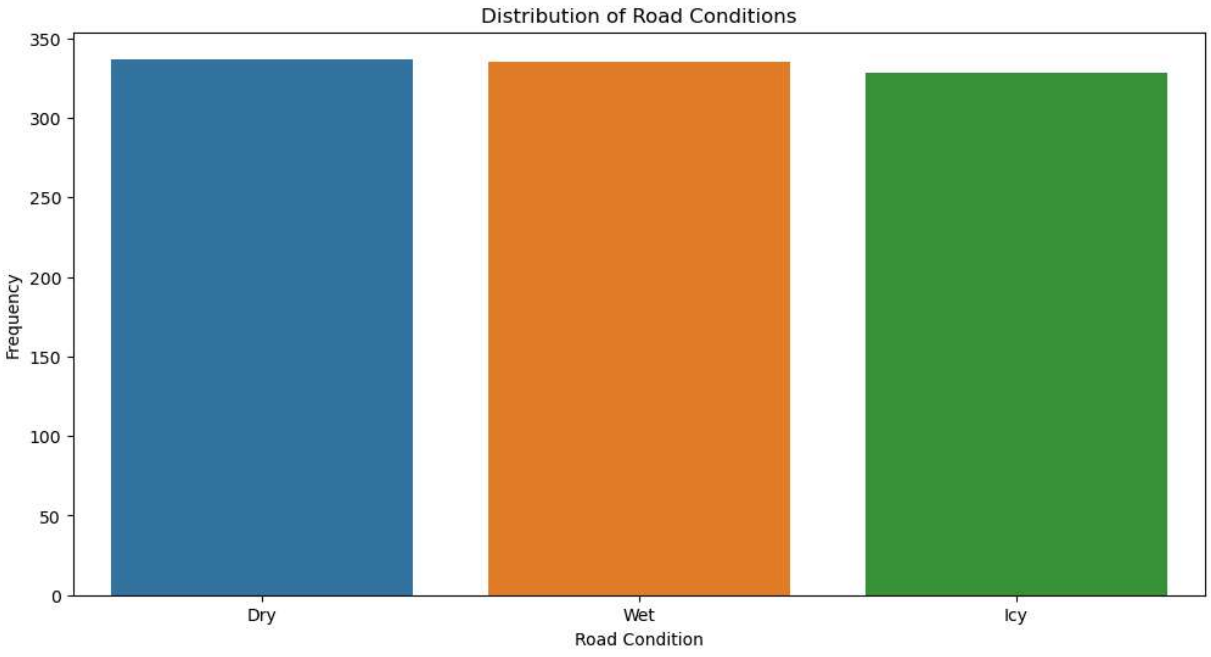
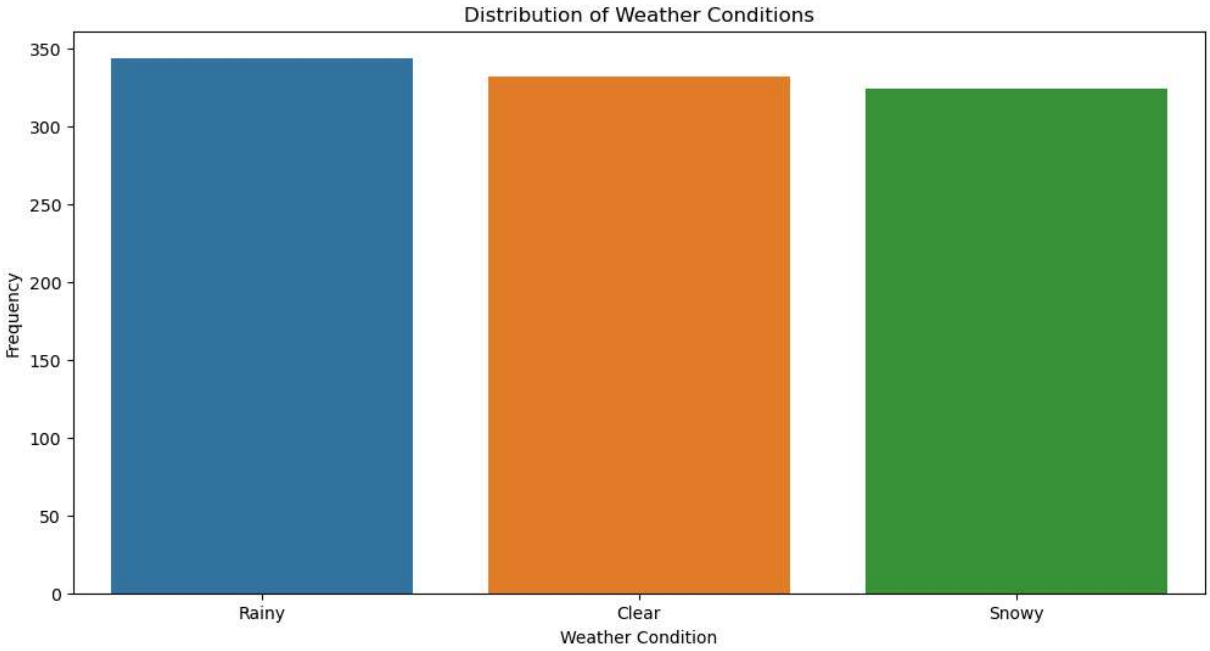
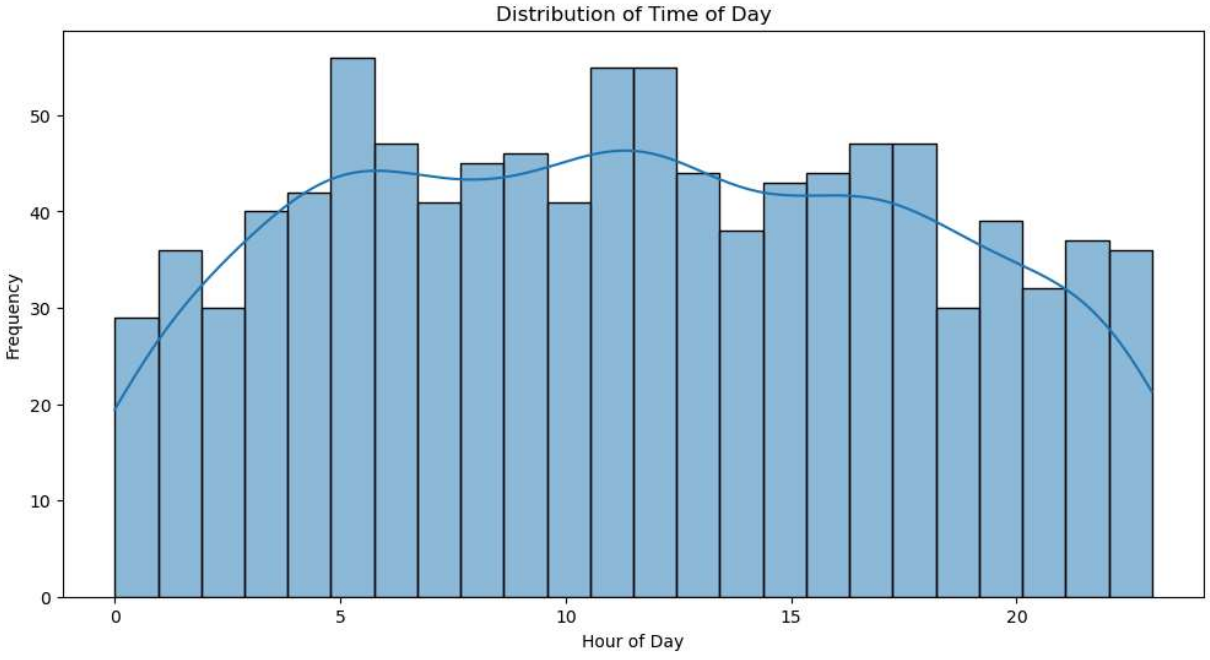
In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans

# Generate synthetic data
np.random.seed(0)
n_samples = 1000

# Simulate road conditions (0: dry, 1: wet, 2: icy)
road_conditions = np.random.choice(['Dry', 'Wet', 'Icy'], size=n_samples)

# Simulate weather conditions (0: clear, 1: rainy, 2: snowy)
weather_conditions = np.random.choice(['Clear', 'Rainy', 'Snowy'], size=n_samples)

# Simulate time of day (0-23 hours)
time_of_day = np.random.randint(0, 24, size=n_samples)

# Simulate accident locations (latitude and longitude)
latitude = np.random.uniform(40, 41, size=n_samples)
longitude = np.random.uniform(-74, -73, size=n_samples)

# Simulate accident severity (random integers from 1 to 5)
severity = np.random.randint(1, 6, size=n_samples)

# Create a synthetic DataFrame
data = pd.DataFrame({
    'road_condition': road_conditions,
    'weather': weather_conditions,
    'time_of_day': time_of_day,
    'latitude': latitude,
    'longitude': longitude,
    'severity': severity
})

# Data preprocessing
# Exploratory Data Analysis (EDA)
# Visualize distributions of road conditions, weather, and time of day
plt.figure(figsize=(12, 6))
sns.countplot(x='road_condition', data=data)
plt.title('Distribution of Road Conditions')
plt.xlabel('Road Condition')
plt.ylabel('Frequency')
plt.show()

plt.figure(figsize=(12, 6))
sns.countplot(x='weather', data=data)
plt.title('Distribution of Weather Conditions')
plt.xlabel('Weather Condition')
plt.ylabel('Frequency')
plt.show()

plt.figure(figsize=(12, 6))
```

```
sns.histplot(data['time_of_day'], bins=24, kde=True)
plt.title('Distribution of Time of Day')
plt.xlabel('Hour of Day')
plt.ylabel('Frequency')
plt.show()

# Identify accident hotspots using clustering (K-means)
X = data[['latitude', 'longitude']]
kmeans = KMeans(n_clusters=5, random_state=0).fit(X)
data['cluster'] = kmeans.labels_

# Visualize accident hotspots
plt.figure(figsize=(10, 8))
sns.scatterplot(x='longitude', y='latitude', hue='cluster', data=data, palette='vir
plt.title('Accident Hotspots')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.legend(title='Cluster')
plt.show()

# Time of Day Analysis
plt.figure(figsize=(12, 6))
sns.countplot(x='time_of_day', data=data, color='orange')
plt.title('Accidents by Hour of Day')
plt.xlabel('Hour of Day')
plt.ylabel('Number of Accidents')
plt.show()

# Factor Analysis
plt.figure(figsize=(10, 6))
sns.boxplot(x='road_condition', y='severity', data=data)
plt.title('Impact of Road Conditions on Accident Severity')
plt.xlabel('Road Condition')
plt.ylabel('Severity')
plt.show()
```

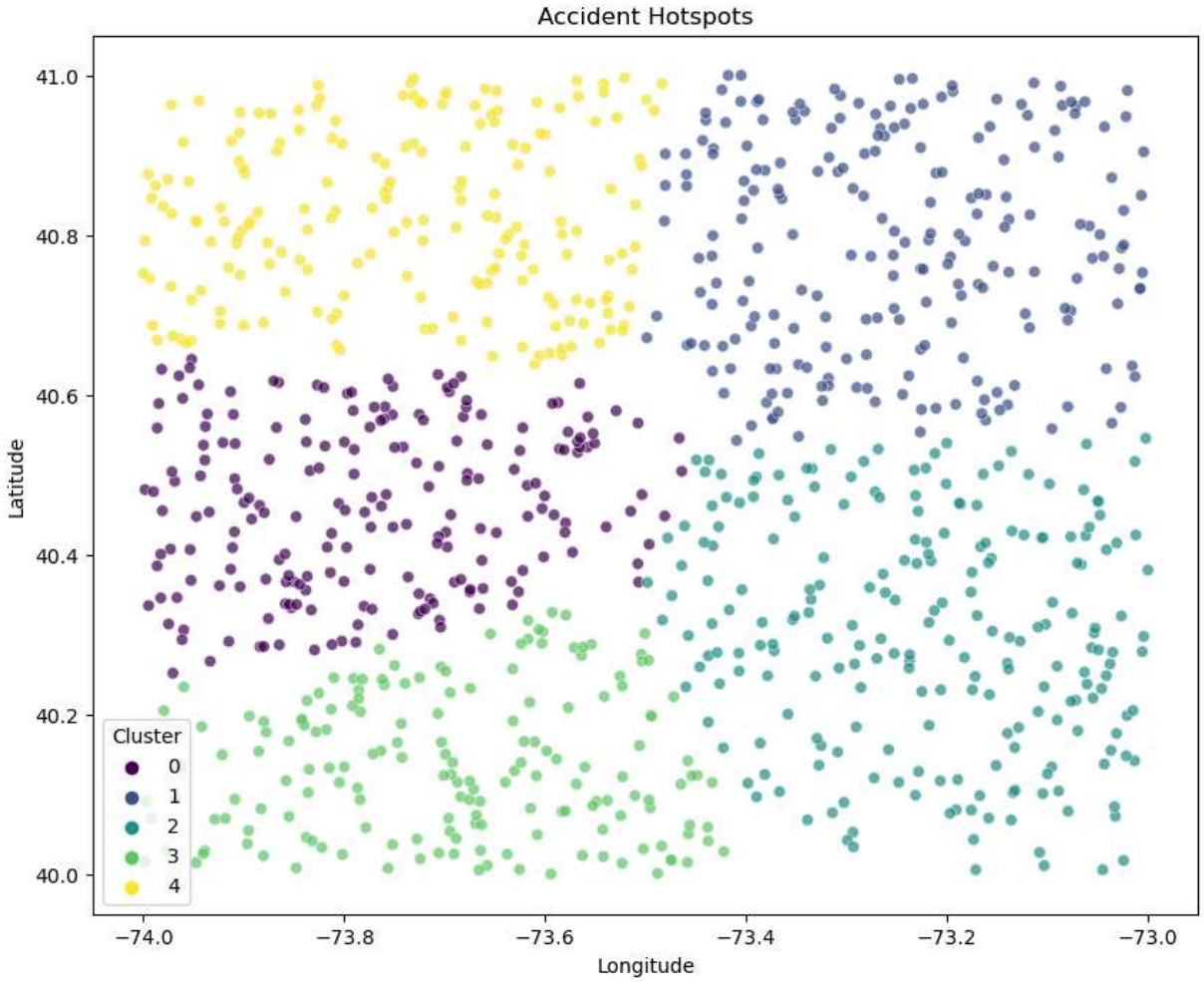


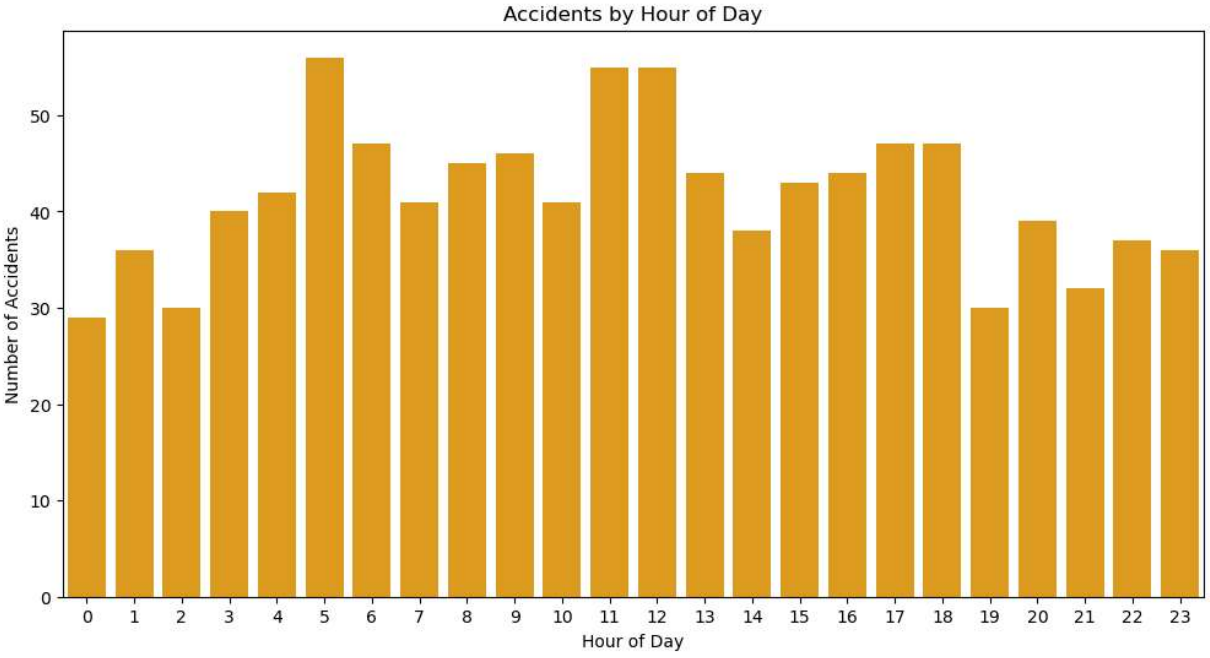


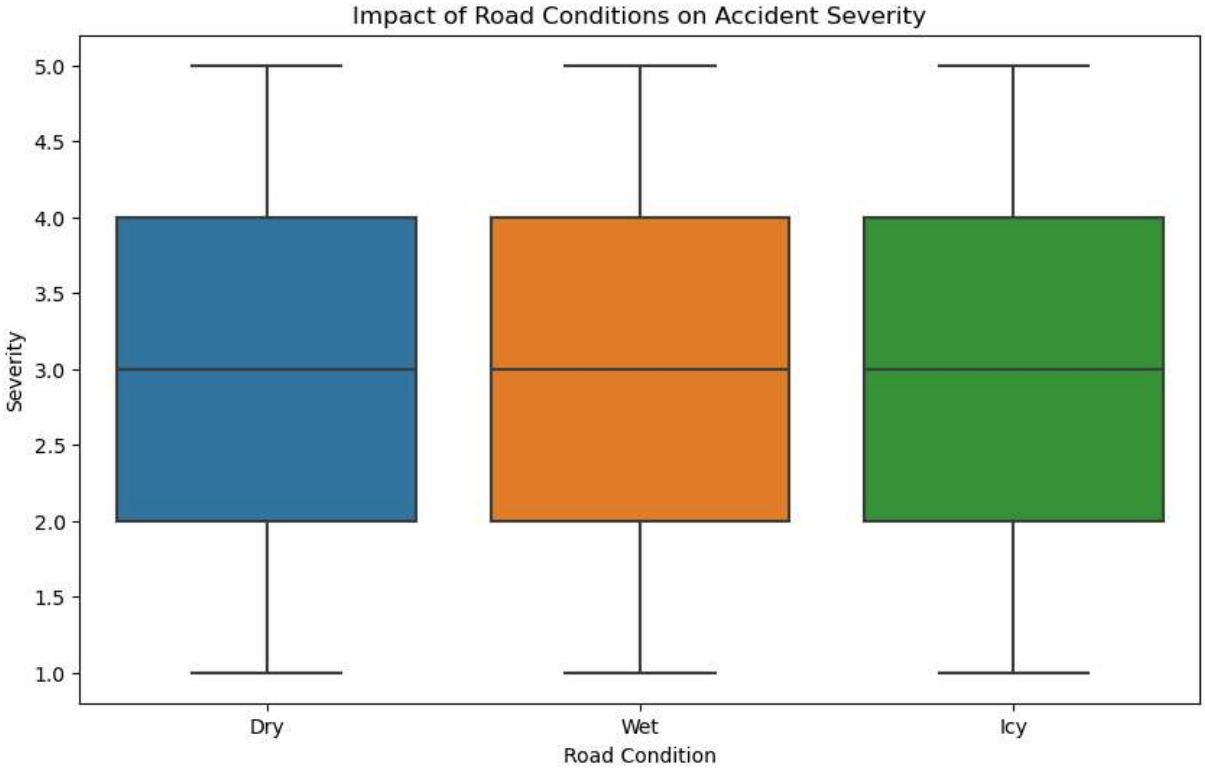

```
C:\Users\devat\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version. Convert
inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```




```
C:\Users\devat\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWar
ning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the va
lue of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\devat\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWarn
ing: KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment variable
OMP_NUM_THREADS=4.
  warnings.warn(
```


Accident Hotspots
Latitude
Longitude
Cluster
0
1
2
3
4
41.0
40.8
40.6
40.4
40.2
40.0
−74.0
−73.8
−73.6
−73.4
−73.2
−73.0


Accidents by Hour of Day
Number of Accidents
Hour of Day
0
10
20
30
40
50
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23


Impact of Road Conditions on Accident Severity
Severity
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
Dry
Wet
Icy
Road Condition