```python
# Step 1: Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Step 2: Load Dataset
data = pd.read_csv(r"C:\Users\HP\Downloads\Vehicle_Insurance.csv")
data
```

```
            id  Gender  Age  Driving_License  Region_Code
Previously_Insured  \
0            1    Male   44                1         28.0
0
1            2    Male   76                1          3.0
0
2            3    Male   47                1         28.0
0
3            4    Male   21                1         11.0
1
4            5  Female   29                1         41.0
1
...        ...     ...  ...              ...          ...
...
381104  381105    Male   74                1         26.0
1
381105  381106    Male   30                1         37.0
1
381106  381107    Male   21                1         30.0
1
381107  381108  Female   68                1         14.0
0
381108  381109    Male   46                1         29.0
0

       Vehicle_Age Vehicle_Damage  Annual_Premium
Policy_Sales_Channel  \
0        > 2 Years            Yes         40454.0
26.0
1         1-2 Year             No         33536.0
26.0
2        > 2 Years            Yes         38294.0
26.0
3         < 1 Year             No         28619.0
152.0
4         < 1 Year             No         27496.0
152.0
...            ...            ...             ...                    ..
.
381104    1-2 Year             No         30170.0
```

```
26.0
381105    < 1 Year              No          40016.0
152.0
381106    < 1 Year              No          35118.0
160.0
381107   > 2 Years             Yes          44617.0
124.0
381108    1-2 Year              No          41777.0
26.0

        Vintage  Response
0           217         1
1           183         0
2            27         1
3           203         0
4            39         0
...         ...       ...
381104       88         0
381105      131         0
381106      161         0
381107       74         0
381108      237         0

[381109 rows x 12 columns]
```

```
# Step 3: Basic Data Info
print("Shape of dataset:", data.shape)
```

```
Shape of dataset: (381109, 12)
```

```
print("\nFirst 5 rows:\n", data.head(5))
```

```
First 5 rows:
    id  Gender  Age  Driving_License  Region_Code  Previously_Insured
\
0   1    Male   44                1         28.0                   0

1   2    Male   76                1          3.0                   0

2   3    Male   47                1         28.0                   0

3   4    Male   21                1         11.0                   1

4   5  Female   29                1         41.0                   1


  Vehicle_Age Vehicle_Damage  Annual_Premium  Policy_Sales_Channel
Vintage  \
0   > 2 Years            Yes         40454.0                  26.0
217
```

```
1    1-2 Year              No          33536.0                    26.0
183
2   > 2 Years             Yes          38294.0                    26.0
27
3    < 1 Year              No          28619.0                   152.0
203
4    < 1 Year              No          27496.0                   152.0
39

   Response
0         1
1         0
2         1
3         0
4         0
```

```
print("Dataset Info:")
print(data.info())
```

```
Dataset 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
None
```

```
# Step 4: Check Missing Values
print("\nMissing values in each column:\n", data.isnull().sum())
```

```

Missing values in each column:
 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
```

```
# Fill missing numerical values with mean, categorical with mode
for col in data.columns:
    if data[col].dtype in ["float64", "int64"]:  # This code makes
sure that only number columns are filled with the mean
        data[col] = data[col].fillna(data[col].mean())
    else:
        data[col] = data[col].fillna(data[col].mode()[0]) # This line
fills missing values in text columns using the mode.

print("\nMissing values after filling:\n", data.isnull().sum())
```

```
Missing values after filling:
 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
```

```
# Step 5: Summary Statistics
print("\nSummary Statistics:\n", data.describe())
```

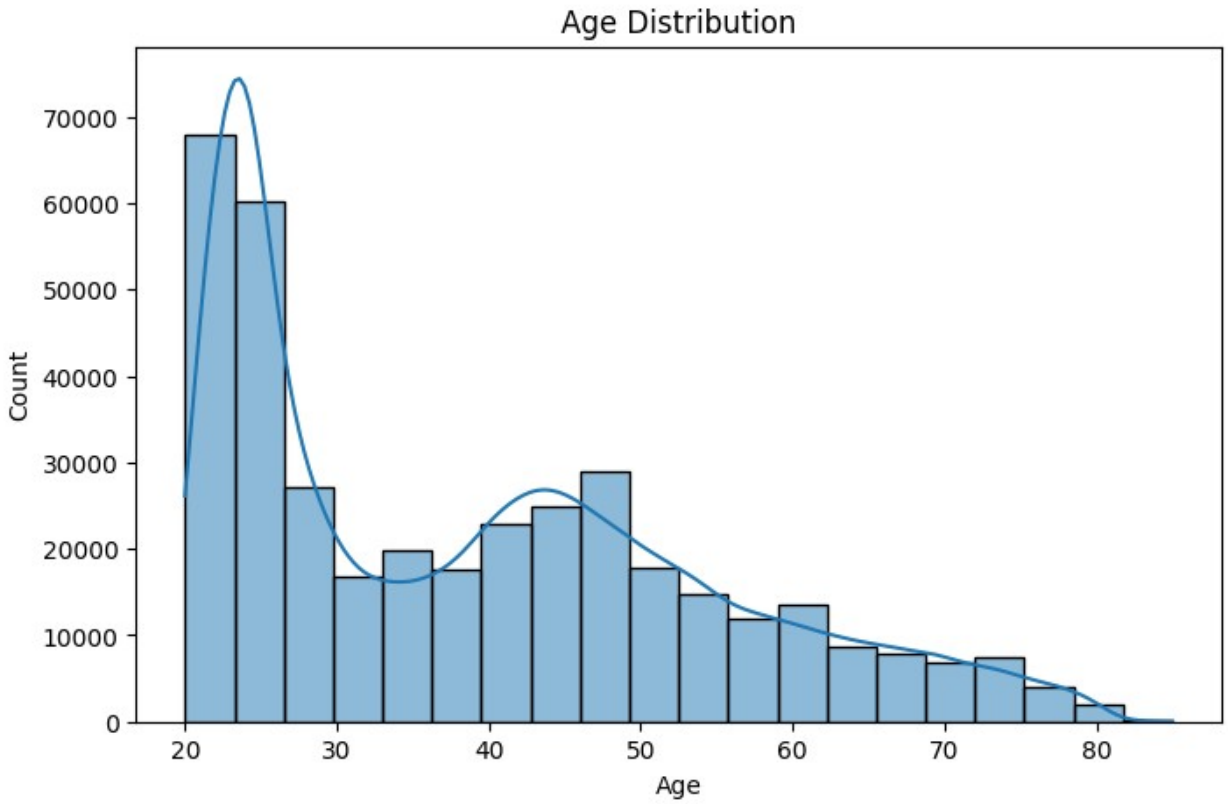
```
Summary Statistics:
                  id            Age  Driving_License
Region_Code  \
count  381109.000000  381109.000000    381109.000000  381109.000000
mean   190555.000000      38.822584         0.997869      26.388807
std    110016.836208      15.511611         0.046110      13.229888
min         1.000000      20.000000         0.000000       0.000000
25%     95278.000000      25.000000         1.000000      15.000000
50%    190555.000000      36.000000         1.000000      28.000000
```

```
75%    285832.000000      49.000000          1.000000      35.000000
max    381109.000000      85.000000          1.000000      52.000000

       Previously_Insured  Annual_Premium  Policy_Sales_Channel  \
count       381109.000000   381109.000000         381109.000000
mean             0.458210    30564.389581            112.034295
std              0.498251    17213.155057             54.203995
min              0.000000     2630.000000              1.000000
25%              0.000000    24405.000000             29.000000
50%              0.000000    31669.000000            133.000000
75%              1.000000    39400.000000            152.000000
max              1.000000   540165.000000            163.000000

             Vintage       Response
count  381109.000000  381109.000000
mean      154.347397       0.122563
std        83.671304       0.327936
min        10.000000       0.000000
25%        82.000000       0.000000
50%       154.000000       0.000000
75%       227.000000       0.000000
max       299.000000       1.000000
```

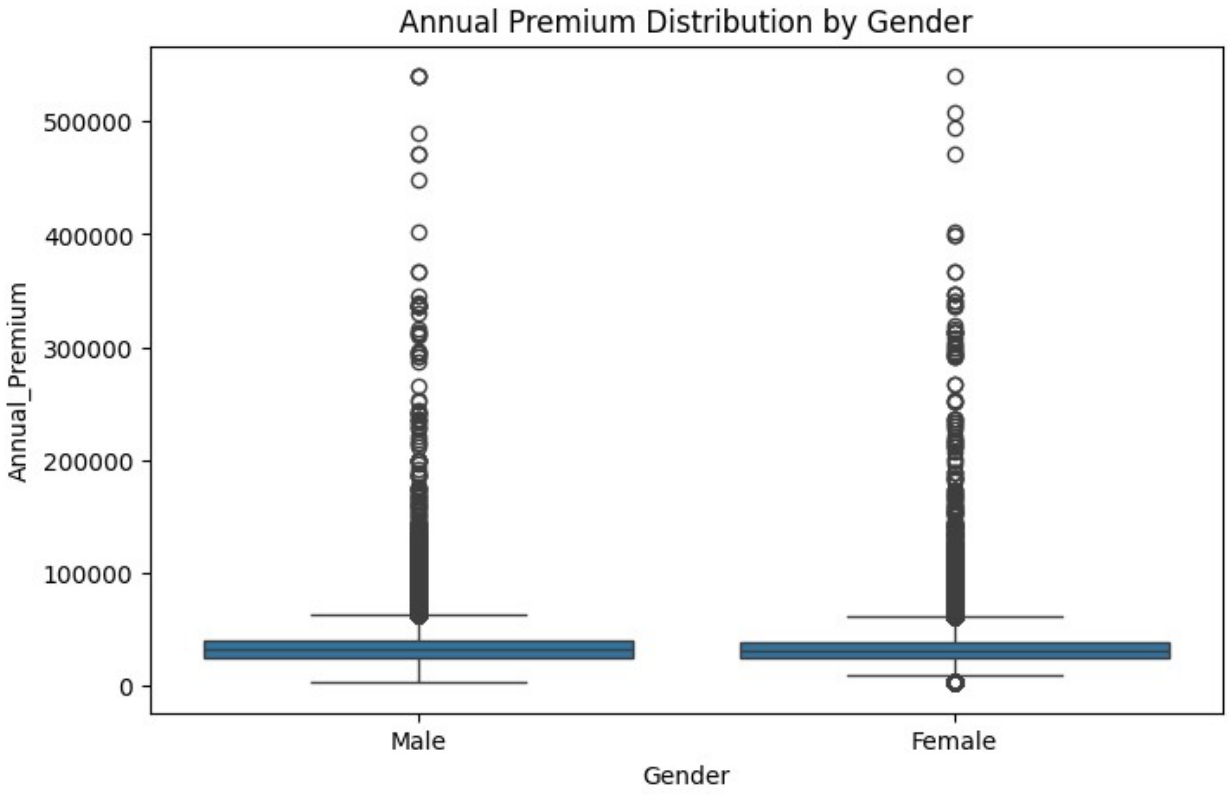
```
# Step 6: Detect and Handle Outliers
Q1 = data['Annual_Premium'].quantile(0.25)
Q3 = data['Annual_Premium'].quantile(0.75)
IQR = Q3 - Q1
filtered_data = data[~((data['Annual_Premium'] < (Q1 - 1.5 * IQR)) |
                       (data['Annual_Premium'] > (Q3 + 1.5 * IQR)))]

print("\nOriginal rows:", len(data))
print("Rows after removing outliers:", len(filtered_data))
```

```
Original rows: 381109
Rows after removing outliers: 370789
```

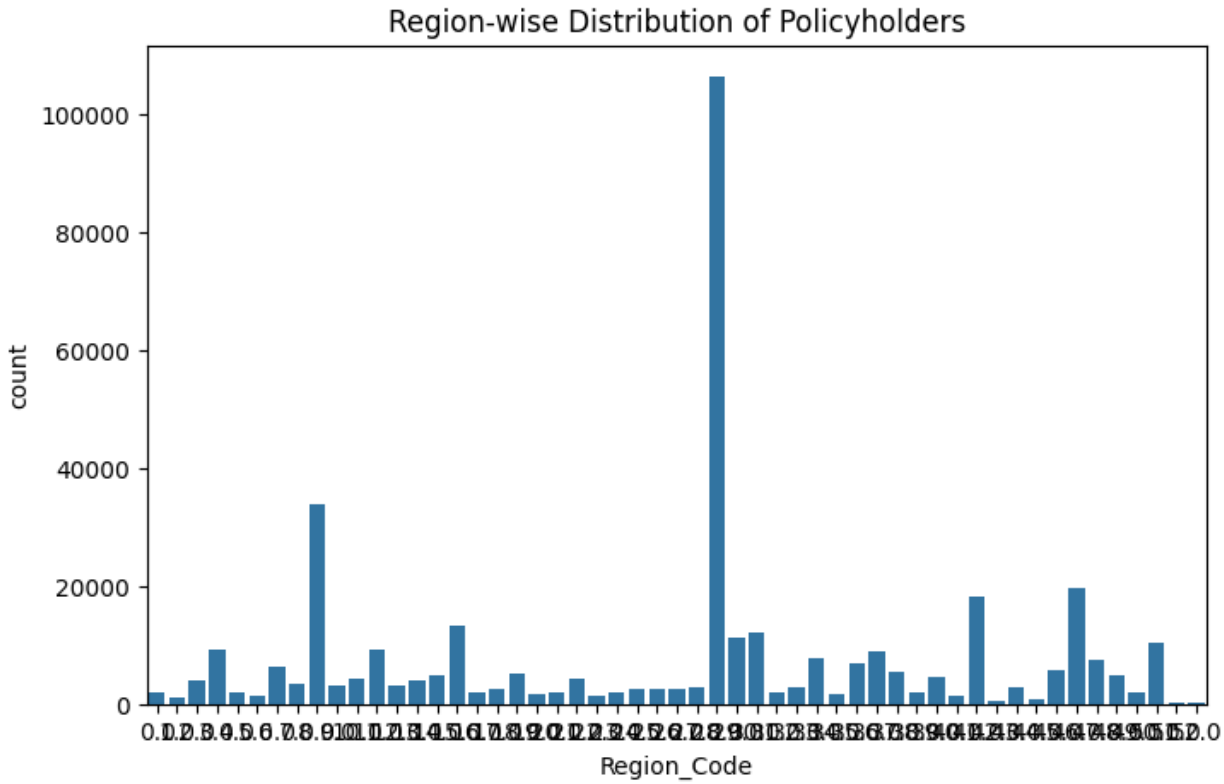
```
# Step 7: Visualizations
# Age Distribution
plt.figure(figsize=(8,5))
sns.histplot(data['Age'], kde=True, bins=20)
plt.title("Age Distribution")
plt.show()
```



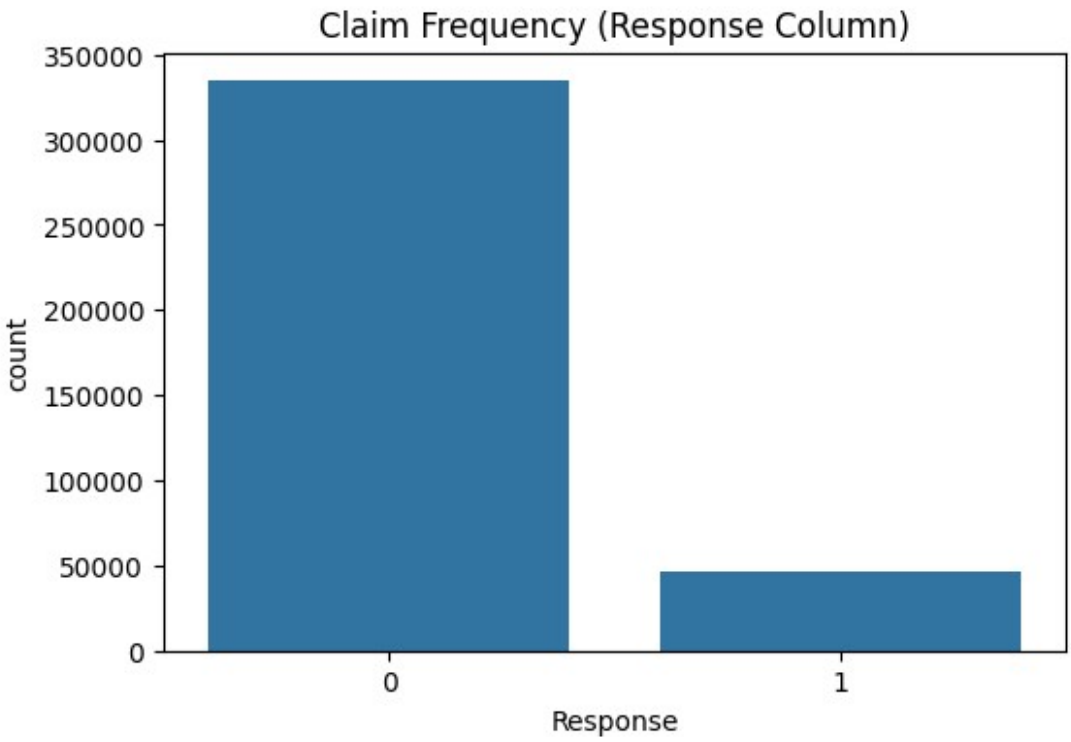

```
# Premium Distribution by Gender
plt.figure(figsize=(8,5))
sns.boxplot(x=data['Gender'], y=data['Annual_Premium'])
plt.title("Annual Premium Distribution by Gender")
plt.show()
```



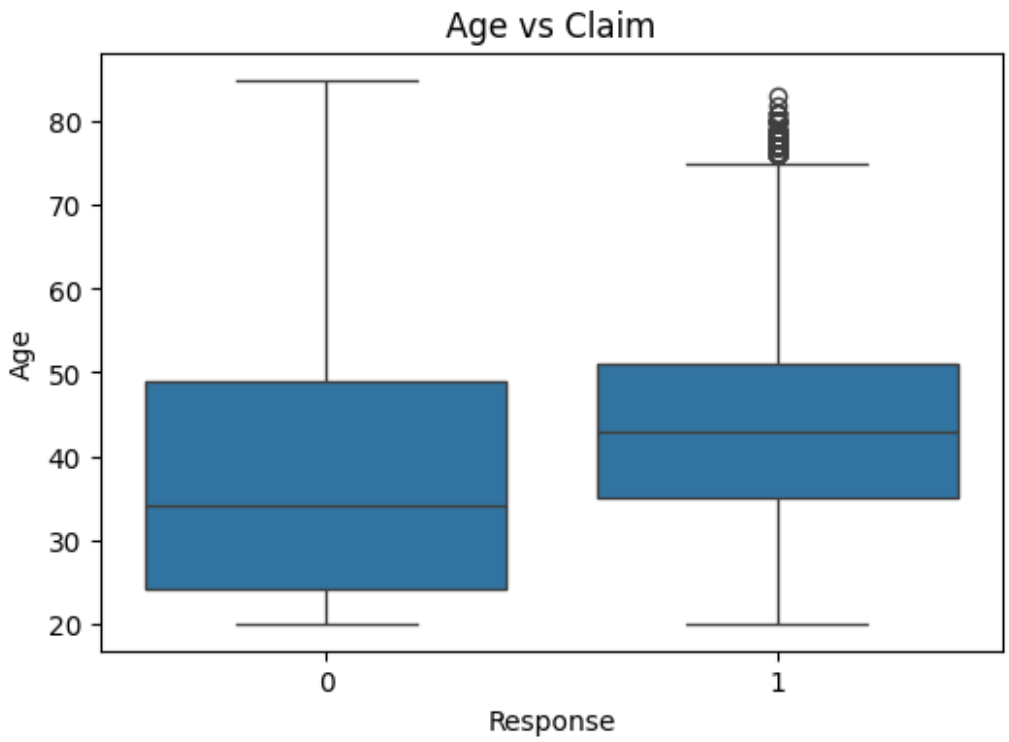

```
# Region-wise Distribution
plt.figure(figsize=(8,5))
sns.countplot(x=data['Region_Code'])
plt.title("Region-wise Distribution of Policyholders")
plt.show()
```



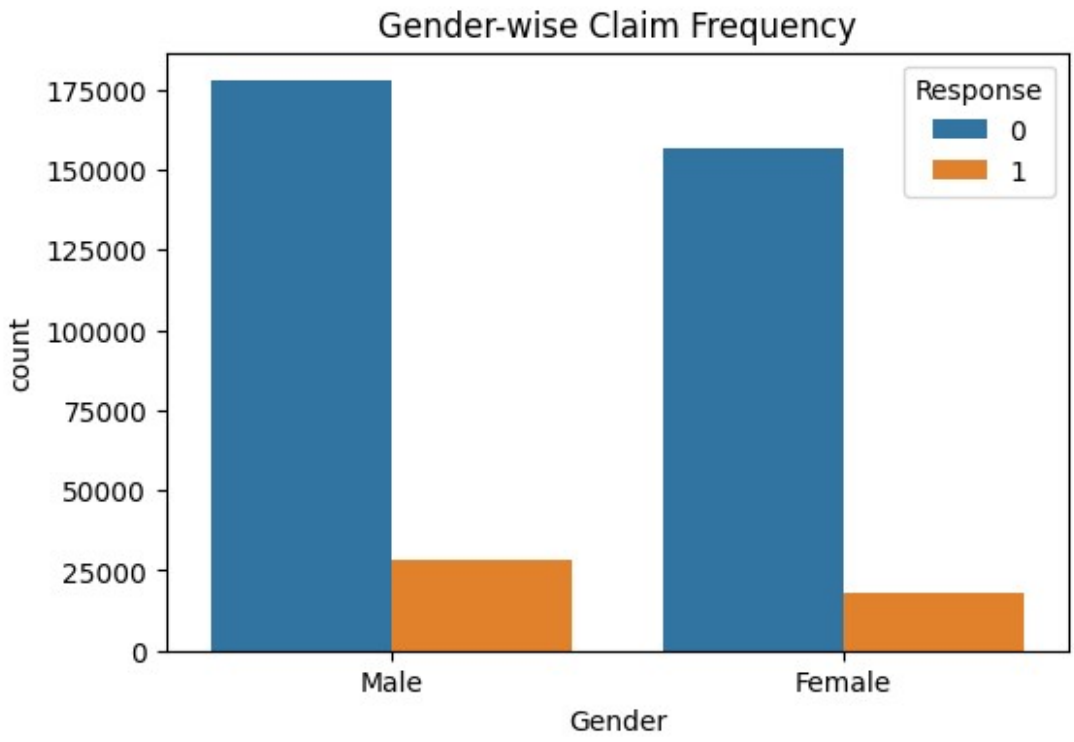

```
# Step 8: Claim Analysis
plt.figure(figsize=(6,4))
sns.countplot(x='Response', data=data)
plt.title("Claim Frequency (Response Column)")
plt.show()
```



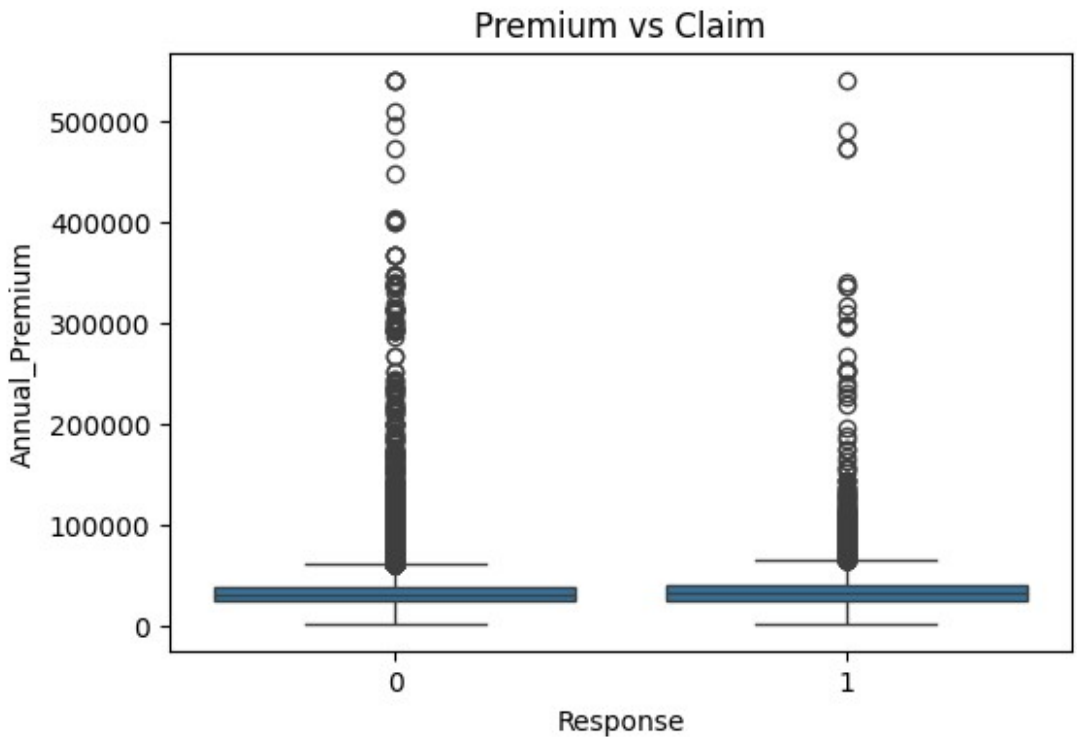

```
# Step 9: Age vs Claim
plt.figure(figsize=(6,4))
sns.boxplot(x='Response', y='Age', data=data)
plt.title("Age vs Claim")
plt.show()
```



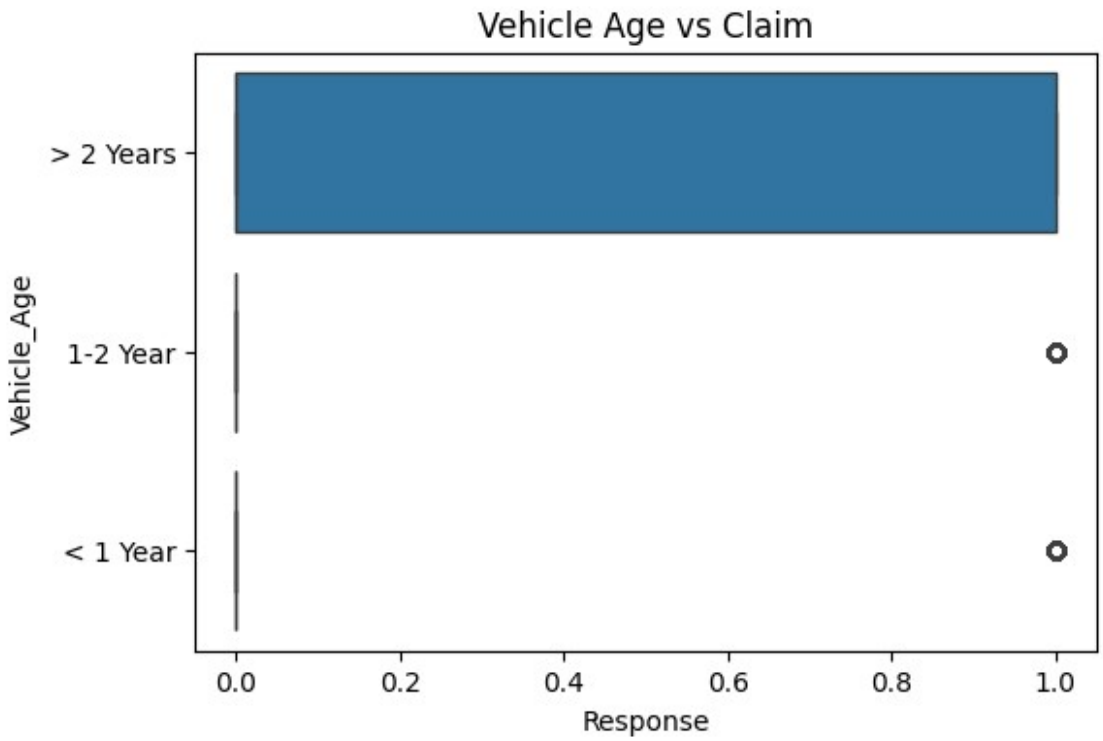

```
# Step 10: Gender vs Claim
plt.figure(figsize=(6,4))
sns.countplot(x='Gender', hue='Response', data=data)
plt.title("Gender-wise Claim Frequency")
plt.show()
```




```
# Step 11: Premium vs Claim
plt.figure(figsize=(6,4))
sns.boxplot(x='Response', y='Annual_Premium', data=data)
plt.title("Premium vs Claim")
plt.show()
```




```
# Step 12: Vehicle Age vs Claim
plt.figure(figsize=(6,4))
sns.boxplot(x='Response', y='Vehicle_Age', data=data)
plt.title("Vehicle Age vs Claim")
plt.show()
```



Summary

1. Younger or older drivers may show different claim rates.
2. Premium amounts might be higher for frequent claimers.
3. Some regions could have more claims than others.
4. Gender and vehicle age also influence claim likelihood.