Powerful and scary

ADVANCED FUNCTIONS AND DEBUGGING

Topics

- Introduction: GUI and basic calculations
- Coding 1: Scripts, style, and variable classes
- Coding 2: Control statements and loops
- Visualization 1: Basics, subplots, get and set
- Coding 3: Functions
- Visualization 2: Descriptive plots
- Coding 4: Basic input and output
- Visualization 3: Distribution and 3D plots
- Coding 5: Input and output specials – last lecture before holidays
- Machine Learning 1: Introduction and dimension reduction
- Machine Learning 2: Clustering
- Machine Learning 3: Classification
- Coding 6: Efficiency and debugging basics
- Coding 7: Advanced functions and debugging

Anonymous Functions

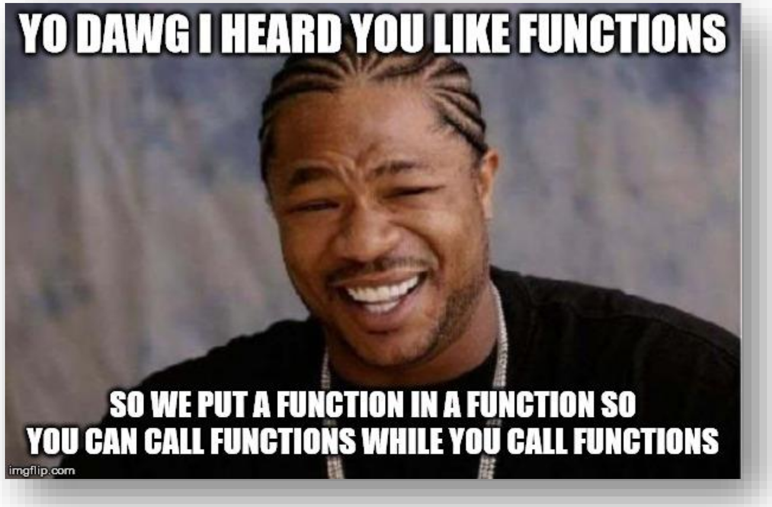
- Functions can also be created as handles, usually in just one line
  - myFunc = @(input) usage(input);
    - mySquareFunc = @(x) x^2;
  - Multiple inputs are possible as well
    - myLineFunc = @(x,a,b) a*x+b;
- They can be used like regular functions inside the script or larger function
- Beware: They DO have access to the workspace they have been created in!
  - Starting in MATLAB 2017

Functions Input Parser

- Instead of having necessary parameters to pass a function, we can use keys and values
  - E.g. “raincloud_plot(myData, 'box_on', 1, ‘connecting_lines’, 1)”
  - Mandatory and optional input can be differentiated and default values can be set
  - Input checks happen in the parsing with specified conditions for each parameter
    - Built-in or self-built anonymous functions possible

Hidden Functions

- It is possible to create a function after another function in the same document
  - It will only be known to MATLAB from inside the first function!
  - It will also again have ist own workspace
  - Only use this if you are sure that only your larger function will ever make use of the hidden function!


YO DAWG I HEARD YOU LIKE FUNCTIONS
SO WE PUT A FUNCTION IN A FUNCTION SO YOU CAN CALL FUNCTIONS WHILE YOU CALL FUNCTIONS
imgflip.com