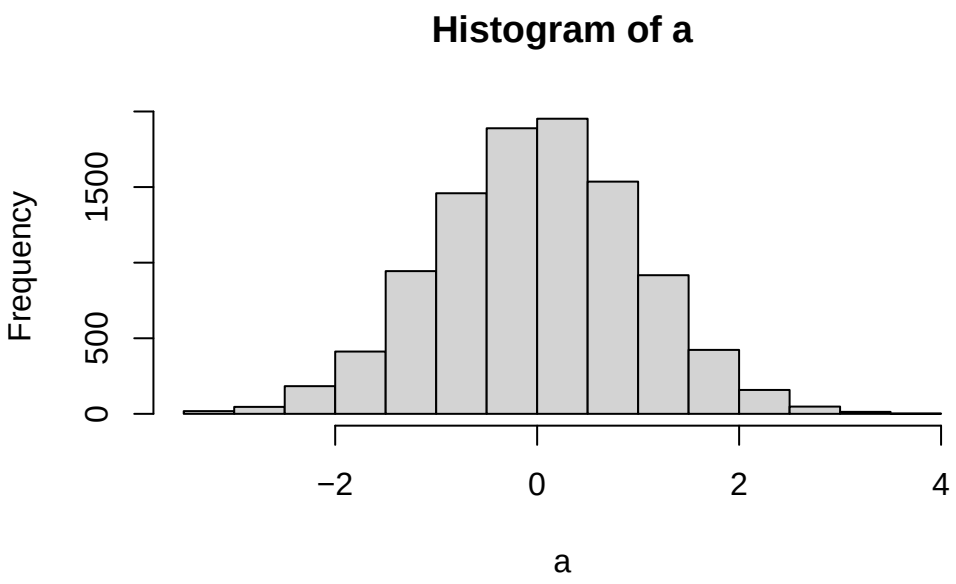
# Class7

Laura Biggs

### K means clustering

Test how this method works with made up data

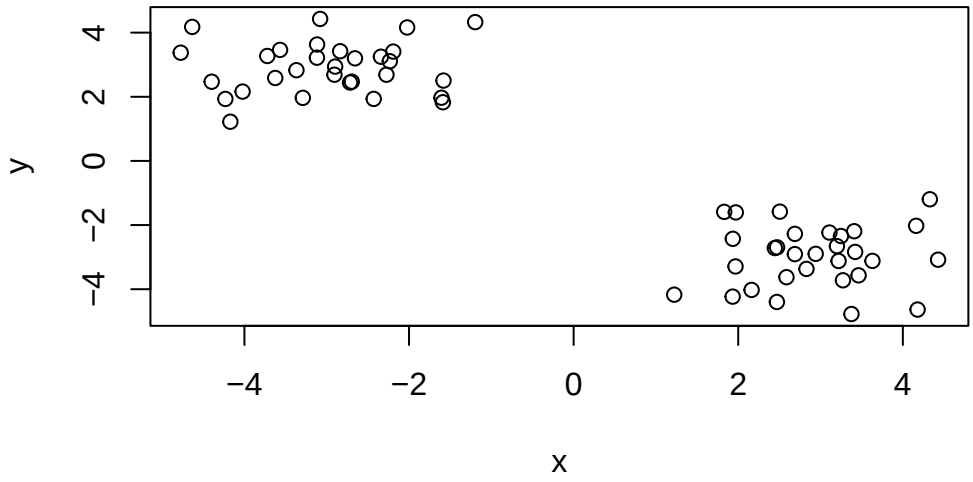
```
a <- rnorm(10000)
hist(a)
```




Make some numbers centered on -3

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

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




K means

```r
km <- kmeans(b, centers = 2, nstart = 20)
km
```

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

Cluster means:
          x         y
1  2.902746 -2.976274
2 -2.976274  2.902746

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] 44.93557 44.93557
```

```
 (between_SS / total_SS =  92.0 %)

Available components:

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

```
#How many points are in each cluster?
km$size
```

```
[1] 30 30
```

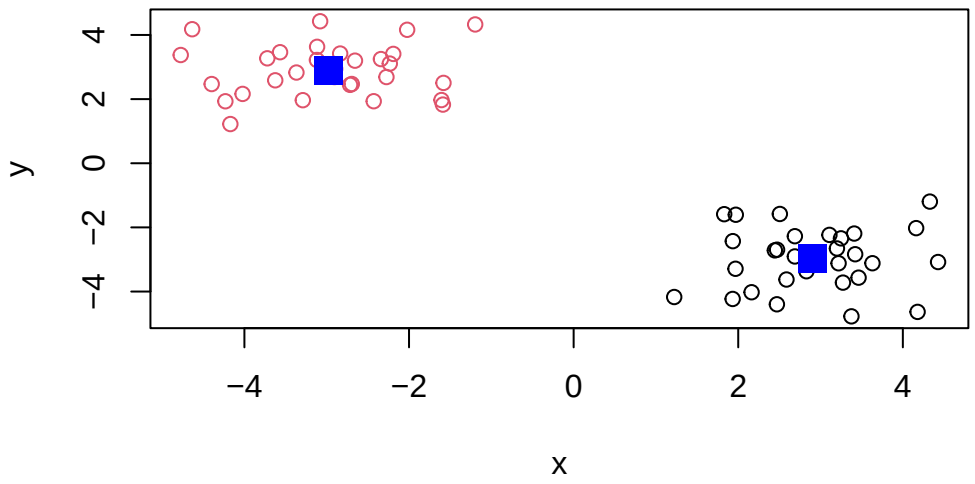
```
#Cluster assignment?
km$cluster
```

```
 [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
```

```
#Cluster center?
km$centers
```

```
          x         y
1  2.902746 -2.976274
2 -2.976274  2.902746
```

```
#Plot of b colored by results
plot(b, col = km$cluster)
points(km$centers, col ='blue', pch=15, cex =2)
```



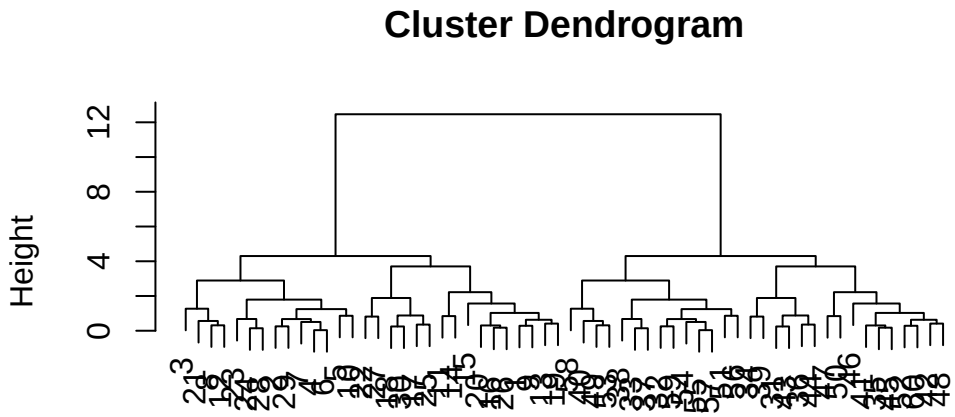

#Heirarchical clustering

The 'hclust()' function requires an input distance matrix

```
# Use hclust()
hc <- hclust(dist(b))
hc
```

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

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

```
plot(hc)
```

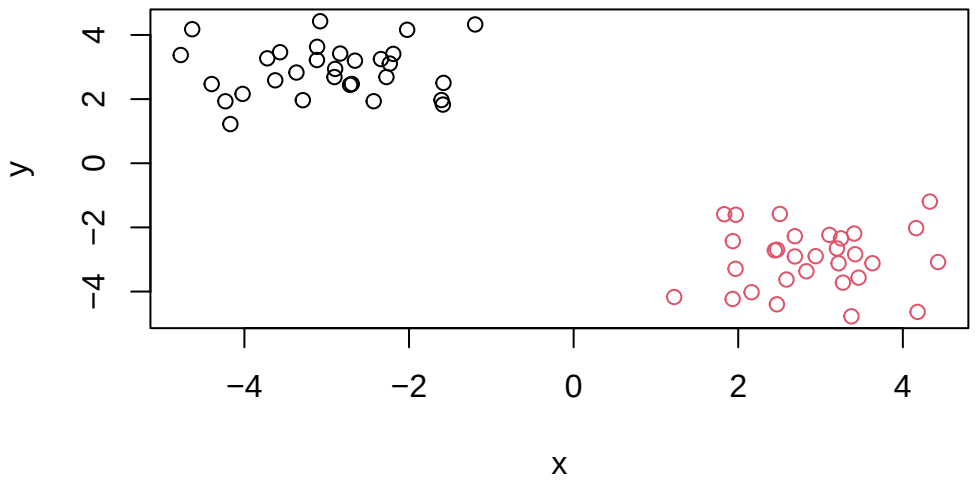



Cut the tree to yield separate branches with each leave being a cluster. Use the 'cutree()' function

```
#cutree with height
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 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
```

```
#cutree with k=2
grps <- cutree(hc, k=2)

#plot of data colored by groups
plot(b, col=grps)
```




## Q1

Import UK foods data

```
url <- "https://tinyurl.com/UK-foods"
x <- read.csv(url)
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
7      Fresh_potatoes     720   874      566      1033
8           Fresh_Veg     253   265      171       143
9           Other_Veg     488   570      418       355
10 Processed_potatoes     198   203      220       187
11      Processed_Veg     360   365      337       334
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
```

```
#What are the dimensions of the data?
dim(x)
```

```
[1] 17  5
```

```
#Preview the 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
```

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

17 rows, 5 columns dim(x)

Fix the # of columns in the data

```
#Assign the row names to the first column w/rownames
#Remove the X column with [,-1]; be careful as x[,-1] deletes columns for every repeat lin
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
```

```
dim(x)
```

```
[1] 17  4
```

Preferred approach to assigning row names

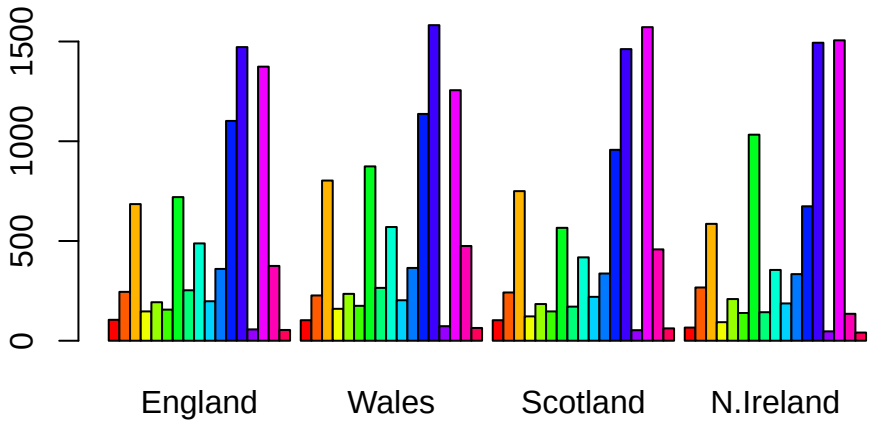
```
#Automatically assigns the first column as the row names
#x <- read.csv(url, row.names=1)
#head(x)
```

> 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?

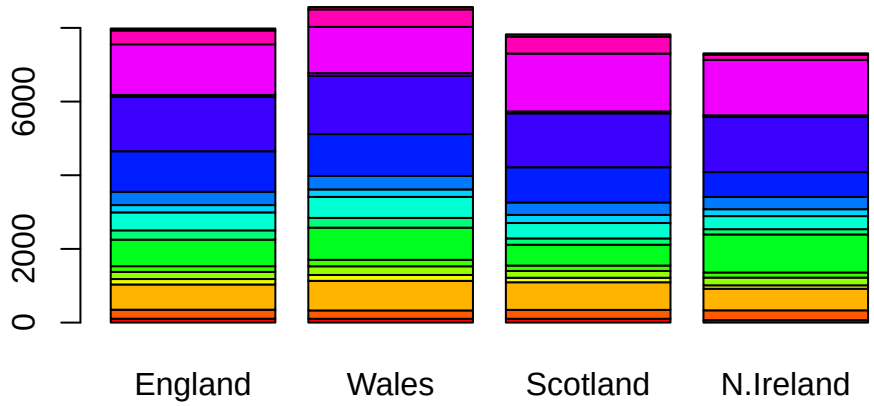
Piping in the row.names = 1 when the initial data is loaded in is preferable as the minus indexing removes columns for every subsequent run and is therefore more prone to human error.

Visualizing the data

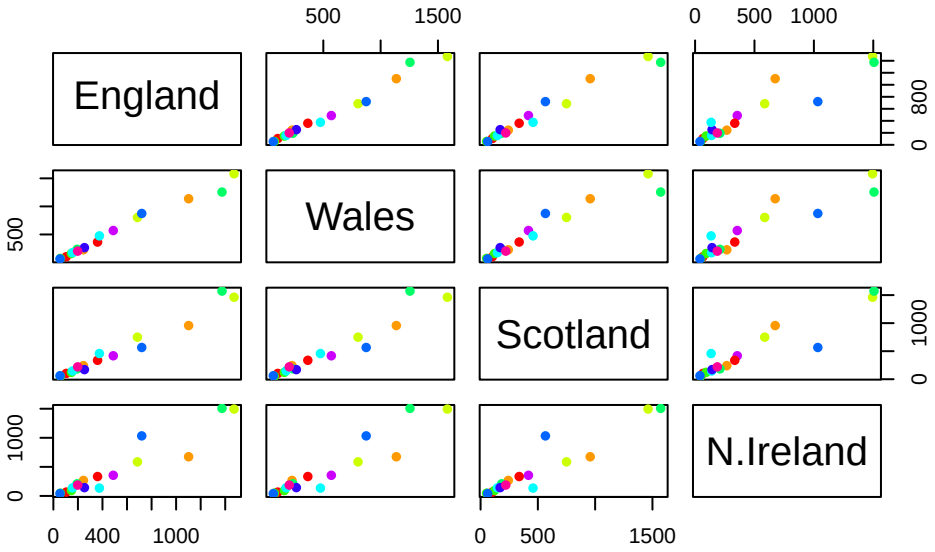
```
#As a bar plot with each row represented by a color
barplot(as.matrix(x), beside=T, col=rainbow(nrow(x)))
```



```r
#Stacked bar plots
barplot(as.matrix(x), beside=F, col=rainbow(nrow(x)))
```




```r
#Pairwise plot
pairs(x, col=rainbow(10), pch=16)
```




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

Changing the beside argument to false creates stacked bar plots. barplot(as.matrix(x), beside=F, col=rainbow(nrow(x)))

Q4: Missing

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?

Pairwise plots were difficult to interpret. It would be easier to interpret with fewer variables. Points that lie on the diagonal with the 2 countries being compared represent similarity between the two values. The more linear the correlation looks, the more similar the variable are between the 2 countries.

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

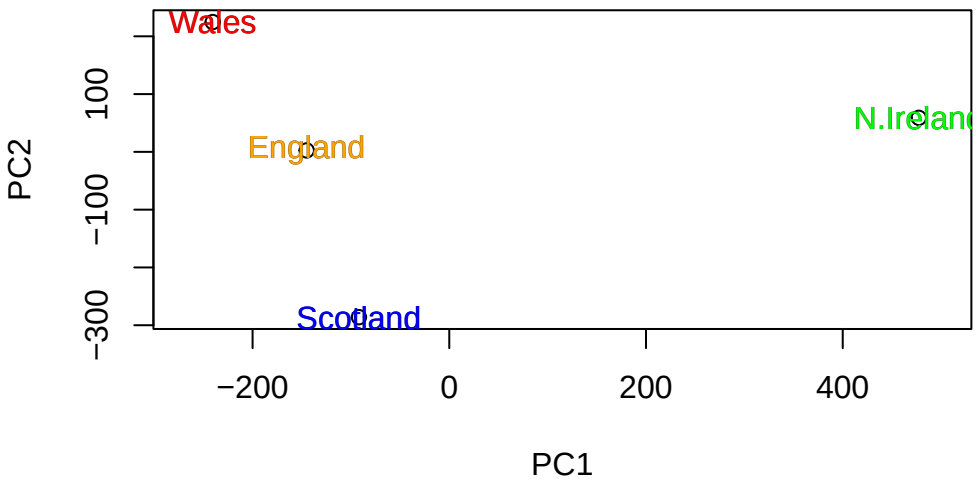
The main differences are the heightened consumption of fresh potatoes in Northern Ireland and the reduced consumption of fresh fruit relative to the other contries in the UK.

Base R PCA plots

```
#Must first transpose data as prcomp takes observations as rows and variables as columns
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
```

```
#PC1 vs PC2; PC1 is first col, PC2 2nd col; index with transposed pca information
plot(pca$x[,1], pca$x[,2], xlab="PC1", ylab="PC2", xlim=c(-270,500))
text(pca$x[,1], pca$x[,2], colnames(x))

#Add color to the text
text(pca$x[,1], pca$x[,2], colnames(x), col = c("orange", "red", "blue", "green"))
```




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.

text(pca$x[, 1], pca$x[,2], colnames(x), col = c("orange", "red", "blue", "green"))

Principal component variation

```
#How much variation each PC accounts for with sdev
v <- round(pca$sdev^2/sum(pca$sdev^2) * 100)
v
```

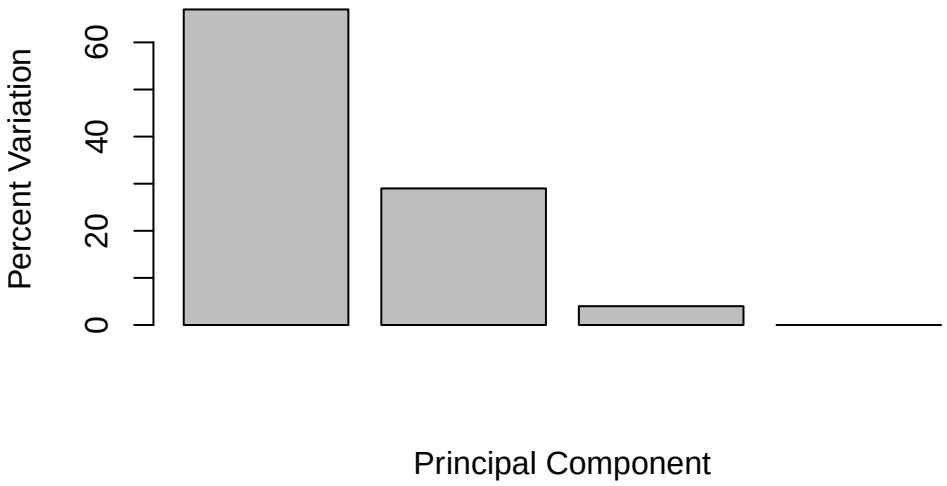
```
[1] 67 29  4  0
```

```
#Simplified version
z <- summary(pca)
z$importance
```

```
                             PC1       PC2      PC3           PC4
Standard deviation     324.15019 212.74780 73.87622 4.188568e-14
Proportion of Variance   0.67444   0.29052  0.03503 0.000000e+00
Cumulative Proportion    0.67444   0.96497  1.00000 1.000000e+00
```

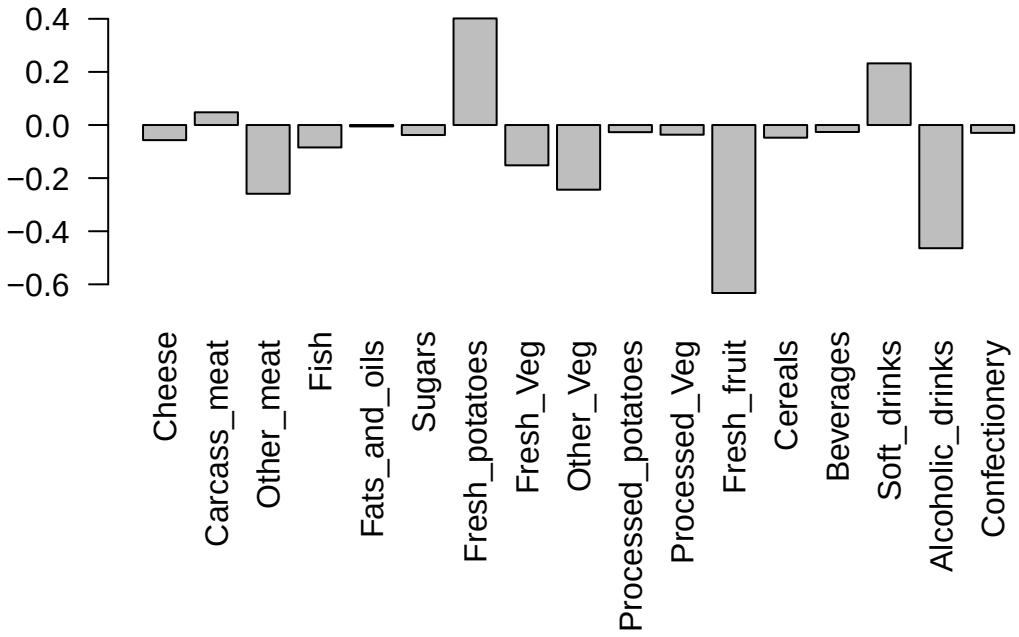
Barplot of variances

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

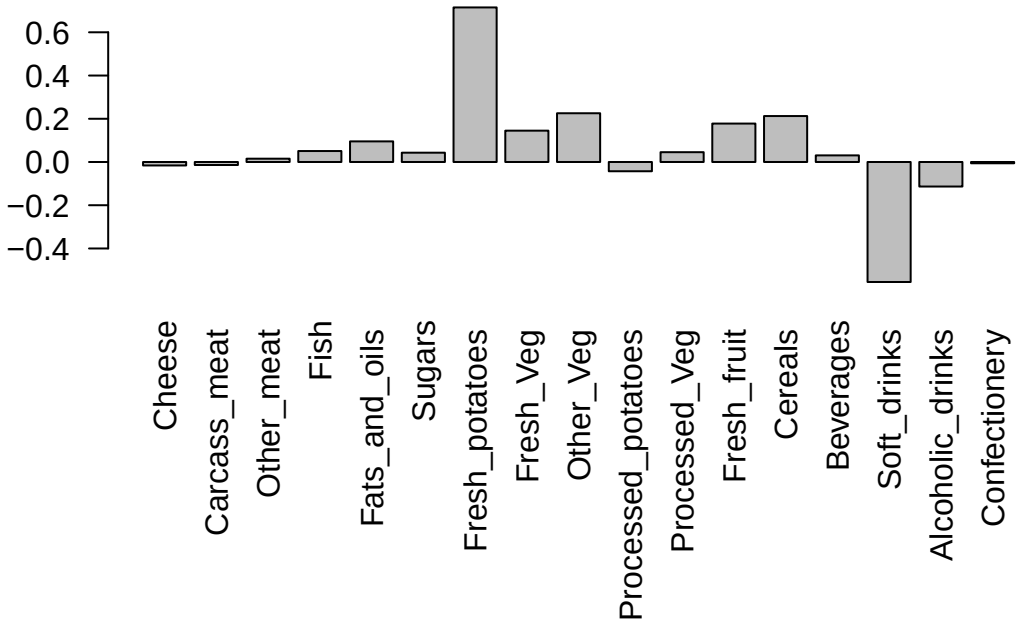



Variable loadings; positive loading score

```
#PC1
par(mar=c(10, 3, 0.35, 0))
barplot(pca$rotation[,1], las=2)
```




```r
#PC2
par(mar=c(10, 3, 0.35, 0))
barplot(pca$rotation[,2], las=2)
```

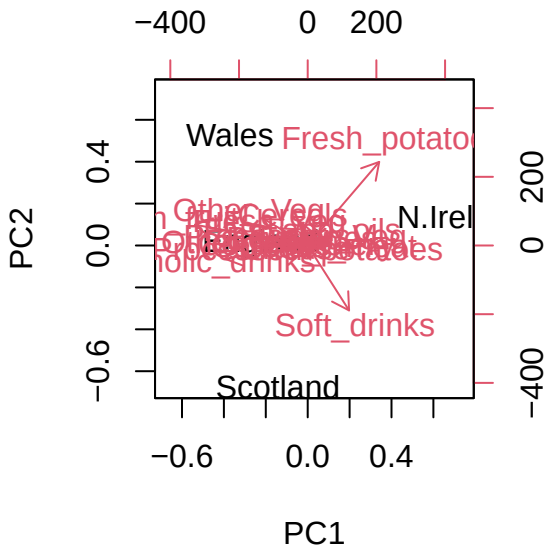



Q9: Generate a similar 'loadings plot' for PC2. What two food groups feature prominantely and what does PC2 maninly tell us about?

PC2 features fresh potatoes and soft drinks which contribute to the most to the vertical component of the variation across the 17 dimensions of 4 countries. par(mar=c(10, 3, 0.35, 0)) barplot(pca$rotation[,2], las=2)

Biplots

```
biplot(pca)
```




## Q2

Read in 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)
```

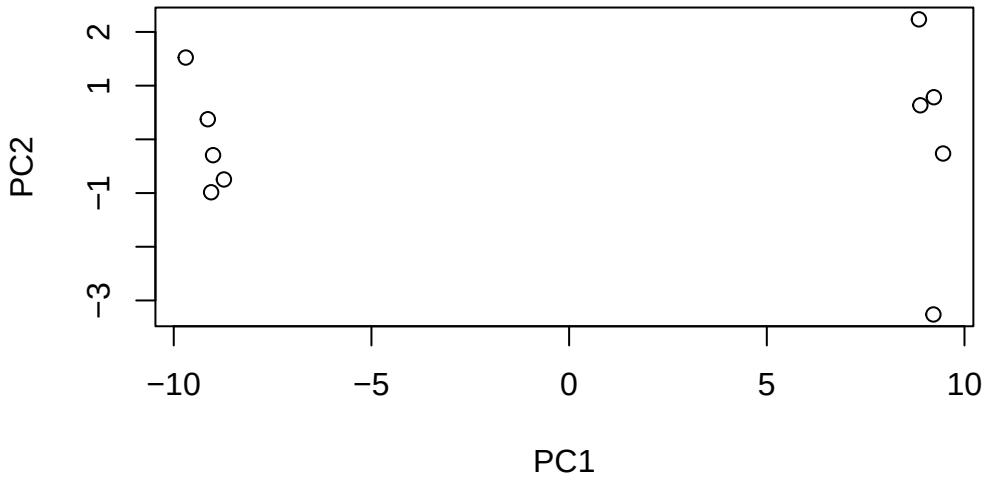
```
[1] 100  10
```

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

100 genes and 10 samples of differing conditions. dim(rna.data)

RNA seq PCA

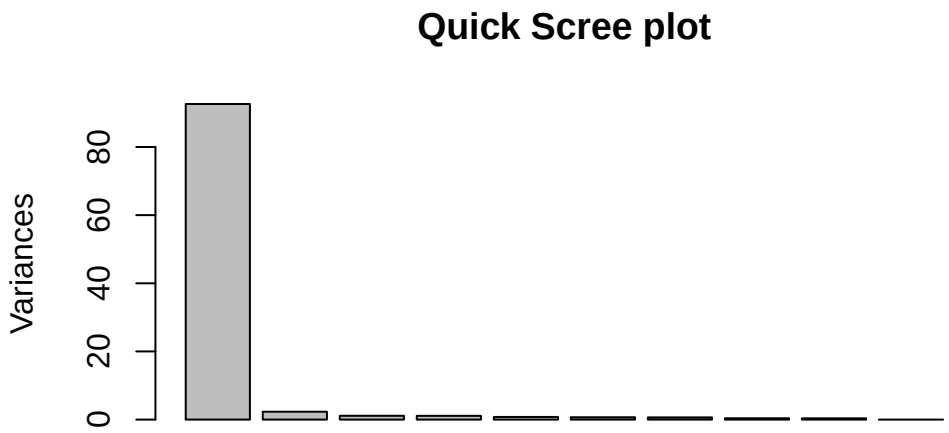
```
#Transpose rna.data
pca <- prcomp(t(rna.data), scale = TRUE)

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




```
summary(pca)
```

```
Importance of components:
                          PC1    PC2     PC3     PC4     PC5     PC6     PC7
Standard deviation     9.6237 1.5198 1.05787 1.05203 0.88062 0.82545 0.80111
Proportion of Variance 0.9262 0.0231 0.01119 0.01107 0.00775 0.00681 0.00642
Cumulative Proportion  0.9262 0.9493 0.96045 0.97152 0.97928 0.98609 0.99251
                           PC8     PC9      PC10
Standard deviation     0.62065 0.60342 3.348e-15
Proportion of Variance 0.00385 0.00364 0.000e+00
Cumulative Proportion  0.99636 1.00000 1.000e+00
```

```r
#Scree plot
plot(pca, main="Quick Scree plot")
```



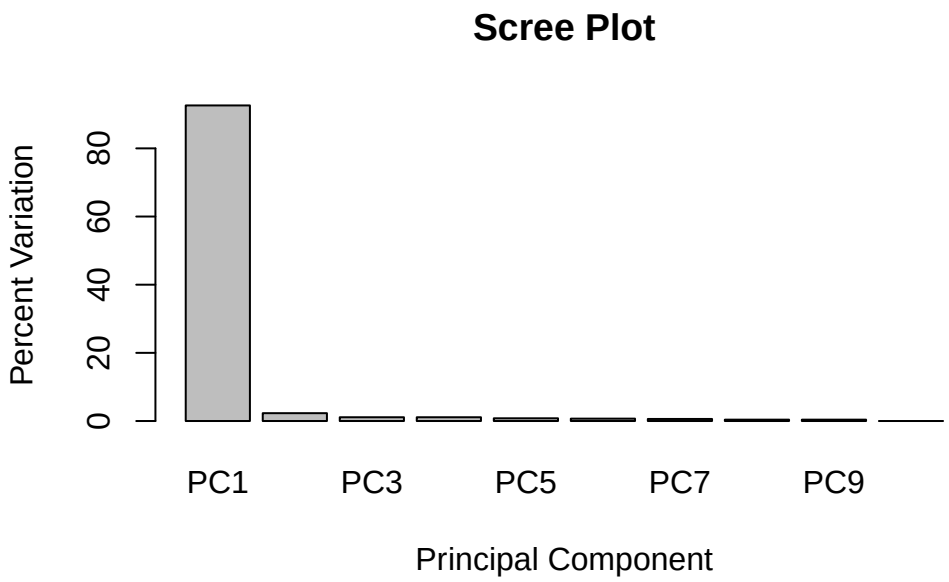

PCA variance x/prcomp

```r
#Variance
pca.var <- pca$sdev^2

#Percent variation
pca.var.per <- round(pca.var/sum(pca.var)*100, 1)
pca.var.per
```

```
 [1] 92.6  2.3  1.1  1.1  0.8  0.7  0.6  0.4  0.4  0.0
```

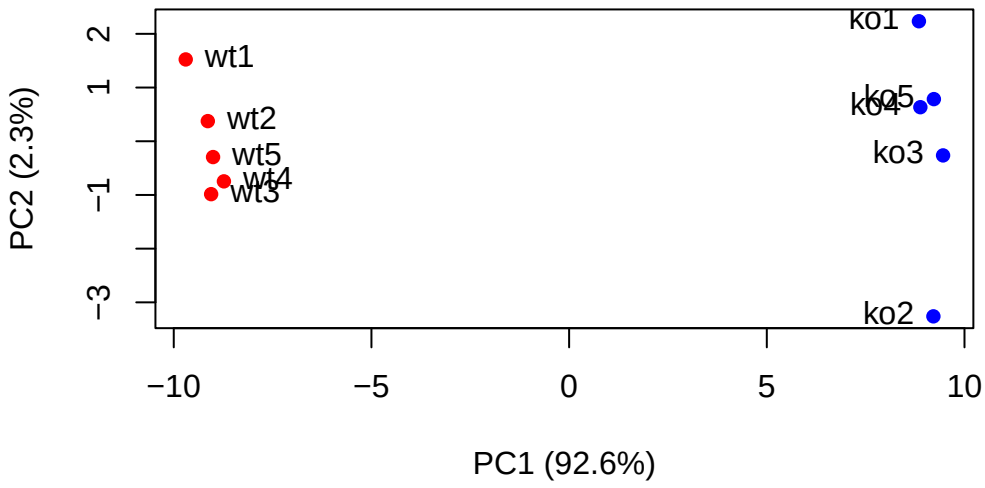
Scree plot

```r
barplot(pca.var.per, main = "Scree Plot",
        names.arg = paste0("PC", 1:10),
        xlab="Principal Component", ylab="Percent Variation")
```




Improved PCA plot w/WT and KO titles

```
colvec <- colnames(rna.data)
colvec[grep("wt", colvec)] <- "red"
colvec[grep("ko", colvec)] <- "blue"

#?grep

plot(pca$x[,1], pca$x[,2], col=colvec, pch=16,
     xlab=paste0("PC1 (", pca.var.per[1], "%)"),
     ylab=paste0("PC2 (", pca.var.per[2], "%)"))

text(pca$x[,1], pca$x[,2], labels = colnames(rna.data), pos=c(rep(4,5), rep(2,5)))
```

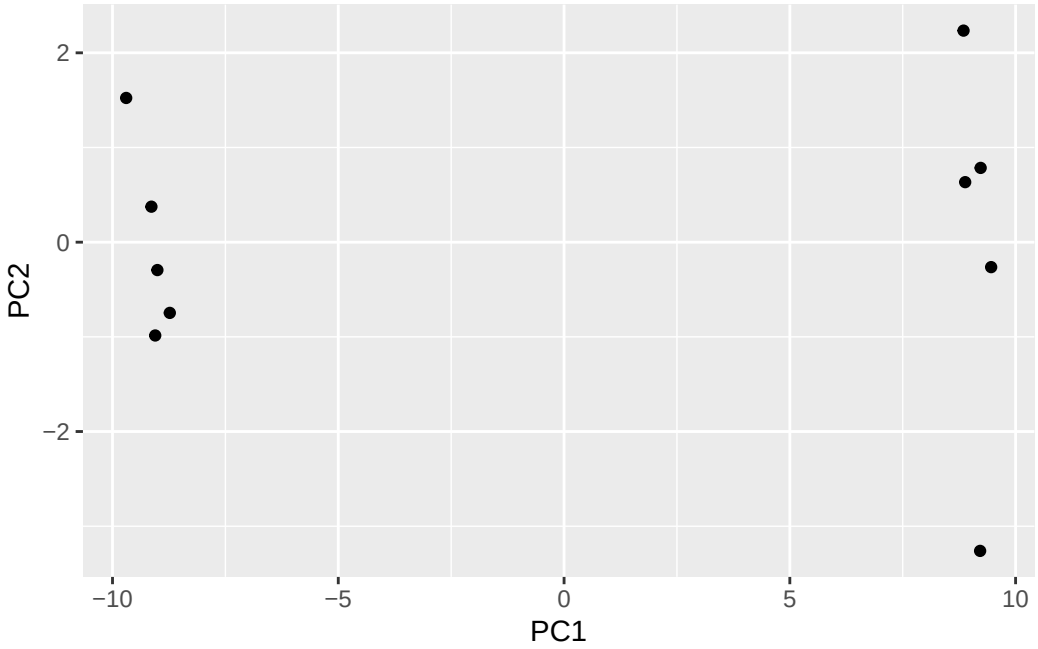



ggplot PCA

```
#Load in package
library(ggplot2)
```

```
Warning: package 'ggplot2' was built under R version 4.1.3
```

```
# Our pca
#pca$x

#ggplot requires data.frame input; convert pca variable to df
df <- as.data.frame(pca$x)

#Plot PCA w/ggplot
ggplot(df) +
  aes(PC1, PC2) +
  geom_point()
```




Adding more variables to ggplot PCA

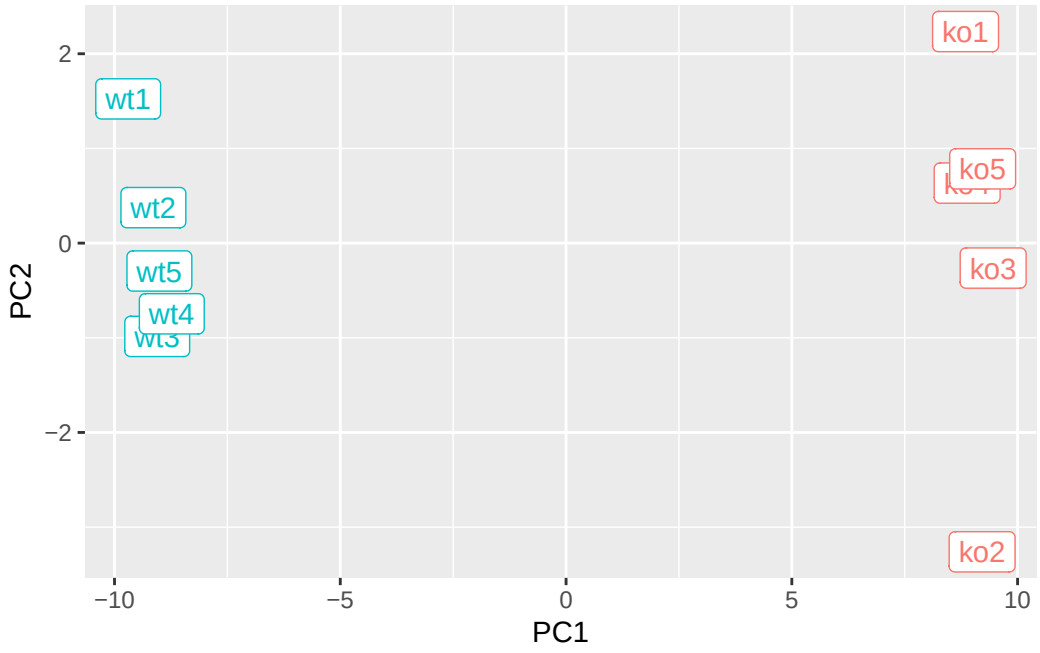
```
#Inspect rna.data
head(rna.data)
```

```
      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
```

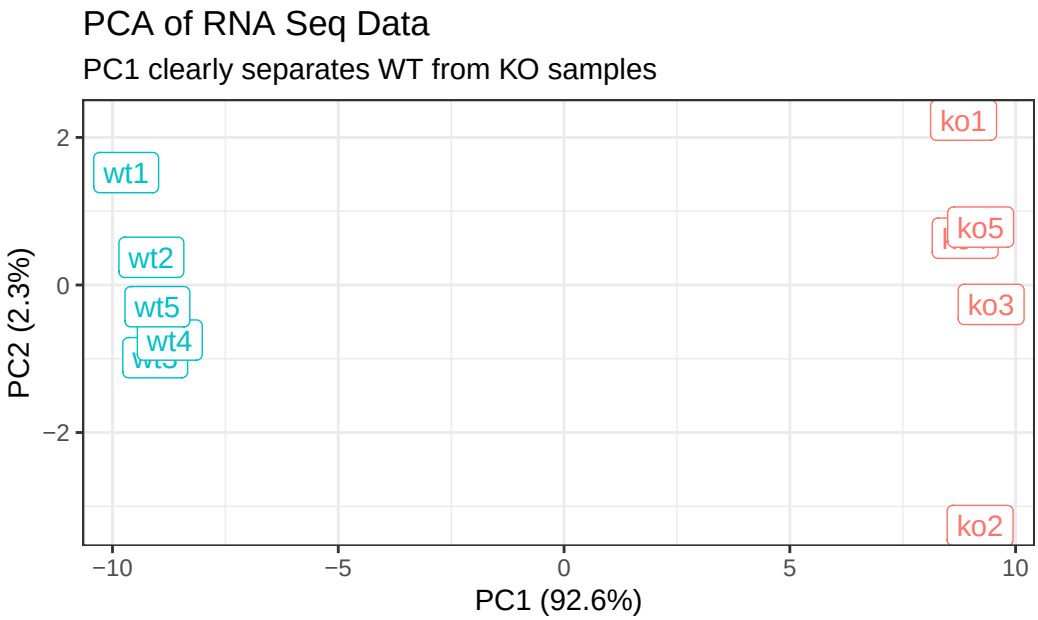
```
#If we want to add WT and KO conditions we need to modify the df
#Adds sample name
df$samples <- colnames(rna.data)

#Adds WT/KO component
df$condition <- substr(colnames(rna.data),1,2)
```

Finalized ggplot

```
p <- ggplot(df) +
      aes(PC1, PC2, label=samples, col=condition) +
      geom_label(show.legend = FALSE)
p
```




```
p + labs(title = "PCA of RNA Seq Data",
         subtitle = "PC1 clearly separates WT from KO samples",
         x=paste0("PC1 (", pca.var.per[1], "%)"),
         y=paste0("PC2 (", pca.var.per[2], "%)"),
         caption = "Class example data") +
         theme_bw()
```




### Q3 Optional: Top 10 PC1 genes

```
#pca$rotation = the relative expression
loading_scores <- pca$rotation[,1]

#Top gene scores of PC1
#Takes the absolute value of pca$rotation
gene_scores <- abs(loading_scores)
head(gene_scores)
```

```
     gene1      gene2      gene3      gene4      gene5      gene6
0.10366601 0.10351475 0.10376138 0.07532086 0.08742833 0.09967083
```

```
#Sorts gene ranks by greatest to smallest expression
gene_scores_ranked <- sort(gene_scores, decreasing = TRUE)
head(gene_scores_ranked)
```

```
  gene100    gene66    gene45    gene68    gene98    gene60
0.1038708 0.1038455 0.1038402 0.1038395 0.1038372 0.1038055
```

```r
#Top 10 genes by NAME
top_10_genes <- names(gene_scores_ranked[1:10])
top_10_genes
```

```
 [1] "gene100" "gene66"  "gene45"  "gene68"  "gene98"  "gene60"  "gene21"
 [8] "gene56"  "gene10"  "gene90"
```

## Answers to lab questions

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

17 rows, 5 columns dim(x)

> 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?

Piping in the row.names = 1 when the initial data is loaded in is preferable as the minus indexing removes columns for every subsequent run and is therefore more prone to human error.

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

Changing the beside argument to false creates stacked bar plots. barplot(as.matrix(x), beside=F, col=rainbow(nrow(x)))

> Q4: Missing

> 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?

Pairwise plots were difficult to interpret. It would be easier to interpret with fewer variables. Points that lie on the diagonal with the 2 countries being compared represent similarity between the two values. The more linear the correlation looks, the more similar the variable are between the 2 countries.

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

The main differences are the heightened consumption of fresh potatoes in Northern Ireland and the reduced consumption of fresh fruit relative to the other contries in the UK.

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.

text(pca$x[,1], pca$x[,2], colnames(x), col = c("orange", "red", "blue", "green"))

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

PC2 features fresh potatoes and soft drinks which contribute to the most to the vertical component of the variation across the 17 dimensions of 4 countries. par(mar=c(10, 3, 0.35, 0)) barplot(pca$rotation[,2], las=2)

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

100 genes and 10 samples of differing conditions. dim(rna.data)