Econometrics_Wk1Test_WSchill

October 7, 2016

0.0.1 Econometrics - Test Week 1

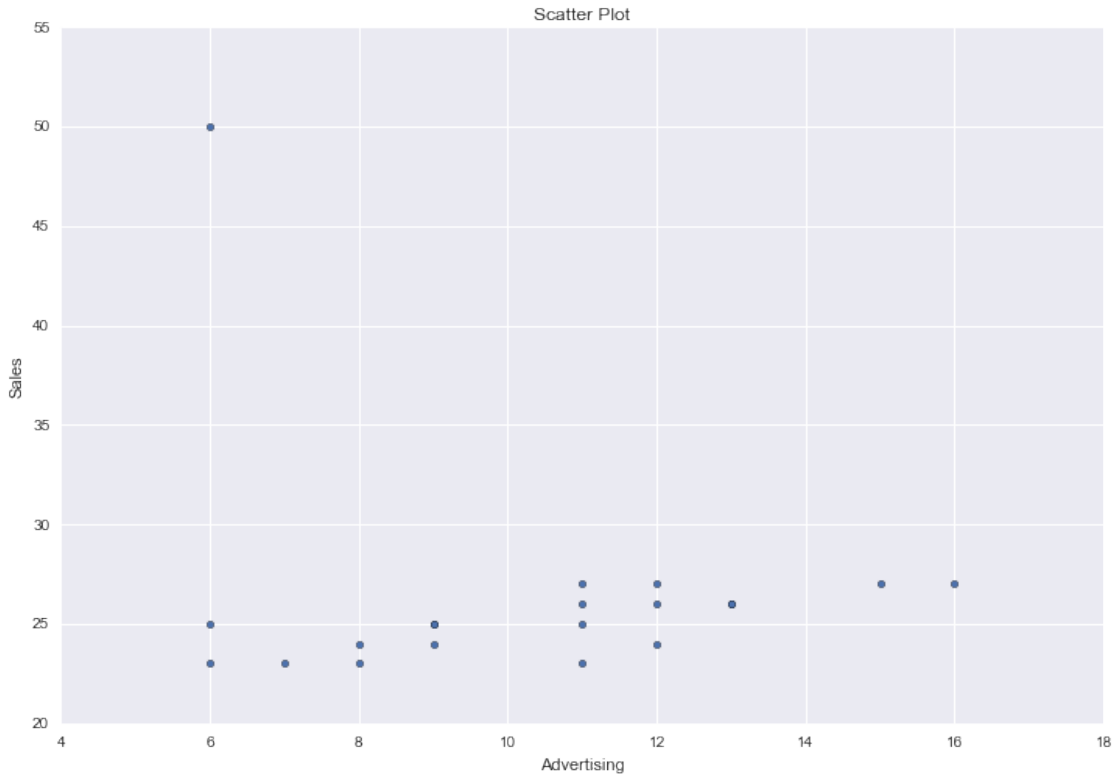
William Schill *Questions*

This exercise considers an example of data that do not satisfy all the standard assumptions of simple regression. In the considered case, one particular observation lies far off from the others, that is, it is an outlier. This violates assumptions A3 and A4, which state that all error terms ϵ_i are drawn from one and the same distribution with mean zero and fixed variance, i.e. $\mu = 0$ and $sigma^2$. The dataset contains twenty weekly observations on sales and advertising of a department store. The question of interest lies in estimating the effect of advertising on sales. One of the weeks was special, as the store was also open in the evenings during this week, but this aspect will first be ignored in the analysis.

(a) Make the scatter diagram with sales on the vertical axis and advertising on the horizontal axis. What do you expect to find if you would fit a regression line to these data?

```
In [2]: import numpy as np
        import pandas as pd
        import seaborn as sns
        %matplotlib inline
        dirc = 'C:\\Users\\SchillW\\Documents\\Econ_Coursera\\Wk1\\'
        f = 'TestExer1-sales-round1.xls'
        df = pd.read_excel(dirc + f)
        df.plot.scatter(y='Sales',x='Advertising',
                        figsize=(12,8),title='Scatter Plot')
```

```
Out[2]: <matplotlib.axes._subplots.AxesSubplot at 0x9d6b278>
```



Assuming the data is normal for all of the questions responded to here, if we were trying to fit a regression line to this data, we would find that the regression was not a good fit due to the presence of the outliar at sales = 50.

(b) Estimate the coefficients a and b in the simple regression model with sales as dependent variable and advertising as explanatory factor. Also compute the standard error and t-value of b. Is b significantly different from 0?

```
In [3]: xwk1 = np.reshape(df['Advertising'],(len(df),1))
        ywk1 = np.reshape(df['Sales'],(len(df),1))

        import statsmodels.api as sm
        xwk1i = sm.add_constant(xwk1)
        mod = sm.OLS(ywk1,xwk1i)
        res = mod.fit()
        print res.summary()
```

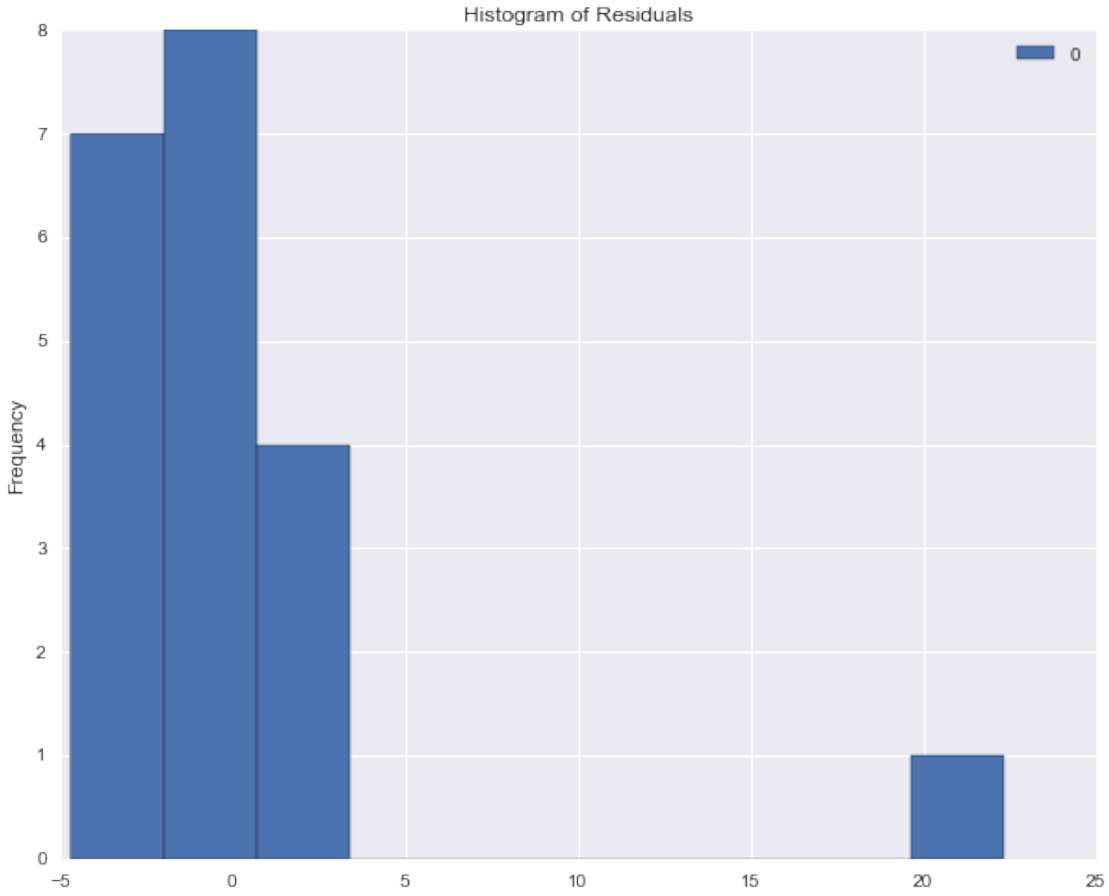
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.027
Model:                            OLS   Adj. R-squared:                 -0.027
Method:                 Least Squares   F-statistic:                    0.5002
Date:                Fri, 07 Oct 2016   Prob (F-statistic):              0.488
Time:                        15:35:21   Log-Likelihood:                -62.608
No. Observations:                  20   AIC:                             129.2
```

```
Df Residuals:                        18   BIC:                                131.2
Df Model:                             1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         29.6269      4.882      6.069      0.000      19.371      39.883
x1            -0.3246      0.459     -0.707      0.488      -1.289       0.640
==============================================================================
Omnibus:                       40.109   Durbin-Watson:                   1.994
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              121.776
Skew:                           3.178   Prob(JB):                     3.60e-27
Kurtosis:                      13.283   Cond. No.                         40.1
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly sp
```

In this example of statsmodels, const refers to our 'a' intercept and x1 refers to our first coefficient 'b'. The values are listed under the heading 'coef' for each. It can be seen by the the regression results summary that the standard error for b is 0.459 and the t-value for b = -0.707. Assuming the null hypothesis (H0) is that b=0, then by the calculated p-value for the t-test, b is not significantly different from zero. Here P ~ 0.488 where the significance of the variable decreases as you approach 1.

(c) Compute the residuals and draw a histogram of these residuals. What conclusion do you draw from this histogram?

```
In [4]: yhat_wk1 = np.reshape(xwk1i.dot(res.params), (len(ywk1),1))
        resids = pd.DataFrame( (ywk1-yhat_wk1) )
        resids.plot.hist(['g'], title='Histogram of Residuals',
                         figsize=(10,8))

Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0xa915588>
```



It can be seen from the histogram of the residuals that the outliar is causing a significant spike in the error and setting a poor regression estimation. The residuals are also not normally distributed.

(d) Apparently, the regression result of part (b) is not satisfactory. Once you realize that the large residual corresponds to the week with opening hours during the evening, how would you proceed to get a more satisfactory regression model?

One way to proceed would be to change the outliar to the mean or the median of the data, potentially resulting in an improved and higher likeliness fit. From moving forward into the questions on the test, it is clear that another possibility is removing the point altogether.

(e) Delete this special week from the sample and use the remaining 19 weeks to estimate the coefficients a and b in the simple regression model with sales as dependent variable and advertising as explanatory factor. Also compute the standard error and t-value of b. Is b significantly different from 0?

```
In [5]: df1 = df[df.Sales<50] #remove the sales=50 entry
        xwk1a = np.reshape(df1['Advertising'],(len(df1),1))
        ywk1a = np.reshape(df1['Sales'],(len(df1),1))
        xwk1ai = sm.add_constant(xwk1a)
        moda = sm.OLS(ywk1a,xwk1ai)
        resa = moda.fit()
        print resa.summary()
```

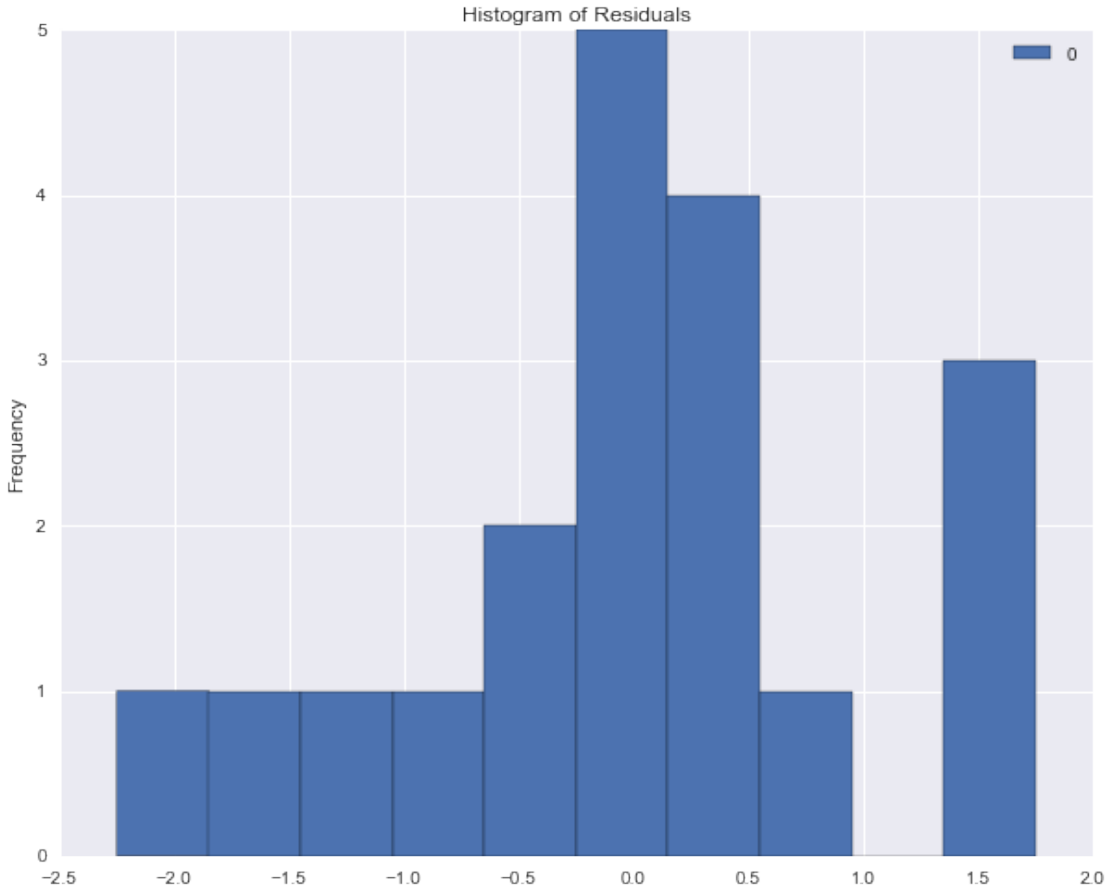
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.515
Model:                            OLS   Adj. R-squared:                  0.487
Method:                 Least Squares   F-statistic:                     18.08
Date:                Fri, 07 Oct 2016   Prob (F-statistic):           0.000538
Time:                        15:35:27   Log-Likelihood:                -26.897
No. Observations:                  19   AIC:                             57.79
Df Residuals:                      17   BIC:                             59.68
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         21.1250      0.955     22.124      0.000      19.110      23.140
x1             0.3750      0.088      4.252      0.001       0.189       0.561
==============================================================================
Omnibus:                        0.597   Durbin-Watson:                   1.749
Prob(Omnibus):                  0.742   Jarque-Bera (JB):                0.204
Skew:                          -0.252   Prob(JB):                        0.903
Kurtosis:                       2.933   Cond. No.                         43.1
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly sp


C:\Users\SchillW\Anaconda2\lib\site-packages\scipy\stats\stats.py:1557: UserWarning
  "anyway, n=%i" % int(n))
```

The regression imporves markedly. Assuming that H0 => b=0, then in this instance with the outliar removed, the value of b is statistically significant and not close to zero since P > |t| ~ 0.001.

(e) Discuss the differences between your findings in parts (b) and (e). Describe in words what you have learned from these results.

It is apparent that the presence of an outliar within nomally distributed data can have hard consequences on the predictive capabilites of models. In this scenario, the outliar at 50 was more than three standard deviations away from the mean of the remainder of the data thus causing a significant skew in linear regression modeling.

```
In [7]: yhat_wk1a = np.reshape(xwk1ai.dot(resa.params), (len(ywk1a),1))
        resids2 = pd.DataFrame( (ywk1a-yhat_wk1a) )
        resids2.plot.hist(['g'], title='Histogram of Residuals',
                          figsize=(10,8))

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0xacc1160>
```



A re-examination of the residuals now shows a much more normal distribution. For this case we cannot reject H0 as b is significantly different than zero.

In [ ]: