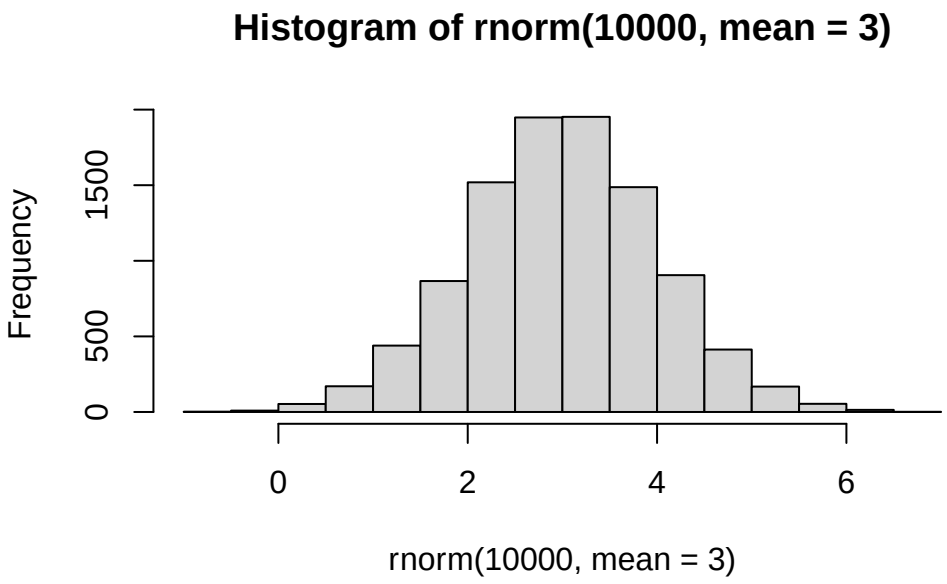
class07

Wanning

```
rnorm(10)
```

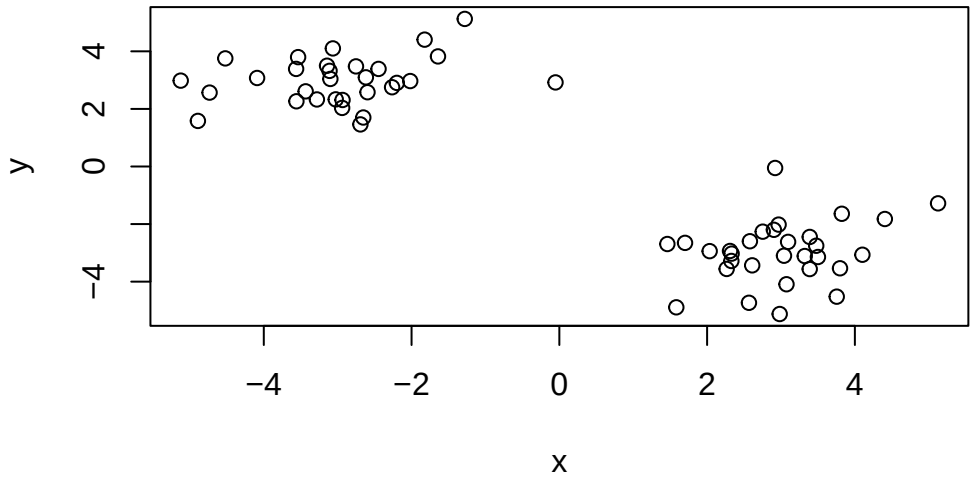
```
 [1] -0.1630050 -1.6360734 -0.7824375 -1.4235010 -0.7134897  0.5172652
 [7] -1.5822077 -1.1554325  0.1361998  0.5683370
```

```
hist(rnorm(10000,mean=3))
```




```
tmp <- c(rnorm(30,3),rnorm(30,-3))
```

```r
x <- cbind (x=tmp,y=rev(tmp))
plot(x)
```



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

```r
k<-kmeans(x, centers=2, nstart=20)
k
```

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

Cluster means:
          x         y
1  2.986566 -2.969690
2 -2.969690  2.986566

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] 53.71756 53.71756
```

```
 (between_SS / total_SS =  90.8 %)

Available components:

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

Q. How many points are in each cluster?

```
k$size
```

```
[1] 30 30
```

Q2. The clustering result i.e. membership vector?

```
k$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
```

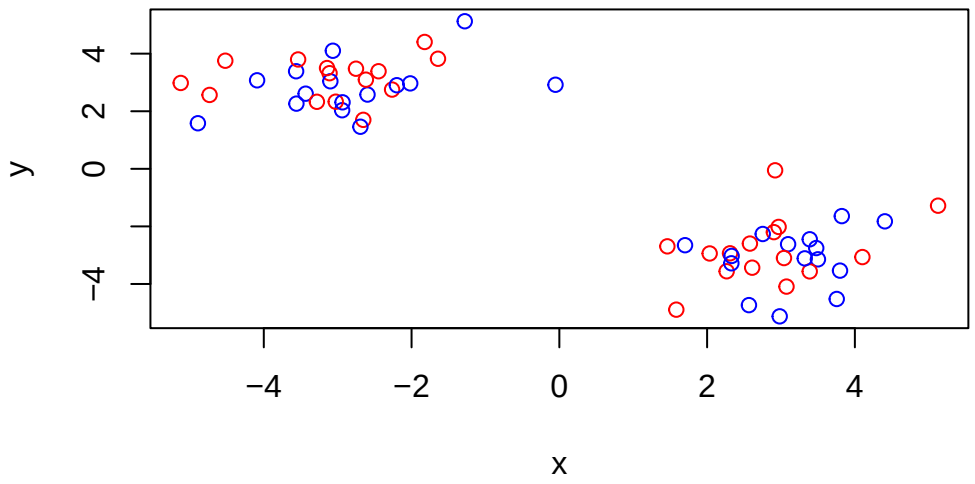
Q3. Cluster centers

```
k$centers
```

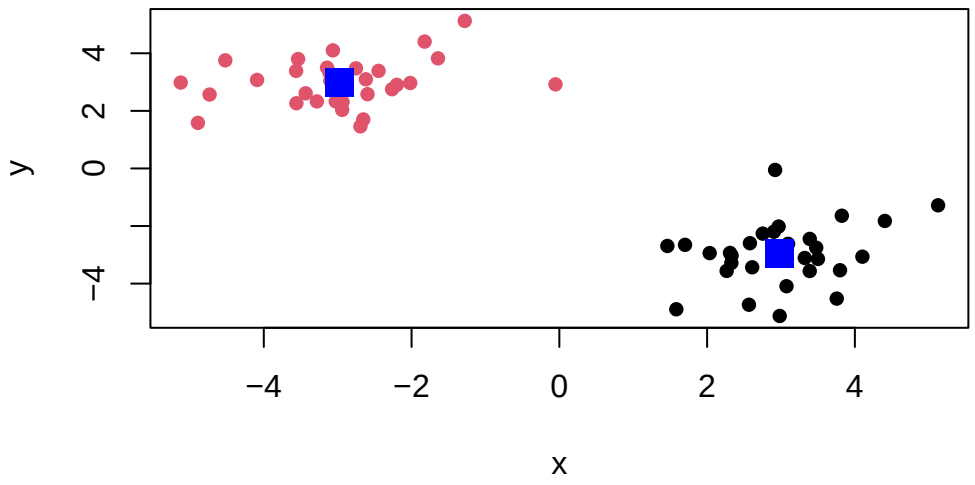
```
          x         y
1  2.986566 -2.969690
2 -2.969690  2.986566
```

Q4.Make a plot of our data colored by clustering results with optionally the cluster centers shown

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

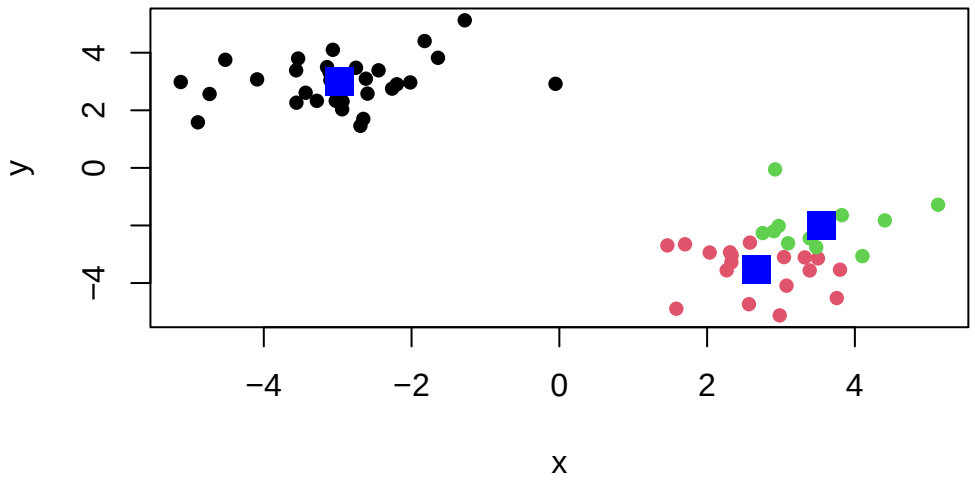



```
plot(x,col=k$cluster,pch=16)
points(k$centers, col="blue", pch=15, cex=2)
```



Q5. Run kmeans again but cluster into 3 groups and plot the results like we did above.

```
k3<-kmeans(x, centers=3, nstart=20)
plot(x,col=k3$cluster,pch=16)
points(k3$centers, col="blue", pch=15, cex=2)
```



K-means will always return a clustering result - even if there is no clear groupings.

#Hierarchical Clustering Hierarchical clustering has an advantage in that it can reveal the structure in your data rather than imposing a structure as k-means will.

The main function in "base" R is called `hclust()`

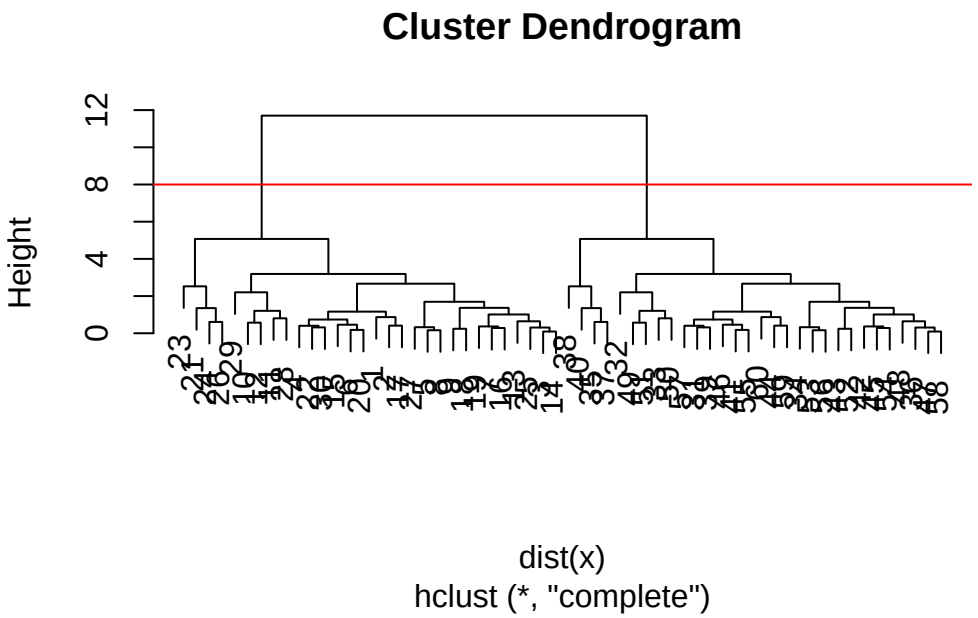
It requires a distance matrix as input, not the raw data itself.

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

```
Call:
hclust(d = dist(x))
```

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

```
plot(hc)
abline(h=8, col="red")
```



The function to get our clusters/groups from a hclust object is called `cutree()`

```
cutree(hc,h=8)
```

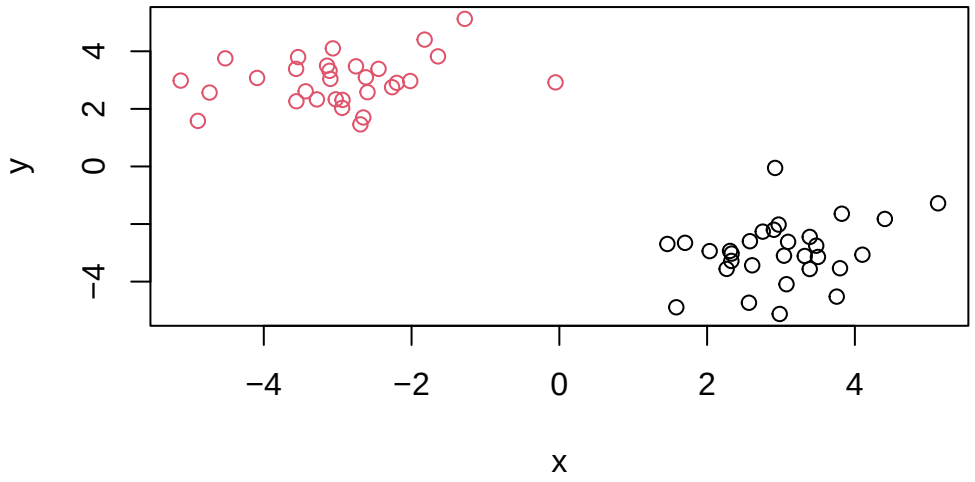
```
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
[39] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

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

Q. Plot our hclust results in terms of our data colored by cluster membership.

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



#Principal Component Analysis (PCA)

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

```
[1] 17  5
```

Q1. 17 rows and 5 columns in my new data frame are named x. We can use dim() function to find out.

```
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
```

```
rownames(x) <- x[,1]
rownames(x)
```

```
 [1] "Cheese"              "Carcass_meat "       "Other_meat "
 [4] "Fish"                "Fats_and_oils "      "Sugars"
 [7] "Fresh_potatoes "     "Fresh_Veg "          "Other_Veg "
[10] "Processed_potatoes " "Processed_Veg "      "Fresh_fruit "
[13] "Cereals "            "Beverages"           "Soft_drinks "
[16] "Alcoholic_drinks "   "Confectionery "
```

```
rownames(x) <- x[,1]
x<-x[,-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
```

```
rownames(x) <- x[,1]
x <- x[,-1]
head(x)
```

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

```
dim(x)
```

```
[1] 17  3
```

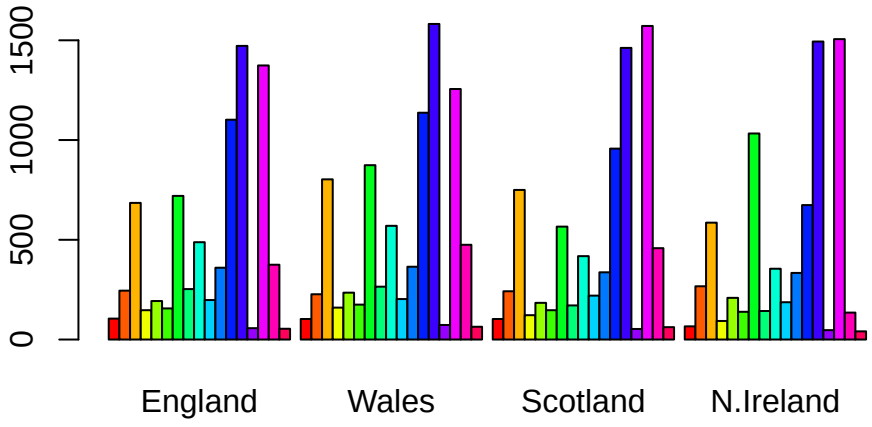
```
x <- read.csv(url, row.names=1)
head(x)
```

```
              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
```

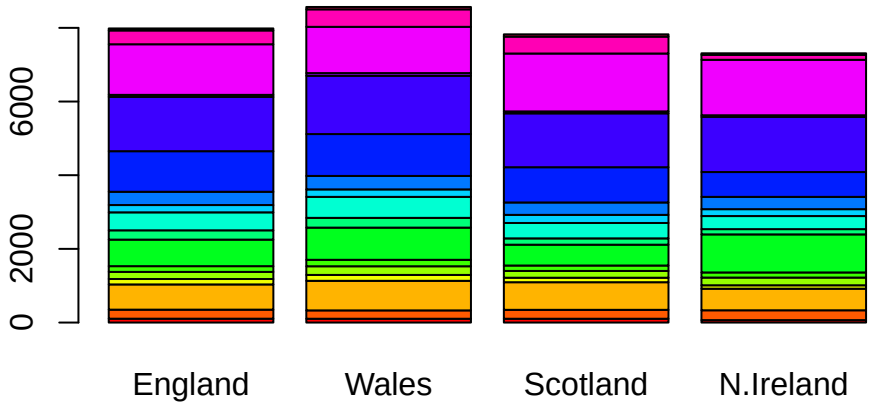
Q2. I prefer the second approach because the data in my columns will not be truncated and the results can be retrieved in a more secure way. The second is more robust overall.

#Spotting major differences and trends

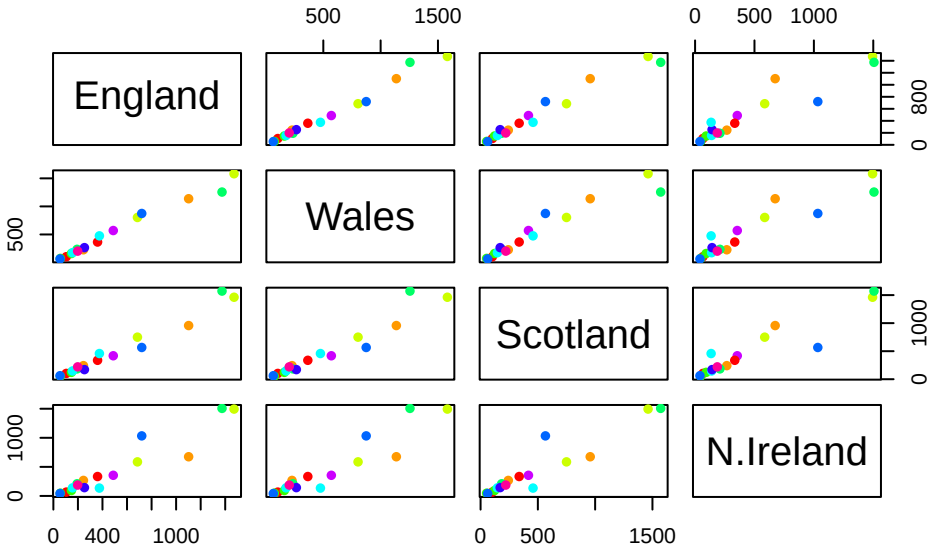
```
barplot(as.matrix(x), beside=T, col=rainbow(nrow(x)))
```



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




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



Q5.If a given point lies on the diagonal, this indicates that the two countries have the same consumptions in that food category.

Q6.N.Ireland has the most variation in the consumptions in different food categories with other countries.

#PCA to the rescue

The main function for PCA in base A is called `pucomp()`

It wants the transpose(with the `t()`) of our food data for analysis

```
t(x)
```

```
           Cheese Carcass_meat  Other_meat  Fish Fats_and_oils  Sugars
England       105           245         685  147            193     156
Wales         103           227         803  160            235     175
Scotland      103           242         750  122            184     147
N.Ireland      66           267         586   93            209     139
           Fresh_potatoes  Fresh_Veg  Other_Veg  Processed_potatoes
England               720        253        488                 198
Wales                 874        265        570                 203
Scotland              566        171        418                 220
N.Ireland            1033        143        355                 187
```

```
          Processed_Veg  Fresh_fruit  Cereals  Beverages Soft_drinks
England             360         1102     1472         57         1374
Wales               365         1137     1582         73         1256
Scotland            337          957     1462         53         1572
N.Ireland           334          674     1494         47         1506
          Alcoholic_drinks  Confectionery
England                375             54
Wales                  475             64
Scotland               458             62
N.Ireland              135             41
```

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

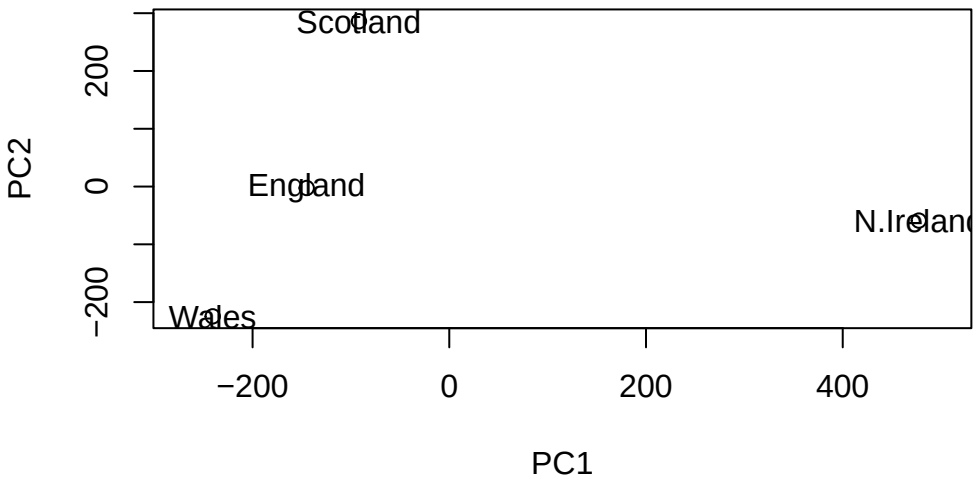
```
Importance of components:
                            PC1      PC2      PC3       PC4
Standard deviation     324.1502 212.7478 73.87622 3.176e-14
Proportion of Variance   0.6744   0.2905  0.03503 0.000e+00
Cumulative Proportion    0.6744   0.9650  1.00000 1.000e+00
```

One of the main results that folks look for is called the "score plot" a.k.a. PC plot, PC1 vs PC2 plot

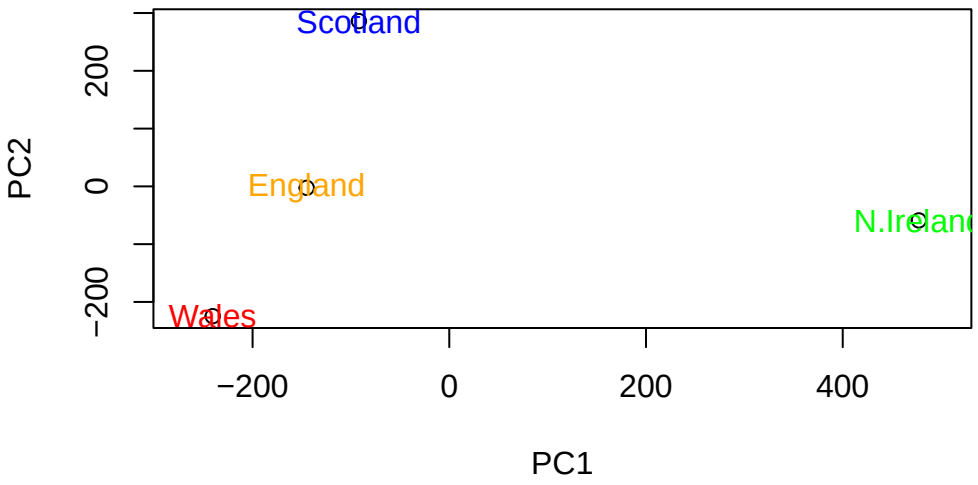
```
pca$x
```

```
                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
```

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




```
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"))
```



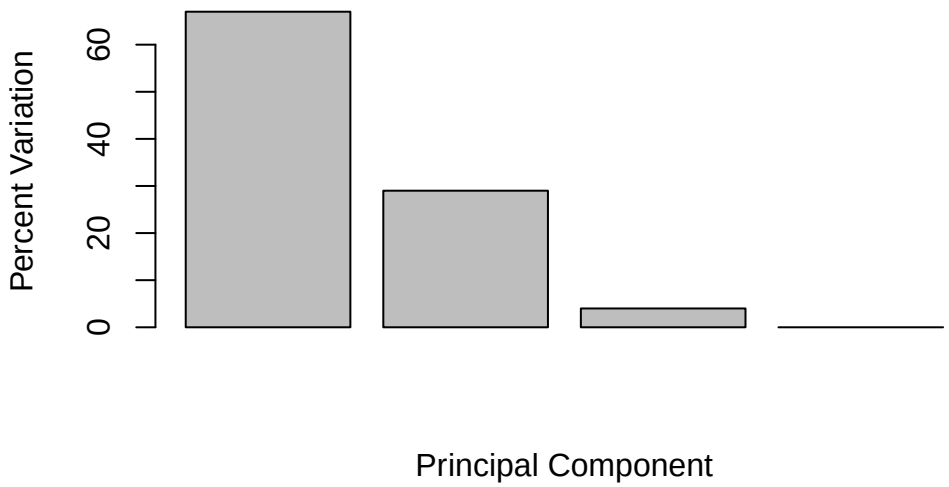
```r
v <- round( pca$sdev^2/sum(pca$sdev^2) * 100 )
v
```

```
[1] 67 29  4  0
```

```r
z <- summary(pca)
z$importance
```

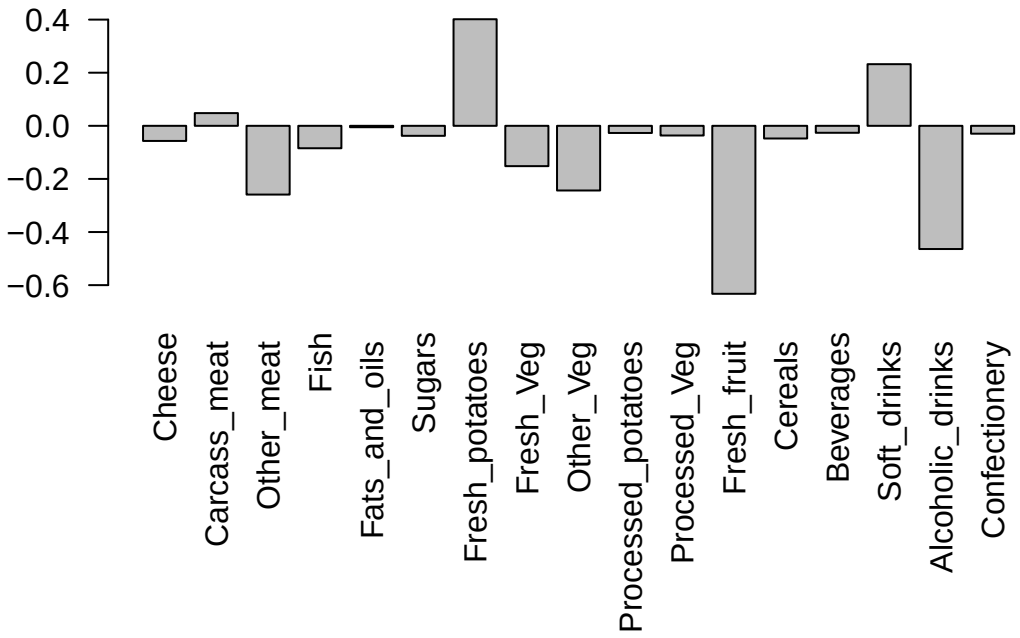
```
                            PC1       PC2      PC3           PC4
Standard deviation     324.15019 212.74780 73.87622 3.175833e-14
Proportion of Variance   0.67444   0.29052  0.03503 0.000000e+00
Cumulative Proportion    0.67444   0.96497  1.00000 1.000000e+00
```

```r
barplot(v, xlab="Principal Component", ylab="Percent Variation")
```

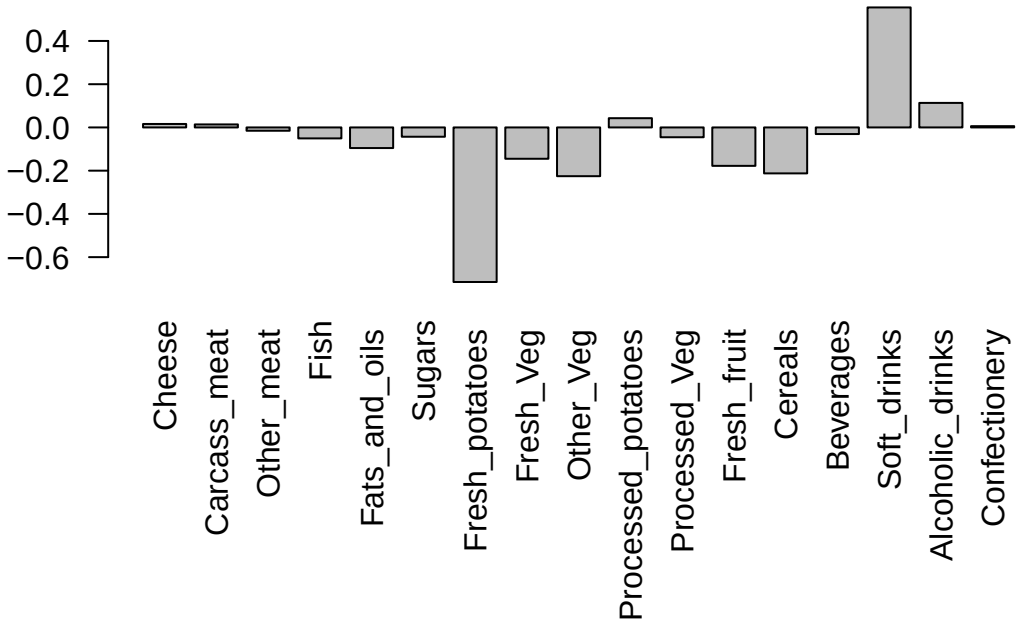


#Digging deeper (variable loading)

```
## Lets focus on PC1 as it accounts for > 90% of variance
par(mar=c(10, 3, 0.35, 0))
barplot( pca$rotation[,1], las=2 )
```




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



Q9. The soft drinks and fresh potatoes food groups feature most prominently in PC2. It tells us that these two food groups have the most variation between Wales and Scotland.