****EDA Vehicle Insurance project****

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb

data = pd.read_csv("C:\\Users\\Nikhil\\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]

data.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
```

Data Cleaning

```
data.isna().sum()
```

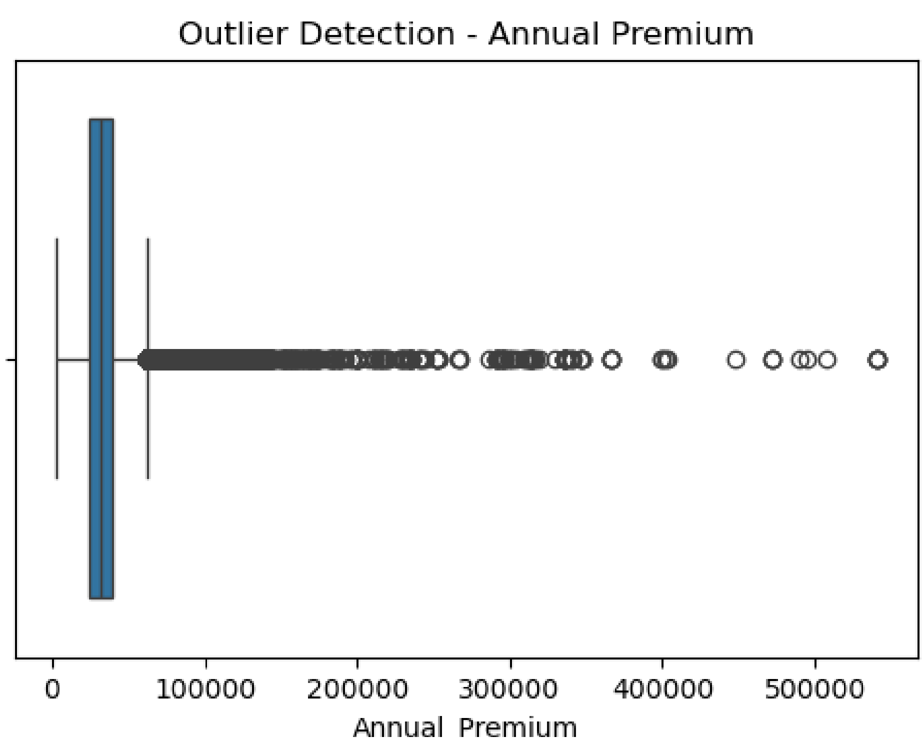
```
id                      0
Gender                  0
Age                     0
Driving_License         0
Region_Code             0
Previously_Insured      0
Vehicle_Age             0
Vehicle_Damage          0
Annual_Premium          0
Policy_Sales_Channel    0
Vintage                 0
Response  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

insight data is clean already as it has no null value

```
data.duplicated().sum()
```

```
np.int64(0)
```

```
#outlier detection
plt.figure(figsize=(6,4))
sb.boxplot(x=data["Annual_Premium"])
plt.title("Outlier Detection - Annual Premium")
plt.show()
```

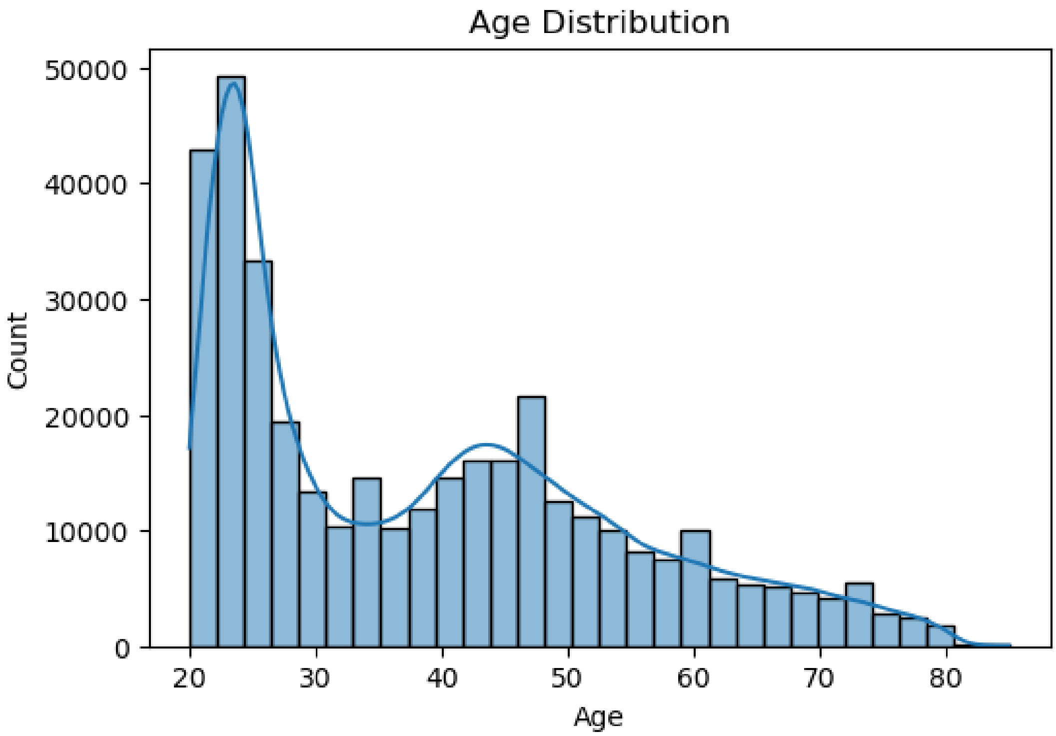


```
Q1 = data["Annual_Premium"].quantile(0.25)
Q3 = data["Annual_Premium"].quantile(0.75)
IQR = Q3 - Q1

lower = Q1 - 1.5 * IQR
upper = Q3 + 1.5 * IQR

data = data[(data["Annual_Premium"] >= lower) &
(data["Annual_Premium"] <= upper)]

data.info()
```

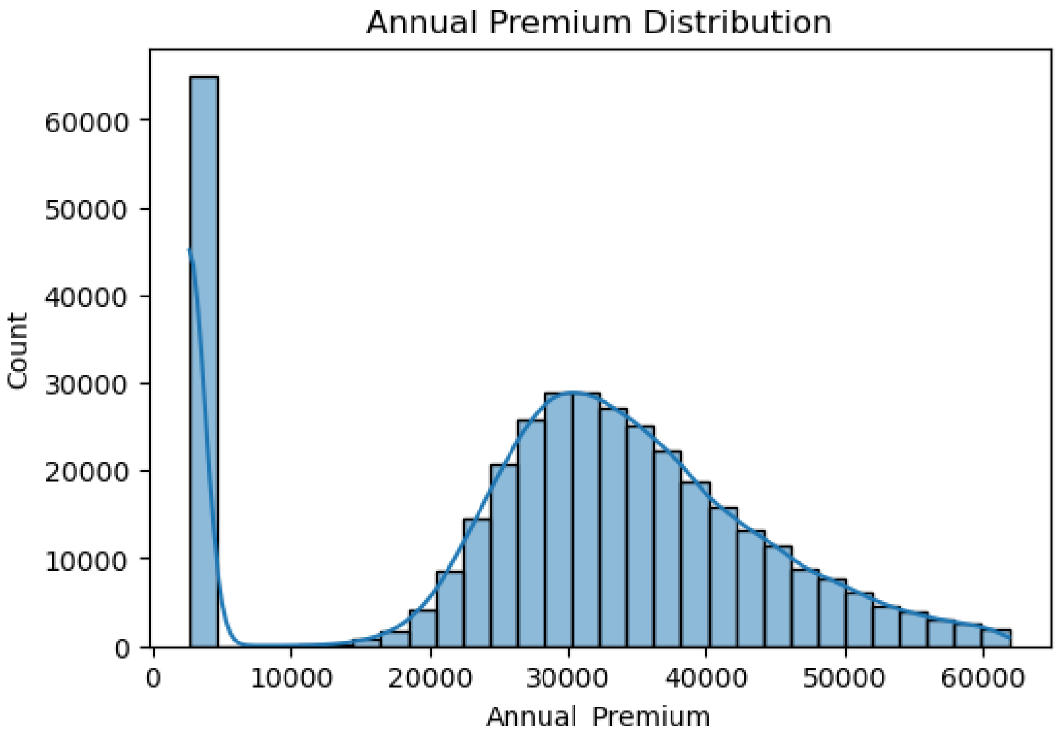
```
<class 'pandas.core.frame.DataFrame'>
Index: 370789 entries, 0 to 381108
Data columns (total 12 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   id                    370789 non-null  int64
 1   Gender                370789 non-null  object
 2   Age                   370789 non-null  int64
 3   Driving_License       370789 non-null  int64
 4   Region_Code           370789 non-null  float64
 5   Previously_Insured    370789 non-null  int64
 6   Vehicle_Age           370789 non-null  object
 7   Vehicle_Damage        370789 non-null  object
 8   Annual_Premium        370789 non-null  float64
 9   Policy_Sales_Channel  370789 non-null  float64
 10  Vintage               370789 non-null  int64
 11  Response              370789 non-null  int64
dtypes: float64(3), int64(6), object(3)
memory usage: 36.8+ MB
```

Data Visualization

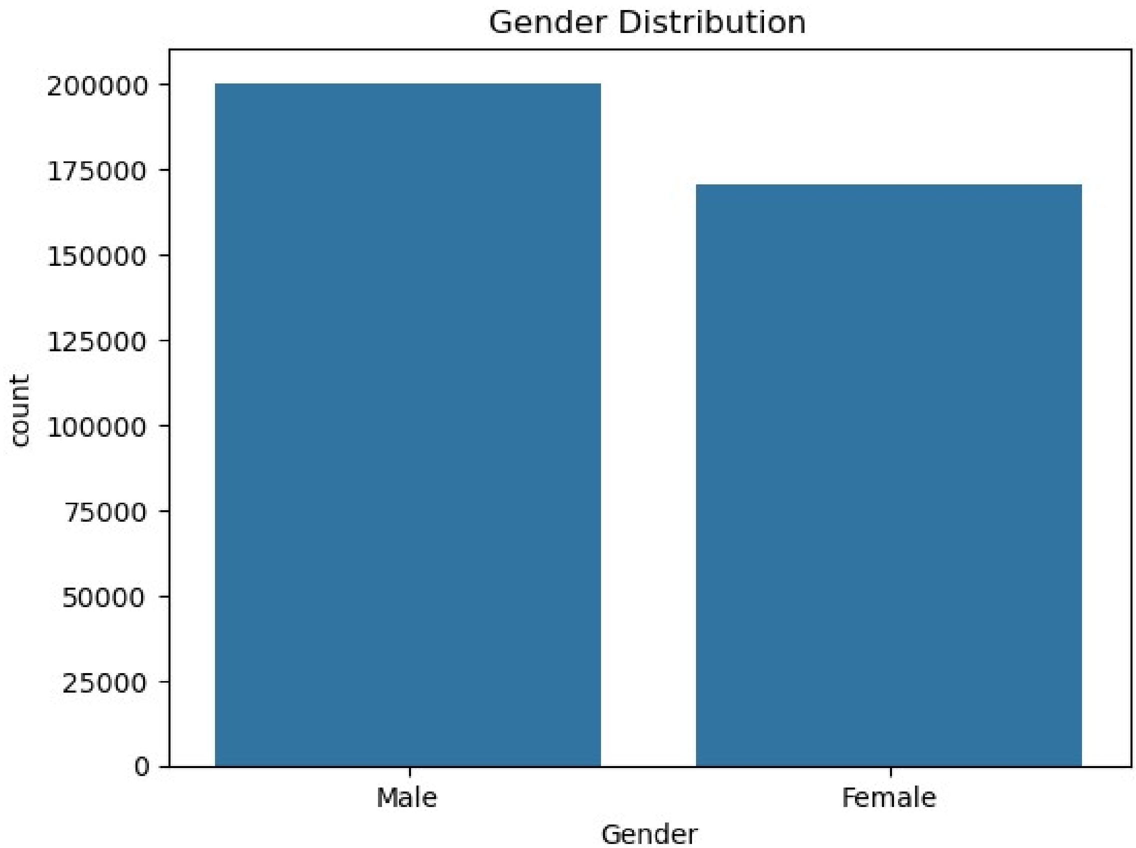
```
#age distriubution
plt.figure(figsize=(6,4))
sb.histplot(data["Age"], bins=30, kde=True)
plt.title("Age Distribution")
plt.show()
```




```
#annual premium distribution
plt.figure(figsize=(6,4))
sb.histplot(data["Annual_Premium"], bins=30, kde=True)
plt.title("Annual Premium Distribution")
plt.show()
```



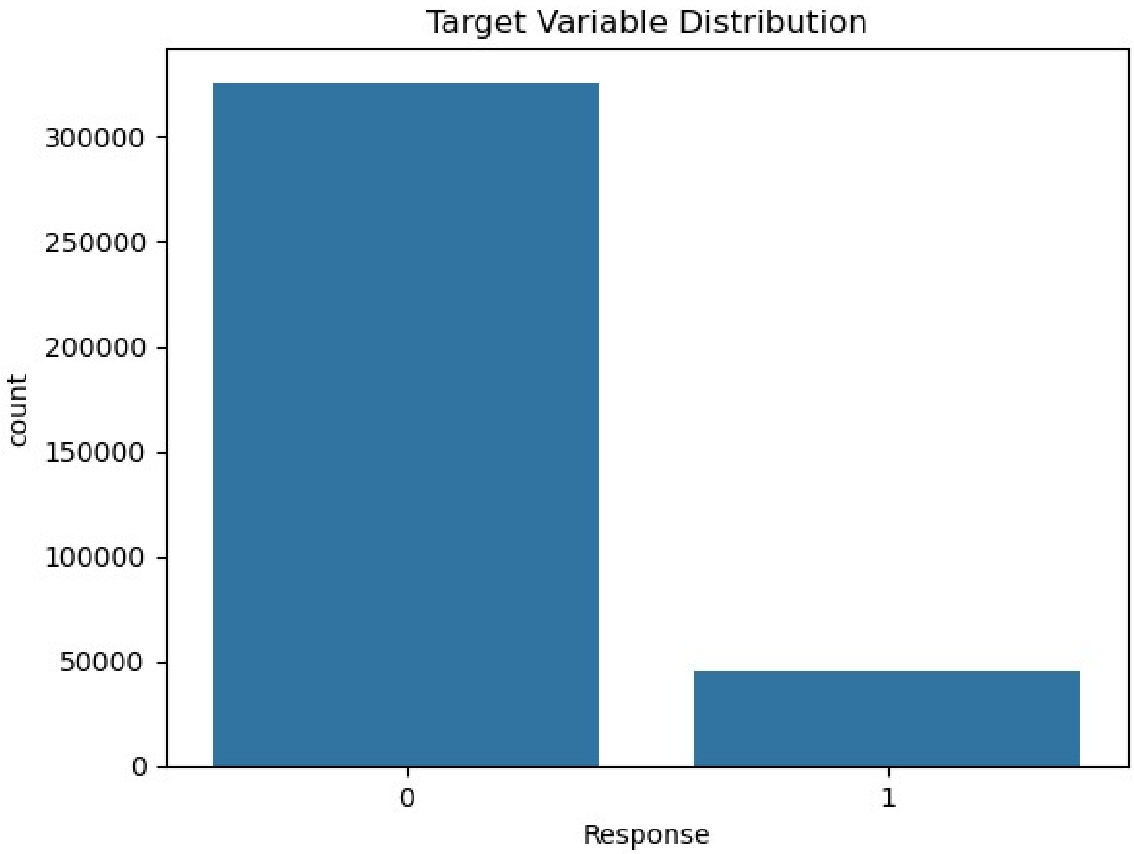

```
#gender distribution
sb.countplot(x="Gender", data=data)
plt.title("Gender Distribution")
plt.show()
```



FEATURE ANALYSIS

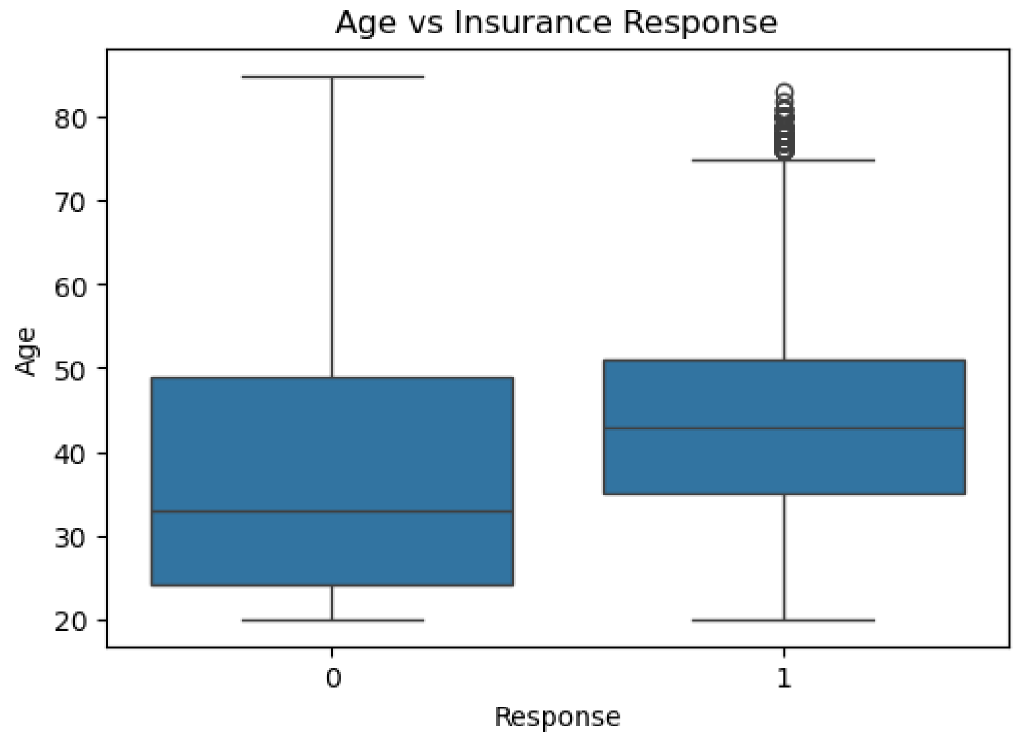
(Relationship between features & target variable – Response)

```
#target Variable Distribution
sb.countplot(x="Response", data=data)
plt.title("Target Variable Distribution")
plt.show()
```



AGE DISTRIBUTION

```
#age vs response
plt.figure(figsize=(6,4))
sb.boxplot(x="Response", y="Age", data=data)
plt.title("Age vs Insurance Response")
plt.show()
```

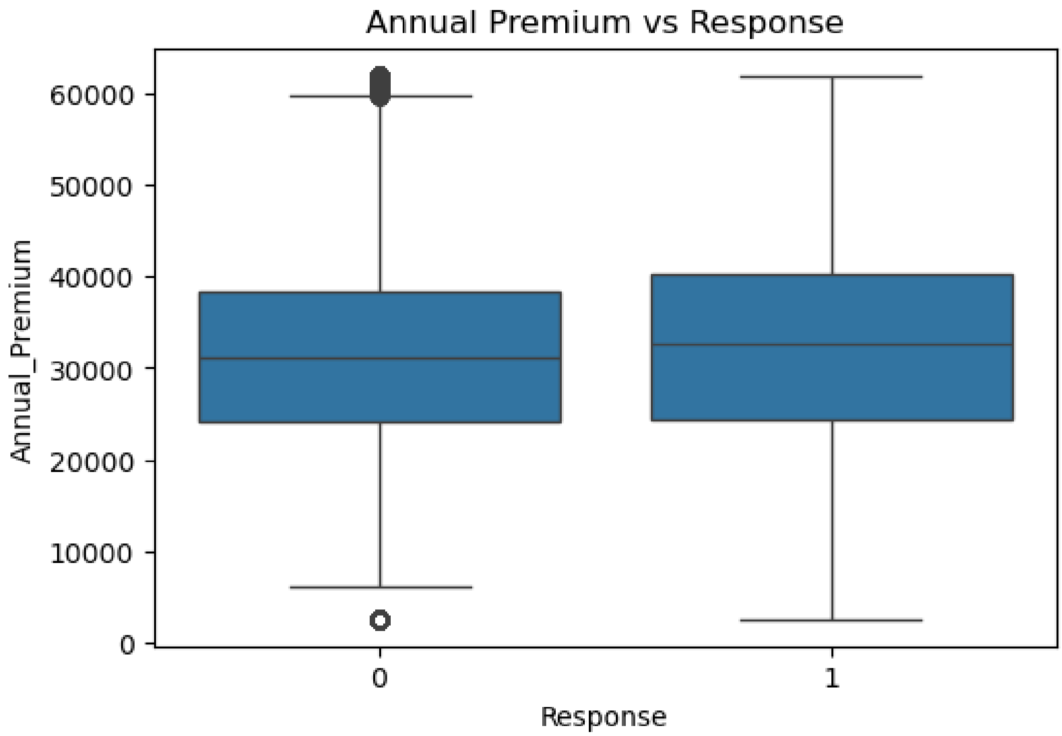



```
data.groupby("Response")["Age"].mean()

Response
0    38.032494
1    43.270180
Name: Age, dtype: float64
```

Insight: Older customers are more likely to respond to insurance offers.

PREMIUM ANALYSIS

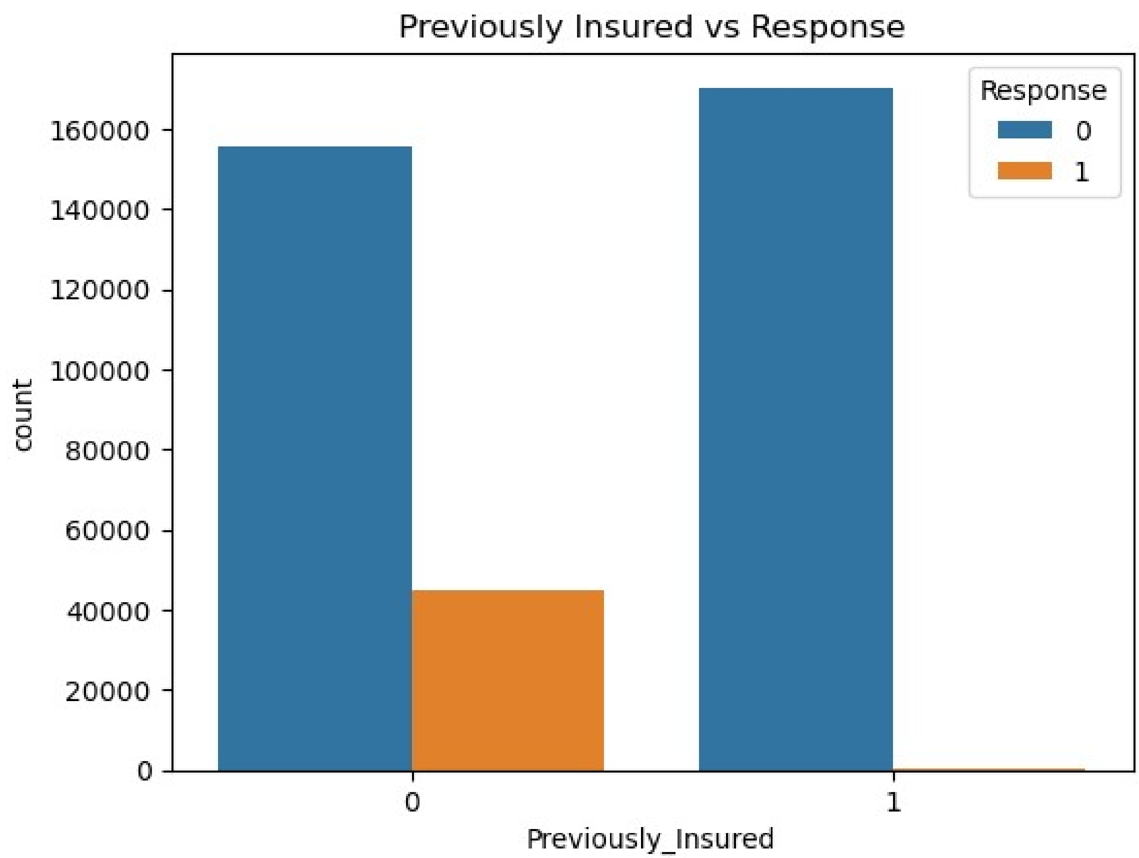
```
#premium vs response
plt.figure(figsize=(6,4))
sb.boxplot(x="Response", y="Annual_Premium", data=data)
plt.title("Annual Premium vs Response")
plt.show()
```




```
data.groupby("Response")["Annual_Premium"].mean()

Response
0    29162.717548
1    29999.683490
Name: Annual_Premium, dtype: float64
```

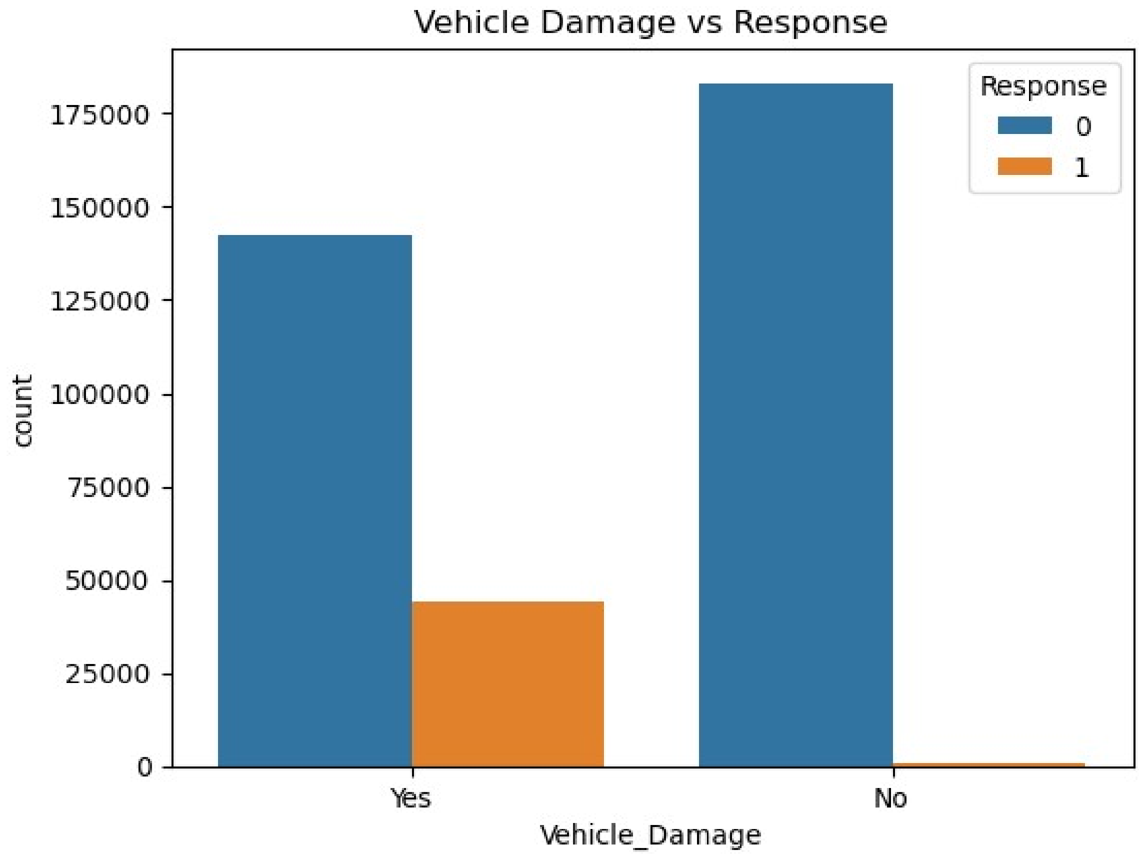
CLAIM FREQUENCIES

```
#previously insured vs response
sb.countplot(x="Previously_Insured", hue="Response", data=data)
plt.title("Previously Insured vs Response")
plt.show()
```



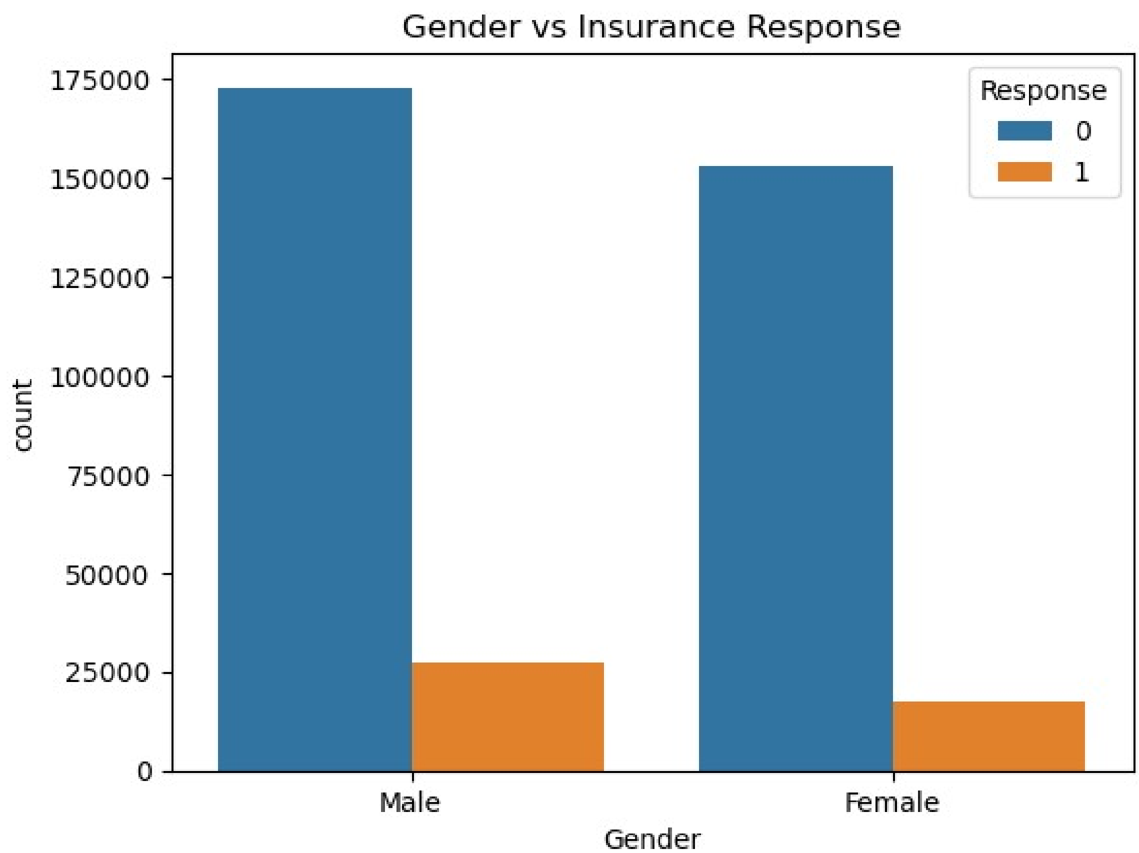
Insight: Customers not previously insured are far more likely to respond.

```
#Vehicle Damage vs Response
sb.countplot(x="Vehicle_Damage", hue="Response", data=data)
plt.title("Vehicle Damage vs Response")
plt.show()
```



Gender analysis

```
#Role of gender in insurance claims
sb.countplot(x="Gender", hue="Response", data=data)
plt.title("Gender vs Insurance Response")
plt.show()
```



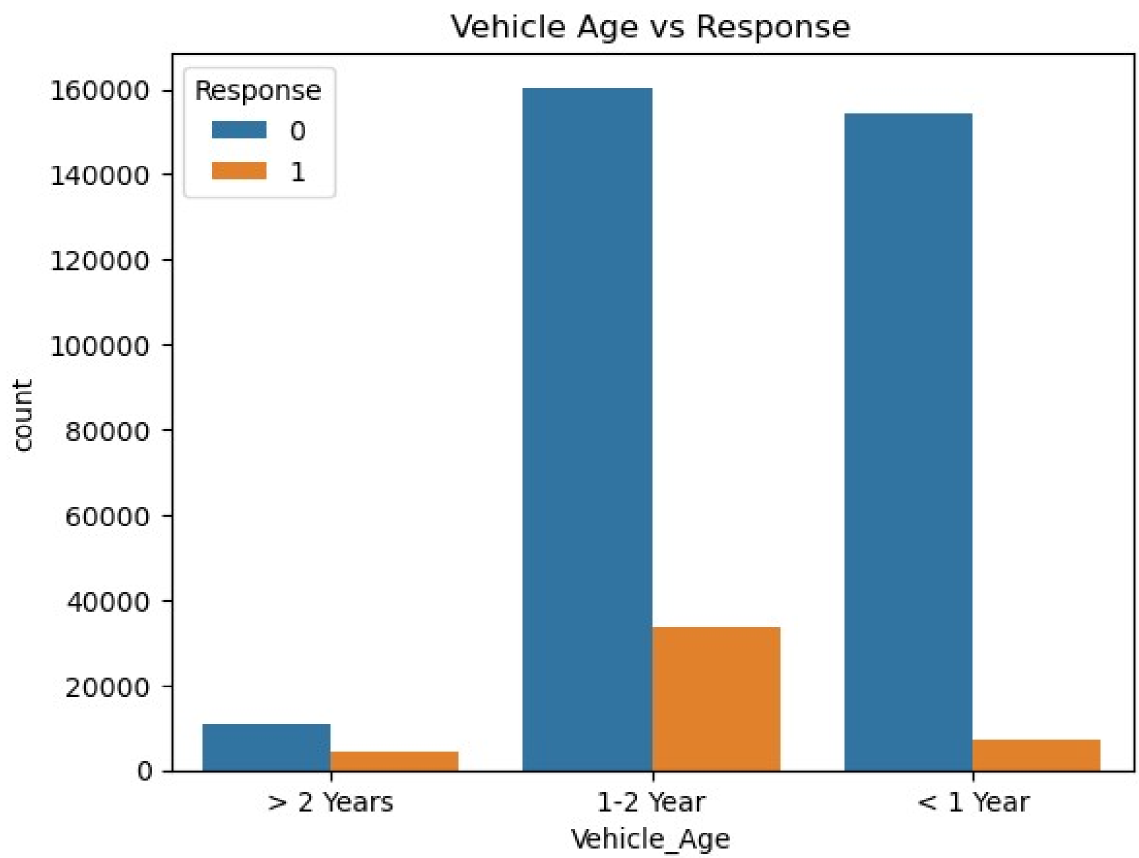

```
data.groupby("Gender")["Response"].mean()
```

```
Gender
Female    0.103238
Male      0.137561
Name: Response, dtype: float64
```

Insight: Males show slightly higher response rates than females.

VEHICLE AGE & CLAIMS

```
#Vehicle Age vs Response
sb.countplot(x="Vehicle_Age", hue="Response", data=data)
plt.title("Vehicle Age vs Response")
plt.show()
```

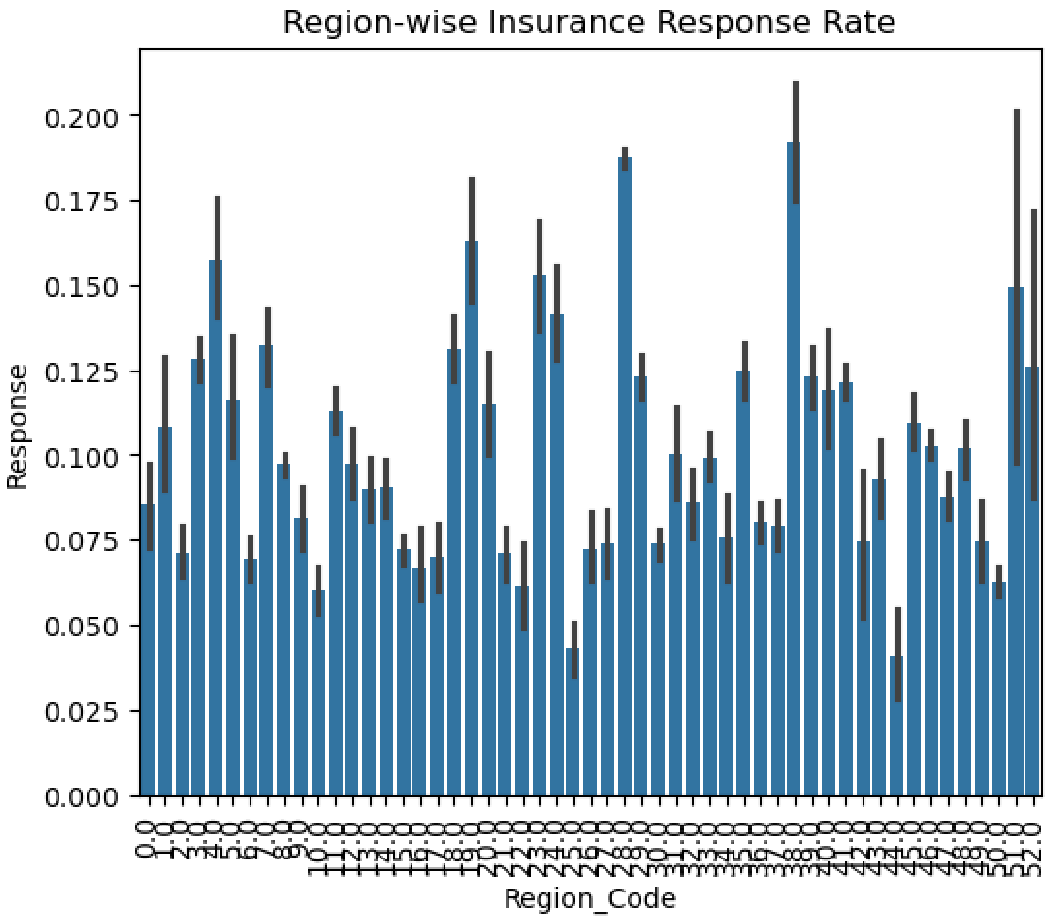



```
data.groupby("Vehicle_Age")["Response"].mean()
```

```
Vehicle_Age
1-2 Year     0.173753
< 1 Year     0.043702
> 2 Years    0.289421
Name: Response, dtype: float64
```

REGION-WISE ANALYSIS (Analyze regional patterns in insurance claims)

```
#Region vs Response
plt.figure(figsize=(6,5))
sb.barplot(
    x="Region_Code",
    y="Response",
    data=data,
    estimator=np.mean
)
plt.title("Region-wise Insurance Response Rate")
plt.xticks(rotation=90)
plt.show()
```

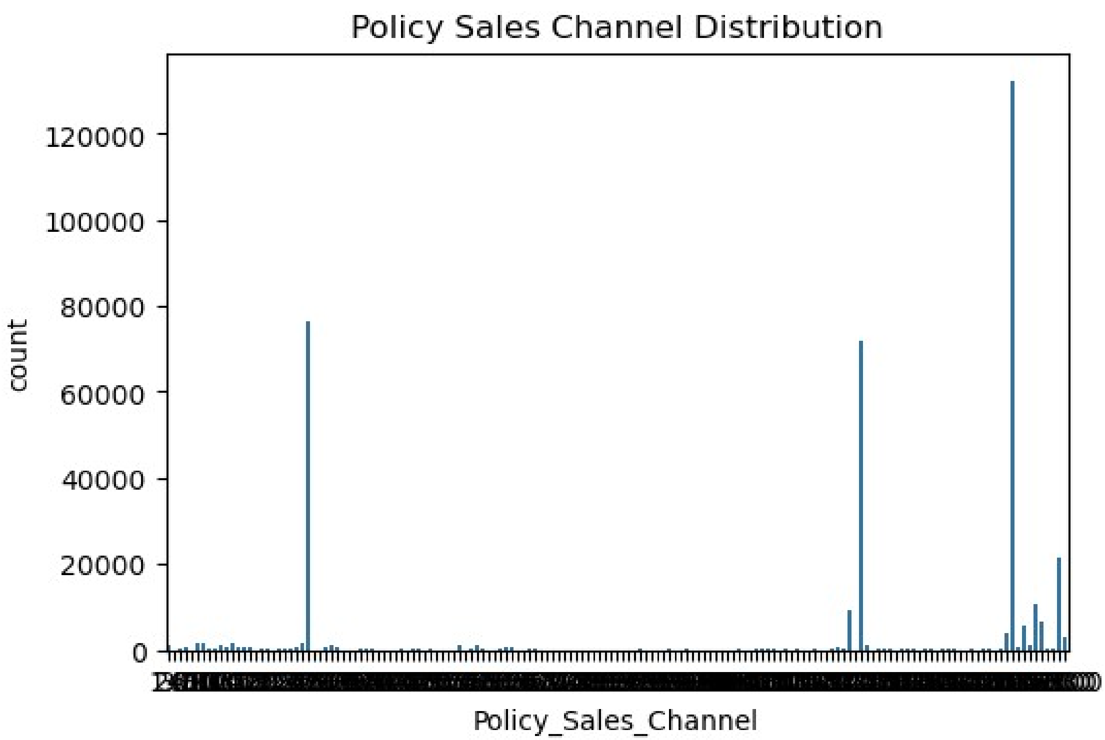



```
data.groupby("Region_Code")
["Response"].mean().sort_values(ascending=False).head(10)

Region_Code
38.0    0.192423
28.0    0.187526
19.0    0.162973
4.0     0.157572
23.0    0.152707
51.0    0.149425
24.0    0.141611
7.0     0.132260
18.0    0.130987
3.0     0.128325
Name: Response, dtype: float64
```

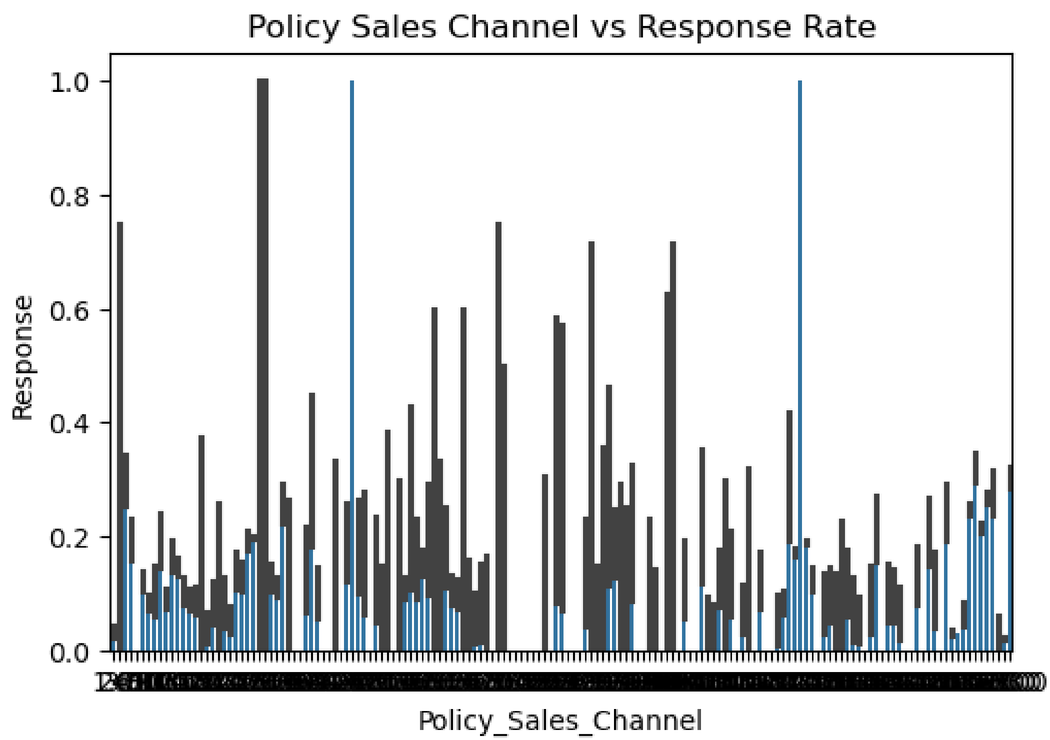
Insight: Certain regions show significantly higher response rates, useful for targeted marketing.

POLICY ANALYSIS (Distribution & impact of different policy types)

```
#Policy Sales Channel Distribution
plt.figure(figsize=(6,4))
sb.countplot(x="Policy_Sales_Channel", data=data)
plt.title("Policy Sales Channel Distribution")
plt.show()
```




```
#Policy Sales Channel vs Response
plt.figure(figsize=(6,4))
sb.barplot(
    x="Policy_Sales_Channel",
    y="Response",
    data=data,
    estimator=np.mean
)
plt.title("Policy Sales Channel vs Response Rate")
plt.show()
```



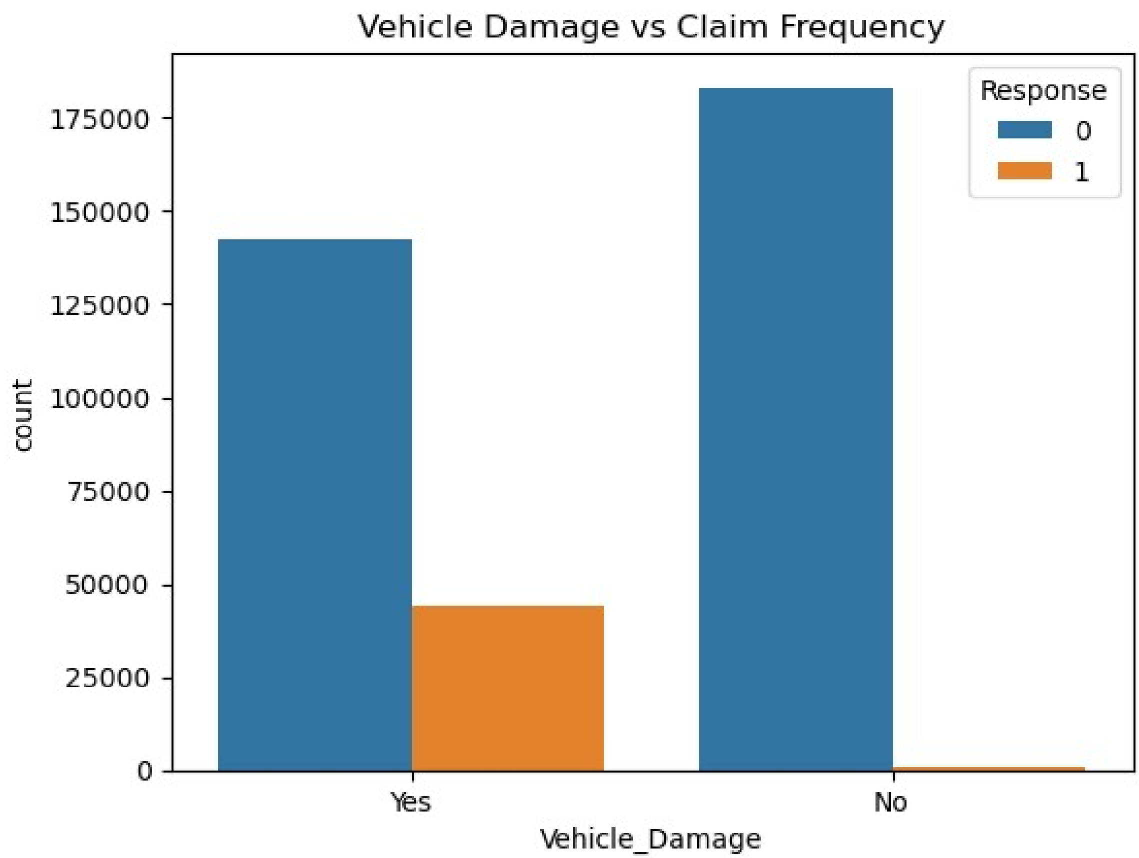

```
data.groupby("Policy_Sales_Channel")["Response"].mean()
```

```
Policy_Sales_Channel
1.0      0.032588
2.0      0.250000
3.0      0.298354
4.0      0.193939
6.0      0.000000
           ...
157.0    0.268072
158.0    0.277207
159.0    0.019608
160.0    0.021918
163.0    0.303571
Name: Response, Length: 155, dtype: float64
```

Insight: Certain sales channels outperform others in customer conversions.

CLAIM FREQUENCY BY VEHICLE DAMAGE

```
sb.countplot(
    x="Vehicle_Damage",
    hue="Response",
    data=data
)
plt.title("Vehicle Damage vs Claim Frequency")
plt.show()
```



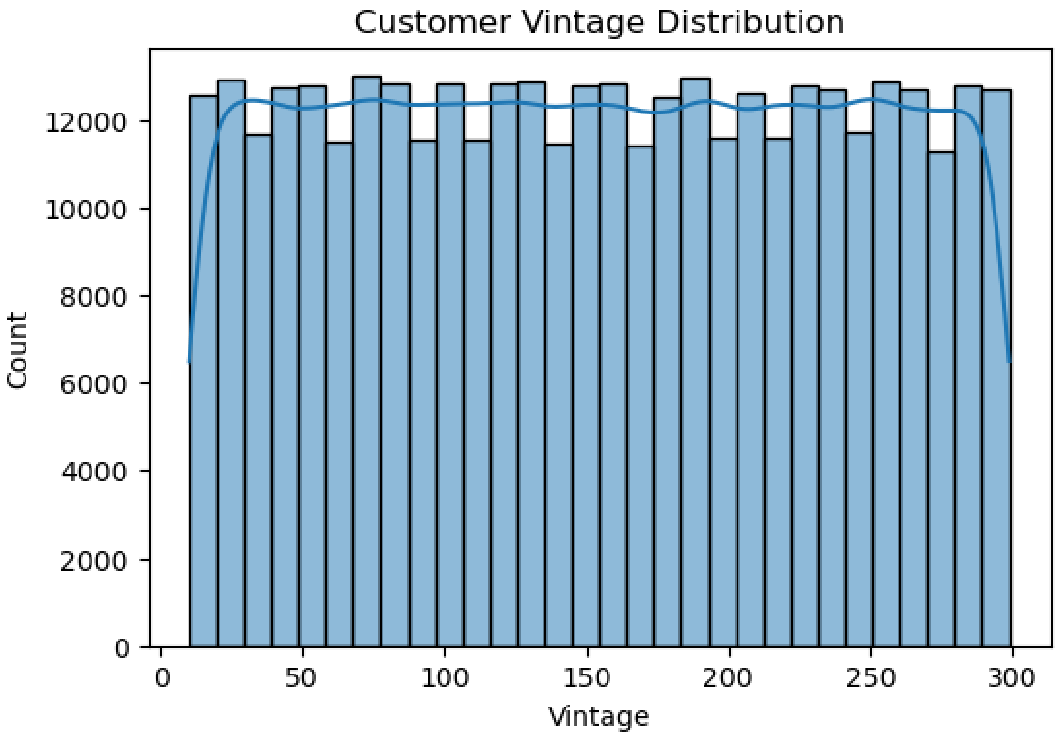

```
data.groupby("Vehicle_Damage")["Response"].mean()
```

```
Vehicle_Damage
No     0.005249
Yes    0.236856
Name: Response, dtype: float64
```

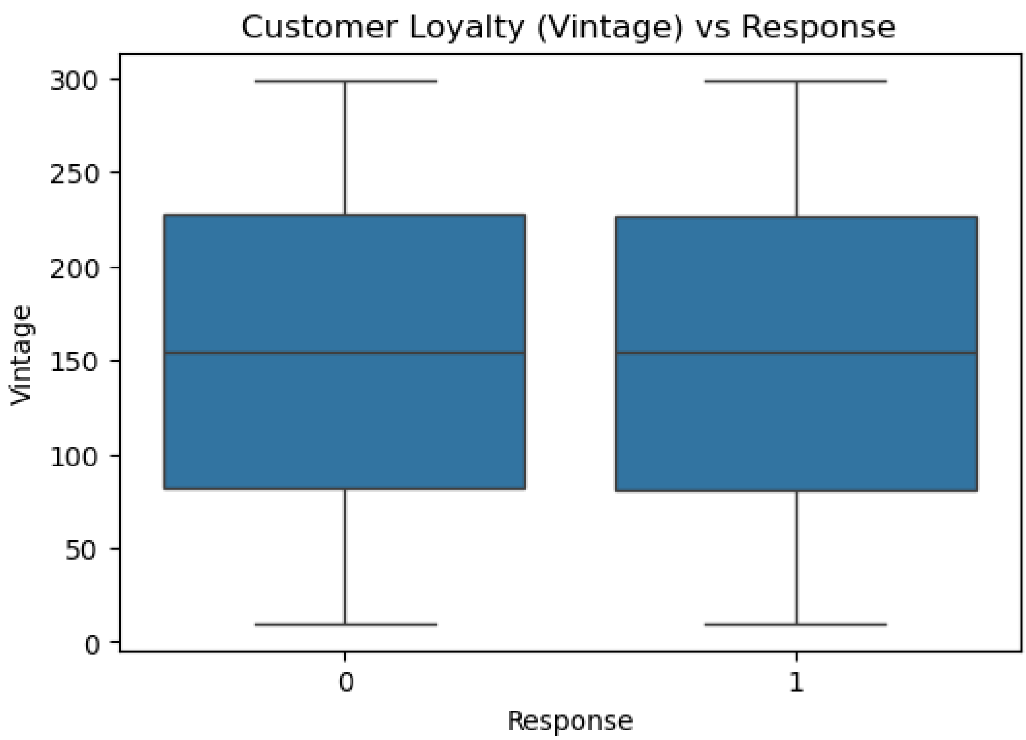
Insight: Customers with vehicle damage = Yes have a much higher probability of filing claims / responding.

CUSTOMER LOYALTY ANALYSIS

```
#Vintage Distribution
plt.figure(figsize=(6,4))
sb.histplot(data["Vintage"], bins=30, kde=True)
plt.title("Customer Vintage Distribution")
plt.show()
```




```
#Vintage vs Response
plt.figure(figsize=(6,4))
sb.boxplot(x="Response", y="Vintage", data=data)
plt.title("Customer Loyalty (Vintage) vs Response")
plt.show()
```




```
data.groupby("Response")["Vintage"].mean()

Response
0    154.396261
1    153.978961
Name: Vintage, dtype: float64
```

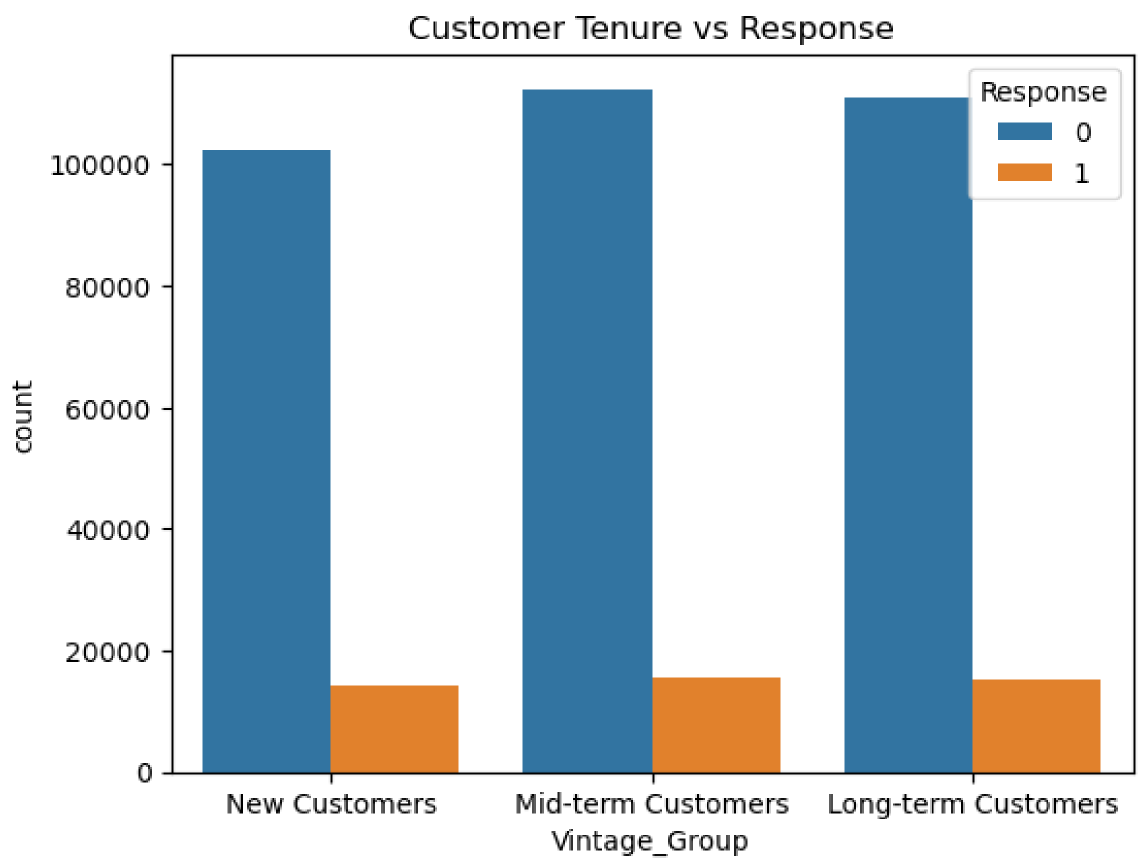
Insight: Long-term customers show slightly higher trust, but newer customers respond more aggressively to offers.

TIME ANALYSIS (Temporal patterns in insurance claims)

```
#Vintage Bins (Customer Age Buckets)
data["Vintage_Group"] = pd.cut(
    data["Vintage"],
    bins=[0, 100, 200, 300],
    labels=["New Customers", "Mid-term Customers", "Long-term
Customers"]
)

C:\Users\Nikhil\AppData\Local\Temp\ipykernel_7252\2520873319.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
```

```
returning-a-view-versus-a-copy
  data["Vintage_Group"] = pd.cut(
```

```
#Vintage Group vs Response
sb.countplot(
    x="Vintage_Group",
    hue="Response",
    data=data
)
plt.title("Customer Tenure vs Response")
plt.show()
```




```
data.groupby("Vintage_Group")["Response"].mean()
```

```
C:\Users\Nikhil\AppData\Local\Temp\ipykernel_7252\2877602394.py:1:
FutureWarning: The default of observed=False is deprecated and will be
changed to True in a future version of pandas. Pass observed=False to
retain current behavior or observed=True to adopt the future default
and silence this warning.
  data.groupby("Vintage_Group")["Response"].mean()
```

```
Vintage_Group
New Customers          0.122075
```

```
Mid-term Customers     0.121727
Long-term Customers    0.121564
Name: Response, dtype: float64
```

Insight: Customers in the mid-tenure phase show the highest response rate.

Key summary:

Vehicle damage is the strongest predictor of claims

Previously uninsured customers are high-value targets

Middle-aged and senior customers respond more

Certain regions and policy channels outperform others

Mid-tenure customers show higher engagement

Business Recommendations:

Focus marketing on damaged-vehicle owners

Target uninsured customers

Optimize high-performing sales channels

Region-specific campaigns can improve conversion