DSA 8070 R Session 12: Cluster Analysis

Whitney

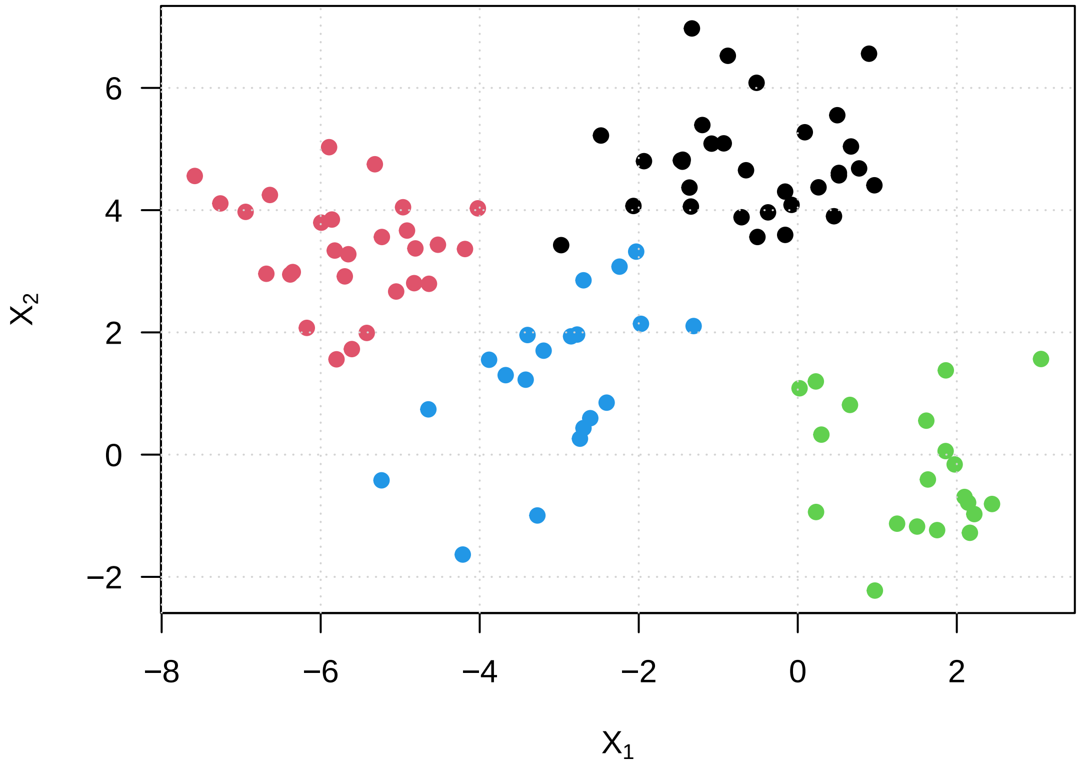
November 13, 2022

Contents

K-Means Clustering

Simulated Example

```r
set.seed(101)
library(scales)
x <- matrix(rnorm(100 * 2), 100, 2)
xmean <- matrix(rnorm(8, sd = 4), 4, 2)
which <- sample(1:4, 100, replace = TRUE)
x = x + xmean[which,]
plot(x, col = which, pch = 19, xlab = expression(X[1]),
     ylab = expression(X[2]), las = 1)
grid()
```

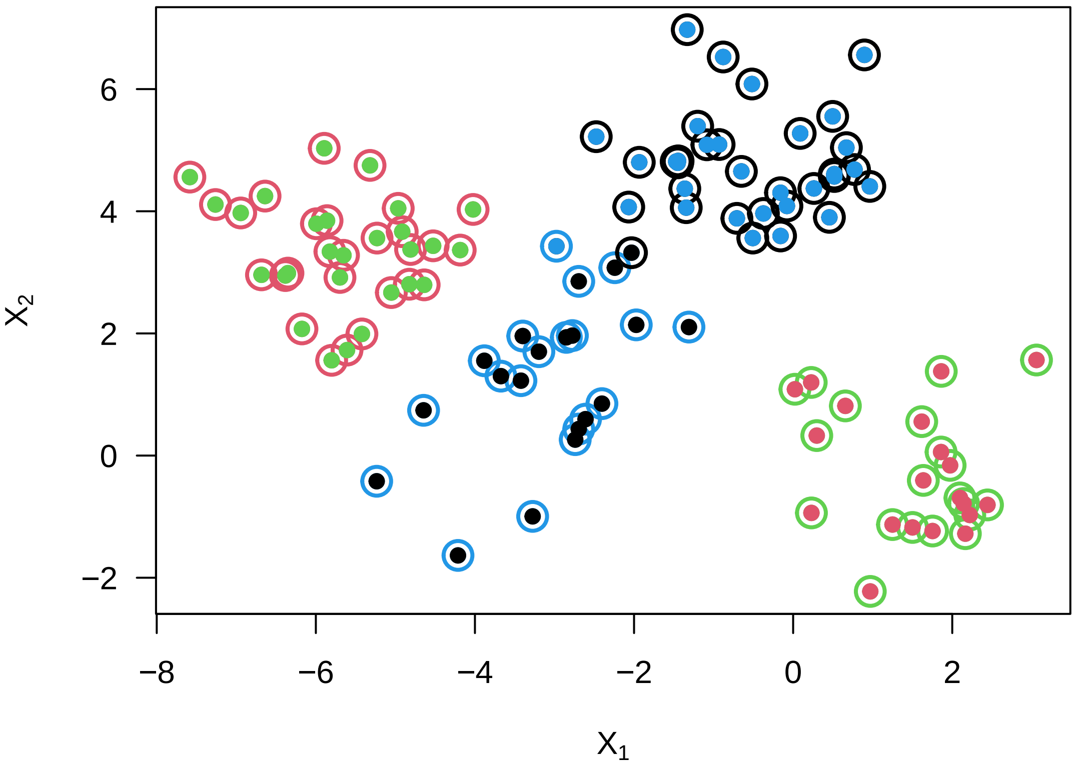


```
km.out <- kmeans(x, 4, nstart = 15)
km.out
```

```
## K-means clustering with 4 clusters of sizes 32, 28, 20, 20
##
## Cluster means:
##         [,1]       [,2]
## 1 -0.5787702  4.7639233
## 2 -5.6518323  3.3513316
## 3  1.4989983 -0.2412154
## 4 -3.1104142  1.2535711
##
## Clustering vector:
##   [1] 2 4 1 2 4 1 2 4 1 1 3 1 1 3 4 3 2 3 2 2 2 2 2 3 1 1 4 2 4 1 2 3 2 4 4 3 3
##  [38] 4 3 3 2 4 4 2 2 3 2 1 2 4 2 1 1 3 3 4 3 1 1 1 4 2 2 2 4 4 1 1 3 2 2 1 1 3
##  [75] 1 3 2 1 1 1 4 1 4 1 2 3 1 2 2 1 1 4 2 4 1 1 3 3 1 1
##
## Within cluster sum of squares by cluster:
## [1] 53.04203 42.40322 34.95921 48.52107
##  (between_SS / total_SS =  85.7 %)
##
## Available components:
##
```

```
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
plot(x, col = km.out$cluster, cex = 2, pch = 1, lwd = 2,
     xlab = expression(X[1]), ylab = expression(X[2]), las = 1)
points(x, col = which, pch = 19)
points(x, col = c(4, 3, 2, 1)[which], pch = 19)
```

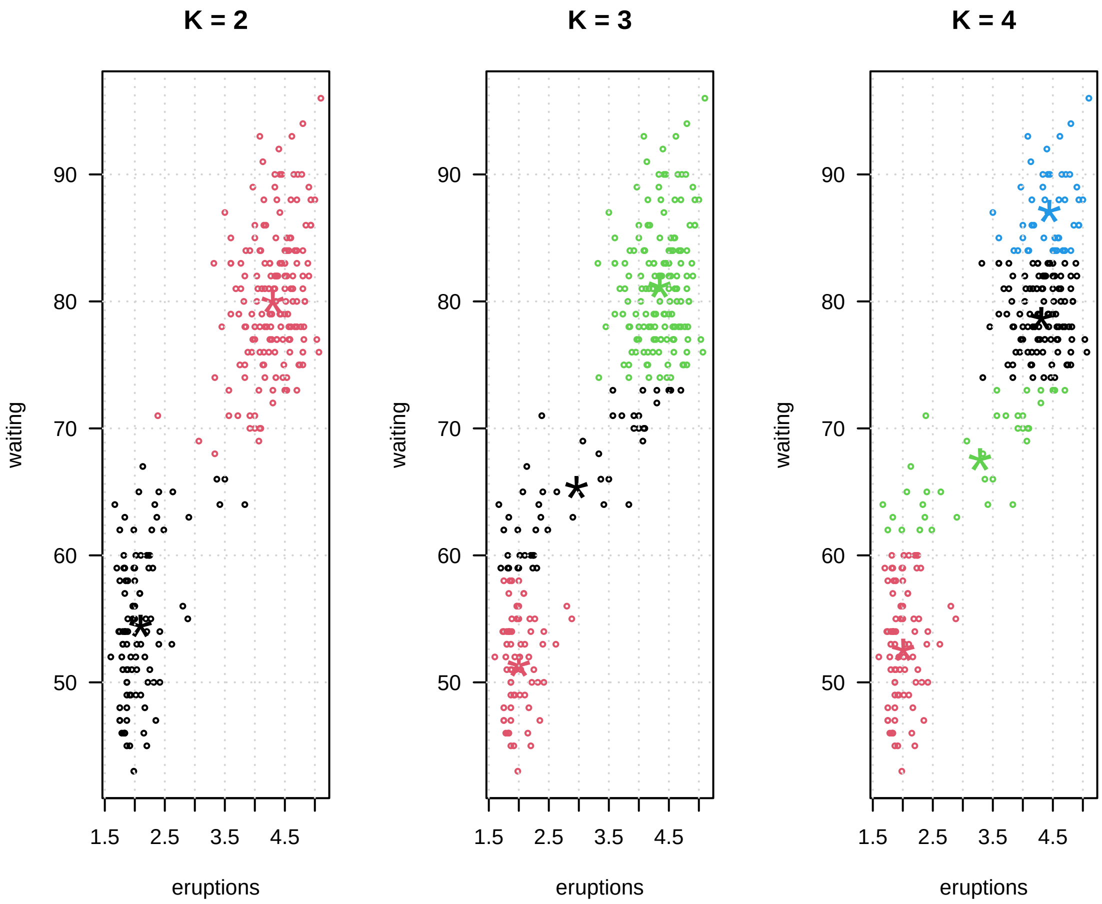


Geyser Example

```
km3.faithful <- kmeans(faithful, 3)
km2.faithful <- kmeans(faithful, 2)
km4.faithful <- kmeans(faithful, 4)

par(las = 1, mfrow = c(1, 3))
plot(faithful, col = km2.faithful$cluster, cex = 0.5, main = "K = 2")
points(km2.faithful$centers, cex = 3, pch = "*", col = 1:2)
grid()
plot(faithful, col = km3.faithful$cluster, cex = 0.5, main = "K = 3")
points(km3.faithful$centers, cex = 3, pch = "*", col = 1:3)
```

```
grid()
plot(faithful, col = km4.faithful$cluster, cex = 0.5, main = "K = 4")
grid()
points(km4.faithful$centers, cex = 3, pch = "*", col = 1:4)
```

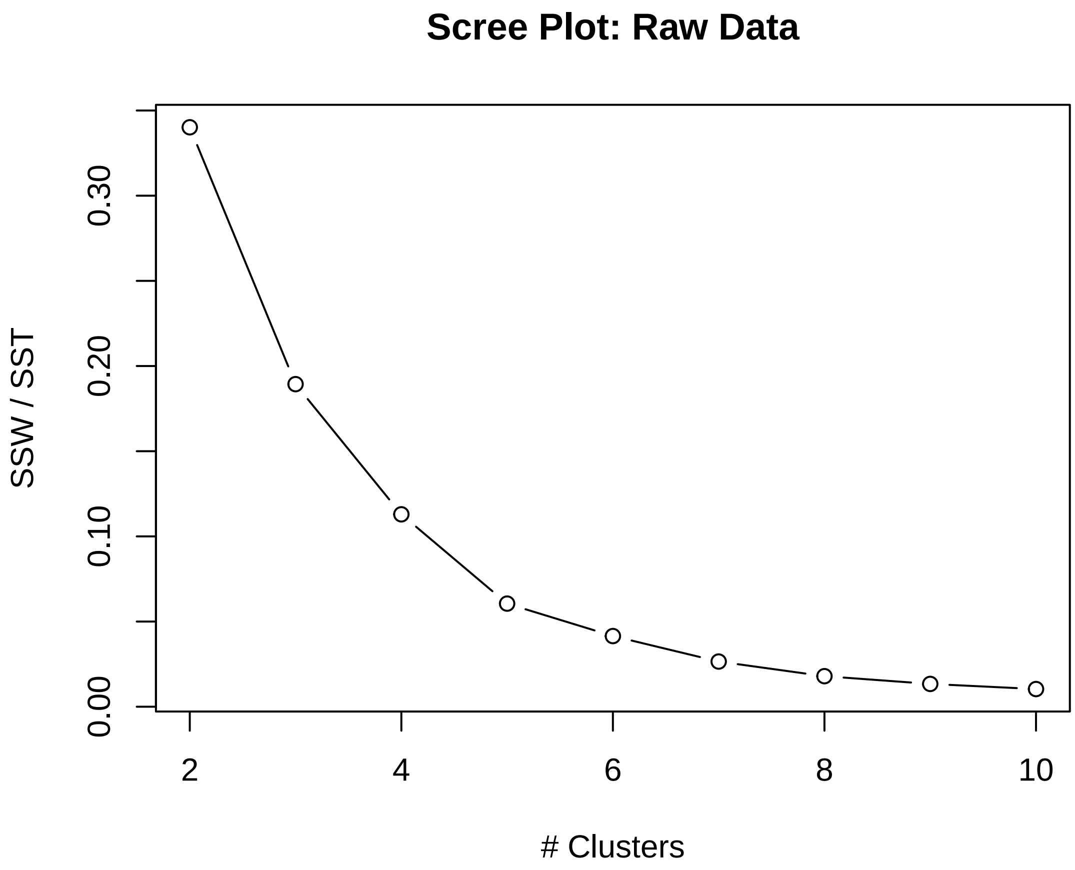


US State Facts and Figures Example

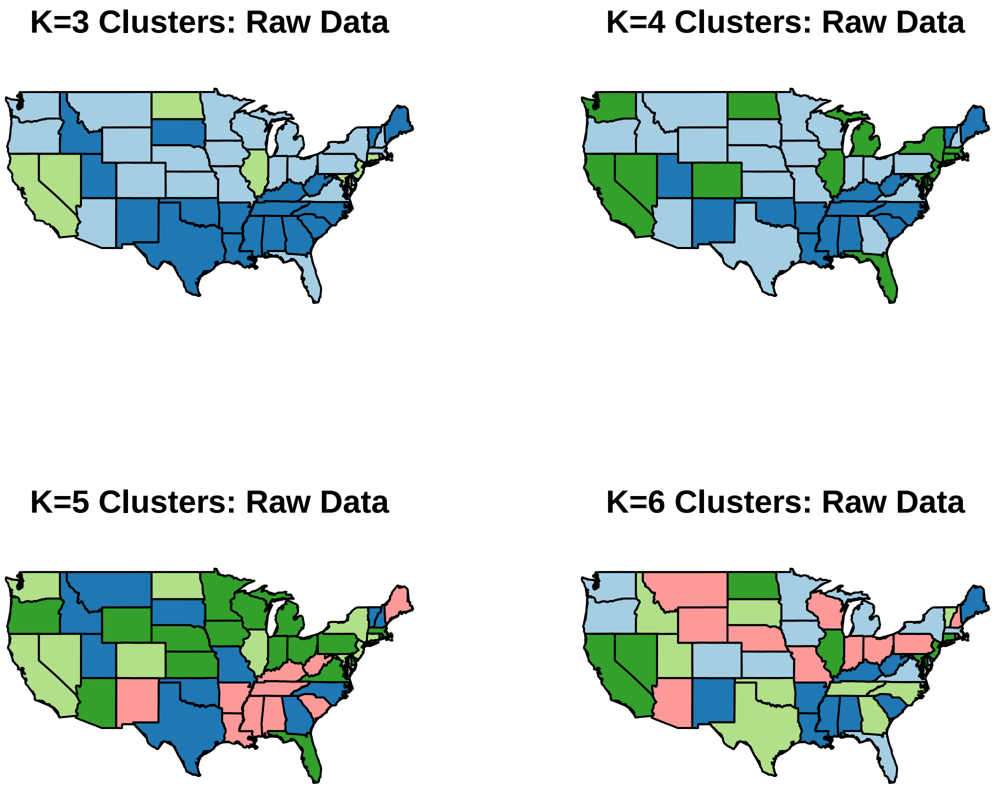
```
# look at states data
vars <- c("Income", "Illiteracy", "Life Exp", "HS Grad")
head(state.x77[, vars])
```

```
##            Income Illiteracy Life Exp HS Grad
## Alabama      3624        2.1    69.05    41.3
## Alaska       6315        1.5    69.31    66.7
## Arizona      4530        1.8    70.55    58.1
## Arkansas     3378        1.9    70.66    39.9
## California   5114        1.1    71.71    62.6
## Colorado     4884        0.7    72.06    63.9
```

```r
# fit k means for k = 2, ..., 10 (raw data)
kmlist <- vector("list", 9)
for(k in 2:10){
  set.seed(1)
  kmlist[[k-1]] <- kmeans(state.x77[, vars], k, nstart = 5000)
}
# scree plot (raw data)
tot.withinss <- sapply(kmlist, function(x) x$tot.withinss)
plot(2:10, tot.withinss / kmlist[[1]]$totss, type = "b", xlab = "# Clusters",
     ylab = "SSW / SST", main = "Scree Plot: Raw Data")
```



```r
# plot results (raw data)
library(maps)
library(RColorBrewer)
par(mfrow = c(2, 2))
for(k in 3:6){
  map(database = "state")
  title(paste0("K=", k, " Clusters: Raw Data"))
  cols <- brewer.pal(k, "Paired")
  for(j in 1:k){
    ix <- names(which(kmlist[[k-1]]$cluster==j))
    if(length(ix) > 1) map(database = "state", regions = ix, col = cols[j],
                           fill = T, add = TRUE)
  }
}
```



K=3 Clusters: Raw Data

K=4 Clusters: Raw Data

K=5 Clusters: Raw Data

K=6 Clusters: Raw Data

Hierarchical Clustering

US State Facts and Figures Example

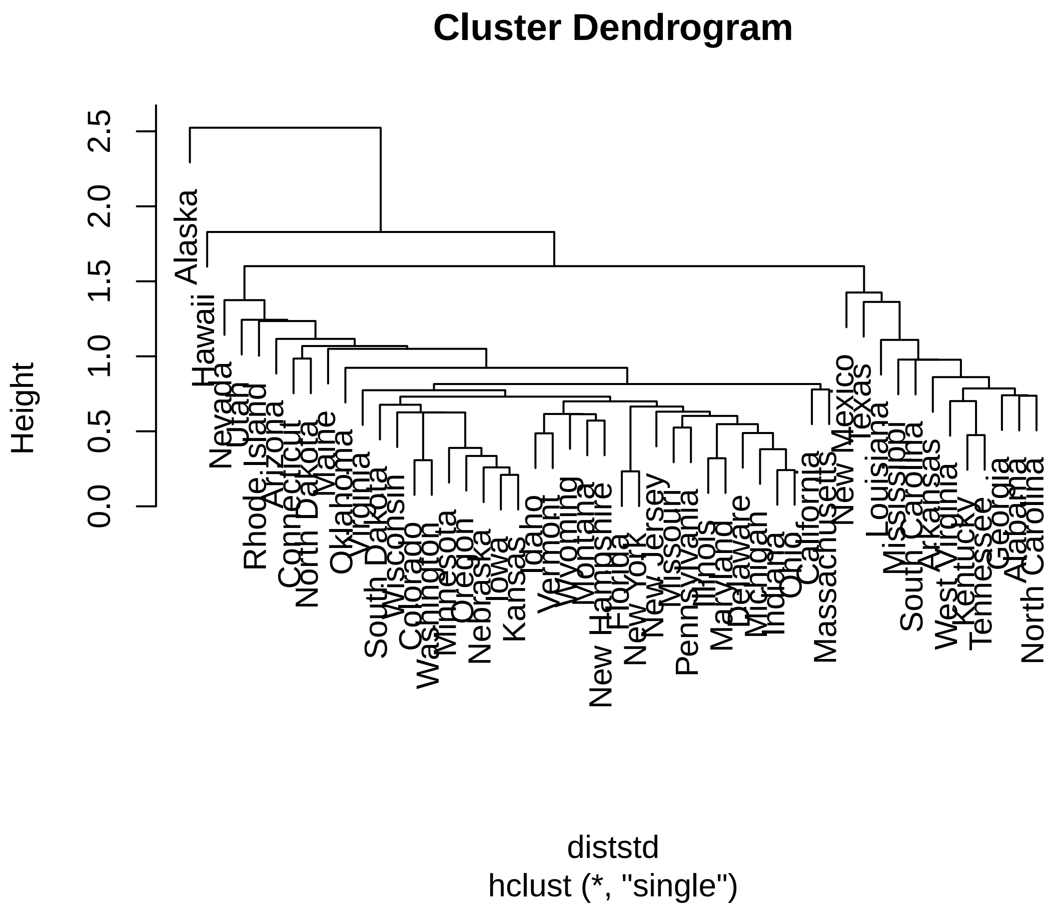
```
apply(state.x77[, vars], 2, mean)
```

```
##     Income Illiteracy   Life Exp    HS Grad
##  4435.8000     1.1700    70.8786    53.1080
```

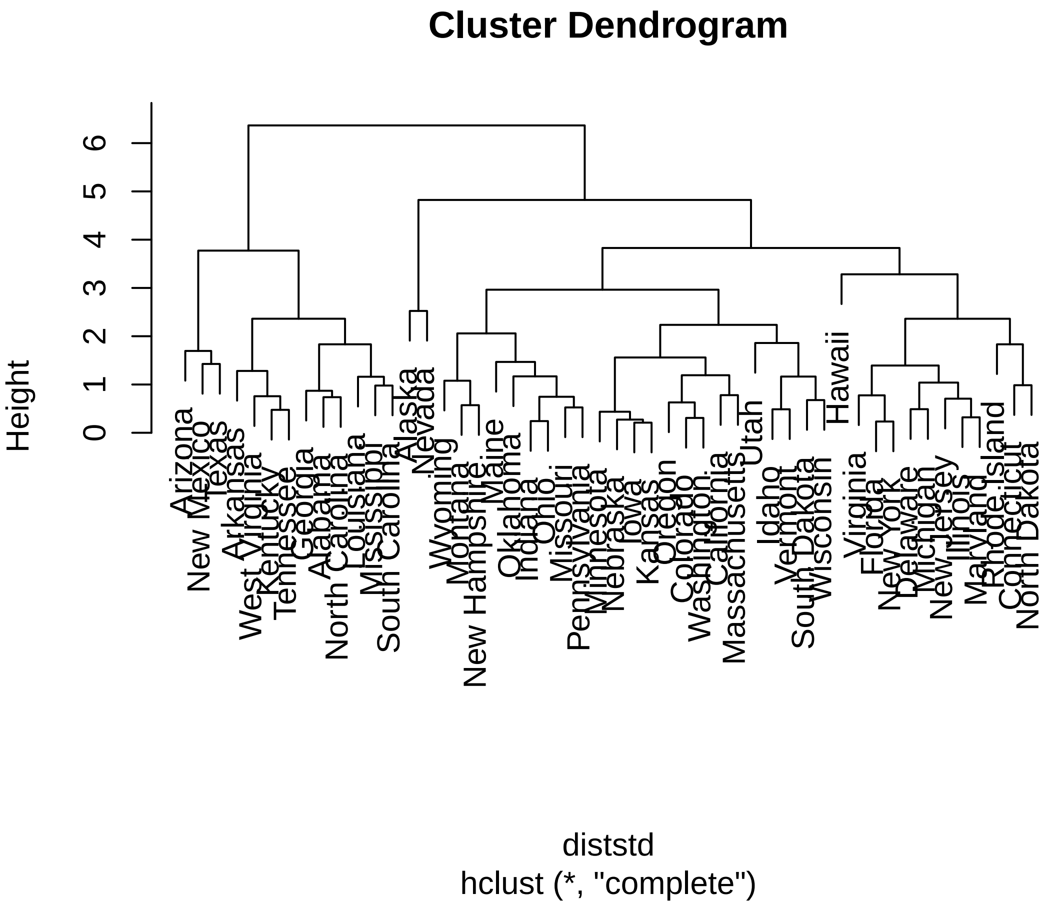
```
apply(state.x77[, vars], 2, sd)
```

```
##      Income  Illiteracy    Life Exp      HS Grad
## 614.4699392   0.6095331   1.3423936    8.0769978
```

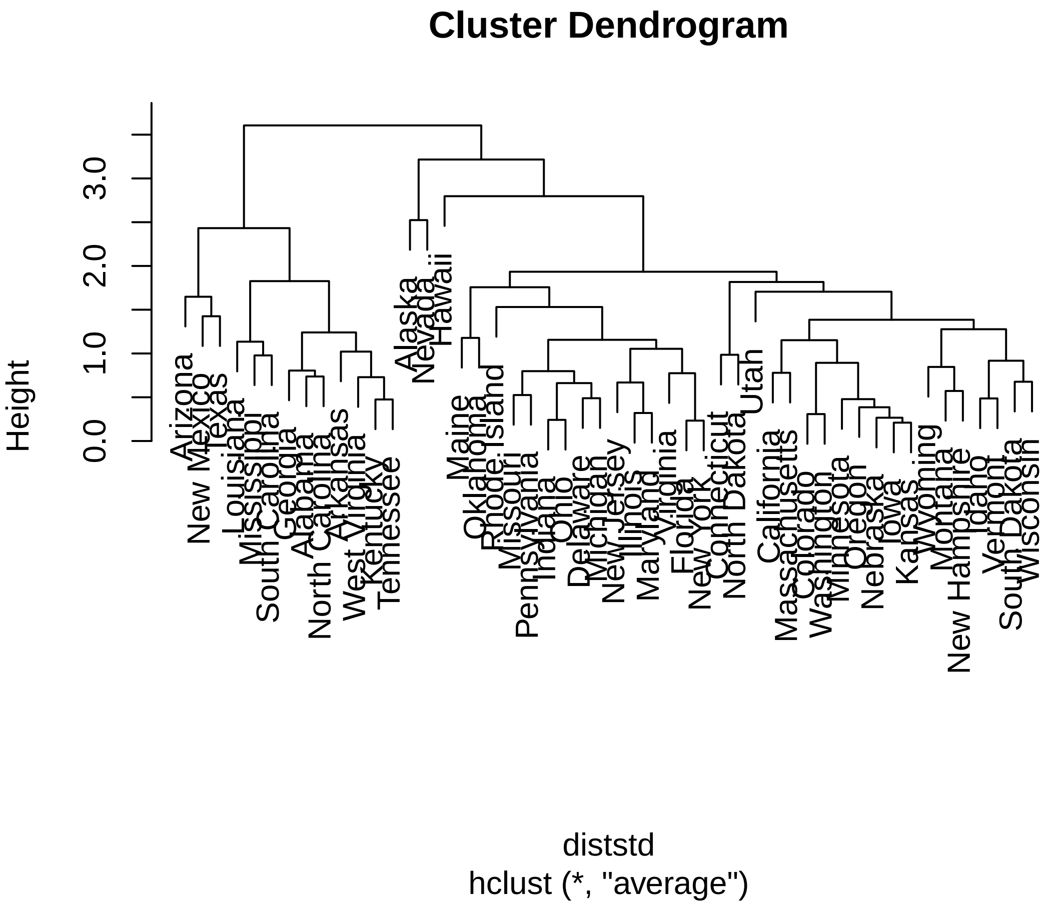
```
# create distance (raw and standardized)
distraw <- dist(state.x77[, vars])
diststd <- dist(scale(state.x77[, vars]))
# hierarchical clustering (standardized data)
hcstdSL <- hclust(diststd, method = "single")
hcstdCL <- hclust(diststd, method = "complete")
hcstdAL <- hclust(diststd, method = "average")
# plot results (standardized data)
plot(hcstdSL)
```



```
plot(hcstdCL)
```



```
plot(hcstdAL)
```

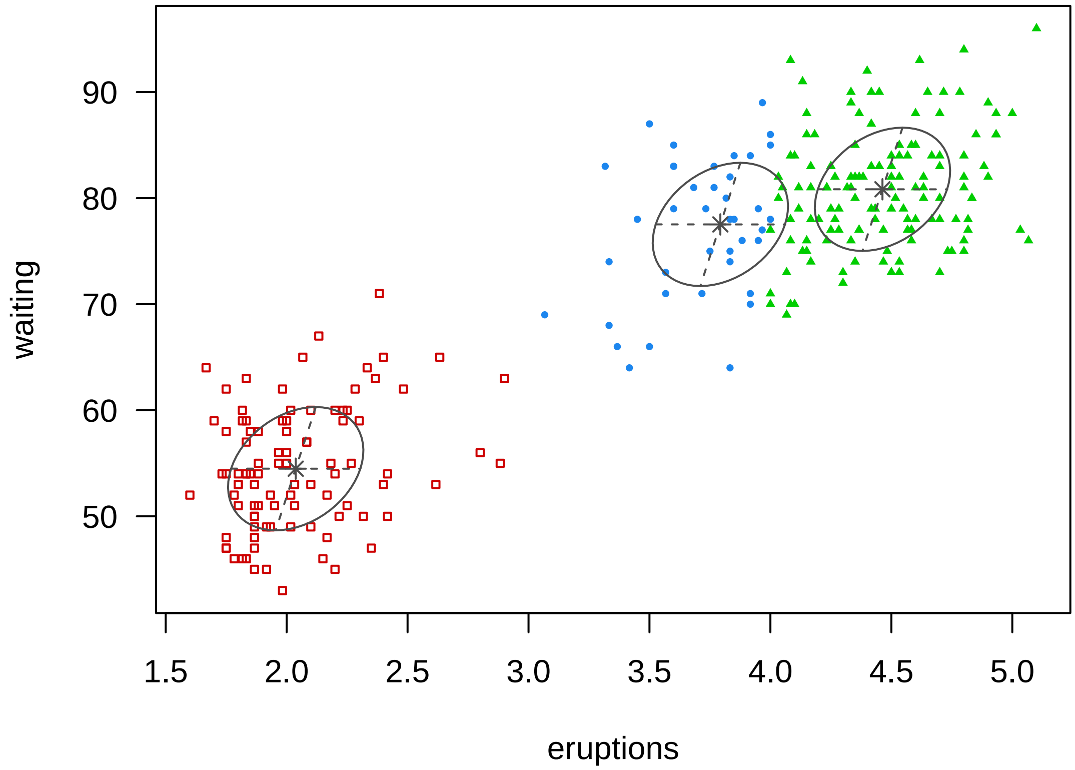


Model-based

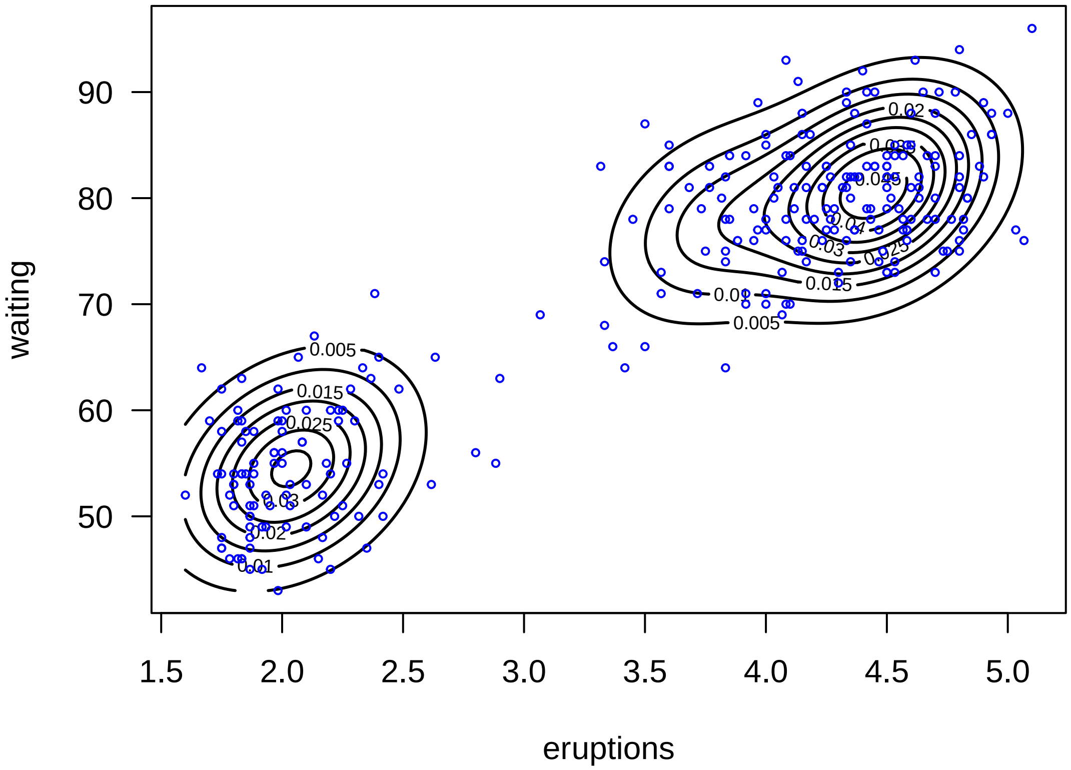
Geyser Example

```
library(mclust)
BIC <- mclustBIC(faithful)
model1 <- Mclust(faithful, x = BIC)

plot(model1, what = "classification", cex = 0.5, las = 1)
```



```
plot(model1, what = "density", col = "black", lwd = 1.5, las = 1)
points(faithful, col = "blue", cex = 0.5)
```



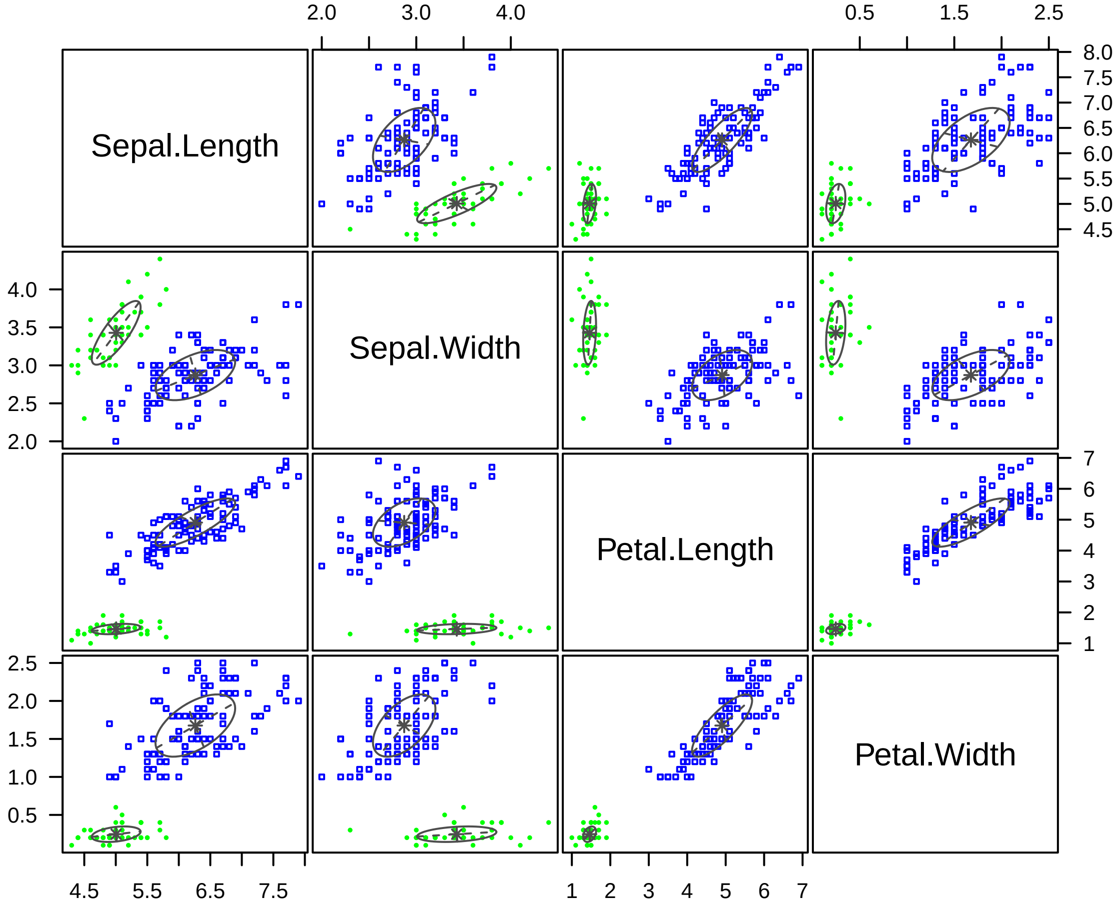
```
(LRT <- mclustBootstrapLRT(faithful, modelName = "VVV"))
```

```
## -------------------------------------------------------------
## Bootstrap sequential LRT for the number of mixture components
## -------------------------------------------------------------
## Model        = VVV
## Replications = 999
##                 LRTS bootstrap p-value
## 1 vs 2   319.065354             0.001
## 2 vs 3     6.130516             0.559
```

Fisher's Iris Data Example

```
data(iris)
attach(iris)
iris$Species <- factor(iris$Species)
dat <- iris[, 1:4]
BIC <- mclustBIC(dat)
model2 <- Mclust(dat, x = BIC)
```

```
par(las = 1)
plot(model2, what = "classification", cex = 0.5, col = c("green", "blue"))
```


Sepal.Length
Sepal.Width
Petal.Length
Petal.Width