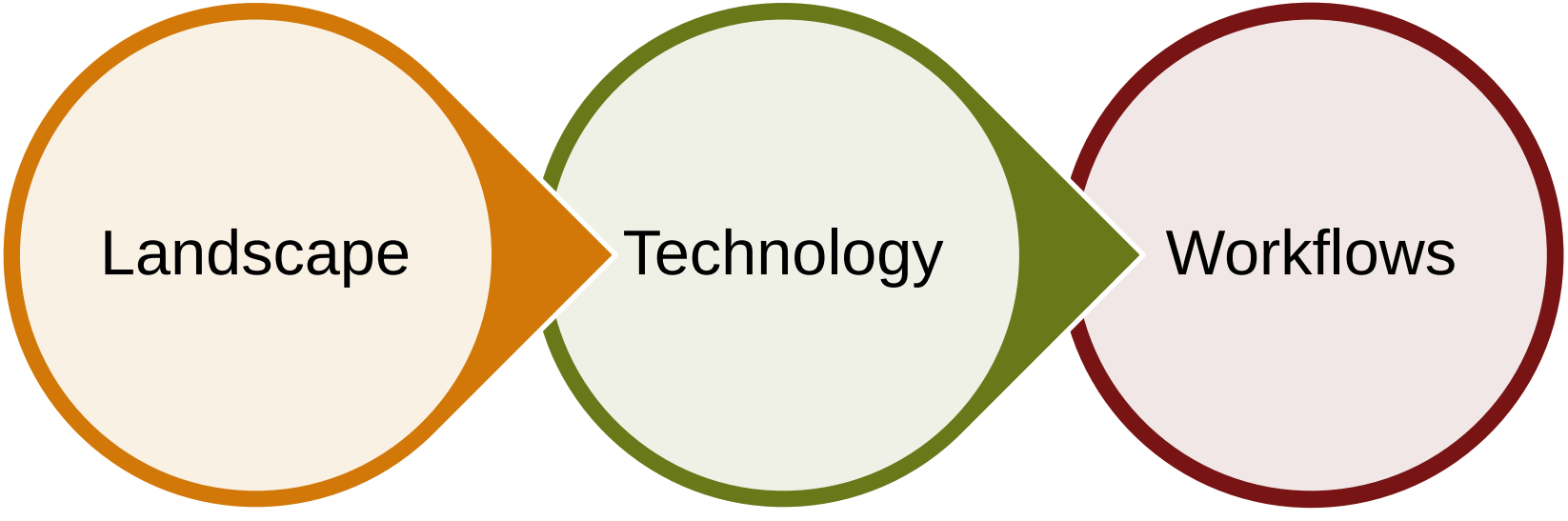
Streamlining Financial Modelling: From Development to Approval to Production with MATLAB

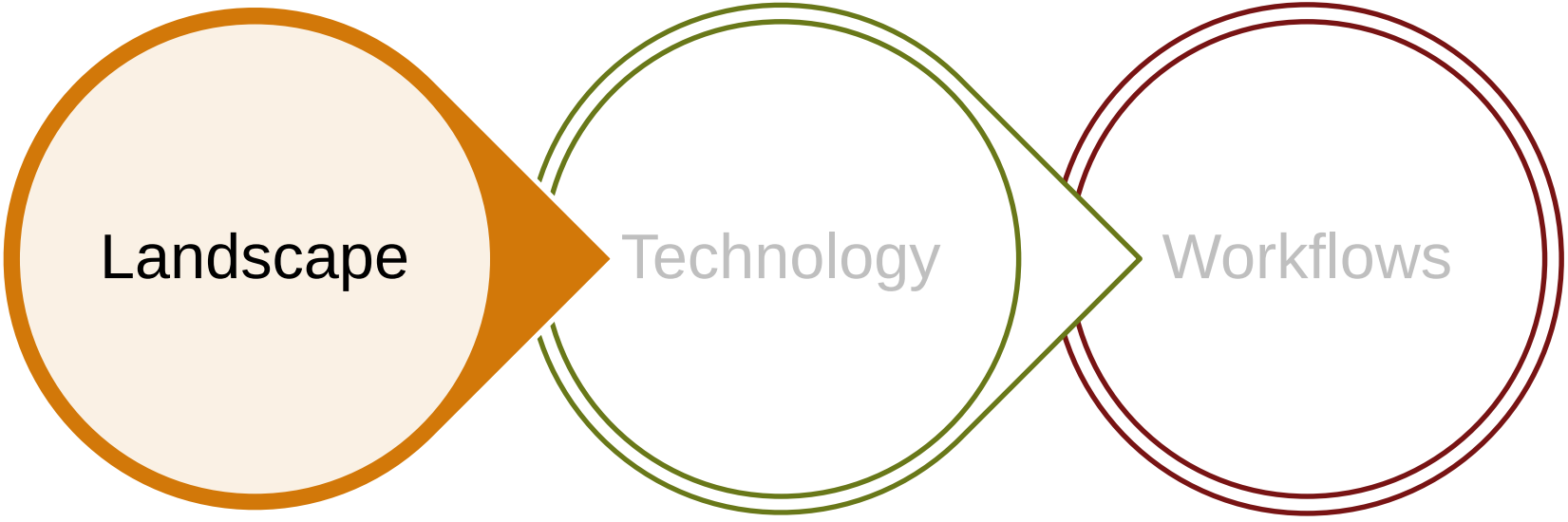
David Sampson
MathWorks



- Requirements
- Challenges

- Environment
- Toolboxes
- Integrations

- How they work
- Why they work

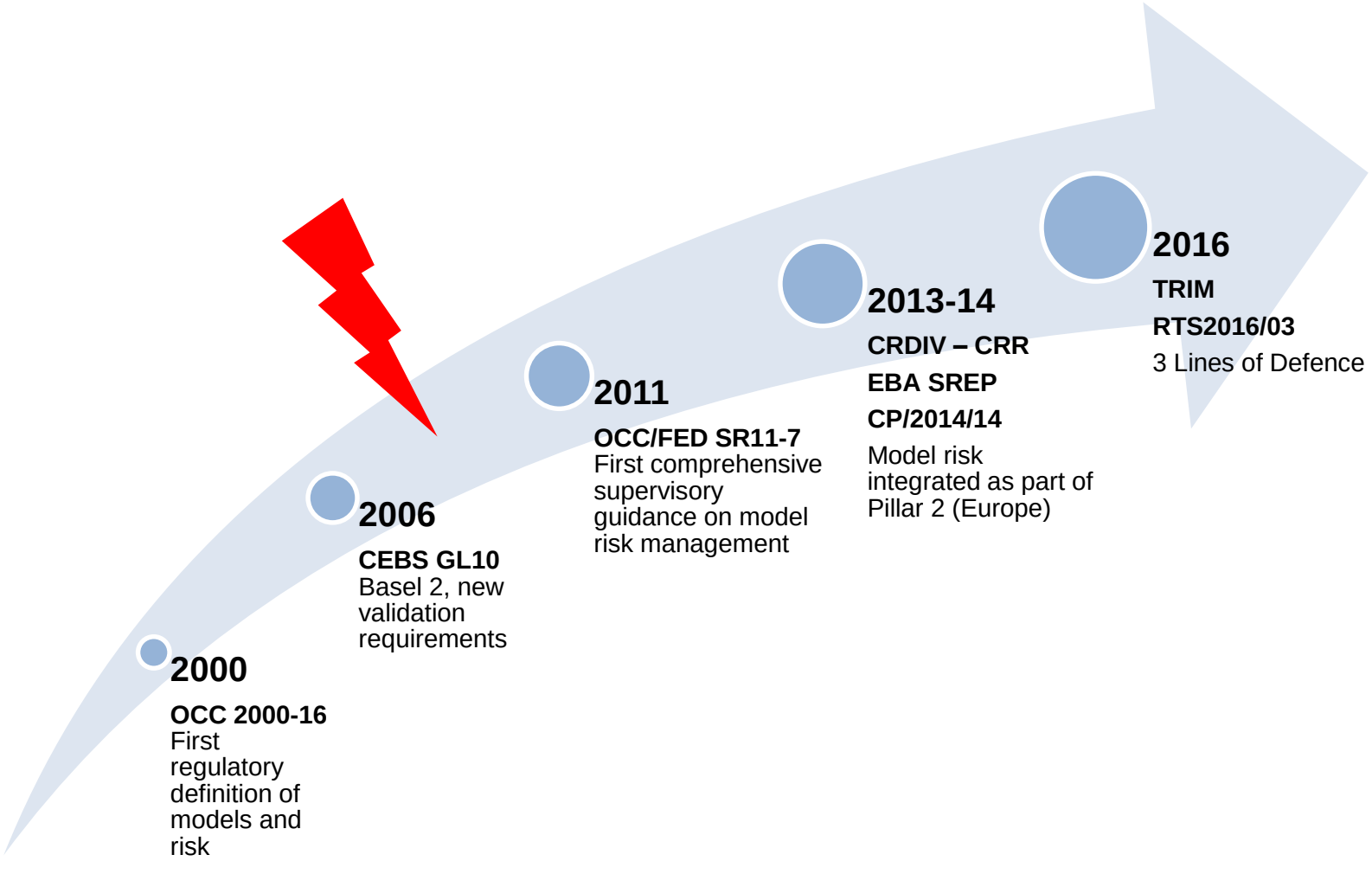


- Requirements
- Challenges

- Environment
- Toolboxes
- Integrations

- How they work
- Why they work

Model risk management regulations




Definition
Framework
Development
Review
Risk Management

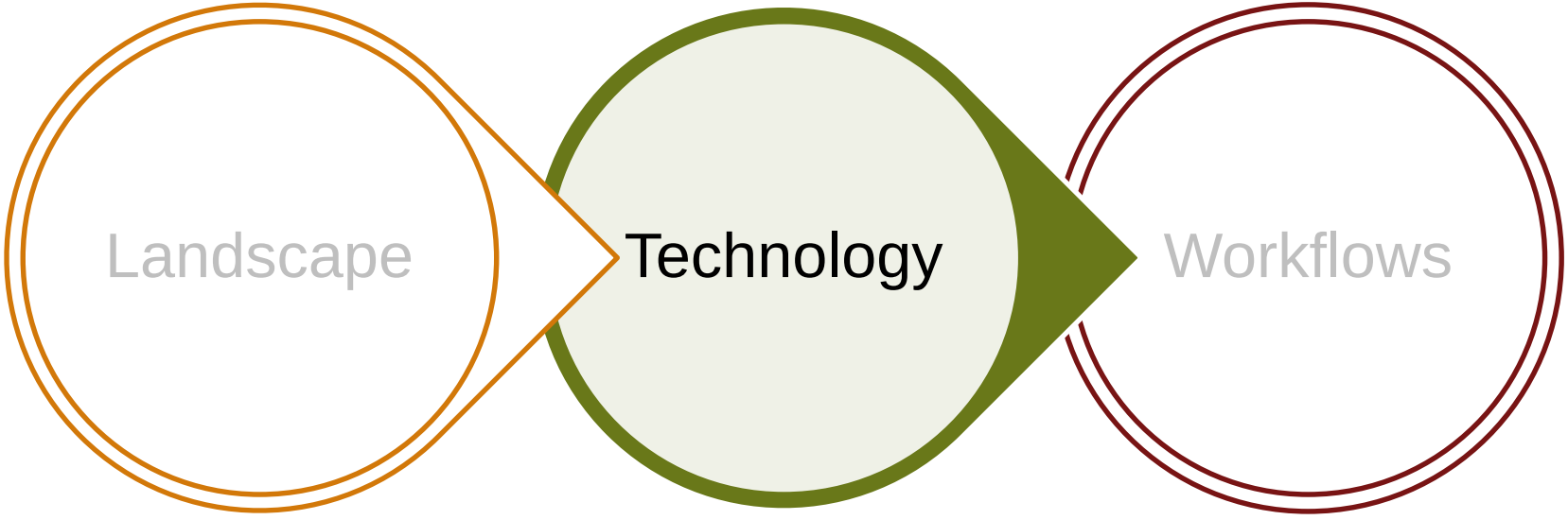
Regulator requirements

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


Backtesting
Alerts
Monitoring
Inventory
Scope
Definition
Integration
Testing
Deployment
Development
Assumptions
Limitations
Approval
Validation
Management
Regulator
Internal
Independent

Institution challenges

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

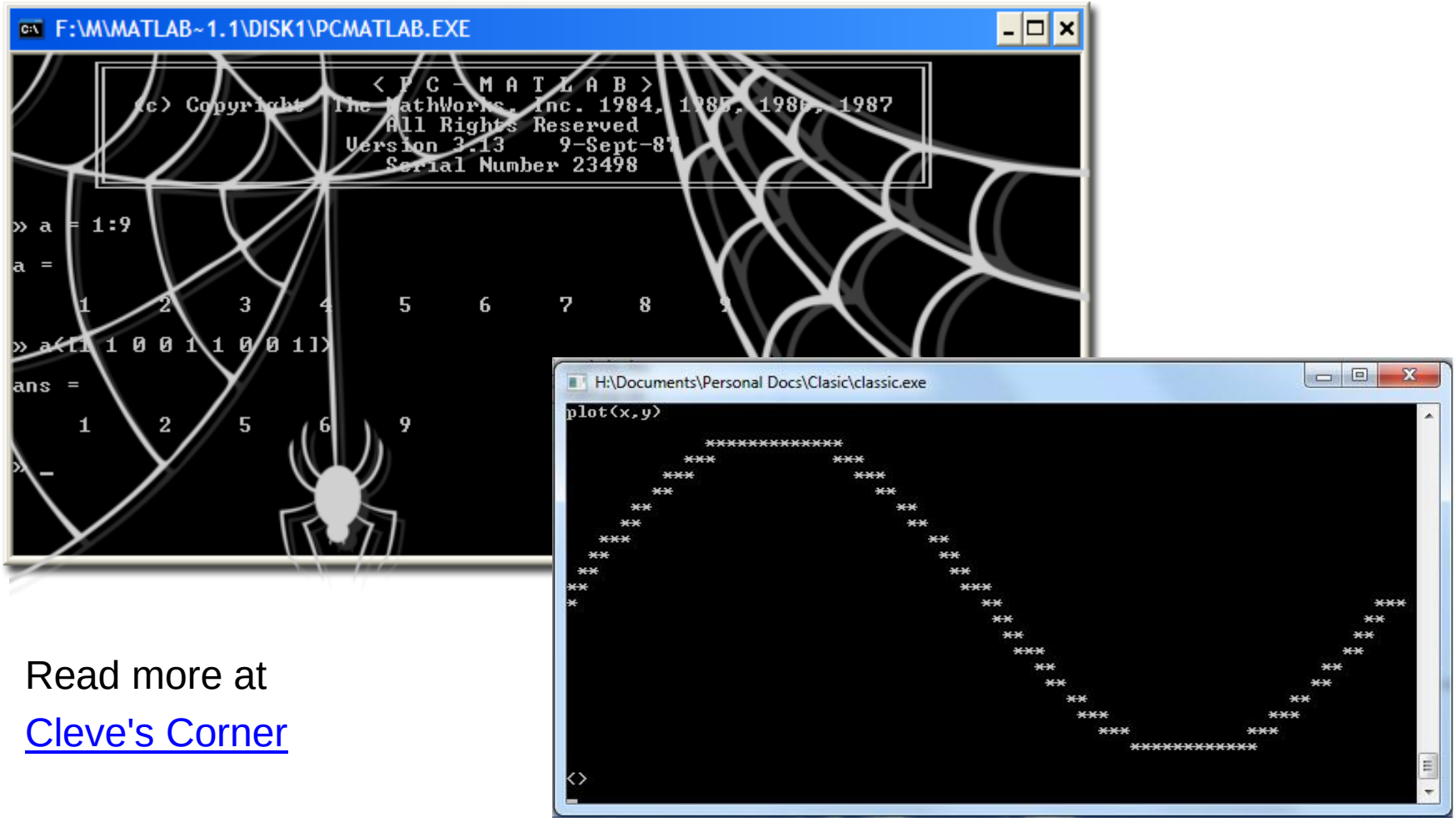


- Requirements
- Challenges

- Environment
- Toolboxes
- Integrations

- How they work
- Why they work

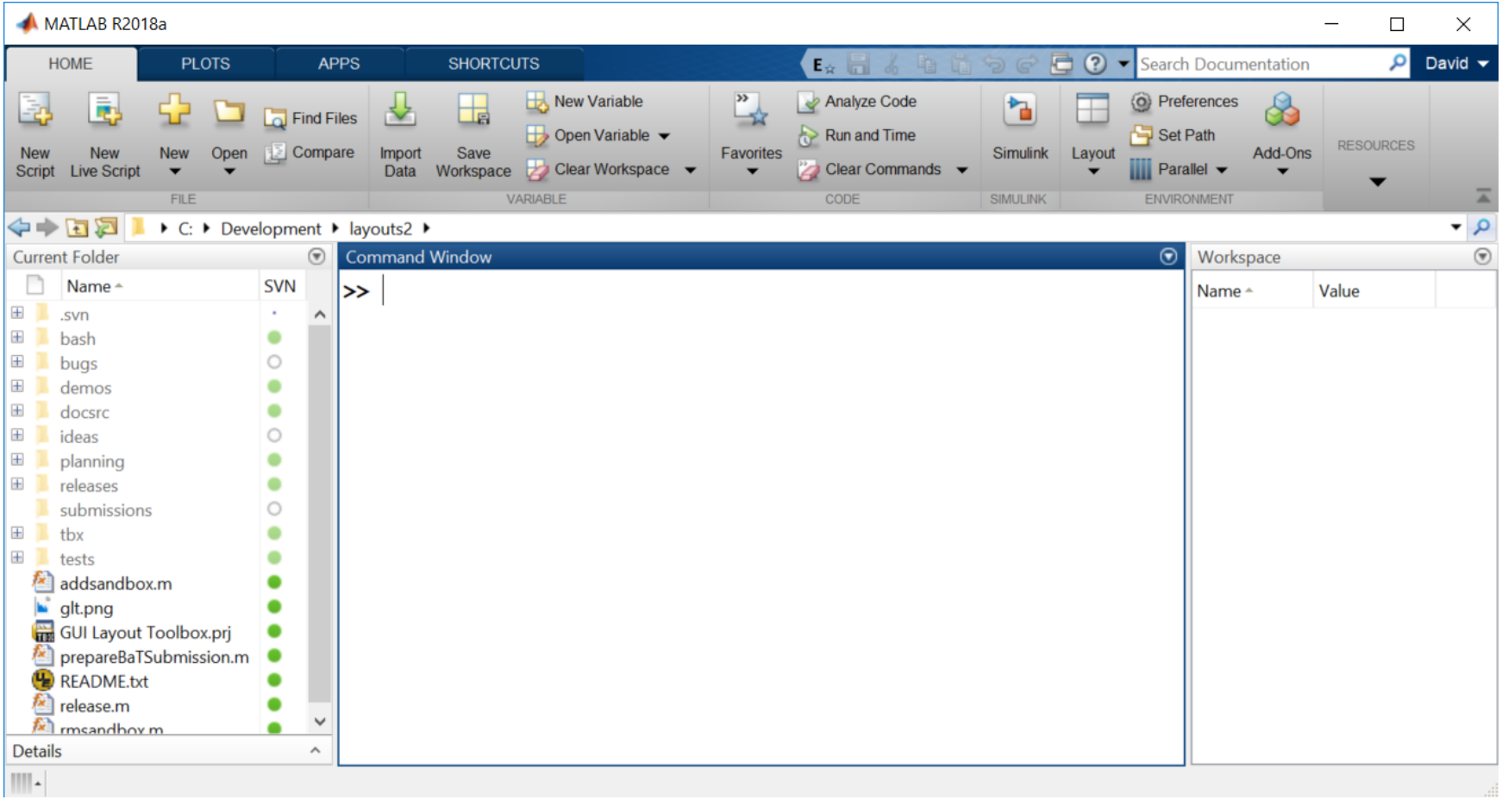
In the beginning…



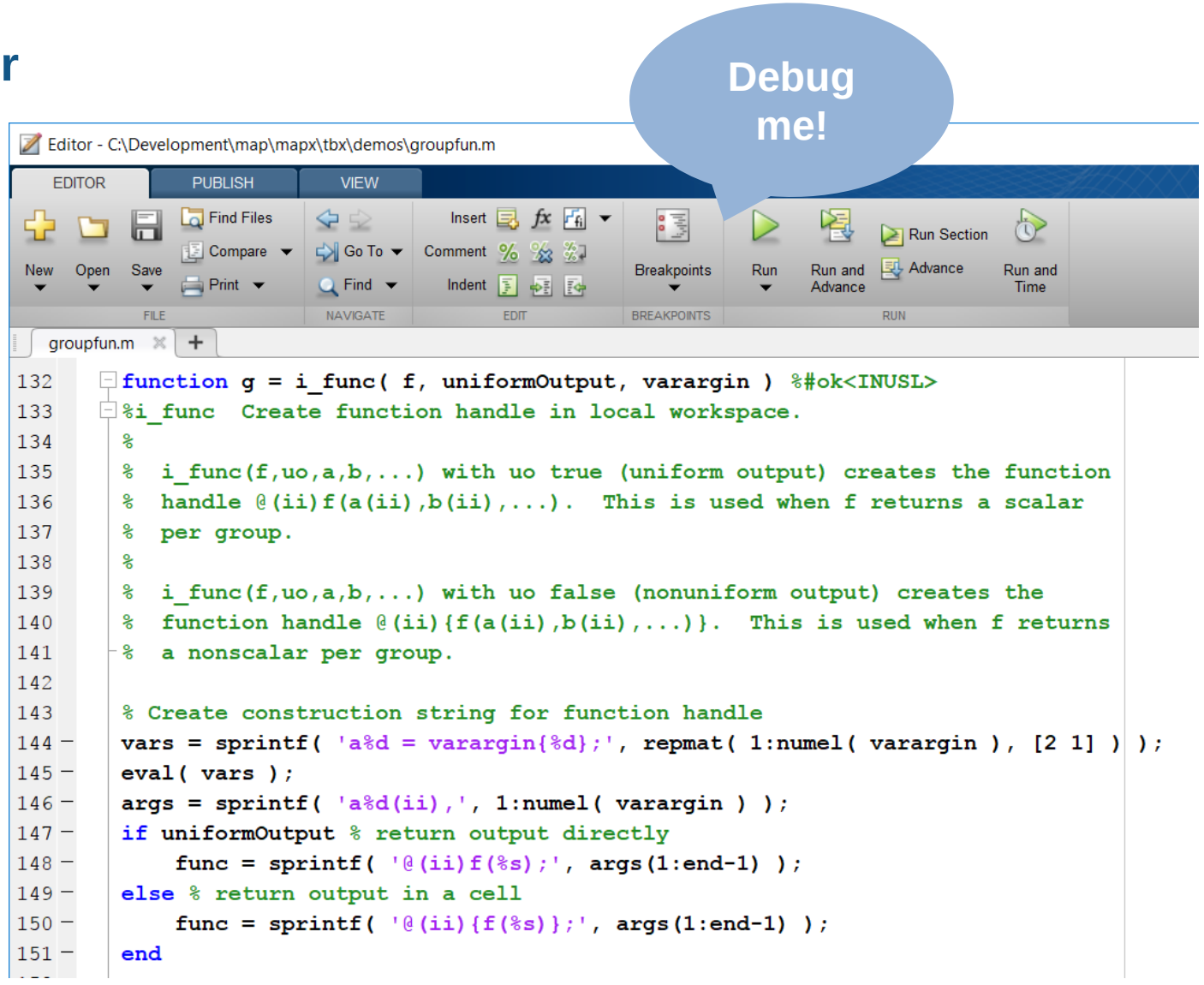
Read more at

[Cleve's Corner]

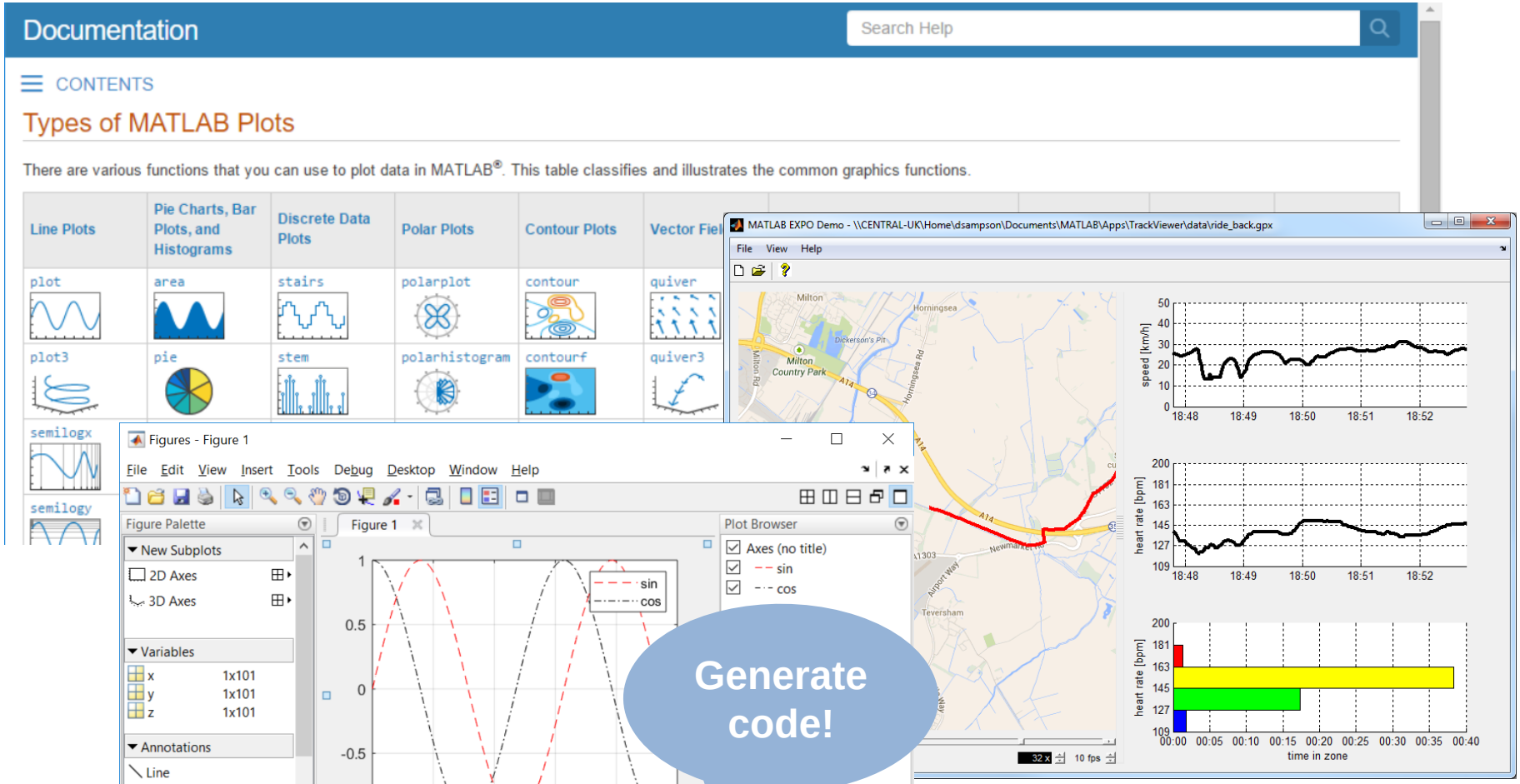
Desktop


MATLAB R2018a
HOME
PLOTS
APPS
SHORTCUTS
Search Documentation
David
New Script
New Live Script
New
Open
Find Files
Compare
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
Favorites
Analyze Code
Run and Time
Clear Commands
Simulink
Layout
Preferences
Set Path
Parallel
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

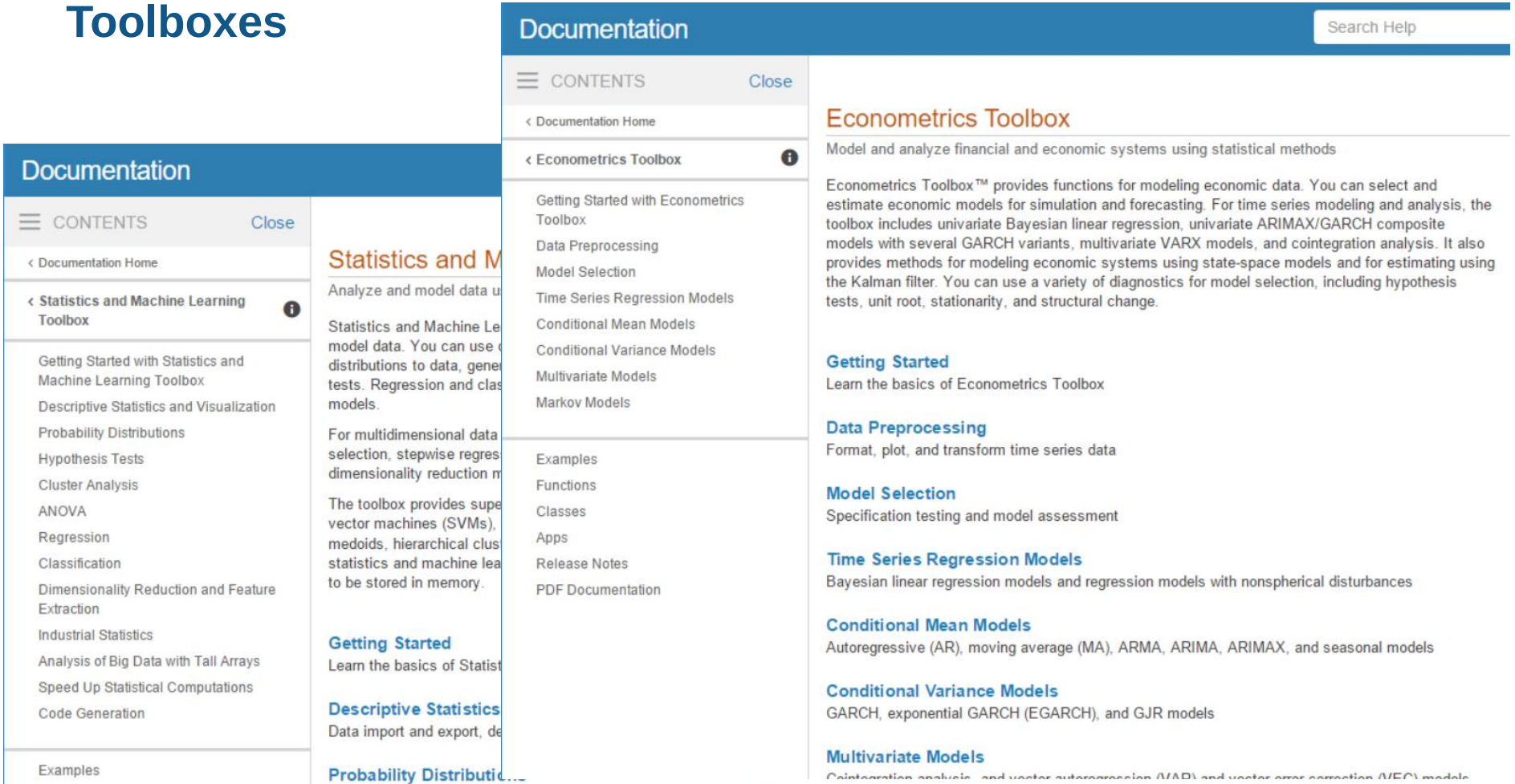
Editor


Debug me!
Editor - C:\Development\map\mapx\tbx\demos\groupfun.m
EDITOR
PUBLISH
VIEW
New
Open
Save
Find Files
Compare
Print
Go To
Find
Insert
Comment
Indent
Breakpoints
Run
Run and Advance
Run Section
Advance
Run and Time
FILE
NAVIGATE
EDIT
BREAKPOINTS
RUN
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
Search Help
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
plot
area
stairs
polarplot
contour
plot3
pie
stem
polarhistogram
contourf
quiver3
semilogx
semilogy
MATLAB EXPO Demo - \\CENTRAL-UK\Home\dsampson\Documents\MATLAB\Apps\TrackViewer\data\ride_back.gpx
File View Help
Milton
Horningsea
Milton Country Park
Dickerson's Pit
A14
Teversham
Airport Way
speed [km/h]
heart rate [bpm]
time in zone
Figures - Figure 1
File Edit View Insert Tools Debug Desktop Window Help
Figure Palette
New Subplots
2D Axes
3D Axes
Variables
x 1x101
y 1x101
z 1x101
Annotations
Line
Arrow
Double Arrow
Figure 1
sin
cos
Plot Browser
Axes (no title)
Add Data
Generate code!

Toolboxes


Documentation
CONTENTS
Close
Documentation Home
Statistics and Machine Learning Toolbox
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
Statistics and M
Analyze and model data u
Statistics and Machine Le
model data. You can use
distributions to data, gene
tests. Regression and clas
models.
For multidimensional data
selection, stepwise regres
dimensionality reduction m
The toolbox provides supe
vector machines (SVMs),
medoids, hierarchical clus
statistics and machine lea
to be stored in memory.
Getting Started
Learn the basics of Statist
Descriptive Statistics
Data import and export, de
Probability Distributio
Data frequency models, random sample generation, parameter estimation
Hypothesis Tests
t-test, F-test, chi-square goodness-of-fit test, and more
Documentation
Search Help
CONTENTS
Close
Documentation Home
Econometrics Toolbox
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
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

Documentation Browser

Documentation

Search Help

CONTENTS Close

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

Explore Examples Explore Add-Ons

MATLAB
MATLAB® is the high-level language and interactive environment used by millions of engineers and scientists to analyze and design systems and products across many domains. Learn MATLAB in one hour.

Getting Started with MATLAB

Click and drag to move MATLAB Documentation or its

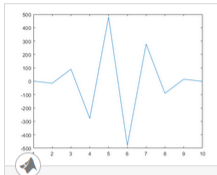
Examples

Search Help

CONTENTS

MATLAB Examples

Language Fundamentals



Basic Matrix Operations

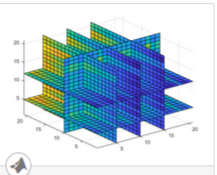
Basic techniques and functions for working with matrices in the MATLAB® language.

Open Live Script

Matrix Manipulation

Do some basic matrix manipulations in MATLAB®.

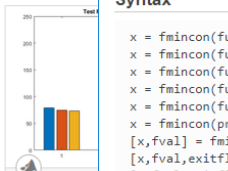
Open Live Script



Manipulating Multidimensional Arrays

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

Open Live Script



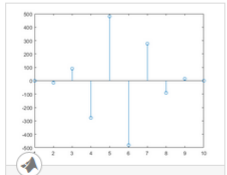
Create Stru

Create a struct is a data type t data using data fields. Each fiel

Mathematics

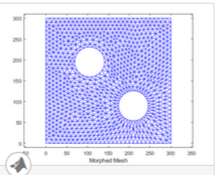
Integer Arithmetic

Perform arithmetic on integer data representing signals and images.



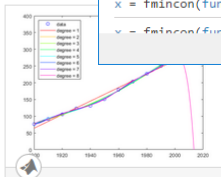
Single Precision Math

Perform arithmetic and linear algebra with single precision data. It



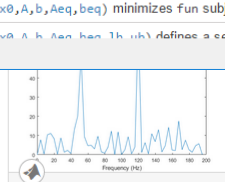
Creating and Editing Delaunay Triangulations

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

file:///C:/Program Files/MATLAB/R2018a/help/matlab/examples/create-a-structure-array.html

Documentation

CONTENTS

fmincon

Find minimum of constrained nonlinear multivariable function

Nonlinear programming solver.

Finds the minimum of a problem specified by

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

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 tha

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 matrix.

Note

Passing Extra Parameters explains how to pass extra parameters to the objective function and nonlinear const

`x = fmincon(fun,x0,A,b,Aeq,beq)` minimizes `fun` subject to the linear equalities `Aeq*x = beq` and `A*x ≤ b`. If no

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

Variable Editor

Command Window

```
>> TMW

TMW =

  1000×5 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

New from Selection | Open | Print | Rows | Columns | Insert | Delete | Transpose | Sort

VARIABLE | SELECTION | EDIT

TMW

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\dsampson\AppData\Local\MathWorks\MATLAB\Examples\matlab\LiveEditorInteractiveNarrative\LiveE

LIVE EDITOR | INSERT | VIEW

New | Open | Save | Find Files | Compare | Print | Go To | Find | Text | Title | Code | Section Break | Run Sectio | Run and Ac | Run to End

FILE | NAVIGATE | TEXT | CODE | SECTION

LiveEditorInteractiveNarrative.mlx

```
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 ar

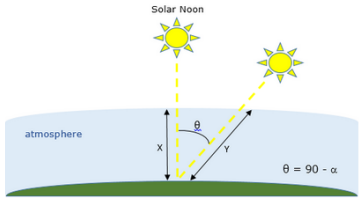
$$sunrise = 12 - \frac{\cos^{-1}(-\tan\phi\tan\delta)}{15^\circ} - \frac{TC}{60} \qquad sunset = 12 + \frac{\cos^{-1}(-\tan\phi\tan\delta)}{15^\circ}$$

```
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 anoth
*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 fu
diagram below, it is a measure of the length of the path of light through the atmosphere (Y) relative to the shortest possible pat



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^2) reaching the ground can be calculated from the empirical equation

$$sRad = 1.353 * 0.7^{AM^{0.678}}.$$

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

LIVE EDITOR | INSERT | VIEW

New | Open | Save | Find Files | Compare | Print | Go To | Find | Text | Title | Code | Section Break | Run Section | Run and Advance | Run to End | Run | Step | Stop

FILE | NAVIGATE | TEXT | CODE | SECTION | RUN

UseInteractiveControls.mlx

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

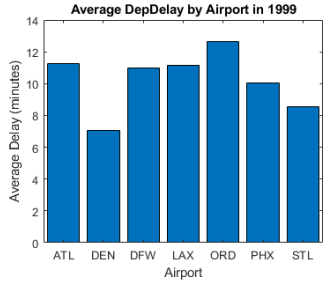
To add an interactive control, select a value in t… button, and select either **Numeric Slider** or **Drop Down**. To configure the interactive control, double-click it. When … s the section that contains the control.

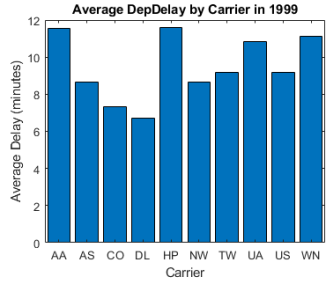
Visualize Airport and Carrier Delays

Use the interactive controls in the live script… arriers to visualize. The live script reads in the airport delay data using the `readDelays` function. The `calcula`… orts and the top carriers. The `plotDelays` function creates bar charts to display the delays for the top airports …

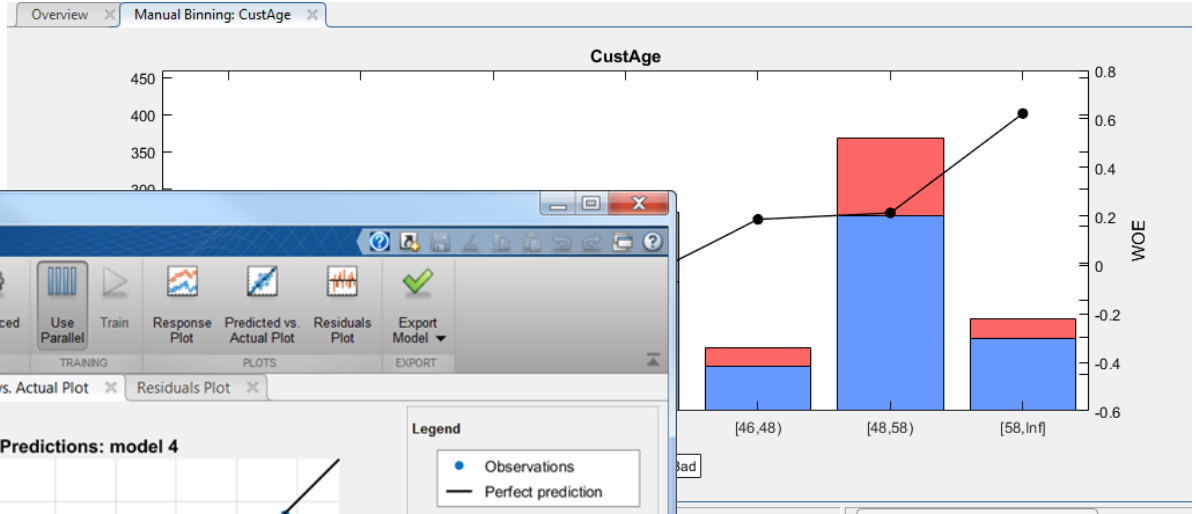
Change me!

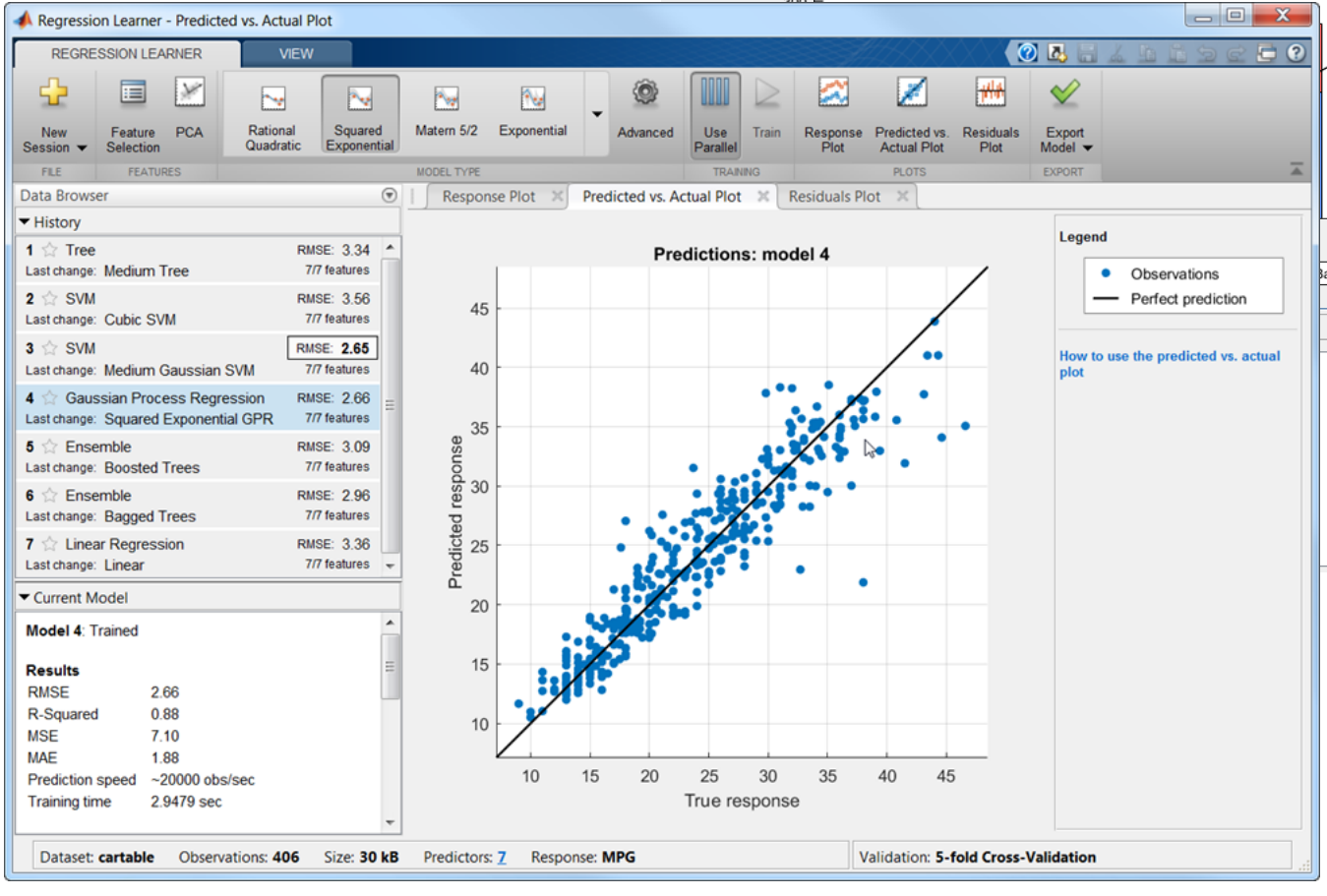
```
delayType   = 'DepDelay';
year        = 1999                ;
topAirports =    7                ;
topCarriers =   10                ;

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

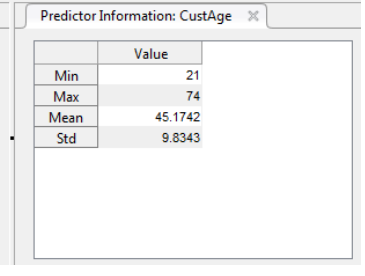




Apps

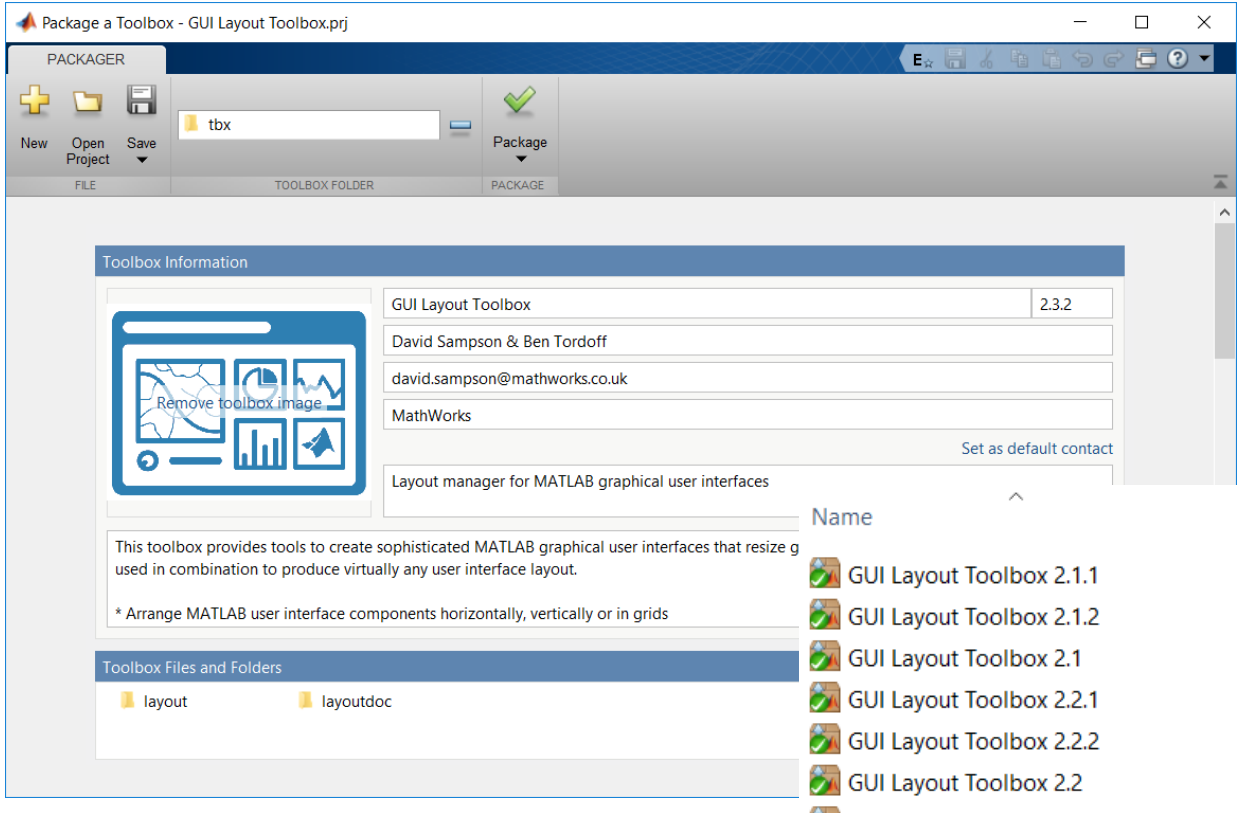


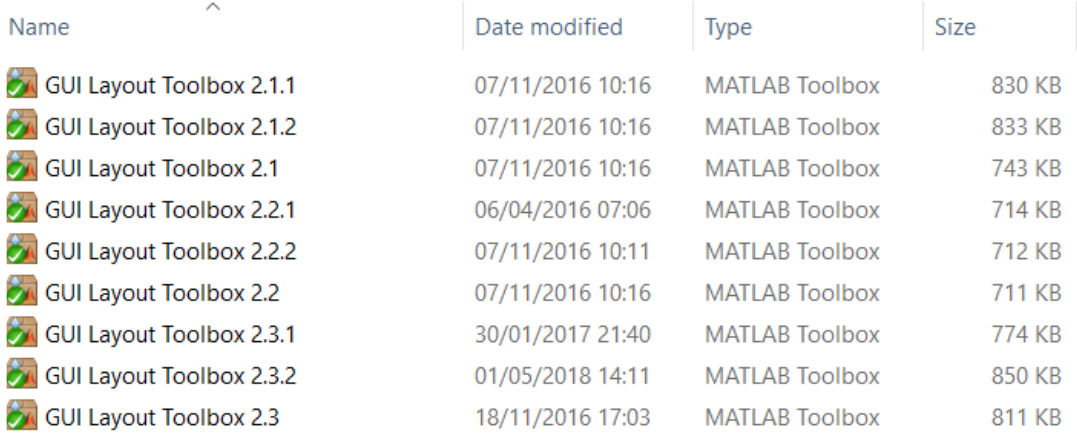




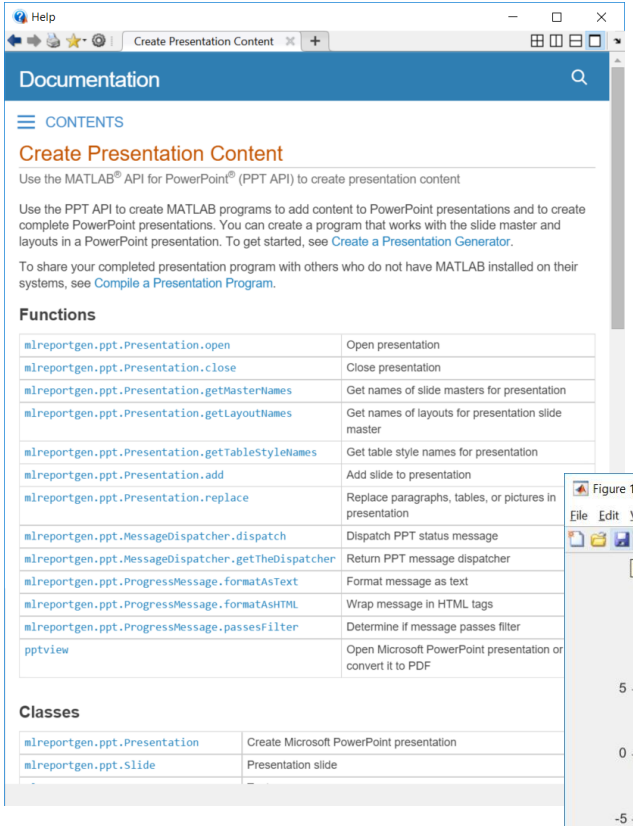
Generate code!

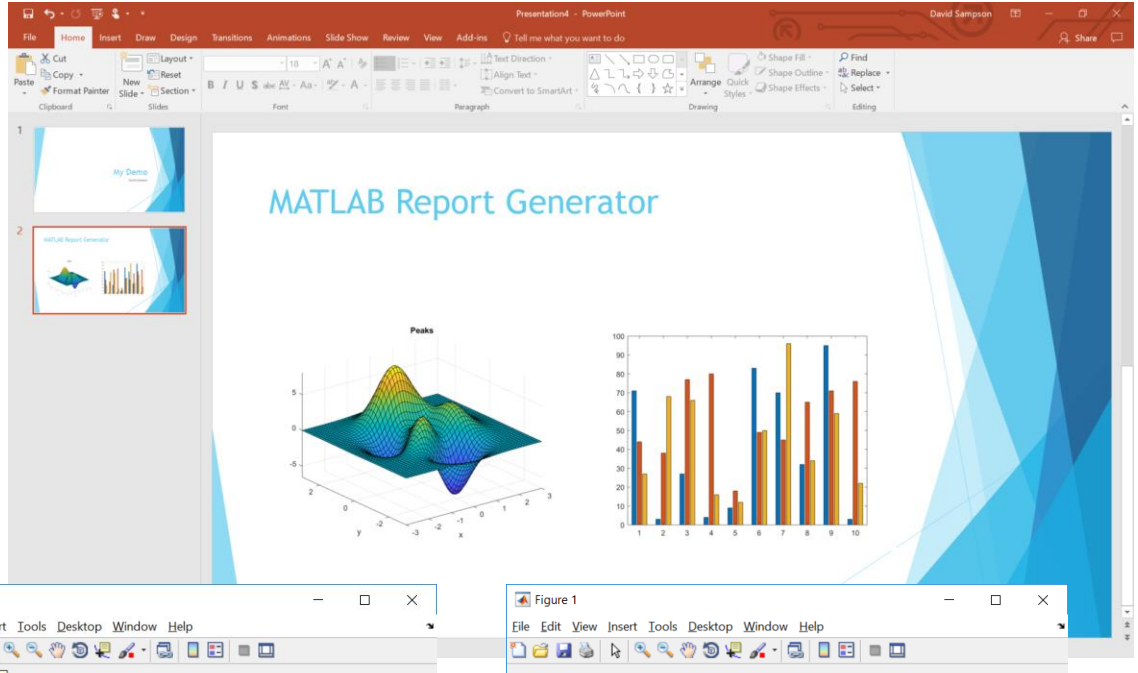
Toolbox Packaging

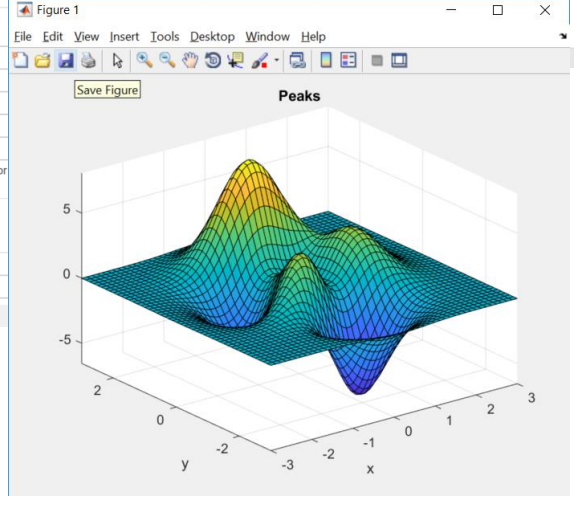


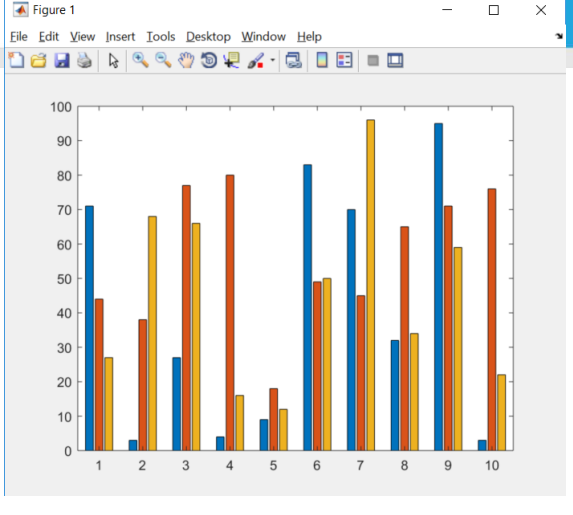


Report Generator

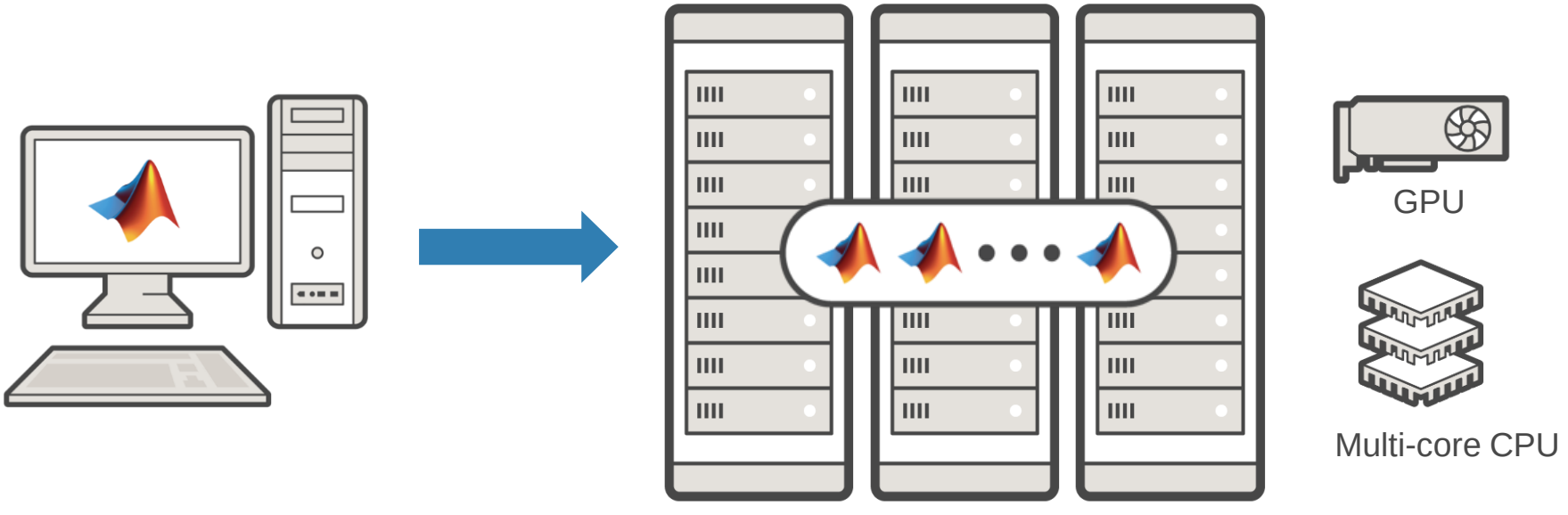






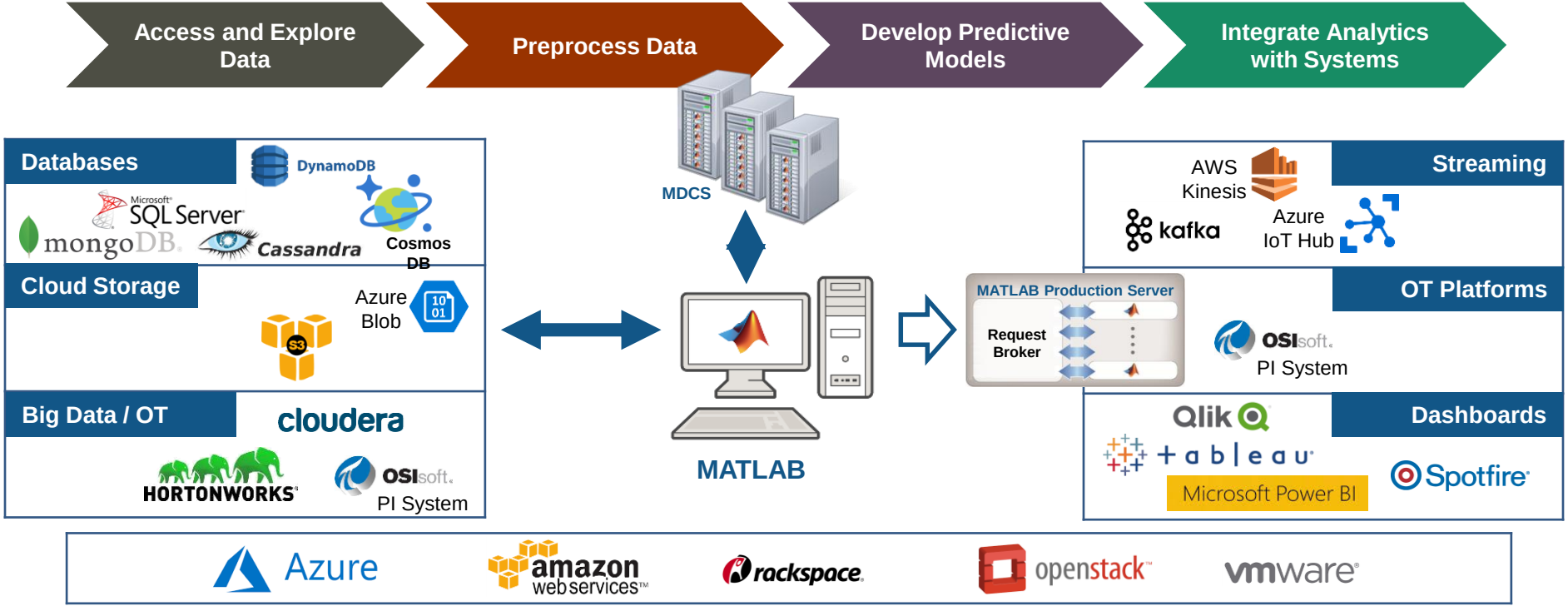


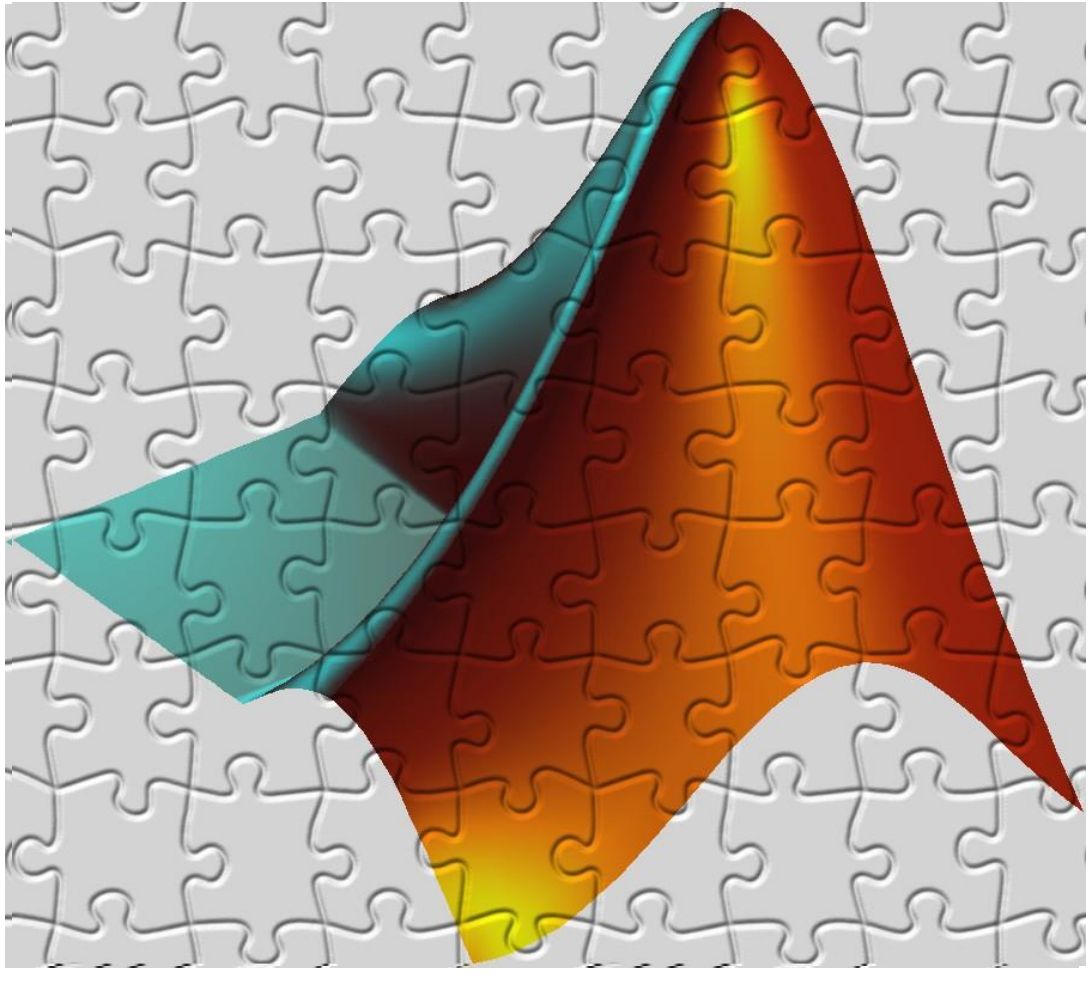
Parallel

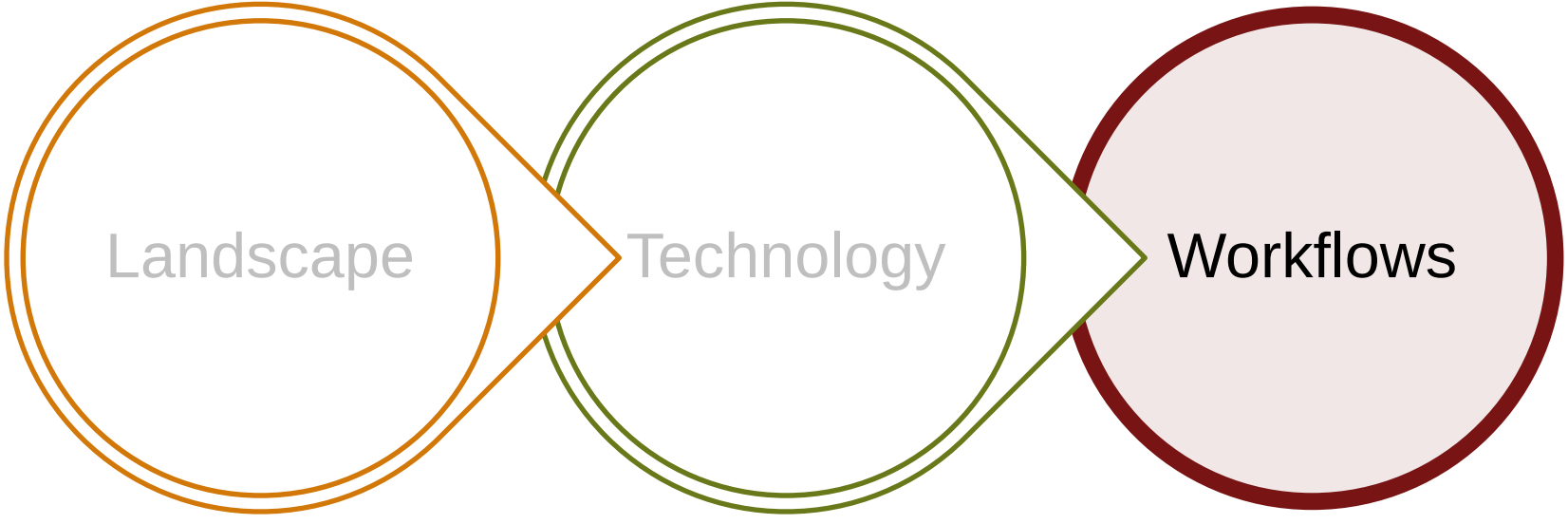


- High-level: `parfor, gpuArray`
- Low-level: `batch, createJob, createTask`
- Big data: `tall, mapreduce`

Enterprise integration







- Requirements
- Challenges

- Environment
- Toolboxes
- Integrations

- How they work
- Why they work

Access and Explore Data

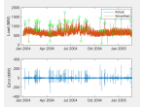
Files

Databases

Sensors

Preprocess Data

Working with Messy Data



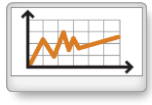
Data Reduction/ Transformation

Feature Extraction

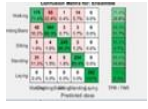
Develop Predictive Models

Model Creation e.g. Machine Learning

Parameter Optimization



Model Validation

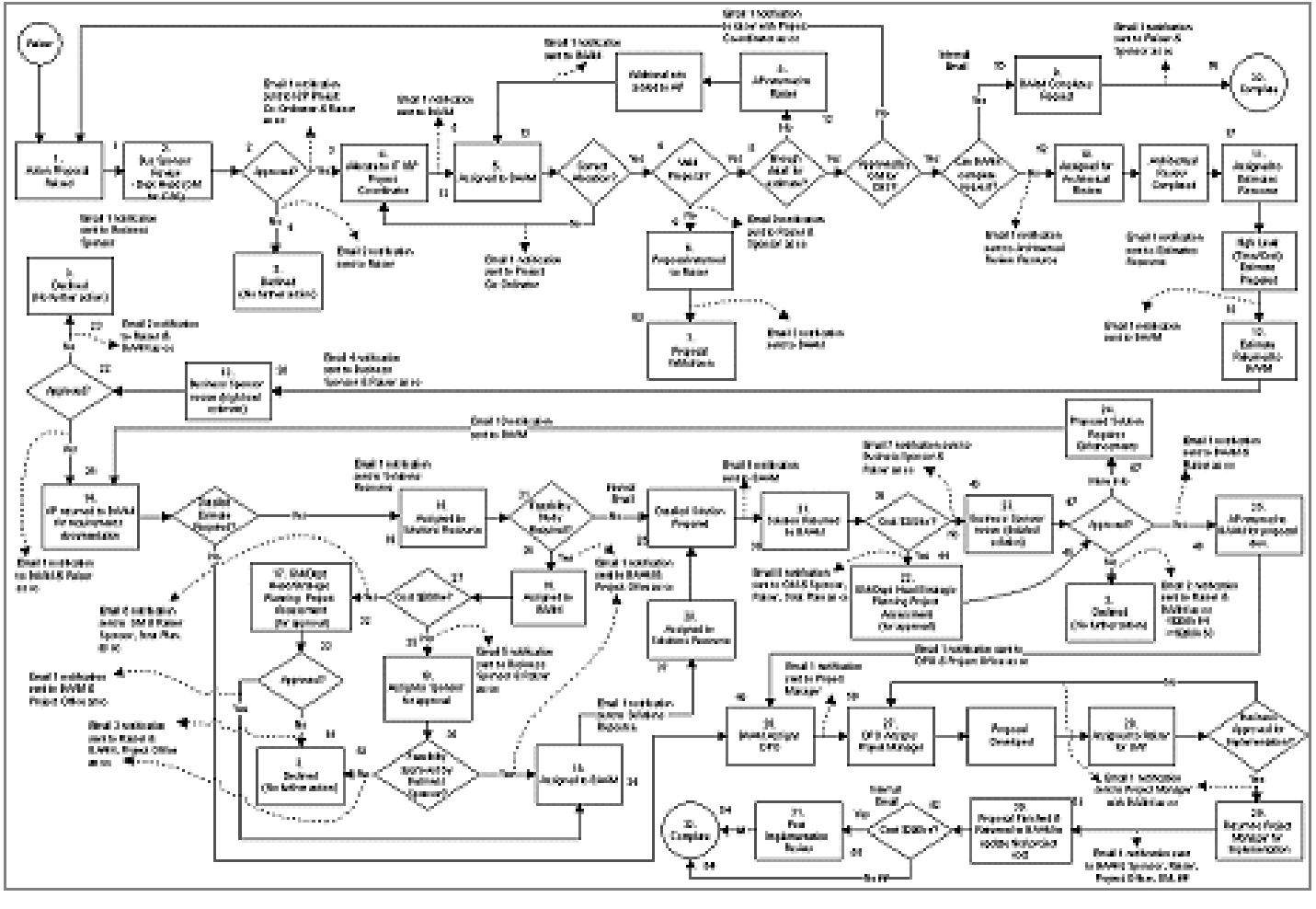


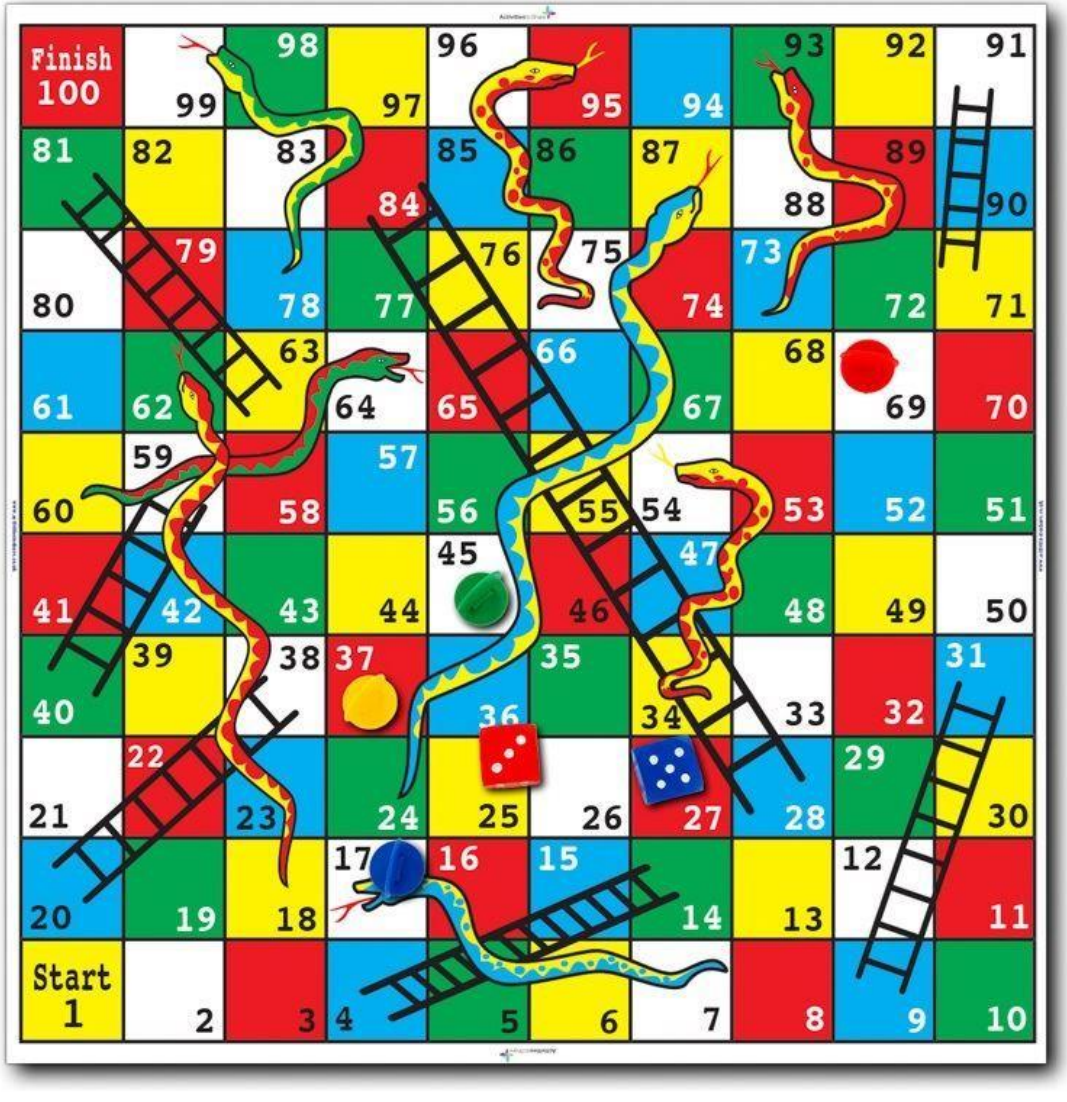
Integrate Analytics with Systems

Desktop Apps

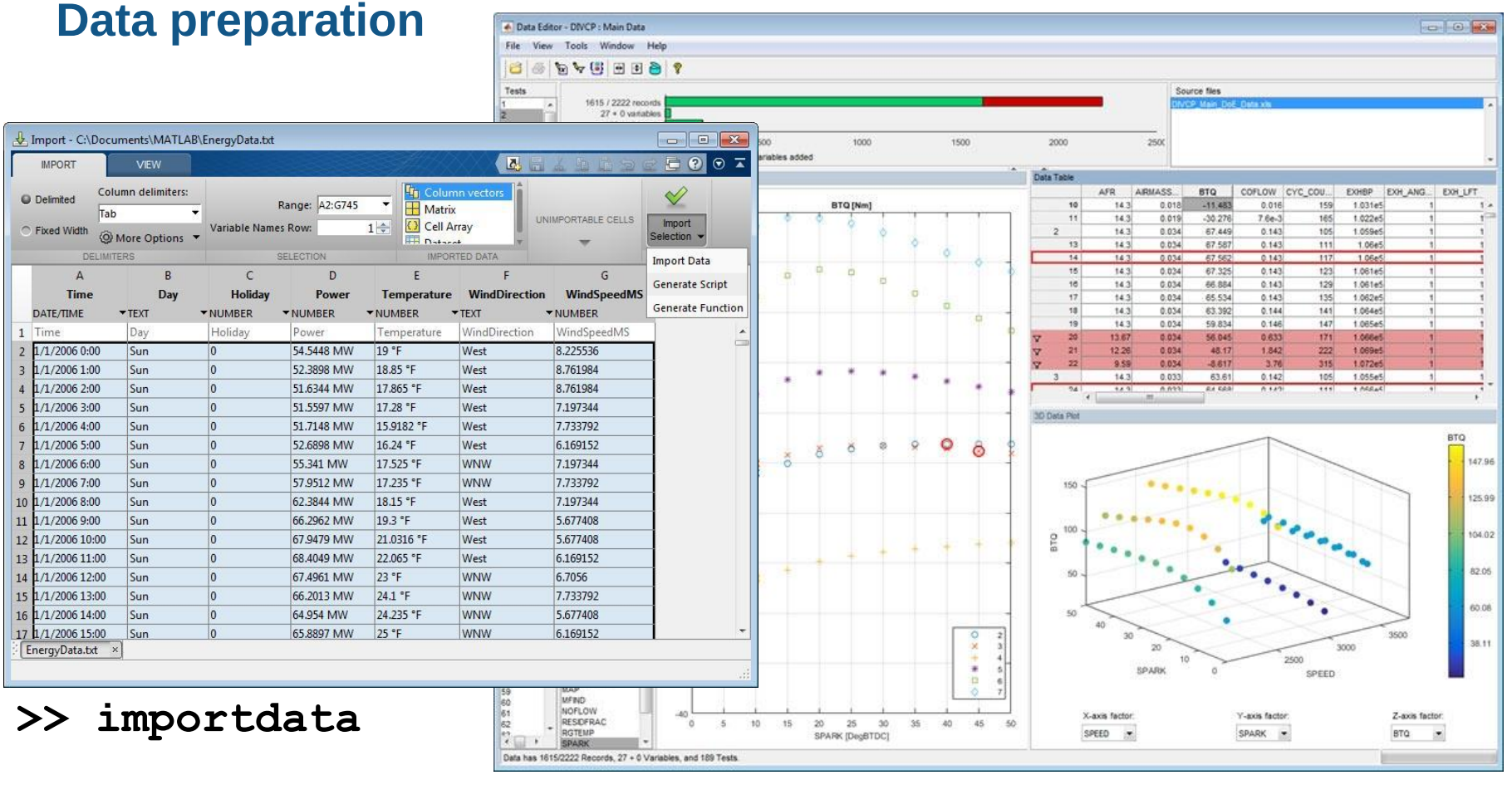
Enterprise Scale Systems

Embedded Devices and Hardware




Finish
100
99
98
97
96
95
94
93
92
91
81
82
83
84
85
86
87
88
89
90
80
79
78
77
76
75
74
73
72
71
61
62
63
64
65
66
67
68
69
70
60
59
58
57
56
55
54
53
52
51
41
42
43
44
45
46
47
48
49
50
40
39
38
37
36
35
34
33
32
31
21
22
23
24
25
26
27
28
29
30
20
19
18
17
16
15
14
13
12
11
Start
1
2
3
4
5
6
7
8
9
10

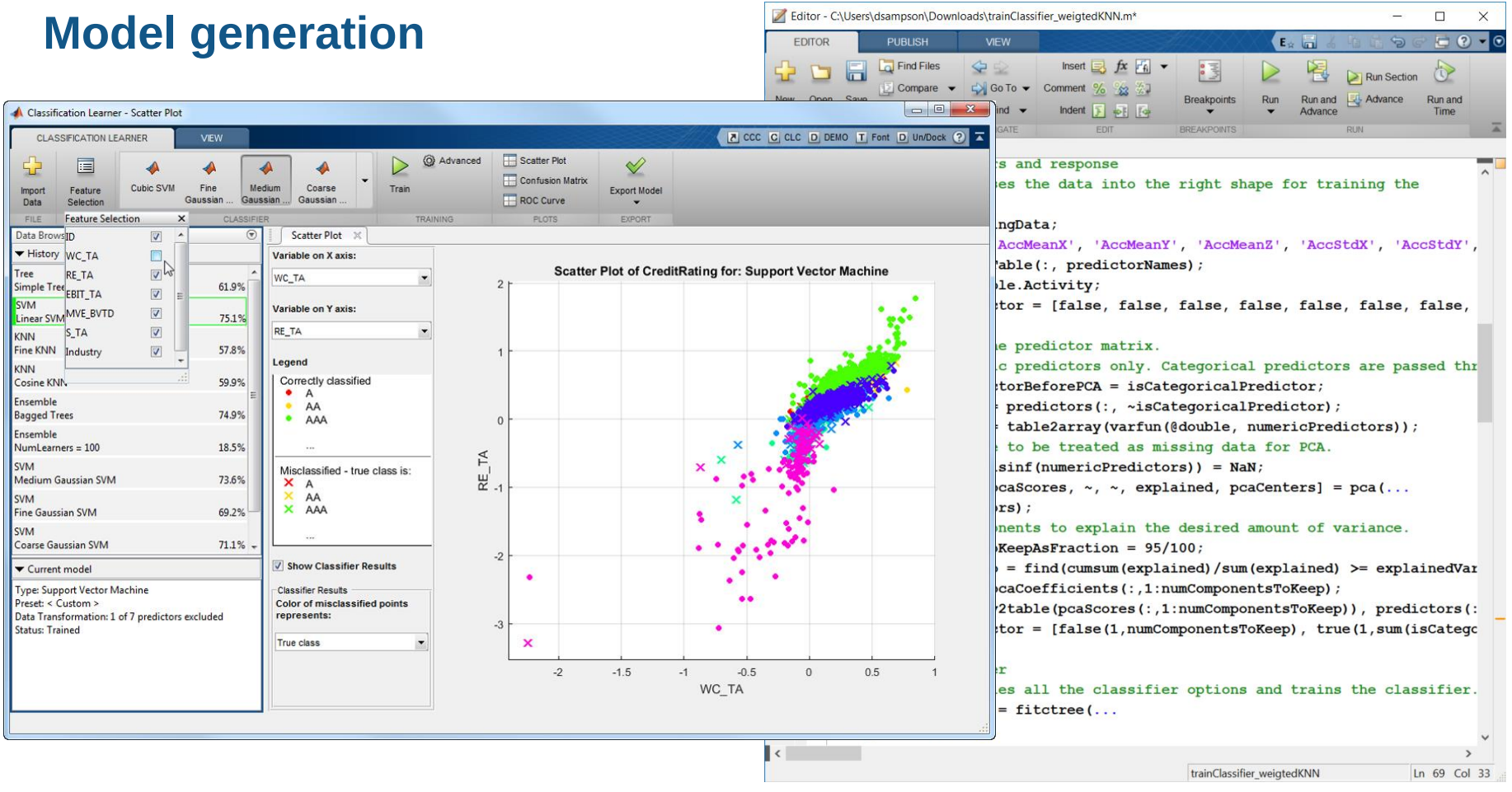
Data preparation




```
>> importdata
```

Assemble a sufficient volume of clean data of known provenance.

Model generation

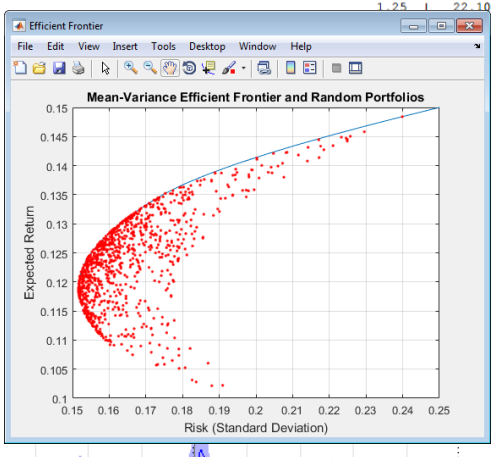


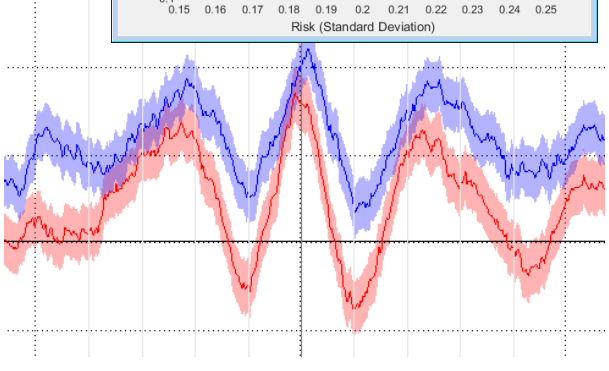
Create models with the required accuracy and reach, while driving insight.

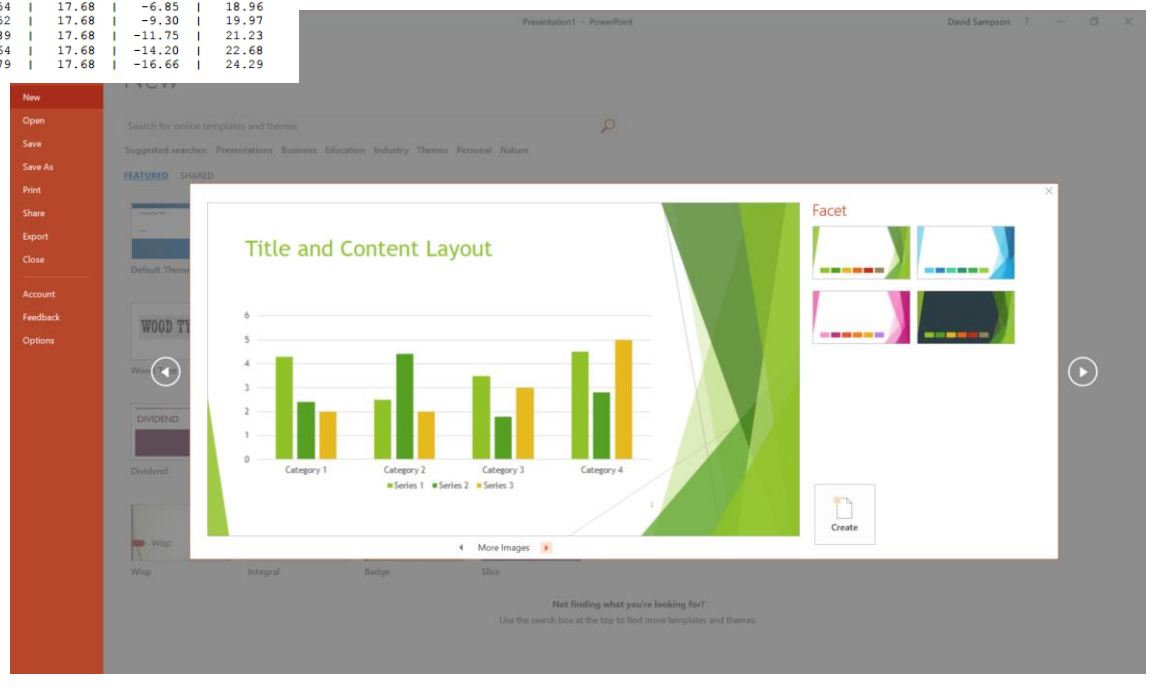
Documentation generation

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

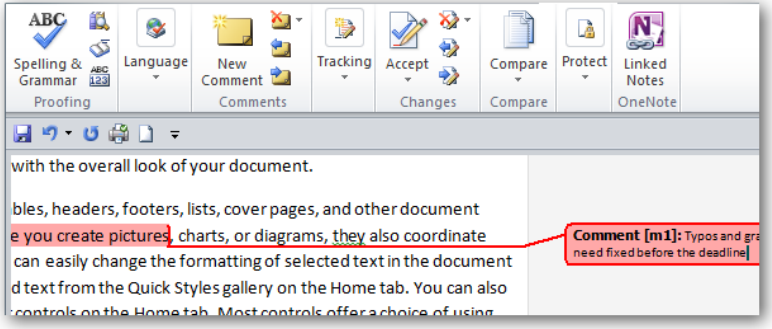






Generate an accurate, insightful description of the study in a timely manner.

Model review



Documents

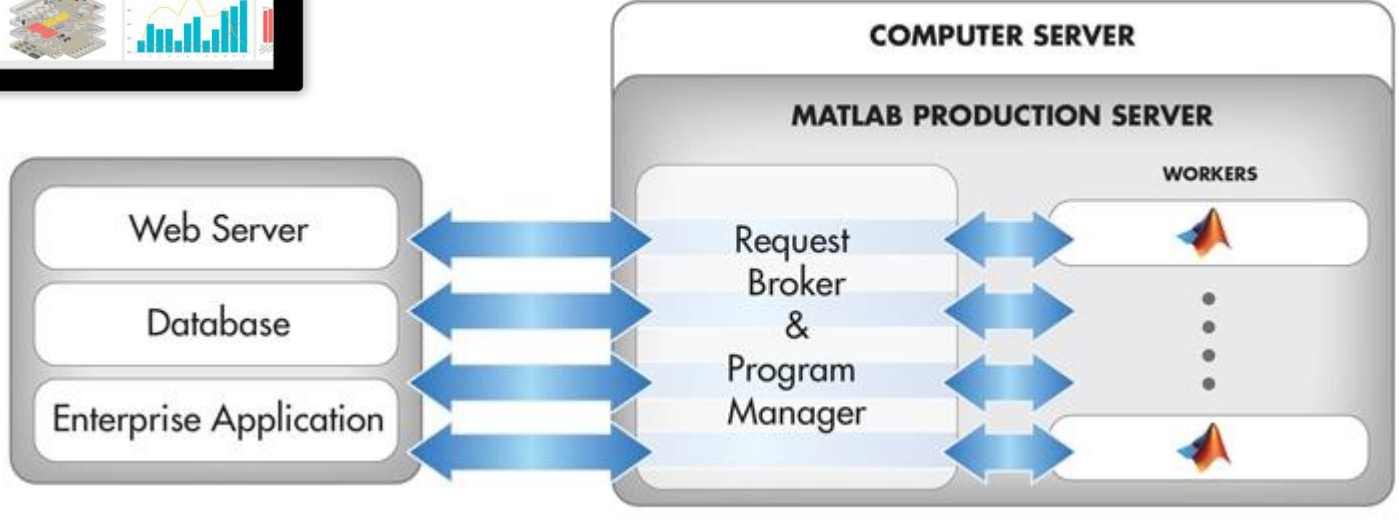
Examples

Models

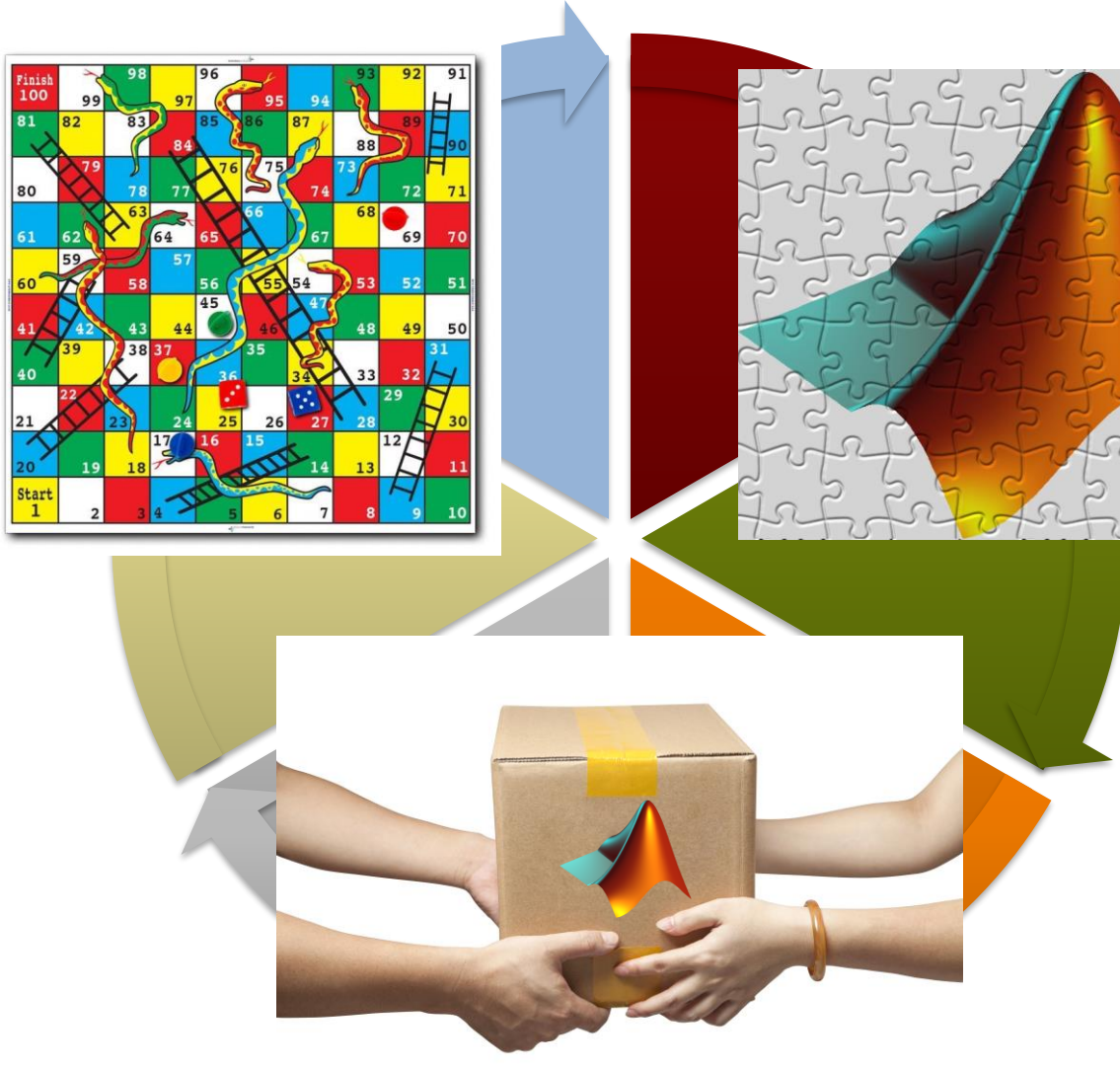
Toolbox code

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.


Finish
100
Start
1