Clustering - discovering groups in data

Søren Højsgaard

17/9/2024

Contents

1 What is clustering

1. Clustering is an example of unsupervised learning where input data have no labels attached.
2. Clustering can be defined as the task of dividing the data points into the certain number of groups or clusters so that the data points in the same group or cluster share similar characteristics.
3. That is, the aim of clustering analysis is to make homogeneous subgroups called clusters.
4. It is difficult (if not impossible) to objectively verify that such clusters represent any truth in the matter being studied.

2 Example: Crime data

```
crime <- doBy::crime_rate
head(crime, 3)
```

```
##         murder rape assault robbery burglary larceny autotheft
## Alabama   14.2 25.2     278    96.8     1136    1882       281
## Alaska    10.8 51.6     284    96.8     1332    3370       753
## Arizona    9.5 34.2     312   138.2     2346    4467       440
```

```
st <- rownames(crime)
crime2 <- scale(crime) ## Standardize data
head(crime2, 3)
```

```
##          murder    rape assault robbery burglary larceny autotheft
## Alabama  1.747 -0.0496   0.668  -0.309   -0.362  -1.087    -0.501
## Alaska   0.868  2.4040   0.725  -0.309    0.092   0.962     1.943
## Arizona  0.532  0.7868   1.007   0.160    2.438   2.474     0.320
```

3 Clustering states

3.1 How similar are states?

One approach: Compute all pairs of Euclidian distances:

```
crime2[1:3,]
```

```
##          murder    rape assault robbery burglary larceny autotheft
## Alabama  1.747 -0.0496   0.668  -0.309   -0.362  -1.087    -0.501
## Alaska   0.868  2.4040   0.725  -0.309    0.092   0.962     1.943
## Arizona  0.532  0.7868   1.007   0.160    2.438   2.474     0.320
```

```
x <- crime2[1,]
y <- crime2[2,]
sqrt(sum((x - y)^2))
```

```
## [1] 4.14
```

```
x <- crime2[1,]
y <- crime2[3,]
sqrt(sum((x - y)^2))
```

```
## [1] 4.87
```

Compute all pairs of Euclidian differences between states (that is, between rows in the data frame):

```
n <- 50        # states
n * (n-1) / 2 # number of pairs
```

```
## [1] 1225
```

```
dvec <- dist(crime2, method = "euclidian")
length(dvec)
```

```
## [1] 1225
```

```
dvec[1:4]
```

```
## [1] 4.14 4.87 1.73 5.01
```

```
as.matrix(dvec)[1:4, 1:4]
```

```
##           Alabama Alaska Arizona Arkansas
## Alabama      0.00   4.14    4.87     1.73
## Alaska       4.14   0.00    3.67     4.43
## Arizona      4.87   3.67    0.00     5.17
## Arkansas     1.73   4.43    5.17     0.00
```

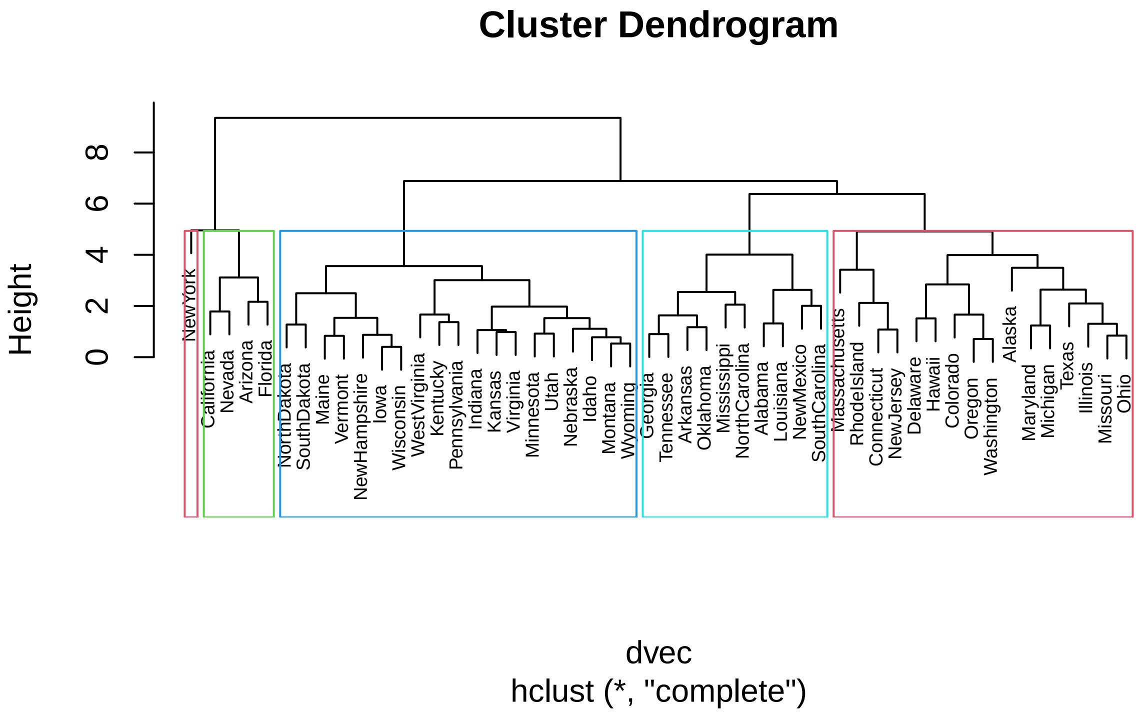
3.2 Cluster states based on distances

```
hc <- hclust(dvec)
hc
```

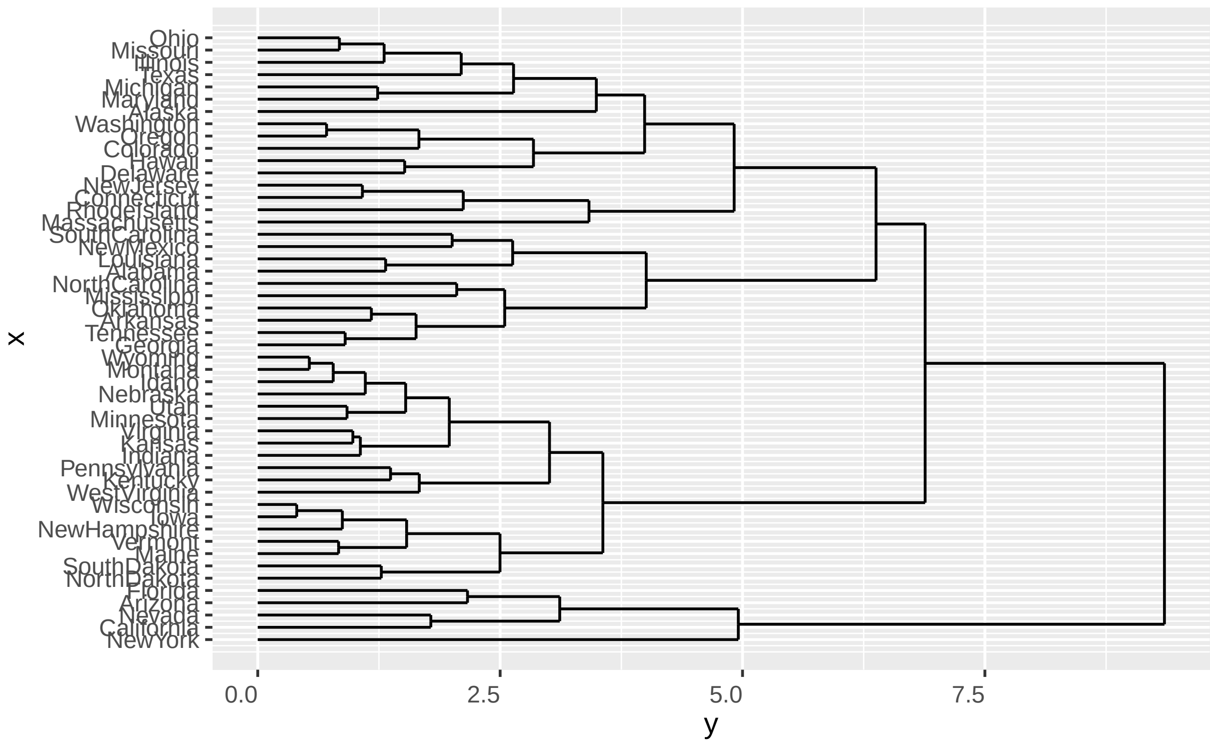
```
##
## Call:
## hclust(d = dvec)
##
## Cluster method   : complete
## Distance         : euclidean
## Number of objects: 50
```

3.3 Display clustering - the dendogram

```r
plot(hc, cex=0.6)
rect.hclust(hc, k = 5, border = 2:5) # add rectangle
```




```r
library(ggdendro)
hc |> ggdendrogram(rotate=TRUE, theme_dendro=FALSE)
```



```r
kvals <- c(4, 5, 7, 9)
cl <- cutree(hc, k=kvals) |> as.data.frame()
cl <- lapply(cl, factor) |> as.data.frame()
names(cl) <- paste0("cluster_", kvals)
rownames(cl) <- hc$labels
```

```r
cl |> head(5)
```

```
##            cluster_4 cluster_5 cluster_7 cluster_9
## Alabama            1         1         1         1
## Alaska             2         2         2         2
## Arizona            3         3         3         3
## Arkansas           1         1         4         4
## California         3         3         3         3
```

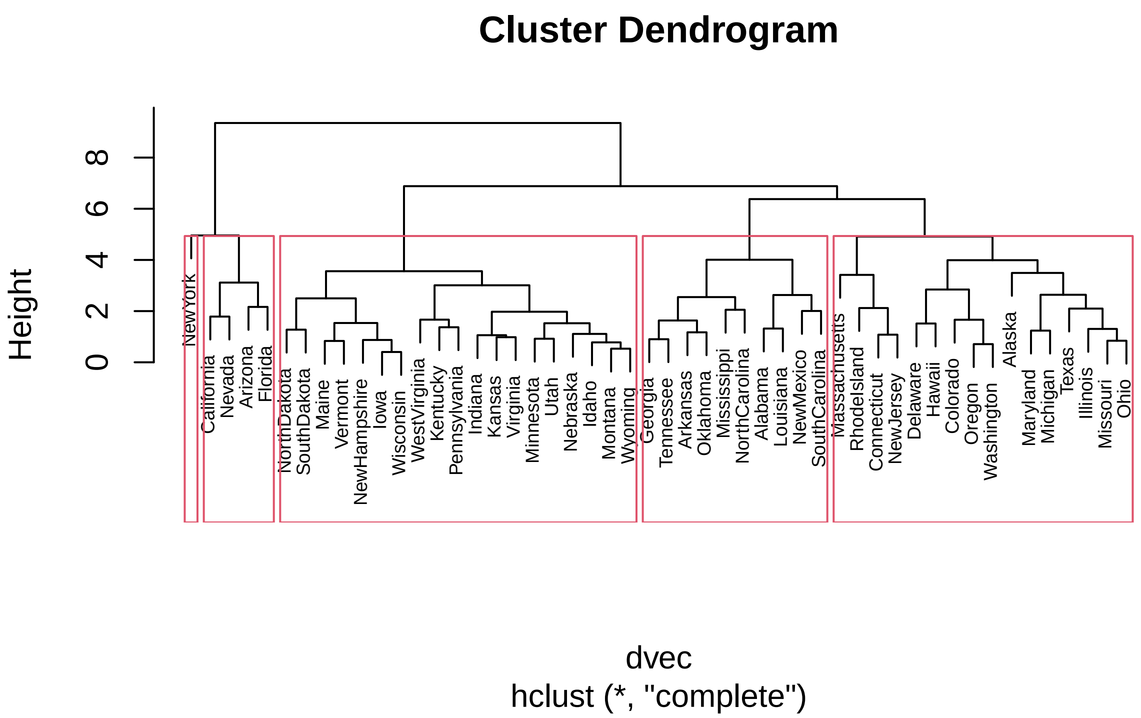
---

4 How many clusters

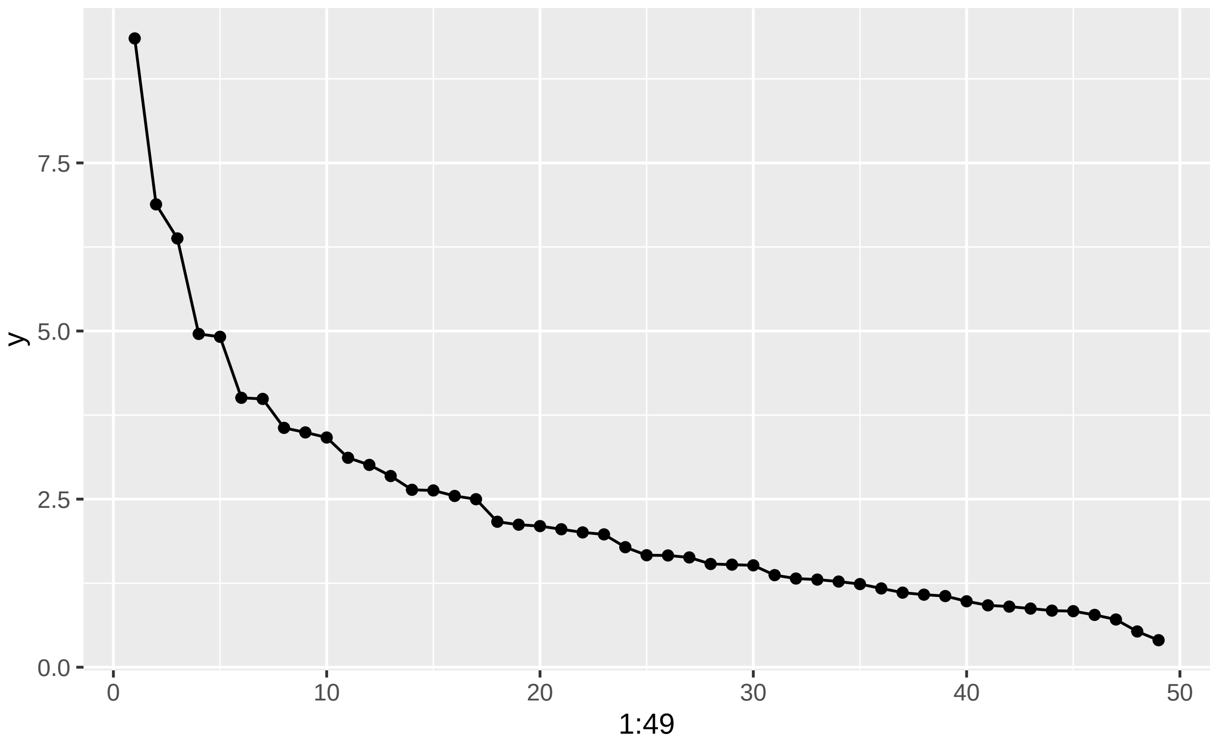
The litterature contains many suggestions for choosing the number of clusters in an objective way.

Perhaps better approach: Small values of `height` indicate that clusters are similar. Hence, let the value if `height` aid in a subjective choice of number of clusters.

```r
plot(hc, cex=0.6)
rect.hclust(hc, k=5)
```




```r
data.frame(y=rev(hc$height)) |>
  ggplot(aes(x=1:49, y=y)) + geom_point() + geom_line()
```



4.1 Relating clusters to cultural regions

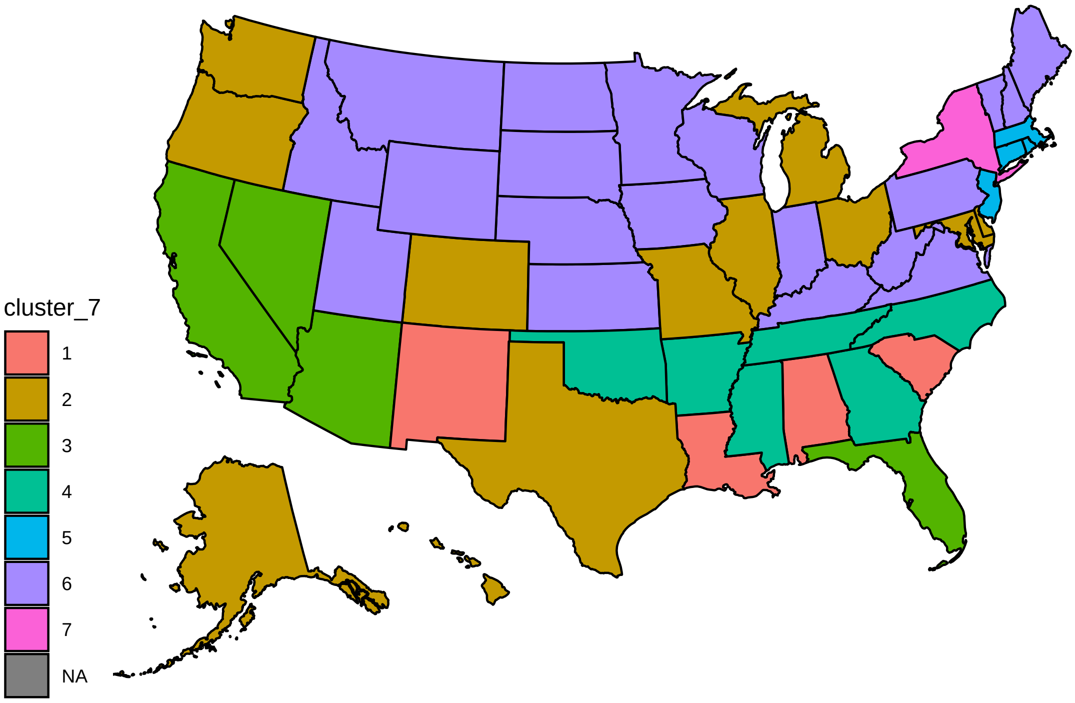
```
library(usmap)
statepop |> head(3)
```

```
## # A tibble: 3 x 4
##   fips  abbr  full     pop_2022
##   <chr> <chr> <chr>       <dbl>
## 1 01    AL    Alabama   5074296
## 2 02    AK    Alaska     733583
## 3 04    AZ    Arizona   7359197
```

```
mydata <- data.frame(state=factor(st), cl)

clusdat <-
  statepop |>
    mutate(name=str_replace_all(full, pattern=" ", replacement="")) |>
    inner_join(mydata, by=c("name"="state"))

clusdat |> head()
```

```
## # A tibble: 6 x 9
##   fips  abbr  full       pop_2022 name   cluster_4 cluster_5 cluster_7 cluster_9
##   <chr> <chr> <chr>         <dbl> <chr>  <fct>     <fct>     <fct>     <fct>
## 1 01    AL    Alabama     5074296 Alaba~ 1         1         1         1
## 2 02    AK    Alaska       733583 Alaska 2         2         2         2
## 3 04    AZ    Arizona     7359197 Arizo~ 3         3         3         3
## 4 05    AR    Arkansas    3045637 Arkan~ 1         1         4         4
## 5 06    CA    California 39029342 Calif~ 3         3         3         3
## 6 08    CO    Colorado    5839926 Color~ 2         2         2         5
```

```
pl1 <- plot_usmap(data=clusdat, values="cluster_7")
pl1
```




```
## # A tibble: 50 x 5
##    state region_9         region_7  region_5  region_4
##    <chr> <chr>            <chr>     <chr>     <chr>
##  1 AK    Alaska og Hawaii West      West      West
##  2 AL    South            South     South     South
##  3 AR    South            South     South     South
##  4 AZ    Southwest        West      West      West
##  5 CA    Pacific Coast    West      West      West
##  6 CO    Rocky Mountains  West      West      West
##  7 CT    New England      Northeast Northeast Northeast
##  8 DE    Mid-Atlantic     South     South     South
##  9 FL    South            South     South     South
## 10 GA    South            South     South     South
## # i 40 more rows
```

USA is in some cases regarded as having 9 regions, in other cases 7 or 5 or 4 regions.

```
states_df
```

```
## # A tibble: 50 x 5
##    state region_9         region_7  region_5  region_4
##    <chr> <chr>            <chr>     <chr>     <chr>
##  1 AK    Alaska og Hawaii West      West      West
##  2 AL    South            South     South     South
##  3 AR    South            South     South     South
##  4 AZ    Southwest        West      West      West
##  5 CA    Pacific Coast    West      West      West
##  6 CO    Rocky Mountains  West      West      West
##  7 CT    New England      Northeast Northeast Northeast
##  8 DE    Mid-Atlantic     South     South     South
##  9 FL    South            South     South     South
## 10 GA    South            South     South     South
## # i 40 more rows
```

```
clusdat |> head()
```

```
## # A tibble: 6 x 9
##   fips  abbr  full       pop_2022 name   cluster_4 cluster_5 cluster_7 cluster_9
##   <chr> <chr> <chr>         <dbl> <chr>  <fct>     <fct>     <fct>     <fct>
## 1 01    AL    Alabama     5074296 Alaba~ 1         1         1         1
## 2 02    AK    Alaska       733583 Alaska 2         2         2         2
## 3 04    AZ    Arizona     7359197 Arizo~ 3         3         3         3
## 4 05    AR    Arkansas    3045637 Arkan~ 1         1         4         4
## 5 06    CA    California 39029342 Calif~ 3         3         3         3
## 6 08    CO    Colorado    5839926 Color~ 2         2         2         5
```

```r
states_df |> head()
```

```
## # A tibble: 6 x 5
##   state region_9         region_7 region_5 region_4
##   <chr> <chr>            <chr>    <chr>    <chr>
## 1 AK    Alaska og Hawaii West     West     West
## 2 AL    South            South    South    South
## 3 AR    South            South    South    South
## 4 AZ    Southwest        West     West     West
## 5 CA    Pacific Coast    West     West     West
## 6 CO    Rocky Mountains  West     West     West
```

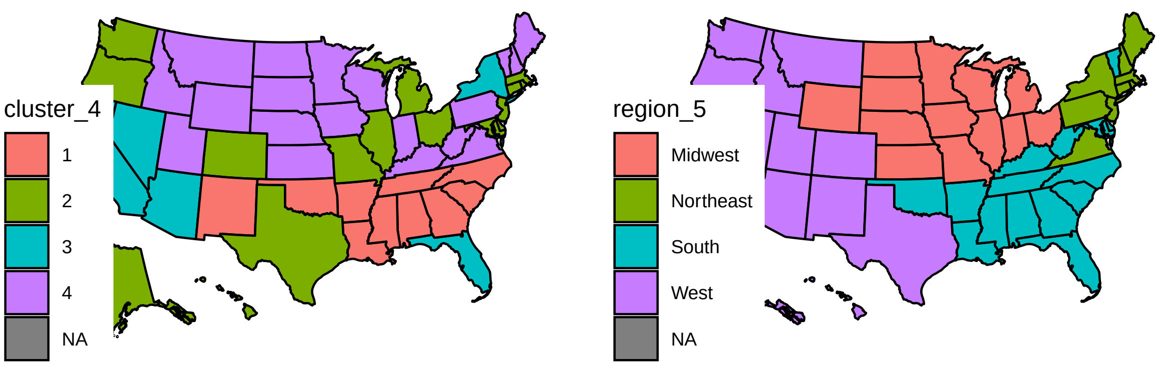
```r
library(dplyr)

clusdat2 <- states_df |> left_join(clusdat, by = join_by(state == abbr))
clusdat2
```

```
## # A tibble: 50 x 13
##    state region_9         region_7  region_5 region_4 fips  full  pop_2022 name
##    <chr> <chr>            <chr>     <chr>    <chr>    <chr> <chr>    <dbl> <chr>
##  1 AK    Alaska og Hawaii West      West     West     02    Alas~   733583 Alas~
##  2 AL    South            South     South    South    01    Alab~  5074296 Alab~
##  3 AR    South            South     South    South    05    Arka~  3045637 Arka~
##  4 AZ    Southwest        West      West     West     04    Ariz~  7359197 Ariz~
##  5 CA    Pacific Coast    West      West     West     06    Cali~ 39029342 Cali~
##  6 CO    Rocky Mountains  West      West     West     08    Colo~  5839926 Colo~
##  7 CT    New England      Northeast Northea~ Northea~ 09    Conn~  3626205 Conn~
##  8 DE    Mid-Atlantic     South     South    South    10    Dela~  1018396 Dela~
##  9 FL    South            South     South    South    12    Flor~ 22244823 Flor~
## 10 GA    South            South     South    South    13    Geor~ 10912876 Geor~
## # i 40 more rows
## # i 4 more variables: cluster_4 <fct>, cluster_5 <fct>, cluster_7 <fct>,
## #   cluster_9 <fct>
```

```r
clusdat2 |> head()
```

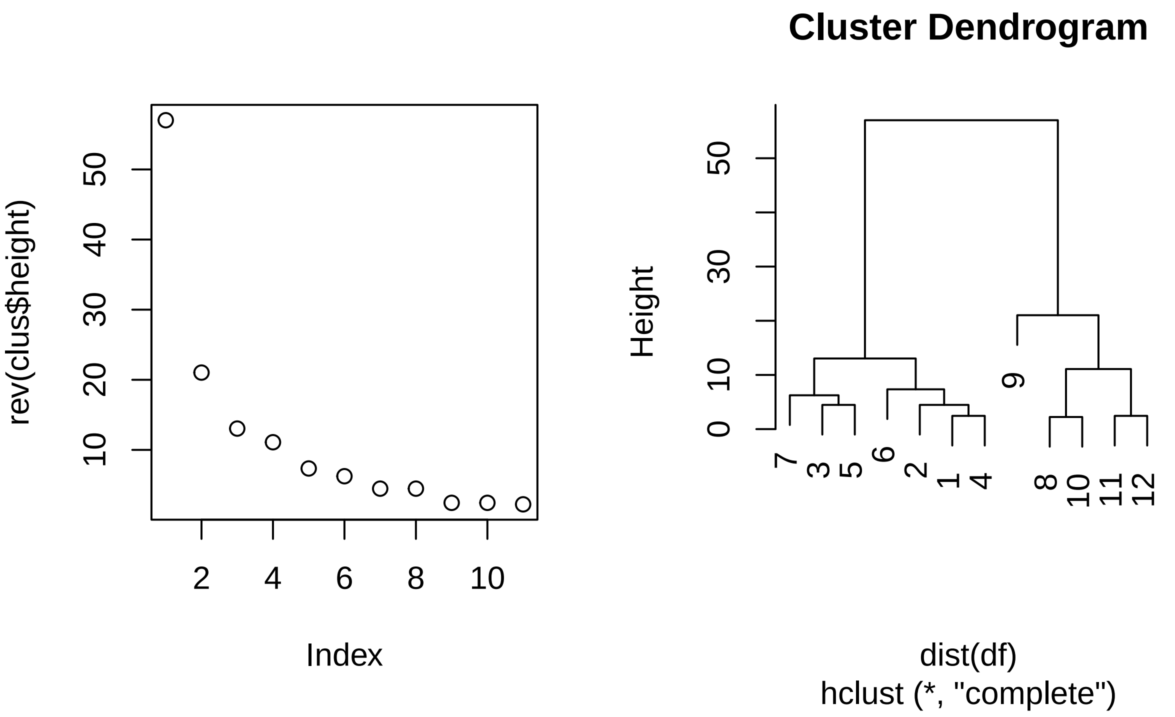
```
## # A tibble: 6 x 13
##   state region_9 region_7 region_5 region_4 fips  full  pop_2022 name  cluster_4
##   <chr> <chr>    <chr>    <chr>    <chr>    <chr> <chr>    <dbl> <chr> <fct>
## 1 AK    Alaska ~ West     West     West     02    Alas~   733583 Alas~ 2
## 2 AL    South    South    South    South    01    Alab~  5074296 Alab~ 1
## 3 AR    South    South    South    South    05    Arka~  3045637 Arka~ 1
## 4 AZ    Southwe~ West     West     West     04    Ariz~  7359197 Ariz~ 3
## 5 CA    Pacific~ West     West     West     06    Cali~ 39029342 Cali~ 3
## 6 CO    Rocky M~ West     West     West     08    Colo~  5839926 Colo~ 2
## # i 3 more variables: cluster_5 <fct>, cluster_7 <fct>, cluster_9 <fct>
```

```r
## Or the old fashioned way:
## clusdat2 <- merge(states_df, clusdat, by.x="state", by.y="abbr") |> head()

pl1 <- plot_usmap(data=clusdat2, values="cluster_4") ## + theme(legend.position = "none")
pl2 <- plot_usmap(data=clusdat2, values="region_5")

library(patchwork)
(pl1 + pl2)
```



5 Details: What is calculated - optional*

```
df <- data.frame(x1 = c(26, 28, 19, 27, 23, 31, 22, 1, 2, 1, 1, 1),
                 x2 = c(5, 5, 7, 5, 7, 4, 2, 0, 0, 0, 0, 1),
                 x3 = c(8, 6, 5, 7, 5, 9, 5, 1, 0, 1, 0, 1),
                 x4 = c(8, 5, 3, 8, 1, 3, 4, 0, 0, 1, 0, 0),
                 x5 = c(1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0),
                 x6 = c(2, 3, 1, 0, 1, 1, 3, 37, 49, 39, 28, 30))
clus <- hclust(dist(df))
par(mfrow=c(1,2))
plot(rev(clus$height))
plot(clus)
```



```
clus$merge  |> head(4)
```

```
##      [,1] [,2]
## [1,]   -8  -10
## [2,]   -1   -4
## [3,]  -11  -12
## [4,]   -2    2
```

```
clus$height
```

```
##  [1]  2.24  2.45  2.45  4.47  4.47  6.24  7.35 11.09 13.04 21.02 57.02
```

```
sum((df[8,]-df[10,])^2) |> sqrt() ## 1. merge: rows 8,10: denoted -8, -10
```

```
## [1] 2.24
```

```
sum((df[1,]-df[ 4,])^2) |> sqrt() ## 2. merge: rows 1,4: denoted -1, -4
```

```
## [1] 2.45
```

```
sum((df[2,]-df[1 ,])^2) |> sqrt() ## 4. merge: row 2, and cluster 2 (rows 1,4)
```

```
## [1] 4.24
```

```
sum((df[2,]-df[4 ,])^2) |> sqrt()
```

```
## [1] 4.47
```

Hence the height is the distance between clusters being merged.

---

6 Turning things around - clustering variables

Above we have clustered states based on crime data: states (rows in the dataframe) that are similar in crime rates are clustered together.

But one can also cluster variables (columns in the dataframe): variables that are similar in their relation to the states are clustered together.

All we have to do is to transpose the data frame:

```
hc <- hclust(dist(t(crime2)))
cutree(hc, k=2:6)
```

```
##           2 3 4 5 6
## murder    1 1 1 1 1
## rape      1 1 1 2 2
## assault   1 1 1 2 3
## robbery   2 2 2 3 4
## burglary  2 3 3 4 5
## larceny   2 3 3 4 5
## autotheft 2 2 4 5 6
```