Overview: This dataset contains detailed information on traffic accidents across various regions and time periods. It includes various metrics such as accident date, weather conditions, lighting conditions, crash types, injuries, and vehicle involvement. The data spans multiple locations and accident types, offering a comprehensive view of traffic incidents and their causes.

https://www.kaggle.com/datasets/oktayrdeki/traffic-accidents/code

Objective

- **Accident Analysis:** Analyze accident trends, types, and the severity of injuries across different locations, time periods, and conditions.
- **Traffic Safety:** Understand the factors contributing to accidents (e.g., weather, lighting, road conditions) to inform traffic safety measures.
- **Predictive Modeling:** Use the dataset to forecast accident hotspots, potential injuries, and the impact of various factors on crash severity.

Characteristics of each columns

- **crash_date:** The date the accident occurred. traffic_control_device: The type of traffic control device involved (e.g., traffic light, sign).
- **weather_condition:** The weather conditions at the time of the accident.
- **lighting_condition:** The lighting conditions at the time of the accident.
- **first_crash_type:** The initial type of the crash (e.g., head-on, rear-end).
- **trafficway_type:** The type of roadway involved in the accident (e.g., highway, local road).
- **alignment:** The alignment of the road where the accident occurred (e.g., straight, curved).
- **roadway_surface_cond:** The condition of the roadway surface (e.g., dry, wet, icy).
- **road_defect:** Any defects present on the road surface.
- **crash_type:** The overall type of the crash.
- **intersection_related_i:** Whether the accident was related to an intersection.
- **damage:** The extent of the damage caused by the accident.
- **prim_contributory_cause:** The primary cause contributing to the crash.
- **num_units:** The number of vehicles involved in the accident.
- **most_severe_injury:** The most severe injury sustained in the crash.
- **injuries_total:** The total number of injuries reported.
- **injuries_fatal:** The number of fatal injuries resulting from the accident.
- **injuries_incapacitating:** The number of incapacitating injuries.
- **injuries_non_incapacitating:** The number of non-incapacitating injuries.
- **injuries_reported_not_evident:** The number of injuries reported but not visibly evident.
- **injuries_no_indication:** The number of cases with no indication of injury.
- **crash_hour:** The hour the accident occurred.
- **crash_day_of_week:** The day of the week the accident occurred.
- **crash_month:** The month the accident occurred.

```
import numpy as np
import pandas as pd
```

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

import warnings
warnings.filterwarnings('ignore')

data = pd.read_csv('traffic_accidents.csv')
data.sample(5)
```

{"type":"dataframe"}

```python
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209306 entries, 0 to 209305
Data columns (total 24 columns):
 #   Column                         Non-Null Count   Dtype
---  ------                         --------------   -----
 0   crash_date                     209306 non-null  object
 1   traffic_control_device         209306 non-null  object
 2   weather_condition              209306 non-null  object
 3   lighting_condition             209306 non-null  object
 4   first_crash_type               209306 non-null  object
 5   trafficway_type                209306 non-null  object
 6   alignment                      209306 non-null  object
 7   roadway_surface_cond           209306 non-null  object
 8   road_defect                    209306 non-null  object
 9   crash_type                     209306 non-null  object
 10  intersection_related_i         209306 non-null  object
 11  damage                         209306 non-null  object
 12  prim_contributory_cause        209306 non-null  object
 13  num_units                      209306 non-null  int64
 14  most_severe_injury             209306 non-null  object
 15  injuries_total                 209306 non-null  float64
 16  injuries_fatal                 209306 non-null  float64
 17  injuries_incapacitating        209306 non-null  float64
 18  injuries_non_incapacitating    209306 non-null  float64
 19  injuries_reported_not_evident  209306 non-null  float64
 20  injuries_no_indication         209306 non-null  float64
 21  crash_hour                     209306 non-null  int64
 22  crash_day_of_week              209306 non-null  int64
 23  crash_month                    209306 non-null  int64
dtypes: float64(6), int64(4), object(14)
memory usage: 38.3+ MB
```

```python
data.shape,data.size,data.ndim
```

```
((209306, 24), 5023344, 2)
```

```python
df = data.copy(deep=True)
```

```
df['crash_date'] = pd.to_datetime(df['crash_date'])
df['crash_min'] = df['crash_date'].dt.minute
df['crash_day_of_week'] = df['crash_date'].dt.day_name()
df['crash_year'] = df['crash_date'].dt.year
df['crash_month'] = df['crash_date'].dt.month_name()
df = df.drop(columns=['crash_date'])

df.sample(5)
```

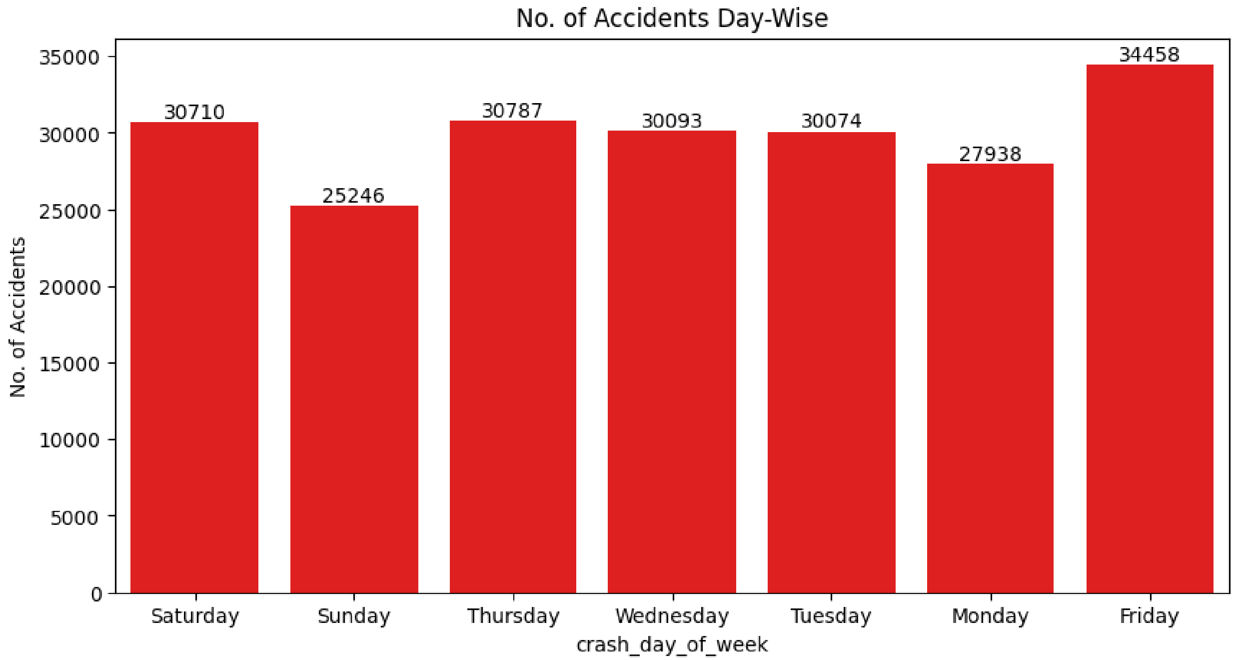
{"type":"dataframe"}

```
df.info()
```

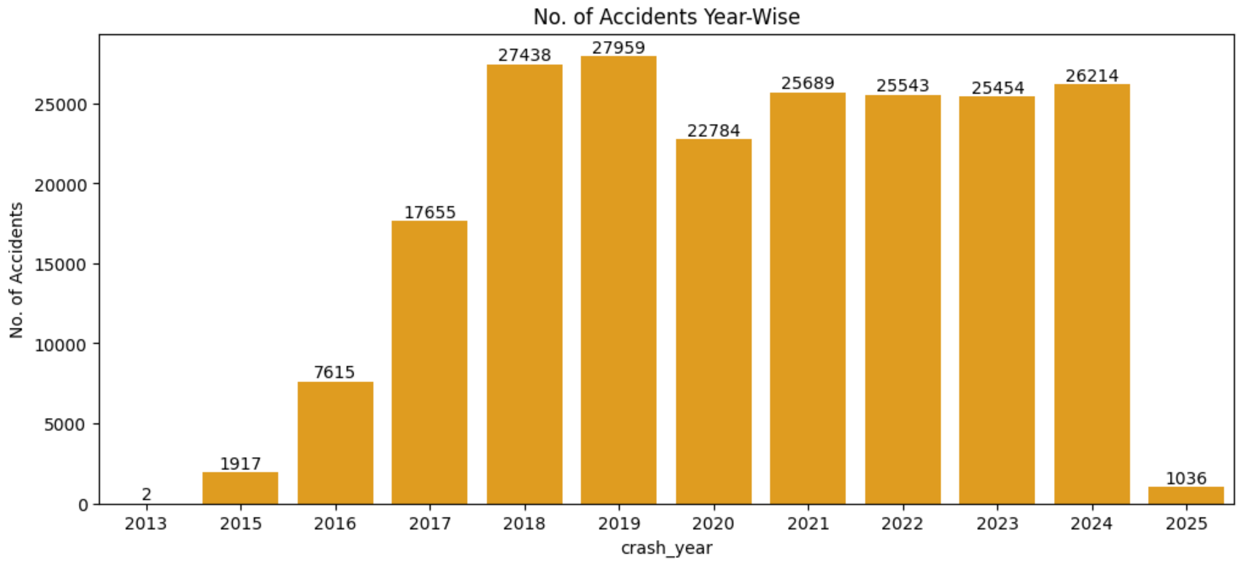
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209306 entries, 0 to 209305
Data columns (total 25 columns):
 #   Column                         Non-Null Count   Dtype
---  ------                         --------------   -----
 0   traffic_control_device         209306 non-null  object
 1   weather_condition              209306 non-null  object
 2   lighting_condition             209306 non-null  object
 3   first_crash_type               209306 non-null  object
 4   trafficway_type                209306 non-null  object
 5   alignment                      209306 non-null  object
 6   roadway_surface_cond           209306 non-null  object
 7   road_defect                    209306 non-null  object
 8   crash_type                     209306 non-null  object
 9   intersection_related_i         209306 non-null  object
 10  damage                         209306 non-null  object
 11  prim_contributory_cause        209306 non-null  object
 12  num_units                      209306 non-null  int64
 13  most_severe_injury             209306 non-null  object
 14  injuries_total                 209306 non-null  float64
 15  injuries_fatal                 209306 non-null  float64
 16  injuries_incapacitating        209306 non-null  float64
 17  injuries_non_incapacitating    209306 non-null  float64
 18  injuries_reported_not_evident  209306 non-null  float64
 19  injuries_no_indication         209306 non-null  float64
 20  crash_hour                     209306 non-null  int64
 21  crash_day_of_week              209306 non-null  object
 22  crash_month                    209306 non-null  object
 23  crash_min                      209306 non-null  int32
 24  crash_year                     209306 non-null  int32
dtypes: float64(6), int32(2), int64(2), object(15)
memory usage: 38.3+ MB
```

```
plt.figure(figsize=(10,5))
ax = sns.countplot(x=df['crash_day_of_week'],color='red')

for i in ax.containers:
```

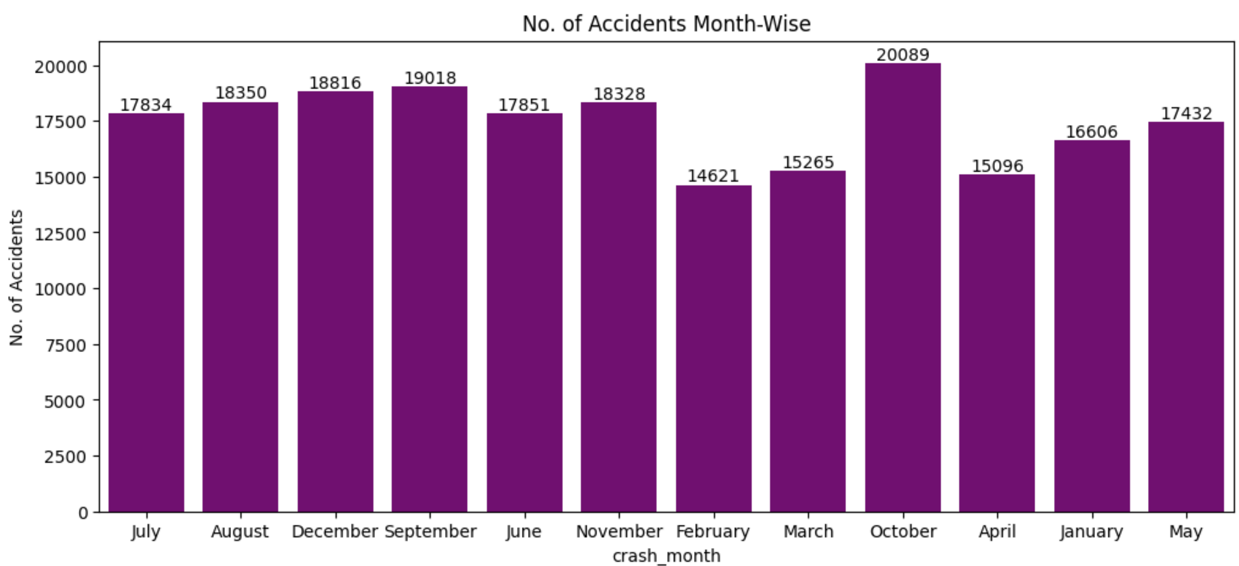
```
  ax.bar_label(i,)

plt.ylabel("No. of Accidents")
plt.title("No. of Accidents Day-Wise")
plt.show()
```




```
plt.figure(figsize=(12,5))
ax = sns.countplot(x=df['crash_year'],color='orange')

for i in ax.containers:
  ax.bar_label(i,)

plt.ylabel("No. of Accidents")
plt.title("No. of Accidents Year-Wise")
plt.show()
```

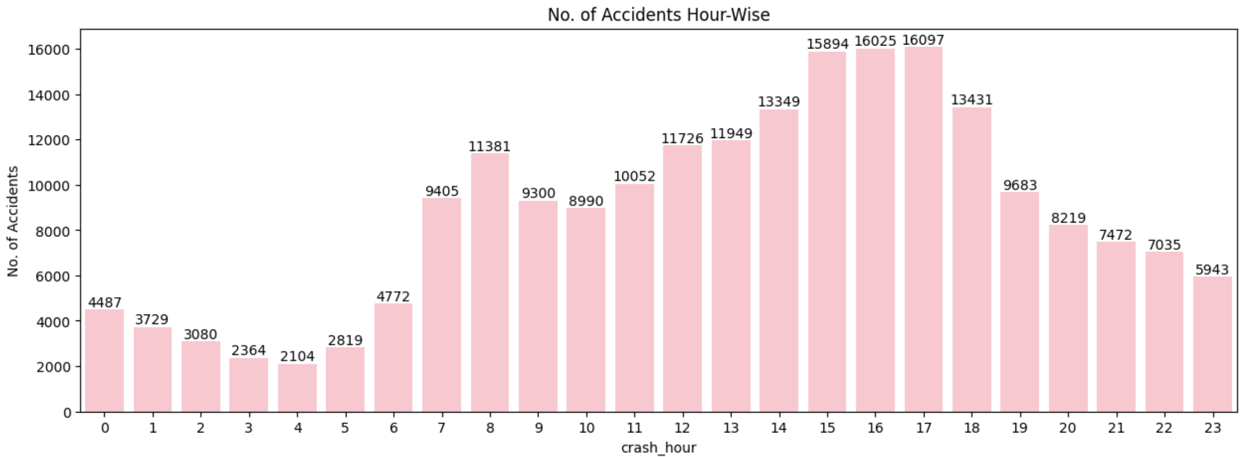



```
plt.figure(figsize=(12,5))
ax = sns.countplot(x=df['crash_month'],color='purple')

for i in ax.containers:
  ax.bar_label(i,)

plt.ylabel("No. of Accidents")
plt.title("No. of Accidents Month-Wise")
plt.show()
```




```
plt.figure(figsize=(15,5))
ax = sns.countplot(x=df['crash_hour'],color='pink')

for i in ax.containers:
```

```
  ax.bar_label(i,)

plt.ylabel("No. of Accidents")
plt.title("No. of Accidents Hour-Wise")
plt.show()
```




```
df.sample(2)
```

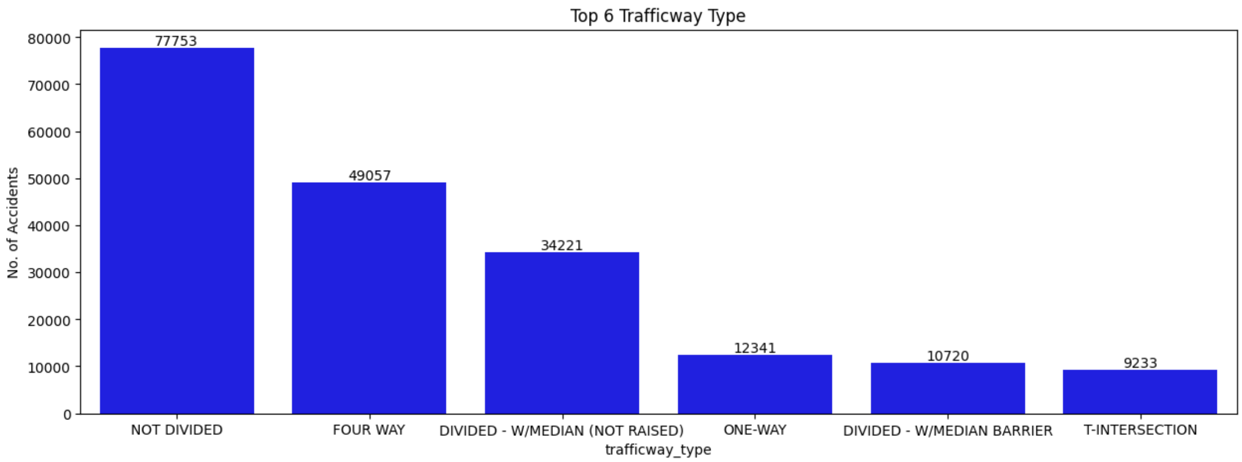
{"type":"dataframe"}

```
#We are onlt taking top 6 trafficway type as per values counts to make
our analysis easy
trafficway_type_top6 =
df['trafficway_type'].value_counts().head(6).reset_index()
trafficway_type_top6
```

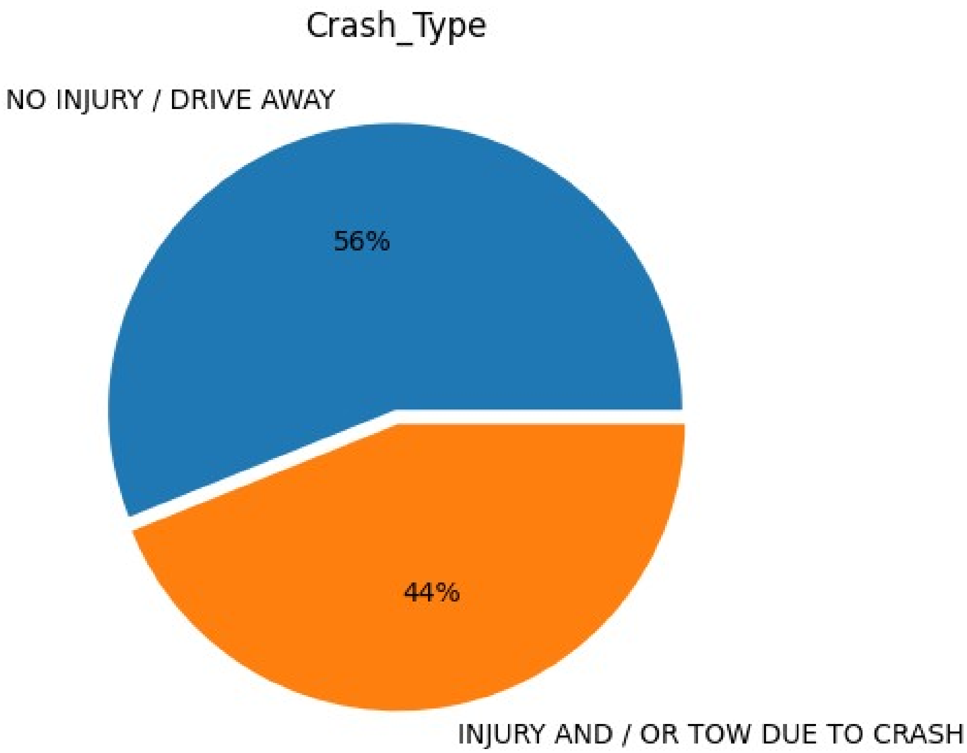
{"summary":"{\n \"name\": \"trafficway_type_top6\",\n \"rows\": 6,\n \"fields\": [\n   {\n   \"column\": \"trafficway_type\",\n   \"properties\": {\n   \"dtype\": \"string\",\n   \"num_unique_values\": 6,\n   \"samples\": [\n   \"NOT DIVIDED\",\n   \"FOUR WAY\",\n   \"T-INTERSECTION\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"count\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 27373,\n   \"min\": 9233,\n   \"max\": 77753,\n   \"num_unique_values\": 6,\n   \"samples\": [\n   77753,\n   49057,\n   9233\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe","variable_name":"trafficway_type_top6"}

```
plt.figure(figsize=(15,5))
ax =
sns.barplot(x=trafficway_type_top6['trafficway_type'],y=trafficway_typ
e_top6['count'],color='blue')
#plt.xticks(rotation=45)
```

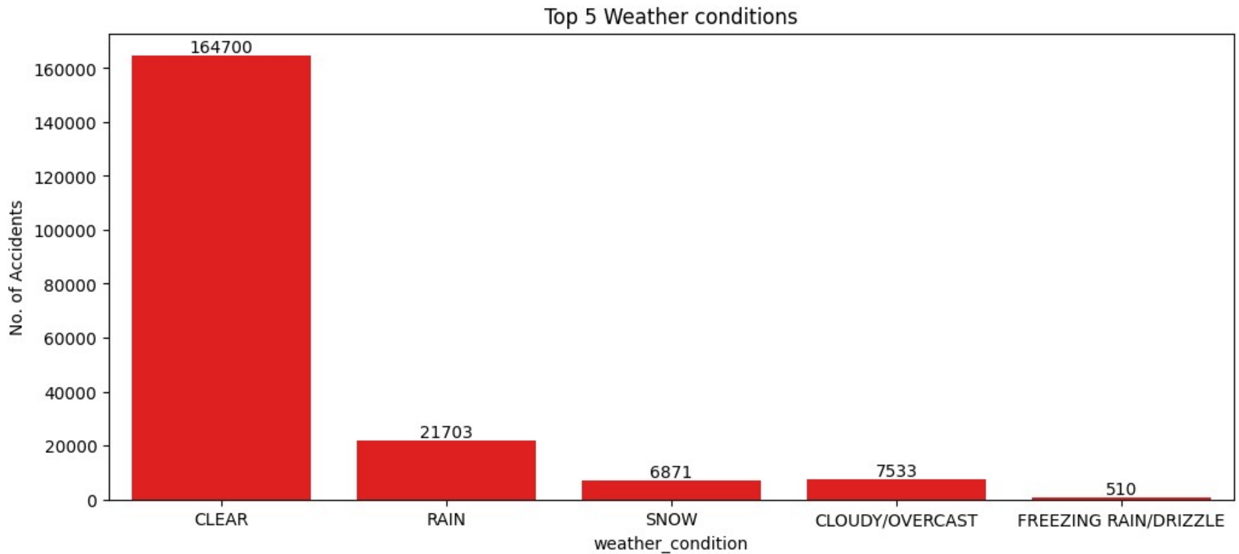
```
for i in ax.containers:
  ax.bar_label(i,)

plt.ylabel("No. of Accidents")
plt.title("Top 6 Trafficway Type")
plt.show()
```



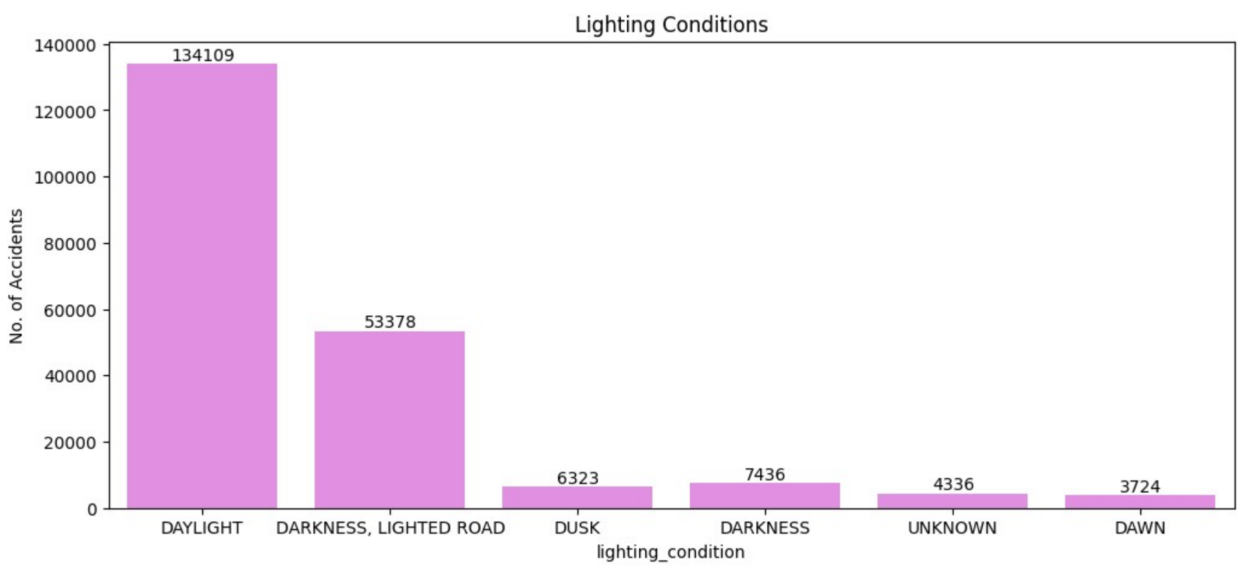

```
data = df['crash_type'].value_counts().values
keys = df['crash_type'].value_counts().index
explode = [0, 0.05]
plt.title("Crash_Type")

plt.pie(data,labels=keys,explode=explode, autopct='%.0f%%')
plt.show()
```




```
#We are taking top 5 weather condtions based on counts to make our
analysis easy
weather_condition_top5 = df[(df['weather_condition'] == 'CLEAR') |
(df['weather_condition'] == 'RAIN') | (df['weather_condition'] ==
'CLOUDY/OVERCAST') | (df['weather_condition'] == 'SNOW') |
(df['weather_condition'] == 'FREEZING RAIN/DRIZZLE')]

plt.figure(figsize=(12,5))
ax =
sns.countplot(x=weather_condition_top5['weather_condition'],color='red
')

for i in ax.containers:
  ax.bar_label(i,)

plt.ylabel("No. of Accidents")
plt.title("Top 5 Weather conditions")
plt.show()
```




```
plt.figure(figsize=(12,5))
ax = sns.countplot(x=df['lighting_condition'],color='violet')

for i in ax.containers:
  ax.bar_label(i,)

plt.ylabel("No. of Accidents")
plt.title("Lighting Conditions")
plt.show()
```




```
df.sample(5)
```

{"type":"dataframe"}

```
pd.crosstab(weather_condition_top5['weather_condition'],df['lighting_condition'],normalize=True,margins=True)
```

{"summary":"{\n \"name\": \"pd\",\n \"rows\": 6,\n \"fields\": [\n   {\n   \"column\": \"weather_condition\",\n   \"properties\": {\n   \"dtype\": \"string\",\n   \"num_unique_values\": 6,\n   \"samples\": [\n   \"CLEAR\",\n   \"CLOUDY/OVERCAST\",\n   \"All\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"DARKNESS\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.015261060417482757,\n   \"min\": 0.00021359348688883701,\n   \"max\": 0.035203187013516,\n   \"num_unique_values\": 6,\n   \"samples\": [\n   0.026674349409140808,\n   0.00045202342574149226,\n   0.035203187013516\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"DARKNESS, LIGHTED ROAD\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.11377514223280288,\n   \"min\": 0.0010480982728731305,\n   \"max\": 0.25703244137355513,\n   \"num_unique_values\": 6,\n   \"samples\": [\n   0.20291381254439517,\n   0.004197360381885285,\n   0.25703244137355513\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"DAWN\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.007716499769200128,\n   \"min\": 5.960748471316382e-05,\n   \"max\": 0.018090871610445217,\n   \"num_unique_values\": 6,\n   \"samples\": [\n   0.013352076575748694,\n   0.0008593412379481116,\n   0.018090871610445217\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"DAYLIGHT\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.2997129488101344,\n   \"min\": 0.0010878365960152396,\n   \"max\": 0.6552104392574895,\n   \"num_unique_values\": 6,\n   \"samples\": [\n   0.5491736912431637,\n   0.030191191007217472,\n   0.6552104392574895\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"DUSK\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.013219047790162 49,\n   \"min\": 0.00010431309824803668,\n   \"max\": 0.03048922843078329,\n   \"num_unique_values\": 6,\n   \"samples\": [\n   0.023241951747741126,\n   0.0014603833754725135,\n   0.03048922843078329\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"UNKNOWN\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.001642723312591244,\n   \"min\": 1.9869161571054606e-05,\n   \"max\": 0.003973832314210921,\n   \"num_unique_values\": 6,\n   \"samples\": [\n   0.002756846167983826 4,\n   0.00025829910042370985,\n   0.003973832314210921\n   ],\n

```
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"All\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.4509623381601344,\n
\"min\": 0.0025333181003094622,\n        \"max\": 1.0,\n
\"num_unique_values\": 6,\n        \"samples\": [\n
0.8181127276881733,\n          0.03741859852868858,\n          1.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe"}
```

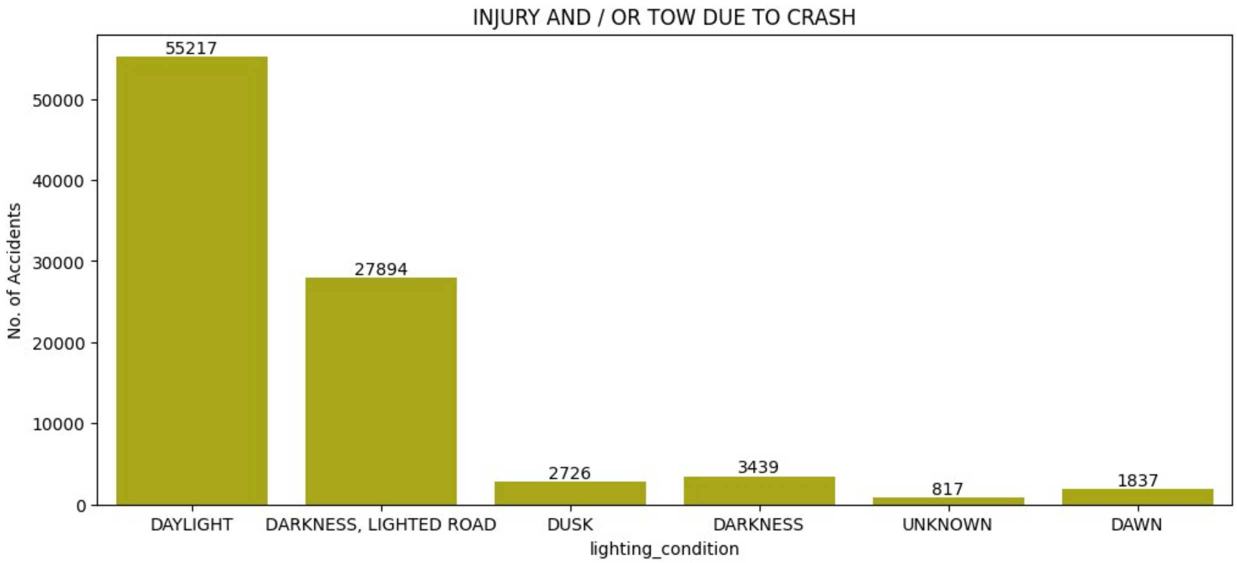

```
df['traffic_control_device'].value_counts().head(5)

traffic_control_device
TRAFFIC SIGNAL        123944
STOP SIGN/FLASHER      49139
NO CONTROLS            29508
UNKNOWN                 4455
OTHER                    670
Name: count, dtype: int64
```

Insights: From the above plots we can conclude that:

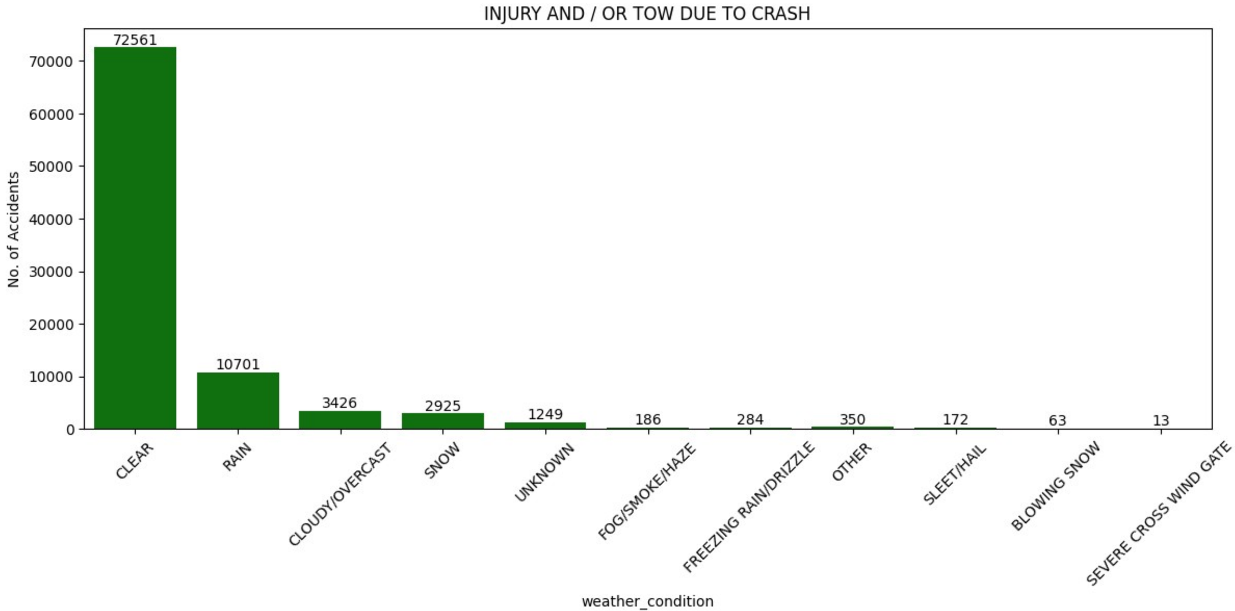
- Number of accient occured on Friday is higher compared to other days and on Sunday least number of accients occured.
- 2018 and 2019 had maximun number of accidents. 2020 had slightly less number of accidents(lockdown Covid).
- September, October, November and December found to have maximun number of crashes. (Festival session/ Winter season)
- From 3 PM to 5 PM accidents counts are at peak. 11 PM to 6 AM least number of accidents(Night time less vechiles on road).
- Non Divided trafficway found to be prime reason for most of the accidents. (Include speed limit or Divided roads).
- 56% of the accidents people had No injuries or they drove away and in 44% cases people had injuries or vechile was towed.
- Weather had not much effect as majority of the accidents happended on Clear Weather conditions. Rain and snow weather conditions found to have some effect on accidents.
- Majority of accidents happended at Daylight lightning conditions whereas Darkness could be also considered as a reason.

```
df['crash_type'].value_counts()

crash_type
NO INJURY / DRIVE AWAY                117376
INJURY AND / OR TOW DUE TO CRASH       91930
Name: count, dtype: int64

plt.figure(figsize=(12,5))
ax = sns.countplot(x=df[df['crash_type'] == "INJURY AND / OR TOW DUE
TO CRASH"]['lighting_condition'],color='y')
```

```
for i in ax.containers:
  ax.bar_label(i,)

plt.ylabel("No. of Accidents")
plt.title("INJURY AND / OR TOW DUE TO CRASH")
plt.show()
```




```
plt.figure(figsize=(14,5))
ax = sns.countplot(x=df[df['crash_type'] == "INJURY AND / OR TOW DUE
TO CRASH"]['weather_condition'],color='g')

for i in ax.containers:
  ax.bar_label(i,)

plt.ylabel("No. of Accidents")
plt.xticks(rotation=45)
plt.title("INJURY AND / OR TOW DUE TO CRASH")
plt.show()
```




```
pd.crosstab(df['crash_type'],df['lighting_condition'],normalize=True,margins=True) #Checking the relationship between 'crash_type' and 'lighting_condtion'
```

{"summary":"{\n \"name\": \"pd\",\n \"rows\": 3,\n \"fields\": [\n   {\n   \"column\": \"crash_type\",\n   \"properties\": {\n   \"dtype\": \"string\",\n   \"num_unique_values\": 3,\n   \"samples\": [\n   \"INJURY AND / OR TOW DUE TO CRASH\",\n   \"NO INJURY / DRIVE AWAY\",\n   \"All\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"DARKNESS\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.010342004990892963,\n   \"min\": 0.016430489331409514,\n   \"max\": 0.03552693186052956,\n   \"num_unique_values\": 3,\n   \"samples\": [\n   0.016430489331409514,\n   0.019096442529120045,\n   0.03552693186052956\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"DARKNESS, LIGHTED ROAD\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.07384377874505169,\n   \"min\": 0.12175475141658625,\n   \"max\": 0.25502374513869647,\n   \"num_unique_values\": 3,\n   \"samples\": [\n   0.133268993722110\2,\n   0.12175475141658625,\n   0.25502374513869647\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"DAWN\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.005137534769806949,\n   \"min\": 0.008776623699272835,\n   \"max\": 0.017792132093681023,\n   \"num_unique_values\": 3,\n   \"samples\": [\n   0.008776623699272835,\n   0.009015508394408187,\n   0.017792132093681023\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\

```
n    },\n    {\n      \"column\": \"DAYLIGHT\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
0.19341666711324482,\n        \"min\": 0.2638099242257747,\n
\"max\": 0.6407317515981387,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          0.2638099242257747,\n
0.3769218273723639,\n          0.6407317515981387\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"DUSK\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.008965472562968648,\n
\"min\": 0.013023993578779395,\n        \"max\":
0.030209358546816622,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          0.013023993578779395,\n
0.017185364968037 23,\n          0.030209358546816622\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"UNKNOWN\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
0.008799187093295423,\n        \"min\": 0.0039033759185116527,\n
\"max\": 0.020716080762137733,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          0.0039033759185116527,\n
0.01681270484362608,\n          0.020716080762137733\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"All\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.29500566776765785,\n
\"min\": 0.43921340047585833,\n        \"max\": 1.0,\n
\"num_unique_values\": 3,\n        \"samples\": [\n
0.43921340047585833,\n          0.5607865995241417,\n          1.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe"}
```

Chi-square test to check if crash_type is dependent on the lighting_condition

Chi-square test is a statistical hypothesis test used to analyze contingency tables and determine the relationship between two categorical variables, or to assess if observed frequencies differ significantly from expected frequencies.

Basically this test determines if two categorical variables are related or independent of each other.

Assumptions: The data must be categorical, observations must be independent, and the expected frequencies in each category should be at least 5 for valid results

Let us assume that:

Null Hypothesis(Ho) -> There is no relationship or association between the categorical variables

Alternate Hypothesis(Ha) -> There is relationship or association between the categorical variables

alpha = 0.05(95% confidence level)

```
contingency_table =
pd.crosstab(df['crash_type'],df['lighting_condition']) #Taking
crosstab between lighting_condition and crash_type column to establish
the connection
contingency_table
```

{"summary":"{\n \"name\": \"contingency_table\",\n \"rows\": 2,\n \"fields\": [\n   {\n   \"column\": \"crash_type\",\n   \"properties\": {\n   \"dtype\": \"string\",\n   \"num_unique_values\": 2,\n   \"samples\": [\n   \"NO INJURY / DRIVE AWAY\",\n   \"INJURY AND / OR TOW DUE TO CRASH\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"DARKNESS\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 394,\n   \"min\": 3439,\n   \"max\": 3997,\n   \"num_unique_values\": 2,\n   \"samples\": [\n   3997,\n   3439\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"DARKNESS, LIGHTED ROAD\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 1704,\n   \"min\": 25484,\n   \"max\": 27894,\n   \"num_unique_values\": 2,\n   \"samples\": [\n   25484,\n   27894\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"DAWN\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 35,\n   \"min\": 1837,\n   \"max\": 1887,\n   \"num_unique_values\": 2,\n   \"samples\": [\n   1887,\n   1837\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"DAYLIGHT\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 16740,\n   \"min\": 55217,\n   \"max\": 78892,\n   \"num_unique_values\": 2,\n   \"samples\": [\n   78892,\n   55217\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"DUSK\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 615,\n   \"min\": 2726,\n   \"max\": 3597,\n   \"num_unique_values\": 2,\n   \"samples\": [\n   3597,\n   2726\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"UNKNOWN\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 1910,\n   \"min\": 817,\n   \"max\": 3519,\n   \"num_unique_values\": 2,\n   \"samples\": [\n   3519,\n   817\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe","variable_name":"contingency_table"}

```
from scipy.stats import chi2_contingency #importing required library
for chi-square test
```

```
chi2_contingency(contingency_table) #performing chi-square contingency test
```

```
Chi2ContingencyResult(statistic=np.float64(3086.6477557667913),
pvalue=np.float64(0.0), dof=5, expected_freq=array([[ 3265.99084594,
23444.3328906 ,  1635.63070337, 58902.46992442,
          2777.14633121,  1904.42930446],
        [ 4170.00915406, 29933.6671094 ,  2088.36929663,
75206.53007558,
          3545.85366879,  2431.57069554]]))
```

From the above chi-square contigency test between two categorial column "crash_type" and "lighting_condition" we can see that p-value(0.0) < alpha(0.05). So we will reject our null hypothesis.

We can conclude that there is relationship or association between the two categorical variables "crash_type" and "lighting_condition".

Final Insights:

1. Accident Trends:
- The highest number of accidents occur on Fridays, with the lowest on Sundays.
- Accident counts peaked in 2018 and 2019, with a slight decrease in 2020.
- September, October, November, and December have the most accidents, possibly due to festivals and winter weather.
- Accidents are most frequent between 3 PM and 5 PM and least frequent between 11 PM and 6 AM.
1. Accident Factors:
- Non-divided roadways are associated with the most accidents, highlighting a need for speed limits or road division.
- In 56% of cases, there were no injuries or the drivers drove away, while 44% involved injuries or towing.
- Most accidents occur in clear weather, but rain and snow also play a role.
- Daylight is the most common lighting condition during accidents, but darkness is also a contributing factor.
1. Injury and Tow Accidents:
- Accidents resulting in injuries or towing are more likely to happen in daylight.
- These types of accidents occur most often in clear weather, followed by rain and snow.

Final Business Insights:

1. Traffic Management:
- Focus on Fridays and peak hours (3 PM-5 PM) for traffic management.
- Improve safety measures on non-divided roadways.
- Implement strategies to reduce accidents during festival seasons and winter months.
1. Road Safety:
- Raise awareness about the risks associated with driving in darkness and adverse weather conditions.

- Promote safe driving practices during peak accident times.

1. Infrastructure Improvements:

- Consider dividing non-divided roadways or implementing speed limits to reduce accidents.
- Improve lighting and visibility on roads to enhance safety.

1. Emergency Response:

- Allocate resources strategically based on accident trends and high-risk areas.
- Ensure prompt emergency response during peak accident times.

These insights can be used by government agencies, transportation departments, and other stakeholders to develop strategies for improving road safety and reducing traffic accidents.