ALLIANCE
UNIVERSITY

Private University established in Karnataka State by Act No.34 of year 2010
Recognized by the University Grants Commission (UGC), New Delhi

Project Report

Introduction to Data Science

Semester – 2

"Road Accident Data"

By

Laishram Ritikumar Singh

Reg no: 2411021240040

GitHub link: https://github.com/Ritikumar2007/IDS-DATASET-PROJECT

Department of Computer Application

Alliance University Chandapura — Anekal Main Road,

Anekal Bengaluru — 562 106

April 2025

Project Overview

In this project, I explored and analyzed a real-world dataset on road accidents. The goal was to understand patterns and trends that could help explain when, where, and why these accidents happen. Using Python and tools like pandas and NumPy, I worked through the data to uncover useful insights that could potentially contribute to improving road safety.

Introduction

Road accidents are an unfortunate but common part of everyday life, and they can have devastating consequences. Understanding the factors behind these accidents—like the time of day, location, or weather conditions—can make a big difference in preventing them. This project is all about diving into a dataset filled with real accident records to see what stories the data tells. With the help of Python, I set out to clean, explore, and visualize this information to make sense of the chaos on the roads

Project Goals

Here's what I set out to achieve with this project:

- **Load and understand** the road accident data from a CSV file.
- **Clean and prepare** the data for meaningful analysis.
- **Explore trends**—like the most dangerous times or locations.
- **Visualize key insights** using graphs and charts.
- **Highlight factors** that could help reduce accidents in the future.

Challenges

Like any real-world project, this one came with a few bumps in the road:

- Some of the data was messy or incomplete, which made cleaning a bit tricky.
- The dataset was pretty large, so performance was a factor when processing it.
- Not every column was self-explanatory, so I had to spend time figuring out what some of the data actually meant.
- And of course, while data can tell us a lot, it's not perfect—some patterns might be influenced by biases or missing context.

Conclusion

Working on this project gave me a deeper appreciation for the power of data in solving real-world problems. By analyzing this road accident data, I was able to find patterns that might help raise awareness or guide better safety measures. While there were a few challenges along the way, the process of turning raw data into actionable insights was both rewarding and eye-opening. It's a great reminder of how much impact data analysis can have when it’s used to tackle important issues.

```
import pandas as pd
import numpy as np

df = pd.read_csv(r"C:\Users\abhin\Downloads\Road Accident Data.csv")

df
```

```
       Accident_Index Accident Date Day_of_Week
Junction_Control  \
0       200901BS70001      1/1/2021    Thursday      Give way or
uncontrolled
1       200901BS70002      1/5/2021      Monday      Give way or
uncontrolled
2       200901BS70003      1/4/2021      Sunday      Give way or
uncontrolled
3       200901BS70004      1/5/2021      Monday           Auto traffic
signal
4       200901BS70005      1/6/2021     Tuesday           Auto traffic
signal
...                ...           ...         ...
...
307968  201091NM01760     2/18/2022    Thursday  Data missing or out
of range
307969  201091NM01881     2/21/2022      Sunday  Data missing or out
of range
307970  201091NM01935     2/23/2022     Tuesday      Give way or
uncontrolled
307971  201091NM01964     2/23/2022     Tuesday      Give way or
uncontrolled
307972  201091NM02142     2/28/2022      Sunday      Give way or
uncontrolled

                            Junction_Detail Accident_Severity
Latitude  \
0                   T or staggered junction           Serious
51.512273
1                                Crossroads           Serious
51.514399
2                   T or staggered junction            Slight
51.486668
3                   T or staggered junction           Serious
51.507804
4                                Crossroads           Serious
51.482076
...                                     ...               ...        .
..
307968  Not at junction or within 20 metres            Slight
57.374005
307969  Not at junction or within 20 metres            Slight
57.232273
```

```
307970              T or staggered junction            Slight
57.585044
307971              T or staggered junction           Serious
57.214898
307972              T or staggered junction           Serious
57.575210

              Light_Conditions Local_Authority_(District)  \
0                    Daylight     Kensington and Chelsea
1                    Daylight     Kensington and Chelsea
2                    Daylight     Kensington and Chelsea
3                    Daylight     Kensington and Chelsea
4        Darkness - lights lit     Kensington and Chelsea
...                        ...                        ...
307968               Daylight                  Highland
307969  Darkness - no lighting                  Highland
307970               Daylight                  Highland
307971  Darkness - no lighting                  Highland
307972               Daylight                  Highland

         Carriageway_Hazards  ...  Number_of_Casualties
Number_of_Vehicles  \
0                        NaN  ...                    1
2
1                        NaN  ...                   11
2
2                        NaN  ...                    1
2
3                        NaN  ...                    1
2
4                        NaN  ...                    1
2
...                      ...  ...                  ...
...
307968                   NaN  ...                    2
1
307969                   NaN  ...                    1
1
307970                   NaN  ...                    1
3
307971                   NaN  ...                    1
2
307972  Other object on road  ...                    1
1

               Police_Force Road_Surface_Conditions
Road_Type  \
0       Metropolitan Police                    Dry      One way
street
1       Metropolitan Police             Wet or damp  Single
```

```
carriageway
2       Metropolitan Police                     Dry  Single
carriageway
3       Metropolitan Police            Frost or ice  Single
carriageway
4       Metropolitan Police                     Dry  Single
carriageway
...                     ...                     ...                  ..
.
307968             Northern                     Dry  Single
carriageway
307969             Northern            Frost or ice  Single
carriageway
307970             Northern            Frost or ice  Single
carriageway
307971             Northern             Wet or damp  Single
carriageway
307972             Northern             Wet or damp    Dual
carriageway

       Speed_limit   Time Urban_or_Rural_Area
Weather_Conditions  \
0               30  15:11               Urban     Fine no high winds

1               30  10:59               Urban     Fine no high winds

2               30  14:19               Urban     Fine no high winds

3               30   8:10               Urban                  Other

4               30  17:25               Urban     Fine no high winds

...            ...    ...                 ...                     ...

307968          60   7:00               Rural     Fine no high winds

307969          60   3:00               Rural     Fine no high winds

307970          30   9:38               Rural     Fine no high winds

307971          60  18:25               Rural     Fine no high winds

307972          60  15:45               Rural  Snowing no high winds


                  Vehicle_Type
0                          Car
1       Taxi/Private hire car
2       Taxi/Private hire car
3       Motorcycle over 500cc
```

```
4                          Car
...                        ...
307968                     Car
307969                     Car
307970                     Car
307971  Motorcycle over 500cc
307972                     Car

[307973 rows x 21 columns]
```

```
df.head()
```

```
  Accident_Index Accident Date Day_of_Week
Junction_Control  \
0  200901BS70001      1/1/2021    Thursday  Give way or uncontrolled

1  200901BS70002      1/5/2021      Monday  Give way or uncontrolled

2  200901BS70003      1/4/2021      Sunday  Give way or uncontrolled

3  200901BS70004      1/5/2021      Monday       Auto traffic signal

4  200901BS70005      1/6/2021     Tuesday       Auto traffic signal


           Junction_Detail Accident_Severity   Latitude  \
0  T or staggered junction           Serious  51.512273
1               Crossroads           Serious  51.514399
2  T or staggered junction            Slight  51.486668
3  T or staggered junction           Serious  51.507804
4               Crossroads           Serious  51.482076

        Light_Conditions Local_Authority_(District)
Carriageway_Hazards  ...  \
0               Daylight     Kensington and Chelsea
NaN  ...
1               Daylight     Kensington and Chelsea
NaN  ...
2               Daylight     Kensington and Chelsea
NaN  ...
3               Daylight     Kensington and Chelsea
NaN  ...
4  Darkness - lights lit     Kensington and Chelsea
NaN  ...

   Number_of_Casualties  Number_of_Vehicles         Police_Force  \
0                     1                   2  Metropolitan Police
1                    11                   2  Metropolitan Police
2                     1                   2  Metropolitan Police
3                     1                   2  Metropolitan Police
4                     1                   2  Metropolitan Police
```

```
  Road_Surface_Conditions           Road_Type Speed_limit   Time  \
0                     Dry      One way street          30  15:11
1             Wet or damp  Single carriageway          30  10:59
2                     Dry  Single carriageway          30  14:19
3            Frost or ice  Single carriageway          30   8:10
4                     Dry  Single carriageway          30  17:25

  Urban_or_Rural_Area  Weather_Conditions           Vehicle_Type
0               Urban  Fine no high winds                    Car
1               Urban  Fine no high winds  Taxi/Private hire car
2               Urban  Fine no high winds  Taxi/Private hire car
3               Urban               Other  Motorcycle over 500cc
4               Urban  Fine no high winds                    Car

[5 rows x 21 columns]
```

```
df.tail()
```

```
       Accident_Index Accident Date Day_of_Week
Junction_Control  \
307968  201091NM01760     2/18/2022    Thursday  Data missing or out
of range
307969  201091NM01881     2/21/2022      Sunday  Data missing or out
of range
307970  201091NM01935     2/23/2022     Tuesday      Give way or
uncontrolled
307971  201091NM01964     2/23/2022     Tuesday      Give way or
uncontrolled
307972  201091NM02142     2/28/2022      Sunday      Give way or
uncontrolled

                           Junction_Detail Accident_Severity
Latitude  \
307968  Not at junction or within 20 metres            Slight
57.374005
307969  Not at junction or within 20 metres            Slight
57.232273
307970              T or staggered junction            Slight
57.585044
307971              T or staggered junction           Serious
57.214898
307972              T or staggered junction           Serious
57.575210

              Light_Conditions Local_Authority_(District)  \
307968                Daylight                   Highland
307969  Darkness - no lighting                   Highland
307970                Daylight                   Highland
307971  Darkness - no lighting                   Highland
```

```
307972                Daylight                    Highland

         Carriageway_Hazards  ...  Number_of_Casualties
Number_of_Vehicles  \
307968                   NaN  ...                     2
1
307969                   NaN  ...                     1
1
307970                   NaN  ...                     1
3
307971                   NaN  ...                     1
2
307972  Other object on road  ...                     1
1

        Police_Force Road_Surface_Conditions           Road_Type
Speed_limit  \
307968      Northern                     Dry  Single carriageway
60
307969      Northern            Frost or ice  Single carriageway
60
307970      Northern            Frost or ice  Single carriageway
30
307971      Northern             Wet or damp  Single carriageway
60
307972      Northern             Wet or damp    Dual carriageway
60

         Time Urban_or_Rural_Area     Weather_Conditions  \
307968   7:00               Rural     Fine no high winds
307969   3:00               Rural     Fine no high winds
307970   9:38               Rural     Fine no high winds
307971  18:25               Rural     Fine no high winds
307972  15:45               Rural  Snowing no high winds

                 Vehicle_Type
307968                    Car
307969                    Car
307970                    Car
307971  Motorcycle over 500cc
307972                    Car

[5 rows x 21 columns]

df.describe()

            Latitude      Longitude  Number_of_Casualties
Number_of_Vehicles  \
count  307973.000000  307973.000000         307973.000000
307973.000000
```

```
mean       52.487005      -1.368884              1.356882
1.829063
std         1.339011       1.356092              0.815857
0.710477
min        49.914488      -7.516225              1.000000
1.000000
25%        51.485248      -2.247937              1.000000
1.000000
50%        52.225943      -1.349258              1.000000
2.000000
75%        53.415517      -0.206810              1.000000
2.000000
max        60.598055       1.759398             48.000000
32.000000

         Speed_limit
count  307973.000000
mean       38.866037
std        14.032933
min        10.000000
25%        30.000000
50%        30.000000
75%        50.000000
max        70.000000

df.describe

<bound method NDFrame.describe of        Accident_Index Accident Date
Day_of_Week              Junction_Control  \
0       200901BS70001      1/1/2021    Thursday      Give way or
uncontrolled
1       200901BS70002      1/5/2021      Monday      Give way or
uncontrolled
2       200901BS70003      1/4/2021      Sunday      Give way or
uncontrolled
3       200901BS70004      1/5/2021      Monday           Auto traffic
signal
4       200901BS70005      1/6/2021     Tuesday           Auto traffic
signal
...                ...           ...         ...
...
307968  201091NM01760     2/18/2022    Thursday  Data missing or out
of range
307969  201091NM01881     2/21/2022      Sunday  Data missing or out
of range
307970  201091NM01935     2/23/2022     Tuesday      Give way or
uncontrolled
307971  201091NM01964     2/23/2022     Tuesday      Give way or
uncontrolled
307972  201091NM02142     2/28/2022      Sunday      Give way or
```

```
uncontrolled

                            Junction_Detail Accident_Severity
Latitude  \
0                    T or staggered junction           Serious
51.512273
1                                 Crossroads           Serious
51.514399
2                    T or staggered junction            Slight
51.486668
3                    T or staggered junction           Serious
51.507804
4                                 Crossroads           Serious
51.482076
...                                      ...               ...         .
..
307968  Not at junction or within 20 metres            Slight
57.374005
307969  Not at junction or within 20 metres            Slight
57.232273
307970               T or staggered junction            Slight
57.585044
307971               T or staggered junction           Serious
57.214898
307972               T or staggered junction           Serious
57.575210

              Light_Conditions Local_Authority_(District)  \
0                    Daylight     Kensington and Chelsea
1                    Daylight     Kensington and Chelsea
2                    Daylight     Kensington and Chelsea
3                    Daylight     Kensington and Chelsea
4        Darkness - lights lit     Kensington and Chelsea
...                        ...                        ...
307968               Daylight                  Highland
307969  Darkness - no lighting                  Highland
307970               Daylight                  Highland
307971  Darkness - no lighting                  Highland
307972               Daylight                  Highland

         Carriageway_Hazards  ...  Number_of_Casualties
Number_of_Vehicles  \
0                        NaN  ...                    1
2
1                        NaN  ...                   11
2
2                        NaN  ...                    1
2
3                        NaN  ...                    1
2
```

```
4                         NaN  ...                         1
2
...                       ...  ...                       ...
...
307968                    NaN  ...                         2
1
307969                    NaN  ...                         1
1
307970                    NaN  ...                         1
3
307971                    NaN  ...                         1
2
307972  Other object on road  ...                         1
1

               Police_Force Road_Surface_Conditions
Road_Type  \
0       Metropolitan Police                      Dry      One way
street
1       Metropolitan Police              Wet or damp  Single
carriageway
2       Metropolitan Police                      Dry  Single
carriageway
3       Metropolitan Police              Frost or ice  Single
carriageway
4       Metropolitan Police                      Dry  Single
carriageway
...                     ...                      ...                  ..
.
307968              Northern                     Dry  Single
carriageway
307969              Northern              Frost or ice  Single
carriageway
307970              Northern              Frost or ice  Single
carriageway
307971              Northern               Wet or damp  Single
carriageway
307972              Northern               Wet or damp    Dual
carriageway

       Speed_limit   Time Urban_or_Rural_Area
Weather_Conditions  \
0               30  15:11               Urban     Fine no high winds

1               30  10:59               Urban     Fine no high winds

2               30  14:19               Urban     Fine no high winds

3               30   8:10               Urban                  Other
```

```
4               30  17:25                Urban     Fine no high winds

...            ...    ...                  ...                     ...

307968          60   7:00                Rural     Fine no high winds

307969          60   3:00                Rural     Fine no high winds

307970          30   9:38                Rural     Fine no high winds

307971          60  18:25                Rural     Fine no high winds

307972          60  15:45                Rural  Snowing no high winds


                  Vehicle_Type
0                          Car
1       Taxi/Private hire car
2       Taxi/Private hire car
3       Motorcycle over 500cc
4                          Car
...                        ...
307968                     Car
307969                     Car
307970                     Car
307971  Motorcycle over 500cc
307972                     Car

[307973 rows x 21 columns]>

df.info

<bound method DataFrame.info of        Accident_Index Accident Date
Day_of_Week              Junction_Control  \
0       200901BS70001      1/1/2021    Thursday      Give way or
uncontrolled
1       200901BS70002      1/5/2021      Monday      Give way or
uncontrolled
2       200901BS70003      1/4/2021      Sunday      Give way or
uncontrolled
3       200901BS70004      1/5/2021      Monday           Auto traffic
signal
4       200901BS70005      1/6/2021     Tuesday           Auto traffic
signal
...               ...           ...         ...
...
307968  201091NM01760     2/18/2022    Thursday  Data missing or out
of range
307969  201091NM01881     2/21/2022      Sunday  Data missing or out
of range
307970  201091NM01935     2/23/2022     Tuesday      Give way or
```

```
uncontrolled
307971  201091NM01964     2/23/2022     Tuesday      Give way or
uncontrolled
307972  201091NM02142     2/28/2022      Sunday      Give way or
uncontrolled

                            Junction_Detail Accident_Severity
Latitude  \
0                    T or staggered junction           Serious
51.512273
1                                Crossroads            Serious
51.514399
2                    T or staggered junction            Slight
51.486668
3                    T or staggered junction           Serious
51.507804
4                                Crossroads            Serious
51.482076
...                                     ...               ...         .
..
307968  Not at junction or within 20 metres            Slight
57.374005
307969  Not at junction or within 20 metres            Slight
57.232273
307970               T or staggered junction            Slight
57.585044
307971               T or staggered junction           Serious
57.214898
307972               T or staggered junction           Serious
57.575210

              Light_Conditions Local_Authority_(District)  \
0                    Daylight     Kensington and Chelsea
1                    Daylight     Kensington and Chelsea
2                    Daylight     Kensington and Chelsea
3                    Daylight     Kensington and Chelsea
4        Darkness - lights lit     Kensington and Chelsea
...                        ...                        ...
307968               Daylight                  Highland
307969  Darkness - no lighting                  Highland
307970               Daylight                  Highland
307971  Darkness - no lighting                  Highland
307972               Daylight                  Highland

         Carriageway_Hazards  ...  Number_of_Casualties
Number_of_Vehicles  \
0                        NaN  ...                    1
2
1                        NaN  ...                   11
2
```

```
2                        NaN  ...                       1
2
3                        NaN  ...                       1
2
4                        NaN  ...                       1
2
...                      ...  ...                     ...
...
307968                   NaN  ...                       2
1
307969                   NaN  ...                       1
1
307970                   NaN  ...                       1
3
307971                   NaN  ...                       1
2
307972  Other object on road  ...                       1
1

                Police_Force Road_Surface_Conditions
Road_Type  \
0       Metropolitan Police                      Dry      One way
street
1       Metropolitan Police              Wet or damp  Single
carriageway
2       Metropolitan Police                      Dry  Single
carriageway
3       Metropolitan Police             Frost or ice  Single
carriageway
4       Metropolitan Police                      Dry  Single
carriageway
...                     ...                      ...                  ..
.
307968              Northern                     Dry  Single
carriageway
307969              Northern             Frost or ice  Single
carriageway
307970              Northern             Frost or ice  Single
carriageway
307971              Northern              Wet or damp  Single
carriageway
307972              Northern              Wet or damp    Dual
carriageway

       Speed_limit   Time Urban_or_Rural_Area
Weather_Conditions  \
0               30  15:11               Urban     Fine no high winds

1               30  10:59               Urban     Fine no high winds
```

```
2                30  14:19                Urban     Fine no high winds

3                30   8:10                Urban                  Other

4                30  17:25                Urban     Fine no high winds

...             ...    ...                  ...                    ...

307968           60   7:00                Rural     Fine no high winds

307969           60   3:00                Rural     Fine no high winds

307970           30   9:38                Rural     Fine no high winds

307971           60  18:25                Rural     Fine no high winds

307972           60  15:45                Rural  Snowing no high winds


                   Vehicle_Type
0                           Car
1       Taxi/Private hire car
2       Taxi/Private hire car
3       Motorcycle over 500cc
4                           Car
...                         ...
307968                      Car
307969                      Car
307970                      Car
307971  Motorcycle over 500cc
307972                      Car

[307973 rows x 21 columns]>

df.nunique()

Accident_Index                197644
Accident Date                    730
Day_of_Week                        7
Junction_Control                   7
Junction_Detail                    9
Accident_Severity                  4
Latitude                      264362
Light_Conditions                   5
Local_Authority_(District)       422
Carriageway_Hazards                5
Longitude                     269856
Number_of_Casualties              28
Number_of_Vehicles                17
Police_Force                      51
Road_Surface_Conditions            5
```

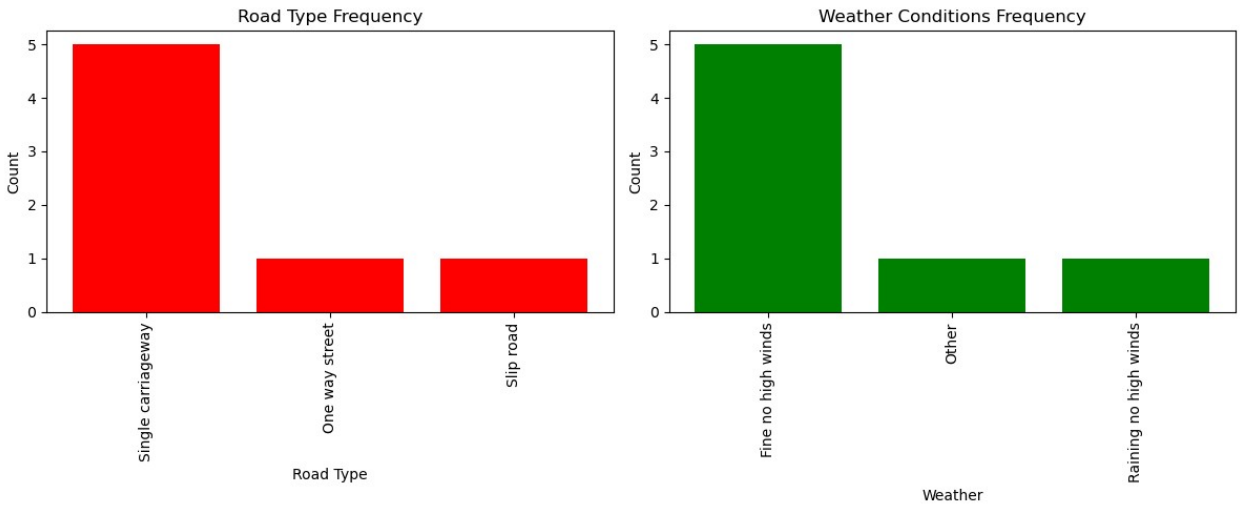
```
Road_Type                          5
Speed_limit                        8
Time                            1439
Urban_or_Rural_Area                2
Weather_Conditions                 8
Vehicle_Type                      15
dtype: int64
```

```
df.isnull().sum()
```

```
Accident_Index                     0
Accident Date                      0
Day_of_Week                        0
Junction_Control                   0
Junction_Detail                    0
Accident_Severity                  0
Latitude                           0
Light_Conditions                   0
Local_Authority_(District)         0
Carriageway_Hazards           302549
Longitude                          0
Number_of_Casualties               0
Number_of_Vehicles                 0
Police_Force                       0
Road_Surface_Conditions          317
Road_Type                       1534
Speed_limit                        0
Time                              17
Urban_or_Rural_Area                0
Weather_Conditions              6057
Vehicle_Type                       0
dtype: int64
```

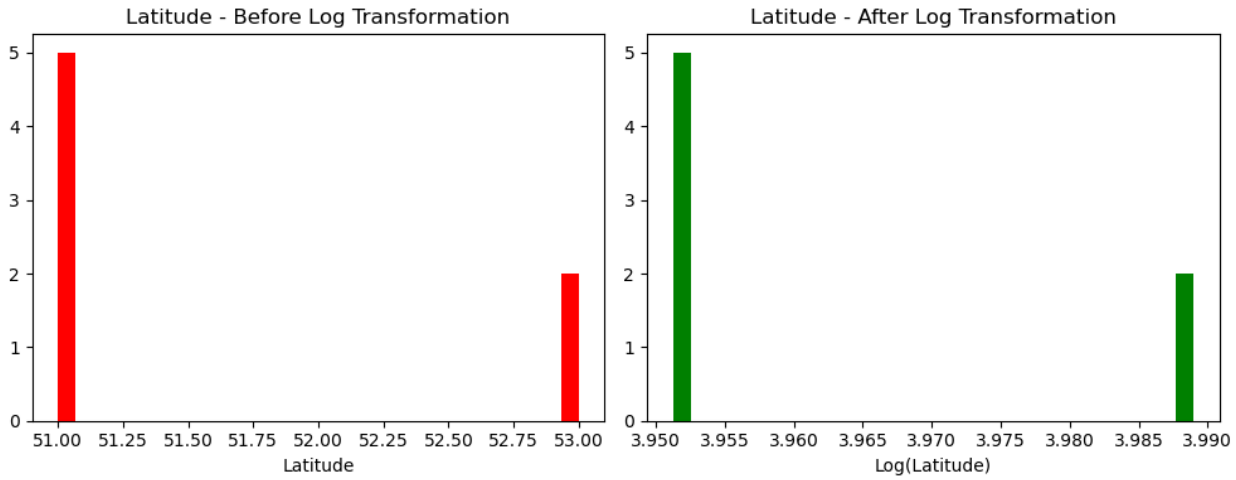
```
import matplotlib.pyplot as plt
# Count frequencies
road_type_counts = df['Road_Type'].value_counts()
weather_counts = df['Weather_Conditions'].value_counts()

# Plotting
plt.figure(figsize=(12, 5))

# Road Type Frequency
plt.subplot(1, 2, 1)
plt.bar(road_type_counts.index, road_type_counts.values, color='red')
plt.title("Road Type Frequency")
plt.xlabel("Road Type")
plt.ylabel("Count")
plt.xticks(rotation=90)

# Weather Conditions Frequency
```

```
plt.subplot(1, 2, 2)
plt.bar(weather_counts.index, weather_counts.values, color='green')
plt.title("Weather Conditions Frequency")
plt.xlabel("Weather")
plt.ylabel("Count")
plt.xticks(rotation=90)

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

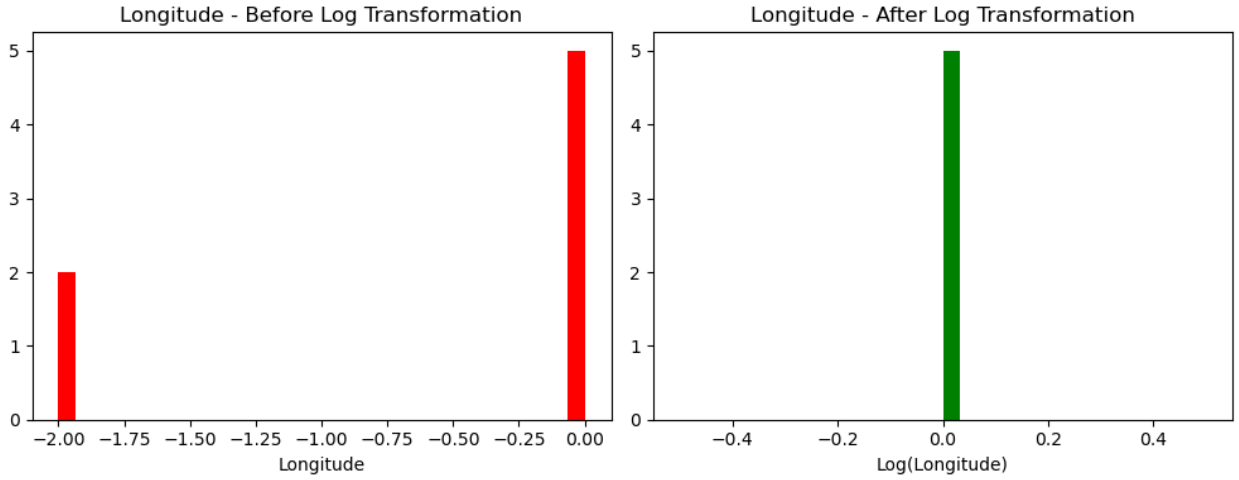


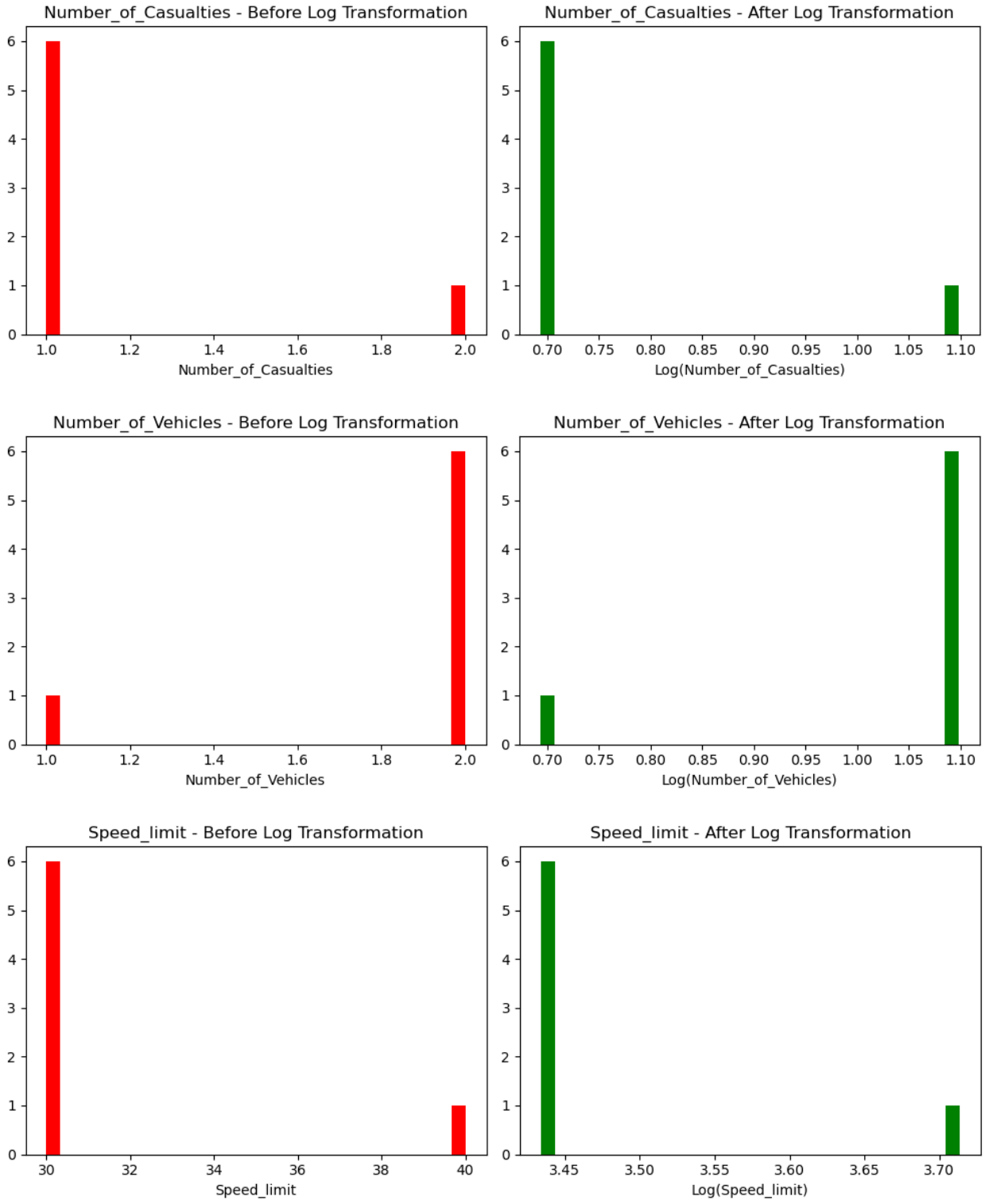

```
import matplotlib.pyplot as plt
import numpy as np

# Select numerical columns
numerical_columns = df.select_dtypes(include=['number']).columns

# Loop through each numerical column
for col in numerical_columns:
    plt.figure(figsize=(10, 4))

    # Original distribution
    plt.subplot(1, 2, 1)
    plt.hist(df[col].dropna(), bins=30, color='red')
    plt.title(f"{col} - Before Log Transformation")
    plt.xlabel(col)

    # Log-transformed distribution
    log_values = np.log(df[col] + 1)
    plt.subplot(1, 2, 2)
    plt.hist(log_values.dropna(), bins=30, color='green')
    plt.title(f"{col} - After Log Transformation")
    plt.xlabel(f"Log({col})")
```

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



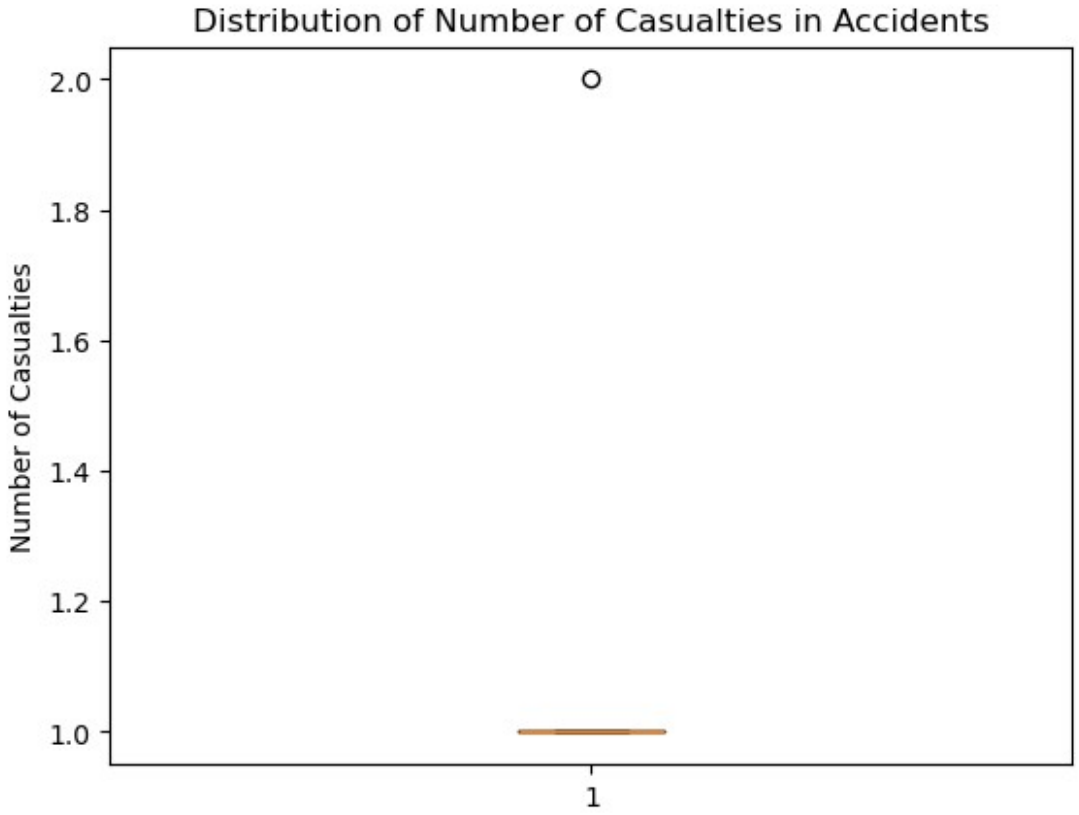

```
C:\Users\abhin\anaconda3\Lib\site-packages\pandas\core\
arraylike.py:399: RuntimeWarning: invalid value encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```



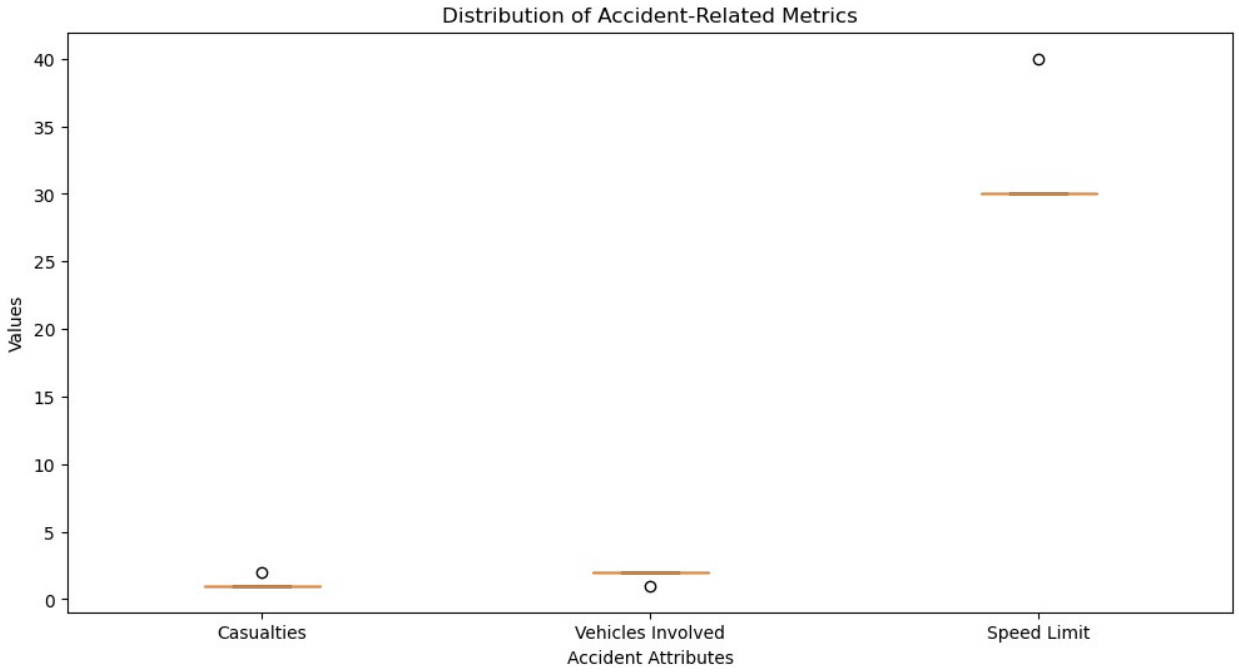



```
data = df['Number_of_Casualties'].dropna()
plt.boxplot(data)
plt.ylabel('Number of Casualties')
```

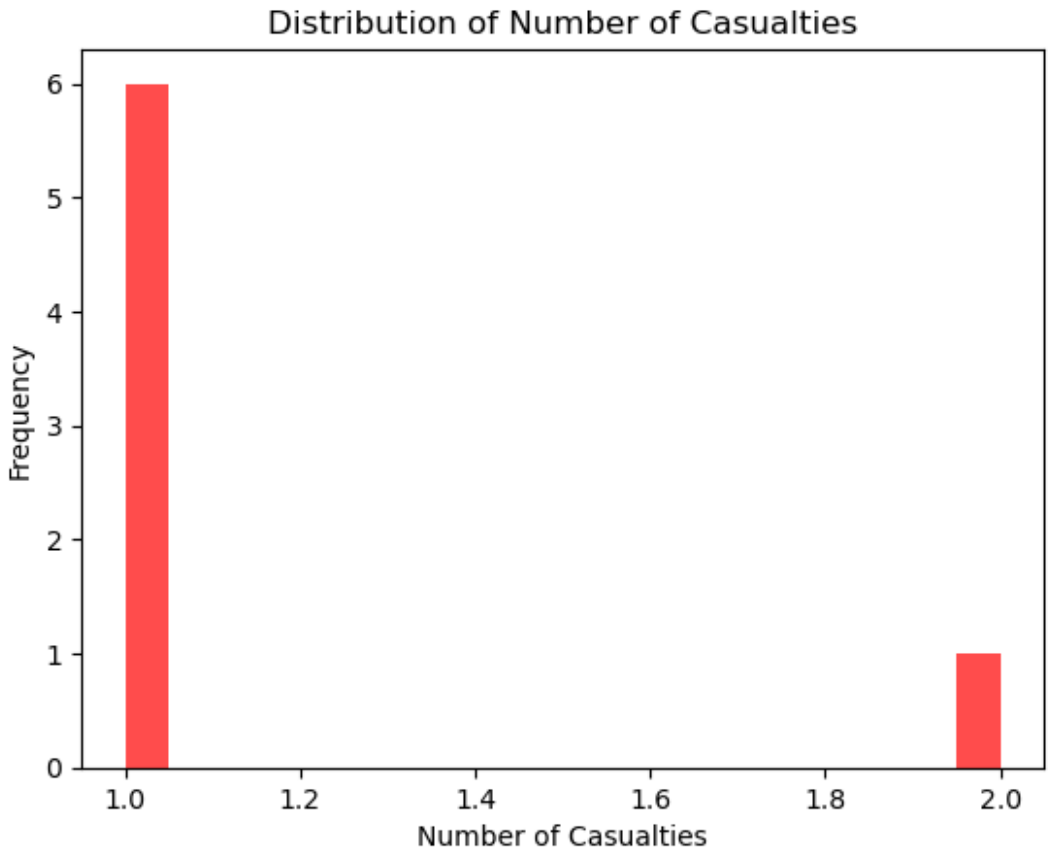
```
plt.title('Distribution of Number of Casualties in Accidents')
plt.show()
```

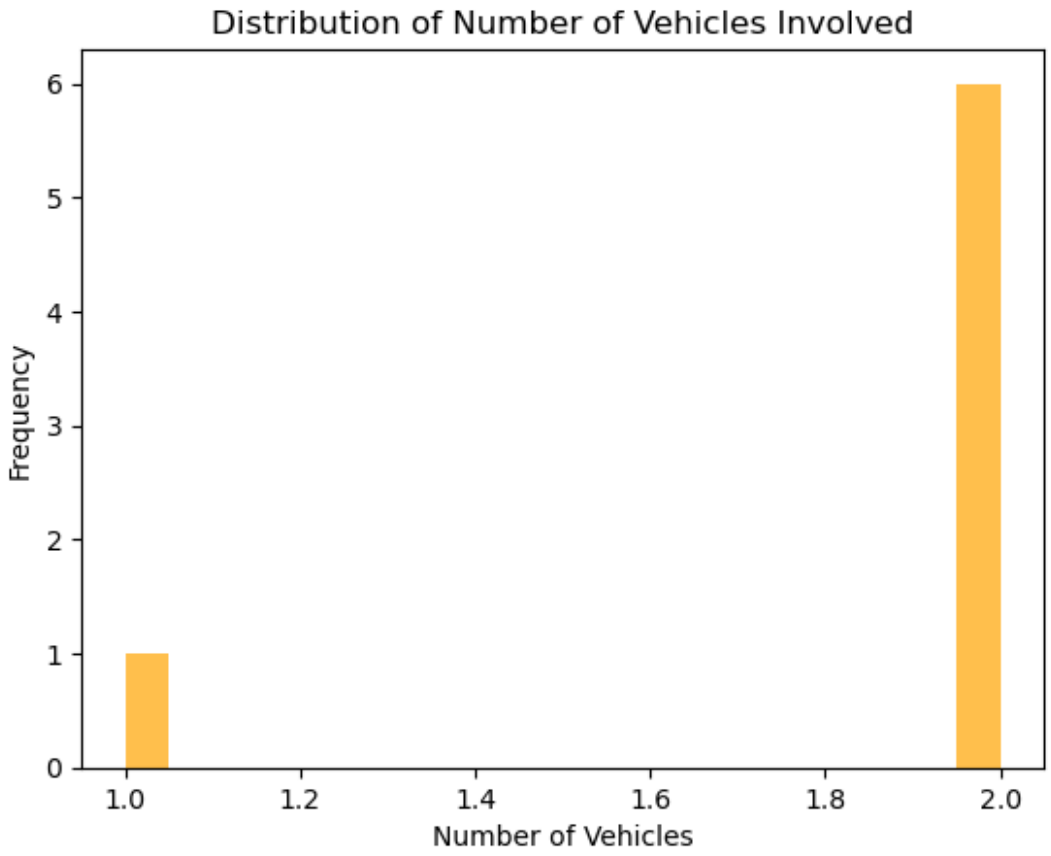



```
import matplotlib.pyplot as plt

# Prepare data from relevant numeric columns
data = [
    df['Number_of_Casualties'].dropna(),
    df['Number_of_Vehicles'].dropna(),
    df['Speed_limit'].dropna()
]

# Plotting
plt.figure(figsize=(12, 6))
plt.boxplot(data, labels=['Casualties', 'Vehicles Involved', 'Speed
Limit'], patch_artist=True)
plt.xlabel('Accident Attributes')
plt.ylabel('Values')
plt.title('Distribution of Accident-Related Metrics')
plt.show()
```

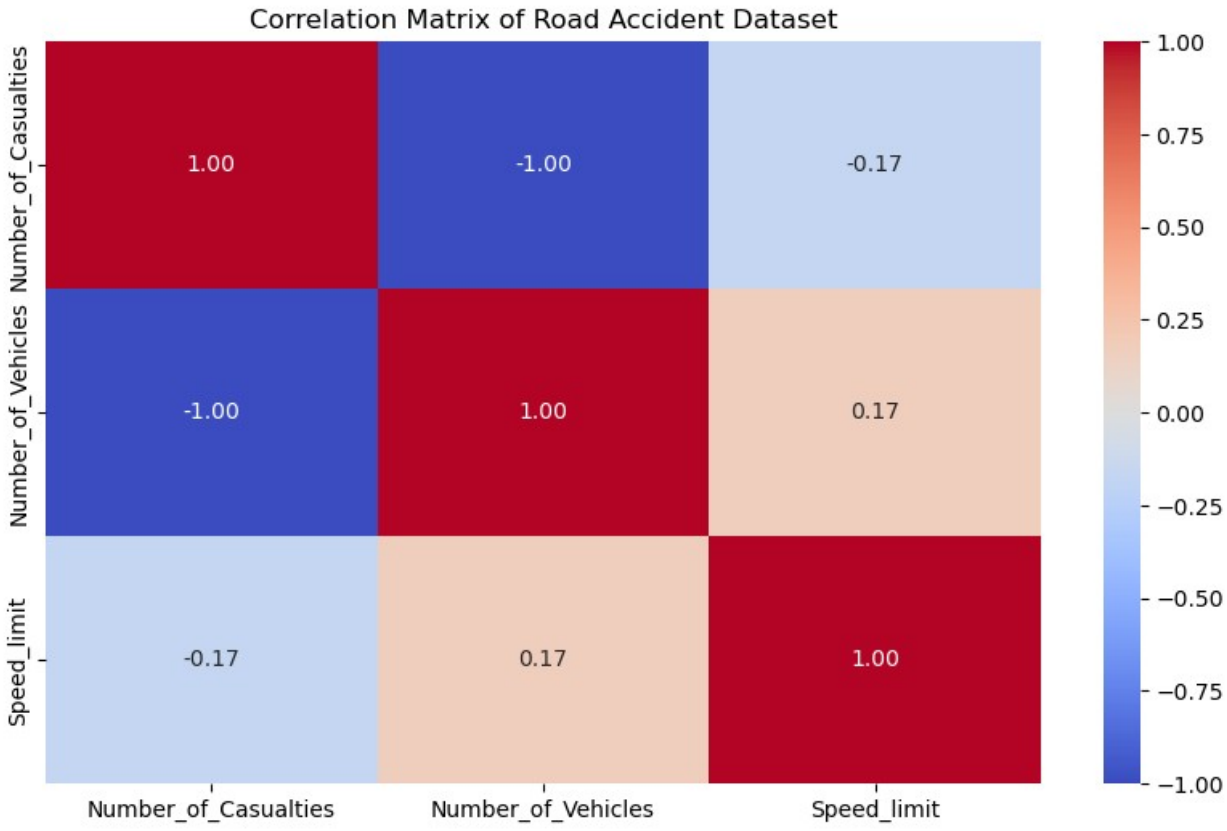



```
import seaborn as sns

import matplotlib.pyplot as plt

# Histogram for Number of Casualties
plt.hist(df['Number_of_Casualties'].dropna(), bins=20, color='red',
alpha=0.7)
plt.title('Distribution of Number of Casualties')
plt.xlabel('Number of Casualties')
plt.ylabel('Frequency')
plt.show()

# Histogram for Number of Vehicles
plt.hist(df['Number_of_Vehicles'].dropna(), bins=20, color='orange',
alpha=0.7)
plt.title('Distribution of Number of Vehicles Involved')
plt.xlabel('Number of Vehicles')
plt.ylabel('Frequency')
plt.show()
```

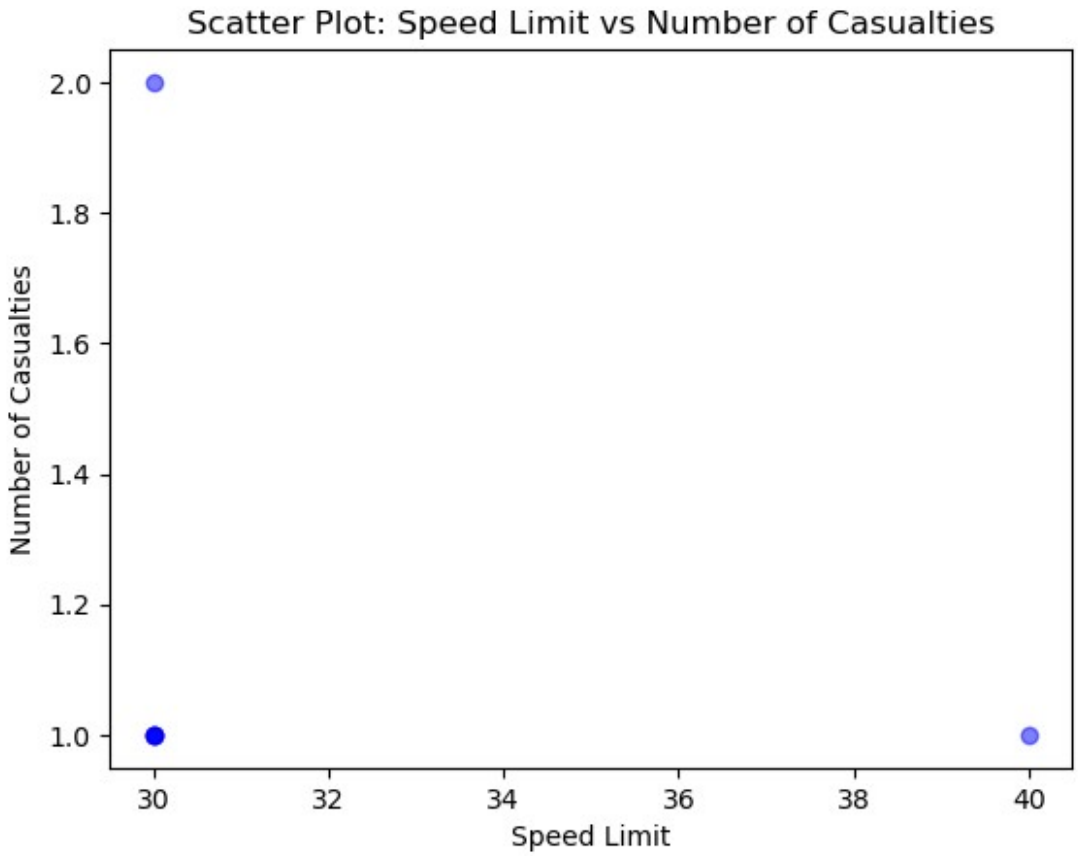

Distribution of Number of Casualties
Frequency
Number of Casualties
0
1
2
3
4
5
6
1.0
1.2
1.4
1.6
1.8
2.0



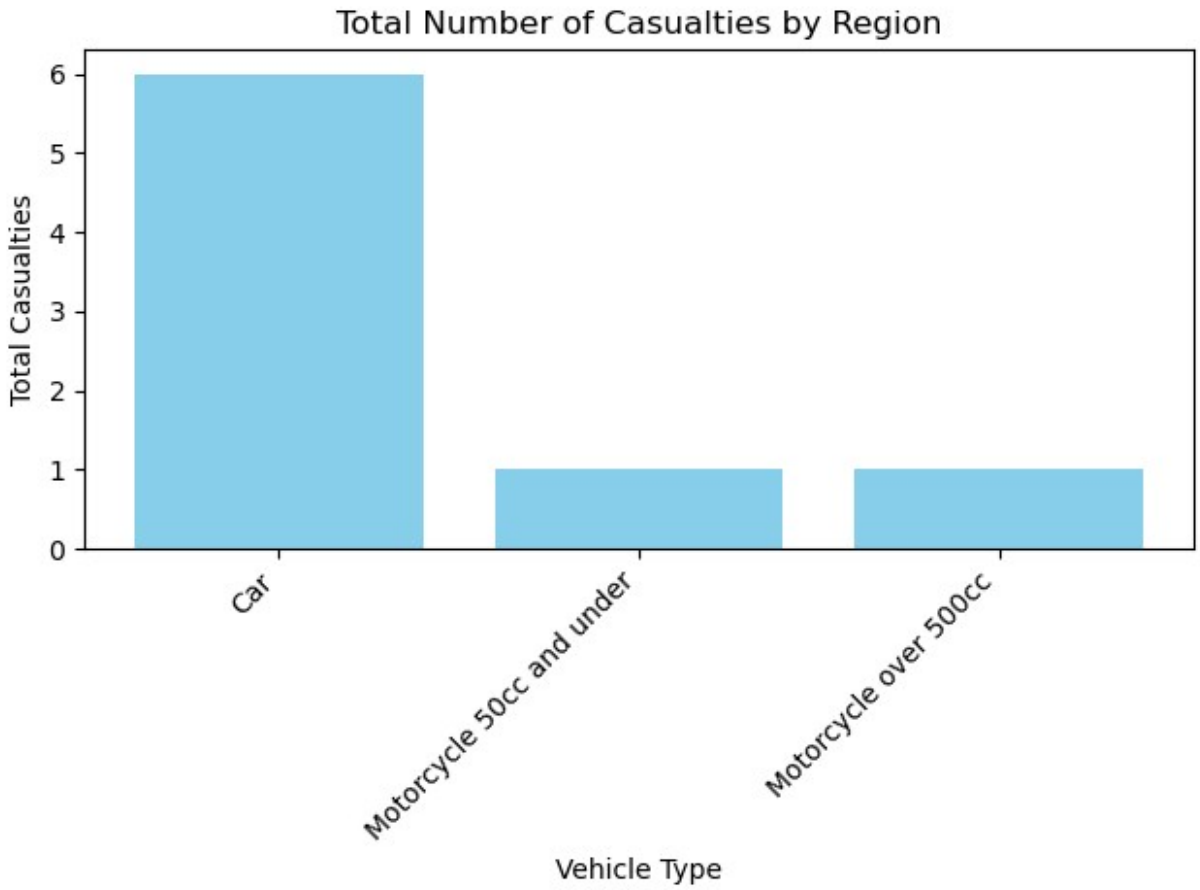

```
import seaborn as sns
import matplotlib.pyplot as plt

# Select numerical columns
numerical_cols = df.select_dtypes(include=['float64',
'int64']).columns

# Compute correlation matrix
corr_matrix = df[numerical_cols].corr()

# Plot heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix of Road Accident Dataset')
plt.show()
```



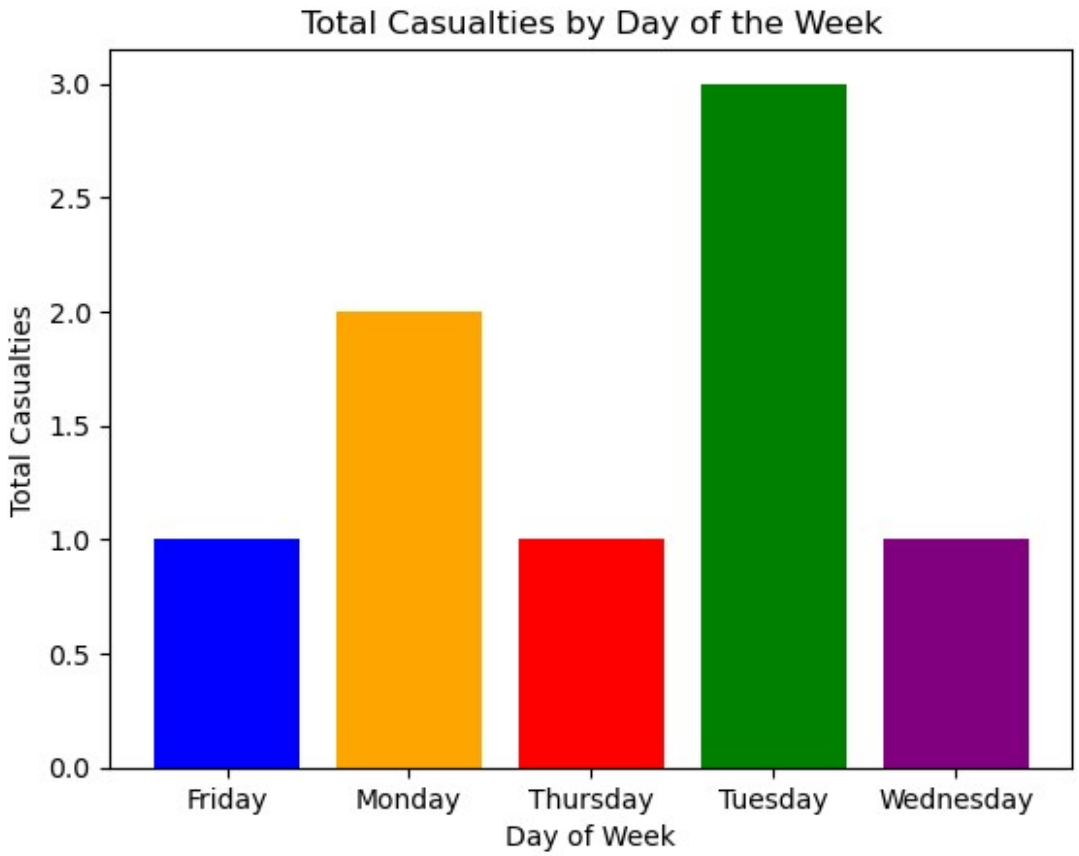

```
import matplotlib.pyplot as plt

plt.scatter(df['Speed_limit'], df['Number_of_Casualties'],
color='blue', alpha=0.5)
plt.title('Scatter Plot: Speed Limit vs Number of Casualties')
plt.xlabel('Speed Limit')
plt.ylabel('Number of Casualties')
plt.show()
```



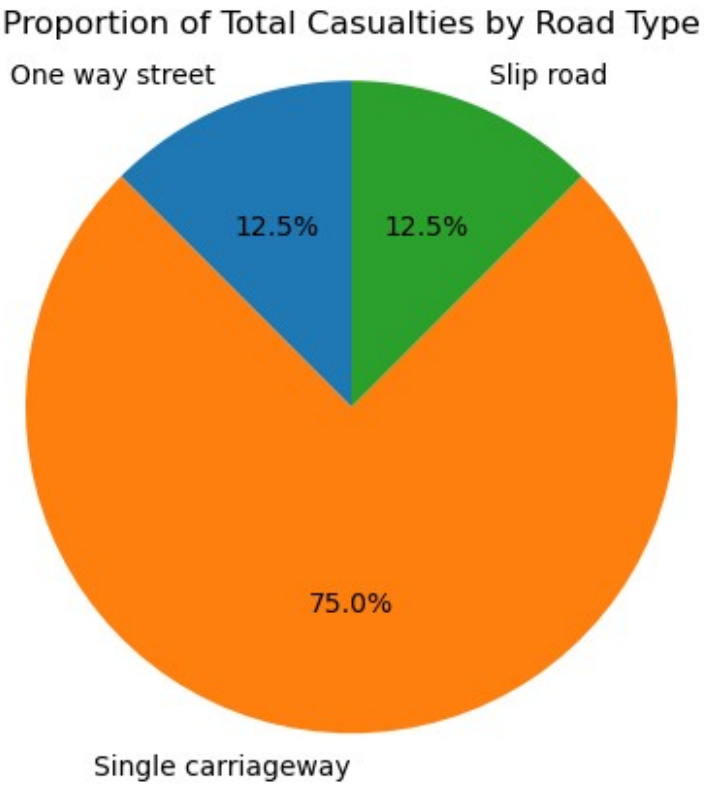

```
import matplotlib.pyplot as plt

# Total casualties grouped by region
casualties_by_region = df.groupby('Vehicle_Type')
['Number_of_Casualties'].sum()

# Plot
plt.bar(casualties_by_region.index, casualties_by_region.values,
color='skyblue')
plt.title('Total Number of Casualties by Region')
plt.xlabel('Vehicle Type')
plt.ylabel('Total Casualties')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



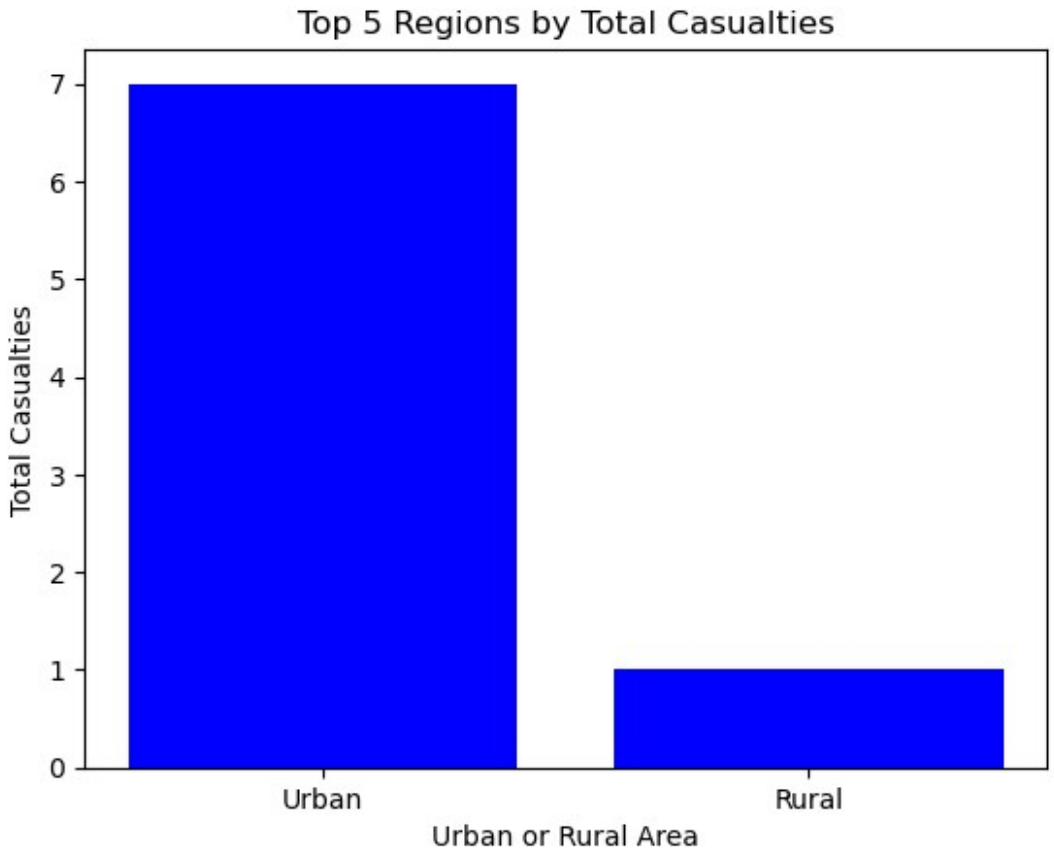

```
import matplotlib.pyplot as plt

# Grouping total casualties by day of the week
casualties_by_day = df.groupby('Day_of_Week')
['Number_of_Casualties'].sum()

# Plot
plt.bar(casualties_by_day.index, casualties_by_day.values,
color=['blue', 'orange', 'red', 'green', 'purple', 'brown', 'gray'])
plt.title('Total Casualties by Day of the Week')
plt.xlabel('Day of Week')
plt.ylabel('Total Casualties')
plt.show()
```



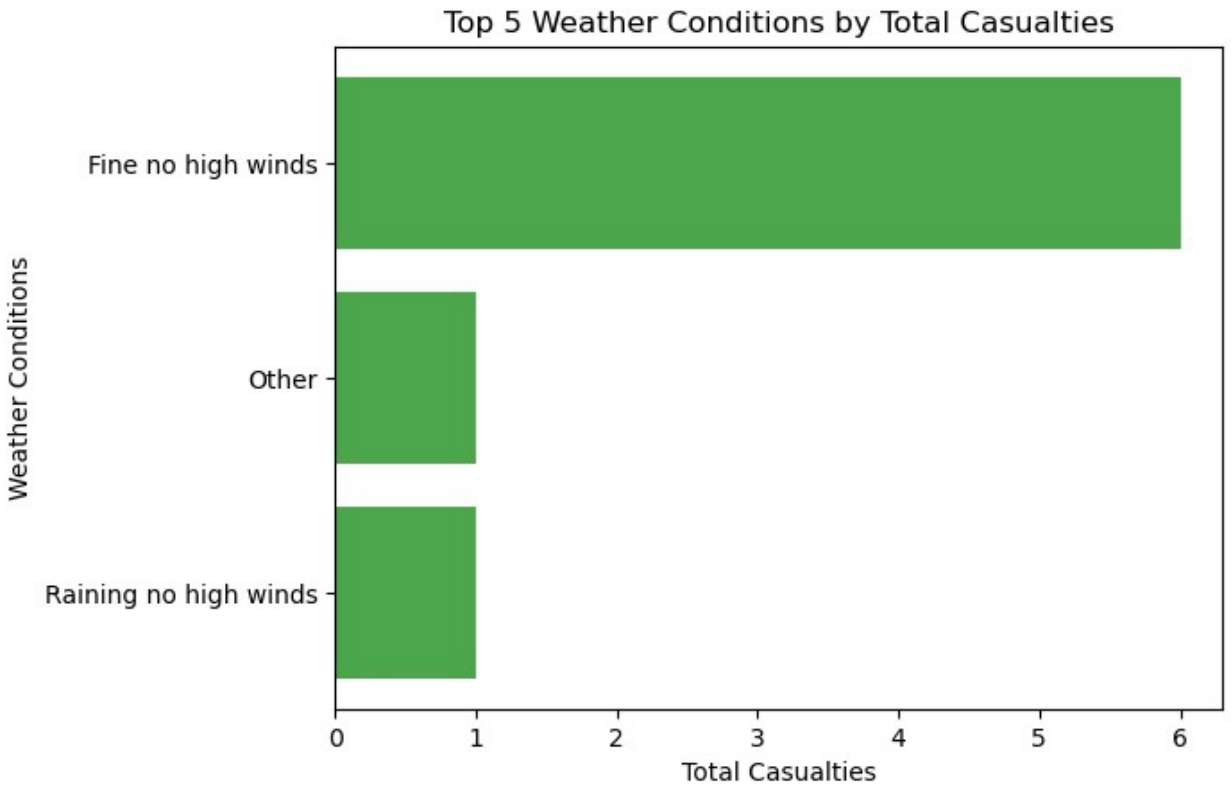

```
import matplotlib.pyplot as plt

# Group by road type and sum casualties
casualties_by_road_type = df.groupby('Road_Type')
['Number_of_Casualties'].sum()

# Plot pie chart
plt.pie(casualties_by_road_type.values,
        labels=casualties_by_road_type.index,
        autopct='%1.1f%%',
        startangle=90)
plt.title('Proportion of Total Casualties by Road Type')
plt.axis('equal')
plt.show()
```



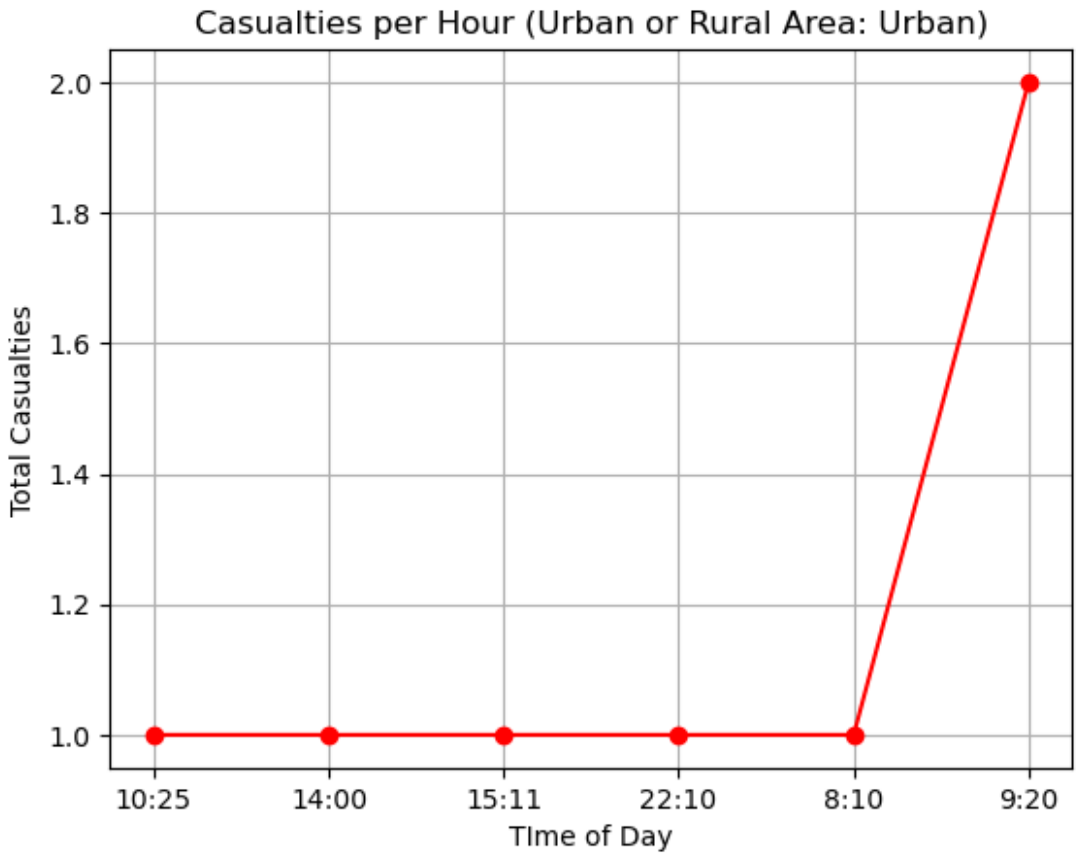

```
import matplotlib.pyplot as plt

# Top 5 regions by total casualties
top_regions = df.groupby('Urban_or_Rural_Area')
['Number_of_Casualties'].sum().sort_values(ascending=False).head(5)

# Plot
plt.bar(top_regions.index, top_regions.values, color='blue')
plt.xlabel('Urban or Rural Area')
plt.ylabel('Total Casualties')
plt.title('Top 5 Regions by Total Casualties')
plt.show()
```



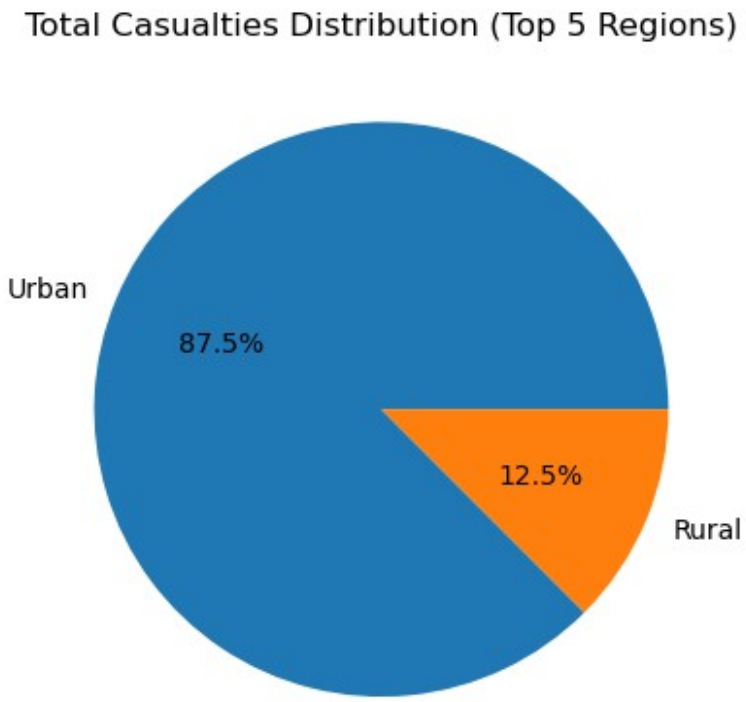

```
import matplotlib.pyplot as plt

# Top 5 weather conditions by total casualties
top_weather_conditions = df.groupby('Weather_Conditions')
['Number_of_Casualties'].sum().sort_values(ascending=False).head(5)

# Horizontal bar plot
plt.barh(top_weather_conditions.index, top_weather_conditions.values,
color='green', alpha=0.7)
plt.xlabel("Total Casualties")
plt.ylabel("Weather Conditions")
plt.title("Top 5 Weather Conditions by Total Casualties")
plt.gca().invert_yaxis()  # Highest at the top
plt.show()
```



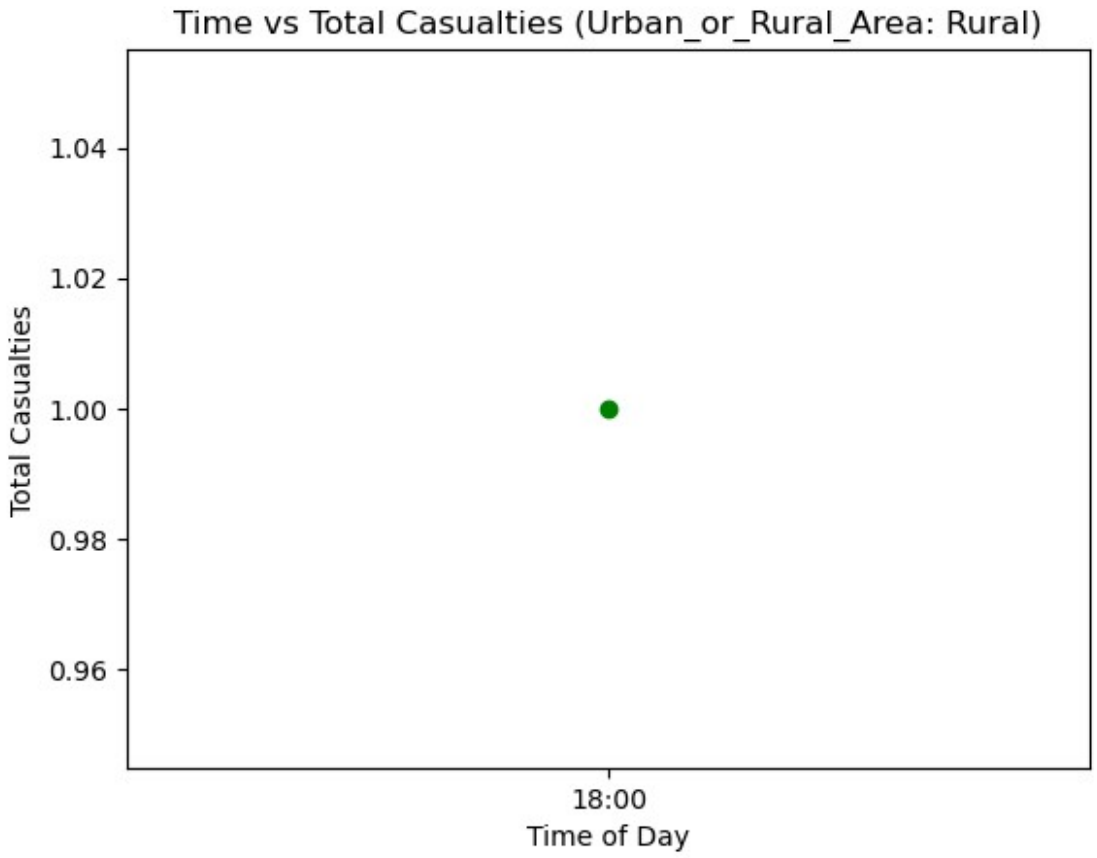

```
import matplotlib.pyplot as plt

# Filter for a specific region (e.g., London)
accident_data = df[df['Urban_or_Rural_Area'] == 'Urban']

# Group by hour and sum casualties
casualties_per_hour = accident_data.groupby('Time')
['Number_of_Casualties'].sum()

# Line plot
plt.plot(casualties_per_hour.index, casualties_per_hour.values,
color='red', marker='o')
plt.xlabel('TIme of Day')
plt.ylabel('Total Casualties')
plt.title('Casualties per Hour (Urban or Rural Area: Urban)')
plt.grid(True)
plt.show()
```



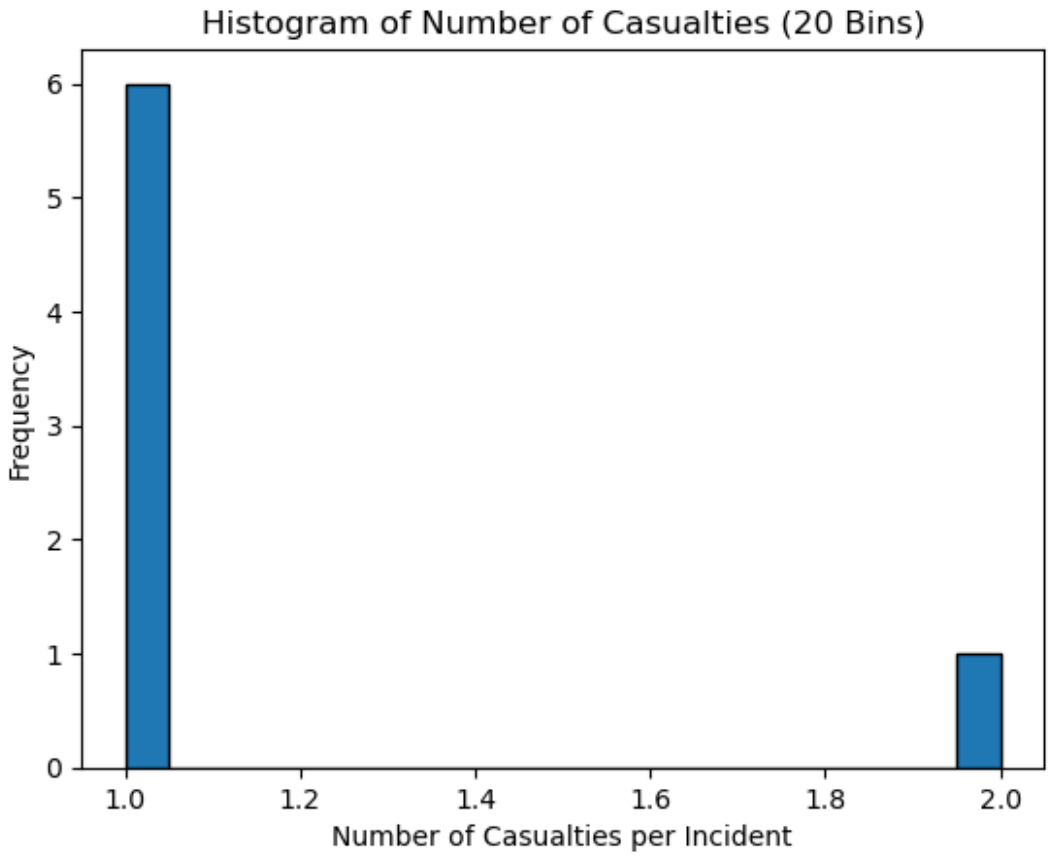

```
import matplotlib.pyplot as plt

# Top 5 regions by total casualties
top_regions = df.groupby('Urban_or_Rural_Area')
['Number_of_Casualties'].sum().sort_values(ascending=False).head(5)

# Pie chart
plt.pie(top_regions.values, labels=top_regions.index, autopct='%1.1f%
%')
plt.title('Total Casualties Distribution (Top 5 Regions)')
plt.show()
```



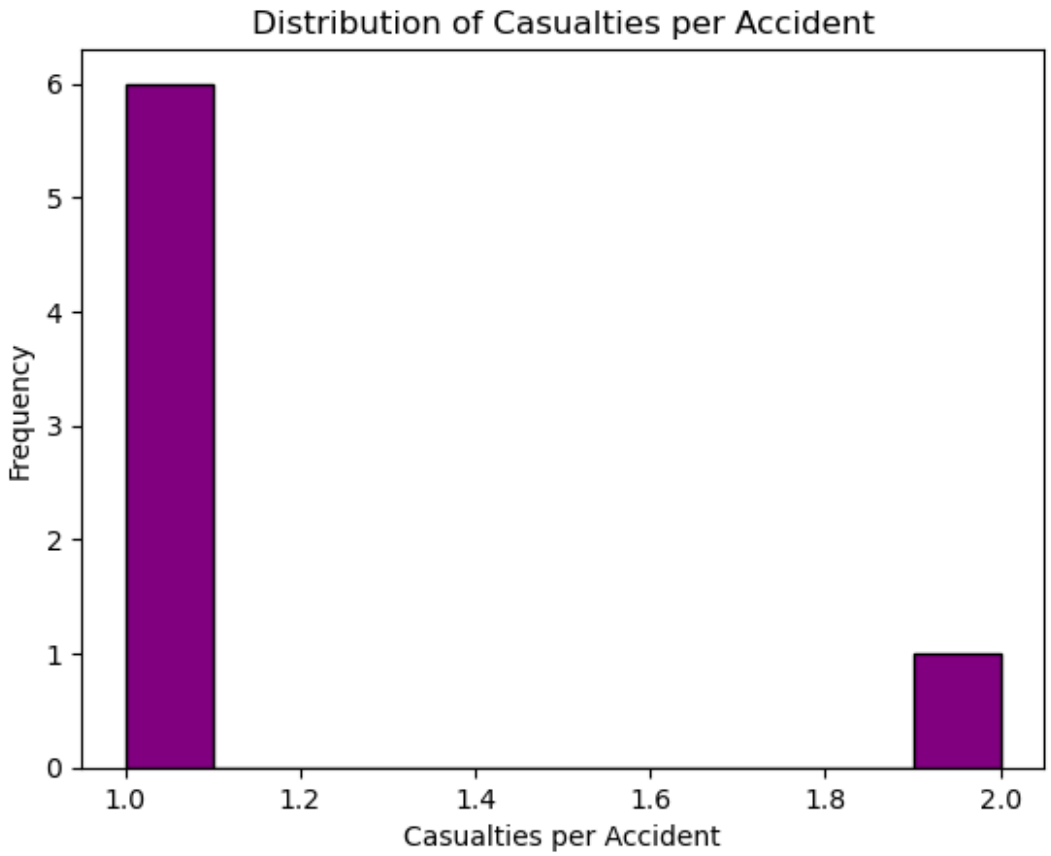

```
import matplotlib.pyplot as plt

# Filter for a specific region
accident_data = df[df['Urban_or_Rural_Area'] == 'Rural']

# Group by hour and sum casualties
casualties_per_hour = accident_data.groupby('Time')
['Number_of_Casualties'].sum()

# Scatter plot
plt.scatter(casualties_per_hour.index, casualties_per_hour.values,
color='green')
plt.xlabel('Time of Day')
plt.ylabel('Total Casualties')
plt.title('Time vs Total Casualties (Urban_or_Rural_Area: Rural)')
plt.show()
```




```
import matplotlib.pyplot as plt

# Histogram of number of casualties with 20 bins
plt.hist(df['Number_of_Casualties'], bins=20, edgecolor='black')
plt.xlabel('Number of Casualties per Incident')
plt.ylabel('Frequency')
plt.title('Histogram of Number of Casualties (20 Bins)')
plt.show()
```




```
import matplotlib.pyplot as plt

# Histogram for casualties per accident
plt.hist(df['Number_of_Casualties'], bins=10, color='purple',
edgecolor='black')
plt.xlabel('Casualties per Accident')
plt.ylabel('Frequency')
plt.title('Distribution of Casualties per Accident')
plt.show()
```

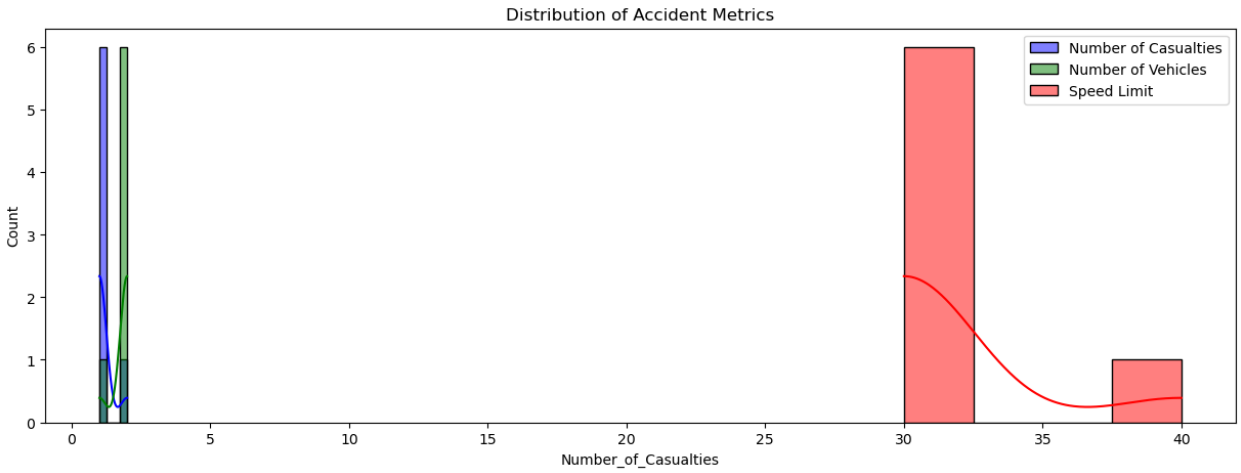


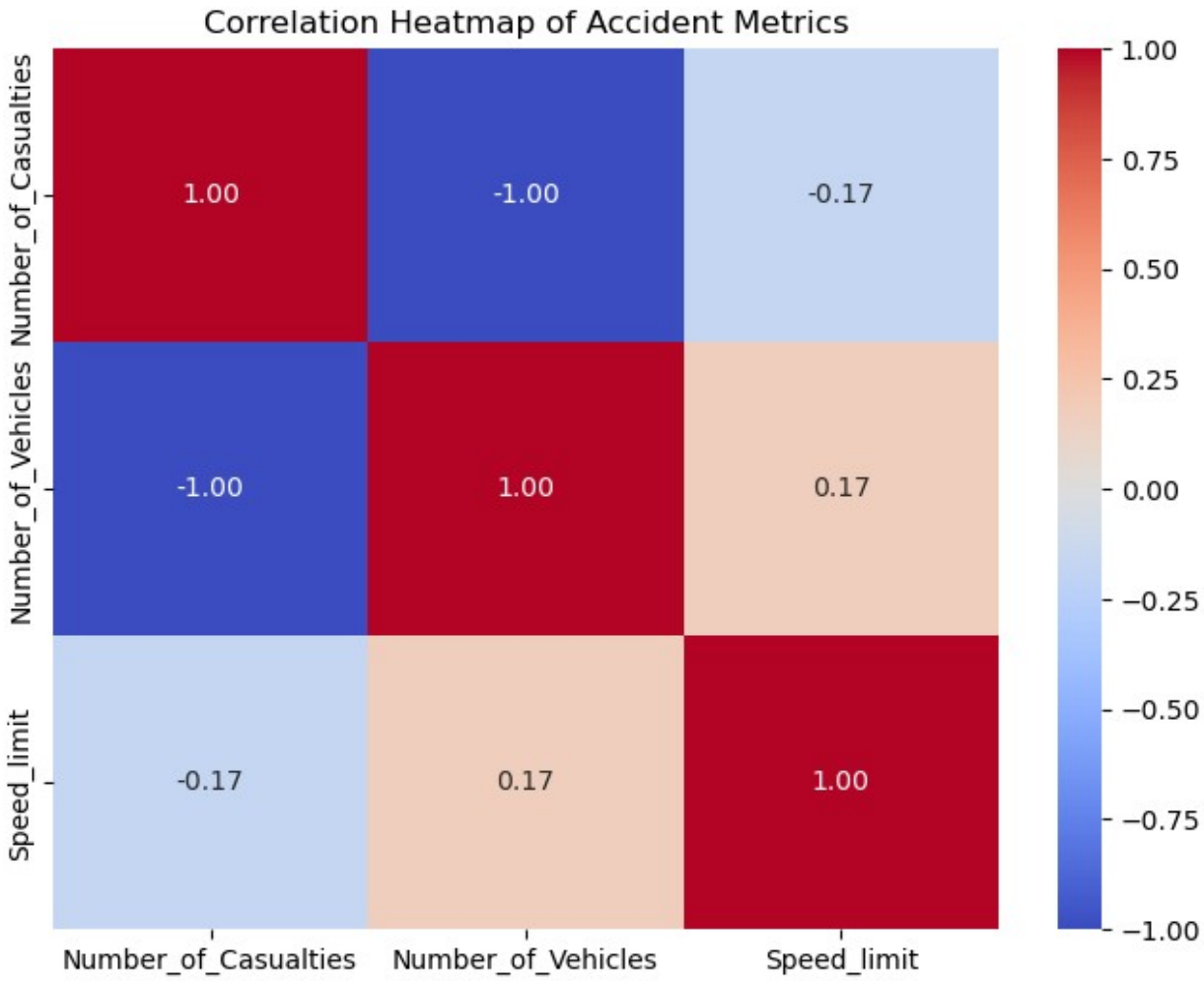

```
import seaborn as sns
import matplotlib.pyplot as plt

# Missing values and statistics
print("Missing values:\n", df.isnull().sum())
print("\nSummary Statistics:\n", df.describe())

# Histogram distributions
plt.figure(figsize=(15, 5))
sns.histplot(df['Number_of_Casualties'], kde=True, color='blue',
label='Number of Casualties')
sns.histplot(df['Number_of_Vehicles'], kde=True, color='green',
label='Number of Vehicles')
sns.histplot(df['Speed_limit'], kde=True, color='red', label='Speed
Limit')
plt.legend()
plt.title('Distribution of Accident Metrics')
plt.show()

# Correlation heatmap
plt.figure(figsize=(8, 6))
correlation_matrix = df[['Number_of_Casualties', 'Number_of_Vehicles',
'Speed_limit']].corr()
```

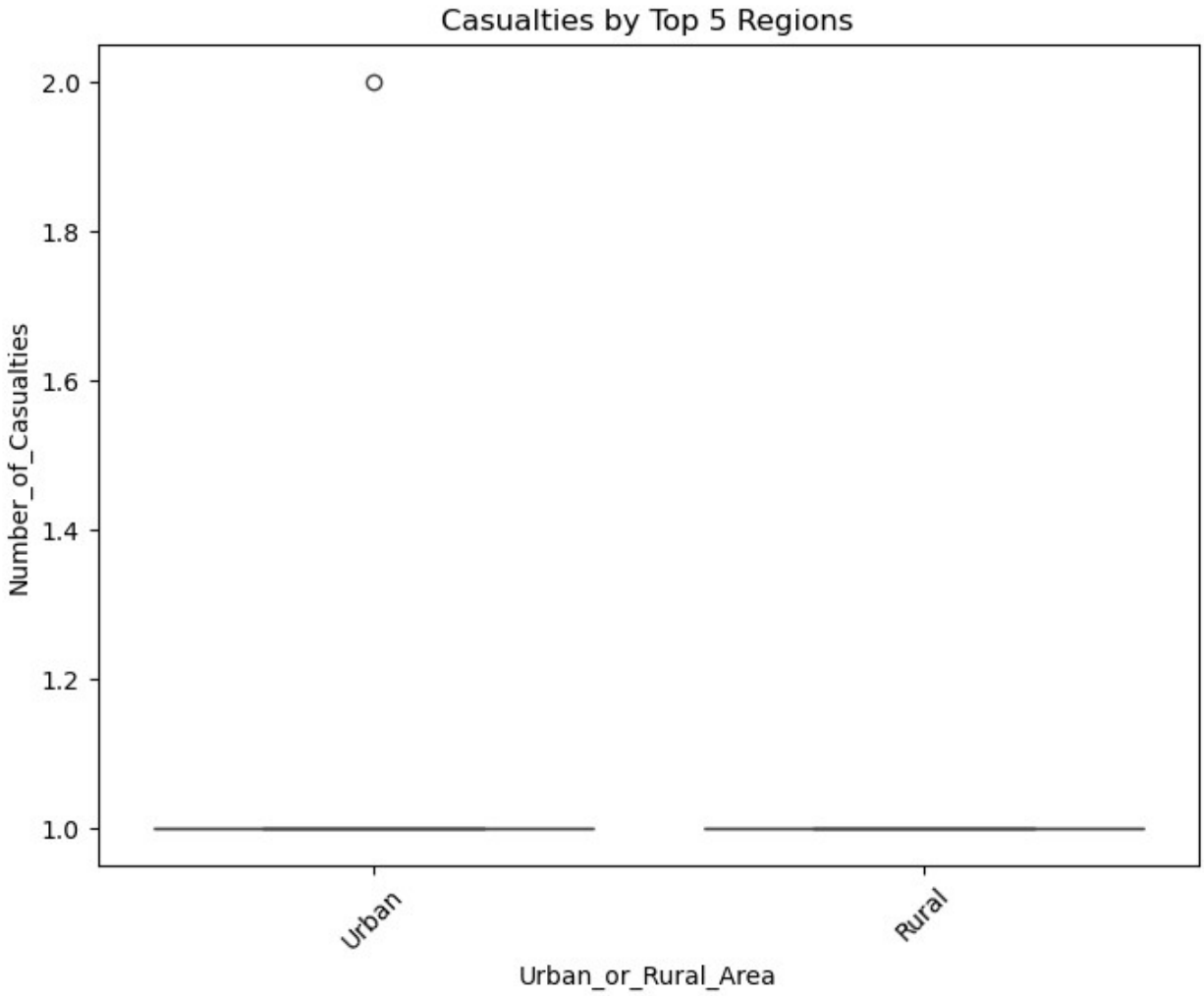
```
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
fmt='.2f')
plt.title('Correlation Heatmap of Accident Metrics')
plt.show()

# Top 5 regions by frequency
top_regions = df['Urban_or_Rural_Area'].value_counts().head(5).index
plt.figure(figsize=(8, 6))
sns.boxplot(x='Urban_or_Rural_Area', y='Number_of_Casualties',
data=df[df['Urban_or_Rural_Area'].isin(top_regions)])
plt.title('Casualties by Top 5 Regions')
plt.xticks(rotation=45)
plt.show()

# Top 5 road types by frequency
top_road_types = df['Road_Type'].value_counts().head(5).index
plt.figure(figsize=(10, 6))
sns.boxplot(x='Road_Type', y='Number_of_Casualties',
data=df[df['Road_Type'].isin(top_road_types)])
plt.xticks(rotation=45)
plt.title('Casualties by Top 5 Road Types')
plt.show()

Missing values:
 Junction_Control               0
Accident_Index                  0
Accident Date                   0
Accident_Severity               0
Latitude                        0
Light_Conditions                0
Local_Authority_(District)      0
Carriageway_Hazards             7
Longitude                       0
Number_of_Casualties            0
Number_of_Vehicles              0
Police_Force                    0
Road_Surface_Conditions         0
Road_Type                       0
Speed_limit                     0
Time                            0
Urban_or_Rural_Area             0
Weather_Conditions              0
Vehicle_Type                    0
dtype: int64

Summary Statistics:
         Latitude  Longitude  Number_of_Casualties  Number_of_Vehicles
\
count   7.000000   7.000000              7.000000            7.000000
```

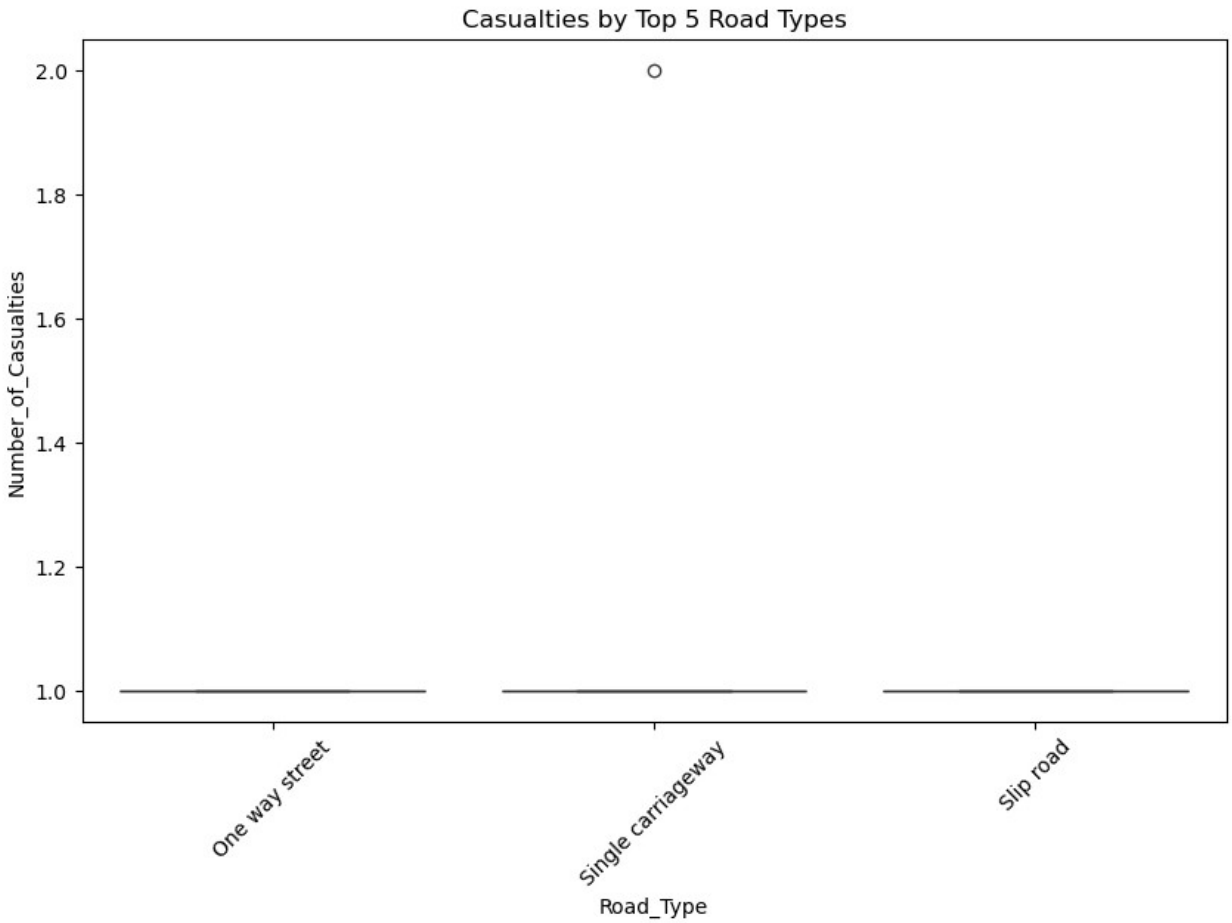
mean	51.571429	-0.571429	1.142857	1.857143
std	0.975900	0.975900	0.377964	0.377964
min	51.000000	-2.000000	1.000000	1.000000
25%	51.000000	-1.000000	1.000000	2.000000
50%	51.000000	0.000000	1.000000	2.000000
75%	52.000000	0.000000	1.000000	2.000000
max	53.000000	0.000000	2.000000	2.000000

	Speed_limit
count	7.000000
mean	31.428571
std	3.779645
min	30.000000
25%	30.000000
50%	30.000000
75%	30.000000
max	40.000000




Correlation Heatmap of Accident Metrics
Number_of_Casualties
Number_of_Vehicles
Speed_limit
1.00
-1.00
-0.17
-1.00
1.00
0.17
-0.17
0.17
1.00
Number_of_Casualties
Number_of_Vehicles
Speed_limit
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00


Casualties by Top 5 Regions
Number_of_Casualties
2.0
1.8
1.6
1.4
1.2
1.0
Urban
Rural
Urban_or_Rural_Area




```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
X = df[['Speed_limit', 'Number_of_Vehicles', 'Number_of_Casualties']]
y = df['Latitude']
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
print("X_train shape:", X_train.shape)
print("X_test shape:", X_test.shape)
print("y_train shape:", y_train.shape)
print("y_test shape:", y_test.shape)
```

```
X_train shape: (5, 3)
X_test shape: (2, 3)
y_train shape: (5,)
y_test shape: (2,)
```

```
model = LogisticRegression()
model.fit(X_train, y_train)
```

```
LogisticRegression()
```

```
y_pred = model.predict(X_test)
```

```
y_pred
```

```
array([51, 51])
```

```
y_pred1=model.predict(X)
```

```
y_pred
```

```
array([51, 51])
```

```
df['Prediction']=y_pred1
```

```
C:\Users\abhin\AppData\Local\Temp\ipykernel_24240\2454872504.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df['Prediction']=y_pred1
```

```
df
```

```


Junction_Control  \
Day_of_Week Junction_Detail

Thursday    T or staggered junction                           Give way
or uncontrolled
Monday      T or staggered junction                             Auto
traffic signal
            Not at junction or within 20 metres         Data missing
or out of range
Tuesday     Crossroads
Authorised person
Wednesday   T or staggered junction
Stop sign
Tuesday     Not at junction or within 20 metres  Not at junction or
within 20 metres
Friday      Other junction
Auto traffic sigl

                                                 Accident_Index
Accident Date  \
Day_of_Week Junction_Detail

Thursday    T or staggered junction               200901BS70001
```

```
1/1/2021
Monday      T or staggered junction               200901BS70004
1/5/2021
            Not at junction or within 20 metres   200901BS70015
1/12/2021
Tuesday     Crossroads                            200901BS70469
7/21/2021
Wednesday   T or staggered junction               200901CW11182
6/17/2021
Tuesday     Not at junction or within 20 metres   200906A008293
1/6/2021
Friday      Other junction                             2.01E+12
6/19/2021

                                                 Accident_Severity
Latitude  \
Day_of_Week Junction_Detail

Thursday    T or staggered junction                           Se
51
Monday      T or staggered junction                           Se
51
            Not at junction or within 20 metres               Sl
51
Tuesday     Crossroads                                        Sl
51
Wednesday   T or staggered junction                           Sl
51
Tuesday     Not at junction or within 20 metres               Sl
53
Friday      Other junction                                    Sl
53

                                                    Light_Conditions
\
Day_of_Week Junction_Detail

Thursday    T or staggered junction                         Daylight

Monday      T or staggered junction                         Daylight

            Not at junction or within 20 metres             Daylight

Tuesday     Crossroads                                      Daylight

Wednesday   T or staggered junction             Darkness - lights lit

Tuesday     Not at junction or within 20 metres             Daylight

Friday      Other junction                                  Daylight
```

```
Local_Authority_(District)  \
Day_of_Week Junction_Detail

Thursday    T or staggered junction                  Kensington and
Chelsea
Monday      T or staggered junction                  Kensington and
Chelsea
            Not at junction or within 20 metres     Kensington and
Chelsea
Tuesday     Crossroads                               Kensington and
Chelsea
Wednesday   T or staggered junction
Westminster
Tuesday     Not at junction or within 20 metres
Manchester
Friday      Other junction                                    Cheshire
East

                                                 Carriageway_Hazards  \
Day_of_Week Junction_Detail
Thursday    T or staggered junction                            NaN
Monday      T or staggered junction                            NaN
            Not at junction or within 20 metres                NaN
Tuesday     Crossroads                                         NaN
Wednesday   T or staggered junction                            NaN
Tuesday     Not at junction or within 20 metres                NaN
Friday      Other junction                                     NaN

                                                  Longitude  \
Day_of_Week Junction_Detail
Thursday    T or staggered junction                    0
Monday      T or staggered junction                    0
            Not at junction or within 20 metres        0
Tuesday     Crossroads                                 0
Wednesday   T or staggered junction                    0
Tuesday     Not at junction or within 20 metres       -2
Friday      Other junction                            -2

                                                 Number_of_Casualties
\
Day_of_Week Junction_Detail

Thursday    T or staggered junction                                1

Monday      T or staggered junction                                1

            Not at junction or within 20 metres                    1
```

```
Tuesday     Crossroads                                                1

Wednesday   T or staggered junction                                   1

Tuesday     Not at junction or within 20 metres                       2

Friday      Other junction                                            1


                                                  Number_of_Vehicles  \
Day_of_Week Junction_Detail
Thursday    T or staggered junction                                  2
Monday      T or staggered junction                                  2
            Not at junction or within 20 metres                      2
Tuesday     Crossroads                                               2
Wednesday   T or staggered junction                                   2
Tuesday     Not at junction or within 20 metres                      1
Friday      Other junction                                           2


Police_Force  \
Day_of_Week Junction_Detail

Thursday    T or staggered junction              Metropolitan Police

Monday      T or staggered junction              Metropolitan Police

            Not at junction or within 20 metres  Metropolitan Police

Tuesday     Crossroads                           Metropolitan Police

Wednesday   T or staggered junction              Metropolitan Police

Tuesday     Not at junction or within 20 metres   Greater Manchester

Friday      Other junction                                 Cheshire



Road_Surface_Conditions  \
Day_of_Week Junction_Detail

Thursday    T or staggered junction
Dry
Monday      T or staggered junction                        Frost or
ice
            Not at junction or within 20 metres             Wet or
damp
Tuesday     Crossroads
Dry
```

```
Wednesday   T or staggered junction
Dry
Tuesday     Not at junction or within 20 metres            Frost or
ice
Friday      Other junction
Dry

                                                        Road_Type  \
Day_of_Week Junction_Detail
Thursday    T or staggered junction                 One way street
Monday      T or staggered junction              Single carriageway
            Not at junction or within 20 metres  Single carriageway
Tuesday     Crossroads                           Single carriageway
Wednesday   T or staggered junction              Single carriageway
Tuesday     Not at junction or within 20 metres  Single carriageway
Friday      Other junction                              Slip road

                                                 Speed_limit   Time  \
Day_of_Week Junction_Detail
Thursday    T or staggered junction                       30  15:11
Monday      T or staggered junction                       30   8:10
            Not at junction or within 20 metres           30  14:00
Tuesday     Crossroads                                    30  10:25
Wednesday   T or staggered junction                       30  22:10
Tuesday     Not at junction or within 20 metres           30   9:20
Friday      Other junction                                40  18:00

                                                Urban_or_Rural_Area  \
Day_of_Week Junction_Detail
Thursday    T or staggered junction                           Urban
Monday      T or staggered junction                           Urban
            Not at junction or within 20 metres               Urban
Tuesday     Crossroads                                        Urban
Wednesday   T or staggered junction                           Urban
Tuesday     Not at junction or within 20 metres               Urban
Friday      Other junction                                    Rural

                                                   Weather_Conditions
\
Day_of_Week Junction_Detail

Thursday    T or staggered junction                Fine no high winds

Monday      T or staggered junction                              Other

            Not at junction or within 20 metres  Raining no high winds

Tuesday     Crossroads                             Fine no high winds

Wednesday   T or staggered junction                Fine no high winds
```

```
Tuesday     Not at junction or within 20 metres     Fine no high winds

Friday      Other junction                          Fine no high winds



Vehicle_Type  \
Day_of_Week Junction_Detail

Thursday    T or staggered junction
Car
Monday      T or staggered junction                  Motorcycle over
500cc
            Not at junction or within 20 metres
Car
Tuesday     Crossroads
Car
Wednesday   T or staggered junction
Car
Tuesday     Not at junction or within 20 metres
Car
Friday      Other junction                       Motorcycle 50cc and
under

                                                 Prediction
Day_of_Week Junction_Detail
Thursday    T or staggered junction                    51
Monday      T or staggered junction                    51
            Not at junction or within 20 metres        51
Tuesday     Crossroads                                 51
Wednesday   T or staggered junction                    51
Tuesday     Not at junction or within 20 metres        51
Friday      Other junction                             53
```

```
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

```
C:\Users\abhin\anaconda3\Lib\site-packages\sklearn\metrics\
_classification.py:386: UserWarning: A single label was found in
'y_true' and 'y_pred'. For the confusion matrix to have the correct
shape, use the 'labels' parameter to pass all known labels.
  warnings.warn(
```

```
cm
```

```
array([[2]], dtype=int64)
```

```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 1.0
```

```
print("Classification Report:\n", classification_report(y_test,
y_pred))
```

```
Classification Report:
               precision    recall  f1-score   support

          51       1.00      1.00      1.00         2

    accuracy                           1.00         2
   macro avg       1.00      1.00      1.00         2
weighted avg       1.00      1.00      1.00         2
```

```
# Select relevant numerical columns
numerical_cols = ['Speed_limit', 'Number_of_Vehicles',
'Number_of_Casualties', 'Longitude', 'Latitude']

# Compute correlation matrix
df_corr = df[numerical_cols].corr()

# Print the correlation matrix
print(df_corr)
```

```
                      Speed_limit  Number_of_Vehicles
Number_of_Casualties  \
Speed_limit              1.000000            0.166667              -
0.166667
Number_of_Vehicles       0.166667            1.000000              -
1.000000
Number_of_Casualties    -0.166667           -1.000000
1.000000
Longitude               -0.645497            0.645497              -
0.645497
Latitude                 0.645497           -0.645497
0.645497

                      Longitude  Latitude
Speed_limit           -0.645497  0.645497
Number_of_Vehicles     0.645497 -0.645497
Number_of_Casualties  -0.645497  0.645497
Longitude              1.000000 -1.000000
Latitude              -1.000000  1.000000
```

```
df.describe().T
```

```
                      count       mean       std   min   25%   50%
75%   max
Latitude                7.0  51.571429  0.975900  51.0  51.0  51.0
52.0  53.0
```

```
Longitude               7.0  -0.571429  0.975900  -2.0  -1.0   0.0
0.0   0.0
Number_of_Casualties    7.0   1.142857  0.377964   1.0   1.0   1.0
1.0   2.0
Number_of_Vehicles      7.0   1.857143  0.377964   1.0   2.0   2.0
2.0   2.0
Speed_limit             7.0  31.428571  3.779645  30.0  30.0  30.0
30.0  40.0
Prediction              7.0  51.285714  0.755929  51.0  51.0  51.0
51.0  53.0
```

```
from sklearn.metrics import mean_squared_error, mean_absolute_error,
r2_score
mae=mean_absolute_error(y_test,y_pred)
mse=mean_squared_error(y_test,y_pred)
rmse=np.sqrt(mse)
r2=r2_score(y_test,y_pred)

print('mae',mae)
print('mse',mse)
print('rmse',rmse)
print('r2',r2)
```

```
mae 0.0
mse 0.0
rmse 0.0
r2 1.0
```

```
df.shape
```

```
(7, 20)
```

```
df.shape[0]
```

```
7
```

```
from sklearn.model_selection import train_test_split

# Feature and target selection based on the road accident dataset
x = df['Speed_limit'].values.reshape(-1, 1)     # Feature
y = df['Number_of_Casualties'].values.reshape(-1, 1)  # Target

# Splitting the dataset
x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size=0.2, random_state=0)

x_train.shape
```

```
(5, 1)
```

```
y_train.shape
```

```
(5, 1)
```

```
from sklearn.linear_model import LinearRegression
model=LinearRegression()
model.fit(x_train, y_train)
```

```
LinearRegression()
```