

Machine Learning 1

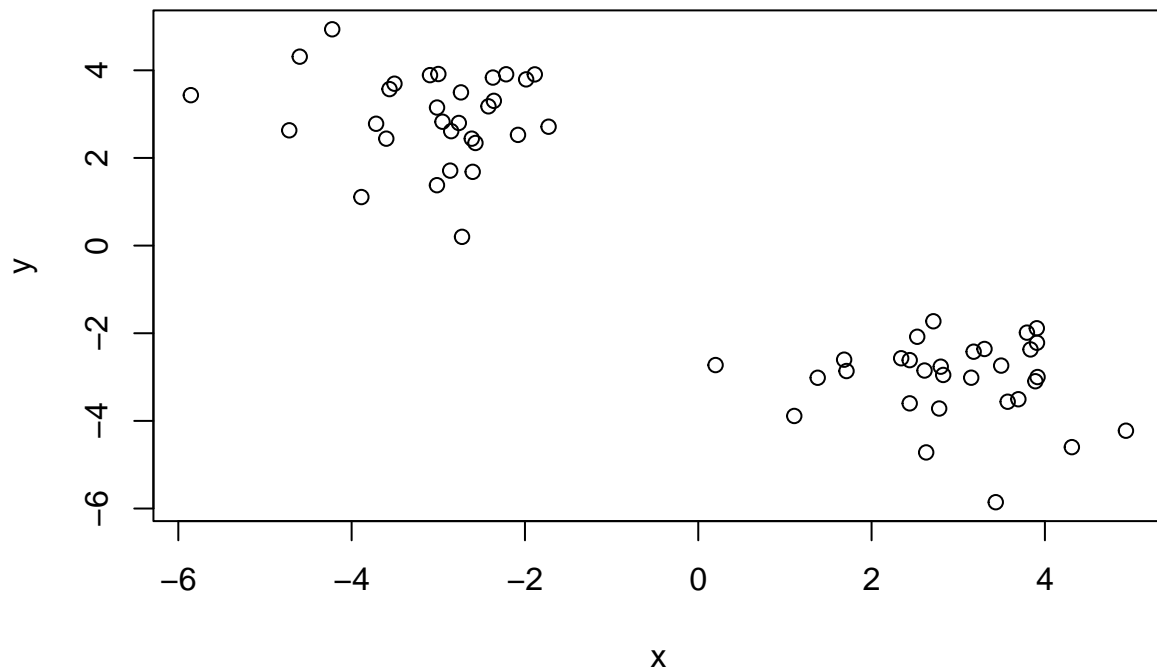
Haroon Riyaz (PID A15377799)

2/14/2022

First: Kmeans()

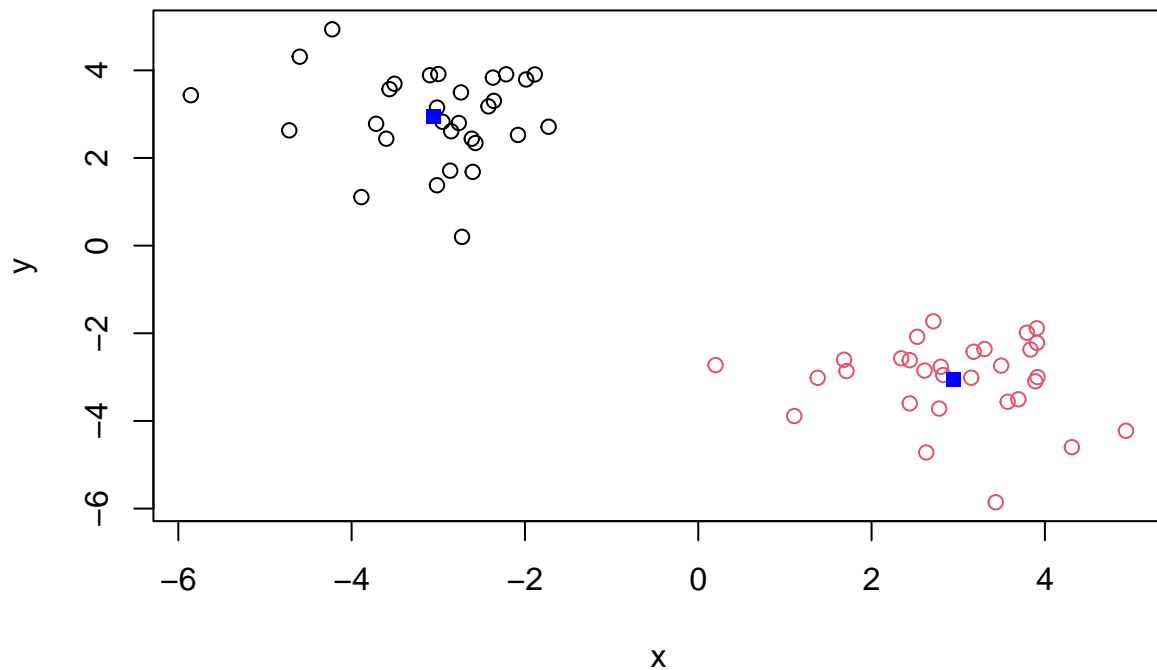
Demo for using kmeans function in Base R! First make data with known structure!

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



Now we have made up data in 'x' let's see how kmeans works with this!

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

hclust() or Hierarchial Clustering

We will cluster the same data 'x' with hclust! In this case, hclust requires a distance matrix as input.

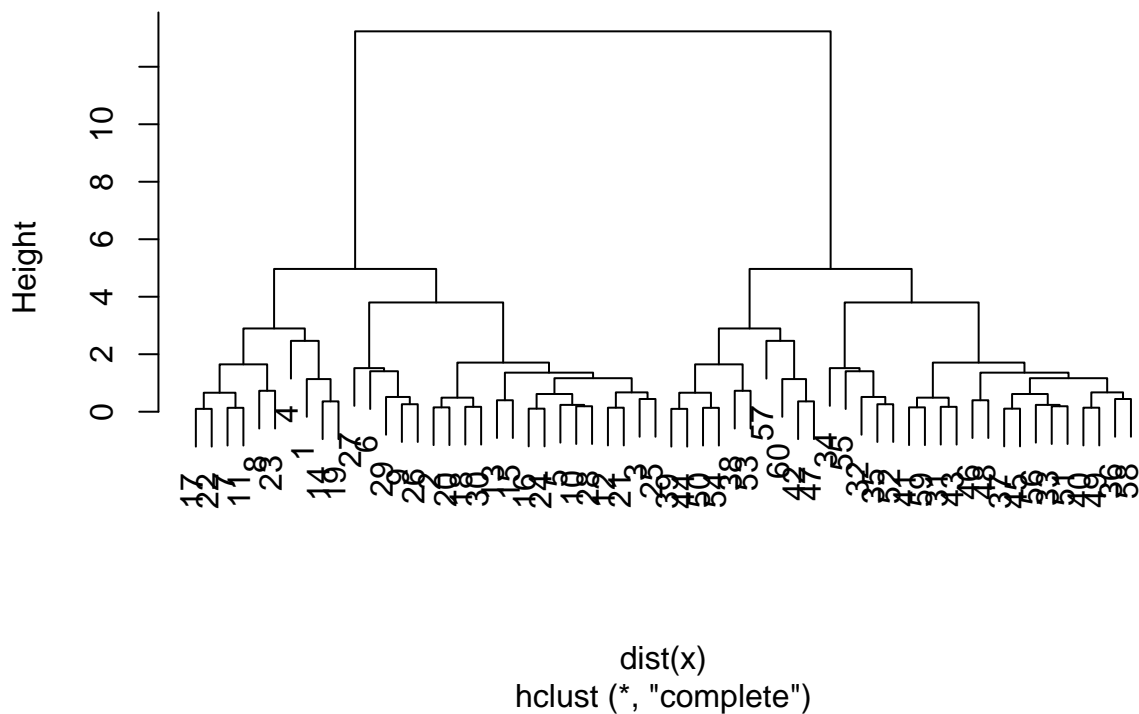
```
hc <- hclust(dist(x))
hc
```

```
##
## Call:
## hclust(d = dist(x))
##
## Cluster method   : complete
## Distance         : euclidean
## Number of objects: 60
```

Let's plot hclust result!

```
plot(hc)
```

Cluster Dendrogram



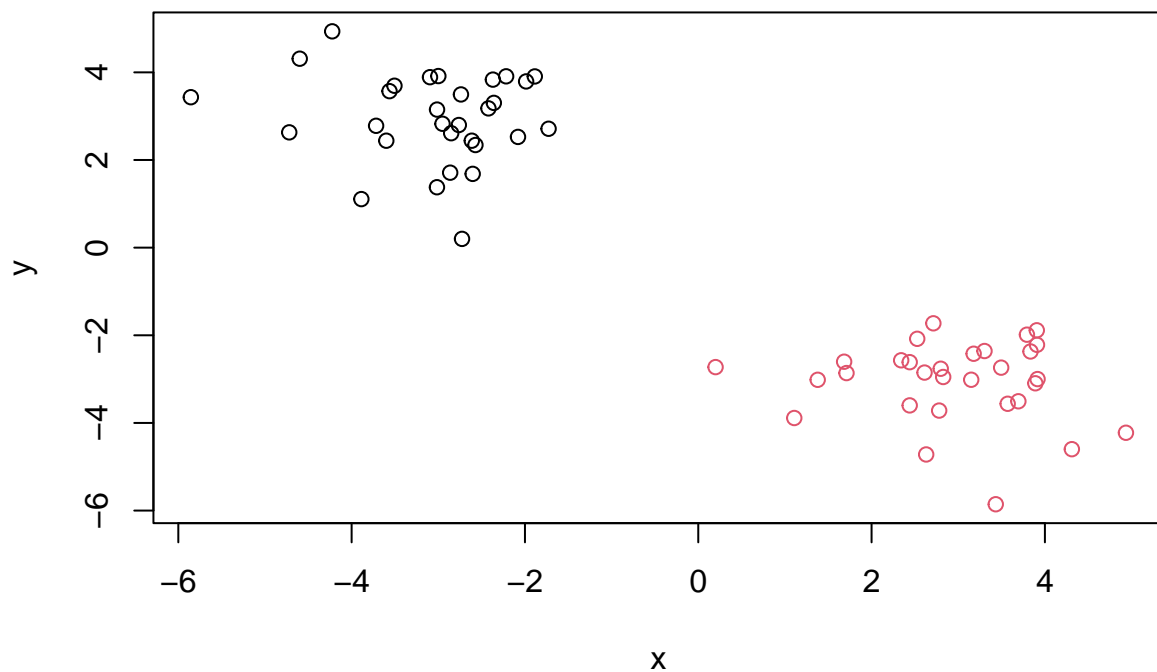
To get cluster membership vector we must “cut” tree with ‘cutree()’.

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

[illegible]

Now we plot our data with hclust results!

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



Principial Component Analysis

PCA of UK food data

Read data from website and try a few visualizations!

```
url <- "https://tinyurl.com/UK-foods"
x <- read.csv(url, row.names=1)
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
## Fresh_potatoes	720	874	566	1033
## Fresh_Veg	253	265	171	143
## Other_Veg	488	570	418	355
## Processed_potatoes	198	203	220	187
## Processed_Veg	360	365	337	334
## Fresh_fruit	1102	1137	957	674
## Cereals	1472	1582	1462	1494

```
## Beverages          57    73    53    47
## Soft_drinks       1374  1256  1572  1506
## Alcoholic_drinks   375   475   458   135
## Confectionery      54    64    62    41
```

Q1. How many rows and columns are in your new data frame named x? What R functions could you use to answer this questions?

```
dim(x)
```

```
## [1] 17  4
```

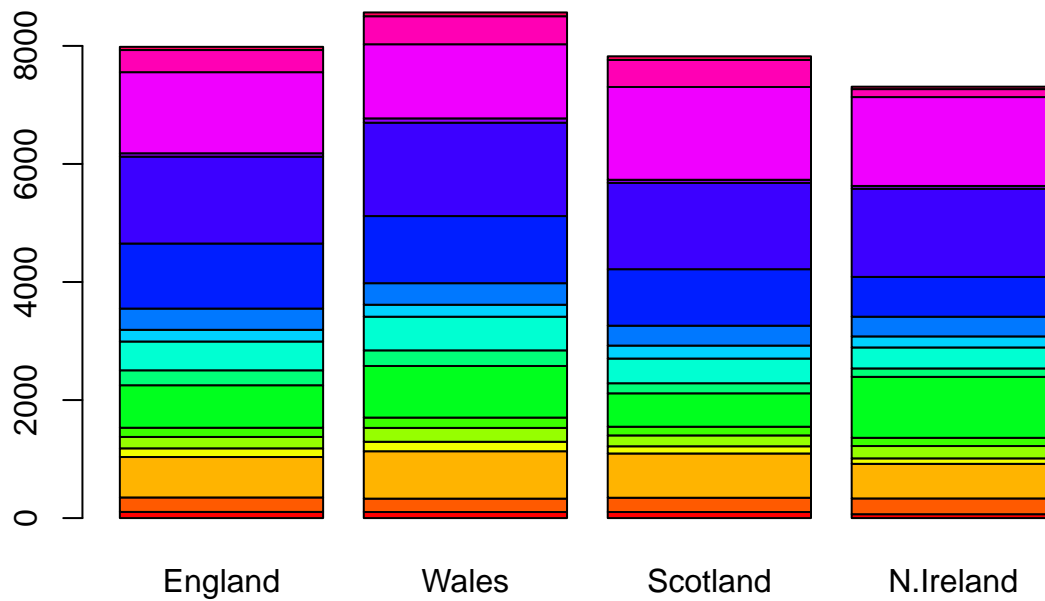
Q2. Which approach to solving the ‘row-names problem’ mentioned above do you prefer and why? Is one approach more robust than another under certain circumstances?

I prefer the alternative, second option as `x <- x[,-1]` may accidentally remove additional columns as the code block is repeated multiple times.

```
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
```

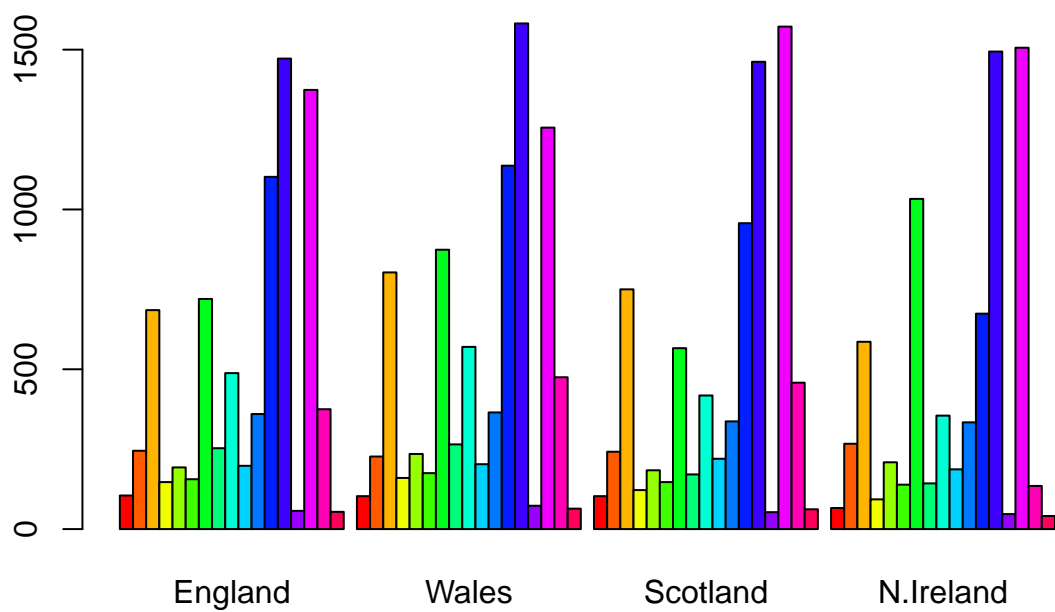
```
cols <- rainbow(nrow(x))
barplot(as.matrix(x), col = cols)
```



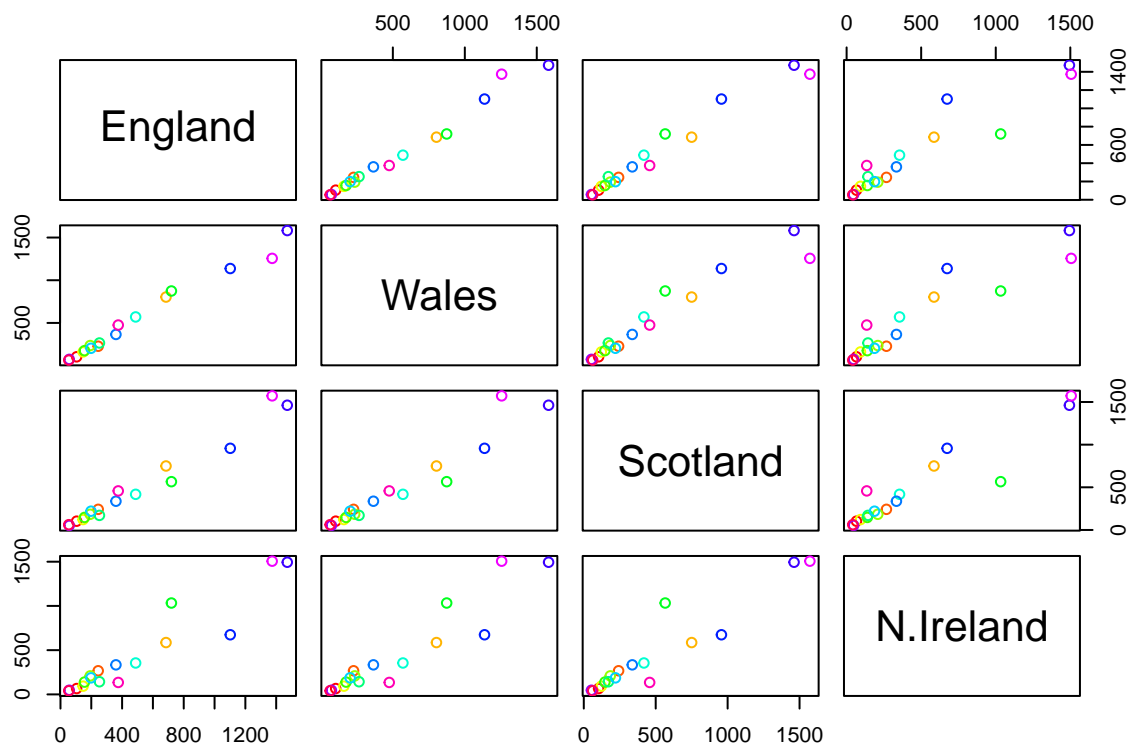
Q3: Changing what optional argument in the *BELOW* (REVERSED ORDER) `barplot()` function results in the following plot (*ABOVE*)?

Setting `beside=FALSE` changes the graph to the one above since the bars are no longer beside one another. By default, bars are on top of each other unless `beside` is set to true.

```
barplot(as.matrix(x), col = cols, beside = TRUE)
```



```
pairs(x, col=cols)
```

> 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?

Each graph represents one country being compared with the other in pairs (hence the name of the function `pairs()`). A given point that lies diagonal on a plot means that two values are similar to each other (example: countries drink similar amount of milk). Points that are not diagonal represent values that are not similar to each other (one country eats a lot of a specific food that another country doesn't).

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

N. Ireland has more data points that are off the diagonal indicating that their eating habits differ the most compared to the other countries.

PCA to the rescue! The main base R PCA function is '`prcomp()`' and it will be trasnposed!

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

```
## Standard deviations (1, ..., p=4):
## [1] 3.241502e+02 2.127478e+02 7.387622e+01 4.188568e-14
##
## Rotation (n x k) = (17 x 4):
##
##          PC1          PC2          PC3          PC4
## Cheese    -0.056955380 -0.016012850 -0.02394295 -0.691718038
## Carcass_meat  0.047927628 -0.013915823 -0.06367111  0.635384915
## Other_meat  -0.258916658  0.015331138  0.55384854  0.198175921
```

```
## Fish -0.084414983 0.050754947 -0.03906481 -0.015824630
## Fats_and_oils -0.005193623 0.095388656 0.12522257 0.052347444
## Sugars -0.037620983 0.043021699 0.03605745 0.014481347
## Fresh_potatoes 0.401402060 0.715017078 0.20668248 -0.151706089
## Fresh_Veg -0.151849942 0.144900268 -0.21382237 0.056182433
## Other_Veg -0.243593729 0.225450923 0.05332841 -0.080722623
## Processed_potatoes -0.026886233 -0.042850761 0.07364902 -0.022618707
## Processed_Veg -0.036488269 0.045451802 -0.05289191 0.009235001
## Fresh_fruit -0.632640898 0.177740743 -0.40012865 -0.021899087
## Cereals -0.047702858 0.212599678 0.35884921 0.084667257
## Beverages -0.026187756 0.030560542 0.04135860 -0.011880823
## Soft_drinks 0.232244140 -0.555124311 0.16942648 -0.144367046
## Alcoholic_drinks -0.463968168 -0.113536523 0.49858320 -0.115797605
## Confectionery -0.029650201 -0.005949921 0.05232164 -0.003695024
```

Summary of what PCA is doing!

```
summary(pca)
```

```
## Importance of components:
##              PC1      PC2      PC3      PC4
## Standard deviation 324.1502 212.7478 73.87622 4.189e-14
## Proportion of Variance 0.6744 0.2905 0.03503 0.000e+00
## Cumulative Proportion 0.6744 0.9650 1.00000 1.000e+00
```

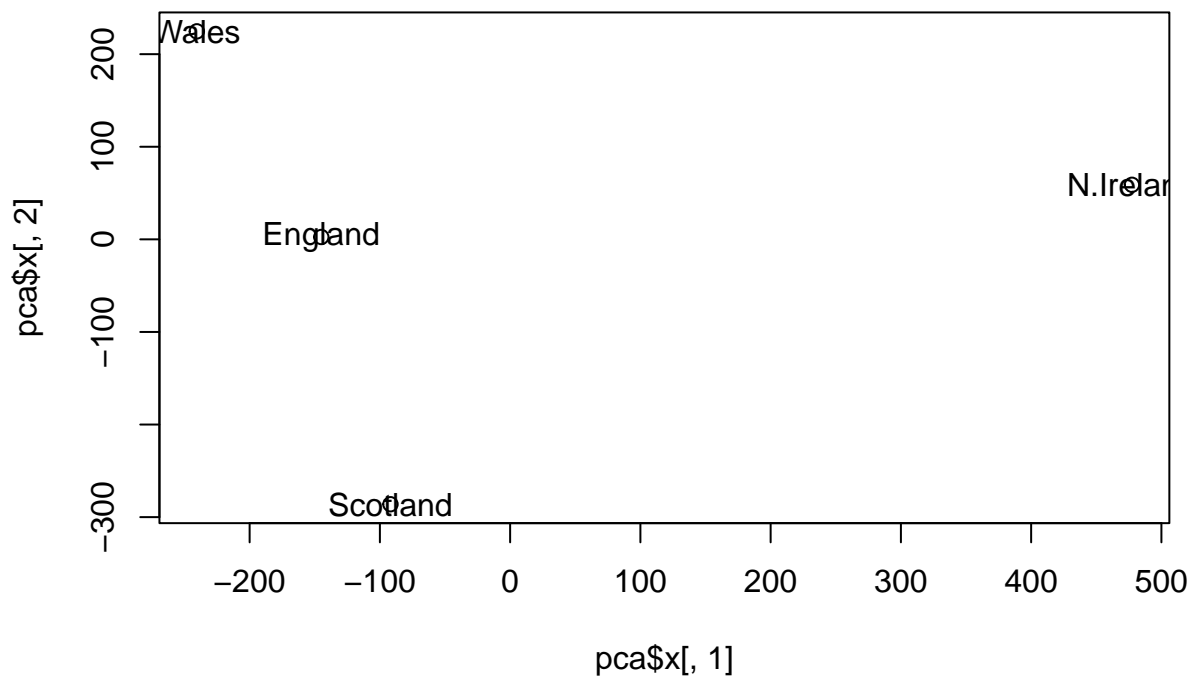
```
attributes(pca)
```

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

To make new informative PCA plot (PCA scores) we access 'pca\$x'

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

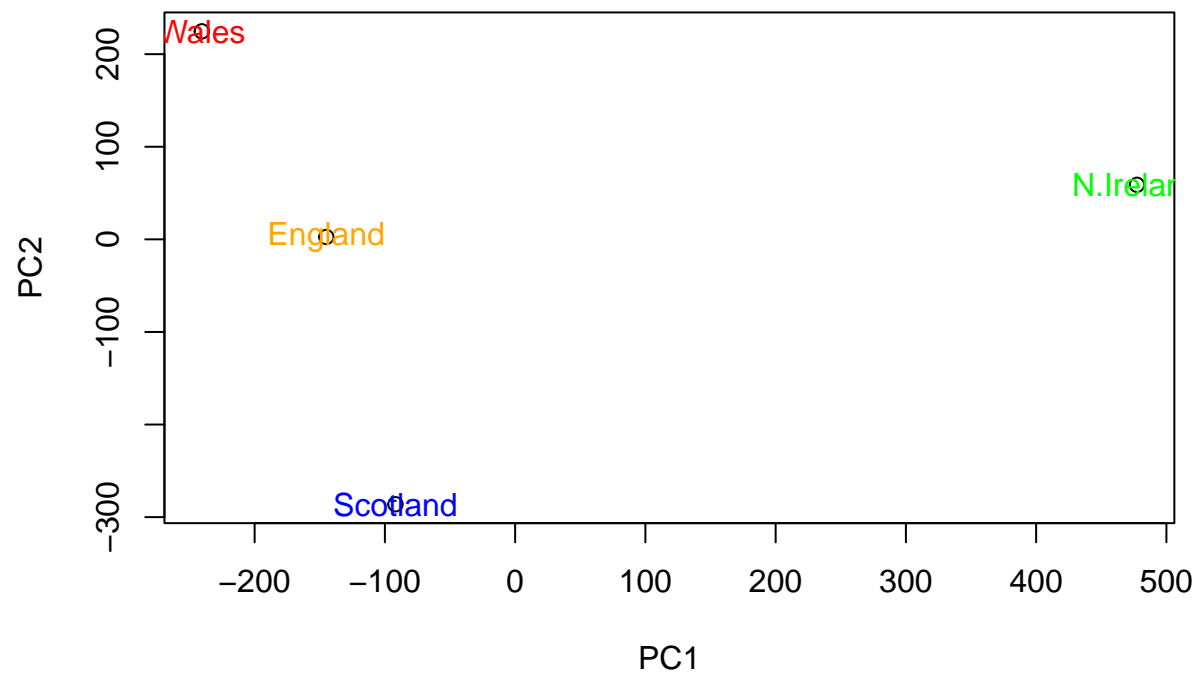
```
plot(pca$x[,1], pca$x[,2])
text(pca$x[,1], pca$x[,2], colnames(x))
```



Color the plot!

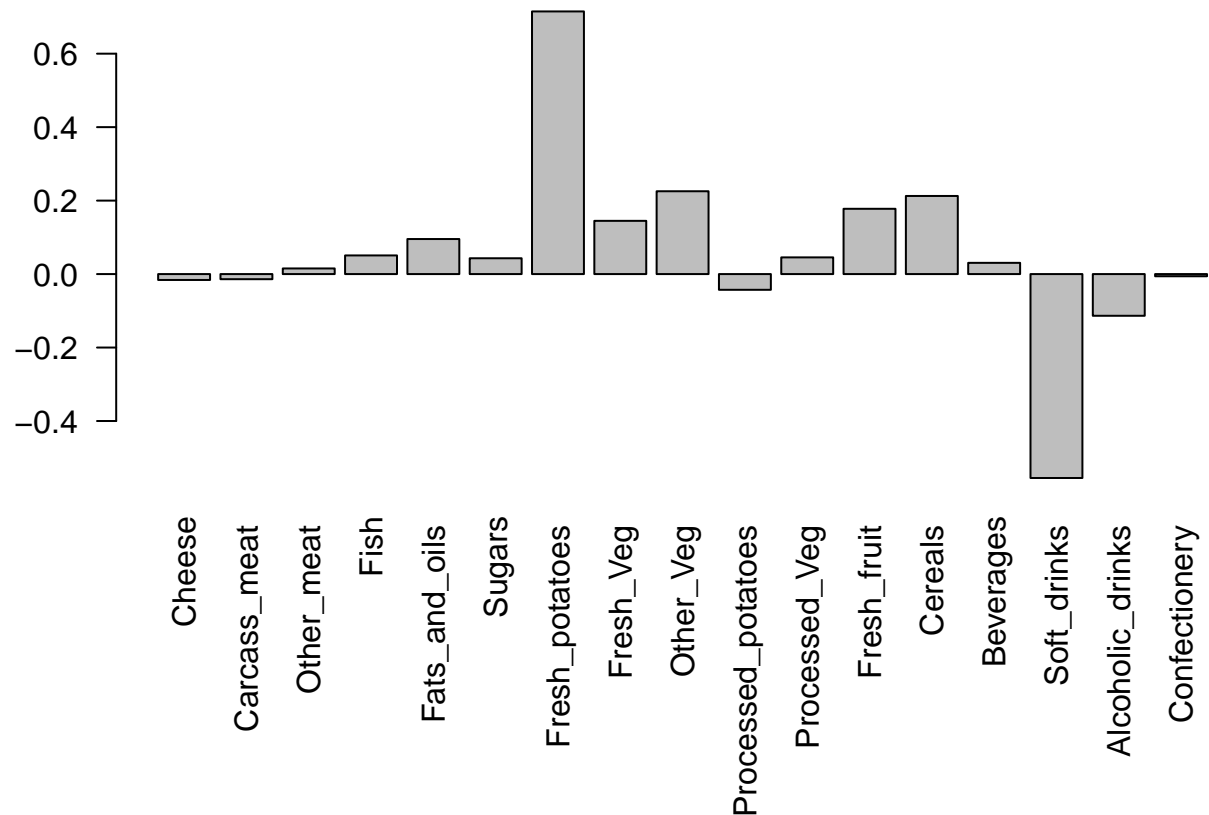
Q8. Customize your plot so that the colors of the country names match the colors in our UK and Ireland map and table at start of this document.

```
country_cols <- c("orange", "red", "blue", "green")
plot(pca$x[,1], pca$x[,2], xlab = "PC1", ylab="PC2")
text(pca$x[,1], pca$x[,2], colnames(x),
     col=country_cols)
```



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

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



Soft drinks and fresh potatoes are the two most prominent food groups! PC2 tells us about the variation between the data and shows strong patterns.

PCA of RNA-Seq Data

Read data from website

```
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
```

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

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

```
## [1] 100  10
```

There are 10 samples and 6 genes.

```
pca <- prcomp(t(rna.data))
summary(pca)
```

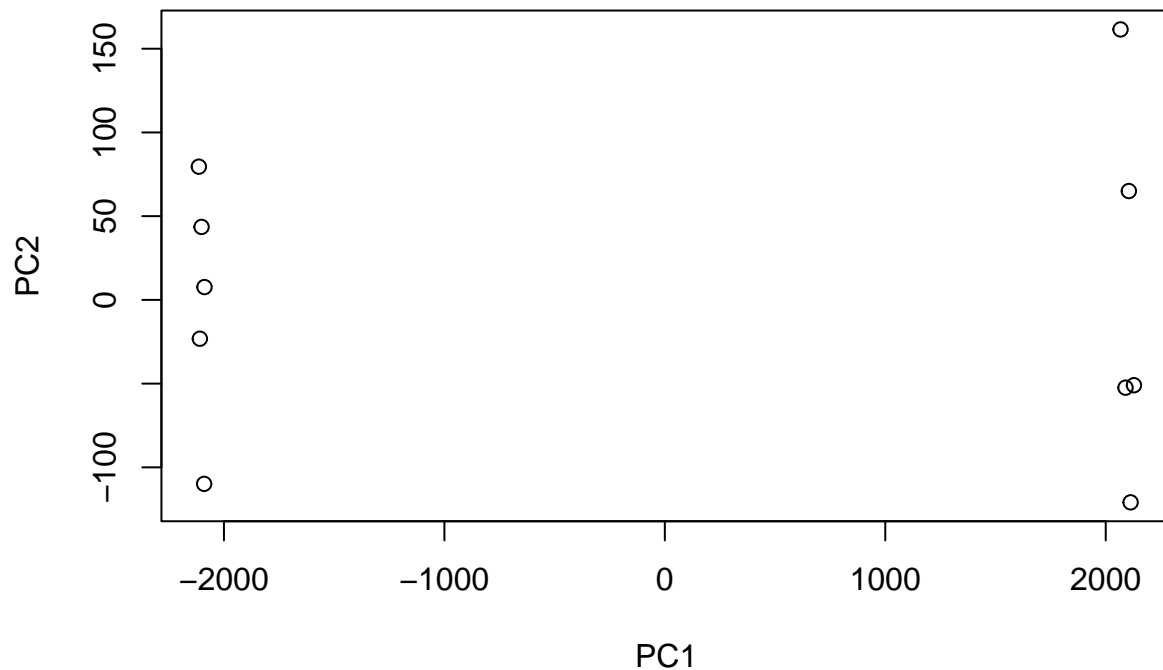
Importance of components:

##	PC1	PC2	PC3	PC4	PC5	PC6
## Standard deviation	2214.2633	88.9209	84.33908	77.74094	69.66341	67.78516
## Proportion of Variance	0.9917	0.0016	0.00144	0.00122	0.00098	0.00093
## Cumulative Proportion	0.9917	0.9933	0.99471	0.99593	0.99691	0.99784

##	PC7	PC8	PC9	PC10
## Standard deviation	65.29428	59.90981	53.20803	3.142e-13
## Proportion of Variance	0.00086	0.00073	0.00057	0.000e+00
## Cumulative Proportion	0.99870	0.99943	1.00000	1.000e+00

PCR plot of RNA-Seq Data

```
plot(pca$x[,1], pca$x[,2], xlab = "PC1", ylab = "PC2")
```



```
plot(pca$x[,1], pca$x[,2], xlab = "PC1", ylab = "PC2")
text(pca$x[,1], pca$x[,2], colnames(rna.data))
```

