

What's New in MATLAB for Finance Professionals

R2021a

Steve Notley

Application Engineer, Financial Services
snotley@mathworks.com

Greg McGean

Account Manager, Financial Services
gmcgean@mathworks.com

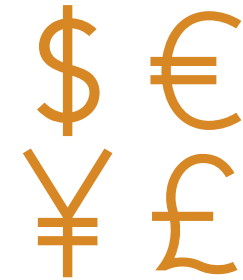
Agenda



Develop as Fast as you Think



Make Sense of your Data



Speak Finance



Think in Models



Engineer Robust Models



Collaborate and Share

Agenda



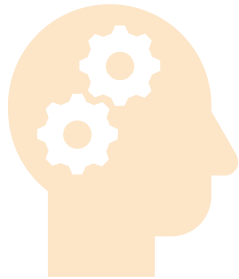
Develop as Fast as you Think



Make Sense of your Data



Speak Finance



Think in Models



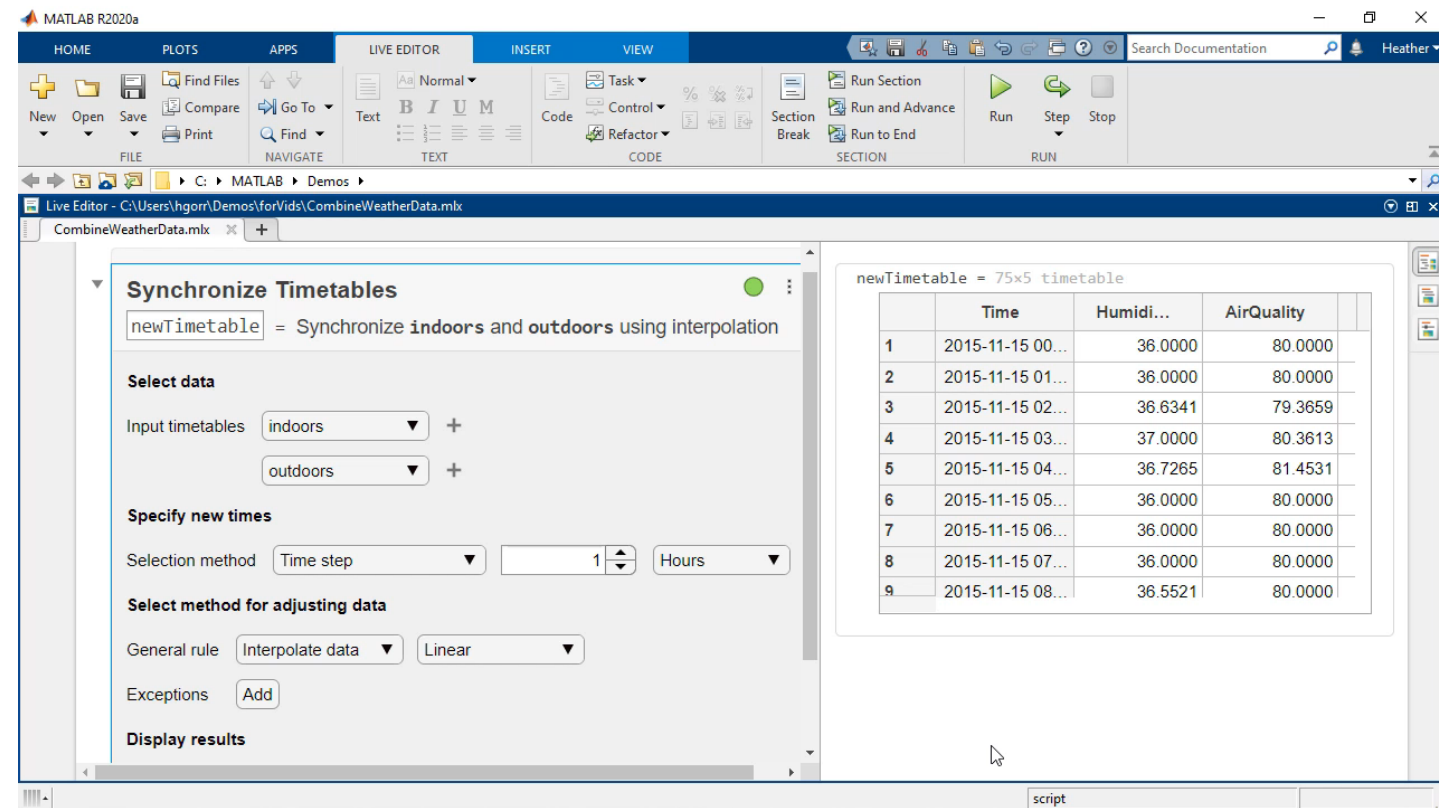
Engineer Robust Models



Collaborate and Share

Automating Even More of Your Tasks

- New tasks available:
 - **MATLAB**
 - Interactively manipulating tables and timetables (R2020a)
 - Interactively Plot Data (R2021a)
 - **Optimization Toolbox**
 - Interactively define and solve optimization problems (R2020b)
 - **Symbolic Math Toolbox**
 - Solve symbolic equations and simplify symbolic expressions (R2020a)



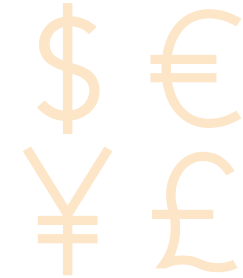
Agenda



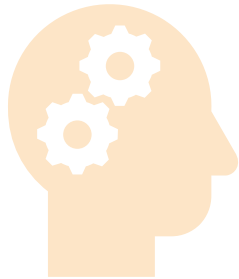
Develop as Fast as you Think



Make Sense of your Data



Speak Finance



Think in Models



Engineer Robust Models



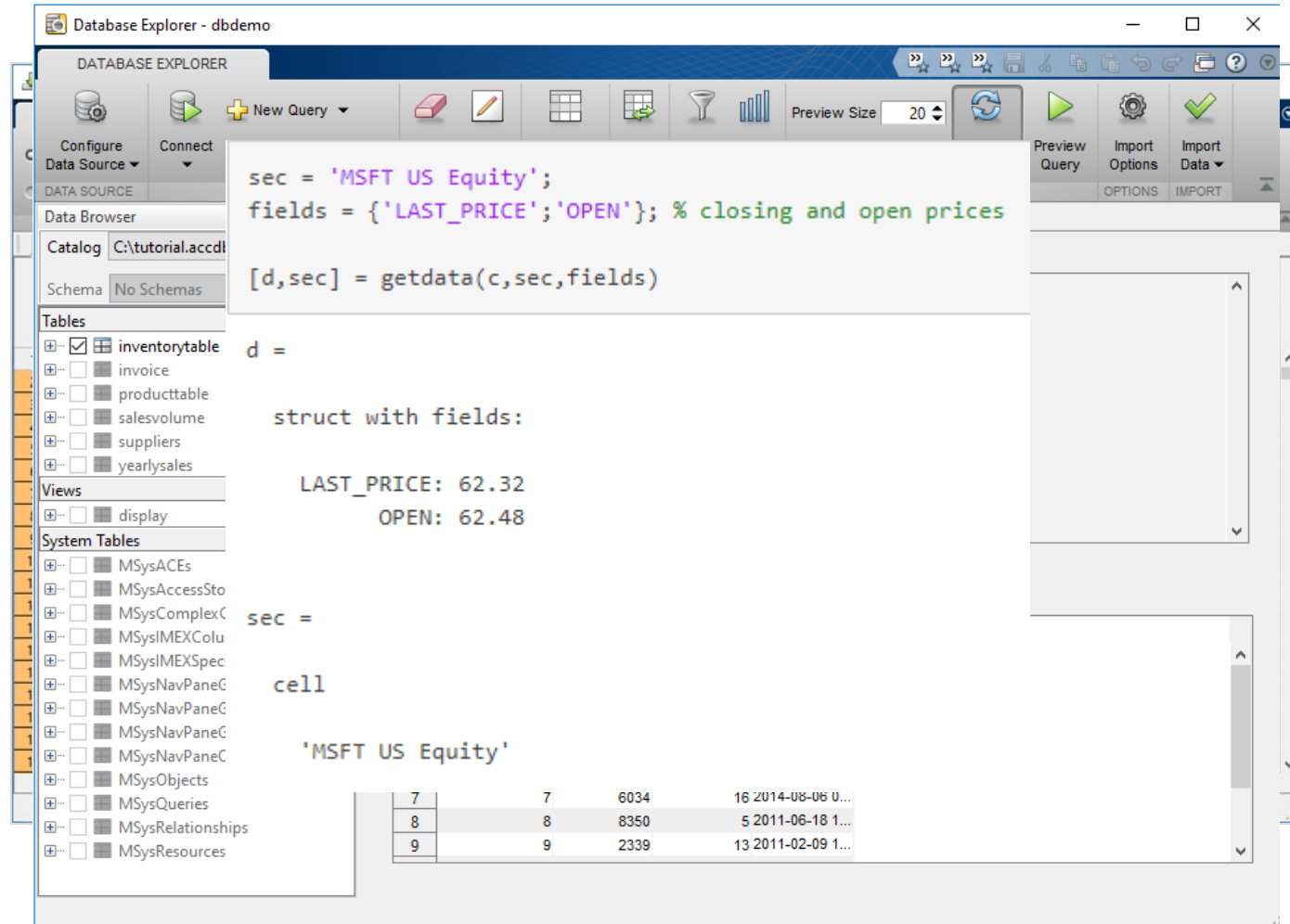
Collaborate and Share

Getting Your Data into MATLAB

- I have data in a file
 - Use the import tool to import data
 - Generate code for similar files

- I have a database
 - Use Database Explorer app for SQL to view and query*
 - Or connect and query via code*

- I need live data
 - MATLAB can connect to a host of live datafeeds**



The screenshot shows the Database Explorer interface with the following content:

```

sec = 'MSFT US Equity';
fields = {'LAST_PRICE'; 'OPEN'}; % closing and open prices

[d, sec] = getdata(c, sec, fields)

```

The output shows:

```

d =

struct with fields:

    LAST_PRICE: 62.32
         OPEN: 62.48

sec =

cell

    'MSFT US Equity'

```

Below the code, a table of data is displayed:

7	7	6034	16 2014-08-06 0...
8	8	8350	5 2011-06-18 1...
9	9	2339	13 2011-02-09 1...

* Requires Database Toolbox

** Requires Datafeed Toolbox

MATLAB

Representing Financial Data in MATLAB

- Tables
 - Good for mixed-type tabular data
 - Gives flexibility for indexing and organization

- Datetimes
 - Allow for expression and arithmetic with dates and durations
 - Accounts for time zones, daylight savings and more

- Timetables
 - The best of both above – easily work with, format, and organize time-stamped data

```
data(timerange("01-Jan-2017","17-Mar-2017"),:)
```

```
ans = 161x4 timetable
```

	begin_timestamp	state	event_type	event_narrative	damage_total
1	21-Jan-2017 13:02:00	GEORGIA	Thunderstorm...	"a tree was blown d...	0
2	21-Jan-2017 05:14:00	ALABAMA	Tornado	"the tornado first tou...	750
3	05-Jan-2017 04:00:00	OHIO	Winter Weather	"the county garage ...	0
4	05-Mar-2017 18:00:00	OREGON	Snow	"there were reports ...	0
5	04-Feb-2017 12:15:00	WYOMING	Wind	"the wydot sensor a...	0
6	08-Feb-2017 08:00:00	INDIANA	Winter Weather	"the observers locat...	0
7	18-Jan-2017 18:00:00	CALIFORNIA	Winter Weather	"a spotter in moonri...	0
8	07-Feb-2017 07:00:00	CALIFORNIA	Flood	"major flooding from...	0
9	13-Jan-2017 15:00:00	KANSAS	Ice Storm	"ice accretion was 3...	0
10	02-Jan-2017 00:00:00	NEW YORK	Wind	"a meteor station	50

Big Data Analysis Without Big Changes

One file

Access Data

```
measured = readtable('PumpData.csv');
measured = table2timetable(measured);
```

Preprocess Data

Select data of interest

```
measured = measured(timerange(seconds(1),seconds(2)), 'Speed')
```

Work with missing data

```
measured = fillmissing(measured, 'linear');
```

Calculate statistics

```
m = mean(measured.Speed);
s = std(measured.Speed);
```

One hundred files

Access Data

```
measured = datastore('PumpData*.csv');
measured = tall(measured);
measured = table2timetable(measured);
```

Preprocess Data

Select data of interest

```
measured = measured(timerange(seconds(1),seconds(2)), 'Speed')
```

Work with missing data

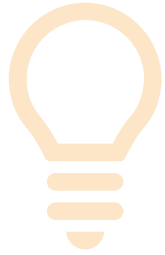
```
measured = fillmissing(measured, 'linear');
```

Calculate statistics

```
m = mean(measured.Speed);
s = std(measured.Speed);
```

```
[m,s] = gather(m,s);
```


Agenda



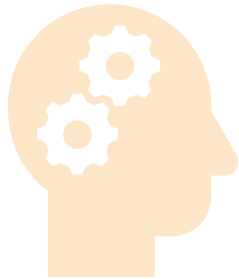
Develop as Fast as you Think



Make Sense of your Data



Speak Finance



Think in Models



Engineer Robust Models



Collaborate and Share

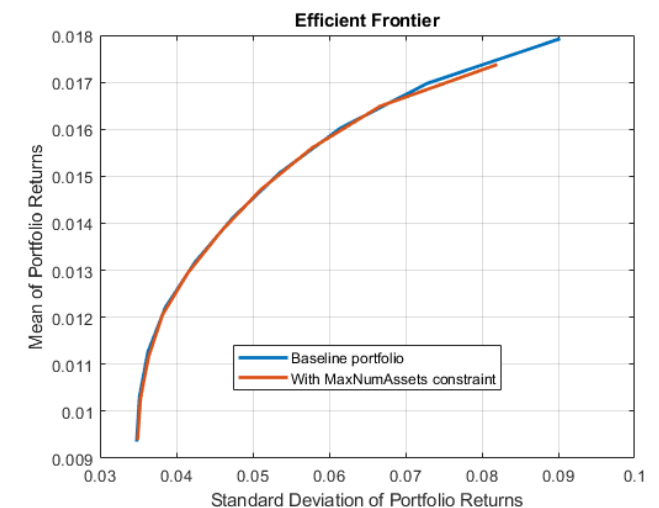
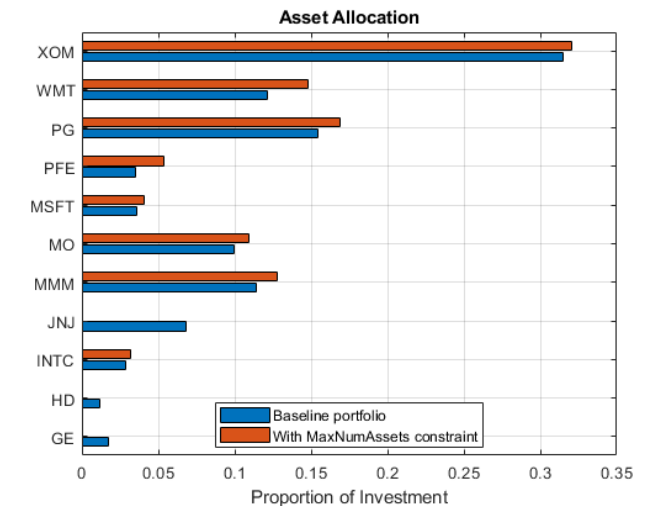
Analyze Financial Data and Develop Financial Models

- **New Backtesting Framework**
 - Define strategy logic for backtests
 - Run backtests with configured properties to compare performance across strategies

- **Portfolio Optimization framework**
 - Allows for constrained mean-variance, MAD, and CVaR optimization
 - Simulate optimized portfolios

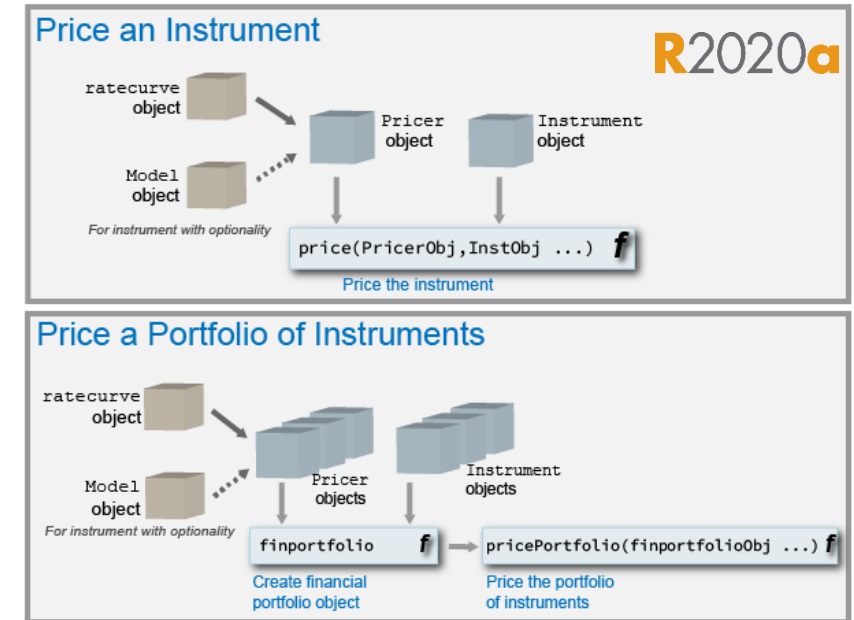
New MATLAB Examples

- Black-Litterman Portfolio Optimization
- Hedging Using CVaR Portfolio Optimization
- Hierarchical Risk Parity
- Portfolio Optimization Using Factor Models
- Machine Learning Examples for Statistical Arbitrage



Design, Price, and Hedge Complex Financial Instruments

- **Object-based framework for pricing financial instruments**
- Monte Carlo Simulation pricing supported for pricing equity, FX, and Commodity instruments
- Construct inflation curves and price inflation bonds and swaps
- Vanna-Volga Method: Vanilla, barrier, double barrier, One-touch and double one-touch options
- Price Variance swaps
- Double Barrier options with Closed Form and Finite Differences

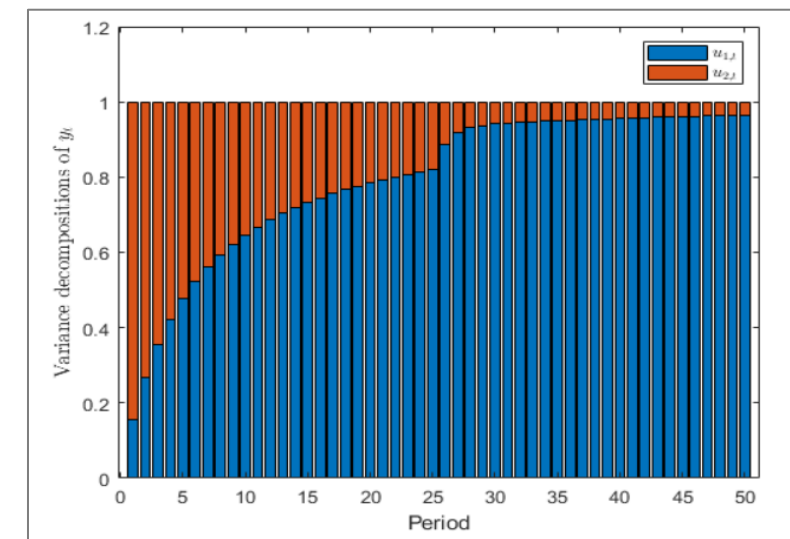
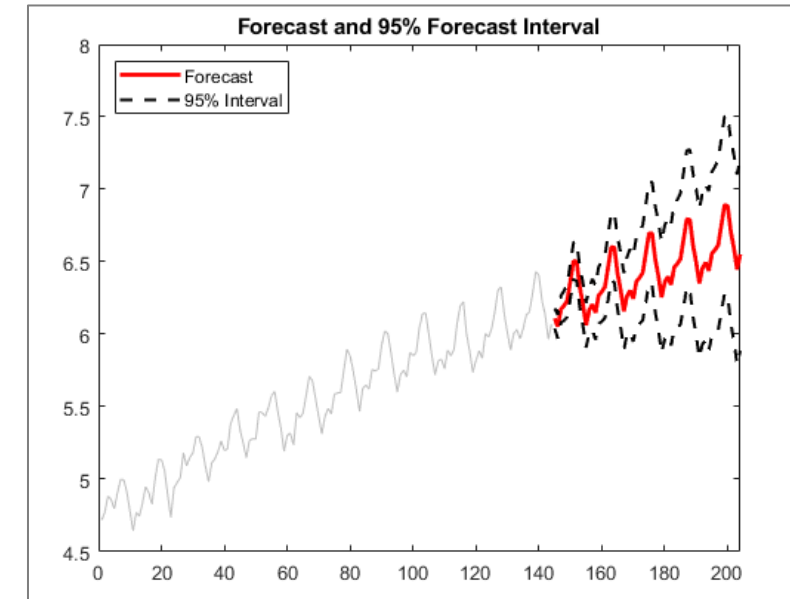


Pricing Interest-rate options with negative

Normal Volatility Model (Bachelier Model)	Shifted Black Model	Shifted SABR Model
European swaption (<code>swaptionbynormal</code>)	European swaption (<code>swaptionbyblk</code>)	Implied Black volatility (<code>blackvolbysabr</code>)
Cap (<code>capbynormal</code>)	Cap (<code>capbyblk</code>)	Option sensitivity (<code>optsensbysabr</code>)
Floor (<code>floorbynormal</code>)	Floor (<code>floorbyblk</code>)	
Caplet volatilities (<code>capvolstrip</code>)		
Floorlet volatilities (<code>floorvolstrip</code>)		

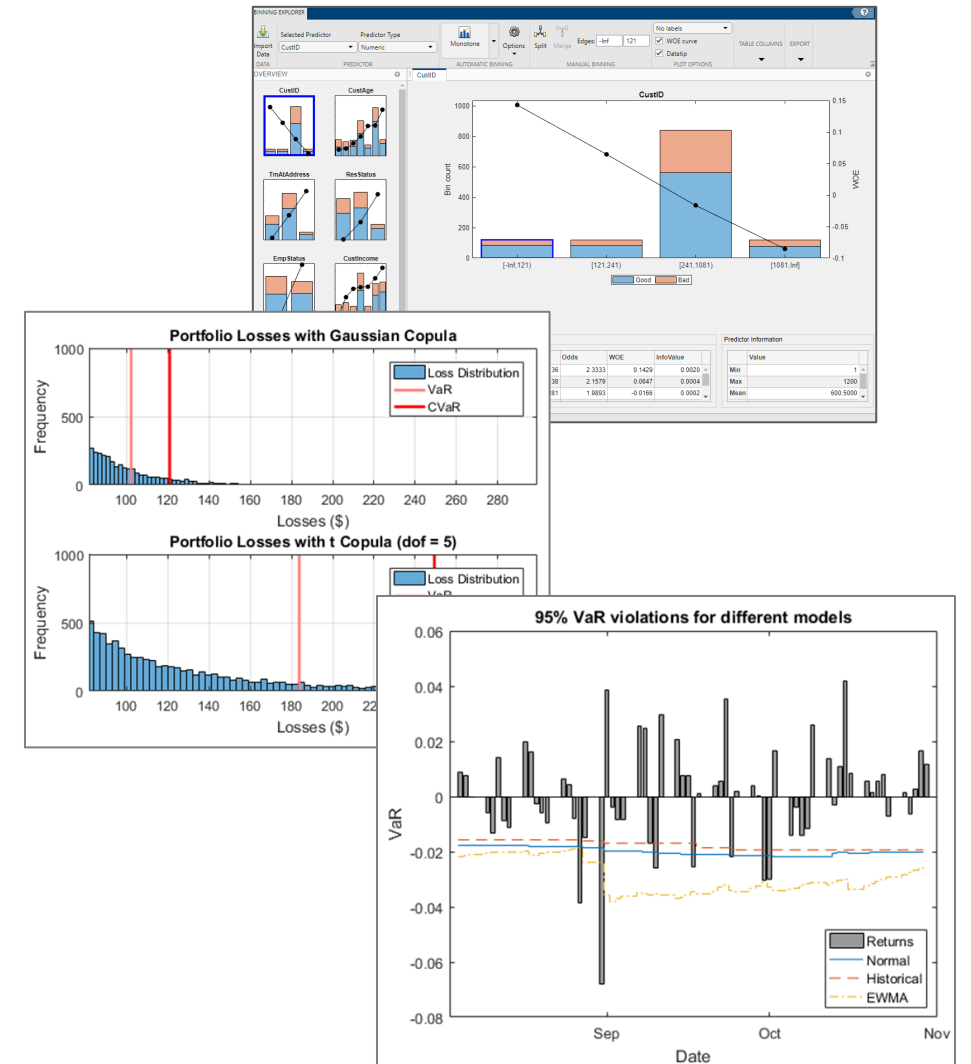
Model and Analyze Econometric Systems

- Econometric Modeler App
 - Popular econometric models, tests, and visualizations available as interactive workflow
 - Document your entire model development workflow with one click
- New Model types
 - Bayesian vector autoregression models
 - Markov-switching autoregression models
- Characterize dynamics of State-Space Models
 - Forecast error variance decomposition (FEVD)
 - Model-implied temporal correlations or covariances



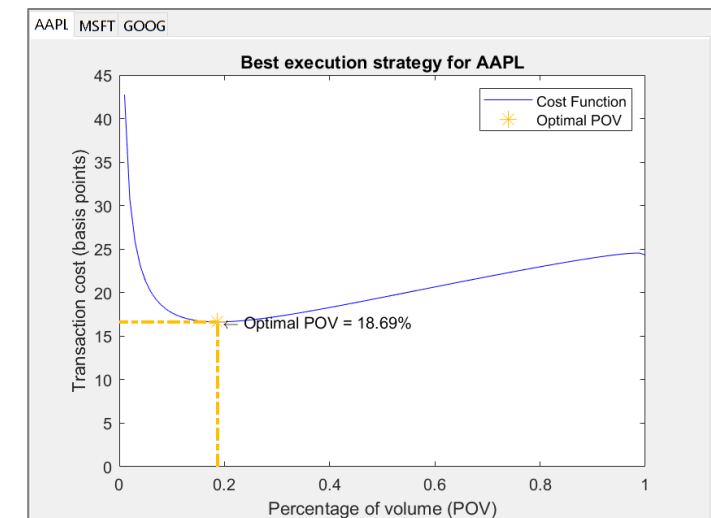
Develop Models for Risk and Run Simulations

- **New Insurance Analysis Framework**
 - Chain ladder, expected claims, Bornhuetter Ferguson, Cape Cod methods for analyzing insurance claim reserves
- Credit scorecard framework to model probability of default
- Assess and simulate corporate and consumer credit risk, as well as market risk
- Backtest VaR and ES models
 - Acerbi and Szekely ES tests
 - Du and Escanciano ES tests
- Lifetime Credit Risk Modeling:
 - Regression and Tobit Loss Given Default (LGD) Models
 - ROC and Accuracy plotting for lifetime PD models



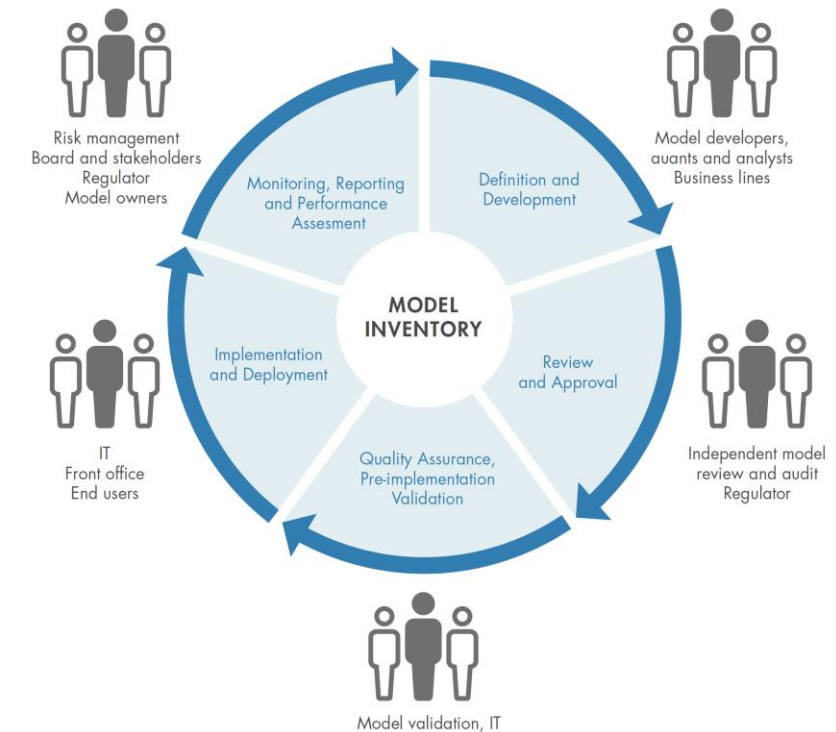
Access Financial Data and Execute Trading Strategies

- **Access prices, analyze transaction costs, and send orders to trading systems**
 - Erstwhile Trading Toolbox functionality merged into Datafeed Toolbox
 - Bloomberg EMSX, CQG, Trading Technologies X_TRADER
- Financial data access (real-time, current, intraday, and historical data)
- Access News and Social Data
 - Refinitiv machine-readable news
 - Twitter data for sentiment analysis



The MathWorks Model Risk Management Solution

- **Unified system** of technologies addressing key business, modeling, workflow, and governance needs
 - Manage model risk with **automation** and **transparency**
- Modeling platform integrated across 1st & 2nd lines of defense, covering **research to production**
 - ✓ Eliminate inefficiencies, reduce cost/time
 - ✓ Enhance communication
 - ✓ Accelerate regulatory approval



Perform end-to-end modeling **faster, better, cheaper**

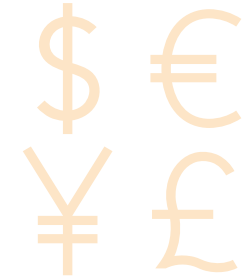
Agenda



Develop as Fast as you Think



Make Sense of your Data



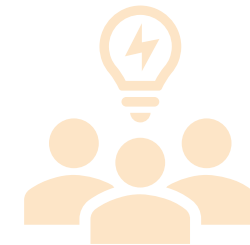
Speak Finance



Think in Models



Engineer Robust Models



Collaborate and Share

MATLAB as a Platform for Advanced Analytics

- MATLAB can support traditional modeling as well as your advanced analytics work

- Consolidated platform for:
 - Optimization
 - Regression
 - Time Series Modeling
 - Machine Learning
 - Deep Learning
 - Natural Language Processing
 - Reinforcement Learning

▼ **Math, Statistics, and Optimization**

- Curve Fitting Toolbox
- Deep Learning Toolbox
- Global Optimization Toolbox
- Optimization Toolbox
- Partial Differential Equation Toolbox
- Statistics and Machine Learning Toolbox
- Symbolic Math Toolbox
- Text Analytics Toolbox

▼ **Computational Finance**

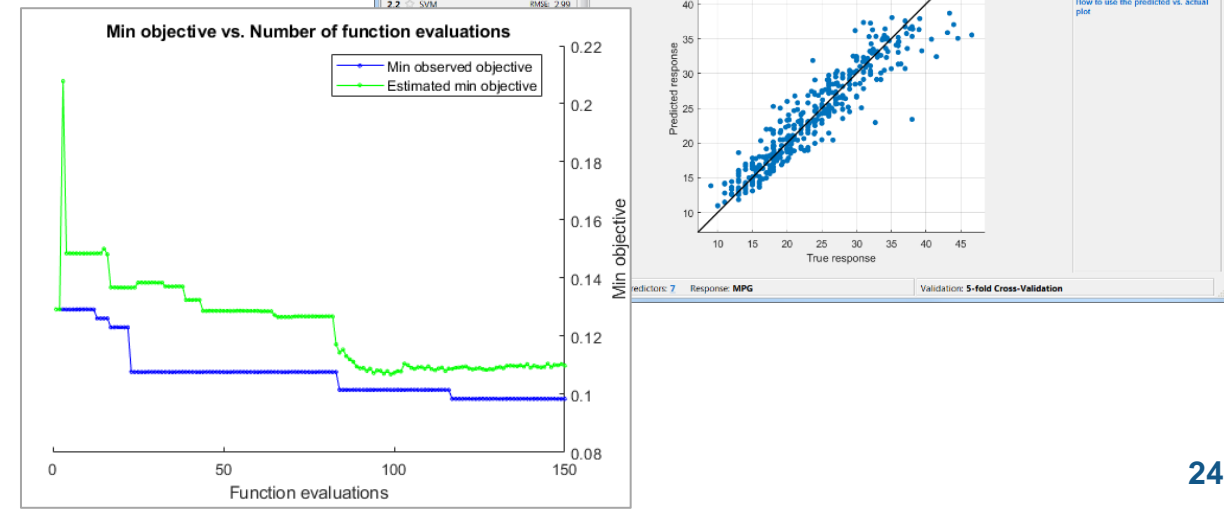
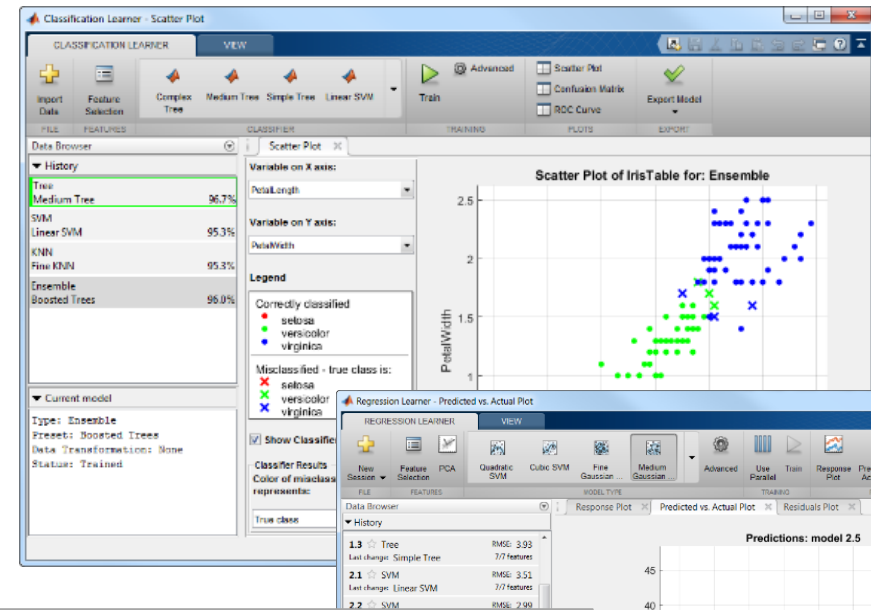
- Database Toolbox
- Datafeed Toolbox
- Econometrics Toolbox
- Financial Instruments Toolbox
- Financial Toolbox
- Risk Management Toolbox
- Spreadsheet Link
- Trading Toolbox

▼ **Data Science and Deep Learning**

- Deep Learning Toolbox
- Predictive Maintenance Toolbox
- Reinforcement Learning Toolbox
- Statistics and Machine Learning Toolbox
- Text Analytics Toolbox

Train, Compare, and Optimize Machine Learning Models

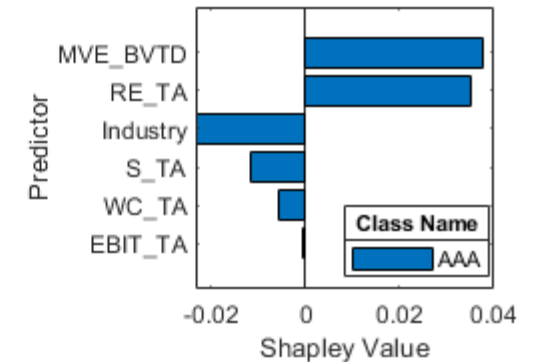
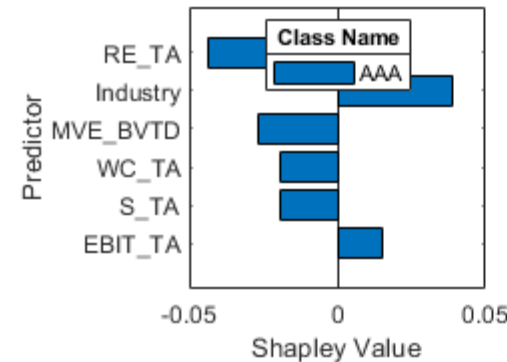
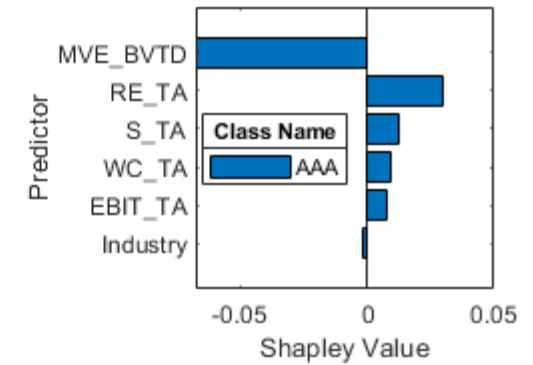
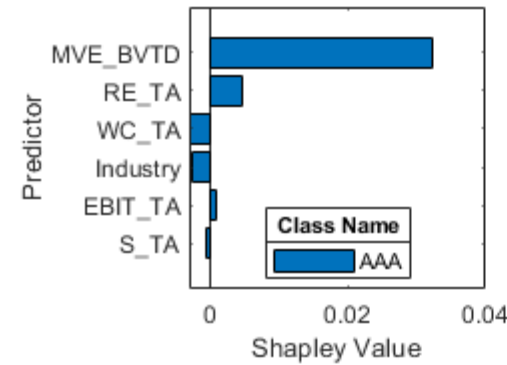
- Use apps to interactively train and compare regression and classification models
 - Use Bayesian optimization in the apps to tune hyperparameters
- Automatically pick the best model via optimization using AutoML functionality
- Support for Incremental/Online learning and semi-supervised learning
- Shallow Neural Networks now supported in Classification/Regression Learner Apps



Interpretability for Machine Learning Models

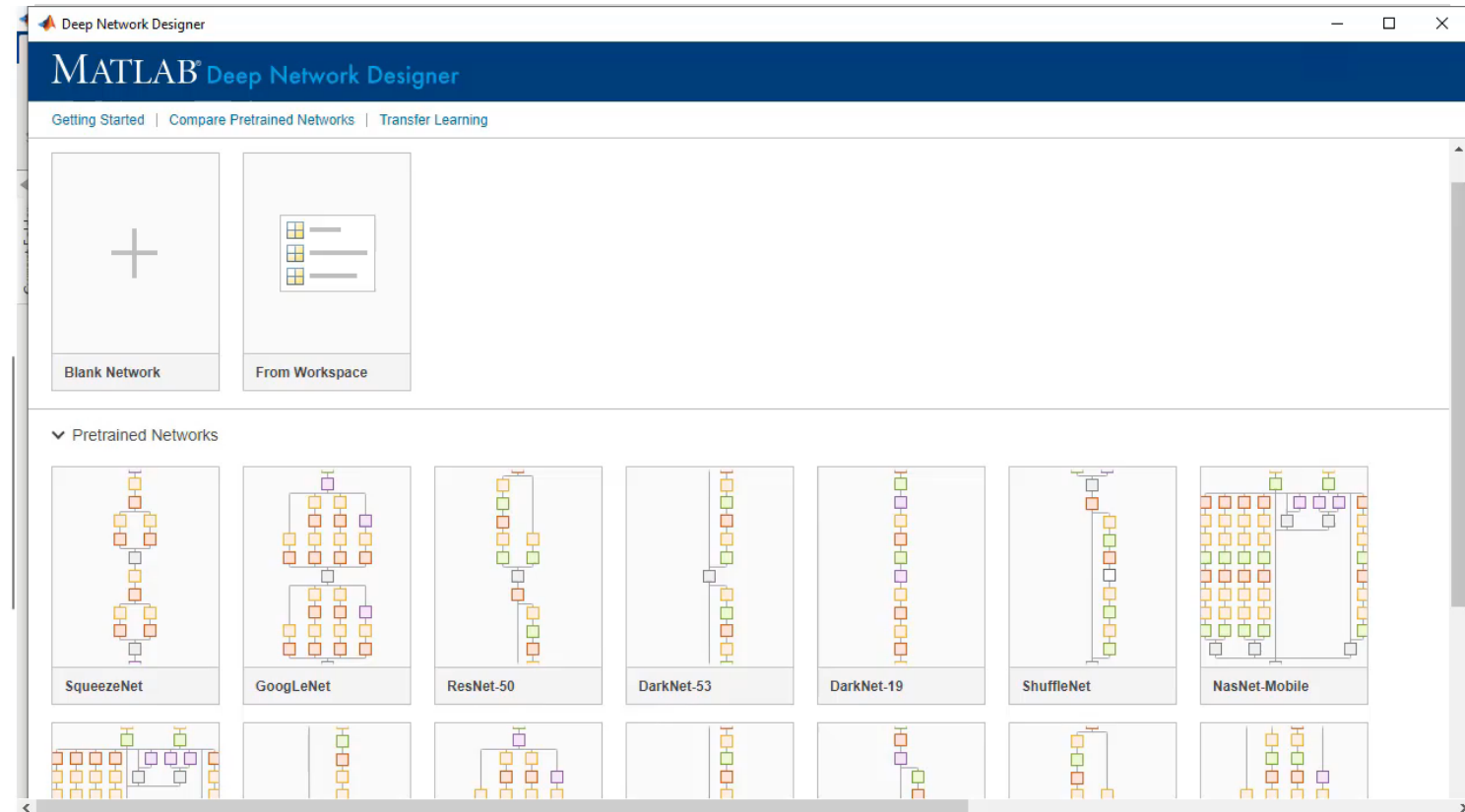
- Support for Common Interpretability Techniques
 - Shapley
 - LIME
 - Partial Dependence Plots (PDP)
 - Individual Conditional Expectation (ICE)

- Support for Interpretable Models
 - Linear Models
 - Tree-based Models
 - Generalized Additive Models (GAMs)



Efficiently Design Deep Networks

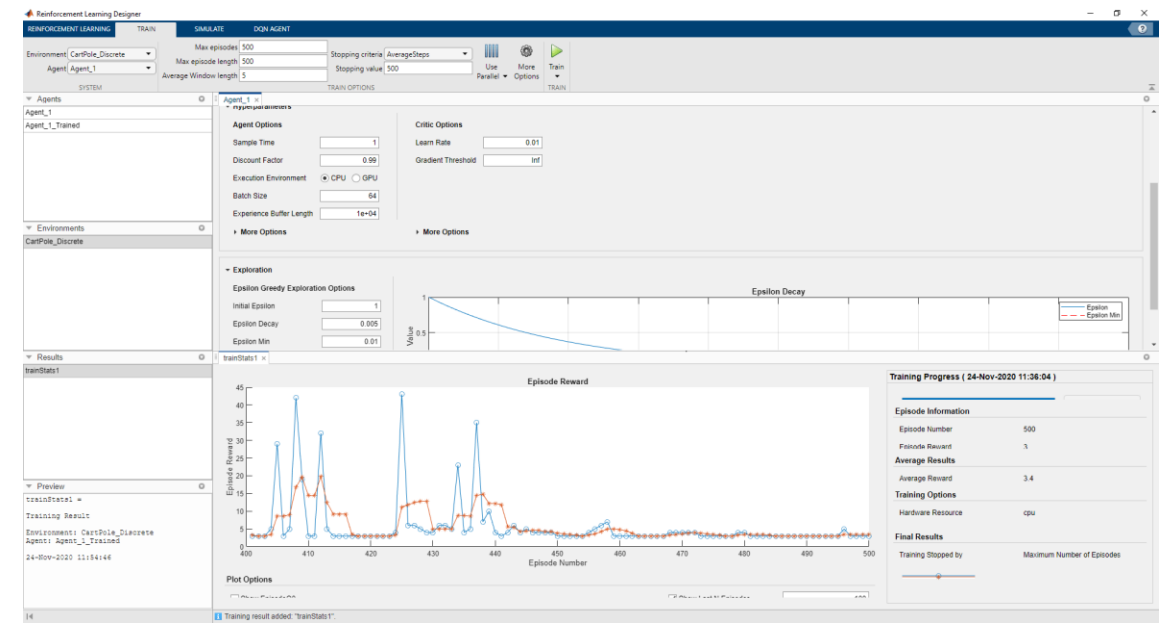
- Deep Network Designer
 - Easily visualize and design a network
 - Start from scratch or with a pretrained model for transfer learning
- Experiment Manager
 - Conduct experiments with varying network architecture/parameters
 - One location to manage all of your network test data



Leverage Advanced Deep Learning Techniques

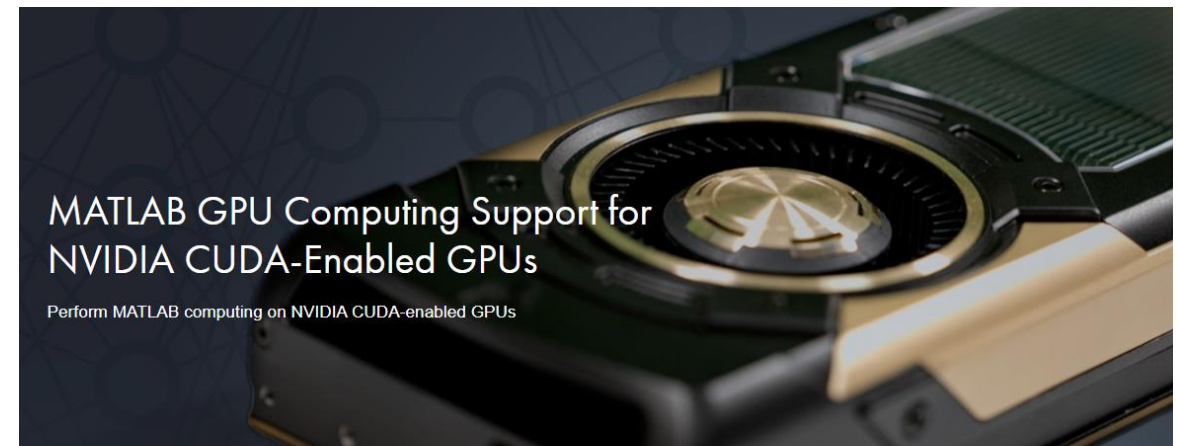
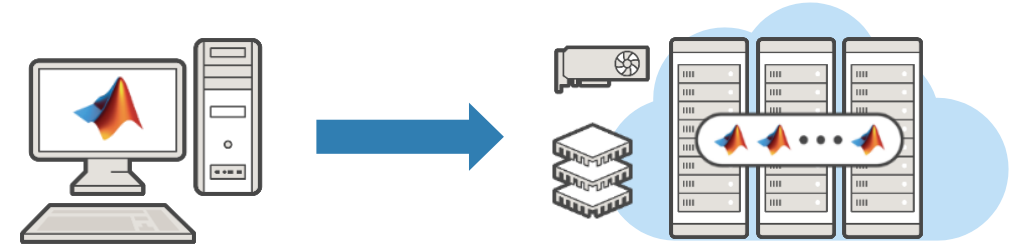
- **Text Analytics Toolbox** offers tools for performing Natural Language Processing
 - Text preprocessing and visualization
 - Machine Learning and Topic Modeling algorithms
 - Document summarization algorithms

- **Reinforcement Learning Toolbox** allows for the training of Reinforcement Learning Agents
 - App to design, train, and simulate agents interactively
 - Includes popular algorithms like Deep Q-Learning, DDPS, Actor-Critic and others



Get the Most out of Your Hardware with Parallel Computing

- New pool type: Thread-based parallel pool
 - In process (explicit workers without extra MATLAB processes)
 - Optimized for reduced memory usage, faster scheduling, and less data transfer
- Scaling to clusters*
 - [License updates](#) provide increased scaling at same price
 - Cloud Support
 - AWS, Azure, NVIDIA GPU Cloud, Containers
- Support for GPU Computing
 - Matrix math computed on GPUs
 - Train Deep Networks on GPUs



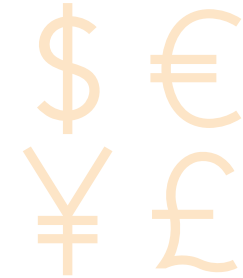
Agenda



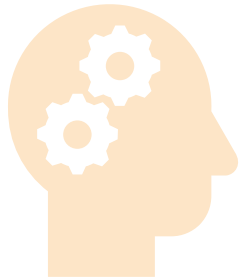
Develop as Fast as you Think



Make Sense of your Data



Speak Finance



Think in Models



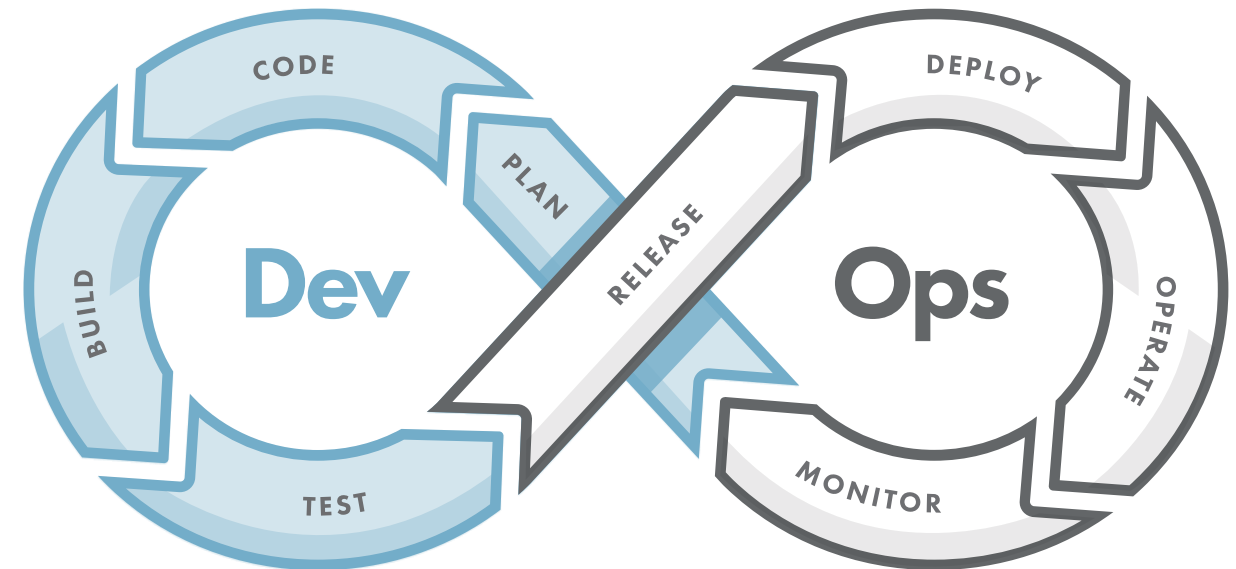
Engineer Robust Models



Collaborate and Share

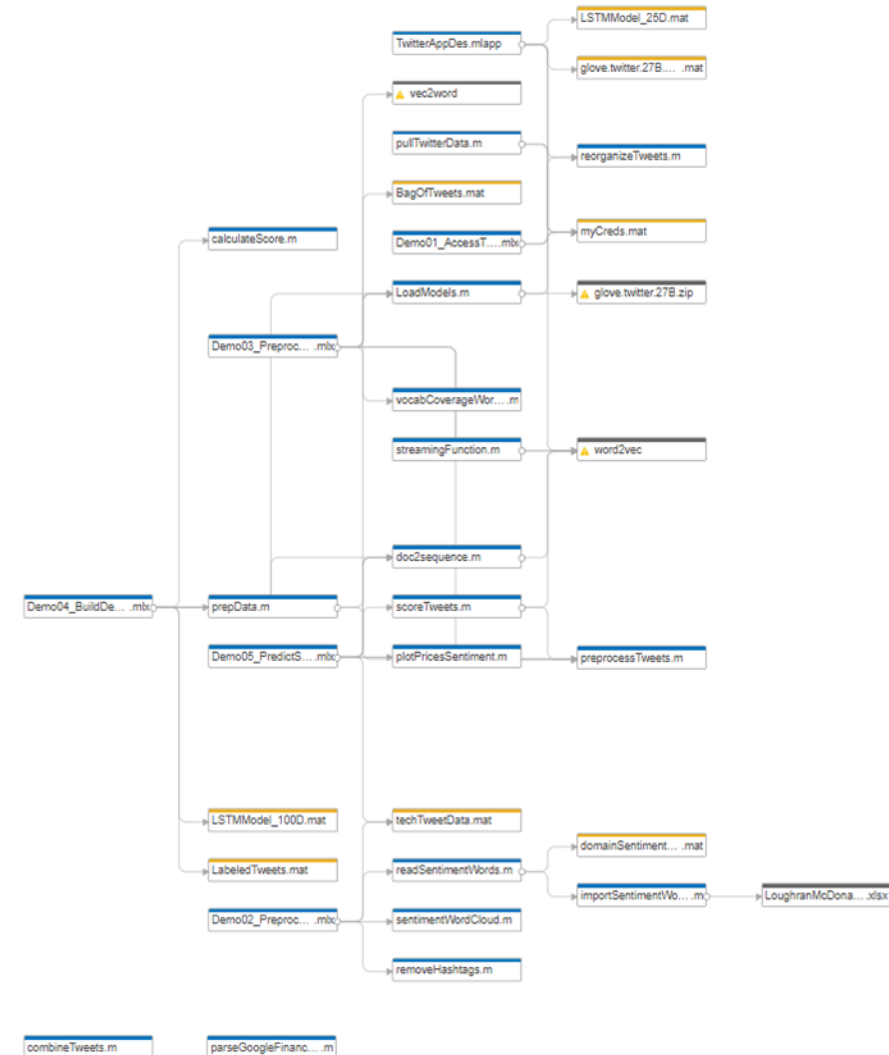
A Fully-Featured Software Engineering Environment

- Familiar Software Engineering Features
 - Class structures
 - Unit testing framework
- Support for Dev Ops Tasks
 - CI/CD Workflows
 - Seamless source control integration



Use MATLAB Projects to Collaborate with Other MATLAB Developers

- Configure your environment
- Analyze dependencies
- Package and share projects
- Full integration with source control
 - Tools to view and merge changes



Rich Support for Object-Oriented Design

CLASS DIAGRAM VIEWER

+ New | Open | Save | Import | Refresh | Clear All | Add | Remove | Superclass | All Superclasses | Go to Source | Auto Arrange | Collapse | Expand | Package Name | Mixins | Select | Pan Mode | Layout | Export

FILE | DIAGRAM | CLASS | VIEW | ZOOM & PAN | ENVIRONMENT | SHARE

Class Browser

- unittest
 - constraints
 - diagnostics
 - fixtures
 - internal
 - measurement
 - meta
 - parameters
 - plugins
 - qualifications
 - selectors

View Content

Classes

- matlab
 - unittest
 - Test
 - TestSuite
 - TestCase
 - FunctionTestCase
 - mixin
 - Copyable
 - handle

Overview

Class Diagram Viewer

```

classDiagram
    class TestSuite {
        +fromClass
        +fromFile
        +fromFolder
        +fromMethod
        +fromName
        +fromPackage
        +getFixtureRequirementMatrix
        +run
        +selectIf
        +sortByFixtures
        +TestSuite
    }
    class Test {
    }
    class Copyable {
        +handle
    }
    class TestCase {
    }
    class FunctionTestCase {
        +handle
    }
    TestSuite <|-- Test
    TestSuite <|-- Copyable
    TestSuite <|-- TestCase
    Copyable <|-- FunctionTestCase
  
```

Inspector

Method

General	
Name	fromClass
Defined In	TestSuite

Access

Static	<input checked="" type="checkbox"/>
Abstract	<input type="checkbox"/>
Access	public
Sealed	<input type="checkbox"/>
Hidden	<input type="checkbox"/>

Legend

- Protected
- Read-only
- Constant/Static

CLASSES

- Handle Class
- Value Class
- Abstract Class
- Hidden Class
- Enumeration
- Super Class
- Out Of Sync
- Indirect Inheritance
- Inheritance

Framework for Creating, Running, and Reporting on Tests

- MATLAB Unit Testing Framework
 - Includes script-, function-, and class-based testing
 - Works with continuous integration servers
- Performance Testing Framework
 - Time MATLAB code automatically
 - Track performance over time
- App Testing Framework
 - Author automated test for App Designer apps

Run the tests and examine results.

```
result = runtests('test_Predictions.mlx')
```

```
Running test_Predictions
```

```
..
```

```
Done test_Predictions
```

```
_____
result =
```

```
1x2 TestResult array with properties:
```

```
Name
Passed
Failed
Incomplete
Duration
Details
```

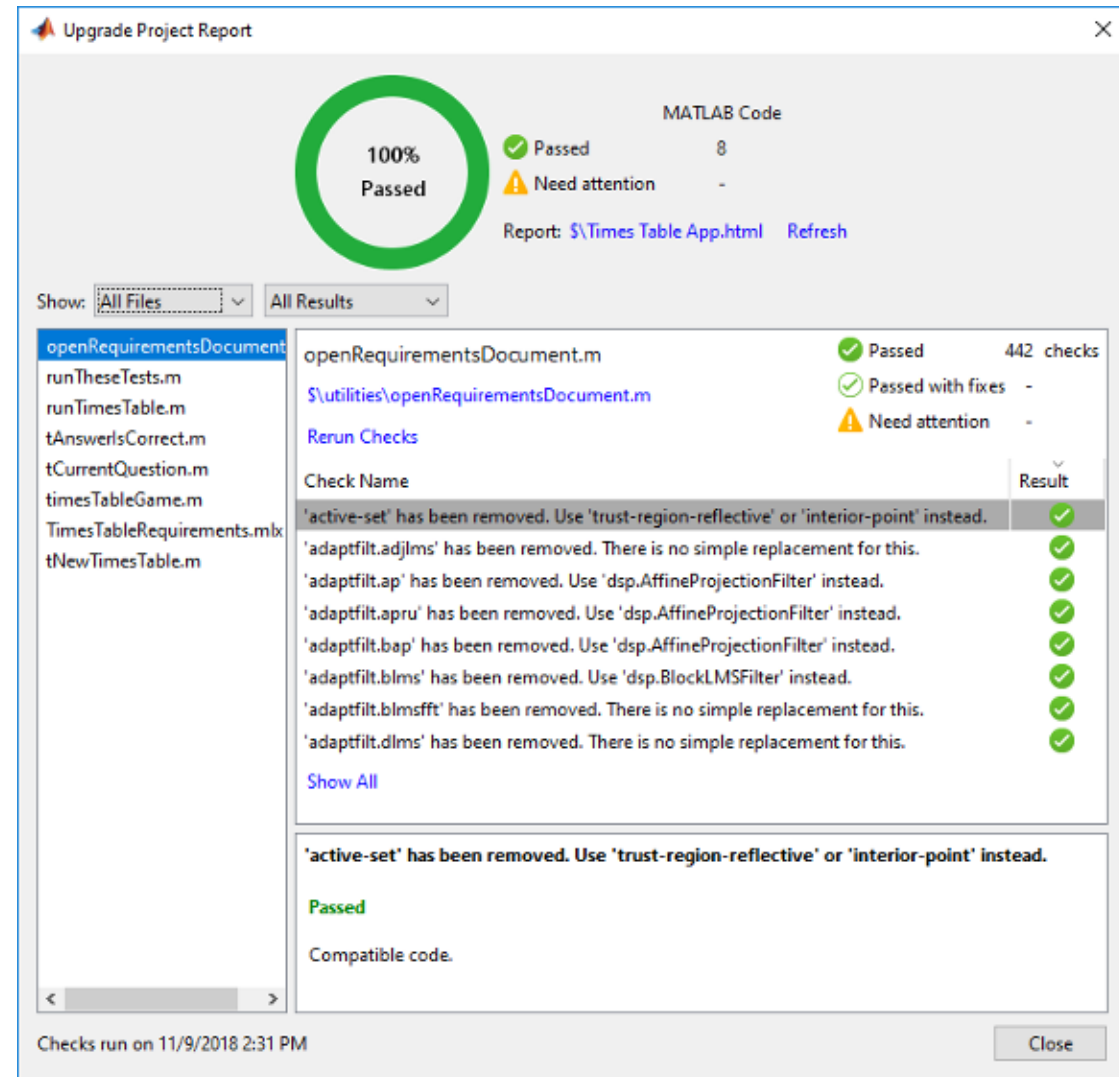
```
Totals:
```

```
2 Passed, 0 Failed, 0 Incomplete.
```

```
0.41712 seconds testing time.
```

Upgrading to the latest MATLAB – Upgrade Projects

- Tool to help upgrade code to latest and greatest MATLAB
- Hundreds of checks for incompatibilities, errors, and warnings
- Applies fixes automatically



Upgrade Project Report

100% Passed

MATLAB Code

Passed 8

Need attention -

Report: [S:\Times Table App.html](#) Refresh

Show: All Files All Results

File	Status	Checks
openRequirementsDocument.m	Passed	442 checks
S:\utilities\openRequirementsDocument.m	Passed with fixes	-
	Need attention	-

Rerun Checks

Check Name	Result
'active-set' has been removed. Use 'trust-region-reflective' or 'interior-point' instead.	✓
'adaptfilt.adjlms' has been removed. There is no simple replacement for this.	✓
'adaptfilt.ap' has been removed. Use 'dsp.AffineProjectionFilter' instead.	✓
'adaptfilt.apru' has been removed. Use 'dsp.AffineProjectionFilter' instead.	✓
'adaptfilt.bap' has been removed. Use 'dsp.AffineProjectionFilter' instead.	✓
'adaptfilt.blms' has been removed. Use 'dsp.BlockLMSFilter' instead.	✓
'adaptfilt.blmsfft' has been removed. There is no simple replacement for this.	✓
'adaptfilt.dlms' has been removed. There is no simple replacement for this.	✓

Show All

'active-set' has been removed. Use 'trust-region-reflective' or 'interior-point' instead.

Passed

Compatible code.

Checks run on 11/9/2018 2:31 PM

Close

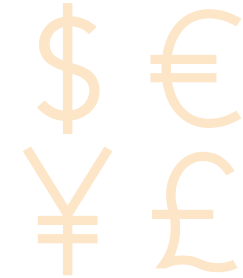
Agenda



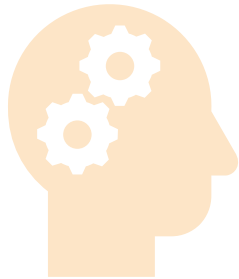
Develop as Fast as you Think



Make Sense of your Data



Speak Finance



Think in Models



Engineer Robust Models



Collaborate and Share

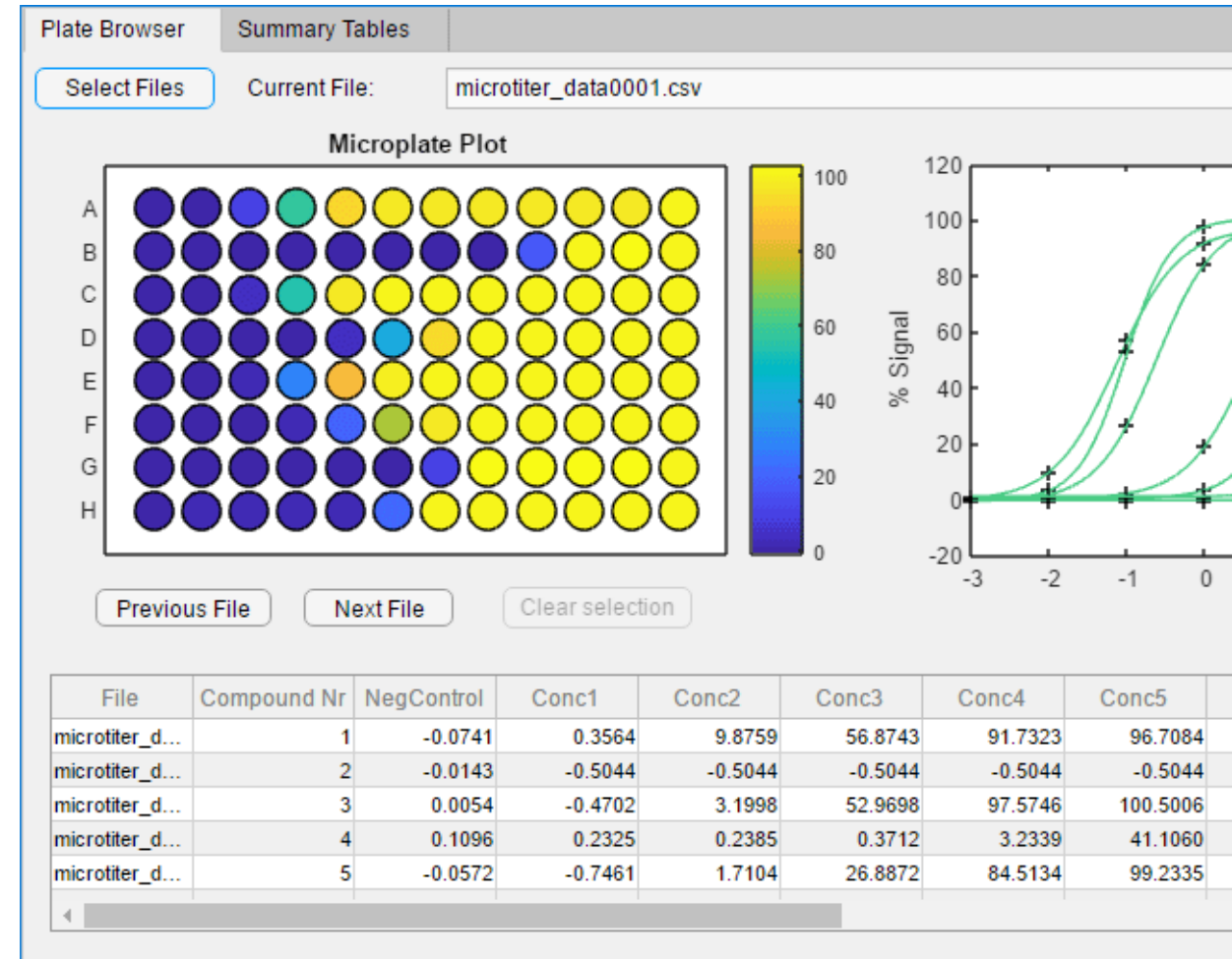
Host MATLAB for Browser Access

- **Access MATLAB in-browser, no need for downloads or installs**
 - Hosted by your IT organization on-prem or in the cloud
- Users can more easily collaborate on MATLAB, co-locate data with compute, and access from anywhere
- IT organizations have a centralized, scalable way to manage MATLAB



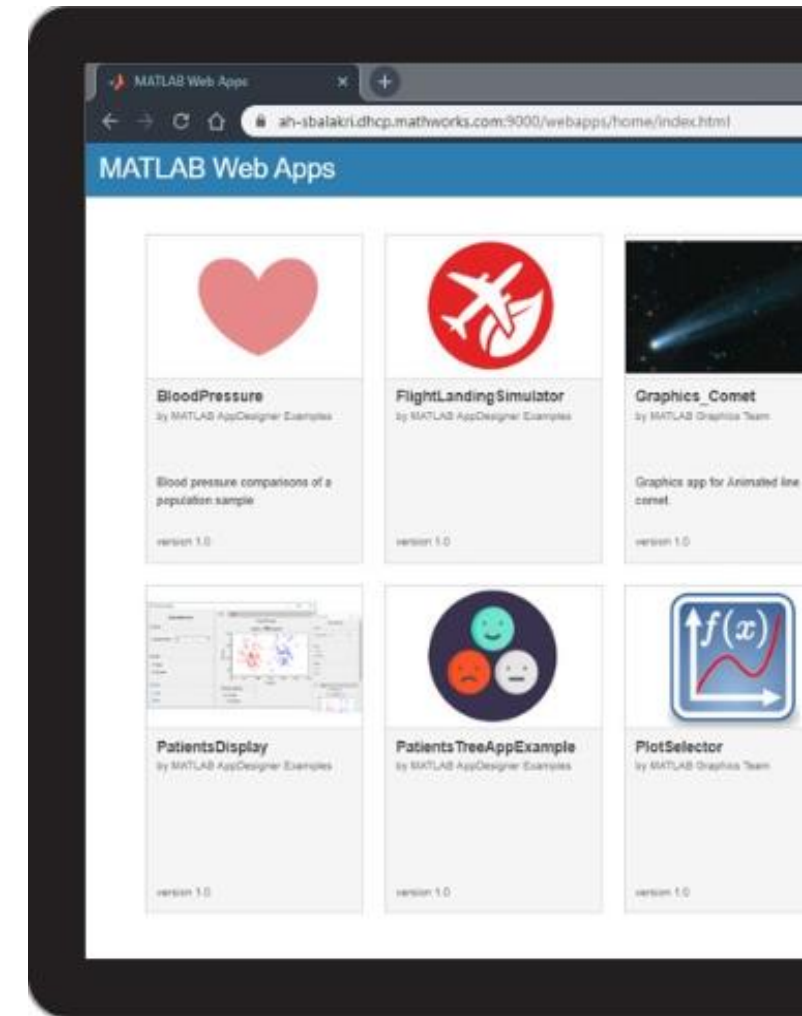
Create and Share Apps as a Vehicle for Your Models

- App Designer can be used to design GUIs for your models
 - No need to learn another language to make your models interactive
- These apps can be shared to end users **who don't have a MATLAB License**
 - Can be shared as an executable
 - Can be shared as an in-browser web application



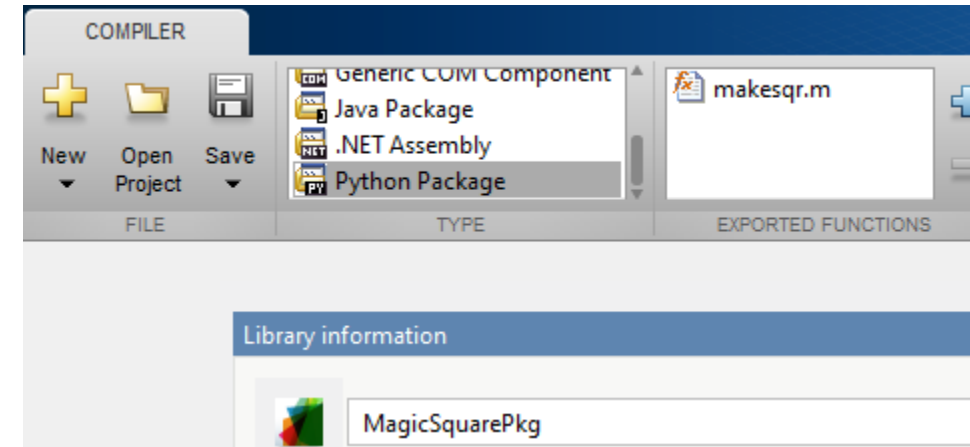
Share MATLAB Apps as Browser-Based Web Apps

- MATLAB Web App Server allows for access of App Designer apps in-browser
- Provides:
 - Authentication using OpenID Connect and LDAP
 - Support for apps developed in different releases of MATLAB or Simulink
 - Role-based access and editing privileges
 - **Policy-based access control: decide group permissions at an app-by-app level**



Deployment to Other Languages and Systems

- Share your models with colleagues who use other tools
 - Create libraries for Python, Java, C/C++, and .NET*
 - Create Excel Add-ins
- Create MapReduce applications that run against Hadoop
- Create MATLAB applications that run against an Apache Spark enabled cluster



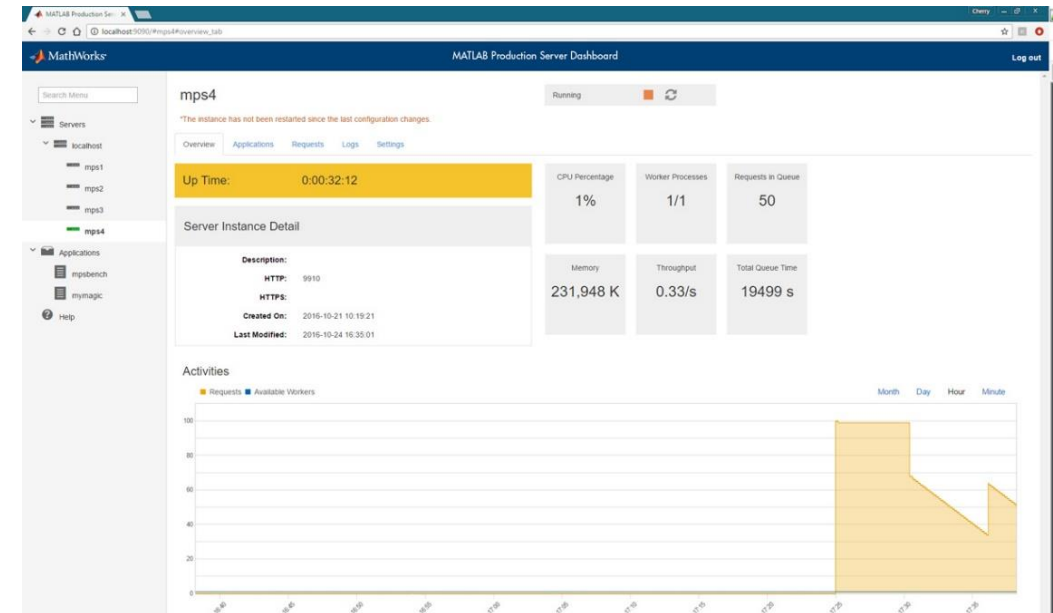
```
python makesqrSample1.py
```

```
[[8.0,1.0,6.0],  
 [3.0,5.0,7.0],  
 [4.0,9.0,2.0]]
```

Integrate MATLAB Analytics into Enterprise Applications

Deploy MATLAB algorithms without recoding or creating custom infrastructure

- Now supports OAuth2 providers (R2021a)
- Develop clients for MATLAB Production Server in any programming language that supports HTTP using RESTful API and JSON
- Configure and manage multiple server instances using a web-based interface
- Discover the list of APIs provided by installed applications through a RESTful interface



Using MATLAB with Other Languages

Calling Libraries Written in Another Language From MATLAB



- Java
 - Python
 - C
 - C++
 - Fortran
 - COM components and ActiveX[®] controls
 - RESTful, HTTP, and WSDL web services
- Execute Python functions out of process **R2019b**
- Call C++ libraries directly from MATLAB **R2019a**

Calling MATLAB from Another Language



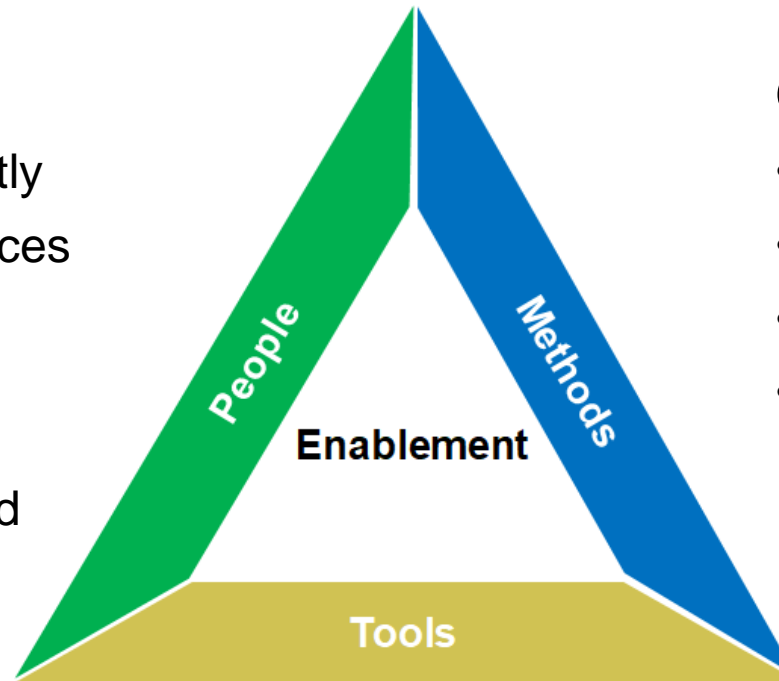
- Java
- Python
- C/C++
- Fortran
- COM Automation server

MathWorks Services – Enabling Customers

The MathWorks “Whole Product” approach ensures that end users, teams, and organizations are fully enabled.

Training

- Ramp up efficiently
- Adopt best practices
- Establish consistency
- Self-paced, instructor-led, and custom courses



Consulting

- Expand proficiency
- Accelerate projects
- Improve processes
- Consultants with deep expertise in specific areas

Products

- Technical computing
- Vertical solutions

Questions? Contact Us!

Steve Notley

Application Engineer, Financial Services
snotley@mathworks.com

Greg McGean

Account Manager, Financial Services
gmcgean@mathworks.com