Class 7: Machine Learning 1

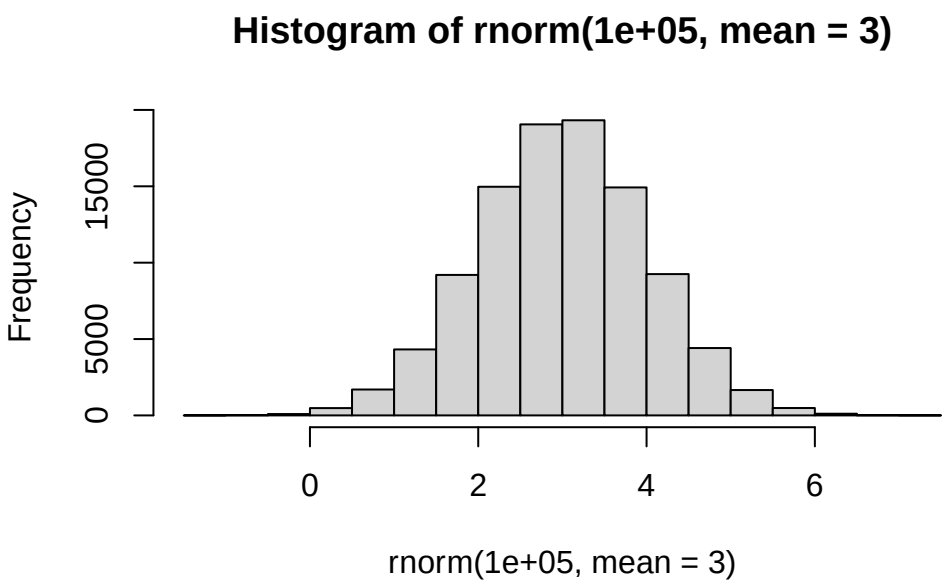
Yu-Chia Huang (PID: A59026739)

#Clustering

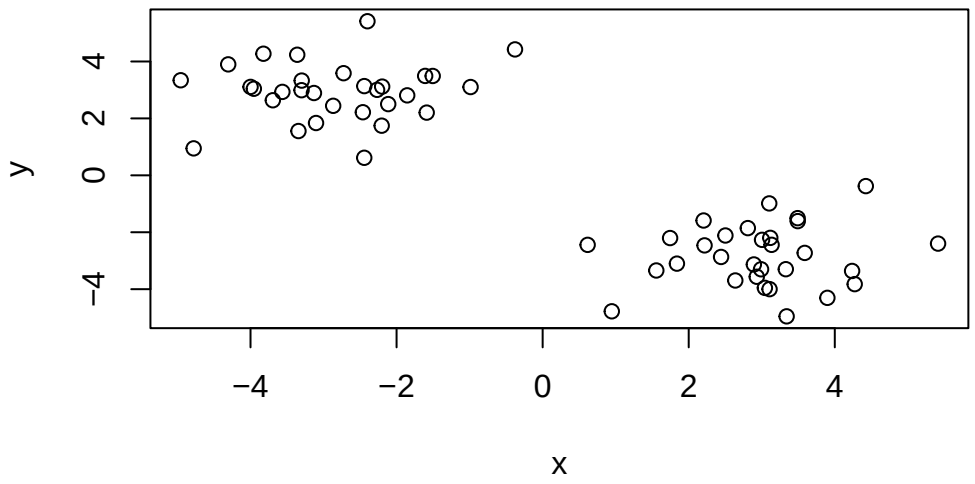
We will star with k-means clustering, one of the most prevelent of all clustering methods.

To get started let's make some data up:

```
hist(rnorm(100000, mean = 3))
```




```
tmp <- c( rnorm(30,3), rnorm(30, -3))
x <- cbind(x=tmp, y=rev(tmp))
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  2.944200 -2.821859
2 -2.821859  2.944200

Clustering vector:
 [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

Within cluster sum of squares by cluster:
[1] 63.67385 63.67385
 (between_SS / total_SS =  88.7 %)

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? Q. What 'component' of your result object details - cluster size? - cluster assignment/membership? - cluster center?

```
k$cluster
```

```
 [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
```

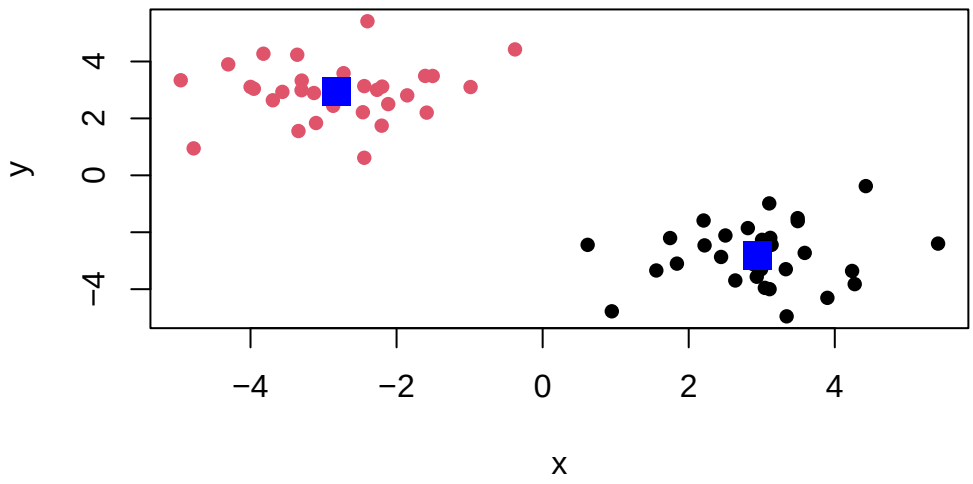
Q3. Cluster centers

```
k$centers
```

```
          x         y
1  2.944200 -2.821859
2 -2.821859  2.944200
```

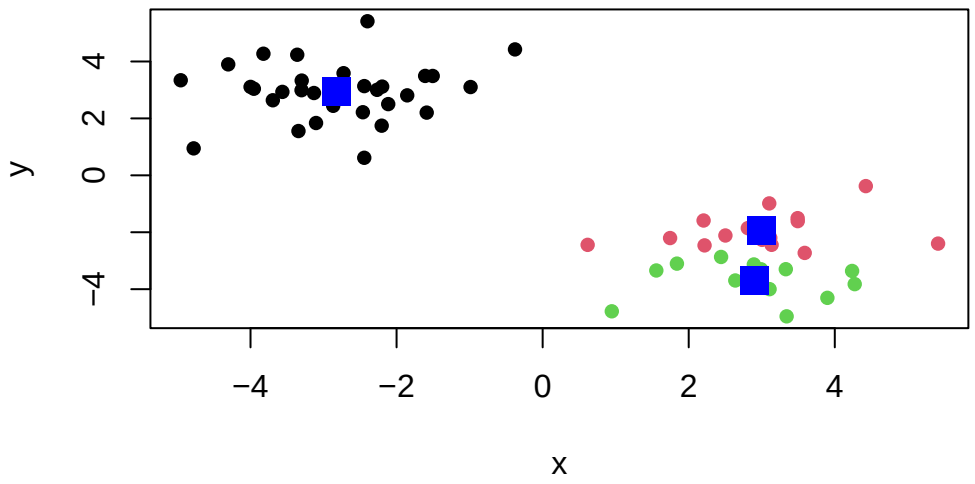
Q4. Make a plot of our data colored by clustering results with optionally the cluster centers shown.

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



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

```
k3 <- kmeans(x, centers=3, nstart=20)

plot(x, col=k3$cluster, pch=16)
points(k3$centers, col="blue", pch=15, cex=2)
```



Hierarchical Clustering

Hierarchical clustering it 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 in "base" R is called `hclust()`.

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

```
dist(x)
```

```
           1          2          3          4          5          6
2  0.61000114
3  2.24785989 1.99498415
4  3.41222585 3.06838433 1.21060396
5  1.40071673 1.64006928 1.57184426 2.74719144
6  3.77370569 3.90852976 2.51269599 2.88136518 2.37769411
7  2.82901486 2.44610631 0.78684186 0.64810906 2.34958383 3.02419154
8  0.85688467 1.46647260 2.76712875 3.97571670 1.44364837 3.70619156
9  0.92482607 1.23709164 1.75856300 2.96812286 0.48112328 2.85876355
10 2.49504732 2.46596304 0.93813900 1.72054888 1.30253178 1.60069663
11 2.01311074 1.78953656 0.24567479 1.45590420 1.34699818 2.51843216
```

12	2.59763351	2.46074540	0.62030312	1.27762758	1.58569306	1.91428572
13	3.00865473	3.00211946	1.37203971	1.82348978	1.73743133	1.15364014
14	2.22560435	2.00799930	0.13268902	1.28399168	1.47359902	2.39421518
15	1.54607104	1.54448669	0.96066469	2.15481651	0.61780462	2.37768931
16	2.72664206	2.34802120	0.69844263	0.73407087	2.25415901	2.99504420
17	3.96133068	3.44793796	2.24003599	1.45520193	3.76652089	4.33186546
18	2.98935344	3.46038758	3.37211697	4.36164353	2.00861263	2.35894663
19	3.20274085	3.04649625	1.09831621	1.08142872	2.15118482	1.80296700
20	2.38501166	2.53859728	1.56639519	2.46468812	0.99437897	1.38918119
21	2.04517228	2.06408743	0.95079874	1.99358758	0.84892321	1.87569350
22	1.15602163	1.07944495	1.12165206	2.32294546	0.79683690	2.84502538
23	2.68156595	2.45310848	0.46050615	0.91582479	1.84793718	2.27081788
24	0.84859577	1.17915091	1.80191259	3.01240789	0.55938885	2.93697393
25	1.35353587	1.38450563	1.11090540	2.31693499	0.52837444	2.52473765
26	1.00346822	0.46440659	1.61463456	2.63059915	1.63367884	3.73519925
27	2.43525180	1.82530056	2.30606072	2.70863784	3.06371673	4.79812036
28	1.60905776	1.29327602	0.70833868	1.80577291	1.35682268	2.98551062
29	2.53196435	2.75870124	1.93634439	2.81875076	1.13715941	1.30181328
30	1.21822189	1.36596222	1.37878036	2.58247643	0.32171706	2.57256256
31	10.39478809	10.34600906	8.42237243	7.73268443	9.07539117	6.80850271
32	9.31646675	9.32443018	7.47154207	6.92954623	7.95880989	5.62756476
33	10.30467382	10.19387734	8.22270449	7.40981104	9.04474483	6.89516584
34	11.58085831	11.39479530	9.39987925	8.44031489	10.40829827	8.41550700
35	11.20950591	11.10098394	9.12982900	8.31009254	9.94398775	7.77371116
36	10.24971562	10.18781998	8.25250549	7.53836279	8.94192332	6.69816331
37	10.83083251	10.79132505	8.87540256	8.19722819	9.50294599	7.21799817
38	9.14935124	9.02978299	7.05565240	6.24378117	7.90445030	5.80429554
39	10.49197624	10.41577657	8.46782784	7.72054869	9.19647078	6.97451130
40	9.55619956	9.50048610	7.57316715	6.88577197	8.24514646	6.00090955
41	9.31443329	9.29810552	7.41428653	6.81830351	7.97287989	5.67096983
42	8.51200778	8.40622192	6.44188176	5.67232882	7.25514239	5.14227836
43	10.14238815	10.23121741	8.49829119	8.11480636	8.74872810	6.37107179
44	9.72119558	9.46927907	7.48807006	6.43468023	8.66138172	6.93916992
45	9.54631738	9.38703102	7.39647227	6.50101010	8.34788567	6.33877092
46	10.06047330	9.99579887	8.05860260	7.34217933	8.75572609	6.51944140
47	9.56203606	9.45412377	7.48595635	6.69025998	8.30208299	6.16272852
48	8.59297612	8.54124010	6.62209838	5.96844543	7.28201681	5.04868184
49	9.08248956	8.98936807	7.03223269	6.27627256	7.80903055	5.64875156
50	9.77511665	9.66986303	7.70292042	6.90895552	8.51135973	6.36095498
51	9.10960610	9.04637537	7.11376367	6.42074100	7.80686723	5.58434347
52	10.75419428	10.71572059	8.80109201	8.12635534	9.42567199	7.13995499
53	11.69805808	11.68251004	9.78929823	9.14652783	10.35119905	8.03054107
54	9.48602887	9.32163318	7.32989391	6.42563468	8.29468453	6.30202512

55	8.02670075	8.05030536	6.23098951	5.77244585	6.66278473	4.32597901
56	10.28844892	10.25775352	8.35293886	7.70308885	8.95502984	6.66278473
57	8.87346384	8.69670660	6.70300629	5.78339971	7.70308885	5.77244585
58	9.60425841	9.48834882	7.51542924	6.70300629	8.35293886	6.23098951
59	11.54135193	11.44961901	9.48834882	8.69670660	10.25775352	8.05030536
60	11.60132079	11.54135193	9.60425841	8.87346384	10.28844892	8.02670075
	7	8	9	10	11	12
2						
3						
4						
5						
6						
7						
8	3.45333425					
9	2.48974867	1.04447655				
10	1.59357447	2.73517716	1.70580898			
11	1.00264136	2.52146400	1.51475816	0.91805362		
12	1.16180657	2.96204969	1.91757739	0.44946311	0.71242258	
13	1.88112229	3.18069335	2.17899373	0.53774888	1.41007182	0.76254151
14	0.90429166	2.70939507	1.68776783	0.81090976	0.21869571	0.52448205
15	1.73292260	1.87844935	0.83669455	0.95089698	0.73041586	1.08898315
16	0.10330084	3.35020258	2.38846728	1.53489782	0.90737262	1.11230306
17	1.46297927	4.70331551	3.83340664	3.03546736	2.43370508	2.59022720
18	4.14584253	2.40033478	2.25203123	2.64190403	3.20057300	3.08470746
19	1.29064579	3.55470053	2.51120048	0.87436001	1.26409424	0.60576704
20	2.29852894	2.38507116	1.47478414	0.74415508	1.45452732	1.19026381
21	1.71937556	2.27210576	1.24073563	0.46507559	0.81214095	0.75591480
22	1.80060487	1.65677609	0.71563623	1.39224903	0.87817427	1.44451957
23	0.76610993	3.14611648	2.11279141	0.84861363	0.67184330	0.40261575
24	2.52333908	0.98435028	0.07831834	1.77649114	1.55710006	1.97934084
25	1.86421786	1.69099424	0.65536647	1.14154018	0.87066922	1.28312965
26	1.99914301	1.82968659	1.33387109	2.21097742	1.43447108	2.13578714
27	2.12014687	3.29170060	2.83607617	3.19795478	2.27990255	2.92583721
28	1.22454366	2.24303640	1.34397519	1.39668889	0.53502169	1.24644936
29	2.66905706	2.40474357	1.60713862	1.10170793	1.81756164	1.55085105
30	2.13300700	1.45059330	0.40614030	1.31348573	1.13946956	1.51145864
31	8.33437483	10.48212924	9.54255736	7.90587086	8.58763213	7.89036116
32	7.48920218	9.33145962	8.43486045	6.85997769	7.61269090	6.90246389
33	8.03755690	10.47846209	9.49569503	7.81244254	8.40927685	7.73662162
34	9.08708756	11.85193955	10.83410413	9.12873407	9.60957789	8.98331707
35	8.93983149	11.37546527	10.39685088	8.71645006	9.31650513	8.64343524
36	8.14562152	10.35613755	9.40604244	7.75719778	8.42225789	7.72906183
37	8.79671679	10.90285050	9.97233117	8.34551681	9.03798126	8.33844400

38	6.86996634	9.34247660	8.35047979	6.66039742	7.24380581	6.57459951
39	8.33550781	10.61687015	9.65737872	7.99736509	8.64295460	7.95521440
40	7.48555969	9.65839353	8.70992941	7.06490745	7.73949236	7.04346532
41	7.39354905	9.36345661	8.44529169	6.84030593	7.56509447	6.85863794
42	6.28894557	8.69108098	7.70428723	6.01967681	6.62433725	5.94723156
43	8.62574328	10.03762907	9.22982074	7.78143890	8.60827482	7.89795280
44	7.08127801	10.09061545	9.05066723	7.35939226	7.71368151	7.14570500
45	7.14088805	9.79114336	8.78029535	7.07744868	7.59765865	6.95269208
46	7.94969421	10.17188137	9.21898624	7.56734847	8.22905393	7.53661954
47	7.31393779	9.73634841	8.75270922	7.06953714	7.67072229	6.99594239
48	6.55756149	8.69687072	7.74644877	6.10234335	6.78478789	6.08620947
49	6.89105830	9.23974470	8.26328755	6.58788632	7.21119933	6.52890461
50	7.53257269	9.94440567	8.96317525	7.28197899	7.88705807	7.21119933
51	7.02104531	9.22537545	8.26927866	6.61645890	7.28197899	6.58788632
52	8.72489913	10.82517916	9.89520017	8.26927866	8.96317525	8.26328755
53	9.73820265	11.73231168	10.82517916	9.22537545	9.94440567	9.23974470
54	7.06655906	9.73820265	8.72489913	7.02104531	7.53257269	6.89105830
55	6.30202512	8.03054107	7.13995499	5.58434347	6.36095498	5.64875156
56	8.29468453	10.35119905	9.42567199	7.80686723	8.51135973	7.80903055
57	6.42563468	9.14652783	8.12635534	6.42074100	6.90895552	6.27627256
58	7.32989391	9.78929823	8.80109201	7.11376367	7.70292042	7.03223269
59	9.32163318	11.68251004	10.71572059	9.04637537	9.66986303	8.98936807
60	9.48602887	11.69805808	10.75419428	9.10960610	9.77511665	9.08248956

	13	14	15	16	17	18
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14	1.26067445					
15	1.47509341	0.87152278				
16	1.84649587	0.81996845	1.63687718			
17	3.24240595	2.36308187	3.15053783	1.54313905		
18	2.69529752	3.24638419	2.54886916	4.06561017	5.60830308	
19	0.76830793	1.05003381	1.69040104	1.28951228	2.52890745	3.42941677
20	0.90358482	1.43399786	1.05377183	2.22895853	3.75603590	1.89826630

21	0.96359525	0.82217132	0.52043906	1.63955732	3.18184134	2.42650475
22	1.92592775	1.08178102	0.46833931	1.69788234	3.11860136	2.80472578
23	1.12053740	0.45609843	1.27952856	0.72613223	2.18785740	3.46686583
24	2.25341269	1.73526512	0.89410096	2.42151317	3.85288889	2.29625881
25	1.65992555	1.03411670	0.19627213	1.76531385	3.25524708	2.52182490
26	2.74729121	1.64789826	1.36429625	1.90317827	2.98747424	3.58273832
27	3.67443351	2.41376796	2.62631345	2.07751560	2.24940434	5.06152747
28	1.91589013	0.74305445	0.80557326	1.12149172	2.49193202	3.34502651
29	1.15497735	1.80411772	1.35259696	2.59997912	4.12530847	1.55899682
30	1.80481129	1.29863138	0.43414085	2.03387568	3.51746362	2.32715083
31	7.38615209	8.37614725	8.85637591	8.39092306	8.64714654	8.93252000
32	6.32683535	7.40982349	7.80080066	7.53320530	8.00549767	7.65063727
33	7.31701620	8.19249625	8.75870516	8.10441467	8.17747204	9.13731237
34	8.67211867	9.39096243	10.05075083	9.16682716	8.95092090	10.72757639
35	8.21875147	9.09972813	9.66345282	9.00752339	9.05049534	9.99136770
36	7.24182818	8.20943589	8.70809347	8.20416986	8.42850918	8.85442365
37	7.82295923	8.82735935	9.29531857	8.85241908	9.11658498	9.30803979
38	6.17167552	7.02671383	7.60405528	6.93631703	7.04222123	8.08844324
39	7.48701069	8.42869206	8.94791440	8.39690055	8.56746023	9.15207898
40	6.54780969	7.52763806	8.01572459	7.54165822	7.81971020	8.16714957
41	6.31196628	7.35845925	7.78692044	7.44199743	7.84719958	7.75777261
42	5.52536238	6.40828794	6.96600468	6.35145567	6.54329001	7.42754699
43	7.24431627	8.41982750	8.68711035	8.65692541	9.29810716	8.09664736
44	6.96055590	7.49813518	8.23356156	7.16870816	6.79314918	9.29810716
45	6.61049527	7.37909188	8.00837589	7.21448878	7.16870816	8.65692541
46	7.05306593	8.01602557	8.51823592	8.00837589	8.23356156	8.68711035
47	6.57399237	7.45425633	8.01602557	7.37909188	7.49813518	8.41982750
48	5.58447064	6.57399237	7.05306593	6.61049527	6.96055590	7.24431627
49	6.08620947	6.99594239	7.53661954	6.95269208	7.14570500	7.89795280
50	6.78478789	7.67072229	8.22905393	7.59765865	7.71368151	8.60827482
51	6.10234335	7.06953714	7.56734847	7.07744868	7.35939226	7.78143890
52	7.74644877	8.75270922	9.21898624	8.78029535	9.05066723	9.22982074
53	8.69687072	9.73634841	10.17188137	9.79114336	10.09061545	10.03762907
54	6.55756149	7.31393779	7.94969421	7.14088805	7.08127801	8.62574328
55	5.04868184	6.16272852	6.51944140	6.33877092	6.93916992	6.37107179
56	7.28201681	8.30208299	8.75572609	8.34788567	8.66138172	8.74872810
57	5.96844543	6.69025998	7.34217933	6.50101010	6.43468023	8.11480636
58	6.62209838	7.48595635	8.05860260	7.39647227	7.48807006	8.49829119
59	8.54124010	9.45412377	9.99579887	9.38703102	9.46927907	10.23121741
60	8.59297612	9.56203606	10.06047330	9.54631738	9.72119558	10.14238815

	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  1.55846052
21  1.30281773  0.64832633
22  2.04839621  1.50362697  0.98543260
23  0.66224735  1.58134985  1.07787238  1.53496250
24  2.57535805  1.55310145  1.31150368  0.73506765  2.16528806
25  1.88567506  1.16890246  0.69802067  0.33885845  1.45788244  0.70779565
26  2.69778207  2.41212065  1.86021660  0.90961466  2.07491188  1.29822279
27  3.31163177  3.64410761  3.00791893  2.27059296  2.66092283  2.80797405
28  1.79277691  1.78968527  1.15398304  0.62836391  1.16884375  1.36111881
29  1.87141544  0.37115225  1.00622448  1.77384228  1.94626308  1.68364895
30  2.10605651  1.18632694  0.85004994  0.47786907  1.71355031  0.46946384
31  7.32681610  8.09926605  8.34971706  9.29148684  7.97018876  9.61939176
32  6.37572427  6.96835090  7.28400486  8.25201455  7.03492946  8.51277889
33  7.14759904  8.09966338  8.27065831  9.17051154  7.76306074  9.57036877
34  8.37877397  9.51272725  9.59370195 10.42716997  8.94197764 10.90520403
35  8.05476215  8.99436615  9.17388678 10.07658999  8.67013439 10.47176627
36  7.15944808  7.97141498  8.20476056  9.13877432  7.79823611  9.48248432
37  7.77877296  8.52278973  8.78646792  9.73339384  8.42451505 10.04944352
38  5.98328884  6.97081234  7.12061496  8.01135603  6.59578625  8.42451505
39  7.37895699  8.23184794  8.44843952  9.37351715  8.01135603  9.73339384
40  6.47807484  7.27377810  7.51106034  8.44843952  7.12061496  8.78646792
41  6.31607988  6.98803078  7.27377810  8.23184794  6.97081234  8.52278973
42  5.36117643  6.31607988  6.47807484  7.37895699  5.98328884  7.77877296
43  7.42754699  7.75777261  8.16714957  9.15207898  8.08844324  9.30803979
44  6.54329001  7.84719958  7.81971020  8.56746023  7.04222123  9.11658498
45  6.35145567  7.44199743  7.54165822  8.39690055  6.93631703  8.85241908
46  6.96600468  7.78692044  8.01572459  8.94791440  7.60405528  9.29531857
```

47	6.40828794	7.35845925	7.52763806	8.42869206	7.02671383	8.82735935
48	5.52536238	6.31196628	6.54780969	7.48701069	6.17167552	7.82295923
49	5.94723156	6.85863794	7.04346532	7.95521440	6.57459951	8.33844400
50	6.62433725	7.56509447	7.73949236	8.64295460	7.24380581	9.03798126
51	6.01967681	6.84030593	7.06490745	7.99736509	6.66039742	8.34551681
52	7.70428723	8.44529169	8.70992941	9.65737872	8.35047979	9.97233117
53	8.69108098	9.36345661	9.65839353	10.61687015	9.34247660	10.90285050
54	6.28894557	7.39354905	7.48555969	8.33550781	6.86996634	8.79671679
55	5.14227836	5.67096983	6.00090955	6.97451130	5.80429554	7.21799817
56	7.25514239	7.97287989	8.24514646	9.19647078	7.90445030	9.50294599
57	5.67232882	6.81830351	6.88577197	7.72054869	6.24378117	8.19722819
58	6.44188176	7.41428653	7.57316715	8.46782784	7.05565240	8.87540256
59	8.40622192	9.29810552	9.50048610	10.41577657	9.02978299	10.79132505
60	8.51200778	9.31443329	9.55619956	10.49197624	9.14935124	10.83083251

	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	1.24778235					
27	2.58601680	1.51989923				
28	0.82932421	0.90723197	1.85497042			
29	1.43505352	2.68085429	3.97192074	2.12379709		

30	0.26910219	1.32095112	2.74205815	1.06418085	1.39310453	
31	9.04459246	10.02189870	10.44103285	9.11919724	8.09872571	9.18581226
32	7.98194133	9.03510862	9.60918927	8.14770379	6.92901358	8.09872571
33	8.95220209	9.83695898	10.10486170	8.92977943	8.14770379	9.11919724
34	10.24694613	11.00111757	11.04924432	10.10486170	9.60918927	10.44103285
35	9.85670130	10.74412789	11.00111757	9.83695898	9.03510862	10.02189870
36	8.89766026	9.85670130	10.24694613	8.95220209	7.98194133	9.04459246
37	9.48248432	10.47176627	10.90520403	9.57036877	8.51277889	9.61939176
38	7.79823611	8.67013439	8.94197764	7.76306074	7.03492946	7.97018876
39	9.13877432	10.07658999	10.42716997	9.17051154	8.25201455	9.29148684
40	8.20476056	9.17388678	9.59370195	8.27065831	7.28400486	8.34971706
41	7.97141498	8.99436615	9.51272725	8.09966338	6.96835090	8.09926605
42	7.15944808	8.05476215	8.37877397	7.14759904	6.37572427	7.32681610
43	8.85442365	9.99136770	10.72757639	9.13731237	7.65063727	8.93252000
44	8.42850918	9.05049534	8.95092090	8.17747204	8.00549767	8.64714654
45	8.20416986	9.00752339	9.16682716	8.10441467	7.53320530	8.39092306
46	8.70809347	9.66345282	10.05075083	8.75870516	7.80080066	8.85637591
47	8.20943589	9.09972813	9.39096243	8.19249625	7.40982349	8.37614725
48	7.24182818	8.21875147	8.67211867	7.31701620	6.32683535	7.38615209
49	7.72906183	8.64343524	8.98331707	7.73662162	6.90246389	7.89036116
50	8.42225789	9.31650513	9.60957789	8.40927685	7.61269090	8.58763213
51	7.75719778	8.71645006	9.12873407	7.81244254	6.85997769	7.90587086
52	9.40604244	10.39685088	10.83410413	9.49569503	8.43486045	9.54255736
53	10.35613755	11.37546527	11.85193955	10.47846209	9.33145962	10.48212924
54	8.14562152	8.93983149	9.08708756	8.03755690	7.48920218	8.33437483
55	6.69816331	7.77371116	8.41550700	6.89516584	5.62756476	6.80850271
56	8.94192332	9.94398775	10.40829827	9.04474483	7.95880989	9.07539117
57	7.53836279	8.31009254	8.44031489	7.40981104	6.92954623	7.73268443
58	8.25250549	9.12982900	9.39987925	8.22270449	7.47154207	8.42237243
59	10.18781998	11.10098394	11.39479530	10.19387734	9.32443018	10.34600906
60	10.24971562	11.20950591	11.58085831	10.30467382	9.31646675	10.39478809

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

1.39310453						
1.06418085	2.12379709					
2.74205815	3.97192074	1.85497042				
1.32095112	2.68085429	0.90723197	1.51989923			
0.26910219	1.43505352	0.82932421	2.58601680	1.24778235		
0.46946384	1.68364895	1.36111881	2.80797405	1.29822279	0.70779565	
1.71355031	1.94626308	1.16884375	2.66092283	2.07491188	1.45788244	
0.47786907	1.77384228	0.62836391	2.27059296	0.90961466	0.33885845	
0.85004994	1.00622448	1.15398304	3.00791893	1.86021660	0.69802067	
1.18632694	0.37115225	1.78968527	3.64410761	2.41212065	1.16890246	
2.10605651	1.87141544	1.79277691	3.31163177	2.69778207	1.88567506	
2.32715083	1.55899682	3.34502651	5.06152747	3.58273832	2.52182490	
3.51746362	4.12530847	2.49193202	2.24940434	2.98747424	3.25524708	
2.03387568	2.59997912	1.12149172	2.07751560	1.90317827	1.76531385	
0.43414085	1.35259696	0.80557326	2.62631345	1.36429625	0.19627213	
1.29863138	1.80411772	0.74305445	2.41376796	1.64789826	1.03411670	
1.80481129	1.15497735	1.91589013	3.67443351	2.74729121	1.65992555	
1.51145864	1.55085105	1.24644936	2.92583721	2.13578714	1.28312965	
1.13946956	1.81756164	0.53502169	2.27990255	1.43447108	0.87066922	
1.31348573	1.10170793	1.39668889	3.19795478	2.21097742	1.14154018	
0.40614030	1.60713862	1.34397519	2.83607617	1.33387109	0.65536647	
1.45059330	2.40474357	2.24303640	3.29170060	1.82968659	1.69099424	
2.13300700	2.66905706	1.22454366	2.12014687	1.99914301	1.86421786	
2.57256256	1.30181328	2.98551062	4.79812036	3.73519925	2.52473765	

```
56  0.32171706  1.13715941  1.35682268  3.06371673  1.63367884  0.52837444
57  2.58247643  2.81875076  1.80577291  2.70863784  2.63059915  2.31693499
58  1.37878036  1.93634439  0.70833868  2.30606072  1.61463456  1.11090540
59  1.36596222  2.75870124  1.29327602  1.82530056  0.46440659  1.38450563
60  1.21822189  2.53196435  1.60905776  2.43525180  1.00346822  1.35353587
              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  2.16528806
```

39	0.73506765	1.53496250				
40	1.31150368	1.07787238	0.98543260			
41	1.55310145	1.58134985	1.50362697	0.64832633		
42	2.57535805	0.66224735	2.04839621	1.30281773	1.55846052	
43	2.29625881	3.46686583	2.80472578	2.42650475	1.89826630	3.42941677
44	3.85288889	2.18785740	3.11860136	3.18184134	3.75603590	2.52890745
45	2.42151317	0.72613223	1.69788234	1.63955732	2.22895853	1.28951228
46	0.89410096	1.27952856	0.46833931	0.52043906	1.05377183	1.69040104
47	1.73526512	0.45609843	1.08178102	0.82217132	1.43399786	1.05003381
48	2.25341269	1.12053740	1.92592775	0.96359525	0.90358482	0.76830793
49	1.97934084	0.40261575	1.44451957	0.75591480	1.19026381	0.60576704
50	1.55710006	0.67184330	0.87817427	0.81214095	1.45452732	1.26409424
51	1.77649114	0.84861363	1.39224903	0.46507559	0.74415508	0.87436001
52	0.07831834	2.11279141	0.71563623	1.24073563	1.47478414	2.51120048
53	0.98435028	3.14611648	1.65677609	2.27210576	2.38507116	3.55470053
54	2.52333908	0.76610993	1.80060487	1.71937556	2.29852894	1.29064579
55	2.93697393	2.27081788	2.84502538	1.87569350	1.38918119	1.80296700
56	0.55938885	1.84793718	0.79683690	0.84892321	0.99437897	2.15118482
57	3.01240789	0.91582479	2.32294546	1.99358758	2.46468812	1.08142872
58	1.80191259	0.46050615	1.12165206	0.95079874	1.56639519	1.09831621
59	1.17915091	2.45310848	1.07944495	2.06408743	2.53859728	3.04649625
60	0.84859577	2.68156595	1.15602163	2.04517228	2.38501166	3.20274085

	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						

5.60830308						
4.06561017	1.54313905					
2.54886916	3.15053783	1.63687718				
3.24638419	2.36308187	0.81996845	0.87152278			
2.69529752	3.24240595	1.84649587	1.47509341	1.26067445		
3.08470746	2.59022720	1.11230306	1.08898315	0.52448205	0.76254151	
3.20057300	2.43370508	0.90737262	0.73041586	0.21869571	1.41007182	
2.64190403	3.03546736	1.53489782	0.95089698	0.81090976	0.53774888	
2.25203123	3.83340664	2.38846728	0.83669455	1.68776783	2.17899373	
2.40033478	4.70331551	3.35020258	1.87844935	2.70939507	3.18069335	
4.14584253	1.46297927	0.10330084	1.73292260	0.90429166	1.88112229	
2.35894663	4.33186546	2.99504420	2.37768931	2.39421518	1.15364014	
2.00861263	3.76652089	2.25415901	0.61780462	1.47359902	1.73743133	
4.36164353	1.45520193	0.73407087	2.15481651	1.28399168	1.82348978	
3.37211697	2.24003599	0.69844263	0.96066469	0.13268902	1.37203971	
3.46038758	3.44793796	2.34802120	1.54448669	2.00799930	3.00211946	
2.98935344	3.96133068	2.72664206	1.54607104	2.22560435	3.00865473	

49	50	51	52	53	54

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

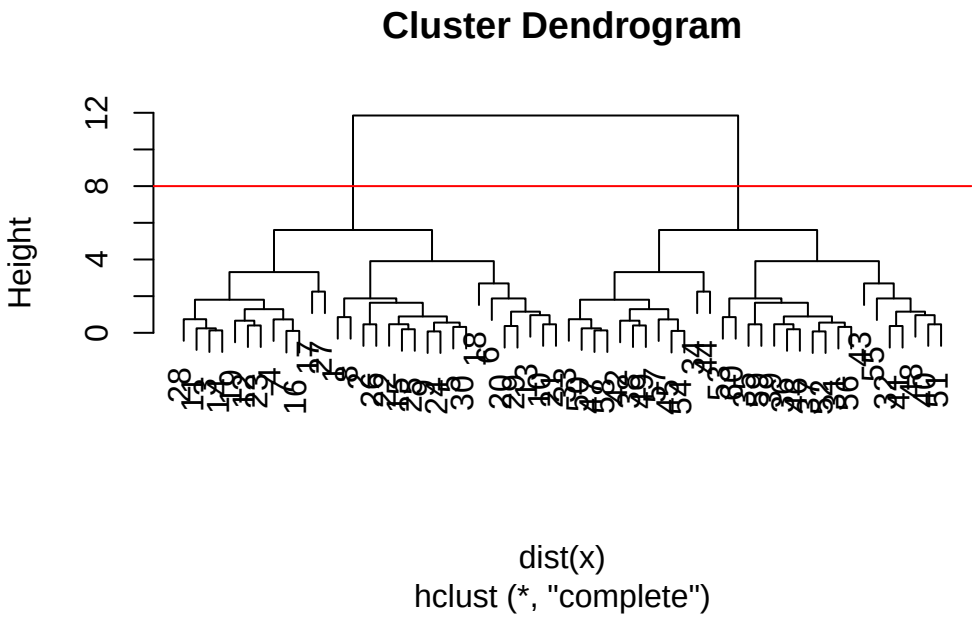
```
48
49
50  0.71242258
51  0.44946311  0.91805362
52  1.91757739  1.51475816  1.70580898
53  2.96204969  2.52146400  2.73517716  1.04447655
54  1.16180657  1.00264136  1.59357447  2.48974867  3.45333425
55  1.91428572  2.51843216  1.60069663  2.85876355  3.70619156  3.02419154
56  1.58569306  1.34699818  1.30253178  0.48112328  1.44364837  2.34958383
57  1.27762758  1.45590420  1.72054888  2.96812286  3.97571670  0.64810906
58  0.62030312  0.24567479  0.93813900  1.75856300  2.76712875  0.78684186
59  2.46074540  1.78953656  2.46596304  1.23709164  1.46647260  2.44610631
60  2.59763351  2.01311074  2.49504732  0.92482607  0.85688467  2.82901486
              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  2.37769411
57  2.88136518  2.74719144
58  2.51269599  1.57184426  1.21060396
59  3.90852976  1.64006928  3.06838433  1.99498415
60  3.77370569  1.40071673  3.41222585  2.24785989  0.61000114
```

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

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

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

```
plot(hc)
abline(h=8, col="red")
```



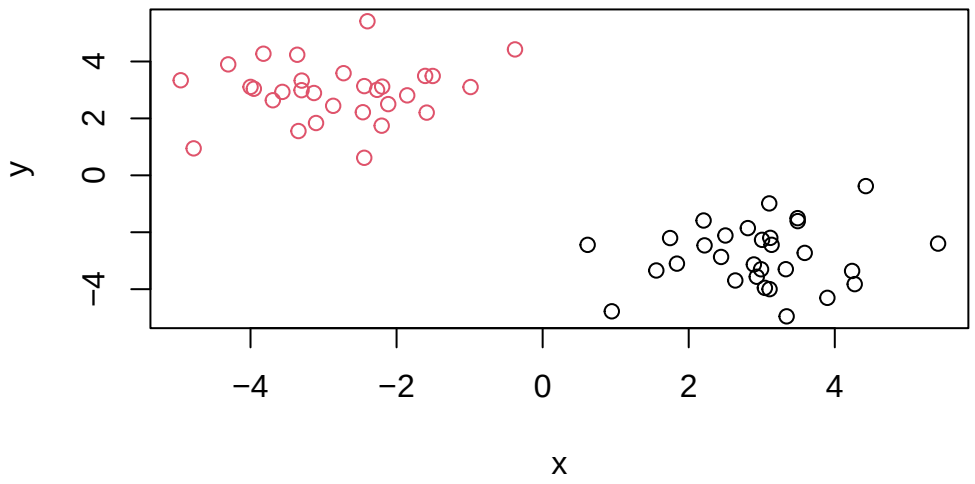
The function to get our clusters/groups from a hclust abject is claaed `cutree()`.

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

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

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



Principal Component Analysis (PCA)

One of the most useful tools (!?)

#We will work on data from the strange stuff folks there eat. It has 17 different foods for 4 countries.

#Load the 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? #We can use `nrow()`, `ncol()`, or `dim()` functions.

```
dim(x) #Return the row number and column number.
```

```
[1] 17  5
```

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?

#We can do it like this below:

```
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 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
```

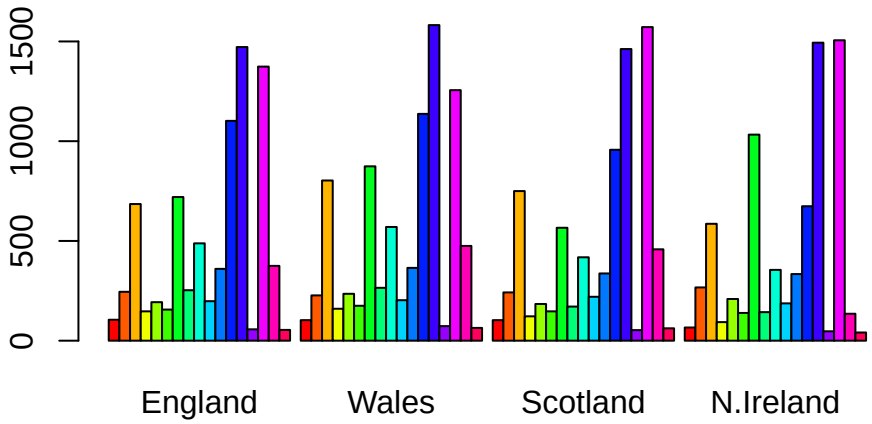
#Preview the first 6 rows. (We can use `view()`, `head()`, and `tail()` functions.)

```
head(x,6)
```

```
              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
```

#Use the `barplot` function.

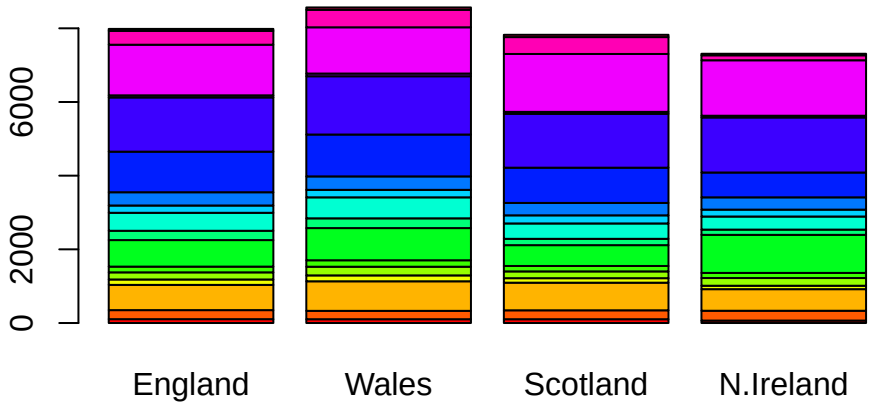
```
barplot(as.matrix(x), beside=T, col=rainbow(nrow(x)))
```



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

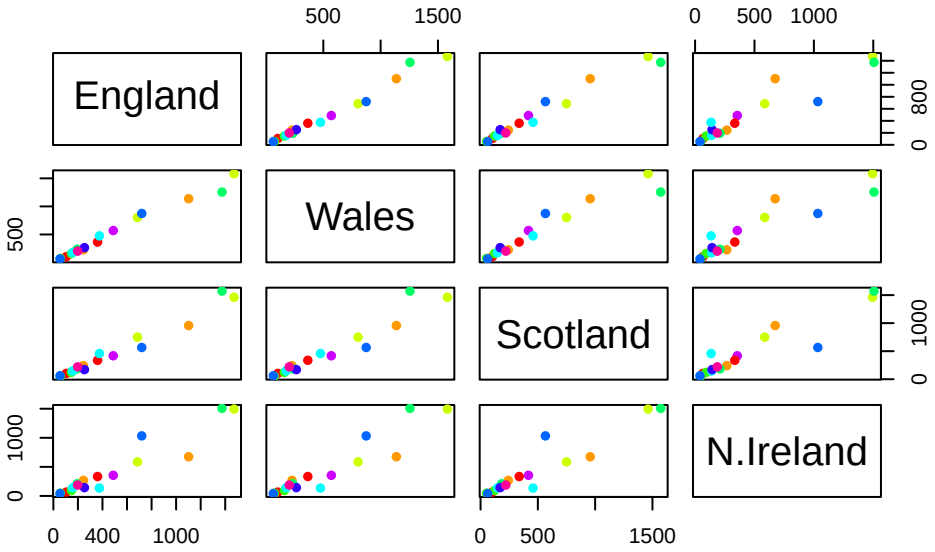
#When looking for `?barplot`, the document shows that beside
a logical value. If FALSE, the columns of height are portrayed as stacked bars, and if TRUE the columns are portrayed as juxtaposed bars. (The default seems to be FALSE?)

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



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)
```



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

When we look at the right plots of N. Ireland,we can draw a diagonal in each plot. The blue dots is far from the diagonal, which is the main difference.

#PCA to the rescue

Help me make sense of this data… The main function for PCA in base R is called `prcomp()`.

It wants the transpose (with the `t()`) of our food data for analysis.

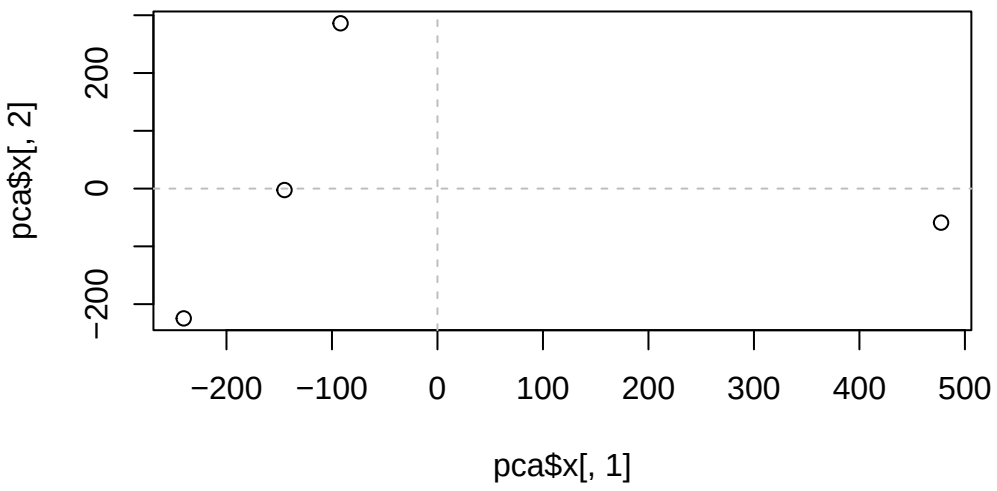
```
pca <- prcomp(t(x))
summary(pca) #Give SD, proportion of variance and cumulative proportion.
```

```
Importance of components:
                            PC1      PC2      PC3       PC4
Standard deviation     324.1502 212.7478 73.87622 3.176e-14
Proportion of Variance   0.6744   0.2905  0.03503 0.000e+00
Cumulative Proportion    0.6744   0.9650  1.00000 1.000e+00
```

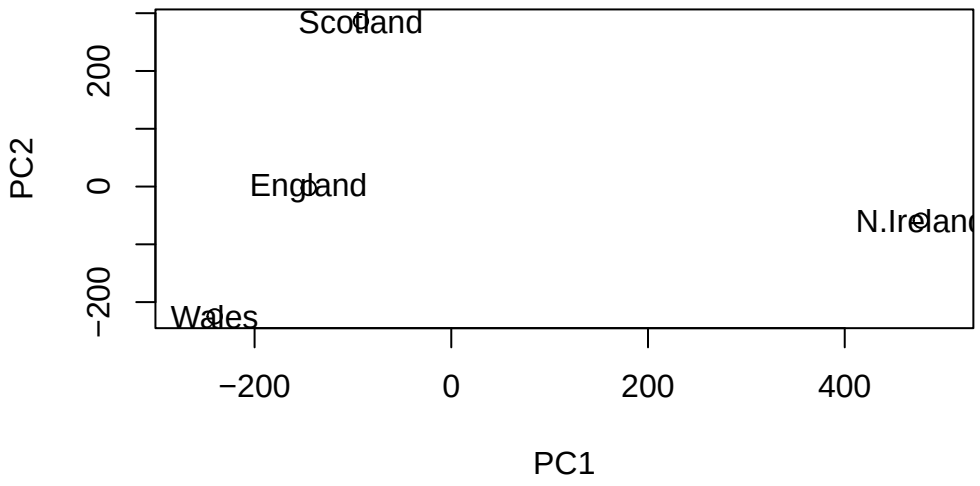
One of the main results that folks look for is called the “score plot” a.k.a. PC plot, PC1 vs PC2 plot…

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])
abline(h=0, col="gray", lty=2)
abline(v=0, col="gray", lty=2)
```

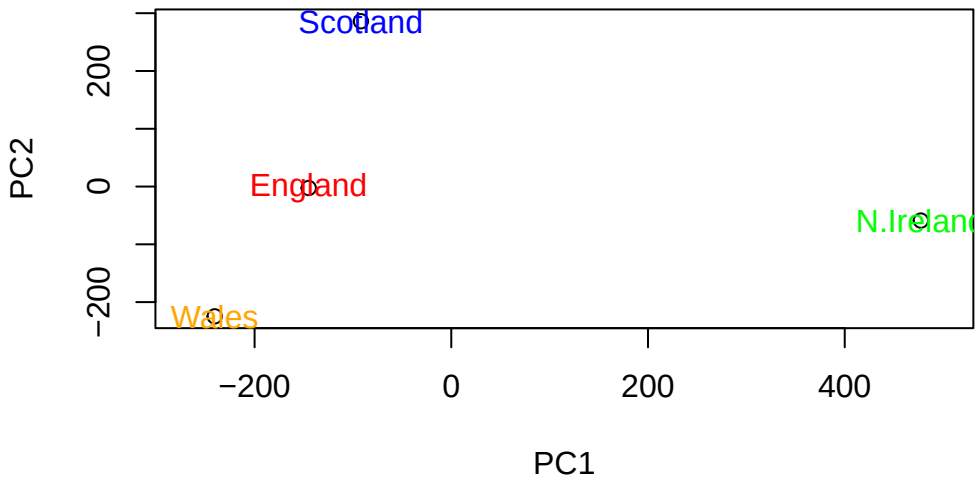



```
# Plot PC1 vs PC2
plot(pca$x[,1], pca$x[,2], xlab="PC1", ylab="PC2", xlim=c(-270,500))
text(pca$x[,1], pca$x[,2], colnames(x))
```



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.

```
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("red", "orange", "blue", "green"))
```



Standard deviation:

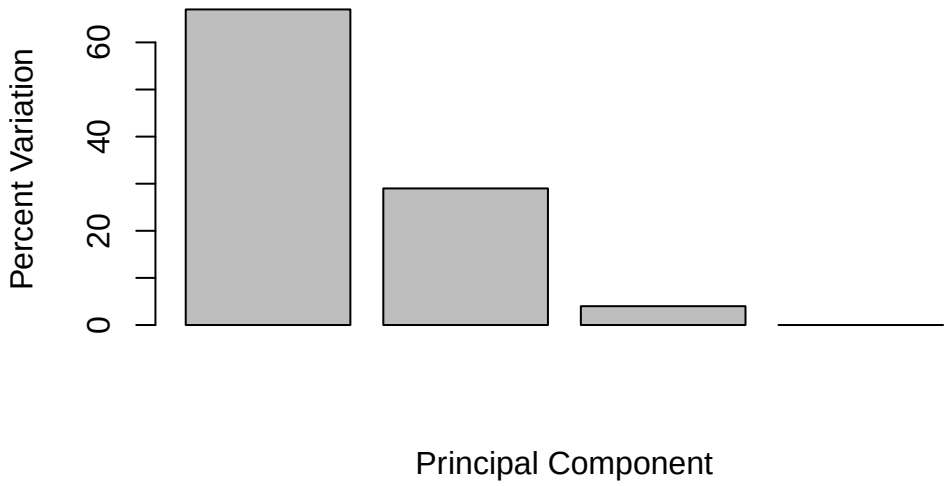
```
v <- round( pca$sdev^2/sum(pca$sdev^2) * 100 )
v
```

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

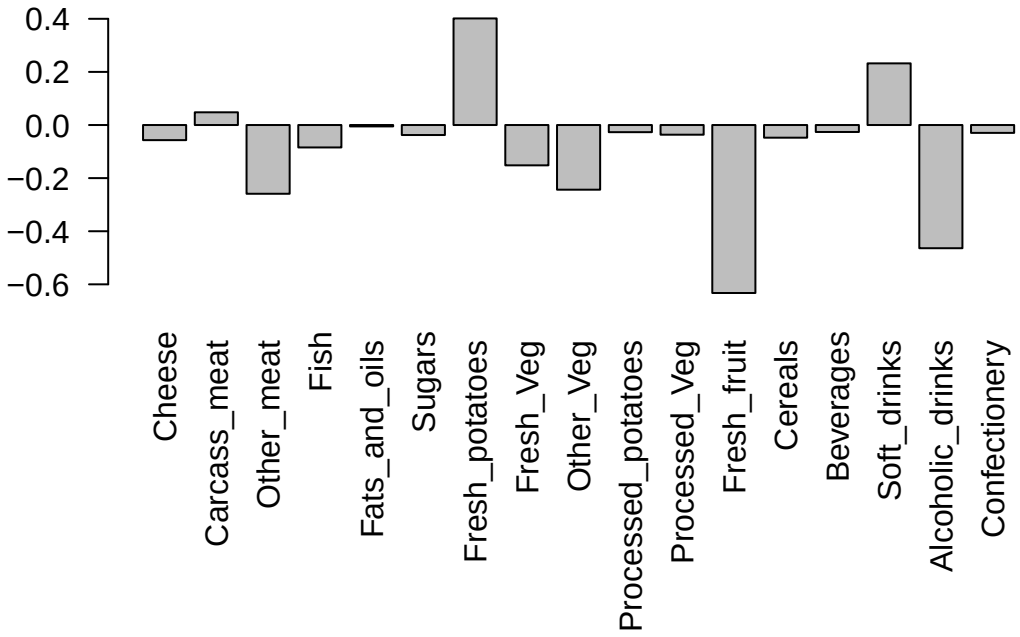
```
## or the second row here...
z <- summary(pca)
z$importance
```

```
                             PC1       PC2      PC3           PC4
Standard deviation     324.15019 212.74780 73.87622 3.175833e-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(v, xlab="Principal Component", ylab="Percent Variation")
```

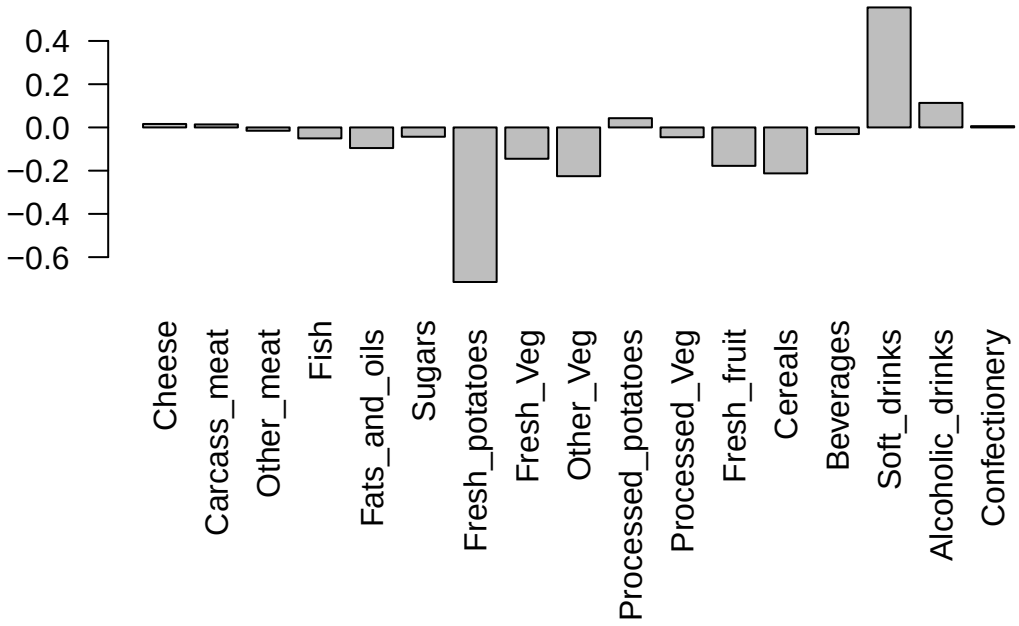



```
## Lets focus on PC1 as it accounts for > 90% of variance
par(mar=c(10, 3, 0.35, 0))
barplot( pca$rotation[,1], las=2 )
```



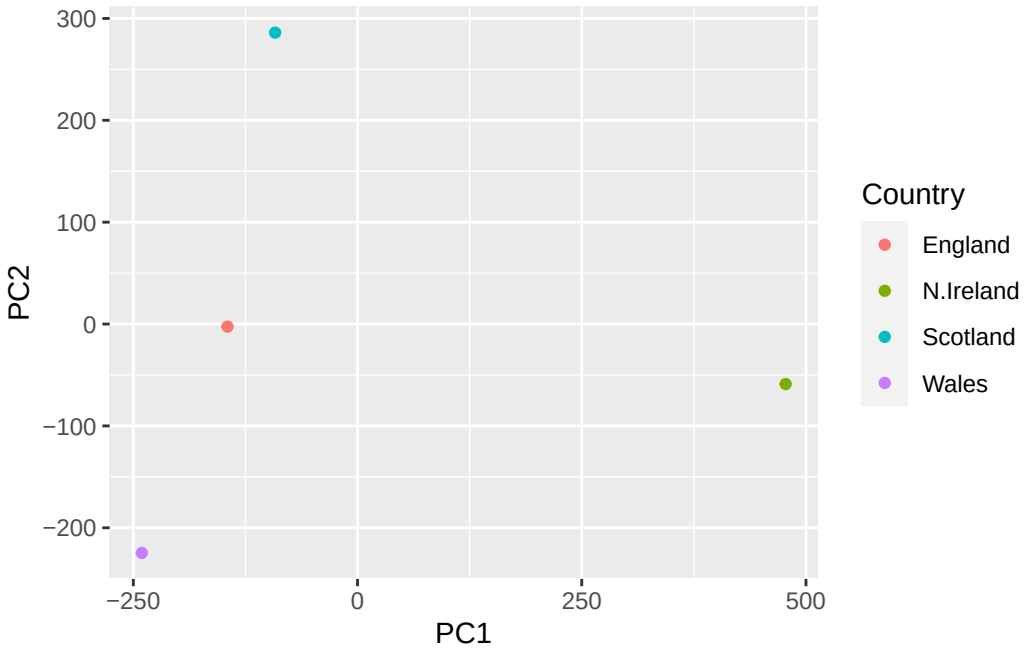
Q9: Generate a similar 'loadings plot' for PC2. What two food groups feature prominantely and what does PC2 maninly tell us about?

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



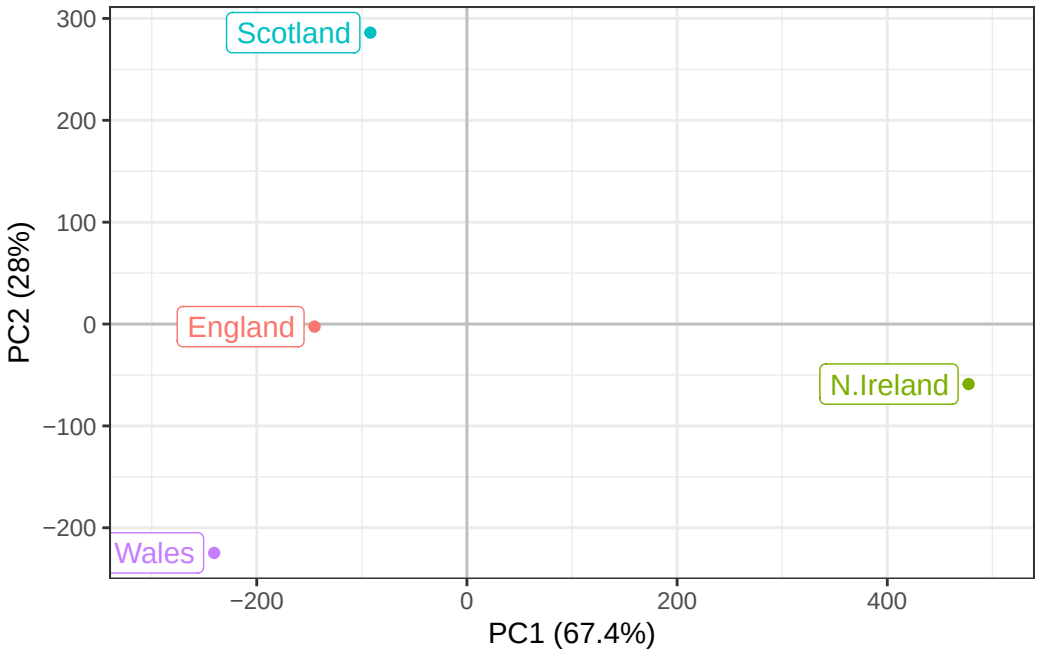
##Using ggplot for these figures

```
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()
```



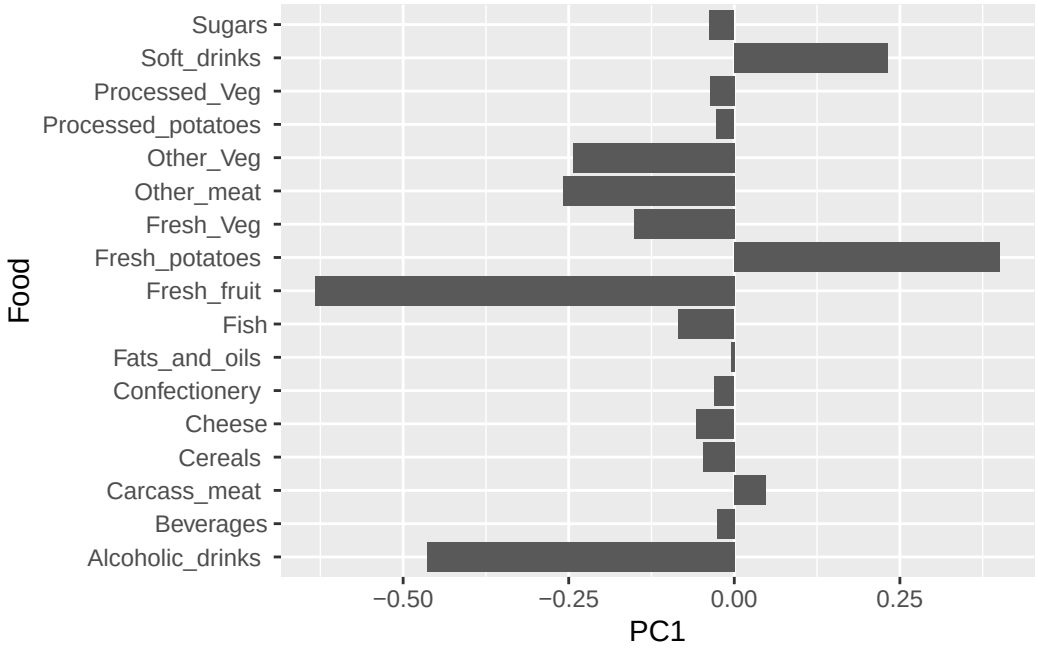
To change the appearance:

```
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()
```



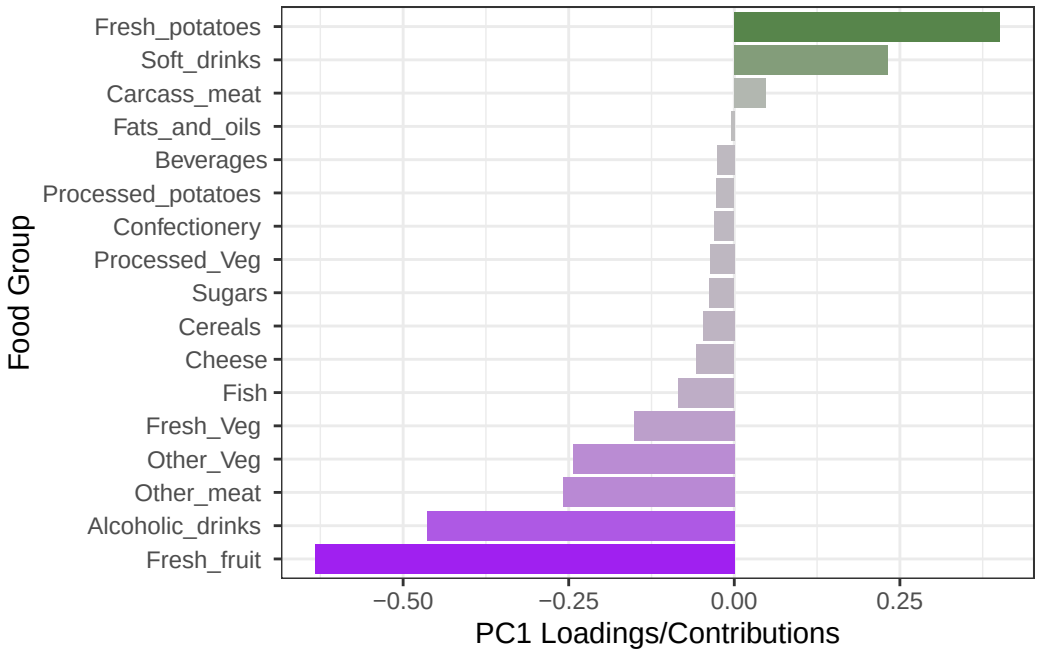
To load figures using `pca$rotation` Which do you prefer, base graphics or ggplot?

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

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



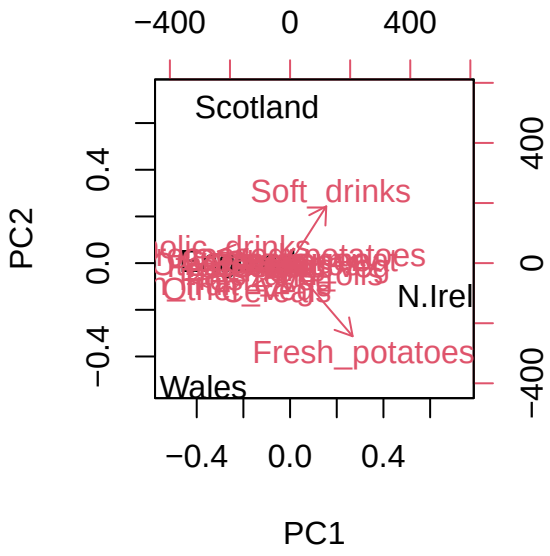
To add some additional features:

```
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()
```



Biplots

```
## The inbuilt biplot() can be useful for small datasets
biplot(pca)
```



##2. PCA of 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
```

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

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

```
[1] 100
```

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

```
[1] 10
```

There are 100 genes and 10 samples in this data set.