

Vehicle Insurance Data Set-Classification

```
In [2]:  
#importing required Libraries  
import pandas as pd  
import numpy as np  
import seaborn as sns  
import matplotlib.pyplot as plt  
%matplotlib inline  
sns.set_style('whitegrid')
```

1:EDA on Vehicle Insurance DataSet

```
In [6]:  
veh=pd.read_csv(r"C:\Users\ADMIN\Desktop\Projects\Data sets\train.csv")  
veh.head()
```

```
Out[6]:  


|          | <b>id</b> | <b>Gender</b> | <b>Age</b> | <b>Driving_License</b> | <b>Region_Code</b> | <b>Previously_Insured</b> | <b>Vehicle_Age</b> | <b>Vehicle_Damage</b> |
|----------|-----------|---------------|------------|------------------------|--------------------|---------------------------|--------------------|-----------------------|
| <b>0</b> | 1         | Male          | 44         | 1                      | 28.0               | 0                         | > 2 Years          | Yes                   |
| <b>1</b> | 2         | Male          | 76         | 1                      | 3.0                | 0                         | 1-2 Year           | No                    |
| <b>2</b> | 3         | Male          | 47         | 1                      | 28.0               | 0                         | > 2 Years          | Yes                   |
| <b>3</b> | 4         | Male          | 21         | 1                      | 11.0               | 1                         | < 1 Year           | No                    |
| <b>4</b> | 5         | Female        | 29         | 1                      | 41.0               | 1                         | < 1 Year           | No                    |

```

```
In [7]:  
veh.shape
```

```
Out[7]: (381109, 12)
```

```
In [8]:  
veh.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 381109 entries, 0 to 381108  
Data columns (total 12 columns):  
 #   Column           Non-Null Count  Dtype     
---  --  
 0   id              381109 non-null  int64    
 1   Gender          381109 non-null  object    
 2   Age             381109 non-null  int64    
 3   Driving_License 381109 non-null  int64    
 4   Region_Code     381109 non-null  float64  
 5   Previously_Insured 381109 non-null  int64    
 6   Vehicle_Age     381109 non-null  object    
 7   Vehicle_Damage   381109 non-null  object    
 8   Annual_Premium   381109 non-null  float64  
 9   Policy_Sales_Channel 381109 non-null  float64  
 10  Vintage          381109 non-null  int64    
 11  Response         381109 non-null  int64    
dtypes: float64(3), int64(6), object(3)  
memory usage: 34.9+ MB
```

```
In [9]:
```

```
print(veh.Response.value_counts())
veh['Response'].value_counts(normalize=True)

0    334399
1    46710
Name: Response, dtype: int64
0    0.877437
1    0.122563
Name: Response, dtype: float64
```

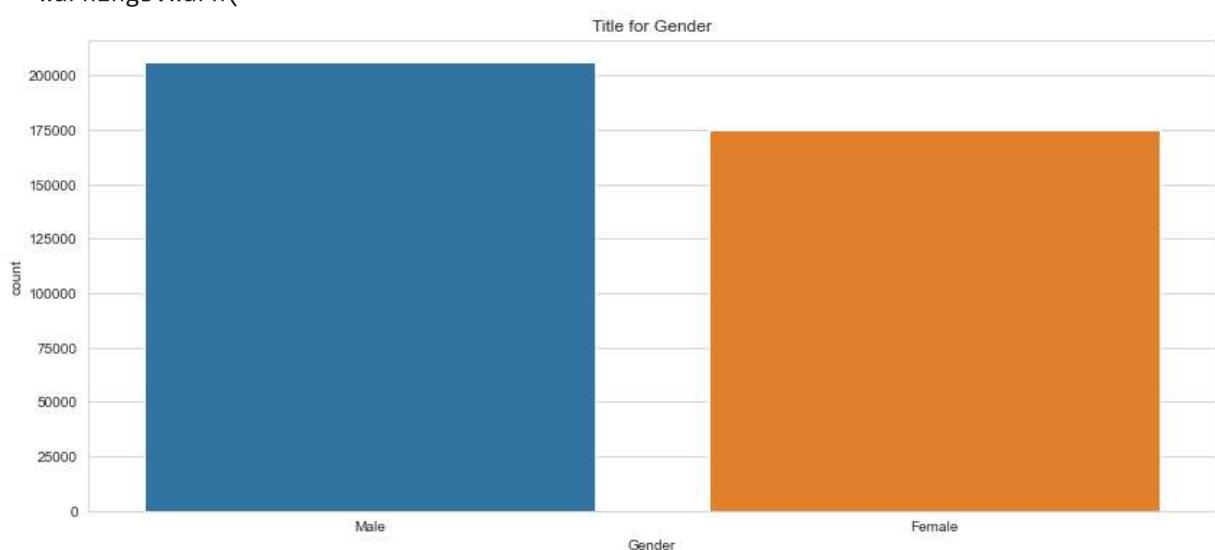
```
In [10]: veh.isnull().sum()
```

```
Out[10]: id              0
Gender          0
Age             0
Driving_License 0
Region_Code     0
Previously_Insured 0
Vehicle_Age     0
Vehicle_Damage   0
Annual_Premium   0
Policy_Sales_Channel 0
Vintage          0
Response         0
dtype: int64
```

```
In [11]: #Categorical Features
def plott(coln):
    plt.figure(figsize=(14,6))
    plt.title('Title for ' + coln)
    sns.countplot(veh[coln])
    plt.show()
```

```
In [12]: plott('Gender')
```

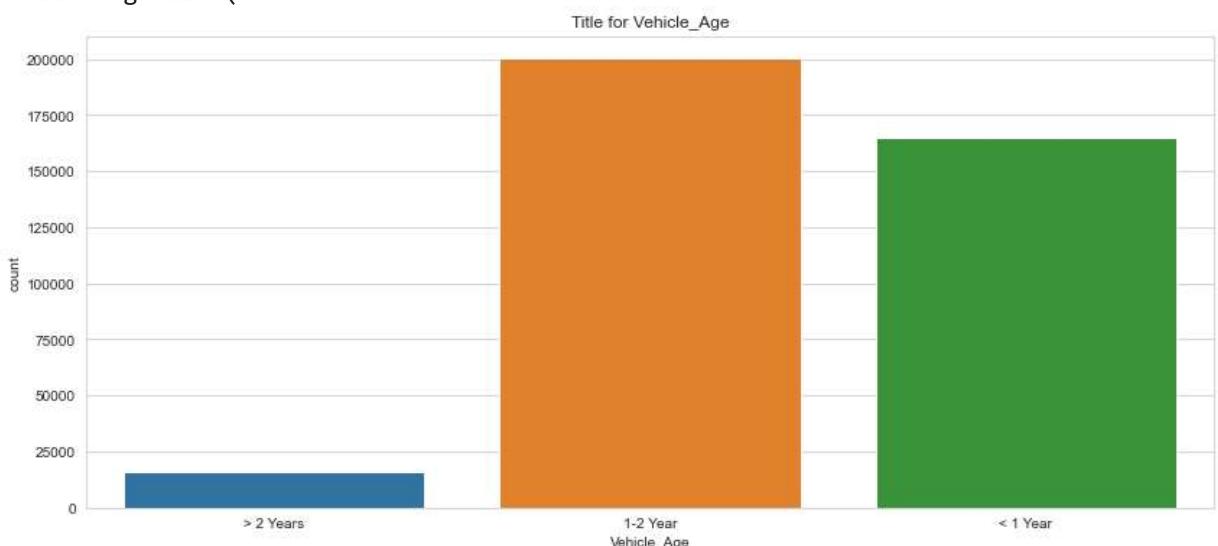
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only valid p
ositional argument will be `data`, and passing other arguments without an explicit k
eyword will result in an error or misinterpretation.
warnings.warn(



```
In [13]: plott('Vehicle_Age')
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:  
Pass the following variable as a keyword arg: x. From version 0.12, the only valid p  
ositional argument will be `data`, and passing other arguments without an explicit k  
eyword will result in an error or misinterpretation.
```

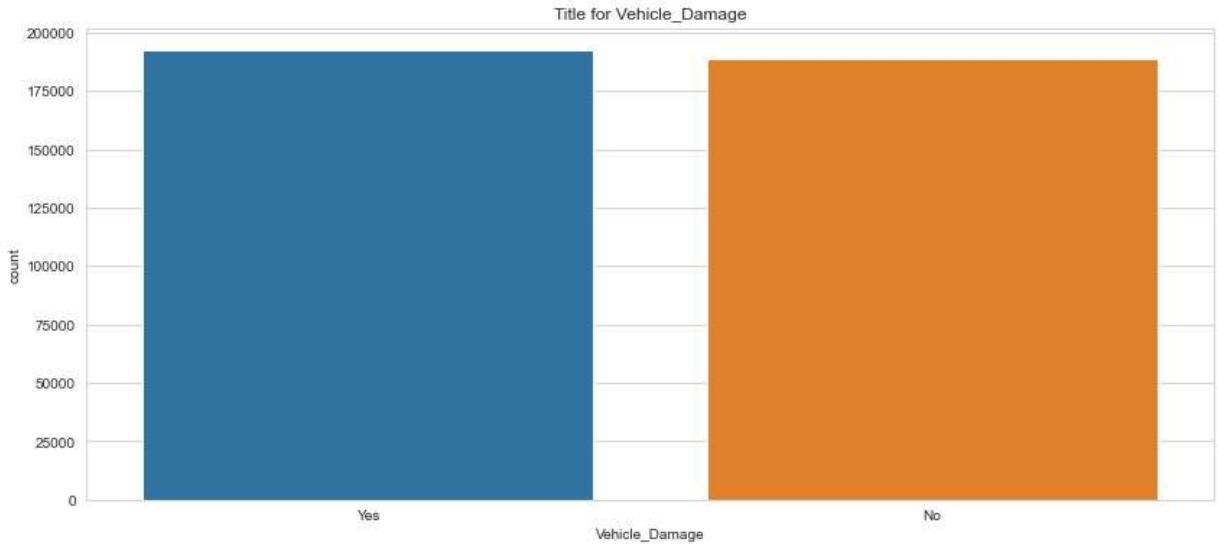
```
    warnings.warn(
```



```
In [14]: plott('Vehicle_Damage')
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:  
Pass the following variable as a keyword arg: x. From version 0.12, the only valid p  
ositional argument will be `data`, and passing other arguments without an explicit k  
eyword will result in an error or misinterpretation.
```

```
    warnings.warn(
```



```
In [15]: veh.describe()
```

```
Out[15]:
```

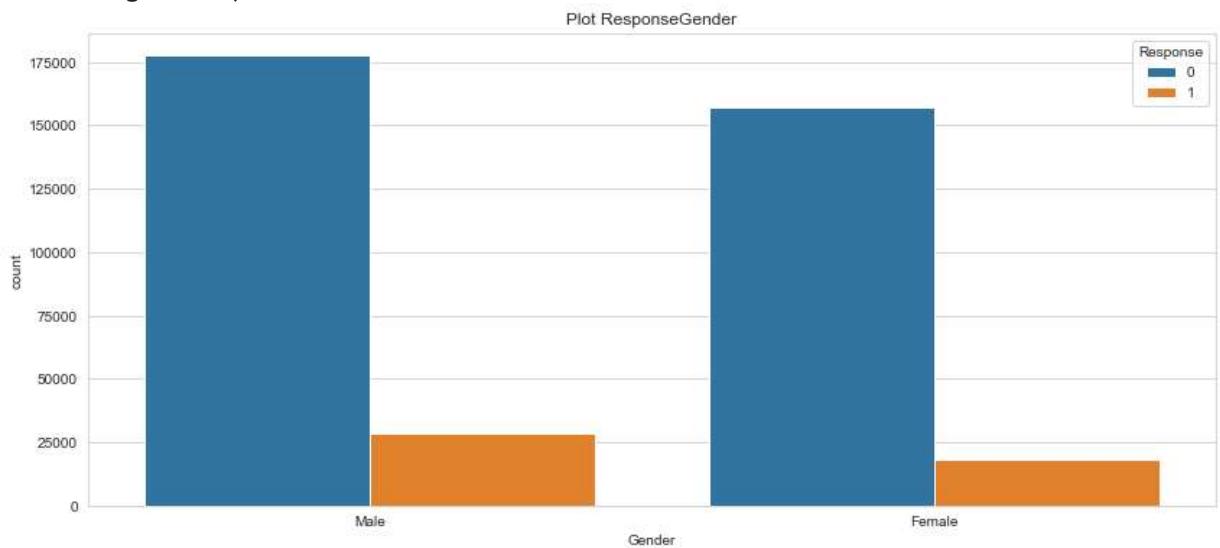
	id	Age	Driving_License	Region_Code	Previously_Insured	Annual_Premium
count	381109.000000	381109.000000	381109.000000	381109.000000	381109.000000	381109.0
mean	190555.000000	38.822584	0.997869	26.388807	0.458210	30564.3
std	110016.836208	15.511611	0.046110	13.229888	0.498251	17213.1
min	1.000000	20.000000	0.000000	0.000000	0.000000	2630.0
25%	95278.000000	25.000000	1.000000	15.000000	0.000000	24405.0
50%	190555.000000	36.000000	1.000000	28.000000	0.000000	31669.0

	id	Age	Driving_License	Region_Code	Previously_Insured	Annual_Premium
75%	285832.000000	49.000000	1.000000	35.000000	1.000000	39400.000000
max	381109.000000	85.000000	1.000000	52.000000	1.000000	540165.000000

```
In [16]: cols=['Gender', 'Vehicle_Age', 'Vehicle_Damage']
for i in cols:
    plt.figure(figsize=(14,6))
    plt.title('Plot Response'+ i)
    sns.countplot(veh[i],hue=veh['Response'])
    plt.show()
    print()
```

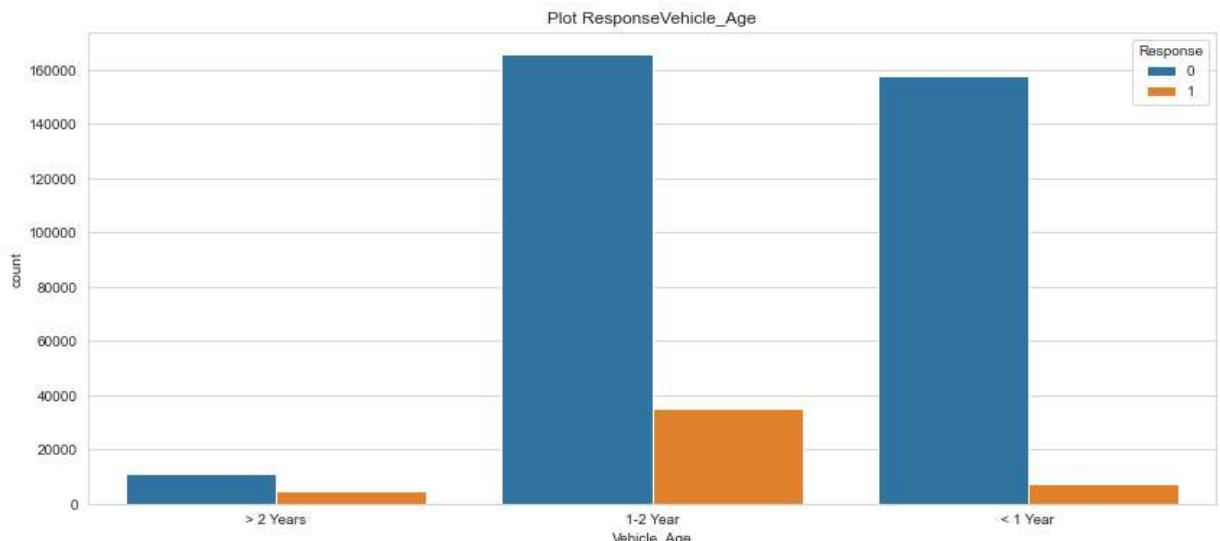
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only valid p
ositional argument will be `data`, and passing other arguments without an explicit k
eyword will result in an error or misinterpretation.

```
warnings.warn(
```



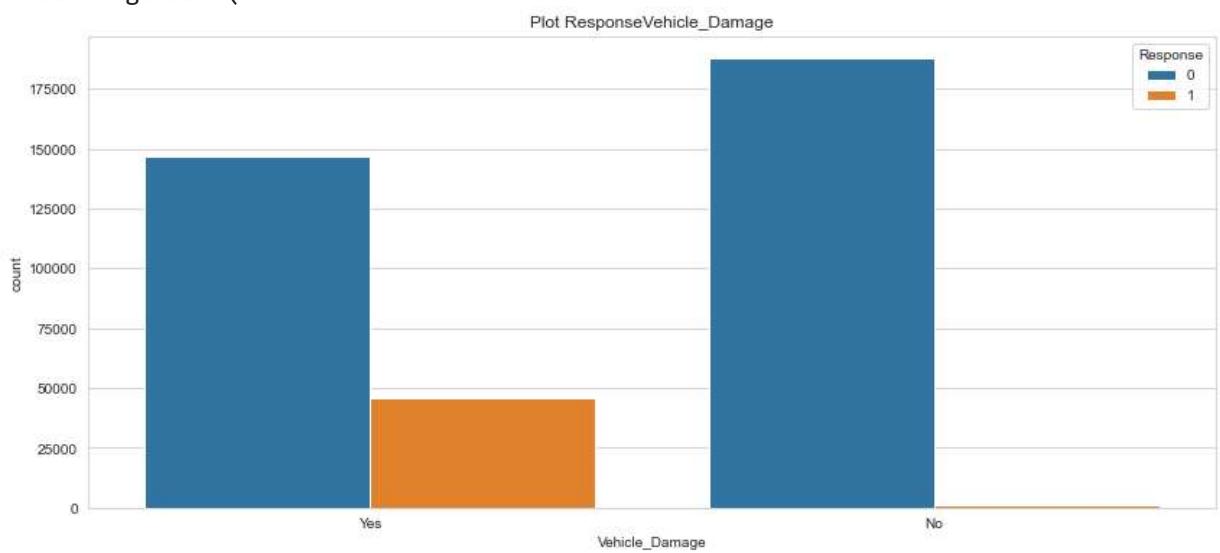
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only valid p
ositional argument will be `data`, and passing other arguments without an explicit k
eyword will result in an error or misinterpretation.

```
warnings.warn(
```



```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:  
Pass the following variable as a keyword arg: x. From version 0.12, the only valid p  
ositional argument will be `data`, and passing other arguments without an explicit k  
eyword will result in an error or misinterpretation.
```

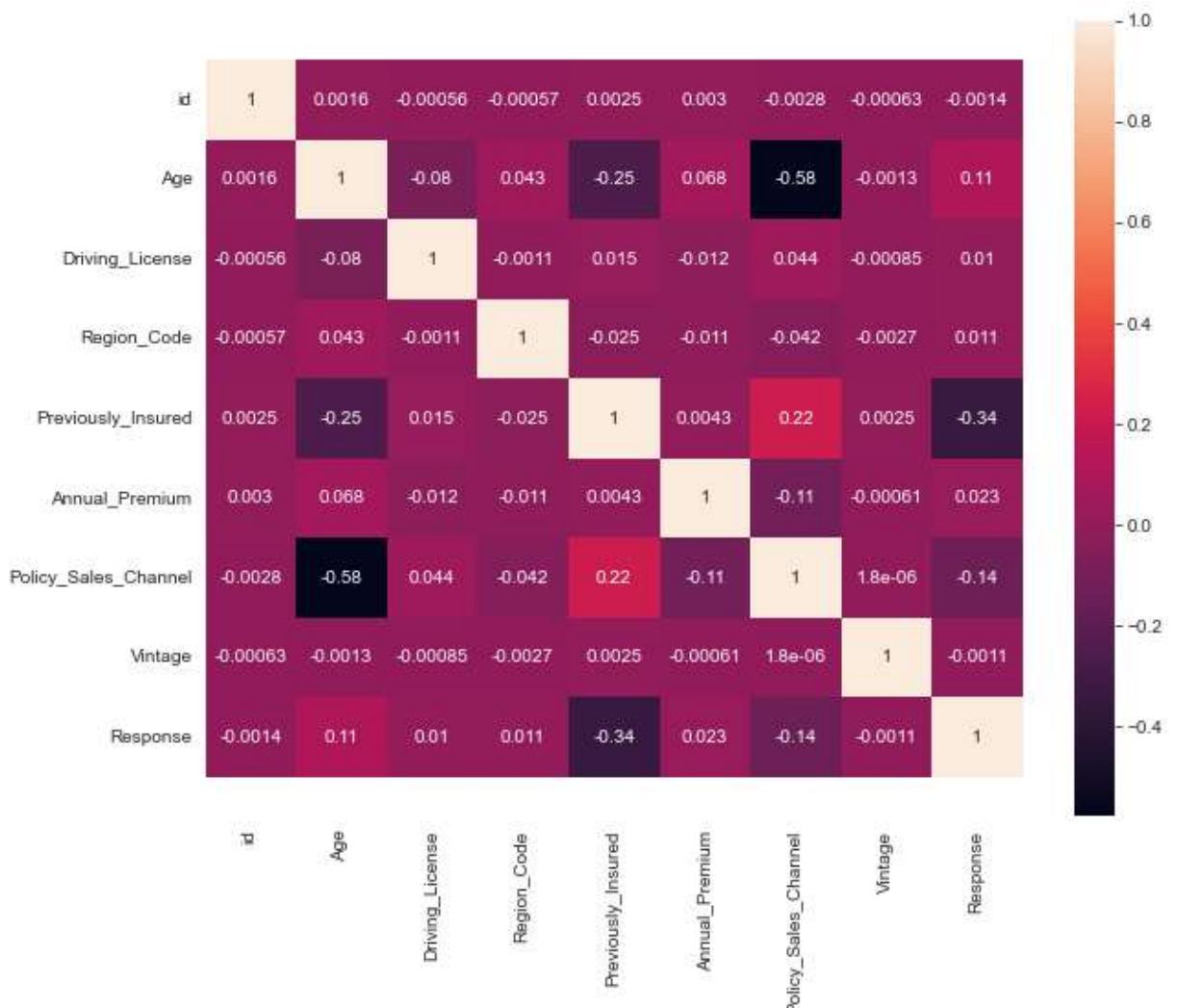
```
    warnings.warn(
```



2: Statistical Analysis on Vehicle Insurance DataSet

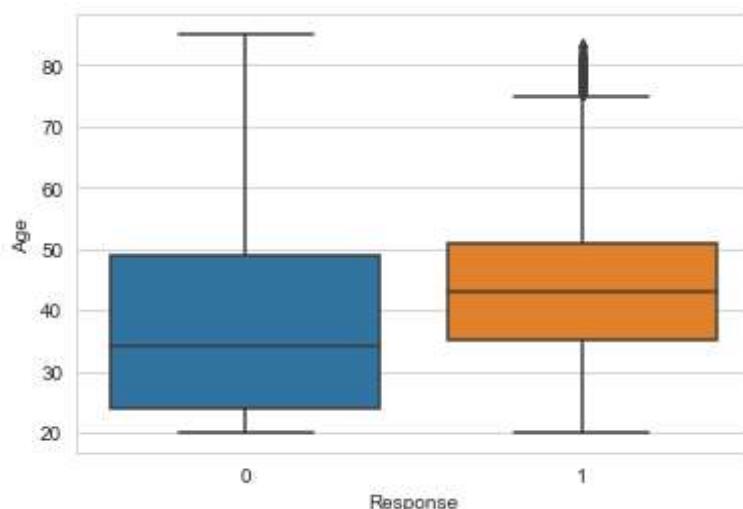
```
In [21]:
```

```
#Heatmap to show the correlation between various variables of the dataset  
plt.figure(figsize=(10, 8))  
cor = veh.corr()  
ax = sns.heatmap(cor, annot=True)  
bottom, top = ax.get_ylim()  
ax.set_ylim(bottom + 0.5, top - 0.5)  
plt.show()
```



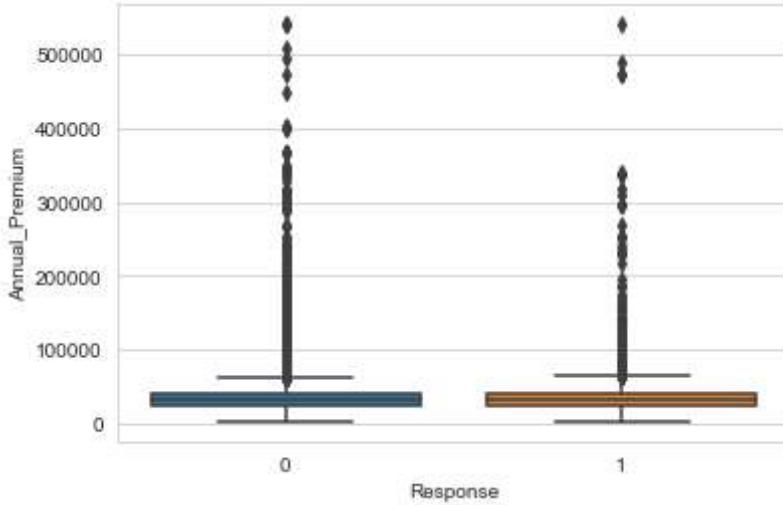
```
In [22]: sns.boxplot(x=veh.Response,y=veh.Age)
```

```
Out[22]: <AxesSubplot:xlabel='Response', ylabel='Age'>
```



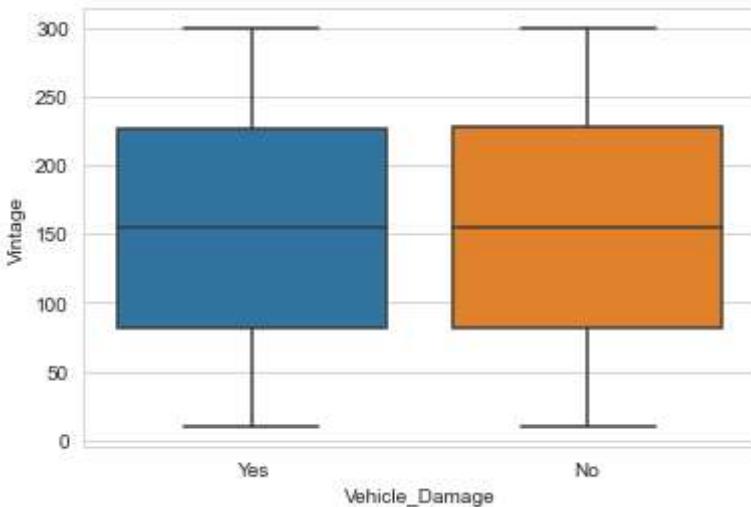
```
In [23]: sns.boxplot(x=veh.Response,y=veh.Annual_Premium)
```

```
Out[23]: <AxesSubplot:xlabel='Response', ylabel='Annual_Premium'>
```



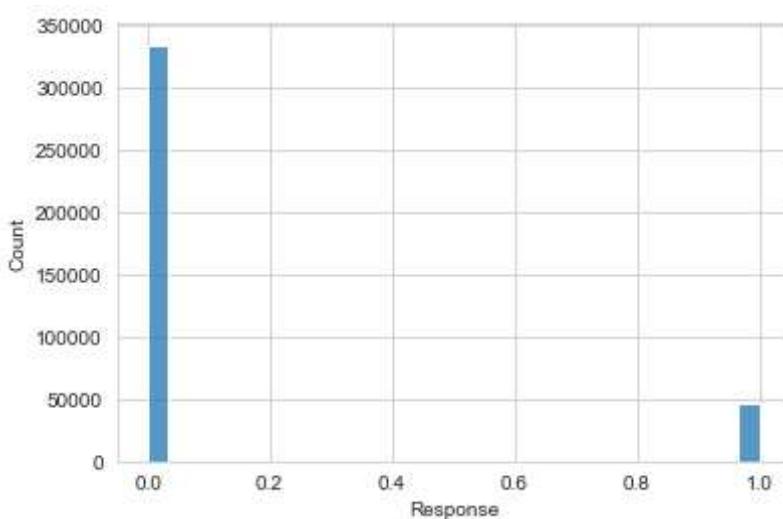
```
In [24]: sns.boxplot(x=veh.Vehicle_Damage,y=veh.Vintage)
```

```
Out[24]: <AxesSubplot:xlabel='Vehicle_Damage', ylabel='Vintage'>
```



```
In [25]: sns.histplot(data=veh['Response'],bins=30)
```

```
Out[25]: <AxesSubplot:xlabel='Response', ylabel='Count'>
```



3: Build the model for Vehicle Insurance

DataSet

```
In [27]: #Label Encoder:Encode target Labels with value between 0 and n_classes-1.  
from sklearn.preprocessing import LabelBinarizer,LabelEncoder  
veh1=veh.select_dtypes(exclude=np.number)  
veh1.columns
```

```
Out[27]: Index(['Gender', 'Vehicle_Age', 'Vehicle_Damage'], dtype='object')
```

```
In [28]: lb=LabelEncoder()  
lb.fit_transform(np.array(veh['Vehicle_Age']).reshape(-1,1))
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConver  
rsionWarning: A column-vector y was passed when a 1d array was expected. Please chan  
ge the shape of y to (n_samples, ), for example using ravel().  
    return f(*args, **kwargs)  
array([2, 0, 2, ..., 1, 2, 0])
```

```
In [29]: for i in veh1.columns:  
    veh[i]=lb.fit_transform(np.array(veh[i]).reshape(-1,1))
```

```
In [30]: veh.head()
```

```
Out[30]:   id  Gender  Age  Driving_License  Region_Code  Previously_Insured  Vehicle_Age  Vehicle_Damage  
0     1        1   44                  1          28.0                 0            2             1  
1     2        1   76                  1          3.0                 0            0             0  
2     3        1   47                  1          28.0                 0            2             1  
3     4        1   21                  1          11.0                1            1             0  
4     5        0   29                  1          41.0                1            1             0
```



```
In [31]: veh.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 381109 entries, 0 to 381108  
Data columns (total 12 columns):  
 #   Column           Non-Null Count  Dtype    
---  --    
 0   id              381109 non-null  int64  
 1   Gender          381109 non-null  int32  
 2   Age              381109 non-null  int64  
 3   Driving_License 381109 non-null  int64  
 4   Region_Code     381109 non-null  float64  
 5   Previously_Insured 381109 non-null  int64  
 6   Vehicle_Age     381109 non-null  int32  
 7   Vehicle_Damage  381109 non-null  int32  
 8   Annual_Premium  381109 non-null  float64  
 9   Policy_Sales_Channel 381109 non-null  float64  
 10  Vintage          381109 non-null  int64  
 11  Response         381109 non-null  int64
```

```
dtypes: float64(3), int32(3), int64(6)
memory usage: 30.5 MB
```

```
In [33]: vaho=pd.read_csv(r"C:\Users\ADMIN\Desktop\Projects\Data sets\test.csv")
vaho.shape
```

```
Out[33]: (127037, 11)
```

```
In [34]: vaho.head()
```

```
Out[34]:
```

	id	Gender	Age	Driving_License	Region_Code	Previously_Insured	Vehicle_Age	Vehicle_Dai
0	381110	Male	25	1	11.0	1	< 1 Year	
1	381111	Male	40	1	28.0	0	1-2 Year	
2	381112	Male	47	1	28.0	0	1-2 Year	
3	381113	Male	24	1	27.0	1	< 1 Year	
4	381114	Male	27	1	28.0	1	< 1 Year	

```
◀ ▶
```

```
In [35]: #using only train data
X=veh.drop(['id','Response'],axis=1)
y=veh['Response']
```

```
In [36]: #Splitting the data into train and test
```

```
from sklearn.model_selection import train_test_split

X_train , X_test , y_train , y_test = train_test_split(X,y,test_size = 0.30 , random_state=42)

print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(266776, 10)
(114333, 10)
(266776,)
(114333,)
```

4: Test the model on Vehicle Insurance DataSet

```
In [41]: from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report,confusion_matrix,f1_score,accuracy
logmodel = LogisticRegression()
logmodel.fit(X_train,y_train)
print('Train acc:',accuracy_score(y_train,logmodel.predict(X_train)))
print('Test acc:',accuracy_score(y_test,logmodel.predict(X_test)))
```

```
Train acc: 0.8767167961135934
Test acc: 0.8791162656450894
```

```
In [42]: predictions = logmodel.predict(X_test)
print(classification_report(y_test,predictions))
```

	precision	recall	f1-score	support
0	0.88	1.00	0.94	100512
1	0.00	0.00	0.00	13821
accuracy			0.88	114333
macro avg	0.44	0.50	0.47	114333
weighted avg	0.77	0.88	0.82	114333

```
C:\Users\ADMIN\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1248:
UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 i
n labels with no predicted samples. Use `zero_division` parameter to control this be
havior.
```

```
_warn_prf(average, modifier, msg_start, len(result))
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1248:
UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 i
n labels with no predicted samples. Use `zero_division` parameter to control this be
havior.
```

```
_warn_prf(average, modifier, msg_start, len(result))
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1248:
UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 i
n labels with no predicted samples. Use `zero_division` parameter to control this be
havior.
```

```
_warn_prf(average, modifier, msg_start, len(result))
```

```
In [43]: cm=confusion_matrix(y_test,predictions)
cm
```

```
Out[43]: array([[100512,      0],
       [ 13821,      0]], dtype=int64)
```

```
In [ ]:
```

In []:

In []: