Lab7 PID: A59000602

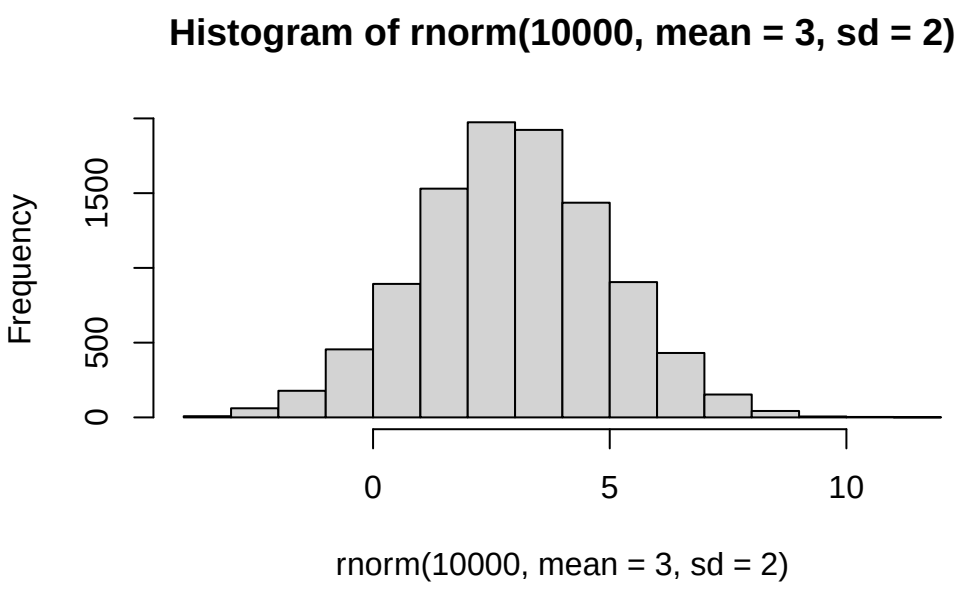
Blair Chang

#Clustering

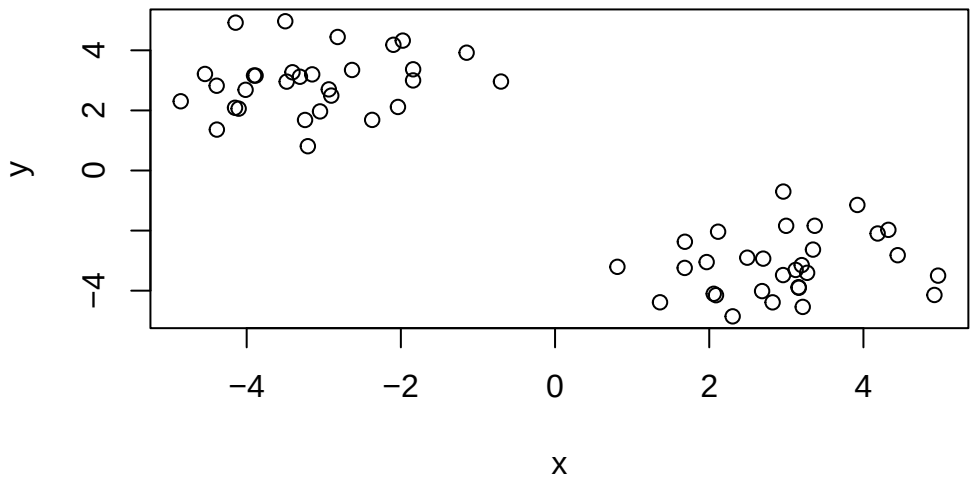
We will start with k-means clustering, one of the most prevelent of all clustering methods.

To get started let's make some data up:

```
hist(rnorm(10000, mean = 3, sd = 2)) #from normal distribution
```




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



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

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

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

Cluster means:
          x         y
1 -3.133934  2.942855
2  2.942855 -3.133934

Clustering vector:
 [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1
[39] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Within cluster sum of squares by cluster:
[1] 60.77718 60.77718
 (between_SS / total_SS =  90.1 %)

Available components:
```

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

#components can be containers (list) for your data

Q1. how many points are there in each cluster?

```
k$size
```

```
[1] 30 30
```

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

```
k$cluster
```

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

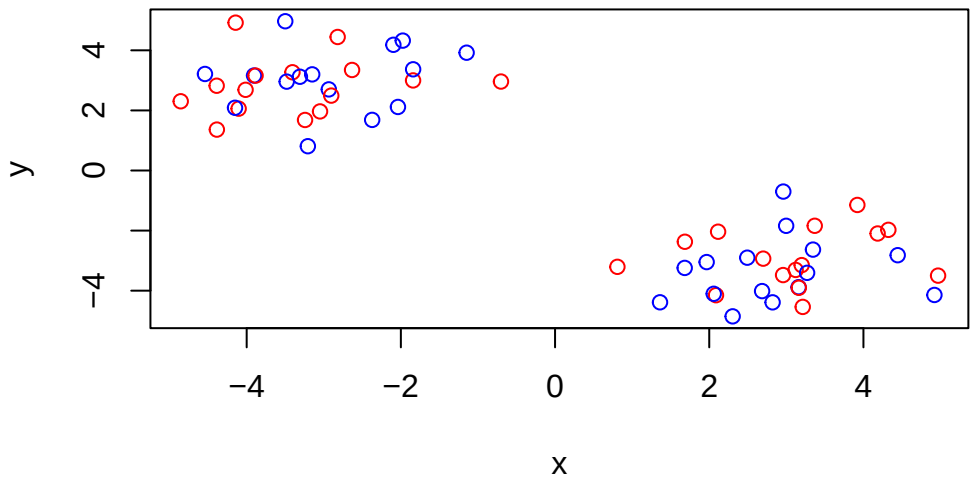
Q3. Cluster centers

```
k$centers
```

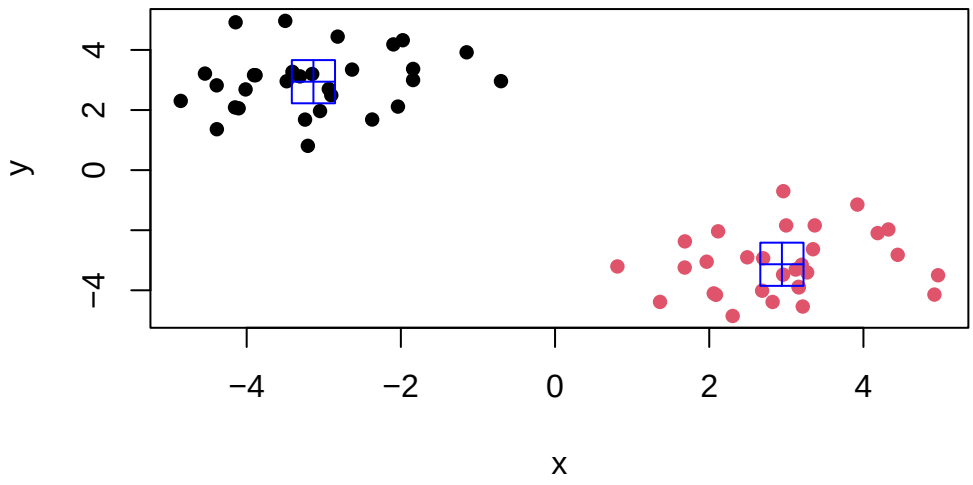
```
          x         y
1 -3.133934  2.942855
2  2.942855 -3.133934
```

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

```
plot(x, col=c("red", "blue")) #recycle red/blue till it reaches 60
```

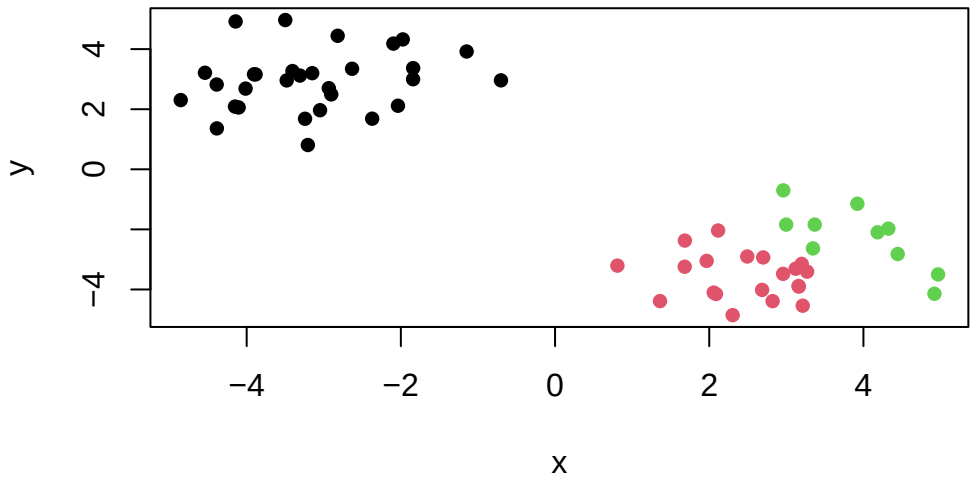



```
plot(x, col= k$cluster, pch=16)
points(k$centers, col="blue", pch=12, cex=3)
```



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)
```



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

Hierarchical Clustering

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

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

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

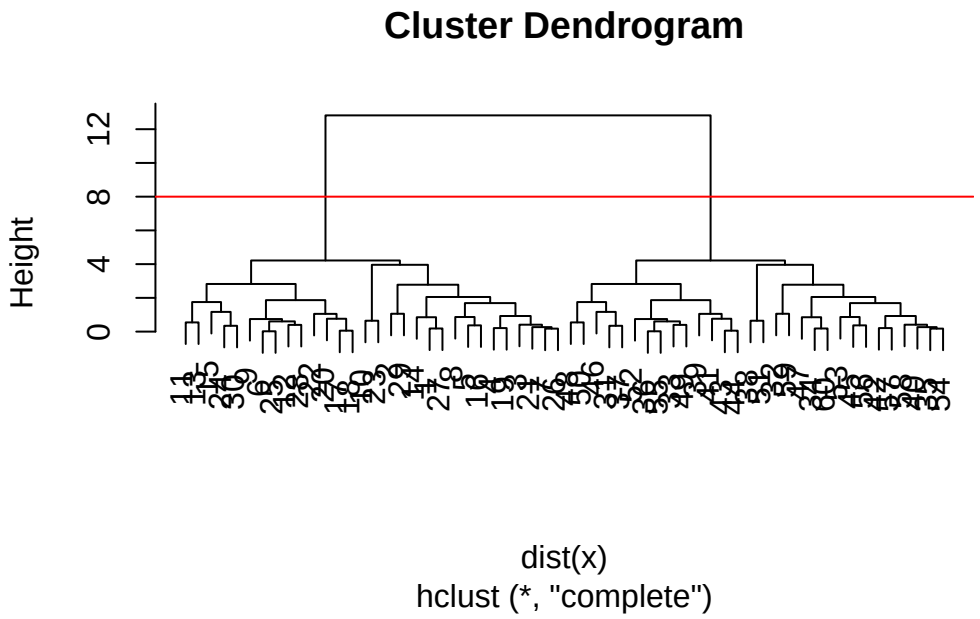
```
hc <- hclust(dist(x))
hc
```

```
Call:
```

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

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

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



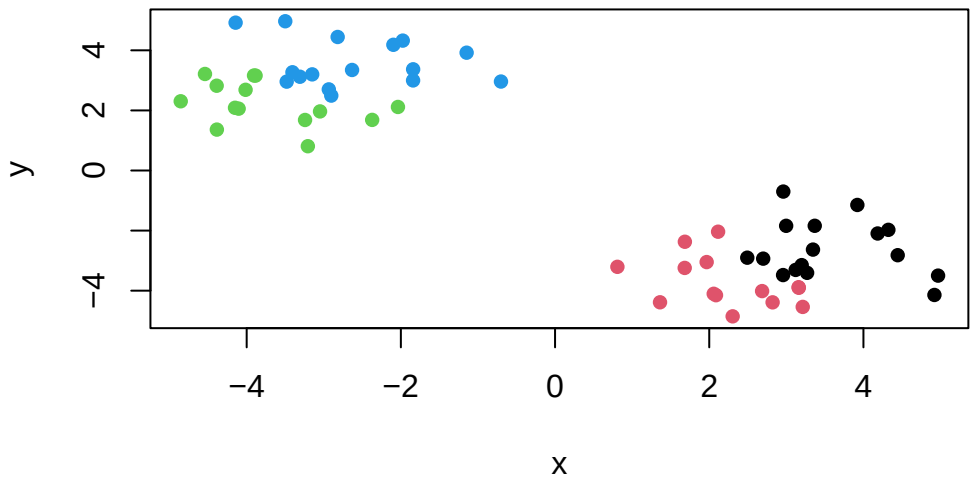
The function to get our clustering/ groups from a hcluct object is called `cutree()`

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

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

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

```
plot(x, col=grps, pch=16)
```



Principal Component Analysis (PCA)

WE will work on data from the UK about the strange stuff folks there eat.

```
url <- "https://tinyurl.com/UK-foods"
x <- read.csv(url)
#Q1. How many rows and columns are in your new data frame named x? What R functions could

dim(x)
```

```
[1] 17  5
```

```
#Q2 Which approach to solving the 'row-names problem' mentioned above do you prefer and wh

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

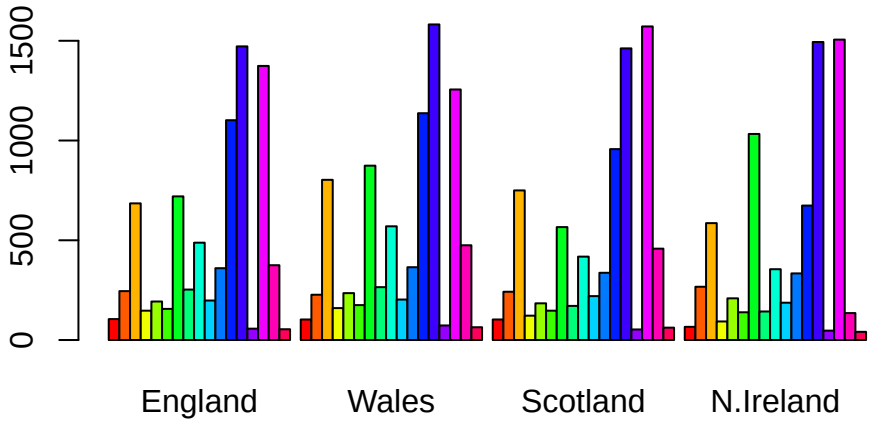
```
              England Wales Scotland N.Ireland
Cheese            105   103      103        66
Carcass_meat      245   227      242       267
Other_meat        685   803      750       586
Fish              147   160      122        93
Fats_and_oils     193   235      184       209
Sugars        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

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

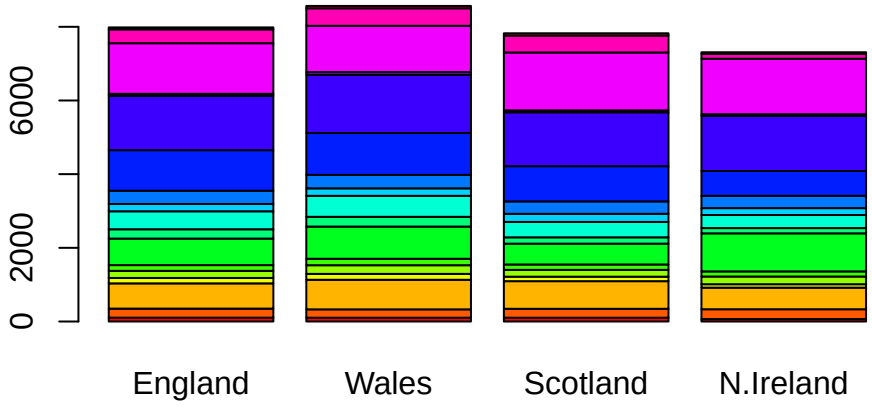
```
              England Wales Scotland N.Ireland
Cheese            105   103      103        66
Carcass_meat      245   227      242       267
Other_meat        685   803      750       586
Fish              147   160      122        93
Fats_and_oils     193   235      184       209
Sugars        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

Q3: Changing what optional argument in the above barplot() function results in the following plot?

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



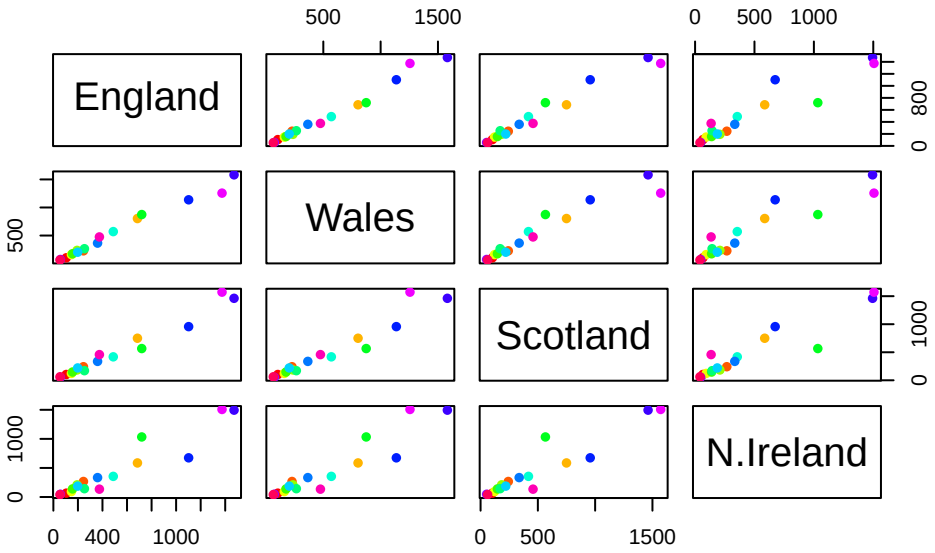

```
barplot(as.matrix(x), beside=F, col=rainbow(nrow(x)))
```



Q5: Generating all pairwise plots may help somewhat. Can you make sense of the following code and resulting figure? What does it mean if a given point lies on the diagonal for a given plot?

Yes. It plots and compares each country. The distance of that point bw two countries is the same, meaning that it is the most similar factor bw those countries.

```
pairs(x, col=rainbow(17), pch=16)
```



Q6. What is the main differences between N. Ireland and the other countries of the UK in terms of this data-set?

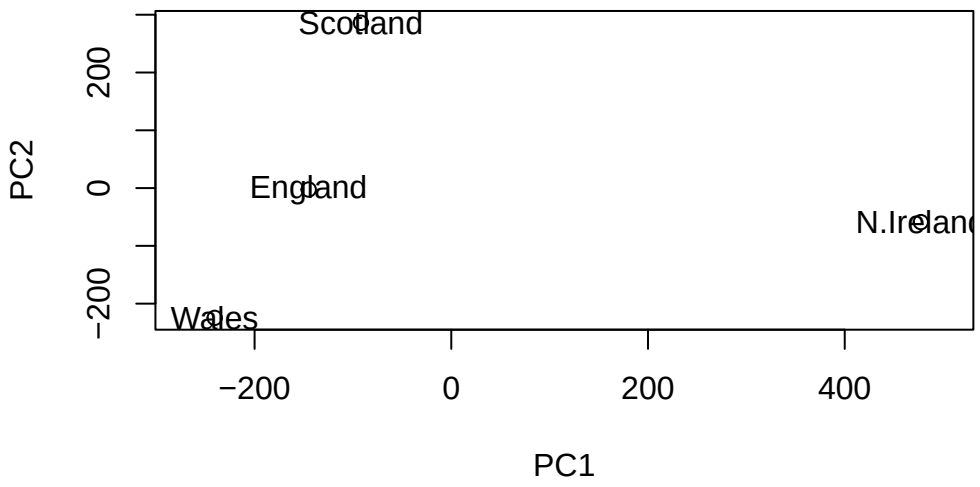
N. Ireland looks very different from other 3 countries in terms of food preference, the blue and green and pink points are far away from the diagonal.

```
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
```

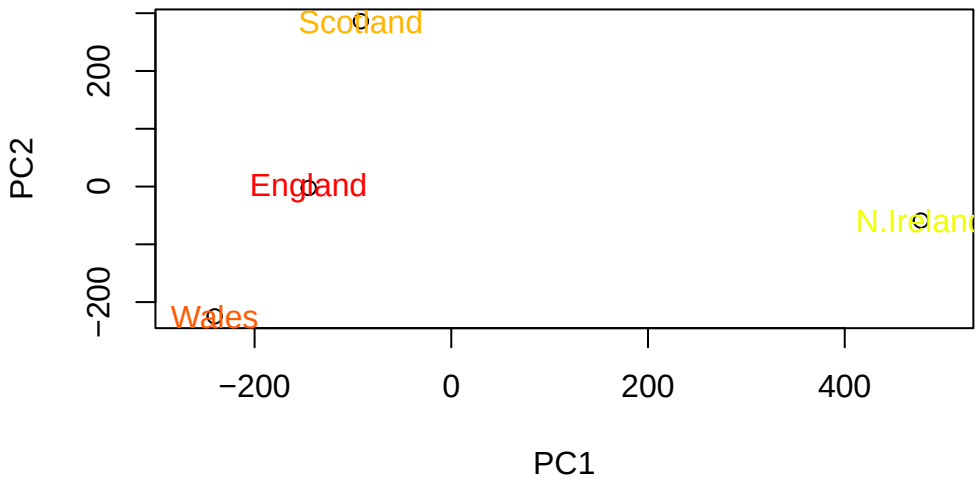
Q7. Complete the code below to generate a plot of PC1 vs PC2. The second line adds text labels over the data points.

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



Q8. Customize your plot so that the colors of the country names match the colors in our UK and Ireland map and table at start of this document.

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



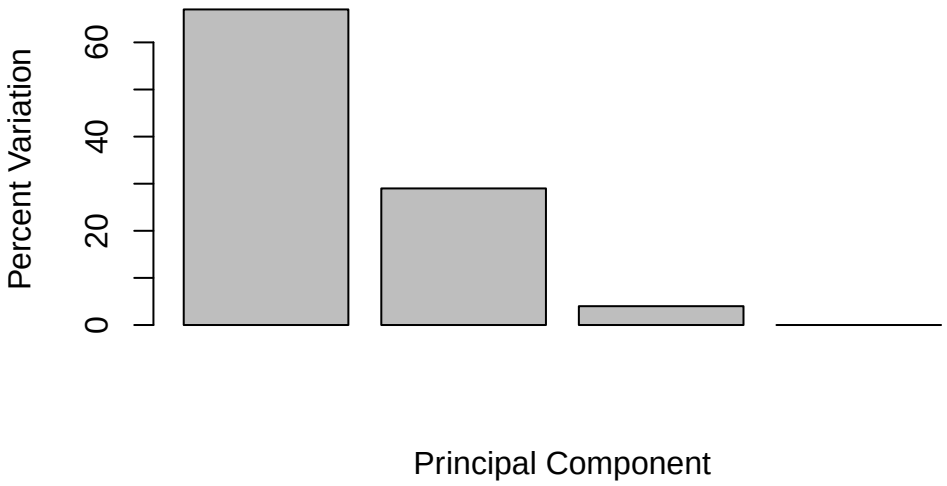

```
v <- round( pca$sdev^2/sum(pca$sdev^2) * 100 )
v
```

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

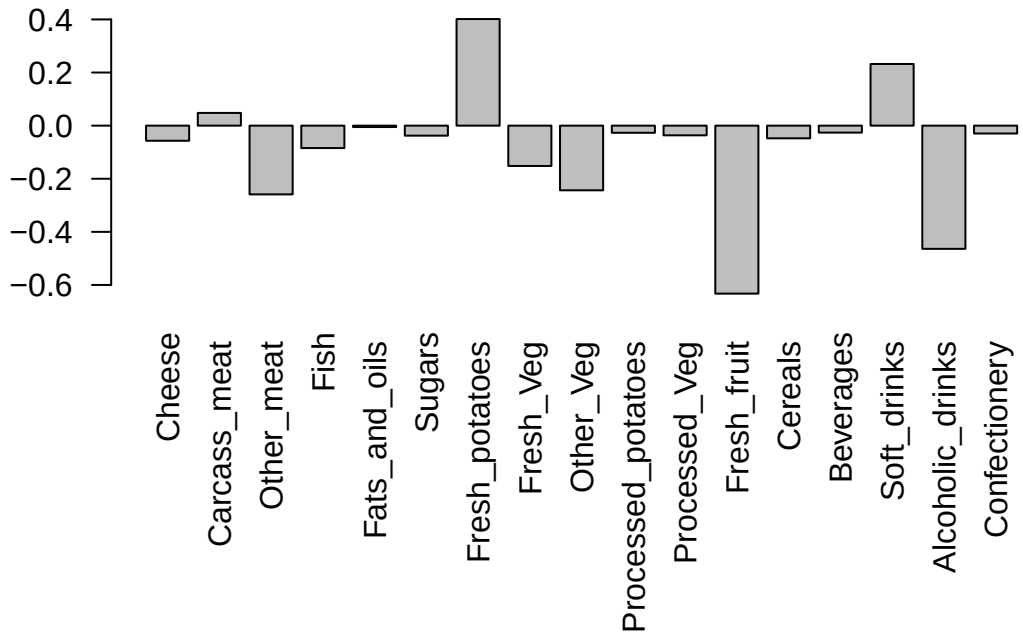
```
z <- summary(pca)
z$importance
```

```
                             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
```

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

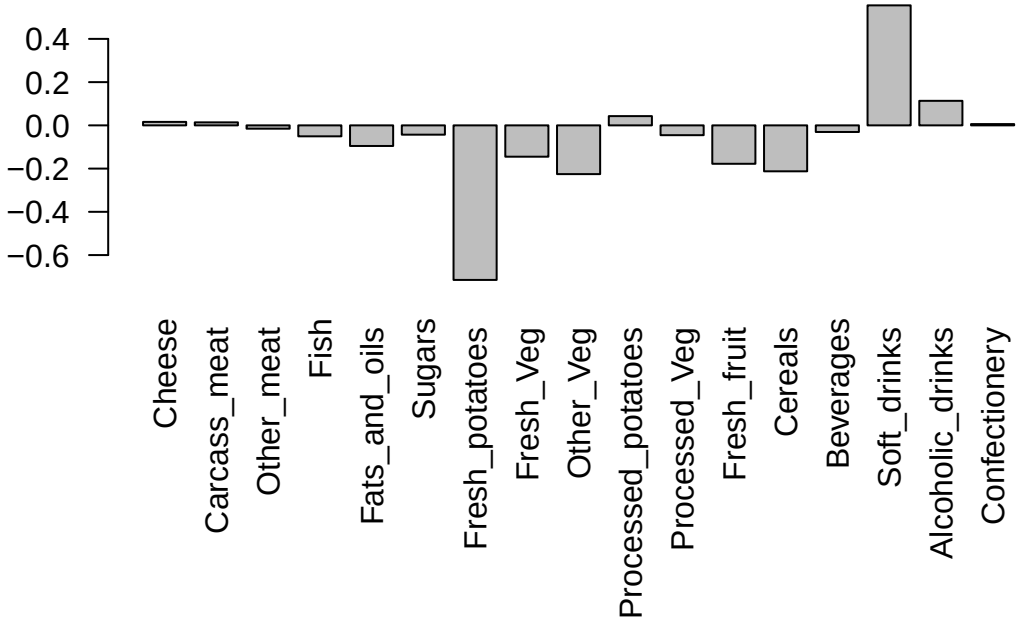



```
par(mar=c(10, 3, 0.35, 0))
barplot( pca$rotation[,1], las=2 )
```



Q9: Generate a similar 'loadings plot' for PC2. What two food groups feature prominantely and what does PC2 maninly tell us about?

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



Here we see observations (foods) with the largest positive loading scores that effectively "push" N. Ireland to right positive side of the plot (Soft_drinks and Acoholic_drinks).

We can also see the observations/foods with high negative scores that push the other countries to the left side of the plot (Fresh_potatoes).

PCA to the rescue

Help me make sense of this data The main function for PCA in base R is called `prcomp()`

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

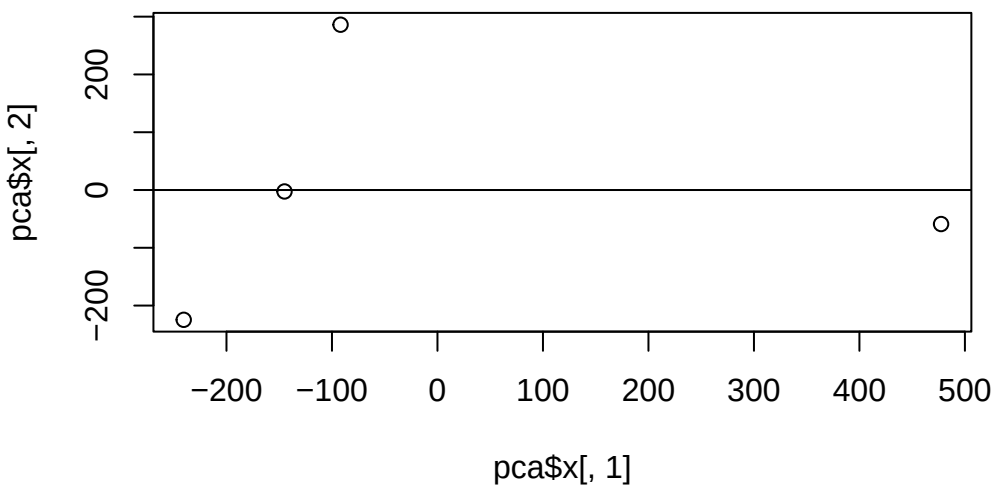
```
pca <- prcomp(t(x))

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" aka PC plot, PC1 vs PC2 plot

```
plot(pca$x[,1], pca$x[,2])
abline(h=0)
```



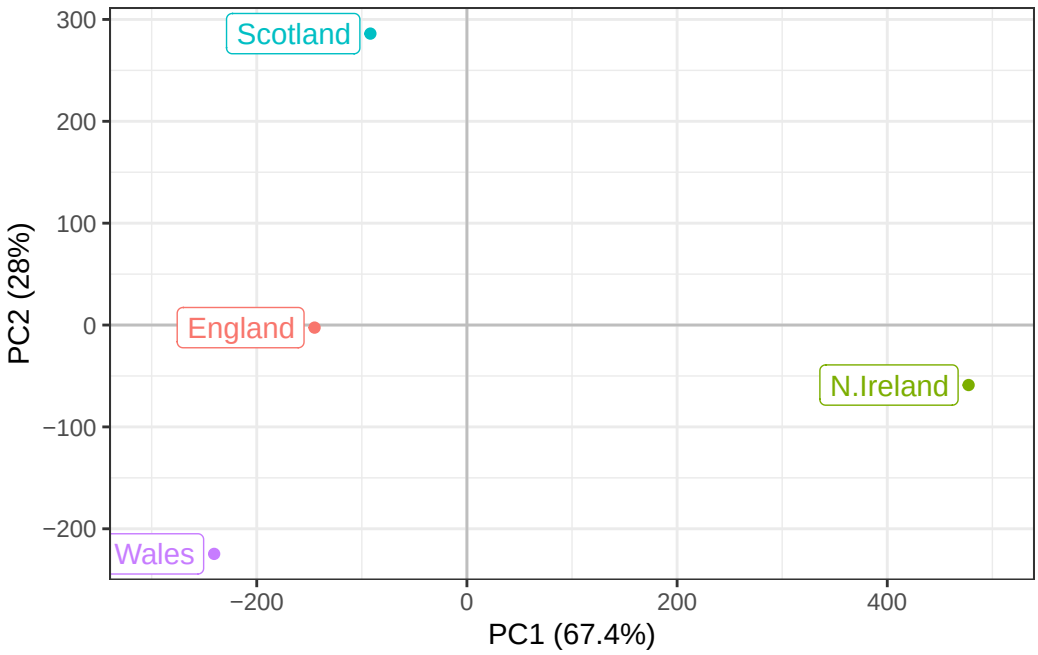

```
library(ggplot2)

df <- as.data.frame(pca$x)
df_lab <- tibble::rownames_to_column(df, "Country")

# Our first basic plot


ggplot(df_lab) +
```

```
  aes(PC1, PC2, col=Country, label=Country) +
  geom_hline(yintercept = 0, col="gray") +
  geom_vline(xintercept = 0, col="gray") +
  geom_point(show.legend = FALSE) +
  geom_label(hjust=1, nudge_x = -10, show.legend = FALSE) +
  expand_limits(x = c(-300,500)) +
  xlab("PC1 (67.4%)") +
  ylab("PC2 (28%)") +
  theme_bw()
```



##PCA of RNA-seq data

```
  url2 <- "https://tinyurl.com/expression-CSV"
  rna.data <- read.csv(url2, row.names=1)
  head(rna.data)
```

```
       wt1 wt2  wt3  wt4 wt5 ko1 ko2 ko3 ko4 ko5
gene1  439 458  408  429 420  90  88  86  90  93
gene2  219 200  204  210 187 427 423 434 433 426
gene3 1006 989 1030 1017 973 252 237 238 226 210
gene4  783 792  829  856 760 849 856 835 885 894
```

```
gene5  181 249  204  244 225 277 305 272 270 279
gene6  460 502  491  491 493 612 594 577 618 638
```

```
dim(rna.data)
```

```
[1] 100  10
```

Q10: How many genes and samples are in this data set?

100 genes and 10 samples