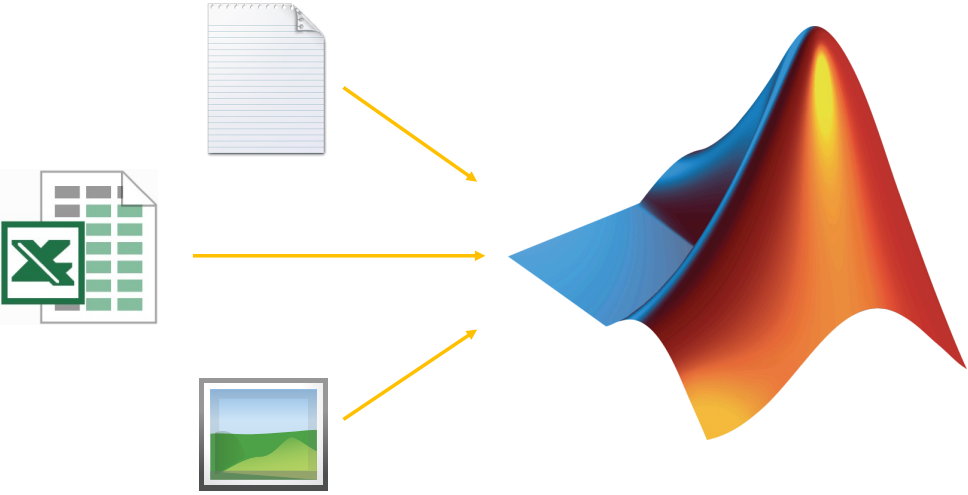


Summary: MATLAB Desktop Tools and Troubleshooting Scripts

Get Data into MATLAB



The diagram illustrates the process of importing data into MATLAB. On the left, three icons represent different data sources: a Microsoft Excel spreadsheet, a plain text document, and a landscape image. Three yellow arrows point from these icons towards a 3D surface plot on the right. The surface plot is colored with a gradient from blue (low values) to red (high values), showing a complex, wavy surface. The entire diagram is enclosed in a rectangular frame with three small circles in the top right corner, resembling a window.

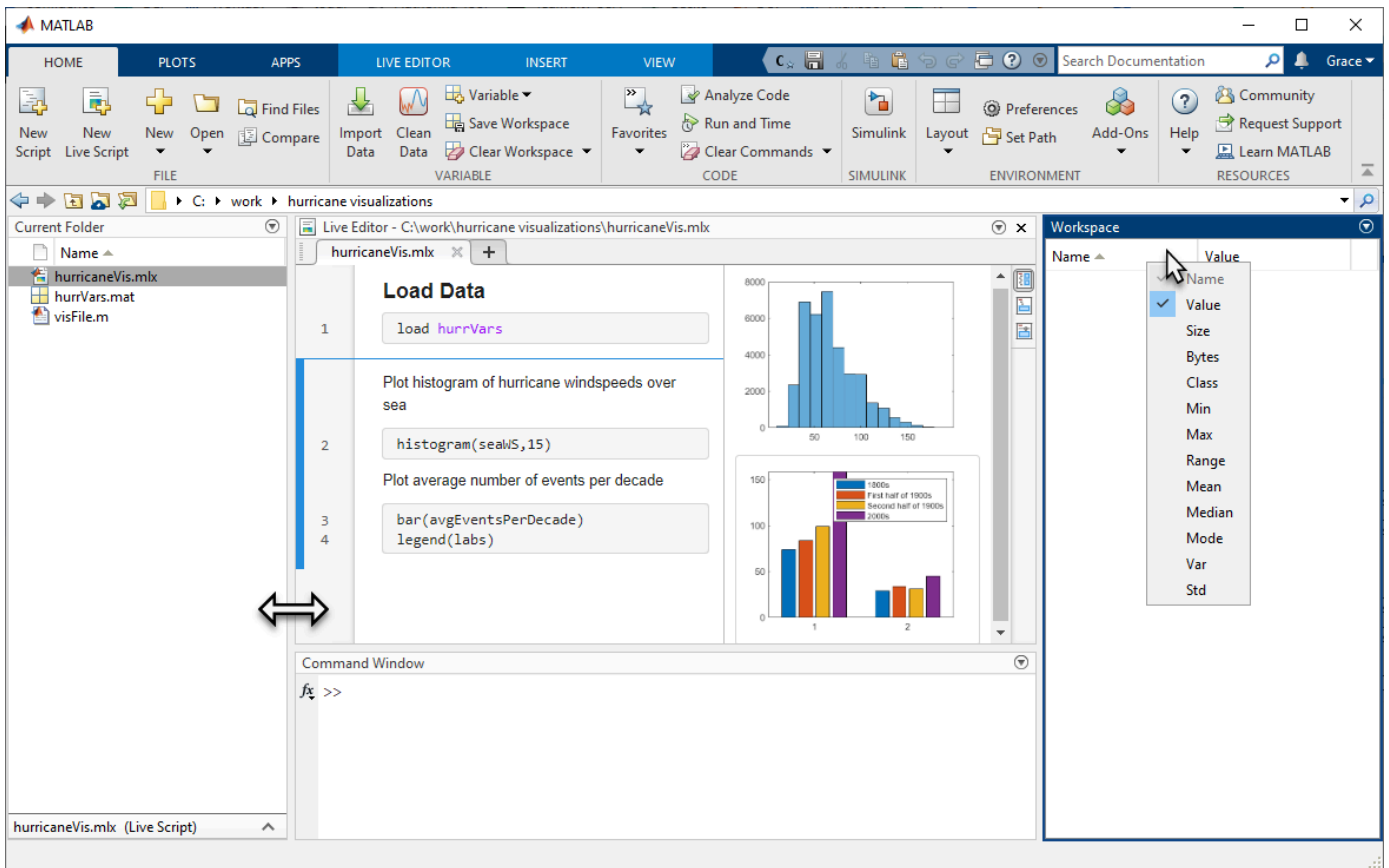
In MATLAB, you can interactively import data files having several formats such as: TXT, CSV, XLS, XLSX, JPG, PNG, etc.

See the documentation for a complete list of [supported file formats for import](#).

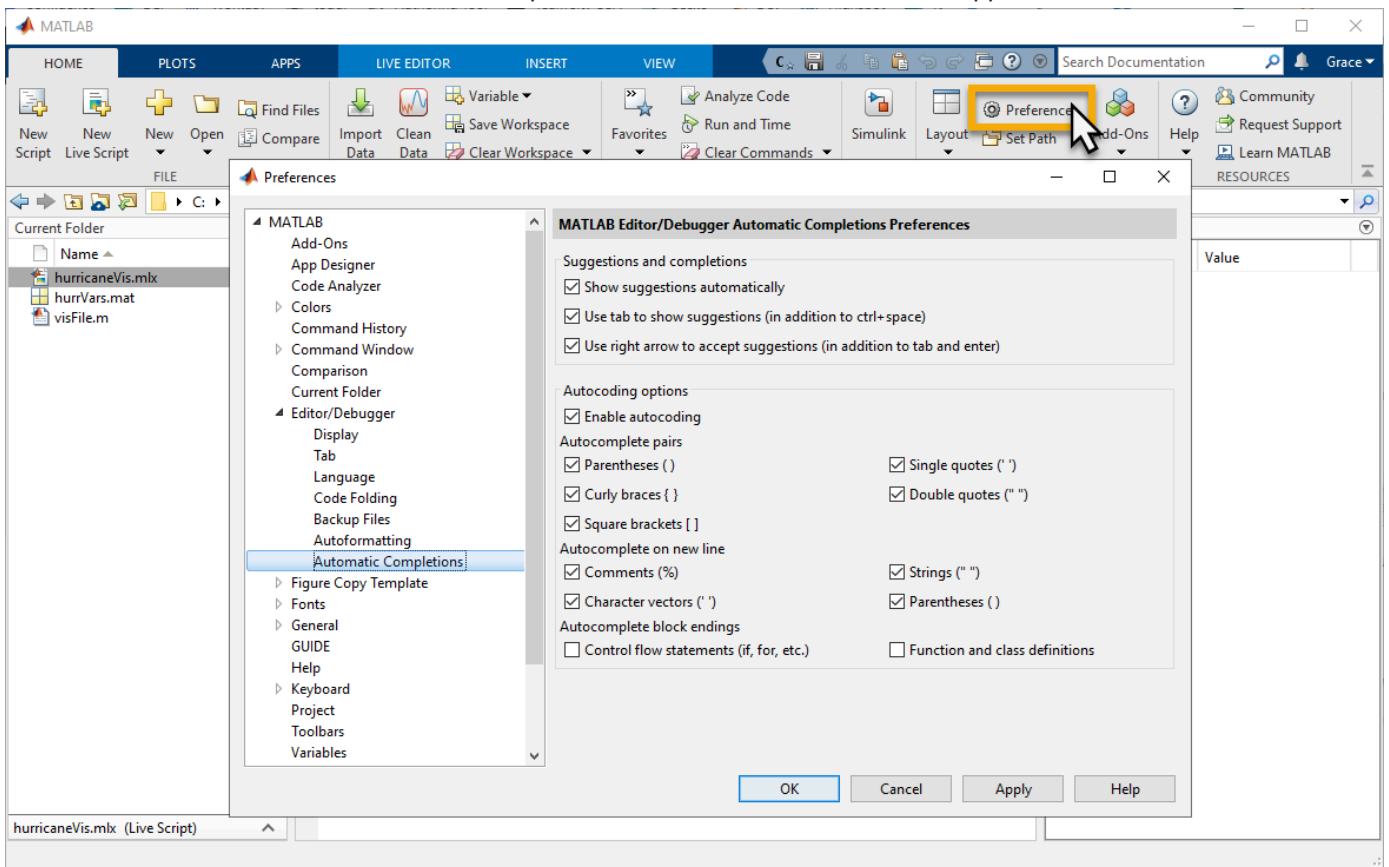
The Desktop Environment

By default, there are three to four windows in the desktop environment: the **Current Folder** browser, the **Workspace Window**, **Command Window**, and if you have a script open, the **Editor Window**. Some common customizations of the desktop environment are

- Resizing or moving windows in the desktop.
- Changing the information displayed for variables in the workspace.



Click on **Preferences** in the Home tab to set preferences about the behavior and appearance of MATLAB.



Live Script Functionality

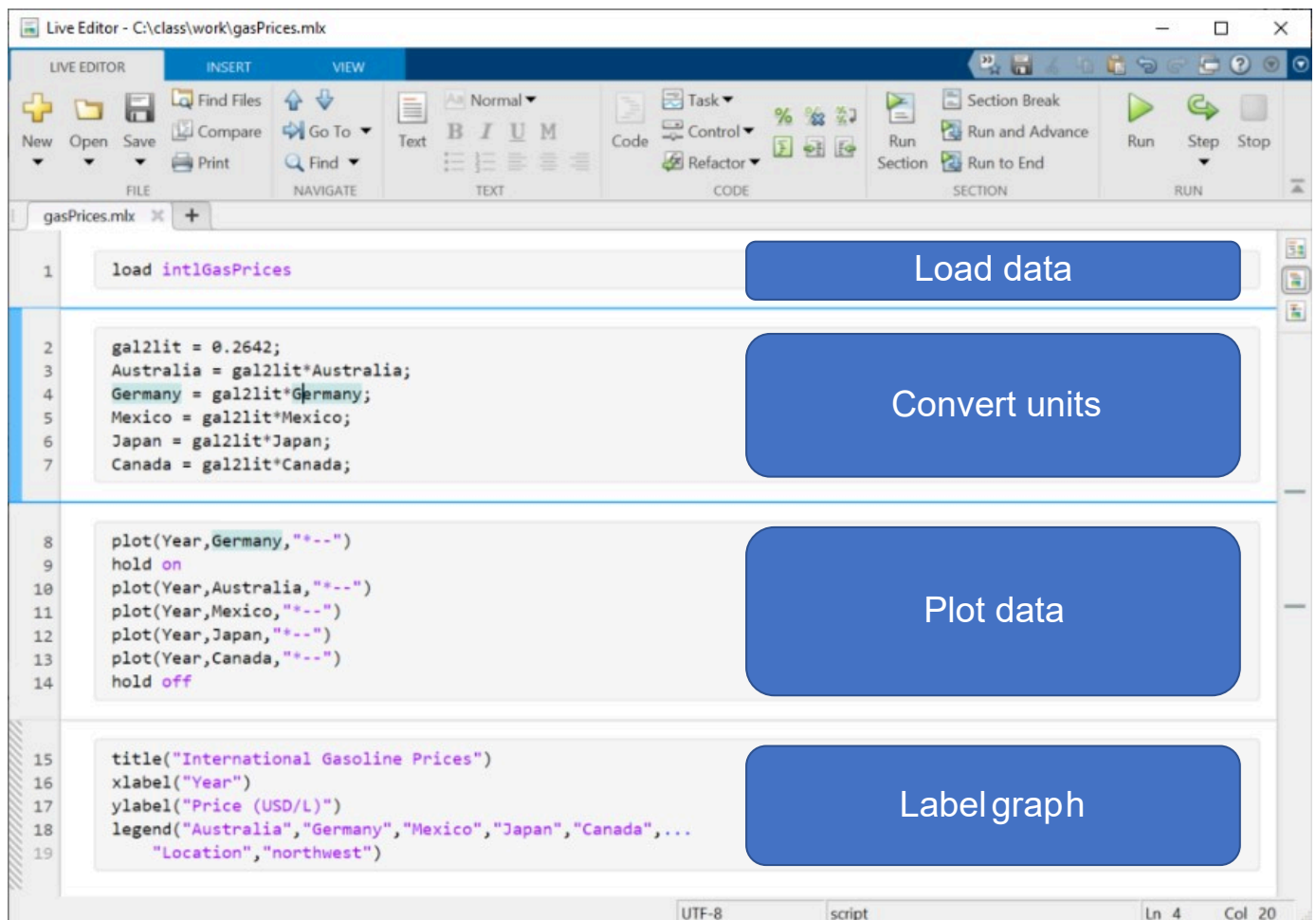
Create and Run a Script

Use the controls in the MATLAB toolstrip to create and run scripts.

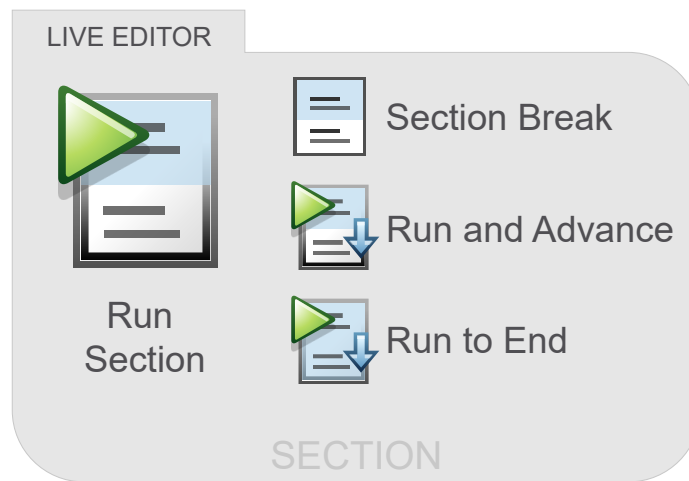


Code Sections

Code sections allow you to organize your code and run sections of code independently. On the **Live Editor** tab, in the **Section** section, click **Section Break** to create a new code section, or press **Ctrl+Alt+Enter**.



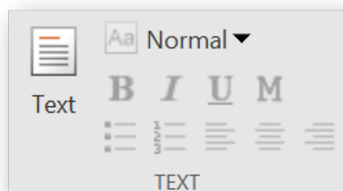
You can run and add code sections in the **Section** section of the **Live Editor** tab in the toolstrip.



Comments and Text

To insert a line of text, click the  **Text** button in the **Text** section of the **Live Editor** tab in the MATLAB Toolstrip.

Format the text using the formatting options provided in the **Text** section.



Comments

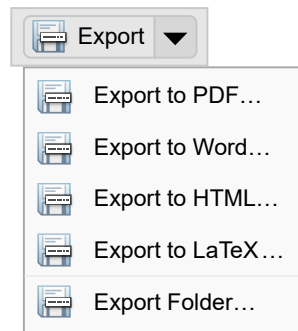
To create a comment, add *% comment* where you want to add more information.

```
load gCosts
% Converts from US$/gal to US$/L
gal2lit = 0.2642;    % conversion factor
Germany = gal2lit*Germany;
Australia = gal2lit*Australia;
Mexico = gal2lit*Mexico;
```

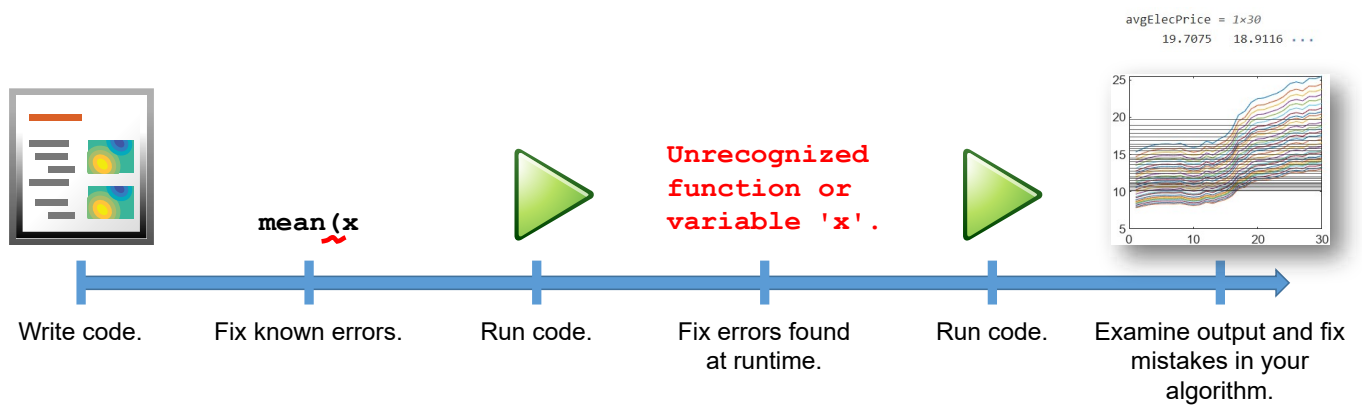
Exporting Live Script Files

You can export your live script and results using the **Export** button in the **Live Editor** tab.

Available formats include PDF, Word, HTML, and LaTeX.



Troubleshooting Code



Code Analyzer

Use the MATLAB **Code Analyzer** messages shown in the Editor to identify and fix syntax errors.

The image shows a MATLAB script editor window titled 'polyPrediction.mlx'. The script is divided into three sections:

- Define Variables:**

```
1 filename = "elec_res.csv";
2 predictdate = datetime(2019,1,1);
3 degree = 5;
```
- Import data and fill missing values:**

```
4 data = readtable(filename,"TextType","string");
5 dates = data{:,1};
6 usage = data{:,2};
7 sector = data.Properties.VariableNames{2};
8 usage = fillmissing(usage,"spline");
```
- Fit polynomial and predict usage for a future date.**

```
9 elapsedYears = years(dates-dates(1);
10 c = polyfit(elapsedYears,usage,degree);
11 endDuration = years(predictdate-dates(1));
12 usageFit = polyval(c,elapsedYears);
13 predictedusage = polyval(c,endDuration);
```

On the right side, the **Code Analyzer** toolbar is visible, featuring a red circle with an exclamation point icon, which indicates that there are errors in the script.

◀ The small red circle with an exclamation point at the top of the **Code Analyzer**. indicates there are errors in the script. ▶

Inspecting Variables

Runtime and algorithm errors are bugs found at runtime.

The image shows a MATLAB script editor window with the following script:

```
1 load elecPrices
2 avgYrPrices = priceYr.*dollar2019;
3 avgElecPrice = mean(avgYrPrices);
4
5 plot(avgYrPrices)
6 yline(avgElecPrice)
```

To the right of the script, a plot is displayed. The x-axis ranges from 0 to 30, and the y-axis ranges from 5 to 25. The plot shows multiple colored lines representing different data series, with a horizontal gray line indicating the average value.

◀ Runtime errors can produce an execution-stopping error or just be something you didn't mean to do, like an algorithm error. An effective way to troubleshoot them is to inspect variables. ▶

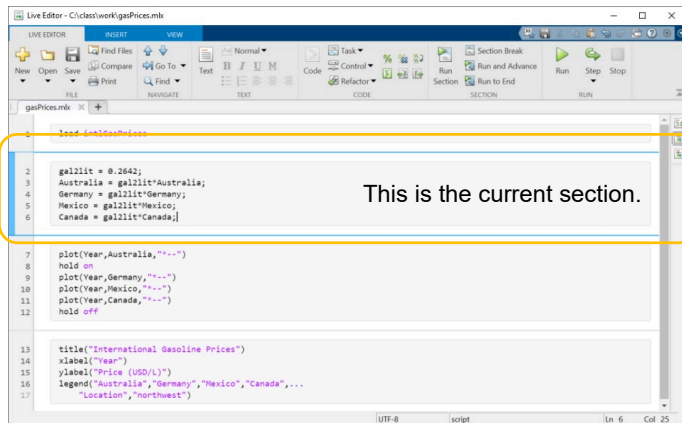
Stepping Through Code

When variables change throughout a script, you can step through your code to inspect intermediate values. You can

run section by section or set breakpoints.

Run and Advance

You can run scripts section by section. In the **Section** section of the **Live Editor** tab in the Toolstrip, you can break up your code into sections to run one at a time.



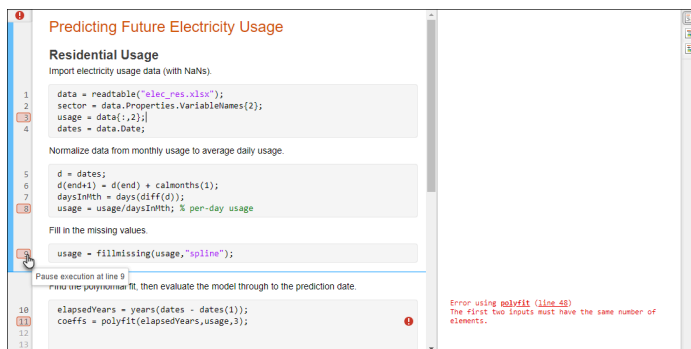
Section Break: Add a section break to create a code section.



Run and Advance: Run code in the current section, then move to the next section.

Setting Breakpoints

You can also set breakpoints to stop code execution before specific lines. Add breakpoints by clicking line numbers.



Continue: Run code until the next breakpoint (or the end of the script).



Step: Run only the next line of code.



Stop: Stop code execution and exit debug mode.

Don't forget to clear your breakpoints and save your work!