In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
import spacy
import numpy as np
```

In [2]:
```
df=pd.read_csv('cleaned.csv')
```

In [3]:
```
df.head()
```

Out[3]:

	Age_band_of_driver	Sex_of_driver	Educational_level	Vehicle_driver_relation	Driving_experienc
0	18-30	Male	Above high school	Employee	1-2
1	31-50	Male	Junior high school	Employee	Above 10
2	18-30	Male	Junior high school	Employee	1-2
3	18-30	Male	Junior high school	Employee	5-10
4	18-30	Male	Junior high school	Employee	2-5

In [4]:
```
df.rename(columns={'Sex_of_driver': 'Gender'}, inplace=True)
```

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

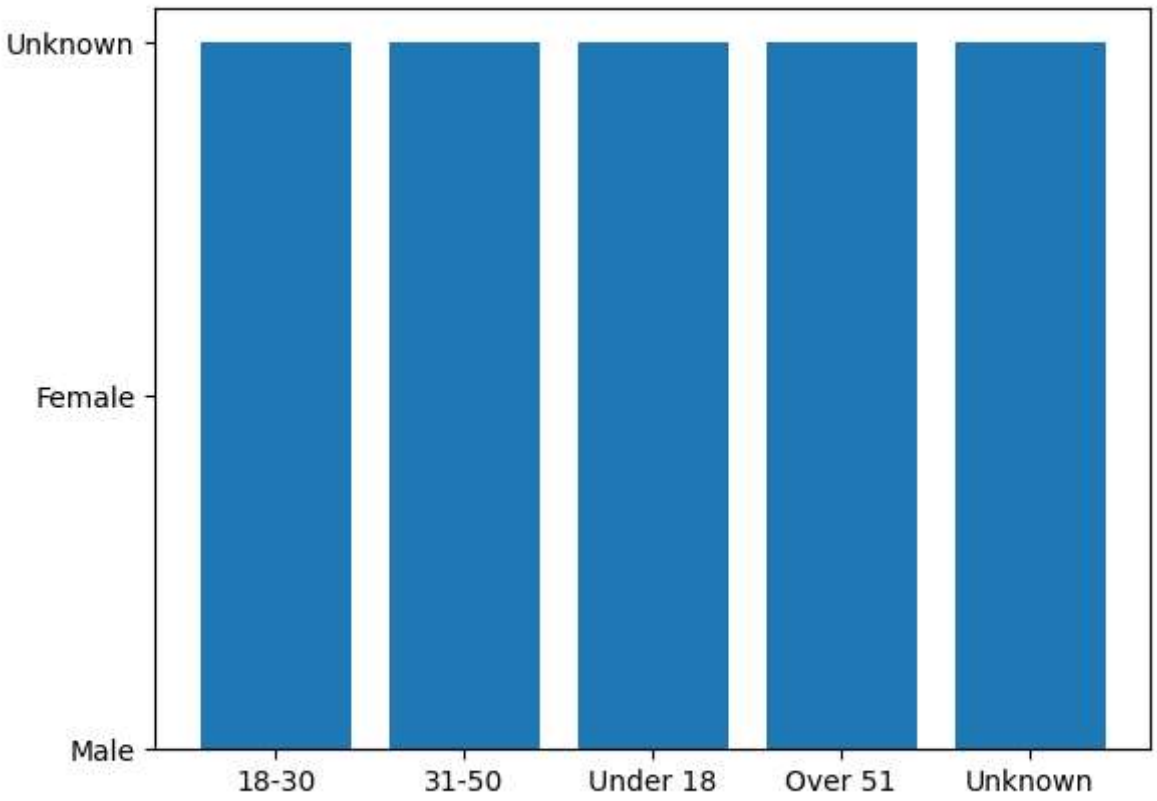
Out[5]:
```
Index(['Age_band_of_driver', 'Gender', 'Educational_level',
       'Vehicle_driver_relation', 'Driving_experience', 'Lanes_or_Medians',
       'Types_of_Junction', 'Road_surface_type', 'Light_conditions',
       'Weather_conditions', 'Type_of_collision', 'Vehicle_movement',
       'Pedestrian_movement', 'Cause_of_accident', 'Accident_severity'],
      dtype='object')
```

In [6]:
```
1 df.isna().sum()
```

Out[6]:
```
Age_band_of_driver          0
Gender                      0
Educational_level           0
Vehicle_driver_relation     0
Driving_experience          0
Lanes_or_Medians            0
Types_of_Junction           0
Road_surface_type           0
Light_conditions            0
Weather_conditions          0
Type_of_collision           0
Vehicle_movement            0
Pedestrian_movement         0
Cause_of_accident           0
Accident_severity           0
dtype: int64
```

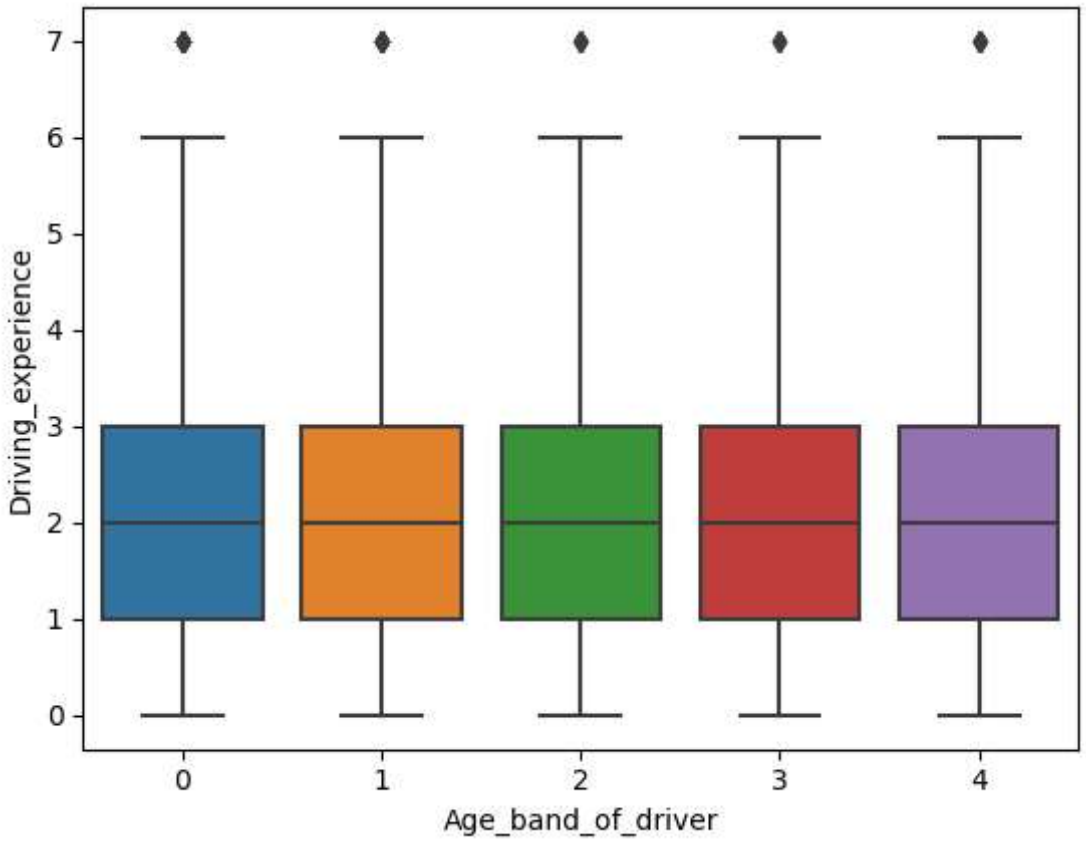
In [7]:
```
1 plt.bar(df['Age_band_of_driver'],df['Gender'])
```

Out[7]:
```
<BarContainer object of 12316 artists>
```



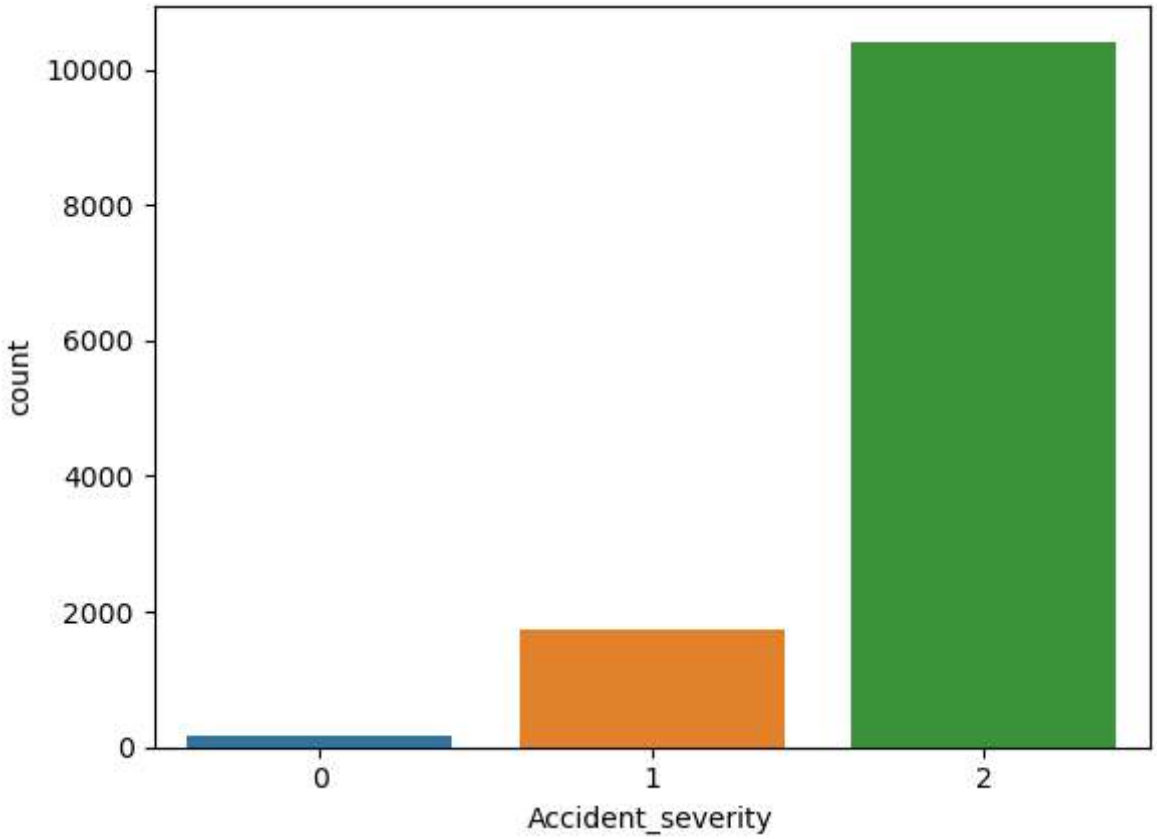
In [8]:
```
le=LabelEncoder()
df['Age_band_of_driver']=le.fit_transform(df['Age_band_of_driver'])
df['Driving_experience']=le.fit_transform(df['Driving_experience'])

sns.boxplot(x='Age_band_of_driver',y='Driving_experience',data=df)
```

Out[8]: <Axes: xlabel='Age_band_of_driver', ylabel='Driving_experience'>

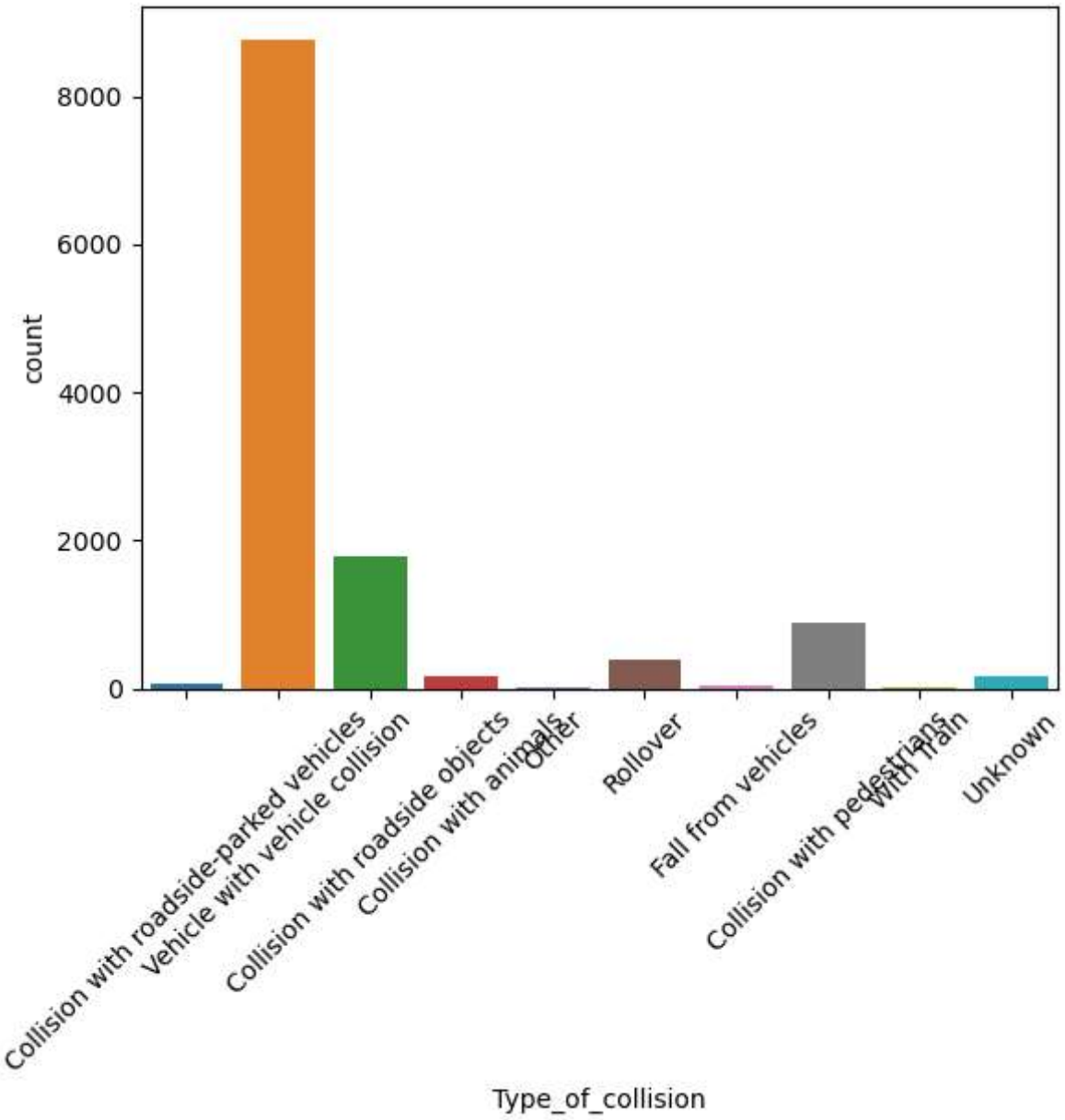


In [9]:
```
1 sns.countplot(x='Accident_severity',data=df)
```

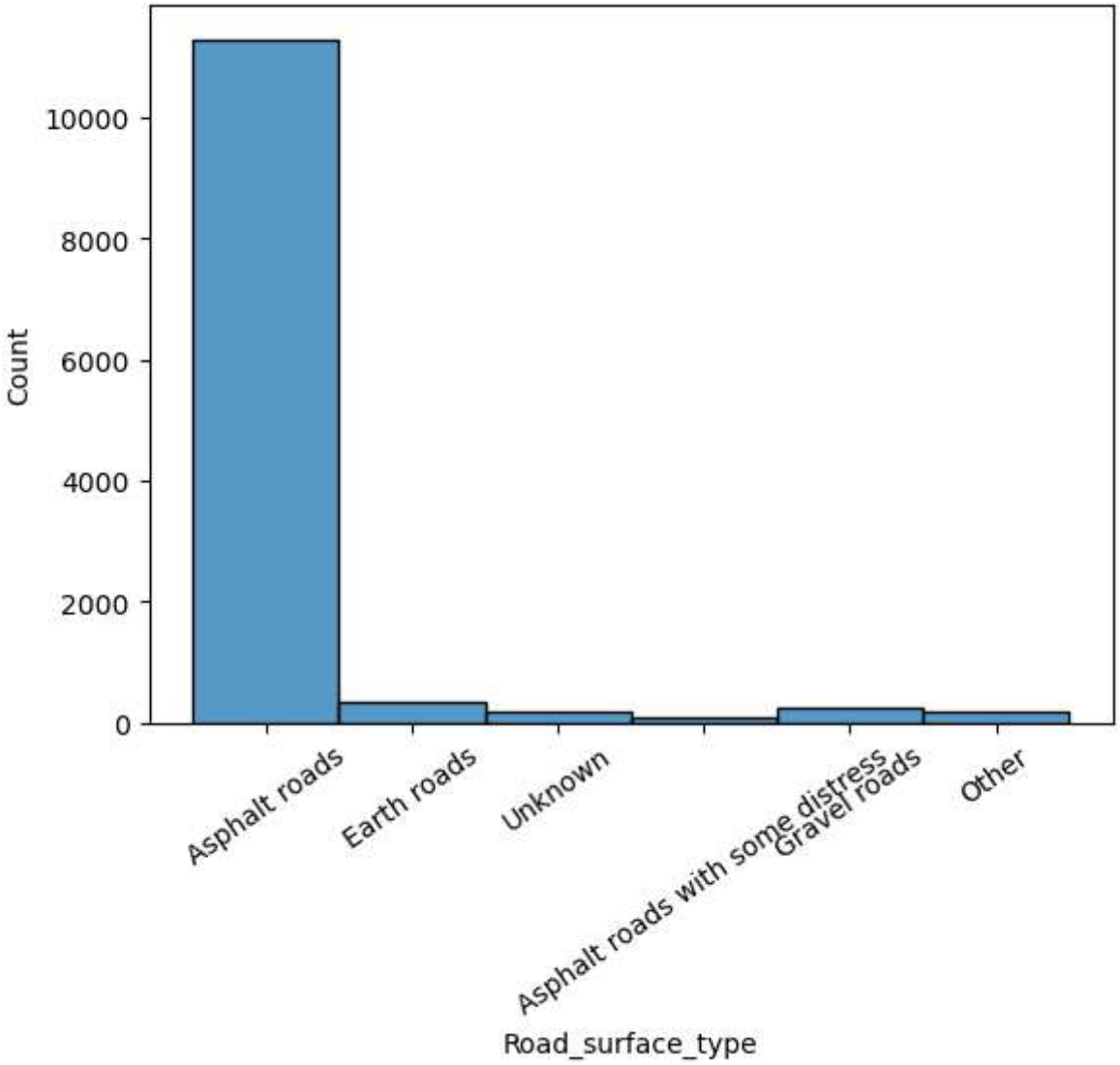
Out[9]: <Axes: xlabel='Accident_severity', ylabel='count'>



In [10]:
```
1 sns.countplot(x='Type_of_collision',data=df)
2 plt.xticks(rotation=45)
3 plt.show()
```

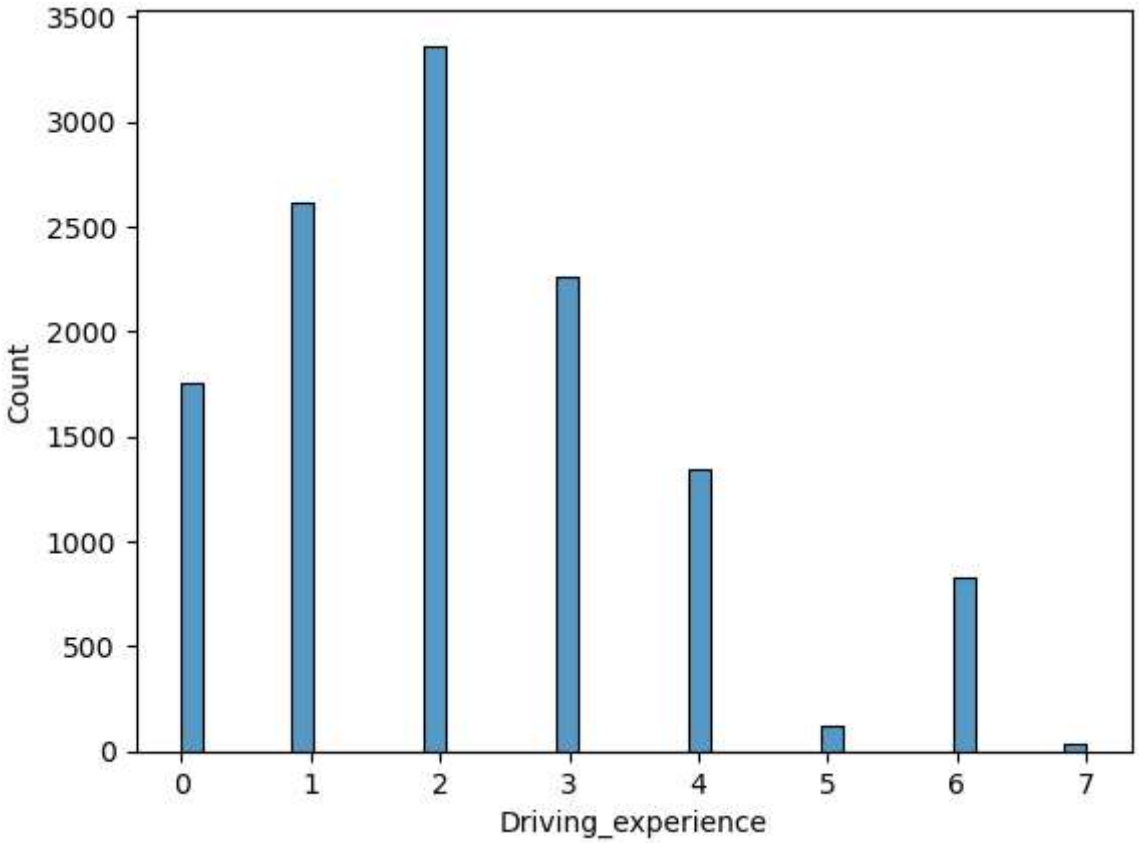


In [11]:
```
sns.histplot(x='Road_surface_type',data=df)
plt.xticks(rotation=35)
plt.show()
```



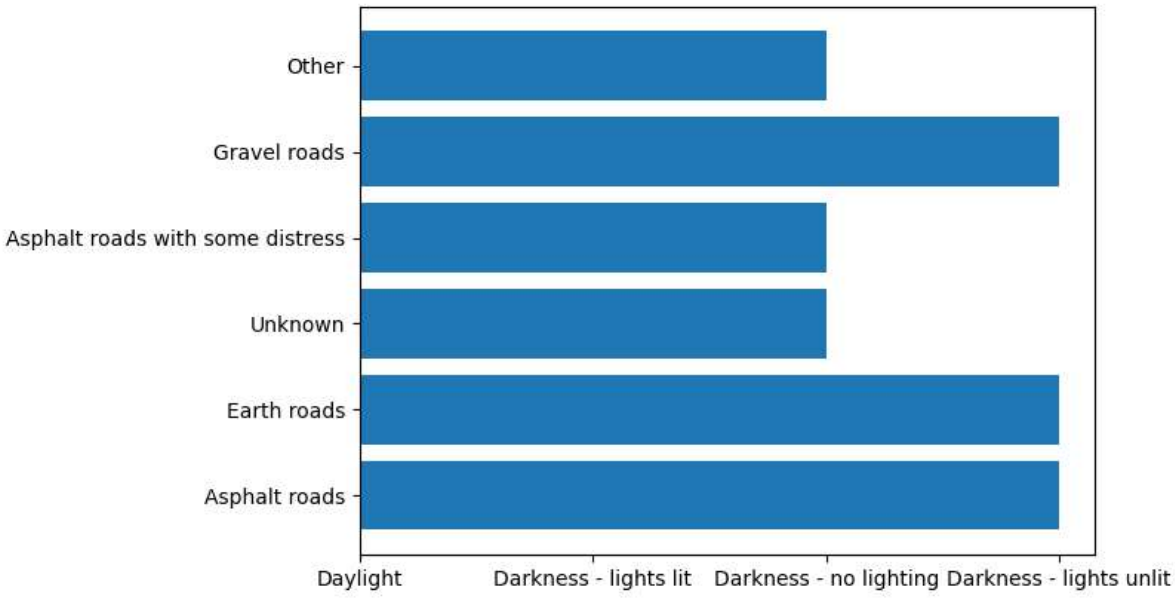
In [12]:
```
1 sns.histplot(df['Driving_experience'])
```

Out[12]: <Axes: xlabel='Driving_experience', ylabel='Count'>



In [13]:
```
1 plt.barh(df['Road_surface_type'],df['Light_conditions'])
```

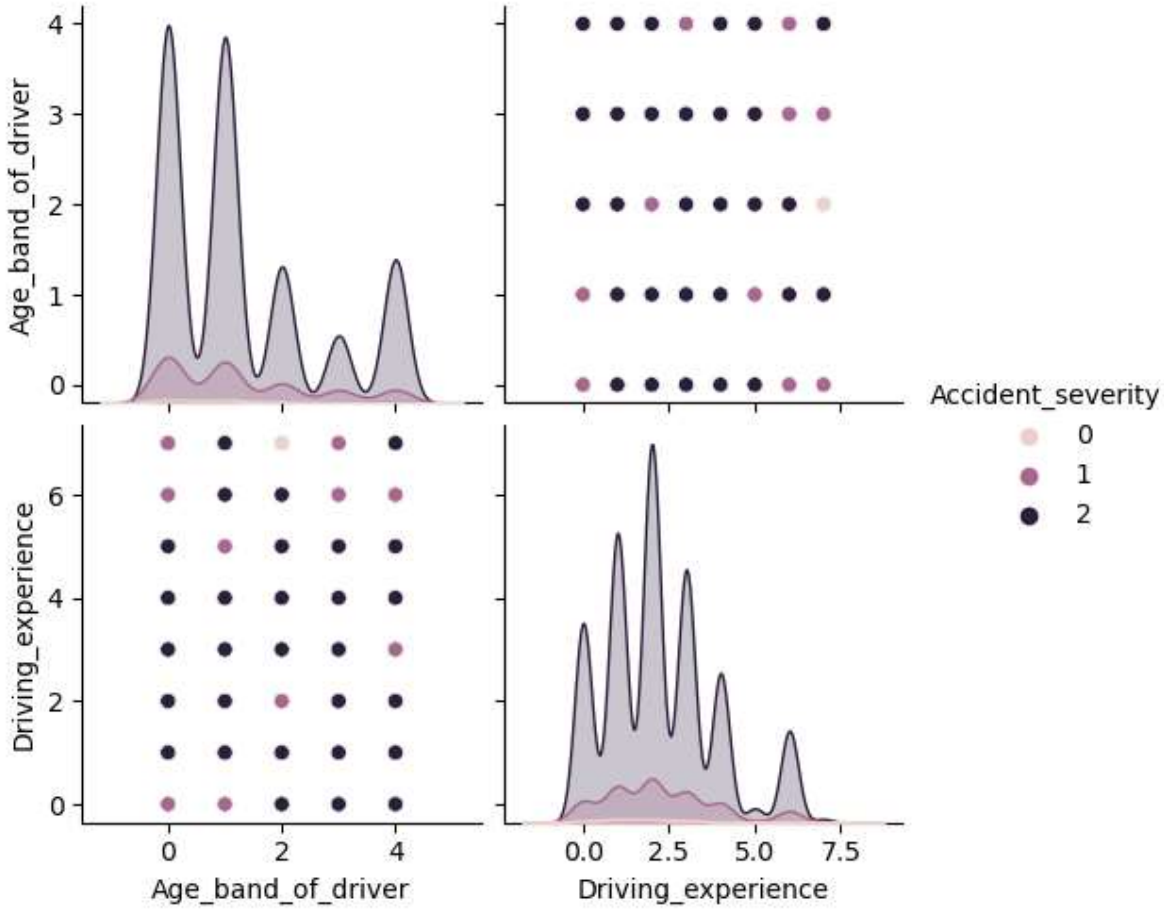
Out[13]: <BarContainer object of 12316 artists>



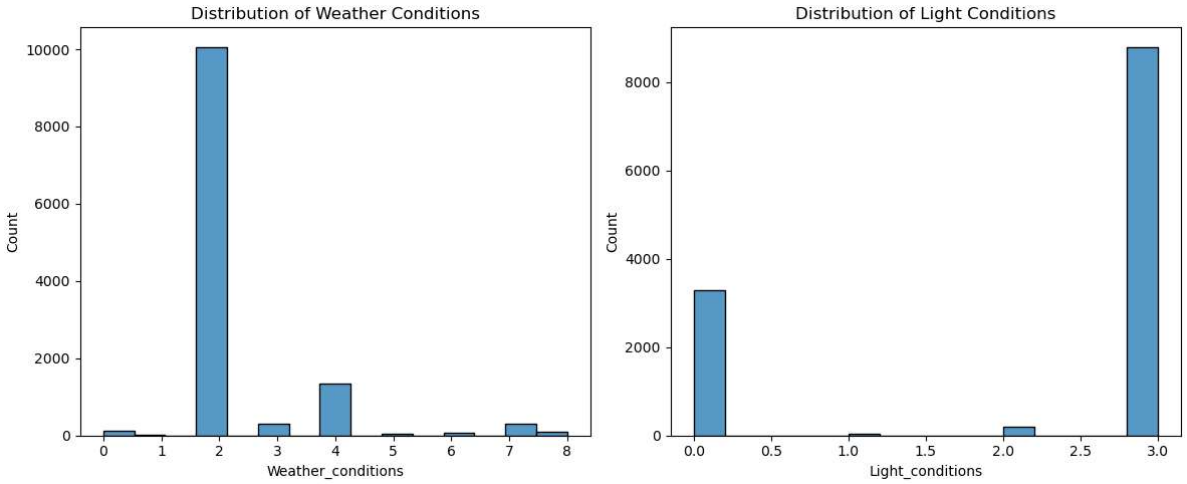
In [14]:
```
1 sns.pairplot(df,hue='Accident_severity')
```

```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarni
ng: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[14]: <seaborn.axisgrid.PairGrid at 0x23d67bec590>

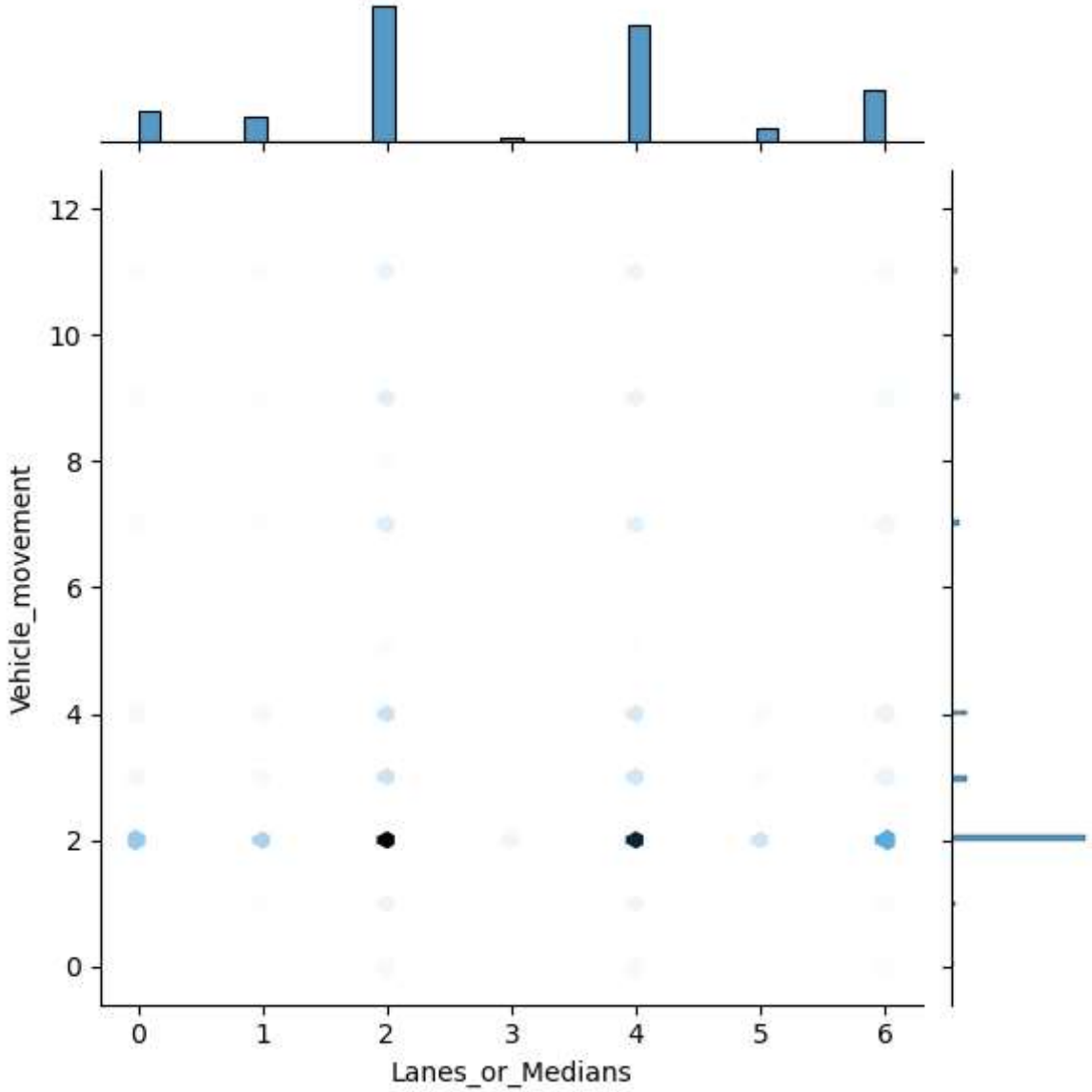


In [15]:
```
df['Weather_conditions']=le.fit_transform(df['Weather_conditions'])
df['Light_conditions']=le.fit_transform(df['Light_conditions'])
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 5))
sns.histplot(df['Weather_conditions'], ax=ax1)
ax1.set_title('Distribution of Weather Conditions')
sns.histplot(df['Light_conditions'], ax=ax2)
ax2.set_title('Distribution of Light Conditions')
plt.tight_layout()
plt.show()
```



In [16]:
```
df['Lanes_or_Medians']=le.fit_transform(df['Lanes_or_Medians'])
df['Vehicle_movement']=le.fit_transform(df['Vehicle_movement'])
sns.jointplot(x='Lanes_or_Medians',y='Vehicle_movement',data=df,kind='hex'
```

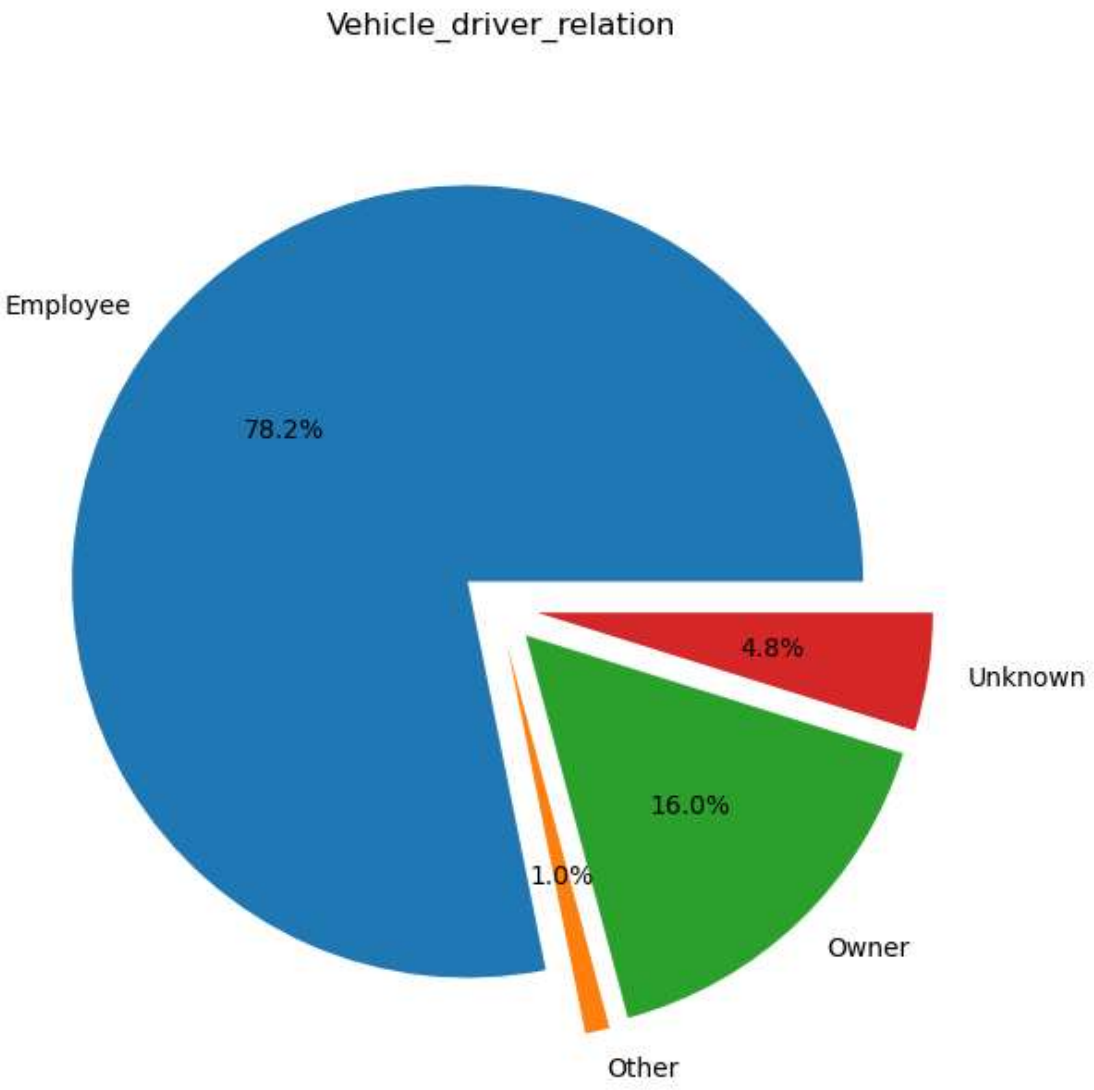
Out[16]: <seaborn.axisgrid.JointGrid at 0x23d6ff89810>



Here you can check any columns value distribution in percentage by putting column name in input

In [73]:
```
columns=input()
unique_values, counts = np.unique(df[columns], return_counts=True)
percentages = counts / counts.sum() * 100
plt.figure(figsize=(8, 8))
explode = [0.1] * len(counts)
plt.pie(x=counts, labels=unique_values, autopct="%1.1f%%", explode=explode
plt.title((columns), y=1.1)
plt.axis("equal")
plt.xticks(rotation=140)
plt.gca().axis("equal")
plt.subplots_adjust(bottom=0.25)
plt.show()
```

```
Vehicle_driver_relation
```



In [69]:

```
columns=(input('Pie chart 1 :'))
label1,counts1=np.unique(df[columns],return_counts=True)
percentage=counts1/counts1.sum()*100
columns=(input('Pie chart 2 :'))
label2,counts2=np.unique(df[columns],return_counts=True)
percentage=counts2/counts2.sum()*100
columns=(input('Pie chart 3 :'))
label3,counts3=np.unique(df[columns],return_counts=True)
percentage=counts3/counts3.sum()*100
columns=(input('Pie chart 4 :'))
label4,counts4=np.unique(df[columns],return_counts=True)
percentage=counts4/counts4.sum()*100
explode1 = [0.1] * len(counts1)
explode2 = [0.1] * len(counts2)
explode3 = [0.1] * len(counts3)
explode4 = [0.1] * len(counts4)
fig, (ax1, ax2, ax3, ax4) = plt.subplots(1, 4, figsize=(12, 5))
ax1.pie(x=counts1, labels=label1, autopct="%1.1f%%", explode=explode1)
ax2.pie(x=counts2, labels=label2, autopct="%1.1f%%", explode=explode2)
ax3.pie(x=counts3, labels=label3, autopct="%1.1f%%", explode=explode3)
ax4.pie(x=counts4, labels=label4, autopct="%1.1f%%", explode=explode4)
plt.tight_layout()
plt.show()
```

```
Pie chart 1 :Gender
Pie chart 2 :Lanes_or_Medians
Pie chart 3 :Vehicle_movement
Pie chart 4 :Weather_conditions
```