Knowledge based systems

Lab 1

```
library(cluster)
```

Task 1

a. Installed `cluster`

b. There are 50 flowers per species.

c. Average petal length is 3.758 and sepal length 5.843.

```
summary(iris)
```

```
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
 Median :5.800   Median :3.000   Median :4.350   Median :1.300
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
       Species
 setosa    :50
 versicolor:50
 virginica :50
```

d. `iris$Species` is a character vector of class “factor” with three levels (“setosa”, “versicolor”, “virginica”).

```
str(iris$Species)
```

```
 Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

e. I will focus on sepal length and width.

```r
f1 <- "Sepal.Length"
f2 <- "Sepal.Width"
features <- c(f1, f2)
selectedIris <- iris[, features]
```

Task 2

a. The general k-means clustering algorithm is to select k random points as initial centers, assign all points to nearest center, compute new center and reassign all points to nearest center. Repeat until centers don't change.

b.

```r
specs <- iris$Species # select last column
specsLevs <- levels(specs) # select levels of variables (levels works only
with Factor)
specsNum <- length(specsLevs) # how many = length

fitKmeans <- kmeans(selectedIris, specsNum, iter.max = 100)
```

```r
clusplot(
  selectedIris,
  fitKmeans$cluster,
  color = T,
  shade = T,
  labels = 0,
  lines = 0,
  main = "k-means clustering"
)
```

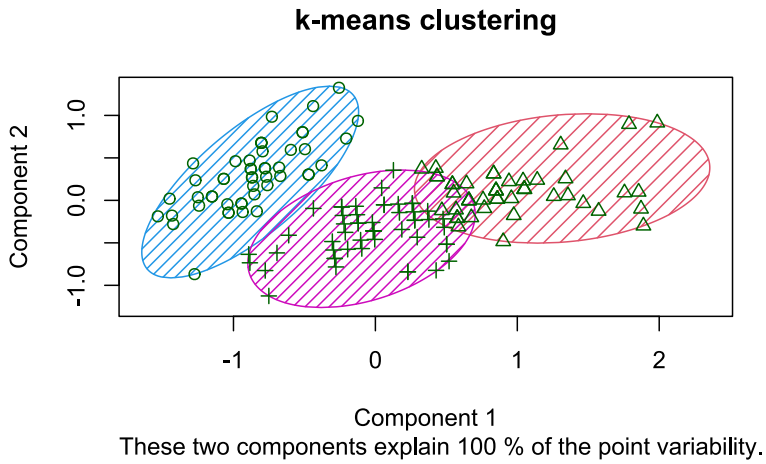

Fig.1

c. The clusters may change depending on random starting points.

d. The clusters are mostly separate with a little overlap.

Task 3

```
d <- dist(selectedIris) # estimate distance
fitHier <- hclust(d, method = "ward.D")
clusterCut <- cutree(fitHier, specsNum)
```

a. The general hierarchical clustering algorithm is to compute a distance matrix, unify the two points with smallest distance, recompute the new distance matrix, and repeat until all points are in the same group.

b.

```
# Fig.2a
clusplot(
  selectedIris,
  clusterCut,
  color = TRUE,
  shade = TRUE,
  labels = 0,
  lines = 0,
  main = "hierarchical clustering"
)
```

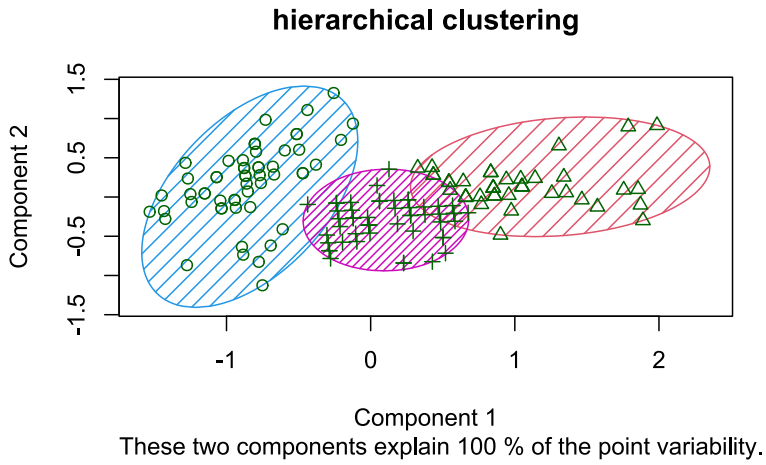

Fig.2a

c. The clusters will be the same when using the same distance method.

d. Clusters are separated with a little overlap.

e.

```
# Fig.2b
dend <- as.dendrogram(fitHier)
nodeParams <- list(pch = c(NA, 18), cex = 0.5, col = "darkgray")
plot(
  dend,
  type = "rectangle",
  nodePar = nodeParams,
  leaflab = "none",
  main = "hierarchical clustering"
)
rect.hclust(fitHier, k = 3, border = "darkorange")
```

Fig.2b

f. A dendrogram will show how far apart the clusters are (compared to cladogram).

g. A dendrogram shows how points cluster together at all different levels/number of groups, as well as the length of the branches showing the distance between nodes.

Task 4

```
hierCentr <- aggregate(
  data.frame(clusterCut, selectedIris),
  by = list(clusterCut),
  FUN = mean
)
```

```
plot(selectedIris, type = "n", main = "k-means clustering")
text(
```

```
  selectedIris,
  labels = iris$Species,
  col = c("orangered", "limegreen", "dodgerblue")[fitKmeans$cluster]
)
points(fitKmeans$centers[, features], col = "yellow", pch = 15, cex = 1.5)

# hierarchical
plot(selectedIris, type = "n", main = "hierarchical clustering")
text(
  selectedIris,
  labels = iris$Species,
  col = c("darkviolet", "darkorange", "seagreen4")[clusterCut]
)
points(hierCentr[, features], col = "yellow", pch = 15, cex = 1.5)
```

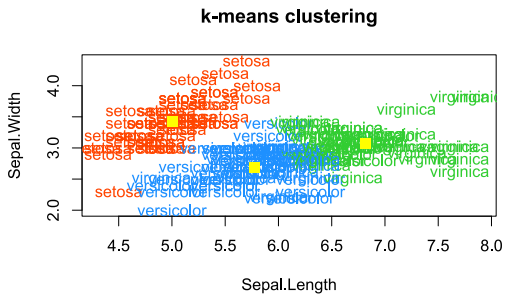

Figure 1: k-means clustering

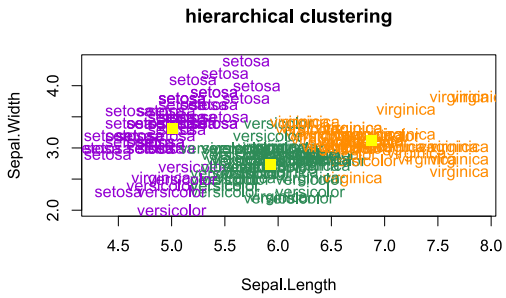

Figure 2: hierarchiacal clustering

Fig.3

a. In the hierarchical clustering there are a few versicolor wrongly grouped with setosa.

b. The yellow squares represent the centroids of each group.

Task 5

a. Unsupervised learning uses the data without potentially being biased due to *a priori* assumptions.

b.

- Pros: Avoids potential bias, helps identify unknown groups.
- Cons: k-means doesn't handle non-uniform/globular shaped clusters reliably, only uses two dimensions.

c. Differentiating between species based on physiological measurements, and determining evolutionary relationships.