Homework 3

STAT 437 Spring 2022

Christopher Mims
(10436827)

February 26, 2022

```
library(knitr)
library(formatR)
library(tidyr)
library(dplyr)
library(nycflights13)
library(ggplot2)
library(cluster)
library(ggdendro)
source('Plotggdendro.r')
```

Conceptual Exercises

K-means Clustering Methodology

1.1) Give a few examples of dissimilarity measures that can be used to measure how dissimilar two observations are. What is the main disadvantage of the squared Euclidean distance as a dissimilarity measure?

Dissimilarity can be subdivided into two subcategories. These two categories define the dissimilarity measure based on either quantitative features (mainly numerical values) or qualitative values (mainly non-numerical values).

For quantitative data, two observations' dissimilarity can be calculated by the distance between the two observations. The smaller the distance, the more similar, and the larger the distance, the more dissimilar. In calculating these distances, one could use a correlation distance, Manhattan distance, or many others. When calculating the correlation distance, one finds the Pearson correlation coefficient and subtracts it from the value of one (1). This is due to the fact that a strong, positive correlation will have a Pearson correlation coefficient value closest to one (1). Sticking with similar observations being closer together, $1 - corr(\mathbf{x}_i, \mathbf{x}_j)$ results in a smaller value for more similar observations. Another quantitative dissimilarity calculation is the Manhattan distance. This calculation results from the sum of the absolute differences between the $\boldsymbol{p}$ paired entries of $\mathbf{x}_i$ and $\mathbf{x}_j$. This calculation gets it name from the similarity of calculating the distance to walk from one location in Manhattan to another. Since you cannot walk through buildings and must remain on the sidewalks, this calculation takes that into account.

For qualitative data, two observations' dissimilarity is calculated differently. First, each of the values must be mapped to numerical value, and then the calculations can be applied. Two categories within qualitative data are ordinal variables (such as academic grades) and nominal variables (such as pass or fail). With ordinal variables the distance can be calculated by the following:

$$y \mapsto \tilde{y} = [rank(y) - 2/M]M$$

where M is the total number of ordinal values the variable can assume, and y is the observed value. Another calculation for dissimilarity with qualitative data performs the calculations on nominal observations. Here, the distance value ($d(a, b)$) is set to zero (0) when the two observations are the same ($a = b$) and when they are not the same, the distance value ($d(a, b)$) is set to one (1).

Source: Stat 437 Lecture Notes 3 by Xiongzhi Chen

The main disadvantage of the squared Euclidean distance as a dissimilarity measure is that it can only be used on quantitative data. The measure is calculated from the vectors of the observations and these values must be numeric. Also, there is the 'curse of dimensionality' where the more dimensions an observation has, the less reliable the distance metric become, due to the fact that there becomes little difference in the distances between the observations.

1.2) Is it true that standardization of data should be done when features are measured on very different scales? Is it true that employing more features gives more accurate clustering results? Is it true that employing standardized observations gives more accurate clustering results than employing non-standardized ones? Explain each of your answers.

It is common practice to scale data when the features of the observations are measured on very different scales. For example, if income was measured in dollars for one feature and in cents for another feature. This could have a large impact on the overall results. The use of employing more features does not always correlate to more accurate clustering results. Adding more features to an observation with a small number of features may help increase accuracy of the clustering algorithm, but adding many more features to any sample may detract from the accuracy of the clustering algorithm. This was explained in the previous section where I talked about the 'curse of dimensionality'. The employment of standardized observations versus non-standardized observations for more accurate results depends on the application at hand. There are many applications where standardized observations will produce a more accurate clustering result over non-standardized observations, and vice versa. One needs to take into account all aspects of the observations and how standardizing could affect the accuracy of the clustering result.

1.3) Take $K = 2$. Provide the loss function that K-means clustering tries to minimize. You need to provide the definition and meaning of each term that appears in the loss function.

The loss function used in K-means is as follows:

$$W(C) = \sum_{\{\mathbf{x}_i \text{ in Cluster 1}\}} d^2(\mathbf{x}_i, \bar{\mathbf{x}}_1) + \sum_{\{\mathbf{x}_i \text{ in Cluster 2}\}} d^2(\mathbf{x}_i, \bar{\mathbf{x}}_2)$$

In the loss function we will find the distance (d) between any observation ($\mathbf{x}_i$) and the center of the cluster ($\bar{\mathbf{x}}_1$), which is the mean point of the cluster, in cluster 1. This distance will then be squared (this ensures that it is a positive distance). Then all of the squared distances will be summed. This same process happens for cluster 2 and once all calculations and sums for each cluster have been made, the two cluster summations are summed. The loss function will try to assign each of the observations to the closest centroid. This causes the distances between each observation and the centroid to decrease. The smallest value obtained from this process gives the best clustering result.

1.4) What is the "centroid" for a cluster? Is the algorithm, Algorithm 10.1 on page 388 of the Text, guaranteed to converge to the global minimum of the loss function? Why or why not? What does the argument `nstart` refer to in the command `kmeans`? Why is `nstart`suggested to take a relatively large value? Why do you need to set a random seed by `set.seed()` before you apply `kmeans`?

A centroid is the center point in the cluster. This does not mean it is an observation, only a point within the cluster. Algorithm 10.1 in the Text does not guarantee that the algorithm will converge to the global minimum of the loss function. This is due to the fact that by selecting random observations you cannot guarantee that they will not be evenly dispersed to create the correct clusters to minimize the loss function. The argument `nstart` sets the number of random, initial configurations of cluster memberships to the value given. Specifying a large value for `nstart` allows for many different configurations to be used to start the clustering

algorithm, increasing the changes that the best result will be found. It is best to apply the `set.seed()` function before running `kmeans` do to the fact that it will keep the data in the same order, allowing for the different starts to begin with the same data in the same place.

1.5) Suppose there are 2 underlying clusters but you set the number of clusters to be different than 2 and apply `kmeans`, will you have a good clustering results? Why or why not?

If there are already two underlying clusters, you many not have a good clustering result if you set the number of clusters not equal to two. This is due to the fact that the algorithm will place observations near the chosen centroids and this could lead to wildly inaccurate results.

1.6) Is the true number K_0 of clusters in data is known? When using the command `clusGap` to estimate K_0, what does its argument `B` refer to?

The true number of clusters is not know. There can be many factors that lead to the formation of a cluster and a cluster could possibly be able to be sub divided into clusters itself. The argument 'B' in the `clusGap` function refers to the number of Monte Carlo samples. It is suggested that the larger the value for 'B', the more computations need to be made.

Hierarchiacal Clustering

2.1) What are some advantages of hierarchical clustering over K-means clustering? What is the relationship between the dissimilarity between two clusters and the height of these clusters in the dendrogram that represents a bottom-up tree?

When using hierarchical clustering over k-means clustering, you are able to see how the observations are situated with one another. Since the pairs of the observations are made off of the smallest distance between two observations or groups, one can see how relatively similar observations are. Hierarchical clustering doesn't have the disadvantage of not converging. They will always coverage on the most granular level. The dissimilarity of the two clusters is directly related to the height of the clusters. The more dissimilar, the larger the height shown on the dendrogram.

2.2) Explain what it means by saying that "the clusters obtained at different heights from a dendrogram are nested". If a data set has two underlying clustering structures that can be obtained by two different criteria, will these two sets of clusters necessarily be nested? Explain your answer.

All of the observations will be under one cluster. This one cluster will then become two clusters. Therefore, this means that when you cut the dendrogram to form three clusters, these three clusters are nested under two clusters, which is nested under one cluster. For how ever many clusters there are, they will all ultimately be clustered under one cluster. If there are two clusters that can be obtained by two different criteria, then these two clusters will not necessarily be nested. This is due to the fact that if you had cars that were either Ford or Chevy and either V6 or V8, you could not nest each of these into their clusters when each group would have to contain each of the other groups.

2.3) Why is the distance based on Pearson's sample correlation not effected by the magnitude of observations in terms of Euclidean distance? What is the definition of average linkage? Why are average linkage and complete linkage preferred than single linkage in practice?

The distance based on Pearson's sample correlation are not effected by the magnitude of observations in terms of Euclidean distance because it is a measurement of how correlated two observations are and not the distance between them. When two observations are highly correlated they have a value near one and the distance between the observations is $1 - corr$, making their distance very small. "Average linkage" takes the average of all the dissimilarities in a cluster and then compares that to the average dissimilarity of another cluster. When deciding on a linkage one should aim at capturing latent patterns in the data. By using single linkage you are only getting information from two observations and not an overall picture. This distorts the visual and the overall theme of the data may be skewed or lost.

2.4) What does the command `scale` do? Does `scale` apply row-wise or column-wise? When `scale` is applied to a variable, what will happen to the observations of the variable?

The command `scale` standardizes the data. It will take the values contained within a *column* and set them to have a sample mean of 0 and a standard deviation of 1. Therefore, it would apply column-wise. When scale is applied to an observation, the features of the observation will be standardized. This would affect its relation to other observations, because each observation would have its own mean and standard deviation, not allowing them to be compared properly.

2.5) What is `hclust$height`? How do you find the height at which to cut a dendrogram in order to obtain 5 clusters?

`hclust$height` is the column of values that stores the heights of each nested cluster. Therefore, to find the height in which to cut a dendrogram into 5 clusters would be the same as finding the 5th largest height. The heights are stored in ascending order, therefore, when you call `name_of_hierarchical_cluster$height[length(name_of_hierarchical_cluster$height)-5]`, it will return the height that is in the 5th to last index of the height column, which is the the 5th largest height. If you needed to know the value of the height at that point, you can simply display or print that value.

2.6) When creating a dendrogram, what are some advantages of the command `ggdendrogram{ggdendro}` over the R base command `plot`?

Using the `ggdendrogram` command allows for better fine-grained control over the plot. It extracts all of the plot data from dendrogram objects which leads to better customization.

Applied Exercises

The goal of the following exercise is use the four features extracted for each observation of the `flights` data set to identify if an observation belongs to a specific carrier or a specific month. To begin, we must filter the data set by choosing the three `carrier` values of "UA", "AA", and "DL", the `month` values of 7, and 2, and the four features `dep_delay`, `arr_delay`, `distance`, and `air_time`. In order to achieve the best results, we must remove all `na` values from the subset as well.

```
p3 <- select(flights, carrier, month, dep_delay, arr_delay, distance, air_time) %>%
        dplyr::filter(carrier %in% c('UA', 'AA', 'DL'),
                      month %in% c(2, 7))
p3 <- na.omit(p3)
```

Part 1

Next we will apply K-means with $K = 2$ and 3 respectively and each with `set.seed(1)` and `nstart=20`.

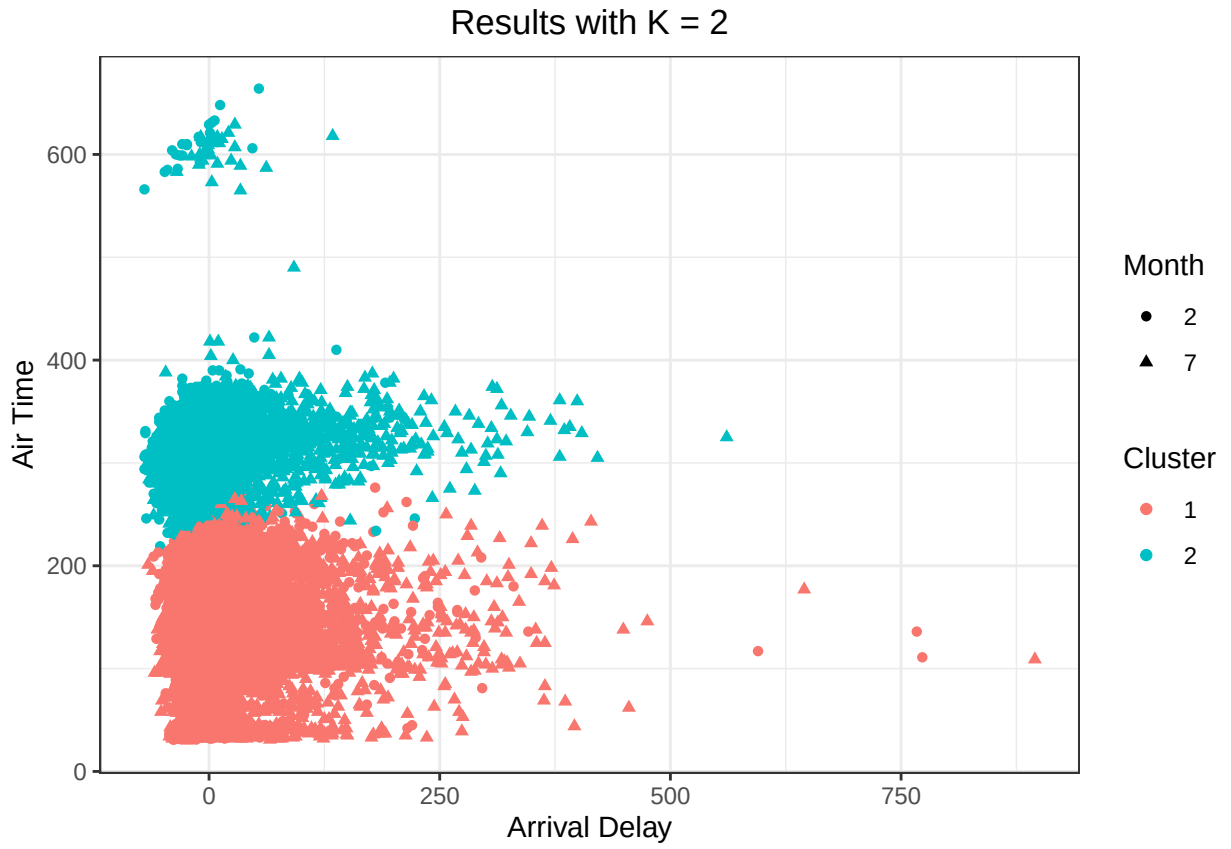
For $K = 2$, we will provide a visualization of the clustering results based on true clusters given by `month`.

```
# Set seed to get consistent results
set.seed(1)

# Perform k-means clustering into two clusters with an `nstart` of 20
# `nstart` sets the number of random, initial configurations of
# cluster memberships
km.part3.1a <- kmeans(p3[, 3:6], 2, nstart = 20)

# Create a dataframe to store the results from clustering
# This data will then be used to plot a visualization of
# how the clustering algorithm clustered the observatons
results.p3.1a <- data.frame(p3)
results.p3.1a$cluster <- factor(km.part3.1a$cluster)
```

```r
# Plot the results of the clustering using color for cluster
# and using shape for month
plt.p3.1a <- ggplot(results.p3.1a,
                    aes(results.p3.1a$arr_delay, results.p3.1a$air_time)) +
               geom_point(aes(shape = as.factor(month), color = cluster)) +
               xlab("Arrival Delay") +
               ylab("Air Time") +
               ggtitle("Results with K = 2") +
               labs(shape = 'Month', color = 'Cluster') +
               theme_bw() +
               theme(plot.title = element_text(hjust = 0.5))
plt.p3.1a
```



For $K = 3$, we will provide a visualization of the clustering results based on true clusters given by `carrier`.

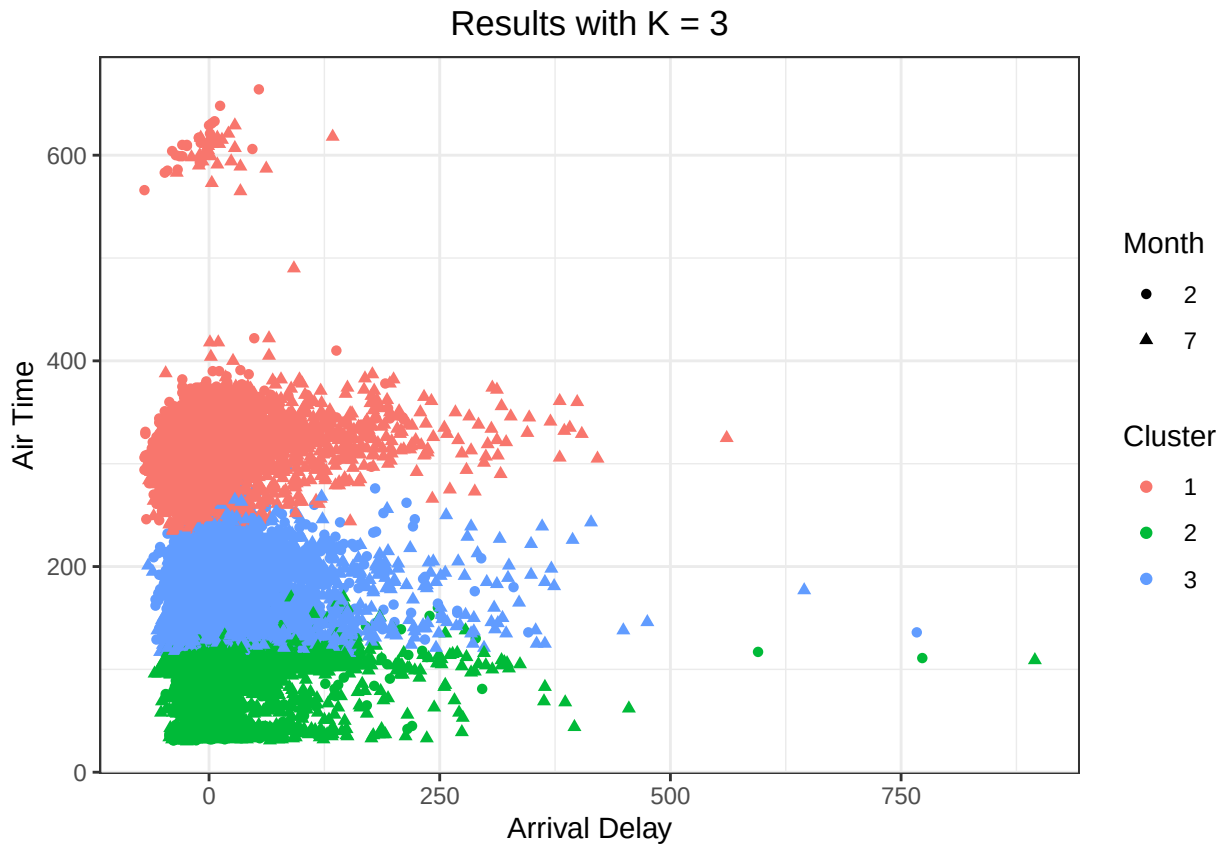
```r
# Set seed to obtain reproducible results
set.seed(1)

# Perform k-means clustering into three clusters with an `nstart` of 20
# `nstart` sets the number of random, initial configurations of
# cluster memberships
km.part3.1b <- kmeans(p3[, 3:6], 3, nstart = 20)

# Create a dataframe to store the results from clustering
# This data will then be used to plot a visualization of
# how the clustering algorithm clustered the observatons
```

```
results.p3.1b <- data.frame(p3)
results.p3.1b$cluster <- factor(km.part3.1b$cluster)

# Plot the results of the clustering using color for cluster
# and using shape for month
plt.p3.1b <- ggplot(results.p3.1b,
                    aes(results.p3.1b$arr_delay, results.p3.1b$air_time)) +
               geom_point(aes(shape = as.factor(month), color = cluster)) +
               xlab("Arrival Delay") +
               ylab("Air Time") +
               ggtitle("Results with K = 3") +
               labs(shape = 'Month', color = 'Cluster') +
               theme_bw() +
               theme(plot.title = element_text(hjust = 0.5))
plt.p3.1b
```



Part 2

Now we will use `set.seed(123)` to randomly extract 50 observations from the subset. We will then apply hierarchical clustering on these 50 observations with average linkage.

```
# Set seed to obtain reproducible results
set.seed(123)

# Randomly select rows without replacement
row.slct <- sample(1:dim(p3)[1], size = 50, replace = FALSE)
```

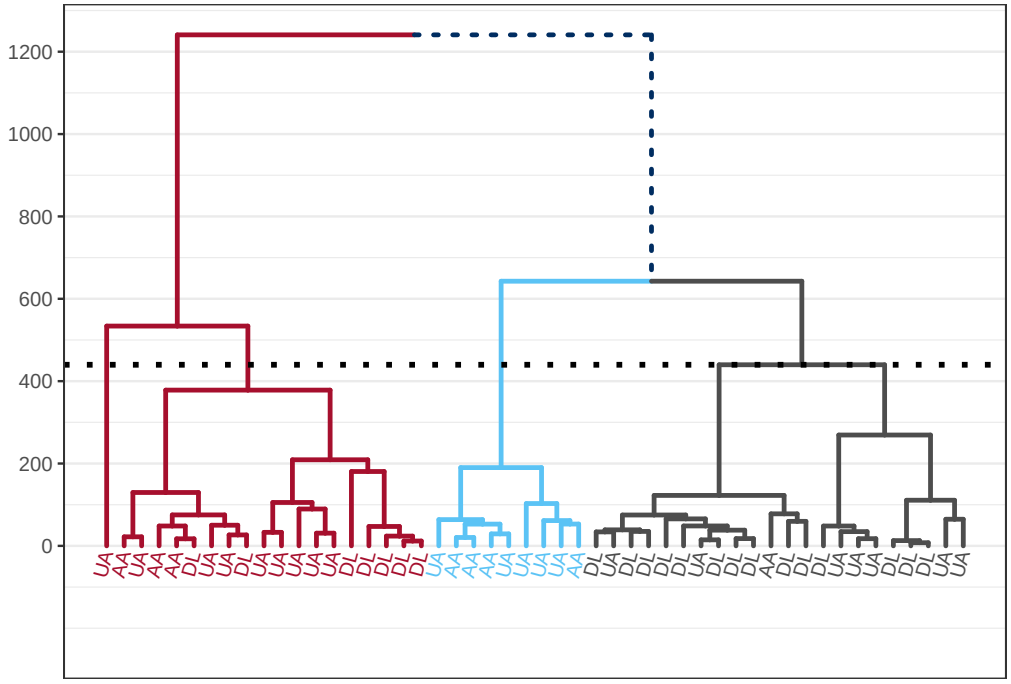
```
# Builds dataframe from the 50 randomly selected rows
p3.2 <- p3[row.slct,]
p3.2$month <- as.character(p3.2$month)

# Create the dataframe with `carrier` as the leaf labels
p3.2ct <- t(p3.2)                # Transpose the dataframe
colnames(p3.2ct) <- p3.2ct[1,]   # Assign column names to row containing `carrier` values
p3.2ct <- p3.2ct[3:6,]           # Select only the relevant data
p3.2c <- t(p3.2ct)               # Flip back to the dataframe to use in algorithm

# Create the dataframe with `month` as the leaf labels
p3.2mt <- t(p3.2)
colnames(p3.2mt) <- p3.2mt[2,]
p3.2mt <- p3.2mt[3:6,]
p3.2m <- t(p3.2mt)
```

Part i We will cut the dendrogram to obtain three (3) clusters with leaves annotated by `carrier` names with the resulting clusters colored distinctly, and the corresponding height of the cut will the be reported.

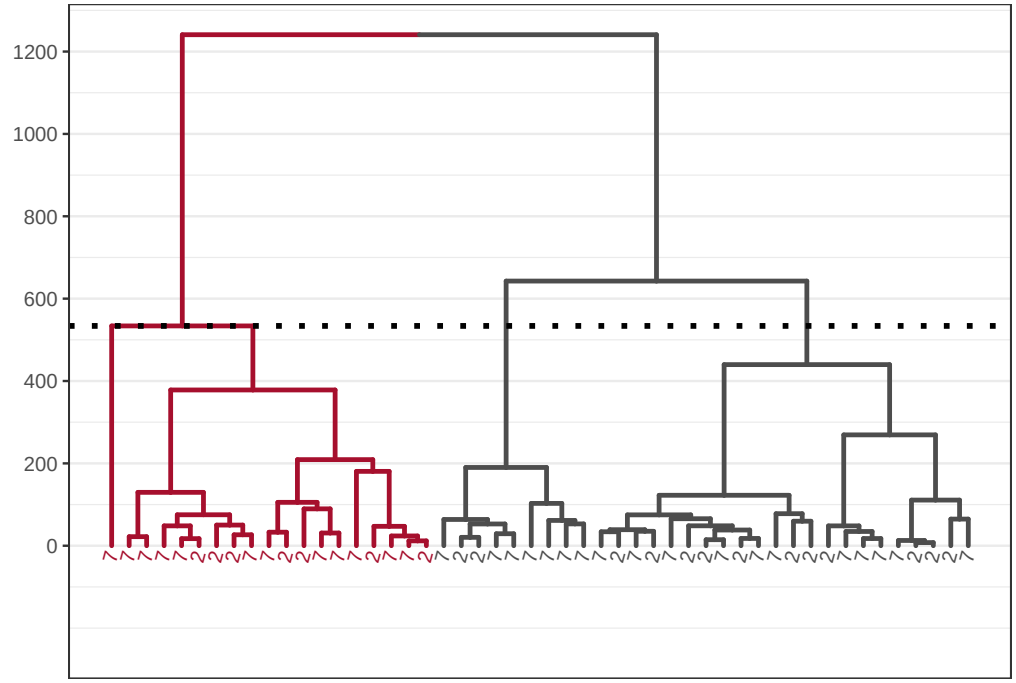
```
hc.p3.2c <- hclust(dist(p3.2c), method = 'average')
cut.height <- hc.p3.2c$height[length(hc.p3.2c$height)-3]
plt.hc.p3.2c <- plot_ggdendro(dendro_data_k(hc.p3.2c, 3),
                              direction = 'tb',
                              heightReferece = cut.height,
                              scale.color = c('#002D61', '#4D4D4D', '#A60F2D',
                              '#5BC3F5'),
                              expand.y = 0.2)
plt.hc.p3.2c
```



The height of the cut to obtain the three clusters is 439.8130877.

Part *ii* Now we will cut the dendrogram to obtain two (2) clusters with leaves annotated by `month` numbers with the resulting clusters colored distinctly, and the corresponding height of the cut will be reported.

```
hc.p3.2m <- hclust(dist(p3.2m), method = 'average')
cut.height <- hc.p3.2m$height[length(hc.p3.2m$height)-2]
plt.hc.p3.2m <- plot_ggdendro(dendro_data_k(hc.p3.2m, 2),
                              direction = 'tb',
                              heightReferece = cut.height,
                              scale.color = c('#4D4D4D', '#A60F2D'),
                              expand.y = 0.2)
plt.hc.p3.2m
```



The height of the cut to obtain the three clusters is 533.7757547.