In [3]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [7]:
```
df = pd.read_excel(r"D:\NIRALI\Nirali College DSBA\Road Accident Data.csv.xlsx")
print(df.head())
```

```
  Accident_Index        Accident Date Day_of_Week          Junction_Control  \
0  200901BS70001  2021-01-01 00:00:00    Thursday  Give way or uncontrolled
1  200901BS70002  2021-05-01 00:00:00      Monday  Give way or uncontrolled
2  200901BS70003  2021-04-01 00:00:00      Sunday  Give way or uncontrolled
3  200901BS70004  2021-05-01 00:00:00      Monday       Auto traffic signal
4  200901BS70005  2021-06-01 00:00:00     Tuesday       Auto traffic signal

           Junction_Detail Accident_Severity   Latitude  \
0  T or staggered junction           Serious  51.512273
1              Crossroads           Serious  51.514399
2  T or staggered junction            Slight  51.486668
3  T or staggered junction           Serious  51.507804
4              Crossroads           Serious  51.482076

       Light_Conditions Local_Authority_(District) Carriageway_Hazards  ...  \
0              Daylight     Kensington and Chelsea                 NaN  ...
1              Daylight     Kensington and Chelsea                 NaN  ...
2              Daylight     Kensington and Chelsea                 NaN  ...
3              Daylight     Kensington and Chelsea                 NaN  ...
4  Darkness - lights lit     Kensington and Chelsea                 NaN  ...

   Number_of_Casualties  Number_of_Vehicles         Police_Force  \
0                     1                   2  Metropolitan Police
1                    11                   2  Metropolitan Police
2                     1                   2  Metropolitan Police
3                     1                   2  Metropolitan Police
4                     1                   2  Metropolitan Police

  Road_Surface_Conditions           Road_Type Speed_limit      Time  \
0                     Dry      One way street          30  15:11:00
1             Wet or damp  Single carriageway          30  10:59:00
2                     Dry  Single carriageway          30  14:19:00
3            Frost or ice  Single carriageway          30  08:10:00
4                     Dry  Single carriageway          30  17:25:00

  Urban_or_Rural_Area  Weather_Conditions           Vehicle_Type
0               Urban  Fine no high winds                    Car
1               Urban  Fine no high winds  Taxi/Private hire car
2               Urban  Fine no high winds  Taxi/Private hire car
3               Urban               Other  Motorcycle over 500cc
4               Urban  Fine no high winds                    Car

[5 rows x 21 columns]
```

In [8]:
```
print(df.columns)
```

```
Index(['Accident_Index', 'Accident Date', 'Day_of_Week', 'Junction_Control',
       'Junction_Detail', 'Accident_Severity', 'Latitude', 'Light_Conditions',
       'Local_Authority_(District)', 'Carriageway_Hazards', 'Longitude',
       'Number_of_Casualties', 'Number_of_Vehicles', 'Police_Force',
       'Road_Surface_Conditions', 'Road_Type', 'Speed_limit', 'Time',
       'Urban_or_Rural_Area', 'Weather_Conditions', 'Vehicle_Type'],
      dtype='object')
```

In [9]:

```python
# Summary statistics
print(df.describe(include='all'))

# Check missing values
print(df.isnull().sum())

# Accident severity distribution
print(df['Accident_Severity'].value_counts())
```

	Accident_Index	Accident Date	Day_of_Week	Junction_Control	\
count	3.079730e+05	307973	307973	307973	
unique	1.976440e+05	730	7	7	
top	2.010000e+12	11/13/2021	Friday	Give way or uncontrolled	
freq	1.103040e+05	692	50529	150045	
mean	NaN	NaN	NaN	NaN	
std	NaN	NaN	NaN	NaN	
min	NaN	NaN	NaN	NaN	
25%	NaN	NaN	NaN	NaN	
50%	NaN	NaN	NaN	NaN	
75%	NaN	NaN	NaN	NaN	
max	NaN	NaN	NaN	NaN	

	Junction_Detail	Accident_Severity	Latitude	\
count	307973	307973	307973.000000	
unique	9	4	NaN	
top	Not at junction or within 20 metres	Slight	NaN	
freq	123094	263280	NaN	
mean	NaN	NaN	52.487005	
std	NaN	NaN	1.339011	
min	NaN	NaN	49.914488	
25%	NaN	NaN	51.485248	
50%	NaN	NaN	52.225943	
75%	NaN	NaN	53.415517	
max	NaN	NaN	60.598055	

	Light_Conditions	Local_Authority_(District)	Carriageway_Hazards	...	\
count	307973	307973	5424	...	
unique	5	422	5	...	
top	Daylight	Birmingham	Other object on road	...	
freq	227286	6165	2243	...	
mean	NaN	NaN	NaN	...	
std	NaN	NaN	NaN	...	
min	NaN	NaN	NaN	...	
25%	NaN	NaN	NaN	...	
50%	NaN	NaN	NaN	...	
75%	NaN	NaN	NaN	...	
max	NaN	NaN	NaN	...	

	Number_of_Casualties	Number_of_Vehicles	Police_Force	\
count	307973.000000	307973.000000	307973	
unique	NaN	NaN	51	
top	NaN	NaN	Metropolitan Police	
freq	NaN	NaN	46789	
mean	1.356882	1.829063	NaN	
std	0.815857	0.710477	NaN	
min	1.000000	1.000000	NaN	
25%	1.000000	1.000000	NaN	
50%	1.000000	2.000000	NaN	
75%	1.000000	2.000000	NaN	
max	48.000000	32.000000	NaN	

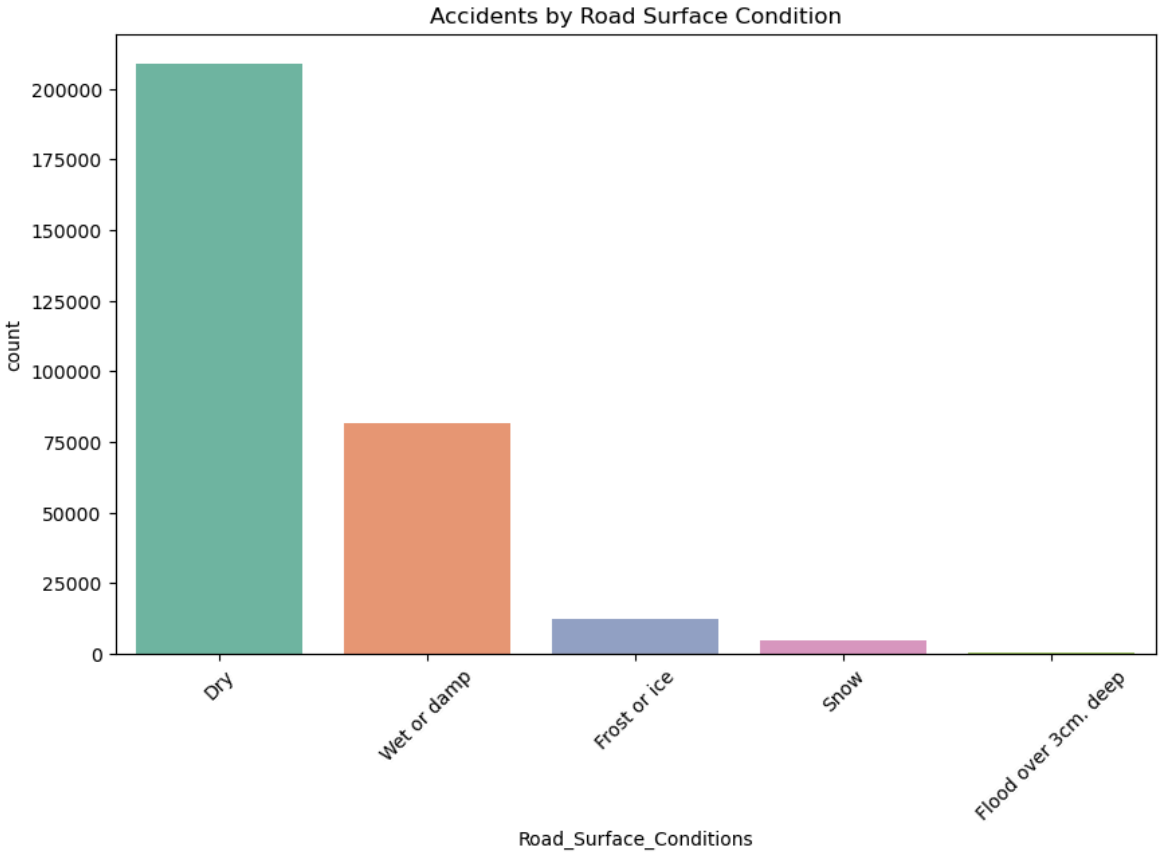
	Road_Surface_Conditions	Road_Type	Speed_limit	Time	\
count	307656	306439	307973.000000	307956	
unique	5	5	NaN	1439	
top	Dry	Single carriageway	NaN	17:00:00	
freq	208967	230612	NaN	2933	
mean	NaN	NaN	38.866037	NaN	
std	NaN	NaN	14.032933	NaN	
min	NaN	NaN	10.000000	NaN	

```
25%                          NaN                NaN      30.000000       NaN
50%                          NaN                NaN      30.000000       NaN
75%                          NaN                NaN      50.000000       NaN
max                          NaN                NaN      70.000000       NaN

       Urban_or_Rural_Area  Weather_Conditions Vehicle_Type
count               307973              301916       307973
unique                   2                   8           15
top                  Urban  Fine no high winds          Car
freq                198532              244496       239794
mean                   NaN                 NaN          NaN
std                    NaN                 NaN          NaN
min                    NaN                 NaN          NaN
25%                    NaN                 NaN          NaN
50%                    NaN                 NaN          NaN
75%                    NaN                 NaN          NaN
max                    NaN                 NaN          NaN

[11 rows x 21 columns]
Accident_Index                    0
Accident Date                     0
Day_of_Week                       0
Junction_Control                  0
Junction_Detail                   0
Accident_Severity                 0
Latitude                          0
Light_Conditions                  0
Local_Authority_(District)        0
Carriageway_Hazards          302549
Longitude                         0
Number_of_Casualties              0
Number_of_Vehicles                0
Police_Force                      0
Road_Surface_Conditions         317
Road_Type                      1534
Speed_limit                       0
Time                             17
Urban_or_Rural_Area               0
Weather_Conditions             6057
Vehicle_Type                      0
dtype: int64
Accident_Severity
Slight     263280
Serious     40740
Fatal        3904
Fetal          49
Name: count, dtype: int64
```

In [10]:

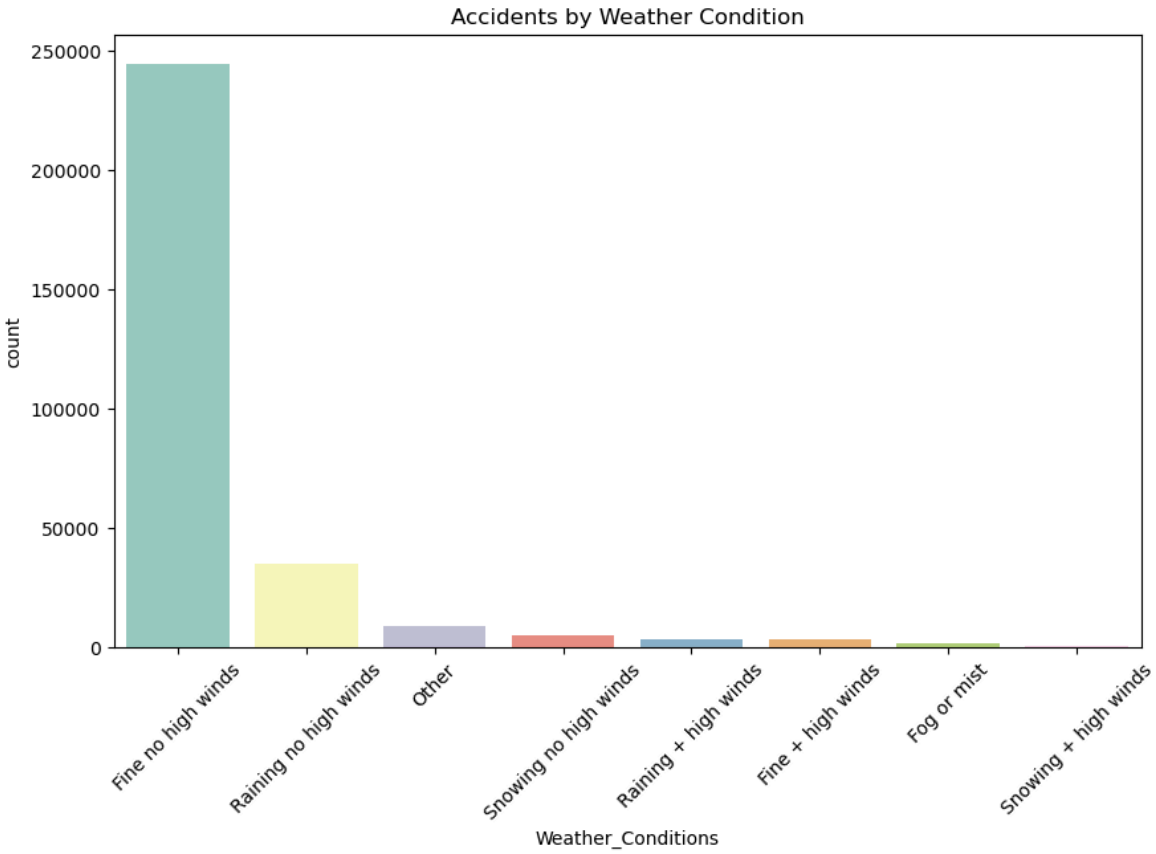
```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(10,6))
sns.countplot(data=df, x='Road_Surface_Conditions', order=df['Road_Surface_Condi
plt.title("Accidents by Road Surface Condition")
plt.xticks(rotation=45)
plt.show()
```

```
C:\Users\ADMIN\AppData\Local\Temp\ipykernel_14200\2488167669.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.countplot(data=df, x='Road_Surface_Conditions', order=df['Road_Surface_Cond
itions'].value_counts().index, palette="Set2")
```



In [11]:
```
plt.figure(figsize=(10,6))
sns.countplot(data=df, x='Weather_Conditions', order=df['Weather_Conditions'].va
plt.title("Accidents by Weather Condition")
plt.xticks(rotation=45)
plt.show()
```

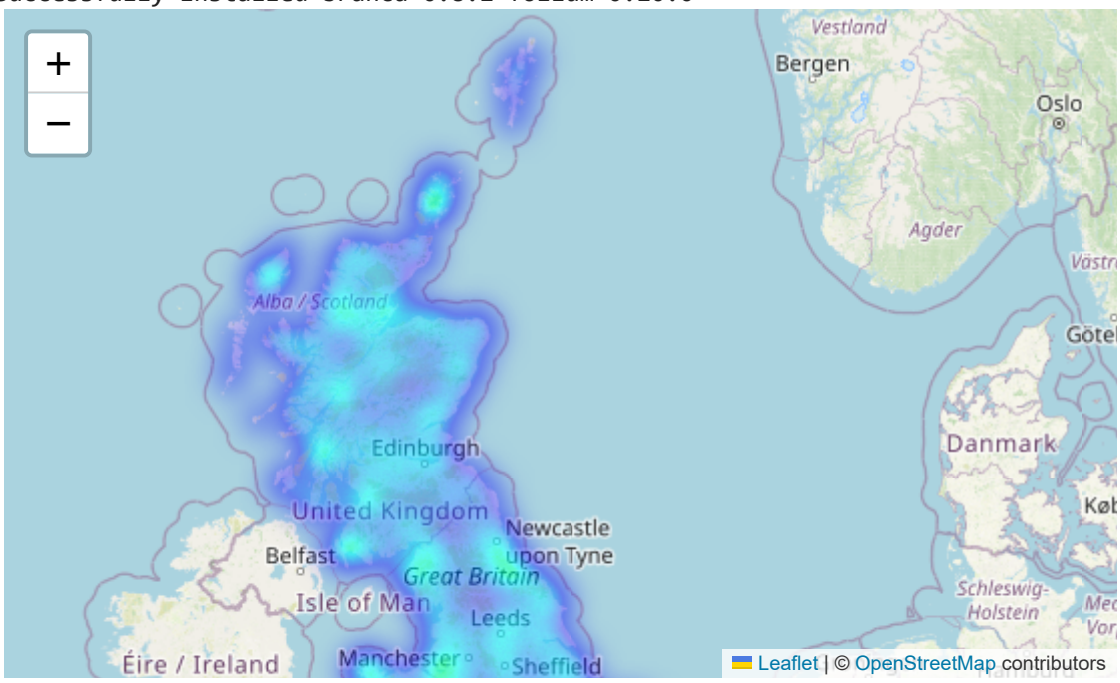
```
C:\Users\ADMIN\AppData\Local\Temp\ipykernel_14200\3894682365.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.countplot(data=df, x='Weather_Conditions', order=df['Weather_Conditions'].v
alue_counts().index, palette="Set3")
```



In [16]:
```
!pip install folium
import folium
from folium.plugins import HeatMap

# Center map on average location
m = folium.Map(location=[df['Latitude'].mean(), df['Longitude'].mean()], zoom_st

# Add heatmap
HeatMap(data=df[['Latitude','Longitude']], radius=8).add_to(m)

m
```

```
Collecting folium
  Downloading folium-0.20.0-py2.py3-none-any.whl.metadata (4.2 kB)
Collecting branca>=0.6.0 (from folium)
  Downloading branca-0.8.2-py3-none-any.whl.metadata (1.7 kB)
Requirement already satisfied: jinja2>=2.9 in c:\users\admin\anaconda3\lib\site-p
ackages (from folium) (3.1.6)
Requirement already satisfied: numpy in c:\users\admin\anaconda3\lib\site-package
s (from folium) (2.1.3)
Requirement already satisfied: requests in c:\users\admin\anaconda3\lib\site-pack
ages (from folium) (2.32.3)
Requirement already satisfied: xyzservices in c:\users\admin\anaconda3\lib\site-p
ackages (from folium) (2022.9.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\admin\anaconda3\lib\si
te-packages (from jinja2>=2.9->folium) (3.0.2)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\admin\anacond
a3\lib\site-packages (from requests->folium) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in c:\users\admin\anaconda3\lib\site-
packages (from requests->folium) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\admin\anaconda3\lib
\site-packages (from requests->folium) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\admin\anaconda3\lib
\site-packages (from requests->folium) (2025.7.14)
Downloading folium-0.20.0-py2.py3-none-any.whl (113 kB)
Downloading branca-0.8.2-py3-none-any.whl (26 kB)
Installing collected packages: branca, folium

   ---------------------- ------------------- 1/2 [folium]
   ---------------------------------------- 2/2 [folium]

Successfully installed branca-0.8.2 folium-0.20.0
```

Out[16]:



In [ ]: