HW4

Yifei Chen

December 2, 2019

Honor Code:"The codes and results derived by using these codes constitutemy own work. I have consulted the following resources regarding this assignment: Chao Cheng

1

```
# a
data = read.csv('customers_data.csv', header = TRUE)
summary(data)
```

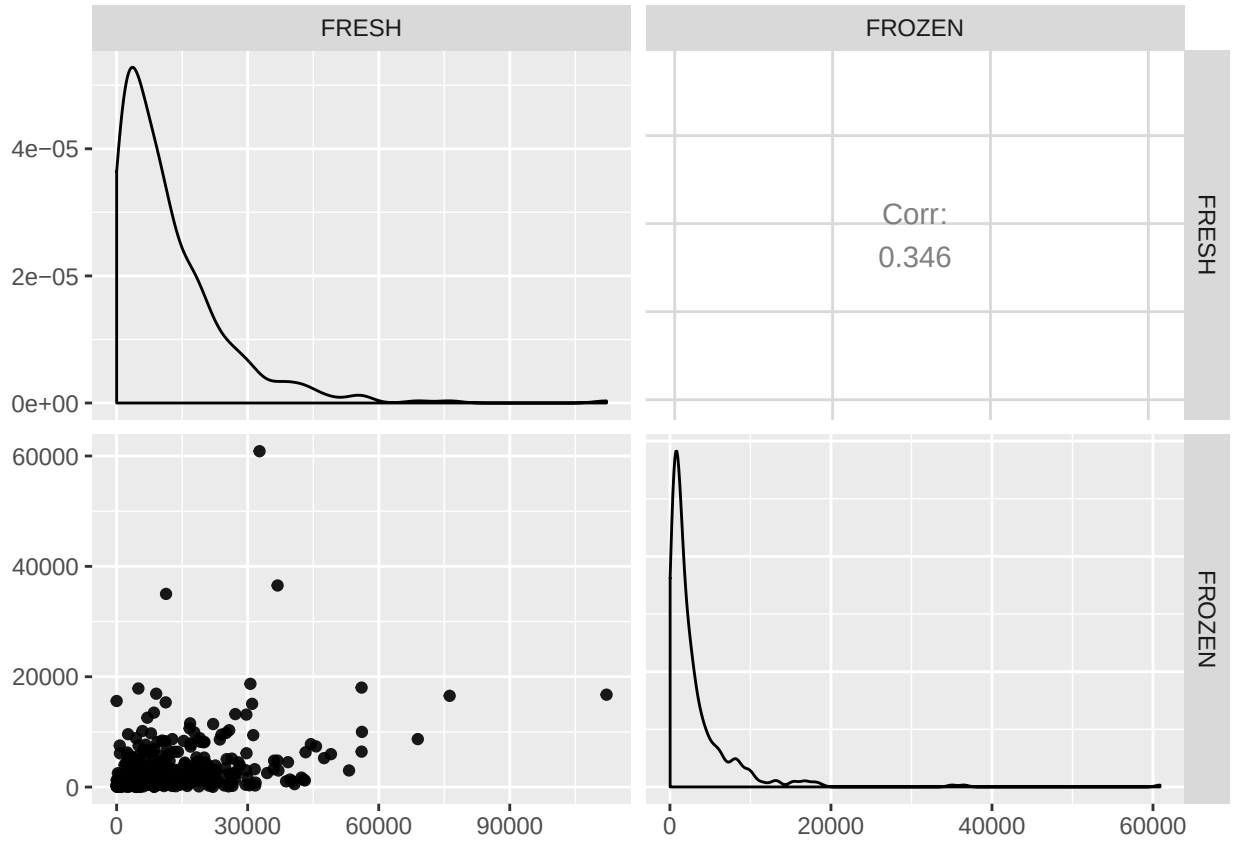
```
##     Channel          Region          Fresh             Milk
##  Min.   :1.000   Min.   :1.000   Min.   :     3   Min.   :   55
##  1st Qu.:1.000   1st Qu.:2.000   1st Qu.:  3128   1st Qu.: 1533
##  Median :1.000   Median :3.000   Median :  8504   Median : 3627
##  Mean   :1.323   Mean   :2.543   Mean   : 12000   Mean   : 5796
##  3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.: 16934   3rd Qu.: 7190
##  Max.   :2.000   Max.   :3.000   Max.   :112151   Max.   :73498
##     Grocery          Frozen        Detergents_Paper    Delicassen
##  Min.   :    3   Min.   :   25.0   Min.   :    3.0   Min.   :    3.0
##  1st Qu.: 2153   1st Qu.:  742.2   1st Qu.:  256.8   1st Qu.:  408.2
##  Median : 4756   Median : 1526.0   Median :  816.5   Median :  965.5
##  Mean   : 7951   Mean   : 3071.9   Mean   : 2881.5   Mean   : 1524.9
##  3rd Qu.:10656   3rd Qu.: 3554.2   3rd Qu.: 3922.0   3rd Qu.: 1820.2
##  Max.   :92780   Max.   :60869.0   Max.   :40827.0   Max.   :47943.0
```

```
# b
customers_2 = data.frame('FRESH' = data$Fresh, 'FROZEN' = data$Frozen)
# c
library(GGally)
```

```
## Loading required package: ggplot2
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
ggpairs(customers_2, 1:2, mapping = ggplot2::aes(alpha = 0.5),
        diag = list(continuous = wrap("densityDiag")),
        lower= list(continuous = wrap("points", alpha=0.9)))
```



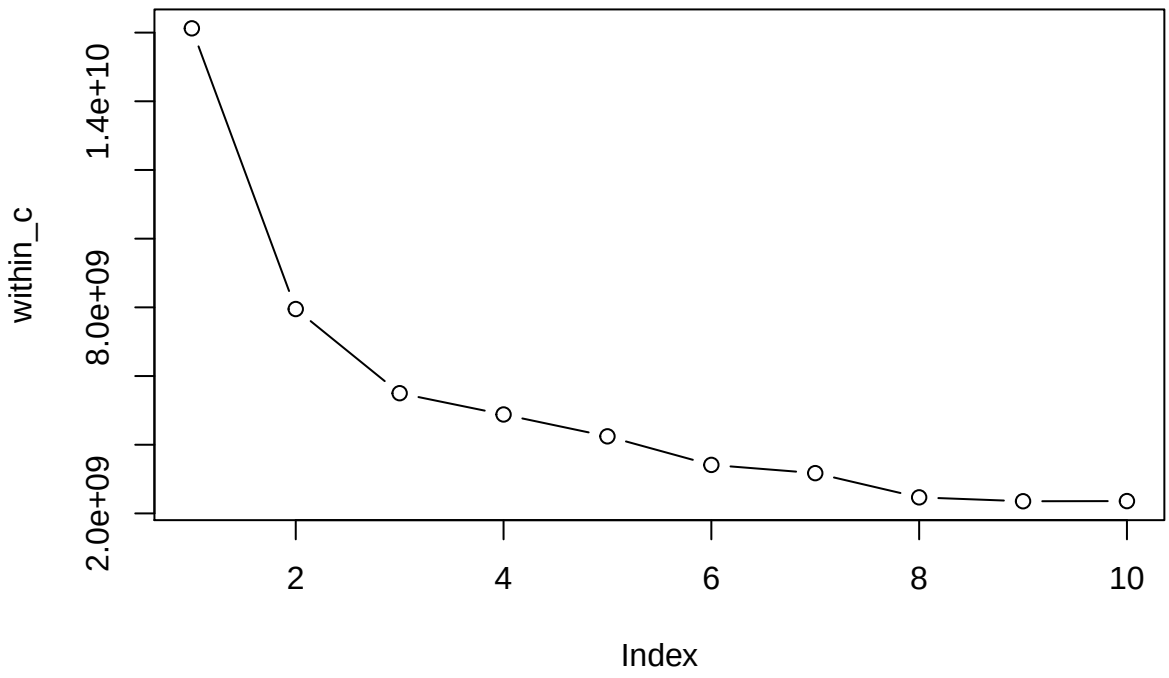
2

```
## Choice of k studying the within variance
# functions
n=dim(customers_2)[1]

## Choice of k studying the within variance

withinss <- function(group, x, centers, assignments) {
  cent <- centers[group, ]
  m <- rbind(cent, x[assignments==group, 1:2])
  sum((as.matrix(dist(m))[1, ])^2)
}

predict.kmeans <- function(object,
                           newdata,
                           method = c("centers", "classes")) {
  method <- match.arg(method)

  centers <- object$centers
  ss_by_center <- apply(centers, 1, function(x) {
    colSums((t(newdata) - x) ^ 2)
  })
  best_clusters <- apply(ss_by_center, 1, which.min)

  if (method == "centers") {
    centers[best_clusters, ]
```

```
  } else {
    best_clusters
  }
}

set.seed(12)
train_p=0.8;
# run 100 times
N=100
# k= 1,...,10
n_clust=10
within_c_n=matrix(0,n_clust,N)
within_c=vector('numeric',0)
dfxy=customers_2

for(cl in 1:n_clust){
  for(iter in 1:N){
    # draw randomly the 80% of observations in customer_2 and use them as a training data-set.
    train=sample(1:n,train_p*n)
    train=sort(train)
    test=sort(setdiff(1:n,train))
    # run the function kmeans on the training data-set with k centers,20 random starts and 100 maximum
    dfxy.km_train=kmeans(dfxy[train,1:2],centers=cl, nstart = 20, iter.max = 100)
    centers <- dfxy.km_train$centers
    # use the estimated centers to allocate the observations of the test data-set to a specific group a
    assignments <-  as.numeric(row.names(predict.kmeans(dfxy.km_train, (dfxy[test,1:2]))))
    # calculate the deviance within estimated groups in the test data-set.
    within_c_n[cl,iter]=sum(sapply(seq(nrow(centers)), function(y){withinss(group=y,x = dfxy[test,1:2],
  }
  # For each k, average the deviance within groups over the 100 runs
  within_c[cl]=(mean(within_c_n[cl,]))
}

within_c
```

```
##  [1] 16123668468  7951142311  5500510756  4880624956  4244403334
##  [6]  3412735472  3173091650  2467738735  2354705337  2358766521
```

```
# plot this average over the number of clusters
plot(within_c, type = "b")
```



From the plot, we can see that 3 can be considered as an indicator of the appropriate number of clusters

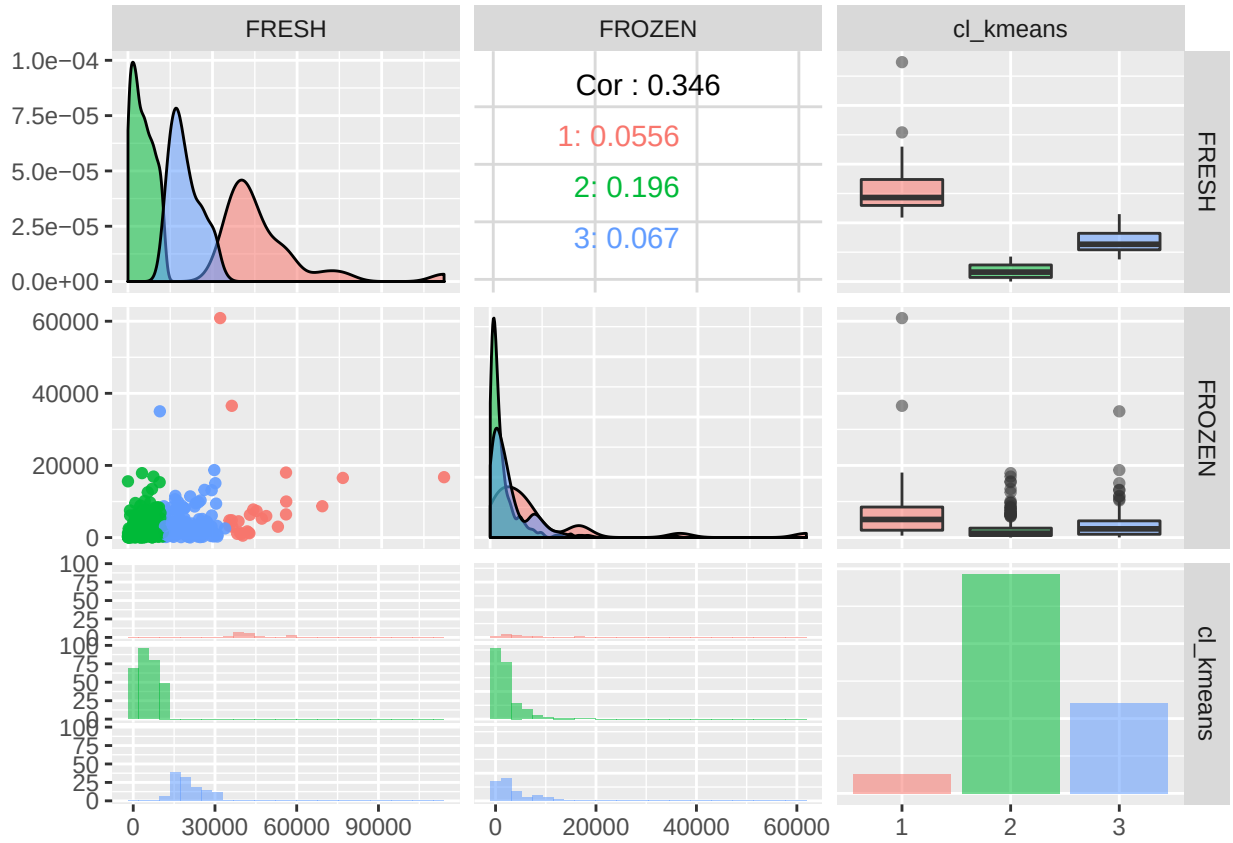
```
# re-apply kmeans with the selected number of clusters,20 random starts and 100 maximum iterations, and
dfxy.km = kmeans(dfxy[,1:2], centers=3, nstart = 20, iter.max = 100)
dfxy.km$cluster
```

```
##   [1] 2 2 2 3 3 2 2 2 2 2 2 3 3 3 3 2 2 2 3 2 3 2 3 3 3 3 2 3 2 1 3 2 3 3 2
##  [36] 2 3 3 2 1 3 3 2 2 2 2 2 1 2 2 2 2 1 2 3 2 2 2 3 2 2 1 2 2 2 2 2 3 2 2
##  [71] 3 3 2 3 2 3 2 2 2 2 2 2 2 3 2 3 3 1 2 3 2 3 2 3 2 2 2 2 2 2 2 2 2 1 3
## [106] 3 2 2 2 2 2 2 3 3 3 2 2 2 3 2 3 2 2 2 1 1 3 3 2 1 2 2 3 2 2 2 2 2 3 2
## [141] 3 3 1 2 3 3 2 2 2 3 3 2 3 2 2 2 2 3 2 2 2 2 3 2 2 3 2 2 2 2 2 2 2 2 2
## [176] 2 1 2 2 2 2 1 2 1 2 2 2 2 2 2 3 3 2 2 2 3 3 2 2 2 2 2 3 2 2 2 2 2 2 2
## [211] 3 2 2 2 2 2 2 2 3 2 2 3 2 2 2 2 2 3 2 2 2 2 2 3 2 3 2 2 3 2 1 3 3 3 2 2
## [246] 2 2 3 3 2 2 2 2 2 3 2 3 2 2 1 1 2 2 3 2 2 2 2 3 2 3 2 2 2 1 2 2 3 2 2 3
## [281] 2 2 1 3 1 1 2 3 3 1 2 2 2 2 3 2 3 2 2 2 3 2 2 2 2 2 2 3 2 2 2 3 2 2 2
## [316] 2 2 2 2 2 2 2 3 3 3 1 2 2 3 2 2 2 3 2 3 3 3 2 2 2 2 2 2 2 2 2 2 3 2 2
## [351] 2 2 2 2 3 2 3 2 2 2 3 2 2 2 2 2 2 2 3 2 1 3 2 3 2 2 2 1 2 2 3 3 3 2 2
## [386] 2 2 3 2 2 2 2 2 3 2 2 2 2 2 2 2 3 3 3 3 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2
## [421] 2 3 3 3 3 2 3 3 2 2 2 2 3 2 3 3 1 3 2 2
```

```
# provide a scatter-plot matrix of FRESH and FROZEN conditional on the estimated cluster memberships vi
customers_2$cl_kmeans = as.factor(dfxy.km$cluster)
ggpairs(customers_2, 1:3, mapping = ggplot2::aes(color = cl_kmeans, alpha = 0.5),
        diag = list(continuous = wrap("densityDiag")),
        lower=list(continuous = wrap("points", alpha=0.9)))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

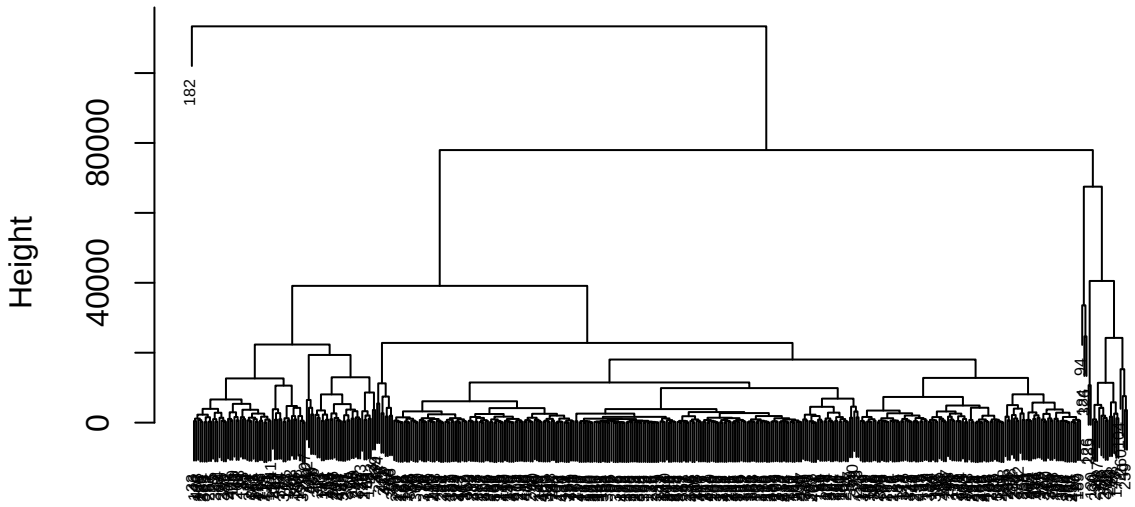
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



From the density graph of the FRESH column, we can see that most of the data belongs to the first cluster, while only several data is counted as the third cluster. Same results as the density graph of the FROZEN column. From the boxplot graph of the FRESH column, we can see that the values in the third cluster are the highest, while the data in the first cluster has the lowest values. However, from the boxplot graph of the FROZEN column, we can see that the values from different clusters are mostly within the same range, but the values of the third cluster are more scatter comparing to the other two.

3

```
# Estimate a hierachical partition on FRESH and FROZEN by the complete linkage usingthe number of clust
ass_hclust=function(i,method='complete',test){
  dist_hclust=as.matrix(dist(rbind(dfxy[test,1:2],dfxy[assignment_train==i,1:2])))
  n_dist=length(test)+length(which(assignment_train==i))
  dist_hclust=as.matrix(dist_hclust)
  if(method=='complete'){
    return(sapply(1:length(test),function(j){max(dist_hclust[(length(test)+1):n_dist,j])}))}
  if(method=='single'){
    return(sapply(1:length(test),function(j){min(dist_hclust[(length(test)+1):n_dist,j])}))}
  if(method=='average'){
    return(sapply(1:length(test),function(j){mean(dist_hclust[(length(test)+1):n_dist,j])}))}
}
dfxy.complete = hclust(dist(customers_2), method = "complete")
plot(dfxy.complete ,main = "Complete Linkage", xlab="", sub ="", cex =.5)
```

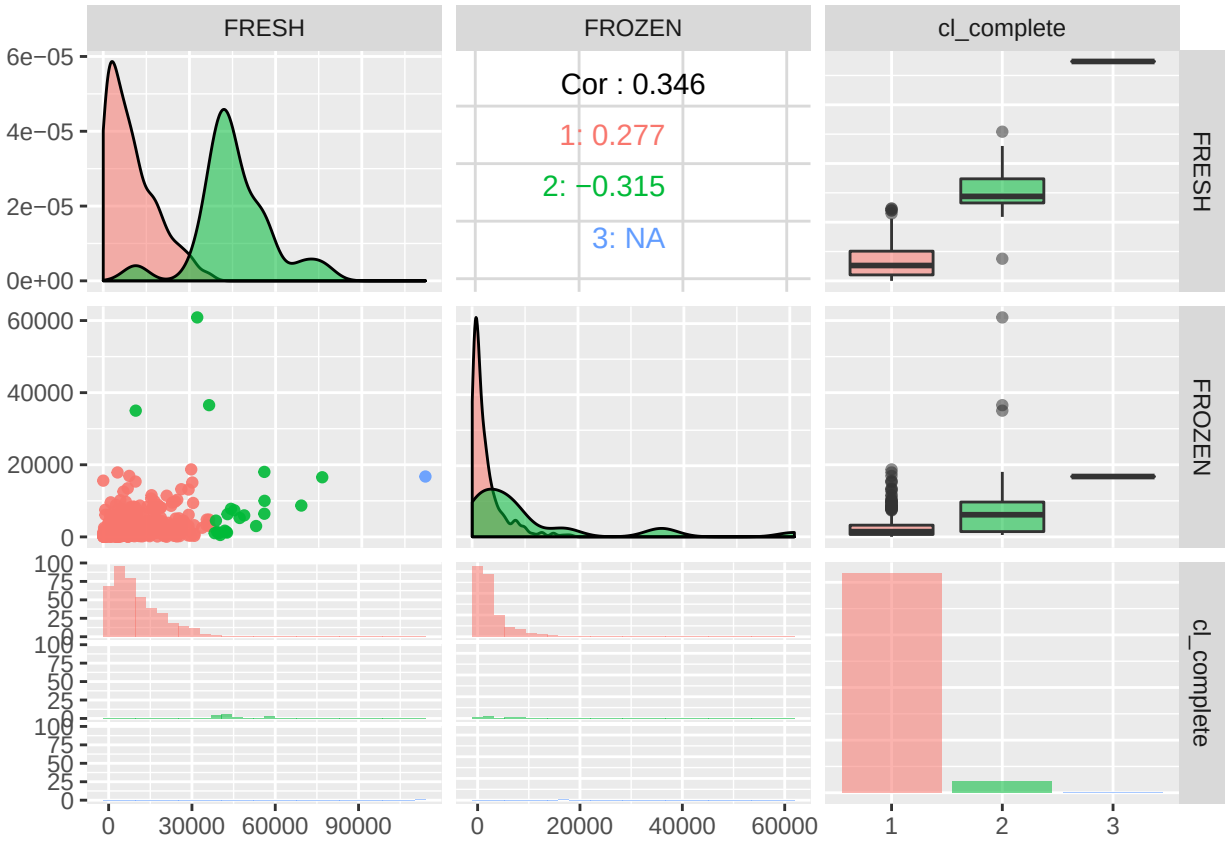



```
# Derive the estimated cluster memberships.
# choose k=3
assignment=cutree(dfxy.complete, 3)
table(assignment)
```

```
## assignment
##   1   2   3
## 417  22   1
```

```
# Provide a scatter-plot matrix of FRESH and FROZEN conditional on the estimated cluster memberships vi
customers_2$cl_complete=as.factor(assignment)
ggpairs(customers_2, c(1:2,4), mapping = ggplot2::aes(color = cl_complete, alpha = 0.5),
        diag = list(continuous = wrap("densityDiag")),
        lower=list(continuous = wrap("points", alpha=0.9)))
```

```
## Warning: Groups with fewer than two data points have been dropped.

## Warning: Groups with fewer than two data points have been dropped.

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



From the scatter plot of FRESH, we can see that there is only one value on the third cluster, and the value is larger than the those in the first and the second cluster. In addition, from the density graph of the FRESH column, we observe that most of the data belongs to the first cluster, then the second cluster. Same as the FROZEN column. By looking at the boxplot of the FRESH column, we can see that the value in the third cluster is the largest, and the data in the first cluster has the lowest values. And from the boxplot of the FROZEN column, the values in the first and the second clusters are within the same range, but the second cluster is more spread compared to the first one. By comparing this outcome to the k-means one, we can see that the number of values in the third cluster are obviously less and larger than the k-means one. However, the density and the range of the first and second clusters in both outcomes are similar.

R Appendix

```
knitr::opts_chunk$set(echo = TRUE)
# a
data = read.csv('customers_data.csv', header = TRUE)
summary(data)
# b
customers_2 = data.frame('FRESH' = data$Fresh, 'FROZEN' = data$Frozen)
# c
library(GGally)
ggpairs(customers_2, 1:2, mapping = ggplot2::aes(alpha = 0.5),
        diag = list(continuous = wrap("densityDiag")),
        lower= list(continuous = wrap("points", alpha=0.9)))
## Choice of k studying the within variance
# functions
```

```r
n=dim(customers_2)[1]

## Choice of k studying the within variance

withinss <- function(group, x, centers, assignments) {
  cent <- centers[group, ]
  m <- rbind(cent, x[assignments==group, 1:2])
  sum((as.matrix(dist(m))[1, ])^2)
}

predict.kmeans <- function(object,
                           newdata,
                           method = c("centers", "classes")) {
  method <- match.arg(method)

  centers <- object$centers
  ss_by_center <- apply(centers, 1, function(x) {
    colSums((t(newdata) - x) ^ 2)
  })
  best_clusters <- apply(ss_by_center, 1, which.min)

  if (method == "centers") {
    centers[best_clusters, ]
  } else {
    best_clusters
  }
}

set.seed(12)
train_p=0.8;
# run 100 times
N=100
# k= 1,...,10
n_clust=10
within_c_n=matrix(0,n_clust,N)
within_c=vector('numeric',0)
dfxy=customers_2

for(cl in 1:n_clust){
  for(iter in 1:N){
    # draw randomly the 80% of observations in customer_2 and use them as a training data-set.
    train=sample(1:n,train_p*n)
    train=sort(train)
    test=sort(setdiff(1:n,train))
    # run the function kmeans on the training data-set with k centers,20 random starts and 100 maximum
    dfxy.km_train=kmeans(dfxy[train,1:2],centers=cl, nstart = 20, iter.max = 100)
    centers <- dfxy.km_train$centers
    # use the estimated centers to allocate the observations of the test data-set to a specific group a
    assignments <-  as.numeric(row.names(predict.kmeans(dfxy.km_train, (dfxy[test,1:2]))))
    # calculate the deviance within estimated groups in the test data-set.
    within_c_n[cl,iter]=sum(sapply(seq(nrow(centers)), function(y){withinss(group=y,x = dfxy[test,1:2],
  }
  # For each k, average the deviance within groups over the 100 runs
```

```
  within_c[cl]=(mean(within_c_n[cl,]))
}

within_c
# plot this average over the number of clusters
plot(within_c, type = "b")
# re-apply kmeans with the selected number of clusters,20 random starts and 100 maximum iterations, and
dfxy.km = kmeans(dfxy[,1:2], centers=3, nstart = 20, iter.max = 100)
dfxy.km$cluster
# provide a scatter-plot matrix of FRESH and FROZEN conditional on the estimated cluster memberships vi
customers_2$cl_kmeans = as.factor(dfxy.km$cluster)
ggpairs(customers_2, 1:3, mapping = ggplot2::aes(color = cl_kmeans, alpha = 0.5),
        diag = list(continuous = wrap("densityDiag")),
        lower=list(continuous = wrap("points", alpha=0.9)))
# Estimate a hierachical partition on FRESH and FROZEN by the complete linkage usingthe number of clust
ass_hclust=function(i,method='complete',test){
  dist_hclust=as.matrix(dist(rbind(dfxy[test,1:2],dfxy[assignment_train==i,1:2])))
  n_dist=length(test)+length(which(assignment_train==i))
  dist_hclust=as.matrix(dist_hclust)
  if(method=='complete'){
    return(sapply(1:length(test),function(j){max(dist_hclust[(length(test)+1):n_dist,j])}))}
  if(method=='single'){
    return(sapply(1:length(test),function(j){min(dist_hclust[(length(test)+1):n_dist,j])}))}
  if(method=='average'){
    return(sapply(1:length(test),function(j){mean(dist_hclust[(length(test)+1):n_dist,j])}))}
}
dfxy.complete = hclust(dist(customers_2), method = "complete")
plot(dfxy.complete ,main = "Complete Linkage", xlab="", sub ="", cex =.5)
# Derive the estimated cluster memberships.
# choose k=3
assignment=cutree(dfxy.complete, 3)
table(assignment)
# Provide a scatter-plot matrix of FRESH and FROZEN conditional on the estimated cluster memberships vi
customers_2$cl_complete=as.factor(assignment)
ggpairs(customers_2, c(1:2,4), mapping = ggplot2::aes(color = cl_complete, alpha = 0.5),
        diag = list(continuous = wrap("densityDiag")),
        lower=list(continuous = wrap("points", alpha=0.9)))
```