Lesson 2: Matrices

%TO WARM UP:

% 1) comment out the following line and copy it in the 'command window' to

% run it. then write a comment describing the output you got (how many row x

% columns?)

[1, 4, 6; 2, 3, 7]

%2) create a matrix 3 x 2, (pizza and cake \* carrot, tomato, beans).

% run the matrix

% comment here the error you got.

% assign values to pizza, cake, carrot, tomato, and beans.

%re-run the matrix 3 \* 2 of point (2). Call the matrix "food\_matrix".

food\_matrix = [cake, pizza; carrot, tomato, beans]

%ERROR: Error using vertcat

% Dimensions of arrays being concatenated are not consistent.

%fix suppressing beans. 2x2 matrix

food\_matrix = [cake, pizza; carrot, tomato]

% run food\_matrix, but suppress the output. (search on google

%"how to suppress matlab output")

food\_matrix = [cake, pizza; carrot, tomato]; %use ";"

% 3) check the size of food\_matrix (use the "size()" function)

size(food\_matrix)

# %% Rand

%run this:

one\_by\_one = rand(1)

two\_by\_two = rand(2)

five\_by\_five = rand (5)

%what does rand do?

% it creates 1by1, 2 by 2, and 5 by 5 matrices

%try this:

two\_by\_three = rand(2,3) %first argument = rows, second argument = columns

three\_by\_two = rand(3,2)

%what happen when rand takes two arguments?

%Use the zeros function to create a matrix of all zeros (use the zeros()func )

%that has 6 rows and 3 columns (6-by-3). Assign the result to a variable named "vuoto".

vuoto = zeros(6,3)

%by the way, vuoto means "empty" in italian.

%check the size of "vuoto", using the size() func.

size(vuoto)

% 4) "BORING TASK": create a vector from 1 to 20 named

% "what\_is\_the\_point\_of\_this\_exercise"

what\_is\_the\_point\_of\_this\_exercise = [1, 2, 3, 4, 5, 6, 7, 8,9 ,10, 11, 12, 13, 14, 15, 16, 17, 18, 19,20]

%END first part. DO NOT start next part right away.

# Vectors

%Go to the documentation and read about linspace: check the example

%section, specifically "vector with specified number of values".

% Using linspace(),create a vector that starts at 10, ends at 15, and has 10 elements.

%call the vector "lin"

lin = linspace(10,15,10)

%transpose lin using the transpose operator (')

linT = lin'

%check the length of "lin" by using the function "length()".

length(lin)

length(linT)

%STOP HERE! WAIT TO DO NEXT PART

# Vector indexing

% create a random one-dimensional 1 by 8 (aka, make a vector).

%you can use rand(row, columns). col the vector "v"

v = rand(1,8)

% select the 3rd element of the vector v. Copy the line of code in the

% command window to check.

v(3)

% change the value of the 3rd element of v to 0.999

v(3) = 0.999

%A single range of index values can be used to reference a subset of vector elements.

%returns a subset of vector v containing the elements from 3 to the end.

%(use :end)

x = v(3:end)

%Using a range of index values, create a vector named "little\_v" containing

%the 2nd through 5th elements of vector v.

little\_v = v(2:5)

# Matrix indexing

% Create a matrix 5 by 5 of random numbers(function "rand()"). name this

% matrix "data".

data = rand(5)

%Create a variable named "second" that contains the second column of the

%matrix named data.

second = data(:,2)

%create a variable named "last\_two\_col" containing the last two COLUMNS of data.

%HINT: The colon operator can refer to a range of values.

%The following syntax creates a matrix containing the first, second,

%and third ROWS of the matrix A. x = A(1:3,:)

last\_two\_col = data(:,4:5)

%run this section of the script and check if it works.