

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from google.colab import files
import io
```

```
import pandas as pd
```

```
# Sample data dictionary
```

```
data = {
    'longitude': [-73.935242, -74.0060, -118.2437, -87.6298, -77.0369],
    'latitude': [40.730610, 40.7128, 34.0522, 41.8781, 38.9072],
    'severity': [2, 3, 1, 2, 3],
    'road_condition': ['Dry', 'Wet', 'Dry', 'Ice', 'Snow'],
    'weather_condition': ['Clear', 'Rain', 'Clear', 'Snow', 'Fog'],
    'time_of_day': ['13:45', '09:30', '17:00', '21:15', '05:30']
}
```

```
# Create a DataFrame
```

```
df = pd.DataFrame(data)
```

```
# Save DataFrame to a CSV file
```

```
df.to_csv('sample_traffic_accidents.csv', index=False)
```

```
uploaded = files.upload()
```



Choose Files No file chosen

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.



```
for fn in uploaded.keys():
```

```
    print('User uploaded file "{name}" with length {length} bytes'.format(name=fn, length=len(uploaded[fn]))
```



User uploaded file "sample_traffic_accidents (1).csv" with length 248 bytes

```
df = pd.read_csv(io.BytesIO(uploaded[fn]))
```

```
print(df.head())
```



	longitude	latitude	severity	road_condition	weather_condition	time_of_day
0	-73.935242	40.73061	2	Dry	Clear	13:45
1	-74.006000	40.71280	3	Wet	Rain	09:30
2	-118.243700	34.05220	1	Dry	Clear	17:00
3	-87.629800	41.87810	2	Ice	Snow	21:15
4	-77.036900	38.90720	3	Snow	Fog	05:30

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



```
longitude      0
latitude       0
severity       0
road_condition 0
weather_condition 0
time_of_day    0
dtype: int64
```

```
df = df.dropna()
```

```
df['time_of_day'] = pd.to_datetime(df['time_of_day'], format='%H:%M').dt.hour
```

```
road_conditions_counts = df['road_condition'].value_counts()
print(road_conditions_counts)
```

```
↗ ↘ road_condition
Dry      2
Wet      1
Ice      1
Snow     1
Name: count, dtype: int64
```

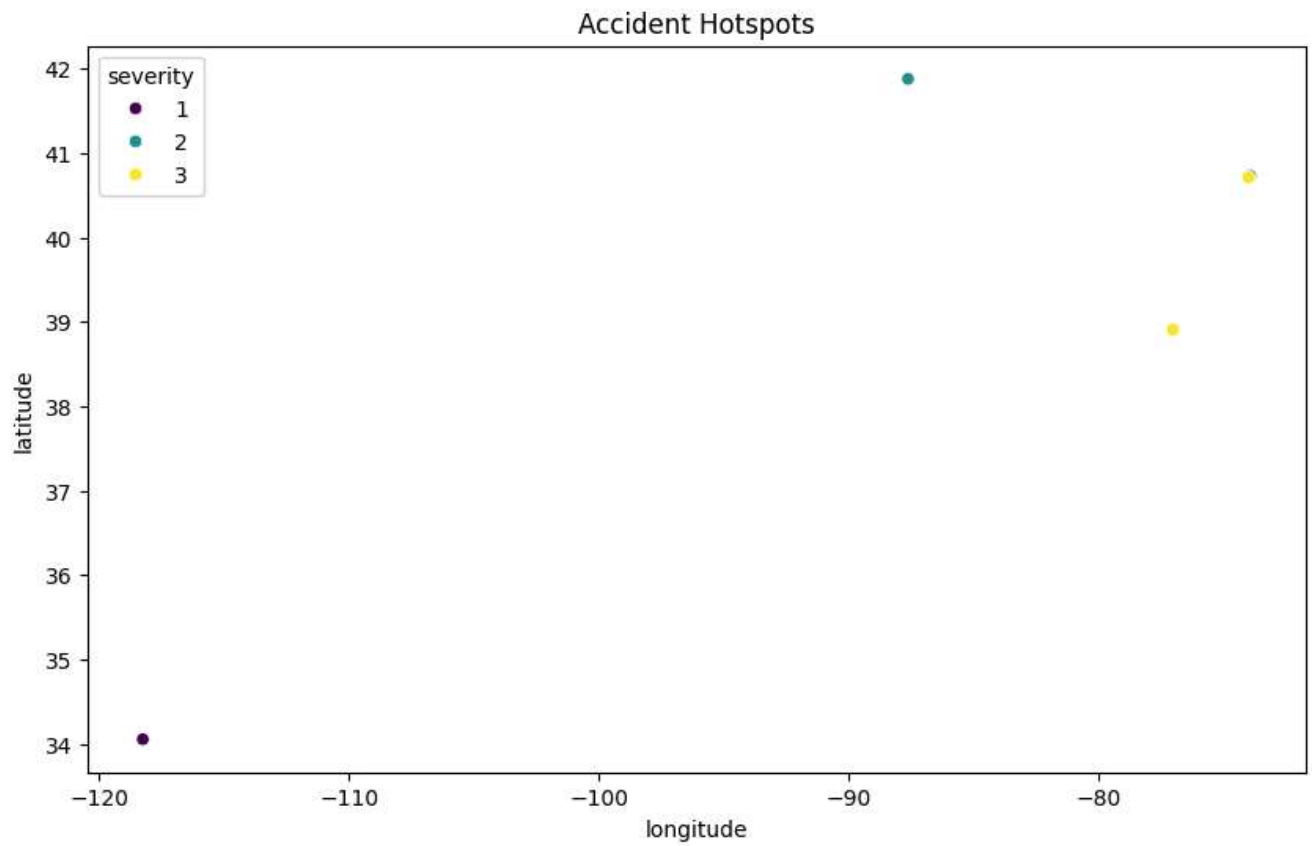
```
weather_conditions_counts = df['weather_condition'].value_counts()
print(weather_conditions_counts)
```

```
↗ ↘ weather_condition
Clear    2
Rain     1
Snow     1
Fog      1
Name: count, dtype: int64
```

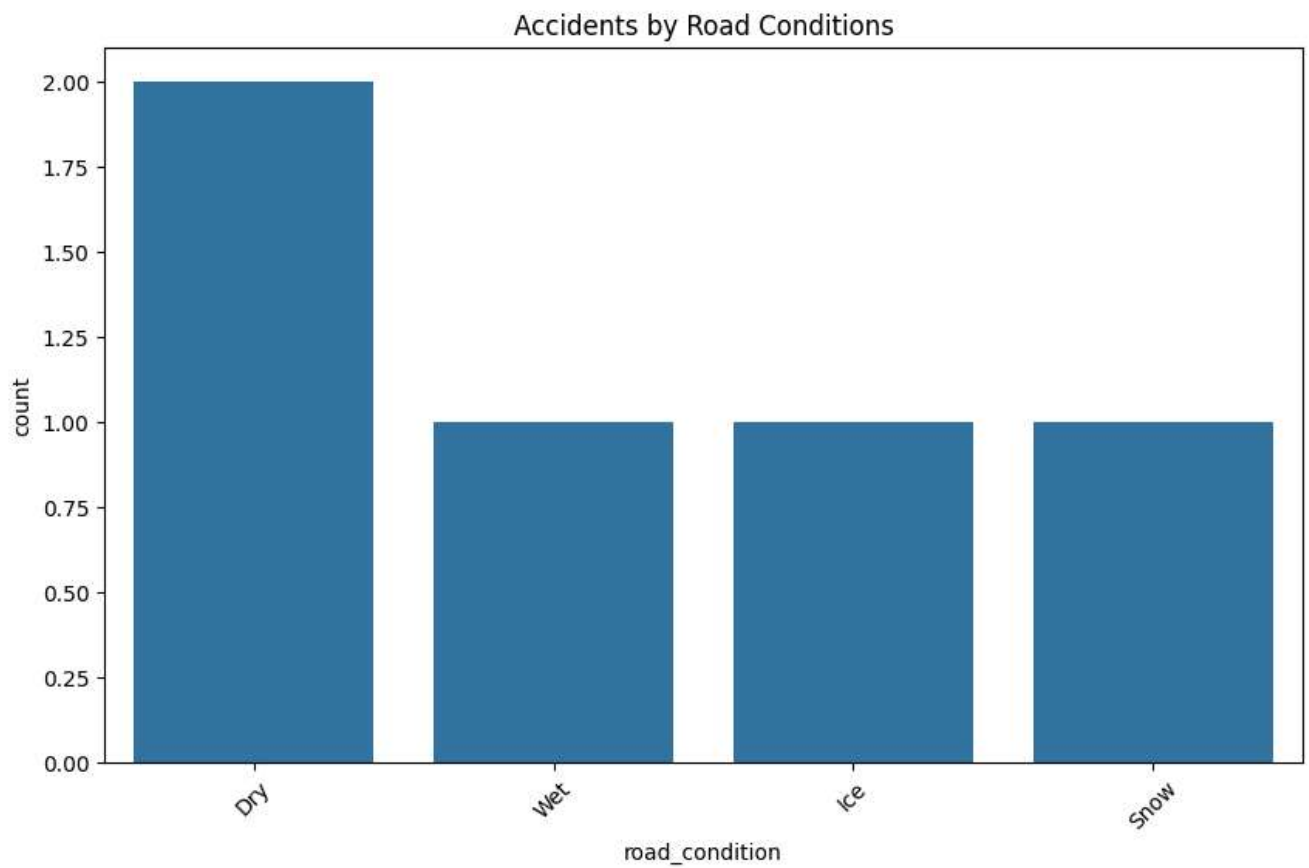
```
time_of_day_counts = df['time_of_day'].value_counts()
print(time_of_day_counts)
```

```
↗ ↘ time_of_day
13      1
9       1
17      1
21      1
5       1
Name: count, dtype: int64
```

```
# Step 4: Visualize accident hotspots and contributing factors
# Accident hotspots by location (assuming latitude and longitude columns)
plt.figure(figsize=(10, 6))
sns.scatterplot(x='longitude', y='latitude', hue='severity', data=df, palette='viridis')
plt.title('Accident Hotspots')
plt.show()
```



```
# Accidents by road conditions
plt.figure(figsize=(10, 6))
sns.countplot(x='road_condition', data=df)
plt.title('Accidents by Road Conditions')
plt.xticks(rotation=45)
plt.show()
```



```
# Accidents by weather conditions
plt.figure(figsize=(10, 6))
sns.countplot(x='weather_condition', data=df)
plt.title('Accidents by Weather Conditions')
plt.xticks(rotation=45)
plt.show()
```



Accidents by Weather Conditions

2.00



```
plt.figure(figsize=(10, 6))
sns.countplot(x='time_of_day', data=df, palette='coolwarm')
plt.title('Accidents by Time of Day')
plt.xlabel('Hour of Day')
plt.show()
```



<ipython-input-16-1d5d95d40985>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x`

```
sns.countplot(x='time_of_day', data=df, palette='coolwarm')
```

Accidents by Time of Day

1.0

