lab7: machine learning I

Elsa Chen (A16632961)

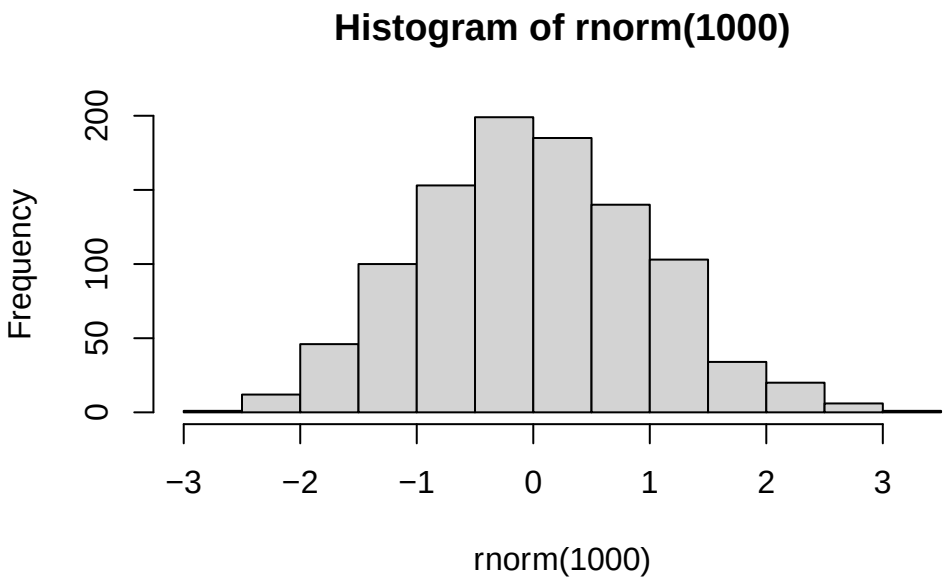
Table of contents

Today we're exploring first part of machine learing, clustering - finding patterns in data and dimensionality reduction

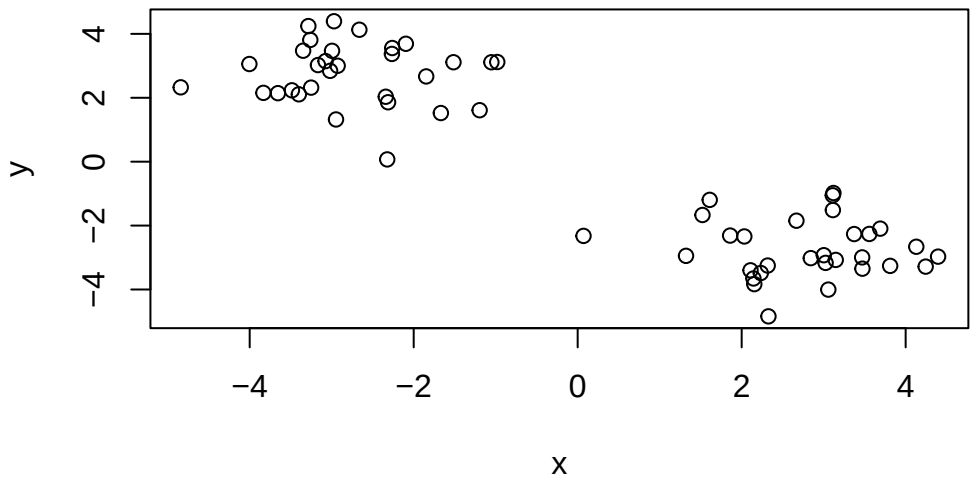
Clustering

Starting with "k-means" clustering (`kmeans()`)

```
# making up data
hist(rnorm(1000))
```




```
temp <- c(rnorm(30, -3), rnorm(30, 3))
x <- cbind(x = temp, y = rev(temp))
plot(x)
```



now running `kmeans()`

```
km <- kmeans(x, centers = 2)
km
```

```
K-means clustering with 2 clusters of sizes 30, 30

Cluster means:
          x         y
1 -2.732907  2.764193
2  2.764193 -2.732907

Clustering vector:
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
[39] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

Within cluster sum of squares by cluster:
[1] 50.80205 50.80205
 (between_SS / total_SS =  89.9 %)

Available components:
```

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

```
attributes(km)
```

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

$class
[1] "kmeans"
```

Q. How many points in each cluster?

```
km$size
```

```
[1] 30 30
```

Q. What component of your result object details cluster assignment/membership?

```
km$cluster
```

```
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
[39] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

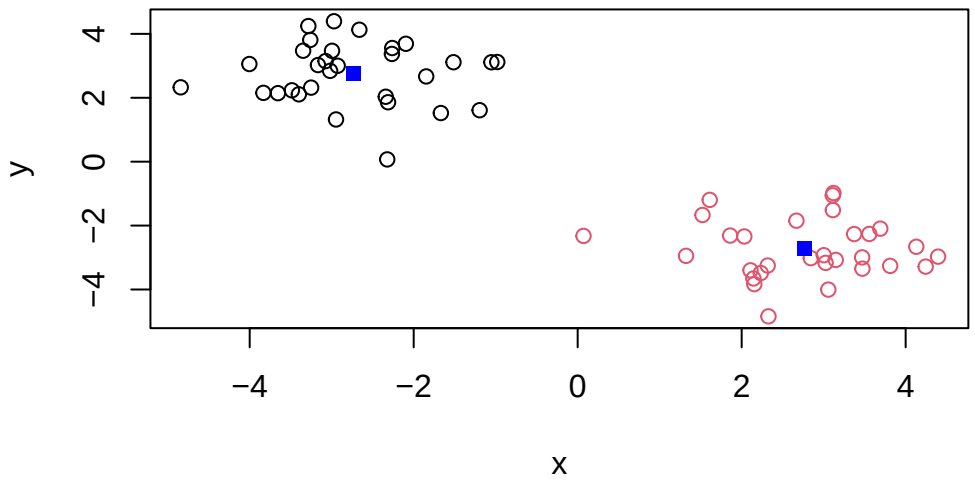
Q. What are centers/mean values of each cluster?

```
km$centers
```

```
          x         y
1 -2.732907  2.764193
2  2.764193 -2.732907
```

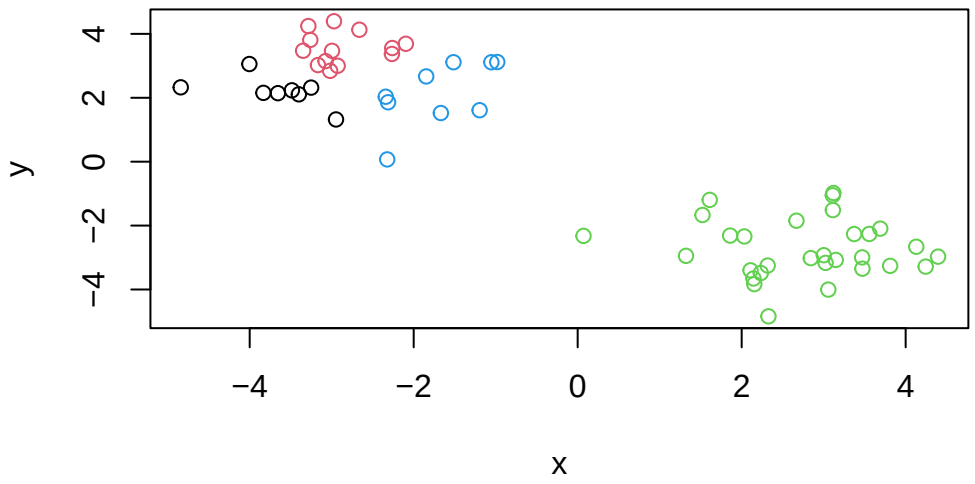
Q. Make a plot of your data showing your clustering results (groupings/clusters and cluster centers)

```
plot(x, col = km$cluster)
points(km$centers, col = "blue", pch = 15)
```



Q. Run `kmeans()` again and cluster in 4 groups and plot results

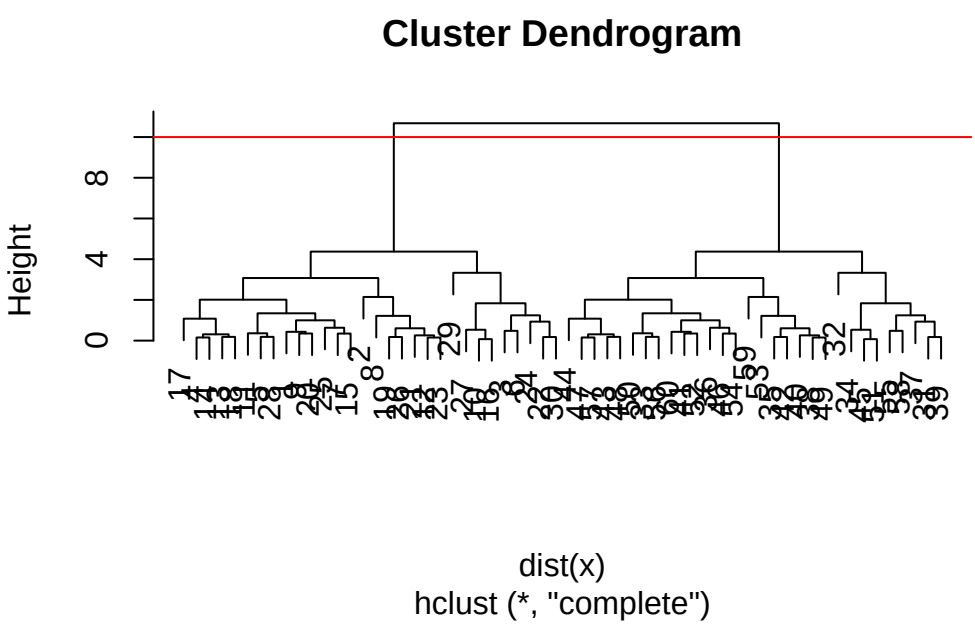
```
km2 <- kmeans(x, centers = 4)
plot(x, col = km2$cluster)
```



Hierarchical Clustering

reveal structure in data by grouping points into a smaller number of clusters `hclust()` this function does not take our imput data directly but wants a "distance matrix" that details how (dis)similar all our input points are to each other.

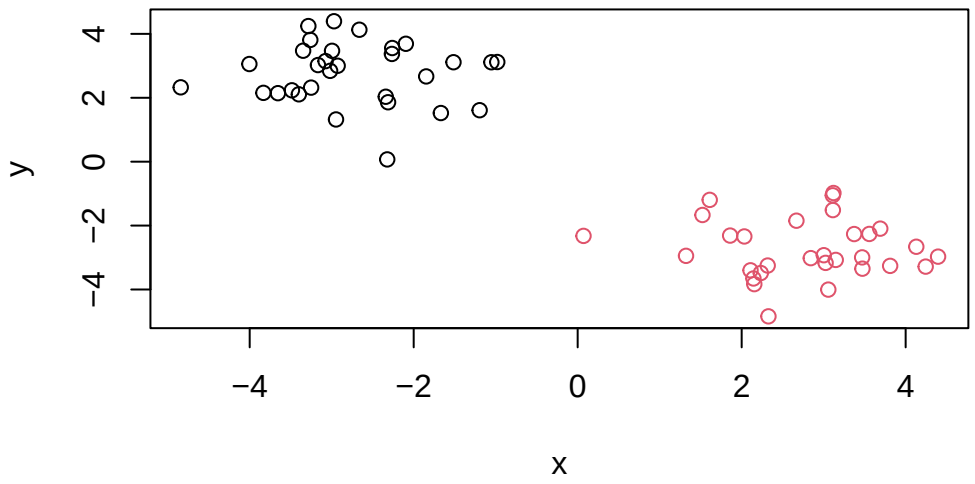
```
hc <- hclust(dist(x)) # dist() measures distance pairwise between each point
plot(hc)
abline(h = 10, col = "red")
```



to get my main result, main cluster

```
grps <- cutree(hc, h = 10)
```

```
plot(x, col = grps)
```



Principal Component Analysis

```
url <- "https://tinyurl.com/UK-foods"
x <- read.csv(url)
```

Q1. How many rows and columns are in your new data frame named x? What R functions could you use to answer this questions?

```
dim(x)
```

```
[1] 17  5
```

```
head(x)
```

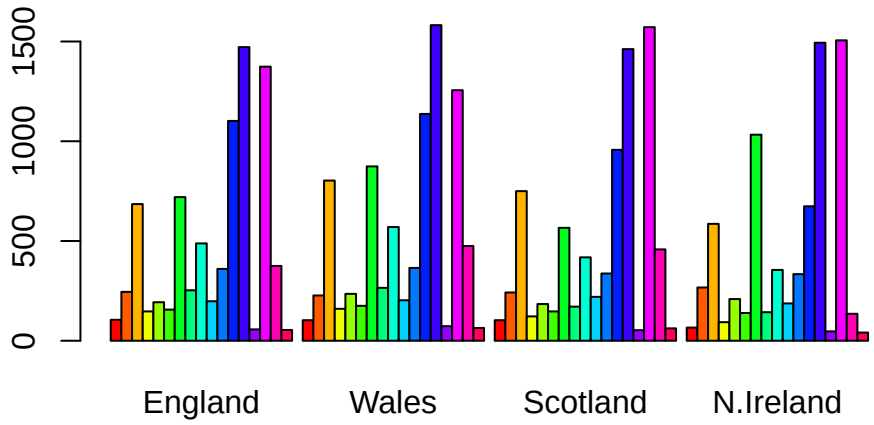
```
              X England Wales Scotland N.Ireland
1        Cheese     105   103      103        66
2  Carcass_meat     245   227      242       267
3    Other_meat     685   803      750       586
4          Fish     147   160      122        93
```

```
5 Fats_and_oils      193    235      184       209
6         Sugars      156    175      147       139
```

Q2. Which approach to solving the 'row-names problem' mentioned above do you prefer and why? Is one approach more robust than another under certain circumstances?

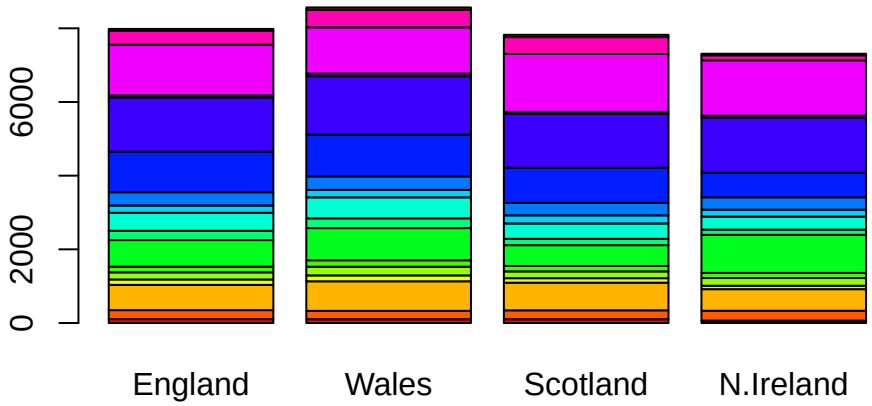
```
x <- read.csv(url, row.names = 1)
```

```
barplot(as.matrix(x), beside=T, col=rainbow(nrow(x)))
```



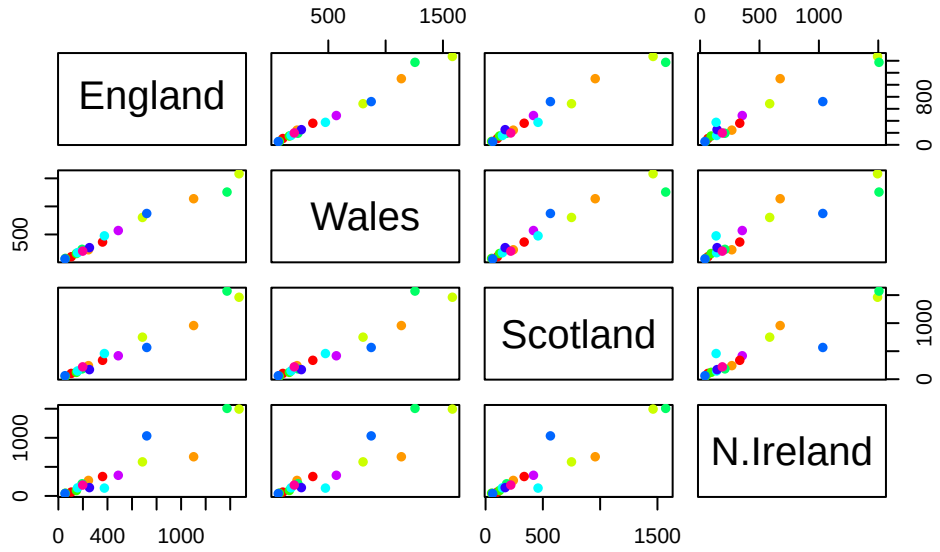
Q3: Changing what optional argument in the above barplot() function results in the following plot?

```
barplot(as.matrix(x), beside=F, col=rainbow(nrow(x)))
```



Q5: Generating all pairwise plots may help somewhat. Can you make sense of the following code and resulting figure? What does it mean if a given point lies on the diagonal for a given plot?

```
pairs(x, col=rainbow(10), pch=16)
```



if the points lie on the diagonal, there is an association between the country on the x and y axes

hard to understand even for small data set, so let's use PCA `prcomp()`

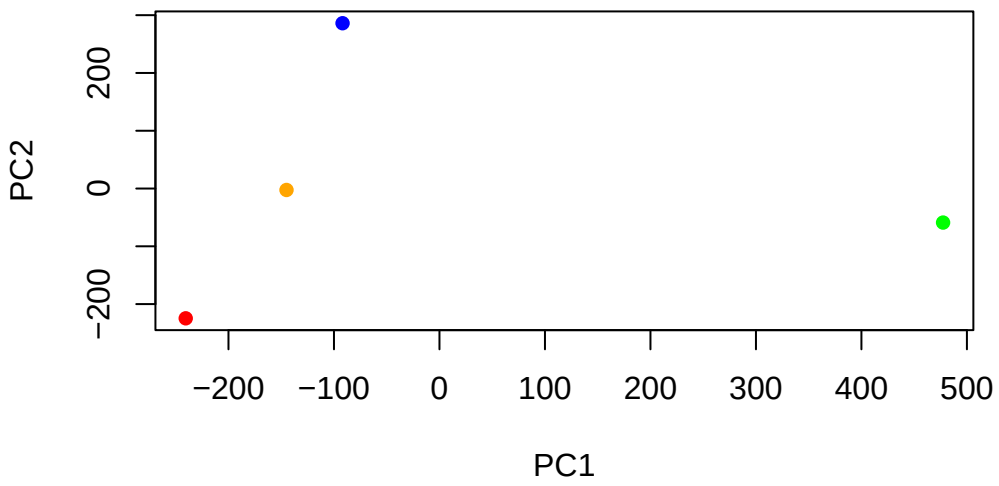
```
pca <- prcomp(t(x))
summary(pca)
```

```
Importance of components:
                            PC1      PC2      PC3       PC4
Standard deviation     324.1502 212.7478 73.87622 3.176e-14
Proportion of Variance   0.6744   0.2905  0.03503 0.000e+00
Cumulative Proportion    0.6744   0.9650  1.00000 1.000e+00
```

```
pca$x
```

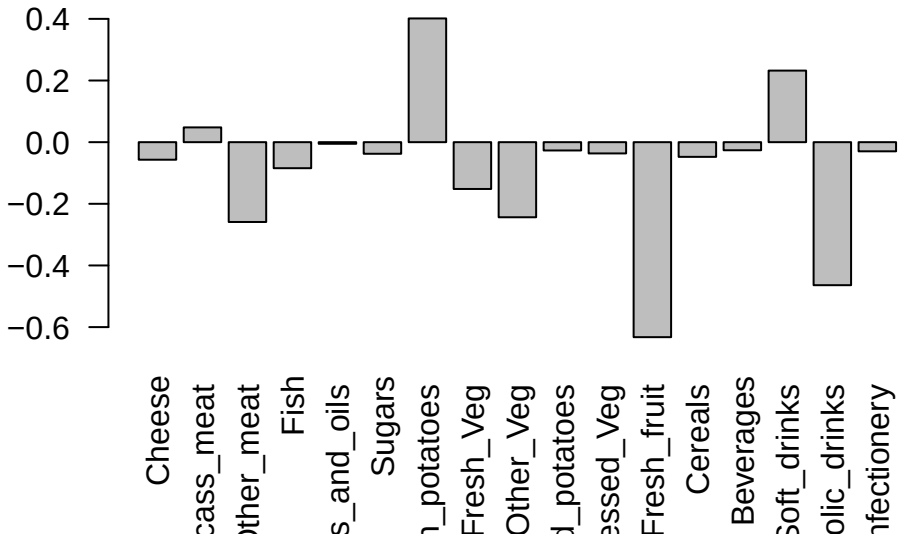
```
                PC1         PC2        PC3           PC4
England   -144.99315   -2.532999 105.768945 -4.894696e-14
Wales     -240.52915 -224.646925 -56.475555  5.700024e-13
Scotland   -91.86934  286.081786 -44.415495 -7.460785e-13
N.Ireland  477.39164  -58.901862  -4.877895  2.321303e-13
```

```r
colors <- c("orange", "red", "blue", "green")
plot(pca$x[,1], pca$x[,2], col = colors, pch = 16,
     xlab = "PC1", ylab = "PC2")
```



the "rotation" component tells us how much the original variables contribute to the new PCs

```r
barplot( pca$rotation[,1], las=2 )
```



PCA is useful for gaining insight into high dimensional data that is difficult to examine in other ways