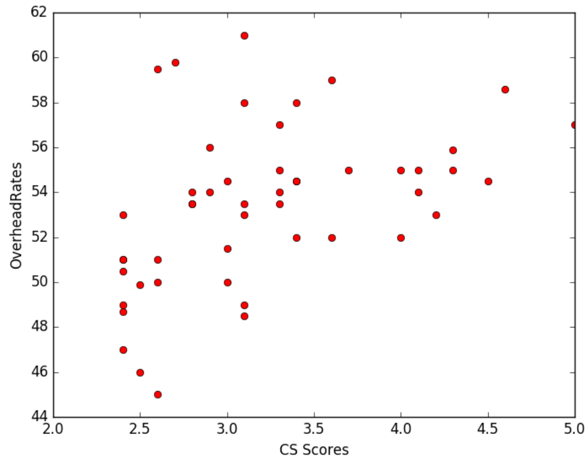
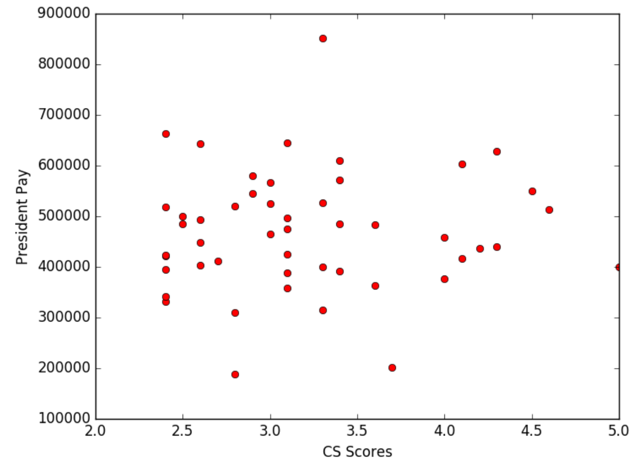
Joseph Boyd, Steven Walinski
Person #’s: 37165689, 50106260
Project 1 Report
CSE 474

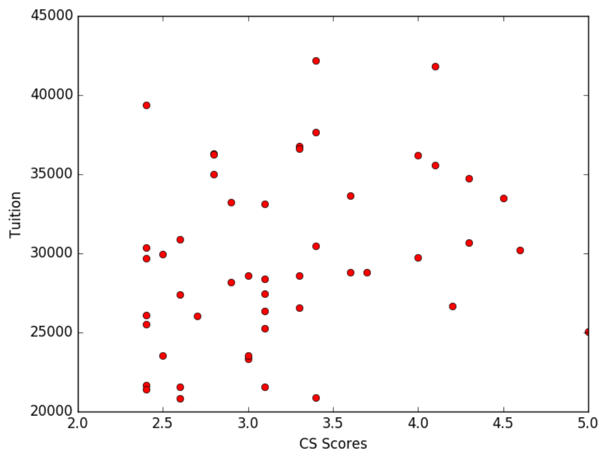
The beginnings of our program are self explanatory enough. All they aim to accomplish is to import the necessary libraries we will be utilizing within our project, moving on to importing the data from the provided ‘university data’ spreadsheet, which was done so manually. From there, we move to determine the mean, variance, and standard deviation of each of these data sets. We accomplish this by using available functions from ‘numpy’, in the form of ‘np.mean’, ‘np.var’, and ‘np.std’, and then printing those values while rounding each value to the requested three significant figures. Graphing these values does not really do much for us as the values from the dataset are extremely different, we do not gain any knowledge comparing the mean of CS GPA’s to the mean of the President’s Yearly Salary in a bar graph as the integer values are simply too far off. That is not to say comparing the values in a meaningful way is not valuable, on the contrary, which is why we have determined that representing the results in a table form is far more meaningful. Below we have a table allowing us to compare the mean, variance, and STD of each of our datasets. (Table generated using https://plot.ly/)

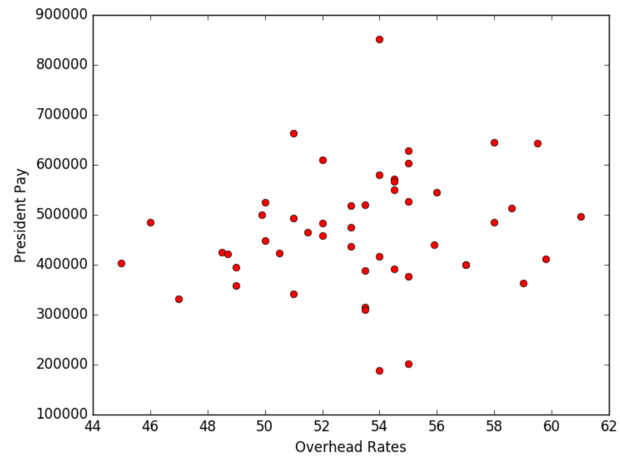
Data	CS Score	Overhead Rates	President Pay	Tuition
Mean	3.214	53.386	469178.816	29711.959
Variance	0.448	12.588	13900134681.7	30727538.733
Standard Deviation	0.669	3.548	117898.832	5543.243

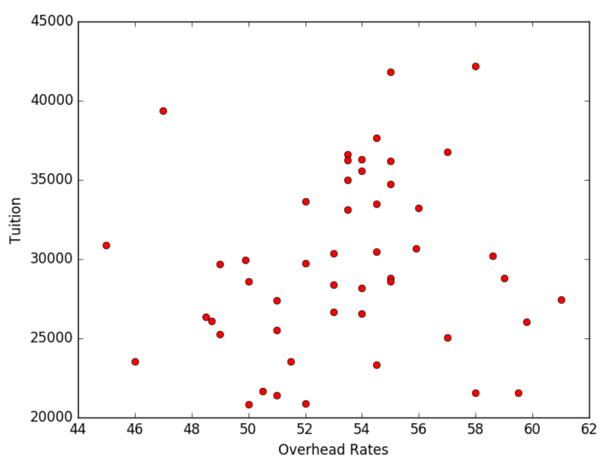
The Pairwise Data points are listed below for the six different pairs, each of the graphs are labeled.

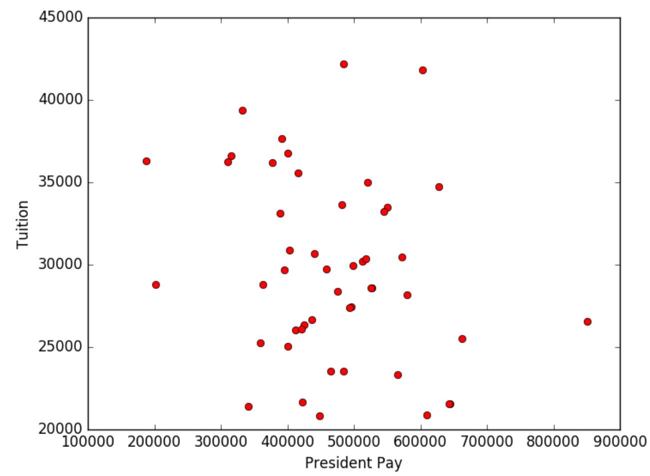






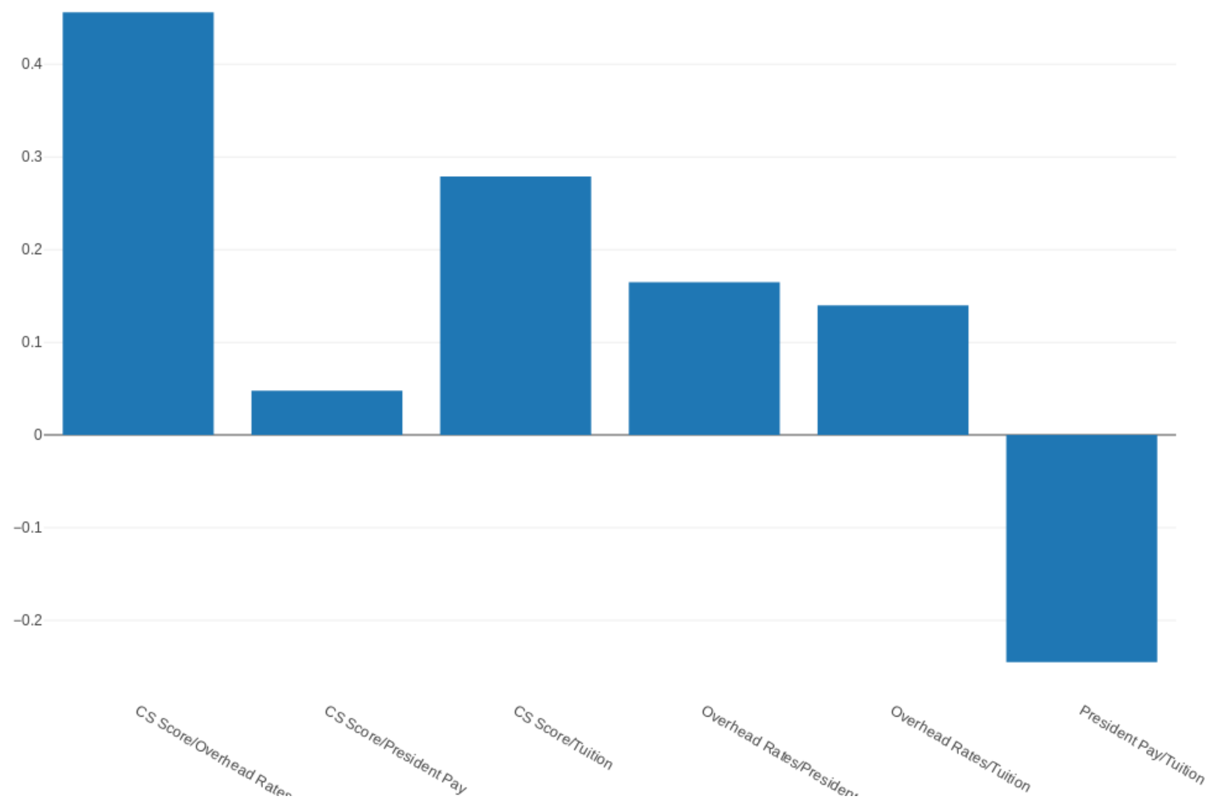






As stated below, our most highly correlated values are CS Scores and Overhead Rates, with our least correlated values being CS Score and President Pay. Interestingly, CS Scores and Research Overhead seem to have a large impact on one another while CS Scores and President Pay don't seem to effect each other at all.

Covariance and Correlation were determined using a combination of methods provided to us through ‘numpy’ in ‘np.cov’ and ‘np.corrcoef’, as well as some manual labor due to the methods returning us some duplicate values. For covariance, we determined the covariance of each pair of datasets, of which there were six, determined the required values we needed from the return of np.cov, and entered them into our matrix, ‘covarianceMat’. The same steps were taken with correlation, just of course using the ‘np.corrcoef’, instead of ‘np.cov’. CS Score and Overhead Rate were the two most highly correlated sets of data with a Correlation Coefficient of .456, whereas CS Score and President Pay were the two least correlated sets of data with a Correlation Coefficient of .048. A graph of Correlation Coefficients for each pair is below, only one pair was negatively correlated, that being President Pay and Tuition.



To calculate log likelihood supposing the variables are independent, we took the PDF of each of the original datasets using ‘stats.norm.pdf’ and passing in the dataset, the mean, and the STD. Once this was found we were able to take the log of each of the respective PDFs. We wrote a helper function called ‘listsum’ which allowed us to add each of the log values of each value in the list together, at which point we could determine the log likelihood by adding all these values together. We received a calculated log likelihood value of -1315.099.

To calculate log likelihood supposing the 4 variables are not independent, we used mostly the same strategy as as calculating with the variables being independent. The main difference was instead of using the normal PDF function (‘stats.norm.pdf’) we used the multivariate ('stats.multivariate_normal.pdf') and passed in the dataset, the mean, and covariance along the diagonal (which is just equal to the variance). Then we were able to use the same strategy to determine the log likelihood, just using the multivariate PDF.

The output of running our program is listed below:

```
UBitName = jnboyd and shwalins
personNumber = 37165689 and 50106260
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
covarianceMat:
[[0.488 1.106 3879.782 1058.480]
 [1.106 12.588 70279.377 2805.789]
 [3879.782 70279.377 13900134681.701 -160345117.967]
 [1058.480 2805.789 -160345117.967 30727538.733]]
correlationMat:
[[1.000 0.456 0.048 0.279]
 [0.456 1.000 0.165 0.140]
 [0.048 0.165 1.000 -0.245]
 [0.279 0.140 -0.245 1.000]]
(Independent) logLikelihood:
-1315.099
(Dependent) logLikelihood:
-1331.86
```