Ethics Pledge

Consistent with the above statements, all homework exercises, tests and exams that are designated as individual assignments MUST contain the following signed statement before they can be accepted for grading.

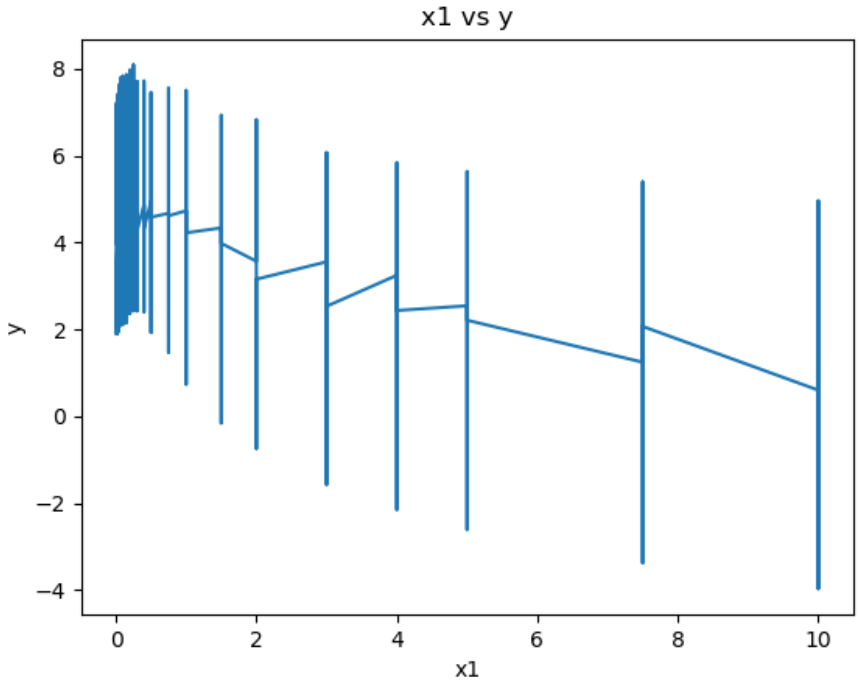
I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination. I further pledge that I have not copied any material from a book, article, the Internet or any other source except where I have expressly cited the source.

Signature: Haodong Zhao Date: Feb 19th 2019

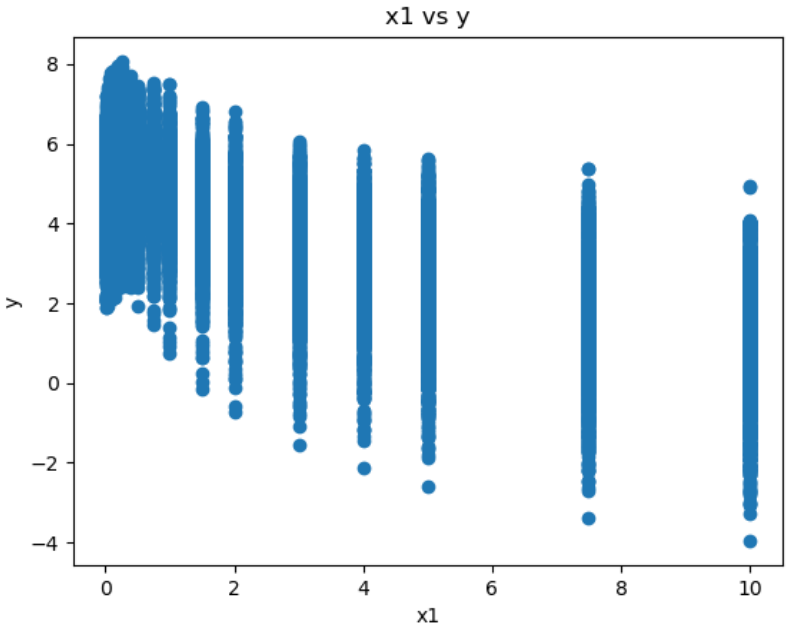
Please note that assignments in this class may be submitted to www.turnitin.com, a web- based anti-plagiarism system, for an evaluation of their originality.

1. Plot x1 vs y:

Following is the figure of x1 vs y

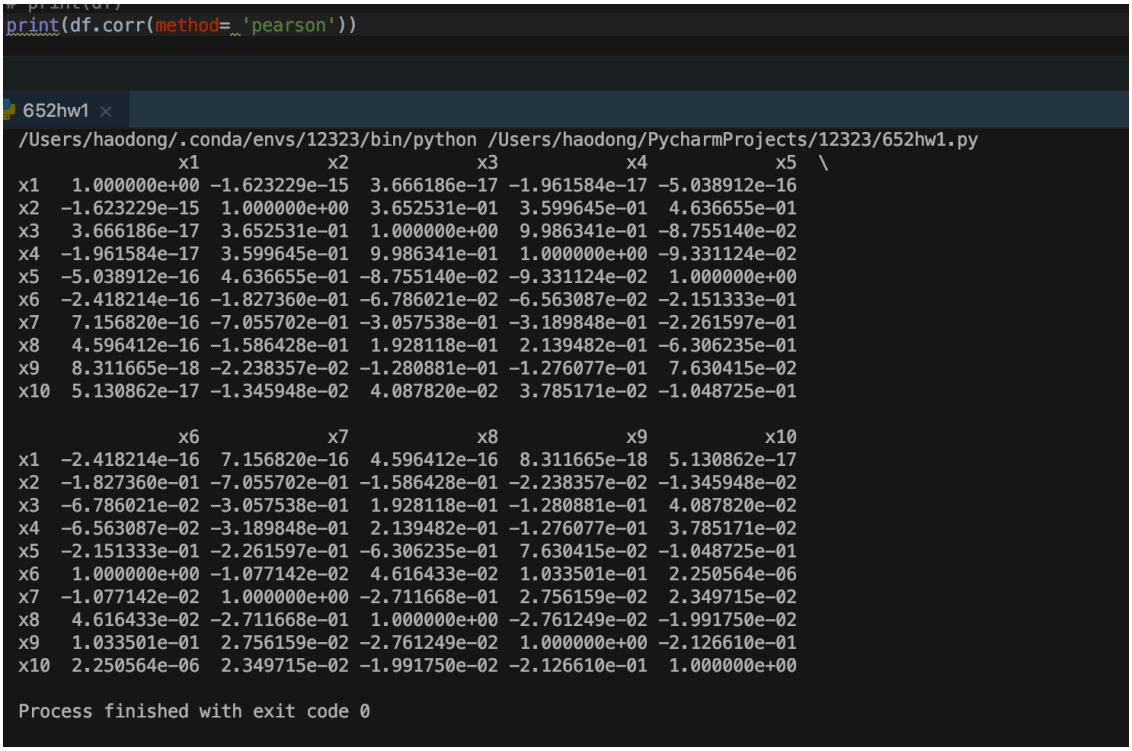


Following is the scatter plot of x1 vs y



2. Present correlation matrix of Xs:

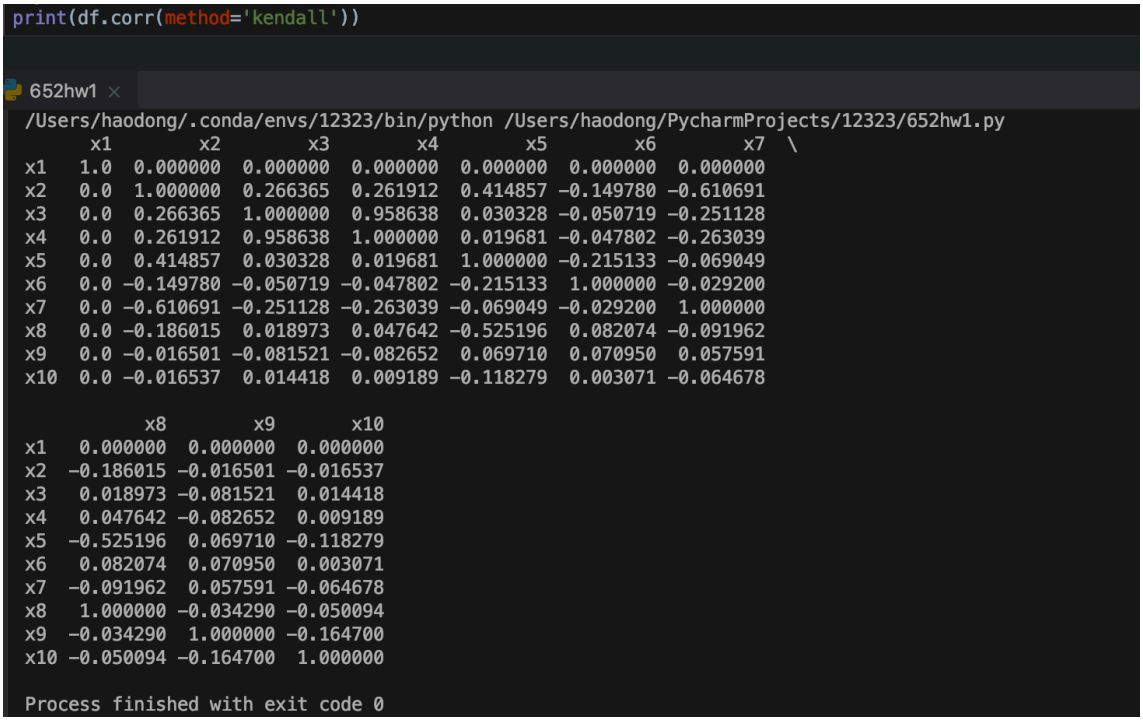
Following is the figure of correlation matrix by using ‘Pearson’ method



```
print(df.corr(method='pearson'))
```

```
652hw1
/Users/haodong/.conda/envs/12323/bin/python /Users/haodong/PycharmProjects/12323/652hw1.py
              x1            x2            x3            x4            x5  \
x1   1.000000e+00 -1.623229e-15  3.666186e-17 -1.961584e-17 -5.038912e-16
x2  -1.623229e-15  1.000000e+00  3.652531e-01  3.599645e-01  4.636655e-01
x3   3.666186e-17  3.652531e-01  1.000000e+00  9.986341e-01 -8.755140e-02
x4  -1.961584e-17  3.599645e-01  9.986341e-01  1.000000e+00 -9.331124e-02
x5  -5.038912e-16  4.636655e-01 -8.755140e-02 -9.331124e-02  1.000000e+00
x6  -2.418214e-16 -1.827360e-01 -6.786021e-02 -6.563087e-02 -2.151333e-01
x7   7.156820e-16 -7.055702e-01 -3.057538e-01 -3.189848e-01 -2.261597e-01
x8   4.596412e-16 -1.586428e-01  1.928118e-01  2.139482e-01 -6.306235e-01
x9   8.311665e-18 -2.238357e-02 -1.280881e-01 -1.276077e-01  7.630415e-02
x10  5.130862e-17 -1.345948e-02  4.087820e-02  3.785171e-02 -1.048725e-01

              x6            x7            x8            x9           x10
x1  -2.418214e-16  7.156820e-16  4.596412e-16  8.311665e-18  5.130862e-17
x2  -1.827360e-01 -7.055702e-01 -1.586428e-01 -2.238357e-02 -1.345948e-02
x3  -6.786021e-02 -3.057538e-01  1.928118e-01 -1.280881e-01  4.087820e-02
x4  -6.563087e-02 -3.189848e-01  2.139482e-01 -1.276077e-01  3.785171e-02
x5  -2.151333e-01 -2.261597e-01 -6.306235e-01  7.630415e-02 -1.048725e-01
x6   1.000000e+00 -1.077142e-02  4.616433e-02  1.033501e-01  2.250564e-06
x7  -1.077142e-02  1.000000e+00 -2.711668e-01  2.756159e-02  2.349715e-02
x8   4.616433e-02 -2.711668e-01  1.000000e+00 -2.761249e-02 -1.991750e-02
x9   1.033501e-01  2.756159e-02 -2.761249e-02  1.000000e+00 -2.126610e-01
x10  2.250564e-06  2.349715e-02 -1.991750e-02 -2.126610e-01  1.000000e+00

Process finished with exit code 0
```

Following is the figure of correlation matrix by using ‘Kendall’ method



```
print(df.corr(method='kendall'))
```

```
652hw1
/Users/haodong/.conda/envs/12323/bin/python /Users/haodong/PycharmProjects/12323/652hw1.py
      x1        x2        x3        x4        x5        x6        x7  \
x1   1.0  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
x2   0.0  1.000000  0.266365  0.261912  0.414857 -0.149780 -0.610691
x3   0.0  0.266365  1.000000  0.958638  0.030328 -0.050719 -0.251128
x4   0.0  0.261912  0.958638  1.000000  0.019681 -0.047802 -0.263039
x5   0.0  0.414857  0.030328  0.019681  1.000000 -0.215133 -0.069049
x6   0.0 -0.149780 -0.050719 -0.047802 -0.215133  1.000000 -0.029200
x7   0.0 -0.610691 -0.251128 -0.263039 -0.069049 -0.029200  1.000000
x8   0.0 -0.186015  0.018973  0.047642 -0.525196  0.082074 -0.091962
x9   0.0 -0.016501 -0.081521 -0.082652  0.069710  0.070950  0.057591
x10  0.0 -0.016537  0.014418  0.009189 -0.118279  0.003071 -0.064678

          x8        x9       x10
x1   0.000000  0.000000  0.000000
x2  -0.186015 -0.016501 -0.016537
x3   0.018973 -0.081521  0.014418
x4   0.047642 -0.082652  0.009189
x5  -0.525196  0.069710 -0.118279
x6   0.082074  0.070950  0.003071
x7  -0.091962  0.057591 -0.064678
x8   1.000000 -0.034290 -0.050094
x9  -0.034290  1.000000 -0.164700
x10 -0.050094 -0.164700  1.000000

Process finished with exit code 0
```

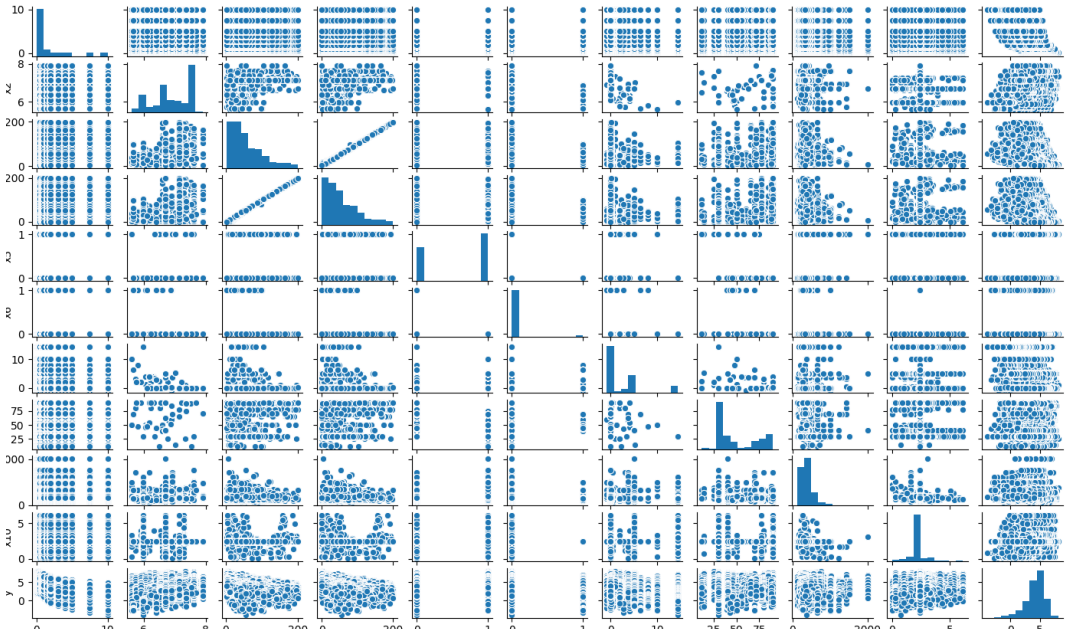
Following is the figure of correlation matrix by using ‘Spearman’ method

```
#print(df)
print(df.corr(method='spearman'))
```

652hw1

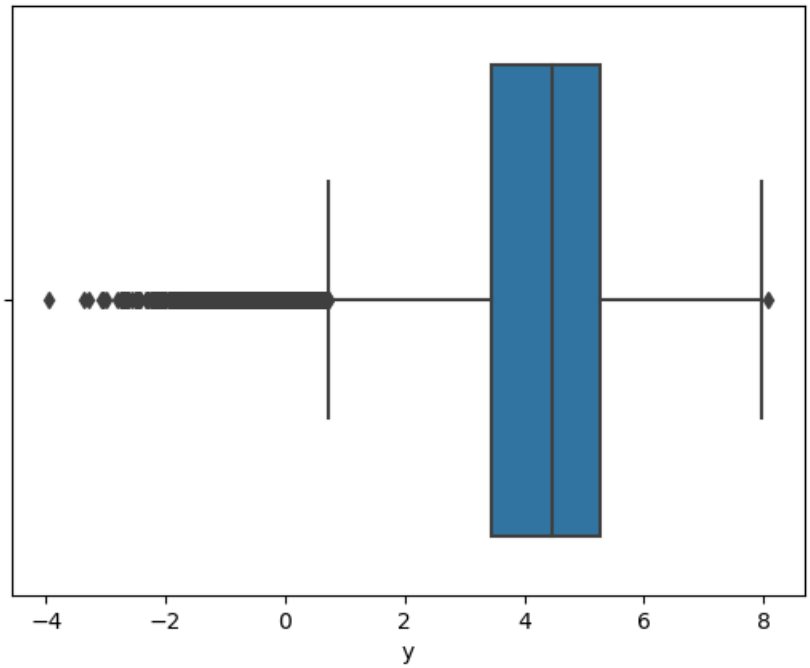
```
/Users/haodong/.conda/envs/12323/bin/python /Users/haodong/PycharmProjects/12323/652hw1.py
      x1        x2        x3        x4        x5        x6        x7  \
x1   1.0  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
x2   0.0  1.000000  0.389659  0.381014  0.480250 -0.173389 -0.744038
x3   0.0  0.389659  1.000000  0.996366  0.037127 -0.062090 -0.342631
x4   0.0  0.381014  0.996366  1.000000  0.024085 -0.058499 -0.354875
x5   0.0  0.480250  0.037127  0.024085  1.000000 -0.215133 -0.075815
x6   0.0 -0.173389 -0.062090 -0.058499 -0.215133  1.000000 -0.032061
x7   0.0 -0.744038 -0.342631 -0.354875 -0.075815 -0.032061  1.000000
x8   0.0 -0.301155  0.029906  0.068361 -0.590695  0.092309 -0.068992
x9   0.0 -0.018102 -0.122256 -0.125329  0.084500  0.086003  0.076125
x10  0.0 -0.024866  0.016993  0.009365 -0.128533  0.003337 -0.059042

           x8        x9       x10
x1   0.000000  0.000000  0.000000
x2  -0.301155 -0.018102 -0.024866
x3   0.029906 -0.122256  0.016993
x4   0.068361 -0.125329  0.009365
x5  -0.590695  0.084500 -0.128533
x6   0.092309  0.086003  0.003337
x7  -0.068992  0.076125 -0.059042
x8   1.000000 -0.043348 -0.061774
x9  -0.043348  1.000000 -0.215509
x10 -0.061774 -0.215509  1.000000

Process finished with exit code 0
```

Following is the correlation matrix in plot form with ‘Pearson’ method



3. Find outliers of Y

First plot y:



Then print outlier for y in an array:

```
# q3
# sb.boxplot(x = data1['y'])
# plt.show()

data = pd.DataFrame(data1, columns=['y'])
# print(data)

quartile_1, quartile_3 = np.percentile(data, [25, 75])
iqr = quartile_3 - quartile_1
lower_bound = quartile_1 - (iqr * 1.5)
upper_bound = quartile_3 + (iqr * 1.5)
outlier = np.where((data > upper_bound) | (data < lower_bound))
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
print(outlier)
```

```
652hw1 ×
/Users/haodong/.conda/envs/12323/bin/python /Users/haodong/PycharmProjects/12323/652hw1.py
(array([ 9859, 17507, 17849, ..., 25736, 25738, 25739]), array([0, 0, 0, ..., 0, 0, 0]))

Process finished with exit code 0
```

We can find the outlier for y from above output. For example, the 9859th, 17507th, 17849th y value are outliers, and some values after 17849th are also outliers.

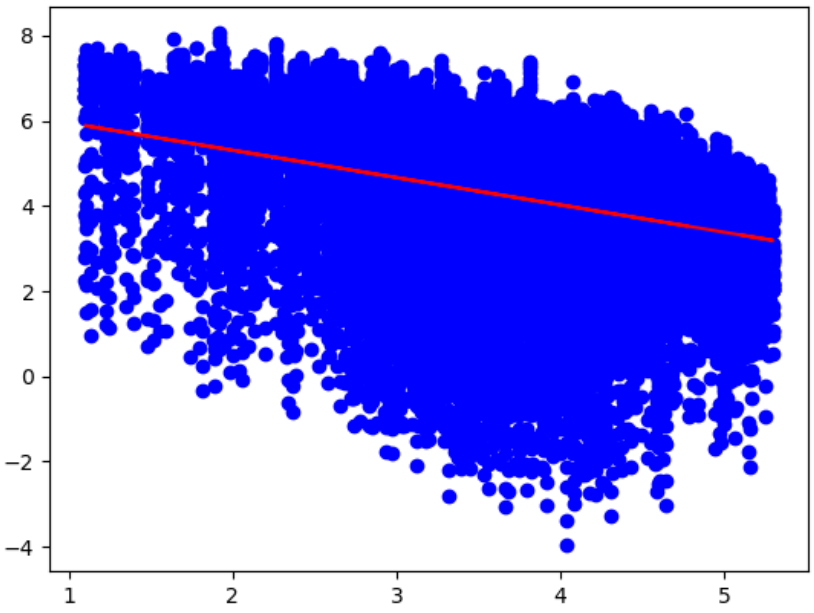
4. Predict Y using Ln(x3) and show the ANOVA table

In this model, Y is y and X is ln(x3), after fit the model, we can get following result.

```
652hw1
/Users/haodong/.conda/envs/12323/bin/python /Users/haodong/PycharmProjects/12323/652hw1.py
                            OLS Regression Results
==============================================================================
Dep. Variable:                      Y   R-squared:                       0.124
Model:                            OLS   Adj. R-squared:                  0.124
Method:                 Least Squares   F-statistic:                     3650.
Date:                Tue, 19 Feb 2019   Prob (F-statistic):               0.00
Time:                        16:21:52   Log-Likelihood:                -47362.
No. Observations:               25746   AIC:                         9.473e+04
Df Residuals:                   25744   BIC:                         9.474e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      6.5952      0.041    161.902      0.000       6.515       6.675
X             -0.6418      0.011    -60.411      0.000      -0.663      -0.621
==============================================================================
Omnibus:                     6824.900   Durbin-Watson:                   0.265
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            15941.560
Skew:                          -1.499   Prob(JB):                         0.00
Kurtosis:                       5.422   Cond. No.                         17.5
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
Intercept    6.595198
X           -0.641832
dtype: float64
```

From the model, we can fix a function for y and ln(x3):

$$y = 6.595198 - 0.641832 * \ln(x3)$$



Above figure is the plot of the model.

After using ANOVA test, we can obtain following ANOVA table.

```
                sum_sq       df            F  PR(>F)
X         8465.092794      1.0  3649.534826     0.0
Residual  59713.185176  25744.0          NaN     NaN

Process finished with exit code 0
```