Keosotra Veng
Stat 6620 Section 2

Homework 2

1. Perform the cancer diagnosis kNN analysis. Produce a report explaining the data, the analysis, and the findings.
    - Organize you report using the Five Steps.
    - Be sure to include:
        1. Show the prediction that the algorithm produced.
        2. Give the Accuracy of the predictions. See Page 318 (or 299).
        3. Include the confusion matrix.

Step 1 – collecting data

We got the data from here:

http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29

Step 2 – exploring, preparing, normalizing the data and creating training and test datasets

- In the data we have 357 observations Benign and 212 observations malignant.

```
table(wbcd$diagnosis)

  B   M
357 212
```

- In total, we have 52.7% of the observations are benign and 37.3% of the observations are malignant

```
round(prop.table(table(wbcd$diagnosis)) * 100, digits = 1)

   Benign Malignant
     62.7      37.3
```

- These are statistical summary of radio mean, area mean, and smoothness mean.

```
summary(wbcd[c("radius_mean", "area_mean", "smoothness_mean")])

  radius_mean       area_mean      smoothness_mean
 Min.   : 6.981   Min.   : 143.5   Min.   :0.05263
 1st Qu.:11.700   1st Qu.: 420.3   1st Qu.:0.08637
 Median :13.370   Median : 551.1   Median :0.09587
 Mean   :14.127   Mean   : 654.9   Mean   :0.09636
 3rd Qu.:15.780   3rd Qu.: 782.7   3rd Qu.:0.10530
 Max.   :28.110   Max.   :2501.0   Max.   :0.16340
```

- We apply min-max normalization method to the dataset excluding the first column.

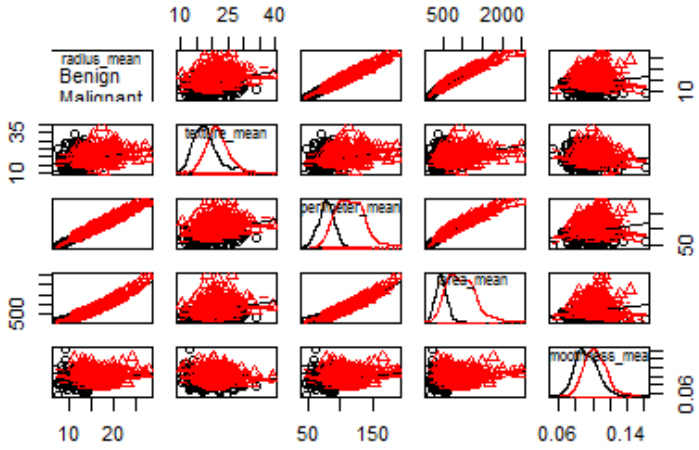
```
normalize <- function(x) {
```

```
  return ((x - min(x)) / (max(x) - min(x)))}

wbcd_n <- as.data.frame(lapply(wbcd[2:31], normalize))
```

-We have 569 observations, now we set 469 observations as train data and the rest is for test data.

```
wbcd_train <- wbcd_n[1:469, ]

wbcd_test <- wbcd_n[470:569, ]
```

-We use scatterplotMatrix to make a matrix of graphs with radius mean, texture mean, perimeter mean, area mean, and smoothness mean. Red refers to malignant and black refers to benign.



Step 3 - training a model on the data and classifying

-we train our model on the data and predict whether the diagnose is benign or malignant

wbcd_test_pred <- knn(train = wbcd_train, test = wbcd_test,

cl = wbcd_train_labels, k = 21)

Step 4 - evaluating model/algorithm performance

With K = 21,

We predict 61 out of 61 that the result is benign.

We predict 37 out of 39 that the result is malignant, so there are two wrong predictions that we predicted benign when the actual result is malignant which is a false negative.

```
Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  100
```

wbcd_test_labels	wbcd_test_pred: Benign	Malignant	Row Total
Benign	61	0	61
	1.000	0.000	0.610
	0.968	0.000	
	0.610	0.000	
Malignant	2	37	39
	0.051	0.949	0.390
	0.032	1.000	
	0.020	0.370	
Column Total	63	37	100
	0.630	0.370	

Step 5 - improving model/algorithm performance

-We tried normalizing our data by using z score method but this didn't improve the result, actually it got worse.

-We try varying K between 1, 5, 11, 15, 21, 27. With assumption that it's better to have false positive rather than false negative, K = 1 have the best prediction.

2. Find an interesting dataset from the UCI ML Repository that is appropriate for applying the kNN algorithm and try to load the data into R and proceed to classify the data using kNN.

Step 1 – Collecting data

Keosotra Veng
Stat 6620 Section 2

We got the data from here: http://archive.ics.uci.edu/ml/datasets/Iris

Step 2 – exploring, preparing, normalizing the data and creating training and test datasets

- First we need to load the data and randomize the data:

```
iris <- read.csv("iris_data.csv", stringsAsFactors = TRUE)

irisd <- iris[sample(nrow(iris)),]
```

- In the data we have 150 observations. 50 are Iris-setosa and 50 are Iris-virginica and 50 are Iris-versicolor.

```
table(irisd$Class)

Iris-setosa     Iris-versicolor     Iris-virginica
     50              50                  50
```

- In total, we have 52.7% of the observations are benign and 37.3% of the observations are malignant

```
round(prop.table(table(irisd$Class)) * 100, digits = 1)


    Iris-setosa     Iris-versicolor   Iris-virginica
        33.3             33.3              33.3
```

- These are statistical summary of sepal length, sepal width, petal length, and petal width.

```
summary(irisd[c("Sepal_Length", "Sepal_Width", "Petal_Length", "Petal_Width")
])
  Sepal_Length    Sepal_Width     Petal_Length    Petal_Width
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
 Median :5.800   Median :3.000   Median :4.350   Median :1.300
 Mean   :5.843   Mean   :3.054   Mean   :3.759   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```

- We apply min-max normalization method to the dataset excluding the last column which is the column we want to predict.

```
normalize <- function(x) {

  return ((x - min(x)) / (max(x) - min(x)))}

irisd_n <- as.data.frame(lapply(irisd[1:4], normalize))
```

Keosotra Veng
Stat 6620 Section 2

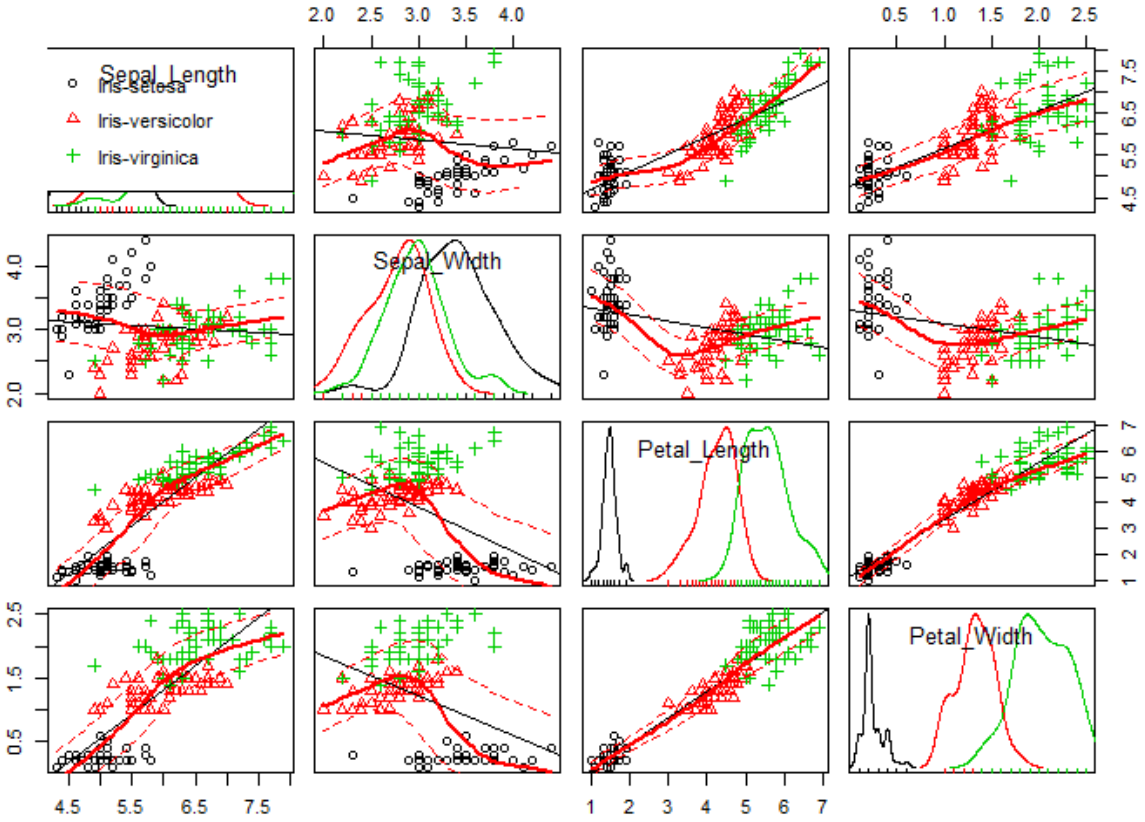
-We have 150 observations, now we set 120 observations as train data and the rest is for test data.

```
irisd_train <- irisd_n[1:120, ]

irisd_test <- irisd_n[121:150, ]
```

-We create label for training and test data for that last column which is the fifth column

```
irisd_train_labels <- irisd[1:120, 5]

irisd_test_labels <- irisd[121:150, 5]
```

-We use scatterplotMatrix to make a matrix of graphs with sepal length, sepal width, petal length, petal width. Black refers to iris setosa and red refers to iris versicolor and green refers to iris virginica.



Step 3 - training a model on the data and classifying

-we train our model on the data and predict whether the class is iris setosa or iris versicolor or iris virginica.

```
irisd_test_pred <- knn(train = irisd_train, test = irisd_test,

                       cl = irisd_train_labels, k = 20)
```

Step 4 - evaluating model/algorithm performance

With K = 20,

Based on the row total, we predict 13 out of 13 that the result is Iris-setosa.

Based on the row total, we predict 8 out of 9 that the result is Iris-versicolor, so there is one wrong prediction which we predicted to be Iris-virginica but the actual result is iris-vericolor.

Based on the row total, we predict 7 out of 8 that the result is Iris-virginica, so there is one wrong prediction that we predicted Iris-versicolor when the actual result is Iris-virginica.

Cell Contents

N	
N / Row Total	
N / Col Total	
N / Table Total	

Total Observations in Table: 30

irisd_test_labels	irisd_test_pred: Iris-setosa	Iris-versicolor	Iris-virginica	Row Total
Iris-setosa	13	0	0	13
	1.000	0.000	0.000	0.433
	1.000	0.000	0.000	
	0.433	0.000	0.000	
Iris-versicolor	0	8	1	9
	0.000	0.889	0.111	0.300
	0.000	0.889	0.125	
	0.000	0.267	0.033	
Iris-virginica	0	1	7	8
	0.000	0.125	0.875	0.267
	0.000	0.111	0.875	
	0.000	0.033	0.233	
Column Total	13	9	8	30
	0.433	0.300	0.267	

Keosotra Veng
Stat 6620 Section 2

Step 5 - improving model/algorithm performance

-We tried normalizing our data by using z score method but this didn't improve the result. It produced the same result.

-We try varying K between 1, 5, 10, 15, 20. We don't see any different with varying K, except when K = 5, we have slightly worse prediction.

3. Do problem 7a,b,c, see page 54, in An Introduction to Statistical Learning.

Compute the Euclidean distance between each observation and the test point,X1 =X2 =X3 =0.

1, Red: sqrt((0-0)^2 + (0-3)^2 + (0-0)^2)= Distance: sqrt(9) = 3
2, Red: sqrt((0-2)^2 + (0-0)^2 + (0-0)^2)= Distance sqrt(4) = 2
3, Red: sqrt((0-0)^2 + (0-1)^2 + (0-3)^2)= Distance: sqrt(10) = 3.162278
4, Green: sqrt((0-0)^2 + (0-1)^2 + (0-2)^2)= Distance: sqrt(5)= 2.236068
5, Green: sqrt((0+1)^2 + (0-0)^2 + (0-1)^2)= Distance: sqrt(2) =1.414214
6, Red: sqrt((0-1)^2 + (0-1)^2 + (0-1)^2)= Distance: sqrt(3)=1.732051

(b) What is our prediction with K = 1? Why?

Since K = 1, then the closest distance would be 1.411 which is Green. So our prediction would be green.

(c) What is our prediction with K = 3? Why?

Since K = 3, the closest distance would be 1.411, 1.732, and 2 which Green, Red, Red respectively. So our prediction would be Red.