# First Name: Last Name:

In [1]:

**import** pandas **as** pd **import** numpy **as** np **import** seaborn **as** sns

**import** matplotlib.pyplot **as** plt

**import** statsmodels.formula.api **as** smf

In [2]:

pd.set\_option('display.float\_format', **lambda** x:'%.2f'**%**x)

gapminder **=** pd.read\_csv('gapminder.csv', low\_memory**=False**) gapminder.head()

Out[2]:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **country** | **incomeperperson** | **alcconsumption** | **armedforcesrate** | **breastcancerper100th** | **c** |
| **0** Afghanistan |  | .03 | .5696534 | 26.8 |  |
| **1** Albania | 1914.99655094922 | 7.29 | 1.0247361 | 57.4 | 22374 |
| **2** Algeria | 2231.99333515006 | .69 | 2.306817 | 23.5 | 29321 |
| **3** Andorra | 21943.3398976022 | 10.17 |  |  |  |
| **4** Angola | 1381.00426770244 | 5.57 | 1.4613288 | 23.1 |  |

In [3]:

gapminder['oilperperson'] **=** pd.to\_numeric(gapminder['oilperperson'],errors**=**'coerce') gapminder['relectricperperson'] **=** pd.to\_numeric(gapminder['relectricperperson'],errors**=**'coe gapminder['co2emissions'] **=** pd.to\_numeric(gapminder['co2emissions'],errors**=**'coerce')

# Scenario 1 - Linear & Multiple sub1

In [4]:

sub1 **=** gapminder[['oilperperson', 'relectricperperson', 'co2emissions']].dropna() sub1.head()

Out[4]:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **oilperperson** | **relectricperperson** | **co2emissions** |
| **2** | 0.42 | 590.51 | 2932108666.67 |
| **6** | 0.64 | 768.43 | 5872119000.00 |
| **9** | 1.91 | 2825.39 | 12970092666.67 |
| **10** | 1.55 | 2068.12 | 4466084333.33 |
| **11** | 0.36 | 921.56 | 511107666.67 |

# Centre oilperperson, relectricperperson and co2emissions

**use sub1**

In [5]:

*# center quantitative variables for regression analysis*

sub1['oilperperson\_c'] **=** (sub1['oilperperson'] **-** sub1['oilperperson'].mean()) sub1['relectricperperson\_c'] **=** (sub1['relectricperperson'] **-** sub1['relectricperperson'].mea sub1['co2emissions\_c'] **=** (sub1['co2emissions'] **-** sub1['co2emissions'].mean())

sub1.head()

Out[5]:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **oilperperson** | **relectricperperson** | **co2emissions** | **oilperperson\_c** | **relectricperperson\_c** | **co2e** |
| **2** | 0.42 | 590.51 | 2932108666.67 | -1.06 | -1145.94 | -1235 |
| **6** | 0.64 | 768.43 | 5872119000.00 | -0.85 | -968.02 | -941 |
| **9** | 1.91 | 2825.39 | 12970092666.67 | 0.43 | 1088.94 | -231 |
| **10** | 1.55 | 2068.12 | 4466084333.33 | 0.06 | 331.68 | -1081 |
| **11** | 0.36 | 921.56 | 511107666.67 | -1.12 | -814.89 | -1477 |

# Multi variable linear regression

**predict co2emission (y) using relectricperperson(x1) and oilperperson(x2)**

# use sub1

In [20]:

reg1 **=** smf.ols('co2emissions\_c ~ relectricperperson\_c + oilperperson\_c', data**=**sub1).fit() print (reg1.summary())

OLS Regression Results

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ============================================================================  == | | | | | | |
| Dep. Variable:  20  Model: | co2emissions\_c  OLS | | R-squared:  Adj. R-squared: | | 0.0  -0.0 | |
| 12  Method: | Least Squares | | F-statistic: | | 0.62 | |
| 05  Date: | Sun, 25 Mar 2018 | | Prob (F-statistic): | | 0.5 | |
| 41  Time: | 12:57:49 | | Log-Likelihood: | | -163 | |
| 2.7  No. Observations: | 63 | | AIC: | | 327 | |
| 1.  Df Residuals: | 60 | | BIC: | | 327 | |
| 8.  Df Model: | 2 | |  | |  | |
| Covariance Type: | nonrobust | |  | |  | |
| ============================================================================  ============ | | | | | | |
| 0.975] | coef | std err | | t | P>|t| | [0.025 |
|  |  |  | |  |  |  |
| Intercept 1.12e+10  relectricperperson\_c | -1.669e-06  3.434e+06 | 5.62e+09  3.24e+06 | | -2.97e-16  1.058 | 1.000  0.294 | -1.12e+10  -3.06e+06 |
| 9.92e+06  oilperperson\_c | -9.47e+08 | 3.65e+09 | | -0.260 | 0.796 | -8.25e+09 |
| 6.35e+09 |  |  | |  |  |  |

============================================================================

==

|  |  |  |  |
| --- | --- | --- | --- |
| Omnibus: | 116.246 | Durbin-Watson: | 1.7 |
| 62 |  |  |  |
| Prob(Omnibus): | 0.000 | Jarque-Bera (JB): | 4225.1 |
| 98 |  |  |  |
| Skew: | 5.891 | Prob(JB): | 0. |
| 00 |  |  |  |
| Kurtosis: | 41.351 | Cond. No. | 2.04e+ |
| 03 |  |  |  |

============================================================================

==

Warnings:

1. Standard Errors assume that the covariance matrix of the errors is corre ctly specified.
2. The condition number is large, 2.04e+03. This might indicate that there are

strong multicollinearity or other numerical problems.

# Scenario 2 - Linear

**sub2**

In [7]:

*# convert to numeric format*

gapminder['employrate'] **=** pd.to\_numeric(gapminder['employrate'], errors**=**'coerce') sub2 **=** gapminder[['relectricperperson', 'employrate']].dropna()

sub2.head()

Out[7]:

**relectricperperson employrate 1** 636.34 51.40

**2** 590.51 50.50

**4** 173.00 75.70

**6** 768.43 58.40

**7** 603.76 40.10

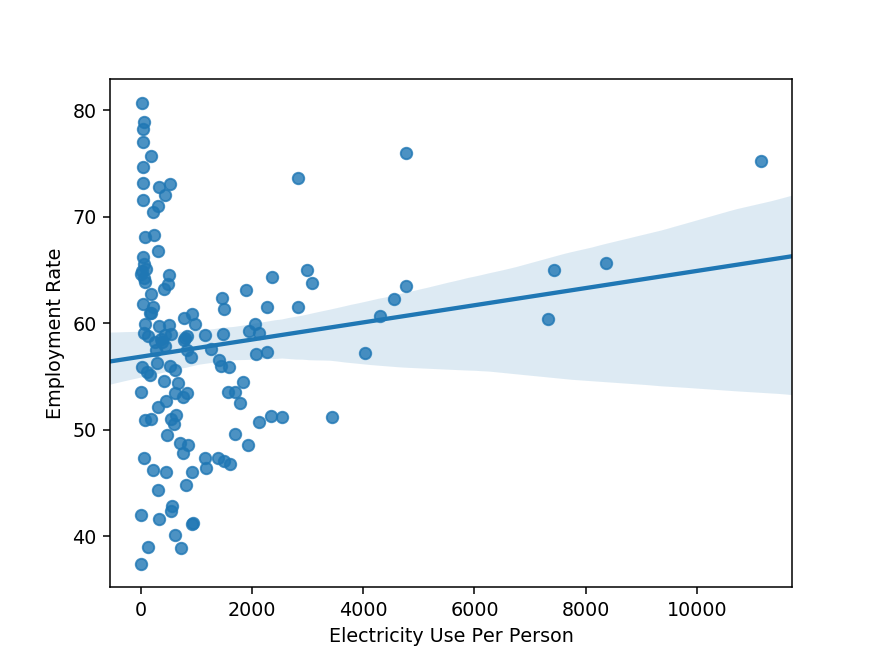
# scatter plot to show relationship between employment rate (x) and electricity use per person (y)

In [8]:

**%**matplotlib notebook plt.figure()

scat1 **=** sns.regplot(x**=**"relectricperperson", y**=**"employrate", fit\_reg**=True**, data**=**sub2)

plt.xlabel('Electricity Use Per Person') plt.ylabel('Employment Rate')



Out[8]:

Text(0,0.5,'Employment Rate')

# Centre relectricperperson and employrate use sub2

In [9]:

sub2['relectricperperson\_c'] **=** (sub2['relectricperperson'] **-** sub2['relectricperperson'].mea sub2['employrate\_c'] **=** (sub2['employrate'] **-** sub2['employrate'].mean())

sub2.head()

Out[9]:

**relectricperperson employrate relectricperperson\_c employrate\_c 1** 636.34 51.40 -543.99 -6.41

**2** 590.51 50.50 -589.82 -7.31

**4** 173.00 75.70 -1007.33 17.89

**6** 768.43 58.40 -411.90 0.59

**7** 603.76 40.10 -576.57 -17.71

# Linear regression between relectricperperson (x) and employrate (y)

**use sub2**

In [10]:

reg2 **=** smf.ols('employrate\_c ~ relectricperperson\_c', data**=**sub2).fit() print (reg2.summary())

OLS Regression Results

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ============================================================================  == | | | | | | |
| Dep. Variable:  21  Model: | employrate\_c  OLS | | R-squared:  Adj. R-squared: | | 0.0  0.0 | |
| 14  Method: | Least Squares | | F-statistic: | | 2.8 | |
| 77  Date: | Sun, 25 Mar 2018 | | Prob (F-statistic): | | 0.09 | |
| 22  Time: | 12:48:40 | | Log-Likelihood: | | -487. | |
| 37  No. Observations: | 134 | | AIC: | | 97 | |
| 8.7  Df Residuals: | 132 | | BIC: | | 98 | |
| 4.5  Df Model: | 1 | |  | |  | |
| Covariance Type: | nonrobust | |  | |  | |
| ============================================================================  ============ | | | | | | |
| 0.975] | coef | std err | | t | P>|t| | [0.025 |
|  |  |  | |  |  |  |
| Intercept 1.582  relectricperperson\_c | 5.662e-15  0.0008 | 0.800  0.000 | | 7.08e-15  1.696 | 1.000  0.092 | -1.582  -0.000 |
| 0.002 |  |  | |  |  |  |

============================================================================

==

|  |  |  |  |
| --- | --- | --- | --- |
| Omnibus: | 1.259 | Durbin-Watson: | 2.0 |
| 02 |  |  |  |
| Prob(Omnibus): | 0.533 | Jarque-Bera (JB): | 1.2 |
| 53 |  |  |  |
| Skew: | 0.228 | Prob(JB): | 0.5 |
| 34 |  |  |  |
| Kurtosis: | 2.874 | Cond. No. | 1.68e+ |
| 03 |  |  |  |

============================================================================

==

Warnings:

1. Standard Errors assume that the covariance matrix of the errors is corre ctly specified.
2. The condition number is large, 1.68e+03. This might indicate that there are

strong multicollinearity or other numerical problems.

# Scenario 3 - Polynomial

**scatter plot to show polynomial (order 2) relationship between employment rate (x) and electricity use per person (y)**

In [11]:

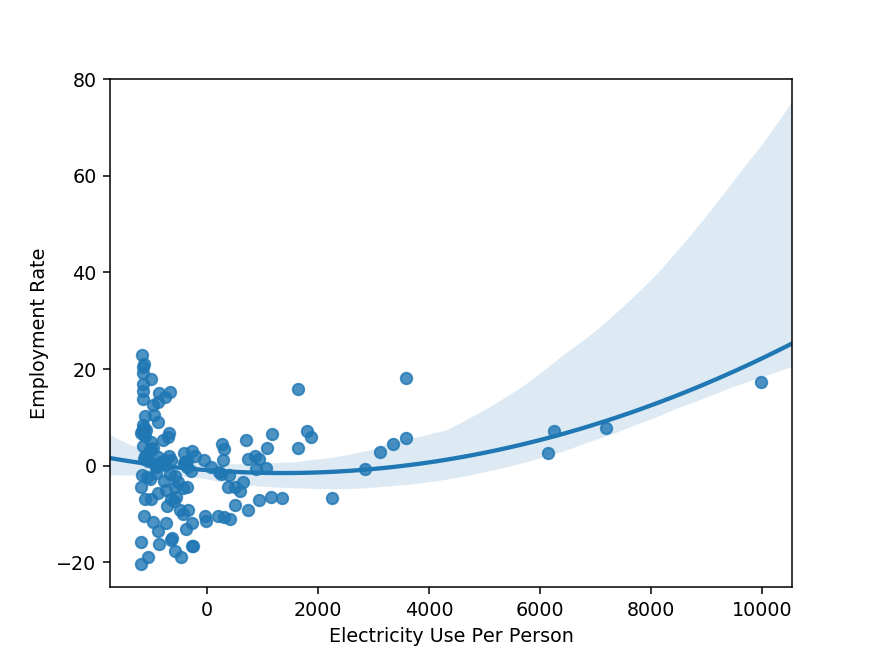
*#fit second order polynomial*

*# run the 2 scatterplots together to get second order fit lines*

plt.figure()

scat1 **=** sns.regplot(x**=**"relectricperperson\_c", y**=**"employrate\_c", order**=**2, data**=**sub2) plt.xlabel('Electricity Use Per Person')

plt.ylabel('Employment Rate')



Out[11]:

Text(0,0.5,'Employment Rate')

# Polynomial regression between relectricperperson (x - order 2) and employrate (y)

**use sub2**

In [12]:

reg2 **=** smf.ols('employrate\_c ~ I(relectricperperson\_c\*\*2)', data**=**sub2).fit() print (reg2.summary())

OLS Regression Results

============================================================================

==

|  |  |  |  |
| --- | --- | --- | --- |
| Dep. Variable: | employrate\_c | R-squared: | 0.0 |
| 54 |  |  |  |
| Model: | OLS | Adj. R-squared: | 0.0 |
| 47 |  |  |  |
| Method: | Least Squares | F-statistic: | 7.6 |
| 06 |  |  |  |
| Date: | Sun, 25 Mar 2018 | Prob (F-statistic): | 0.006 |
| 64 |  |  |  |
| Time: | 12:49:22 | Log-Likelihood: | -485. |
| 06 |  |  |  |
| No. Observations: | 134 | AIC: | 97 |
| 4.1 |  |  |  |
| Df Residuals: | 132 | BIC: | 97 |
| 9.9 |  |  |  |
| Df Model: | 1 |  |  |
| Covariance Type: | nonrobust |  |  |

============================================================================

====================

coef std err t P>|t|

[0.025 0.975]

Intercept -0.5780 0.814 -0.710 0.479

-2.187 1.032

I(relectricperperson\_c \*\* 2) 2.037e-07 7.39e-08 2.758 0.007

5.76e-08 3.5e-07

============================================================================

==

|  |  |  |  |
| --- | --- | --- | --- |
| Omnibus: | 0.919 | Durbin-Watson: | 2.0 |
| 29 |  |  |  |
| Prob(Omnibus): | 0.632 | Jarque-Bera (JB): | 0.8 |
| 72 |  |  |  |
| Skew: | 0.194 | Prob(JB): | 0.6 |
| 47 |  |  |  |
| Kurtosis: | 2.924 | Cond. No. | 1.14e+ |
| 07 |  |  |  |

============================================================================

==

Warnings:

1. Standard Errors assume that the covariance matrix of the errors is corre ctly specified.
2. The condition number is large, 1.14e+07. This might indicate that there are

strong multicollinearity or other numerical problems.

# Scenario 4 - Multiple & poly

**sub3**

In [13]:

sub3 **=** gapminder[['oilperperson', 'relectricperperson', 'co2emissions','employrate']].dropn sub3.head()

Out[13]:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **oilperperson** | **relectricperperson** | **co2emissions** | **employrate** |
| **2** | 0.42 | 590.51 | 2932108666.67 | 50.50 |
| **6** | 0.64 | 768.43 | 5872119000.00 | 58.40 |
| **9** | 1.91 | 2825.39 | 12970092666.67 | 61.50 |
| **10** | 1.55 | 2068.12 | 4466084333.33 | 57.10 |
| **11** | 0.36 | 921.56 | 511107666.67 | 60.90 |

# Centre employrate, oilperperson, relectricperperson and co2emissions

**use sub3**

In [14]:

sub3['employrate\_c'] **=** (sub3['employrate'] **-** sub3['employrate'].mean()) sub3['oilperperson\_c'] **=** (sub3['oilperperson'] **-** sub3['oilperperson'].mean()) sub3['relectricperperson\_c'] **=** (sub3['relectricperperson'] **-** sub3['relectricperperson'].mea sub3['co2emissions\_c'] **=** (sub3['co2emissions'] **-** sub3['co2emissions'].mean())

# Multiple and polynomial regression between oilperperson(x1) + co2emissions(x2) relectricperperson(x3 - order 2) and employrate (y)

**use sub3**

In [15]:

reg3 **=** smf.ols('employrate\_c ~ oilperperson\_c + co2emissions\_c + I(relectricperperson\_c\*\*2) print (reg3.summary())

OLS Regression Results

============================================================================

==

|  |  |  |  |
| --- | --- | --- | --- |
| Dep. Variable: | employrate\_c | R-squared: | 0.1 |
| 86 |  |  |  |
| Model: | OLS | Adj. R-squared: | 0.1 |
| 44 |  |  |  |
| Method: | Least Squares | F-statistic: | 4.4 |
| 81 |  |  |  |
| Date: | Sun, 25 Mar 2018 | Prob (F-statistic): | 0.006 |
| 70 |  |  |  |
| Time: | 12:50:27 | Log-Likelihood: | -210. |
| 13 |  |  |  |
| No. Observations: | 63 | AIC: | 42 |
| 8.3 |  |  |  |
| Df Residuals: | 59 | BIC: | 43 |
| 6.8 |  |  |  |
| Df Model: | 3 |  |  |
| Covariance Type: | nonrobust |  |  |

============================================================================

====================

coef std err t P>|t|

[0.025 0.975]

Intercept -0.8489 0.937 -0.906 0.369

-2.724 1.026

oilperperson\_c 0.6155 0.522 1.179 0.243

-0.429 1.660

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| co2emissions\_c | 1.533e-11 | 2.01e-11 | 0.761 | 0.450 |
| -2.5e-11 5.56e-11 |  |  |  |  |
| I(relectricperperson\_c | \*\* 2) 2.047e-07 | 7.43e-08 | 2.755 | 0.008 |
| 5.6e-08 3.53e-07 |  |  |  |  |

============================================================================

==

|  |  |  |  |
| --- | --- | --- | --- |
| Omnibus: | 0.228 | Durbin-Watson: | 2.3 |
| 24 |  |  |  |
| Prob(Omnibus): | 0.892 | Jarque-Bera (JB): | 0.0 |
| 68 |  |  |  |
| Skew: | 0.080 | Prob(JB): | 0.9 |
| 67 |  |  |  |
| Kurtosis: | 2.998 | Cond. No. | 4.67e+ |
| 10 |  |  |  |

============================================================================

==

Warnings:

1. Standard Errors assume that the covariance matrix of the errors is corre ctly specified.
2. The condition number is large, 4.67e+10. This might indicate that there are

strong multicollinearity or other numerical problems.

# Evaluating model

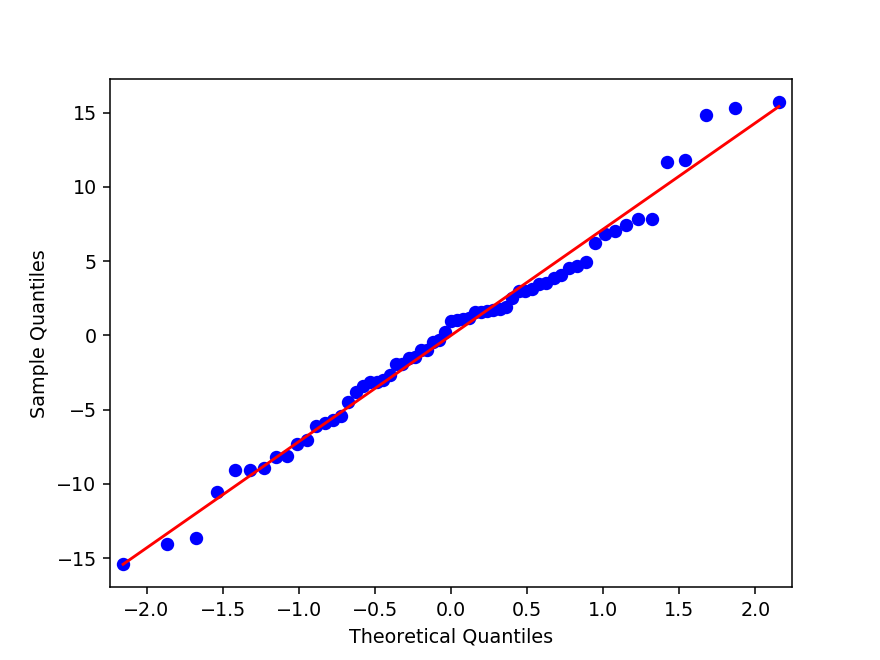
**Plot qqplot for the above regression (reg3)**

In [16]:

**import** statsmodels.api **as** sm fig4**=**sm.qqplot(reg3.resid, line**=**'r')

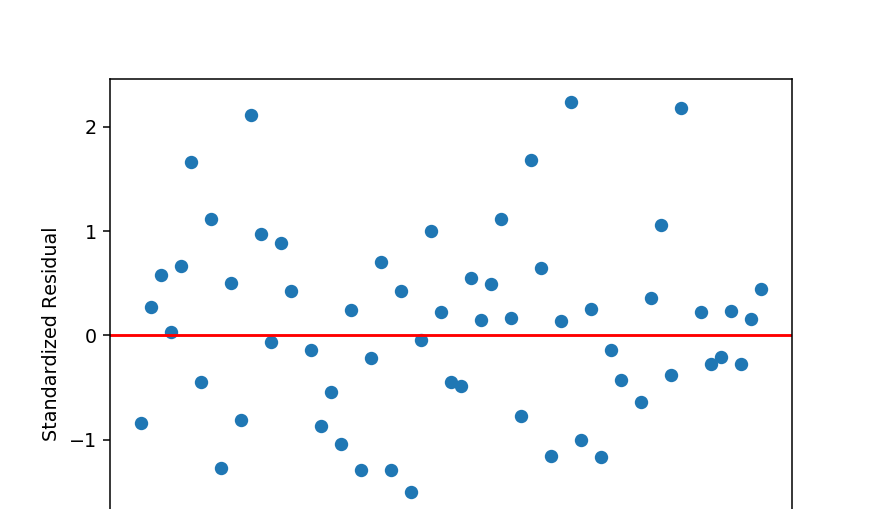
C:\Users\jc443343\AppData\Local\Continuum\anaconda3\lib\site-packages\statsm odels\compat\pandas.py:56: FutureWarning: The pandas.core.datetools module i s deprecated and will be removed in a future version. Please use the pandas. tseries module instead.

from pandas.core import datetools



# Residual plot for the above regression (reg3)

In [17]:



*# simple plot of residuals* stdres**=**pd.DataFrame(reg3.resid\_pearson) plt.figure()

plt.plot(stdres, 'o', ls**=**'None') l **=** plt.axhline(y**=**0, color**=**'r')

plt.ylabel('Standardized Residual') plt.xlabel('Observation Number')

# Calculate percentage of observations over 2 standardized deviation

In [18]:

percentage\_over2sd **=** (np.count\_nonzero( stdres[0] **>** 2) **+** np.count\_nonzero( stdres[0] **< -**2)) print (percentage\_over2sd)

7.936507936507936

# Calculate percentage of observations over 2.5 standardized deviation

In [19]:

percentage\_over2\_5sd **=** (np.count\_nonzero( stdres[0] **>** 2.5) **+** np.count\_nonzero( stdres[0] **<**

print (percentage\_over2\_5sd)

0.0

# Example answer - students can do any combination

**On your own, perform**

# Multiple and polynomial regression between oilperperson, co2emissions, relectricperperson to predict employrate (y)

**experiment with explanatory variable (oilperperson, co2emissions, relectricperperson) and their order**

# use sub3

In [39]:

reg4 **=** smf.ols('employrate\_c ~ oilperperson\_c + I(co2emissions\_c\*\*2) + I(relectricperperson print (reg4.summary())

OLS Regression Results

============================================================================

==

|  |  |  |  |
| --- | --- | --- | --- |
| Dep. Variable: | employrate\_c | R-squared: | -0.3 |
| 96 |  |  |  |
| Model: | OLS | Adj. R-squared: | -0.4 |
| 19 |  |  |  |
| Method: | Least Squares | F-statistic: | -17. |
| 31 |  |  |  |
| Date: | Sun, 25 Mar 2018 | Prob (F-statistic): | 1. |
| 00 |  |  |  |
| Time: | 13:27:39 | Log-Likelihood: | -227. |
| 11 |  |  |  |
| No. Observations: | 63 | AIC: | 45 |
| 8.2 |  |  |  |
| Df Residuals: | 61 | BIC: | 46 |
| 2.5 |  |  |  |
| Df Model: | 1 |  |  |
| Covariance Type: | nonrobust |  |  |

============================================================================

====================

coef std err t P>|t|

[0.025 0.975]

Intercept 4.706e-15 1.92e-15 2.449 0.017

8.64e-16 8.55e-15

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| oilperperson\_c | -7.646e-23 | 3.12e-23 | -2.449 | 0.017 | - |
| 1.39e-22 -1.4e-23 |  |  |  |  |  |
| I(co2emissions\_c \*\* 2) | -3.936e-22 | 2.03e-22 | -1.934 | 0.058 |  |
| -8e-22 1.33e-23 |  |  |  |  |  |
| I(relectricperperson\_c | \*\* 2) 2.095e-07 | 8.55e-08 | 2.449 | 0.017 |  |
| 3.84e-08 3.81e-07 |  |  |  |  |  |

============================================================================

==

|  |  |  |  |
| --- | --- | --- | --- |
| Omnibus: | 39.077 | Durbin-Watson: | 2.1 |
| 99 |  |  |  |
| Prob(Omnibus): | 0.000 | Jarque-Bera (JB): | 155.1 |
| 98 |  |  |  |
| Skew: | 1.682 | Prob(JB): | 1.99e- |
| 34 |  |  |  |
| Kurtosis: | 9.915 | Cond. No. | 1.41e+ |
| 30 |  |  |  |

============================================================================

==

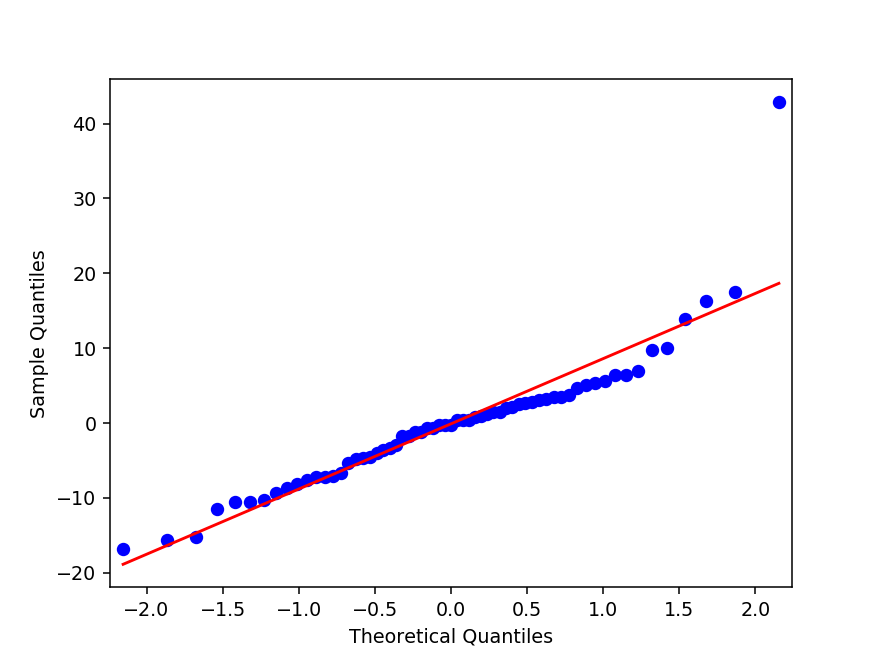
Warnings:

1. Standard Errors assume that the covariance matrix of the errors is corre ctly specified.
2. The smallest eigenvalue is 5.22e-15. This might indicate that there are strong multicollinearity problems or that the design matrix is singular.

# Evaluate your model Use ggplot

In [40]:

**import** statsmodels.api **as** sm fig5**=**sm.qqplot(reg4.resid, line**=**'r')



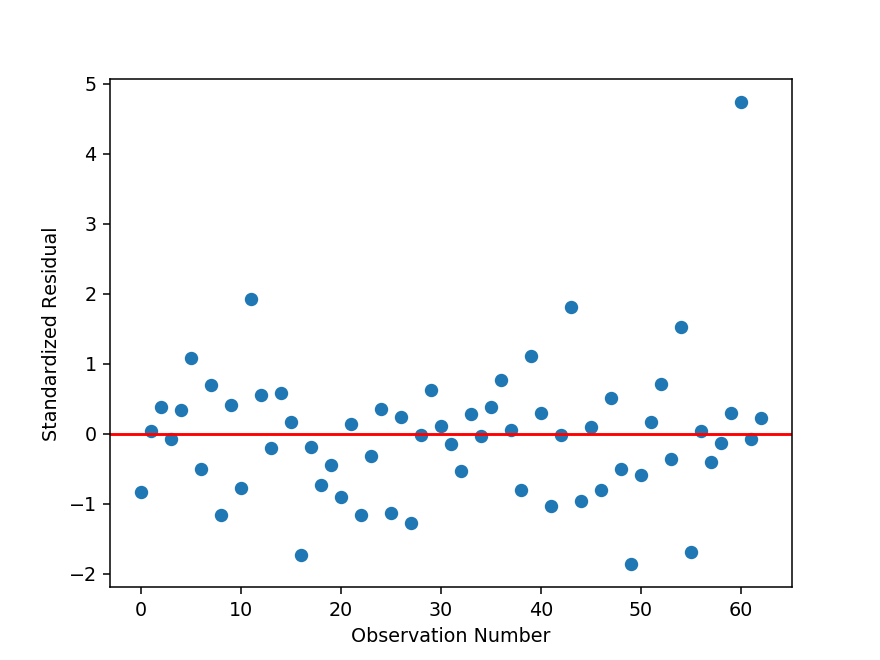
# Evaluate your model Use residual plot

In [41]:

*# simple plot of residuals* stdres**=**pd.DataFrame(reg4.resid\_pearson) plt.figure()

plt.plot(stdres, 'o', ls**=**'None') l **=** plt.axhline(y**=**0, color**=**'r')

plt.ylabel('Standardized Residual') plt.xlabel('Observation Number')



Out[41]:

Text(0.5,0,'Observation Number')

# Calculate percentage of observations over 2 standardized deviation

In [42]:

percentage\_over2sd **=** (np.count\_nonzero( stdres[0] **>** 2) **+** np.count\_nonzero( stdres[0] **< -**2)) print (percentage\_over2sd)

1.5873015873015872

# Calculate percentage of observations over 2.5 standardized deviation

In [43]:

percentage\_over2\_5sd **=** (np.count\_nonzero( stdres[0] **>** 2.5) **+** np.count\_nonzero( stdres[0] **<**

print (percentage\_over2\_5sd)

1.5873015873015872