

## 8.3 Empirical Models

### *MATLAB* Quick Review Questions

*Introduction to Computational Science: Modeling and Simulation for the Sciences*

Angela B. Shiflet and George W. Shiflet

Wofford College

© 2006 by Princeton University Press

This file contains system-dependent Quick Review Questions and answers in *MATLAB* for Module 8.3 on "Empirical Models." Complete all code development in *MATLAB*.

### Linear Empirical Model

**Quick Review Question 1** List the expressions that are linear combinations of  $u$  and  $v$ .

- A.  $5u - 18v$       B.  $-18v + 5u$       C.  $7u$   
D.  $15uv$       E.  $u/5 + v/3$       F.  $5/u + 3/v$

**Quick Review Question 2** Consider the set of points  $pts = \{(0.2, 0.1), (0.4, 0.3), (0.3, 0.3), (0.3, 0.6)\}$ .

- Assign to  $xLst$  a vector of the  $x$ -coordinates.
- Assign to  $yLst$  a vector of the  $y$ -coordinates.
- Give the command to return a vector of the coefficients for the least-squares line that best fits the set of points.
- Assign to  $x$  a sequence of numbers from 0 to 0.6 varying by 0.1, and do not display the result.
- Assign to  $lineValues$  the corresponding list of  $y$ -values for the line of Part c.
- Graph the line.

### Non-Linear One-Term Model

**Quick Review Question 3** Suppose  $pts$  is an array of points, where the first column contains the  $x$ -coordinates and the second column contains the corresponding  $y$ -coordinates.

- Give the command to obtain the first column of  $pts$ .
- Select the operator to apply to the answer of Part a to return a row vector  
A.  $.$       B.  $*$       C.  $;$       D.  $?$   
E.  $'$       F.  $:$       G. nothing
- Give the command involving your answers from Parts a and b to assign to  $xLst$  the  $x$ -coordinates of  $pts$ .
- Select the symbol so that the result of the command to obtain the vector of  $x$ -coordinates is not displayed.  
A.  $.$       B.  $,$       C.  $;$       D.  $?$   
E.  $!$       F.  $:$       G. nothing

- e. Give the command to assign to  $yLst$  a row vector of  $y$ -coordinates from  $pts$  and not to display the result.
- f. Suppose instead of constructing  $xLst$  and  $yLst$ , we wish to read the data from *DanWoodEM.dat*, where each line contains an  $x$  and then a  $y$ -value. Give the command to read this data and to assign to  $pts$  the list of data points.
- g. Give the name of the function to plot these points.
- h. Give the commands to produce a plot similar to that of Figure 8.3.5. Assign the graphics to variable  $lp$ .
- i. Give the command to display a line from the first point, (1.309, 2.138), to the last, (1.68, 5.66). Assign the graphics to variable  $lne$ .
- j. Give the command between the display of  $lp$  from Part h and the display of  $lne$  from Part i so that both appear on the same graph.
- k. Give the command after the display of  $lne$  from Part i so that these graphs do remain for a subsequent figure.

### Solving for $y$ in a One-Term Model

**Quick Review Question 4** Suppose  $xLst$  and  $yLst$  are lists of  $x$  and  $y$  values, respectively. Give the command in an appropriate software system to assign to  $pts$  the list of ordered pairs of corresponding  $x$  and  $y$  values but not to display the result.

**Quick Review Question 5** Give the command in an appropriate software system to generate the plot in Figure 8.3.13, where  $pts$  is the list of ordered pairs for the data and the points are larger.

### Answers to Quick Review Question

1. The following are linear combinations of  $u$  and  $v$ :
  - A.  $5u - 18v = (5)u + (-18)v$
  - B.  $-18v + 5u = (-18)v + (5)u$
  - C.  $7u = (7)u + (0)v$
  - D.  $u/5 + v/3 = (1/5)u + (1/3)v$
2.
  - a. `xLst = [0.2, 0.4, 0.3, 0.3];`
  - b. `yLst = [0.1, 0.3, 0.3, 0.6];`
  - c. `lineCoeffs = polyfit(xLst, yLst, 1)`
  - d. `x = 0:0.1:0.6;`
  - e. `lineValues = polyval(lineCoeffs, x);`
  - f. `plot(x, lineValues);`
3.
  - a. `pts(:, 1)`
  - b. `E. '`, an apostrophe
  - c. `xLst = pts(:, 1)'`
  - d. `C. ;`
  - e. `yLst = pts(:, 2)';`

- f.** `pts = load('DanWoodEM.dat')`
  - g.** *plot*
  - h.** `lp = plot(xLst, yLst, 'o', 'MarkerSize', 12, ...  
          'MarkerFaceColor', 'k')  
      xlabel('x')  
      ylabel('y')`
  - i.** `lne = plot([1.309, 1.68], [2.138, 5.66]);`
  - j.** `hold on;`
- 4.** `pts = [xLst; yLst]';`
- 5.** `lp = plot(pts(:, 1)', pts(:, 2)', 'o', 'MarkerSize', 12, ...  
          'MarkerFaceColor', 'k')`

Alternatively, we can use column vectors for the  $x$ - and  $y$ - coordinates, as follows:

```
lp = plot(pts(:, 1), pts(:, 2), 'o', 'MarkerSize', 12, ...  
          'MarkerFaceColor', 'k')
```