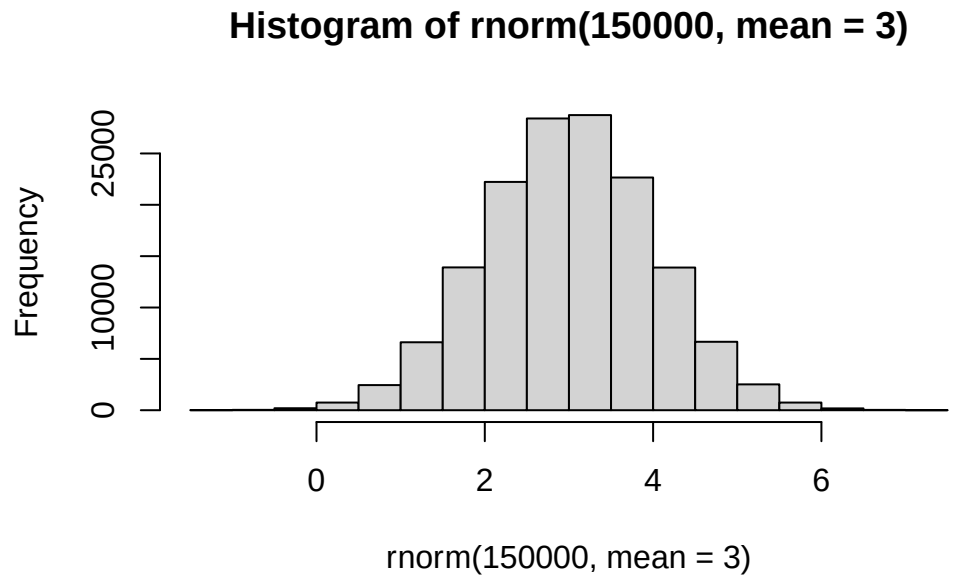
Class 7: Machine Learning 1

Abel (PID 59018056)

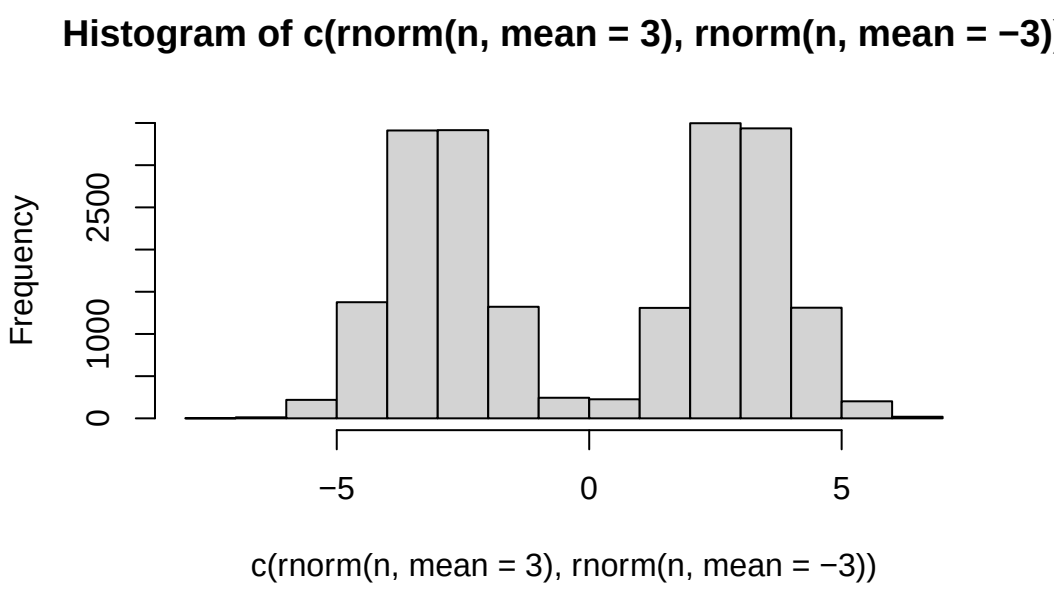
Before we get into clustering methods let's make some sample data to cluster where we know what the aswer should be.

To help with this, I will use the `rnomrm()` function

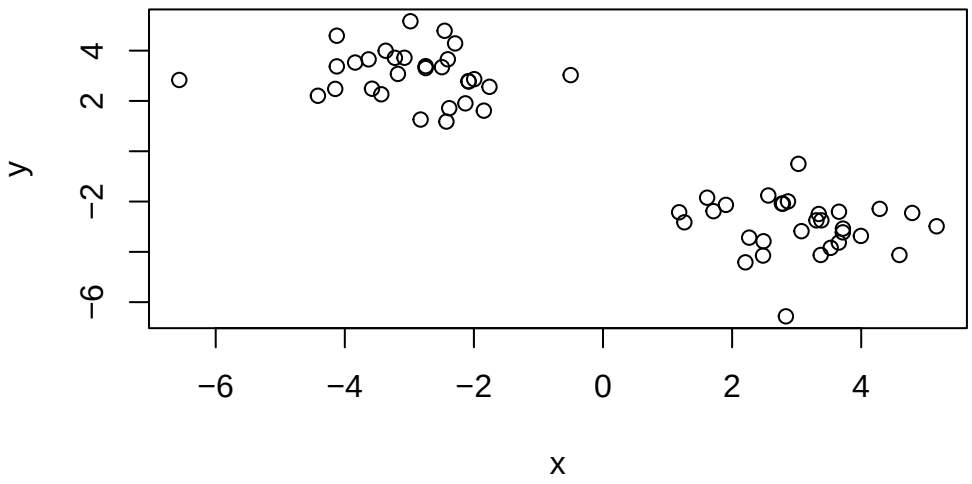
```
hist(rnorm(150000, mean = 3))
```




```
n = 10000
hist(c(rnorm(n, mean = 3), rnorm(n, mean=-3)))
```




```
n = 30
x <- c(rnorm(n, mean = 3), rnorm(n, mean=-3))
y <- rev(x)


z <- cbind(x, y)
plot(z)
```



K-mean clustering

The function in base R for K-means clustering is called `kmeans()`.

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

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

Cluster means:
          x         y
1  3.050938 -2.962040
2 -2.962040  3.050938

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] 64.04112 64.04112
 (between_SS / total_SS =  89.4 %)
```

```
Available components:

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

```
km$centers
```

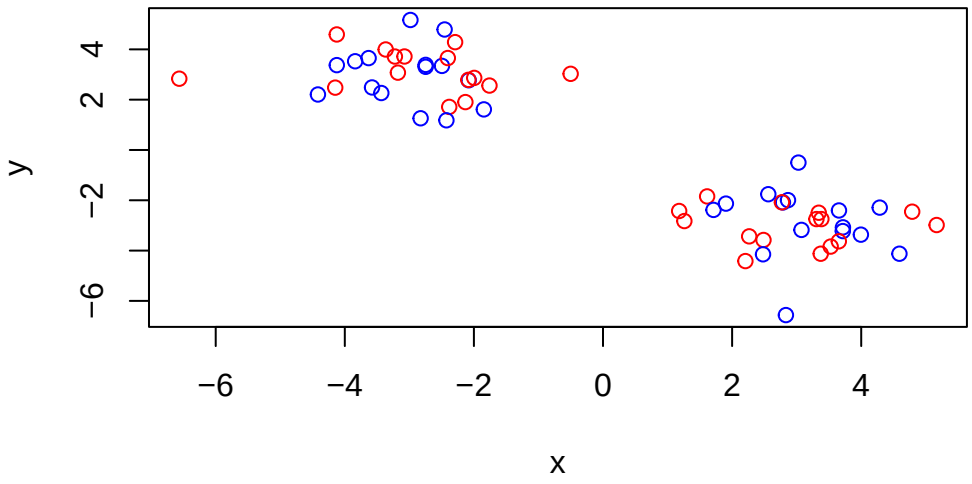
```
          x         y
1  3.050938 -2.962040
2 -2.962040  3.050938
```

Q. Print out the cluster membership vectro (i.e our main answer)

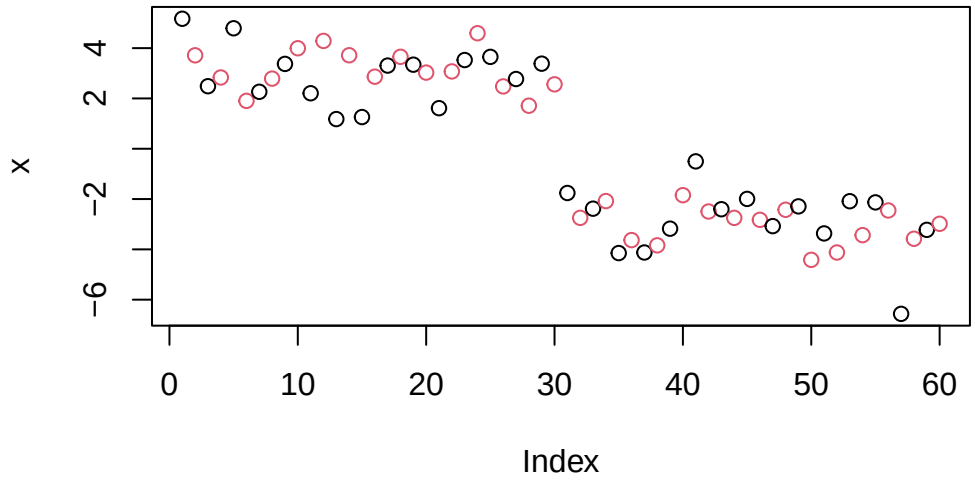
```
km$cluster
```

```
 [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 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
```

```
plot(z, col = c("red", "blue"))
```

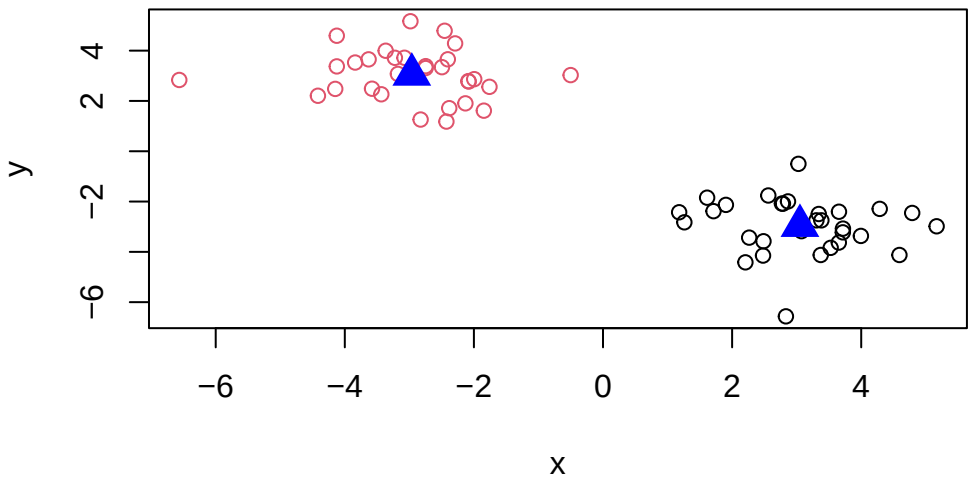


```
plot(x, col = c(1,2))
```



Plot with clustering result and add cluster centers:

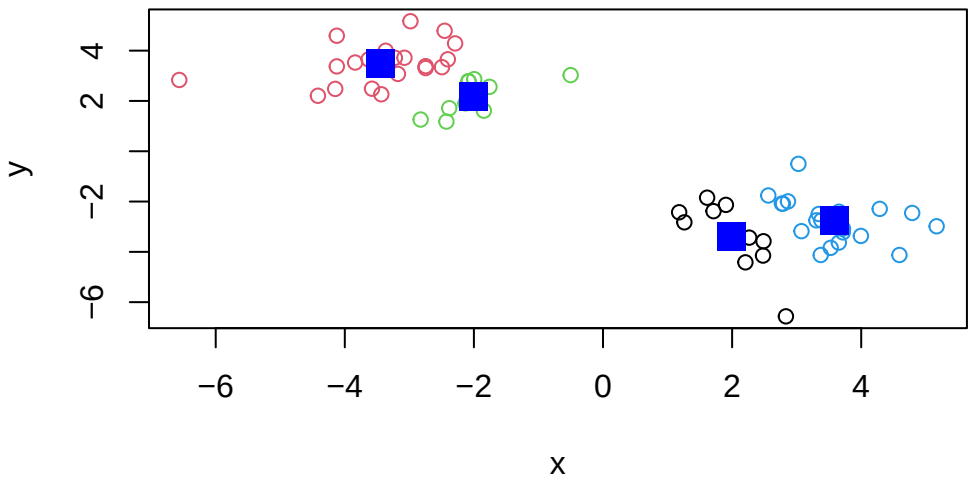
```
plot(z, col = km$cluster)
points(km$centers, col = "blue", pch = 17, cex = 2)
```



phc = different shapes # cex = character ambelishment

Q. Can you cluster our data in z into four cluster please

```
km4 <- kmeans(z, centers = 4)
plot(z, col = km4$cluster)
points(km4$centers, col = "blue", pch = 15, cex = 2)
```



Hierarchal Clustering

The manin function for hierarchal clustering is `hclust()`

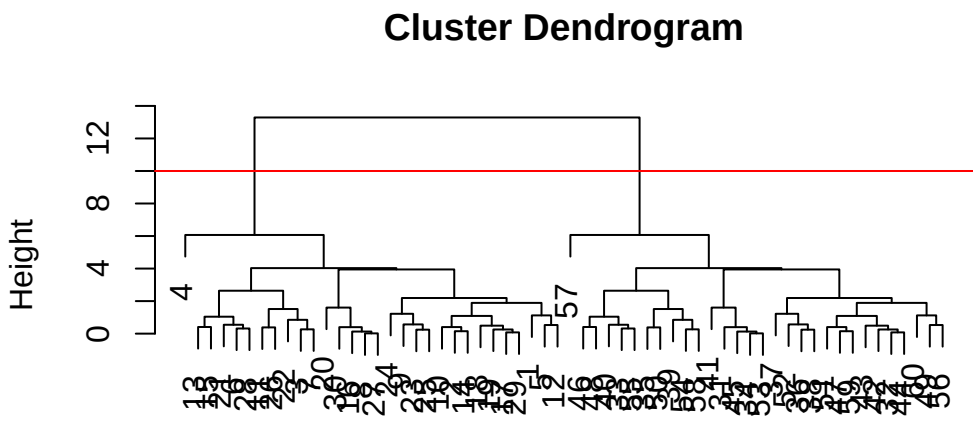
Unlike `kmeans()` I cannot just pass in my data as input. I first need a distance matrix from my data

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

```
Call:
hclust(d = d)

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

There is a specific hclust plot() method...

```
plot(hc)
abline(h = 10, col = "red")
```

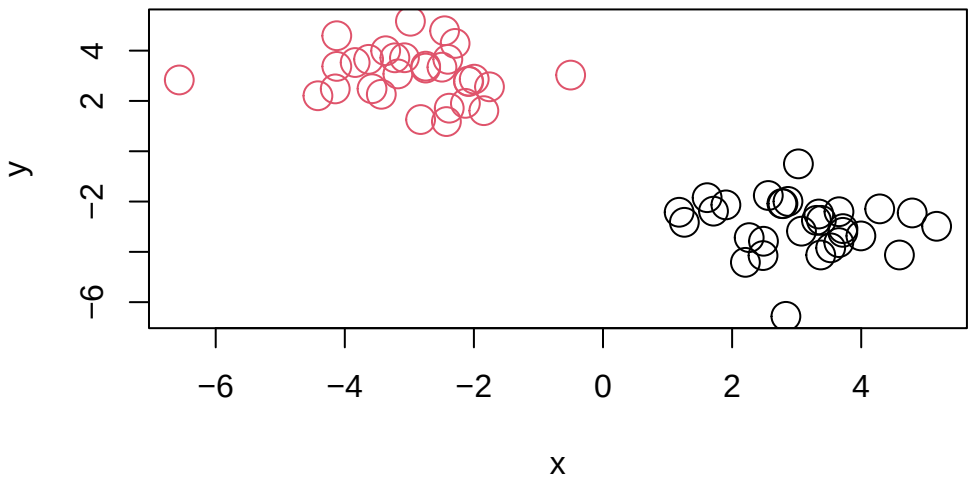


To get my main clusterign result (i.e the membership vector) I can "cut" my tree at a given height. To do this I will use the `cutreee()`

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

```
 [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
```

```
plot(z, col = grps, pch = 1, cex = 2)
```




```
url <- "https://tinyurl.com/UK-foods"
x <- read.csv(url, row.names = 1)
nrow(x)
```

```
[1] 17
```

```
ncol(x)
```

```
[1] 4
```

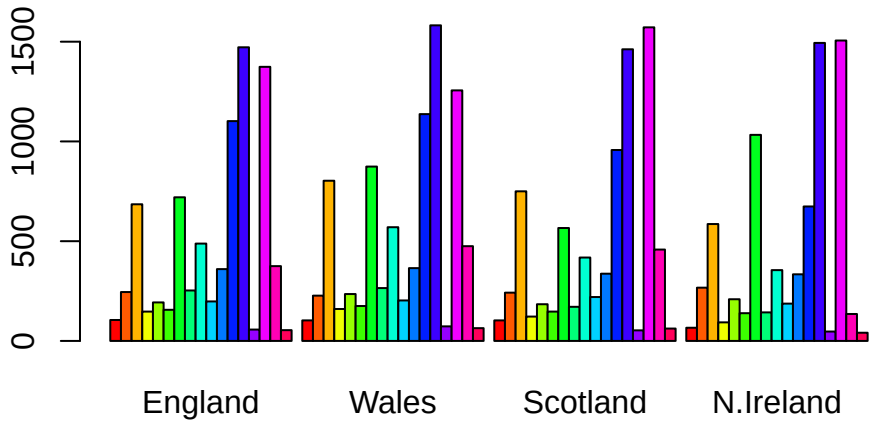
```
 ## x <- x[,-1]

##head(x)
```

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

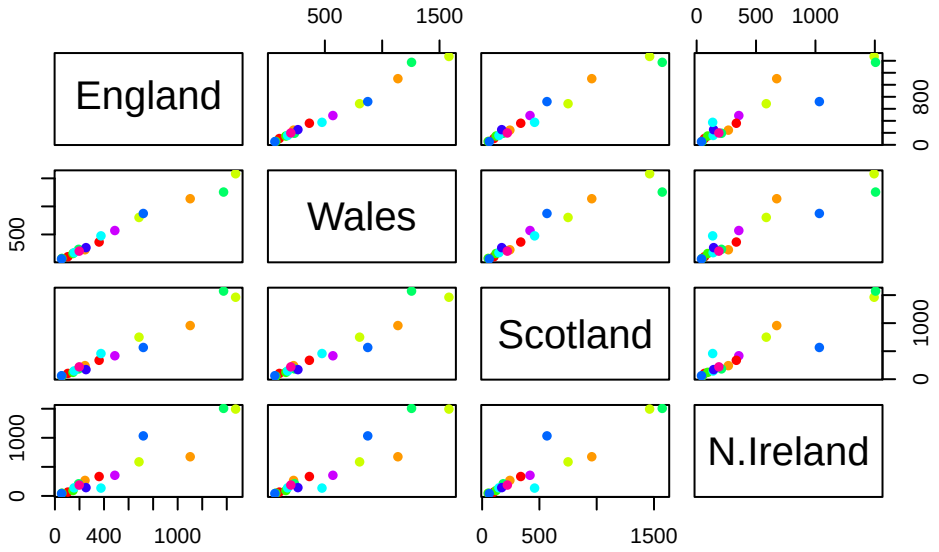
```
              England Wales Scotland N.Ireland
Cheese            105   103      103        66
Carcass_meat      245   227      242       267
```

```
Other_meat          685    803       750        586
Fish                147    160       122         93
Fats_and_oils       193    235       184        209
Sugars              156    175       147        139
```

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




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



Principal Component Analysis (PCA)

The main function to do PCA in base R is called `prcomp()`.

Note that I need to take the transpose of this particlular data as that is what the `prcomp()` help page was asking for.

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

```
Importance of components:
                            PC1      PC2      PC3       PC4
Standard deviation     324.1502 212.7478 73.87622 2.921e-14
Proportion of Variance   0.6744   0.2905  0.03503 0.000e+00
Cumulative Proportion    0.6744   0.9650  1.00000 1.000e+00
```

Let's see what is inside our result object `pca` that wwe just calulated

```
attributes(pca)
```

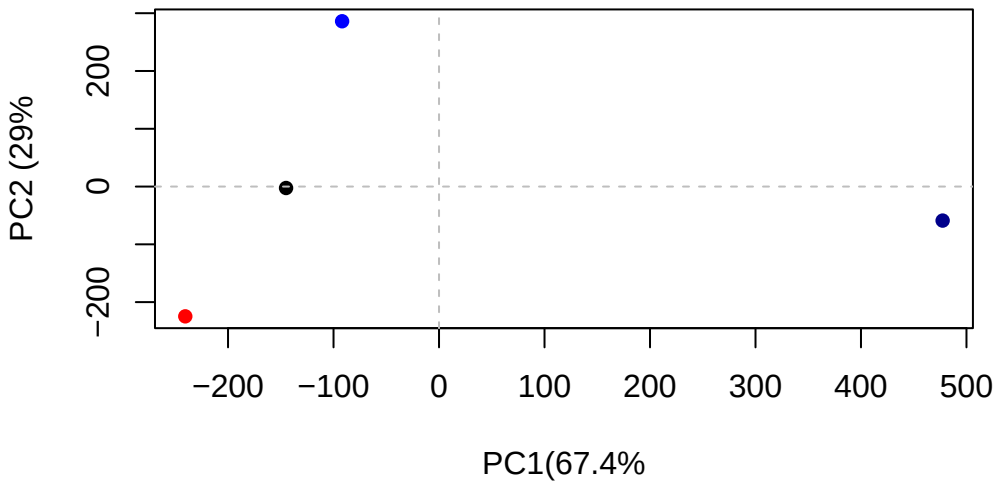
```
$names
[1] "sdev"     "rotation" "center"   "scale"    "x"

$class
[1] "prcomp"
```

```
pca$x
```

```
                PC1         PC2        PC3           PC4
England   -144.99315   -2.532999 105.768945 -9.152022e-15
Wales     -240.52915 -224.646925 -56.475555  5.560040e-13
Scotland   -91.86934  286.081786 -44.415495 -6.638419e-13
N.Ireland  477.39164  -58.901862  -4.877895  1.329771e-13
```

To make our main result figure, called a "PC plot" (or "score plot", "ordination plot", or "PC1 vs PC2 plot").

```
plot(pca$x[,1], pca$x[,2], col = c("black", "red", "blue", "darkblue"), pch = 16, xlab = "PC
abline(h = 0, col = "gray", lty = 2)
abline(v = 0, col = "gray", lty = 2)
```



Variable Loading plot

Can give us insight on how the original variable

```
par(mar=c(10, 3, 0.35, 0))
barplot( pca$rotation[,1], las=2 )
```