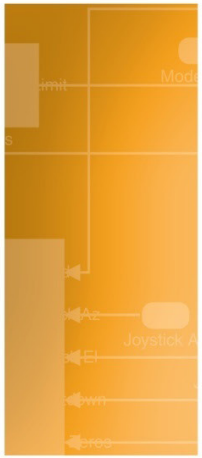
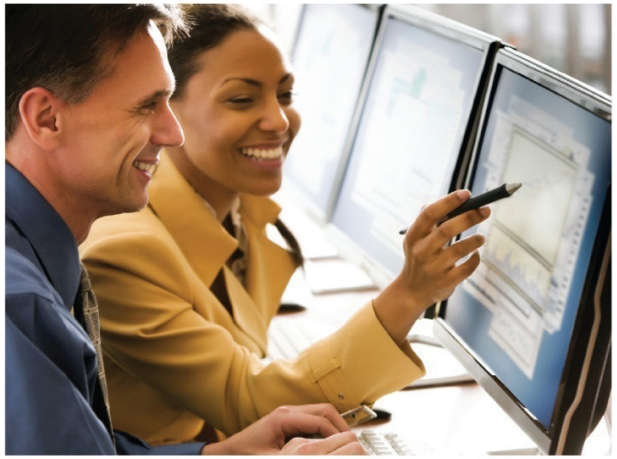
Increasing Automation with Functions

MATLAB® Fundamentals for Aerospace Applications

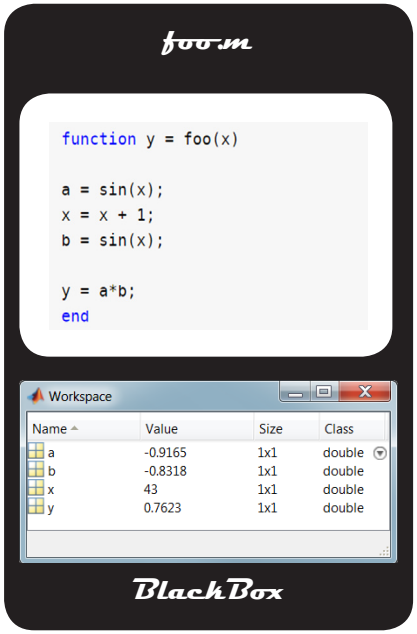


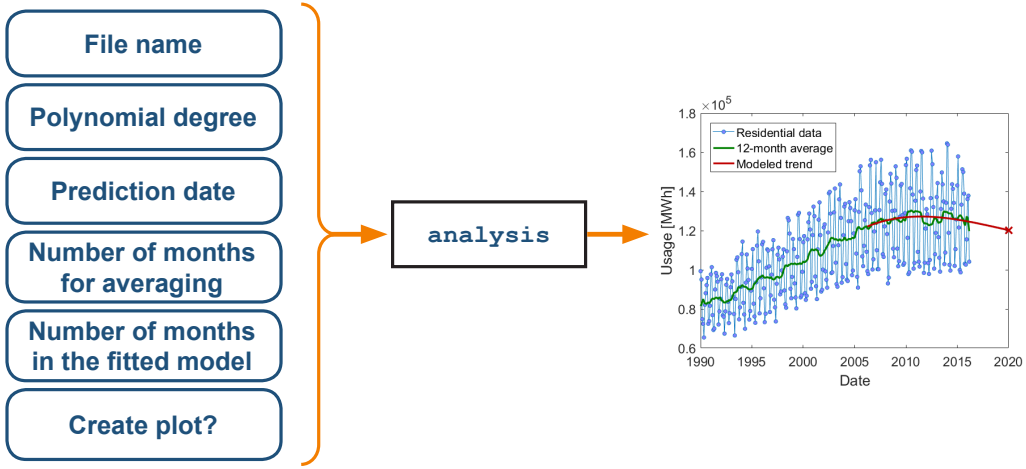


MathWorks | Training Services

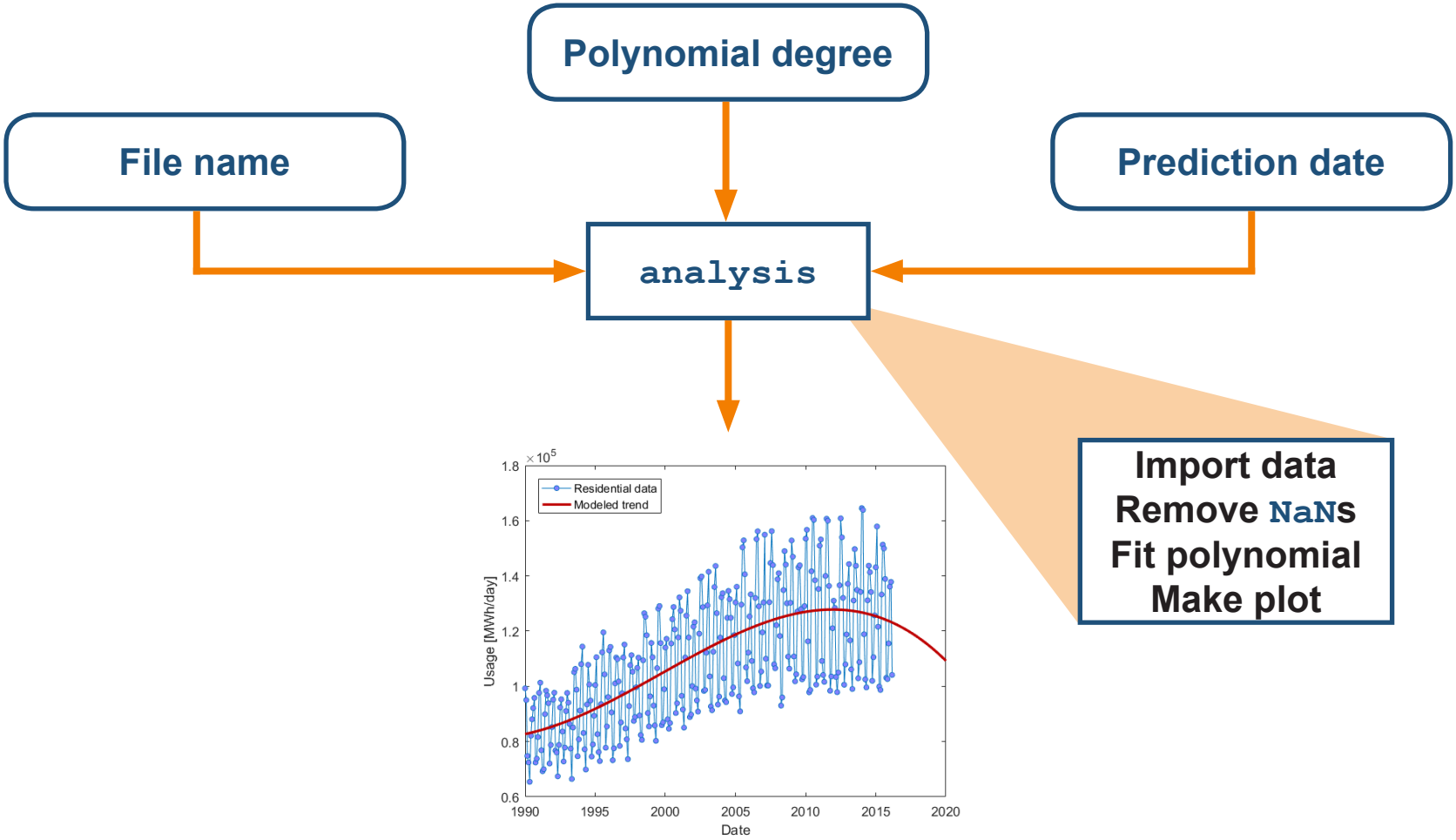
Outline

- Creating and calling functions
- Workspaces
- Plain text code files
- Path and precedence
- Debugging
- Using structures

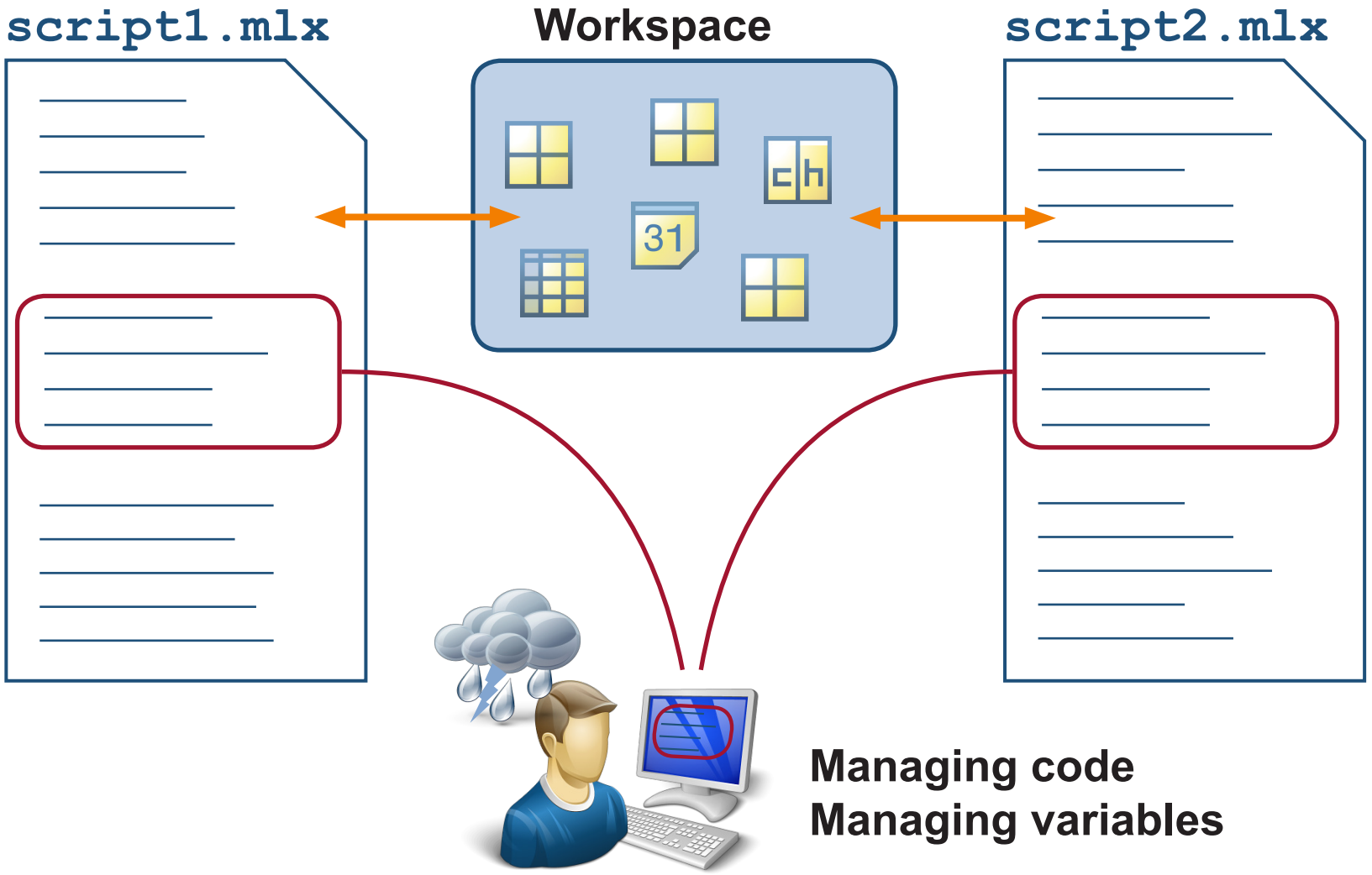




Course Example: Electricity Modeling



Why Use Functions?



Creating a Function

```
datafile = 'elec_res.xlsx';
polydegree = 3;
predictdate = datetime(2020,1,1);
...
data = readtable(datafile);
...
c = polyfit(elapsedYears,usage,polydegree);
...
endDuration = years(predictdate - dates(1));
...
predictedusage = usageFit(end);
```

Function declaration:

```
function outputs = funcname(inputs)
```

```
function predictedusage = analysis_func(datafile,polydegree,predictdate)
...
data = readtable(datafile);
...
c = polyfit(elapsedYears,usage,polydegree);
...
endDuration = years(predictdate - dates(1));
...
predictedusage = usageFit(end);
```

Calling a Function

```
enddate = datetime(2018,9,1);
com2018 = polyPrediction('elec_com.xlsx',2,enddate)
```

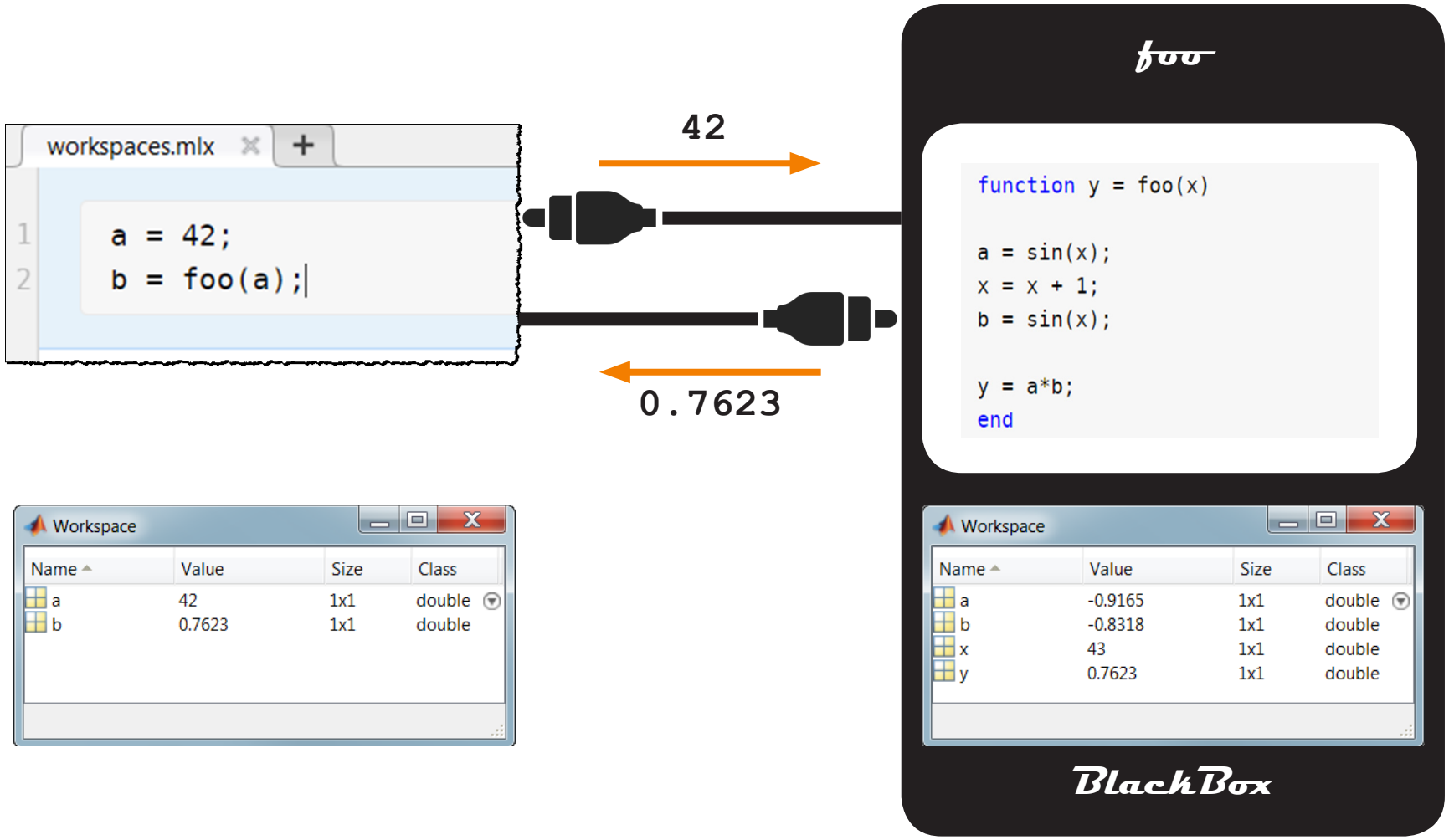
1.2598e+05

```
function predictedusage = polyPrediction(datafile,polydegree,predictdate)
...
data = readtable(datafile);
...
c = polyfit(elapsedYears,usage,polydegree);
...
endDuration = years(predictdate - dates(1));
...
predictedusage = usageFit(end);
```

'elec_com.xlsx' 2 01-Sep-2018

1.2598e+05

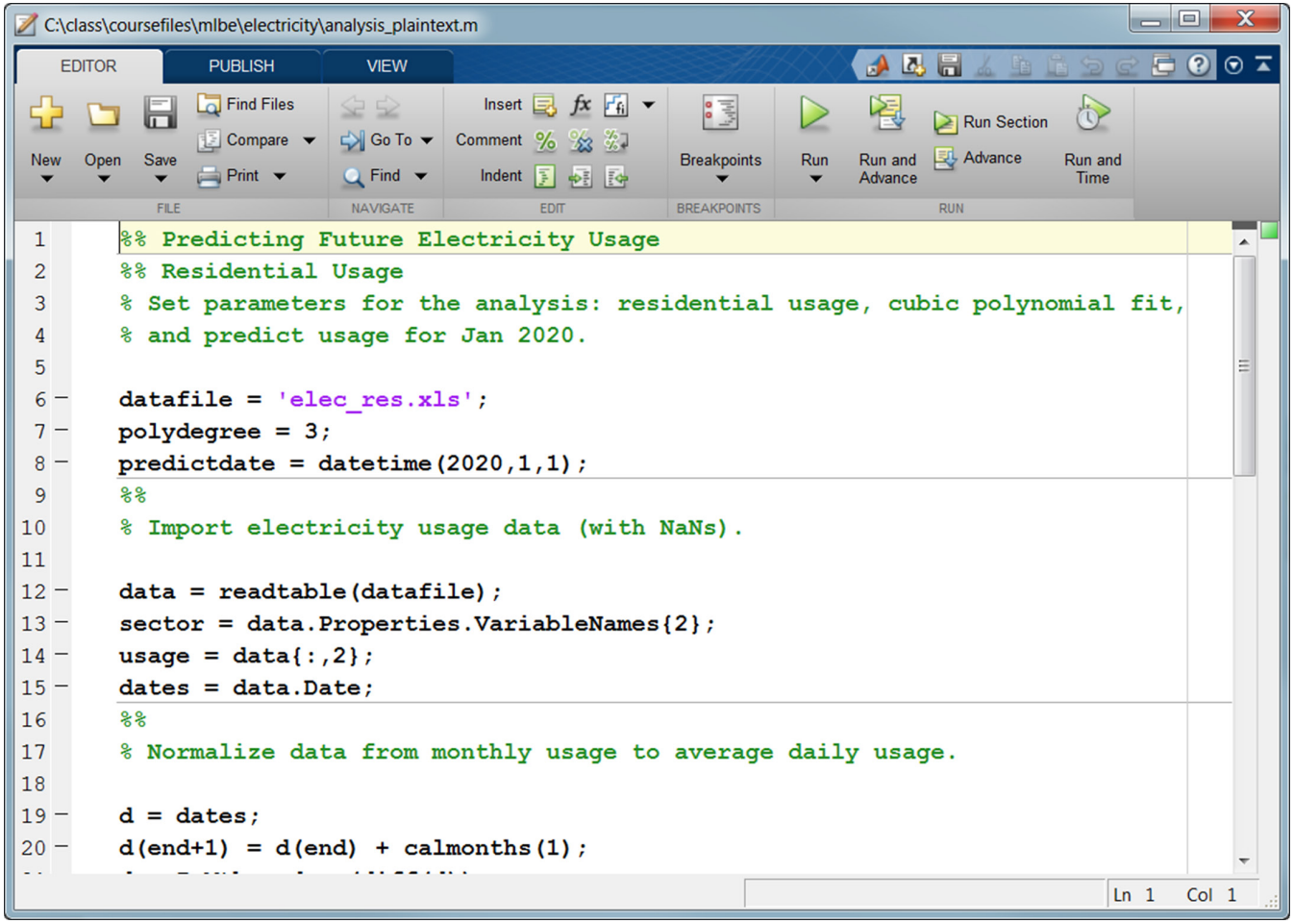
Workspaces


workspaces.mlx
1 a = 42;
2 b = foo(a);
42
0.7623
foo
function y = foo(x)
a = sin(x);
x = x + 1;
b = sin(x);
y = a*b;
end
Workspace
Name Value Size Class
a 42 1x1 double
b 0.7623 1x1 double
Workspace
Name Value Size Class
a -0.9165 1x1 double
b -0.8318 1x1 double
x 43 1x1 double
y 0.7623 1x1 double
BlackBox

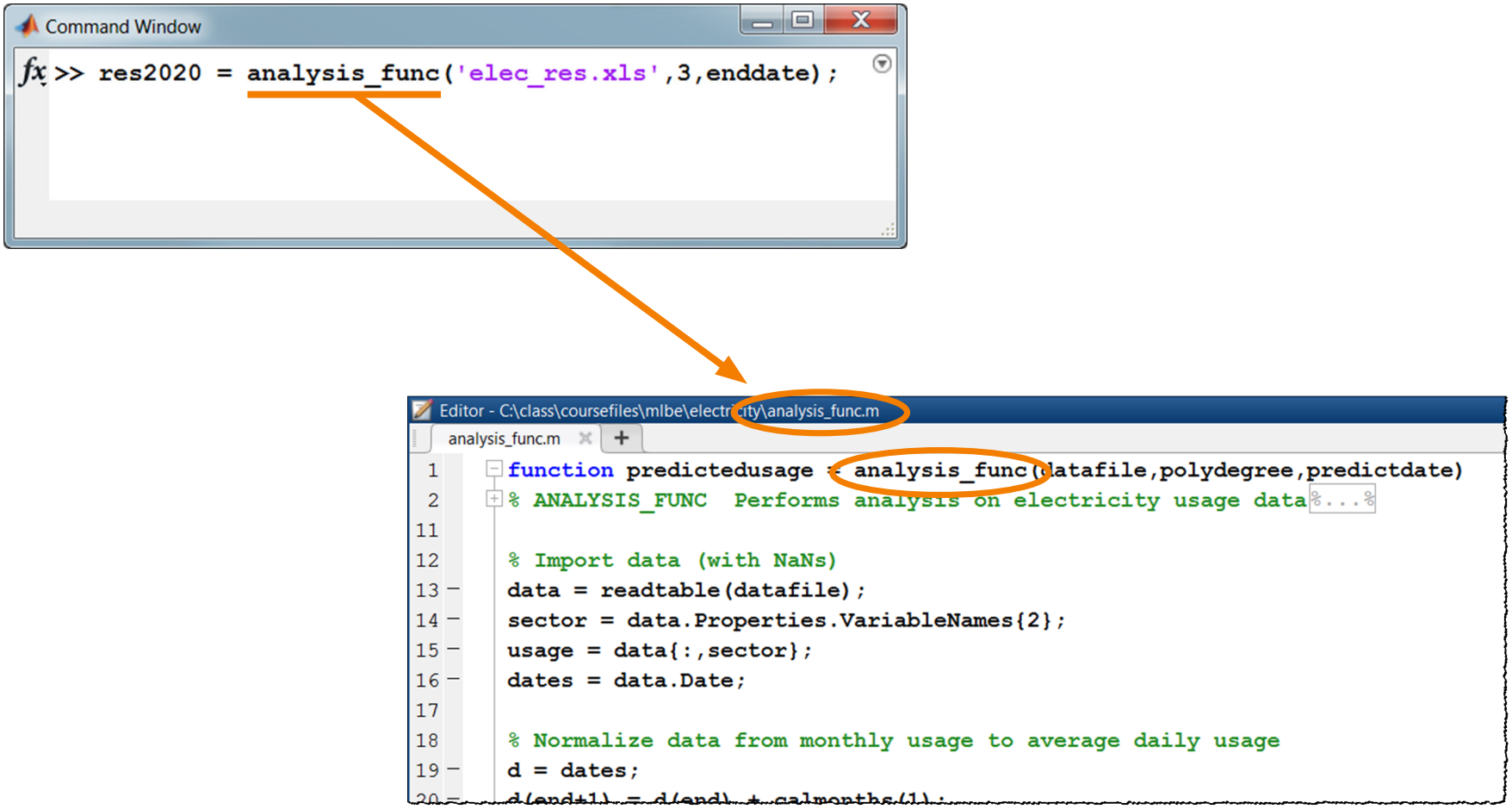
The MATLAB® Editor

Plain Text Code File

(.m extension)



Creating a Function File



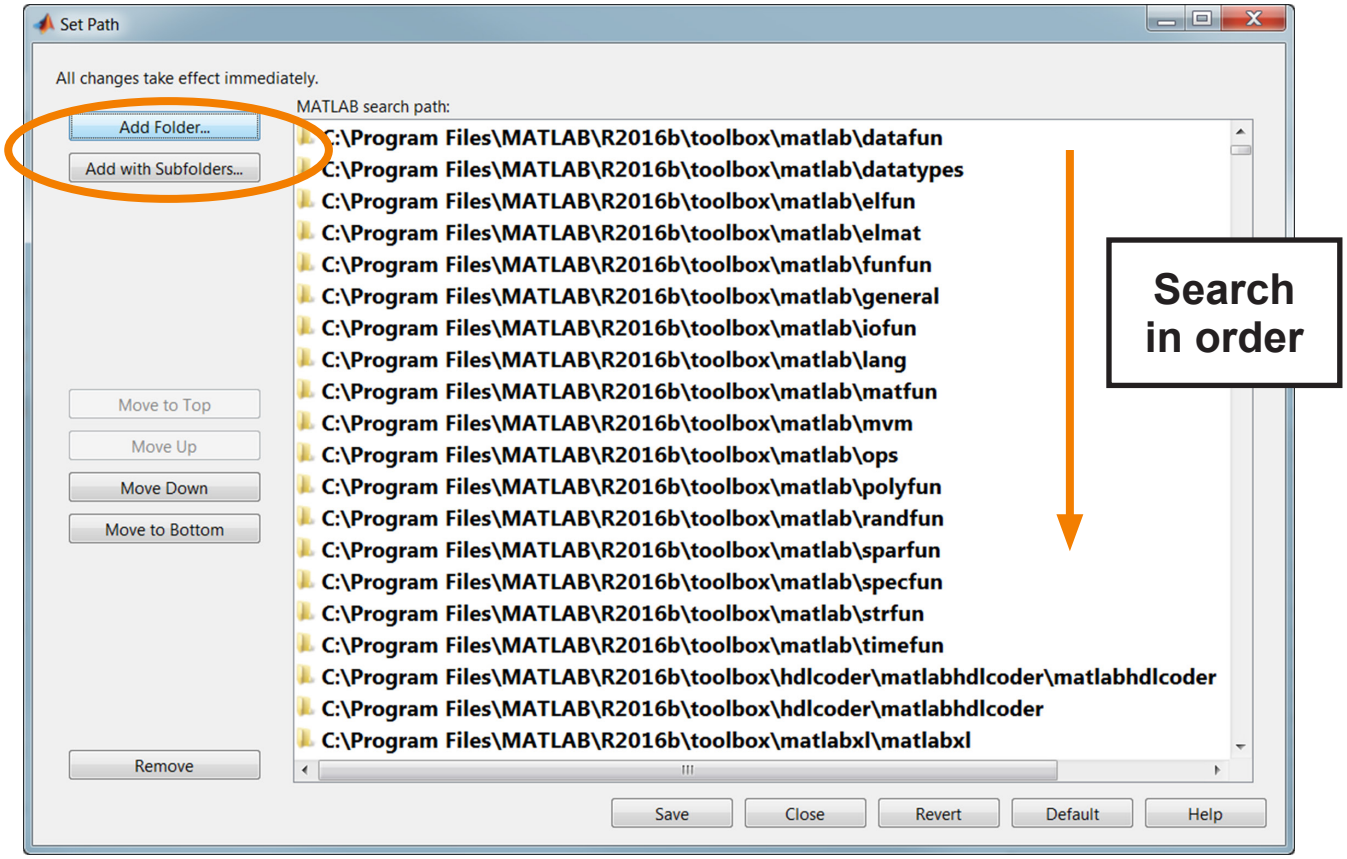
Calling Precedence

```
>> analysis
```

1. Variable
2. Imported package function
3. Nested function
4. Local function
5. Private function
6. Class method
7. Class constructor
8. File in current directory
9. File on the path

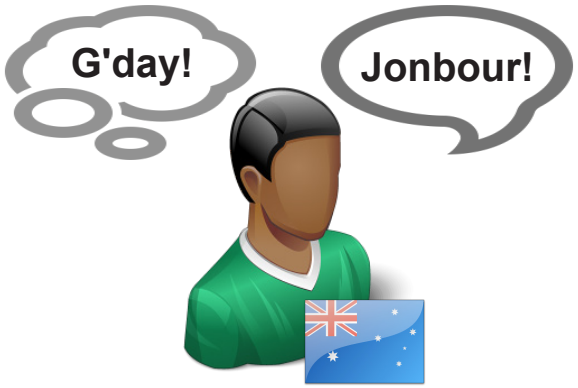
1. Built-in files
2. MATLAB executable (MEX) files
3. Simulink® models
4. MATLAB live scripts
5. Obfuscated MATLAB code files
6. MATLAB code files

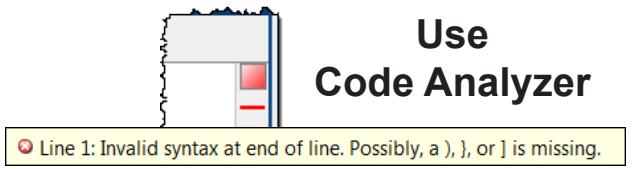
The MATLAB® Path



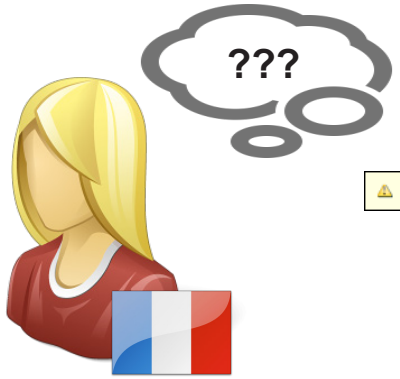
Debugging

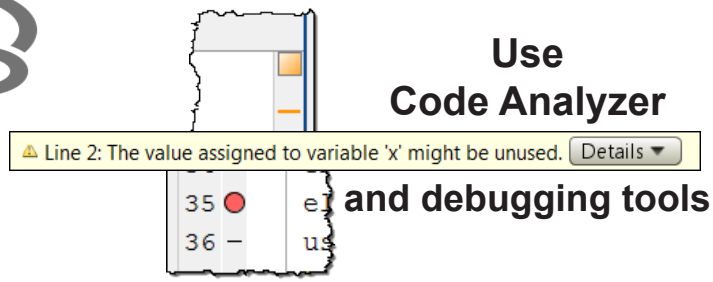
Syntax error





Run-time error

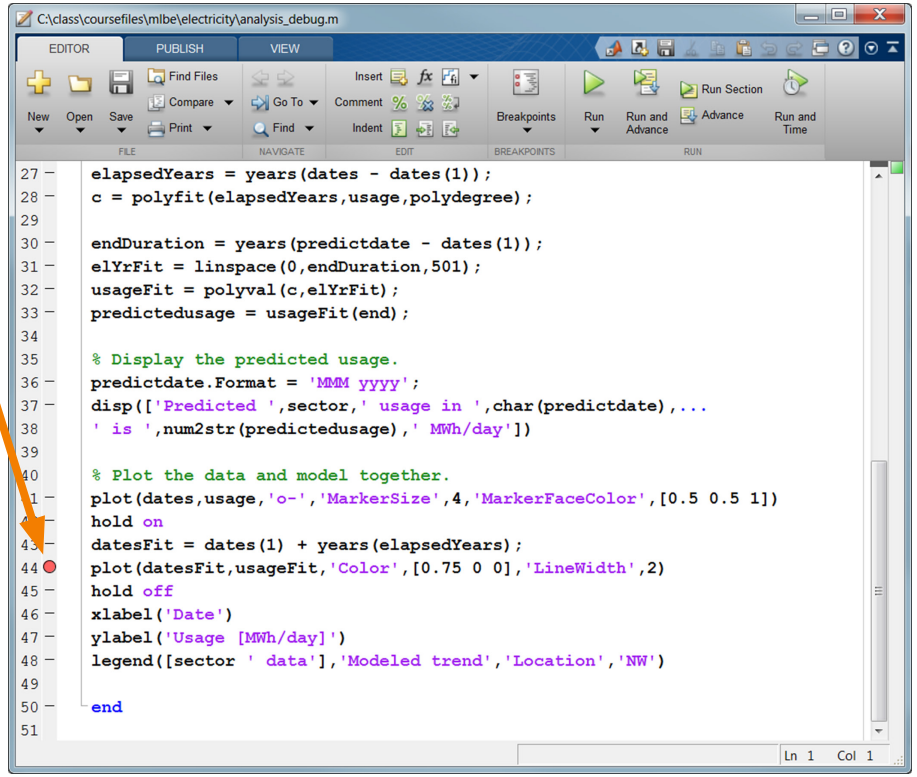




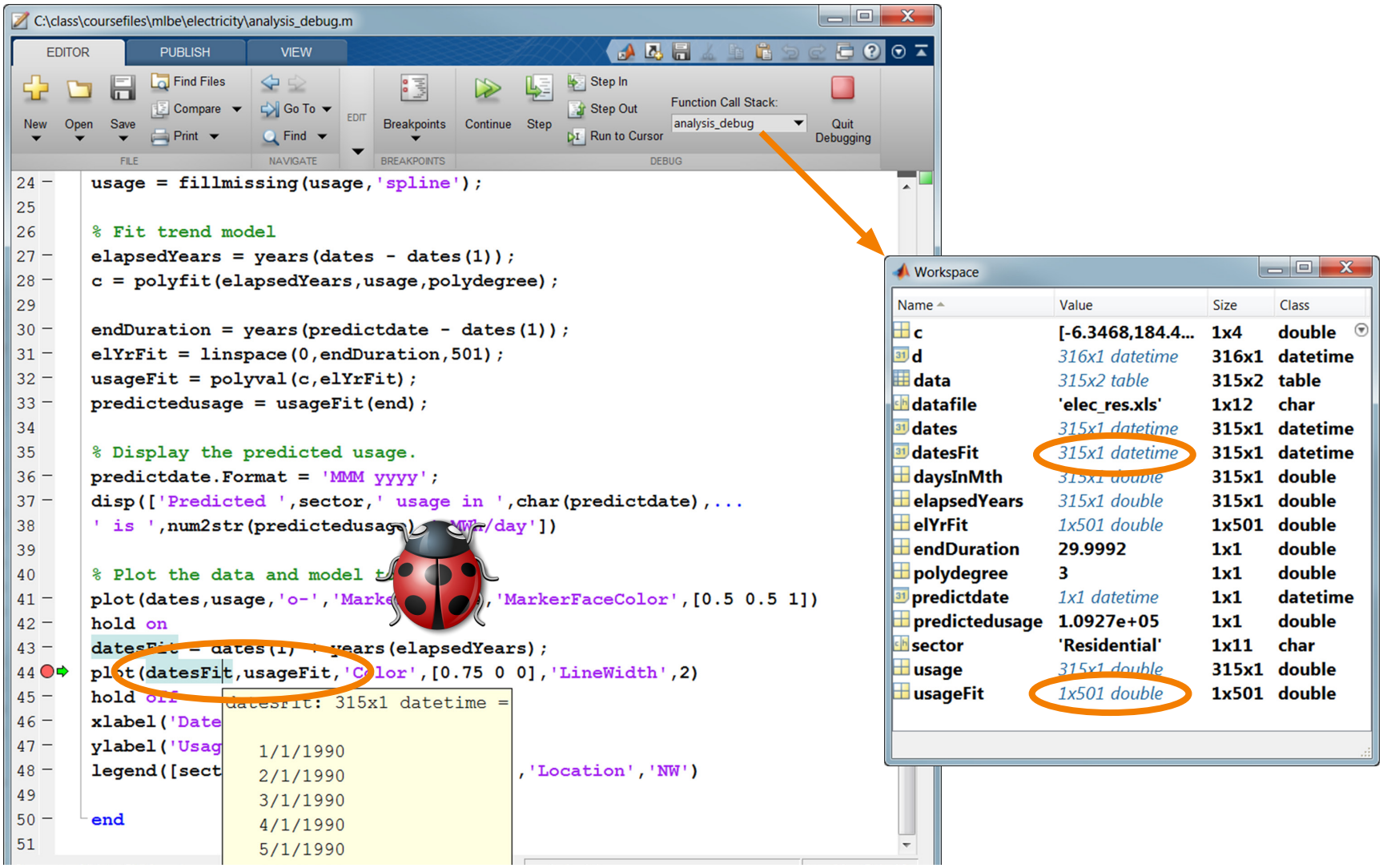
Using Breakpoints

```
>> res2020 = analysis_debug('elec_res.xlsx',3,enddate)
Error using plot
Vectors must be the same length.

Error in analysis_debug (line 44)
plot(datesFit,usageFit,'Color',[0.75 0 0],'LineWidth',2)
```

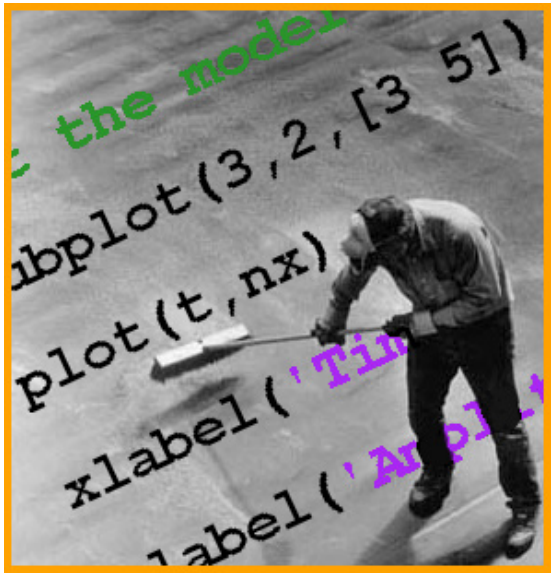


Examining Values



Ending Debugging

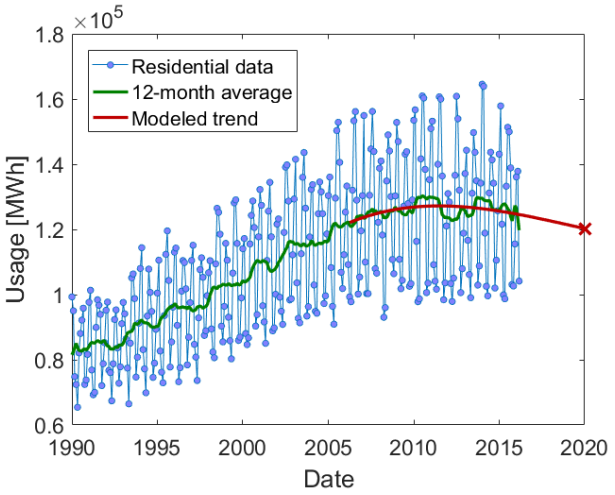
- Exit debug mode
- Correct errors
- Save changes
- Test and confirm
- Clear or disable breakpoints



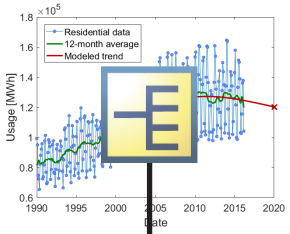
Course Example: Adding Model Parameters

- File name
- Polynomial degree
- Prediction date
- Number of months for averaging
- Number of months in the fitted model
- Create plot?

```
analysis
```



Combining Heterogeneous Data with Structures



File name

Polynomial degree

Prediction date

Number of months for averaging

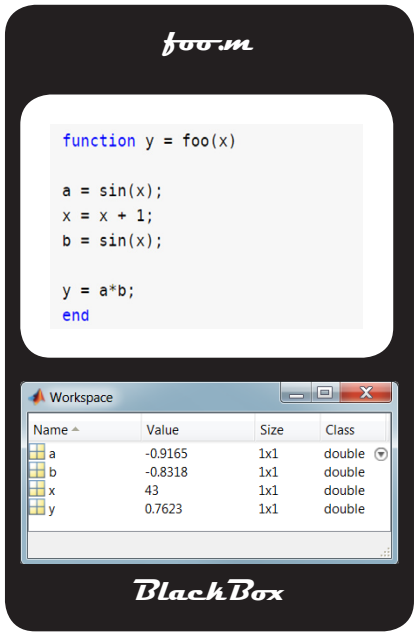
Number of months in the fitted model

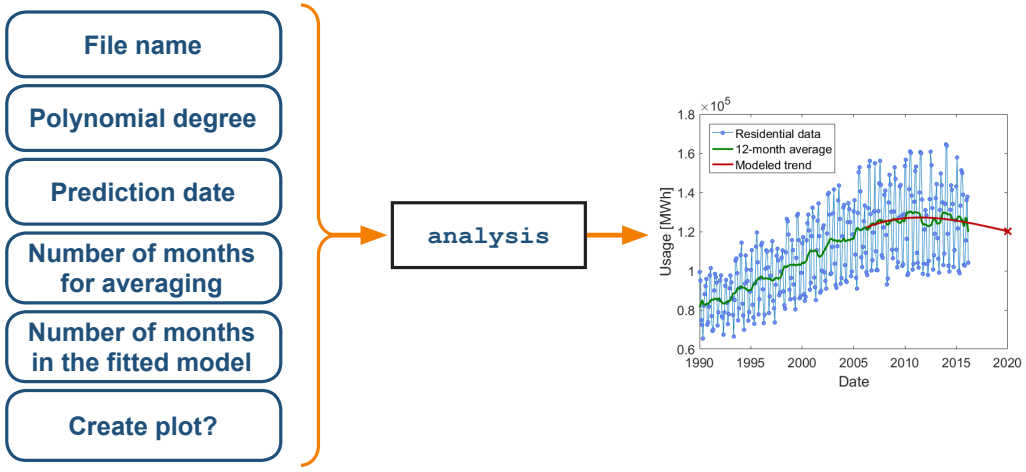
Create plot?

```
analysis
```

Summary

- Creating and calling functions
- Workspaces
- Plain text code files
- Path and precedence
- Debugging
- Using structures





Test Your Knowledge

1. Suppose the workspace contains a vector `x` and the function `foo` creates a scalar variable also called `x`. What will happen to the vector `x` if you issue the command `z = foo(7)`?

   A. It will change to the scalar value defined in `foo`.

   B. Nothing, because the two variables are of different dimensions (i.e., a vector cannot be changed to a scalar).

   C. Nothing, because the vector `x` is not passed as an argument to `foo`.

   D. Nothing, because `foo` maintains its own workspace, so there is no name conflict.

Test Your Knowledge

2. Which of the following is a valid function declaration?

   A. `function [x,y] = foo(a,b)`

   B. `function foo(a,b) = [x,y]`

   C. `[x,y] = function foo(a,b)`

   D. `[x,y] = foo(a,b)`

Test Your Knowledge

3. (Select all that apply) Which of the following are valid calls to a function with the declaration line `function [x,y] = foo(a,b)`?

   A. `[x,y] = foo(a)`

   B. `x = foo(a,b)`

   C. `[x,y] = foo[a,b]`

   D. `(x,y) = foo(a)`

   E. `[x,y] = foo(a,b)`