Code:

```
from google.colab import files
uploaded = files.upload()
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load uploaded CSV
df = pd.read_csv("road_accidents.csv")

# Show basic info
print("Data Head:\n", df.head())
print("\nMissing Values:\n", df.isnull().sum())

# Drop rows with missing values in important columns
df.dropna(subset=['Accident_Severity', 'Number_of_Vehicles'], inplace=True)

# Convert Date to datetime format
df['Date'] = pd.to_datetime(df['Date'])

# Extract Month
df['Month'] = df['Date'].dt.month

# 1. Monthly accident count
monthly_accidents = df['Month'].value_counts().sort_index()
sns.barplot(x=monthly_accidents.index, y=monthly_accidents.values,
palette='viridis')
```

```
plt.title("Monthly Road Accidents")
plt.xlabel("Month")
plt.ylabel("Accidents")
plt.show()

# 2. Severity distribution
severity_counts = df['Accident_Severity'].value_counts()
sns.barplot(x=severity_counts.index, y=severity_counts.values, palette='rocket')
plt.title("Accident Severity")
plt.xlabel("Severity")
plt.ylabel("Count")
plt.show()

# 3. Number of vehicles involved
sns.histplot(df['Number_of_Vehicles'], bins=10, kde=True)
plt.title("Vehicles Involved in Accidents")
plt.xlabel("Number of Vehicles")
plt.ylabel("Frequency")
plt.show()

# 4. Accidents by weather
weather_accidents = df['Weather_Conditions'].value_counts()
sns.barplot(x=weather_accidents.index, y=weather_accidents.values,
palette='coolwarm')
plt.title("Accidents by Weather")
plt.xlabel("Weather")
plt.ylabel("Count")
plt.xticks(rotation=45)
```

```
plt.show()

# 5. Correlation heatmap
numeric_df = df.select_dtypes(include='number')
sns.heatmap(numeric_df.corr(), annot=True, cmap='magma')
plt.title("Correlation Matrix")
plt.show()
```

Result:

Data Head:

	Date	Time	Location	Accident_Severity	Number_of_Vehicles	\
0	2024-01-01	22:48	Highway 1	Fatal	1	
1	2024-01-02	05:52	Highway 2	Slight	2	
2	2024-01-03	18:23	City C	Slight	4	
3	2024-01-04	20:49	Highway 2	Slight	3	
4	2024-01-05	14:45	Highway 2	Fatal	4	

	Weather_Conditions	Light_Conditions	Road_Surface_Conditions
0	Snow	Darkness - Lights On	Wet
1	Rain	Daylight	Wet
2	Rain	Darkness - No Lights	Dry
3	Clear	Daylight	Wet
4	Clear	Daylight	Dry

Missing Values:

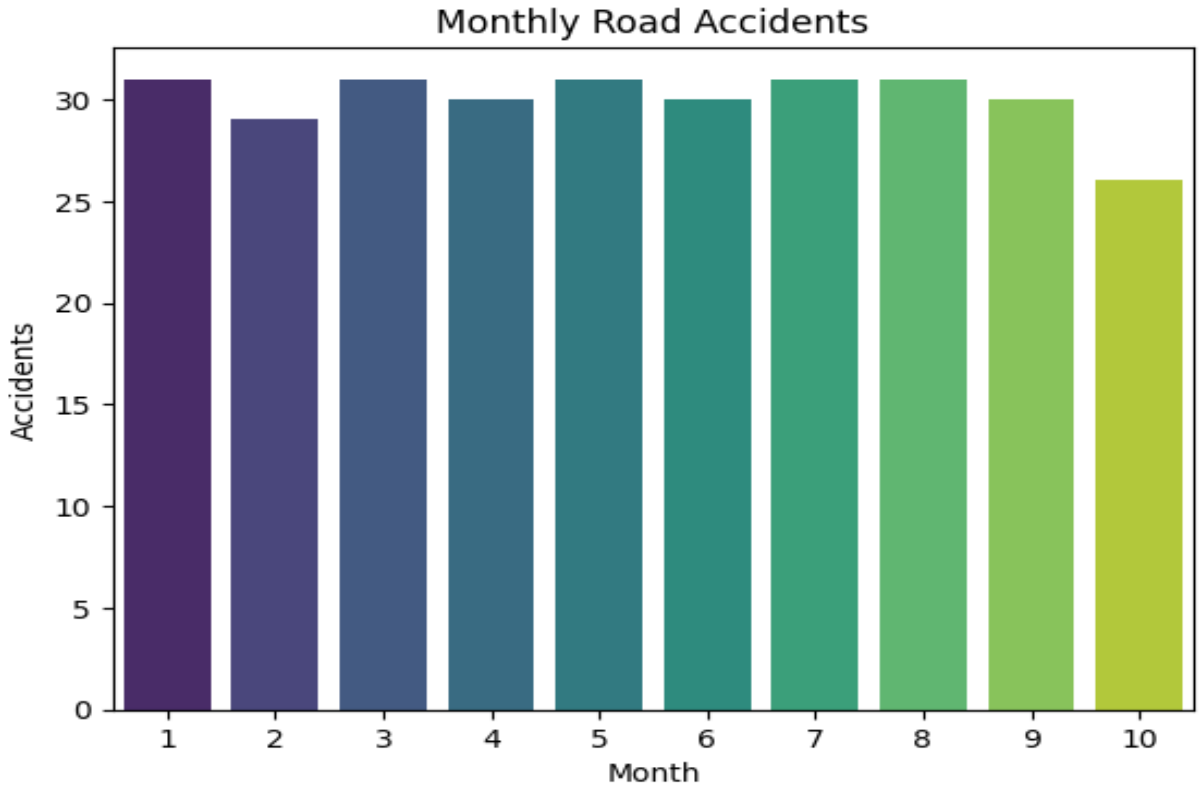
Date	0
Time	0

```
Location                0
Accident_Severity       0
Number_of_Vehicles      0
Weather_Conditions      0
Light_Conditions        0
Road_Surface_Conditions 0
dtype: int64
```

<ipython-input-2-9c2090a145d2>:23: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

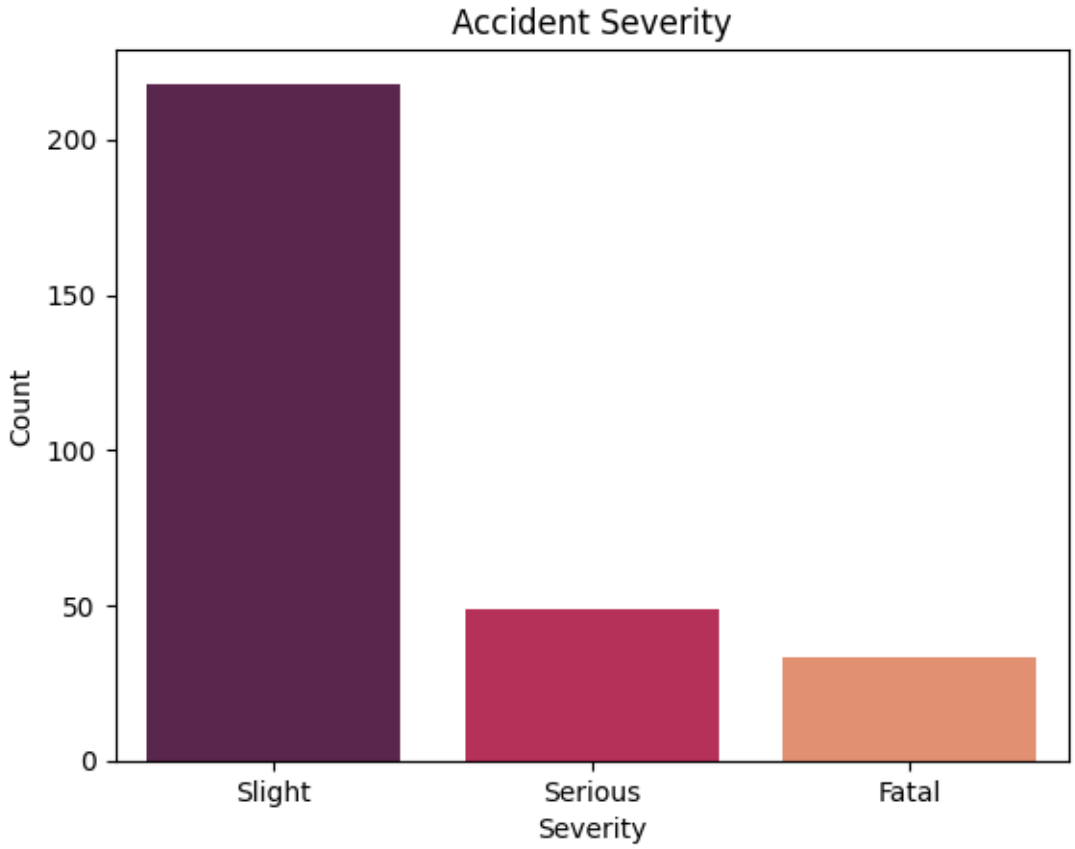
```
  sns.barplot(x=monthly_accidents.index,
y=monthly_accidents.values, palette='viridis')
```

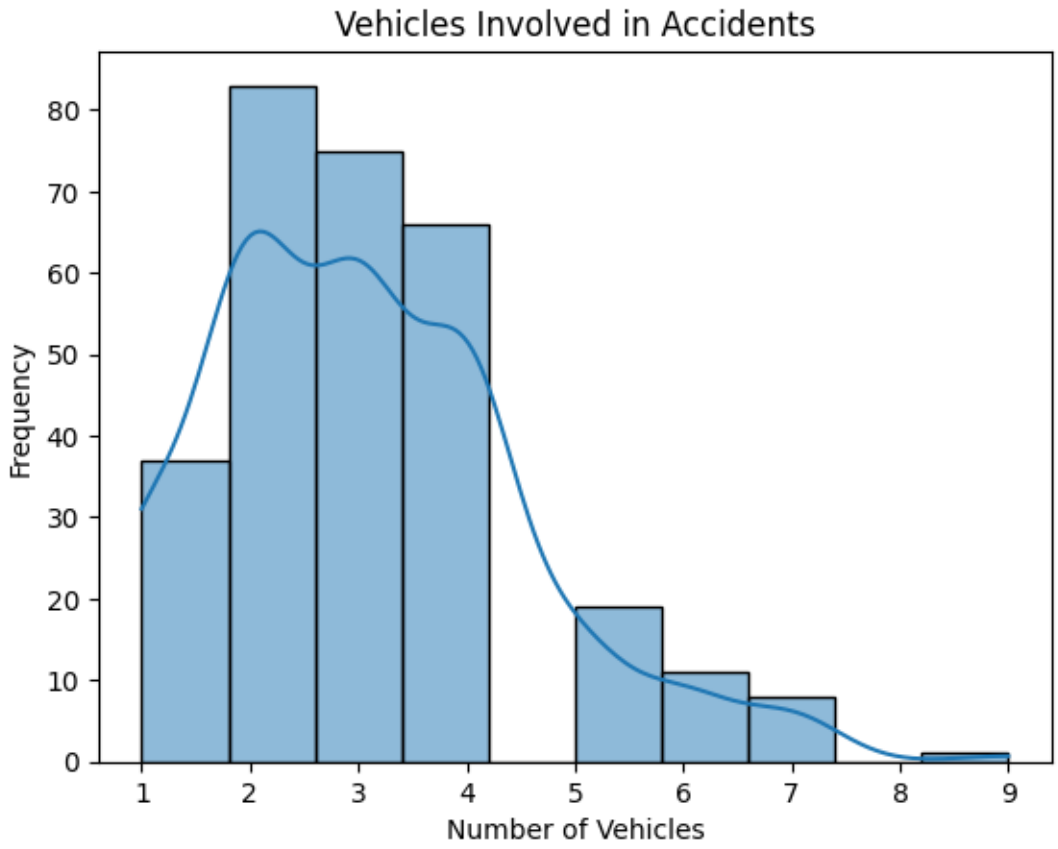


<ipython-input-2-9c2090a145d2>:31: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
  sns.barplot(x=severity_counts.index, y=severity_counts.values, palette='rocket')
```

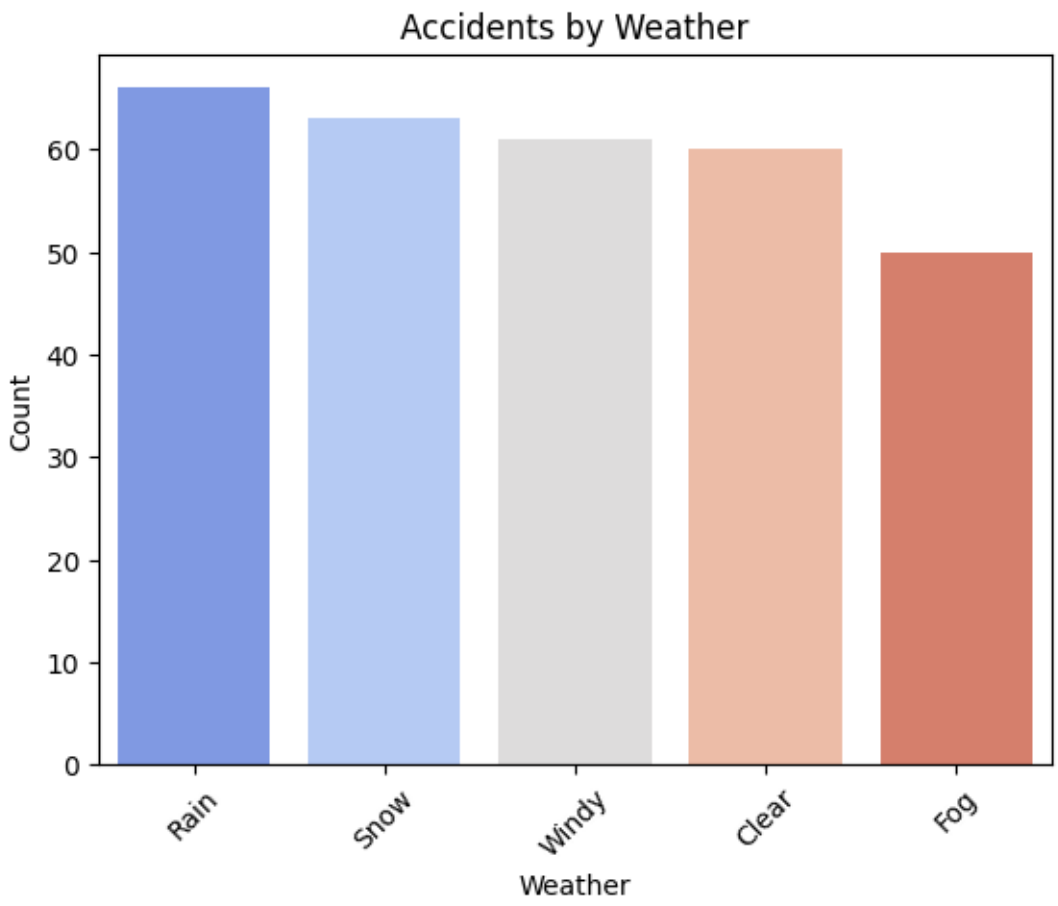


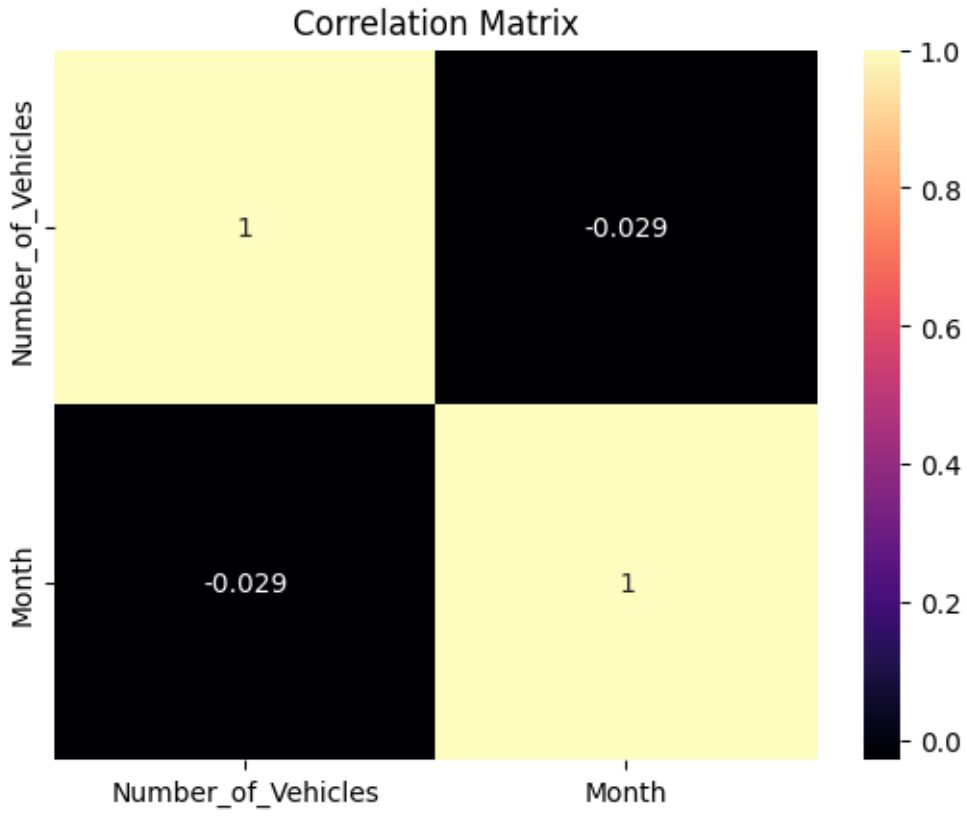


<ipython-input-2-9c2090a145d2>:46: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

sns.barplot(x=weather_accidents.index, y=weather_accidents.values, palette='coolwarm')


Accidents by Weather
Count
0
10
20
30
40
50
60
Rain
Snow
Windy
Clear
Fog
Weather


Correlation Matrix
Number_of_Vehicles
Month
1
-0.029
-0.029
1
Number_of_Vehicles
Month
1.0
0.8
0.6
0.4
0.2
0.0