Stat 427/627 Statistical Machine Learning

In-class Lab 11. Introduction to Unsupervised Learning: PCA and Clustering

Contents

1 PCA (Review)

We first introduced Principal Component Analysis and its application when we discussed dimension reduction. Please see lab 8 for more details.

Here is a review example.

In the data set `teeth.cvs`, the numbers of different teeth (8 types) of 32 mammals is given. The teeth are top incisors, bottom incisors, top canines, bottom canines, top premolars, bottom premolars, top molars, and bottom molars. A cluster analysis will be used to identify the mammals that have similar counts across the eight teeth types.

```
teeth <- read.csv("../Data/teeth.csv", header=T)
rownames(teeth) <- teeth$mammal
teeth <- teeth[, -1]
head(teeth, 3)
```

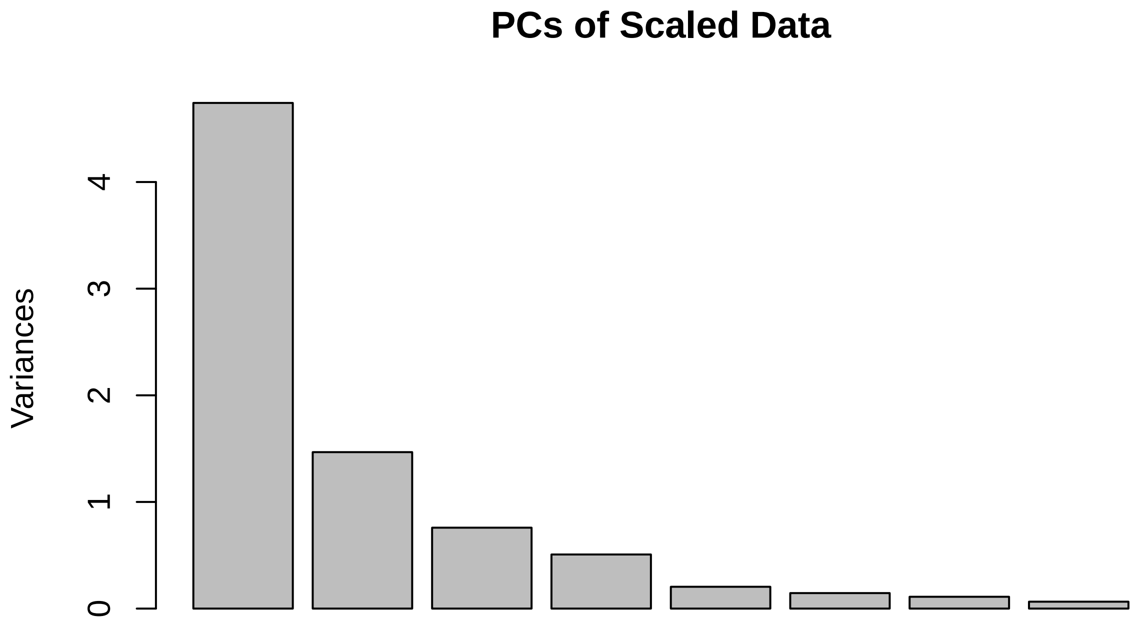
```
##                   inctop incbot cantop canbot pretop prebot moltop molbot
## BROWN BAT              2      3      1      1      3      3      3      3
## MOLE                   3      2      1      0      3      3      3      3
## SILVER HAIR BAT        2      3      1      1      2      3      3      3
```

- Use `scale = TRUE` to standardize the data before computing the PCs.

```
teeth.pcs <- prcomp(teeth, scale=TRUE)
teeth.pcs$rotation[ , c(1:3)] # Extract the loading (i.e., transformation)
```

```
##               PC1         PC2        PC3
## inctop  0.3596792  0.33035725 -0.2143159
## incbot  0.1682294 -0.69061491 -0.1527100
## cantop  0.3827299 -0.19801013 -0.4371581
## canbot  0.3911168  0.04945103 -0.4979039
## pretop  0.3727780 -0.19259178  0.5525092
## prebot  0.3709015 -0.32330600  0.3483855
## moltop -0.3815157 -0.30733168 -0.1556981
## molbot -0.3475482 -0.36904010 -0.2021190
```

```
plot(teeth.pcs, main="PCs of Scaled Data")
```

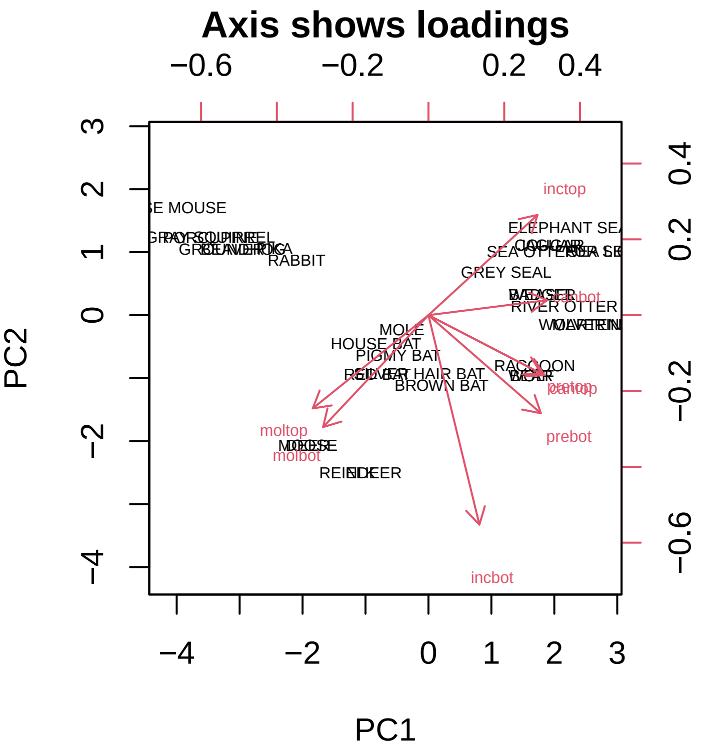


```
summary(teeth.pcs)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.1775 1.2112 0.87104 0.71221 0.45233 0.38124 0.33293
## Proportion of Variance 0.5927 0.1834 0.09484 0.06341 0.02557 0.01817 0.01386
## Cumulative Proportion  0.5927 0.7761 0.87090 0.93431 0.95988 0.97805 0.99190
##                           PC8
## Standard deviation     0.2545
## Proportion of Variance 0.0081
## Cumulative Proportion  1.0000
```

- Function `biplot()` plots first 2 PCs' scores (bottom, left) and their loadings (top, right). Use `scale = 0` to show the PC scores and PC loadings as is.

```
biplot(teeth.pcs, scale=0, cex=0.5, main="Axis shows loadings")
```



- The PC scores are already saved in the output object (`prcomp.out$x`). They can also be calculated using the `predict()` function.

2 K-mean Cluster

```
teeth.KM2 <- kmeans(teeth, 2)
teeth.KM2
```

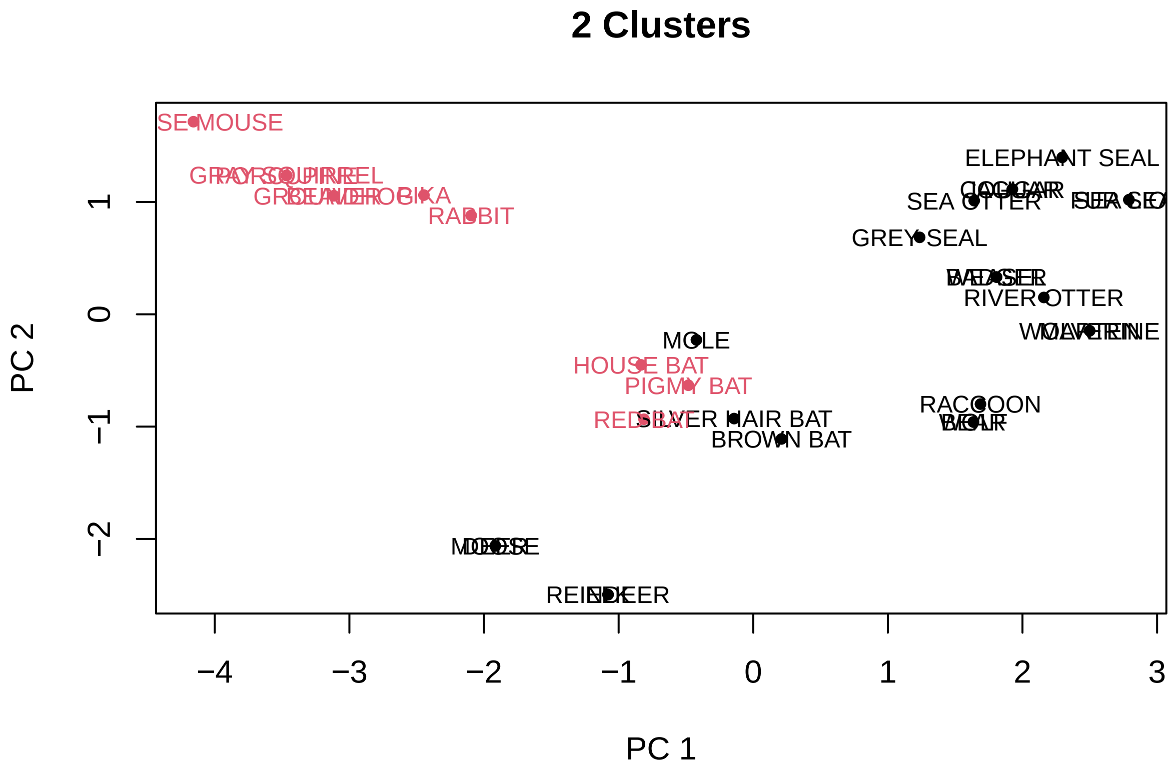
```
## K-means clustering with 2 clusters of sizes 22, 10
##
## Cluster means:
##     inctop   incbot    cantop    canbot   pretop   prebot   moltop   molbot
## 1 2.318182 2.863636 0.9090909 0.7727273 3.363636 3.272727 1.863636 2.181818
## 2 1.400000 1.600000 0.3000000 0.3000000 1.600000 1.400000 3.000000 3.000000
##
## Clustering vector:
##       BROWN BAT            MOLE SILVER HAIR BAT       PIGMY BAT       HOUSE BAT
##               1               1               1               2               2
##         RED BAT            PIKA          RABBIT          BEAVER       GROUNDHOG
##               2               2               2               2               2
##   GRAY SQUIRREL     HOUSE MOUSE       PORCUPINE            WOLF            BEAR
##               2               2               2               1               1
##         RACCOON          MARTEN          WEASEL       WOLVERINE          BADGER
##               1               1               1               1               1
##     RIVER OTTER       SEA OTTER          JAGUAR          COUGAR        FUR SEAL
##               1               1               1               1               1
##        SEA LION       GREY SEAL   ELEPHANT SEAL        REINDEER             ELK 
##               1               1               1               1               1 
##            DEER           MOOSE 
##               1               1 
## 
## Within cluster sum of squares by cluster:
## [1] 94.36364 25.80000
##  (between_SS / total_SS =  39.9 %)
## 
## Available components:
## 
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
names(teeth.KM2)
```

```
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

Visualizing K-mean cluster can be done for 2-dimensional data. Below is an illustration using the PCs of the teeth data.

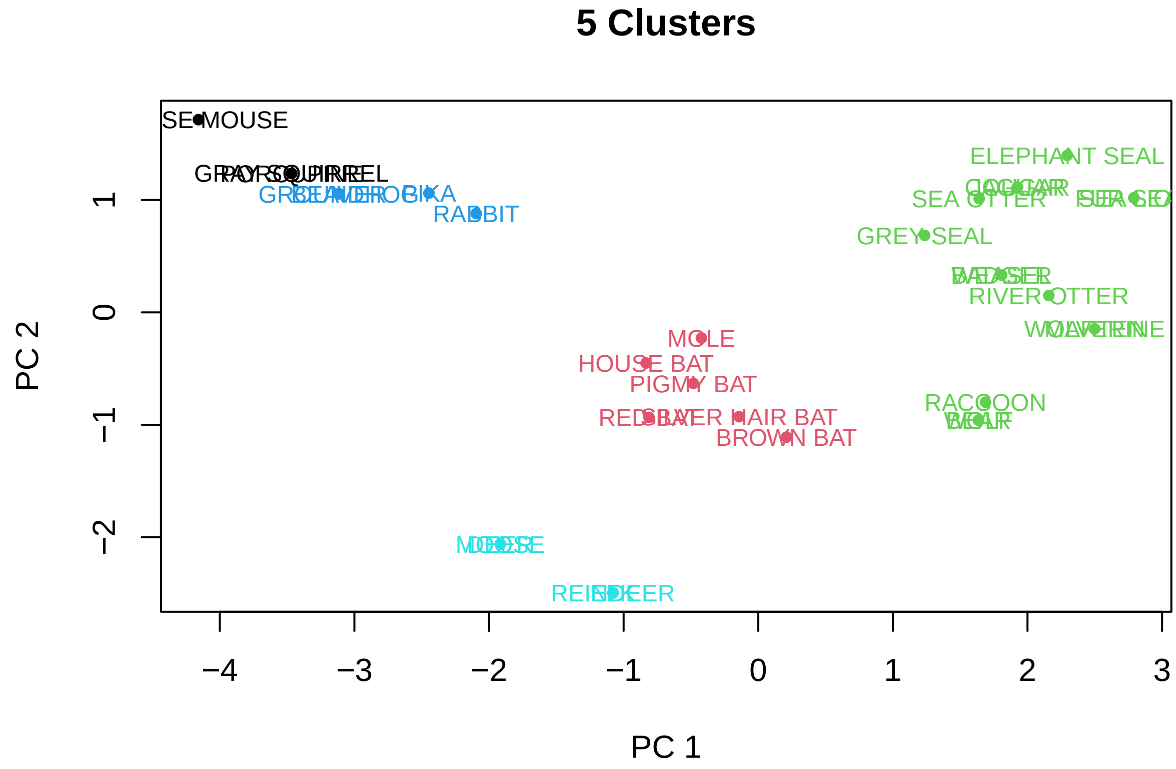
```
plot(teeth.pcs$x[,1], teeth.pcs$x[,2], pch=20, xlab="PC 1", ylab="PC 2",
     col=teeth.KM2$cluster, main="2 Clusters")
text(teeth.pcs$x[,1], teeth.pcs$x[,2], labels = row.names(teeth), cex=0.75,
     col=teeth.KM2$cluster)
```



More clusters?

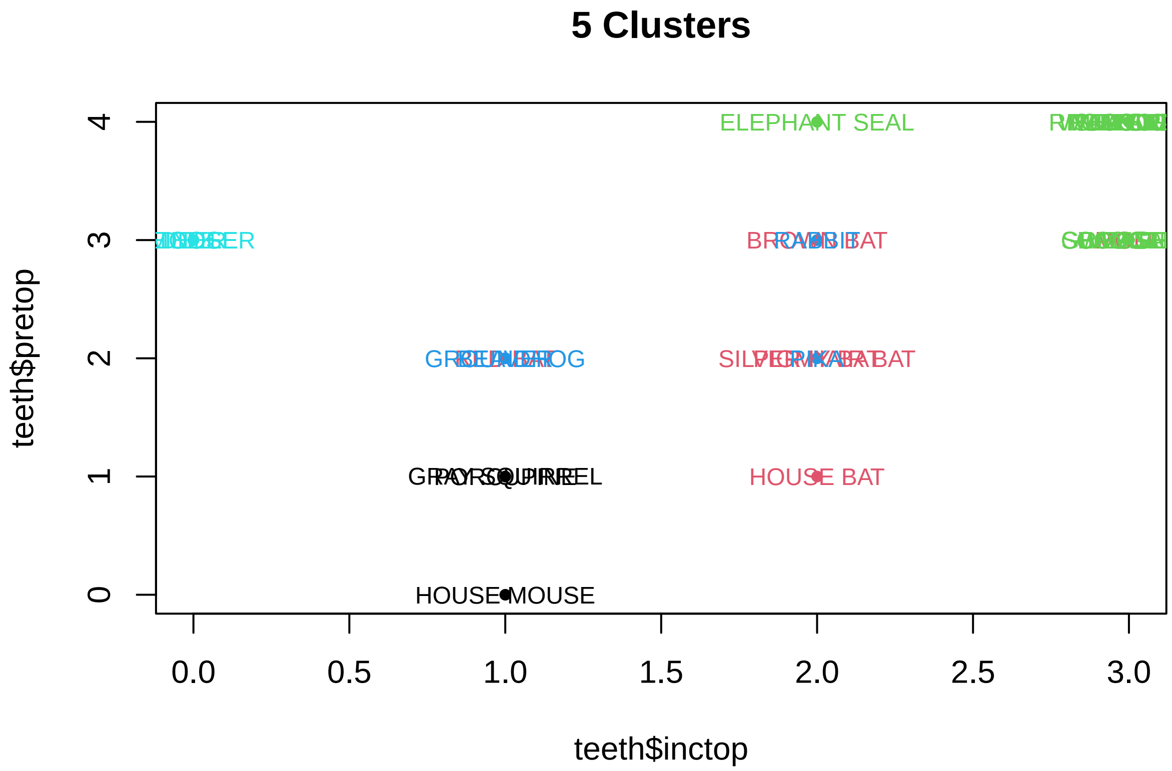
```
teeth.KM5 <- kmeans(teeth, 5)

plot(teeth.pcs$x[,1], teeth.pcs$x[,2], pch=20, xlab="PC 1", ylab="PC 2",
     col=teeth.KM5$cluster, main="5 Clusters")
text(teeth.pcs$x[,1], teeth.pcs$x[,2], labels = row.names(teeth), cex=0.75,
     col=teeth.KM5$cluster)
```



- Note that being "close" on a 2-dimensional plot does not mean the observations are "close" in higher-dimension.
- We can also plot pairs of the original variables and the clusters.

```
plot(teeth$inctop, teeth$pretop, pch=20, col=teeth.KM5$cluster, main="5 Clusters")
text(teeth$inctop, teeth$pretop, labels = row.names(teeth), cex=0.75,
     col=teeth.KM5$cluster)
```

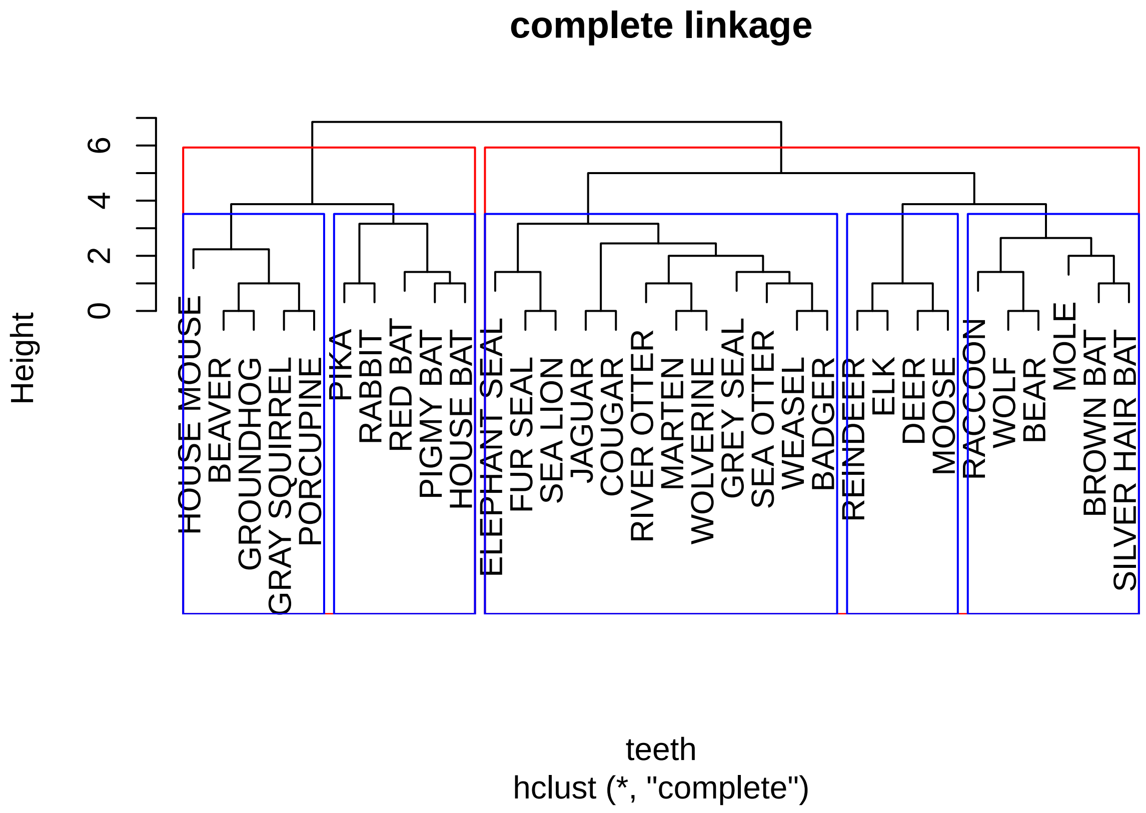


3 Hierarchical Cluster

Functions `dist()` and `hclust()` works together to conducts Hierarchical Cluster Analysis.

- `dist()` computes the dissimilarities of the observations. The default is Euclidean distance. Use argument `method =` to change the dissimilarity measures.
- `hclust()` builds the hierarchical clusters. The default linkage (i.e. "agglomeration") method is the "complete" linkage. Use argument `method =` to change the linkage.
- Other relevant functions:
  - `plot()`, `rect.hclust()`, plot the dendrogram and draw rectangles to highlight the cluster.
  - `cutree()`, sets the number of clusters.

```
comp <- hclust(dist(teeth), method="complete") # max inter-cluster dissimilarity
plot(comp, xlab="teeth", main="complete linkage", labels=row.names(teeth))
rect.hclust(comp, k=2, border="red")
rect.hclust(comp, k=5, border="blue")
```

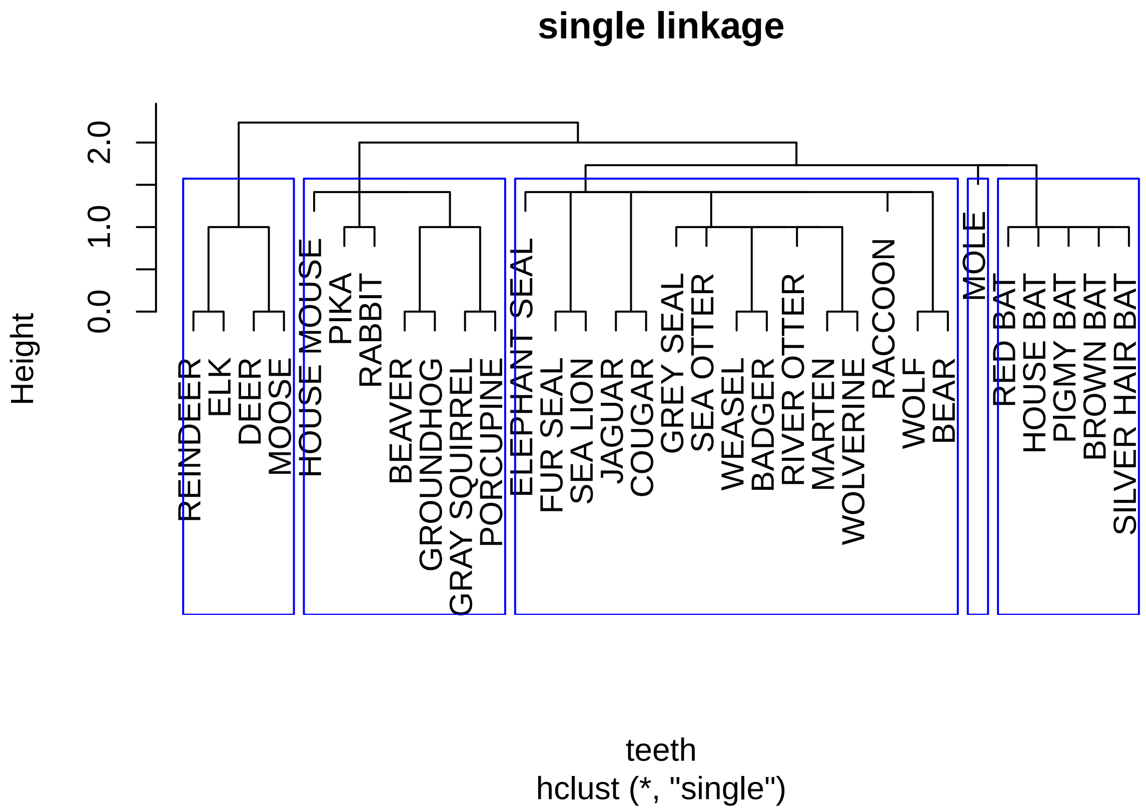


```
comp5 <- cutree(comp, 5)
comp5
```

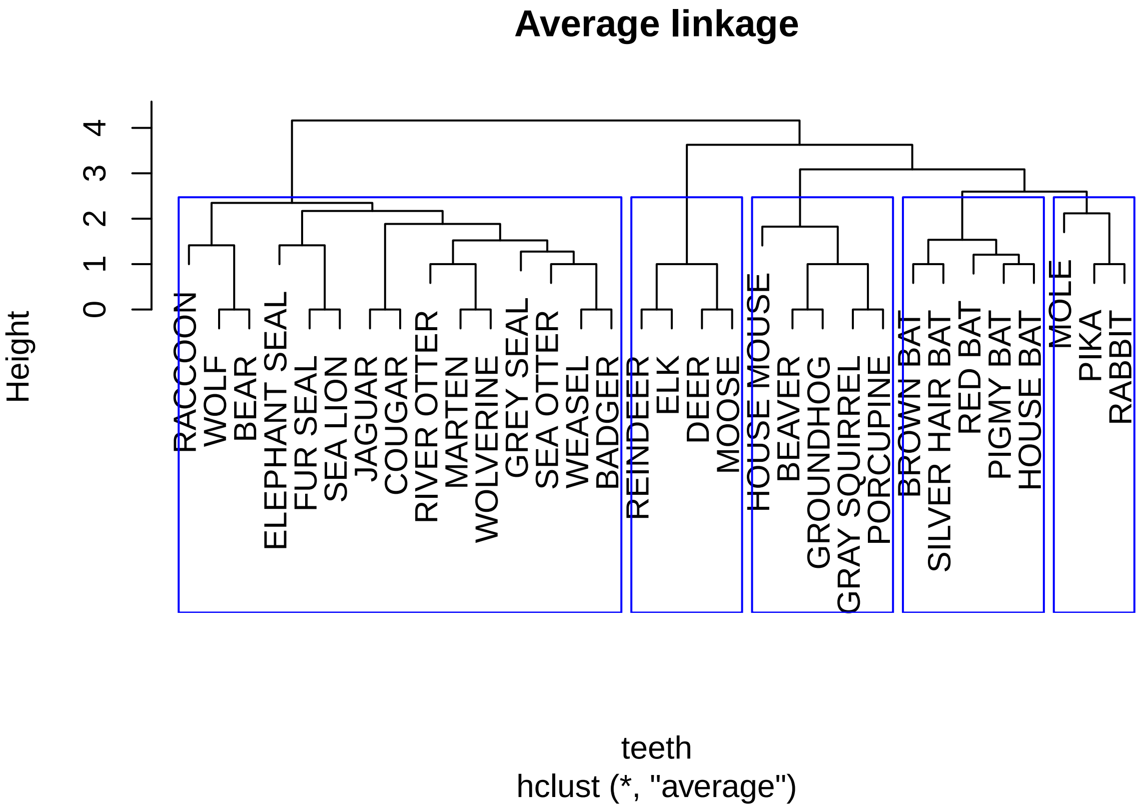
```
##       BROWN BAT            MOLE SILVER HAIR BAT       PIGMY BAT       HOUSE BAT
##               1               1               1               2               2
##         RED BAT            PIKA          RABBIT          BEAVER       GROUNDHOG
##               2               2               2               3               3
##   GRAY SQUIRREL     HOUSE MOUSE       PORCUPINE            WOLF            BEAR
##               3               3               3               1               1
##         RACCOON          MARTEN          WEASEL       WOLVERINE          BADGER
##               1               4               4               4               4
##     RIVER OTTER       SEA OTTER          JAGUAR          COUGAR        FUR SEAL
##               4               4               4               4               4
##         SEA LION       GREY SEAL   ELEPHANT SEAL        REINDEER             ELK
##               4               4               4               5               5
##            DEER           MOOSE
##               5               5
```

Other linkage methods.

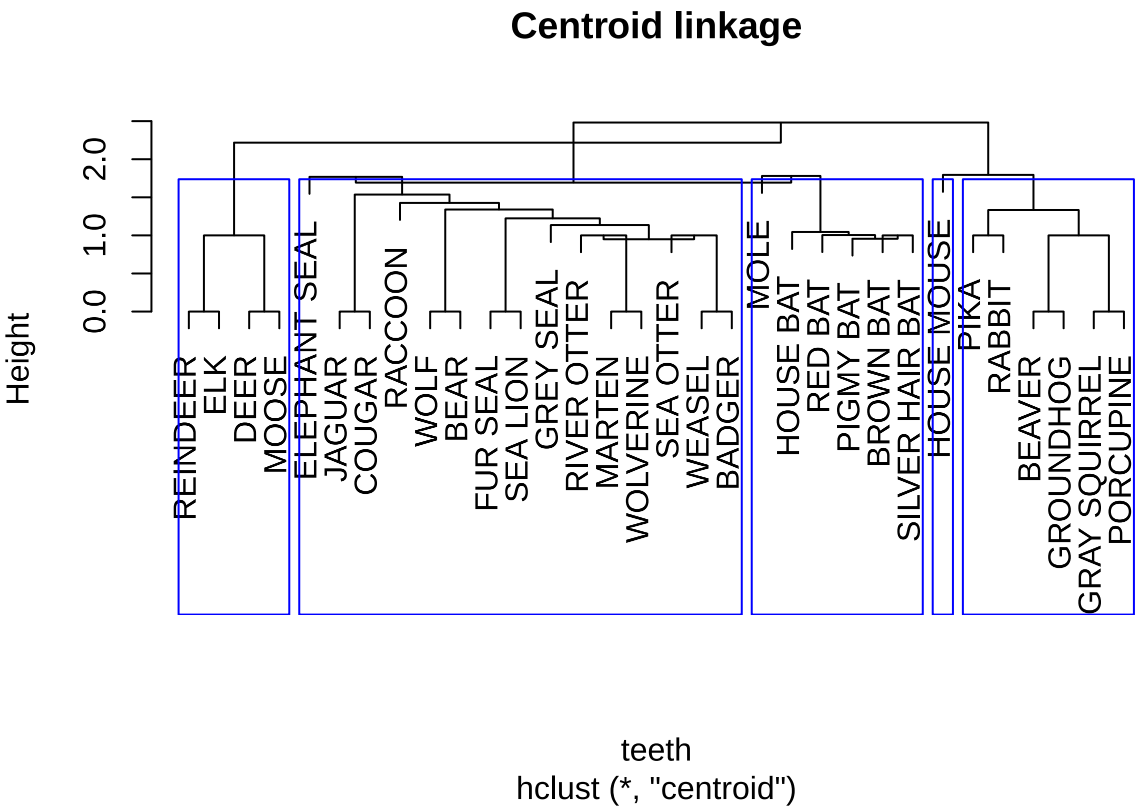
```
sing <- hclust(dist(teeth), method="single") # min inter-cluster dissimilarity
plot(sing, xlab="teeth", main="single linkage", labels=row.names(teeth))
rect.hclust(sing, k=5, border="blue")
```



```r
aver <- hclust(dist(teeth), method="average") # mean inter-cluster dissimilarity
plot(aver, xlab="teeth", main="Average linkage", labels=row.names(teeth))
rect.hclust(aver, k=5, border="blue")
```

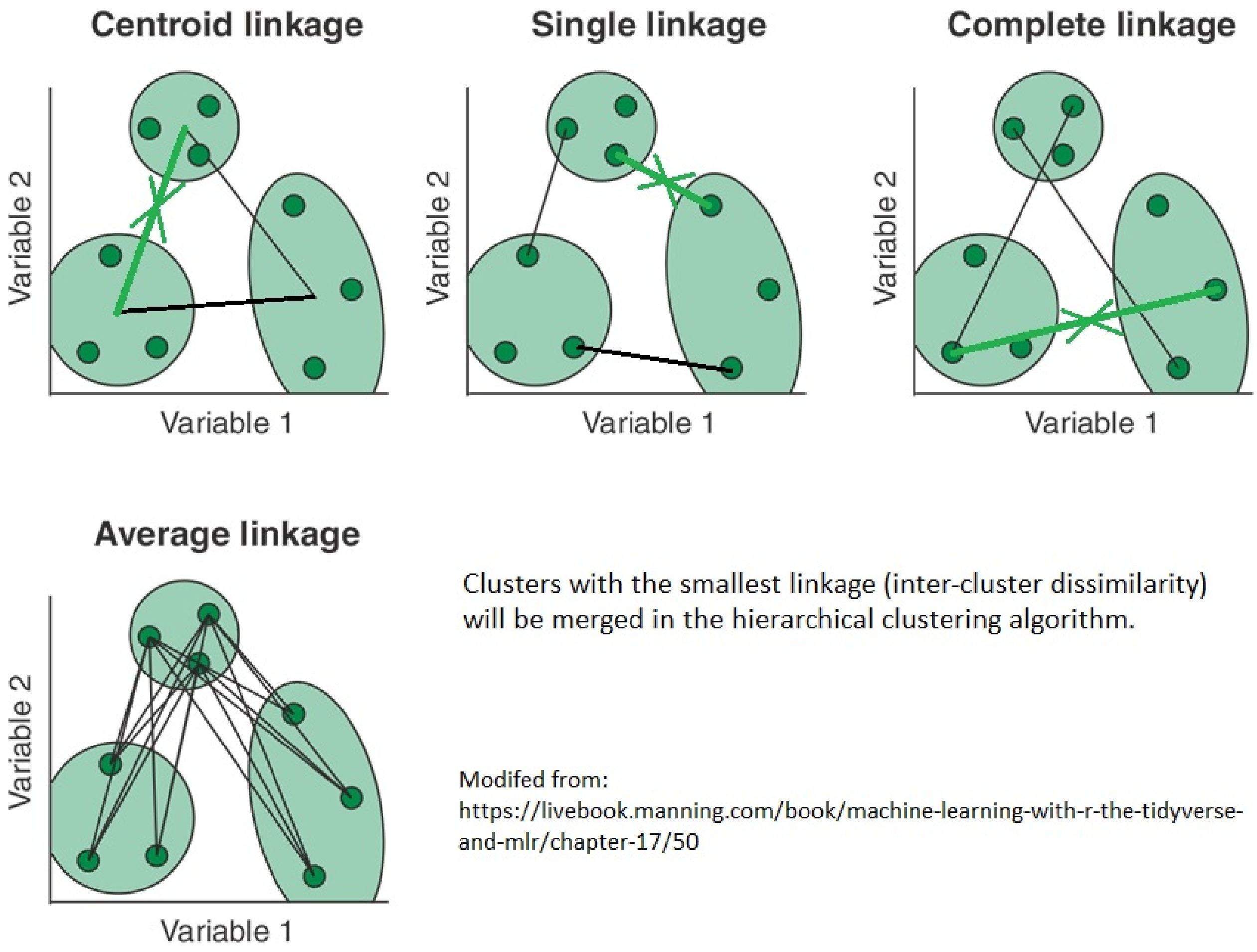


```r
cent <- hclust(dist(teeth), method="centroid") # dissimilarity between cluster centers
plot(cent, xlab="teeth", main="Centroid linkage", labels=row.names(teeth))
rect.hclust(cent, k=5, border="blue")
```



Remarks.

- If the variables' scales are very different, use `scale()` function to standardize all variables.
- The clusters can be sensitive to the choices of dissimilarity measures, linkages, number of clusters, etc.



Centroid linkage

Single linkage

Complete linkage

Average linkage

Clusters with the smallest linkage (inter-cluster dissimilarity) will be merged in the hierarchical clustering algorithm.

Modifed from:
https://livebook.manning.com/book/machine-learning-with-r-the-tidyverse-and-mlr/chapter-17/50

Figure 1: Cluster Linkage