%

% file i/o

%

close all;

clear all;

%

% Part B) Process Imported Data - final\_scores\_example.xlsm

%

% - right click on final\_scores\_example.xlsm and select Import Data

% - with your mouse select all rows and columns of data

% - then go to Output Type and select Numeric Matrix

% - lastly, click on Import Selection button

% - then in Workspace click on finalscoresexample matrix to display the data

%

% open in Workspace the matrix finalscoresexample and by hand

% change NaN with the number 0

%

% read file newdatafile.mat

% get size of matrix

%

load newdatafile

sizefse = size(finalscoresexample)

%

% extract numbers in chosen column

% display scores

%

col = finalscoresexample(:,1)

[r c] = size(col)

figure(1);

%stem(col,'rs-','MarkerFaceColor','b','MarkerSize',8,'MarkerEdgeColor','g');

plot(col,'rs-','MarkerFaceColor','b','MarkerSize',8,'MarkerEdgeColor','g');

title('Student Scores');

xlabel('Number of Students');

ylabel('Scores');

axis([0 length(col) -5 max(col)\*1.2]);

grid minor

%

% find mean

%

meanscores = mean(col);

var\_scores = var(col);

std\_scores = sqrt(var\_scores);

meanL = ones(length(col))\*meanscores;

%

%

% do a histogram of scores using 30 bins

%

figure(2);

hist(col,50)

grid minor;

%

% sort scores in ascending order

%

[scoresa, indexsa] = sort(col)

%

% one way to sort scores in descending order

% [scoresd, indexsd] = sort(col,'descend')

% flip the order of ascend sort

scoresd = scoresa(end:-1:1)

indexsd = indexsa(end:-1:1)

%

% display scores and location (index)

%

figure(3);

stem(scoresa,'k-o','MarkerFaceColor','r','MarkerSize',8,'MarkerEdgeColor','g');

title('Student Scores - Ascending order');

xlabel('Number of Students');

ylabel('Scores');

grid minor;

%

figure(4);

hold on;

stem(indexsa,scoresa,'b-o','MarkerFaceColor','b','MarkerSize',8,'MarkerEdgeColor','g');

title('Student Scores - Student Number');

xlabel('Student Number');

ylabel('Scores');

plot(meanL,'r:o');

axis([0 131 0 110])

grid minor;

%

spread\_mean = col - meanscores;

%

figure(5);

hold on;

stem(spread\_mean,'b-o','MarkerFaceColor','b','MarkerSize',8,'MarkerEdgeColor','g');

title('Student Scores - Student Number');

xlabel('Student Number');

ylabel('Scores from the Mean');

%plot(meanL,'r:o');

grid minor;

%

% find minimum and maximum scores and their index (location)

%

[maxs, maxi] = max(col)

[mins, mini] = min(col)

%

% find index (location) of students in ranges and the number

% of students in each range

% scores >= 90 - A

% scores >= 80 and less than 90 - B

% scores >= 70 and less than 80 - C

% scores >= 60 and less than 70 - D or less

%

% for 1D vectors gives location

%

% finds students with score > 90 - logical array

% 1 or T are > 90; 0 or F are < 90

% grabs True values

%

% finds students with their socred and grade

%

s90 = find(col>=90);

ss90 = (col >= 90);

AS = length(s90)

S90 = ss90 .\* col;

%

s80\_90 = find(col>=80 & col<90);

ss80\_90 = (col>=80 & col<90);

BS = length(s80\_90)

S80\_90 = ss80\_90 .\* col;

%

s70\_80 = find(col>=70 & col<80);

ss70\_80 = (col>=70 & col<80);

CS = length(s70\_80)

S70\_80 = ss70\_80 .\* col;

%

s60\_70 = find(col>=60 & col<70);

ss60 = (col>=60 & col<70);

DS = length(s60\_70)

S60 = ss60 .\* col;

%

% for 2D matrices

%

%[i90,j90] = find(col>90);

%[i80,j80] = find(col>80 & col<90);

%[i70,j70] = find(col>70 & col<80);

%[i60,j60] = find(col>60 & col<70);

%

figure(6)

data = [AS BS CS DS];

explode = [1 0 0 0 ];

pie(data, explode);

title('Final Grades');

legend('As','Bs','Cs','Ds','Location','Best');

%

disp('>>> END of arrays\_final\_scores\_9b.m <<<')