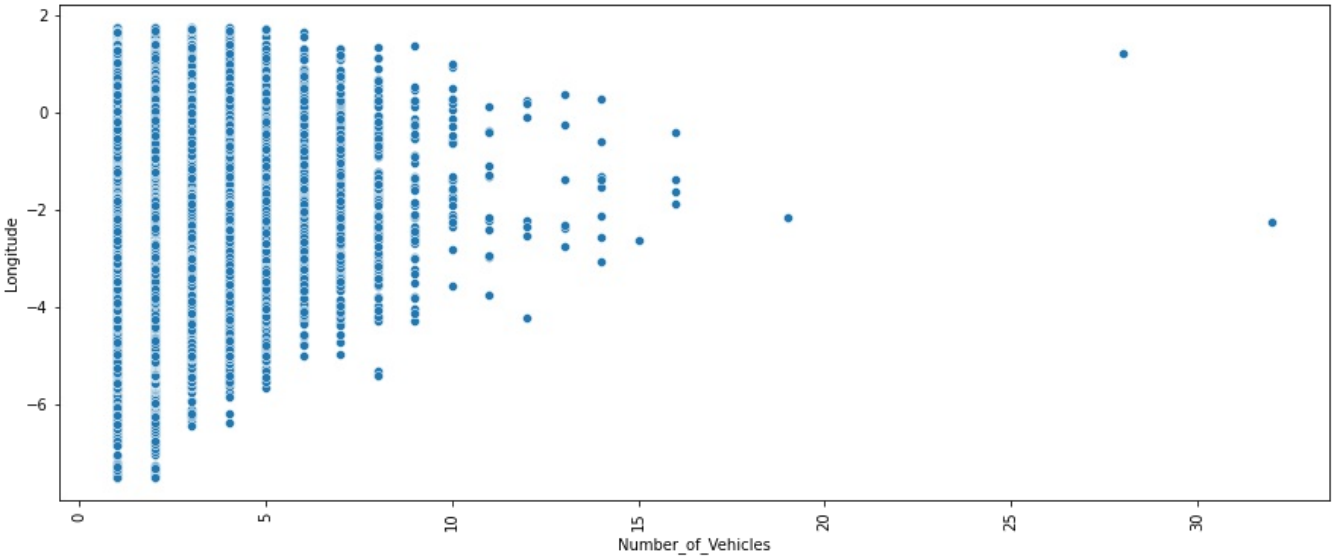

Number_of_Vehicles
30
25
20
15
10
5
0
−6
−4
−2
0
2
Longitude

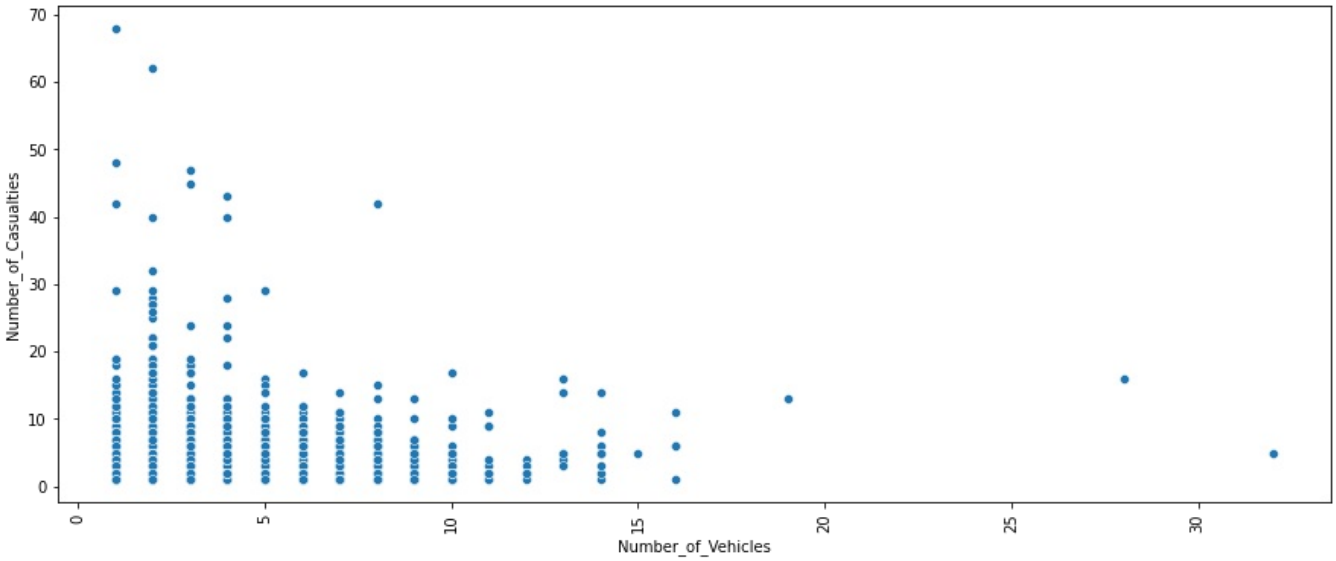

Latitude
Number_of_Casualties


Longitude
Number_of_Casualties

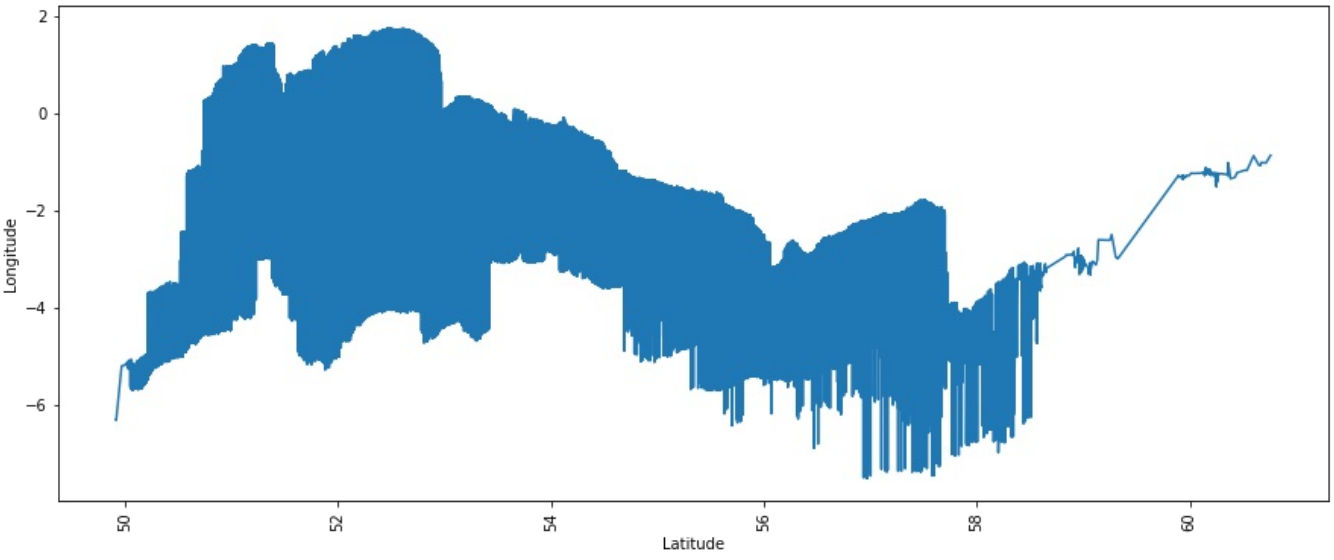

Number_of_Vehicles
Number_of_Casualties

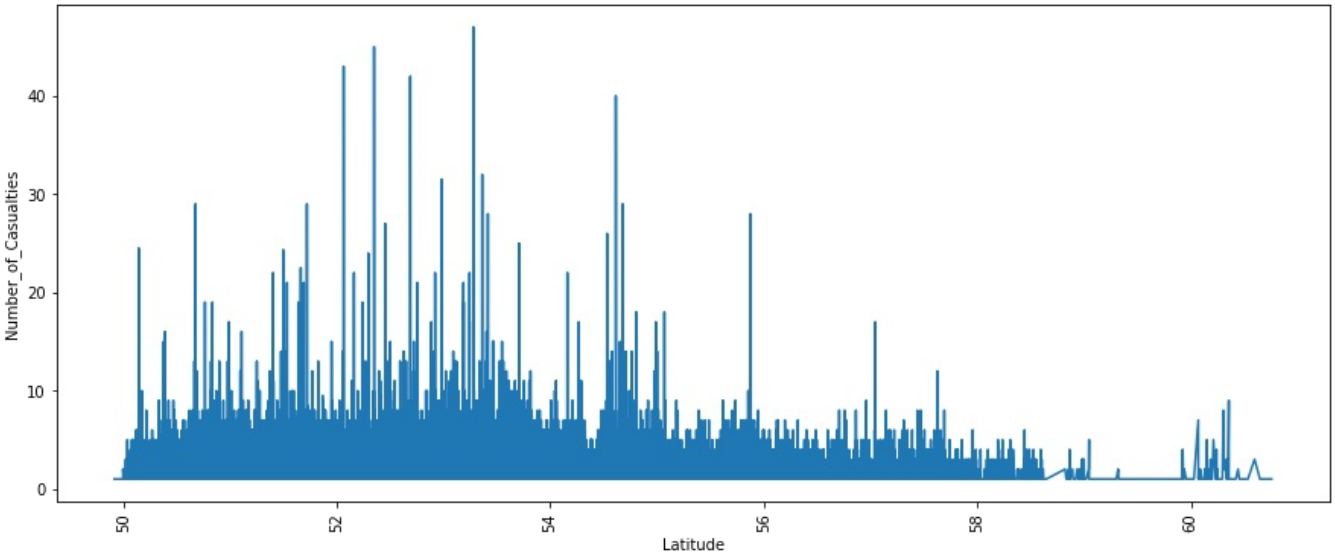


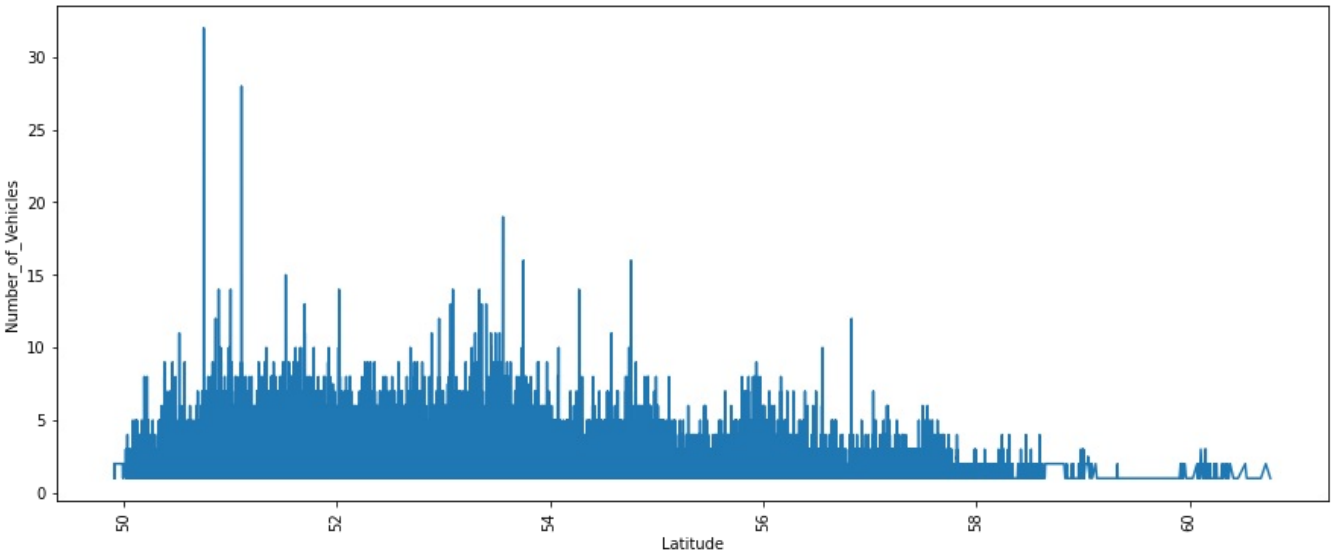


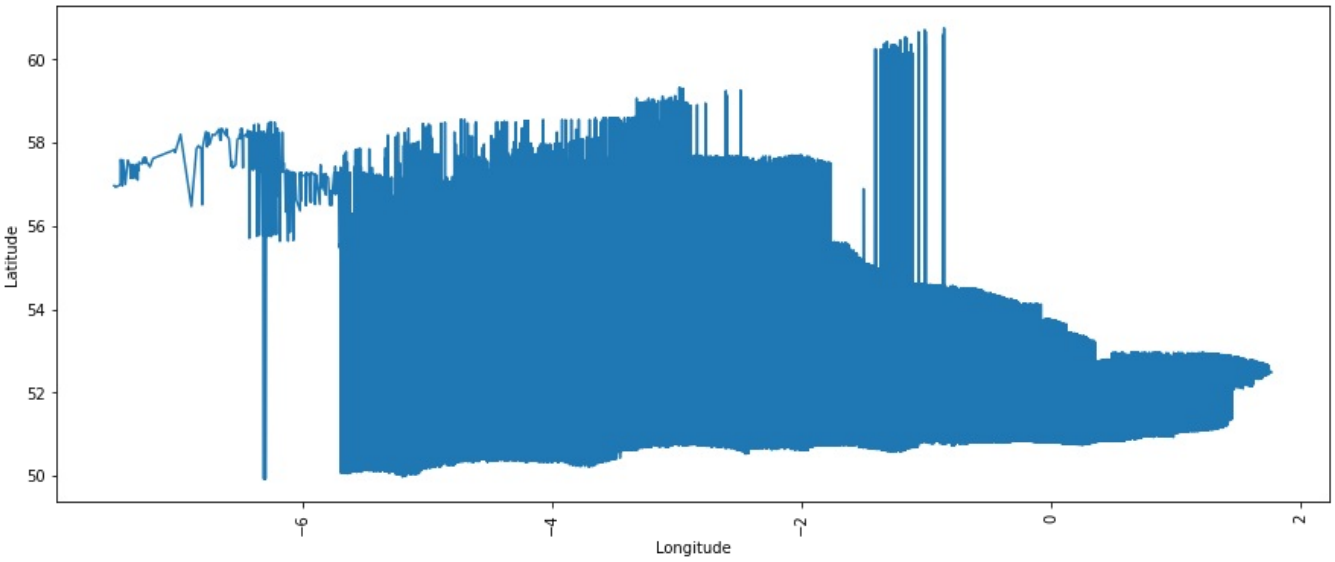


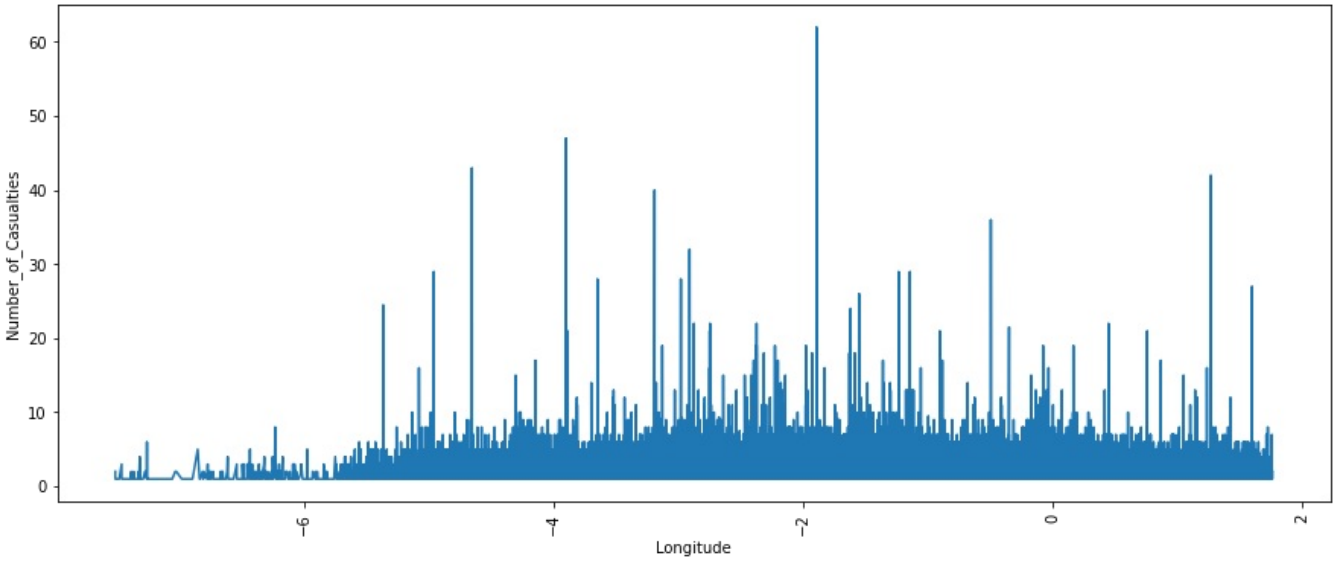
In [40]:
```
for i in continuous:
    for j in continuous:
        if i != j:
            plt.figure(figsize=(15,6))
            sns.lineplot(x = i, y = j, data = df, ci = None, palette='hls')
            plt.xticks(rotation = 90)
            plt.show()
```

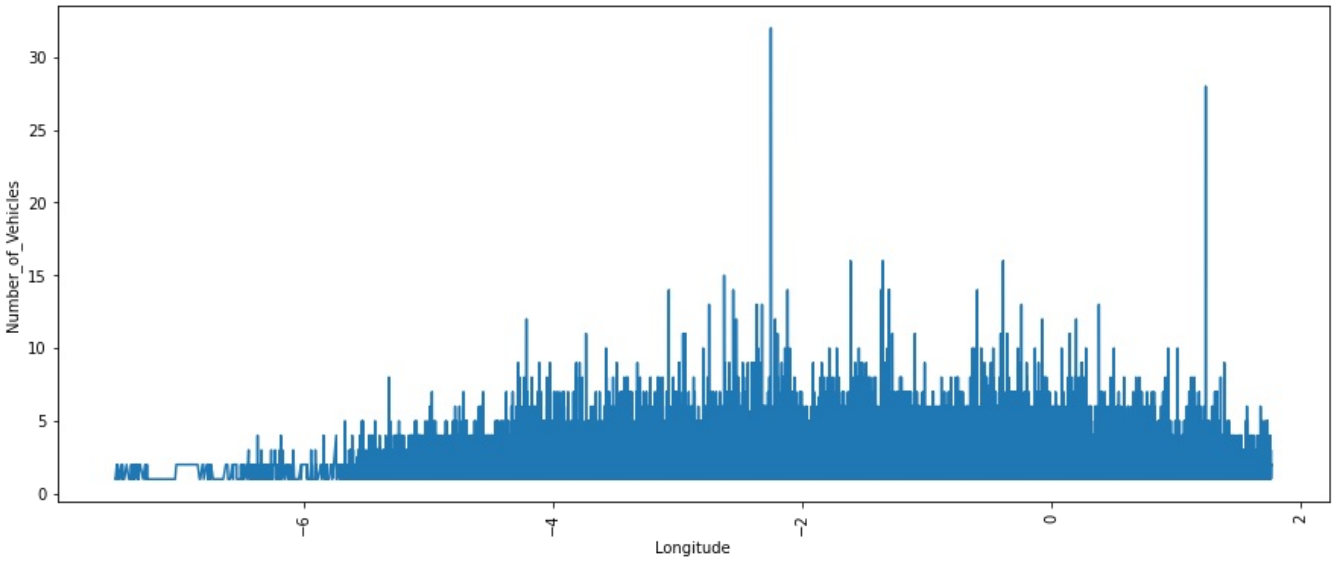

Longitude
2
0
−2
−4
−6
50
52
54
56
58
60
Latitude

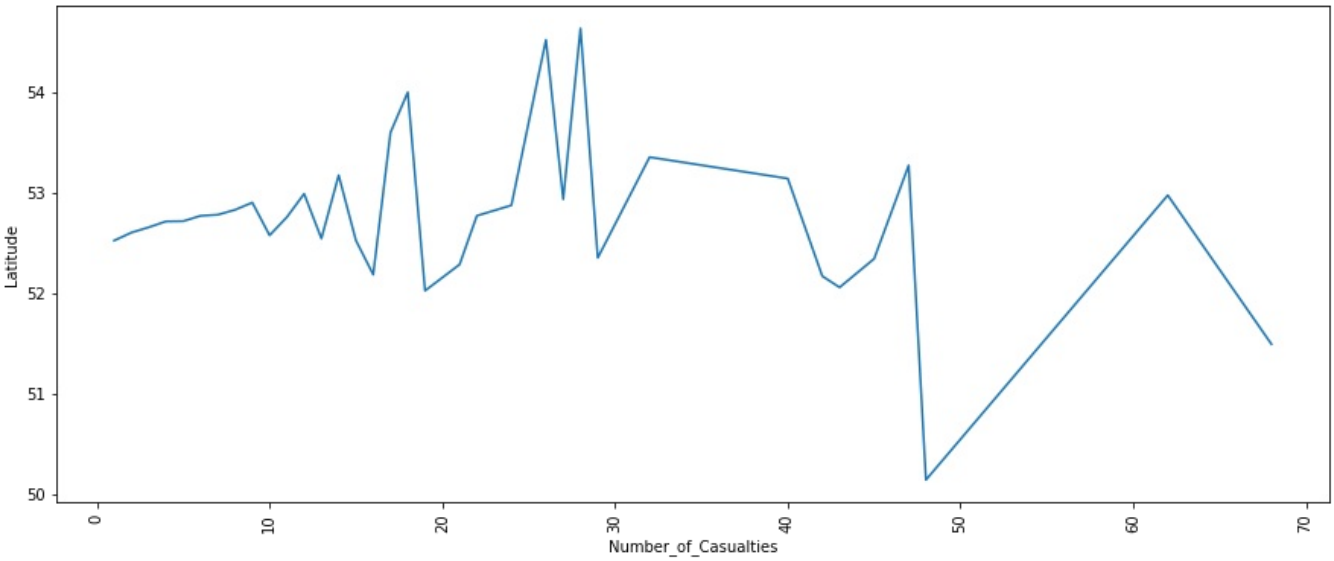

Number_of_Casualties
40
30
20
10
0
50
52
54
56
58
60
Latitude

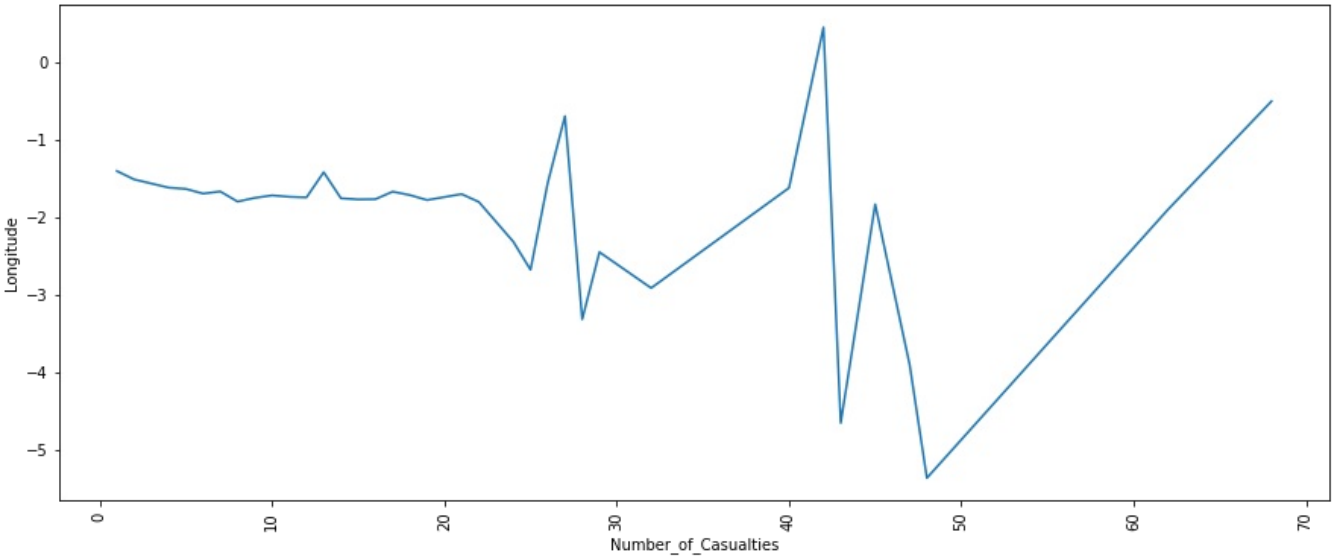

Number_of_Vehicles
30
25
20
15
10
5
0
50
52
54
56
58
60
Latitude

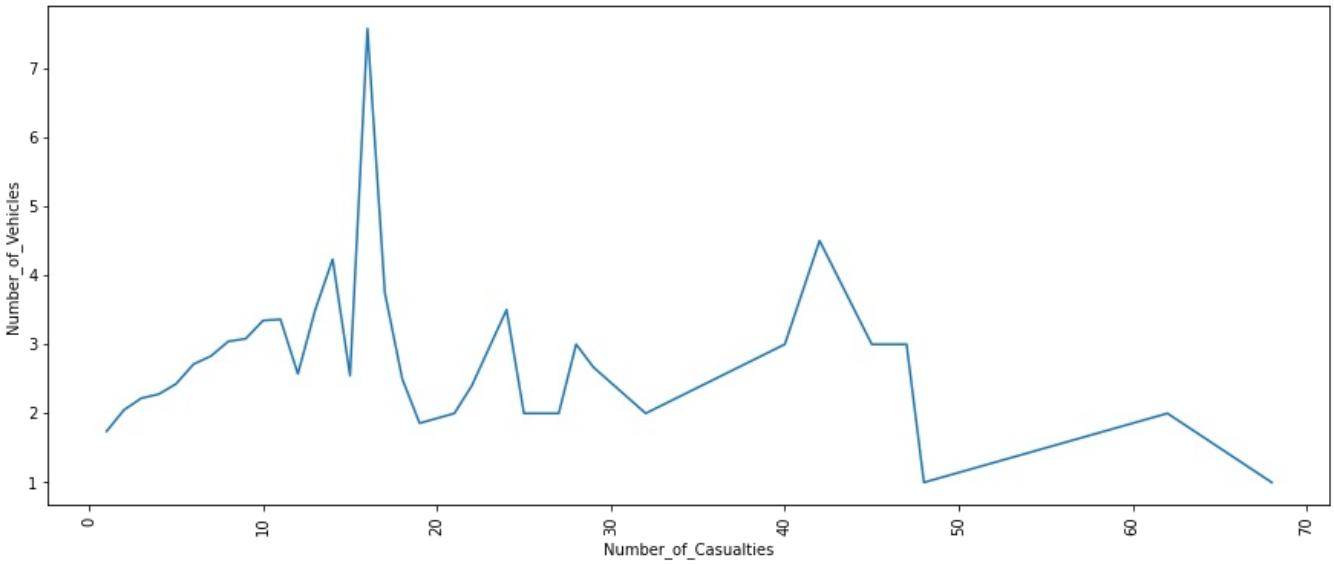

Latitude
60
58
56
54
52
50
−6
−4
−2
0
2
Longitude

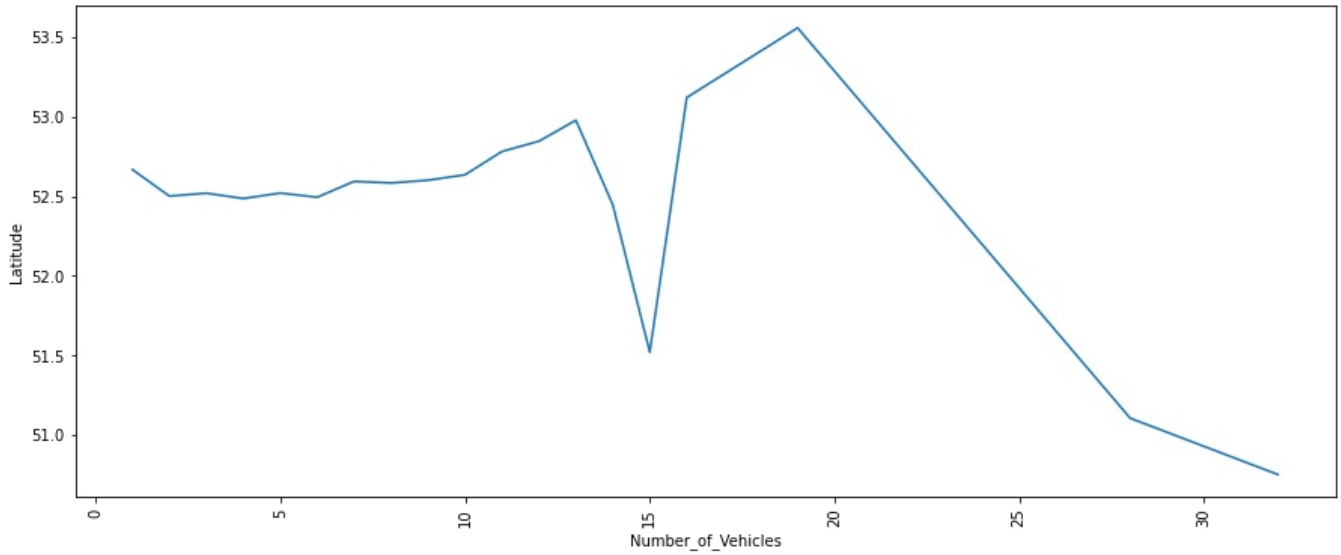

Number_of_Casualties
60
50
40
30
20
10
0
−6
−4
−2
0
2
Longitude

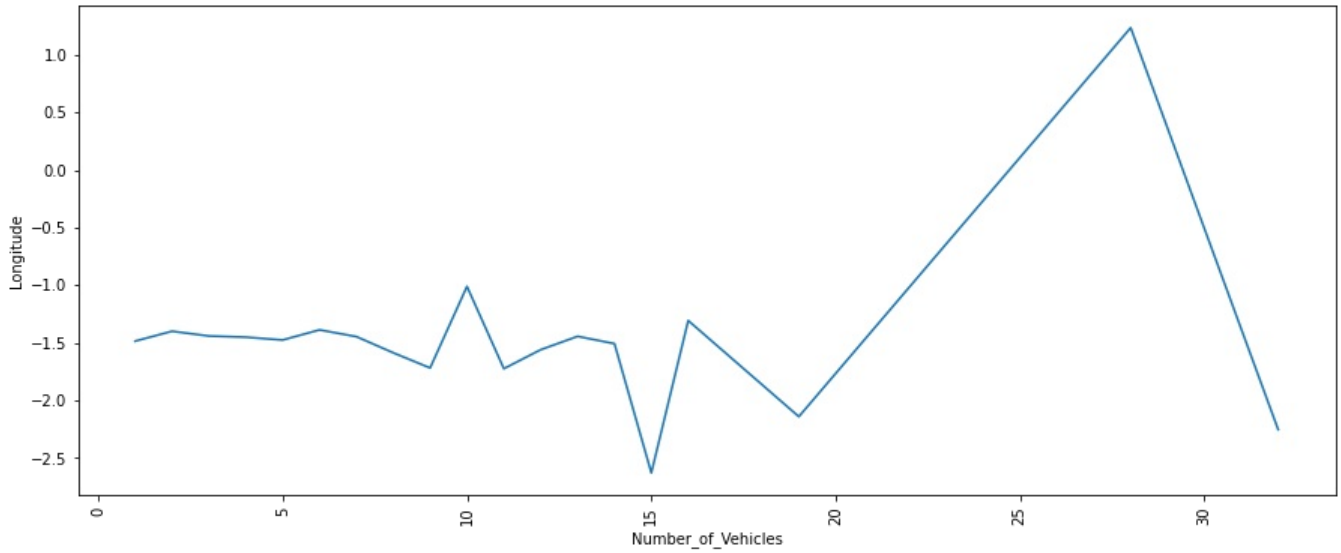

Number_of_Vehicles
30
25
20
15
10
5
0
−6
−4
−2
0
2
Longitude

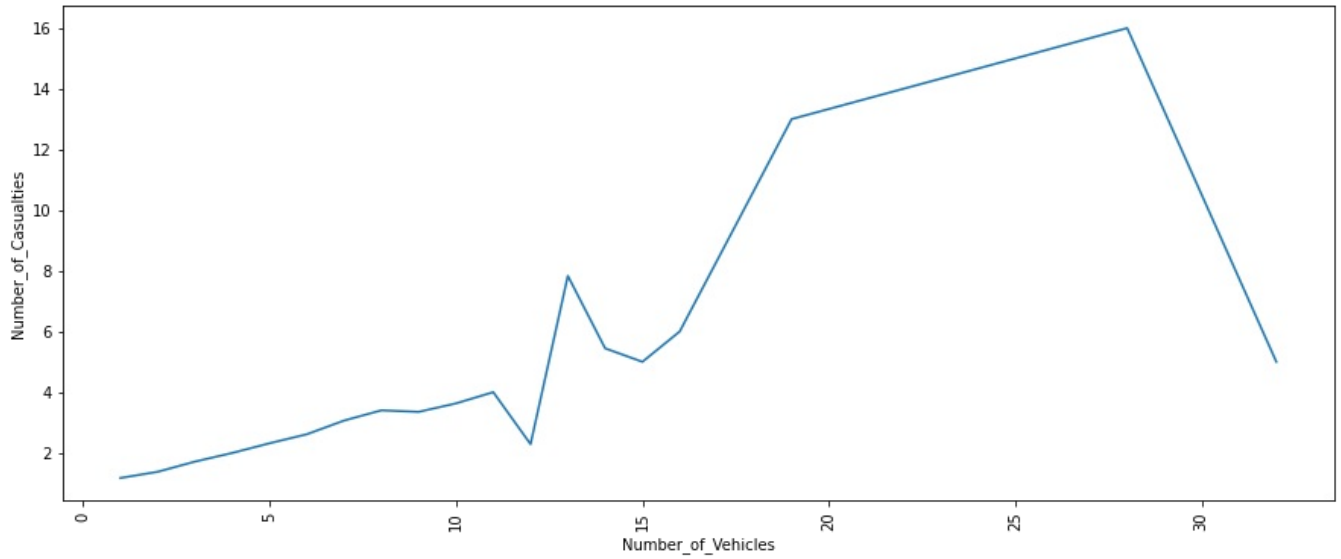

Latitude
54
53
52
51
50
0
10
20
30
40
50
60
70
Number_of_Casualties


Longitude
0
−1
−2
−3
−4
−5
0
10
20
30
40
50
60
70
Number_of_Casualties

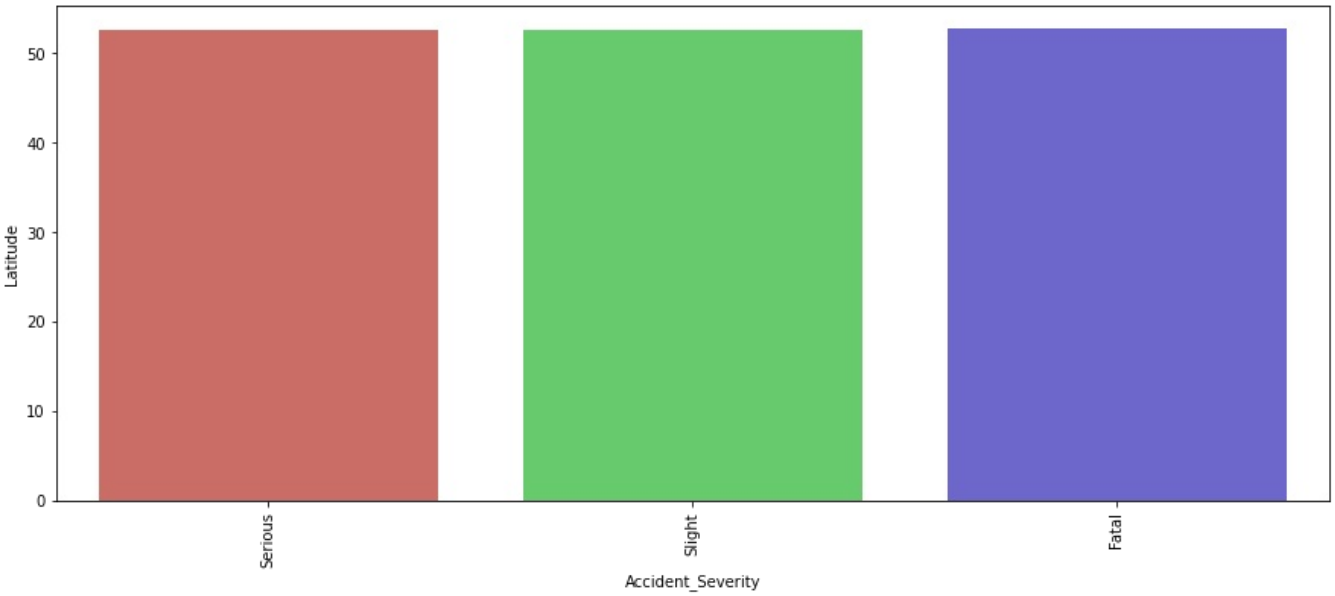

Number_of_Vehicles
7
6
5
4
3
2
1
0
10
20
30
40
50
60
70
Number_of_Casualties

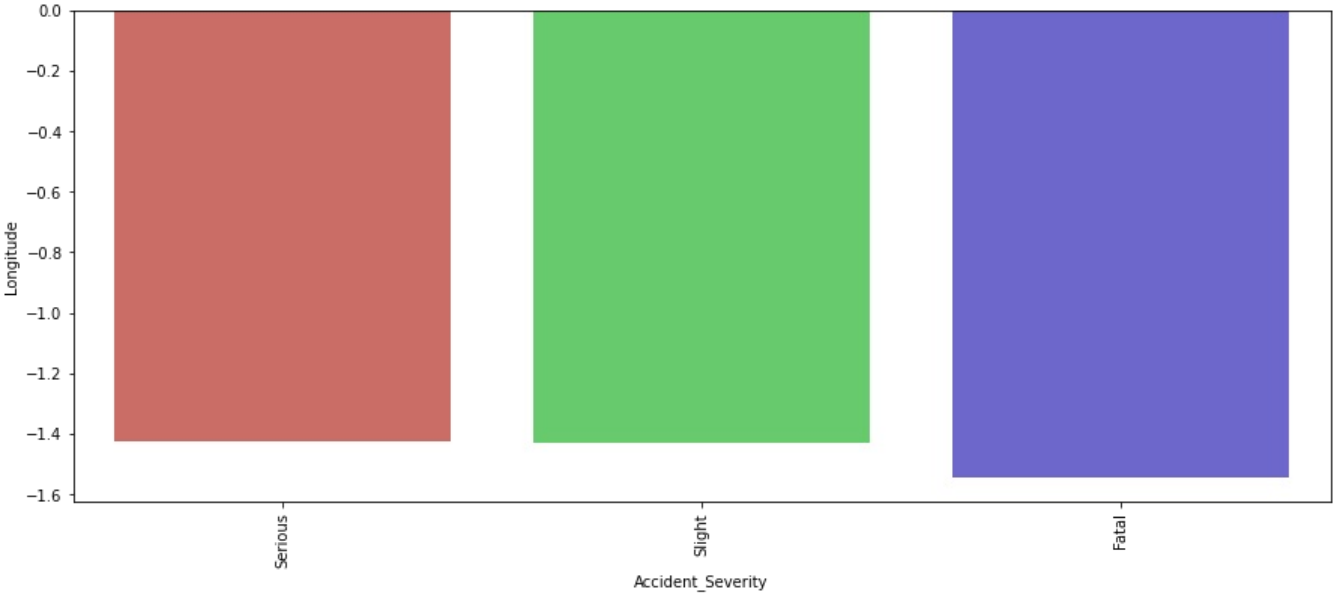


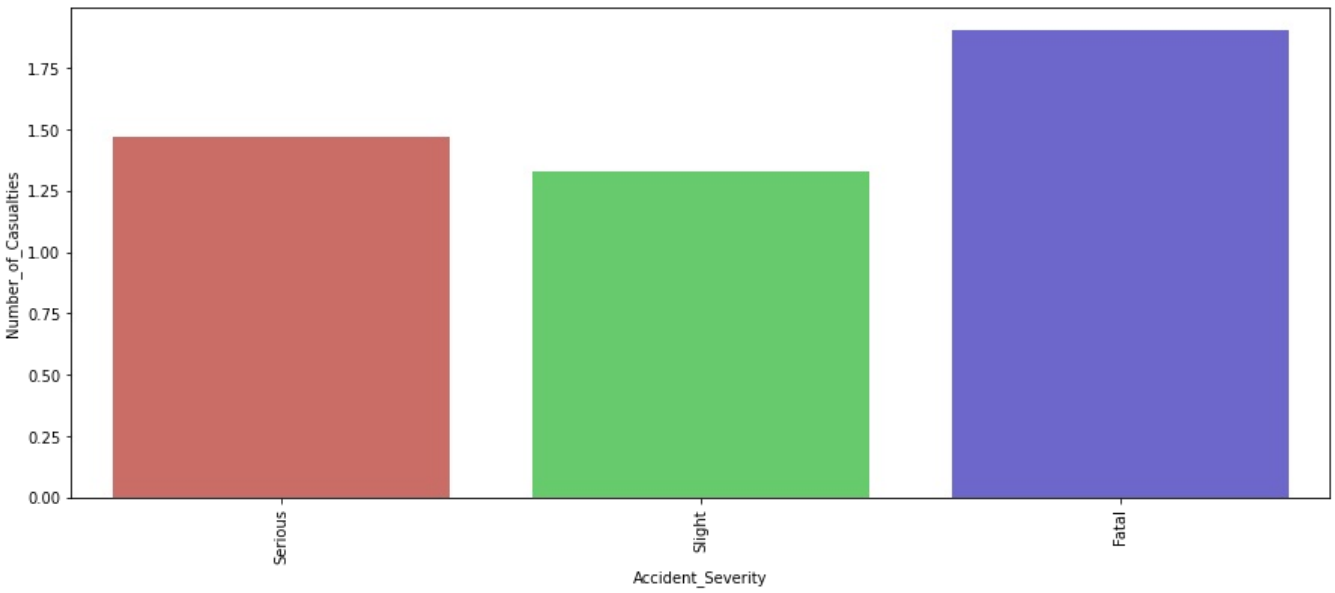


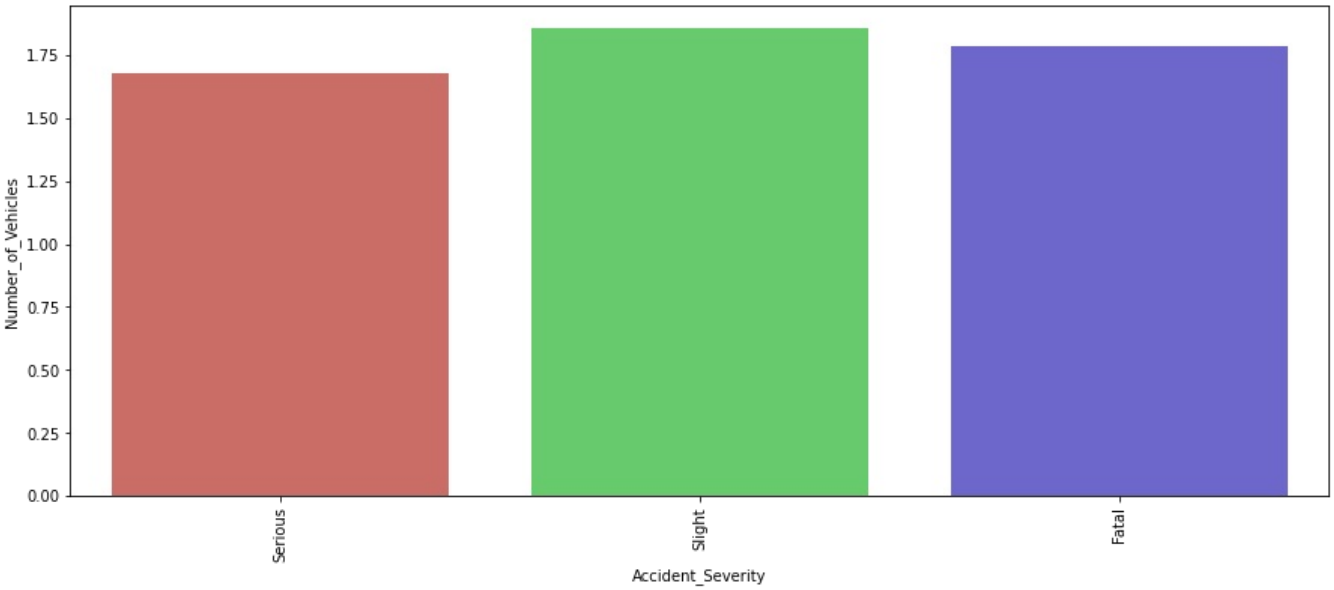


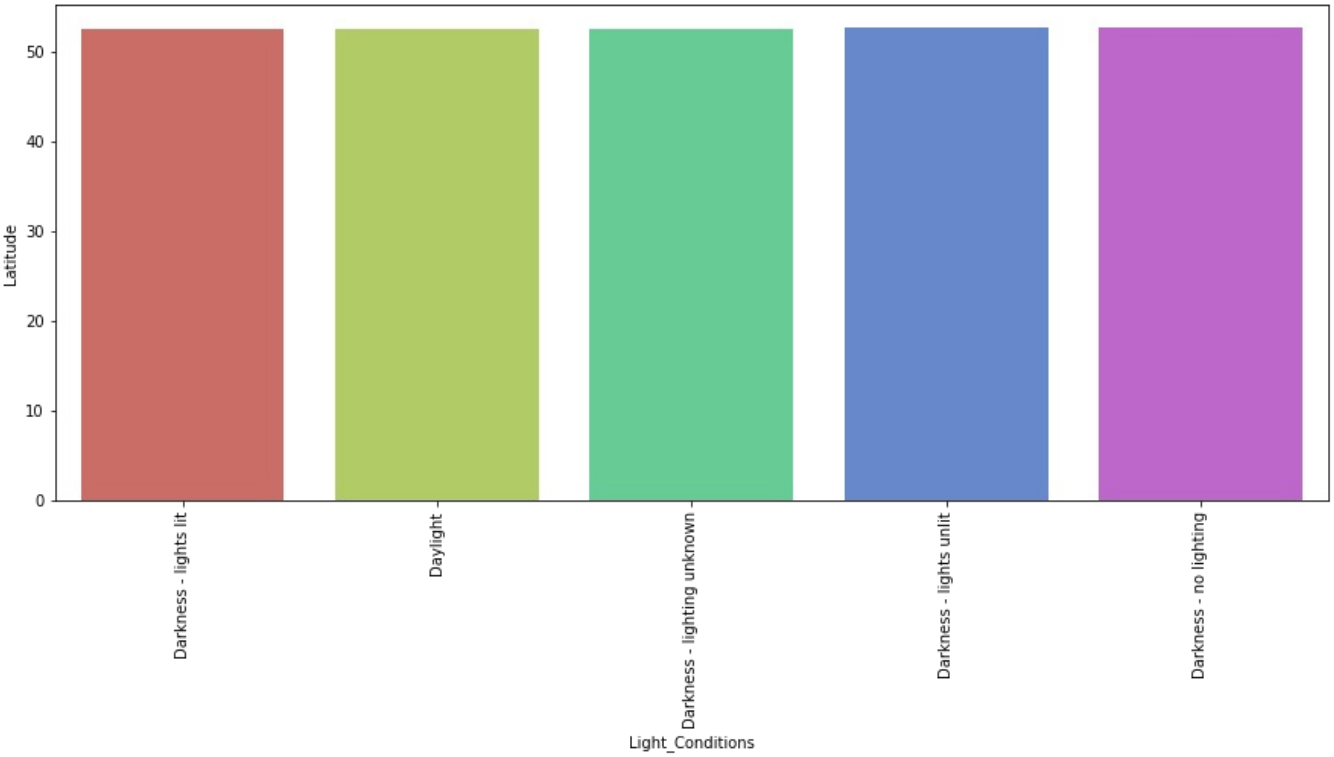
In [41]:
```
for i in categorical:
    for j in continuous:
        plt.figure(figsize=(15,6))
        sns.barplot(x = df[i], y = df[j], data = df, ci = None, palette = 'hls')
        plt.xticks(rotation = 90)
        plt.show()
```

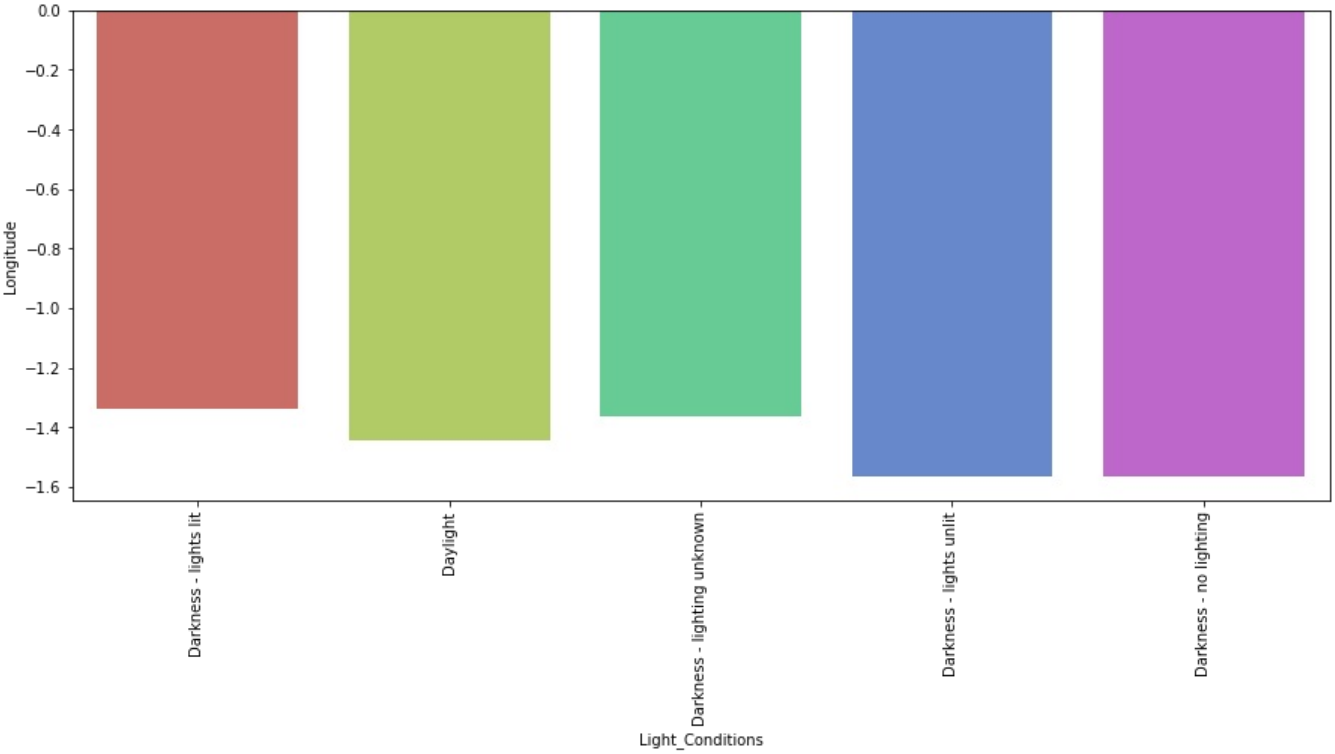

Latitude
0
10
20
30
40
50
Serious
Slight
Fatal
Accident_Severity

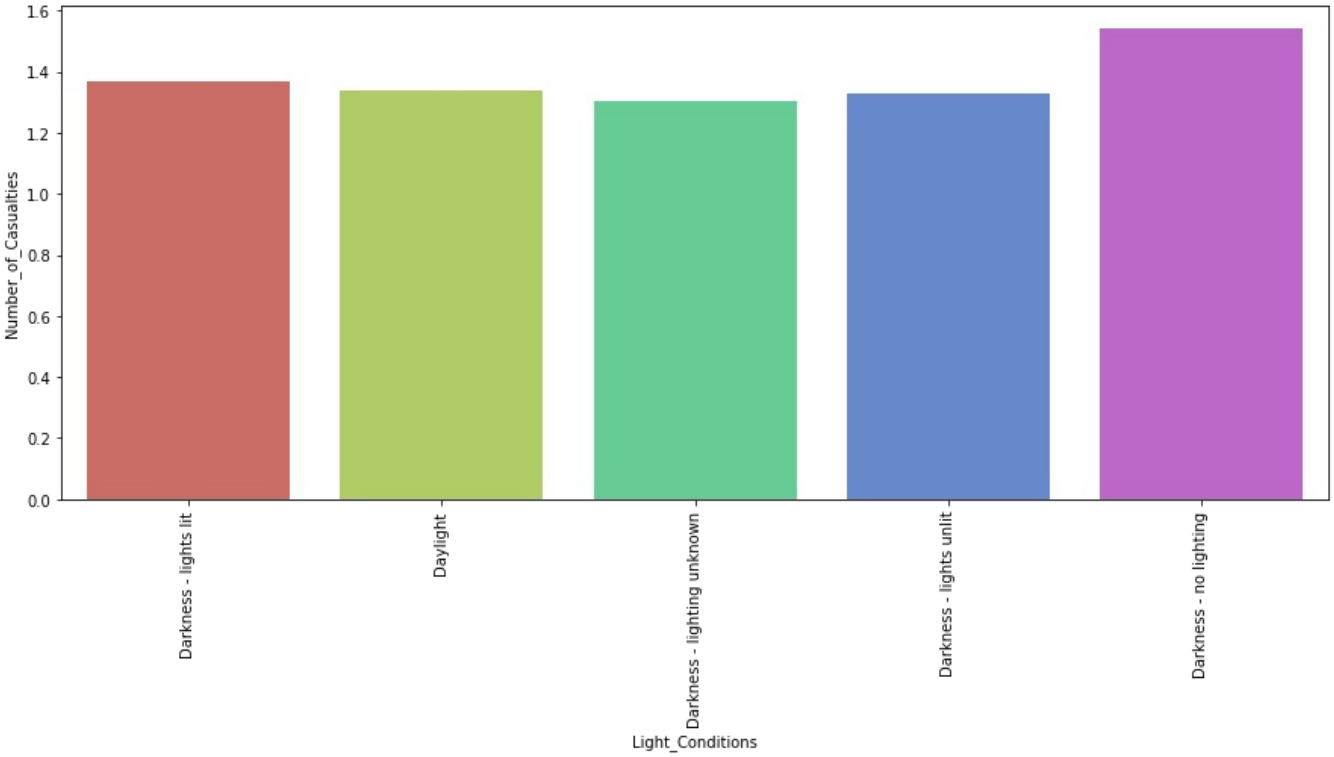

Longitude
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
−1.2
−1.4
−1.6
Serious
Slight
Fatal
Accident_Severity

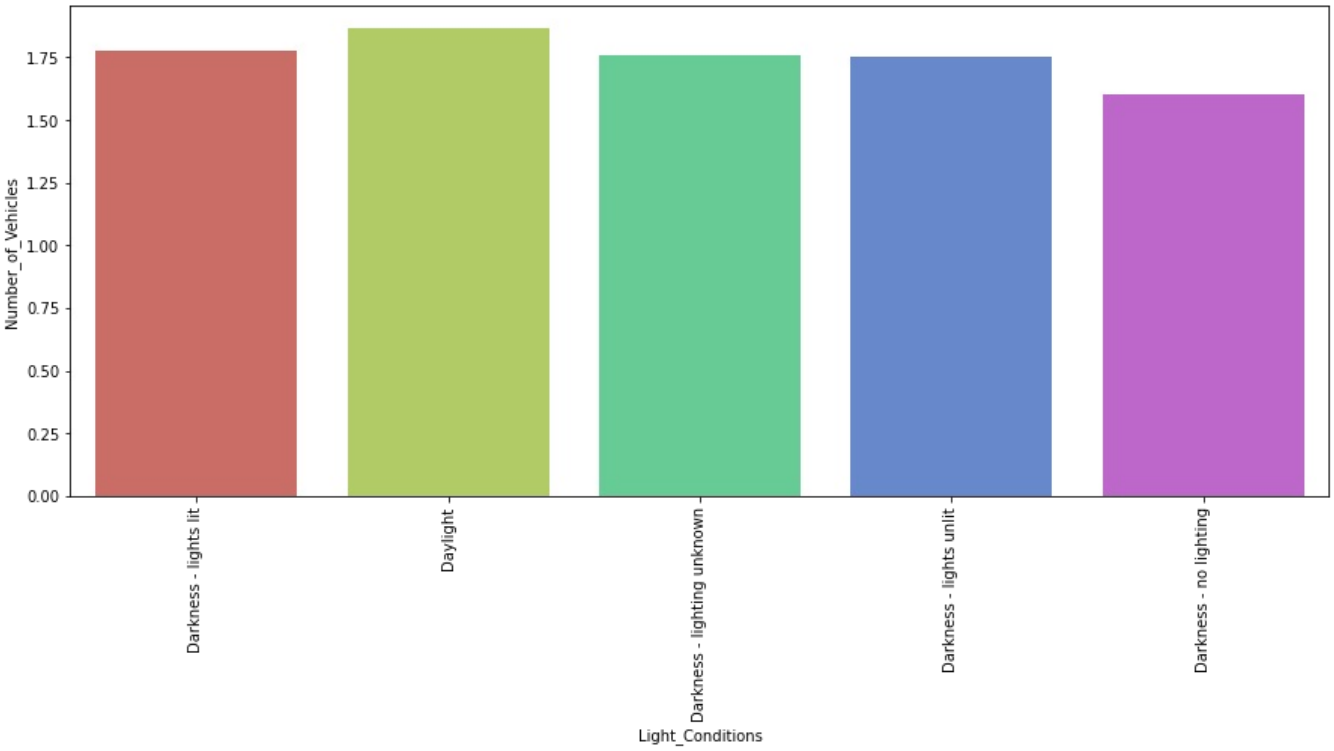

Number_of_Casualties
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
Serious
Slight
Fatal
Accident_Severity

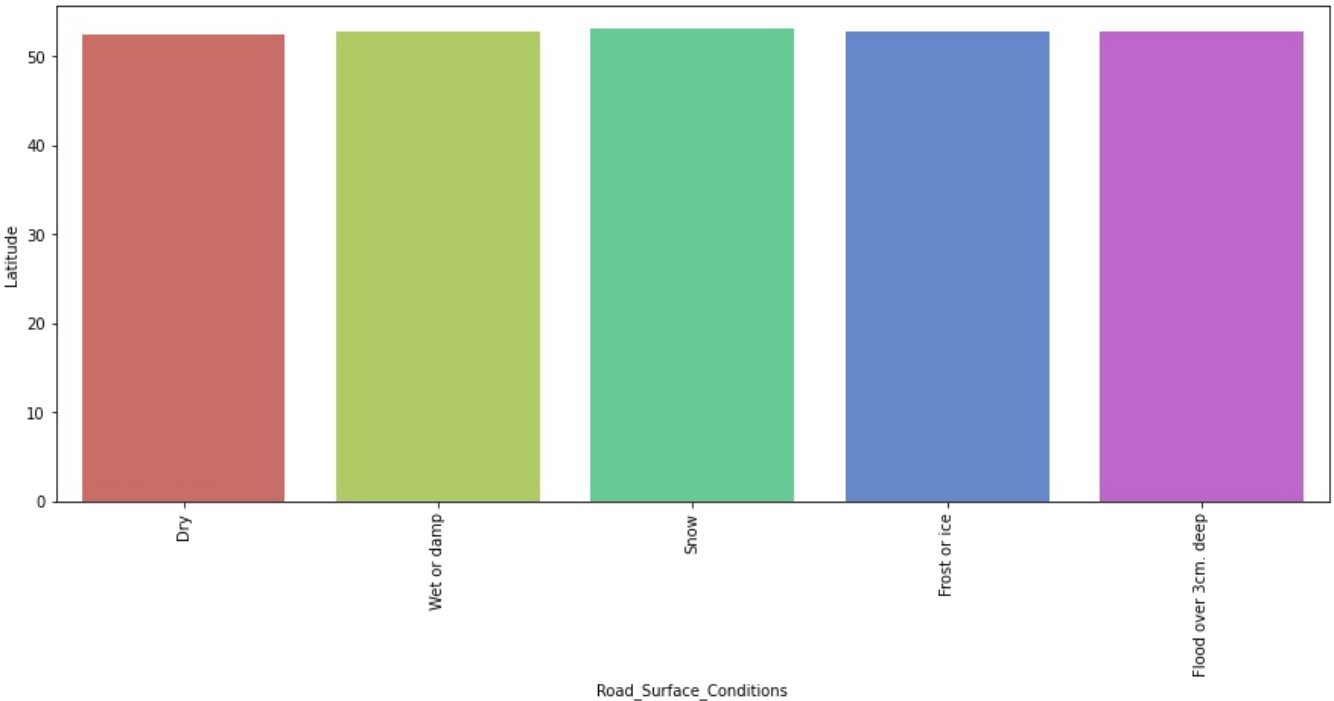

Number_of_Vehicles
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Serious
Slight
Fatal
Accident_Severity

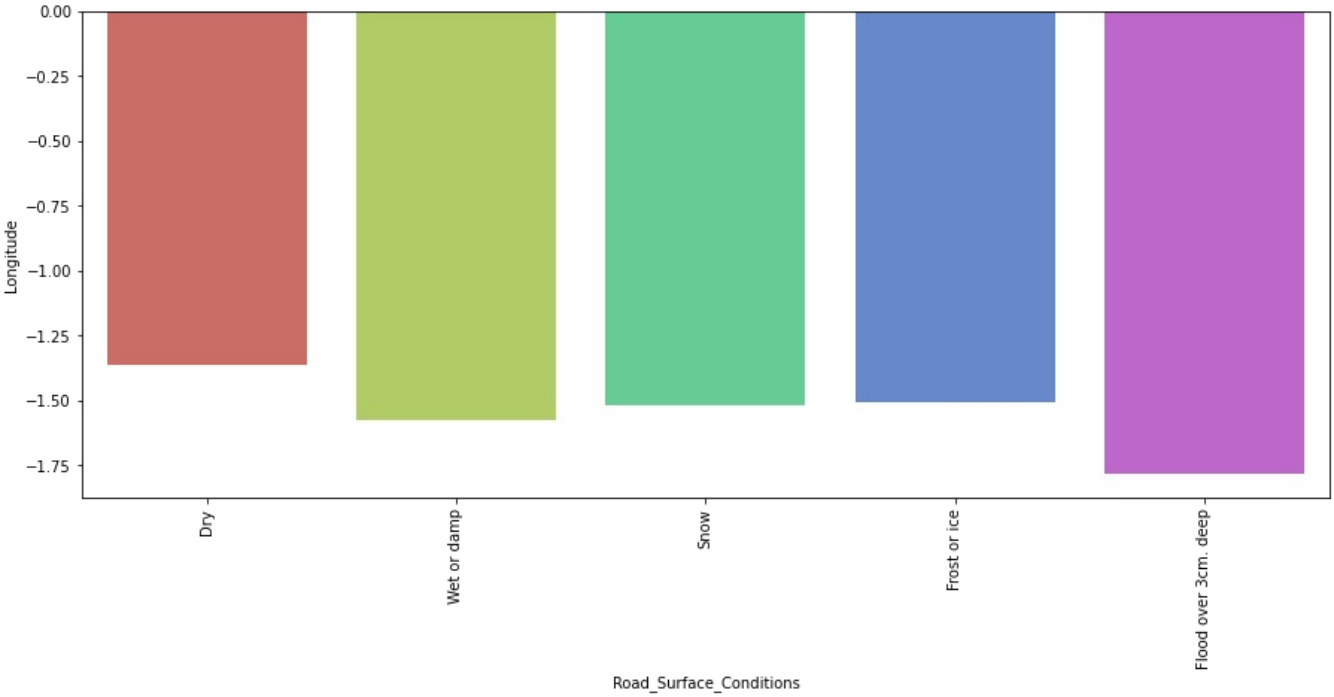

Latitude
50
40
30
20
10
0
Darkness - lights lit
Daylight
Darkness - lighting unknown
Darkness - lights unlit
Darkness - no lighting
Light_Conditions

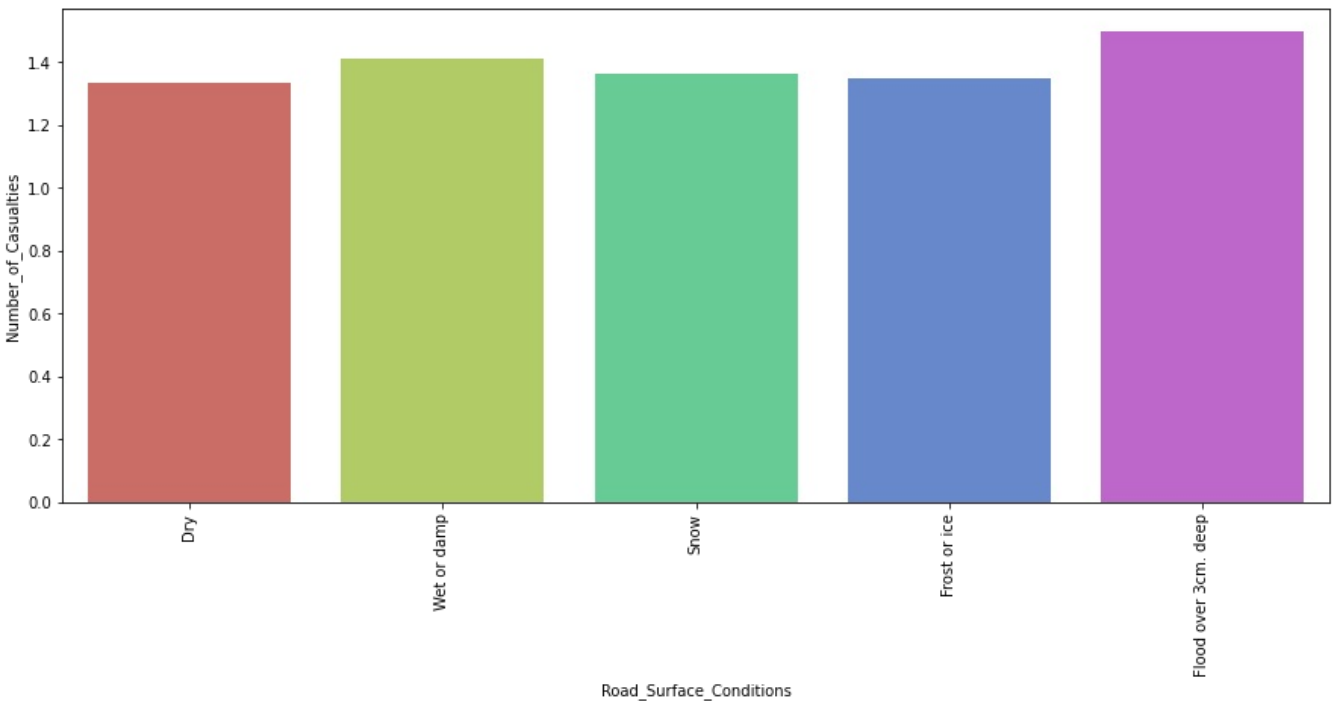

Longitude
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
−1.2
−1.4
−1.6
Darkness - lights lit
Daylight
Darkness - lighting unknown
Darkness - lights unlit
Darkness - no lighting
Light_Conditions

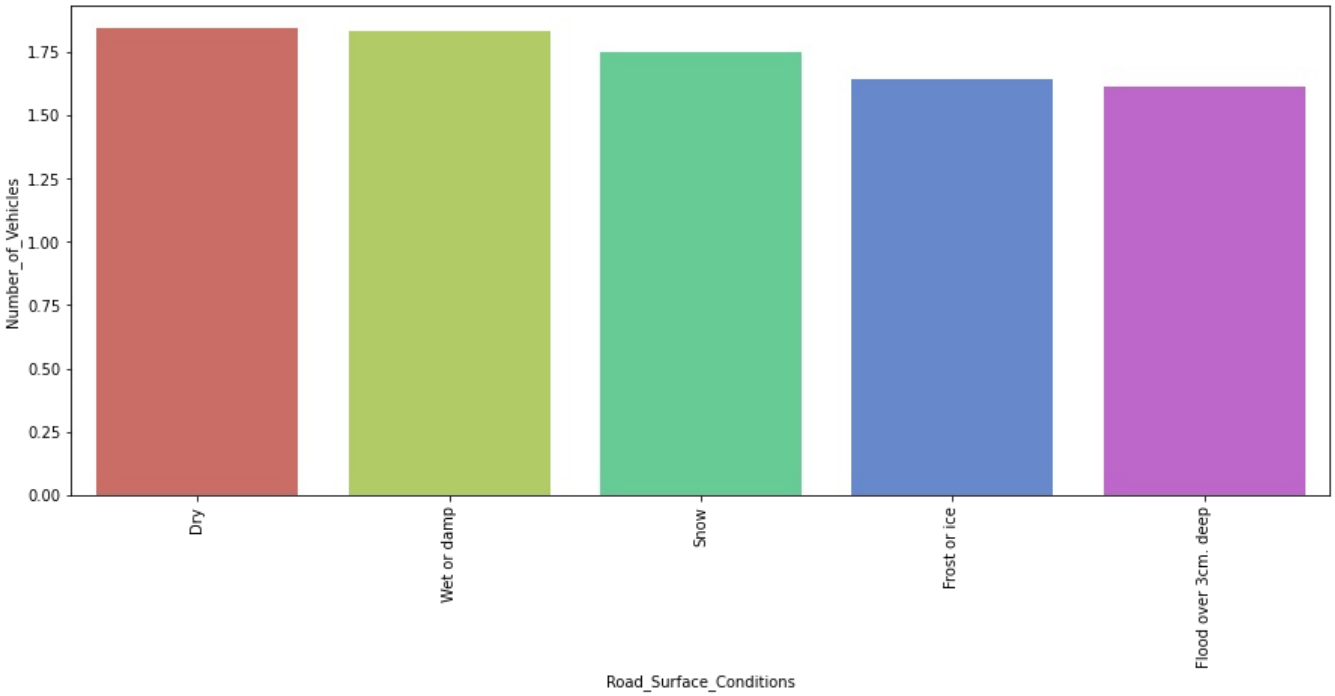

Number_of_Casualties
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Darkness - lights lit
Daylight
Darkness - lighting unknown
Darkness - lights unlit
Darkness - no lighting
Light_Conditions

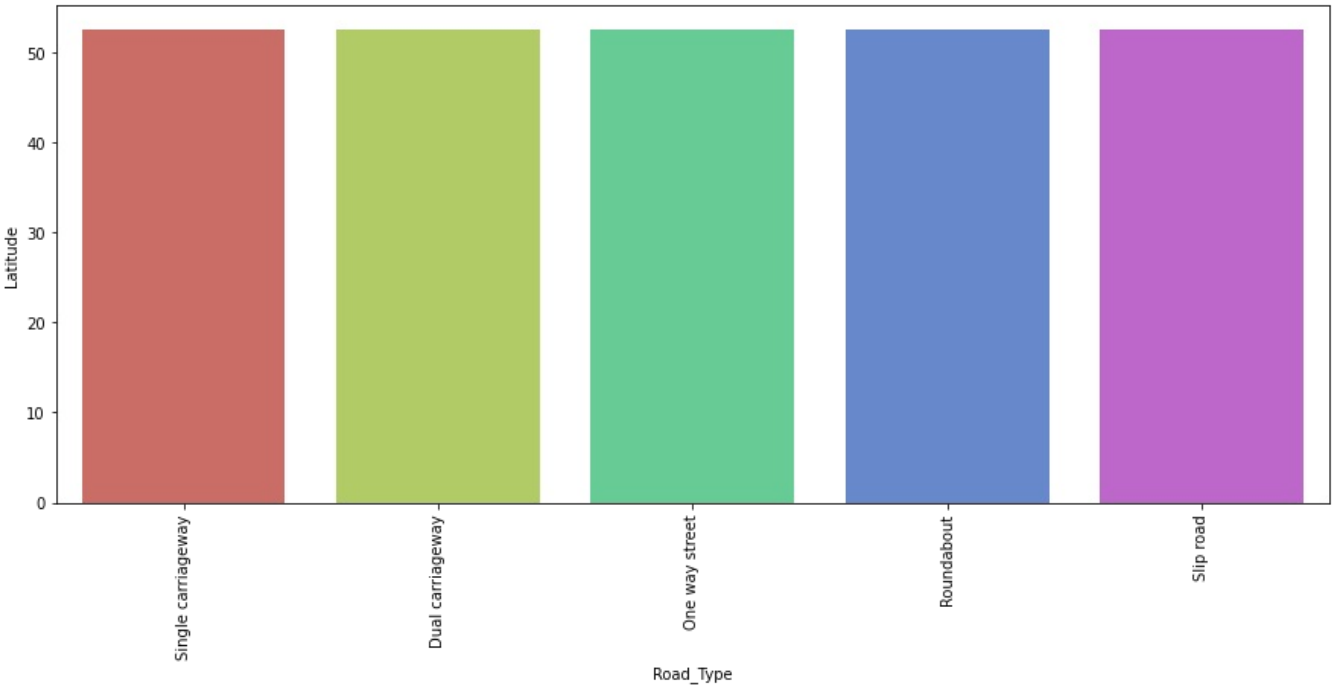

Number_of_Vehicles
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Darkness - lights lit
Daylight
Darkness - lighting unknown
Darkness - lights unlit
Darkness - no lighting
Light_Conditions

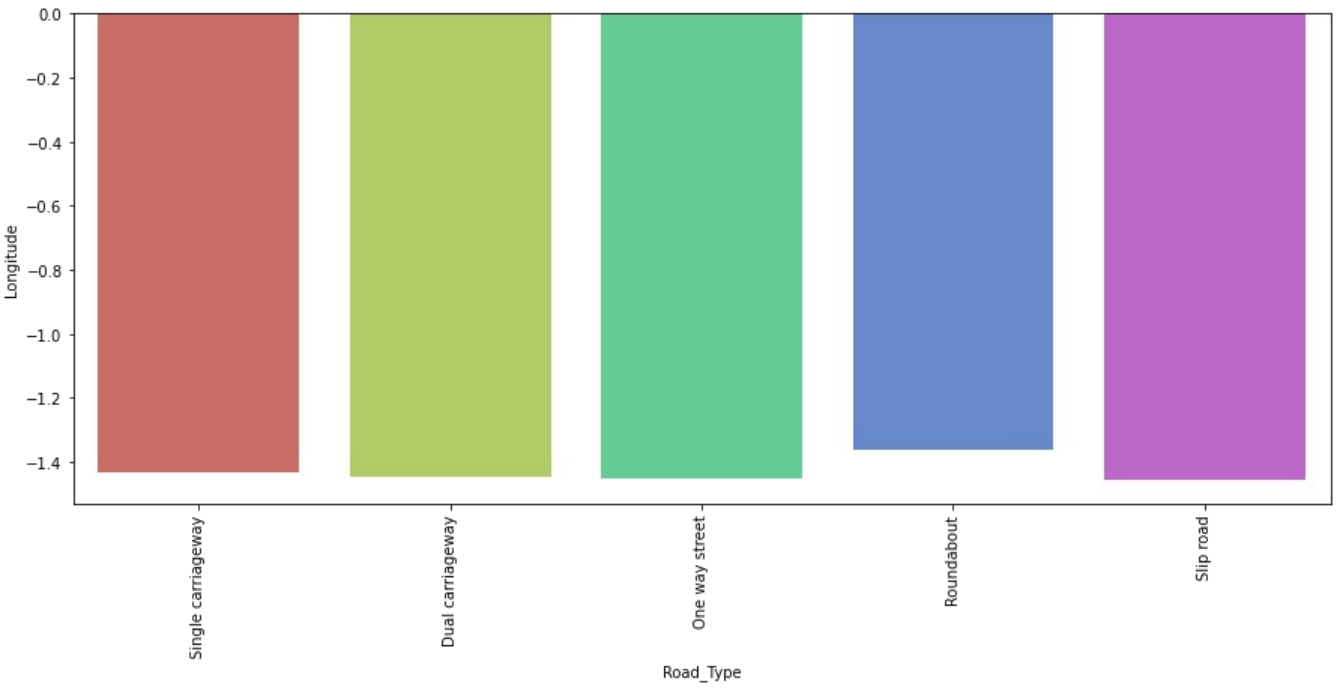

Latitude
0
10
20
30
40
50
Dry
Wet or damp
Snow
Frost or ice
Flood over 3cm. deep
Road_Surface_Conditions

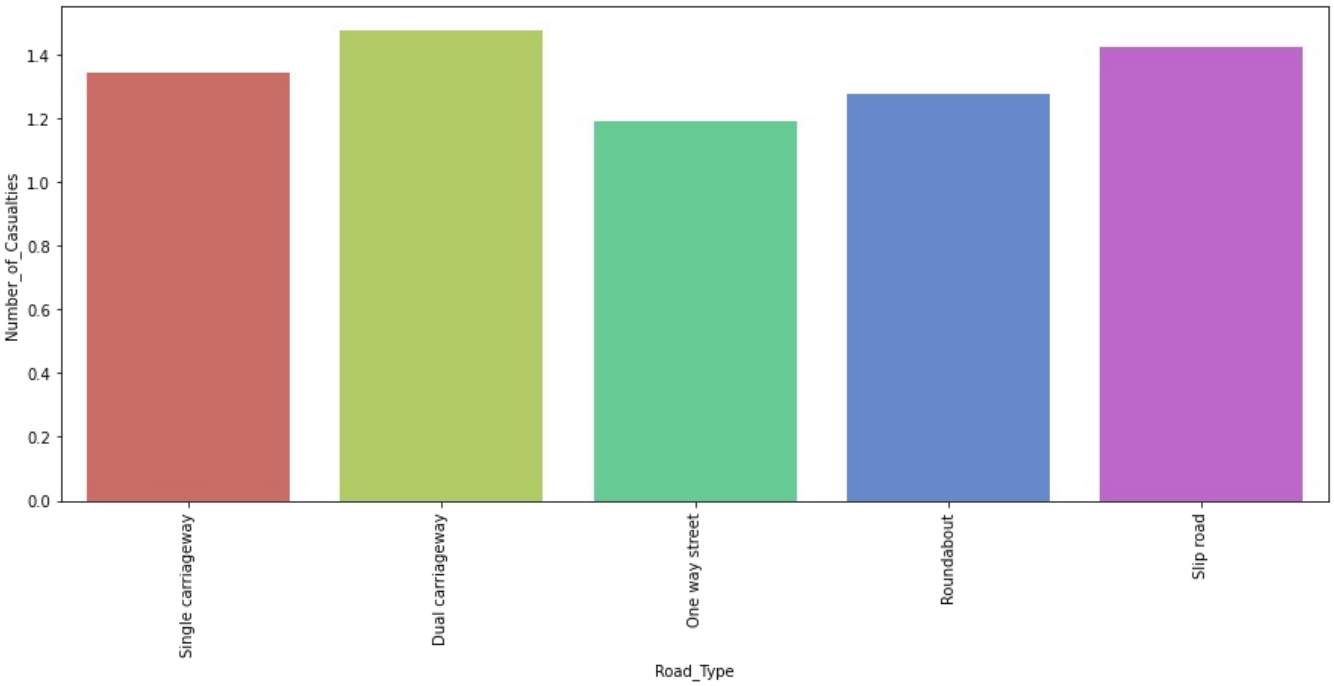

Longitude
0.00
−0.25
−0.50
−0.75
−1.00
−1.25
−1.50
−1.75
Dry
Wet or damp
Snow
Frost or ice
Flood over 3cm. deep
Road_Surface_Conditions

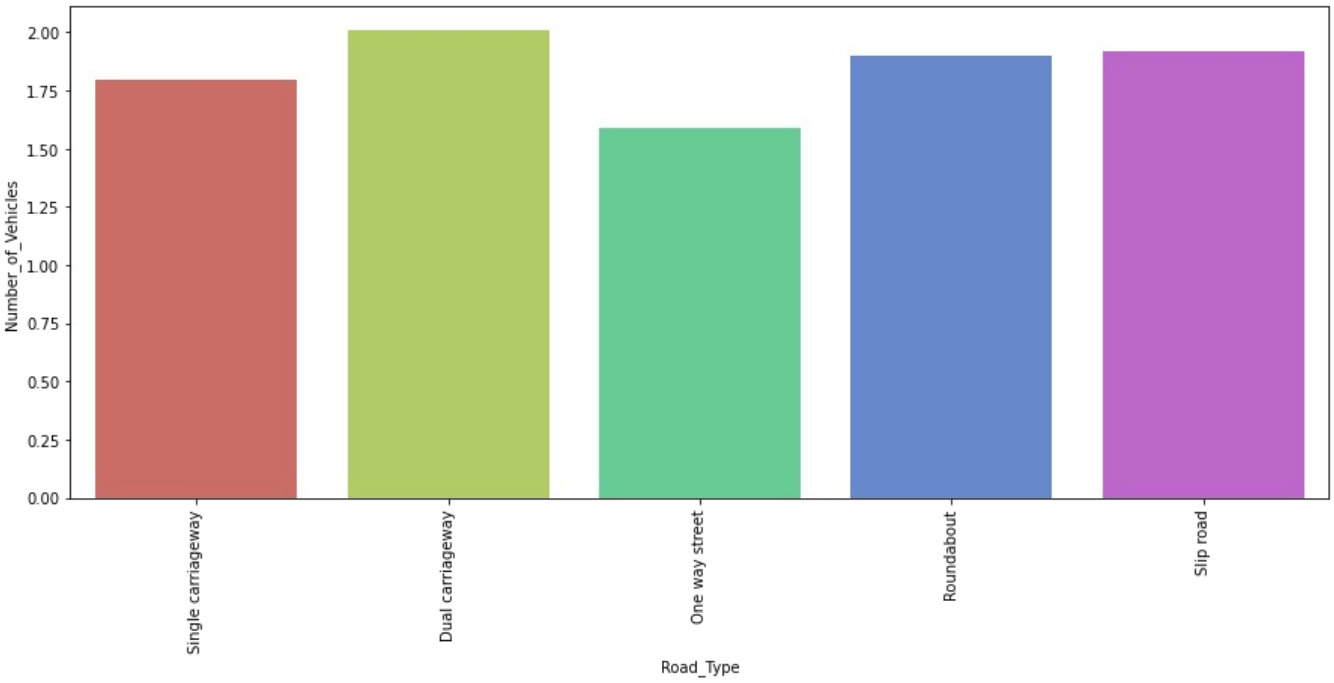

Number_of_Casualties
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
Dry
Wet or damp
Snow
Frost or ice
Flood over 3cm. deep
Road_Surface_Conditions

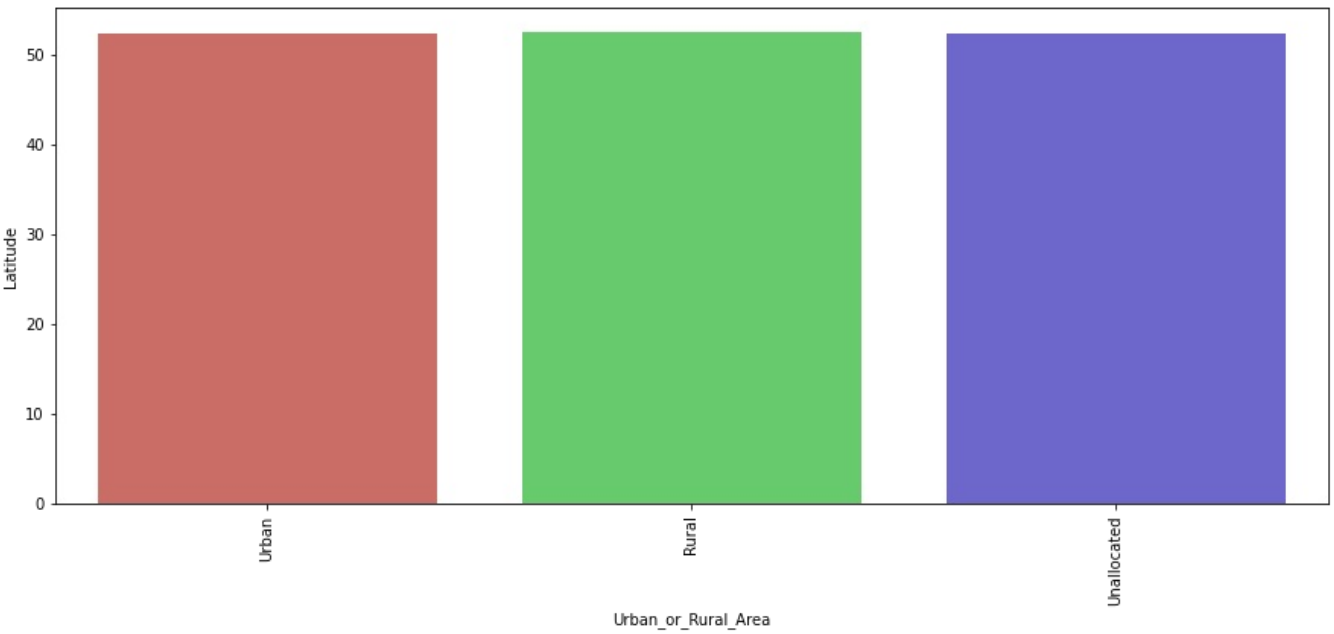

Number_of_Vehicles
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Dry
Wet or damp
Snow
Frost or ice
Flood over 3cm. deep
Road_Surface_Conditions

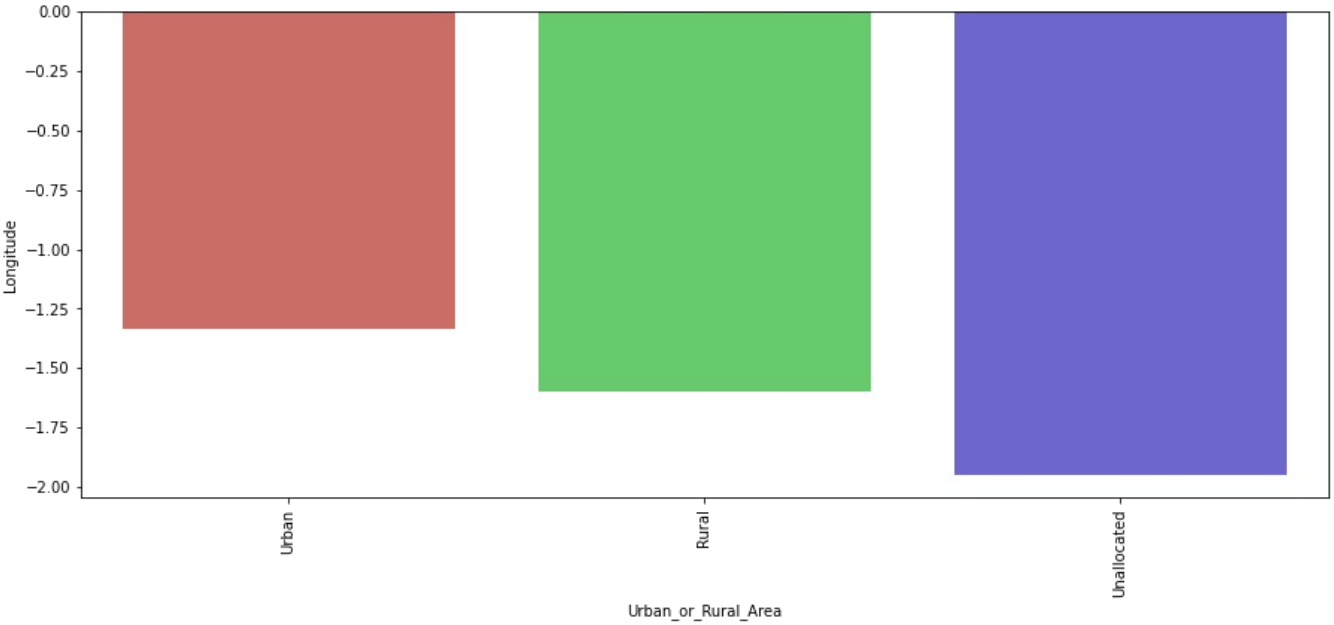

Latitude
50
40
30
20
10
0
Single carriageway
Dual carriageway
One way street
Roundabout
Slip road
Road_Type

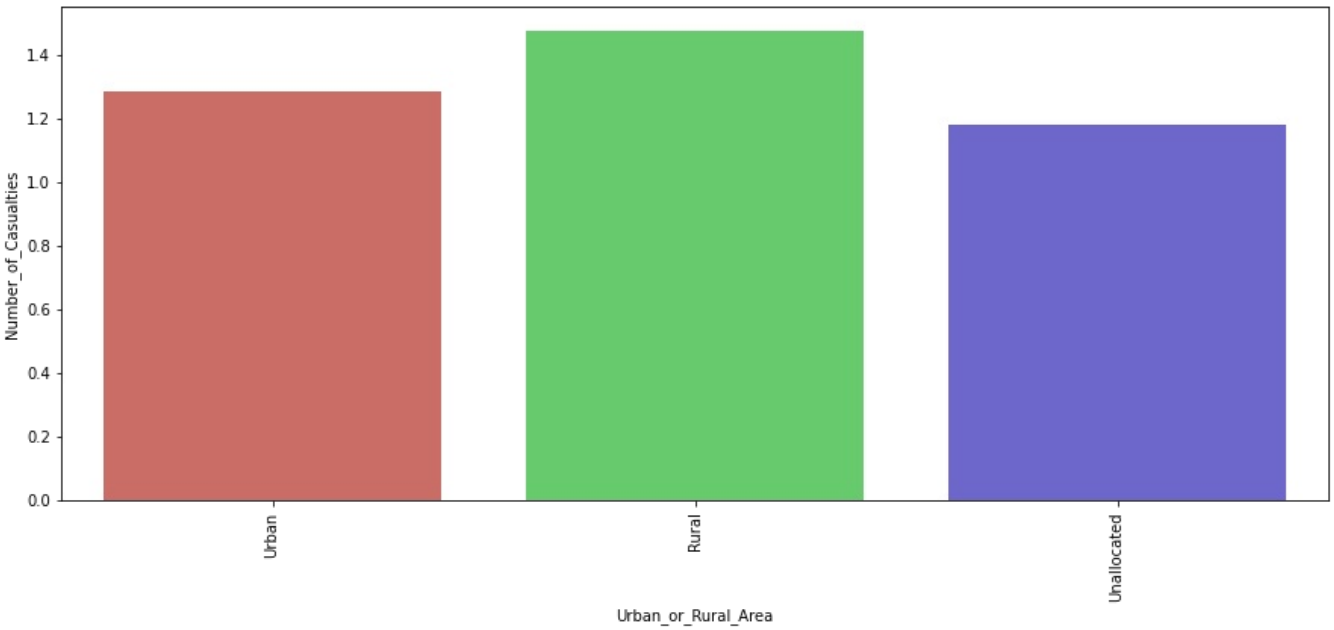

Longitude
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
−1.2
−1.4
Single carriageway
Dual carriageway
One way street
Roundabout
Slip road
Road_Type

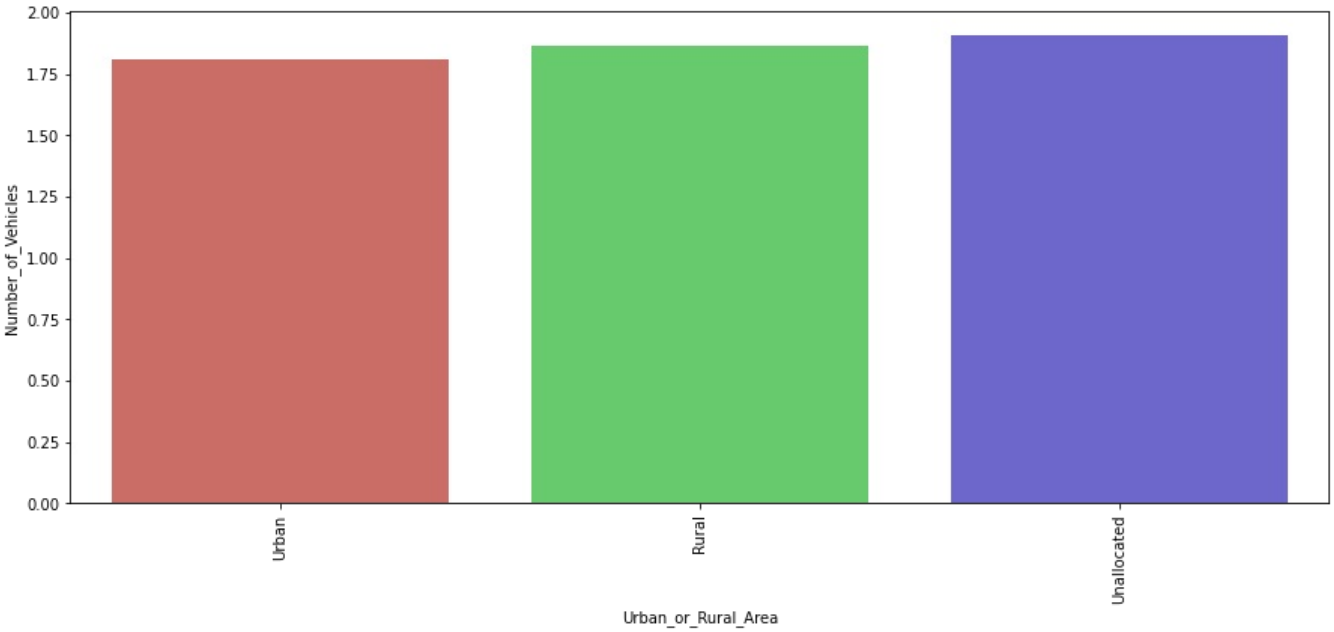

Number_of_Casualties
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
Single carriageway
Dual carriageway
One way street
Roundabout
Slip road
Road_Type

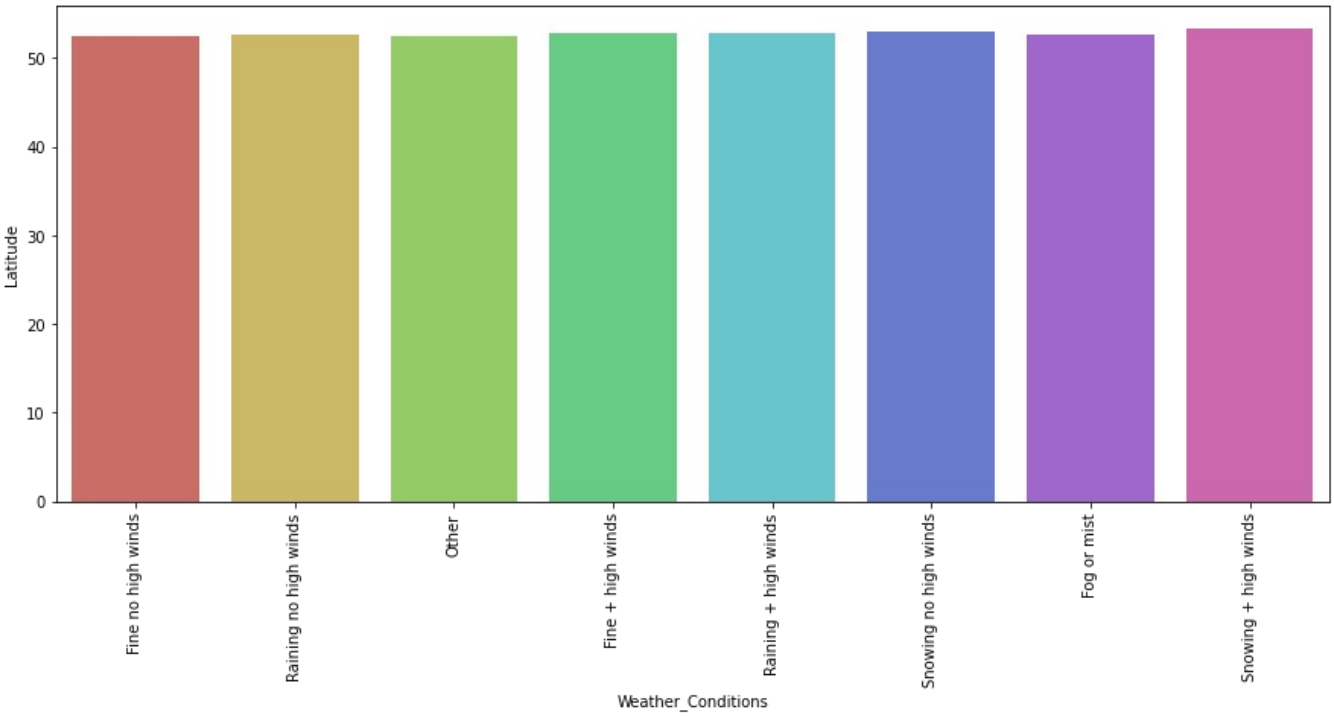

Number_of_Vehicles
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Single carriageway
Dual carriageway
One way street
Roundabout
Slip road
Road_Type

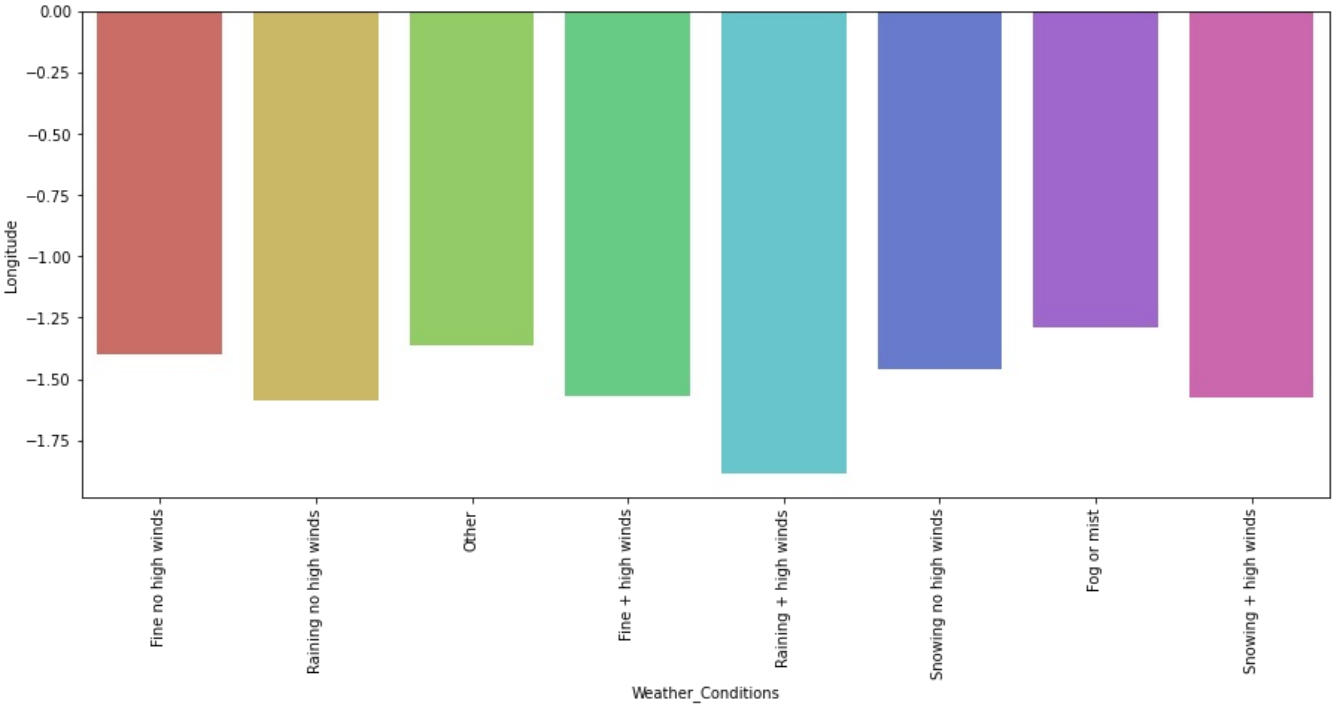

Latitude
0
10
20
30
40
50
Urban
Rural
Unallocated
Urban_or_Rural_Area

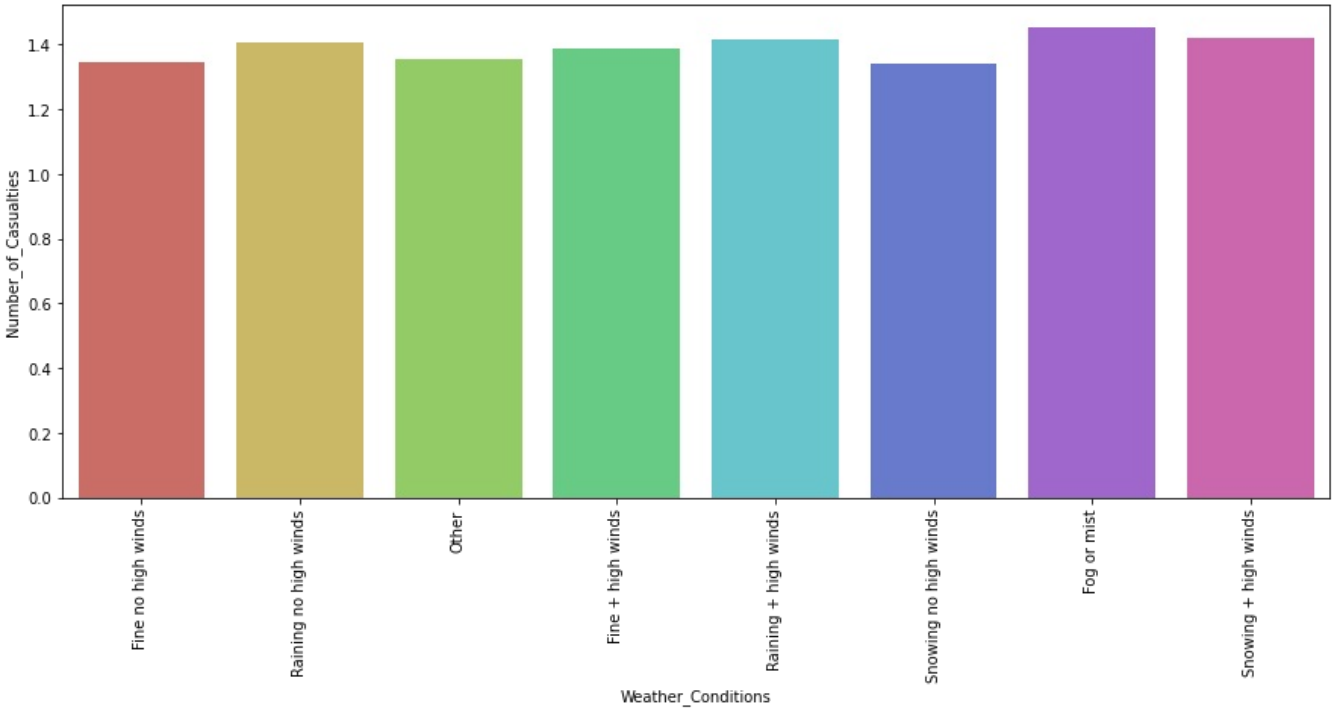

0.00
−0.25
−0.50
−0.75
−1.00
−1.25
−1.50
−1.75
−2.00
Longitude
Urban
Rural
Unallocated
Urban_or_Rural_Area

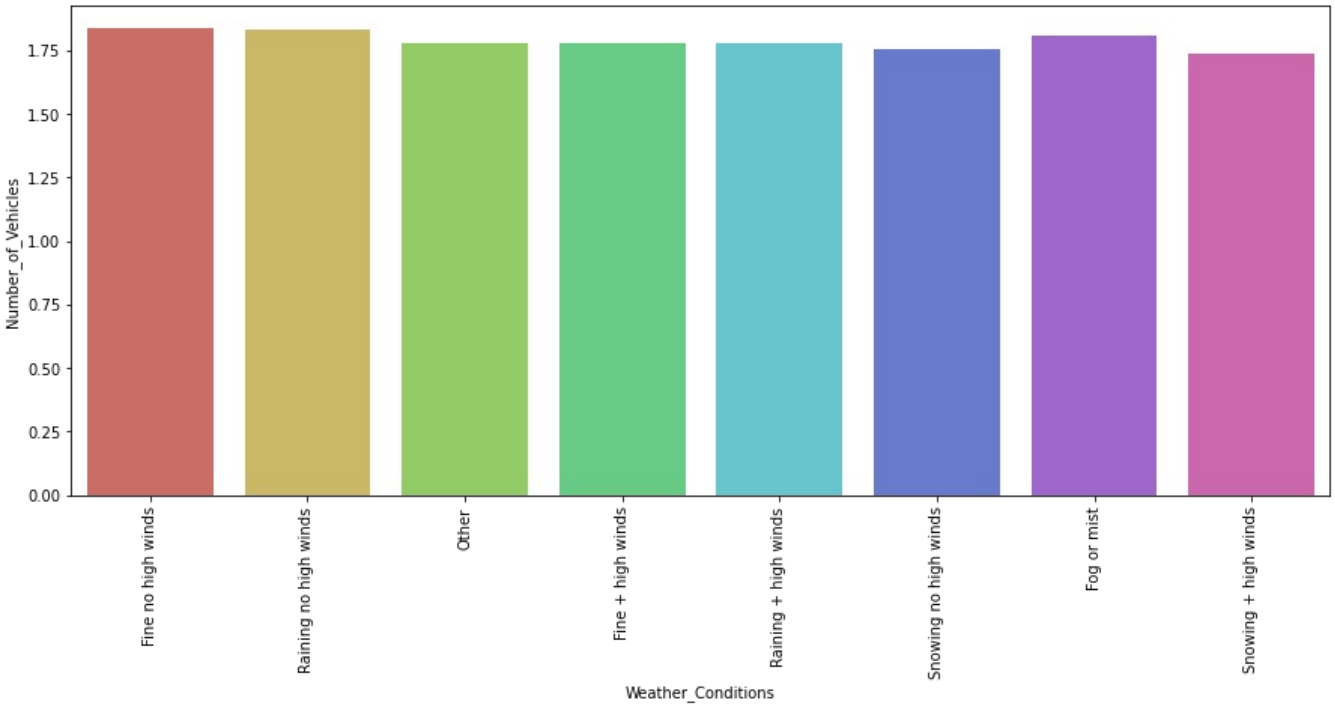

1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Number_of_Casualties
Urban
Rural
Unallocated
Urban_or_Rural_Area


2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Number_of_Vehicles
Urban
Rural
Unallocated
Urban_or_Rural_Area

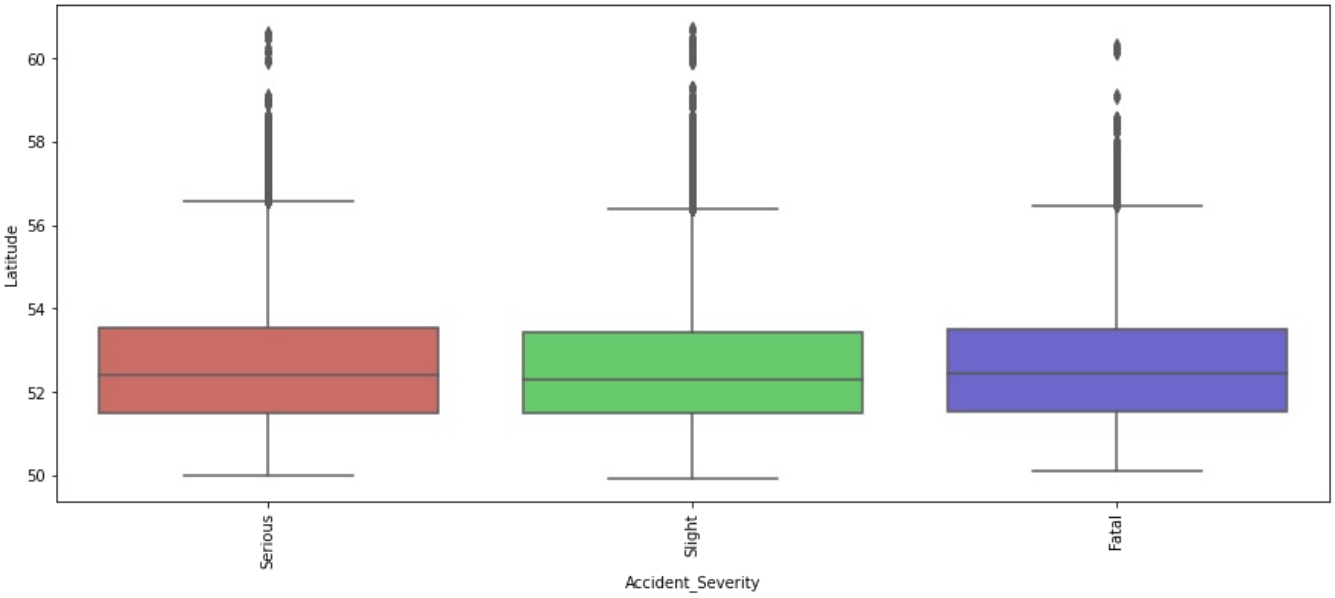

Latitude
0
10
20
30
40
50
Fine no high winds
Raining no high winds
Other
Fine + high winds
Raining + high winds
Snowing no high winds
Fog or mist
Snowing + high winds
Weather_Conditions

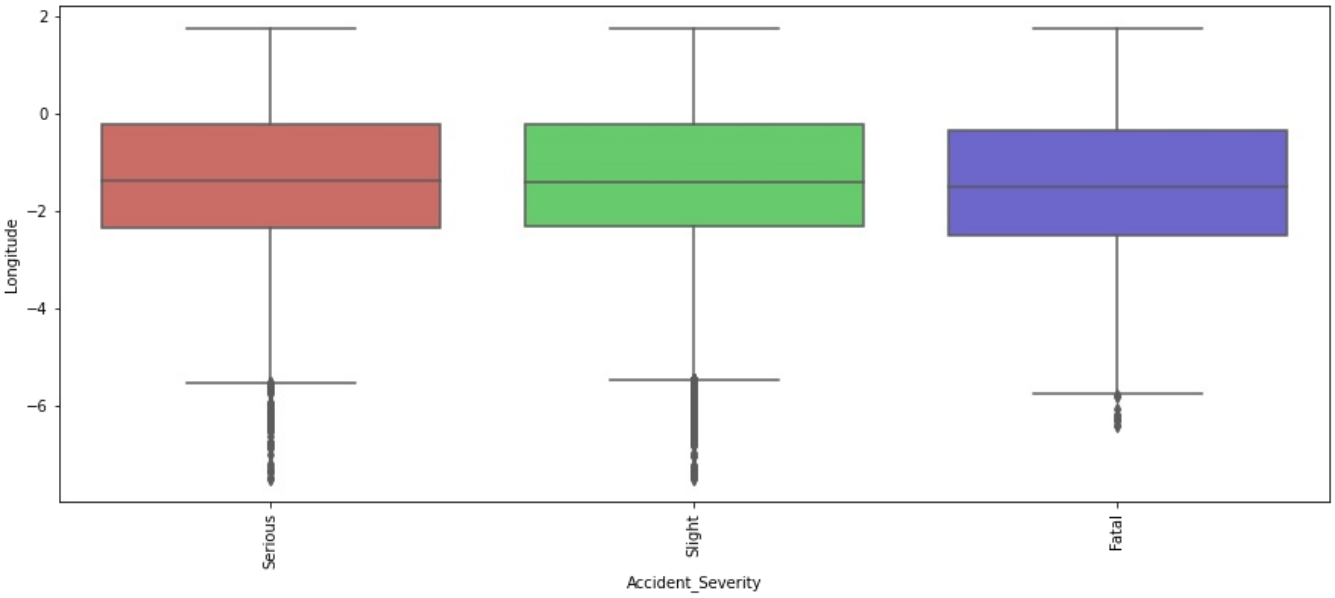

Longitude
0.00
−0.25
−0.50
−0.75
−1.00
−1.25
−1.50
−1.75
Fine no high winds
Raining no high winds
Other
Fine + high winds
Raining + high winds
Snowing no high winds
Fog or mist
Snowing + high winds
Weather_Conditions

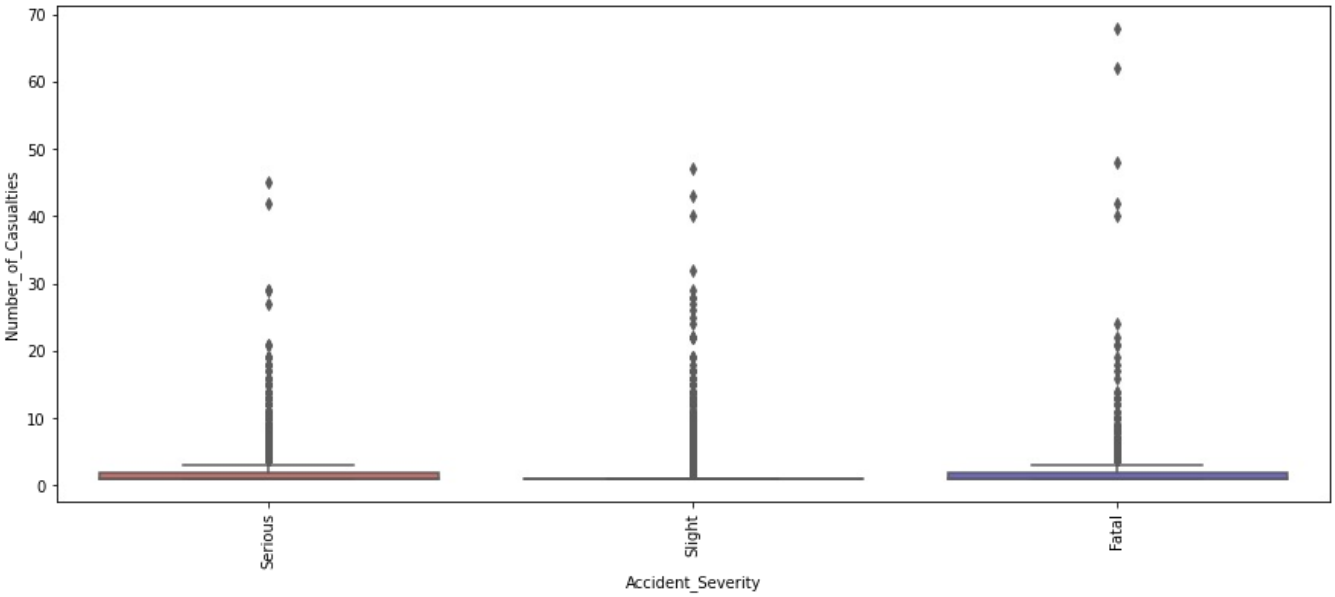

Number_of_Casualties
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
Fine no high winds
Raining no high winds
Other
Fine + high winds
Raining + high winds
Snowing no high winds
Fog or mist
Snowing + high winds
Weather_Conditions

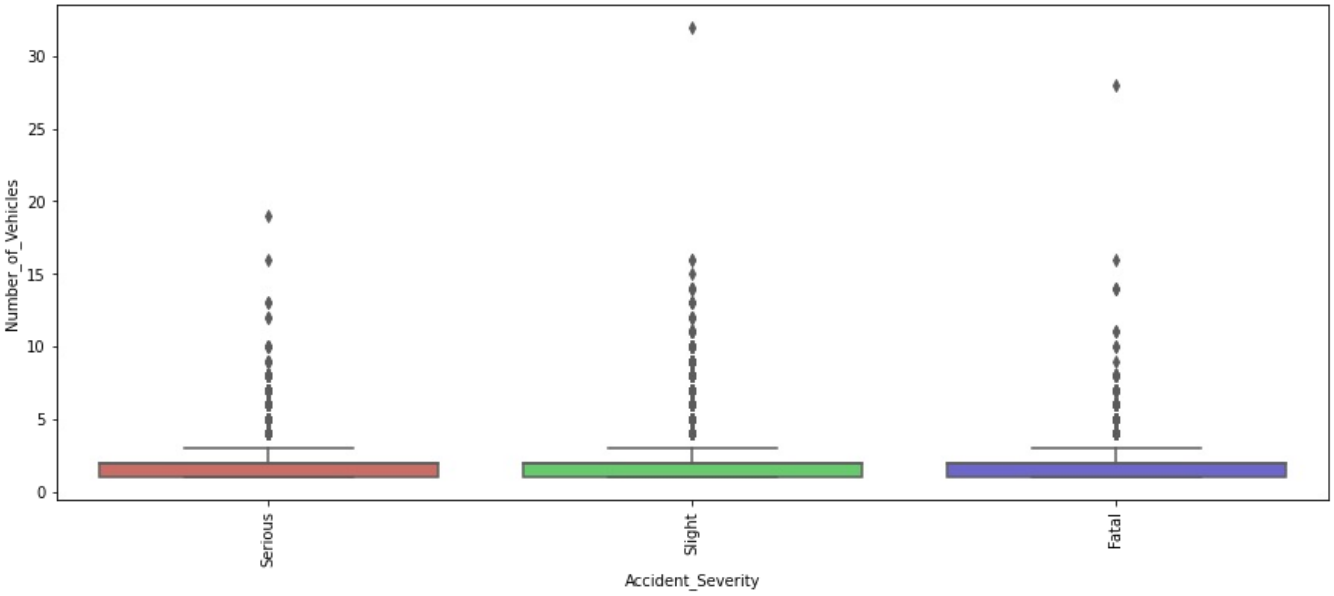


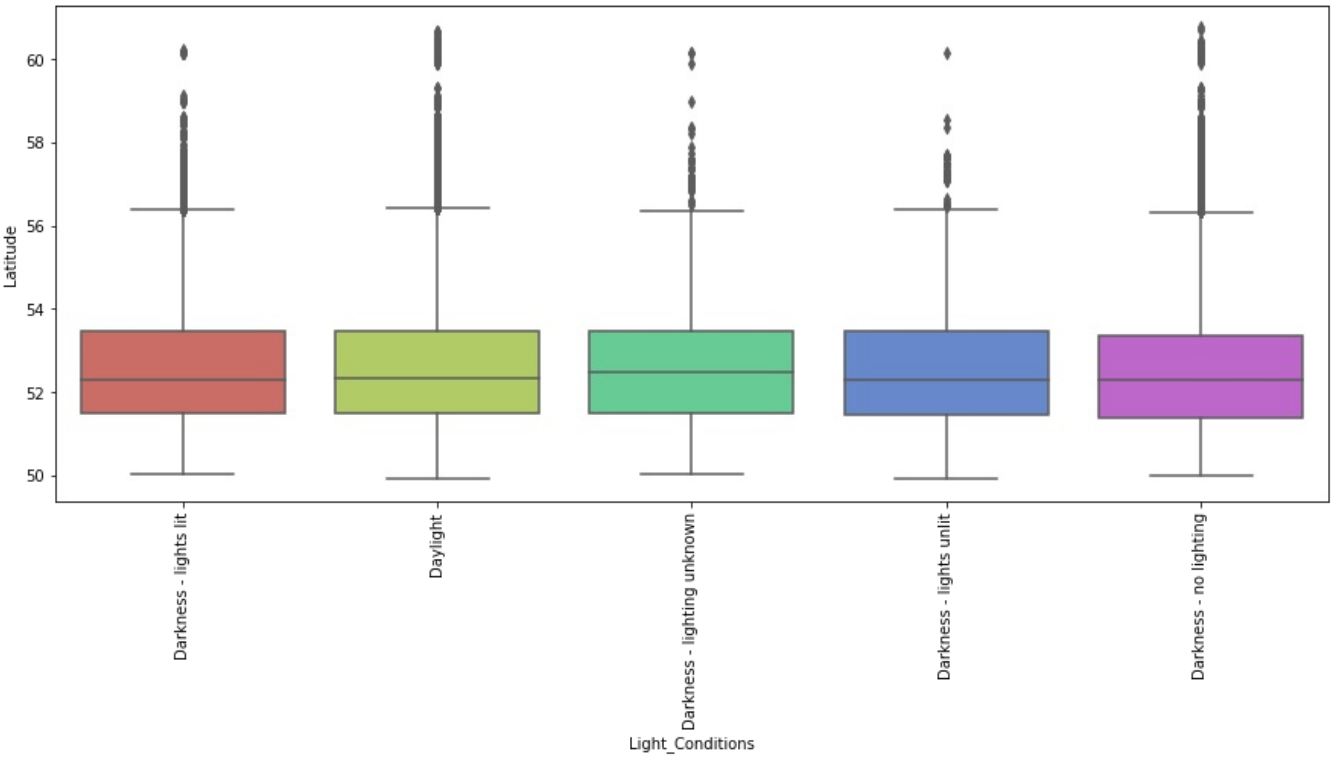
In [42]:
```
for i in categorical:
    for j in continuous:
        plt.figure(figsize=(15,6))
        sns.boxplot(x = df[i], y = df[j], data = df, palette = 'hls')
        plt.xticks(rotation = 90)
        plt.show()
```

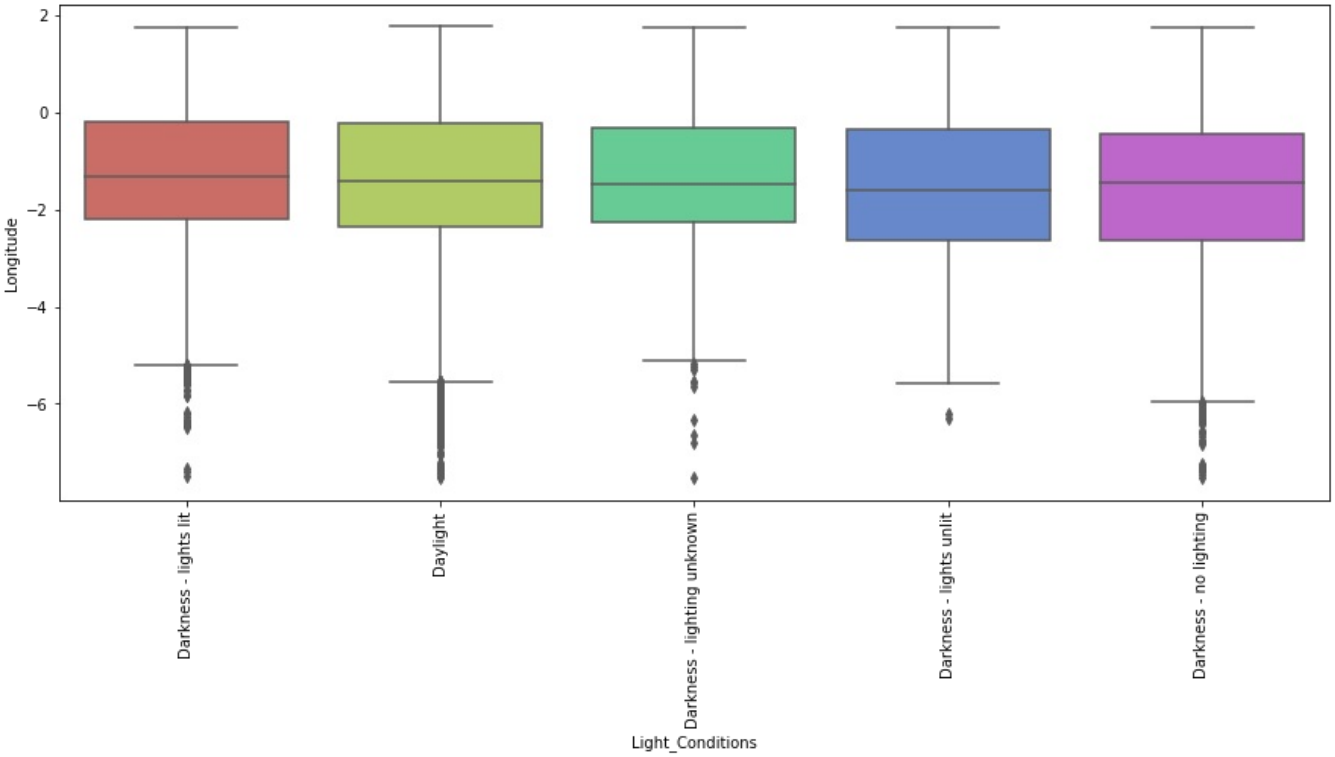


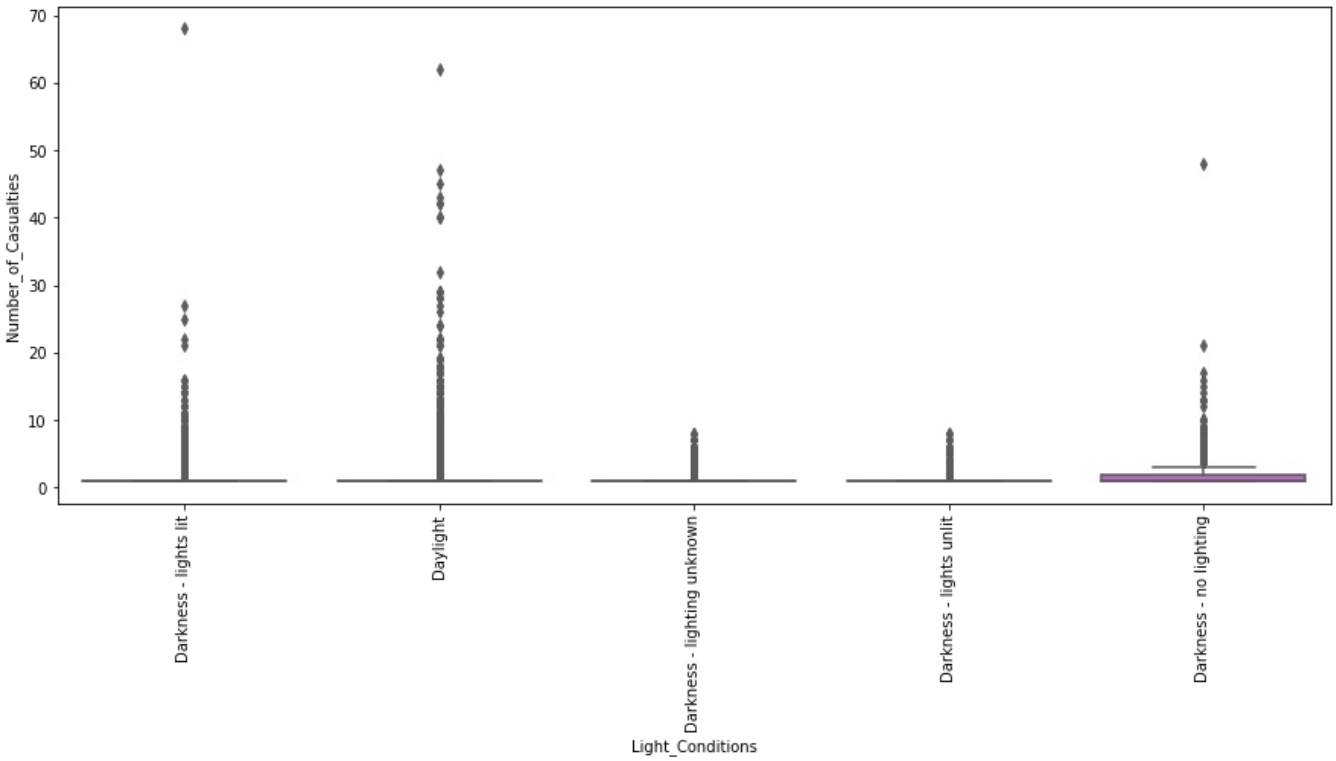


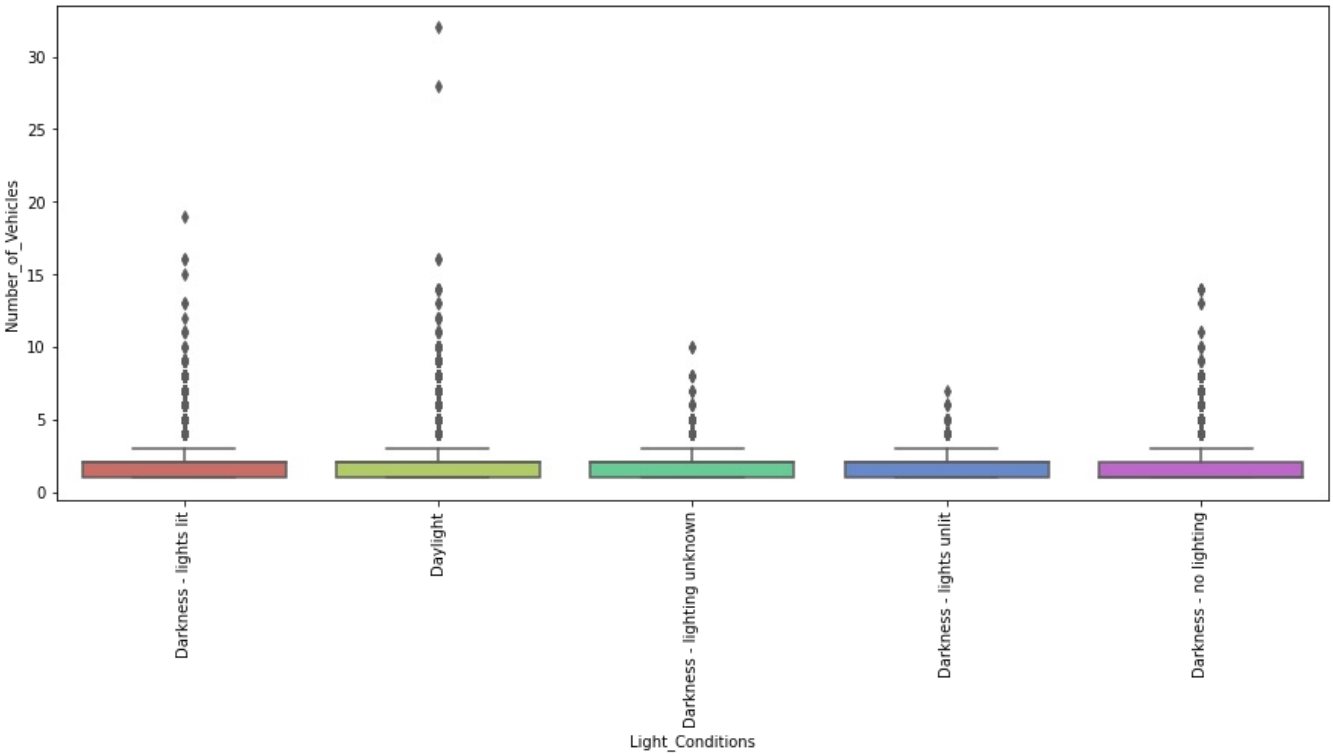

Number_of_Casualties
0
10
20
30
40
50
60
70
Serious
Slight
Fatal
Accident_Severity

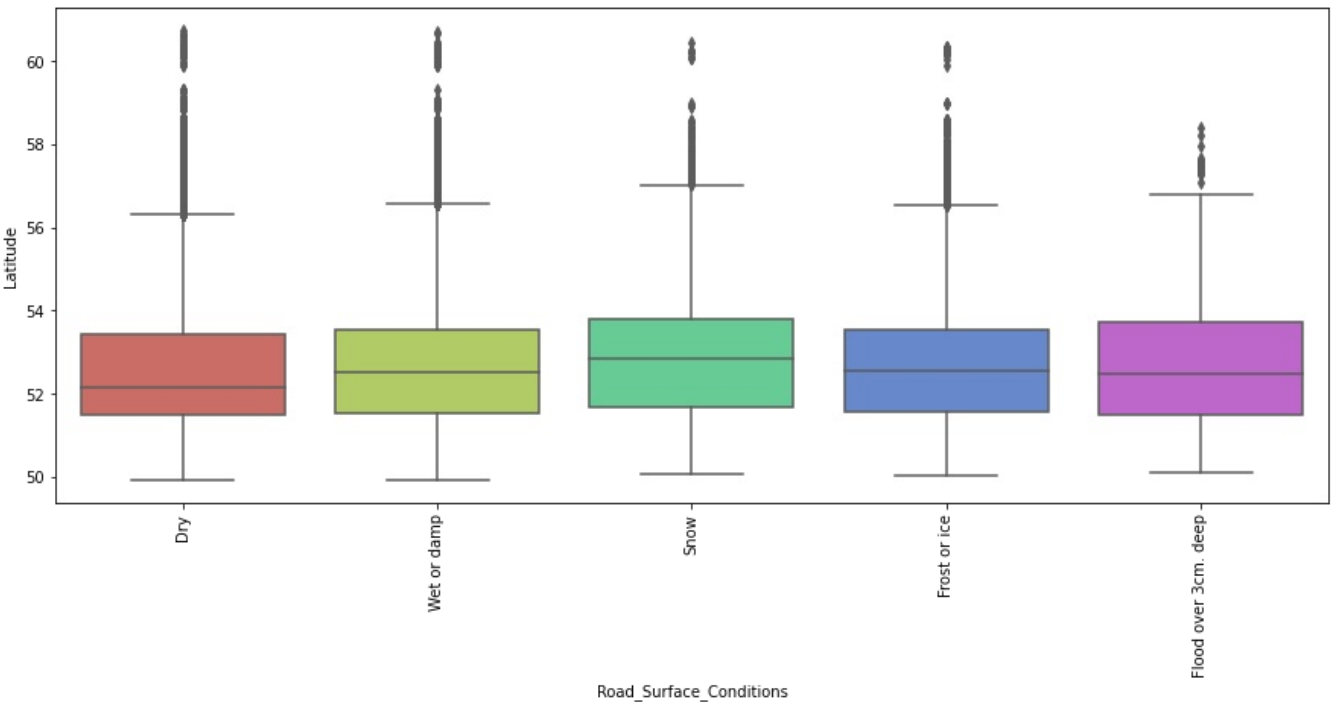

Number_of_Vehicles
0
5
10
15
20
25
30
Serious
Slight
Fatal
Accident_Severity

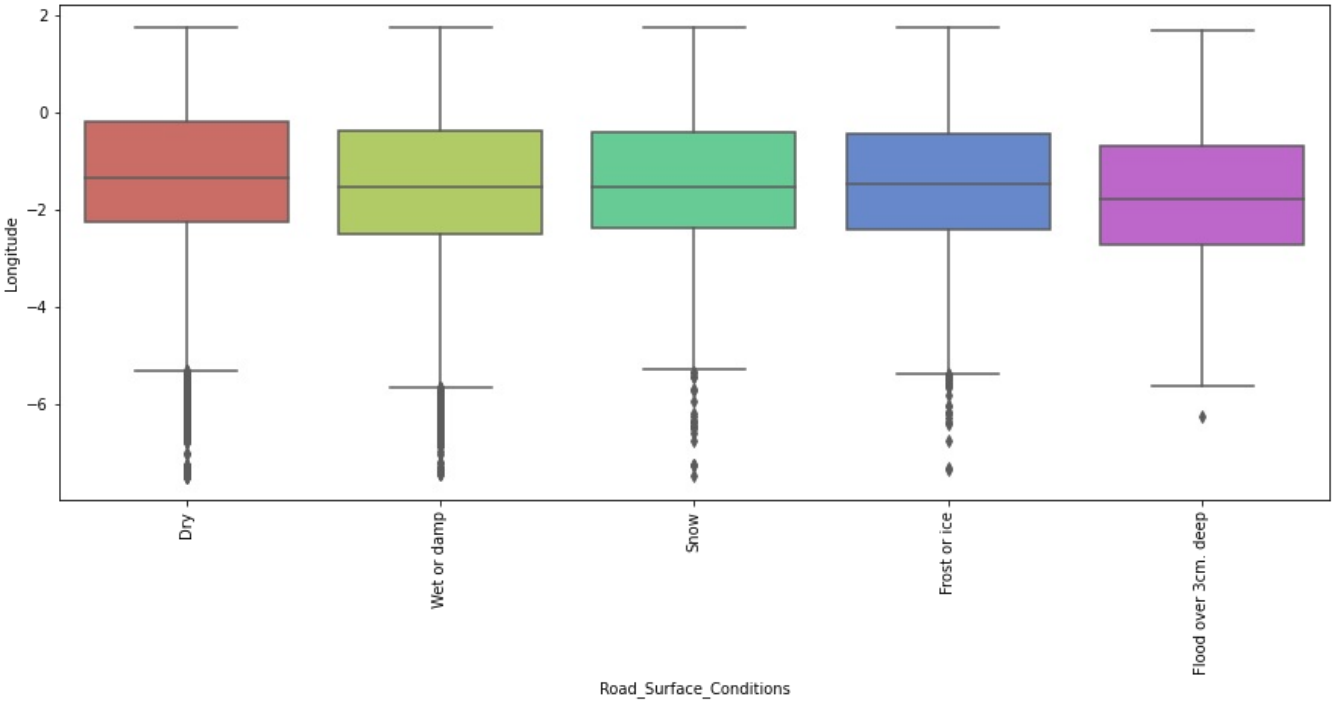

Latitude
50
52
54
56
58
60
Darkness - lights lit
Daylight
Darkness - lighting unknown
Darkness - lights unlit
Darkness - no lighting
Light_Conditions

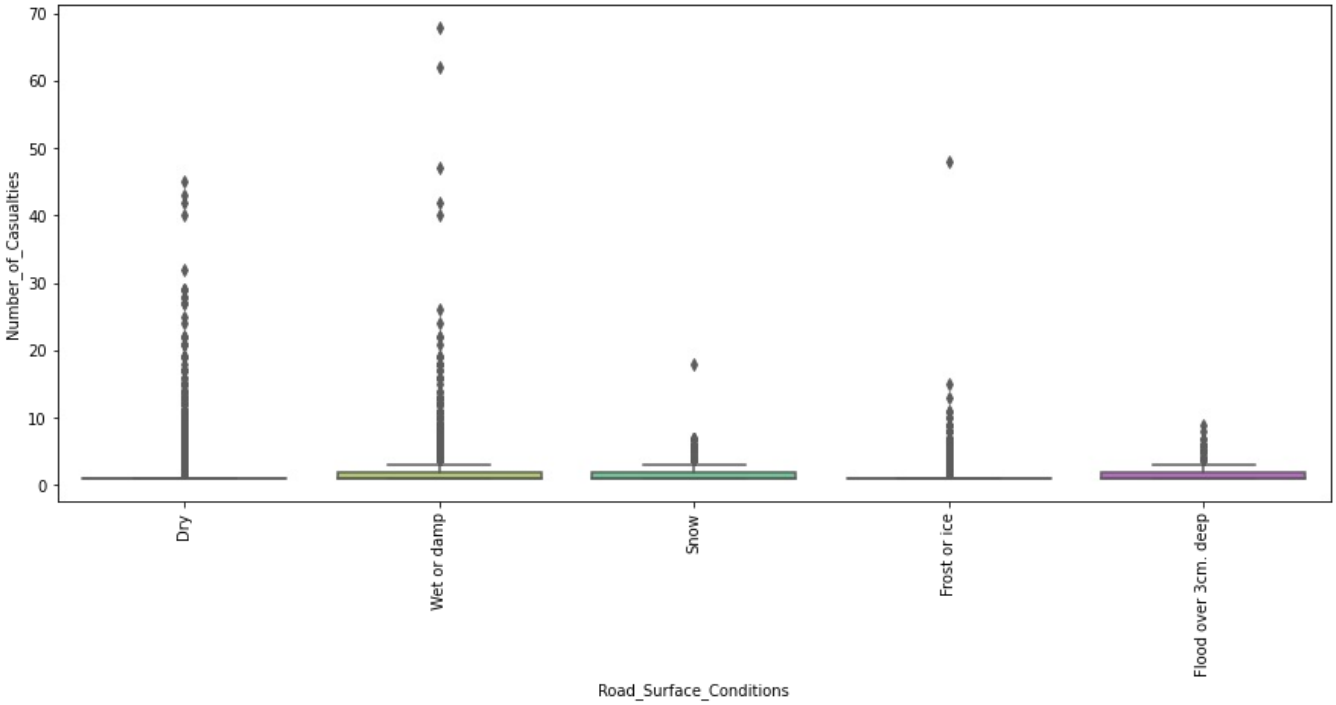

Longitude
2
0
−2
−4
−6
Darkness - lights lit
Daylight
Darkness - lighting unknown
Darkness - lights unlit
Darkness - no lighting
Light_Conditions

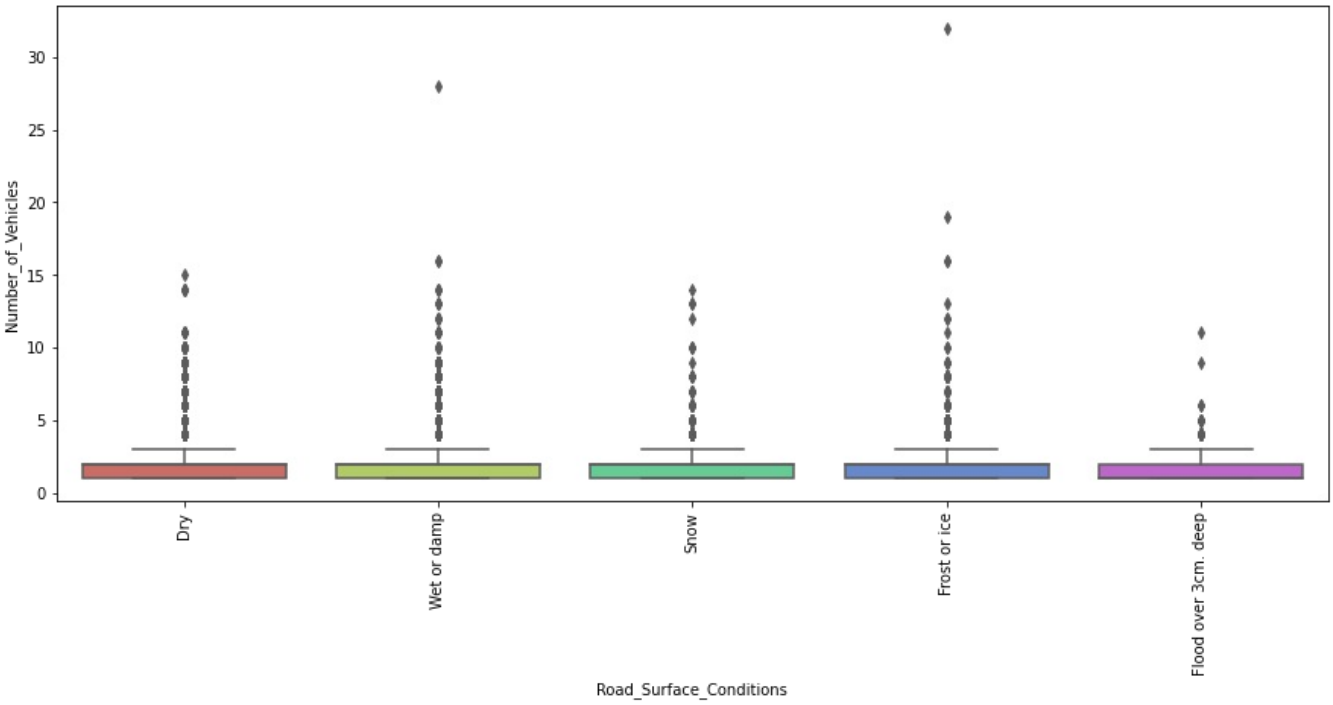

Number_of_Casualties
70
60
50
40
30
20
10
0
Darkness - lights lit
Daylight
Darkness - lighting unknown
Darkness - lights unlit
Darkness - no lighting
Light_Conditions

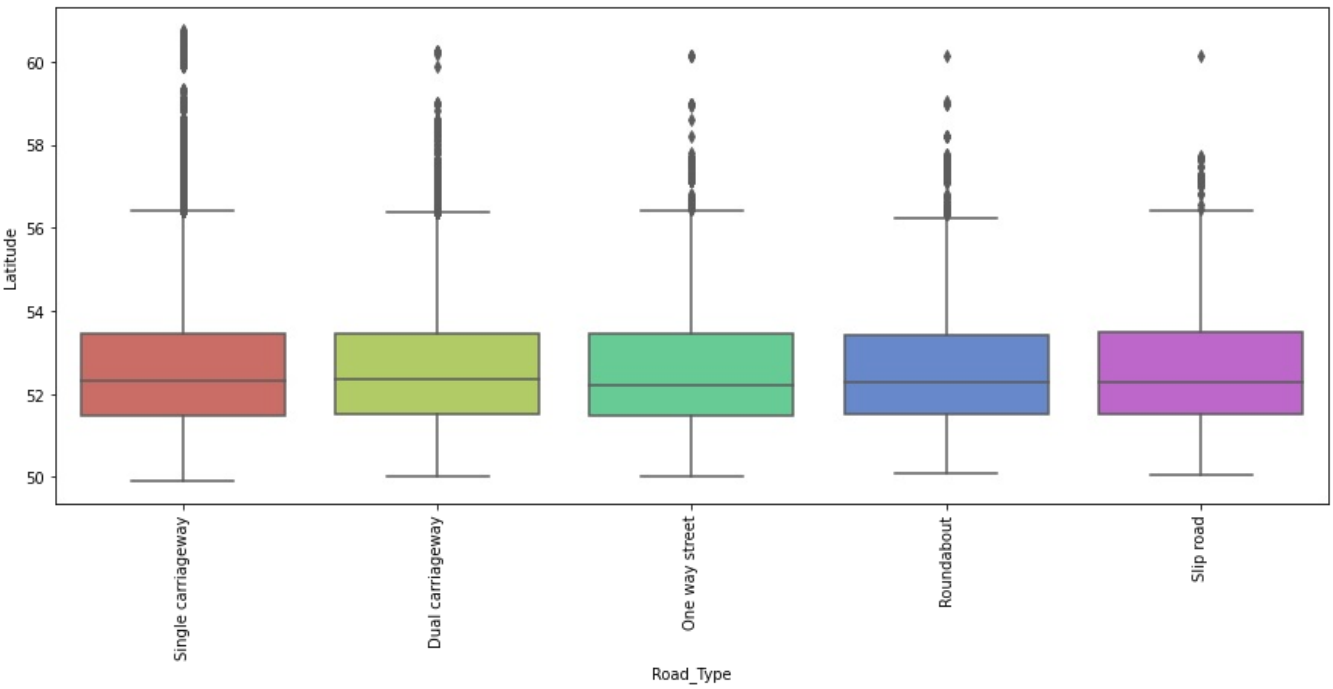

Number_of_Vehicles
0
5
10
15
20
25
30
Darkness - lights lit
Daylight
Darkness - lighting unknown
Darkness - lights unlit
Darkness - no lighting
Light_Conditions

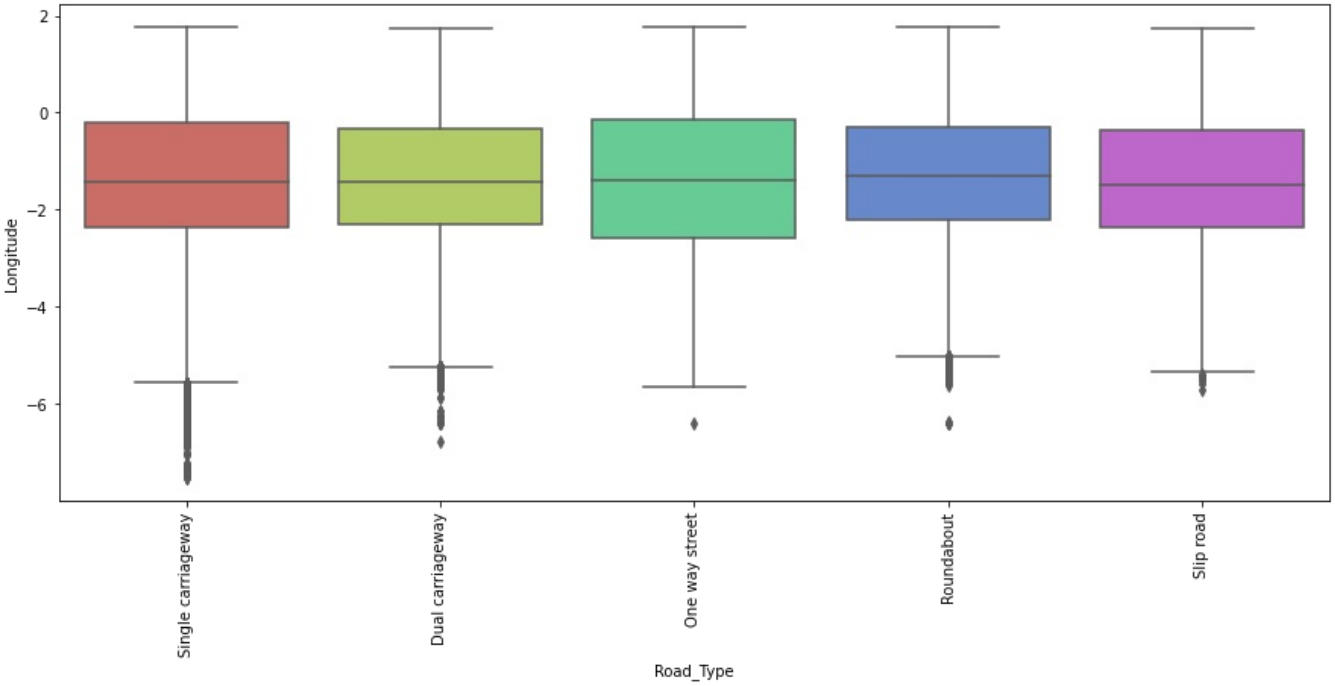

Latitude
50
52
54
56
58
60
Dry
Wet or damp
Snow
Frost or ice
Flood over 3cm. deep
Road_Surface_Conditions

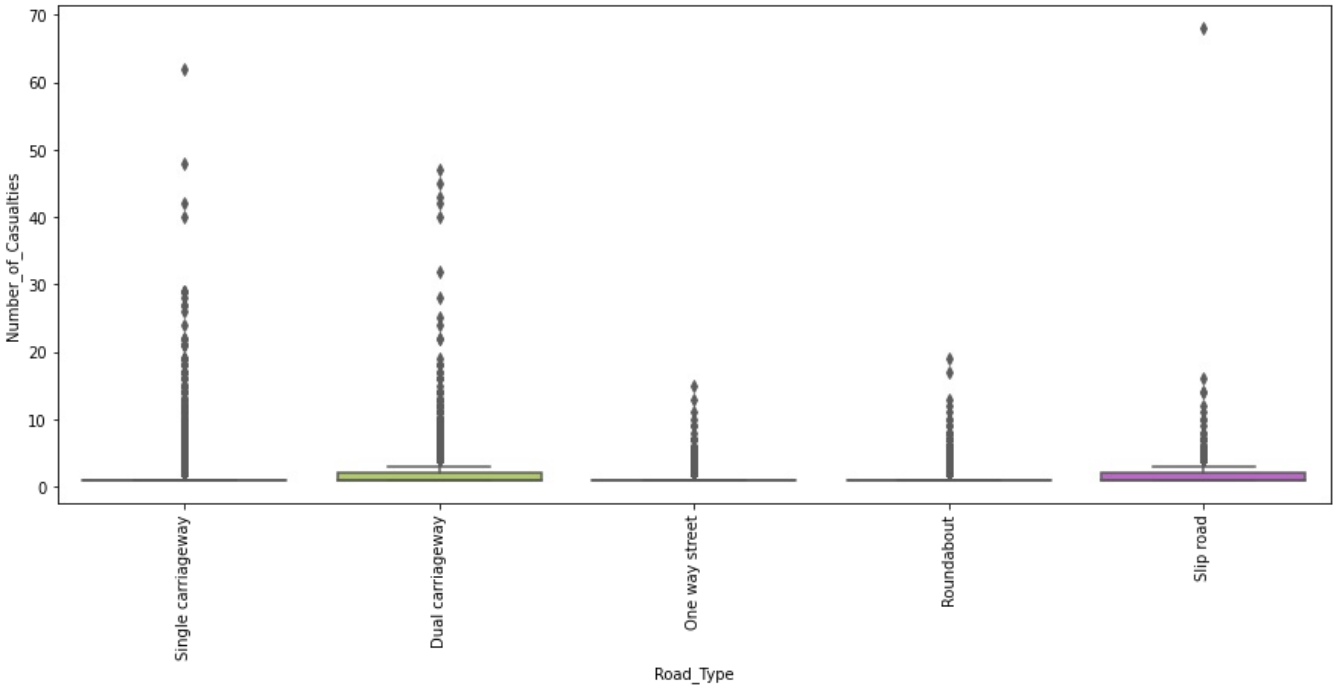

Longitude
2
0
−2
−4
−6
Dry
Wet or damp
Snow
Frost or ice
Flood over 3cm. deep
Road_Surface_Conditions

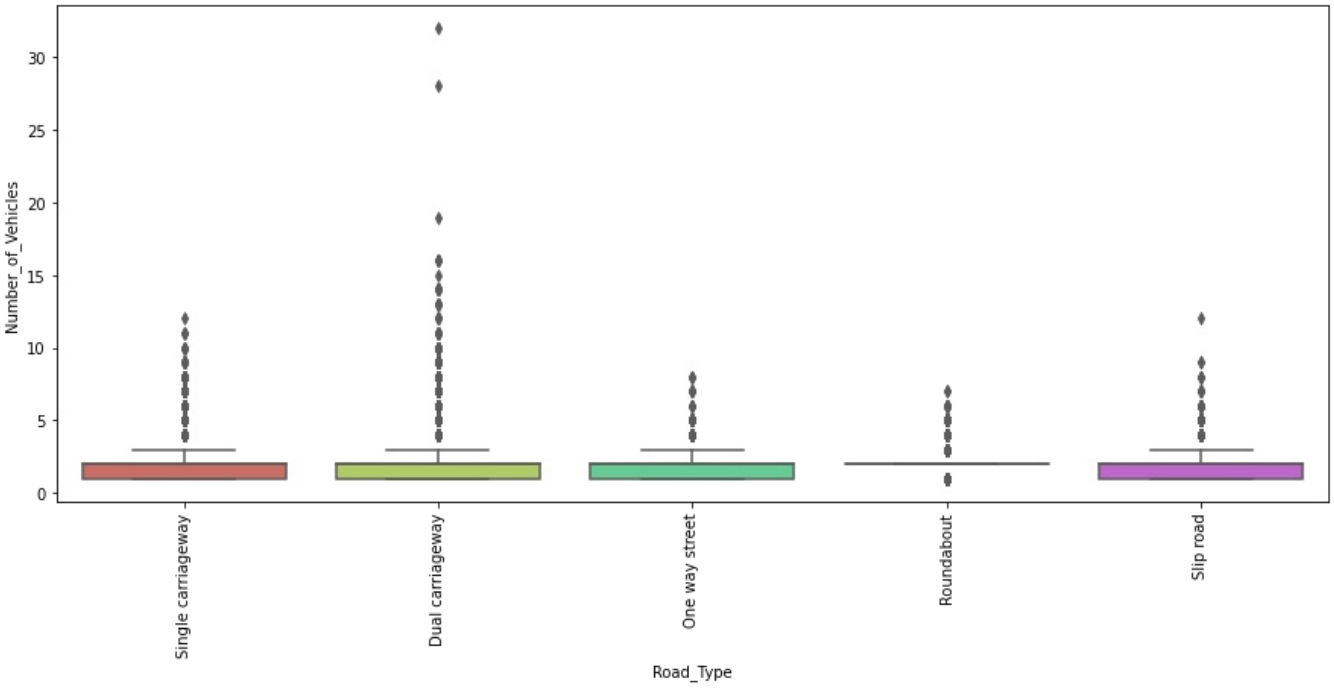

Number_of_Casualties
70
60
50
40
30
20
10
0
Dry
Wet or damp
Snow
Frost or ice
Flood over 3cm. deep
Road_Surface_Conditions

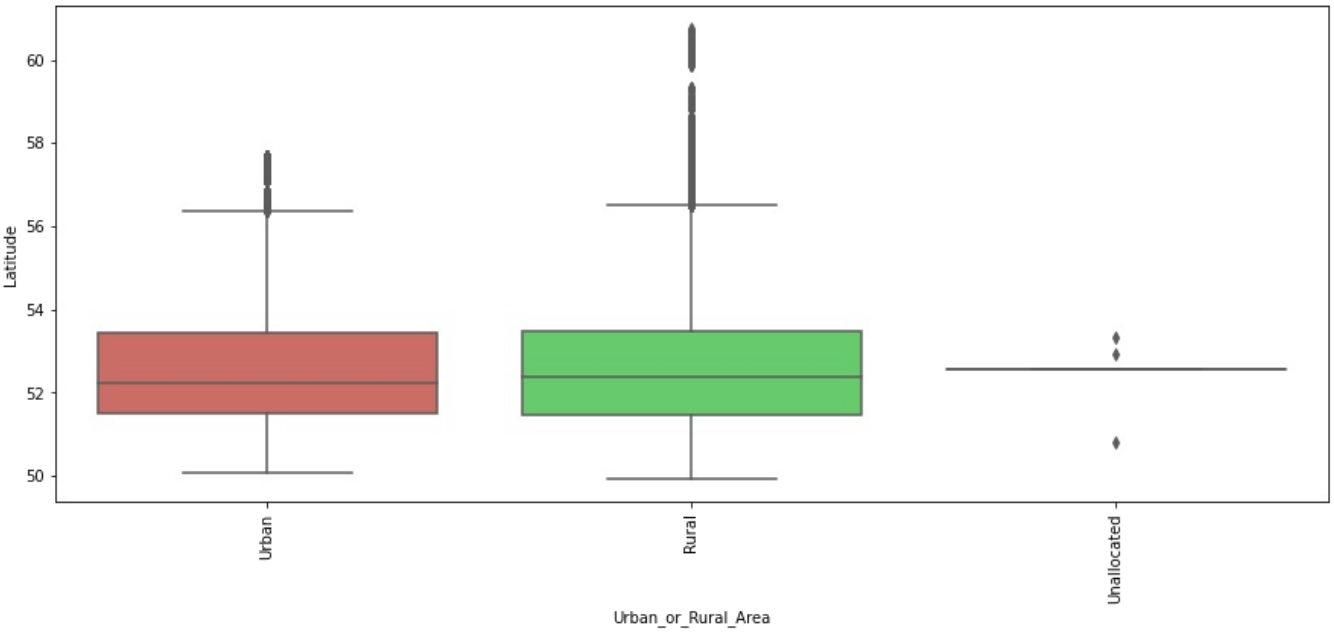

Number_of_Vehicles
0
5
10
15
20
25
30
Dry
Wet or damp
Snow
Frost or ice
Flood over 3cm. deep
Road_Surface_Conditions

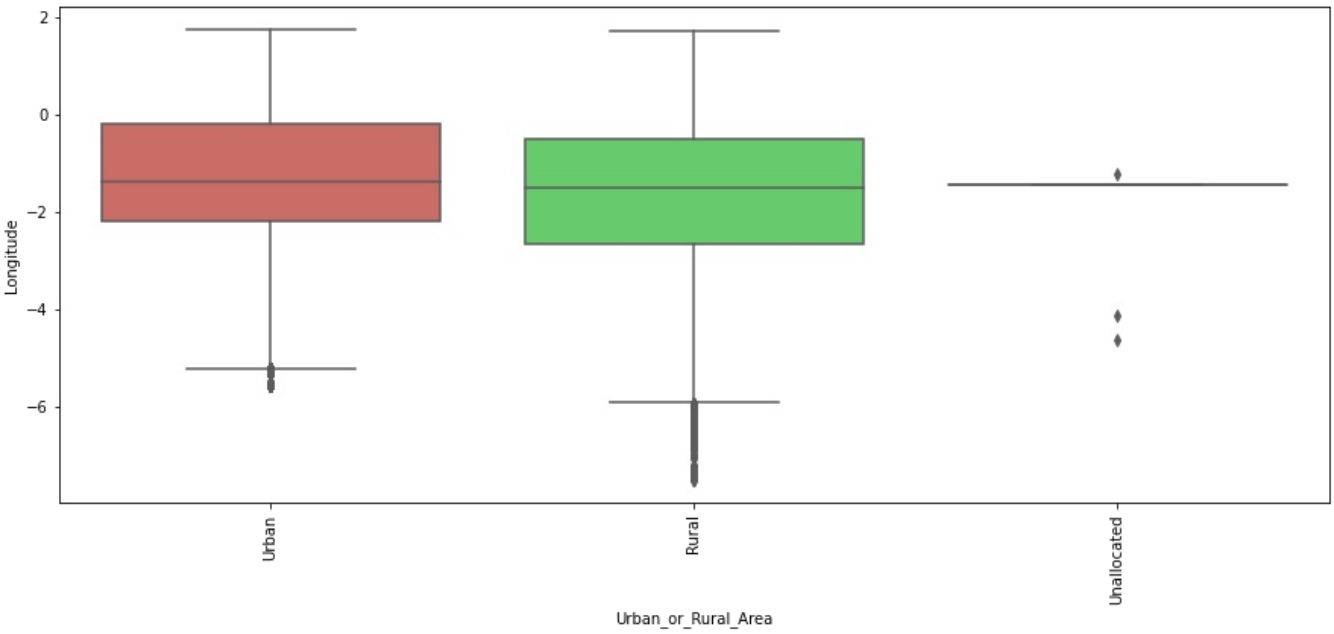

Latitude
50
52
54
56
58
60
Single carriageway
Dual carriageway
One way street
Roundabout
Slip road
Road_Type

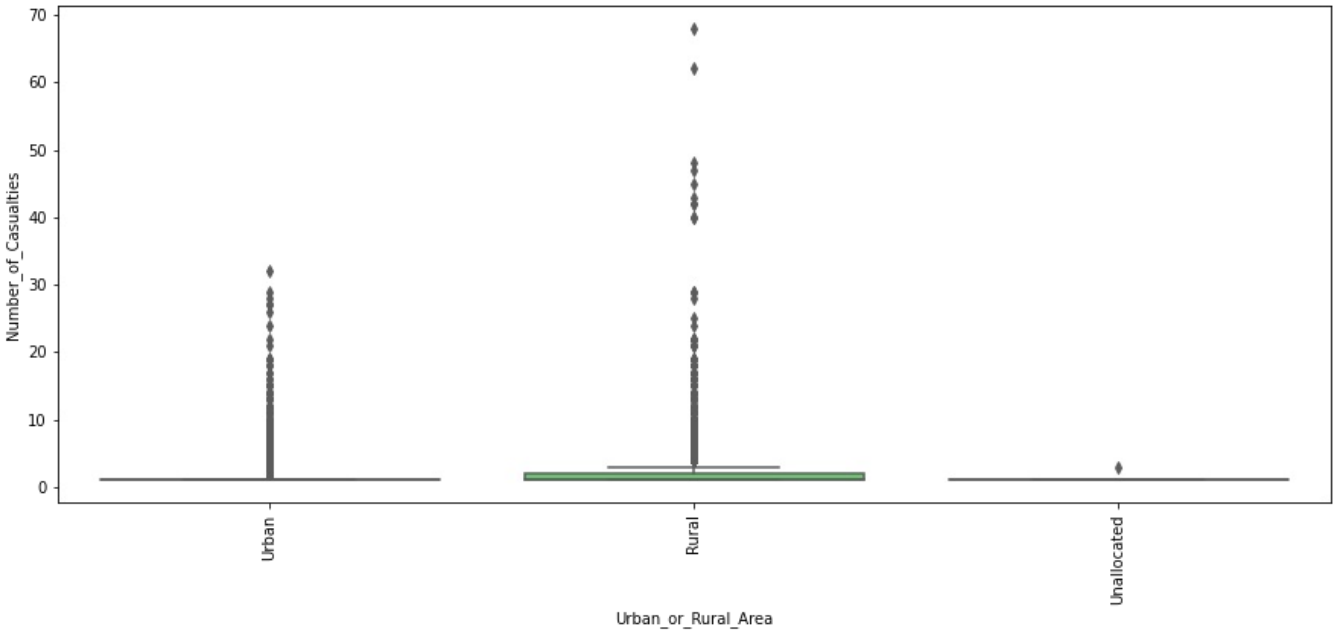

Longitude
2
0
−2
−4
−6
Single carriageway
Dual carriageway
One way street
Roundabout
Slip road
Road_Type

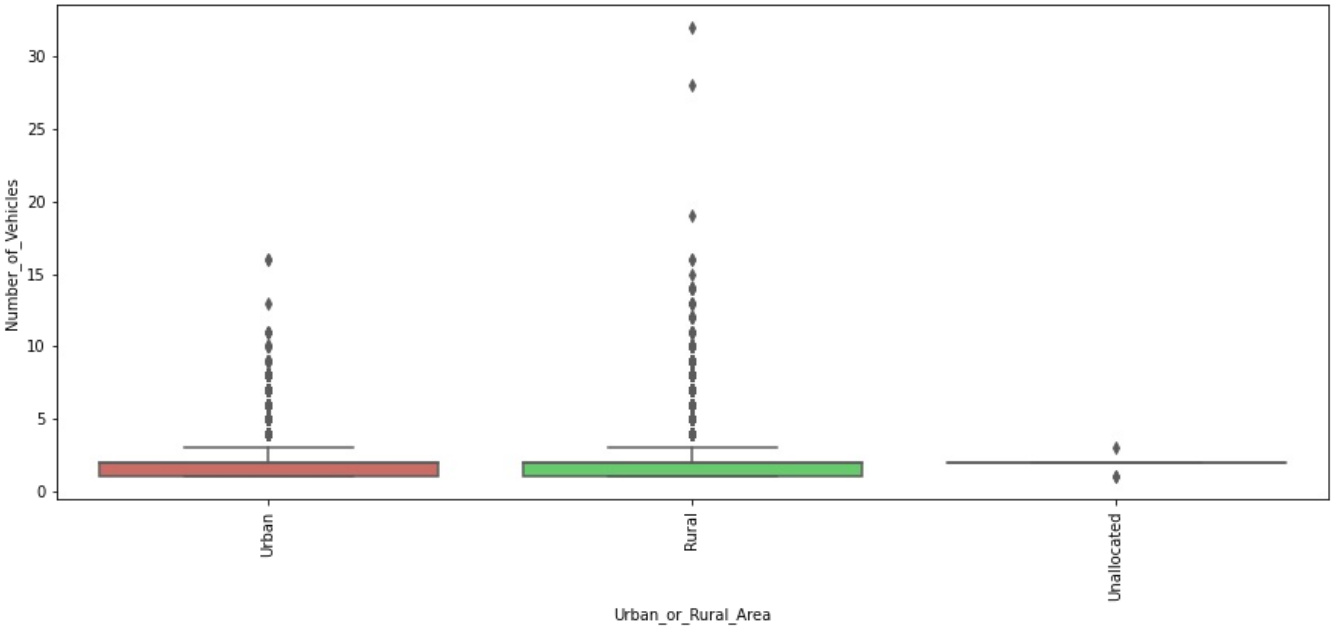

Number_of_Casualties
70
60
50
40
30
20
10
0
Single carriageway
Dual carriageway
One way street
Roundabout
Slip road
Road_Type

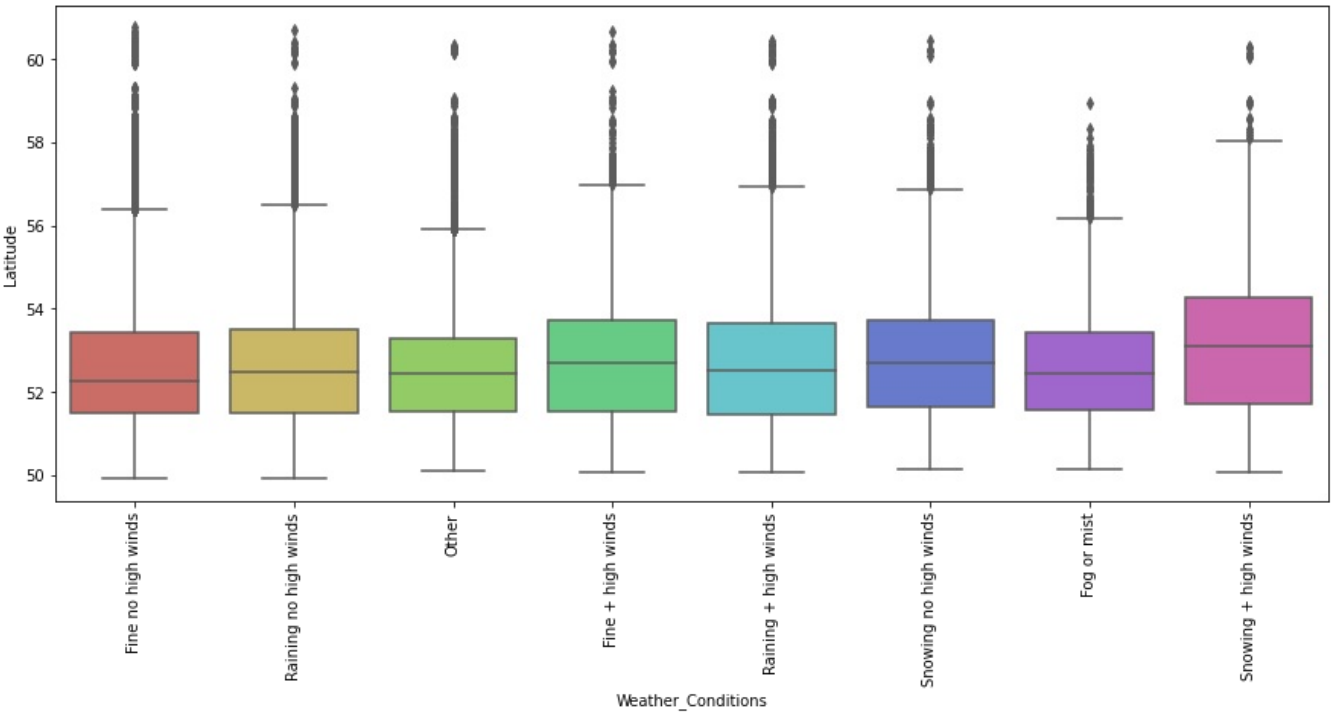

Number_of_Vehicles
0
5
10
15
20
25
30
Single carriageway
Dual carriageway
One way street
Roundabout
Slip road
Road_Type

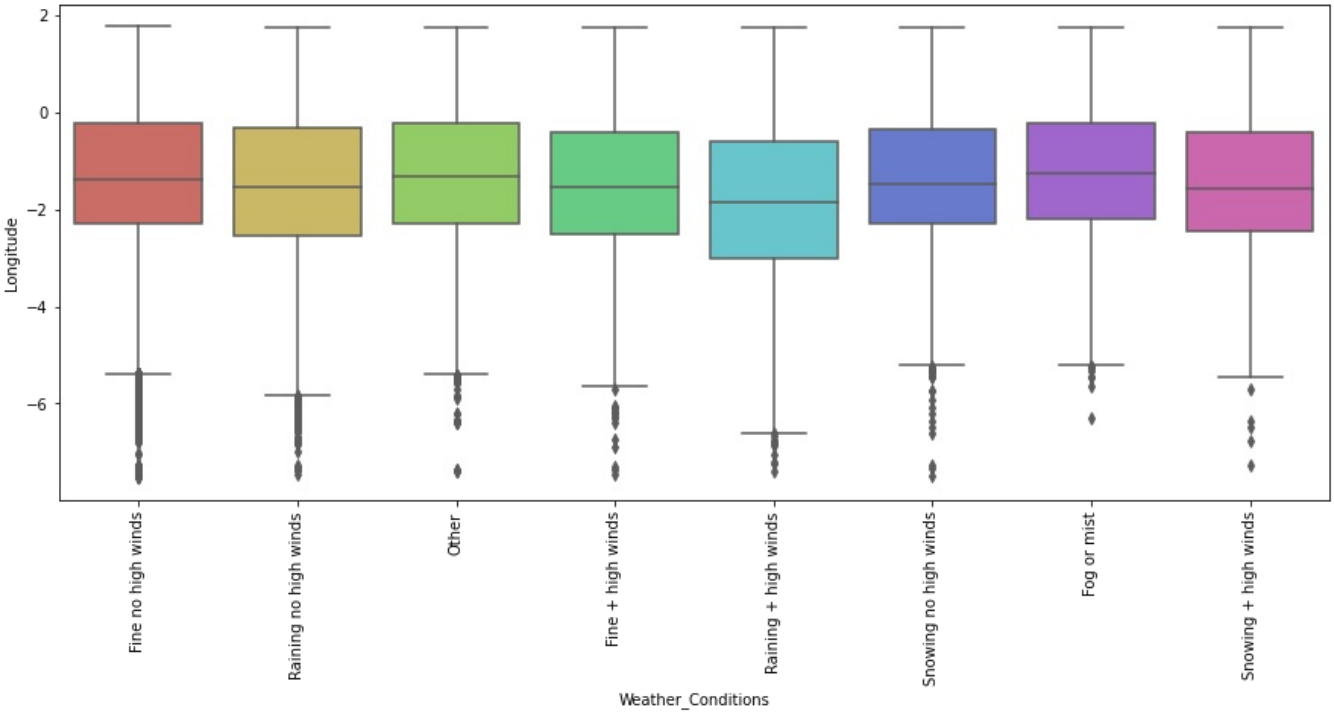

Latitude
50
52
54
56
58
60
Urban
Rural
Unallocated
Urban_or_Rural_Area

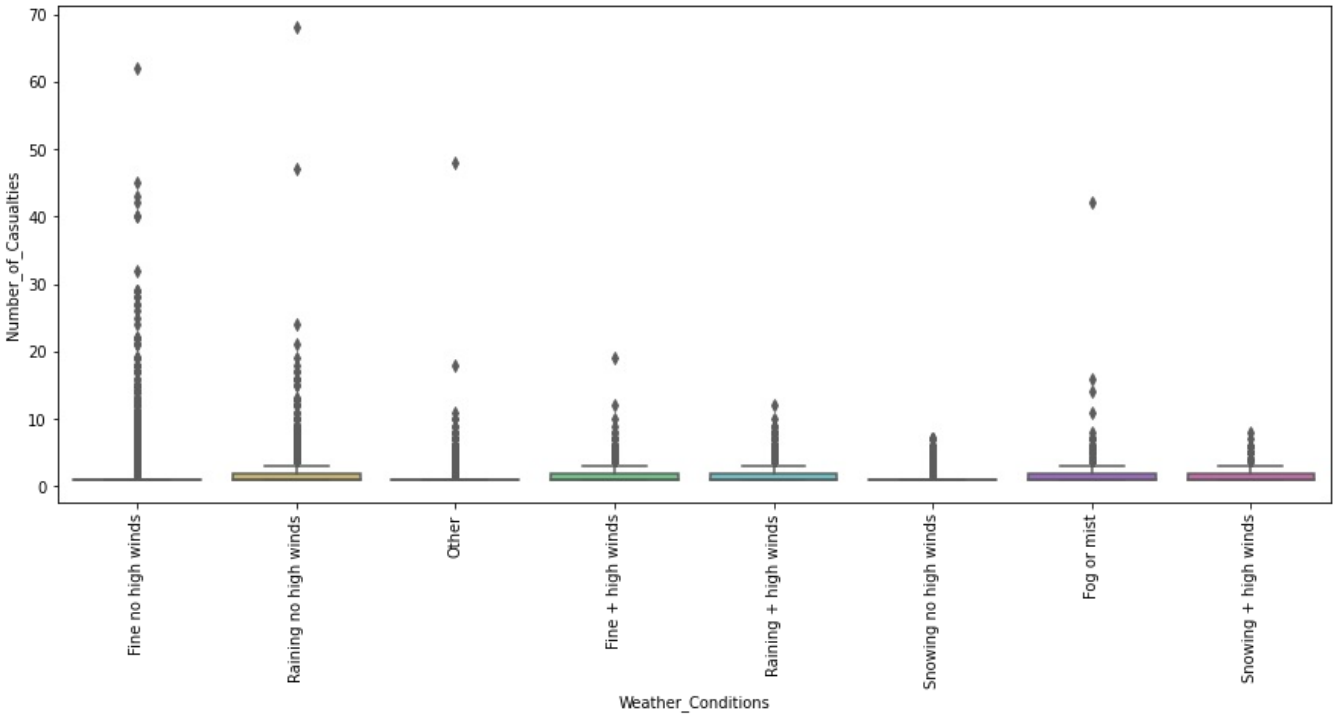

Longitude
2
0
−2
−4
−6
Urban
Rural
Unallocated
Urban_or_Rural_Area

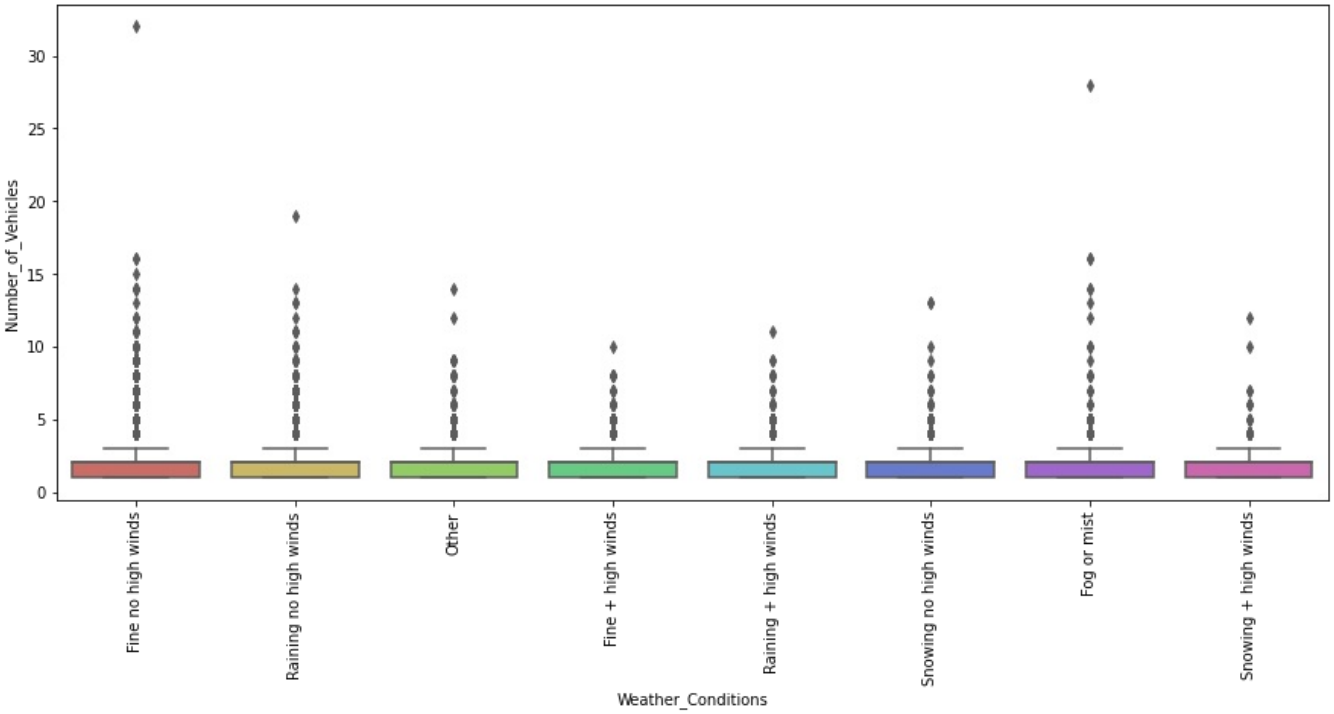

Number_of_Casualties
70
60
50
40
30
20
10
0
Urban
Rural
Unallocated
Urban_or_Rural_Area


Number_of_Vehicles
0
5
10
15
20
25
30
Urban
Rural
Unallocated
Urban_or_Rural_Area

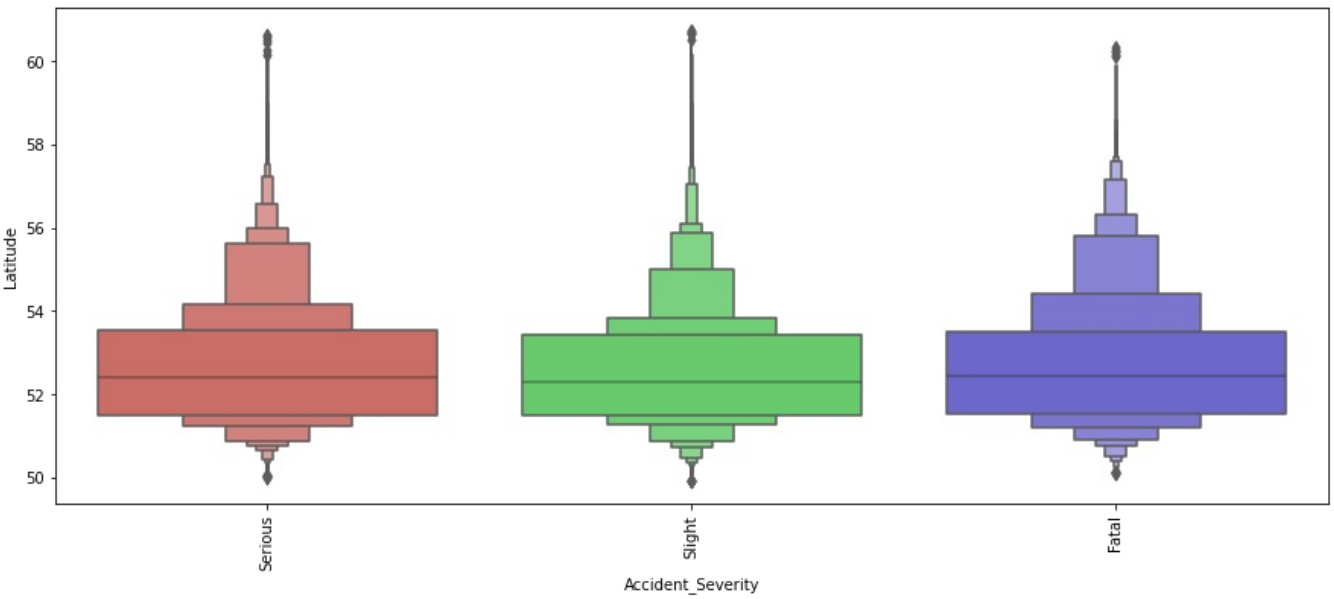

Latitude
50
52
54
56
58
60
Fine no high winds
Raining no high winds
Other
Fine + high winds
Raining + high winds
Snowing no high winds
Fog or mist
Snowing + high winds
Weather_Conditions

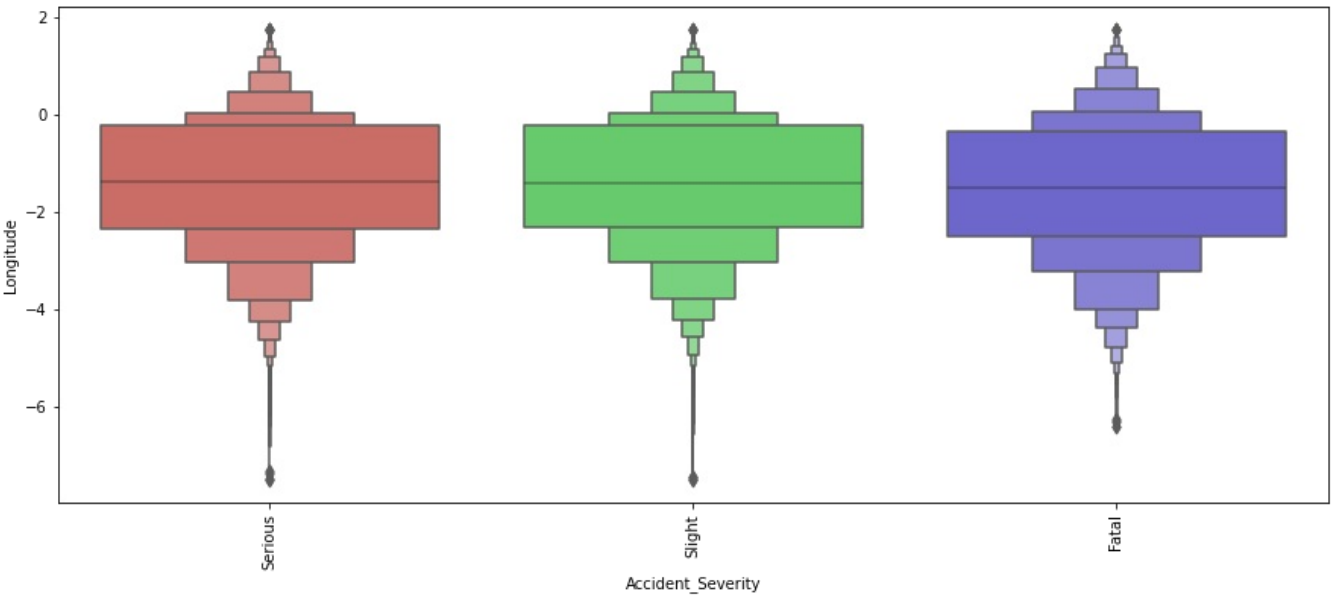

Longitude
2
0
−2
−4
−6
Fine no high winds
Raining no high winds
Other
Fine + high winds
Raining + high winds
Snowing no high winds
Fog or mist
Snowing + high winds
Weather_Conditions

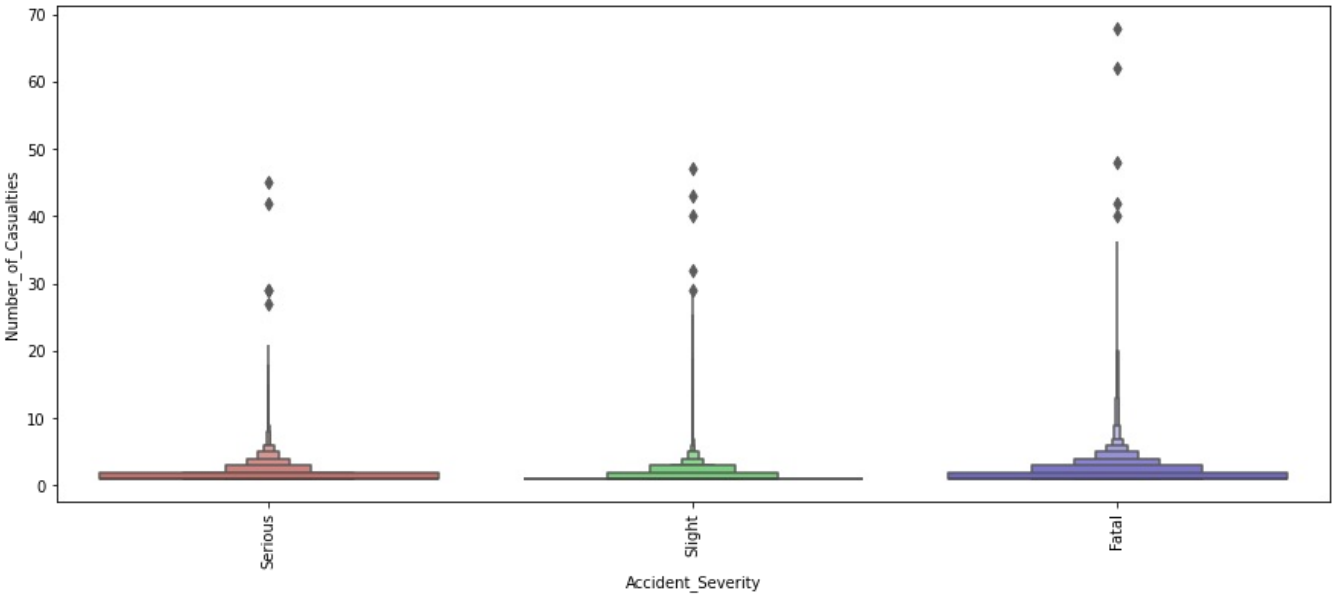

Number_of_Casualties
70
60
50
40
30
20
10
0
Fine no high winds
Raining no high winds
Other
Fine + high winds
Raining + high winds
Snowing no high winds
Fog or mist
Snowing + high winds
Weather_Conditions

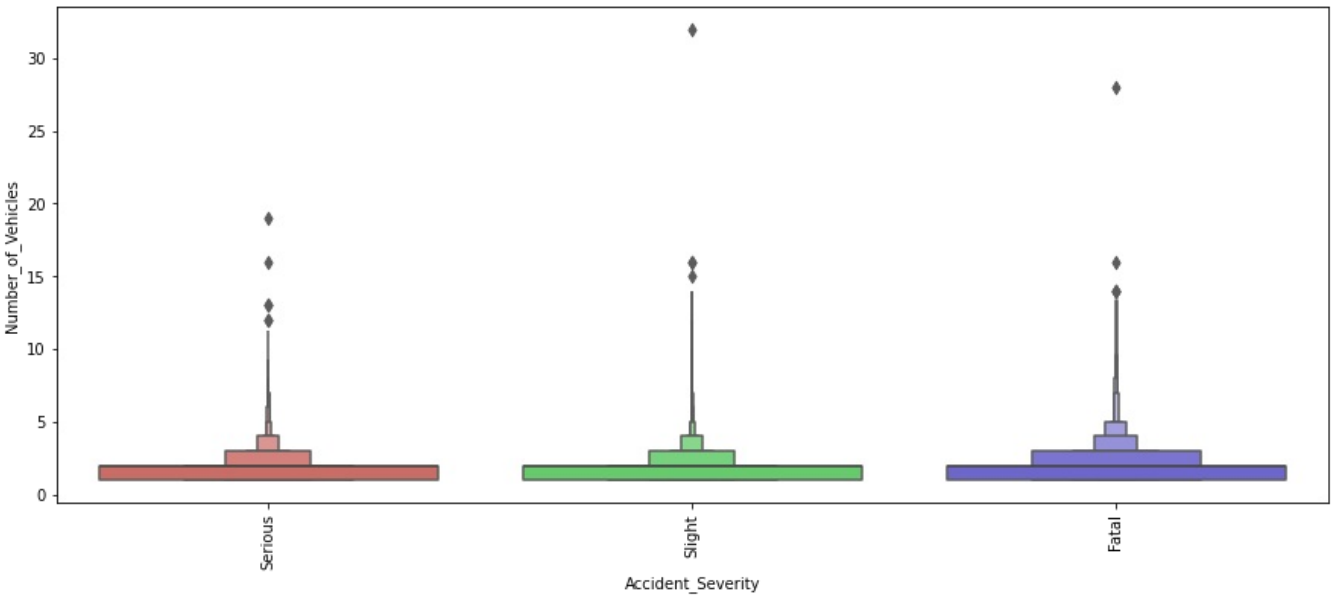


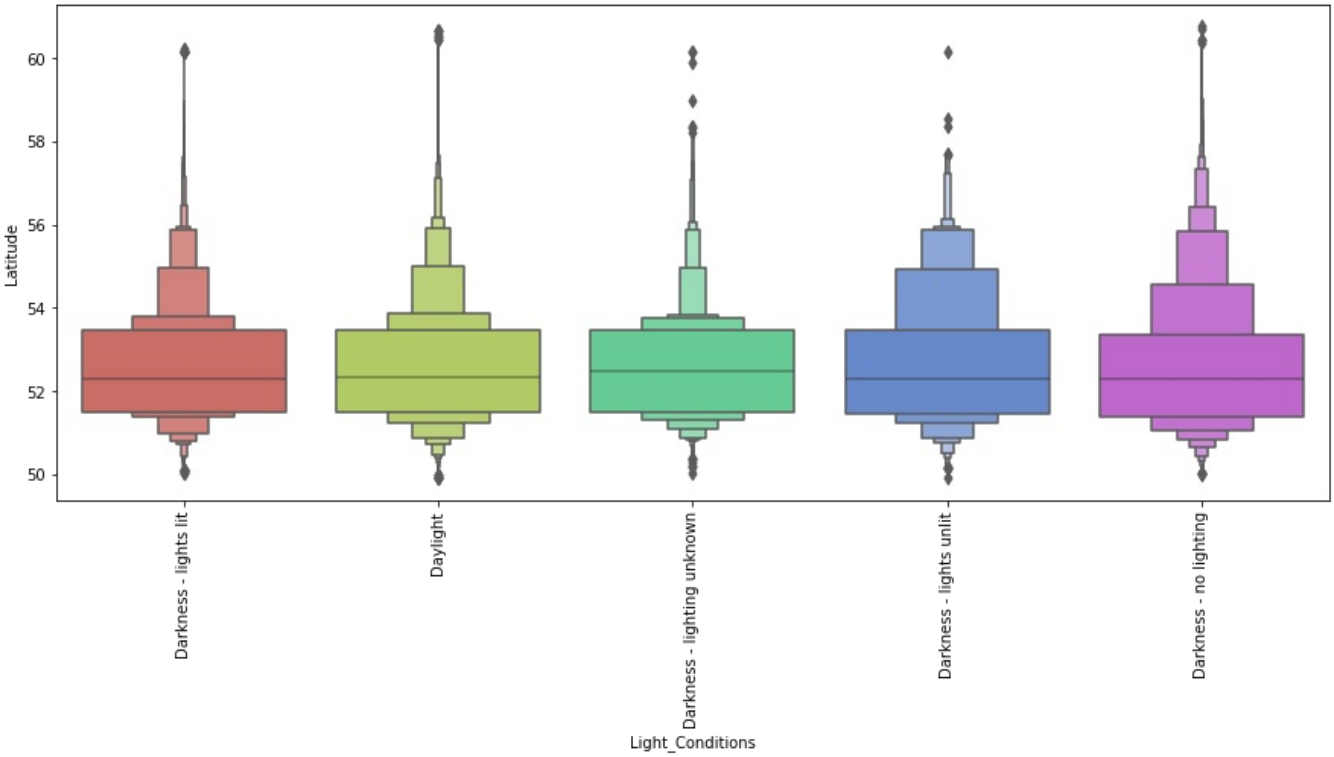
In [43]:
```
for i in categorical:
    for j in continuous:
        plt.figure(figsize=(15,6))
        sns.boxenplot(x = df[i], y = df[j], data = df, palette = 'hls')
        plt.xticks(rotation = 90)
        plt.show()
```

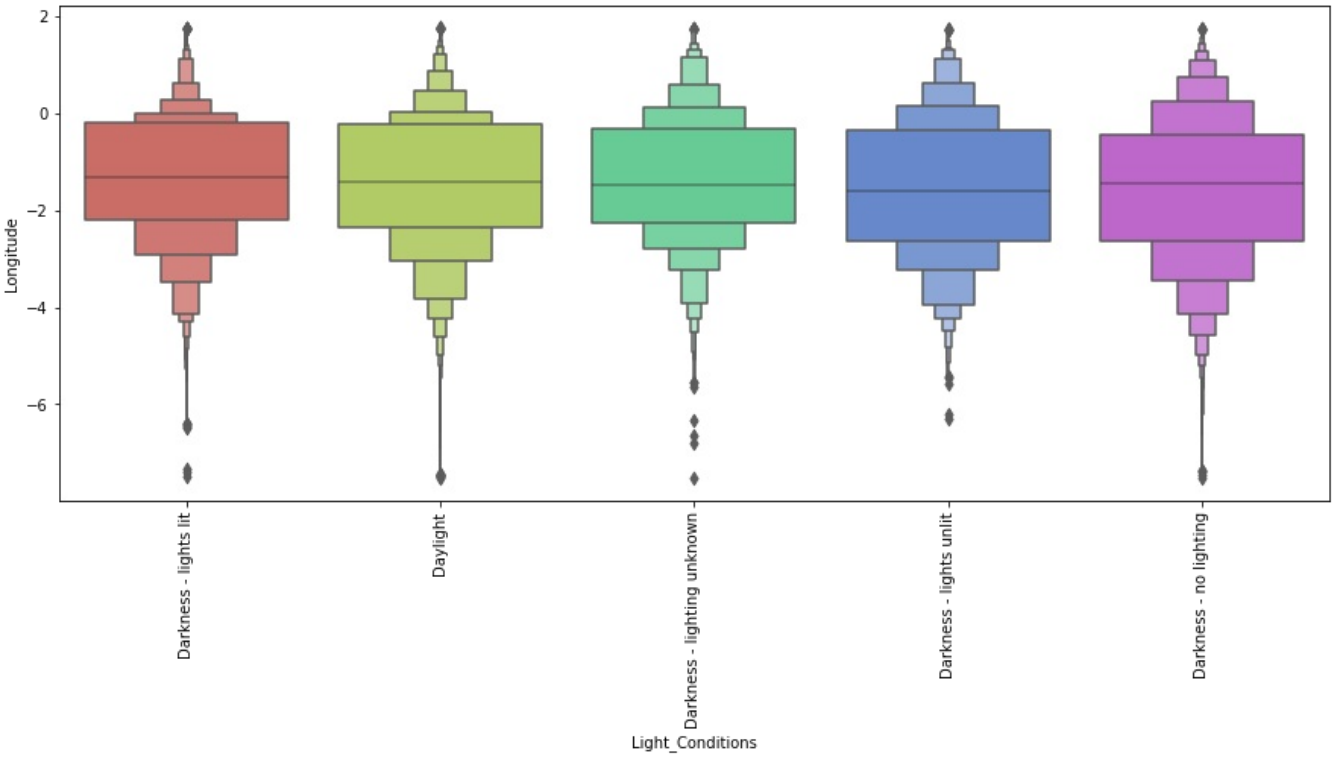


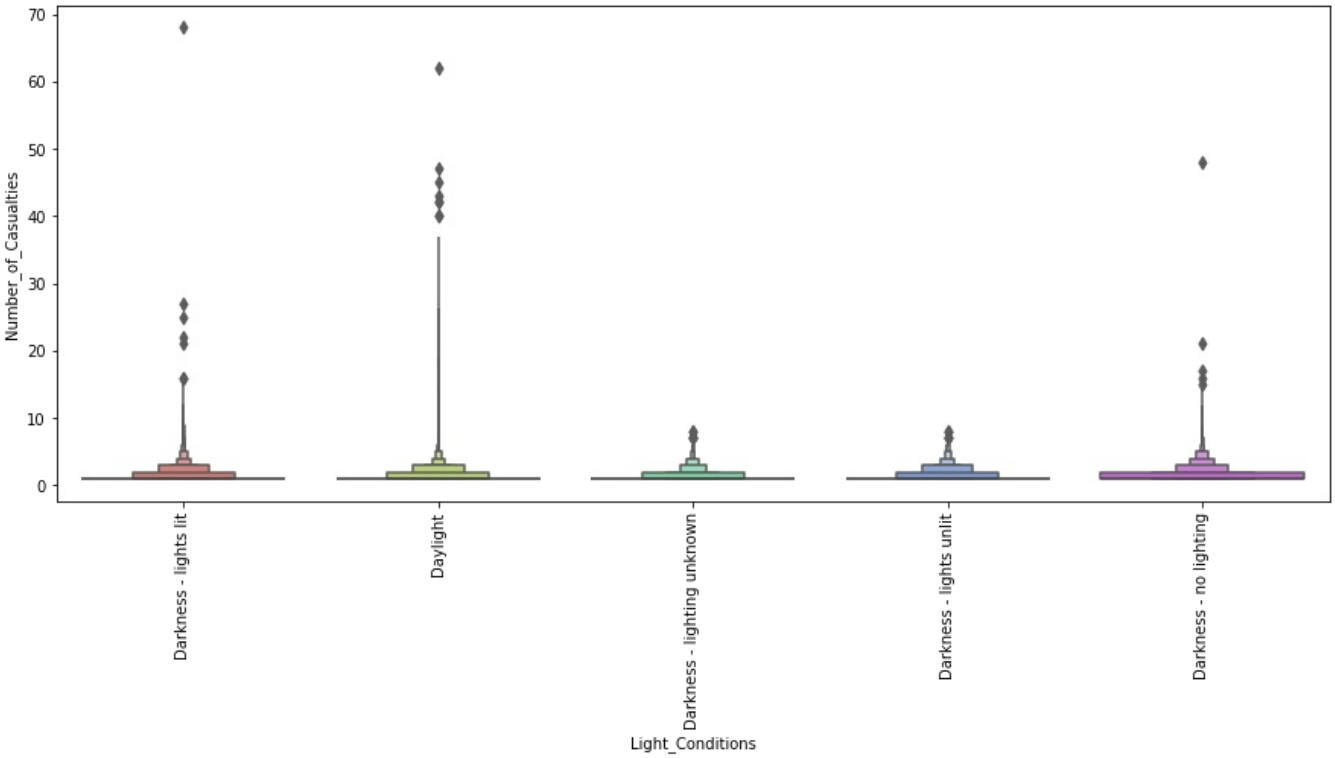

Longitude
2
0
−2
−4
−6
Serious
Slight
Fatal
Accident_Severity

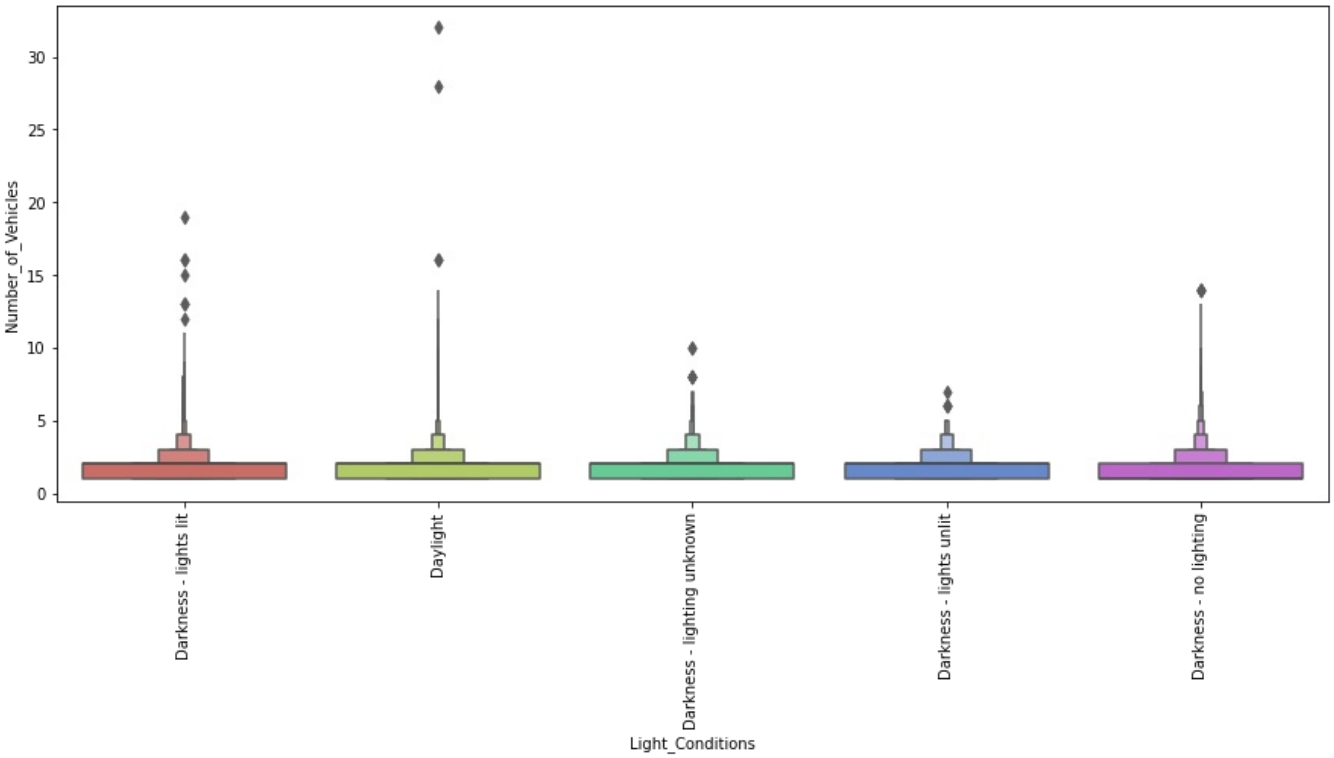

Number_of_Casualties
70
60
50
40
30
20
10
0
Serious
Slight
Fatal
Accident_Severity

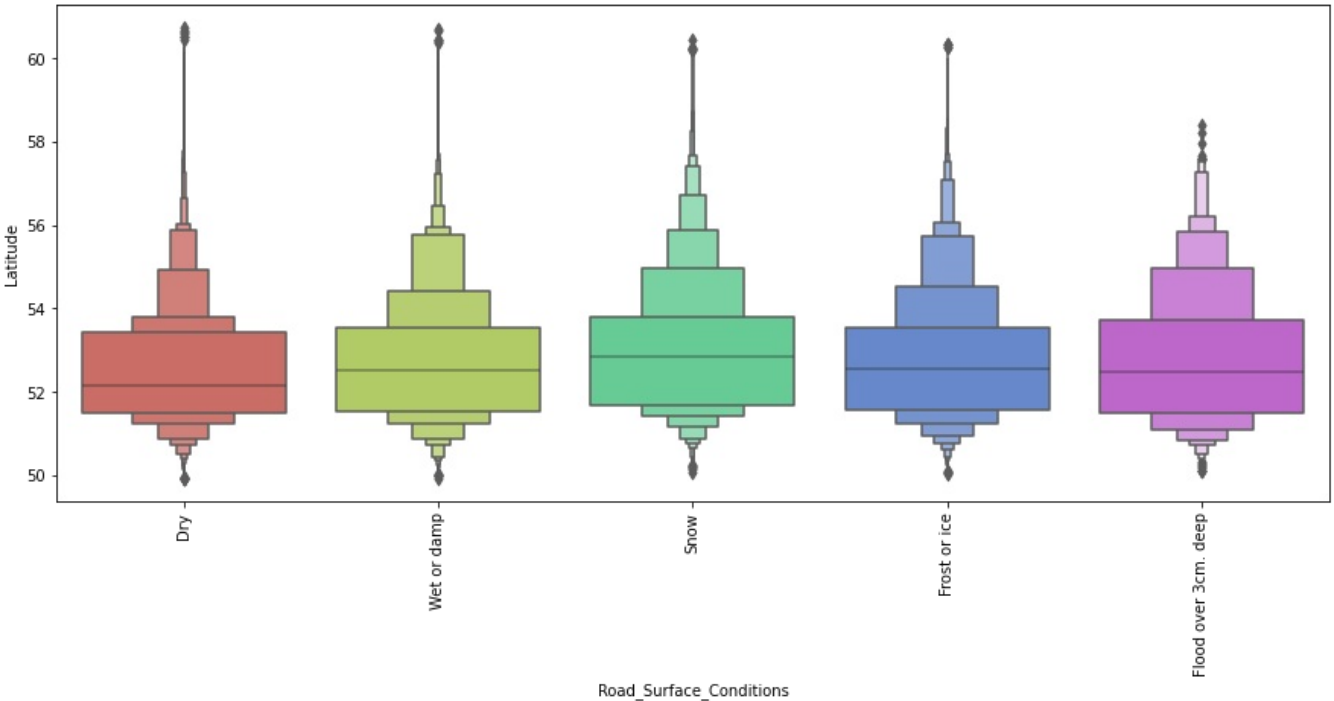

Number_of_Vehicles
30
25
20
15
10
5
0
Serious
Slight
Fatal
Accident_Severity

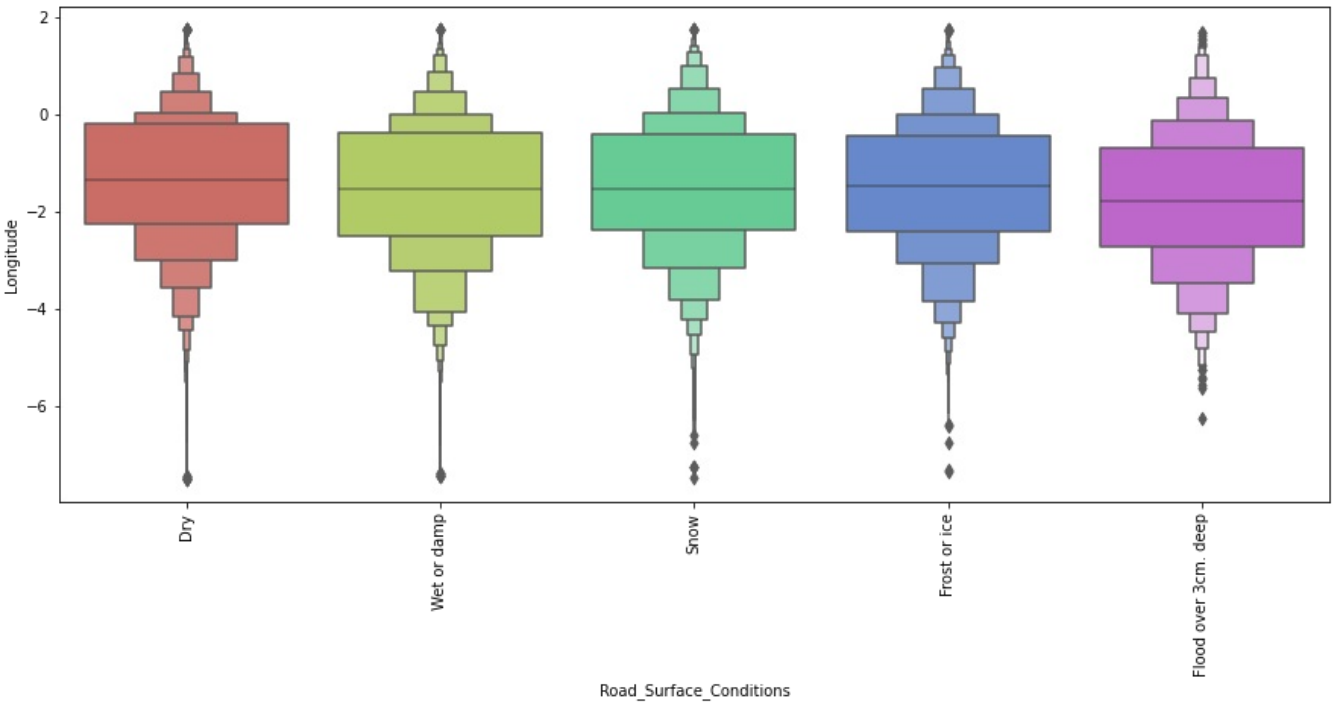

Latitude
60
58
56
54
52
50
Darkness - lights lit
Daylight
Darkness - lighting unknown
Darkness - lights unlit
Darkness - no lighting
Light_Conditions

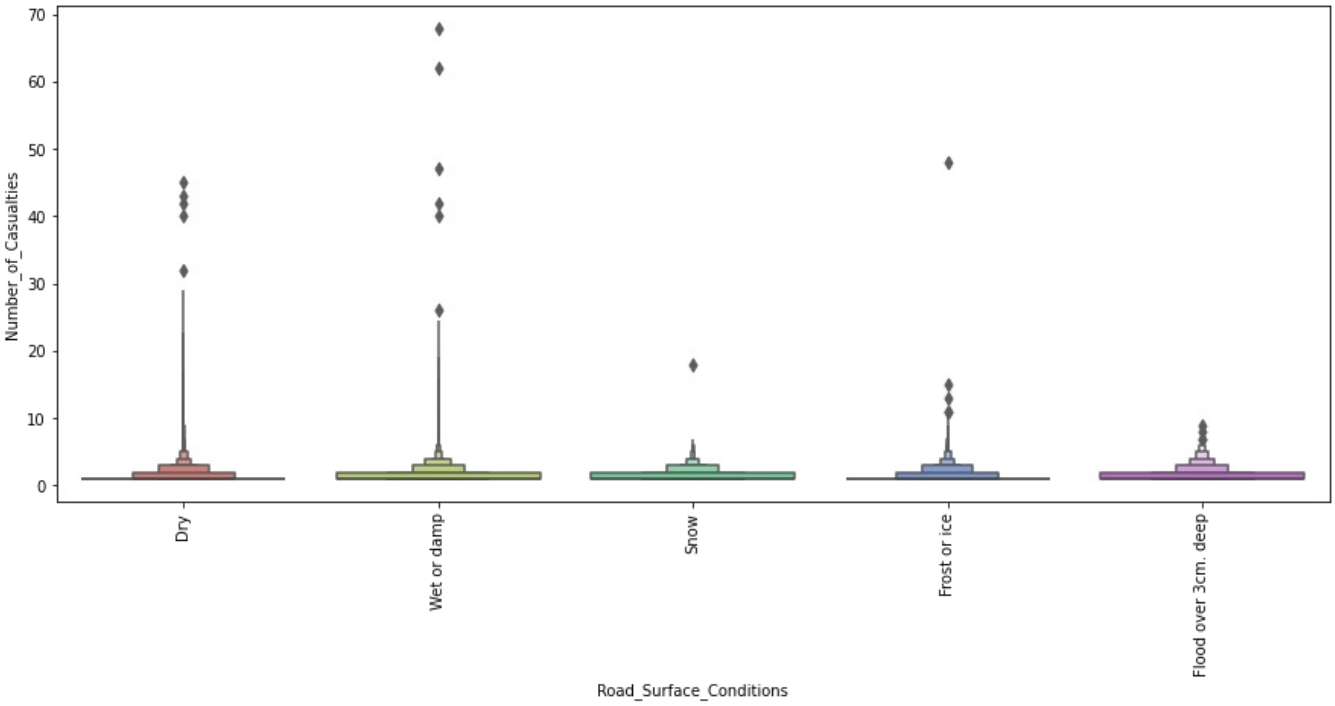

Longitude
2
0
−2
−4
−6
Darkness - lights lit
Daylight
Darkness - lighting unknown
Darkness - lights unlit
Darkness - no lighting
Light_Conditions

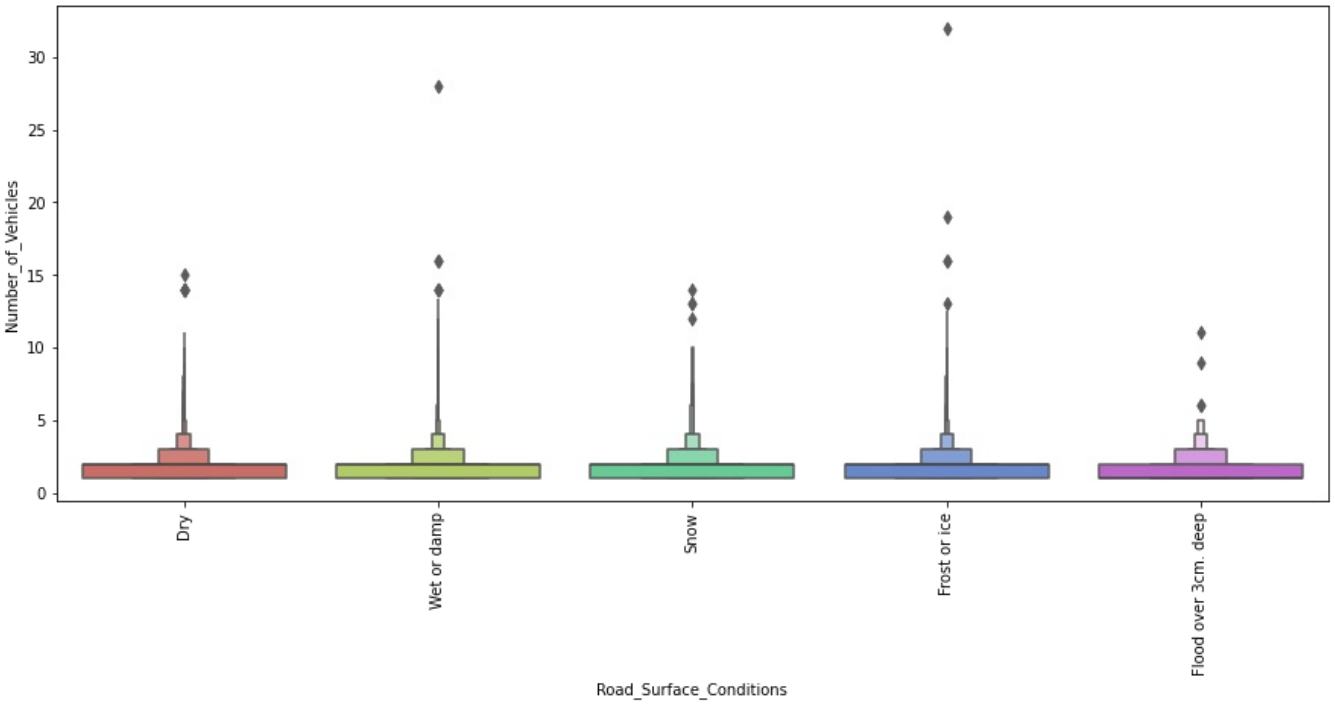

Number_of_Casualties
0
10
20
30
40
50
60
70
Darkness - lights lit
Daylight
Darkness - lighting unknown
Darkness - lights unlit
Darkness - no lighting
Light_Conditions

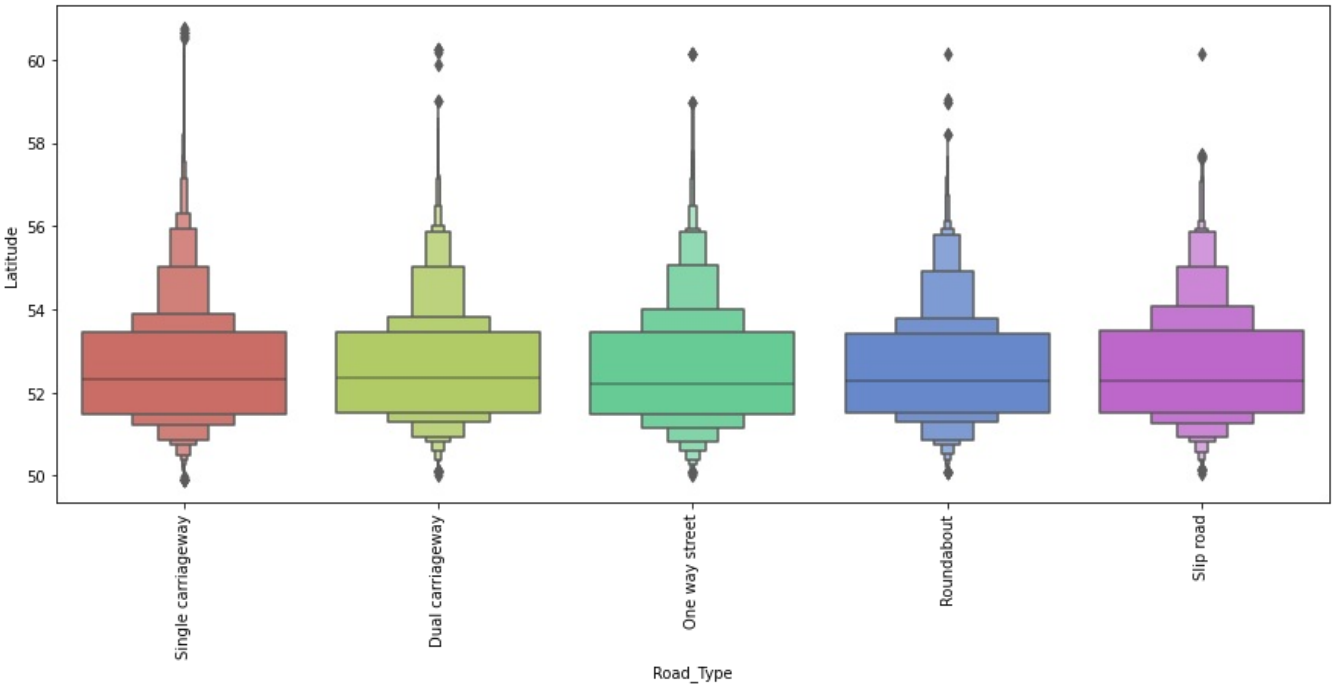

Number_of_Vehicles
0
5
10
15
20
25
30
Darkness - lights lit
Daylight
Darkness - lighting unknown
Darkness - lights unlit
Darkness - no lighting
Light_Conditions

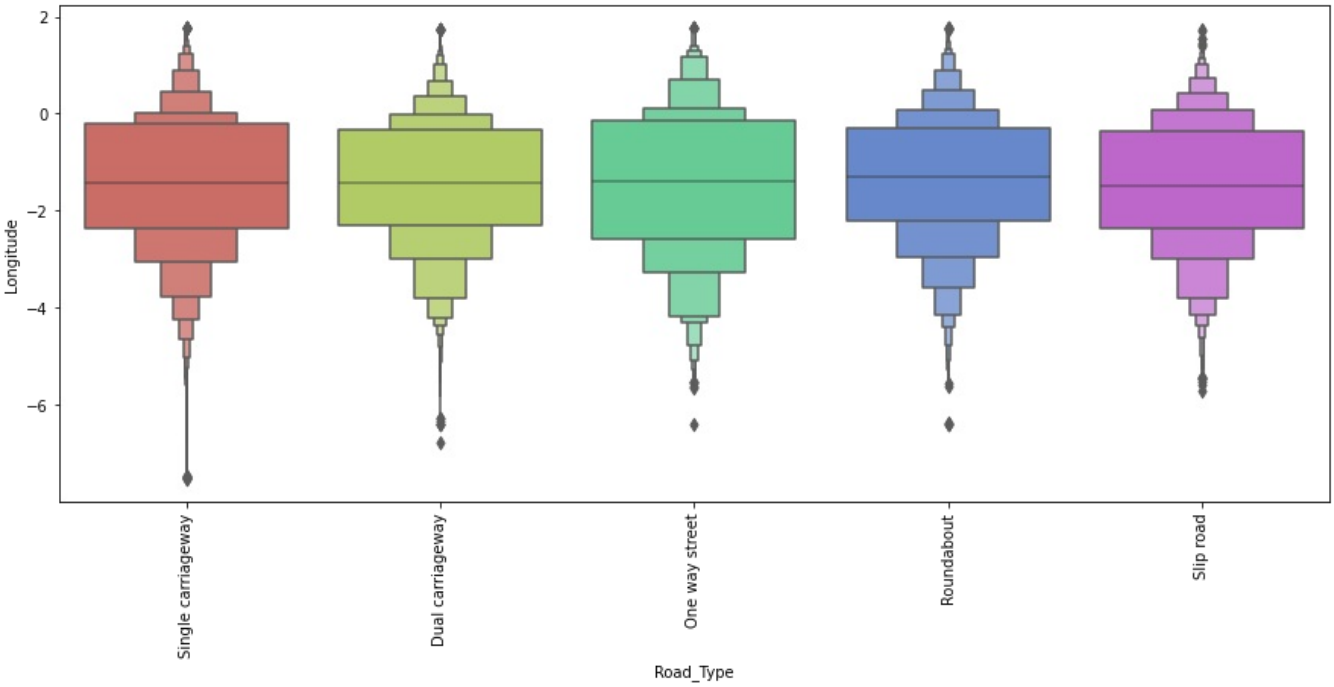

Latitude
60
58
56
54
52
50
Dry
Wet or damp
Snow
Frost or ice
Flood over 3cm. deep
Road_Surface_Conditions

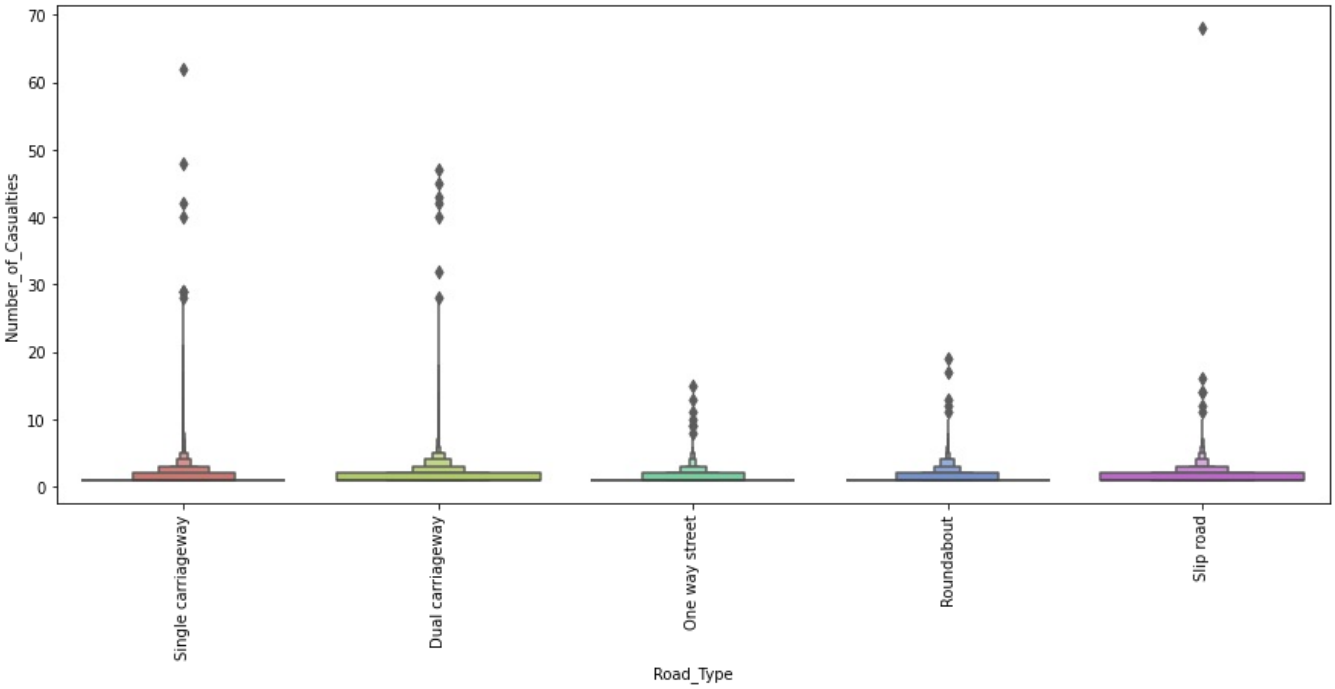

Longitude
2
0
−2
−4
−6
Dry
Wet or damp
Snow
Frost or ice
Flood over 3cm. deep
Road_Surface_Conditions

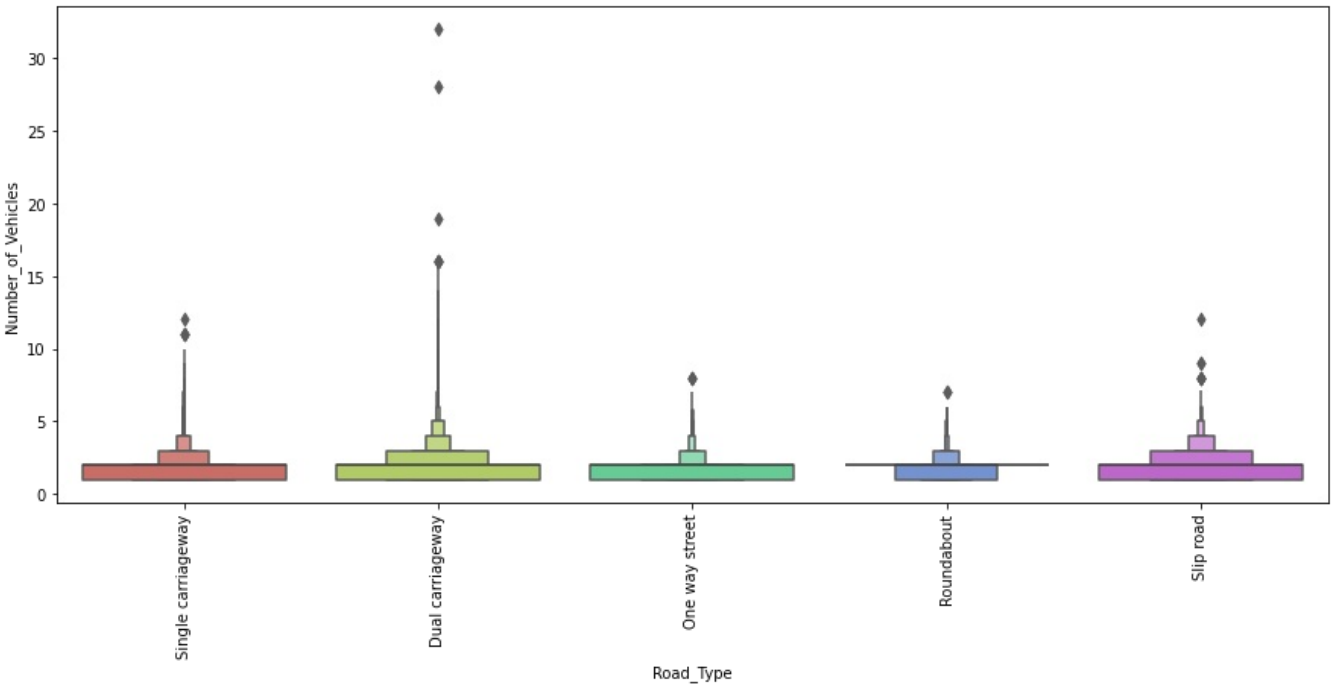

Number_of_Casualties
0
10
20
30
40
50
60
70
Dry
Wet or damp
Snow
Frost or ice
Flood over 3cm. deep
Road_Surface_Conditions

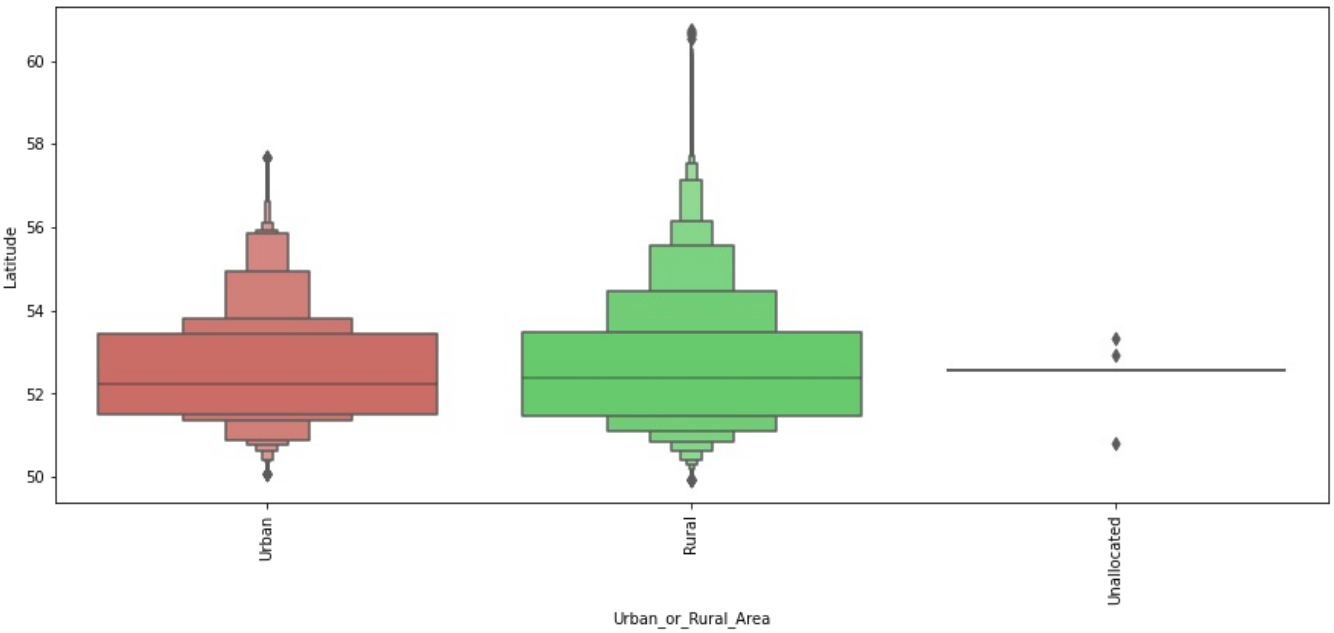

Number_of_Vehicles
0
5
10
15
20
25
30
Dry
Wet or damp
Snow
Frost or ice
Flood over 3cm. deep
Road_Surface_Conditions

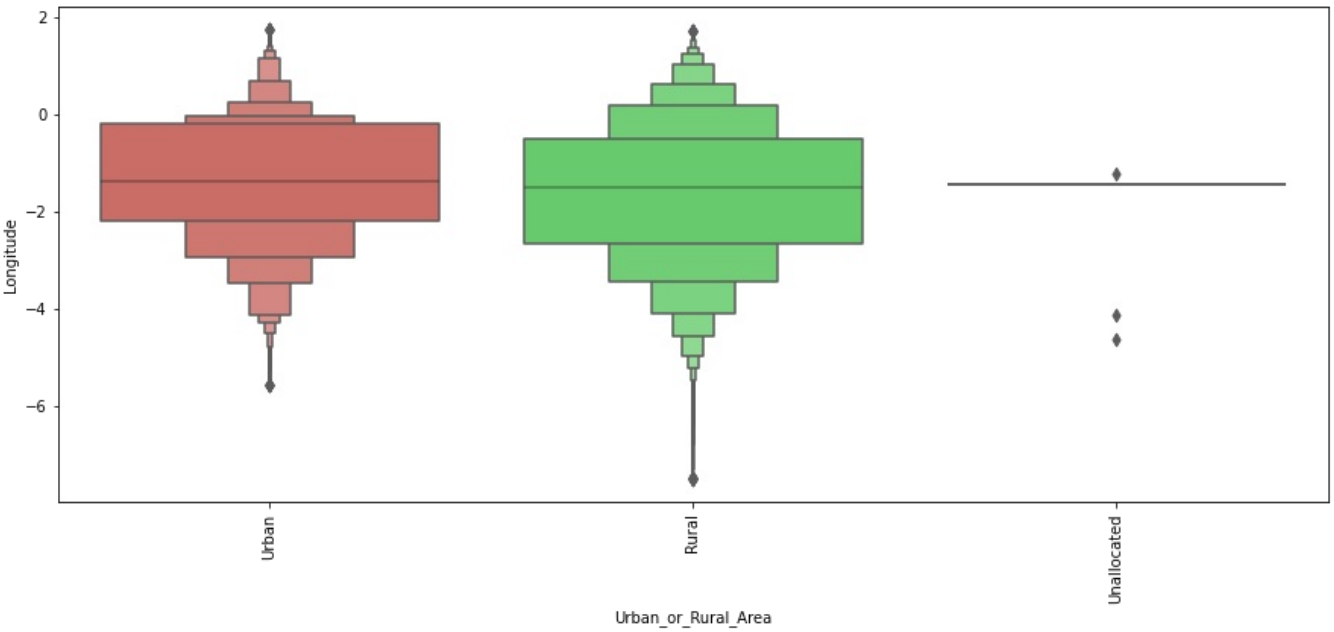

Latitude
60
58
56
54
52
50
Single carriageway
Dual carriageway
One way street
Roundabout
Slip road
Road_Type

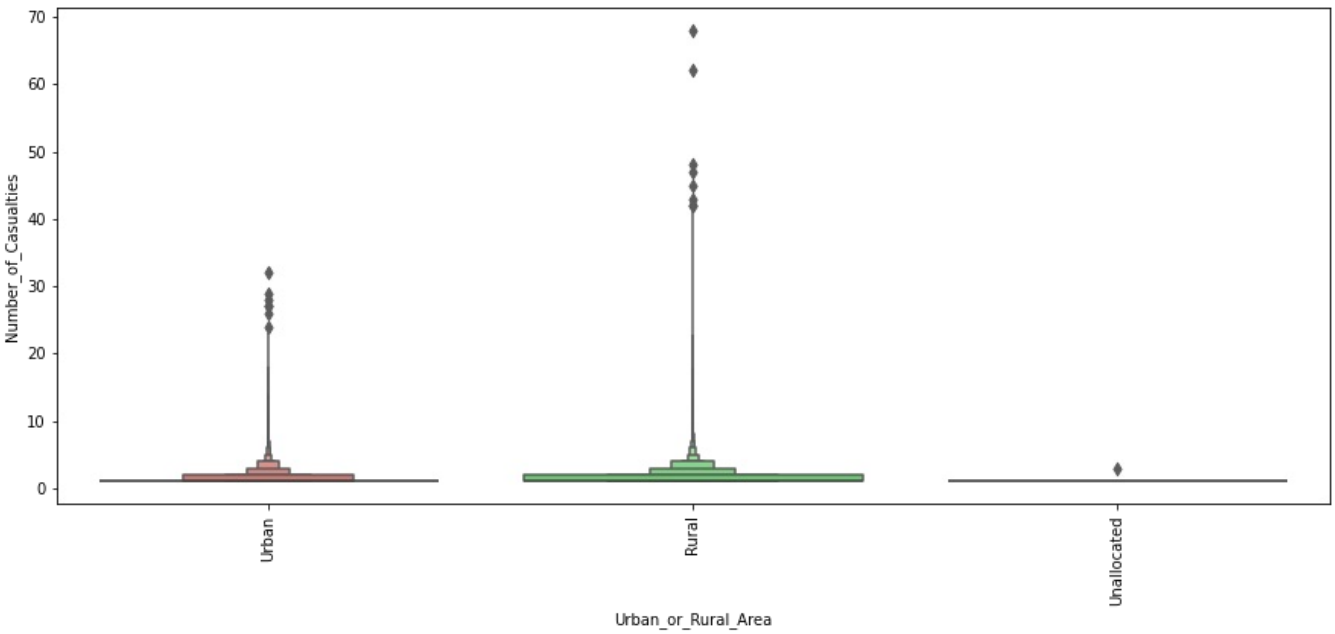

Longitude
2
0
−2
−4
−6
Single carriageway
Dual carriageway
One way street
Roundabout
Slip road
Road_Type

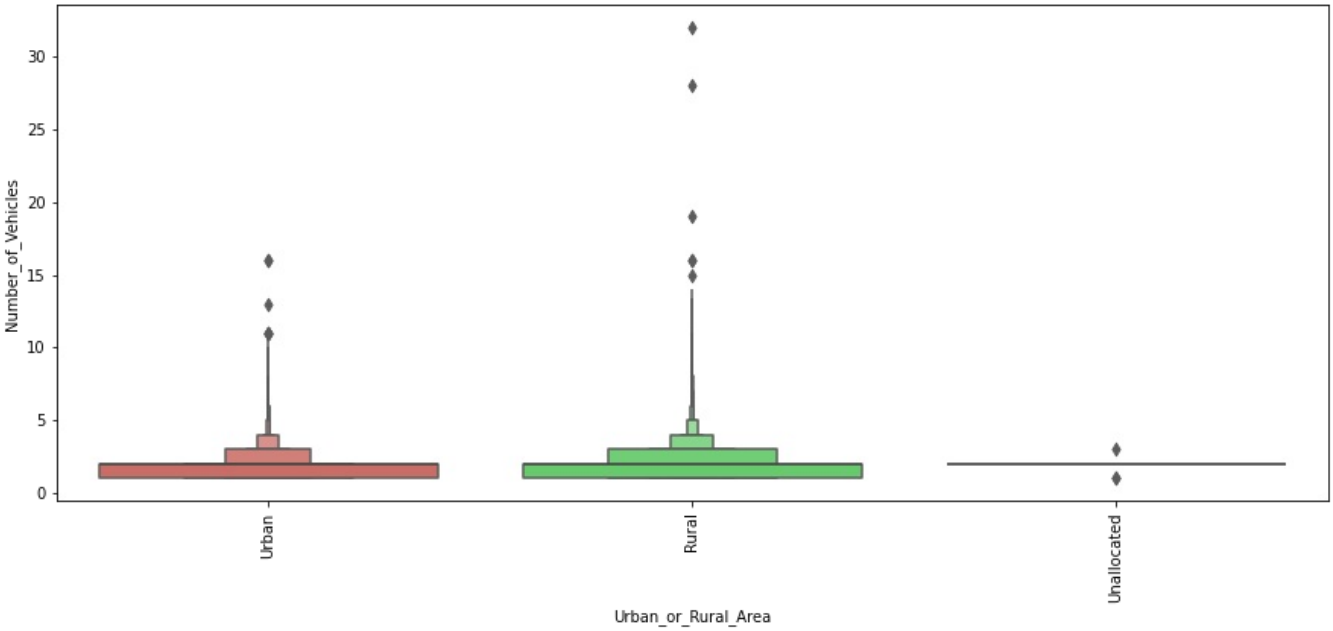

Number_of_Casualties
0
10
20
30
40
50
60
70
Single carriageway
Dual carriageway
One way street
Roundabout
Slip road
Road_Type

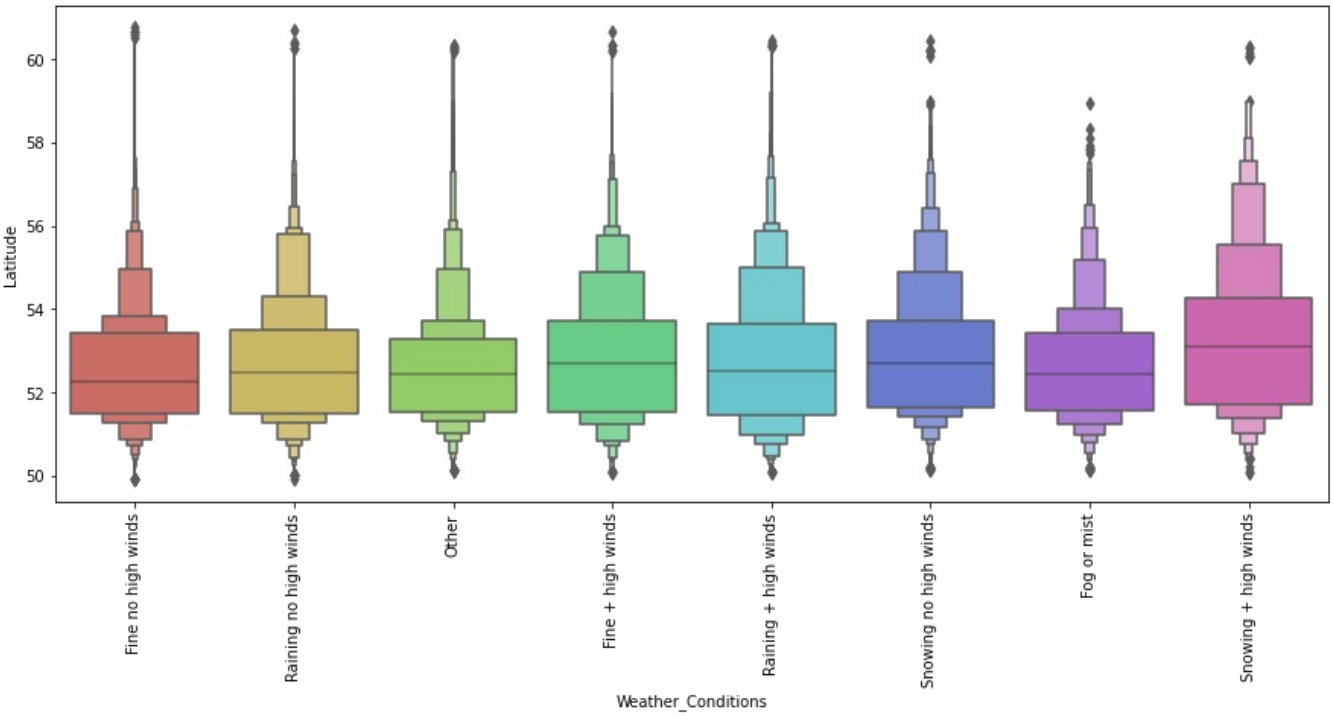

Number_of_Vehicles
0
5
10
15
20
25
30
Single carriageway
Dual carriageway
One way street
Roundabout
Slip road
Road_Type

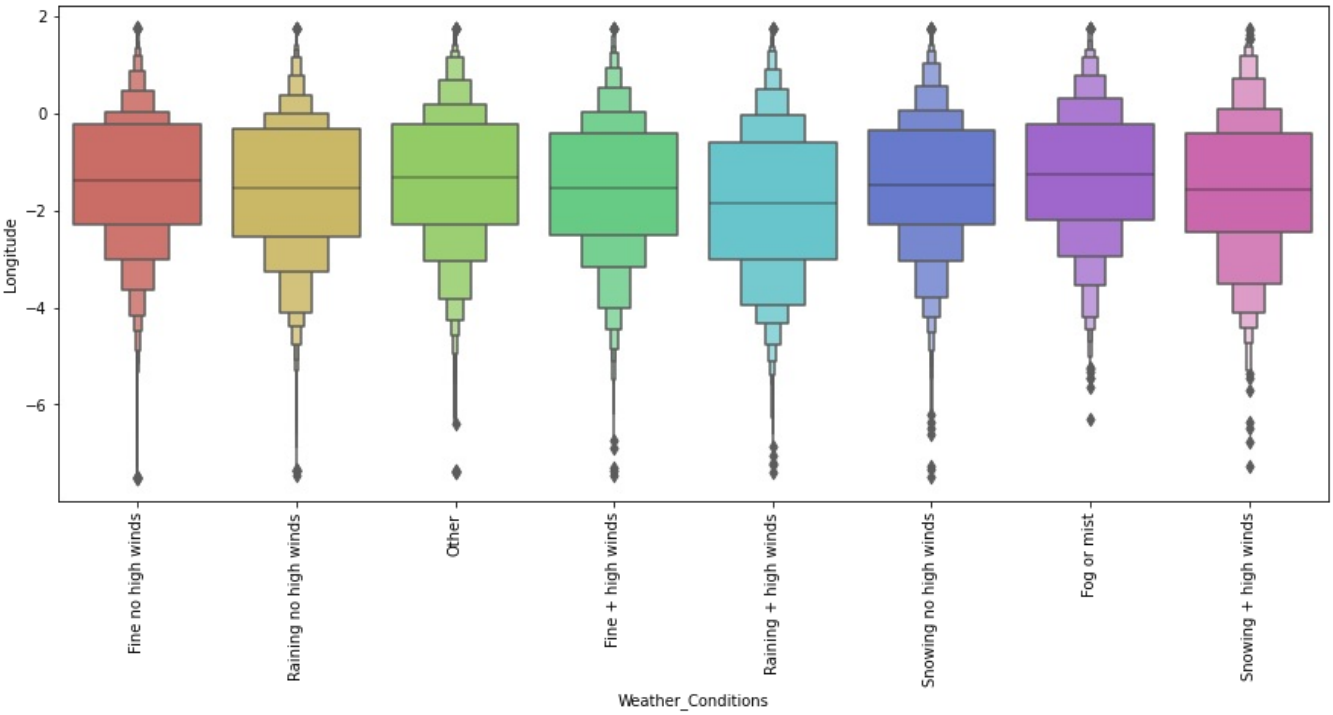

Latitude
60
58
56
54
52
50
Urban
Rural
Unallocated
Urban_or_Rural_Area

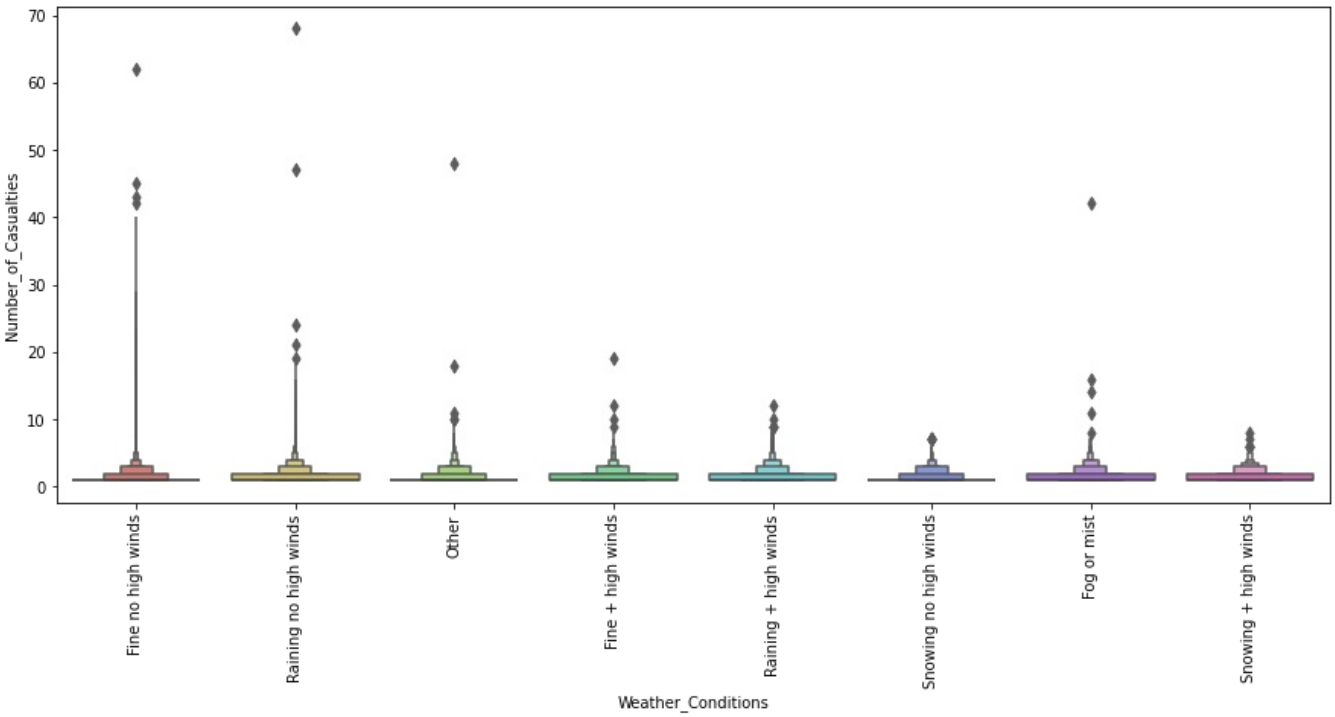

Longitude
2
0
−2
−4
−6
Urban
Rural
Unallocated
Urban_or_Rural_Area

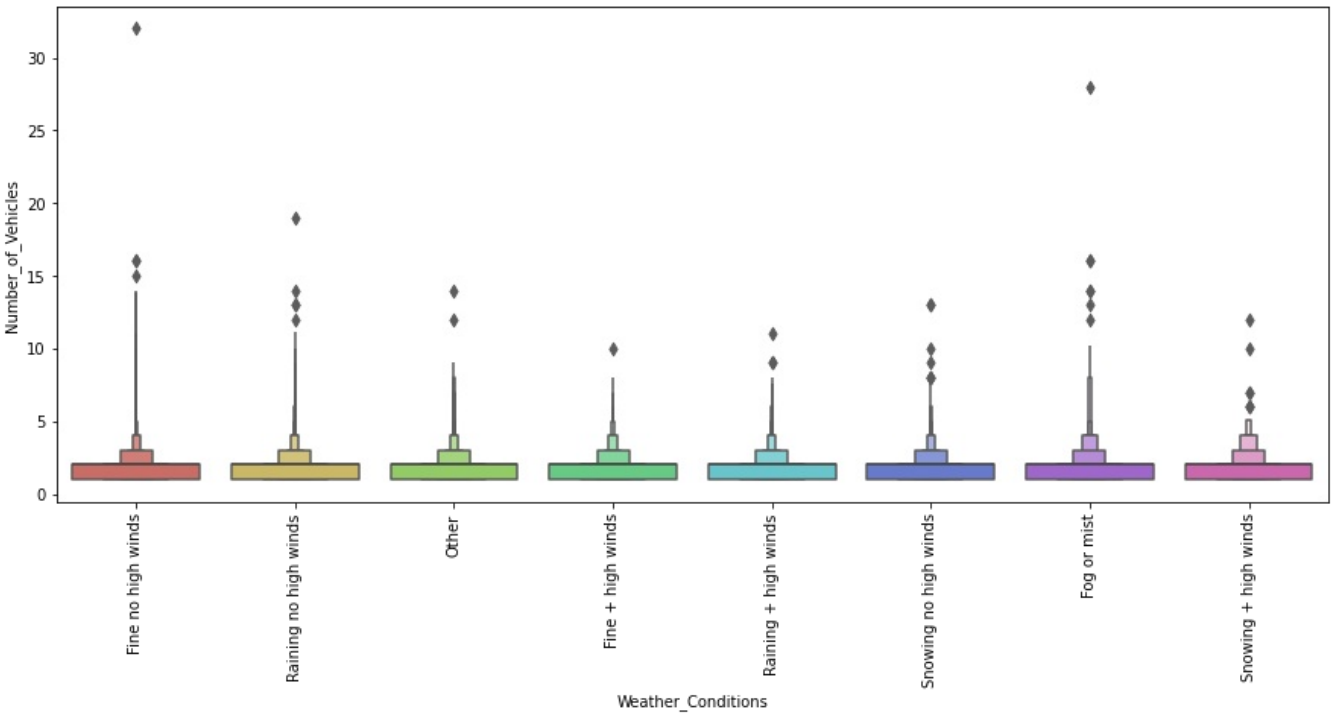

Number_of_Casualties
70
60
50
40
30
20
10
0
Urban
Rural
Unallocated
Urban_or_Rural_Area


Number_of_Vehicles
0
5
10
15
20
25
30
Urban
Rural
Unallocated
Urban_or_Rural_Area

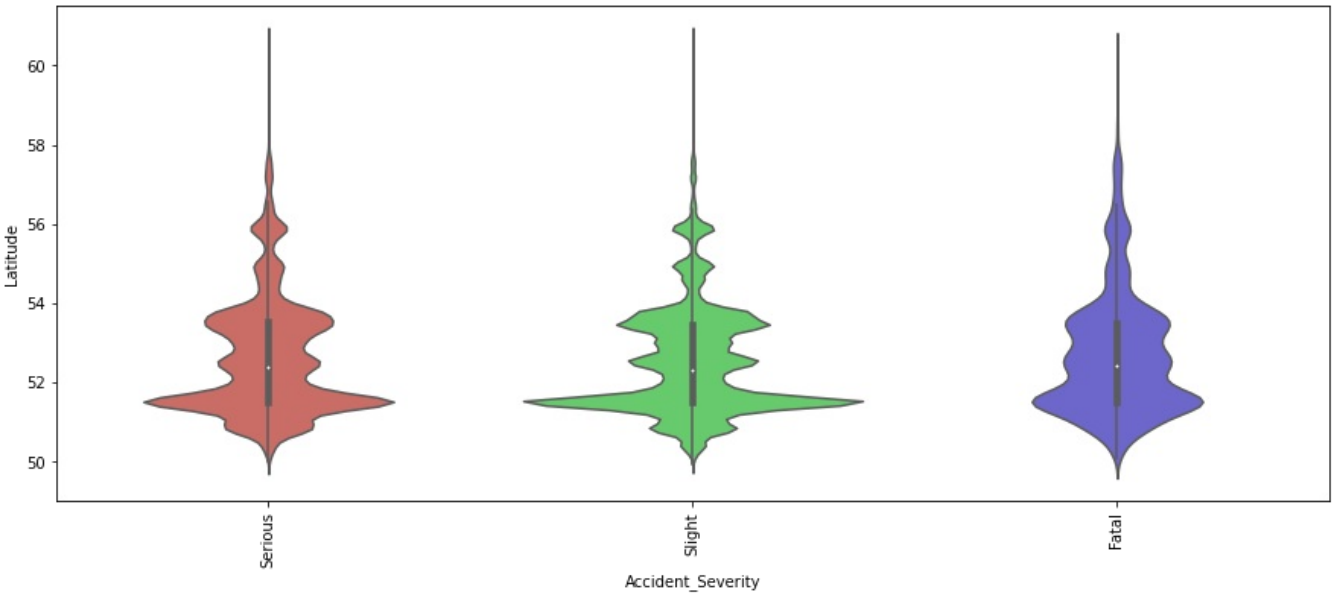

Latitude
50
52
54
56
58
60
Fine no high winds
Raining no high winds
Other
Fine + high winds
Raining + high winds
Snowing no high winds
Fog or mist
Snowing + high winds
Weather_Conditions

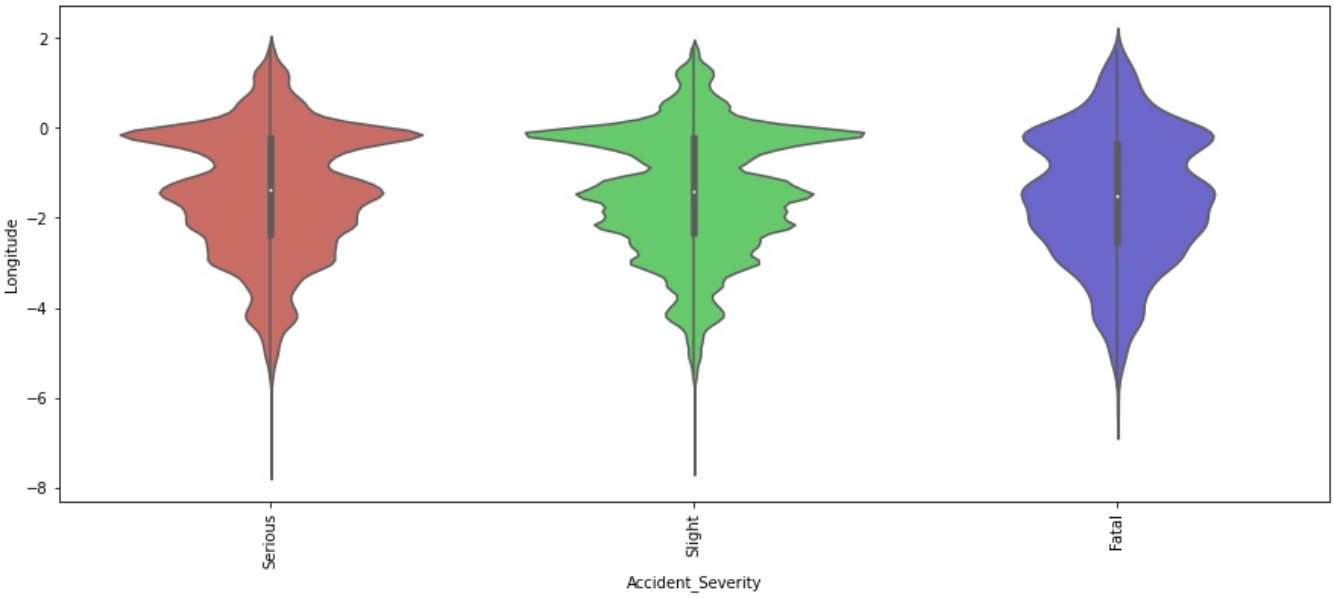

Longitude
−6
−4
−2
0
2
Fine no high winds
Raining no high winds
Other
Fine + high winds
Raining + high winds
Snowing no high winds
Fog or mist
Snowing + high winds
Weather_Conditions

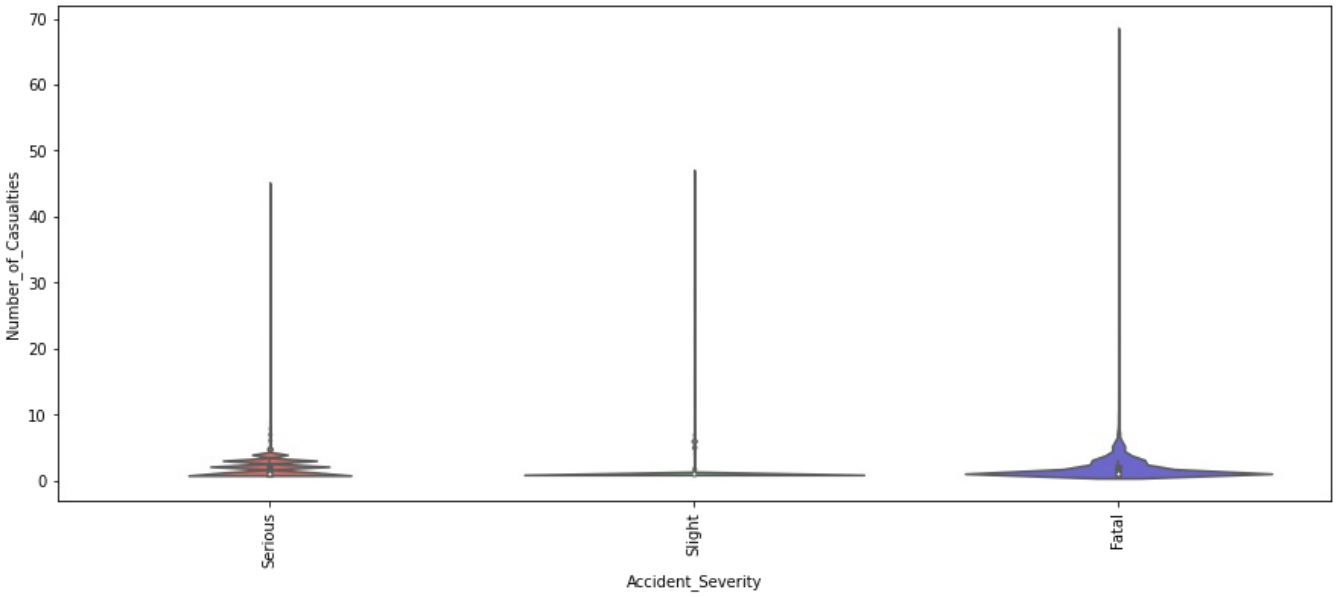


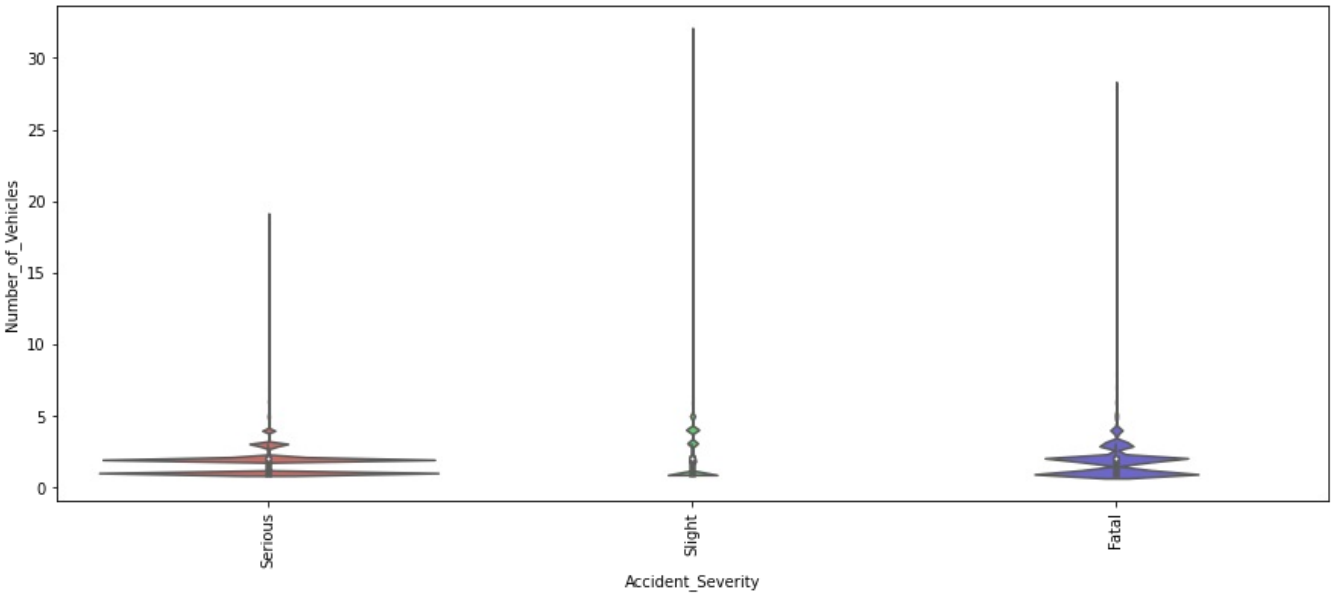


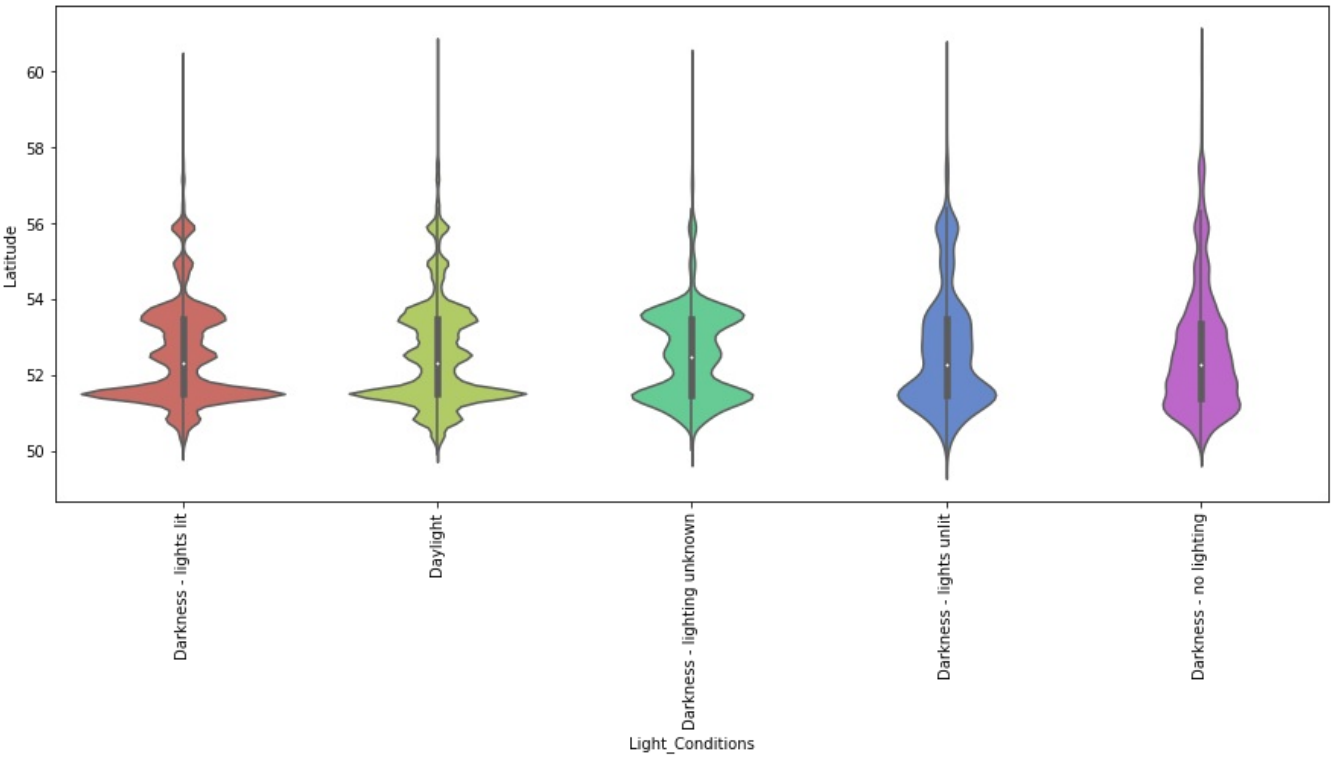
In [44]:
```
for i in categorical:
    for j in continuous:
        plt.figure(figsize=(15,6))
        sns.violinplot(x = df[i], y = df[j], data = df, palette = 'hls')
        plt.xticks(rotation = 90)
        plt.show()
```

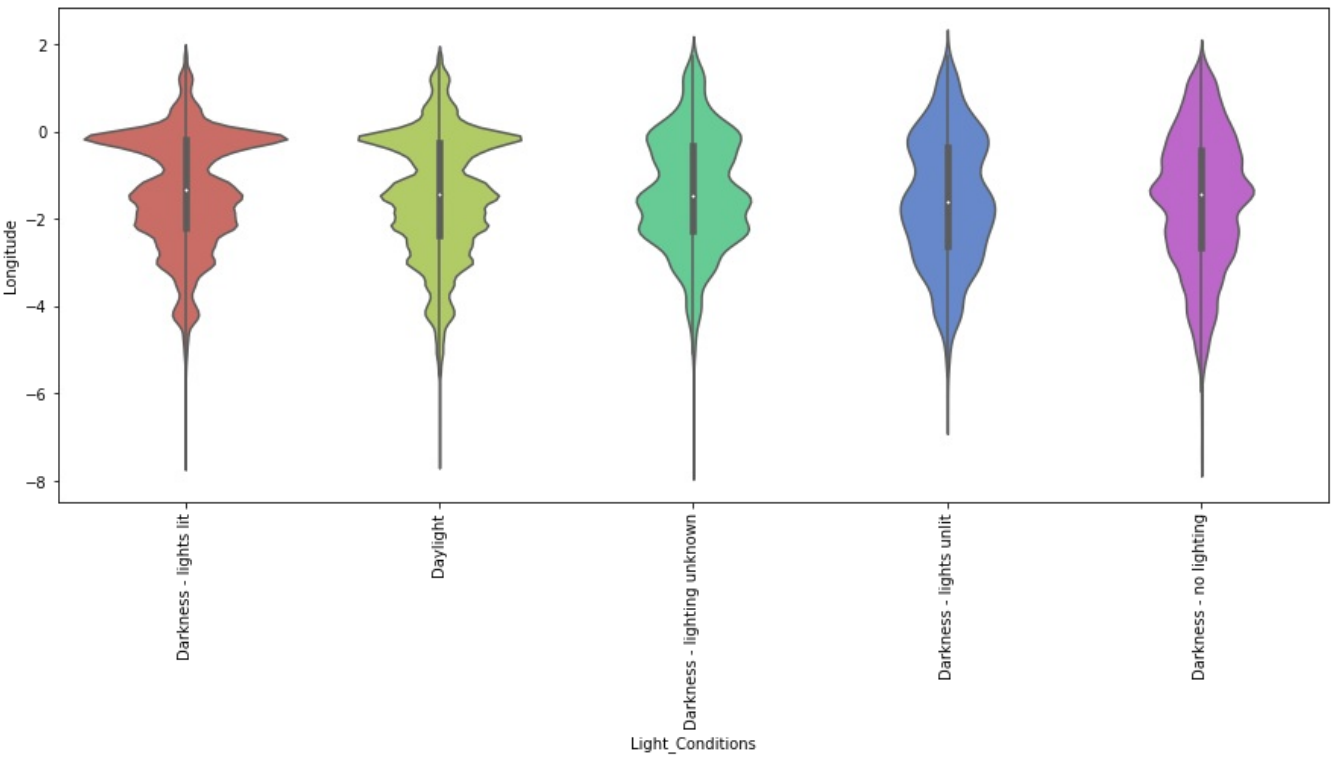

Latitude
60
58
56
54
52
50
Serious
Slight
Fatal
Accident_Severity

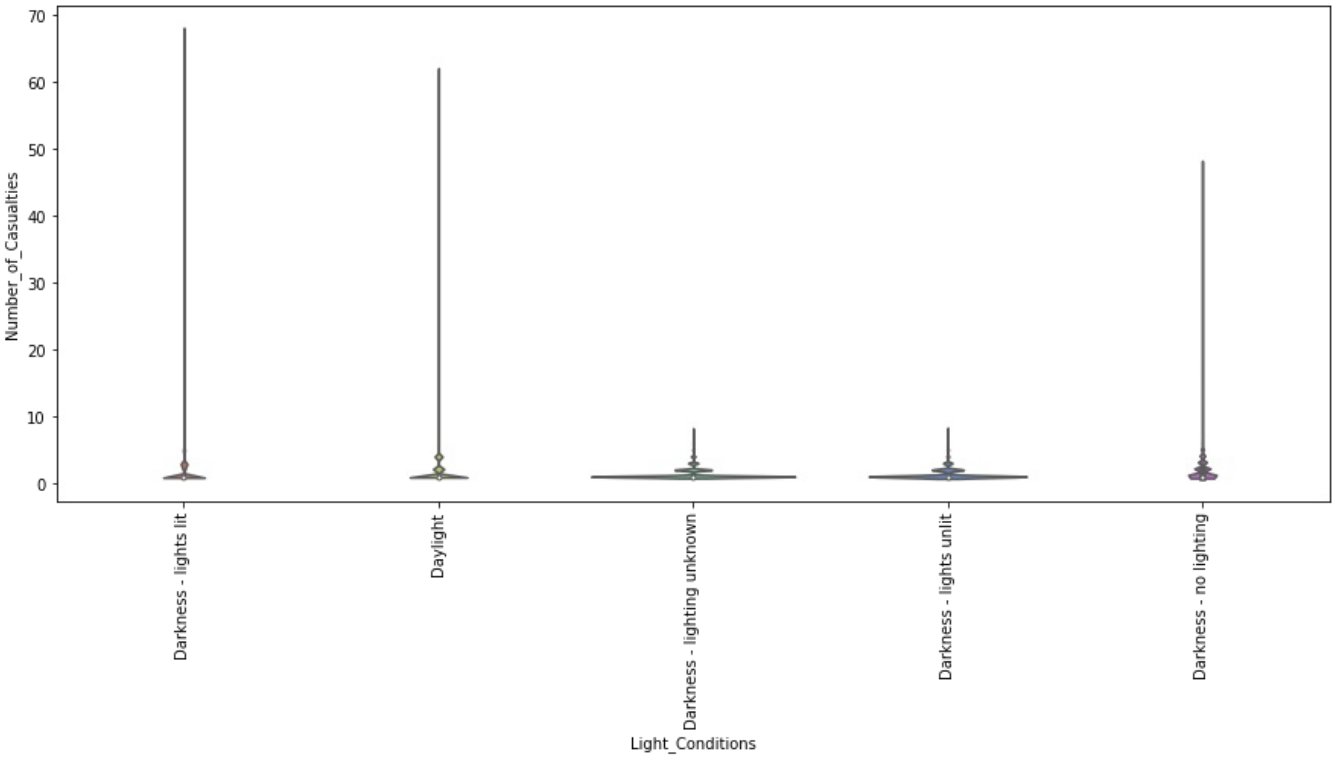

Longitude
2
0
−2
−4
−6
−8
Serious
Slight
Fatal
Accident_Severity

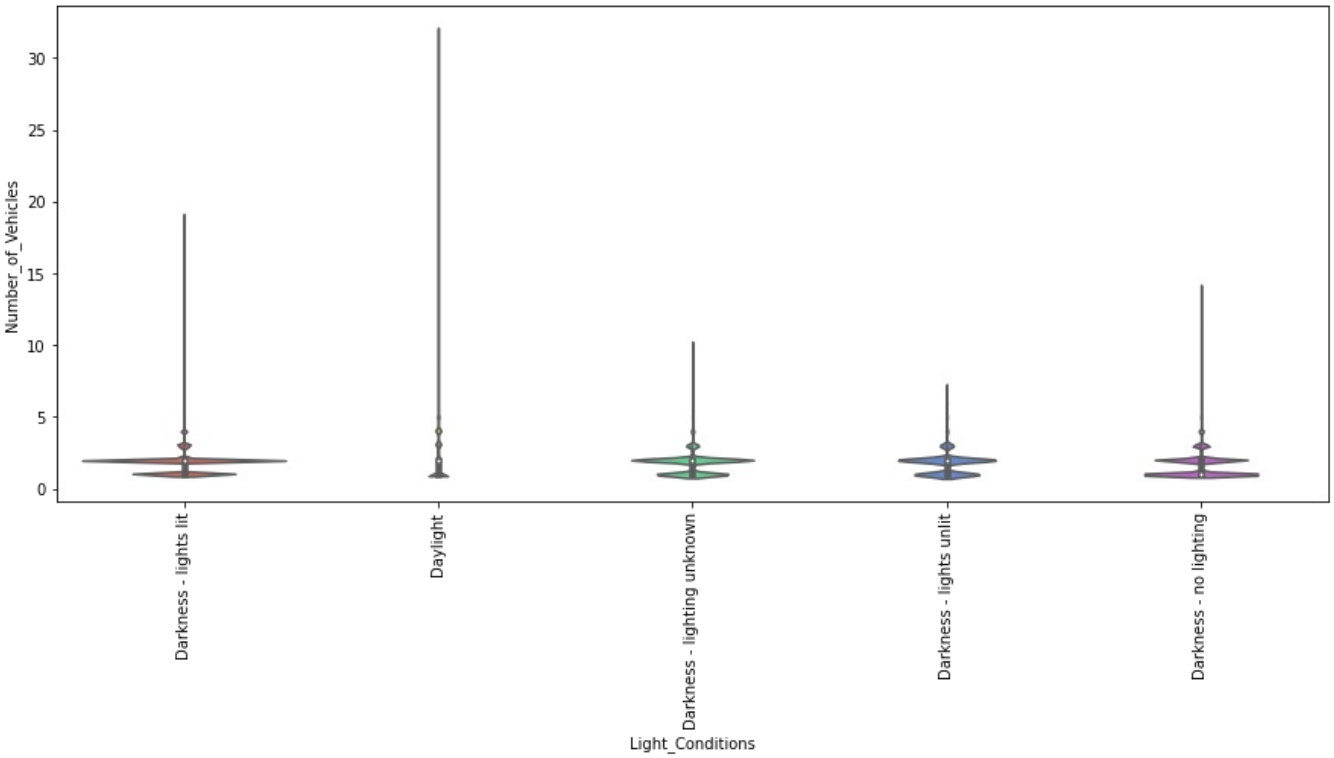

Number_of_Casualties
70
60
50
40
30
20
10
0
Serious
Slight
Fatal
Accident_Severity

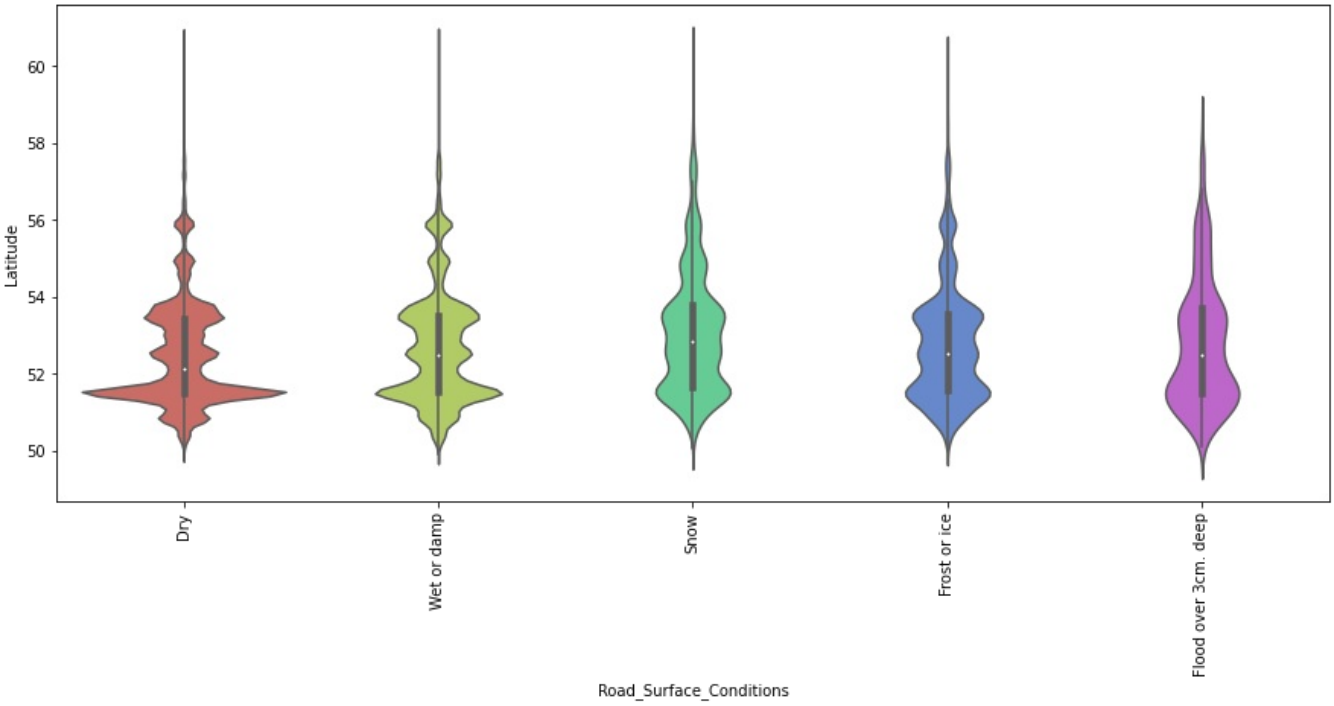

Number_of_Vehicles
0
5
10
15
20
25
30
Serious
Slight
Fatal
Accident_Severity

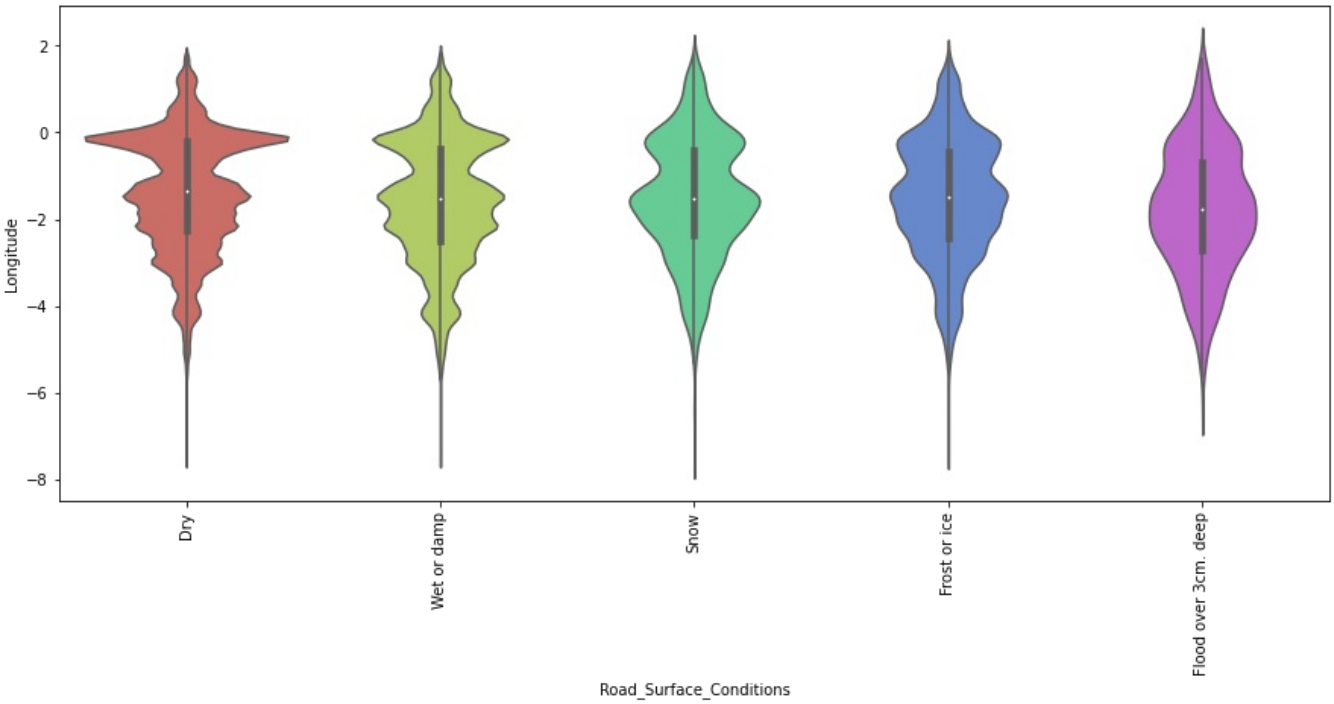

Latitude
50
52
54
56
58
60
Darkness - lights lit
Daylight
Darkness - lighting unknown
Darkness - lights unlit
Darkness - no lighting
Light_Conditions

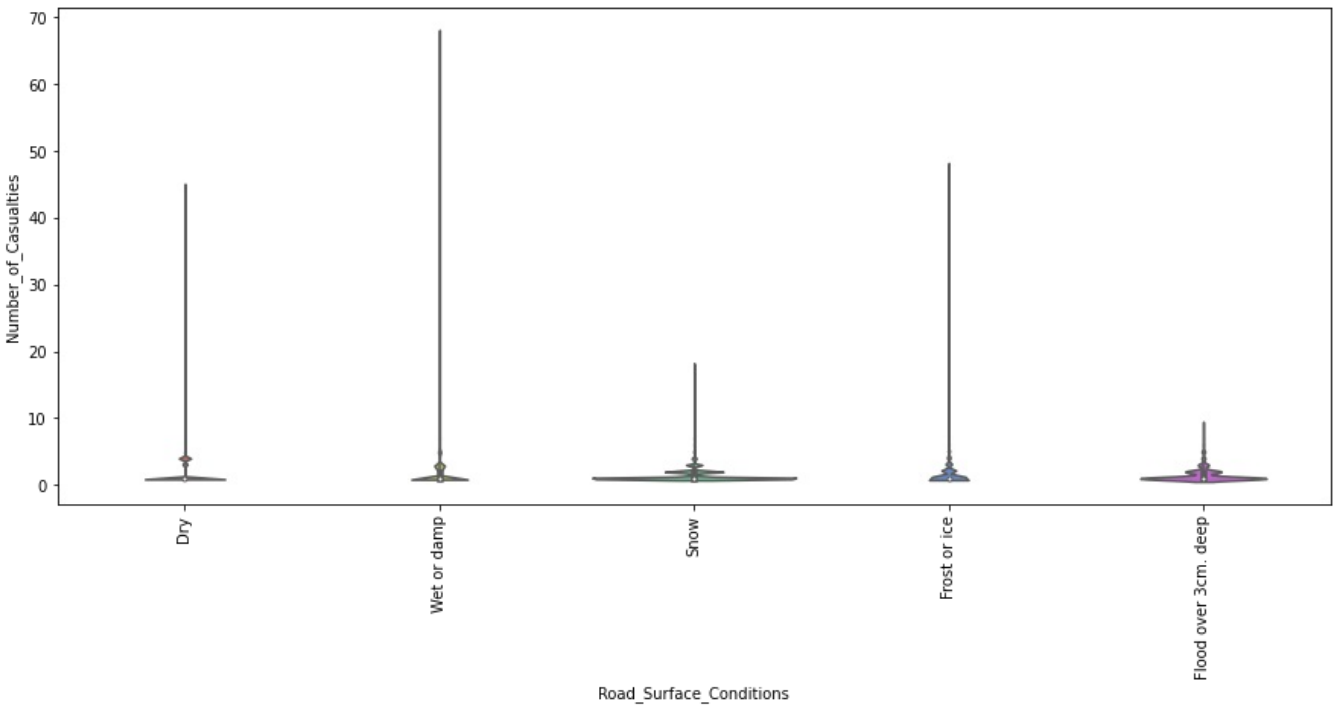

Longitude
−8
−6
−4
−2
0
2
Darkness - lights lit
Daylight
Darkness - lighting unknown
Darkness - lights unlit
Darkness - no lighting
Light_Conditions

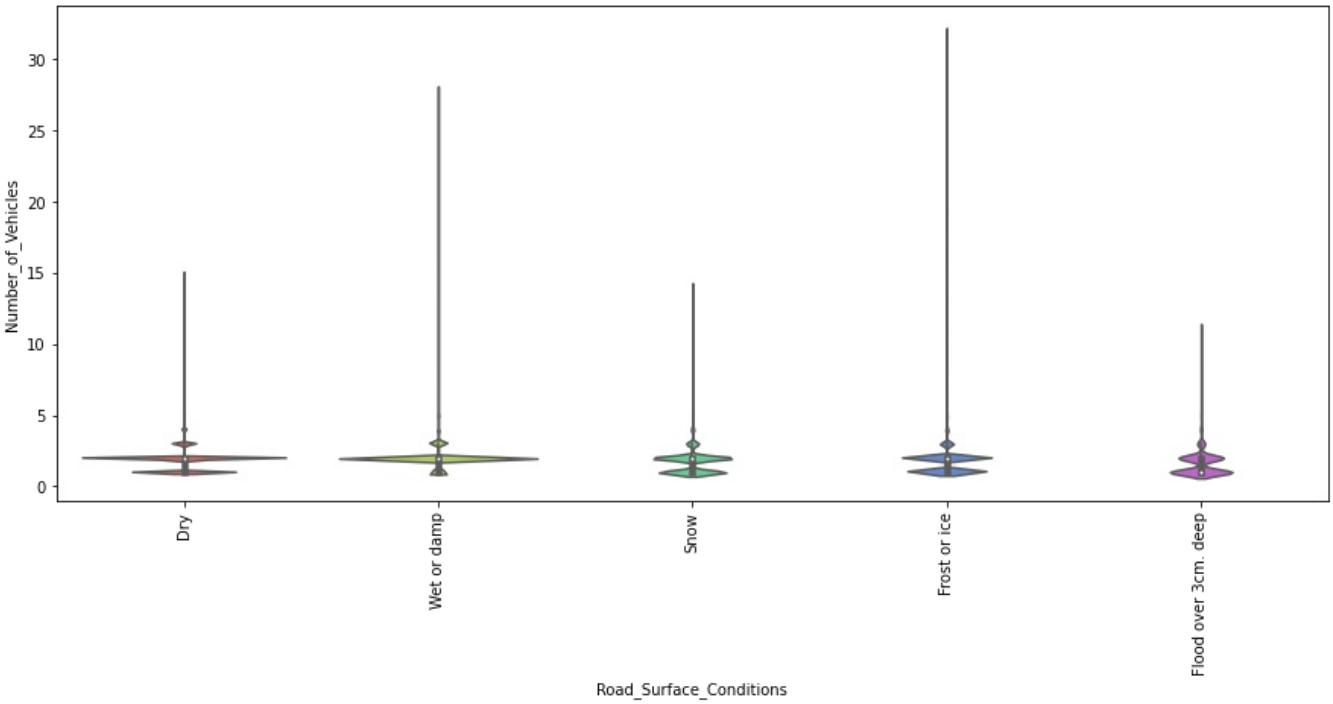

Number_of_Casualties
70
60
50
40
30
20
10
0
Darkness - lights lit
Daylight
Darkness - lighting unknown
Darkness - lights unlit
Darkness - no lighting
Light_Conditions

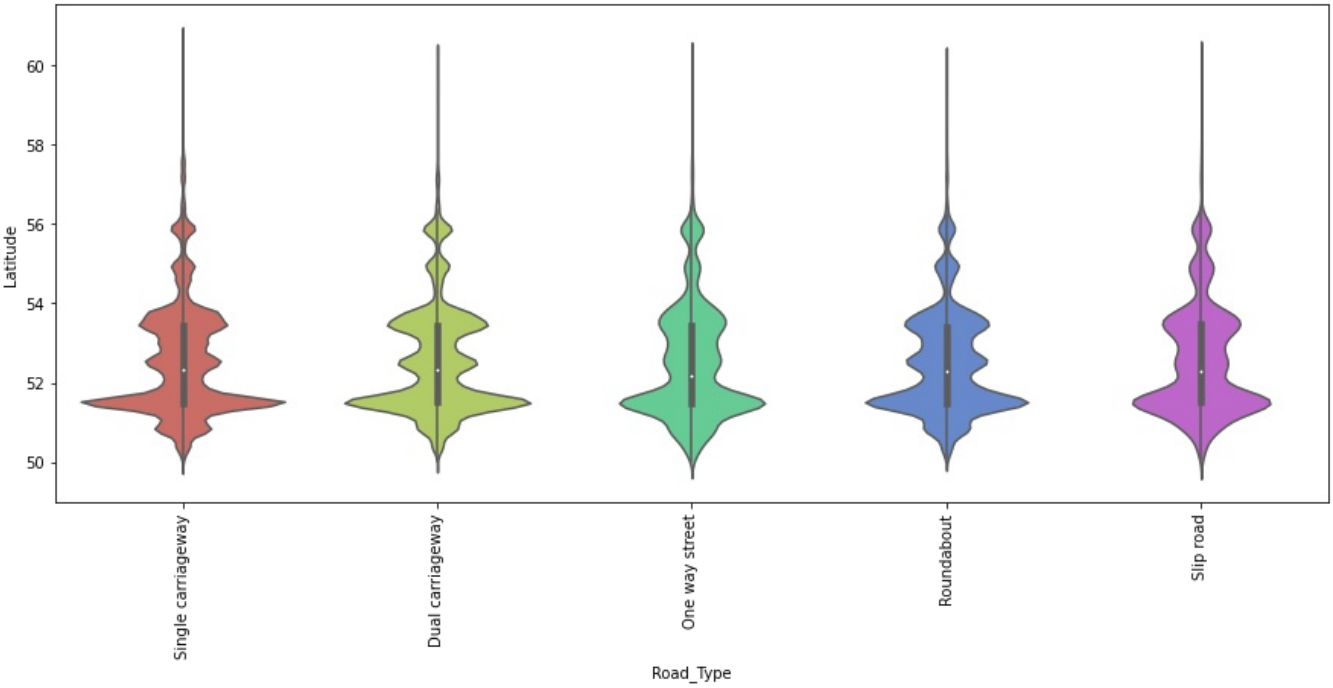

Number_of_Vehicles
30
25
20
15
10
5
0
Darkness - lights lit
Daylight
Darkness - lighting unknown
Darkness - lights unlit
Darkness - no lighting
Light_Conditions

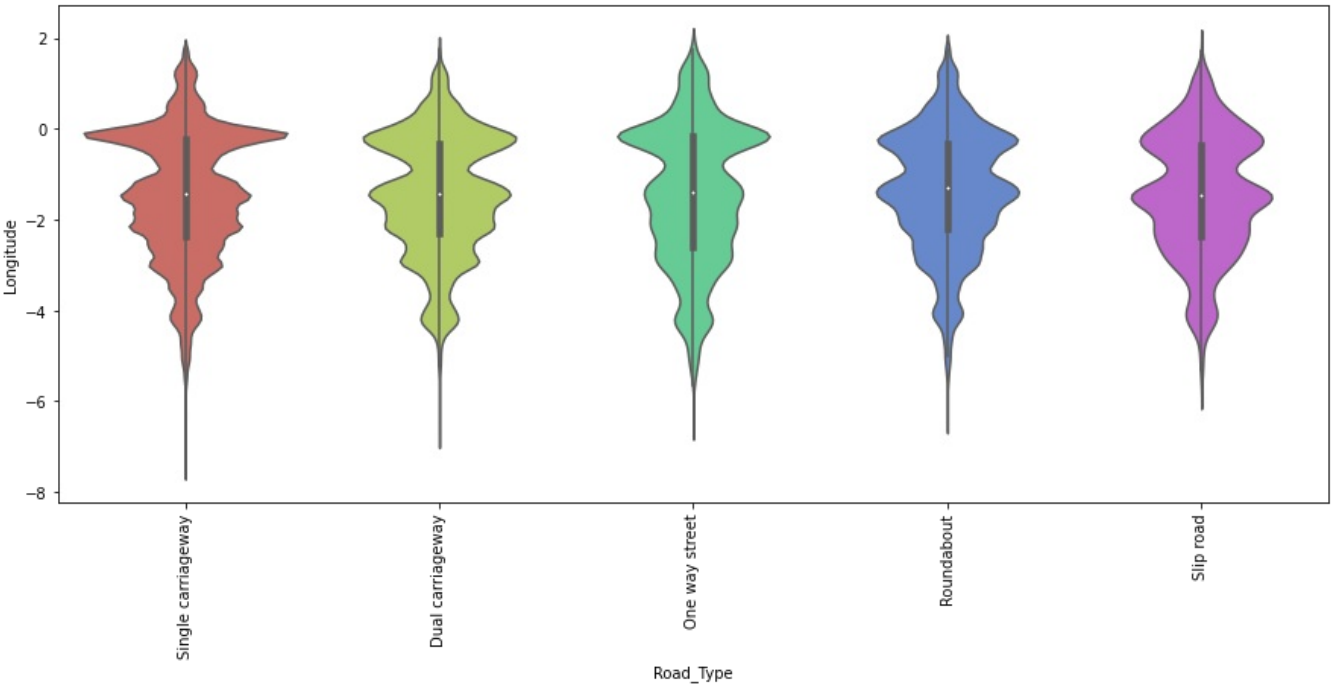

Latitude
60
58
56
54
52
50
Dry
Wet or damp
Snow
Frost or ice
Flood over 3cm. deep
Road_Surface_Conditions

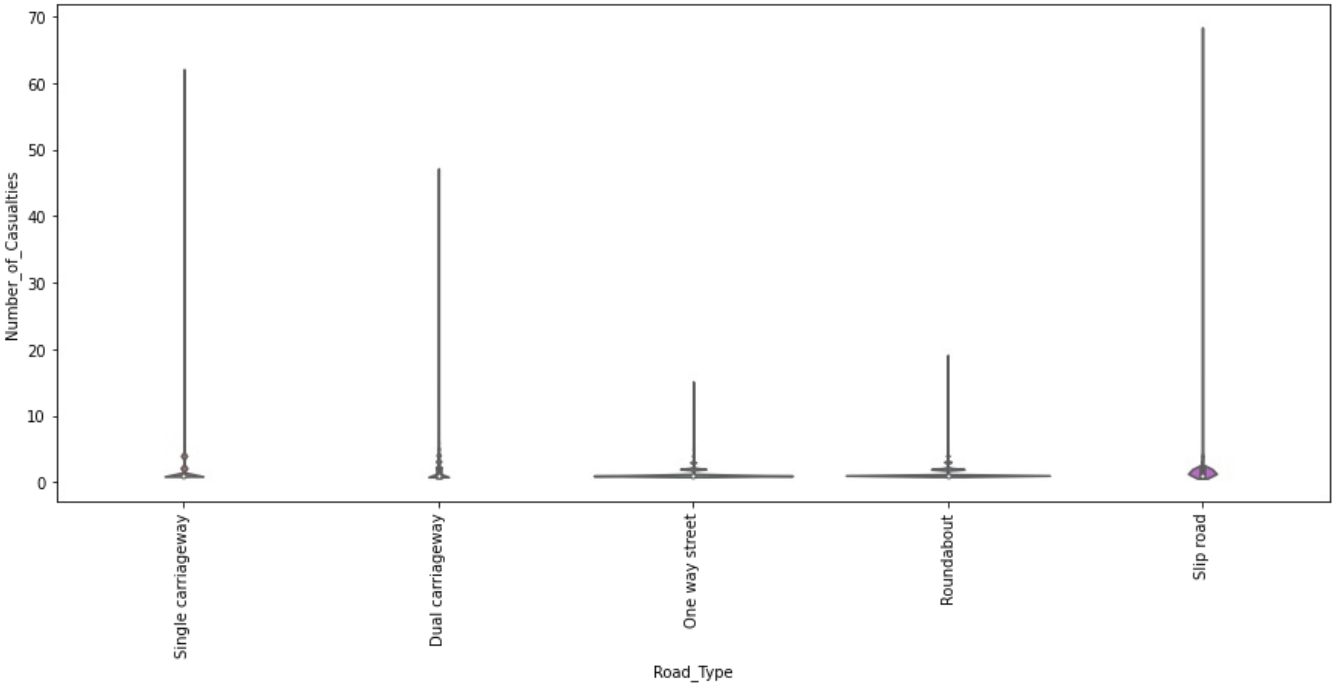

Longitude
2
0
−2
−4
−6
−8
Dry
Wet or damp
Snow
Frost or ice
Flood over 3cm. deep
Road_Surface_Conditions

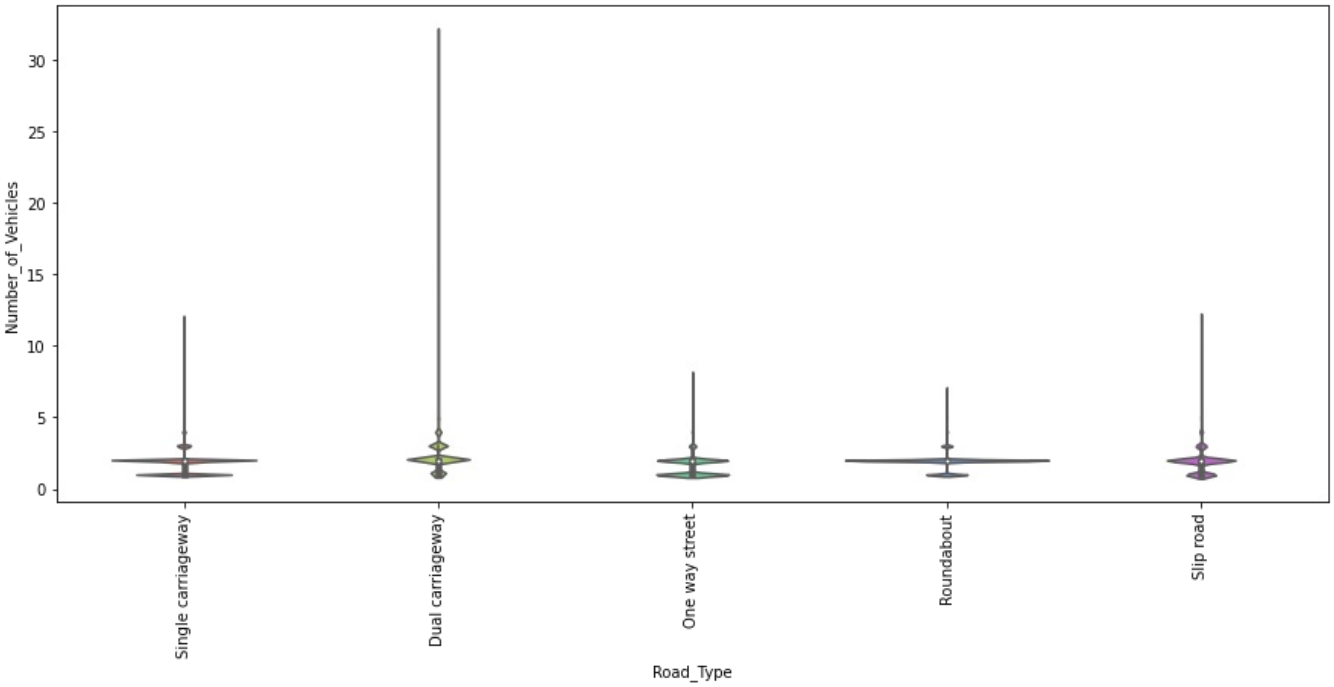

Number_of_Casualties
70
60
50
40
30
20
10
0
Dry
Wet or damp
Snow
Frost or ice
Flood over 3cm. deep
Road_Surface_Conditions

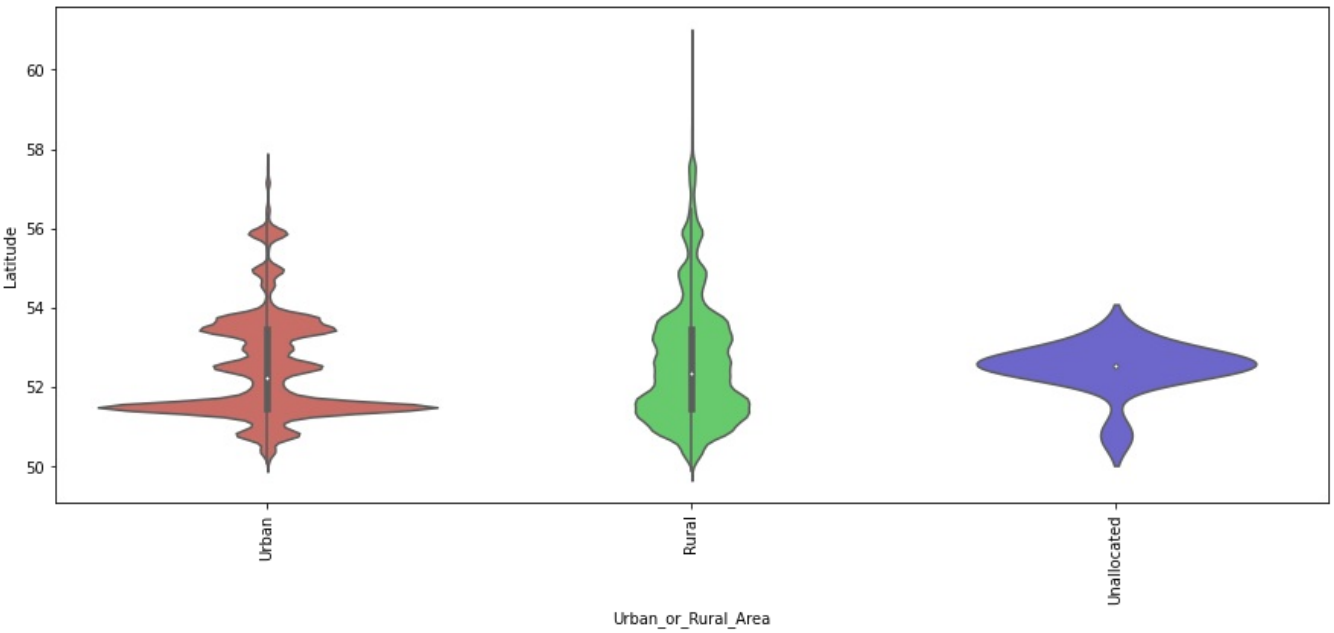

Number_of_Vehicles
0
5
10
15
20
25
30
Dry
Wet or damp
Snow
Frost or ice
Flood over 3cm. deep
Road_Surface_Conditions

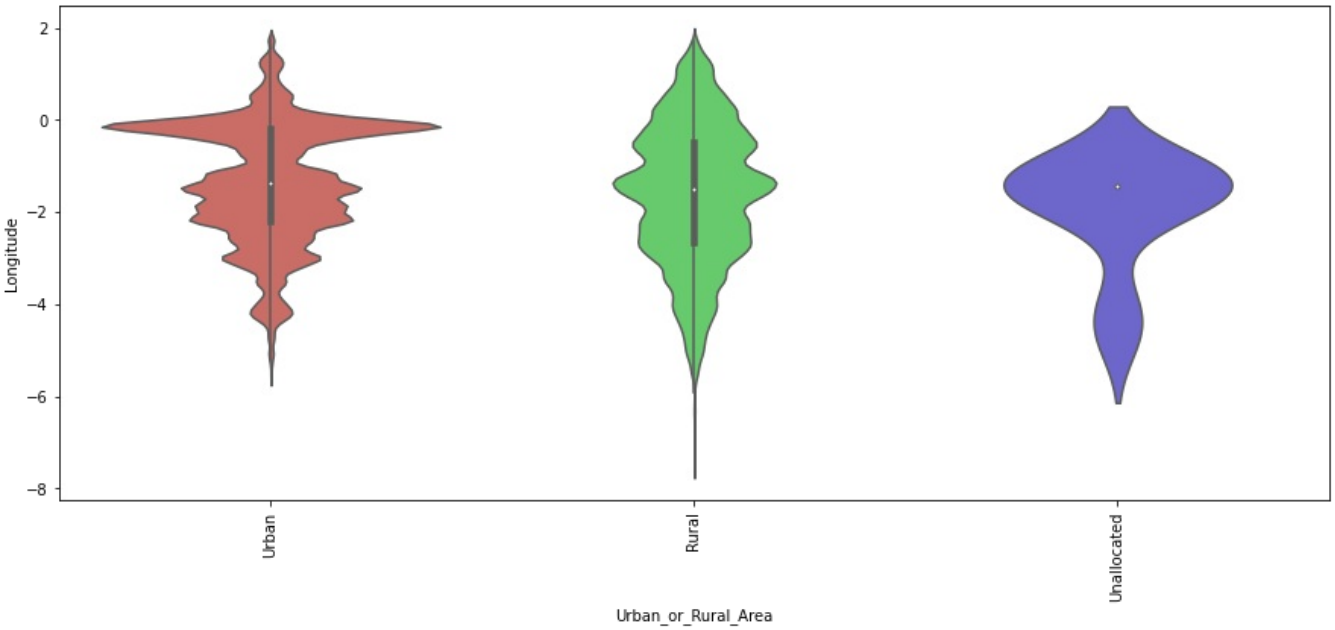

Latitude
50
52
54
56
58
60
Single carriageway
Dual carriageway
One way street
Roundabout
Slip road
Road_Type

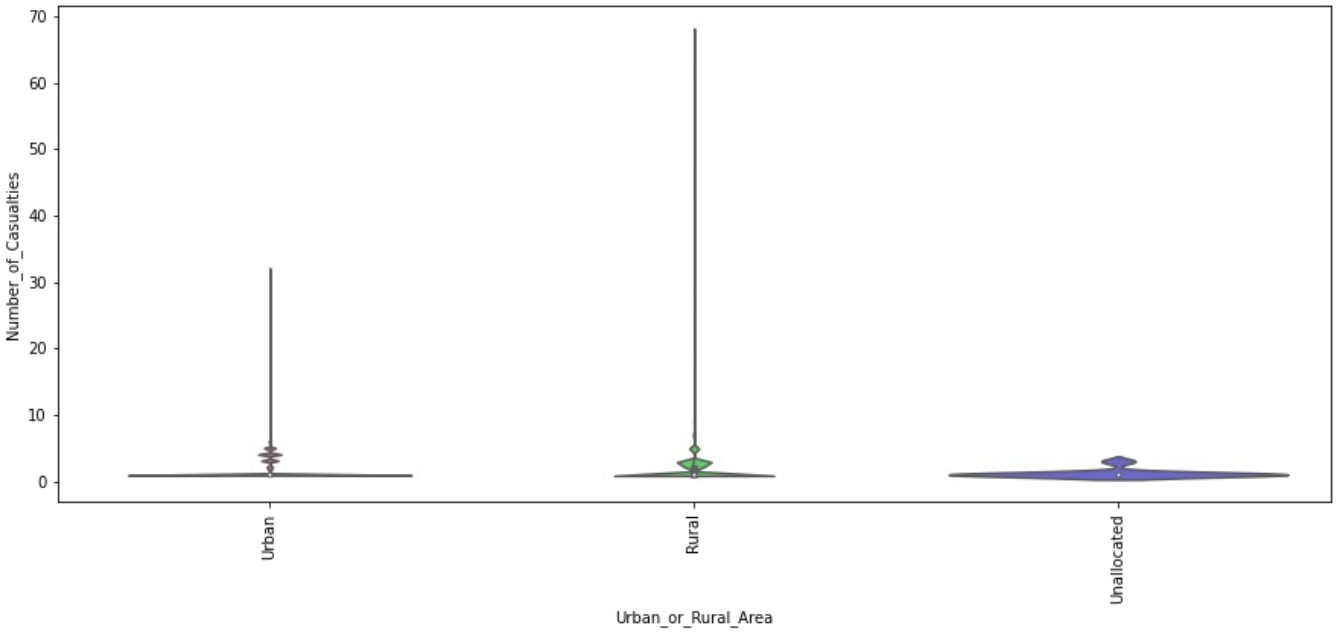

Longitude
−8
−6
−4
−2
0
2
Single carriageway
Dual carriageway
One way street
Roundabout
Slip road
Road_Type

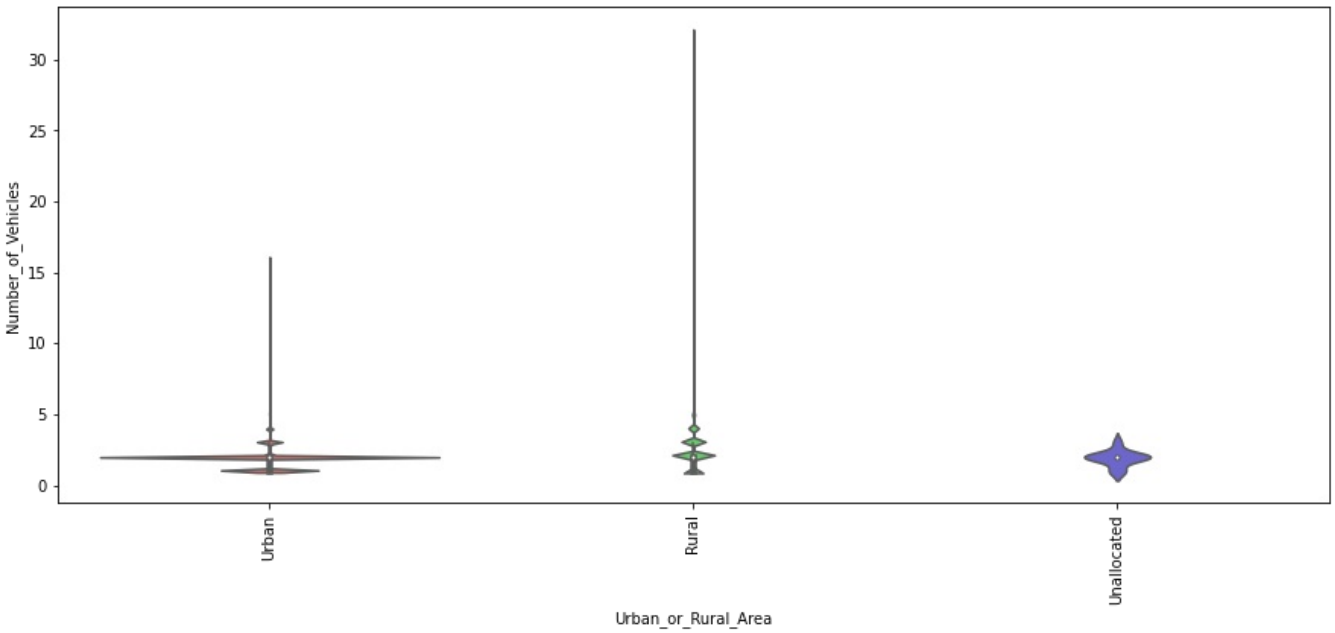

Number_of_Casualties
0
10
20
30
40
50
60
70
Single carriageway
Dual carriageway
One way street
Roundabout
Slip road
Road_Type

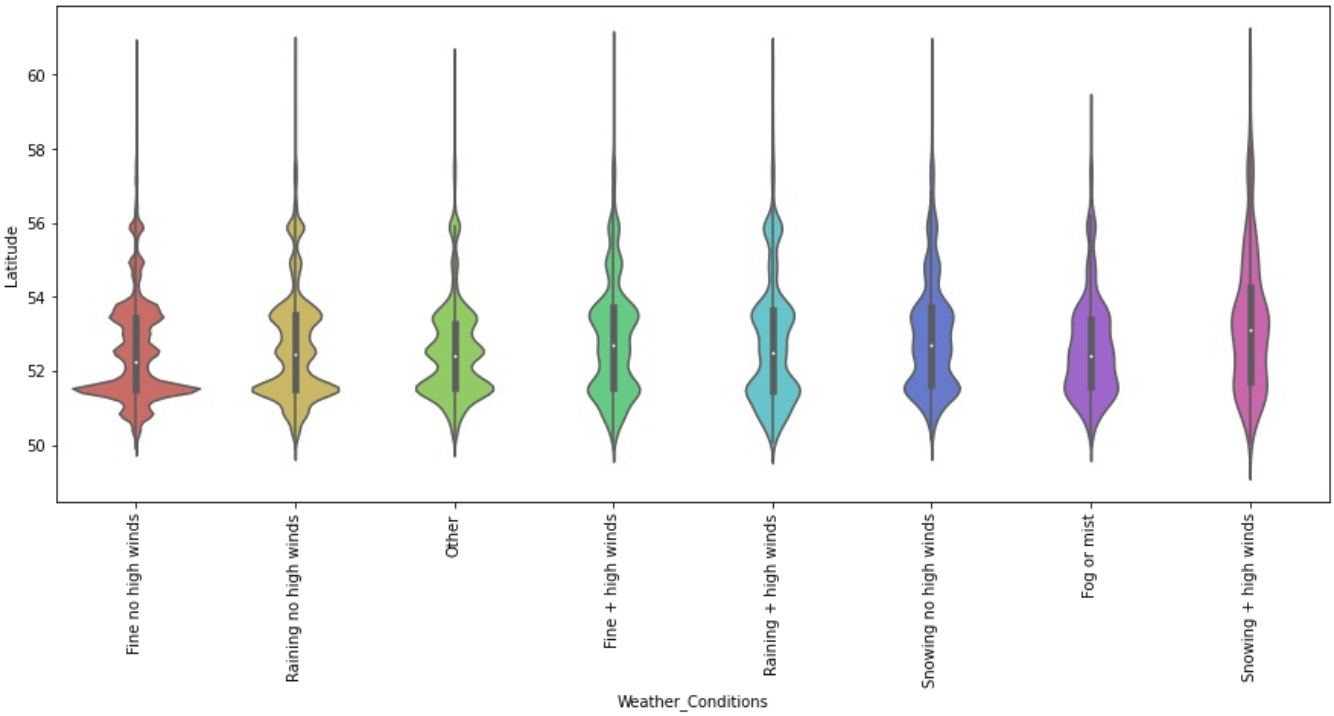

Number_of_Vehicles
0
5
10
15
20
25
30
Single carriageway
Dual carriageway
One way street
Roundabout
Slip road
Road_Type

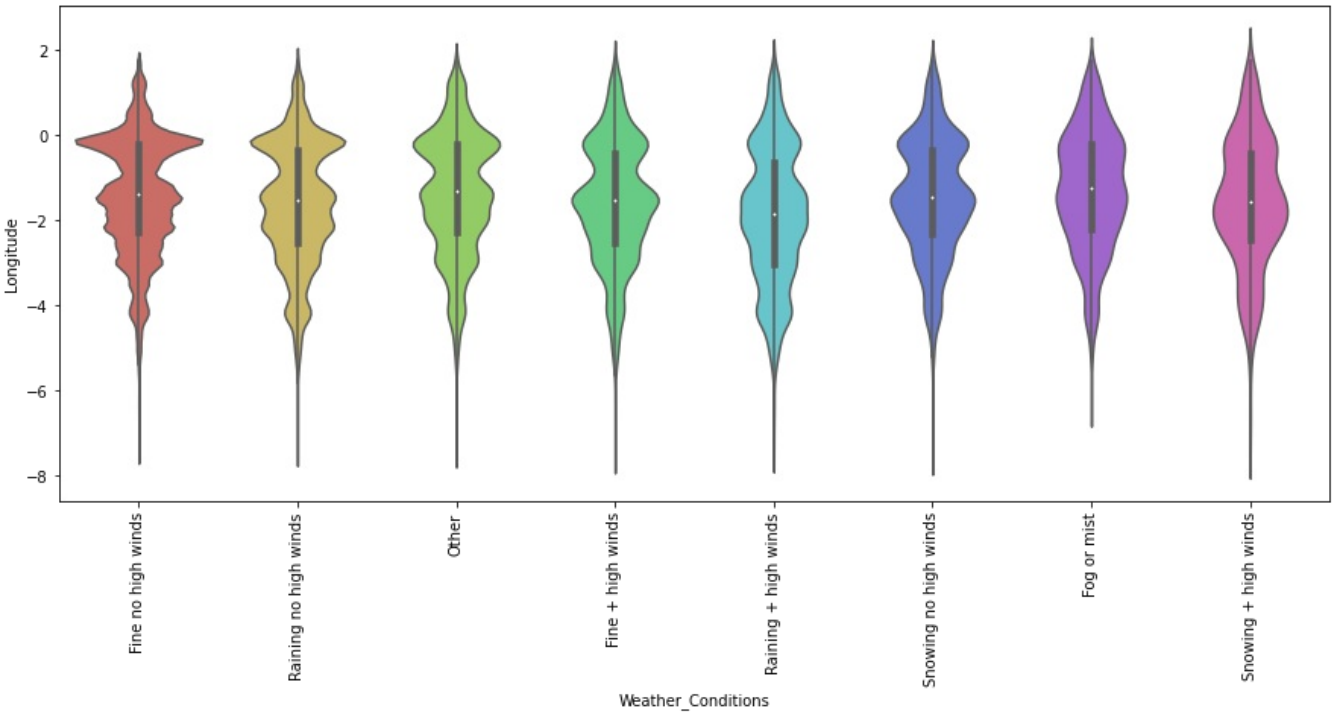

Latitude
50
52
54
56
58
60
Urban
Rural
Unallocated
Urban_or_Rural_Area

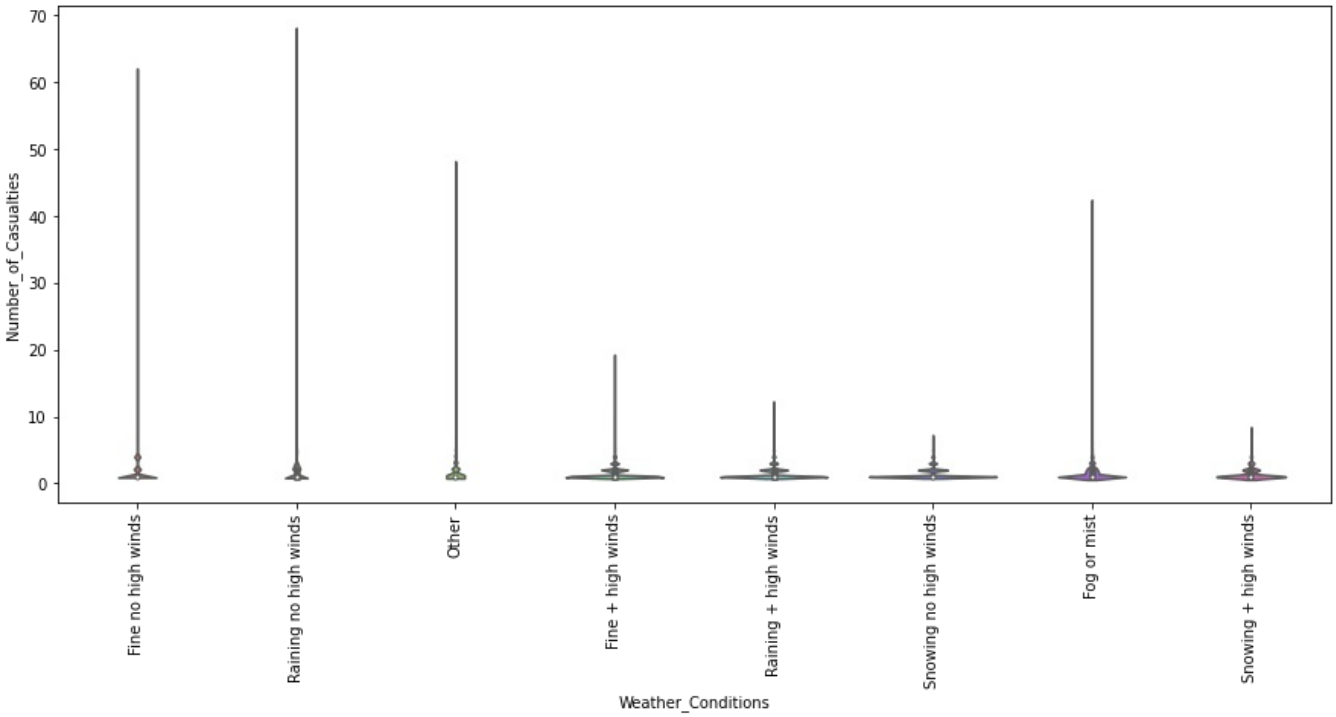

Longitude
2
0
−2
−4
−6
−8
Urban
Rural
Unallocated
Urban_or_Rural_Area

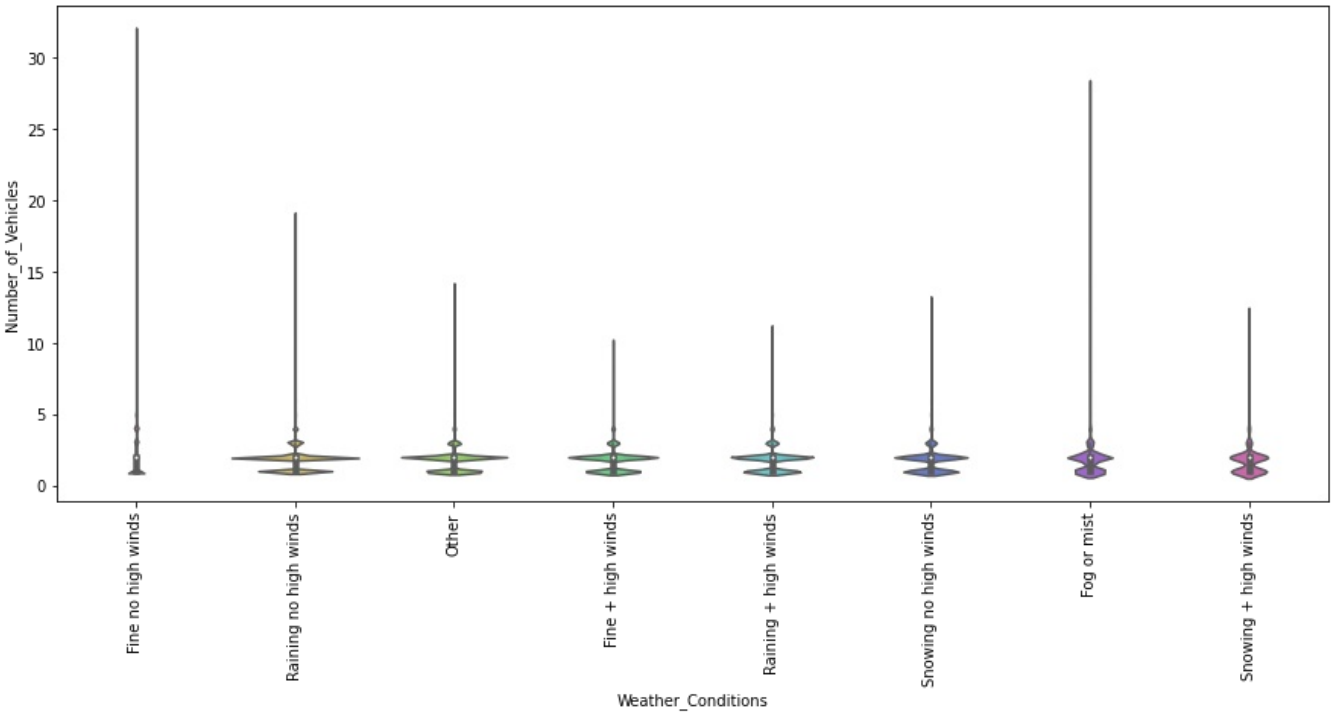

Number_of_Casualties
70
60
50
40
30
20
10
0
Urban
Rural
Unallocated
Urban_or_Rural_Area


Number_of_Vehicles
30
25
20
15
10
5
0
Urban
Rural
Unallocated
Urban_or_Rural_Area


Latitude
60
58
56
54
52
50
Fine no high winds
Raining no high winds
Other
Fine + high winds
Raining + high winds
Snowing no high winds
Fog or mist
Snowing + high winds
Weather_Conditions


Longitude
2
0
−2
−4
−6
−8
Fine no high winds
Raining no high winds
Other
Fine + high winds
Raining + high winds
Snowing no high winds
Fog or mist
Snowing + high winds
Weather_Conditions


Number_of_Casualties
70
60
50
40
30
20
10
0
Fine no high winds
Raining no high winds
Other
Fine + high winds
Raining + high winds
Snowing no high winds
Fog or mist
Snowing + high winds
Weather_Conditions



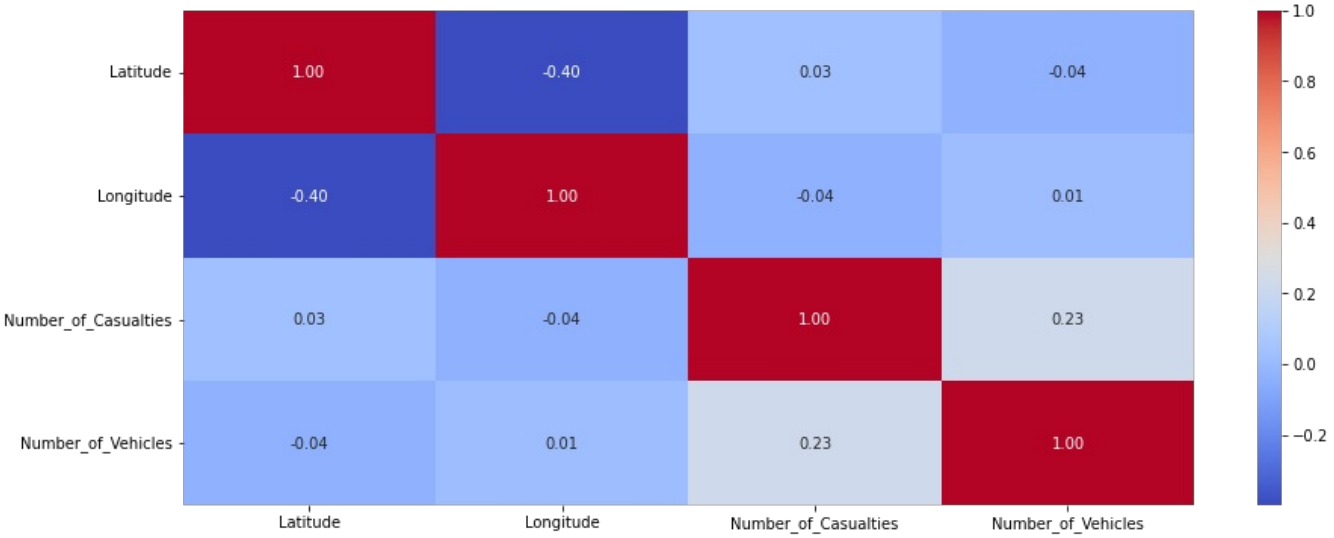
In [45]:
```
correlation_matrix = df.corr()
```

In [46]:
```
correlation_matrix
```

Out[46]:

	Latitude	Longitude	Number_of_Casualties	Number_of_Vehicles
Latitude	1.000000	-0.398116	0.032198	-0.040028
Longitude	-0.398116	1.000000	-0.040404	0.014716
Number_of_Casualties	0.032198	-0.040404	1.000000	0.228889
Number_of_Vehicles	-0.040028	0.014716	0.228889	1.000000

In [47]:
```
plt.figure(figsize=(15, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.show()
```



In [48]:
```
df_new = df.copy()
```

In [49]:
```
df_new['Accident Date'] = pd.to_datetime(df_new['Accident Date'])
```

In [50]:
```
df_new.set_index('Accident Date', inplace=True)
```

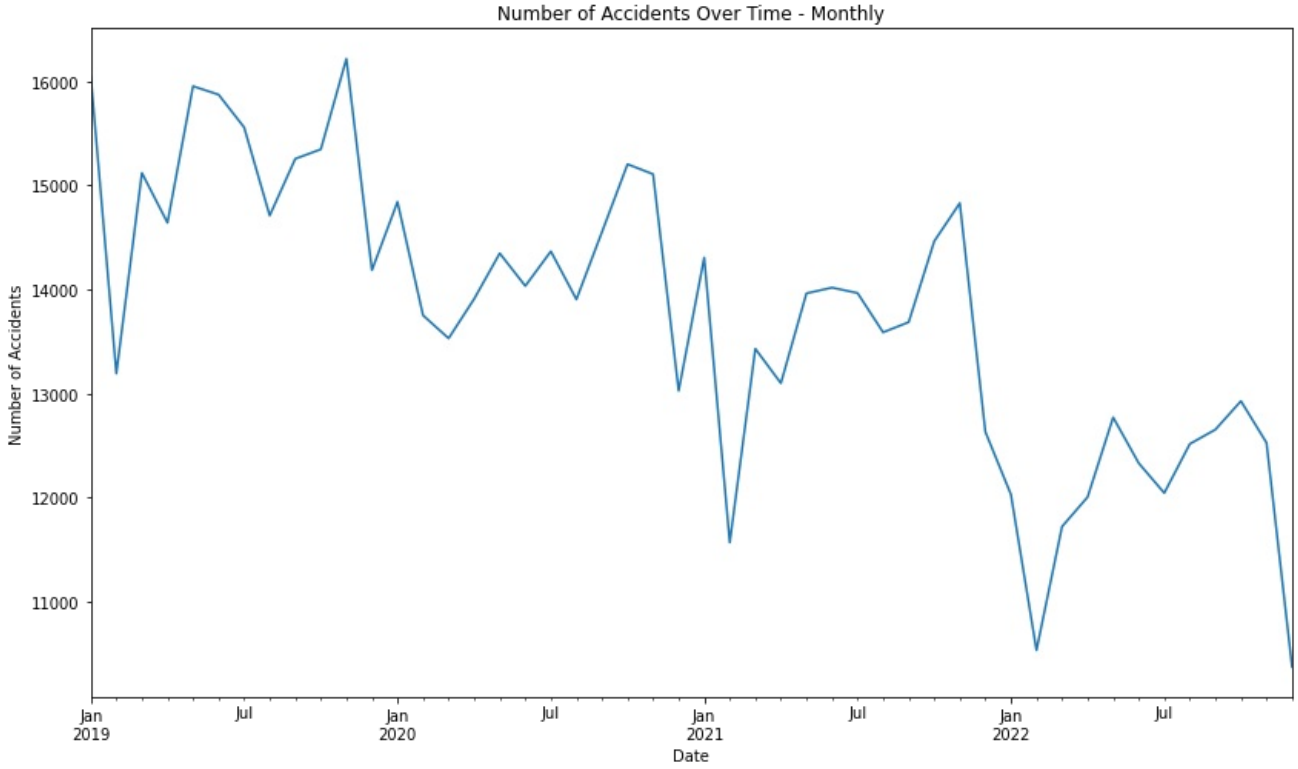
In [51]:
```
df_new
```

Out[51]:

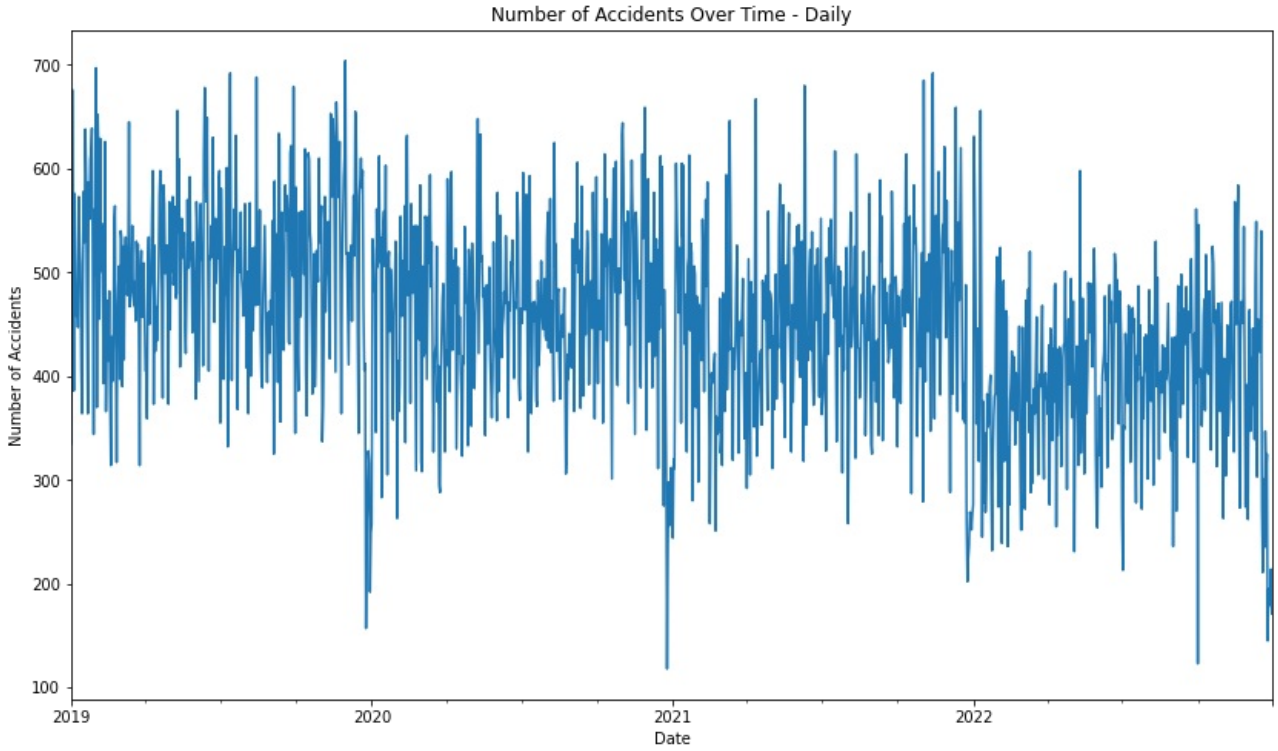
Accident Date	Index	Accident_Severity	Latitude	Light_Conditions	District Area	Longitude	Number_of_Casualties	Number_of_Vehicles
2019-05-06	200701BS64157	Serious	51.506187	Darkness - lights lit	Kensington and Chelsea	-0.209082	1	2
2019-02-07	200701BS65737	Serious	51.495029	Daylight	Kensington and Chelsea	-0.173647	1	2
2019-08-26	200701BS66127	Serious	51.517715	Darkness - lighting unknown	Kensington and Chelsea	-0.210215	1	3
2019-08-16	200701BS66128	Serious	51.495478	Daylight	Kensington and Chelsea	-0.202731	1	4
2019-03-09	200701BS66837	Slight	51.488576	Darkness - lights lit	Kensington and Chelsea	-0.192487	1	2
...	...	...	...	...	...	...	...	...
2022-02-18	201091NM01760	Slight	57.374005	Daylight	Highland	-3.467828	2	1
2022-02-21	201091NM01881	Slight	57.232273	Darkness - no lighting	Highland	-3.809281	1	1
2022-02-23	201091NM01935	Slight	57.585044	Daylight	Highland	-3.862727	1	3
2022-02-23	201091NM01964	Serious	57.214898	Darkness - no lighting	Highland	-3.823997	1	2
2022-02-28	201091NM02142	Serious	57.575210	Daylight	Highland	-3.895673	1	1

660660 rows × 13 columns

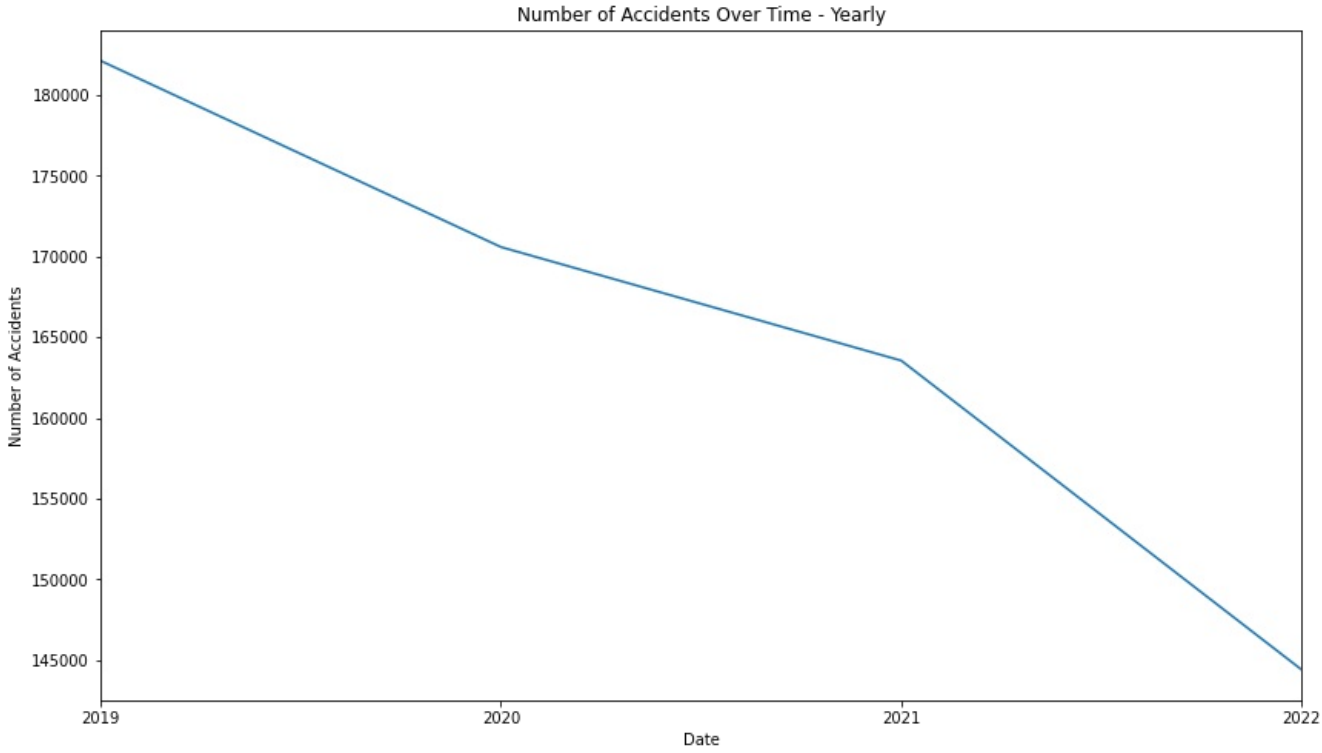
In [52]:
```
plt.figure(figsize=(14, 8))
df_new.resample('M').size().plot(legend=False)
plt.title('Number of Accidents Over Time - Monthly')
plt.xlabel('Date')
plt.ylabel('Number of Accidents')
plt.show()
```



In [53]:
```
plt.figure(figsize=(14, 8))
df_new.resample('D').size().plot(legend=False)
plt.title('Number of Accidents Over Time - Daily')
plt.xlabel('Date')
plt.ylabel('Number of Accidents')
plt.show()
```



In [54]:
```
plt.figure(figsize=(14, 8))
df_new.resample('Y').size().plot(legend=False)
plt.title('Number of Accidents Over Time - Yearly')
plt.xlabel('Date')
plt.ylabel('Number of Accidents')
plt.show()
```



In [55]:
```
pivot_table_result = pd.pivot_table(df, values='Number_of_Casualties', index='Accident Date', columns='Accident
pivot_table_result
```

Out[55]:

Accident_Severity	Fatal	Serious	Slight
Accident Date			
01-01-2019	3.0	6.0	8.0
01-01-2020	6.0	9.0	8.0
01-01-2021	7.0	4.0	5.0
01-01-2022	5.0	4.0	5.0
01-02-2019	3.0	5.0	6.0
...	...	...	...
31-10-2022	4.0	6.0	7.0
31-12-2019	3.0	3.0	9.0
31-12-2020	5.0	5.0	6.0
31-12-2021	2.0	6.0	5.0
31-12-2022	5.0	8.0	6.0

1461 rows × 3 columns

In [56]:
```
pivot_table_result = pd.pivot_table(df, values='Number_of_Casualties', index='Accident Date', columns='Accident
pivot_table_result
```

Out[56]:

Accident_Severity	Fatal	Serious	Slight
Accident Date			
01-01-2019	1.0	1.0	1.0
01-01-2020	1.0	1.0	1.0
01-01-2021	1.0	1.0	1.0
01-01-2022	1.0	1.0	1.0
01-02-2019	1.0	1.0	1.0
...	...	...	...
31-10-2022	1.0	1.0	1.0
31-12-2019	1.0	1.0	1.0
31-12-2020	1.0	1.0	1.0
31-12-2021	1.0	1.0	1.0
31-12-2022	1.0	1.0	1.0

1461 rows × 3 columns

In [57]:
```
pivot_table_result = pd.pivot_table(df, values='Number_of_Vehicles', index='Accident Date', columns='Accident_S
pivot_table_result
```

Out[57]:

Accident_Severity	Fatal	Serious	Slight
Accident Date			
01-01-2019	2.0	3.0	6.0
01-01-2020	3.0	2.0	7.0
01-01-2021	2.0	4.0	4.0
01-01-2022	1.0	3.0	5.0
01-02-2019	2.0	5.0	12.0
...	...	...	...
31-10-2022	1.0	6.0	6.0
31-12-2019	3.0	2.0	3.0
31-12-2020	3.0	4.0	16.0
31-12-2021	3.0	4.0	6.0
31-12-2022	2.0	3.0	4.0

1461 rows × 3 columns

In [58]:
```
pivot_table_result = pd.pivot_table(df, values='Number_of_Vehicles', index='Accident Date', columns='Accident_S
pivot_table_result
```

Out[58]:

Accident_Severity	Fatal	Serious	Slight
Accident Date			
01-01-2019	1.0	1.0	1.0
01-01-2020	1.0	1.0	1.0
01-01-2021	1.0	1.0	1.0
01-01-2022	1.0	1.0	1.0
01-02-2019	1.0	1.0	1.0
...	...	...	...
31-10-2022	1.0	1.0	1.0
31-12-2019	1.0	1.0	1.0
31-12-2020	1.0	1.0	1.0
31-12-2021	2.0	1.0	1.0
31-12-2022	1.0	1.0	1.0

1461 rows × 3 columns

In [59]:
```
crosstab_result = pd.crosstab(index=df['Accident_Severity'], columns=df['Weather_Conditions'])
crosstab_result
```

Out[59]:

Weather_Conditions	Fine + high winds	Fine no high winds	Fog or mist	Other	Raining + high winds	Raining no high winds	Snowing + high winds	Snowing no high winds
Accident_Severity								
Fatal	175	7207	82	165	145	848	3	36
Serious	1245	73285	483	1801	1261	9468	109	565
Slight	7133	454507	2963	15182	8209	69380	772	5636

In [60]:
```
crosstab_result = pd.crosstab(index=df['Accident_Severity'], columns=df['Light_Conditions'])
crosstab_result
```

Out[60]:

Light_Conditions	Darkness - lighting unknown	Darkness - lights lit	Darkness - lights unlit	Darkness - no lighting	Daylight
Accident_Severity					
Fatal	68	1860	45	1612	5076
Serious	794	19130	360	7174	60759
Slight	5622	108345	2138	28646	419031

In [61]:
```
crosstab_result = pd.crosstab(index=df['Accident_Severity'], columns=df['Road_Surface_Conditions'])
crosstab_result
```

Out[61]:

Road_Surface_Conditions	Dry	Flood over 3cm. deep	Frost or ice	Snow	Wet or damp
Accident_Severity					
Fatal	5790	23	193	35	2620
Serious	61708	152	2007	565	23785
Slight	381038	842	16314	5288	160300

In [62]:
```
crosstab_result = pd.crosstab(index=df['Accident_Severity'], columns=df['Road_Type'])
crosstab_result
```

Out[62]:

Road_Type	Dual carriageway	One way street	Roundabout	Single carriageway	Slip road
Accident_Severity					
Fatal	1815	95	142	6560	49
Serious	11746	1655	3665	70540	611
Slight	85855	11809	40182	419555	6381

In [63]:
```
crosstab_result = pd.crosstab(index=df['Accident_Severity'], columns=df['Urban_or_Rural_Area'])
crosstab_result
```

Out[63]:

Urban_or_Rural_Area	Rural	Unallocated	Urban
Accident_Severity			
Fatal	5601	0	3060
Serious	37312	1	50904
Slight	196061	10	367711

In [64]:
```
groupby_result = df.groupby(['Accident_Severity', 'Weather_Conditions'])['Number_of_Casualties'].max().unstack(
groupby_result
```

Out[64]:

Weather_Conditions	Fine + high winds	Fine no high winds	Fog or mist	Other	Raining + high winds	Raining no high winds	Snowing + high winds	Snowing no high winds
Accident_Severity								
Fatal	9	62	42	48	8	68	3	7
Serious	10	45	11	18	9	21	6	7
Slight	19	43	14	9	12	47	8	7

In [65]:
```
groupby_result = df.groupby(['Accident_Severity', 'Weather_Conditions'])['Number_of_Casualties'].min().unstack(
groupby_result
```

Out[65]:

Weather_Conditions	Fine + high winds	Fine no high winds	Fog or mist	Other	Raining + high winds	Raining no high winds	Snowing + high winds	Snowing no high winds
Accident_Severity								
Fatal	1	1	1	1	1	1	2	1
Serious	1	1	1	1	1	1	1	1
Slight	1	1	1	1	1	1	1	1

In [66]:
```
groupby_result = df.groupby(['Accident_Severity', 'Light_Conditions'])['Number_of_Casualties'].max().unstack()
groupby_result
```

Out[66]:

Light_Conditions	Darkness - lighting unknown	Darkness - lights lit	Darkness - lights unlit	Darkness - no lighting	Daylight
Accident_Severity					
Fatal	6	68	6	48	62
Serious	8	15	7	21	45
Slight	8	27	8	13	47

In [67]:
```
groupby_result = df.groupby(['Accident_Severity', 'Light_Conditions'])['Number_of_Casualties'].min().unstack()
groupby_result
```

Out[67]:

Light_Conditions	Darkness - lighting unknown	Darkness - lights lit	Darkness - lights unlit	Darkness - no lighting	Daylight
Accident_Severity					
Fatal	1	1	1	1	1
Serious	1	1	1	1	1
Slight	1	1	1	1	1

In [68]:
```
groupby_result = df.groupby(['Accident_Severity', 'Road_Surface_Conditions'])['Number_of_Casualties'].max().uns
groupby_result
```

Out[68]:

Road_Surface_Conditions	Dry	Flood over 3cm. deep	Frost or ice	Snow	Wet or damp
Accident_Severity					
Fatal	40	7	48	7	68
Serious	45	7	13	18	21
Slight	43	9	15	7	47

In [69]:
```
groupby_result = df.groupby(['Accident_Severity', 'Road_Surface_Conditions'])['Number_of_Casualties'].min().uns
groupby_result
```

Out[69]:

Road_Surface_Conditions	Dry	Flood over 3cm. deep	Frost or ice	Snow	Wet or damp
Accident_Severity					
Fatal	1	1	1	1	1
Serious	1	1	1	1	1
Slight	1	1	1	1	1

In [70]:
```
groupby_result = df.groupby(['Accident_Severity', 'Road_Type'])['Number_of_Casualties'].max().unstack()
groupby_result
```

Out[70]:

Road_Type	Dual carriageway	One way street	Roundabout	Single carriageway	Slip road
Accident_Severity					
Fatal	42	8	4	62	68
Serious	45	15	19	42	16
Slight	47	13	17	29	14

In [71]:
```
groupby_result = df.groupby(['Accident_Severity', 'Road_Type'])['Number_of_Casualties'].min().unstack()
groupby_result
```

Out[71]:

Road_Type	Dual carriageway	One way street	Roundabout	Single carriageway	Slip road
Accident_Severity					
Fatal	1	1	1	1	1
Serious	1	1	1	1	1
Slight	1	1	1	1	1

In [72]:
```
groupby_result = df.groupby(['Accident_Severity', 'Urban_or_Rural_Area'])['Number_of_Casualties'].max().unstack
groupby_result
```

Out[72]:

Urban_or_Rural_Area	Rural	Unallocated	Urban
Accident_Severity			
Fatal	68.0	NaN	21.0
Serious	45.0	1.0	29.0
Slight	47.0	3.0	32.0

In [73]:
```
groupby_result = df.groupby(['Accident_Severity', 'Urban_or_Rural_Area'])['Number_of_Casualties'].min().unstack
groupby_result
```

Out[73]:

Urban_or_Rural_Area	Rural	Unallocated	Urban
Accident_Severity			
Fatal	1.0	NaN	1.0
Serious	1.0	1.0	1.0
Slight	1.0	1.0	1.0

In [74]:
```
df
```

Out[74]:

	Index	Accident_Severity	Accident Date	Latitude	Light_Conditions	District Area	Longitude	Number_of_Casualties	Number_of_
0	200701BS64157	Serious	05-06-2019	51.506187	Darkness - lights lit	Kensington and Chelsea	-0.209082	1	
1	200701BS65737	Serious	02-07-2019	51.495029	Daylight	Kensington and Chelsea	-0.173647	1	
2	200701BS66127	Serious	26-08-2019	51.517715	Darkness - lighting unknown	Kensington and Chelsea	-0.210215	1	
3	200701BS66128	Serious	16-08-2019	51.495478	Daylight	Kensington and Chelsea	-0.202731	1	
4	200701BS66837	Slight	03-09-2019	51.488576	Darkness - lights lit	Kensington and Chelsea	-0.192487	1	
...	...	...	...	...	...	...	...	...	...
660674	201091NM01760	Slight	18-02-2022	57.374005	Daylight	Highland	-3.467828	2	
660675	201091NM01881	Slight	21-02-2022	57.232273	Darkness - no lighting	Highland	-3.809281	1	
660676	201091NM01935	Slight	23-02-2022	57.585044	Daylight	Highland	-3.862727	1	
660677	201091NM01964	Serious	23-02-2022	57.214898	Darkness - no lighting	Highland	-3.823997	1	
660678	201091NM02142	Serious	28-02-2022	57.575210	Daylight	Highland	-3.895673	1	

660660 rows × 14 columns

In [75]:
```
df = df.drop(['Index', 'Accident Date', 'Latitude', 'Longitude'], axis=1)
```

In [76]:
```
label_mapping = {'Slight': 0, 'Serious': 1, 'Fatal': 2}
df['Accident_Severity'] = df['Accident_Severity'].map(label_mapping)
```

In [77]:
```
df = pd.get_dummies(df, columns=['Light_Conditions', 'District Area', 'Road_Surface_Conditions', 'Road_Type', '
```

In [78]:
```
df
```

Out[78]:

	Accident_Severity	Number_of_Casualties	Number_of_Vehicles	Light_Conditions_Darkness - lighting unknown	Light_Conditions_Darkness - lights lit	Light_Conditio
0	1	1	2	0	1	
1	1	1	2	0	0	
2	1	1	3	1	0	
3	1	1	4	0	0	
4	0	1	2	0	1	
...	...	...	...	...	...	
660674	0	2	1	0	0	
660675	0	1	1	0	0	
660676	0	1	3	0	0	
660677	1	1	2	0	0	
660678	1	1	1	0	0	

660660 rows × 467 columns

In [79]:
```
from sklearn.model_selection import train_test_split

X = df.drop('Accident_Severity', axis=1)
y = df['Accident_Severity']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, stratify = y, random_state=42)
```

In [80]:
```
from sklearn.linear_model import LogisticRegression
```

In [81]:
```
model = LogisticRegression(random_state=42)
```

In [82]:
```
model.fit(X_train, y_train)
```

Out[82]:
```
LogisticRegression(random_state=42)
```

In [83]:
```
y_pred = model.predict(X_test)
```

In [84]:
```
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

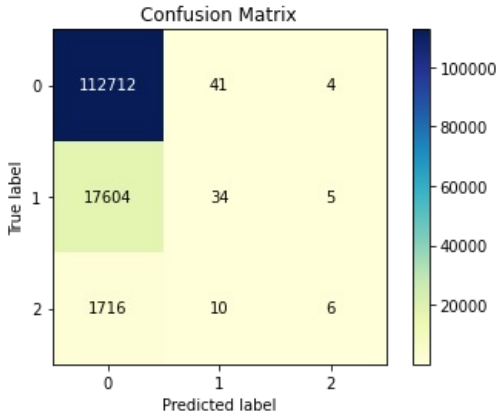
In [85]:
```
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)

print(f'Accuracy: {accuracy:.2f}')
print(f'Confusion Matrix:\n{conf_matrix}')
print(f'Classification Report:\n{classification_rep}')
```

```
Accuracy: 0.85
Confusion Matrix:
[[112712     41      4]
 [ 17604     34      5]
 [  1716     10      6]]
Classification Report:
              precision    recall  f1-score   support

           0       0.85      1.00      0.92    112757
           1       0.40      0.00      0.00     17643
           2       0.40      0.00      0.01      1732

    accuracy                           0.85    132132
   macro avg       0.55      0.33      0.31    132132
weighted avg       0.79      0.85      0.79    132132
```

In [86]:
```
from scikitplot.metrics import plot_confusion_matrix
```

In [87]:
```
plot_confusion_matrix(y_test, y_pred, cmap= 'YlGnBu')
```

```
plt.show()
```



Thanks !!!