task5

April 30, 2024

1 Import the required Libraries

[1]:

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

2 Load the Dataset

[3]:

```
# Use raw string literal or double backslashes for file path
df = pd.read_csv(r"C:\Users\aakas\OneDrive\Desktop\Rprog\RTA Dataset.csv")

# Check the first few rows of the DataFrame
print(df.head())
```

```
       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]
```

3 Data Pre-processing

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

[4]:
```
       Number_of_vehicles_involved  Number_of_casualties
count                 12316.000000          12316.000000
mean                      2.040679              1.548149
std                       0.688790              1.007179
min                       1.000000              1.000000
25%                       2.000000              1.000000
50%                       2.000000              1.000000
75%                       2.000000              2.000000
max                       7.000000              8.000000
```

[5]:
```
df.shape
```

[5]:
```
(12316, 32)
```

[6]:
```
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_casuality            9118 non-null   object
 28  Fitness_of_casuality         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
```

[7]:

```
#finding duplicate values
df.duplicated().sum()
```

[7]:

```
0
```

4 Handling the missing values

[8]:

```
#checking missing values
df.isna().sum()
```

[8]:

```
Time                              0
Day_of_week                       0
Age_band_of_driver                0
```

```
Sex_of_driver                    0
Educational_level              741
Vehicle_driver_relation        579
Driving_experience             829
Type_of_vehicle                950
Owner_of_vehicle               482
Service_year_of_vehicle       3928
Defect_of_vehicle             4427
Area_accident_occured          239
Lanes_or_Medians               385
Road_allignment                142
Types_of_Junction              887
Road_surface_type              172
Road_surface_conditions          0
Light_conditions                 0
Weather_conditions               0
Type_of_collision              155
Number_of_vehicles_involved      0
Number_of_casualties             0
Vehicle_movement               308
Casualty_class                   0
Sex_of_casualty                  0
Age_band_of_casualty             0
Casualty_severity                0
Work_of_casuality             3198
Fitness_of_casuality          2635
Pedestrian_movement              0
Cause_of_accident                0
Accident_severity                0
dtype: int64
```

[9]:
```
#dropping columns which has more than 2500 missing values and Time column
df.drop(['Service_year_of_vehicle','Defect_of_vehicle','Work_of_casuality',␣
 ↪'Fitness_of_casuality','Time'],
        axis = 1, inplace = True)
df.head()
```

[9]:
```
  Day_of_week Age_band_of_driver Sex_of_driver   Educational_level  \
0      Monday              18-30          Male   Above high school
1      Monday              31-50          Male  Junior high school
2      Monday              18-30          Male  Junior high school
3      Sunday              18-30          Male  Junior high school
4      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 Area_accident_occured   Lanes_or_Medians  …  \
0            Owner     Residential areas                NaN  …
1            Owner          Office areas  Undivided Two way  …
2            Owner    Recreational areas              other  …
3     Governmental          Office areas              other  …
4            Owner      Industrial areas              other  …

  Number_of_vehicles_involved Number_of_casualties Vehicle_movement  \
0                           2                    2   Going straight
1                           2                    2   Going straight
2                           2                    2   Going straight
3                           2                    2   Going straight
4                           2                    2   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

   Pedestrian_movement           Cause_of_accident Accident_severity
0     Not a Pedestrian             Moving Backward     Slight Injury
1     Not a Pedestrian                  Overtaking     Slight Injury
2     Not a Pedestrian   Changing lane to the left    Serious Injury
3     Not a Pedestrian  Changing lane to the right     Slight Injury
4     Not a Pedestrian                  Overtaking     Slight Injury

[5 rows x 27 columns]
```

[10]:

```
#storing categorical column names to a new variable
categorical=[i for i in df.columns if df[i].dtype=='O']
print('Categorical variables:',categorical)
```

```
Categorical variables: ['Day_of_week', 'Age_band_of_driver', 'Sex_of_driver',
'Educational_level', 'Vehicle_driver_relation', 'Driving_experience',
'Type_of_vehicle', 'Owner_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', 'Vehicle_movement', 'Casualty_class', 'Sex_of_casualty',
'Age_band_of_casualty', 'Casualty_severity', 'Pedestrian_movement',
'Cause_of_accident', 'Accident_severity']
```

[11]:

```
#for categorical values we can replace the null values with the Mode of it
for i in categorical:
    df[i].fillna(df[i].mode()[0],inplace=True)
```

[12]:

```
#checking the current null values
df.isna().sum()
```

[12]:

```
Day_of_week                   0
Age_band_of_driver            0
Sex_of_driver                 0
Educational_level             0
Vehicle_driver_relation       0
Driving_experience            0
Type_of_vehicle               0
Owner_of_vehicle              0
Area_accident_occured         0
Lanes_or_Medians              0
Road_allignment               0
Types_of_Junction             0
Road_surface_type             0
Road_surface_conditions       0
Light_conditions              0
Weather_conditions            0
Type_of_collision             0
Number_of_vehicles_involved   0
Number_of_casualties          0
Vehicle_movement              0
Casualty_class                0
Sex_of_casualty               0
Age_band_of_casualty          0
Casualty_severity             0
Pedestrian_movement           0
Cause_of_accident             0
Accident_severity             0
dtype: int64
```

5 Exploratory Data Analysis

[13]:

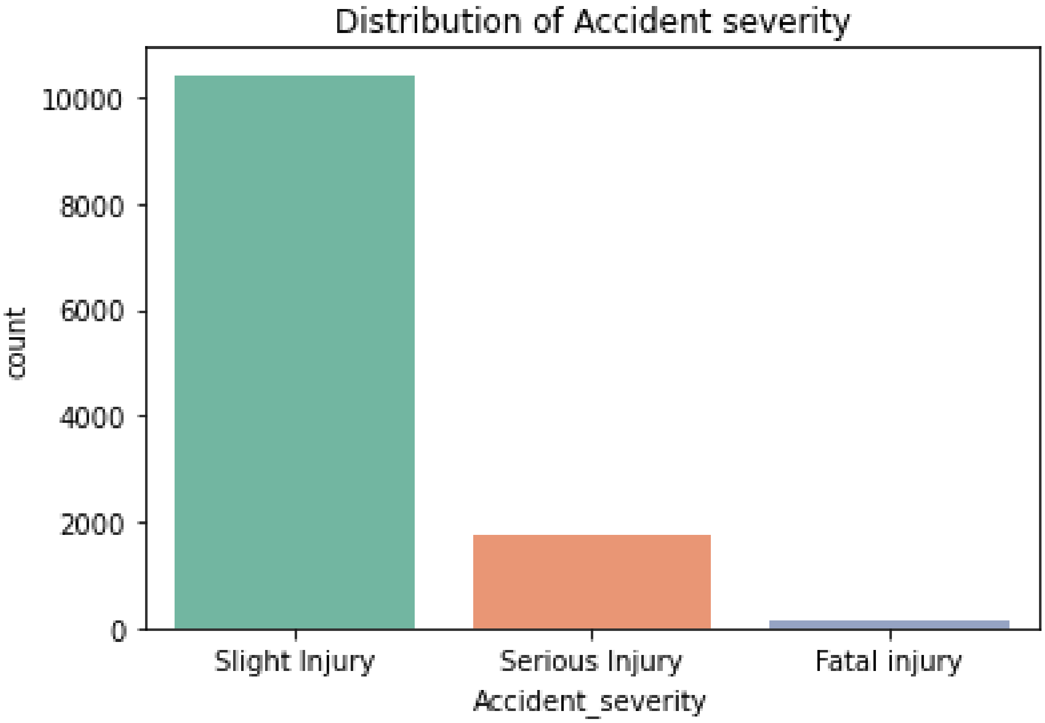
```
#Distribution of Accident severity
df['Accident_severity'].value_counts()
```

[13]:

```
Slight Injury     10415
Serious Injury     1743
Fatal injury        158
Name: Accident_severity, dtype: int64
```

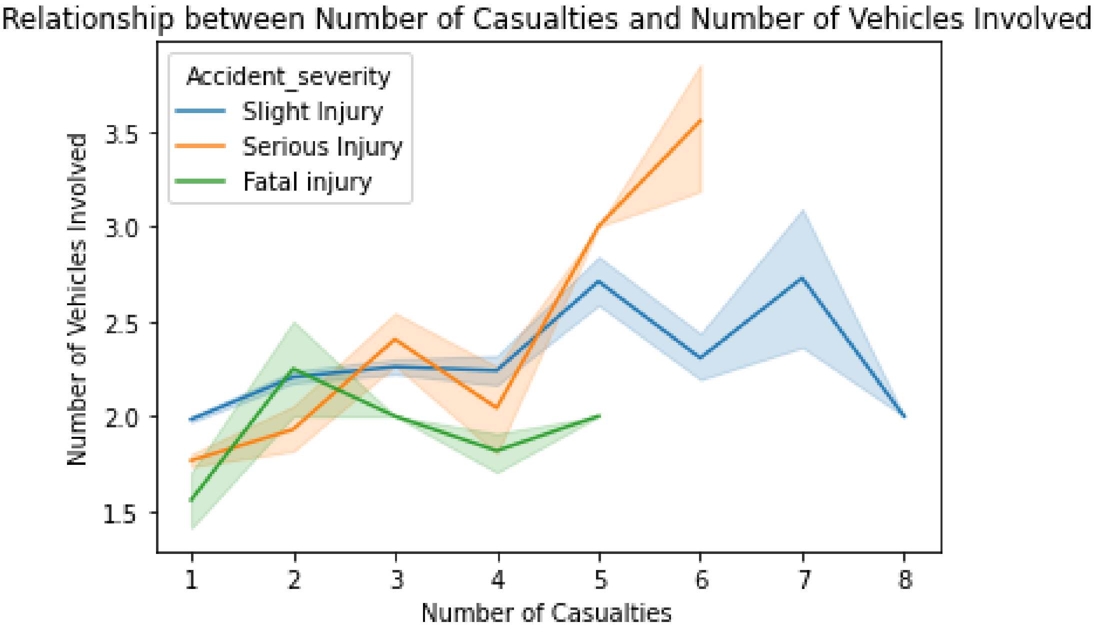
6 Bar Plot for the Disribution of Accident Severity

[14]:
```python
sns.countplot(x=df['Accident_severity'], palette='Set2')
plt.title('Distribution of Accident severity')
plt.show()
```



7 Line Chart for Different Casualities VS No.of Vehicles

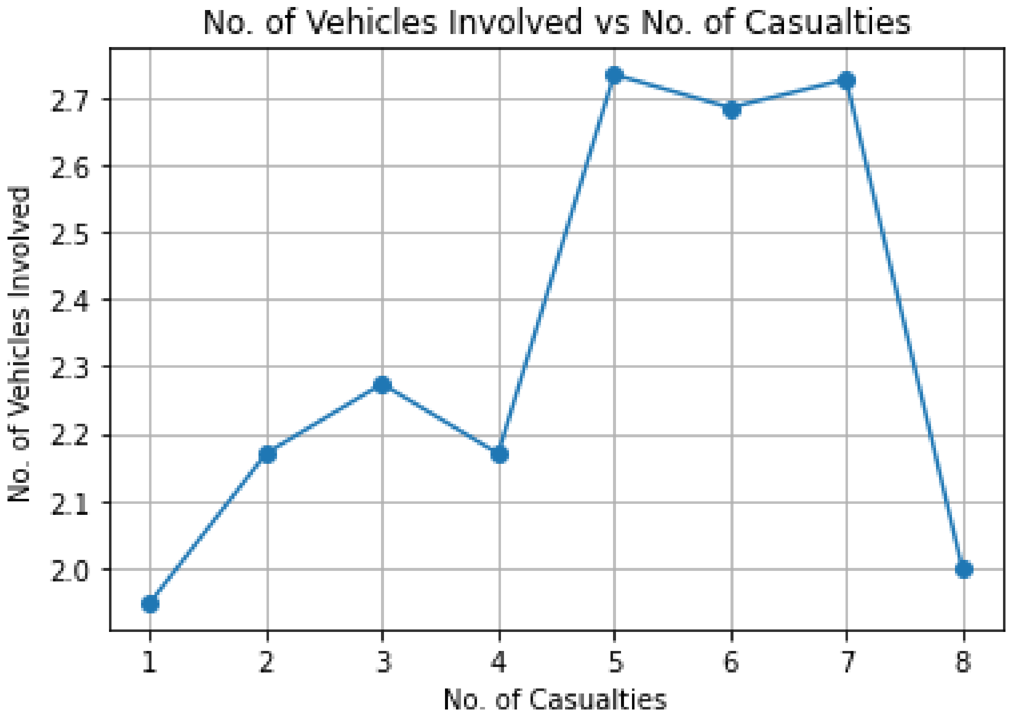
[15]:
```python
sns.lineplot(x=df['Number_of_casualties'], y=df['Number_of_vehicles_involved'],␣
  ↪hue=df['Accident_severity'])
plt.title('Relationship between Number of Casualties and Number of Vehicles␣
  ↪Involved')
plt.xlabel('Number of Casualties')
plt.ylabel('Number of Vehicles Involved')
plt.show()
```



8 Line Chart for No.of Casualities VS No.of Vehicles

[16]:
```
# Calculate mean values
mean_values = df.groupby('Number_of_casualties')['Number_of_vehicles_involved'].
  ↪mean().reset_index()

# Plotting line chart
plt.plot(mean_values['Number_of_casualties'],␣
  ↪mean_values['Number_of_vehicles_involved'], marker='o', linestyle='-')
plt.title('No. of Vehicles Involved vs No. of Casualties')
plt.xlabel('No. of Casualties')
plt.ylabel('No. of Vehicles Involved')
plt.grid(True)
plt.show()
```



9 Correlation between numerical columns

[17]:

```
# Drop non-numeric columns
numeric_df = df.select_dtypes(include=['number'])

# Calculate correlation
correlation_matrix = numeric_df.corr()

# Display correlation matrix
print(correlation_matrix)
```

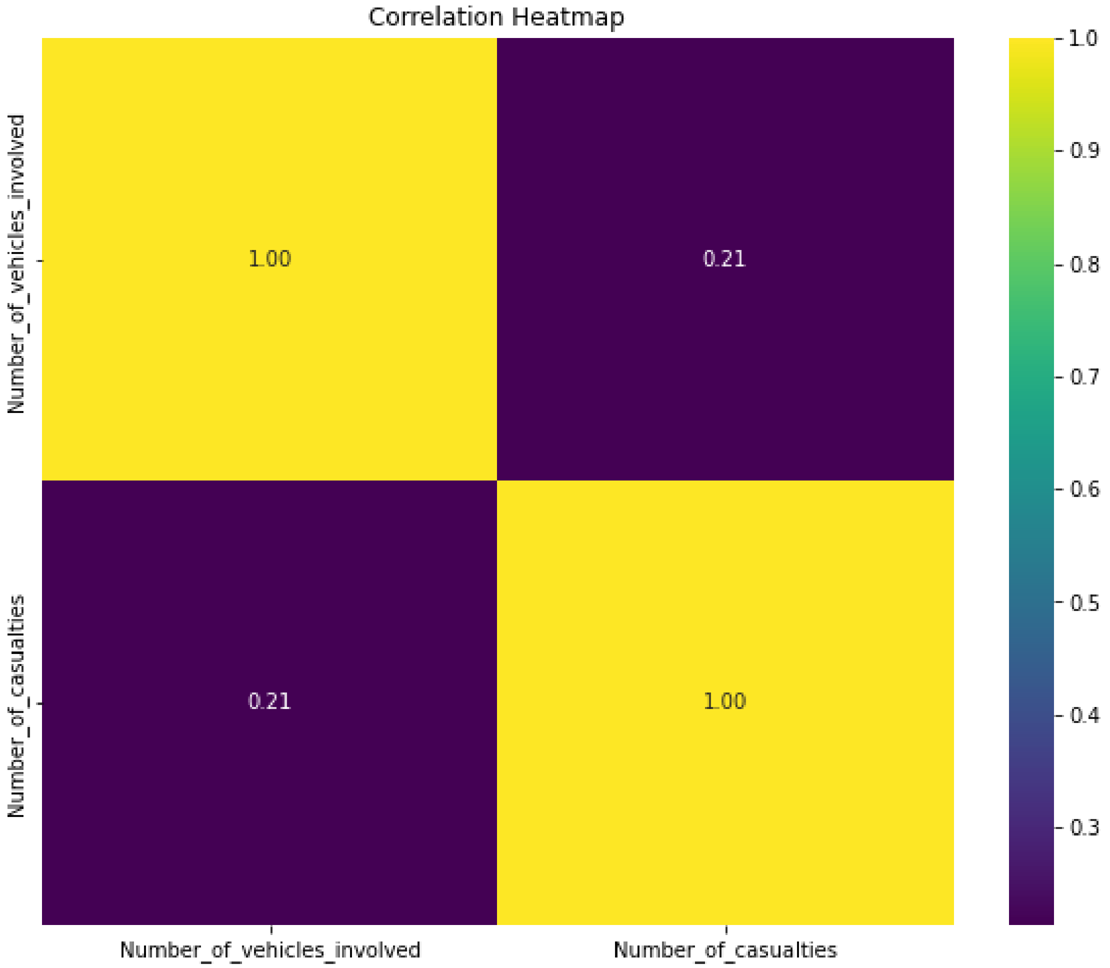
```
                             Number_of_vehicles_involved  Number_of_casualties
Number_of_vehicles_involved                     1.000000              0.213427
Number_of_casualties                            0.213427              1.000000
```

[18]:

```
#plotting the correlation using heatmap
# Select only numeric columns
numeric_df = df.select_dtypes(include=['number'])

# Calculate correlation matrix
correlation_matrix = numeric_df.corr()
```

```
# Create heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='viridis', fmt=".2f")
plt.title('Correlation Heatmap')
plt.show()
```



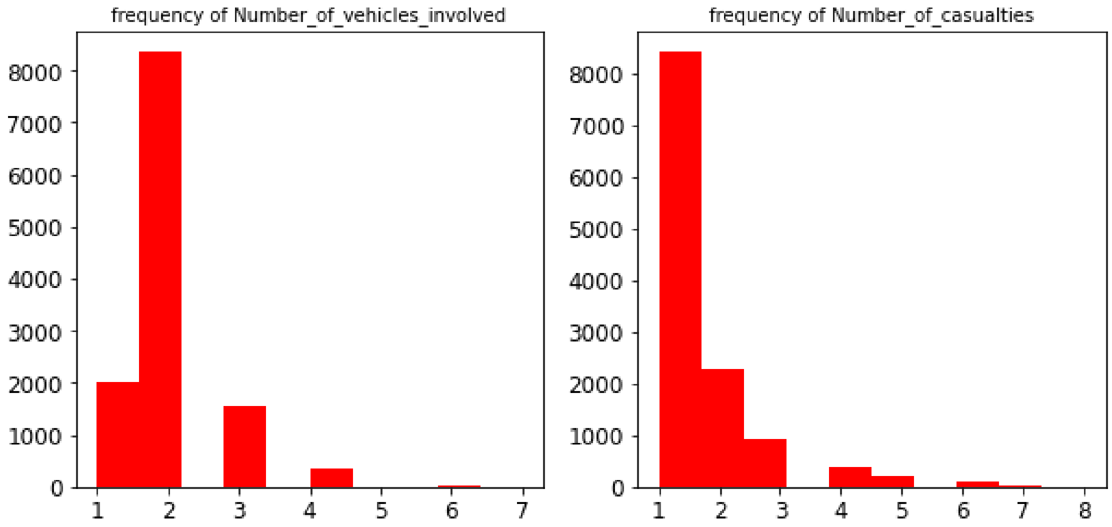
10 Visualisation of Frequencies of Numerical Columns

[19]:
```
#storing numerical column names to a variable
numerical=[i for i in df.columns if df[i].dtype!='O']
print('The numerica variables are',numerical)
```

```
The numerica variables are ['Number_of_vehicles_involved',
'Number_of_casualties']
```

[21]:

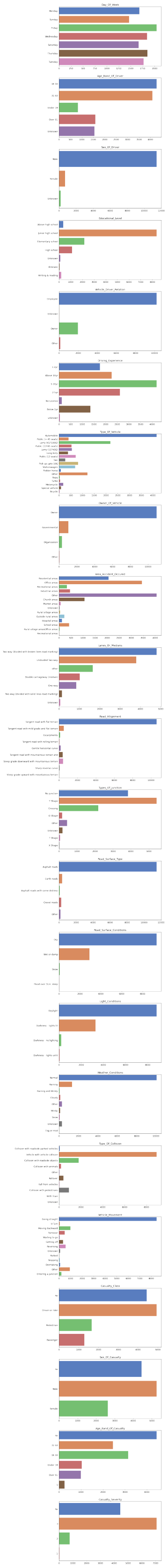
```python
#distribution for numerical columns
plt.figure(figsize=(10,10))
plotnumber = 1
for i in numerical:
    if plotnumber <= df.shape[1]:
        ax1 = plt.subplot(2,2,plotnumber)
        plt.hist(df[i],color='red')
        plt.xticks(fontsize=12)
        plt.yticks(fontsize=12)
        plt.title('frequency of '+i, fontsize=10)
    plotnumber +=1
```

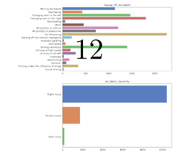


[22]:

```python
#count plot for categorical values
plt.figure(figsize=(10,200))
plotnumber = 1

for col in categorical:
    if plotnumber <= df.shape[1] and col!='Pedestrian_movement':
        ax1 = plt.subplot(28,1,plotnumber)
        sns.countplot(data=df, y=col, palette='muted')
        plt.xticks(fontsize=12)
        plt.yticks(fontsize=12)
        plt.title(col.title(), fontsize=14)
        plt.xlabel('')
        plt.ylabel('')
    plotnumber +=1
```


Day_Of_Week
Monday
Sunday
Friday
Wednesday
Saturday
Thursday
Tuesday
0
250
500
750
1000
1250
1500
1750
2000
Age_Band_Of_Driver
18-30
31-50
Under 18
Over 51
Unknown
0
500
1000
1500
2000
2500
3000
3500
4000
Sex_Of_Driver
Male
Female
Unknown
0
2000
4000
6000
8000
10000
12000
Educational_Level
Above high school
Junior high school
Elementary school
High school
Unknown
Illiterate
Writing & reading
0
1000
2000
3000
4000
5000
6000
7000
8000
Vehicle_Driver_Relation
Employee
Unknown
Owner
Other
0
2000
4000
6000
8000
10000
Driving_Experience
1-2yr
Above 10yr
5-10yr
2-5yr
No Licence
Below 1yr
unknown
0
500
1000
1500
2000
2500
3000
3500
4000
Type_Of_Vehicle
Automobile
Public (> 45 seats)
Lorry (41?100Q)
Public (13?45 seats)
Lorry (11?40Q)
Long lorry
Public (12 seats)
Taxi
Pick up upto 10Q
Stationwagen
Ridden horse
Other
Bajaj
Turbo
Motorcycle
Special vehicle
Bicycle
0
500
1000
1500
2000
2500
3000
3500
4000
Owner_Of_Vehicle
Owner
Governmental
Organization
Other
0
2000
4000
6000
8000
10000
Area_Accident_Occured
Residential areas
Office areas
Recreational areas
Industrial areas
Other
Church areas
Market areas
Unknown
Rural village areas
Outside rural areas
Hospital areas
School areas
Rural village areasOffice areas
Recreational areas
0
500
1000
1500
2000
2500
3000
3500
4000
Lanes_Or_Medians
Two-way (divided with broken lines road marking)
Undivided Two way
other
Double carriageway (median)
One way
Two-way (divided with solid lines road marking)
Unknown
0
1000
2000
3000
4000
5000
Road_Allignment
Tangent road with flat terrain
Tangent road with mild grade and flat terrain
Escarpments
Tangent road with rolling terrain
Gentle horizontal curve
Tangent road with mountainous terrain and
Steep grade downward with mountainous terrain
Sharp reverse curve
Steep grade upward with mountainous terrain
0
2000
4000
6000
8000
10000
Types_Of_Junction
No junction
Y Shape
Crossing
O Shape
Other
Unknown
T Shape
X Shape
0
1000
2000
3000
4000
5000
Road_Surface_Type
Asphalt roads
Earth roads
Asphalt roads with some distress
Gravel roads
Other
0
2000
4000
6000
8000
10000
12000
Road_Surface_Conditions
Dry
Wet or damp
Snow
Flood over 3cm. deep
0
2000
4000
6000
8000
Light_Conditions
Daylight
Darkness - lights lit
Darkness - no lighting
Darkness - lights unlit
0
2000
4000
6000
8000
Weather_Conditions
Normal
Raining
Raining and Windy
Cloudy
Other
Windy
Snow
Unknown
Fog or mist
0
2000
4000
6000
8000
10000
Type_Of_Collision
Collision with roadside-parked vehicles
Vehicle with vehicle collision
Collision with roadside objects
Collision with animals
Other
Rollover
Fall from vehicles
Collision with pedestrians
With Train
Unknown
0
2000
4000
6000
8000
Vehicle_Movement
Going straight
U-Turn
Moving Backward
Turnover
Waiting to go
Getting off
Reversing
Unknown
Parked
Stopping
Overtaking
Other
Entering a junction
0
1000
2000
3000
4000
5000
6000
7000
8000
Casualty_Class
na
Driver or rider
Pedestrian
Passenger
0
1000
2000
3000
4000
5000
Sex_Of_Casualty
na
Male
Female
0
1000
2000
3000
4000
5000
Age_Band_Of_Casualty
na
31-50
18-30
Under 18
Over 51
5
0
1000
2000
3000
4000
Casualty_Severity
na
3
2
1
0
1000
2000
3000
4000
5000
6000
7000


Cause_Of_Accident
Moving Backward
Overtaking
Changing lane to the left
Changing lane to the right
Overloading
Other
No priority to vehicle
No priority to pedestrian
No distancing
Getting off the vehicle improperly
Improper parking
Overspeed
Driving carelessly
Driving at high speed
Driving to the left
Unknown
Overturning
Turnover
Driving under the influence of drugs
Drunk driving
0
500
1000
1500
2000
Accident_Severity
Slight Injury
Serious Injury
Fatal injury
0
2000
4000
6000
8000
10000

```
[25]: # THANK YOU
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```