8106hw5

Ze Li

```r
library(rsample)
library(ISLR)
library(tidyverse)
library(caret)
library(kernlab)
library(e1071)
library(ggplot2)
library(RColorBrewer)
library(factoextra)
```

Problem 1

```r
auto = read.csv("/Users/zeze/Library/Mobile Documents/com~apple~CloudDocs/2024/24S BIST P8106 DS II/hw5/
auto <- auto |>
  mutate(mpg_cat=as.factor(mpg_cat))
head(auto)
```

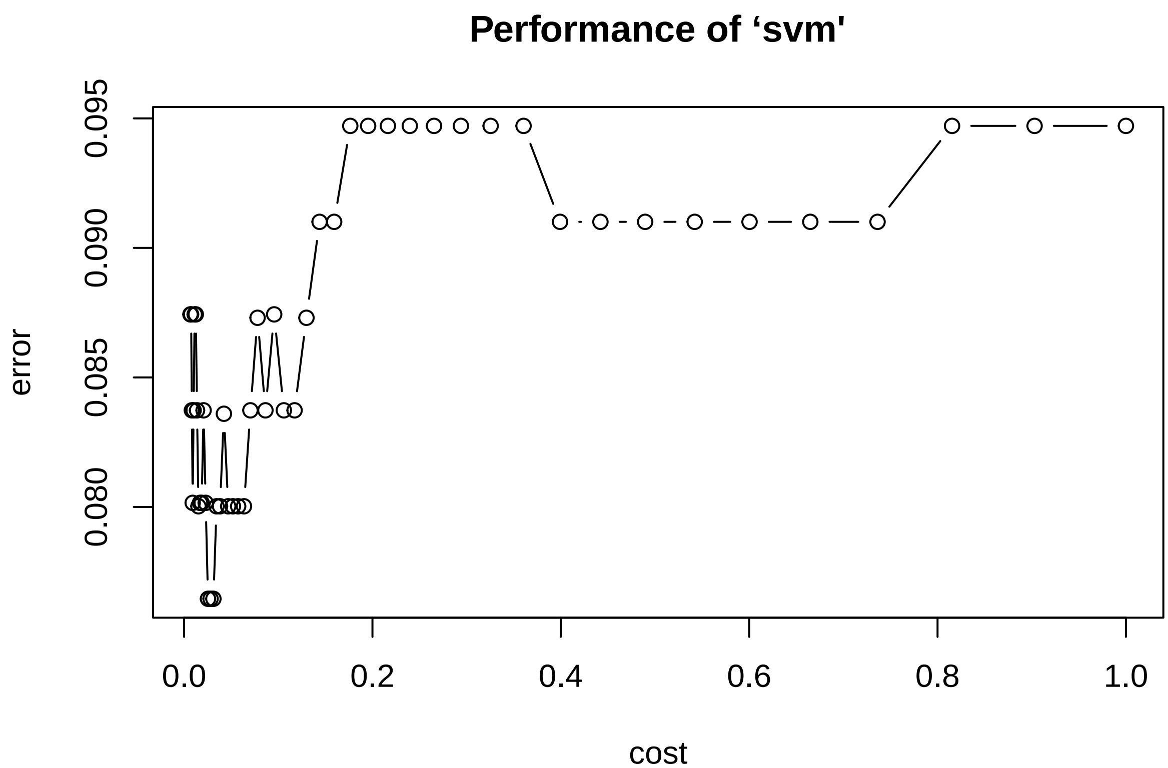
```
##   cylinders displacement horsepower weight acceleration year origin mpg_cat
## 1         8          307        130   3504         12.0   70      1     low
## 2         8          350        165   3693         11.5   70      1     low
## 3         8          318        150   3436         11.0   70      1     low
## 4         8          304        150   3433         12.0   70      1     low
## 5         8          302        140   3449         10.5   70      1     low
## 6         8          429        198   4341         10.0   70      1     low
```

```r
data_split <- initial_split(auto, prop = 0.7)
train <- training(data_split)
test <- testing(data_split)
x_test <- model.matrix(mpg_cat ~ ., test)[, -1]
head(train)
```

```
##   cylinders displacement horsepower weight acceleration year origin mpg_cat
## 1         8          260         90   3420         22.2   79      1    high
## 2         6          232         90   3265         18.2   79      1     low
## 3         4           85         70   1945         16.8   77      3    high
## 4         4           98         90   2265         15.5   73      2    high
## 5         4           89         71   1925         14.0   79      2    high
## 6         4          121        113   2234         12.5   70      2    high
```

(a) Fit a support vector classifier to the training data. What are the training and test error rates?

```r
set.seed(1)
linear.tune <- tune.svm(mpg_cat ~ .,
                        data = train,
                        kernel = "linear",
                        cost = exp(seq(-5, 0, len = 50)),
                        scale = TRUE)
plot(linear.tune)
```




```r
# show the best parameters
linear.tune$best.parameters
```

```
##          cost
## 14 0.02538824
```

```r
best.linear <- linear.tune$best.model
# summary
summary(best.linear)
```

```
##
## Call:
```

```
## best.svm(x = mpg_cat ~ ., data = train, cost = exp(seq(-5, 0, len = 50)),
##     kernel = "linear", scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  0.02538824
##
## Number of Support Vectors:  99
##
##  ( 50 49 )
##
##
## Number of Classes:  2
##
## Levels:
##  high low
```

```r
set.seed(1)
# Training error rate
confusionMatrix(data = best.linear$fitted, reference = train$mpg_cat)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction high low
##       high  139  15
##       low     6 114
##
##                Accuracy : 0.9234
##                  95% CI : (0.8852, 0.9519)
##     No Information Rate : 0.5292
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.8456
##
##  Mcnemar's Test P-Value : 0.08086
##
##             Sensitivity : 0.9586
##             Specificity : 0.8837
##          Pos Pred Value : 0.9026
##          Neg Pred Value : 0.9500
##              Prevalence : 0.5292
##          Detection Rate : 0.5073
##    Detection Prevalence : 0.5620
##       Balanced Accuracy : 0.9212
##
##        'Positive' Class : high
##
```

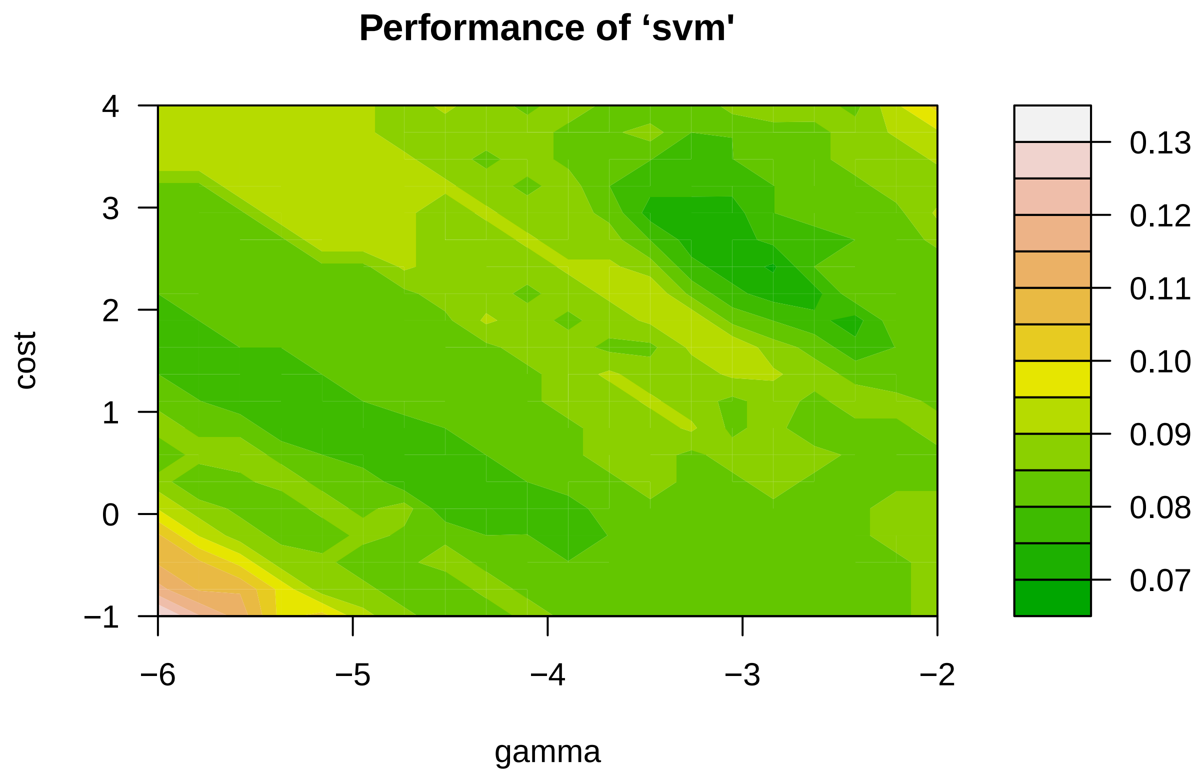
```r
# Test error rate
pred.linear <- predict(best.linear, newdata = test)
confusionMatrix(data = pred.linear, reference = test$mpg_cat)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction high low
##       high   49  11
##       low     2  56
##
##                Accuracy : 0.8898
##                  95% CI : (0.819, 0.94)
##     No Information Rate : 0.5678
##     P-Value [Acc > NIR] : 2.353e-14
##
##                   Kappa : 0.7802
##
##  Mcnemar's Test P-Value : 0.0265
##
##             Sensitivity : 0.9608
##             Specificity : 0.8358
##          Pos Pred Value : 0.8167
##          Neg Pred Value : 0.9655
##              Prevalence : 0.4322
##          Detection Rate : 0.4153
##    Detection Prevalence : 0.5085
##       Balanced Accuracy : 0.8983
##
##        'Positive' Class : high
##
```

The support vector classifier's train accuracy is 0.9197 so error rate is 0.0803%, and test accuracy is 0.9153 so error rate is 0.0847%.

(b) Fit a support vector machine with a radial kernel to the training data. What are the training and test error rates?

```
set.seed(1)
radial.tune <- tune.svm(mpg_cat ~ .,
                        data = train,
                        kernel = "radial",
                        cost = exp(seq(-1,4,len = 20)),
                        gamma = exp(seq(-6,-2,len = 20)))
plot(radial.tune, transform.y = log, transform.x = log,
     color.palette = terrain.colors)
```




```r
best.radial <- radial.tune$best.model
summary(best.radial)
```

```
##
## Call:
## best.svm(x = mpg_cat ~ ., data = train, gamma = exp(seq(-6, -2, len = 20)),
##     cost = exp(seq(-1, 4, len = 20)), kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  11.2577
##
## Number of Support Vectors:  59
##
##  ( 29 30 )
##
##
## Number of Classes:  2
##
## Levels:
##  high low
```

```r
# Training error rate
confusionMatrix(data = best.radial$fitted, reference = train$mpg_cat)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction high low
##       high  142  11
##       low     3 118
##
##                Accuracy : 0.9489
##                  95% CI : (0.9158, 0.9718)
##     No Information Rate : 0.5292
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.8971
##
##  Mcnemar's Test P-Value : 0.06137
##
##             Sensitivity : 0.9793
##             Specificity : 0.9147
##          Pos Pred Value : 0.9281
##          Neg Pred Value : 0.9752
##              Prevalence : 0.5292
##          Detection Rate : 0.5182
##    Detection Prevalence : 0.5584
##       Balanced Accuracy : 0.9470
##
##        'Positive' Class : high
##
```

```r
# Test error rate
pred.radial <- predict(best.radial, newdata = test)
confusionMatrix(data = pred.radial, reference = test$mpg_cat)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction high low
##       high   50  10
##       low     1  57
##
##                Accuracy : 0.9068
##                  95% CI : (0.8393, 0.9525)
##     No Information Rate : 0.5678
##     P-Value [Acc > NIR] : 5.413e-16
##
##                   Kappa : 0.814
##
##  Mcnemar's Test P-Value : 0.01586
##
##             Sensitivity : 0.9804
##             Specificity : 0.8507
```

```
##          Pos Pred Value : 0.8333
##          Neg Pred Value : 0.9828
##              Prevalence : 0.4322
##          Detection Rate : 0.4237
##    Detection Prevalence : 0.5085
##       Balanced Accuracy : 0.9156
##
##        'Positive' Class : high
##
```

The support vector machine with a radial kernel's train accuracy is 0.9635 so error rate is 0.0365% and test accuracy is 0.9068 error rate is 0.0932%.

Problem 2

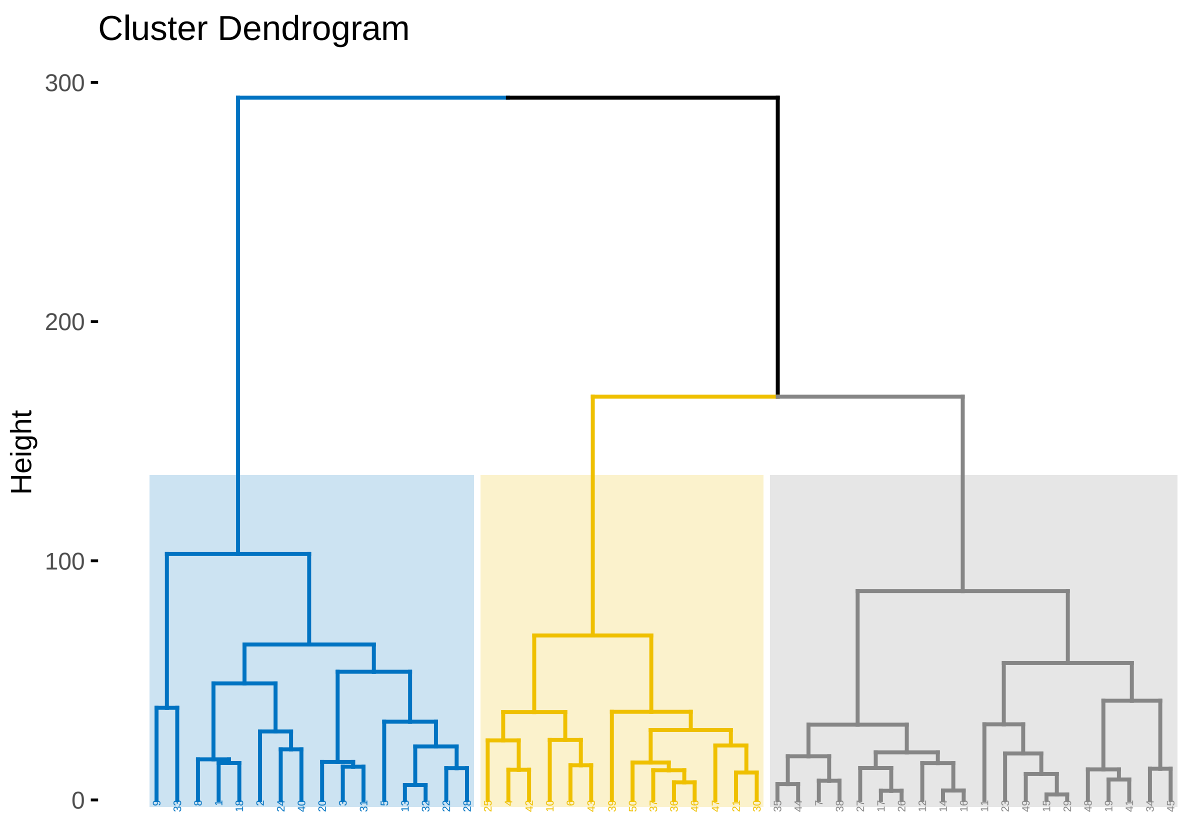
```r
data("USArrests")
USArrests = USArrests %>%
  as_tibble()
head(USArrests)
```

```
## # A tibble: 6 x 4
##   Murder Assault UrbanPop  Rape
##    <dbl>   <int>    <int> <dbl>
## 1   13.2     236       58  21.2
## 2   10       263       48  44.5
## 3    8.1     294       80  31
## 4    8.8     190       50  19.5
## 5    9       276       91  40.6
## 6    7.9     204       78  38.7
```

(a) Using hierarchical clustering with complete linkage and Euclidean distance, cluster the states. Cut the dendrogram at a height that results in three distinct clusters. Which states belong to which clusters?

```r
hc.complete <- hclust(dist(USArrests), method = "complete")
fviz_dend(hc.complete, k = 3,
          cex = 0.3,
          palette = "jco",
          color_labels_by_k = TRUE,
          rect = TRUE, rect_fill = TRUE, rect_border = "jco",
          labels_track_height = 2.5)
```

```
## 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 <https://github.com/kassambara/factoextra/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```r
ind3.complete <- cutree(hc.complete, 3)

# The states in different clusters
cl1 <- USArrests[ind3.complete == 1,]
cl1
```

```
## # A tibble: 16 x 4
##    Murder Assault UrbanPop  Rape
##     <dbl>   <int>    <int> <dbl>
##  1   13.2     236       58  21.2
##  2   10       263       48  44.5
##  3    8.1     294       80  31
##  4    9       276       91  40.6
##  5    5.9     238       72  15.8
##  6   15.4     335       80  31.9
##  7   10.4     249       83  24
##  8   15.4     249       66  22.2
##  9   11.3     300       67  27.8
## 10   12.1     255       74  35.1
## 11   16.1     259       44  17.1
## 12   12.2     252       81  46
## 13   11.4     285       70  32.1
## 14   11.1     254       86  26.1
## 15   13       337       45  16.1
## 16   14.4     279       48  22.5
```

```r
cl2 <- USArrests[ind3.complete == 2,]
cl2
```

```
## # A tibble: 14 x 4
##    Murder Assault UrbanPop  Rape
##     <dbl>   <int>    <int> <dbl>
##  1    8.8     190       50  19.5
##  2    7.9     204       78  38.7
##  3   17.4     211       60  25.8
##  4    4.4     149       85  16.3
##  5    9       178       70  28.2
##  6    7.4     159       89  18.8
##  7    6.6     151       68  20  
##  8    4.9     159       67  29.3
##  9    3.4     174       87   8.3
## 10   13.2     188       59  26.9
## 11   12.7     201       80  25.5
## 12    8.5     156       63  20.7
## 13    4       145       73  26.2
## 14    6.8     161       60  15.6
```

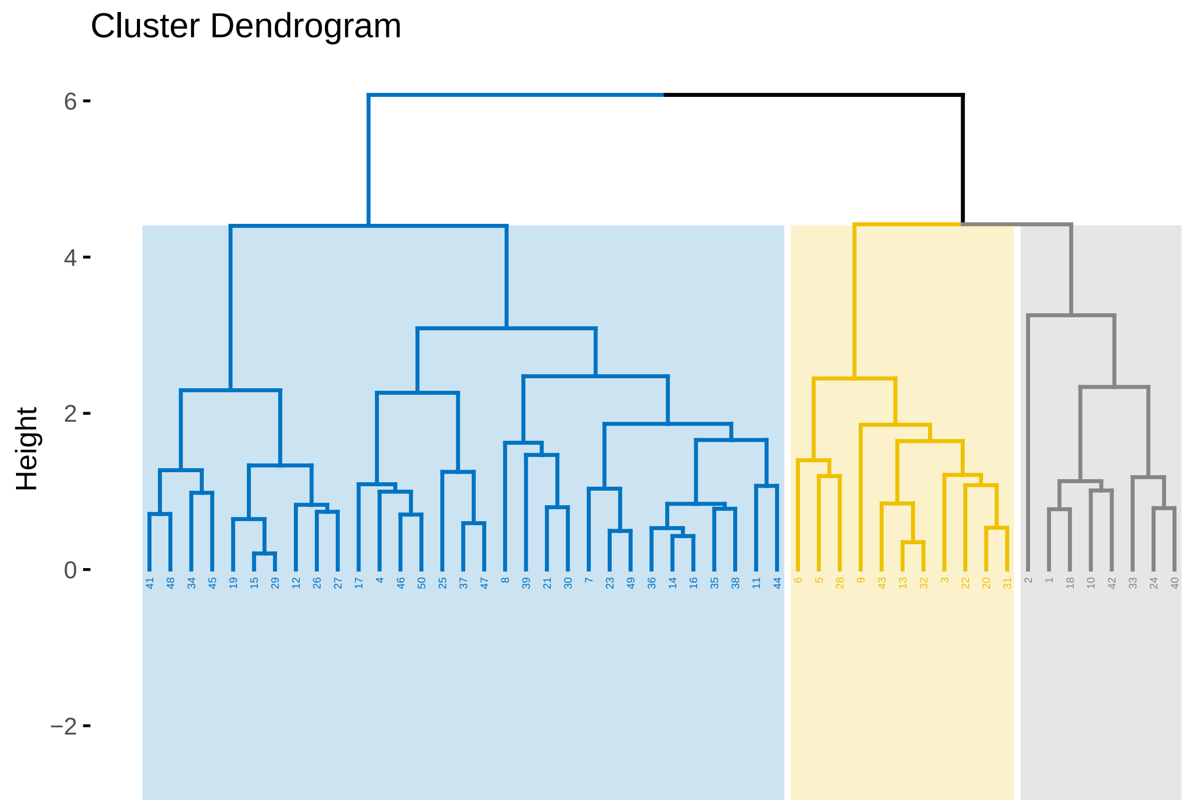
```r
cl3 <- USArrests[ind3.complete == 3,]
cl3
```

```
## # A tibble: 20 x 4
##    Murder Assault UrbanPop  Rape
##     <dbl>   <int>    <int> <dbl>
##  1    3.3     110       77  11.1
##  2    5.3      46       83  20.2
##  3    2.6     120       54  14.2
##  4    7.2     113       65  21  
##  5    2.2      56       57  11.3
##  6    6       115       66  18  
##  7    9.7     109       52  16.3
##  8    2.1      83       51   7.8
##  9    2.7      72       66  14.9
## 10    6       109       53  16.4
## 11    4.3     102       62  16.5
## 12    2.1      57       56   9.5
## 13    0.8      45       44   7.3
## 14    7.3     120       75  21.4
## 15    6.3     106       72  14.9
## 16    3.8      86       45  12.8
## 17    3.2     120       80  22.9
## 18    2.2      48       32  11.2
## 19    5.7      81       39   9.3
## 20    2.6      53       66  10.8
```

(b) Hierarchically cluster the states using complete linkage and Euclidean distance, after scaling the variables to have standard deviation one.

```r
scale.usa <- scale(USArrests)

hc.complete.scaled <- hclust(dist(scale.usa), method = "complete")
fviz_dend(hc.complete.scaled, k = 3,
          cex = 0.3,
          palette = "jco",
          color_labels_by_k = TRUE,
          rect = TRUE, rect_fill = TRUE, rect_border = "jco",
          labels_track_height = 2.5)
```




```r
ind3.complete.scaled <- cutree(hc.complete.scaled, 3)

# The states in different clusters for standardized data
scaled.cl1 <- USArrests[ind3.complete.scaled == 1,]
scaled.cl1
```

```
## # A tibble: 8 x 4
##   Murder Assault UrbanPop  Rape
##    <dbl>   <int>    <int> <dbl>
## 1   13.2     236       58  21.2
## 2   10       263       48  44.5
## 3   17.4     211       60  25.8
## 4   15.4     249       66  22.2
## 5   16.1     259       44  17.1
```

```
## 6   13       337       45  16.1
## 7   14.4     279       48  22.5
## 8   13.2     188       59  26.9
```

```
scaled.cl2 <- USArrests[ind3.complete.scaled == 2,]
scaled.cl2
```

```
## # A tibble: 11 x 4
##    Murder Assault UrbanPop  Rape
##     <dbl>   <int>    <int> <dbl>
##  1    8.1     294       80  31
##  2    9       276       91  40.6
##  3    7.9     204       78  38.7
##  4   15.4     335       80  31.9
##  5   10.4     249       83  24
##  6   11.3     300       67  27.8
##  7   12.1     255       74  35.1
##  8   12.2     252       81  46
##  9   11.4     285       70  32.1
## 10   11.1     254       86  26.1
## 11   12.7     201       80  25.5
```

```
scaled.cl3 <- USArrests[ind3.complete.scaled == 3,]
scaled.cl3
```

```
## # A tibble: 31 x 4
##    Murder Assault UrbanPop  Rape
##     <dbl>   <int>    <int> <dbl>
##  1    8.8     190       50  19.5
##  2    3.3     110       77  11.1
##  3    5.9     238       72  15.8
##  4    5.3      46       83  20.2
##  5    2.6     120       54  14.2
##  6    7.2     113       65  21
##  7    2.2      56       57  11.3
##  8    6       115       66  18
##  9    9.7     109       52  16.3
## 10    2.1      83       51   7.8
## # i 21 more rows
```

(c) Does scaling the variables change the clustering results? Why? In your opinion, should the variables be scaled before the inter-observation dissimilarities are computed?

Based on the results, scaling the variables lead to significant changes in the clustering results. Since the algorithm will assign larger weight to the predictors with larger value, the states in the same cluster share more similarities than the first model.

Scaling variables before computing inter-observation dissimilarities in hierarchical clustering ensures that each variable contributes equally, prevents disproportionate influence from variables with larger scales, and maintains distance metric consistency. It enhances clustering performance by producing more reliable and interpretable clusters, free from biases due to variable scale discrepancies.