Cluster_Analysis

Markus Köfler

2023-05-12

3.1) Distances

(a)

Use the “USArrests” data as discussed in the lecture. Calculate the euclidean distances between states using ‘ `for` -loops’.

Euclidean Distance for multiple dimensions:

$$D_{A,B} = \sqrt{(A1 - B1)^2 + (A2 - B2)^2 + \cdots + (An - Bn)^2}$$

```
data("USArrests")
#show(USArrests)
#n <- nrow(USArrests)
#comb_states<-combn(1:n, 2, simplify = FALSE)
#comb_states
USArrests[1:5, ]
```

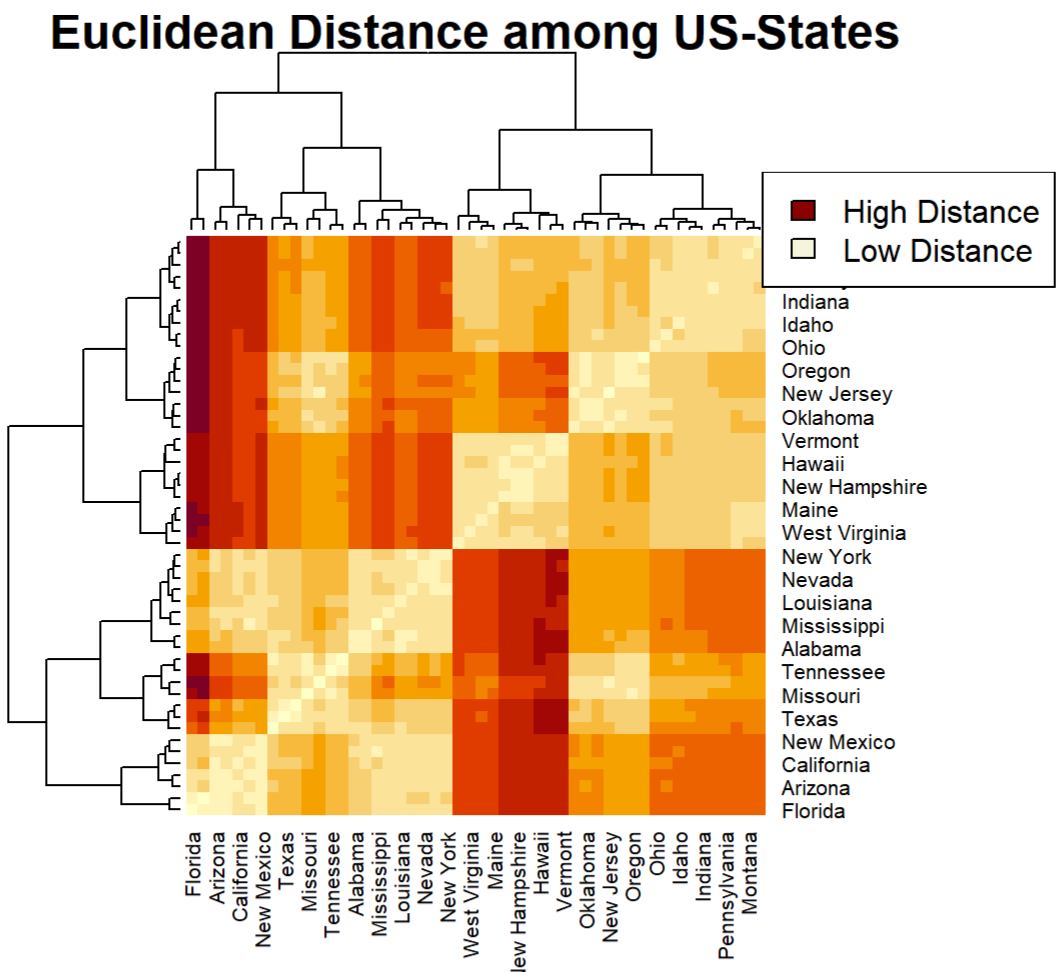
```
##            Murder Assault UrbanPop Rape
## Alabama      13.2     236       58 21.2
## Alaska       10.0     263       48 44.5
## Arizona       8.1     294       80 31.0
## Arkansas      8.8     190       50 19.5
## California    9.0     276       91 40.6
```

```
# creating a csv file
#write.csv(USArrests, file='USArrests.csv')
```

```r
states <- c()
euclids <- c()
for (i in 1:(nrow(USArrests)-1)){
  n <- i+1
  # nested loop
  while (n <= (nrow(USArrests))){
    A <- USArrests[i, ]
    B <- USArrests[n, ]
    euclid <- (A-B)**2 %>% sum() %>% sqrt() %>% as.numeric()
    states <- states %>% append(paste0(rownames(A),' - ', rownames(B)))
    euclids <- euclids %>% append(euclid)
    n <- n+1
  }
}

arrests <- data.frame(states, euclids)

# outputs the 5 state-combinations with the lowest eurclidean distance
arrests[order(arrests$euclids, decreasing = F), ] %>% head(n=5)
```

```
##                    states  euclids
## 609  Iowa - New Hampshire 2.291288
## 673    Kentucky - Montana 3.834058
## 561      Indiana - Kansas 3.929377
## 541   Illinois - New York 6.236986
## 1114          Ohio - Utah 6.637771
```

```r
euclid_mtx <- dist(USArrests, method = "euclidean") %>% as.matrix()
par(mar = c(5, 6, 4, 2) + 0.1, cex.lab = 1.5, cex.axis = 1.2)
heatmap(euclid_mtx,
        main='Euclidean Distance among US-States')
legend("topright",
       legend = c("High Distance", "Low Distance"),
       fill = c("darkred", "beige"),
       bty = "o")
```



(b)

Identify two states with the maximal/minimal distance.

```
arrests[order(arrests$euclids, decreasing = F), ] %>% head(n=1)
```

```
##                   states   euclids
## 609 Iowa - New Hampshire 2.291288
```

```
arrests[order(arrests$euclids, decreasing = F), ] %>% tail(n=1)
```

```
##                     states   euclids
## 389 Florida - North Dakota 293.6228
```

(c)

Calculate the weighted euclidean distances between states (scaled data; the weight of ‘UrbanPop’ should be 0.25; all other weights should be 1).

Weighted Euclidean Distance:

$$D_{A,B} = \sqrt{\alpha_1(A1-B1)^2 + \alpha_2(A2-B2)^2 + \cdots + \alpha_n(An-Bn)}$$

To scale the data (assign weights)

```
# create a vector of weights
weights <- c(1, 1, 0.25, 1)

states <- c()
euclids <- c()

for (i in 1:(nrow(USArrests)-1)){
  n <- i+1
  while (n <= (nrow(USArrests))){
    A <- USArrests[i, ]
    B <- USArrests[n, ]

    # calculate the weighted euclidean distance
    diff <- (A - B) * weights
    euclid <- sqrt(sum(diff^2))

    # append the results to the output vectors
    states <- states %>% append(paste0(rownames(A),' - ', rownames(B)))
    euclids <- euclids %>% append(euclid)

    n <- n+1
  }
}

arrests_weighted <- data.frame(states, euclids)
```

```
# state combination with lowest weighted eucledian distance
arrests_weighted[order(arrests_weighted$euclids, decreasing = F),][1,]
```

```
##                    states  euclids
## 609 Iowa - New Hampshire 2.076656
```

```
# state combination with highest weighted eucledian distance
arrests_weighted[order(arrests_weighted$euclids, decreasing = F),][nrow(arrests_weighted),]
```

```
##                               states  euclids
## 1073 North Carolina - North Dakota 292.3873
```

lowest euclidean distance remain the same: Iowa - New Hampshire

Highest eucledian distance is now different: North Carolina - North Dakota (without weights: Florida - North Dakota

(3.2) Hierarchical clustering

Note: Implementation in Python (Google Colab Notebook also on GitHub)

(a)

Run hierarchical clustering analysis with scaled "USArrests" data. Use the 'complete' method as linkage function. Plot a dendrogram and argue how many clusters you would choose.

Note: Regarding the scale function which defaults to `scale(x, center=TRUE, scale=TRUE)`

- If `center` is `TRUE` then centering is done by subtracting the column means
- If `scale` is `TRUE` then scaling is done by dividing the (centered) columns of `x` by their standard deviations if `center` is `TRUE`, and the root mean square otherwise

This implies that each row is standardized/normalized (like z-score), that is, subtracting the mean of each column's value and dividing the standard deviation. Thereby, the features of the data is preserved, however, it is compressed into a a smaller range of values, making computations much more efficient especially on huge data sets

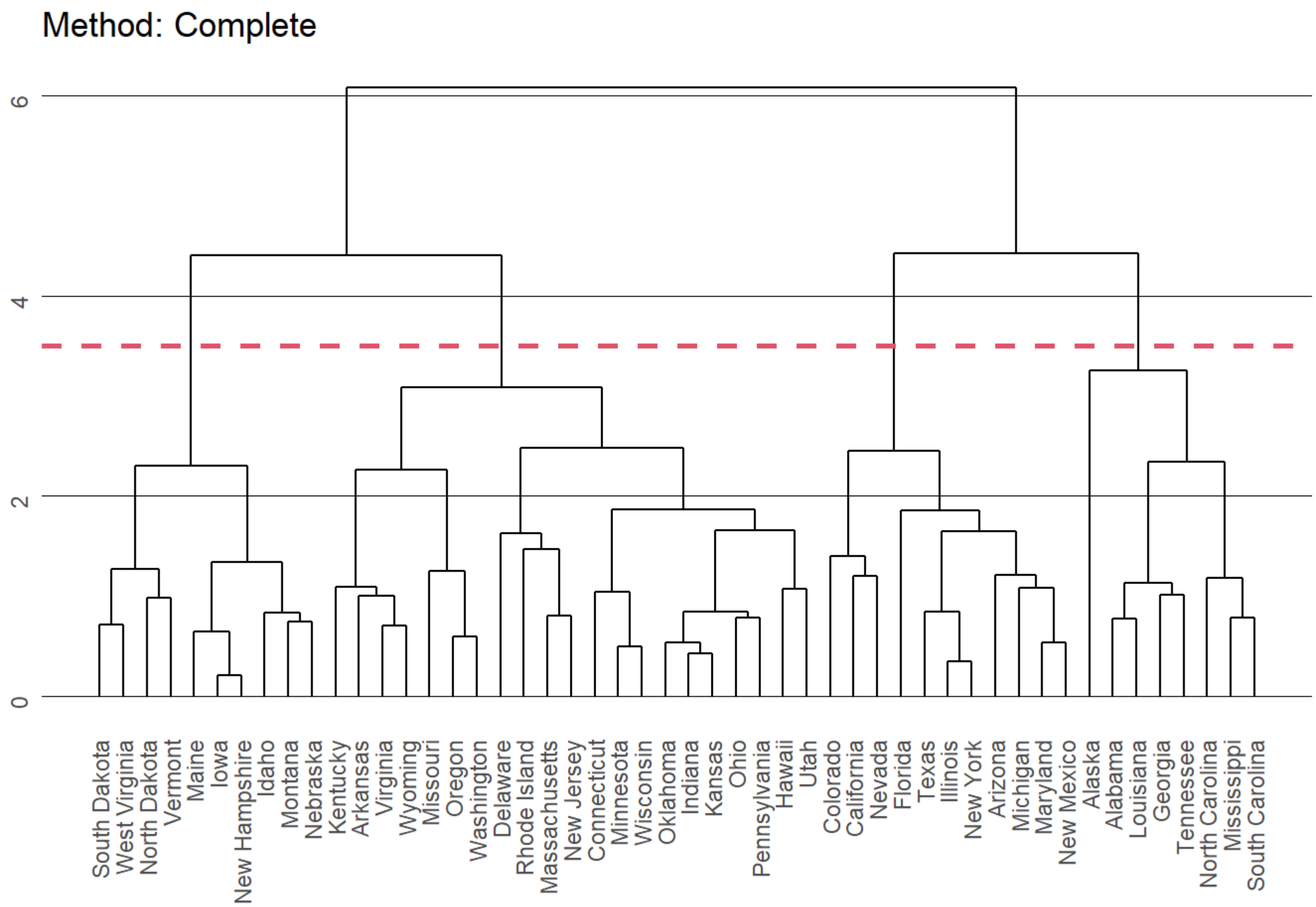
$$\tilde{x} = \frac{x_i - \bar{x}}{\sigma}$$

As a result, the mean of the series $\tilde{x}$ will be 0 and the standard deviation will be 1 (normal distribution $\tilde{x} \sim \mathcal{N}$)

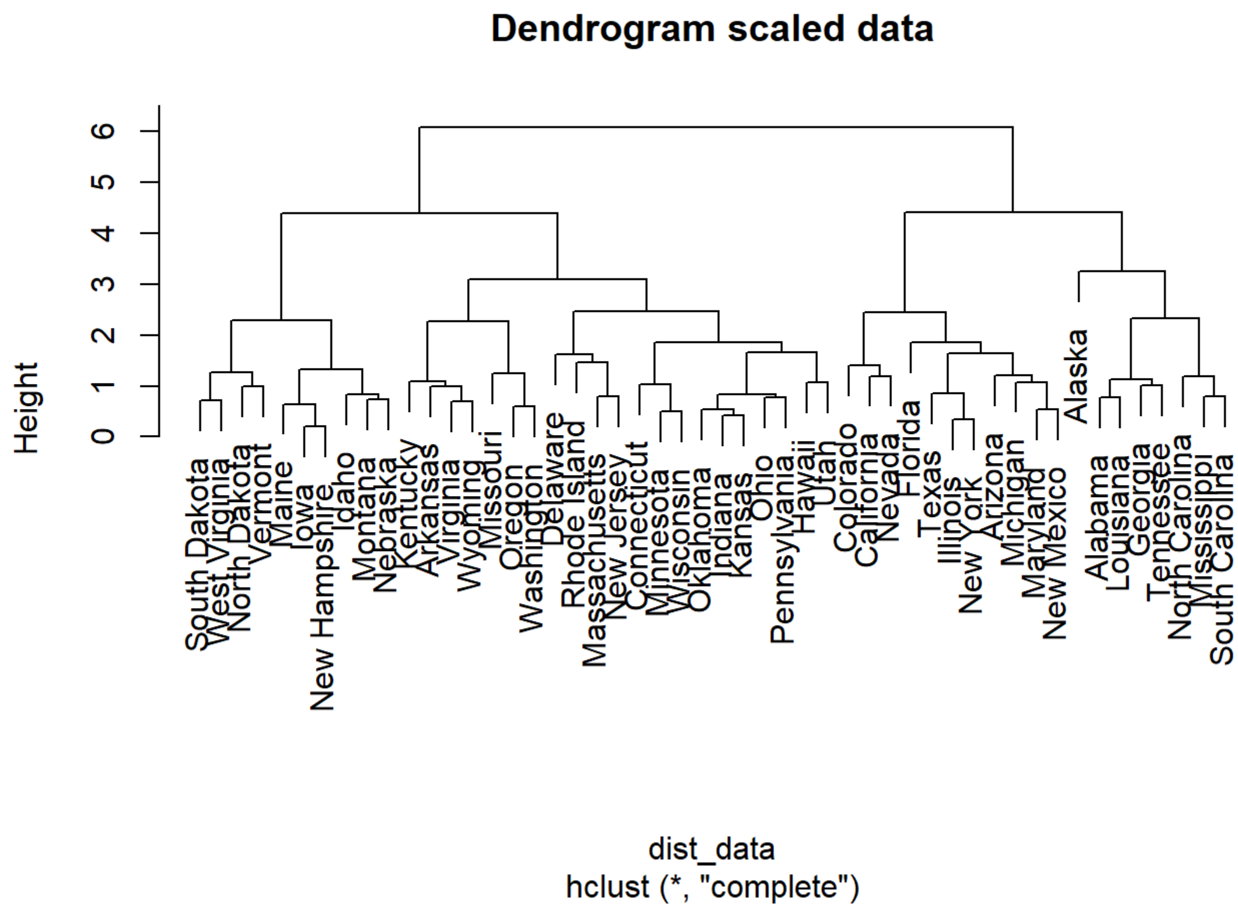
```
dist_data<-dist(scale(USArrests))
dist_data_unscaled <-dist(USArrests)
hclust_cmplt_scaled <- hclust(dist_data, method = 'complete')
hclust_cmplt <- hclust(dist_data_unscaled, method = 'complete')


dendro <- ggdendrogram(hclust_cmplt_scaled)
dendro +
  theme(panel.grid.major.y = element_line(color = "black", size = 0.3)) +
  ggtitle('Method: Complete') +
  geom_hline(yintercept = 3.5, col=2, lwd=1, lty=2)
```

```
## Warning: The `size` argument of `element_line()` is deprecated as of ggplot2 3.4.0.
## ℹ Please use the `linewidth` argument instead.
```

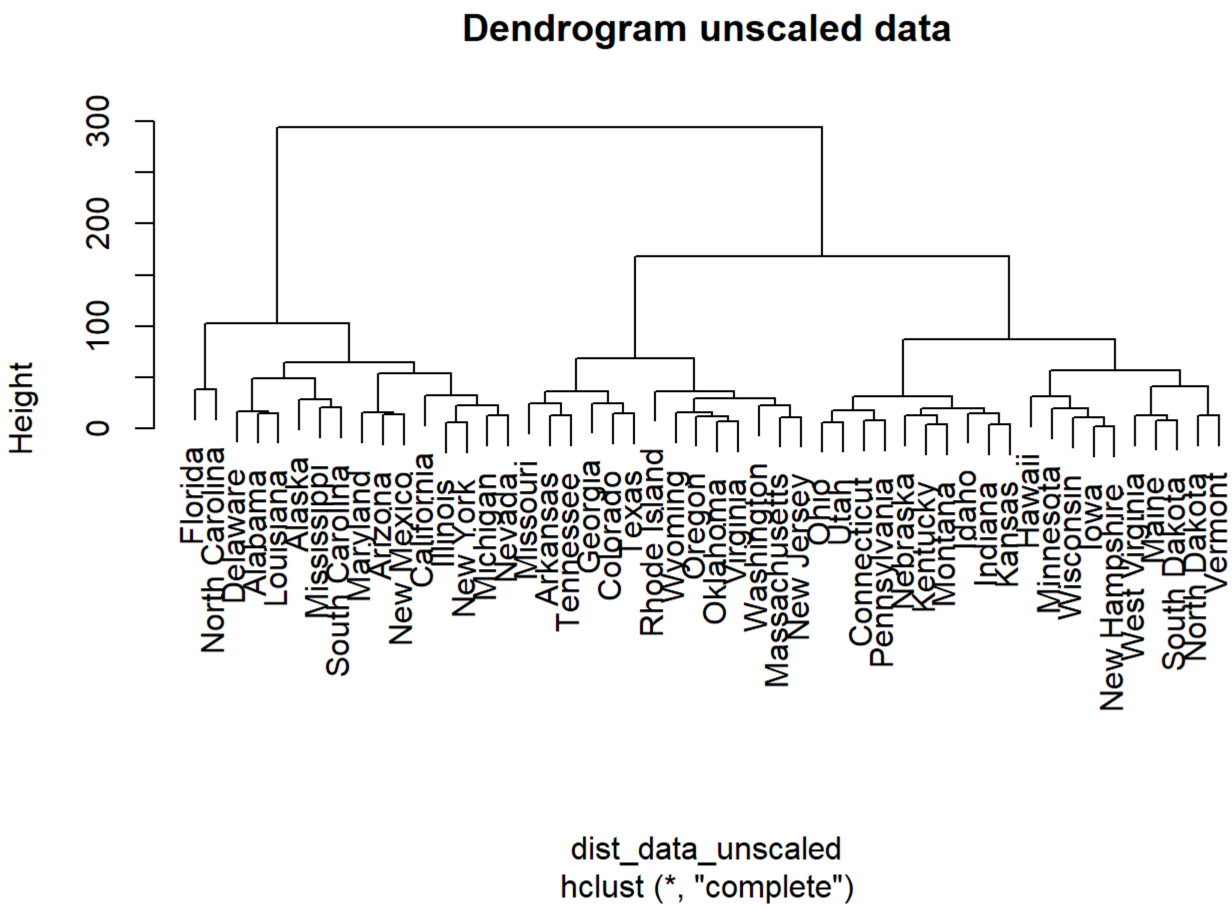
```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
```




```
#dist_data<-dist(scale(USArrests))
#hclust_cmplt_scaled <- hclust(dist_data, method = 'complete')
plot(hclust_cmplt_scaled, main = 'Dendrogram scaled data') #dendrogram
```




```
plot(hclust_cmplt, main='Dendrogram unscaled data') #dendrogram
```

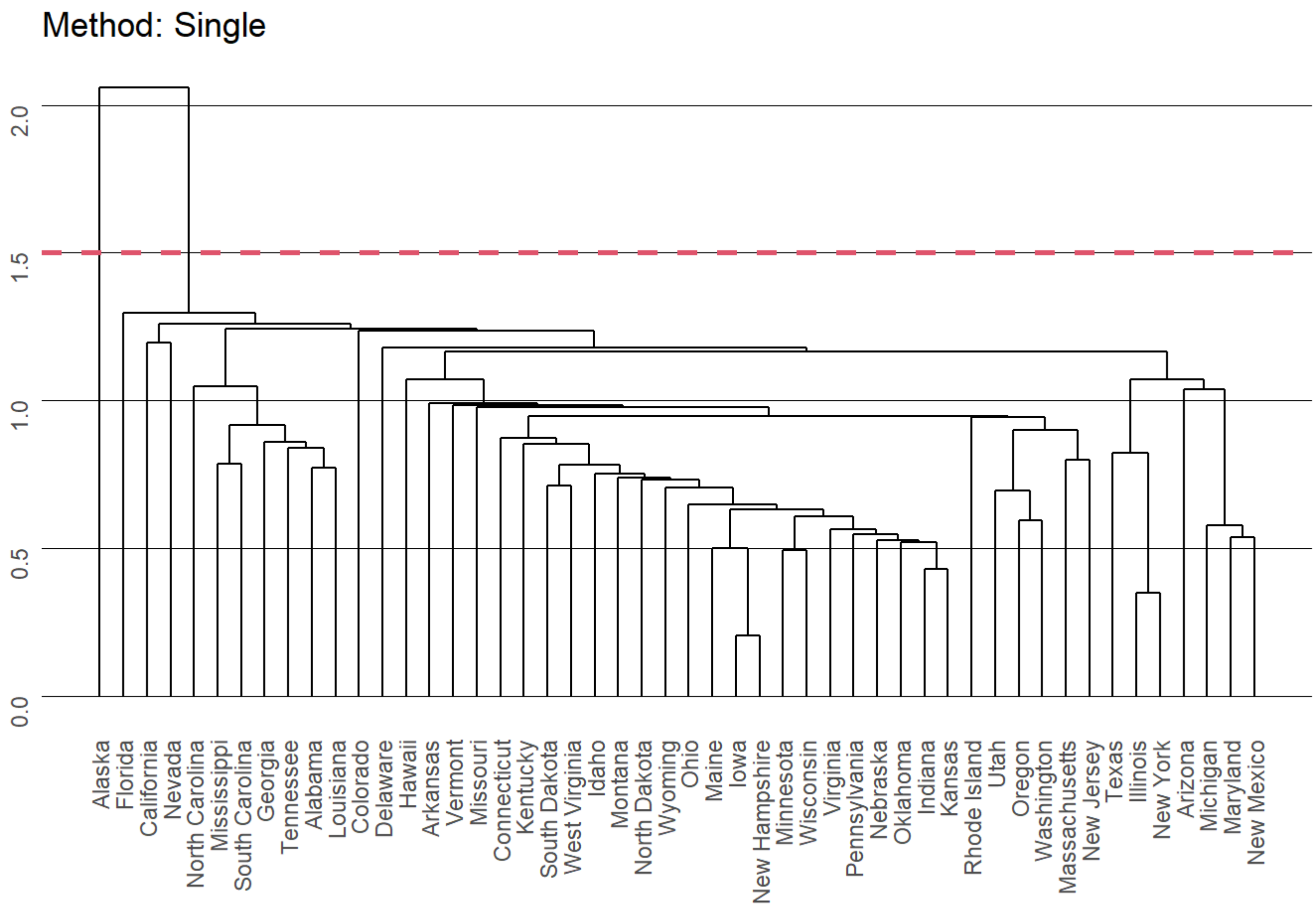


In this example, I would choose **4** clusters. As the dendrogram suggests, the difference is only quite large for 4 individual paths, the distance among the smaller potential clusters is quite small.

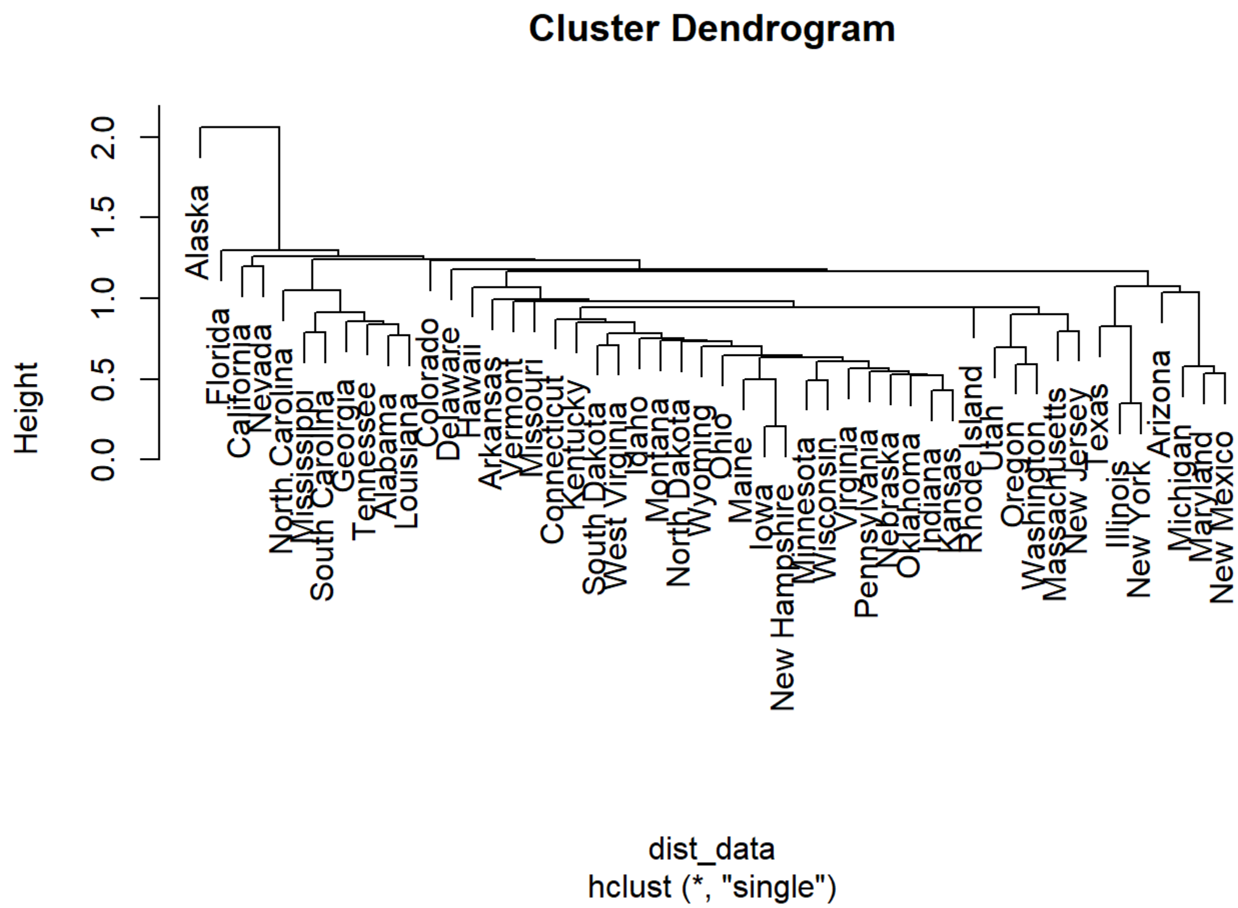
(b)

Run hierarchical clustering analysis with scaled “USArrests” data. Use the ‘single’ method as linkage function. Plot a dendrogram and argue how many clusters you would choose.

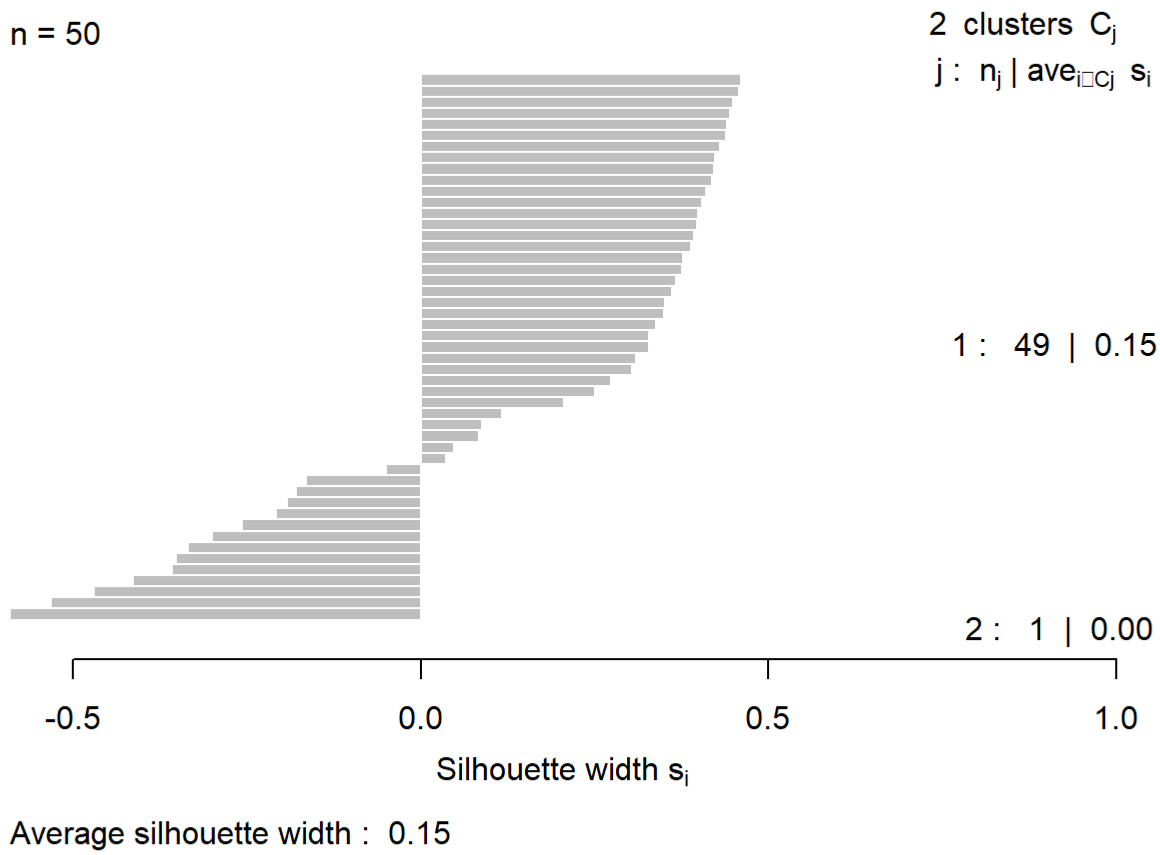
```
hclust_sngl <- hclust(dist_data, method = 'single')
dendro <- ggdendrogram(hclust_sngl)

dendro +
  theme(panel.grid.major.y = element_line(color = "black", size = 0.3)) +
  ggtitle('Method: Single') +
  geom_hline(yintercept = 1.5, col=2, lwd=1, lty=2)
```




```
plot(hclust_sngl)
```



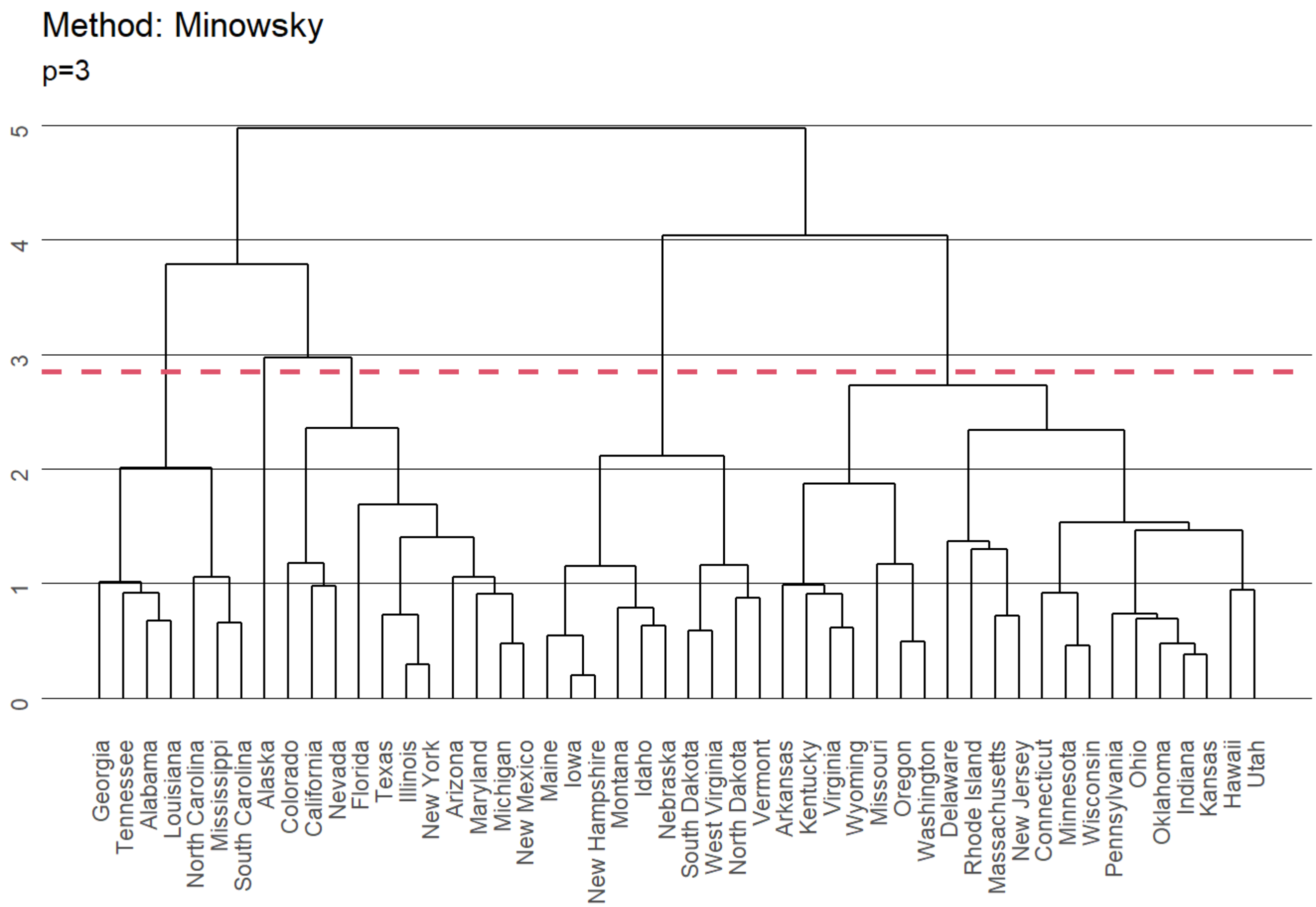

```
k_clust <- cutree(hclust_sngl, k = 2)
silhouette(k_clust, dist(dist_data)) %>% plot(main='')
```



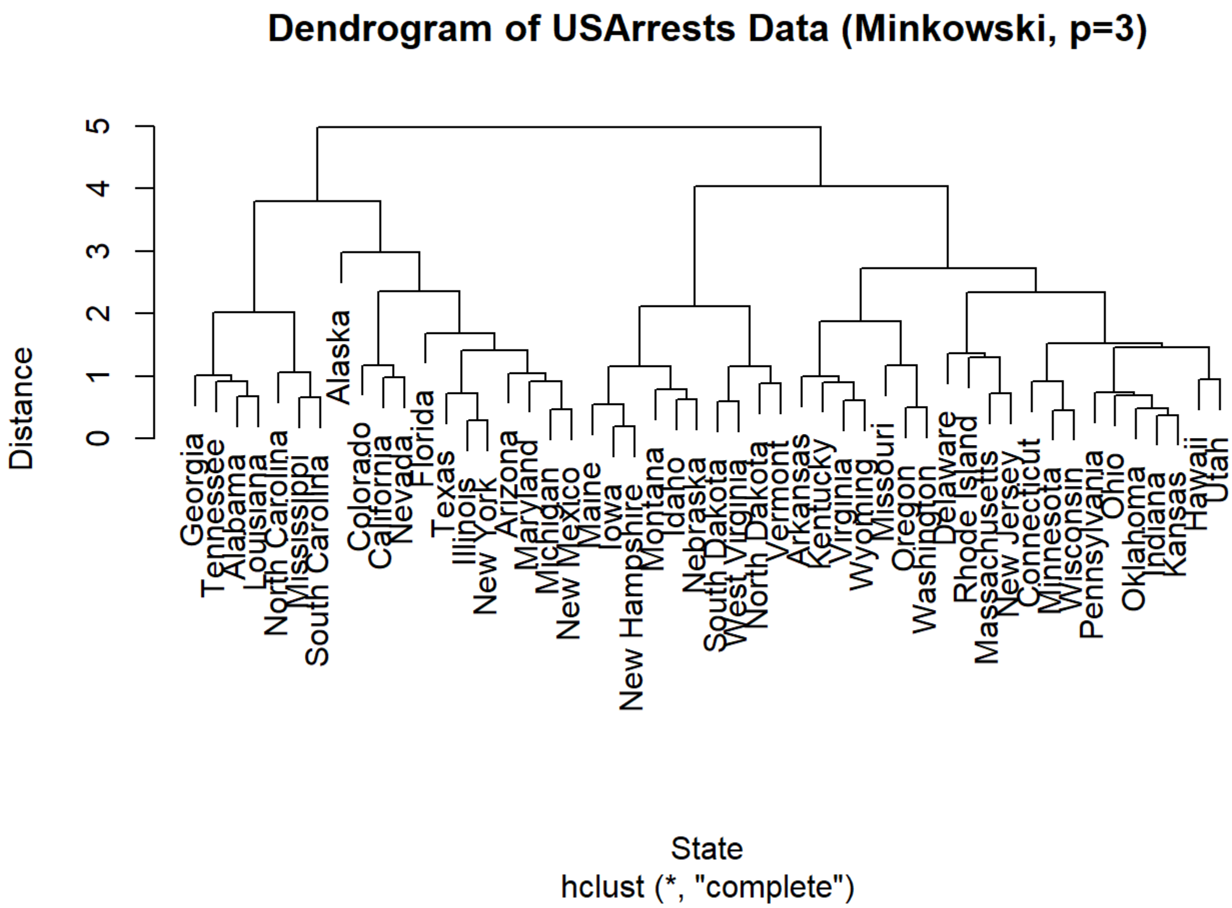
(c)

Run hierarchical clustering analysis with scaled "USArrests" data with Minkowski distance measure (p=3). Use the 'complete' method as linkage function. Plot a dendrogram and argue how many clusters you would choose.

```
# scale the data
scaled_data <- scale(USArrests)

# hierarchical clustering with Minkowski distance measure
hc_minkowski <- hclust(dist(scaled_data, method = "minkowski", p = 3), method = "complete")
dendro <- ggdendrogram(hc_minkowski)

dendro +
  theme(panel.grid.major.y = element_line(color = "black", size = 0.3)) +
  ggtitle('Method: Minowsky', subtitle = 'p=3') +
  geom_hline(yintercept = 2.85, col=2, lwd=1, lty=2)
```




```
# Plot dendrogram
plot(hc_minkowski, main = "Dendrogram of USArrests Data (Minkowski, p=3)", xlab = "State", ylab
= "Distance")
```




```
k2_clusters <- cutree(hc_minkowski, k = 5)

silhouette(k2_clusters, dist(scaled_data, method = "minkowski", p = 3)) %>% plot(main='')
```

(3.3) k-means clustering on scaled “USArrests” data

(a)

Run k-means clustering analysis with 2 clusters. Which cluster would you denote as potentially high/low crime cluster? Is Ohio high or low crime state?

--> See: `Clustering.ipynb` on GitHub

```
data(USArrests)
# determining index location of Ohio
iloc_ohio <- which(rownames(USArrests)=="Ohio")
# scale the data
#USArrests <- data.frame(scale(USArrests))
# scaling the data
arrests_sc <- scale(USArrests, center=T, scale=T) %>% dist()

km <- kmeans(arrests_sc, centers = 2, nstart = 10)
km$cluster[iloc_ohio]
```

```
## Ohio
##    2
```

```
# Load the USArrests dataset
data("USArrests")

# k-means clustering with 2 clusters
set.seed(123) # for reproducibility
k <- 2
km <- kmeans(USArrests, centers = k, nstart = 41)

# tag each state according to their cluster
clust_labs <- ifelse(km$cluster == 1, "high crime",  "~low crime")
USArrests_clust <- data.frame(USArrests, cluster = clust_labs)
USArrests_clust
```

```
##                Murder Assault UrbanPop Rape    cluster
## Alabama          13.2     236       58 21.2 high crime
## Alaska           10.0     263       48 44.5 high crime
## Arizona           8.1     294       80 31.0 high crime
## Arkansas          8.8     190       50 19.5 high crime
## California        9.0     276       91 40.6 high crime
## Colorado          7.9     204       78 38.7 high crime
## Connecticut       3.3     110       77 11.1 ~low crime
## Delaware          5.9     238       72 15.8 high crime
## Florida          15.4     335       80 31.9 high crime
## Georgia          17.4     211       60 25.8 high crime
## Hawaii            5.3      46       83 20.2 ~low crime
## Idaho             2.6     120       54 14.2 ~low crime
## Illinois         10.4     249       83 24.0 high crime
## Indiana           7.2     113       65 21.0 ~low crime
## Iowa              2.2      56       57 11.3 ~low crime
## Kansas            6.0     115       66 18.0 ~low crime
## Kentucky          9.7     109       52 16.3 ~low crime
## Louisiana        15.4     249       66 22.2 high crime
## Maine             2.1      83       51  7.8 ~low crime
## Maryland         11.3     300       67 27.8 high crime
## Massachusetts     4.4     149       85 16.3 ~low crime
## Michigan         12.1     255       74 35.1 high crime
## Minnesota         2.7      72       66 14.9 ~low crime
## Mississippi      16.1     259       44 17.1 high crime
## Missouri          9.0     178       70 28.2 ~low crime
## Montana           6.0     109       53 16.4 ~low crime
## Nebraska          4.3     102       62 16.5 ~low crime
## Nevada           12.2     252       81 46.0 high crime
## New Hampshire     2.1      57       56  9.5 ~low crime
## New Jersey        7.4     159       89 18.8 ~low crime
## New Mexico       11.4     285       70 32.1 high crime
## New York         11.1     254       86 26.1 high crime
## North Carolina   13.0     337       45 16.1 high crime
## North Dakota      0.8      45       44  7.3 ~low crime
## Ohio              7.3     120       75 21.4 ~low crime
## Oklahoma          6.6     151       68 20.0 ~low crime
## Oregon            4.9     159       67 29.3 ~low crime
## Pennsylvania      6.3     106       72 14.9 ~low crime
## Rhode Island      3.4     174       87  8.3 ~low crime
## South Carolina   14.4     279       48 22.5 high crime
## South Dakota      3.8      86       45 12.8 ~low crime
## Tennessee        13.2     188       59 26.9 high crime
## Texas            12.7     201       80 25.5 high crime
## Utah              3.2     120       80 22.9 ~low crime
## Vermont           2.2      48       32 11.2 ~low crime
## Virginia          8.5     156       63 20.7 ~low crime
## Washington        4.0     145       73 26.2 ~low crime
## West Virginia     5.7      81       39  9.3 ~low crime
## Wisconsin         2.6      53       66 10.8 ~low crime
## Wyoming           6.8     161       60 15.6 ~low crime
```

```
# Print the number of states in each cluster
table(USArrests_clust$cluster)
```

```
##
## ~low crime high crime
##         29         21
```

```
cat('\n')
```

```
Ohio_clust <- USArrests_clust[which(row.names(USArrests_clust) ==
                                      "Ohio"), "cluster"]

Ohio_clust
```

```
## [1] "~low crime"
```

```
USArrests_clust[which(rownames(USArrests_clust)=="Ohio"), ]
```

```
##      Murder Assault UrbanPop Rape    cluster
## Ohio    7.3     120       75 21.4 ~low crime
```

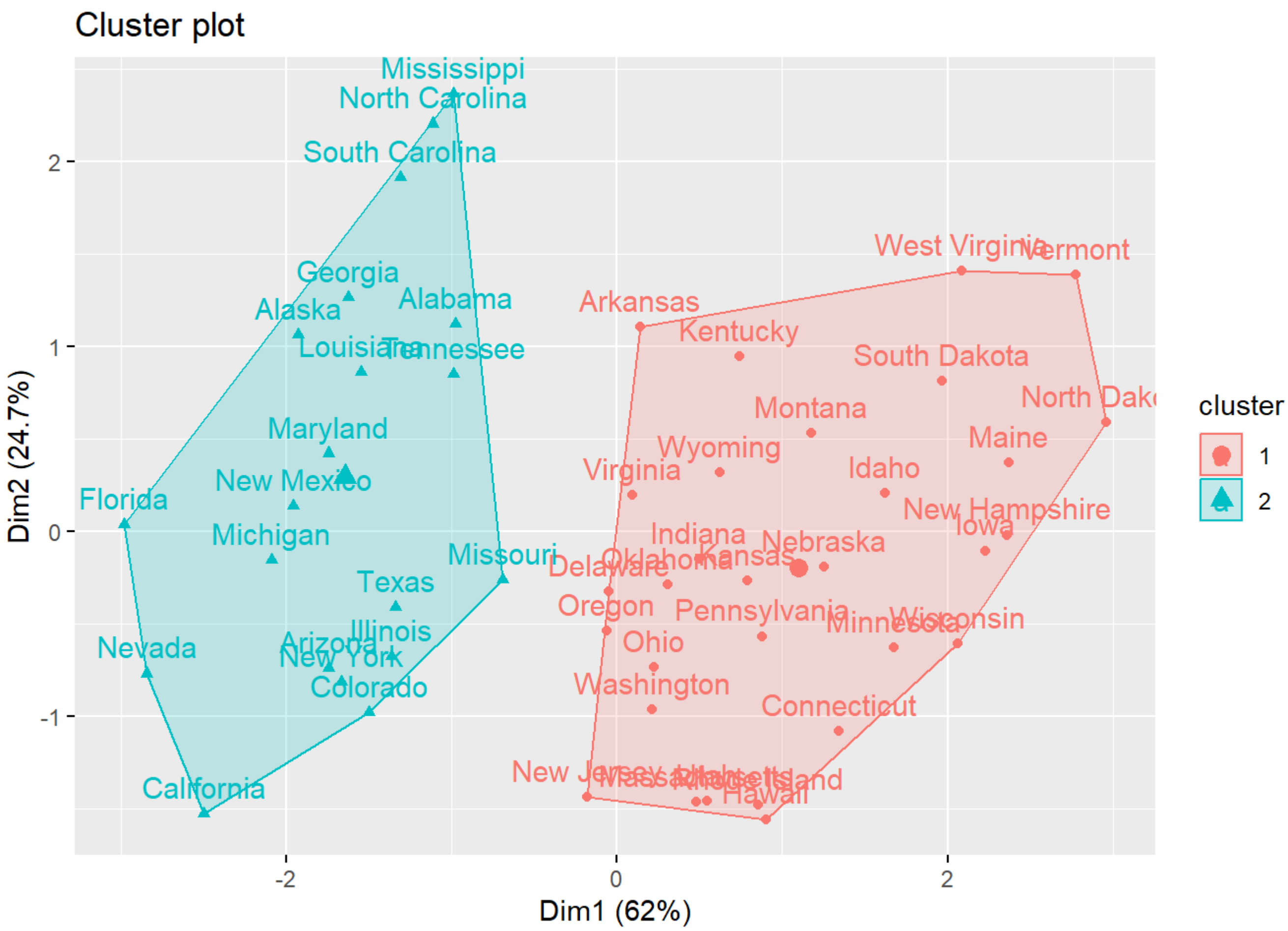
```
data(USArrests)
iloc_ohio <- which(rownames(USArrests)=="Ohio")
USArrests <- data.frame(scale(USArrests))



set.seed(200996)
A <- kmeans(USArrests, 2, nstart = 100)
A$centers
```

```
##      Murder    Assault   UrbanPop       Rape
## 1 -0.669956 -0.6758849 -0.1317235 -0.5646433
## 2  1.004934  1.0138274  0.1975853  0.8469650
```

```
#A$cluster[iloc_ohio]
cat("\nOhio is in cluster:\t", A$cluster[iloc_ohio])
```

```
##
## Ohio is in cluster:   1
```

```
fviz_cluster(A, data = USArrests)
```




```
#? fviz_cluster
```

(b)

Run k-means clustering analysis with 4 clusters. How can you characterize these clusters? In which cluster is Ohio?

see table above!!!

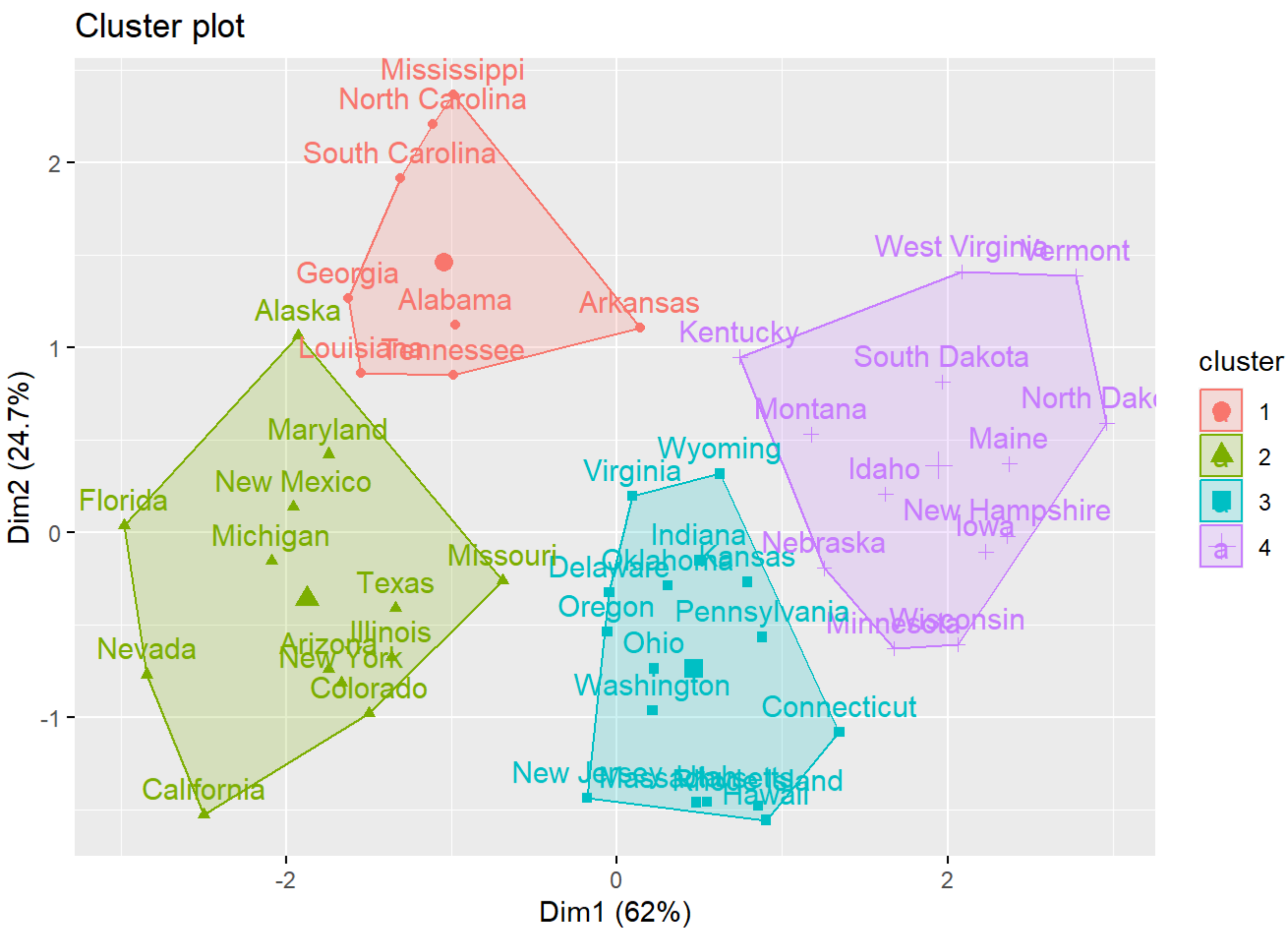
```
B <- kmeans(USArrests, 4, nstart = 100)
B$centers
```

```
##       Murder    Assault   UrbanPop        Rape
## 1  1.4118898  0.8743346 -0.8145211  0.01927104
## 2  0.6950701  1.0394414  0.7226370  1.27693964
## 3 -0.4894375 -0.3826001  0.5758298 -0.26165379
## 4 -0.9615407 -1.1066010 -0.9301069 -0.96676331
```

```
#B$cluster[iloc_ohio]
cat("\nOhio is in cluster:\t", B$cluster[iloc_ohio])
```

```
##
## Ohio is in cluster:   3
```

```
fviz_cluster(B, data = USArrests)
```



(c)

Remove the item ‘UrbanPop’ from the data set. Run k-means clustering analysis with 3 clusters. How can you characterize these clusters? In which cluster is Ohio?

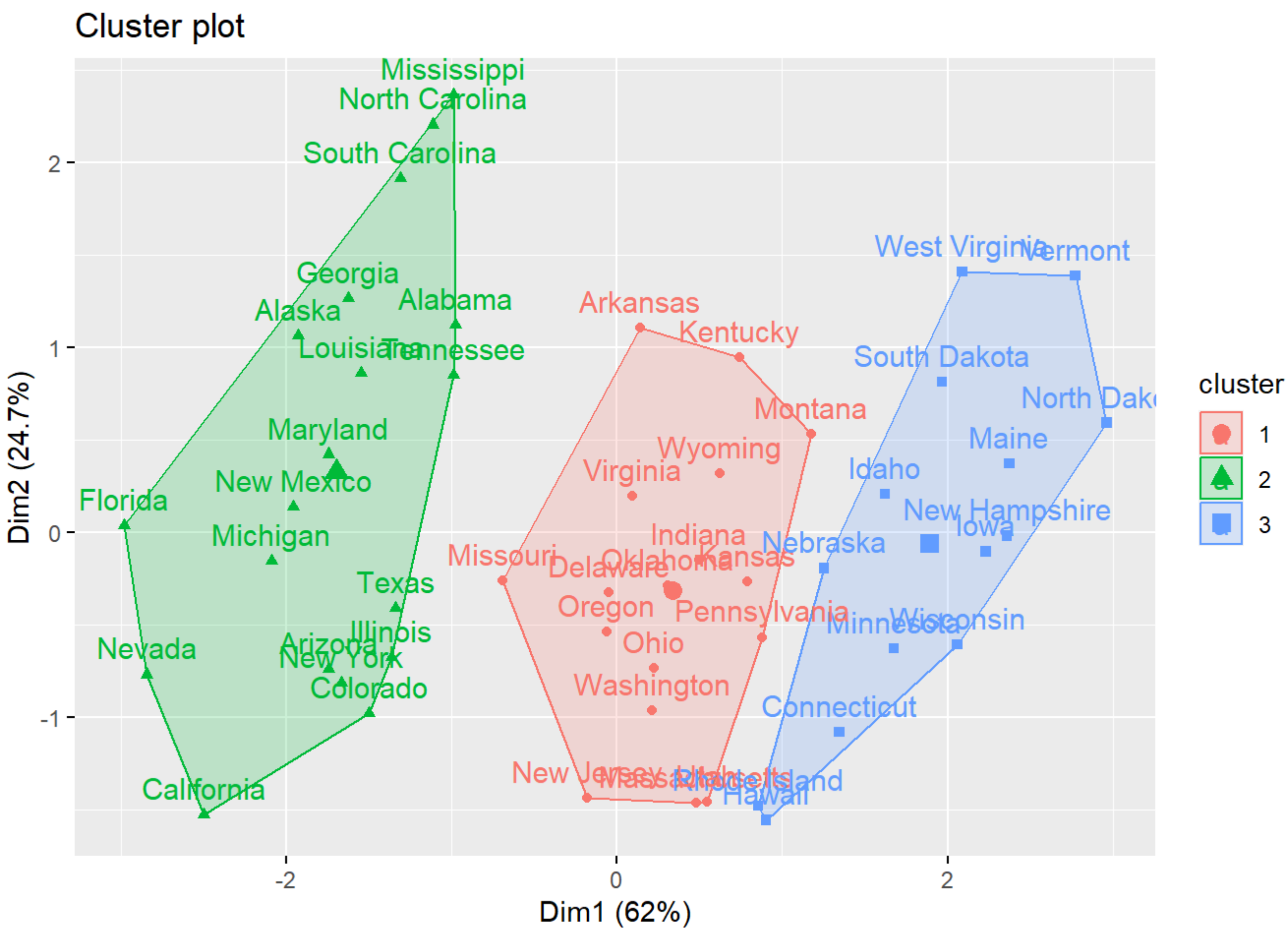
```
arrests3dim <- USArrests %>% select(-UrbanPop)
C <- kmeans(arrests3dim, 3, nstart = 100)
C$centers
```

```
##       Murder   Assault       Rape
## 1 -0.2754591 -0.299928 -0.1233698
## 2  1.0431796  1.062614  0.8523875
## 3 -1.0812577 -1.077921 -1.0070054
```

```
cat("\nOhio is in cluster:\t", C$cluster[iloc_ohio])
```

```
##
## Ohio is in cluster:   1
```

```
fviz_cluster(C, data = USArrests)
```



(3.4) DBSCAN clustering on scaled “USArrests” data

(a)

Run DBSCAN clustering analysis with size of the epsilon neighborhood (threshold density) equal 0.5 (i.e. eps = 0.5) and number of minimum points in the eps region equal 3. How many clusters can you identify? In which cluster is Ohio?

```
# suppress warnings
options(warn = 2)

# load data afresh
data(USArrests)
USArrests <- data.frame(scale(USArrests))

db1 <- dbscan(USArrests, eps = 0.5, MinPts = 3)
```

```
## Error in dbscan(USArrests, eps = 0.5, MinPts = 3): (converted from warning) converting argume
nt MinPts (fpc) to minPts (dbscan)!
```

```
fviz_cluster(db1, USArrests, geom = c("point","text"), labelsize=8,
             xlab=colnames(USArrests)[1], ylab=colnames(USArrests)[2]) +
  labs(subtitle = 'epsilon = 0.5, min.points = 3')
```

```
## Error in fviz_cluster(db1, USArrests, geom = c("point", "text"), labelsize = 8, : object 'db
1' not found
```

```
fviz_cluster(db1, USArrests, geom = c("point","text"), labelsize=8, axes=c(1,4),
             xlab=colnames(USArrests)[1], ylab=colnames(USArrests)[4]) +
  labs(subtitle = 'epsilon = 0.5, min.points = 3')
```

```
## Error in fviz_cluster(db1, USArrests, geom = c("point", "text"), labelsize = 8, : object 'db
1' not found
```

```
# since the order of the states is preserved we can label the cluster
# sequences by their corresponding state and eventually find Ohio
names(db1$cluster) <- USArrests %>% rownames()
```

```
## Error in names(db1$cluster) <- USArrests %>% rownames(): object 'db1' not found
```

```
cat('Ohio is in cluster', db1$cluster[which(names(db1$cluster)=="Ohio")])
```

```
## Error in cat("Ohio is in cluster", db1$cluster[which(names(db1$cluster) == : object 'db1' not
found
```

(b)

Run DBSCAN clustering analysis with size of the epsilon neighborhood (thresh□old density) equal 1 (i.e. eps = 1) and number of minimum points in the eps region equal 3. How many clusters can you identify? In which cluster is Ohio?

```
db2 <- dbscan(USArrests, eps = 1, MinPts = 3)
```

```
## Error in dbscan(USArrests, eps = 1, MinPts = 3): (converted from warning) converting argument
MinPts (fpc) to minPts (dbscan)!
```

```
fviz_cluster(db2, USArrests, geom = c("point","text"), labelsize=8,
             xlab=colnames(USArrests)[1], ylab=colnames(USArrests)[2]) +
  labs(subtitle="epsilon = 1, min.points = 3")
```

```
## Error in fviz_cluster(db2, USArrests, geom = c("point", "text"), labelsize = 8, : object 'db
2' not found
```

```
fviz_cluster(db2, USArrests, geom = c("point","text"), labelsize=8, axes=c(1,4),
             xlab=colnames(USArrests)[1], ylab=colnames(USArrests)[4]) +
  labs(subtitle = 'epsilon = 0.5, min.points = 3')
```

```
## Error in fviz_cluster(db2, USArrests, geom = c("point", "text"), labelsize = 8, : object 'db
2' not found
```

```
# since the order of the states is preserved we can label the cluster
# sequences by their corresponding state and eventually find Ohio
names(db2$cluster) <- USArrests %>% rownames()
```

```
## Error in names(db2$cluster) <- USArrests %>% rownames(): object 'db2' not found
```

```
cat('Ohio is in cluster', db2$cluster[which(names(db1$cluster)=="Ohio")])
```

```
## Error in cat("Ohio is in cluster", db2$cluster[which(names(db1$cluster) == : object 'db2' not
found
```

(c)

Run DBSCAN clustering analysis with size of the epsilon neighborhood (threshold density) equal 1.5 (i.e. eps = 1.5) and number of minimum points in the eps region equal 3. How many clusters can you identify? In which cluster is Ohio?

```
db3 <- dbscan(USArrests, eps = 1.5, MinPts = 3)
```

```
## Error in dbscan(USArrests, eps = 1.5, MinPts = 3): (converted from warning) converting argume
nt MinPts (fpc) to minPts (dbscan)!
```

```
fviz_cluster(db3, USArrests, geom = c("point","text"), labelsize=8,
             xlab=colnames(USArrests)[1], ylab=colnames(USArrests)[2]) +
  labs(subtitle = "epsilon = 1.5, min.points = 3")
```

```
## Error in fviz_cluster(db3, USArrests, geom = c("point", "text"), labelsize = 8, : object 'db
3' not found
```

```
names(db3$cluster) <- USArrests %>% rownames()
```

```
## Error in names(db3$cluster) <- USArrests %>% rownames(): object 'db3' not found
```

```
cat('Ohio is in cluster', db3$cluster[which(names(db1$cluster)=="Ohio")])
```

```
## Error in cat("Ohio is in cluster", db3$cluster[which(names(db1$cluster) == : object 'db3' not
found
```

(3.5) Clustering analysis of the countries

Variable	Explanation
country	self explanatory
population	total population of respective country
life_expect	life expect. at birth assuming const. patterns of mortality the entire life (no extraord. risk)
fertility	children per woman within “child bearing age”
fertility_adol	children per 1000 women aged 15-19
mortal_5	children dying before turning 5
mobile_phones	communication tech infrastructure (phones, internet)
migration	net migration (immigration - emigration)
electricity	% of pop having access to electricity
gdp (per capita), inflation (CPI), pop growth, unrate	self explanatory

(a)

Use the socioeconomic profiles of countries in 2020 (main source: Worldbank) and check the considered indicators.

```
#"/Users/markuskofler/OneDrive - Alpen-Adria Universität Klagenfurt/R/DataAnalytics/Ex3/countrie
s_indicators.xlsx"
path_to_file <- "C:/Users/HP/OneDrive - Alpen-Adria Universität Klagenfurt/R/DataAnalytics/Ex3/c
ountries_indicators.xlsx"
```

```
se <- read.csv('https://raw.githubusercontent.com/MarkusStefan/Data_Analytics/main/Exercise3/cou
ntries_data1.csv')
exp <- readxl::read_xlsx(path_to_file)[1:2]



se %>% colnames()
```

```
##  [1] "country"           "population"        "life_expect"
##  [4] "fertility"         "fertility_adol"    "mortal_5"
##  [7] "mobile_phones"     "migration"         "electricity"
## [10] "gdp"               "inflation"         "pop_growth"
## [13] "surface_area"      "unemployment_rate"
```

```
exp
```

```
## # A tibble: 13 × 2
##    Indicator         `Indicator Name`
##    <chr>             <chr>
##  1 population        Population, total
##  2 life_expect       Life expectancy at birth, total (years)
##  3 fertility         Fertility rate, total (births per woman)
##  4 fertility_adol    Adolescent fertility rate (births per 1,000 women ages 15-…
##  5 mortal_5          Mortality rate, under-5 (per 1,000 live births)
##  6 mobile_phones     Mobile cellular subscriptions (per 100 people)
##  7 migration         Net migration
##  8 electricity       Access to electricity (% of population)
##  9 gdp               GDP per capita (constant 2015 US$)
## 10 inflation         Inflation, consumer prices (annual %)
## 11 pop_growth        Population growth (annual %)
## 12 surface_area      Surface area (sq. km)
## 13 unemployment_rate Unemployment, total (% of total labor force) (modeled ILO …
```

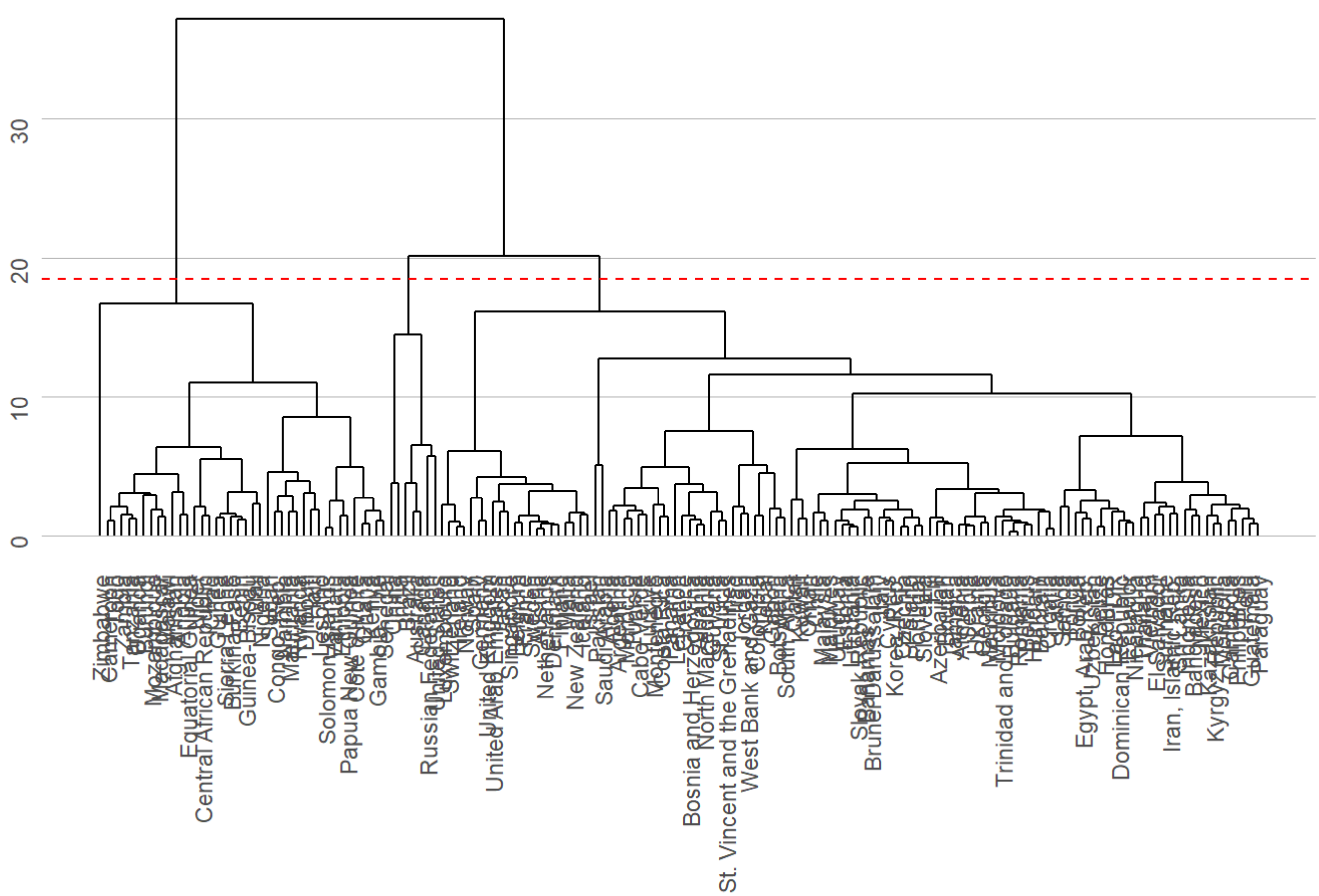
```
se %>% head()
```

```
##       country population life_expect fertility fertility_adol mortal_5
## 1 Afghanistan   38972230       62.58      4.75          84.30     57.8
## 2     Albania    2837849       76.99      1.40          14.67      9.4
## 3     Algeria   43451666       74.45      2.94          12.06     22.9
## 4      Angola   33428486       62.26      5.37         139.83     72.1
## 5   Argentina   45376763       75.89      1.91          39.87      7.7
## 6     Armenia    2805608       72.17      1.58          18.57     11.3
##   mobile_phones migration electricity      gdp inflation pop_growth
## 1         58.19    166821       97.70   553.04      5.60       3.13
## 2         91.35     -9117      100.00  4410.46      1.62      -0.57
## 3        104.84    -18797       99.80  3873.51      2.42       1.73
## 4         43.81      7557       46.89  2347.79     22.27       3.27
## 5        121.60      2344      100.00 11341.27     42.02       0.97
## 6        124.35    -12825      100.00  4256.13      1.21      -0.53
##   surface_area unemployment_rate
## 1       652860             11.71
## 2        28750             13.07
## 3      2381741             12.25
## 4      1246700             10.35
## 5      2780400             11.46
## 6        29740             12.18
```

(b)

Run hierarchical clustering analysis of the countries.

```
# setting country as index as functions only work with numerical datase
se_ <- se
rownames(se_) <- se$country
tryCatch(
  expr = {
    se_ <- se_ %>% select(-country)
  },

  finally = {
    dist <- se_ %>% scale() %>% dist()

    hc <- hclust(dist, method = "ward.D2")

    ggdendrogram(hc) +
      theme(panel.grid.major.y = element_line(color = "gray70", size = 0.3),
            panel.grid.minor.y = element_blank())
  }
)
dist <- se_ %>% scale() %>% dist()
hc <- hclust(dist, method = "ward.D2")

ggdendrogram(hc) +
  theme(panel.grid.major.y = element_line(color = "gray70", size = 0.05),
            panel.grid.minor.y = element_blank()) +
  geom_hline(yintercept = 18.5, col='red', lty=2)
```



I would choose 3 clusters using the HC method.

(c)

Run k-means clustering analysis of the countries.

```
set.seed(45)
scaled <- scale(se_, center=T, scale=T)

km <- kmeans(scaled, 3, nstart = 100, algorithm = "Hartigan-Wong")
#km$centers
#km$cluster

which(names(km$cluster)== "Austria") # 8
```

```
## [1] 8
```

```
km$cluster[8]
```

```
## Austria
##       2
```

```
#which(km$cluster==2) %>% names()
#which(km$cluster == 3) %>% names()
#which(km$cluster == 1) %>% names()

fviz_cluster(km, data=se_, axes=c(1,2), repel=T, xlab=colnames(se_)[1], ylab=colnames(se_)[2])
```

```
## Error: (converted from warning) ggrepel: 144 unlabeled data points (too many overlaps). Consi
der increasing max.overlaps
```

Also with K-Means, 3 Clusters seem to be the best choice.