**1 Assignment 1 (due April 15, 10 points)**

**1.1 Problem 1 (5 points)**

**Consider the following program:**

*import random*

*def make\_data(n=1000):*

*table = []*

*for i in range(n):*

*a,b = random.random(),random.random()*

*table.append((a+2.0\*b, a\*b, a-b))*

*return table*

**It returns three columns of numbers. Write a program to compute the**

**mean, variance, and standard deviation of each column. Also compute the**

**correlation and covariance of each two columns.**

**Analyze the program and determine the theoretical expected values for**

**mean, variance, standard deviations, covariances and correlations.**

Analyze:

For every Xi in uniform distribution from 0 to 1, the expected value is 0.5. Consequently, according to strong law of large number, if Xi is from a sequence of independent and identically uniform distributed random variables with the same expected value and finite variance, then

Thus, expected value for a and b is 0.5.

For the three rows in the table, expected value for column one is

E[a+2b] = E[a]+2E[b] = 1.5,

Expected value for column two is

E[a\*b] = E[a] \* E[b] = 0.25,

Column three`s expected value is E[a-b] = E[a] - E[b] = 0.

Python computing results for those numbers: C:\Users\felix\Desktop\csc521\assignment 1\4.jpg

Using code:

*#compute the means for each column*

*mu\_x=sum(row[0] for row in table)/len(table)*

*mu\_y=sum(row[1] for row in table)/len(table)*

*mu\_z=sum(row[2] for row in table)/len(table)*

For variances of three columns:

E[x2] = ,

E[x]2 =E[X]\*E[X], so E[A]2=E[B]2=1/4,

E[a2] - E[a]2  = E[b2]-E[b]2=1/12,

E[a2]=E[b2] = E[ab]=1/12(independent event).

So for the first column variance = E[x2] - E[x]2 = E[(a+2b)2]-E[a+2b] 2 = E[a2]+E[4b2]+E[4ab] = 8/3 - 9/4 = 5/12 = 0.416.

Second column`s variance = E[(ab)2] - E[ab]2 = 1/9 - 1/16 = 0.0486

Third variance = E[(a-b)2] - (E[a-b])2 = 4/6 - 3/6 = 1/6 = 0.167

Standard Deviation for three columns, according to previous computations, ,are 0.645, 0.22,0.409. which is also almost equal to the results I got from Python(see below graph).

Python codes for those parts are:

*#standard variance*

*sigma\_X=va\_X\*\*0.5*

*sigma\_Y=va\_Y\*\*0.5*

*sigma\_Z=va\_Z\*\*0.5*

*#standard error*

*dmu\_X=sigma\_X/(1000\*\*0.5)*

*dmu\_Y=sigma\_Y/(1000\*\*0.5)*

*dmu\_Z=sigma\_Z/(1000\*\*0.5)*

result:

C:\Users\felix\Desktop\csc521\assignment 1\QQ截图20120415173841.jpg

For variance and correlation:

Cov(x,y) = E[XY]-E[X]E[Y], so Cov(row[0],row[1]) = E[(a+2b)(a\*b)]-E[a+2b]E[a\*b] = E[a2b]+E[2ab2]-3/2\*1/12, since

E[a2b] = E[ab2] = E[a3], E[X3] = 3dx = 1/4 X4(FROM 0 TO 1) = 1/4,

Cov(row[0],row[1])= E[a2b]+E[2ab2]-3/2\*1/12 = 1/4 +2\*(1/4) - 1/2 = 1/8 = 0.125(shown in python as cov\_xy)

As proven, Cov(row[0], row[2]) = 1/3+1/3-2\*(1/3) = 0(shown in Python as cov\_xz)

And Cov(row[1],row[2]) = 0 (shown in python as cov\_yx)

C:\Users\felix\Desktop\csc521\assignment 1\3.jpg

Correlation for row[0] and row[1] = Cov[row[0],row[1]]/std(x)\*std(y) = 0.125/(0.645\*0.22) =0.881

Corr(row[0],row[2]) = cov(row[0],row[2])/std\_x\*std\_z = 0

Corr(row[1],row[2]) = cov(row[1],row[2])/std\_z\*std\_y = 0

C:\Users\felix\Desktop\csc521\assignment 1\3.jpg