Homework 6

Prepare your answers as a **single PDF file**.
Group work: You may work in groups of 1-3. Include all group member names in the PDF file. You may work with students in both sections (375-01, -02). Only one person in the group should submit to Canvas.
Due: check on Canvas.

Bryce Lin, Serop Kelkelian

1. Consider the toy dataset below which shows if 4 subjects have diabetes or not, along with two diagnostic measurements. (Note: do **NOT** write any code for this problem. The answers are to be computed by hand.)

Preg	**BP**	**HasDiabetes**	**Preg.Norm**	**BP.Norm**
2	74	No	0.5	1
3	58	Yes	1.0	0.2
2	58	Yes	0.5	0.2
1	54	No	0	0
2	70	?	0.5	0.8

a. Which variable is the “Class” variable?
HasDiabetes

b. Normalize the Preg and BP values by scaling the minimum-maximum range of each column to 0-1. Fill in the empty columns in the table.
Norm = (x - min)/(max-min)

c. Predict whether a subject with Preg=2, BP=70 will have diabetes using the 1-NN algorithm and
   i. Using Euclidean distance on the original variables
   ii. Using Euclidean distance on the normalized variables

Preg	BP	Euclidean Distance	HasDiabetes	Preg.Norm	BP.Norm	Euclidean Distance
2	74	**4**	No	0.5	1	**0.2**
3	58	12.04	Yes	1.0	0.2	0.78
2	58	12	Yes	0.5	0.2	0.6
1	54	16.03	No	0	0	0.94
2	70		?	0.5	0.8	

For each of these cases, give the nearest distance, nearest neighbor (e.g., "Row 1" or "Row 2"), and prediction.

Both of them are Row 1, so prediction is No.

2. The `pima-indians-diabetes-resampled.csv` file on Canvas contains records indicating whether the subjects have diabetes or not, along with certain diagnostic measurements. All subjects are of Pima Indian heritage and this dataset is called the Pima Indian Diabetes Database[1]. The goal is to see if it is possible to predict if a subject has diabetes given some of the diagnostic measurements. (**Note: this problem is an extension of the classwork assignment; R code from the class is also posted on Canvas.**)

a. Read the data file [code]

```
library(tidyverse)
diabetes <- read.csv('pima-indians-diabetes-resampled.csv')
```

b. What does "Preg" represent in the dataset? (2-3 sentences. Search for the Pima Indian Diabetes Database online and read up on its background.)

   It represents how many pregnancies the person has on the database. One research I found is about diabetic pregnancy on the offspring among the Pima Indians.

c. 0 values in the Glucose column indicate missing values. Remove rows which contain missing values in the Glucose column. You should have 763 rows. [code]

```
diabetes <- diabetes %>% filter(Glucose != 0)
```

d. Create three new columns/variables which are the normalized versions of Preg, Pedigree, and Glucose columns, scaling the minimum-maximum range of each column

[1] https://github.com/jbrownlee/Datasets/blob/master/pima-indians-diabetes.names

to 0-1 (you can use the code developed in class). [code]

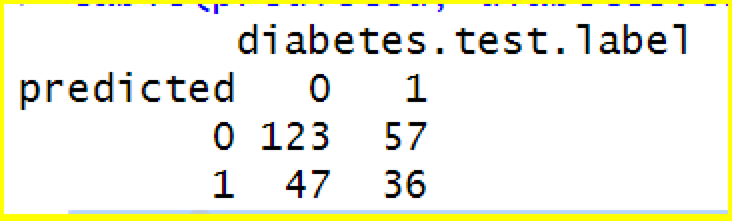
```
diabetes <- diabetes %>% mutate(Preg.norm = (Preg - min(Preg)) /
(max(Preg)-min(Preg))) %>% mutate(Glucose.norm = (Glucose - min(Glucose)) /
(max(Glucose)-min(Glucose))) %>% mutate(Pedigree.norm = (Pedigree -
min(Pedigree)) / (max(Pedigree)-min(Pedigree)))
```

e. Split the dataset into train and test datasets with the *first 500 rows* for training, and the remaining rows for test. Do NOT randomly sample the data (though resampling is usually done, this hw problem does not use this step for ease of grading).

```
library(class)
```

f. Train and test a k-nearest neighbor classifier with the dataset. *Consider only the normalized Preg and Pedigree columns*. Set k=1. What is the error rate (number of misclassifications)? [code, error rate]

```
diabetes.train.feature <- diabetes[1:500,c(10, 7)]
diabetes.train.label <- diabetes[1:500, 9]
diabetes.test.feature <- diabetes[501:nrow(diabetes), c(10,7)]
diabetes.test.label <- diabetes[501:nrow(diabetes), 9]
predicted <- knn(train = diabetes.train.feature, test = diabetes.test.feature,
cl=diabetes.train.label, k = 1)
table(predicted, diabetes.test.label)
```

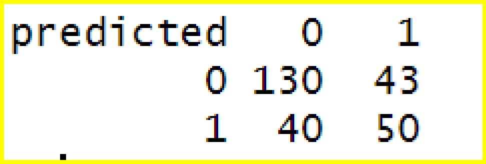


```
          diabetes.test.label
predicted   0   1
        0 123  57
        1  47  36
```

Error Rate = 39.54%

g. Repeat part (f) but *consider the normalized Preg, Pedigree, and Glucose columns*. Set k=1. What is the error rate? Will the error rate always decrease with a larger number of features? Why or why not: answer in 2-3 sentences? [code, error rate, answer]

```
diabetes.train.feature <- diabetes[1:500,c(10, 7, 2)]
diabetes.train.label <- diabetes[1:500, 9]
diabetes.test.feature <- diabetes[501:nrow(diabetes), c(10,7,2)]
diabetes.test.label <- diabetes[501:nrow(diabetes), 9]
predicted <- knn(train = diabetes.train.feature, test = diabetes.test.feature,
cl=diabetes.train.label, k = 1)
table(predicted, diabetes.test.label)
```



```
predicted   0   1
        0 130  43
        1  40  50
```

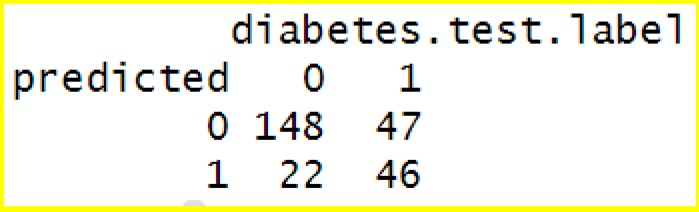
Error Rate = 31.56%

Yes, it will decrease with larger number of feature, but it need to be relative

information to the value that we are trying to predict. If it is a random variable, it may increase the error rate instead.

h. Repeat part (g) but set k=5. What is the error rate? [code, error rate]

```
predicted <- knn(train = diabetes.train.feature, test = diabetes.test.feature,
cl=diabetes.train.label, k = 5)
table(predicted, diabetes.test.label)
```

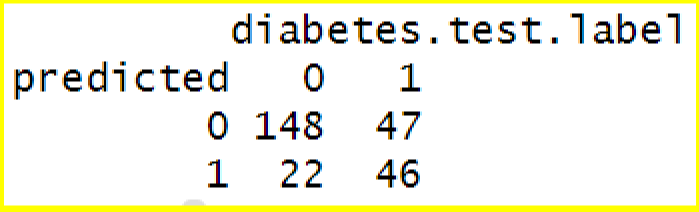


```
          diabetes.test.label
predicted   0   1
        0 148  47
        1  22  46
```

Error Rate: 26.24%

i. Repeat part (h) but set k=11. What is the error rate? Considering your observations from (g)-(i), which is the best value for k? [code, error rate, answer]

```
predicted <- knn(train = diabetes.train.feature, test = diabetes.test.feature,
cl=diabetes.train.label, k = 11)
table(predicted, diabetes.test.label)
```



```
          diabetes.test.label
predicted   0   1
        0 148  47
        1  22  46
```

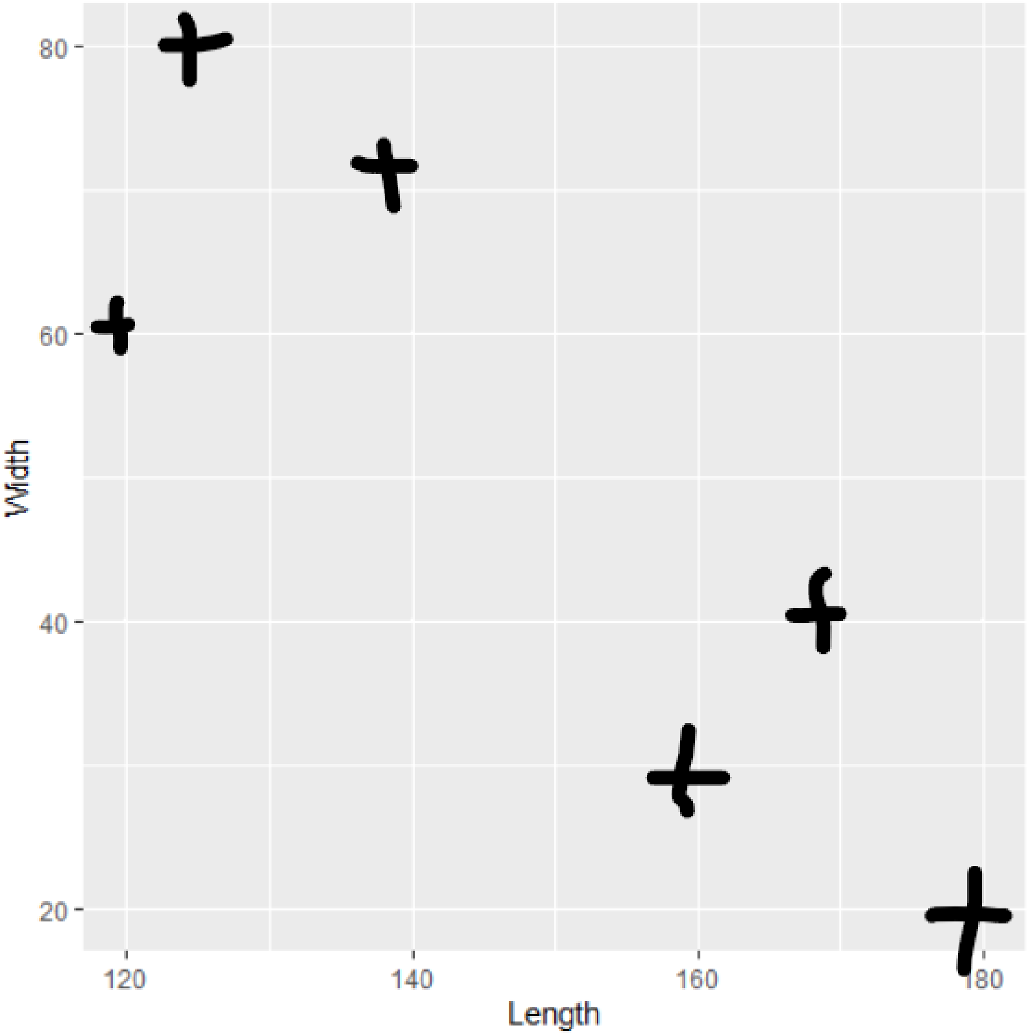
Error Rate: 24.71%

Best Value: 11 is the best value out of the three k values.

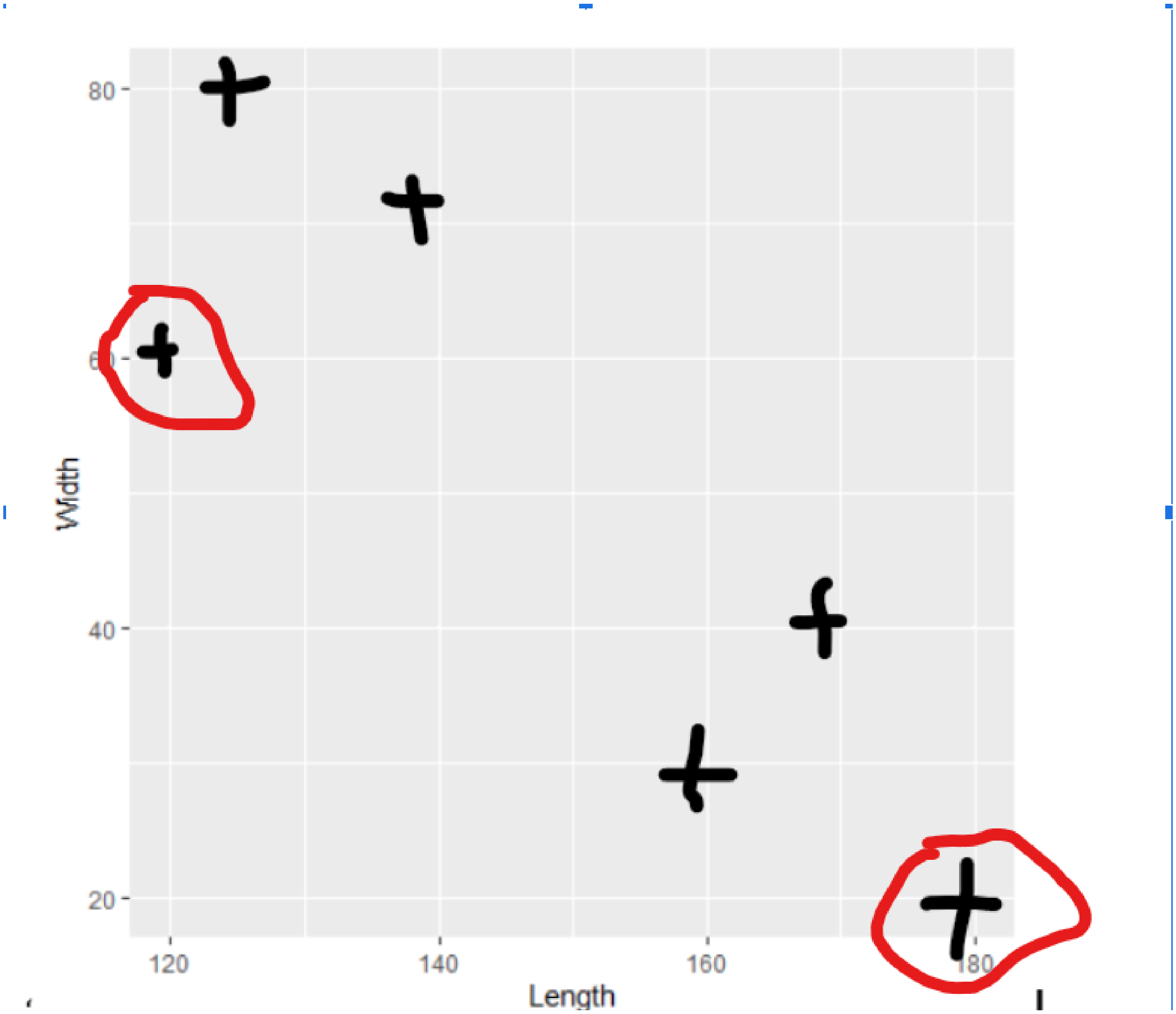
3. Consider the following dataset. (Note: do **NOT** write any code for this problem. The answers are to be computed by hand and marked on the graph. You can even visually guess some of the answers.)

Length	120	140	130	170	160	180
Width	60	70	80	40	30	20

a) Mark the data points on the graph below (*use '+' to indicate each point*).



b) Let k=2. Let one of the two initial centers be (Length=120, Width=60). Select the second center using the **Farthest Distance Heuristic**. Indicate the two centers on the graph (*circle the centers*).



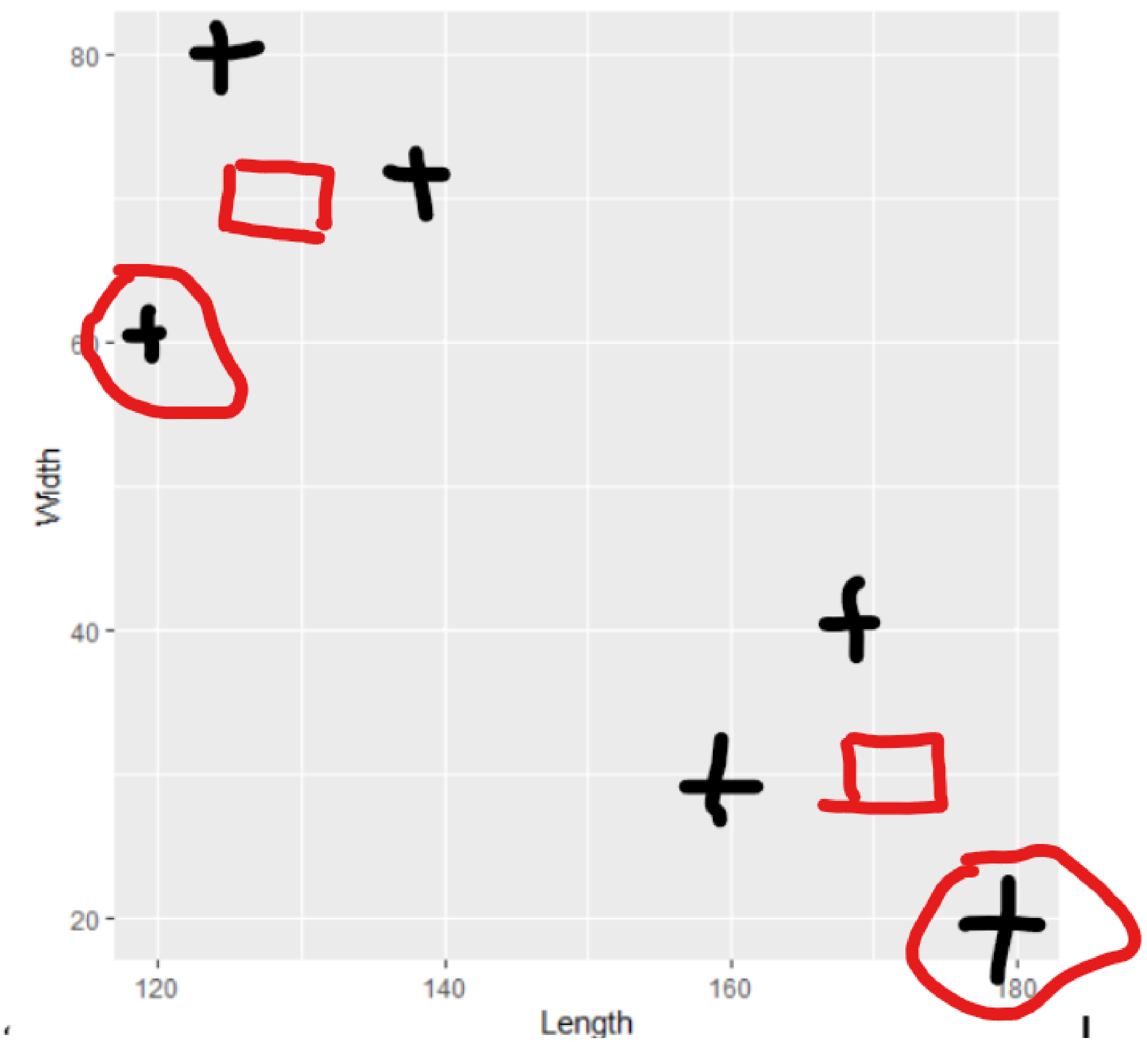
L = 180, W = 20

c) Recompute the centers after the first iteration of the k-means algorithm.

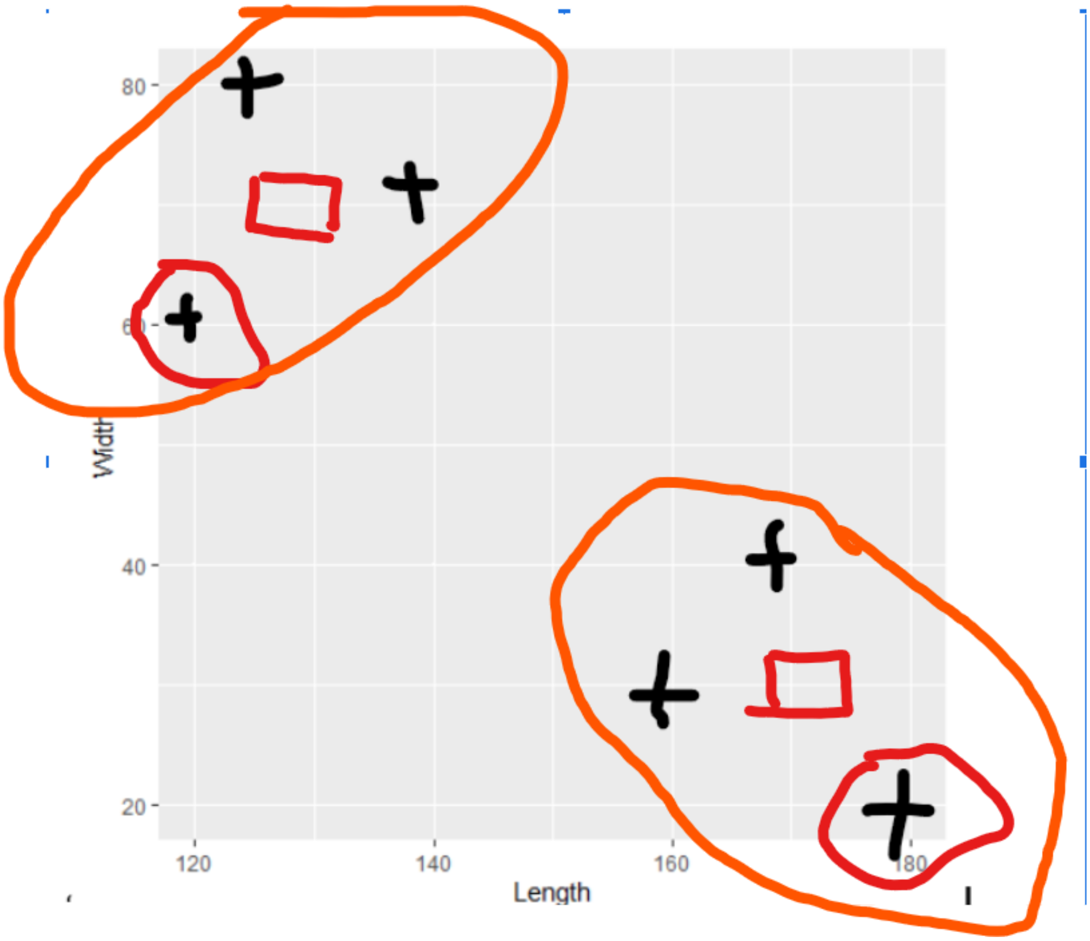
New center 1:__(130, 70)

New center 2:__(170, 30)

Indicate the two new centers on the graph (*mark new centers with squares*).



d) What are the two clusters after this first iteration? *Draw two ovals, each containing all the points in one cluster in the graph above.*



e) Will the k-means algorithm terminate after this first iteration or will it continue? Answer in 1-2 sentences.

No, because the center had converged. All the points are at the correct cluster.

f) If a new point (Length=140, Width=60) is given, to which cluster will it belong?

Will be part of the cluster k.1, centered at (130, 70)

4. Consider the file `breast-cancer-wisconsin.csv` (in the Datasets module on Canvas) which contains "Features computed from a digitized image of a fine needle aspirate (FNA) of a breast mass."[2] The goal is to cluster the data based on the features to distinguish Benign and Malignant cases.

a. Read the data from the file into an object called "mydata". Column 1 ("Code") is the anonymized subject code and will not be used here. Columns 2-10 are the 9 features. Column 11 is the diagnosis: [B]enign or [M]alignant.

   i. How many total cases are there in the data?: **683**
   nrow(bcancer)

   ii. How many [B]enign cases are there in the data?: **444**
   bcancer %>% filter(Class == "B") %>% summarise(n())

[2] Original dataset from Breast Cancer Wisconsin (Diagnostic) Data Set https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

iii. How many [M]alignant cases are there in the data?: 239
bcancer %>% filter(Class == "M") %>% summarise(n())

b. Run k-means clustering using **all the rows** and **only the following features**: **ClumpThickness**, **CellSize**, and **Nuclei**. Use nstart=10.

i. What should be the value of k? **k = 2**

ii. Give R code:**cluster <- bcancer %>% select(ClumpThickness, CellSize, Nuclei) %>% kmeans(centers = 2, nstart = 10)**

c. Evaluation: Compare the resulting clusters with the known diagnosis.

i. Complete the contingency table of your clustering. (Hint: use R's table() function. You can arbitrarily assign cluster 1/2 to Benign/Malignant)

	Cluster 1	Cluster 2
Benign	437	7
Malignant	20	219

ii. Give R code:**table(bcancer$Class, cluster$cluster)**

5. Using the contingency table that you obtained from the previous problem (3.c), calculate the following metrics (consider Malignant as the Positive class): [TOPIC WILL BE COVERED ON 4/11]

1. Accuracy = **(437 + 219) / (437 + 219 + 20 + 7) = 656 / 683 = 96.05%**
2. Error = **1 - 96.05 % = 3.95%**
3. Precision = **219 / (7 + 219) = 96.90%**
4. Recall = **219 / (219 + 20) = 91.63%**
5. F-score = **2 x 0.969 x 0.9163 / (.969 + .9163) = 1.7757894 / 1.8853 = 94.19%**

Consider a “silly” classifier for this problem that makes every prediction as Malignant. Calculate the metrics for this “silly” classifier.

1. Accuracy = **239 / 683 = 34.99%**
2. Error = 1 - **34.99% = 65.01%**
3. Precision = **239 / 683 = 34.99%**
4. Recall = **239 / 239 = 100%**
5. F-score = **2 x 34.99% x 100% / (1 + .3499) = 51.84%**