Class07

Hannah Kim

4/26/23

Importing Data

```
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 you use to answer this questions?

```
## Complete the following code to find out how many rows and columns are in x?
dim(x)
```

```
[1] 17  5
```

There are 17 rows and 5 columns in x.

Checking Your Data

```
## Preview the first 6 rows
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
```

The row names are incorrectly placed in the first column of our 'x' data frame.

```
# Note how the minus indexing works
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 We could solve this problem like this
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
```

We can check the dimensions again.

```
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?

An example when we select the wrong number

Example of hierarchical clustering

Let's Use the same data as before, which we stored in 'x'. We will use the 'hclust()' function.

```
dist(x)
```

	Cheese	Carcass_meat	Other_meat	Fish
Carcass_meat	307.73040			
Other_meat	1231.01950	940.85121		
Fish	78.12170	242.42318	1157.72449	
Fats_and_oils	228.42504	97.44742	1013.49544	158.24348
Sugars	122.67844	189.82624	1112.49404	55.19964
Fresh_potatoes	1456.75118	1155.83995	489.82752	1385.22958
Fresh_Veg	242.28289	148.07093	1003.77189	164.80898
Other_Veg	739.94865	464.13145	506.63892	663.80795
Processed_potatoes	216.74640	98.33107	1018.46453	151.29441
Processed_Veg	510.14606	213.96962	729.06927	437.83559
Fresh_fruit	1778.24211	1496.47018	579.68785	1703.33203
Cereals	2818.19002	2517.03854	1599.25545	2745.19762
Beverages	77.87811	378.39265	1305.22565	150.15326
Soft_drinks	2677.93409	2374.34265	1483.91846	2608.79551
Alcoholic_drinks	584.86751	377.47053	701.67585	515.62486
Confectionery	80.17481	382.78715	1309.44683	155.46382

	Fats_and_oils	Sugars	Fresh_potatoes	Fresh_Veg
Carcass_meat				
Other_meat				
Fish				
Fats_and_oils				
Sugars	106.00943			
Fresh_potatoes	1229.20706	1334.72619		
Fresh_Veg	95.00000	134.53996	1239.79635	
Other_Veg	524.71135	621.56737	792.35598	504.18548
Processed_potatoes	53.18834	100.90094	1248.26159	105.85840
Processed_Veg	289.52202	389.66781	964.22145	292.37989
Fresh_fruit	1566.40959	1662.13267	704.87942	1543.03014
Cereals	2595.14142	2697.54611	1442.95010	2590.81686
Beverages	296.89224	193.66208	1522.01675	313.71962
Soft_drinks	2458.85237	2559.57008	1345.14126	2461.56292
Alcoholic_drinks	413.85505	484.45640	1047.04059	376.05452
Confectionery	302.77054	198.88187	1528.96566	319.82339

	Other_Veg	Processed_potatoes	Processed_Veg	Fresh_fruit
Carcass_meat				

```
Other_meat
Fish
Fats_and_oils
Sugars
Fresh_potatoes
Fresh_Veg
Other_Veg
Processed_potatoes    534.99252
Processed_Veg         255.75574              296.28702
Fresh_fruit          1044.39791             1571.59473       1283.17887
Cereals              2092.76301             2602.98483       2308.43194    1123.54350
Beverages             812.93235              290.29296        583.17922    1853.23123
Soft_drinks          1977.90521             2461.59501       2172.76000    1076.38004
Alcoholic_drinks      267.94402              405.78443        258.00581    1227.31210
Confectionery         818.36667              293.83158        587.97194    1857.78013
                        Cereals   Beverages Soft_drinks  Alcoholic_drinks
Carcass_meat
Other_meat
Fish
Fats_and_oils
Sugars
Fresh_potatoes
Fresh_Veg
Other_Veg
Processed_potatoes
Processed_Veg
Fresh_fruit
Cereals
Beverages            2891.08907
Soft_drinks           357.94413 2751.36693
Alcoholic_drinks     2298.64199  659.16386   2174.53420
Confectionery        2895.90349   14.38749   2754.92087         661.52400
```

```r
clustering <- hclust(dist(x))
clustering
```

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

Cluster method   : complete
```

```
Distance         : euclidean
Number of objects: 17
```