Problem Set 03 · Plotting Data

Instructions:

1. Each problem in this problem set has a set of deliverables for you to submit. You are responsible for following the appropriate guidelines and instructions below. Create appropriately named files as instructed.

2. Save all files to your Purdue career account in a folder specific to PS03.

3. Compress all deliverables into one zip file named **PS03_*yourlogin*.zip**. Submit the zip file to the Blackboard drop box for PS03 before the due date. *REMEMBER*:
   - Only include deliverables. Do not include the problem document, blank templates, etc.
   - Only compress files using **.zip** format. No other compression format will be accepted.

Deliverables List

Item	Type	Deliverable
Problem 1: Matrix Manipulation II	Paired	PS03_matrix_magic_yourlogin1_yourlogin2.m PS03_matrix_magic_yourlogin1_yourlogin2_report.pdf
Problem 2: Manufacturing Learning Curves	Individual	PS03_labor_*yourlogin*.m PS03_labor_*yourlogin*_report.pdf All data files that are loaded into your m-file
Problem 3: Genetically Modified Corn	Individual	PS03_INcorn_*yourlogin*.m PS03_INcorn_*yourlogin*_report.pdf All data files that are loaded into your m-file

Plotting Tips

Creating Plots of Data

- When analyzing bivariate (two variables) data, you must determine which is the independent variable and which is the dependent variable.
- A common way to phrase a request for a plot is to say, "Plot variable 1 versus variable 2." Variable 1 refers to the y-axis variable; variable 2 refers to the x-axis variable.
- When formatting plots of data, data markers are used for each data point but no line is used to connect those points. A line implies a relationship between the independent and dependent variable is known or modeled.

Creating Plots of Models

- When plotting models, lines are used with no data markers. The points used to generate the plot are selected for convenience and do not refer to actual data.

- Models may be presented with the raw data plot on the same plot. The model is a line with no data markers. The raw data are shown with data markers and no connecting line.

Testing and Debugging Plot Code

- **Always close all figure windows before re-running your code.** Otherwise, your code will add or remove things from the existing plot displays. This will appear to be random acts that may or may not reflect the presence of any coding problems.

Problem 1: Matrix Manipulation II

Paired Programming

Learning Objectives

Below are learning objectives that may be used when assessing your work on this problem. Use the links to find the full evidence lists for each topic.

Variables	02.01 Apply MATLAB rules and "good" programming conventions to name variables
	02.04 Assign a matrix to a variable
Arrays	03.03 Copy an array from a larger array and assign it to a variable
	03.04 Concatenate arrays
	03.05 Replace elements of arrays
	03.06 Reference an array using appropriate indexing
Scripts	04.01 Create a script that adheres to programming standards
	04.02 Execute a script from the MATLAB Command Window
Text Display	05.01 Format text output for technical presentation
	05.02 Suppress output from a command once debugging is complete

Problem Background

This problem will introduce you to Paired Programming and will review Matrix Manipulation concepts. Be sure to follow the Paired Programming guidelines presented in class.

Document, Test, Debug, and Finalize Your Code

- Comment your code **while you are coding, not afterwards**. It is easy to forget what each line of code represents if you delay commenting, and waiting until the end to add comments increases the time you will spend on commenting.
- Re-save, run, and debug your code often, preferably after each new line or closely related 2-5 new lines of code are added. This allows you to identify the true location of problems more easily. MATLAB identifies the first line of code that fails, but the actual error could be on any previous line.
- Suppress printing of code that is functioning properly. Only formatted displays should be printed in the Command Window once your code is functional.

Problem Set 03 · Plotting Data

Problem Steps

1. Download the script **PS03_matrix_magic_template.m** file.
2. Open template and complete the header information.
   a. Add the problem set number, your name, and your section-team number.
   b. Declare your partner in the paired programming area.
   c. List any additional contributors who work with you and your paired partner.
3. Save your script with the name format required by the deliverables list.
4. Use MATLAB to learn what these built-in functions do: `zeros` and `sum`.
5. In the `INITIALIZATION` section of your script file, create matrices `A` and `vals`, using the steps described below.
   a. Use the function `zeros` to create a 4x4 matrix named `A`.

```
A =

     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
```

   b. Use MATLAB matrix creation commands to create a matrix, `vals`, that matches the matrix below.
   Note: The numbers are not sequential.

```
vals =

     1     3     2     4
     5     6     7     8
     9    10    11    12
    13    15    14    16
```

6. In the `COPY & CONCATENATE` section of your script file, perform the following:

Note: Do not hardcode assignments unless told to do so.

   a. Copy the center 2x2 matrix of `vals` and assign it to `M`.
   b. Copy from `vals` a 2-element row vector `[3 2]` and assign it to `C`.
   c. Copy from `vals` a 2-element row vector `[15 14]` and assign it to `D`
   d. Create 1x4 row vector `E` that concatenates `D` between the first and fourth elements in the first row of `vals` to create the vector `[1 15 14 4]`, and uses square brackets to complete the concatenation in one line of code.
   e. Create 1x4 row vector `F` that concatenates `C` between the first and fourth elements in the fourth row of `vals` to create the vector `[13 3 2 16]`, and uses square brackets to complete the concatenation in one line of code.

   Hint: See PS01, Problem 5 “Useful MATLAB Commands” in the green box for help with array indexing.

Problem Set 03 · Plotting Data

7. In the `REPLACE MATRIX ELEMENTS` section of your script file, perform the following:

   a. Use only `M`, `E`, and `F` to replace the first row of `A`, the fourth row of `A`, and the center 2x2 matrix of `A`. Matrix `A` should look like the matrix below once these replacements are complete.

```
A =

     1    15    14     4
     0     6     7     0
     0    10    11     0
    13     3     2    16
```

   b. Complete the following replacements **without hardcoding values**:

      - Replace the 0 directly below the 1 in matrix `A` with the 12 from matrix `vals`.
      - Replace the 0 directly above the 13 in matrix `A` with the 8 from matrix `vals`.
      - Replace the 0 directly below the 4 in matrix `A` with the 9 from matrix `vals`.
      - Replace the 0 directly above the 16 in matrix `A` with the 5 from matrix `vals`.

8. In the `FINAL MATRIX` section of your script file, perform the following:

   a. Create a vector X that contains the sums of the columns of A.

   b. Concatenate vector X to the bottom of matrix A to create a new matrix, G. Concatenation requires the use of square brackets.

   c. Create a vector Y that contains the sums of the rows of G.

   d. Concatenate vector Y to the right of matrix G to create a new matrix, H. Concatenation requires the use of square brackets.

   e. Replace the lower right corner value of H with the sum of the first four values on the diagonal from the upper left corner and moving toward the lower right corner.

9. In the `FORMATTED TEXT DISPLAY` section of your script file, use three **`fprintf`** statements to display your results as shown:

   Note: Do not hardcode the numerical values within your **`fprintf`** statements; use array indexing of H to identify the appropriate values of H to display.

```
After doing step 8.e, the value in the center of H is ___.


After doing step 8.e, the value in the upper left of H is ___,
and the value in the upper right of H is ___.


After doing step 8.e, the value in the lower left of H is ___,
and the value in lower right of H is ___.
```

10. Publish your script as **PS03_matrix_magic_*yourlogin1_yourlogin2*_report.pdf**.

Problem 2: Manufacturing Learning Curves

Individual Programming

Learning Objectives

Below are learning objectives that may be used when assessing your work on this problem. Learning objectives from past assignments may also be used to assess your work. Use the links to find the full evidence lists for each topic.

Scripts	04.00 Create and execute a script
Variables	02.00 Assign and manage variables
Arrays	03.00 Manipulate arrays (vectors or matrices)
Import Data	06.00 Import numeric data stored in .csv and .txt files
Plotting	07.01 Create an x-y plot from a single data set
	07.02 Create multiple plots in separate figure windows
	07.03 Create an x-y plot with multiple data sets in a single figure window
	07.05 Format plots for technical presentation
	07.06 Interpret the shape of x-y plots

Problem Setup

Image 1: Combine Harvester

Learning curves are used to estimate production costs and target labor force training in manufacturing. Take the production of a combine harvester (a piece of equipment used to harvest crops like corn and soybeans). Heavy-duty agricultural equipment production is complex and requires a lot of labor. The cost to produce each machine decreases over time due to learning. As experience is gained (or learning takes place) in manufacturing the equipment, the cost of manufacturing the equipment decreases. Some of the sources of cost improvement include product re-design to facilitate manufacturing, plant layout, and process improvements.

One indicator of cost is direct labor hours per unit manufactured. Direct labor hours per unit drop rapidly during production startup. As more units are manufactured, the cost reduction from one unit to the next becomes smaller and smaller. Reductions in cost can continue to occur for decades as learning continues.

As an industrial engineer, you want to improve the training program for a manufacturing company. You need to examine the learning curves for two of their production facilities that have a historical training record of direct labor hours per machine manufactured. The facilities have similar, but not identical, training programs. One facility is in Illinois; the other one is in Nebraska. Your teammate modeled the data for each facility and gave you the model equations.

Illinois plant: $L_{IL} = 61 \cdot n^{-0.5}$

Problem Set 03 · Plotting Data

Nebraska plant: $L_{NE} = 75 \cdot n^{-0.62}$

Where

n = the cumulative number of machines manufactured during the training period

L_{IL} = the direct labor hours at the Illinois plant (hours/unit)

L_{NE} = the direct labor hours at the Illinois plant (hours/unit)

Use the historical data provided in a file called **Data_direct_labor_costs.txt** and the model equations to complete the problem steps below.

Problem Steps

1. Download the **PS03_labor_template.m** file.
2. Save your script with the name format required by the deliverables list.
3. Open **Data_direct_labor_costs.txt** and review the information it contains.
4. In the `INITIALIZATION` section of your script, write the code to:
   a. Import the data using the appropriate MATLAB built-in function for the provided data file format.
   b. Copy each data column into its own variable.
5. In the `CALCULATIONS` section of your script, write the code to:
   a. Calculate the model's direct labor hours for the Illinois plant.
   b. Calculate the model's direct labor hours for the Nebraska plant.

   Note: Both calculations should be done over the same range of units as the data set.
6. In the `MODEL SMOOTHNESS CHECK` section of your script, write the code to:
   a. Create one figure that contains a plot of the Illinois model curve. This curve must have enough data points to create a smooth curve. If the curve is not smooth, then redo your model calculations until you get a smooth curve.
   b. Format the plot with a descriptive title, useful axes labels with units, and gridlines.
7. In the `FORMATTED FIGURE` section of your script, write the code to:
   a. Create one figure that contains a single plot of the original data for both plants.
   b. Overlay smooth models for each plant.
   c. Format the plot with a descriptive title, useful axes labels with units, and gridlines. Each model must be a different style and color. Each set of data points must be a different marker style but match the color of its model. Add a legend and label the models and data sets appropriately.
8. In the `ANALYSIS` section of your script, answer the questions below:

   Q1: Which plant had the most efficient training program? Justify your answer using the plot.

   Q2: Do you anticipate that the training has minimized the labor costs at either facility? Justify your answer using the plot.
9. Publish your script as a PDF and name it as required in the Deliverables List.

Image Reference: https://www.deere.com/en/harvesting/s-series-combines/

Problem 3: Genetically Engineered Corn

Individual Programming

Learning Objectives

Below are learning objectives that may be used when assessing your work on this problem. Learning objectives from past assignments may also be used to assess your work. Use the links to find the full evidence lists for each topic.

Scripts	04.00 Create and execute a script
Arrays	03.00 Manipulate arrays (vectors or matrices)
Import Data	06.00 Import numeric data stored in .csv and .txt files
Plotting	07.00 Create and evaluate x-y plots suitable for technical presentation
	07.01 Create an x-y plot from a single data set
	07.02 Create multiple plots in separate figure windows
	07.03 Create an x-y plot with multiple data sets in a single figure window
	07.04 Create multiple plots in a single figure window
	07.05 Format plots for technical presentation
	07.06 Interpret the shape of x-y plots

Problem Setup

Genetically modified (GM) corn was approved for use in the United States in 1995. The first GM corn were insect-resistant (Bt) varieties, which were quickly followed by herbicide-tolerant (Ht) varieties. Stacked-gene corn, which blends multiple traits in the same plant, followed later.

In Indiana, corn in a major agricultural crop. As an agricultural engineer, your task is to examine the adoption of GM corn in Indiana. You must write a MATLAB script that will generate several figures from which you can understand and answer questions about:

- The prevalence of GM corn in Indiana corn plantings
- The popularity of various types of GM corn
- The effect GM corn has had on Indiana's corn yields and crop value

The Unites States Department of Agriculture performs annual crop surveys and publishes the data. You have been provided with a data file, named **Data_Indiana_GMcorn.csv**, that contains several columns of relevant data related to GM corn planted and produced in Indiana.

Problem Steps

A. Initialize your script:

1. Open the **PS03_INcorn_template.m** file and complete the header information.

Problem Set 03 · Plotting Data

2. Save your script with the name format required by the deliverables list.
3. Open **Data_Indiana_GMcorn.csv** and review the information it contains.
4. In the `INITIALIZATION` section of your script, write the code to:
   a. Import the data file using the appropriate MATLAB built-in functions for the provided data file format.
   b. Copy each data column into separate variables.

B. Examine Indiana's GM corn planting trends:

5. In the `GM CORN VARIETIES PLOT` section of your script, write the code to:
   a. Create one figure (later referred to as Figure 1) that contains a single plot that shows percentages for insect-resistant, herbicide-resistant, gene stacked, and total GM plantings over time.
      - In general, you must plot measured data in a scatter plot for this course (see LO 07.01). Because this is time-series data and you need to examine the trends in the data, you can plot the data with markers or with markers connected by line segments.

   b. Format the plot for technical presentation. Ensure the legend does not cover the plotted data.
6. In the `ANALYSIS` section of your script, answer the following questions. Justify your answers using the plot:

   Q1: Explain how the prevalence of each type of GM corn planted in Indiana has changed between 2000 and 2016.

C. Examine corn yield and value trends

7. You need to use the data to determine what may explain the trend you see in Figure 1. You will examine the change in corn yields (i.e., the productivity of each farmed acre) and corn value (i.e., the monetary value of the state-wide corn crop). In the `CORN PRODUCTION PLOT` section of your script, write the code to:
   a. Create a second figure that contains a 2x2 arrangement of subplots, as follows:

Subplot A	Subplot C
Subplot B	Subplot D

   b. Subplot A must show the Indiana corn yield over time.
   c. Subplot B must show the value of Indiana corn over time.
   d. Subplot C must show how the percentage of GM plantings affects corn yield.
   e. Subplot D must show how the percentage of GM plantings affects Indiana's corn value.
   f. Format each subplot with a concise title, concise axes labels with units, and gridlines.
8. Examine the four subplots. Decide which of the four plots best explains the trend you see in Figure 1.
9. In the `ANALYSIS` section of your script, answer the following questions. Justify your answers using the plot:

   Q2: Indicate which subplot best explains the trend you see in Figure 1.

Problem Set 03 · Plotting Data

Q3: Justify how the subplot in your answer to Q2 explains the GM planting trend in Figure 1.

Q4: A farm company is considering a plan to transition to GM corn. Would you recommend that move? If so, what type of GM variety would you recommend? Base your recommendation on the plotted data.

D. Publish your script:

10. Publish your script as a PDF and name it as required in the Deliverables List.

References:
https://quickstats.nass.usda.gov/

Image Reference
http://indianapublicmedia.org/news/usda-predicts-record-corn-yield-soybeans-good-54024/