109 HW2

October 9, 2018

0.1 1 For each of the three data sets plotted above (I, II and III), answer the following:

a. Does the data show a positive or negative correlation between x and y? plot 1: no correlation between x and y, y value is random around 20 plot 2:positive correlation plot 3:negative correlation

b. Which function (equation) best describes each data set? In these equations, represents random noise, with mean value = 0 and standard deviation = 1 .

c. f(x) = 1 + 2x +

ii. f(x) = 20 +

iii. f(x) = 20 x +

```
i describes plot 2
ii decribes plot 1
iii decribes plot 3
```

c. Which regression table corresponds to each plot?

i matches plot 2 ii matches plot 3 ii match plot 1

0.2 2. ISLR chapter 3, problem 3 (page 120)

Y = 50 + 20(gpa) + 0.07(iq) + 35(gender) + 0.01(gpa * iq) - 10 (gpa * gender)

```
a. iii is true
   Because 35*gender - 10 * gpa * gender are the only factor that is related to gender tha
   Given a high gpa over 3.5., (35*gender - 10 * gpa * gender) is larger than 0.
   So For a fixed value of IQ and GPA, males earn more on average than females provided tha

b.Predict the salary of a female with IQ of 110 and a GPA of 4.0.
= 50 + 20(4) + 0.07(110) + 35(1) + 0.01(4 * 110) - 10 (4 * 1)
= 137.1

c. False. we still need to consider the p-value of the regression coefficient.
```

0.3 3. ISLR chapter 3, problem 4 (pages 120-121)

```
a.  Since we know that the relationship between X and Y is linear, we can asusume the least sq
    Therefore, the training RSS for the linear regression should be lower than the one for the

b. If there is overfitting in training, then the test RRS should be higher due to the divergen
So in testing,the test RRS of cubic regression should be higher than the test RRS of the linea

c. Cubic regression has lower train RSS than the linear regression because a more complex mode

d. In this case, the info is not enough to predic which test RSS would be lower because we do
```

1 4. In this problem, we will simulate a dataset and use multiple linear regression to investigate it.

Imagine we conduct a survey of N=100 students and ask them how much time per week they spend on work ( ) and how much time on play ( ). We also ask them about their overall level x1 x2 of satisfaction (y), which we take to be the outcome. Download the dataset HW2.csv from the course website, which contains these data.

```
In [9]: import urllib
        import matplotlib.pyplot as plt
        import pandas as pd
        import numpy as np

        #read data
        broken_df = pd.read_csv('/Users/xuzhaokai/Desktop/109 HW2/HW2.csv')
        dataFrame = broken_df.values
        print dataFrame.shape
```

```
(100, 3)
```

```
In [10]: # 4. a Make a scatter plot showing y vs. x1 .
         # Comment on the relationship between these variables:
         # dothey appear correlated (positively or negatively)?
         # Is their relationship linear or non-linear?

         import matplotlib.pyplot as plt
         x1 = dataFrame[:,0]
         x2 = dataFrame[:,1]
         y = dataFrame[:,2]

         plt.plot(x1, y, 'ro')
         plt.xlabel("x1: time per week spend on work  ")
         plt.ylabel("y: overal level of satisfaction ")
         plt.title (" work VS satisfaction ")
         plt.show()
```

```
print ("the relationship between x1 and y :")
print ("x1 and y are positively correlated and they are in non-linear relationship")
```


```
the relationship between x1 and y :
x1 and y are positively correlated and they are in non-linear relationship
```

```
In [12]: #4. b  Fit a simple linear regression of y vs. x1
         # Report the estimated intercept and slope, and make a plot showing
         # the data points together with the regression line.
         # Is there a statistically significant effect of x1 on y ?

         # sm is better for sklearn to analyze data
         from sklearn import linear_model

         reg = linear_model.LinearRegression()

         # array to matrix
         x1= x1.reshape(-1, 1)
         y = y.reshape(-1,1)

         # train the model
         reg.fit (x1, y)
```

```
#obtain slope and intercept
slope = reg.coef_[0][0]
intercept = reg.intercept_

print "slope: ",round(slope, 5)
print "intercept: ",round(intercept, 5)
```

```
slope:  0.07105
intercept:  0.06039
```

In [13]:
```
# plot the data
plt.xlabel("x1: time per week spend on work  ")
plt.ylabel("y: overal level of satisfaction ")
plt.title (" work VS satisfaction ")
plt.plot(x1, y, '.')
plt.plot(x1, slope * x1 + intercept, '-')
plt.text(5,4,"y = 0.07105 * x1 + 0.06039")
plt.show()
```

In [19]:
```
#4 b Is there a statistically significant effect of x1 on y ?
print "I do not thnk their is a statistically significant effect of x1 on y because \
the p value is zero accourding to the below table, which means the null hypothesis is
```

I do not thnk their is a statistically significant effect of x1 on y because the p value is ze

In [21]:
```
# 4.c
import numpy as np
import statsmodels.api as sm

X_1=sm.add_constant(x1)
model = sm.OLS(y,X_1)
results = model.fit()
print results.summary()
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.263
Model:                            OLS   Adj. R-squared:                  0.255
Method:                 Least Squares   F-statistic:                     34.90
Date:                Tue, 09 Oct 2018   Prob (F-statistic):           5.04e-08
Time:                        17:15:12   Log-Likelihood:                -176.08
No. Observations:                 100   AIC:                             356.2
Df Residuals:                      98   BIC:                             361.4
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0604      0.291      0.207      0.836      -0.517       0.638
x1             0.0711      0.012      5.907      0.000       0.047       0.095
==============================================================================
Omnibus:                        1.420   Durbin-Watson:                   2.256
Prob(Omnibus):                  0.492   Jarque-Bera (JB):                0.913
Skew:                          -0.025   Prob(JB):                        0.634
Kurtosis:                       3.465   Cond. No.                         49.6
==============================================================================

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

In [23]:
```
# 4.c What is the 95% confidence interval for the slope of x1 ?
print "\n the confidence interval for the slope of x1 is",results.conf_int(alpha=0.05
```

```
 the confidence interval for the slope of x1 is [[-0.51732074  0.63810476]
 [ 0.0471836   0.09492252]]
```

In [27]:
```
# 4. d
# fit a multiple linear regression with x1 and x2 as independent variables.
```

```python
#Report a table with the regression results (similar to Table 3.9 on page 88 in ISLR)

# THE table contains: coeffecent /  std error  //  t-statistic    /  p-value
from sklearn import linear_model
import numpy as np
import statsmodels.api as sm

#stack x1 and x2 to a 2D matrix
x2 = x2.reshape(-1,1)
xData = np.vstack((x1,x2))
xData = xData.reshape(-1,2)
X_1=sm.add_constant(xData)

# train the model
model2 = sm.OLS(y,X_1)
results2 = model2.fit()
print results2.summary()
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.000
Model:                            OLS   Adj. R-squared:                 -0.020
Method:                 Least Squares   F-statistic:                   0.01086
Date:                Tue, 09 Oct 2018   Prob (F-statistic):              0.989
Time:                        17:17:35   Log-Likelihood:                -191.30
No. Observations:                 100   AIC:                             388.6
Df Residuals:                      97   BIC:                             396.4
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          1.6114      0.432      3.726      0.000       0.753       2.470
x1            -0.0006      0.015     -0.044      0.965      -0.030       0.029
x2            -0.0019      0.014     -0.140      0.889      -0.029       0.025
==============================================================================
Omnibus:                        5.551   Durbin-Watson:                   1.993
Prob(Omnibus):                  0.062   Jarque-Bera (JB):                5.045
Skew:                           0.432   Prob(JB):                       0.0803
Kurtosis:                       3.682   Cond. No.                         79.4
==============================================================================

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

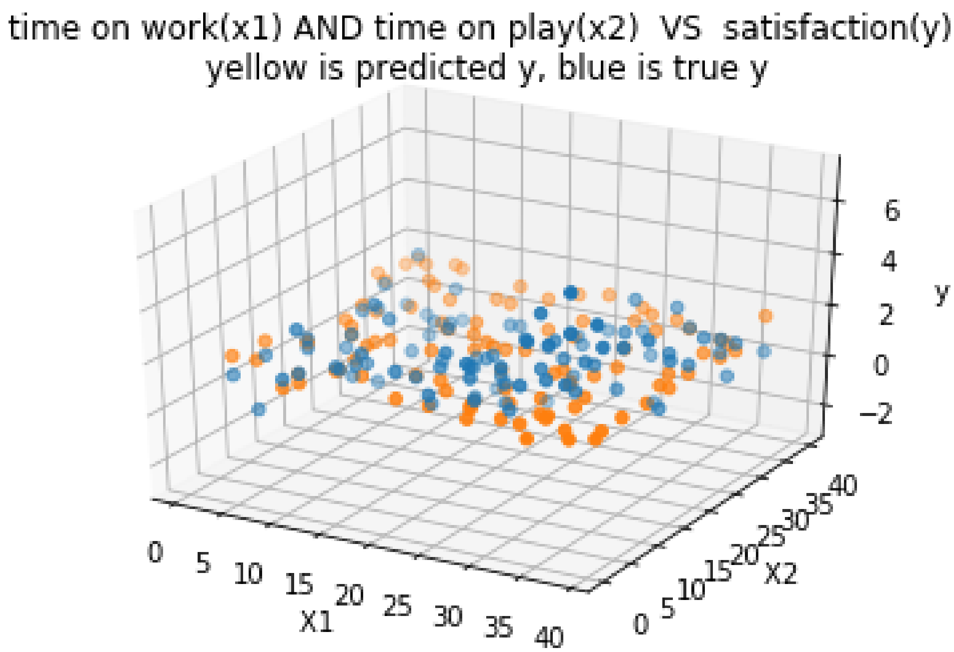
In [28]:
```python
# Which parameters have a statistically significant effect?
print "x2 has a more statistically significant effect"
```

x2 has a more statistically significant effect

```
In [29]: #  4. e Make a scatter plot showing y vs. y, the predicted value of y
         from mpl_toolkits.mplot3d import Axes3D
         import matplotlib.pyplot as plt

         # plot all points
         fig = plt.figure()
         ax = fig.add_subplot(111, projection='3d')

         ax.set_title(' time on work(x1) AND time on play(x2)  VS  satisfaction(y)\
          \n yellow is predicted y, blue is true y')
         ax.set_xlabel('X1 ')
         ax.set_ylabel('X2 ')
         ax.set_zlabel('y ')
         #scatter plot showing y
         ax.scatter(x1, x2, y)
         #surface plot showing y
         y_predict = 1.6114 * X_1[:,0] +  (-0.0006) * X_1[:,1] + (-0.0019)* X_1[:,2]
         ax.scatter(x1, x2, y_predict)
         plt.show()
```




```
In [134]: # 4. f   Create a categorical variable with 3 levels called WorkType,
          # where WorkType=Idle for x1 < 10,
          # WorkType=Diligent for 10  x1 < 30 ,
```

```
# and WorkType=Workaholic for x1 30 .
# Fit a linear regression of y against WorkType and x2 , and report the regression t
import numpy as np

#break into three types
Idle = []
Diligent = []
Workaholic = []

for row in range(dataFrame.shape[0]):
    data = dataFrame[row,0]
    if (data <10):
        Idle.append(dataFrame[row,:])
    elif data <=30 and data >=10:
        Diligent.append(dataFrame[row,:])
    else:
        Workaholic.append(dataFrame[row,:])

print len(Idle), len(Diligent), len(Workaholic)
```

```
22 46 32
```

```
In [138]: import statsmodels.api as sm
          #tpye1 = Idle
          Idle = np.asarray(Idle)
          Diligent = np.asarray(Diligent)
          Workaholic = np.asarray(Workaholic)

          type1 =sm.add_constant(Idle)
          type2 =sm.add_constant(Diligent)
          type3 =sm.add_constant(Workaholic)
```

```
In [139]: # train the model
          print "linear regression model of Type 1"
          model_1 = sm.OLS(type1[:,3], type1[:,0:3])
          result_1 = model_1.fit()
          print result_1.summary()
```

```
linear regression model of Type 1
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.212
Model:                            OLS   Adj. R-squared:                  0.129
Method:                 Least Squares   F-statistic:                     2.561
Date:                Tue, 09 Oct 2018   Prob (F-statistic):              0.104
Time:                        20:24:14   Log-Likelihood:                -30.015
No. Observations:                  22   AIC:                             66.03
Df Residuals:                      19   BIC:                             69.30
```

```
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          1.4974      0.724      2.067      0.053      -0.019       3.014
x1             0.0027      0.089      0.030      0.976      -0.184       0.190
x2            -0.0398      0.019     -2.131      0.046      -0.079      -0.001
==============================================================================
Omnibus:                        0.062   Durbin-Watson:                   2.199
Prob(Omnibus):                  0.969   Jarque-Bera (JB):                0.155
Skew:                           0.100   Prob(JB):                        0.926
Kurtosis:                       2.641   Cond. No.                         87.0
==============================================================================

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

```
In [141]: print "linear regression model of Type 2"
          model_2 = sm.OLS(type2[:,3],type2[:,0:3])
          result_2 = model_2.fit()
          print result_2.summary()
```

```
linear regression model of Type 2
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.523
Model:                            OLS   Adj. R-squared:                  0.501
Method:                 Least Squares   F-statistic:                     23.55
Date:                Tue, 09 Oct 2018   Prob (F-statistic):           1.24e-07
Time:                        20:24:25   Log-Likelihood:                -61.357
No. Observations:                  46   AIC:                             128.7
Df Residuals:                      43   BIC:                             134.2
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          1.7291      0.538      3.215      0.002       0.644       2.814
x1             0.0492      0.023      2.128      0.039       0.003       0.096
x2            -0.0817      0.013     -6.385      0.000      -0.107      -0.056
==============================================================================
Omnibus:                        2.289   Durbin-Watson:                   1.870
Prob(Omnibus):                  0.318   Jarque-Bera (JB):                1.372
Skew:                          -0.368   Prob(JB):                        0.504
Kurtosis:                       3.418   Cond. No.                         107.
==============================================================================
```

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

In [142]:
```
print "linear regression model of Type 3"
model_3 = sm.OLS(type3[:,3], type3[:,0:3])
result_3 = model_3.fit()
print result_3.summary()
```

```
linear regression model of Type 3
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.652
Model:                            OLS   Adj. R-squared:                  0.628
Method:                 Least Squares   F-statistic:                     27.17
Date:                Tue, 09 Oct 2018   Prob (F-statistic):           2.25e-07
Time:                        20:24:37   Log-Likelihood:                -45.998
No. Observations:                  32   AIC:                             98.00
Df Residuals:                      29   BIC:                             102.4
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          1.6310      2.110      0.773      0.446      -2.685       5.947
x1             0.0913      0.060      1.515      0.141      -0.032       0.215
x2            -0.1225      0.017     -7.245      0.000      -0.157      -0.088
==============================================================================
Omnibus:                        0.042   Durbin-Watson:                   1.909
Prob(Omnibus):                  0.979   Jarque-Bera (JB):                0.223
Skew:                          -0.060   Prob(JB):                        0.894
Kurtosis:                       2.609   Cond. No.                         436.
==============================================================================

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

In [146]:
```
# 4.g  different levels of this categorical variable.
# What is your interpretation of the term corresponding to WorkType=Workaholic?
print "Type 1"
print result_1.params
print "Type 2"
print result_2.params
print "Type 3"
print result_3.params
print "According to the above data, I think when workType == Workaholic\
```

```
          the time on work contributes more to their satisfaction and  \
          the time on play has a negative effect on their satisfaction "
```

```
Type 1
[ 1.49740182  0.00271597 -0.03983865]
Type 2
[ 1.72912567  0.04920683 -0.0816793 ]
Type 3
[ 1.63103149  0.0913222  -0.12248419]
According to the above data, I think when workType == Workaholicthe time on work contributes mo
```