Q1

b) The experiment to run is as shown below:

# -\*- coding: utf-8 -\*-

"""

Created on Fri Oct 19 08:38:45 2018

@author: tchat

"""

import numpy as np

import matplotlib.pyplot as plt

# Generate dataset

x1 = np.random.uniform(-1,1,1000)

x2 = np.random.uniform(-1,1,1000)

y1 = x1\*\*2

y2 = x2\*\*2

a = x1+x2

b = -x1\*x2

# Plot g\_bar vs f

for i in range(0,1000):

x = [x1[i],x2[i]]

y = [y1[i],y2[i]]

plt.plot(x,y,'g')

x\_value = np.arange(-1,1,0.01)

y\_value = x\_value\*\*2

plt.plot(x\_value,y\_value,'r')

plt.title('Plot of g\_(x) and f(x) together')

plt.xlabel('Input values between -1 and 1')

plt.ylabel('Output values of the function')

# Calculate g\_bar

x = np.random.uniform(-1,1,1000)

a\_gbar = np.mean(x1+x2)

b\_gbar = -np.mean(x1)\*np.mean(x2)

g\_bar = a\_gbar \*x + b\_gbar

plt.plot(x,g\_bar, 'b')

# Calculate bias

f\_x = x\*\*2

bias = np.mean((g\_bar - f\_x)\*\*2)

print('Value of Bias is : ', bias)

# Calculate variance

g\_x = a\*x + b

var = np.mean((g\_x - g\_bar)\*\*2)

print('Value of Variance is : ', var)

# Calculate Eout

eout = np.mean((g\_x - f\_x)\*\*2)

print('Value of Eout is : ', eout)

print('Value of E[Eout] is : ', np.mean(eout))

c) The results of the experiment are:

Value of Bias is : 0.184952445711

Value of Variance is : 0.338417840272

Value of Eout is : 0.540673960157

Value of E[Eout] is : 0.540673960157

A close up of a logo

Description generated with very high confidence

As we can see from our results,

Expected value of Eout is almost equal to the value of variance + bias experimentally.