

road-accidents-analysis

September 5, 2024

1 ROAD ACCIDENTS ANALYSIS

```
[1]: #importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
[2]: #reading the file
df=pd.read_csv(r'C:\Users\Windows\Desktop\projects\RTA Dataset.csv')
df.head()
```

```
[2]:      Time Day_of_week Age_band_of_driver Sex_of_driver Educational_level \
0  17:02:00      Monday      18-30           Male  Above high school
1  17:02:00      Monday      31-50           Male  Junior high school
2  17:02:00      Monday      18-30           Male  Junior high school
3   1:06:00       Sunday      18-30           Male  Junior high school
4   1:06:00       Sunday      18-30           Male  Junior high school
```

```
      Vehicle_driver_relation Driving_experience      Type_of_vehicle \
0      Employee      1-2yr      Automobile
1      Employee  Above 10yr  Public (> 45 seats)
2      Employee      1-2yr      Lorry (41?100Q)
3      Employee      5-10yr  Public (> 45 seats)
4      Employee      2-5yr      NaN
```

```
      Owner_of_vehicle Service_year_of_vehicle ... Vehicle_movement \
0      Owner      Above 10yr ...  Going straight
1      Owner      5-10yrs ...  Going straight
2      Owner      NaN ...  Going straight
3  Governmental      NaN ...  Going straight
4      Owner      5-10yrs ...  Going straight
```

```
      Casualty_class Sex_of_casualty Age_band_of_casualty Casualty_severity \
0      na      na      na      na
1      na      na      na      na
2  Driver or rider      Male      31-50      3
```

3	Pedestrian	Female	18-30	3
4	na	na	na	na

	Work_of_casuality	Fitness_of_casuality	Pedestrian_movement	\
0	NaN	NaN	Not a Pedestrian	
1	NaN	NaN	Not a Pedestrian	
2	Driver	NaN	Not a Pedestrian	
3	Driver	Normal	Not a Pedestrian	
4	NaN	NaN	Not a Pedestrian	

	Cause_of_accident	Accident_severity
0	Moving Backward	Slight Injury
1	Overtaking	Slight Injury
2	Changing lane to the left	Serious Injury
3	Changing lane to the right	Slight Injury
4	Overtaking	Slight Injury

[5 rows x 32 columns]

2 Great!! Let's Move to EDA

```
[3]: df.shape
```

```
[3]: (12316, 32)
```

```
[4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12316 entries, 0 to 12315
Data columns (total 32 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Time                                  12316 non-null  object
1   Day_of_week                           12316 non-null  object
2   Age_band_of_driver                    12316 non-null  object
3   Sex_of_driver                         12316 non-null  object
4   Educational_level                      11575 non-null  object
5   Vehicle_driver_relation                11737 non-null  object
6   Driving_experience                     11487 non-null  object
7   Type_of_vehicle                       11366 non-null  object
8   Owner_of_vehicle                      11834 non-null  object
9   Service_year_of_vehicle               8388 non-null   object
10  Defect_of_vehicle                     7889 non-null   object
11  Area_accident_occured                 12077 non-null  object
12  Lanes_or_Medians                     11931 non-null  object
13  Road_allignment                      12174 non-null  object
14  Types_of_Junction                    11429 non-null  object
```

```

15 Road_surface_type          12144 non-null object
16 Road_surface_conditions    12316 non-null object
17 Light_conditions           12316 non-null object
18 Weather_conditions         12316 non-null object
19 Type_of_collision           12161 non-null object
20 Number_of_vehicles_involved 12316 non-null int64
21 Number_of_casualties        12316 non-null int64
22 Vehicle_movement           12008 non-null object
23 Casualty_class              12316 non-null object
24 Sex_of_casualty             12316 non-null object
25 Age_band_of_casualty        12316 non-null object
26 Casualty_severity           12316 non-null object
27 Work_of_casualty            9118 non-null object
28 Fitness_of_casualty         9681 non-null object
29 Pedestrian_movement         12316 non-null object
30 Cause_of_accident           12316 non-null object
31 Accident_severity           12316 non-null object
dtypes: int64(2), object(30)
memory usage: 3.0+ MB

```

```
[5]: df.columns
```

```

[5]: Index(['Time', 'Day_of_week', 'Age_band_of_driver', 'Sex_of_driver',
        'Educational_level', 'Vehicle_driver_relation', 'Driving_experience',
        'Type_of_vehicle', 'Owner_of_vehicle', 'Service_year_of_vehicle',
        'Defect_of_vehicle', 'Area_accident_occured', 'Lanes_or_Medians',
        'Road_allignment', 'Types_of_Junction', 'Road_surface_type',
        'Road_surface_conditions', 'Light_conditions', 'Weather_conditions',
        'Type_of_collision', 'Number_of_vehicles_involved',
        'Number_of_casualties', 'Vehicle_movement', 'Casualty_class',
        'Sex_of_casualty', 'Age_band_of_casualty', 'Casualty_severity',
        'Work_of_casualty', 'Fitness_of_casualty', 'Pedestrian_movement',
        'Cause_of_accident', 'Accident_severity'],
        dtype='object')

```

```
[6]: df.dtypes
```

```

[6]: Time          object
    Day_of_week    object
    Age_band_of_driver  object
    Sex_of_driver   object
    Educational_level  object
    Vehicle_driver_relation  object
    Driving_experience  object
    Type_of_vehicle    object
    Owner_of_vehicle    object
    Service_year_of_vehicle  object

```

Defect_of_vehicle	object
Area_accident_occured	object
Lanes_or_Medians	object
Road_allignment	object
Types_of_Junction	object
Road_surface_type	object
Road_surface_conditions	object
Light_conditions	object
Weather_conditions	object
Type_of_collision	object
Number_of_vehicles_involved	int64
Number_of_casualties	int64
Vehicle_movement	object
Casualty_class	object
Sex_of_casualty	object
Age_band_of_casualty	object
Casualty_severity	object
Work_of_casualty	object
Fitness_of_casualty	object
Pedestrian_movement	object
Cause_of_accident	object
Accident_severity	object
dtype:	object

```
[7]: #cheking for null values
df.isnull().sum().sort_values(ascending = False)
```

```
[7]: Defect_of_vehicle      4427
Service_year_of_vehicle   3928
Work_of_casualty          3198
Fitness_of_casualty       2635
Type_of_vehicle           950
Types_of_Junction         887
Driving_experience         829
Educational_level         741
Vehicle_driver_relation    579
Owner_of_vehicle          482
Lanes_or_Medians          385
Vehicle_movement          308
Area_accident_occured     239
Road_surface_type         172
Type_of_collision         155
Road_allignment           142
Sex_of_driver              0
Age_band_of_driver        0
Day_of_week                0
Time                       0
```

```

Weather_conditions      0
Number_of_vehicles_involved  0
Light_conditions        0
Road_surface_conditions  0
Casualty_class          0
Number_of_casualties     0
Age_band_of_casualty    0
Sex_of_casualty         0
Casualty_severity       0
Pedestrian_movement     0
Cause_of_accident       0
Accident_severity       0
dtype: int64

```

```
[8]: #checking for duplicates
df.duplicated().sum()
```

```
[8]: np.int64(0)
```

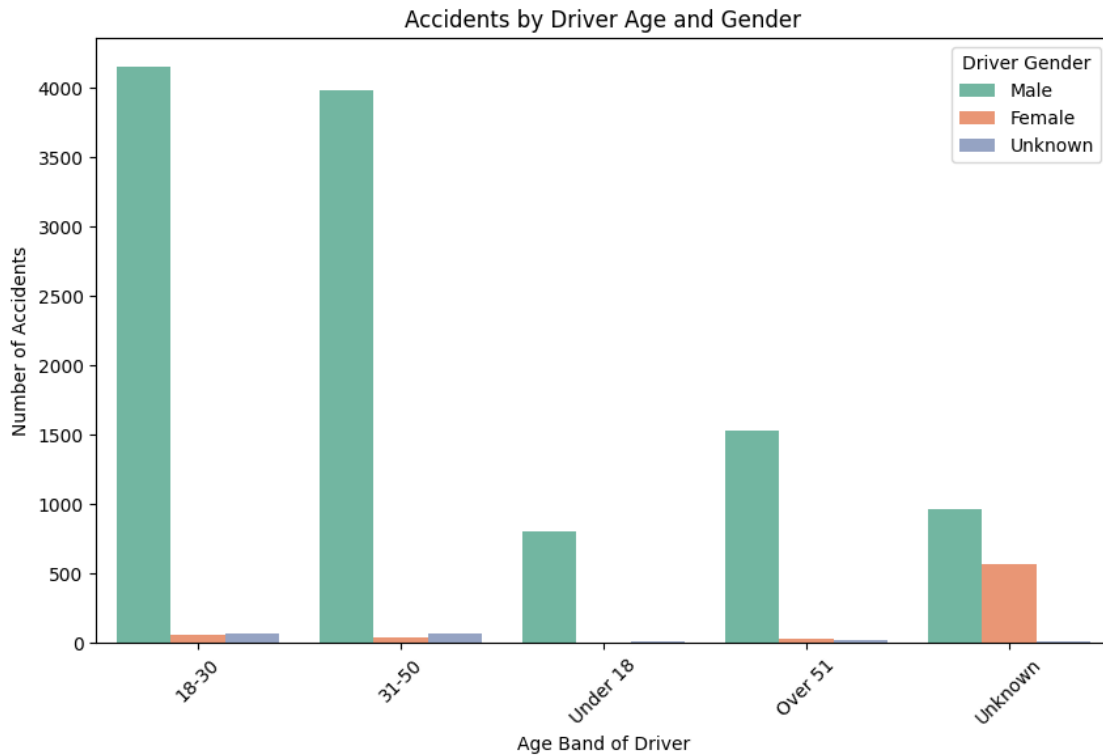
```
[9]: df.fillna(df.mode(),inplace = True)
```

```
[10]: # Convert the 'Time' column to datetime format and extract the hour
df['Hour'] = pd.to_datetime(df['Time'], format='%H:%M:%S', errors='coerce').dt.
    ↪hour
```

3 Let's Dive into Visualisations

4 Visualizing accident distribution by Age and Gender of the drivers

```
[11]: plt.figure(figsize=(10,6))
sns.countplot(data=df, x='Age_band_of_driver', hue='Sex_of_driver',
    ↪palette='Set2')
plt.title('Accidents by Driver Age and Gender')
plt.xlabel('Age Band of Driver')
plt.ylabel('Number of Accidents')
plt.xticks(rotation=45)
plt.legend(title='Driver Gender')
plt.show()
```



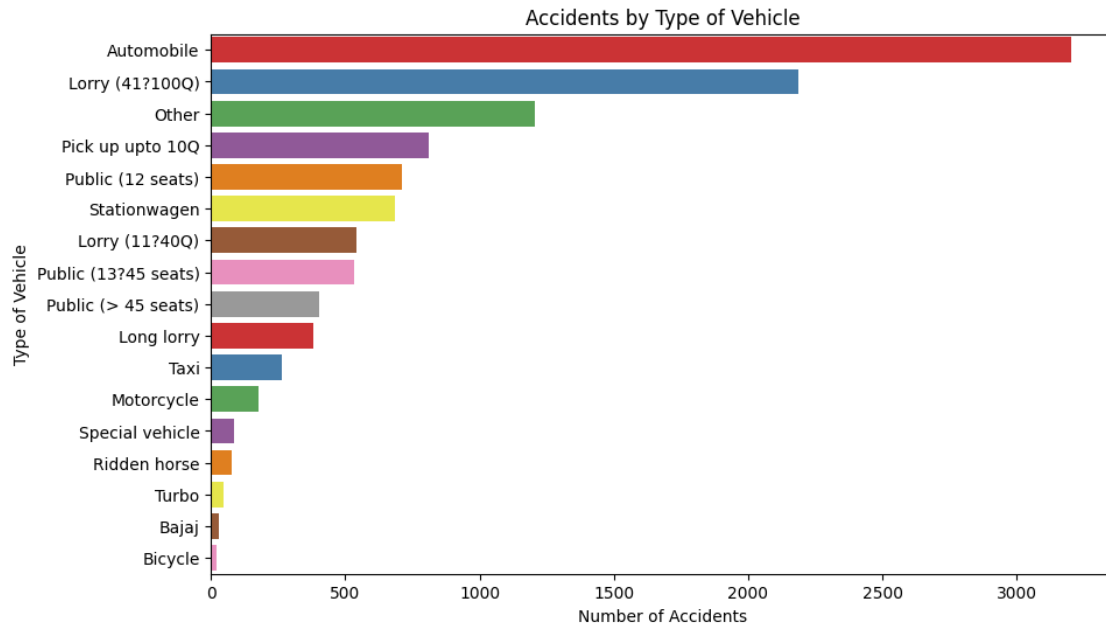
5 Visualizing accident distribution by Type of Vehicle involved

```
[12]: plt.figure(figsize=(10,6))
sns.countplot(data=df, y='Type_of_vehicle', order=df['Type_of_vehicle'].
    ↳value_counts().index, palette='Set1')
plt.title('Accidents by Type of Vehicle')
plt.xlabel('Number of Accidents')
plt.ylabel('Type of Vehicle')
plt.show()
```

C:\Users\Windows\AppData\Local\Temp\ipykernel_11244\727022823.py:2:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.countplot(data=df, y='Type_of_vehicle',
order=df['Type_of_vehicle'].value_counts().index, palette='Set1')
```



6 Accidents by Day of the Week

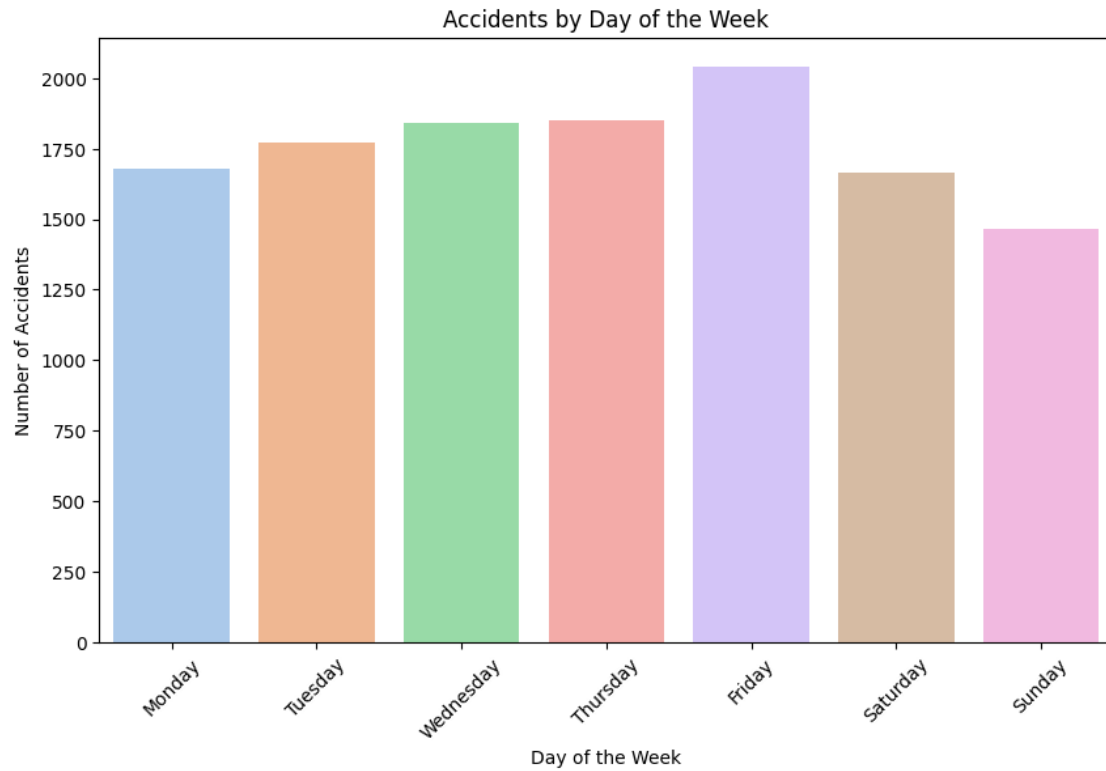
```
[13]: plt.figure(figsize=(10,6))
sns.countplot(data=df, x='Day_of_week', order=['Monday', 'Tuesday', '
        ↪ 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'], palette='pastel')
plt.title('Accidents by Day of the Week')
plt.xlabel('Day of the Week')
plt.ylabel('Number of Accidents')
plt.xticks(rotation=45)
plt.show()
```

C:\Users\Windows\AppData\Local\Temp\ipykernel_11244\3369223899.py:2:

FutureWarning:

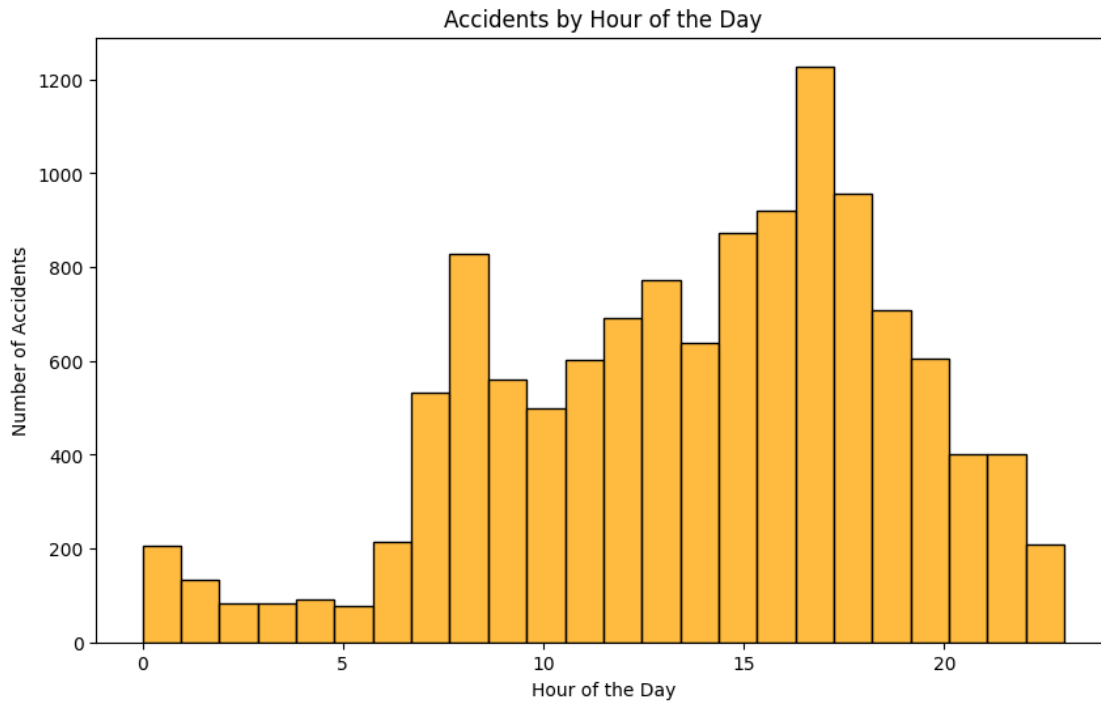
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.countplot(data=df, x='Day_of_week', order=['Monday', 'Tuesday',
'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'], palette='pastel')
```



7 Accidents by Hour of the Day

```
[14]: plt.figure(figsize=(10,6))
sns.histplot(df['Hour'].dropna(), bins=24, kde=False, color='orange')
plt.title('Accidents by Hour of the Day')
plt.xlabel('Hour of the Day')
plt.ylabel('Number of Accidents')
plt.show()
```

Accidents by Weather Condition

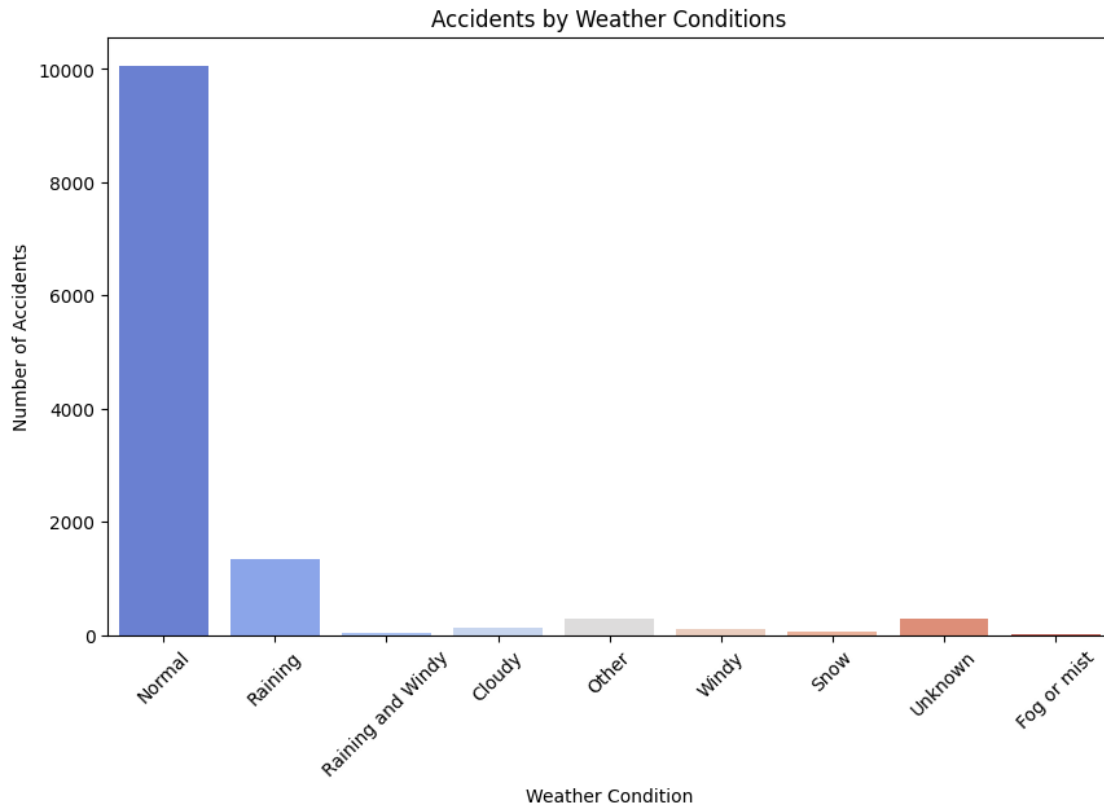
```
[15]: plt.figure(figsize=(10,6))
sns.countplot(data=df, x='Weather_conditions', palette='coolwarm')
plt.title('Accidents by Weather Conditions')
plt.xlabel('Weather Condition')
plt.ylabel('Number of Accidents')
plt.xticks(rotation=45)
plt.show()
```

C:\Users\Windows\AppData\Local\Temp\ipykernel_11244\2741901610.py: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 `legend=False` for the same effect.

```
sns.countplot(data=df, x='Weather_conditions', palette='coolwarm')
```



8 Accidents by Light Condition

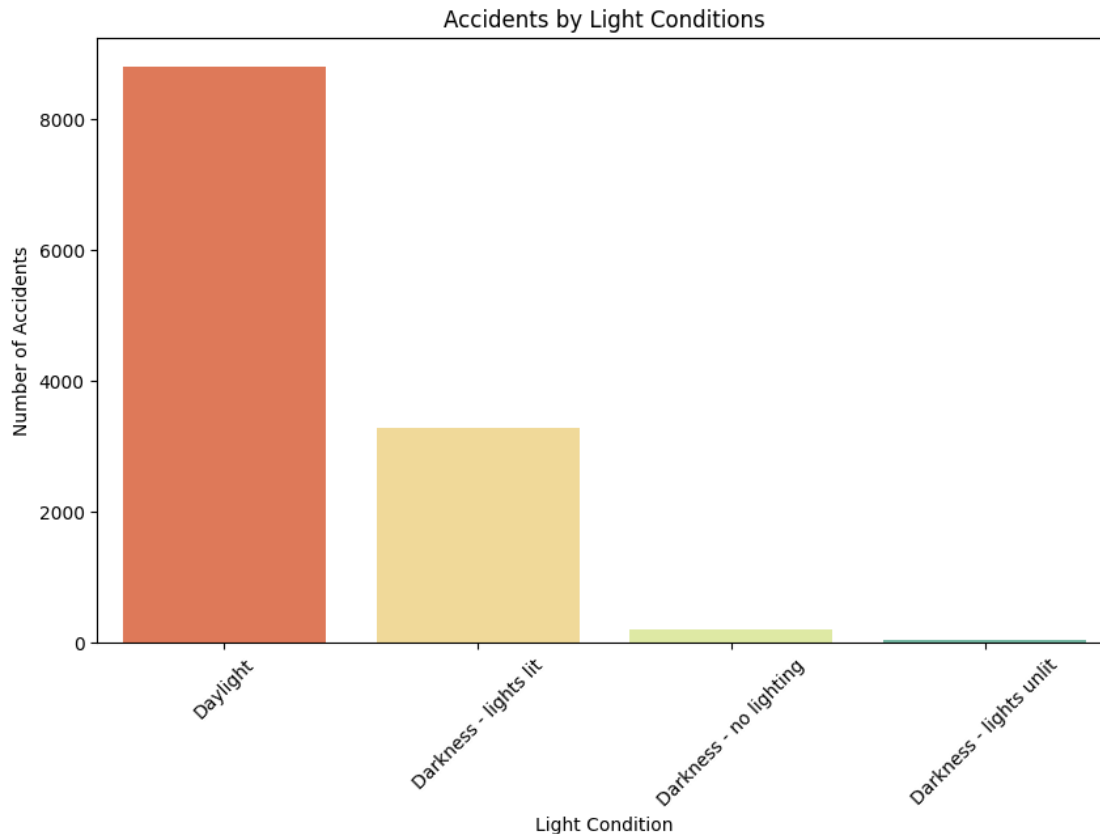
```
[16]: plt.figure(figsize=(10,6))
sns.countplot(data=df, x='Light_conditions', palette='Spectral')
plt.title('Accidents by Light Conditions')
plt.xlabel('Light Condition')
plt.ylabel('Number of Accidents')
plt.xticks(rotation=45)
plt.show()
```

C:\Users\Windows\AppData\Local\Temp\ipykernel_11244\1570878483.py: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 `legend=False` for the same effect.

```
sns.countplot(data=df, x='Light_conditions', palette='Spectral')
```



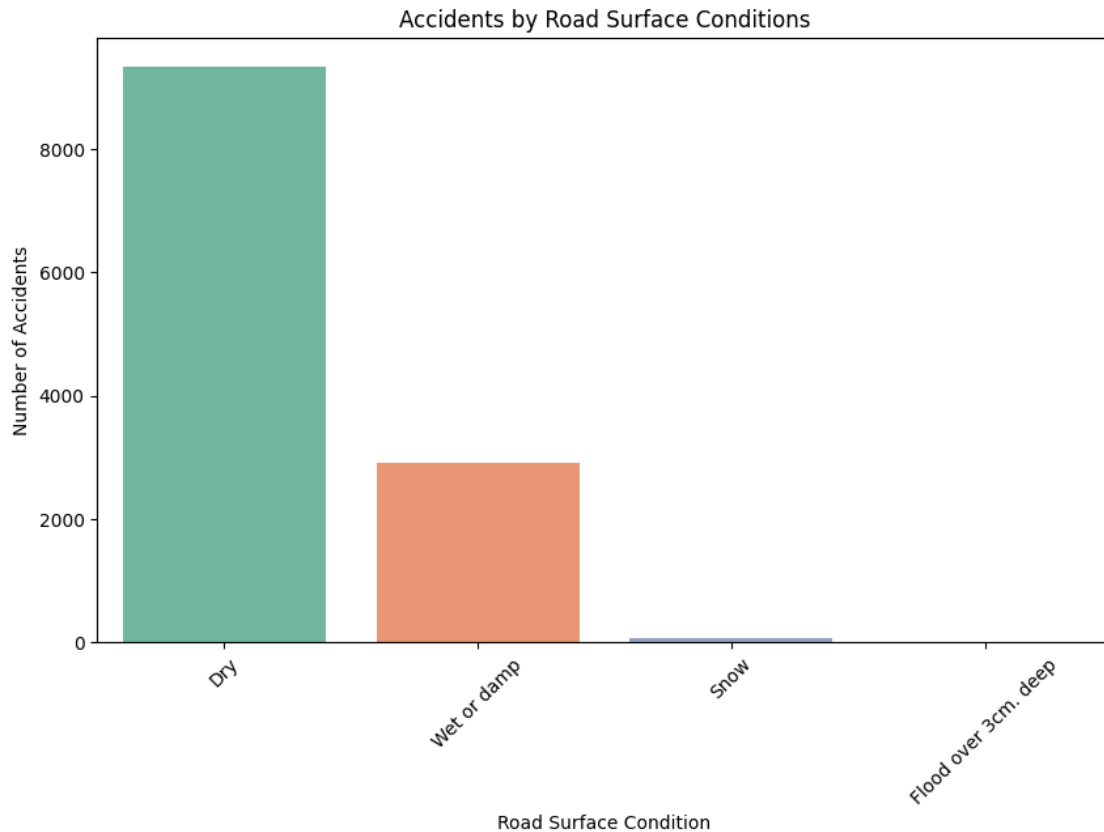
9 Accidents by Road Surface Condition

```
[17]: plt.figure(figsize=(10,6))
sns.countplot(data=df, x='Road_surface_conditions', palette='Set2')
plt.title('Accidents by Road Surface Conditions')
plt.xlabel('Road Surface Condition')
plt.ylabel('Number of Accidents')
plt.xticks(rotation=45)
plt.show()
```

C:\Users\Windows\AppData\Local\Temp\ipykernel_11244\3026959786.py:2:
FutureWarning:

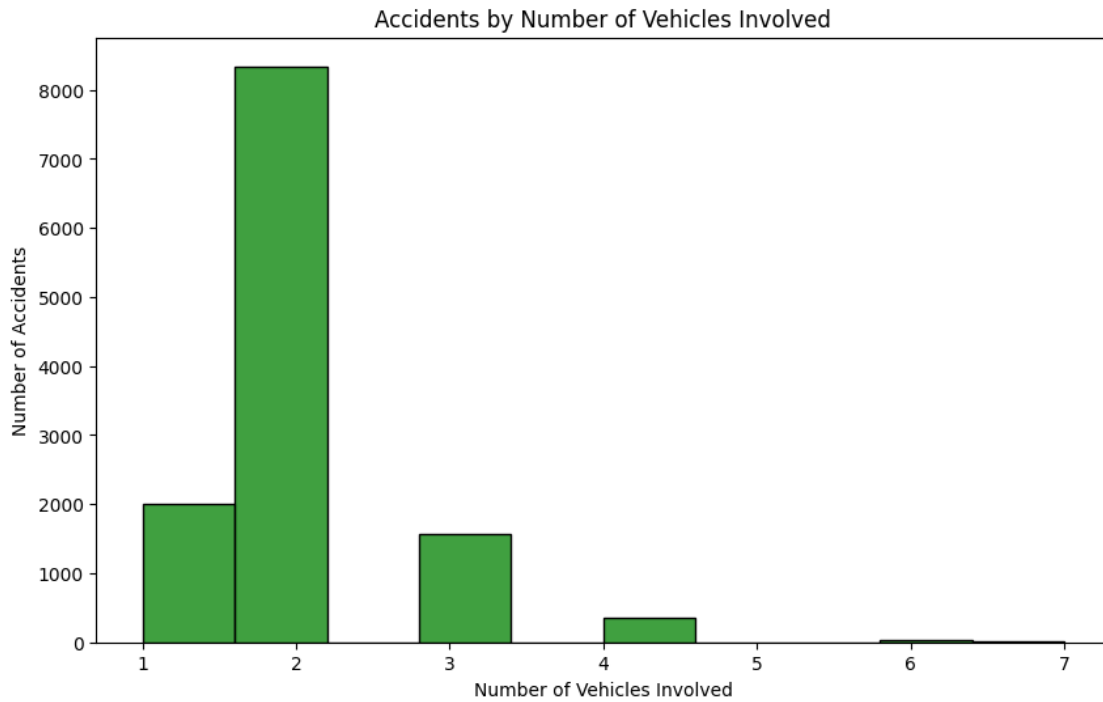
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.countplot(data=df, x='Road_surface_conditions', palette='Set2')
```



10 Accidents by Number of Vehicles Involved

```
[18]: plt.figure(figsize=(10,6))
sns.histplot(df['Number_of_vehicles_involved'], bins=10, kde=False, color='green')
plt.title('Accidents by Number of Vehicles Involved')
plt.xlabel('Number of Vehicles Involved')
plt.ylabel('Number of Accidents')
plt.show()
```



Accidents by Severity

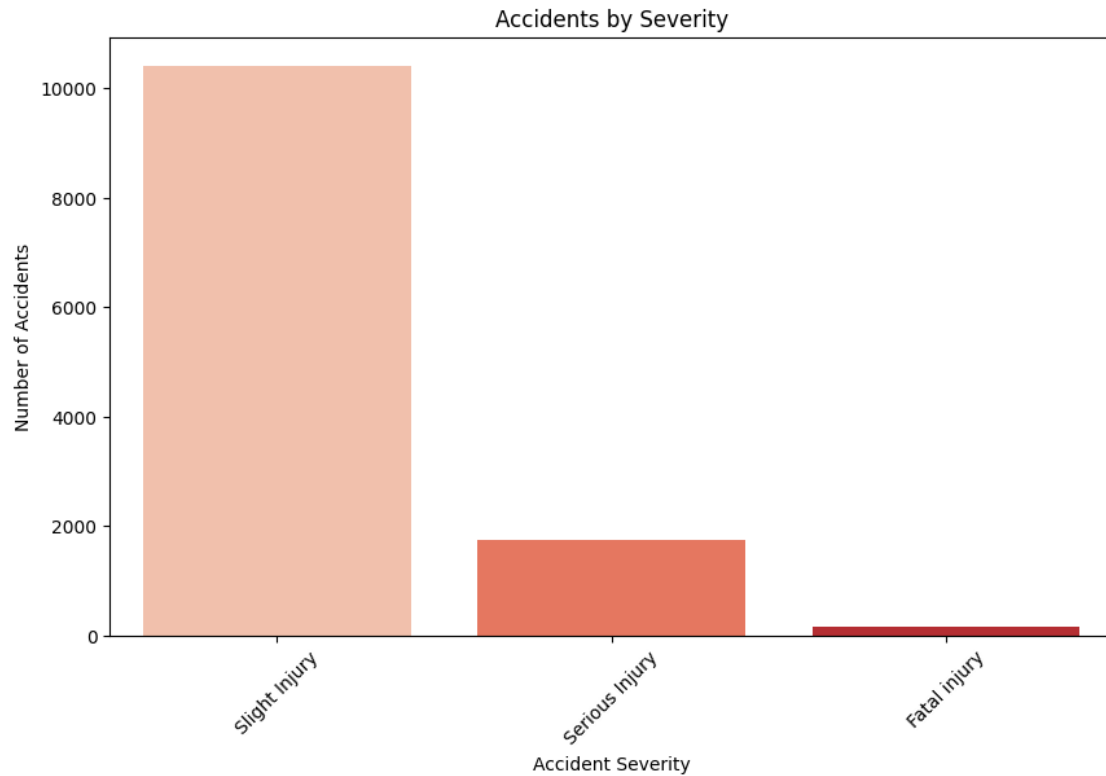
```
[19]: plt.figure(figsize=(10,6))
sns.countplot(data=df, x='Accident_severity', palette='Reds')
plt.title('Accidents by Severity')
plt.xlabel('Accident Severity')
plt.ylabel('Number of Accidents')
plt.xticks(rotation=45)
plt.show()
```

C:\Users\Windows\AppData\Local\Temp\ipykernel_11244\3350149324.py: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 `legend=False` for the same effect.

```
sns.countplot(data=df, x='Accident_severity', palette='Reds')
```



The road accident data shows that accidents peak on weekends, during rush hours, and in poor weather or low-light conditions. Younger drivers and motorcycles are more frequently involved, indicating riskier behavior and vulnerability. Poor road surfaces and multiple-vehicle accidents are also common, especially in congested areas. Urban areas see more minor incidents, while rural areas experience more severe accidents. These trends highlight the need for better infrastructure, stricter traffic regulations, and targeted driver education.

[]: