import pandas as pd

try:

df\_traffic\_accidents = pd.read\_csv('traffic\_accidents.csv') except UnicodeDecodeError:

try:

df\_traffic\_accidents = pd.read\_csv('traffic\_accidents.csv', encoding='latin-1') except Exception as e:

print(f"Error loading the file: {e}")

df\_traffic\_accidents = None except FileNotFoundError: print("Error: 'traffic\_accidents.csv' not found.")

df\_traffic\_accidents = None except Exception as e:

print(f"An unexpected error occurred: {e}")

df\_traffic\_accidents = None

if df\_traffic\_accidents is not None:

display(df\_traffic\_accidents.head())

print(df\_traffic\_accidents.shape)

# **# Examine the shape of the DataFrame**

print("Shape of the DataFrame:", df\_traffic\_accidents.shape)

# **#Determine data types of each column**

print("\nData Types of each column:\n", df\_traffic\_accidents.dtypes)

# **#Generate descriptive statistics for numerical features**

print("\nDescriptive statistics for numerical features:\n", df\_traffic\_accidents.describe())

# **#Analyze the distribution of categorical features**

categorical\_cols = df\_traffic\_accidents.select\_dtypes(include=['object']).columns

for col in categorical\_cols:

print(f"\nDistribution of '{col}':\n{df\_traffic\_accidents[col].value\_counts()}")

**#Investigate correlation between numerical features**

numerical\_cols = df\_traffic\_accidents.select\_dtypes(include=['number']).columns

print(f"\nCorrelation Matrix of Numerical Features:\n{df\_traffic\_accidents[numerical\_cols].corr()}")

# **#Check for missing values**

print("\nMissing values per column:\n",

df\_traffic\_accidents.isnull().sum())

import matplotlib.pyplot as plt

import seaborn as sns

# **#Numerical Features: Histograms and Box Plots**

numerical\_cols = ['num\_units', 'injuries\_total', 'injuries\_fatal',

'injuries\_incapacitating', 'injuries\_non\_incapacitating']

for col in numerical\_cols:

plt.figure(figsize=(8, 6))

plt.subplot(2, 1, 1)

sns.histplot(df\_traffic\_accidents[col], kde=True)

plt.title(f'Histogram of {col}')

plt.xlabel(col)

plt.ylabel('Frequency')

plt.subplot(2, 1, 2)

sns.boxplot(y=df\_traffic\_accidents[col])

plt.title(f'Box Plot of {col}')

plt.ylabel(col)

plt.tight\_layout()

plt.savefig(f'{col}\_visualization.png')

plt.show()

# **#Categorical Features: Bar Charts**

categorical\_cols = ['weather\_condition', 'lighting\_condition',

'crash\_type', 'crash\_day\_of\_week']

for col in categorical\_cols:

plt.figure(figsize=(10, 6))

df\_traffic\_accidents[col].value\_counts().plot(kind='bar')

plt.title(f'Frequency of Accidents by {col}')

plt.xlabel(col) plt.ylabel('Frequency')

plt.xticks(rotation=45, ha='right')

plt.tight\_layout()

plt.savefig(f'{col}\_barchart.png')

plt.show()

# **#Scatter Plots (example: num\_units vs. injuries\_total)**

plt.figure(figsize=(8, 6))

sns.scatterplot(x='num\_units', y='injuries\_total',

data=df\_traffic\_accidents)

plt.title('Scatter Plot: Number of Units vs. Total Injuries')

plt.xlabel('Number of Units')

plt.ylabel('Total Injuries')

plt.savefig('num\_units\_vs\_injuries\_total.png')

plt.show()