Regression_MarketingEngagement

May 5, 2019

```
In [1]: import matplotlib.pyplot as plt
        import pandas as pd
```

1 1. Loading Data

```
In [2]: df = pd.read_csv('Data/WA_Fn-UseC_-Marketing-Customer-Value-Analysis.csv')

In [3]: df.shape

Out[3]: (9134, 24)

In [4]: df.head()
```

Out[4]:

	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]
```

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

Out[5]:

	Customer	State	Customer Lifetime Value	Response	Coverage	\
9129	LA72316	California	23405.987980	No	Basic	
9130	PK87824	California	3096.511217	Yes	Extended	
9131	TD14365	California	8163.890428	No	Extended	
9132	UP19263	California	7524.442436	No	Extended	
9133	Y167826	California	2611.836866	No	Extended	

	Education	Effective To Date	EmploymentStatus	Gender	Income	...	\
9129	Bachelor	2/10/11	Employed	M	71941	...	
9130	College	2/12/11	Employed	F	21604	...	
9131	Bachelor	2/6/11	Unemployed	M	0	...	
9132	College	2/3/11	Employed	M	21941	...	
9133	College	2/14/11	Unemployed	M	0	...	

	Months Since Policy Inception	Number of Open Complaints	\
9129	89	0	
9130	28	0	
9131	37	3	
9132	3	0	
9133	90	0	

	Number of Policies	Policy Type	Policy	Renew Offer Type	\
9129	2	Personal Auto	Personal L1	Offer2	
9130	1	Corporate Auto	Corporate L3	Offer1	
9131	2	Corporate Auto	Corporate L2	Offer1	
9132	3	Personal Auto	Personal L2	Offer3	
9133	1	Corporate Auto	Corporate L3	Offer4	

	Sales Channel	Total Claim Amount	Vehicle Class	Vehicle Size
9129	Web	198.234764	Four-Door Car	Medsize
9130	Branch	379.200000	Four-Door Car	Medsize
9131	Branch	790.784983	Four-Door Car	Medsize
9132	Branch	691.200000	Four-Door Car	Large

```
9133    Call Center         369.600000   Two-Door Car      Medsize

[5 rows x 24 columns]
```

In [6]:
```
# Encode the y variable as 1 for 'Yes' and as 0 for 'No'

df['Engaged'] = df['Response'].apply(lambda x:0 if x == 'No' else 1)
df.head()
```

Out[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	...

	Number of Open Complaints	Number of Policies	Policy Type	Policy
0	0	1	Corporate Auto	Corporate L3
1	0	8	Personal Auto	Personal L3
2	0	2	Personal Auto	Personal L3
3	0	7	Corporate Auto	Corporate L2
4	0	1	Personal Auto	Personal L1

	Renew Offer Type	Sales Channel	Total Claim Amount	Vehicle Class
0	Offer1	Agent	384.811147	Two-Door Car
1	Offer3	Agent	1131.464935	Four-Door Car
2	Offer1	Agent	566.472247	Two-Door Car
3	Offer1	Call Center	529.881344	SUV
4	Offer1	Agent	138.130879	Four-Door Car

	Vehicle Size	Engaged
0	Medsize	0
1	Medsize	0
2	Medsize	0
3	Medsize	0
4	Medsize	0

[5 rows x 25 columns]

2 2. Data Analysis

```
In [7]: list(df.columns)
```

```
Out[7]: ['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',
         'Engaged']
```

2.1 Engagement Rate

```
In [8]: engagement_rate_df = pd.DataFrame(
            df.groupby('Engaged').count()['Response'] / df.shape[0] * 100.0
        )
```

```
In [9]: engagement_rate_df
```

```
Out[9]:           Response
        Engaged
        0        85.679877
        1        14.320123
```

```
In [10]: # Transpose the df
         engagement_rate_df.T
```

```
Out[10]: Engaged           0          1
         Response  85.679877  14.320123
```

Notice that about 14% of the customers have responded to marketing calls, and the remaining 86% of the customers have not responded.

2.1.1 - by Renew Offer Type

```
In [11]: engagement_by_offer_type_df = pd.pivot_table(
             df, values='Response', index='Renew Offer Type', columns='Engaged', aggfunc=len
         ).fillna(0.0)

         engagement_by_offer_type_df.columns = ['Not Engaged', 'Engaged']
```

aggfunc() allows us to supply the type of aggregation we want to perform. We use len function to simply count the number of clients for each group.

```
In [12]: engagement_by_offer_type_df
```

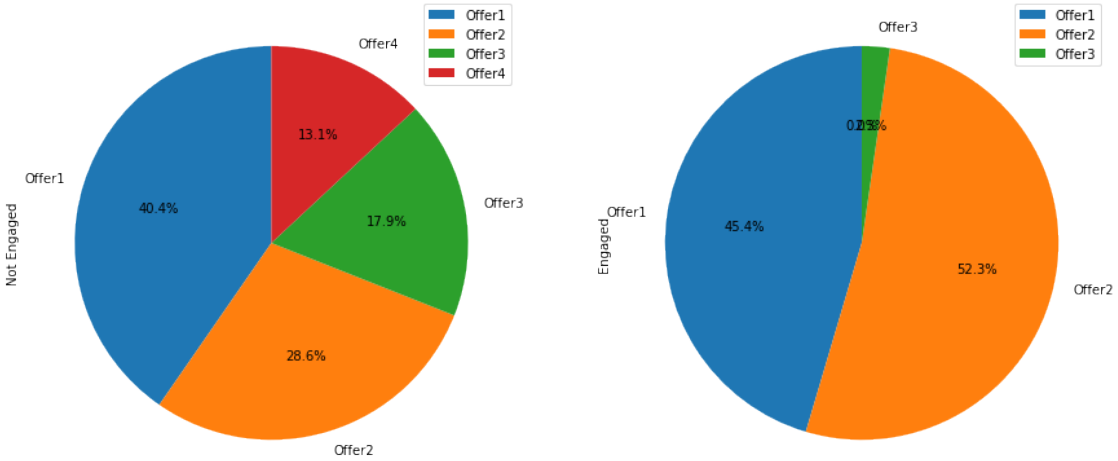
Out[12]:

Renew Offer Type	Not Engaged	Engaged
Offer1	3158.0	594.0
Offer2	2242.0	684.0
Offer3	1402.0	30.0
Offer4	1024.0	0.0

```
In [13]: # Visualize the renew offer type distributions between engaged
         # and non-engaged customers

         engagement_by_offer_type_df.plot(
             kind='pie',
             figsize=(15, 7),
             startangle=90,
             subplots=True,
             autopct=lambda x: '%0.1f%%' % x
         )

         plt.show()
```



2.1.2 - by Sales Channel

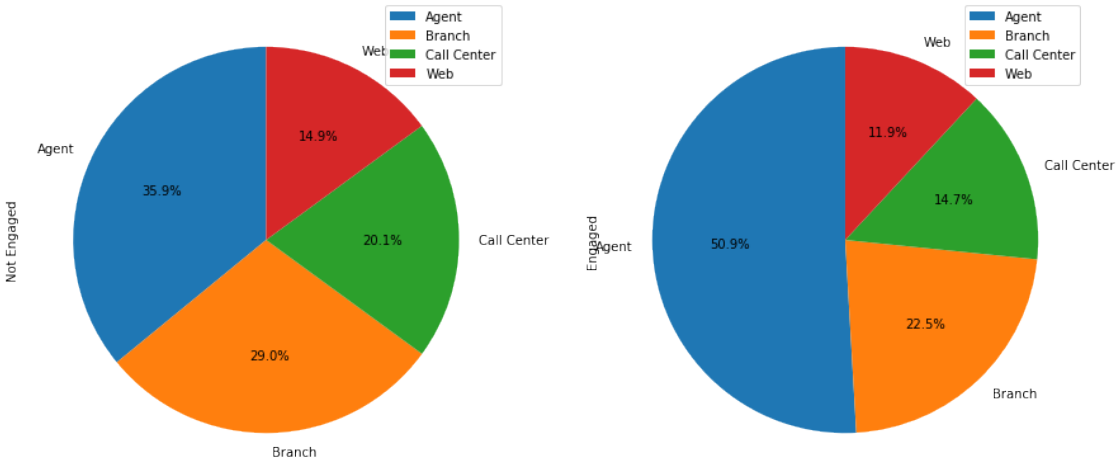
In [14]:
```
engagement_by_sales_channel_df = pd.pivot_table(
    df, values='Response', index='Sales Channel', columns='Engaged', aggfunc=len
).fillna(0.0)

engagement_by_sales_channel_df.columns = ['Not Engaged', 'Engaged']
```

In [15]:
```
# Visualize the sales channel distributions between engaged
# and non-engaged customers

engagement_by_sales_channel_df.plot(
    kind='pie',
    figsize=(15, 7),
    startangle=90,
    subplots=True,
    autopct=lambda x: '%0.1f%%' % x
)

plt.show()
```



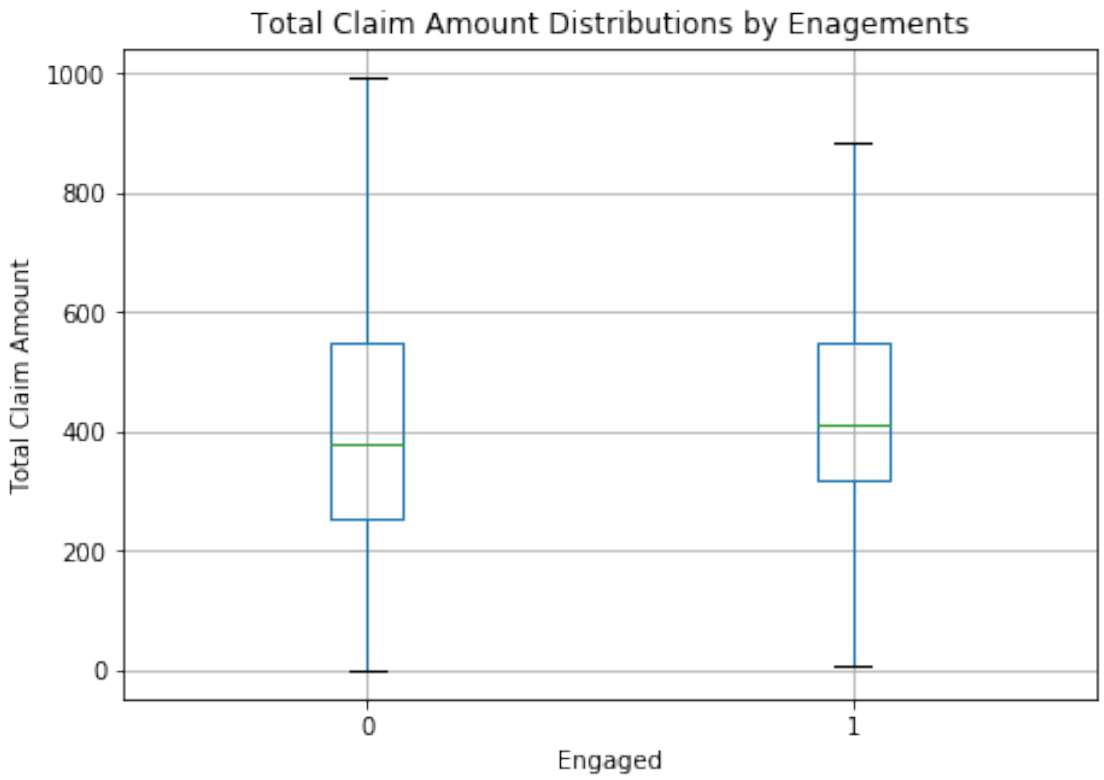
2.1.3 - by Total Claim Amount Distributions

In [16]:
```
# Look at the differences in the distributions of Total Claim Amount between engaged
# and non-engaged customers by using box plots

ax = df[['Engaged', 'Total Claim Amount']].boxplot(
    by='Engaged',
    showfliers=False,
    figsize=(7,5)
)
```

```
ax.set_xlabel('Engaged')
ax.set_ylabel('Total Claim Amount')
ax.set_title('Total Claim Amount Distributions by Enagements')

plt.suptitle("")
plt.show()
```



Box plots are a great way to visualize the distribution of continuous variables. They show the min, max, first quartile, median and third quartile, all in one view. The central rectangle spans from the first quartile to the third quartile, and the green line shows the median. The lower and upper ends show the minimum and the maximum of each distribution.

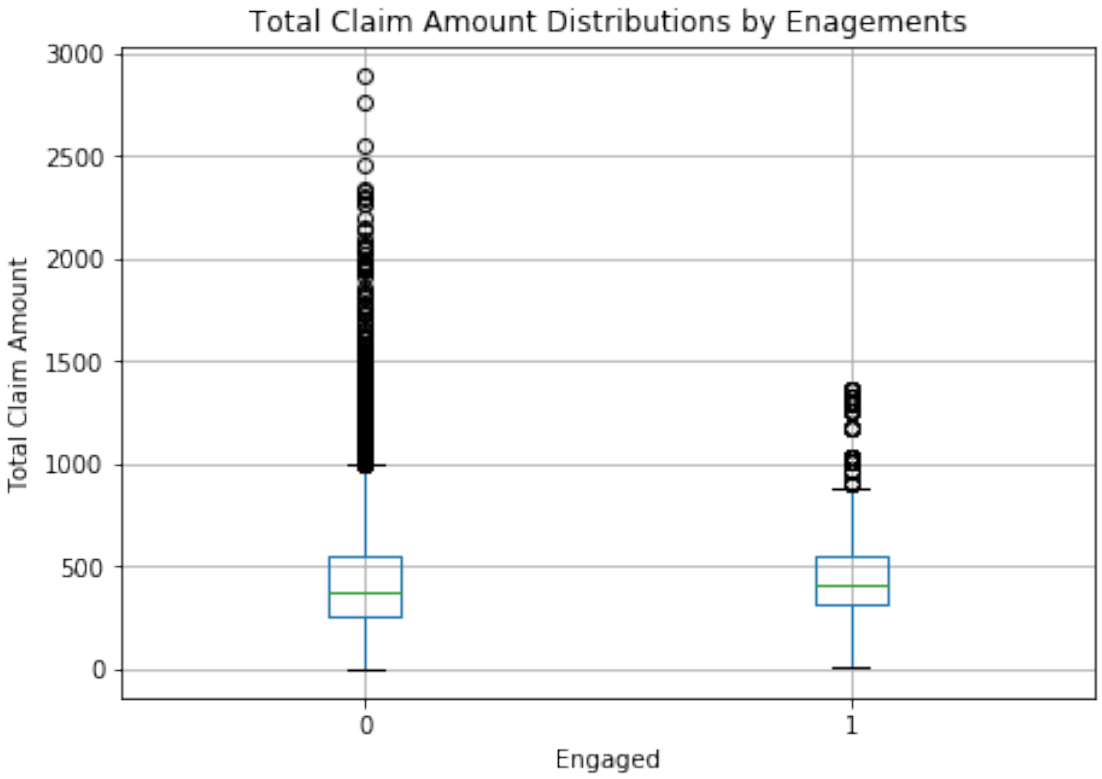
In [17]:
```
# See what happens if we set showfliers=True

ax = df[['Engaged', 'Total Claim Amount']].boxplot(
    by='Engaged',
    showfliers=True,
    figsize=(7,5)
)

ax.set_xlabel('Engaged')
ax.set_ylabel('Total Claim Amount')
```

```
        ax.set_title('Total Claim Amount Distributions by Enagements')

        plt.suptitle("")
        plt.show()
```



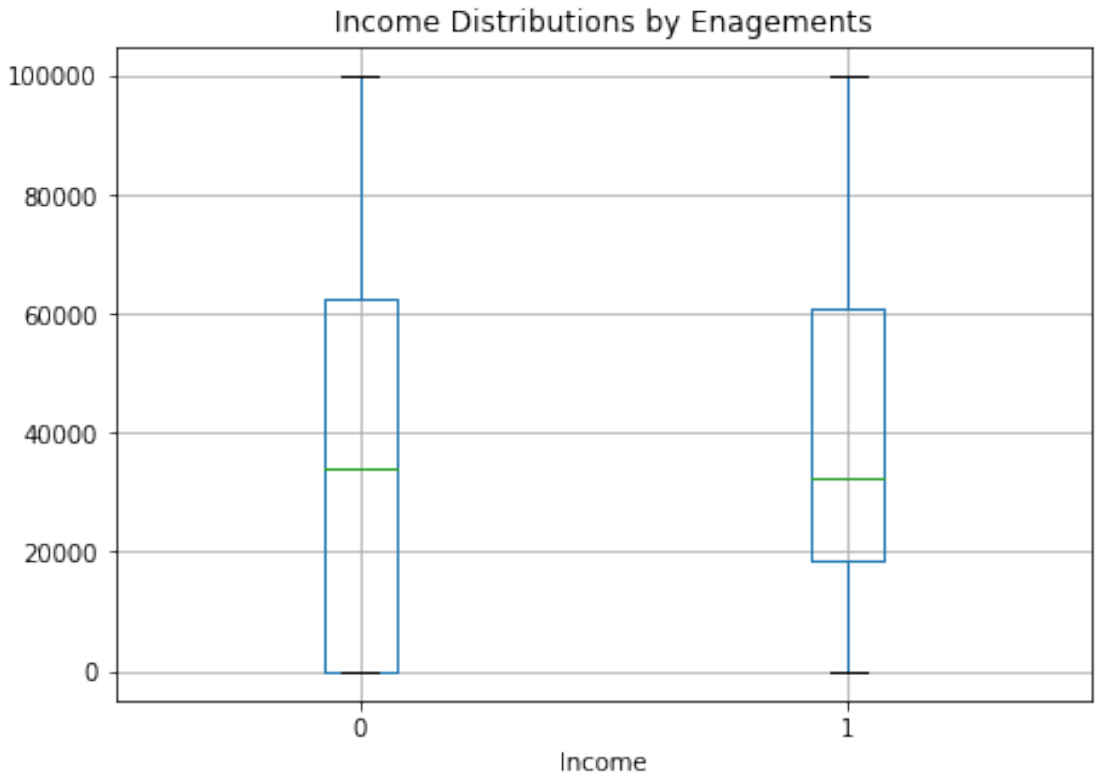
The dots above the upper boundary line show the suspected ouliers that are decided based on the **'Interquartile range (IQR)'**. The points that fall **1.5 * IQR** above the third quartile or **1.5 * IQR** below the first quartile are suspected outliers and are drawn with the dots.

2.1.4 - by Income Distributions

```
In [18]: ax = df[['Engaged', 'Income']].boxplot(
             by='Engaged',
             showfliers=True,
             figsize=(7,5)
         )

         ax.set_xlabel('Engaged')
         ax.set_xlabel('Income')
         ax.set_title('Income Distributions by Enagements')

         plt.suptitle("")
         plt.show()
```




```
In [19]: df.groupby('Engaged').describe()['Income'].T
```

```
Out[19]: Engaged             0             1
         count      7826.000000   1308.000000
         mean      37509.190008  38544.027523
         std       30752.259099  28043.637944
         min           0.000000      0.000000
         25%           0.000000  18495.000000
         50%       34091.000000  32234.000000
         75%       62454.250000  60880.000000
         max       99981.000000  99845.000000
```

3 3. Regression Analysis with Continuous Variables Only

```
In [20]: import statsmodels.formula.api as sm
```

```
In [21]: df.describe()
```

```
Out[21]:        Customer Lifetime Value        Income  Monthly Premium Auto  \
         count              9134.000000   9134.000000           9134.000000
         mean               8004.940475  37657.380009             93.219291
         std                6870.967608  30379.904734             34.407967
```

```
min                1898.007675      0.000000             61.000000
25%                3994.251794      0.000000             68.000000
50%                5780.182197  33889.500000             83.000000
75%                8962.167041  62320.000000            109.000000
max               83325.381190  99981.000000            298.000000

       Months Since Last Claim  Months Since Policy Inception  \
count              9134.000000                    9134.000000
mean                 15.097000                      48.064594
std                  10.073257                      27.905991
min                   0.000000                       0.000000
25%                   6.000000                      24.000000
50%                  14.000000                      48.000000
75%                  23.000000                      71.000000
max                  35.000000                      99.000000

       Number of Open Complaints  Number of Policies  Total Claim Amount  \
count                9134.000000         9134.000000         9134.000000
mean                    0.384388            2.966170          434.088794
std                     0.910384            2.390182          290.500092
min                     0.000000            1.000000            0.099007
25%                     0.000000            1.000000          272.258244
50%                     0.000000            2.000000          383.945434
75%                     0.000000            4.000000          547.514839
max                     5.000000            9.000000         2893.239678

           Engaged
count  9134.000000
mean      0.143201
std       0.350297
min       0.000000
25%       0.000000
50%       0.000000
75%       0.000000
max       1.000000
```

In [22]:
```
df['Income'].dtype
```

Out[22]:
```
dtype('int64')
```

In [23]:
```
df['Customer Lifetime Value'].dtype
```

Out[23]:
```
dtype('float64')
```

In [24]:
```
# Store the continuous variables in a separate variable

continuous_vars = [
    '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'
    ]
```

In [25]:
```
# Initiate a logistic regression model supplying the Engaged column as the output
# variable (target) and the continuous variables as the input variables

logit = sm.Logit(
    df['Engaged'],
    df[continuous_vars]
)
```

In [26]:
```
# Train or fit this model

logit_fit = logit.fit()
```

```
Optimization terminated successfully.
         Current function value: 0.421189
         Iterations 6
```

In [27]:
```
# Get a detailed description of the trained model

logit_fit.summary()
```

Out[27]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                           Logit Regression Results
==============================================================================
Dep. Variable:                Engaged   No. Observations:                 9134
Model:                          Logit   Df Residuals:                     9126
Method:                           MLE   Df Model:                            7
Date:                Sun, 05 May 2019   Pseudo R-squ.:                -0.02546
Time:                        22:35:43   Log-Likelihood:                -3847.1
converged:                       True   LL-Null:                       -3751.6
                                        LLR p-value:                    1.000
==============================================================================
                                 coef    std err          z      P>|z|      [0.025
------------------------------------------------------------------------------
Customer Lifetime Value    -6.741e-06   5.04e-06     -1.337      0.181   -1.66e-05
Income                     -2.857e-06   1.03e-06     -2.766      0.006   -4.88e-06
Monthly Premium Auto          -0.0084      0.001     -6.889      0.000      -0.011
Months Since Last Claim       -0.0202      0.003     -7.238      0.000      -0.026
Months Since Policy Inception -0.0060      0.001     -6.148      0.000      -0.008
Number of Open Complaints     -0.0829      0.034     -2.424      0.015      -0.150
Number of Policies            -0.0810      0.013     -6.356      0.000      -0.106
Total Claim Amount             0.0001      0.000      0.711      0.477      -0.000
==============================================================================
"""
```

By looking at the p-value of "Income", "Monthly Premium Auto", "Months Since Last Claim", "Months Since Policy Inception", "Number of Open Complaints", "Number of Open Complaints", "Number of Policies", these input variables seems to have significant relationships with the output variable "Engaged". By looking at the coefficients, they are all negatively correlated to the Engaged variable.

4 4. Regression Analysis with Categorical Variables

```
In [28]: df.describe()
```

Out[28]:

	Customer Lifetime Value	Income	Monthly Premium Auto	\
count	9134.000000	9134.000000	9134.000000	
mean	8004.940475	37657.380009	93.219291	
std	6870.967608	30379.904734	34.407967	
min	1898.007675	0.000000	61.000000	
25%	3994.251794	0.000000	68.000000	
50%	5780.182197	33889.500000	83.000000	
75%	8962.167041	62320.000000	109.000000	
max	83325.381190	99981.000000	298.000000	

	Months Since Last Claim	Months Since Policy Inception	\
count	9134.000000	9134.000000	
mean	15.097000	48.064594	
std	10.073257	27.905991	
min	0.000000	0.000000	
25%	6.000000	24.000000	
50%	14.000000	48.000000	
75%	23.000000	71.000000	
max	35.000000	99.000000	

	Number of Open Complaints	Number of Policies	Total Claim Amount	\
count	9134.000000	9134.000000	9134.000000	
mean	0.384388	2.966170	434.088794	
std	0.910384	2.390182	290.500092	
min	0.000000	1.000000	0.099007	
25%	0.000000	1.000000	272.258244	
50%	0.000000	2.000000	383.945434	
75%	0.000000	4.000000	547.514839	
max	5.000000	9.000000	2893.239678	

	Engaged
count	9134.000000
mean	0.143201
std	0.350297
min	0.000000
25%	0.000000
50%	0.000000

```
75%       0.000000
max       1.000000
```

```
In [29]: df.columns
```

```
Out[29]: 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', 'Engaged'],
               dtype='object')
```

```
In [30]: df.filter(items=['Gender']).head()
```

```
Out[30]:   Gender
         0      F
         1      F
         2      F
         3      M
         4      M
```

4.1 Different ways to handle categorical variables

4.1.1 1. Factorizing

```
In [31]: gender_values, gender_labels = df['Gender'].factorize()
```

The pandas function **factorize()** encodes categorical variables with numerical variables by enumarating through the values. Let's take a look:

```
In [32]: gender_values
```

```
Out[32]: array([0, 0, 0, ..., 1, 1, 1], dtype=int64)
```

```
In [33]: gender_labels
```

```
Out[33]: Index(['F', 'M'], dtype='object')
```

As you can see, **0 symbolizes female (F)** and **1 symbolizes male (M)**

```
In [34]: labels, levels = df['Education'].factorize()
```

```
In [35]: labels
```

```
Out[35]: array([0, 0, 0, ..., 0, 1, 1], dtype=int64)
```

```
In [36]: levels
```

```
Out[36]: Index(['Bachelor', 'College', 'Master', 'High School or Below', 'Doctor'], dtype='obj
```

This method is not appropriate because the feature of Education has 5 different categories.

4.1.2 2. pandas' Categorical variable series

```
In [37]: categories = pd.Categorical(
             df['Education'],
             categories=['High School or Below', 'Bachelor', 'College', 'Master', 'Doctor']
         )

In [38]: categories.categories

Out[38]: Index(['High School or Below', 'Bachelor', 'College', 'Master', 'Doctor'], dtype='obj

In [39]: categories.codes

Out[39]: array([1, 1, 1, ..., 1, 2, 2], dtype=int8)
```

4.1.3 3. dummy variables

```
In [40]: pd.get_dummies(df['Education']).head(10)

Out[40]:    Bachelor  College  Doctor  High School or Below  Master
         0         1        0       0                     0       0
         1         1        0       0                     0       0
         2         1        0       0                     0       0
         3         1        0       0                     0       0
         4         1        0       0                     0       0
         5         1        0       0                     0       0
         6         0        1       0                     0       0
         7         0        0       0                     0       1
         8         1        0       0                     0       0
         9         0        1       0                     0       0
```

4.2 Adding these encoded variables of Gender and Education

```
In [41]: df['GenderFactorized'] = gender_values

In [42]: df['EducationFactorized'] = categories.codes

In [43]: df.head()

Out[43]:   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  ...     Policy Type  \
         0           2/24/11         Employed      F   56274  ...  Corporate Auto
         1           1/31/11       Unemployed      F       0  ...   Personal Auto
         2           2/19/11         Employed      F   48767  ...   Personal Auto
```

```
3           1/20/11       Unemployed      M       0  ...  Corporate Auto
4            2/3/11         Employed      M   43836  ...   Personal Auto

         Policy  Renew Offer Type  Sales Channel  Total Claim Amount  \
0  Corporate L3            Offer1          Agent          384.811147
1   Personal L3            Offer3          Agent         1131.464935
2   Personal L3            Offer1          Agent          566.472247
3  Corporate L2            Offer1    Call Center          529.881344
4   Personal L1            Offer1          Agent          138.130879

   Vehicle Class  Vehicle Size Engaged GenderFactorized EducationFactorized
0   Two-Door Car       Medsize       0                0                   1
1  Four-Door Car       Medsize       0                0                   1
2   Two-Door Car       Medsize       0                0                   1
3            SUV       Medsize       0                1                   1
4  Four-Door Car       Medsize       0                1                   1

[5 rows x 27 columns]
```

4.3 Fit a logistic regression model with two categorical variables Gender and Education

```
In [44]: logit = sm.Logit(
             df['Engaged'],
             df[[
                 'GenderFactorized',
                 'EducationFactorized'
             ]]
         )

In [45]: logit_fit = logit.fit()

Optimization terminated successfully.
         Current function value: 0.493068
         Iterations 6


In [46]: logit_fit.summary()

Out[46]: <class 'statsmodels.iolib.summary.Summary'>
         """
                                   Logit Regression Results
         ==============================================================================
         Dep. Variable:                Engaged   No. Observations:                 9134
         Model:                          Logit   Df Residuals:                     9132
         Method:                           MLE   Df Model:                            1
         Date:                Sun, 05 May 2019   Pseudo R-squ.:                -0.2005
         Time:                        22:35:45   Log-Likelihood:                -4503.7
         converged:                       True   LL-Null:                       -3751.6
```

```
                                          LLR p-value:                    1.000
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975
------------------------------------------------------------------------------
GenderFactorized      -1.1266      0.047    -24.116      0.000      -1.218      -1.03
EducationFactorized   -0.6256      0.021    -29.900      0.000      -0.667      -0.58
==============================================================================
"""
```

By looking at the p-value, both the 2 input variables are seems to have significant relationships with the output variable Engaged. By looking at the coefficients, both are negative correlated with the output. This suggests that male customers, encoded with 1 in the GenderFactorized, are less likely to be engaged with marketing calls than female customers, encoded with 0. Similarly, the higher the customers' education levels are, the less likely that will be engaged with marketing calls.

5 5. Regression Analysis with both Continuous and Categorical Variables

```
In [47]: logit = sm.Logit(
             df['Engaged'],
             df[['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',
                 'GenderFactorized',
                 'EducationFactorized'
             ]]
         )

In [48]: logit_fit = logit.fit()

Optimization terminated successfully.
         Current function value: 0.420810
         Iterations 6


In [49]: logit_fit.summary()

Out[49]: <class 'statsmodels.iolib.summary.Summary'>
         """
                                 Logit Regression Results
         ==============================================================================
         Dep. Variable:                Engaged   No. Observations:                 9134
```

```
Model:                            Logit   Df Residuals:                     9124
Method:                             MLE   Df Model:                            9
Date:                  Sun, 05 May 2019   Pseudo R-squ.:                -0.02454
Time:                          22:35:46   Log-Likelihood:                -3843.7
converged:                         True   LL-Null:                       -3751.6
                                          LLR p-value:                    1.000
================================================================================
                                   coef    std err          z      P>|z|      [0.025
--------------------------------------------------------------------------------
Customer Lifetime Value      -6.909e-06   5.03e-06     -1.373      0.170   -1.68e-05
Income                        -2.59e-06   1.04e-06     -2.494      0.013   -4.63e-06
Monthly Premium Auto            -0.0081      0.001     -6.526      0.000      -0.011
Months Since Last Claim         -0.0194      0.003     -6.858      0.000      -0.025
Months Since Policy Inception   -0.0057      0.001     -5.827      0.000      -0.008
Number of Open Complaints       -0.0813      0.034     -2.376      0.017      -0.148
Number of Policies              -0.0781      0.013     -6.114      0.000      -0.103
Total Claim Amount               0.0001      0.000      0.943      0.346      -0.000
GenderFactorized                -0.1500      0.058     -2.592      0.010      -0.263
EducationFactorized             -0.0070      0.027     -0.264      0.792      -0.059
================================================================================
"""
```

By looking at the p-value, the **'Income', 'Monthly Premium Auto', 'Months Since Last Claim', 'Months Since Policy Inception','Number of Open Complaints', 'Number of Policies', and 'GenderFactorized'** variables are significant at 0.05 significance level, and all of them have *negative relationships* with the output variable, **Engaged**.

Conclusions: The higher the income is, the less likely that the customer will be engaged with marketing calls. Alike, the policies the customer has, the less likely that the customer will be engaged. In addition, male customers are less likely to engage with marketing calls than female ones, and so on.