class7_clustering_and_pca

David Ma

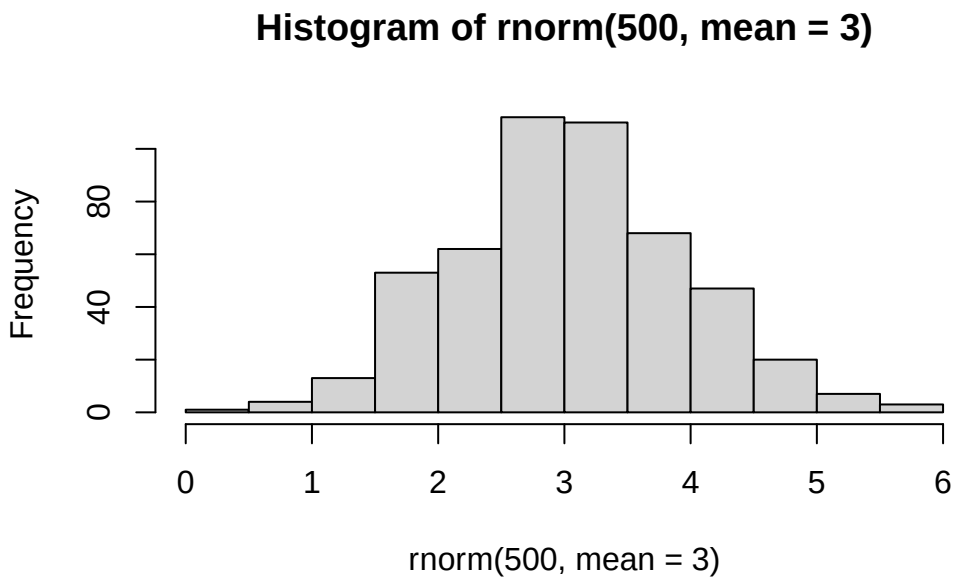
Clustering

We can use `rnorm()` function to get random numbers around a normal distribution around a given `mean`.

```
# Creating a normal distribution
rnorm(15)
```

```
 [1] -0.3098466  0.4085639 -0.9747739 -0.8824962 -0.1643940 -0.7714472
 [7] -0.5877213  0.1176065 -1.1944430  1.9813439  1.1339639 -1.0779714
[13] -1.3324098 -0.5511356  0.7148234
```

```
# Creating histograms of a normal distribution around a given mean
hist( rnorm(500, mean = 3))
```



For 30 points with a mean of 3 and -3:

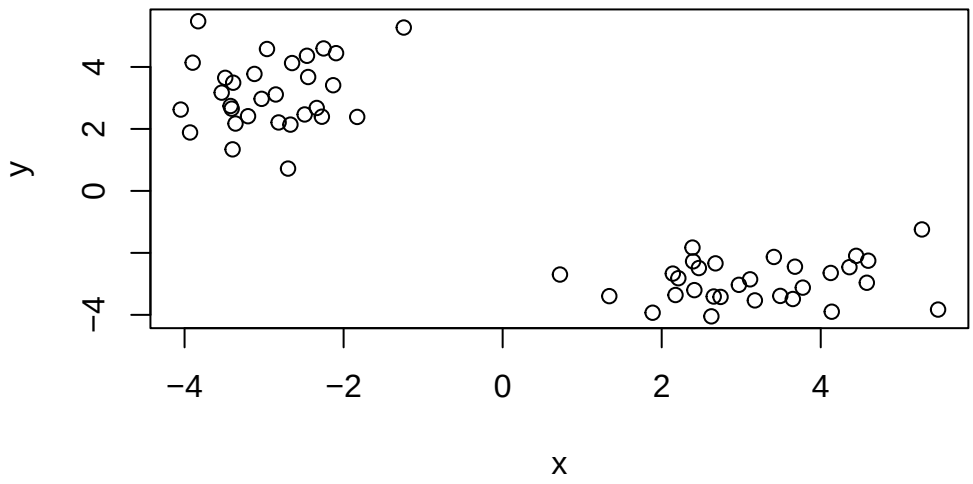
```
c(rnorm(30, mean = 3), rnorm(30, mean = -3))
```

```
 [1]  3.583968  4.034051  3.688673  4.007489  3.767139  1.793775  3.754956
 [8]  2.376047  2.818405  3.712174  2.718872  1.472650  2.444981  3.169642
[15]  2.754872  4.004947  3.670301  3.950462  1.841681  2.849258  2.473323
[22]  3.544672  3.602902  1.273280  4.319420  2.664497  1.534684  2.147134
[29]  2.768336  4.290680 -3.569709 -2.636965 -2.649003 -2.858358 -3.558958
[36] -3.127669 -3.632878 -2.938856 -2.714940 -1.354439 -2.561094 -3.676179
[43] -4.603782 -4.364091 -3.349084 -3.528944 -2.483600 -2.607692 -2.846106
[50] -2.985987 -2.513796 -2.251724 -3.411291 -4.696540 -4.191590 -5.200585
[57] -1.792775 -3.149812 -3.308138 -1.666350
```

Those same points, but putting them together:

```
data <- c(rnorm(30, mean = 3), rnorm(30, mean = -3))
combined_data <- cbind(x = data, y = rev(data))

# What does it look like?
plot(combined_data)
```



K-means clustering

Popular clustering method (especially for big datasets) that we can use with `kmeans()` function in base R.

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

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

Cluster means:
          x         y
1  3.169197 -2.908822
2 -2.908822  3.169197

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] 50.32017 50.32017
 (between_SS / total_SS =  91.7 %)
```

```
Available components:

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

kmeans() practice

```
# What is the size?
km$size
```

```
[1] 30 30
```

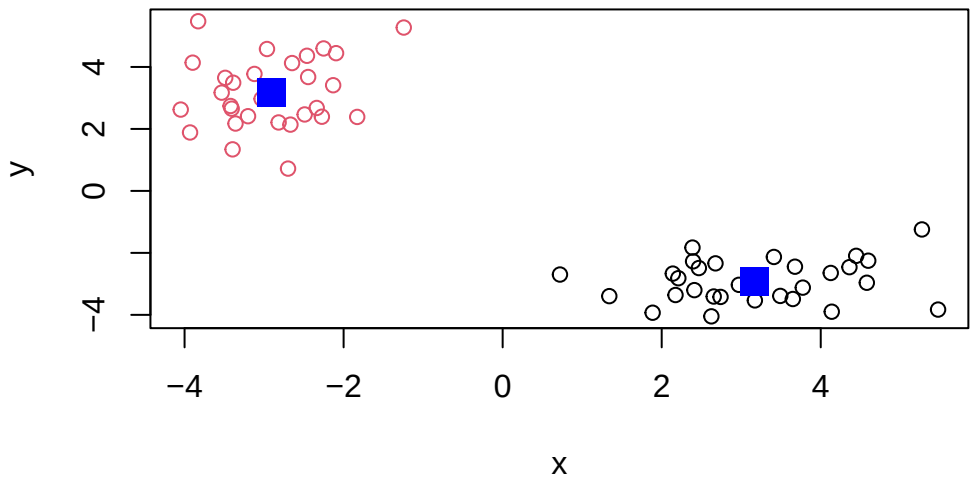
```
# What is the cluster/assignment?
km$cluster
```

```
 [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
```

```
# What is the cluster center?
km$centers
```

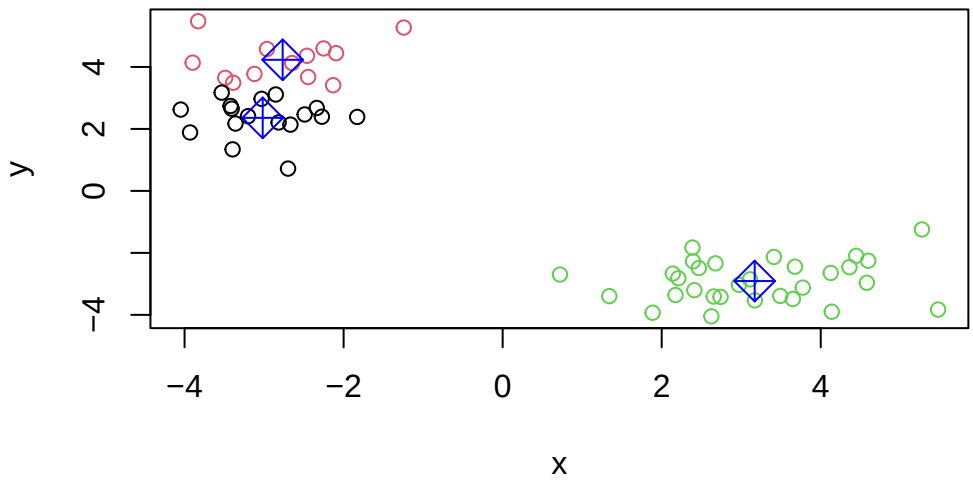
```
          x         y
1  3.169197 -2.908822
2 -2.908822  3.169197
```

```
# Plot x colored by the kmeans cluster assignment and add cluster centers as blue points
plot(combined_data, col = km$cluster)
points(km$centers, col = "blue", pch = 15, cex = 2)
```



Q. Clustering into 3 groups on same `combined_data` and make a plot:

```
km3 <- kmeans(combined_data, centers = 3)
plot(combined_data, col = km3$cluster)
points(km3$centers, col = "blue", pch = 9, cex = 2)
```



Hierarchical Clustering

We can use `hclust()` for hierarchical clustering. Unlike `kmeans()` where we could just pass in our data as input, we need to give `hclust()` a "distance matrix".

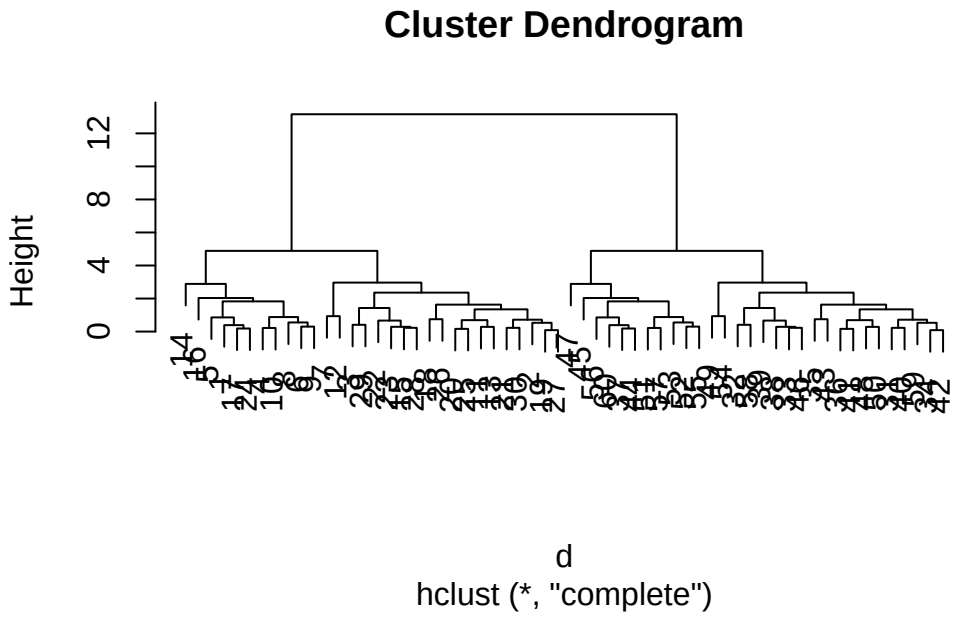
We will use the `dist()` function to start with.

```
d <- dist(combined_data)
hc <- hclust(d)
hc
```

```
Call:
hclust(d = d)

Cluster method   : complete
Distance         : euclidean
Number of objects: 60
```

```
plot(hc)
```



I can now "cut" my tree with the `cutree()` to yield a cluster membership vector.

```
grps <- cutree(hc, h = 8)
```

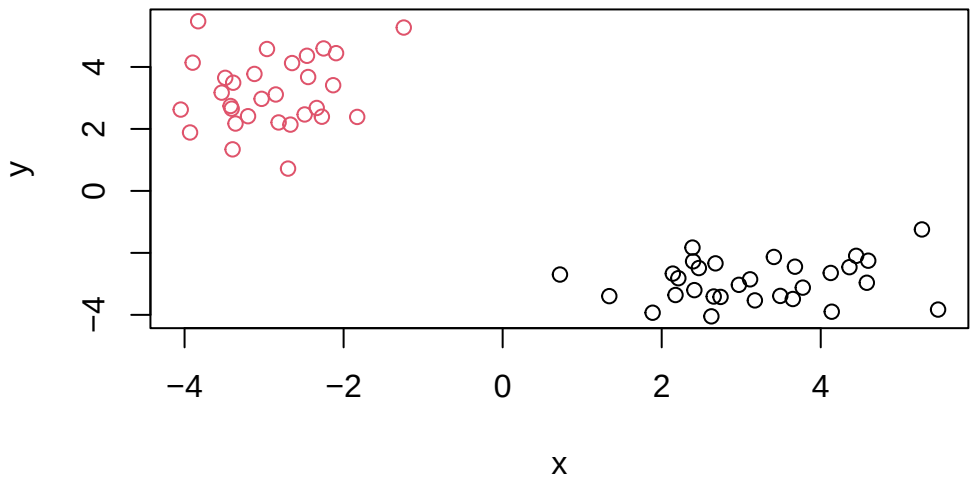
You can also tell `cutree()` to cut where it yields "k" groups.

```
cutree(hc, k = 2)
```

```
 [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
```

Plotting these points with color:

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



Principal Componenet Analysis (PCA)

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

# Q1. How many rows and columns are there in this dataframe?
dim(x)
```

```
[1] 17  5
```

```
# Checking our data, noticing how the row names are incorrectly set as the first column
head(x)
```

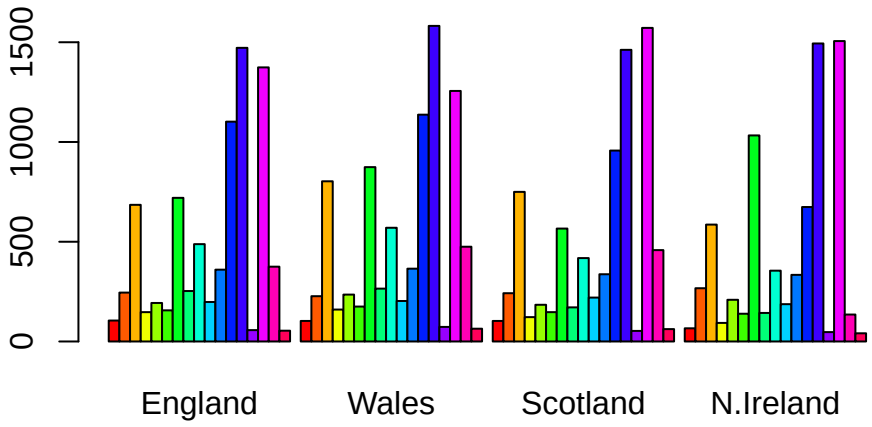
```
              X England Wales Scotland N.Ireland
1        Cheese     105   103      103        66
2  Carcass_meat     245   227      242       267
3    Other_meat     685   803      750       586
4          Fish     147   160      122        93
```

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

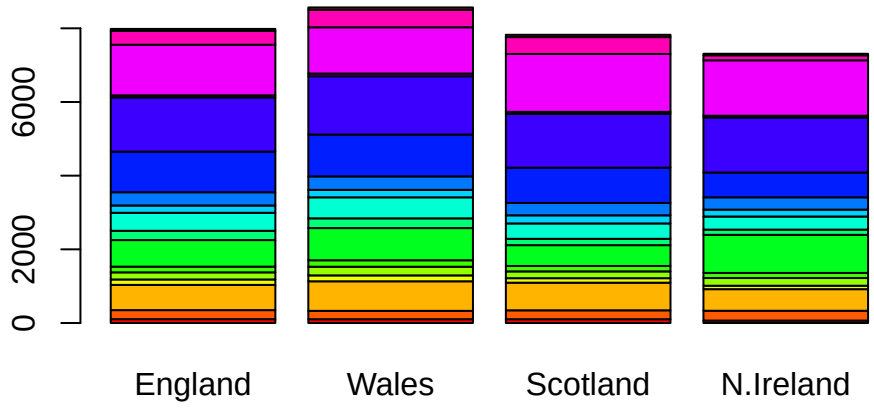
```r
# Q2. Let's fix this by setting the first column. Note that this method is preferred, as i
x <- read.csv(url, row.names=1)
head(x)
```

```
               England Wales Scotland N.Ireland
Cheese             105   103      103        66
Carcass_meat       245   227      242       267
Other_meat         685   803      750       586
Fish               147   160      122        93
Fats_and_oils      193   235      184       209
Sugars             156   175      147       139
```

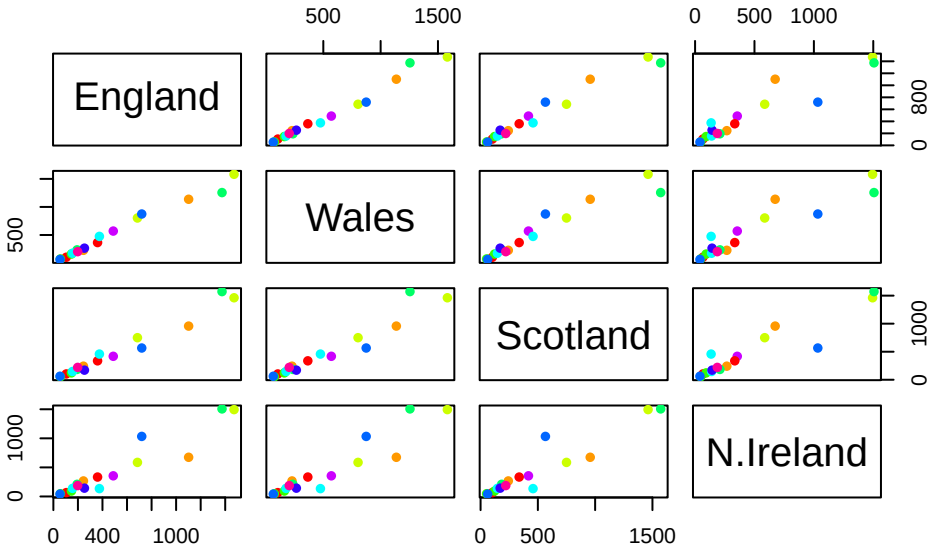
```r
# Beginning to visualize the data
barplot(as.matrix(x), beside=T, col=rainbow(nrow(x)))
```




```r
# Q3. Leaving out the `beside` argument will generate the desired plot, as it is FALSE by
barplot(as.matrix(x), col=rainbow(nrow(x)))
```




```
# Q5. Generating pairwise plots. From the top left box going horizontal, England is the Y
pairs(x, col=rainbow(10), pch=16)
```

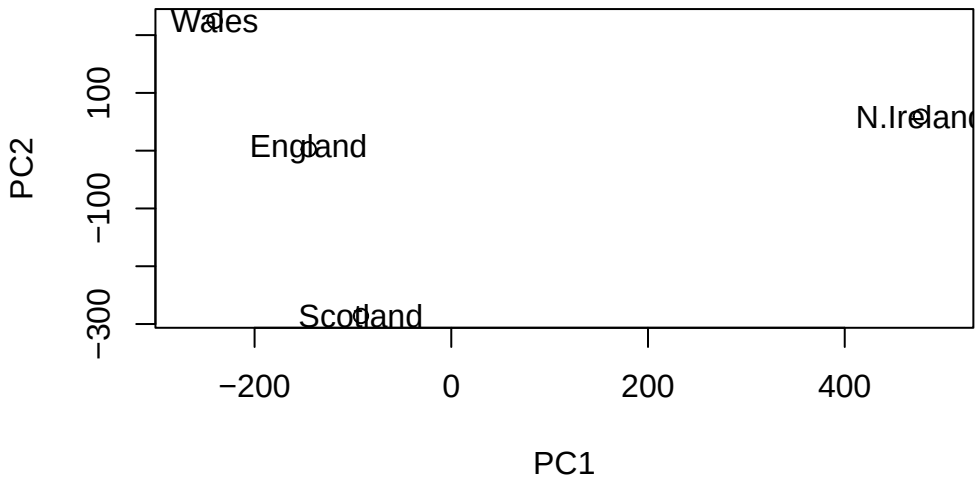


Q6. Looking at N. Ireland, their cyan, blue, and orange dots tend to not be on the diagonal compared to other pairings.

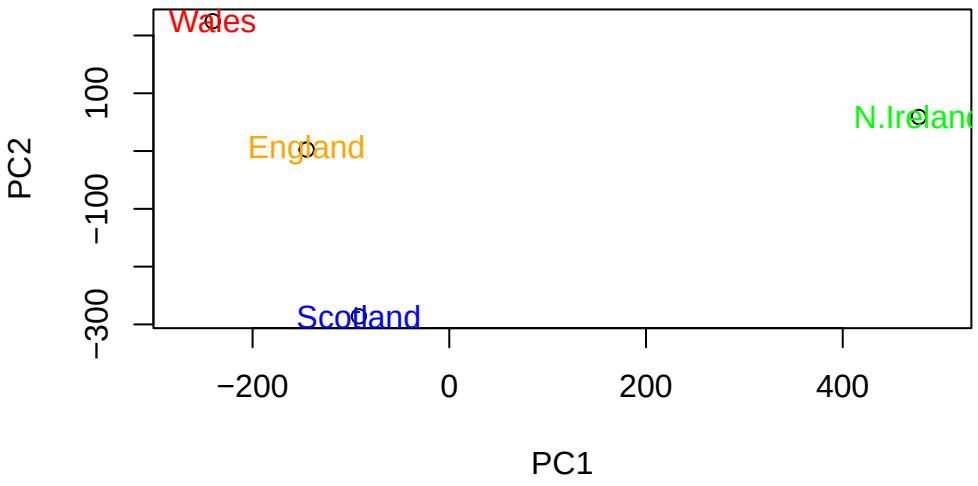
```
# Making sense with PCA
pca <- prcomp( t(x) )
summary(pca)
```

```
Importance of components:
                            PC1      PC2      PC3       PC4
Standard deviation     324.1502 212.7478 73.87622 4.189e-14
Proportion of Variance   0.6744   0.2905  0.03503 0.000e+00
Cumulative Proportion    0.6744   0.9650  1.00000 1.000e+00
```

```
# Q7. PC1 vs PC2 visualization
plot(pca$x[,1], pca$x[,2], xlab="PC1", ylab="PC2", xlim=c(-270,500))
text(pca$x[,1], pca$x[,2], colnames(x))
```




```
# Coloring them in
plot(pca$x[,1], pca$x[,2], xlab="PC1", ylab="PC2", xlim=c(-270,500))
text(pca$x[,1], pca$x[,2], colnames(x), col = c("orange", "red", "blue", "green"))
```


Wales
England
N.Ireland
Scotland
PC2
100
−100
−300
−200
0
200
400
PC1