```python
In [2]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
#importing necessary libraries
```

```python
In [5]: df = pd.read_csv('RTA Dataset.csv')
df.head()
#dataset
```

Out[5]:

	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	...	Vehicle_movement	Casualty_class	Sex_of_casualty	Age_band_of_casualty	Casualty_severity
0	17:02:00	Monday	18-30	Male	Above high school	Employee	1-2yr	Automobile	Owner	Above 10yr	...	Going straight	na	na	na	na
1	17:02:00	Monday	31-50	Male	Junior high school	Employee	Above 10yr	Public (> 45 seats)	Owner	5-10yrs	...	Going straight	na	na	na	na
2	17:02:00	Monday	18-30	Male	Junior high school	Employee	1-2yr	Lorry (41?100Q)	Owner	NaN	...	Going straight	Driver or rider	Male	31-50	3
3	1:06:00	Sunday	18-30	Male	Junior high school	Employee	5-10yr	Public (> 45 seats)	Governmental	NaN	...	Going straight	Pedestrian	Female	18-30	3
4	1:06:00	Sunday	18-30	Male	Junior high school	Employee	2-5yr	NaN	Owner	5-10yrs	...	Going straight	na	na	na	na

5 rows × 32 columns

```python
In [6]: df.shape
#display of shape
```

Out[6]: (12316, 32)

```python
In [7]: df.columns
#display columns
```

```
Out[7]: 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_casuality', 'Fitness_of_casuality', 'Pedestrian_movement',
       'Cause_of_accident', 'Accident_severity'],
      dtype='object')
```

```python
In [8]: df.isnull().sum ()
#Check Missing value
```

```
Out[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
```

```python
In [9]: df.duplicated().sum()
#check for duplicate value
```

Out[9]: 0

```python
In [11]: df.describe(include="all")
#description of data
```

Out[11]:

	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	...	Vehicle_movement	Casualty_class	Sex_of_casualty	Age_band_of_casualty
count	12316	12316	12316	12316	11575	11737	11487	11366	11834	8388	...	12008	12316	12316	12316
unique	NaN	7	5	3	7	4	7	17	4	6	...	13	4	3	6
top	NaN	Friday	18-30	Male	Junior high school	Employee	5-10yr	Automobile	Owner	Unknown	...	Going straight	Driver or rider	Male	na
freq	NaN	2041	4271	11437	7619	9627	3363	3205	10459	2883	...	8158	4944	5253	4443
mean	2024-08-30 14:17:50.768106240	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
min	2024-08-30 00:01:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
25%	2024-08-30 10:31:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
50%	2024-08-30 15:10:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
75%	2024-08-30 18:10:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
max	2024-08-30 23:59:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN

11 rows × 32 columns

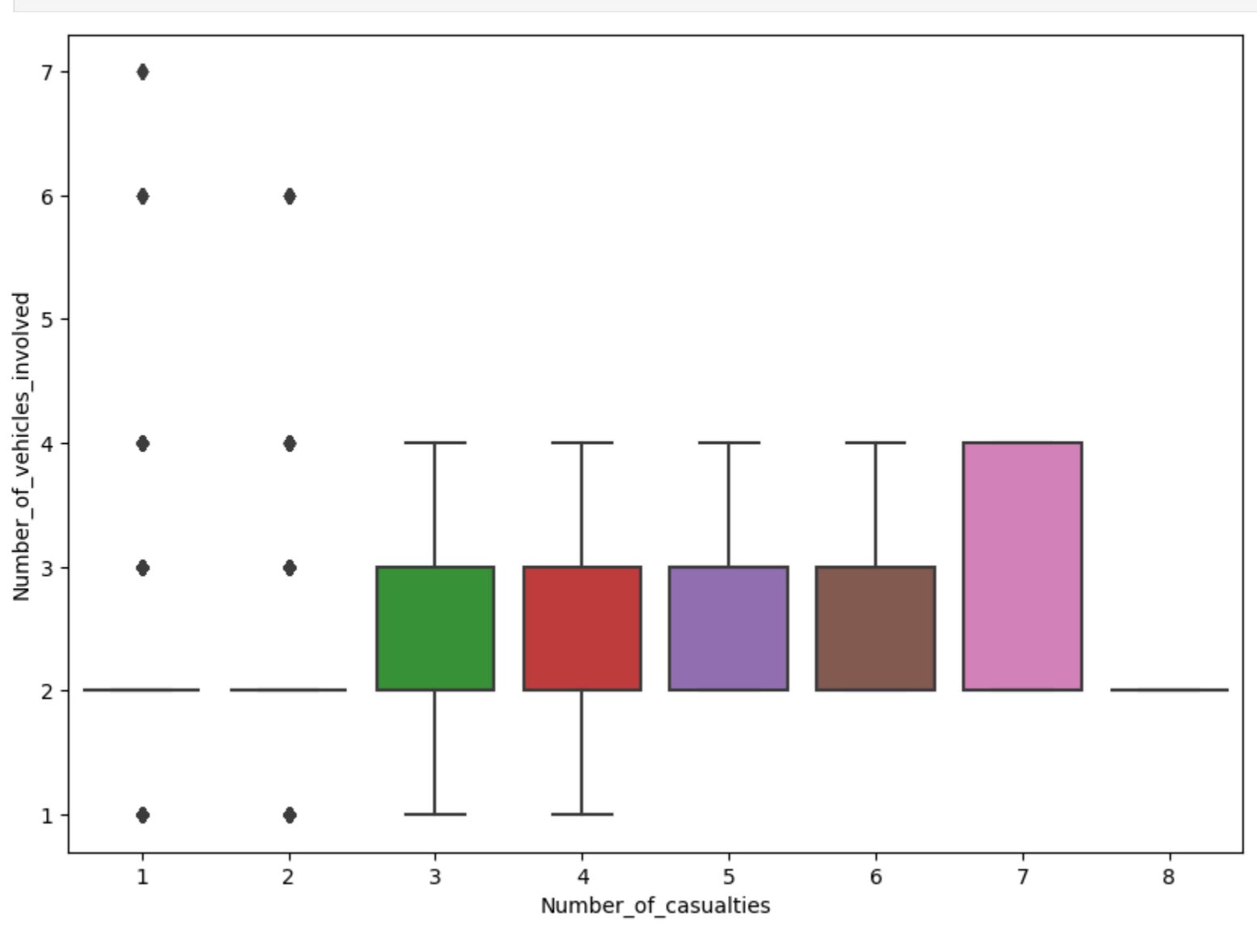
```python
In [12]: df.groupby('Accident_severity').size()
#spliting the column into groups
```

```
Out[12]: Accident_severity
Fatal injury        158
Serious Injury     1743
Slight Injury     10415
dtype: int64
```

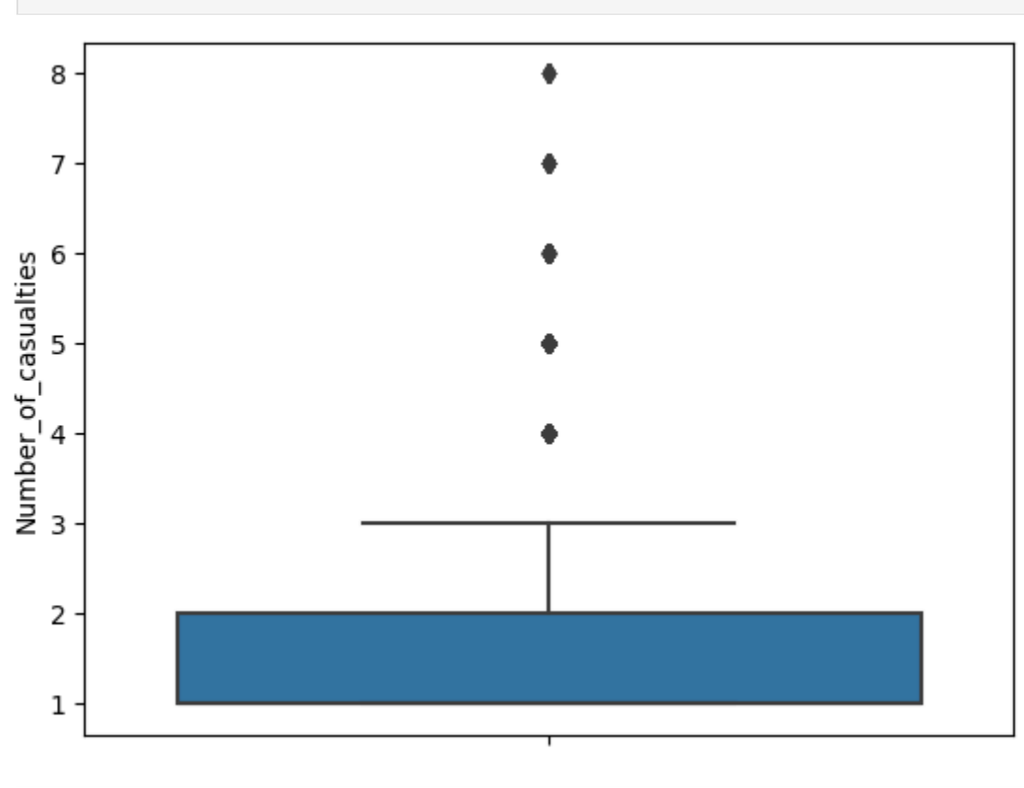
```python
In [13]: df['Number_of_casualties'].value_counts()
#value counts for each unique value in column
```

```
Out[13]: Number_of_casualties
1    8397
2    2290
3     909
4     394
5     207
6      89
7      22
8       8
Name: count, dtype: int64
```

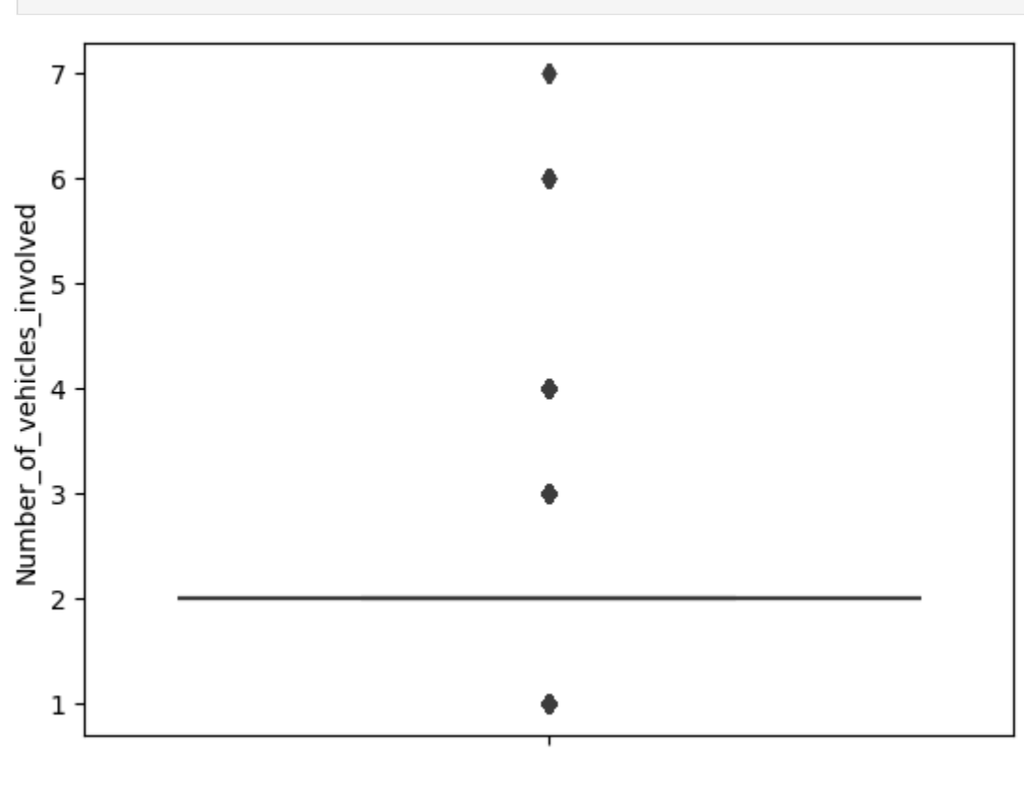
```python
In [14]: plt.figure(figsize=(10,7))
sns.boxplot(data=df, y='Number_of_vehicles_involved', x='Number_of_casualties')
plt.show()
#summary display of columns
```



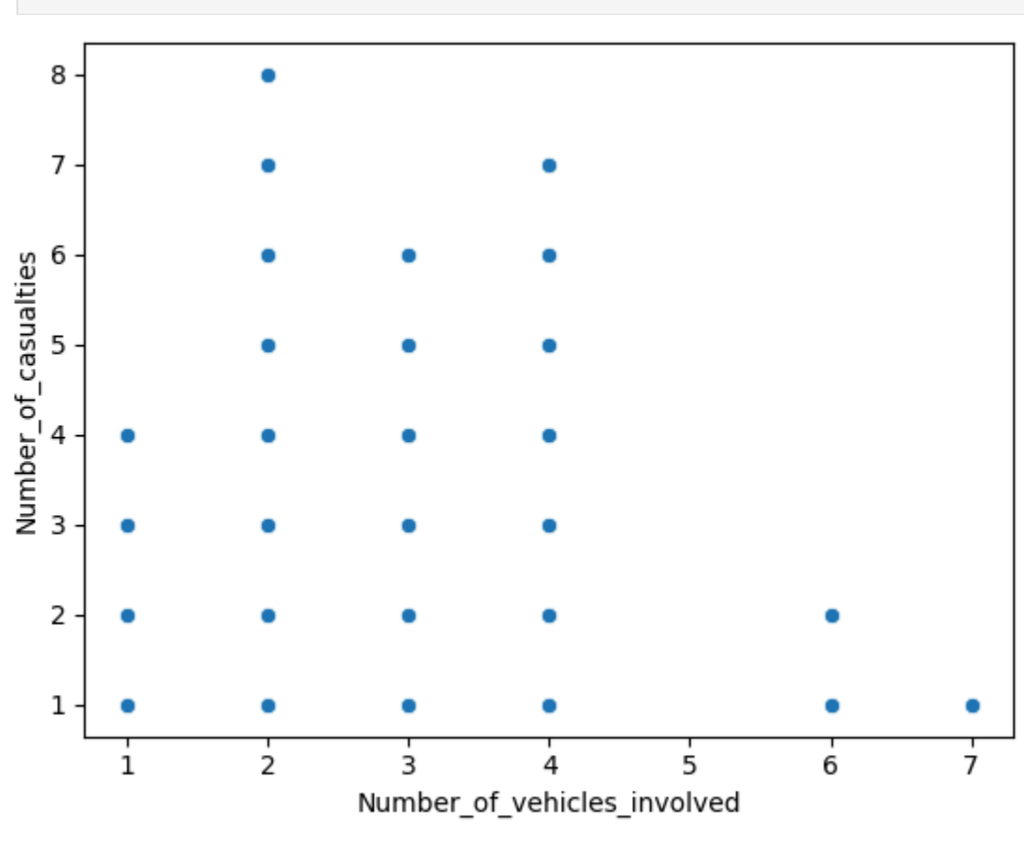

```python
In [16]: sns.boxplot(data=df, y='Number_of_casualties')
plt.show()
```




```python
In [17]: sns.boxplot(data=df, y='Number_of_vehicles_involved')
plt.show()
```

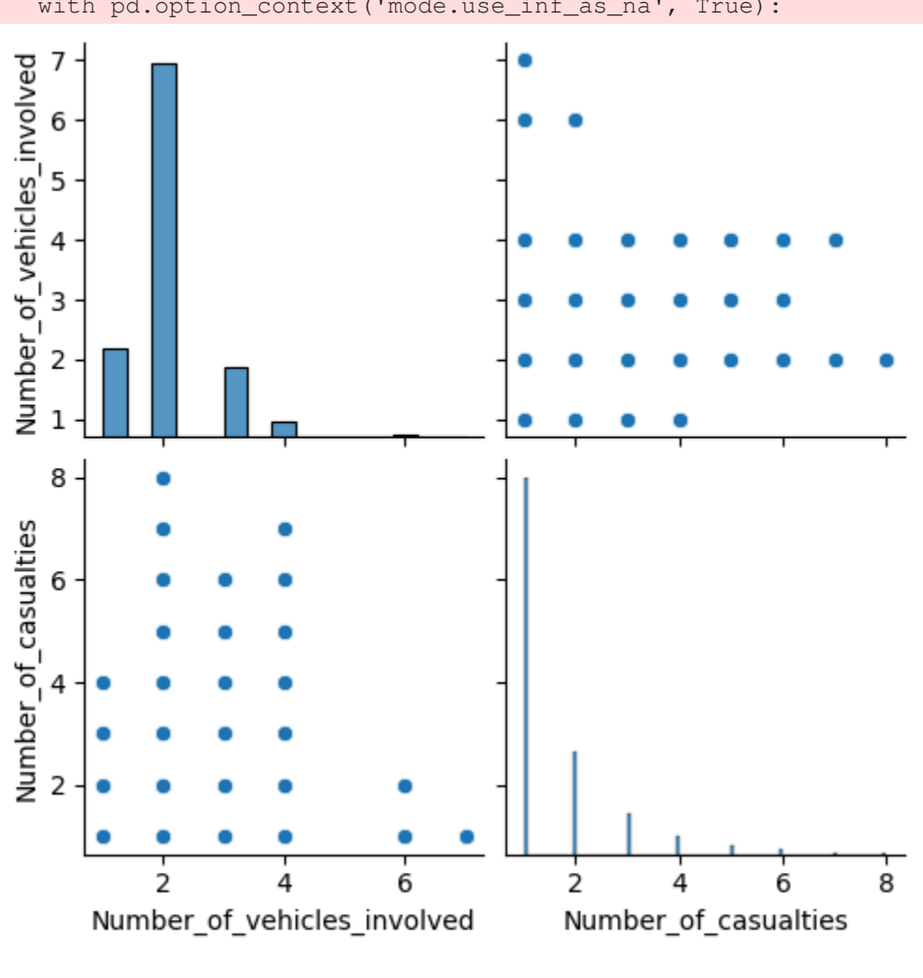



```python
In [18]: sns.scatterplot(x=df['Number_of_vehicles_involved'], y=df['Number_of_casualties'])
plt.show()
#scatter plot to determine releationships in columns
```

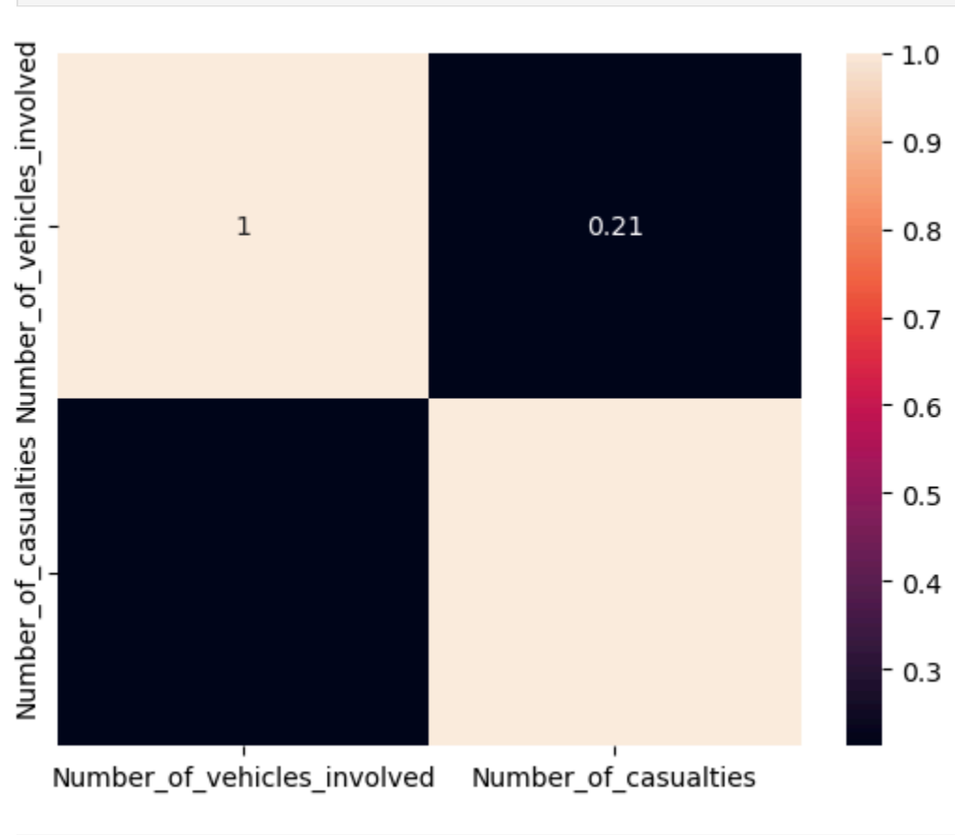



```python
In [19]: sns.pairplot(df[['Number_of_vehicles_involved','Number_of_casualties']])
plt.show()
#pairwsie releationships in dataset
```

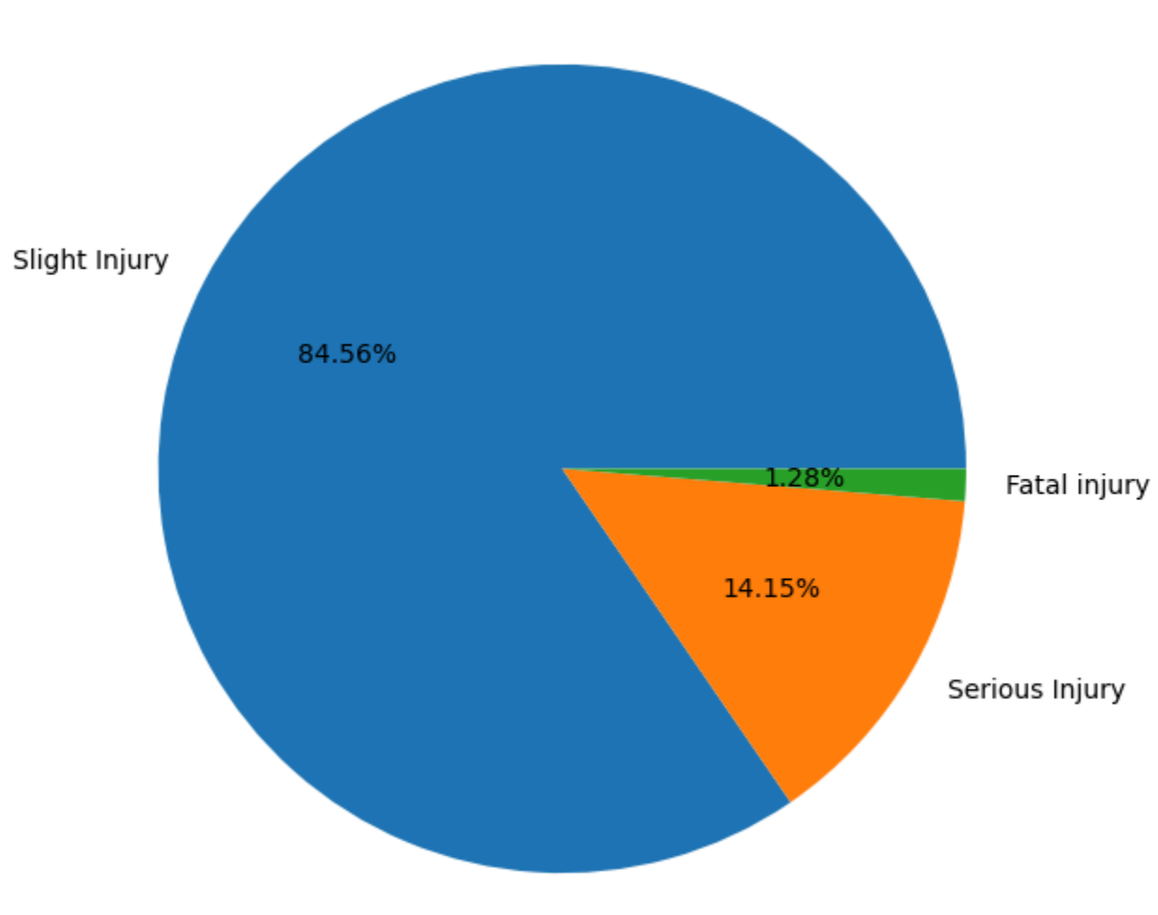
```
C:\Users\Japneet Kaur\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\Japneet Kaur\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```




```python
In [20]: correlation_matrix = df[['Number_of_vehicles_involved','Number_of_casualties']].corr()
sns.heatmap(correlation_matrix, annot=True)
plt.show()
#visplay of heapmap having correlation of columns as attribute.
```

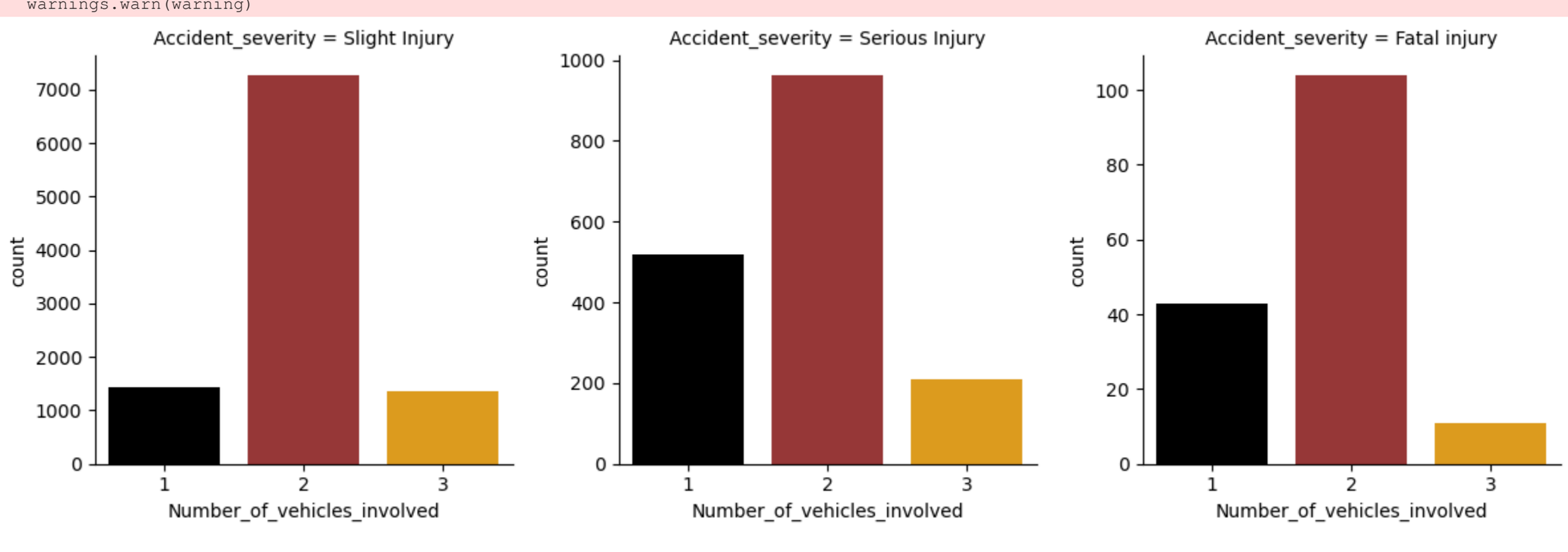



```python
In [21]: plt.figure(figsize=(10,7))
plt.pie(x=df['Accident_severity'].value_counts().values,
        labels=df['Accident_severity'].value_counts().index,
        autopct='%2.2f%%')
plt.show()
#display of pie chart
```




```python
In [22]: grid = sns.FacetGrid(data=df, col='Accident_severity', height=4, aspect=1, sharey=False)
grid.map(sns.countplot, 'Number_of_vehicles_involved', palette=['black', 'brown', 'orange'])
plt.show()
```

```
C:\Users\Japneet Kaur\anaconda3\Lib\site-packages\seaborn\axisgrid.py:712: UserWarning: Using the countplot function without specifying `order` is likely to produce an incorrect plot.
  warnings.warn(warning)
```




```python
In [23]: lists=['Vehicle_driver_relation', 'Work_of_casuality', 'Fitness_of_casuality','Day_of_week',
       'Casualty_severity','Time','Sex_of_driver','Educational_level','Defect_of_vehicle','Owner_of_vehicle','Service_year_of_vehicle',
       'Road_surface_type','Sex_of_casualty']
df.drop(columns = lists, inplace=True)
#drop the particular columns
```

```python
In [24]: df.shape
```

Out[24]: (12316, 19)

```python
In [25]: df.columns
```

```
Out[25]: Index(['Age_band_of_driver', 'Driving_experience', 'Type_of_vehicle',
       'Area_accident_occured', 'Lanes_or_Medians', 'Road_allignment',
       'Types_of_Junction', 'Road_surface_conditions', 'Light_conditions',
       'Weather_conditions', 'Type_of_collision',
       'Number_of_vehicles_involved', 'Number_of_casualties',
       'Vehicle_movement', 'Casualty_class', 'Age_band_of_casualty',
       'Pedestrian_movement', 'Cause_of_accident', 'Accident_severity'],
      dtype='object')
```

```python
In [26]: df['Driving_experience'].fillna(df['Driving_experience'].mode()[0], inplace=True)
df['Type_of_vehicle'].fillna(df['Type_of_vehicle'].mode()[0], inplace=True)
df['Area_accident_occured'].fillna(df['Area_accident_occured'].mode()[0], inplace=True)
df['Road_allignment'].fillna(df['Road_allignment'].mode()[0], inplace=True)
df['Type_of_collision'].fillna(df['Type_of_collision'].mode()[0], inplace=True)
df['Vehicle_movement'].fillna(df['Vehicle_movement'].mode()[0], inplace=True)
df['Lanes_or_Medians'].fillna(df['Lanes_or_Medians'].mode()[0], inplace=True)
df['Types_of_Junction'].fillna(df['Types_of_Junction'].mode()[0], inplace=True)
#filling the missing values with mode column values
```

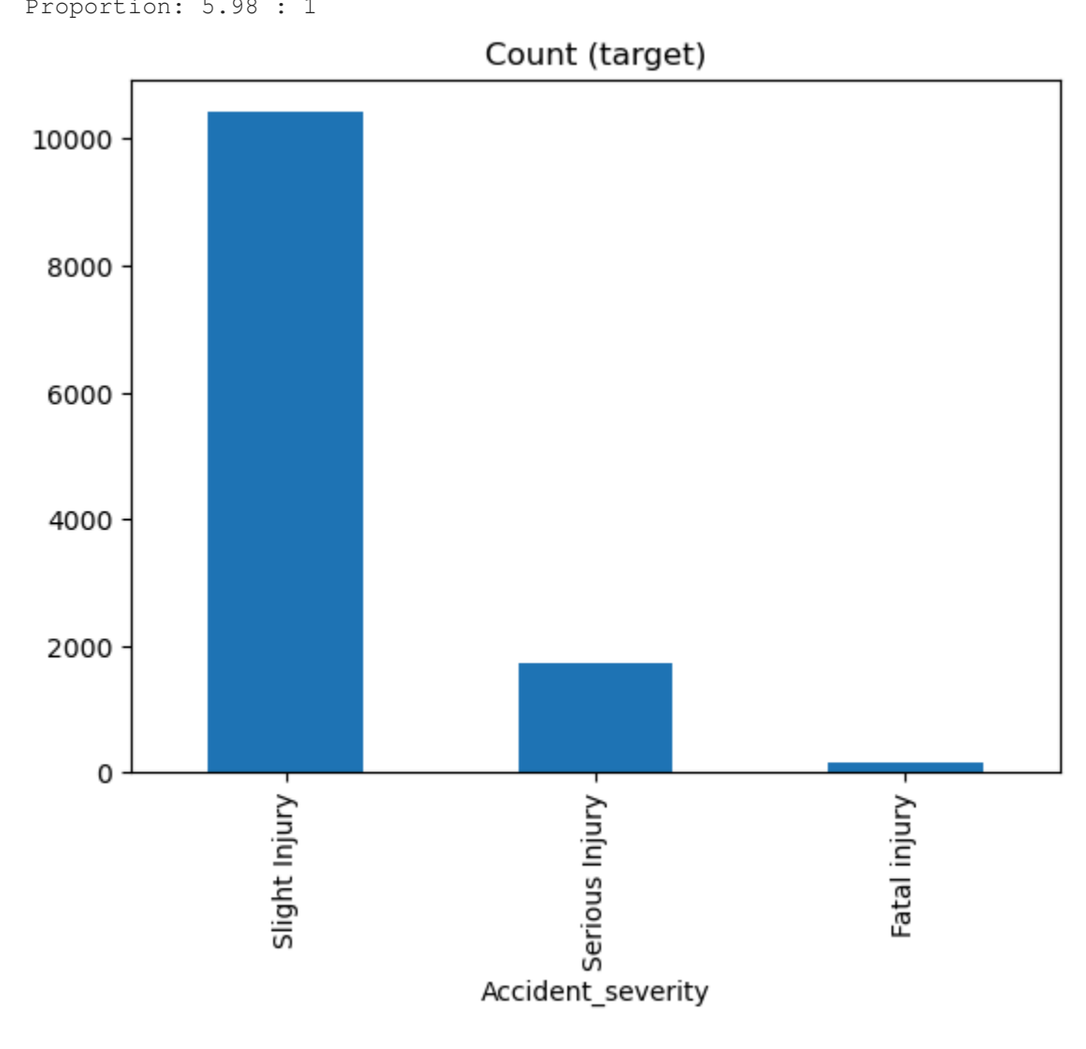
```python
In [27]: df.isnull().sum()
#checking the missing values
```

```
Out[27]: Age_band_of_driver             0
Driving_experience             0
Type_of_vehicle                0
Area_accident_occured          0
Lanes_or_Medians               0
Road_allignment                0
Types_of_Junction              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
Age_band_of_casualty           0
Pedestrian_movement            0
Cause_of_accident              0
Accident_severity              0
dtype: int64
```

```python
In [28]: target_count = df['Accident_severity'].value_counts()
print('Class 0:', target_count[0])
print('Class 1:', target_count[1])
print('Proportion:', round(target_count[0] / target_count[1], 2), ': 1')

target_count.plot(kind='bar', title='Count (target)');
#display of countplot which counts of observations in each categorical bin using bars.
```

```
C:\Users\Japneet Kaur\AppData\Local\Temp\ipykernel_1124\3174507065.py:2: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will always be treated as labels (consistent with Da
taFrame behavior). To access a value by position, use `ser.iloc[pos]`
  print('Class 0:', target_count[0])
C:\Users\Japneet Kaur\AppData\Local\Temp\ipykernel_1124\3174507065.py:3: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will always be treated as labels (consistent with Da
taFrame behavior). To access a value by position, use `ser.iloc[pos]`
  print('Class 1:', target_count[1])
C:\Users\Japneet Kaur\AppData\Local\Temp\ipykernel_1124\3174507065.py:4: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will always be treated as labels (consistent with Da
taFrame behavior). To access a value by position, use `ser.iloc[pos]`
  print('Proportion:', round(target_count[0] / target_count[1], 2), ': 1')
Class 0: 10415
Class 1: 1743
Proportion: 5.98 : 1
```




```python
In [29]: from sklearn.preprocessing import LabelEncoder
LE = LabelEncoder()
df=df.apply(LE.fit_transform)
#LabelEncoder is used to convert categorical columns into numerical ones
```

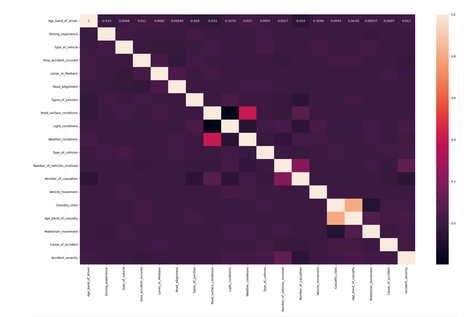
```python
In [30]: df.corr()
#display pairwise correlation of columns, excluding null values.
```

Out[30]:

	Age_band_of_driver	Driving_experience	Type_of_vehicle	Area_accident_occured	Lanes_or_Medians	Road_allignment	Types_of_Junction	Road_surface_conditions	Light_conditions	Weather_conditions	Type_of_collision	Number_of_vehicles_involved
Age_band_of_driver	1.000000	-0.013407	0.004830	0.011472	0.008176	0.000486	-0.028025	-0.032533	-0.007766	0.025482	0.000496	0.001690
Driving_experience	-0.013407	1.000000	0.014333	-0.000290	-0.010444	-0.015602	0.009245	-0.003696	-0.015577	-0.000084	0.012574	-0.008222
Type_of_vehicle	0.004830	0.014333	1.000000	-0.007631	-0.020156	-0.020910	-0.005553	0.019018	-0.003739	0.006903	0.025911	0.001140
Area_accident_occured	0.011472	-0.000290	-0.007631	1.000000	0.022760	-0.025859	0.014580	-0.007290	-0.013597	0.003127	-0.003940	-0.016498
Lanes_or_Medians	0.008176	-0.010444	-0.020156	0.022760	1.000000	0.042815	-0.002125	0.011774	0.004624	0.005828	0.007008	-0.001199
Road_allignment	0.000486	-0.015602	-0.020910	-0.025859	0.042815	1.000000	0.007188	-0.000918	0.005625	0.018304	0.020418	-0.001536
Types_of_Junction	-0.028025	0.009245	-0.005553	0.014580	-0.002125	0.007188	1.000000	0.004724	0.037063	-0.001816	0.007516	-0.020214
Road_surface_conditions	-0.032533	-0.003696	0.019018	-0.007290	0.011774	-0.000918	0.004724	1.000000	-0.198447	0.400485	0.000456	-0.020963
Light_conditions	-0.007766	-0.015577	-0.003739	-0.013597	0.004624	0.005625	0.037063	-0.198447	1.000000	-0.067367	0.014883	0.032493
Weather_conditions	0.025482	-0.000084	0.006903	0.003127	0.005828	0.018304	-0.001816	0.400485	-0.067367	1.000000	-0.013347	-0.037733
Type_of_collision	0.000496	0.012574	0.025911	-0.003940	0.007008	0.020418	0.007516	0.000456	0.014883	-0.013347	1.000000	0.007304
Number_of_vehicles_involved	0.001690	-0.008222	0.001140	-0.016498	-0.001199	-0.001536	-0.020214	-0.020963	0.032493	-0.037733	0.007304	1.000000
Number_of_casualties	-0.054474	-0.007118	0.003022	-0.004690	0.002188	-0.011434	-0.043389	0.072828	-0.055116	0.014138	-0.020694	0.220531
Vehicle_movement	-0.006562	0.004794	0.010841	0.004691	-0.002089	-0.004267	-0.010761	-0.004667	0.022327	0.007648	0.000640	-0.004480
Casualty_class	0.005386	-0.002783	-0.006116	0.011325	0.006034	-0.003117	0.002423	-0.002313	-0.012499	0.003313	0.012333	0.005281
Age_band_of_casualty	0.000043	-0.010891	-0.008934	0.008001	0.006677	0.005540	0.007946	0.006415	-0.011780	-0.002245	0.004962	0.001796
Pedestrian_movement	0.000372	-0.018790	0.010500	0.000949	0.005848	0.005219	-0.008855	0.006157	0.014229	0.018292	-0.008081	0.021109
Cause_of_accident	-0.000703	-0.007728	-0.004200	-0.009589	0.016384	-0.000287	-0.006561	0.002811	0.004213	0.001320	0.011757	-0.021815
Accident_severity	0.013185	0.000509	-0.003709	-0.018374	-0.011844	-0.005967	-0.008861	0.004579	0.025741	0.010172	0.025867	0.100205

```python
In [31]: plt.figure(figsize=[25,15])
sns.heatmap(df.corr(),annot=True)
```

Out[31]: <Axes: >


Age_band_of_driver
Driving_experience
Type_of_vehicle
Area_accident_occured
Lanes_or_Medians
Road_allignment
Types_of_Junction
Road_surface_conditions
Light_conditions
Weather_conditions
Type_of_collision
Number_of_vehicles_involved
Number_of_casualties
Vehicle_movement
Casualty_class
Age_band_of_casualty
Pedestrian_movement
Cause_of_accident
Accident_severity
1
-0.013
-0.00648
0.011
0.00082
0.00999
-0.0126
-0.033
-0.0078
0.025
0.00605
0.00157
-0.054
-0.0066
0.00254
-6.3e-05
0.00037
-0.00017
0.023
1.0
0.8
0.6
0.4
0.2
0.0