

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

```
In [2]: df=pd.read_csv("RTA Dataset.csv")
df.head()
```

```
Out[2]:
```

	Time	Day_of_week	Age_band_of_driver	Sex_of_driver	Educational_level	Vehicle_c
--	------	-------------	--------------------	---------------	-------------------	-----------

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	
---	---------	--------	-------	------	--------------------	--

5 rows × 32 columns



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

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

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

```
Out[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
-----	----------	----------

```
In [5]: df.describe(include="all")
```

Out[5]:

	Time	Day_of_week	Age_band_of_driver	Sex_of_driver	Educational_level	Vel
count	12316	12316	12316	12316	11575	
unique	1074	7	5	3	7	
top	15:30:00	Friday	18-30	Male	Junior high school	
freq	120	2041	4271	11437	7619	
mean	NaN	NaN	NaN	NaN	NaN	
std	NaN	NaN	NaN	NaN	NaN	
min	NaN	NaN	NaN	NaN	NaN	
25%	NaN	NaN	NaN	NaN	NaN	
50%	NaN	NaN	NaN	NaN	NaN	
75%	NaN	NaN	NaN	NaN	NaN	
max	NaN	NaN	NaN	NaN	NaN	

11 rows × 32 columns



In [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_alignment                       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
```

```
In [7]: df.duplicated().sum()
```

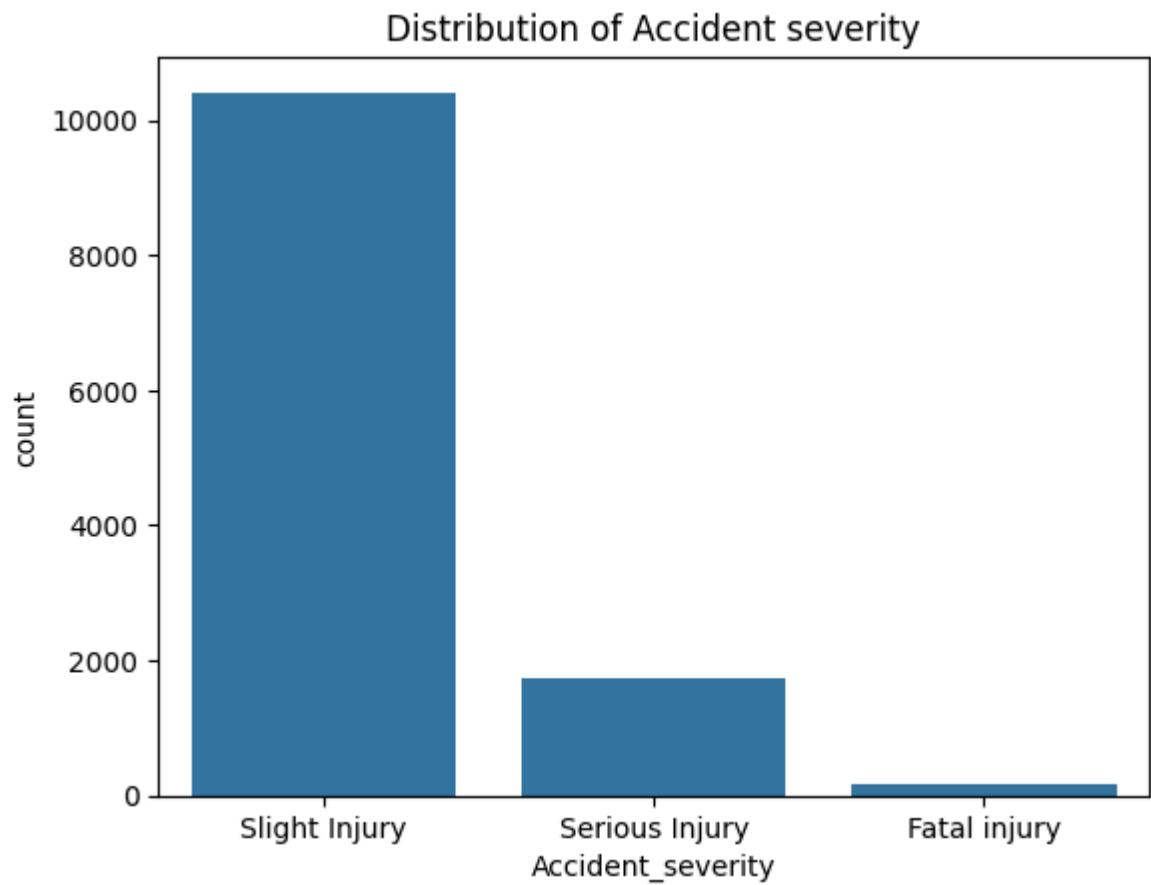
```
Out[7]: np.int64(0)
```

```
In [8]: df['Accident_severity'].value_counts()
```

```
Out[8]: Accident_severity
Slight Injury    10415
Serious Injury   1743
Fatal injury      158
Name: count, dtype: int64
```

```
In [9]: sns.countplot(x = df['Accident_severity'])
plt.title('Distribution of Accident severity')
```

```
Out[9]: Text(0.5, 1.0, 'Distribution of Accident severity')
```



```
In [10]: df.isna().sum()
```

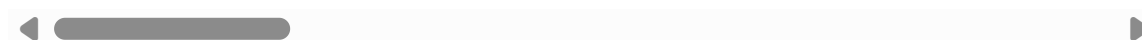
```
Out[10]: Time 0
Day_of_week 0
Age_band_of_driver 0
Sex_of_driver 0
Educational_level 741
Vehicle_driver_relation 579
Driving_experience 829
Type_of_vehicle 950
Owner_of_vehicle 482
Service_year_of_vehicle 3928
Defect_of_vehicle 4427
Area_accident_occured 239
Lanes_or_Medians 385
Road_allignment 142
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_casualty 3198
Fitness_of_casualty 2635
Pedestrian_movement 0
Cause_of_accident 0
Accident_severity 0
dtype: int64
```

```
In [11]: df.drop(['Service_year_of_vehicle', 'Defect_of_vehicle', 'Work_of_casualty', 'Fit
            axis = 1, inplace = True)
df.head()
```

```
Out[11]:
```

	Day_of_week	Age_band_of_driver	Sex_of_driver	Educational_level	Vehicle_driver_rela
0	Monday	18-30	Male	Above high school	Emple
1	Monday	31-50	Male	Junior high school	Emple
2	Monday	18-30	Male	Junior high school	Emple
3	Sunday	18-30	Male	Junior high school	Emple
4	Sunday	18-30	Male	Junior high school	Emple

5 rows × 27 columns



```
In [12]: categorical=[i for i in df.columns if df[i].dtype=='O']
print('The categorical variables are', categorical)
```

The categorical variables are ['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_alignment', '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']

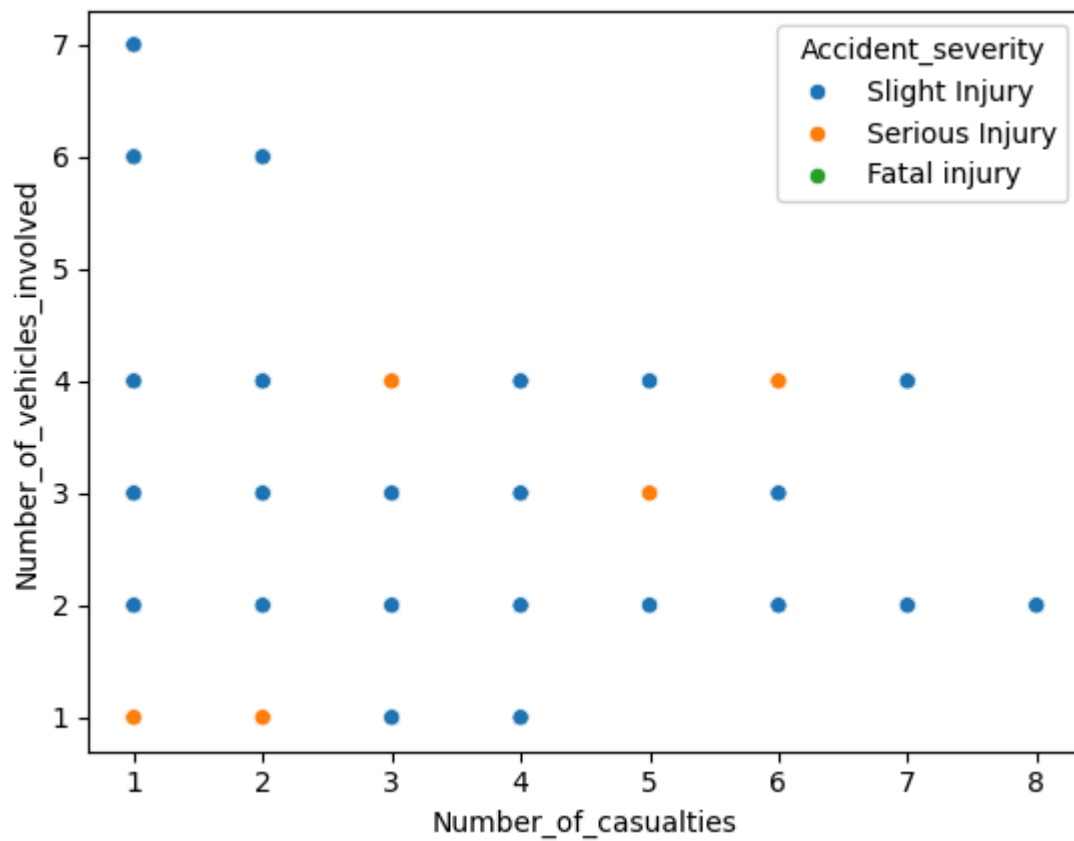
```
In [13]: for i in categorical:
          df[i].fillna(df[i].mode()[0],inplace=True)
```

```
In [14]: df.isna().sum()
```

```
Out[14]: 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_alignment              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
```

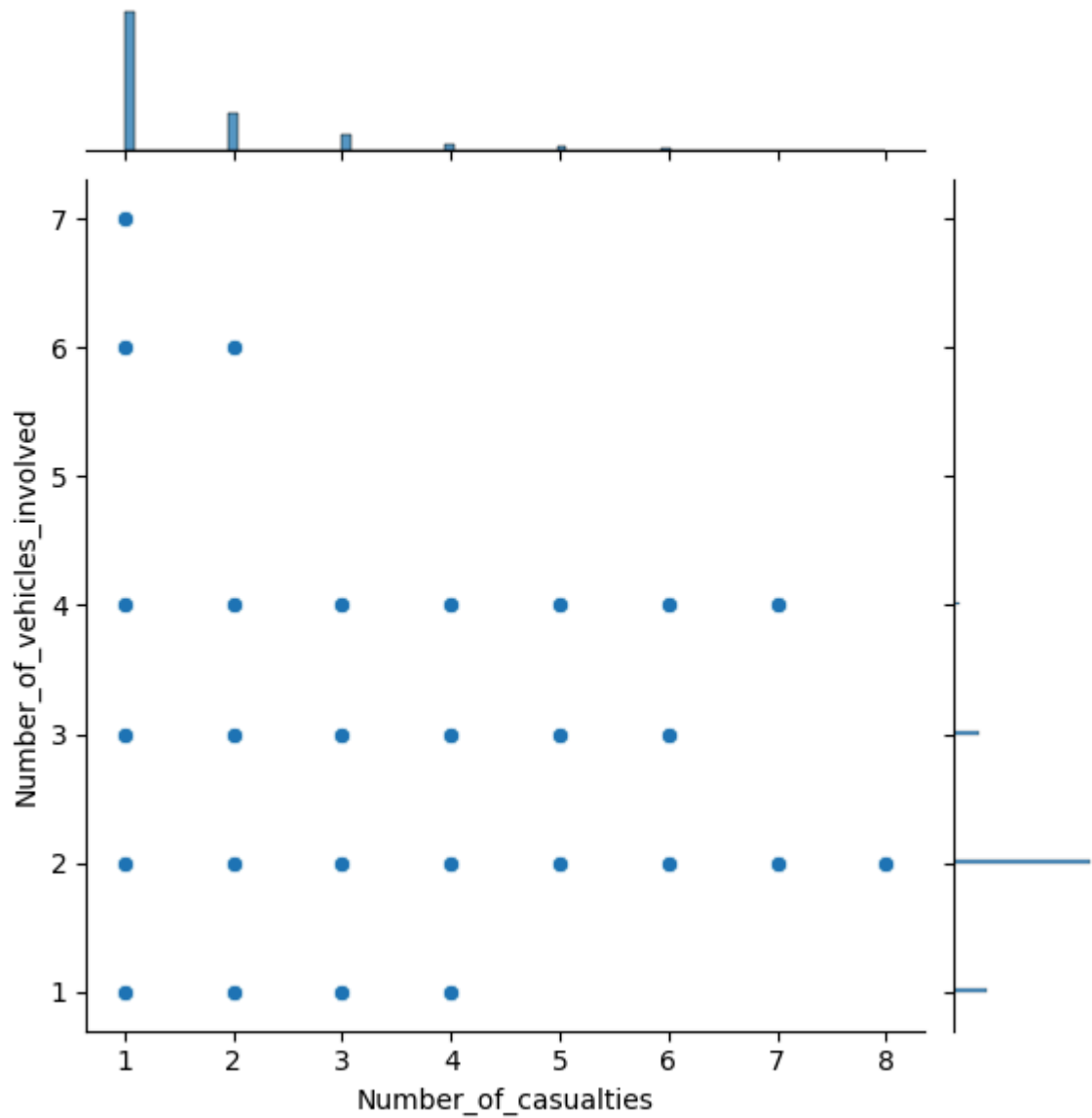
```
In [15]: sns.scatterplot(x=df['Number_of_casualties'], y=df['Number_of_vehicles_involved'])
```

```
Out[15]: <Axes: xlabel='Number_of_casualties', ylabel='Number_of_vehicles_involved'>
```



```
In [17]: sns.jointplot(x='Number_of_casualties',y='Number_of_vehicles_involved',data=df)
```

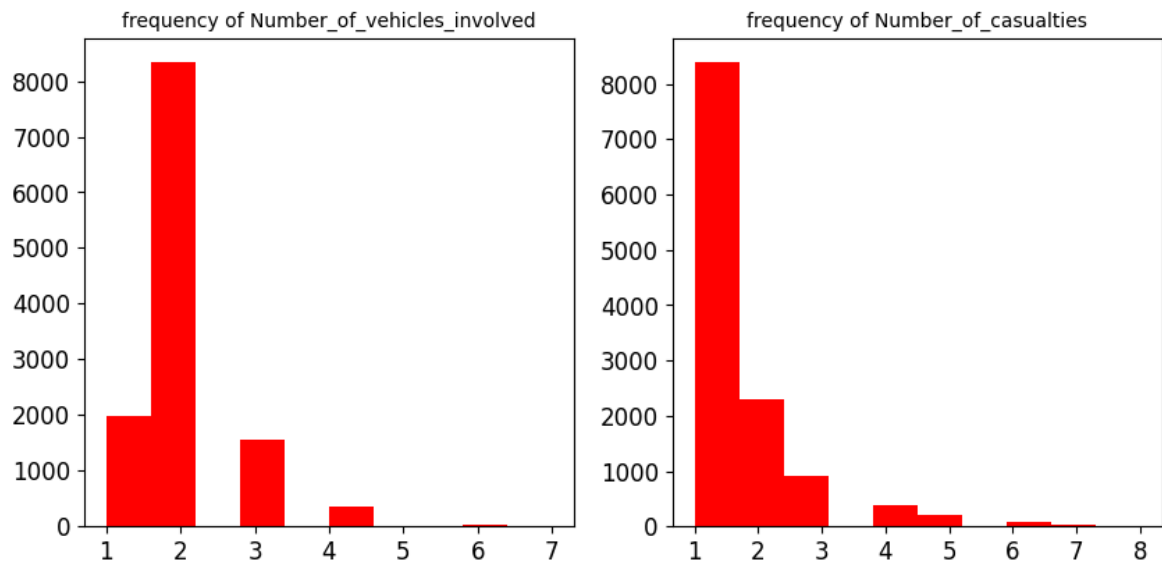
```
Out[17]: <seaborn.axisgrid.JointGrid at 0x1a8c9649d30>
```



```
In [18]: numerical=[i for i in df.columns if df[i].dtype!='0']
          print('The numerica variables are',numerical)
```

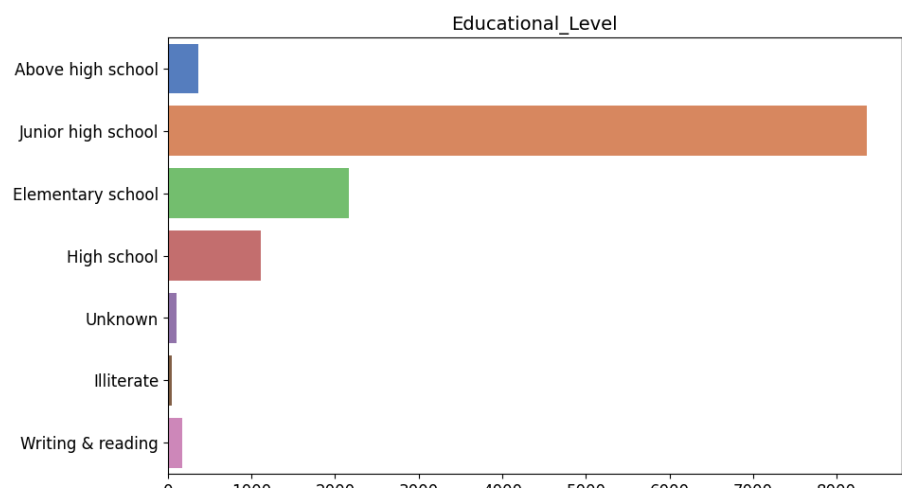
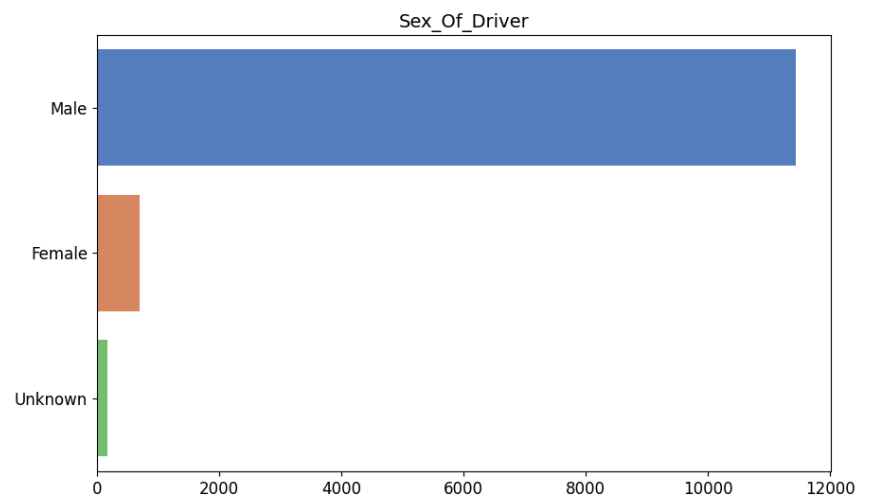
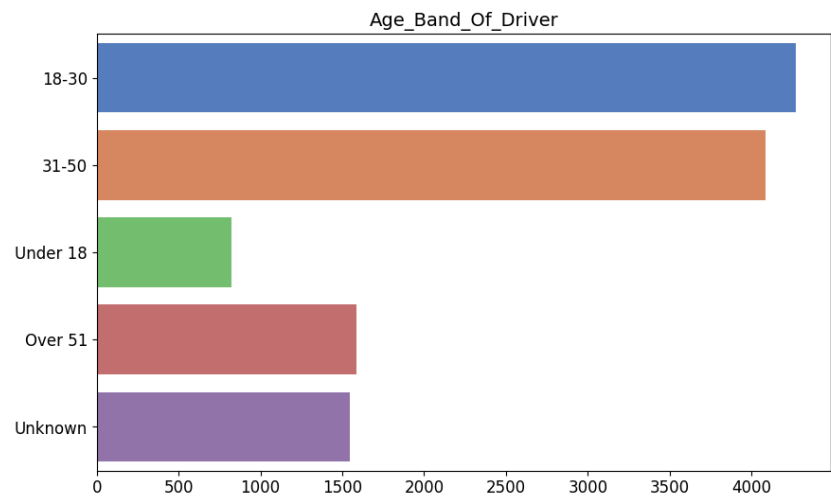
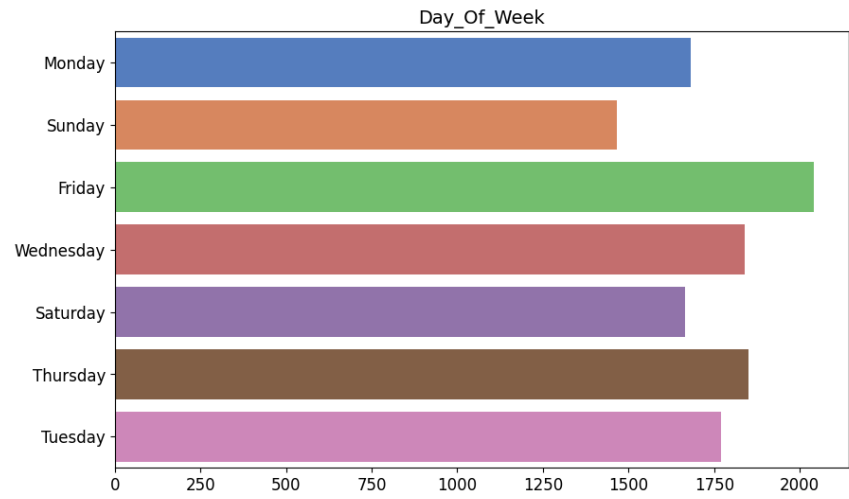
The numerica variables are ['Number_of_vehicles_involved', 'Number_of_casualties']

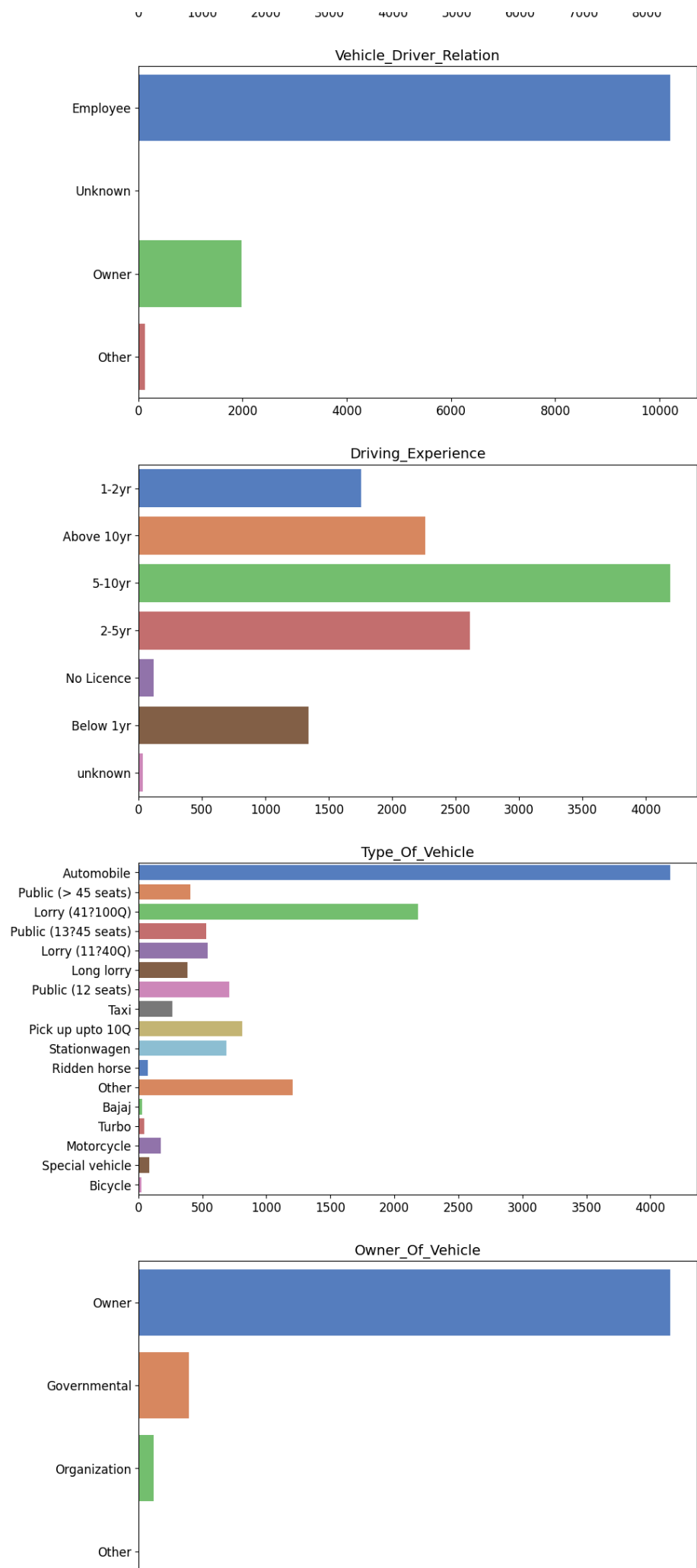
```
In [19]: plt.figure(figsize=(10,10))
          plotnumber = 1
          for i in numerical:
              if plotnumber <= df.shape[1]:
                  ax1 = plt.subplot(2,2,plotnumber)
                  plt.hist(df[i],color='red')
                  plt.xticks(fontsize=12)
                  plt.yticks(fontsize=12)
                  plt.title('frequency of '+i, fontsize=10)
                  plotnumber +=1
```

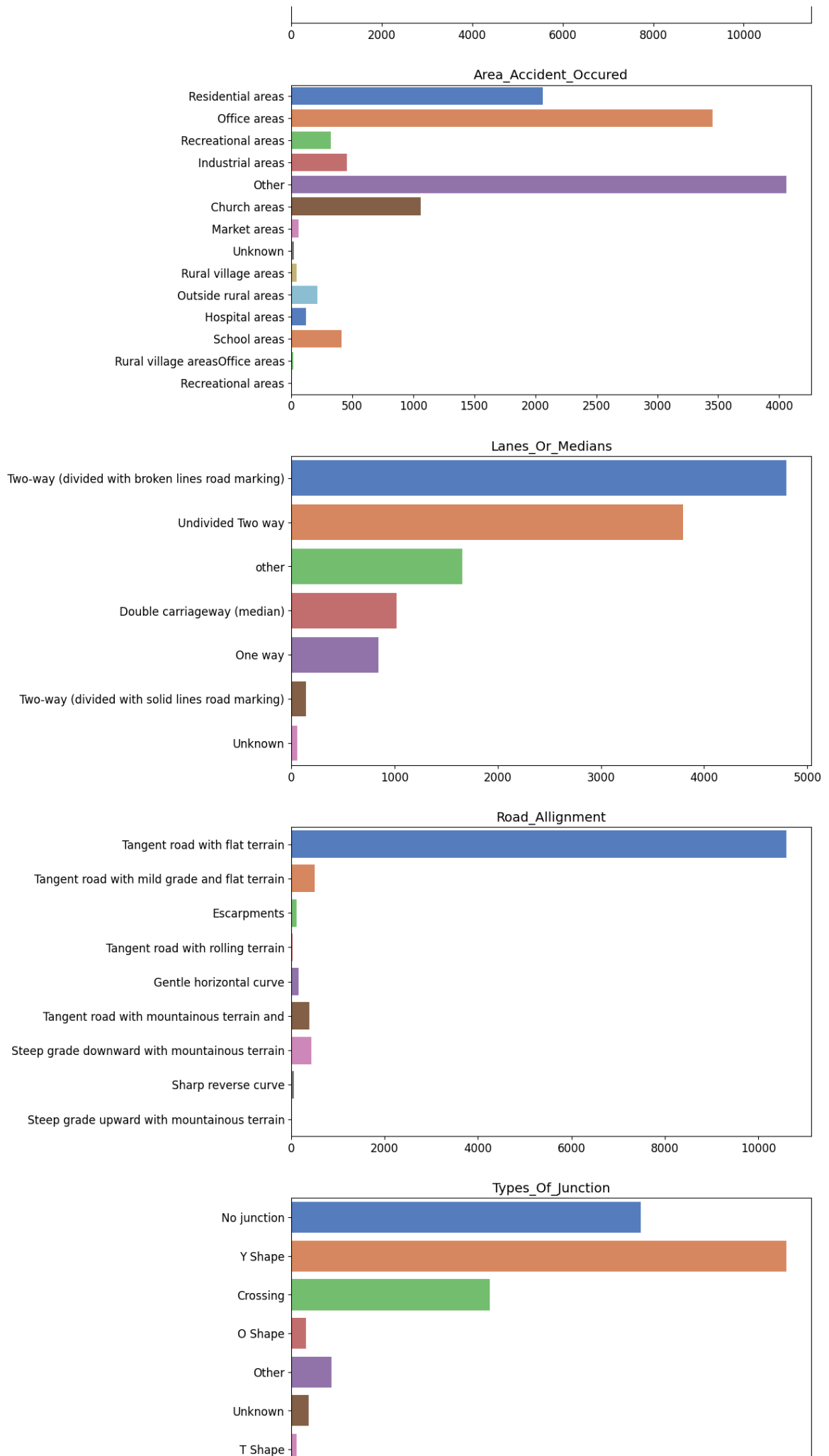



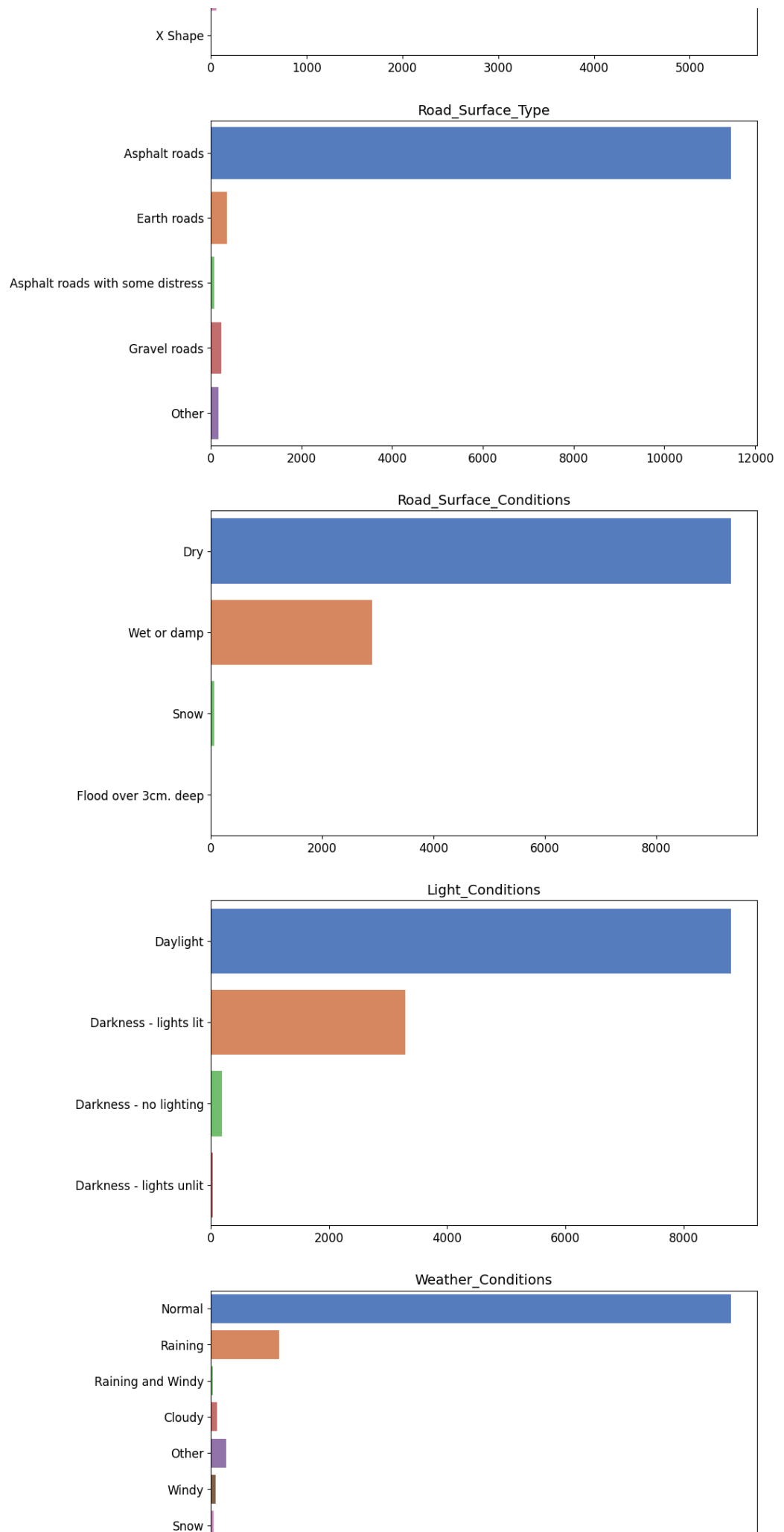
```
In [20]: plt.figure(figsize=(10,200))
plotnumber = 1

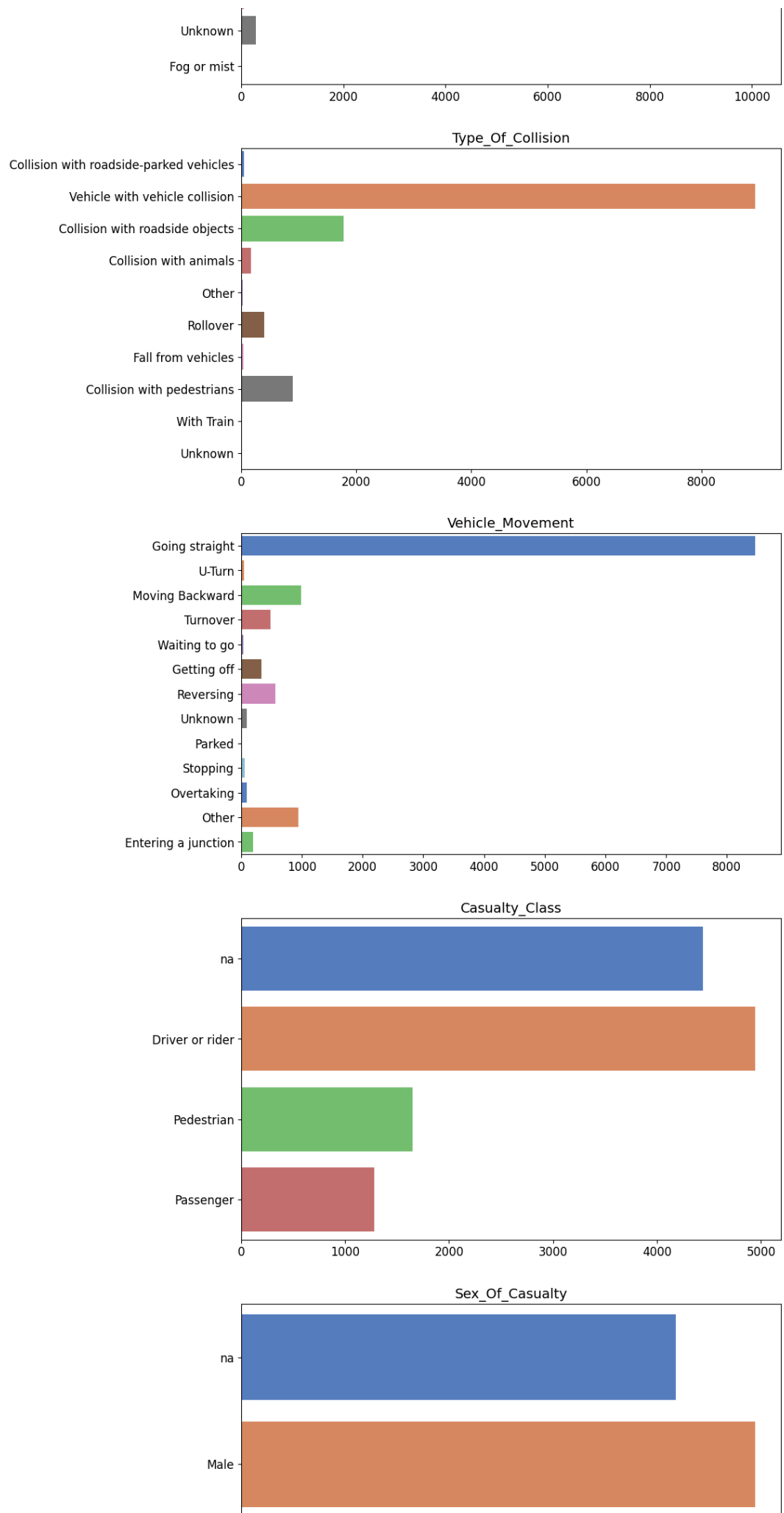
for col in categorical:
    if plotnumber <= df.shape[1] and col!='Pedestrian_movement':
        ax1 = plt.subplot(28,1,plotnumber)
        sns.countplot(data=df, y=col, palette='muted')
        plt.xticks(fontsize=12)
        plt.yticks(fontsize=12)
        plt.title(col.title(), fontsize=14)
     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

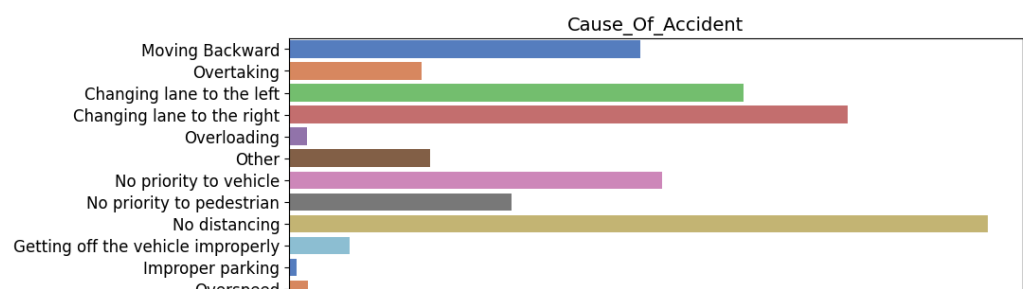
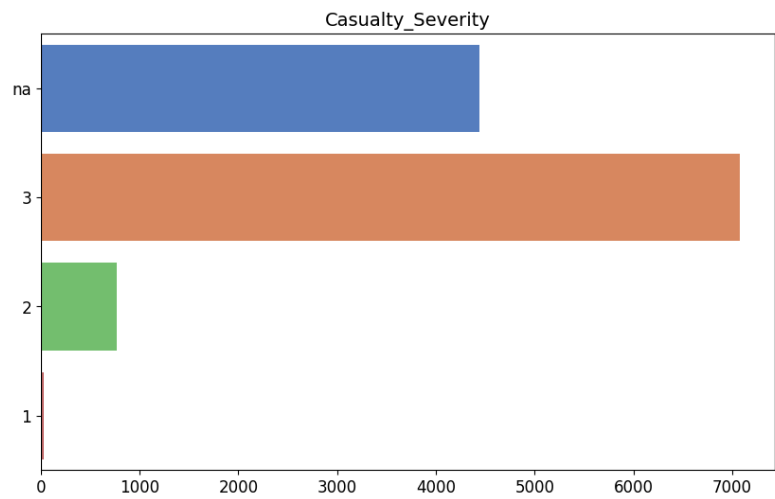
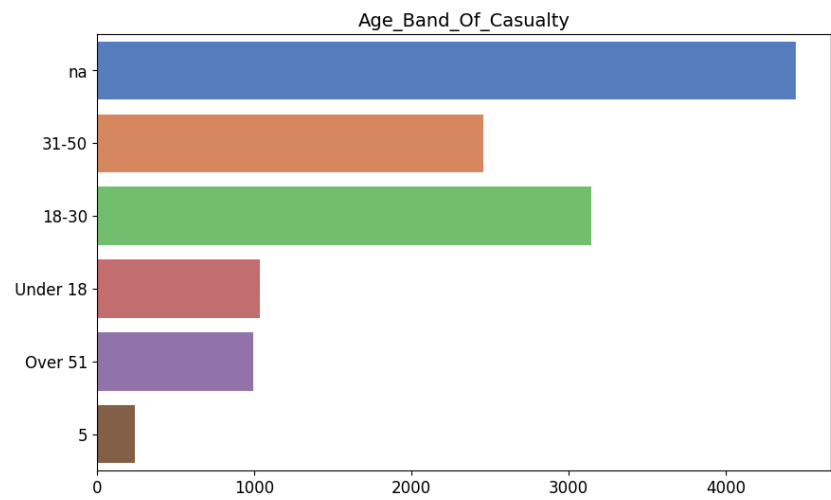
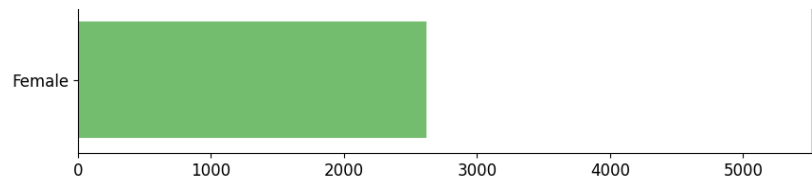


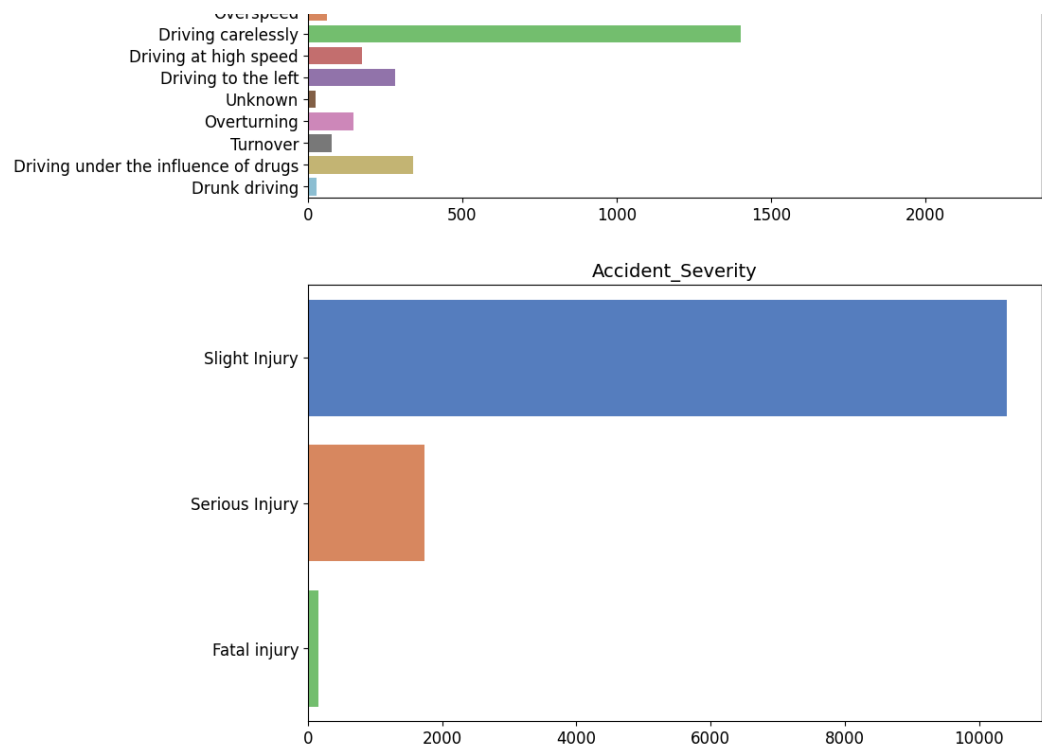












In [21]: `df.dtypes`

```
Out[21]: 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
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
Pedestrian_movement        object
Cause_of_accident          object
Accident_severity          object
dtype: object
```

```
In [22]: from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()

df1=pd.DataFrame()

for i in categorical:
```



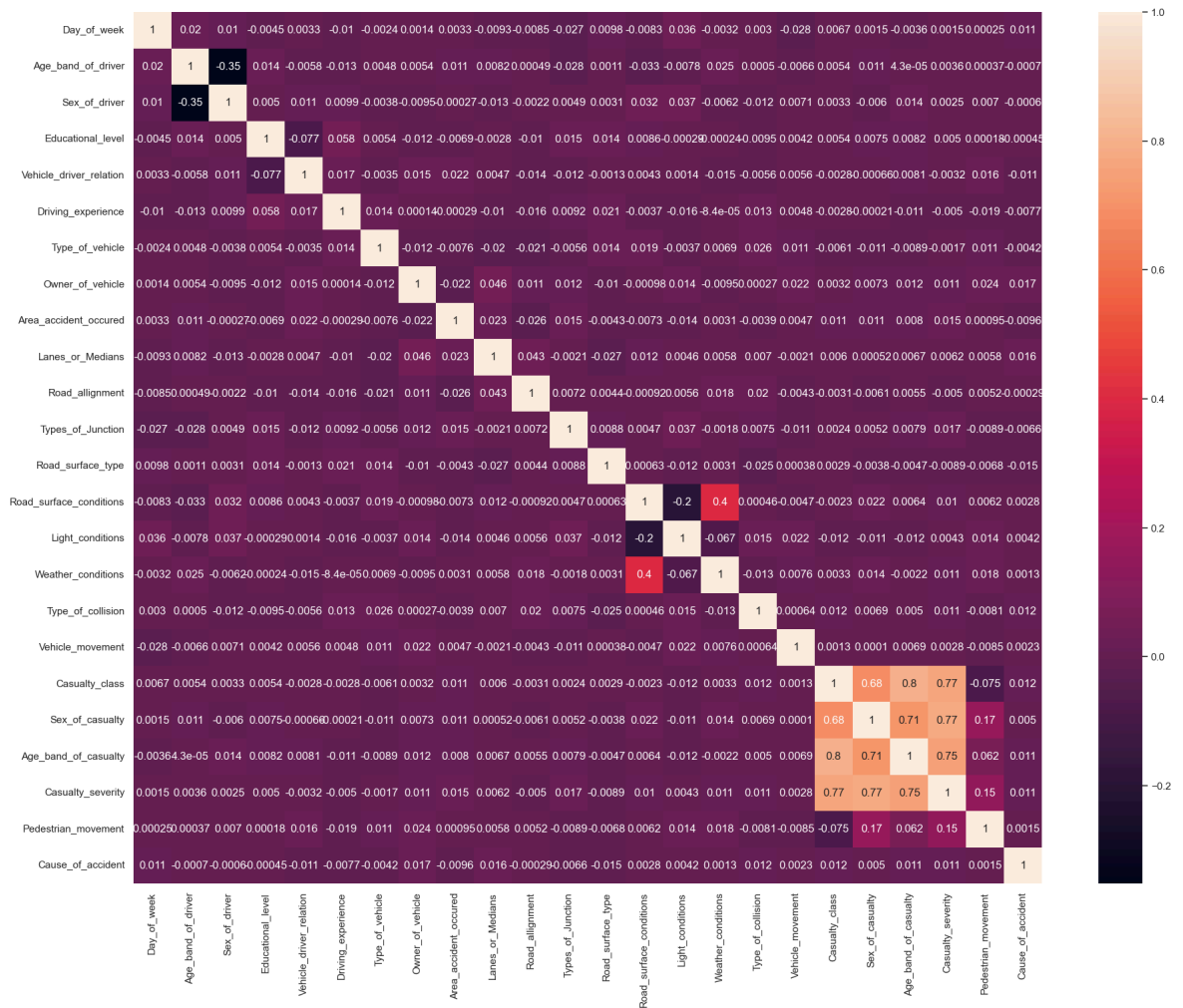
```
if i!= 'Accident_severity':
    df1[i]=le.fit_transform(df[i])
```

In [23]: df1.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12316 entries, 0 to 12315
Data columns (total 24 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Day_of_week                           12316 non-null  int64
1   Age_band_of_driver                    12316 non-null  int64
2   Sex_of_driver                         12316 non-null  int64
3   Educational_level                     12316 non-null  int64
4   Vehicle_driver_relation                12316 non-null  int64
5   Driving_experience                    12316 non-null  int64
6   Type_of_vehicle                       12316 non-null  int64
7   Owner_of_vehicle                     12316 non-null  int64
8   Area_accident_occured                 12316 non-null  int64
9   Lanes_or_Medians                     12316 non-null  int64
10  Road_alignment                        12316 non-null  int64
11  Types_of_Junction                    12316 non-null  int64
12  Road_surface_type                    12316 non-null  int64
13  Road_surface_conditions               12316 non-null  int64
14  Light_conditions                     12316 non-null  int64
15  Weather_conditions                   12316 non-null  int64
16  Type_of_collision                     12316 non-null  int64
17  Vehicle_movement                     12316 non-null  int64
18  Casualty_class                       12316 non-null  int64
19  Sex_of_casualty                      12316 non-null  int64
20  Age_band_of_casualty                 12316 non-null  int64
21  Casualty_severity                    12316 non-null  int64
22  Pedestrian_movement                 12316 non-null  int64
23  Cause_of_accident                    12316 non-null  int64
dtypes: int64(24)
memory usage: 2.3 MB
```

```
In [24]: plt.figure(figsize=(22,17))
sns.set(font_scale=1)
sns.heatmap(df1.corr(), annot=True)
```

Out[24]: <Axes: >



In [25]: `df1.head()`

Out[25]:

	Day_of_week	Age_band_of_driver	Sex_of_driver	Educational_level	Vehicle_driver_rela
0	1		0	1	0
1	1		1	1	4
2	1		0	1	4
3	3		0	1	4
4	3		0	1	4

5 rows × 24 columns



In [26]: `from sklearn.feature_selection import chi2`
`f_p_values=chi2(df1,df['Accident_severity'])`

In [27]: `f_p_values`

```
Out[27]: (array([ 0.15822071,  8.91539214,  0.1431894 ,  0.17458477,  5.34534549,
 4.49967858,  1.07767124,  1.10426215,  3.61654037,  3.28161464,
 0.1319306 ,  3.08648691,  6.99480557,  0.61510308, 16.08282359,
 1.14934538, 10.09632283,  2.20071197,  3.2168602 ,  0.12594479,
13.77841337,  0.20273788,  0.39747982,  3.19366551]),
array([9.23937958e-01, 1.15890328e-02, 9.30908116e-01, 9.16409114e-01,
6.90673790e-02, 1.05416165e-01, 5.83427189e-01, 5.75721597e-01,
1.63937473e-01, 1.93823502e-01, 9.36163348e-01, 2.13686893e-01,
3.02759144e-02, 7.35244973e-01, 3.21854237e-04, 5.62889079e-01,
6.42112839e-03, 3.32752607e-01, 2.00201664e-01, 9.38969394e-01,
1.01872169e-03, 9.03599597e-01, 8.19763078e-01, 2.02536988e-01]))
```

```
In [28]: f_p_values1=pd.DataFrame({'features':df1.columns, 'Fscore': f_p_values[0], 'Pval':
f_p_values1
```

Out[28]:

	features	Fscore	Pvalues
0	Day_of_week	0.158221	0.923938
1	Age_band_of_driver	8.915392	0.011589
2	Sex_of_driver	0.143189	0.930908
3	Educational_level	0.174585	0.916409
4	Vehicle_driver_relation	5.345345	0.069067
5	Driving_experience	4.499679	0.105416
6	Type_of_vehicle	1.077671	0.583427
7	Owner_of_vehicle	1.104262	0.575722
8	Area_accident_occured	3.616540	0.163937
9	Lanes_or_Medians	3.281615	0.193824
10	Road_allignment	0.131931	0.936163
11	Types_of_Junction	3.086487	0.213687
12	Road_surface_type	6.994806	0.030276
13	Road_surface_conditions	0.615103	0.735245
14	Light_conditions	16.082824	0.000322
15	Weather_conditions	1.149345	0.562889
16	Type_of_collision	10.096323	0.006421
17	Vehicle_movement	2.200712	0.332753
18	Casualty_class	3.216860	0.200202
19	Sex_of_casualty	0.125945	0.938969
20	Age_band_of_casualty	13.778413	0.001019
21	Casualty_severity	0.202738	0.903600
22	Pedestrian_movement	0.397480	0.819763
23	Cause_of_accident	3.193666	0.202537

```
In [29]: f_p_values1.sort_values(by='Pvalues',ascending=True)
```

```
Out[29]:
```

	features	Fscore	Pvalues
14	Light_conditions	16.082824	0.000322
20	Age_band_of_casualty	13.778413	0.001019
16	Type_of_collision	10.096323	0.006421
1	Age_band_of_driver	8.915392	0.011589
12	Road_surface_type	6.994806	0.030276
4	Vehicle_driver_relation	5.345345	0.069067
5	Driving_experience	4.499679	0.105416
8	Area_accident_occured	3.616540	0.163937
9	Lanes_or_Medians	3.281615	0.193824
18	Casualty_class	3.216860	0.200202
23	Cause_of_accident	3.193666	0.202537
11	Types_of_Junction	3.086487	0.213687
17	Vehicle_movement	2.200712	0.332753
15	Weather_conditions	1.149345	0.562889
7	Owner_of_vehicle	1.104262	0.575722
6	Type_of_vehicle	1.077671	0.583427
13	Road_surface_conditions	0.615103	0.735245
22	Pedestrian_movement	0.397480	0.819763
21	Casualty_severity	0.202738	0.903600
3	Educational_level	0.174585	0.916409
0	Day_of_week	0.158221	0.923938
2	Sex_of_driver	0.143189	0.930908
10	Road_allignment	0.131931	0.936163
19	Sex_of_casualty	0.125945	0.938969

```
In [30]: df2=df.drop(['Owner_of_vehicle', 'Type_of_vehicle', 'Road_surface_conditions', '
                'Casualty_severity', 'Educational_level', 'Day_of_week', 'Sex_of_driver', '
                'Sex_of_casualty'],axis=1)
df2.head()
```

Out[30]:

	Age_band_of_driver	Vehicle_driver_relation	Driving_experience	Area_accident_occurec
--	--------------------	-------------------------	--------------------	-----------------------

0	18-30	Employee	1-2yr	Residential areas
---	-------	----------	-------	-------------------

1	31-50	Employee	Above 10yr	Office areas
---	-------	----------	------------	--------------

2	18-30	Employee	1-2yr	Recreational areas
---	-------	----------	-------	--------------------

3	18-30	Employee	5-10yr	Office areas
---	-------	----------	--------	--------------

4	18-30	Employee	2-5yr	Industrial areas
---	-------	----------	-------	------------------



In [31]: df2.shape

Out[31]: (12316, 17)

In [32]: df2.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12316 entries, 0 to 12315
Data columns (total 17 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Age_band_of_driver                    12316 non-null  object
1   Vehicle_driver_relation                12316 non-null  object
2   Driving_experience                     12316 non-null  object
3   Area_accident_occured                  12316 non-null  object
4   Lanes_or_Medians                       12316 non-null  object
5   Types_of_Junction                     12316 non-null  object
6   Road_surface_type                     12316 non-null  object
7   Light_conditions                       12316 non-null  object
8   Weather_conditions                    12316 non-null  object
9   Type_of_collision                     12316 non-null  object
10  Number_of_vehicles_involved            12316 non-null  int64
11  Number_of_casualties                   12316 non-null  int64
12  Vehicle_movement                       12316 non-null  object
13  Casualty_class                         12316 non-null  object
14  Age_band_of_casualty                   12316 non-null  object
15  Cause_of_accident                      12316 non-null  object
16  Accident_severity                      12316 non-null  object
dtypes: int64(2), object(15)
memory usage: 1.6+ MB
```

In [33]: categorical_new=[i for i in df2.columns if df2[i].dtype=='O']
print(categorical_new)

```
['Age_band_of_driver', 'Vehicle_driver_relation', 'Driving_experience', 'Area_accident_occured', 'Lanes_or_Medians', 'Types_of_Junction', 'Road_surface_type', 'Light_conditions', 'Weather_conditions', 'Type_of_collision', 'Vehicle_movement', 'Casualty_class', 'Age_band_of_casualty', 'Cause_of_accident', 'Accident_severity']
```

```
In [34]: for i in categorical_new:  
         print(df2[i].value_counts())
```

```

Age_band_of_driver
18-30      4271
31-50      4087
Over 51     1585
Unknown    1548
Under 18    825
Name: count, dtype: int64
Vehicle_driver_relation
Employee    10206
Owner       1973
Other       123
Unknown     14
Name: count, dtype: int64
Driving_experience
5-10yr      4192
2-5yr       2613
Above 10yr   2262
1-2yr       1756
Below 1yr    1342
No Licence   118
unknown      33
Name: count, dtype: int64
Area_accident_occured
Other                        4058
Office areas                 3451
Residential areas            2060
  Church areas               1060
  Industrial areas           456
School areas                 415
  Recreational areas         327
  Outside rural areas        218
  Hospital areas             121
  Market areas               63
Rural village areas          44
Unknown                      22
Rural village areasOffice areas 20
Recreational areas           1
Name: count, dtype: int64
Lanes_or_Medians
Two-way (divided with broken lines road marking) 4796
Undivided Two way                                3796
other                                              1660
Double carriageway (median)                      1020
One way                                            845
Two-way (divided with solid lines road marking)  142
Unknown                                           57
Name: count, dtype: int64
Types_of_Junction
Y Shape      5430
No junction  3837
Crossing     2177
Other        445
Unknown     191
O Shape     164
T Shape     60
X Shape     12
Name: count, dtype: int64
Road_surface_type
Asphalt roads 11468
Earth roads   358

```

Gravel roads	242
Other	167
Asphalt roads with some distress	81
Name: count, dtype: int64	
Light_conditions	
Daylight	8798
Darkness - lights lit	3286
Darkness - no lighting	192
Darkness - lights unlit	40
Name: count, dtype: int64	
Weather_conditions	
Normal	10063
Raining	1331
Other	296
Unknown	292
Cloudy	125
Windy	98
Snow	61
Raining and Windy	40
Fog or mist	10
Name: count, dtype: int64	
Type_of_collision	
Vehicle with vehicle collision	8929
Collision with roadside objects	1786
Collision with pedestrians	896
Rollover	397
Collision with animals	171
Collision with roadside-parked vehicles	54
Fall from vehicles	34
Other	26
Unknown	14
With Train	9
Name: count, dtype: int64	
Vehicle_movement	
Going straight	8466
Moving Backward	985
Other	937
Reversing	563
Turnover	489
Getting off	339
Entering a junction	193
Overtaking	96
Unknown	88
Stopping	61
U-Turn	50
Waiting to go	39
Parked	10
Name: count, dtype: int64	
Casualty_class	
Driver or rider	4944
na	4443
Pedestrian	1649
Passenger	1280
Name: count, dtype: int64	
Age_band_of_casualty	
na	4443
18-30	3145
31-50	2455
Under 18	1035
Over 51	994


```

5          244
Name: count, dtype: int64
Cause_of_accident
No distancing          2263
Changing lane to the right 1808
Changing lane to the left 1473
Driving carelessly      1402
No priority to vehicle   1207
Moving Backward         1137
No priority to pedestrian 721
Other                   456
Overtaking              430
Driving under the influence of drugs 340
Driving to the left      284
Getting off the vehicle improperly 197
Driving at high speed    174
Overturning             149
Turnover                78
Overspeed               61
Overloading             59
Drunk driving           27
Improper parking        25
Unknown                 25
Name: count, dtype: int64
Accident_severity
Slight Injury    10415
Serious Injury   1743
Fatal injury     158
Name: count, dtype: int64

```

```

In [35]: dummy=pd.get_dummies(df2[['Age_band_of_driver', 'Vehicle_driver_relation', 'Drive
      'Area_accident_occured', 'Lanes_or_Medians', 'Types_of
      'Light_conditions', 'Weather_conditions', 'Type_of_col
      'Casualty_class', 'Age_band_of_casualty', 'Cause_of_ac
dummy.head()

```

```

Out[35]:
   Age_band_of_driver_31-50  Age_band_of_driver_Over 51  Age_band_of_driver_Under 18  Age_bai
0                False                False                False
1                 True                False                False
2                False                False                False
3                False                False                False
4                False                False                False

```

5 rows × 102 columns



```

In [36]: df3=pd.concat([df2,dummy],axis=1)
df3.head()

```

Out[36]:

	Age_band_of_driver	Vehicle_driver_relation	Driving_experience	Area_accident_occurec
0	18-30	Employee	1-2yr	Residential areas
1	31-50	Employee	Above 10yr	Office areas
2	18-30	Employee	1-2yr	Recreational areas
3	18-30	Employee	5-10yr	Office areas
4	18-30	Employee	2-5yr	Industrial areas

5 rows × 119 columns

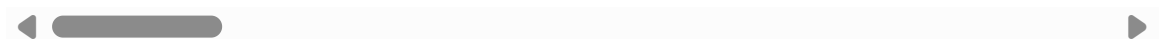


```
In [37]: df3.drop(['Age_band_of_driver', 'Vehicle_driver_relation', 'Driving_experience',  
                'Types_of_Junction', 'Road_surface_type', 'Light_conditions', 'Weather',  
                'Vehicle_movement', 'Casualty_class', 'Age_band_of_casualty', 'Cause_of  
df3.head()
```

Out[37]:

	Number_of_vehicles_involved	Number_of_casualties	Accident_severity	Age_band_of_d
0	2	2	Slight Injury	
1	2	2	Slight Injury	
2	2	2	Serious Injury	
3	2	2	Slight Injury	
4	2	2	Slight Injury	

5 rows × 105 columns



```
In [38]: x=df3.drop(['Accident_severity'],axis=1)  
x.shape
```

Out[38]: (12316, 104)

```
In [39]: x.head()
```

Out[39]:

	Number_of_vehicles_involved	Number_of_casualties	Age_band_of_driver_31-50	Age_ban
0	2	2	False	
1	2	2	True	
2	2	2	False	
3	2	2	False	
4	2	2	False	

5 rows × 104 columns



In []: