Chapter 14

Cluster Analysis

Contents

14.1 Clustering defines groups by similarity

Cluster analysis is an exploratory tool for locating and grouping observations that are similar to each other across features. Cluster analysis can also be used to group variables that are similar across observations.

Clustering or grouping is distinct from discriminant analysis and classification. In discrimination problems there are a given number of known groups to compare or distinguish. The aim in cluster analysis is to define groups based on similarities. The clusters are then examined for underlying characteristics that might help explain the grouping.

There are a variety of clustering algorithms[1]. I will discuss a simple **(agglomerative) hierarchical clustering method** for grouping observations. The method begins with each observation as an individual cluster or group. The two most similar observations are then grouped, giving one cluster with two observations. The remaining clusters have one observation. The clusters are then joined sequentially until one cluster is left.

[1] `http://cran.r-project.org/web/views/Cluster.html`

14.1.1 Agglomerative hierarchical clustering

To illustrate the steps, suppose eight observations are collected on two features X_1 and X_2. A plot of the data is given below.

Step 1. Each observation is a cluster.

Step 2. Form a new cluster by grouping the two clusters that are most similar, or closest to each other. This leaves seven clusters.

Step 3. Form a new cluster by grouping the two clusters that are most similar, or closest to each other. This leaves six clusters.

Step 4–7. Continue the process of merging clusters one at a time.

Step 8. Merge (fuse or combine) the remaining two clusters.

Finally Use a tree or dendrogram to summarize the steps in the cluster formation.

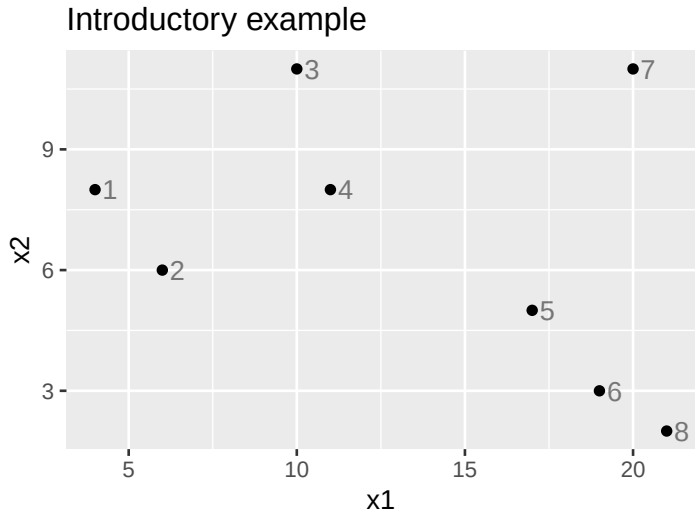
Below we have a fake dataset for illustration. We plot both the original data and the PCA scores, since in higher-dimension problems it is the PCA scores which are often used to visualize the clusters in space.

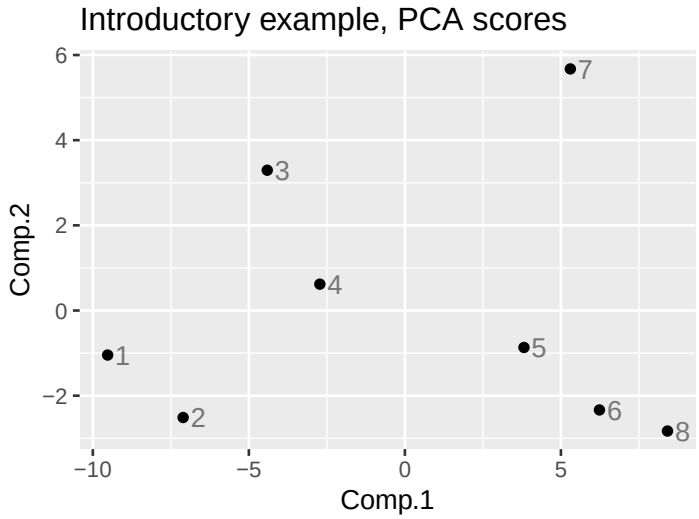
```
library(tidyverse)

#### Example: Fake data cluster illustration
# convert to a data.frame by reading the text table
dat_intro <-
  read.table(text = "
x1 x2
 4  8
 6  6
10 11
11  8
17  5
19  3
20 11
21  2
", header=TRUE)
str(dat_intro)
```

```
## 'data.frame': 8 obs. of  2 variables:
##  $ x1: int  4 6 10 11 17 19 20 21
##  $ x2: int  8 6 11 8 5 3 11 2
```

```
# perform PCA on covariance matrix
pca_intro <-
  princomp(
    ~ x1 + x2
```

```
  , data = dat_intro
  )

# plot original data
library(ggplot2)
p1 <- ggplot(dat_intro, aes(x = x1, y = x2))
p1 <- p1 + geom_point()                                                    # points
p1 <- p1 + geom_text(aes(label = 1:nrow(dat_intro)), hjust = -0.5, alpha = 0.5)  # labels
p1 <- p1 + labs(title = "Introductory example")
print(p1)

# plot PCA scores (data on PC-scale centered at 0)
library(ggplot2)
p2 <- ggplot(as.data.frame(pca_intro$scores), aes(x = Comp.1, y = Comp.2))
p2 <- p2 + geom_point()                                                    # po
p2 <- p2 + geom_text(aes(label = rownames(pca_intro$scores)), hjust = -0.5, alpha = 0.5) # lab
p2 <- p2 + labs(title = "Introductory example, PCA scores")
print(p2)
```

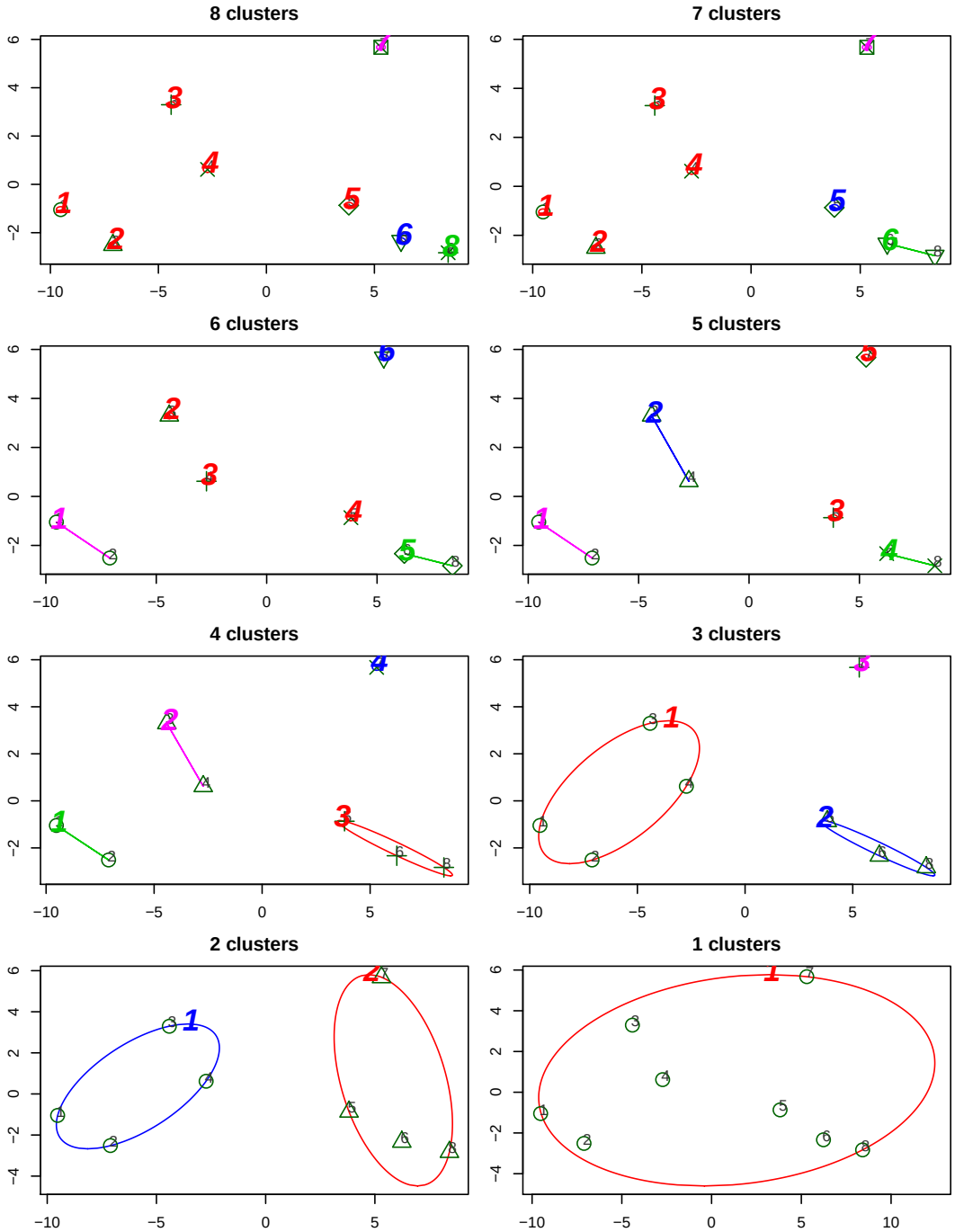




Here are the results of one distance measure, which will be discussed in more detail after the plots. The clustering algorithm order for average linkage is plotted here.

```
# create distance matrix between points
dist_intro <-
  dist(
    dat_intro
  )
dat_intro_hc_average <-
  hclust(
    dist_intro
  , method = "average"
  )
```

```r
  op <- par(no.readonly = TRUE) # save original plot options
  par(mfrow = c(4, 2), mar = c(2, 2, 2.5, 1))  # margins are c(bottom, left, top, right)

library(cluster)
for (n_clus in 8:1) {
  # create PCA scores plot with ellipses
  library(cluster)
  clusplot(
      dat_intro
    , cutree(dat_intro_hc_average, k = n_clus)
    , color   = TRUE
    , labels  = 2
    , lines   = 0
    , cex     = 2
    , cex.txt = 1
    , col.txt = "gray20"
    , main = paste(n_clus, "clusters")
    , sub     = NULL
    )
}

  par(op) # reset plot options
```



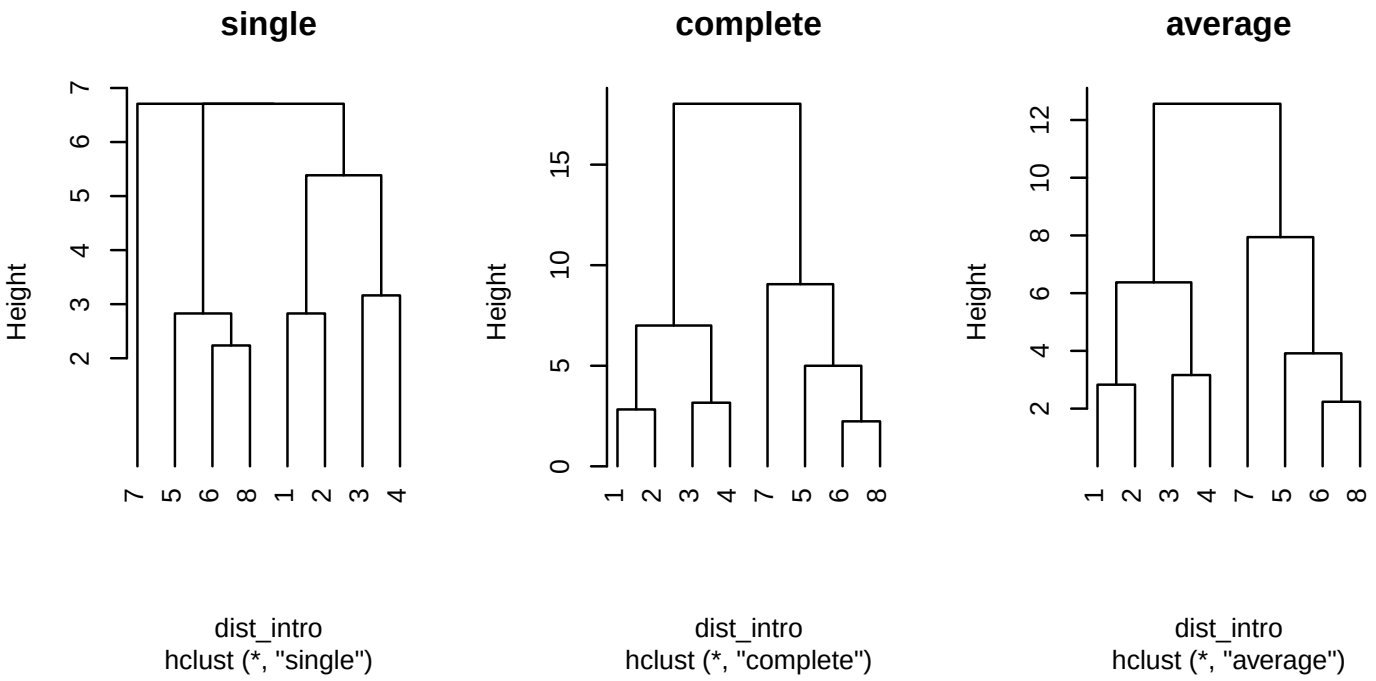
The order of clustering is summarized in the average linkage dendrogram on the right reading the tree from the bottom upwards.

```
# create distance matrix between points
dist_intro <- dist(dat_intro)
dist_intro
```

```
##           1         2         3         4         5         6         7
## 2  2.828427
## 3  6.708204  6.403124
## 4  7.000000  5.385165  3.162278
## 5 13.341664 11.045361  9.219544  6.708204
## 6 15.811388 13.341664 12.041595  9.433981  2.828427
## 7 16.278821 14.866069 10.000000  9.486833  6.708204  8.062258
## 8 18.027756 15.524175 14.212670 11.661904  5.000000  2.236068  9.055385
```

```
  op <- par(no.readonly = TRUE) # save original plot options
```

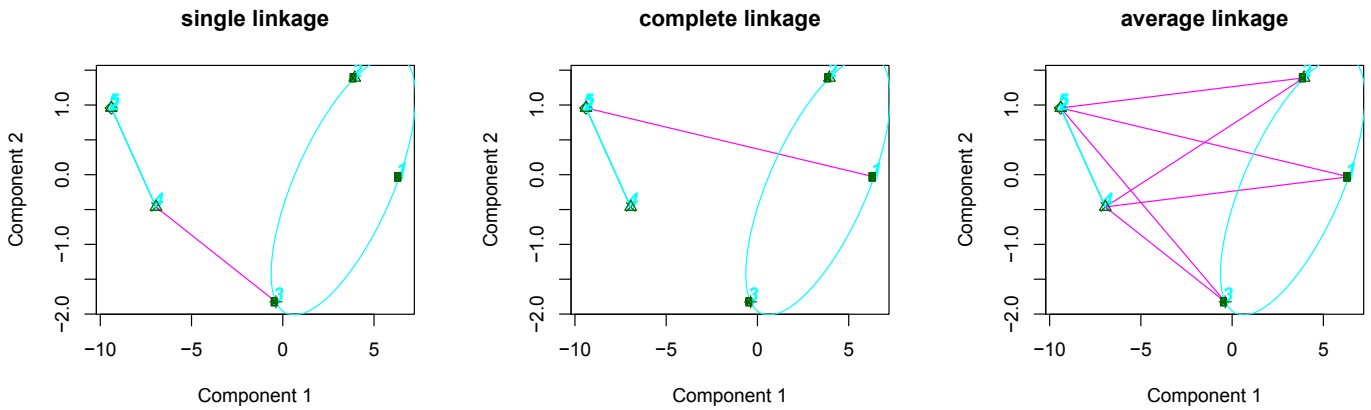
```r
  par(mfrow = c(1, 3))  # margins are c(bottom, left, top, right)

dat_intro_hc_single   <- hclust(dist_intro, method = "single"   )
plot(dat_intro_hc_single  , hang = -1,      main   = "single"   )

dat_intro_hc_complete <- hclust(dist_intro, method = "complete" )
plot(dat_intro_hc_complete, hang = -1,      main   = "complete" )

dat_intro_hc_average  <- hclust(dist_intro, method = "average"  )
plot(dat_intro_hc_average , hang = -1,      main   = "average"  )

  par(op) # reset plot options
```



14.1.2 Distance measures

There are several accepted measures of distance between clusters. The **single linkage** distance is the minimum distance between points across two clusters. The **complete linkage** distance is the maximum distance between points across two clusters. The **average linkage** distance is the average distance between points across two clusters. In these three cases the distance between points is the Euclidean or "ruler" distance. The pictures below illustrate the measures.

Given a distance measure, the distance between each pair of clusters is evaluated at each step. The two clusters that are closest to each other are merged. The observations are usually standardized prior to clustering to eliminate the effect of different variability on the distance measure.



Different distance measures can produce different **shape** clusters.

Single uses the length of the **shortest** line between points in clusters. Single linkage has the ability to produce and detect elongated and irregular clusters.

Complete uses the length of the **longest** line between points in clusters. Complete linkage is biased towards producing clusters with roughly equal diameters.

Average uses the **average** length of all line between points in clusters. Average linkage tends to produce clusters with similar variability.

Try different distances to decide the most sensible measure for your problem.

14.2 Example: Mammal teeth

The table below gives the numbers of different types of teeth for 32 mammals. The columns, from left to right, give the numbers of (v1) top incisors, (v2) bottom incisors, (v3) top canines, (v4) bottom canines, (v5) top premolars, (v6) bottom premolars, (v7) top molars, (v8) bottom molars, respectively. A cluster analysis will be used to identify the mammals that have similar counts across the eight types of teeth.

```
#### Example: Mammal teeth
## Mammal teeth data
# mammal = name
#      number of teeth
# v1 = top incisors
# v2 = bottom incisors
# v3 = top canines
# v4 = bottom canines
# v5 = top premolars
# v6 = bottom premolars
# v7 = top molars
# v8 = bottom molars

dat_teeth <-
  read_table2(
    "http://statacumen.com/teach/ADA2/notes/ADA2_notes_Ch14_teeth.dat"
  )
```

```
## Parsed with column specification:
## cols(
##   mammal = col_character(),
##   v1 = col_double(),
##   v2 = col_double(),
##   v3 = col_double(),
##   v4 = col_double(),
##   v5 = col_double(),
##   v6 = col_double(),
##   v7 = col_double(),
##   v8 = col_double()
## )
```

```
str(dat_teeth)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 32 obs. of  9 variables:
##  $ mammal: chr  "Brown_Bat" "Mole" "Silver_Hair_Bat" "Pigmy_Bat" ...
##  $ v1    : num  2 3 2 2 2 1 2 2 1 1 ...
##  $ v2    : num  3 2 3 3 3 3 1 1 1 1 ...
##  $ v3    : num  1 1 1 1 1 1 0 0 0 0 ...
##  $ v4    : num  1 0 1 1 1 1 0 0 0 0 ...
```

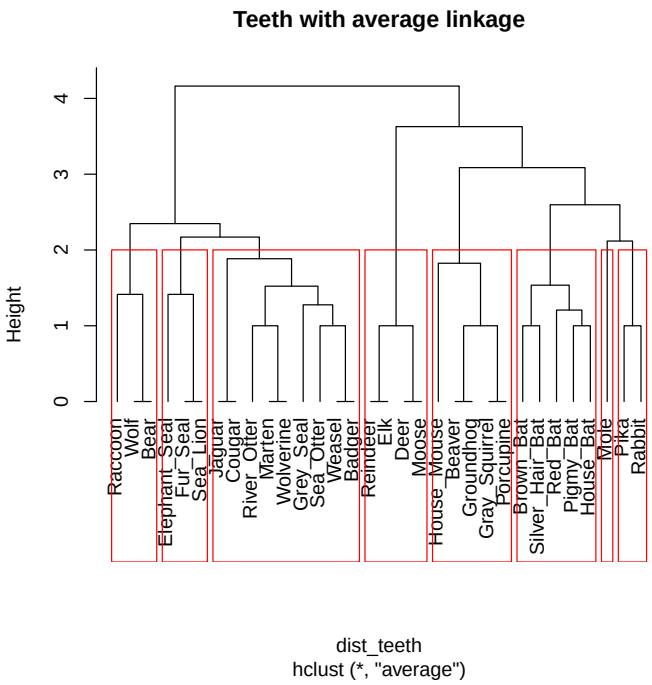
```
##  $ v5    : num  3 3 2 2 1 2 2 3 2 2 ...
##  $ v6    : num  3 3 3 2 2 2 2 2 1 1 ...
##  $ v7    : num  3 3 3 3 3 3 3 3 3 3 ...
##  $ v8    : num  3 3 3 3 3 3 3 3 3 3 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   mammal = col_character(),
##   ..   v1 = col_double(),
##   ..   v2 = col_double(),
##   ..   v3 = col_double(),
##   ..   v4 = col_double(),
##   ..   v5 = col_double(),
##   ..   v6 = col_double(),
##   ..   v7 = col_double(),
##   ..   v8 = col_double()
##   .. )
```

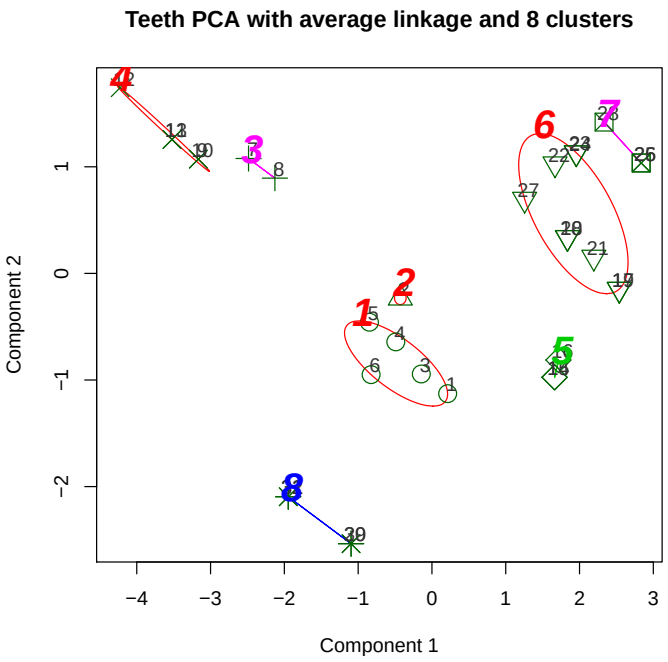
	mammal	v1	v2	v3	v4	v5	v6	v7	v8
1	Brown_Bat	2	3	1	1	3	3	3	3
2	Mole	3	2	1	0	3	3	3	3
3	Silver_Hair_Bat	2	3	1	1	2	3	3	3
4	Pigmy_Bat	2	3	1	1	2	2	3	3
5	House_Bat	2	3	1	1	1	2	3	3
6	Red_Bat	1	3	1	1	2	2	3	3
7	Pika	2	1	0	0	2	2	3	3
8	Rabbit	2	1	0	0	3	2	3	3
9	Beaver	1	1	0	0	2	1	3	3
10	Groundhog	1	1	0	0	2	1	3	3
11	Gray_Squirrel	1	1	0	0	1	1	3	3
12	House_Mouse	1	1	0	0	0	0	3	3
13	Porcupine	1	1	0	0	1	1	3	3
14	Wolf	3	3	1	1	4	4	2	3
15	Bear	3	3	1	1	4	4	2	3
16	Raccoon	3	3	1	1	4	4	3	2
17	Marten	3	3	1	1	4	4	1	2
18	Weasel	3	3	1	1	3	3	1	2
19	Wolverine	3	3	1	1	4	4	1	2
20	Badger	3	3	1	1	3	3	1	2
21	River_Otter	3	3	1	1	4	3	1	2
22	Sea_Otter	3	2	1	1	3	3	1	2
23	Jaguar	3	3	1	1	3	2	1	1
24	Cougar	3	3	1	1	3	2	1	1
25	Fur_Seal	3	2	1	1	4	4	1	1
26	Sea_Lion	3	2	1	1	4	4	1	1
27	Grey_Seal	3	2	1	1	3	3	2	2
28	Elephant_Seal	2	1	1	1	4	4	1	1
29	Reindeer	0	4	1	0	3	3	3	3
30	Elk	0	4	1	0	3	3	3	3
31	Deer	0	4	0	0	3	3	3	3
32	Moose	0	4	0	0	3	3	3	3

The program below produces cluster analysis summaries for the mammal teeth data.

```
# create distance matrix between points
dist_teeth <-
  dist(
```

```
    dat_teeth[,-1]  # only use numeric columns, not mammal labels
  )

# number of clusters to identify with red boxes and ellipses
n_clus <- 8

# create dendrogram
hc_teeth_average <-
  hclust(
    dist_teeth
  , method = "average"
  )
plot(
    hc_teeth_average
  , hang   = -1
  , main   = paste("Teeth with average linkage") # and", n_clus, "clusters")
  , labels = dat_teeth$mammal  # if error: "invalid dendrogram input", make character list
  )
rect.hclust(
    hc_teeth_average
  , k = n_clus
  )

# create PCA scores plot with ellipses
library(cluster)
clusplot(
    dat_teeth[,-1]
  , cutree(hc_teeth_average, k = n_clus)
  , color   = TRUE
  , labels  = 2
  , lines   = 0
  , cex     = 2
  , cex.txt = 1
  , col.txt = "gray20"
  , main    = paste("Teeth PCA with average linkage and", n_clus, "clusters")
  , sub     = NULL
  )
```


Teeth with average linkage
Height
0
1
2
3
4
Raccoon
Wolf
Bear
Elephant_Seal
Fur_Seal
Sea_Lion
Jaguar
Cougar
River_Otter
Marten
Wolverine
Grey_Seal
Sea_Otter
Weasel
Badger
Reindeer
Elk
Deer
Moose
House_Mouse
Beaver
Groundhog
Gray_Squirrel
Porcupine
Brown_Bat
Silver_Hair_Bat
Red_Bat
Pigmy_Bat
House_Bat
Mole
Pika
Rabbit
dist_teeth
hclust (*, "average")


Teeth PCA with average linkage and 8 clusters
Component 2
−2
−1
0
1
−4
−3
−2
−1
0
1
2
3
Component 1
1
2
3
4
5
6
7
8

14.3 Identifying the Number of Clusters

Cluster analysis can be used to produce an "optimal" splitting of the data into a prespecified number of groups or clusters, with different algorithms[2] usually giving different clusters. However, the important issue in many analyses revolves around identifying the number of clusters in the data. A simple empirical method is to continue grouping until the clusters being fused are relatively dissimilar, as measured by the normalized RMS between clusters. Experience with your data is needed to provide a reasonable stopping rule.

```
# NbClust provides methods for determining the number of clusters
#library(NbClust)

# Data needs to be a numeric matrix
dat_teeth_num <-
  dat_teeth[,-1] %>%
  as.matrix() %>%
  as.numeric()

NC_out <-
  NbClust::NbClust(
    dat_teeth_num
  , method = "average"
  , index  = "all"
  )
```

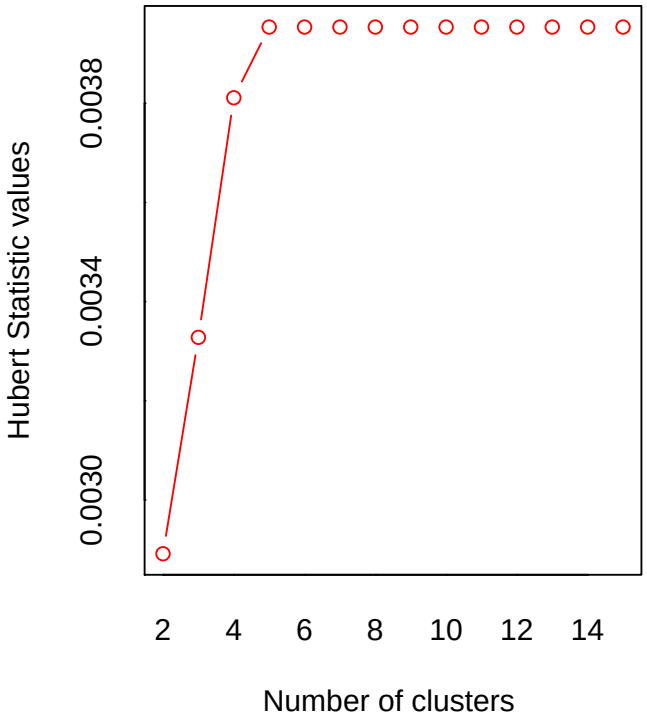
```
## Warning in max(DiffLev[, 5], na.rm = TRUE): no non-missing arguments to max; returning
-Inf
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to
##                 significant increase of the value of the measure i.e the significant peak i
##                 index second differences plot.
##
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in
##                 second differences plot) that corresponds to a significant increase of the
##                 the measure.
##
## Warning in matrix(c(results), nrow = 2, ncol = 26):  data length [51] is not a sub-multiple
or multiple of the number of rows [2]
## Warning in matrix(c(results), nrow = 2, ncol = 26, dimnames = list(c("Number_clusters",
:  data length [51] is not a sub-multiple or multiple of the number of rows [2]
## *******************************************************************
```

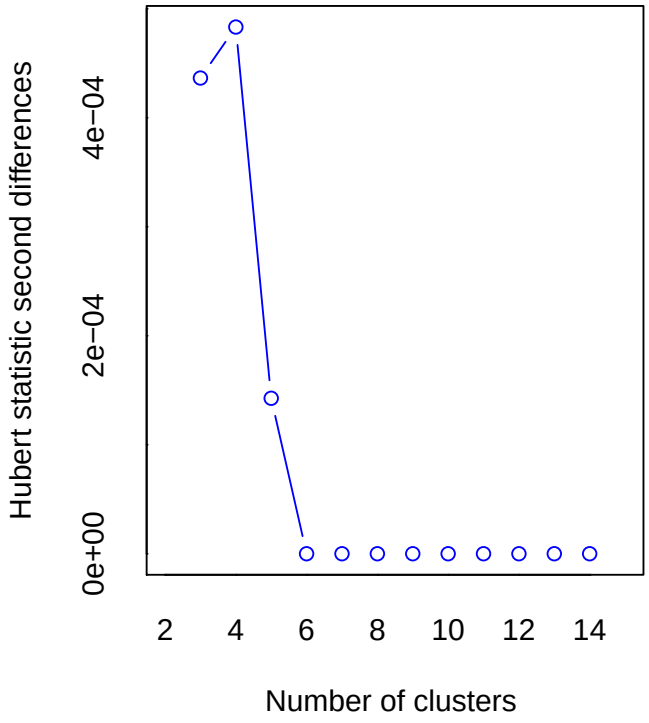
[2]There are thirty in this package: `http://cran.r-project.org/web/packages/NbClust/NbClust.pdf`

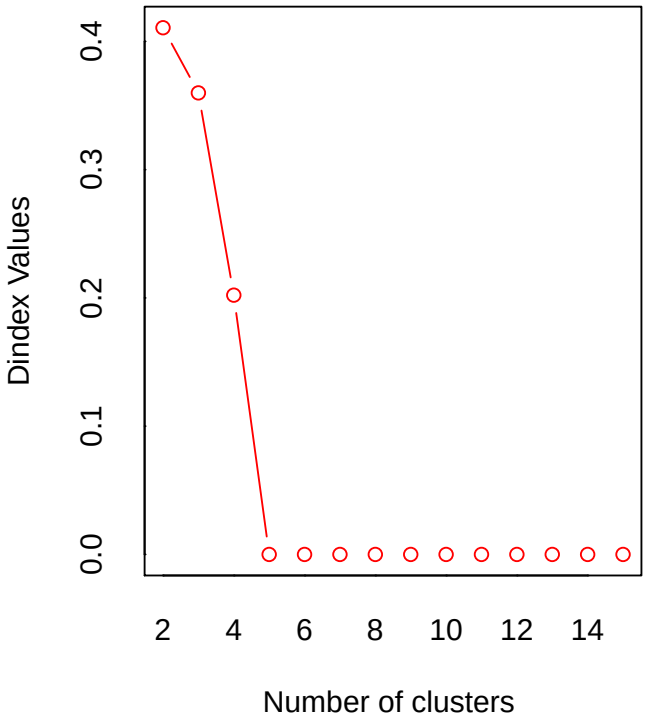
```
## * Among all indices:
## * 1 proposed 4 as the best number of clusters
## * 5 proposed 5 as the best number of clusters
##
##                     ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  5
##
##
## *******************************************************************
```

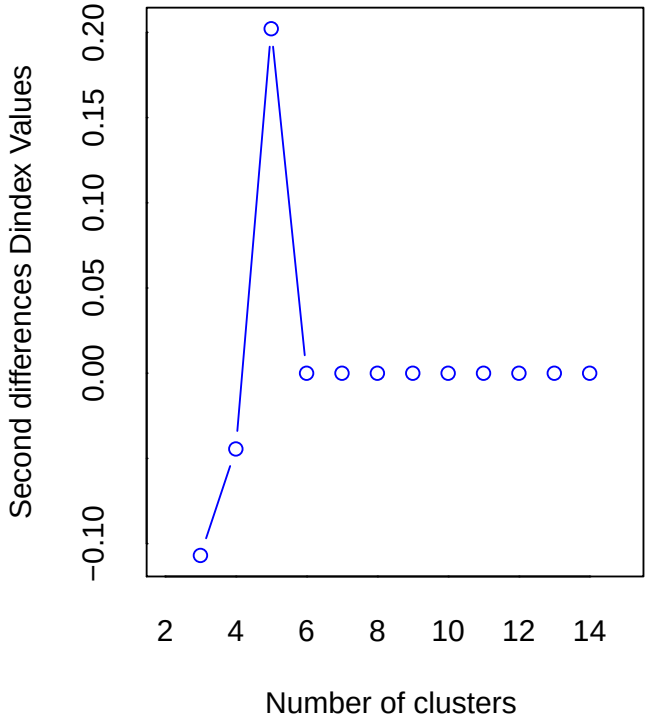
```
# most of the methods suggest 4 or 5 clusters, as do the plots
NC_out$Best.nc
```

```
##                   KL  CH Hartigan      CCC    Scott Marriot TrCovW TraceW
## Number_clusters   5   5        4   5.0000    5.000       5   -Inf 25.875
## Value_Index     Inf Inf      Inf 369.1341 7787.404     414      5  5.000
##                     Friedman         Rubin Cindex DB Silhouette   Duda PseudoT2
## Number_clusters 8.720698e+14 -9.810785e+14      0  0          1 0.4663 168.2472
## Value_Index     5.000000e+00  6.000000e+00      5  5          2 2.0000   2.0000
##                   Beale Ratkowsky    Ball PtBiserial Frey McClain Dunn Hubert
## Number_clusters 0.3789    0.4737 61.8333     0.7713   NA       0  Inf      0
## Value_Index     3.0000    3.0000  3.0000     1.0000    5       5    0      2
##                 SDindex Dindex SDbw
## Number_clusters     Inf      0    0
## Value_Index           0      5    5
```


Hubert Statistic values
0.0030
0.0034
0.0038
2
4
6
8
10
12
14
Number of clusters


Hubert statistic second differences
0e+00
2e−04
4e−04
2
4
6
8
10
12
14
Number of clusters


Dindex Values
0.0
0.1
0.2
0.3
0.4
2
4
6
8
10
12
14
Number of clusters


Second differences Dindex Values
−0.10
0.00
0.05
0.10
0.15
0.20
2
4
6
8
10
12
14
Number of clusters

There are several statistical methods for selecting the number of clusters. No method is best. They suggest using the cubic clustering criteria (`ccc`), a pseudo-F statistic, and a pseudo-t statistic. At a given step, the pseudo-t statistic is the distance between the center of the two clusters to be merged, relative to the variability within these clusters. A large pseudo-t statistic implies that the clusters to be joined are relatively dissimilar (i.e., much more variability between the clusters to be merged than within these clusters). The pseudo-F statistic at a given step measures the variability among the centers of the current clusters relative to the variability within the clusters. A large pseudo-F value implies that the clusters merged consist of fairly similar observations. As clusters are joined, the pseudo-t statistic tends to increase, and the pseudo-F statistic tends to decrease. The ccc is more difficult to describe.

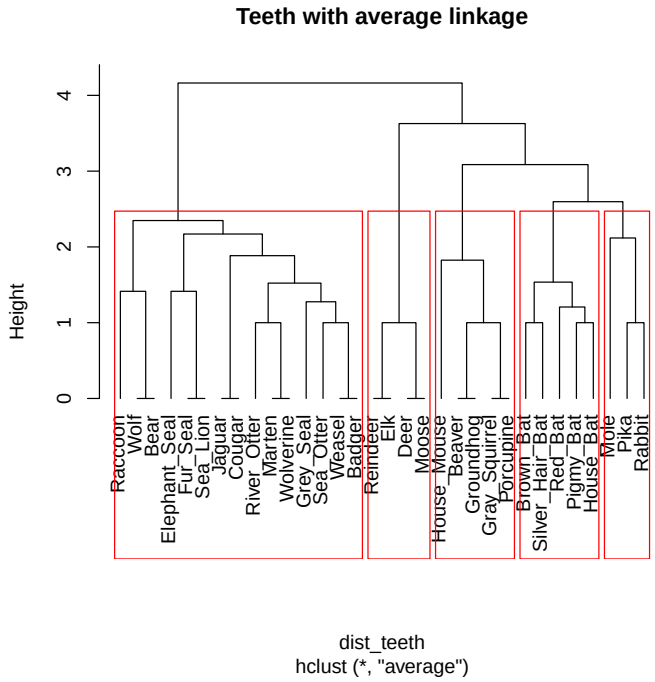
The RSQ summary is also useful for determining the number of clusters. RSQ is a pseudo-R^2 statistic that measures the proportion of the total variation explained by the differences among the existing clusters at a given step. RSQ will typically decrease as the pseudo-F statistic decreases.

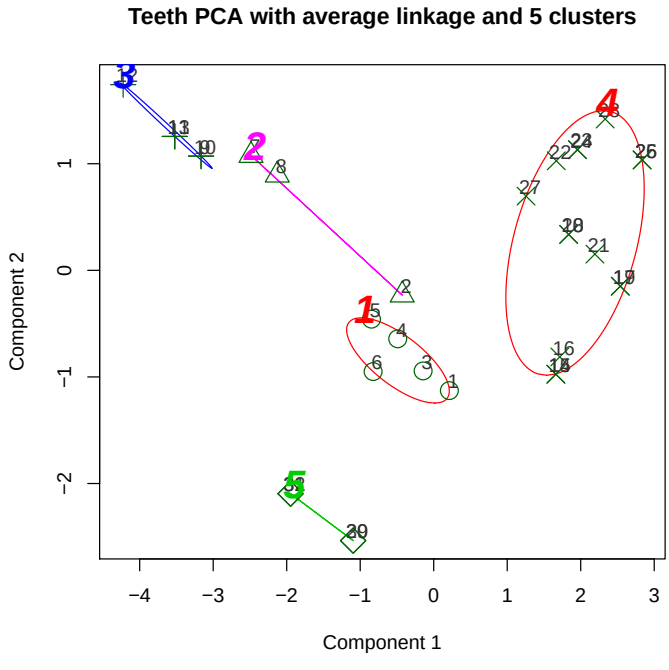
A common recommendation on cluster selection is to choose a cluster size where the values of ccc and the pseudo-F statistic are relatively high (compared to what you observe with other numbers of clusters), and where the pseudo-t statistic is relatively low and increases substantially at the next proposed merger. For the mammal teeth data this corresponds to four clusters. Six clusters is a sensible second choice.

Let's look at the results of 5 clusters.

```
# create distance matrix between points
dist_teeth <-
  dist(
    dat_teeth[,-1]  # only use numeric columns, not mammal labels
  )

# number of clusters to identify with red boxes and ellipses
n_clus <- 5

# create dendrogram
hc_teeth_average <-
  hclust(
    dist_teeth
```

```
  , method = "average"
  )
plot(
    hc_teeth_average
  , hang   = -1
  , main   = paste("Teeth with average linkage") # and", n_clus, "clusters")
  , labels = dat_teeth$mammal  # if error: "invalid dendrogram input", make character list
  )
rect.hclust(
    hc_teeth_average
  , k = n_clus
  )

# create PCA scores plot with ellipses
library(cluster)
clusplot(
    dat_teeth[,-1]
  , cutree(hc_teeth_average, k = n_clus)
  , color    = TRUE
  , labels   = 2
  , lines    = 0
  , cex      = 2
  , cex.txt  = 1
  , col.txt  = "gray20"
  , main     = paste("Teeth PCA with average linkage and", n_clus, "clusters")
  , sub      = NULL
  )
```





The mammals within each cluster are more similar to each other than to

those in other clusters. Recalling the meaning of each of `v1` to `v8` (recopied into the code below), you'll find that the clusters largely make sense with your understanding of mammals.

```
## Mammal teeth data
# mammal = name
#      number of teeth
# v1 = top incisors
# v2 = bottom incisors
# v3 = top canines
# v4 = bottom canines
# v5 = top premolars
# v6 = bottom premolars
# v7 = top molars
# v8 = bottom molars

# print the observations in each cluster
for (i_cut in 1:n_clus) {
  print(paste("Cluster", i_cut, " ----------------------------- "))
  print(dat_teeth[(cutree(hc_teeth_average, k = n_clus) == i_cut),])
}
```

```
## [1] "Cluster 1  ----------------------------- "
## # A tibble: 5 x 9
##   mammal             v1    v2    v3    v4    v5    v6    v7    v8
##   <chr>           <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Brown_Bat           2     3     1     1     3     3     3     3
## 2 Silver_Hair_Bat     2     3     1     1     2     3     3     3
## 3 Pigmy_Bat           2     3     1     1     2     2     3     3
## 4 House_Bat           2     3     1     1     1     2     3     3
## 5 Red_Bat             1     3     1     1     2     2     3     3
## [1] "Cluster 2  ----------------------------- "
## # A tibble: 3 x 9
##   mammal    v1    v2    v3    v4    v5    v6    v7    v8
##   <chr>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Mole       3     2     1     0     3     3     3     3
## 2 Pika       2     1     0     0     2     2     3     3
## 3 Rabbit     2     1     0     0     3     2     3     3
## [1] "Cluster 3  ----------------------------- "
## # A tibble: 5 x 9
##   mammal           v1    v2    v3    v4    v5    v6    v7    v8
##   <chr>         <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Beaver            1     1     0     0     2     1     3     3
## 2 Groundhog         1     1     0     0     2     1     3     3
## 3 Gray_Squirrel     1     1     0     0     1     1     3     3
## 4 House_Mouse       1     1     0     0     0     0     3     3
## 5 Porcupine         1     1     0     0     1     1     3     3
## [1] "Cluster 4  ----------------------------- "
## # A tibble: 15 x 9
```

```
##    mammal           v1    v2    v3    v4    v5    v6    v7    v8
##    <chr>         <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 Wolf              3     3     1     1     4     4     2     3
##  2 Bear              3     3     1     1     4     4     2     3
##  3 Raccoon           3     3     1     1     4     4     3     2
##  4 Marten            3     3     1     1     4     4     1     2
##  5 Weasel            3     3     1     1     3     3     1     2
##  6 Wolverine         3     3     1     1     4     4     1     2
##  7 Badger            3     3     1     1     3     3     1     2
##  8 River_Otter       3     3     1     1     4     3     1     2
##  9 Sea_Otter         3     2     1     1     3     3     1     2
## 10 Jaguar            3     3     1     1     3     2     1     1
## 11 Cougar            3     3     1     1     3     2     1     1
## 12 Fur_Seal          3     2     1     1     4     4     1     1
## 13 Sea_Lion          3     2     1     1     4     4     1     1
## 14 Grey_Seal         3     2     1     1     3     3     2     2
## 15 Elephant_Seal     2     1     1     1     4     4     1     1
## [1] "Cluster 5  ----------------------------- "
## # A tibble: 4 x 9
##   mammal      v1    v2    v3    v4    v5    v6    v7    v8
##   <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Reindeer     0     4     1     0     3     3     3     3
## 2 Elk          0     4     1     0     3     3     3     3
## 3 Deer         0     4     0     0     3     3     3     3
## 4 Moose        0     4     0     0     3     3     3     3
```

14.4 Example: 1976 birth and death rates

Below are the 1976 crude birth and death rates in 74 countries. A data plot and output from a complete and single linkage cluster analyses are given.

```
#### Example: Birth and death rates
dat_bd76 <-
  read_table2(
    "http://statacumen.com/teach/ADA2/notes/ADA2_notes_Ch14_birthdeath.dat"
  )
```

```
## Parsed with column specification:
## cols(
##   country = col_character(),
##   birth = col_double(),
##   death = col_double()
## )
```

```
str(dat_bd76)
```

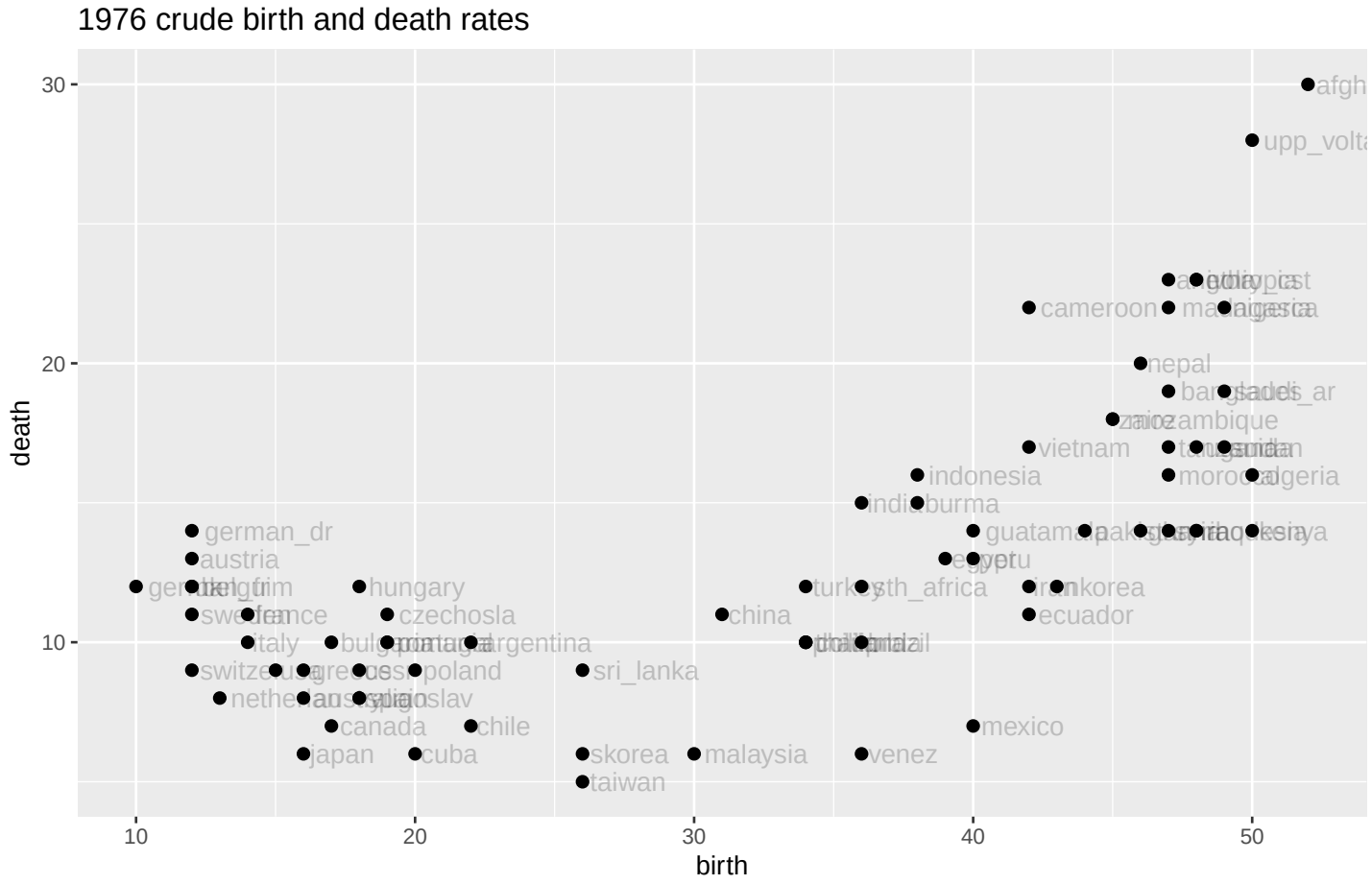
```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 74 obs. of  3 variables:
##  $ country: chr  "afghan" "algeria" "angola" "argentina" ...
##  $ birth  : num  52 50 47 22 16 12 47 12 36 17 ...
##  $ death  : num  30 16 23 10 8 13 19 12 10 10 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   country = col_character(),
##   ..   birth = col_double(),
##   ..   death = col_double()
##   .. )
```

	country	birth	death
1	afghan	52	30
2	algeria	50	16
3	angola	47	23
4	argentina	22	10
5	australia	16	8
6	austria	12	13
7	banglades	47	19
8	belguim	12	12
9	brazil	36	10
10	bulgaria	17	10
11	burma	38	15
12	cameroon	42	22
13	canada	17	7
14	chile	22	7
15	china	31	11
16	taiwan	26	5
17	columbia	34	10
18	cuba	20	6
19	czechosla	19	11
20	ecuador	42	11
21	egypt	39	13
22	ethiopia	48	23
23	france	14	11
24	german_dr	12	14
25	german_fr	10	12

	country	birth	death
26	ghana	46	14
27	greece	16	9
28	guatamala	40	14
29	hungary	18	12
30	india	36	15
31	indonesia	38	16
32	iran	42	12
33	iraq	48	14
34	italy	14	10
35	ivory_cst	48	23
36	japan	16	6
37	kenya	50	14
38	nkorea	43	12
39	skorea	26	6
40	madagasca	47	22
41	malaysia	30	6
42	mexico	40	7
43	morocco	47	16
44	mozambique	45	18
45	nepal	46	20
46	netherlan	13	8
47	nigeria	49	22
48	pakistan	44	14
49	peru	40	13
50	phillip	34	10

	country	birth	death
51	poland	20	9
52	portugal	19	10
53	rhodesia	48	14
54	romania	19	10
55	saudi_ar	49	19
56	sth_africa	36	12
57	spain	18	8
58	sri_lanka	26	9
59	sudan	49	17
60	sweden	12	11
61	switzer	12	9
62	syria	47	14
63	tanzania	47	17
64	thailand	34	10
65	turkey	34	12
66	ussr	18	9
67	uganda	48	17
68	uk	12	12
69	usa	15	9
70	upp_volta	50	28
71	venez	36	6
72	vietnam	42	17
73	yugoslav	18	8
74	zaire	45	18

```
# plot original data
library(ggplot2)
p1 <- ggplot(dat_bd76, aes(x = birth, y = death))
p1 <- p1 + geom_point(size = 2)                                    # points
p1 <- p1 + geom_text(aes(label = country), hjust = -0.1, alpha = 0.2)   # labels
p1 <- p1 + coord_fixed(ratio = 1) # makes 1 unit equal length on x- and y-axis
p1 <- p1 + labs(title = "1976 crude birth and death rates")
print(p1)
```



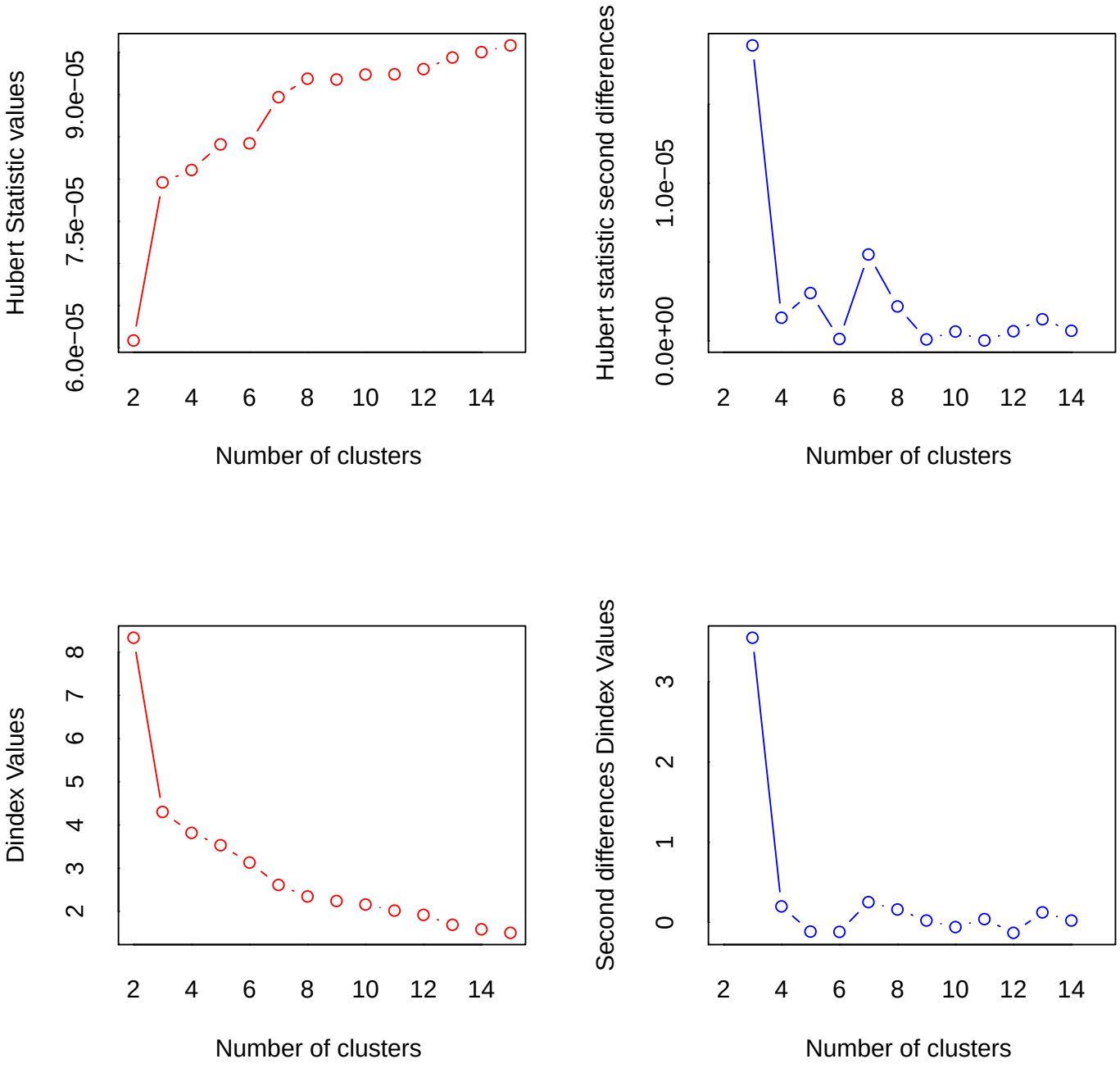
14.4.1 Complete linkage

```
#library(NbClust)
dat_bd76_num <- as.matrix(dat_bd76[,-1])
NC_out <- NbClust::NbClust(dat_bd76_num, method = "complete", index = "all")
```

```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to
##                 significant increase of the value of the measure i.e the significant peak i
##                 index second differences plot.
##
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in
##                 second differences plot) that corresponds to a significant increase of the
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 2 proposed 2 as the best number of clusters
## * 14 proposed 3 as the best number of clusters
## * 2 proposed 5 as the best number of clusters
## * 1 proposed 12 as the best number of clusters
## * 1 proposed 13 as the best number of clusters
## * 1 proposed 14 as the best number of clusters
## * 2 proposed 15 as the best number of clusters
##
##                    ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  3
##
##
## *******************************************************************
```

```
# most of the methods suggest 2 to 6 clusters, as do the plots
NC_out$Best.nc
```

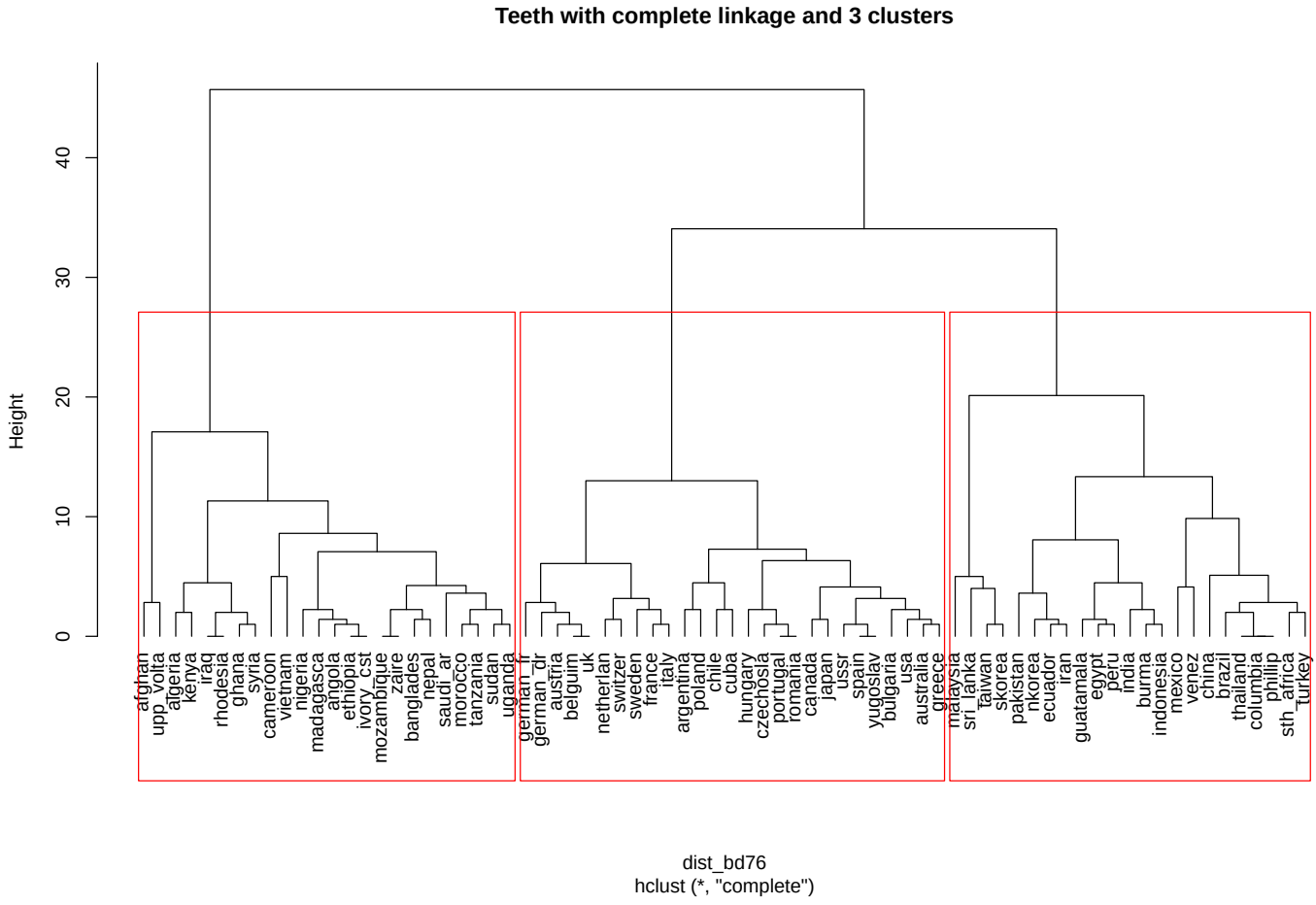
```
##                     KL       CH Hartigan     CCC    Scott  Marriot   TrCovW
## Number_clusters   3.0000  15.0000   3.0000  3.0000   3.0000        3        3
## Value_Index     127.2375 330.3549 168.7866 13.2151 129.4425 11183728 14408135
##                   TraceW Friedman    Rubin Cindex     DB Silhouette   Duda
## Number_clusters    3.000  12.0000  13.0000 3.0000 5.0000     3.0000 3.0000
## Value_Index     4430.299 186.6013 -40.8443 0.3007 0.5965     0.5696 0.4439
##                 PseudoT2  Beale Ratkowsky     Ball PtBiserial Frey McClain
## Number_clusters   3.0000 3.0000    2.0000    3.000     3.0000    1  2.0000
## Value_Index      26.3105 1.1959    0.5455 2754.832     0.7049   NA  0.3866
##                    Dunn Hubert SDindex Dindex    SDbw
## Number_clusters 14.0000      0  5.0000      0 15.0000
## Value_Index      0.2325      0  0.2448      0  0.0121
```

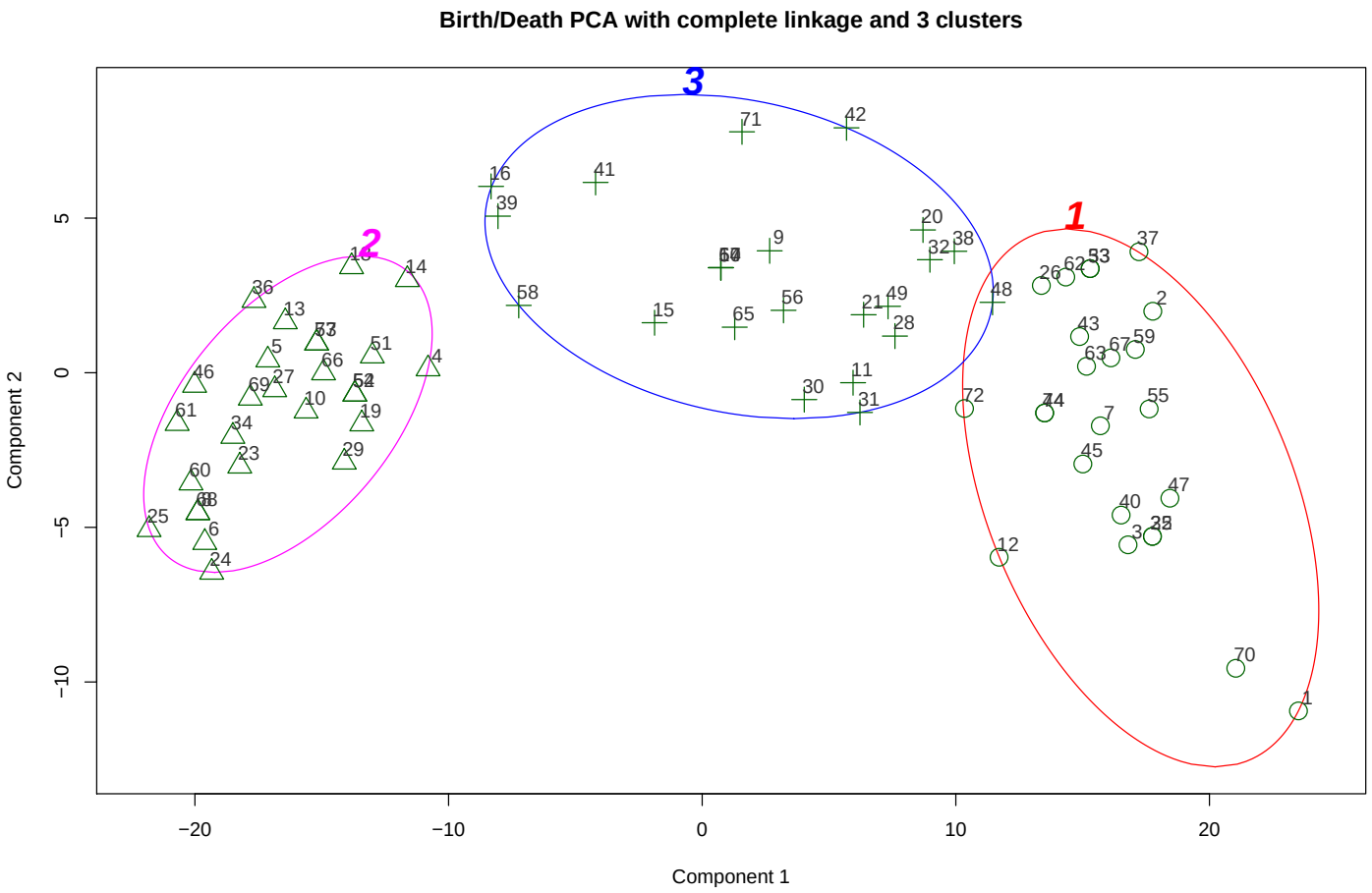


Let's try 3 clusters based on the dendrogram plots below. First we'll use complete linkage.

```
# create distance matrix between points
dist_bd76 <- dist(dat_bd76_num)

# number of clusters to identify with red boxes and ellipses
n_clus <- 3

# create dendrogram
```

```
hc_bd76_complete  <- hclust(dist_bd76, method = "complete")
plot(hc_bd76_complete, hang = -1
      , main = paste("Teeth with complete linkage and", n_clus, "clusters")
      , labels = dat_bd76$country)
rect.hclust(hc_bd76_complete, k = n_clus)

# create PCA scores plot with ellipses
clusplot(
    dat_bd76[,-1], cutree(hc_bd76_complete, k = n_clus)
  , color = TRUE, labels = 2, lines = 0
  , cex = 2, cex.txt = 1, col.txt = "gray20"
  , main = paste("Birth/Death PCA with complete linkage and", n_clus, "clusters")
  , sub = NULL)

# create a column with group membership
dat_bd76$cut_comp <- factor(cutree(hc_bd76_complete, k = n_clus))
```



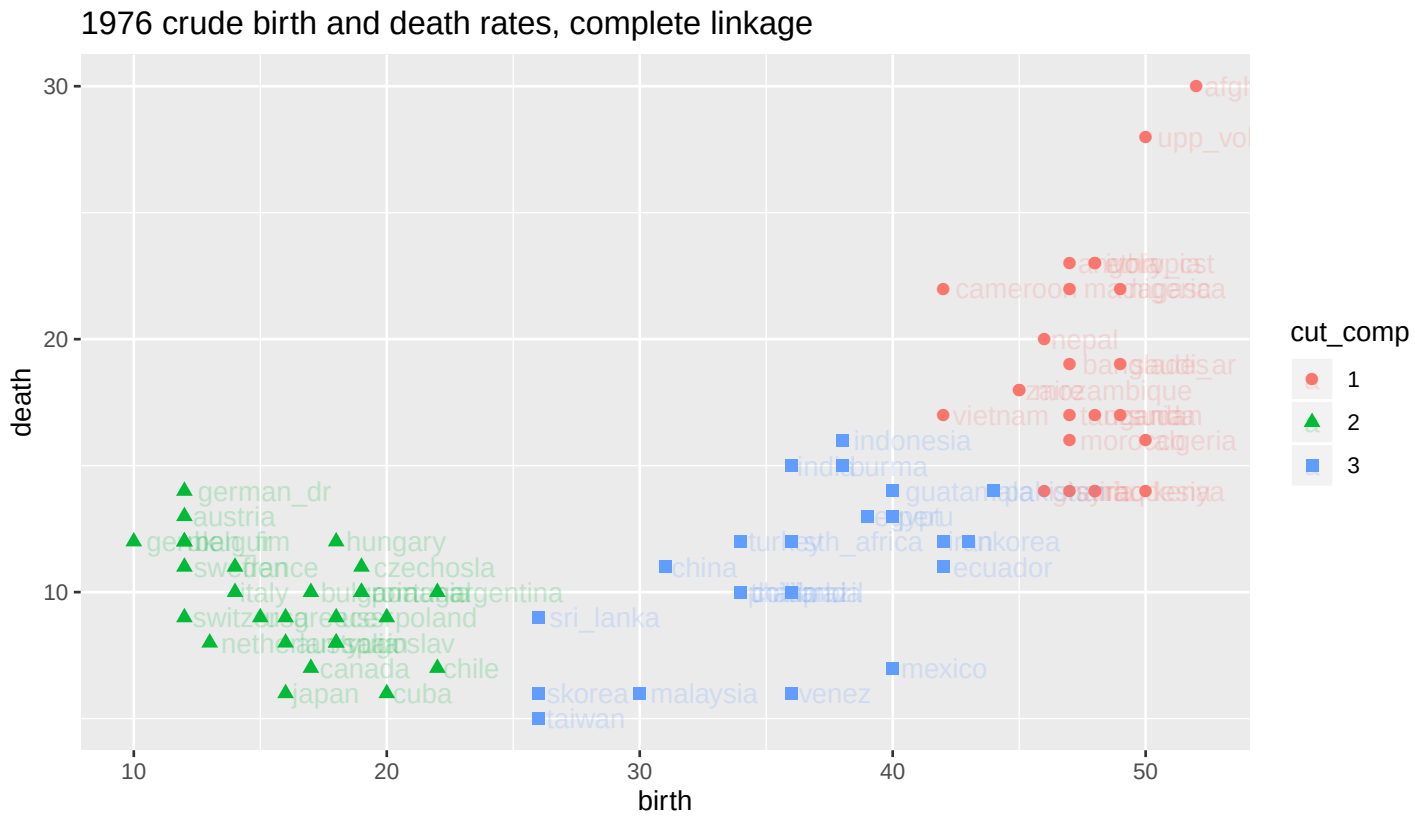



```
# print the observations in each cluster
for (i_cut in 1:n_clus) {
  print(paste("Cluster", i_cut, " ----------------------------- "))
  print(dat_bd76[(cutree(hc_bd76_complete, k = n_clus) == i_cut),], n = Inf)
}
## [1] "Cluster 1  ----------------------------- "
## # A tibble: 24 x 4
##    country    birth death cut_comp
##    <chr>      <dbl> <dbl> <fct>
##  1 afghan        52    30 1
##  2 algeria       50    16 1
##  3 angola        47    23 1
##  4 banglades     47    19 1
##  5 cameroon      42    22 1
##  6 ethiopia      48    23 1
##  7 ghana         46    14 1
##  8 iraq          48    14 1
##  9 ivory_cst     48    23 1
## 10 kenya         50    14 1
## 11 madagasca     47    22 1
## 12 morocco       47    16 1
## 13 mozambique    45    18 1
## 14 nepal         46    20 1
```

```
## 15 nigeria       49    22 1
## 16 rhodesia      48    14 1
## 17 saudi_ar      49    19 1
## 18 sudan         49    17 1
## 19 syria         47    14 1
## 20 tanzania      47    17 1
## 21 uganda        48    17 1
## 22 upp_volta     50    28 1
## 23 vietnam       42    17 1
## 24 zaire         45    18 1
## [1] "Cluster 2  ---------------------------- "
## # A tibble: 27 x 4
##    country   birth death cut_comp
##    <chr>     <dbl> <dbl> <fct>
##  1 argentina    22    10 2
##  2 australia    16     8 2
##  3 austria      12    13 2
##  4 belguim      12    12 2
##  5 bulgaria     17    10 2
##  6 canada       17     7 2
##  7 chile        22     7 2
##  8 cuba         20     6 2
##  9 czechosla    19    11 2
## 10 france       14    11 2
## 11 german_dr    12    14 2
## 12 german_fr    10    12 2
## 13 greece       16     9 2
## 14 hungary      18    12 2
## 15 italy        14    10 2
## 16 japan        16     6 2
## 17 netherlan    13     8 2
## 18 poland       20     9 2
## 19 portugal     19    10 2
## 20 romania      19    10 2
## 21 spain        18     8 2
## 22 sweden       12    11 2
## 23 switzer      12     9 2
## 24 ussr         18     9 2
## 25 uk           12    12 2
## 26 usa          15     9 2
## 27 yugoslav     18     8 2
## [1] "Cluster 3  ---------------------------- "
## # A tibble: 23 x 4
##    country    birth death cut_comp
##    <chr>      <dbl> <dbl> <fct>
##  1 brazil        36    10 3
##  2 burma         38    15 3
##  3 china         31    11 3
```

```
##  4 taiwan        26     5 3
##  5 columbia      34    10 3
##  6 ecuador       42    11 3
##  7 egypt         39    13 3
##  8 guatamala     40    14 3
##  9 india         36    15 3
## 10 indonesia     38    16 3
## 11 iran          42    12 3
## 12 nkorea        43    12 3
## 13 skorea        26     6 3
## 14 malaysia      30     6 3
## 15 mexico        40     7 3
## 16 pakistan      44    14 3
## 17 peru          40    13 3
## 18 phillip       34    10 3
## 19 sth_africa    36    12 3
## 20 sri_lanka     26     9 3
## 21 thailand      34    10 3
## 22 turkey        34    12 3
## 23 venez         36     6 3
```

```
# plot original data
library(ggplot2)
p1 <- ggplot(dat_bd76, aes(x = birth, y = death, colour = cut_comp, shape = cut_comp))
p1 <- p1 + geom_point(size = 2)                                          # points
p1 <- p1 + geom_text(aes(label = country), hjust = -0.1, alpha = 0.2)  # labels
p1 <- p1 + coord_fixed(ratio = 1) # makes 1 unit equal length on x- and y-axis
p1 <- p1 + labs(title = "1976 crude birth and death rates, complete linkage")
print(p1)
```



In very general/loose terms[3], it appears that at least some members of the "Four Asian Tigers[4]" are toward the bottom of the swoop, while the countries with more Euro-centric wealth are mostly clustered on the left side of the swoop, and many developing countries make up the steeper right side of the swoop. Perhaps the birth and death rates of a given country are influenced in part by the primary means by which the country has obtained wealth[5] (if it is considered a wealthy country). For example, the Four Asian Tigers have supposedly developed wealth in more recent years through export-driven economies, and the Tiger Cub Economies[6] are currently developing in a similar fashion[7].

[3]Thanks to Drew Enigk from Spring 2013 who provided this interpretation.

[4]`http://en.wikipedia.org/wiki/Four_Asian_Tigers`

[5]`http://www.povertyeducation.org/the-rise-of-asia.html`

[6]`http://en.wikipedia.org/wiki/Tiger_Cub_Economies`

[7]`http://www.investopedia.com/terms/t/tiger-cub-economies.asp`

14.4.2 Single linkage

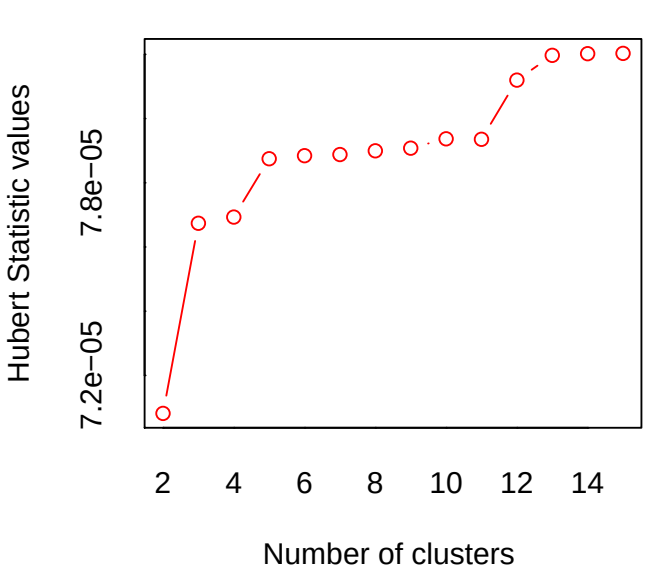
Now we'll use single linkage to compare.

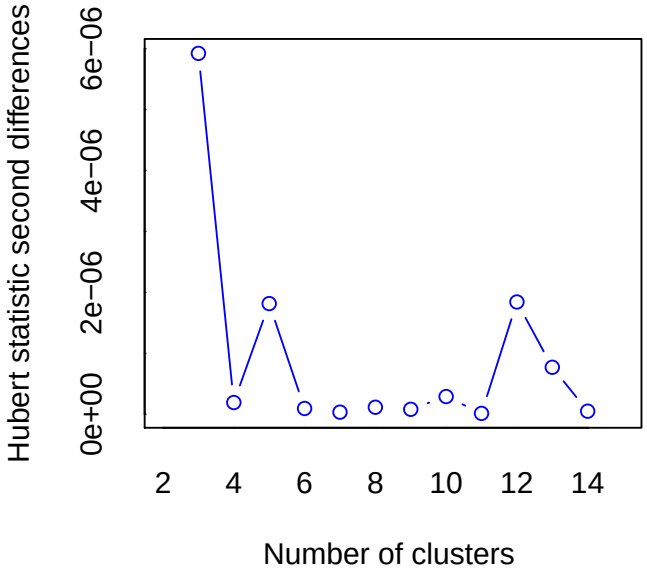
```
#library(NbClust)
#dat_bd76_num <- as.matrix(dat_bd76[,-1])  # use the previous dat_bd76_num
NC_out <- NbClust(dat_bd76_num, method = "single", index = "all")
```

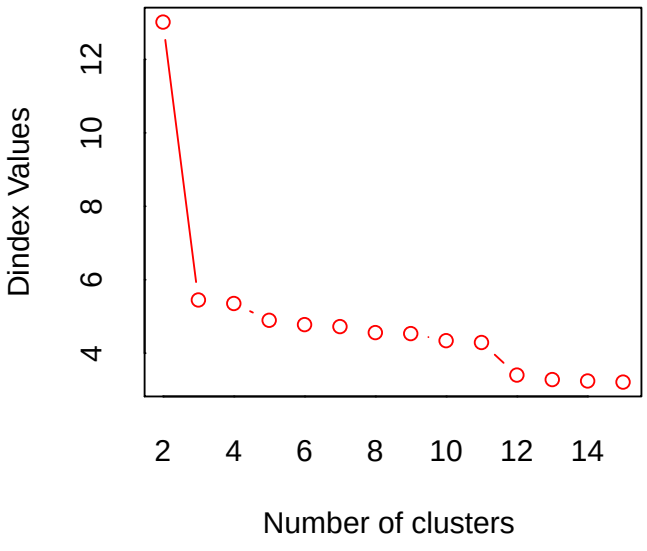
```
## Warning in pf(beale, pp, df2):  NaNs produced
## Warning in pf(beale, pp, df2):  NaNs produced
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##                 index second differences plot.
##
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 1 proposed 2 as the best number of clusters
## * 19 proposed 3 as the best number of clusters
## * 1 proposed 8 as the best number of clusters
## * 1 proposed 12 as the best number of clusters
## * 1 proposed 14 as the best number of clusters
##
##                    ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  3
##
##
## *******************************************************************
```

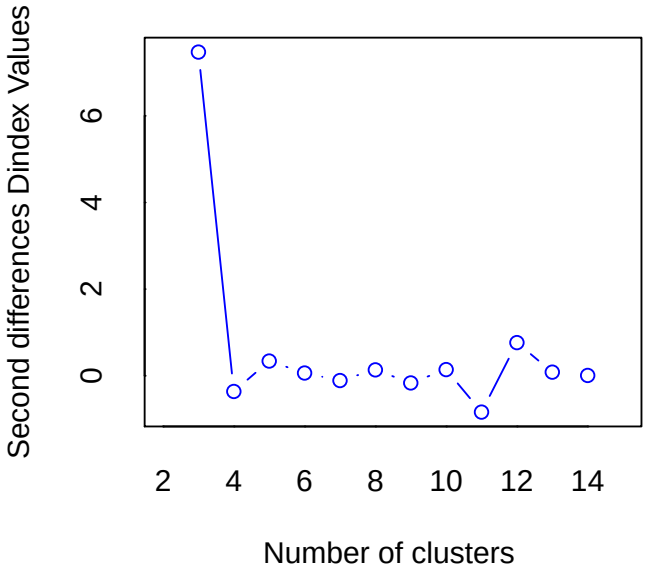
```
# most of the methods suggest 4 to 11 clusters, as do the plots
NC_out$Best.nc
```

```
##                       KL       CH Hartigan     CCC    Scott  Marriot   TrCovW
## Number_clusters 12.0000   3.0000   3.0000  3.0000   3.0000        3        3
## Value_Index     24.3966 162.9476 298.1745 10.0542 141.9607 39129554 51595106
##                   TraceW Friedman   Rubin Cindex     DB Silhouette   Duda
## Number_clusters     3.00   3.0000   3.000 8.0000 3.0000       3.00 3.0000
## Value_Index     11850.37  40.7738 -29.266 0.3414 0.5031       0.53 1.0769
##                 PseudoT2   Beale Ratkowsky     Ball PtBiserial Frey McClain
## Number_clusters   3.0000  3.0000    3.0000    3.000     3.0000    1  2.0000
## Value_Index      -2.7851 -0.0696    0.4826 6426.275     0.8127   NA  0.0367
##                   Dunn Hubert SDindex Dindex    SDbw
## Number_clusters 3.0000      0  3.0000      0 14.0000
## Value_Index     0.2417      0  0.2714      0  0.0121
```



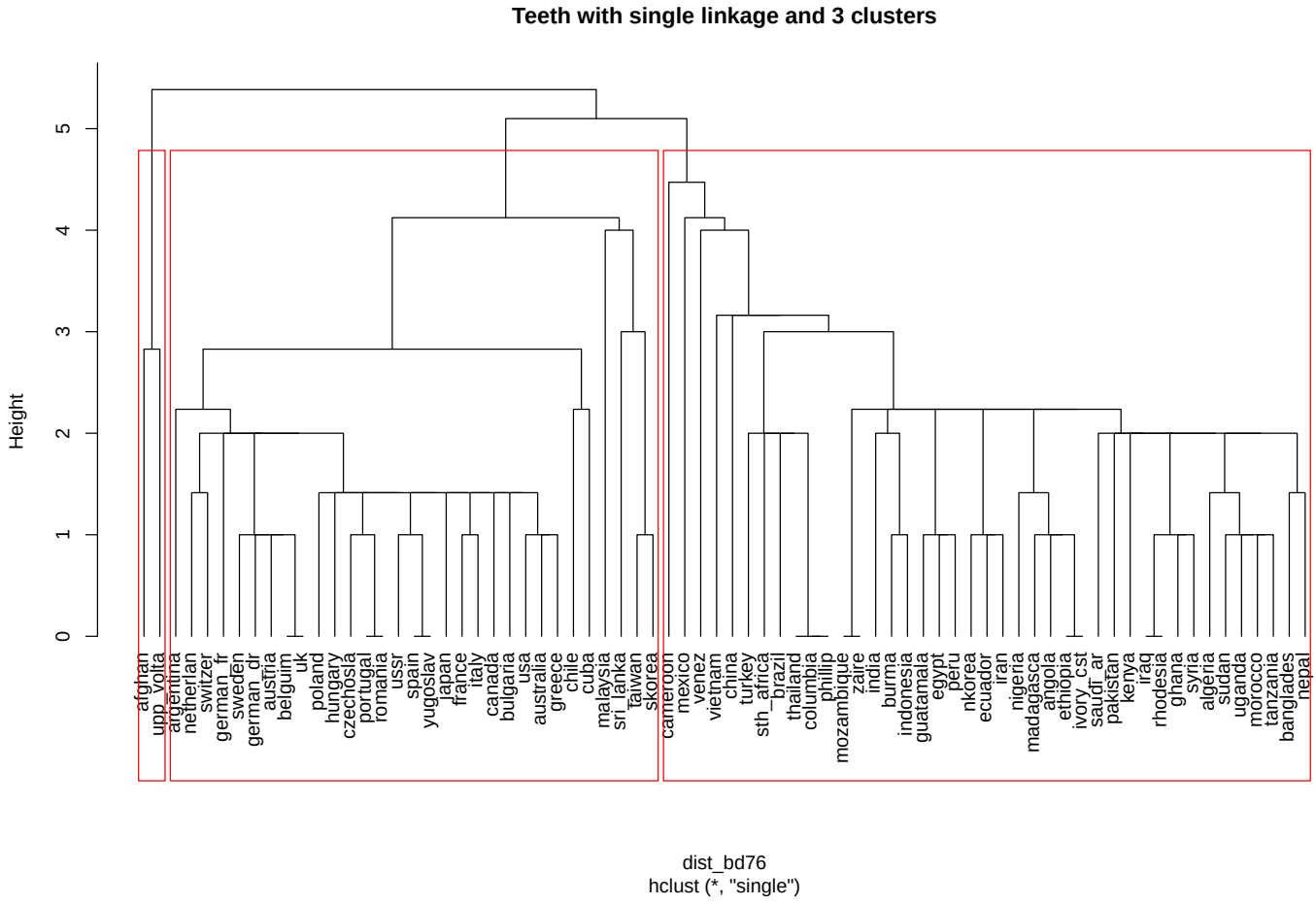


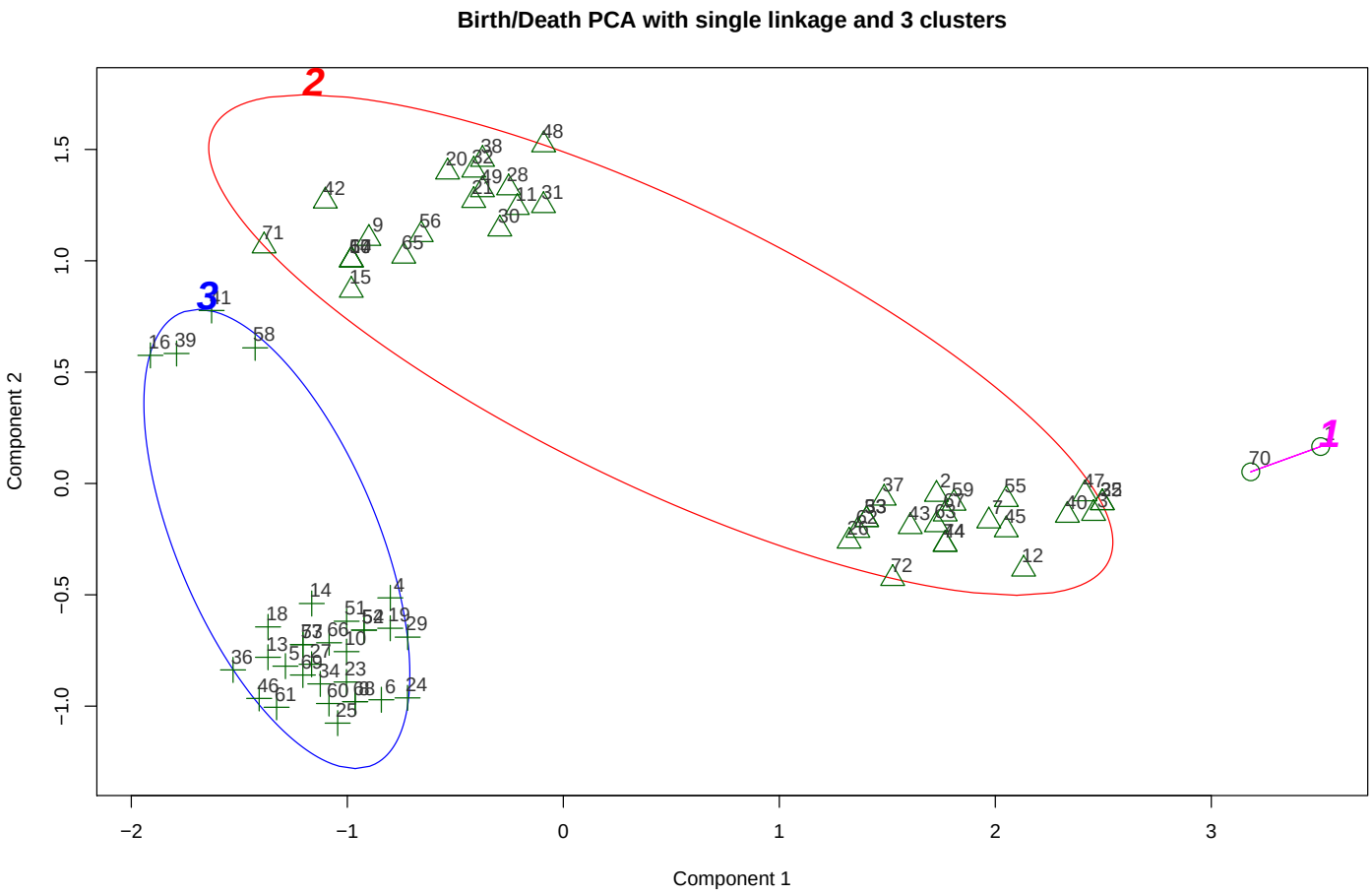





```
# create distance matrix between points
dist_bd76 <- dist(dat_bd76_num)

# number of clusters to identify with red boxes and ellipses
n_clus <- 3

# create dendrogram
hc_bd76_single  <- hclust(dist_bd76, method = "single")
plot(hc_bd76_single, hang = -1
      , main = paste("Teeth with single linkage and", n_clus, "clusters")
      , labels = dat_bd76$country)
rect.hclust(hc_bd76_single, k = n_clus)

# create PCA scores plot with ellipses
clusplot(dat_bd76[,-1], cutree(hc_bd76_single, k = n_clus)
        , color = TRUE, labels = 2, lines = 0
```

```
        , cex = 2, cex.txt = 1, col.txt = "gray20"
        , main = paste("Birth/Death PCA with single linkage and", n_clus, "clusters")
        , sub = NULL)

# create a column with group membership
dat_bd76$cut_sing <- factor(cutree(hc_bd76_single, k = n_clus))
```





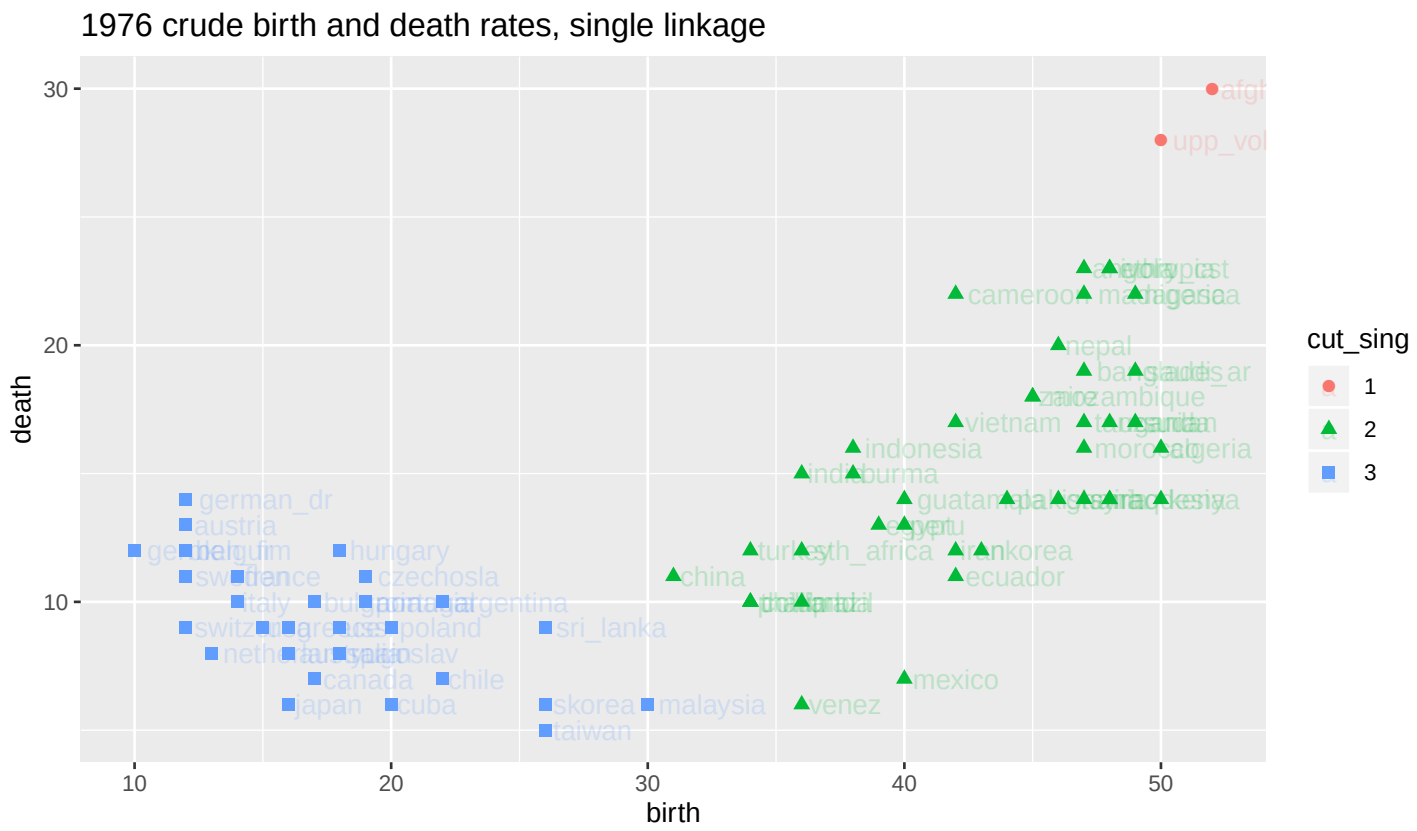

```
# print the observations in each cluster
for (i_cut in 1:n_clus) {
  print(paste("Cluster", i_cut, " ----------------------------- "))
  print(dat_bd76[(cutree(hc_bd76_single, k = n_clus) == i_cut),], n = Inf)
}
```

```
## [1] "Cluster 1  ----------------------------- "
## # A tibble: 2 x 5
##   country   birth death cut_comp cut_sing
##   <chr>     <dbl> <dbl> <fct>    <fct>
## 1 afghan       52    30 1        1
## 2 upp_volta    50    28 1        1
## [1] "Cluster 2  ----------------------------- "
## # A tibble: 41 x 5
##    country    birth death cut_comp cut_sing
##    <chr>      <dbl> <dbl> <fct>    <fct>
##  1 algeria       50    16 1        2
##  2 angola        47    23 1        2
##  3 banglades     47    19 1        2
##  4 brazil        36    10 3        2
##  5 burma         38    15 3        2
##  6 cameroon      42    22 1        2
##  7 china         31    11 3        2
##  8 columbia      34    10 3        2
```

```
##  9 ecuador       42    11 3        2
## 10 egypt         39    13 3        2
## 11 ethiopia      48    23 1        2
## 12 ghana         46    14 1        2
## 13 guatamala     40    14 3        2
## 14 india         36    15 3        2
## 15 indonesia     38    16 3        2
## 16 iran          42    12 3        2
## 17 iraq          48    14 1        2
## 18 ivory_cst     48    23 1        2
## 19 kenya         50    14 1        2
## 20 nkorea        43    12 3        2
## 21 madagasca     47    22 1        2
## 22 mexico        40     7 3        2
## 23 morocco       47    16 1        2
## 24 mozambique    45    18 1        2
## 25 nepal         46    20 1        2
## 26 nigeria       49    22 1        2
## 27 pakistan      44    14 3        2
## 28 peru          40    13 3        2
## 29 phillip       34    10 3        2
## 30 rhodesia      48    14 1        2
## 31 saudi_ar      49    19 1        2
## 32 sth_africa    36    12 3        2
## 33 sudan         49    17 1        2
## 34 syria         47    14 1        2
## 35 tanzania      47    17 1        2
## 36 thailand      34    10 3        2
## 37 turkey        34    12 3        2
## 38 uganda        48    17 1        2
## 39 venez         36     6 3        2
## 40 vietnam       42    17 1        2
## 41 zaire         45    18 1        2
## [1] "Cluster 3  ---------------------------- "
## # A tibble: 31 x 5
##    country   birth death cut_comp cut_sing
##    <chr>     <dbl> <dbl> <fct>    <fct>
##  1 argentina    22    10 2        3
##  2 australia    16     8 2        3
##  3 austria      12    13 2        3
##  4 belguim      12    12 2        3
##  5 bulgaria     17    10 2        3
##  6 canada       17     7 2        3
##  7 chile        22     7 2        3
##  8 taiwan       26     5 3        3
##  9 cuba         20     6 2        3
## 10 czechosla    19    11 2        3
## 11 france       14    11 2        3
```

```
## 12 german_dr    12    14 2        3
## 13 german_fr    10    12 2        3
## 14 greece       16     9 2        3
## 15 hungary      18    12 2        3
## 16 italy        14    10 2        3
## 17 japan        16     6 2        3
## 18 skorea       26     6 3        3
## 19 malaysia     30     6 3        3
## 20 netherlan    13     8 2        3
## 21 poland       20     9 2        3
## 22 portugal     19    10 2        3
## 23 romania      19    10 2        3
## 24 spain        18     8 2        3
## 25 sri_lanka    26     9 3        3
## 26 sweden       12    11 2        3
## 27 switzer      12     9 2        3
## 28 ussr         18     9 2        3
## 29 uk           12    12 2        3
## 30 usa          15     9 2        3
## 31 yugoslav     18     8 2        3
```

```r
# plot original data
library(ggplot2)
p1 <- ggplot(dat_bd76, aes(x = birth, y = death, colour = cut_sing, shape = cut_sing))
p1 <- p1 + geom_point(size = 2)                                          # points
p1 <- p1 + geom_text(aes(label = country), hjust = -0.1, alpha = 0.2)    # labels
p1 <- p1 + coord_fixed(ratio = 1) # makes 1 unit equal length on x- and y-axis
p1 <- p1 + labs(title = "1976 crude birth and death rates, single linkage")
print(p1)
```



The two methods suggest three clusters. Complete linkage also suggests 14 clusters, but the clusters were unappealing so this analysis will not be presented here.

The three clusters generated by the two methods are very different. The same tendency was observed using average linkage and Ward's method.

An important point to recognize is that different clustering algorithms may agree on the number of clusters, but they may not agree on the composition of the clusters.