APSTA-GE 2011. Homework #2

January 2025: Yael Beshaw

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

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

# standardize the data
penguins.stdz <- penguins
penguins.stdz[, 3:5] <- scale(penguins[, 3:5])
```

You will need to understand k-means clustering and choosing the number of clusters using the Calinski-Harabasz, or C(g) criterion, discussed in Handout 2 and in class.

Q1:

Use k-means clustering (algorithm=Hartigan-Wong), finding solutions for g=2,3,4,5,6,7,8,9 groups. ALWAYS start each clustering run by set.seed(2011) before each call to kmeans AND add this option to the call: nstart=100 (this generates 100 random starts and keeps kmeans from producing local minimum solutions). Remember: you do not need to include the body mass measure. [0 pts, but you must do it to proceed]

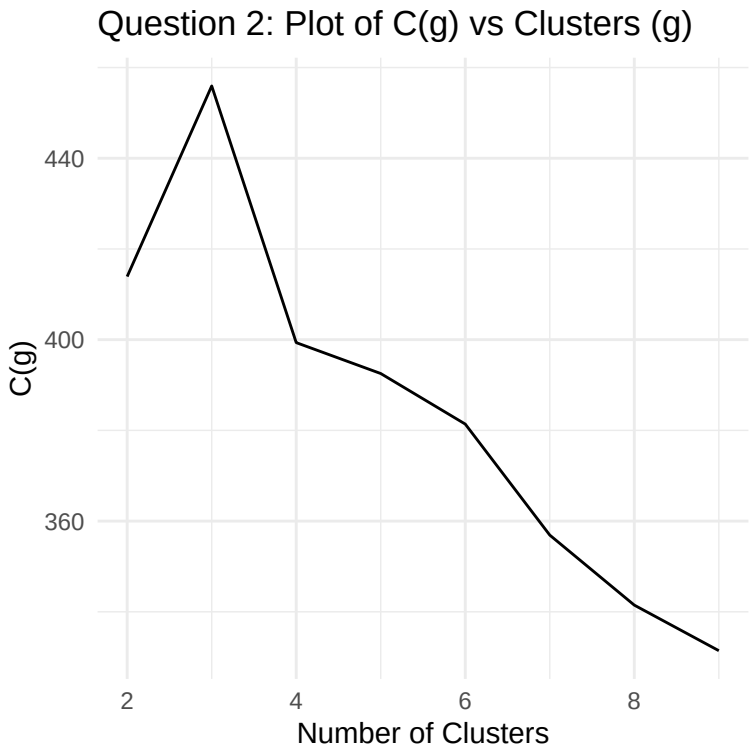
```
# specify the columns of interest
penguins.stdz_data <- penguins.stdz[, c("bill_length_mm", "bill_depth_mm",
    "flipper_length_mm")]


# loop through the solutions for g and store the results
kmeans_g <- list()

for (g in 2:9) {
    set.seed(2011)
```

```
    penguinz_kmeans <- kmeans(penguins.stdz_data, centers = g,
        nstart = 100, algorithm = "Hartigan-Wong")
    kmeans_g[[(g)]] <- penguinz_kmeans

}
```

Q2:

For each value of g, compute $C(g) = (\sum \text{msb})/(\sum \text{msw})$. In R, use the `calinhara` function given in package `fpc`. For this assignment, do not use the `NbClust` function in the `NbClust` R library, because it runs the kmeans fits for you, which will make it harder to generate them equivalently afterward. PLOT $C(g)$ as a function of g. [2pts]

```
# create a list that stores the clusters for each g that we
# will run the calinhara function on

Cg_kmeans <- list()

for (g in 2:9) {
    kmeans <- kmeans_g[[(g)]]$cluster
    Cg_kmeans[[g]] <- kmeans
}


# store the results of the Cg for each in a list
Cg_g <- list()

# run the calinhara() function for each g
for (g in 2:9) {
    set.seed(2011)
    Cg <- calinhara(penguins.stdz_data, Cg_kmeans[[g]], cn = max(Cg_kmeans[[g]]))
    Cg_g[[(g)]] <- Cg
}

# turn into table in order to plot C(g) as a function of g
Cg <- data.frame(g = 2:9, Cg = unlist(Cg_g))

# plot
ggplot(Cg, aes(x = g, y = Cg)) + geom_line() + labs(title = "Question 2: Plot of C(g) vs Clusters (g)",
    x = "Number of Clusters", y = "C(g)") + theme_minimal()
```



Q3:

Choose the g such that $C(g)$ is maximized. Call this variable g_{opt}. [1pt]

```
# The g such that C(g) is maximized is 3
g_opt <- 3
```

g = 3

Q4:

Compute the k-means cluster solutions using the following number of groups: g_{opt}, $g_{opt} + 1$, and $g_{opt} + 2$; compare the results using crosstabs (ignoring the $g_{opt} + 1$ to $g_{opt} + 2$ comparison) [1 pt each; 2 pts total] and comment on the effect of changing the number of clusters for each crosstab. [0.5pts each; 1pt total]

```
# compute the kmeans cluster solutions
km.penguins.opt <- kmeans(penguins.stdz_data, g_opt)
km.penguins.opt1 <- kmeans(penguins.stdz_data, g_opt + 1)
km.penguins.opt2 <- kmeans(penguins.stdz_data, g_opt + 2)

# compare the results using crosstabs
xtabs(~km.penguins.opt$cluster + km.penguins.opt1$cluster)
```

```
##                         km.penguins.opt1$cluster
## km.penguins.opt$cluster   1   2   3   4
##                       1 146   0   0   0
##                       2   0   0  50  69
##                       3   0  68   0   0
```

```
xtabs(~km.penguins.opt$cluster + km.penguins.opt2$cluster)
```

```
##                          km.penguins.opt2$cluster
## km.penguins.opt$cluster  1  2  3  4  5
##                       1  0  0 79 67  0
##                       2 50  0  0  0 69
##                       3  0 64  0  4  0
```

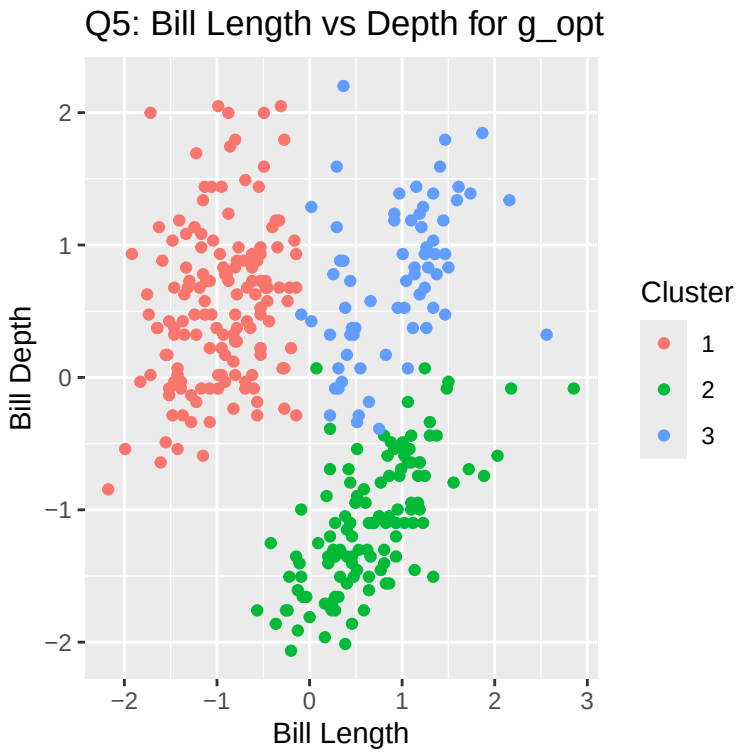
When analyzing the effect of adding more clusters, we see that in the first cluster from our g_opt = 3 is defined in both g_opt + 1 and in g_opt + 2 with 119 penguins. However, in the comparison of g_opt and g_opt + 1, there is a split occurring with g_opt's second cluster in which g_opt + 1 separates the 146 penguins into two separate clusters almost down the middle. It also separates g_opt's third cluster into 2 clusters but with only 6% of the penguins in a separate cluster. Thus, we have evidence that g_opt's Cluster 2 could be divided further. When we assess the comparision of g_opt and g_opt + 2, we see a similar story. The second cluster of g_opt is again split across 3 different clusters with near equal separation whereas Cluster 3 is not seperated much. This supports the conclusion that while 3 clusters is the most optimal according to C(g), there is evidence that we may have a basis for further dividing Cluster 2, resulting in 4 clusters instead.

Q5

Color the bivariate plot of bill depth (y axis) vs. bill length (x-axis) for each of the three solutions with $g = \{g_{opt}, g_{opt} + 1, g_{opt} + 2\}$ and describe what changes with each solution. [1 pt ea. plot; 1 pt ea. comment; 6 pts total]

g = g_opt

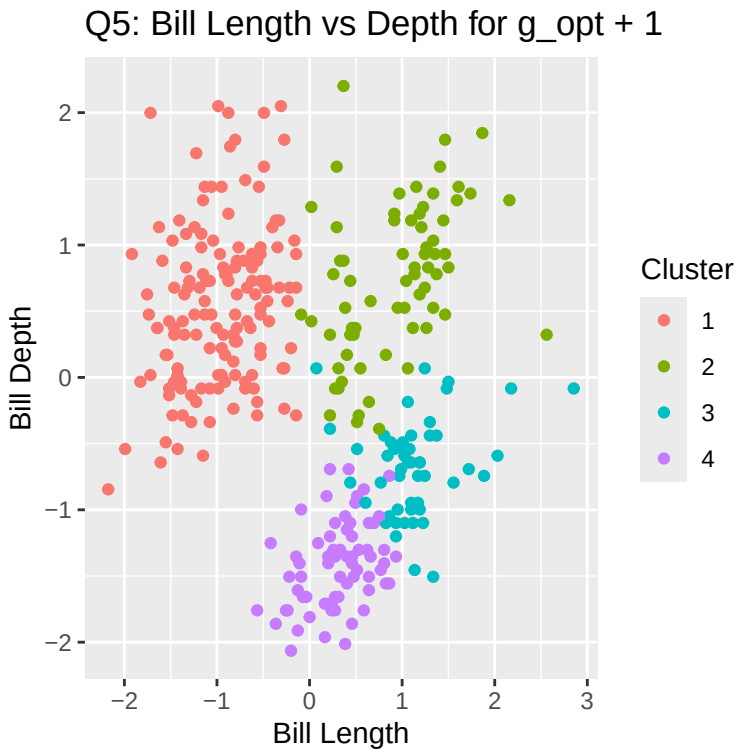
```
# g = g_opt
km.penguins.opt <- km.penguins.opt$cluster
penguins.stdz_data$km.penguins.opt <- km.penguins.opt
ggplot(data = penguins.stdz_data, aes(x = bill_length_mm, y = bill_depth_mm,
    color = factor(km.penguins.opt))) + geom_point() + labs(title = "Q5: Bill Length vs Depth for g_opt
    x = "Bill Length", y = "Bill Depth", color = "Cluster")
```



Taking a look here at the bivariate plot for Bill Depth vs Length, we can see that the three clusters are defined with overlap in between which is expected. Cluster 2 (green) seems to have the most observations with minimal overlap into Clusters 1 or 3. Overall, 3 clusters provides the most distinct cluster groups and is optimal based on our definition of C(g).

g = g_opt + 1

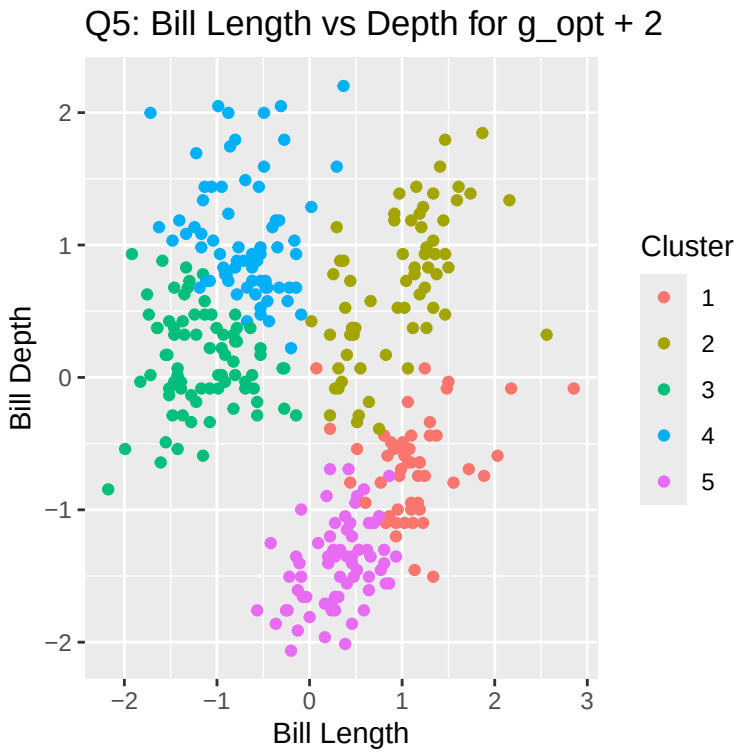
```
# g = g_opt + 1
km.penguins.opt1 <- km.penguins.opt1$cluster
penguins.stdz_data$km.penguins.opt1 <- km.penguins.opt1
ggplot(data = penguins.stdz_data, aes(x = bill_length_mm, y = bill_depth_mm,
    color = factor(km.penguins.opt1))) + geom_point() + labs(title = "Q5: Bill Length vs Depth for g_op
    x = "Bill Length", y = "Bill Depth", color = "Cluster")
```



Once we add another cluster to our data, we see that Cluster 2 in g_opt is now split into 2 separate clusters (1 and 4). We see here that this split is almost right down the middle and provides evidence that while these groups are similar, they have a distinctive feature that we can investigate to assess whether or not 4 clusters is more in line with our data and the questions we are trying to answer.

g = g_opt + 2

```
# g = g_opt + 1
km.penguins.opt2 <- km.penguins.opt2$cluster
penguins.stdz_data$km.penguins.opt2 <- km.penguins.opt2
ggplot(data = penguins.stdz_data, aes(x = bill_length_mm, y = bill_depth_mm,
    color = factor(km.penguins.opt2))) + geom_point() + labs(title = "Q5: Bill Length vs Depth for g_opt
    x = "Bill Length", y = "Bill Depth", color = "Cluster")
```



Finally, once we split the data into 5 clusters. Our original Cluster 1 and 3 remain unchanged for the exception of 2-4 observations each. More importantly, we see that our orginal Cluster 2 is now split into three clusters. This clustering is not as clear and informative as g_opt or g_opt + 1, but it gives us insight about the original Cluster 2 possibly being oversimplified.