## Intro to MATLAB

## Learning Objectives

* Get to know the MATLAB environment.
* Learn about MATLAB arrays.
* Read tabular data from a file into a program.
* Assign values to variables.
* Select individual values and subsections from data.
* Perform operations on arrays of data.
* Display simple graphs.

**Part 1 – The MATLAB Environment**

Current Folder

Command Window

Workspace => Temporary memory

Search Documentation

**Part 2 – Assigning Variables**

Assigning a variable

X=1

Y=2

Z=X+Y

Variable names must begin with a letter, and can contain numbers or underscores

Variable arithmetic (plus, negative, divide, multiply)

Vectors (row, column, transpose, size) - use [ ]

Matrices

Characters and strings – use ‘ ‘

CHALLENGE:

* Assigning variables from the command line
* Make a variable called "Age\_2015", set it equal to your current age
* Make another variable called "Age\_2025", set this variable to Age\_2015 plus 10.
* Make another variable (choose the name yourself) and set it equal to half your age.

**Part 3 – Loading Data**

Inflammation data (each row: individual patient, each column: consecutive day)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Day 1 | Day 2 | Day 3 | … | Day 40 |
| Patient 1 | … |  |  |  |  |
| Patient 2 |  | … |  |  |  |
| Patient 3 |  |  | … |  |  |
| … |  |  |  | … |  |
| Patient 60 |  |  |  |  | … |

“read csv” in documentation – string as input

Look at csv

patient\_data = csvread('inflammation-01.csv');

**Indexing Data – “Taking Slices”**

Look at the first patient, first day – use matrix\_name(row index, column index )

Patient1\_Day1 = patient\_data(1,1)

Look at second patient, all days

Patient2\_AllDays = patient\_data(2,:)

Look at all patients, day 5 through to 10

AllPatients\_5Days = patient\_data(:,[5,6,7,8,9,10])

% OR

AllPatient\_5Days = patient\_data(:,5:10)

More vector making

AllNumbers = 1:10;

EverySecondNumber = 1:2:10; % start : step : end

Look at all patients, every second day

AllPatients\_AlternateDays = patient\_data(:,1:2:end)

Introduce disp command - Use disp to display mixed type variables

disp(‘this will display in the command line’)

disp([‘Patient 1 inflammation on day 5 is’ num2str(patient\_data(1,5))])

CHALLENGE:

* Move to software carpentry for challenge

**RECAP**

Define vectors and matrices with [ ]

Define strings and characters with ‘ ‘, and squish them together with [ ]

Index a matrix or vector with ( )

**Part 4 – Analysing Data**

Finding the mean of the inflammation on all days – use documentation

mean(patient\_data(:))

 maximum, minimum and standard deviation

max(patient\_data(:))

min(patient\_data(:))

std(patient\_data(:))

We can display those values nicely:

disp(['Maximum inflammation: ', num2str(max(patient\_data(:)))]);

disp(['Minimum inflammation: ', num2str(min(patient\_data(:)))]);

disp(['Standard deviation: ', num2str(std(patient\_data(:)))]);

Display the max for patient 1

Patient1\_Max=max(patient\_data(1,:));

disp(['Max inflammation of patient 1 is ', num2str(Patient1\_Max)]);

Performing operation across dimensions

mean(patient\_data, 1)

Dimension 2

mean(patient\_data, 2)

**Part 4 – Displaying Data**

Heat map

imagesc(patient\_data)

Look at avg inflammation over time

ave\_inflammation = mean(patient\_data, 1);

plot(ave\_inflammation);

Looks weird

Look at other statistics

plot(max(patient\_data, [], 1));

title('Maximum inflammation per day');

and

plot(min(patient\_data, [], 1));

title('Minimum inflammation per day');

Challenge

* Look at software carpentry for challenge

**SUBPLOTTING**

subplot(1, 2, 1);

plot(max(patient\_data, [], 1));

ylabel('max')

subplot(1, 2, 2);

plot(min(patient\_data, [], 1));

ylabel('min')