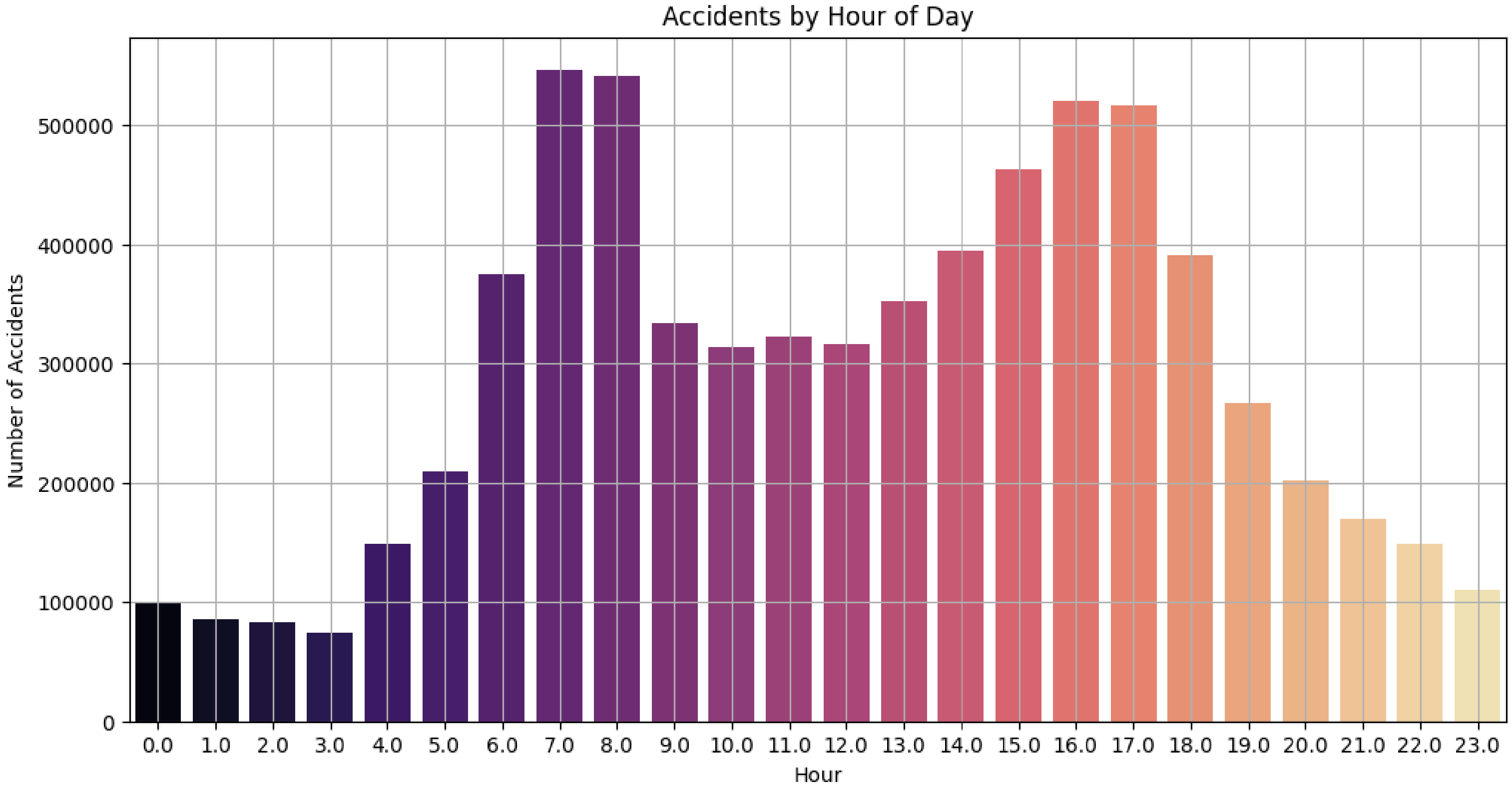
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
import folium
from folium.plugins import HeatMap
```

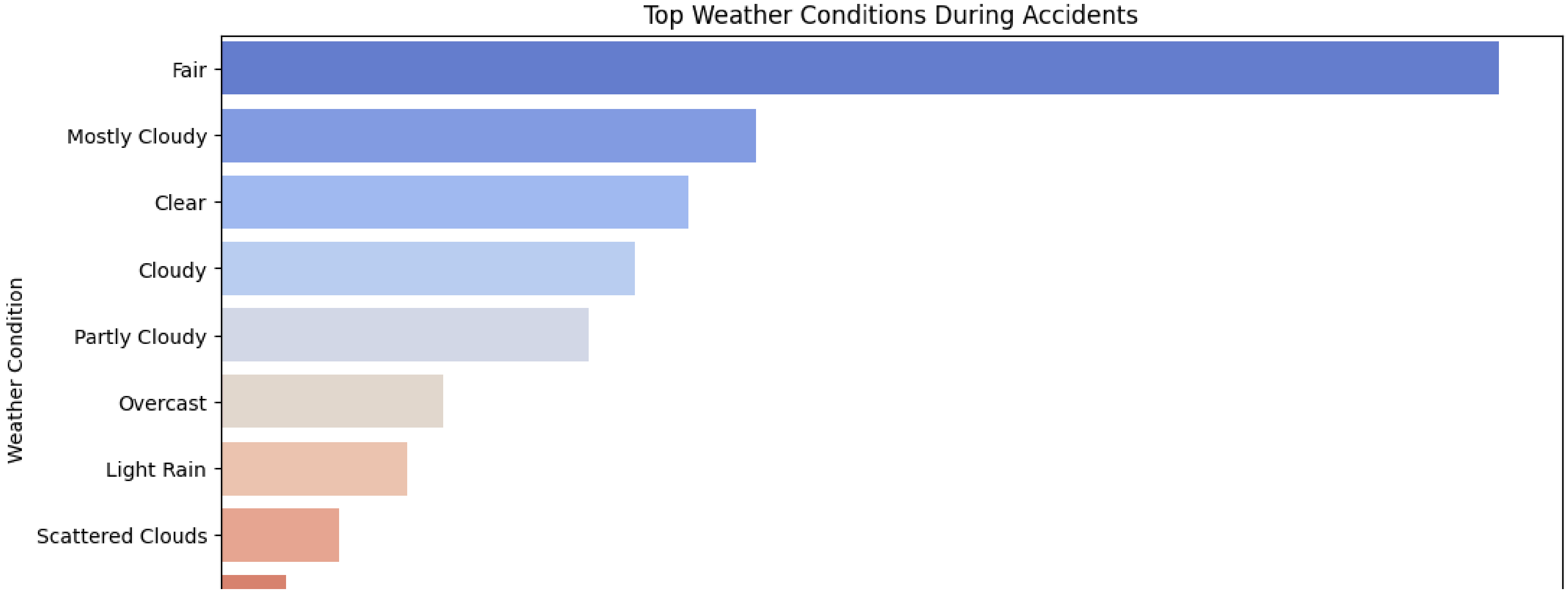
```
df1 = pd.read_csv('/content/drive/MyDrive/US_Accidents_March23.csv', nrows=0).columns.tolist()
```

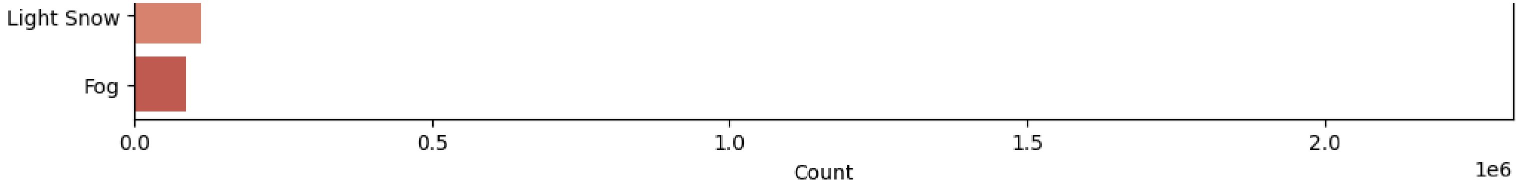
```
df1
```

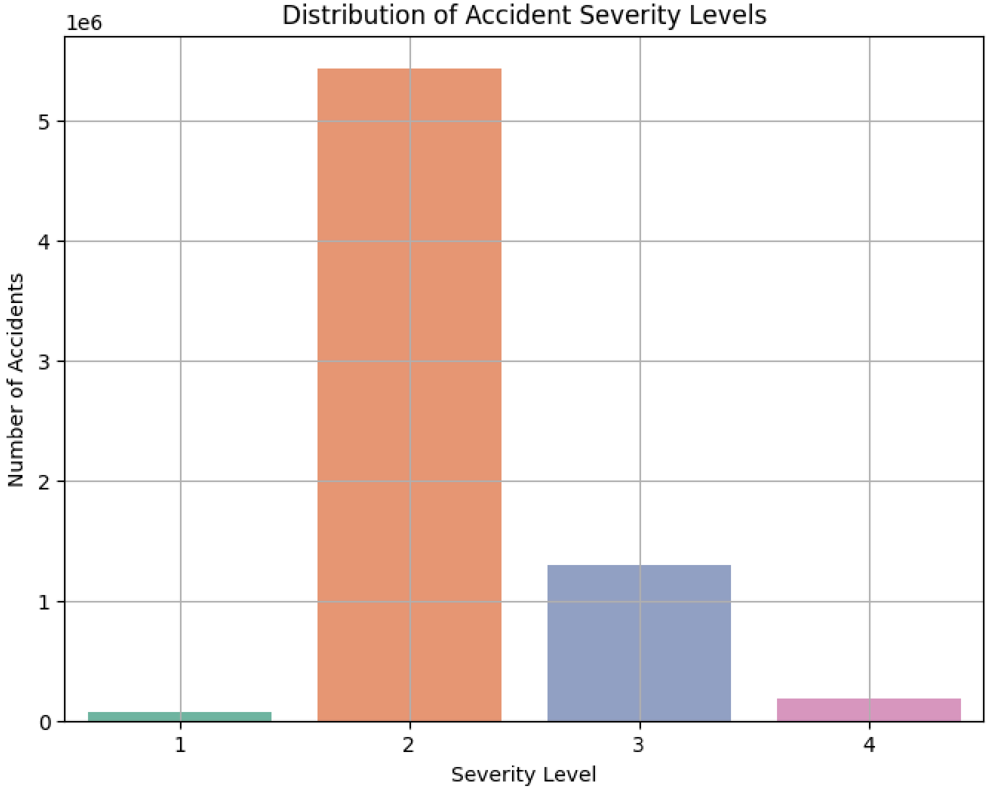
```
['ID',
 'Source',
 'Severity',
 'Start_Time',
 'End_Time',
 'Start_Lat',
 'Start_Lng',
 'End_Lat',
 'End_Lng',
 'Distance(mi)',
 'Description',
 'Street',
 'City',
 'County',
 'State',
 'Zipcode',
 'Country',
 'Timezone',
 'Airport_Code',
 'Weather_Timestamp',
 'Temperature(F)',
 'Wind_Chill(F)',
 'Humidity(%)',
 'Pressure(in)',
 'Visibility(mi)',
 'Wind_Direction',
 'Wind_Speed(mph)',
 'Precipitation(in)',
 'Weather_Condition',
 'Amenity',
 'Bump',
 'Crossing',
 'Give_Way',
 'Junction',
 'No_Exit',
 'Railway',
 'Roundabout',
 'Station',
 'Stop',
      'Traffic_Calming',
      'Traffic_Signal',
      'Turning_Loop',
      'Sunrise_Sunset',
      'Civil_Twilight',
      'Nautical_Twilight',
      'Astronomical_Twilight']


# Load only necessary columns
cols = ['Severity', 'Start_Time', 'Start_Lat', 'Start_Lng',
        'City', 'State', 'Weather_Condition', 'Temperature(F)', 'Humidity(%)',
        'Visibility(mi)', 'Wind_Speed(mph)', 'Precipitation(in)',
        'Amenity', 'Crossing', 'Junction', 'Traffic_Signal', 'Sunrise_Sunset']

df = pd.read_csv('/content/drive/MyDrive/US_Accidents_March23.csv', usecols=cols, low_memory=False)



df['Start_Time'] = pd.to_datetime(df['Start_Time'], errors='coerce')


df['Hour'] = df['Start_Time'].dt.hour
df['Day'] = df['Start_Time'].dt.day_name()

df.dropna(subset=['Start_Lat', 'Start_Lng', 'Start_Time'], inplace=True)



plt.figure(figsize=(12,6))
sns.countplot(x='Hour', data=df, palette='magma')
plt.title("Accidents by Hour of Day")
plt.xlabel("Hour")
plt.ylabel("Number of Accidents")
plt.grid(True)
plt.show()

plt.figure(figsize=(12,6))
top_weather = df['Weather_Condition'].value_counts().nlargest(10).index
sns.countplot(y='Weather_Condition', data=df[df['Weather_Condition'].isin(top_weather)], order=top_weather, palette='coolwarm')
plt.title("Top Weather Conditions During Accidents")
plt.xlabel("Count")
plt.ylabel("Weather Condition")
plt.show()

plt.figure(figsize=(8,6))
sns.countplot(x='Severity', data=df, palette='Set2')
plt.title("Distribution of Accident Severity Levels")
plt.xlabel("Severity Level")
plt.ylabel("Number of Accidents")
plt.grid(True)
```

```
    plt.show()
```

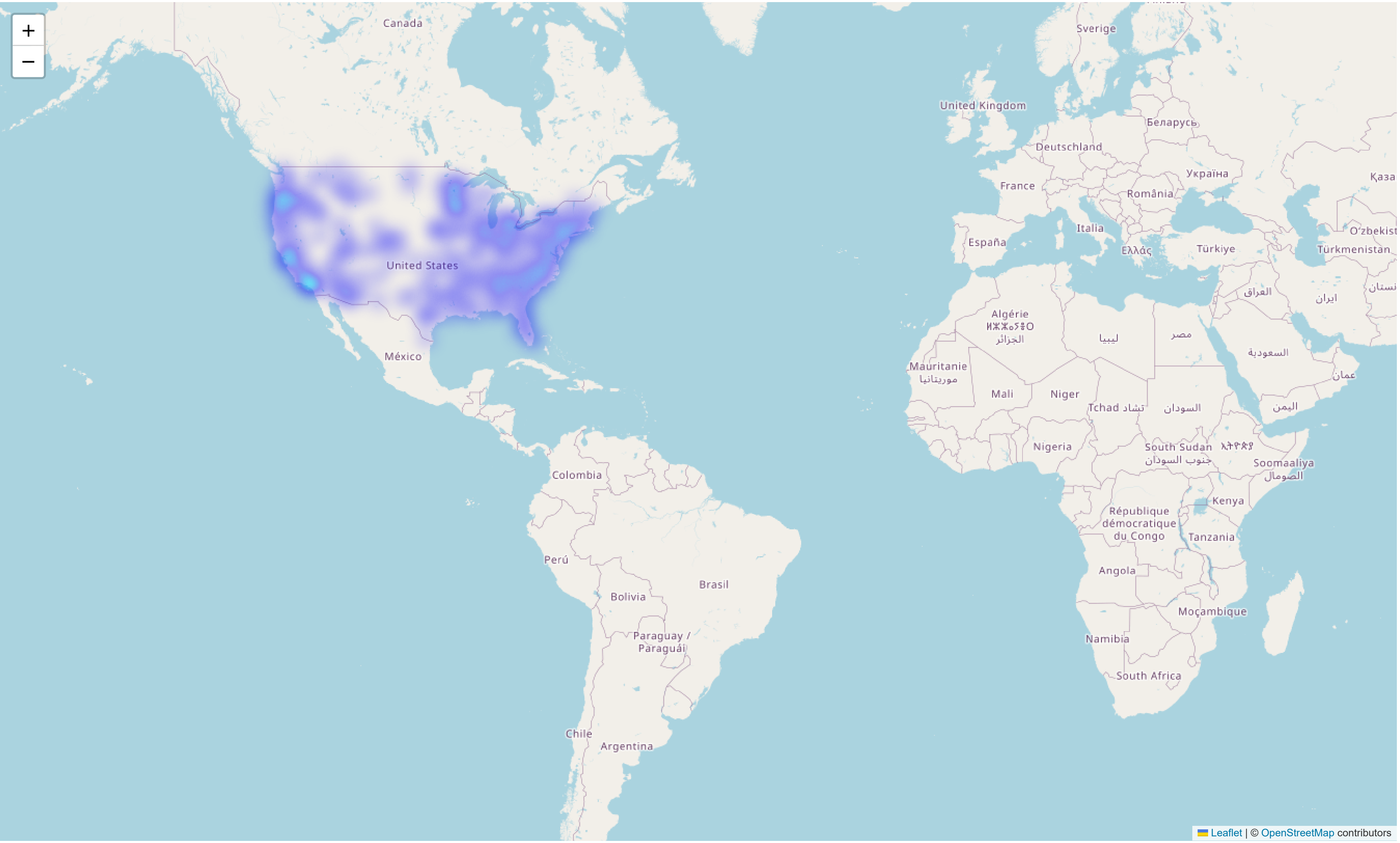

Accidents by Hour of Day
Number of Accidents
0
100000
200000
300000
400000
500000
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
13.0
14.0
15.0
16.0
17.0
18.0
19.0
20.0
21.0
22.0
23.0
Hour


Top Weather Conditions During Accidents
Weather Condition
Fair
Mostly Cloudy
Clear
Cloudy
Partly Cloudy
Overcast
Light Rain
Scattered Clouds


Light Snow
Fog
0.0
0.5
1.0
1.5
2.0
Count
1e6

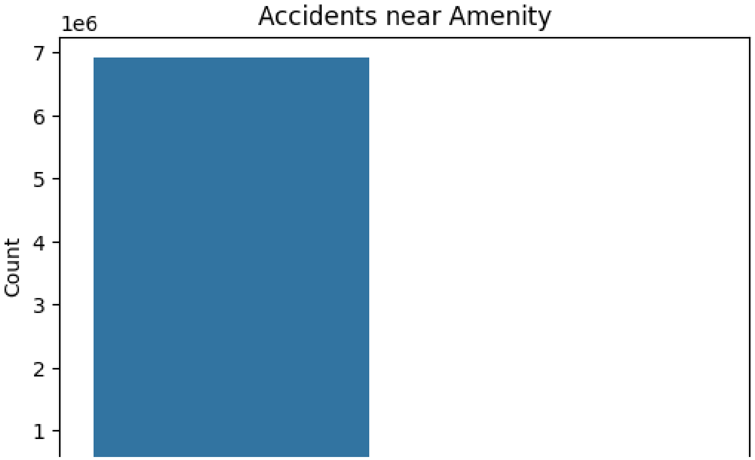

Distribution of Accident Severity Levels
1e6
Number of Accidents
0
1
2
3
4
5
1
2
3
4
Severity Level

```
heat_df = df[['Start_Lat', 'Start_Lng']].dropna().sample(n=5000, random_state=1)
heat_map = folium.Map(location=[heat_df['Start_Lat'].mean(), heat_df['Start_Lng'].mean()], zoom_start=5)
HeatMap(heat_df.values, radius=6).add_to(heat_map)

# Display map directly in notebook
heat_map
```



```
contributing_cols = ['Amenity', 'Crossing', 'Junction', 'Traffic_Signal']

for col in contributing_cols:
    print(f"\n{col} Impact:")
    print(df[col].value_counts())
    plt.figure(figsize=(6,4))
    sns.countplot(x=col, data=df)
    plt.title(f'Accidents near {col}')
    plt.ylabel('Count')
    plt.show()
```

```
Amenity Impact:
Amenity
False    6897635
True       87593
Name: count, dtype: int64
```


Accidents near Amenity
1e6
Count