

# MATLAB EXPO

## What's New in MATLAB R2020a

*Dr Peter Brady*  
*Senior Application Engineer*



2,611

# Agenda

1. Core MATLAB
2. Server Suite
3. AI Applications
4. Industrial Tools
5. Financial Models

*“High level to low level –  
Deep or Wide  
– you choose for you”*

# Core MATLAB

1. Live scripts and task
2. Stateflow for MATLAB
3. Projects

*“Abstraction allows for rapid development”*

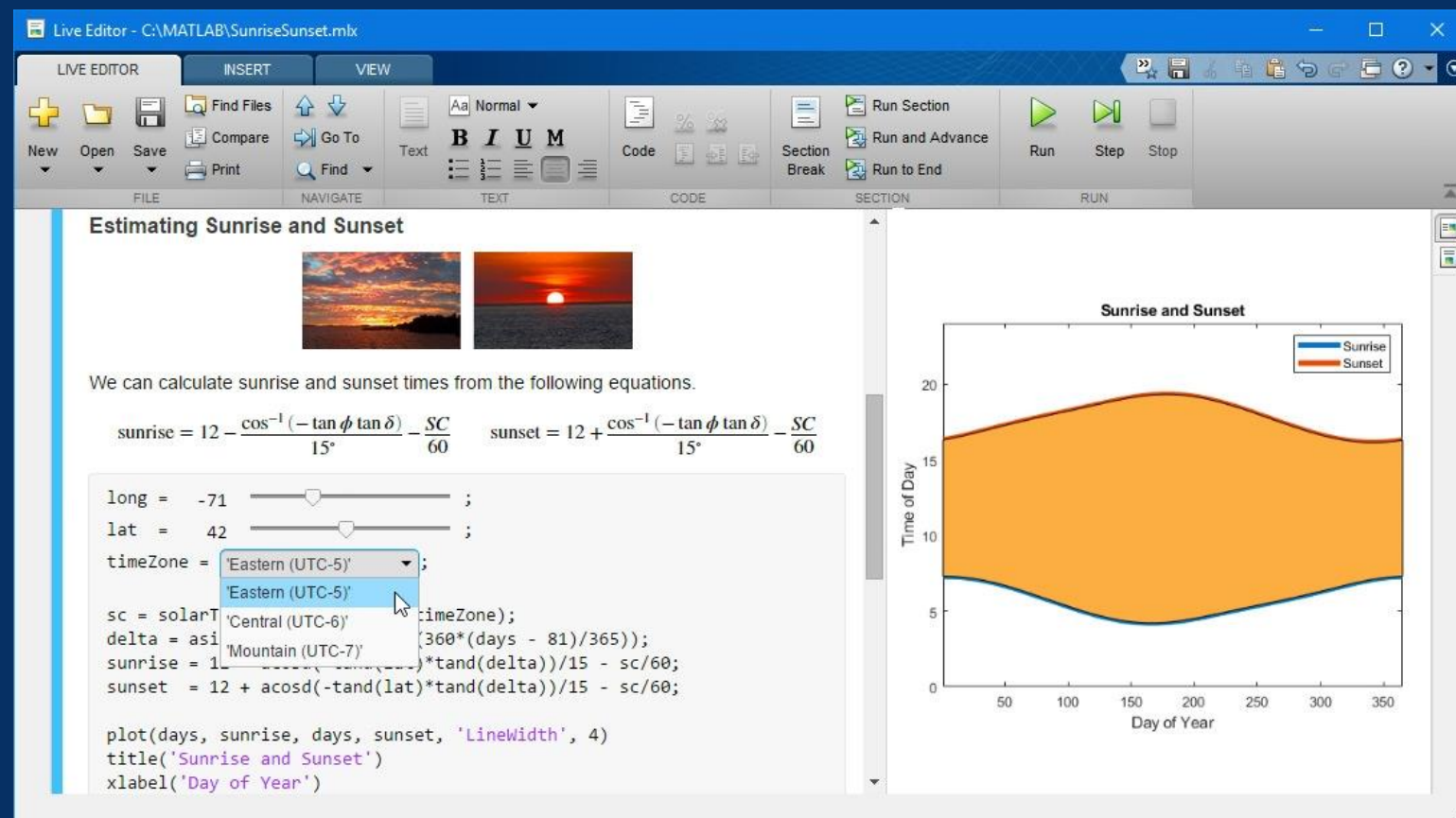
# Create executable notebooks for sharing, presenting, teaching

Code + Output + Formatted Text = **Executable Notebook**

Contextual hints  
while coding

View interactive outputs  
next to the code

Add rich text formatting,  
equations, images,  
and hyperlinks



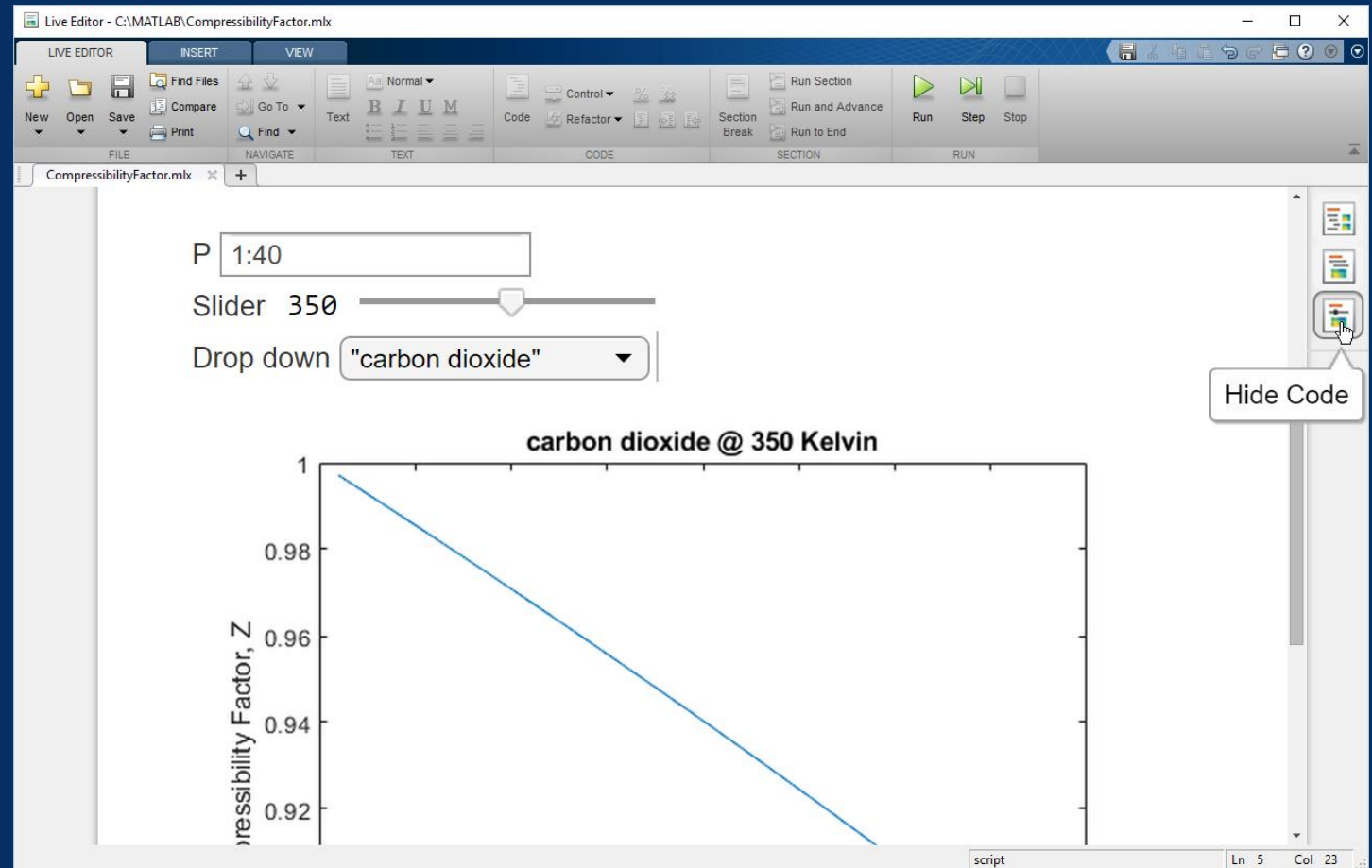
Live Editor

# Turn a script into a simple app

Add **interactive controls** to modify script variables

- Numeric sliders
- Drop-down lists
- Edit fields

Hide the code to create simple applications and dashboards



Live Editor

# Complete steps interactively

Use **tasks** to explore parameters and options

Automatically generate MATLAB code for the completed task

The screenshot displays the MATLAB Live Editor interface. The top toolbar includes tabs for LIVE EDITOR, INSERT, FIGURE, and VIEW. Below these are various toolbars for file operations, navigation, text formatting, code editing, and execution. The main workspace is divided into three panes:

- Left Pane (Task Configuration):** Shows the 'Select data' section with 'nyiso' as input data and 'DUNWOD' as the X-axis. The 'Specify method' section has 'Fill missing' and 'Linear interpolation' selected. The 'Display results' section has checkboxes for 'Cleaned data' and 'Filled missing entries'.
- Center Pane (Code Generation):** Displays the MATLAB code generated from the task. The code is enclosed in a blue box and includes comments and function calls for filling missing data and plotting the results.
- Right Pane (Figure):** Shows a line plot titled 'Number of filled missing entries: 4'. The plot displays 'Cleaned data' as a blue line and 'Filled missing entries' as red dots. The x-axis represents time from July 2007 to April 2008, and the y-axis represents the number of entries, ranging from 200 to 1600.

```
% Fill missing data
[cleanedData,missingIndices] = fillmissing(nyiso.DUNWOD,'linear',...
    'SamplePoints',nyiso.Date);

% Display results
clf
plot(nyiso.Date,cleanedData,'Color',[0 114 189]/255,'Linewidth',1.5,...
    'DisplayName','Cleaned data')
hold on

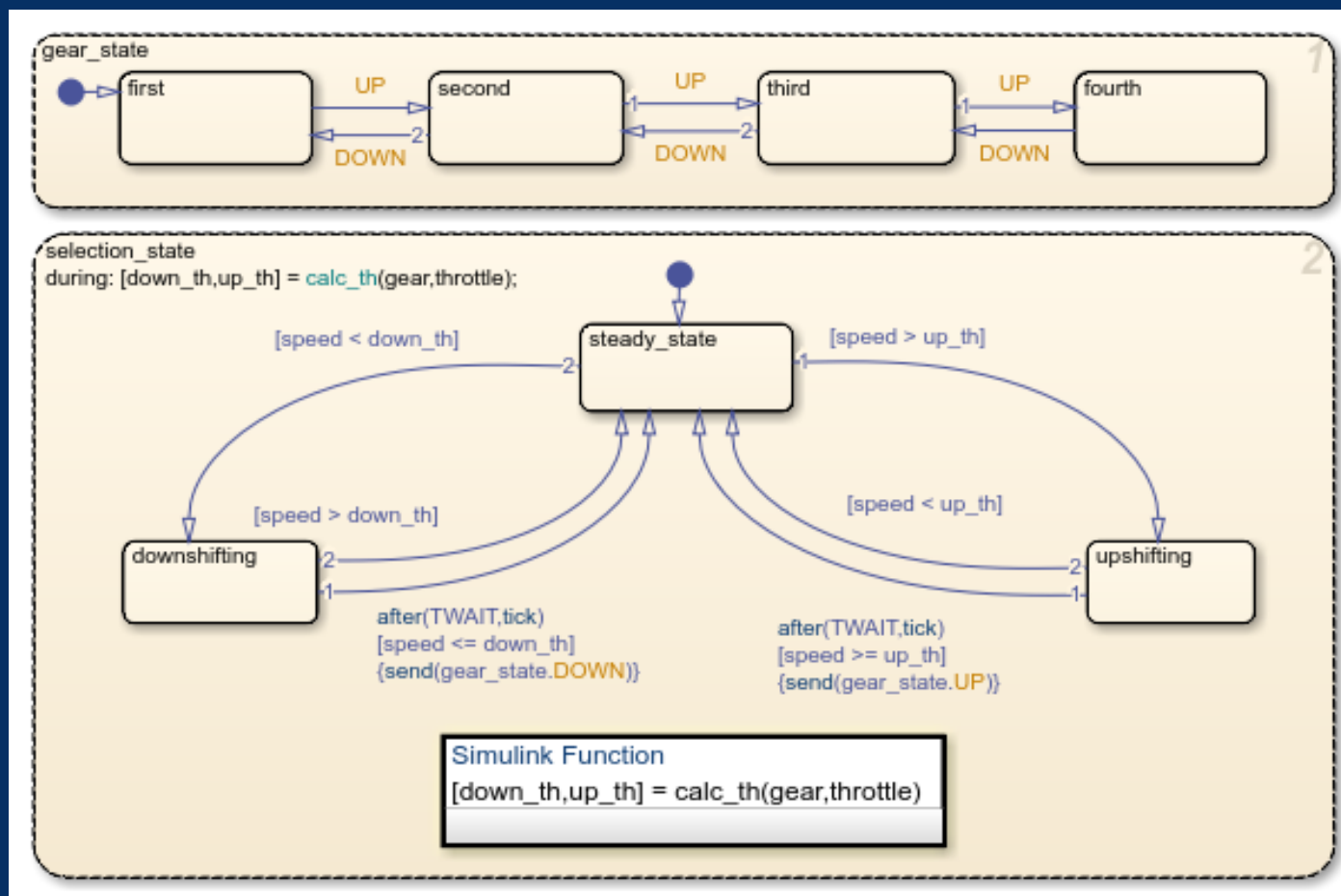
% Plot filled missing entries
plot(nyiso.Date(missingIndices),cleanedData(missingIndices),'.','MarkerSize',12,...
    'Color',[217 83 25]/255,'DisplayName','Filled missing entries')
title(['Number of filled missing entries: ' num2str(nnz(missingIndices))])

hold off
legend
clear missingIndices
```

Live Editor

# Design decision logic at a higher level of abstraction

Graphically program, debug  
and execute state machines



# Design decision logic at a higher level of abstraction – in MATLAB

The image displays the MATLAB Stateflow environment. The left pane shows a script editor with the following code:

```

1 %% Highway Simulator
2
3 % Reset random number generator to default values.
4
5 rngPrev = rng(0);
6
7 % Close Highway Scenario from previous run
8
9 if ~isempty(findobj('Name','Highway Scenario'))
10     close 'Highway Scenario';
11 end
12
13 % Open chart if Stateflow license exists
14
15 if license('test','stateflow')
16     % edit sf_driver.sfx;
17 end
18
19 %% Create road environment
20
21 numCars = 20;
22
23 scenario = HighwayScenario(...
24     'NumCars',numCars,...
25     'NumLanes',5,...
26     'LaneWidth',3,...
27     'Dt',1e-1,...
28     'AnimateChart',true,...
29     'Plot',true,...
30     'AverageDesiredSpeed',20,...
31     'Sigma',3,...
32     'EgoDesiredSpeed',30,...
33     'NumUserControlledCars',0);
34
35 %% Run simulation
36
37 for i = 1:5000
38     scenario.step();
39 end
40
41 %% Clean up
42
43 for i = 1:numCars
44     delete(scenario.Drivers(1));
45 end
46
47 delete(scenario);
48
49 rng(rngPrev);

```

The right pane shows a 3D visualization of the highway scenario, titled "Highway Scenario". It displays a perspective view of a multi-lane highway with several cars (represented as colored blocks) moving along it. The cars are colored orange, blue, and green.

Below the visualization is a "Symbols" table listing various variables and their values:

TYPE	NAME	VALUE
double	rWidth	3
double	target	-3
double	me	1
double	deltaLane	0
double	laneCenters	-6 -3 0
double	myLane	4
double	myPos	106.49
double	width	0
double	zoneCar	1
double	isZoneOccupied	1
double	positions	zeros(1
double	frontF	0.9
double	errLane	0
double	topLane	2
double	isLaneChanging	0
double	delay	13
double	numCars	
double	maxSpeed	30
double	myVel	18.644
double	velocities	[ 1 0 0]
double	slowCar	0
boolean	checkZone	
boolean	getLane	
boolean	getVel	

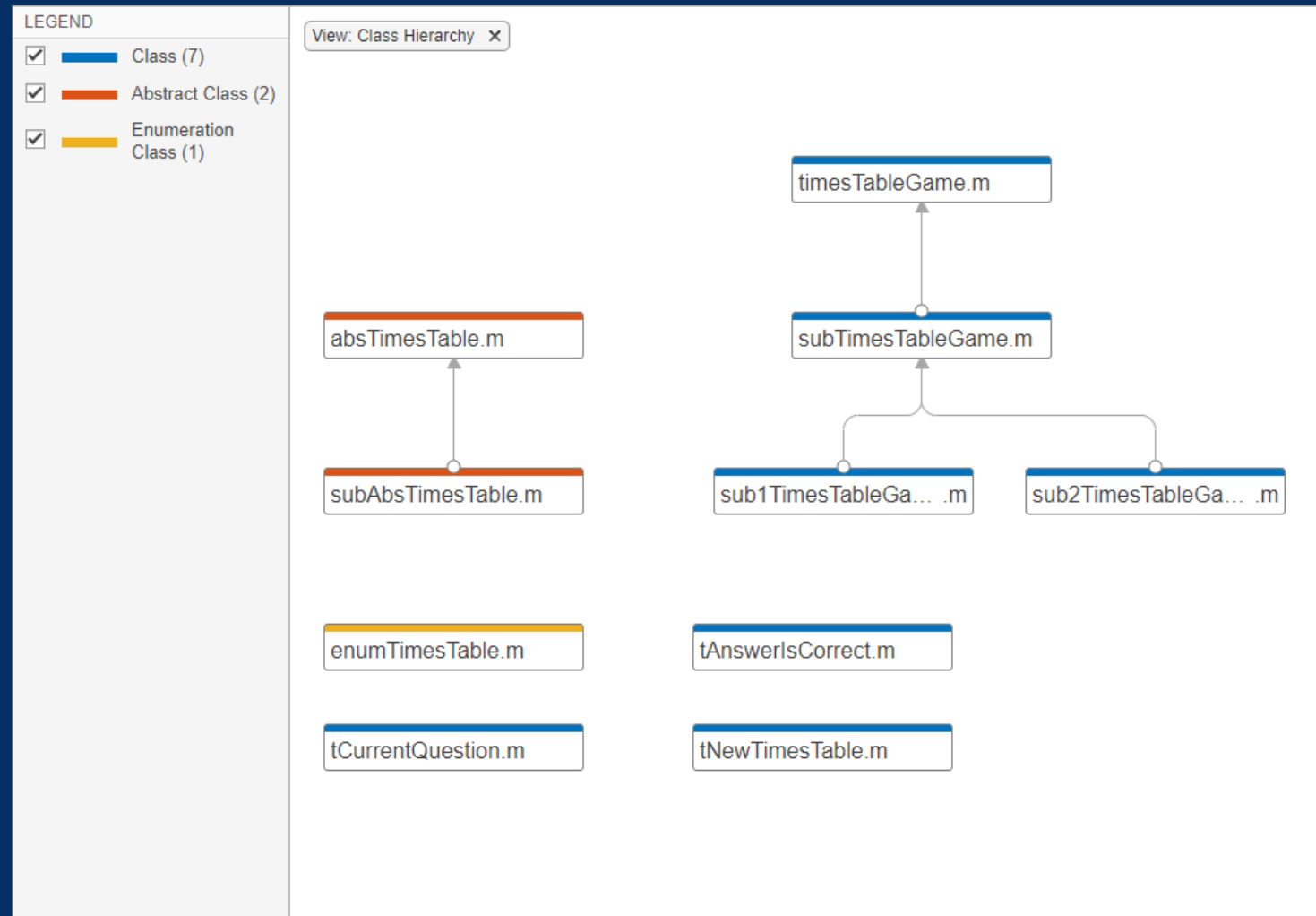
# Manage system complexity with projects

**Projects** in MATLAB help you to organize, manage, and share your code and models

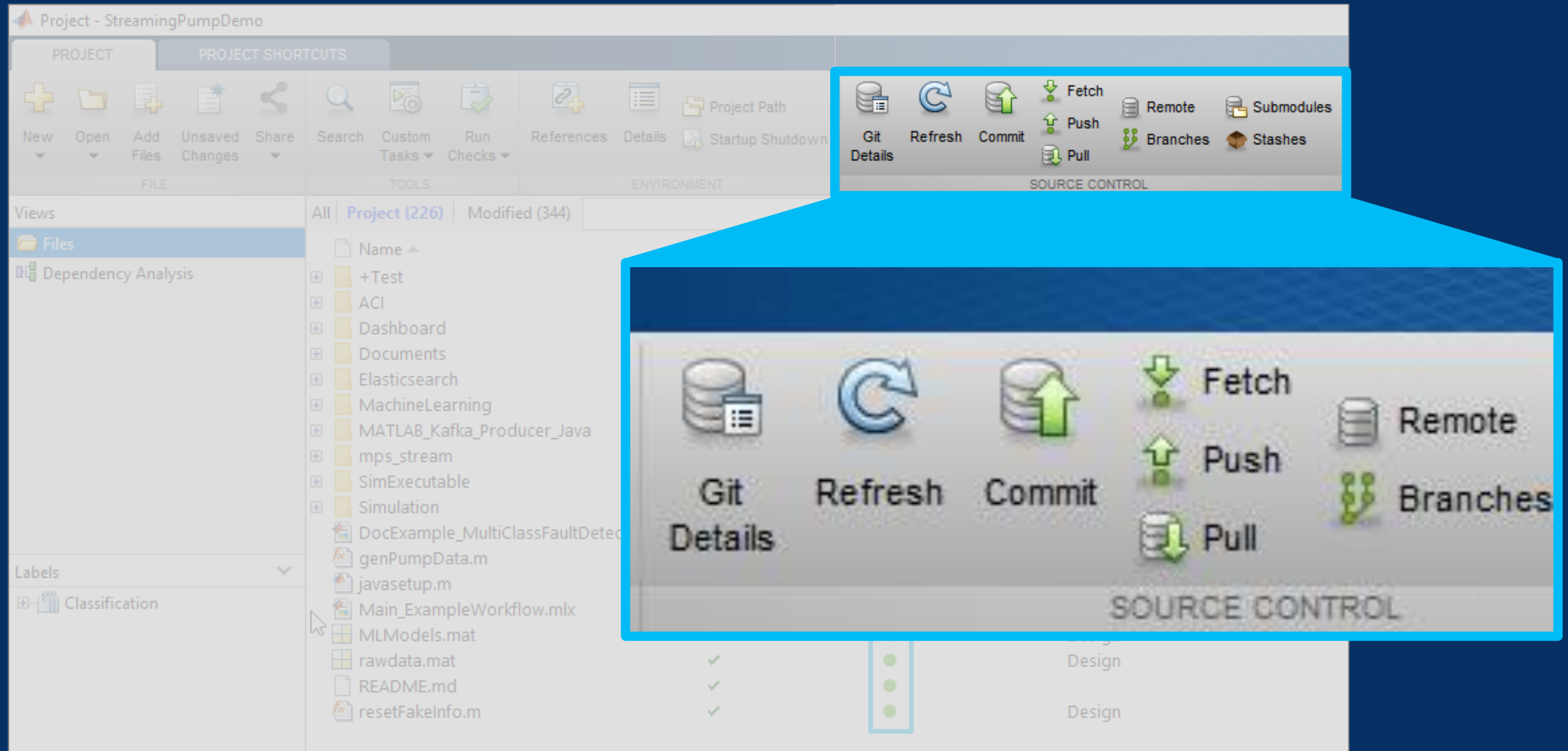


# Explore file dependencies and impact analysis

Explore and visualize project structure



# Use source control systems (Git, Subversion) with projects



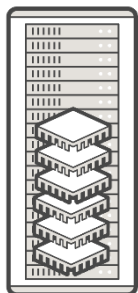
# Server Products

1. Overview of our server products
2. New WebApp Server
3. Scaling with reference architectures
4. Support DevOps workflows

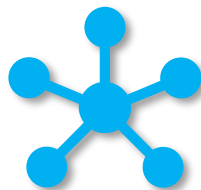
*“Server suite scales with  
your needs”*

# Big picture of our server products

I want to....



**Speed up  
computation**



**Integrate with  
enterprise  
applications**



**Publish  
Web Apps**



**Use MATLAB  
in a browser**



**MATLAB  
Parallel  
Server**



**MATLAB  
Production  
Server**



**MATLAB  
Web App  
Server**



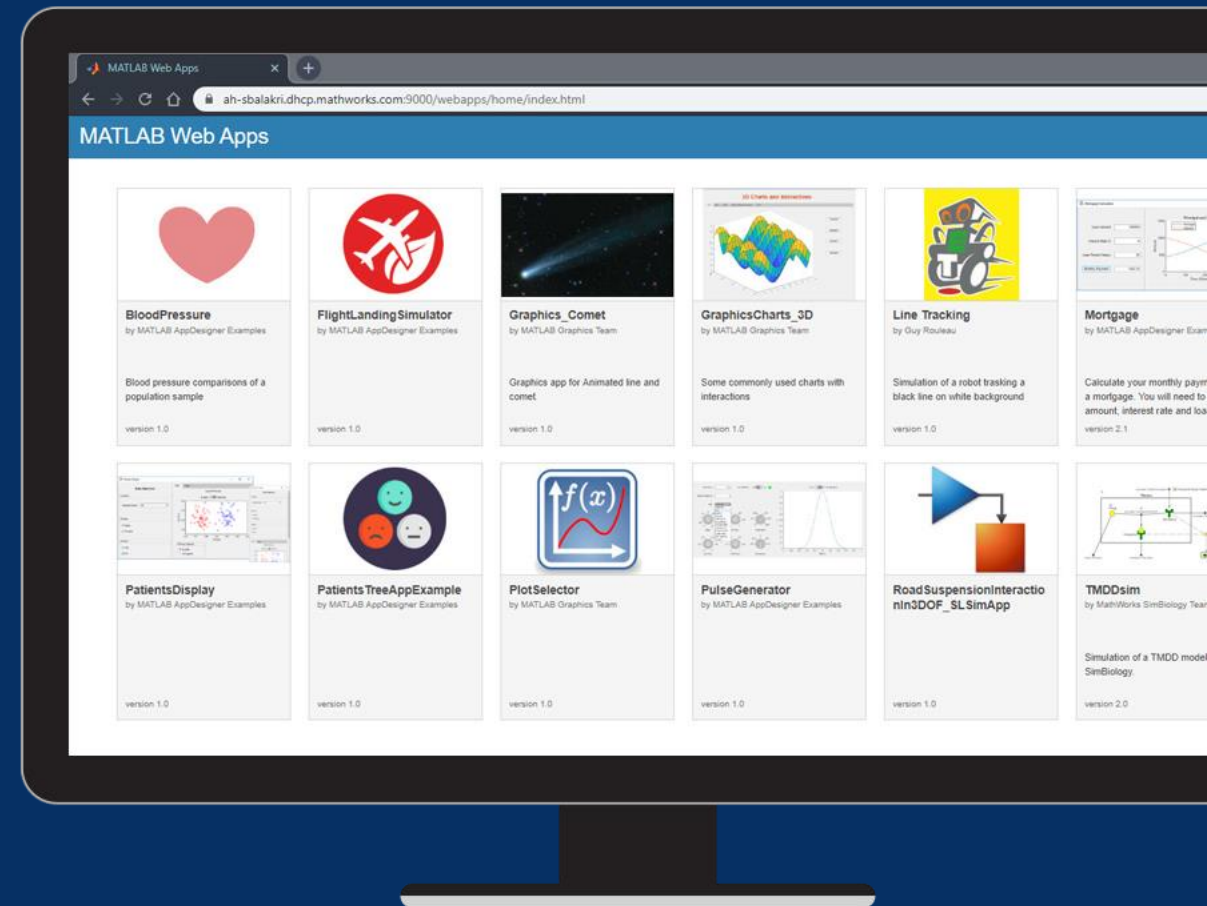
**MATLAB Online  
Server**

# Share MATLAB apps as browser-based web apps

Create apps using App Designer  
and host them using  
MATLAB Web App Server

For IT: full authentication support

For domain experts: upload your app  
on demand



# Scale as you need



Public Clouds



On-Premise/Private Cloud



Hosting Provider



MathWorks Cloud

**Reference  
Architectures**

**Cloud Data Services**

**Cloud Center**

**MATLAB Dockerfile**

 **NVIDIA NGC** Deep Learning Container

**MATLAB Online Server**

**NIMBIX**



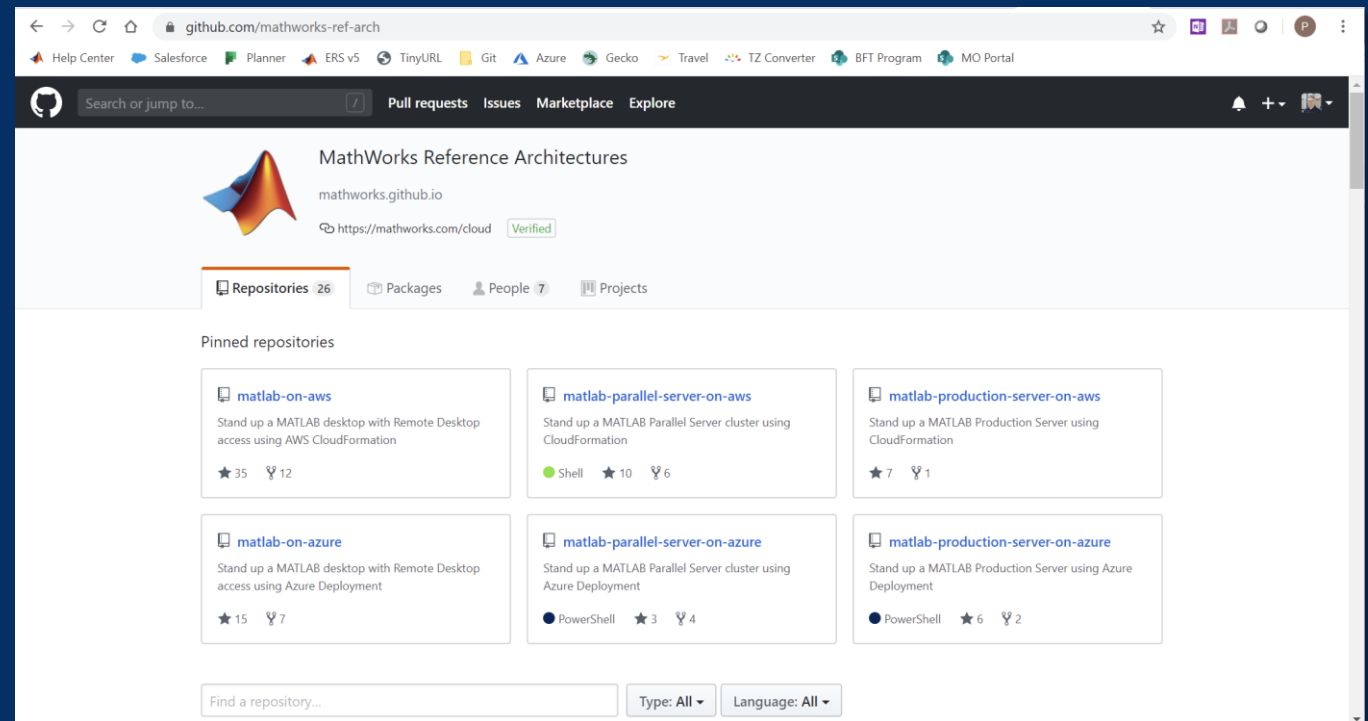
**SABALCORE**



**MATLAB Online  
MATLAB Drive**

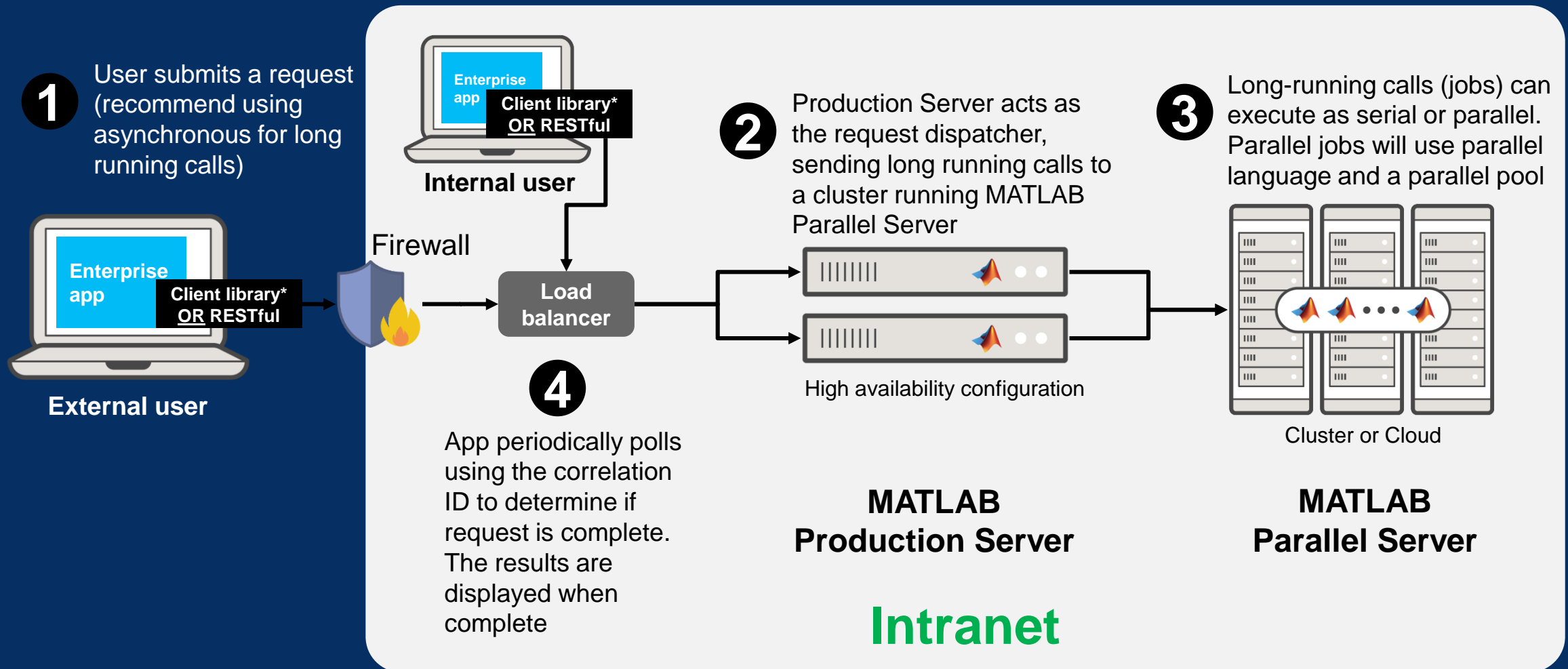
# Reference Architectures on GitHub

- Official MathWorks guidelines
- One click to deploy
  - After some initial setup 😞
- Monolithic Architectures
- Connectors/Utilities
- Containers



<https://github.com/mathworks-ref-arch>

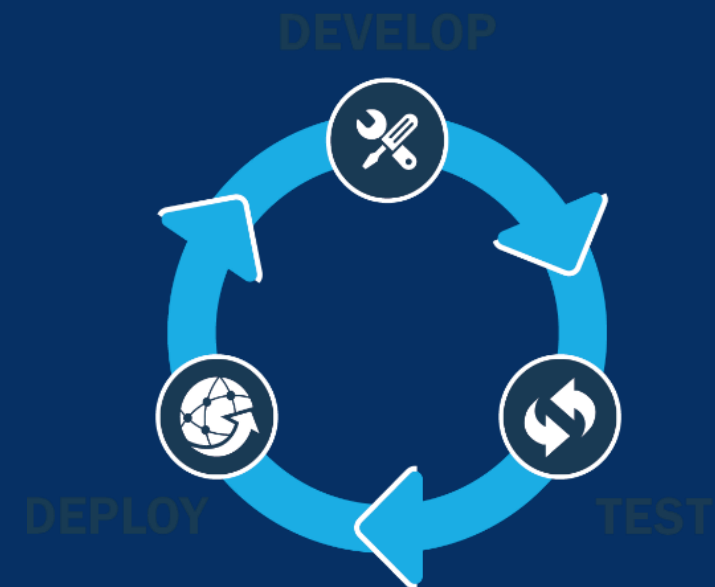
# Scale Out Production Server with Parallel Server



\* Provided by MathWorks <https://www.mathworks.com/products/matlab-production-server/client-libraries.html>

# Automated Testing with Continuous Integration (CI)

- A system to automate the building, testing, integration, and deployment of code as it is being developed and maintained
- Popular CI systems: Jenkins, Travis, CircleCI , Azure DevOps, and others...
- Benefits:
  - Detect integration bugs early
  - Allow you to stop bugs from being accepted
  - Track and report testing history
  - Flexible testing schedules and triggers





# MATLAB Plugin for Jenkins

- Install MATLAB Plugin for Jenkins directly from the Jenkins Plugin Manager
- Easily connect and configure MATLAB with Jenkins
- Schedule automatic code and model testing
  - MATLAB Unit Test Framework
  - Simulink Test

The screenshot shows the Jenkins Plugin Manager interface. At the top, there's a header with the Jenkins logo and the user name 'Adam Sifounakis' with a 'log out' link. Below the header, there are navigation links: 'Back to Dashboard' and 'Manage Jenkins'. A search filter is set to 'MATLAB'. The 'Available' tab is selected, showing a table of available plugins. The 'MATLAB' plugin is listed with a version of 1.1.2 and a description: 'This plugin integrates MATLAB (R) with Jenkins and provides Jenkins interface to run MATLAB and Simulink (R) tests.' Below the table, there are two buttons: 'Install without restart' (highlighted with an orange box) and 'Download now and install after restart'. A 'Check now' button is also present. At the bottom, it says 'Update information obtained: 4 min 43 sec ago'.

Install ↓	Name	Version
<input checked="" type="checkbox"/>	<a href="#">MATLAB</a> This plugin integrates MATLAB (R) with Jenkins and provides Jenkins interface to run MATLAB and Simulink (R) tests.	1.1.2

[Install without restart](#) [Download now and install after restart](#) [Check now](#)

Update information obtained: 4 min 43 sec ago

Page generated: Feb 20, 2020 4:13:08 PM EST [REST API](#) [Jenkins ver. 2.204.2](#)



# Testing Reports in Jenkins

- View testing results
- View code coverage
- View testing reports

### Test Result

1 failures (+1)

3 tests (±0)  
Took 1.6 sec.  
[add description](#)

### All Failed Tests

Test Name

[designTest.testOvershoot](#)

Stack Trace

```

=====
Verification failed in designTest/testOvershoot.m:
-----
Framework Diagnostic:
verifyLessThan failed.
--> The value must be less than
-----
Actual Value:
0.082943282378937
Maximum Value (Exclusive):
0.0100000000000000
-----
Stack Information:
-----
In /Users/Shared/Jenkins/Home/jenkins/workspace/Project_Mass-Spring-Damper/workspace/tests/designTest.m
=====

```

### File Coverage summary

Name	Classes	Lines	Conditionals
simulateSystem.m	100% 1/1	90% 9/10	100% 0/0

### Coverage Breakdown by Class

[simulateSystem](#) 90%

### Source

```

simulateSystem.m
1 function [x, t] = simulateSystem
2
3 2 springMassDamperDesign; % Create
4
5 2 if ~isstruct(design) || ~all(isfinite(design))
6 0 error('simulateSystem:InvalidDesign','The design should be a struct with fields: c, k, m, z0')
7
8 end
9
10 % Design variables
11 2 c = design.c;
12 2 k = design.k;
13
14 % Constant variables
15 2 z0 = [-0.1; 0]; % Initial Position
16 2 m = 1500; % Mass
17
18 2 odefun = @(t,z) [0 1; -k/m -c/m]*z + z0;
19 2 [t, z] = ode45(odefun, [0, 3], z0);
20
21 % The first column is the position
22 2 x = z(:, 1);

```

### Project Mass-Spring-Damper

[add description](#)  
[Disable Project](#)

[Coverage Report](#)  
[Workspace](#)  
[Recent Changes](#)

### Permalinks

- Last build (#89). 1 mo 24 days ago
- Last stable build (#88). 1 mo 29 days ago
- Last successful build (#88). 1 mo 29 days ago
- Last failed build (#89). 1 mo 24 days ago
- Last unstable build (#52). 2 mo 7 days ago
- Last unsuccessful build (#89). 1 mo 24 days ago
- Last completed build (#89). 1 mo 24 days ago

### TAP Tests

TAP Tests Count

Failed Passed Skipped ToDo

### Code Coverage

100% Files 100% Classes 100% Lines 93% Conditionals

%

Classes Conditionals Files Lines Packages

### Test Result Trend

Count

(just show failures) [enlarge](#)

# AI: Machine and Deep Learning

1. The MATLAB Advantage
2. Deep Network Designer App
3. Experiment Manager App
4. AutoML

*“Apps to simplify your dev cycle with code to scale”*



# Artificial Intelligence (AI)

MATLAB

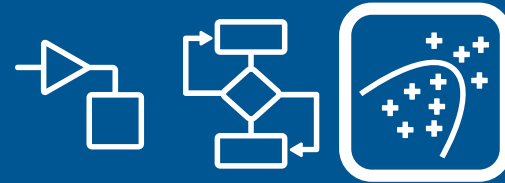
Access Data



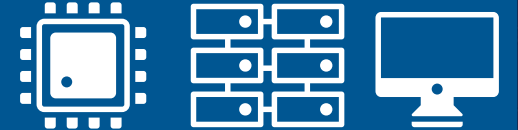
Preprocess Data



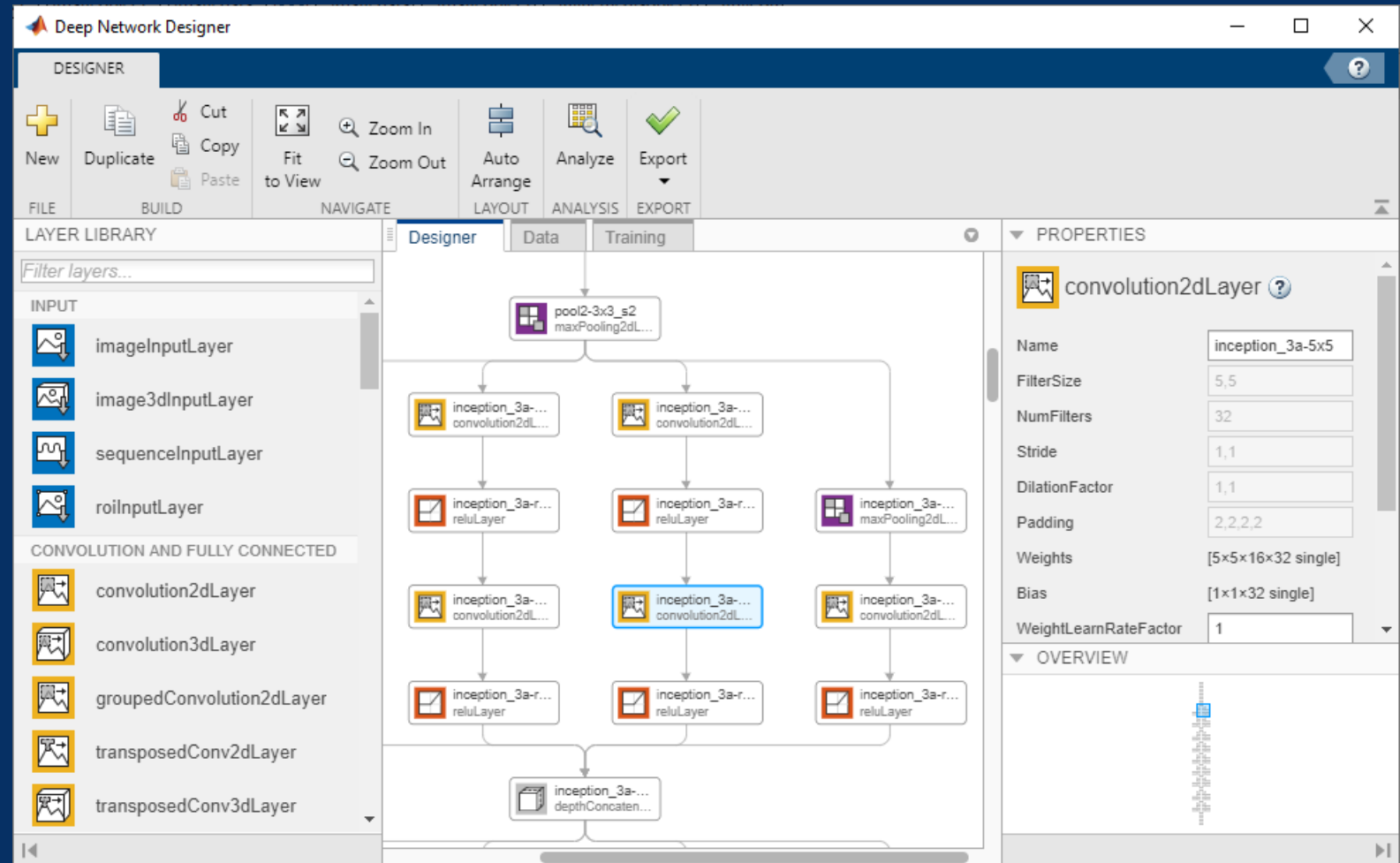
Develop



Deploy



# Interactively access models, and develop and train networks

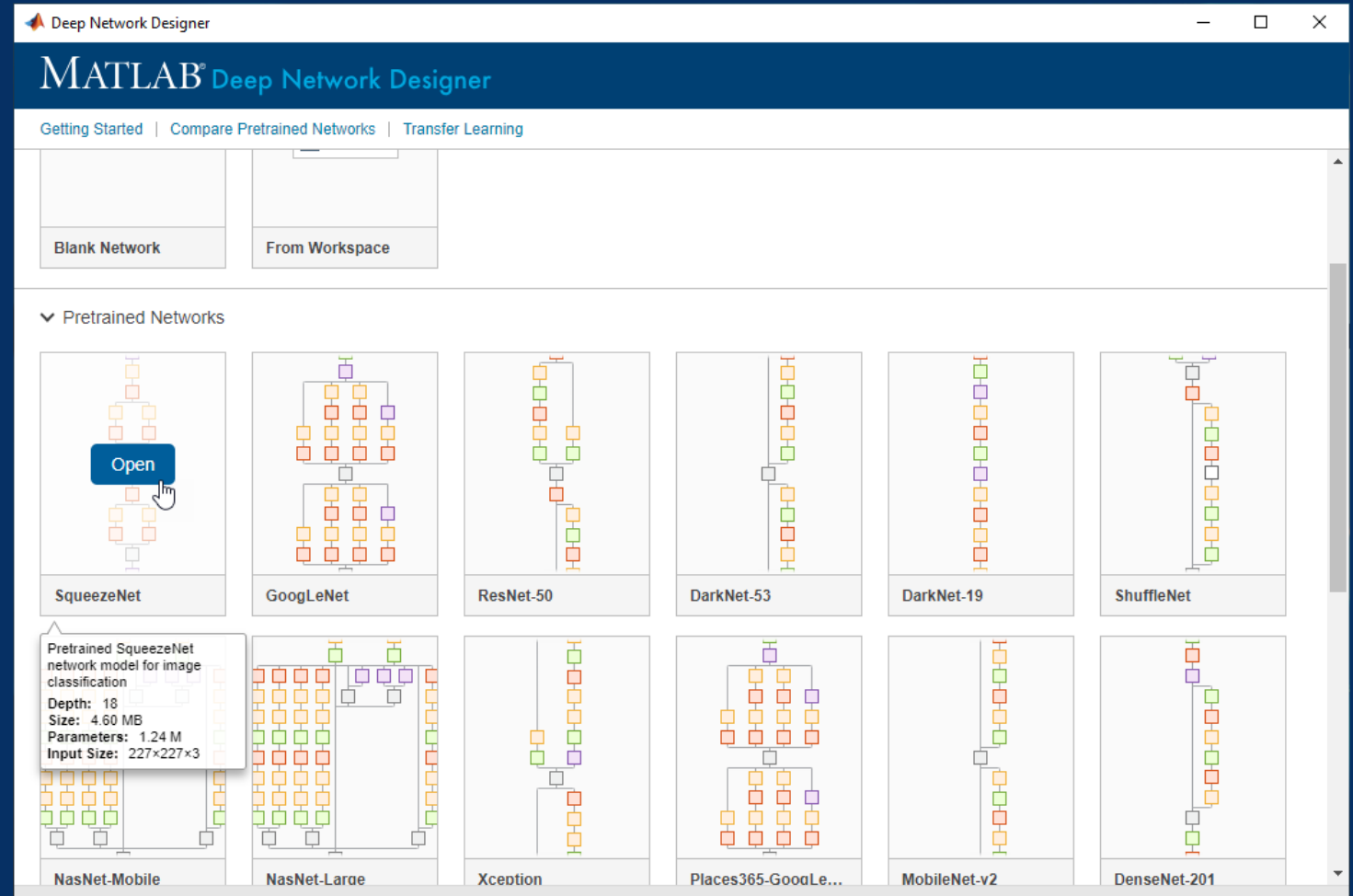


Deep Network Designer App

# Interactively access models, and develop and train networks



Import pretrained  
networks for transfer  
learning



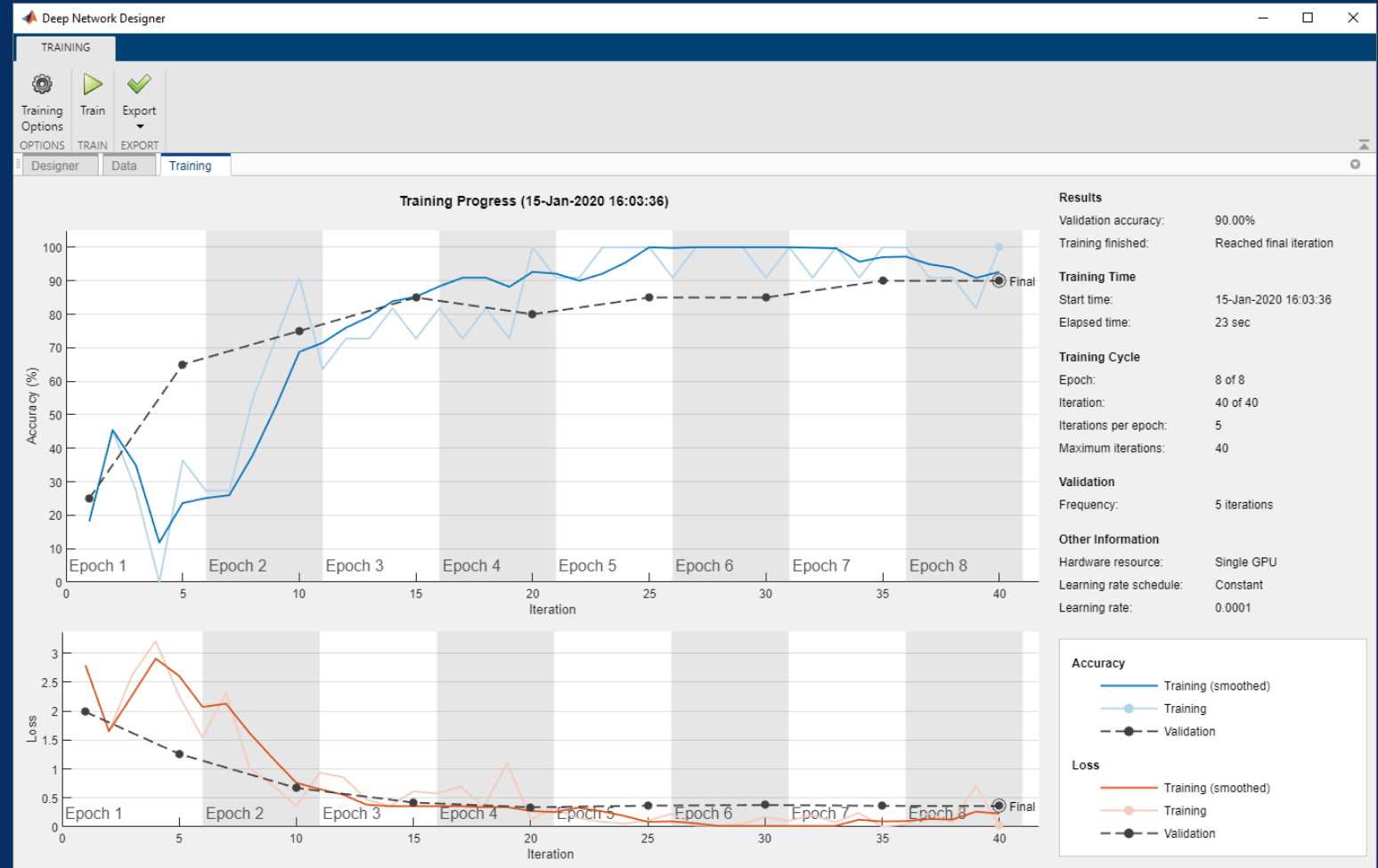
Deep Network Designer App

# Interactively access models, and develop and train networks



Import pretrained networks for transfer learning

Train networks and generate MATLAB code



Deep Network Designer App

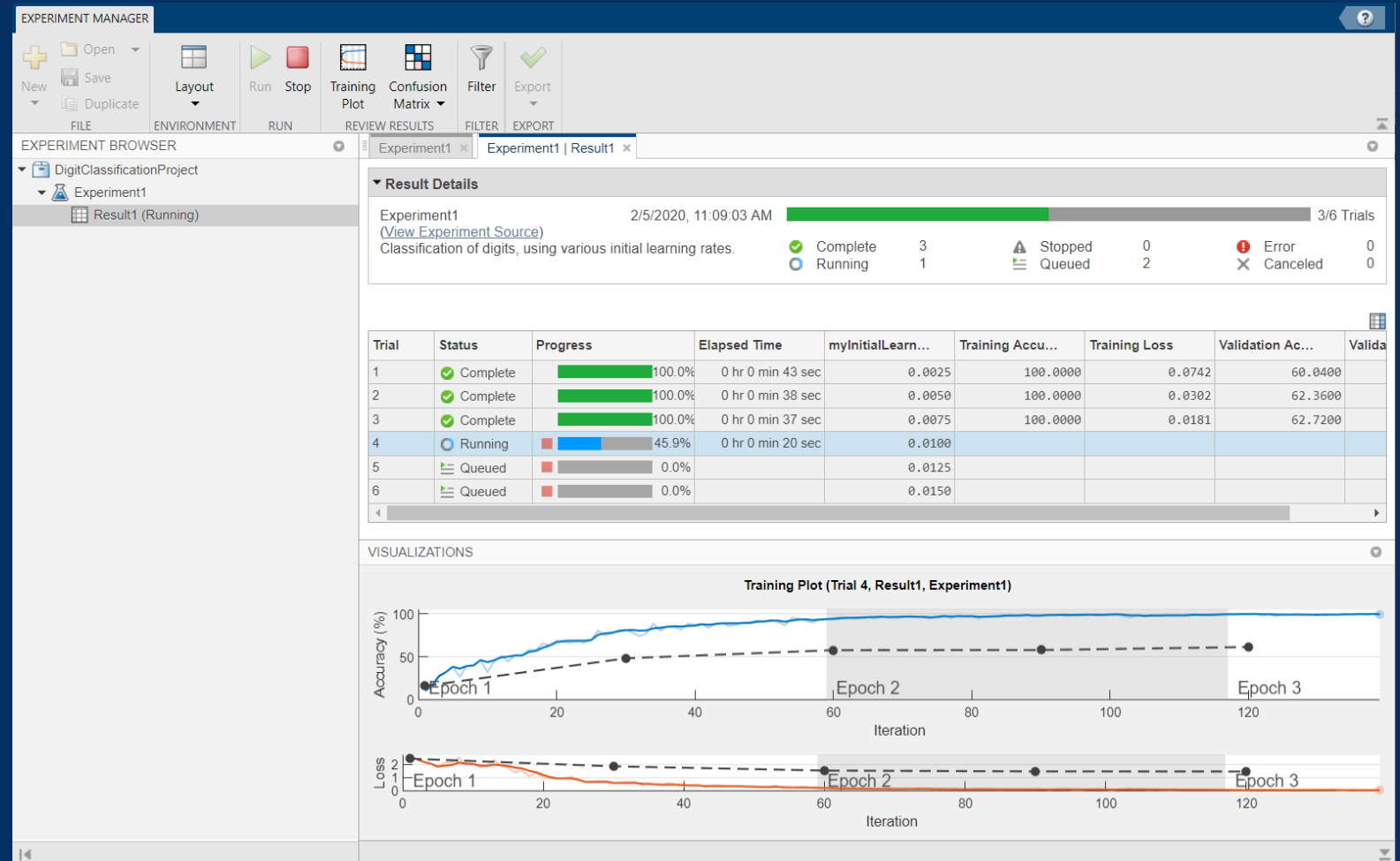


# Manage multiple deep learning experiments

Keep track of training parameters

Reuse training data across multiple networks

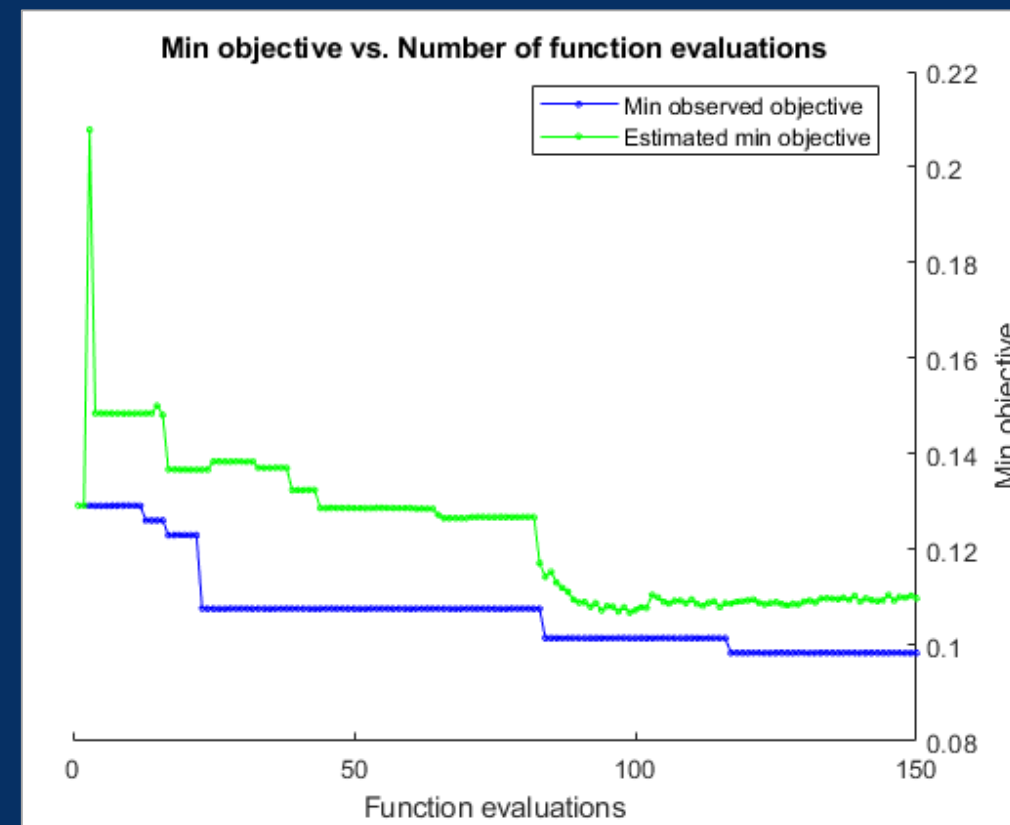
Analyze and compare results



Experiment Manager App

# Machine Learning – AutoML

- Automated machine learning
  - Choose a classification model automatically, across a selection of classifier types and hyperparameter values (**fitcauto**)
  - Uses Bayesian optimization



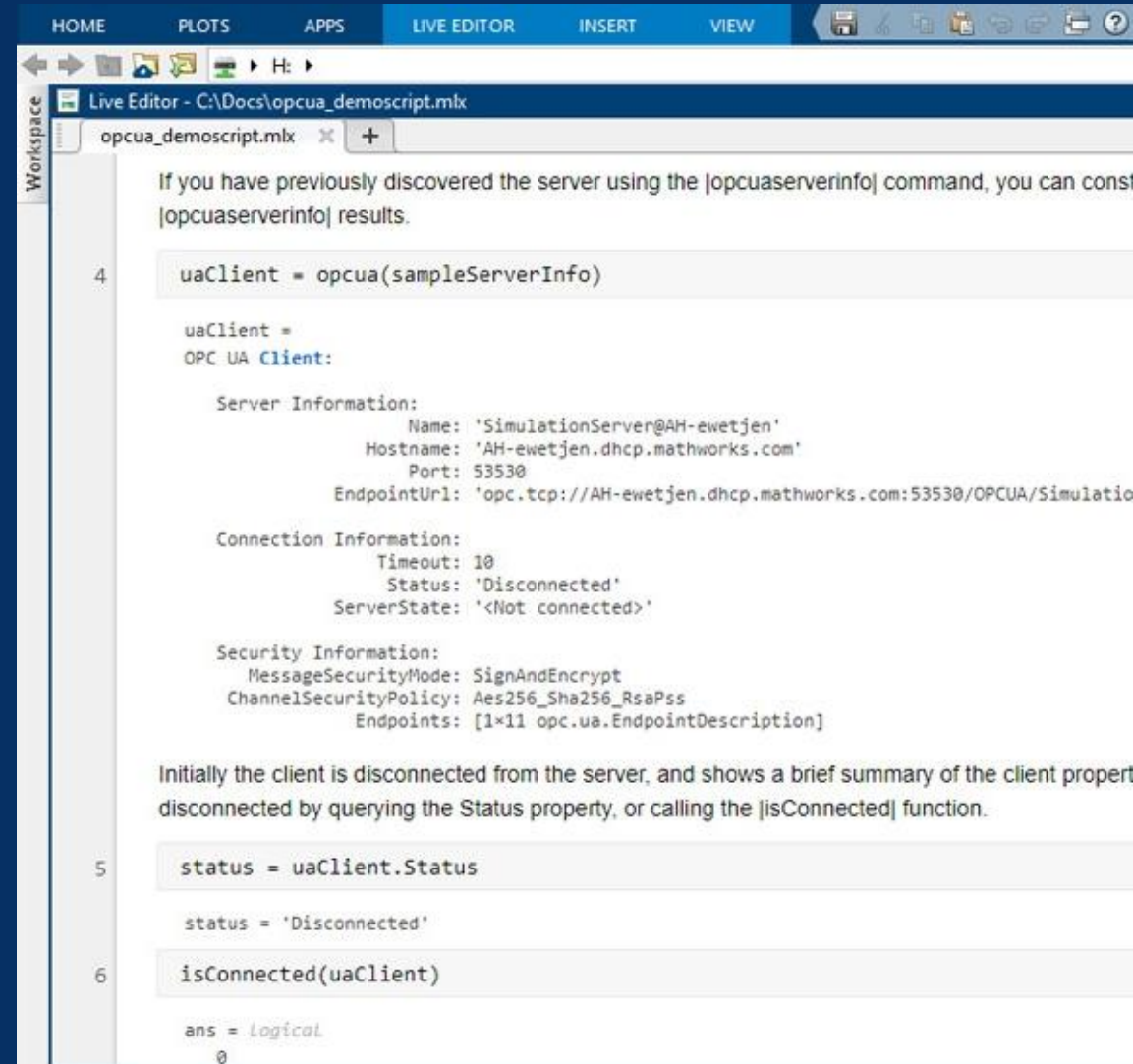
# Industrial Tools

1. OPC UA
2. Feature Extraction

*“Secure data connectivity  
with MATLAB analytics  
drives your insights”*

# Access plant data securely from OPC UA servers

- Establish secure OPC UA connections
  - Authenticate with username and password or X509 user certificate credentials
  - Sign and encrypt messages

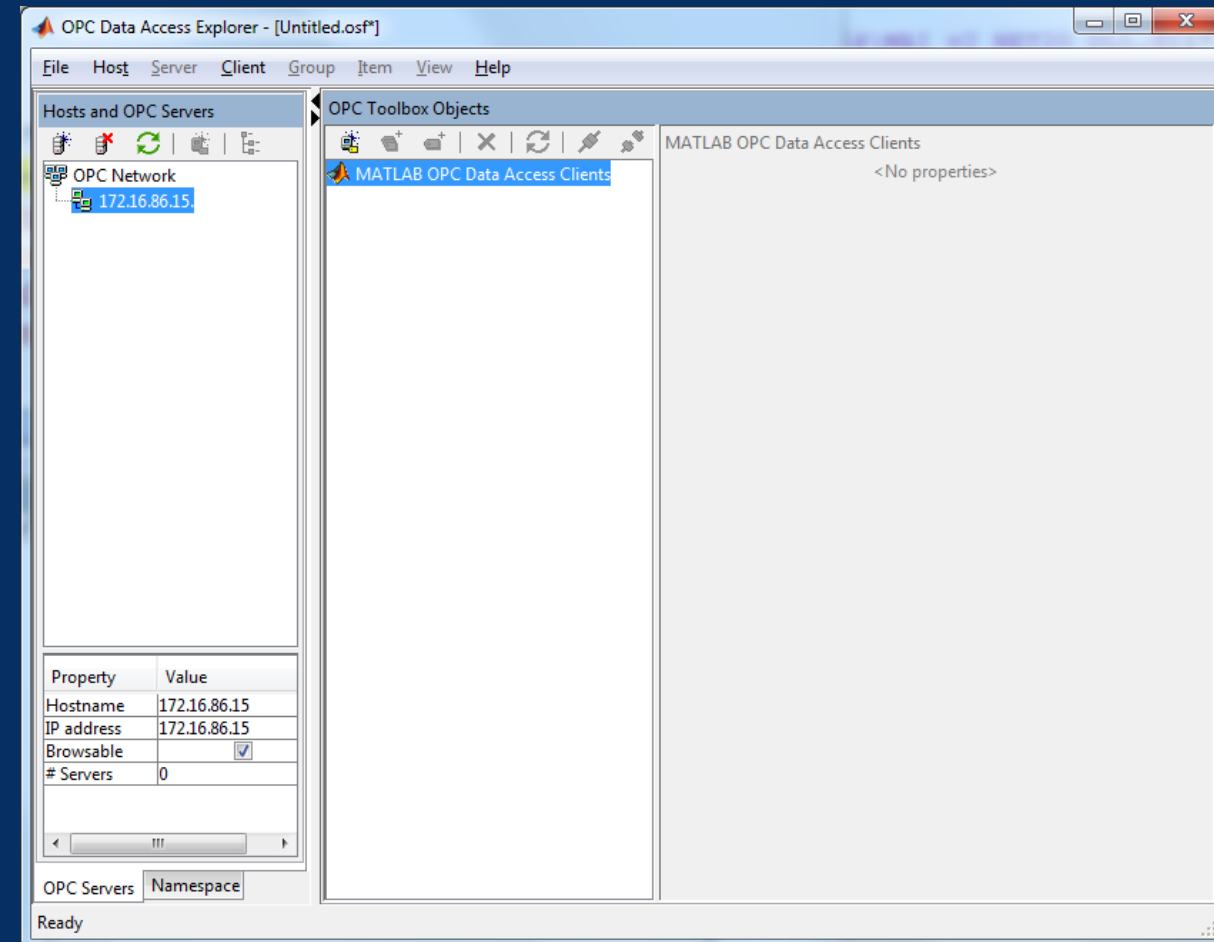


The screenshot shows the MATLAB Live Editor interface with a script titled 'opcua\_demo.mlx'. The script defines an OPC UA client and displays its properties. The output shows the client is disconnected.

```
Live Editor - C:\Docs\opcua_demo.mlx
opcua_demo.mlx x +
Workspace
If you have previously discovered the server using the [opcuaServerInfo] command, you can construct the client using the [opcuaClient] function.
4 uaClient = opcua(sampleServerInfo)
uaClient =
  OPC UA Client:
    Server Information:
      Name: 'SimulationServer@AH-ewetjen'
      Hostname: 'AH-ewetjen.dhcp.mathworks.com'
      Port: 53530
      EndpointUrl: 'opc.tcp://AH-ewetjen.dhcp.mathworks.com:53530/OPCUA/SimulationServer'
    Connection Information:
      Timeout: 10
      Status: 'Disconnected'
      ServerState: '<Not connected>'
    Security Information:
      MessageSecurityMode: SignAndEncrypt
      ChannelSecurityPolicy: Aes256_Sha256_RsaPss
      Endpoints: [1x11 opc.ua.EndpointDescription]
Initially the client is disconnected from the server, and shows a brief summary of the client properties. You can check the client status by querying the Status property, or calling the [isConnected] function.
5 status = uaClient.Status
status = 'Disconnected'
6 isConnected(uaClient)
ans = logical
0
```

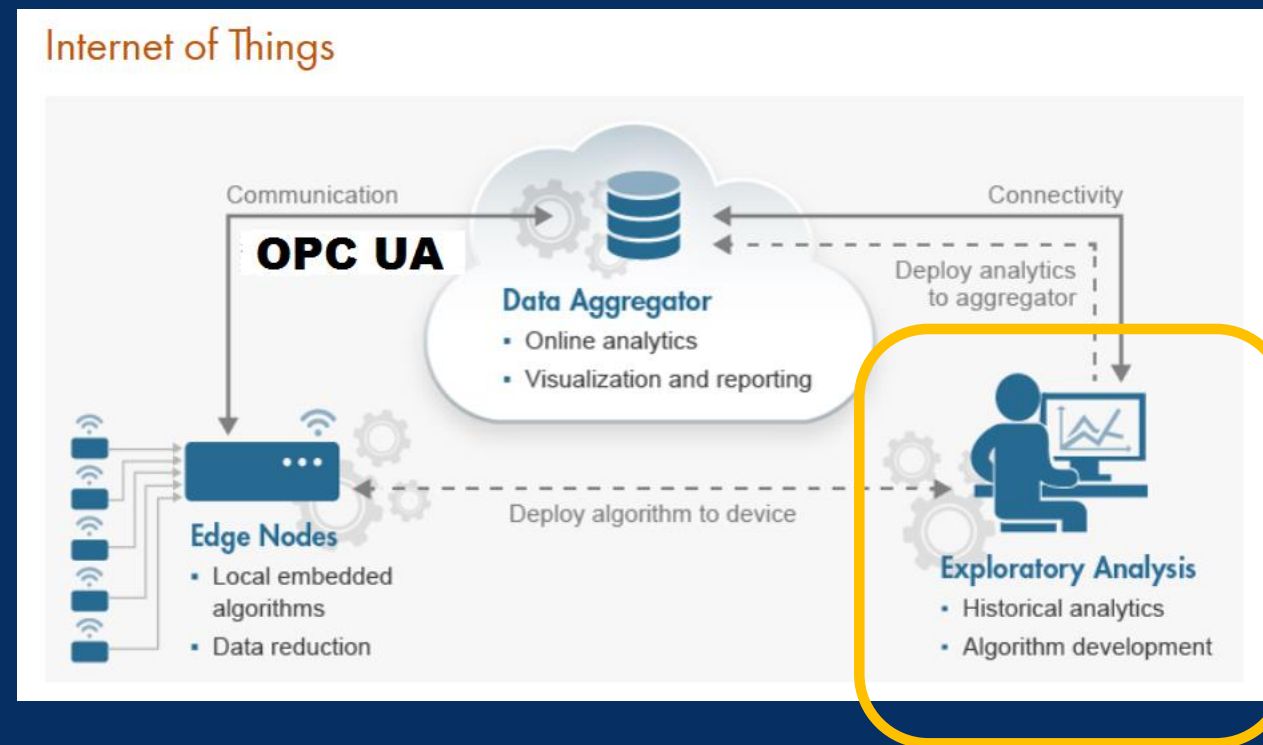
# OPC Data Access Explorer App

- Rapidly connect
- Visually browse data
- Export data to the workspace



# OPC UA drives IoT

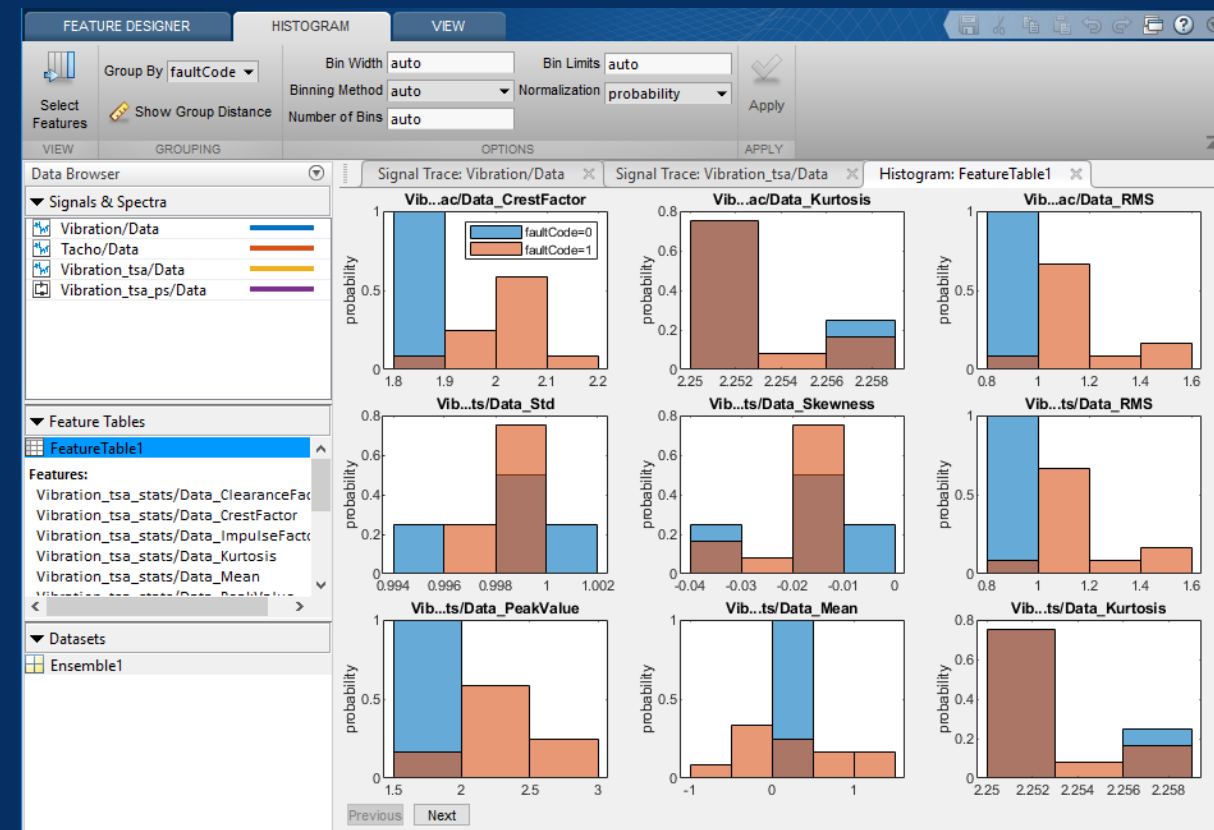
- OPC UA is vendor neutral
- Access PLCs or data aggregators directly from MATLAB and Simulink



Enables predictive maintenance workflows

# Diagnostic Feature Designer

- Import multiple data sources
- Interactively visualise base features
- Generate new features, e.g. timeseries
- Prognostic ranking to estimate remaining useful life
- Automatically generate code



# Financial Tools

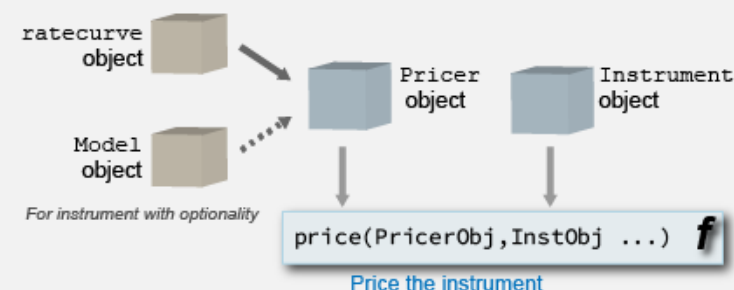
1. Object Framework for Pricing
2. Econometrics Models

*“Object Orientated  
framework encourages  
speed and code reuse”*

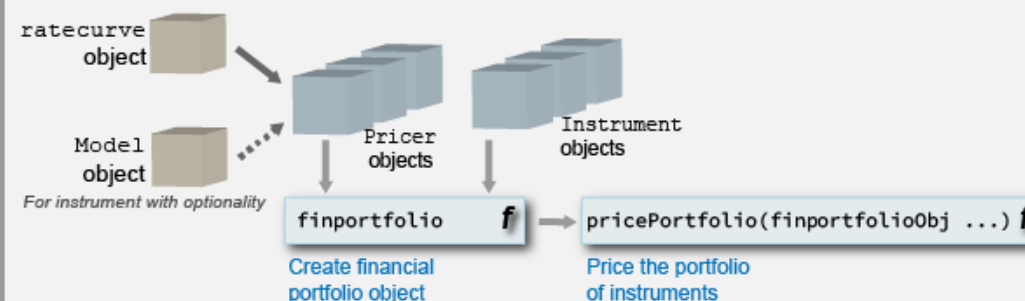
# Pricing and Valuation

- Price various types of financial instruments individually or collectively as a portfolio using new **object-oriented framework**
- The object-based workflow is an alternative to pricing financial instruments using functions
- Modular objects can easily be reused to compare instrument prices for different models and pricing engines

## Price an Instrument

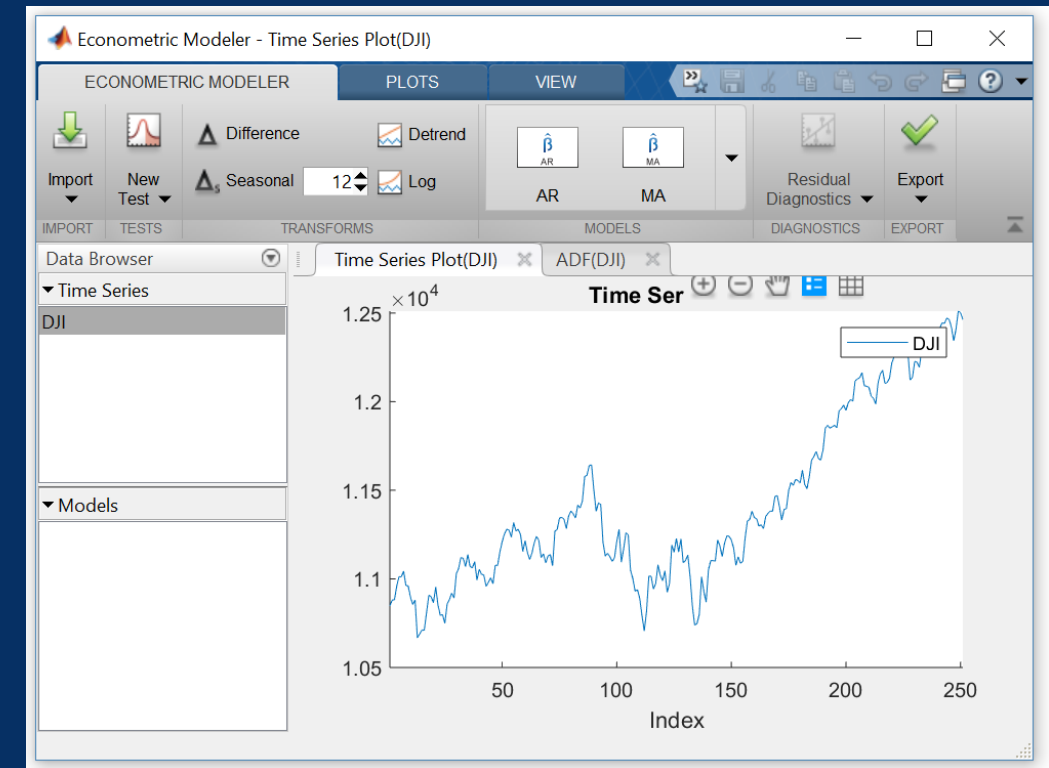


## Price a Portfolio of Instruments



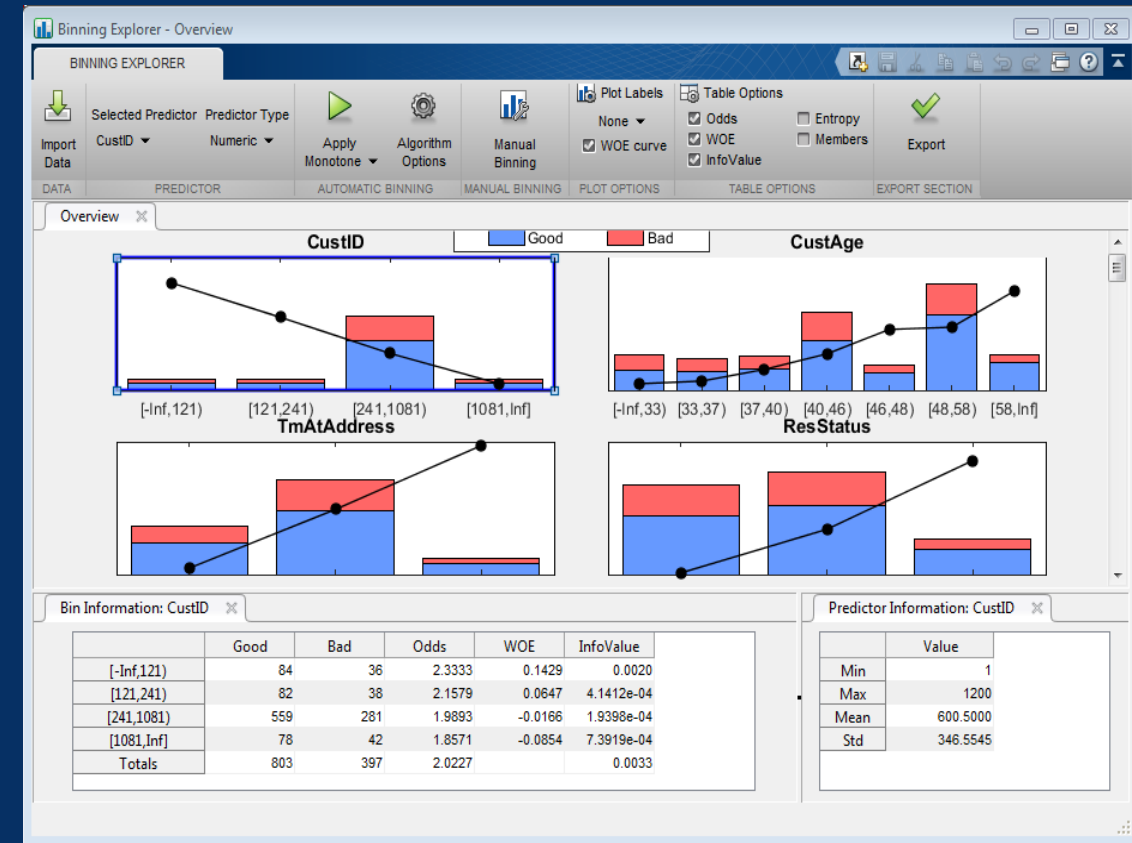
# Econometrics Toolbox

- Bayesian vector autoregression models
- Markov-switching autoregression models
- Granger Causality Test



# Risk Management Toolbox

- Support for constraints in credit scorecards
- Predictor screening for credit scorecards
  - Support for data that is too big to fit in memory (Big Data)



2,611

# Summary

1. Core MATLAB
  1. Live scripts; Stateflow; Projects
2. Server Suite
  1. Four core products; Web Apps Server; Reference architectures
3. AI Applications
  1. Deep Network Designer; Experiment; AutoML
4. Industrial Tools
  1. OPC UA, Diagnostic Feature Designer
5. Financial Models
  1. New object framework