Project 1: Probability Distributions and Bayesian Networks

Avinash Kommineni

50248877

akommineni@buffalo.edu

September 25, 2017

Problem 1.1

The excel file is read and loaded into the workspace by the *read_excel* query of *pandas* library as a dataframe. By selecting only the necessary columnsand omitting the others, it is easier to handle the required data. The mean(μ), variance(σ^2) and standard deviation(σ) can be calculated either from pandas inbuilt functions or numpy inbulit functions. But there is a slight difference between these two in case of variance and standard-deviation. The difference being, the *df.var()* considers the number of elements as *N-1* where N is the total number of rows (49 in this case). The numpy way of calculating is called *Unbiased Estimator*. So for the above purposes, the libraries numpy and pandas are used (imported).

Listing 1: Calculating Mean Variance Standard deviation.

```
1  import numpy as np
2  import pandas as pd
3
4  df = pd.read_excel('DataSet/university data.xlsx')
5  FORMAT = ['CS Score (USNews)', 'Research Overhead %', 'Admin Base ↩
       Pay$', 'Tuition(out-state)$']
6  df = df[FORMAT]
7  print('UBitName = akommine')
8  print('personNumber = 50248877')
9
10 print('\nMeans')
11 mu = np.mean(df)
12 for i in range(mu.size):
13 print('mu{} {:.3f}'.format(i+1,mu[i]))
14
```

```
15 print('\nVariance')
16 var = df.var()
17 for i in range(var.size):
18 print('var{} {:.3f}'.format(i+1,var[i]))
19
20 print('\nVariance using Unbiased Estimator')
21 var2 = np.var(df)
22 for i in range(var2.size):
23 print('var{} {:.3f}'.format(i+1,var2[i]))
24
25 print('\nStandard Deviation')
26 std = df.std()
27 for i in range(std.size):
28 print('sigma{} {:.3f}'.format(i+1,std[i]))
29
30 print('\nStandard Deviation using Unbiased Estimator')
31 std2 = np.std(df)
32 for i in range(std2.size):
33 print('sigma{} {:.3f}'.format(i+1,std2[i]))
```

Problem 1.2

All the data from dataframe is stored as array for easy access of just the values and not the labels, and also for further future purposes. The *covarianceMat* and the *correlationMat* are calculated from *numpy*'s in-built functions *np.cov* and *np.corrcoef*.

Listing 2: Calculating covarianceMat and correlationMat matrices.

```
1 Y = np.vstack((df['CS Score (USNews)'], df['Research Overhead %'], ↩
    df['Admin Base Pay$'], df['Tuition(out-state)$']))
2 Y = np.delete(Y,-1,1)
3 covarianceMat = np.cov(Y)
4 np.set_printoptions(formatter={'float': lambda x: "{:0.3f}".format(↩
    x)})
5 print('\nCovarianceMat:\n{}'.format(np.round(covarianceMat,decimals↩
    =3)))
6 correlationMat = np.corrcoef(Y)
7 print('\ncorrelationMat:\n{}'.format(correlationMat))
```

Problem 1.3

The *logLikelihood* of the data is calculated by the formula

$$L(\mu, \sigma^2 : x_1, x_2, ..., x_n) = -\frac{n}{2}\ln(2\pi\sigma^2) - \frac{n}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2$$

The equation has been implemented in a effecient way by the use of broadcasting.

Listing 3: Calculating logLikelihood value.

```
1 loglike =  -(df.count()[0]/2)*np.log(2*np.pi*var2) - (0.5/var2)*np.↩
    sum((Y-mu[:,np.newaxis])**2,axis=1)
2 totalLog = np.sum(loglike)
3 print('\nlogLikelihood is: {}\n'.format(totalLog))
```

Problem 1.4

The *logLikelihood* of the multivariate gaussian distribution is calculated using the inbuilt function of multivariate_normal from scipy library.

Listing 4: Calculating logLikelihood for multivariate gaussian.

```
1 from scipy.stats import multivariate_normal
2
3 multivariate_normalvar = multivariate_normal.logpdf(Y.T, mu, cov = ↩
    covarianceMat,allow_singular = True)
4 multivariate_normalvar = sum(multivariate_normalvar)
5 print('logLikelihood for multivariate gaussian is: {}\n'.format(↩
    multivariate_normalvar))
```

Code Output

Listing 5: Code output.

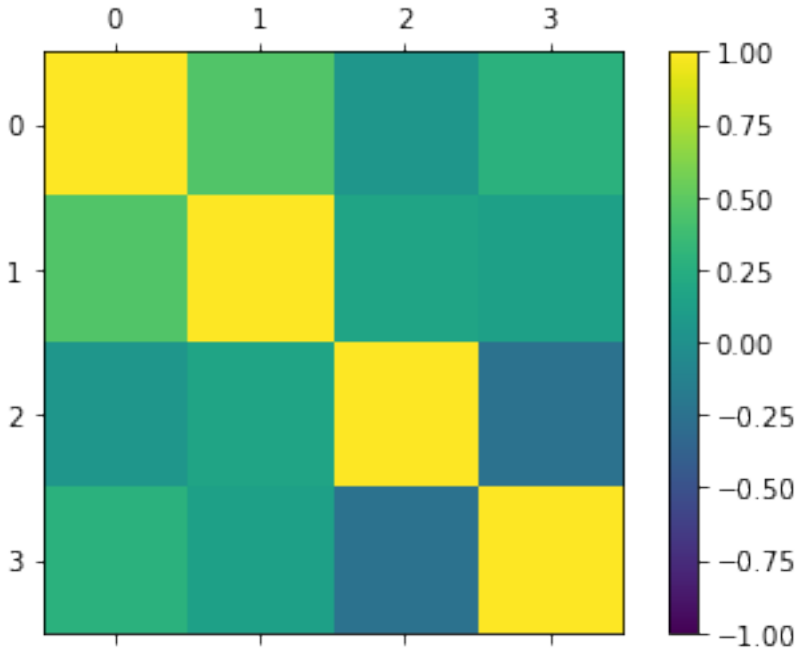
```
1 UBitName = akommine
2 personNumber = 50248877
```

```
 3
 4 Means
 5 mu1 3.214
 6 mu2 53.386
 7 mu3 469178.816
 8 mu4 29711.959
 9
10 Variance
11 var1 0.457
12 var2 12.850
13 var3 14189720820.903
14 var4 31367695.790
15
16 Variance using Unbiased Estimator
17 var1 0.448
18 var2 12.588
19 var3 13900134681.701
20 var4 30727538.733
21
22 Standard Deviation
23 sigma1 0.676
24 sigma2 3.585
25 sigma3 119120.615
26 sigma4 5600.687
27
28 Standard Deviation using Unbiased Estimator
29 sigma1 0.669
30 sigma2 3.548
31 sigma3 117898.832
32 sigma4 5543.243
33
34 CovarianceMat:
35 [[0.457 1.106 3879.782 1058.480]
36 [1.106 12.850 70279.376 2805.789]
37 [3879.782 70279.376 14189720820.903 -163685641.258]
38 [1058.480 2805.789 -163685641.258 31367695.790]]
39
40 correlationMat:
41 [[1.000 0.456 0.048 0.279]
42 [0.456 1.000 0.165 0.140]
43 [0.048 0.165 1.000 -0.245]
```

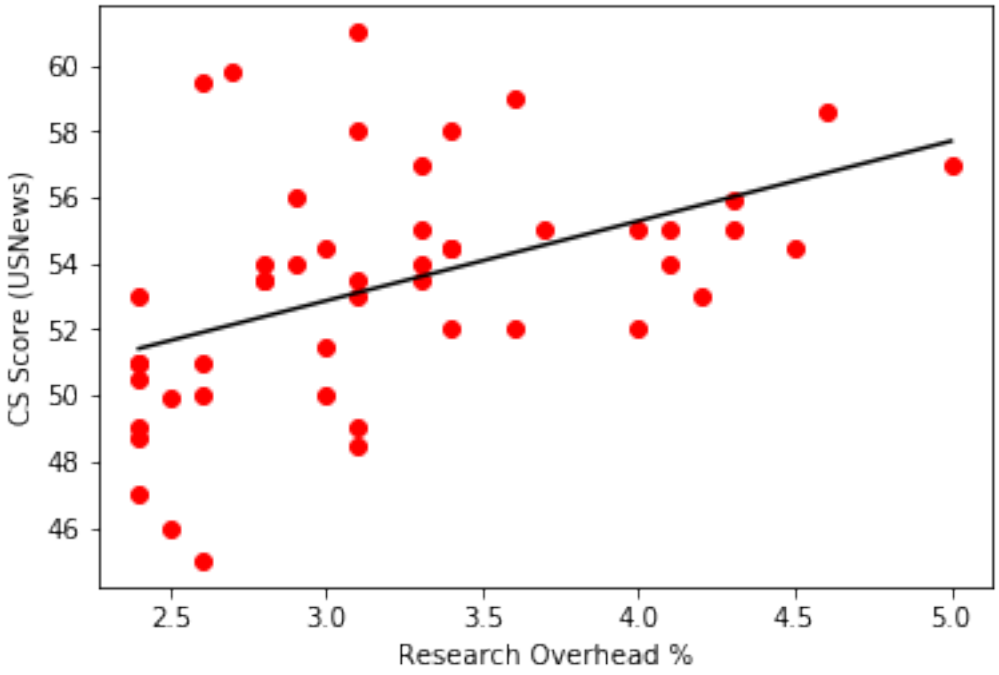
```
[0.279 0.140 -0.245 1.000]]

logLikelihood is: -1315.098792560739

logLikelihood for multivariate gaussian is: -1262.32720006137
```

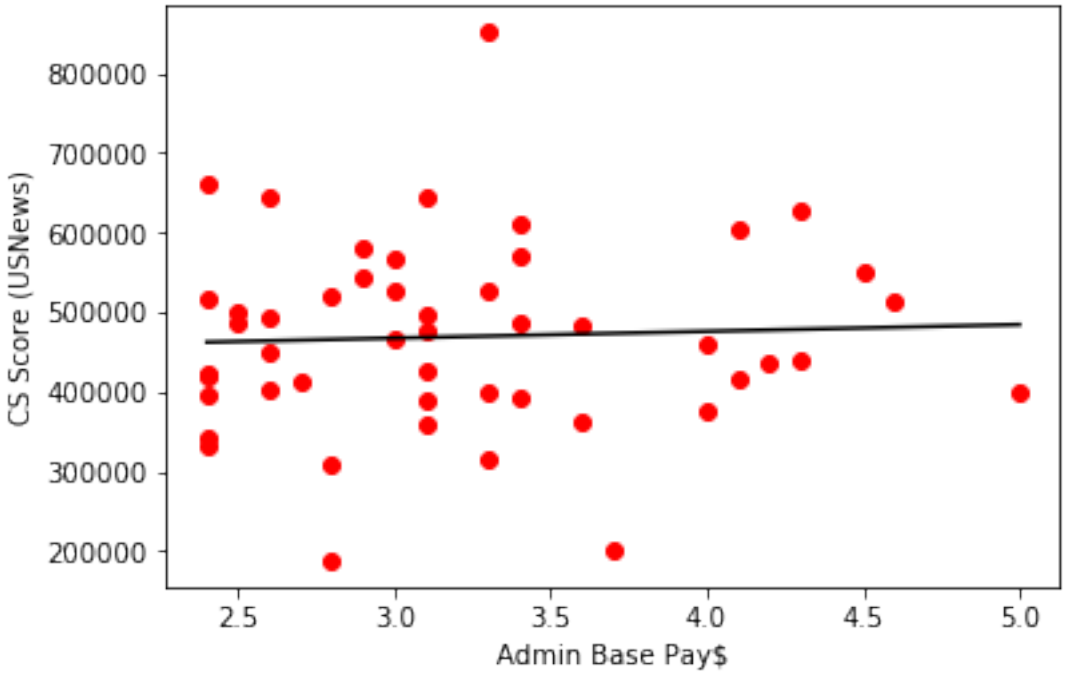
Results

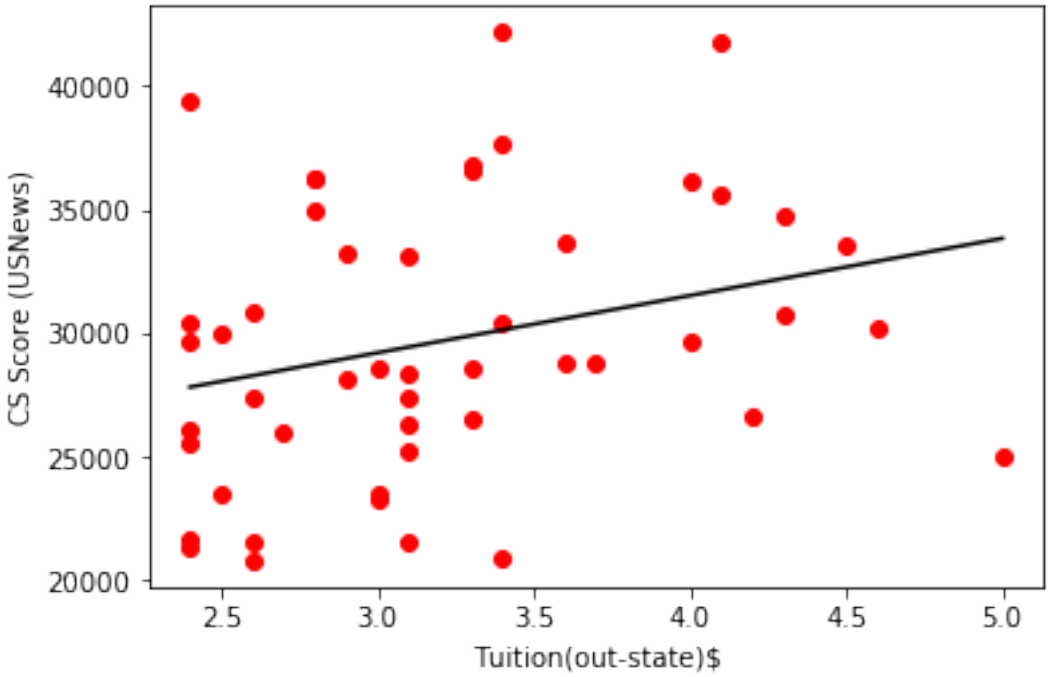
To better understand the correlation matrix, the following plot is drawn to portray the correlation between each of the four variables...

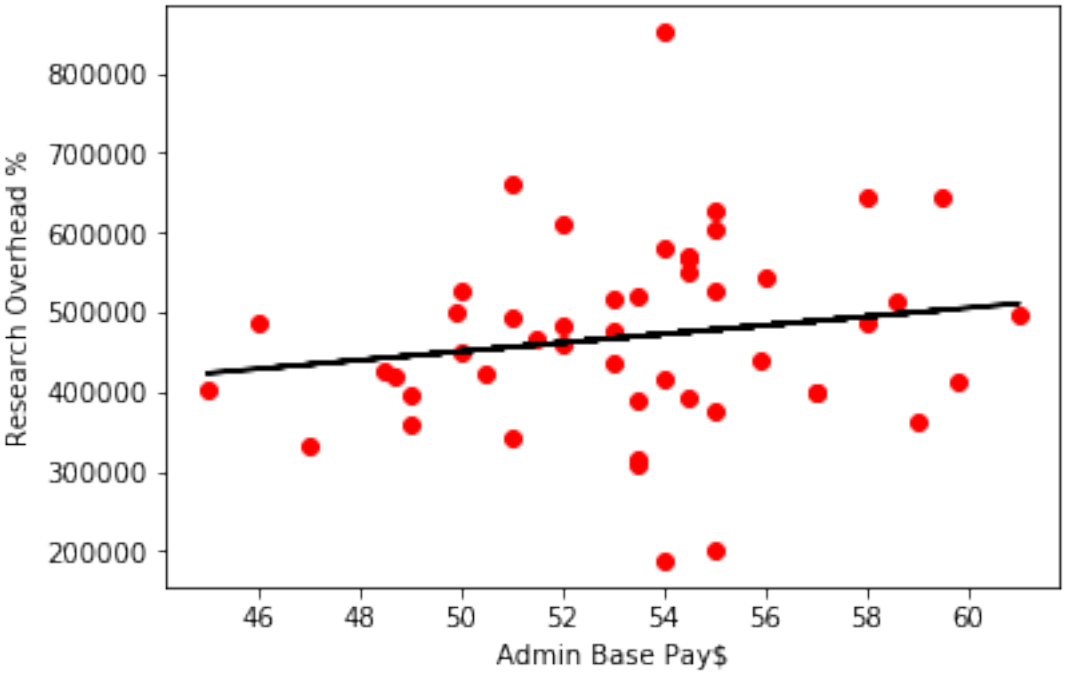


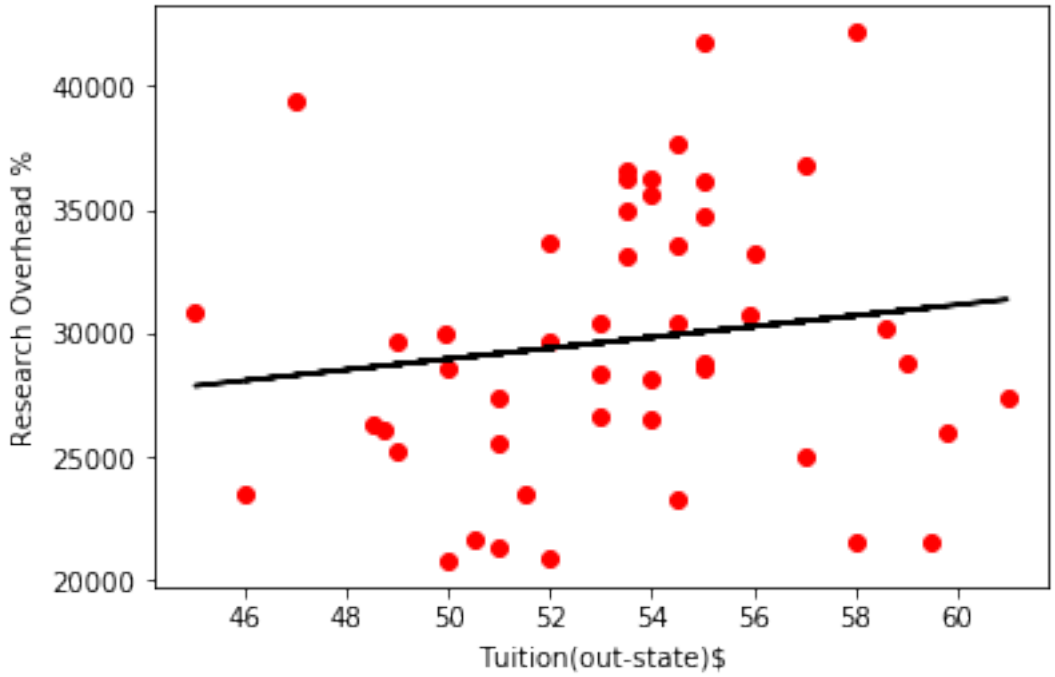
The follwing are plots for each pairs of variables and a linear fit to the data.

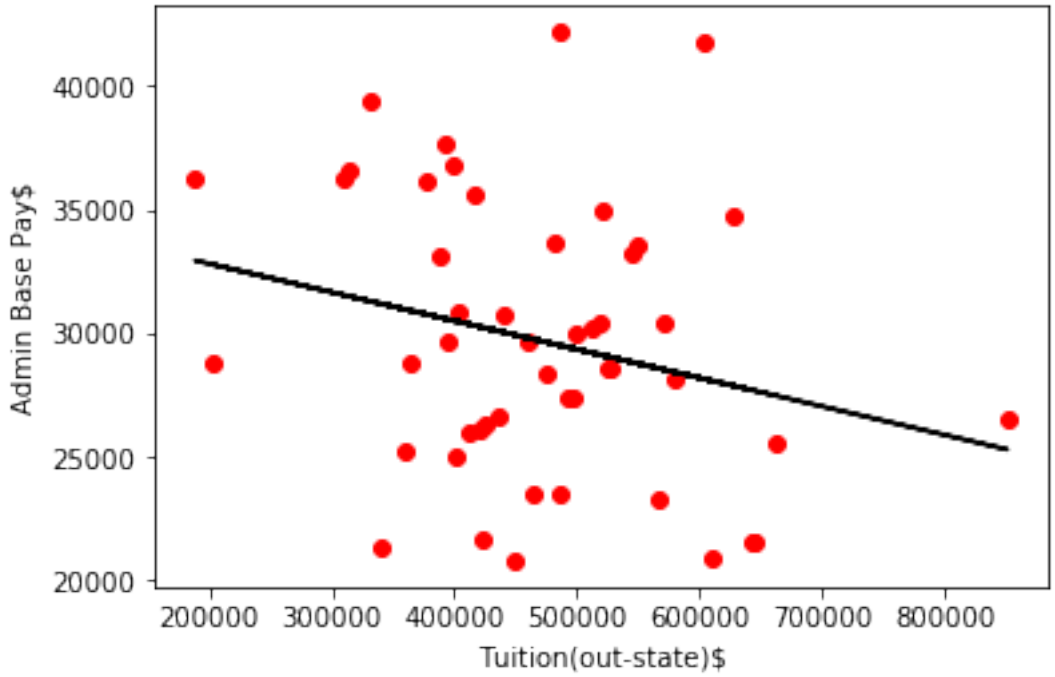

CS Score (USNews)
60
58
56
54
52
50
48
46
2.5
3.0
3.5
4.0
4.5
5.0
Research Overhead %


CS Score (USNews)
800000
700000
600000
500000
400000
300000
200000
2.5
3.0
3.5
4.0
4.5
5.0
Admin Base Pay$


CS Score (USNews)
40000
35000
30000
25000
20000
2.5
3.0
3.5
4.0
4.5
5.0
Tuition(out-state)$


Research Overhead %
800000
700000
600000
500000
400000
300000
200000
46
48
50
52
54
56
58
60
Admin Base Pay$


Research Overhead %
40000
35000
30000
25000
20000
46
48
50
52
54
56
58
60
Tuition(out-state)$


Admin Base Pay$
40000
35000
30000
25000
20000
200000
300000
400000
500000
600000
700000
800000
Tuition(out-state)$