```
# IMPORTANT: RUN THIS CELL IN ORDER TO IMPORT YOUR KAGGLE DATA SOURCES,
# THEN FEEL FREE TO DELETE THIS CELL.
# NOTE: THIS NOTEBOOK ENVIRONMENT DIFFERS FROM KAGGLE'S PYTHON
# ENVIRONMENT SO THERE MAY BE MISSING LIBRARIES USED BY YOUR
# NOTEBOOK.
import kagglehub
khushikyad001_india_road_accident_dataset_predictive_analysis_path = kagglehub.dataset_download('k

print('Data source import complete.')
```

```
Downloading from https://www.kaggle.com/api/v1/datasets/download/khushikyad001/india-road-acci
100%|██████████| 68.2k/68.2k [00:00<00:00, 41.0MB/s]Extracting files...
Data source import complete.
```

Table of Content

Start coding or generate with AI.

Importing Libraries

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

```
data=pd.read_csv('/content/accident_prediction_india.csv')
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 22 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   State Name                   3000 non-null   object
 1   City Name                    3000 non-null   object
 2   Year                         3000 non-null   int64
 3   Month                        3000 non-null   object
 4   Day of Week                  3000 non-null   object
 5   Time of Day                  3000 non-null   object
 6   Accident Severity            3000 non-null   object
 7   Number of Vehicles Involved  3000 non-null   int64 
 8   Vehicle Type Involved        3000 non-null   object
 9   Number of Casualties         3000 non-null   int64 
 10  Number of Fatalities         3000 non-null   int64 
 11  Weather Conditions           3000 non-null   object
 12  Road Type                    3000 non-null   object
 13  Road Condition               3000 non-null   object
 14  Lighting Conditions          3000 non-null   object
 15  Traffic Control Presence     2284 non-null   object
 16  Speed Limit (km/h)           3000 non-null   int64 
 17  Driver Age                   3000 non-null   int64 
 18  Driver Gender                3000 non-null   object
 19  Driver License Status        2025 non-null   object
 20  Alcohol Involvement          3000 non-null   object
 21  Accident Location Details    3000 non-null   object
dtypes: int64(6), object(16)
memory usage: 515.8+ KB
```

Cleaning the Data

Check for null values:

```
#First it's better to take a copy of dataset and work on a copy
df=data.copy()

#Checking null values accross the dataset
df.isnull().sum().sort_values(ascending=False)
```

```
                                    0
Driver License Status             975
Traffic Control Presence          716
Year                                0
Month                               0
State Name                          0
City Name                           0
Time of Day                         0
Day of Week                         0
Accident Severity                   0
Number of Vehicles Involved         0
Number of Fatalities                0
Weather Conditions                  0
Vehicle Type Involved               0
Number of Casualties                0
Road Condition                      0
Road Type                           0
Speed Limit (km/h)                  0
Lighting Conditions                 0
Driver Age                          0
Driver Gender                       0
Alcohol Involvement                 0
Accident Location Details           0

dtype: int64
```

```
df['Driver License Status'].value_counts()
```

```
                          count
Driver License Status
        Valid              1057
       Expired              968

dtype: int64
```

```
df['Traffic Control Presence'].value_counts()
```

```
                                count
Traffic Control Presence
                  Signs           812
                Signals           736
       Police Checkpost           736

dtype: int64
```

```
df['Driver License Status']=df['Driver License Status'].fillna('Unknown')
df['Traffic Control Presence']=df['Traffic Control Presence'].fillna('Unknown')
```

Check for duplicated rows:

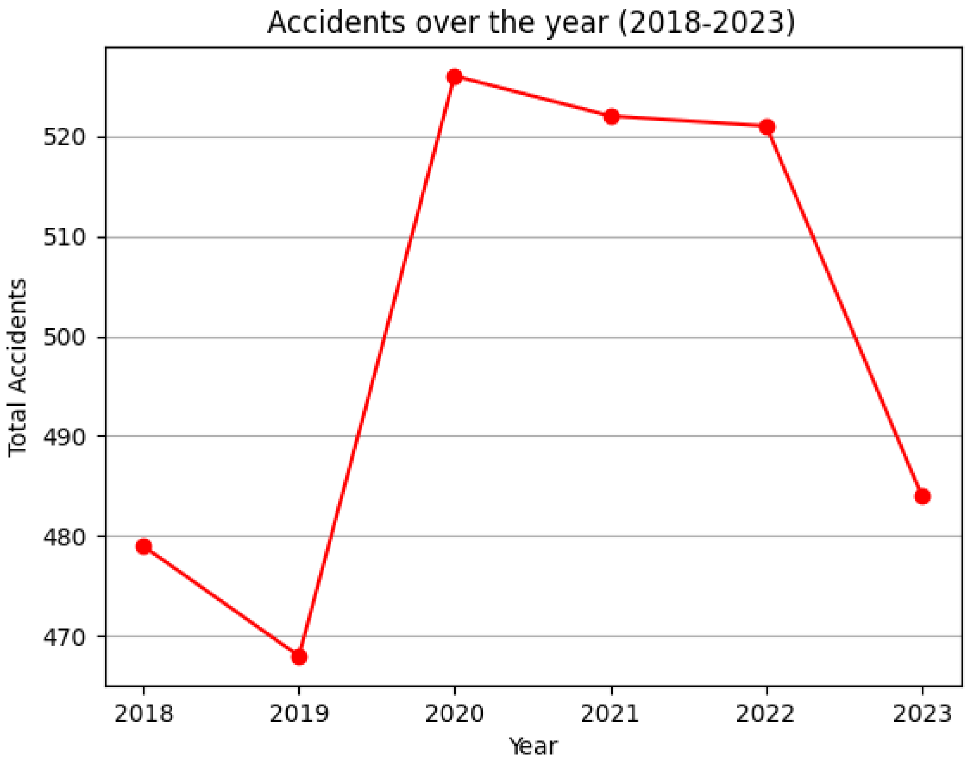
```
df.duplicated().sum()
```

```
np.int64(0)
```

- Fortunately, there are no duplicate rows.

Date related Analysis

```
df['Year']=pd.to_datetime(df['Year'],format='%Y')
yearly_accident=df.groupby('Year')['Accident Severity'].count()

#Let's plot yearly_accidents to hvae better intuition.

plt.plot(yearly_accident.index,yearly_accident.values,marker='o',linestyle='-',color='red')
plt.title('Accidents over the year (2018-2023)')
plt.xlabel('Year')
plt.ylabel('Total Accidents')
plt.grid(axis='y')
plt.show()
```



Categorical Analysis

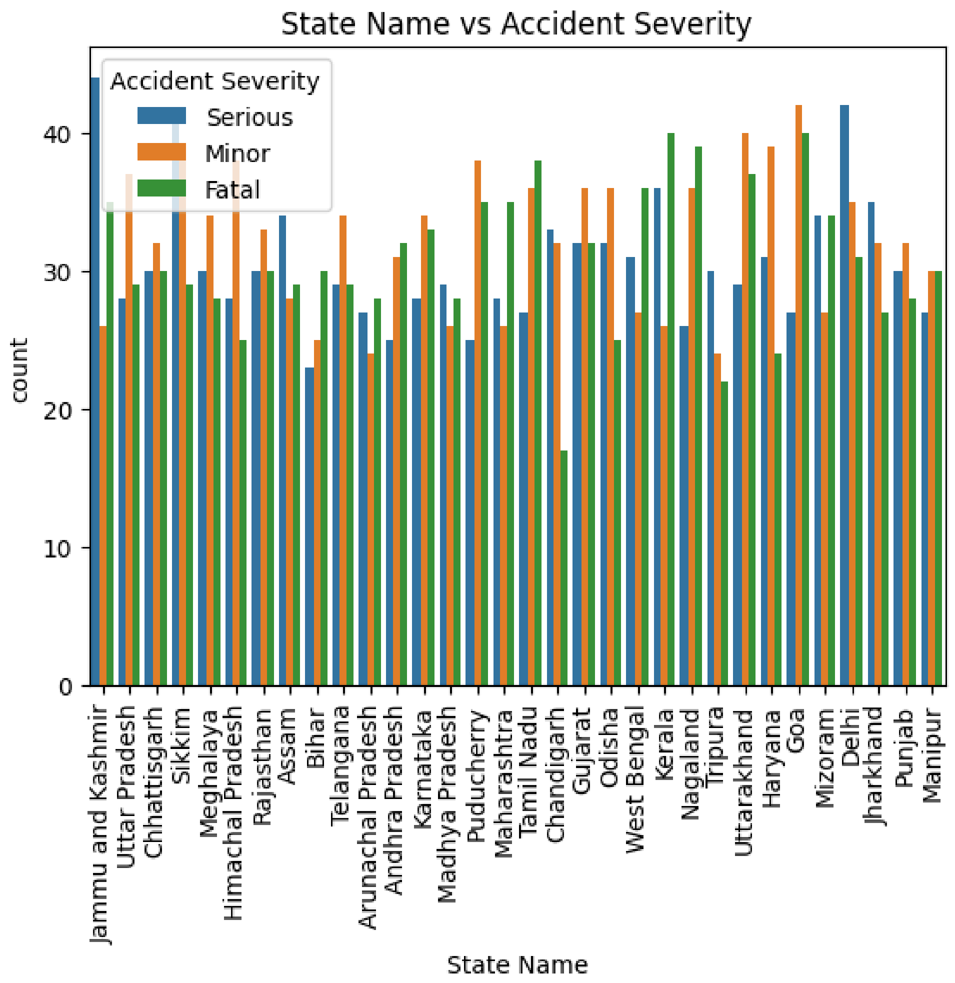
```
#Let's see the number of unique categories in each column:

cat_features=list(df.select_dtypes(include=object).columns)
for i in cat_features:
    print(f'{i}: {df[i].nunique()}')
```

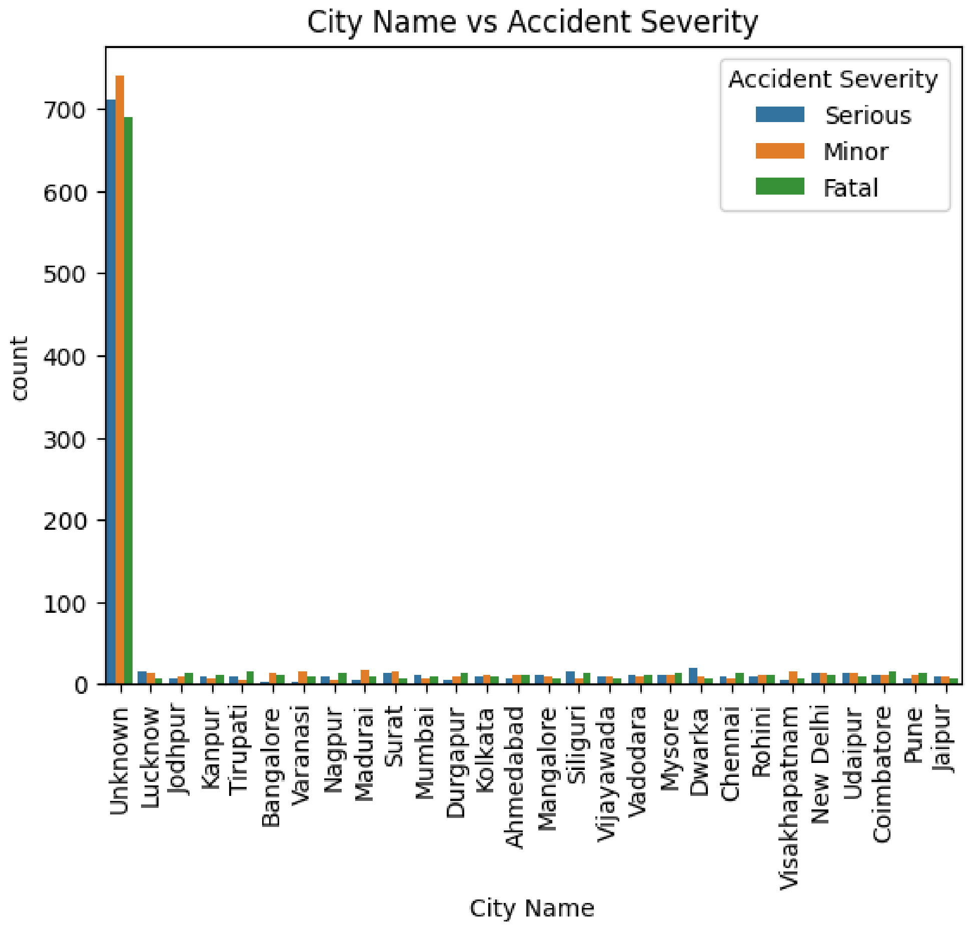
```
State Name: 32
City Name: 28
Month: 12
Day of Week: 7
Accident Severity: 3
Vehicle Type Involved: 7
Weather Conditions: 5
Road Type: 4
Road Condition: 4
Lighting Conditions: 4
Traffic Control Presence: 4
Driver Gender: 2
Driver License Status: 3
Alcohol Involvement: 2
Accident Location Details: 4
Time_Category: 4
```

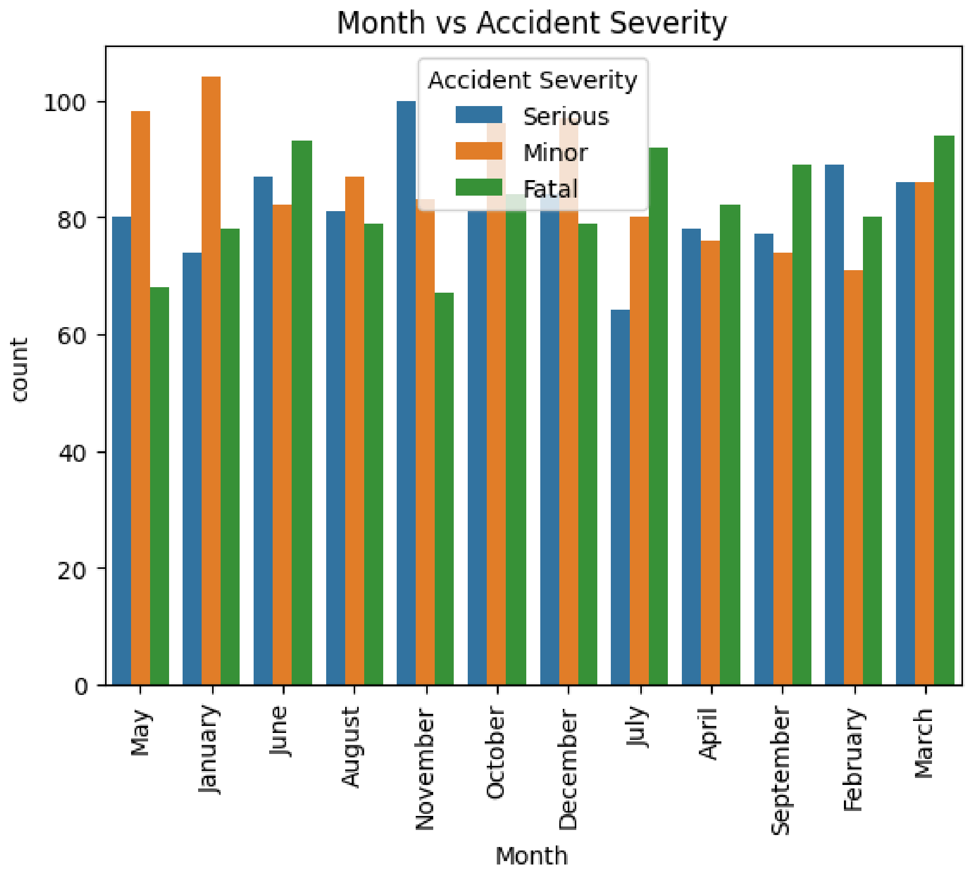
```
#Since Time of Day has a lot of unique values it's not recomended to plot it using countplots so I
cat_columns=cat_features.remove('Time of Day')
```

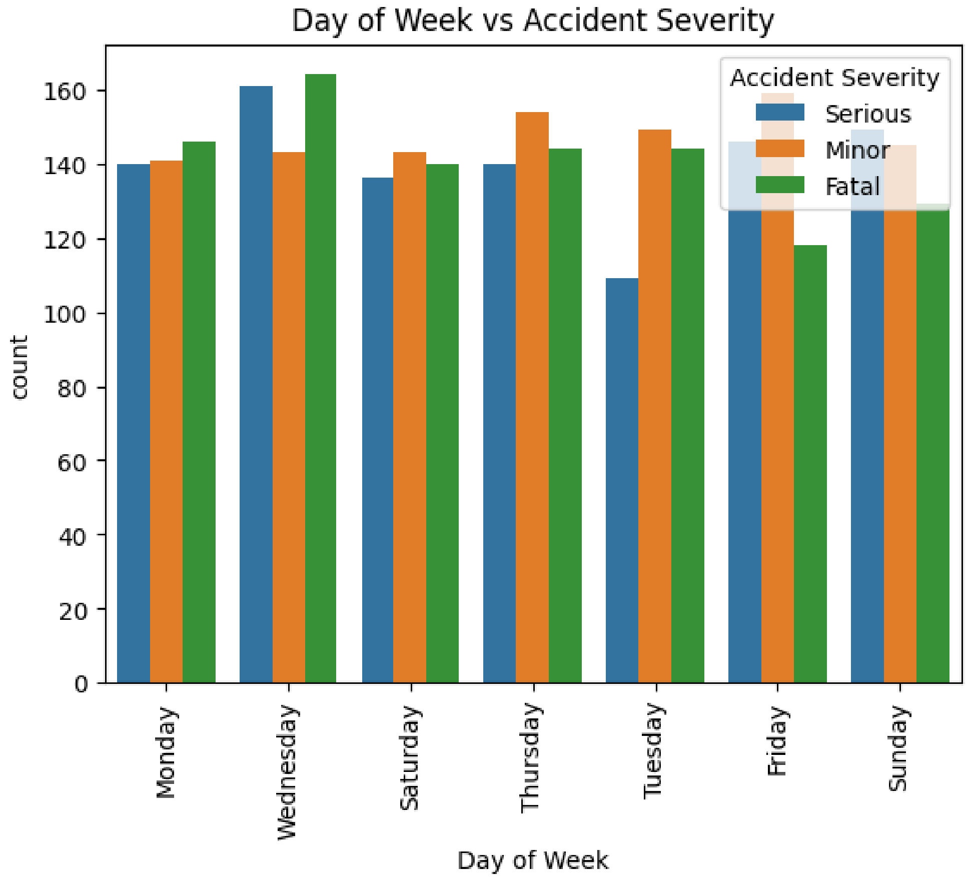
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-38-65ef615b119c> in <cell line: 0>()
      1 #Since Time of Day has a lot of unique values it's not recomended to plot it using
countplots so I remove it from the plot list
----> 2 cat_columns=cat_features.remove('Time of Day')

ValueError: list.remove(x): x not in list
```

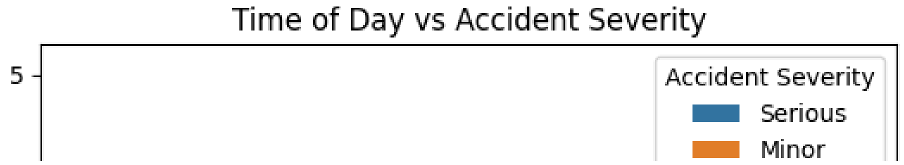
```
for i in cat_features:
    plt.figure()
    sns.countplot(data=df,x=i,hue='Accident Severity')
    plt.xticks(rotation=90)
    plt.title(f'{i} vs Accident Severity')

plt.tight_layout()
plt.show()
```

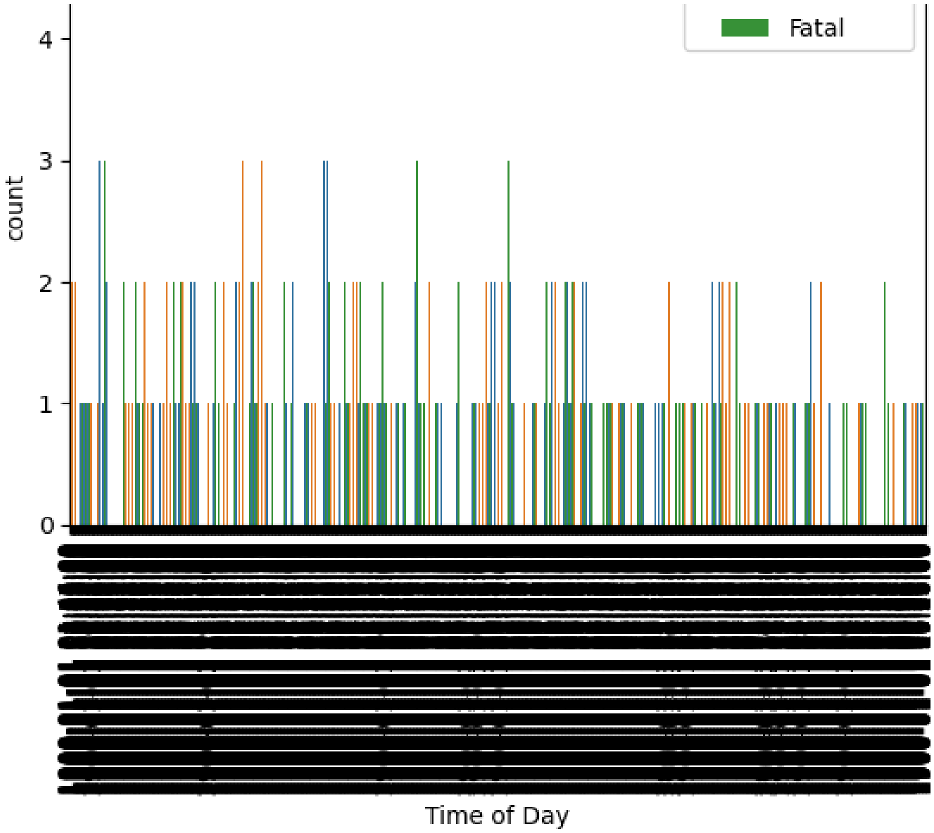

State Name vs Accident Severity
Accident Severity
Serious
Minor
Fatal
count
0
10
20
30
40
Jammu and Kashmir
Uttar Pradesh
Chhattisgarh
Sikkim
Meghalaya
Himachal Pradesh
Rajasthan
Assam
Bihar
Telangana
Arunachal Pradesh
Andhra Pradesh
Karnataka
Madhya Pradesh
Puducherry
Maharashtra
Tamil Nadu
Chandigarh
Gujarat
Odisha
West Bengal
Kerala
Nagaland
Tripura
Uttarakhand
Haryana
Goa
Mizoram
Delhi
Jharkhand
Punjab
Manipur
State Name

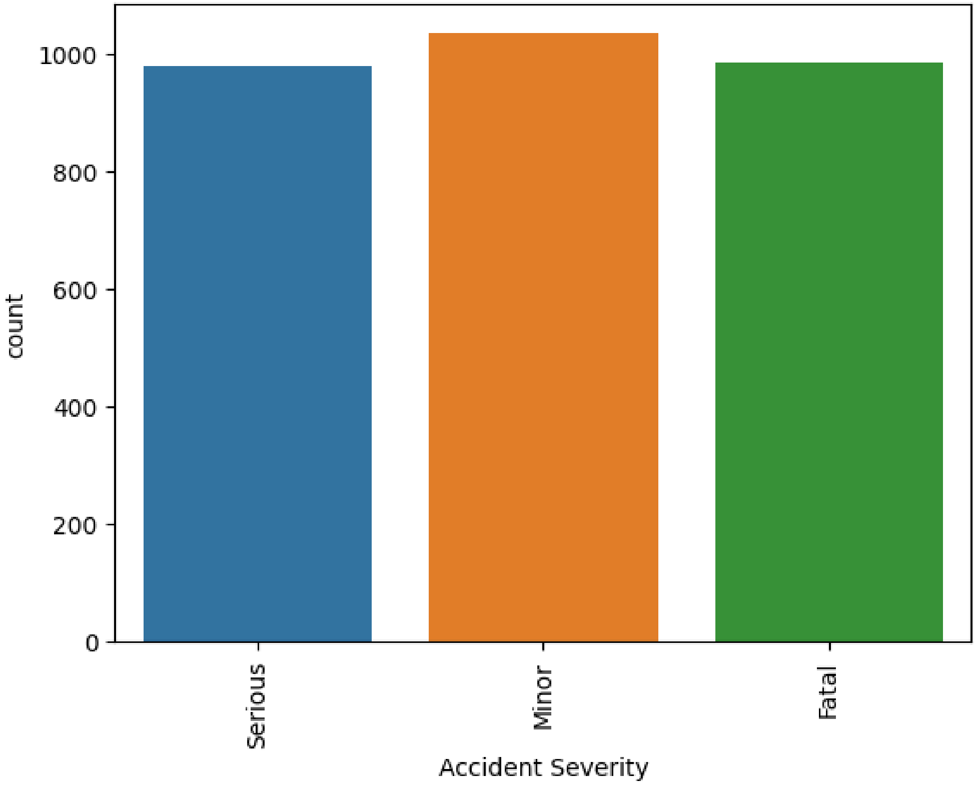

City Name vs Accident Severity
Accident Severity
Serious
Minor
Fatal
count
0
100
200
300
400
500
600
700
Unknown
Lucknow
Jodhpur
Kanpur
Tirupati
Bangalore
Varanasi
Nagpur
Madurai
Surat
Mumbai
Durgapur
Kolkata
Ahmedabad
Mangalore
Siliguri
Vijayawada
Vadodara
Mysore
Dwarka
Chennai
Rohini
Visakhapatnam
New Delhi
Udaipur
Coimbatore
Pune
Jaipur
City Name

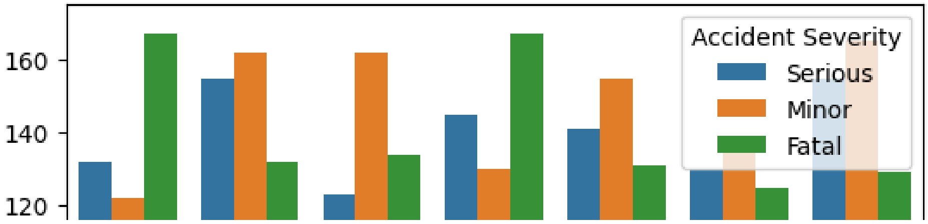

Month vs Accident Severity

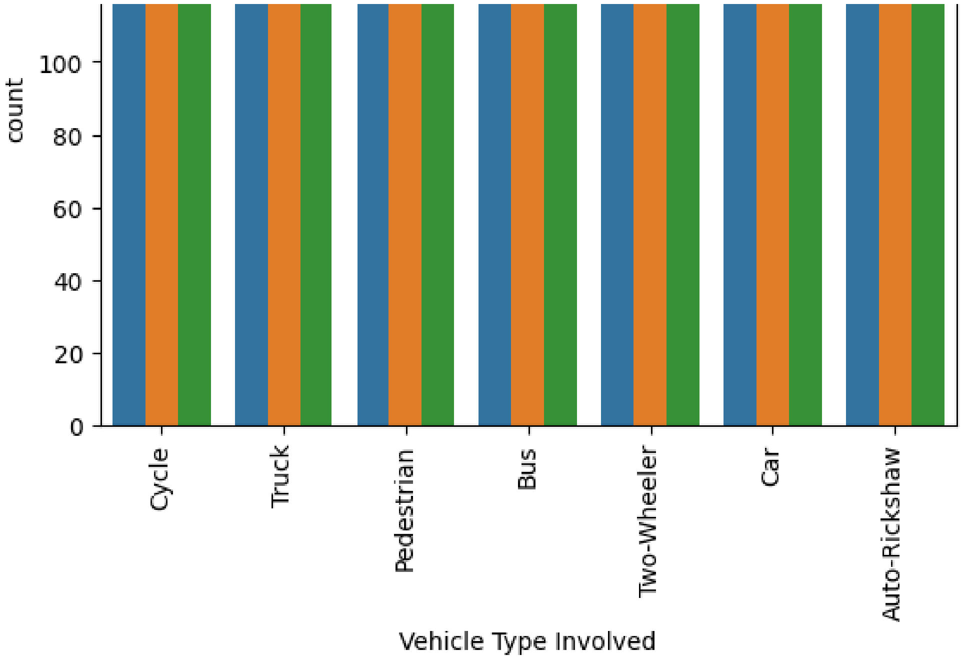

Day of Week vs Accident Severity


Time of Day vs Accident Severity

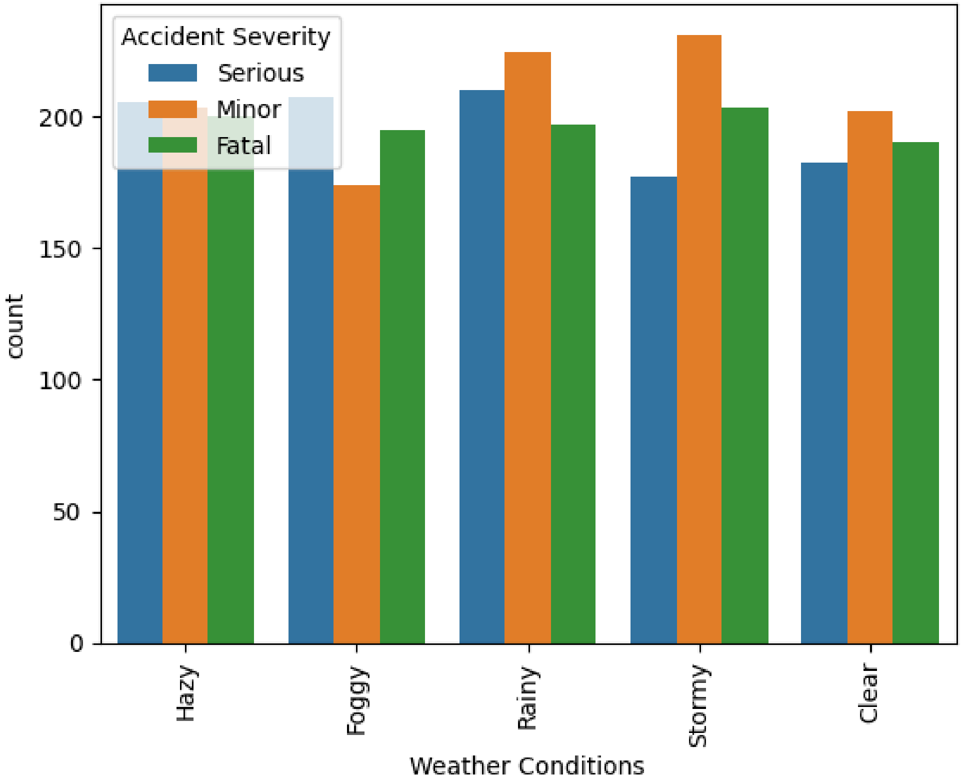



Accident Severity vs Accident Severity

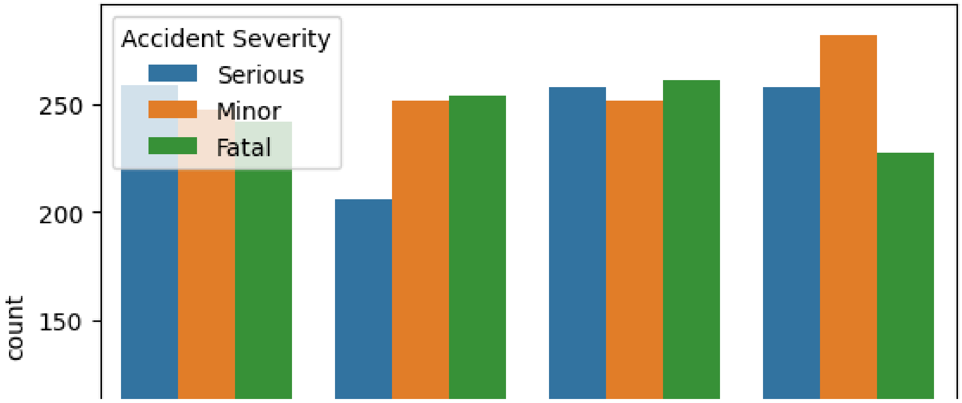

Vehicle Type Involved vs Accident Severity

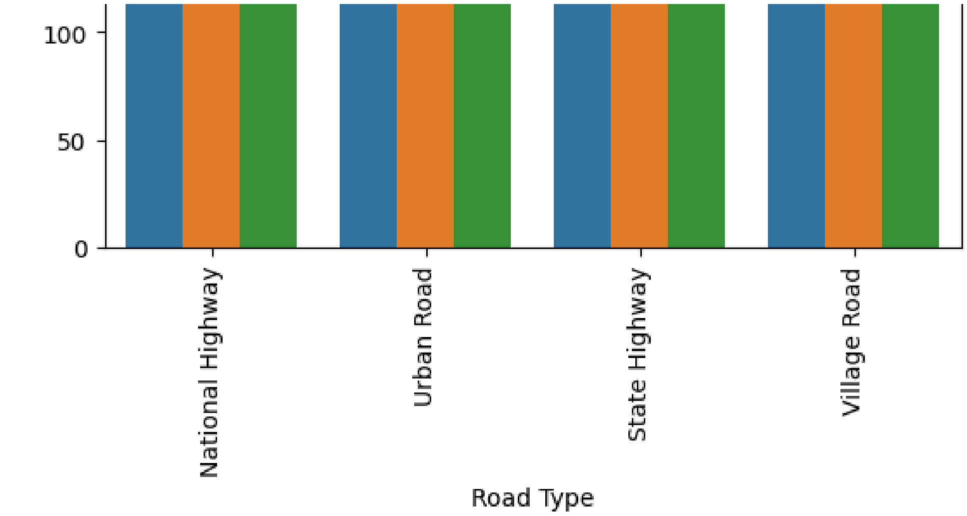


Weather Conditions vs Accident Severity

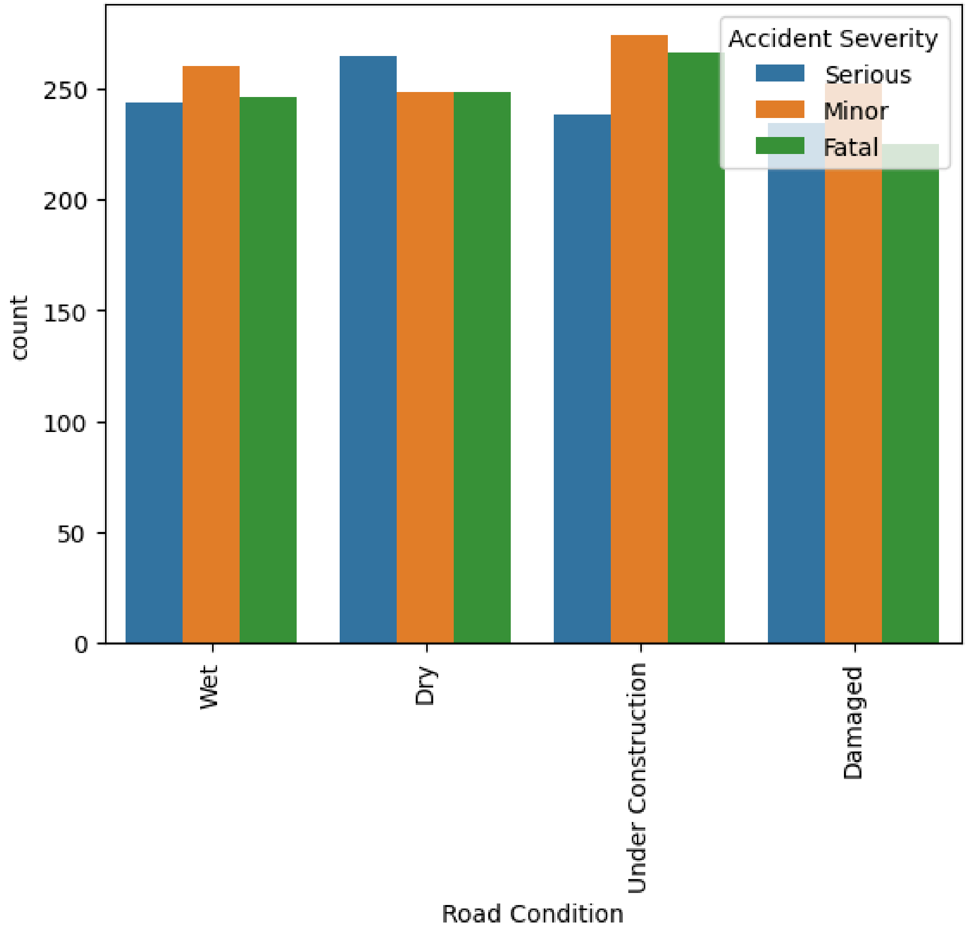


Road Type vs Accident Severity

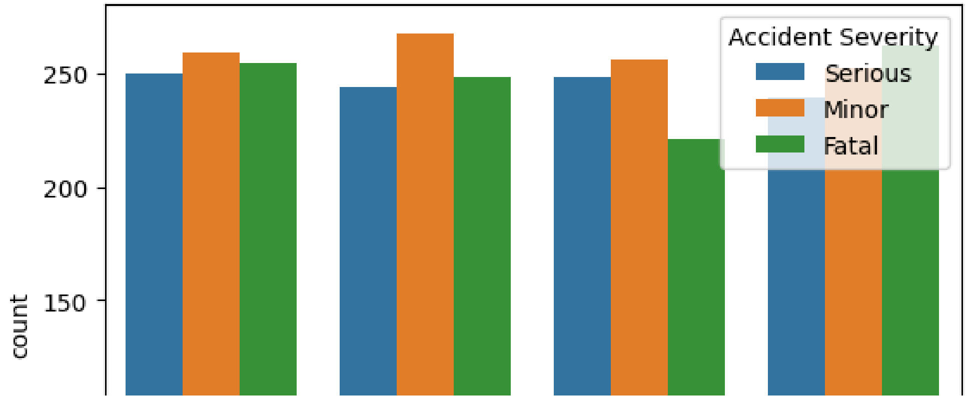


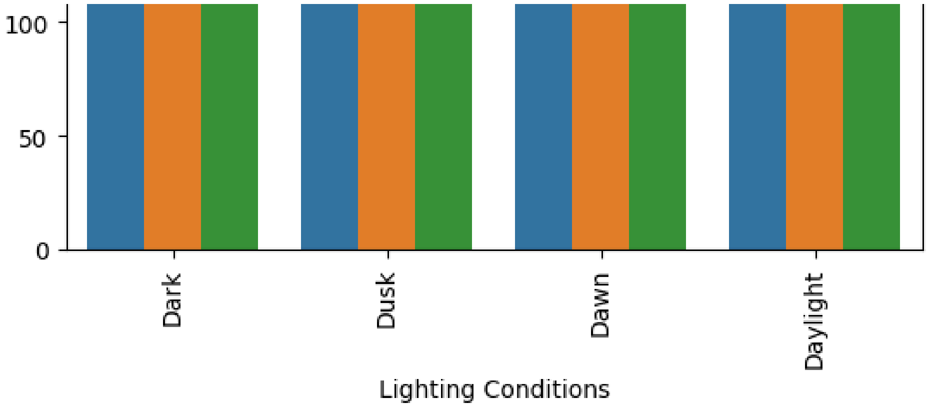


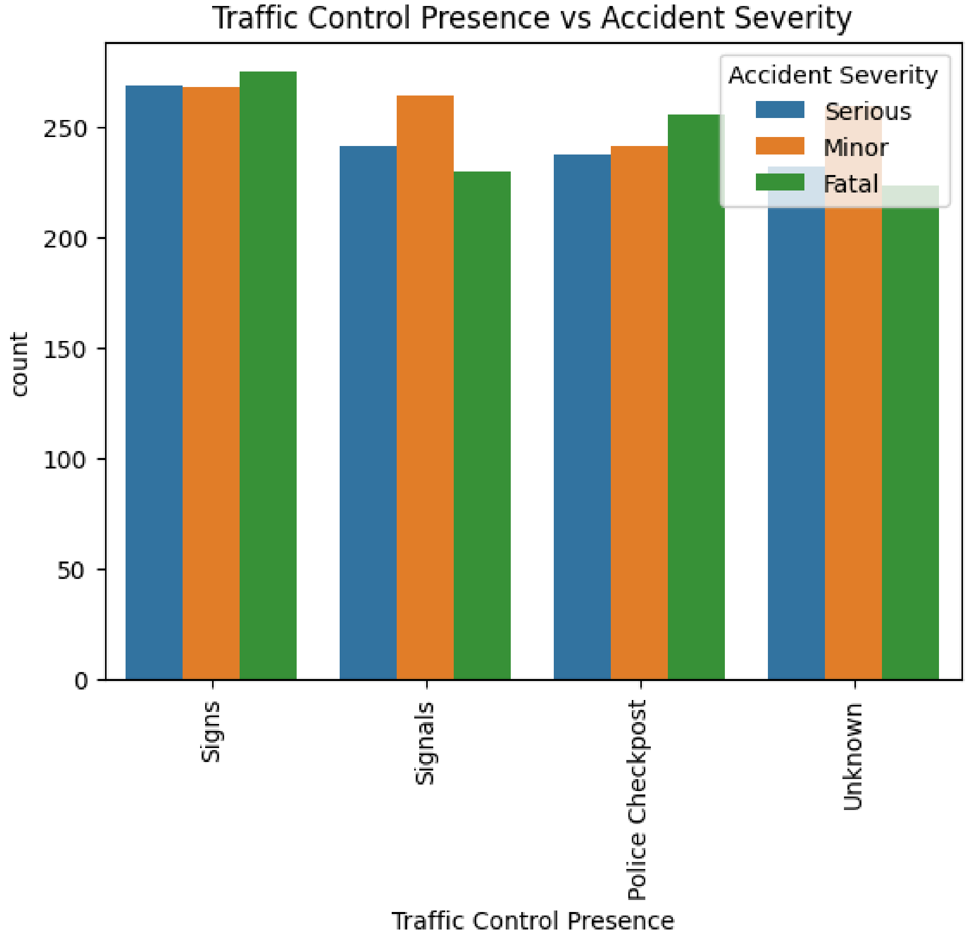
Road Condition vs Accident Severity

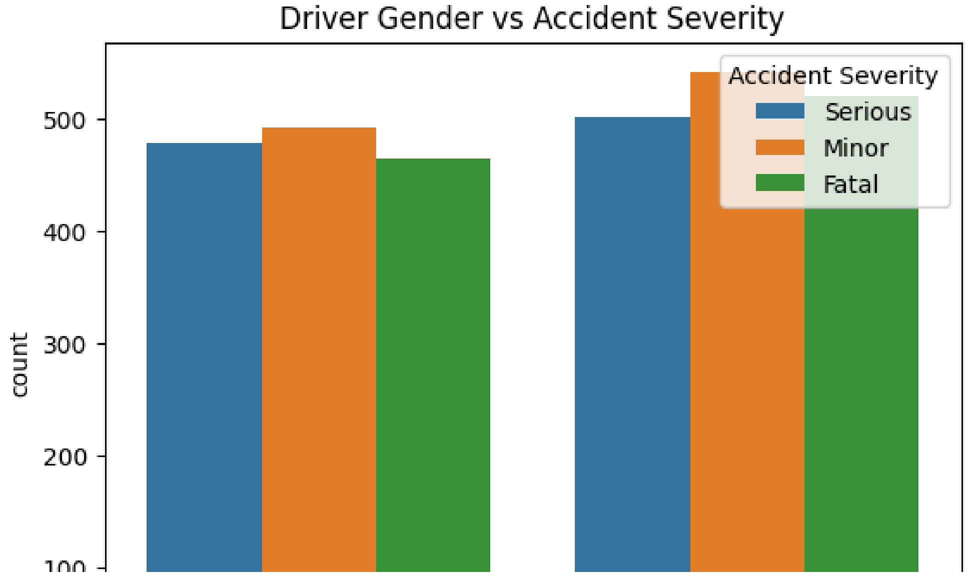


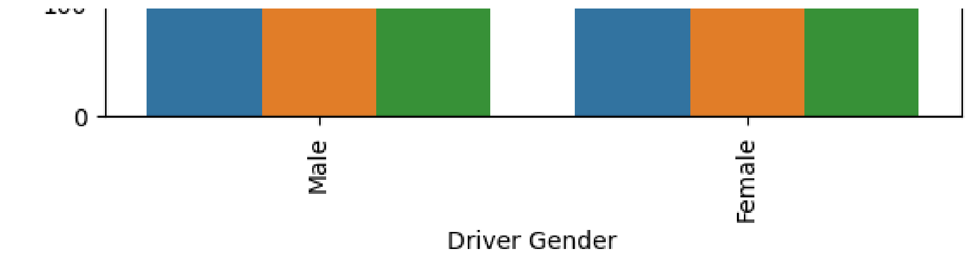
Lighting Conditions vs Accident Severity



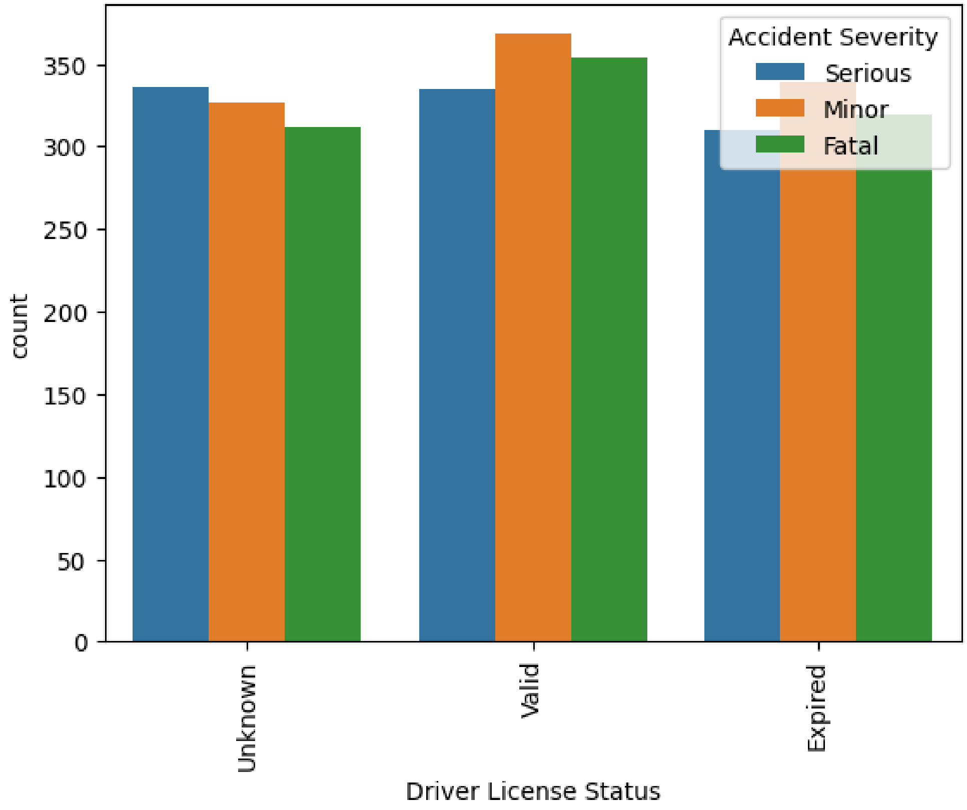



Traffic Control Presence vs Accident Severity

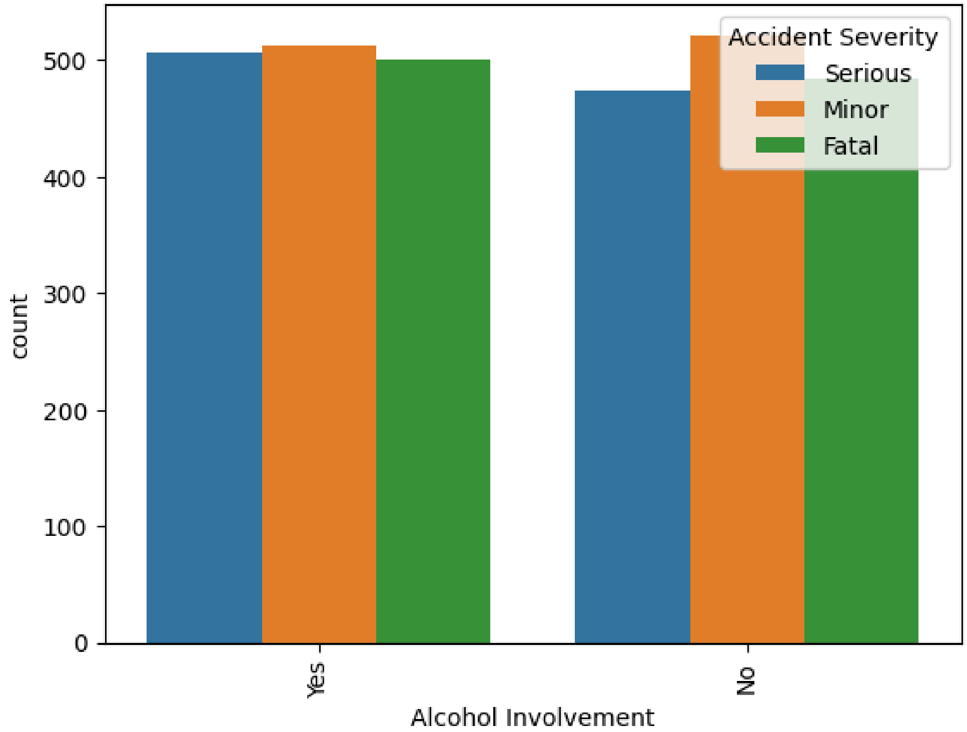

Driver Gender vs Accident Severity



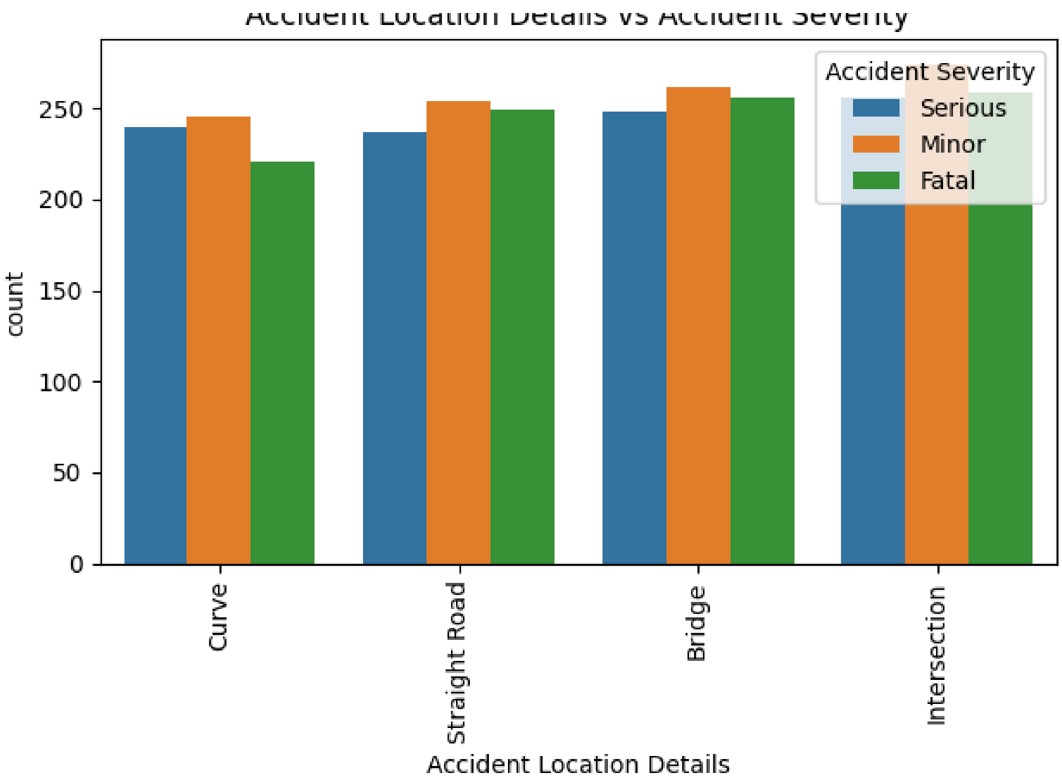
Driver License Status vs Accident Severity



Alcohol Involvement vs Accident Severity



Accident Location Details vs Accident Severity


Accident Location Details vs Accident Severity
Accident Severity
Serious
Minor
Fatal
count
0
50
100
150
200
250
Curve
Straight Road
Bridge
Intersection
Accident Location Details