COEN 45, MATLAB Programming
Winter Quarter, 2011

Lab assignment #2

Regression line for test data
Jan. 18, 19, 20

The purpose of this lab is to give you practice in reading data from a flat file, doing computations that operate on whole arrays, and plotting.

A wind tunnel, as you probably know, is a test chamber where scale models of aircraft are subjected to high-velocity air flows. NASA Ames in Mt. View has a huge one. Here is wind-tunnel data for a lifting body in terms of flight-path angle and coefficient of lift.

degrees	Coefficient of lift
-4	-0.182
-2	-0.056
0	0.097
2	0.238
4	0.421
6	0.479
8	0.654
10	0.792
12	0.924
14	1.035
15	1.076
16	1.103
17	1.120
18	2.121
19	1.121
20	1.099
21	1.059

You can either type this data into a plain text file `wtunnel.dat` or download the file from the class web page, `www.gibson2.com/coen45`. After you have the data in a file, enter it into your workspace by typing `load wtunnel.dat`. Notice that it creates an array called `wtunnel`.

1. Put the first column of `wtunnel` into a variable `x` and the second into `y`. Set `n` equal to the number of data points (`help length`).

2. Make a scatter plot of your data. Use the `plot` function and create a plot having flight-path angle (`x`) on the horizontal axis and coefficient of lift (`y`) on the vertical axis. Instead of the default plot with lines connecting the points, make your plot with stars at each point (*) and no line. See `help plot` to find out how to do this. Also, use `fprintf` to report the number of data points found.

3. Examine your plot. Do any of the points look suspicious? Sometimes experiments produce data that is obviously wrong, due to instrumentation failures, mis-typed data, etc. Such

points are called "outliers" and we sometimes just throw them out. To discard an outlier, find its row number in the `wtunnel` array and remove it using the null value `[ ]` (as explained in class but not in your text). Suppose, for example, that you want to discard row 13. Use `input` to get the proper row number from the command window (e.g., `outlier = input('Index for outlier: ')` and use `fprintf` to show the value that is being dropped. Look at `wtunnel` in your command window to verify that the bad data point is gone. Then re-create the plot.

4. Next, create a regression line. This is a straight line that fits the data better than any other straight line. The straight line is written as $\hat{y} = \alpha + \beta x$ and the coefficients α and β are given by

$$\alpha = \frac{\sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i}{n \sum x_i^2 - (\sum x_i)^2}$$

$$\beta = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2}$$

   Each sum runs from $i = 1$ to $i = n$ where n is the number of data points.

   Use the `sum` command for the summations in the formula. Multiplications are element-by-element operations (text section 3.4) which means you must use `.*`, not `*`. To get clear on this, you might want to practice something like $\sum x_i^2$ in the command window. Note: do not use `.*` where `*` is adequate, e.g. when multiplying a scalar by a vector.

   There are easier ways of getting regression lines in MATLAB but the purpose of this lab is to get familiar with array operations in MATLAB, and gain a little plotting experience. So please program the formulas shown above.

5. Once you have your α and β, use them to plot the straight line that they represent. Do `hold on`, create `yhat` and then `plot(x,yhat,'r')`. (`hold on` adds the next plot to the existing figure without deleting any previous plot.)

6. Lastly, calculate the coefficient of regression R^2, which shows how nearly the data approximates a straight line. Its value is $+1$ if all points lie exactly on a straight line and 0 if there is no resemblance to a straight line.

$$R^2 = 1 - \frac{\sum (y - \hat{y})^2}{\sum (y - \bar{y})^2}$$

   where $\bar{y}$ is the mean or average value of y (see `help mean`) Show your R^2 value on your plot at some nice location `xt, yt`:

```
message = sprintf('R^2 = %5.3f',R2);
text(xt,yt,message);
```

Label your plots axes and make a title with your name and ID. After you have gotten everything to work make a script file and demonstrate it to the lab assistant. Submit a signed printout of the script file and a printout of your plots.