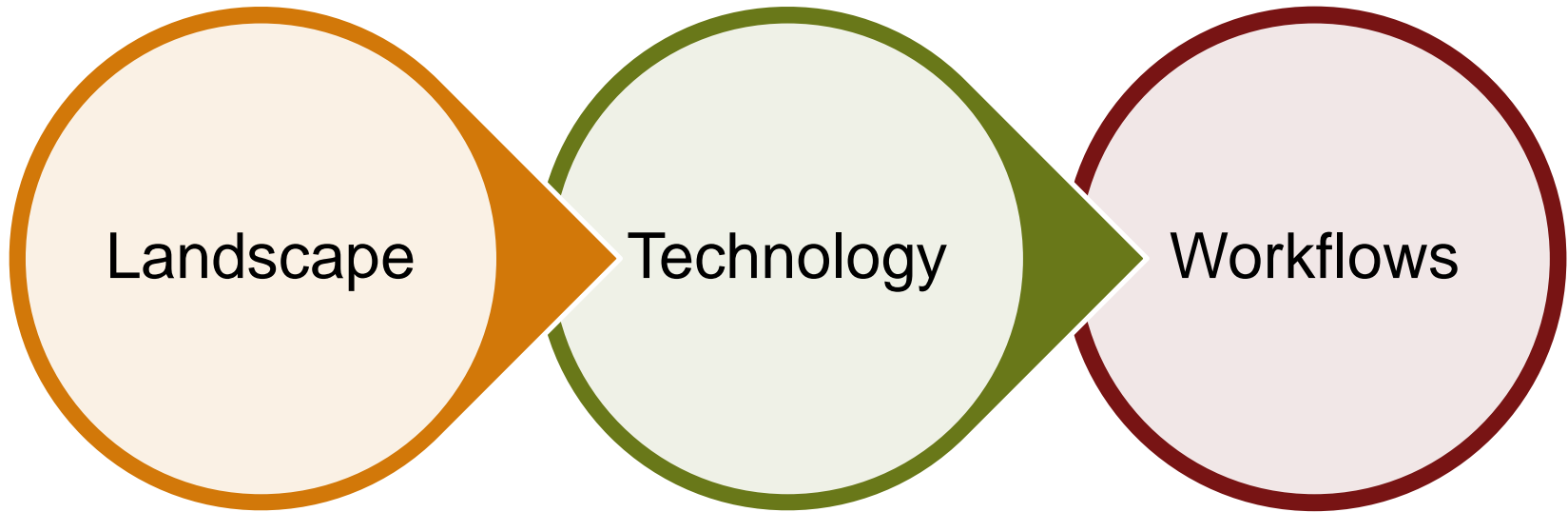


# Streamlining Financial Modelling: From Development to Approval to Production with MATLAB

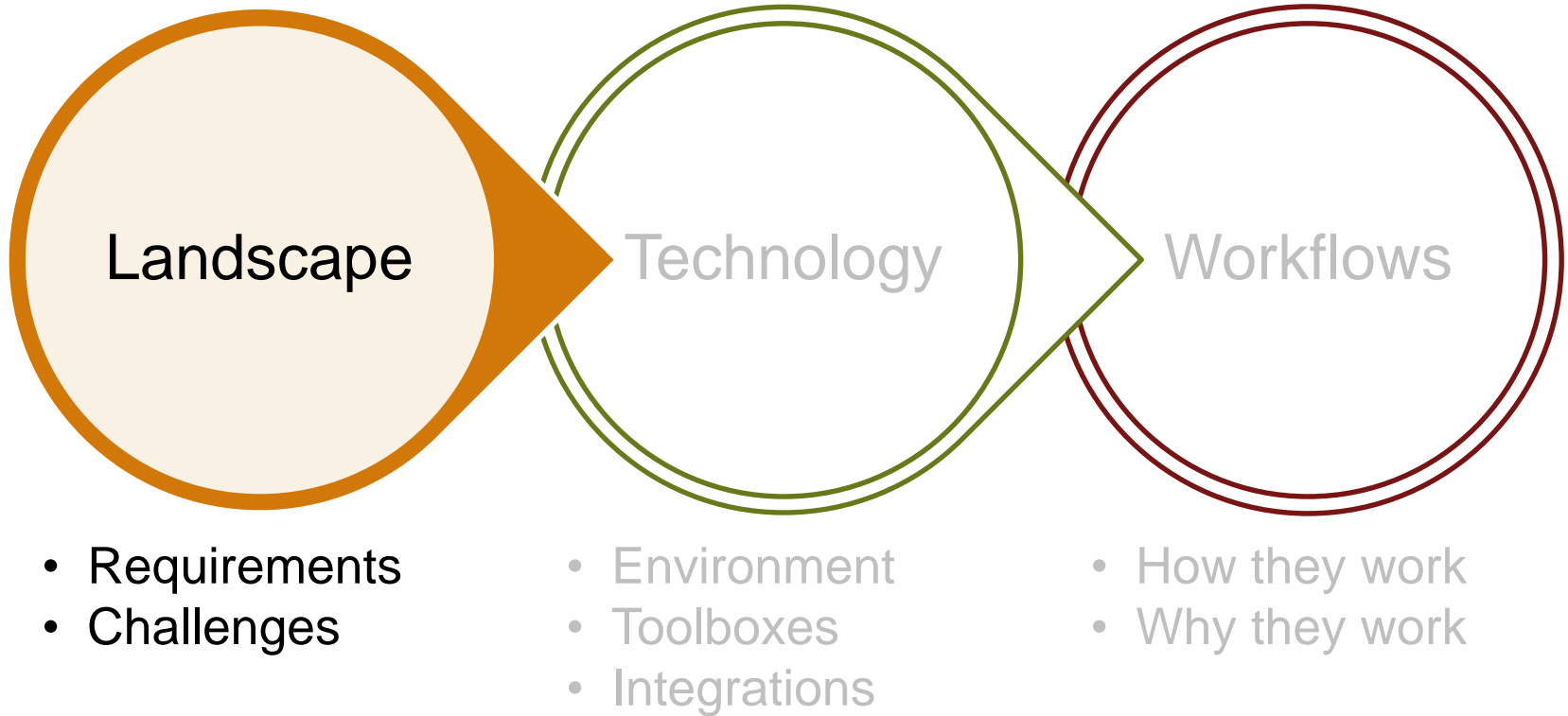
**David Sampson**  
**MathWorks**



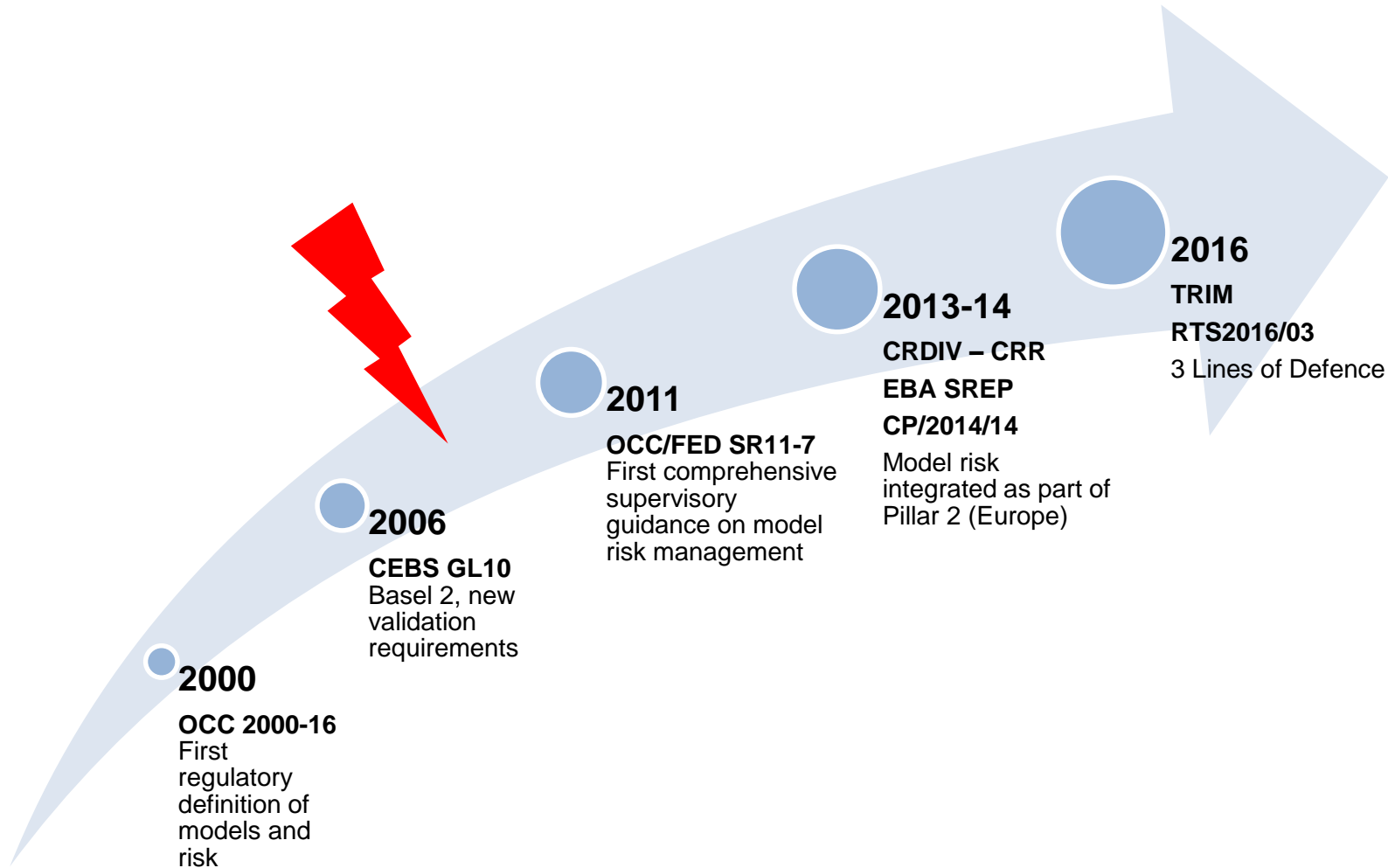
- Requirements
- Challenges

- Environment
- Toolboxes
- Integrations

- How they work
- Why they work



# Model risk management regulations





# Regulator requirements

quantitative limitations qualitative  
roles competence  
judgement  
 challenge uncertainty models impact  
 external inventory verify credibility  
 independence capabilities  
understand process thirdparty data sensitivities  
 proportionate  
 responsibilities  
transparency



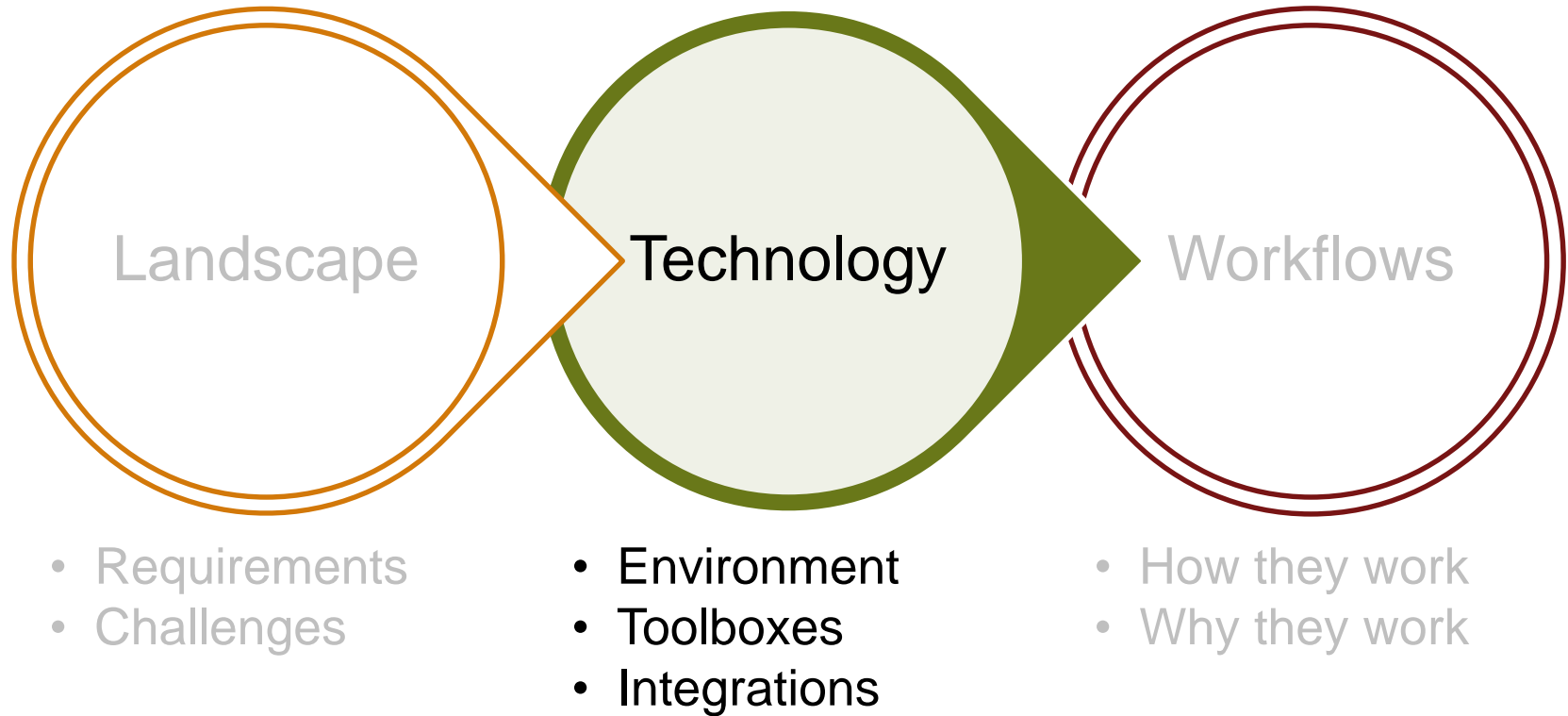


# Institution challenges



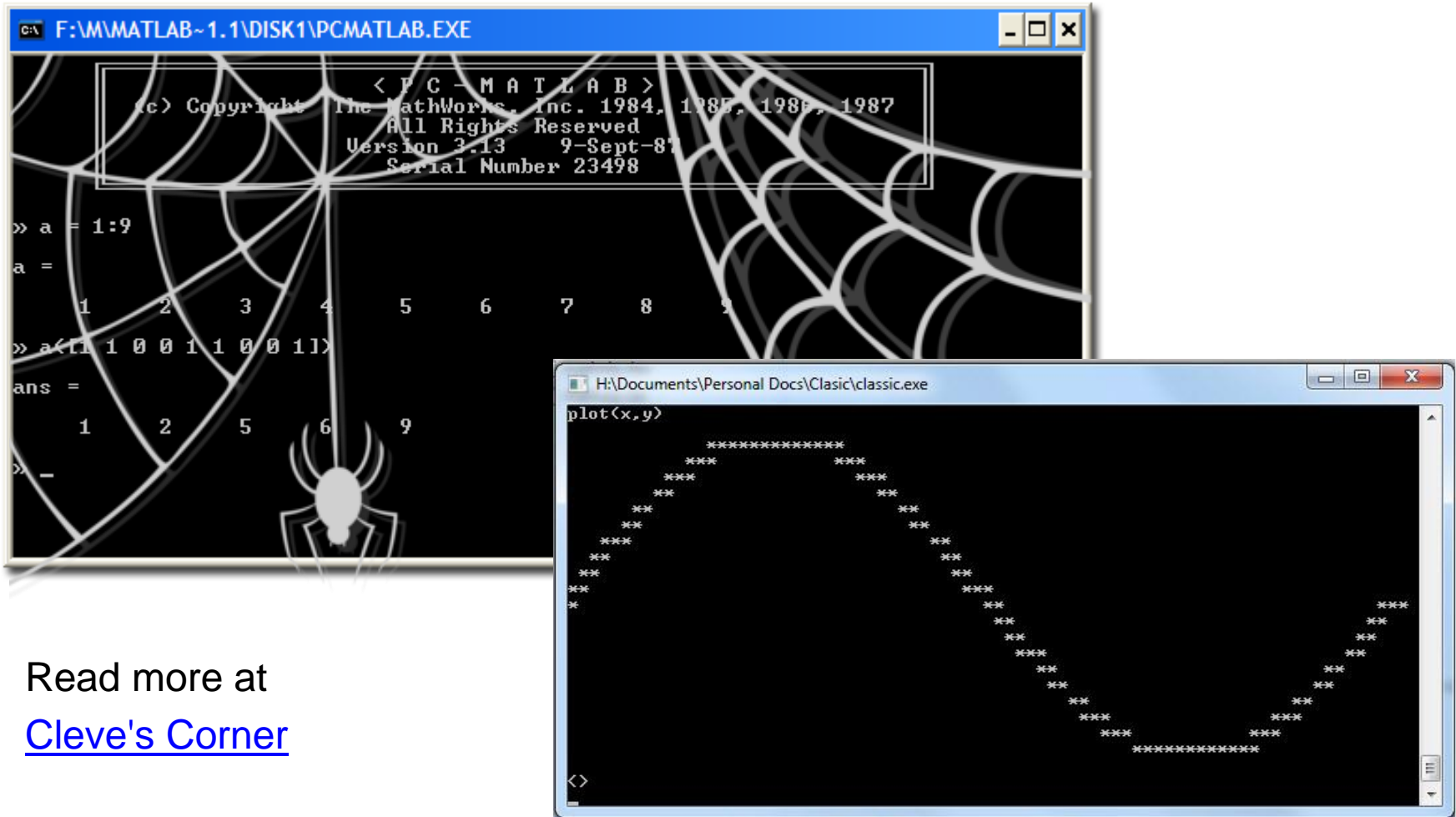
consistency  
 availability  
 validity  
 readability  
 rework  
 data  
 model  
 robustness  
 integration  
 external  
 reuse  
 effort  
 range  
 review  
 insight  
 templates  
 throughput  
 standards  
 interoperability  
 usability  
 sensitivity  
 recoding  
 security  
 internal  
 segmentation  
 traceability







## In the beginning...



Read more at  
[Cleve's Corner](#)

# Desktop

MATLAB R2018a

HOME | PLOTS | APPS | SHORTCUTS | E | Search Documentation | David

New Script | New Live Script | New | Open | Compare | Find Files | Import Data | Save Workspace | New Variable | Open Variable | Clear Workspace | Favorites | Analyze Code | Run and Time | Clear Commands | Simulink | Layout | Preferences | Set Path | Add-Ons | RESOURCES

FILE | VARIABLE | CODE | SIMULINK | ENVIRONMENT

C:\Development\layouts2

Current Folder

Name ^	SVN
.svn	•
bash	●
bugs	○
demos	●
docsrc	●
ideas	○
planning	●
releases	●
submissions	○
tbx	●
tests	●
addsandbox.m	●
glt.png	●
GUI Layout Toolbox.prj	●
prepareBaTSubmission.m	●
README.txt	●
release.m	●
rmsandbox.m	●

Details ^

Command Window

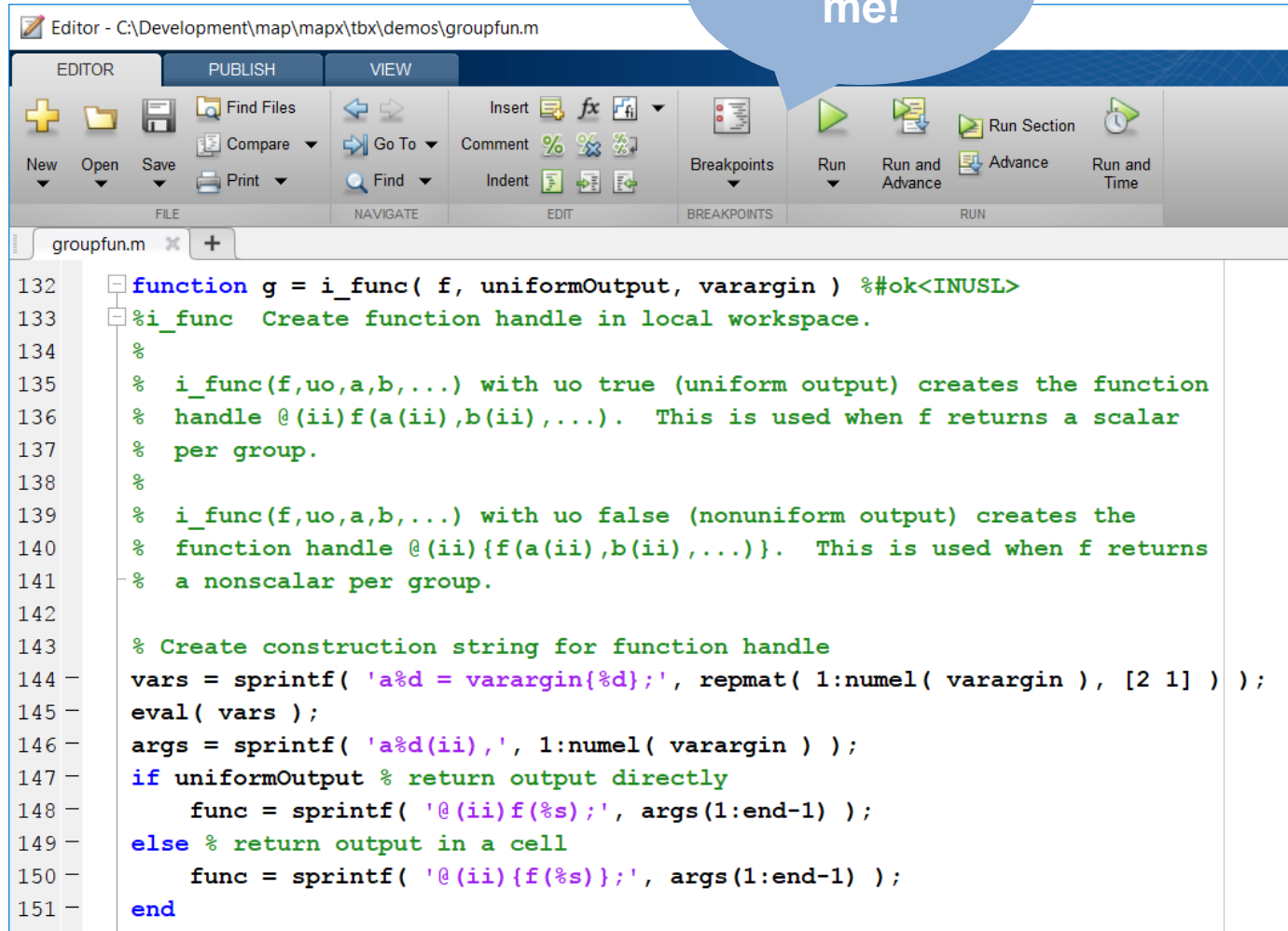
```
>> |
```

Workspace

Name ^	Value

# Editor

Debug  
me!



Editor - C:\Development\map\map\tbx\demos\groupfun.m

EDITOR PUBLISH VIEW

New Open Save Find Files Compare Print

Go To Find Comment Indent

Insert fx Breakpoints

Run Run and Advance Run Section Advance Run and Time

groupfun.m

```

132 function g = i_func( f, uniformOutput, varargin ) %#ok<INUSL>
133 %i_func Create function handle in local workspace.
134 %
135 % i_func(f,uo,a,b,...) with uo true (uniform output) creates the function
136 % handle @(ii)f(a(ii),b(ii),...). This is used when f returns a scalar
137 % per group.
138 %
139 % i_func(f,uo,a,b,...) with uo false (nonuniform output) creates the
140 % function handle @(ii){f(a(ii),b(ii),...)}. This is used when f returns
141 % a nonscalar per group.
142
143 % Create construction string for function handle
144 vars = sprintf( 'a%d = varargin{%d};', repmat( 1:numel( varargin ), [2 1] ) );
145 eval( vars );
146 args = sprintf( 'a%d(ii),', 1:numel( varargin ) );
147 if uniformOutput % return output directly
148     func = sprintf( '@(ii)f(%s);', args(1:end-1) );
149 else % return output in a cell
150     func = sprintf( '@(ii){f(%s)};', args(1:end-1) );
151 end

```

# Graphics

## Documentation

CONTENTS

### Types of MATLAB Plots

There are various functions that you can use to plot data in MATLAB®. This table classifies and illustrates the common graphics functions.

Line Plots	Pie Charts, Bar Plots, and Histograms	Discrete Data Plots	Polar Plots	Contour Plots	Vector Field
<b>plot</b> 	<b>area</b> 	<b>stairs</b> 	<b>polarplot</b> 	<b>contour</b> 	<b>quiver</b> 
<b>plot3</b> 	<b>pie</b> 	<b>stem</b> 	<b>polarhistogram</b> 	<b>contourf</b> 	<b>quiver3</b> 
<b>semilogx</b> 					
<b>semilogy</b> 					

Generate code!

# Toolboxes

**Documentation**
Close

☰ CONTENTS

[< Documentation Home](#)

< **Statistics and Machine Learning Toolbox** i

- Getting Started with Statistics and Machine Learning Toolbox
- Descriptive Statistics and Visualization
- Probability Distributions
- Hypothesis Tests
- Cluster Analysis
- ANOVA
- Regression
- Classification
- Dimensionality Reduction and Feature Extraction
- Industrial Statistics
- Analysis of Big Data with Tall Arrays
- Speed Up Statistical Computations
- Code Generation

- Examples
- Functions
- Classes
- Apps
- Release Notes

**Documentation**
Close

☰ CONTENTS

[< Documentation Home](#)

< **Econometrics Toolbox** i

- Getting Started with Econometrics Toolbox
- Data Preprocessing
- Model Selection
- Time Series Regression Models
- Conditional Mean Models
- Conditional Variance Models
- Multivariate Models
- Markov Models

- Examples
- Functions
- Classes
- Apps
- Release Notes
- PDF Documentation

Search Help

## Econometrics Toolbox

Model and analyze financial and economic systems using statistical methods

Econometrics Toolbox™ provides functions for modeling economic data. You can select and estimate economic models for simulation and forecasting. For time series modeling and analysis, the toolbox includes univariate Bayesian linear regression, univariate ARIMAX/GARCH composite models with several GARCH variants, multivariate VARX models, and cointegration analysis. It also provides methods for modeling economic systems using state-space models and for estimating using the Kalman filter. You can use a variety of diagnostics for model selection, including hypothesis tests, unit root, stationarity, and structural change.

### Getting Started

Learn the basics of Econometrics Toolbox

### Data Preprocessing

Format, plot, and transform time series data

### Model Selection

Specification testing and model assessment

### Time Series Regression Models

Bayesian linear regression models and regression models with nonspherical disturbances

### Conditional Mean Models

Autoregressive (AR), moving average (MA), ARMA, ARIMA, ARIMAX, and seasonal models

### Conditional Variance Models

GARCH, exponential GARCH (EGARCH), and GJR models

### Multivariate Models

Cointegration analysis and vector autoregression (VAR) and vector error correction (VEC) models



# Documentation Browser

Documentation
Search Help

CONTENTS

My Products

- MATLAB
- Simulink
- Aerospace Blockset
- Aerospace Toolbox
- Audio System Toolbox
- Automated Driving System Toolbox
- Bioinformatics Toolbox
- Communications System Toolbox
- Computer Vision System Toolbox
- Control System Toolbox
- Curve Fitting Toolbox
- Data Acquisition Toolbox
- Database Toolbox
- Datafeed Toolbox
- DO Qualification Kit (for DO-178)
- DSP System Toolbox
- Econometrics Toolbox
- Embedded Coder
- Financial Instruments Toolbox
- Financial Toolbox
- Fixed-Point Designer
- Fuzzy Logic Toolbox
- Global Optimization Toolbox
- GPU Coder
- HDL Coder
- IEC Certification Kit (for ISO 26262 and IEC 61508)
- Image Acquisition Toolbox
- Image Processing Toolbox
- Instrument Control Toolbox
- Manning Toolbox

Examples

CONTENTS

## MATLAB Examples

Language Fundamentals

**Basic Matrix Operations**

Basic techniques and functions for working with matrices in the MATLAB® language.

[Open Live Script](#)

**Matrix Manipulation**

Do some basic matrix manipulations in MATLAB®.

[Open Live Script](#)

**Manipulating Multidimensional Arrays**

Work with arrays having more than two dimensions. Multidimensional arrays can be numeric, character, cell, or structure arrays.

[Open Live Script](#)

**Create Structure Array**

Create a structure array. Each field is a data type that can be used in data using data fields. Each field is a data type that can be used in data using data fields.

Mathematics

**Integer Arithmetic**

Perform arithmetic on integer data representing signals and images.

**Single Precision Math**

Perform arithmetic and linear algebra with single precision data. It

**Creating and Editing Delaunay Triangulations**

Create, edit, and query Delaunay triangulations using the

**Predicting the US Population**

Using polynomials of even modest degree to predict the future by

**FFT for Spectral Analysis**

The use of the FFT function for spectral analysis. A common use of

Close

Explore Examples

Explore Add-Ons

Search Help

Documentation

CONTENTS

**fmincon**

Find minimum of constrained nonlinear multivariable function

Nonlinear programming solver.

Finds the minimum of a problem specified by

$$\min_x f(x) \text{ such that } \begin{cases} c(x) \leq 0 \\ ceq(x) = 0 \\ A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub, \end{cases}$$

*b* and *beq* are vectors, *A* and *Aeq* are matrices. *c(x)* and *ceq(x)* are functions that return vectors, and *f(x)* is a function that returns a scalar. *x*, *lb*, and *ub* can be passed as vectors or matrices; see [Matrix Arguments](#).

**Syntax**

```
x = fmincon(fun,x0,A,b)
x = fmincon(fun,x0,A,b,Aeq,beq)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
x = fmincon(problem)
[x,fval] = fmincon(__)
[x,fval,exitflag,output] = fmincon(__)
[x,fval,exitflag,output,lambda,grad,hessian] = fmincon(__)
```

**Description**

`x = fmincon(fun,x0,A,b)` starts at `x0` and attempts to find a minimizer `x` of the function described in `fun` subject to the linear inequalities `A*x ≤ b` and the linear equalities `Aeq*x = beq`.

`x = fmincon(fun,x0,A,b,Aeq,beq)` defines a set of lower and upper bounds on the design variables in `x`.

**Note**

[Passing Extra Parameters](#) explains how to pass extra parameters to the objective function and nonlinear constraints.

Explore Examples

Explore Add-Ons

Search Help

Click and drag to move MATLAB Documentation or its



# Variable Editor

Command Window

```
>> TMW

TMW =

1000x5 timetable

    Time      Open      High
    _____  _____  _____
    04-Sep-2012    100    102.19
    05-Sep-2012   100.15   101.05
    06-Sep-2012   100.4    102.38
    07-Sep-2012   101.74   102.37
    10-Sep-2012   99.72    101.55
    11-Sep-2012   98.48    98.66
    12-Sep-2012   96.9     99.18
```

Variables - TMW

PLOTS VARIABLE VIEW

Rows: 1 Columns: []

Transpose, Insert, Delete, Sort

TMW x

1000x5 timetable

	Time	1 Open	2 High	3 Low	4 Close	5 Volume
1	04-Sep-2012	100	102.1900	98.5700	100.8900	7479170
2	05-Sep-2012	100.1500	101.0500	98.4500	100.4300	6505120
3	06-Sep-2012	100.4000	102.3800	100.3400	99.9700	6046860
4	07-Sep-2012	101.7400	102.3700	98.9700	99.5100	7247900
5	10-Sep-2012	99.7200	101.5500	98.0500	98.3600	5732900
6	11-Sep-2012	98.4800	98.6600	96.6300	96.9000	5756600
7	12-Sep-2012	96.9000	99.1800	96.5400	96.7800	6210400

# Live Editor

Live Editor - C:\Users\dsmppson\AppData\Local\MathWorks\MATLAB\Examples\matlab\LiveEditorInteractiveNarrative\LiveEditorInteractiveNarrative.mlx

**LIVE EDITOR** INSERT VIEW

Find Files, Compare, Go To, Print, Text, Code, Section Break, Run Section, Run and Advance, Run to End

```

omega = 15*(solarTime.Hour + solarTime.Minute/60 - 12); % Hour angle
alpha = asind(sind(delta)*sind(phi) + ... % Elevation
    cosd(delta)*cosd(phi)*cosd(omega));
disp([' Solar Declination = ' num2str(delta) ' Solar Elevation = ' num2str(alpha)])

Knowing the sun's declination and the local latitude allows us to calculate the time of sunrise and sunset. Sunrise and sunset are
    sunrise = 12 - cos-1(-tan φ tan δ) / 15° - TC / 60    sunset = 12 + cos-1(-tan φ tan δ) / 15°

midnight = dateshift(localTime, 'start', 'day');
sr = 12 - acosd(-tand(phi)*tand(delta))/15 - solarCorr/60;
sunrise = timeofday(midnight + hours(sr));
ss = 12 + acosd(-tand(phi)*tand(delta))/15 - solarCorr/60;
sunset = timeofday(midnight + hours(ss));
disp([' Sunrise = ' datestr(sunrise, 'HH:MM:SS') ' Sunset = ' datestr(sunset, 'HH:MM:SS') ])
    
```

**Air Mass and Solar Radiation**

Include images to illustrate important points in your story. To include an image from a file, copy and paste an image from another document, or use the **Image** button.

As light from the sun passes through the earth's atmosphere, some of the solar radiation will be absorbed. The air mass is a fundamental concept in solar radiation. In the diagram below, it is a measure of the length of the path of light through the atmosphere (Y) relative to the shortest possible path (X).

The larger the air mass, the less radiation reaches the ground. The air mass can be calculated from the equation

$$AM = \frac{1}{\cos(90 - \alpha) + 0.5057(6.0799 + \alpha)^{-1.6364}}$$

Then the solar radiation (in Kw/m<sup>2</sup>) reaching the ground can be calculated from the empirical equation

$$sRad = 1.353 * 0.74AM^{0.678}$$

Live Editor - C:\Users\dsmppson\AppData\Local\MathWorks\MATLAB\Examples\matlab\UseInteractiveControlsExample\UseInteractiveControls.mlx

**LIVE EDITOR** INSERT VIEW

Find Files, Compare, Go To, Print, Text, Code, Section Break, Run Section, Run and Advance, Run to End

Use Interactive Controls in a Live Script

This example shows how you can add interactive controls to a live script. Adding interactive controls to a live script is useful when you want to share the live script with others. Use interactive controls to set and change the values of variables in your live script using familiar user interface components such as numeric sliders and drop-down lists.

**Add Interactive Controls to Your Script**

To add an interactive control, select a value in the live script, click the **Interactive Controls** button, and select either **Numeric Slider** or **Drop Down**. To configure the interactive control, double-click it. When you double-click it, the live editor highlights the section that contains the control.

**Visualize Airport and Carrier Delays**

Use the interactive controls in the live script to set the year, top airports, and top carriers to visualize. The live script reads in the airport delay data using the readDelays function. The calculateDelays function calculates the carrier delays for the airports and the top carriers. The plotDelays function creates bar charts to display the delays for the top airports and carriers.

delayType = 'DepDelay';  
 year = 1999;  
 topAirports = 7;  
 topCarriers = 10;

```

allDelays = readDelays;
[airportDelays, carrierDelays] = calculateDelays(allDelays, year, topAirports, topCarriers);
plotDelays(year, delayType, airportDelays, carrierDelays)
    
```

**Change me!**

**Average DepDelay by Airport in 1999**

Airport	Average Delay (minutes)
ATL	11.5
DEN	7.0
DFW	11.0
LAX	11.0
ORD	12.5
PHX	10.0
STL	8.5

**Average DepDelay by Carrier in 1999**

Carrier	Average Delay (minutes)
AA	11.5
AS	8.5
CO	7.5
DL	6.5
HP	11.5
NW	8.5
TW	9.0
UA	11.0
US	9.0
WN	11.0

# Apps

Regression Learner - Predicted vs. Actual Plot

REGRESSION LEARNER VIEW

FILE FEATURES MODEL TYPE TRAINING PLOTS EXPORT

Data Browser

History

- 1 ☆ Tree RMSE: 3.34 Last change: Medium Tree 777 features
- 2 ☆ SVM RMSE: 3.56 Last change: Cubic SVM 777 features
- 3 ☆ SVM RMSE: 2.65 Last change: Medium Gaussian SVM 777 features
- 4 ☆ Gaussian Process Regression RMSE: 2.66 Last change: Squared Exponential GPR 777 features
- 5 ☆ Ensemble RMSE: 3.09 Last change: Boosted Trees 777 features
- 6 ☆ Ensemble RMSE: 2.96 Last change: Bagged Trees 777 features
- 7 ☆ Linear Regression RMSE: 3.36 Last change: Linear 777 features

Current Model

Model 4: Trained

Results

- RMSE 2.66
- R-Squared 0.88
- MSE 7.10
- MAE 1.88
- Prediction speed ~20000 obs/sec
- Training time 2.9479 sec

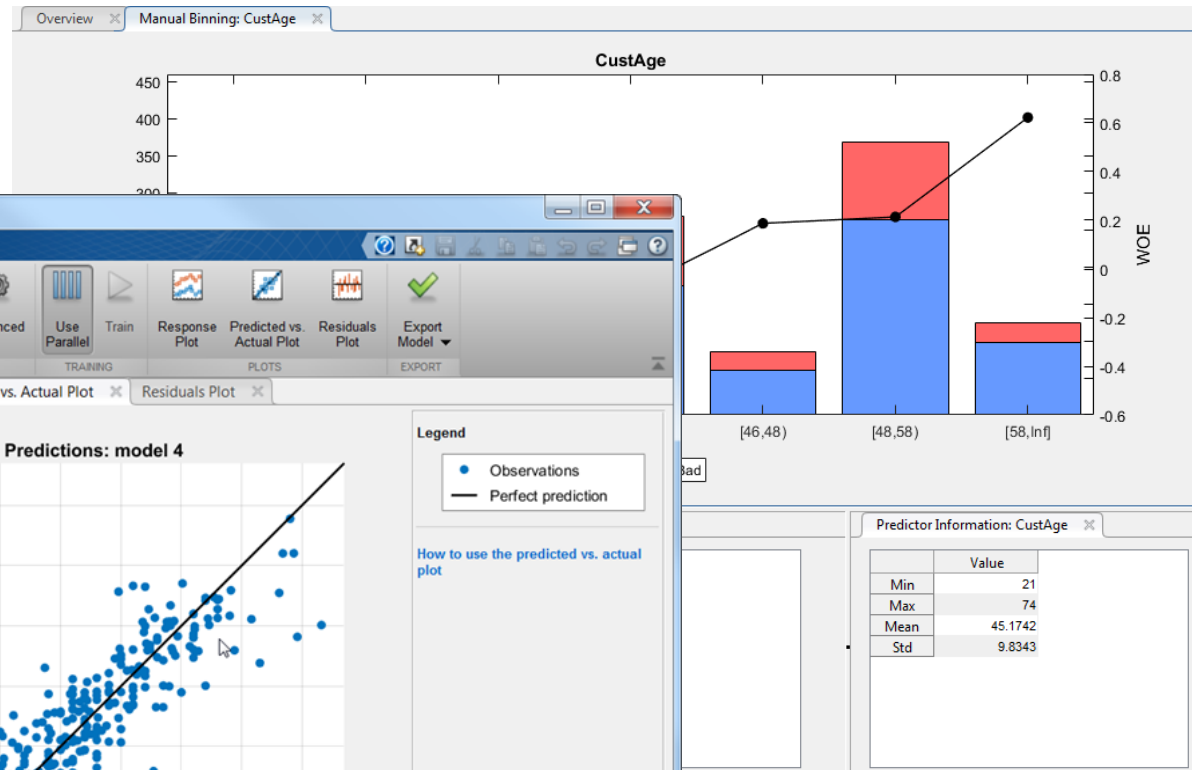
Predictions: model 4

Legend

- Observations
- Perfect prediction

How to use the predicted vs. actual plot

Dataset: cartable Observations: 406 Size: 30 kB Predictors: 7 Response: MPG Validation: 5-fold Cross-Validation



Generate code!

# Toolbox Packaging

Package a Toolbox - GUI Layout Toolbox.prj

PACKAGER

FILE TOOLBOX FOLDER PACKAGE

New Open Project Save Package

tbx

Toolbox Information

GUI Layout Toolbox 2.3.2

David Sampson & Ben Tordoff

david.sampson@mathworks.co.uk

MathWorks

Set as default contact

Layout manager for MATLAB graphical user interfaces

This toolbox provides tools to create sophisticated MATLAB graphical user interfaces that resize g used in combination to produce virtually any user interface layout.

\* Arrange MATLAB user interface components horizontally, vertically or in grids

Toolbox Files and Folders

layout layoutdoc

Name	Date modified	Type	Size
GUI Layout Toolbox 2.1.1	07/11/2016 10:16	MATLAB Toolbox	830 KB
GUI Layout Toolbox 2.1.2	07/11/2016 10:16	MATLAB Toolbox	833 KB
GUI Layout Toolbox 2.1	07/11/2016 10:16	MATLAB Toolbox	743 KB
GUI Layout Toolbox 2.2.1	06/04/2016 07:06	MATLAB Toolbox	714 KB
GUI Layout Toolbox 2.2.2	07/11/2016 10:11	MATLAB Toolbox	712 KB
GUI Layout Toolbox 2.2	07/11/2016 10:16	MATLAB Toolbox	711 KB
GUI Layout Toolbox 2.3.1	30/01/2017 21:40	MATLAB Toolbox	774 KB
GUI Layout Toolbox 2.3.2	01/05/2018 14:11	MATLAB Toolbox	850 KB
GUI Layout Toolbox 2.3	18/11/2016 17:03	MATLAB Toolbox	811 KB

# Report Generator

Help

Create Presentation Content

## Documentation

CONTENTS

### Create Presentation Content

Use the MATLAB® API for PowerPoint® (PPT API) to create presentation content

Use the PPT API to create MATLAB programs to add content to PowerPoint presentations and to create complete PowerPoint presentations. You can create a program that works with the slide master and layouts in a PowerPoint presentation. To get started, see [Create a Presentation Generator](#).

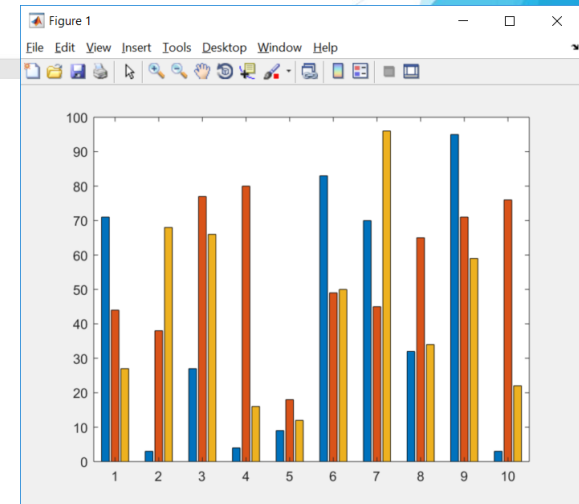
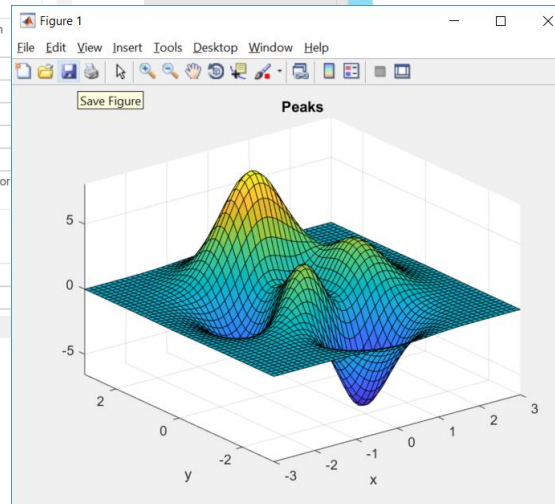
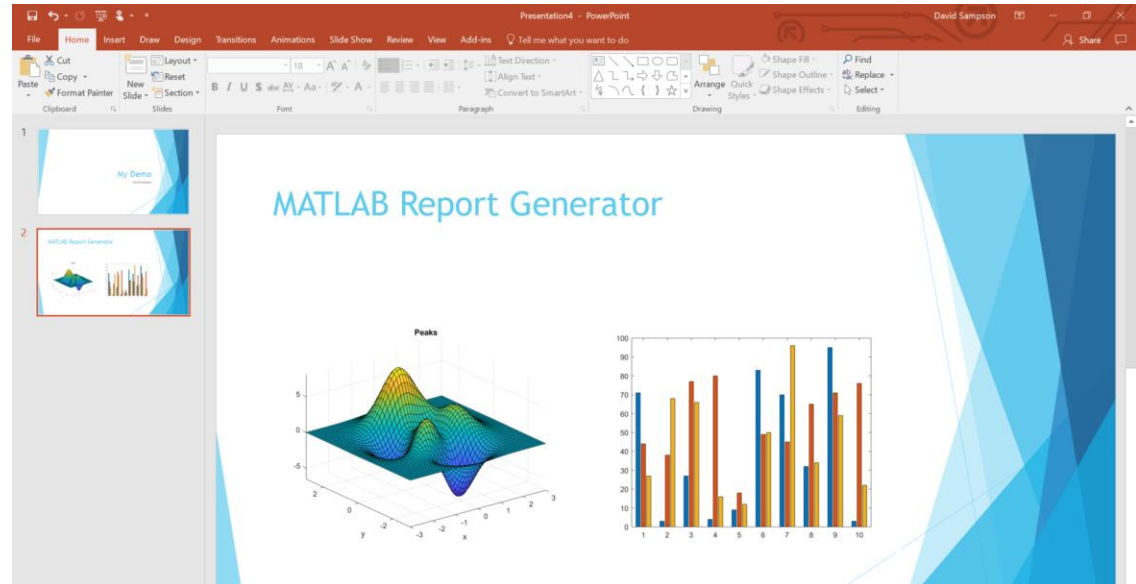
To share your completed presentation program with others who do not have MATLAB installed on their systems, see [Compile a Presentation Program](#).

#### Functions

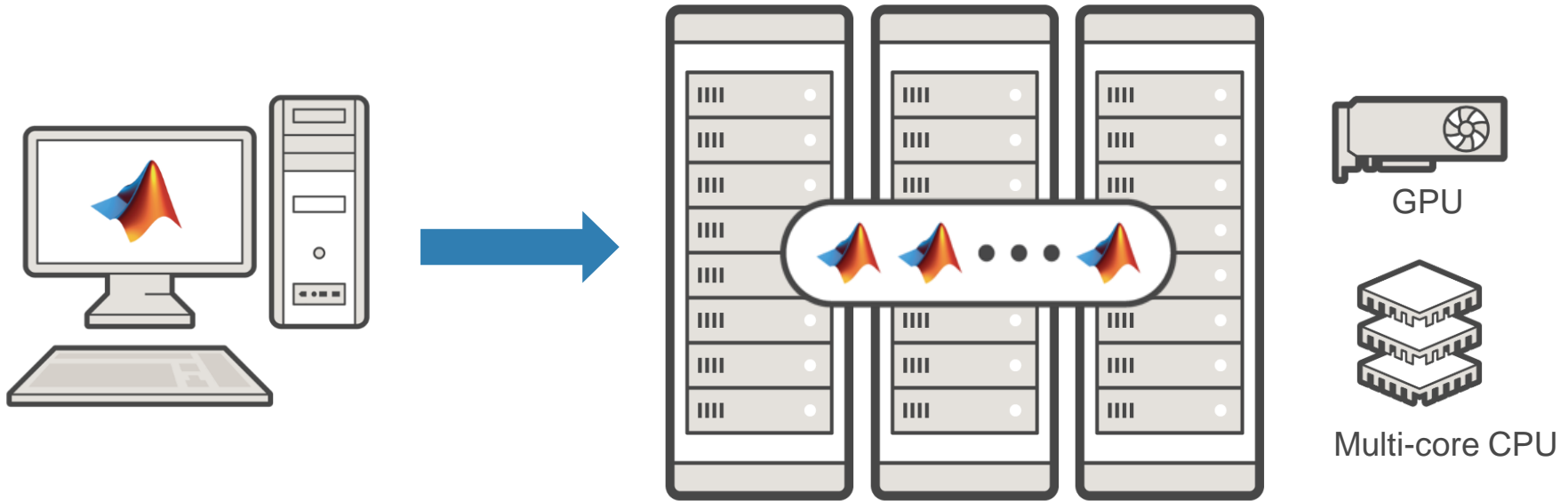
<code>mlreportgen.ppt.Presentation.open</code>	Open presentation
<code>mlreportgen.ppt.Presentation.close</code>	Close presentation
<code>mlreportgen.ppt.Presentation.getMasterNames</code>	Get names of slide masters for presentation
<code>mlreportgen.ppt.Presentation.getLayoutNames</code>	Get names of layouts for presentation slide master
<code>mlreportgen.ppt.Presentation.getTableStyleNames</code>	Get table style names for presentation
<code>mlreportgen.ppt.Presentation.add</code>	Add slide to presentation
<code>mlreportgen.ppt.Presentation.replace</code>	Replace paragraphs, tables, or pictures in presentation
<code>mlreportgen.ppt.MessageDispatcher.dispatch</code>	Dispatch PPT status message
<code>mlreportgen.ppt.MessageDispatcher.getTheDispatcher</code>	Return PPT message dispatcher
<code>mlreportgen.ppt.ProgressMessage.formatAsText</code>	Format message as text
<code>mlreportgen.ppt.ProgressMessage.formatAsHTML</code>	Wrap message in HTML tags
<code>mlreportgen.ppt.ProgressMessage.passesFilter</code>	Determine if message passes filter
<code>pptview</code>	Open Microsoft PowerPoint presentation or convert it to PDF

#### Classes

<code>mlreportgen.ppt.Presentation</code>	Create Microsoft PowerPoint presentation
<code>mlreportgen.ppt.Slide</code>	Presentation slide



# Parallel

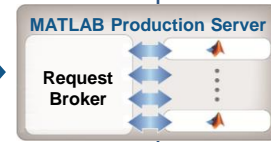
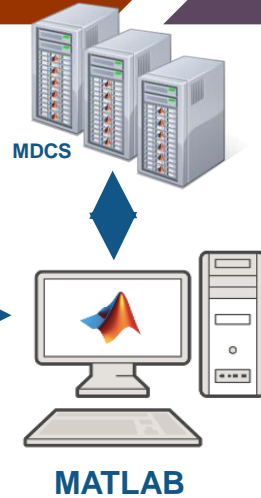


- High-level: `parfor`, `gpuArray`
- Low-level: `batch`, `createJob`, `createTask`
- Big data: `tall`, `mapreduce`

# Enterprise integration



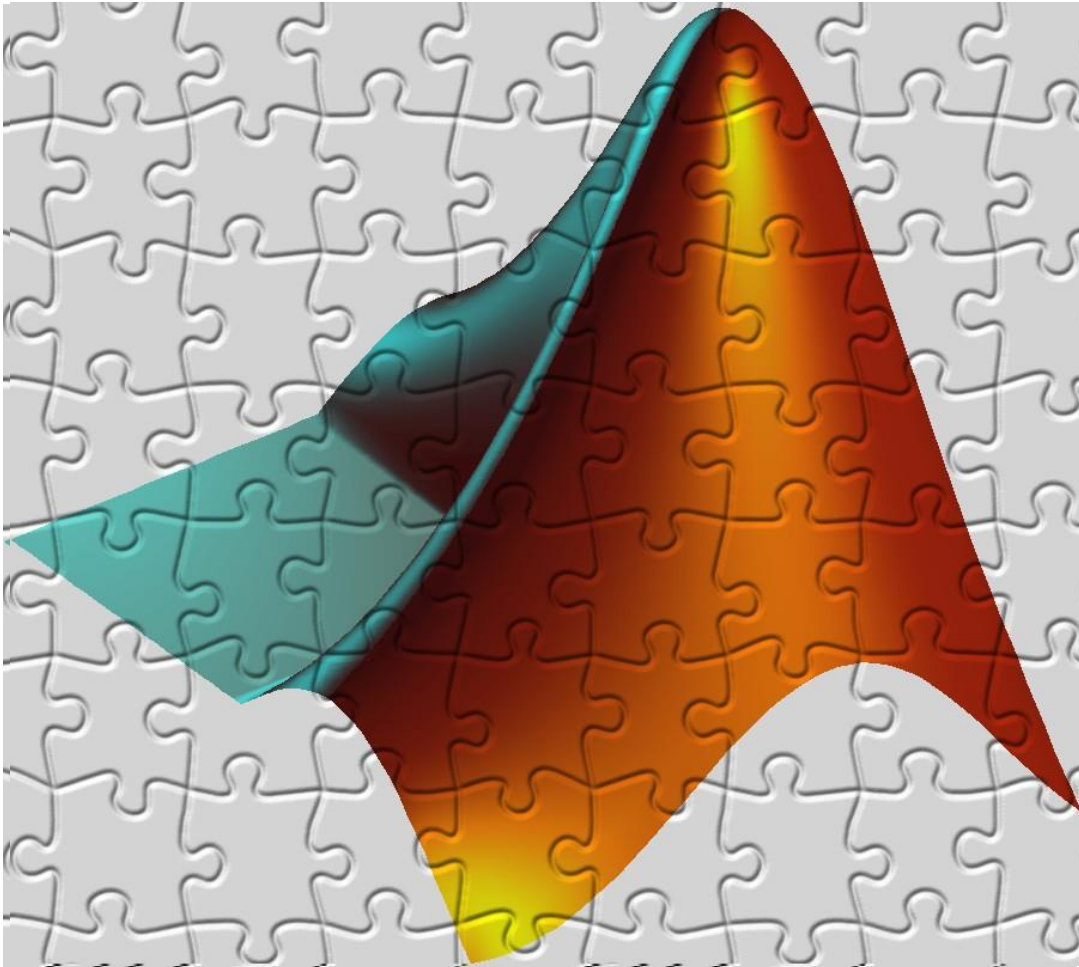
<b>Databases</b>	DynamoDB Microsoft SQL Server mongoDB Cassandra Cosmos DB
<b>Cloud Storage</b>	S3 Azure Blob 1001
<b>Big Data / OT</b>	cloudera HORTONWORKS OSIsoft. PI System



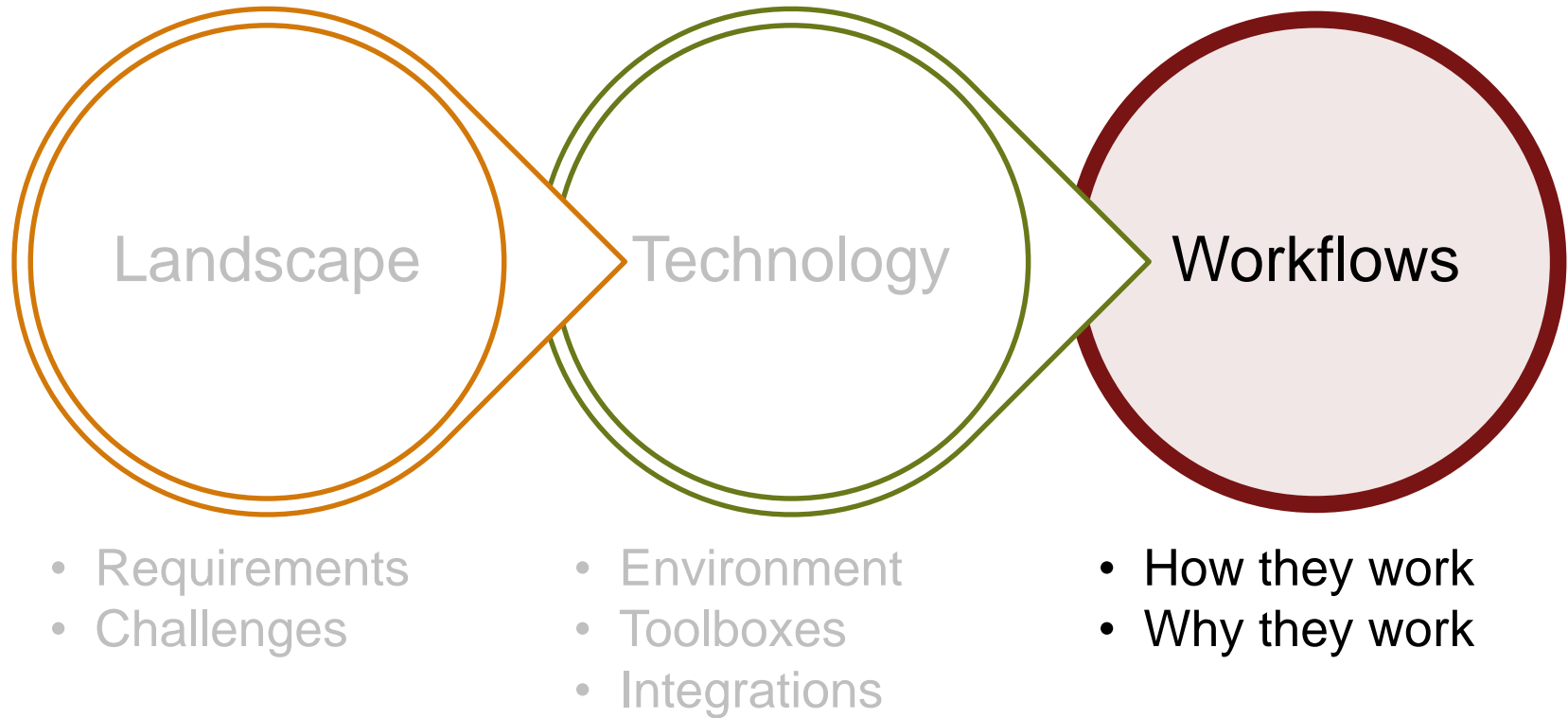
AWS Kinesis kafka Azure IoT Hub	<b>Streaming</b>
OSIsoft. PI System	<b>OT Platforms</b>
Qlik tableau Microsoft Power BI	<b>Dashboards</b>
Spotfire	

Azure    amazon web services™    rackspace    openstack™    vmware®









**Access and Explore Data**

**Files**

---

**Databases**

---

**Sensors**

**Preprocess Data**

**Working with Messy Data**

---

**Data Reduction/Transformation**

---

**Feature Extraction**

**Develop Predictive Models**

**Model Creation e.g. Machine Learning**

---

**Parameter Optimization**

---

**Model Validation**

**Integrate Analytics with Systems**

**Desktop Apps**

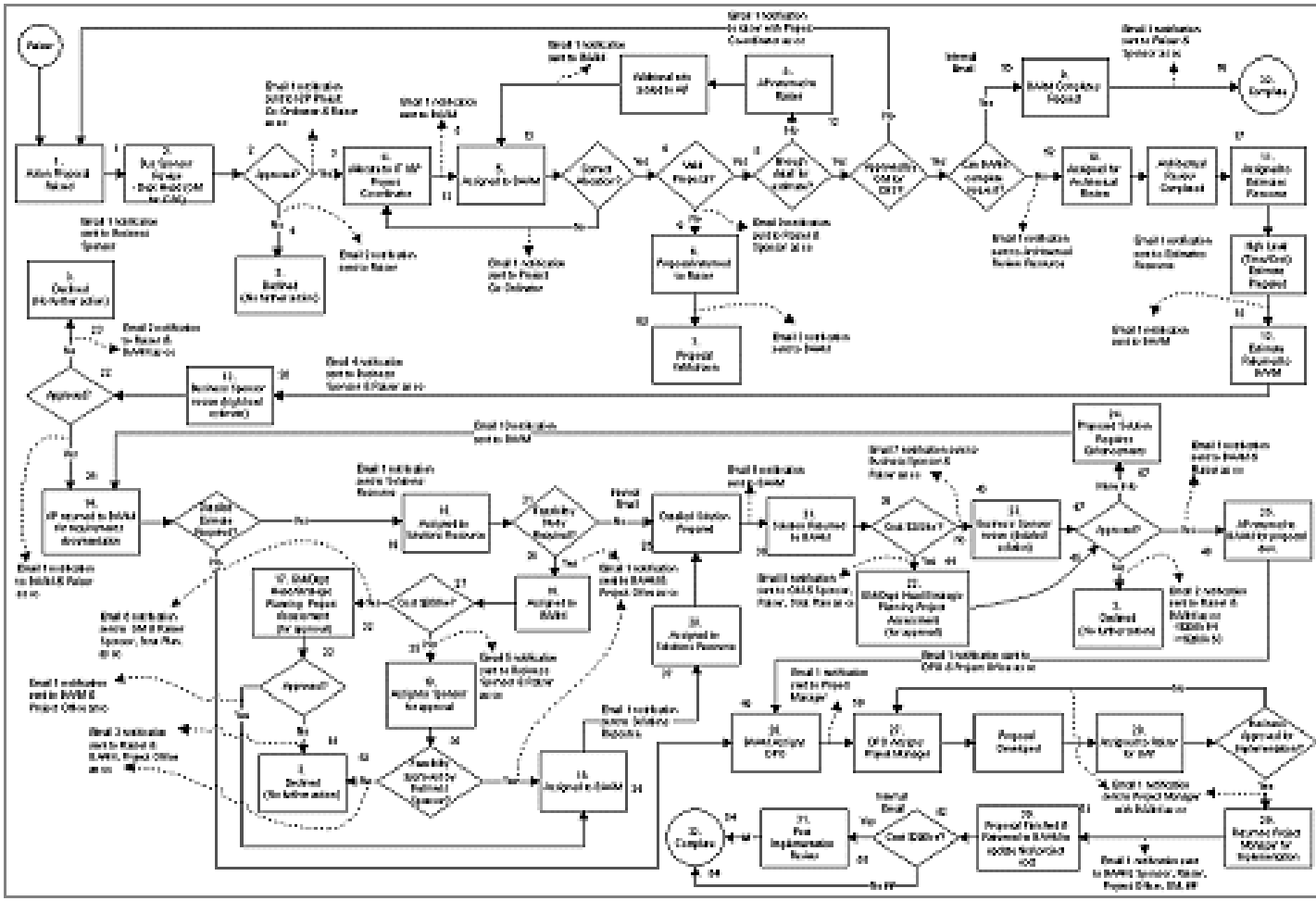
---

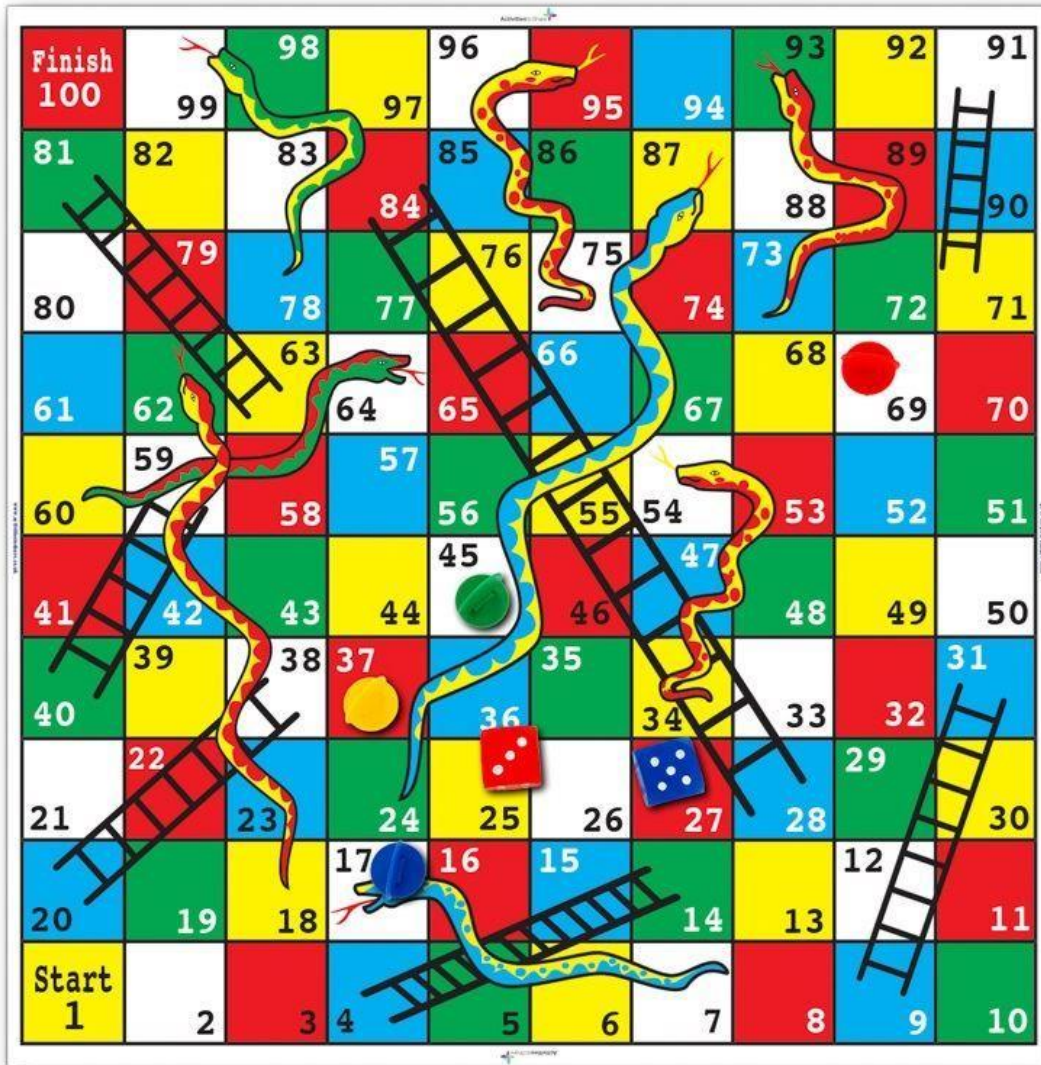
**Enterprise Scale Systems**

**MATLAB** Excel  
.NET C/C++  
.exe .dll  
Java

---

**Embedded Devices and Hardware**





# Data preparation

**Import - C:\Documents\MATLAB\EnergyData.txt**

IMPORT VIEW

Delimited Column delimiters: Tab Range: A2:G745 Variable Names Row: 1

Column vectors Matrix Cell Array

UNIMPORTABLE CELLS Import Selection

Import Data Generate Script Generate Function

Time	Day	Holiday	Power	Temperature	WindDirection	WindSpeedMS
DATE/TIME	TEXT	NUMBER	NUMBER	NUMBER	TEXT	NUMBER
1/1/2006 0:00	Sun	0	54.5448 MW	19 °F	West	8.225536
1/1/2006 1:00	Sun	0	52.3898 MW	18.85 °F	West	8.761984
1/1/2006 2:00	Sun	0	51.6344 MW	17.865 °F	West	8.761984
1/1/2006 3:00	Sun	0	51.5597 MW	17.28 °F	West	7.197344
1/1/2006 4:00	Sun	0	51.7148 MW	15.9182 °F	West	7.733792
1/1/2006 5:00	Sun	0	52.6898 MW	16.24 °F	West	6.169152
1/1/2006 6:00	Sun	0	55.341 MW	17.525 °F	WNW	7.197344
1/1/2006 7:00	Sun	0	57.9512 MW	17.235 °F	WNW	7.733792
1/1/2006 8:00	Sun	0	62.3844 MW	18.15 °F	West	7.197344
1/1/2006 9:00	Sun	0	66.2962 MW	19.3 °F	West	5.677408
1/1/2006 10:00	Sun	0	67.9479 MW	21.0316 °F	West	5.677408
1/1/2006 11:00	Sun	0	68.4049 MW	22.065 °F	West	6.169152
1/1/2006 12:00	Sun	0	67.4961 MW	23 °F	WNW	6.7056
1/1/2006 13:00	Sun	0	66.2013 MW	24.1 °F	WNW	7.733792
1/1/2006 14:00	Sun	0	64.954 MW	24.235 °F	WNW	5.677408
1/1/2006 15:00	Sun	0	65.8897 MW	25 °F	WNW	6.169152

2D Plot: BTQ [Nm] vs SPARK [DegBTDC]

3D Plot: BTQ vs SPARK vs SPEED

Data Table:

	AFR	AIRMASS	BTQ	COFLOW	CYC_COU	EXHBP	EXH_ANG	EXH_LFT
10	14.3	0.018	-11.483	0.016	159	1.031e5	1	1
11	14.3	0.019	-30.276	7.6e-3	165	1.022e5	1	1
2	14.3	0.034	67.449	0.143	105	1.059e5	1	1
13	14.3	0.034	67.587	0.143	111	1.06e5	1	1
14	14.3	0.034	67.562	0.143	117	1.06e5	1	1
15	14.3	0.034	67.325	0.143	123	1.061e5	1	1
16	14.3	0.034	66.884	0.143	129	1.061e5	1	1
17	14.3	0.034	65.534	0.143	135	1.062e5	1	1
18	14.3	0.034	63.392	0.144	141	1.064e5	1	1
19	14.3	0.034	59.834	0.146	147	1.065e5	1	1
20	13.67	0.034	56.045	0.633	171	1.066e5	1	1
21	12.26	0.034	48.47	1.842	222	1.069e5	1	1
22	9.59	0.034	-8.617	3.76	315	1.072e5	1	1
3	14.3	0.033	63.61	0.142	105	1.055e5	1	1

3D Data Plot: BTQ vs SPARK vs SPEED

X-axis factor: SPEED Y-axis factor: SPARK Z-axis factor: BTQ

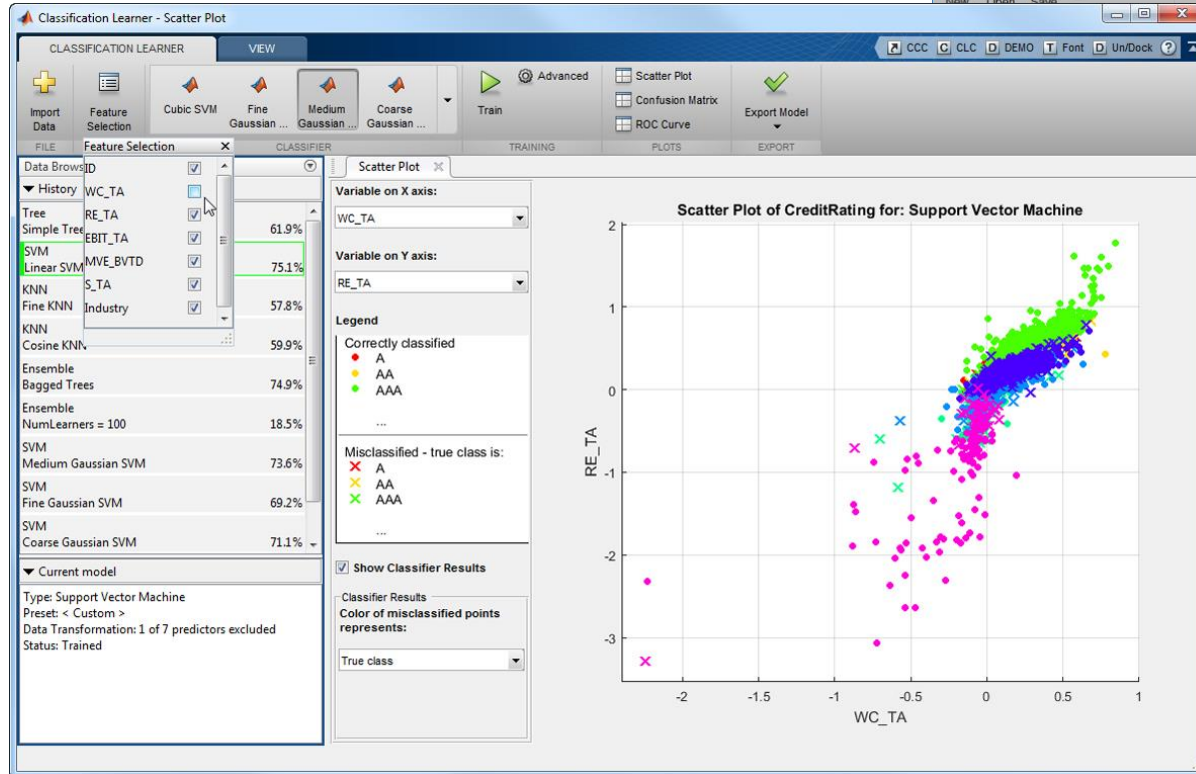
Data has 1615/2222 Records, 27 + 0 Variables, and 189 Tests.

>> importdata

Assemble a sufficient volume of clean data of known provenance.



# Model generation



```

Editor - C:\Users\dsampson\Downloads\trainClassifier_weightedKNN.m
EDITOR PUBLISH VIEW
Find Files Go To Insert Comment % Indent Breakpoints Run Run and Advance Run and Time
Run Run and Advance Run and Time

and response
es the data into the right shape for training the

ngData;
AccMeanX', 'AccMeanY', 'AccMeanZ', 'AccStdX', 'AccStdY',
table(:, predictorNames);
le.Activity;
tor = [false, false, false, false, false, false, false,

e predictor matrix.
c predictors only. Categorical predictors are passed thro
torBeforePCA = isCategoricalPredictor;
predictors(:, ~isCategoricalPredictor);
table2array(varfun(@double, numericPredictors));
to be treated as missing data for PCA.
sinf(numericPredictors)) = NaN;
caScores, ~, ~, explained, pcaCenters] = pca(...
rs);
ments to explain the desired amount of variance.
KeepAsFraction = 95/100;
= find(cumsum(explained)/sum(explained) >= explainedVar
caCoefficients(:, 1:numComponentsToKeep);
2table(pcaScores(:, 1:numComponentsToKeep), predictors(:
tor = [false(1,numComponentsToKeep), true(1,sum(isCateg

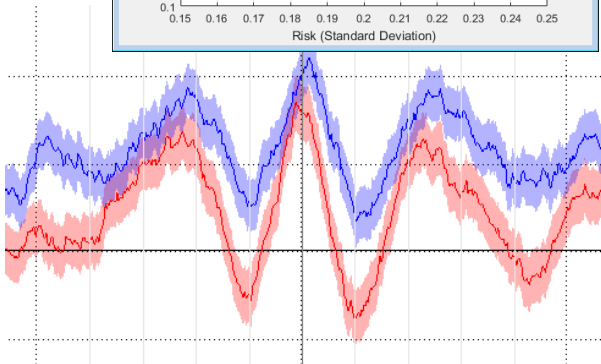
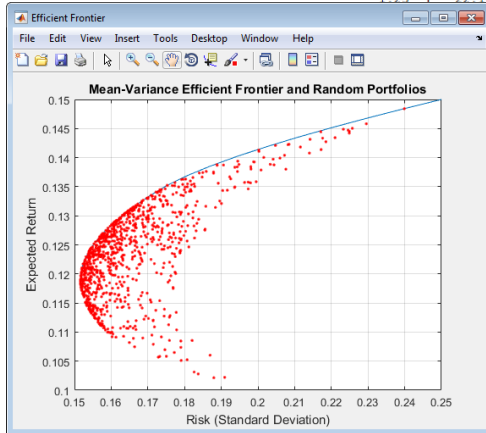
r
es all the classifier options and trains the classifier.
= fitctree(...
    
```

Create models with the required accuracy and reach, while driving insight.

# Documentation generation

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident,

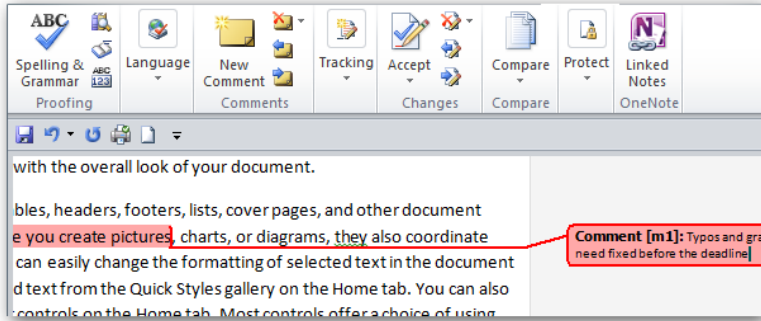
t	x(t)	y(t)	vx(t)	vy(t)	v(t)
0.00	0.00	0.00	17.68	17.68	25.00
0.25	4.42	4.11	17.68	15.23	23.33
0.50	8.84	7.61	17.68	12.77	21.81
0.75	13.26	10.50	17.68	10.32	20.47
1.00	17.68	12.77	17.68	7.87	19.35
1.25	22.10	14.43	17.68	5.42	18.49
		15.48	17.68	2.96	17.92
		15.91	17.68	0.51	17.69
		15.74	17.68	-1.94	17.78
		14.94	17.68	-4.39	18.22
		13.54	17.68	-6.85	18.96
		11.52	17.68	-9.30	19.97
		8.89	17.68	-11.75	21.23
		5.64	17.68	-14.20	22.68
		1.79	17.68	-16.66	24.29



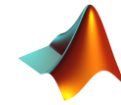
The presentation software interface includes a sidebar with options like 'New', 'Open', 'Save', 'Share', 'Export', 'Close', 'Account', 'Feedback', and 'Options'. The main slide area displays a bar chart with four categories (Category 1 to Category 4) and three series (Series 1, Series 2, Series 3). A 'Facet' panel on the right shows different chart styles. The bottom of the interface has navigation controls and a search bar.

Generate an accurate, insightful description of the study in a timely manner.

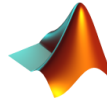
# Model review



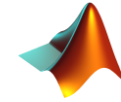
Documents



Examples



Models



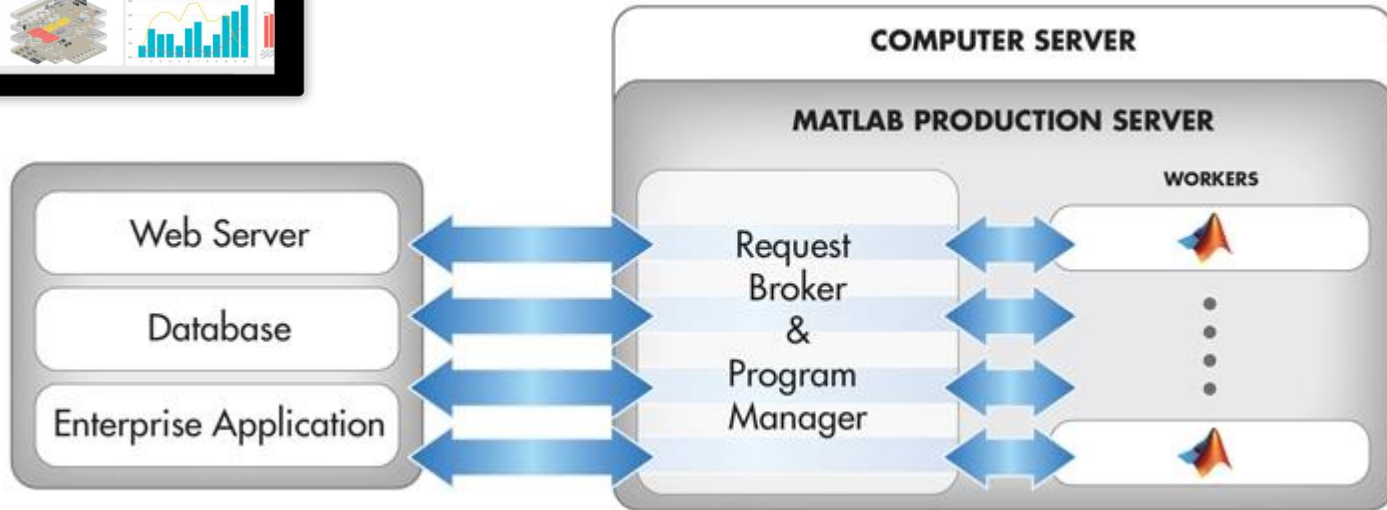
Toolbox code



Provide an accurate, thorough view of the study that allows others to engage.



# Model deployment



Provide approved, accurate, current models for use throughout the business.

