Chapter exercise3 from ISLR book

Exercise 3.1

''' The null hypotheses to Intercept is that means Except the benefit from the TV, radio and newspaper , other benefit – other budget is larger than 0, which rejects the H_0 which is cost = budget in more than 99%.

The null hypotheses to TV is that the benefit of TV ads possibility not equal than the budget of TV ads. There exists more than 99% possibility that benefit of TV ads is not equal than cost of it.

The null hypotheses to Radio is that the benefit of radio ads possibility not equal than the budget of radio ads. There exists more than 99% possibility that benefit of radio ads is not equal than cost of it.

The null hypotheses to newspaper is that the benefit of newspaper ads possibility not equal than the budget of newspaper ads. There exists more than 99% possibility that benefit of newspaper ads is not equal than cost of it. '''

Exercise 3.2

KNN regression averages the closest observations to estimate prediction

KNN classifier assigns classification group based on majority of closest observations.

Exercise 3.5

In [2]:

```
from IPython.display import Latex
Latex(r"""\begin{eqnarray}
\hat{y}_{i} = x_{i} \times \frac{\sum_{i'=1}^{n}\left ( x_{i'} y_{i'} \right )}{\sum_{j=1}^{n} x_{j}^{2}} \\
\hat{y}_{i} = \sum_{i'=1}^{n} \frac{\left ( x_{i'} y_{i'} \right ) \times x_{i}}{\sum_{j=1}^{n} x_{j}^{2}} \\
\hat{y}_{i} = \sum_{i'=1}^{n} \left ( \frac{ x_{i} x_{i'} } { \sum_{j=1}^{n} x_{j}^{2} } \times y_{i'} \right ) \\
a_{i'} = \frac{ x_{i} x_{i'} } { \sum_{j=1}^{n} x_{j}^{2} } \\

\end{eqnarray}""")
```

Out[2]:

\begin{eqnarray} \hat{y}_{i} = x_{i} \times \frac{\sum_{i'=1}^{n}\left (x_{i'} y_{i'} \right)}{\sum_{j=1}^{n} x_{j}^{2}} \\ \hat{y}_{i} = \sum_{i'=1}^{n} \frac{\left (x_{i'} y_{i'} \right) \times x_{i}}{\sum_{j=1}^{n} x_{j}^{2}} \\ \hat{y}_{i} = \sum_{i'=1}^{n} \left (\frac{ x_{i} x_{i'} } { \sum_{j=1}^{n} x_{j}^{2} } \times y_{i'} \right) \\ a_{i'} = \frac{ x_{i} x_{i'} } { \sum_{j=1}^{n} x_{j}^{2} } \\ \end{eqnarray}

Exercise 3.6

In [3]:

```
from IPython.display import Latex

Latex(r"""\begin{eqnarray}
Using \; equation \;(3.4), when \; x_{i}=\bar{x}, \\
then \; \hat{\beta_{1}}=0 \; and \; \hat{\beta_{0}}=\bar{y} \\
and \; the \; equation \; for\; \hat{y_{i}}\; evaluates\; to \; equal \; \bar{y}
\end{eqnarray}""")
```

Out[3]:

\begin{eqnarray} Using \; equation \;(3.4), when \; x_{i}=\bar{x}, \\ then \; \hat{\beta_{1}}=0 \; and \; \hat{\beta_{0}}=\bar{y} \\ and \; the \; equation \; for\; \hat{y_{i}}\; evaluates\; to \; equal \; \bar{y} \end{eqnarray}

Exercise 3.11

In [4]:

```
# part (a)
import numpy as np

np.random.seed(11)

s = np.random.normal(0,1,100)
x = s
d = np.random.normal(0,1,100)

y = 2*x + d

import statsmodels.api as sm

model1 = sm.OLS(y,x).fit()

model1.summary()
```

Out[4]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.790
Model:	OLS	**Adj. R-squared:**	0.787
Method:	Least Squares	**F-statistic:**	371.4
Date:	Thu, 31 Jan 2019	**Prob (F-statistic):**	2.82e-35
Time:	14:16:24	**Log-Likelihood:**	-136.94
No. Observations:	100	**AIC:**	275.9
Df Residuals:	99	**BIC:**	278.5
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	1.9754	0.103	19.272	0.000	1.772	2.179

Omnibus:	0.555	**Durbin-Watson:**	1.775
Prob(Omnibus):	0.758	**Jarque-Bera (JB):**	0.687
Skew:	0.077	**Prob(JB):**	0.709
Kurtosis:	2.624	**Cond. No.**	1.00

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

The estimated coefficient 2.04 means with one unit increase in x, y will increase 2.04 units. Standard error rate shows the one standard deviation of beta_x is nearly 0. So the 95% confidence interval of beta_x is [1.872,2.211] P-value close to 0 shows that x is statistically significant.

In [5]:

```
# part (b)

model2 = sm.OLS(x,y).fit()

model2.summary()
```

Out[5]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.790
Model:	OLS	**Adj. R-squared:**	0.787
Method:	Least Squares	**F-statistic:**	371.4
Date:	Thu, 31 Jan 2019	**Prob (F-statistic):**	2.82e-35
Time:	14:16:24	**Log-Likelihood:**	-57.048
No. Observations:	100	**AIC:**	116.1
Df Residuals:	99	**BIC:**	118.7
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	0.3997	0.021	19.272	0.000	0.359	0.441

Omnibus:	0.720	**Durbin-Watson:**	2.036
Prob(Omnibus):	0.698	**Jarque-Bera (JB):**	0.823
Skew:	-0.105	**Prob(JB):**	0.663
Kurtosis:	2.608	**Cond. No.**	1.00

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

The estimated coefficient 0.4175 means with one unit increase in y, x will increase nearly 0.4175. Standard error rate shows the one standard deviation of beta_y is 3.08e^-8. So the 95% confidence interval of beta_y is [0.383, 0.452] P-value close to 0 shows that y is statistically significant.

In [6]:

```
# part (c)
from IPython.display import Latex
Latex(r"""\begin{eqnarray}
\hat {x} = \hat{\beta_{x}} \times y \; versus \; \hat {y} = \hat{\beta_{y}} \times x, the \; betas
\; should \; be \; inverse \; of \; each \; other \; (\hat{\beta_{x}}=\frac{1}{\hat{\beta_{y}}})
\; but \; they \; are \; somewhat \; off

\end{eqnarray}""")
```

Out[6]:

\begin{eqnarray} \hat {x} = \hat{\beta_{x}} \times y \; versus \; \hat {y} = \hat{\beta_{y}} \times x, the \; betas \; should \; be \; inverse \; of \; each \; other \; (\hat{\beta_{x}}=\frac{1}{\hat{\beta_{y}}}) \; but \; they \; are \; somewhat \; off \end{eqnarray}

part(e)

I used t- statistic formula from (d) and plug both (x,y) and (y,x) in it. I got same result.

In conclusion, the two regression lines should be the same just with the axes switched.

In [7]:

```
# part(f)

X = sm.add_constant(x)

model3 = sm.OLS(y,X).fit()

model3.summary()
```

Out[7]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.790

Dep. Variable:	y	**R-squared:**	0.790
Model:	OLS	**Adj. R-squared:**	0.787
Method:	Least Squares	**F-statistic:**	367.6
Date:	Thu, 31 Jan 2019	**Prob (F-statistic):**	6.22e-35
Time:	14:16:24	**Log-Likelihood:**	-136.94
No. Observations:	100	**AIC:**	277.9
Df Residuals:	98	**BIC:**	283.1
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	0.0042	0.096	0.044	0.965	-0.187	0.195
x1	1.9753	0.103	19.172	0.000	1.771	2.180

Omnibus:	0.555	**Durbin-Watson:**	1.775
Prob(Omnibus):	0.758	**Jarque-Bera (JB):**	0.687
Skew:	0.077	**Prob(JB):**	0.709
Kurtosis:	2.624	**Cond. No.**	1.07

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

In [8]:

```
Y = sm.add_constant(y)
model4 = sm.OLS(x,Y).fit()

model4.summary()
```

Out[8]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.790
Model:	OLS	**Adj. R-squared:**	0.787
Method:	Least Squares	**F-statistic:**	367.6
Date:	Thu, 31 Jan 2019	**Prob (F-statistic):**	6.22e-35
Time:	14:16:24	**Log-Likelihood:**	-57.048
No. Observations:	100	**AIC:**	118.1
Df Residuals:	98	**BIC:**	123.3
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	0.0007	0.043	0.017	0.986	-0.085	0.087
x1	0.3997	0.021	19.172	0.000	0.358	0.441

Omnibus:	0.720	**Durbin-Watson:**	2.036
Prob(Omnibus):	0.698	**Jarque-Bera (JB):**	0.823
Skew:	-0.105	**Prob(JB):**	0.663
Kurtosis:	2.608	**Cond. No.**	2.07

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

In conclusion, the t- statistic of two linear regression line are same.

exercise 3.12

In [9]:

```
# part(a)

from IPython.display import Latex
Latex(r"""\begin{eqnarray}
When \; x_{i}=y_{i}, or\; more\; generally\; when\; the \;beta \;denominators\; are\; equal\; \sum
x_{i}^2=\sum y_{i}^2

\end{eqnarray}""")
```

Out[9]:

\begin{eqnarray} When \; x_{i}=y_{i}, or\; more\; generally\; when\; the \;beta \;denominators\; are\; equal\; \sum x_{i}^2=\sum y_{i}^2 \end{eqnarray}

In [10]:

```
# part(b)

import pandas as pd

print("Model1's coefficient for Beta1 : ",model1.params)

print("Model2's coefficient for Beta2 : ",model2.params)
```

```
Model1's coefficient for Beta1 :  [1.97540554]
Model2's coefficient for Beta2 :  [0.39968647]
```

In [11]:

```
# part(c)
np.random.seed(111)

u = np.random.normal(1000,0.1,100)
v = np.random.normal(1000,0.1,100)


model5 = sm.OLS(v,u).fit()
model6 = sm.OLS(u,v).fit()

print("Model5's coefficient for Beta1 : ",model5.params)

print("Model6's coefficient for Beta2 : ",model6.params)
```

```
Model5's coefficient for Beta1 :  [1.00000464]
Model6's coefficient for Beta2 :  [0.99999534]
```

Excercise 3.13

In [12]:

```
#part (a)
np.random.seed(1111)

X1 = np.random.normal(0,1,100)
```

In [13]:

```
#part(b)
```

```
eps = np.random.normal(0,0.25,100)
```

In [14]:

```
#part(c)
Y1 = -1 + 0.5*X1 + eps

print(len(Y1))

Latex(r"""\begin{eqnarray}
\beta_{0}=-1 \\
\beta_{1}=0.5
\end{eqnarray}""")
```

```
100
```

Out[14]:

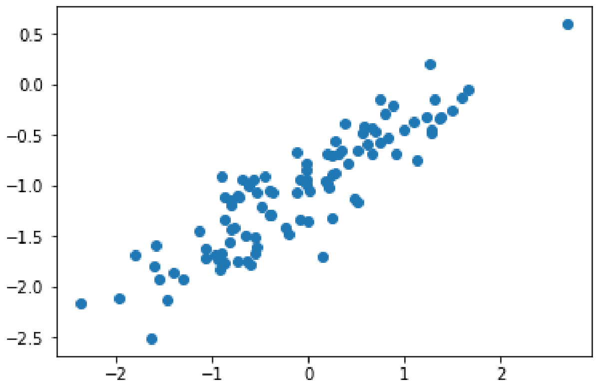
\begin{eqnarray} \beta_{0}=-1 \\ \beta_{1}=0.5 \end{eqnarray}

In [15]:

```
#part(d)
import matplotlib.pyplot as plt

plt.scatter(X1,Y1)
```

Out[15]:

```
<matplotlib.collections.PathCollection at 0x18136ff0cf8>
```



X and Y are nearly positively relative.

In [16]:

```
#part(e)
X2= sm.add_constant(X)

model8 = sm.OLS(Y1,X2).fit()

print(model8.params)

Latex(r"""\begin{eqnarray}
\beta_{0}= -0.75049222 \\
\beta_{1}= 0.44058306
\end{eqnarray}""")
```

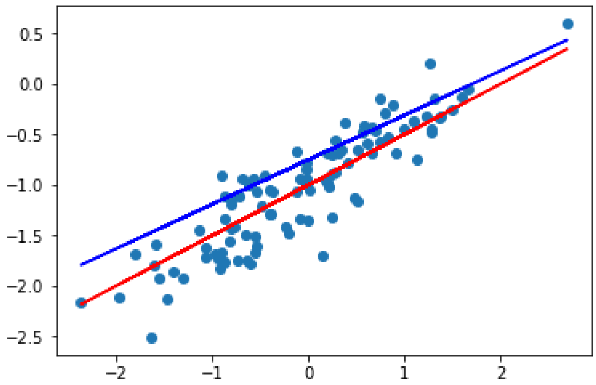
```
[-1.04409137  0.01356952]
```

Out[16]:

\begin{eqnarray} \beta_{0}= -0.75049222 \\ \beta_{1}= 0.44058306 \end{eqnarray}

In [17]:

```
#part(f)
plt.scatter(X1,Y1)
plt.plot(X1,-1 + 0.5*X1,'r')
plt.plot(X1, -0.75049222 + 0.44058306*X1, 'b' )

plt.show()
```



In [18]:

```
#part(g)
import statsmodels.formula.api as smf

data = {"Y1": Y1, "X1": X1}

model9 = smf.ols(formula = 'Y1 ~ np.power(X1,2) + X1', data = data).fit()

print(model9.params)

table = sm.stats.anova_lm(model9,type = 2)

print(table)
```

```
Intercept          -0.996210
np.power(X1, 2)     0.005290
X1                  0.564663
dtype: float64
                   df     sum_sq    mean_sq           F        PR(>F)
np.power(X1, 2)   1.0   0.009219   0.009219    0.134242  7.148715e-01
X1                1.0  27.351537  27.351537  398.277387  4.092429e-36
Residual         97.0   6.661435   0.068675         NaN           NaN
```

from sklearn.preprocessing import PolynomialFeatures from sklearn.linear_model import LinearRegression

poly_reg = PolynomialFeatures(2) xpoly = poly_reg.fit_transform(X1)

print(xpoly)

linearreg_2 = LinearRegression() linearreg_2.fit(xpoly,Y1)

plt.scatter(X1,Y1) plt.plot(X1,-1 + 0.5*X1,'r') plt.plot(X1, -0.75049222 + 0.44058306*X1, 'b') plt.plot(X1, linearreg_2.predict(xpoly))

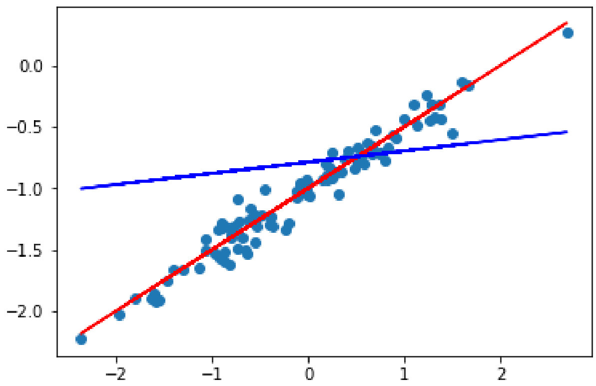
plt.show()

In [19]:

```
#part(h)

np.random.seed(11111)

eps2 = np.random.normal(0,0.1,100)

Y2 = -1 + 0.5*X1 + eps2

model10 = sm.OLS(Y2,X2).fit()

print(model10.params)

plt.scatter(X1,Y2)
```

```
plt.plot(X1,-1 + 0.5*X1,'r')
plt.plot(X1, -0.7887987 + 0.09120311*X1, 'b' )
plt.show()
```

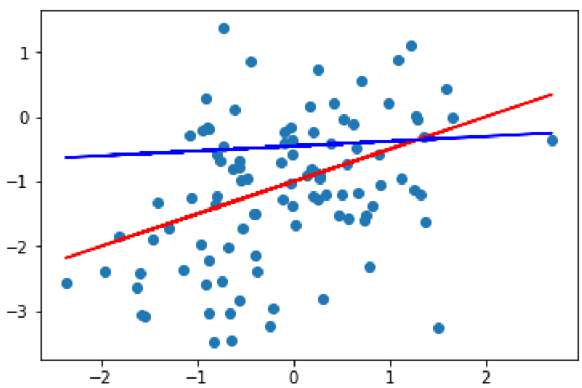
```
[-1.05679919  0.00845146]
```



In [20]:

```
# part(i)
np.random.seed(11111)

eps3 = np.random.normal(0,1,100)

Y3 = -1 + 0.5*X1 + eps3

model11 = sm.OLS(Y3,X2).fit()

print(model11.params)

plt.scatter(X1,Y3)
plt.plot(X1,-1 + 0.5*X1,'r')
plt.plot(X1, -0.45189999 + 0.07522537*X1, 'b' )
plt.show()
```

```
[-1.15037271  0.02118244]
```



part(j)

In [21]:

```
model8.summary()
```

Out[21]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.000
Model:	OLS	**Adj. R-squared:**	-0.010
Method:	Least Squares	**F-statistic:**	0.04620
[illegible]	Thu, 31 Jan	**Prob (F-**	[illegible]

Date:	Thu, 31 Jan 2019	**Prob (F-statistic):**	0.830
Time:	14:16:25	**Log-Likelihood:**	-87.962
No. Observations:	100	**AIC:**	179.9
Df Residuals:	98	**BIC:**	185.1
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-1.0441	0.059	-17.723	0.000	-1.161	-0.927
x1	0.0136	0.063	0.215	0.830	-0.112	0.139

Omnibus:	0.419	**Durbin-Watson:**	1.794
Prob(Omnibus):	0.811	**Jarque-Bera (JB):**	0.572
Skew:	0.027	**Prob(JB):**	0.751
Kurtosis:	2.633	**Cond. No.**	1.07

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

In [22]:

```
model10.summary()
```

Out[22]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.000
Model:	OLS	**Adj. R-squared:**	-0.010
Method:	Least Squares	**F-statistic:**	0.02733
Date:	Thu, 31 Jan 2019	**Prob (F-statistic):**	0.869
Time:	14:16:25	**Log-Likelihood:**	-66.854
No. Observations:	100	**AIC:**	137.7
Df Residuals:	98	**BIC:**	142.9
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-1.0568	0.048	-22.155	0.000	-1.151	-0.962
x1	0.0085	0.051	0.165	0.869	-0.093	0.110

Omnibus:	0.074	**Durbin-Watson:**	1.959
Prob(Omnibus):	0.964	**Jarque-Bera (JB):**	0.235
Skew:	0.029	**Prob(JB):**	0.889
Kurtosis:	2.770	**Cond. No.**	1.07

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

In [23]:

```
model11.summary()
```

Out[23]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.000
Model:	OLS	**Adj. R-squared:**	-0.010
Method:	Least Squares	**F-statistic:**	0.03133
Date:	Thu, 31 Jan 2019	**Prob (F-statistic):**	0.860
Time:	14:16:25	**Log-Likelihood:**	-151.91
No. Observations:	100	**AIC:**	307.8
Df Residuals:	98	**BIC:**	313.0
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-1.1504	0.112	-10.301	0.000	-1.372	-0.929
x1	0.0212	0.120	0.177	0.860	-0.216	0.259

Omnibus:	2.211	**Durbin-Watson:**	1.959
Prob(Omnibus):	0.331	**Jarque-Bera (JB):**	1.785
Skew:	-0.176	**Prob(JB):**	0.410
Kurtosis:	2.448	**Cond. No.**	1.07

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

In [24]:

```
from matplotlib import pyplot as plt
from pandas.tools.plotting import scatter_matrix
from mpl_toolkits.mplot3d import Axes3D
from statsmodels.stats.outliers_influence import OLSInfluence

%matplotlib inline
plt.style.use('ggplot')
```

Exercise 3.14

In [25]:

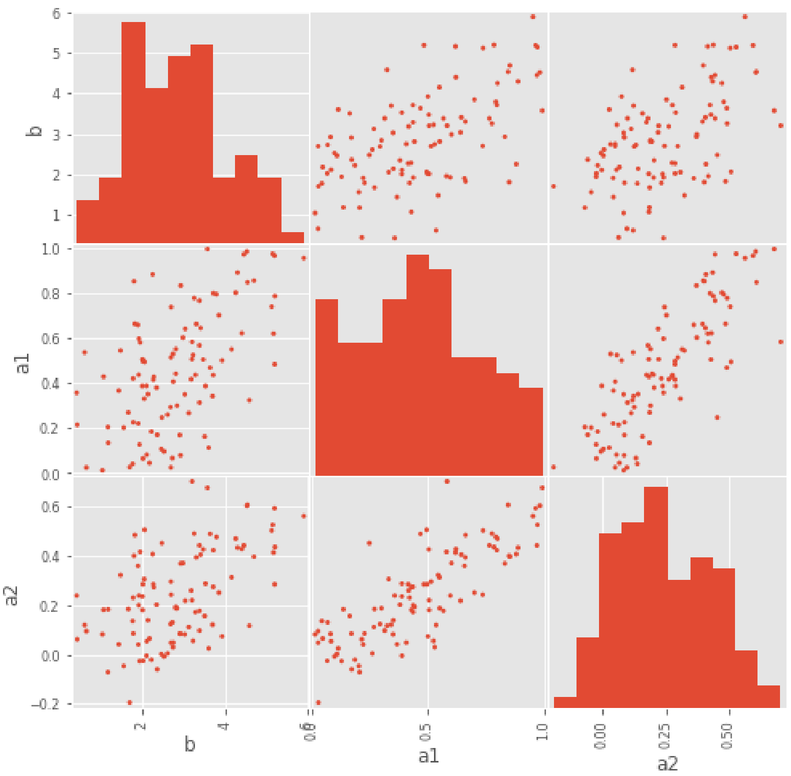
```
#part(a)

np.random.seed(2)

a1 = np.random.random(100)

a2 = 0.5*a1 + np.random.randn(100)/10

b =2+2* a1 +0.3* a2 + np.random.randn(100)

data2 = {"b":b , "a1": a1, "a2": a2}
```

In [26]:

```
#part(b)
from numpy import corrcoef
print(corrcoef(a1,a2))

df = pd.DataFrame(np.column_stack((b,a1,a2)), columns=['b','a1','a2'] )
df.head()


scatter_matrix(df, figsize = (8,8),alpha=1);
```

```
[[1.         0.81145744]
 [0.81145744 1.        ]]
```

```
C:\Users\husiw\Anaconda3\lib\site-packages\ipykernel_launcher.py:9: FutureWarning: 'pandas.tools.plotting.scatter_matrix' is deprecated, import 'pandas.plotting.scatter_matrix' instead.
  if __name__ == '__main__':
```



In [27]:

```python
#part(c)
model12 = smf.ols(formula = 'b ~ a1 + a2', data = data2).fit()

print(model12.summary())

print(model12.params)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      b   R-squared:                       0.335
Model:                            OLS   Adj. R-squared:                  0.322
Method:                 Least Squares   F-statistic:                     24.47
Date:                Thu, 31 Jan 2019   Prob (F-statistic):           2.48e-09
Time:                        14:16:26   Log-Likelihood:                -137.23
No. Observations:                 100   AIC:                             280.5
Df Residuals:                      97   BIC:                             288.3
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      1.7199      0.195      8.817      0.000       1.333       2.107
a1             1.8561      0.624      2.975      0.004       0.618       3.095
a2             1.1244      0.874      1.287      0.201      -0.610       2.859
==============================================================================
Omnibus:                        0.358   Durbin-Watson:                   1.843
Prob(Omnibus):                  0.836   Jarque-Bera (JB):                0.501
Skew:                          -0.120   Prob(JB):                        0.778
Kurtosis:                       2.749   Cond. No.                         12.1
==============================================================================

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

```
Intercept    1.719888
a1           1.856144
a2           1.124409
dtype: float64
```

The intercept $\beta_0 = 1.72$ is significant and β_1 is barely significant so we reject the hypothesis that $\beta_0=0$ and the hypothesis that $\beta_1=0$ but we can't reject the hypothesis that $\beta_2 = 0$. Also notice the SE and confidence intervals for all three coeffecients are verly large.

In [28]:

```
# part(c)-(e)
model13 = smf.ols(formula = 'b ~ a1', data = data2).fit()

print(model13.summary())

model14 = smf.ols(formula = 'b ~ a2', data = data2).fit()

print(model14.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      b   R-squared:                       0.324
Model:                            OLS   Adj. R-squared:                  0.317
Method:                 Least Squares   F-statistic:                     46.98
Date:                Thu, 31 Jan 2019   Prob (F-statistic):           6.43e-10
Time:                        14:16:26   Log-Likelihood:                -138.08
No. Observations:                 100   AIC:                             280.2
Df Residuals:                      98   BIC:                             285.4
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      1.6908      0.194      8.698      0.000       1.305       2.077
a1             2.5078      0.366      6.854      0.000       1.782       3.234
==============================================================================
Omnibus:                        0.247   Durbin-Watson:                   1.868
Prob(Omnibus):                  0.884   Jarque-Bera (JB):                0.341
Skew:                          -0.112   Prob(JB):                        0.843
Kurtosis:                       2.823   Cond. No.                         4.61
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
                            OLS Regression Results
==============================================================================
Dep. Variable:                      b   R-squared:                       0.275
Model:                            OLS   Adj. R-squared:                  0.267
Method:                 Least Squares   F-statistic:                     37.12
Date:                Thu, 31 Jan 2019   Prob (F-statistic):           2.18e-08
Time:                        14:16:26   Log-Likelihood:                -141.60
No. Observations:                 100   AIC:                             287.2
Df Residuals:                      98   BIC:                             292.4
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      2.0661      0.163     12.702      0.000       1.743       2.389
a2             3.2334      0.531      6.093      0.000       2.180       4.286
==============================================================================
Omnibus:                        1.173   Durbin-Watson:                   1.787
Prob(Omnibus):                  0.556   Jarque-Bera (JB):                1.095
Skew:                          -0.093   Prob(JB):                        0.578
Kurtosis:                       2.522   Cond. No.                         5.58
==============================================================================

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

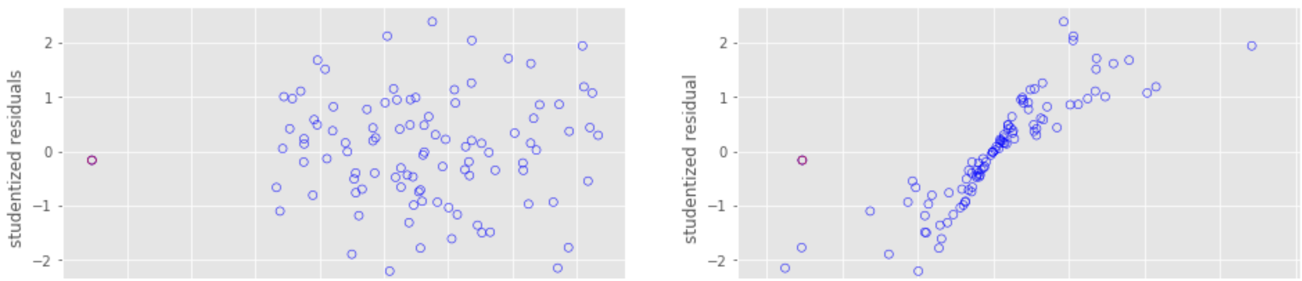
part(f)

In model13, $b \sim a_1$, we find that the coeffecient is close to the true value and is now very significant. In model14 $b \sim a_2$, we find that the coeffecient for a_2 is now very significant. Note there is no contradiction here. a_1 and a_2 are strongly correlated and each is related to b independent of each other.

In [31]:

```
df.loc[len(df)] = [0.1, 0.8, 6]

A = sm.add_constant(df[['a1','a2']])
model = sm.OLS(df.b, A)
estimate = model.fit()

print(estimate.summary())

# Obtain the residuals, studentized residuals and the leverages
fitted_values = estimate.fittedvalues.values
residuals = estimate.resid.values
studentized_residuals = OLSInfluence(estimate).resid_studentized_internal
leverages = OLSInfluence(estimate).influence

# Plot
fig, (ax1,ax2) = plt.subplots(1,2,figsize=(16,4))

# Studentized Residuals
ax1.scatter(fitted_values[:-1], studentized_residuals[:-1], facecolors='none', edgecolors='b');
# Plot the possible Outlier in red
ax1.scatter(fitted_values[-1], studentized_residuals[-1], facecolors='none', edgecolors='r');
ax1.set_xlabel('fitted values');
ax1.set_ylabel('studentized residuals');

# Leverages
ax2.scatter(leverages[:-1], studentized_residuals[:-1], facecolors='none', edgecolors='b');
# plot the possible high leverager in red
ax2.scatter(leverages[-1], studentized_residuals[-1], facecolors='none', edgecolors='r');
ax2.set_xlabel('Leverage');
ax2.set_ylabel('studentized residual');
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      b   R-squared:                       0.403
Model:                            OLS   Adj. R-squared:                  0.391
Method:                 Least Squares   F-statistic:                     33.77
Date:                Thu, 31 Jan 2019   Prob (F-statistic):           6.22e-12
Time:                        14:16:26   Log-Likelihood:                -141.94
No. Observations:                 103   AIC:                             289.9
Df Residuals:                     100   BIC:                             297.8
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          1.6794      0.195      8.611      0.000       1.292       2.066
a1             2.8495      0.384      7.423      0.000       2.088       3.611
a2            -0.6217      0.104     -5.961      0.000      -0.829      -0.415
==============================================================================
Omnibus:                        0.198   Durbin-Watson:                   1.908
Prob(Omnibus):                  0.906   Jarque-Bera (JB):                0.196
Skew:                          -0.098   Prob(JB):                        0.907
Kurtosis:                       2.913   Cond. No.                         5.70
==============================================================================

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




0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
fitted values
−0.075
−0.050
−0.025
0.000
0.025
0.050
0.075
0.100
Leverage