

AAE 637: Lab 1*

1/28/2015

Can one be a good data analyst without being a half-good programmer? The short answer to that is, 'No.' The long answer to that is, 'No.'

— Frank Harrell

Programming, at least in this class, is nothing more than writing a set of instructions for performing a calculation or carrying out an algorithm.

Numerical Precision

Computers can't do everything implied by math notation since it must store things concretely rather than abstractly. For example, $\sqrt{2}$ is a well-defined mathematical concept, but computers face the problem of storing its infinitely long series of digits in a finite space. The method that computers use to store non-integers is called floating point.

In math notation, the scale of the numbers doesn't matter. In MATLAB, scale often means the difference between code that works and code that chokes. Scale all variables to within a few orders of magnitude. For example, if you have number of children in a household and annual income as RHS variables, scale income to thousands of dollars.

Version Control

Assignment_1.m

Assignment_1_test.m

Assignment_1_test3.m

Assignment_1_testFinal.m

Assignment_1_testFinalFINAL.m

...we've all been there. Solve it with a version control system. It's similar to Word's "track changes". I use GitHub.

*prepared by Travis McArthur, UW-Madison (<http://www.aae.wisc.edu/tmcarthur/teaching.asp>)

Debugging strategies

1. Unmask the hidden state of the program.
2. Isolate the problem. Run your program to yield a known output to a simpler calculation to diagnose the problem when the output of the desired calculation is unknown.
3. Make functions rather than copy-paste
 - (a) If there is a bug, you want to squash it on one place rather than allow it to haunt you everywhere
 - (b) Just add argument options and use `if` to make functions do slightly different things rather than writing a new function
4. Make your variable names descriptive so you will understand your program when you view it later
5. In comments, write down where you got formulas, e.g. book pages or equation numbers
6. Also use comments to temporarily set aside questionable code rather than deleting it outright
7. Read and understand error messages.

Bits and Pieces

- To pass a function as an argument to another function, use “function handles”, which is just typing `@` before the function name.
- `*`, `/`, and `^` perform the matrix operations. Use `.*`, `./`, and `.^` for element-by-element multiplication, division, and exponentiation.
- Pre-allocate your matrices; don’t grow them.
- Know the “scope”, i.e. what MATLAB can “see” at any given time.
- Global variables
 - You must declare the global variable via `global <varname>` in the workspace *and* within the function that you intend to use it. You’d use globals for functions that are several layers deep, like the step length for computing numerical derivatives.
 - Professional programmers believe that coding with globals is coding dangerously. The alternative is to explicitly pass each argument through each function “layer”, but this can become cumbersome.
- MATLAB can only see functions that are in your working directory
- Use built-in functions where appropriate, e.g. `chol()` for checking matrix positive definiteness, `blkdiag()` to create block diagonal matrices, etc.

Common Error Messages¹

Example of Error Message	Means of Resolving
Undefined function or variable 'exampleArray'.	Did you mean to type "exampleArray?" If so, you need to make sure you define it properly before you use it.
Error using * Inner matrix dimensions must agree.	The matrices you are trying to multiply are not conformable. You aren't trying to perform element-by-element matrix multiplication are you?
Error: Expression or statement is incorrect—possibly unbalanced (, {, or [.	A (, {, [,), }, or] is likely missing. Did you forget to close (or open) a bracket? Or did you put in one (or more) too many?
Index exceeds matrix dimensions.	You are trying to access a part of the matrix that does not exist. Check the size of the matrix you are working with.
if exampleArray(i) = 1 Error: The expression to the left of the equals sign is not a valid target for an assignment.	When using if statements, you must use == (two equal signs) instead of = (one equal sign) to conduct a logical test. Other acceptable operators in if statements are > (if this is "more than"), >= (if this is "at least as high as"), < (if this is "less than"), <= (if this is "no more than"), (if this "or" that), & (if this "and" that), etc.
Error using functionName (line 1) Not enough input arguments.	You are not feeding the function enough inputs. Type "help functionName" for more information on what inputs the function requires.
Undefined function 'functionName' for input arguments of type 'double' or 'char'.	The function you are trying to access probably does not exist. Try searching for it in the User's Guide. It may have been deprecated or not in your Current Folder.
Conversion to cell from char is not possible.	You are trying to treat a cell in a cell array like an element in a regular array.
Cell contents assignment to a non-cell array object.	You are trying to treat an element in a regular array like a cell in a cell array.
Function definitions are not permitted in this context.	You cannot call functions in the same script you created them in. Create a separate M file for this purpose.
Warning: Matrix is close to singular or badly scaled.	The determinant of a matrix is very close to 0 (that is, the matrix is close to not being invertible). If so, the columns (or rows) of the matrix are close to being linearly dependent. Check the exogenous variable definitions. Did you forget to take the log of a variable?

¹Table courtesy of Chris Taylor