Car Fraud Insurance

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

In [6]:
```
insurance=pd.read_csv("interview_final.csv")
```

In [61]:
```
insurance.head()
```

Out[61]:

	months_as_customer	age	policy_number	policy_bind_date	policy_state	policy_csl	policy_dedu
0	328	48	521585	2014/10/17	OH	250/500	
1	228	42	342868	2006/6/27	IN	250/500	
2	134	29	687698	2000/6/9	OH	100/300	
3	256	41	227811	1990/5/25	IL	250/500	
4	228	44	367455	2014/6/6	IL	500/1000	

5 rows × 43 columns

In [18]:
```
insurance.info()
```

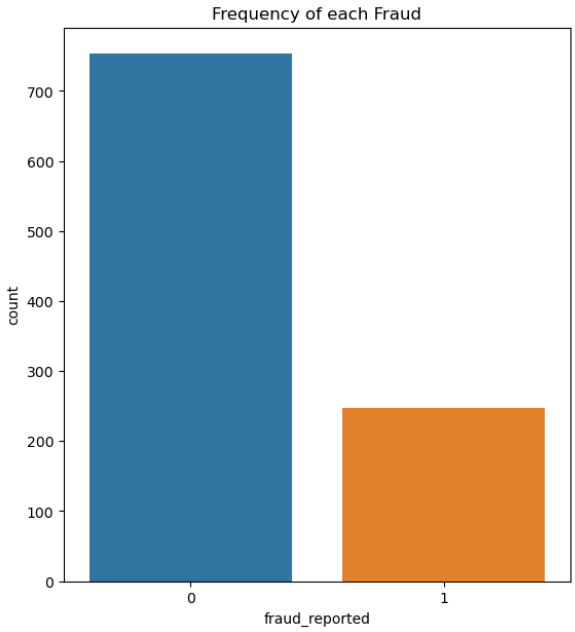
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 43 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   months_as_customer           1000 non-null   int64  
 1   age                          1000 non-null   int64  
 2   policy_number                1000 non-null   int64  
 3   policy_bind_date             1000 non-null   object 
 4   policy_state                 1000 non-null   object 
 5   policy_csl                   1000 non-null   object 
 6   policy_deductable            1000 non-null   int64  
 7   policy_annual_premium        1000 non-null   float64
 8   umbrella_limit               1000 non-null   int64  
 9   insured_zip                  1000 non-null   int64  
 10  insured_sex                  1000 non-null   object 
 11  insured_education_level      1000 non-null   object 
 12  insured_occupation           1000 non-null   object 
 13  insured_hobbies              1000 non-null   object 
 14  insured_relationship         1000 non-null   object 
 15  capital_gains                1000 non-null   int64  
 16  capital_loss                 1000 non-null   int64  
 17  incident_date                1000 non-null   object 
 18  incident_type                1000 non-null   object 
 19  collision_type               1000 non-null   object 
 20  incident_severity            1000 non-null   object 
 21  authorities_contacted        1000 non-null   object 
 22  incident_state               1000 non-null   object 
 23  incident_city                1000 non-null   object 
 24  incident_location            1000 non-null   object 
 25  incident_hour_of_the_day     1000 non-null   int64  
 26  number_of_vehicles_involved  1000 non-null   int64  
 27  property_damage              1000 non-null   int64  
 28  bodily_injuries              1000 non-null   int64  
 29  witnesses                    1000 non-null   int64  
 30  police_report_available      1000 non-null   int64  
 31  total_claim_amount           1000 non-null   int64  
 32  injury_claim                 1000 non-null   int64  
 33  property_claim               1000 non-null   int64  
 34  vehicle_claim                1000 non-null   int64  
 35  auto_make                    1000 non-null   object 
 36  auto_model                   1000 non-null   object 
 37  auto_year                    1000 non-null   int64  
 38  fraud_reported               1000 non-null   int64  
 39  age_group                    1000 non-null   object 
 40  bmi                          1000 non-null   float64
 41  bloodpressure                1000 non-null   int64  
 42  smoker                       1000 non-null   int64  
dtypes: float64(2), int64(22), object(19)
memory usage: 336.1+ KB
```

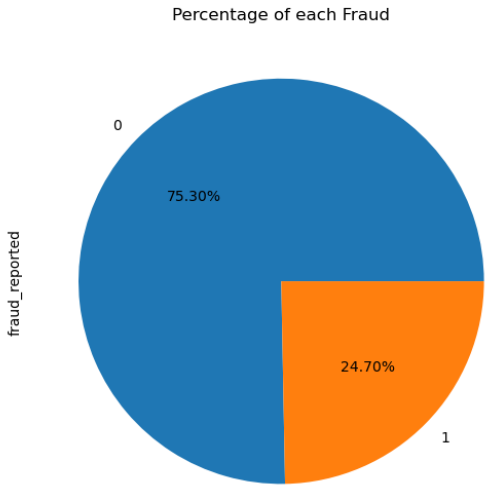
Descriptive analysis

In [9]:

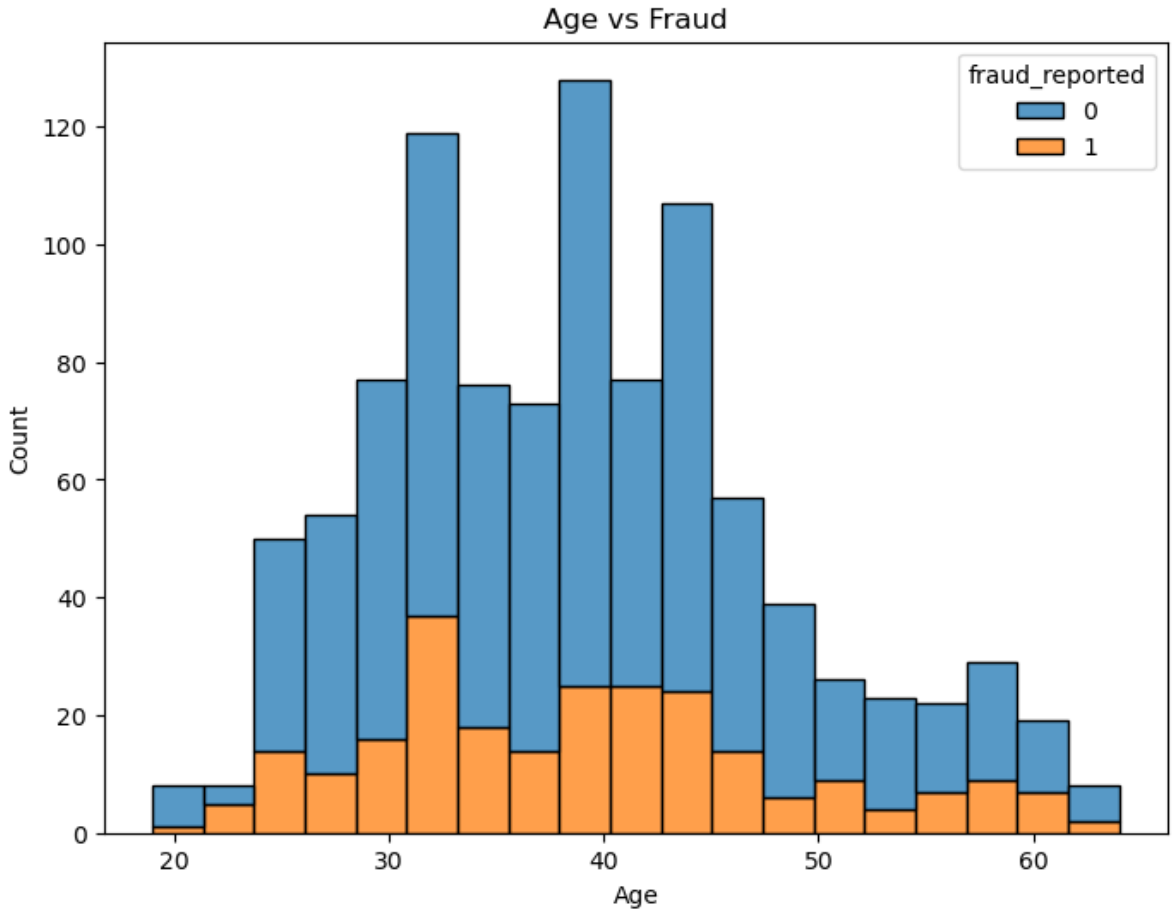
```
fig,axs = plt.subplots(1,2,figsize=(14,7))
sns.countplot(x='fraud_reported',data=insurance,ax=axs[0])
axs[0].set_title("Frequency of each Fraud")

insurance["fraud_reported"].value_counts().plot(x=None,y=None,kind="pie",ax=axs[1]
axs[1].set_title("Percentage of each Fraud")
plt.show()
```

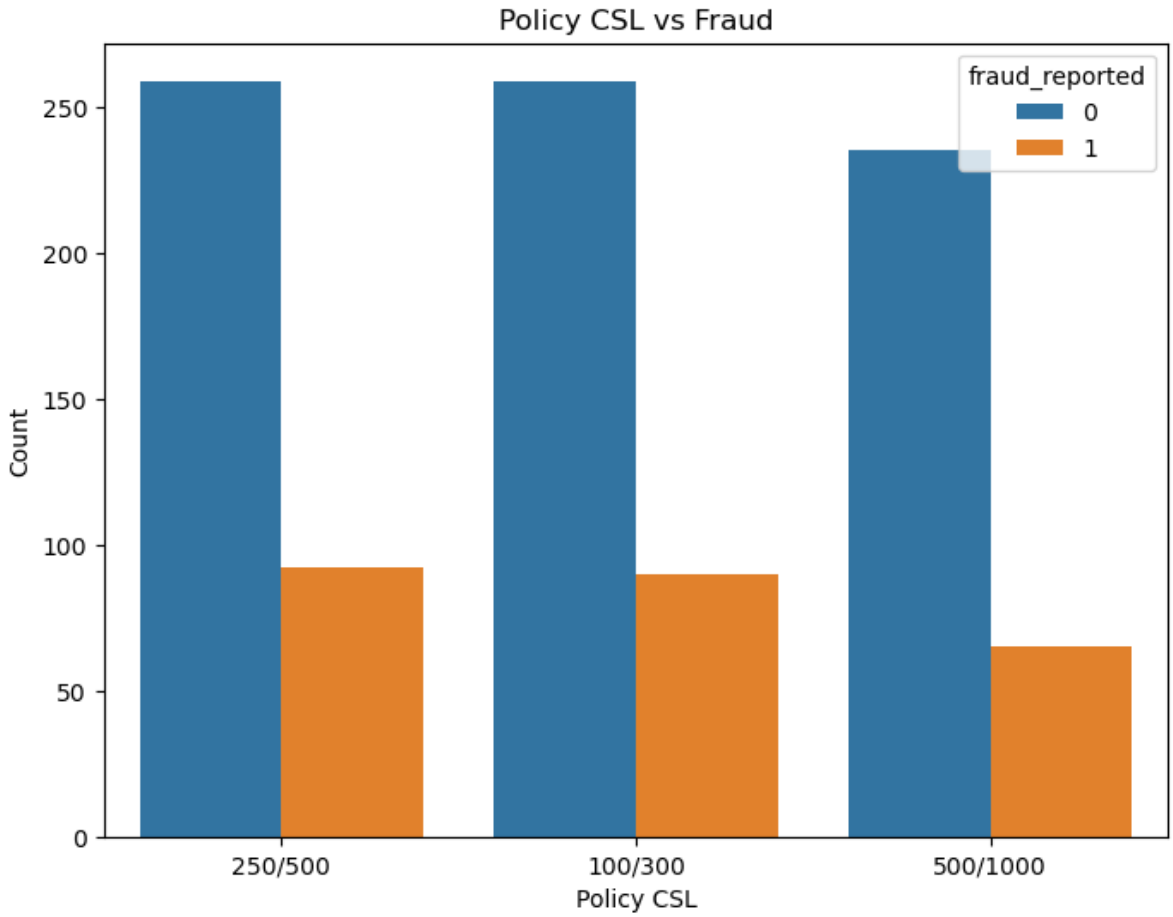




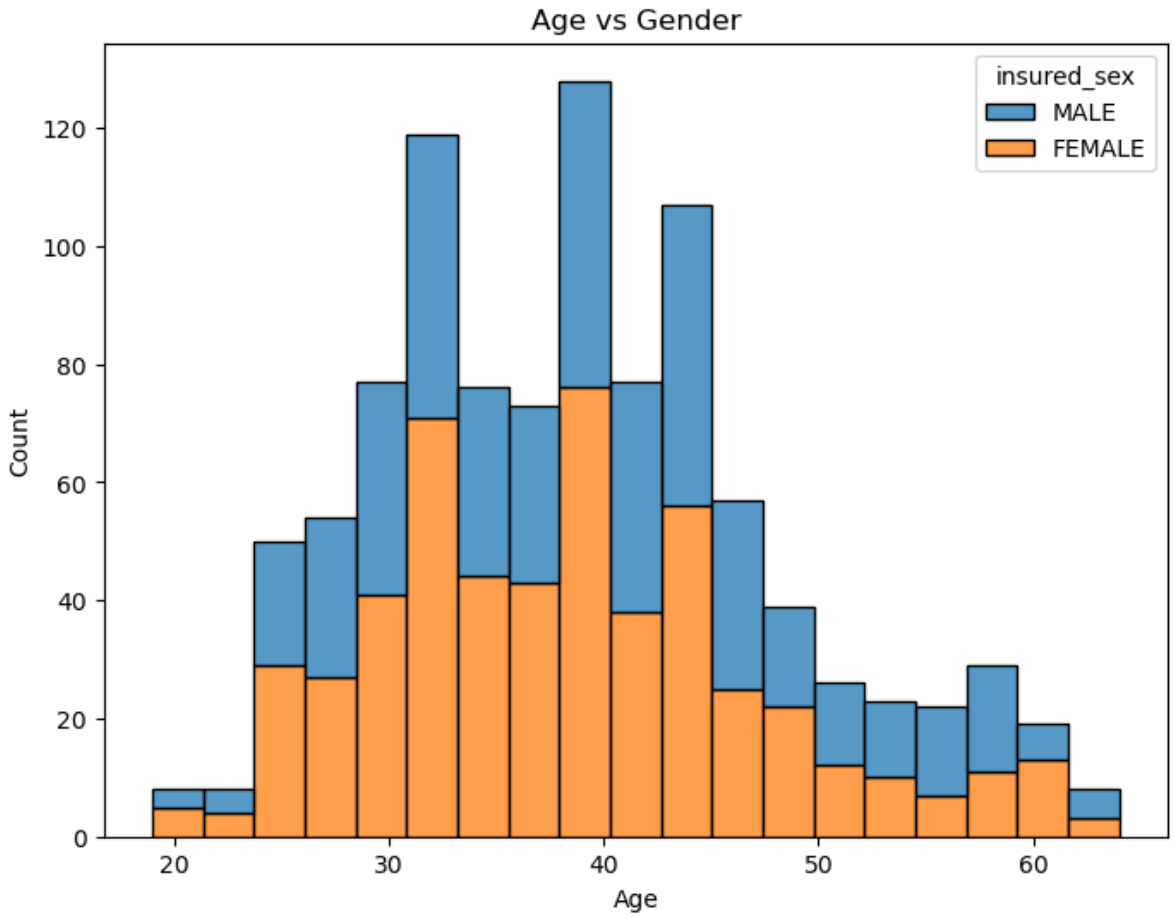
In [11]:
```
plt.figure(figsize=(8,6))
sns.histplot(x='age',hue='fraud_reported',data=insurance,multiple='stack')
plt.xlabel('Age')
plt.ylabel('Count')
plt.title('Age vs Fraud')
plt.show()
```



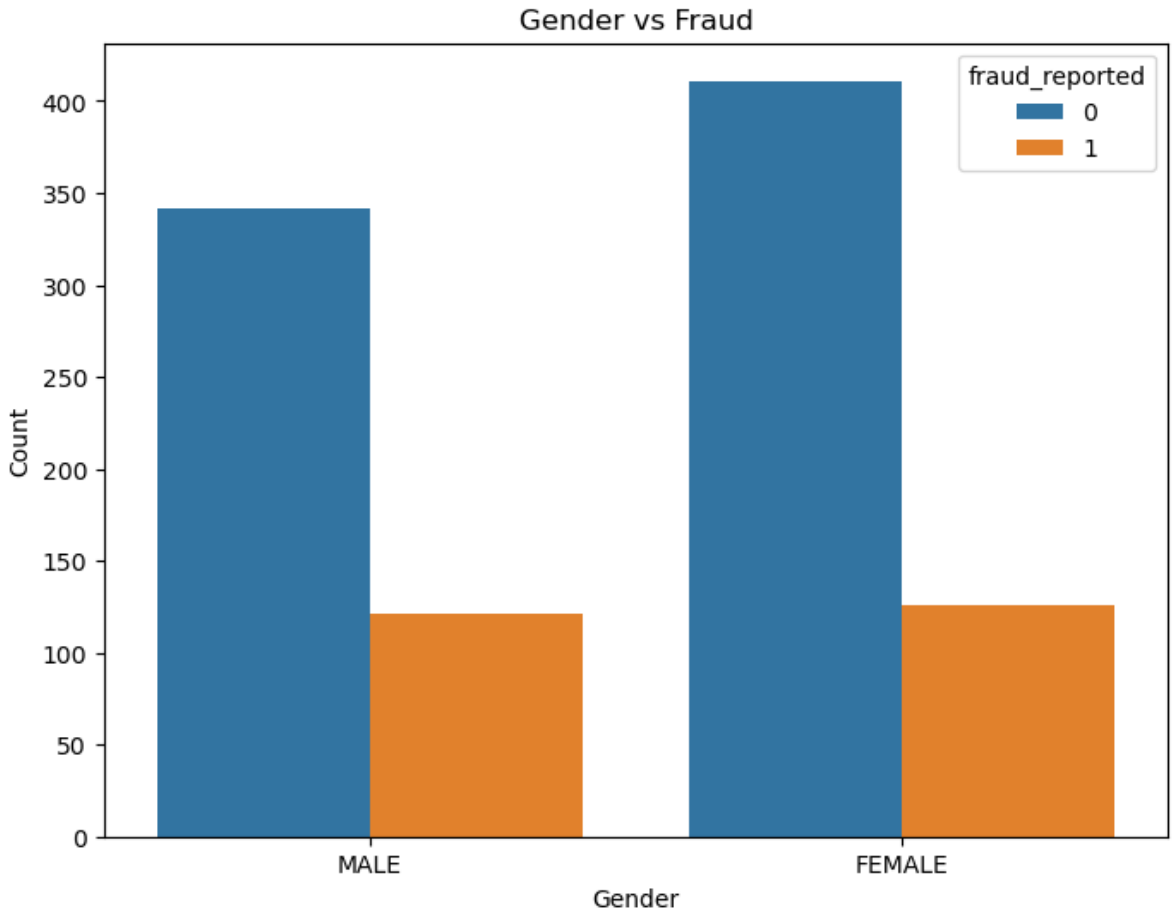
In [12]:
```
plt.figure(figsize=(8,6))
sns.countplot(x='policy_csl',hue='fraud_reported',data=insurance)
plt.xlabel('Policy CSL')
plt.ylabel('Count')
plt.title('Policy CSL vs Fraud')
plt.show()
```



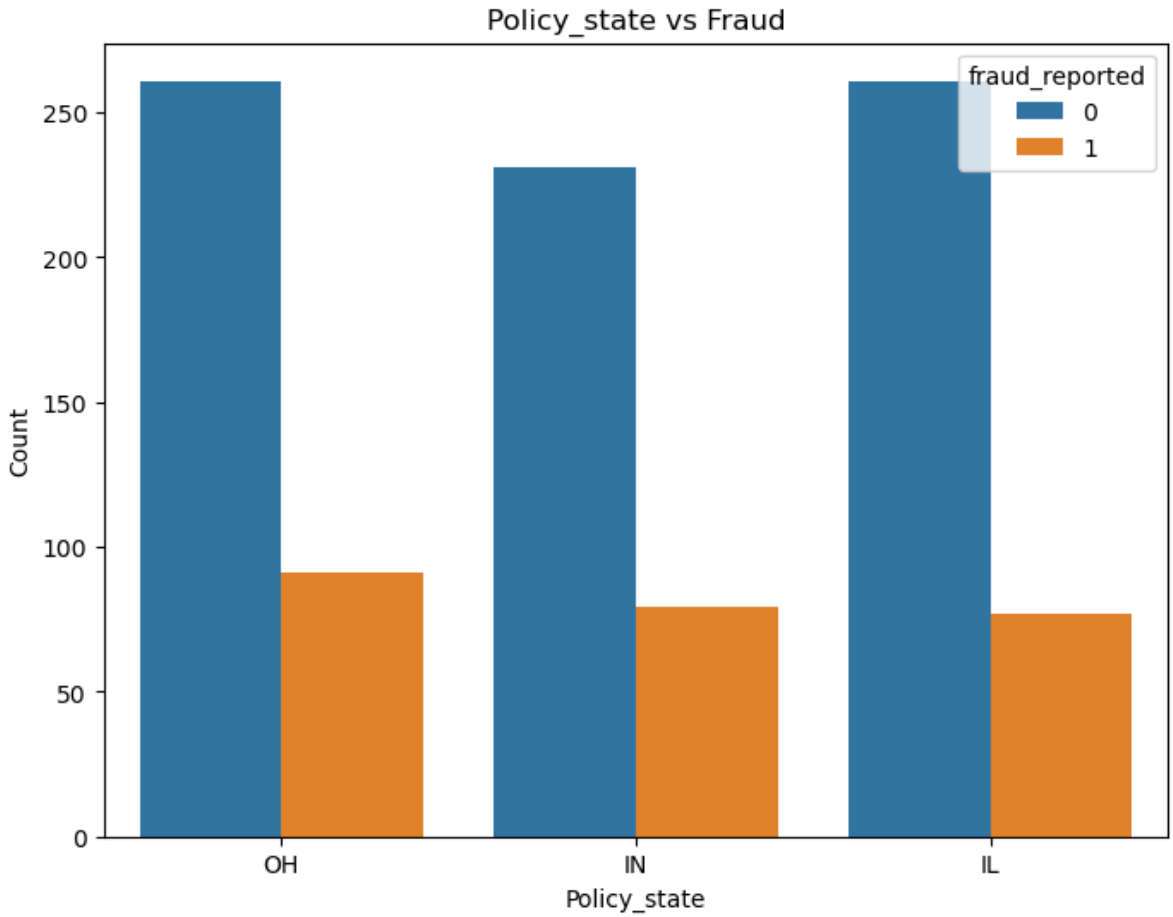
In [13]:
```
plt.figure(figsize=(8,6))
sns.histplot(x='age',hue='insured_sex',data=insurance,multiple='stack')
plt.xlabel('Age')
plt.ylabel('Count')
plt.title('Age vs Gender')
plt.show()
```



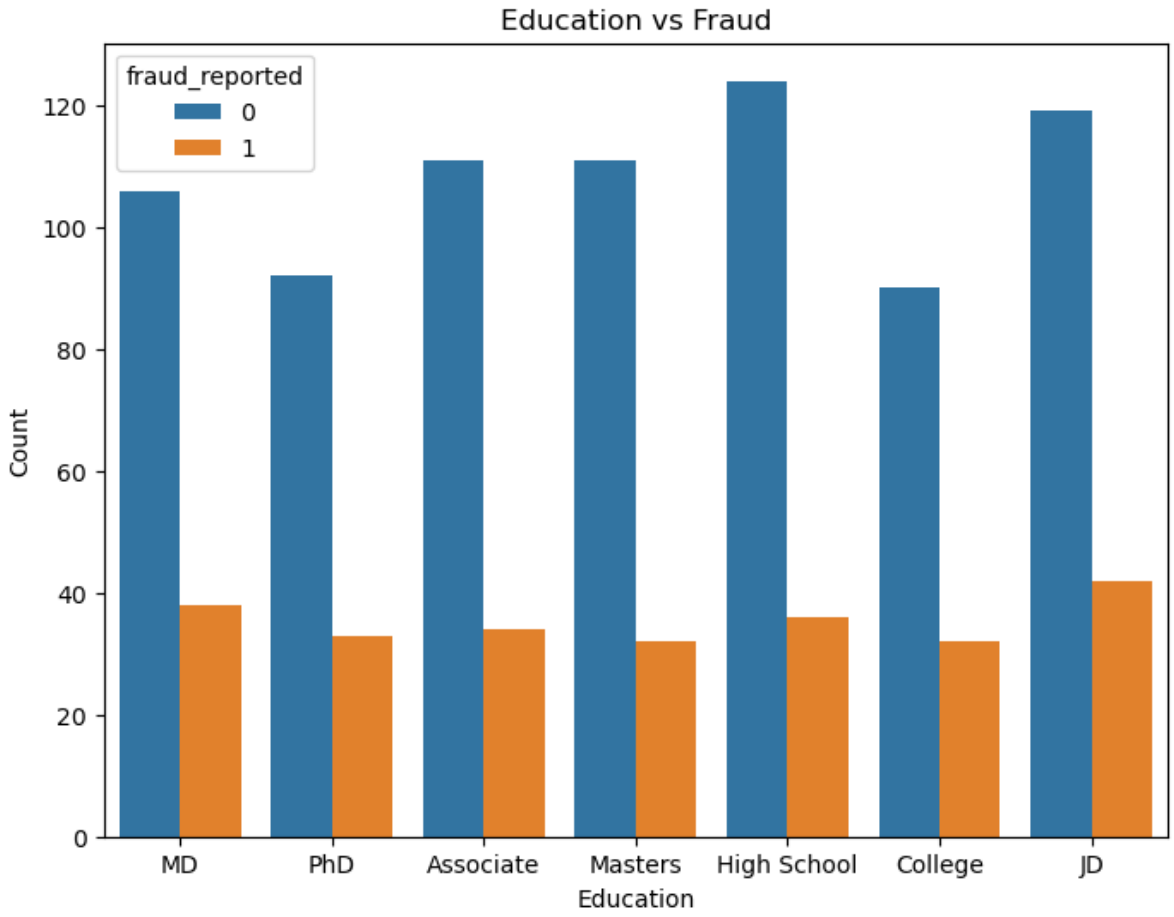
In [14]:
```
plt.figure(figsize=(8,6))
sns.countplot(x='insured_sex',hue='fraud_reported',data=insurance)
plt.xlabel('Gender')
plt.ylabel('Count')
plt.title('Gender vs Fraud')
plt.show()
```



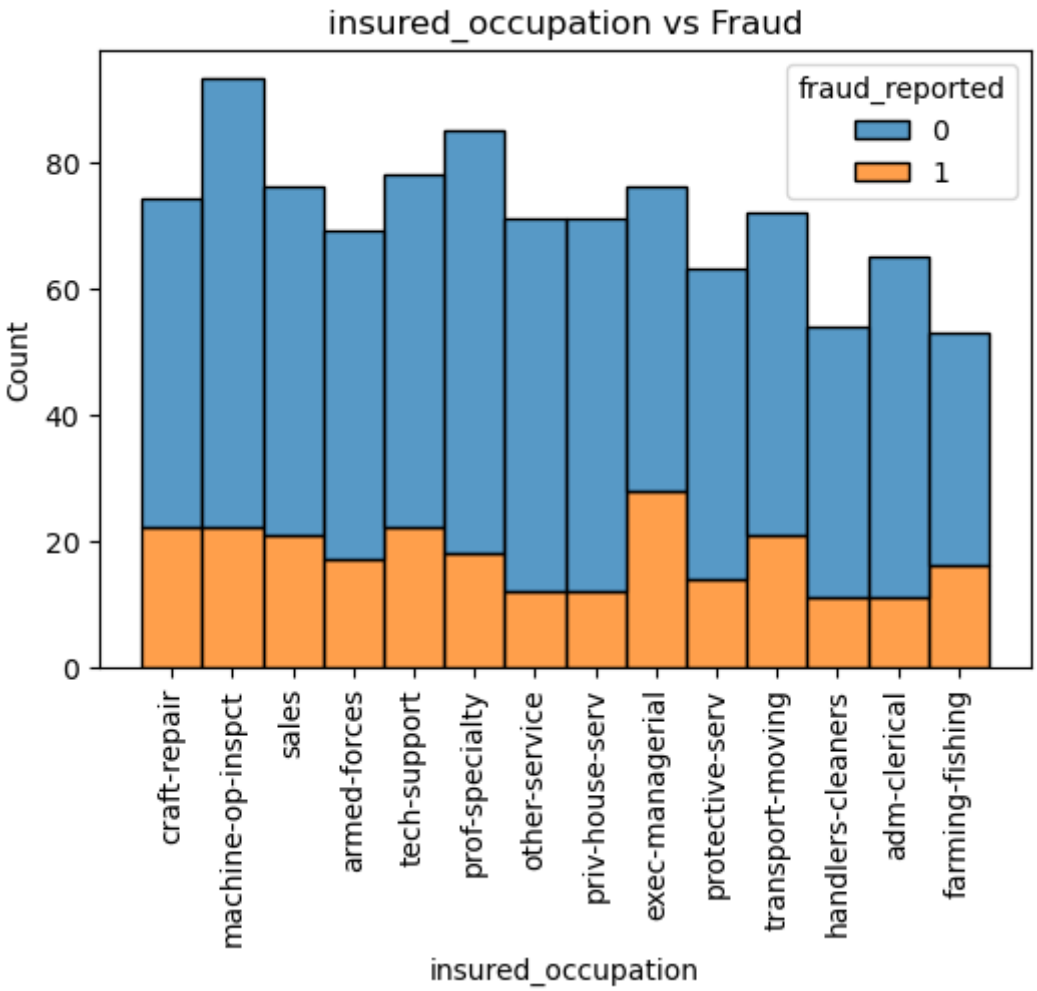
In [15]:
```
plt.figure(figsize=(8,6))
sns.countplot(x='policy_state',hue='fraud_reported',data=insurance)
plt.xlabel('Policy_state')
plt.ylabel('Count')
plt.title('Policy_state vs Fraud')
plt.show()
```



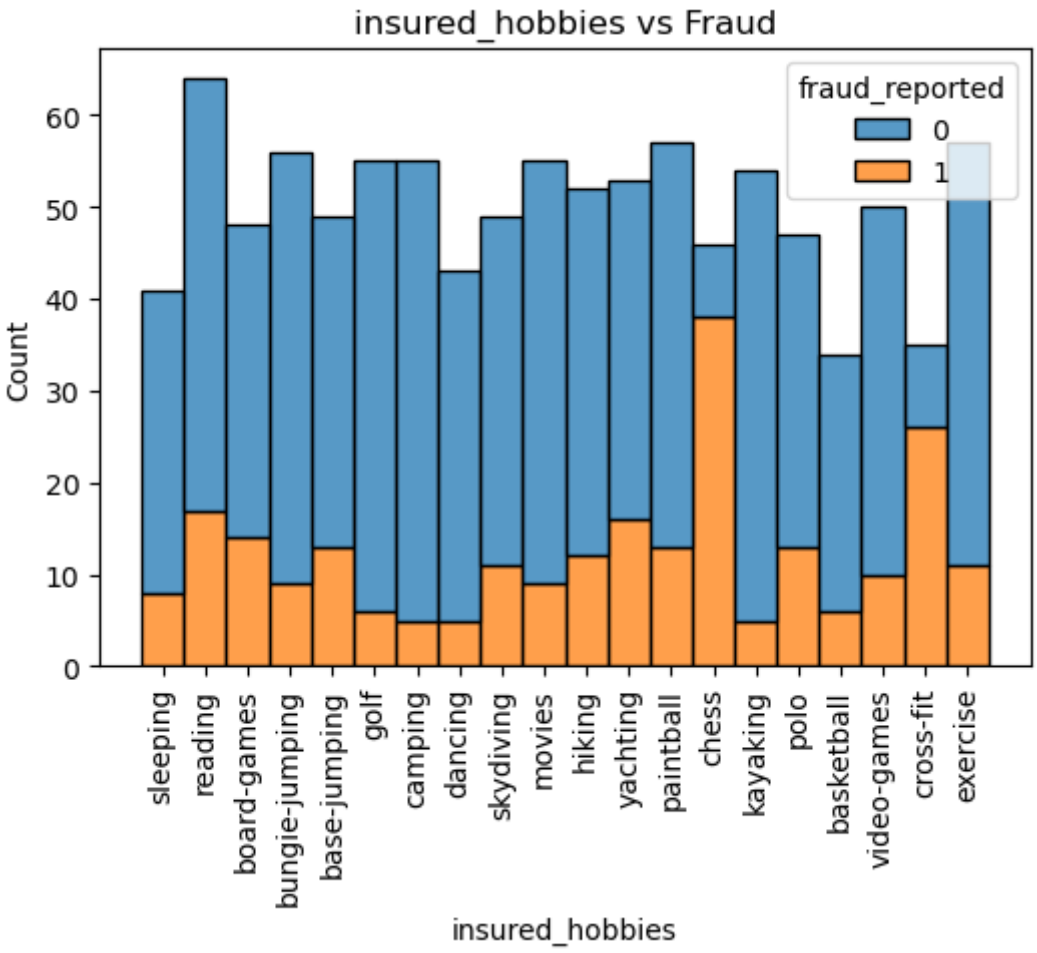
In [16]:
```
plt.figure(figsize=(8,6))
sns.countplot(x='insured_education_level',hue='fraud_reported',data=insurance)
plt.xlabel('Education')
plt.ylabel('Count')
plt.title('Education vs Fraud')
plt.show()
```



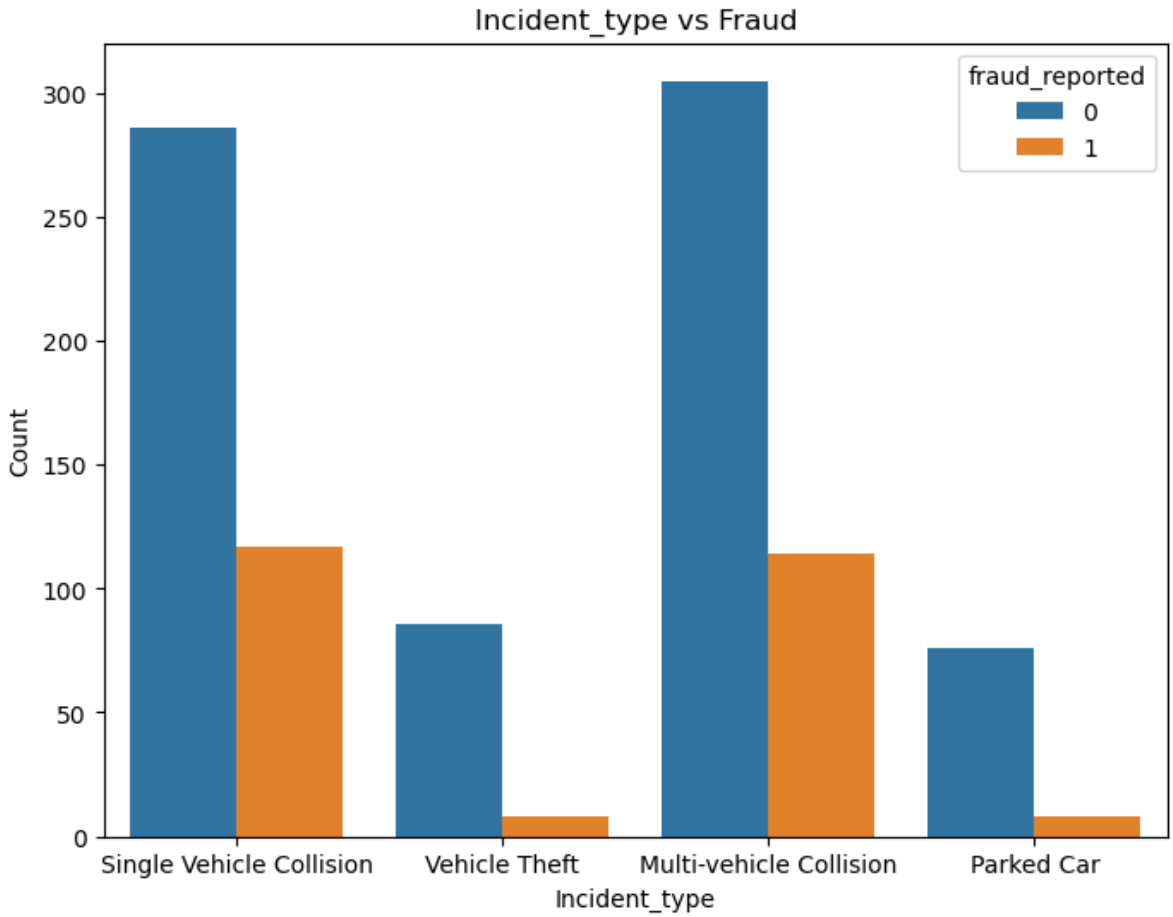
In [9]:
```
plt.figure(figsize=(6,4))
sns.histplot(x='insured_occupation',hue='fraud_reported',data=insurance,multiple='
plt.xlabel('insured_occupation')
plt.xticks(rotation=90)
plt.ylabel('Count')
plt.title('insured_occupation vs Fraud')
plt.show()
```



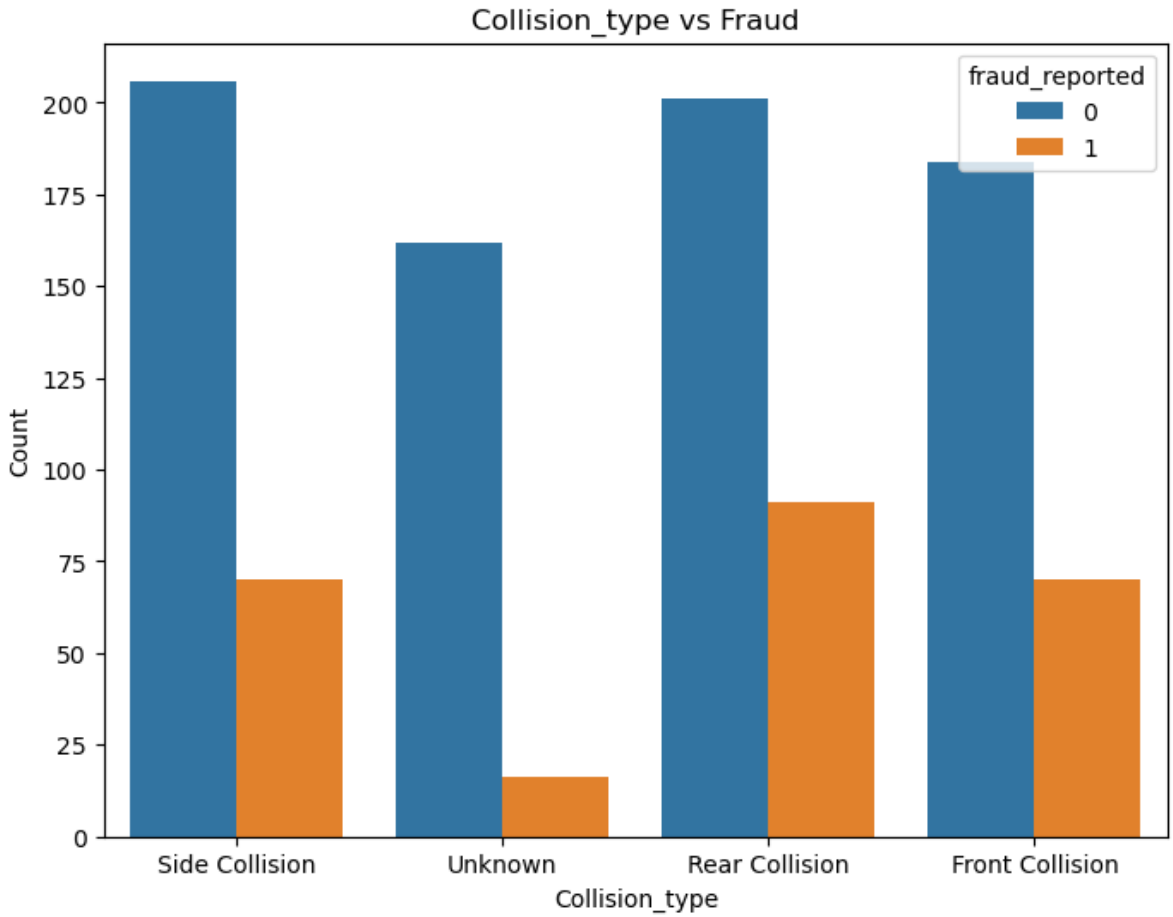
In [10]:
```
plt.figure(figsize=(6,4))
sns.histplot(x='insured_hobbies',hue='fraud_reported',data=insurance,multiple='sta
plt.xlabel('insured_hobbies')
plt.xticks(rotation=90)
plt.ylabel('Count')
plt.title('insured_hobbies vs Fraud')
plt.show()
```



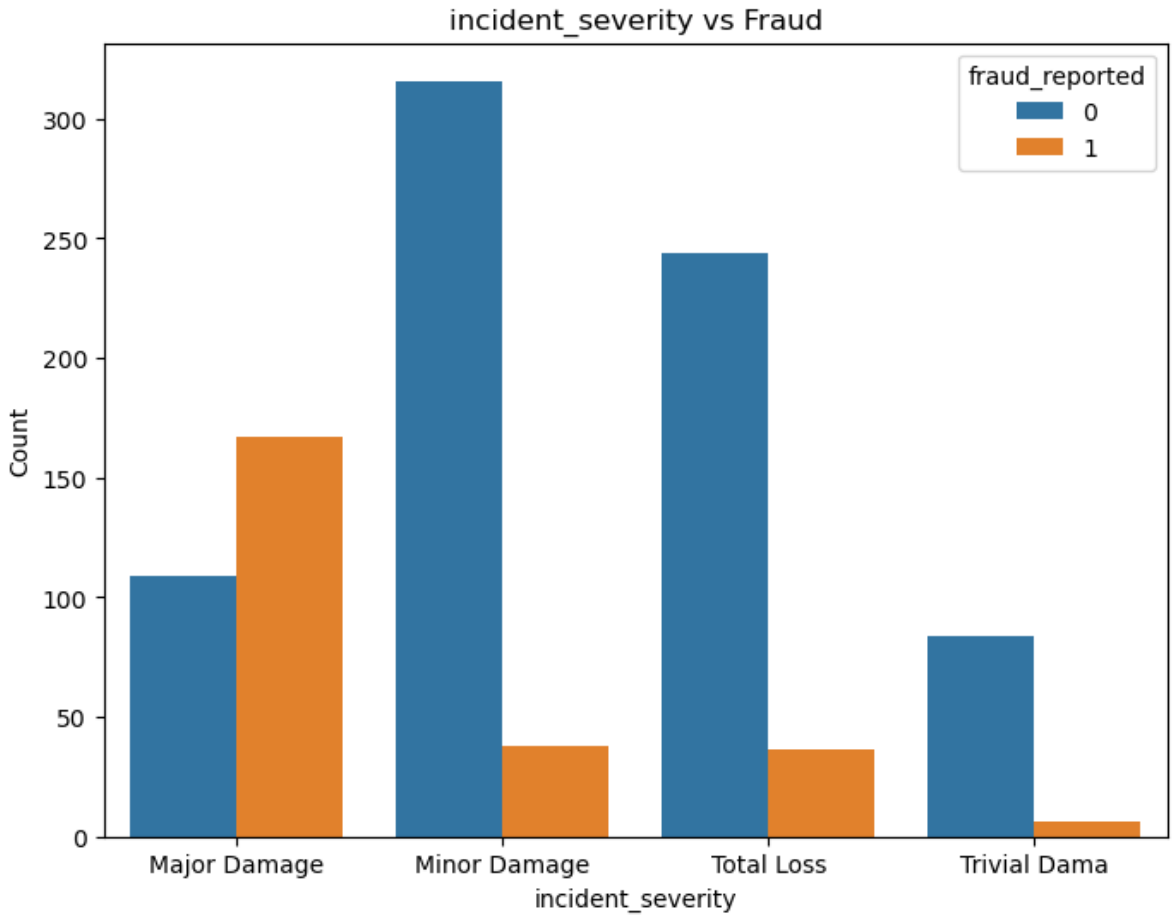
In [25]:
```
plt.figure(figsize=(8,6))
sns.countplot(x='incident_type',hue='fraud_reported',data=insurance)
plt.xlabel('Incident_type')
plt.ylabel('Count')
plt.title('Incident_type vs Fraud')
plt.show()
```



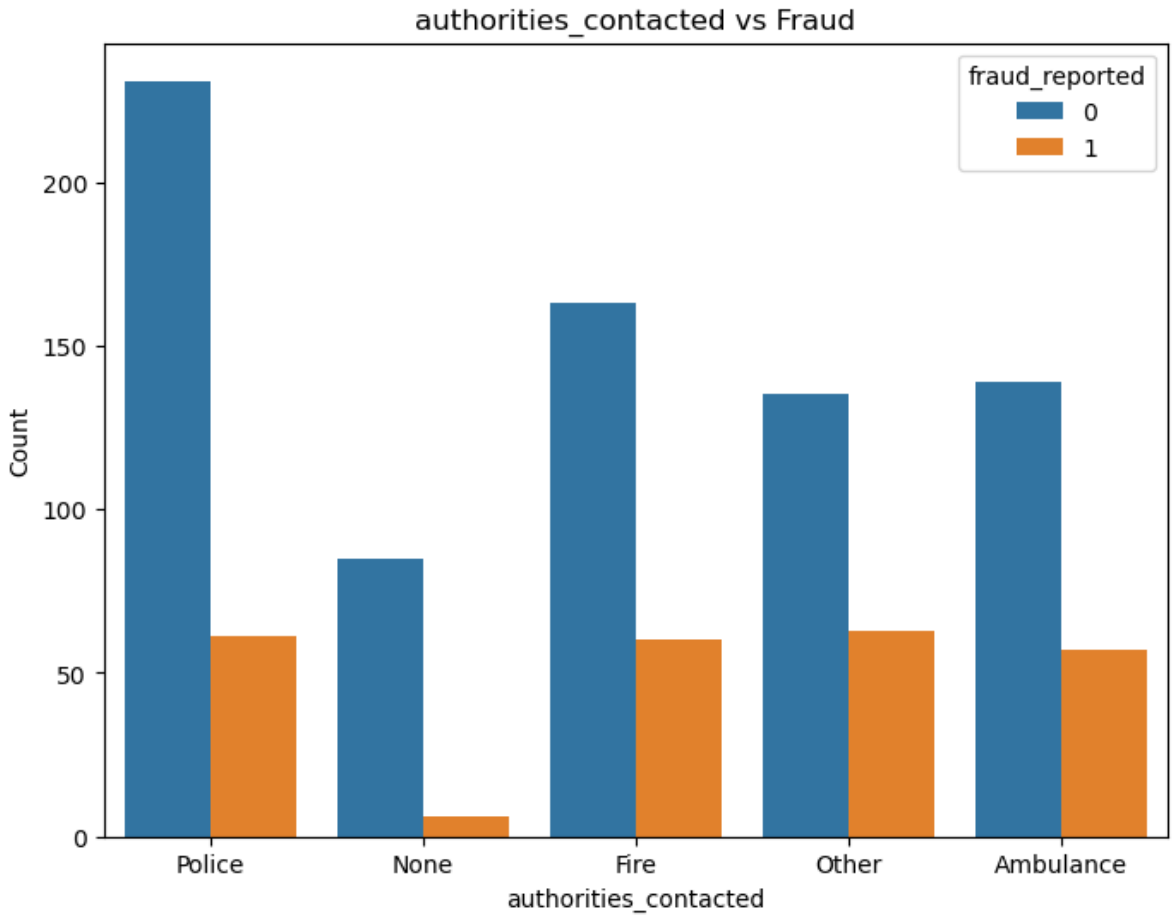
In [26]:
```
plt.figure(figsize=(8,6))
sns.countplot(x='collision_type',hue='fraud_reported',data=insurance)
plt.xlabel('Collision_type')
plt.ylabel('Count')
plt.title('Collision_type vs Fraud')
plt.show()
```



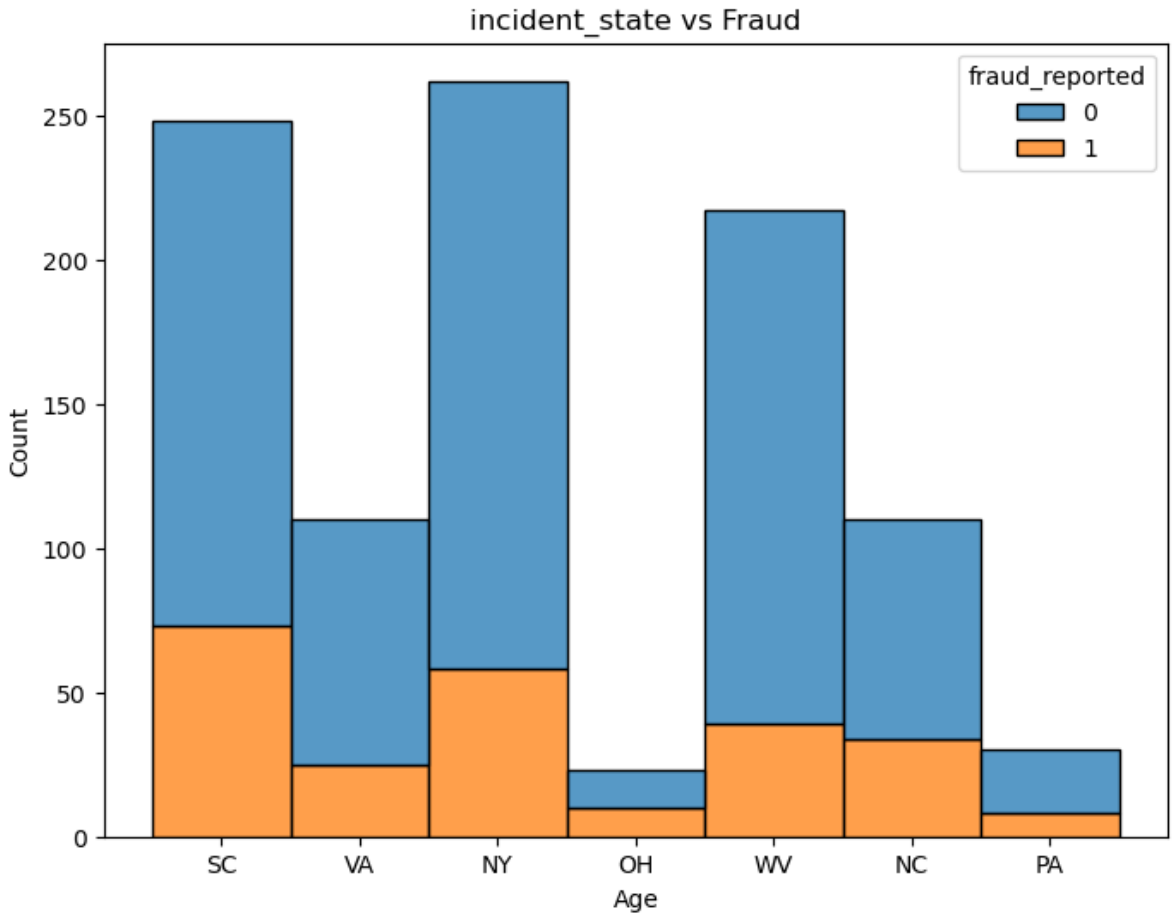
In [3]:
```
plt.figure(figsize=(8,6))
sns.countplot(x='incident_severity',hue='fraud_reported',data=insurance)
plt.xlabel('incident_severity')
plt.ylabel('Count')
plt.title('incident_severity vs Fraud')
plt.show()
```



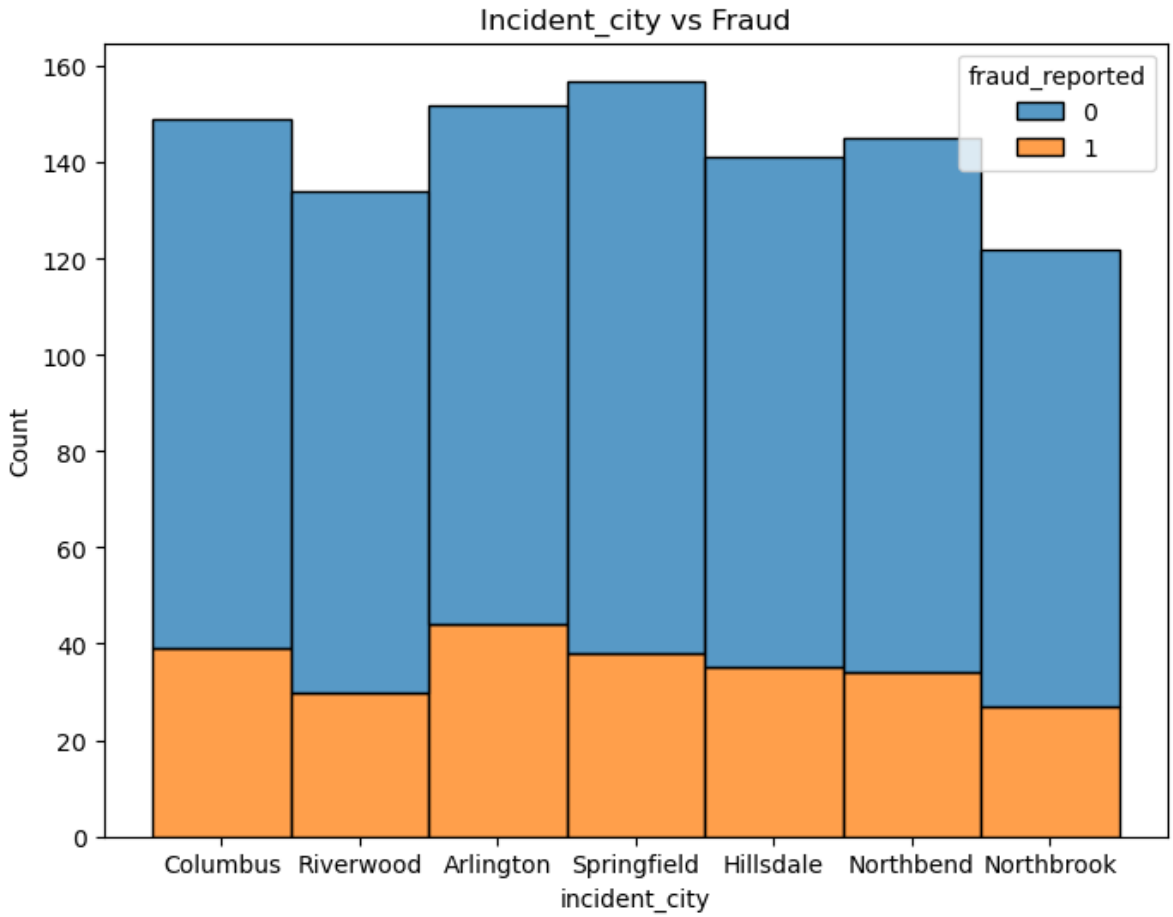
In [4]:
```
plt.figure(figsize=(8,6))
sns.countplot(x='authorities_contacted',hue='fraud_reported',data=insurance)
plt.xlabel('authorities_contacted')
plt.ylabel('Count')
plt.title('authorities_contacted vs Fraud')
plt.show()
```



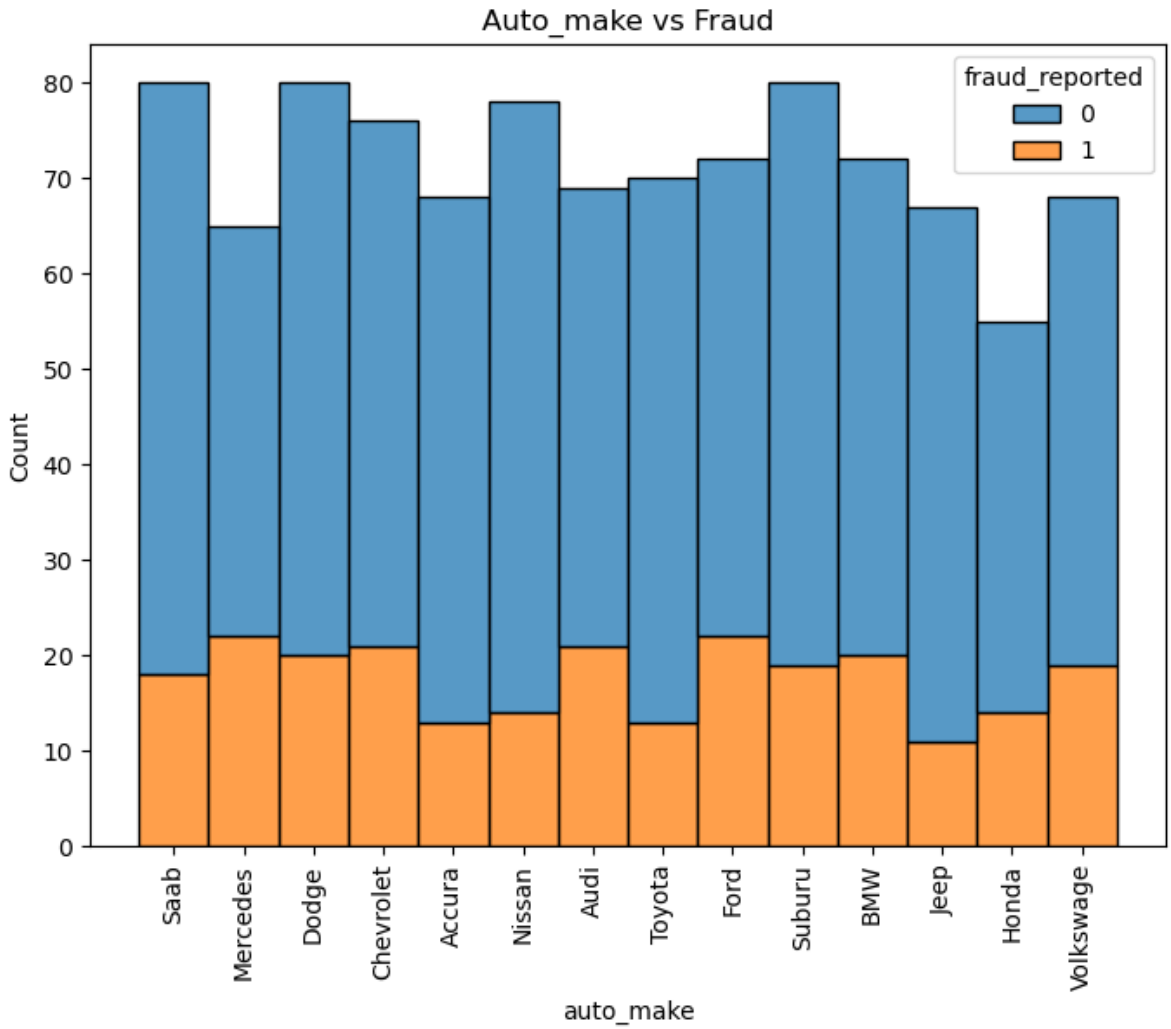
In [5]:
```
plt.figure(figsize=(8,6))
sns.histplot(x='incident_state',hue='fraud_reported',data=insurance,multiple='stacl
plt.xlabel('Age')
plt.ylabel('Count')
plt.title('incident_state vs Fraud')
plt.show()
```



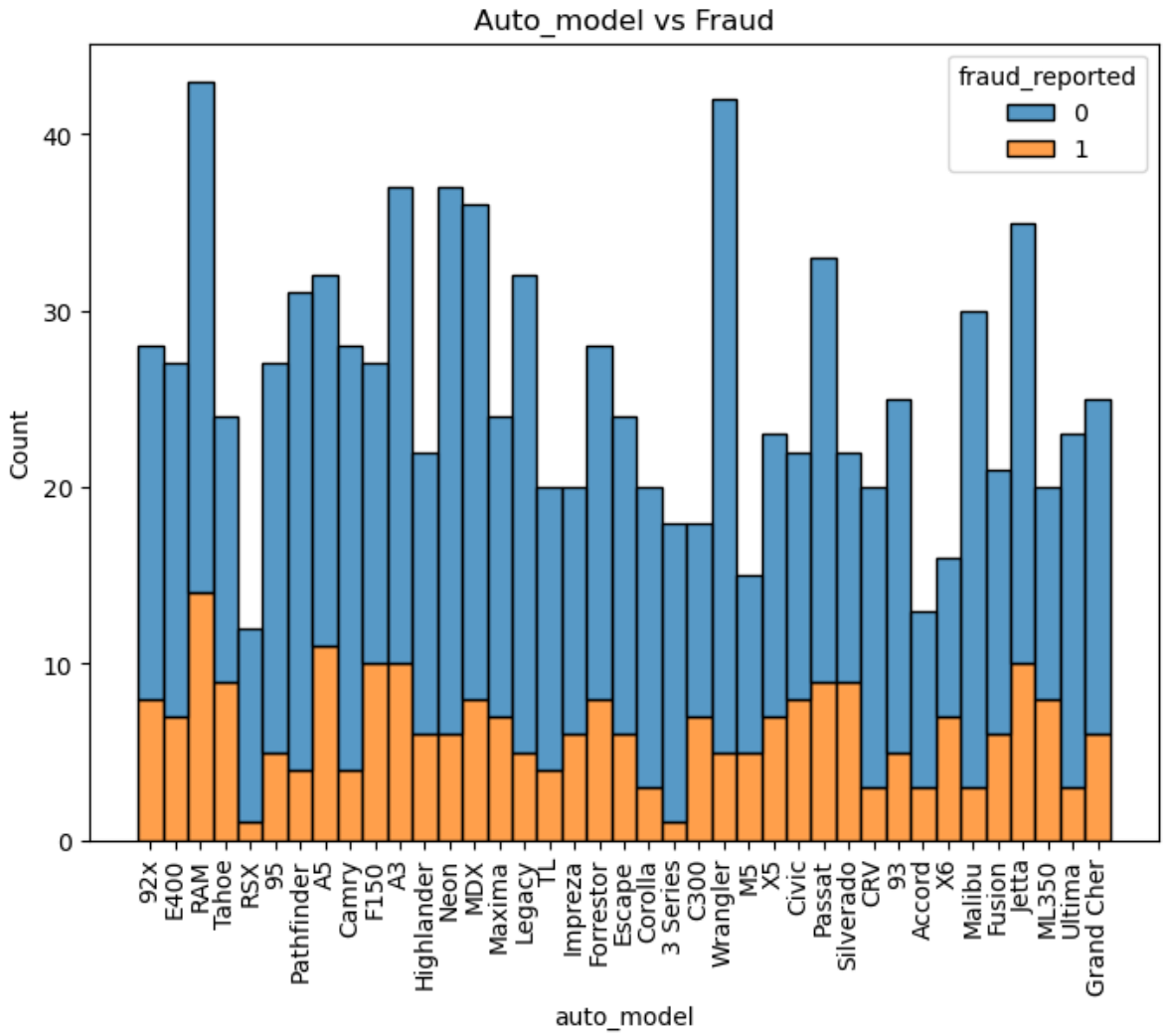
In [13]:
```
plt.figure(figsize=(8,6))
sns.histplot(x='incident_city',hue='fraud_reported',data=insurance,multiple='stack
plt.xlabel('incident_city')
plt.ylabel('Count')
plt.title('Incident_city vs Fraud')
plt.show()
```



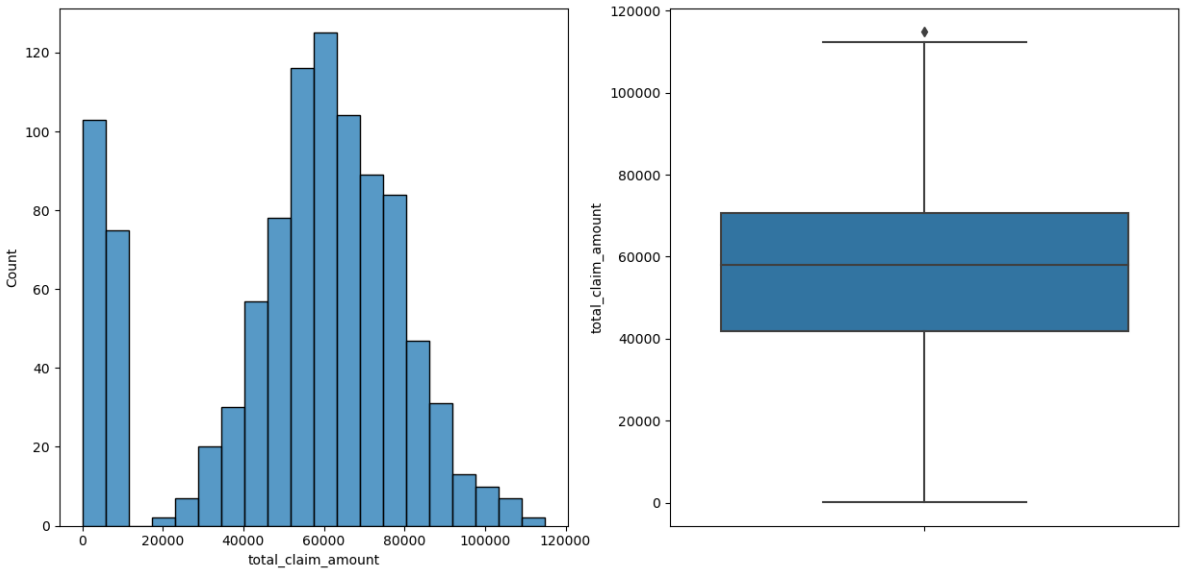
In [15]:
```
plt.figure(figsize=(8,6))
sns.histplot(x='auto_make',hue='fraud_reported',data=insurance,multiple='stack')
plt.xlabel('auto_make')
plt.xticks(rotation=90)
plt.ylabel('Count')
plt.title('Auto_make vs Fraud')
plt.show()
```



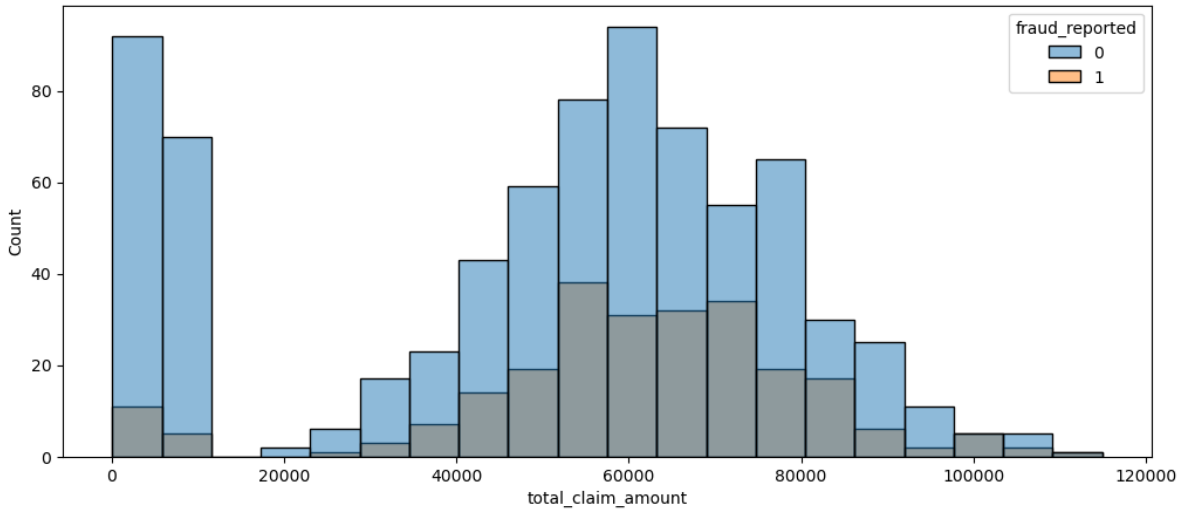
In [16]:
```
plt.figure(figsize=(8,6))
sns.histplot(x='auto_model',hue='fraud_reported',data=insurance,multiple='stack')
plt.xlabel('auto_model')
plt.xticks(rotation=90)
plt.ylabel('Count')
plt.title('Auto_model vs Fraud')
plt.show()
```



In [19]:
```
figure,axes = plt.subplots(1,2,figsize=[15,7])
sns.histplot(insurance,x='total_claim_amount', ax=axes[0])
sns.boxplot(insurance,y='total_claim_amount',ax=axes[1])
plt.show()
```



In [21]:
```
plt.subplots(1,1,figsize=[12,5])
sns.histplot(insurance,x='total_claim_amount',hue='fraud_reported')
plt.show()
```



Correlation analysis

In [17]:
```
correlation_matrix = insurance.corr()
plt.figure(figsize=(12,10))
sns.heatmap(correlation_matrix,vmax=0.9,linewidths=0.05,cmap="RdGy")
plt.show()
```