class07: Machine Learning I

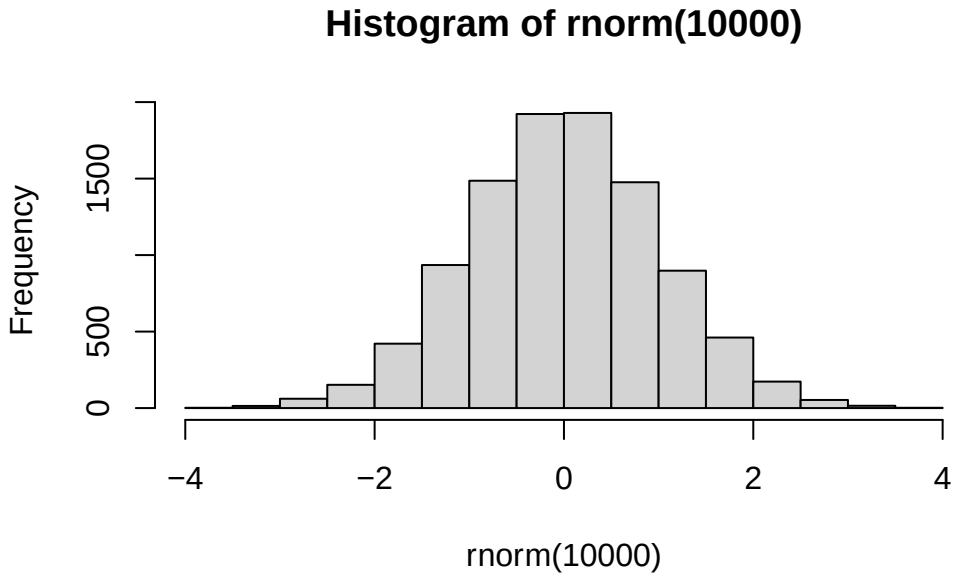
Saba Heydari Seradj (A17002175)

Clustering

```
# make up data with clear groups using nnorm function
rnorm(10)
```

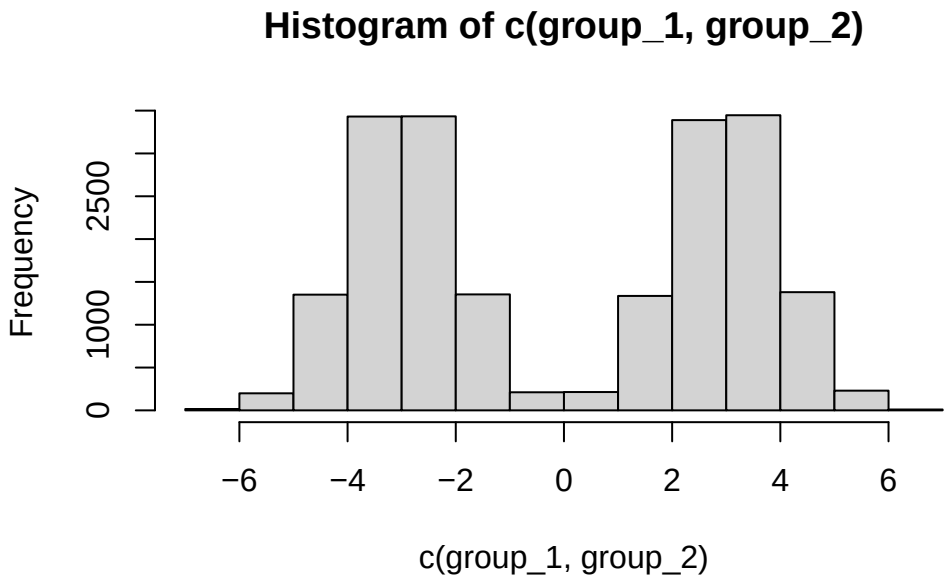
```
 [1]  0.2385475  0.1899933  1.2053969  0.4026503 -0.1563063 -0.3672235
 [7] -1.7222557  0.3805008 -2.3541861  1.6479814
```

```
# plot a histogram of 10k points data
hist(rnorm(10000))
```

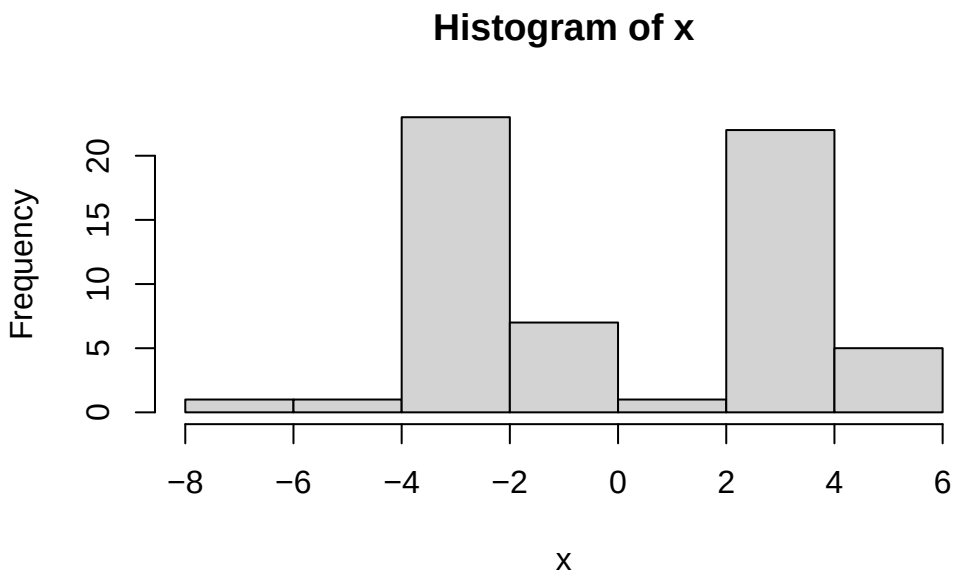


Now, I'll make two groups with different peaks.

```
group_1 <- rnorm(10000, mean=-3)
group_2 <- rnorm(10000, mean=3)
hist(c(group_1, group_2))
```



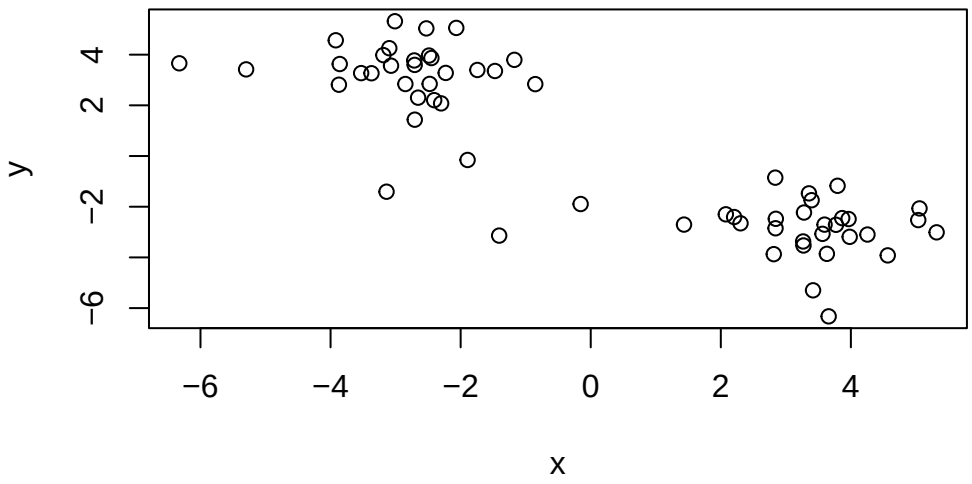

```
n <- 30
x <- c(rnorm(n, -3), rnorm(n, +3))
hist(x)
```




```
# Reverses version of its argument
y <- rev(x)
# Takes the x and y coordinates and
z <- cbind(x, y)
head(z)
```

```
              x         y
[1,] -3.1392927 -1.407388
[2,] -2.7063229  3.595184
[3,] -1.1740769  3.795043
[4,] -3.8716527  2.814295
[5,] -0.8525996  2.837387
[6,] -3.3708827  3.265837
```

```
plot(z)
```



K-means Clustering

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

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

Cluster means:
          x         y
1 -2.846333  3.194692
2  3.194692 -2.846333

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] 91.91902 91.91902
 (between_SS / total_SS =  85.6 %)
```

```
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"
```

What is the cluster size?

```
km$size
```

```
[1] 30 30
```

Cluster assignment/membership?

```
km$cluster
```

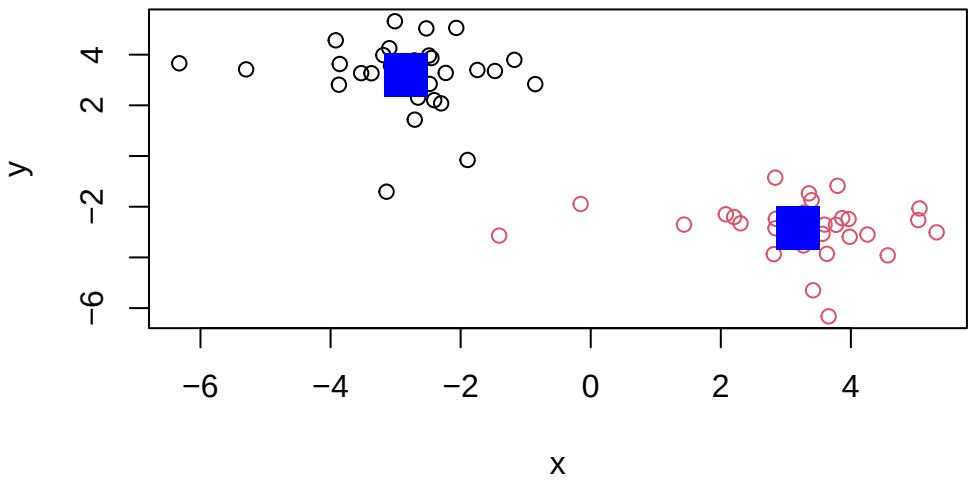
```
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
[39] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

Cluster center?

```
km$centers
```

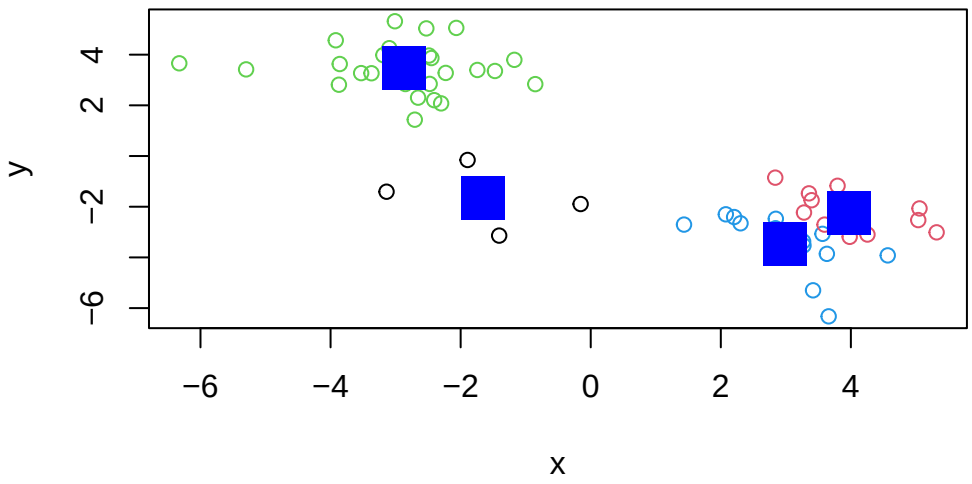
```
          x         y
1 -2.846333  3.194692
2  3.194692 -2.846333
```

```
plot(z, col=km$cluster)
points(km$centers, col='blue', pch=15, cex=3)
```



Now let's try it with 4 clusters instead of 2.

```
km4 <- kmeans(z, centers=4)
plot(z, col=km4$cluster)
points(km4$centers, col='blue', pch=15, cex=3)
```

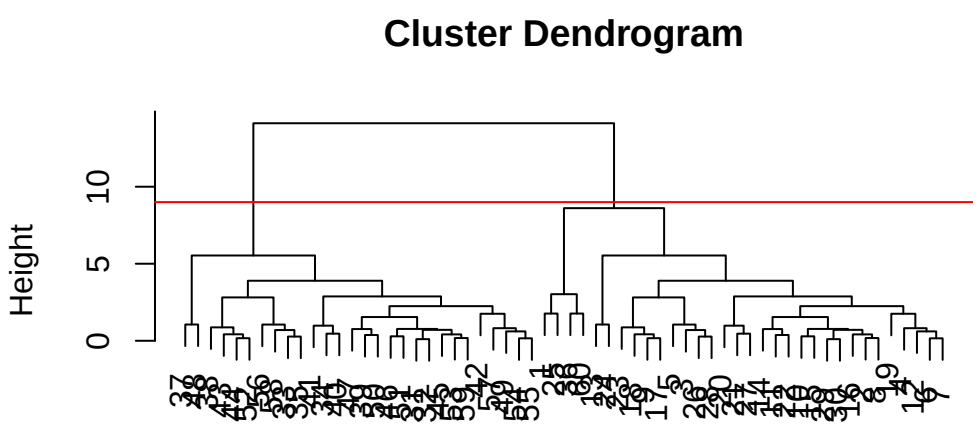


Hierarchical Clustering

Using hclust() function to run hierarchical clustering.

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

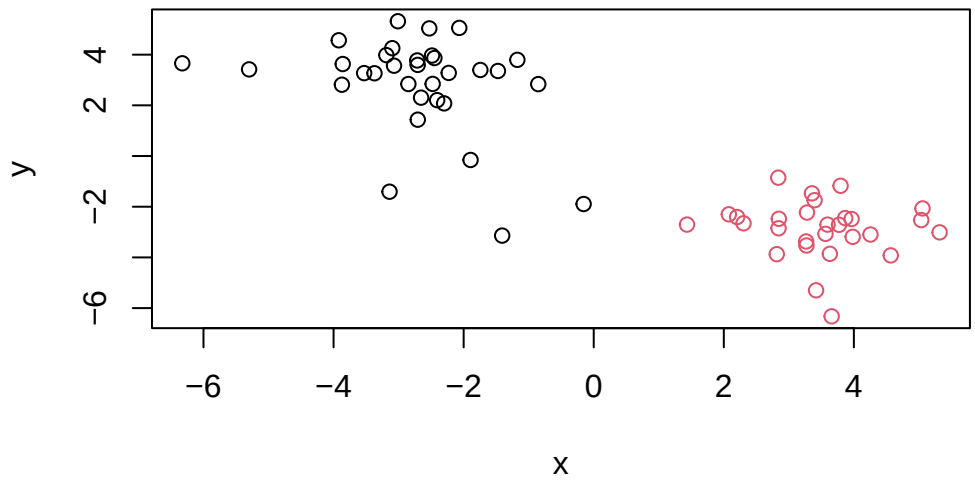
```
plot(hc)
abline(h=9, col='red')
```



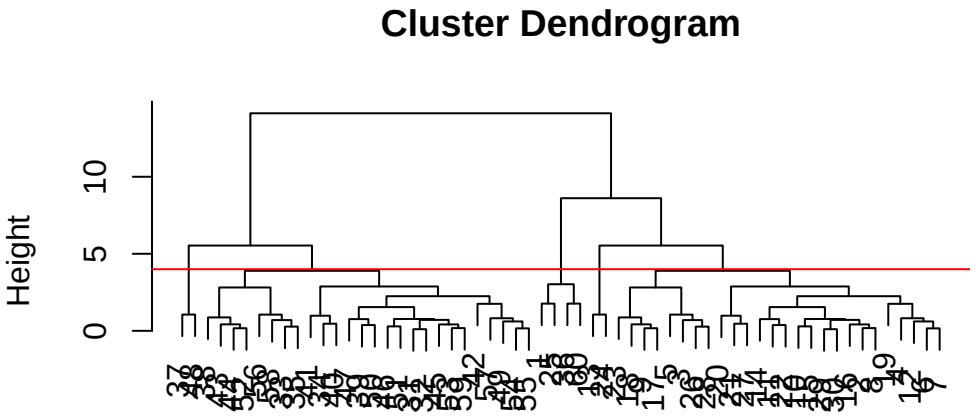

```
grps <- cutree(hc, h=9)
grps
```

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

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



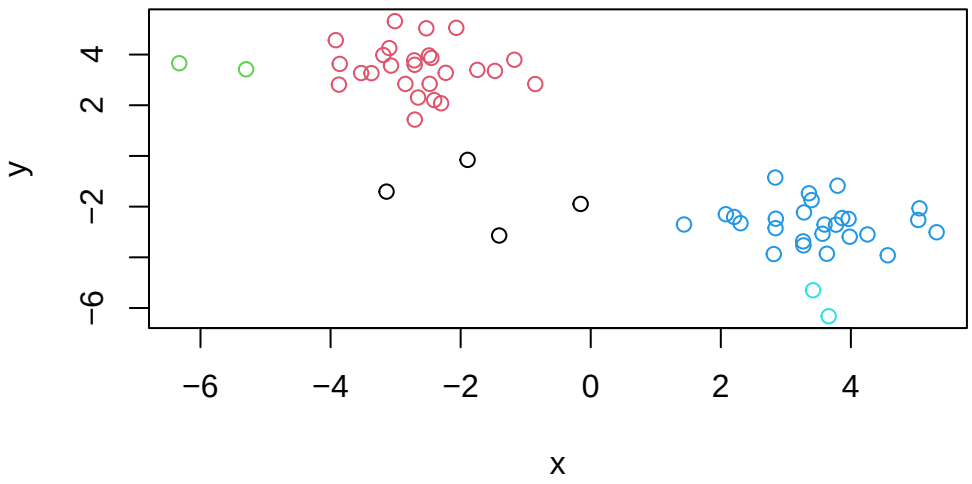

```
plot(hc)
abline(h=4, col='red')
```



```
grps4 <- cutree(hc, h=4)
grps4
```

```
 [1] 1 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 3 1 2 2 2 2 2 4 4 4 4 4 1 5 4
[39] 4 4 4 4 4 4 4 4 4 5 4 4 4 4 4 4 4 4 4 4 4 4 1
```

```
plot(z, col=grps4)
```



Importing and Checking UK Food Data

```
# saving input data file into project directory
fna.data <- 'UK_foods.csv'

# store as x.
# I like to set my first column to be the rownames while reading in the dataset
x <- read.csv(fna.data, row.names=1)
```

```
head(x)
```

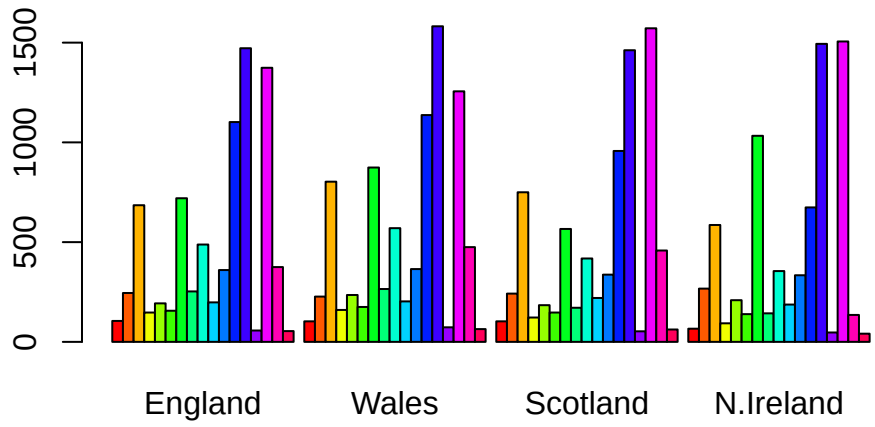
```
               England Wales Scotland N.Ireland
Cheese             105   103      103        66
Carcass_meat       245   227      242       267
Other_meat         685   803      750       586
Fish               147   160      122        93
Fats_and_oils      193   235      184       209
Sugars  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

```
dim(x)
```

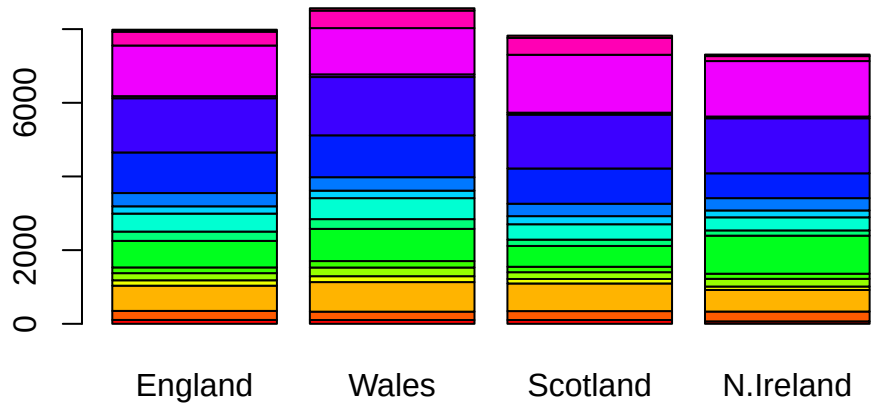
```
[1] 17  4
```

The dataset has 17 rows and 4 column. The columns are England, Wales, Scotland and Ireland. The rows are food items.

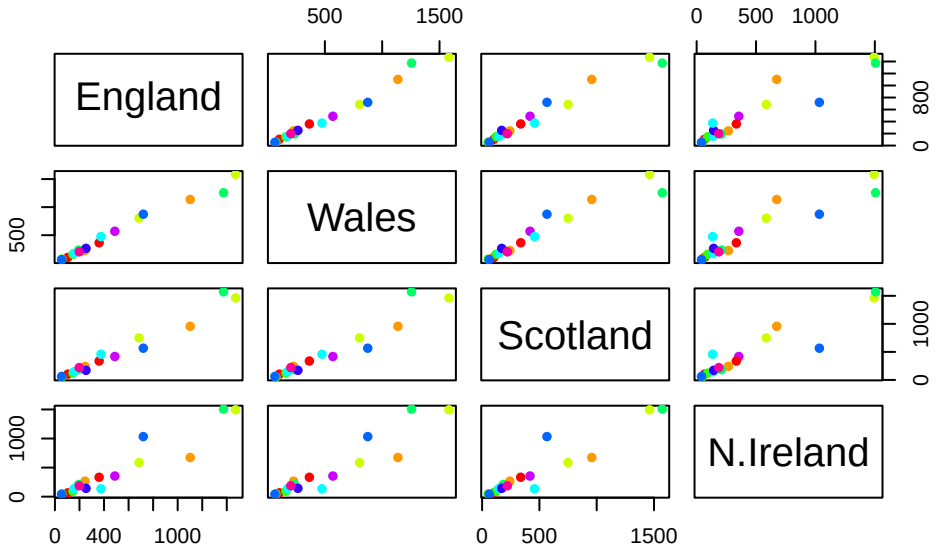
```
barplot(as.matrix(x), beside=T, col=rainbow(nrow(x)))
```



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




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



This code provides a pairwise scatterplot matrix. Each dot represents one of the rows (foods) that are being compared pairwise for the different countries. Even with this small dataset, this is difficult to interpret.

PCA to the rescue

In R, PCA is performed mainly using prcomp() function.

```
# transposing the values and performing a pca on it
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
```

We can see the results using `summary()`. We can see that PC1 captuers 67.44% of the variance in the data.

What is inside this pca object?

```
attributes(pca)
```

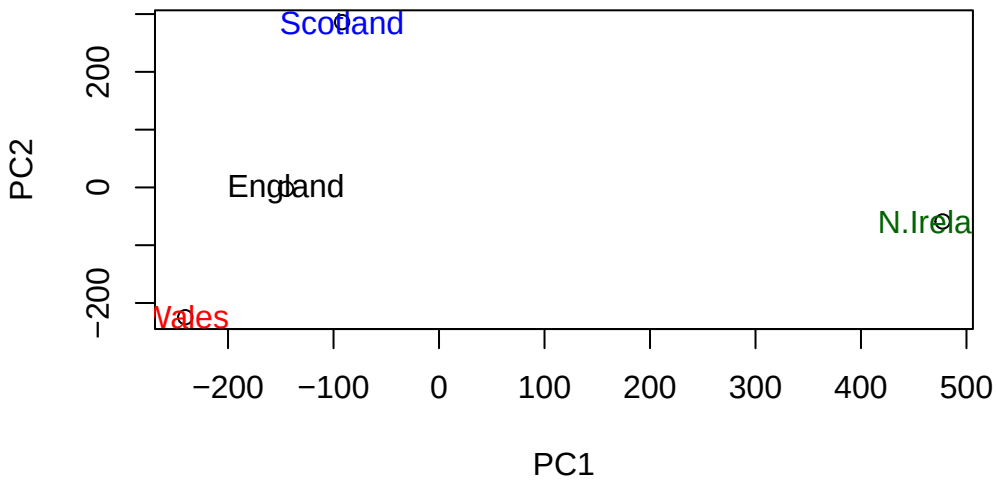
```
$names
[1] "sdev"     "rotation" "center"   "scale"    "x"

$class
[1] "prcomp"
```

```
pca$x
```

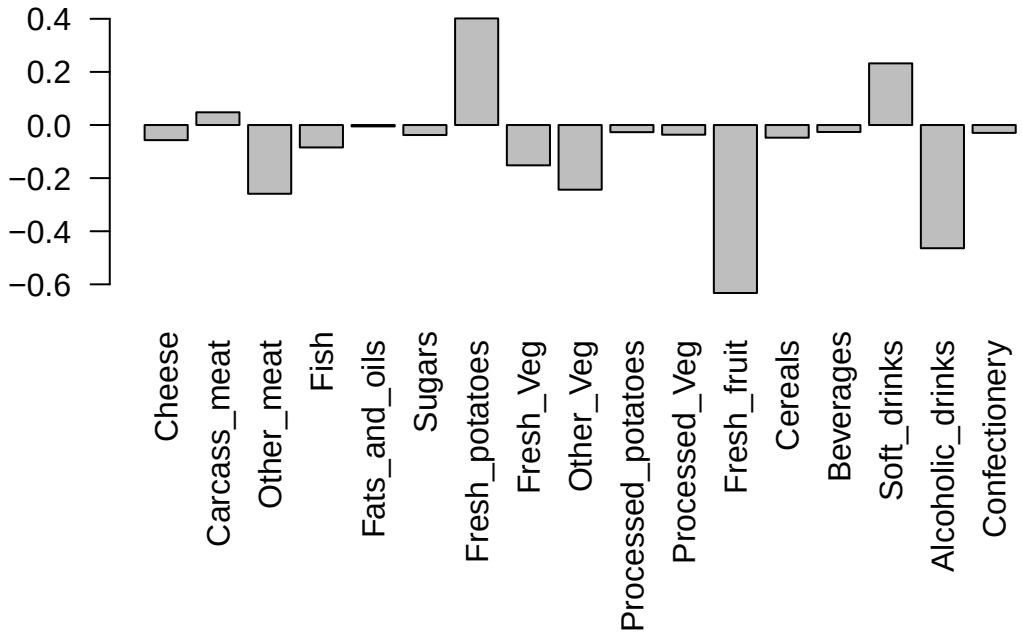
```
                PC1         PC2        PC3           PC4
England   -144.99315   -2.532999 105.768945 -9.152022e-15
Wales     -240.52915 -224.646925 -56.475555  5.560040e-13
Scotland   -91.86934  286.081786 -44.415495 -6.638419e-13
N.Ireland  477.39164  -58.901862  -4.877895  1.329771e-13
```

Let's make a plot of pc2 vs. pc1.

```
plot(pca$x[,1],pca$x[,2],xlab = "PC1", ylab = "PC2")
text(pca$x[,1], pca$x[,2], colnames(x), col=c('black', 'red', 'blue', 'darkgreen'))
```




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


0.4
0.2
0.0
−0.2
−0.4
−0.6
Cheese
Carcass_meat
Other_meat
Fish
Fats_and_oils
Sugars
Fresh_potatoes
Fresh_Veg
Other_Veg
Processed_potatoes
Processed_Veg
Fresh_fruit
Cereals
Beverages
Soft_drinks
Alcoholic_drinks
Confectionery