INDENG 242A HW1 Fengyi Chen

Problem 1

a) $RSS = \sum_{i=1}^{n} (y_i - \alpha x_i - \gamma(1-x_i))^2$

To minimize the RSS error, need to set the partial derivatives $\frac{\partial RSS}{\partial \alpha}$ and $\frac{\partial RSS}{\partial \gamma} = 0$.

$$\frac{\partial RSS}{\partial \alpha} = \sum_{i=1}^{n} 2(y_i - \alpha x_i - \gamma(1-x_i)) \cdot (-x_i)$$

$$0 = \sum_{i=1}^{n} (x_i y_i - \alpha x_i^2 - \gamma x_i (1-x_i))$$

$$\alpha \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} (x_i y_i - \gamma x_i (1-x_i))$$

$\gamma x_i(1-x_i)$ is always zero since x_i is binary $\in \{0,1\}$

$$\alpha \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i$$

$$\hat{\alpha} = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i}$$

$x_i^2 = x_i$ since $x_i \in \{0,1\}$

$$\frac{\partial RSS}{\partial \gamma} = \sum_{i=1}^{n} -2(y_i - \alpha x_i - \gamma(1-x_i))(1-x_i)$$

$$0 = \sum_{i=1}^{n} y_i(1-x_i) - \alpha x_i(1-x_i) - \gamma(1-x_i)^2$$

$\alpha x_i(1-x_i)$ is always zero since x_i is binary $\in \{0,1\}$

$$\gamma \sum_{i=1}^{n} (1-x_i)^2 = \sum_{i=1}^{n} y_i(1-x_i)$$

$$\hat{\gamma} = \frac{\sum_{i=1}^{n} y_i(1-x_i)}{\sum_{i=1}^{n} (1-x_i)}$$

$(1-x_i)^2 = (1-x_i)$ since $x_i \in \{0,1\}$

$$\frac{\partial^2 RSS}{\partial \alpha^2} = 2\sum_{i=1}^{n} x_i^2 \geq 0 \quad \text{for all } \alpha$$

$$\frac{\partial^2 RSS}{\partial \gamma^2} = 2\sum_{i=1}^{n} (1-x_i)^2 \geq 0 \quad \text{for all } \gamma$$

$$\frac{\partial^2 RSS}{\partial \alpha \partial \gamma} = 2\sum_{i=1}^{n} x_i(1-x_i) \geq 0 \quad \text{for all } \alpha, \gamma$$

b) $\varepsilon_i \overset{i.i.d}{\sim} N(0, \sigma^2)$, x_i, α, γ, and σ are all known and fixed
Thus, $y_i = \alpha x_i + \gamma(1-x_i) + \varepsilon_i$ is a random variable. $\hat{\alpha}$ and $\hat{\gamma}$ are random variables depending on y_i.

c) $E[\hat{\alpha}] = E\left[\frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i}\right]$

$= \frac{\sum_{i=1}^{n} x_i E[y_i]}{\sum_{i=1}^{n} x_i}$ since x_i is fixed and deterministic

$= \frac{\sum_{i=1}^{n} x_i (\alpha x_i + \gamma(1-x_i))}{\sum_{i=1}^{n} x_i}$ since $E[\varepsilon_i] = 0$

$= \frac{\alpha \sum_{i=1}^{n} x_i^2 + \gamma \sum_{i=1}^{n} x_i(1-x_i)}{\sum_{i=1}^{n} x_i}$ is always zero since x_i is binary $\in \{0,1\}$

$= \frac{\alpha \sum_{i=1}^{n} x_i}{\sum_{i=1}^{n} x_i}$ $x_i^2 = x_i$ since $x_i \in \{0,1\}$

$= \alpha$

$E[\hat{\gamma}] = E\left[\frac{\sum_{i=1}^{n} y_i(1-x_i)}{\sum_{i=1}^{n} (1-x_i)}\right]$

$= \frac{\sum_{i=1}^{n} E[y_i](1-x_i)}{\sum_{i=1}^{n} (1-x_i)}$ since x_i is fixed and deterministic

$= \frac{\sum_{i=1}^{n} (\alpha x_i + \gamma(1-x_i))(1-x_i)}{\sum_{i=1}^{n} (1-x_i)}$ since $E[\varepsilon_i] = 0$

$= \frac{\alpha \sum_{i=1}^{n} x_i(1-x_i) + \gamma \sum_{i=1}^{n} (1-x_i)^2}{\sum_{i=1}^{n} (1-x_i)}$ is always zero since x_i is binary $\in \{0,1\}$

$$= \frac{\gamma \sum_{i=1}^{n} (1-x_i)}{\sum_{i=1}^{n} (1-x_i)} \qquad (1-x_i)^2 = (1-x_i) \text{ since } x_i \in \{0,1\}$$

$$= \gamma$$

$$\text{var}(\hat{\alpha}) = \text{var}\left(\frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i} \right)$$

$$= \frac{\sum_{i=1}^{n} \text{var}(x_i y_i)}{\left(\sum_{i=1}^{n} x_i \right)^2} \qquad \text{since } x_i \text{ is fixed, } y_i \text{ is independent}$$
$$\text{var}(aX) = a^2 \text{var}(X)$$

$$= \frac{\sum_{i=1}^{n} x_i^2 \text{var}(y_i)}{\left(\sum_{i=1}^{n} x_i \right)^2}$$

$$= \frac{\sigma^2 \sum_{i=1}^{n} x_i}{\left(\sum_{i=1}^{n} x_i \right)^2} \qquad \text{var}(y_i) = \text{var}(\varepsilon_i) = \sigma^2 \text{ since } x_i, \alpha, \gamma \text{ are fixed}$$
$$x_i^2 = x_i \text{ since } x_i \in \{0,1\}$$

$$= \frac{\sigma^2}{\sum_{i=1}^{n} x_i}$$

$$\text{var}(\hat{\gamma}) = \text{var}\left(\frac{\sum_{i=1}^{n} y_i (1-x_i)}{\sum_{i=1}^{n} (1-x_i)} \right)$$

$$= \frac{\sum_{i=1}^{n} \text{var}(y_i (1-x_i))}{\left(\sum_{i=1}^{n} (1-x_i) \right)^2} \qquad \text{since } x_i \text{ is fixed, } y_i \text{ is independent}$$
$$\text{var}(aX) = a^2 \text{var}(X)$$

$$= \frac{\sigma^2 \sum_{i=1}^{n} (1-x_i)^2}{\left(\sum_{i=1}^{n} (1-x_i) \right)^2} \qquad \text{var}(y_i) = \text{var}(\varepsilon_i) = \sigma^2 \text{ since } x_i, \alpha, \gamma \text{ are fixed}$$

$$= \frac{\sigma^2}{\sum_{i=1}^{n} (1-x_i)} \qquad (1-x_i)^2 = (1-x_i) \text{ since } x_i \in \{0,1\}$$

d) $\hat{\alpha}$ and $\hat{\gamma}$ are statistically independent.

$$E[\hat{\alpha}\hat{\gamma}] = E\left[\frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i} \cdot \frac{\sum_{i=1}^{n} y_i (1-x_i)}{\sum_{i=1}^{n} (1-x_i)}\right]$$

$$= \frac{\sum_{i=1}^{n} x_i E[y_i]}{\sum_{i=1}^{n} x_i} \cdot \frac{\sum_{i=1}^{n} E[y_i](1-x_i)}{\sum_{i=1}^{n} (1-x_i)}$$

$$= \frac{\sum_{i=1}^{n} x_i (\alpha x_i + \gamma(1-x_i))}{\sum_{i=1}^{n} x_i} \cdot \frac{\sum_{i=1}^{n} (\alpha x_i + \gamma(1-x_i))(1-x_i)}{\sum_{i=1}^{n} (1-x_i)}$$

$x_i^2 = x_i$
$(1-x_i)^2 = (1-x_i)$
since $x_i \in \{0,1\}$

$$= \frac{\alpha \sum_{i=1}^{n} x_i^2 + \gamma \sum_{i=1}^{n} x_i(1-x_i)}{\sum_{i=1}^{n} x_i} \cdot \frac{\alpha \sum_{i=1}^{n} x_i(1-x_i) + \gamma \sum_{i=1}^{n} (1-x_i)^2}{\sum_{i=1}^{n} (1-x_i)}$$

is always zero since x_i is binary $\in \{0,1\}$

$$= \frac{\alpha \sum_{i=1}^{n} x_i}{\sum_{i=1}^{n} x_i} \cdot \frac{\gamma \sum_{i=1}^{n} (1-x_i)}{\sum_{i=1}^{n} (1-x_i)}$$

$$= \alpha\gamma$$
$$= E[\hat{\alpha}]\,E[\hat{\gamma}]$$

e) Assume $\hat{Y}_1 = \hat{\alpha}x + \hat{\gamma}(1-x)$ and $\hat{Y}_2 = \hat{\beta}_0 + \hat{\beta}_1 x$

$$\hat{Y}_1 = \hat{\alpha}x + \hat{\gamma} - \hat{\gamma}x$$
$$= (\hat{\alpha} - \hat{\gamma})x + \hat{\gamma}$$

Make $\hat{\beta}_0 = \hat{\gamma}$ and $\hat{\beta}_1 = \hat{\alpha} - \hat{\gamma}$, then $\hat{Y}_1 = \hat{Y}_2$.

Thus, the new model does not add any predictive value over the previous model. They are essentially the same.

Problem 2

In [1]:
```
# import the necessary packages
import numpy as np
import pandas as pd
import matplotlib as plt
import seaborn as sns
import statsmodels.api as sm
import statsmodels.formula.api as smf
from statsmodels.stats.outliers_influence import variance_inflation_factor

%matplotlib inline
```

In [2]:
```
# input the Honda Accord dataset
accord = pd.read_csv('Accord-242A-Spring24.csv')
```

a)

In [3]:
```
#split the training and testing dataset; cutoff point is 2018
accord_train = accord[accord['Year'] <= 2018]
accord_test = accord[accord['Year'] > 2018]

len(accord_train), len(accord_test)
```

Out[3]:
```
(60, 59)
```

In [4]:
```
# Choose the features to be used
cols_1 = ['Unemployment', 'AccordQueries', 'CPIAll', 'CPIEnergy', 'MilesTravel
X_train_1 = accord_train[cols_1]
y_train = accord_train['AccordSales']

# Add an intercept
X_train_1 = sm.add_constant(X_train_1)

# Fit the data to the model
model_1 = sm.OLS(y_train, X_train_1).fit() #ordinary least square
print(model_1.summary())

# Calculate Variance Inflation Factor for each explanatory variable
def VIF(df, columns):
    values = sm.add_constant(df[columns]).values
    num_columns = len(columns)+1
    vif = [variance_inflation_factor(values, i) for i in range(num_columns)]
    return pd.Series(vif[1:], index=columns)

VIF(X_train_1, cols_1)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:            AccordSales   R-squared:                       0.254
Model:                            OLS   Adj. R-squared:                  0.185
Method:                 Least Squares   F-statistic:                     3.683
Date:                Mon, 12 Feb 2024   Prob (F-statistic):            0.00612
Time:                        15:27:30   Log-Likelihood:                -595.60
No. Observations:                  60   AIC:                             1203.
Df Residuals:                      54   BIC:                             1216.
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          1.735e+05   1.49e+05      1.164      0.249   -1.25e+05    4.72e+05
Unemployment  -1833.6589   4680.041     -0.392      0.697   -1.12e+04    7549.259
AccordQueries   225.2323    113.727      1.980      0.053      -2.776     453.240
CPIAll        -1443.0327    807.855     -1.786      0.080   -3062.685     176.620
CPIEnergy       192.5350    120.986      1.591      0.117     -50.028     435.098
MilesTraveled     0.5860      0.417      1.406      0.165      -0.249       1.421
==============================================================================
Omnibus:                        7.924   Durbin-Watson:                   1.479
Prob(Omnibus):                  0.019   Jarque-Bera (JB):                8.969
Skew:                           0.534   Prob(JB):                       0.0113
Kurtosis:                       4.565   Cond. No.                     5.81e+07
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 5.81e+07. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Out[4]:
```
Unemployment     31.683094
AccordQueries     1.939028
CPIAll           45.304922
CPIEnergy        12.340357
MilesTraveled    18.510210
dtype: float64
```

In [5]:
```
# Choose the features to be used
cols_2 = ['AccordQueries', 'CPIAll', 'CPIEnergy', 'MilesTraveled']
X_train_2 = accord_train[cols_2]
y_train = accord_train['AccordSales']

# Add an intercept
X_train_2 = sm.add_constant(X_train_2)

# Fit the data to the model
model_2 = sm.OLS(y_train, X_train_2).fit() #ordinary least square
print(model_2.summary())
```

```
# calculate Variance Inflation Factor for each explanatory variable
VIF(X_train_2, cols_2)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:            AccordSales   R-squared:                       0.252
Model:                            OLS   Adj. R-squared:                  0.198
Method:                 Least Squares   F-statistic:                     4.636
Date:                Mon, 12 Feb 2024   Prob (F-statistic):            0.00269
Time:                        15:27:30   Log-Likelihood:                -595.69
No. Observations:                  60   AIC:                             1201.
Df Residuals:                      55   BIC:                             1212.
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          1.171e+05    3.8e+04      3.084      0.003     4.1e+04    1.93e+05
AccordQueries   229.7962    112.255      2.047      0.045       4.833     454.759
CPIAll        -1180.1984    446.651     -2.642      0.011   -2075.308    -285.089
CPIEnergy       155.0753     73.567      2.108      0.040       7.643     302.507
MilesTraveled     0.5533      0.405      1.366      0.178      -0.259       1.365
==============================================================================
Omnibus:                        8.016   Durbin-Watson:                   1.465
Prob(Omnibus):                  0.018   Jarque-Bera (JB):                9.143
Skew:                           0.537   Prob(JB):                       0.0103
Kurtosis:                       4.583   Cond. No.                     1.49e+07
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.49e+07. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Out[5]:
```
AccordQueries     1.918686
CPIAll           14.065391
CPIEnergy         4.634036
MilesTraveled    17.765021
dtype: float64
```

In [6]:
```
# Choose the features to be used
cols_3 = ['AccordQueries', 'CPIAll', 'CPIEnergy']
X_train_3 = accord_train[cols_3]
y_train = accord_train['AccordSales']

# Add an intercept
X_train_3 = sm.add_constant(X_train_3)

# Fit the data to the model
model_3 = sm.OLS(y_train, X_train_3).fit() #ordinary least square
```

```
print(model_3.summary())

# calculate Variance Inflation Factor for each explanatory variable
VIF(X_train_3, cols_3)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:            AccordSales   R-squared:                       0.227
Model:                            OLS   Adj. R-squared:                  0.185
Method:                 Least Squares   F-statistic:                     5.475
Date:                Mon, 12 Feb 2024   Prob (F-statistic):            0.00227
Time:                        15:27:30   Log-Likelihood:                -596.69
No. Observations:                  60   AIC:                             1201.
Df Residuals:                      56   BIC:                             1210.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          1.419e+05   3.36e+04      4.224      0.000    7.46e+04    2.09e+05
AccordQueries   245.2054    112.546      2.179      0.034      19.749     470.662
CPIAll         -610.9156    161.762     -3.777      0.000    -934.963    -286.868
CPIEnergy        67.0291     35.720      1.877      0.066      -4.527     138.585
==============================================================================
Omnibus:                        8.025   Durbin-Watson:                   1.390
Prob(Omnibus):                  0.018   Jarque-Bera (JB):                9.328
Skew:                           0.526   Prob(JB):                      0.00943
Kurtosis:                       4.620   Cond. No.                     1.66e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.66e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Out[6]:
```
AccordQueries    1.899305
CPIAll           1.816792
CPIEnergy        1.075859
dtype: float64
```

(i)

The linear regression equation (approximated to two decimal places) produced by the final selected model is monthly AccordSales = 141,900.00 + 245.21 *AccordQueries - 610.92* CPIAll + 67.03* CPIEnergy in the US. The monthly Accord sales is 141,900 in the US if all the independent varibales are zero. One more Accord query on Google holding all other features constant will lead to approximately 245 more sales of Accord in a given month. A unit increase in the consumer price index for all products holding all other features constant will lead to approximately 610 fewer sales of Accord in a given month. A unit increase in the

consumer price index for the energy sector holding all other features constant will lead to approximately 67 more sales of Accord in a given month.

ii)

model_1 includes all five features, Unemployment, AccordQueries， CPIAll, CPIEnergy, and MilesTraveled. Except AccordQueries, all other four features have VIF higher than 5. Given Unemployment has the highest p-value around 0.7, I decided to remove Unemployment in model_2 and see if the model performance improves.

model_2 includes AccordQueries, CPIAll, CPIEnergy, and MilesTraveled. CPIAll and MilesTraveled have VIF higher than 5. Given MilesTraveled is the only feature that has the p-value higer than 0.05, I decided to remove MilesTraveled in model_3 and see if the model performance improves. Comparing to model_1, model_2 has higher adjusted R-squared at 0.198 vs 0.185.

model_3 includes AccordQueries, CPIAll, and CPIEnergy. All the VIFs are lower than 5. Although CPIEnergy has the p-value slightly higer than 0.05, I decided to retain it. Comparing to model_2, although model_3 has lower adjusted R-squared at 0.185 vs 0.198, I decided to choose model_3 as the final selected model given reasonable values of VIF, p-value, and adjusted R-suqared comparatively.

iii)

The number of Accord Queries and the consumer price index for the energy sector in the US have a positive impact on the sales when these two independent variables increase. It makes sense as a higher number of Accord Queries in the US indicates there is a higher interest particularly in Accord. However, the reasoning is unclear that the consumer price index for the energy sector has a positive impact. From commen sense, if the gasoline price goes up, which is included in the consumer price index for the energy sector, consumers should demand less gas-powered vehicles but are more likely to switch to electric vehicles. Accord has both gas-powered version and hybrid-electric version and the decomposition of these two versions are not available from the dataset.

On the other hand, the consumer price index for all products has a negative impact on the sales when the independent varibales increases. It makes sense as a higher consumer price index for all products (i.e., inflation) usually indicates an economic recession. Consequently, consumers have less real disposable income and they are less likely to buy Accord and any other goods and services.

iv)

The model performance is very poor given the R-squared is only 0.227 and the adjusted R-squared is only 0.185.

b)

In [7]:
```
# Fit the data to the model
model_4 = smf.ols(formula='AccordSales ~ MonthFactor + AccordQueries + Unemplo
                  data=accord_train).fit() #ordinary least square
print(model_4.summary())
```

OLS Regression Results

Dep. Variable:	AccordSales	R-squared:	0.748
Model:	OLS	Adj. R-squared:	0.654
Method:	Least Squares	F-statistic:	7.982
Date:	Mon, 12 Feb 2024	Prob (F-statistic):	2.66e-08
Time:	15:27:30	Log-Likelihood:	-563.04
No. Observations:	60	AIC:	1160.
Df Residuals:	43	BIC:	1196.
Df Model:	16		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	9.775e+04	1e+05	0.977	0.334	-1.04e+05	3e+05
MonthFactor[T.August]	8697.6749	2713.554	3.205	0.003	3225.272	1.42e+04
MonthFactor[T.December]	3256.3031	2372.470	1.373	0.177	-1528.240	8040.846
MonthFactor[T.February]	-4825.4928	2587.989	-1.865	0.069	-1e+04	393.684
MonthFactor[T.January]	-8189.7564	2380.860	-3.440	0.001	-1.3e+04	-3388.294
MonthFactor[T.July]	3894.8622	2823.701	1.379	0.175	-1799.674	9589.398
MonthFactor[T.June]	1536.5375	2480.984	0.619	0.539	-3466.843	6539.918
MonthFactor[T.March]	430.1538	2582.819	0.167	0.869	-4778.598	5638.906
MonthFactor[T.May]	5573.2245	2463.065	2.263	0.029	605.980	1.05e+04
MonthFactor[T.November]	-1493.6532	2471.136	-0.604	0.549	-6477.174	3489.867
MonthFactor[T.October]	-25.6309	2305.022	-0.011	0.991	-4674.151	4622.889
MonthFactor[T.September]	3046.1712	2378.322	1.281	0.207	-1750.173	7842.515
AccordQueries	10.6620	144.801	0.074	0.942	-281.357	302.681
Unemployment	762.8155	3257.607	0.234	0.816	-5806.776	7332.407
CPIAll	-639.8070	627.629	-1.019	0.314	-1905.542	625.928
CPIEnergy	67.5163	95.714	0.705	0.484	-125.508	260.541
MilesTraveled	0.2497	0.388	0.643	0.524	-0.533	1.033

Omnibus:	11.344	Durbin-Watson:	1.509
Prob(Omnibus):	0.003	Jarque-Bera (JB):	19.675
Skew:	0.549	Prob(JB):	5.34e-05
Kurtosis:	5.582	Cond. No.	5.99e+07

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correct

```
ly specified.
[2] The condition number is large, 5.99e+07. This might indicate that there ar
e
strong multicollinearity or other numerical problems.
```

i)

The linear regression equation (approximated to two decimal places) produced by the final selected model is monthly AccordSales = 97,750 + 8697.67 *[if it is August] + 3256.30* [if it is December] - 4825.49 *[if it is February] - 8189.76* [if it is January] + 3894.86 *[if it is July] +1536.54* [if it is June] + 430.15 *[if it is March] + 5573.22* [if it is May] - 1493.65 *[if it is November] - 25.63* [if it is October] + 3046.17 *[if it is September] + 762.82* Unemployment + 10.66 *AccordQueries - 639.81* CPIAll + 67.52 *CPIEnergy + 0.25* MilesTraveled.

The monthly AccordSales in April is 97,750 if all other features are zero.

The coefficient of each of the MonthFactor dummy varibale is the monthly AccordSales difference comparing to the month of April holding all other features constant. For example, if it is August, the August dummy varibale is 1 and all other MonthFactor dummy varibales are all 0. The monthly AccordSales in August will be around 8697 higher than that in April, leading to a monthly sale of 106,447 in August.

A percentage increase in the unemployment rate holding all other features constant will lead to approximately 762 more sales of Accord in a given month. One more Accord query on Google holding all other features constant will lead to approximately 10 more sales of Accord in a given month. A unit increase in the consumer price index for all products holding all other features constant will lead to approximately 639 fewer sales of Accord in a given month. A unit increase in the consumer price index for the energy sector holding all other features constant will lead to approximately 67 more sales of Accord in a given month. A unit increase in the MilesTraveled holding all other features constant will lead to approximately 0.2 more sales of Accord in a given month.

ii)

The training set R-squared for this model is 0.748. Significant varibales (i,e, p-value is less than 0.05) include MonthFactor[T.August], MonthFactor[T.January], and MonthFactor[T.May].

iii)

I think adding the independent varibale MonthFactor improves the quality of the model as the R-squared of model_4 improves signifantly from model_3, 0.748 vs. 0.227.

iv)

We can model seasonality according to quarters instead of months, i.e., grouping months into quarter 1, 2, 3, and 4.

c)

In [8]:
```
# Fit the data to the model
model_5 = smf.ols(formula='AccordSales ~ MonthFactor + Unemployment + CPIAll +
                  data=accord_train).fit() #ordinary least square
print(model_5.summary())
```

OLS Regression Results

Dep. Variable:	AccordSales	R-squared:	0.748
Model:	OLS	Adj. R-squared:	0.662
Method:	Least Squares	F-statistic:	8.711
Date:	Mon, 12 Feb 2024	Prob (F-statistic):	8.74e-09
Time:	15:27:30	Log-Likelihood:	-563.05
No. Observations:	60	AIC:	1158.
Df Residuals:	44	BIC:	1192.
Df Model:	15		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	9.732e+04	9.87e+04	0.986	0.330	-1.02e+05	2.96e+05
MonthFactor[T.August]	8795.3316	2340.442	3.758	0.001	4078.481	1.35e+04
MonthFactor[T.December]	3184.2821	2136.888	1.490	0.143	-1122.333	7490.897
MonthFactor[T.February]	-4769.6746	2446.346	-1.950	0.058	-9699.962	160.613
MonthFactor[T.January]	-8172.3008	2342.101	-3.489	0.001	-1.29e+04	-3452.107
MonthFactor[T.July]	4023.9196	2188.701	1.838	0.073	-387.118	8434.957
MonthFactor[T.June]	1608.9666	2251.760	0.715	0.479	-2929.157	6147.091
MonthFactor[T.March]	505.0918	2346.868	0.215	0.831	-4224.709	5234.893
MonthFactor[T.May]	5636.0489	2284.300	2.467	0.018	1032.345	1.02e+04
MonthFactor[T.November]	-1527.0102	2401.643	-0.636	0.528	-6367.203	3313.183
MonthFactor[T.October]	-29.9082	2278.098	-0.013	0.990	-4621.113	4561.296
MonthFactor[T.September]	3048.6566	2351.052	1.297	0.201	-1689.577	7786.890
Unemployment	711.2881	3145.391	0.226	0.822	-5627.831	7050.407
CPIAll	-647.2279	612.443	-1.057	0.296	-1881.525	587.070
CPIEnergy	69.1862	91.931	0.753	0.456	-116.089	254.461
MilesTraveled	0.2611	0.353	0.740	0.463	-0.449	0.972

Omnibus:	11.335	Durbin-Watson:	1.506
Prob(Omnibus):	0.003	Jarque-Bera (JB):	19.467
Skew:	0.554	Prob(JB):	5.93e-05
Kurtosis:	5.561	Cond. No.	5.97e+07

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 5.97e+07. This might indicate that there ar

```
e
strong multicollinearity or other numerical problems.
```

In [9]:
```
# Fit the data to the model
model_6 = smf.ols(formula='AccordSales ~ MonthFactor + Unemployment + CPIAll +
                  data=accord_train).fit() #ordinary least square
print(model_6.summary())
```

OLS Regression Results

Dep. Variable:	AccordSales	R-squared:	0.745
Model:	OLS	Adj. R-squared:	0.666
Method:	Least Squares	F-statistic:	9.388
Date:	Mon, 12 Feb 2024	Prob (F-statistic):	3.53e-09
Time:	15:27:30	Log-Likelihood:	-563.42
No. Observations:	60	AIC:	1157.
Df Residuals:	45	BIC:	1188.
Df Model:	14		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	9.016e+04	9.78e+04	0.922	0.361	-1.07e+05	2.87e+05
MonthFactor[T.August]	8078.1900	2119.924	3.811	0.000	3808.444	1.23e+04
MonthFactor[T.December]	3166.6081	2126.003	1.489	0.143	-1115.382	7448.598
MonthFactor[T.February]	-5658.6995	2120.720	-2.668	0.011	-9930.050	-1387.349
MonthFactor[T.January]	-8895.3588	2118.108	-4.200	0.000	-1.32e+04	-4629.270
MonthFactor[T.July]	3673.6399	2126.214	1.728	0.091	-608.775	7956.054
MonthFactor[T.June]	1077.3065	2123.476	0.507	0.614	-3199.594	5354.207
MonthFactor[T.March]	-228.5422	2116.765	-0.108	0.915	-4491.925	4034.841
MonthFactor[T.May]	5020.0755	2116.734	2.372	0.022	756.755	9283.396
MonthFactor[T.November]	-2343.1348	2123.053	-1.104	0.276	-6619.183	1932.914
MonthFactor[T.October]	-591.5583	2137.301	-0.277	0.783	-4896.304	3713.188
MonthFactor[T.September]	2336.0443	2134.256	1.095	0.280	-1962.568	6634.656
Unemployment	1306.6247	3025.594	0.432	0.668	-4787.235	7400.484
CPIAll	-298.4572	389.470	-0.766	0.447	-1082.889	485.975
CPIEnergy	16.8028	58.412	0.288	0.775	-100.844	134.450

Omnibus:	11.971	Durbin-Watson:	1.424
Prob(Omnibus):	0.003	Jarque-Bera (JB):	21.013
Skew:	0.586	Prob(JB):	2.74e-05
Kurtosis:	5.652	Cond. No.	7.30e+04

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 7.3e+04. This might indicate that there are strong multicollinearity or other numerical problems.

In [10]:
```
# Fit the data to the model
model_7 = smf.ols(formula='AccordSales ~ MonthFactor + Unemployment + CPIAll',
                  data=accord_train).fit() #ordinary least square
print(model_7.summary())
```

OLS Regression Results

Dep. Variable:	AccordSales	R-squared:	0.744
Model:	OLS	Adj. R-squared:	0.672
Method:	Least Squares	F-statistic:	10.31
Date:	Mon, 12 Feb 2024	Prob (F-statistic):	1.10e-09
Time:	15:27:30	Log-Likelihood:	-563.47
No. Observations:	60	AIC:	1155.
Df Residuals:	46	BIC:	1184.
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	6.521e+04	4.47e+04	1.459	0.151	-2.47e+04	1.55e+05
MonthFactor[T.August]	8102.4013	2097.027	3.864	0.000	3881.305	1.23e+04
MonthFactor[T.December]	3156.7234	2104.425	1.500	0.140	-1079.264	7392.711
MonthFactor[T.February]	-5696.3335	2095.471	-2.718	0.009	-9914.299	-1478.368
MonthFactor[T.January]	-8886.8303	2096.679	-4.239	0.000	-1.31e+04	-4666.435
MonthFactor[T.July]	3727.7015	2096.670	1.778	0.082	-492.676	7948.079
MonthFactor[T.June]	1126.3039	2095.424	0.538	0.594	-3091.566	5344.174
MonthFactor[T.March]	-245.5634	2094.735	-0.117	0.907	-4462.046	3970.920
MonthFactor[T.May]	5037.8273	2094.633	2.405	0.020	821.551	9254.104
MonthFactor[T.November]	-2338.9184	2101.729	-1.113	0.272	-6569.480	1891.644
MonthFactor[T.October]	-522.2060	2102.380	-0.248	0.805	-4754.078	3709.666
MonthFactor[T.September]	2404.2308	2099.798	1.145	0.258	-1822.444	6630.906
Unemployment	2111.8288	1137.002	1.857	0.070	-176.837	4400.495
CPIAll	-197.0452	163.880	-1.202	0.235	-526.918	132.828

Omnibus:	12.354	Durbin-Watson:	1.421
Prob(Omnibus):	0.002	Jarque-Bera (JB):	22.284
Skew:	0.597	Prob(JB):	1.45e-05
Kurtosis:	5.736	Cond. No.	2.53e+04

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.53e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

In [11]:
```
# Fit the data to the model
model_8 = smf.ols(formula='AccordSales ~ MonthFactor + Unemployment',
                  data=accord_train).fit() #ordinary least square
print(model_8.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:            AccordSales   R-squared:                       0.736
Model:                            OLS   Adj. R-squared:                  0.669
Method:                 Least Squares   F-statistic:                     10.94
Date:                Mon, 12 Feb 2024   Prob (F-statistic):           6.20e-10
Time:                        15:27:30   Log-Likelihood:                -564.40
No. Observations:                  60   AIC:                             1155.
Df Residuals:                      47   BIC:                             1182.
Df Model:                          12
Covariance Type:            nonrobust
==============================================================================
                                 coef    std err          t      P>|t|      [0.0
25      0.975]
------------------------------------------------------------------------------
--------------
Intercept                  1.161e+04   3111.095      3.732      0.001    5350.6
75    1.79e+04
MonthFactor[T.August]      8051.7360   2106.521      3.822      0.000    3813.9
61    1.23e+04
MonthFactor[T.December]    3175.7761   2114.320      1.502      0.140   -1077.6
86    7429.238
MonthFactor[T.February]   -5669.6240   2105.266     -2.693      0.010   -9904.8
72   -1434.376
MonthFactor[T.January]    -8797.2240   2105.266     -4.179      0.000    -1.3e+
04   -4561.976
MonthFactor[T.July]        3747.7360   2106.521      1.779      0.082    -490.0
39    7985.511
MonthFactor[T.June]        1105.6240   2105.266      0.525      0.602   -3129.6
24    5340.872
MonthFactor[T.March]       -217.3120   2104.512     -0.103      0.918   -4451.0
43    4016.419
MonthFactor[T.May]         5024.5120   2104.512      2.387      0.021     790.7
81    9258.243
MonthFactor[T.November]   -2379.5359   2111.399     -1.127      0.265   -6627.1
22    1868.050
MonthFactor[T.October]     -515.0319   2112.317     -0.244      0.808   -4764.4
66    3734.402
MonthFactor[T.September]   2399.6560   2109.728      1.137      0.261   -1844.5
69    6643.881
Unemployment               3315.2002    542.105      6.115      0.000    2224.6
26    4405.774
==============================================================================
Omnibus:                        8.701   Durbin-Watson:                   1.381
Prob(Omnibus):                  0.013   Jarque-Bera (JB):               11.699
Skew:                           0.495   Prob(JB):                      0.00288
Kurtosis:                       4.923   Cond. No.                         64.6
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correct
ly specified.
```

AccordQueries was removed from model_4 to form model_5 as it has the second highest p-value at 0.942 (MonthFactor[T.October] has highest p-value at 0.942) comparing with other features. model_5 has R-squared at 0.748 and adjusted R-squared at 0.662.

MilesTraveled was removed from model_5 to form model_6 as it has a high p-value at 0.463 comparing with other features and the coefficient is close to zero. model_6 has R-squared at 0.745 and adjusted R-squared at 0.666.

CPIEnergy was removed from model_6 to form model_7 as it has the highest p-value other than some of the MonthFactor dummy variables comparing with other features. model_7 has R-squared at 0.744 and adjusted R-squared at 0.672.

CPIAll was removed from model_7 to form model_8 as it has the highest p-value other than some of the MonthFactor dummy variables comparing with other features. model_8 has R-squared at 0.736 and adjusted R-squared at 0.669.

As a result, model_6, which includes MonthFactor, Unemployment, CPIAll, and CPIEnergy as features, is the final model selected given it has reasonable R-squared and adjusted R-squared among the four models compared above. Although it is not the simpliest model, it retains good balance between simplicity and model fittness.

In [12]:
```
def OSR2(model, df_train, df_test, dependent_var):

    y_test = df_test[dependent_var]
    y_pred = model.predict(df_test)
    SSE = np.sum((y_test - y_pred)**2)
    SST = np.sum((y_test - np.mean(df_train[dependent_var]))**2)

    return 1 - SSE/SST
```

In [13]:
```
OSR2(model_6, accord_train, accord_test, 'AccordSales')
```

Out[13]:
```
0.5658545696406424
```

The training set R-squared is 0.745 and the OSR-squared is 0.566. The OSR-squared is lower than the R-squared, which makes sense as the OLS regression model is fitted to the training data and usually out-of-sample prediction has a lower R-squared. However, the OSR-squared value of 0.566 is at a concerning level as it indicates the regression model has limited predictibility. The significant drop is probably due to the pandemic, which had a longlasting period from 2020 to 2023. The inflation and unemployement rate were unusually high ihe US, affecting Unemployment, CPIAll, and CPIEnergy. The training set does not include any similar scenario in the dataset, possibility leading to the poor prediction of a rare event.

d)

In [14]:
```
DFF = pd.read_csv('DFF.csv')
DFF.head()
```

Out[14]:

	DATE	DFF
0	2014-01-01	0.071613
1	2014-02-01	0.066429
2	2014-03-01	0.078065
3	2014-04-01	0.090333
4	2014-05-01	0.087097

In [15]:
```
accord_new = accord
accord_new['DFF']= DFF['DFF']
accord_new.head()
```

Out[15]:

	MonthNumeric	MonthFactor	Year	AccordSales	Unemployment	AccordQueries	CPIAll	CI
0	1	January	2014	20604	6.6	69	235.288	
1	2	February	2014	24622	6.7	74	235.547	
2	3	March	2014	33962	6.7	79	236.028	
3	4	April	2014	34124	6.2	74	236.468	
4	5	May	2014	39637	6.3	75	236.918	

In [16]:
```
#split the training and testing dataset; cutoff point is 2018
accord_train_new = accord_new[accord_new['Year'] <= 2018]
accord_test_new = accord_new[accord_new['Year'] > 2018]
```

In [17]:
```
# Fit the data to the model
model_9 = smf.ols(formula='AccordSales ~ MonthFactor + Unemployment + CPIAll +
                  data=accord_train_new).fit() #ordinary least square
print(model_9.summary())
```

OLS Regression Results

Dep. Variable:	AccordSales	R-squared:	0.747
Model:	OLS	Adj. R-squared:	0.661
Method:	Least Squares	F-statistic:	8.677
Date:	Mon, 12 Feb 2024	Prob (F-statistic):	9.26e-09
Time:	15:27:30	Log-Likelihood:	-563.13
No. Observations:	60	AIC:	1158.
Df Residuals:	44	BIC:	1192.
Df Model:	15		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	3.128e+04	1.34e+05	0.234	0.816	-2.39e+05	3.01e+05
MonthFactor[T.August]	7958.6360	2141.678	3.716	0.001	3642.368	1.23e+04
MonthFactor[T.December]	3169.8796	2139.843	1.481	0.146	-1142.690	7482.450
MonthFactor[T.February]	-5666.2849	2134.552	-2.655	0.011	-9968.192	-1364.377
MonthFactor[T.January]	-8839.0195	2133.662	-4.143	0.000	-1.31e+04	-4538.905
MonthFactor[T.July]	3634.9056	2140.884	1.698	0.097	-679.762	7949.573
MonthFactor[T.June]	987.2640	2141.805	0.461	0.647	-3329.259	5303.787
MonthFactor[T.March]	-234.8331	2130.561	-0.110	0.913	-4528.696	4059.030
MonthFactor[T.May]	4916.2899	2136.517	2.301	0.026	610.422	9222.158
MonthFactor[T.November]	-2542.4752	2158.891	-1.178	0.245	-6893.434	1808.484
MonthFactor[T.October]	-773.6139	2169.472	-0.357	0.723	-5145.898	3598.670
MonthFactor[T.September]	2135.9488	2170.217	0.984	0.330	-2237.837	6509.735
Unemployment	1296.3845	3045.323	0.426	0.672	-4841.061	7433.830
CPIAll	-50.2201	548.065	-0.092	0.927	-1154.773	1054.333
CPIEnergy	18.5634	58.855	0.315	0.754	-100.050	137.177
DFF	-2097.8051	3236.876	-0.648	0.520	-8621.300	4425.690

Omnibus:	11.615	Durbin-Watson:	1.436
Prob(Omnibus):	0.003	Jarque-Bera (JB):	20.048
Skew:	0.571	Prob(JB):	4.43e-05
Kurtosis:	5.592	Cond. No.	9.93e+04

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.93e+04. This might indicate that there ar

```
e
strong multicollinearity or other numerical problems.
```

In [18]:
```
OSR2(model_9, accord_train_new, accord_test_new, 'AccordSales')
```

Out[18]:
```
0.08004426754171201
```

Monthly Average Fed Fund Effective Rate(%) (i.e., DFF) from January 2014 to November 2023 is added as an additional feature to the final model from part (c). I hypothesize it might be related to Honda sales as the Fed Fund Rate level impacts personal loan rate in general, which would likely to affect people's incentive to buy cars. In particular, a higher Fed Fund Rate will likely to increase personal loan rate, which makes it more expensive to borrow from the banks and more difficult to find a cheaper refinancing option.

The new model, model_9, including DFF in addtion to the features from the final model, model_6, from part (c) has a slightly better R-squared value at 0.747 vs. 0.745. However, the OSR-squared shows a significnt drop from 0.566 to 0.080, indicating that DFF is not providing any new predictive value but worsening the out-of-sample model performance. The p-value of DFF is 0.520, which is much higher than 0.05, highlighting that it is not a statistically significant feature. The coefficient of -2097.81 is align with my expectation of the feature.

The possible reasons to explain the phenomenum are: 1) model_9 has higher multicollinearity than model_6, i.e., DFF is highly correlated with unemployment and/or CPI; 2) the DFF in the testing data is much unusual as in the training data given the FED consecutively raised the DFF to control the inflation of the US during the pandemic, so the black swan event limited the out-of-sample predictibility.