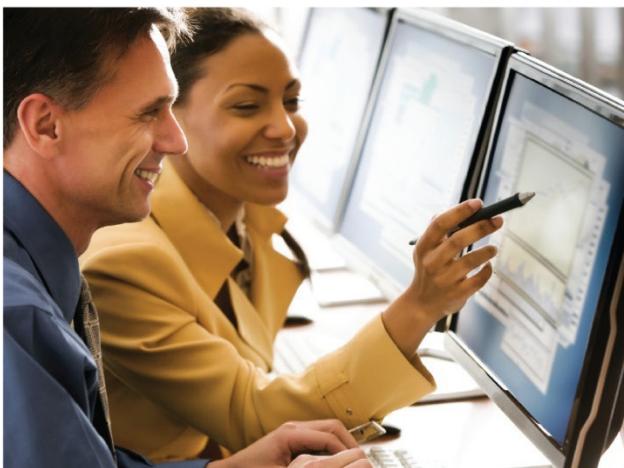


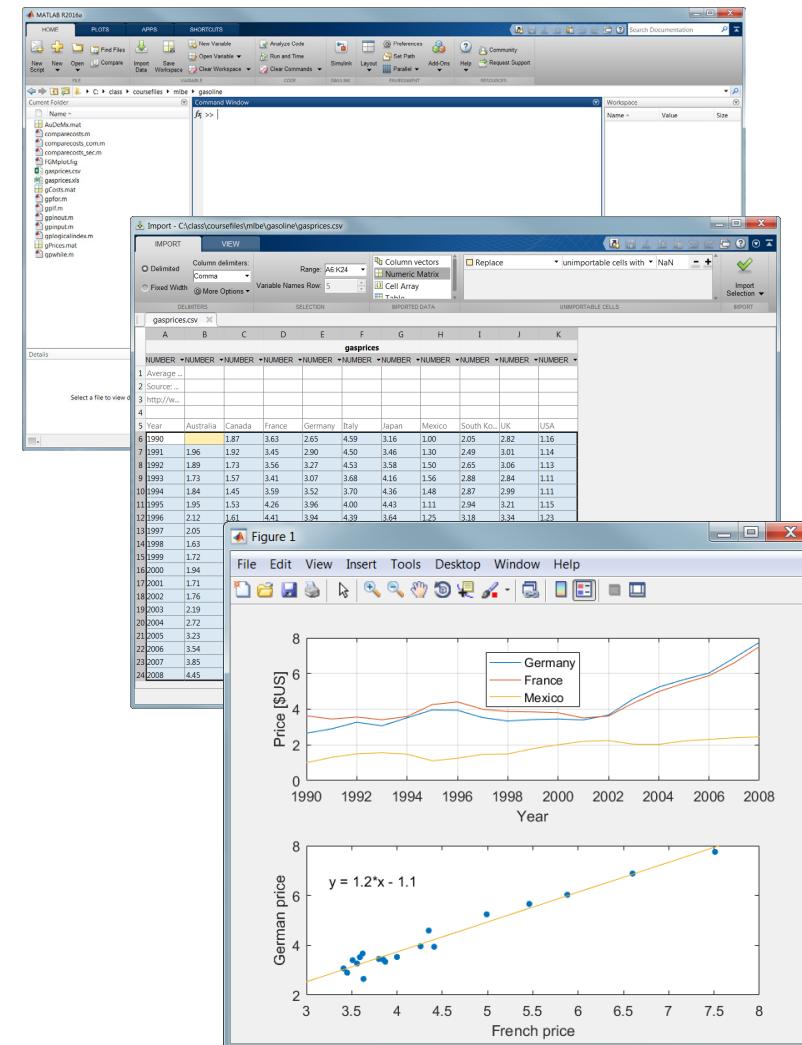
Working with the MATLAB® User Interface

MATLAB® Fundamentals for Aerospace Applications



Outline

- Navigating the interface
- Reading data from file
- Saving and loading variables
- Creating custom plots
- Exporting graphics for use in other applications



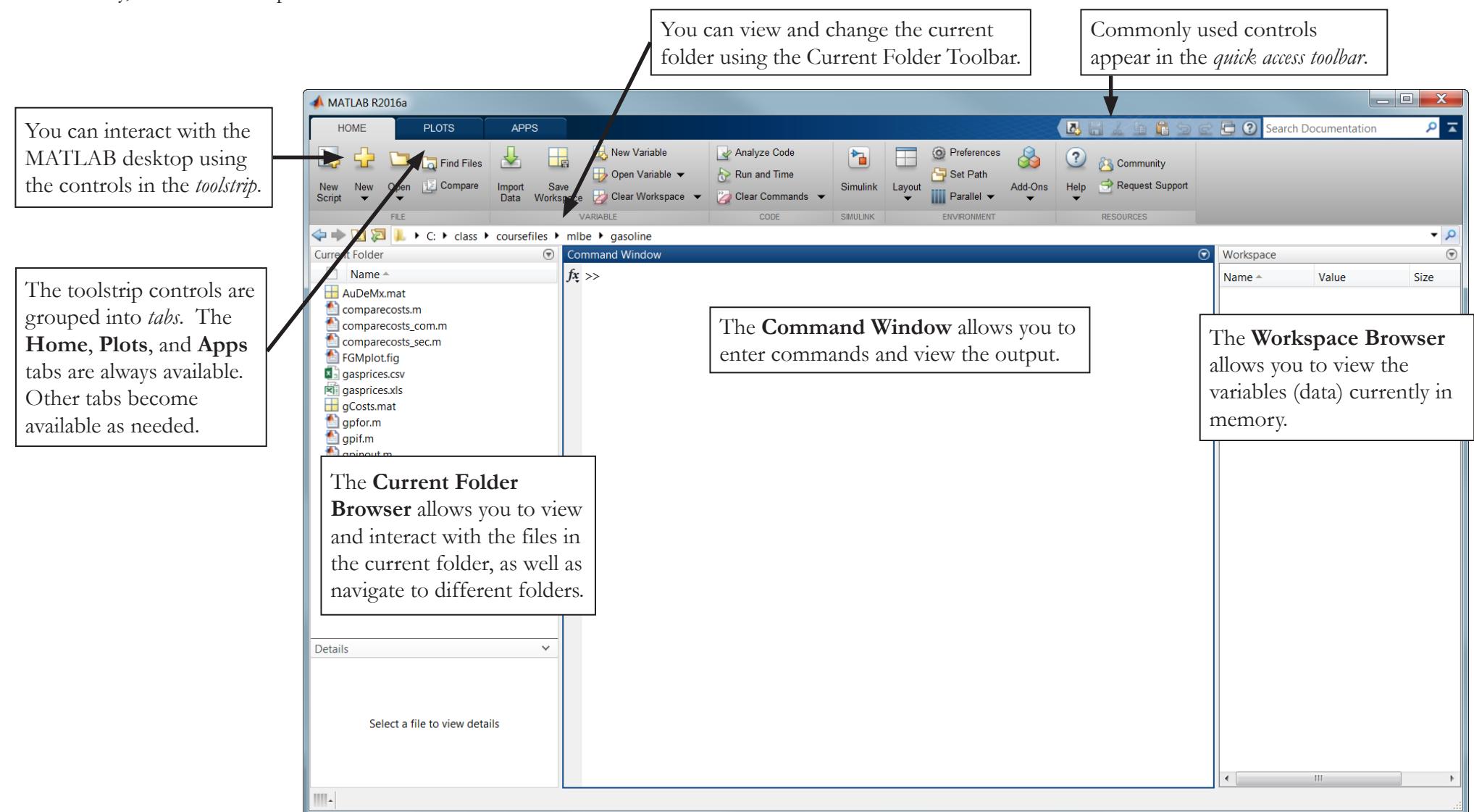
Chapter Learning Outcomes

The attendee will be able to:

- Identify the core components of the MATLAB® desktop environment and explain their purpose.
- Interactively import data into the MATLAB environment.
- Examine data variables using the Variable Editor.
- Create and customize data plots using the plot tools interface.
- Save and load MATLAB variables to and from disk interactively.

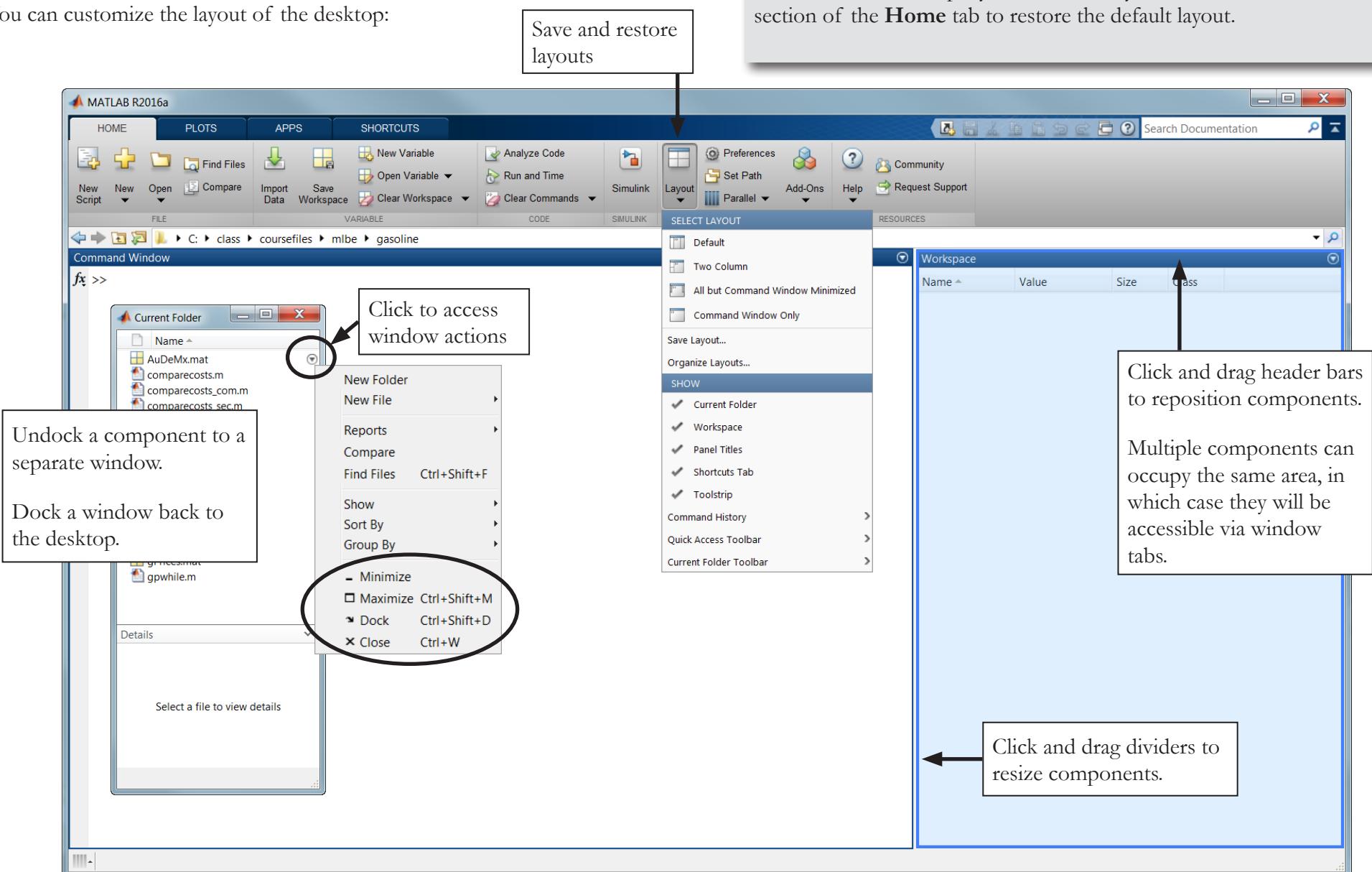
The MATLAB® Desktop

By default, MATLAB displays a desktop interface divided into three core components: the Command Window, the Workspace Browser, and the Current Folder Browser. Other components and windows are automatically opened as necessary, such as when a plot is created.



Customizing the Desktop

You can customize the layout of the desktop:



Try

Customize the desktop layout. Use the Layout menu in the **Environment** section of the **Home** tab to restore the default layout.

Course Example:

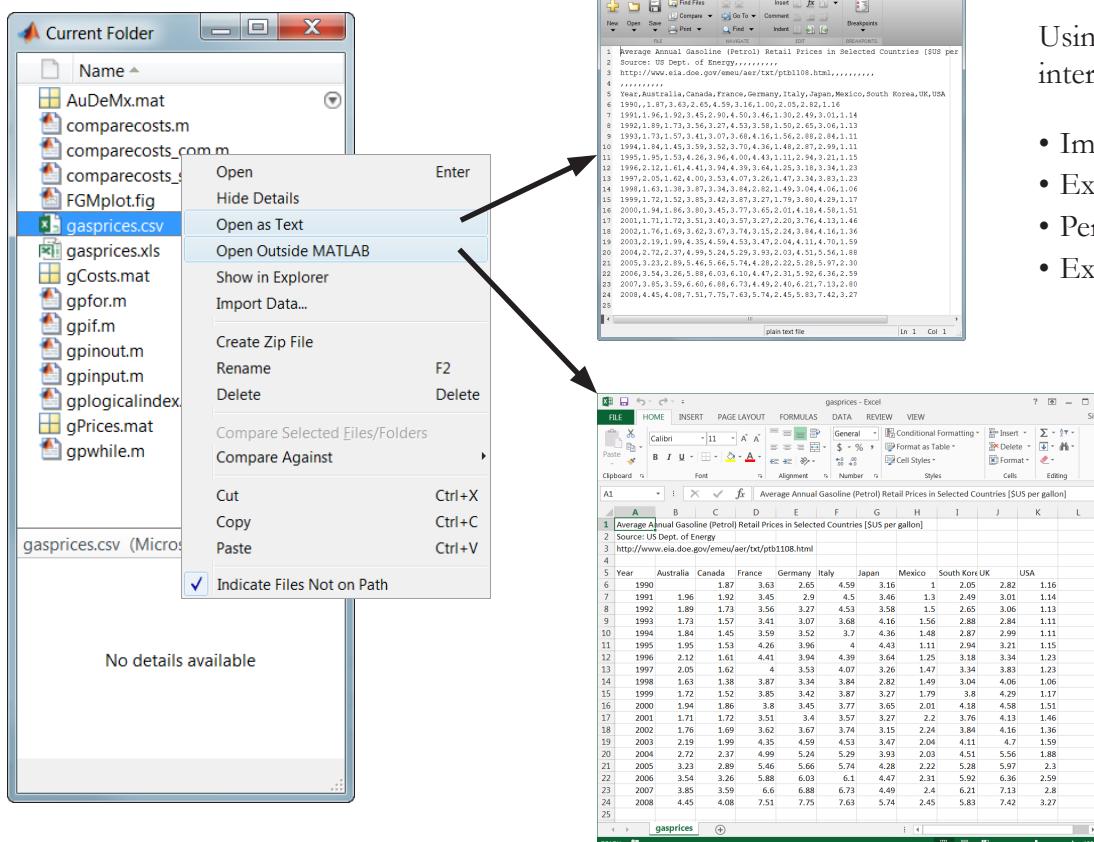
Gasoline Price Data

The file `gasprices.csv` contains historical data on annual retail gasoline prices, in U.S. dollars per gallon, from 10 countries from 1990 to 2008.

Updates can be downloaded from the U.S. Department of Energy at

<http://www.eia.doe.gov>

The file is ASCII text (as opposed to a binary file containing non-ASCII characters), and all of the data values following the initial text header are separated by a comma delimiter.



Try

1. Change your current directory to:
`C:\class\coursefiles\mlbe\gasoline\`
2. Find `gasprices.csv` in the Current Folder browser.
3. Right-click the file name, and choose **Open as Text**.
4. Right-click again and choose **Open Outside MATLAB**.

The file `gasprices.xlsx` contains the exact same data in Microsoft® Excel® format.

Using this data, this chapter illustrates how to use the MATLAB environment interactively to:

- Import data into MATLAB.
- Extract and save portions of data for later use.
- Perform some basic analysis and visualization.
- Export the results to another application.

Interactive Importing

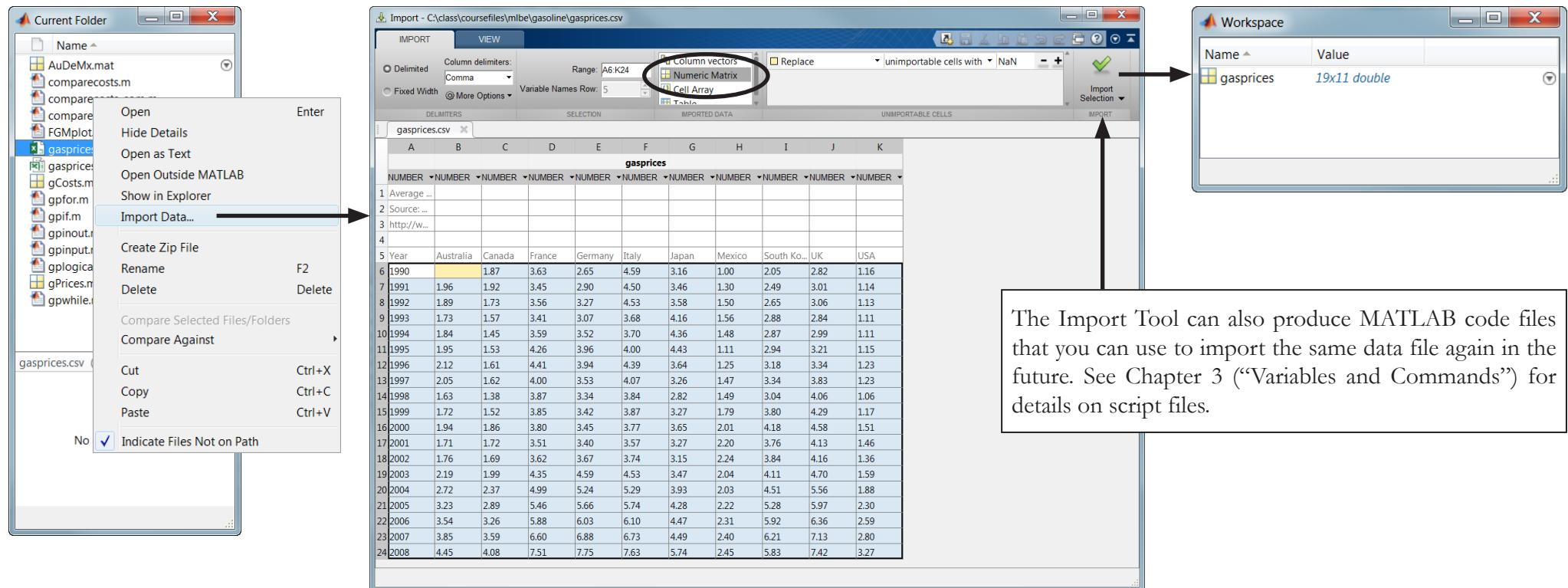
You can interactively import data into MATLAB from many common file formats by clicking the **Import Data** button in the **Variable** section of the **Home** tab of the toolbar, or by right-clicking the file in the Current Folder browser and selecting **Import Data** from the context menu.

Importing a file where the data has a predefined structure (such as `.wav` or `.png`) will open the Import Wizard, which automatically determines the format of the file, and shows a preview of the data and a suggested breakdown into MATLAB variables.

Try

1. Double-click `gasprices.xlsx` in the Current Folder browser.
2. Use the Import Tool to import all the gasoline prices as one matrix.

Importing a delimited text file or Microsoft Excel file in the Current Folder browser opens the Import Tool. The Import Tool will automatically try to identify and select the data in the file, and select an appropriate output format.

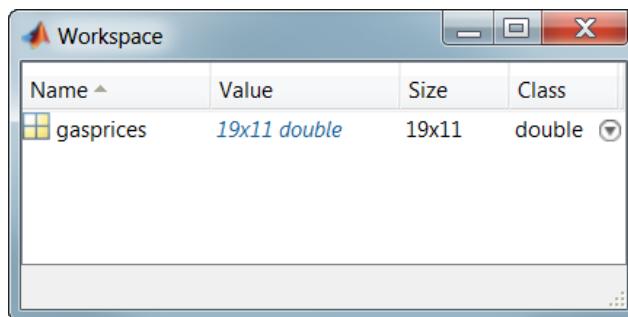


The Import Tool can also produce MATLAB code files that you can use to import the same data file again in the future. See Chapter 3 (“Variables and Commands”) for details on script files.

Variables in the Base Workspace

Data imported using the Import Wizard or the Import Tool is stored in the form of MATLAB *variables* in the *base workspace* (MATLAB memory).

The contents of the base workspace are shown in the Workspace browser window. The browser displays the names of the variables currently in memory and (optionally) information about the variables.



Unless you specify a different name, the imported data is stored in a variable with the same name as the file (without the extension). You can rename a variable by right-clicking it in the Workspace browser and choosing **Rename** from the context menu.

Right-clicking on the **Name** button at the top of the Workspace browser displays a context menu from which you can choose which information to display about the variables.

Try

Change the properties displayed in the Workspace browser to show the size and class of the variable `gasprices`.

Data is stored in MATLAB variables of various *classes* (types). Different classes store different kinds of information. The Import Wizard and Import Tool automatically choose appropriate classes for the variables that will store your data.

Double arrays are the default type for numerical data in MATLAB. “Array” means a variable with multiple values, stored in a regular layout, such as the table of gas price data. “Double” refers to “double precision,” which is a standard format for storing *floating-point* numbers – i.e., numbers that generally may have a fractional part, such as 3.14159.

The Variable Editor

You can view and edit the contents of variables in the base workspace using the *Variable Editor*. Double-clicking the data type icon next to the variable name in the Workspace browser opens the Variable Editor in a new window on the MATLAB desktop, with the contents of the variable displayed.

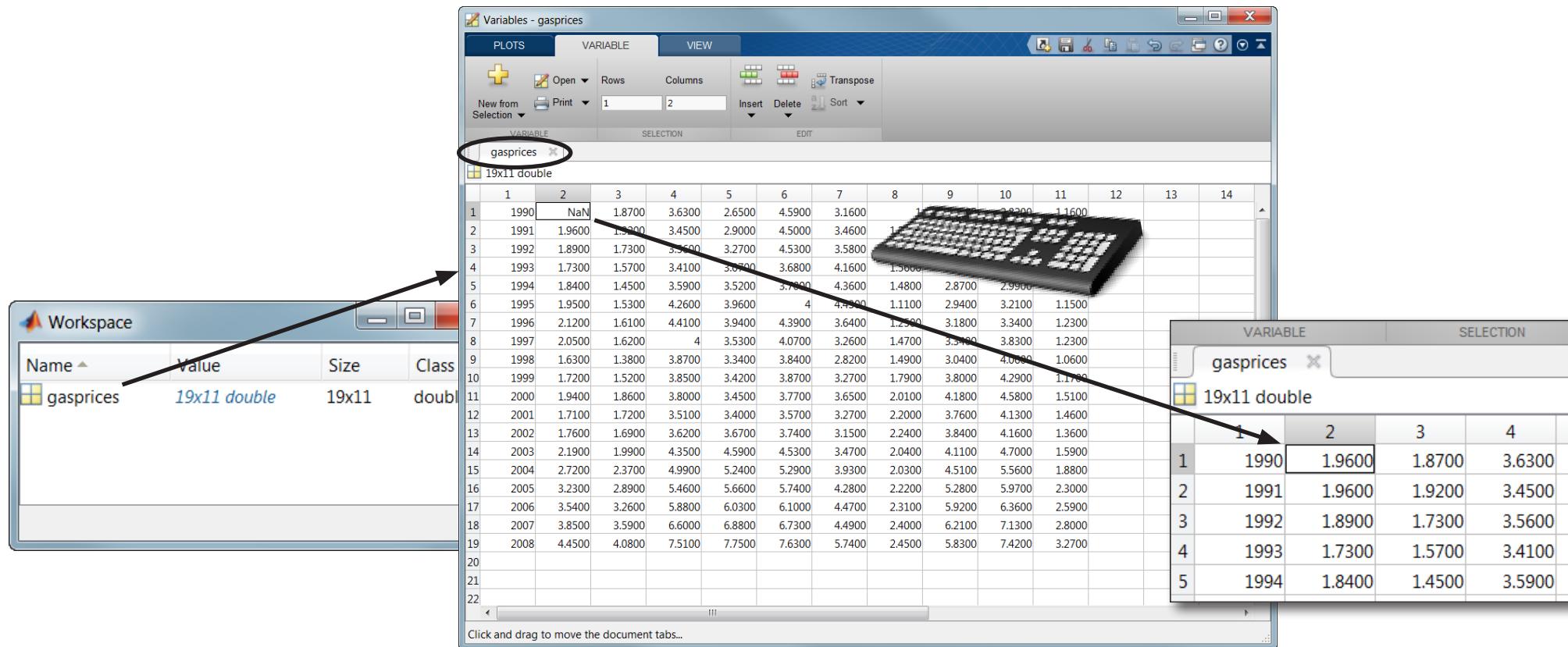
When multiple variables are open in the Variable Editor, you can switch the view from one variable to another using the tabs under the toolbar. Alternatively, variables can be tiled for comparison, using the options on the **Tile** section of the **View** tab.

Try

Open *gasprices* in the Variable Editor. Change the missing value (NaN) to 1.96.

To edit data in a variable, simply select the portion of the data you want to edit and type in new values. When you click away from your changes, the new data values are stored.

Other editing options such as sorting the data according to a given column or deleting rows or columns are available from the **Edit** section of the **Variable** tab.



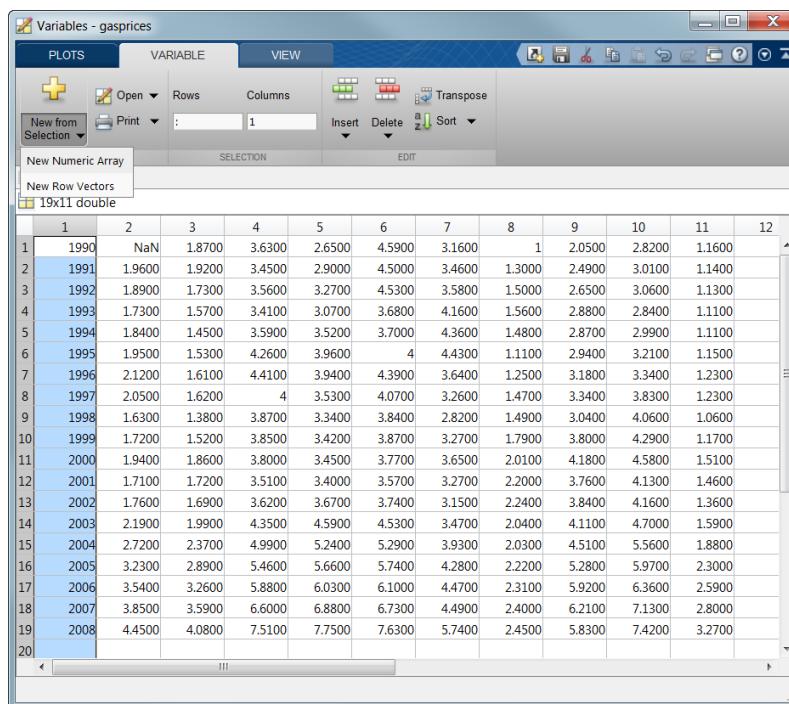
New Variables

You can also use the Variable Editor to create new variables in the base workspace by extracting data from other variables.

The variable `gasprices` is a 19-by-11 array of values, with each column representing a different variable (country) in the data, and each row representing a different (annual) observation. This format – observations \times variables – is typical of how multivariate data is stored in a single MATLAB array. You can use the Variable Editor to extract individual variables (columns) giving gas price data for individual countries.

To do this:

1. Click the numbered column header for the variable you want to select.
2. Click **New from Selection** \rightarrow **New Numeric Array** in the **Variable** section of the **Variable** tab.

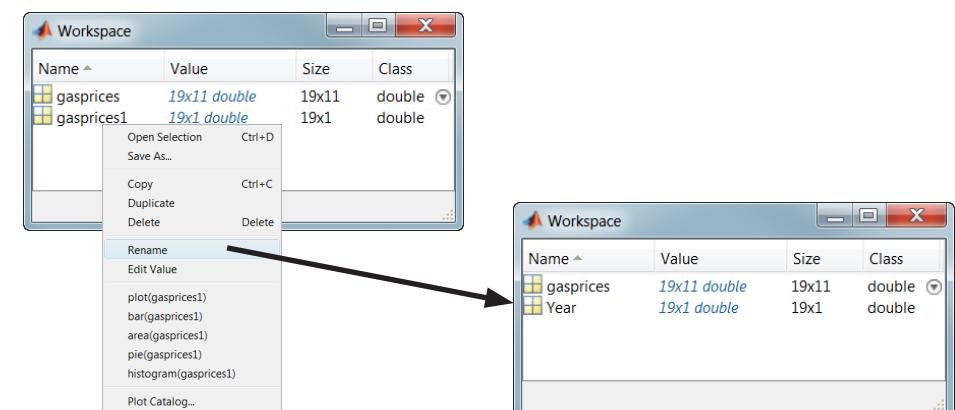


Try

1. In the Variable Editor, select the first column of `gasprices`, and create a new array from this selection.
2. In the Workspace browser, rename the variable `Year`.
3. Create separate variables for France, Germany, and Mexico (columns 4, 5, and 8, respectively) by making a multiple-column selection and creating new column vectors.

A variable with a default name (in this case `gasprices1`) is created in the base workspace. To rename it:

1. Right-click it in the Workspace browser.
2. Choose **Rename** from the context menu.



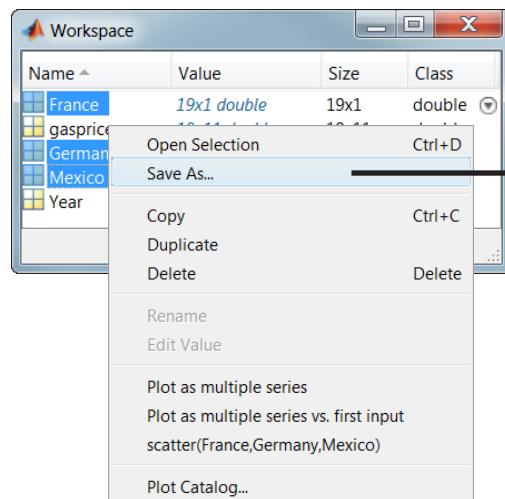
You can **Shift**-click and **Ctrl**-click to select multiple portions of data. When the selected data spans multiple columns, you can create a single (matrix) variable from the selection by clicking **New from Selection** \rightarrow **New Numeric Array**, or multiple (single-column) variables by clicking **New from Selection** \rightarrow **New Column Vectors**.

Saving and Loading Variables

MATLAB variables remain in the workspace until they are explicitly cleared (for example, by selecting them in the Workspace browser and pressing the **Delete** key) or MATLAB is closed. You can save variables in the current workspace to disk in a binary MATLAB format known as a MAT-file. The variables stored in this file can then be loaded into the workspace later. This provides a quick and easy way to save and load data for use in MATLAB.

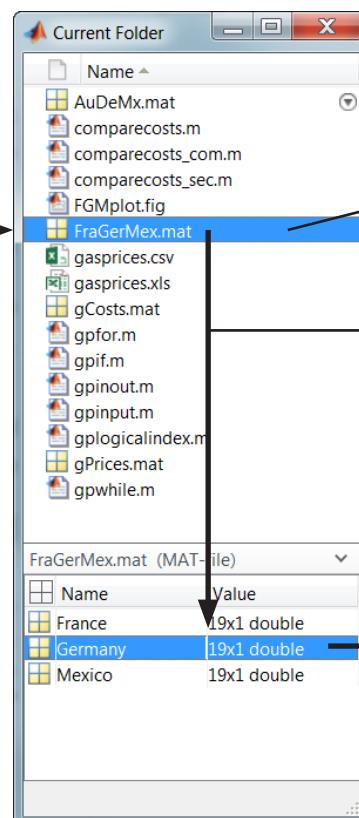
To save variables:

1. Select the variables in the Workspace browser.
2. Right-click on any of the selected variables to bring up a context menu.
3. Choose **Save As...**



The variables will be saved in a MAT-file (.mat extension).

Note that MAT-files save only the data, not the commands or process that created them.



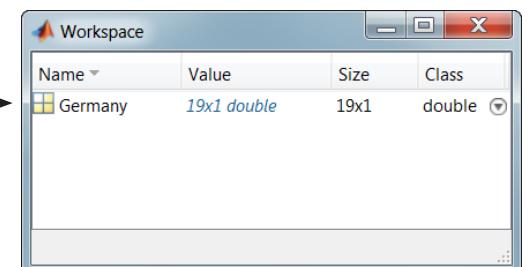
Try

1. Select the variables France, Germany, and Mexico in the Workspace browser (using **Shift**-click or **Ctrl**-click).
2. Save them to a MAT-file called **FraGerMex.mat**.
3. Select France and Mexico in the Workspace browser and press the **Delete** key to clear them from the workspace.
4. Double-click **FraGerMex.mat** in the Current Folder browser.

You can load the variables contained in a MAT-file by double-clicking the file icon in the Current Folder browser.

When a MAT-file is selected in the Current Folder browser, the preview pane shows the variables contained in the file, in a view similar to that of the Workspace browser.

You can load individual variables by dragging them from the preview pane to the Workspace browser.



Plotting the Data

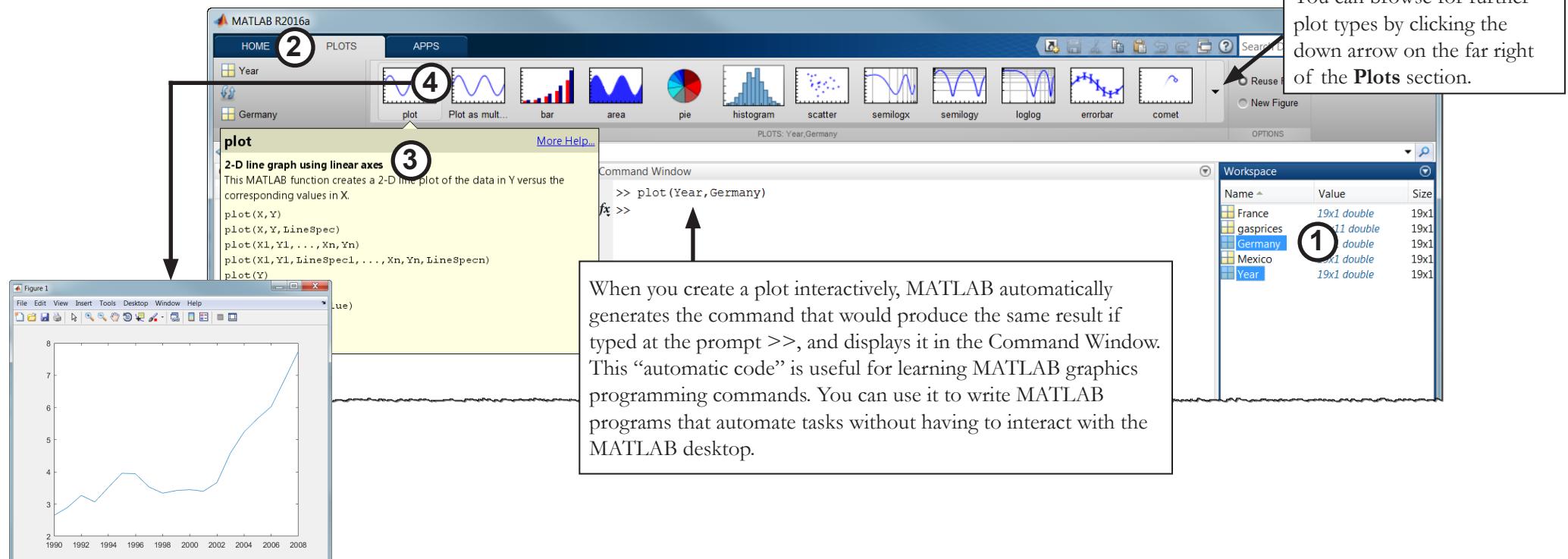
The first step in data analysis is often to create appropriate visualizations of the data. To make a plot from variables in the base workspace:

1. Select the variables in the Workspace browser.
2. Select the **Plots** tab in the toolbar. Some common plot types, suitable for the selected data, are listed.
3. Hover over any of the plot buttons to bring up more information on that type of plot.
4. Click on the desired plot icon to make the plot.

Try

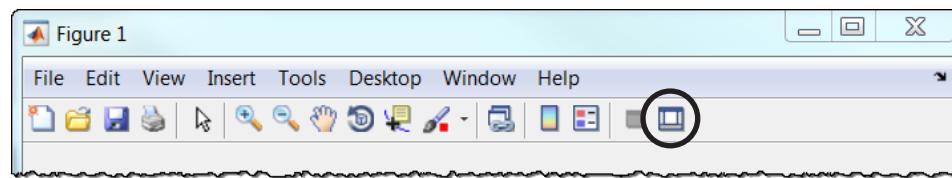
1. Plot the Germany data as a function of Year.
2. Plot France and Germany together as functions of Year by:
 - 1) Selecting Year.
 - 2) Control-clicking to select France and Germany.
 - 3) Choosing the second plot option from the **Plots** tab (“Plots the second and later series against the first series on the same plot”).

You can plot subsets of your data by first selecting a portion of a variable displayed in the Variable Editor.



Plot Tools

You can format the plot so that it conveys exactly the information you want to highlight. The Show Plot Tools button at the top of any MATLAB figure window opens up a variety of tools around the figure to help you format your plot with a high level of control.



The Figure Palette pane to the left of the figure allows you to

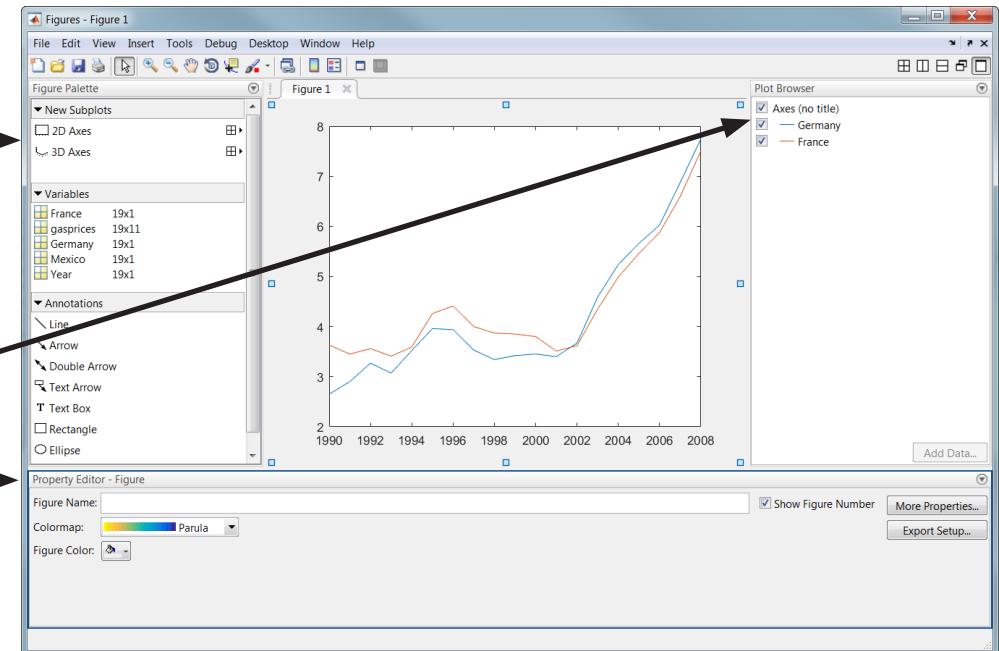
- Add axes to create *subplots* within the figure window.
- Drag and drop variables from the base workspace into any axes.
- Add a variety of annotations to a plot.

The Plot Browser pane to the right of the figure allows you to select or deselect individual axes and plots for editing

When a plot element is selected in the Plot Browser, its individual Property Editor becomes available at the bottom of the figure. The Property Editor shows some of the common properties of that particular plot element, and allows you to change their values interactively. You can view and edit a complete list of the low-level graphics properties of a plot in the Property Inspector by clicking the **More Properties** button.

Try

Open the plot tools interface on your plot of the gasoline price data.



Multiple Plots

Multiple data sets can be plotted in the same figure window either overlaid on a single set of axes or by dividing the figure window into multiple axes.

If you need subplots within the figure to display new variables:

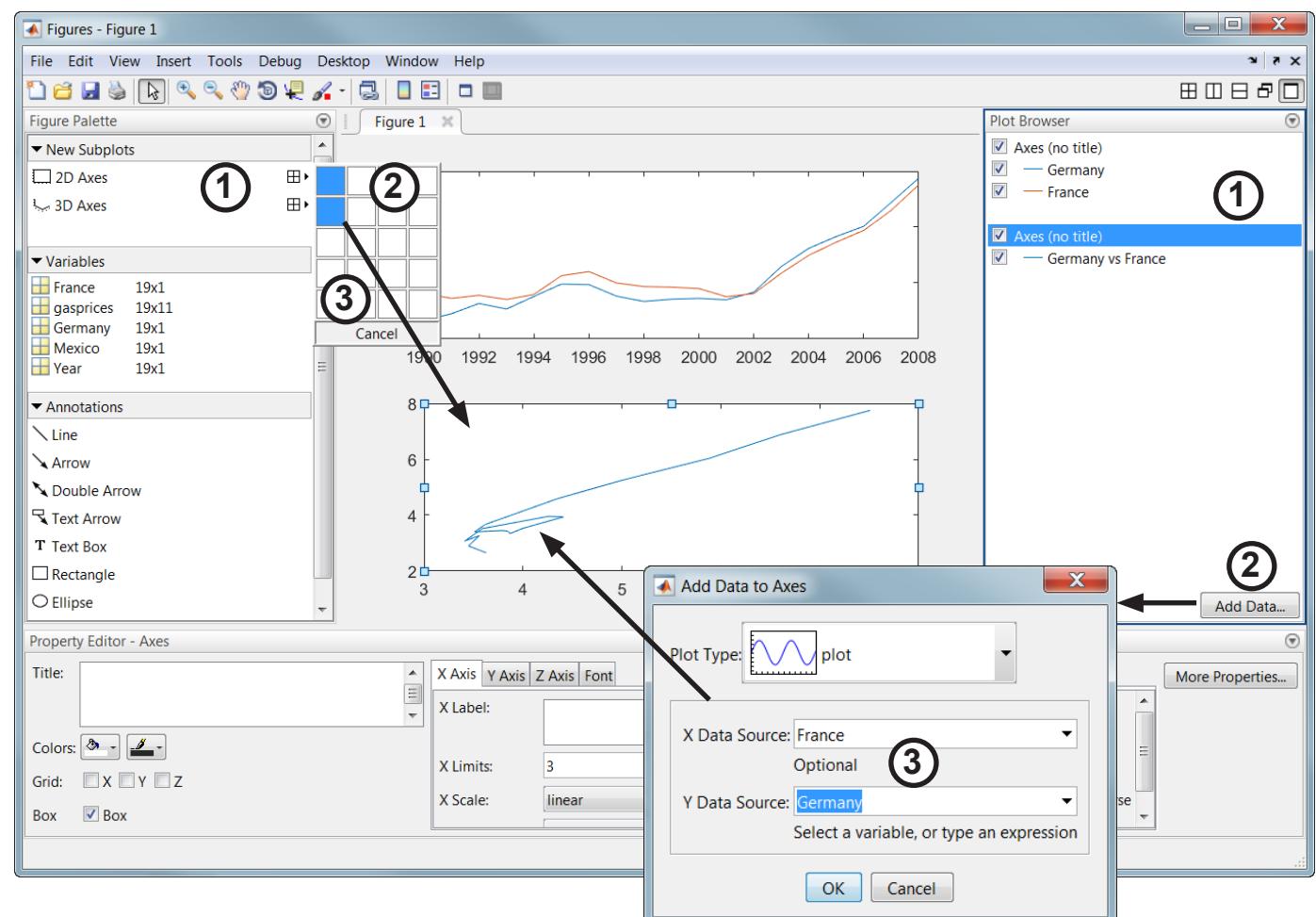
1. Click the **2D Axes** button in the New Subplots area of the Figure Palette.
2. Move your mouse to highlight the array of subplots you would like to see in the figure.
3. Click once to add the new axes to the figure.

To add data to a plot:

1. Select the axes you want to use.
2. Click the **Add Data** button at the bottom of the Plot Browser to the right of the figure.
3. In the dialog that opens, choose the **X Data Source** and **Y Data Source** for the new plot. Pop-up menus allow you to choose workspace variables for either data source, but you can also enter arbitrary MATLAB expressions.

Try

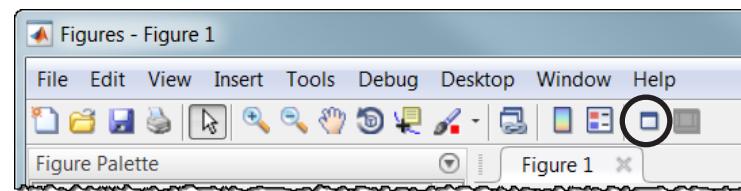
1. Add the Mexico data to the plot of the France and Germany data.
2. Add a new subplot below the current plot.
3. Plot Germany against France on the new subplot.



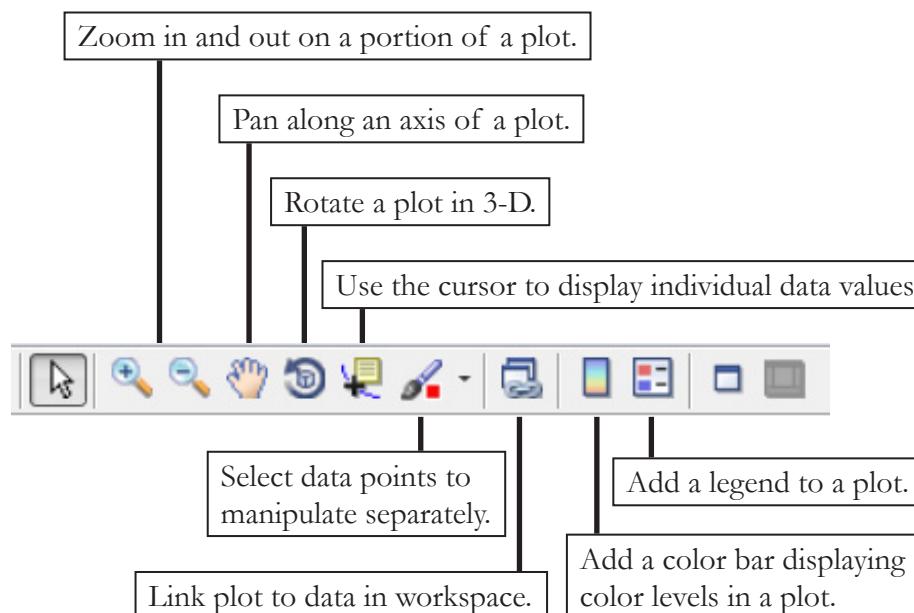
Formatting the Plot

Use the Property Editor to control the properties of individual graphics objects within a plot. The available properties shown in the Property Editor depend on what plot element is currently selected. For an axes object, you can specify a title, axis labels, tick locations and labels, grids, etc. For a line object, you can specify the style, color, and size of the markers and lines.

When plotting and formatting are complete, hide the Plot Tools by clicking the Hide Plot Tools button at the top of the figure.

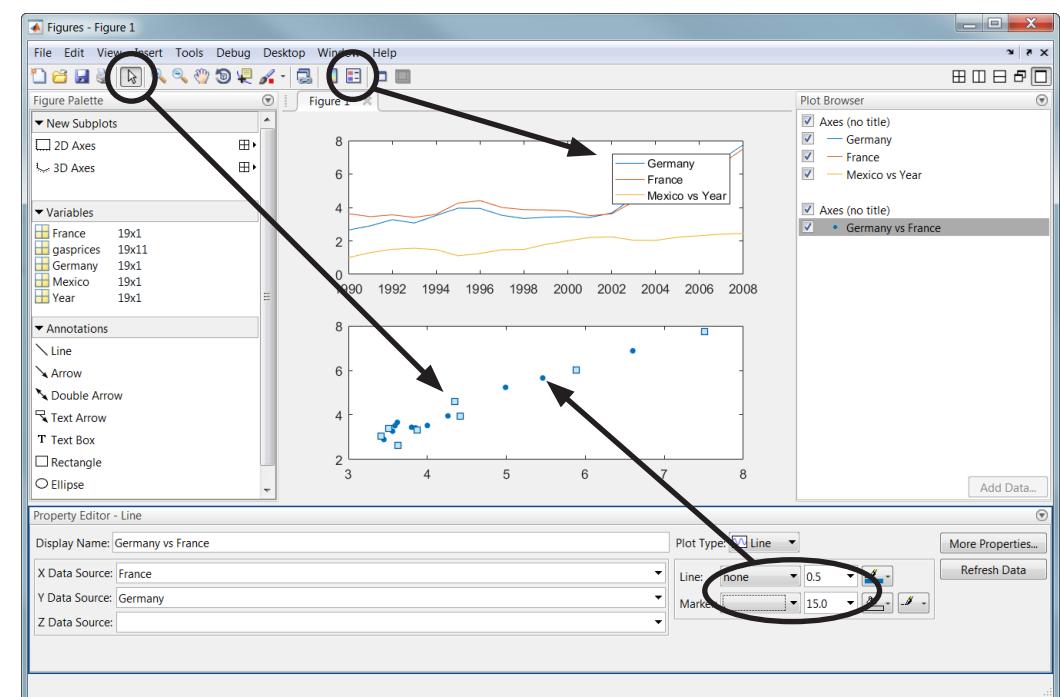


In addition to the Plot Tools, the Figure toolbar at the top of the figure offers many more viewing and formatting options:



Try

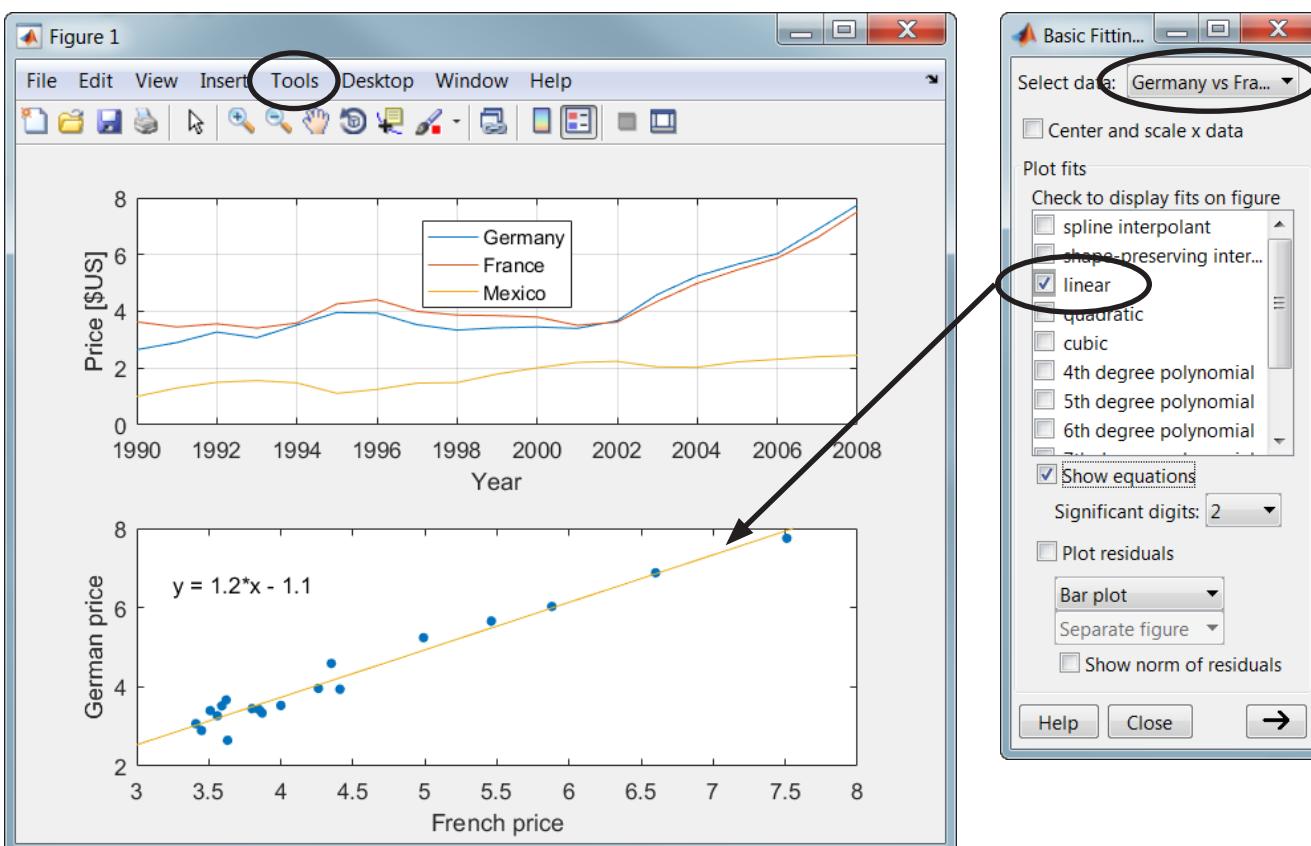
1. Make the France/Germany plot a scatter plot by adding point markers and removing the line.
2. Change the marker properties on the scatter plot.
3. Add a legend and a grid to the upper plot. Edit the text in the legend to provide informative labels.



Basic Fitting

You can interactively apply MATLAB functions for polynomial fitting and vector data interpolation to your data using the Basic Fitting Tool.

After plotting your data, choose **Tools** → **Basic Fitting** from the menu at the top of the figure window.



Try

Add a best-fit line to the scatter plot.

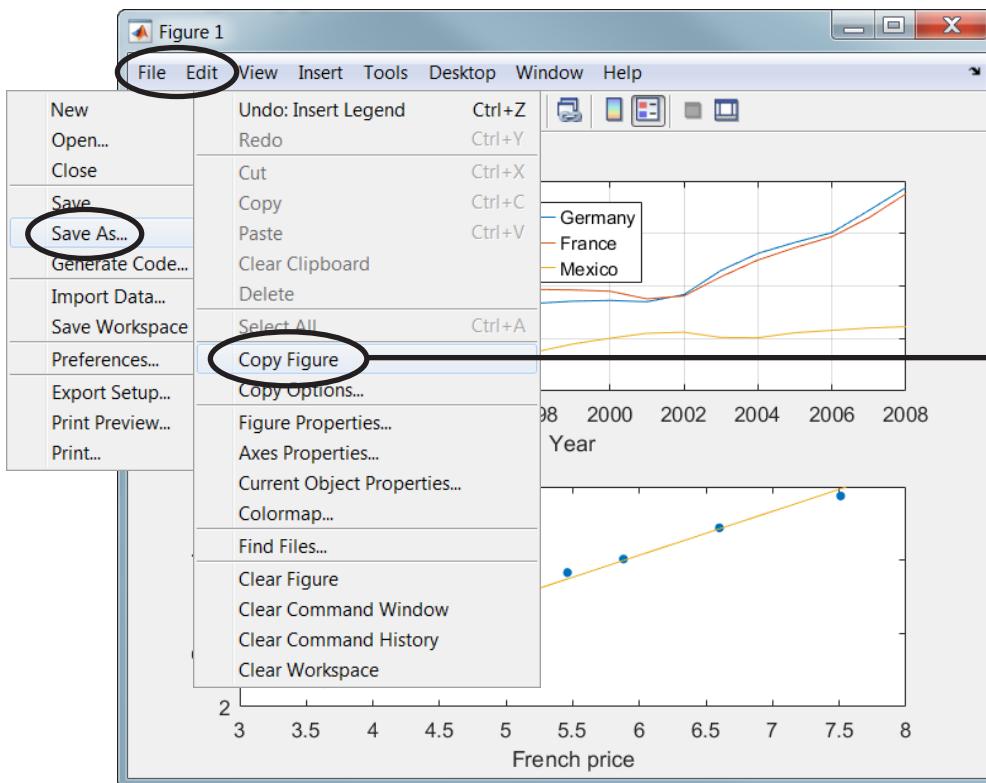
This opens the first panel of the Basic Fitting Tool. Additional panels appear when you click the next button (→) in the lower-right corner.

The Basic Fitting Tool allows you to fit various interpolants and polynomials of degree ≤ 10 . It will plot residuals and compute their norm, and evaluate the fit over arbitrary input vectors x . You can save coefficients of polynomial fits (not interpolants), and values interpolated or extrapolated from the fit, to the workspace.

Exporting to Another Application

You can save, copy, and export MATLAB figures to applications such as word processors, presentation software, and Web pages.

You can save a figure by selecting **File → Save As** from the menus at the top of the figure window. By default, figures are saved as .fig files, which is a binary format specific to MATLAB. These files contain all the information about a figure, including the data, property settings, annotations, etc. You can therefore open figures saved in this format and continue working with them in MATLAB.

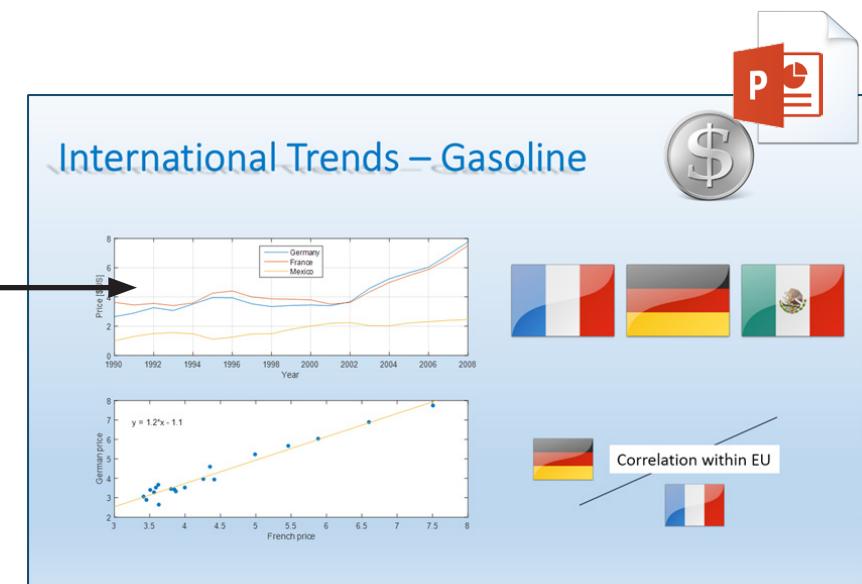


Try

Export your formatted figure to another application such as Word® or PowerPoint®.

To save to a graphical format appropriate for other applications (such as .jpg or .png), use the **Save as Type** pop-up menu in the dialog and select a format.

On a Windows platform, you can also copy a figure to the Windows clipboard. Choose **Edit → Copy Figure** from the menus at the top of a MATLAB figure. When it is copied to the clipboard, you can paste the figure into many applications.



Shortcuts

When you are finished with a particular task in MATLAB, you may want to clean up the desktop before moving on to another task. Some useful commands are

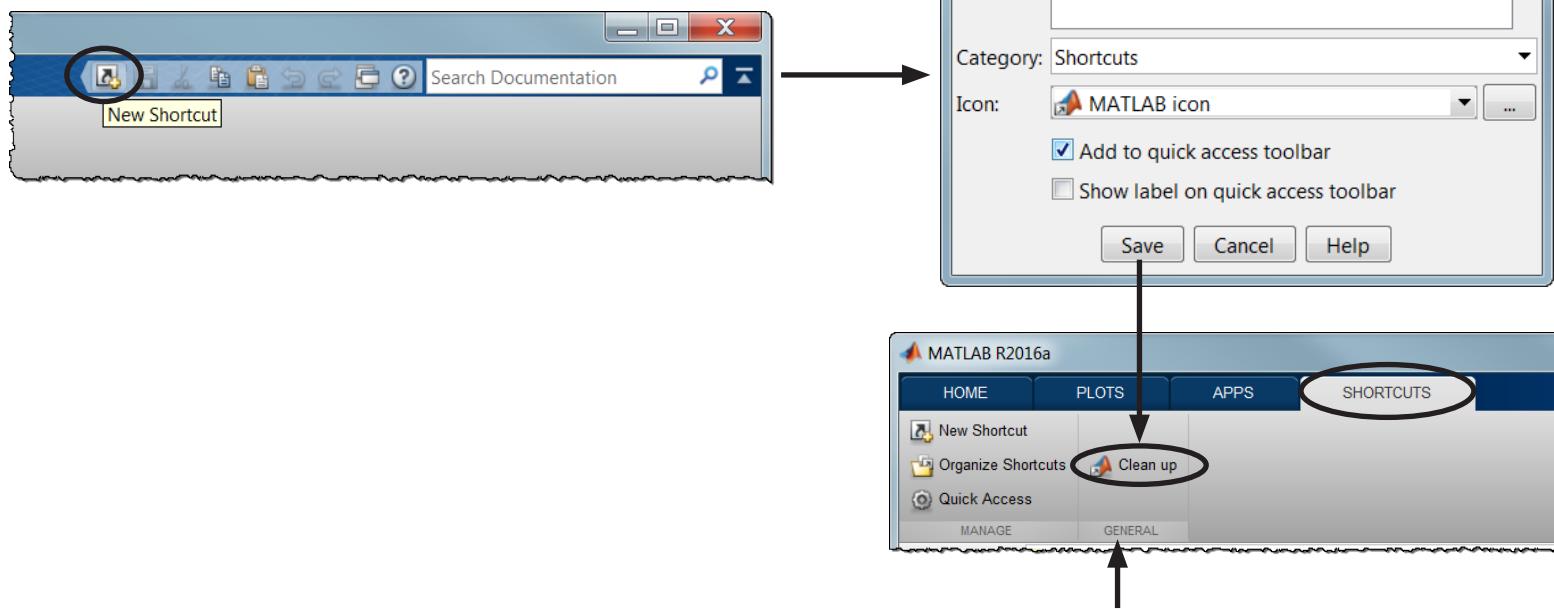
`clear` Clears base workspace variables from memory

`clc` Clears the Command Window (moves prompt to the top)

`close` Closes the current figure

After you begin to use MATLAB regularly, you will probably find other sequences of commands that you execute repeatedly. MATLAB *shortcuts* allow you to execute the commands at the push of a button.

To create a shortcut, click the Create New Command Shortcut button on the quick access toolbar.



Try

Create a shortcut button that executes the callback.

```
clc
close all
clear
```

Add this shortcut to the Quick Access toolbar.

A shortcut consists of a name that explains its purpose and a sequence of commands (called a *callback*) that are executed when the shortcut button is clicked.

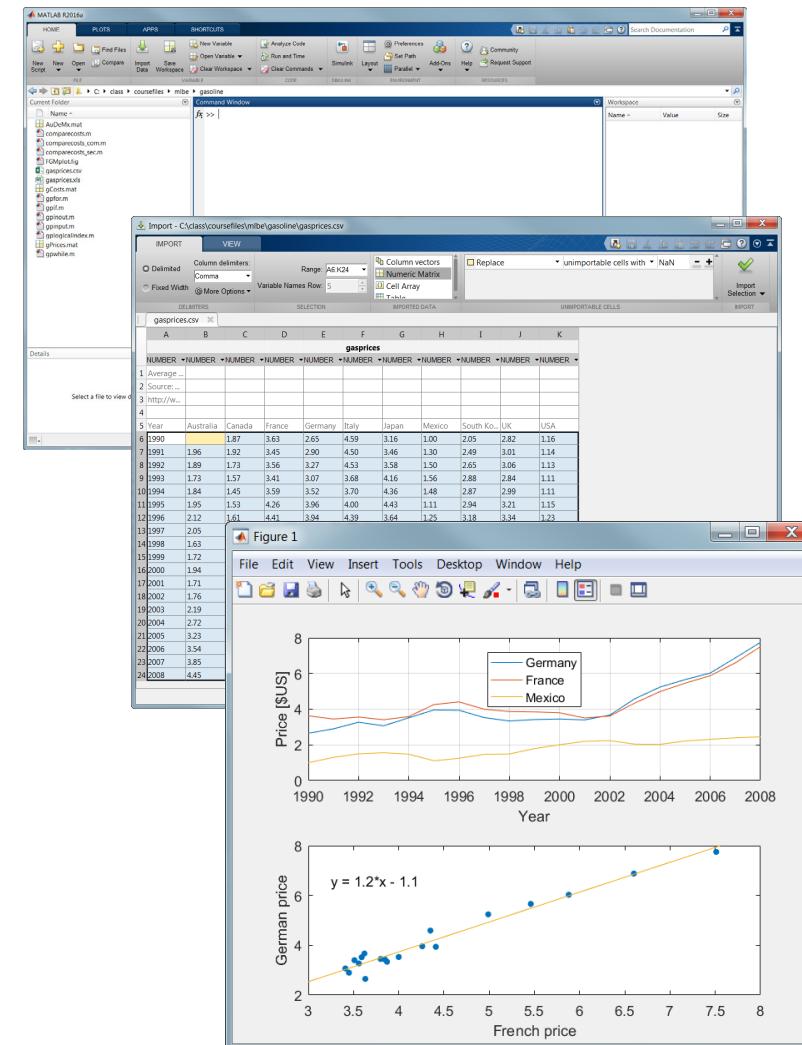
You can choose an icon for your shortcut.

After you create your first shortcut, a **Shortcuts** tab becomes visible on the toolbar and is always available.

You can organize shortcuts into categories that have their own sections in the **Shortcuts** tab.

Summary

- Navigating the interface
- Reading data from file
- Saving and loading variables
- Creating custom plots
- Exporting graphics for use in other applications



Test Your Knowledge

Name: _____

1. Where does MATLAB display a listing of the variables currently in memory and their associated attributes?
 - A. Command Window
 - B. Workspace browser
 - C. Current Folder browser
 - D. Command History
2. The default MATLAB variable type for numeric data is:
 - A. Single
 - B. Double
 - C. Cell
3. T/F: The MATLAB desktop is customizable.