PRODIGY INFOTECH INTERNSHIP

TASK 5:Analyze traffic accident data to identify patterns related to road conditions, weather, and time of day. Visualize accident hotspots and contributing factors.

DONE BY: Krupa J Shetty

Email:krupajshetty@gmail.com

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

IMPORTING THE DATASET

```
data=pd.read_csv('accident data.csv')
data
```

	Index	Accident_Severity	Accident Date	Latitude	Light_Conditions	Dist
0	200701BS64157	Serious	05-06-2019	51.506187	Darkness - lights lit	Kensi Ch
1	200701BS65737	Serious	02-07-2019	51.495029	Daylight	Kensi Ch
2	200701BS66127	Serious	26-08-2019	51.517715	Darkness - lighting unknown	Kensi Ch
3	200701BS66128	Serious	16-08-2019	51.495478	Daylight	Kensi Ch
4	200701BS66837	Slight	03-09-2019	51.488576	Darkness - lights lit	Kensi Ch
...	...	...	...	...	...	
75158	200720K088502	Slight	30-08-2019	52.506153	Daylight	Sar
75159	200720K088602	Slight	01-09-2019	52.500668	Daylight	Sar
75160	200720K089001	Slight	28-10-2019	52.509656	Daylight	Sar
75161	200720K089201	Serious	30-10-2019	52.551004	Daylight	Sar
75162	200720K089301	Slight	30-10-2019	52.539593	Daylight	Sar

75163 rows × 14 columns

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 75163 entries, 0 to 75162
Data columns (total 14 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Index                    75163 non-null  object
 1   Accident_Severity        75163 non-null  object
 2   Accident Date            75163 non-null  object
 3   Latitude                 75163 non-null  float64
 4   Light_Conditions         75163 non-null  object
 5   District Area            75163 non-null  object
 6   Longitude                75163 non-null  float64
 7   Number_of_Casualties     75163 non-null  int64
 8   Number_of_Vehicles       75163 non-null  int64
 9   Road_Surface_Conditions  75141 non-null  object
 10  Road_Type                74347 non-null  object
 11  Urban_or_Rural_Area      75163 non-null  object
 12  Weather_Conditions       73713 non-null  object
 13  Vehicle_Type             75162 non-null  object
```

```
dtypes: float64(2), int64(2), object(10)
memory usage: 8.0+ MB
```

```
data.describe(include='all')
```

```
/usr/local/lib/python3.10/dist-packages/lida/components/summarizer.py:74: UserWarning
  cast_date_col = pd.to_datetime(df[column], errors='coerce')
```

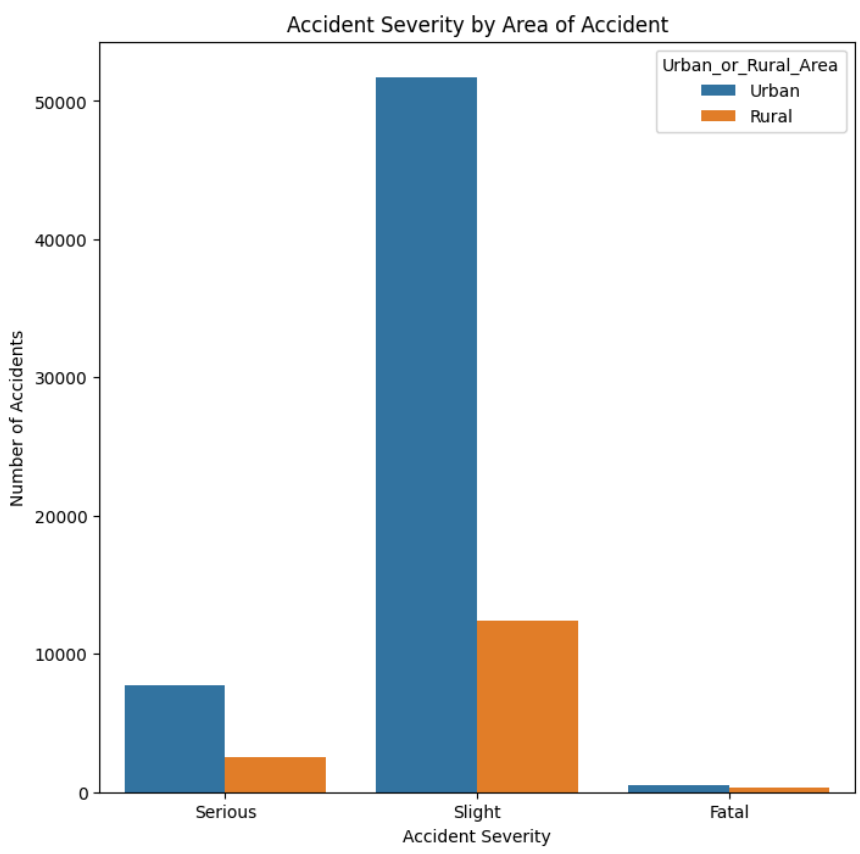
	Index	Accident_Severity	Accident Date	Latitude	Light_Conditions	Distr A
count	75163	75163	75163	75163.000000	75163	75
unique	56217	3	365	NaN	5	
top	2.01E+12	Slight	13-07-2019	NaN	Daylight	Birmingl
freq	18947	64078	299	NaN	54821	3
mean	NaN	NaN	NaN	52.999004	NaN	I
std	NaN	NaN	NaN	1.140564	NaN	I
min	NaN	NaN	NaN	51.292694	NaN	I
25%	NaN	NaN	NaN	51.569584	NaN	I
50%	NaN	NaN	NaN	53.415654	NaN	I
75%	NaN	NaN	NaN	53.758457	NaN	I
max	NaN	NaN	NaN	55.785544	NaN	I

```
data.head(10)
```

	Index	Accident_Severity	Accident Date	Latitude	Light_Conditions	District Area
0	200701BS64157	Serious	05-06-2019	51.506187	Darkness - lights lit	Kensington and Chelsea
1	200701BS65737	Serious	02-07-2019	51.495029	Daylight	Kensington and Chelsea
2	200701BS66127	Serious	26-08-2019	51.517715	Darkness - lighting unknown	Kensington and Chelsea
3	200701BS66128	Serious	16-08-2019	51.495478	Daylight	Kensington and Chelsea
4	200701BS66837	Slight	03-09-2019	51.488576	Darkness - lights lit	Kensington and Chelsea
5	200701BS67159	Serious	18-09-2019	51.497750	Daylight	Kensington and Chelsea
6	200701BS67207	Serious	05-09-2019	51.501405	Daylight	Kensington and Chelsea
7	200701BS67370	Fatal	03-10-2019	51.482260	Darkness - lights lit	Kensington and Chelsea
8	200701BS67515	Slight	31-10-2019	51.493319	Darkness - lights lit	Kensington and Chelsea
9	200701BS67543	Slight	18-10-2019	51.484539	Daylight	Kensington and Chelsea

ANALYSIS

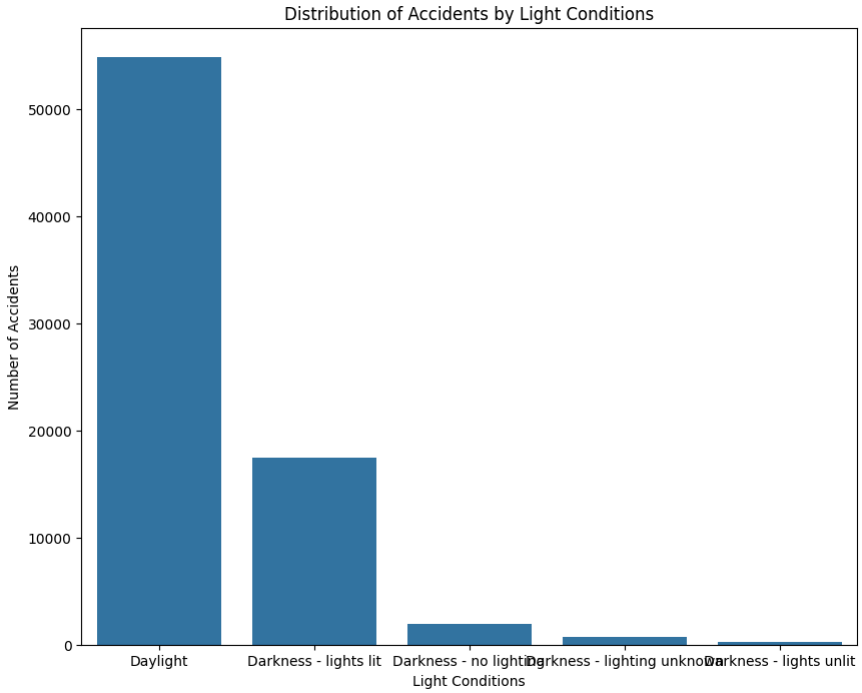
```
plt.figure(figsize=(8,8))
sns.countplot(x='Accident_Severity',data=data,hue='Urban_or_Rural_Area')
plt.title('Accident Severity by Area of Accident')
plt.xlabel('Accident Severity')
plt.ylabel('Number of Accidents')
plt.show()
```



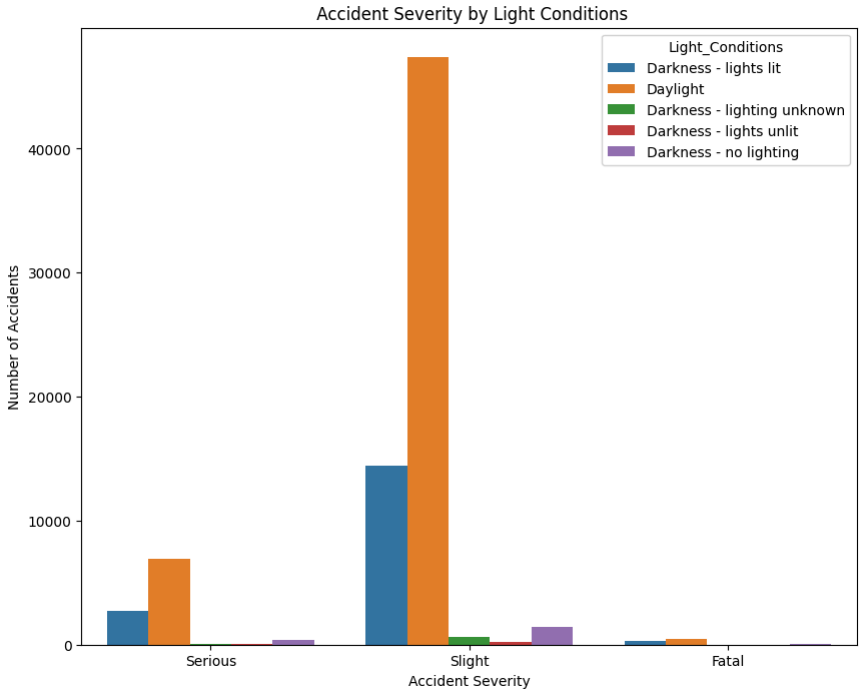

```
light_conditions_counts=data['Light_Conditions'].value_counts()
print(light_conditions_counts)

plt.figure(figsize=(10,8))
sns.barplot(x=light_conditions_counts.index, y=light_conditions_counts.values)
plt.title('Distribution of Accidents by Light Conditions')
plt.xlabel('Light Conditions')
plt.ylabel('Number of Accidents')
plt.show()
```

```
Daylight                       54821
Darkness - lights lit          17472
Darkness - no lighting          1967
Darkness - lighting unknown      676
Darkness - lights unlit          227
Name: Light_Conditions, dtype: int64
```



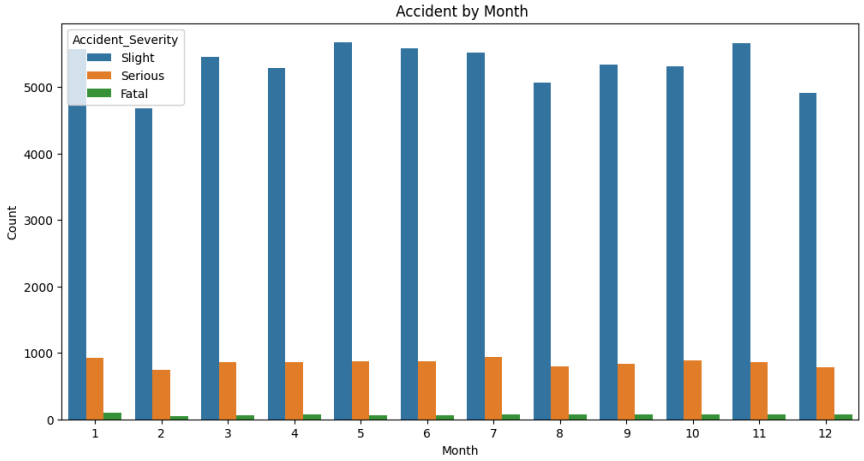

```
plt.figure(figsize=(10,8))
sns.countplot(x='Accident_Severity', data=data,hue='Light_Conditions')
plt.title(' Accident Severity by Light Conditions')
plt.xlabel('Accident Severity')
plt.ylabel('Number of Accidents')
plt.show()
```



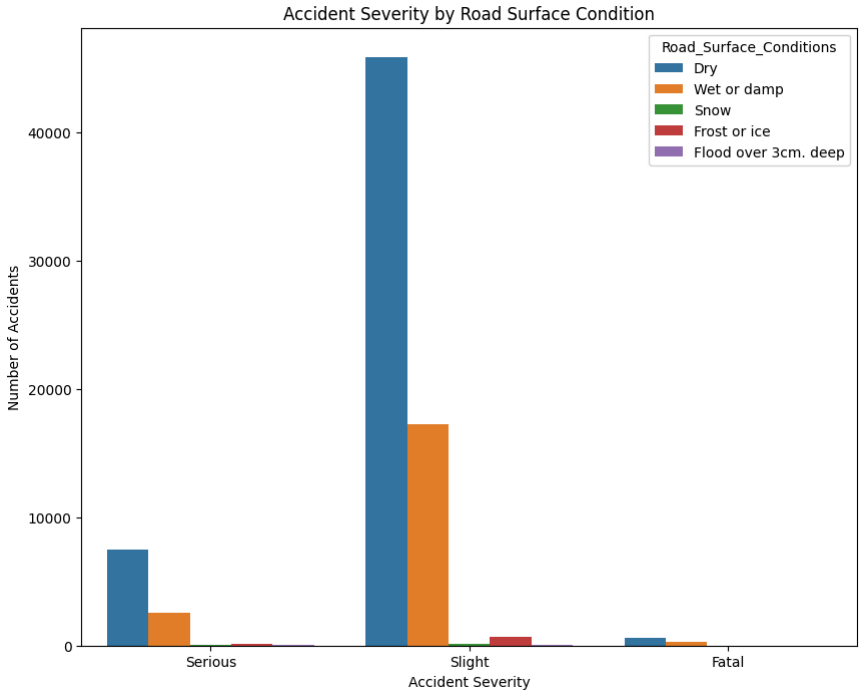
TIME ANALYSIS

```
data['Accident Date']=pd.to_datetime(data['Accident Date'])
data['Month']=data['Accident Date'].dt.month
plt.figure(figsize=(12,6))
sns.countplot(x='Month',data=data,hue='Accident_Severity')
plt.title(' Accident by Month')
plt.xlabel('Month')
plt.ylabel('Count')
plt.show()
```

```
<ipython-input-12-891eba8307c4>:1: UserWarning: Parsing dates in DD/MM/YYYY format wh
  data['Accident Date']=pd.to_datetime(data['Accident Date'])
```




```
plt.figure(figsize=(10,8))
sns.countplot(x='Accident_Severity',data=data,hue='Road_Surface_Conditions')
plt.title('Accident Severity by Road Surface Condition')
plt.xlabel('Accident Severity')
plt.ylabel('Number of Accidents')
plt.show()
```

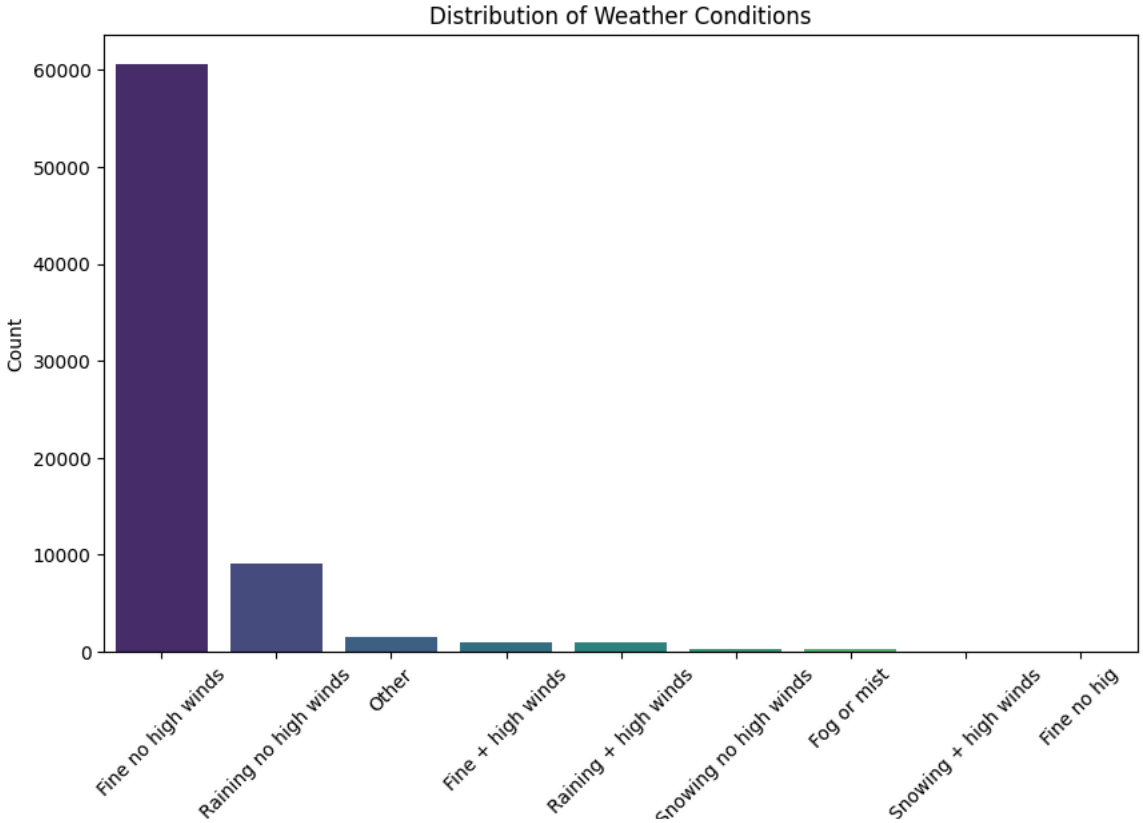


```
plt.figure(figsize=(10,6))
sns.countplot(x='Weather_Conditions',data=data,palette='viridis')
plt.title('Distribution of Weather Conditions')
plt.xlabel('Weather Conditions')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```

```
<ipython-input-14-e29ce067e67b>:2: FutureWarning:

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

  sns.countplot(x='Weather_Conditions',data=data,palette='viridis')
```

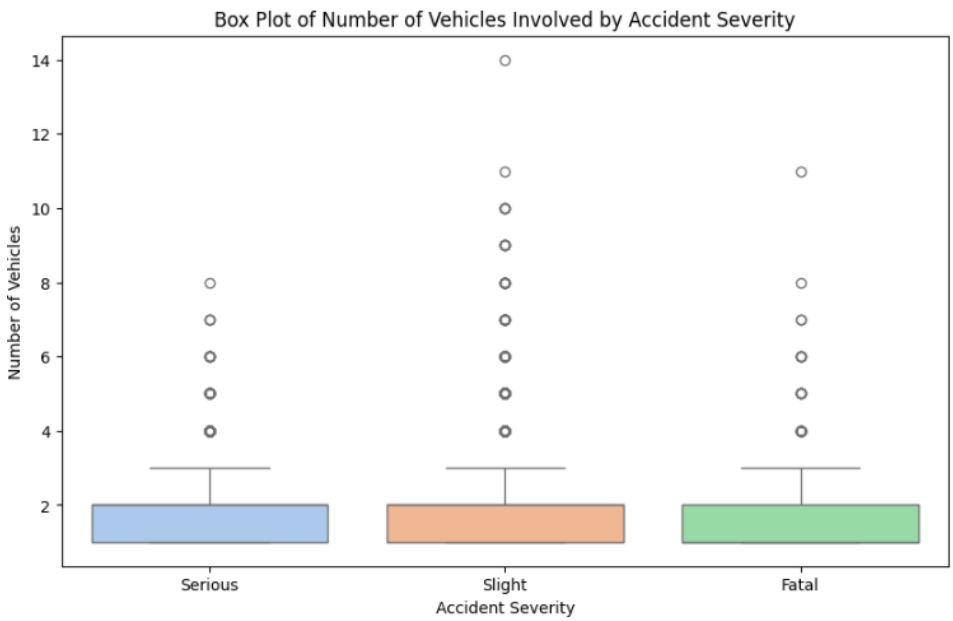



```
plt.figure(figsize=(10,6))
sns.boxplot(x='Accident_Severity',y='Number_of_Vehicles',data=data,palette='pastel')
plt.title('Box Plot of Number of Vehicles Involved by Accident Severity')
plt.xlabel('Accident Severity')
plt.ylabel('Number of Vehicles')
plt.show()
```

```
<ipython-input-16-a756784b405d>:2: FutureWarning:

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

  sns.boxplot(x='Accident_Severity',y='Number_of_Vehicles',data=data,palette='pastel')
```



Inference:- The number of vehicles involved in accidents tends to increase with the severity of the accident. This is likely because more serious accidents often involve multiple vehicles colliding with each other, while slight accidents may only involve two vehicles.

The distribution of the number of vehicles involved in accidents is relatively symmetrical for all three severities. There are a few outliers in each of the severity categories.These outliers could be due to factors such as chain-reaction accidents or accidents involving large trucks or buses.

```
plt.figure(figsize=(10,8))
columns=['Latitude','Longitude','Number_of_Casualties','Number_of_Vehicles','Month']
correlation_matrix=data[columns].corr()
sns.heatmap(correlation_matrix,annot=True,cmap='coolwarm',linewidths=.5)
plt.title('Correlation Matrix')
plt.show()
```