# APSTA-GE 2011. Homework #1

January 2025: Yael Beshaw

```
#likely useful libraries and initial seed set.
library(caret)
library(cluster)
library(NbClust)
library(klaR)
library(ggplot2)
library(ggdendro)
library(GGally)
library(e1071)
library(knitr)
library(foreign)
library(gridExtra)
library(palmerpenguins)
set.seed(2011)
```

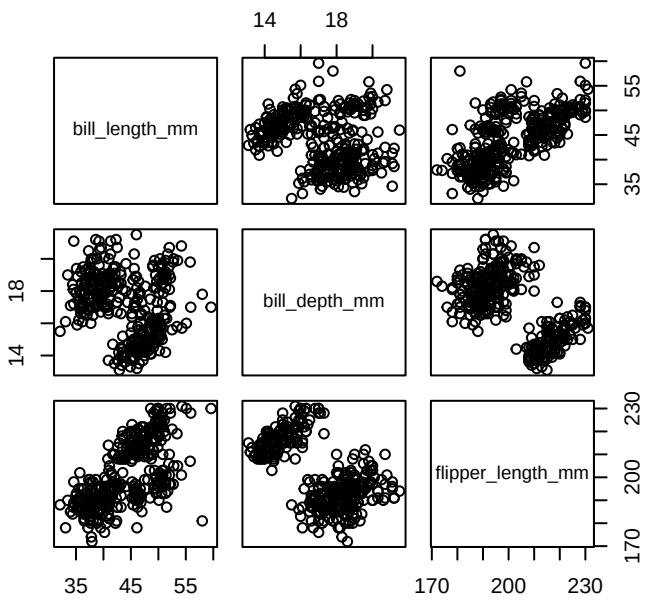
This assignment uses the `penguins` dataset. The written work to hand in consists of selected output from the software package, which you should include in a PDF document as you answer the questions.

```
penguins <- penguins[complete.cases(penguins), ]
```

**Q1**: Generate the matrix of bivariate scatterplots for all possible pairs of the three penguin features (bill length, bill depth, and flipper length) in this dataset. [1pt]

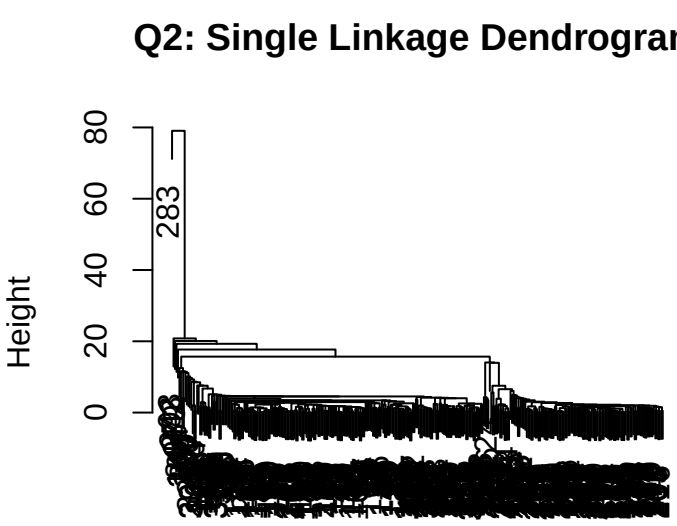
```
pairs(penguins[, 3:5],
      main = "Q1: Bivariate Scatterplot")
```

## Q1: Bivariate Scatterplot




**Q2**: Perform a single linkage hierarchical clustering on the **squared Euclidean distance** between all three feature variables.
Proof of completing this consists of a plot of the dendogram. [1 pt]

```
#create a dataset with the three variables of interest
penguins_3 <- penguins[, c("bill_length_mm", "bill_depth_mm",
                           "flipper_length_mm")]

#squared Euclidean distance
penguins_dist <- dist(penguins_3, method = "euclidean")^2

#single linkage hierarchical clustering
hcl.penguins <- hclust(penguins_dist,meth='single')

#dendrogram
plot(hcl.penguins,
     main= "Q2: Single Linkage Dendrogram",
     xlab= "Penguins",
     sub = "Squared Euclidean Distance")
```



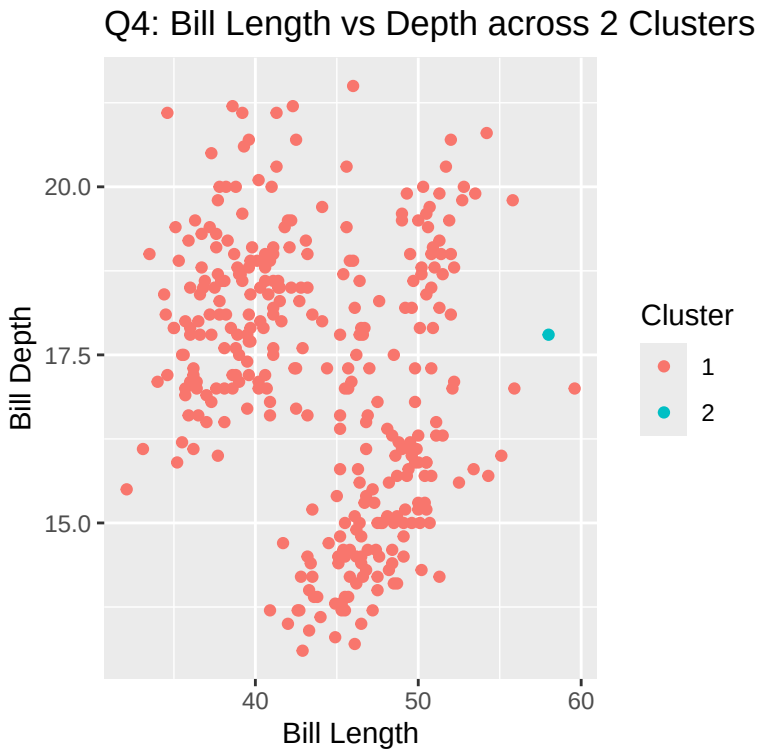

**Q3**: Choose a fixed number of clusters (such as 2, 3, 4, or 5) as your 'solution' to this clustering problem by selecting a cutoff point in the dendogram such that the clusters that would be merged next are very far apart (and thus maybe should not be merged). State that number of clusters as your answer to this question, and assign (save) the cluster labels for that choice to the working dataset. [1pt]

```
check<- factoextra::fviz_cluster(list(data=penguins_3,
                                cluster=cutree(hcl.penguins,2)),
                           choose.vars=c(1:3),main="Single Linkage")

cluster <- cutree(hcl.penguins, 2)
penguins_3$cluster <- cluster
```

Number of Clusters Chosen: 2

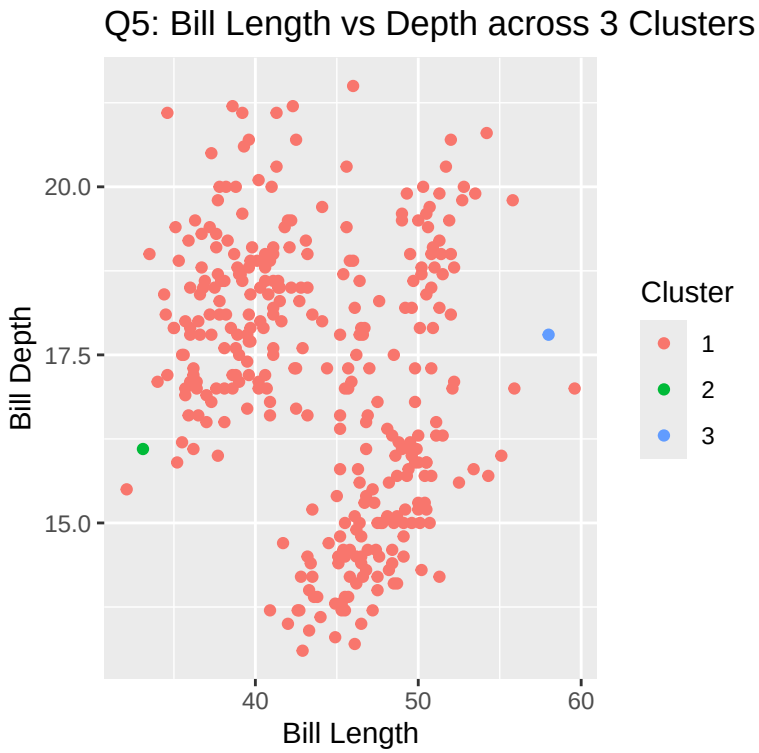
**Q4**: Plot bill length (X-axis) vs. bill depth (Y-axis) and use different colors for markers based on your cluster label variable. [1 pt]

```
ggplot(data = penguins_3, aes(x=bill_length_mm, y= bill_depth_mm,
                              color= factor(cluster))) +
  geom_point() +
  labs(
    title = "Q4: Bill Length vs Depth across 2 Clusters",
    x= "Bill Length",
    y= "Bill Depth",
    color = "Cluster"
  )
```

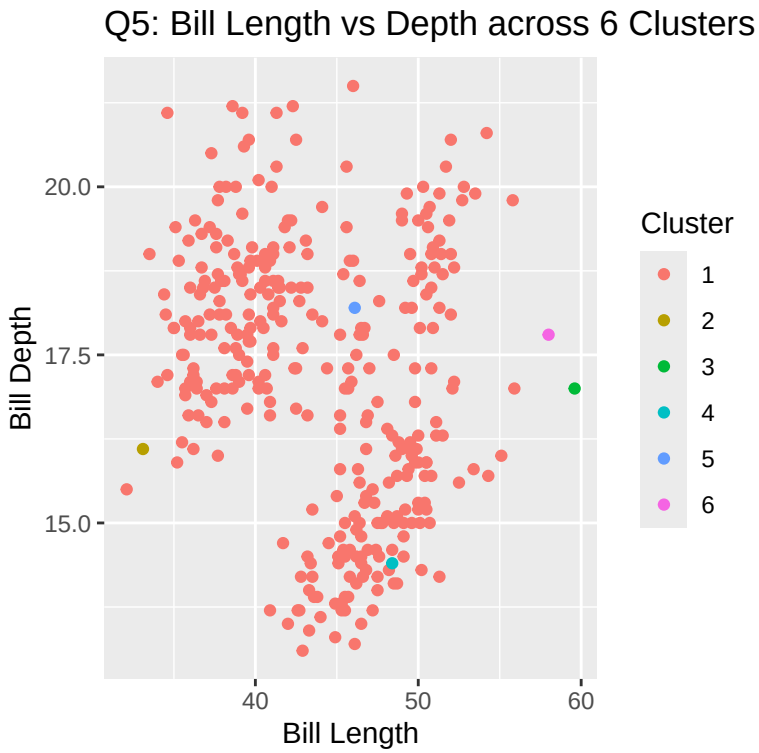



**Q5**: Evaluate the cluster solution by examining the scatterplot in Q4.
No matter what you chose for your first solution, looking at ALL of the solutions for 2, 3, 4, and 5 (even 6) clusters comment on how the additional clusters possibly break up larger clusters with their addition (be especially careful that you do not get misled by the label-switching problem. PLOT AT LEAST 2 ADDITIONAL CLUSTER SOLUTIONS. You are not expecting cluster labels to be optimally matched across cluster solutions. [1 pt each for plot and discussion; 2 pts total]

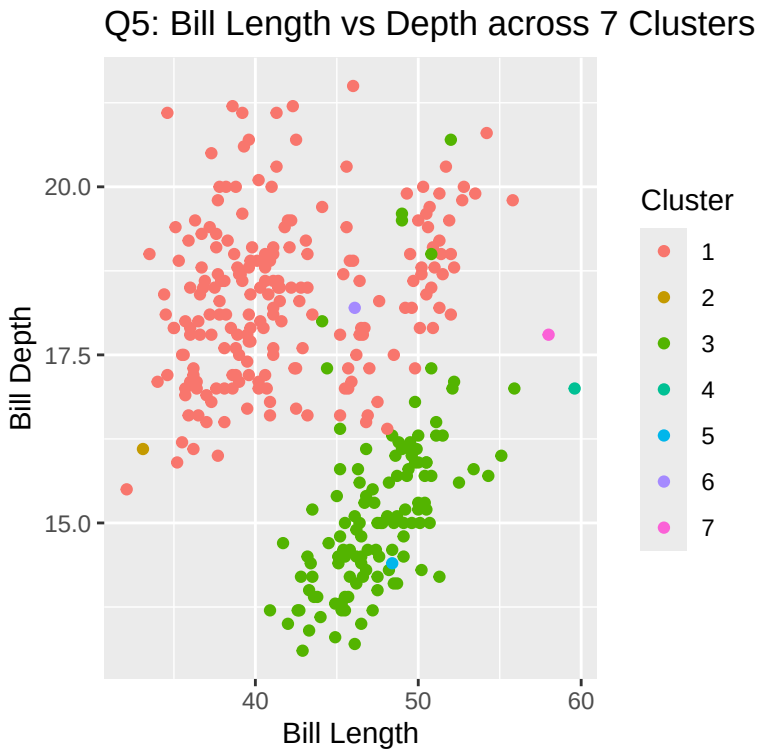
```
#assess all solutions

#k=3
cluster_3 <- cutree(hcl.penguins, k = 3)
penguins_3$cluster_3 <- cluster_3
ggplot(data = penguins_3, aes(x=bill_length_mm, y= bill_depth_mm,
                              color= factor(cluster_3))) +
  geom_point() +
  labs(
    title = "Q5: Bill Length vs Depth across 3 Clusters",
    x= "Bill Length",
    y= "Bill Depth",
    color = "Cluster"
  )
```




```
#k=6
cluster_6 <- cutree(hcl.penguins, k = 6)
penguins_3$cluster_6 <- cluster_6
ggplot(data = penguins_3, aes(x=bill_length_mm, y= bill_depth_mm,
                              color= factor(cluster_6))) +
  geom_point() +
  labs(
    title = "Q5: Bill Length vs Depth across 6 Clusters",
    x= "Bill Length",
    y= "Bill Depth",
    color = "Cluster"
  )
```




```
#k=7
cluster_7 <- cutree(hcl.penguins, k = 7)
penguins_3$cluster_7 <- cluster_7
ggplot(data = penguins_3, aes(x=bill_length_mm, y= bill_depth_mm,
                              color= factor(cluster_7))) +
  geom_point() +
  labs(
    title = "Q5: Bill Length vs Depth across 7 Clusters",
    x= "Bill Length",
    y= "Bill Depth",
    color = "Cluster"
  )
```

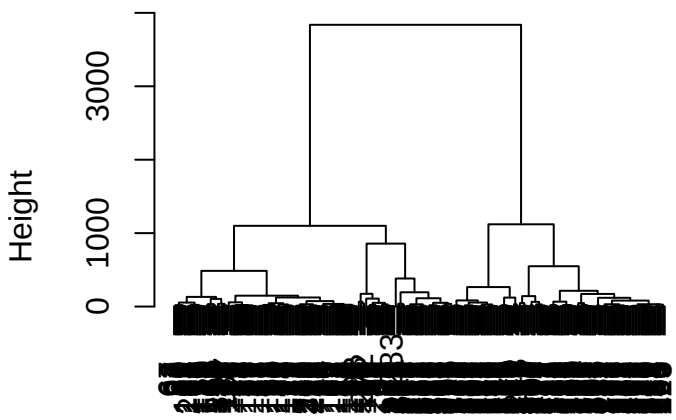



When we started off with two clusters, we see that there is a very large cluster with only one data point assigned to the second cluster. As we add additional clusters, the same pattern occurs where only one data point is assessed to an additional cluster. The overall pattern of the scatterplot remains the same between clusters 2-6, with cluster 1 being the majority and most spread out. To assess if there is a point where this pattern does not occur, I assessed 7 clusters. Here, we finally see the majority split off into two distinct clusters (1(red) and 3(green)).

**Q6**: Redo the analysis by repeating steps 3-5 above, but now for complete and average linkage clustering. [0.5 each for dendrogram and choice; 1 pt for discussion evaluating choice; 2 different methods; 4 pts total]

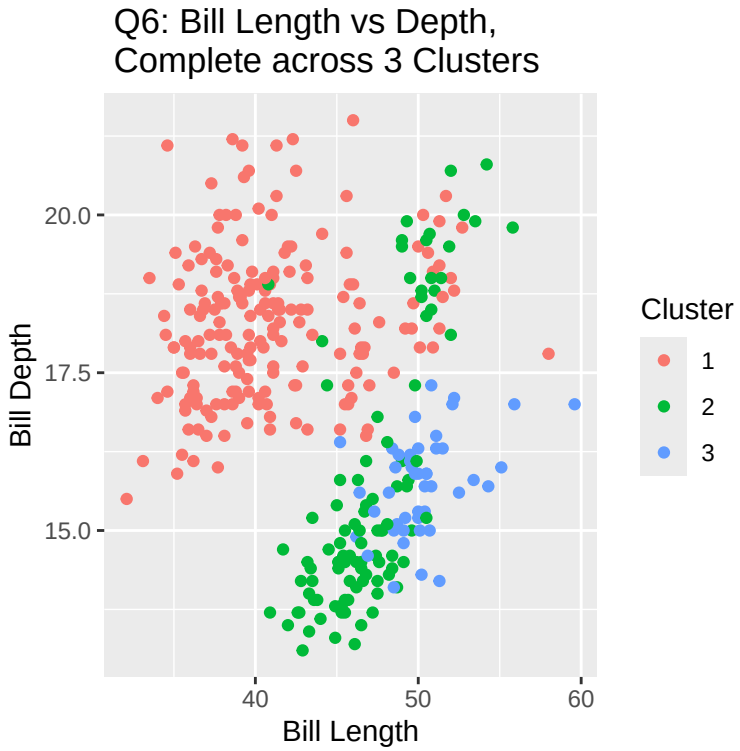
Complete Linkage Clustering

```
#conduct complete linkage clustering
hcl.penguins_complete <- hclust(penguins_dist,meth='complete')

#plot the dendogram
plot(hcl.penguins_complete,
     main= "Complete Linkage Dendrogram",
     xlab= "Penguins",
     sub = "Squared Euclidean Distance")
```

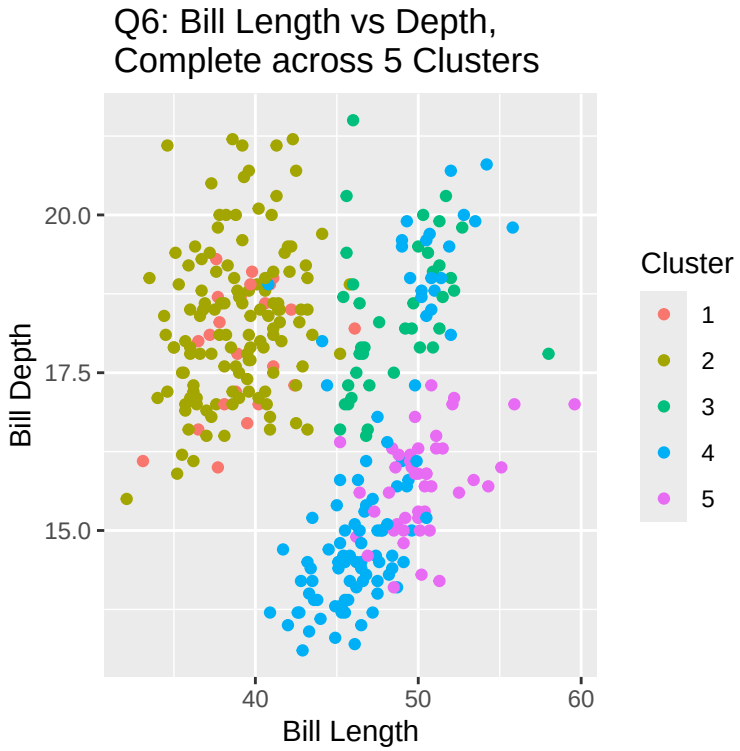



Penguins
Squared Euclidean Distance

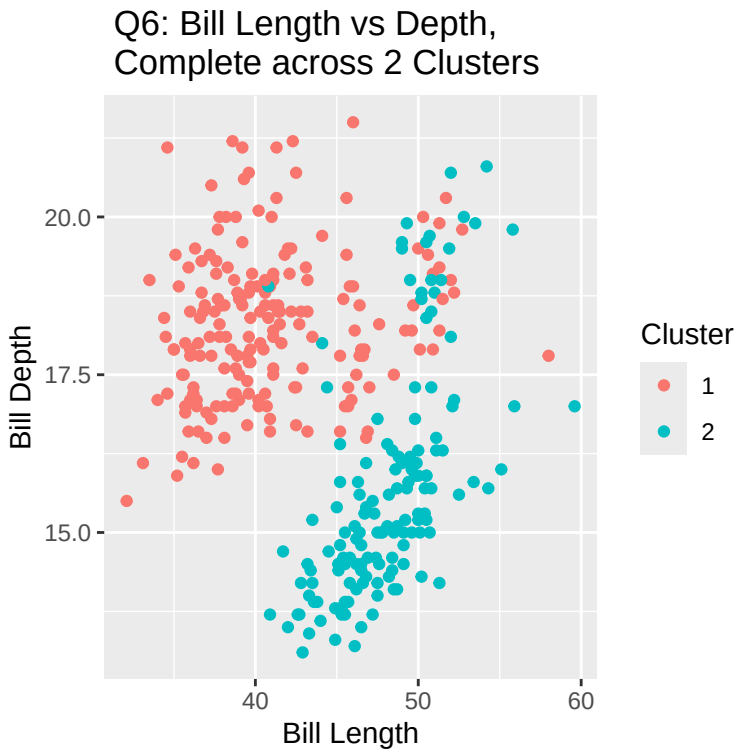
```
#Number of Clusters Chosen: 3
check<- factoextra::fviz_cluster(list(data=penguins_3,
                                      cluster=cutree(hcl.penguins_complete,3)),
                                 choose.vars=c(1:3),main="Single Linkage")
#fviz_cluster shows 3 distinct and appropriate cluster groups

#save the clusters
cluster_complete3 <- cutree(hcl.penguins_complete, 3)
penguins_3$cluster_complete3 <- cluster_complete3

#plot
ggplot(data = penguins_3, aes(x=bill_length_mm, y= bill_depth_mm,
                              color= factor(cluster_complete3))) +
  geom_point() +
  labs(
    title = " Q6: Bill Length vs Depth, \n Complete across 3 Clusters",
    x= "Bill Length",
    y= "Bill Depth",
    color = "Cluster"
  )
```




```
#two plots assessing other k's
#k=5
cluster_complete5 <- cutree(hcl.penguins_complete, k = 5)
penguins_3$cluster_complete5 <- cluster_complete5
ggplot(data = penguins_3, aes(x=bill_length_mm, y= bill_depth_mm,
                              color= factor(cluster_complete5))) +
  geom_point() +
  labs(
    title = " Q6: Bill Length vs Depth, \n Complete across 5 Clusters",
    x= "Bill Length",
    y= "Bill Depth",
    color = "Cluster"
  )
```



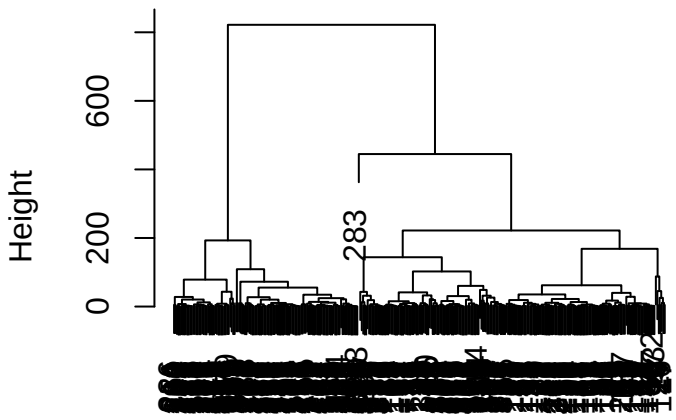

```
#k=2
cluster_complete2 <- cutree(hcl.penguins_complete, k = 2)
penguins_3$cluster_complete2 <- cluster_complete2
ggplot(data = penguins_3, aes(x=bill_length_mm, y= bill_depth_mm,
                              color= factor(cluster_complete2))) +
  geom_point() +
  labs(
    title = " Q6: Bill Length vs Depth, \n Complete across 2 Clusters",
    x= "Bill Length",
    y= "Bill Depth",
    color = "Cluster"
  )
```




For complete linkage clustering in this situation, we do not require as many clusters as single linkage to assess a very distinct pattern in our data points. I chose k=3 after assessing the dendogram and am able to assess 3 distinct groups. When k=2 instead, we see 2 groups clearly outlined with much overlap where the 3rd cluster would belong. When k= 5, we aren't able to parse out 5 unique clusters as there is too much overlap between them. Thus in the case of complete linkage clustering, k= 2 or 3 would be best choice as opposed to k= 7 and above as seen with single linkage clustering.

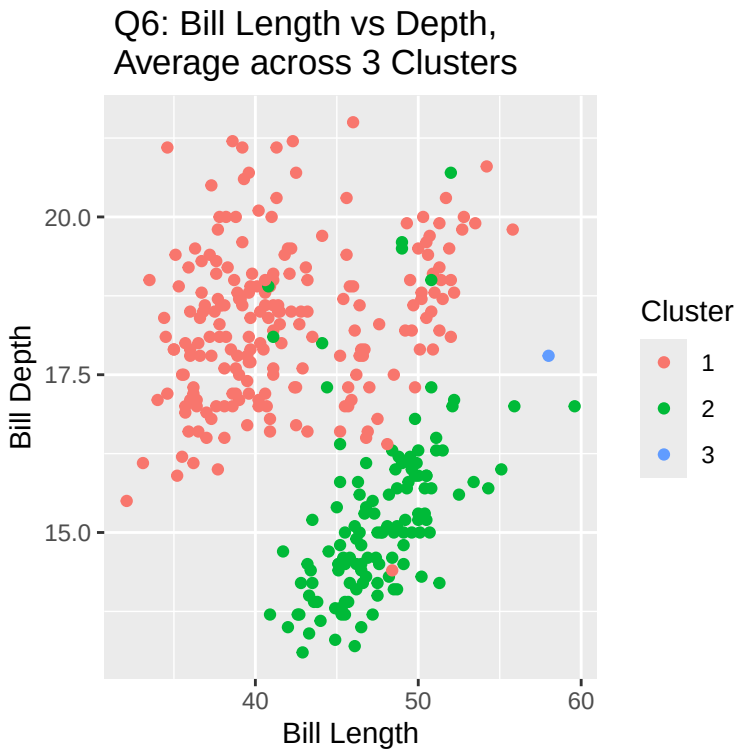
Average Linkage Clustering

```
#average linkage clustering
hcl.penguins_average <- hclust(penguins_dist,meth='average')

plot(hcl.penguins_average,
     main= "Average Linkage Dendrogram",
     xlab= "Penguins",
     sub = "Squared Euclidean Distance")
```

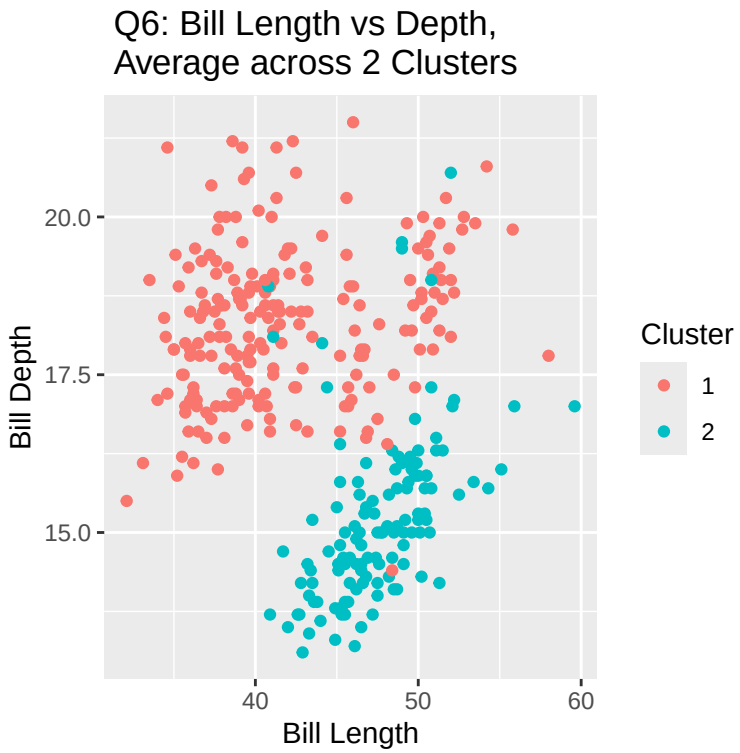



Penguins
Squared Euclidean Distance

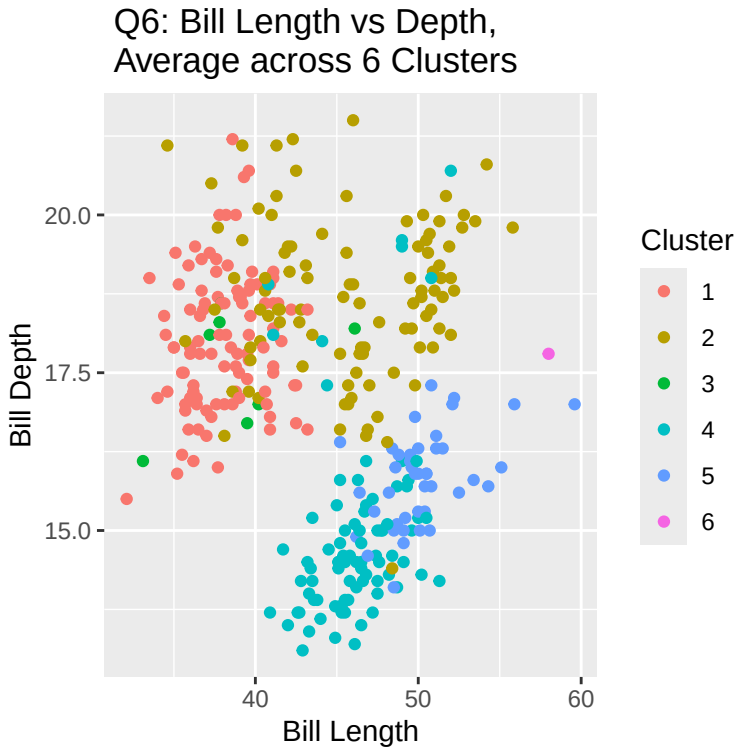
```
#Number of Clusters Chosen: 3
check<- factoextra::fviz_cluster(list(data=penguins_3,
                                      cluster=cutree(hcl.penguins_average,3)),
                                 choose.vars=c(1:3),main="Single Linkage")
#fviz_cluster shows 2 distinct cluster groups but with the third cluster in
#between the first two clusters

#save the clusters
cluster_average3 <- cutree(hcl.penguins_average, 3)
penguins_3$cluster_average3 <- cluster_average3

#plot
ggplot(data = penguins_3, aes(x=bill_length_mm, y= bill_depth_mm,
                              color= factor(cluster_average3))) +
  geom_point() +
  labs(
    title = " Q6: Bill Length vs Depth, \n Average across 3 Clusters",
    x= "Bill Length",
    y= "Bill Depth",
    color = "Cluster"
  )
```




```
#two plots assessing other k's
#k=2
cluster_average2 <- cutree(hcl.penguins_average, 2)
penguins_3$cluster_average2 <- cluster_average2
ggplot(data = penguins_3, aes(x=bill_length_mm, y= bill_depth_mm,
                              color= factor(cluster_average2))) +
  geom_point() +
  labs(
    title = " Q6: Bill Length vs Depth, \n Average across 2 Clusters",
    x= "Bill Length",
    y= "Bill Depth",
    color = "Cluster"
  )
```




```
#k=6
cluster_average6 <- cutree(hcl.penguins_average, 6)
penguins_3$cluster_average6 <- cluster_average6
ggplot(data = penguins_3, aes(x=bill_length_mm, y= bill_depth_mm,
                              color= factor(cluster_average6))) +
  geom_point() +
  labs(
    title = " Q6: Bill Length vs Depth, \n Average across 6 Clusters",
    x= "Bill Length",
    y= "Bill Depth",
    color = "Cluster"
  )
```




For average linkage clustering in this situation, we do not require as many clusters as single linkage to assess a very distinct pattern in our data points. I chose k=3 after assessing the dendogram but am able to assess 2 distinct groups when taking a look at the scatterplot. When k=2 instead, we see 2 groups clearly outlined with considerable overlap but not as much as there was when we assessed complete linkage clustering. When k= 6, we aren't able to parse out 6 unique clusters as there is too much overlap between them. Thus in the case of average linkage clustering, k= 2 would be best choice as opposed.

**Q7**: Do any of the clustering approaches and values of $k$ that you tested above produce approximately equal-sized clusters (all – not just a pair)? Report the approach and corresponding cluster solution (value of $k$) that produces the most similarly sized clusters under that approach? [.5 pt each part; 1 pt total]

```
table(penguins_3$cluster_complete2)
```

```
##
##   1   2
## 189 144
```

```
table(penguins_3$cluster_complete3)
```

```
##
##   1   2   3
## 189  99  45
```

```
table(penguins_3$cluster_complete5)
```

```
##
##   1   2   3   4   5
##  25 125  39  99  45
```

```
table(penguins_3$cluster_average2)
```

```
##
##   1   2
```

```
## 208 125
```

```
table(penguins_3$cluster_average3)
```

```
##
##   1   2   3
## 207 125   1
```

```
table(penguins_3$cluster_average6)
```

```
##
##   1   2   3   4   5   6
## 101  99   7  83  42   1
```

None of the clustering approaches and values of $k$ that I tested produced approximately equal-sized clusters. The Complete Linkage Clustering method with k=2 is the solution that produces the most similarly sized clusters.

**Q8**: Compute the cross-tabs comparing a 3 cluster solution in all three clustering methods, comparing each one against each of the other two. [.5 each cross tab; 1.5pt total] Which two methods yield the most similar clustering? Justify your answer. [1pt]

```
k <- 3
single <- cutree(hcl.penguins,k=k)
complete <- cutree(hcl.penguins_complete,k=k)
average <- cutree(hcl.penguins_average,k=k)

table(single)
```

```
## single
##   1   2   3
## 331   1   1
```

```
table(complete)
```

```
## complete
##   1   2   3
## 189  99  45
```

```
table(average)
```

```
## average
##   1   2   3
## 207 125   1
```

```
xtabs(~single + complete)
```

```
##       complete
## single   1   2   3
##      1 187  99  45
##      2   1   0   0
##      3   1   0   0
```

```
xtabs(~single + average)
```

```
##       average
## single   1   2   3
##      1 206 125   0
##      2   1   0   0
##      3   0   0   1
```

```
xtabs(~average + complete)
```

```
##         complete
## average   1   2   3
##       1 187  20   0
##       2   1  79  45
##       3   1   0   0
```

The two methods that yield the most similar clustering are Complete and Average Linkage Clustering. Although there is not perfect alignment, we see that these two methods have the greatest amount of overlap with cluster one= 187 and cluster two=79. This indicates that there is agreement at least between the first two clusters in these methods, which is the relative greatest amount of simillarity for k= 3.