## Making Choices

## Learning Objectives

* Learn about if, elseif and else
* Learn to test equality, AND, and OR conditions
* Learn to nest loops

**Part 1 – If Statements**

The programs we have written so far always do the same things, regardless of what data they’re given.

We want programs to make choices based on the values they are manipulating

Conditionals

num = 37;

if num>100

disp('number is greater than 100');

else

disp('number is not greater than 100');

end

disp('done')

Try without else block

Other logical operators

>, <, >=, <=, == (test if something is true, not assigning)

Introduce elseif

Code to return sign of number

% check the sign of a number

if num>0

sign = 1;

elseif num==0

sign = 0;

else

sign = -1;

end

AND, OR tests (use whiteboard)

* If (raining && going outside) => bring umbrella.
* If (I’m hungry || I’m bored) => Eat

AND statement (true if both statements are true)

EXPLAIN TRUE/FALSE IN MATLAB

True = 1

False = 0

Lower-case true and false are also ‘special’ words in MATLAB

% test if I need an umbrella

raining=1; %it's raining!

going\_outside=0; % not going outside

if raining && going\_outside

disp('better bring an umbrella!')

else

disp('no need for an umbrella!')

end

% test if I should eat

hungry=0;

bored=0;

if hungry || bored

disp('eat!')

else

disp('dont eat....yet...')

end

OR statement (true if either or both statements are true)

% test OR statements

if (1>0)||(3<4)

disp('at least one part is true')

end

% test if I should eat

hungry=0;

bored=0;

if hungry || bored

disp('eat!')

else

disp('dont eat....yet...')

end

When you use conditionals MATLAB tries to convert everything to either a 0 or a 1 (true or false)

Set hungry = 42

Set hungry = ‘hungry’

The code still works!

CHALLENGE:

% Write code to work out which number is bigger (and try changing the

% values of num1 and num2 to make sure your code always works)

num1 = 40;

num2 = 25;

CHALLENGE EXTENSTION:

Modify the code so that the correct statements are displayed for number 1 and number 2. Check your code with some different values of number 1 and number 2 – does it still work?

**Part 2 - Nesting**

Combining if statements with loops

numbers = [-5, 3, 2, -1, 9, 6]; % list of numbers to test

total = 0; % initialise value to zero

for n = numbers %loops through each element in "numbers" vector

if n >= 0 % checks if the number is positive

total = total + n; %adds it to the total of positive numbers

end % closes the "if" statement

end % closes the "for" loop

disp(['sum of positive values: ', num2str(total)]) %displays the sum

CHALLENGE:

Edit the code so it also displays the sum of the negative values

CHALLENGE ANS:

pos\_total = 0;

neg\_total = 0;

for n = numbers

if n >= 0

pos\_total = total + n;

else

neg\_total = neg\_total + n;

end

end

disp(['sum of positive values: ', num2str(pos\_total)]);

disp(['sum of negative values: ', num2str(neg\_total)]);

EXTENSION:

Using the commands we have learnt previously, write a script that loads in some patient data and select one patient vector (choose any patient you like HINT: remember how to slice a single vector out of a data matrix?). Loop through this patient vector and add up the number of days the patient’s inflammation was above 10% of the mean value for that patient.

EXTENSION ANS:

clear;

clc;

% load patient data

patient\_data = csvread('inflammation-02.csv');

% select the first patient to analyse

my\_patient = patient\_data(1,:);

% find the mean value

patient\_mean = mean(my\_patient);

threshold = patient\_mean + 0.1 \* abs(patient\_mean)

% initialize the sum of days where the inflammation was within 10% of the

% max

sum\_high\_inflammation\_days = 0;

% loop through the patients values

for n = my\_patient

% check if inflammation is higher than threshold

if inflammation\_percent >= threshold

% add one day to the total number

sum\_high\_inflammation\_days = sum\_high\_inflammation\_days + 1;

else

% do nothing (could we just remove this else statement? YES)

end

end

disp(['The total number of days with high inflammation is: ', num2str(sum\_high\_inflammation\_days)]);

**Part 3 – Conditional Vectors**

**\*\* maybe skip this section**

One of the great things about MATLAB is we don’t always need to nest conditional statements in a loop. MATLAB can condition vectors directly.

Why do we use && and || in conditions?

Because & and | are for vector conditionals

vector1 = [1 0 0 1 0 1];

vector2 = [1 0 1 0 0 0];

vector1 & vector2

vector1 | vector2

We can do this vector conditioning with all our logical operators

numbers = [-5, 3, 2, -1, 9, 6];

numbers < 0

numbers >= 0

numbers(numbers < 0)

numbers(numbers >= 0)

pos\_total = sum(numbers(numbers >= 0));

neg\_total = sum(numbers(numbers < 0));

disp(['sum of positive values: ', num2str(pos\_total)]);

disp(['sum of negative values: ', num2str(neg\_total)]);

CHALLENGE

How would you display the total number of negative and positive numbers (without using a loop)

ANS:

sum(numbers >= 0)

sum(numbers < 0)

EXTENSION:

Modify your script that calculated the number of high inflammation days for one patient so that it does not require a loop to do so.   
HINT: one way to do this is to set an upper and lower boundary condition that represent ±10% of the maximum inflammation. Then test when the patient vector is between these two boundary conditions using a command like this:

(my\_patient >= lower\_bound) & (my\_patient <= upper\_bound);

ANS:

clear;

clc;

% load patient data

patient\_data = csvread('inflammation-02.csv');

% select the first patient to analyse

my\_patient = patient\_data(1,:);

% find the maximum value

patient\_max = max(my\_patient);

% set the boundary conditions

lower\_bound = patient\_max - 0.1\*patient\_max;

upper\_bound = patient\_max + 0.1\*patient\_max;

% apply the conditions to the vector

high\_inflammation\_days = (my\_patient >= lower\_bound) & (my\_patient <= upper\_bound);

sum\_high\_inflammation\_days = sum(high\_inflammation\_days);

% display outcome

disp(['The total number of days with high inflammation (10% of max) is: ', num2str(sum\_high\_inflammation\_days)]);