IEOR 142

Homework Assignment 1

1. a) $Y = \beta X + \epsilon$

$f(\beta) = \|Y - X\beta\|_2^2 = (Y - X\beta)^T(Y - X\beta) = Y^TY - 2Y^TX\beta + \beta^TX^TX\beta$

$\nabla_\beta f(\beta) = -2X^TY + 2X^TX\beta$

$-2X^TY + 2X^TX\hat{\beta} = 0$

$\hat{\beta} = (X^TX)^{-1}(X^TY)$

b) So ϵ is a random variable. And no matter the difference where our model is fixed or random, $\hat{\beta}$ is a random variable because X here is a constant because it is observed so in turn, $\hat{\beta}$ can be a random variable. A random variable has the probability distribution of the likelihood of any possible values

c) $\hat{\beta} = (X^TX)^{-1}X^TY = \frac{1}{n}\sum Y_i = \bar{Y} = \hat{\beta}_0$

$\text{var}[\hat{\beta}] = (X^TX)^{-1}\sigma^2 = \frac{\sigma^2}{n}$ ↙ derivation

$E(\hat{\beta}) = E\{(X^TX)^{-1}X^TY\} = (X^TX)^{-1}X^T E(Y)$

$= (X^TX)^{-1}X^TX\beta = I\beta = \beta$

$\text{Var}(\hat{\beta}) = \text{var}\{(X^TX)^{-1}X^TY\}$

$= (X^TX)^{-1}X^T\text{Var}(Y)X(X^TX)^{-1}$

$= \sigma^2(X^TX)^{-1}(X^T)IX(X^TX)^{-1} = \sigma^2(X^TX)^{-1}$

$\hat{\beta} = (X^TX)^{-1}(X^T(X\beta + \epsilon))$

$E\hat{\beta} = E[(X^TX)^{-1}(X^T(X\beta + \epsilon))]$

d) $\epsilon_1 \ldots \epsilon_n$ are normally distributed

$\hat{\beta}$ is also normal because it can be written as a linear combination.

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

Question 2

a

In [2]:
```
data = pd.read_csv("RAV4-142-Spring2021.csv")
data.info()
data.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 120 entries, 0 to 119
Data columns (total 8 columns):
MonthNumeric    120 non-null int64
MonthFactor     120 non-null object
Year            120 non-null int64
RAV4Sales       120 non-null int64
Unemployment    120 non-null float64
RAV4Queries     120 non-null int64
CPIAll          120 non-null float64
CPIEnergy       120 non-null float64
dtypes: float64(3), int64(4), object(1)
memory usage: 7.6+ KB
```

Out[2]:

	MonthNumeric	MonthFactor	Year	RAV4Sales	Unemployment	RAV4Queries	CPIAll	CPIEnergy
0	1	January	2011	11196	9.1	29	221.187	229.258
1	2	February	2011	12562	9.0	29	221.898	232.068
2	3	March	2011	16082	9.0	29	223.046	240.079
3	4	April	2011	15586	9.1	27	224.093	247.977
4	5	May	2011	8624	9.0	28	224.806	250.744

In [3]:
```
# from sklearn.model_selection import train_test_split
train = data[data['Year'] <= 2016]
test = data[data['Year'] > 2016]
len(train), len(test)
```

Out[3]: (72, 48)

In [10]:
```
import statsmodels.formula.api as smf
ols = smf.ols(formula='RAV4Sales~ RAV4Queries ',data=train)
model1 =ols.fit()
print(model1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              RAV4Sales   R-squared:                       0.725
Model:                            OLS   Adj. R-squared:                  0.721
Method:                 Least Squares   F-statistic:                     184.1
Date:                Thu, 11 Feb 2021   Prob (F-statistic):           2.79e-21
Time:                        11:49:03   Log-Likelihood:                -696.68
No. Observations:                  72   AIC:                             1397.
Df Residuals:                      70   BIC:                             1402.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept  -6727.5531   2040.356     -3.297      0.002   -1.08e+04   -2658.191
RAV4Queries   682.3590     50.285     13.570      0.000     582.069     782.649
==============================================================================
Omnibus:                        2.895   Durbin-Watson:                   1.402
Prob(Omnibus):                  0.235   Jarque-Bera (JB):                2.446
Skew:                          -0.153   Prob(JB):                        0.294
Kurtosis:                       3.850   Cond. No.                         180.
==============================================================================

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

The equation is RAV4Sales=682.3590∗RAV4Queries. In order to see how different varibales affected the RAV4Sales, I formulated a linear regression model using the different variables like Unemployment, RAV4Queries, CPIEnergy, and CPIAll. After trying the different ones, RAV4Queries have the best influence on RAV4sales and its coefficient is positve which indicates that there will be

an increase in sales, whereas Uemployment and CPIEnergy had negative coefficients. CPIAll also had a positive coefficient but its R squared value was less than RAV4Queries'. The model right here has a R squared value of 0.725.

b

In [11]:
```
ols = smf.ols(formula='RAV4Sales~ Unemployment+RAV4Queries+CPIEnergy+CPIAll+
model1 =ols.fit()
print(model1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              RAV4Sales   R-squared:                       0.884
Model:                            OLS   Adj. R-squared:                  0.853
Method:                 Least Squares   F-statistic:                     28.51
Date:                Thu, 11 Feb 2021   Prob (F-statistic):           8.55e-21
Time:                        12:03:21   Log-Likelihood:                -665.48
No. Observations:                  72   AIC:                             1363.
Df Residuals:                      56   BIC:                             1399.
Df Model:                          15
Covariance Type:            nonrobust
==============================================================================
                                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept                   7.754e+04   8.82e+04      0.879      0.383   -9.92e+04    2.54e+05
MonthFactor[T.August]       2422.0989   1679.828      1.442      0.155    -943.001    5787.199
MonthFactor[T.December]     1885.2249   1704.333      1.106      0.273   -1528.965    5299.414
MonthFactor[T.February]    -2922.8349   1644.025     -1.778      0.081   -6216.214     370.544
MonthFactor[T.January]     -4543.8071   1648.557     -2.756      0.008   -7846.264   -1241.350
MonthFactor[T.July]         -193.7079   1687.715     -0.115      0.909   -3574.607    3187.191
MonthFactor[T.June]        -1426.1733   1666.576     -0.856      0.396   -4764.726    1912.380
MonthFactor[T.March]         466.8540   1639.891      0.285      0.777   -2818.243    3751.951
MonthFactor[T.May]          2010.0694   1640.329      1.225      0.226   -1275.904    5296.043
MonthFactor[T.November]    -1540.1770   1687.060     -0.913      0.365   -4919.765    1839.411
MonthFactor[T.October]     -1808.4239   1695.135     -1.067      0.291   -5204.188    1587.340
MonthFactor[T.September]   -1879.2594   1651.944     -1.138      0.260   -5188.501    1429.982
Unemployment               -3687.3648   1437.100     -2.566      0.013   -6566.223    -808.507
RAV4Queries                  228.4423    116.205      1.966      0.054
```

```
-4.343     461.228
CPIEnergy                     1.1895     40.426      0.029      0.977
-79.794      82.173
CPIAll                     -175.4430    379.958     -0.462      0.646    -
936.590     585.704
==============================================================================
=====
Omnibus:                        7.146   Durbin-Watson:
1.287
Prob(Omnibus):                  0.028   Jarque-Bera (JB):                1
1.806
Skew:                           0.206   Prob(JB):                       0.
00273
Kurtosis:                       4.941   Cond. No.                      8.7
0e+04
==============================================================================
=====

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is co
rrectly specified.
[2] The condition number is large, 8.7e+04. This might indicate that ther
e are
strong multicollinearity or other numerical problems.
```

The equation is RAV4Sales=228.4423∗RAV4Queries -3687.3648∗Unemployment + 1.1895∗CPIEnergy -175.4430∗CPIAll. This model has the new independent variable MonthFactor so that the model can also take seasonality into account to predict the sales accurately. The coefficients of each monthly variable depicts either a positive or negative value and when some coefficient months like (June and May) are positive which means that they tend to affect the sales positively. All other months have negative coefficients which affect the sales in the negatively. The new R squared value is .884 and the new significant variables are RAV4 , MonthFactorAugust, MonthFactorMay, and MonthFactorDecember. Adding an independent variable is insightful for the model as we discovered new significant variables and also the R squared value increased from the previous model. I think to improve the model, we can calculate with only significant independent monthly variables.

c

In [12]:
```
ols = smf.ols(formula='RAV4Sales~ RAV4Queries+CPIEnergy+MonthFactor',data=tr
model1 =ols.fit()
print(model1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              RAV4Sales   R-squared:                       0.823
Model:                            OLS   Adj. R-squared:                  0.784
Method:                 Least Squares   F-statistic:                     20.81
Date:                Thu, 11 Feb 2021   Prob (F-statistic):           3.52e-17
Time:                        12:20:55   Log-Likelihood:                -680.67
No. Observations:                  72   AIC:                             1389.
Df Residuals:                      58   BIC:                             1421.
Df Model:                          13
Covariance Type:            nonrobust
===================================================================================
                      coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------
Intercept        8624.1725   9353.043      0.922      0.360   -1.01e+04    2.73e+04
MonthFactor[T.August]     1255.5472   2020.862      0.621      0.537   -2789.647    5300.741
MonthFactor[T.December]   3299.8868   1994.327      1.655      0.103    -692.193    7291.967
MonthFactor[T.February]  -3107.3139   1986.328     -1.564      0.123   -7083.382     868.754
MonthFactor[T.January]   -4738.9921   1991.884     -2.379      0.021   -8726.182    -751.802
MonthFactor[T.July]      -1582.4212   2017.397     -0.784      0.436   -5620.680    2455.837
MonthFactor[T.June]      -2482.5140   2004.795     -1.238      0.221   -6495.548    1530.520
MonthFactor[T.March]       322.4250   1985.441      0.162      0.872   -3651.868    4296.718
MonthFactor[T.May]        2206.3199   1985.491      1.111      0.271   -1768.071    6180.711
MonthFactor[T.November]   -283.9308   1987.790     -0.143      0.887   -4262.924    3695.062
MonthFactor[T.October]    -611.4919   1986.632     -0.308      0.759   -4588.167    3365.183
MonthFactor[T.September] -1756.3385   1989.174     -0.883      0.381   -5738.102    2225.425
RAV4Queries                586.4704     77.764      7.542      0.000     430.808     742.132
CPIEnergy                  -47.9223     28.966     -1.654      0.103    -
```

105.904 10.059

Omnibus:	5.917	Durbin-Watson:	0.973
Prob(Omnibus):	0.052	Jarque-Bera (JB):	7.536
Skew:	-0.268	Prob(JB):	0.0231
Kurtosis:	4.491	Cond. No.	5.40e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 5.4e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

In [13]:
```
ols = smf.ols(formula='RAV4Sales~ RAV4Queries+CPIEnergy+MonthFactor',data=te
model1 =ols.fit()
print(model1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              RAV4Sales   R-squared:                       0.682
Model:                            OLS   Adj. R-squared:                  0.560
Method:                 Least Squares   F-statistic:                     5.599
Date:                Thu, 11 Feb 2021   Prob (F-statistic):           2.62e-05
Time:                        14:01:34   Log-Likelihood:                -463.60
No. Observations:                  48   AIC:                             955.2
Df Residuals:                      34   BIC:                             981.4
Df Model:                          13
Covariance Type:            nonrobust
===========================================================================
                              coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------
Intercept                -4594.3439   1.27e+04     -0.361      0.720   -3.04e+04    2.13e+04
MonthFactor[T.August]     1.344e+04   3248.488      4.137      0.000    6835.804       2e+04
MonthFactor[T.December]   9486.5446   3251.640      2.917      0.006    2878.417    1.61e+04
MonthFactor[T.February]    574.8912   3229.172      0.178      0.860   -5987.576    7137.359
MonthFactor[T.January]   -2674.0244   3211.628     -0.833      0.411   -9200.838    3852.789
MonthFactor[T.July]       9630.8203   3261.708      2.953      0.006    3002.231    1.63e+04
MonthFactor[T.June]       7046.8050   3241.147      2.174      0.037     460.001    1.36e+04
MonthFactor[T.March]      2432.7239   3190.099      0.763      0.451   -4050.337    8915.784
MonthFactor[T.May]         1.01e+04   3192.822      3.162      0.003    3608.025    1.66e+04
MonthFactor[T.November]   7211.3926   3233.726      2.230      0.032     639.670    1.38e+04
MonthFactor[T.October]    6629.6823   3229.020      2.053      0.048      67.523    1.32e+04
MonthFactor[T.September]  9741.2666   3224.511      3.021      0.005    3188.271    1.63e+04
RAV4Queries                115.9049     44.866      2.583      0.014      24.727     207.083
CPIEnergy                  121.5095     60.188      2.019      0.051      -0.808     243.827
```

```
==============================================================================
=====
Omnibus:                        1.718   Durbin-Watson:
1.177
Prob(Omnibus):                  0.424   Jarque-Bera (JB):
1.158
Skew:                          -0.009   Prob(JB):
0.560
Kurtosis:                       2.239   Cond. No.                     4.3
7e+03
==============================================================================
=====

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is co
rrectly specified.
[2] The condition number is large, 4.37e+03. This might indicate that the
re are
strong multicollinearity or other numerical problems.
```

The R squared value is 0.823 and osr squared is 0.682. I think this model can be useful bc the osr squared value can help predict RAV4 Sales.

d

In [31]:
```
newd = pd.read_csv("geoMap.csv")
filterd = newd.dropna()
filterd.info()
filterd.head(10)
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 22 entries, City to Dallas
Data columns (total 1 columns):
Category: All categories    22 non-null object
dtypes: object(1)
memory usage: 352.0+ bytes
```

Out[31]:

	Category: All categories
City	Toyota RAV4: (1/1/11 - 1/1/20)
Yonkers	100
San Jose	85
Boston	71
Houston	62
Portland	62
New York	60
San Diego	59
Philadelphia	59
Los Angeles	59

In [33]:
```
# frames = [data, filterd]
# final = pd.concat(frames, axis =1)
# final.info()
# final.head()
# filterd.append(data, ignore_index=False)
city = filterd["Category: All categories"]
data = data.join(city)
```

```
---------------------------------------------------------------------------
--
ValueError                                Traceback (most recent call las
t)
<ipython-input-33-96268641aeb7> in <module>()
      5 # filterd.append(data, ignore_index=False)
      6 city = filterd["Category: All categories"]
----> 7 data = data.join(city)

/Users/gyelgireddy/anaconda/lib/python3.6/site-packages/pandas/core/fram
e.py in join(self, other, on, how, lsuffix, rsuffix, sort)
   6813         # For SparseDataFrame's benefit
   6814         return self._join_compat(other, on=on, how=how, lsuffix=l
suffix,
-> 6815                                  rsuffix=rsuffix, sort=sort)
   6816
   6817     def _join_compat(self, other, on=None, how='left', lsuffix
='', rsuffix='',

/Users/gyelgireddy/anaconda/lib/python3.6/site-packages/pandas/core/fram
e.py in _join_compat(self, other, on, how, lsuffix, rsuffix, sort)
   6828             return merge(self, other, left_on=on, how=how,
   6829                          left_index=on is None, right_index=True,
-> 6830                          suffixes=(lsuffix, rsuffix), sort=sort)
   6831         else:
   6832             if on is not None:

/Users/gyelgireddy/anaconda/lib/python3.6/site-packages/pandas/core/resha
pe/merge.py in merge(left, right, how, on, left_on, right_on, left_index,
right_index, sort, suffixes, copy, indicator, validate)
     46                          copy=copy, indicator=indicator,
     47                          validate=validate)
---> 48     return op.get_result()
     49
     50

/Users/gyelgireddy/anaconda/lib/python3.6/site-packages/pandas/core/resha
pe/merge.py in get_result(self)
    550
    551         llabels, rlabels = items_overlap_with_suffix(ldata.items,
lsuf,
--> 552                                                      rdata.items,
rsuf)
    553
    554         lindexers = {1: left_indexer} if left_indexer is not None
else {}

/Users/gyelgireddy/anaconda/lib/python3.6/site-packages/pandas/core/inter
nals/managers.py in items_overlap_with_suffix(left, lsuffix, right, rsuff
```

```
ix)
   1970         if not lsuffix and not rsuffix:
   1971             raise ValueError('columns overlap but no suffix speci
fied: '
-> 1972                              '{rename}'.format(rename=to_rename))
   1973
   1974         def lrenamer(x):

ValueError: columns overlap but no suffix specified: Index(['Category: Al
l categories'], dtype='object')
```

Struggled with joining both tables but if it worked I would do the below. The new data value I wanted to exist to our existing data frame is the city where RAV4s were sold. Based on the R squared value and OSR squared value, I would be able to tell how much the variable of city be significant or not.

In [ ]:
```
ols = smf.ols(formula='RAV4Sales~ RAV4Queries+CPIEnergy+MonthFactor+City',da
model1 =ols.fit()
print(model1.summary())
```