INFS692 Final Project: Model 3

Yanfei Chen

2022-12-15

Helper packages

```
library(rsample)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(readr)
library(factoextra)

## Loading required package: ggplot2

## Welcome! Want to learn more? See two factoextra-related books at
https://goo.gl/ve3WBa

library(cluster)
library(stringr)
library(gridExtra)

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine

library(mclust)

## Package 'mclust' version 6.0.0
## Type 'citation("mclust")' for citing this R package in publications.

library(tidyverse)

## ── Attaching packages
## ─────────────────────────────────────
## tidyverse 1.3.2 ──
```

```
## ✓ tibble  3.1.8     ✓ purrr   0.3.5
## ✓ tidyr   1.2.1     ✓ forcats 0.5.2
## ── Conflicts ─────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ gridExtra::combine() masks dplyr::combine()
## ✖ dplyr::filter()      masks stats::filter()
## ✖ dplyr::lag()         masks stats::lag()
## ✖ purrr::map()         masks mclust::map()
```

Preprocess data

Load dataset

```
data <- read_csv("/Users/chenyanfei/Desktop/radiomics_completedata.csv")

## Rows: 197 Columns: 431
## ── Column specification
──────────────────────────────────────────────────────
## Delimiter: ","
## chr   (1): Institution
## dbl (430): Failure.binary, Failure, Entropy_cooc.W.ADC, GLNU_align.H.PET,
Mi...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.

data$Failure.binary = as.factor(data$Failure.binary)
```

Check for null/missing

```
data_clean <- na.omit(data)
dim(data)

## [1] 197 431

dim(data_clean)

## [1] 197 431

# There's no null/missing value in the dataset.
```

Normalize the continuous variables

```
nor_data <- scale(data_clean[c(3:431)])
# combine with the categorical variables
new_data <- cbind(data_clean[2], nor_data)
# change label type
levels(new_data$Failure.binary)=c("No","Yes")
new_data %>%
  mutate(Failure.binary = factor(Failure.binary,
```

```
                          labels = make.names(levels(Failure.binary))))
# all features
Features <- data.matrix(new_data[,-1])
```

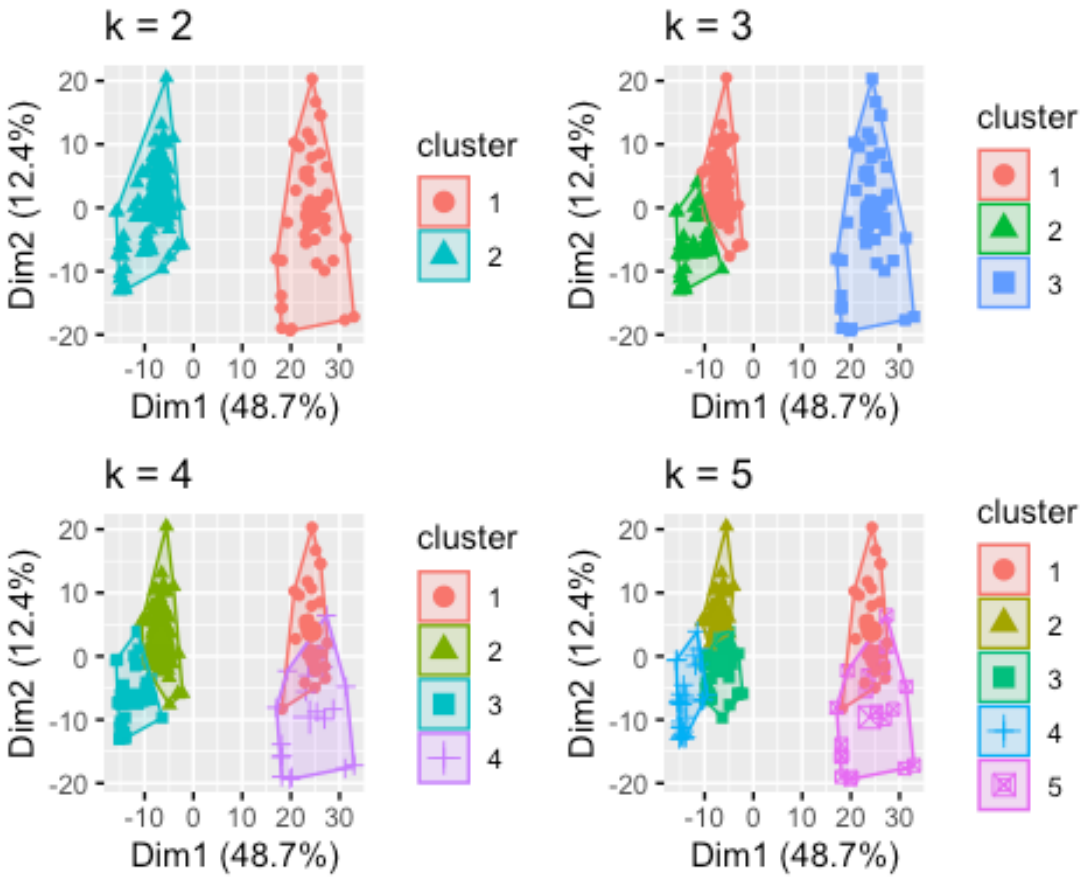
Split the data into training and testing

```
data_split <- initial_split(new_data, prop = .8, strata = "Failure.binary")
data_train <- training(data_split)
data_test  <- testing(data_split)
```

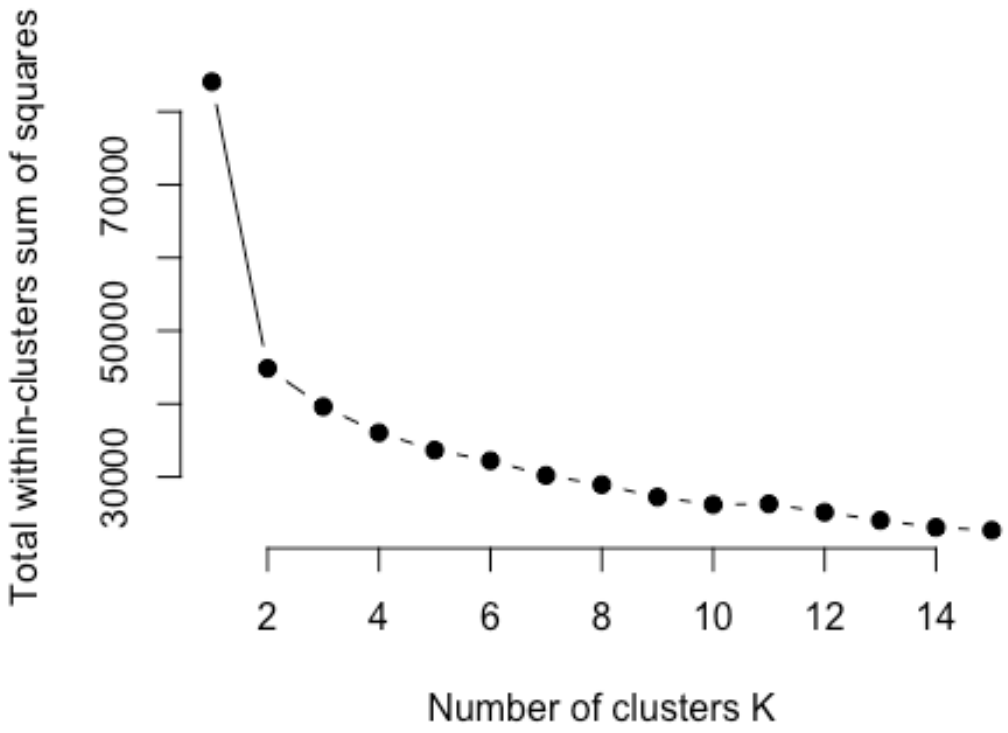
Model 3

K-Means

```
k2 <- kmeans(Features, centers = 2, nstart = 25)
k3 <- kmeans(Features, centers = 3, nstart = 25)
k4 <- kmeans(Features, centers = 4, nstart = 25)
k5 <- kmeans(Features, centers = 5, nstart = 25)

p1 <- fviz_cluster(k2, geom = "point", data = Features) + ggtitle("k = 2")
p2 <- fviz_cluster(k3, geom = "point",  data = Features) + ggtitle("k = 3")
p3 <- fviz_cluster(k4, geom = "point",  data = Features) + ggtitle("k = 4")
p4 <- fviz_cluster(k5, geom = "point",  data = Features) + ggtitle("k = 5")
grid.arrange(p1, p2, p3, p4, nrow = 2)
```



```r
#Determining Optimal Number of Clusters
set.seed(123)
# Compute and plot wss for k = 1 to k = 15
k.values <- 1:15
#function to compute total within-cluster sum of square
wss <- function(k) {
  kmeans(Features, k, nstart = 10 )$tot.withinss
}
# extract wss for 2-15 clusters
wss_values <- map_dbl(k.values, wss)

plot(k.values, wss_values,
     type="b", pch = 19, frame = FALSE,
     xlab="Number of clusters K",
     ylab="Total within-clusters sum of squares")
```




```r
# Compute k-means clustering with k = 2
set.seed(123)
final <- kmeans(Features, 2, nstart = 25)

#final data
fviz_cluster(final, data = Features)
```

Hierarchical

```
set.seed(123)

# Dissimilarity matrix
d <- dist(Features, method = "euclidean")

# Hierarchical clustering using Complete Linkage
hc1 <- hclust(d, method = "complete" )

set.seed(123)

# Compute maximum or complete linkage clustering with agnes
hc2 <- agnes(Features, method = "complete")

# Agglomerative coefficient
hc2$ac

## [1] 0.8489113

# methods to assess
m <- c( "average", "single", "complete", "ward")
names(m) <- c( "average", "single", "complete", "ward")
```

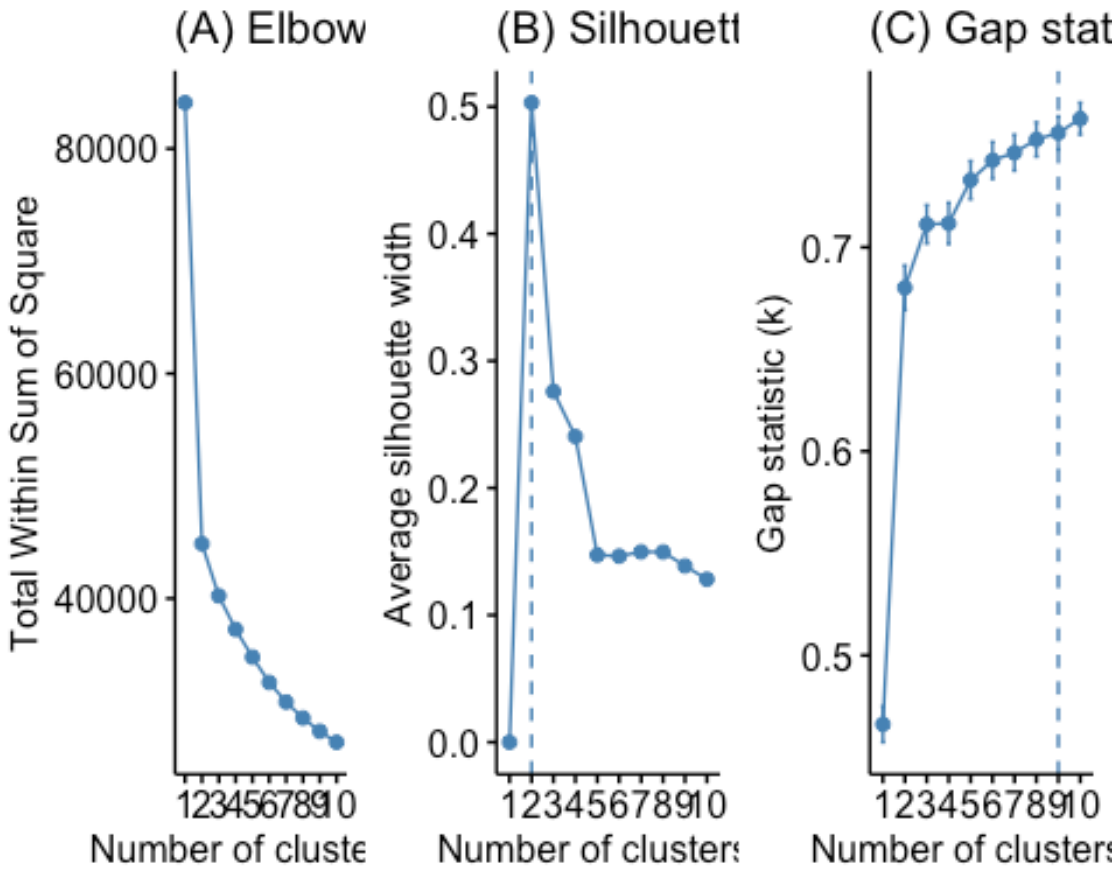
```r
# function to compute coefficient
ac <- function(x) {
  agnes(Features, method = x)$ac
}

# get agglomerative coefficient for each linkage method
purrr::map_dbl(m, ac)
```

```
##   average    single  complete      ward
## 0.7616680 0.7098672 0.8489113 0.9654737
```

```r
# compute divisive hierarchical clustering
hc4 <- diana(Features)

# Divise coefficient; amount of clustering structure found
hc4$dc
```

```
## [1] 0.8428381
```

```r
# Plot cluster results
p1 <- fviz_nbclust(Features, FUN = hcut, method = "wss",
                   k.max = 10) +
  ggtitle("(A) Elbow method")
p2 <- fviz_nbclust(Features, FUN = hcut, method = "silhouette",
                   k.max = 10) +
  ggtitle("(B) Silhouette method")
p3 <- fviz_nbclust(Features, FUN = hcut, method = "gap_stat",
                   k.max = 10) +
  ggtitle("(C) Gap statistic")

# Display plots side by side
gridExtra::grid.arrange(p1, p2, p3, nrow = 1)
```

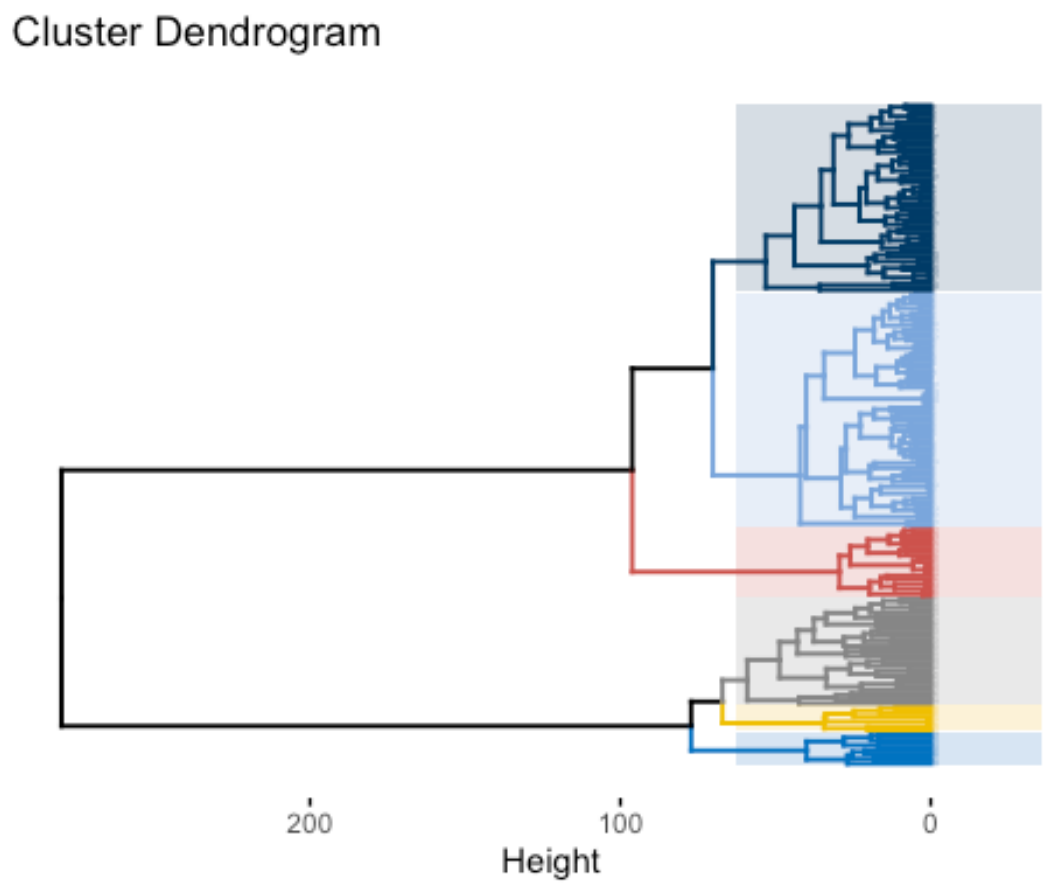



```r
# Ward's method
hc5 <- hclust(d, method = "ward.D2" )

# Cut tree into 6 groups
sub_grp <- cutree(hc5, k = 6)

# Number of members in each cluster
table(sub_grp)
```

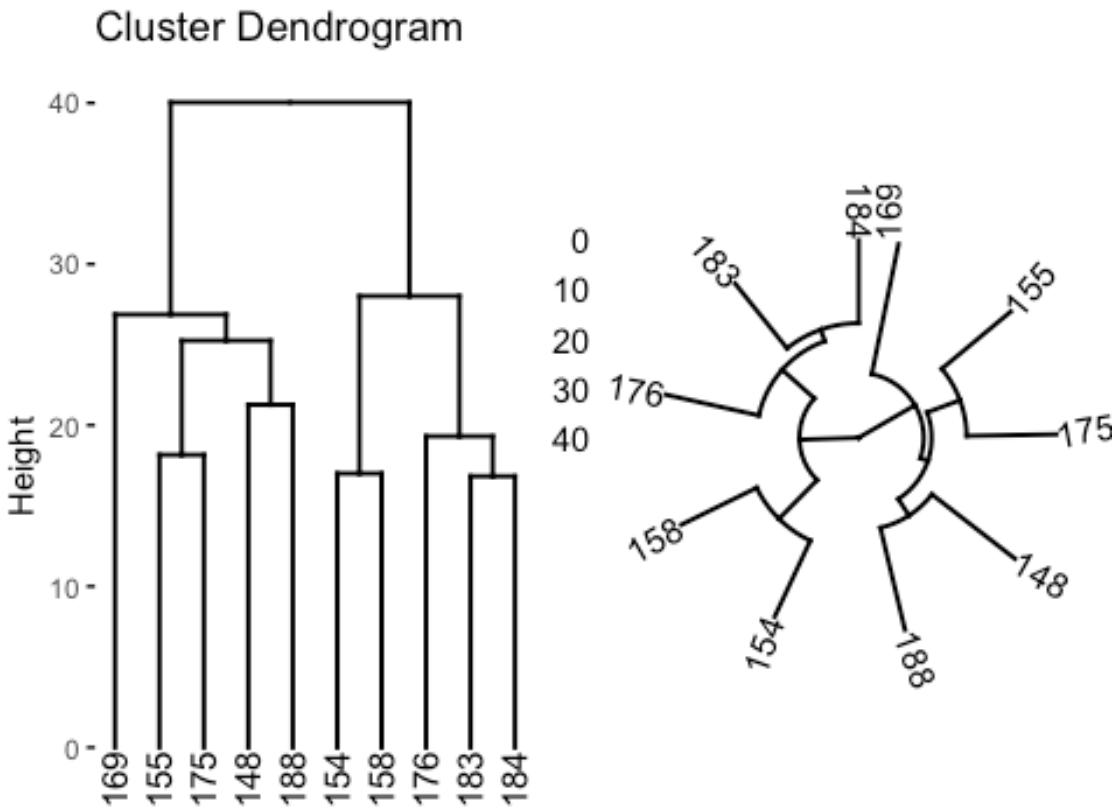
```
## sub_grp
##  1  2  3  4  5  6
## 70 56 21 10 32  8
```

```r
# Plot full dendogram
fviz_dend(
  hc5,
  k = 6,
  horiz = TRUE,
  rect = TRUE,
  rect_fill = TRUE,
  rect_border = "jco",
  k_colors = "jco",
```

```
  cex = 0.1
)

## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use
"none" instead as
## of ggplot2 3.3.4.
## ℹ The deprecated feature was likely used in the factoextra package.
##   Please report the issue at
<]8;;https://github.com/kassambara/factoextra/issueshttps://github.com/kassam
bara/factoextra/issues]8;;>.
```

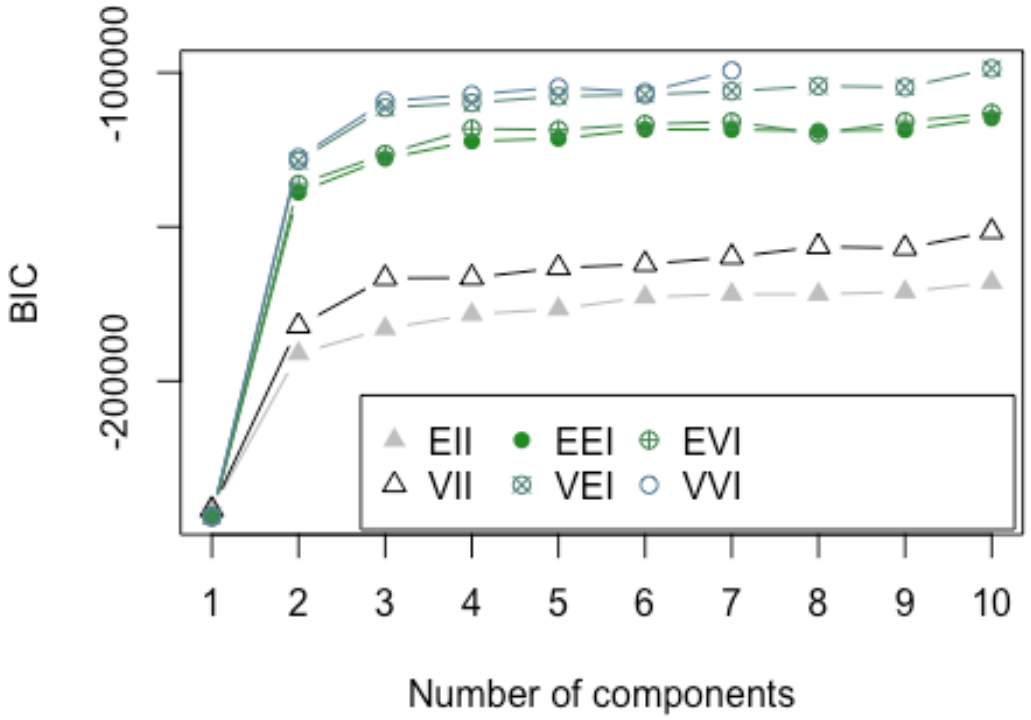



```
dend_plot <- fviz_dend(hc5)                 # create full dendogram
dend_data <- attr(dend_plot, "dendrogram") # extract plot info
dend_cuts <- cut(dend_data, h = 70.5)      # cut the dendogram at
# designated height
# Create sub dendrogram plots
p1 <- fviz_dend(dend_cuts$lower[[1]])
p2 <- fviz_dend(dend_cuts$lower[[1]], type = 'circular')

# Side by side plots
gridExtra::grid.arrange(p1, p2, nrow = 1)
```

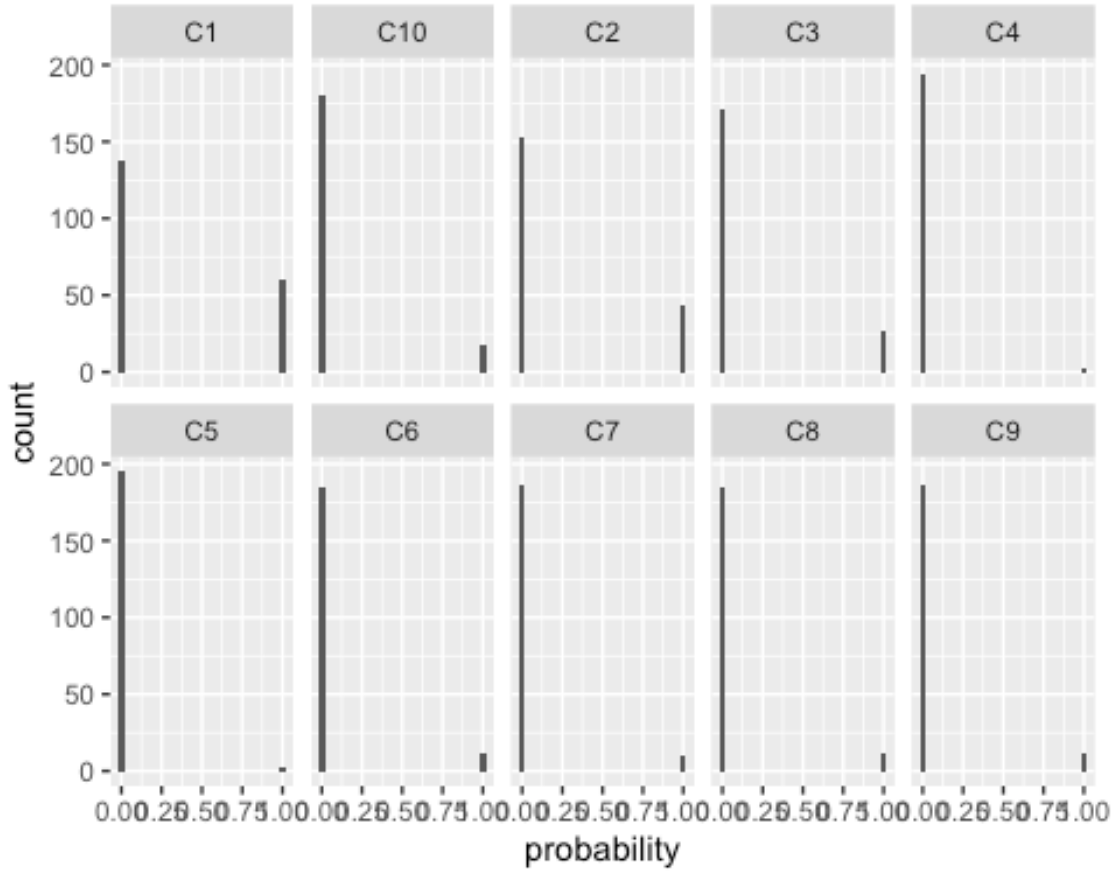


Model Based

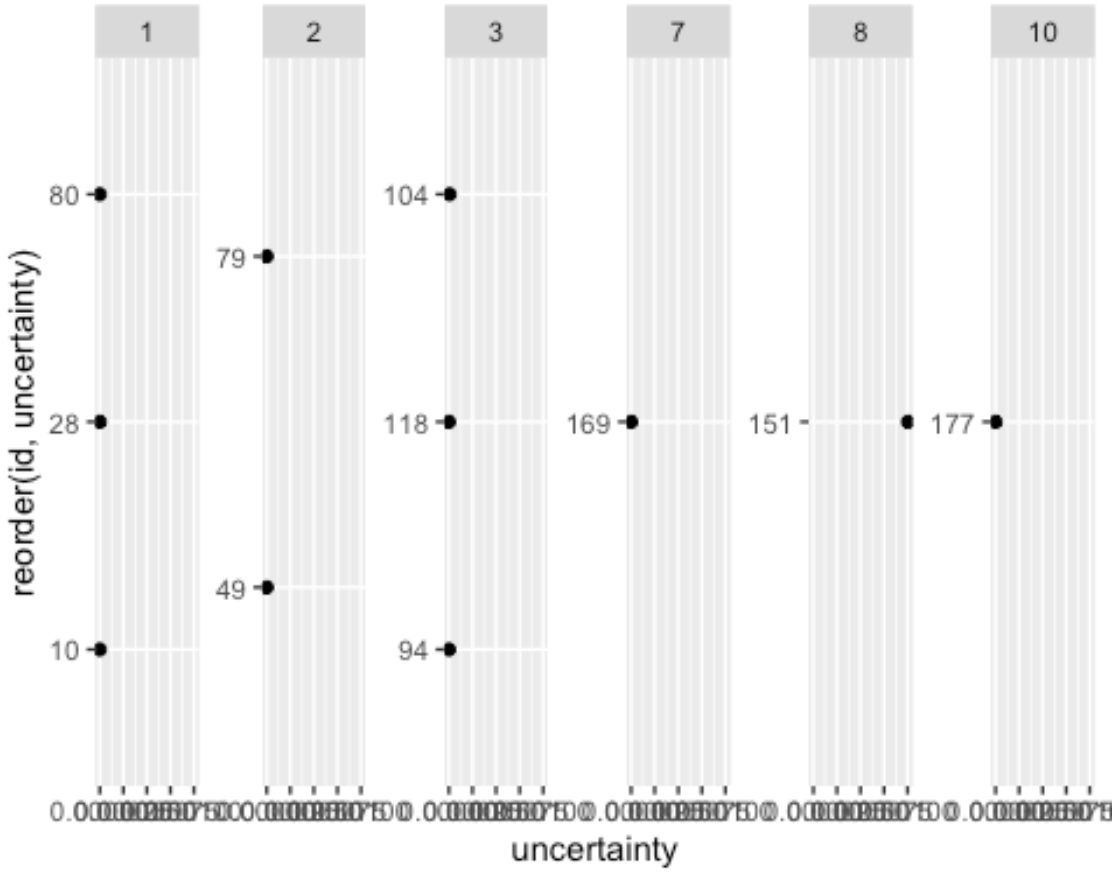
```
F_mc <- Mclust(Features, 1:10)
sort(F_mc$uncertainty, decreasing = TRUE) %>% head()

## [1] 1.005054e-03 6.809868e-06 5.766987e-11 7.110534e-12 3.083311e-12
## [6] 2.578382e-12

plot(F_mc, what = 'BIC',
     legendArgs = list(x = "bottomright", ncol = 5))
```



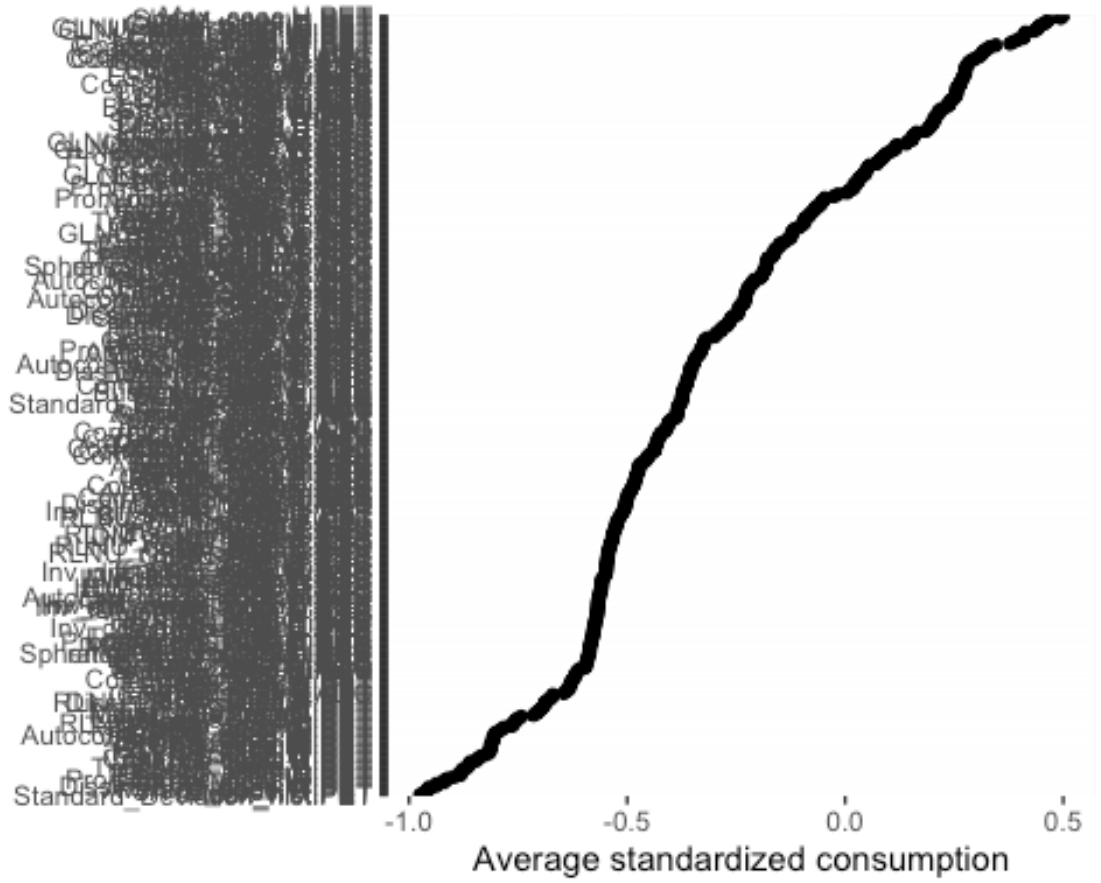

```
probabilities <- F_mc$z
colnames(probabilities) <- paste0('C', 1:10)

probabilities <- probabilities %>%
  as.data.frame() %>%
  mutate(id = row_number()) %>%
  tidyr::gather(cluster, probability, -id)

ggplot(probabilities, aes(probability)) +
  geom_histogram() +
  facet_wrap(~ cluster, nrow = 2)

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




```
uncertainty <- data.frame( id = 1:nrow(Features), cluster =
                           F_mc$classification,uncertainty =
                           F_mc$uncertainty
)

uncertainty %>%
  group_by(cluster) %>%
  filter(uncertainty > 0.0) %>%
  ggplot(aes(uncertainty, reorder(id, uncertainty))) +
  geom_point() +
  facet_wrap(~ cluster, scales = 'free_y', nrow = 1)
```




```
cluster2 <- Features %>%
  scale() %>%
  as.data.frame() %>%
  mutate(cluster = F_mc$classification) %>%
  filter(cluster == 2) %>%
  select(-cluster)

cluster2 %>%
  tidyr::gather(product, std_count) %>%
  group_by(product) %>%
  summarize(avg = mean(std_count)) %>%
  ggplot(aes(avg, reorder(product, avg))) +
  geom_point() +
  labs(x = "Average standardized consumption", y = NULL)
```


-1.0
-0.5
0.0
0.5
Average standardized consumption