

21+ MATLAB Features You Need Now!

This is a quick summary of the features covered in the talk 21 MATLAB Features You Need Now.

Table of Contents

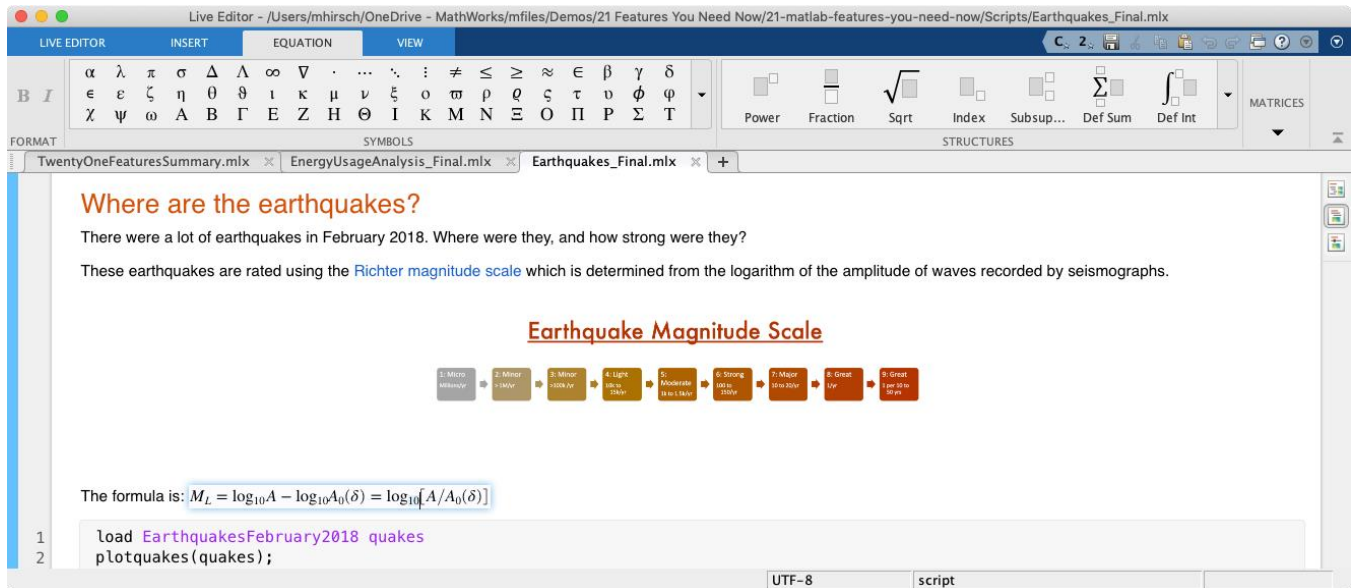
Live Editor	2
Formatted text, images, and equations	2
Output on right or inline	3
Interactive outputs	5
Function hints	5
Interactive controls	6
Live Editor Tasks	6
Document export	7
Graphics	7
Default plot interactivity.....	7
Geographic plots	8
tiledlayout	8
Plots tab.....	9
Building and Sharing Apps	10
App Designer.....	10
Web Apps	11
Data Preprocessing	12
table and timetable	12
stackedplot	12
Missing data functions.....	13
Preprocessing Live Editor Tasks.....	13
Apps	14
Hardware support	15
Add-On Explorer	16
Programming	17
Code Compatibility Report.....	17
File Comparison Tool	18
Code Analyzer	18
Multi-release Release Notes	19

Live Editor

You can create scripts that combine code, output, and formatted text with the [Live Editor](#). In fact, this document was created with the Live Editor. Some of our favorite features of the Live Editor are:

Formatted text, images, and equations

You can use formatted text and insert images and equations. There's a really nice equation editor, or you can use LaTeX if you prefer:



The screenshot shows the MATLAB Live Editor interface. The title bar indicates the file path: `Live Editor - /Users/mhirsch/OneDrive - MathWorks/mfiles/Demos/21 Features You Need Now/21-matlab-features-you-need-now/Scripts/Earthquakes_Final.mlx`. The interface includes a ribbon with tabs for **LIVE EDITOR**, **INSERT**, **EQUATION**, and **VIEW**. The **FORMAT** section contains a rich text editor with various symbols and formatting options. The **STRUCTURES** section includes mathematical symbols like Power, Fraction, Sqrt, Index, Subsup..., Def Sum, and Def Int. The **MATRICES** section is also visible.


The main content area displays a document with the following text:

Where are the earthquakes?

There were a lot of earthquakes in February 2018. Where were they, and how strong were they?

These earthquakes are rated using the **Richter magnitude scale** which is determined from the logarithm of the amplitude of waves recorded by seismographs.

Earthquake Magnitude Scale



The formula is: $M_L = \log_{10} A - \log_{10} A_0(\delta) = \log_{10} [A/A_0(\delta)]$

At the bottom, a code block is visible with the following MATLAB code:

```
1 load EarthquakesFebruary2018 quakes
2 plotquakes(quakes);
```

The status bar at the bottom right shows the encoding as **UTF-8** and the file type as **script**.

Output on right or inline


Output appears immediately adjacent to your code, either below or side-by-side:

Live Editor - /Users/mhirsch/OneDrive - MathWorks/mfiles/Demos/21 Features You Need Now/21-matlab-features-you-ne...

LIVE EDITOR INSERT FIGURE VIEW

TwentyOneFeaturesSummary.mlx Earthquakes.mlx * +

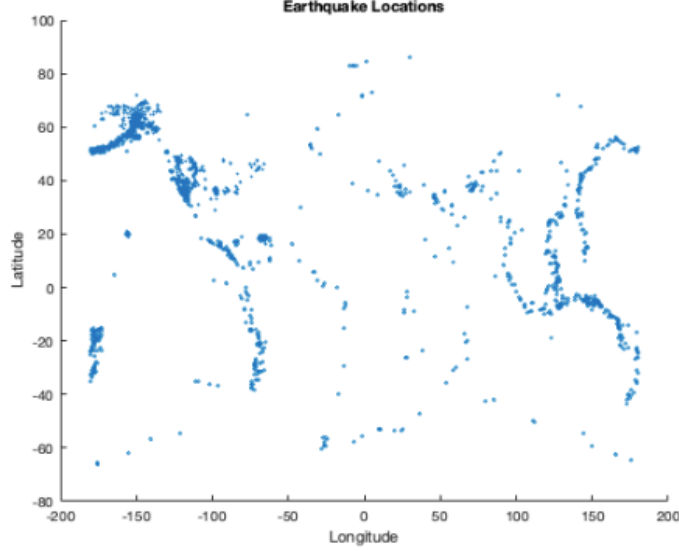
Earthquake Magnitude Scale



Category	Frequency
1: Micro	Millions/yr
2: Minor	> 1 M/yr
3: Minor	> 100k/yr
4: Light	10k to 100k/yr
5: Moderate	1k to 10k/yr
6: Strong	100 to 1000/yr
7: Major	10 to 100/yr
8: Great	1/yr
9: Great	1 per 50 to 500 yr

The formula is: $M_L = \log_{10}A - \log_{10}A_0(\delta) = \log_{10}[A/A_0(\delta)]$. [Earthquake Data Credit](#): U.S. Geological Survey; Department of the Interior/USGS

```
1 load EarthquakesFebruary2018 quakes
2 plotquakes(quakes);
```



Earthquakes around the world

Let's take a look at where earthquakes occur and how strong they are. Filter out the smallest earthquakes.

```
3 minmagnitude = 5;
4 theseQuakes = quakes(quakes.Magnitude > minmagnitude,:);
5 scatter(theseQuakes, "Latitude", "Longitude", "SizeVariable", "Magnitude");
```

UTF-8 script Ln 6 Col 31

Live Editor - /Users/mhirsch/OneDrive - MathWorks/mfiles/Demos/21 Features You Need Now/21-matlab-features-you-need-now/Earthquakes.mlx

LIVE EDITOR INSERT FIGURE VIEW C 2

TwentyOneFeaturesSummary.mlx Earthquakes.mlx +

Where are the earthquakes?

There were a lot of earthquakes in February 2018. Where were they, and how strong were they?

These earthquakes are rated using the **Richter magnitude scale** which is determined from the logarithm of the amplitude of waves recorded by seismographs.

Earthquake Magnitude Scale

1: Micro 1mm/yr	2: Minor 1/100 yr	3: Minor 1/100 yr	4: Light 10 to 100 yr	5: Moderate 10 to 200 yr	6: Strong 100 to 200 yr	7: Major 10 to 200 yr	8: Great 1 yr	9: Great 1 per 10 to 100 yr
--------------------	----------------------	----------------------	--------------------------	-----------------------------	----------------------------	--------------------------	------------------	--------------------------------

The formula is: $M_L = \log_{10} A - \log_{10} A_0(\delta) = \log_{10} [A/A_0(\delta)]$. Earthquake Data Credit: U.S. Geological Survey; Department of the Interior/USGS

```

1    load EarthquakesFebruary2018 quakes
2    plotquakes(quakes);

```

Earthquakes around the world

Let's take a look at where earthquakes occur and how strong they are. Filter out the smallest earthquakes.

```

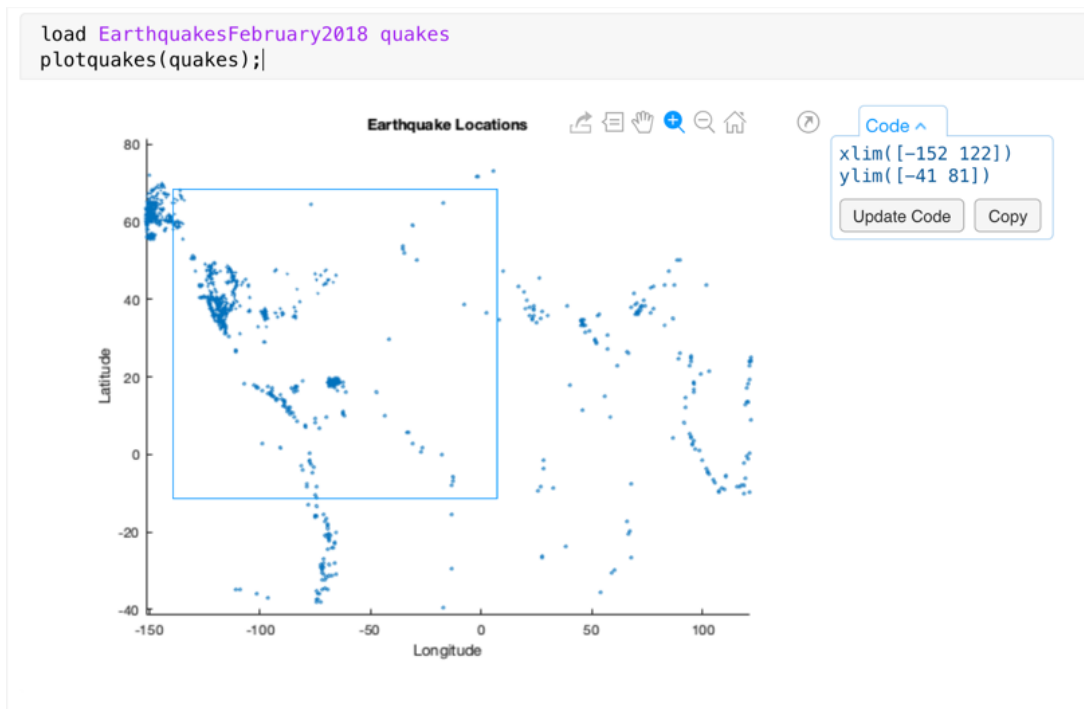
3    minmagnitude = 5;
4    theseQuakes = quakes(quakes.Magnitude > minmagnitude,:);
5    geobubble(theseQuakes,"Latitude","Longitude","SizeVariable","Magnitude"
6           "Basemap","colorterrain");

```

UTF-8 script Ln 6 Col 31

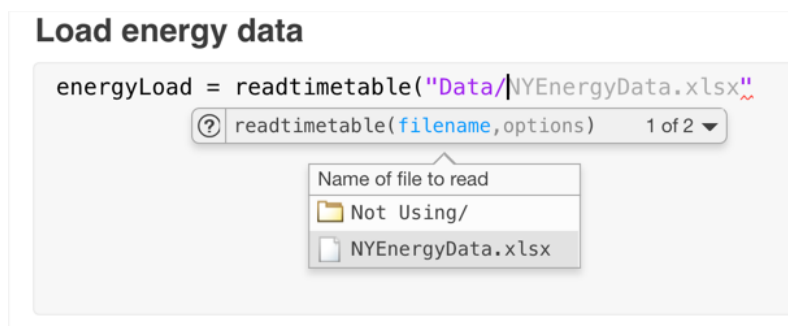
Interactive outputs

You can interact with outputs and generate MATLAB code for your interactions. For instance, you can zoom into a plot and then automatically update your program to set the new axis limits.



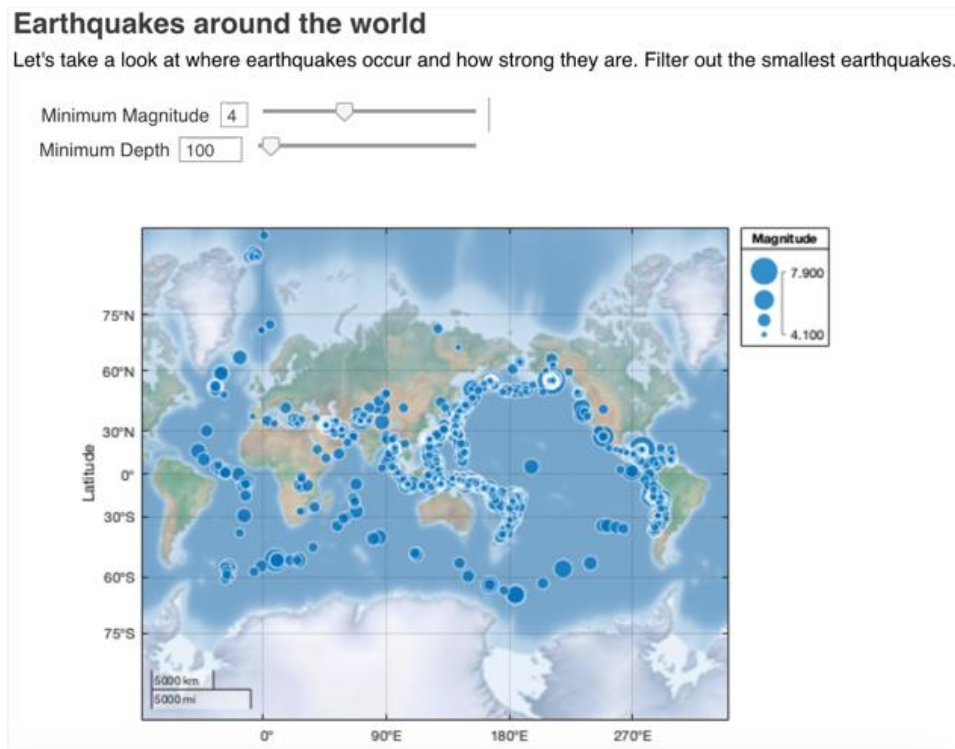
Function hints

Function hints make it easy to call functions without having to look at the documentation, and even to navigate to select files to import:



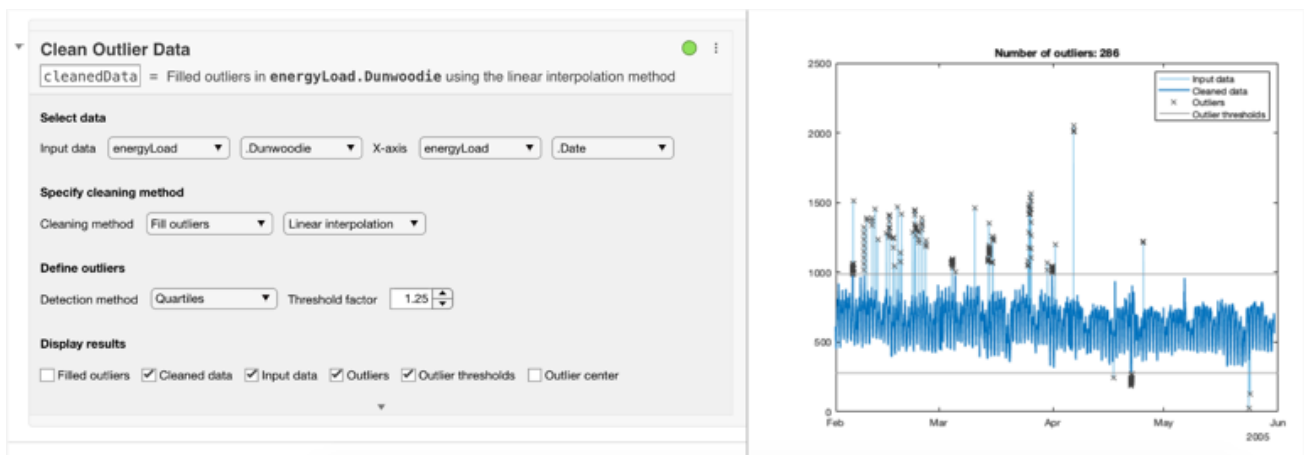
Interactive controls

You can replace the value of variables with interactive controls to make it easy to interact with your program, and even hide the code altogether:



Live Editor Tasks

You can use Live Editor tasks to preprocess data interactively and the MATLAB code is automatically written for you. There are also Live Editor tasks for working with symbolic expressions, designing control systems, and more.



Document export

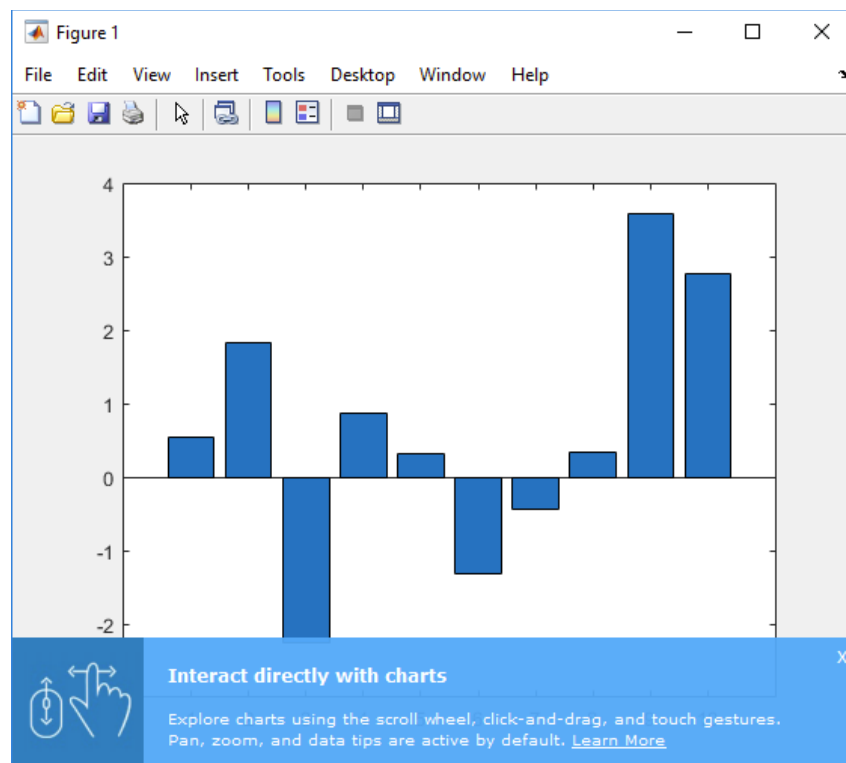
You can export your Live Scripts to PDF, Word, HTML, or LaTeX documents.

Graphics

There are a lot of new and some undiscovered graphics features. Here are just a few of our favorites:

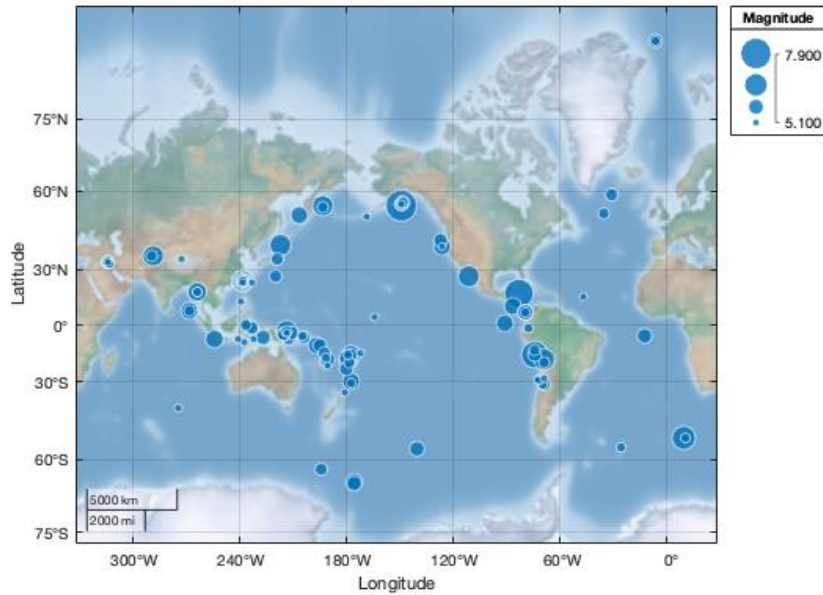
Default plot interactivity

You can use your mouse or trackpad to zoom, pan, rotate, and add datatips without having to click separate toolbar buttons first:



Geographic plots

You can now create [geographic plots](#) in MATLAB, such as this geographic bubble chart showing the location and magnitude of earthquakes around the world:



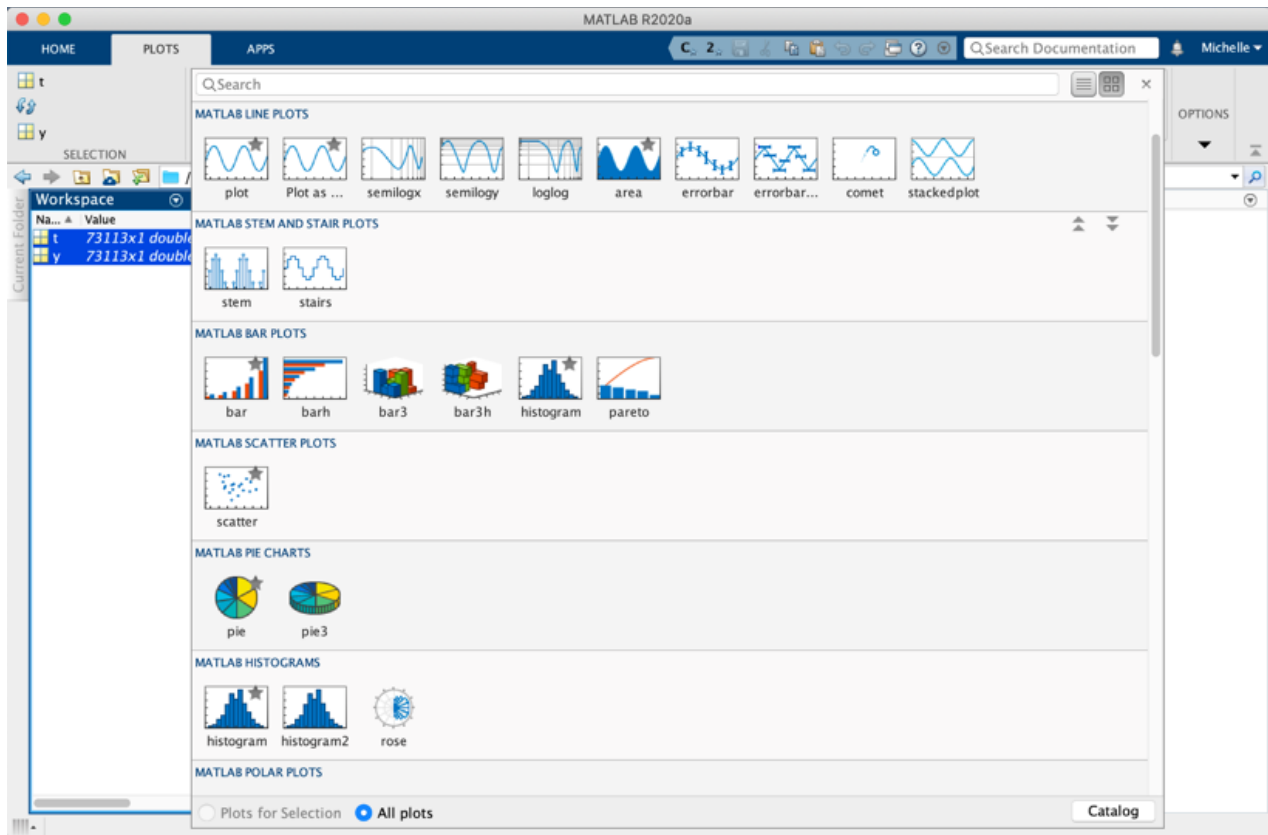
tiledlayout

You can much more easily control the layout of figures with multiple plots using [tiledlayout](#) instead of subplot.



Plots tab

You can easily plot your data with the Plots tab of the MATLAB Toolstrip. Just select your data in the Workspace browser or Variable Editor and you'll get a list of plots that work with your specific data. The equivalent MATLAB code is automatically entered in the command window for you to access later.

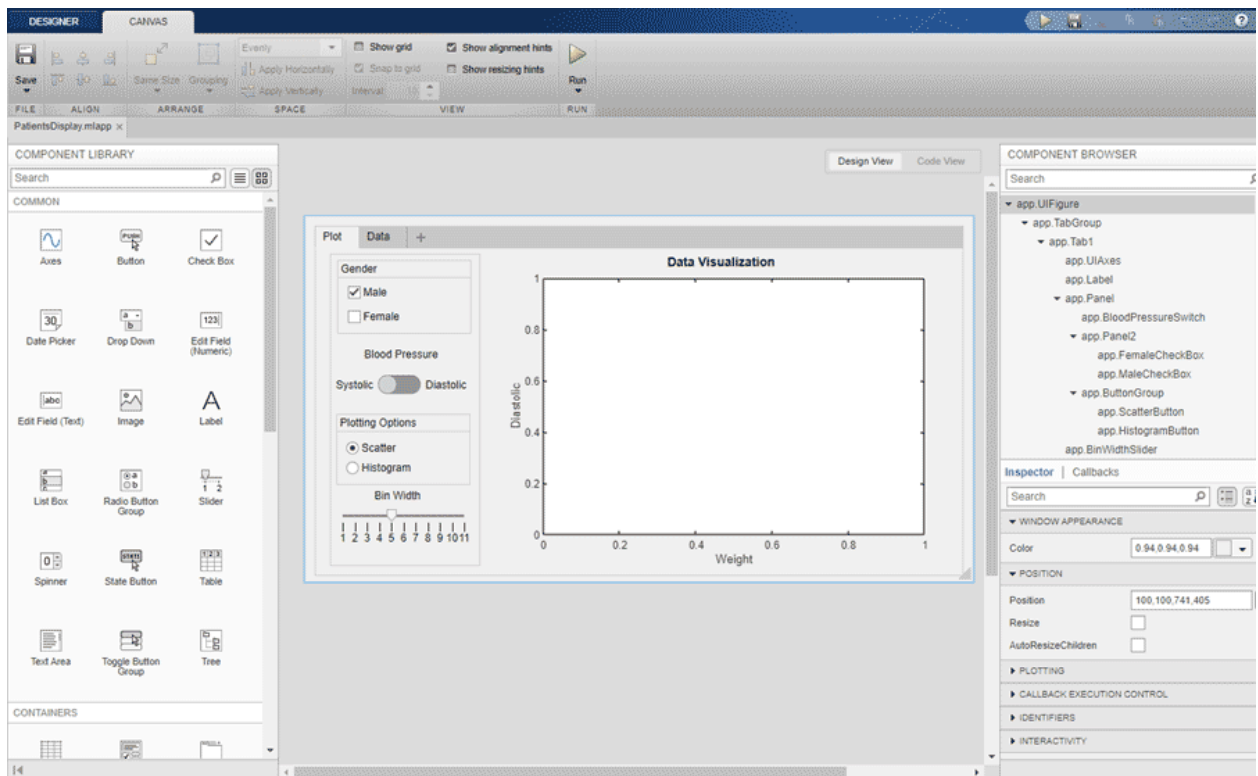


Building and Sharing Apps

You can build and share custom apps with MATLAB.

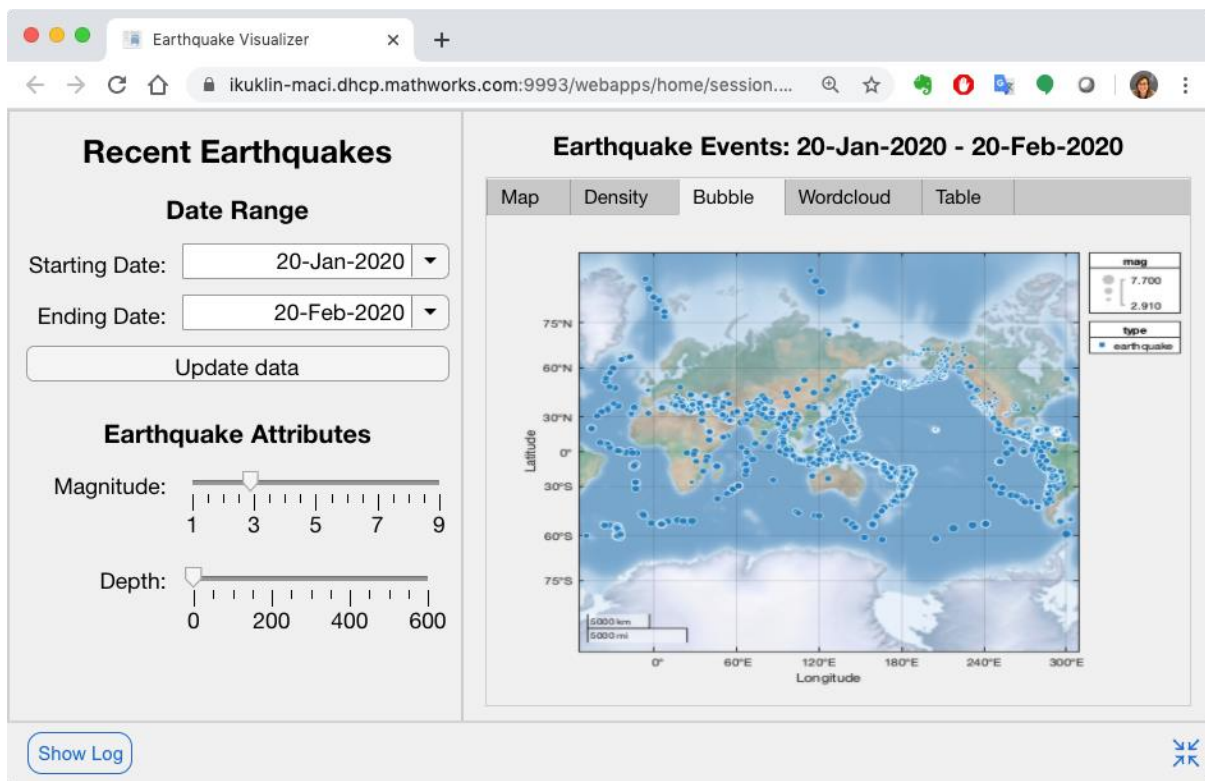
App Designer

[App Designer](#) lets you create professional apps without having to be a professional software developer. Drag and drop visual components to lay out the design of your graphical user interface (GUI) and use the integrated editor to quickly program its behavior.



Web Apps

Using the MATLAB Compiler and the [MATLAB Web App Server](#), you can share your apps as web apps that anybody in your organization can access from their web browser.

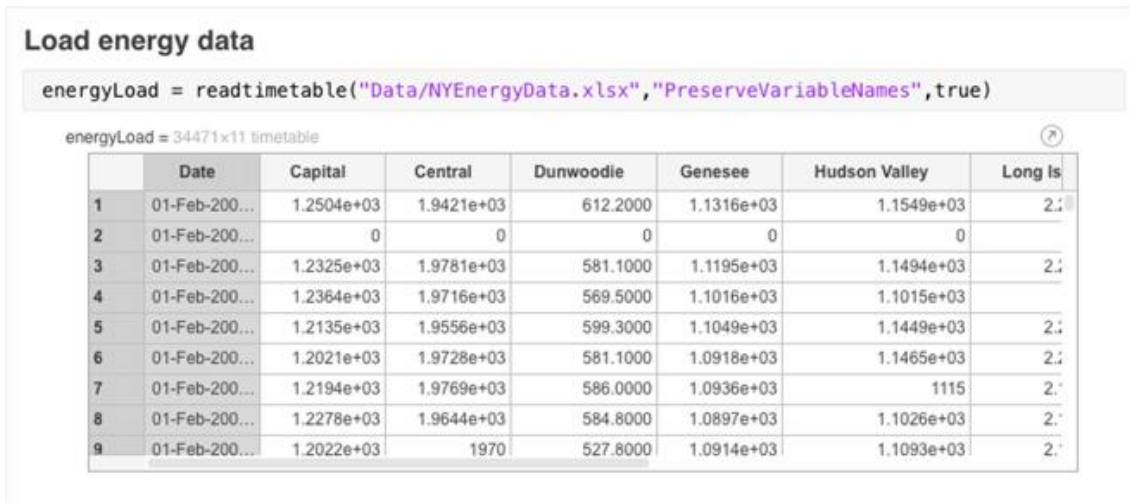


Data Preprocessing

MATLAB takes the pain out of preprocessing your data to get it ready for analysis.

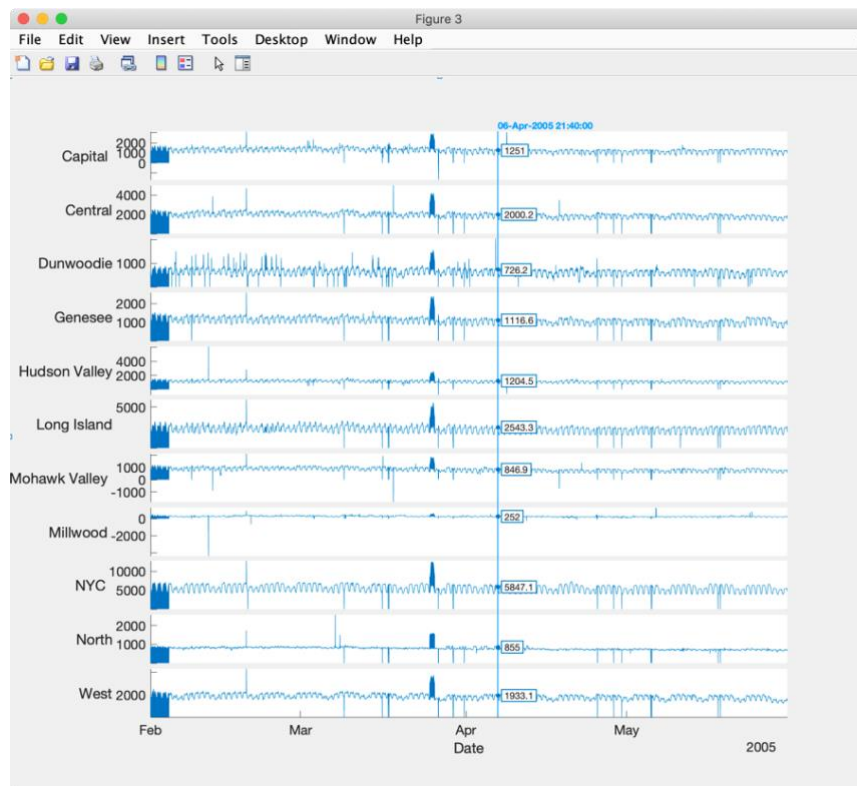
table and timetable

You can organize tabular data, such as you might store in spreadsheets, using the `table` and `timetable` data types. Timetables provide additional capabilities for managing data that varies vs. time, such as synchronizing and resampling.



stackedplot

Visualize every variable in your table or timetable with a `stackedplot`:



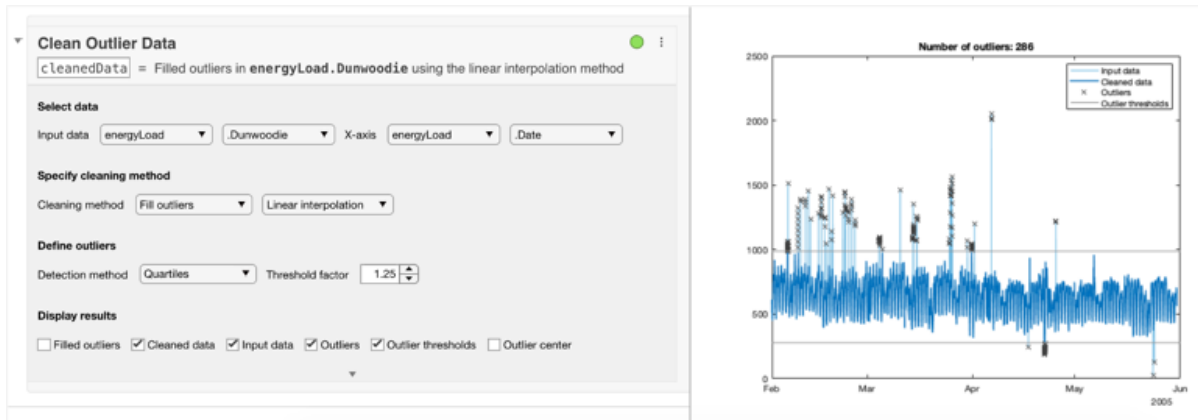
Missing data functions

Use a straightforward collection of functions for cleaning up missing data and outliers:

Functions expand all	
Missing Data and Outliers	
<code>ismissing</code>	Find missing values
<code>rmmissing</code>	Remove missing entries
<code>fillmissing</code>	Fill missing values
<code>missing</code>	Create missing values
<code>standardizeMissing</code>	Insert standard missing values
<code>isoutlier</code>	Find outliers in data
<code>filloutliers</code>	Detect and replace outliers in data
<code>rmoutliers</code>	Detect and remove outliers in data
<code>movmad</code>	Moving median absolute deviation

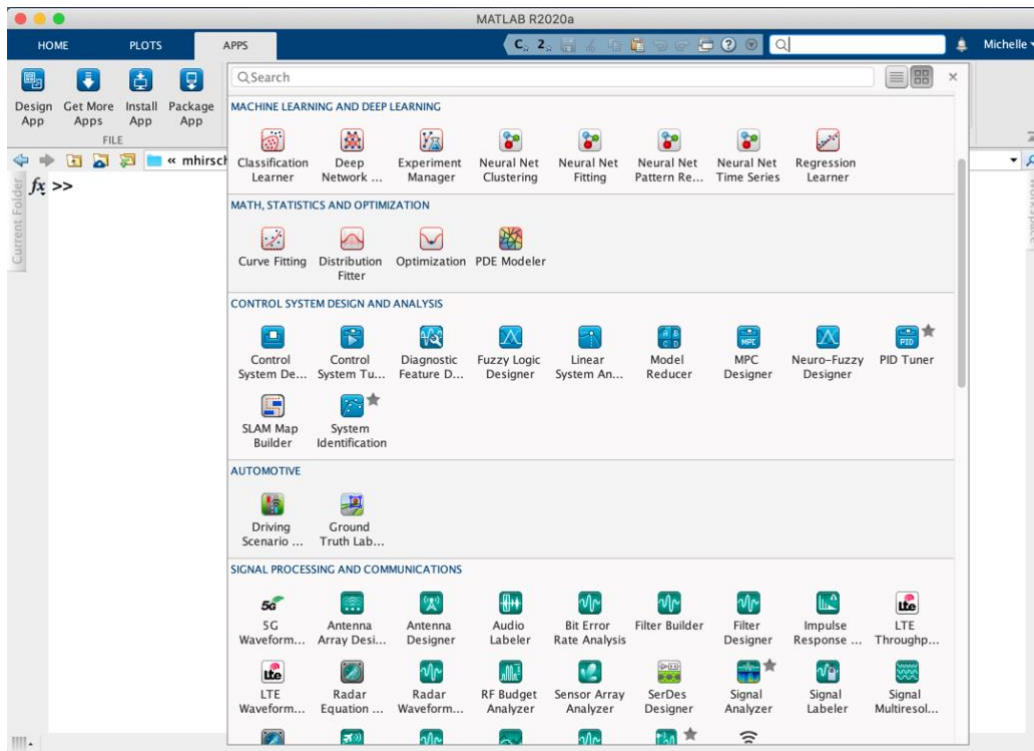
Preprocessing Live Editor Tasks

As mentioned above, you can use Live Editor tasks to preprocess data interactively and the MATLAB code is automatically written for you.



Apps

Even though a programming language is the heart of MATLAB, you don't have to write code to complete common tasks in MATLAB. You can design control systems or deep neural networks, analyze signals or radar waveforms, process images and videos, and so much more, without writing a single line of code. Even better - they write the code for you, so you don't have to choose between the ease and speed of using an app and the automation and flexibility of writing a program. Here are some of the almost 100 different [MATLAB Apps](#) available on the Apps tab of the MATLAB Toolstrip:



Hardware support

MATLAB and Simulink connect to the hardware you use, whether you want to capture images from a webcam or a high-end IP camera, automatically generate real-time C code for rapid prototyping or deploying on an embedded processor, or give students hands-on experiences for project-based learning. Find and install hardware support directly within MATLAB:

The screenshot displays the MATLAB Add-On Explorer window. The top navigation bar includes 'Contribute' and 'Manage Add-Ons' links, along with a search bar containing 'Clear Filters' and 'Search for add-ons'. On the left, there are three filter sections: 'Filter by Source' (MathWorks: 185, Community: 123), 'Filter by Category' (Using MATLAB: 6, Applications: 192, Using Simulink: 47), and 'Filter by Type' (Hardware Support Packages: 308, selected). The main content area is titled 'Hardware Support Packages (308)' and shows a grid of nine packages. Each package card includes an image, a title, a brief description, download count, and a star rating. The 'MATLAB Support Package for USB Webcams' package is marked as 'Installed'.

Package Name	Description	Downloads	Rating
MATLAB Support Package for Arduino Hardware	Acquire inputs and send outputs on Arduino boards	1740	★★★★★
Legacy MATLAB and Simulink Support for Arduino	MATLAB class and Simulink blocks for communicating with an Arduino microcontroller board	1132	★★★★★
Simulink Support Package for Arduino Hardware	Run models on Arduino boards.	1025	★★★★★
MATLAB Support Package for USB Webcams	Acquire images and video from UVC compliant webcams.	460	★★★★★
Communications Toolbox Support Package for RTL-SDR Radio	Acquire RF data using RTL-SDR.	444	★★★★★
Image Acquisition Toolbox Support Package for OS Generic Video Interface	Acquire video and images from generic video capture devices.	423	★★★★★

Add-On Explorer

Download, install, and use add-ons without leaving the MATLAB environment, using the [Add-On Explorer](#). Whether you need additional toolboxes, apps, hardware support packages, or community submissions, you can easily browse and find what you're looking for.

The screenshot displays the MATLAB Add-On Explorer window. At the top, there is a search bar with the text "Search for add-ons" and a search icon. Below the search bar, there are two tabs: "Contribute" and "Manage Add-Ons".

Filter by Source
Community 37,835

Filter by Category

Using MATLAB

- Get Started with MATLAB 27
- Language Fundamentals 923
- Data Import and Analysis 1,048
- Mathematics 1,410
- Graphics 1,875
- Programming 402
- App Building 416
- Software Development Tools 150
- External Language Interfaces 445
- Environment and Settings 134
- Installation, Licensing, and Activation 13
- Parallel Computing 131
- Application Deployment 63
- Database Access and Reporting 121

Applications

- Science and Industry 3,531
- Image Processing and Computer Vision 2,412
- AI, Data Science, and Statistics 1,483
- Signal Processing 1,564
- Wireless Communications 696
- RF and Mixed Signal 117
- Math and Optimization 1,009
- Control Systems 912
- Autonomous Systems 205
- Hardware, IoT, and Test & Measurement 481
- Games 519

Using Simulink

- Simulink 400
- Simulation Graphics and Reporting 20
- Real-Time Simulation and Testing 35

Community Toolboxes [Show All 1,232](#)

- Simulink Onramp**
Learn the basics of how to create, edit, and simulate Simulink models through an interactive tutorial.
1809 Downloads ⭐⭐⭐⭐⭐
- GUI Layout Toolbox**
Layout manager for MATLAB graphical user interfaces
1158 Downloads ⭐⭐⭐⭐⭐
- Numerical Computing with MATLAB**
Toolbox containing files and app from Numerical Computing with MATLAB
830 Downloads ⭐⭐⭐⭐⭐

Community Apps [Show All 316](#)

- PID Controller Simulator**
PID controller simulator on an LTI system w/ or w/o input delays
716 Downloads ⭐⭐⭐⭐⭐
- Transfer Learning**
Transfer Learning of Pre-trained Neural Network or Imported ONNX Classification Model in GUI
410 Downloads ⭐⭐⭐⭐⭐
- Aircraft Intuitive Design (AID)**
An academic tool intended to assist in developing an intuitive understanding of aircraft design.
383 Downloads ⭐⭐⭐⭐⭐

Community Simulink Models [Show All 2,426](#)

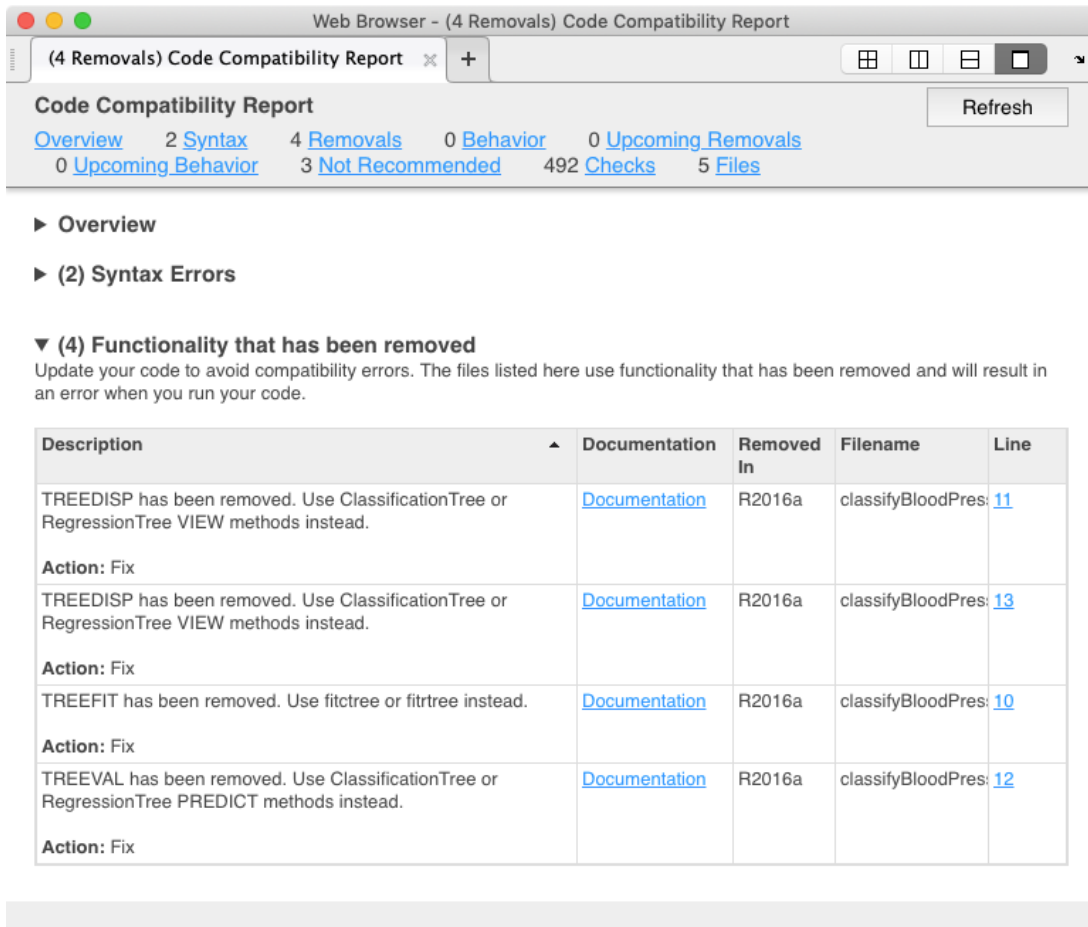
- Simulation plot showing a graph with multiple curves.
- Diagram of a car chassis with a person silhouette.
- Blue abstract geometric shapes.

Programming

We barely scratched the surface on programming, but touched on a few of our favorite tools that make programming easier:

Code Compatibility Report

You can run the [Code Compatibility Report](#) to analyze compatibility of your code. This report helps you identify and address any potential compatibility issues, and estimate the effort required (if any) to upgrade to a newer MATLAB release.



Code Compatibility Report Refresh

[Overview](#) 2 [Syntax](#) 4 [Removals](#) 0 [Behavior](#) 0 [Upcoming Removals](#)
0 [Upcoming Behavior](#) 3 [Not Recommended](#) 492 [Checks](#) 5 [Files](#)

► **Overview**

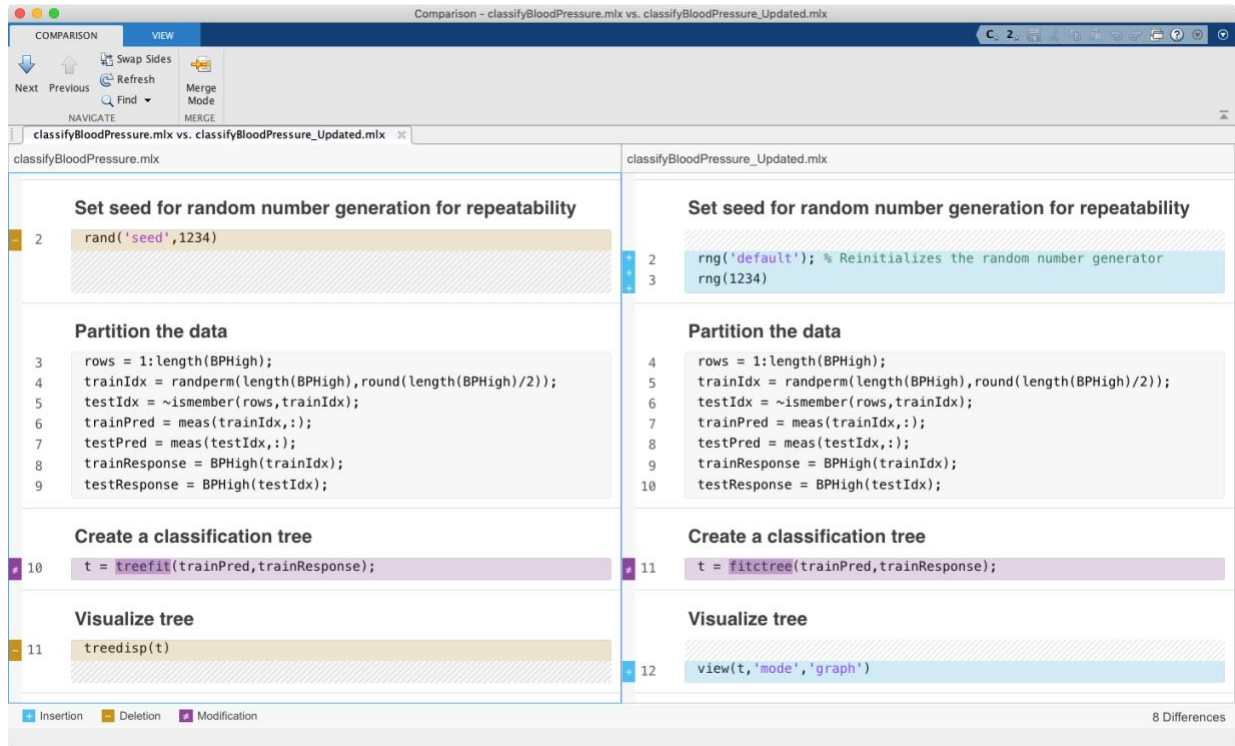
► **(2) Syntax Errors**

▼ **(4) Functionality that has been removed**
Update your code to avoid compatibility errors. The files listed here use functionality that has been removed and will result in an error when you run your code.

Description	Documentation	Removed In	Filename	Line
TREEDISP has been removed. Use ClassificationTree or RegressionTree VIEW methods instead. Action: Fix	Documentation	R2016a	classifyBloodPres:	11
TREEDISP has been removed. Use ClassificationTree or RegressionTree VIEW methods instead. Action: Fix	Documentation	R2016a	classifyBloodPres:	13
TREEFIT has been removed. Use fitctree or fitrtree instead. Action: Fix	Documentation	R2016a	classifyBloodPres:	10
TREEVAL has been removed. Use ClassificationTree or RegressionTree PREDICT methods instead. Action: Fix	Documentation	R2016a	classifyBloodPres:	12

File Comparison Tool

Compare two versions of a program and merge changes using the [File Comparison Tool](#).



Code Analyzer

Did you ever wonder what those orange and red squiggles in your code meant? These are the [Code Analyzer](#) at work, telling you about code that will definitely error when you try to run it (red), or that could potentially be improved (orange).

```
% Find input indices that are not line objects
nothandle = ~ishandle(hline);
for nh = 1:prod(size(hline))
    notline(nh) = ~ishandle(hline(nh)) || ~strcmp('line', lower(get(hline(nh), 'type')));
end

len = zeros(size(hline));
for nl = 1:prod(size(hline))
    % If it's a line, get the data and compute the length
    if ~notline(nl)
        flds = get(hline(nl));
        fdata = {'XData', 'YData', 'ZData'};
        for nd = 1:length(fdata)
            data{nd} = getfield(flds, fdata{nd});
        end
        % If there's no 3rd dimension, or all the data in one dimension is
        % unique, then consider it to be a 2D line.
        if isempty(data{3}) || ...
```

Multi-release Release Notes

We know it's hard to keep up with so many great new capabilities released twice a year, every year. That's why we've given you more control over the [release notes](#), so you can view all of the changes for a given section of MATLAB or add-on product across a range of releases.

The screenshot shows a web browser window with the URL 'MATLAB Release Notes'. The page title is 'Documentation' and it includes a search bar. On the left, there is a 'CONTENTS' sidebar with a 'Category' list where 'Graphics' is selected. Below the categories are various toolboxes like Simulink, 5G Toolbox, Aerospace Blockset, etc. A 'Text Filter' and 'Release Range' section are also present, with the range set to 'R2018b' to 'R2020a'. The main content area shows 'MATLAB Release Notes' for the 'Graphics' category, with 'Found 53 notes' and a 'Sort by' dropdown set to 'Release: Latest to Earliest'. The notes are grouped by release version: R2020a, R2019b, R2019a, and R2018b, each with a sub-header 'New Features, Bug Fixes, Compatibility Considerations'. At the bottom, there is a feedback section asking 'How useful was this information?' with five star icons.