Class 7 Lab

Garrett Cole

Data Import

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

Question 1

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

For # of rows in a data frame, you can use the nrow() function

```
nrow(x)
```

```
[1] 17
```

For # of columns in a data frame, you can use the ncol() function

```
ncol(x)
```

```
[1] 5
```

Checking your data

```
#View(x)
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
```

```
tail(x)
```

```
                  X England Wales Scotland N.Ireland
12      Fresh_fruit    1102  1137      957       674
13          Cereals    1472  1582     1462      1494
14        Beverages      57    73       53        47
15      Soft_drinks    1374  1256     1572      1506
16 Alcoholic_drinks     375   475      458       135
17    Confectionery      54    64       62        41
```

Fix alignment of rows

```
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
```

```
#Or can do it like this when reading in file
#x <- read.csv(url, row.names=1)
#head(x)
```

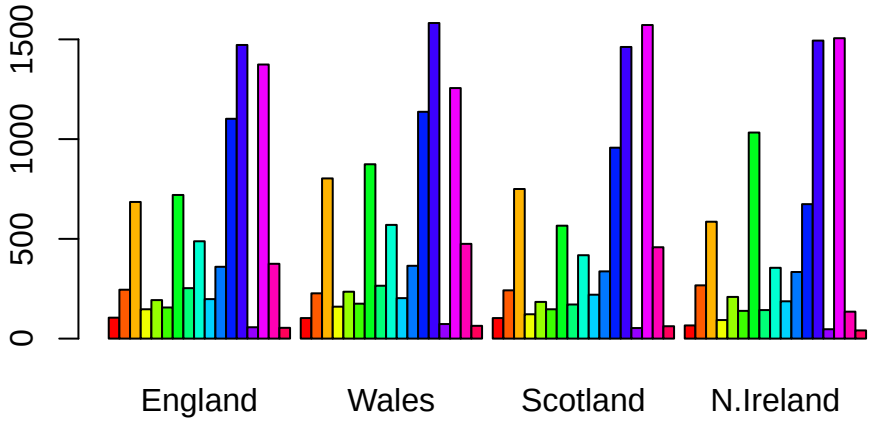
Question 2

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?

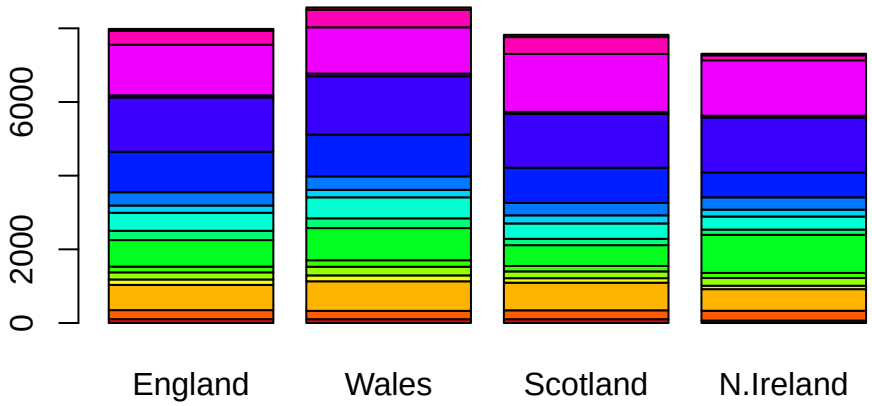
I prefer the read.csv way since it is more simple and requires only one line of code. This approach is more robust since if you do the first approach numerous times then it will delete one the left most column causing the row names to be deleted and will keep deleting the columns if you repeat.

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)))
```



Question 3

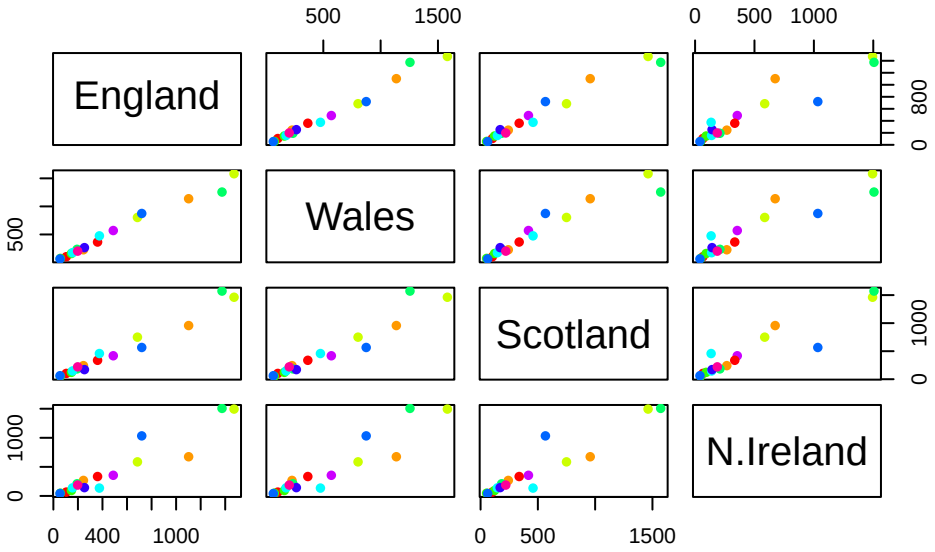
Changing what optional argument in the above barplot() function results in the following plot?

Changing the optional argument beside() from true to false such as beside = F would result in that plot

Question 5

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?

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



If a given point lines up on the diagonal for a given plot then that means the food category that the plot is in, is equal between the two countries that the plot is comparing.

Question 6

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

The main differences between N. Ireland and the other countries is that in the plots comparing N. Ireland and the rest of the countries most of the data points are skewed towards the other countries' side of the plots. This represents that N. Ireland has the least amount of food for most categories compared to the other countries

PCA to the Rescue

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

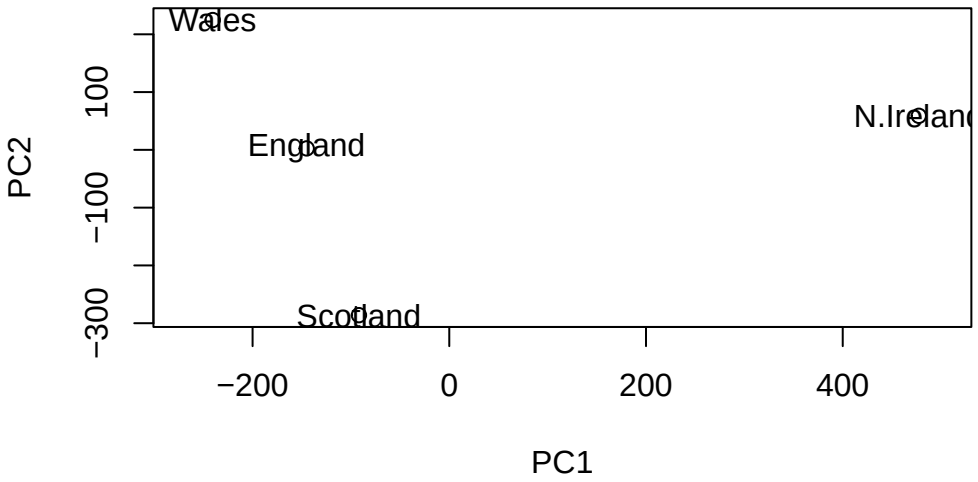
```
Importance of components:
                            PC1      PC2      PC3        PC4
Standard deviation     324.1502 212.7478 73.87622 4.189e-14
Proportion of Variance   0.6744   0.2905  0.03503 0.000e+00
```

```
Cumulative Proportion    0.6744   0.9650  1.00000 1.000e+00
```

Question 7

Complete the code below to generate a plot of PC1 vs. PC2.

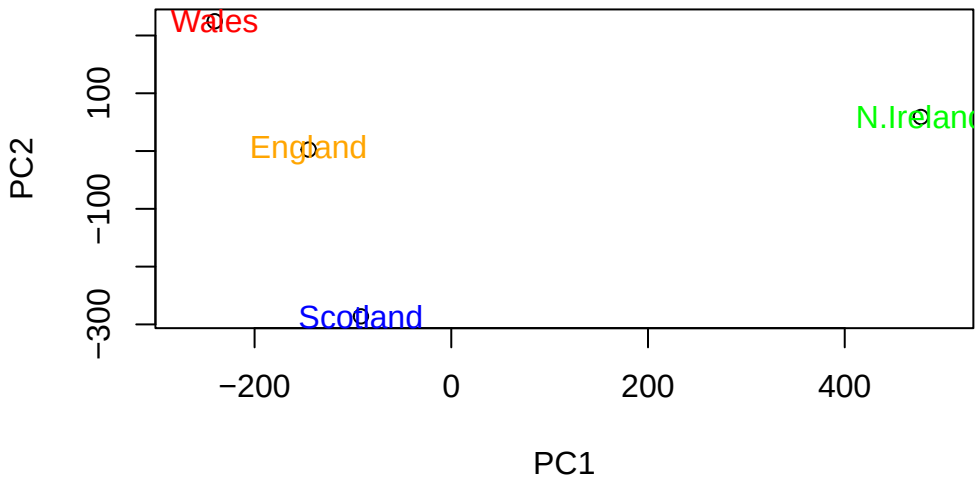
```
plot(pca$x[,1], pca$x[,2], xlab="PC1", ylab="PC2", xlim=c(-270,500))
text(pca$x[,1], pca$x[,2], colnames(x))
```



Question 8

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

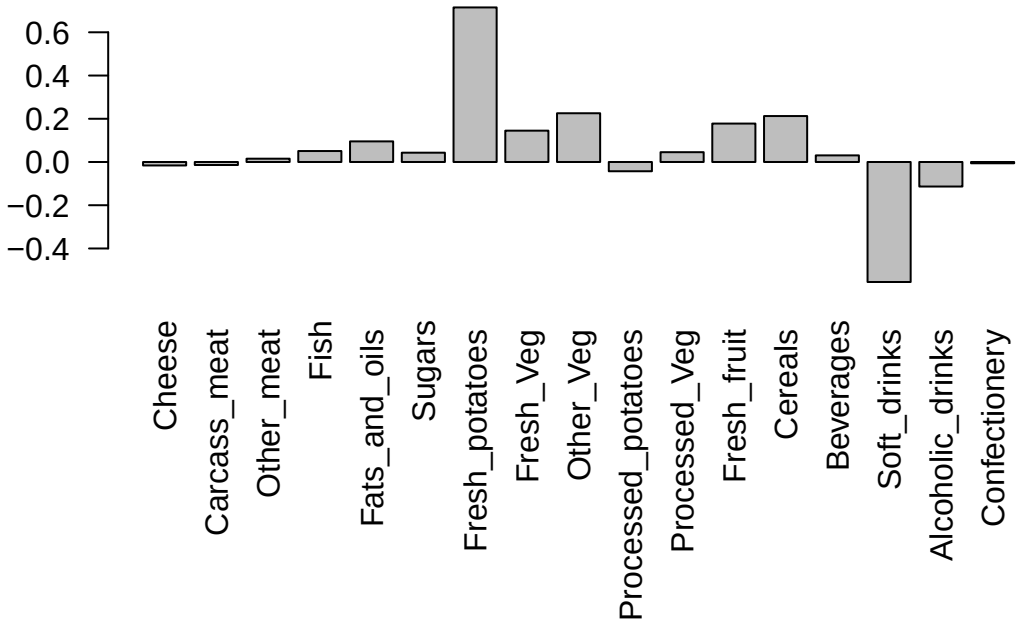
```
plot(pca$x[,1], pca$x[,2], xlab="PC1", ylab="PC2", xlim=c(-270,500))
text(pca$x[,1], pca$x[,2], colnames(x), col = c("orange", "red", "blue", "green"))
```



Question 9

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

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



The two food groups feature prominantely are Fresh_potatoes and Soft_drinks, the highest positive score is Fresh_potatoes which tells us that it is one of the main contributors to "pushing" N. Ireland to the right positive side of the plot. The highest negative score is Soft_drinks which tells us that it is a main contributor to "pushing" other countries to the left side of the plot.

PCA of RNA-seq data Read in data from website

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

Question 10

How many genes and samples are in this data set? Genes: 100

```
nrow(rna.data)
```

```
[1] 100
```

Samples: 10

```r
ncol(rna.data)
```

```
[1] 10
```