9a code:

%

% file i/o

%

%close all;

%clear all;

%clc;

%

% Part A) Import 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

%

% save new data to disk "newdatafile"

% get size of matrix

%

save newdatafile finalscoresexample

size (finalscoresexample)

%

% read file newdatafile.mat

%load newdatafile

%

disp('>>> END of arrays\_scores\_9a.m <<<')

9a output:

>> arrays\_final\_scores\_9a

ans =

131 11

>>> END of arrays\_scores\_9a.m <<<

9B CODE:

%

% 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 column 11

% display scores

%

col11 = finalscoresexample(:,11)

[r c] = size(col11)

figure(1);

%stem(col11,'rs-','MarkerFaceColor','b','MarkerSize',8,'MarkerEdgeColor','g');

plot(col11,'rs-','MarkerFaceColor','b','MarkerSize',8,'MarkerEdgeColor','g');

title('Student Scores');

xlabel('Number of Students');

ylabel('Scores');

axis([0 length(col11) -5 max(col11)\*1.2]);

grid minor

%

% find mean

%

meanscores = mean(col11);

var\_scores = var(col11);

std\_scores = sqrt(var\_scores);

meanL = ones(length(col11))\*meanscores;

%

%

% do a histogram of scores using 30 bins

%

figure(2);

hist(col11,50)

grid minor;

%

% sort scores in ascending order

%

[scoresa, indexsa] = sort(col11)

%

% one way to sort scores in descending order

% [scoresd, indexsd] = sort(col11,'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 = col11 - 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(col11)

[mins, mini] = min(col11)

%

% 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(col11>=90);

ss90 = (col11 >= 90);

AS = length(s90)

S90 = ss90 .\* col11;

%

s80\_90 = find(col11>=80 & col11<90);

ss80\_90 = (col11>=80 & col11<90);

BS = length(s80\_90)

S80\_90 = ss80\_90 .\* col11;

%

s70\_80 = find(col11>=70 & col11<80);

ss70\_80 = (col11>=70 & col11<80);

CS = length(s70\_80)

S70\_80 = ss70\_80 .\* col11;

%

s60\_70 = find(col11>=60 & col11<70);

ss60 = (col11>=60 & col11<70);

DS = length(s60\_70)

S60 = ss60 .\* col11;

%

% for 2D matrices

%

%[i90,j90] = find(col11>90);

%[i80,j80] = find(col11>80 & col11<90);

%[i70,j70] = find(col11>70 & col11<80);

%[i60,j60] = find(col11>60 & col11<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 <<<')

9B output:

>> arrays\_final\_scores\_9b

sizefse =

131 11

col11 =

91.2335

101.4908

103.0043

94.9816

87.1978

78.0378

93.9005

92.8638

78.2205

86.0930

102.7881

96.7124

90.5600

103.5578

86.3514

99.6259

74.1708

91.3135

31.4314

85.2876

105.4411

98.2303

101.7935

83.0130

99.5622

85.9243

70.6886

99.6865

81.1676

102.8422

88.2032

83.8346

75.7297

93.7114

92.1622

76.2843

95.2605

86.3092

81.8108

95.8876

101.7200

75.8411

94.0670

26.5222

50.0541

81.8714

93.6432

81.6173

84.0724

20.0000

89.2281

90.0141

67.7027

88.8714

103.5978

84.2605

92.4281

90.4724

58.6335

93.1308

76.4508

84.6584

92.2130

91.9005

97.7124

75.5276

75.9470

94.2141

99.4908

81.2519

81.3730

104.3070

74.8108

77.7568

93.2249

87.5341

96.8432

63.0865

92.7968

90.0984

81.1351

97.8335

80.8584

92.9914

83.9427

65.1892

80.2162

84.4638

83.6530

88.5686

99.0768

84.6984

74.9232

81.0854

73.6389

65.6000

80.8335

86.3784

81.2389

88.1708

86.5232

36.0714

56.8292

77.6032

72.5795

86.3924

82.5546

42.5946

23.7838

90.8097

41.6216

93.8443

68.0789

62.4184

70.7157

87.3286

74.2962

96.2897

93.2941

91.2649

90.6086

20.0897

103.6897

98.6076

94.8649

91.9373

41.8638

37.0270

88.6876

93.6216

14.0541

r =

131

c =

1

scoresa =

14.0541

20.0000

20.0897

23.7838

26.5222

31.4314

36.0714

37.0270

41.6216

41.8638

42.5946

50.0541

56.8292

58.6335

62.4184

63.0865

65.1892

65.6000

67.7027

68.0789

70.6886

70.7157

72.5795

73.6389

74.1708

74.2962

74.8108

74.9232

75.5276

75.7297

75.8411

75.9470

76.2843

76.4508

77.6032

77.7568

78.0378

78.2205

80.2162

80.8335

80.8584

81.0854

81.1351

81.1676

81.2389

81.2519

81.3730

81.6173

81.8108

81.8714

82.5546

83.0130

83.6530

83.8346

83.9427

84.0724

84.2605

84.4638

84.6584

84.6984

85.2876

85.9243

86.0930

86.3092

86.3514

86.3784

86.3924

86.5232

87.1978

87.3286

87.5341

88.1708

88.2032

88.5686

88.6876

88.8714

89.2281

90.0141

90.0984

90.4724

90.5600

90.6086

90.8097

91.2335

91.2649

91.3135

91.9005

91.9373

92.1622

92.2130

92.4281

92.7968

92.8638

92.9914

93.1308

93.2249

93.2941

93.6216

93.6432

93.7114

93.8443

93.9005

94.0670

94.2141

94.8649

94.9816

95.2605

95.8876

96.2897

96.7124

96.8432

97.7124

97.8335

98.2303

98.6076

99.0768

99.4908

99.5622

99.6259

99.6865

101.4908

101.7200

101.7935

102.7881

102.8422

103.0043

103.5578

103.5978

103.6897

104.3070

105.4411

indexsa =

131

50

122

109

44

19

102

128

111

127

108

45

103

59

114

78

86

96

53

113

27

115

105

95

17

117

73

93

66

33

42

67

36

61

104

74

6

9

87

97

83

94

81

29

99

70

71

48

39

46

107

24

89

32

85

49

56

88

62

92

20

26

10

38

15

98

106

101

5

116

76

100

31

90

129

54

51

52

80

58

13

121

110

1

120

18

64

126

35

63

57

79

8

84

60

75

119

130

47

34

112

7

43

68

125

4

37

40

118

12

77

65

82

22

124

91

69

25

16

28

2

41

23

11

30

3

14

55

123

72

21

scoresd =

105.4411

104.3070

103.6897

103.5978

103.5578

103.0043

102.8422

102.7881

101.7935

101.7200

101.4908

99.6865

99.6259

99.5622

99.4908

99.0768

98.6076

98.2303

97.8335

97.7124

96.8432

96.7124

96.2897

95.8876

95.2605

94.9816

94.8649

94.2141

94.0670

93.9005

93.8443

93.7114

93.6432

93.6216

93.2941

93.2249

93.1308

92.9914

92.8638

92.7968

92.4281

92.2130

92.1622

91.9373

91.9005

91.3135

91.2649

91.2335

90.8097

90.6086

90.5600

90.4724

90.0984

90.0141

89.2281

88.8714

88.6876

88.5686

88.2032

88.1708

87.5341

87.3286

87.1978

86.5232

86.3924

86.3784

86.3514

86.3092

86.0930

85.9243

85.2876

84.6984

84.6584

84.4638

84.2605

84.0724

83.9427

83.8346

83.6530

83.0130

82.5546

81.8714

81.8108

81.6173

81.3730

81.2519

81.2389

81.1676

81.1351

81.0854

80.8584

80.8335

80.2162

78.2205

78.0378

77.7568

77.6032

76.4508

76.2843

75.9470

75.8411

75.7297

75.5276

74.9232

74.8108

74.2962

74.1708

73.6389

72.5795

70.7157

70.6886

68.0789

67.7027

65.6000

65.1892

63.0865

62.4184

58.6335

56.8292

50.0541

42.5946

41.8638

41.6216

37.0270

36.0714

31.4314

26.5222

23.7838

20.0897

20.0000

14.0541

indexsd =

21

72

123

55

14

3

30

11

23

41

2

28

16

25

69

91

124

22

82

65

77

12

118

40

37

4

125

68

43

7

112

34

47

130

119

75

60

84

8

79

57

63

35

126

64

18

120

1

110

121

13

58

80

52

51

54

129

90

31

100

76

116

5

101

106

98

15

38

10

26

20

92

62

88

56

49

85

32

89

24

107

46

39

48

71

70

99

29

81

94

83

97

87

9

6

74

104

61

36

67

42

33

66

93

73

117

17

95

105

115

27

113

53

96

86

78

114

59

103

45

108

127

111

128

102

19

44

109

122

50

131

maxs =

105.4411

maxi =

21

mins =

14.0541

mini =

131

AS =

54

BS =

39

CS =

18

DS =

6

>>> END of arrays\_final\_scores\_9b.m <<<