Hierarchical Clustering

2023-06-14

Data Set-up

```
gapminder <- as_tibble(gapminder)
head(gapminder)
```

```
## # A tibble: 6 x 9
##   country    year infant_mortality life_expectancy fertility population      gdp
##   <fct>     <int>            <dbl>           <dbl>     <dbl>      <dbl>    <dbl>
## 1 Albania    1960            115.             62.9      6.19    1636054 NA
## 2 Algeria    1960            148.             47.5      7.65   11124892  1.38e10
## 3 Angola     1960            208              36.0      7.32    5270844 NA
## 4 Antigua ~  1960             NA              63.0      4.43      54681 NA
## 5 Argentina  1960             59.9            65.4      3.11   20619075  1.08e11
## 6 Armenia    1960             NA              66.9      4.55    1867396 NA
## # i 2 more variables: continent <fct>, region <fct>
```

```
clean_gapminder <- gapminder %>%  filter(year == 2011, !is.na(gdp)) %>%  mutate(log_gdp = log(gdp))
clean_gapminder
```

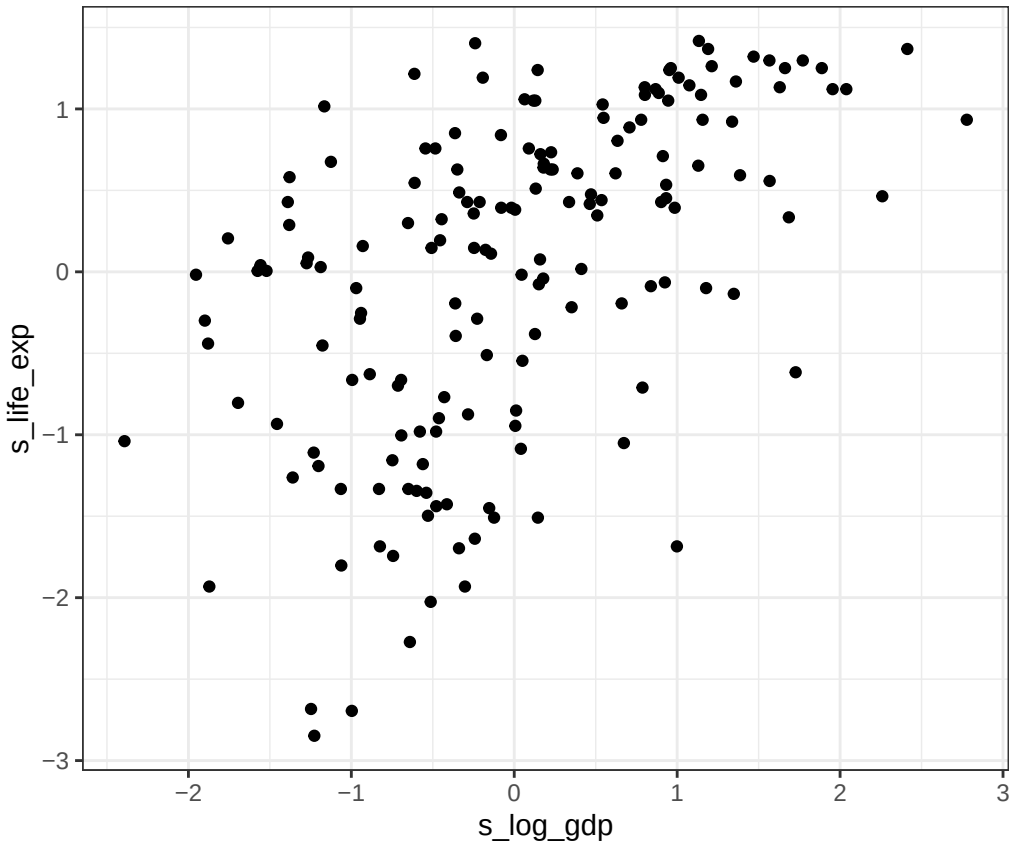
```
## # A tibble: 168 x 10
##    country    year infant_mortality life_expectancy fertility population     gdp
##    <fct>     <int>            <dbl>           <dbl>     <dbl>      <dbl>   <dbl>
##  1 Albania    2011             14.3            77.4      1.75    2886010 6.32e 9
##  2 Algeria    2011             22.8            76.1      2.83   36717132 8.11e10
##  3 Angola     2011            107.             58.1      6.1    21942296 2.70e10
##  4 Antigua ~  2011              7.2            75.9      2.12      88152 8.02e 8
##  5 Argentina  2011             12.7            76        2.2    41655616 4.73e11
##  6 Armenia    2011             15.3            73.5      1.5     2967984 4.29e 9
##  7 Australia  2011              3.8            82.2      1.88   22542371 5.73e11
##  8 Austria    2011              3.4            80.7      1.44    8423559 2.31e11
##  9 Azerbaij~  2011             32.5            70.8      1.96    9227512 2.14e10
## 10 Bahamas    2011             11.1            72.6      1.9      366711 6.76e 9
## # i 158 more rows
## # i 3 more variables: continent <fct>, region <fct>, log_gdp <dbl>
```

Standardization

```
clean_gapminder <- clean_gapminder %>%
  mutate(s_log_gdp = as.numeric(scale(log_gdp, center = TRUE, scale = TRUE)), s_life_exp = as.numeric(s

clean_gapminder %>%
  ggplot(aes(x = s_log_gdp, y = s_life_exp))+
  geom_point() +
```

```
  theme_bw() +
  coord_fixed()
```



Computing the distance matrix

Pairwise Euclidean Distance:

```
gap_dist <- dist(dplyr::select(clean_gapminder, s_log_gdp, s_life_exp))
```

Crafting the Matrix:

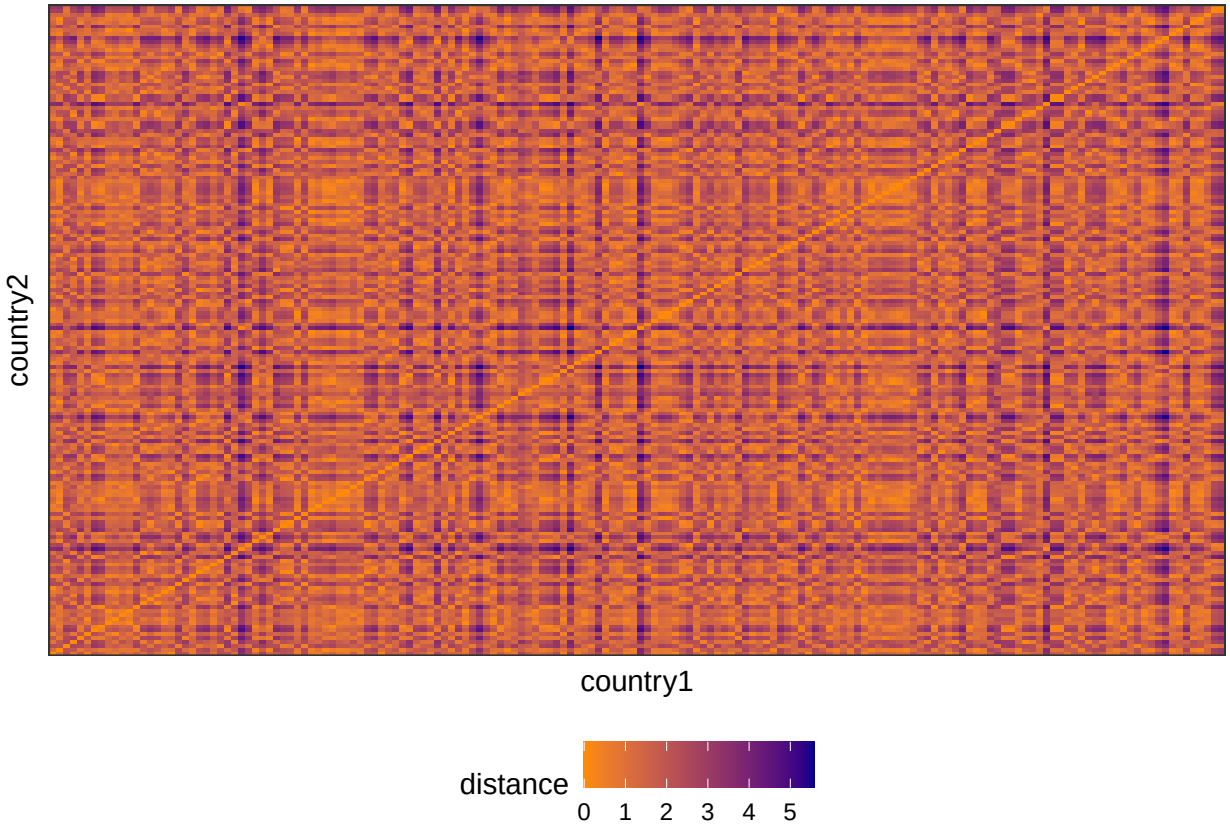
```
gap_dist_matrix <- as.matrix(gap_dist)
rownames(gap_dist_matrix) <- clean_gapminder$country
colnames(gap_dist_matrix) <- clean_gapminder$country
head(gap_dist_matrix[1:3, 1:3])
```

```
##          Albania  Algeria   Angola
## Albania 0.000000 1.116567 2.352044
## Algeria 1.116567 0.000000 2.166692
## Angola  2.352044 2.166692 0.000000
```

Plotting Similarities

```
long_dist_matrix <- as_tibble(gap_dist_matrix) %>%
  mutate(country1 = rownames(gap_dist_matrix)) %>%
  pivot_longer(cols = -country1, names_to = "country2", values_to = "distance")
```

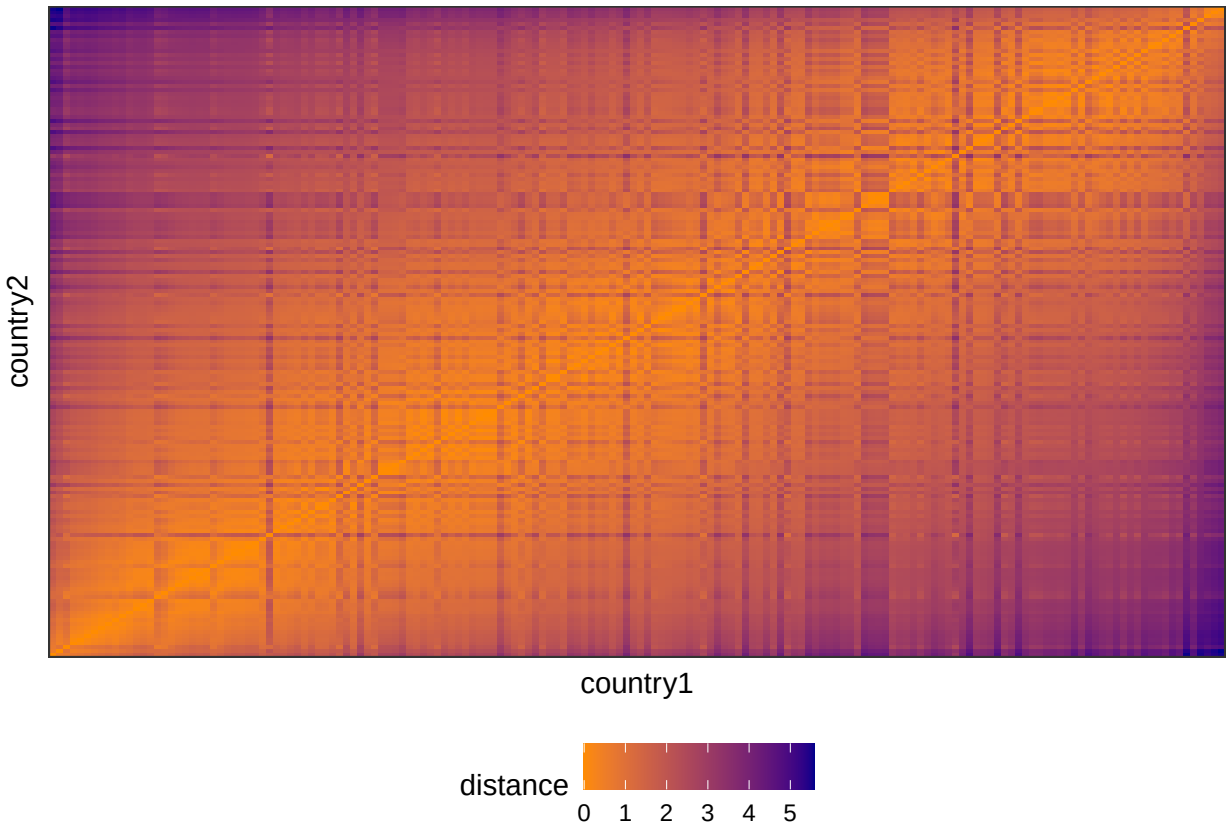
```
long_dist_matrix %>%
  ggplot(aes(x = country1, y = country2, fill = distance)) +
  geom_tile() +
  theme_bw() +
  theme(axis.text = element_blank(), axis.ticks = element_blank(), legend.position = "bottom") +  scale
```




```
gap_dist_seriate <- seriate(gap_dist)

gap_order <- get_order(gap_dist_seriate)

gap_countries_order <-
  as.character(clean_gapminder$country[gap_order])
                                      long_dist_matrix$country1 <- as_factor(long_dist_matrix$country1)
                                      long_dist_matrix$country2 <- as_factor(long_dist_matrix$country2)
                                      long_dist_matrix %>%                                              muta
                                        ggplot(aes(x = country1, y = country2,                fill = dis
                                        scale_fill_gradient(low = "darkorange", high = "darkblue")
```



Agglomerative Hierarchical Clustering

Pretend all n observations are their own cluster

- Step 1: Compute the pairwise dissimilarities between each cluster (e.g., distance matrix)
- Step 2: Idenitfy the pair of clusters that are least dissimilar
- Step 3: Fuse these two clusters into a new cluster
- Repeat Steps 1 to 3 until all observations are in the same cluster
- **Bottom-up** agglomerative clusters that forms a tree/hierarchy of merging

How do we Define Dissimilarity between Clusters?

We need a linkage function!

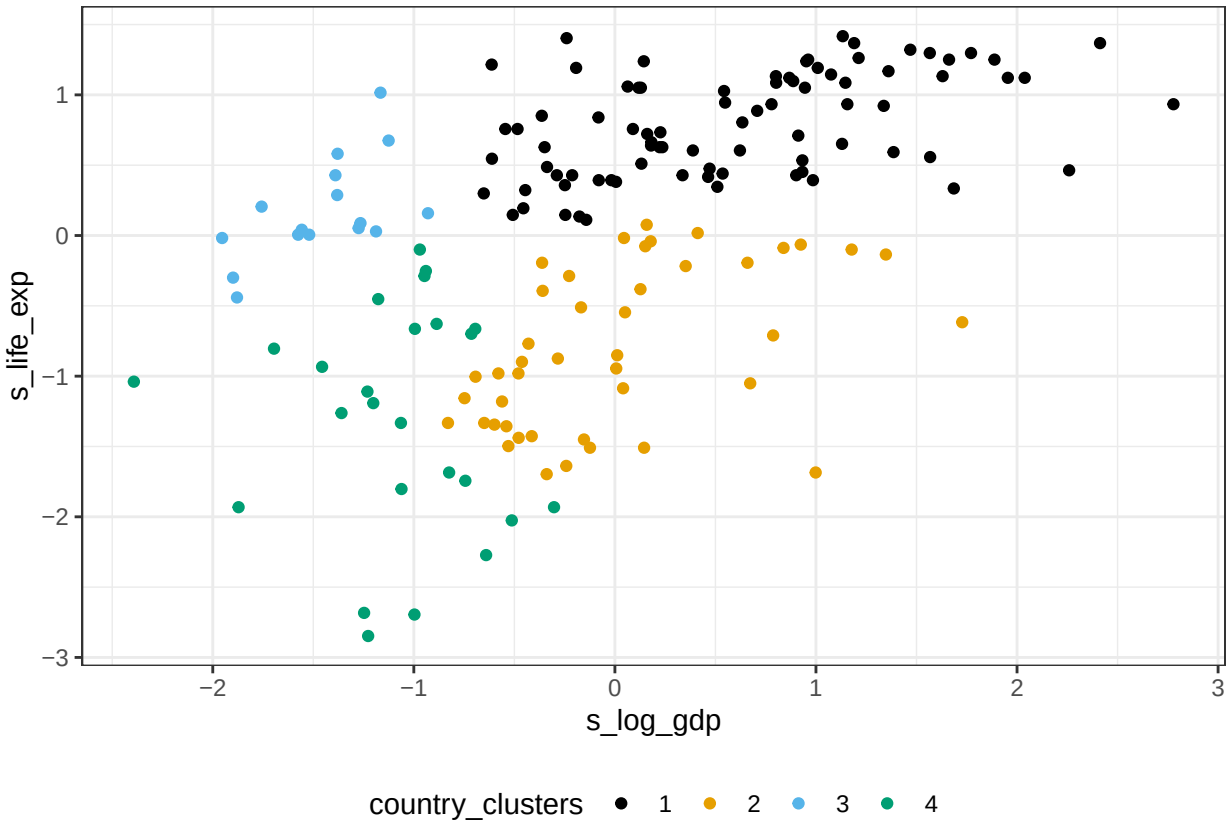
- Complete linkage: uses maximum value of these dissimilarities (i.e., distance)
- Single linkage: uses minimum value of these dissimilarities (i.e., distance)
- Average Linkage: uses average value of these dissimilarities (i.e., distance)

Define dissimilarity between two clusters based on our initial dissimilarity matrix between observations

Complete Linkage Example

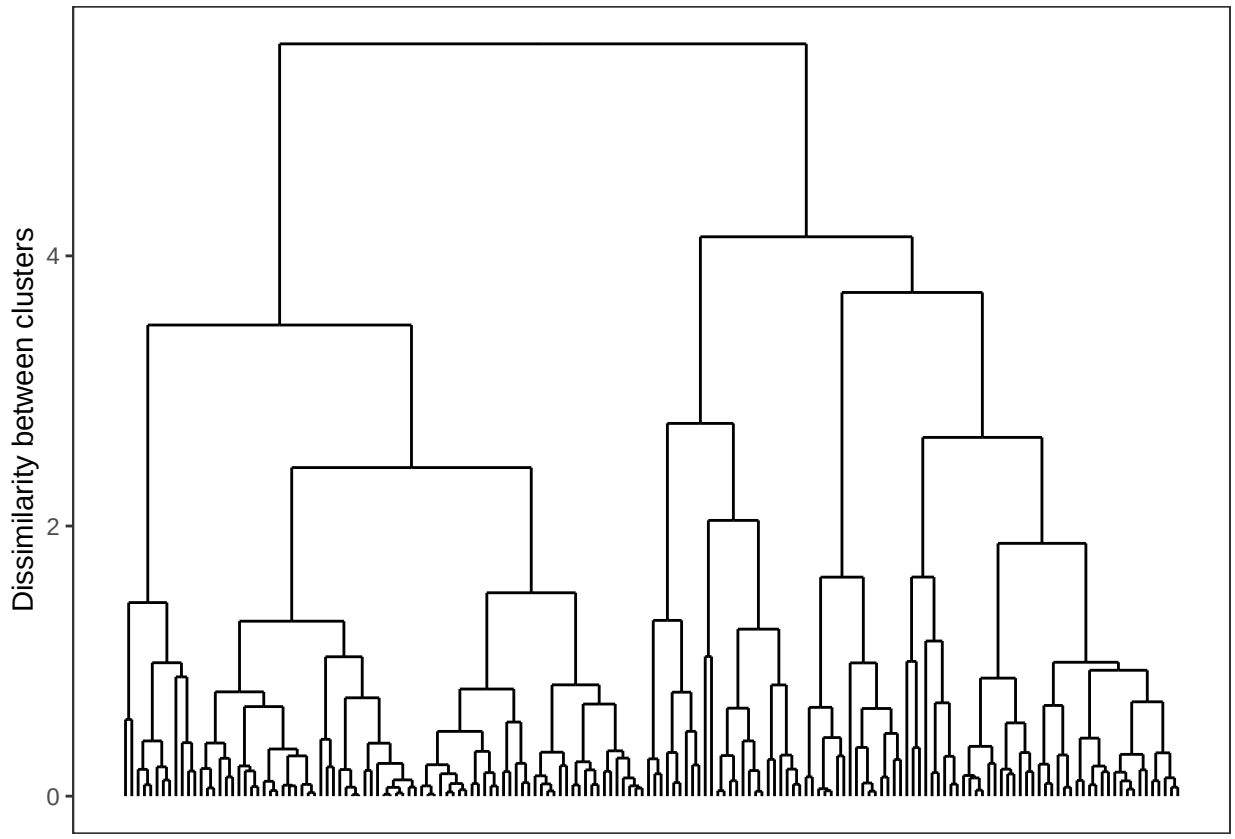
```
gap_complete_hclust <- hclust(gap_dist, method = "complete")
```

```
clean_gapminder %>%  mutate(country_clusters =         as.factor(cutree(gap_complete_hclust, k = 4))) %
  ggplot(aes(x = s_log_gdp, y = s_life_exp,
             color = country_clusters)) +  geom_point() +
  ggthemes::scale_color_colorblind() +
  theme_bw() +
  theme(legend.position = "bottom")
```



Dendrogram

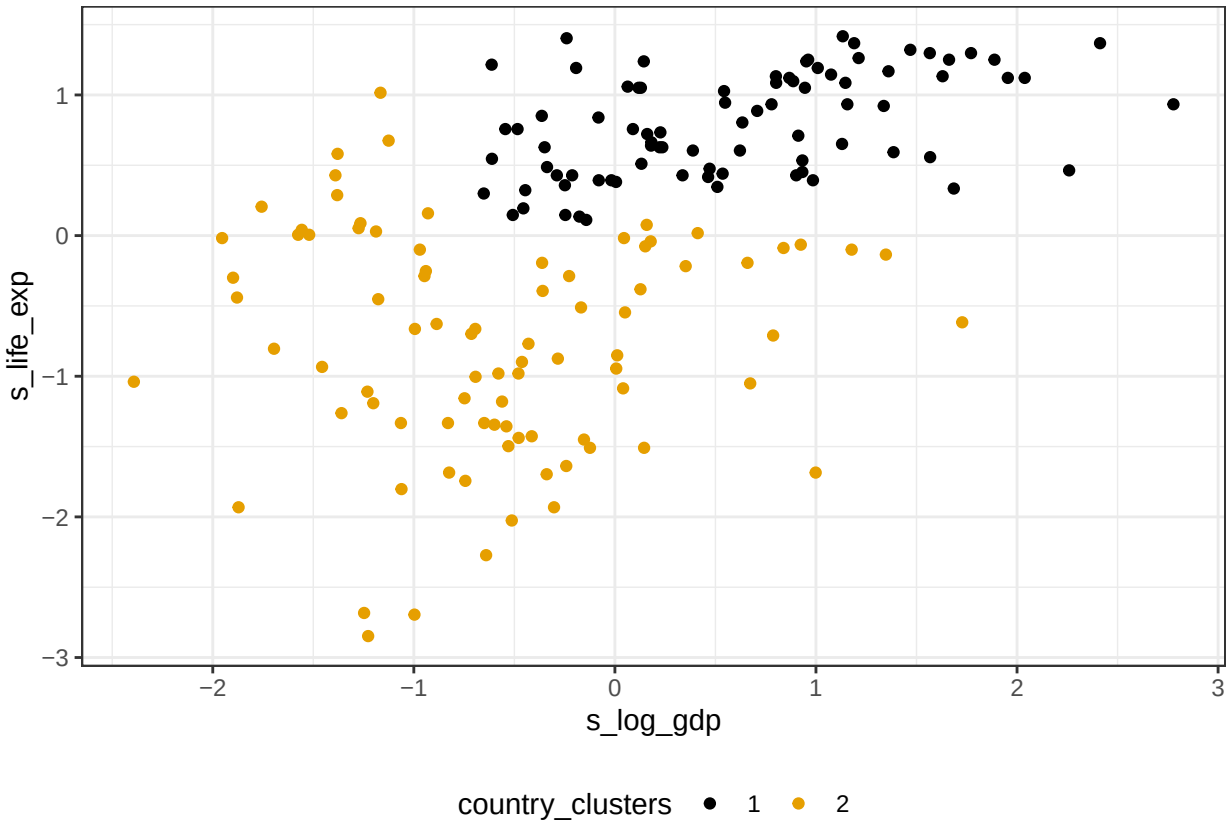
```
ggdendrogram(gap_complete_hclust,              theme_dendro = FALSE, labels = FALSE,          leaf_lab
  labs(y = "Dissimilarity between clusters")  +
  theme_bw() +
  theme(axis.text.x = element_blank(),         axis.title.x = element_blank(), axis.ticks.x = element_bl
```



- Each leaf = one observation
- Height of branch represents the dissimilarity between clusters (Horizontal position along the x-axis means nothing after the first step)

You can specify the height to cut with h (height) instead of k

```
clean_gapminder %>%  mutate(country_clusters =         as.factor(cutree(gap_complete_hclust, h = 5))) %
  ggplot(aes(x = s_log_gdp, y = s_life_exp,
             color = country_clusters)) +  geom_point() +
  ggthemes::scale_color_colorblind() +
  theme_bw() +
  theme(legend.position = "bottom")
```



NOTE: YOU WILL GET DIFFERENT RESULTS BASED ON HOW YOU DEFINE THE LINKAGE FUNCTION

More Linkage Functions

- Centroid Linkage: Computes the dissimilarity between the centroid for cluster 1 and the centroid for cluster 2 (i.e., the distance between the averages of the two clusters)
- Ward's linkage: Merges a pair of clusters to minimize the within-cluster variance (i.e., aim is to minimize the objective function from *K-means*)
- Minimax Linkage

Each cluster is defined **by a prototype** observation (most representative)

Identify the point whose farthest point is closest

Use this minimum-maximum distance as the measure of cluster dissimilarity

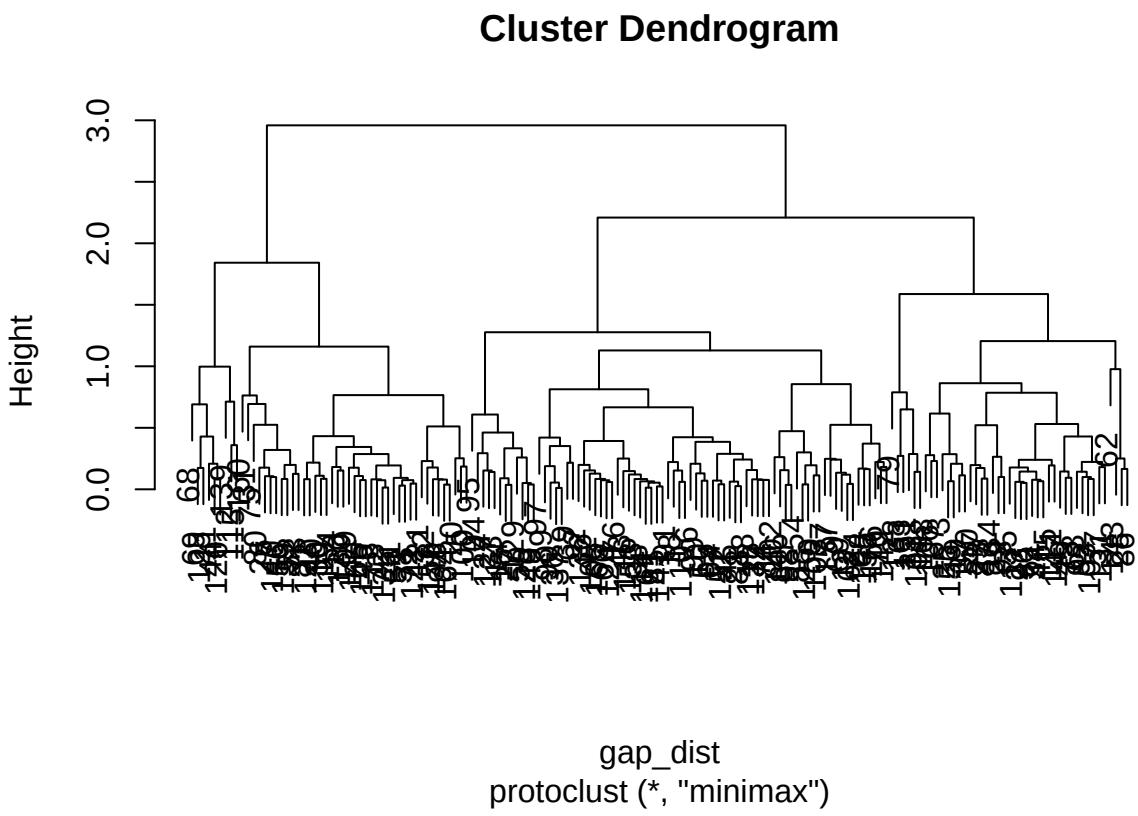
Dendrogram interpretation: each point is less than or equal to h in dissimilarity to the the prototype of the cluster

- **Cluster centers are chosen among the observations themselves – hence the prototype**

Minimax Linkage Example

Dendrogram

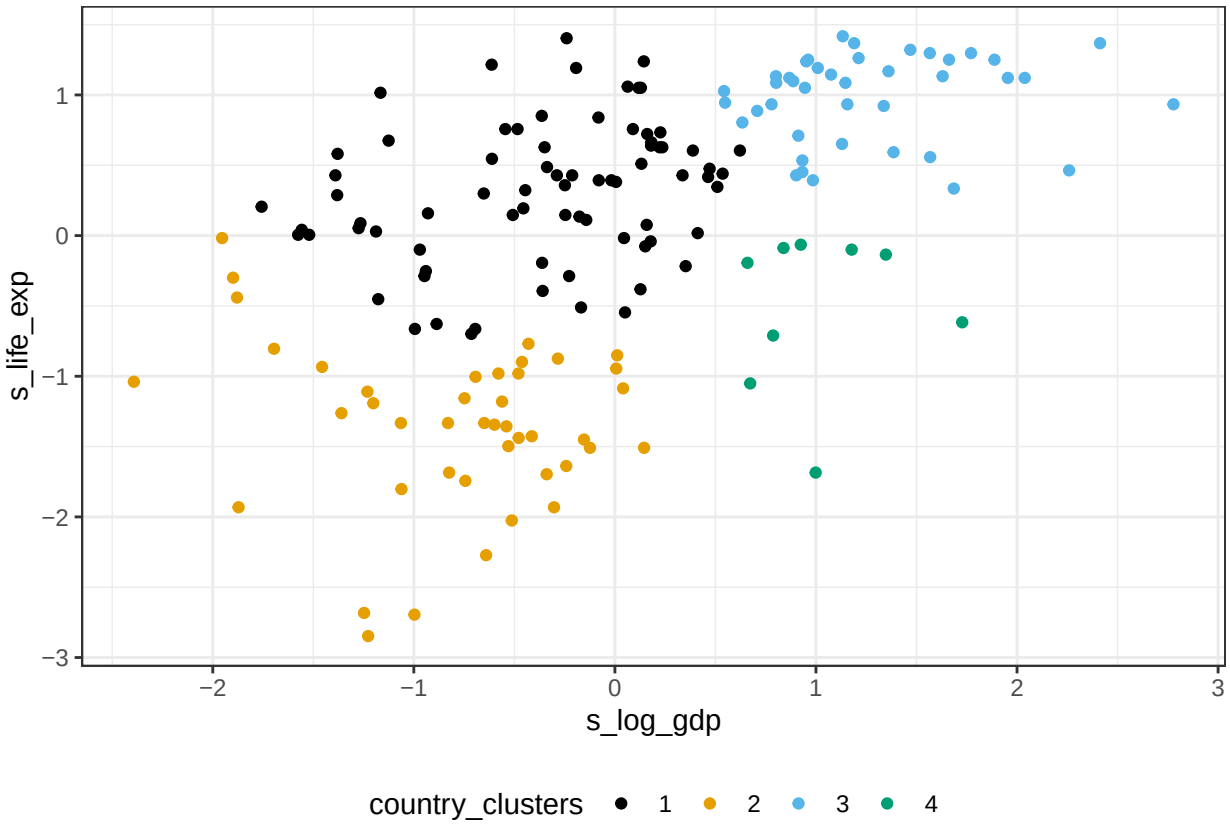
```
gap_minimax <- protoclust(gap_dist)
plot(gap_minimax)
```



Scatterplot

```
minimax_country_clusters <-   protocut(gap_minimax, k = 4)

clean_gapminder %>%
  mutate(country_clusters =     as.factor(minimax_country_clusters$cl)) %>%
  ggplot(aes(x = s_log_gdp, y = s_life_exp,             color = country_clusters)) +  geom_point() +
  theme_bw() +
  theme(legend.position = "bottom")
```



To find prototypes:

```
minimax_country_clusters$protos
```

```
## [1]  91 150  26 115
```

Indices of the prototypes (in the order of the clusters)

Finding countries with these indices:

```
clean_gapminder %>%  dplyr::select(country, gdp, life_expectancy,                population, infant_mo
```

```
## # A tibble: 4 x 5
##   country                gdp life_expectancy population infant_mortality
##   <fct>                <dbl>           <dbl>      <dbl>            <dbl>
## 1 Macedonia, FYR   4713514754            75.6    2065888              7.5
## 2 Togo             1658132200            59.6    6566179             57.9
## 3 Canada         894251850391            81.6   34499905              4.7
## 4 Pakistan       118790417253            64.9  173669648             72.1
```

How are these clusters related to the continents?

```
table("Clusters" = minimax_country_clusters$cl, "Continents" = clean_gapminder$continent)
```

```
##          Continents
## Clusters Africa Americas Asia Europe Oceania
##        1     10       19   24     20       2
##        2     36        1    0      0       6
##        3      0        9   13     18       1
##        4      3        0    5      1       0
```