

APM 153 LECTURE FOURTEEN – Importing Data, Calculating a Least Squares Regression using Matrix Math, Using ezplot to Plot a Function. Algorithm for Least Squares Regression.

Importing Data from a .txt File

(1) Take a look at the data shown below. If we were to plot the two columns of numbers on a graph, what would the plot look like?

```
SITE #1. Little Sandy Creek
Stage Height (m) Discharge Rate (cms)
  10.0           10.0
  20.0           20.0
  30.0           30.0
  40.0           40.0
  50.0           50.0
  60.0           60.0
  70.0           70.0
  80.0           80.0
  90.0           90.0
```

(2) Now take a look at the next dataset. In what way would the plot of the data below resemble the plot of the data above in (1)?

```
SITE #2. Kanawha River
Stage Height (m) Discharge Rate (cms)
  11.2           1225.4
  12.3           2226.1
  13.4           3327.2
  14.5           4428.9
  15.6           5529.3
  16.7           5630.3
  17.8           5831.4
  18.9           6632.9
  22.0           7835.5
  26.9           10112.4
```

(3) The data in the two files above can be imported into Matlab by double clicking on the name of the data file, in this case, SITE1.DAT and SITE2.DAT

(4) Double clicking on the filename opens up the Matlab “import wizard”. (A “wizard” is a tool to help you do something.)

(5) Once we have imported the data, Matlab stores the data in a variable which, not surprisingly, is called **data**.

(6) In turn we can assign the two columns of data to the variables x and y using subscripts.

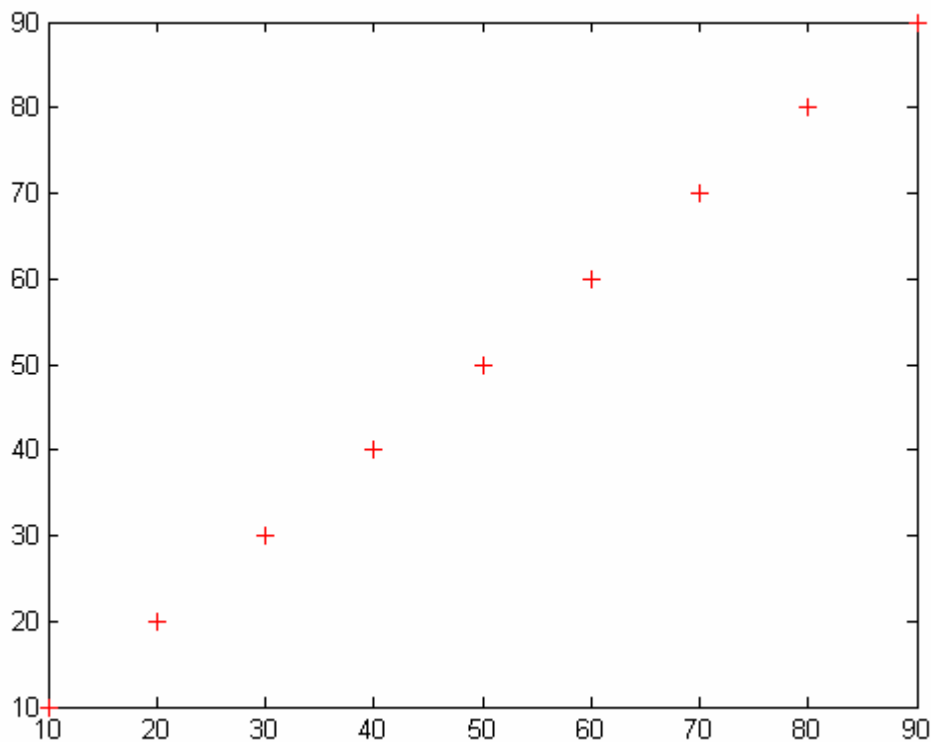
```
>> xvalues = data(:, 1)          and          >> yvalues = data(:, 2)
```

(7) Remember, when we use subscripts, the colon represents “every row”.

(8) Now that the data have been assigned to two separate variables, we can plot the data using the plot function. Let’s plot just the data points themselves as a series of + signs.

```
>> plot(xvalues, yvalues, '+' )    (Note the + sign in single quotes)
```

which gives us the following plot.



(9) Just looking at the graph, we can see that the data form a perfectly straight line. If we wanted to, we could draw a straight line right through the center of each data point. Such a line is called a regression line.

(10) This week in lab we will be calculating regression lines for three datafiles that you will download from the APM 153 website. (SITE1.DAT, SITE2.DAT and SITE3.DAT)

Calculating a Least Squares Regression Using Matrix Math

(11) To calculate a least squares regression for each set of data we will first need to import each datafile and assign the two columns of x and y data to separate variables.

(12) Next we will use the x values to create an **augmented matrix**.

(13) The augmented matrix will consist of two columns. The first column will be the x values themselves. The second column will consist entirely of **ones**.

(14) How long should our augmented matrix be? We can find out by using the **length** command.

```
>> length(xvalues)           which returns the values    9
```

(15) Next, create a matrix Aug_x with two columns of ones.

```
>> Aug_x = ones(9, 2)
```

(16) We could even combine the two steps.

```
>> Aug_x = ones(length(xvalues), 2)
```

(17) Next, we reassign the values in the first column of Aug_x to be the x values from SITE1.DAT.

```
>> Aug_x(:, 1) = xvalues           giving us,...
```

```
Aug_x =  $\begin{bmatrix} 10 & 1 \\ 20 & 1 \\ 30 & 1 \\ 40 & 1 \\ 50 & 1 \\ 60 & 1 \\ 70 & 1 \\ 80 & 1 \\ 90 & 1 \end{bmatrix}$ 
```

(18) Creating the augmented matrix is necessary if we want to use matrix math to solve for the least square regression. Why?

(19) Because solving this equation using matrix math involves using the inverse of a matrix. Can we take the inverse of the augmented matrix?

(20) However, if we multiply Aug_x by the transpose of Aug_x, we get a 2 X 2 square matrix. What is the transpose?

(21) The transpose of matrix is when we “flip” the matrix on its side so that the columns become the rows and the row become the columns. Therefore, the transpose of Aug_x would be,...

```
>> transAug_x      = transpose(Aug_x) =  $\begin{bmatrix} 10 & 20 & 30 & 40 & 50 & 60 & 70 & 80 & 90 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$ 
```

(22) Notice that the value in the first row, first column in the transposed matrix is the same as in the original matrix. We sometimes say that the transpose matrix “pivots” on this value.

(23) In Matlab, we can transpose a matrix using either the word transpose, or by using an apostrophes. (An apostrophe is just a single, single quote!)

```
>> transAug_x = Aug_x'
```

(24) Now we can solve for the least squares regression. Remember that the least squares regression is a straight line. What we will be solving for is the slope and y intercept of that line.

$$\text{slope_intercept} = \text{inv}(\text{Aug_x}' * \text{Aug_x}) * (\text{Aug_x}' * y)$$

Using ezplot to Plot the Least Squares Regression

(25) The ezplot command in Matlab plots a function $f(x)$ over a given range. In order to use ezplot, to plot our least squares regression line, we must set up the function f using symbolic language.

(26) First, we tell Matlab what symbols we will be using. In this case we only need one symbol.

(27) We type, ... `>> syms x` which creates the symbol **x**

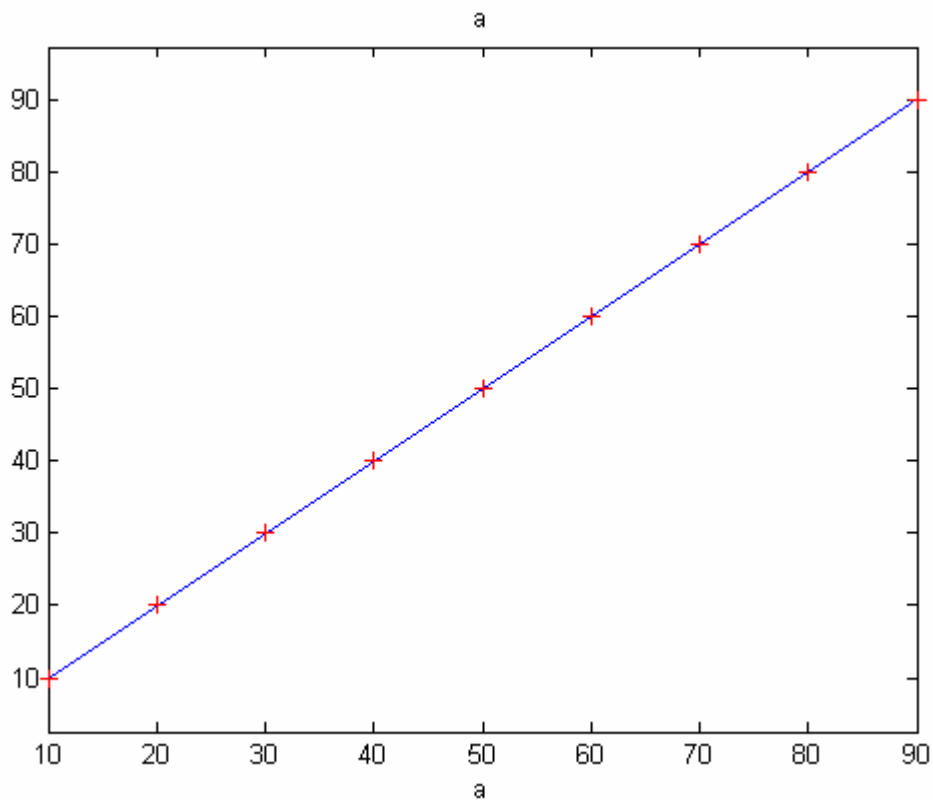
(28) Next, we set up our function *f*

```
>> f = intercept + slope * x
```

(29) Finally, we plot our function using `ezplot` for a range of values.

```
>> ezplot (f, [0, 100])
```

(30) If we plot the original *x* values and *y* values on top of our function we get, ...



(31) Wash, rinse, repeat.

(32) Question: Could this series of commands be put into a script file or function to automate the tasks?

