Marketing-Customer-Value-Analysis_EDA-1

October 11, 2024

0.1 Dự đoán phản hồi của khách hàng - Marketing-Customer-Value-Analysis (2.5 điểm)

Cung cấp bộ dữ liệu Marketing-Customer-Value-Analysis.csv chứa thông tin khách hàng liên quan đến việc bán bảo hiểm xe hơi, bao gồm: Response (target), các cột còn lại chứa thông tin: 1. Customer: ID của khách hàng 2. State: Bang mà khách hàng cư trú. 3. Customer Lifetime Value: Giá trị trọn đời, một chỉ số quan trọng để đánh giá mức độ đóng góp của khách hàng đối với công ty trong suốt mối quan hệ của họ. 4. Response: Cho biết khách hàng có phản hồi đối với một chiến dịch tiếp thị cụ thể hay không ('Yes' hoặc 'No'). 5. Coverage: Mức độ bảo hiểm mà khách hàng đã chọn cho hợp đồng bảo hiểm của họ (ví dụ: 'Basic', 'Extended', 'Premium'). 6. Education: Trình độ học vấn. 7. Effective To Date: Ngày kết thúc hiệu lực của hợp đồng bảo hiểm. 8. EmploymentStatus: Tình trạng việc làm hiện tại. 9. Gender: Giới tính. 10. Income: Thu nhập hàng năm. 11. Location Code: Phân loại khu vực địa lý nơi khách hàng sinh sống (ví dụ: 'Urban', 'Suburban', 'Rural'). 12. Marital Status: Tình trạng hôn nhân. 13. Monthly Premium Auto: Phí bảo hiểm xe hàng tháng mà khách hàng phải trả. 14. Months Since Last Claim: Số tháng kể từ lần yêu cầu bồi thường bảo hiểm gần đây nhất. 15. Months Since Policy Inception: Số tháng kể từ khi hợp đồng bảo hiểm có hiệu lực. 16. Number of Open Complaints: Số lượng khiếu nại đang mở. 17. Number of Policies: Số lượng hợp đồng bảo hiểm mà khách hàng hiện có với công ty. 18. Policy Type: Loại hợp đồng bảo hiểm (ví dụ: 'Corporate Auto', 'Personal Auto'). 19. Policy: Tên cụ thể của hợp đồng bảo hiểm. 20.Renew Offer Type: Loại đề nghị gia hạn hợp đồng bảo hiểm được cung cấp cho khách hàng. 21. Sales Channel: Kênh bán hàng mà qua đó khách hàng đã mua hợp đồng bảo hiểm. 22. Total Claim Amount: Tổng số tiền yêu cầu bồi thường bảo hiểm của khách hàng. 23. Vehicle Class: Loại xe mà khách hàng sở hữu. 24. Vehicle Size: Kích thước xe. ##### Chú ý: Cần lựa chọn các thuộc tính phù hợp khi đưa vào build model

0.1.1 Import libraries

[2]:
```
import pandas as pd
import numpy as np
import dataprep
import matplotlib.pyplot as plt
import seaborn as sns
```

[3]:
```
data = pd.read_csv("Marketing-Customer-Value-Analysis.csv")
```

[5]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9134 entries, 0 to 9133
```

```
Data columns (total 24 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   Customer                       9134 non-null   object
 1   State                          9134 non-null   object
 2   Customer Lifetime Value        9134 non-null   float64
 3   Response                       9134 non-null   object
 4   Coverage                       9134 non-null   object
 5   Education                      9134 non-null   object
 6   Effective To Date              9134 non-null   object
 7   EmploymentStatus               9134 non-null   object
 8   Gender                         9134 non-null   object
 9   Income                         9134 non-null   int64
 10  Location Code                  9134 non-null   object
 11  Marital Status                 9134 non-null   object
 12  Monthly Premium Auto           9134 non-null   int64
 13  Months Since Last Claim        9134 non-null   int64
 14  Months Since Policy Inception  9134 non-null   int64
 15  Number of Open Complaints      9134 non-null   int64
 16  Number of Policies             9134 non-null   int64
 17  Policy Type                    9134 non-null   object
 18  Policy                         9134 non-null   object
 19  Renew Offer Type               9134 non-null   object
 20  Sales Channel                  9134 non-null   object
 21  Total Claim Amount             9134 non-null   float64
 22  Vehicle Class                  9134 non-null   object
 23  Vehicle Size                   9134 non-null   object
dtypes: float64(2), int64(6), object(16)
memory usage: 1.7+ MB
```

[6]:
```
data.head()
```

[6]:

	Customer	State	Customer Lifetime Value	Response	Coverage	Education	\
0	BU79786	Washington	2763.519279	No	Basic	Bachelor	
1	QZ44356	Arizona	6979.535903	No	Extended	Bachelor	
2	AI49188	Nevada	12887.431650	No	Premium	Bachelor	
3	WW63253	California	7645.861827	No	Basic	Bachelor	
4	HB64268	Washington	2813.692575	No	Basic	Bachelor	

	Effective To Date	EmploymentStatus	Gender	Income	…	\
0	2/24/11	Employed	F	56274	…	
1	1/31/11	Unemployed	F	0	…	
2	2/19/11	Employed	F	48767	…	
3	1/20/11	Unemployed	M	0	…	
4	2/3/11	Employed	M	43836	…	

	Months Since Policy Inception	Number of Open Complaints	Number of Policies	\

```
0                              5                          0                   1
1                             42                          0                   8
2                             38                          0                   2
3                             65                          0                   7
4                             44                          0                   1

      Policy Type        Policy  Renew Offer Type  Sales Channel  \
0  Corporate Auto  Corporate L3            Offer1          Agent
1   Personal Auto   Personal L3            Offer3          Agent
2   Personal Auto   Personal L3            Offer1          Agent
3  Corporate Auto  Corporate L2            Offer1    Call Center
4   Personal Auto   Personal L1            Offer1          Agent

  Total Claim Amount  Vehicle Class Vehicle Size
0         384.811147   Two-Door Car      Medsize
1        1131.464935  Four-Door Car      Medsize
2         566.472247   Two-Door Car      Medsize
3         529.881344            SUV      Medsize
4         138.130879  Four-Door Car      Medsize

[5 rows x 24 columns]
```

[7]:
```
# EDA
from dataprep.eda import create_report, plot
```

[8]:
```
# REPORT of DATA
report = create_report(data)
```

```
  0%|
↪              | 0/2802 [00:00<…
C:\Users\user\miniconda3\lib\site-packages\dask\core.py:127: RuntimeWarning:
invalid value encountered in divide
  return func(*(_execute_task(a, cache) for a in args))
C:\Users\user\miniconda3\lib\site-
packages\dataprep\eda\distribution\render.py:274: FutureWarning: The
frame.append method is deprecated and will be removed from pandas in a future
version. Use pandas.concat instead.
  df = df.append(pd.DataFrame({col: [nrows - npresent]}, index=["Others"]))
C:\Users\user\miniconda3\lib\site-
packages\dataprep\eda\distribution\render.py:274: FutureWarning: The
frame.append method is deprecated and will be removed from pandas in a future
version. Use pandas.concat instead.
  df = df.append(pd.DataFrame({col: [nrows - npresent]}, index=["Others"]))
```

[9]:
```
report
```

[9]:

[ ]:
```
# Du lieu khong bi thieu, khong trung
```

[11]:
```
# ANALYZE A DATAFRAME
# Xem moi loai target co bao nhieu mau, co bi lech hay khong?
data[['Customer', 'Response']].groupby('Response').count()
```

[11]:
```
          Customer
Response
No            7826
Yes           1308
```

[12]:
```
data.columns
```

[12]:
```
Index(['Customer', 'State', 'Customer Lifetime Value', 'Response', 'Coverage',
       'Education', 'Effective To Date', 'EmploymentStatus', 'Gender',
       'Income', 'Location Code', 'Marital Status', 'Monthly Premium Auto',
       'Months Since Last Claim', 'Months Since Policy Inception',
       'Number of Open Complaints', 'Number of Policies', 'Policy Type',
       'Policy', 'Renew Offer Type', 'Sales Channel', 'Total Claim Amount',
       'Vehicle Class', 'Vehicle Size'],
      dtype='object')
```

[13]:
```
# Các thuộc tính object
data.select_dtypes(include='object').columns
```

[13]:
```
Index(['Customer', 'State', 'Response', 'Coverage', 'Education',
       'Effective To Date', 'EmploymentStatus', 'Gender', 'Location Code',
       'Marital Status', 'Policy Type', 'Policy', 'Renew Offer Type',
       'Sales Channel', 'Vehicle Class', 'Vehicle Size'],
      dtype='object')
```

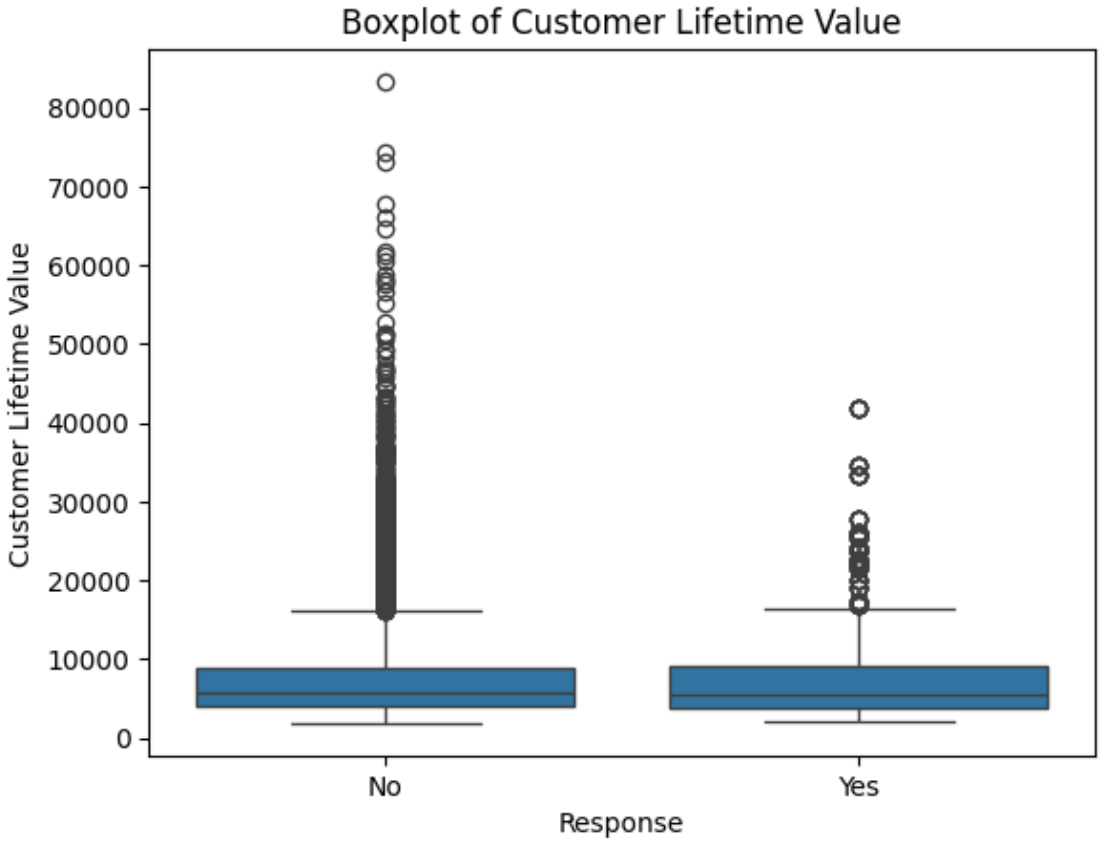
[15]:
```
# Các thuộc tính phân loại
col_cat = ['State', 'Response', 'Coverage', 'Education','Effective To Date',␣
 ↪'EmploymentStatus', 'Gender', 'Location Code',
       'Marital Status', 'Policy Type', 'Policy', 'Renew Offer Type', 'Sales␣
 ↪Channel', 'Vehicle Class', 'Vehicle Size']
```

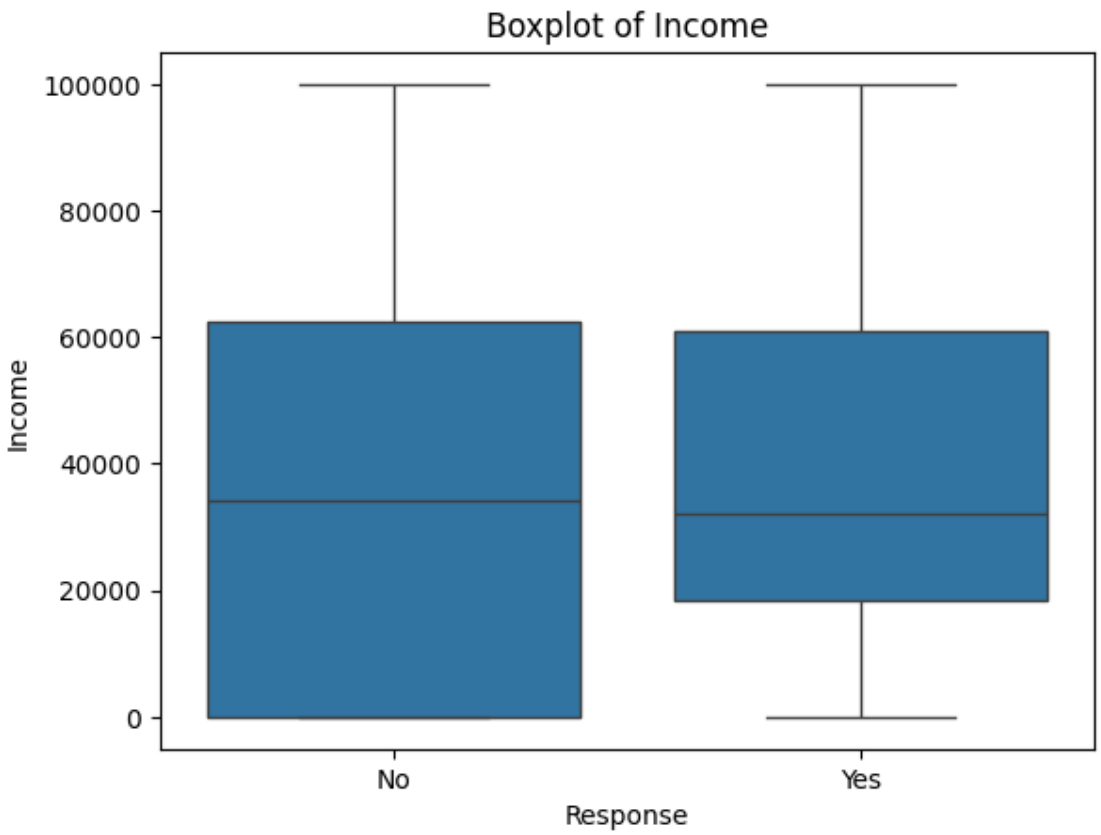
[16]:
```
# Các thuộc tính kiểu số thực
data.select_dtypes(['number']).columns
```

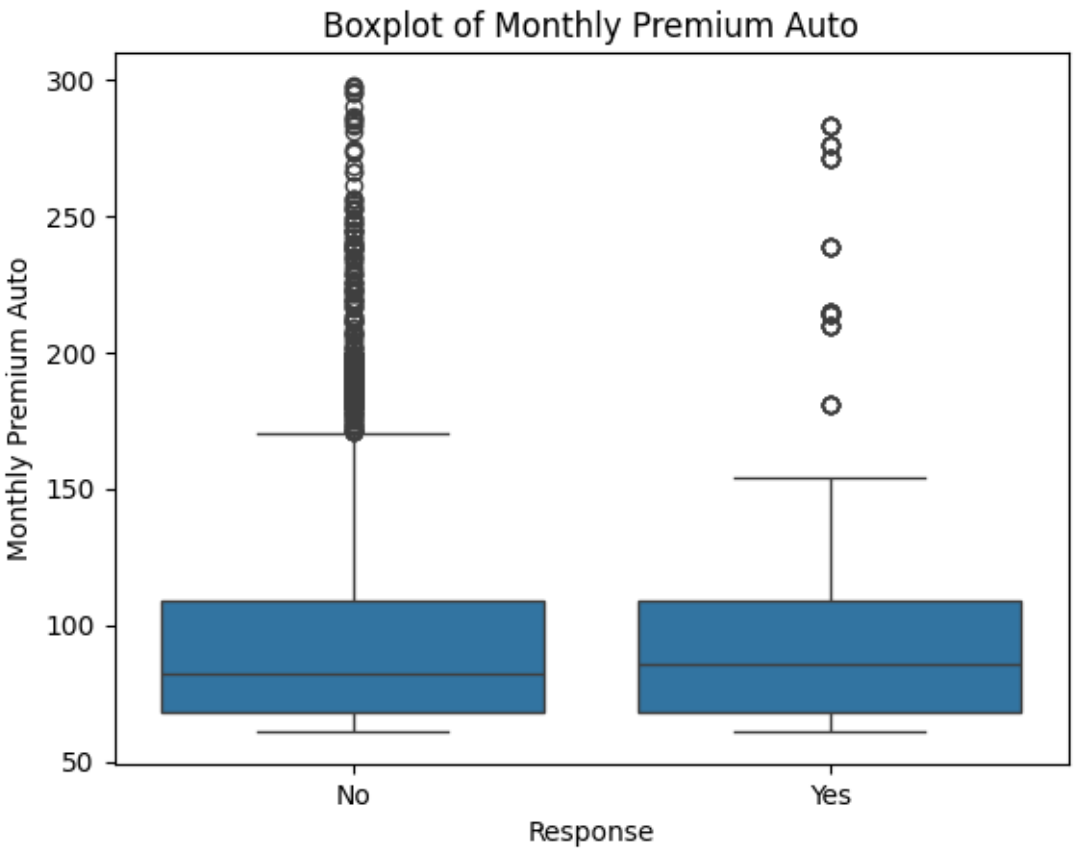
[16]:
```
Index(['Customer Lifetime Value', 'Income', 'Monthly Premium Auto',
       'Months Since Last Claim', 'Months Since Policy Inception',
       'Number of Open Complaints', 'Number of Policies',
       'Total Claim Amount'],
      dtype='object')
```

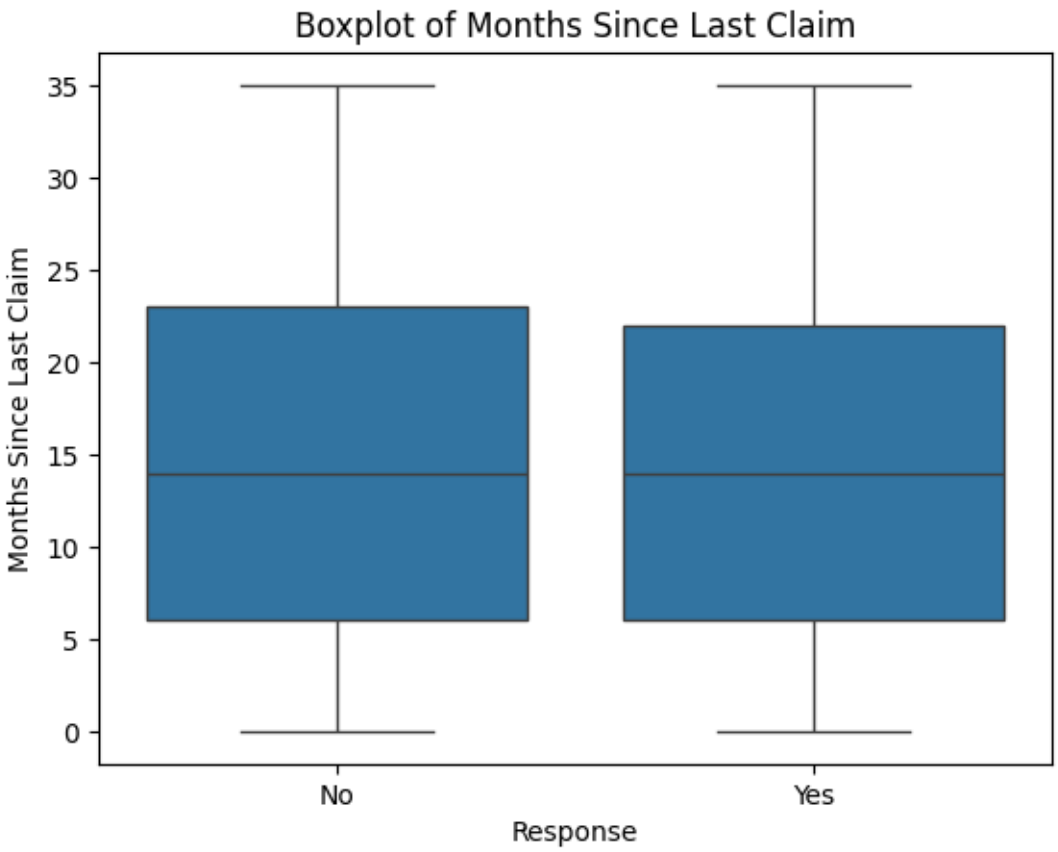
[17]:
```
col_num = ['Customer Lifetime Value', 'Income', 'Monthly Premium Auto',
       'Months Since Last Claim', 'Months Since Policy Inception',
       'Number of Open Complaints', 'Number of Policies',
       'Total Claim Amount']
```

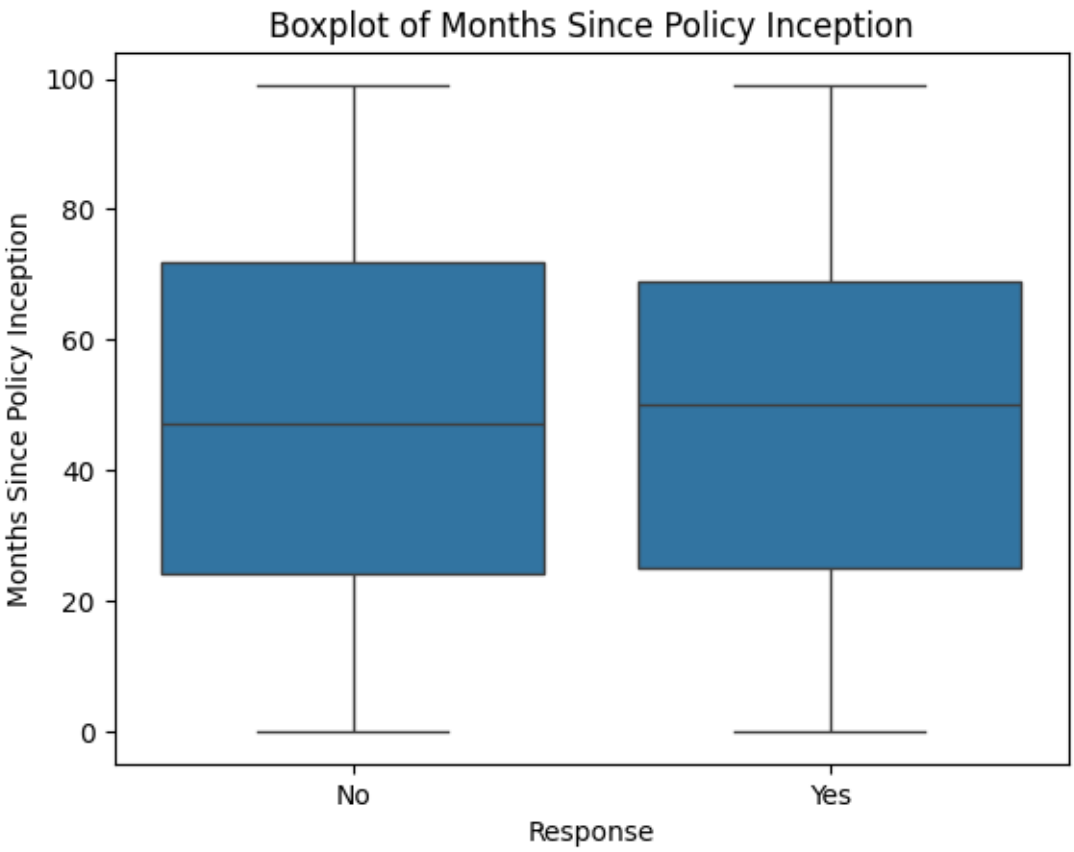
[18]:
```
# Continuous features to visualize
# cột numeric vs categorical -> boxplot
continuous_features = ['Customer Lifetime Value', 'Income', 'Monthly Premium␣
 ↪Auto',
       'Months Since Last Claim', 'Months Since Policy Inception',
       'Number of Open Complaints', 'Number of Policies',
       'Total Claim Amount']
# Plotting boxplots for each continuous feature
for feature in continuous_features:
    sns.boxplot(x='Response', y=feature, data=data)
    plt.title(f"Boxplot of {feature}")
    plt.show()
```

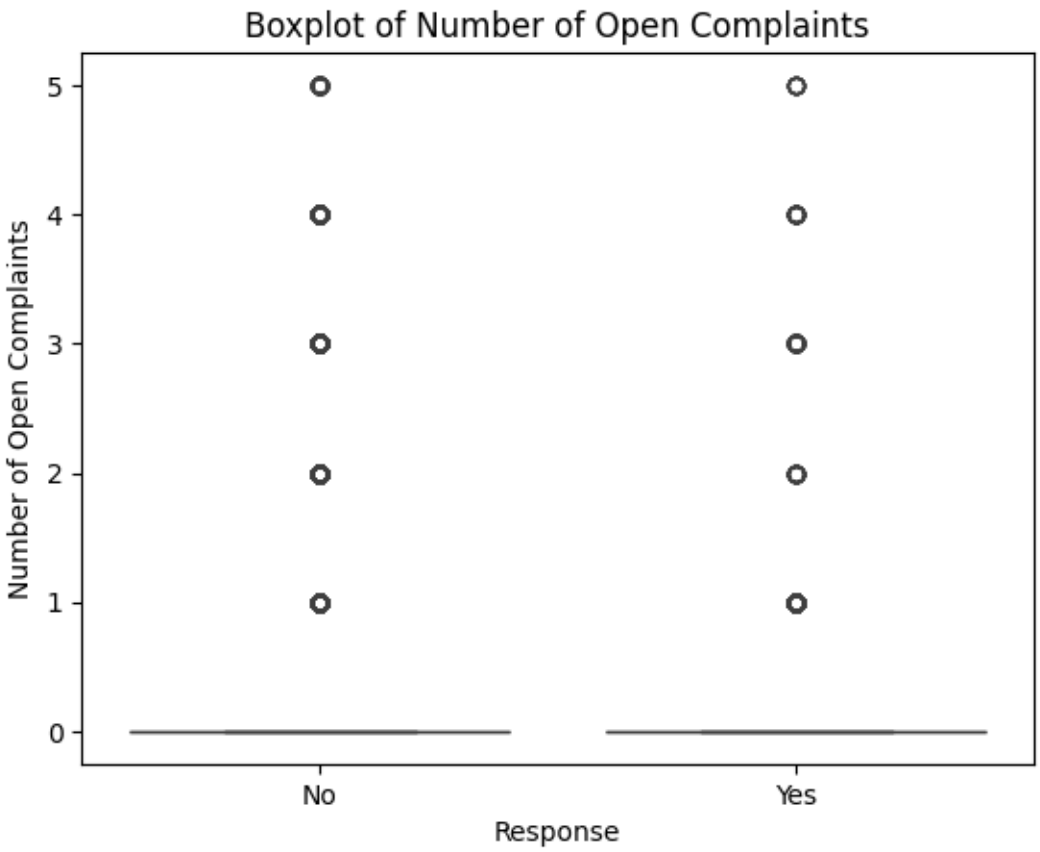


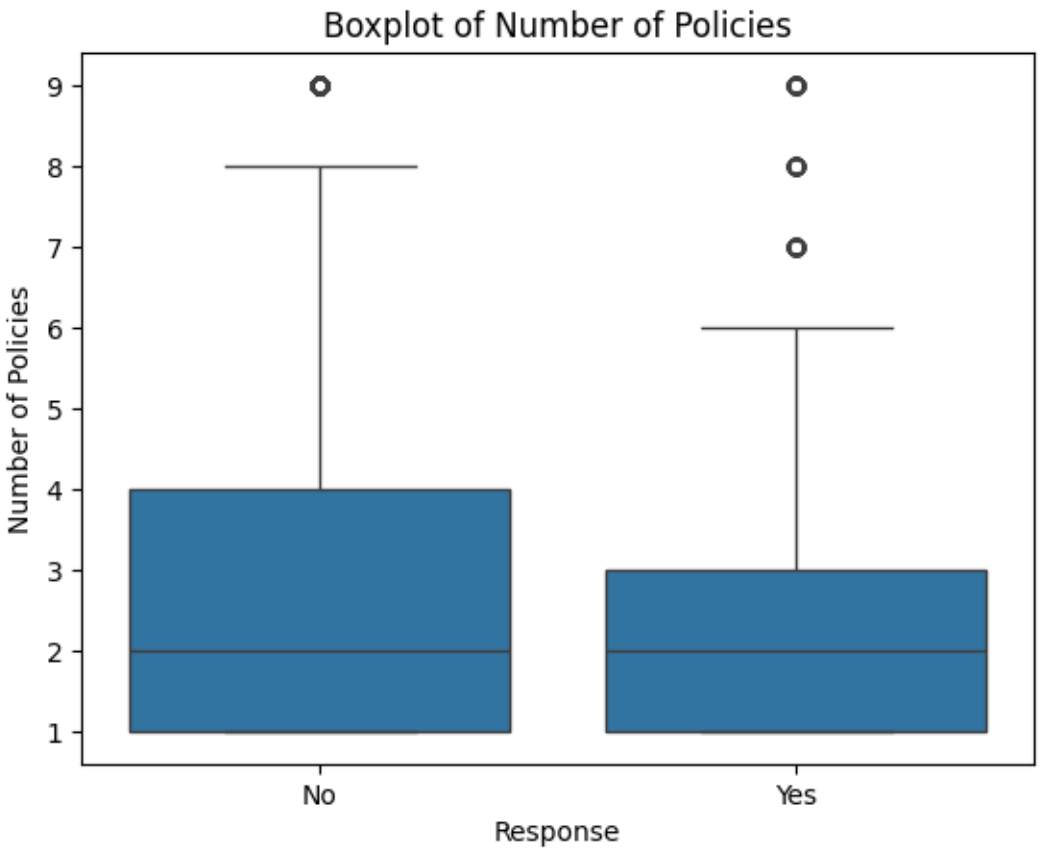

Boxplot of Income
100000
80000
60000
40000
20000
0
Income
No
Yes
Response

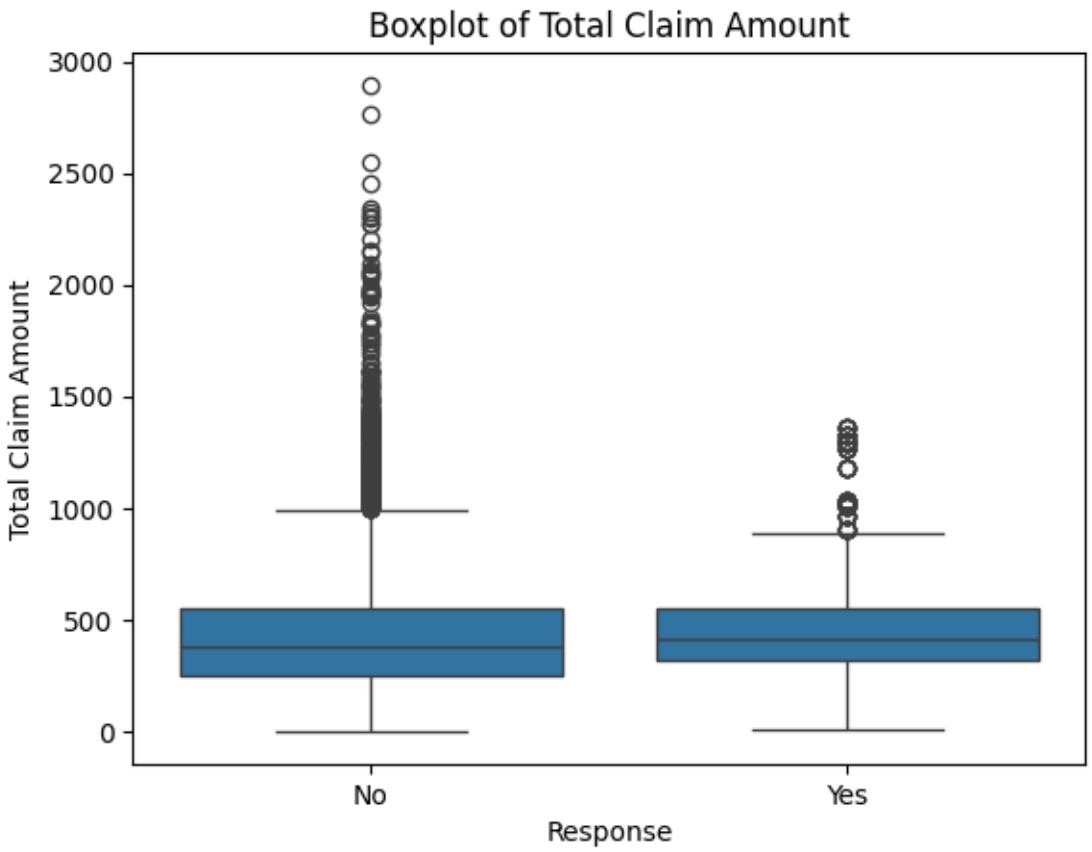

Boxplot of Monthly Premium Auto
Monthly Premium Auto
300
250
200
150
100
50
No
Yes
Response


Boxplot of Months Since Last Claim
Months Since Last Claim
35
30
25
20
15
10
5
0
No
Yes
Response

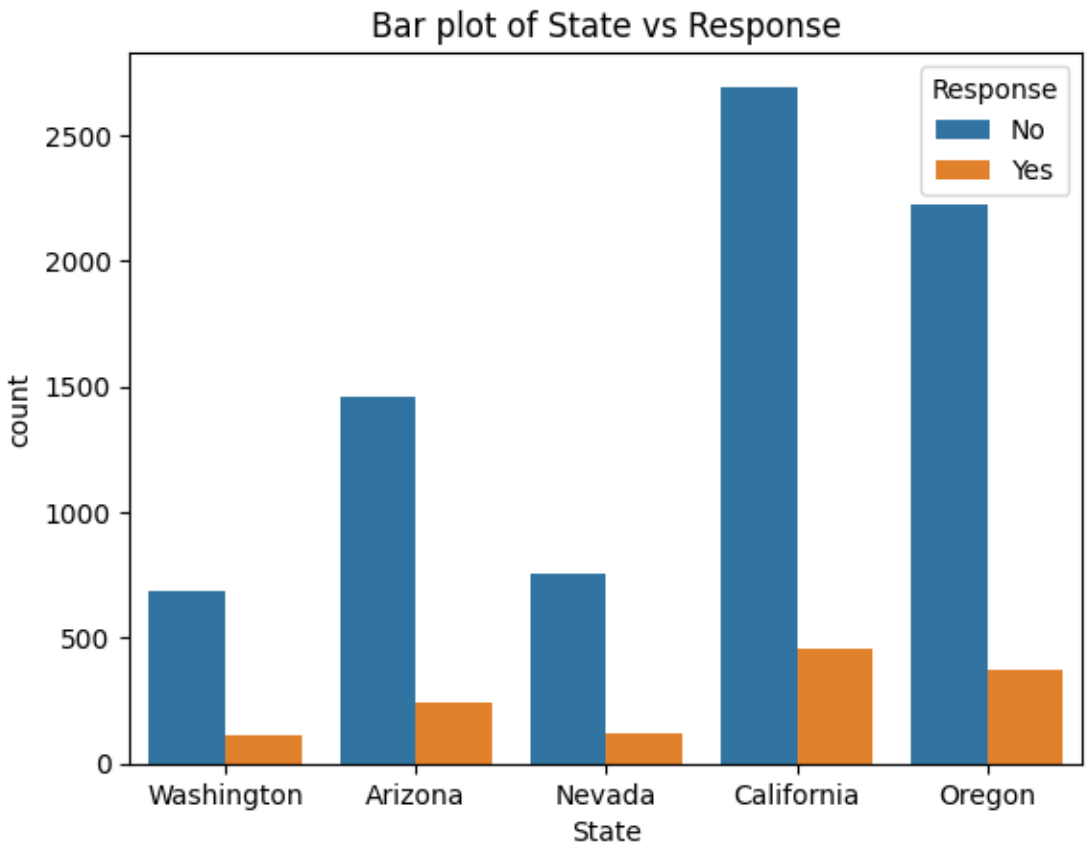

Boxplot of Months Since Policy Inception
Months Since Policy Inception
100
80
60
40
20
0
No
Yes
Response

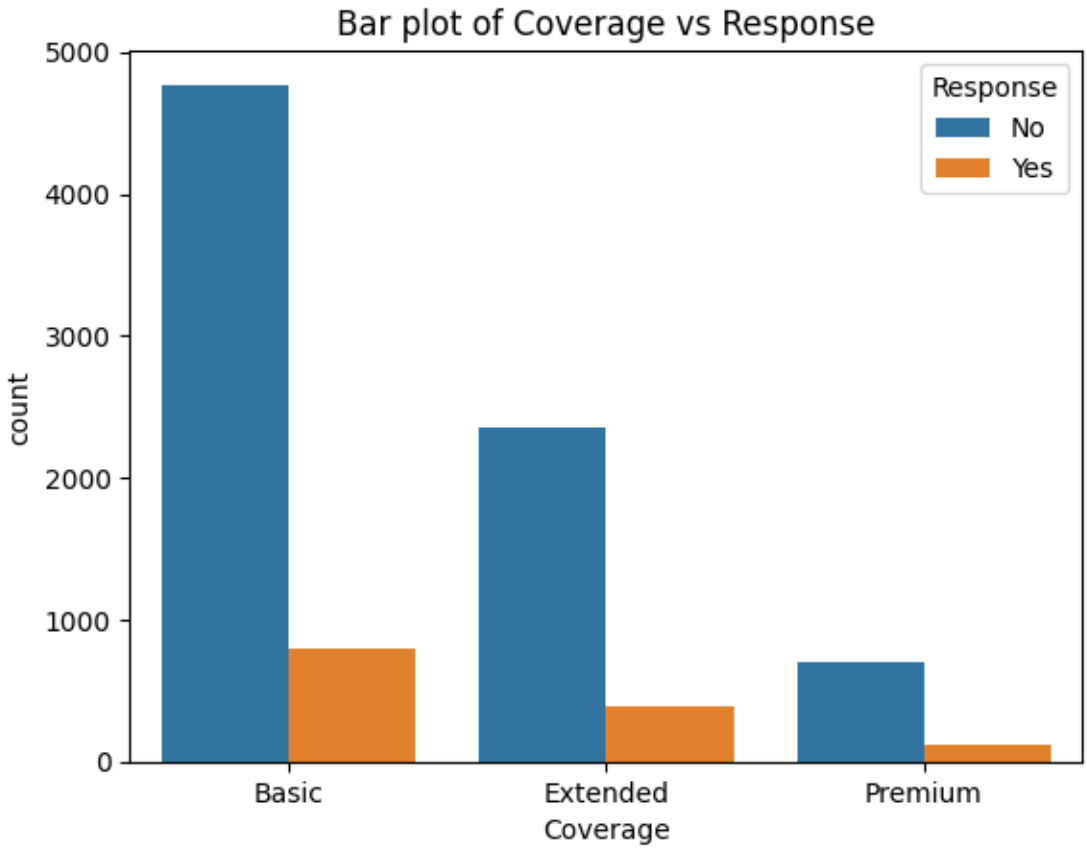

Boxplot of Number of Open Complaints
Number of Open Complaints
0
1
2
3
4
5
No
Yes
Response

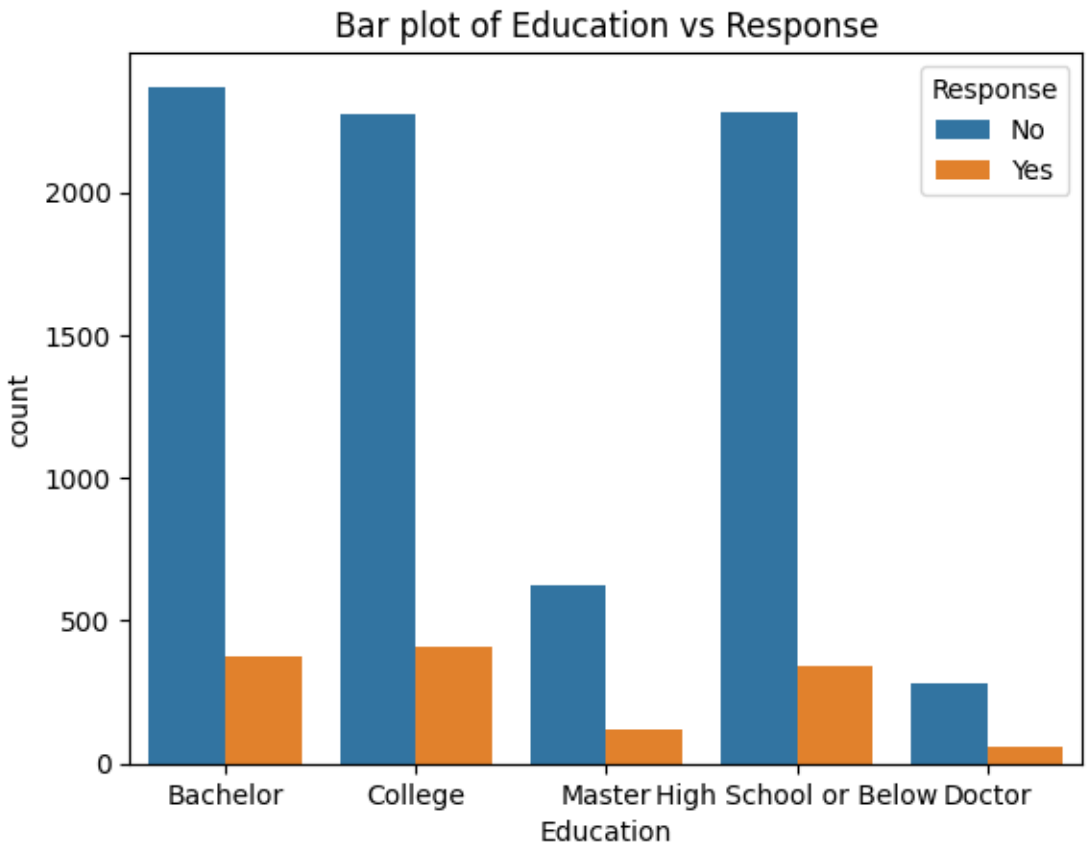

Boxplot of Number of Policies
Number of Policies
9
8
7
6
5
4
3
2
1
No
Yes
Response



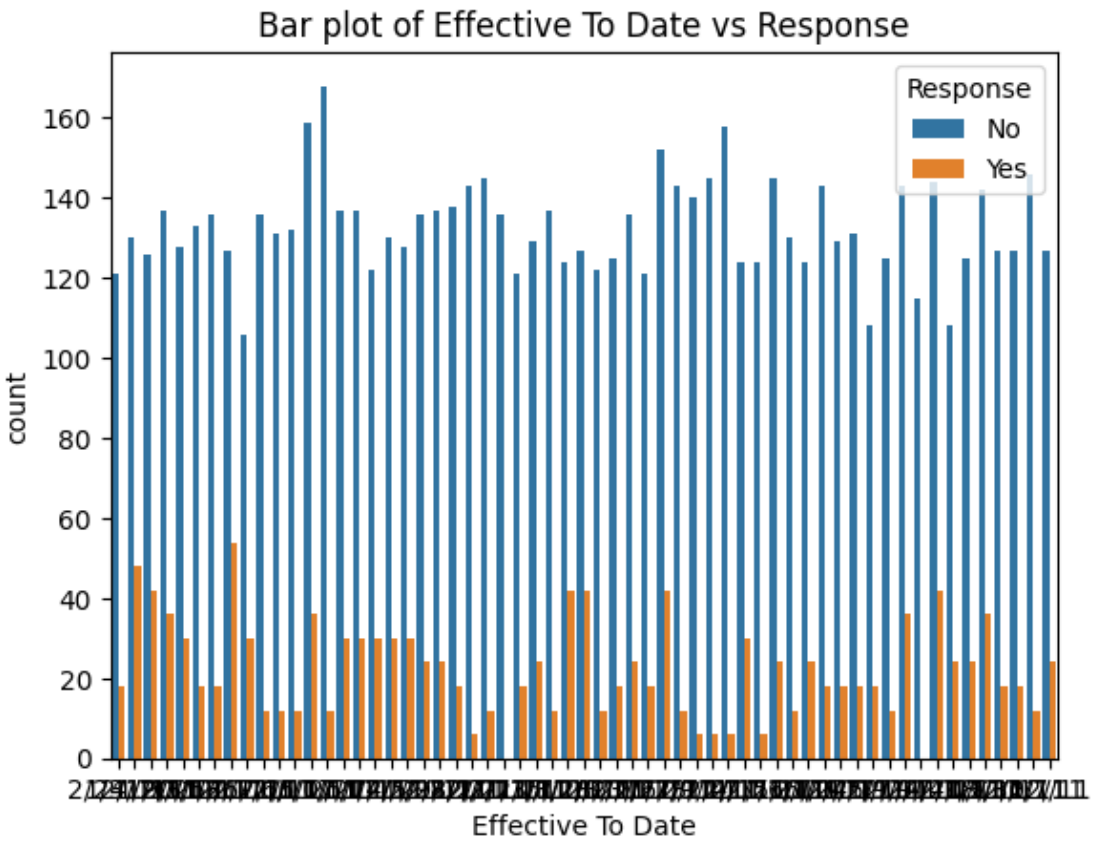
[7]:
```
# Categorical features to visualize
# categorical vs categorical -> barplot để xem tỉ lệ
categorical_features = ['State', 'Coverage', 'Education','Effective To Date',␣
 ↪'EmploymentStatus', 'Gender', 'Location Code',
       'Marital Status', 'Policy Type', 'Policy', 'Renew Offer Type', 'Sales␣
 ↪Channel', 'Vehicle Class', 'Vehicle Size']
# Plotting bar plots for each categorical feature
for feature in categorical_features:
    sns.countplot(x=feature, hue='Response', data=data)
    plt.title(f"Bar plot of {feature} vs Response")
    plt.legend(title='Response', loc="upper right")
    plt.show()
```

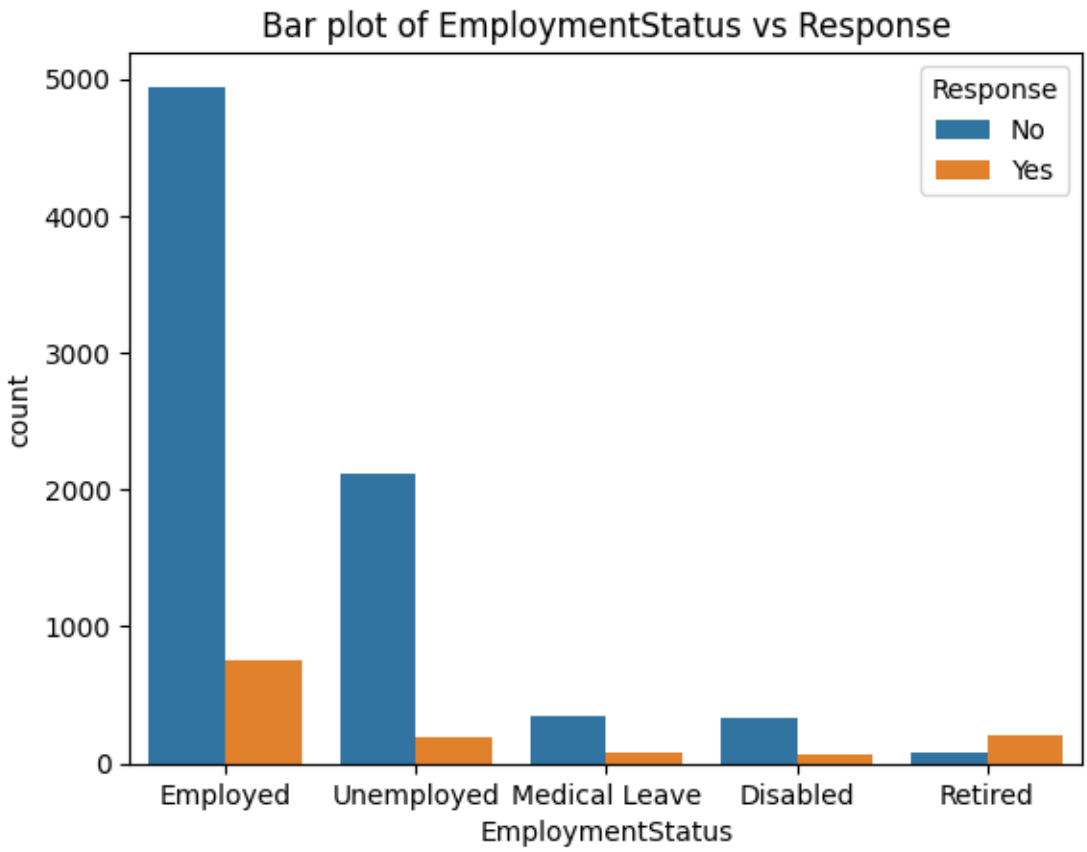

Bar plot of State vs Response
Response
No
Yes
count
2500
2000
1500
1000
500
0
Washington
Arizona
Nevada
California
Oregon
State

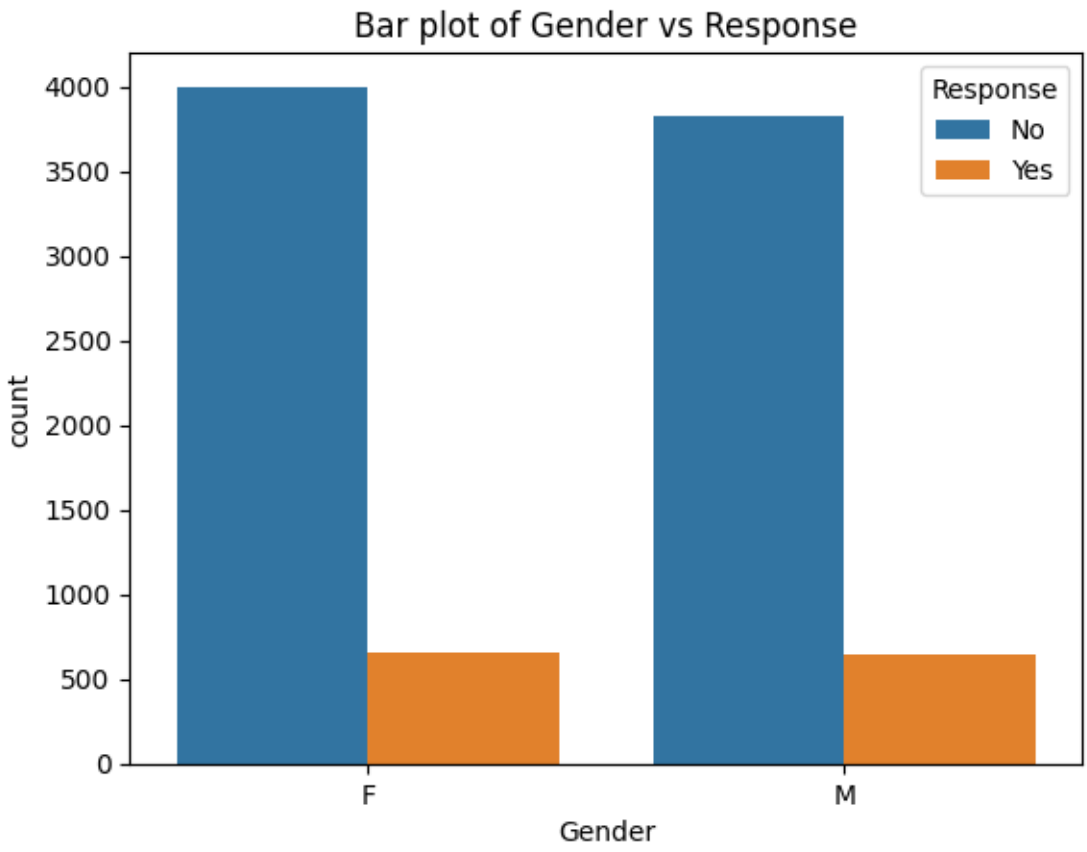

Bar plot of Coverage vs Response
Response
No
Yes
count
5000
4000
3000
2000
1000
0
Basic
Extended
Coverage
Premium

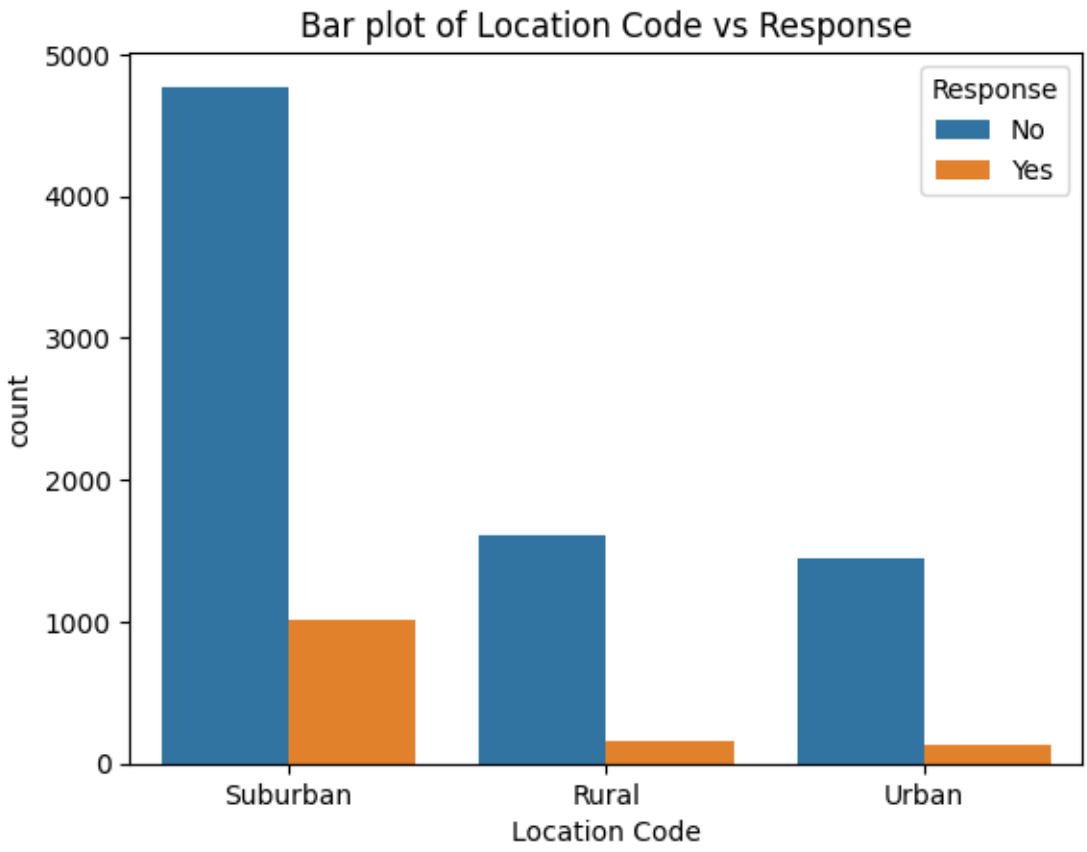


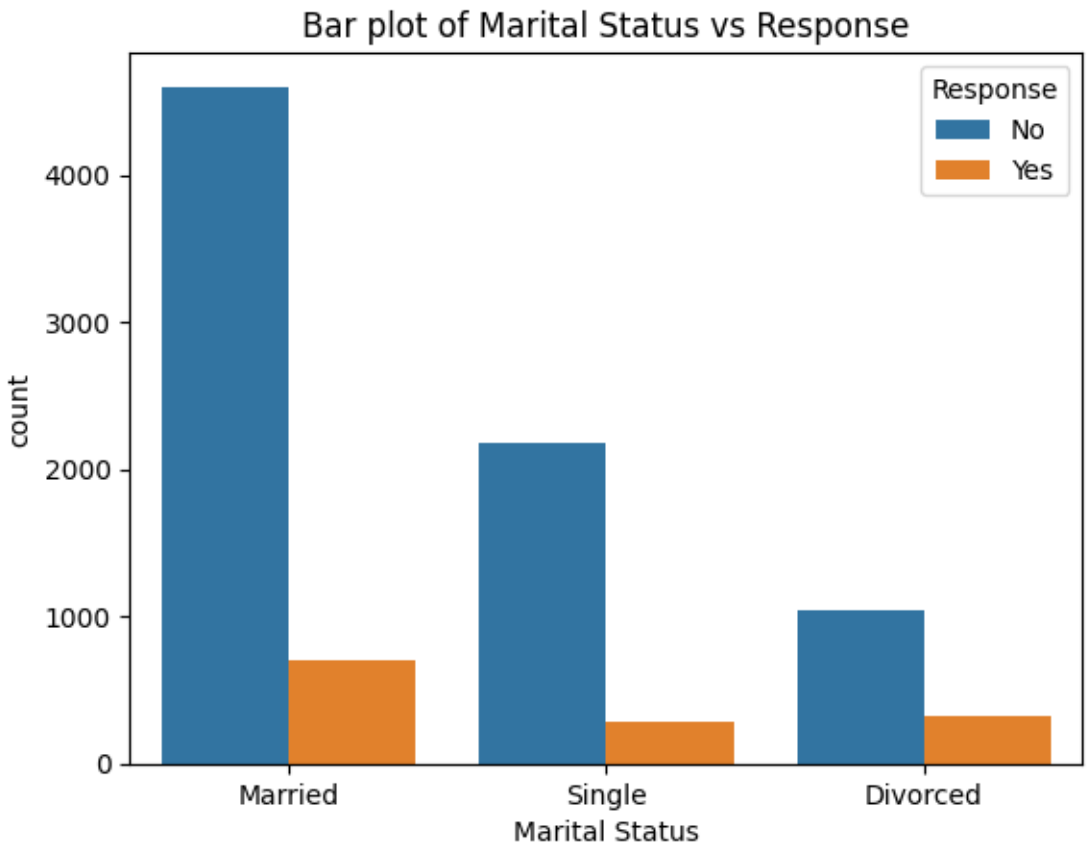

```
Using categorical units to plot a list of strings that are all parsable as
floats or dates. If these strings should be plotted as numbers, cast to the
appropriate data type before plotting.
Using categorical units to plot a list of strings that are all parsable as
floats or dates. If these strings should be plotted as numbers, cast to the
appropriate data type before plotting.
```

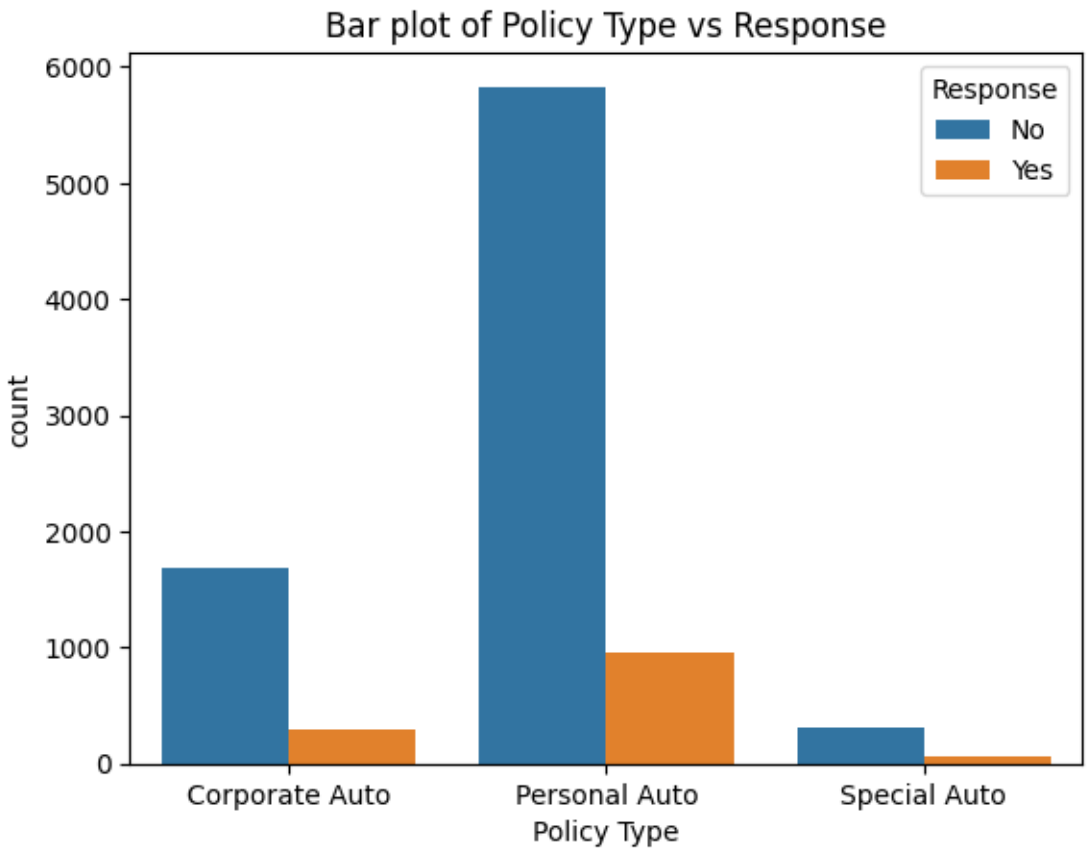

Bar plot of Effective To Date vs Response
Response
No
Yes
count
0
20
40
60
80
100
120
140
160
Effective To Date

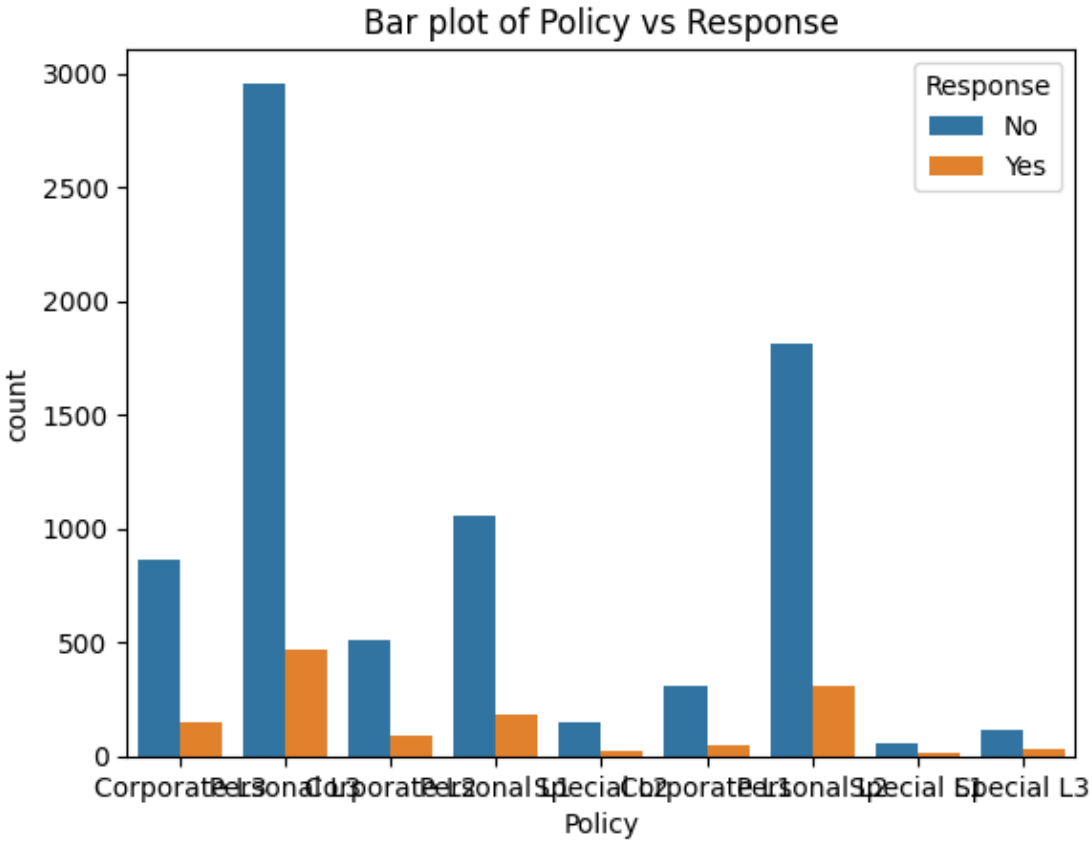

Bar plot of EmploymentStatus vs Response
Response
No
Yes
count
5000
4000
3000
2000
1000
0
Employed
Unemployed
Medical Leave
Disabled
Retired
EmploymentStatus

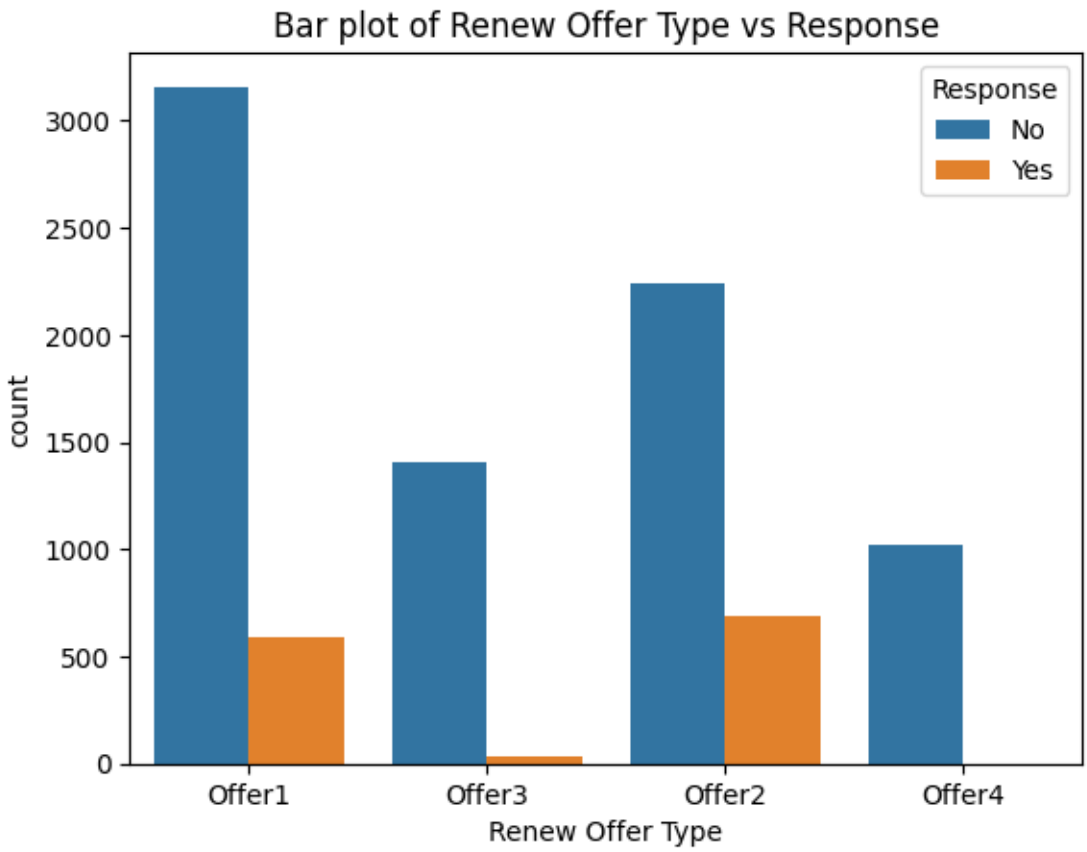

Bar plot of Gender vs Response
Response
No
Yes
count
4000
3500
3000
2500
2000
1500
1000
500
0
F
M
Gender

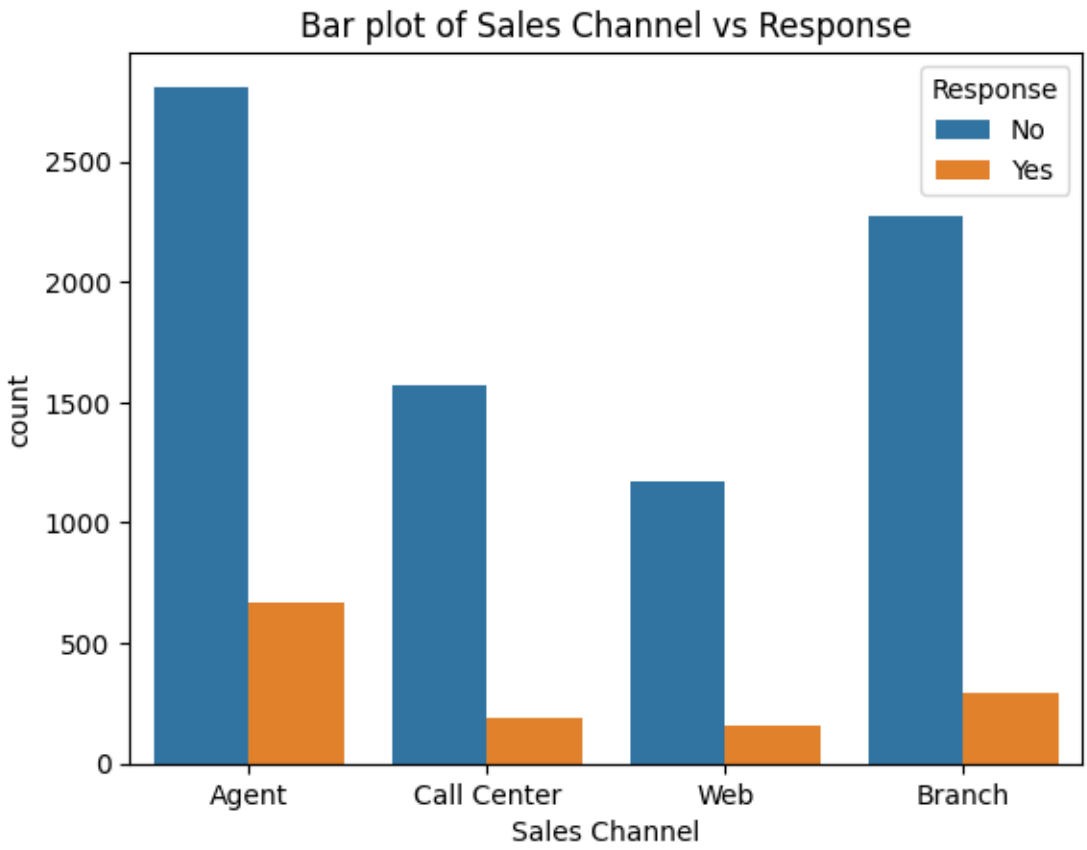

Bar plot of Location Code vs Response
Response
No
Yes
count
5000
4000
3000
2000
1000
0
Suburban
Rural
Urban
Location Code

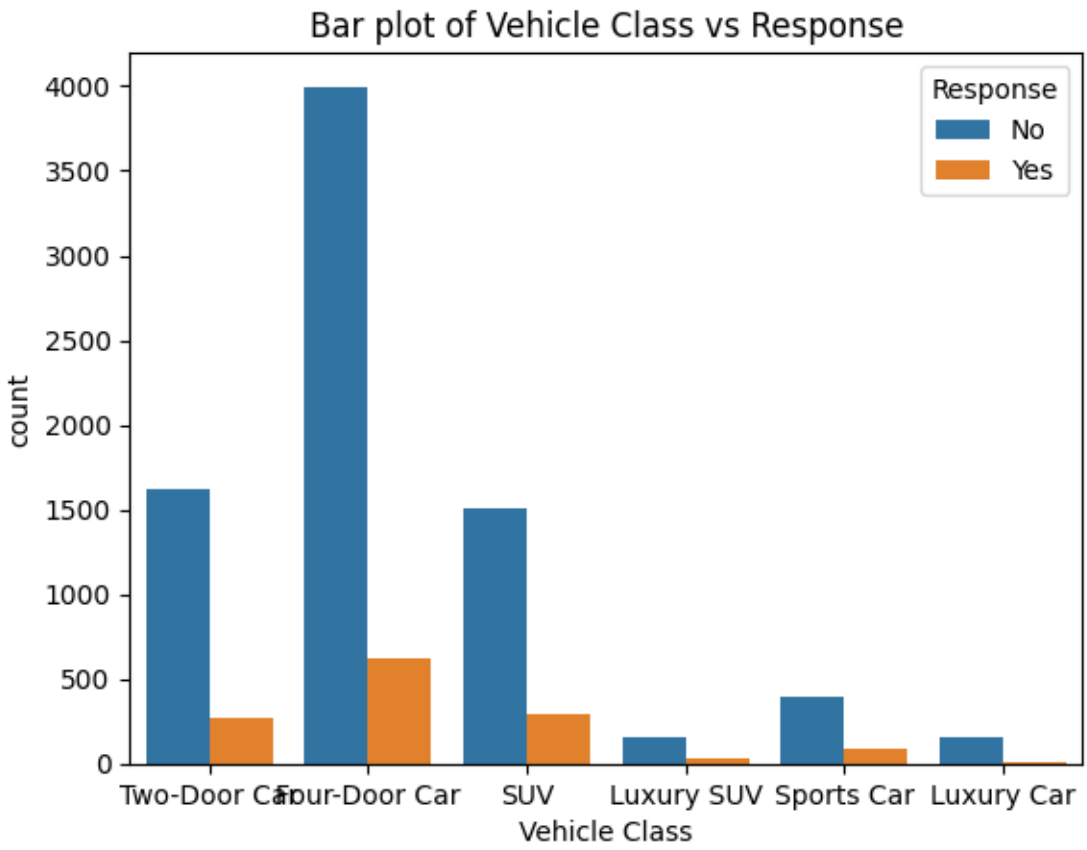

Bar plot of Marital Status vs Response
Response
No
Yes
count
4000
3000
2000
1000
0
Married
Single
Divorced
Marital Status

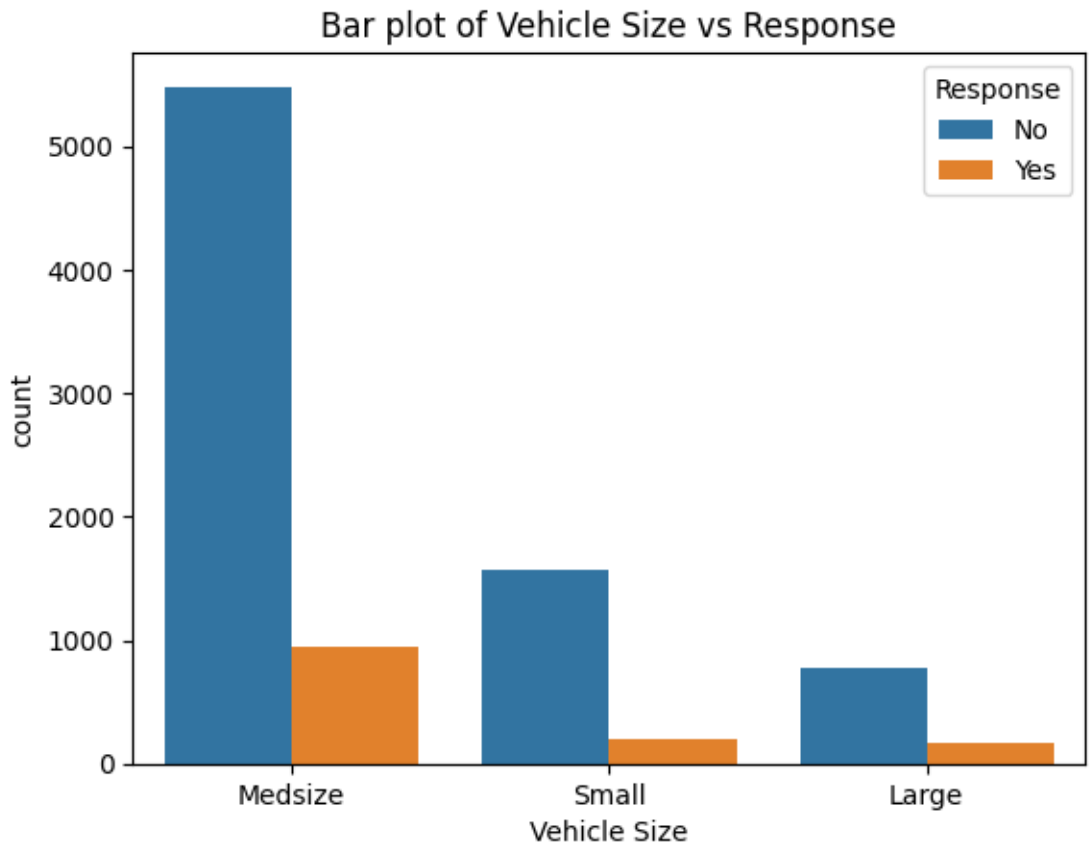

Bar plot of Policy Type vs Response
Response
No
Yes
count
6000
5000
4000
3000
2000
1000
0
Corporate Auto
Personal Auto
Special Auto
Policy Type


Bar plot of Policy vs Response
Response
No
Yes
count
3000
2500
2000
1500
1000
500
0
Policy


Bar plot of Renew Offer Type vs Response
Response
No
Yes
count
3000
2500
2000
1500
1000
500
0
Offer1
Offer3
Offer2
Offer4
Renew Offer Type


Bar plot of Sales Channel vs Response
Response
No
Yes
count
2500
2000
1500
1000
500
0
Agent
Call Center
Web
Branch
Sales Channel


Bar plot of Vehicle Class vs Response
Response
No
Yes
count
4000
3500
3000
2500
2000
1500
1000
500
0
Two-Door Car
Four-Door Car
SUV
Luxury SUV
Sports Car
Luxury Car
Vehicle Class



[ ]:
```
# Các biên ảnh hưởng đến Response của khách hàng:
#- Categorical features: 'State', 'Coverage', 'Effective To Date',␣
 ↪'EmploymentStatus', 'Location Code',
#                        'Marital Status', 'Policy Type', 'Policy', 'Renew␣
 ↪Offer Type', 'Sales Channel', 'Vehicle Class', 'Vehicle Size'
#                ('Education', 'Gender' ảnh hưởng ít đến Response)
#    ** Nhận xét: các biến phân loại đa phần đều có ảnh hưởng đến Response, chỉ␣
 ↪'Education', 'Gender' ảnh hưởng rất ít nên có thể bỏ qua.
#- Continuous features: 'Customer Lifetime Value', 'Income', 'Monthly Premium␣
 ↪Auto', 'Months Since Last Claim', 'Months Since Policy Inception',
#       'Number of Open Complaints', 'Number of Policies', 'Total Claim Amount'
#    ** Nhận xét: hầu hết các biến liên tục đều ảnh hưởng đến Response
```

[8]:
```
# Đổi tên cột để khi sử dụng BigQuery không bị lỗi
data = data.rename(columns={'Customer Lifetime Value':␣
 ↪'Customer_Lifetime_Value','Effective To Date': 'Effective_To_Date',␣
 ↪'Location Code': 'Location_Code',
                            'Marital Status': 'Marital_Status','Monthly Premium␣
 ↪Auto': 'Monthly_Premium_Auto',
```

```
                        'Months Since Last Claim':
 ↪'Months_Since_Last_Claim','Months Since Policy Inception':
 ↪'Months_Since_Policy_Inception',
                        'Number of Open Complaints':
 ↪'Number_of_Open_Complaints','Number of Policies':␣
 ↪'Number_of_Policies','Policy Type': 'Policy_Type',
                        'Renew Offer Type': 'Renew_Offer_Type','Sales Channel':␣
 ↪'Sales_Channel','Total Claim Amount': 'Total_Claim_Amount',
                        'Vehicle Class':'Vehicle_Class','Vehicle Size':
 ↪'Vehicle_Size'})
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9134 entries, 0 to 9133
Data columns (total 24 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   Customer                       9134 non-null   object
 1   State                          9134 non-null   object
 2   Customer_Lifetime_Value        9134 non-null   float64
 3   Response                       9134 non-null   object
 4   Coverage                       9134 non-null   object
 5   Education                      9134 non-null   object
 6   Effective_To_Date              9134 non-null   object
 7   EmploymentStatus               9134 non-null   object
 8   Gender                         9134 non-null   object
 9   Income                         9134 non-null   int64
 10  Location_Code                  9134 non-null   object
 11  Marital_Status                 9134 non-null   object
 12  Monthly_Premium_Auto           9134 non-null   int64
 13  Months_Since_Last_Claim        9134 non-null   int64
 14  Months_Since_Policy_Inception  9134 non-null   int64
 15  Number_of_Open_Complaints      9134 non-null   int64
 16  Number_of_Policies             9134 non-null   int64
 17  Policy_Type                    9134 non-null   object
 18  Policy                         9134 non-null   object
 19  Renew_Offer_Type               9134 non-null   object
 20  Sales_Channel                  9134 non-null   object
 21  Total_Claim_Amount             9134 non-null   float64
 22  Vehicle_Class                  9134 non-null   object
 23  Vehicle_Size                   9134 non-null   object
dtypes: float64(2), int64(6), object(16)
memory usage: 1.7+ MB
```

[9]:
```
data.shape
```

[9]: (9134, 24)

[10]:
```
data.to_csv("Marketing-Customer-Value-Analysis_new.csv")
```

[ ]:
```

```