Cluster Analysis

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
expect <- read.csv("/Users/poojadesai/Downloads/MVA/lfe.csv")

library(cluster)
library(readr)
head(expect)

##       Country Year     Status Life.expectancy Adult.Mortality
infant.deaths
## 1 Afghanistan 2015 Developing            65.0             263
62
## 2 Afghanistan 2014 Developing            59.9             271
64
## 3 Afghanistan 2013 Developing            59.9             268
66
## 4 Afghanistan 2012 Developing            59.5             272
69
## 5 Afghanistan 2011 Developing            59.2             275
71
## 6 Afghanistan 2010 Developing            58.8             279
74
##   Alcohol percentage.expenditure Hepatitis.B Measles  BMI
under.five.deaths
## 1    0.01              71.279624          65    1154 19.1
83
## 2    0.01              73.523582          62     492 18.6
86
## 3    0.01              73.219243          64     430 18.1
89
## 4    0.01              78.184215          67    2787 17.6
93
## 5    0.01               7.097109          68    3013 17.2
97
## 6    0.01              79.679367          66    1989 16.7
102
##   Polio Total.expenditure Diphtheria HIV.AIDS       GDP Population
## 1     6              8.16         65      0.1 584.25921   33736494
## 2    58              8.18         62      0.1 612.69651     327582
## 3    62              8.13         64      0.1 631.74498   31731688
## 4    67              8.52         67      0.1 669.95900    3696958
## 5    68              7.87          68       0.1  63.53723    2978599
## 6    66              9.20          66       0.1 553.32894    2883167
##   thinness..1.19.years thinness.5.9.years Income.composition.of.resources
## 1                 17.2               17.3                           0.479
## 2                 17.5               17.5                           0.476
## 3                 17.7               17.7                           0.470
## 4                 17.9               18.0                           0.463
## 5                 18.2               18.2                           0.454
## 6                 18.4               18.4                           0.448
##   Schooling
## 1      10.1
## 2      10.0
## 3       9.9
## 4       9.8
## 5       9.5
## 6       9.2
```

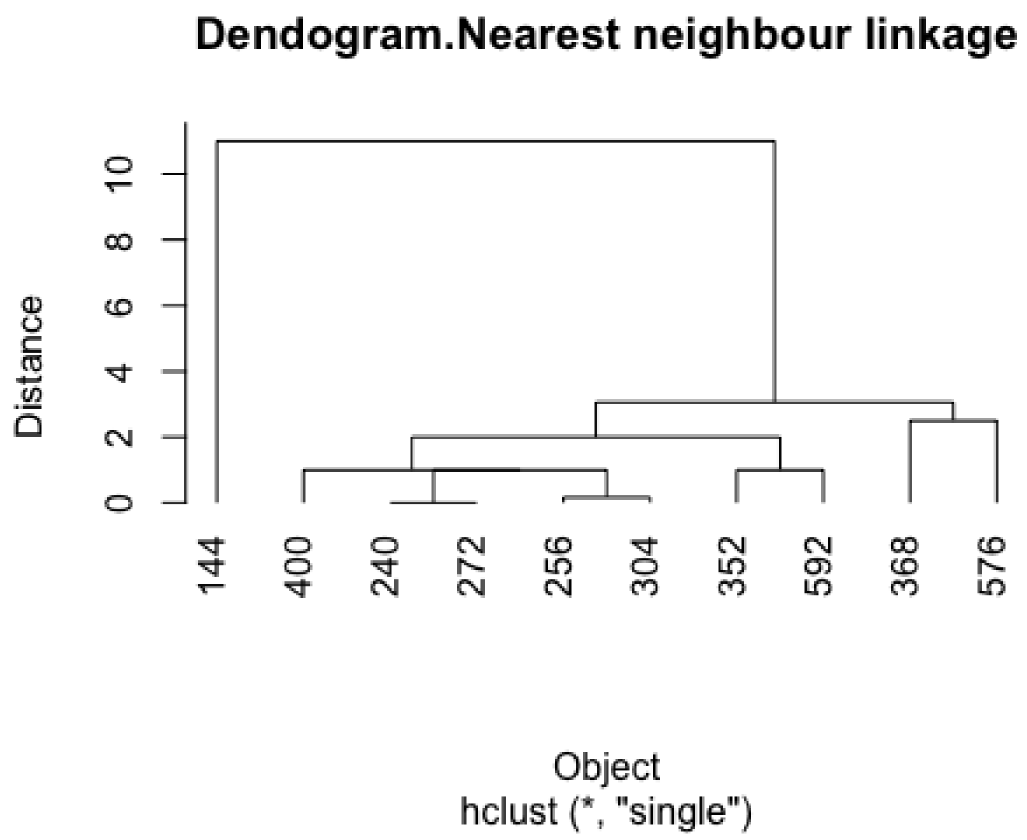
```
sapply(expect, function(x) sum(is.na(x)))
```

```
##                         Country                            Year
##                               0                               0
##                          Status                 Life.expectancy
##                               0                              10
##                 Adult.Mortality                   infant.deaths
##                              10                               0
##                         Alcohol          percentage.expenditure
##                             194                               0
##                     Hepatitis.B                         Measles
##                             553                               0
##                             BMI               under.five.deaths
##                              34                               0
##                           Polio               Total.expenditure
##                              19                             226
##                      Diphtheria                        HIV.AIDS
##                              19                               0
##                             GDP                      Population
##                             448                             652
##            thinness..1.19.years              thinness.5.9.years
##                              34                              34
## Income.composition.of.resources                       Schooling
##                             167                             163
```

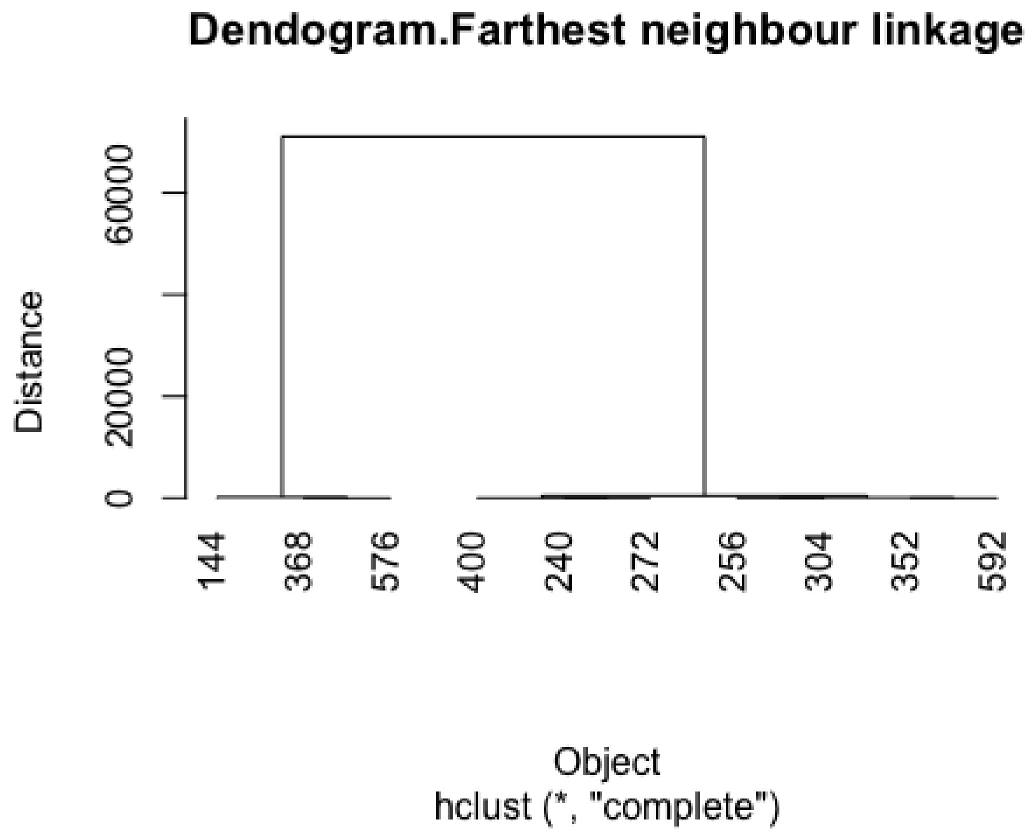
```
expect <- expect[complete.cases(expect),]  ## to remove which has null values
expect_x <- subset.data.frame(expect, Year == "2000")
expect_y <- expect_x[4:13,5:14]

colnames(expect_y) <- rownames(expect_y)
dist.expect <- as.dist(expect_y)
dist.expect
```

```
##             144          240          256          272          304
352
## 240   247.00000                                                     
## 256    11.00000      1.00000                                        
## 272   196.00000      0.00000      4.79000                           
## 304   312.00000      1.00000      0.17000     93.35873              
## 352   647.00000      2.00000      5.37000    250.89165     86.00000
## 368   183.00000    111.00000      7.26000    179.47773     94.00000
36.00000
## 400   163.00000      1.00000      9.69000     15.23573     94.00000
46.00000
## 576   115.00000    490.00000      3.06000     17.46057      6.00000
71093.00000
## 592   167.00000     18.00000      4.66000    477.13418     78.00000
1.00000
##             368          400          576
## 240                                     
## 256                                     
## 272                                     
## 304                                     
## 352                                     
## 368                                     
## 400    57.00000                         
## 576     2.50000    608.00000            
## 592    46.70000     21.00000     82.00000
```

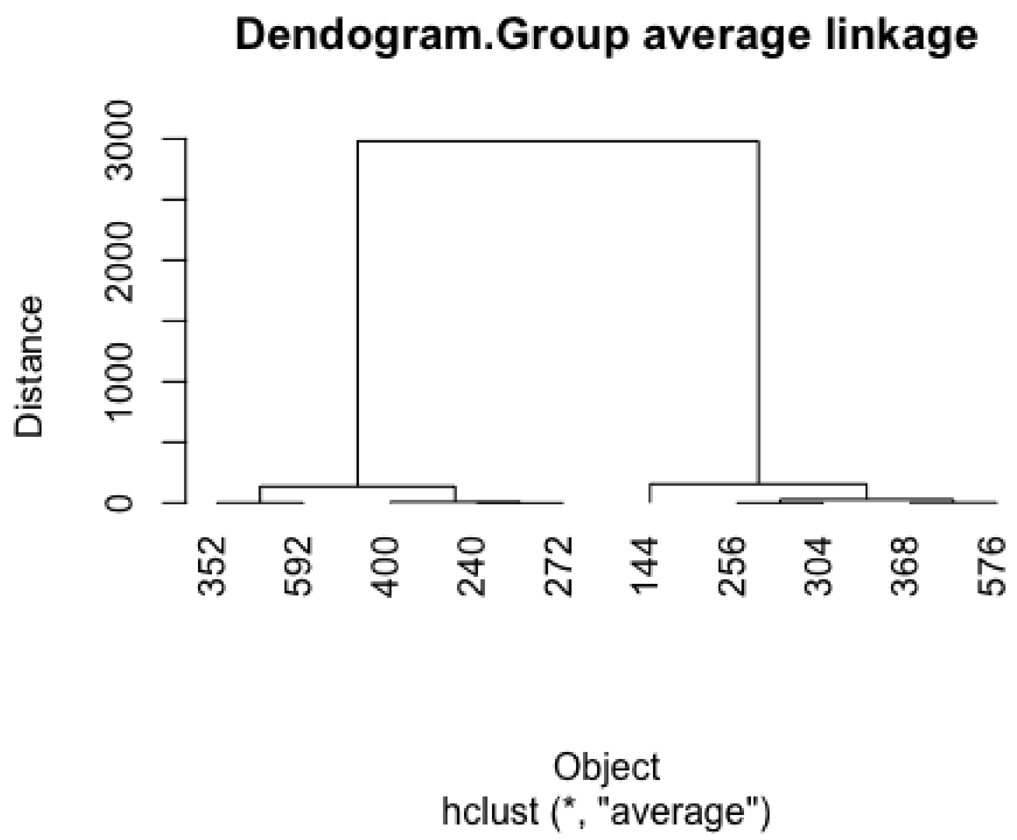
```
expect_single <- hclust(dist.expect,method = "single")
plot(expect_single,hang=-1,xlab="Object",ylab="Distance",
     main="Dendogram.Nearest neighbour linkage")
```



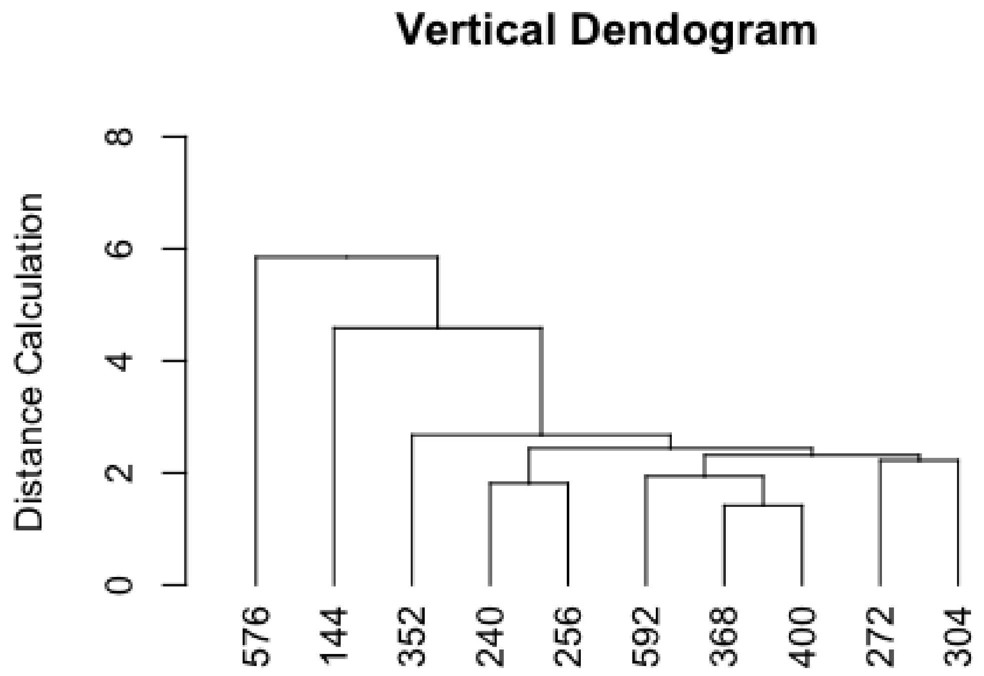
```
expect_complete <- hclust(dist.expect)
plot(expect_complete,hang=-1,xlab="Object",ylab="Distance",
     main="Dendogram.Farthest neighbour linkage")
```



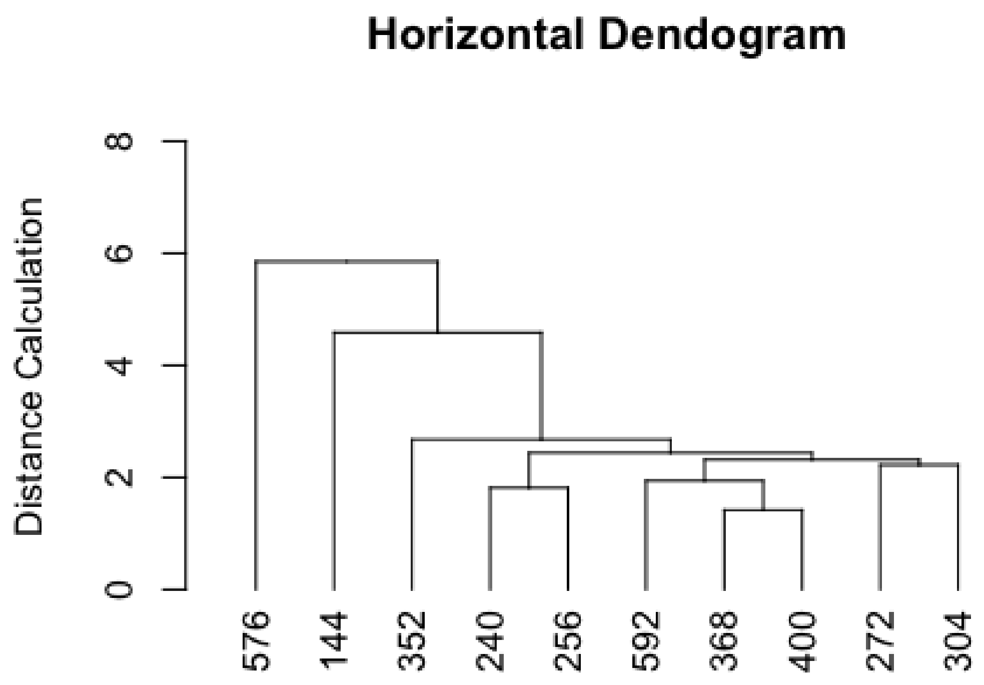
```
expect_average <- hclust(dist.expect,method = "average")
plot(expect_average,hang=-1,xlab="Object",ylab="Distance",
     main="Dendogram.Group average linkage")
```



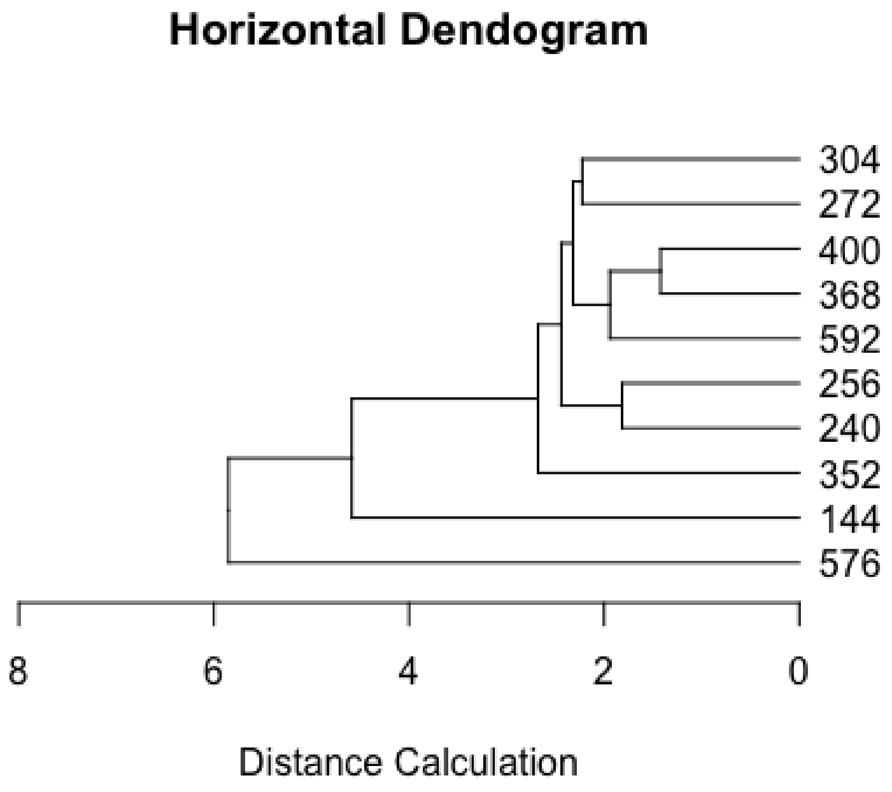
```
expect.can <- scale(expect_y)
expect.euclidean <- dist(expect.can,method = "euclidean")

#Invoking hclust command (cluster analysis by single linkage method)
expect_hclust <- hclust(expect.euclidean,method="single")

#Plotting vertical dendogram
plot(as.dendrogram(expect_hclust),ylab="Distance Calculation",
     ylim= c(0,8),main="Vertical Dendogram")
```



```
plot(as.dendrogram(expect_hclust),ylab="Distance Calculation",
     ylim= c(0,8),main="Horizontal Dendogram")
```



```
plot(as.dendrogram(expect_hclust),xlab="Distance Calculation",
     xlim= c(8,0),horiz = TRUE,main="Horizontal Dendogram")
```



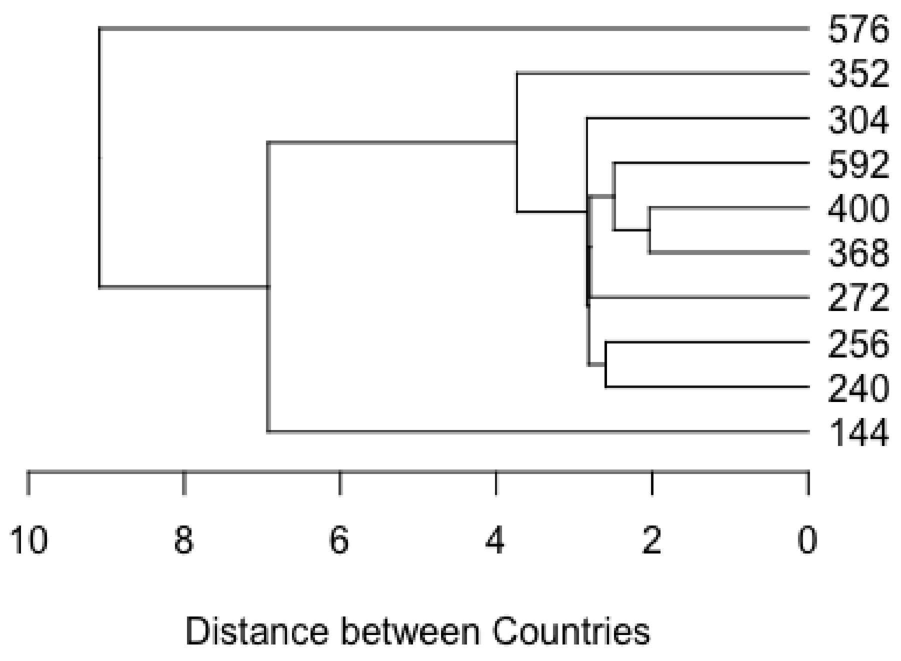
```
agnes

## function (x, diss = inherits(x, "dist"), metric = "euclidean",
##     stand = FALSE, method = "average", par.method, keep.diss = n <
##         100, keep.data = !diss, trace.lev = 0)
## {
##     METHODS <- c("average", "single", "complete", "ward", "weighted",
##         "flexible", "gaverage")
##     meth <- pmatch(method, METHODS)
##     if (is.na(meth))
##         stop("invalid clustering method")
##     if (meth == -1)
##         stop("ambiguous clustering method")
##     cl. <- match.call()
##     method <- METHODS[meth]
##     if (method == "flexible") {
##         stopifnot((np <- length(a <- as.numeric(par.method))) >=
##             1)
##         attr(method, "par") <- par.method <- if (np == 1)
##             c(a, a, 1 - 2 * a, 0)
##         else if (np == 3)
##             c(a, 0)
##         else if (np == 4)
##             a
##         else stop("'par.method' must be of length 1, 3, or 4")
##     }
##     else if (method == "gaverage") {
##         attr(method, "par") <- par.method <- if (missing(par.method)) {
##             beta <- -0.1
##             c(1 - beta, 1 - beta, beta, 0)
##         }
##         else {
##             stopifnot((np <- length(b <- as.numeric(par.method))) >=
##                 1)
##             if (np == 1)
##                 c(1 - b, 1 - b, b, 0)
##             else if (np == 3)
##                 c(b, 0)
##             else if (np == 4)
##                 b
##             else stop("'par.method' must be of length 1, 3, or 4")
##         }
##     }
##     else par.method <- double()
##     if ((diss <- as.logical(diss))) {
##         if (anyNA(x))
##             stop("NA-values in the dissimilarity matrix not allowed.")
##         if (data.class(x) != "dissimilarity") {
##             if (!is.null(dim(x))) {
##                 x <- as.dist(x)
##             }
##             else {
##                 if (!is.numeric(x) || is.na(n <- sizeDiss(x)))
##                   stop("'x' is not and cannot be converted to class
\"dissimilarity\"")
##                 attr(x, "Size") <- n
##             }
##             class(x) <- dissiCl
##             if (is.null(attr(x, "Metric")))
##                 attr(x, "Metric") <- "unspecified"
##         }
##         n <- attr(x, "Size")
##         dv <- x[lower.to.upper.tri.inds(n)]
##         dv <- c(0, dv)
##         jp <- 1L
##         mdata <- FALSE
##         ndyst <- 0
##         x2 <- double(1)
##     }
##     else {
##         x <- data.matrix(x)
##         if (!is.numeric(x))
##             stop("x is not a numeric dataframe or matrix.")
##         x2 <- if (stand)
##             scale(x, scale = apply(x, 2, meanabsdev))
##         else x
##         storage.mode(x2) <- "double"
##         ndyst <- if (metric == "manhattan")
##             2
##         else 1
##         n <- nrow(x2)
##         jp <- ncol(x2)
##         if ((mdata <- any(inax <- is.na(x2)))) {
##             jtmd <- integer(jp)
##             jtmd[apply(inax, 2L, any)] <- -1L
##             valmisdat <- 1.1 * max(abs(range(x2, na.rm = TRUE)))
##             x2[inax] <- valmisdat
##         }
##         dv <- double(1 + (n * (n - 1))/2)
##     }
##     if (n <= 1)
##         stop("need at least 2 objects to cluster")
##     stopifnot(length(trace.lev <- as.integer(trace.lev)) == 1)
##     C.keep.diss <- keep.diss && !diss
##     res <- .C(twins, as.integer(n), as.integer(jp), x2, dv, dis =
double(if (C.keep.diss) length(dv) else 1),
##         jdyss = if (C.keep.diss) diss + 10L else as.integer(diss),
##         if (mdata) rep(valmisdat, jp) else double(1), if (mdata) jtmd else
integer(jp),
##         as.integer(ndyst), 1L, meth, integer(n), ner = integer(n),
##         ban = double(n), ac = double(1), par.method, merge = matrix(0L,
##             n - 1, 2), trace = trace.lev)
##     if (!diss) {
##         if (res$jdyss == -1)
##             stop("No clustering performed, NA-values in the dissimilarity
matrix.\n")
##         if (keep.diss) {
##             disv <- res$dis[-1]
##             disv[disv == -1] <- NA
##             disv <- disv[upper.to.lower.tri.inds(n)]
##             class(disv) <- dissiCl
##             attr(disv, "Size") <- nrow(x)
##             attr(disv, "Metric") <- metric
##             attr(disv, "Labels") <- dimnames(x)[[1]]
##         }
##         if (length(dimnames(x)[[1]]) != 0)
##             order.lab <- dimnames(x)[[1]][res$ner]
##     }
##     else {
##         if (keep.diss)
##             disv <- x
##         if (length(attr(x, "Labels")) != 0)
##             order.lab <- attr(x, "Labels")[res$ner]
##     }
##     clustering <- list(order = res$ner, height = res$ban[-1],
##         ac = res$ac, merge = res$merge, diss = if (keep.diss) disv,
##         call = cl., method = METHODS[meth])
##     if (exists("order.lab"))
##         clustering$order.lab <- order.lab
##     if (keep.data && !diss) {
##         if (mdata)
##             x2[x2 == valmisdat] <- NA
##         clustering$data <- x2
##     }
##     class(clustering) <- c("agnes", "twins")
##     clustering
## }
## <bytecode: 0x7f9480d8d150>
## <environment: namespace:cluster>
```

```
agn.expect <- agnes(expect_y, metric="euclidean", stand=TRUE, method =
"single")
```

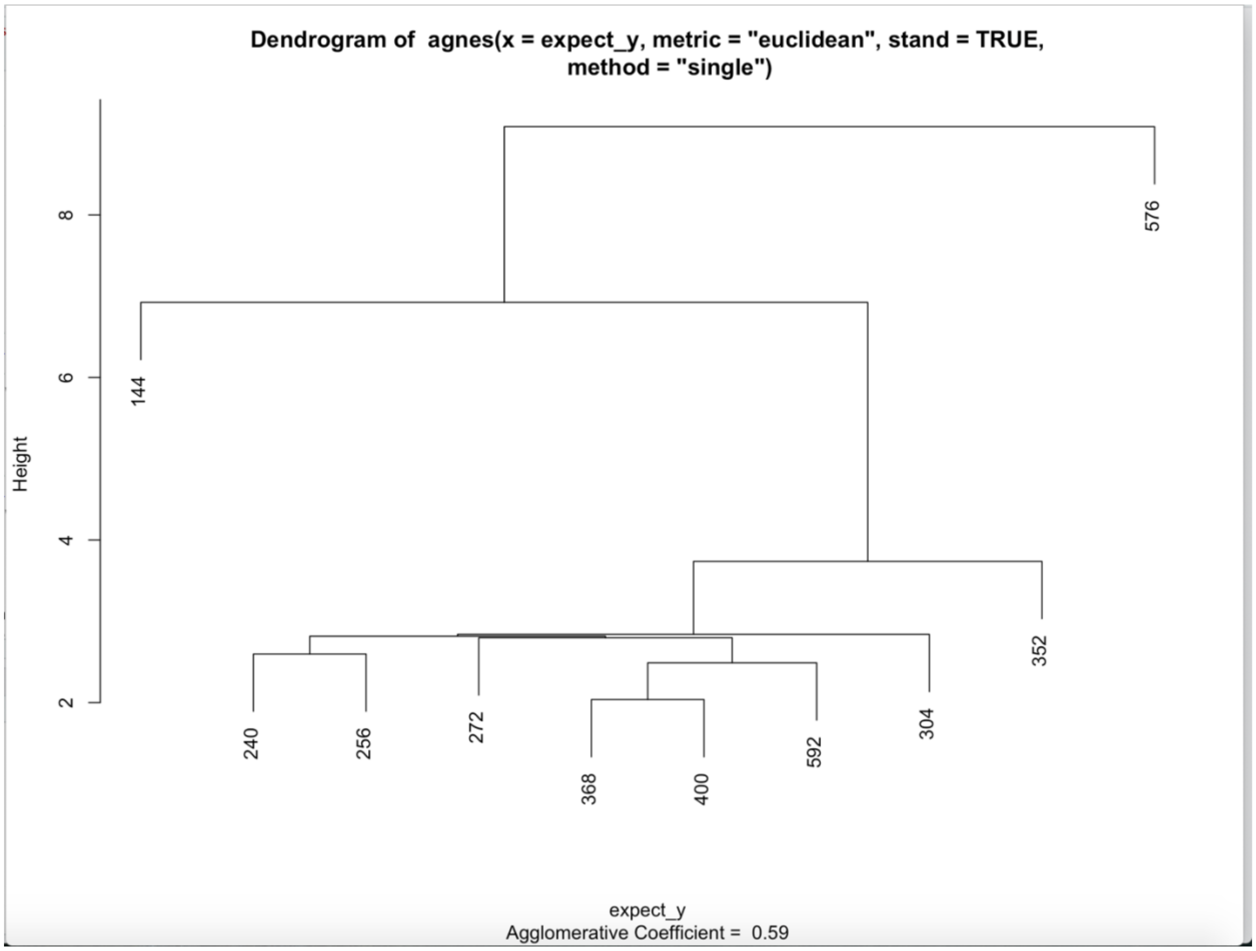
```
agn.expect$merge
```

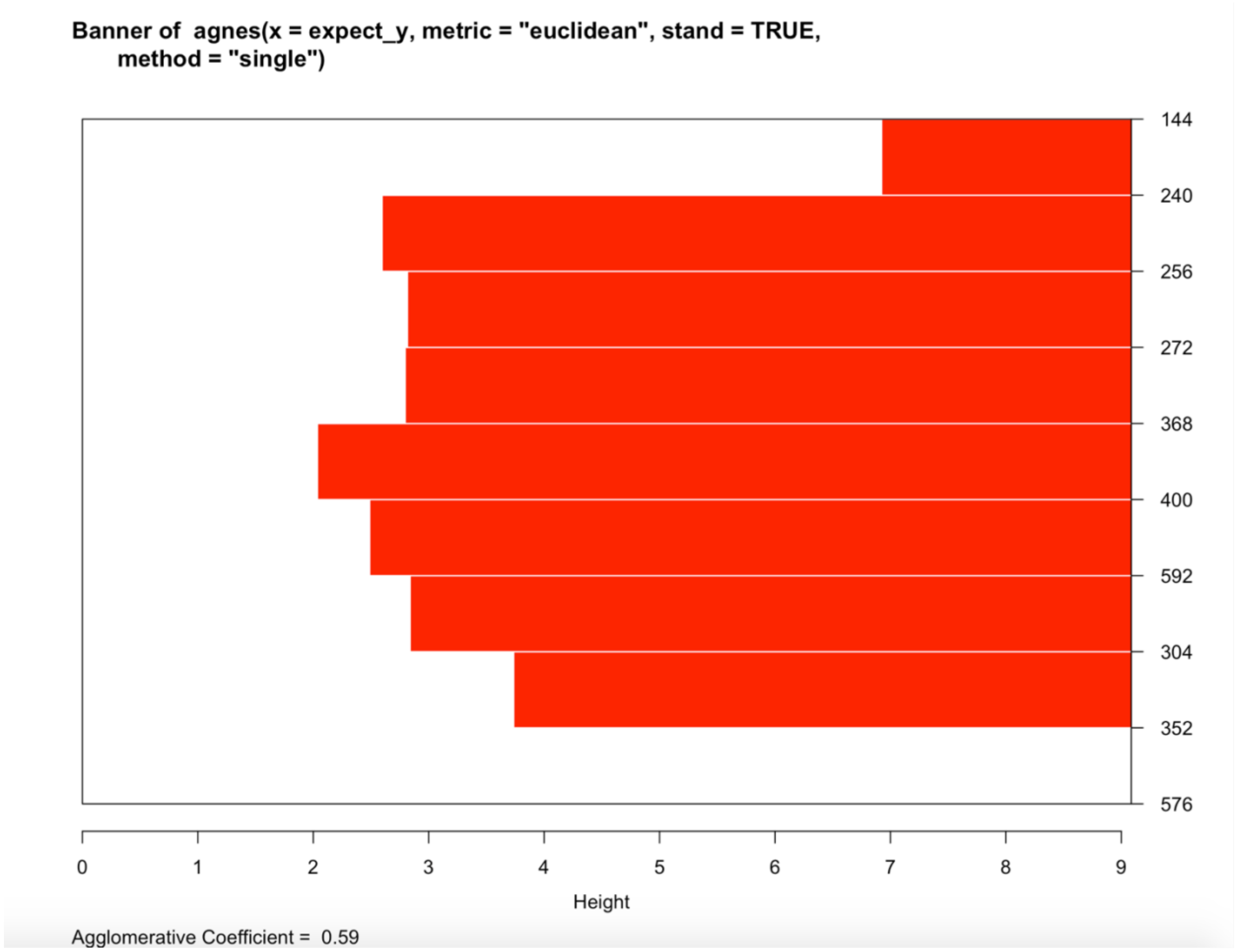
```
##       [,1] [,2]
##  [1,]   -7   -8
##  [2,]    1  -10
##  [3,]   -2   -3
##  [4,]   -4    2
##  [5,]    3    4
##  [6,]    5   -5
##  [7,]    6   -6
##  [8,]   -1    7
##  [9,]    8   -9
```

```
#Dendogram
plot(as.dendrogram(agn.expect), xlab= "Distance between
Countries",xlim=c(10,0),
     horiz = TRUE)
```



```
#Interactive Plots
plot(agn.expect,ask=TRUE)
plot(agn.expect, which.plots=2)
```


Dendrogram of agnes(x = expect_y, metric = "euclidean", stand = TRUE, method = "single")
Height
2
4
6
8
144
240
256
272
368
400
592
304
352
576
expect_y
Agglomerative Coefficient = 0.59



```
attach(expect_y)

matstd.employ <- scale(expect_y)

kmeans2.expect <- kmeans(expect_y,2,nstart = 1)

perc.var.2 <- round(100*(1 -
kmeans2.expect$betweenss/kmeans2.expect$totss),1)
names(perc.var.2) <- "Perc. 2 clus"
perc.var.2

## Perc. 2 clus
##          0.4
```

```
(kmeans3.expect <- kmeans(expect_y,3,nstart = 1))

## K-means clustering with 3 clusters of sizes 8, 1, 1
##
## Cluster means:
##      144     240      256        272    304      352     368   400  576
592
## 1 240.75  16.875  7.01625  193.32249 67.375   399.25 38.5625  19.5 94.5
6.21125
## 2 115.00 490.000  3.06000   17.46057  6.000 71093.00  2.5000 608.0 86.0
4.60000
## 3  96.00   0.000 13.20000 3557.45551 33.000     0.00  5.1000   0.0 71.0
1.60000
##
## Clustering vector:
## 144 240 256 272 304 352 368 400 576 592
##   3   1   1   1   1   1   1   1   2   1
##
## Within cluster sum of squares by cluster:
## [1] 6484019       0       0
##  (between_SS / total_SS =  99.9 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
"tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"

perc.var.3 <- round(100*(1 -
kmeans3.expect$betweenss/kmeans3.expect$totss),1)
names(perc.var.3) <- "Perc. 3 clus"
perc.var.3

## Perc. 3 clus
##          0.1

clus1 <- matrix(names(kmeans3.expect$cluster[kmeans3.expect$cluster == 1]),
                ncol=1,
nrow=length(kmeans3.expect$cluster[kmeans3.expect$cluster == 1]))
colnames(clus1) <- "Cluster 1"
clus2 <- matrix(names(kmeans3.expect$cluster[kmeans3.expect$cluster == 2]),
                ncol=1,
nrow=length(kmeans3.expect$c3luster[kmeans3.expect$cluster == 2]))
colnames(clus2) <- "Cluster 2"
clus3 <- matrix(names(kmeans3.expect$cluster[kmeans3.expect$cluster == 3]),
                ncol=1,
nrow=length(kmeans3.expect$cluster[kmeans3.expect$cluster == 3]))
colnames(clus3) <- "Cluster 3"
list(clus1,clus2,clus3)
```

```
## [[1]]
##      Cluster 1
## [1,] "240"
## [2,] "256"
## [3,] "272"
## [4,] "304"
## [5,] "352"
## [6,] "368"
## [7,] "400"
## [8,] "592"
##
## [[2]]
##      Cluster 2
##
## [[3]]
##      Cluster 3
## [1,] "144"
```