

MATLAB EXPO

Low Code Data Analysis in MATLAB

Onomitra Ghosh, MathWorks



(he/him)

Lola Davidson, MathWorks

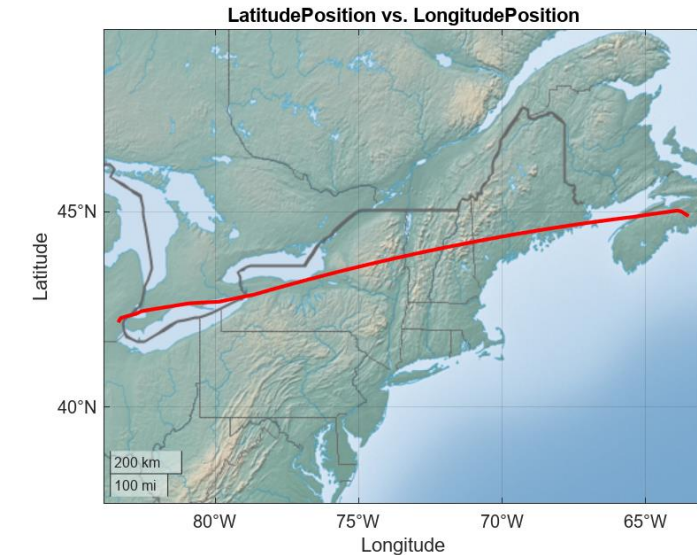


(she/her)



Case Study: Get Flight Sensor Data ready for Modeling

- **Objective:**
 - Explore, analyze and prepare flight sensor data for modeling
- **Inputs:**
 - Excel file with raw flight sensor data
- **Output:**
 - Cleaned sensor data that can be trained to predict Air Speed
 - Reusable code
- **Source:**
 - [NASA Dash Link: Sample Flight Data](#)



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Time	FuelQuantity	Latitude	Longitude	OilPressure	OilTemperature	FlightPhaseFromACMS	WeightOnWheels	Altitude	ExhaustGasTemperature	FuelFlow	TrueAirSpeed	WindDirection
2	6/2/2001 5:41	8048	44.8915135	-63.519183	0	23.67477417	Planning	0	419	17.5	0	0	0
3	6/2/2001 5:41	8048	44.8915135	-63.519183	0	23.67477417	Planning	0	419	17.5	0	0	0
4	6/2/2001 5:41	8048	44.8915135	-63.519183	0	23.67477417	Planning	0	420	17.5	0	0	0
5	6/2/2001 5:41	8048	44.8915135	-63.519183	0	23.67477417	Planning	0	419	17.5	0	0	0
6	6/2/2001 5:41	8048	44.8915135	-63.519183	0	23.67477417	Planning	0	419	17.5	0	0	0
7	6/2/2001 5:41	8048	44.8915135	-63.519183	0	23.67477417	Planning	0	420	17.5	0	0	0
8	6/2/2001 5:41	8048	44.8915135	-63.519183	0	23.67477417	Planning	0	419	17.5	0	0	0
9	6/2/2001 5:41	8048	44.8915135	-63.519183	0	23.67477417	Planning	0	419	17.5	0	0	0
10	6/2/2001 5:41	8048	44.8915135	-63.5189992	0	23.67477417	Planning	0	419	17.5	0	0	0
11	6/2/2001 5:41	8048	44.8915135	-63.5189992	0	23.67477417	Planning	0	418	17.5	0	0	0
12	6/2/2001 5:41	8048	44.8915135	-63.5189992	0	23.67477417	Planning	0	420	17.5	0	0	0
13	6/2/2001 5:41	8048	44.8915135	-63.5189992	0	23.67477417	Planning	0	419	17.5	0	0	0
14	6/2/2001 5:41	8040	44.8915135	-63.5189992	0	23.67477417	Planning	0	419	17	0	0	0
15	6/2/2001 5:41	8040	44.8915135	-63.5189992	0	23.67477417	Planning	0	419	17	0	0	0
16	6/2/2001 5:41	8040	44.8915135	-63.5189992	0	23.67477417	Planning	0	418	17	0	0	0
17	6/2/2001 5:41	8040	44.8915135	-63.5189992	0	23.67477417	Planning	0	419	17	0	0	0
18	6/2/2001 5:41	8032	44.8915135	-63.5189992	0	25.0178833	Planning	0	418	17	0	0	0
19	6/2/2001 5:41	8032	44.8915135	-63.5189992	0	25.0178833	Planning	0	418	17	0	0	0

What are “low code” tools?

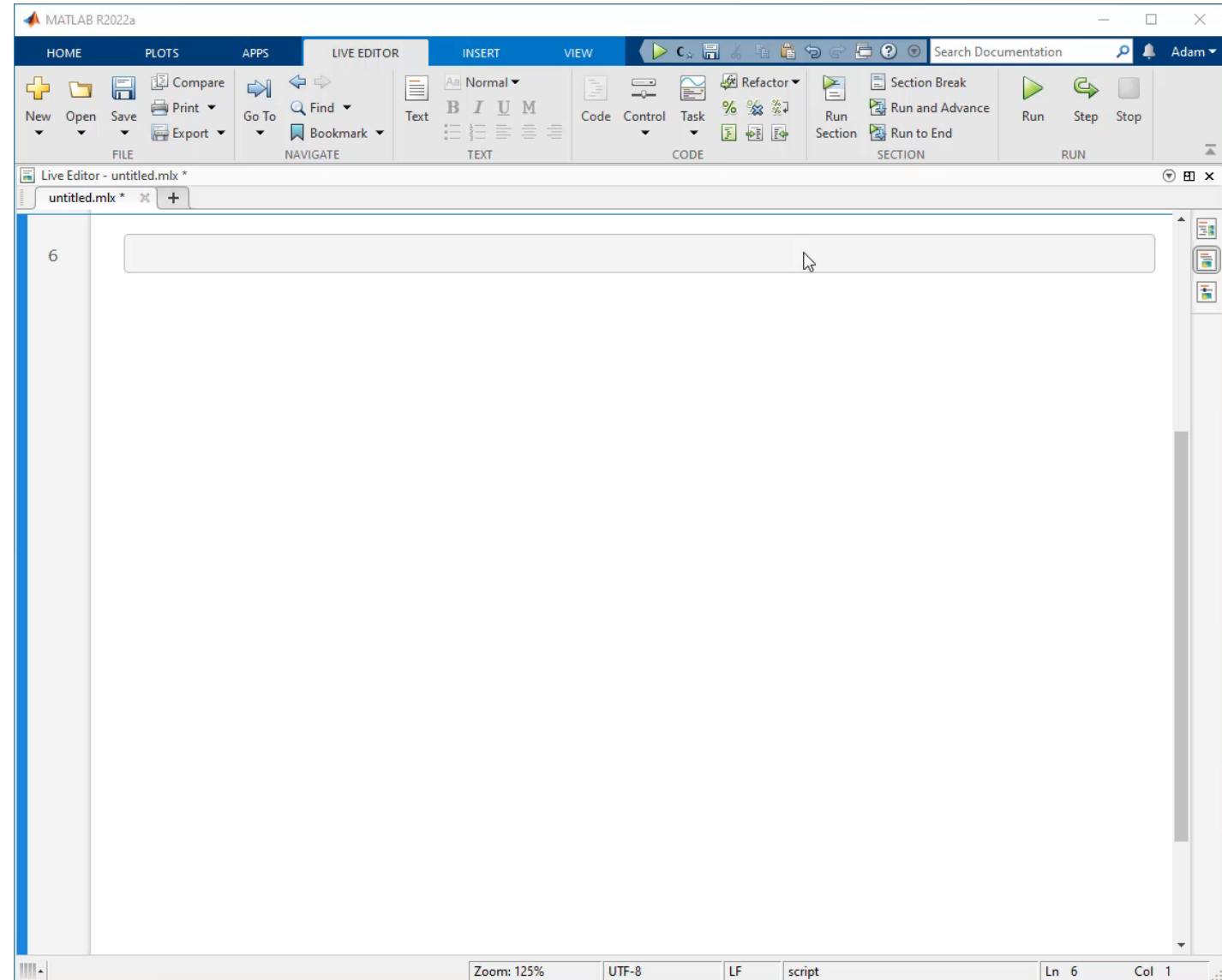
Low code tools enable:

- rapid software development
- minimal manual coding

Benefits of low code tools:

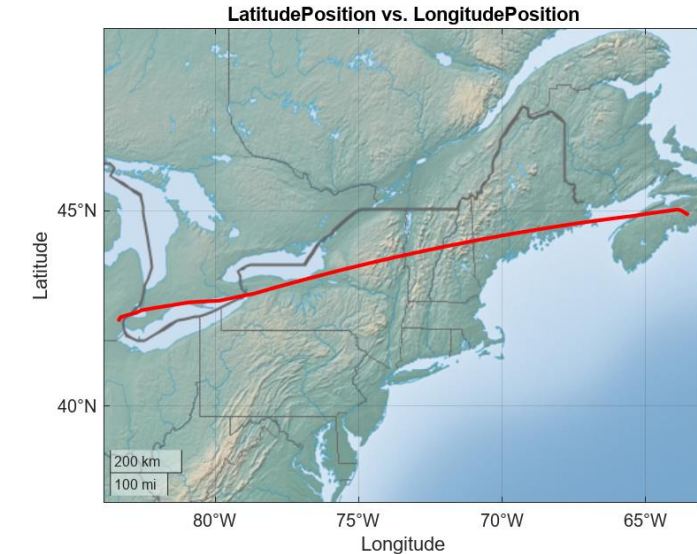
- Shallow learning curve
- Teach *how* to code
- Solve task first, code later

Not just for beginners



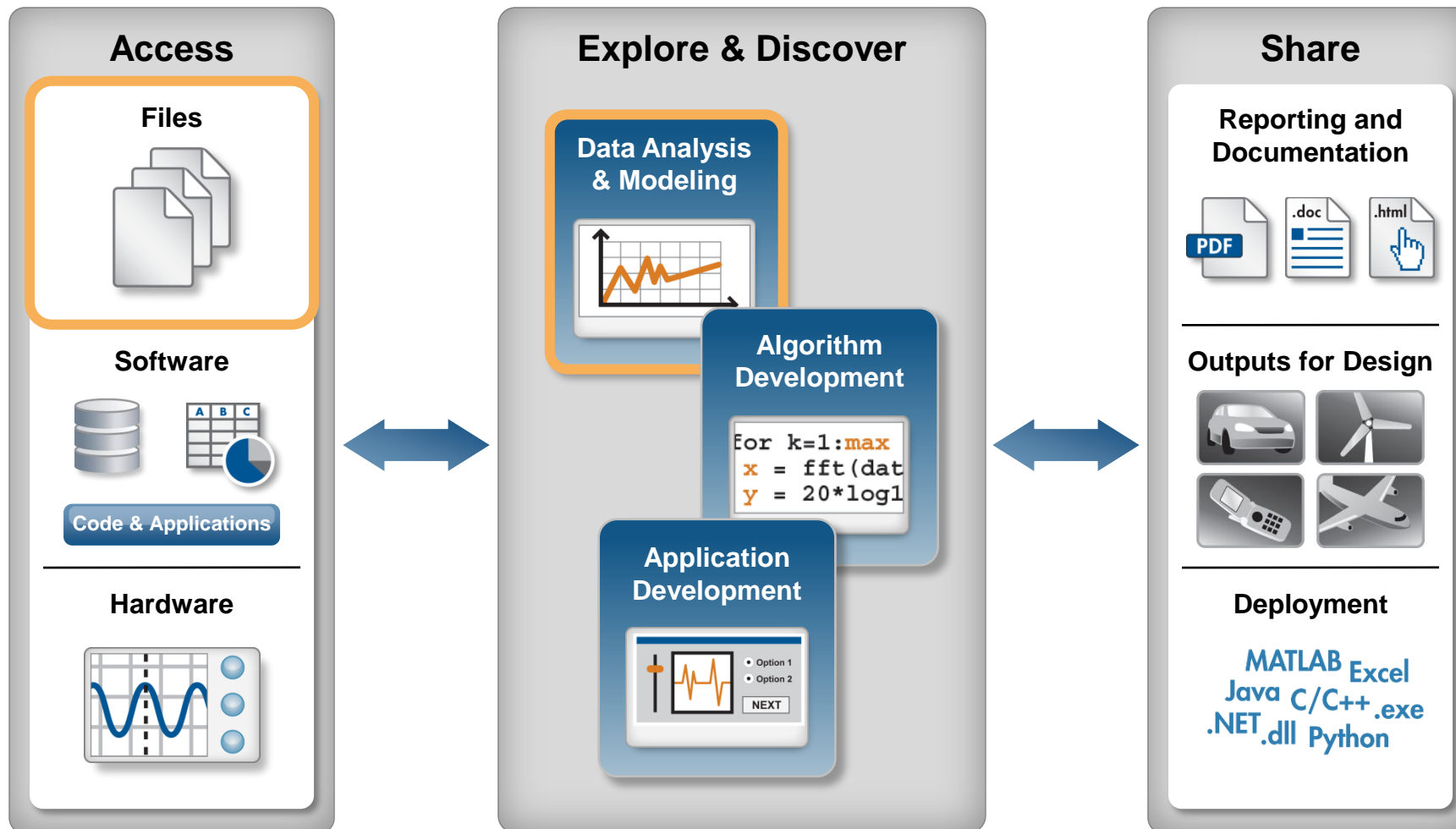
Case Study: Get Flight Sensor Data ready for Modeling

- **Objective:**
 - Explore, analyze and prepare flight sensor data for modeling
- **Inputs:**
 - Excel file with raw flight sensor data
- **Output:**
 - Cleaned sensor data that can be trained to predict Air Speed
 - Reusable code
- **Source:**
 - [NASA Dash Link: Sample Flight Data](#)



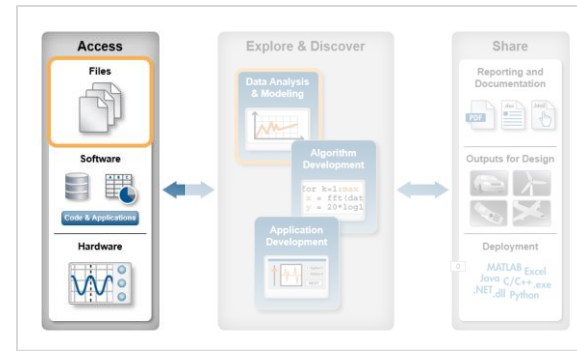
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Time	FuelQuantity	Latitude	Longitude	OilPressure	OilTemperature	FlightPhaseFromACMS	WeightOnWheels	Altitude	ExhaustGasTemperature	FuelFlow	TrueAirSpeed	WindDirection
2	6/2/2001 5:41	8048	44.8915135	-63.519183	0	23.67477417	Planning	0	419	17.5	0	0	0
3	6/2/2001 5:41	8048	44.8915135	-63.519183	0	23.67477417	Planning	0	419	17.5	0	0	0
4	6/2/2001 5:41	8048	44.8915135	-63.519183	0	23.67477417	Planning	0	420	17.5	0	0	0
5	6/2/2001 5:41	8048	44.8915135	-63.519183	0	23.67477417	Planning	0	419	17.5	0	0	0
6	6/2/2001 5:41	8048	44.8915135	-63.519183	0	23.67477417	Planning	0	419	17.5	0	0	0
7	6/2/2001 5:41	8048	44.8915135	-63.519183	0	23.67477417	Planning	0	420	17.5	0	0	0
8	6/2/2001 5:41	8048	44.8915135	-63.519183	0	23.67477417	Planning	0	419	17.5	0	0	0
9	6/2/2001 5:41	8048	44.8915135	-63.519183	0	23.67477417	Planning	0	419	17.5	0	0	0
10	6/2/2001 5:41	8048	44.8915135	-63.5189992	0	23.67477417	Planning	0	419	17.5	0	0	0
11	6/2/2001 5:41	8048	44.8915135	-63.5189992	0	23.67477417	Planning	0	418	17.5	0	0	0
12	6/2/2001 5:41	8048	44.8915135	-63.5189992	0	23.67477417	Planning	0	420	17.5	0	0	0
13	6/2/2001 5:41	8048	44.8915135	-63.5189992	0	23.67477417	Planning	0	419	17.5	0	0	0
14	6/2/2001 5:41	8040	44.8915135	-63.5189992	0	23.67477417	Planning	0	419	17	0	0	0
15	6/2/2001 5:41	8040	44.8915135	-63.5189992	0	23.67477417	Planning	0	419	17	0	0	0
16	6/2/2001 5:41	8040	44.8915135	-63.5189992	0	23.67477417	Planning	0	418	17	0	0	0
17	6/2/2001 5:41	8040	44.8915135	-63.5189992	0	23.67477417	Planning	0	419	17	0	0	0
18	6/2/2001 5:41	8032	44.8915135	-63.5189992	0	25.0178833	Planning	0	418	17	0	0	0
19	6/2/2001 5:41	8032	44.8915135	-63.5189992	0	25.0178833	Planning	0	418	17	0	0	0

MATLAB simplifies the data analysis workflow with low code tools



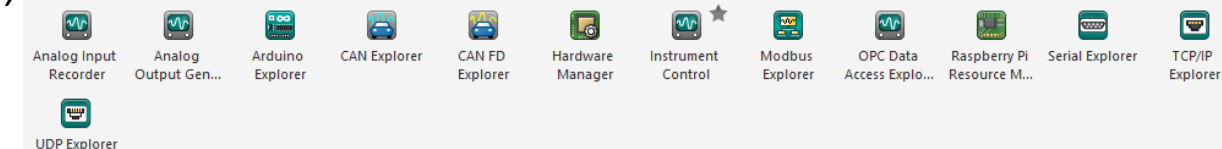
Use low code tools for easy access to files, databases, and hardware

- Import Tool and Import Live Task
 - Text, CSV, and Excel files
- Database Explorer (*Database Toolbox*)
 - ODBC & JDBC SQL Databases
- Measurement hardware and industrial data
 - Data acquisition hardware (*Data Acquisition Toolbox*)
 - Stand-alone instruments and hardware (*Instrument Control Toolbox*)
 - OPC UA and Aveva PI Server, Modbus devices (*Industrial Communication Toolbox*)
 - CAN, J1939, and XCP (*Vehicle Network Toolbox*)



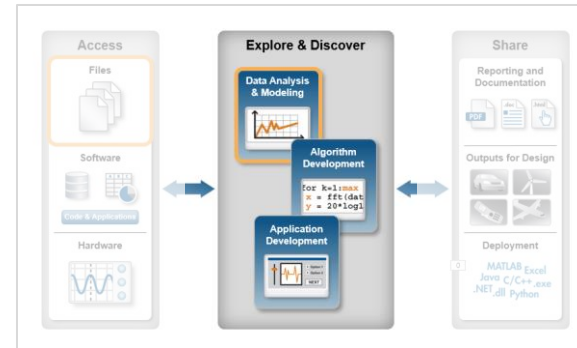
The screenshot shows the MATLAB Database Explorer interface. The main window displays a table of flight data with columns: Time, FuelQuantity, OilPressure, OilTemperature, Latitude, Longitude, Altitude, ExhaustTemperature, FuelFlow, FuelSpeed, TrueAirSpeed, WindDirection, WindSpeed, and Weight. The data is organized into rows, with the first row being the header. A SQL query window is open, showing a query that selects inventorytable.productnumber, inventorytable.quantity, inventorytable.price, and inventorytable.inventorydate from the inventorytable, joined with the producttable on inventorytable.productnumber = producttable.productnumber. The query results are displayed in a table with columns: productnumber, quantity, price, and inventorydate. The results show 10 rows of data.

TEST AND MEASUREMENT



Over 100 low code tools for data analysis, engineering, and AI

- Data Analysis
 - Visualize, manipulate, and preprocess
 - Math, statistics, and optimization

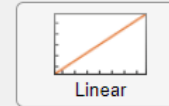


Optimize

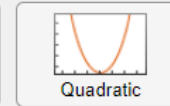
Minimize a function with or without constraints

Specify problem type

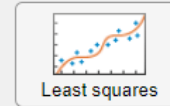
Objective



Linear



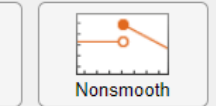
Quadratic



Least squares

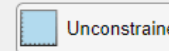


Nonlinear

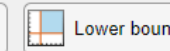


Nonsmooth

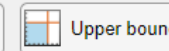
Select an objective type to see example functions



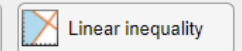
Unconstrained



Lower bounds

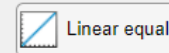


Upper bounds

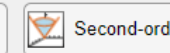


Linear inequality

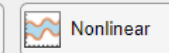
Constraints



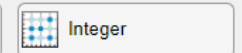
Linear equality



Second-order cone



Nonlinear



Integer

Select constraint types to see example formulas

Solver

fminsearch - Unconstrained derivative-free nonlinear minimization (recommended)



Select problem data

Objective function

From file

Browse...

New...



Initial point (x0)

select

Specify solver options

Display progress

Text display

Final output

Plot

☐ Current point

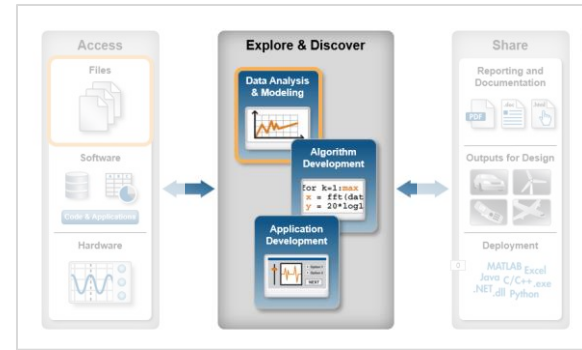
☐ Evaluation count

☐ Objective value

Optimize Live Task (Optimization Toolbox)

Over 100 low code tools for data analysis, engineering, and AI

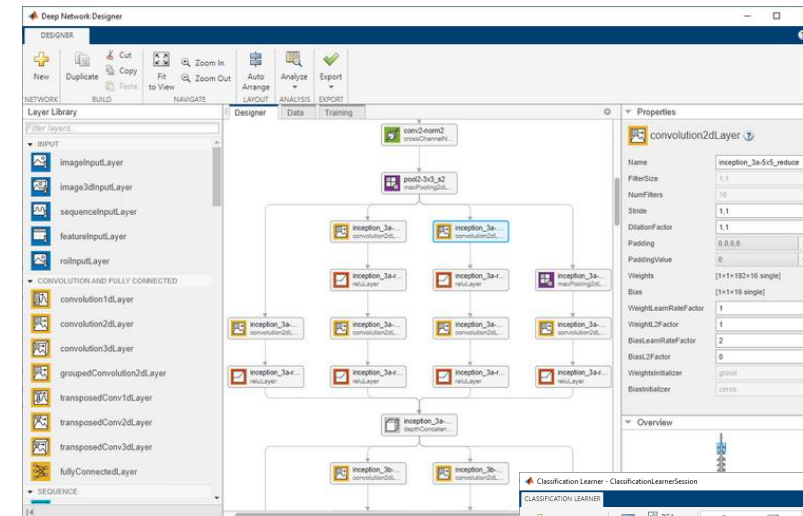
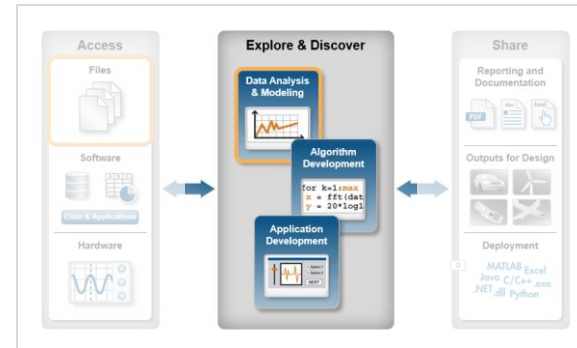
- Data Analysis
 - Visualize, manipulate, and preprocess
 - Math, statistics, and optimization
- Engineering
 - Control system design and analysis
 - Signal processing and communications
 - Image processing and computer vision



Signal Analyzer (Signal Processing Toolbox)

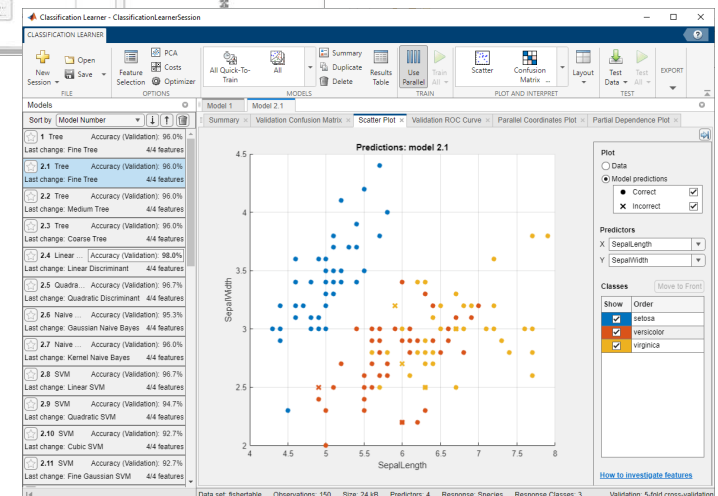
Over 100 low code tools for data analysis, engineering, and AI

- Data Analysis
 - Visualize, manipulate, and preprocess
 - Math, statistics, and optimization
- Engineering
 - Control system design and analysis
 - Signal processing and communications
 - Image processing and computer vision
- Artificial Intelligence
 - Ground truth labeling
 - Network design, training, and validation
 - Quantization and deployment



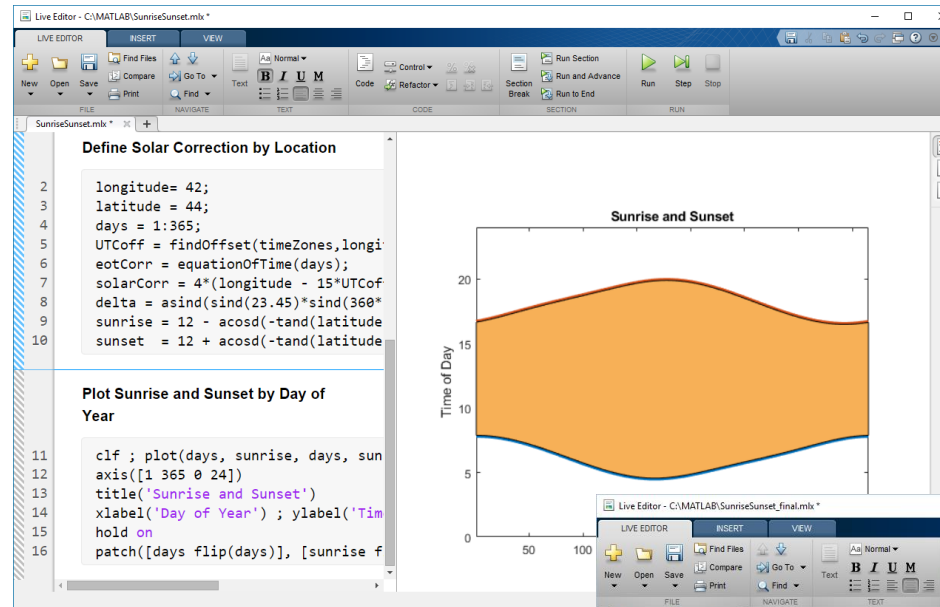
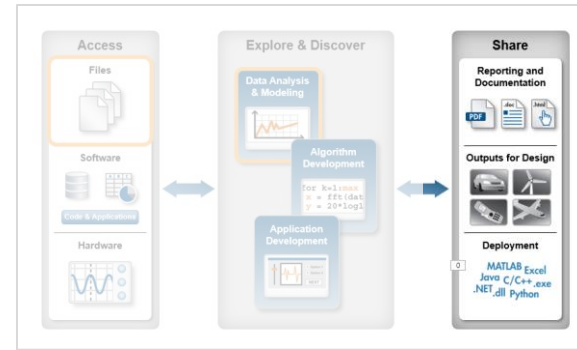
Deep Network Designer
(Deep Learning Toolbox)

Classification Learner
(Statistics and Machine Learning Toolbox)



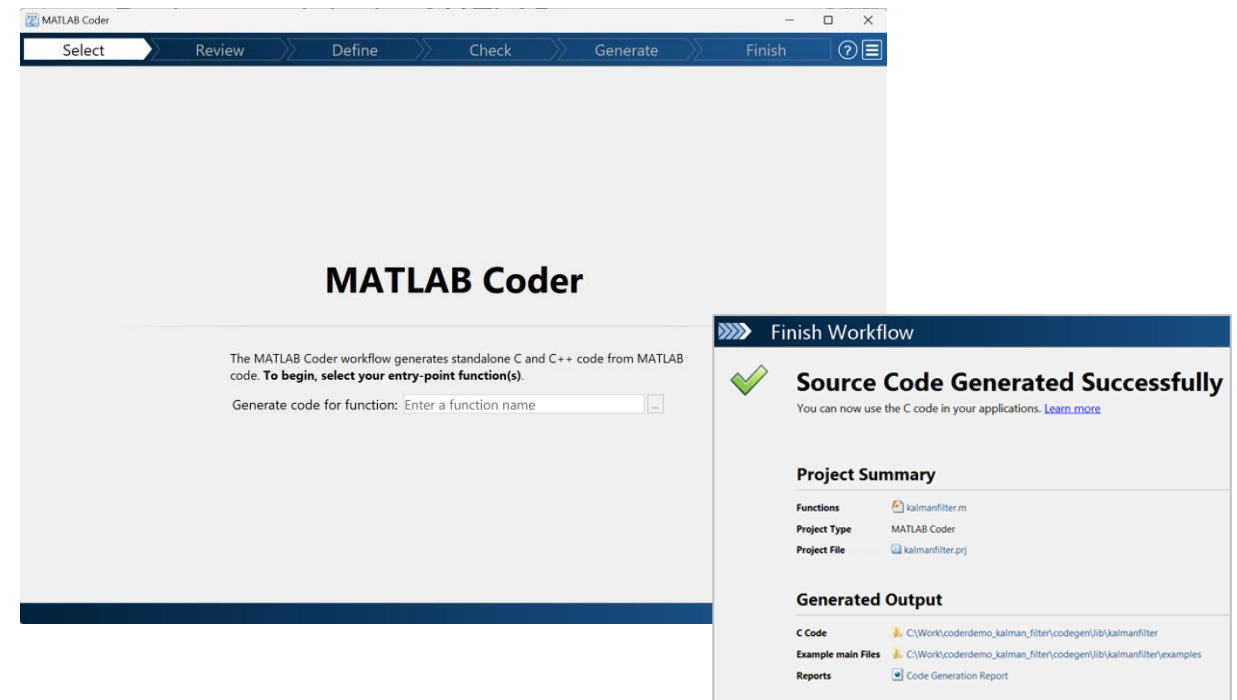
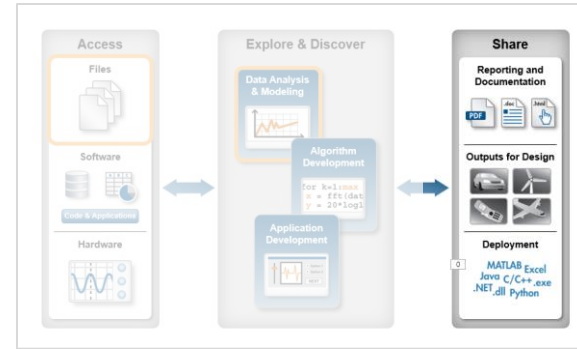
Document as you go – *your script is your report*

- Divide code into sections
- Embed outputs next to the code
- Add rich text formatting, equations, images, and hyperlinks
- Save directly to PDF, HTML, Word, and LaTeX

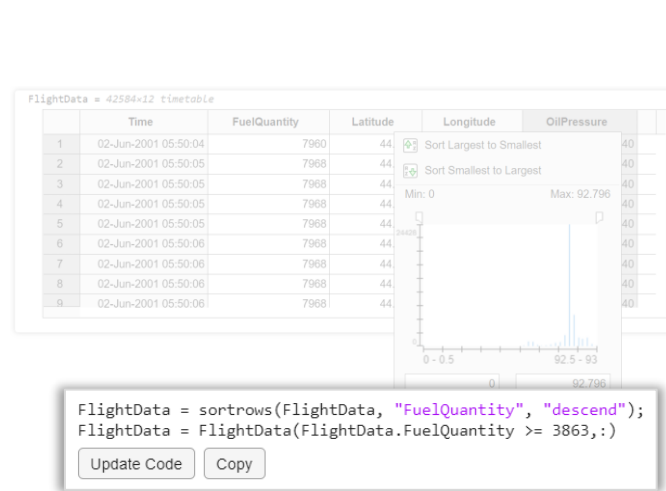


Deploy and integrate MATLAB code

- Package and deploy MATLAB programs
- Generate code (C, Mex, GPU, HDL)



Start with low code ... and switch to code easily when needed



County General Ho...

County General Hospital

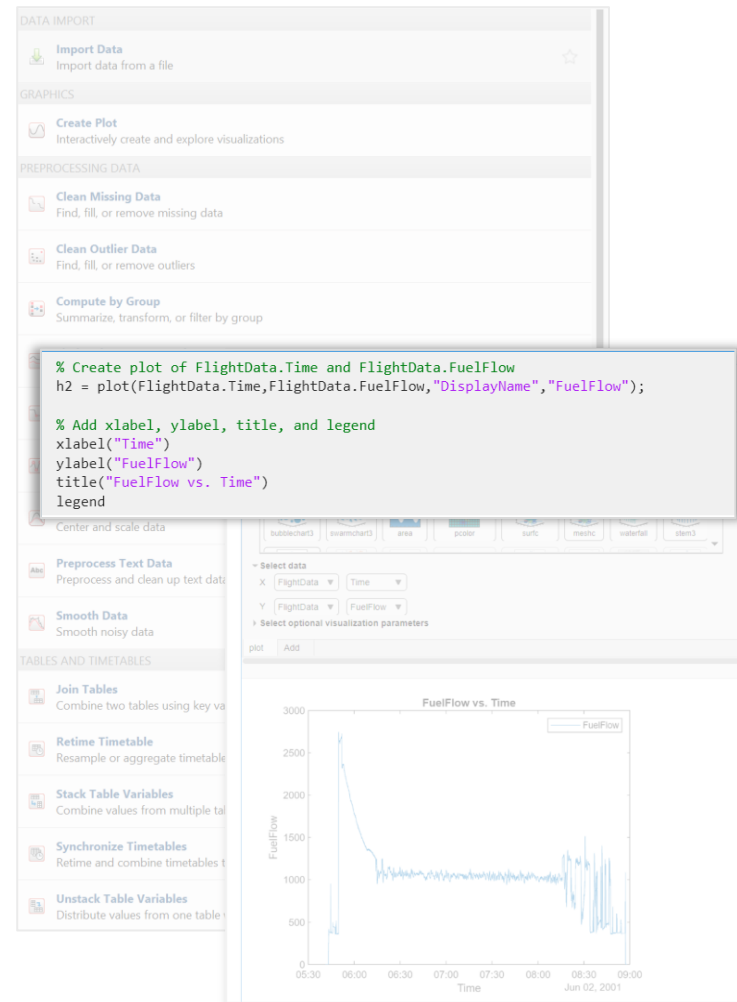
VA Hospital

St. Mary's Medical Center

filename = C:\Work\data1.mat

50 -100 100

2



```
function [trainedClassifier, validationAccuracy] = trainClassifier(trainingData, responseData)
% [trainedClassifier, validationAccuracy] = trainClassifier(trainingData,
% responseData)
% Returns a trained classifier and its accuracy. This code recreates the
% classification model trained in Classification Learner app. Use the
% generated code to automate training the same model with new data, or to
% learn how to programmatically train models.
%
% Input:
%   trainingData: A matrix with the same number of columns and data type
%   as the matrix imported into the app.
%   responseData: A vector with the same data type as the vector
%   imported into the app. The length of responseData and the number of
%   rows of trainingData must be equal.
%
% Output:
%   trainedClassifier: A struct containing the trained classifier. The
%   struct contains various fields with information about the trained
%   classifier.
%   trainedClassifier.predictFcn: A function to make predictions on new
%   data.
%   validationAccuracy: A double representing the validation accuracy as
%   a percentage. In the app, the Models pane displays the validation
%   accuracy for each model.
%
% Auto-generated by MATLAB

% Extract predictors and response
inputTable = array2table(trainingData, 'VariableNames', {'column_1', 'column_2', 'column_3', 'column_4', 'column_5'});
predictorNames = {'column_1', 'column_2', 'column_3', 'column_4', 'column_5'};
predictors = inputTable(:, predictorNames);
response = responseData;
isCategoricalPredictor = [false, false, false, false, false];
classNames = {'Female', 'Male'};

% Train a classifier
% This code specifies all the classifier options and trains the classifier.
classificationTree = fitctree(...
    predictors, ...
    response, ...
    'SplitCriterion', 'gdi', ...
    'MaxNumSplits', 100, ...
    'Surrogate', 'off', ...
    'ClassNames', classNames);

% Create the result struct with predict function
predictorExtensionFcn = @(x) array2table(x, {'column_1', 'column_2', 'column_3', 'column_4', 'column_5'});
```

Interactive Controls

Live Tasks

Apps

MATLAB EXPO

Thank you



© 2023 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.



MathWorks ✓

@MathWorks

Share the EXPO experience
#MATLABEXPO

Onomitra Ghosh



oghosh@mathworks.com



@linkedin.com/in/onomitra



Lola Davidson



ldavidso@mathworks.com



@linkedin.com/in/lola-davidson-
07b073146

