Joanne Kwon

PID #: A15359545

**Section A**

**QUESTION #1**

* Average (mean) age of participants in 1950: 46.0650
  + Code: mean(data(:,2))
* Median age of participants in 1950: 47
  + Code: median(data(:,2))
* Minimum height of the participants in 1950: 61 inches
  + Code: min(data(:,6))
* Maximum height of participants in 1950: 75 inches
  + Code: max(data(:,6))

**QUESTION #2**

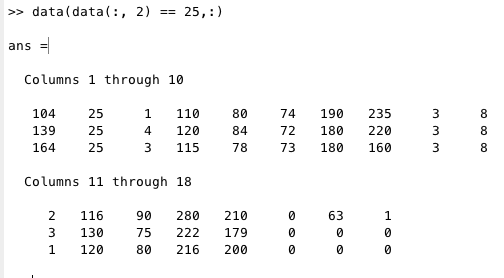
* Most weight gained of participant between 1963 and 1950: -34
* Most weight lost of participant between 1963 and 1950: 62

Code: data(:,15)-data(:,7)

**QUESTION #3**

Code: data(data(:, 2) == 25,:)

* 104 25 1 110 80 74 190 235 3 8 2 116 90 280 210 0 63 1
* 139 25 4 120 84 72 180 220 3 8 3 130 75 222 179 0 0 0
* 164 25 3 115 78 73 180 160 3 8 1 120 80 216 200 0 0 0

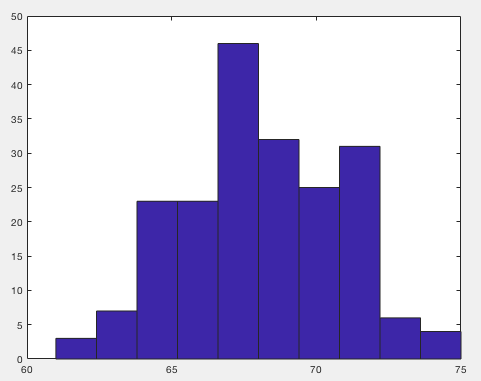


**QUESTION #4**

* Average weight difference of those who died: -4.9532
  + Code: mean(data(data(:, 18)==1, 15))-mean(data(data(:, 18)==1, 7))
* Average weight difference of those who didn’t die: 1.3015
  + Code: mean(data(data(:, 18)==0, 15))-mean(data(data(:, 18)==0, 7))
* Average age of those who died: 52.3125
  + Code: mean(data(data(:,18)==1,2))
* Average age of those who didn’t die: 43.1250
  + Code: mean(data(data(:,18)==0,2))
* When comparing the average weight difference and the average age of those who died versus those who didn’t die, age correlates better with whether a participant lived or died.

**QUESTION #5**

Code: hist(data(:,6))



**Section B**

**QUESTION #6**

1. Average (mean) for cell sizes of benign instances: 67.7273

Code: mean(data(data(:,11)== 2, 6))

Standard deviation for cell sizes of benign instances: 2.6882

Code: std(data(data(:,11)== 2, 6))

1. Average (mean)for cell sizes of malignant instances: 68.2581

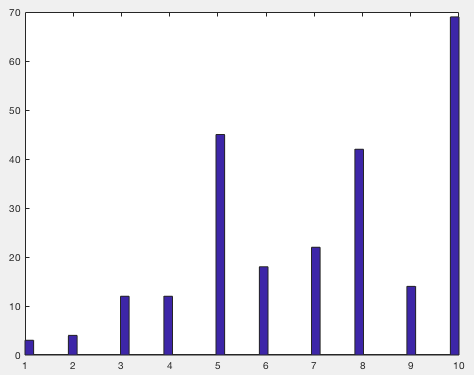
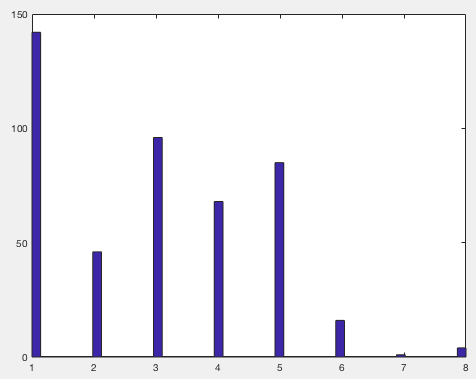
Code: mean(data(data(:,11)== 4, 6))

Standard deviation for cell sizes of malignant instances: 2.7443

Code: std(data(data(:,11)== 4, 6))

**QUESTION #7**

Benign Malignant

1. 
2. Clump thickness (benign)

Code: hist(M(M(:,11)==2,2),50)

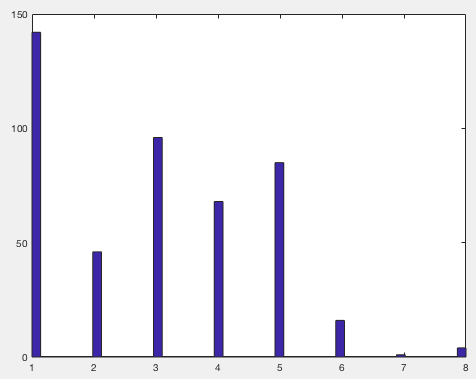
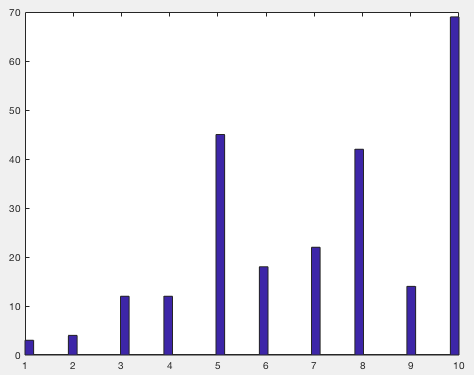
1. Clump thickness (malignant)

Code: hist(M(M(:,11)==4,2),50)

1. The benign histogram has thicker clumps when there are lower id numbers (ex: 1), while the malignant histogram has thicker clumps when there are higher id numbers (ex: 10). The benign and malignant histogram, in a sense, are reflected oppositely of one another.

**Question 16**

y-axis Benign y-axis Malignant

1.  x-axis x-axis

x-axis = number of patients

y-axis = clump thickness

1. Mean of clump thickness (benign): 2.9563

Code: mean(M(M(:,11)==2,2))

Mean of clump thickness (malignant): 7.1950

Code: mean(M(M(:,11)==4,2))

SD of clump thickness (benign): 1.6743

Code: std(M(M(:,11)==2,2))

SD of clump thickness (malignant): 2.4288

Code: std(data(data(:,11)==4,2))

* The mean of the clump thickness of benign is higher when the id numbers are lower, whereas the clump thickness of malignant is lower when the id numbers are lower. Overall, the benign graph shows a gradual decrease of clump thickness as the id numbers increase. On the other hand, the malignant graph shows a gradual increase of clump thickness as the id numbers increase.

**QUESTION #8**

1. 1. Code: z\_clump = zscore(M(:,2))

2. Code: z\_size = zscore(M(:,3))

3. Code: z\_shape = zscore(M(:,4))

4. Code: z\_adhesion = zscore(M(:,5))

5. Code: z\_single = zscore(M(:,6))

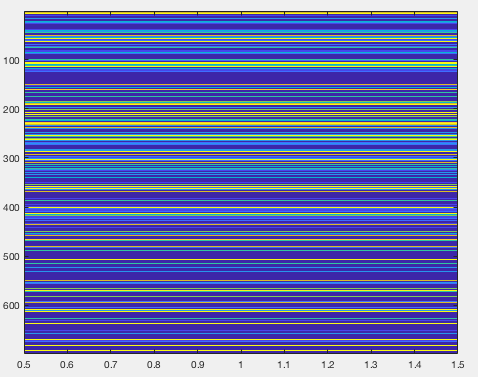
6. Code: z\_nuclie = zscore(M(:,7))

7. Code: z\_chromatin = zscore(M(:,8))

8. Code: z\_normal = zscore(M(:,9))

9. Code: z\_mitoses = zscore(M(:,10))

b) Code: imagesc(M(:,3));

* 

**Section C**

**Question #9**

a) Average of Proanthocyanins (region A): 1.8993

Code: mean(W(W(:,1)==1,10))

Standard deviation of Proanthocyanins (region A): 0.4121

Code: std(W(W(:,1)==1,10))

Average of Proanthocyanins (region B): 1.6303

Code: mean(W(W(:,1)==2,10))

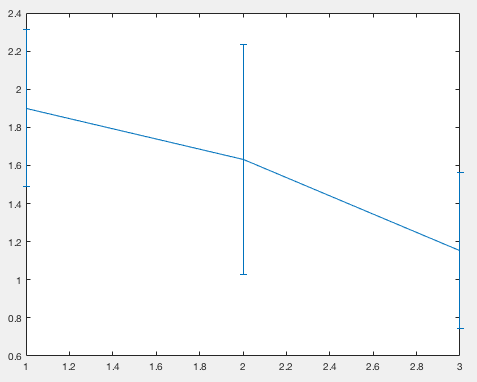
Standard deviation of Proanthocyanins (region B): 0.6021

Code: std(W(W(:,1)==2,10))

Average of Proanthocyanins (region C): 1.1535

Code: mean(W(W(:,1)==3,10))

Standard deviation of Proanthocyanins (region C): 0.4088

Code: std(W(W(:,1)==3,10))

b) Code: xlim([0.5 3.5])

x= 1:1:3;

y = [1.8993, 1.6303, 1.1535];

err = [0.4121, 0.6021, 0.4088];

errorbar(x,y,err)

c) Flavanoids

* Code: xlim([0.5 3.5])

x= 1:1:3;

y = [2.9824 2.0808 0.7815];

err = [0.3975 0.7057 0.2935];

errorbar(x,y,err)

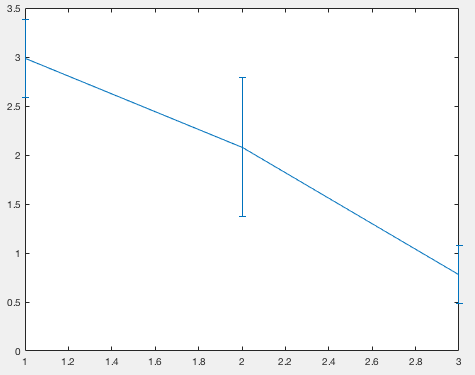
Total Phenols

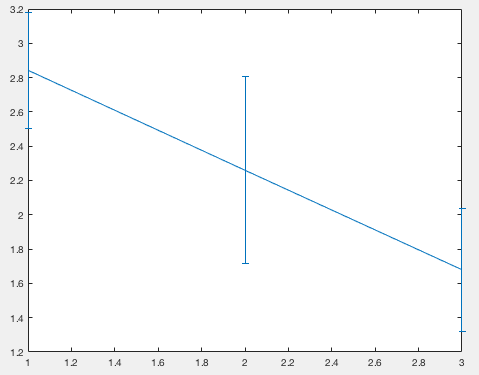
* Code: xlim([0.5 3.5])

x= 1:1:3;

y = [2.8402 2.2589 1.6788];

err = [0.3390 0.5454 0.3570];

errorbar(x,y,err)



Total Phenols Flavanoids

**Question #10**

1. 1. Code: z\_alcohol = zscore(W(:,2));

2. Code: z\_malic = zscore(W(:,3));

3. Code: z\_ash = zscore(W(:,4));

4. Code: z\_alcalinity = zscore(W(:,5));

5. Code: z\_magnesium = zscore(W(:,6));

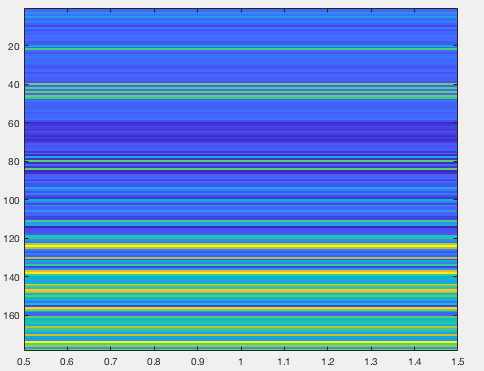
6. Code: z\_total = zscore(W(:,7));

7. Code: z\_flavanoids = zscore(W(:,8));

8. Code: z\_nonflavanoid = zscore(W(:,9));

9. Code: z\_proanthocyanins = zscore(W(:,10));

1. Code: imagesc(W(:,3));



**Section D**

**QUESTION #11**

1. Code: %Joanne Kwon

%PID #: A15359545

max(Q(Q(:,1) <= 300, 2));

1. Code: Q(:,1) = Q(:,1);

Output:

14.2300 4.0000

13.2000 4.0000

13.1600 1.0000

14.3700 4.0000

13.2400 3.0000

14.2000 1.0000

14.3900 2.0000

14.0600 3.0000

14.8300 4.0000

13.8600 4.0000

14.1000 1.0000

14.1200 4.0000

13.7500 4.0000

14.7500 2.0000

14.3800 4.0000

13.6300 1.0000

14.3000 2.0000

13.8300 4.0000

14.1900 4.0000

13.6400 4.0000

14.0600 3.0000

12.9300 1.0000

13.7100 4.0000

12.8500 4.0000

13.5000 3.0000

13.0500 4.0000

13.3900 3.0000

13.3000 2.0000

13.8700 3.0000

14.0200 1.0000

13.7300 3.0000

13.5800 1.0000

13.6800 2.0000

13.7600 1.0000

13.5100 1.0000

13.4800 4.0000

13.2800 3.0000

13.0500 2.0000

13.0700 4.0000

14.2200 1.0000

13.5600 2.0000

13.4100 2.0000

13.8800 4.0000

13.2400 4.0000

13.0500 1.0000

14.2100 2.0000

14.3800 2.0000

13.9000 3.0000

14.1000 3.0000

13.9400 4.0000

13.0500 2.0000

13.8300 3.0000

13.8200 3.0000

13.7700 1.0000

13.7400 1.0000

13.5600 2.0000

14.2200 4.0000

13.2900 2.0000

13.7200 3.0000

12.3700 1.0000

12.3300 4.0000

12.6400 2.0000

13.6700 3.0000

12.3700 3.0000

12.1700 4.0000

12.3700 4.0000

13.1100 3.0000

12.3700 1.0000

13.3400 1.0000

12.2100 2.0000

12.2900 4.0000

13.8600 2.0000

13.4900 4.0000

12.9900 1.0000

11.9600 4.0000

11.6600 2.0000

13.0300 1.0000

11.8400 2.0000

12.3300 3.0000

12.7000 2.0000

12.0000 2.0000

12.7200 4.0000

12.0800 3.0000

13.0500 3.0000

11.8400 4.0000

12.6700 2.0000

12.1600 4.0000

11.6500 4.0000

11.6400 2.0000

12.0800 3.0000

12.0800 1.0000

12.0000 1.0000

12.6900 3.0000

12.2900 4.0000

11.6200 4.0000

12.4700 1.0000

11.8100 3.0000

12.2900 2.0000

12.3700 1.0000

12.2900 2.0000

12.0800 1.0000

12.6000 4.0000

12.3400 2.0000

11.8200 3.0000

12.5100 1.0000

12.4200 3.0000

12.2500 2.0000

12.7200 3.0000

12.2200 3.0000

11.6100 3.0000

11.4600 2.0000

12.5200 1.0000

11.7600 1.0000

11.4100 4.0000

12.0800 1.0000

11.0300 4.0000

11.8200 3.0000

12.4200 4.0000

12.7700 1.0000

12.0000 2.0000

11.4500 1.0000

11.5600 4.0000

12.4200 1.0000

13.0500 4.0000

11.8700 4.0000

12.0700 4.0000

12.4300 1.0000

11.7900 2.0000

12.3700 2.0000

12.0400 4.0000

12.8600 2.0000

12.8800 4.0000

12.8100 1.0000

12.7000 2.0000

12.5100 1.0000

12.6000 1.0000

12.2500 4.0000

12.5300 3.0000

13.4900 3.0000

12.8400 1.0000

12.9300 4.0000

13.3600 3.0000

13.5200 2.0000

13.6200 3.0000

12.2500 2.0000

13.1600 1.0000

13.8800 1.0000

12.8700 1.0000

13.3200 1.0000

13.0800 1.0000

13.5000 2.0000

12.7900 1.0000

13.1100 4.0000

13.2300 4.0000

12.5800 2.0000

13.1700 2.0000

13.8400 2.0000

12.4500 4.0000

14.3400 2.0000

13.4800 1.0000

12.3600 4.0000

13.6900 2.0000

12.8500 1.0000

12.9600 2.0000

13.7800 1.0000

13.7300 1.0000

13.4500 4.0000

12.8200 4.0000

13.5800 3.0000

13.4000 1.0000

12.2000 1.0000

12.7700 2.0000

14.1600 4.0000

13.7100 1.0000

13.4000 1.0000

13.2700 1.0000

13.1700 3.0000

14.1300 3.0000

1. Code: for i = 1:178

sum(Q(:,1).\*Q(:,2));

return

end

Output: 5.7568e+03

1. Code: sum(Q(:,2),1);

Output: 443

Full Script:

%Joanne Kwon

%PID #: A15359545

sum(Q(:,2),1) %output 3

max(Q(Q(:,1) <= 300, 2)) %11a

Q(:,1) = Q(:,1) %output 1

for i = 1:178 %output 2

sum(Q(:,1).\*Q(:,2))

return

end