IIT ROORKEE

NPTEL ONLINE CERTIFICATION COURSE

Regression Analysis Model Building (Interaction)- II

Dr. A. Ramesh

DEPARTMENT OF MANAGEMENT STUDIES

Agenda

- Incorporating Interaction of the independent variable to the regression model
- Python demo

Interaction

- If the original data set consists of observations for *y* and two independent variables *x*1 and *x*2, we can develop a second-order model with two predictor variables by setting $z_1 = x_1$, $z_2 = x_2$, $z_3 = x_1^2$, $z_4 = x_2^2$, and $z_5 = x_1 x_2$ in the general linear model of equation
- The model obtained is

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2 + \epsilon$$

- In this second-order model, the variable $z_5 = x_1 x_2$ is added to account for the potential effects of the two variables acting together.
- This type of effect is called **interaction**.

Example – Interaction

- A company introduces a new shampoo product.
- Two factors believed to have the most influence on sales are unit selling price and advertising expenditure.
- To investigate the effects of these two variables on sales, prices of \$2.00, \$2.50, and \$3.00 were paired with advertising expenditures of \$50,000 and \$100,000 in 24 test markets.

Source: Statistics for Business and Economics,11th Edition by David R. Anderson (Author), Dennis J. Sweeney (Author), Thomas A. Williams (Author)

Price	Advertising Expenditure ($1000s)	Sales (1000s)
2	50	478
2.5	50	373
3	50	335
2	50	473
2.5	50	358
3	50	329
2	50	456
2.5	50	360
3	50	322
2	50	437
2.5	50	365
3	50	342
2	100	810
2.5	100	653
3	100	345
2	100	832
2.5	100	641
3	100	372
2	100	800
2.5	100	620
3	100	390
2	100	790
2.5	100	670
3	100	393

MEAN UNIT SALES (1000s)

		Price		
		$2.00	$2.50	$3.00
Advertising Expenditure	$50,000	461	364	332
	$100,000	808	646	375

Mean sales of 808,000 units when price = $2.00 and advertising expenditure = $100,000

Interpretation of interaction

- Note that the sample mean sales corresponding to a price of $2.00 and an advertising expenditure of $50,000 is 461,000, and the sample mean sales corresponding to a price of $2.00 and an advertising expenditure of $100,000 is 808,000.
- Hence, with price held constant at $2.00, the difference in mean sales between advertising expenditures of $50,000 and $100,000 is 808,000 - 461,000 = 347,000 units.

Interpretation of interaction

- When the price of the product is \$2.50, the difference in mean sales is 646,000 -364,000 = 282,000 units.
- Finally, when the price is \$3.00, the difference in mean sales is 375,000 - 332,000 = 43,000 units.
- Clearly, the difference in mean sales between advertising expenditures of \$50,000 and \$100,000 depends on the price of the product.
- In other words, at higher selling prices, the effect of increased advertising expenditure diminishes.
- These observations provide evidence of interaction between the price and advertising expenditure variables.

Importing Data

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

In [8]:
```
tbl1 = pd.read_excel('Tyler.xlsx')
tbl1.head()
```

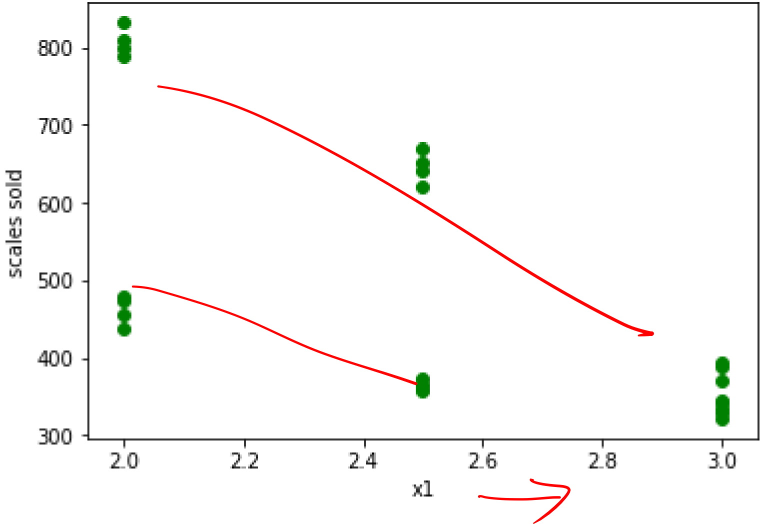
Out[8]:

	Price	AdvertisingExpenditure($1000s)	Sales(1000s)
0	2.0	50	478
1	2.5	50	373
2	3.0	50	335
3	2.0	50	473
4	2.5	50	358

Mean unit sales (1000s) as a function of selling price

In [7]:
```
plt.scatter(tbl1['Price'],tbl1['Sales(1000s)'], color='green')
plt.ylabel('scales sold')
plt.xlabel('x1')
```

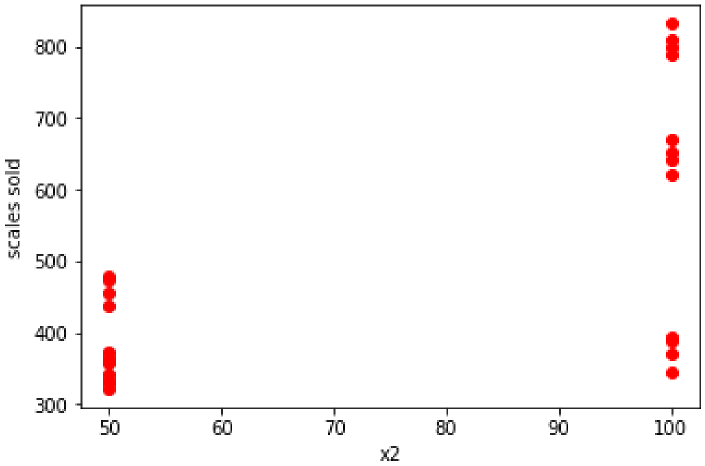
Out[7]:
```
Text(0.5,0,'x1')
```



Mean unit sales (1000s) as a function of **Advertising Expenditure($1000s)**

In [6]:
```
plt.scatter(tbl1['AdvertisingExpenditure($1000s)'],tbl1['Sales(1000s)'], color='red')
plt.ylabel('scales sold')
plt.xlabel('x2')
```

Out[6]:
```
Text(0.5,0,'x2')
```



Need for study the interaction between variable

- When interaction between two variables is present, we cannot study the effect of one variable on the response *y* independently of the other variable.
- In other words, meaningful conclusions can be developed only if we consider the joint effect that both variables have on the response.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \epsilon$$

$$\begin{aligned} y &= \text{unit sales (1000s)} \\ x_1 &= \text{price (\$)} \\ x_2 &= \text{advertising expenditure (\$1000s)} \end{aligned}$$

Estimated regression equation, a general linear model involving three independent variables (z_1, z_2, and z_3)

$$y = \beta_0 + \beta_1 z_1 + \beta_2 z_2 + \beta_3 z_3 + \epsilon$$

$$z_1 = x_1$$

$$z_2 = x_2$$

$$\boxed{z_3 = x_1 x_2}$$

Interaction variable

- *The data for the* PriceAdv *independent variable is obtained by multiplying each value of* Price *times the corresponding value of* AdvExp.

In [11]:
```
z1 =tbl1['AdvertisingExpenditure($1000s)']
z2 = tbl1['Price']
z3 = z1*z2
```

New Model

In [12]:
```
x_new =np.column_stack((z1,z2,z3))
y = tbl1['Sales(1000s)']
xnew2 = sm.add_constant(x_new)
model2 = sm.OLS(y,xnew2)
Model2 = model2.fit()
print(Model2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:           Sales(1000s)   R-squared:                       0.978
Model:                            OLS   Adj. R-squared:                  0.975
Method:                 Least Squares   F-statistic:                     297.9
Date:                Thu, 12 Sep 2019   Prob (F-statistic):           9.26e-17
Time:                        13:12:52   Log-Likelihood:                -111.99
No. Observations:                  24   AIC:                             232.0
Df Residuals:                      20   BIC:                             236.7
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       -275.8333    112.842     -2.444      0.024    -511.218     -40.449
x1            19.6800      1.427     13.788      0.000      16.703      22.657
x2           175.0000     44.547      3.928      0.001      82.077     267.923
x3            -6.0800      0.563    -10.790      0.000      -7.255      -4.905
==============================================================================
Omnibus:                        0.641   Durbin-Watson:                   2.842
Prob(Omnibus):                  0.726   Jarque-Bera (JB):                0.565
Skew:                           0.335   Prob(JB):                        0.754
Kurtosis:                       2.661   Cond. No.                     4.53e+03
==============================================================================
```

New Model

$$\text{Sales} = -276 + 175\,\text{Price} + 19.7\,\text{AdvExp} - 6.08\,\text{PriceAdv}$$

where

$$\begin{aligned}
\text{Sales} &= \text{unit sales (1000s)} \\
\text{Price} &= \text{price of the product (\$)} \\
\text{AdvExp} &= \text{advertising expenditure (\$1000s)} \\
\text{PriceAdv} &= \text{interaction term (Price times AdvExp)}
\end{aligned}$$

Interpretation

- Because the model is significant (*p*-value for the *F* test is 0.000) and the *p*-value corresponding to the *t* test for PriceAdv is 0.000, we conclude that interaction is significant given the linear effect of the price of the product and the advertising expenditure.
- Thus, the regression results show that the effect of advertising xpenditure on sales depends on the price.

Transformations Involving the Dependent Variable

$\boxed{y} = b_0 + b_1 x_1 + b_2 x_2$

$x_2 = 0, 1$

Miles per Gallon (y)	Weight (x)
28.7	2289
29.2	2113
34.2	2180
27.9	2448
33.3	2026
26.4	2702
23.9	2657
30.5	2106
18.1	3226
19.5	3213
14.3	3607
20.9	2888

Importing data

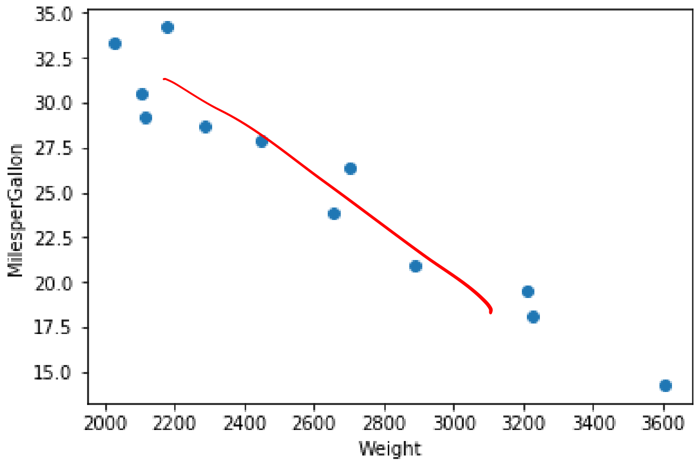
```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import statsmodels.api as sm

In [2]: tbl1 = pd.read_excel('MPG.xlsx')
        tbl1

Out[2]:
```

	MilesperGallon	Weight
0	28.7	2289
1	29.2	2113
2	34.2	2180
3	27.9	2448
4	33.3	2026
5	26.4	2702
6	23.9	2657
7	30.5	2106
8	18.1	3226
9	19.5	3213
10	14.3	3607
11	20.9	2888

Scatter diagram

In [3]:
```
plt.scatter(tbl1['Weight'],tbl1['MilespergGallon'])
plt.ylabel('MilesperGallon')
plt.xlabel('Weight')
```

Out[3]: `Text(0.5,0,'Weight')`



Model 1

```
In [4]: x =tbl1['Weight']
        y = tbl1['MilesperGallon']
        x2 = sm.add_constant(x)
        model = sm.OLS(y,x2)
        Model = model.fit()
        print(Model.summary())
```

```
C:\Users\Somi\Anaconda3\lib\site-packages\scipy\stats\stats.py:1394: UserWarning: kurtosistest only va
ing anyway, n=12
  "anyway, n=%i" % int(n))
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:          MilesperGallon   R-squared:                       0.935
Model:                             OLS   Adj. R-squared:                  0.929
Method:                  Least Squares   F-statistic:                     144.8
Date:                 Thu, 12 Sep 2019   Prob (F-statistic):           2.85e-07
Time:                         15:27:08   Log-Likelihood:                -22.091
No. Observations:                   12   AIC:                             48.18
Df Residuals:                       10   BIC:                             49.15
Df Model:                            1
Covariance Type:             nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         56.0957      2.582     21.725      0.000      50.342      61.849
Weight        -0.0116      0.001    -12.032      0.000      -0.014      -0.009
==============================================================================
Omnibus:                         2.266   Durbin-Watson:                   2.213
Prob(Omnibus):                   0.322   Jarque-Bera (JB):                0.951
Skew:                            0.690   Prob(JB):                        0.621
Kurtosis:                        3.025   Cond. No.                     1.43e+04
```

Standardized residual plot corresponding to the first-order model.

In [6]:
```
E=Model.resid_pearson
E
```

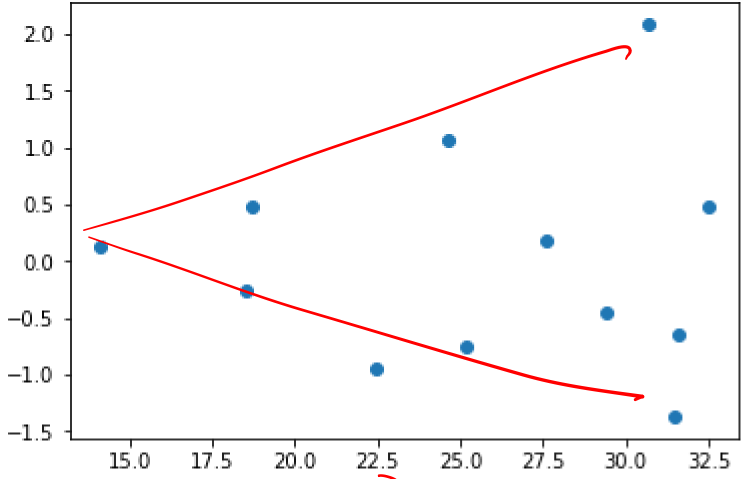
Out[6]:
```
array([-0.44511273, -1.37252481,  2.08753315,  0.18422536,  0.47540179,
        1.05668329, -0.75350063, -0.64311699, -0.25953343,  0.4879158 ,
        0.12130227, -0.93927307])
```

In [7]:
```
yhat = Model.predict(x2)
yhat
```

Out[7]:
```
0     29.443573
1     31.492839
2     30.712721
3     27.592247
4     32.505829
5     24.634783
6     25.158743
7     31.574344
8     18.533557
9     18.684924
10    14.097361
11    22.469081
dtype: float64
```

Standardized residual plot corresponding to the first-order model

```
In [8]: plt.scatter(yhat,E)

Out[8]: <matplotlib.collections.PathCollection at 0x23f77072a58>
```



$\rightarrow x$

$\log(y) = b_0 + b_1 x_1$

Model 2

```
In [12]: Y = np.log(y)

In [13]: model2 = sm.OLS(Y,x2)
         Model2 = model2.fit()
         print(Model2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:          MilesperGallon   R-squared:                       0.948
Model:                             OLS   Adj. R-squared:                  0.942
Method:                  Least Squares   F-statistic:                     181.2
Date:                 Thu, 12 Sep 2019   Prob (F-statistic):           9.84e-08
Time:                         15:34:13   Log-Likelihood:                 17.005
No. Observations:                   12   AIC:                            -30.01
Df Residuals:                       10   BIC:                            -29.04
Df Model:                            1
Covariance Type:             nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          4.5242      0.099     45.553      0.000       4.303       4.746
Weight        -0.0005   3.72e-05    -13.462      0.000      -0.001      -0.000
==============================================================================
Omnibus:                         0.899   Durbin-Watson:                   2.284
Prob(Omnibus):                   0.638   Jarque-Bera (JB):                0.779
Skew:                            0.484   Prob(JB):                        0.677
Kurtosis:                        2.211   Cond. No.                     1.43e+04
==============================================================================
```

Residual plot for model 2

In [14]:
```
E2=Model2.resid_pearson
E2
```

Out[14]:
```
array([-0.31630114, -1.42005514,  1.5623004 ,  0.48370101, -0.0537228 ,
        1.60448776, -0.29474869, -0.79674991, -0.18335787,  0.87474775,
       -0.87956572, -0.58073564])
```

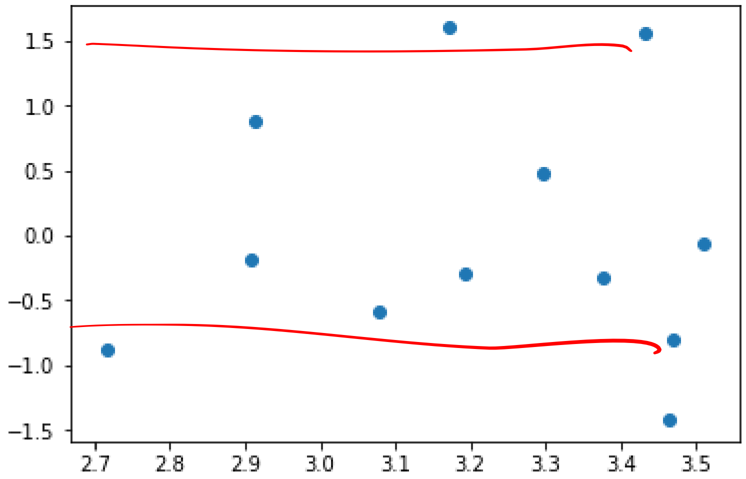
In [15]:
```
yhat = Model2.predict(x2)
yhat
```

Out[15]:
```
0     3.377221
1     3.465414
2     3.431840
3     3.297547
4     3.509009
5     3.170268
6     3.192817
7     3.468922
8     2.907694
9     2.914208
10    2.716776
11    3.077064
dtype: float64
```

Residual plot of model 2

```
In [16]: plt.scatter(yhat,E2)
```

```
Out[16]: <matplotlib.collections.PathCollection at 0x23f7737be10>
```



- The miles-per-gallon estimate is obtained by finding the number whose natural logarithm is 3.2675.
- Using a calculator with an exponential function, or raising *e* to the power 3.2675, we obtain 26.2 miles per gallon.

$$\text{LogeMPG} = 4.52 - 0.000501\ \text{Weight}$$

$$\text{LogeMPG} = 4.52 - 0.000501\,(2500) = 3.2675$$

e

Nonlinear Models That Are Intrinsically Linear

$$E(y) = \beta_0 \beta_1^x$$

$$E(y) = 500(1.2)^x$$

$$\log E(y) = \log \beta_0 + x \log \beta_1$$

$$y' = \log E(y), \beta_0' = \log \beta_0, \text{and } \beta_1' = \log \beta_1,$$

$$y' = \beta_0' + \beta_1' x \qquad \hat{y}' = b_0' + b_1' x$$

Thank You