STAT 5650 Statistical Learning and Data Mining 1 Spring 2020

Name: Michael Huber

Submission Date: 03/13/2020

Homework 3 (03/13/2020)

100 Points — Due Friday 03/13/2020 (via Canvas by 11:59pm)

(i) **Question 1:** This is a continuation of the analyses on the data for three bird species—(Northern) Flicker, (Mountain) Chickadee, and (Red-naped) Sapsucker—plus a bunch of sites at which none of these species of birds are nesting. In the previous homework you analyzed these data using logistic regression and LDA/QDA; in this question I would like you to apply k-Nearest Neighbor classification to the data and to compare your results with the results for LDA, QDA, and logistic regression.

(a) Apply k-NN classification to the combined dataset for all 3 species using 'Nest' as the response variable. What value of k did you select. Compare the results of this classification with the results you obtained for LDA, QDA, and logistic regression in the previous homework.

```
      predicted
actual  1  2
     0 96 10
     1 12 95
```

```
 [,1]                                [,2]
 "Percent Correctly Classified = " "92.02"
 "Specificity = "                  "86.79"
 "Sensitivity = "                  "97.2"
 "Kappa ="                         "0.8403"
 "AUC= "                           "0.9584"
```

Comparison of Methods:

For this KNN on the nest data setting of the value of k to 2, 3 and 4 all returned similar results, so I set the value to 2 since it is just working to split the data into 2 groups.

Below I have included the scores I got from the three different methods we are comparing to KNN. Looking at the scores below and the output of the KNN model it is clear to see that KNN outperforms the other three models. KNN has an accuracy over 92%, while the other three are all around 80%. The Sensitivity and Specificity of the KNN also outperform the other three models. So in this instance, it appears KNN would be the best model to choose for classification of this model.

- LDA: Accuracy = 78.4% Sensitivity = 74.53% SPecificity = 82.24%

- QDA: Accuracy = 81.22% Sensitivity = 80.19% SPecificity = 82.24%
- Logistic Regression: Accuracy = 79.34% Sensitivity = 76.42% Specificity = 82.24%

(b) Now apply k-NN classification to the three datasets for the individual. What values of k did you select. Compare the results of this classification with the results you obtained for LDA, QDA, and logistic regression in the previous homework.

Chickadee:

```
      predicted
actual  1  2
     0 99  7
     1 15 27
```

```
 [,1]                                [,2]
 "Percent Correctly Classified = " "83.78"
 "Specificity = "                  "84.91"
 "Sensitivity = "                  "80.95"
 "Kappa ="                         "0.6228"
 "AUC= "                           "0.863"
```

Comparison of Methods:

In deciding what value of k to use for this data set I tested numbers from 2 to 10 to see how the accuracy performed. All of the accuracy rates were very similar between all of the groups. But each time I ran the test the accuracy rate would change. But only two values of k which were 8 and 2 got up over 87% The rest staid between 83% and 85%. So I am not sure what value will be printed in my end result when I compile this document. But I chose 8 as the number for k since it seemed to return a value greater than 87% most often. The last test I ran it returned the results listed below with the values for LDA, QDA and Logistic Regression.

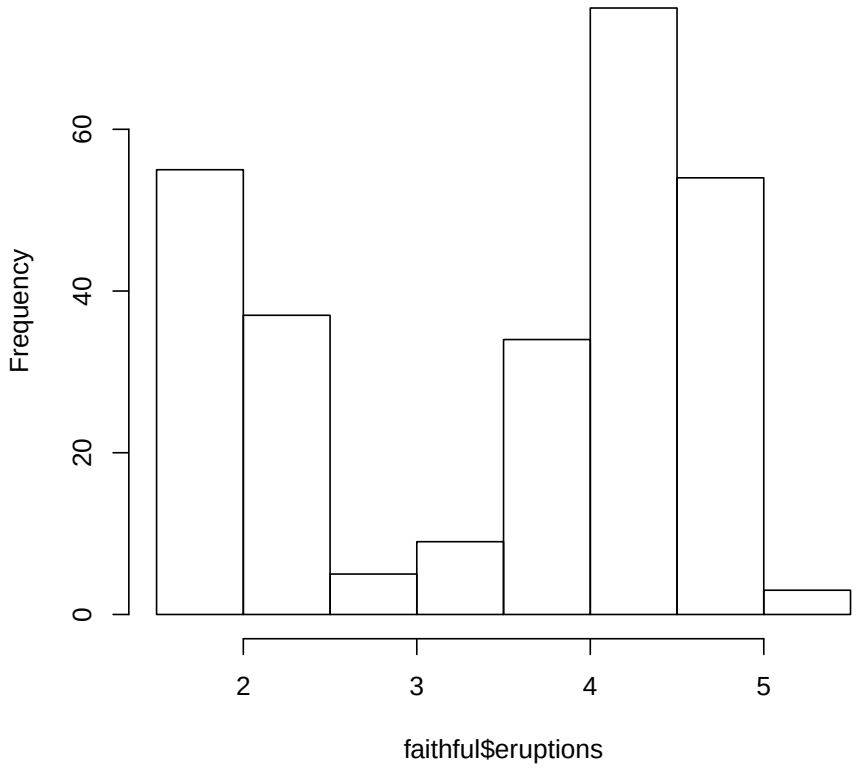
Looking at the results of KNN vs the other three methods it outperforms the other three by around 7% at its highest accuracy. Even at its low accuracy of around 83% it out performs the other three methods. However KNN does under perform in the area of Sensitivity. It comes in about 8% below LDA and QDA and about 15% under Logistic Regression. But I did see on other

runs that the sensitivty did come close to that of LDA and QDA. When KNN had an accuracy of 83.78% it had a Sensitivity of 80.95%. With Specificity however, KNN does much better than the other three. At its high it is around 94% while the other three are in the 60% range. Even when KNN had the lower accuracy rate of 83.78% its Specificity is 84.91%.

- KNN, K=8: Accuracy = 87.84% Sensitivity = 71.43% Specificity = 94.34%
- LDA: Accuracy = 80.41% Sensitivity = 85.85% Specificity = 66.67%
- QDA: Accuracy = 79.73% Sensitivity = 86.79% Specificity = 61.90%
- Logistic Regression: Accuracy = 79.73% Sensitivity = 87.74% Specificity = 59.52%

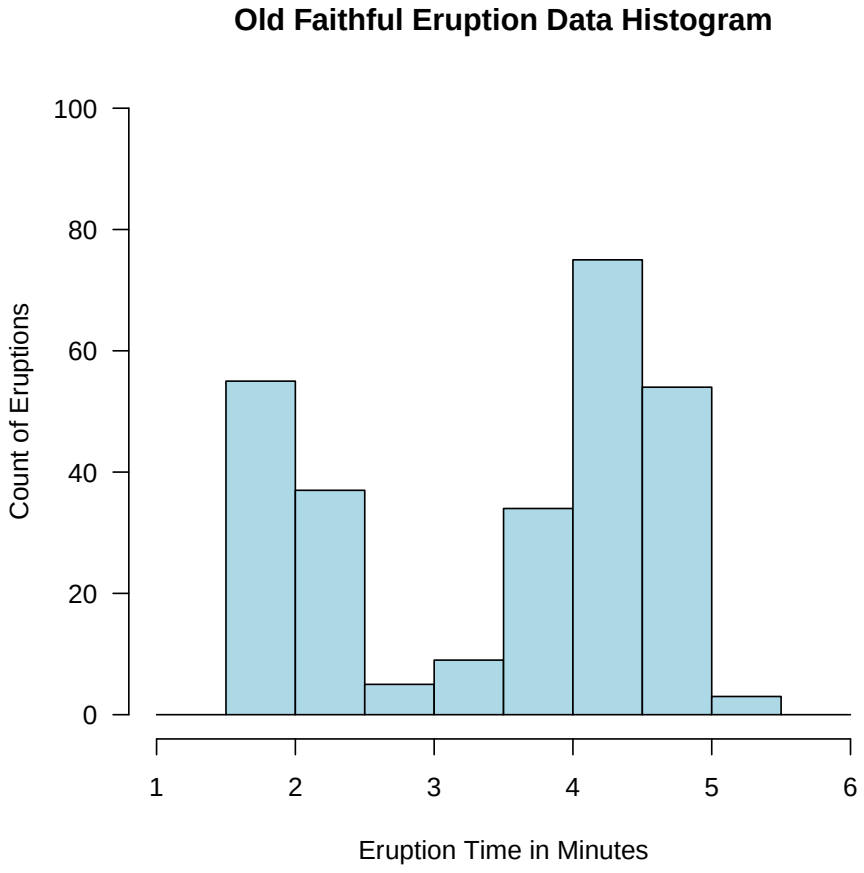
(c) (1 Point) Repeat (b) from above, now using the *hist* function from baseR.

```
> hist(faithful$eruptions)
```



(d) (3 Points) Repeat (c) from above, now using the *hist* function from baseR.

```
> hist(faithful$eruptions,
+      breaks = seq(1, 6, .5),
+      xlab = "Eruption Time in Minutes",
+      ylab = "Count of Eruptions",
+      main = "Old Faithful Eruption Data Histogram",
+      col = "light blue",
+      las = 1,
+      ylim = c(0, 100),
+      xlim = c(1, 6))
```



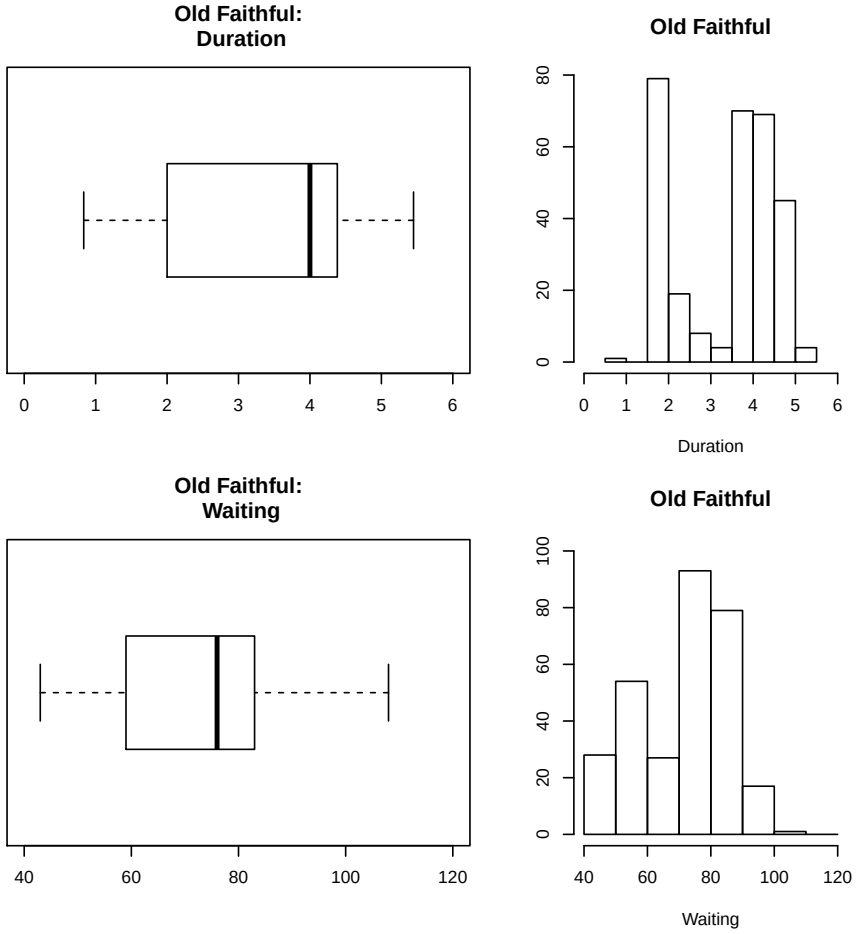
Answer:

- <Answer Part 1>
- <Answer Part 2>

(ii) **Question 2:** <Description of Question 2>

(a) (6 Points) Recreate the graphs (and layout) below using baseR. Use a ruler to check that the width and height proportions in your graphs match the ones I have used. I worked with integer multiples! Include your R code and the resulting graphs. Hint: You can create a new line via `\n` without any extra spaces before/after `\n`.

```
> grid <- matrix(c(1, 1, 1, 2, 2, 3, 3, 3, 4, 4),
+                nrow = 2, ncol = 5, byrow = TRUE)
> layout(grid)
> par(mar = c(4, 3, 4, 2))
> boxplot(geyser$duration,
+         horizontal = TRUE,
+         main = "Old Faithful:\n Duration",
+         ylim = c(0, 6))
> hist(geyser$duration,
+      main = "Old Faithful",
+      xlab = "Duration",
+      ylab = "Count",
+      xlim = c(0, 6))
> boxplot(geyser$waiting,
+         horizontal = TRUE,
+         main = "Old Faithful:\n Waiting",
+         ylim = c(40, 120))
> hist(geyser$waiting,
+      main = "Old Faithful",
+      xlab = "Waiting",
+      ylab = "Count",
+      xlim = c(40, 120),
+      ylim = c(0, 100),
+      breaks = seq(40, 120, by = 10))
```

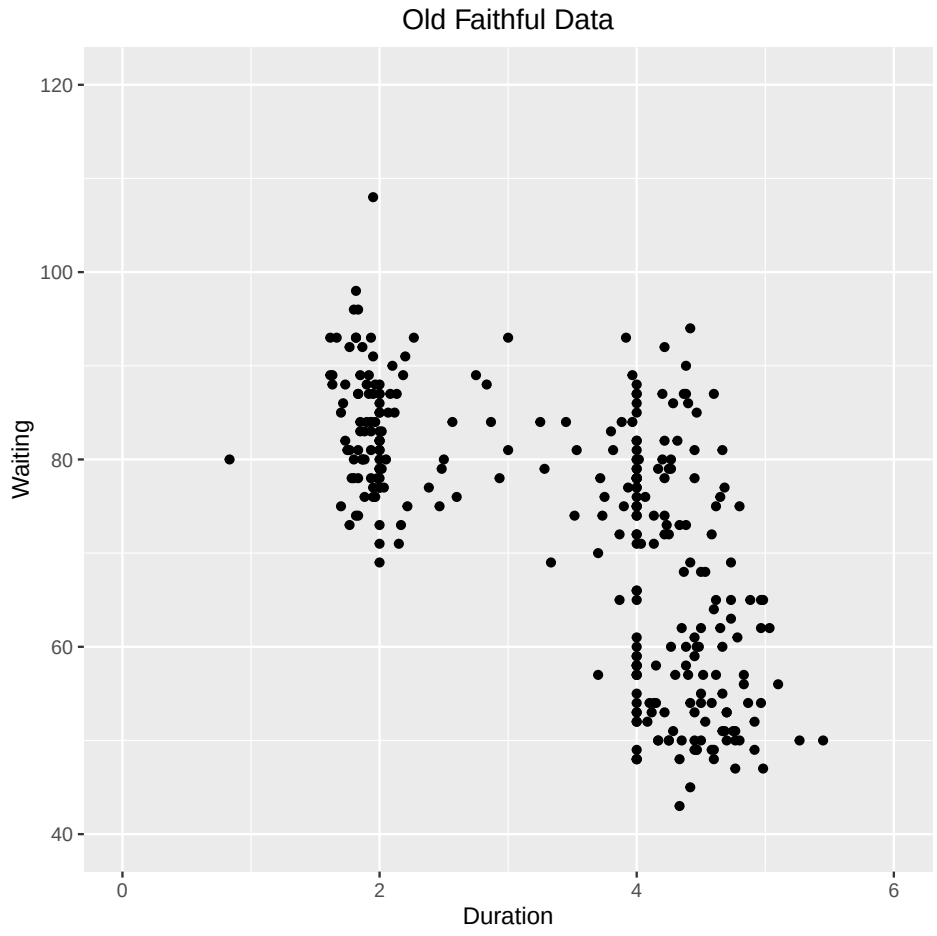


Refernces:

- https://www.statmethods.net/advgraphs/layout.html
- https://stackoverflow.com/questions/31319942/change-the-size-of-a-plot-when-plotting-multiple-plots-in-r
- https://www.youtube.com/watch?v=Z3V4Pbxeahg

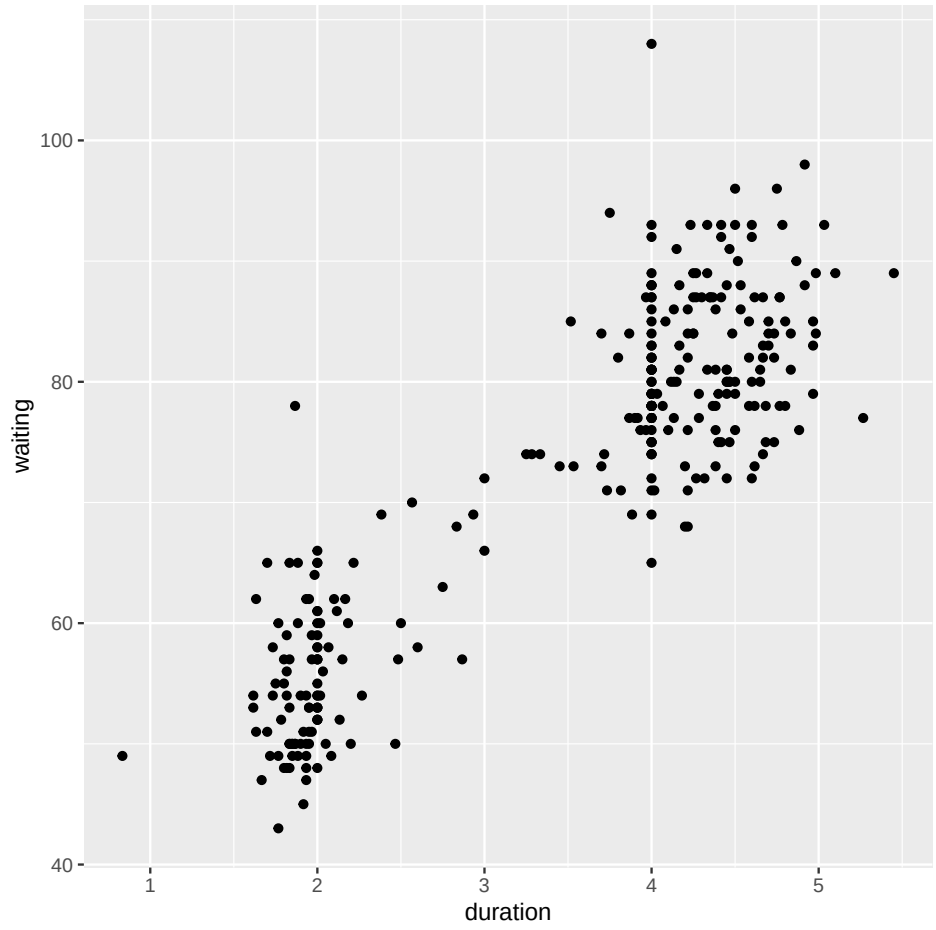
(b) (2 Points) Recreate the graph below using ggplot2. Include your R code and the resulting graph.

```
> ggplot(geyser, aes(x=duration, y=waiting)) +
+   geom_point() +
+   xlab("Duration") +
+   ylab("Waiting") +
+   xlim(0, 6) +
+   ylim(40, 120) +
+   ggtitle("Old Faithful Data") +
+   theme(plot.title = element_text(hjust = 0.5))
```



(c) (2 Points) Doesn't the scatterplot in (b) above look rather different than the scatterplot in Question 1 (j)? Note that the help page for *geyser* states `waiting numeric Waiting time for this eruption` and `The waiting time was incorrectly described as the time to the next eruption in the original files, and corrected for MASS version 7.3-30.` Use this information to create a basic scatterplot for the *geyser* data that matches the overall appearance in Question 1 (j). Include your R code and the resulting graph. No need to refine this scatterplot.

```
> duration <- geyser$duration[1:298]
> waiting <- geyser$waiting[2:299]
> df <- data.frame(duration, waiting)
> ggplot(df, aes(x = duration, y = waiting )) +
+          geom_point()
```



Answer: The geyser data appears to have a negative correlation with three clusters where the faithful data has a positive correlation with 2 clusters.

General Instructions

(i) <Instruction 1>

(ii) <Instruction 2>