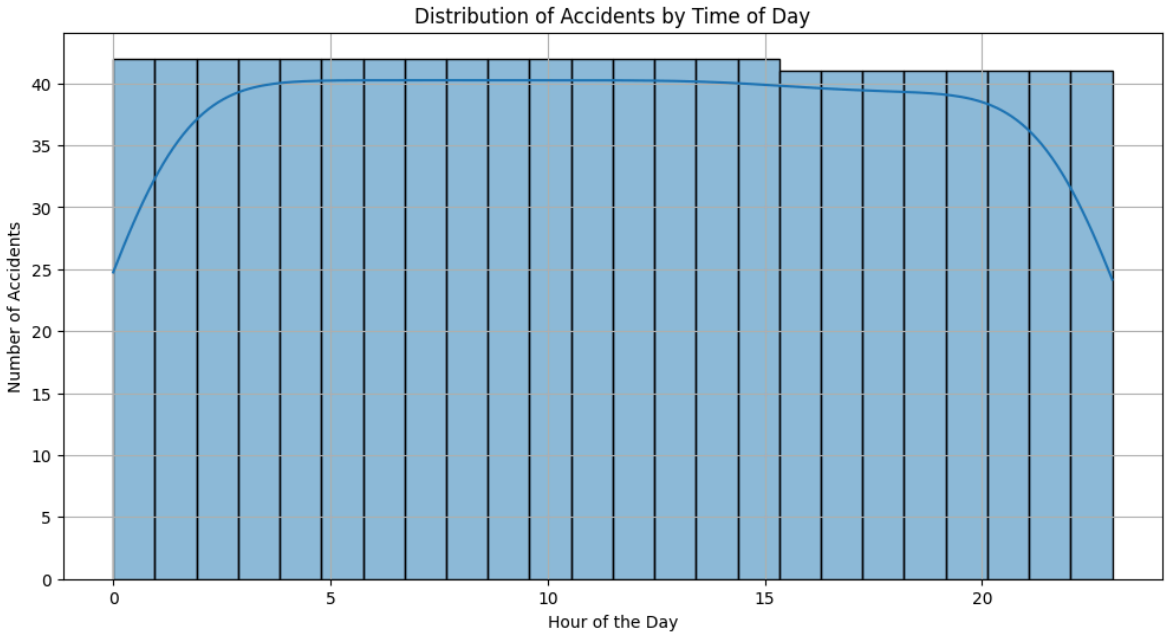
In [8]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import gaussian_kde

# Create a synthetic dataset
np.random.seed(0)

# Generate random dates and times
dates = pd.date_range(start='1/1/2023', periods=1000, freq='H')
time_of_day = dates.hour

# Generate random locations (latitude and longitude)
latitudes = np.random.uniform(low=34.0, high=36.0, size=1000)
longitudes = np.random.uniform(low=-118.0, high=-116.0, size=1000)

# Generate random road conditions and weather
road_conditions = np.random.choice(['Dry', 'Wet', 'Snow', 'Ice'], size=1000, p=[
weather_conditions = np.random.choice(['Clear', 'Rain', 'Fog', 'Snow'], size=100

# Create a DataFrame
data = pd.DataFrame({
    'Date': dates,
    'Time of Day': time_of_day,
    'Latitude': latitudes,
    'Longitude': longitudes,
    'Road Condition': road_conditions,
    'Weather': weather_conditions
})
```

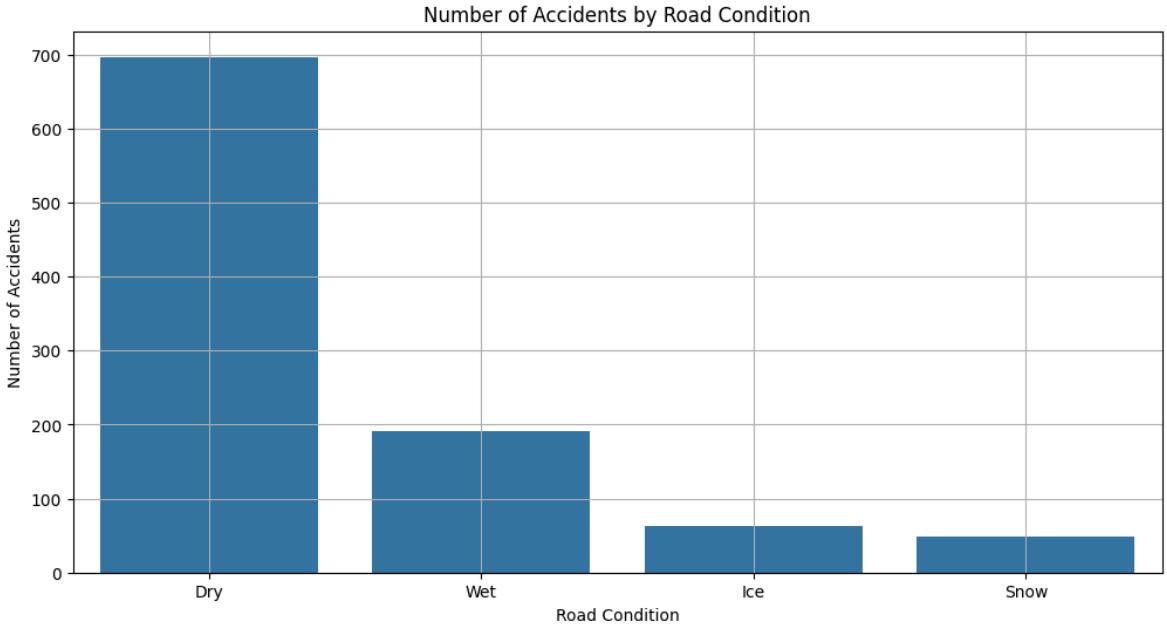
In [9]:
```
print(data.head())
```

```
                 Date  Time of Day   Latitude   Longitude Road Condition  \
0 2023-01-01 00:00:00            0  35.097627 -116.814239            Wet
1 2023-01-01 01:00:00            1  35.430379 -117.979873            Dry
2 2023-01-01 02:00:00            2  35.205527 -117.048348            Dry
3 2023-01-01 03:00:00            3  35.089766 -116.582459            Dry
4 2023-01-01 04:00:00            4  34.847310 -117.912049            Dry

  Weather
0   Clear
1   Clear
2    Rain
3    Rain
4    Rain
```

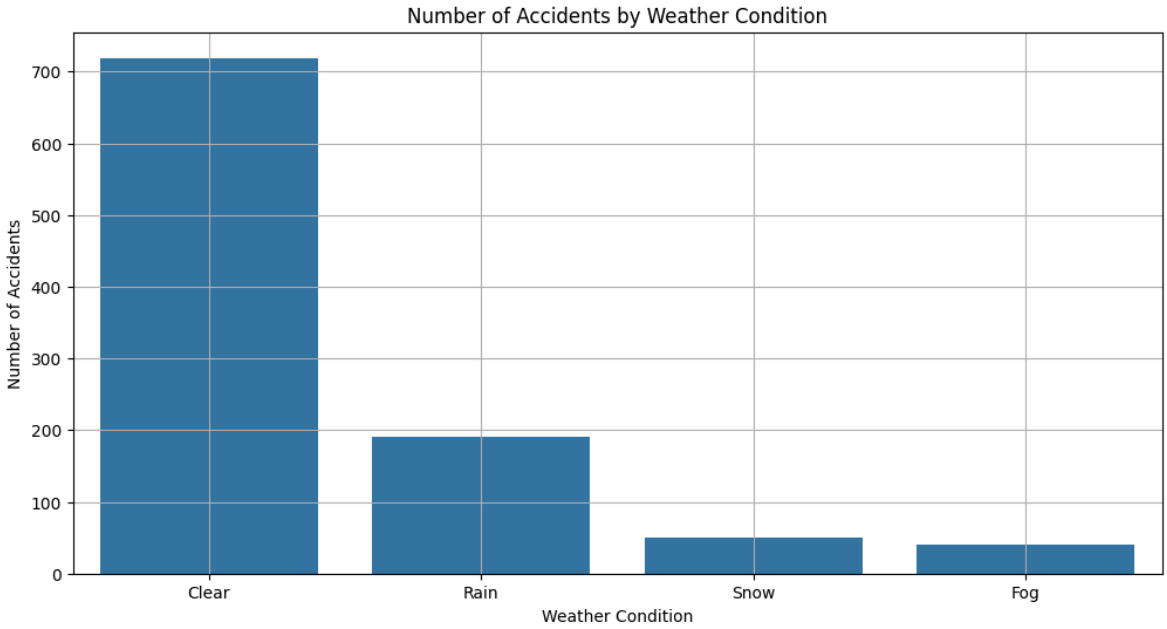
In [10]:
```
plt.figure(figsize=(12, 6))
sns.histplot(data['Time of Day'], bins=24, kde=True)
plt.title('Distribution of Accidents by Time of Day')
plt.xlabel('Hour of the Day')
plt.ylabel('Number of Accidents')
plt.grid(True)
plt.show()
```



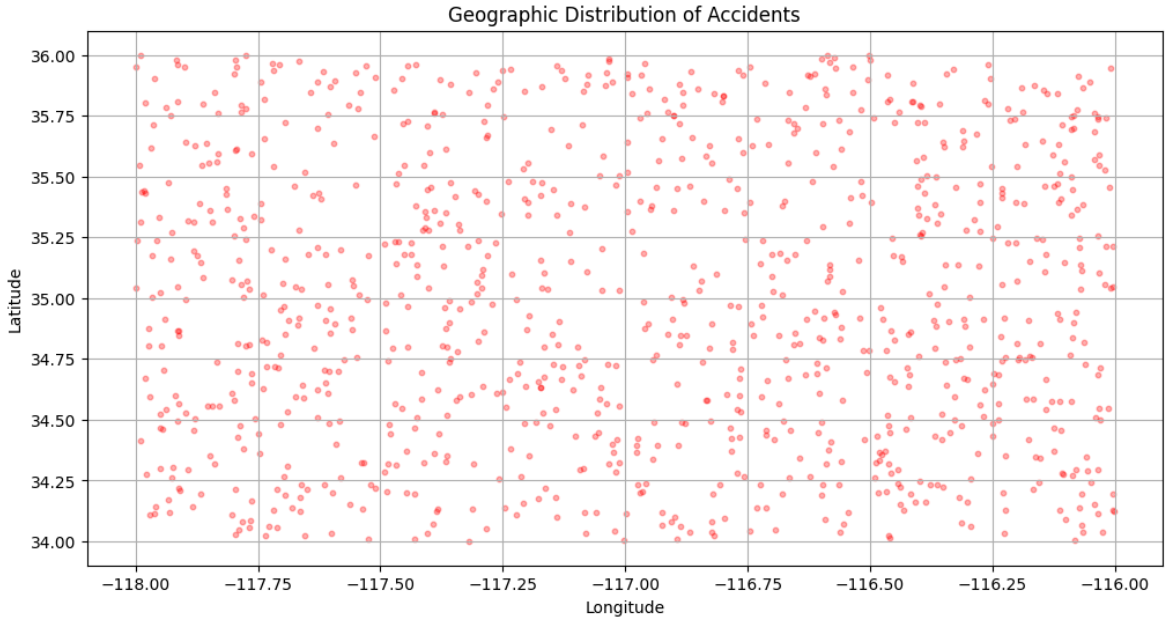

In [11]:
```
plt.figure(figsize=(12, 6))
sns.countplot(data=data, x='Road Condition', order=data['Road Condition'].value_
plt.title('Number of Accidents by Road Condition')
plt.xlabel('Road Condition')
plt.ylabel('Number of Accidents')
plt.grid(True)
plt.show()
```



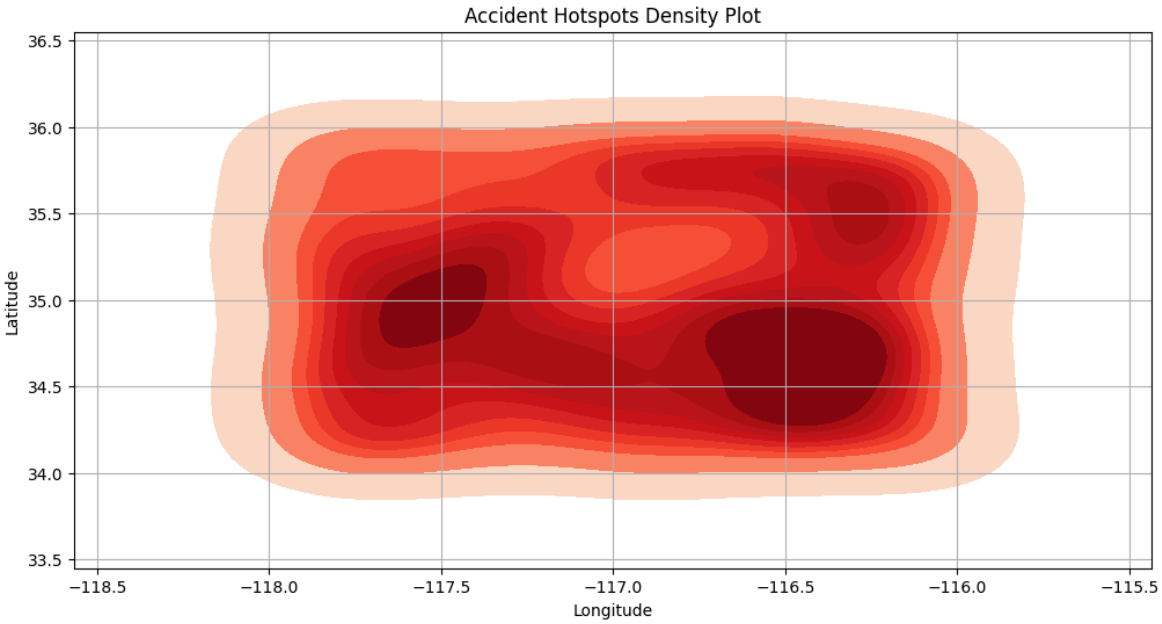

In [12]:
```
plt.figure(figsize=(12, 6))
sns.countplot(data=data, x='Weather', order=data['Weather'].value_counts().index
plt.title('Number of Accidents by Weather Condition')
plt.xlabel('Weather Condition')
plt.ylabel('Number of Accidents')
plt.grid(True)
plt.show()
```




In [13]:
```
plt.figure(figsize=(12, 6))
plt.scatter(data['Longitude'], data['Latitude'], alpha=0.3, c='red', s=10)
plt.title('Geographic Distribution of Accidents')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.grid(True)
plt.show()
```




In [14]:
```
plt.figure(figsize=(12, 6))
sns.kdeplot(x=data['Longitude'], y=data['Latitude'], cmap='Reds', fill=True)
plt.title('Accident Hotspots Density Plot')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.grid(True)
plt.show()
```


Accident Hotspots Density Plot
36.5
36.0
35.5
35.0
34.5
34.0
33.5
Latitude
−118.5
−118.0
−117.5
−117.0
−116.5
−116.0
−115.5
Longitude