Insurance claims fraud Detection

Importing Libraries

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Loading the Dataset

In [2]:
```
df=pd.read_csv(r"https://raw.githubusercontent.com/dsrscientist/Data-Science-ML-Capstone-Projects/master/Automobile_insurance_fraud.csv")
df.head()
```

Out[2]:

	months_as_customer	age	policy_number	policy_bind_date	policy_state	policy_csl	policy_deductable	policy_annual_premium	umbrella_limit	insured_zip	...	police_report_available	total_claim_amount	injury_claim	property_claim	vehicle_claim	auto_make	auto_model	auto_year
0	328	48	521585	17-10-2014	OH	250/500	1000	1406.91	0	466132	...	YES	71610	6510	13020	52080	Saab	92x	2004
1	228	42	342868	27-06-2006	IN	250/500	2000	1197.22	5000000	468176	...	?	5070	780	780	3510	Mercedes	E400	2007
2	134	29	687698	06-09-2000	OH	100/300	2000	1413.14	5000000	430632	...	NO	34650	7700	3850	23100	Dodge	RAM	2007
3	256	41	227811	25-05-1990	IL	250/500	2000	1415.74	6000000	608117	...	NO	63400	6340	6340	50720	Chevrolet	Tahoe	2014
4	228	44	367455	06-06-2014	IL	500/1000	1000	1583.91	6000000	610706	...	NO	6500	1300	650	4550	Accura	RSX	2009

5 rows × 40 columns

In [3]:
```
df
```

Out[3]:

	months_as_customer	age	policy_number	policy_bind_date	policy_state	policy_csl	policy_deductable	policy_annual_premium	umbrella_limit	insured_zip	...	police_report_available	total_claim_amount	injury_claim	property_claim	vehicle_claim	auto_make	auto_model	auto_year
0	328	48	521585	17-10-2014	OH	250/500	1000	1406.91	0	466132	...	YES	71610	6510	13020	52080	Saab	92x	2004
1	228	42	342868	27-06-2006	IN	250/500	2000	1197.22	5000000	468176	...	?	5070	780	780	3510	Mercedes	E400	2007
2	134	29	687698	06-09-2000	OH	100/300	2000	1413.14	5000000	430632	...	NO	34650	7700	3850	23100	Dodge	RAM	2007
3	256	41	227811	25-05-1990	IL	250/500	2000	1415.74	6000000	608117	...	NO	63400	6340	6340	50720	Chevrolet	Tahoe	2014
4	228	44	367455	06-06-2014	IL	500/1000	1000	1583.91	6000000	610706	...	NO	6500	1300	650	4550	Accura	RSX	2009
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	3	38	941851	16-07-1991	OH	500/1000	1000	1310.80	0	431289	...	?	87200	17440	8720	61040	Honda	Accord	2006
996	285	41	186934	05-01-2014	IL	100/300	1000	1436.79	0	608177	...	?	108480	18080	18080	72320	Volkswagen	Passat	2015
997	130	34	918516	17-02-2003	OH	250/500	500	1383.49	3000000	442797	...	YES	67500	7500	7500	52500	Suburu	Impreza	1996
998	458	62	533940	18-11-2011	IL	500/1000	2000	1356.92	5000000	441714	...	YES	46980	5220	5220	36540	Audi	A5	1998
999	456	60	556080	11-11-1996	OH	250/500	1000	766.19	0	612260	...	?	5060	460	920	3680	Mercedes	E400	2007

1000 rows × 40 columns

In [5]:
```
df.head()  # let's take a look at the data
```

Out[5]:

	months_as_customer	age	policy_number	policy_bind_date	policy_state	policy_csl	policy_deductable	policy_annual_premium	umbrella_limit	insured_zip	...	police_report_available	total_claim_amount	injury_claim	property_claim	vehicle_claim	auto_make	auto_model	auto_year
0	328	48	521585	17-10-2014	OH	250/500	1000	1406.91	0	466132	...	YES	71610	6510	13020	52080	Saab	92x	2004
1	228	42	342868	27-06-2006	IN	250/500	2000	1197.22	5000000	468176	...	?	5070	780	780	3510	Mercedes	E400	2007
2	134	29	687698	06-09-2000	OH	100/300	2000	1413.14	5000000	430632	...	NO	34650	7700	3850	23100	Dodge	RAM	2007
3	256	41	227811	25-05-1990	IL	250/500	2000	1415.74	6000000	608117	...	NO	63400	6340	6340	50720	Chevrolet	Tahoe	2014
4	228	44	367455	06-06-2014	IL	500/1000	1000	1583.91	6000000	610706	...	NO	6500	1300	650	4550	Accura	RSX	2009

5 rows × 40 columns

In [6]:
```
df.tail(5)
```

Out[6]:

	months_as_customer	age	policy_number	policy_bind_date	policy_state	policy_csl	policy_deductable	policy_annual_premium	umbrella_limit	insured_zip	...	police_report_available	total_claim_amount	injury_claim	property_claim	vehicle_claim	auto_make	auto_model	auto_year
995	3	38	941851	16-07-1991	OH	500/1000	1000	1310.80	0	431289	...	?	87200	17440	8720	61040	Honda	Accord	2006
996	285	41	186934	05-01-2014	IL	100/300	1000	1436.79	0	608177	...	?	108480	18080	18080	72320	Volkswagen	Passat	2015
997	130	34	918516	17-02-2003	OH	250/500	500	1383.49	3000000	442797	...	YES	67500	7500	7500	52500	Suburu	Impreza	1996
998	458	62	533940	18-11-2011	IL	500/1000	2000	1356.92	5000000	441714	...	YES	46980	5220	5220	36540	Audi	A5	1998
999	456	60	556080	11-11-1996	OH	250/500	1000	766.19	0	612260	...	?	5060	460	920	3680	Mercedes	E400	2007

5 rows × 40 columns

Checking The Null Values

In [10]:
```
df.isnull()
```

Out[10]:

	months_as_customer	age	policy_number	policy_bind_date	policy_state	policy_csl	policy_deductable	policy_annual_premium	umbrella_limit	insured_zip	...	police_report_available	total_claim_amount	injury_claim	property_claim	vehicle_claim	auto_make	auto_model	auto_year
0	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False
996	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False
997	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False
998	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False
999	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False

1000 rows × 40 columns

In [11]:
```
df.isnull().sum()
```

Out[11]:
```
months_as_customer                0
age                               0
policy_number                     0
policy_bind_date                  0
policy_state                      0
policy_csl                        0
policy_deductable                 0
policy_annual_premium             0
umbrella_limit                    0
insured_zip                       0
insured_sex                       0
insured_education_level           0
insured_occupation                0
insured_hobbies                   0
insured_relationship              0
capital-gains                     0
capital-loss                      0
incident_date                     0
incident_type                     0
collision_type                    0
incident_severity                 0
authorities_contacted             0
incident_state                    0
incident_city                     0
incident_location                 0
incident_hour_of_the_day          0
number_of_vehicles_involved       0
property_damage                   0
bodily_injuries                   0
witnesses                         0
police_report_available           0
total_claim_amount                0
injury_claim                      0
property_claim                    0
vehicle_claim                     0
auto_make                         0
auto_model                        0
auto_year                         0
fraud_reported                    0
_c39                           1000
dtype: int64
```

In [12]:
```
df.isnull().sum().sum()
```

Out[12]:
```
1000
```

In [13]:
```
df.columns
```

Out[13]:
```
Index(['months_as_customer', 'age', 'policy_number', 'policy_bind_date',
       'policy_state', 'policy_csl', 'policy_deductable',
       'policy_annual_premium', 'umbrella_limit', 'insured_zip', 'insured_sex',
       'insured_education_level', 'insured_occupation', 'insured_hobbies',
       'insured_relationship', 'capital-gains', 'capital-loss',
       'incident_date', 'incident_type', 'collision_type', 'incident_severity',
       'authorities_contacted', 'incident_state', 'incident_city',
       'incident_location', 'incident_hour_of_the_day',
       'number_of_vehicles_involved', 'property_damage', 'bodily_injuries',
       'witnesses', 'police_report_available', 'total_claim_amount',
       'injury_claim', 'property_claim', 'vehicle_claim', 'auto_make',
       'auto_model', 'auto_year', 'fraud_reported', '_c39'],
      dtype='object')
```

In [14]:
```
df.sample(5)
```

Out[14]:

	months_as_customer	age	policy_number	policy_bind_date	policy_state	policy_csl	policy_deductable	policy_annual_premium	umbrella_limit	insured_zip	...	police_report_available	total_claim_amount	injury_claim	property_claim	vehicle_claim	auto_make	auto_model	auto_year
621	87	27	326289	03-01-2004	OH	100/300	500	1048.39	0	620962	...	YES	34650	6300	3150	25200	Ford	F150	1996
950	101	27	557218	23-11-1997	IL	500/1000	500	982.70	6000000	440865	...	NO	5170	940	470	3760	Toyota	Camry	2001
531	342	53	110408	14-11-2005	IN	100/300	1000	1028.44	0	602304	...	NO	80960	14720	7360	58880	Accura	MDX	2000
964	163	37	390256	25-11-2009	IN	500/1000	1000	1200.33	4000000	477631	...	YES	3900	390	780	2730	Volkswagen	Jetta	2008
850	13	21	990998	18-10-2006	IN	100/300	1000	1556.31	0	463809	...	YES	94930	8630	8630	77670	Accura	RSX	2014

5 rows × 40 columns

In [15]:
```
df.shape  let's check the shape of the dataset
```

Out[15]:
```
(1000, 40)
```

In [44]:
```
df.tail()
```

Out[44]:

	months_as_customer	age	policy_number	policy_bind_date	policy_state	policy_csl	policy_deductable	policy_annual_premium	umbrella_limit	insured_zip	...	police_report_available	total_claim_amount	injury_claim	property_claim	vehicle_claim	auto_make	auto_model	auto_year
995	3	38	941851	1991-07-16	OH	500/1000	1000	1310.80	0	431289	...	NaN	87200	17440	8720	61040	Honda	Accord	2006
996	285	41	186934	2014-05-01	IL	100/300	1000	1436.79	0	608177	...	NaN	108480	18080	18080	72320	Volkswagen	Passat	2015
997	130	34	918516	2003-02-17	OH	250/500	500	1383.49	3000000	442797	...	YES	67500	7500	7500	52500	Suburu	Impreza	1996
998	458	62	533940	2011-11-18	IL	500/1000	2000	1356.92	5000000	441714	...	YES	46980	5220	5220	36540	Audi	A5	1998
999	456	60	556080	1996-11-11	OH	250/500	1000	766.19	0	612260	...	NaN	5060	460	920	3680	Mercedes	E400	2007

5 rows × 40 columns

Describe Data

In [17]:
```
# let's describe the data
# It will demonstrate the count, mean, std dev, min, max, etc values for the Numerical features present in the data.
df.describe()
```

Out[17]:

	months_as_customer	age	policy_number	policy_deductable	policy_annual_premium	umbrella_limit	insured_zip	capital-gains	capital-loss	incident_hour_of_the_day	number_of_vehicles_involved	bodily_injuries	witnesses	total_claim_amount	injury_claim	proper
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1.000000e+03	1000.000000	1000.000000	1000.000000	1000.000000	1000.00000	1000.000000	1000.000000	1000.00000	1000.000000	100
mean	203.954000	38.948000	546238.648000	1136.000000	1256.406150	1.101000e+06	501214.488000	25126.100000	-26793.700000	11.644000	1.83900	0.992000	1.487000	52761.94000	7433.420000	739
std	115.113174	9.140287	257063.005276	611.864673	244.167395	2.297407e+06	71701.610941	27872.187708	28104.096686	6.951373	1.01888	0.820127	1.111335	26401.53319	4880.951853	482
min	0.000000	19.000000	100804.000000	500.000000	433.330000	-1.000000e+06	430104.000000	0.000000	-111100.000000	0.000000	1.00000	0.000000	0.000000	100.00000	0.000000	
25%	115.750000	32.000000	335980.250000	500.000000	1089.607500	0.000000e+00	448404.500000	0.000000	-51500.000000	6.000000	1.00000	0.000000	1.000000	41812.50000	4295.000000	444
50%	199.500000	38.000000	533135.000000	1000.000000	1257.200000	0.000000e+00	466445.500000	0.000000	-23250.000000	12.000000	1.00000	1.000000	1.000000	58055.00000	6775.000000	675
75%	276.250000	44.000000	759099.750000	2000.000000	1415.695000	0.000000e+00	603251.000000	51025.000000	0.000000	17.000000	3.00000	2.000000	2.000000	70592.50000	11305.000000	1088
max	479.000000	64.000000	999435.000000	2000.000000	2047.590000	1.000000e+07	620962.000000	100500.000000	0.000000	23.000000	4.00000	2.000000	3.000000	114920.00000	21450.000000	2367

In [18]:
```
# let's get the information about the dataset
df.info()
```

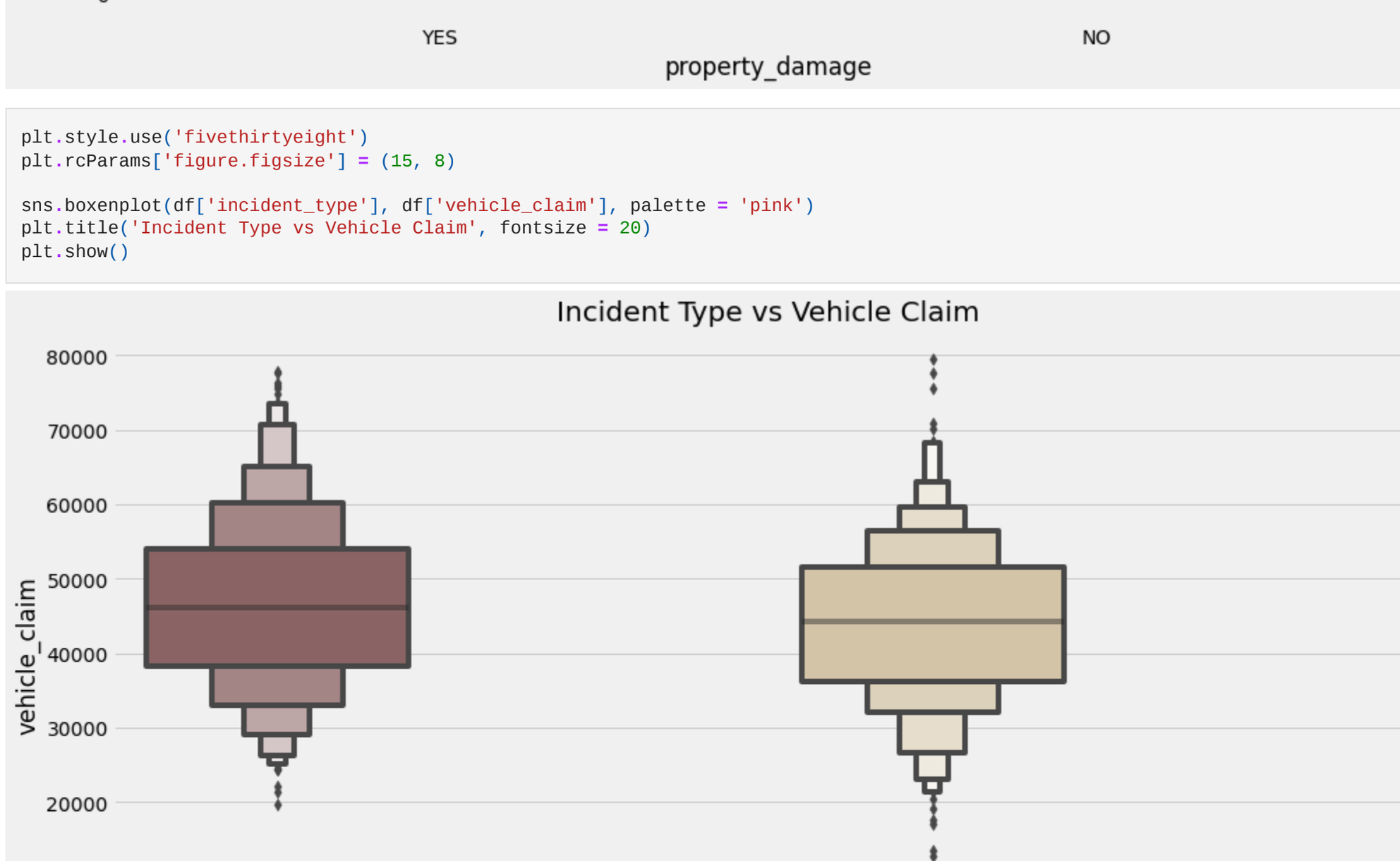
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 40 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   months_as_customer           1000 non-null   int64
 1   age                          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   object
 28  bodily_injuries              1000 non-null   int64
 29  witnesses                    1000 non-null   int64
 30  police_report_available      1000 non-null   object
 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   object
 39  _c39                         0 non-null      float64
dtypes: float64(2), int64(17), object(21)
memory usage: 312.6+ KB
```

In [20]:
```
df.isnull().any()
```

Out[20]:
```
months_as_customer             False
age                            False
policy_number                  False
policy_bind_date               False
policy_state                   False
policy_csl                     False
policy_deductable              False
policy_annual_premium          False
umbrella_limit                 False
insured_zip                    False
insured_sex                    False
insured_education_level        False
insured_occupation             False
insured_hobbies                False
insured_relationship           False
capital-gains                  False
capital-loss                   False
incident_date                  False
incident_type                  False
collision_type                  True
incident_severity              False
authorities_contacted          False
incident_state                 False
incident_city                  False
incident_location              False
incident_hour_of_the_day       False
number_of_vehicles_involved    False
property_damage                 True
bodily_injuries                False
witnesses                      False
police_report_available         True
total_claim_amount             False
injury_claim                   False
property_claim                 False
vehicle_claim                  False
auto_make                      False
auto_model                     False
auto_year                      False
fraud_reported                 False
_c39                            True
dtype: bool
```

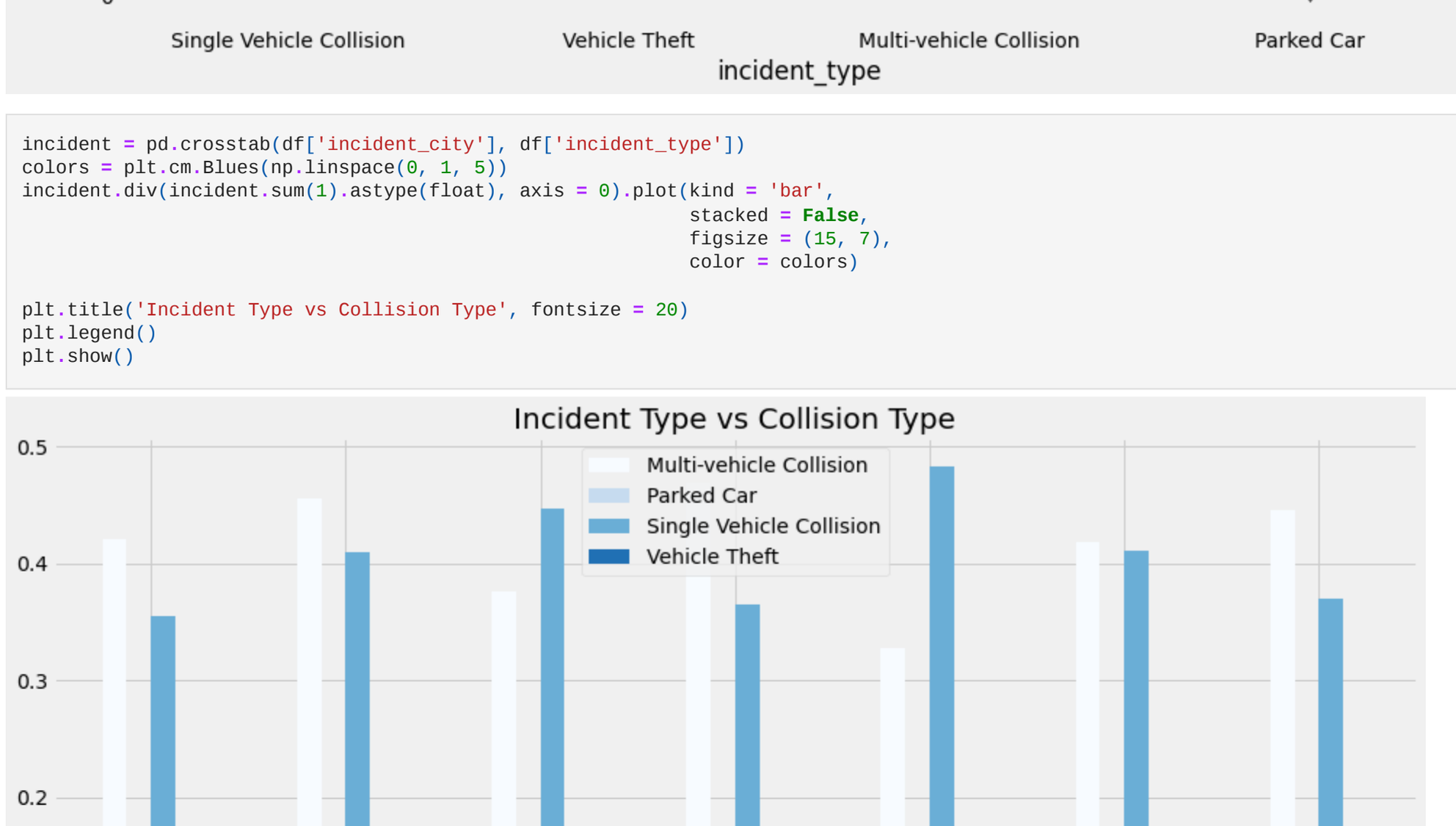
In [24]:
```
import warnings
warnings.filterwarnings('ignore')
plt.style.use('fivethirtyeight')
plt.rcParams['figure.figsize'] = (15, 8)

sns.stripplot(df['property_damage'], df['property_claim'], palette = 'bone')
plt.title('Incident Type vs Vehicle Claim', fontsize = 20)
plt.show()
```



In [25]:
```
plt.style.use('fivethirtyeight')
plt.rcParams['figure.figsize'] = (15, 8)

sns.boxenplot(df['incident_type'], df['vehicle_claim'], palette = 'pink')
plt.title('Incident Type vs Vehicle Claim', fontsize = 20)
plt.show()
```



In [26]:
```
incident = pd.crosstab(df['incident_city'], df['incident_type'])
colors = plt.cm.Blues(np.linspace(0, 1, 5))
incident.div(incident.sum(1).astype(float), axis = 0).plot(kind = 'bar',
                                                          stacked = False,
                                                          figsize = (15, 7),
                                                          color = colors)

plt.title('Incident Type vs Collision Type', fontsize = 20)
plt.legend()
plt.show()
```

In [30]:
```
df['policy_bind_date'] = pd.to_datetime(df['policy_bind_date'], errors = 'coerce')
```

In [29]:
```
df['fraud_reported'] = df['fraud_reported'].replace(('Y','N'),(0,1))
```

In [31]:
```
x = df.drop(['fraud_reported'], axis = 1)
y = df['fraud_reported']
print("Shape of x :", x.shape)
print("Shape of y :", y.shape)
```

```
Shape of x : (1000, 39)
Shape of y : (1000,)
```

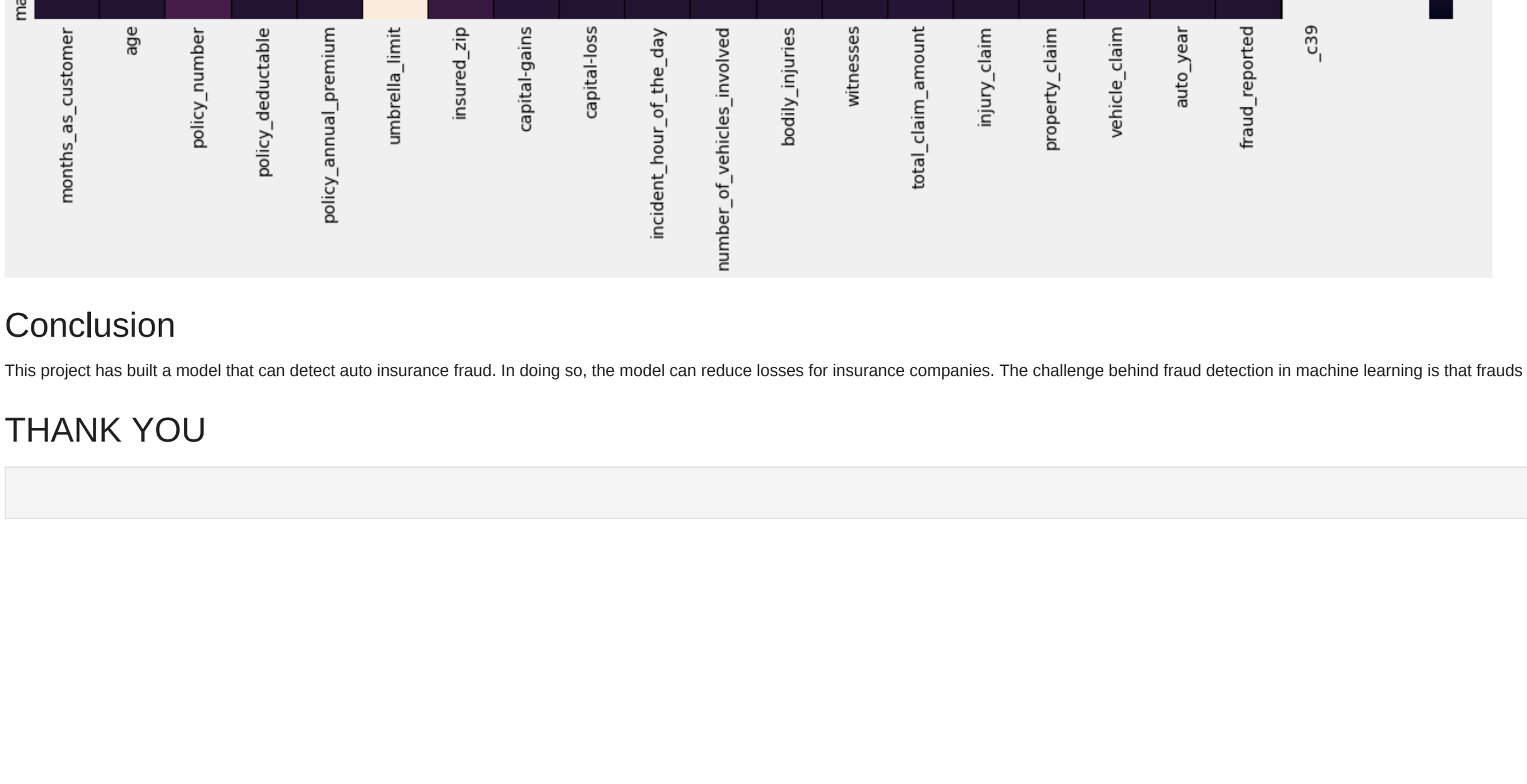
Correlation

In [33]:
```
plt.rcParams['figure.figsize'] = (15, 10)
sns.heatmap(df.corr(), cmap = 'copper')
plt.title('Heat Map for Correlations', fontsize = 20)
plt.show()
```

Heat Map for Correlations
months_as_customer
age
policy_number
policy_deductable
policy_annual_premium
umbrella_limit
insured_zip
capital-gains
capital-loss
incident_hour_of_the_day
number_of_vehicles_involved
bodily_injuries
witnesses
total_claim_amount
injury_claim
property_claim
vehicle_claim
auto_year
fraud_reported
_c39
1.0
0.8
0.6
0.4
0.2
0.0

In [39]:
```
import matplotlib.pyplot as plt
plt.figure(figsize=(22,7))
sns.heatmap(df.describe(),annot=True,linewidths=0.1,linecolor="black",fmt="0.2f")
```

Out[39]:
```
<AxesSubplot:>
```



count
mean
std
min
25%
50%
75%
max
months_as_customer
age
policy_number
policy_deductable
policy_annual_premium
umbrella_limit
insured_zip
capital-gains
capital-loss
incident_hour_of_the_day
number_of_vehicles_involved
bodily_injuries
witnesses
total_claim_amount
injury_claim
property_claim
vehicle_claim
auto_year
fraud_reported
_c39
1e7
1.0
0.8
0.6
0.4
0.2
0.0

Conclusion

This project has built a model that can detect auto insurance fraud. In doing so, the model can reduce losses for insurance companies. The challenge behind fraud detection in machine learning is that frauds are far less common as compared to legit insurance claims.

THANK YOU

In [ ]: