Name: DIXIT GURUNG

Class and Section: EGR 223 -02

Instructor: Professor Baine

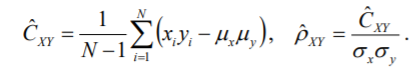
Laboratory # 8

Laboratory Title: Linear Regression for Curve Fitting

Date: 03/30/20

**Introduction:** The main goal of this lab was to work with expectation and multiple variables. In addition to that MATLAB was used to explore the formulations for linear regression.

**Procedure and Results:  
 1) Write a MATLAB function (include error checking) to compute and return the sample-based estimates of the covariance of X and Y and the correlation coefficient between X and Y as**



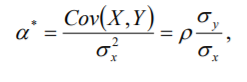
A function named covarandcorel.m was created which a matrix as an input argument and returns covariance, correlation coefficient, mean and standard deviation. The two columns of input matrix was divided into two separate matrices. Next, above two equations were used to calculate the covariance and correlation coefficient**.** The program for this part can be found in **APPENDIX A** below.

**2)Download the 2 data files labeled Baseline.txt and Reperfusion.txt. Each data file consists of two columns of simultaneously collected data from an intact guinea pig heart. Column 1 is left ventricular electrical data and column 2 is left ventricular pressure data – each sampled at 400 Hz. The baseline data were collected during normal sinus rhythm while the reperfusion data were collected after the heart was subjected to 30 min of ischemic insult resulting in loss of cardiac function. Perform a linear regression analysis between the columns for each of the data sets. You may assume that the electrical data is the independent random variable. A simple way to read a text file into MATLAB uses the load command:**

**Report values for {alpha, beta} and {correlation coeff} for baseline, and {alpha, beta} and {correlation coeff} for reperfusion. Comment on your findings – what were the differences, what were the similarities, could you use this type of analysis to create an objective measure of cardiac function?**

Load function was used to read both text files.

covarandcorel.m was used to find the correlation coefficient. Then, the two equations below were used to calculate alpha and beta.





The values of alpha, beta and correlation coeff can be seen in **table 1** below.

Table 1: Results and findings for part 2

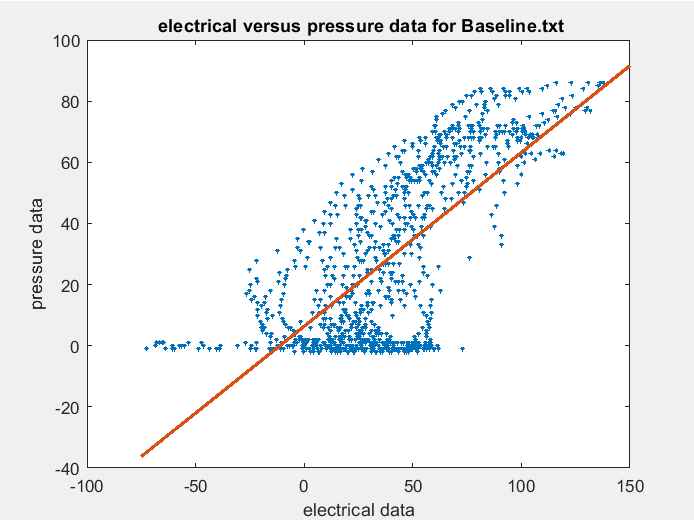
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Alpha** | **Beta** | **Correlation coeff** | **Covariance** |
| **Baseline** | 0.568 | 6.376 | 0.7057 | 710.2 |
| **Reperfusion** | 0.00776 | 15.36 | 0.0756 | 25.36 |

According to the table above, the correlation coefficient for baseline and reperfusion are 0.7057 and 0.0756 respectively. Since 0.7057 is closer to 1, the variables in baseline are almost linearly dependent. Similarly, 0.0756 is closer to 0, meaning variables in reperfusion are almost uncorrelated. This type of analysis cannot be used to create an objective measure of cardiac function.

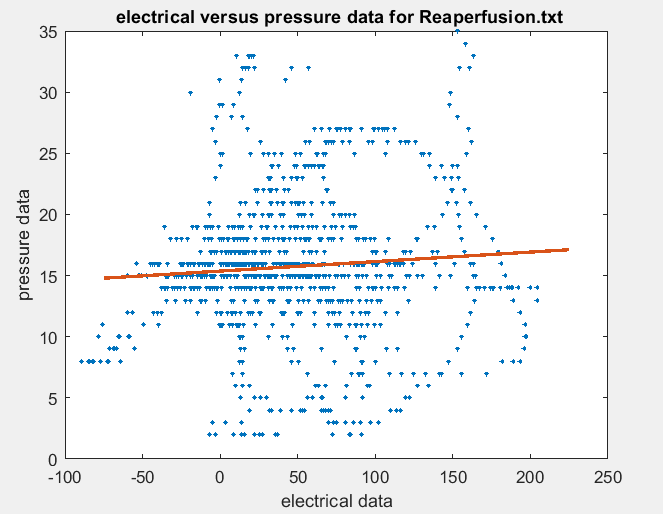
The code for this part can also be found in **APPENDIX A** below.

**For each data file, create a scatter plot of electrical versus pressure data and then create a line plot using the empirically derived coefficients (use a different color for each plot line; use just markers without lines to create the scatter plot, investigate different options that the plot command provides; you can keep the current plot window active by issuing the hold command to toggle plot hold on/off). Comment on the “goodness of fit” of your plot**

For this part of the lab, normal plot command was used to create markers. Hold on/off command was also used to hold the plot. The plot for Baseline and Reperfusion can be seen in  **figure 1 and 2** respectively.The code for this part can be found in **Appendix A** below



**Figure 1:** Plots for Baselline.txt

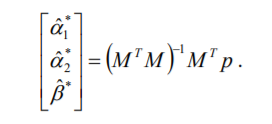


**Figure 2:** Plot for Reperfusion.txt

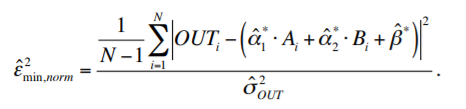
**3)Let’s try multiple variable linear regression:**

**where “OUT” is output voltage, “A” is the voltage on input A and “B” is the input voltage on input B. Use the matrix-based approach to find {alpha1,alpha2,beta}. To quantify the “goodness” of your fit, compute the empirical normalized prediction error of**

At first, lab8\_data.mat file was read using load function. Then, the vector A, B and ones column vector was merged to create a 1000\*3 matrix M. Next, the equation below was used to calculate alpha1, alpha2 and beta.



Next, the empirical normalized prediction error was computed using the equation below.



The findings from this part is listed below:

Alpha1 = -5.0217

Alpha2 = 6.0239

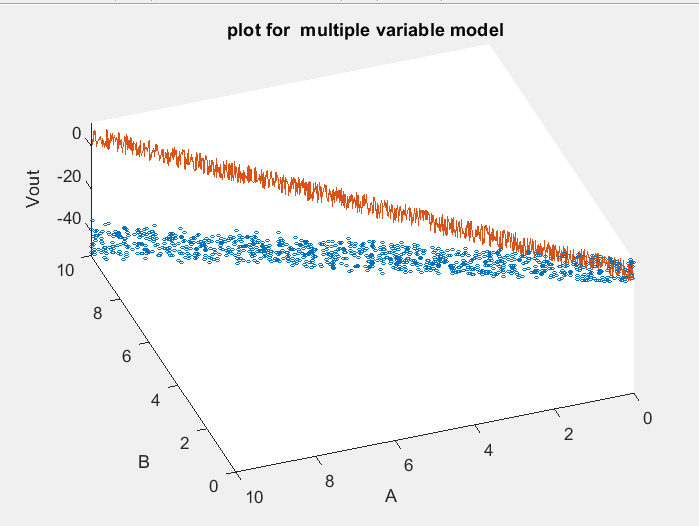
Beta = -0.009696

E = 2.075

The value of E is 2.075 which is greater than 1. This implies a poor fit.

**Use the 3-D plotting capabilities of MATLAB to display your results. First of all, create a 3-D scatter plot of the given inputs versus the output (use either scatter3 or plot3 – the later can be used to create a scatter plot by using a plotting format of no line with an ‘o’ symbol).**

Given inputs(A and B) versus the output(Vout) was plotted using plot3 function. Next, 1-D test vectors for A and B was created that linearly incrementally cover this range. Then, the predicted output signal was calculated and plotted using plot3 function. This plot can be seen in **figure 3** below. From the figure, It is clear that the predicted output is a poor fit as mentioned earlier.



**Figure 3:** Plot for part 3

**Vary the view angle to get the best view to see how well your curve models your data. Comment on the goodness of fit.**

More plots from different views can be found in **APPENDIX B** below.

**4. What do you think the “black box” circuit is? Design something that behaves this way. Include a sketch of your circuit in your report.**

It is clear from the plot above that the Vout decreases when A and B increases and increases when A and B decreases. In the figure above, I absorbed a linear relationship between output and inputs. Therefore, the black box should be simple resistor circuit.

**APPENDIX A**

**Covarandcorel.m**

function [covar,corel,stdX,stdY,meanX,meanY] = covarandcorel(matrixA)

%marirxA is a input argument where column 1 is X and column 2 is Y

[m,n] = size(matrixA);

column1 = matrixA(:,1); %copying the 1st column from the matrix A

column2 = matrixA(:,2); %copying the 2nd column from the matrix A

meanX = mean(column1);

meanY = mean(column2);

%calculating covariance

covar = column1 .\* column2;

covar = covar - (meanX \* meanY);

covar = sum(covar);

covar = covar/(m-1);

%calculating coeff of correlation

stdX = std(column1);

stdY = std(column2);

corel = covar/(stdX\*stdY);

end

**lab8.m**

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Title: EGR223-02 Lab 7

% Filename: lab8.m

% Author: Dixit Gurung

% Date: 3/25/2020

% Instructor: Dr. Nicholas Baine

% Description: Work with expectation and multiple variables.

% Explore formulations for linear regression.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc

clear all

close all

%{

%%%PART 2

%%%%%%%%%%%%baseline.txt%%%%%%%%%%%%%%%%%%%%%

baseline = load('Baseline.txt', '-ascii');

[covar1,corel1,stdX1,stdY1,meanX1,meanY1] = covarandcorel(baseline);

alpha1 = (corel1\*stdY1)/stdX1;

beta1 = meanY1 - (alpha1\*meanX1);

figure

plot(baseline(:,1),baseline(:,2),'\*','MarkerSize',2)

xlabel('electrical data');

ylabel('pressure data');

title(' electrical versus pressure data for Baseline.txt');

hold on

x= linspace(-75,150, 100);

y= (alpha1\*x) + beta1;

plot(x,y,'LineWidth',2)

hold off

%%%%Reaperfusion.txt%%%%%%%%%%%%%%%%%%%%%%%%%%%

Reaperfusion = load('Reaperfusion.txt', '-ascii');

[covar2,corel2,stdX2,stdY2,meanX2,meanY2] = covarandcorel(Reaperfusion);

alpha2 = (corel2\*stdY2)/stdX2;

beta2 = meanY2 - (alpha2\*meanX2);

figure

plot(Reaperfusion(:,1),Reaperfusion(:,2),'\*','MarkerSize',2)

xlabel('electrical data');

ylabel('pressure data');

title(' electrical versus pressure data for Reaperfusion.txt');

hold on

x= linspace(-75,225, 400);

y= (alpha2\*x) + beta2;

plot(x,y,'LineWidth',2)

hold off

%}

%%{

%%%%part 3%%%%%%%%%%%%%%%%%%%%%

S = load('lab8\_data.mat', '-mat');

one = ones(1000,1);

M = [S.A(:),S.B(:),one(:)]; %merging three column vectors to form a 3\*1000 [MATRIX M]

P = S.Vout;

P = (P');

MT = (M');

unknown = MT\*M;

unknown = inv(unknown);

unknown = unknown \* MT \*P;

alpha1 = unknown(1,1);

alpha2 = unknown(2,1);

beta = unknown(3,1);

%Empirical normalized prediction error

stdP = std(P);

A = S.A;

A = A';

B = S.A;

B = B';

E = abs((alpha1\*A) + (alpha2\*B) + beta );

E = (P - E).^(2);

E = (sum(E))/999;

E = E / (stdP^2);

E = sqrt(E);

figure

plot3(A,B,P,'o','MarkerSize',2)

xlabel('A');

ylabel('B');

zlabel('Vout');

title(' plot for multiple variable model');

hold on

minX1 = min(A);

maxX1 = max(A);

minX2 = min(B);

maxX2 = max(B);

XA= linspace(minX1,maxX1, 1000);

XB= linspace(minX2,maxX2, 1000);

Y = (alpha1\*A) +(alpha2\*B) + beta;

plot3(XA, XB, Y);

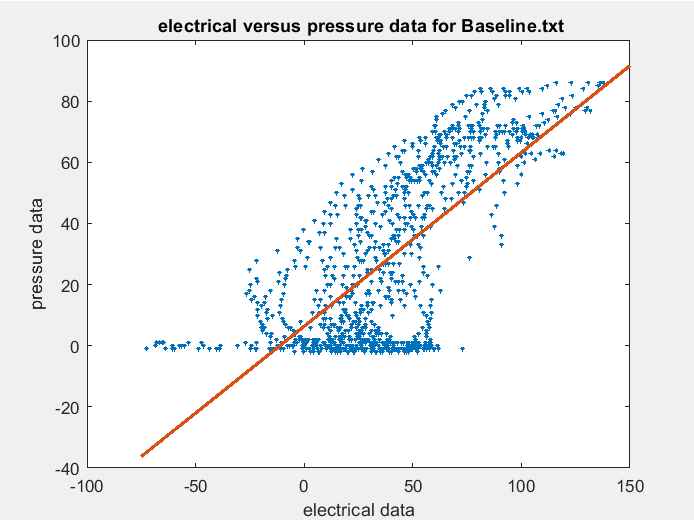
view(200,300)

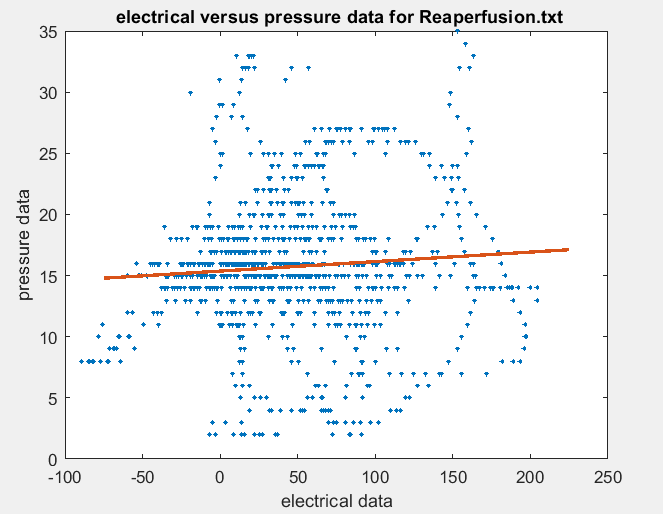
hold off

%%}

**APPENDIX B**

**Figures of all the plots**





**PLOT FOR PART 3**

