Report

I used Anaconda to install xlrd, numpy and matplotlib library. I used Spyder to code.

Firstly I use xlrd to get the data from xlsx file, then I compute the mean, variance, and standard deviation of CS Score, Research Overhead, Admin Base Pay, Tuition using functions in numpy library.

Here are the results:

mu1 = 3.214

mu2 = 53.386

mu3 = 469178.816

mu4 = 29711.959

var1 = 0.448

var2 = 12.588

var3 = 13900134681.7

var4 = 30727538.733

sigma1 = 0.669

sigma2 = 3.548

sigma3 = 117898.832

sigma4 = 5543.243

After that I used function cov() and corrcoef() to compute covariance and correlation. I used matplotlib.pyplot to plot 6 graphs.

Here are the results:

covarianceMat =

[[ 4.57500000e-01 1.10562500e+00 3.87978185e+03 1.05847976e+03]

[ 1.10562500e+00 1.28504167e+01 7.02793765e+04 2.80578899e+03]

[ 3.87978185e+03 7.02793765e+04 1.41897208e+10 -1.63685641e+08]

[ 1.05847976e+03 2.80578899e+03 -1.63685641e+08 3.13676958e+07]]

correlationMat =

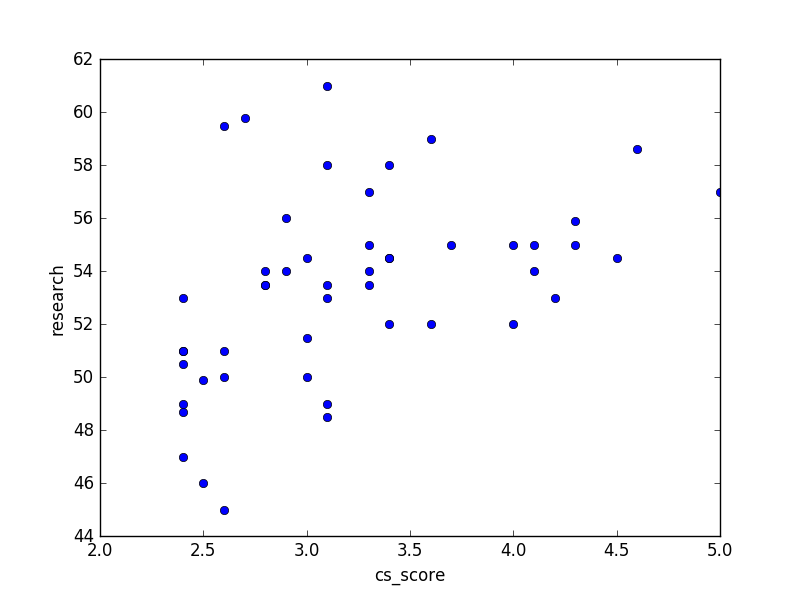
[[ 1. 0.4559883 0.04815316 0.27941242]

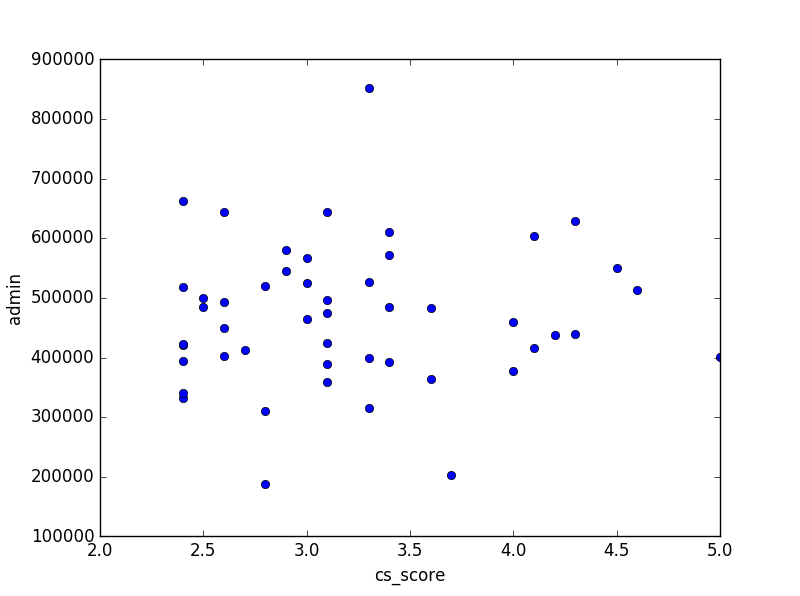
[ 0.4559883 1. 0.16458201 0.13975105]

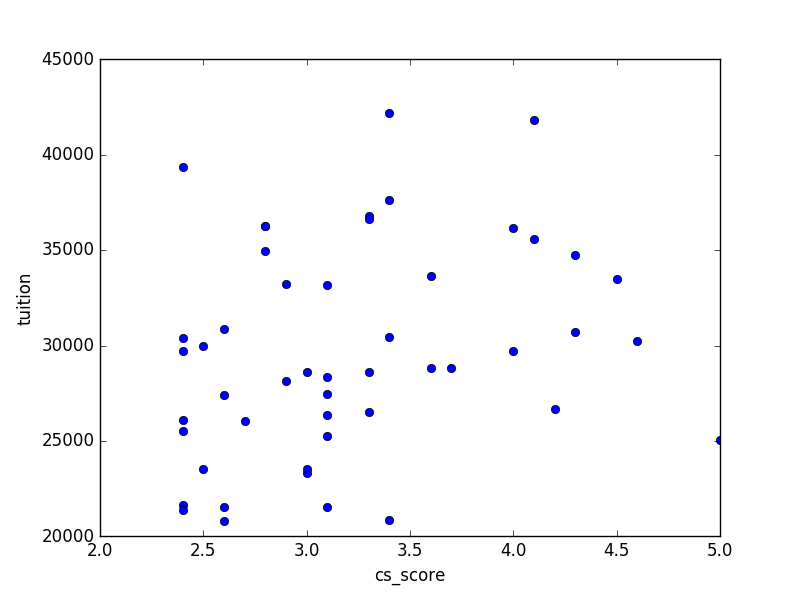
[ 0.04815316 0.16458201 1. -0.2453479 ]

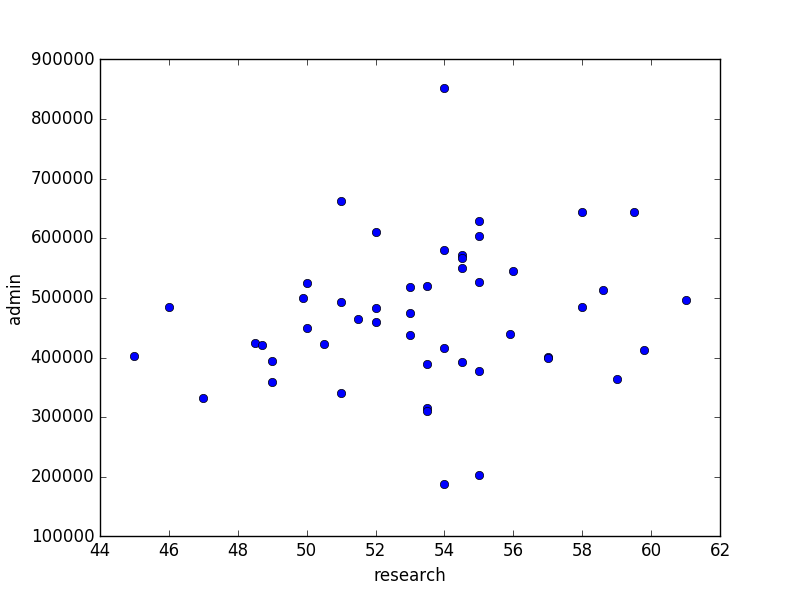
[ 0.27941242 0.13975105 -0.2453479 1. ]]

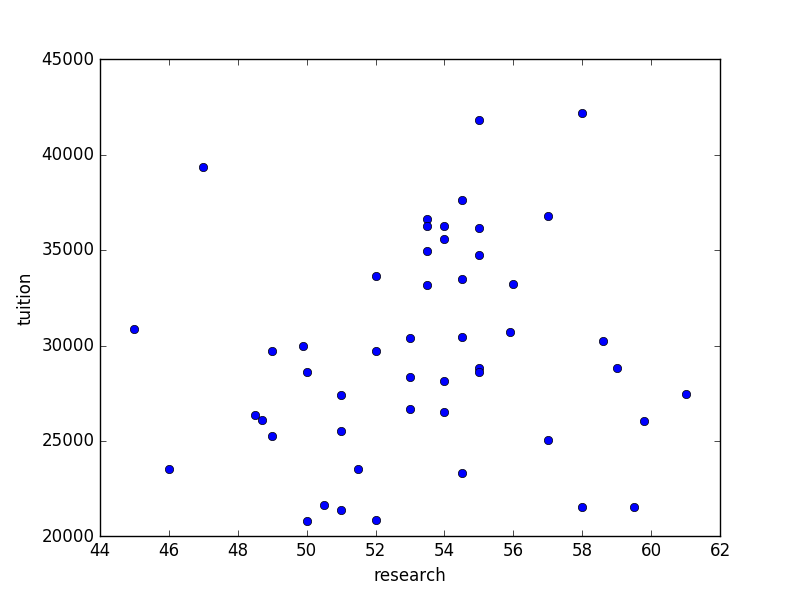
Here are the scatter plots:

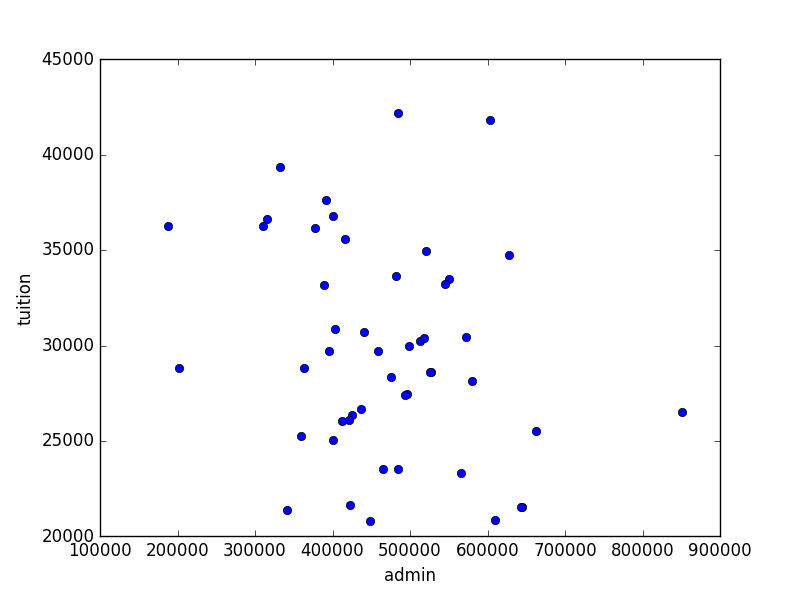












Log-Likelihood function of N independent samples **…** is

 (1)

And the probability density function (pdf) of normal distribution is:

 (2)

So

(3)

Let mean be μstandard deviation beσ, we can compute loglikelihood of every variable. Then we add them together.

The logLikelihood is -1315.09879256

I choose

[[0 0 0 0]

[1 0 0 0]

[0 0 0 1]

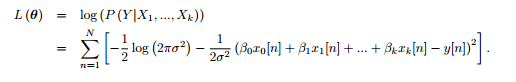
[1 0 0 0]]

As my Bayesian Network because in task 2 I got the matrix of correlation and I found the abs of research and cs\_score, admin and tuition, tuition and cs\_score are relatively higher, so I make them connected in BNgraph.

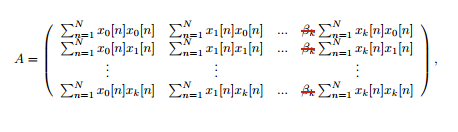
Given a Bayesian Network G of d variables, the log-likelihood is:

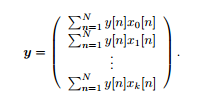
 (4)

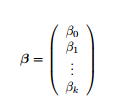
And the log-likelihood of Linear Gaussian Model is:

 (5)

In order to make *L*(***θ***) as large as possible, let

 (6)

 (7)

 (8)

I got an equation , then use numpy.linalog.solve to solve this equation and get *β*.

Use βto compute :

 (9)

Then use equation (5) we can compute log-likelihood of each variable( if a variable does not have a parent, then use equation (3)). Then we add them together to get the final log-likelihood

The BNloglikelihood is -1306.35338203