Stat 437 HW3

Nam Jun Lee (11606459)

General rule

You must complete both Conceptual and Applied exercises. Please show your work and submit your computer codes in order to get points. Providing correct answers without supporting details does not receive full credits. Please upload your answers to the course space. This HW covers

- K-means clustering
- Hierarchical clustering

For an assignment or project, you DO NOT have to submit your answers or reports using typesetting software. However, your answers must be well organized and well legible for grading. Please upload your answers in a document to the course space. Specifically, if you are not able to knit a .Rmd/.rmd file into an output file such as a .pdf, .doc, .docx or .html file that contains your codes, outputs from your codes, your interpretations on the outputs, and your answers in text (possibly with math expressions), please organize your codes, their outputs and your answers in a document in the format given below:

```
Problem or task or question ...
Codes ...
Outputs ...
Your interpretations ...
```

It is absolutely not OK to just submit your codes only. This will result in a considerable loss of points on your assignments or projects.

Conceptual exercises

1. Consider the 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?

If $x = (x1, ., xp)$ and $y = (y1, ., yp)$ of p quantitative features are set, two measures of difference can be used to determine how different the two observations are.

First, the `Correlation distance` can be determined by using $1 - corr(x, y)$ through Pearson correlation between p samples (xk, yk).

Second, `Manhattan distance` can be determined by the sum of the absolute differences between x and y composed of p pairs.

As a difference scale, the main drawback of Euclidean distance squared reduces computational complexity, but reduces the robustness of the methodology for outliers, and is not the only measure of specificity. (reference of lecture note)

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.

True, data standardization should be implemented when measuring features on very different scales. Since the similarity between data points is judged based on distance, standardizing and using data as 0 or 1 shows better results if the characteristics are very different scales.

True, more accurate results can be obtained when clustering is executed by employing more functions. If the K-means clustering results are low in accuracy, better results can be obtained by repeated initial center selection and using different distance measurements again.

True, using standardized observations provides more accurate results than using non-standardized observations. k-means clustering cannot obtain more accurate results than standardized observations when observations with different measurement units of variables used to measure results based on distance are used. However, standardization is not always appropriate. There are variables that need to be standardized and variables that do not need to be used, so it should be judged and used.

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.

K-means minimize lose function take $K = 2$:

$$W(C) = \sum d^2(x_i, \bar{x}_1) + \sum d^2(x_i, \bar{x}_2) = N_1 \times {S_1}^2 + N_2 \times {S_2}^2$$

In $W(C)$, **W** is represents the results of Loss function with mapping **C**. **C** is creates clusters for which observations within a cluster are quite similar but those between clusters are quite dissimilar.

$\sum d^2(x_i, \bar{x}_1)$ is first cluster and for observations x_i and $\bar{x}_1$ means they are more similar and less dissimilar. Also, **d** is Euclidean distances.

$\sum d^2(x_i, \bar{x}_2)$ is second cluster and for observations x_i and $\bar{x}_2$ means they are more similar and less dissimilar. Also, **d** is Euclidean distances.

In $N_1 \times {S_1}^2$ and $N_2 \times {S_2}^2$, **N** means cluster and **S** means sample variance.

Each summand is the within-cluster variability for the corresponding cluster. This means, K-means attempts to minimize the total within-cluster variability.

1.4) What is the "centroid" for a cluster? Is the algorithm, Algorithm 10.1 on page 388 of the Text (which is also provided in the lecture slides), 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`?

Centroid for Cluster k function:

$$\bar{x}_k = \frac{1}{N_k} \sum x_i$$

Centroid for Cluster k is the sample mean of its observations.

TRUE, Randomly generate a number between 1 and K in each observation as an initial cluster member and repeat computing cluster duplication for each cluster until the cluster allocation change stops, you can continue to reduce the target value at each stage to ensure convergence to the global minimum of the loss function.

In the `kmeans` command, the argument `nstart` means the number of arbitrary initial configurations (datasets) of cluster membership.

`nstart` is relatively large, such as 20 or 50, to use a random case at the center of the initial data set. In addition, if you check the total within-cluster sum of squares of the value of `nstart` with a smaller value and a larger value, you can see that the data is more clustered using larger value of `nstart`. For reference, the smaller the total within-cluster sum of squares, the better they are. For example:

```
set.seed(1)
# set the dataset
ex <- matrix(rnorm(100 * 2), ncol = 2)
ex[1:50, 1] = ex[1:50, 1] + 3
ex[1:50, 2] = ex[1:50, 2] - 4
set.seed(123)
first_ex <- kmeans(ex, 3, nstart = 1)
second_ex <- kmeans(ex, 3, nstart = 20)
# check the total within-cluster sum of squares first_ex
# and second_ex
first_ex$tot.withinss
```

```
## [1] 132.4976
```

```
second_ex$tot.withinss
```

```
## [1] 132.1317
```

From the results, it can be seen that the total square sum in the cluster for designating `nstart` as 1 is *132.4976*, and the total square sum for designating 20 is *132.1317*, this shows that the data with a large `nstart` is better aggregated.

If any random number has not been set to set.seed() before applying kmeans, different random numbers will continue to be generated each time the program is repeated, and the results will continue to vary. Therefore, when setting a random number, you should use set.seed() to make the same random number.

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 good clustering results? Why or why not?

FALSE, perfect clustering is achieved when the group matches the cluster. If the group is divided into two groups, but the number of clusters is set to 3, there is an unknown boundary within the data cluster, and the exact result cannot be confirmed.

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

FALSE, the actual number of clusters in the data $K_0 $ is unknown. $K_0$ varies in various estimates based on the number of features.

When estimating $K_0$ using the command `clusGap`, the argument `B` means the number of Monte Carlo (bootstrap) samples. Larger B requires more.

2. Consider hierarchical 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?

Compared to K-means clustering, the advantages of hierarchical clustering do not have two disadvantages of K-means.
# * There is no need to set the number of clusters to be obtained.
# * There is no need to set the initial center of the cluster to start the optimization process.

As the dendrogram, representing a bottom-up tree, moves upward from the bottom of the tree, the leaves begin to fuse into branches and the leaves and branches fuse into branches. The difference between the two clusters indicates the height at which the two clusters must be fused in the dendrogram, and the larger the height, the more different the branches are at this height.

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.

"the clusters obtained at different heights from a dendrogram are nested" means hierarchical clusters when there is only one criterion for determining clusters or groups of data.

hierarchical clustering will not necessarily be produced when sets of clusters are produced by different criteria. This means that two clusters should be overlap.

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?

Pearson's sample correlation-based distance is judged to be similar if the correlation is high even if the observed value is far from the Euclidean distance side. Pearson's sample correlation-based distance is not affected by the size of the observation in terms of Euclidean distance because it is based on the standardized items of each observation depending on how the correlation is calculated.

Define average linkage: Calculate all pairwise differences between observations in cluster A and observations in cluster B and record the average of these differences.

The reason why average and complete linkages are preferred over single linkage in practice is that average and complete linkages tend to yield more balanced dendrograms than single connections.

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` applies to each column of the matrix.
The `scale` is applied column-wise.

Applying a `scale` to a variable standardizes the observations of the variable.

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` called “clustering heights”. It is a set of n - 1 real value and each height is the value of the dissimilarity measure for the corresponding agglomeration.

To obtain five clusters, use the `cutree` command to divide them into several groups by specifying the desired number of groups or cutting height.
For example:

```
# using Example 1 in 'LectureNotes3_notes.pdf' randomly
# generate observations from 2 covariance matrix; first 20
# observations have mean vector (3, 0) and second 20
# observations have mean vector (0, -4)
set.seed(123)
ce = matrix(rnorm(40 * 2), ncol = 2)
ce[1:20, 1] = ce[1:20, 1] + 3
ce[21:40, 2] = ce[1:20, 2] - 4
# clustering (average)
ce1 = hclust(dist(ce), method = "average")
# cut 5 clusters (set k: desired number of groups)
usingK <- cutree(ce1, k = 5)
usingK
```

```
##  [1] 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 2 3 1 1 4 4 4 5 5 4 4 4 5 4 4 4 4 5 4 5 4 4
## [39] 4 4
```

```
# cut 5 clusters (set h: heights where the tree should be
# cut)
lh = length(ce1$height)
usingH <- cutree(ce1, h = ce1$height[lh - 4])
usingH
```

```
##  [1] 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 2 3 1 1 4 4 4 5 5 4 4 4 5 4 4 4 4 5 4 5 4 4
## [39] 4 4
```

```
# check several groups by specifying the desired number of
# groups and cutting height are equal.
all.equal(usingK, usingH)
```

```
## [1] TRUE
```

Each k value or h value can be set within the `cutree` command to divide the cluster into five. When checking the above results, it can be seen that the clusters of the two are cut into five in common.

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

The advantage of the ggdendrogram{ggdendro} command over the R basic command 'plot' is that the R basic command 'plot' applies to the hclust object to generate the dendrogram, while the ggdendrogram{ggdendro} uses the hclust output to generate the dendrogram. In addition, ggdendrogram{ggdendro} can easily show leaf labels through a built-in function called leaf_labels, and can easily rotate the plot 90 degrees through a rotate built-in function.

Applied exercises

3. Please refer to the NYC flight data nycflights13 that has been discussed in the lecture notes and whose manual can be found at https://cran.r-project.org/web/packages/nycflights13/index.html. We will use flights, a tibble from nycflights13.

Select from flights observations that are for 3 carrier "UA", "AA" or "DL", for month 7 and 2, and for 4 features dep_delay, arr_delay, distance and air_time. Let us try to see if we can use the 4 features to identify if an observation belongs a specific carrier or a specific month. The following tasks and questions are based on the extracted observations. Note that you need to remove na's from the extracted observations.

```
# select row from flights, for which month is 7, 2, and
# carrier is UA, AA, DL, and for 4 features dep_delay,
# arr_delay, distance, air_time.
dat <- flights %>%
    select(carrier, month, dep_delay, arr_delay, distance, air_time) %>%
    filter(month %in% c(7, 2), carrier %in% c("UA", "AA", "DL"))
# remove rows that gave any NA
dat = na.omit(dat)
# check na in data
sum(is.na(dat))
```

```
## [1] 0
```

After selecting carrier, month, dep_delay, arr_delay, distance, and air_time from the flight dataset, three carrier and two month were used to extract observation confirm that the observation was well extracted. It does not have any na.

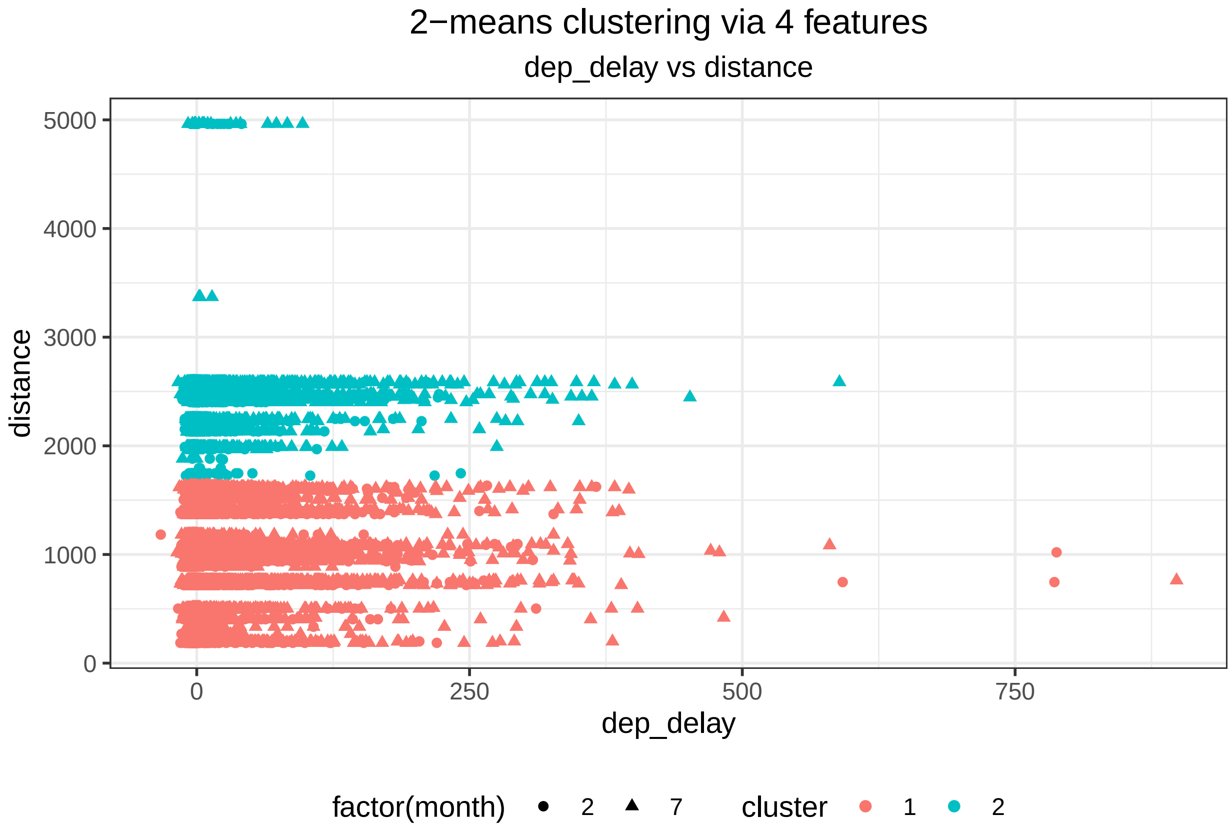
3.1) Apply K-means with $K = 2$ and 3 respectively but all with set.seed(1) and nstart=20. For $K = 3$, provide visualization of the clustering results based on true clusters given by carrier, whereas for $K = 2$, provide visualization of the clustering results based on true clusters given by month. Summarize your findings based on the clustering results. You can use the same visualization scheme that is provided by Example 2 in "LectureNotes3_notes.pdf". Try visualization based on different sets of 2 features if your visualization has overlayed points.

K-means clustering (K=2)

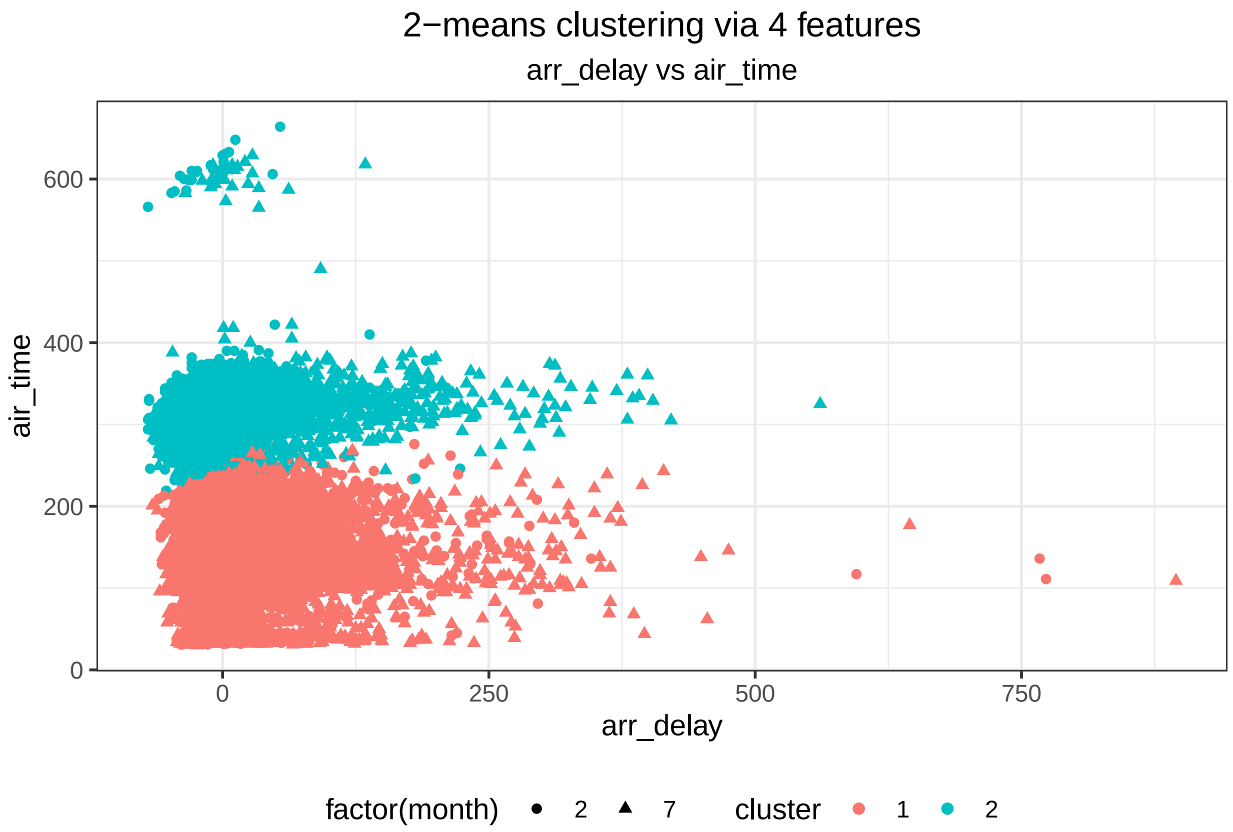
```
# set K=2 cluster
set.seed(1)
# K=2 kmeans use 4 features dep_delay, arr_delay, distance,
# air_time
km.out1 = kmeans(dat[, 3:6], 2, nstart = 20)
# show the totalss = betweenss + withinss
km.out1$totss
```

```
## [1] 11697443536
```

```
# augment flights data with K = 2 cluster
dat1 <- dat
dat1$cluster = factor(km.out1$cluster)
# plot the K=2 via 4 features; dep_delay vs distance and
# given by month
pk2 <- ggplot(dat1, aes(dep_delay, distance)) + geom_point(aes(shape = factor(month),
    color = cluster)) + theme_bw() + labs(title = "2-means clustering via 4 features",
    subtitle = "dep_delay vs distance") + theme(plot.title = element_text(hjust = 0.5),
    legend.position = "bottom", plot.subtitle = element_text(hjust = 0.5))
pk2
```



```r
# plot the K=2 via 4 features; arr_delay and air_time and
# given by month
pk2.1 <- ggplot(dat1, aes(arr_delay, air_time)) + geom_point(aes(shape = factor(month),
    color = cluster)) + theme_bw() + labs(title = "2-means clustering via 4 features",
    subtitle = "arr_delay vs air_time") + theme(plot.title = element_text(hjust = 0.5),
    legend.position = "bottom", plot.subtitle = element_text(hjust = 0.5))
pk2.1
```



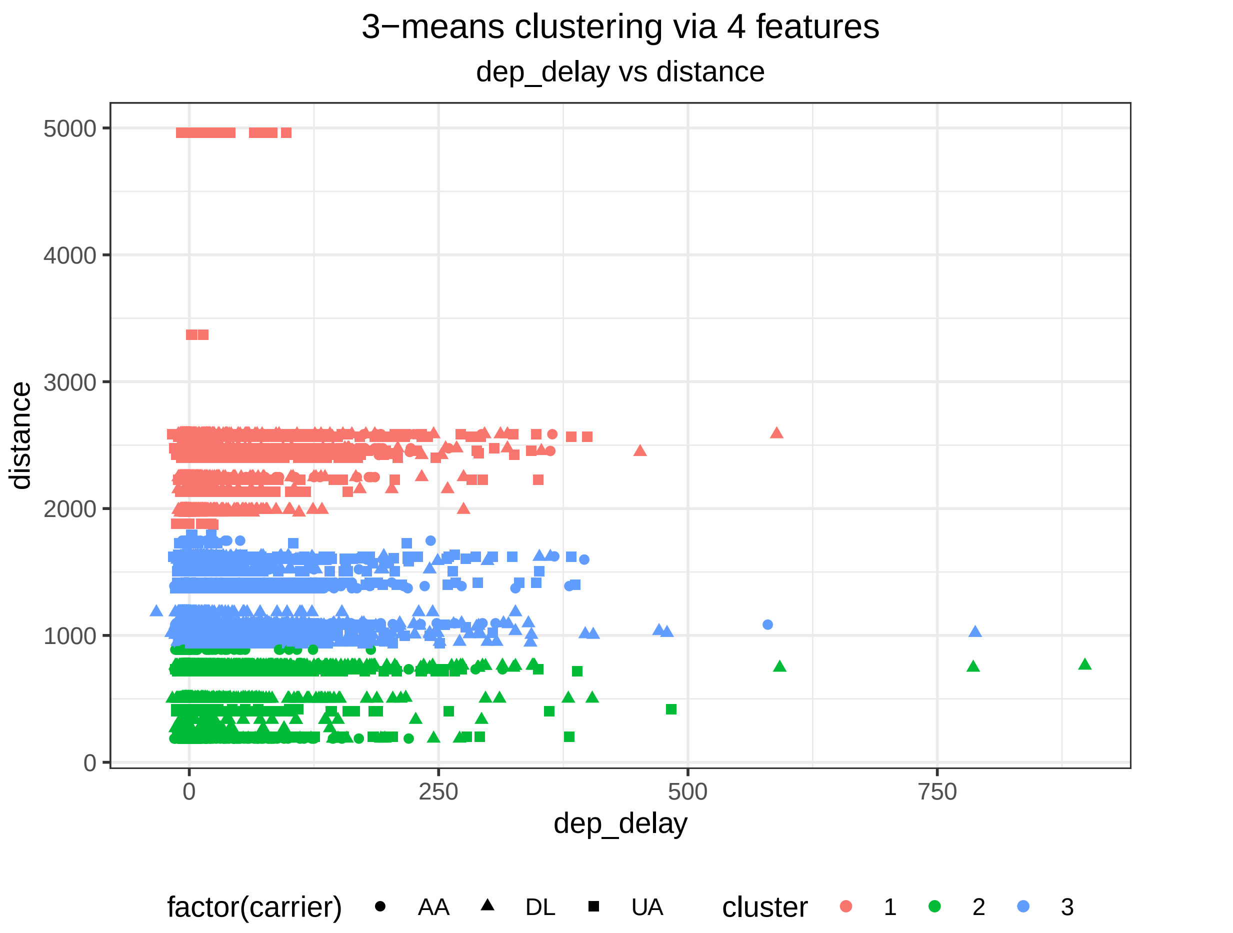
The above result is a visualization of clustering results based on the actual cluster provided by `month` with k-means applied as 2, and two plots were created based on the other two feature sets due to overlapping visualization. As a result of checking the total square sum, it can be seen that **11697443536** has a large variance between clusters and dense observations. This means that the cluster is well classified.
If you check the `dep_delay` vs `distance` plot, you can see that it was well classified into two clusters, and distance is an important measure for classifying two clusters.
If you look at the `air_time` vs `arr_delay` plot, you can see that it was well classified into two clusters, and air_time is an important measure for classifying two clusters.

K-means clustering (K=3)

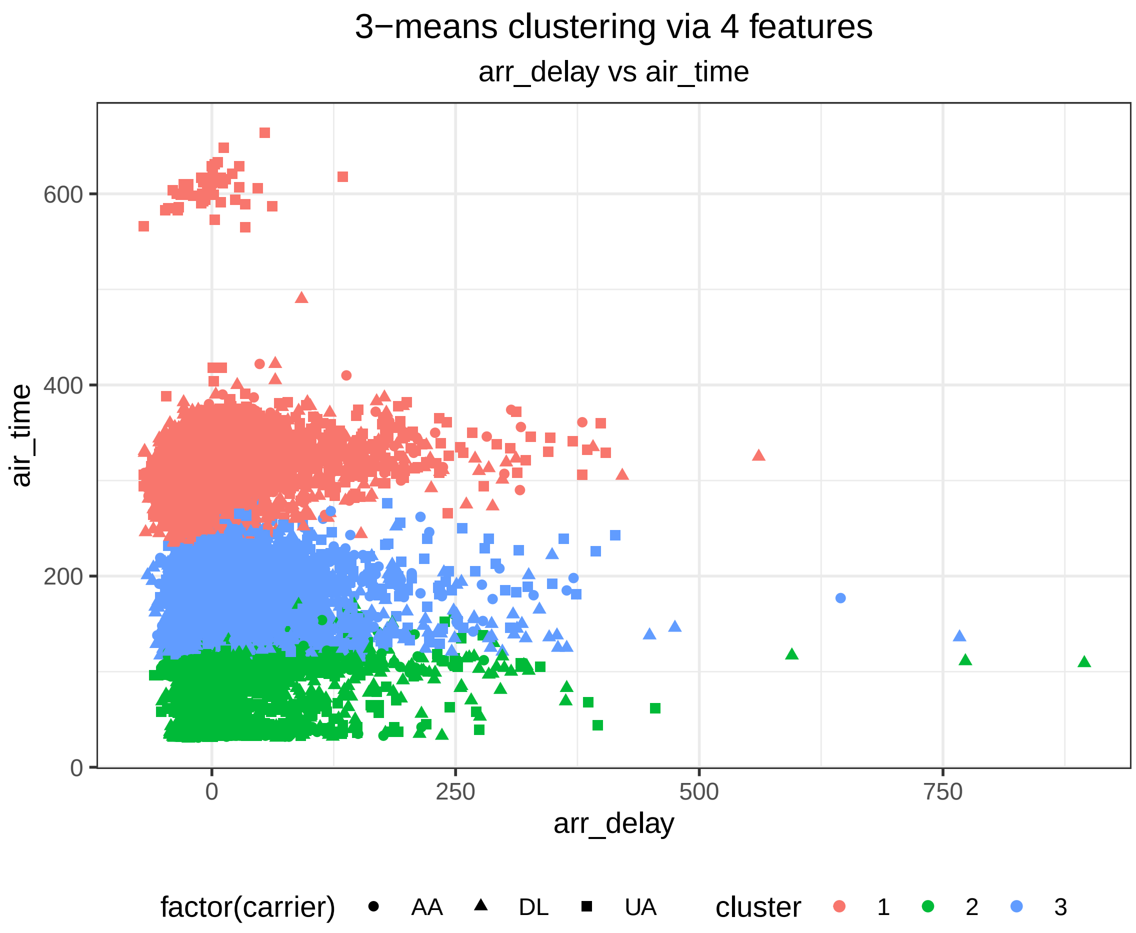
```r
# set K=3 cluster
set.seed(1)
# K=3 kmeans use 4 features dep_delay, arr_delay, distance,
# air_time
km.out2 = kmeans(dat[, 3:6], 3, nstart = 20)
# show the totalss = betweenss + withinss
km.out2$totss
```

```
## [1] 11697443536
```

```r
# augment flights data with K = 3 cluster
dat2 <- dat
dat2$cluster = factor(km.out2$cluster)
# plot the K=3 via 4 features; dep_delay vs distance and
# given by carrier and adding centroid points of cluster
pk3 <- ggplot(dat2, aes(dep_delay, distance)) + geom_point(aes(shape = factor(carrier),
    color = cluster)) + theme_bw() + labs(title = "3-means clustering via 4 features",
    subtitle = "dep_delay vs distance") + theme(plot.title = element_text(hjust = 0.5),
    legend.position = "bottom", plot.subtitle = element_text(hjust = 0.5))
pk3
```



```
# plot the K=3 via 4 features; arr_delay vs air_time and
# given by carrier
pk3.1 <- ggplot(dat2, aes(arr_delay, air_time)) + geom_point(aes(shape = factor(carrier),
    color = cluster)) + theme_bw() + labs(title = "3-means clustering via 4 features",
    subtitle = "arr_delay vs air_time") + theme(plot.title = element_text(hjust = 0.5),
    legend.position = "bottom", plot.subtitle = element_text(hjust = 0.5))
pk3.1
```



The above result is a visualization of clustering results based on the actual cluster provided by `carrier` by applying the k-means of 3, and two plots were created based on the other two feature sets due to overlapping visualization. As a result of checking the total square sum, it can be seen that **11697443536** has a large variance between clusters and dense observations. This means that the cluster is well classified.

If you check the `dep_delay` vs `distance` plot, you can see that it was well classified into three clusters, and distance is an important measure for classifying three clusters.

If you look at the `air_time` vs `arr_delay` plot, you can see that it was well classified into three clusters, and air time is an important measure for classifying three clusters.

3.2) Use set.seed(123) to randomly extract 50 observations, and to these 50 observations, apply hierarchical clustering with average linkage. (i) Cut the dendrogram to obtain 3 clusters with leafs annotated by carrier names and resulting clusters colored distinctly, and report the corresponding height of cut. (ii) In addition, cut the dendrogram to obtain 2 clusters with leafs annotated by month numbers and resulting clusters colored distinctly, and report the corresponding height of cut. Here are some hints: say, you save the randomly extracted 50 observations into an object ds3sd, for these observations save their carrier names by keeping their object type but save month numbers as a character vector, make sure that ds3sd is a matrix, transpose ds3sd into tmp, assign to tmp column names with their corresponding carrier names or month numbers, and then transpose tmp and save it as ds3sd; this way, you are done assigning cluster labels to each observation in ds3sd; then you are ready to use the commands in the file Plotggdendro.r to create the desired dendrograms.

```
set.seed(123)
# subset of data (randomly extract 50 observations) into
# object
n = dim(dat)[1]
ds3sd = sample(1:n, size = 50, replace = FALSE)
ds3sd = dat[ds3sd, ]
# change the month type (number as char)
ds3sd$month <- as.character(ds3sd$month)
# change the tibble to matrix
ds3sd <- as.matrix(ds3sd)
# transpose ds3sd into tmp and tmp column names with
# carrier names.
tmp <- t(ds3sd)
colnames(tmp) <- tmp[1, ]
tmp <- tmp[3:6, ]
# transpose tmp as ds3sd1
ds3sd1 <- t(tmp)
# transpose ds3sd into tmp1 and tmp1 column names with
# month names.
tmp1 <- t(ds3sd)
colnames(tmp1) <- tmp1[2, ]
tmp1 <- tmp1[3:6, ]
# transpose tmp1 as ds3sd2
ds3sd2 <- t(tmp1)
```

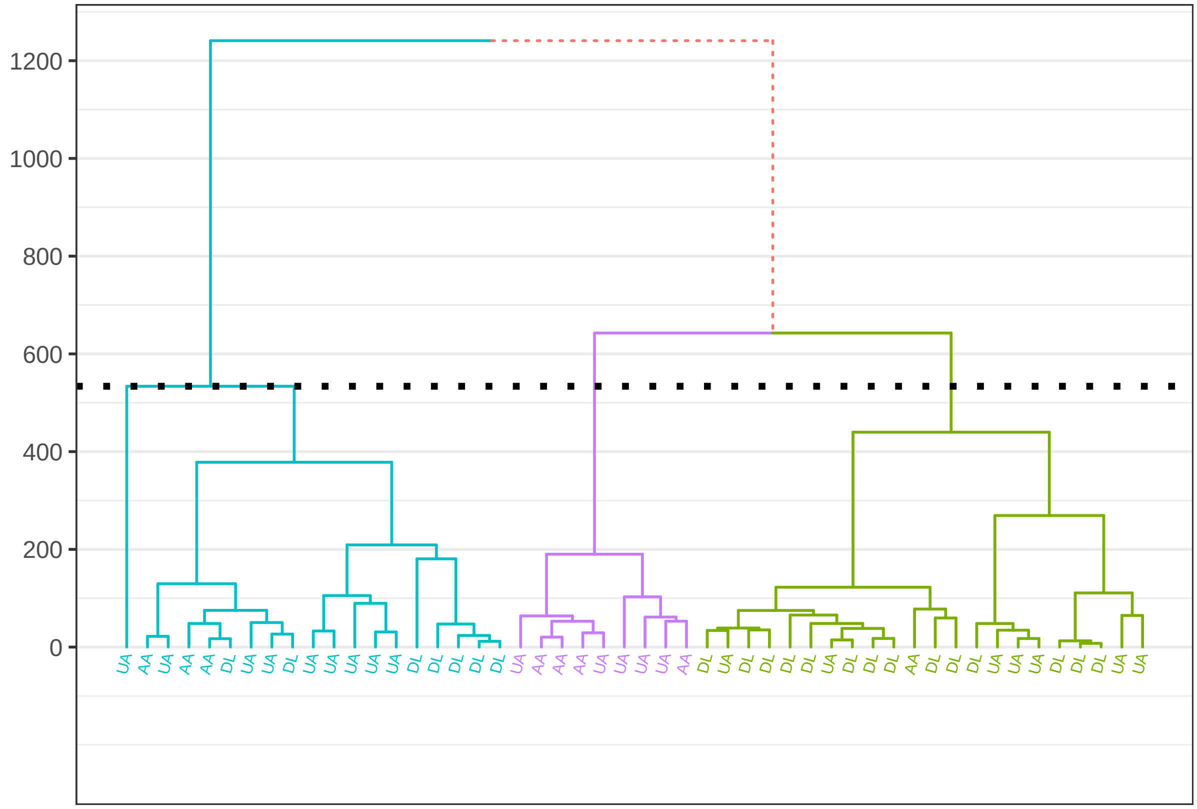
Each data was set as a matrix to set up a dendrogram by cutting three clusters annotated with carrier name and two clusters annotated with month number. (Initial work)

```
# (i) Cut the dendrogram to obtain 3 clusters with leafs
# annotated by carrier names and resulting clusters colored
# distinctly, and report the corresponding height of cut.
# import source file Plotggdendro.r
source("Plotggdendro.r", encoding = "UTF-8")
# average linkage
```

```
hc.carrier = hclust(dist(ds3sd1), method = "average")
# cut the obtain 3 clusters with leafs annotated by carrier
cutheight_CA <- hc.carrier$height[length(hc.carrier$height) -
    2]
# check boolean cut tree to obtain 3 clusters
all.equal(cutree(hc.carrier, h = cutheight_CA), cutree(hc.carrier,
    k = 3))
```

```
## [1] TRUE
```

```
# show plot using Plotggdendro.r
dendro_CA <- plot_ggdendro(dendro_data_k(hc.carrier, 3), direction = "tb",
    heightReferece = cutheight_CA, expand.y = 0.2, label.size = 2,
    branch.size = 0.5)
dendro_CA
```



```
# show height of cut of 3 clusters with leafs annotated by
# carrier (numeric)
cutheight_CA
```

```
## [1] 533.7758
```

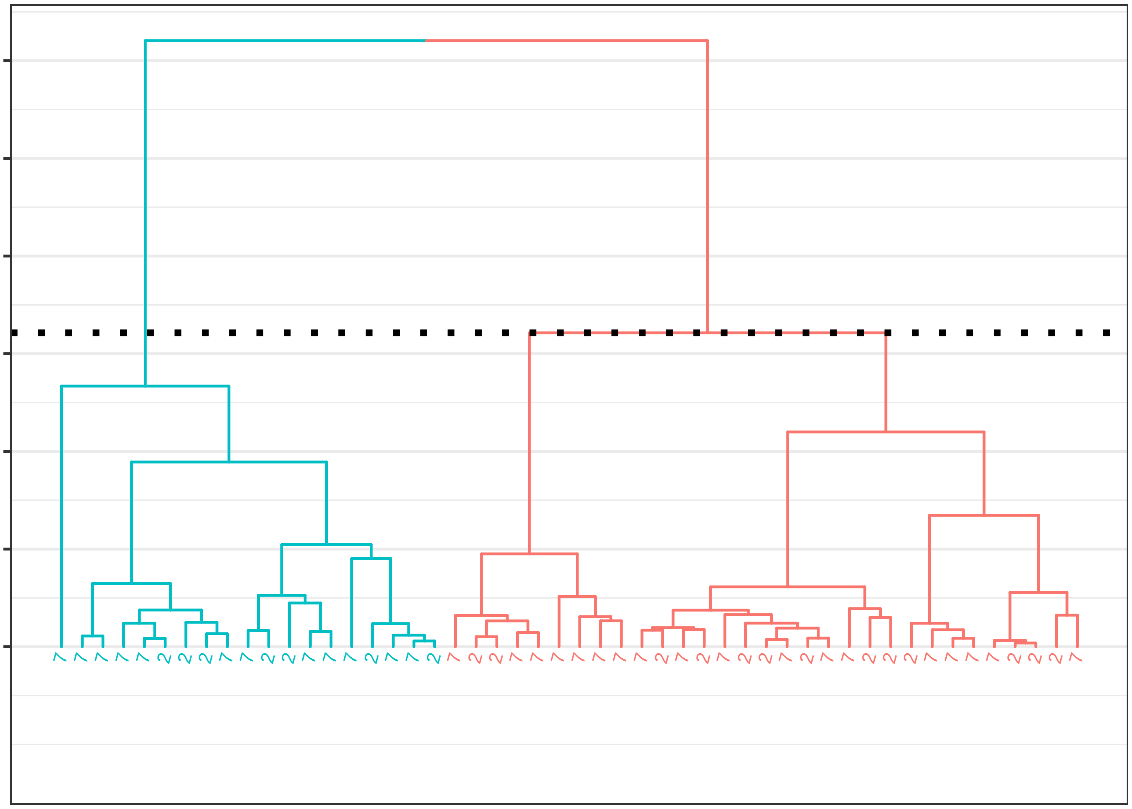
As a result of performing a hierarchical clustering for `carrier`, it can be confirmed that three clusters are well classified. It can also be seen that the cutting height is **533.7758**.

```r
# (ii) In addition, cut the dendrogram to obtain 2 clusters
# with leafs annotated by month numbers and resulting
# clusters colored distinctly, and report the corresponding
# height of cut.

# import source file Plotggdendro.r
source("Plotggdendro.r", encoding = "UTF-8")
# average linkage
hc.month = hclust(dist(ds3sd2), method = "average")
# cut the obtain 2 clusters with leafs annotated by month
# numbers
cutheight_MT <- hc.month$height[length(hc.month$height) - 1]
# check boolean cut tree to obtain 2 clusters
all.equal(cutree(hc.month, h = cutheight_MT), cutree(hc.month,
    k = 2))
```

```
## [1] TRUE
```

```r
# show plot using Plotggdendro.r
dendro_MT <- plot_ggdendro(dendro_data_k(hc.month, 2), direction = "tb",
    heightReferece = cutheight_MT, expand.y = 0.2, label.size = 2.5,
    branch.size = 0.5)
dendro_MT
```



```
# show height of cut of 2 clusters with leafs annotated by
# month numbers (numeric)
cutheight_MT
```

```
## [1] 642.6994
```

As a result of performing a hierarchical clustering for `month`, it can be confirmed that two clusters are well classified. It can also be seen that the cutting height is **642.6994**.