Econometrics 322 Lab #4

Basic OLS Regression

Prof. Paczkowski

Enter your Name in the Next Cell

Maanya Tandon

Grading Rubric

Score:___: Max(0, 20 - Total Deductions)

Content Area	Deduction	Times Deducted	Check	Comments
Abstract				
Missing	5		[ ]	
Insufficient/Wrong Focus	1		[ ]	
Data Dictionary (Metadata)				
Missing	5		[ ]	
Insufficient/Wrong Form or Wording	1		[ ]	
Graphs				
Missing	5		[ ]	
Missing Title	1		[ ]	
Missing/Wrong Labels	1		[ ]	
Pre-Lab				
Missing	5		[ ]	
Insufficient/Wrong Answer	2 Each		[ ]	
No/Incorrect/Insufficient Model Specification	2		[ ]	
No/Incorrect Statistical Hypothesis Statement	2 Each		[ ]	
Post-Lab				
Missing	5		[ ]	
Insufficient/Wrong Answer	2 Each		[ ]	
Correlations				
Missing	5		[ ]	
Insufficient/Wrong Analysis	2		[ ]	
Missing Graph	2		[ ]	
Estimations				
Missing	5		[ ]	
No or incorrect discussion/interpretation of...				
Hypothesis tests and p-values	2 Each		[ ]	
R^2	2		[ ]	
F-Statistic	2		[ ]	
Multicollinearity/VIF	2		[ ]	
Heteroskedasticity/Test	2		[ ]	
Autocorrelation/Test	2		[ ]	
No/insufficient model selection	2		[ ]	
Elasticities				
Missing	5		[ ]	
Incorrect Interpretation	2		[ ]	
Missing Summary Table	2		[ ]	
Model Portfolio			[ ]	
Missing	5		[ ]	
General Comments:				

Contents

Collaboration Policy

Back to Contents

```
1. Study groups are allowed but I expect students to understand and complete their own assignmen
ts and to hand in one
assignment per student.
2. If you worked in a group, please put the names of your study group in the following table.
3. Just like all other classes at Rutgers, the student Honor Code is taken seriously.

The submitted assignment must be your work.
```

Collaborator(s) Name(s)
name(s) here

Introduction

Back to Contents

Purpose

Back to Contents

This lab will introduce you to doing a simple OLS estimation using Statsmodels.

At the end of this lab, you will be able to:

- run an OLS estimation;
- retrieve some basic OLS relevant data.

Problem

Back to Contents

This is a repeat of the water consumption problem discussed in class. What determnines the demand for bottled water?

Assignment

Back to Contents

Use the water consumption data to estimate a simple regression model. The water consumption data was introduced at the beginning of the semester and is available on Sakai in the *Resources* tab for this lab. The unknown parameters of a demand function have to be estimated. Estimate a simple OLS model real per capita water consumption as a function of the real price per gallon. No other variables are to be used since the purpose of this lab is just to have you become familiar with commands.

Documentation

Back to Contents

Abstract

Back to Contents

In this lab, I used the water consumption data to estimate a regression model. I estimated an OLS model using the given data that describes average water consumtpion and the price of bottled water. This lab helped me to understand the regression model by calculating simple statistics like R-squared and standard errors. I also learned more about residuals in the process.

Data Dictionary

Back to Contents

Variable	Values	Source	Mnemonic
Aggregate Consumption of Bottled Water	million gallons per year	International Bottled Water Association	aggConsumption
Per Capita Consumption of Bottled Water	gallons per year	International Bottled Water Association	perCapitaCons
Food Consumer Price Index	index	International Bottled Water Association	foodCPI
US Population	million	International Bottled Water Association	pop
Price of Bottled Water	dollar per gallon	International Bottled Water Association	price
Price of Bottled Water adjusted for inflation	dollar per gallon	International Bottled Water Association	realPrice
Aggregate Revenue	dollar	International Bottled Water Association	aggRevenue
Real Disposable Income per Capita	dollar	International Bottled Water Association	realDisIncome

Tasks

Back to Contents

Load the Pandas and Statsmodels packages and give them aliases. I recommend 'pd' and 'sm'. You will also need the Statsmodels formula API for formulas. See Lesson #4 for examples.

In [1]:

```
## Analysis
import pandas as pd
import numpy as np
import statsmodels.api as sm
import statsmodels.formula.api as smf
import statsmodels.stats.weightstats as ztest
from statsmodels.iolib.summary2 import summary_col
## Visualization
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
## sns.set()
sns.set(rc={'figure.figsize':(11.7,8.27)})
```

Import the water consumption data. Set the row index to the years.

In [2]:

```
data_path = '/Users/maanyatandon/Documents/fall2020/econometrics/Lab4/'
df    = pd.read_csv( data_path+'071410_1 water consumption.csv')
```

Print the first five (5) records.

In [3]:

```
df.head()
#df['myRealPrice'] = df.price / df.foodCPI
#df
```

Out[3]:

	Year	aggConsumption	perCapitaCons	foodCPI	pop	price	realPrice	aggRevenue	realDisIncome
0	1993	2689.4	10.333711	141.6	260.255	1.069644	0.007554	2876.7	24044
1	1994	2966.4	11.260420	144.9	263.436	1.066714	0.007362	3164.3	24517
2	1995	3226.9	12.101314	148.9	266.657	1.091419	0.007330	3521.9	24951
3	1996	3495.1	12.960800	153.7	269.667	1.097365	0.007140	3835.4	25475
4	1997	3794.3	13.903016	157.7	272.912	1.112906	0.007057	4222.7	26061

Estimate an OLS model using per capita consumption as the dependent variable and real price as the independent variable. Display the summary report. See Lesson #4 for an example.

In [4]:

```
##
## Enter the code here
## Step 1: Formula = 'Y ~ X1 + X2 + X3'
formula = 'perCapitaCons ~ realPrice'
## Step 2: Initialized
mod = smf.ols(formula, data=df)
## Step 3: Fit
reg01 = mod.fit()
## Step 4: Summarize the fitted model
print(reg01.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:          perCapitaCons   R-squared:                       0.376
Model:                            OLS   Adj. R-squared:                  0.332
Method:                 Least Squares   F-statistic:                     8.453
Date:                Sun, 25 Oct 2020   Prob (F-statistic):             0.0115
Time:                        12:25:53   Log-Likelihood:                -47.992
No. Observations:                  16   AIC:                             99.98
Df Residuals:                      14   BIC:                             101.5
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     79.1733     20.774      3.811      0.002      34.618     123.728
realPrice  -8425.8093   2898.079     -2.907      0.011   -1.46e+04   -2210.047
==============================================================================
Omnibus:                       14.458   Durbin-Watson:                   0.176
Prob(Omnibus):                  0.001   Jarque-Bera (JB):                2.157
Skew:                          -0.095   Prob(JB):                        0.340
Kurtosis:                       1.211   Cond. No.                     2.23e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.23e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

```
/opt/anaconda3/lib/python3.8/site-packages/scipy/stats/stats.py:1603: UserWarning: kurtosistest only
valid for n>=20 ... continuing anyway, n=16
  warnings.warn("kurtosistest only valid for n>=20 ... continuing "
```

Retrieve and display the estimated parameters.

In [5]:

```
print('Estimated Params:\n{}'.format(reg01.params))
```

```
Estimated Params:
Intercept      79.173291
realPrice   -8425.809311
dtype: float64
```

Retrieve the residuals and verify that the sum of the residuals is zero.

In [6]:

```
##
## Enter the code here
##
print('Residulas:\n')
print(reg01.resid)
print('Sum of Residulas:{:0.4f}'.format(reg01.resid.sum()))
```

```
Residulas:

0    -5.191172
1    -5.884385
2    -5.311813
3    -6.055125
4    -5.808414
5    -5.132406
6    -3.405570
7     2.291645
8     3.408827
9     5.920903
10    5.875743
11    4.861320
12    4.793943
13    5.091817
14    4.520746
15    0.023940
dtype: float64
Sum of Residulas:-0.0000
```

Calculate the standard error of the regression.

In [7]:

```
sse = reg01.ssr
print('Sum of Squared Errors:   {:0.4f}'.format(sse))
se_reg = np.sqrt( sse / (reg01.nobs-2))
print('Std Error of Regression: {:0.4f}'.format(se_reg))
```

```
Sum of Squared Errors:   377.5631
Std Error of Regression: 5.1931
```

Well done!

Make sure your name is on this notebook at the top and on the file.
Please submit this notebook as a PDF file. Nothing else will be accepted.

In [ ]: