Chapter 14

Cluster Analysis

14.1 Introduction

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 Illustration

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.

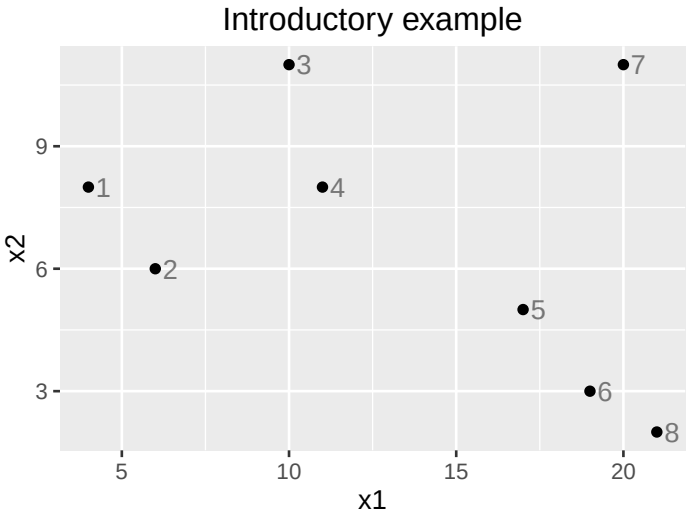
Finally Use a tree or dendrogram to summarize the steps in the cluster formation.

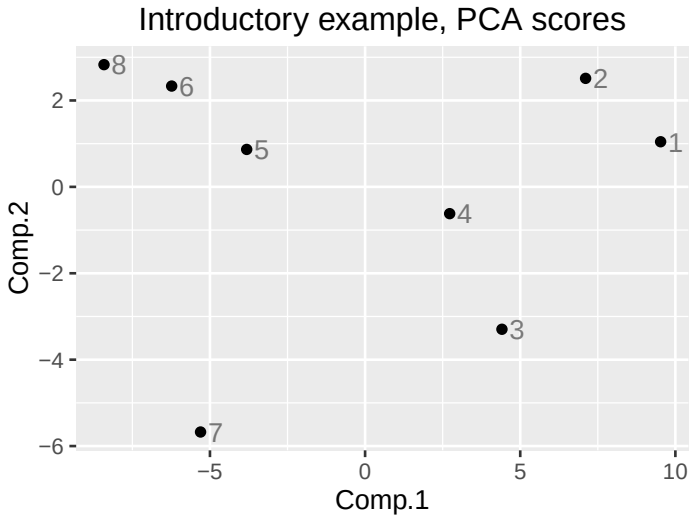
```
#### Example: Fake data cluster illustration
# convert to a data.frame by reading the text table
intro <- read.table(text = "
x1 x2
 4  8
 6  6
10 11
11  8
17  5
19  3
20 11
21  2
", header=TRUE)
str(intro)
```

```
## '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
intro.pca <- princomp( ~ x1 + x2, data = intro)


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

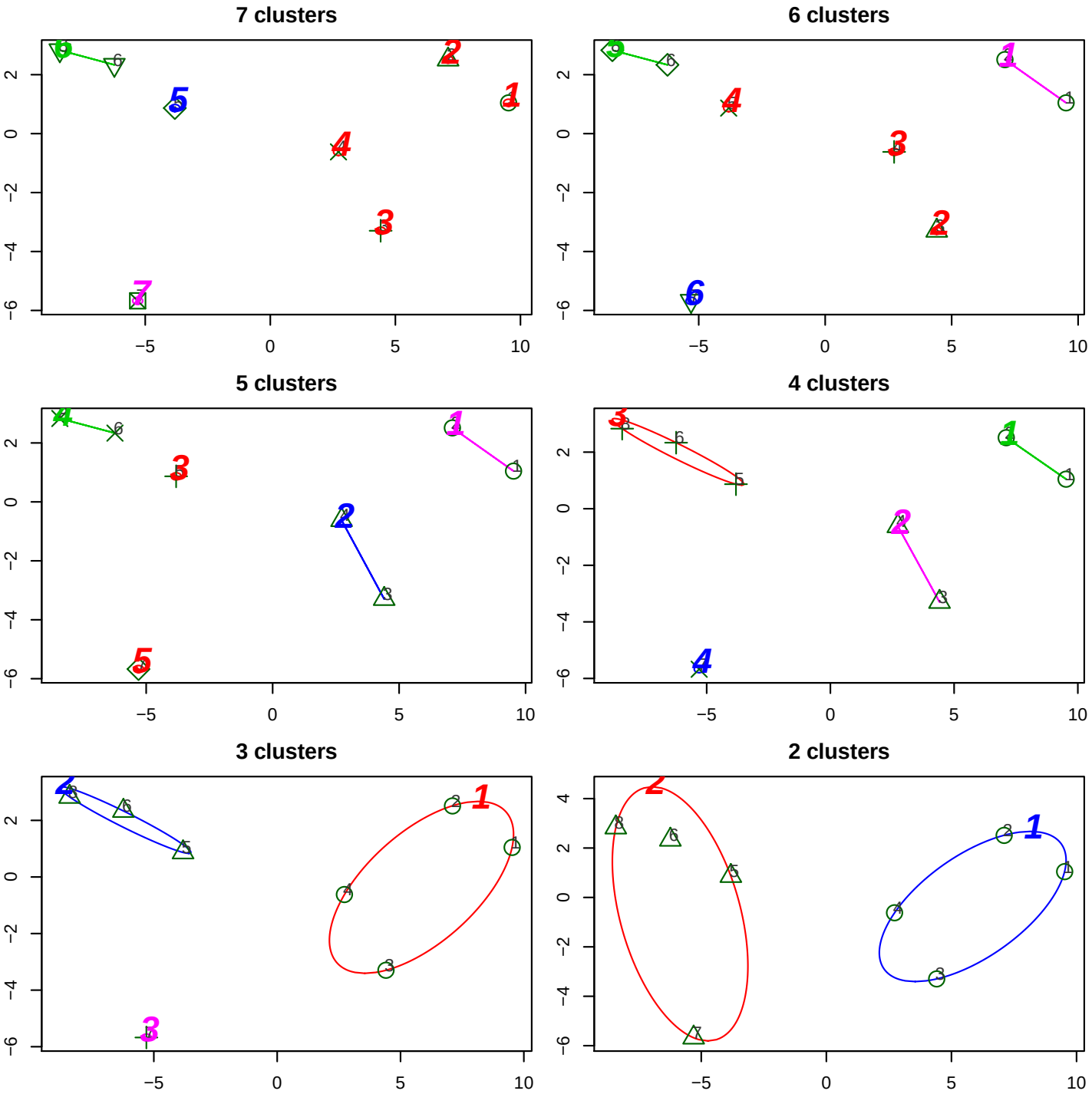
```
# plot PCA scores (data on PC-scale centered at 0)
library(ggplot2)
p2 <- ggplot(as.data.frame(intro.pca$scores), aes(x = Comp.1, y = Comp.2))
p2 <- p2 + geom_point()            # points
p2 <- p2 + geom_text(aes(label = rownames(intro.pca$scores)), hjust = -0.5, alpha = 0.5) # lab
p2 <- p2 + labs(title = "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
intro.dist <- dist(intro)
intro.hc.average  <- hclust(intro.dist, method = "average")

    op <- par(no.readonly = TRUE) # save original plot options
    par(mfrow = c(3,2), mar = c(2, 2, 2.5, 1))  # margins are c(bottom, left, top, right)

library(cluster)
for (i.clus in 7:2) {
  clusplot(intro, cutree(intro.hc.average, k = i.clus)
           , color = TRUE, labels = 2, lines = 0
           , cex = 2, cex.txt = 1, col.txt = "gray20"
           , main = paste(i.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[2].

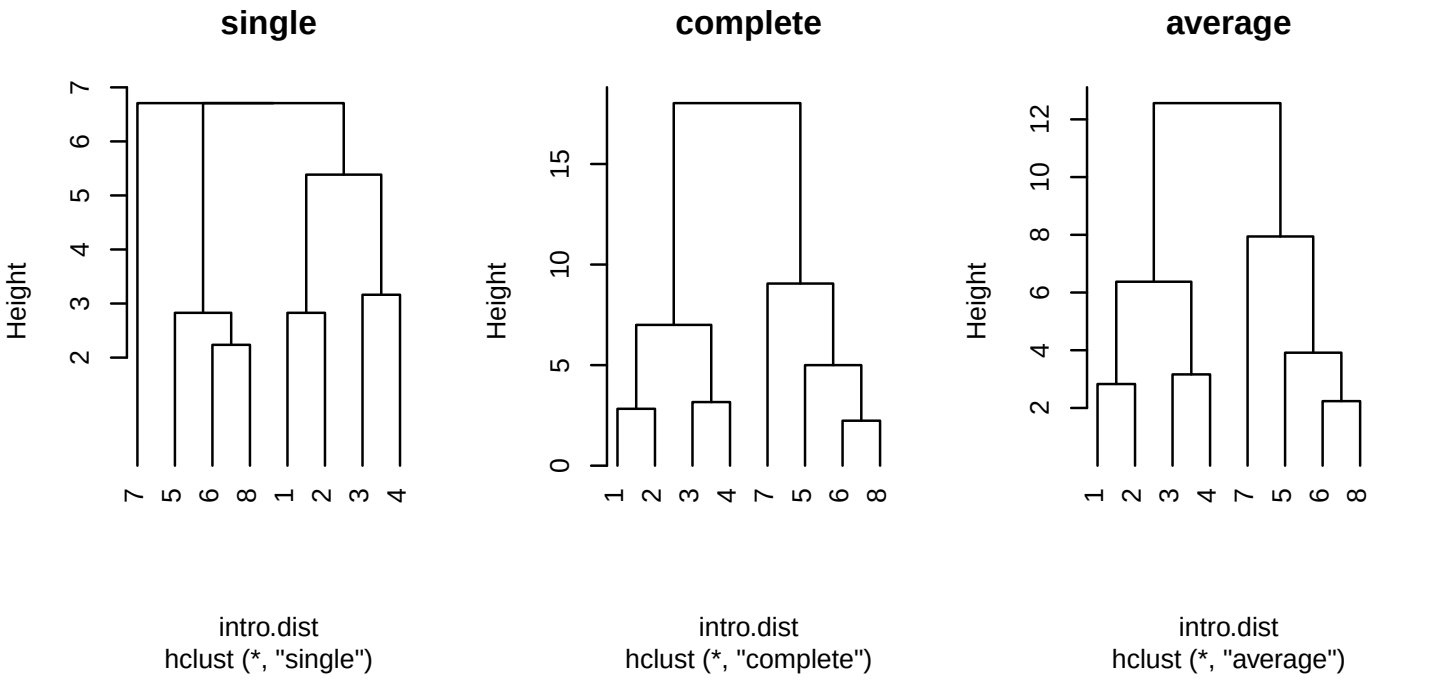
```
# create distance matrix between points
intro.dist <- dist(intro)
intro.dist
```

```
##          1        2        3        4        5        6
```

[2]There are many ways to create dendrograms in R, see `http://gastonsanchez.com/blog/how-to/2012/10/03/Dendrograms.html` for several examples.

```
## 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
##           7
## 2
## 3
## 4
## 5
## 6
## 7
## 8  9.055385
```

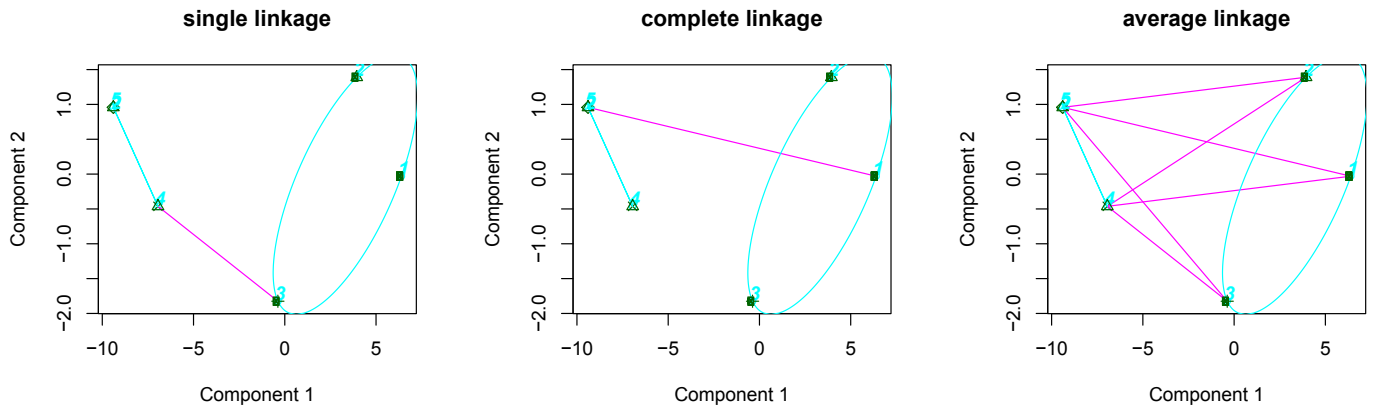
```
    op <- par(no.readonly = TRUE) # save original plot options
    par(mfrow = c(1,3))  # margins are c(bottom, left, top, right)

intro.hc.single   <- hclust(intro.dist, method = "single")
# note: plotting used to use plclust()
plot(intro.hc.single, hang = -1,       main = "single")

intro.hc.complete <- hclust(intro.dist, method = "complete")
plot(intro.hc.complete, hang = -1,     main = "complete")

intro.hc.average  <- hclust(intro.dist, method = "average")
plot(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.

You should 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
```

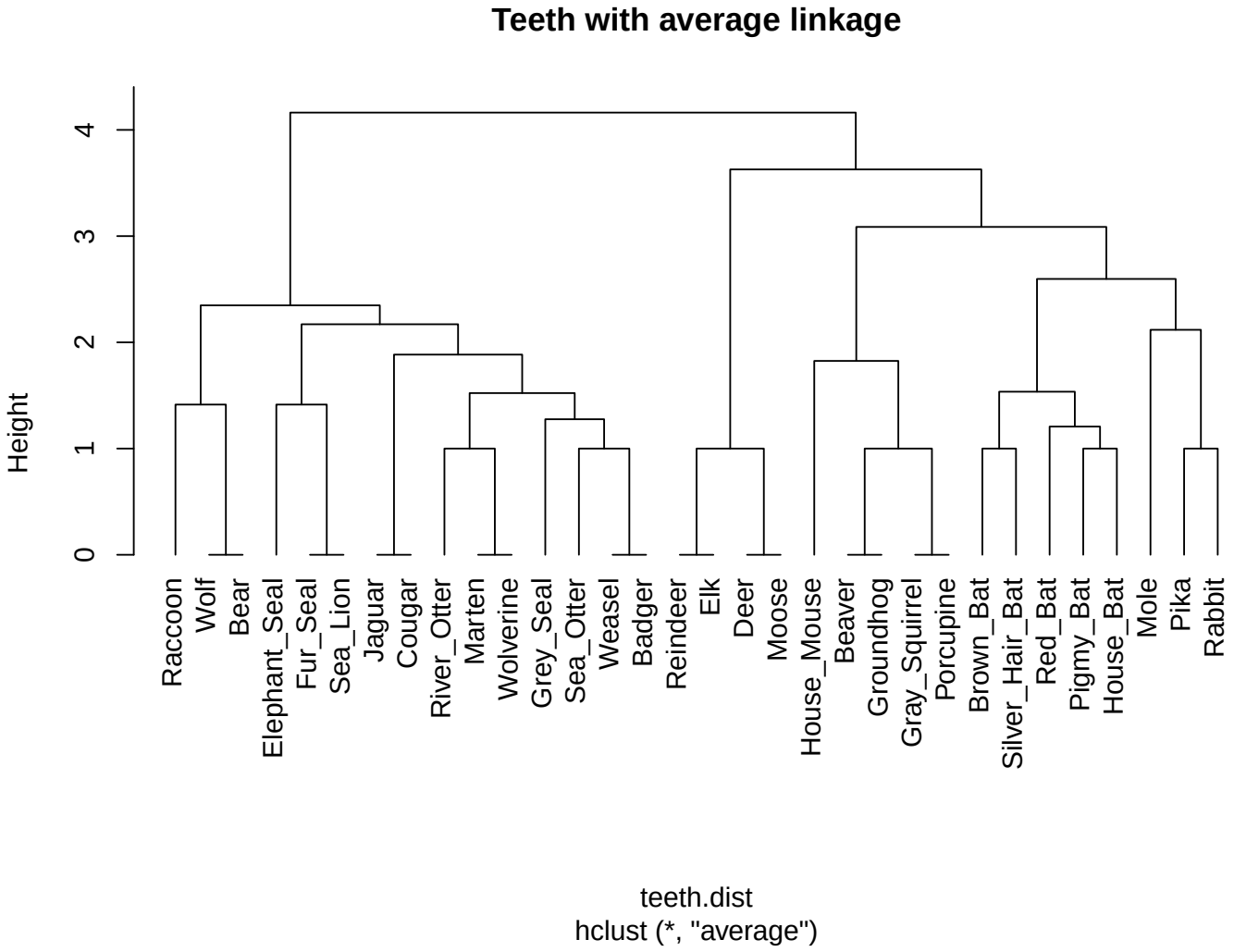
```
# v1 = top incisors
# v2 = bottom incisors
# v3 = top canines
# v4 = bottom canines
# v5 = top premolars
# v6 = bottom premolars
# v7 = top molars
# v8 = bottom molars

fn.data <- "http://statacumen.com/teach/ADA2/ADA2_notes_Ch14_teeth.dat"
teeth <- read.table(fn.data, header = TRUE)
str(teeth)
```

```
## 'data.frame': 32 obs. of  9 variables:
##  $ mammal: Factor w/ 32 levels "Badger","Bear",..: 4 17 29 19 13 24 20 22 3 12 ...
##  $ v1    : int  2 3 2 2 2 1 2 2 1 1 ...
##  $ v2    : int  3 2 3 3 3 3 1 1 1 1 ...
##  $ v3    : int  1 1 1 1 1 1 0 0 0 0 ...
##  $ v4    : int  1 0 1 1 1 1 0 0 0 0 ...
##  $ v5    : int  3 3 2 2 1 2 2 3 2 2 ...
##  $ v6    : int  3 3 3 2 2 2 2 2 1 1 ...
##  $ v7    : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ v8    : int  3 3 3 3 3 3 3 3 3 3 ...
```

	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
teeth.dist <- dist(teeth[,-1])

# number of clusters to identify with red boxes and ellipses
# i.clus <- 8

# create dendrogram
teeth.hc.average  <- hclust(teeth.dist, method = "average")
plot(teeth.hc.average, hang = -1
      , main = paste("Teeth with average linkage") # and", i.clus, "clusters")
      , labels = teeth[,1])
# rect.hclust(teeth.hc.average, k = i.clus)

# # create PCA scores plot with ellipses
# clusplot(teeth, cutree(teeth.hc.average, k = i.clus)
#          , color = TRUE, labels = 2, lines = 0
#          , cex = 2, cex.txt = 1, col.txt = "gray20"
#          , main = paste("Teeth PCA with average linkage and", i.clus, "clusters")
#          , sub = NULL)
```



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[3] 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.

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

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

```
## 'data.frame': 32 obs. of  9 variables:
##  $ mammal: Factor w/ 32 levels "Badger","Bear",..: 4 17 29 19 13 24 20 22 3 12 ...
##  $ v1    : int  2 3 2 2 2 1 2 2 1 1 ...
##  $ v2    : int  3 2 3 3 3 3 1 1 1 1 ...
##  $ v3    : int  1 1 1 1 1 1 0 0 0 0 ...
##  $ v4    : int  1 0 1 1 1 1 0 0 0 0 ...
##  $ v5    : int  3 3 2 2 1 2 2 3 2 2 ...
##  $ v6    : int  3 3 3 2 2 2 2 2 1 1 ...
##  $ v7    : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ v8    : int  3 3 3 3 3 3 3 3 3 3 ...
```

```r
# Because the data type is "int" for integer, the routine fails (error expected)
NbClust(teeth[,-1], method = "average", index = "all")
```

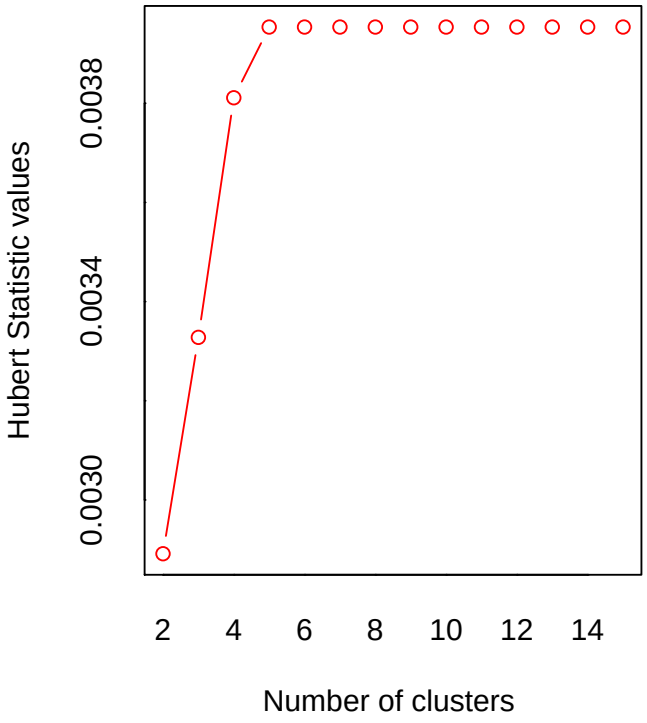
```
## Error in solve.default(W): system is computationally singular:  reciprocal condition number
= 1.51394e-16
```

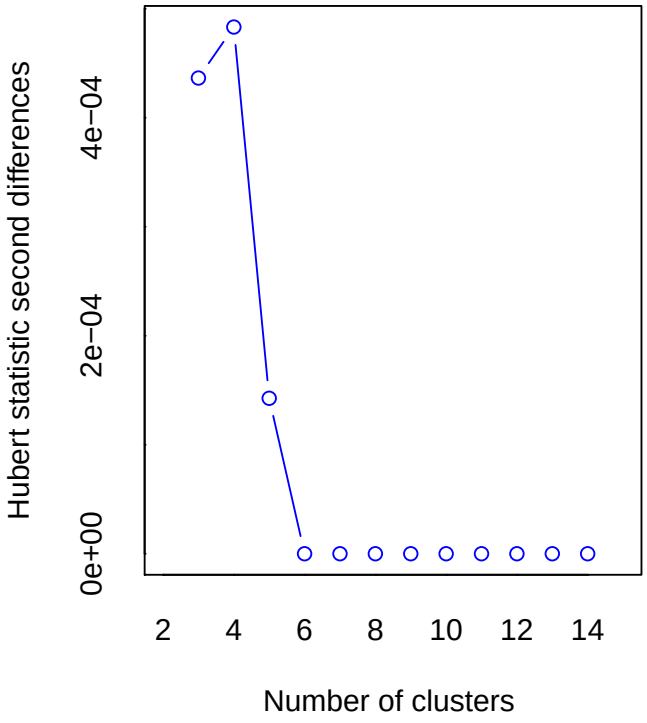
```r
# However, change the data type from integer to numeric and it works just fine!
teeth.num <- as.numeric(as.matrix(teeth[,-1]))
NC.out <- NbClust(teeth.num, method = "average", index = "all")
```

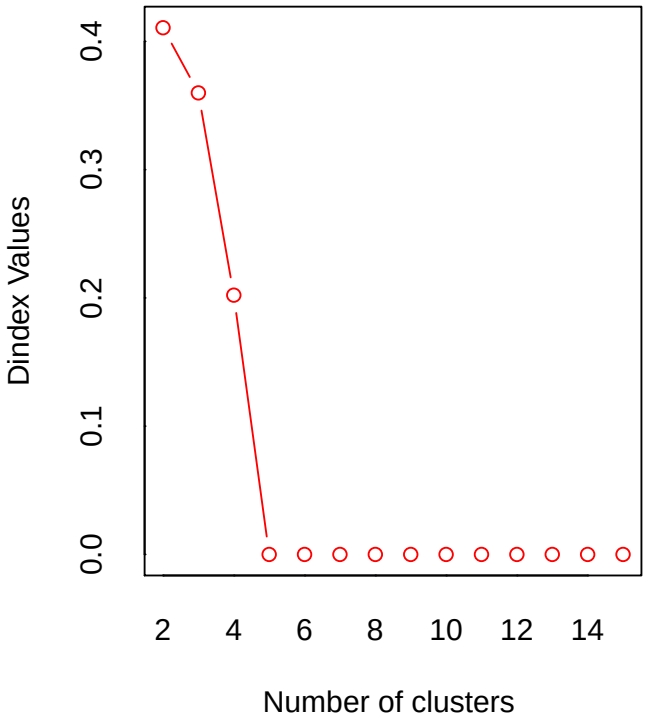
```
## Warning in max(DiffLev[, 5], na.rm = TRUE): no non-missing arguments to max; returning
-Inf
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to
##                 significant 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]
## *******************************************************************
## * 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
##
##
## *******************************************************************
```

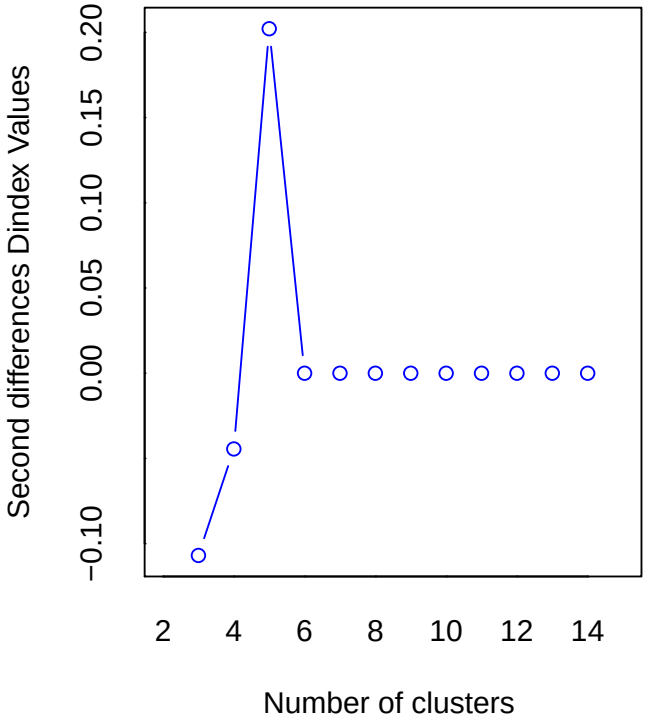
```
# most of the methods suggest 4 or 5 clusters, as do the plots
NC.out$Best.nc
```

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

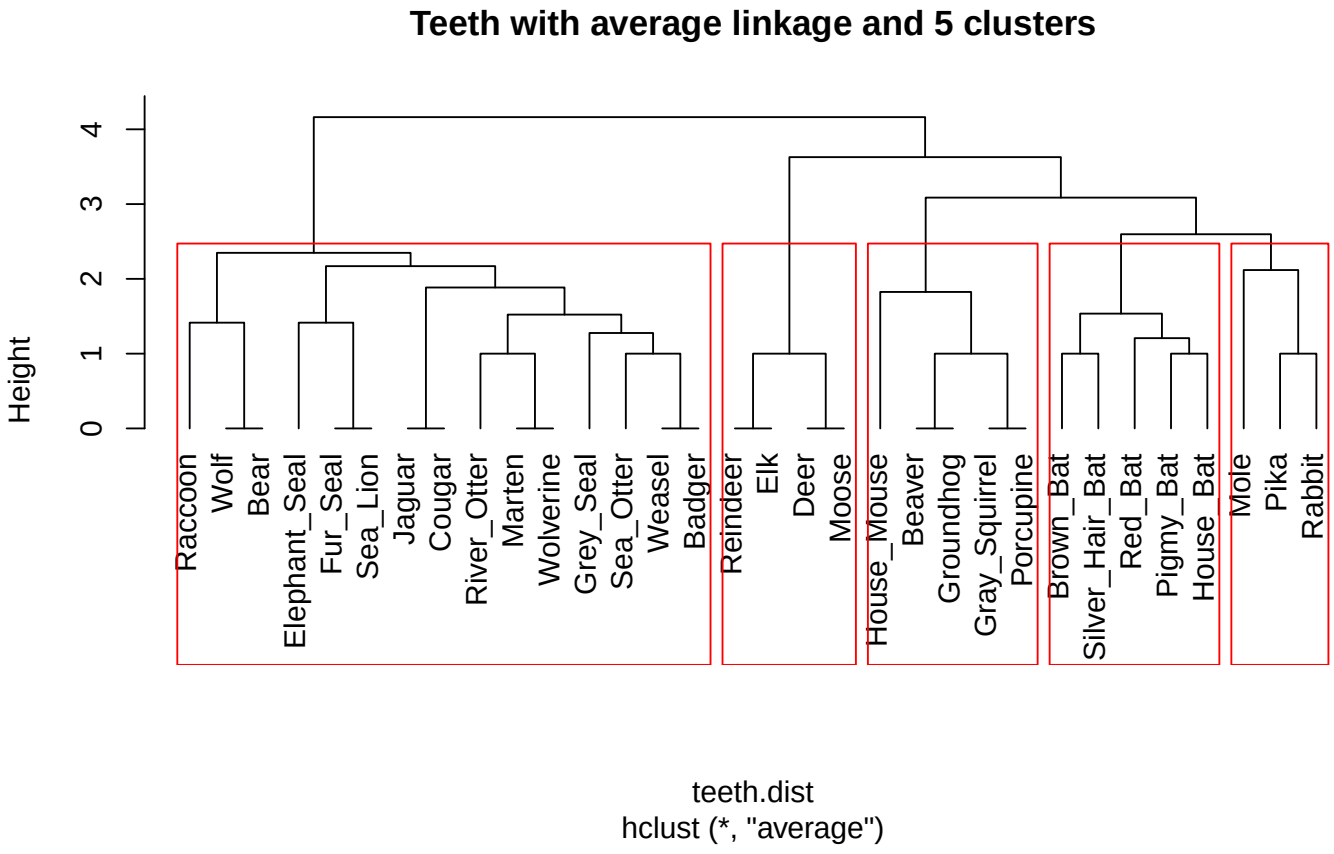
The RSQ summary is also useful for determining the number of clusters. RSQ is a pseudo-R^2 statistic that measures the proportion of the total variation explained by the differences among the existing clusters at a given step. RSQ will typically decrease as the pseudo-F statistic decreases.

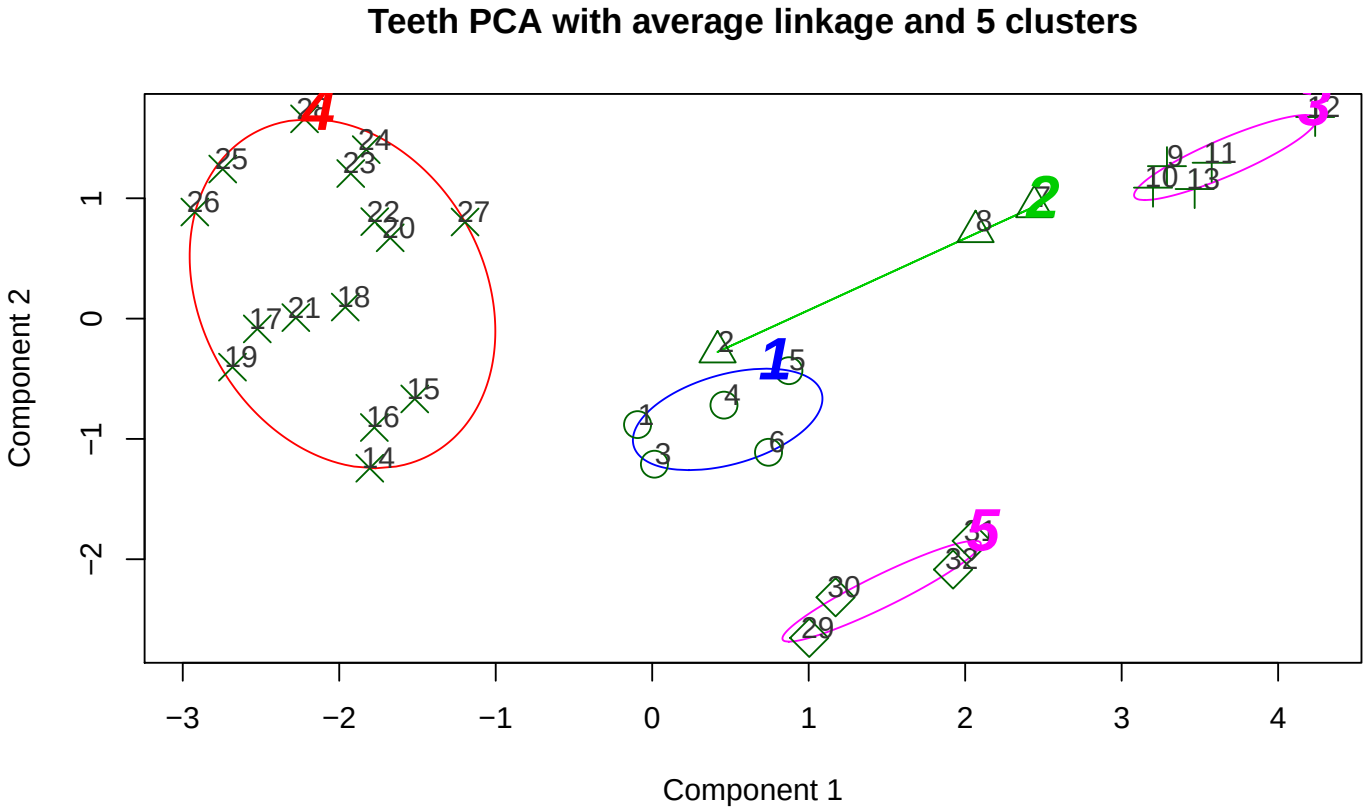
A common recommendation on cluster selection is to choose a cluster size where the values of ccc and the pseudo-F statistic are relatively high (compared to what you observe with other numbers of clusters), and where the pseudo-t statistic is relatively low and increases substantially at the next proposed 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
teeth.dist <- dist(teeth[,-1])

# number of clusters to identify with red boxes and ellipses
i.clus <- 5

# create dendrogram
teeth.hc.average  <- hclust(teeth.dist, method = "average")
plot(teeth.hc.average, hang = -1
      , main = paste("Teeth with average linkage and", i.clus, "clusters")
      , labels = teeth[,1])
rect.hclust(teeth.hc.average, k = i.clus)
```

```
# create PCA scores plot with ellipses
clusplot(teeth, cutree(teeth.hc.average, k = i.clus)
         , color = TRUE, labels = 2, lines = 0
         , cex = 2, cex.txt = 1, col.txt = "gray20"
         , main = paste("Teeth PCA with average linkage and", i.clus, "clusters")
         , sub = NULL)
```






```
# print the observations in each cluster
for (i.cut in 1:i.clus) {
  print(paste("Cluster", i.cut, " ---------------------------- "))
  print(teeth[(cutree(teeth.hc.average, k = i.clus) == i.cut),])
}
```

```
## [1] "Cluster 1  ---------------------------- "
##            mammal v1 v2 v3 v4 v5 v6 v7 v8
## 1       Brown_Bat  2  3  1  1  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
## [1] "Cluster 2  ---------------------------- "
##   mammal v1 v2 v3 v4 v5 v6 v7 v8
## 2   Mole  3  2  1  0  3  3  3  3
## 7   Pika  2  1  0  0  2  2  3  3
## 8 Rabbit  2  1  0  0  3  2  3  3
## [1] "Cluster 3  ---------------------------- "
##           mammal v1 v2 v3 v4 v5 v6 v7 v8
## 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
```

```
## [1] "Cluster 4  ---------------------------- "
##           mammal v1 v2 v3 v4 v5 v6 v7 v8
## 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
## [1] "Cluster 5  ---------------------------- "
##      mammal v1 v2 v3 v4 v5 v6 v7 v8
## 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
```

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
fn.data <- "http://statacumen.com/teach/ADA2/ADA2_notes_Ch14_birthdeath.dat"
bd <- read.table(fn.data, header = TRUE)
str(bd)
```

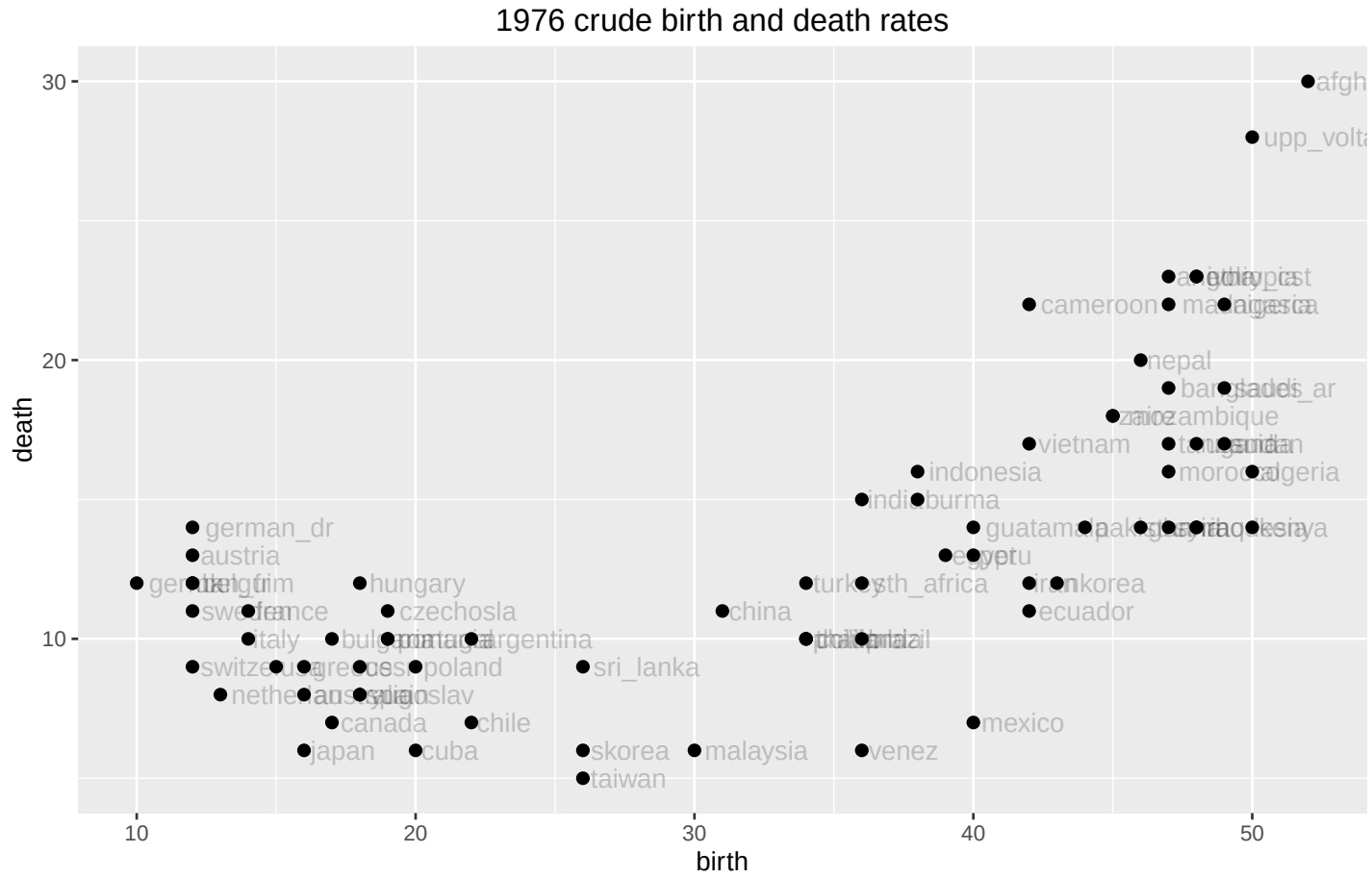
```
## 'data.frame': 74 obs. of  3 variables:
##  $ country: Factor w/ 74 levels "afghan","algeria",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ birth  : int  52 50 47 22 16 12 47 12 36 17 ...
##  $ death  : int  30 16 23 10 8 13 19 12 10 10 ...
```

	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(bd, 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)
```

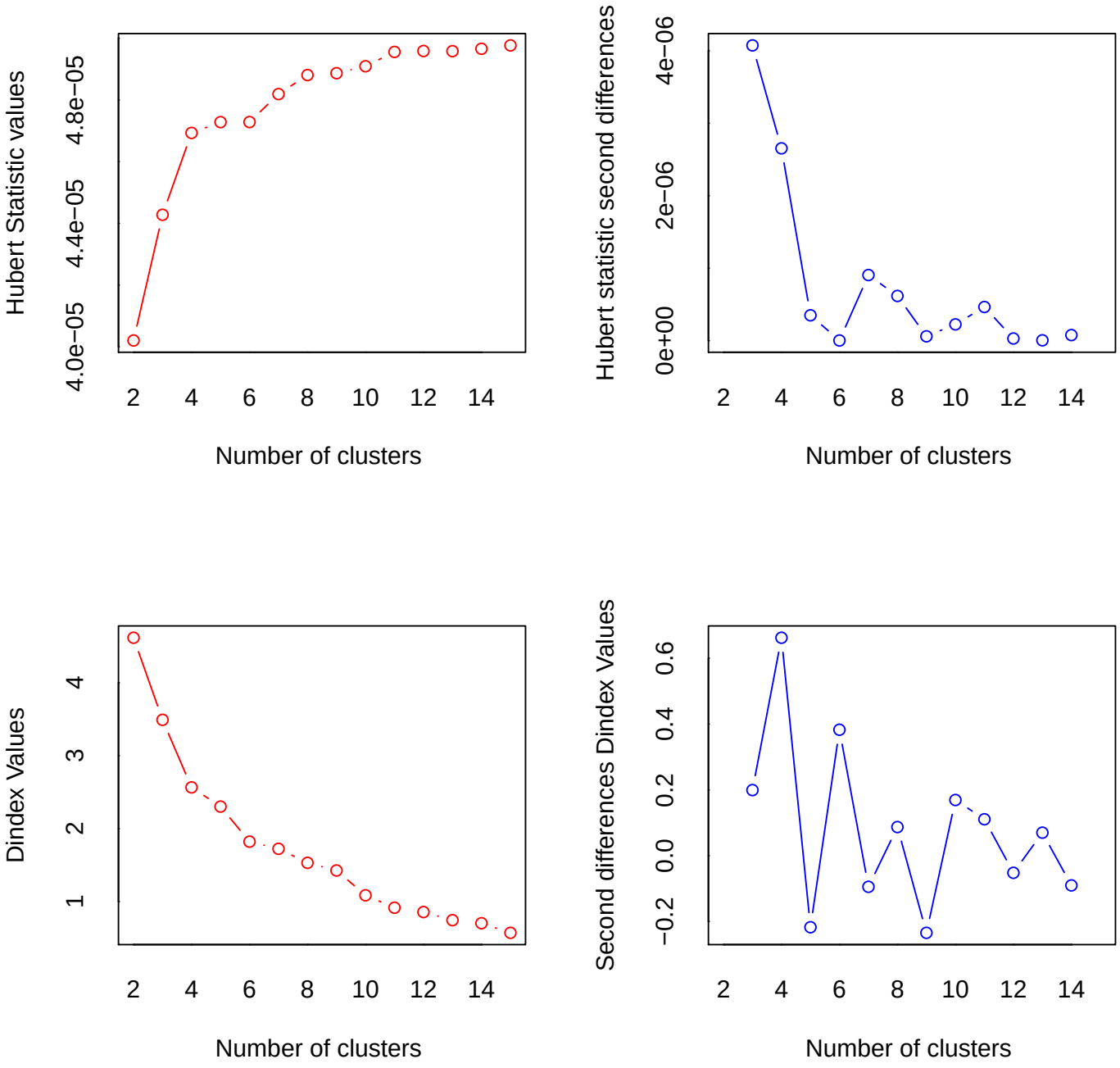


14.4.1 Complete linkage

```
library(NbClust)
# Change integer data type to numeric
bd.num <- as.numeric(as.matrix(bd[,-1]))
NC.out <- NbClust(bd.num, method = "complete", 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.
```

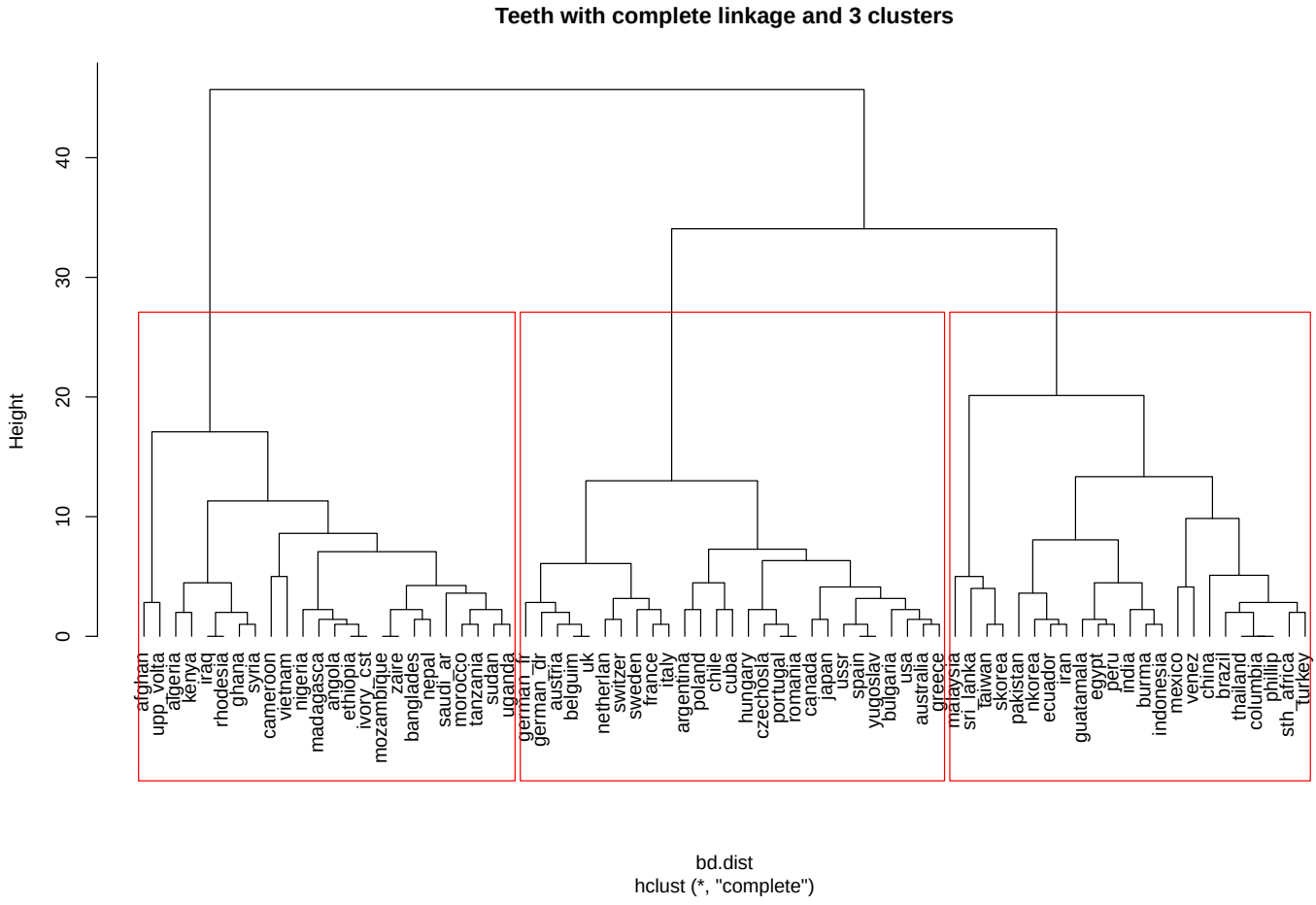
```
##
## Warning in matrix(c(results), nrow = 2, ncol = 26):  data length [51] is not a sub-multiple
or multiple of the number of rows [2]
## Warning in matrix(c(results), nrow = 2, ncol = 26, dimnames = list(c("Number_clusters",
:  data length [51] is not a sub-multiple or multiple of the number of rows [2]
## *****************************************************************
## * Among all indices:
## * 2 proposed 2 as the best number of clusters
## * 1 proposed 4 as the best number of clusters
## * 1 proposed 5 as the best number of clusters
## * 1 proposed 6 as the best number of clusters
## * 1 proposed 15 as the best number of clusters
##
##                    ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  2
##
##
## *****************************************************************
# most of the methods suggest 2 to 6 clusters, as do the plots
NC.out$Best.nc
##                     KL       CH Hartigan     CCC   Scott  Marriot
## Number_clusters 2.000   15.000   5.0000  2.0000  4.0000    6.000
## Value_Index     3.333 1780.714 209.2456 20.7606 86.7855 9041.261
##                 TrCovW   TraceW Friedman     Rubin Cindex     DB
## Number_clusters   -Inf 854.6162  395.428 -131.4723 0.2254 0.4292
## Value_Index          4  15.0000   13.000    2.0000 2.0000 2.0000
##                 Silhouette   Duda PseudoT2  Beale Ratkowsky     Ball
## Number_clusters     0.7468 0.2486 142.0413 0.9864    0.4628 5166.333
## Value_Index         2.0000 2.0000   2.0000 3.0000    3.0000    2.000
##                 PtBiserial   Frey McClain   Dunn Hubert SDindex
## Number_clusters     0.8512 3.5386  0.1705 0.3333      0  0.3167
## Value_Index         3.0000 2.0000 13.0000 0.0000      3  0.0000
##                 Dindex   SDbw
## Number_clusters      0 0.0073
## Value_Index         15 2.0000
```

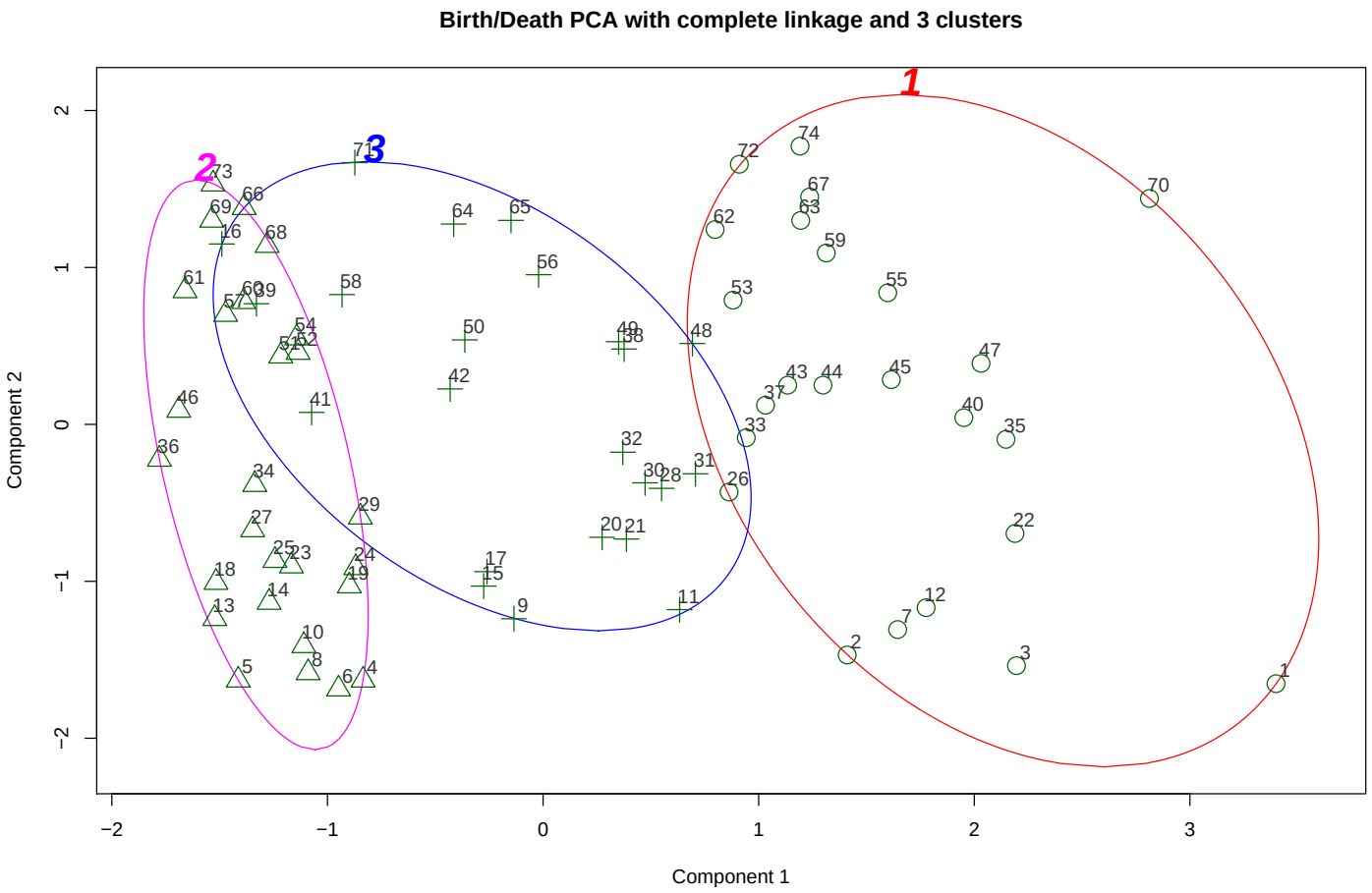


Let's try 3 clusters based on the dendrogram plots below. First we'll use complete linkage.

```
# create distance matrix between points
bd.dist <- dist(bd[,-1])

# number of clusters to identify with red boxes and ellipses
i.clus <- 3

# create dendrogram
```

```
bd.hc.complete  <- hclust(bd.dist, method = "complete")
plot(bd.hc.complete, hang = -1
      , main = paste("Teeth with complete linkage and", i.clus, "clusters")
      , labels = bd[,1])
rect.hclust(bd.hc.complete, k = i.clus)

# create PCA scores plot with ellipses
clusplot(bd, cutree(bd.hc.complete, k = i.clus)
        , color = TRUE, labels = 2, lines = 0
        , cex = 2, cex.txt = 1, col.txt = "gray20"
        , main = paste("Birth/Death PCA with complete linkage and", i.clus, "clusters"), sub =

# create a column with group membership
bd$cut.comp <- factor(cutree(bd.hc.complete, k = i.clus))
```





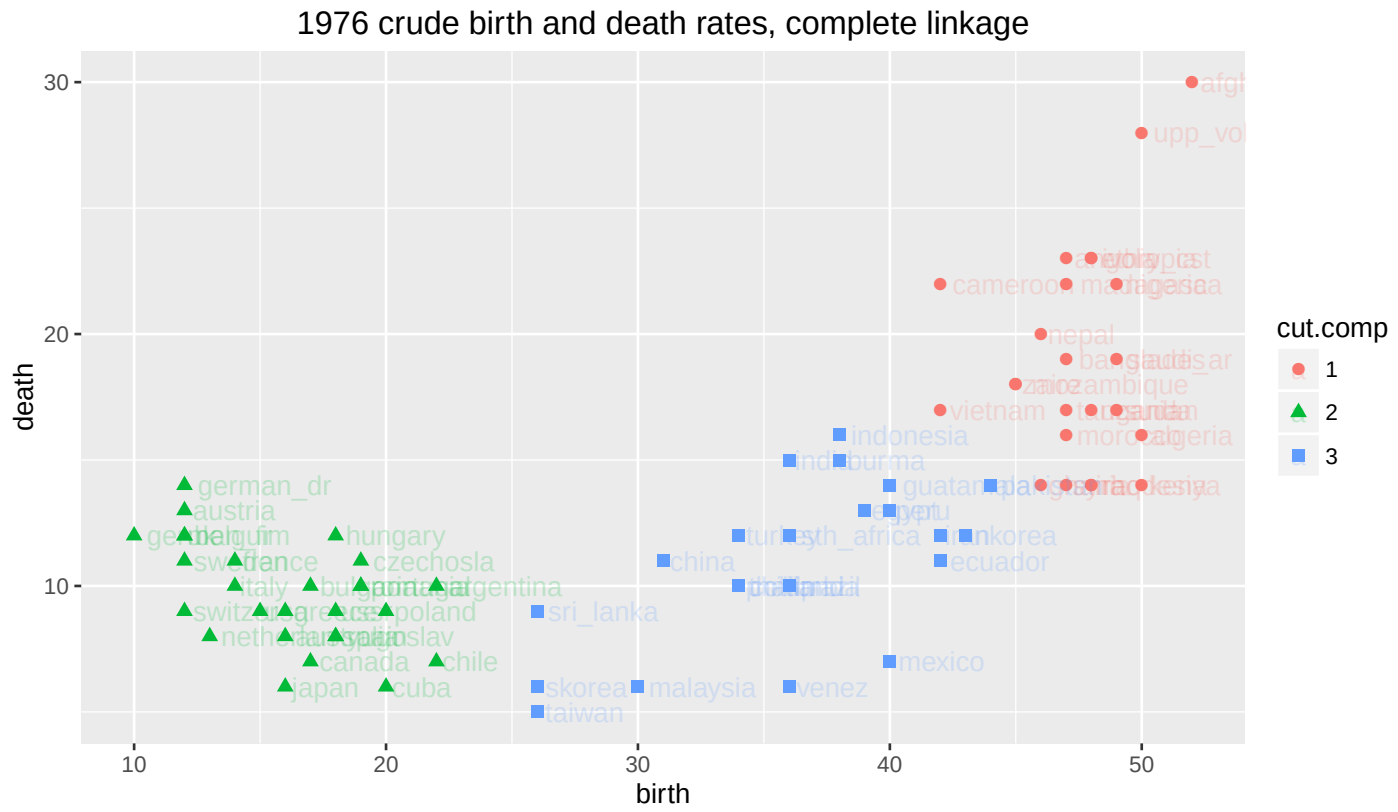

```
# print the observations in each cluster
for (i.cut in 1:i.clus) {
  print(paste("Cluster", i.cut, " ----------------------------- "))
  print(bd[(cutree(bd.hc.complete, k = i.clus) == i.cut),])
}
```

```
## [1] "Cluster 1  ----------------------------- "
##       country birth death cut.comp
## 1      afghan    52    30        1
## 2     algeria    50    16        1
## 3      angola    47    23        1
## 7   banglades    47    19        1
## 12   cameroon    42    22        1
## 22   ethiopia    48    23        1
## 26      ghana    46    14        1
## 33       iraq    48    14        1
## 35  ivory_cst    48    23        1
## 37      kenya    50    14        1
## 40  madagasca    47    22        1
## 43    morocco    47    16        1
## 44 mozambique    45    18        1
## 45      nepal    46    20        1
## 47    nigeria    49    22        1
## 53   rhodesia    48    14        1
```

```
## 55   saudi_ar    49    19        1
## 59      sudan    49    17        1
## 62      syria    47    14        1
## 63   tanzania    47    17        1
## 67     uganda    48    17        1
## 70  upp_volta    50    28        1
## 72    vietnam    42    17        1
## 74      zaire    45    18        1
## [1] "Cluster 2  ---------------------------- "
##      country birth death cut.comp
## 4  argentina    22    10        2
## 5  australia    16     8        2
## 6    austria    12    13        2
## 8    belguim    12    12        2
## 10  bulgaria    17    10        2
## 13    canada    17     7        2
## 14     chile    22     7        2
## 18      cuba    20     6        2
## 19 czechosla    19    11        2
## 23    france    14    11        2
## 24 german_dr    12    14        2
## 25 german_fr    10    12        2
## 27    greece    16     9        2
## 29   hungary    18    12        2
## 34     italy    14    10        2
## 36     japan    16     6        2
## 46 netherlan    13     8        2
## 51    poland    20     9        2
## 52  portugal    19    10        2
## 54   romania    19    10        2
## 57     spain    18     8        2
## 60    sweden    12    11        2
## 61   switzer    12     9        2
## 66      ussr    18     9        2
## 68        uk    12    12        2
## 69       usa    15     9        2
## 73  yugoslav    18     8        2
## [1] "Cluster 3  ---------------------------- "
##       country birth death cut.comp
## 9      brazil    36    10        3
## 11      burma    38    15        3
## 15      china    31    11        3
## 16     taiwan    26     5        3
## 17   columbia    34    10        3
## 20    ecuador    42    11        3
## 21      egypt    39    13        3
## 28  guatamala    40    14        3
## 30      india    36    15        3
```

```
## 31  indonesia    38    16        3
## 32       iran    42    12        3
## 38     nkorea    43    12        3
## 39     skorea    26     6        3
## 41   malaysia    30     6        3
## 42     mexico    40     7        3
## 48   pakistan    44    14        3
## 49       peru    40    13        3
## 50    phillip    34    10        3
## 56 sth_africa    36    12        3
## 58  sri_lanka    26     9        3
## 64   thailand    34    10        3
## 65     turkey    34    12        3
## 71      venez    36     6        3
```

```
# plot original data
library(ggplot2)
p1 <- ggplot(bd, 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[4], it appears that at least some members of the "Four Asian Tigers[5]" 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[6] (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[7] are currently developing in a similar fashion[8].

14.4.2 Single linkage

Now we'll use single linkage to compare.

```
library(NbClust)
# Change integer data type to numeric
bd.num <- as.numeric(as.matrix(bd[,-1]))
NC.out <- NbClust(bd.num, method = "single", 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 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.
##
## 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]
## *******************************************************************
## * Among all indices:
## * 1 proposed 2 as the best number of clusters
## * 1 proposed 5 as the best number of clusters
## * 2 proposed 6 as the best number of clusters
## * 1 proposed 7 as the best number of clusters
## * 1 proposed 11 as the best number of clusters
##
##                    ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  6
##
##
## *******************************************************************
```

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

[5]http://en.wikipedia.org/wiki/Four_Asian_Tigers

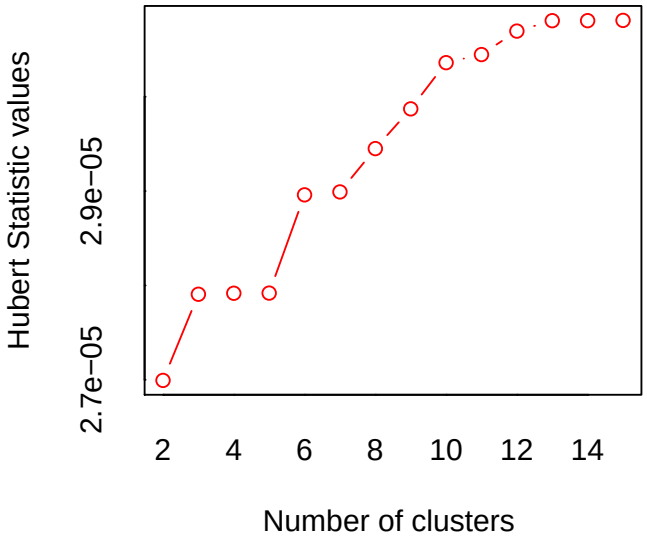
[6]http://www.povertyeducation.org/the-rise-of-asia.html

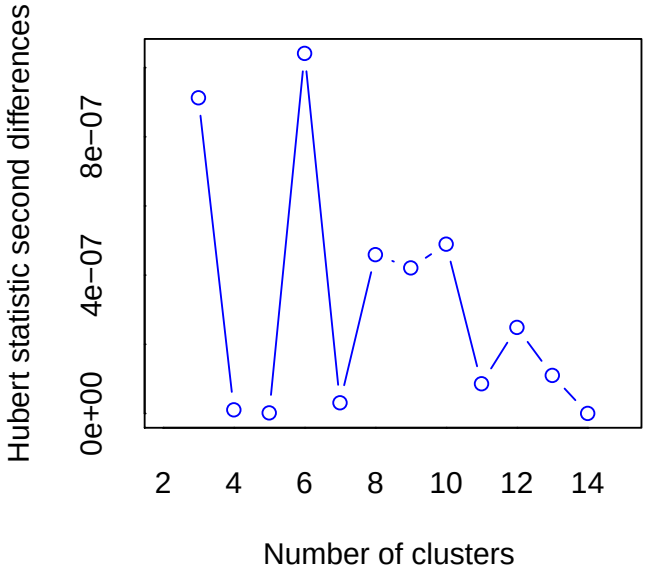
[7]http://en.wikipedia.org/wiki/Tiger_Cub_Economies

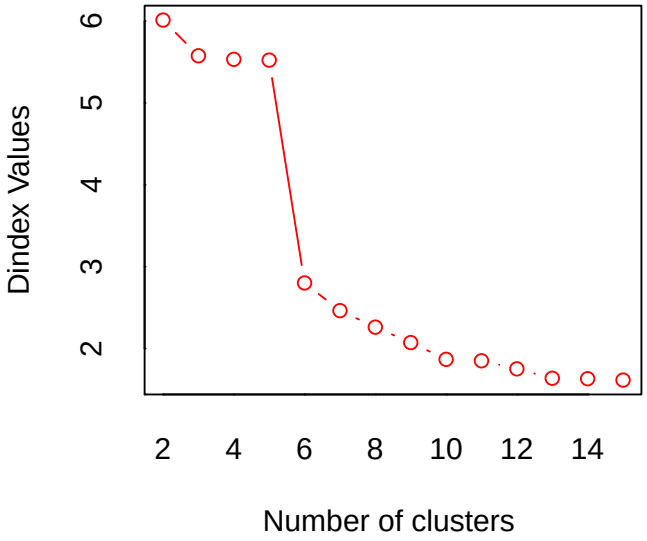
[8]http://www.investopedia.com/terms/t/tiger-cub-economies.asp

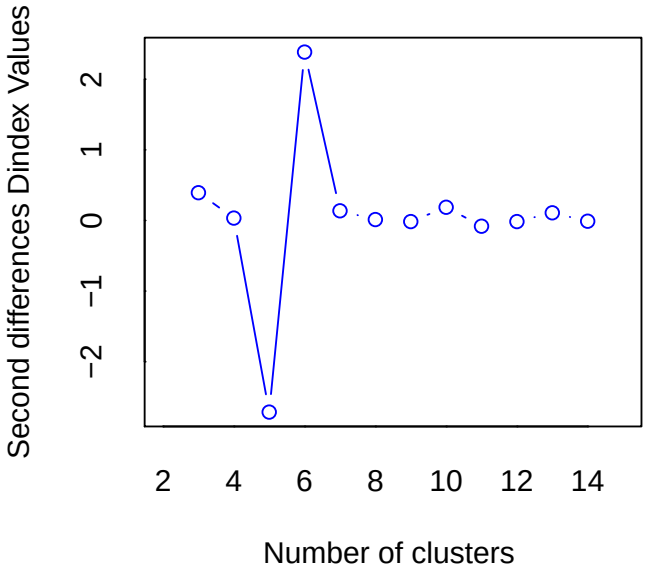
```
# most of the methods suggest 4 to 11 clusters, as do the plots
NC.out$Best.nc
```

```
##                     KL       CH Hartigan     CCC    Scott  Marriot
## Number_clusters 7.0000  11.0000   5.0000  2.0000   6.0000      6.0
## Value_Index     8.5944 342.3491 473.8986 13.7816 221.8442 115129.2
##                 TrCovW   TraceW Friedman    Rubin Cindex    DB
## Number_clusters   -Inf 4990.876  20.3754 -12.0601  0.189 0.341
## Value_Index          6    6.000   6.0000   7.0000 15.000 2.000
##                 Silhouette   Duda PseudoT2 Beale Ratkowsky     Ball
## Number_clusters     0.7364 0.4667  53.7092 0.373    0.3521 9161.833
## Value_Index         2.0000 2.0000   2.0000 7.000    3.0000    2.000
##                 PtBiserial   Frey McClain   Dunn Hubert SDindex
## Number_clusters     0.8462 4.0846  0.1235 0.1364      0  0.5134
## Value_Index         2.0000 2.0000  3.0000 0.0000      3  0.0000
##                 Dindex   SDbw
## Number_clusters      0 0.0442
## Value_Index         15 7.0000
```



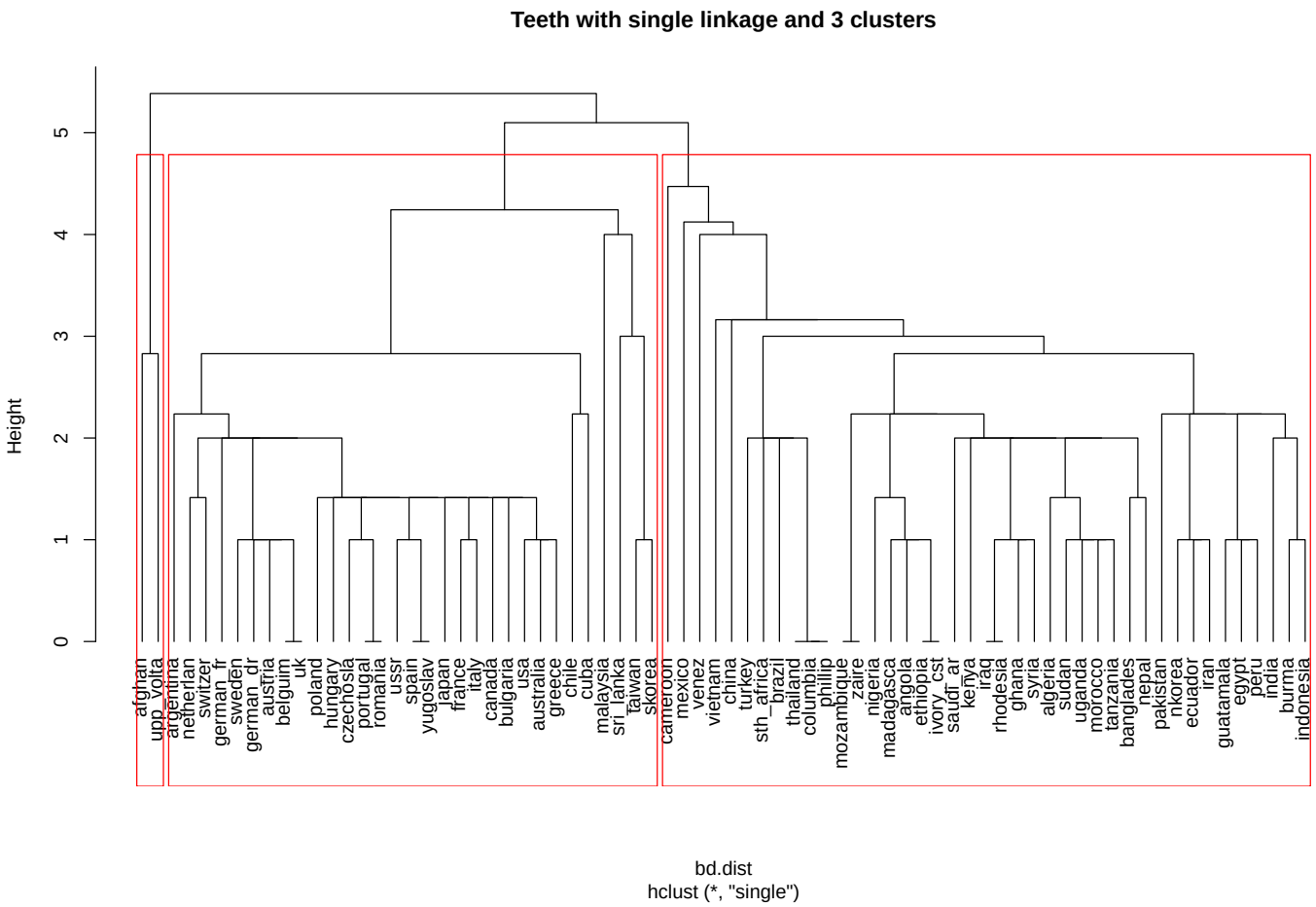


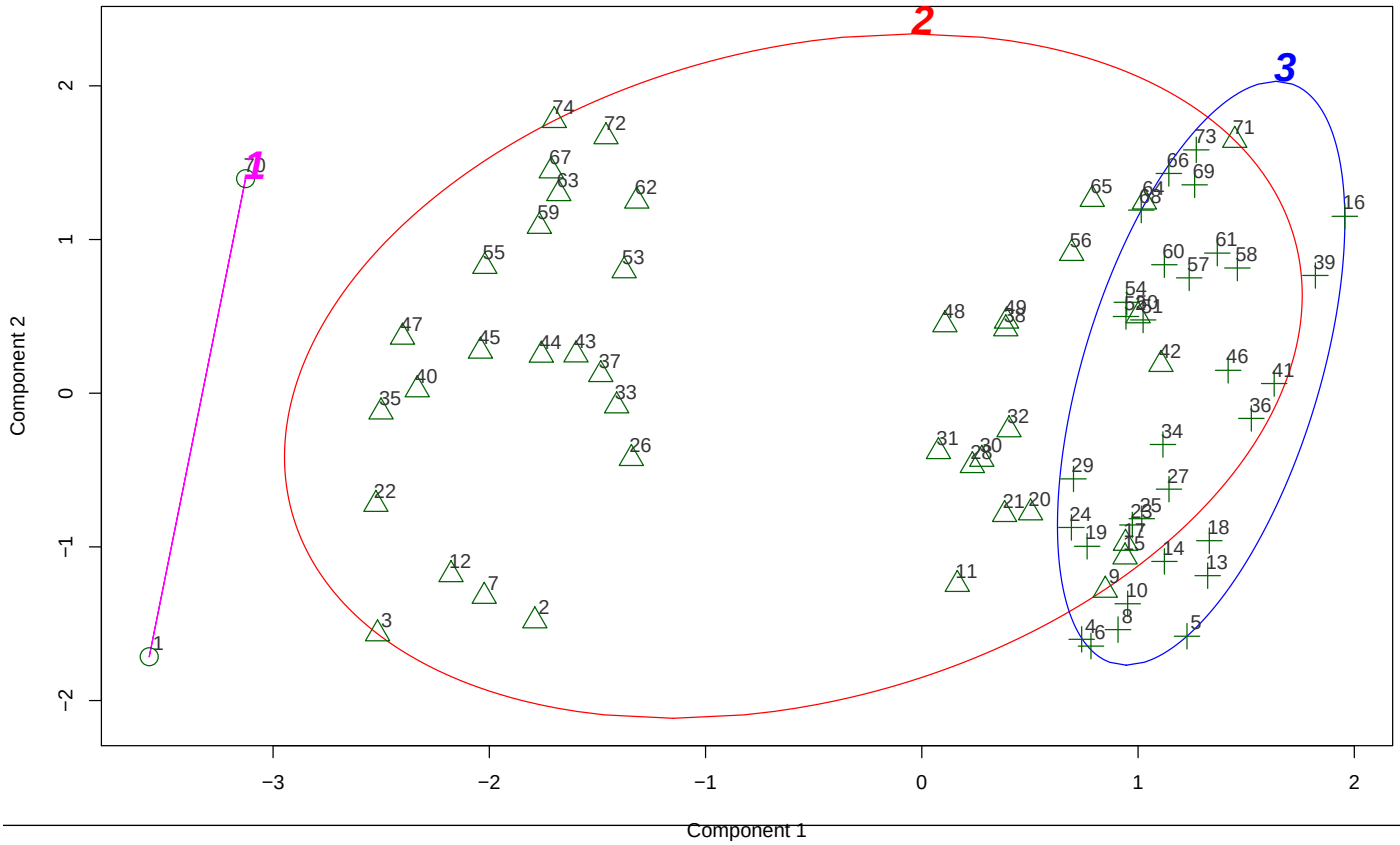





```
# create distance matrix between points
bd.dist <- dist(bd[,-1])

# number of clusters to identify with red boxes and ellipses
i.clus <- 3

# create dendrogram
bd.hc.single  <- hclust(bd.dist, method = "single")
plot(bd.hc.single, hang = -1
      , main = paste("Teeth with single linkage and", i.clus, "clusters")
      , labels = bd[,1])
rect.hclust(bd.hc.single, k = i.clus)

# create PCA scores plot with ellipses
clusplot(bd, cutree(bd.hc.single, k = i.clus)
        , color = TRUE, labels = 2, lines = 0
```

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

# create a column with group membership
bd$cut.sing <- factor(cutree(bd.hc.single, k = i.clus))
```


Teeth with single linkage and 3 clusters
Height
0
1
2
3
4
5
afghan
upp_volta
argentina
netherlan
switzer
german_fr
sweden
german_dr
austria
belguim
uk
poland
hungary
czechosla
portugal
romania
ussr
spain
yugoslav
japan
france
italy
canada
bulgaria
usa
australia
greece
chile
cuba
malaysia
sri_lanka
taiwan
skorea
cameroon
mexico
venez
vietnam
china
turkey
sth_africa
brazil
thailand
columbia
phillip
mozambique
zaire
nigeria
madagasca
angola
ethiopia
ivory_cst
saudi_ar
kenya
iraq
rhodesia
ghana
syria
algeria
sudan
uganda
morocco
tanzania
banglades
nepal
pakistan
nkorea
ecuador
iran
guatamala
egypt
peru
india
burma
indonesia
bd.dist
hclust (*, "single")

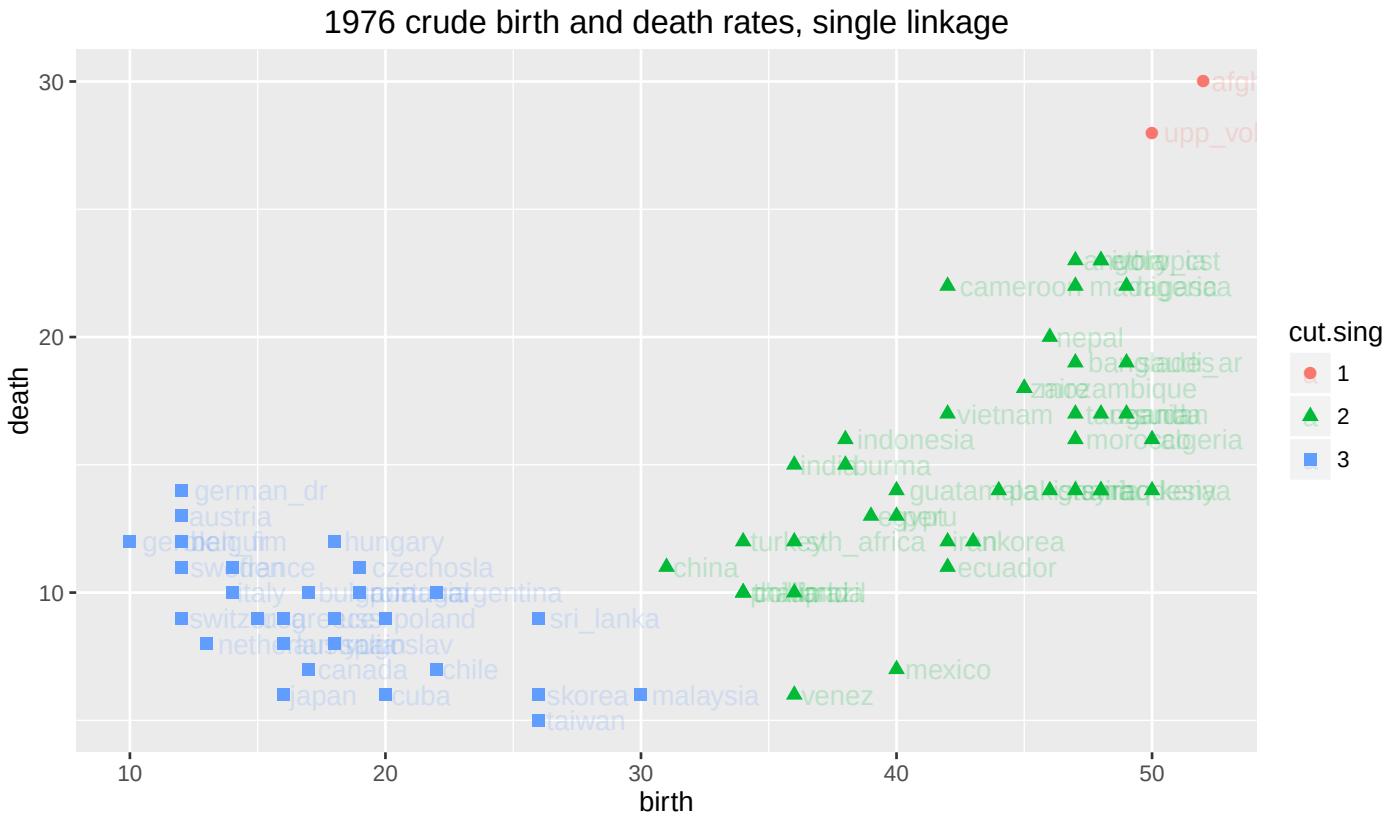

Birth/Death PCA with single linkage and 3 clusters
1
2
3
Component 2
−2
−1
0
1
2
−3
−2
−1
0
1
2
Component 1

```r
# print the observations in each cluster
for (i.cut in 1:i.clus) {
  print(paste("Cluster", i.cut, " ----------------------------- "))
  print(bd[(cutree(bd.hc.single, k = i.clus) == i.cut),])
}
```

```
## [1] "Cluster 1  ----------------------------- "
##      country birth death cut.comp cut.sing
## 1     afghan    52    30        1        1
## 70 upp_volta    50    28        1        1
## [1] "Cluster 2  ----------------------------- "
##       country birth death cut.comp cut.sing
## 2     algeria    50    16        1        2
## 3      angola    47    23        1        2
## 7   banglades    47    19        1        2
## 9      brazil    36    10        3        2
## 11      burma    38    15        3        2
## 12   cameroon    42    22        1        2
## 15      china    31    11        3        2
## 17   columbia    34    10        3        2
## 20    ecuador    42    11        3        2
## 21      egypt    39    13        3        2
## 22   ethiopia    48    23        1        2
## 26      ghana    46    14        1        2
## 28  guatamala    40    14        3        2
## 30      india    36    15        3        2
## 31  indonesia    38    16        3        2
## 32       iran    42    12        3        2
## 33       iraq    48    14        1        2
## 35  ivory_cst    48    23        1        2
## 37      kenya    50    14        1        2
## 38     nkorea    43    12        3        2
## 40  madagasca    47    22        1        2
## 42     mexico    40     7        3        2
## 43    morocco    47    16        1        2
## 44 mozambique    45    18        1        2
## 45      nepal    46    20        1        2
## 47    nigeria    49    22        1        2
## 48   pakistan    44    14        3        2
## 49       peru    40    13        3        2
## 50    phillip    34    10        3        2
## 53   rhodesia    48    14        1        2
## 55   saudi_ar    49    19        1        2
## 56 sth_africa    36    12        3        2
## 59      sudan    49    17        1        2
## 62      syria    47    14        1        2
## 63   tanzania    47    17        1        2
## 64   thailand    34    10        3        2
## 65     turkey    34    12        3        2
```

```
## 67     uganda    48    17         1         2
## 71      venez    36     6         3         2
## 72    vietnam    42    17         1         2
## 74      zaire    45    18         1         2
## [1] "Cluster 3  ---------------------------- "
##      country birth death cut.comp cut.sing
## 4  argentina    22    10        2         3
## 5  australia    16     8        2         3
## 6    austria    12    13        2         3
## 8    belguim    12    12        2         3
## 10  bulgaria    17    10        2         3
## 13    canada    17     7        2         3
## 14     chile    22     7        2         3
## 16    taiwan    26     5        3         3
## 18      cuba    20     6        2         3
## 19 czechosla    19    11        2         3
## 23    france    14    11        2         3
## 24 german_dr    12    14        2         3
## 25 german_fr    10    12        2         3
## 27    greece    16     9        2         3
## 29   hungary    18    12        2         3
## 34     italy    14    10        2         3
## 36     japan    16     6        2         3
## 39    skorea    26     6        3         3
## 41  malaysia    30     6        3         3
## 46 netherlan    13     8        2         3
## 51    poland    20     9        2         3
## 52  portugal    19    10        2         3
## 54   romania    19    10        2         3
## 57     spain    18     8        2         3
## 58 sri_lanka    26     9        3         3
## 60    sweden    12    11        2         3
## 61   switzer    12     9        2         3
## 66      ussr    18     9        2         3
## 68        uk    12    12        2         3
## 69       usa    15     9        2         3
## 73  yugoslav    18     8        2         3
```

```
# plot original data
library(ggplot2)
p1 <- ggplot(bd, 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.