Class 7: Machine Learning 1

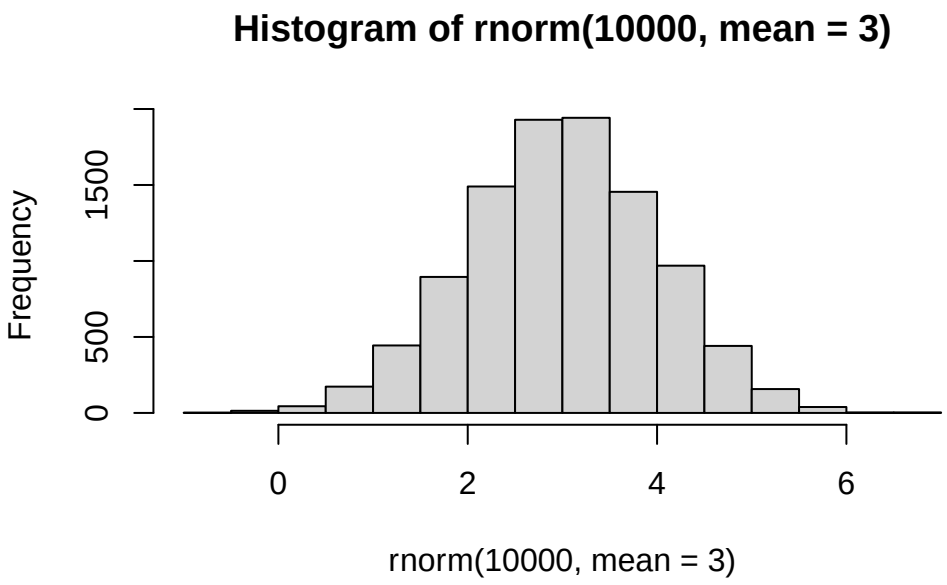
Jason Hsiao (PID: A15871650)

Clustering

We will start with k-means clustering, one of the most prevalent of all clustering methods.

To get started let's make up some data:

```
hist( rnorm(10000, mean = 3) )
```



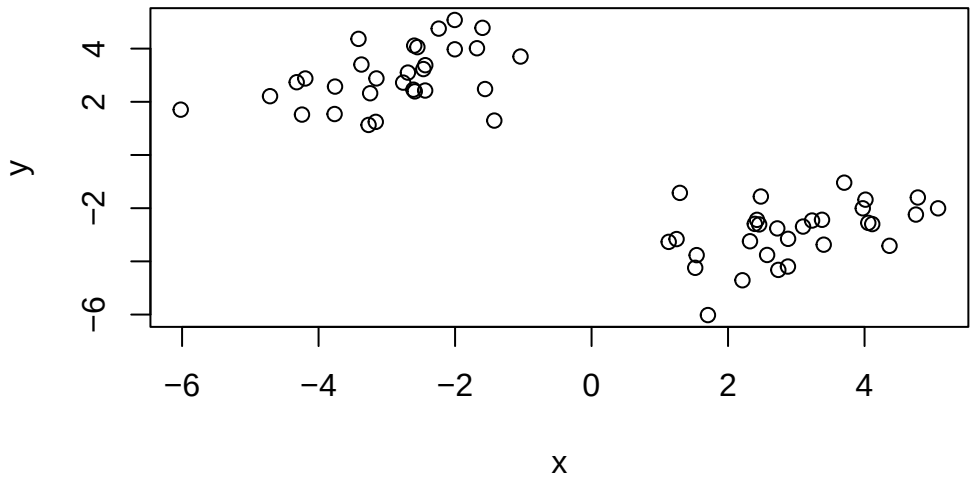

```
tmp <- c( rnorm(30,3), rnorm(30, -3))
tmp
```

```
 [1]  2.879270  1.539834  5.077270  2.210861  3.376840  4.365716  4.054800
 [8]  1.130313  3.231119  2.455928  2.572094  4.013586  2.480226  3.401627
[15]  3.098684  4.780768  3.702022  2.877222  2.321431  1.293929  2.721148
[22]  2.391724  4.752520  1.705171  4.112709  2.424112  1.245857  1.518727
[29]  2.736345  3.973897 -2.003955 -4.320436 -4.243024 -3.162015 -2.438352
[36] -2.598156 -6.019735 -2.239794 -2.590143 -2.760598 -1.425068 -3.244289
[43] -4.194173 -1.040783 -1.598070 -2.691046 -3.372093 -1.560717 -1.679061
[50] -3.758887 -2.610714 -2.464274 -3.267200 -2.552296 -3.414925 -2.436652
[57] -4.711054 -2.005091 -3.765200 -3.152162
```

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

```
           x         y
[1,] 2.879270 -3.152162
[2,] 1.539834 -3.765200
[3,] 5.077270 -2.005091
[4,] 2.210861 -4.711054
[5,] 3.376840 -2.436652
[6,] 4.365716 -3.414925
```

```r
plot(x)
```



The main function in R for K-means clustering is called `kmeans()`.

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

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

Cluster means:
          x         y
1 -2.910665  2.948192
2  2.948192 -2.910665

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] 70.31432 70.31432
 (between_SS / total_SS =  88.0 %)

Available components:

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

Q1. How many points are in each cluster?

```
k$size
```

```
[1] 30 30
```

Q2. The clustering result i.e. membership vector?

```
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
```

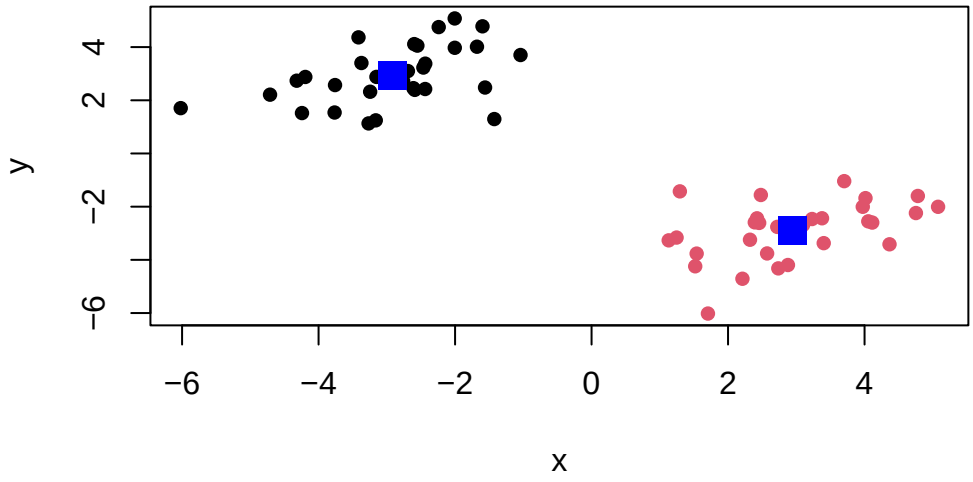
Q3. Cluster centers

```
k$centers
```

```
          x         y
1 -2.910665  2.948192
2  2.948192 -2.910665
```

Q4. Make a plot of our data colored by clustering results with optionally the cluster centers shown.

```
plot(x, col = k$cluster, pch = 16)
points(k$centers, col = "blue", pch = 15, cex = 2)
```



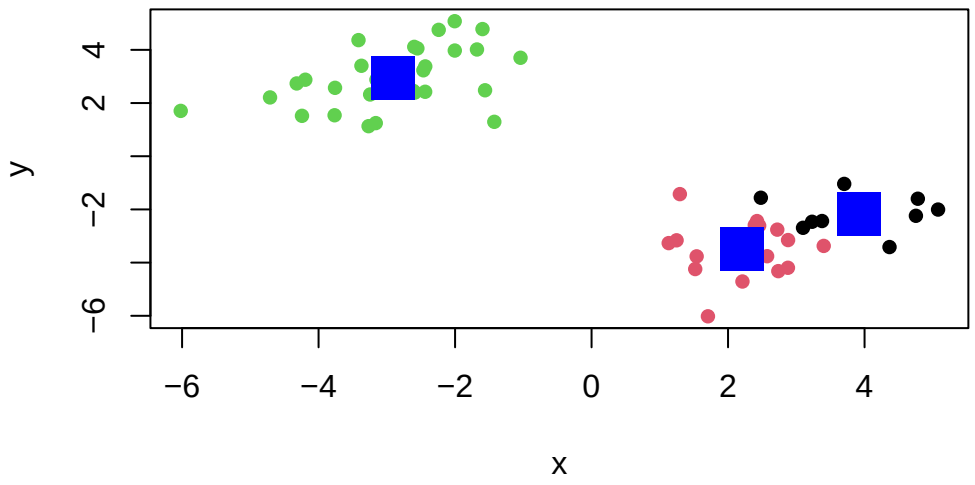
Q5. Run kmeans again but cluster into 3 groups and plot the results like we did above.

```
k <- kmeans(x, centers = 3, nstart = 20)
k
```

```
K-means clustering with 3 clusters of sizes 13, 17, 30

Cluster means:
          x         y
1  3.924628 -2.175755
2  2.201505 -3.472655
```

```
3 -2.910665  2.948192

Clustering vector:
 [1] 2 2 1 2 1 1 1 2 1 2 2 1 1 2 1 1 1 2 2 2 2 2 1 2 1 2 2 2 2 1 3 3 3 3 3 3 3 3
[39] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3

Within cluster sum of squares by cluster:
[1] 11.18543 24.86579 70.31432
 (between_SS / total_SS =  90.9 %)

Available components:

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

```r
plot(x, col = k$cluster, pch = 16)
points(k$centers, col = "blue", pch = 15, cex = 3)
```



K-means will always return a clustering result - even if there is no clear groupings.

Hierarchical Clustering (bottom up)

Hierarchical clustering has an advantage in that it can reveal the structure in your data rather than imposing a structure as k-means will.

The main function for this in base R is called `hclust()`

It requires a distance matrix as input, not the raw data itself.

```
             x         y
 [1,]  2.879270 -3.152162
 [2,]  1.539834 -3.765200
 [3,]  5.077270 -2.005091
 [4,]  2.210861 -4.711054
 [5,]  3.376840 -2.436652
 [6,]  4.365716 -3.414925
 [7,]  4.054800 -2.552296
 [8,]  1.130313 -3.267200
 [9,]  3.231119 -2.464274
[10,]  2.455928 -2.610714
[11,]  2.572094 -3.758887
[12,]  4.013586 -1.679061
[13,]  2.480226 -1.560717
[14,]  3.401627 -3.372093
[15,]  3.098684 -2.691046
[16,]  4.780768 -1.598070
[17,]  3.702022 -1.040783
[18,]  2.877222 -4.194173
[19,]  2.321431 -3.244289
[20,]  1.293929 -1.425068
[21,]  2.721148 -2.760598
[22,]  2.391724 -2.590143
[23,]  4.752520 -2.239794
[24,]  1.705171 -6.019735
[25,]  4.112709 -2.598156
[26,]  2.424112 -2.438352
[27,]  1.245857 -3.162015
[28,]  1.518727 -4.243024
[29,]  2.736345 -4.320436
[30,]  3.973897 -2.003955
[31,] -2.003955  3.973897
[32,] -4.320436  2.736345
[33,] -4.243024  1.518727
```

```
[34,] -3.162015  1.245857
[35,] -2.438352  2.424112
[36,] -2.598156  4.112709
[37,] -6.019735  1.705171
[38,] -2.239794  4.752520
[39,] -2.590143  2.391724
[40,] -2.760598  2.721148
[41,] -1.425068  1.293929
[42,] -3.244289  2.321431
[43,] -4.194173  2.877222
[44,] -1.040783  3.702022
[45,] -1.598070  4.780768
[46,] -2.691046  3.098684
[47,] -3.372093  3.401627
[48,] -1.560717  2.480226
[49,] -1.679061  4.013586
[50,] -3.758887  2.572094
[51,] -2.610714  2.455928
[52,] -2.464274  3.231119
[53,] -3.267200  1.130313
[54,] -2.552296  4.054800
[55,] -3.414925  4.365716
[56,] -2.436652  3.376840
[57,] -4.711054  2.210861
[58,] -2.005091  5.077270
[59,] -3.765200  1.539834
[60,] -3.152162  2.879270
```

	1	2	3	4	5	6
2	1.47306003					
3	2.47930953	3.95113158				
4	1.69614716	1.15970528	3.94189536			
5	0.87151031	2.26707570	1.75433961	2.55585828		
6	1.50949194	2.84750829	1.57922215	2.51462740	1.39100489	
7	1.31973826	2.79216599	1.15968917	2.83907496	0.68775230	0.91695034
8	1.75273713	0.64475695	4.14383832	1.80341272	2.39514033	3.23877446
9	0.77264920	2.13374142	1.90239981	2.46757873	0.14831637	1.48021932
10	0.68730212	1.47379330	2.69039278	2.11458852	0.93721735	2.07220790
11	0.68005364	1.03227962	3.05805649	1.01838620	1.54787679	1.82630497
12	1.85921990	3.23595817	1.11252915	3.52743529	0.98964106	1.77122074
13	1.64071104	2.39668150	2.63478773	3.16183148	1.25346664	2.64445855
14	0.56676749	1.90284127	2.16251623	1.79185369	0.93576904	0.96504083

15	0.51065674	1.89309835	2.09411966	2.20650441	0.37694438	1.45923651
16	2.45578846	3.89873181	0.50356685	4.03671827	1.63530862	1.86366107
17	2.26601922	3.47814724	1.67964228	3.96162248	1.43324619	2.46516613
18	1.04201327	1.40450167	3.10359373	0.84332843	1.82715603	1.68013162
19	0.56539568	0.93927761	3.02163241	1.47092672	1.32897179	2.05139454
20	2.34438961	2.35301667	3.82754523	3.41151981	2.31556124	3.65997445
21	0.42228596	1.55071881	2.47428848	2.01610328	0.73135029	1.76995770
22	0.74402104	1.45136999	2.74853561	2.12860841	0.99700280	2.13937224
23	2.08362142	3.55643271	0.40068502	3.54501845	1.38969304	1.23715452
24	3.09862633	2.26058926	5.24294033	1.40298571	3.95385414	3.72337693
25	1.35214336	2.82518575	1.13230130	2.84277335	0.75338281	0.85505829
26	0.84657749	1.59451304	2.68830173	2.28268424	0.95273005	2.17336673
27	1.63344280	0.67101038	4.00227412	1.82503536	2.25105289	3.13009324
28	1.74386315	0.47829022	4.20375734	0.83552411	2.59144077	2.96497806
29	1.17698411	1.31906226	3.29253009	0.65476400	1.98969241	1.86408163
30	1.58637532	3.00443804	1.10337356	3.23058512	0.73736261	1.46436290
31	8.63866988	8.51187834	9.26779695	9.65365493	8.36947471	9.75537909
32	9.30109053	8.75287666	10.52606752	9.90563397	9.27404759	10.64365386
33	8.51729354	7.83334779	9.96419533	8.97010619	8.58529922	9.92226509
34	7.47259660	6.87154108	8.85745351	8.02200607	7.50449863	8.85379184
35	7.70531915	7.35755046	8.72367034	8.51620646	7.57914854	8.96603081
36	9.09838165	8.89856214	9.81527620	10.04915063	8.86536574	10.25479384
37	10.13834199	9.33123992	11.70083628	10.43602624	10.26890076	11.57898886
38	9.41747367	9.31864447	9.96015707	10.45789409	9.12308984	10.50428118
39	7.78775648	7.41379919	8.83862018	8.57316069	7.67581276	9.06096862
40	8.14271950	7.78244316	9.15256840	8.94164532	8.01692238	9.40402820
41	6.18829950	5.86390876	7.29136027	7.01995665	6.08075285	7.46367800
42	8.21329396	7.74176386	9.37907986	8.90025276	8.15344740	9.52984589
43	9.29446493	8.77499918	10.47838957	9.93007514	9.24972999	10.62369193
44	7.89598994	7.90056898	8.36670266	9.01959161	7.56298315	8.93762692
45	9.10922400	9.10384635	9.51882636	10.22754334	8.76589357	10.13588400
46	8.37266432	8.06307984	9.29490531	9.22066715	8.21335472	9.60338534
47	9.05713455	8.68852276	10.03116802	9.84809388	8.92376572	10.31208372
48	7.17197937	6.97271595	8.01130141	8.12030233	6.96815377	8.35915238
49	8.49272165	8.41847903	9.04834107	9.55252961	8.19559029	9.57716582
50	8.76540074	8.26061427	9.95129689	9.41711953	8.71815063	10.09225404
51	7.84796852	7.47860213	8.88852075	8.63789442	7.73227993	9.11797721
52	8.32464678	8.06110170	9.18111042	9.21601834	8.13893397	9.52988331
53	7.49124098	6.86102198	8.91408698	8.00816553	7.54098883	8.88372699
54	9.02453444	8.82598067	9.74333459	9.97636521	8.79167866	10.18114436
55	9.80486598	9.52163025	10.61624114	10.67882145	9.61250738	11.00348915
56	8.41943606	8.17442243	9.24252211	9.32809774	8.22151977	9.61250738
57	9.29381711	8.64794205	10.65765173	9.78906555	9.32809774	10.67882145

58	9.56977241	9.52658264	10.01597147	10.65765173	9.24252211	10.61624114	
59	8.13411426	7.50245108	9.52658264	8.64794205	8.17442243	9.52163025	
60	8.52973359	8.13411426	9.56977241	9.29381711	8.41943606	9.80486598	
	7	8	9	10	11	12	
2							
3							
4							
5							
6							
7							
8	3.01060055						
9	0.82837126	2.24901689					
10	1.59993865	1.47926715	0.78890100				
11	1.91161688	1.52331568	1.45269976	1.15403477			
12	0.87420778	3.29172464	1.10851921	1.81501362	2.53052899		
13	1.86078269	2.17585645	1.17484251	1.05027806	2.20008907	1.53791942	
14	1.04818995	2.27373467	0.92369223	1.21410140	0.91527801	1.80023663	
15	0.96613094	2.05096103	0.26261066	0.64775648	1.19062259	1.36424302	
16	1.19899039	4.01395371	1.77530957	2.53580948	3.08988254	0.77144603	
17	1.55213590	3.40156225	1.49935931	2.00435394	2.94360844	0.71026088	
18	2.02050743	1.97761765	1.76572746	1.63854536	0.53157946	2.75991193	
19	1.86639295	1.19133869	1.19831326	0.64769308	0.57240184	2.30506546	
20	2.98212276	1.84938407	2.19833024	1.66012067	2.66090599	2.73149173	
21	1.34982144	1.66955183	0.58981169	0.30464143	1.00935603	1.68526482	
22	1.66350728	1.43162986	0.84877994	0.06741988	1.18258066	1.86024405	
23	0.76450651	3.76509602	1.53787254	2.32635190	2.65742361	0.92760184	
24	4.18854272	2.81192264	3.86908499	3.49071050	2.42136081	4.91632319	
25	0.07386826	3.05651849	0.89169774	1.65682770	1.92893463	0.92442546	
26	1.63466448	1.53652370	0.80742322	0.17527368	1.32880072	1.76151958	
27	2.87435494	0.15625181	2.10430619	1.32973834	1.45435947	3.13997966	
28	3.04798750	1.05028495	2.46905588	1.88222808	1.15929754	3.57746122	
29	2.20559336	1.92058455	1.92097299	1.73256505	0.58507687	2.93397467	
30	0.55427705	3.11155307	0.87384985	1.63474347	2.24607182	0.32730989	
31	8.90503923	7.89031887	8.29795494	7.95284060	8.98533204	8.25631478	
32	9.90526656	8.10883555	9.16910127	8.63192646	9.47074523	9.43142197	
33	9.24267953	7.19568286	8.46918630	7.86945071	8.61969013	8.85423373	
34	8.15526748	6.22830334	7.39170019	6.81428121	7.61101014	7.74883113	
35	8.18081105	6.71761890	7.48593501	7.02164214	7.95826967	7.64614471	
36	9.41722469	8.26828490	8.78846698	8.41119412	9.41772344	8.78975096	
37	10.93719728	8.70905581	10.14704719	9.51124241	10.18211475	10.58869885	
38	9.64273038	8.69905309	9.05610283	8.73309908	9.77743917	8.97050601	
39	8.28242738	6.77238578	7.58075222	7.10543599	8.02986360	7.75761019	
40	8.61735812	7.14139316	7.92396797	7.45928261	8.39216661	8.07783343	

41	6.69495343	5.22818001	5.98365808	5.50530365	6.44269008	6.19819473
42	8.77666894	7.09717806	8.05194870	7.53780684	8.41430826	8.28738038
43	9.87548647	8.13044130	9.14694185	8.62213992	9.47735858	9.38759914
44	8.06730837	7.29956961	7.50149011	7.21648314	8.28963520	7.38259374
45	9.25898367	8.49787454	8.70699185	8.43023830	9.50347260	8.55687259
46	8.80000147	7.42477411	8.12517980	7.68690962	8.64447444	8.23279626
47	9.51882012	8.04642189	8.83239464	8.37341460	9.30625165	8.96446468
48	7.54057819	6.34622356	6.88547587	6.48468326	7.48376045	6.95503554
49	8.71710762	7.80400034	8.12850112	7.80893606	8.85910013	8.05061748
50	9.34414755	7.61588065	8.61540453	8.09230745	8.95336019	8.85910013
51	8.33734954	6.83735942	7.63776194	7.16531472	8.09230745	7.80893606
52	8.71471298	7.42625113	8.05450212	7.63776194	8.61540453	8.12850112
53	8.19593191	6.21902237	7.42625113	6.83735942	7.61588065	7.80400034
54	9.34384550	8.19593191	8.71471298	8.33734954	9.34414755	8.71710762
55	10.18114436	8.88372699	9.52988331	9.11797721	10.09225404	9.57716582
56	8.79167866	7.54098883	8.13893397	7.73227993	8.71815063	8.19559029
57	9.97636521	8.00816553	9.21601834	8.63789442	9.41711953	9.55252961
58	9.74333459	8.91408698	9.18111042	8.88852075	9.95129689	9.04834107
59	8.82598067	6.86102198	8.06110170	7.47860213	8.26061427	8.41847903
60	9.02453444	7.49124098	8.32464678	7.84796852	8.76540074	8.49272165

	13	14	15	16	17	18
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14	2.03225485					
15	1.28846184	0.74538476				
16	2.30084542	2.24704023	2.00599200			
17	1.32782426	2.35058385	1.75709592	1.21419214		
18	2.66321165	0.97509812	1.51935404	3.21919892	3.25947351	
19	1.69104379	1.08772990	0.95404420	2.95945564	2.60028295	1.10053818
20	1.19402806	2.86937209	2.20450547	3.49112910	2.43856317	3.18979031
21	1.22382845	0.91486475	0.38388957	2.36505972	1.97986811	1.44204659
22	1.03322323	1.27724289	0.71412541	2.58684071	2.02913780	1.67589430
23	2.37159486	1.76267219	1.71429277	0.64234537	1.59410557	2.70856791

24	4.52587564	3.14451454	3.60860774	5.38613257	5.36445544	2.16941929
25	1.93423845	1.05100674	1.01826983	1.20269577	1.61061362	2.01834007
26	0.87942734	1.35181581	0.72034872	2.50198031	1.89374137	1.81334374
27	2.02183574	2.16598093	1.91174769	3.86542616	3.24536138	1.93046683
28	2.84943010	2.07456783	2.21470114	4.19960682	3.87571012	1.35937293
29	2.77157785	1.15842750	1.66919168	3.40454761	3.41886794	0.18917882
30	1.55804802	1.48300168	1.11269564	0.90320755	1.00080858	2.44943898
31	7.12319059	9.12052013	8.39395051	8.77948156	7.59639361	9.51542284
32	8.04448560	9.84597678	9.19238377	10.08062854	8.86715997	9.99191483
33	7.39493587	9.07528550	8.46303012	9.54689794	8.34714651	9.12880828
34	6.30172540	8.02538775	7.39564476	8.43657068	7.23489624	8.12811866
35	6.33018792	8.22808253	7.53814421	8.26399735	7.05051022	8.48864122
36	7.61431087	9.59268708	8.87384204	9.33067637	8.13945460	9.94907303
37	9.10578742	10.70236686	10.12285997	11.29434691	10.10211966	10.67511608
38	7.88261053	9.89115512	9.16002296	9.46669323	8.29864616	10.30665633
39	6.42887493	8.31401714	7.62871584	8.38145499	7.16752691	8.55956295
40	6.76761432	8.66605954	7.97640456	8.69067568	7.47780554	8.92225740
41	4.83738832	6.71332518	6.02862829	6.84660945	5.63364272	6.97344686
42	6.91672949	8.75125149	8.08444404	8.93107105	7.71723501	8.94013385
43	8.01516775	9.83616339	9.17558628	10.02884921	8.81479797	10.00046317
44	6.33197677	8.35332877	7.61620049	7.87282876	6.70733891	8.81479797
45	7.53969096	9.56379161	8.82539029	9.02103964	7.87282876	10.02884921
46	6.96075295	8.88772284	8.18791545	8.82539029	7.61620049	9.17558628
47	7.67297162	9.57948569	8.88772284	9.56379161	8.35332877	9.83616339
48	5.71475708	7.67297162	6.96075295	7.53969096	6.33197677	8.01516775
49	6.95503554	8.96446468	8.23279626	8.55687259	7.38259374	9.38759914
50	7.48376045	9.30625165	8.64447444	9.50347260	8.28963520	9.47735858
51	6.48468326	8.37341460	7.68690962	8.43023830	7.21648314	8.62213992
52	6.88547587	8.83239464	8.12517980	8.70699185	7.50149011	9.14694185
53	6.34622356	8.04642189	7.42477411	8.49787454	7.29956961	8.13044130
54	7.54057819	9.51882012	8.80000147	9.25898367	8.06730837	9.87548647
55	8.35915238	10.31208372	9.60338534	10.13588400	8.93762692	10.62369193
56	6.96815377	8.92376572	8.21335472	8.76589357	7.56298315	9.24972999
57	8.12030233	9.84809388	9.22066715	10.22754334	9.01959161	9.93007514
58	8.01130141	10.03116802	9.29490531	9.51882636	8.36670266	10.47838957
59	6.97271595	8.68852276	8.06307984	9.10384635	7.90056898	8.77499918
60	7.17197937	9.05713455	8.37266432	9.10922400	7.89598994	9.29446493

	19	20	21	22	23	24
2						
3						
4						
5						
6						

7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20	2.08933611						
21	0.62747974	1.95463408					
22	0.65791160	1.60079784	0.37091126				
23	2.63043769	3.55325633	2.09707122	2.38665097			
24	2.84304032	4.61303418	3.41382283	3.49763571	4.85533637		
25	1.90424852	3.05313889	1.40100967	1.72100365	0.73333622	4.18371110	
26	0.81245104	1.51791291	0.43826061	0.15520754	2.33685871	3.65283135	
27	1.07871569	1.73761165	1.52892700	1.28064292	3.62590284	2.89439706	
28	1.28132980	2.82690860	1.90877047	1.86926167	3.80399084	1.78646629	
29	1.15336299	3.23476739	1.55991233	1.76427809	2.89724562	1.98769658	
30	2.06617339	2.74177751	1.46351933	1.68727303	0.81355534	4.61233206	
31	8.41493784	6.32652085	8.22678697	7.89991268	9.17931983	10.65975161	
32	8.93769402	6.98845085	8.93310019	8.56881311	10.34796975	10.62905777	
33	8.11038806	6.27086735	8.17388009	7.80401782	9.74916875	9.60257443	
34	7.08728353	5.19511999	7.11781446	6.74973383	8.64809900	8.74518867	
35	7.40177704	5.36153959	7.31448259	6.96221128	8.57091925	9.40570743	
36	8.85029655	6.76869959	8.69122172	8.35626238	9.71528298	11.00840747	
37	9.69908232	7.95537981	9.81560592	9.44469979	11.47188797	10.92466632	
38	9.20617813	7.11686607	9.00321457	8.68133906	9.88862445	11.47188797	
39	7.47584069	5.44554036	7.39974490	7.04542329	8.68133906	9.44469979	
40	7.83667333	5.79916274	7.75235882	7.39974490	9.00321457	9.81560592	
41	5.88486796	3.84524179	5.79916274	5.44554036	7.11686607	7.95537981	
42	7.87111633	5.88486796	7.83667333	7.47584069	9.20617813	9.69908232	
43	8.94013385	6.97344686	8.92225740	8.55956295	10.30665633	10.67511608	
44	7.71723501	5.63364272	7.47780554	7.16752691	8.29864616	10.10211966	
45	8.93107105	6.84660945	8.69067568	8.38145499	9.46669323	11.29434691	
46	8.08444404	6.02862829	7.97640456	7.62871584	9.16002296	10.12285997	
47	8.75125149	6.71332518	8.66605954	8.31401714	9.89115512	10.70236686	
48	6.91672949	4.83738832	6.76761432	6.42887493	7.88261053	9.10578742	
49	8.28738038	6.19819473	8.07783343	7.75761019	8.97050601	10.58869885	

50	8.41430826	6.44269008	8.39216661	8.02986360	9.77743917	10.18211475
51	7.53780684	5.50530365	7.45928261	7.10543599	8.73309908	9.51124241
52	8.05194870	5.98365808	7.92396797	7.58075222	9.05610283	10.14704719
53	7.09717806	5.22818001	7.14139316	6.77238578	8.69905309	8.70905581
54	8.77666894	6.69495343	8.61735812	8.28242738	9.64273038	10.93719728
55	9.52984589	7.46367800	9.40402820	9.06096862	10.50428118	11.57898886
56	8.15344740	6.08075285	8.01692238	7.67581276	9.12308984	10.26890076
57	8.90025276	7.01995665	8.94164532	8.57316069	10.45789409	10.43602624
58	9.37907986	7.29136027	9.15256840	8.83862018	9.96015707	11.70083628
59	7.74176386	5.86390876	7.78244316	7.41379919	9.31864447	9.33123992
60	8.21329396	6.18829950	8.14271950	7.78775648	9.41747367	10.13834199

	25	26	27	28	29	30
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26	1.69614157					
27	2.92177542	1.38274015				
28	3.07153541	2.01904974	1.11491703			
29	2.20468218	1.90780698	1.88772162	1.22007600		
30	0.61019946	1.60951433	2.96366390	3.32284379	2.62633172	
31	8.97805482	7.79260724	7.84107923	8.94019551	9.55334558	8.45396048
32	9.97871879	8.50096562	8.11013352	9.09985828	9.97979510	9.55334558

33	9.31488069	7.75301128	7.21367912	8.14834717	9.09985828	8.94019551
34	8.22788178	6.69165230	6.23367236	7.21367912	8.11013352	7.84107923
35	8.25467007	6.87656292	6.69165230	7.75301128	8.50096562	7.79260724
36	9.49059618	8.25467007	8.22788178	9.31488069	9.97871879	8.97805482
37	11.00840747	9.40570743	8.74518867	9.60257443	10.62905777	10.65975161
38	9.71528298	8.57091925	8.64809900	9.74916875	10.34796975	9.17931983
39	8.35626238	6.96221128	6.74973383	7.80401782	8.56881311	7.89991268
40	8.69122172	7.31448259	7.11781446	8.17388009	8.93310019	8.22678697
41	6.76869959	5.36153959	5.19511999	6.27086735	6.98845085	6.32652085
42	8.85029655	7.40177704	7.08728353	8.11038806	8.93769402	8.41493784
43	9.94907303	8.48864122	8.12811866	9.12880828	9.99191483	9.51542284
44	8.13945460	7.05051022	7.23489624	8.34714651	8.86715997	7.59639361
45	9.33067637	8.26399735	8.43657068	9.54689794	10.08062854	8.77948156
46	8.87384204	7.53814421	7.39564476	8.46303012	9.19238377	8.39395051
47	9.59268708	8.22808253	8.02538775	9.07528550	9.84597678	9.12052013
48	7.61431087	6.33018792	6.30172540	7.39493587	8.04448560	7.12319059
49	8.78975096	7.64614471	7.74883113	8.85423373	9.43142197	8.25631478
50	9.41772344	7.95826967	7.61101014	8.61969013	9.47074523	8.98533204
51	8.41119412	7.02164214	6.81428121	7.86945071	8.63192646	7.95284060
52	8.78846698	7.48593501	7.39170019	8.46918630	9.16910127	8.29795494
53	8.26828490	6.71761890	6.22830334	7.19568286	8.10883555	7.89031887
54	9.41722469	8.18081105	8.15526748	9.24267953	9.90526656	8.90503923
55	10.25479384	8.96603081	8.85379184	9.92226509	10.64365386	9.75537909
56	8.86536574	7.57914854	7.50449863	8.58529922	9.27404759	8.36947471
57	10.04915063	8.51620646	8.02200607	8.97010619	9.90563397	9.65365493
58	9.81527620	8.72367034	8.85745351	9.96419533	10.52606752	9.26779695
59	8.89856214	7.35755046	6.87154108	7.83334779	8.75287666	8.51187834
60	9.09838165	7.70531915	7.47259660	8.51729354	9.30109053	8.63866988

	31	32	33	34	35	36
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						

2.62633172						
3.32284379	1.22007600					
2.96366390	1.88772162	1.11491703				
1.60951433	1.90780698	2.01904974	1.38274015			
0.61019946	2.20468218	3.07153541	2.92177542	1.69614157		
4.61233206	1.98769658	1.78646629	2.89439706	3.65283135	4.18371110	
0.81355534	2.89724562	3.80399084	3.62590284	2.33685871	0.73333622	
1.68727303	1.76427809	1.86926167	1.28064292	0.15520754	1.72100365	
1.46351933	1.55991233	1.90877047	1.52892700	0.43826061	1.40100967	
2.74177751	3.23476739	2.82690860	1.73761165	1.51791291	3.05313889	
2.06617339	1.15336299	1.28132980	1.07871569	0.81245104	1.90424852	
2.44943898	0.18917882	1.35937293	1.93046683	1.81334374	2.01834007	
1.00080858	3.41886794	3.87571012	3.24536138	1.89374137	1.61061362	
0.90320755	3.40454761	4.19960682	3.86542616	2.50198031	1.20269577	
1.11269564	1.66919168	2.21470114	1.91174769	0.72034872	1.01826983	
1.48300168	1.15842750	2.07456783	2.16598093	1.35181581	1.05100674	
1.55804802	2.77157785	2.84943010	2.02183574	0.87942734	1.93423845	
0.32730989	2.93397467	3.57746122	3.13997966	1.76151958	0.92442546	
2.24607182	0.58507687	1.15929754	1.45435947	1.32880072	1.92893463	
1.63474347	1.73256505	1.88222808	1.32973834	0.17527368	1.65682770	
0.87384985	1.92097299	2.46905588	2.10430619	0.80742322	0.89169774	
3.11155307	1.92058455	1.05028495	0.15625181	1.53652370	3.05651849	
0.55427705	2.20559336	3.04798750	2.87435494	1.63466448	0.07386826	
1.46436290	1.86408163	2.96497806	3.13009324	2.17336673	0.85505829	
0.73736261	1.98969241	2.59144077	2.25105289	0.95273005	0.75338281	
3.23058512	0.65476400	0.83552411	1.82503536	2.28268424	2.84277335	
1.10337356	3.29253009	4.20375734	4.00227412	2.68830173	1.13230130	

59	3.00443804	1.31906226	0.47829022	0.67101038	1.59451304	2.82518575
60	1.58637532	1.17698411	1.74386315	1.63344280	0.84657749	1.35214336
	37	38	39	40	41	42
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						
31						
32						
33						
34						
35						
36						
37						
38	4.85533637					
39	3.49763571	2.38665097				
40	3.41382283	2.09707122	0.37091126			
41	4.61303418	3.55325633	1.60079784	1.95463408		

42	2.84304032	2.63043769	0.65791160	0.62747974	2.08933611	
43	2.16941929	2.70856791	1.67589430	1.44204659	3.18979031	1.10053818
44	5.36445544	1.59410557	2.02913780	1.97986811	2.43856317	2.60028295
45	5.38613257	0.64234537	2.58684071	2.36505972	3.49112910	2.95945564
46	3.60860774	1.71429277	0.71412541	0.38388957	2.20450547	0.95404420
47	3.14451454	1.76267219	1.27724289	0.91486475	2.86937209	1.08772990
48	4.52587564	2.37159486	1.03322323	1.22382845	1.19402806	1.69104379
49	4.91632319	0.92760184	1.86024405	1.68526482	2.73149173	2.30506546
50	2.42136081	2.65742361	1.18258066	1.00935603	2.66090599	0.57240184
51	3.49071050	2.32635190	0.06741988	0.30464143	1.66012067	0.64769308
52	3.86908499	1.53787254	0.84877994	0.58981169	2.19833024	1.19831326
53	2.81192264	3.76509602	1.43162986	1.66955183	1.84938407	1.19133869
54	4.18854272	0.76450651	1.66350728	1.34982144	2.98212276	1.86639295
55	3.72337693	1.23715452	2.13937224	1.76995770	3.65997445	2.05139454
56	3.95385414	1.38969304	0.99700280	0.73135029	2.31556124	1.32897179
57	1.40298571	3.54501845	2.12860841	2.01610328	3.41151981	1.47092672
58	5.24294033	0.40068502	2.74853561	2.47428848	3.82754523	3.02163241
59	2.26058926	3.55643271	1.45136999	1.55071881	2.35301667	0.93927761
60	3.09862633	2.08362142	0.74402104	0.42228596	2.34438961	0.56539568

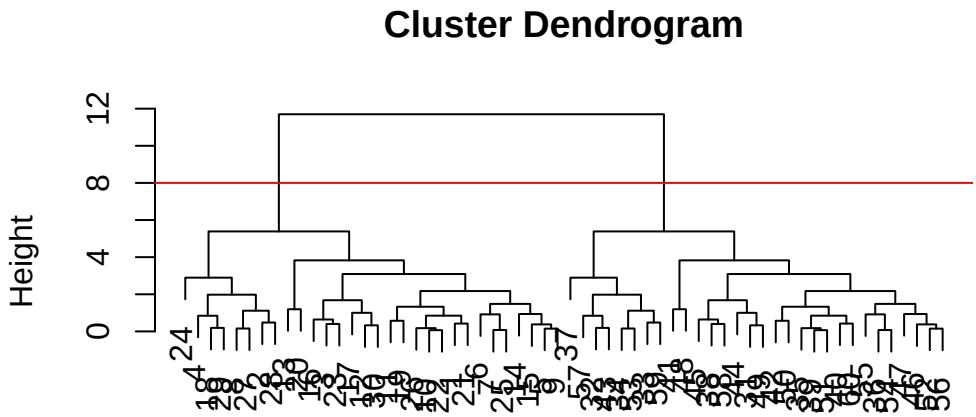
	43	44	45	46	47	48
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						

25						
26						
27						
28						
29						
30						
31						
32						
33						
34						
35						
36						
37						
38						
39						
40						
41						
42						
43						
44	3.25947351					
45	3.21919892	1.21419214				
46	1.51935404	1.75709592	2.00599200			
47	0.97509812	2.35058385	2.24704023	0.74538476		
48	2.66321165	1.32782426	2.30084542	1.28846184	2.03225485	
49	2.75991193	0.71026088	0.77144603	1.36424302	1.80023663	1.53791942
50	0.53157946	2.94360844	3.08988254	1.19062259	0.91527801	2.20008907
51	1.63854536	2.00435394	2.53580948	0.64775648	1.21410140	1.05027806
52	1.76572746	1.49935931	1.77530957	0.26261066	0.92369223	1.17484251
53	1.97761765	3.40156225	4.01395371	2.05096103	2.27373467	2.17585645
54	2.02050743	1.55213590	1.19899039	0.96613094	1.04818995	1.86078269
55	1.68013162	2.46516613	1.86366107	1.45923651	0.96504083	2.64445855
56	1.82715603	1.43324619	1.63530862	0.37694438	0.93576904	1.25346664
57	0.84332843	3.96162248	4.03671827	2.20650441	1.79185369	3.16183148
58	3.10359373	1.67964228	0.50356685	2.09411966	2.16251623	2.63478773
59	1.40450167	3.47814724	3.89873181	1.89309835	1.90284127	2.39668150
60	1.04201327	2.26601922	2.45578846	0.51065674	0.56676749	1.64071104
	49	50	51	52	53	54
2						
3						
4						
5						
6						
7						

```
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50  2.53052899
```

```
51  1.81501362  1.15403477
52  1.10851921  1.45269976  0.78890100
53  3.29172464  1.52331568  1.47926715  2.24901689
54  0.87420778  1.91161688  1.59993865  0.82837126  3.01060055
55  1.77122074  1.82630497  2.07220790  1.48021932  3.23877446  0.91695034
56  0.98964106  1.54787679  0.93721735  0.14831637  2.39514033  0.68775230
57  3.52743529  1.01838620  2.11458852  2.46757873  1.80341272  2.83907496
58  1.11252915  3.05805649  2.69039278  1.90239981  4.14383832  1.15968917
59  3.23595817  1.03227962  1.47379330  2.13374142  0.64475695  2.79216599
60  1.85921990  0.68005364  0.68730212  0.77264920  1.75273713  1.31973826
              55          56          57          58          59
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
```

```
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56  1.39100489
57  2.51462740  2.55585828
58  1.57922215  1.75433961  3.94189536
59  2.84750829  2.26707570  1.15970528  3.95113158
60  1.50949194  0.87151031  1.69614716  2.47930953  1.47306003
```



The function to get our clusters/groups from a hclust object is called `cutree()`

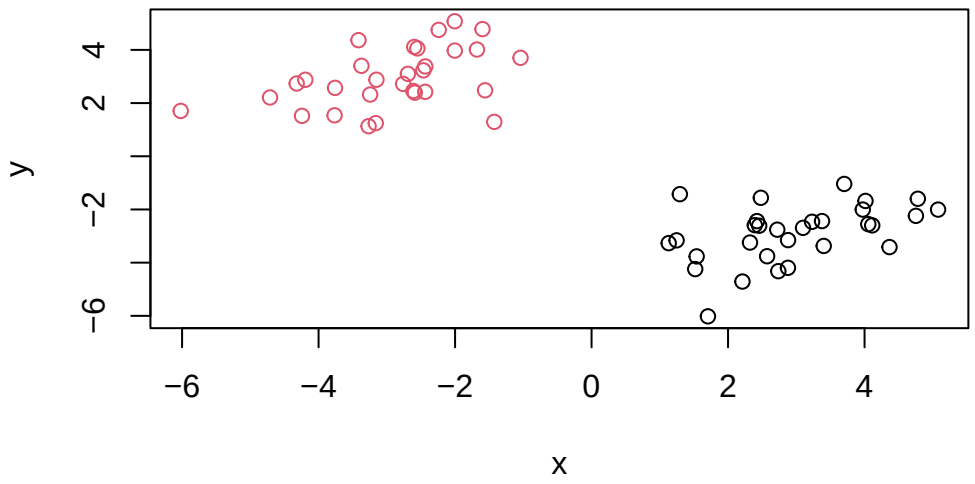
```
groups <- cutree(hc, h = 8)
groups
```

```
 [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
```

We can now see the 2 clusters in the original matrix 'x' and that there is instric hierarchy within.

Q6. Plot our hclust results in terms of our data colored by cluster membership

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



PCA of UK Food Data

```
#Importing UK Food Data
url <- "https://tinyurl.com/UK-foods"
uk_food_data <- 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?

Observing dimensions and first 6 rows

```
dim(uk_food_data)
```

```
[1] 17  5
```

```
head(uk_food_data)
```

```
               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
```

Assigning row names the wrong way...

```
rownames(x) <- x[,1]
x <- x[,-1]
head(x)
```

```
  2.879270479694 1.53983397008687  5.0772704170472 2.21086082389677
       -3.152162        -3.765200        -2.005091        -4.711054
3.37684033250668 4.36571633499587
       -2.436652        -3.414925
```

Assigning it the right way...

```
x <- read.csv(url, row.names=1) #rownames = first column of df
head(x)
```

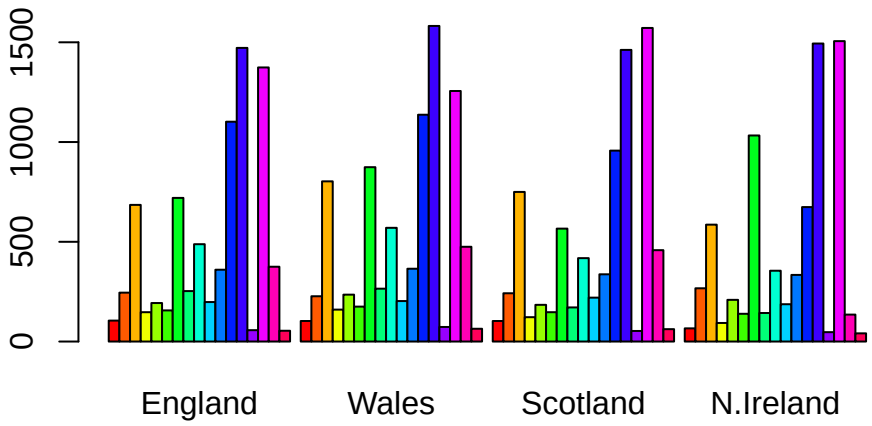
```
               England Wales Scotland N.Ireland
Cheese             105   103      103        66
Carcass_meat       245   227      242       267
Other_meat         685   803      750       586
Fish               147   160      122        93
Fats_and_oils      193   235      184       209
Sugars             156   175      147       139
```

> 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?

Second one, since it does not iteratively remove columns and also assigns the row names correctly as you are importing the dataset into R.

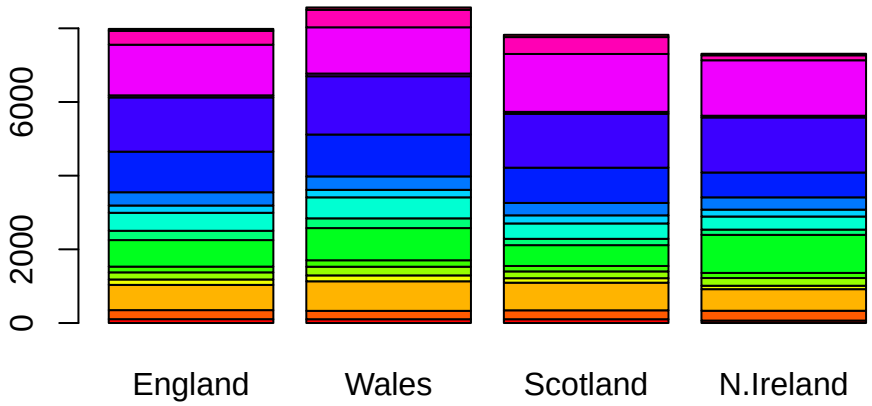
Spotting Major Differences and Trends in Dataset

```
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?

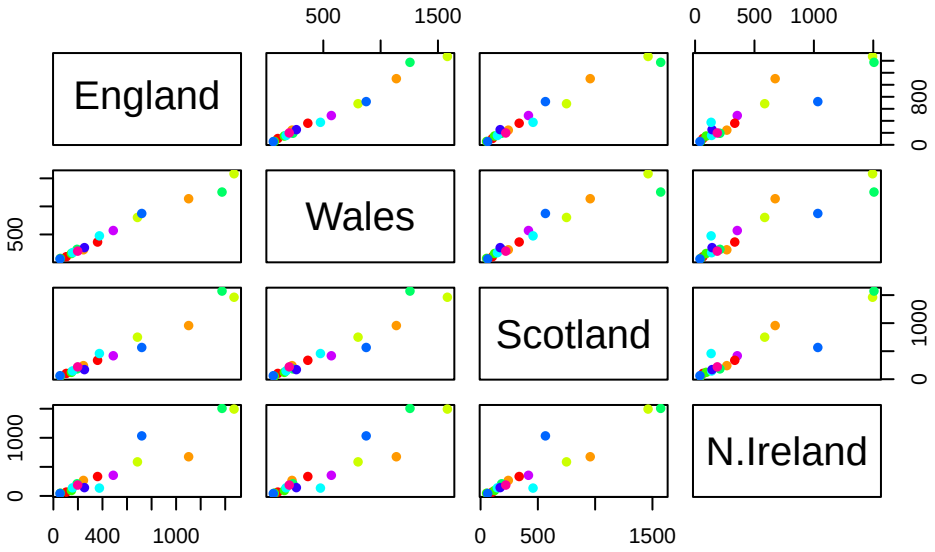
```
barplot(as.matrix(x), beside=F, col=rainbow(nrow(x)))
```




```
#changing beside = T to F results in the shown plot
```

Q4: 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)
```



at the y=x line for each plot implies the different foods are equally favored by both corresponsind countries. Above that line implies the country above favors it more by consumption, and vice versa.

> Q5: 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 function for PCA in base R is called `prcomp()` It wants the transpose (with the `t()`) of our food data for analysis

```
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
```

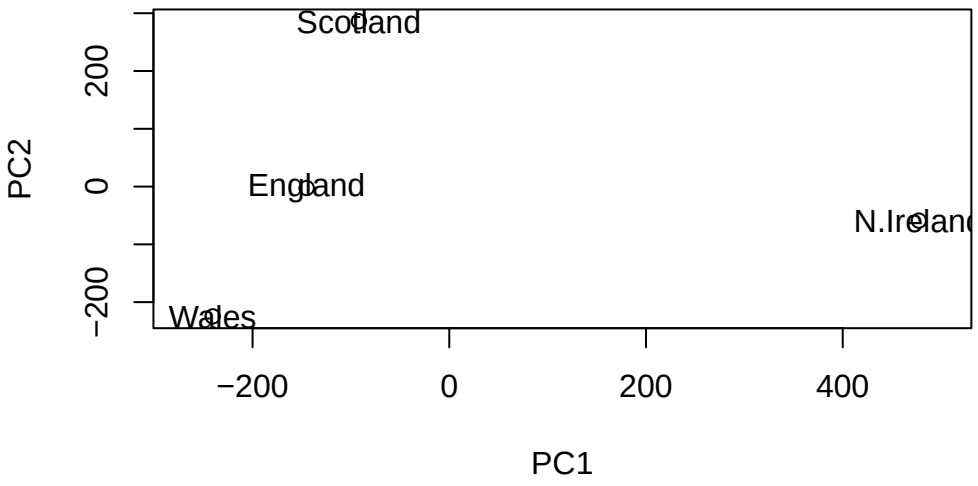
One of the main results that folks look for is called the "score plot" a.k.a PC plot, PC1 vs PC2, etc...

> Q6: Complete the code below to generate a plot of PC1 vs PC2. The second line adds text labels over the data points.

```
pca$x
```

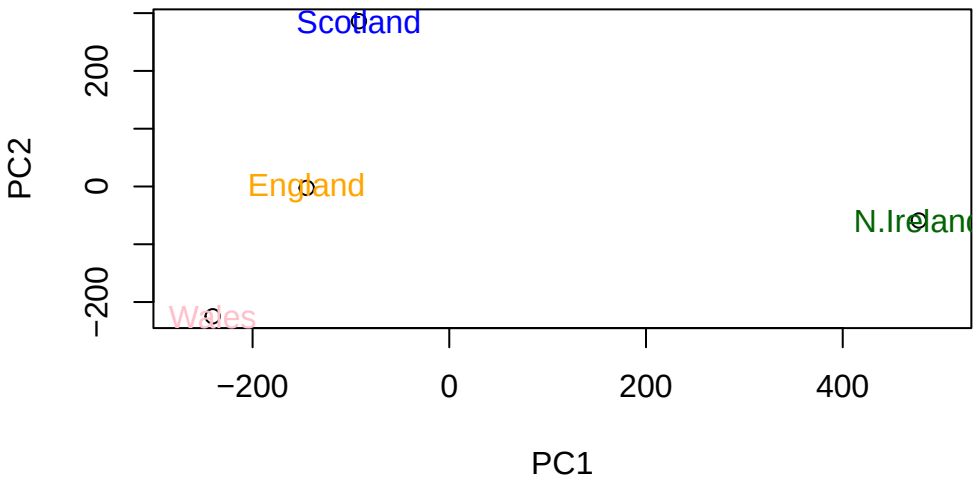
```
                PC1         PC2        PC3           PC4
England   -144.99315   -2.532999 105.768945 -9.152022e-15
Wales     -240.52915 -224.646925 -56.475555  5.560040e-13
Scotland   -91.86934  286.081786 -44.415495 -6.638419e-13
N.Ireland  477.39164  -58.901862  -4.877895  1.329771e-13
```

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



> Q7: 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.

```
colors <- c("orange", "pink", "blue", "#006400")

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 = colors)
```

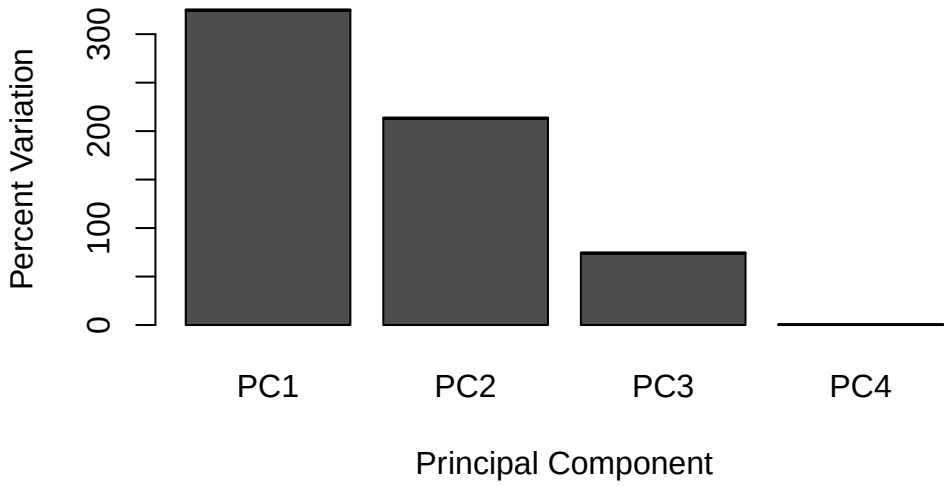


How much variation does each PC account for?

```
z <- summary(pca)
z$importance
```

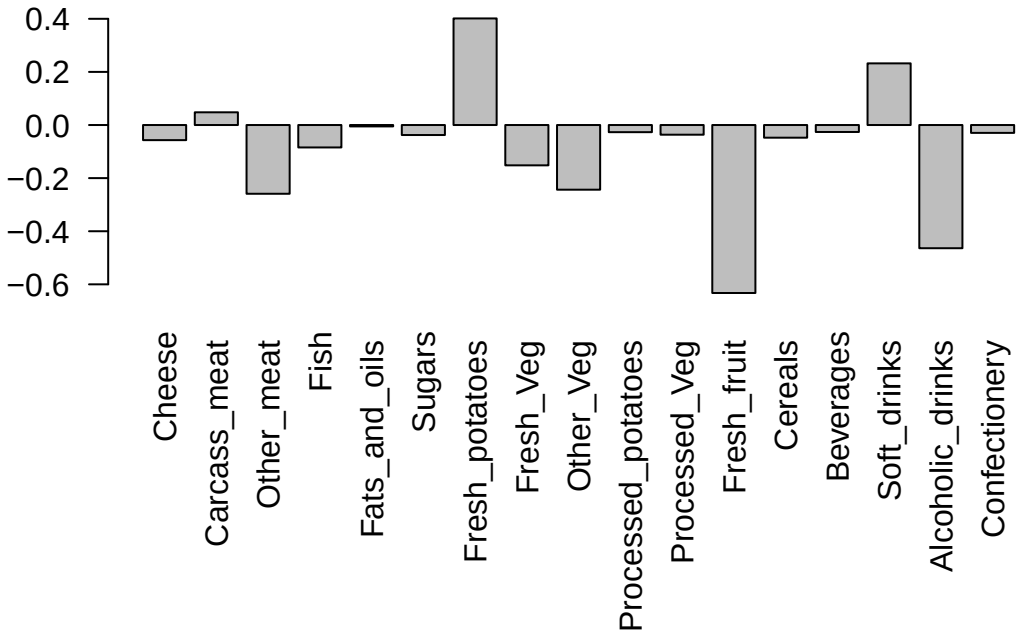
```
                            PC1       PC2      PC3           PC4
Standard deviation     324.15019 212.74780 73.87622 2.921348e-14
Proportion of Variance   0.67444   0.29052  0.03503 0.000000e+00
Cumulative Proportion    0.67444   0.96497  1.00000 1.000000e+00
```

```
#visualizing the above
barplot(z$importance, xlab="Principal Component", ylab="Percent Variation")
```



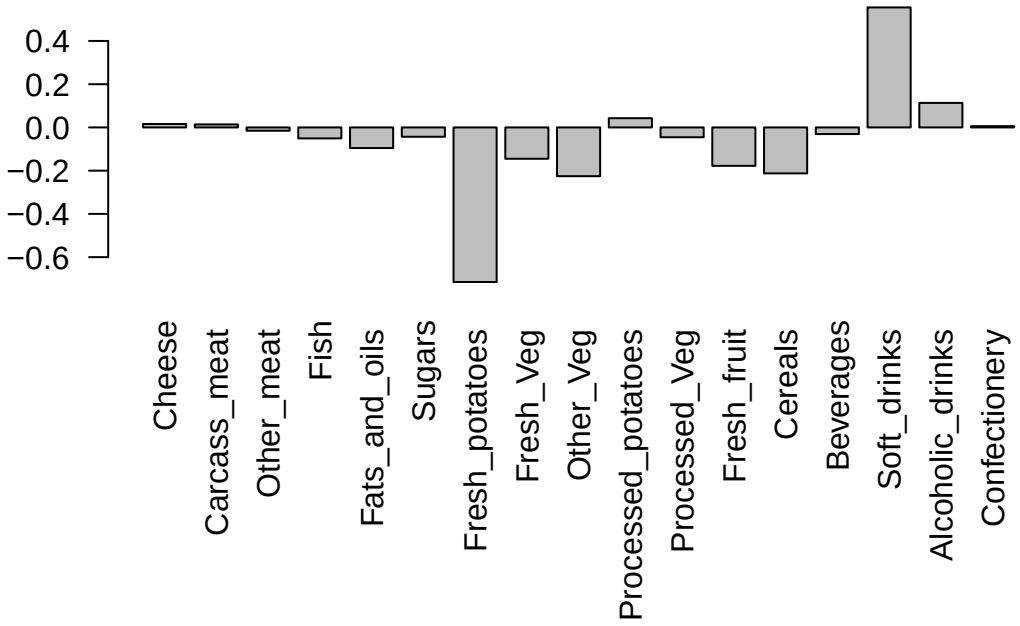
Digging Deeper: Variable Loadings

```
## Lets focus on PC1 as it accounts for > 90% of variance
par(mar=c(10, 3, 0.35, 0))
barplot( pca$rotation[,1], las=2 )
```



Q8: Generate a similar 'loadings plot' for PC2. What two food groups feature prominantely and what does PC2 maninly tell us about?

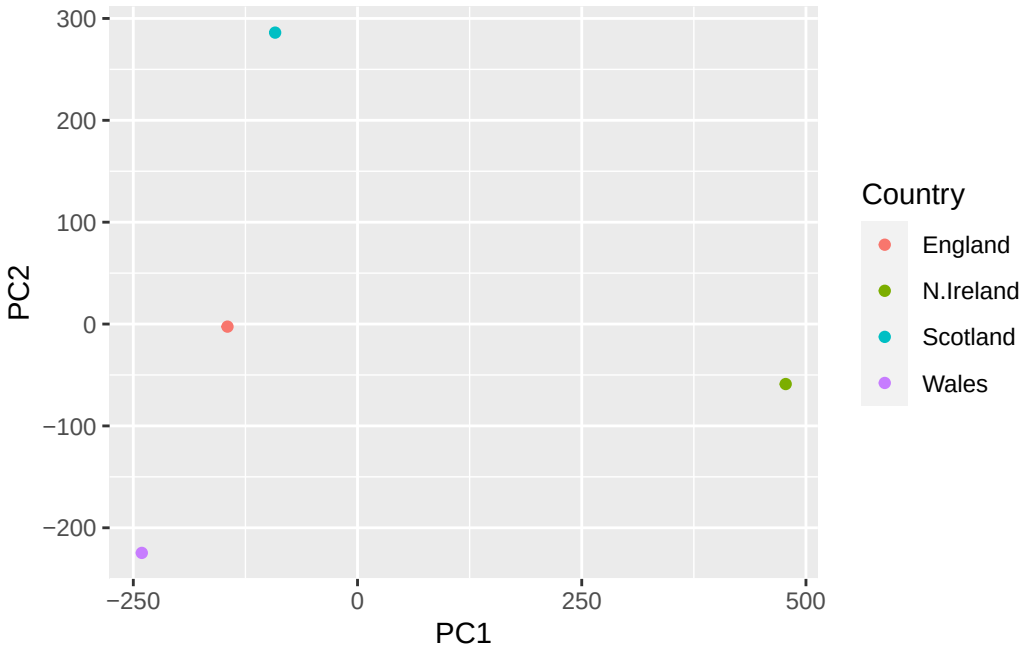
```
par(mar=c(10, 3, 0.35, 0))
barplot( pca$rotation[,2], las = 2)
```



Fresh potatoes and soft drinks contribute to the PC2 loadings the most. The negative PC loading of fresh potatoes implies that Wales consume more of these than the others. And vice versa with soft drinks (Scotland consumes more of these than the rest)

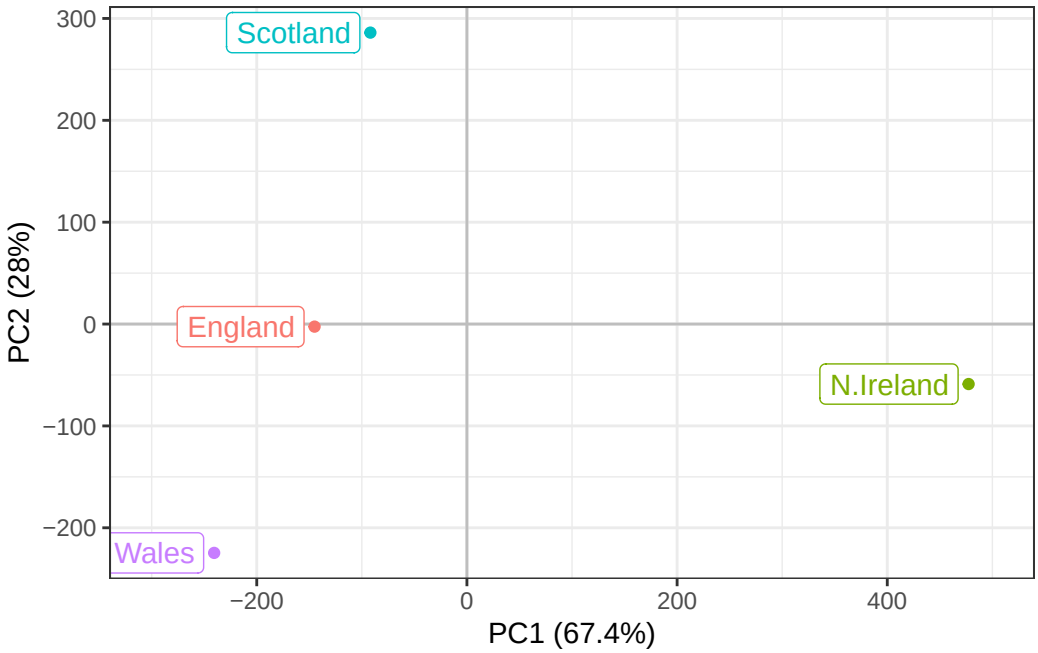
Using ggplot for these figures

```
library(ggplot2)

df <- as.data.frame(pca$x)
df_lab <- tibble::rownames_to_column(df, "Country")

# Our first basic plot
ggplot(df_lab) +
  aes(PC1, PC2, col=Country) +
  geom_point()
```



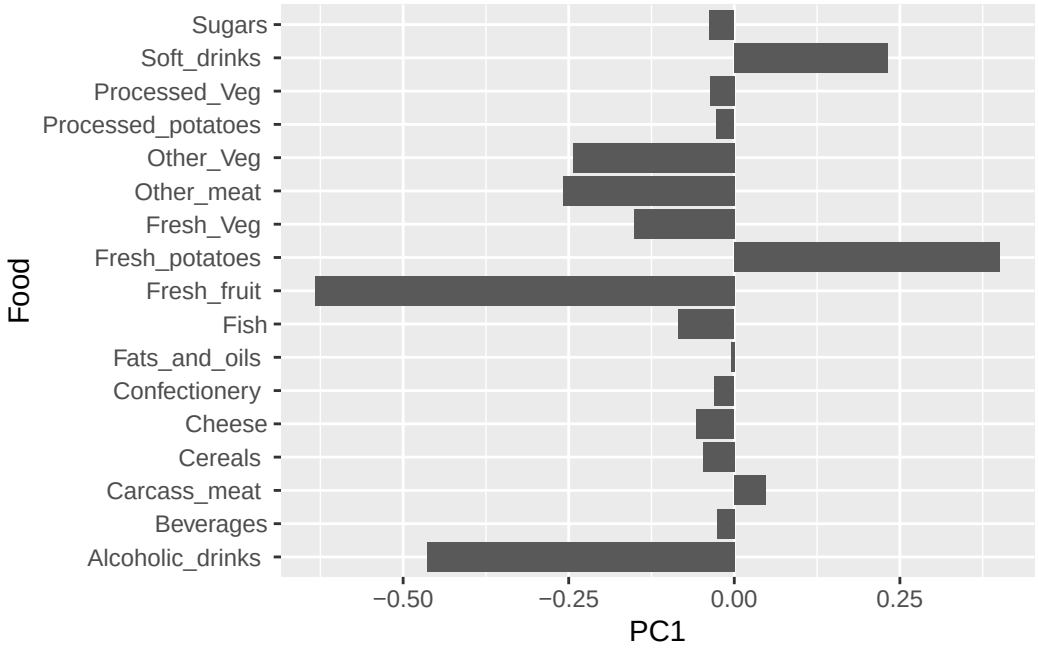
Make it look nicer!

```
ggplot(df_lab) +
 aes(PC1, PC2, col=Country, label=Country) +
 geom_hline(yintercept = 0, col="gray") +
 geom_vline(xintercept = 0, col="gray") +
 geom_point(show.legend = FALSE) +
 geom_label(hjust=1, nudge_x = -10, show.legend = FALSE) +
 expand_limits(x = c(-300,500)) +
 xlab("PC1 (67.4%)") +
 ylab("PC2 (28%)") +
 theme_bw()
```



Looking at loadings...

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

ggplot(ld_lab) +
  aes(PC1, Food) +
  geom_col()
```


Sugars
Soft_drinks
Processed_Veg
Processed_potatoes
Other_Veg
Other_meat
Fresh_Veg
Fresh_potatoes
Fresh_fruit
Fish
Fats_and_oils
Confectionery
Cheese
Cereals
Carcass_meat
Beverages
Alcoholic_drinks
Food
−0.50
−0.25
0.00
0.25
PC1