Class 7: Machine Learning 1

Julia Di Silvestri (PID: A16950824)

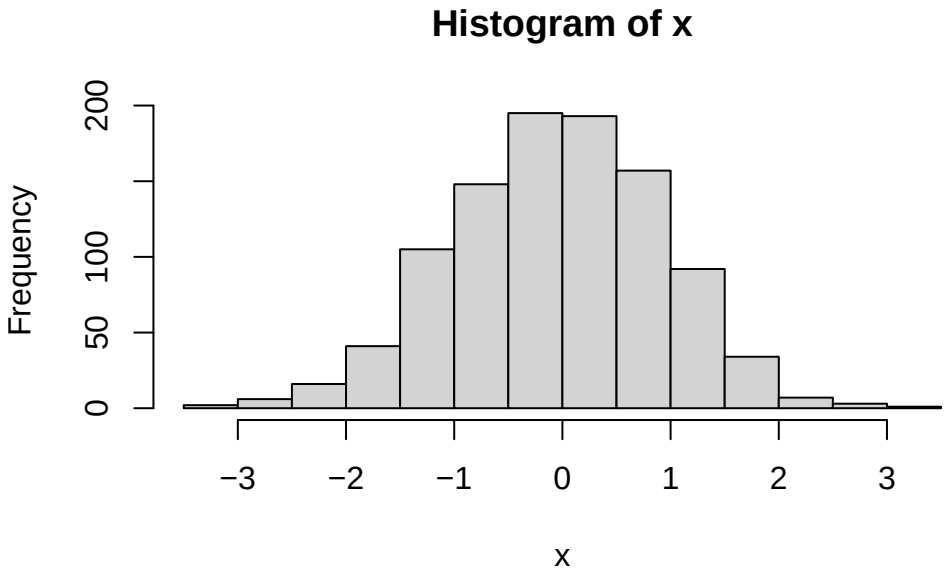
Clustering Methods

The broad goal here is to find groupings (clusters) in your input drug data.

Kmeans

First, we will make up some data where we know what the answer should be.

```
x <- rnorm(1000)
hist(x)
```



rnorm works as follows : rnorm(n (number of points), mean (default = 0), sd (default = 1))

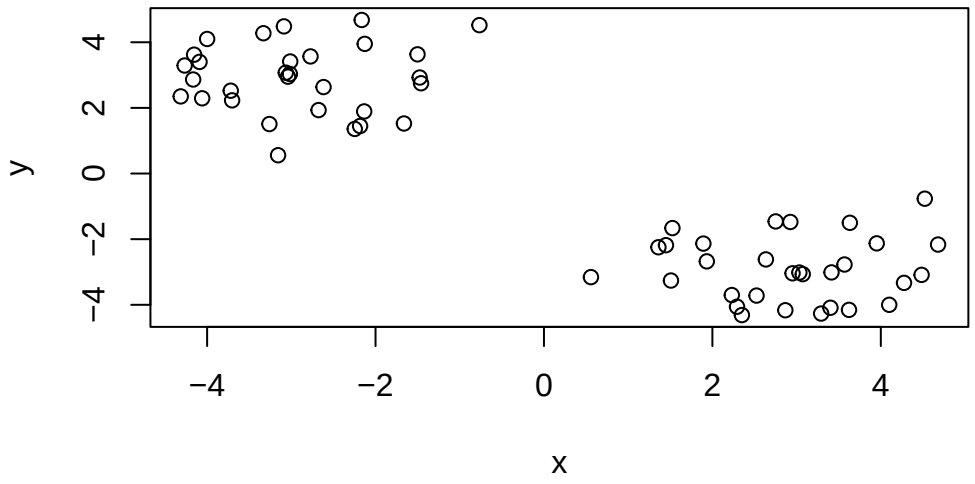
Make a vector of length 60 with 30 point centered at -3 and 30 points centered at +3

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

```
 [1] -4.0588947 -2.1835482 -3.0660046 -3.7031655 -4.1657538 -3.3316277
 [7] -4.2668944 -3.0405042 -1.4594948 -3.1570987 -1.5015703 -0.7664480
[13] -2.6766644 -4.3133928 -2.1347620 -1.4758290 -3.7193422 -2.1643045
[19] -4.0902880 -3.0172761 -3.2604102 -2.1282814 -2.7723140 -2.2468725
[25] -3.9998710 -3.0132188 -4.1545733 -1.6626402 -2.6175365 -3.0880333
[31]  4.4842659  2.6351031  1.5252990  3.6221032  3.4145954  4.0997977
[37]  1.3590088  3.5685645  3.9513526  1.5078974  3.0330362  3.4018658
[43]  4.6798805  2.5241284  2.9255952  1.8927843  2.3502026  1.9333294
[49]  4.5223389  3.6324620  0.5579344  2.7513980  2.9519584  3.2917626
[55]  4.2755490  2.8661514  2.2304137  3.0728262  1.4486832  2.2925299
```

I will make a small x and y dataset with 2 groups of points

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



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

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

Cluster means:
          x         y
1  2.893427 -2.907887
2 -2.907887  2.893427

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] 59.83682 59.83682
 (between_SS / total_SS =  89.4 %)

Available components:

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

Q. From your result object `k` how many points are in each cluster?

```
k$size
```

```
[1] 30 30
```

Q. What "componenet" of your result object details the cluster membership?

```
k$cluster
```

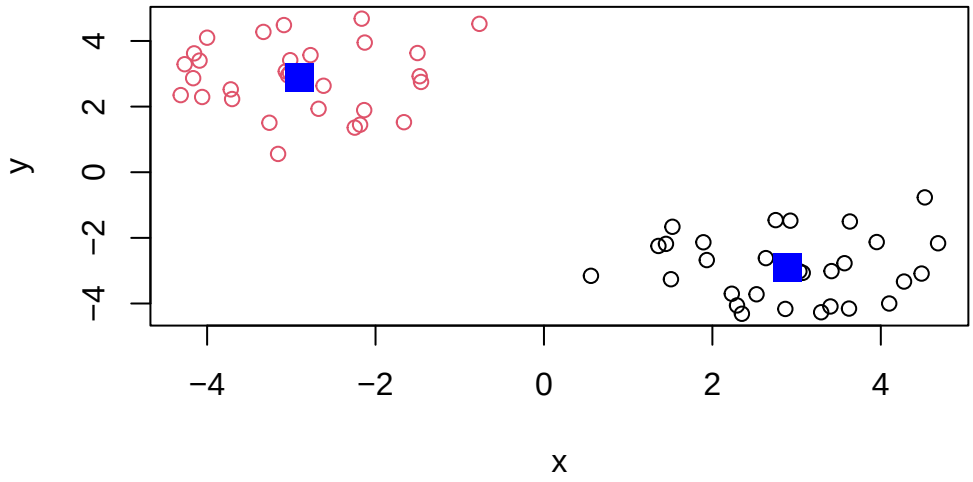
```
 [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1
[39] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Q. Cluster centers?

```
k$centers
```

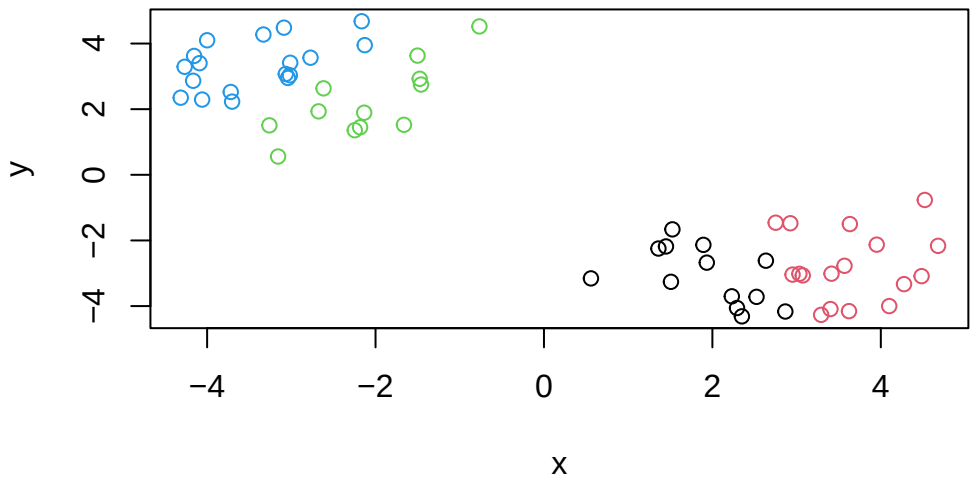
```
          x         y
1  2.893427 -2.907887
2 -2.907887  2.893427
```

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



We can cluster into 4 groups

```r
# kmeans
k4 <- kmeans(x, centers = 4)
# plot results
plot(x, col = k4$cluster)
```



A big limitation of kmeans is that it does what you ask even if it doesn't make sense for the data.

Hierarchical Clustering

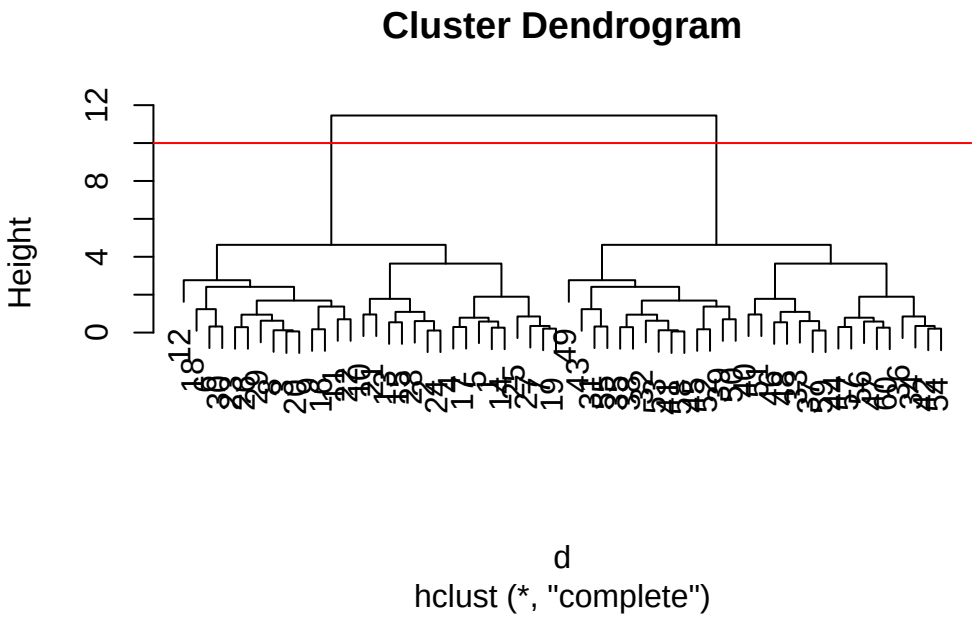
The main base R function for HClustering is `hclust()`. Unlike `kmeans()`, you cannot just pass it your data as input. You first need to calculate a distance matrix.

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

```
Call:
hclust(d = d)

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

Use `plot()` to view results

```
plot(hc)
abline(h = 10, col = "red")
```



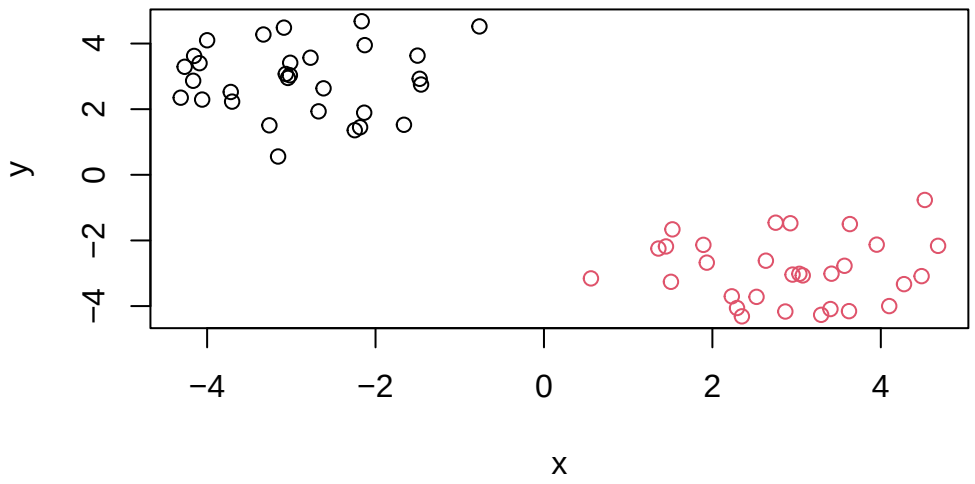
To make the "cut" at the line set and get our cluster membership vector, we can use the `cutree()` function.

```
grps <- cutree(hc, h=10)
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 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
```

Make a plot of our data colored by hclust results

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



Principal Component Analysis (PCA) –> LAB HANDOUT

Here we will do a PCA on some food data from the UK.

```
url <- "https://tinyurl.com/UK-foods"
#reading csv, but setting the first column to be rownames, not data
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. Complete the following code to find out how many rows and columns are in x? ____(x)

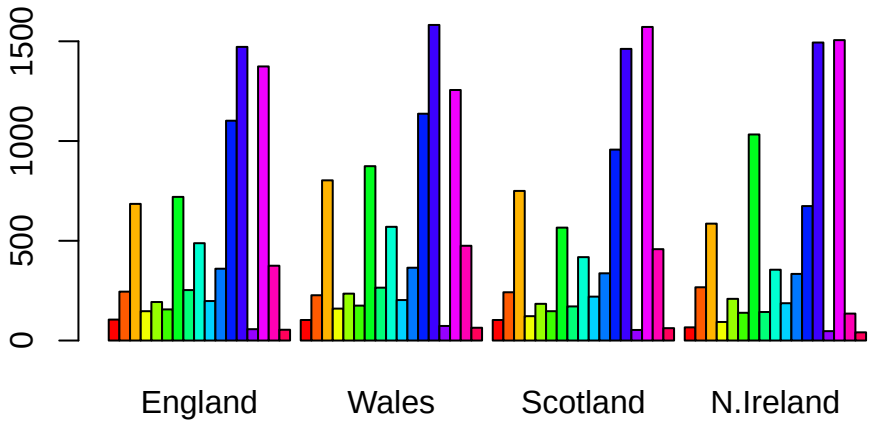
```
dim(x)
```

```
[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?

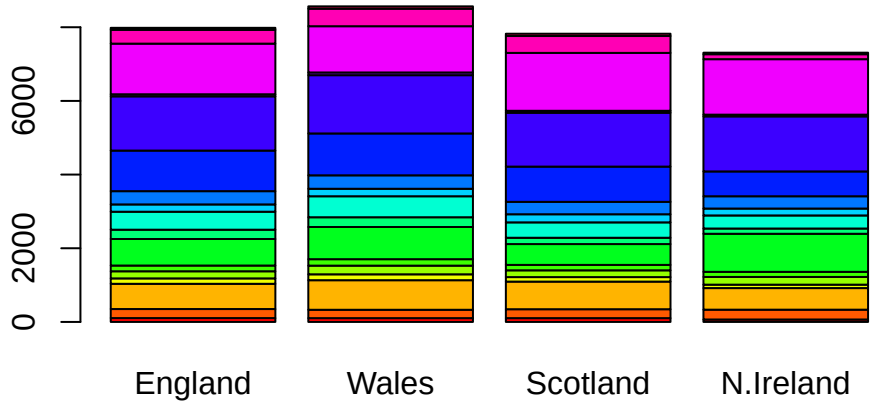
Using the approach `read.csv(row.names = 1)` is best because using the [, -1] method will continue to remove the first row each time it is run (rendered)

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



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

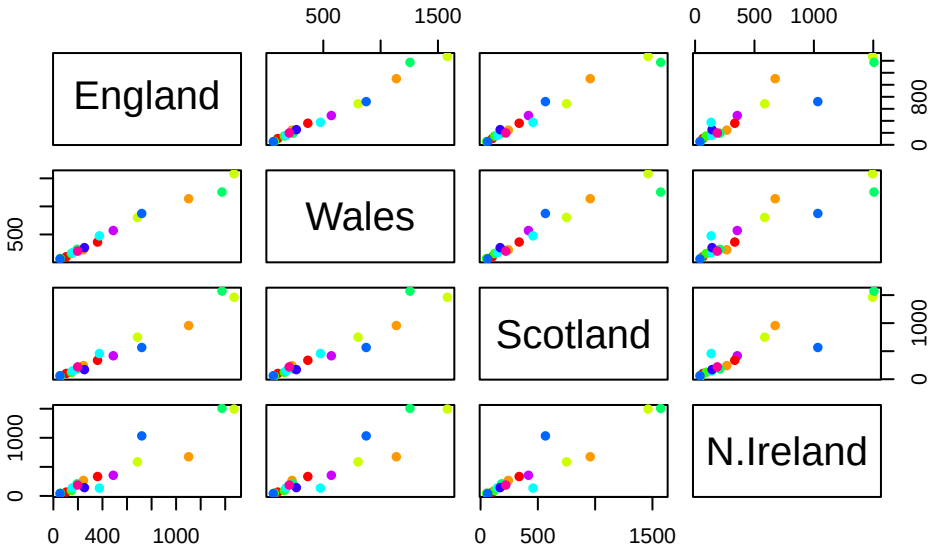
```
barplot(as.matrix(x), beside=F, col=rainbow(nrow(x)))
```




```
# changed `beside = F`
```

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?

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



If a point lies on a diagonal, there is not a significant difference between that data point when comparing the two places for the given plot.

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

PCA to the rescue

The main "base" R function for PCA is called `prcomp()`

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

```
Importance of components:
                            PC1      PC2      PC3        PC4
Standard deviation     324.1502 212.7478 73.87622 2.921e-14
Proportion of Variance   0.6744   0.2905  0.03503 0.000e+00
Cumulative Proportion    0.6744   0.9650  1.00000 1.000e+00
```

> Q. How much variance is captured in 2 PCs?

96.5%

To make out main "PC score plot" (a.k.a "PC1 vs PC2 plot" or "PC plot" or "ordination plot")

```
attributes(pca)
```

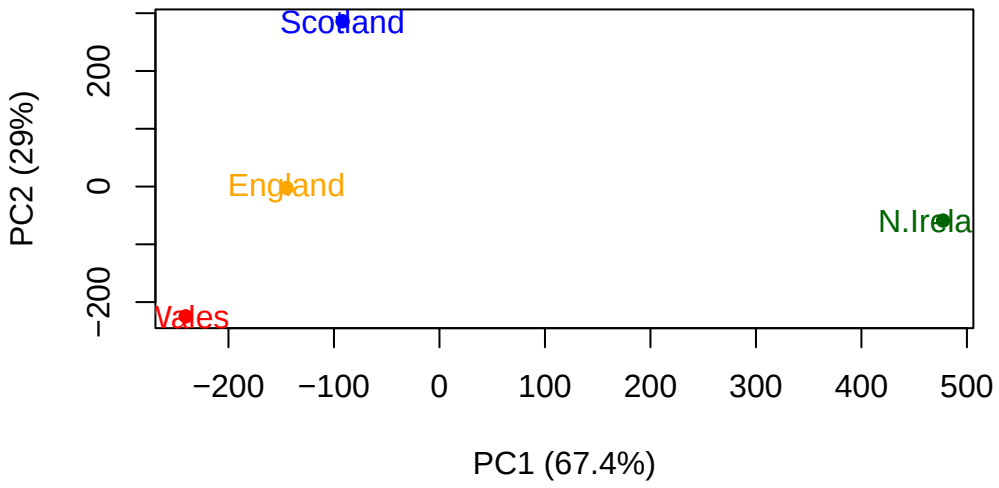
```
$names
[1] "sdev"     "rotation" "center"   "scale"    "x"

$class
[1] "prcomp"
```

We are after the `pca$x` result component to make our main PCA plot.

```
pca$x
```

```
                PC1         PC2        PC3           PC4
England   -144.99315   -2.532999 105.768945 -9.152022e-15
Wales     -240.52915 -224.646925 -56.475555  5.560040e-13
Scotland   -91.86934  286.081786 -44.415495 -6.638419e-13
N.Ireland  477.39164  -58.901862  -4.877895  1.329771e-13
```

```
mycols <- c("orange", "red", "blue", "darkgreen")
plot(pca$x[,1], pca$x[,2], col = mycols, pch = 16, xlab = "PC1 (67.4%)", ylab = "PC2 (29%)
text(pca$x[,1], pca$x[,2], colnames(x), col = mycols)
```



Another important result from PCA is how the original variable (in this case the foods) contribute to the PCs.

What is it that makes N.Ireland so different from the other countries in PC1? This is contained in the `pca$rotation` object – folks often call this the "loadings" or "contributions" to the PCs.

```
pca$rotation
```

```
                       PC1          PC2         PC3          PC4
Cheese         -0.056955380  0.016012850  0.02394295 -0.409382587
Carcass_meat    0.047927628  0.013915823  0.06367111  0.729481922
Other_meat     -0.258916658 -0.015331138 -0.55384854  0.331001134
Fish           -0.084414983 -0.050754947  0.03906481  0.022375878
Fats_and_oils  -0.005193623 -0.095388656 -0.12522257  0.034512161
Sugars         -0.037620983 -0.043021699 -0.03605745  0.024943337
Fresh_potatoes  0.401402060 -0.715017078 -0.20668248  0.021396007
Fresh_Veg      -0.151849942 -0.144900268  0.21382237  0.001606882
Other_Veg      -0.243593729 -0.225450923 -0.05332841  0.031153231
Processed_potatoes -0.026886233  0.042850761 -0.07364902 -0.017379680
Processed_Veg  -0.036488269 -0.045451802  0.05289191  0.021250980
Fresh_fruit    -0.632640898 -0.177740743  0.40012865  0.227657348
```

```
Cereals             -0.047702858 -0.212599678 -0.35884921  0.100043319
Beverages           -0.026187756 -0.030560542 -0.04135860 -0.018382072
Soft_drinks          0.232244140  0.555124311 -0.16942648  0.222319484
Alcoholic_drinks    -0.463968168  0.113536523 -0.49858320 -0.273126013
Confectionery       -0.029650201  0.005949921 -0.05232164  0.001890737
```

Plotting the rotation allows us to see what is making N.Ireland so different from the others.

```
library(ggplot2)

ld <- as.data.frame(pca$rotation)
ld_lab <- tibble::rownames_to_column(ld, "Food")

ggplot(ld_lab) +
  aes(PC1, reorder(Food, PC1), bg=PC1) +
  geom_col() +
  xlab("PC1 Loadings/Contributions") +
  ylab("Food Group") +
  scale_fill_gradient2(low="purple", mid="gray", high="darkgreen", guide=NULL) +
  theme_bw()
```