IIT ROORKEE

NPTEL ONLINE CERTIFICATION COURSE

Categorical Variable Regression

Dr. A. Ramesh

DEPARTMENT OF MANAGEMENT STUDIES

Agenda

- Purpose of this lecture is to show how categorical variables are handled in regression analysis.
- To illustrate the use and interpretation of a categorical independent variable, we will consider two problems
- Demo on python

What are dummy variables?

- Dummy variables, also called indicator variables allow us to include categorical data (like Gender) in regression models
- A dummy variable can take only 2 values, 0 (absence of a category) and 1 (presence of a category)

Example 1: Problem / Background

- Johnson Filtration, Inc., provides maintenance service for water-filtration systems.
- Customers contact Johnson with requests for maintenance service on their water-filtration systems
- To estimate the service time and the service cost, Johnson's managers want to predict the repair time necessary for each maintenance request
- Hence, repair time in hours is the dependent variable
- Repair time is believed to be related to two factors,
  - the number of months since the last maintenance service
  - the type of repair problem (mechanical or electrical).

Source: Statistics for Business & Economics, David R. Anderson, Dennis J. Sweeney, Thomas A. Williams, Jeffrey D. Camm, James J. Cochran, Cengage Learning,2013

Data for the Johnson filtration example

service call	months_since_last_service	type_of_repair	repair_time_in_hours
1	2	electrical	2.9
2	6	mechanical	3
3	8	electrical	4.8
4	3	mechanical	1.8
5	2	electrical	2.9
6	7	electrical	4.9
7	9	mechanical	4.2
8	8	mechanical	4.8
9	4	electrical	4.4
10	6	electrical	4.5

In [23]:
```
import pandas as pd
import matplotlib as mpl
import statsmodels.formula.api as sm
from sklearn.linear_model import LinearRegression
from scipy import stats
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as s
```

In [24]:
```
tbl = pd.read_excel('dummy.xlsx')
tbl
```

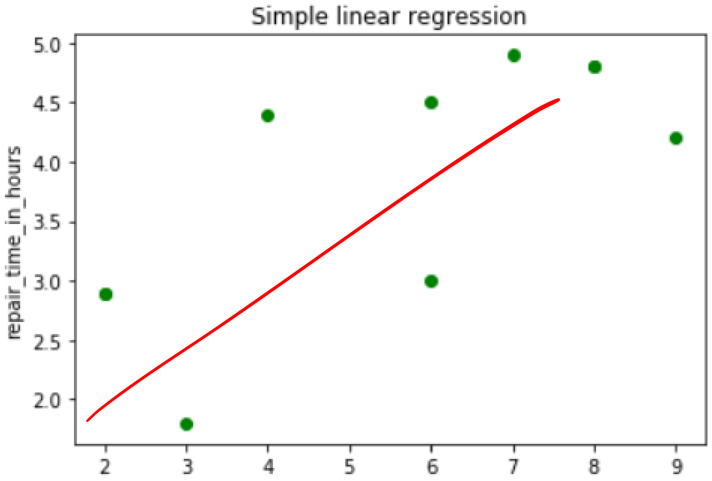
Out[24]:

	servicecall	months_since_last_service	type_of_repair	repair_time_in_hours
0	1	2	electrical	2.9
1	2	6	mechanical	3.0
2	3	8	electrical	4.8
3	4	3	mechanical	1.8
4	5	2	electrical	2.9
5	6	7	electrical	4.9
6	7	9	mechanical	4.2
7	8	8	mechanical	4.8
8	9	4	electrical	4.4
9	10	6	electrical	4.5

Linear Regression

```
In [41]: plt.scatter(tbl['months_since_last_service'], tbl['repair_time_in_hours'], color = "green")
         plt.ylabel('repair_time_in_hours')
         plt.title(' Simple linear regression ')

Out[41]: Text(0.5,1,' Simple linear regression ')
```



OLS Summary

In [44]:
```
from statsmodels.formula.api import ols
Reg = ols(formula ="repair_time_in_hours ~ months_since_last_service", data = tbl)
Fit1 = Reg.fit()
print(Fit1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     repair_time_in_hours   R-squared:                       0.534
Model:                              OLS   Adj. R-squared:                  0.476
Method:                   Least Squares   F-statistic:                     9.174
Date:                  Sat, 07 Sep 2019   Prob (F-statistic):             0.0163
Time:                          13:26:03   Log-Likelihood:                -10.602
No. Observations:                    10   AIC:                             25.20
Df Residuals:                         8   BIC:                             25.81
Df Model:                             1
Covariance Type:              nonrobust
=============================================================================================
                                coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------------
Intercept                     2.1473      0.605      3.549      0.008       0.752       3.542
months_since_last_service     0.3041      0.100      3.029      0.016       0.073       0.536
==============================================================================
Omnibus:                        0.907   Durbin-Watson:                   2.154
Prob(Omnibus):                  0.635   Jarque-Bera (JB):                0.751
Skew:                          -0.501   Prob(JB):                        0.687
Kurtosis:                       2.107   Cond. No.                         15.1
==============================================================================
```

y = 2.1473 +
0.3041
mnths

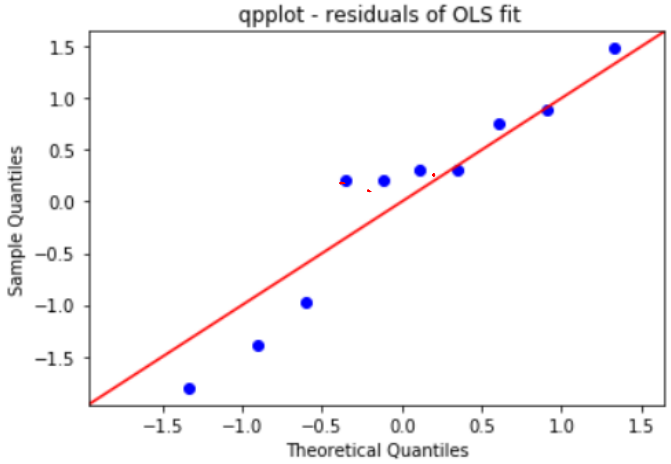
Linear regression

$$y = \beta_0 + \beta_1 x_1 + \epsilon$$

$$\hat{y} = 2.15 + .304 x_1$$

Normal probability plot

```
In [49]: res = Fit1.resid # residuals
         probplot = s.ProbPlot(res,stats.norm, fit=True)
         fig = probplot.qqplot(line='45')
         h = plt.title(' qpplot - residuals of OLS fit')
         plt.show()
```



Creating dummies

In [34]:
```
just_dummies = pd.get_dummies(tbl['type_of_repair'])
just_dummies
```

Out[34]:

	electrical	mechanical
0	1	0
1	0	1
2	1	0
3	0	1
4	1	0
5	1	0
6	0	1
7	0	1
8	1	0
9	1	0

$$y = a + b_1 x_1 + b_2 x_2$$

$$y = a + b_1 x_1 + b_2 (1)$$

$$y = a + b_1 x_1 + b_2 (0)$$

DATA FOR THE JOHNSON FILTRATION EXAMPLE WITH TYPE OF REPAIR INDICATED BYADUMMYVARIABLE ($x2$ = 0 FOR MECHANICAL; $x2$ = 1 FOR ELECTRICAL)

Customer	Months Since Last Service (x_1)	Type of Repair (x_2)	Repair Time in Hours (y)
1	2	1	2.9
2	6	0	3.0
3	8	1	4.8
4	3	0	1.8
5	2	1	2.9
6	7	1	4.9
7	9	0	4.2
8	8	0	4.8
9	4	1	4.4
10	6	1	4.5

Adding dummies to table

In [38]:
```
just_dummies = pd.get_dummies(tbl['type_of_repair'])
step_1 = pd.concat([tbl, just_dummies], axis=1)
step_1
step_1.drop(['type_of_repair', 'mechanical'], inplace=True, axis=1)

# to run the regression we want to get rid of the strings 'mechanical' and 'electrical'
# and we want to get rid of one dummy variable to avoid the dummy variable trap
# arbitrarily chose "mechanical", coefficients on "electrical" would show effect of "electrical"
# relative to "mechanical"
```

In [39]:
```
step_1
```

Out[39]:

	servicecall	months_since_last_service	repair_time_in_hours	electrical
0	1	2	2.9	1
1	2	6	3.0	0
2	3	8	4.8	1
3	4	3	1.8	0
4	5	2	2.9	1
5	6	7	4.9	1
6	7	9	4.2	0
7	8	8	4.8	0
8	9	4	4.4	1
9	10	6	4.5	1

OLS Summary

In [20]:
```
result = sm.OLS(step_1['repair_time_in_hours'], s.add_constant(step_1[['months_since_last_service', 'electrical']])).fit()
print (result.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     repair_time_in_hours   R-squared:                       0.859
Model:                              OLS   Adj. R-squared:                  0.819
Method:                   Least Squares   F-statistic:                     21.36
Date:                  Sat, 07 Sep 2019   Prob (F-statistic):            0.00105
Time:                          13:08:09   Log-Likelihood:                -4.6200
No. Observations:                    10   AIC:                             15.24
Df Residuals:                         7   BIC:                             16.15
Df Model:                             2
Covariance Type:              nonrobust
=============================================================================================
                                coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------------
const                         0.9305      0.467      1.993      0.087      -0.174       2.035
months_since_last_service     0.3876      0.063      6.195      0.000       0.240       0.536
electrical                    1.2627      0.314      4.020      0.005       0.520       2.005
==============================================================================
Omnibus:                        3.357   Durbin-Watson:                   1.136
Prob(Omnibus):                  0.187   Jarque-Bera (JB):                1.663
Skew:                           0.994   Prob(JB):                        0.435
Kurtosis:                       2.795   Cond. No.                         22.0
==============================================================================
```

y = 0.9305 + 0.3876 months_since_last_service + electrical 1.2627

Dummy regression

$$x_2 = \begin{cases} 0 \text{ if the type of repair is mechanical} \\ 1 \text{ if the type of repair is electrical} \end{cases}$$

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

$$\hat{y} = .93 + .388x_1 + 1.26x_2$$

$x_2 = 1$ - Electrical

$x_2 = 0$ - mechanical

Interpreting the Parameters

$$E(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

$$E(y \mid \text{mechanical}) = \beta_0 + \beta_1 x_1 + \beta_2(0) = \beta_0 + \beta_1 x_1$$

Equation 1

$$E(y \mid \text{electrical}) = \beta_0 + \beta_1 x_1 + \beta_2(1) = \beta_0 + \beta_1 x_1 + \beta_2$$
$$= (\beta_0 + \beta_2) + \beta_1 x_1$$

Equation 2

Interpreting the Parameters

- Comparing equations, we see that the mean repair time is a linear function of $x1$ for both mechanical and electrical repairs.
- The slope of both equations is $\beta 1$, but the y-intercept differs.
- The y-intercept is β_0 in equation 1 for mechanical repairs and $(\beta_0 + \beta_2)$ in equation 2 for electrical repairs.

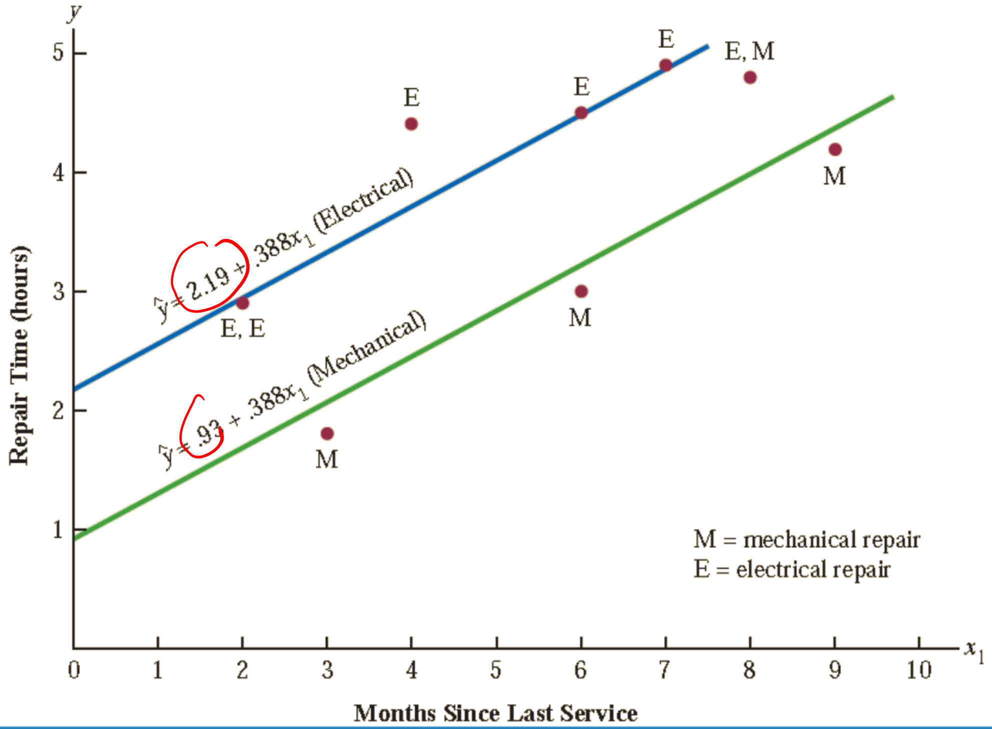
Interpreting the Parameters

- The interpretation of $\beta 2$ is that it indicates the difference between the mean repair time for an electrical repair and the mean repair time for a mechanical repair.
- If $\beta 2$ is positive, the mean repair time for an electrical repair will be greater than that for a mechanical repair; if $\beta 2$ is negative, the mean repair time for an electrical repair will be less than that for a mechanical repair.
- Finally, if $\beta 2 = 0$, there is no difference in the mean repair time between electrical and mechanical repairs and the type of repair is not related to the repair time.

Interpreting the Parameters

- In effect, the use of a dummy variable for type of repair provides two estimated regression equations that can be used to predict the repair time, one corresponding to mechanical repairs and one corresponding to electrical repairs.
- In addition, with $\beta 2 = 1.26$, we learn that, on average, electrical repairs require 1.26 hours longer than mechanical repairs.

Interpreting the Parameters



More Complex Categorical Variables

- A categorical variable with k levels must be modeled using k - 1 dummy variables.
- Care must be taken in defining and interpreting the dummy variables.

Example 2: Problem / Background

- The manager of a small sales force wants to know whether average monthly salary is different for males and females in the sales force.
- He obtains data on monthly salary and experience (in months) for each of the 9 employees as shown on the next slide.

Data

Employee	Salary	Gender	Experience
1	7.5	Male	6
2	8.6	Male	10
3	9.1	Male	12
4	10.3	Male	18
5	13	Male	30
6	6.2	Female	5
7	8.7	Female	13
8	9.4	Female	15
9	9.8	Female	21

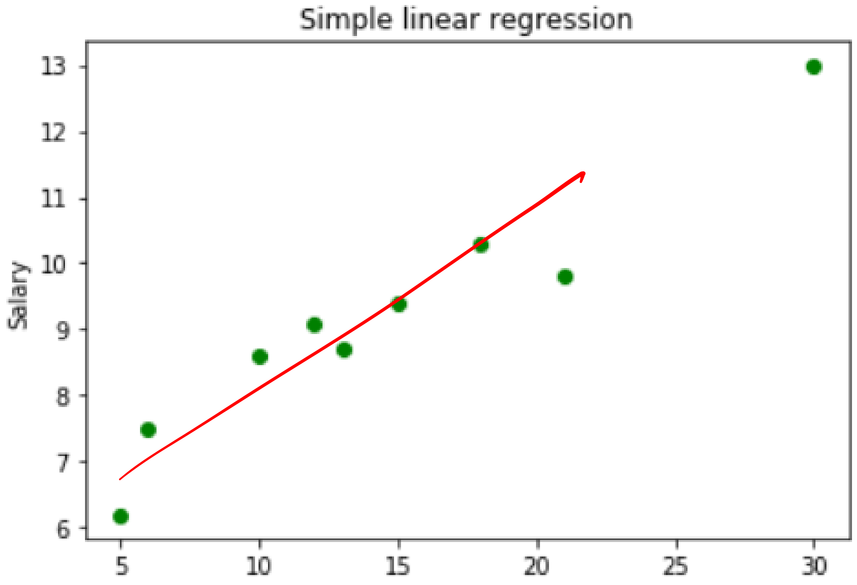
In [50]:
```
tbl2 = pd.read_excel('dummy2.xlsx')
tbl2
```

Out[50]:

	Employee	Salary	Gender	Experience
0	1	7.5	Male	6
1	2	8.6	Male	10
2	3	9.1	Male	12
3	4	10.3	Male	18
4	5	13.0	Male	30
5	6	6.2	Female	5
6	7	8.7	Female	13
7	8	9.4	Female	15
8	9	9.8	Female	21

In [51]:
```
plt.scatter(tbl2['Experience'], tbl2['Salary'], color = "green")
plt.ylabel('Salary')
plt.title(' Simple linear regression ')
```

Out[51]:
```
Text(0.5,1,' Simple linear regression ')
```



```
In [59]: Reg2 = ols(formula ="Salary ~ Experience", data = tbl2)
         Fit2 = Reg2.fit()
         print(Fit2.summary())
```

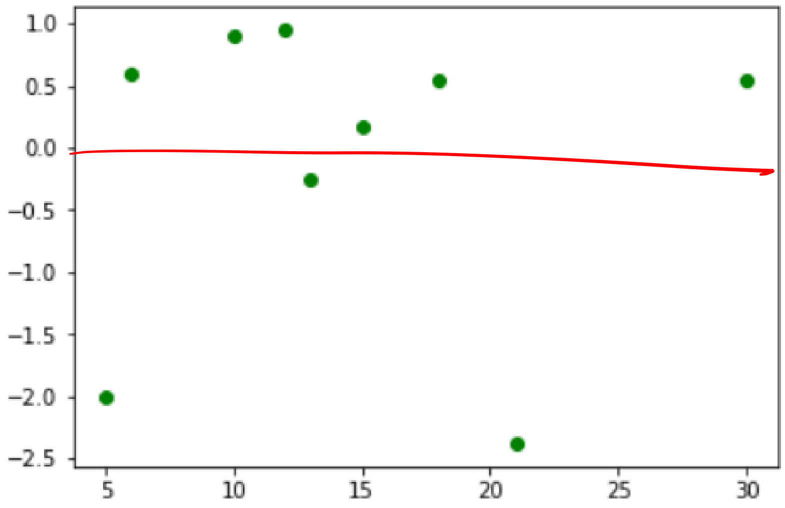
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Salary   R-squared:                       0.926
Model:                            OLS   Adj. R-squared:                  0.915
Method:                 Least Squares   F-statistic:                     87.61
Date:                Sat, 07 Sep 2019   Prob (F-statistic):           3.30e-05
Time:                        14:18:45   Log-Likelihood:                -6.2491
No. Observations:                   9   AIC:                             16.50
Df Residuals:                       7   BIC:                             16.89
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      5.8093      0.404     14.386      0.000       4.854       6.764
Experience     0.2332      0.025      9.360      0.000       0.174       0.292
==============================================================================
Omnibus:                        2.443   Durbin-Watson:                   1.171
Prob(Omnibus):                  0.295   Jarque-Bera (JB):                1.432
Skew:                          -0.918   Prob(JB):                        0.489
Kurtosis:                       2.331   Cond. No.                         35.8
==============================================================================
```

$y = 5.8 + 0.2332\ Exp.$

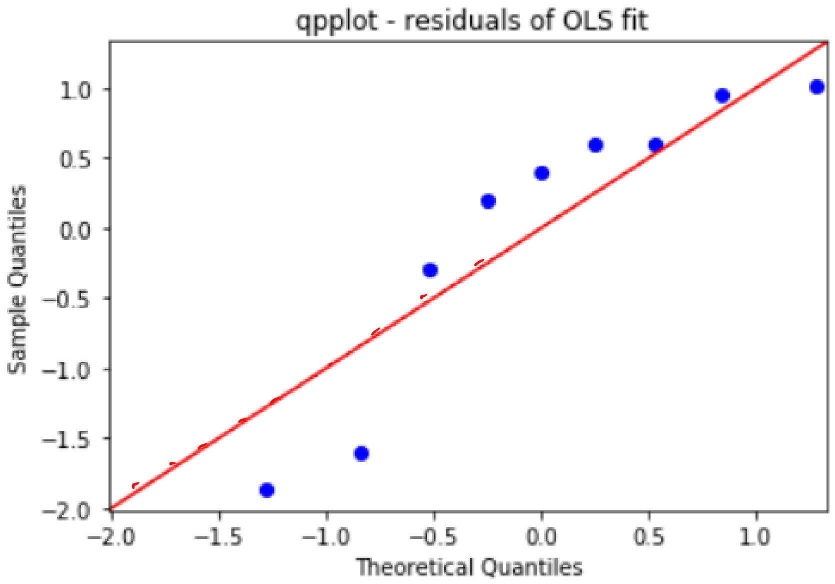
```
In [55]: influence = Fit2.get_influence()
         resid_student = influence.resid_studentized_external
```

```
In [57]: plt.figure()
         plt.scatter(tbl2['Experience'],resid_student, color = "green")
```

```
Out[57]: <matplotlib.collections.PathCollection at 0x2d3e12019b0>
```



In [58]:
```
res =Fit2.resid # residuals
probplot = s.ProbPlot(res,stats.norm, fit=True)
fig = probplot.qqplot(line='45')
h = plt.title(' qpplot - residuals of OLS fit')
plt.show()
```



Creating a dummy variable for gender

- Categorical data is included in regression analysis by using dummy variables
- For example, we can assign a value of 0 for males and 1 for females in our data so that a MR model can be developed

$x_2 = 0$ males

$x_2 = 1$ females

Employee	Salary	Gender
1	7.5	0
2	8.6	0
3	9.1	0
4	10.3	0
5	13	0
6	6.2	1
7	8.7	1
8	9.4	1
9	9.8	1

In [24]:
```
just_dummies2 = pd.get_dummies(tbl2['Gender'])
just_dummies2
```

Out[24]:

	Female	Male
0	0	1
1	0	1
2	0	1
3	0	1
4	0	1
5	1	0
6	1	0
7	1	0
8	1	0

```
In [62]: step_1 = pd.concat([tbl2, just_dummies2], axis=1)
step_1.drop(['Gender', 'Male'], inplace=True, axis=1)
# to run the regression we want to get rid of the strings 'male' and 'female'
# and we want to get rid of one dummy variable to avoid the dummy variable trap
# arbitrarily chose "male", coefficients on "female" would show effect of "female"
# relative to "male"

result = sm.OLS(step_1['Salary'], s.add_constant(step_1[['Female']])).fit()
print (result.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Salary   R-squared:                       0.107
Model:                            OLS   Adj. R-squared:                 -0.020
Method:                 Least Squares   F-statistic:                    0.8426
Date:                Sat, 07 Sep 2019   Prob (F-statistic):              0.389
Time:                        14:23:57   Log-Likelihood:                -17.455
No. Observations:                   9   AIC:                             38.91
Df Residuals:                       7   BIC:                             39.30
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          9.7000      0.853     11.367      0.000       7.682      11.718
Female        -1.1750      1.280     -0.918      0.389      -4.202       1.852
==============================================================================
Omnibus:                        0.387   Durbin-Watson:                   1.912
Prob(Omnibus):                  0.824   Jarque-Bera (JB):                0.280
Skew:                           0.330   Prob(JB):                        0.869
Kurtosis:                       2.441   Cond. No.                         2.51
==============================================================================
```

$y = 9.7 - 1.1750\, x_1$

More on the intercept and slope

- The value of the intercept, 9.70, is the average salary for males (as we coded gender=1 for females and 0 for males)

- The value of the slope, -1.175, tells us that the average females salary is lower than the average male salary by 1.175

In [25]:

```python
step_1 = pd.concat([tbl2, just_dummies2], axis=1)
step_1.drop(['Gender', 'Male'], inplace=True, axis=1)
# to run the regression we want to get rid of the strings 'male' and 'female'
# and we want to get rid of one dummy variable to avoid the dummy variable trap
# arbitrarily chose "male", coefficients on "female" would show effect of "female"
# relative to "male"

result = sm.OLS(step_1['Salary'], s.add_constant(step_1[['Experience', 'Female']])).fit()
print (result.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Salary   R-squared:                       0.974
Model:                            OLS   Adj. R-squared:                  0.965
Method:                 Least Squares   F-statistic:                     111.6
Date:                Sat, 07 Sep 2019   Prob (F-statistic):           1.80e-05
Time:                        12:33:40   Log-Likelihood:                -1.5752
No. Observations:                   9   AIC:                             9.150
Df Residuals:                       6   BIC:                             9.742
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          6.2485      0.291     21.439      0.000       5.535       6.962
Experience     0.2271      0.016     14.089      0.000       0.188       0.267
Female        -0.7890      0.238     -3.309      0.016      -1.372      -0.206
==============================================================================
Omnibus:                        0.110   Durbin-Watson:                   2.181
Prob(Omnibus):                  0.947   Jarque-Bera (JB):                0.198
Skew:                           0.174   Prob(JB):                        0.906
Kurtosis:                       2.363   Cond. No.                         44.8
==============================================================================
```

$y = 6.2485 +$
$0.2271\ Exp$
$- 0.7890\ F$

What would have happened if we had used 0 for females and 1 for males in our data? Would our results be any different?

In [63]:
```
step_1 = pd.concat([tbl2, just_dummies2], axis=1)
step_1.drop(['Gender', 'Female'], inplace=True, axis=1)


result = sm.OLS(step_1['Salary'], s.add_constant(step_1[['Male']])).fit()
print (result.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Salary   R-squared:                       0.107
Model:                            OLS   Adj. R-squared:                 -0.020
Method:                 Least Squares   F-statistic:                    0.8426
Date:                Sat, 07 Sep 2019   Prob (F-statistic):              0.389
Time:                        14:27:56   Log-Likelihood:                -17.455
No. Observations:                   9   AIC:                             38.91
Df Residuals:                       7   BIC:                             39.30
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          8.5250      0.954      8.935      0.000       6.269      10.781
Male           1.1750      1.280      0.918      0.389      -1.852       4.202
==============================================================================
Omnibus:                        0.387   Durbin-Watson:                   1.912
Prob(Omnibus):                  0.824   Jarque-Bera (JB):                0.280
Skew:                           0.330   Prob(JB):                        0.869
Kurtosis:                       2.441   Cond. No.                         2.77
==============================================================================
```

Male = 1, female = 0

- Not really – With coding as above, the intercept would change to 8.525 (the average female salary), the slope for gender would still be 1.175, but now it would have a positive sign (reflecting that average male salary is higher than average female salary by 1.175). *Predicted salaries from the model for males / females would not change no matter how dummy variable is coded*

More on dummy variables

- For gender, we had only 2 categories – female and male – thus we used a single 0/1 variable for this
- When there are more than 2 categories, the number of dummy variables that should be used equals the number of categories minus 1
- No. of Dummy Variables = No. of levels -1

Example: Salary vs. Job Grade

- In this example, the categorical variable job grade has 3 levels, 1 (lowest grade), 2, and 3 (highest job grade)

Employee	Job Grade	Salary ($000)
1	1	7.5
2	3	8.6
3	2	9.1
4	3	10.3
5	3	13
6	1	6.2
7	2	8.7
8	2	9.4
9	3	9.8

Representing 3-level Job Grade using dummy variables Job_1 and Job_2

1 2

Dummy Variables

Employee's Job Grade

Job Grade	Job_1	Job_2
1	1	0
2	0	1
3	0	0

Job Grade 3 is the reference category

Data file with dummy variables for job grade

Employee	Job Grade	Salary	Job_1	Job_2
1	1	7.5	1	0
2	3	8.6	0	0
3	2	9.1	0	1
4	3	10.3	0	0
5	3	13	0	0
6	1	6.2	1	0
7	2	8.7	0	1
8	2	9.4	0	1
9	3	9.8	0	0

Thank You