Task 4 Oblig 3

A)

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggplot2)
library(tidyr)

data <- read.csv("~/Documents/skole/dat320/oblig/3/data/covid.csv")
colnames(data)
```

```
##  [1] "iso_code"
##  [2] "continent"
##  [3] "location"
##  [4] "date"
##  [5] "total_cases"
##  [6] "new_cases"
##  [7] "new_cases_smoothed"
##  [8] "total_deaths"
##  [9] "new_deaths"
## [10] "new_deaths_smoothed"
## [11] "total_cases_per_million"
## [12] "new_cases_per_million"
## [13] "new_cases_smoothed_per_million"
## [14] "total_deaths_per_million"
## [15] "new_deaths_per_million"
## [16] "new_deaths_smoothed_per_million"
## [17] "reproduction_rate"
## [18] "icu_patients"
## [19] "icu_patients_per_million"
## [20] "hosp_patients"
## [21] "hosp_patients_per_million"
## [22] "weekly_icu_admissions"
## [23] "weekly_icu_admissions_per_million"
## [24] "weekly_hosp_admissions"
## [25] "weekly_hosp_admissions_per_million"
## [26] "new_tests"
## [27] "total_tests"
## [28] "total_tests_per_thousand"
## [29] "new_tests_per_thousand"
## [30] "new_tests_smoothed"
## [31] "new_tests_smoothed_per_thousand"
## [32] "positive_rate"
## [33] "tests_per_case"
## [34] "tests_units"
## [35] "total_vaccinations"
## [36] "people_vaccinated"
## [37] "people_fully_vaccinated"
## [38] "total_boosters"
## [39] "new_vaccinations"
## [40] "new_vaccinations_smoothed"
## [41] "total_vaccinations_per_hundred"
## [42] "people_vaccinated_per_hundred"
## [43] "people_fully_vaccinated_per_hundred"
## [44] "total_boosters_per_hundred"
## [45] "new_vaccinations_smoothed_per_million"
## [46] "new_people_vaccinated_smoothed"
## [47] "new_people_vaccinated_smoothed_per_hundred"
## [48] "stringency_index"
## [49] "population"
## [50] "population_density"
## [51] "median_age"
## [52] "aged_65_older"
## [53] "aged_70_older"
## [54] "gdp_per_capita"
## [55] "extreme_poverty"
## [56] "cardiovasc_death_rate"
## [57] "diabetes_prevalence"
## [58] "female_smokers"
## [59] "male_smokers"
## [60] "handwashing_facilities"
## [61] "hospital_beds_per_thousand"
## [62] "life_expectancy"
## [63] "human_development_index"
## [64] "excess_mortality_cumulative_absolute"
## [65] "excess_mortality_cumulative"
## [66] "excess_mortality"
## [67] "excess_mortality_cumulative_per_million"
```

```
unique(data$iso_code)
```

```
##   [1] "AFG"      "OWID_AFR" "ALB"      "DZA"      "AND"      "AGO"
##   [7] "AIA"      "ATG"      "ARG"      "ARM"      "ABW"      "OWID_ASI"
##  [13] "AUS"      "AUT"      "AZE"      "BHS"      "BHR"      "BGD"
##  [19] "BRB"      "BLR"      "BEL"      "BLZ"      "BEN"      "BMU"
##  [25] "BTN"      "BOL"      "BES"      "BIH"      "BWA"      "BRA"     
##  [31] "VGB"      "BRN"      "BGR"      "BFA"      "BDI"      "KHM"     
##  [37] "CMR"      "CAN"      "CPV"      "CYM"      "CAF"      "TCD"     
##  [43] "CHL"      "CHN"      "COL"      "COM"      "COG"      "COK"     
##  [49] "CRI"      "CIV"      "HRV"      "CUB"      "CUW"      "CYP"     
##  [55] "CZE"      "COD"      "DNK"      "DJI"      "DMA"      "DOM"     
##  [61] "ECU"      "EGY"      "SLV"      "GNQ"      "ERI"      "EST"     
##  [67] "SWZ"      "ETH"      "OWID_EUR" "OWID_EUN" "FRO"      "FLK"     
##  [73] "FJI"      "FIN"      "FRA"      "PYF"      "GAB"      "GMB"     
##  [79] "GEO"      "DEU"      "GHA"      "GIB"      "GRC"      "GRL"     
##  [85] "GRD"      "GTM"      "GGY"      "GIN"      "GNB"      "GUY"     
##  [91] "HTI"      "OWID_HIC" "HND"      "HKG"      "HUN"      "ISL"     
##  [97] "IND"      "IDN"      "OWID_INT" "IRN"      "IRQ"      "IRL"     
## [103] "IMN"      "ISR"      "ITA"      "JAM"      "JPN"      "JEY"     
## [109] "JOR"      "KAZ"      "KEN"      "KIR"      "OWID_KOS" "KWT"     
## [115] "KGZ"      "LAO"      "LVA"      "LBN"      "LSO"      "LBR"     
## [121] "LBY"      "LIE"      "LTU"      "OWID_LIC" "OWID_LMC" "LUX"     
## [127] "MAC"      "MDG"      "MWI"      "MYS"      "MDV"      "MLI"     
## [133] "MLT"      "MHL"      "MRT"      "MUS"      "MEX"      "FSM"     
## [139] "MDA"      "MCO"      "MNG"      "MNE"      "MSR"      "MAR"     
## [145] "MOZ"      "MMR"      "NAM"      "NRU"      "NPL"      "NLD"     
## [151] "NCL"      "NZL"      "NIC"      "NER"      "NGA"      "NIU"     
## [157] "OWID_NAM" "MKD"      "OWID_CYN" "NOR"      "OWID_OCE" "OMN"     
## [163] "PAK"      "PLW"      "PSE"      "PAN"      "PNG"      "PRY"     
## [169] "PER"      "PHL"      "PCN"      "POL"      "PRT"      "QAT"     
## [175] "ROU"      "RUS"      "RWA"      "SHN"      "KNA"      "LCA"     
## [181] "SPM"      "VCT"      "WSM"      "SMR"      "STP"      "SAU"     
## [187] "SEN"      "SRB"      "SYC"      "SLE"      "SGP"      "SXM"     
## [193] "SVK"      "SVN"      "SLB"      "SOM"      "ZAF"      "OWID_SAM"
## [199] "KOR"      "SSD"      "ESP"      "LKA"      "SDN"      "SUR"     
## [205] "SWE"      "CHE"      "SYR"      "TWN"      "TJK"      "TZA"     
## [211] "THA"      "TLS"      "TGO"      "TKL"      "TON"      "TTO"     
## [217] "TUN"      "TUR"      "TKM"      "TCA"      "TUV"      "UGA"     
## [223] "UKR"      "ARE"      "GBR"      "USA"      "OWID_UMC" "URY"     
## [229] "UZB"      "VUT"      "VAT"      "VEN"      "VNM"      "WLF"     
## [235] "OWID_WRL" "YEM"      "ZMB"      "ZWE"
```

```
head(data)
```

```
##   iso_code continent    location       date total_cases new_cases
## 1      AFG      Asia Afghanistan 2020-02-24           5         5
## 2      AFG      Asia Afghanistan 2020-02-25           5         0
## 3      AFG      Asia Afghanistan 2020-02-26           5         0
## 4      AFG      Asia Afghanistan 2020-02-27           5         0
## 5      AFG      Asia Afghanistan 2020-02-28           5         0
## 6      AFG      Asia Afghanistan 2020-02-29           5         0
##   new_cases_smoothed total_deaths new_deaths new_deaths_smoothed
## 1                 NA           NA         NA                  NA
## 2                 NA           NA         NA                  NA
## 3                 NA           NA         NA                  NA
## 4                 NA           NA         NA                  NA
## 5                 NA           NA         NA                  NA
## 6              0.714           NA         NA                  NA
##   total_cases_per_million new_cases_per_million new_cases_smoothed_per_million
## 1                   0.126                 0.126                             NA
## 2                   0.126                 0.000                             NA
## 3                   0.126                 0.000                             NA
## 4                   0.126                 0.000                             NA
## 5                   0.126                 0.000                             NA
## 6                   0.126                 0.000                          0.018
##   total_deaths_per_million new_deaths_per_million
## 1                       NA                     NA
## 2                       NA                     NA
## 3                       NA                     NA
## 4                       NA                     NA
## 5                       NA                     NA
## 6                       NA                     NA
##   new_deaths_smoothed_per_million reproduction_rate icu_patients
## 1                              NA                NA           NA
## 2                              NA                NA           NA
## 3                              NA                NA           NA
## 4                              NA                NA           NA
## 5                              NA                NA           NA
## 6                              NA                NA           NA
##   icu_patients_per_million hosp_patients hosp_patients_per_million
## 1                       NA            NA                        NA
## 2                       NA            NA                        NA
## 3                       NA            NA                        NA
## 4                       NA            NA                        NA
## 5                       NA            NA                        NA
## 6                       NA            NA                        NA
##   weekly_icu_admissions weekly_icu_admissions_per_million
## 1                    NA                                NA
## 2                    NA                                NA
## 3                    NA                                NA
## 4                    NA                                NA
## 5                    NA                                NA
## 6                    NA                                NA
##   weekly_hosp_admissions weekly_hosp_admissions_per_million new_tests
## 1                     NA                                 NA        NA
## 2                     NA                                 NA        NA
## 3                     NA                                 NA        NA
## 4                     NA                                 NA        NA
## 5                     NA                                 NA        NA
## 6                     NA                                 NA        NA
##   total_tests total_tests_per_thousand new_tests_per_thousand
## 1          NA                       NA                     NA
## 2          NA                       NA                     NA
## 3          NA                       NA                     NA
## 4          NA                       NA                     NA
## 5          NA                       NA                     NA
## 6          NA                       NA                     NA
##   new_tests_smoothed new_tests_smoothed_per_thousand positive_rate
## 1                 NA                              NA            NA
## 2                 NA                              NA            NA
## 3                 NA                              NA            NA
## 4                 NA                              NA            NA
## 5                 NA                              NA            NA
## 6                 NA                              NA            NA
##   tests_per_case tests_units total_vaccinations people_vaccinated
## 1             NA                             NA                NA
## 2             NA                             NA                NA
## 3             NA                             NA                NA
## 4             NA                             NA                NA
## 5             NA                             NA                NA
## 6             NA                             NA                NA
##   people_fully_vaccinated total_boosters new_vaccinations
## 1                      NA             NA               NA
## 2                      NA             NA               NA
## 3                      NA             NA               NA
## 4                      NA             NA               NA
## 5                      NA             NA               NA
## 6                      NA             NA               NA
##   new_vaccinations_smoothed total_vaccinations_per_hundred
## 1                        NA                             NA
## 2                        NA                             NA
## 3                        NA                             NA
## 4                        NA                             NA
## 5                        NA                             NA
## 6                        NA                             NA
##   people_vaccinated_per_hundred people_fully_vaccinated_per_hundred
## 1                            NA                                  NA
## 2                            NA                                  NA
## 3                            NA                                  NA
## 4                            NA                                  NA
## 5                            NA                                  NA
## 6                            NA                                  NA
##   total_boosters_per_hundred new_vaccinations_smoothed_per_million
## 1                         NA                                    NA
## 2                         NA                                    NA
## 3                         NA                                    NA
## 4                         NA                                    NA
## 5                         NA                                    NA
## 6                         NA                                    NA
##   new_people_vaccinated_smoothed new_people_vaccinated_smoothed_per_hundred
## 1                             NA                                         NA
## 2                             NA                                         NA
## 3                             NA                                         NA
## 4                             NA                                         NA
## 5                             NA                                         NA
## 6                             NA                                         NA
##   stringency_index population population_density median_age aged_65_older
## 1             8.33   39835428             54.422       18.6         2.581
## 2             8.33   39835428             54.422       18.6         2.581
## 3             8.33   39835428             54.422       18.6         2.581
## 4             8.33   39835428             54.422       18.6         2.581
## 5             8.33   39835428             54.422       18.6         2.581
## 6             8.33   39835428             54.422       18.6         2.581
##   aged_70_older gdp_per_capita extreme_poverty cardiovasc_death_rate
## 1         1.337       1803.987              NA               597.029
## 2         1.337       1803.987              NA               597.029
## 3         1.337       1803.987              NA               597.029
## 4         1.337       1803.987              NA               597.029
## 5         1.337       1803.987              NA               597.029
## 6         1.337       1803.987              NA               597.029
##   diabetes_prevalence female_smokers male_smokers handwashing_facilities
## 1                9.59             NA           NA                 37.746
## 2                9.59             NA           NA                 37.746
## 3                9.59             NA           NA                 37.746
## 4                9.59             NA           NA                 37.746
## 5                9.59             NA           NA                 37.746
## 6                9.59             NA           NA                 37.746
##   hospital_beds_per_thousand life_expectancy human_development_index
## 1                        0.5           64.83                   0.511
## 2                        0.5           64.83                   0.511
## 3                        0.5           64.83                   0.511
## 4                        0.5           64.83                   0.511
## 5                        0.5           64.83                   0.511
## 6                        0.5           64.83                   0.511
##   excess_mortality_cumulative_absolute excess_mortality_cumulative
## 1                                   NA                          NA
## 2                                   NA                          NA
## 3                                   NA                          NA
## 4                                   NA                          NA
## 5                                   NA                          NA
## 6                                   NA                          NA
##   excess_mortality excess_mortality_cumulative_per_million
## 1               NA                                      NA
## 2               NA                                      NA
## 3               NA                                      NA
## 4               NA                                      NA
## 5               NA                                      NA
## 6               NA                                      NA
```

```
row.names(data)[1:15]
```

```
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13" "14" "15"
```

```
df <- data %>%
  mutate(date = ymd(date)) %>%
  subset((date >= ymd("2020-09-01")) & (date <= ymd("2021-02-28"))) %>%
  subset((iso_code %in% c("AUT", "CAN", "CHN", "DEU", "DNK", "GBR", "ITA", "NOR", "NLD", "SWE", "USA"))
  dplyr::select(c("date", "iso_code", "new_cases_smoothed_per_million")) %>%
  rename(cases = new_cases_smoothed_per_million) %>%
  reshape(timevar="iso_code", idvar=c("date"), direction = "wide") %>%
  as.data.frame()
row.names(df) <- NULL
head(df)
```

```
##         date cases.AUT cases.CAN cases.CHN cases.DNK cases.DEU cases.ITA
## 1 2020-09-01    30.584    13.300     0.007    14.843    14.357    21.334
## 2 2020-09-02    30.584    13.742     0.007    16.170    14.304    21.239
## 3 2020-09-03    31.769    14.373     0.009    18.357    14.080    21.211
## 4 2020-09-04    33.791    14.673     0.009    20.814    13.889    21.854
## 5 2020-09-05    33.206    14.827     0.009    23.517    14.001    22.446
## 6 2020-09-06    33.254    15.307     0.009    24.378    14.342    22.283
##   cases.NLD cases.NOR cases.SWE cases.GBR cases.USA
## 1    30.446    10.899    13.259    19.646   125.240
## 2    31.628    13.853     9.955    20.607   124.198
## 3    32.443    15.107    10.841    21.051    122.129
## 4    34.473    16.963    14.440    22.446    122.705
## 5    35.745    17.904    14.440    23.921    122.581
## 6    39.081    19.472    14.440    26.583    121.113
```

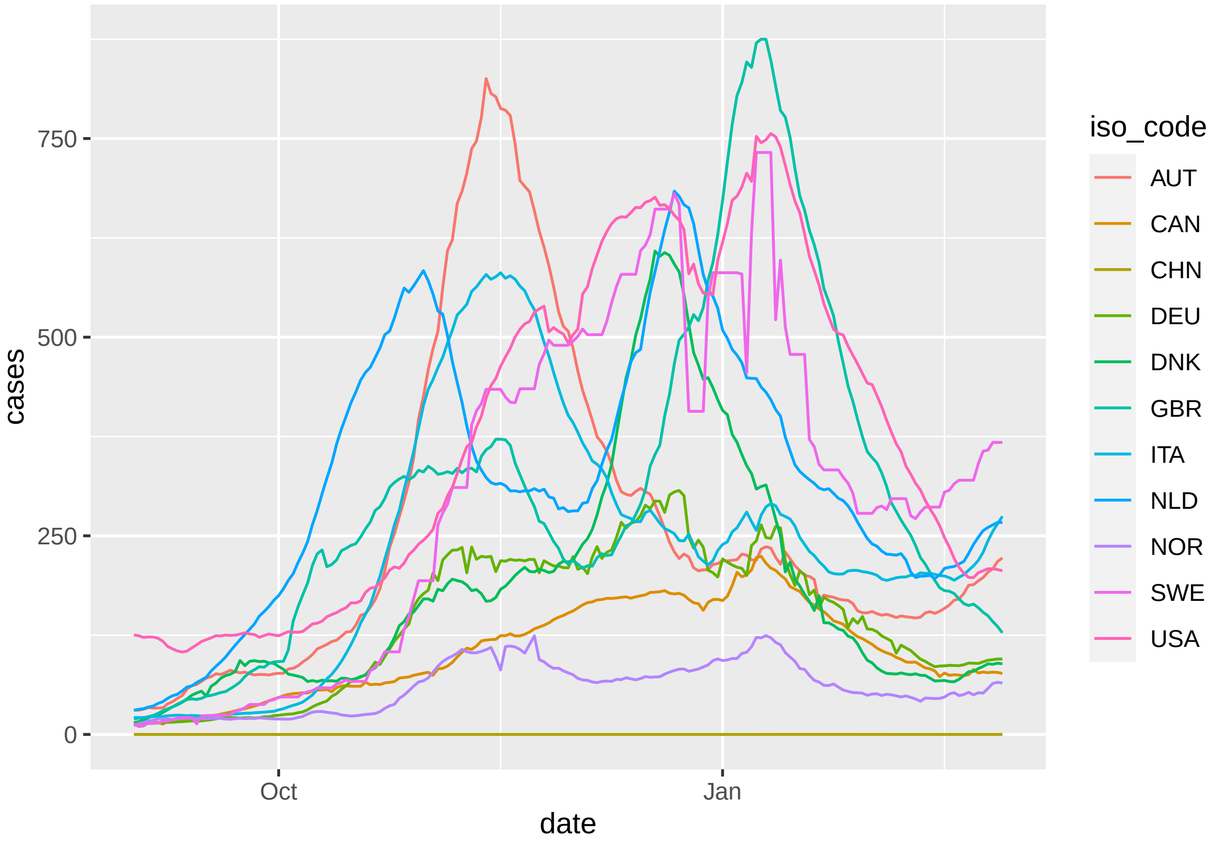
```
colSums(is.na(df))
```

```
##      date cases.AUT cases.CAN cases.CHN cases.DNK cases.DEU cases.ITA cases.NLD
##         0         0         0         0         0         0         0         0
## cases.NOR cases.SWE cases.GBR cases.USA
##         0         0         0         0
```

```
df_long <- df %>% reshape(varying = colnames(df)[-1],
                timevar = "iso_code",
                direction = "long") %>%
  dplyr::select(c("date","iso_code","cases"))
row.names(df_long) <- NULL
head(df_long)
```

```
##         date iso_code  cases
## 1 2020-09-01      AUT 30.584
## 2 2020-09-02      AUT 30.584
## 3 2020-09-03      AUT 31.769
## 4 2020-09-04      AUT 33.791
## 5 2020-09-05      AUT 33.206
## 6 2020-09-06      AUT 33.254
```

```
ggplot(data =df_long, aes(x=date, y=cases, color=iso_code)) +
  geom_line()
```



b):

Compute pair-wise Euclidean distances between the given time series and perform hierarchical clustering. Plot the dendrogram and divide the data into k = 5 clusters. - The task is done in combination with task C).

c):

Repeat task (b) with two other distance measures: Dynamic Time Warping (DTW) and an ARIMA-based distance measure. Compute evaluation metrics and compare the outputs of your models.

- The dun index is a ratio depicting the smalest distance between observations not in the same cluster compared with to the largest intra-cluster disntace (distance between a data item and the cluster centroid). ARIMA model has the best Dunn index. An index in which we are supose to maximize. The Dunn score it self is between [0, inf), so the large are not that large. But a value 2x and 3x the size of the other is quite significant. Read more at: https://cran.r-project.org/web/packages/clValid/vignettes/clValid.pdf
- silhouette width: Silhouette wide is the average of each observation's silhouette value. The silhoute value measures the confindence in the clusters and have the values in the range of [-1, 1]. -1 denote a poor measures, and 1 denotes good measures. Read more at the link above if interested. By evaluating the silhouett values, we can see that both ecludian and Dynamic time warping has negative values. And ARIMA is the only model with strictly posetive values.
- Dendrogram is a figure producsed based on hiarical clustering. THe plots y axis represents distanse and x axis represents labels. The figure is a intuetive tool for deciding the numbers of clusters. We can see that there are some groups ocuring in all of the dendograms. examples: (SWE, USA), (NOR, CHN).
- Plots: When we look at the 3 last plots we can see that ARIMAS plot reprsents the most differentiability between the clusters.

```r
library(TSclust)
```

```
## Loading required package: pdc

## Loading required package: cluster

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(cluster)
library(clValid)
df_dist_eucl <- diss(df[-1], METHOD="EUCL")
df_dist_eucl
```

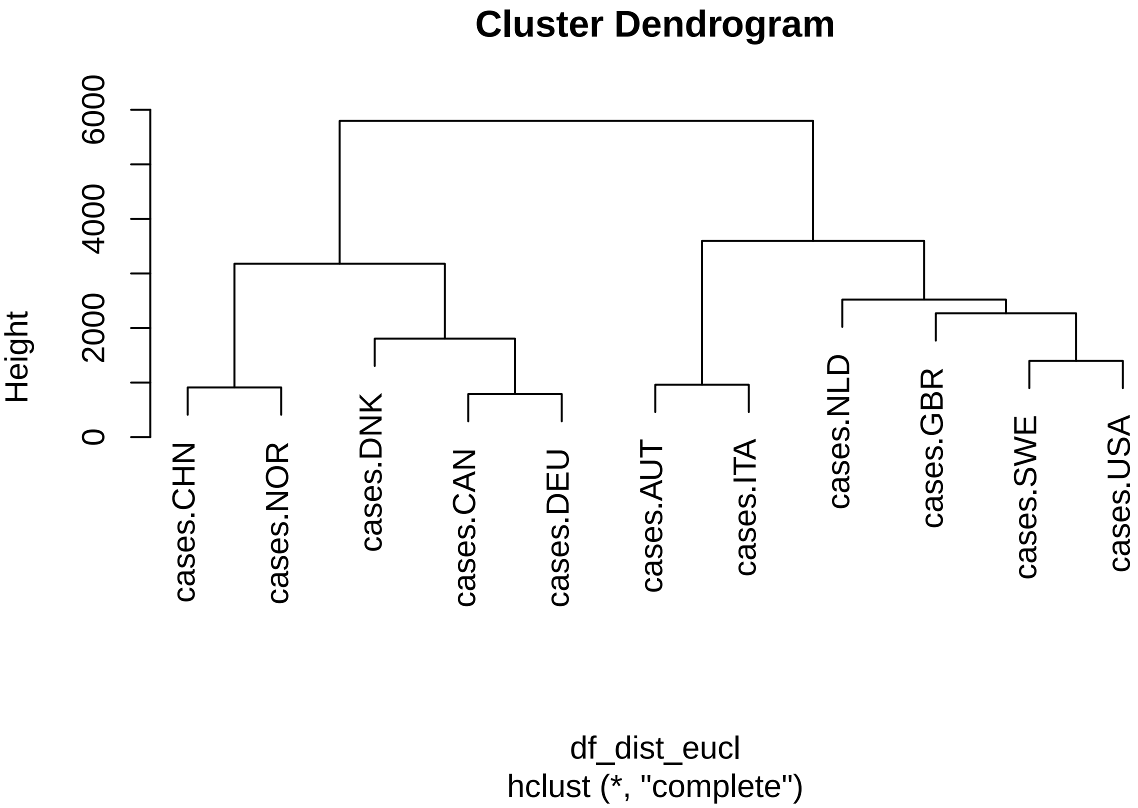
```
##           cases.AUT cases.CAN cases.CHN cases.DNK cases.DEU cases.ITA cases.NLD
## cases.CAN 3284.9025
## cases.CHN 4445.3487 1624.5379
## cases.DNK 3078.2097 1805.1879 3178.2130
## cases.DEU 2676.0893  790.2075 2276.2040 1319.4749
## cases.ITA  960.6933 2573.3943 3864.5320 2455.3857 1940.3283
## cases.NLD 2954.7134 3440.7244 4873.1026 2415.3388 2860.2426 2480.7441
## cases.NOR 3640.3076  794.6686  910.6318 2433.9824 1412.1410 3011.6340 4060.7726
## cases.SWE 2946.6822 3446.4572 5020.2692 2412.4150 2859.4332 2463.9262 2417.3020
## cases.GBR 3597.3342 3551.0318 5047.6405 2976.7883 3149.8610 3012.9991 2278.5309
## cases.USA 3470.4621 4181.8504 5797.1750 3125.8228 3620.8397 3109.5017 2520.9208
##           cases.NOR cases.SWE cases.GBR
## cases.CAN
## cases.CHN
## cases.DNK
## cases.DEU
## cases.ITA
## cases.NLD
## cases.NOR
## cases.SWE 4162.2288
## cases.GBR 4219.7811 2270.0323
## cases.USA 4934.8158 1398.1643 2061.9316
```

```r
df_dist_DRWARP <- diss(df[-1], METHOD="DTWARP")
df_dist_DRWARP
```

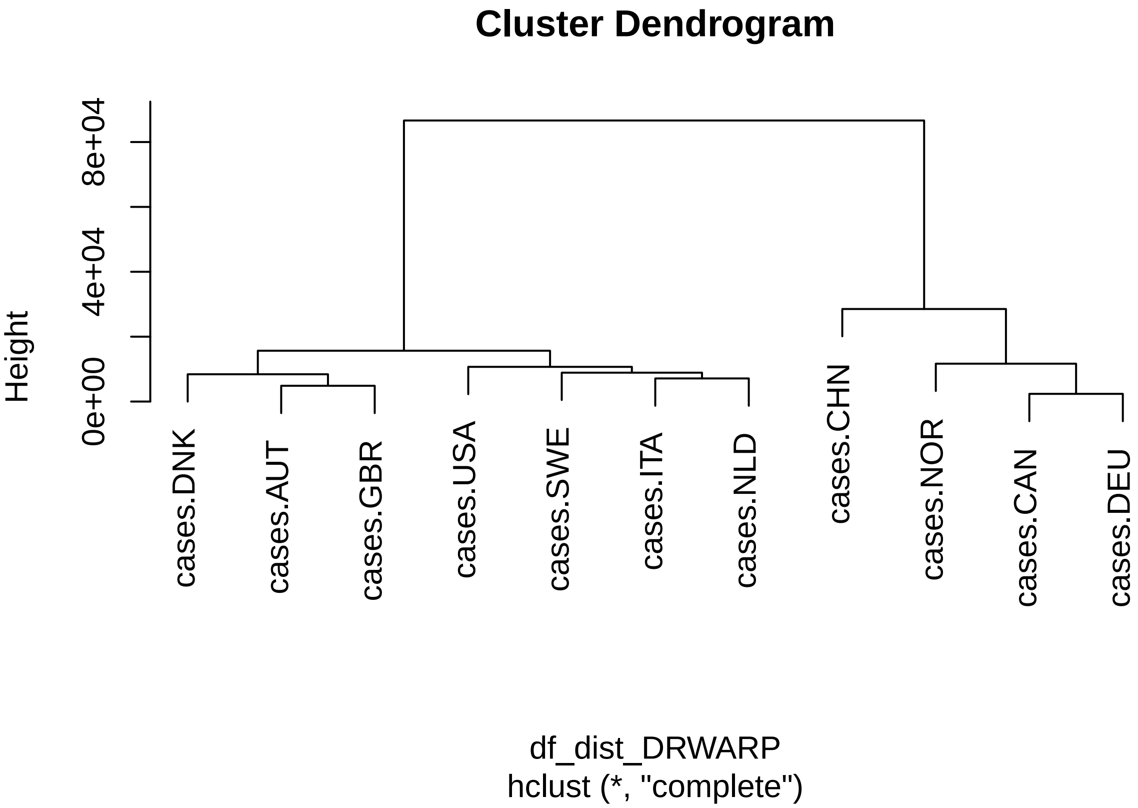
```
##           cases.AUT cases.CAN cases.CHN cases.DNK cases.DEU cases.ITA cases.NLD
## cases.CAN 20685.232
## cases.CHN 52631.942 21549.898
## cases.DNK  7539.645  8983.853 35854.158
## cases.DEU 16209.663  2371.601 28527.286  6692.187
## cases.ITA  5831.485 18155.044 46960.246  7435.800 12832.737
## cases.NLD  8481.387 29080.503 64101.507 12816.873 19978.340  7126.440
## cases.NOR 33342.064  4940.556 12822.844 16466.554 11674.232 32689.136 46247.605
## cases.SWE 11768.166 36294.684 57432.000 15670.554 28701.315  7547.324  8898.624
## cases.GBR  4862.072 24341.507 59605.082  8409.061 17915.319  9044.310  9214.883
## cases.USA  5914.308 41921.468 86633.284 11212.148 33164.639  9381.444  9145.719
##           cases.NOR cases.SWE cases.GBR
## cases.CAN
## cases.CHN
## cases.DNK
## cases.DEU
## cases.ITA
## cases.NLD
## cases.NOR
## cases.SWE 47687.285
## cases.GBR 40058.207 11634.091
## cases.USA 59737.937 10696.951  8688.950
```

```r
df_dist_PIC <- diss(df[-1], METHOD="AR.PIC")
#df_dist_PIC
```

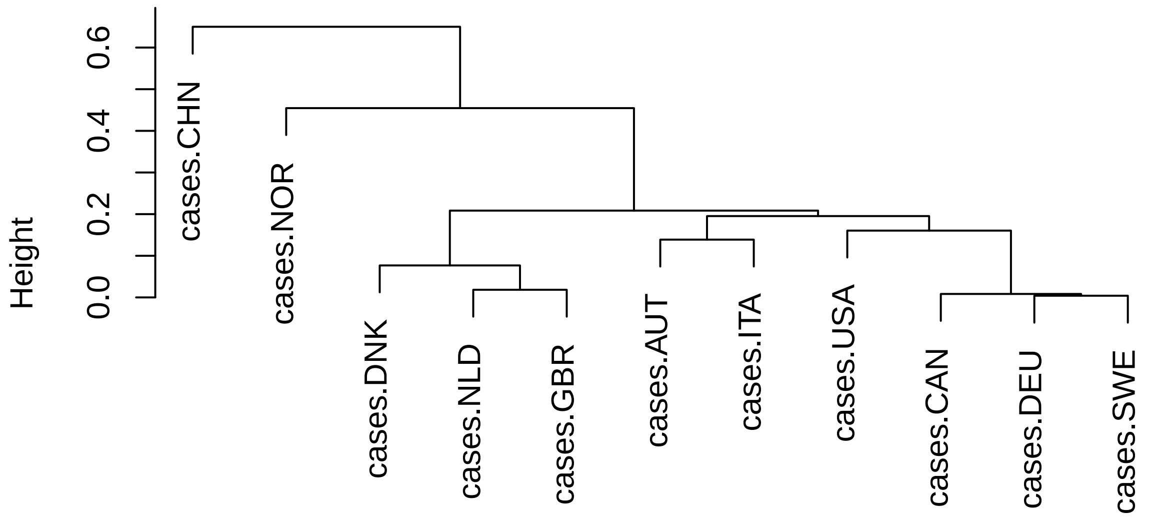
```r
hcl_eucl <- hclust(df_dist_eucl, method="complete")
plot(hcl_eucl)
```



```
hcl_DRWARP <- hclust(df_dist_DRWARP, method="complete")
plot(hcl_DRWARP)
```



```
hcl_PIC <- hclust(df_dist_PIC, method="complete")
plot(hcl_PIC)
```



Cluster Dendrogram

df_dist_PIC
hclust (*, "complete")

```
cl_eucl <- cutree(hcl_eucl, k=5)
cl_eucl
```

```
## cases.AUT cases.CAN cases.CHN cases.DNK cases.DEU cases.ITA cases.NLD cases.NOR
##         1         2         3         2         2         1         4         3
## cases.SWE cases.GBR cases.USA
##         5         5         5
```

```
cl_DRWARP <- cutree(hcl_DRWARP, k=5)
cl_DRWARP
```

```
## cases.AUT cases.CAN cases.CHN cases.DNK cases.DEU cases.ITA cases.NLD cases.NOR
##         1         2         3         1         2         4         4         5
## cases.SWE cases.GBR cases.USA
##         4         1         4
```

```
cl_PIC <- cutree(hcl_PIC, k=5)
cl_PIC
```

```
## cases.AUT cases.CAN cases.CHN cases.DNK cases.DEU cases.ITA cases.NLD cases.NOR
##         1         2         3         4         2         1         4         5
## cases.SWE cases.GBR cases.USA
##         2         4         2
```

```
library(TSclust)
library(cluster)
#install.packages("clValid")
```

```
library(clValid)
```

```
clValid::dunn(distance = df_dist_eucl, clusters = cl_eucl)
```

```
## [1] 0.3500693
```

```
clValid::dunn(distance = df_dist_DRWARP, clusters = cl_DRWARP)
```

```
## [1] 0.4618658
```

```
clValid::dunn(distance = df_dist_PIC, clusters = cl_PIC)
```

```
## [1] 0.9269809
```

```
cluster::silhouette(x = cl_eucl, dist=df_dist_eucl)
```

```
##       cluster neighbor   sil_width
##  [1,]       1        4  0.67486075
##  [2,]       2        3 -0.06788519
##  [3,]       3        2  0.61408211
##  [4,]       2        4  0.35316263
##  [5,]       2        3  0.42801379
##  [6,]       1        2  0.58644926
##  [7,]       4        5  0.00000000
##  [8,]       3        2  0.41132993
##  [9,]       5        4  0.24126224
## [10,]       5        4  0.04939539
## [11,]       5        4  0.31372379
## attr(,"Ordered")
## [1] FALSE
## attr(,"call")
## silhouette.default(x = cl_eucl, dist = df_dist_eucl)
## attr(,"class")
## [1] "silhouette"
```

```
cluster::silhouette(x = cl_DRWARP, dist=df_dist_DRWARP)
```

```
##       cluster neighbor   sil_width
##  [1,]       1        4  0.22477994
##  [2,]       2        5  0.51997285
##  [3,]       3        5  0.00000000
##  [4,]       1        2 -0.01709643
##  [5,]       2        5  0.79685165
##  [6,]       4        1 -0.07248380
##  [7,]       4        1  0.17508390
##  [8,]       5        2  0.00000000
##  [9,]       4        1  0.30532515
## [10,]       1        4  0.31205990
## [11,]       4        1 -0.11664025
## attr(,"Ordered")
## [1] FALSE
## attr(,"call")
## silhouette.default(x = cl_DRWARP, dist = df_dist_DRWARP)
## attr(,"class")
## [1] "silhouette"
```

```
cluster::silhouette(x = cl_PIC, dist=df_dist_PIC)
```

```
##       cluster neighbor  sil_width
##  [1,]       1        2 0.27356602
##  [2,]       2        1 0.70126366
##  [3,]       3        5 0.00000000
##  [4,]       4        2 0.60397614
##  [5,]       2        1 0.70901385
##  [6,]       1        4 0.21216165
##  [7,]       4        1 0.73792790
##  [8,]       5        1 0.00000000
##  [9,]       2        1 0.70207526
## [10,]       4        1 0.74111407
## [11,]       2        4 0.05427411
## attr(,"Ordered")
## [1] FALSE
## attr(,"call")
## silhouette.default(x = cl_PIC, dist = df_dist_PIC)
## attr(,"class")
## [1] "silhouette"
```

```
cl_eucl <- as.factor(cl_eucl)
cl_DRWARP <- as.factor(cl_DRWARP)
cl_PIC <- as.factor(cl_PIC)
```

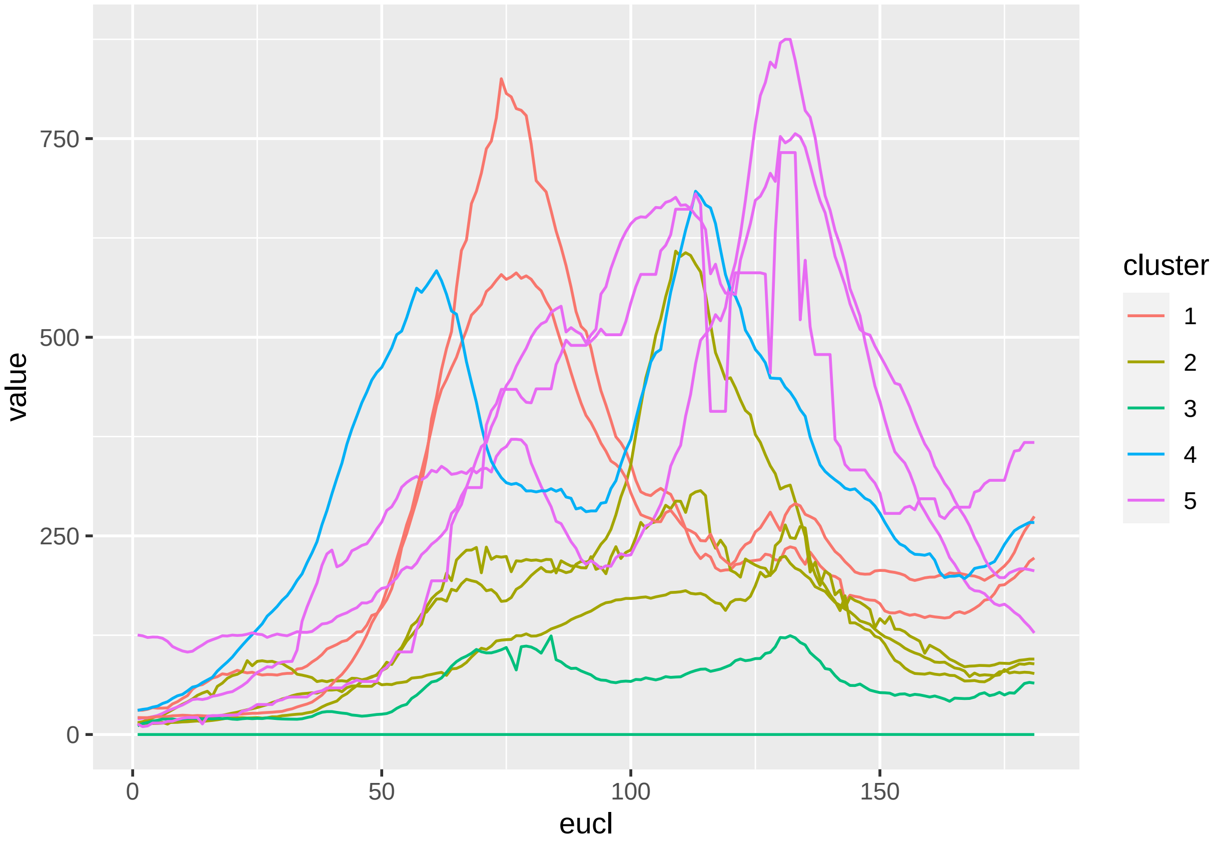
```
eucl <- as.data.frame( t(df[,-1])) %>%
  mutate(cluster = cl_eucl, country=colnames(df)[-1]) %>%
  pivot_longer(cols = 1:181)
eucl$name <- rep(1:181, 11)

dewarp <- as.data.frame( t(df[,-1])) %>%
  mutate(cluster = cl_DRWARP, country=colnames(df)[-1]) %>%
  pivot_longer(cols = 1:181)
dewarp$name <- rep(1:181, 11)

pic <- as.data.frame( t(df[,-1])) %>%
  mutate(cluster = cl_PIC, country=colnames(df)[-1]) %>%
  pivot_longer(cols = 1:181)
pic$name <- rep(1:181, 11)

head(eucl)
```

```
## # A tibble: 6 x 4
##   cluster country    name value
##   <fct>   <chr>     <int> <dbl>
## 1 1       cases.AUT     1  30.6
## 2 1       cases.AUT     2  30.6
## 3 1       cases.AUT     3  31.8
## 4 1       cases.AUT     4  33.8
## 5 1       cases.AUT     5  33.2
## 6 1       cases.AUT     6  33.3
```

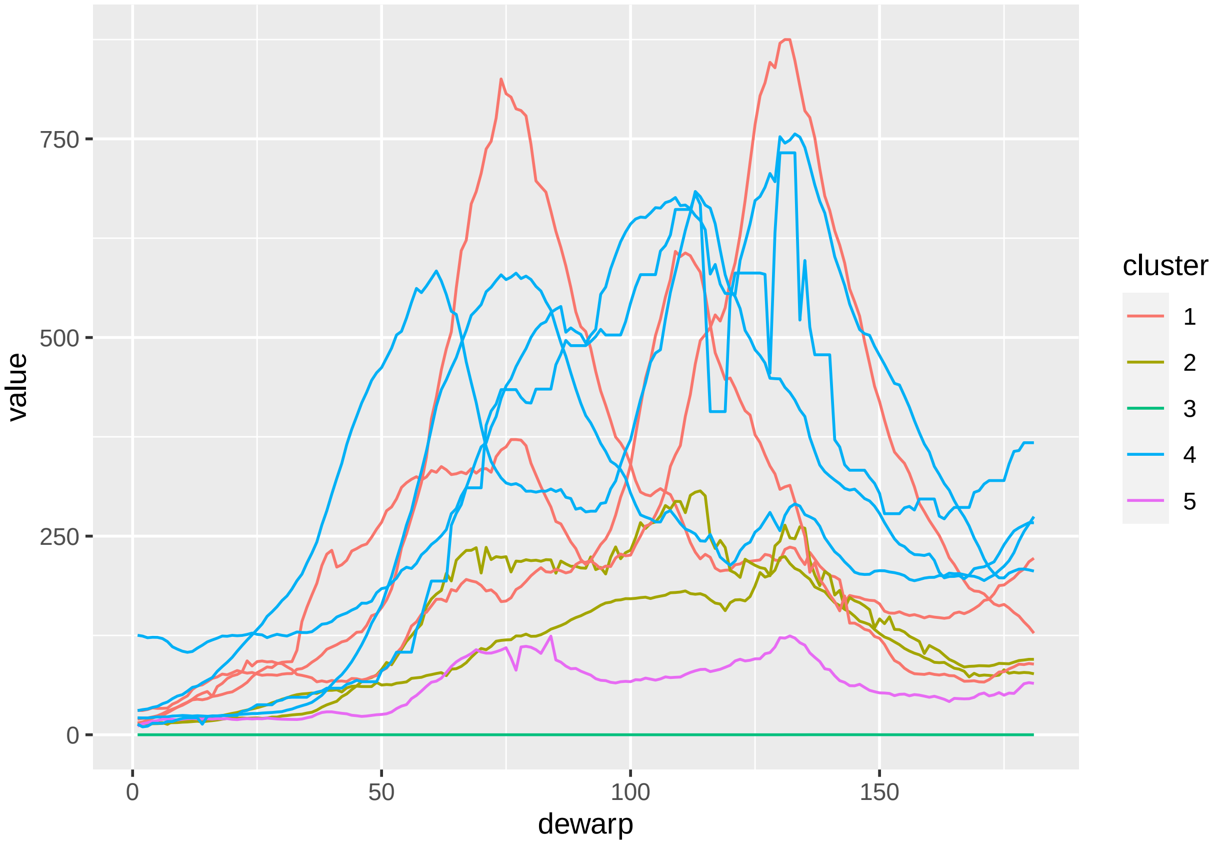
```
head(dewarp)
```

```
## # A tibble: 6 x 4
##   cluster country    name value
##   <fct>    <chr>      <int> <dbl>
## 1 1        cases.AUT      1  30.6
## 2 1        cases.AUT      2  30.6
## 3 1        cases.AUT      3  31.8
## 4 1        cases.AUT      4  33.8
## 5 1        cases.AUT      5  33.2
## 6 1        cases.AUT      6  33.3
```

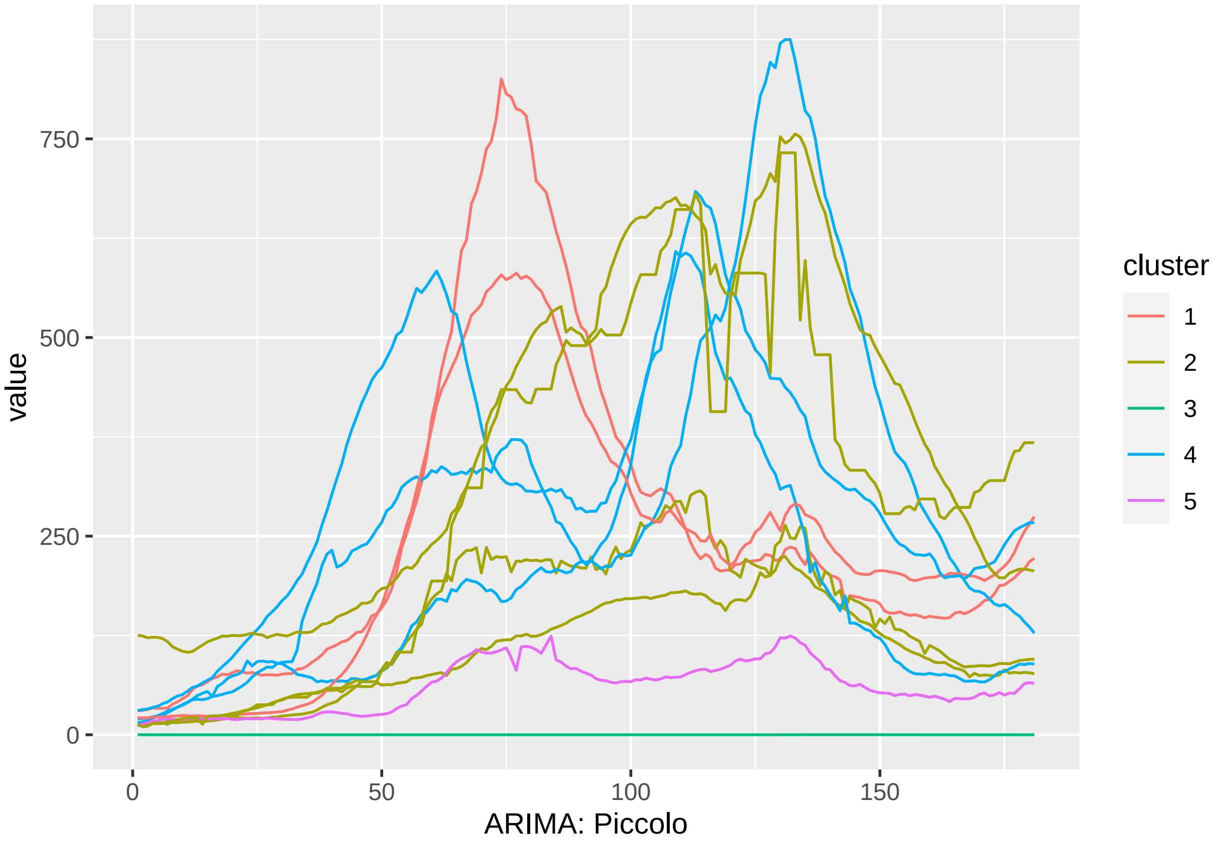
```
eucl %>% ggplot(aes(x = as.numeric(name), y = value, group_by = country , color = cluster)) +
    geom_line() +
    xlab("eucl")
```



```
dewarp %>% ggplot(aes(x = as.numeric(name), y = value, group_by = country , color = cluster)) +
    geom_line() +
    xlab("dewarp")
```

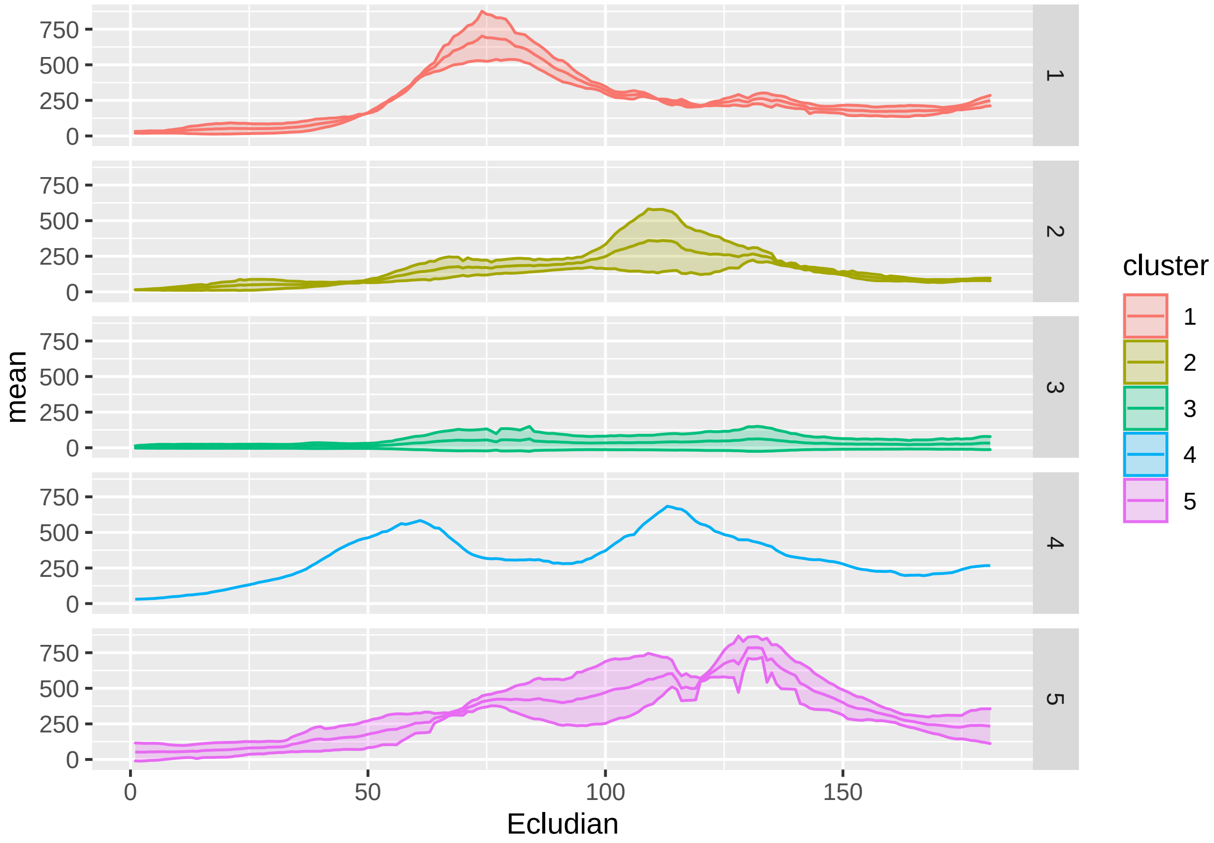


```
pic %>% ggplot(aes(x = as.numeric(name), y = value, group_by = country , color = cluster)) +
    geom_line() +
    xlab("ARIMA: Piccolo")
```



```
eucl %>% group_by(cluster, name) %>%
  summarize(mean = mean(value),
            sd = sd(value)) %>%
  ggplot(aes(x = as.numeric(name), y = mean, color = cluster)) +
    geom_line() +
    geom_ribbon(aes(ymin = mean - sd, ymax = mean + sd, fill = cluster), alpha = 0.25) +
    xlab("Ecludian") +
    facet_grid(cluster ~ .)
```

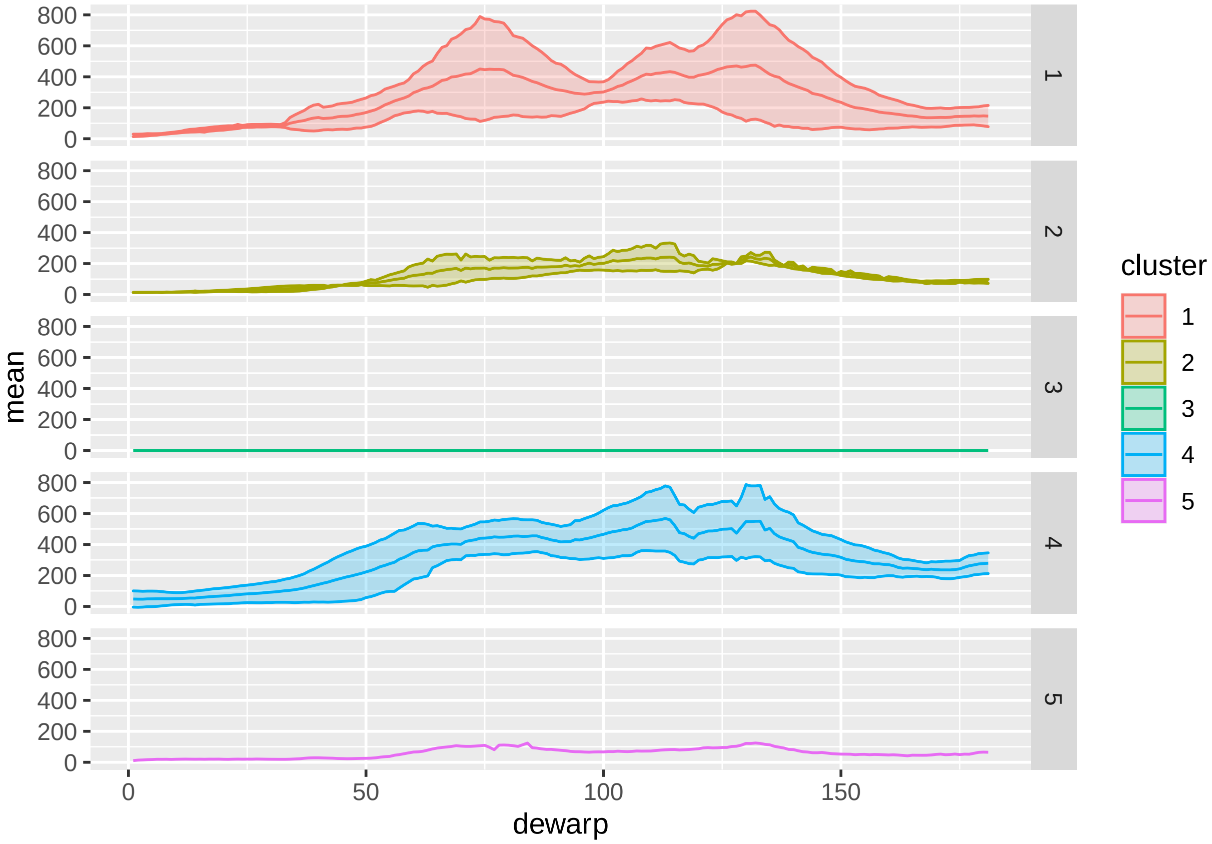
```
## `summarise()` has grouped output by 'cluster'. You can override using the
## `.groups` argument.

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf
```



```
dewarp %>% group_by(cluster, name) %>%
  summarize(mean = mean(value),
            sd = sd(value)) %>%
  ggplot(aes(x = as.numeric(name), y = mean, color = cluster)) +
    geom_line() +
    geom_ribbon(aes(ymin = mean - sd, ymax = mean + sd, fill = cluster), alpha = 0.25) +
    xlab("dewarp") +
    facet_grid(cluster ~ .)
```

```
## `summarise()` has grouped output by 'cluster'. You can override using the
## `.groups` argument.
```

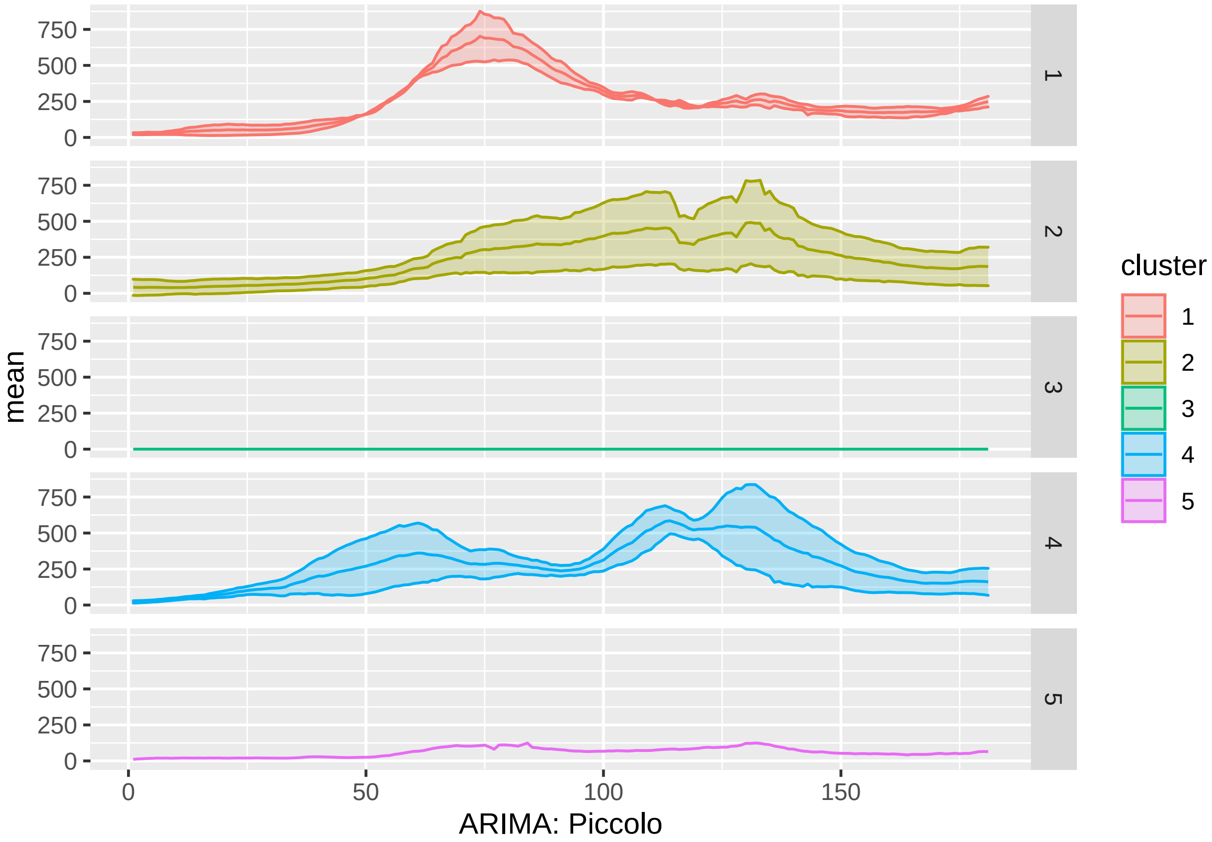
```
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf
```



```r
pic %>% group_by(cluster, name) %>%
  summarize(mean = mean(value),
            sd = sd(value)) %>%
  ggplot(aes(x = as.numeric(name), y = mean, color = cluster)) +
    geom_line() +
    geom_ribbon(aes(ymin = mean - sd, ymax = mean + sd, fill = cluster), alpha = 0.25) +
    xlab("ARIMA: Piccolo") +
    facet_grid(cluster ~ .)
```

```
## `summarise()` has grouped output by 'cluster'. You can override using the
## `.groups` argument.
```

```
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf
```



d)

Compare the resulting clusters- which differences do you see between the methods? The ecludian and the dynamic time warping do both ignore connectivity if two series is out of sync. If we amuse that a country has the tendency of spread the disease to its neighbor country, the latency of the spreading will in both cases increase the distance. In the case of ARIMA, the model distance is based on the relationships of a time points neighboring points. This is not affected by the series latency, but the series over all structur.

Wich distance measure is most appropiate for this type of dataset? For the reason stated above, a will only recommend ARIMA as meaning full model.

Do the clusters support the hypothesis that countries with low spartial proximmit tend to show similar dynamics during the give nperiod in the pandemic? Based on the reslts above, we can see that there are some concretion for countrys with low spartial proximmit. At the same time we can se clear example demonstrating that this is not true. Example is SWE and USA. We can also see that neighboring countrys having a lot of distance. So we can say that there are some support, but equaly much disagreeing