Class7: Machine Learning 1

Alice Lai (PID:A16799081)

Clustering

Finding groups/patterns in data, and then dimensionalilty deduction.

"k-means" clustering The main function in base R for this is `kmeans()`.

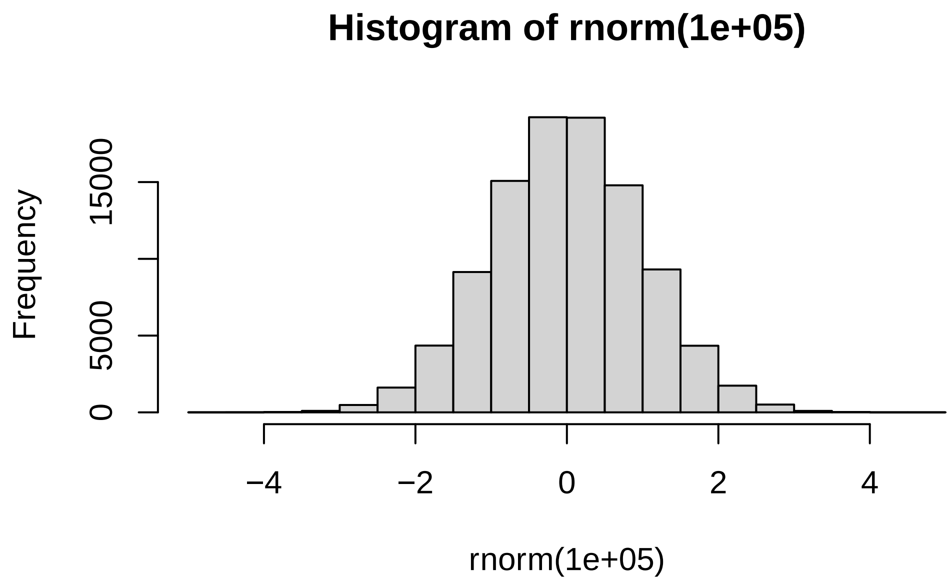
```
# make up some data
hist(rnorm(100000), mean = 3)
```

```
Warning in plot.window(xlim, ylim, "", ...): "mean" is not a graphical
parameter
```

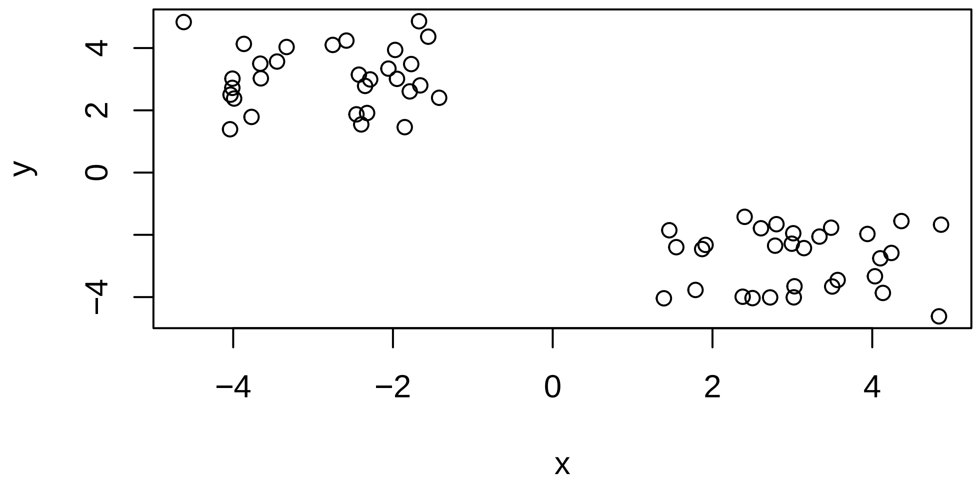
```
Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...): "mean"
is not a graphical parameter
```

```
Warning in axis(1, ...): "mean" is not a graphical parameter
```

```
Warning in axis(2, at = yt, ...): "mean" is not a graphical parameter
```




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




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

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

Cluster means:
          x         y
1 -2.790322  3.058645
2  3.058645 -2.790322

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 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

Within cluster sum of squares by cluster:
[1] 54.90491 54.90491
 (between_SS / total_SS =  90.3 %)

Available components:

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

How many points in each cluster?

```
km$size
```

```
[1] 30 30
```

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 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
```

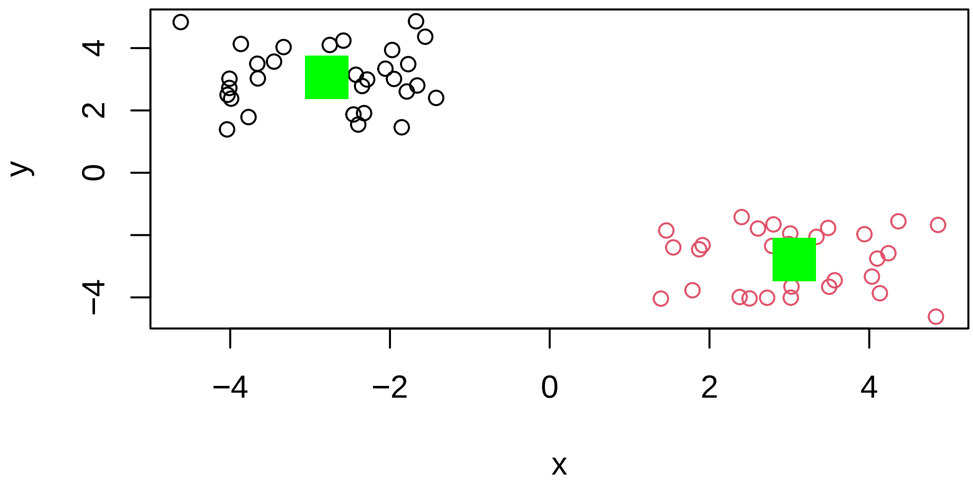
What are centers/mean values of each cluster?

```
km$centers
```

```
          x         y
1 -2.790322  3.058645
2  3.058645 -2.790322
```

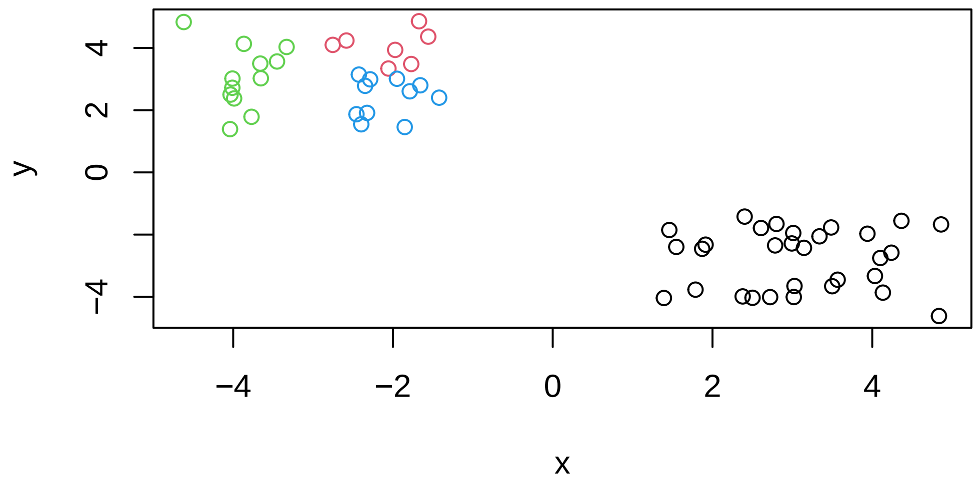
Make a plot of your data showing your clustering results (groupings/clusters and cluster centers)

```
plot(x, col=km$cluster)
points(km$centers, col="green", pch=15, cex=3)
```



Run `kmeans()` again and cluter into 4 groups and plot the results

```
km4 = kmeans(x, centers=4)
plot(x, col=km4$cluster)
```



Hierarchical Clustering

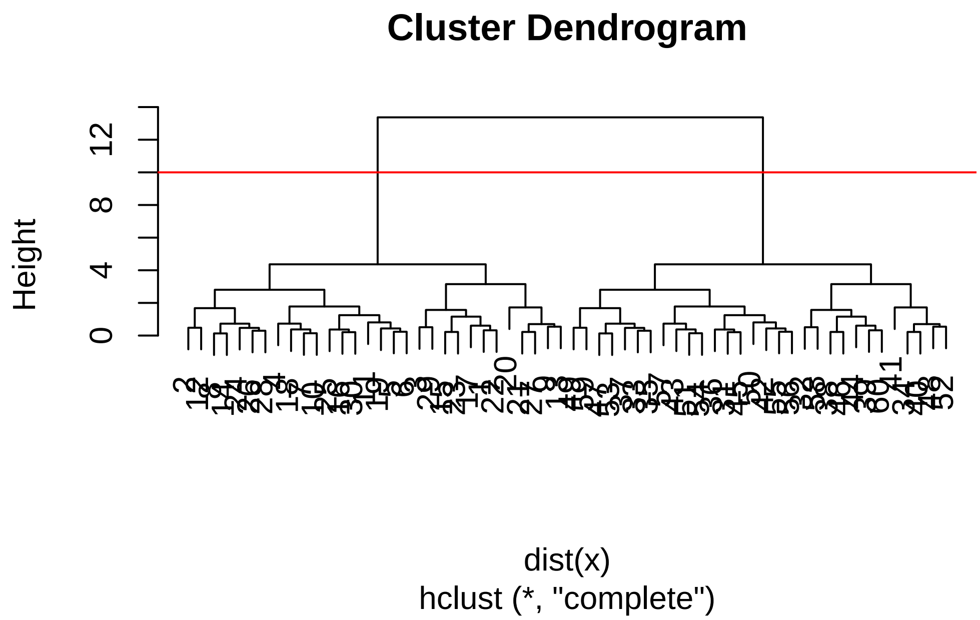
This form of clustering aims to reveal the structure in your data by progressively grouping points into a ever smaller number of clusters.

The main function in base R for this is called `hclust()`. This function does not take our input data directly but wants a “distance matrix” that details how (dis)similar are our input points to one another.

```
hc <- hclust(dist(x))
```

The print out above is not very useful (unclick that from kmeans) but there is a useful `plot()` method.

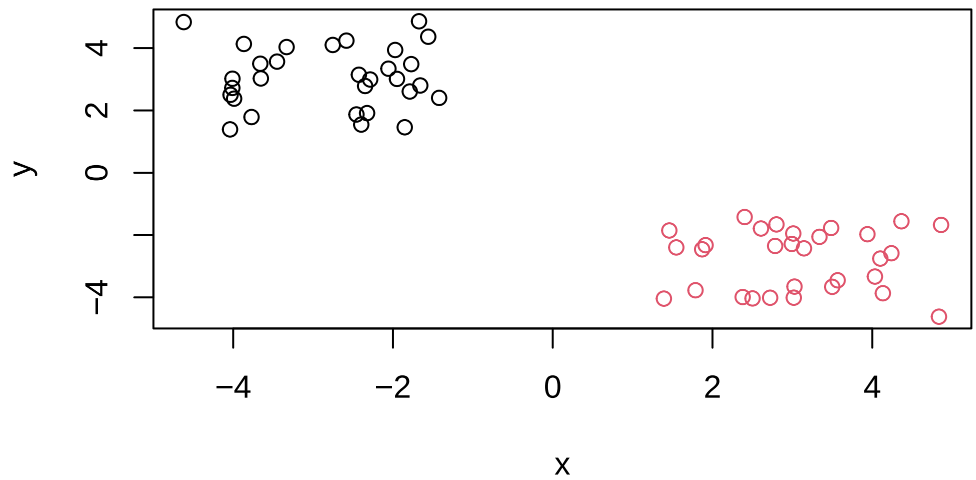
```
plot(hc)
abline(h = 10, col = "red")
```



To get my main result (my cluster membership vector) I need to "cut" my tree using the function `cutree()`

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

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



Principal Component Analysis (PCA)

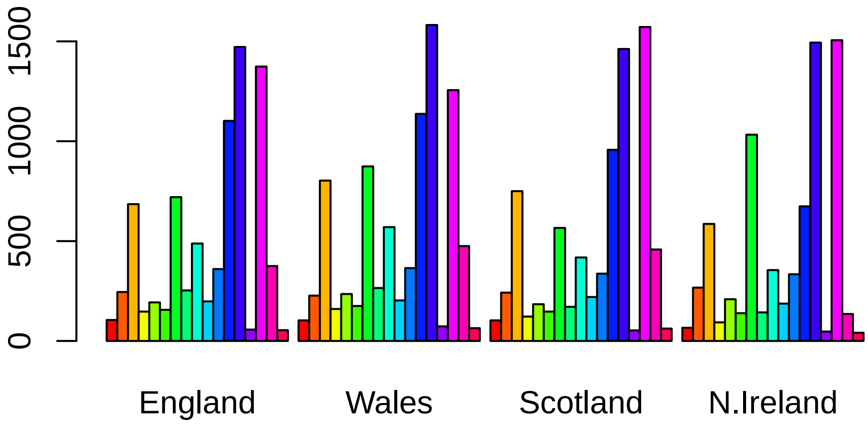
```
url <- "https://tinyurl.com/UK-foods"
x <- read.csv(url, row.names=1)

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
```

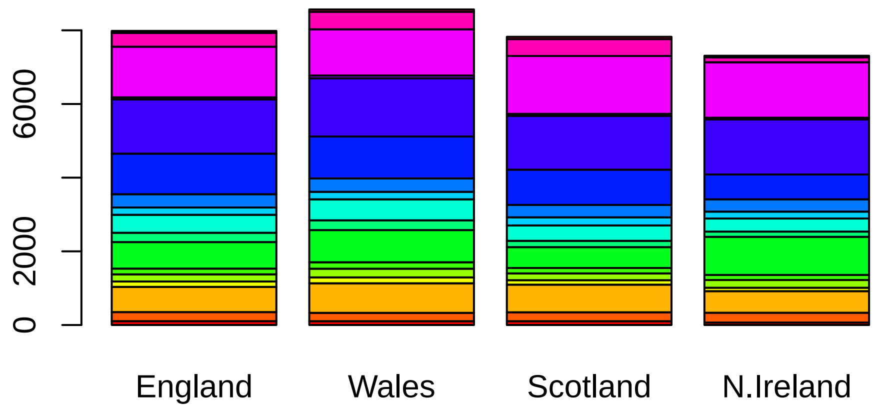
```
dim(x)
```

```
[1] 17  4
```

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

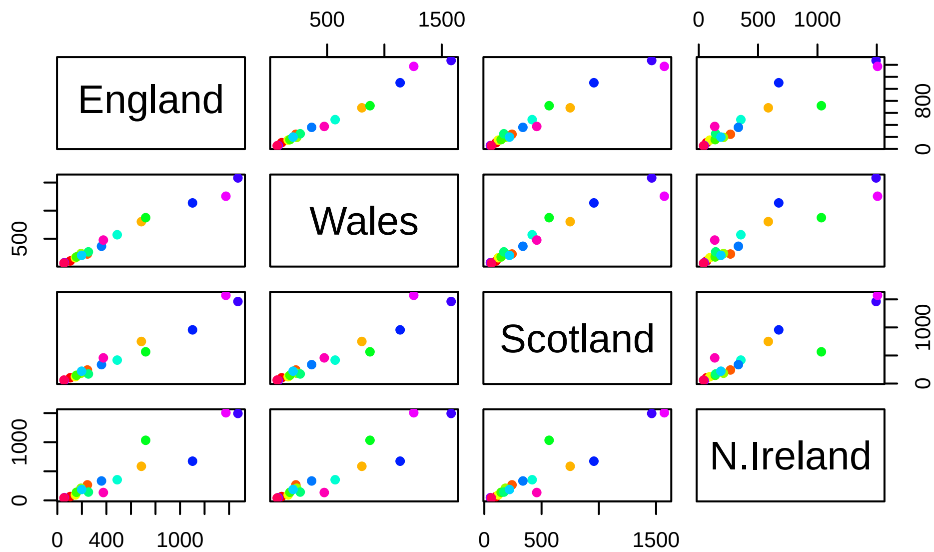



```r
barplot(as.matrix(x), beside=F, col=rainbow(nrow(x)))
```



The so-called “pairs” plot can be useful for small dataset:

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



So the pairs plot is useful for small datasets but it can be lots of work to interpret and gets intractabke for larger datasets.

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

```
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
```

```
attributes(pca)
```

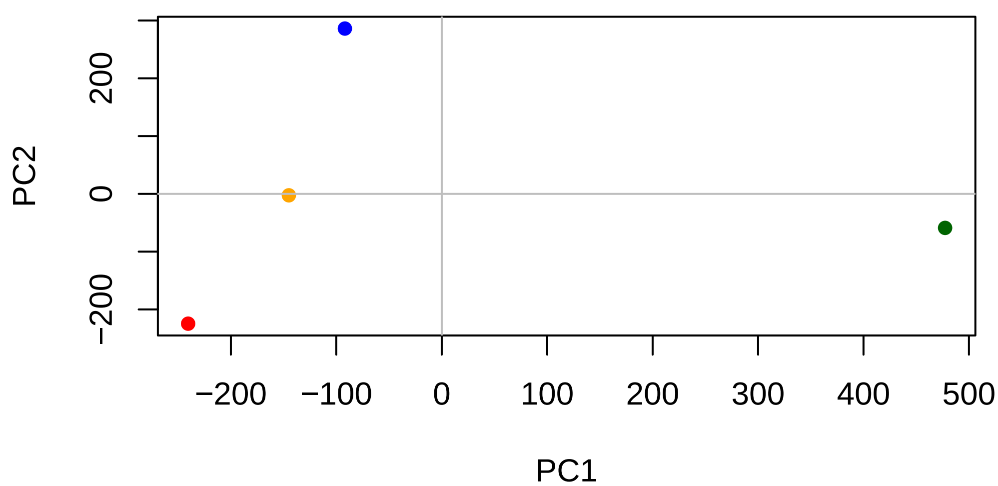
```
$names
[1] "sdev"     "rotation" "center"   "scale"    "x"

$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
```

A major PCA result visualization is called a "PCA plot" (aka a score plot, biplot, PC1 vs PC2 plot, ordination plot)

```
mycols <- c("orange", "red", "blue", "darkgreen")
plot(pca$x[,1], pca$x[,2], col=mycols, pch=16,
     xlab="PC1", ylab="PC2")
abline(h=0, col="gray")
abline(v=0, col="gray")
```



Another important output from PCA is called the "loadings" vector or the "rotation" component - this tells us how much the original variables (the foods in this case) contribute to the new PCs.

```r
pca$rotation
```

```
                            PC1          PC2         PC3          PC4
Cheese             -0.056955380  0.016012850  0.02394295 -0.409382587
Carcass_meat        0.047927628  0.013915823  0.06367111  0.729481922
Other_meat         -0.258916658 -0.015331138 -0.55384854  0.331001134
Fish               -0.084414983 -0.050754947  0.03906481  0.022375878
Fats_and_oils      -0.005193623 -0.095388656 -0.12522257  0.034512161
Sugars             -0.037620983 -0.043021699 -0.03605745  0.024943337
Fresh_potatoes      0.401402060 -0.715017078 -0.20668248  0.021396007
Fresh_Veg          -0.151849942 -0.144900268  0.21382237  0.001606882
Other_Veg          -0.243593729 -0.225450923 -0.05332841  0.031153231
Processed_potatoes -0.026886233  0.042850761 -0.07364902 -0.017379680
Processed_Veg      -0.036488269 -0.045451802  0.05289191  0.021250980
Fresh_fruit        -0.632640898 -0.177740743  0.40012865  0.227657348
Cereals            -0.047702858 -0.212599678 -0.35884921  0.100043319
Beverages          -0.026187756 -0.030560542 -0.04135860 -0.018382072
Soft_drinks         0.232244140  0.555124311 -0.16942648  0.222319484
Alcoholic_drinks   -0.463968168  0.113536523 -0.49858320 -0.273126013
Confectionery      -0.029650201  0.005949921 -0.05232164  0.001890737
```

PCA seems to be super useful method to gain some insight into high dimensional data that is difficult to examine in other ways.