

```
In [ ]: from __future__ import print_function
import pandas as pd
from statsmodels.formula.api import ols
```

```
In [2]: df = pd.read_csv('supermarket.csv')
```

```
In [3]: df.head()
```

Out[3]:

	EggSales	EggPrice	Ads	CookiePrice
0	18	4.77	0	4.74
1	20	4.67	0	4.81
2	23	4.75	0	4.36
3	23	4.74	0	4.29
4	23	4.63	0	4.17

```
In [4]: adYes = df.loc[df['Ads'] == 1]
adYes = adYes.drop(columns = ['Ads'])
adYes.head()
```

Out[4]:

	EggSales	EggPrice	CookiePrice
8	26	4.75	4.21
10	26	4.41	4.62
13	28	4.66	4.19
14	28	4.42	4.37
16	29	4.66	4.57

```
In [5]: price_model_adYes = ols("EggSales~EggPrice+CookiePrice", data=adYes)
price_model_adYes.fit()
print(price_model_adYes.summary())
```

# OLS Regression Results

```

=====
=====
Dep. Variable:          EggSales    R-squared:
0.885
Model:                  OLS        Adj. R-squared:
0.866
Method:                 Least Squares    F-statistic:
46.29
Date:                   Sat, 18 Apr 2020    Prob (F-statistic):
2.28e-06
Time:                   20:55:47    Log-Likelihood:
-33.612
No. Observations:      15    AIC:
73.22
Df Residuals:          12    BIC:
75.35
Df Model:               2
Covariance Type:       nonrobust
=====
=====

```

	coef	std err	t	P> t	[0.02
5	0.975]				
Intercept	161.8078	15.398	10.509	0.000	128.25
9	195.356				
EggPrice	-18.9614	2.061	-9.199	0.000	-23.45
2	-14.470				
CookiePrice	-10.5593	2.623	-4.026	0.002	-16.27
4	-4.845				

```

=====
=====
Omnibus:                0.147    Durbin-Watson:
2.292
Prob(Omnibus):          0.929    Jarque-Bera (JB):
0.361
Skew:                   0.047    Prob(JB):
0.835
Kurtosis:               2.246    Cond. No.
148.
=====
=====

```

## Warnings:

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

```

/Users/Work/PycharmProjects/AdultIncome_hfY/venv/lib/python3.7/site-packages/scipy/stats/stats.py:1535: UserWarning: kurtosistest only valid for n>=20 ... continuing anyway, n=15
  "anyway, n=%i" % int(n))

```

```
In [6]: adNo = df.loc[df['Ads'] == 0]
adNo = adNo.drop(columns = ['Ads'])
adNo.head()
```

Out[6]:

	<b>EggSales</b>	<b>EggPrice</b>	<b>CookiePrice</b>
<b>0</b>	18	4.77	4.74
<b>1</b>	20	4.67	4.81
<b>2</b>	23	4.75	4.36
<b>3</b>	23	4.74	4.29
<b>4</b>	23	4.63	4.17

```
In [7]: price_model_adNo = ols("EggSales~EggPrice+CookiePrice", data=adNo).
fit()
print(price_model_adNo.summary())
```

# OLS Regression Results

```

=====
=====
Dep. Variable:          EggSales    R-squared:
0.802
Model:                  OLS        Adj. R-squared:
0.769
Method:                 Least Squares    F-statistic:
24.26
Date:                   Sat, 18 Apr 2020    Prob (F-statistic):
6.08e-05
Time:                   20:55:47    Log-Likelihood:
-32.652
No. Observations:      15    AIC:
71.30
Df Residuals:          12    BIC:
73.43
Df Model:               2
Covariance Type:       nonrobust
=====
=====

```

	coef	std err	t	P> t	[0.02
5	0.975]				
Intercept	118.3334	13.373	8.849	0.000	89.19
6	147.471				
EggPrice	-11.9195	2.553	-4.669	0.001	-17.48
2	-6.357				
CookiePrice	-8.7290	2.610	-3.345	0.006	-14.41
5	-3.043				

```

=====
=====
Omnibus:                0.967    Durbin-Watson:
1.183
Prob(Omnibus):          0.616    Jarque-Bera (JB):
0.787
Skew:                   0.493    Prob(JB):
0.675
Kurtosis:               2.463    Cond. No.
141.
=====
=====

```

## Warnings:

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

```

/Users/Work/PycharmProjects/AdultIncome_hfY/venv/lib/python3.7/site-packages/scipy/stats/stats.py:1535: UserWarning: kurtosistest only valid for n>=20 ... continuing anyway, n=15
  "anyway, n=%i" % int(n))

```

```
In [8]: coef_adYes = -18.9614
Elasticity_adYes = coef_adYes * adYes.mean(axis = 'rows')[1] / adY
es.mean(axis = 'rows')[0]
print(adYes.mean(axis = 'rows'))
print("Price elasticity of eggs with ads is" + "Elasticity_adYes")
```

```
EggSales      34.200000
EggPrice      4.300000
CookiePrice    4.363333
dtype: float64
Price elasticity of eggs with ads isElasticity_adYes
```

```
In [9]: coef_adNo = -11.9195
Elasticity_adNo = coef_adNo * adNo.mean(axis = 'rows')[1] / adNo.m
ean(axis = 'rows')[0]
print(adNo.mean(axis = 'rows'))
print("Price elasticity of eggs without ads is" + "Elasticity_adNo"
)
```

```
EggSales      25.800000
EggPrice      4.552667
CookiePrice    4.384000
dtype: float64
Price elasticity of eggs without ads isElasticity_adNo
```

```
In [10]: change_in_elasticity_with_ads = Elasticity_adYes - Elasticity_adNo

print("Price elasticity of eggs:")
print("WITHOUT ads:",Elasticity_adNo,",")
print("WITH ads   :",Elasticity_adYes,":")
if change_in_elasticity_with_ads < 0 :
    print("Ads have a desirable effect")
elif change_in_elasticity_with_ads == 0 :
    print("Ads have no effect")
else:
    print("Ads have an undesirable effect")
```

```
Price elasticity of eggs:
WITHOUT ads: -2.103314354005168 ,
WITH ads   : -2.38403567251462 :
Ads have a desirable effect
```