Class 7: Machine Learning 1

Emma Bell(A19247017)

Background

Today we will begin our exploration of some important machine learning methods, namely **clusterring** and **dimensionality reduction**.

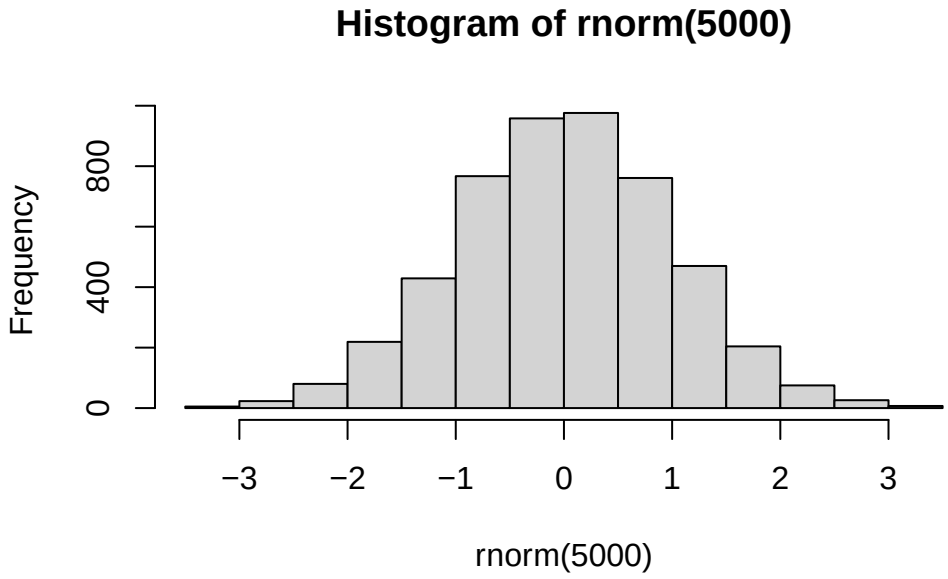
Let's make up some input data for clustering where we know what the natural "clusters" are.

The function `rnorm()` can be useful here.

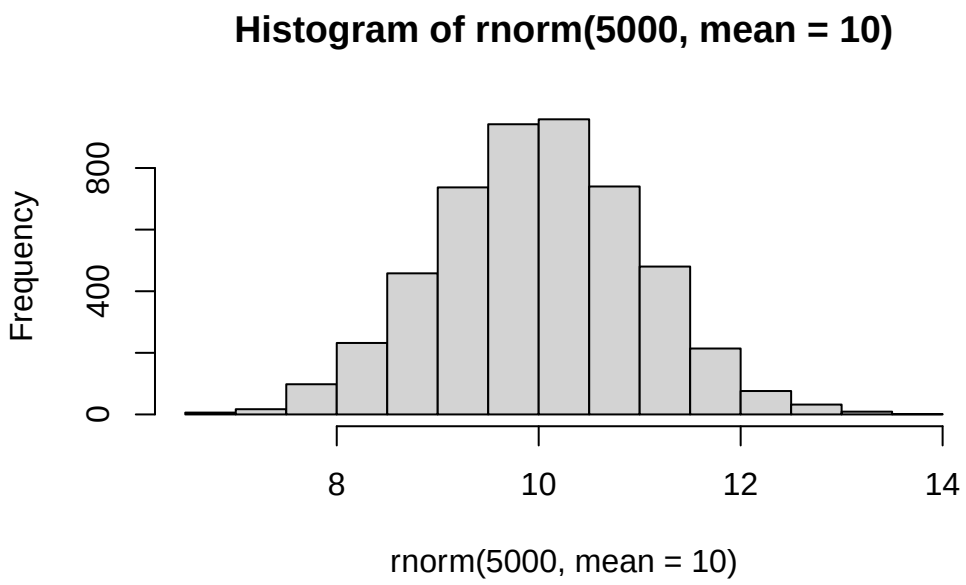
```
rnorm(5)
```

```
[1] -0.35464184 -0.68784544  0.70095536  1.42654408  0.02732219
```

```
hist( rnorm(5000))
```



```
hist( rnorm(5000, mean=10))
```



Q. Generate 30 random numbers centered at +3 and -3.

```
rnorm(30, mean=3)
```

```
 [1] 1.683726 2.785198 3.088022 2.359397 1.757295 3.229118 2.238869 4.900394
 [9] 2.076701 3.834335 4.964922 2.705840 2.009074 3.687983 3.102015 2.809968
[17] 1.679533 2.402457 4.335194 5.331991 2.496954 2.734944 2.115644 3.293312
[25] 3.264640 3.712311 4.006686 3.578342 2.444246 2.041695
```

```
rnorm(30, mean=-3)
```

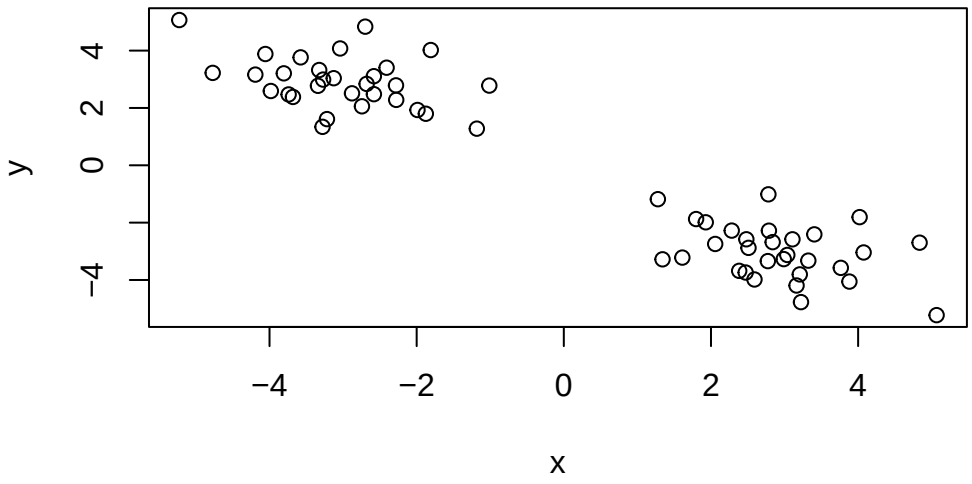
```
 [1] -3.9549624 -3.2821075 -4.8522475 -3.7311254 -3.6334274 -3.4315021
 [7] -4.3240098 -4.5211905 -2.1060477 -4.7487269 -2.0540998 -1.9330294
[13] -3.2766685 -1.6548844 -3.4707809 -3.7285379 -4.0414273 -4.9580103
[19] -4.0292854 -4.6379909 -3.2732007 -1.5040238 -2.1693594 -0.2577144
[25] -2.0925900 -4.7317803 -4.3389915 -4.1363585 -1.8477015 -2.6666854
```

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

```
 [1]  5.066436  2.780148  3.037285  4.835080  2.787264  2.590982  2.509942
 [8]  3.163016  3.323021  4.020269  1.928884  1.797708  2.471486  2.480552
[15]  4.074525  3.107043  2.057426  1.610833  3.764420  3.881274  2.772734
[22]  2.838263  2.281942  2.988901  1.341596  3.404392  3.206764  1.278191
[29]  2.383444  3.223191 -4.771836 -3.681312 -1.182434 -3.805680 -2.408500
[36] -3.279336 -3.271500 -2.278041 -2.679893 -3.341420 -4.056349 -3.576806
[43] -3.219059 -2.743826 -2.581608 -3.040220 -2.581172 -3.739835 -1.875368
[50] -1.988594 -1.808320 -3.324649 -4.192585 -2.878978 -3.981978 -2.280705
[57] -2.699189 -3.127215 -1.012931 -5.226243
```

(rev=reverse), (cbind= puts stuff in columns, rbind puts into rows)

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



K-means clustering

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

```
km <- kmeans(x, 2)
km
```

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

Cluster means:
          x         y
1  2.900234 -3.021186
2 -3.021186  2.900234

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 2

Within cluster sum of squares by cluster:
[1] 51.43469 51.43469
 (between_SS / total_SS =  91.1 %)

Available components:

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

Q. What component of the results object details the cluster sizes? size!

```
km$size
```

```
[1] 30 30
```

Q. What component of the results object details the cluster centres?

```
km$centers
```

```
          x         y
1  2.900234 -3.021186
2 -3.021186  2.900234
```

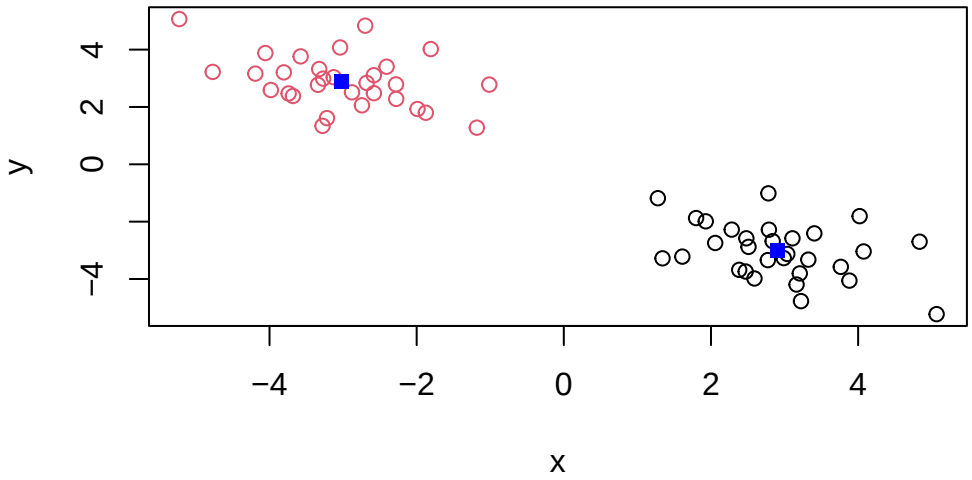
Q. What component of the result objects details the cluster membership vector (ie our main result which points lie in which cluster)

```
km$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 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. Plout our clustering results with points coloured by cluster and also add the cluster centres as new points coloured blue

```
plot(x, col= km$cluster)
points(km$centers, col="blue", pch=15)
```

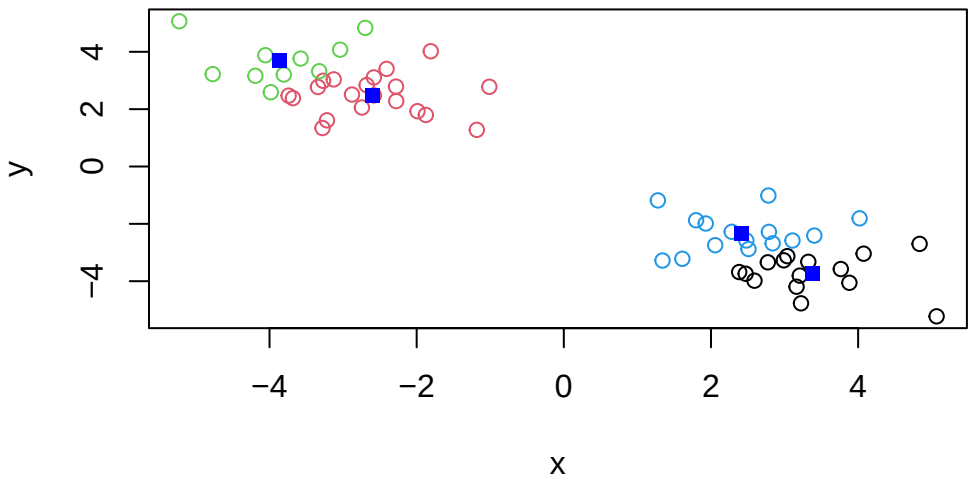


Q. run `kmeans()` again and this time produce four clusters (and call your result object `k4`) and make a results figure like above

```
k4 <- kmeans(x, 4)
k4
```

```
K-means clustering with 4 clusters of sizes 15, 20, 10, 15

Cluster means:
          x         y
1  3.385504 -3.722455
2 -2.598002  2.493915
3 -3.867554  3.712871
4  2.414963 -2.319918

Clustering vector:
 [1] 1 4 1 1 4 1 4 1 1 4 4 4 1 4 1 4 4 4 1 1 1 4 4 1 4 4 1 4 1 1 3 2 2 3 2 2 2 2
[39] 2 2 3 3 2 2 2 3 2 2 2 2 2 3 3 2 3 2 3 2 2 3

Within cluster sum of squares by cluster:
[1] 15.19367 20.07093 10.71300 14.42308
 (between_SS / total_SS =  94.8 %)

Available components:

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

```r
plot(x, col=k4$cluster)
points(k4$centers, col="blue", pch=15)
```



The metric

```
km$tot.withinss
```

```
[1] 102.8694
```

```
k4$tot.withinss
```

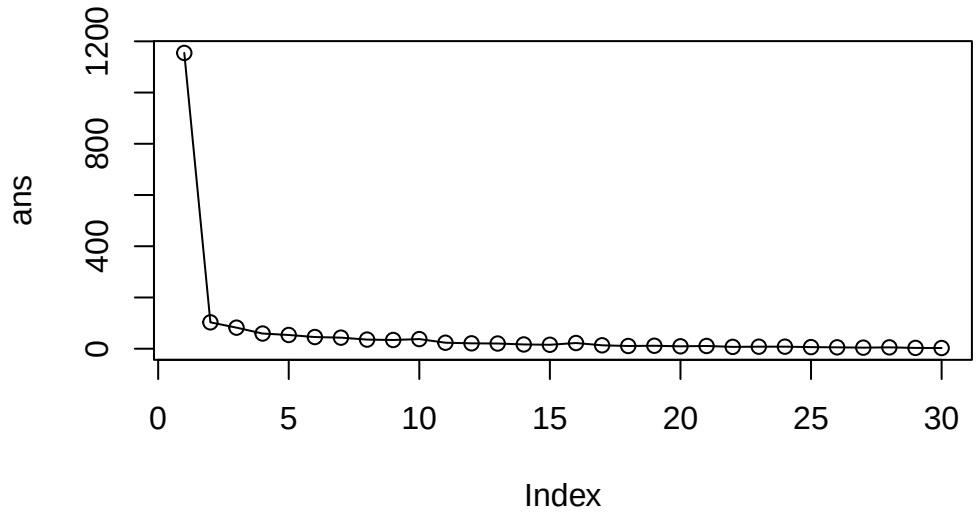
```
[1] 60.40069
```

Q. Let's try different number of K (centers) from 1 to 30 and see what the best result is

```
ans <- NULL
for(i in 1:30){
ans <- c(ans, kmeans(x, centers= i)$tot.withinss)
}
ans
```

```
 [1] 1154.765775  102.869386   82.218628   59.233510   53.576565   45.585260
 [7]   43.210544   35.552846   34.050814   37.501003   23.377940   20.966250
[13]   20.261084   16.571372   15.381341   22.285216   13.274248   10.601875
[19]   11.528190    9.332313   10.679536    7.092023    7.876142    7.955505
[25]    6.211283    5.398248    4.098433    5.259895    3.128570    2.629404
```

```
plot(ans, typ="o")
```



best result is 2- the 'elbow point, meaning the best number of clusters is 2. **Key-Point**: K means will impose a clustering structure on your data even if it is not there- it iwll always give you the answer you asked for even if that answer is silly!

Hierarchal Clustering

The main function for this is called `hclust()`.

Unlike `kmeans()` (which does all the work for you) you can't just pass `hclust()` our raw input data. It needs a "distance matrix" like the one returned from the `dist()` function.

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

```
           1         2         3         4         5         6         7
2  4.7936531
3  2.9194817 2.1298627
4  2.5376224 2.6582343 1.8480453
5  3.7243550 1.2677937 0.8826605 2.0901382
6  2.7705718 2.9750669 0.9642647 2.5848639 1.7125586
```

7	3.4706364	1.8855089	0.5828488	2.3320789	0.6594232	1.1059728	
8	2.1659773	3.2026222	1.0727640	2.2418810	1.9484549	0.6095719	1.4669933
9	2.5798365	2.3746052	0.3473115	1.6363130	1.1733945	0.9838512	0.9272114
10	3.5744461	1.4732765	1.6449142	1.2072962	1.3203977	2.6014713	1.8513240
11	4.5085032	1.2948239	1.5890277	2.9918083	0.9067218	2.1004647	1.0632079
12	4.6811270	1.3072818	1.7617244	3.1471112	1.0693548	2.2510199	1.2306546
13	2.9905147	2.7443169	0.8339257	2.5825414	1.4929085	0.2700237	0.8617148
14	3.6990809	1.5966015	0.7798172	2.3574836	0.4293631	1.4051524	0.2992533
15	2.4005381	2.4052677	1.0408815	0.8335143	1.4946250	1.7572155	1.5728700
16	3.2914002	1.6023757	0.5500481	1.7320323	0.4390915	1.4924325	0.6670527
17	3.9008382	1.8757201	1.0521938	2.7780129	0.8643759	1.3482220	0.4722677
18	3.9962463	2.4968575	1.4294061	3.2658893	1.5048252	1.2420704	0.9612758
19	2.1014018	2.7463147	0.8549018	1.3843856	1.6231794	1.2414189	1.4355067
20	1.6653114	3.2364903	1.2552319	1.6588032	2.0856100	1.2924334	1.8074164
21	2.9687751	2.3285010	0.3403986	2.1600301	1.0608150	0.6658438	0.5318953
22	3.3835864	1.6679744	0.4895988	1.9969104	0.4024324	1.3253578	0.3839659
23	4.0552812	1.3596736	1.1365034	2.5876397	0.5053294	1.7317355	0.6427361
24	2.8525731	2.2681956	0.1521816	1.9328518	1.0111045	0.8143207	0.6192539
25	4.2029610	2.6844041	1.7024990	3.5413275	1.7570491	1.4334125	1.2350377
26	3.2714015	1.5288206	0.8070434	1.4599203	0.6302210	1.7712904	1.0106395
27	2.3401662	2.8251460	0.6993130	1.9686885	1.5816228	0.6405220	1.1594562
28	5.5410464	1.5114913	2.6223244	3.8667820	1.8664139	3.0920654	2.0965383
29	3.0960072	2.6977084	0.8570485	2.6410385	1.4576594	0.3653386	0.8122443
30	1.8984309	3.7849246	1.6550953	2.6256528	2.5289855	1.0117132	2.0227791
31	12.9685982	8.6589374	10.0652884	11.2857174	9.3505544	10.3017258	9.5005661
32	11.5944139	7.2997143	8.6894711	9.9177661	7.9747577	8.9364574	8.1255589
33	9.0197584	4.5772586	6.1002975	7.2131839	5.3314391	6.4736426	5.5601883
34	12.2405093	7.8216985	9.3244555	10.4662802	8.5778340	9.6226457	8.7705955
35	11.4176413	6.8143093	8.5040263	9.4722303	7.7017070	8.9192648	7.9794618
36	10.6201899	6.5008577	7.7375694	9.0648604	7.0657418	7.9247134	7.1644252
37	11.7051174	7.2551434	8.7867880	9.9030805	8.0297802	9.1083397	8.2375406
38	10.5030560	6.0366766	7.5836447	8.6837866	6.8172619	7.9337304	7.0398774
39	11.1822121	6.6815972	8.2627506	9.3347836	7.4895684	8.6196176	7.7214782
40	11.6049856	7.1975595	8.6889173	9.8385517	7.9434263	8.9899688	8.1351271
41	12.8907748	8.4077901	9.9718885	11.0616453	9.2089643	10.2964919	9.4242868
42	12.4714737	7.9519776	9.5520067	10.6083958	8.7775237	9.9019338	9.0101740
43	10.7422081	6.5478715	7.8479893	9.1348473	7.1568136	8.0644969	7.2787258
44	10.6795145	6.3199196	7.7654199	8.9479159	7.0293909	8.0581994	7.2090332
45	11.3108904	6.7618502	8.3927311	9.4191077	7.6060909	8.7755292	7.8585197
46	12.3378347	7.7303880	9.4234351	10.3876635	8.6225589	9.8294294	8.8969380
47	10.8572832	6.3990761	7.9381426	9.0460214	7.1756304	8.2774073	7.3921581
48	11.6963857	7.3926542	8.7906088	10.0132430	8.0738055	9.0402676	8.2272213
49	9.8754510	5.4381542	6.9562220	8.0778827	6.1946419	7.3043174	6.4111102

```
50 10.0483484  5.6031405  7.1290676  8.2450950  6.3662757  7.4773536  6.5842516
51 11.5221639  6.8108108  8.6351849  9.4491204  7.7988248  9.1318028  8.1392260
52 11.9791582  7.4879255  9.0597876 10.1414095  8.2919978  9.3998780  8.5151186
53 12.4943647  8.1275794  9.5832162 10.7640227  8.8516889  9.8522956  9.0238330
54 11.0895521  6.6660592  8.1718852  9.3081542  7.4200133  8.4891395  7.6210837
55 11.9575421  7.6625415  9.0536089 10.2823354  8.3400225  9.2955696  8.4891395
56 10.8718343  6.3288006  7.9537463  8.9852969  7.1671906  8.3400225  7.4200133
57 12.7096480  8.0138859  9.8135250 10.6550651  8.9852969 10.2823354  9.3081542
58 11.6370880  7.1624849  8.7179196  9.8135250  7.9537463  9.0536089  8.1718852
59 10.0529104  5.3642241  7.1624849  8.0138859  6.3288006  7.6625415  6.6660592
60 14.5560467 10.0529104 11.6370880 12.7096480 10.8718343 11.9575421 11.0895521
             8          9         10         11         12         13         14
2
3
4
5
6
7
8
9   0.8825620
10  2.5336943  1.6689544
11  2.5259962  1.9309737  2.0991405
12  2.6895284  2.1040428  2.2235726  0.1732845
13  0.8265574  0.9473607  2.4757790  1.8333736  1.9824765
14  1.7499745  1.1236159  1.7227972  0.8096204  0.9820566  1.1586984
15  1.4692834  0.8035279  1.2330949  2.3894959  2.5574943  1.7490554  1.6587571
16  1.6119493  0.7737932  1.1966443  1.3189857  1.4876606  1.3211442  0.6264912
17  1.8224250  1.3925113  2.1743801  0.7660927  0.9064620  1.0786467  0.4533128
18  1.8322190  1.7154414  2.7920548  1.2709049  1.3566236  1.0059478  1.0785691
19  0.8607389  0.5083463  1.7868974  2.4272632  2.6005476  1.3031721  1.6246859
20  0.7310644  0.9203429  2.2523221  2.8438414  3.0162768  1.4448821  2.0342492
21  0.9363769  0.5505426  1.9765480  1.5944343  1.7606778  0.4994844  0.8144620
22  1.5471598  0.8066605  1.4685980  1.1423058  1.3153002  1.1216069  0.3710832
23  2.1075512  1.4762231  1.8006724  0.4565401  0.6297841  1.4740312  0.3624008
24  0.9373976  0.3383213  1.7901447  1.6641761  1.8352455  0.6978938  0.8573049
25  2.0375461  1.9819435  3.0560076  1.4180700  1.4761999  1.2201269  1.3359098
26  1.8003400  0.9197555  0.8599541  1.5340938  1.6928272  1.6256590  0.9398384
27  0.3893704  0.4948810  2.1566736  2.2214362  2.3898842  0.7382210  1.4236593
28  3.5515598  2.9614888  2.8126014  1.0360003  0.8660573  2.8221005  1.8444890
29  0.9322731  1.0049949  2.4874281  1.7526885  1.8985576  0.1057180  1.1044177
30  0.5823681  1.4506267  3.0688378  3.0694730  3.2282396  1.2767489  2.3131198
31 10.8607373 10.4115764 10.1300158  8.4889553  8.3159011 10.0473597  9.2891205
32  9.4915217  9.0356520  8.7684227  7.1125959  6.9395343  8.6805219  7.9130257
```

33	6.9865820	6.4409053	6.0493528	4.5113399	4.3389145	6.2073916	5.3209160
34	10.1643054	9.6683890	9.2949749	7.7380221	7.5647887	9.3625880	8.5449940
35	9.4210325	8.8391360	8.2765595	6.9207806	6.7504110	8.6518342	7.7285058
36	8.4930011	8.0848605	7.9502336	6.1818874	6.0104273	7.6741706	6.9688178
37	9.6424448	9.1295467	8.7283004	7.1985733	7.0255235	8.8463421	8.0069852
38	8.4572219	7.9250083	7.5099236	5.9946392	5.8219371	7.6694065	6.8039753
39	9.1417945	8.6032817	8.1537048	6.6737827	6.5013246	8.3551221	7.4833455
40	9.5301296	9.0328768	8.6706708	7.1025603	6.9293213	8.7294577	7.9094673
41	10.8308095	10.3140888	9.8794352	8.3832197	8.2102927	10.0346362	9.1920769
42	10.4278057	9.8925483	9.4218353	7.9630270	7.7905459	9.6382400	8.7725777
43	8.6261544	8.1949857	8.0061524	6.2814962	6.1091360	7.8110140	7.0752007
44	8.5996175	8.1100779	7.7908264	6.1809790	6.0076948	7.7979296	6.9864665
45	9.2889871	8.7310543	8.2307702	6.8051498	6.6334627	8.5096118	7.6145423
46	10.3356303	9.7590274	9.1901238	7.8392230	7.6685575	9.5625896	8.6473826
47	8.8049108	8.2800877	7.8723215	6.3493209	6.1764573	8.0138978	7.1583584
48	9.5947481	9.1366904	8.8620511	7.2129530	7.0398519	8.7841336	8.0138978
49	7.8274469	7.2980674	6.9110038	5.3673476	5.1945136	7.0398519	6.1764573
50	8.0007175	7.4708013	7.0762376	5.5401516	5.3673476	7.2129530	6.3493209
51	9.6002688	8.9598261	8.2428697	7.0762376	6.9110038	8.8620511	7.8723215
52	9.9284509	9.4012253	8.9598261	7.4708013	7.2980674	9.1366904	8.2800877
53	10.4023911	9.9284509	9.6002688	8.0007175	7.8274469	9.5947481	8.8049108
54	9.0238330	8.5151186	8.1392260	6.5842516	6.4111102	8.2272213	7.3921581
55	9.8522956	9.3998780	9.1318028	7.4773536	7.3043174	9.0402676	8.2774073
56	8.8516889	8.2919978	7.7988248	6.3662757	6.1946419	8.0738055	7.1756304
57	10.7640227	10.1414095	9.4491204	8.2450950	8.0778827	10.0132430	9.0460214
58	9.5832162	9.0597876	8.6351849	7.1290676	6.9562220	8.7906088	7.9381426
59	8.1275794	7.4879255	6.8108108	5.6031405	5.4381542	7.3926542	6.3990761
60	12.4943647	11.9791582	11.5221639	10.0483484	9.8754510	11.6963857	10.8572832

	15	16	17	18	19	20	21
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							

16	1.0706756						
17	2.0387593	1.0620790					
18	2.4701745	1.6263421	0.6521434				
19	0.6197491	1.1927124	1.8993907	2.1830990			
20	1.0343418	1.6656211	2.2470289	2.4199087	0.4935751		
21	1.3361812	0.8301066	0.9320865	1.1683268	1.0192384	1.3190842	
22	1.2877035	0.2861867	0.7834501	1.3406282	1.2892713	1.7269927	0.6647651
23	1.9478891	0.8791730	0.5170721	1.1558122	1.9709216	2.3917029	1.1711759
24	1.1099864	0.6999346	1.0705541	1.3790656	0.8334516	1.1884096	0.2271934
25	2.7433697	1.8983225	0.8939705	0.2759019	2.4410171	2.6558827	1.4324843
26	0.9209500	0.3440683	1.3880789	1.9682127	1.2225217	1.7154658	1.1266466
27	1.1571247	1.2281276	1.5647728	1.7003301	0.6027965	0.7195819	0.6355466
28	3.3572092	2.3026916	1.7450366	2.0636114	3.4517172	3.8775544	2.6258108
29	1.8085225	1.3164135	0.9925560	0.9003365	1.3849249	1.5440687	0.5167914
30	1.9295760	2.1933056	2.3391950	2.2384853	1.3118792	0.9721090	1.4996669
31	10.8392078	9.7863390	9.0688538	9.0686848	10.9136440	11.3078741	10.0007332
32	9.4640974	8.4103214	7.6955836	7.7068043	9.5374788	9.9329571	8.6272118
33	6.8032560	5.7704127	5.1646212	5.2941094	6.9312595	7.3551651	6.0814607
34	10.0559661	9.0161523	8.3537740	8.4041548	10.1648042	10.5755417	9.2819321
35	9.1412602	8.1396199	7.5990182	7.7475891	9.3189089	9.7582739	8.5059492
36	8.5603500	7.4951567	6.7209898	6.6868025	8.5909943	8.9672823	7.6523331
37	9.5033882	8.4685527	7.8269796	7.8978439	9.6235480	10.0398069	8.7524703
38	8.2873701	7.2562558	6.6373650	6.7367905	8.4164279	8.8380478	7.5586065
39	8.9542572	7.9286586	7.3213320	7.4230364	9.0929793	9.5175931	8.2413363
40	9.4226687	8.3816266	7.7188067	7.7734417	9.5294086	9.9400046	8.6467204
41	10.6779302	9.6479547	9.0149981	9.0846945	10.8068211	11.2254939	9.9400046
42	10.2394332	9.2166106	8.6082422	8.6994581	10.3820612	10.8068211	9.5294086
43	8.6503560	7.5892112	6.8413699	6.8304982	8.6994581	9.0846945	7.7734417
44	8.5132783	7.4668136	6.7899954	6.8413699	8.6082422	9.0149981	7.7188067
45	9.0605163	8.0449677	7.4668136	7.5892112	9.2166106	9.6479547	8.3816266
46	10.0617693	9.0605163	8.5132783	8.6503560	10.2394332	10.6779302	9.4226687
47	8.6473826	7.6145423	6.9864665	7.0752007	8.7725777	9.1920769	7.9094673
48	9.5625896	8.5096118	7.7979296	7.8110140	9.6382400	10.0346362	8.7294577
49	7.6685575	6.6334627	6.0076948	6.1091360	7.7905459	8.2102927	6.9293213
50	7.8392230	6.8051498	6.1809790	6.2814962	7.9630270	8.3832197	7.1025603
51	9.1901238	8.2307702	7.7908264	8.0061524	9.4218353	9.8794352	8.6706708
52	9.7590274	8.7310543	8.1100779	8.1949857	9.8925483	10.3140888	9.0328768
53	10.3356303	9.2889871	8.5996175	8.6261544	10.4278057	10.8308095	9.5301296
54	8.8969380	7.8585197	7.2090332	7.2787258	9.0101740	9.4242868	8.1351271
55	9.8294294	8.7755292	8.0581994	8.0644969	9.9019338	10.2964919	8.9899688
56	8.6225589	7.6060909	7.0293909	7.1568136	8.7775237	9.2089643	7.9434263
57	10.3876635	9.4191077	8.9479159	9.1348473	10.6083958	11.0616453	9.8385517
58	9.4234351	8.3927311	7.7654199	7.8479893	9.5520067	9.9718885	8.6889173

```
59  7.7303880  6.7618502  6.3199196  6.5478715  7.9519776  8.4077901  7.1975595
60 12.3378347 11.3108904 10.6795145 10.7422081 12.4714737 12.8907748 11.6049856
          22         23         24         25         26         27         28
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23  0.6862783
24  0.6104842  1.2193245
25  1.6122480  1.3736241  1.6473236
26  0.6278192  1.1300063  0.9578113  2.2390813
27  1.1845637  1.7857713  0.5768996  1.9380121  1.4110885
28  2.1624539  1.4858905  2.7001344  2.0978611  2.4543774  3.2558897
29  1.0998638  1.4069372  0.7311116  1.1167055  1.6316821  0.8326609  2.7323938
30  2.1270630  2.6655140  1.5185192  2.4016567  2.3702729  0.9662954  4.0825034
31  9.6311994  8.9453527 10.1197851  8.9250723  9.9280738 10.6331133  7.4841405
32  8.2548308  7.5690049  8.7447202  7.5694574  8.5539421  9.2601757  6.1083675
33  5.6420234  4.9647442  6.1724973  5.2097777  5.8847822  6.7164581  3.4798491
34  8.8766386  8.1940364  9.3879841  8.2803356  9.1387273  9.9170945  6.7164581
35  8.0341172  7.3681943  8.5848396  7.6639054  8.2206705  9.1387273  5.8847822
36  7.3210239  6.6354794  7.7827665  6.5349853  7.6639054  8.2803356  5.2097777
37  8.3345320  7.6538482  8.8535440  7.7827665  8.5848396  9.3879841  6.1724973
38  7.1271569  6.4487894  7.6538482  6.6354794  7.3681943  8.1940364  4.9647442
39  7.8038503  7.1271569  8.3345320  7.3210239  8.0341172  8.8766386  5.6420234
40  8.2413363  7.5586065  8.7524703  7.6523331  8.5059492  9.2819321  6.0814607
41  9.5175931  8.8380478 10.0398069  8.9672823  9.7582739 10.5755417  7.3551651
```

42	9.0929793	8.4164279	9.6235480	8.5909943	9.3189089	10.1648042	6.9312595
43	7.4230364	6.7367905	7.8978439	6.6868025	7.7475891	8.4041548	5.2941094
44	7.3213320	6.6373650	7.8269796	6.7209898	7.5990182	8.3537740	5.1646212
45	7.9286586	7.2562558	8.4685527	7.4951567	8.1396199	9.0161523	5.7704127
46	8.9542572	8.2873701	9.5033882	8.5603500	9.1412602	10.0559661	6.8032560
47	7.4833455	6.8039753	8.0069852	6.9688178	7.7285058	8.5449940	5.3209160
48	8.3551221	7.6694065	8.8463421	7.6741706	8.6518342	9.3625880	6.2073916
49	6.5013246	5.8219371	7.0255235	6.0104273	6.7504110	7.5647887	4.3389145
50	6.6737827	5.9946392	7.1985733	6.1818874	6.9207806	7.7380221	4.5113399
51	8.1537048	7.5099236	8.7283004	7.9502336	8.2765595	9.2949749	6.0493528
52	8.6032817	7.9250083	9.1295467	8.0848605	8.8391360	9.6683890	6.4409053
53	9.1417945	8.4572219	9.6424448	8.4930011	9.4210325	10.1643054	6.9865820
54	7.7214782	7.0398774	8.2375406	7.1644252	7.9794618	8.7705955	5.5601883
55	8.6196176	7.9337304	9.1083397	7.9247134	8.9192648	9.6226457	6.4736426
56	7.4895684	6.8172619	8.0297802	7.0657418	7.7017070	8.5778340	5.3314391
57	9.3347836	8.6837866	9.9030805	9.0648604	9.4722303	10.4662802	7.2131839
58	8.2627506	7.5836447	8.7867880	7.7375694	8.5040263	9.3244555	6.1002975
59	6.6815972	6.0366766	7.2551434	6.5008577	6.8143093	7.8216985	4.5772586
60	11.1822121	10.5030560	11.7051174	10.6201899	11.4176413	12.2405093	9.0197584

	29	30	31	32	33	34	35
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  1.3763787
31  9.9433491 11.3066757
32  8.5768599  9.9433491  1.3763787
33  6.1083675  7.4841405  4.0825034  2.7323938
34  9.2601757 10.6331133  0.9662954  0.8326609  3.2558897
35  8.5539421  9.9280738  2.3702729  1.6316821  2.4543774  1.4110885
36  7.5694574  8.9250723  2.4016567  1.1167055  2.0978611  1.9380121  2.2390813
37  8.7447202 10.1197851  1.5185192  0.7311116  2.7001344  0.5768996  0.9578113
38  7.5690049  8.9453527  2.6655140  1.4069372  1.4858905  1.7857713  1.1300063
39  8.2548308  9.6311994  2.1270630  1.0998638  2.1624539  1.1845637  0.6278192
40  8.6272118 10.0007332  1.4996669  0.5167914  2.6258108  0.6355466  1.1266466
41  9.9329571 11.3078741  0.9721090  1.5440687  3.8775544  0.7195819  1.7154658
42  9.5374788 10.9136440  1.3118792  1.3849249  3.4517172  0.6027965  1.2225217
43  7.7068043  9.0686848  2.2384853  0.9003365  2.0636114  1.7003301  1.9682127
44  7.6955836  9.0688538  2.3391950  0.9925560  1.7450366  1.5647728  1.3880789
45  8.4103214  9.7863390  2.1933056  1.3164135  2.3026916  1.2281276  0.3440683
46  9.4640974 10.8392078  1.9295760  1.8085225  3.3572092  1.1571247  0.9209500
47  7.9130257  9.2891205  2.3131198  1.1044177  1.8444890  1.4236593  0.9398384
48  8.6805219 10.0473597  1.2767489  0.1057180  2.8221005  0.7382210  1.6256590
49  6.9395343  8.3159011  3.2282396  1.8985576  0.8660573  2.3898842  1.6928272
50  7.1125959  8.4889553  3.0694730  1.7526885  1.0360003  2.2214362  1.5340938
51  8.7684227 10.1300158  3.0688378  2.4874281  2.8126014  2.1566736  0.8599541
52  9.0356520 10.4115764  1.4506267  1.0049949  2.9614888  0.4948810  0.9197555
53  9.4915217 10.8607373  0.5823681  0.9322731  3.5515598  0.3893704  1.8003400
54  8.1255589  9.5005661  2.0227791  0.8122443  2.0965383  1.1594562  1.0106395
55  8.9364574 10.3017258  1.0117132  0.3653386  3.0920654  0.6405220  1.7712904
56  7.9747577  9.3505544  2.5289855  1.4576594  1.8664139  1.5816228  0.6302210
57  9.9177661 11.2857174  2.6256528  2.6410385  3.8667820  1.9686885  1.4599203
58  8.6894711 10.0652884  1.6550953  0.8570485  2.6223244  0.6993130  0.8070434
59  7.2997143  8.6589374  3.7849246  2.6977084  1.5114913  2.8251460  1.5288206
60 11.5944139 12.9685982  1.8984309  3.0960072  5.5410464  2.3401662  3.2714015
          36         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	1.6473236							
38	1.3736241	1.2193245						
39	1.6122480	0.6104842	0.6862783					
40	1.4324843	0.2271934	1.1711759	0.6647651				
41	2.6558827	1.1884096	2.3917029	1.7269927	1.3190842			
42	2.4410171	0.8334516	1.9709216	1.2892713	1.0192384	0.4935751		
43	0.2759019	1.3790656	1.1558122	1.3406282	1.1683268	2.4199087	2.1830990	
44	0.8939705	1.0705541	0.5170721	0.7834501	0.9320865	2.2470289	1.8993907	
45	1.8983225	0.6999346	0.8791730	0.2861867	0.8301066	1.6656211	1.1927124	
46	2.7433697	1.1099864	1.9478891	1.2877035	1.3361812	1.0343418	0.6197491	
47	1.3359098	0.8573049	0.3624008	0.3710832	0.8144620	2.0342492	1.6246859	
48	1.2201269	0.6978938	1.4740312	1.1216069	0.4994844	1.4448821	1.3031721	
49	1.4761999	1.8352455	0.6297841	1.3153002	1.7606778	3.0162768	2.6005476	
50	1.4180700	1.6641761	0.4565401	1.1423058	1.5944343	2.8438414	2.4272632	

51	3.0560076	1.7901447	1.8006724	1.4685980	1.9765480	2.2523221	1.7868974
52	1.9819435	0.3383213	1.4762231	0.8066605	0.5505426	0.9203429	0.5083463
53	2.0375461	0.9373976	2.1075512	1.5471598	0.9363769	0.7310644	0.8607389
54	1.2350377	0.6192539	0.6427361	0.3839659	0.5318953	1.8074164	1.4355067
55	1.4334125	0.8143207	1.7317355	1.3253578	0.6658438	1.2924334	1.2414189
56	1.7570491	1.0111045	0.5053294	0.4024324	1.0608150	2.0856100	1.6231794
57	3.5413275	1.9328518	2.5876397	1.9969104	2.1600301	1.6588032	1.3843856
58	1.7024990	0.1521816	1.1365034	0.4895988	0.3403986	1.2552319	0.8549018
59	2.6844041	2.2681956	1.3596736	1.6679744	2.3285010	3.2364903	2.7463147
60	4.2029610	2.8525731	4.0552812	3.3835864	2.9687751	1.6653114	2.1014018
	43	44	45	46	47	48	49

44	0.6521434						
45	1.6263421	1.0620790					
46	2.4701745	2.0387593	1.0706756				
47	1.0785691	0.4533128	0.6264912	1.6587571			
48	1.0059478	1.0786467	1.3211442	1.7490554	1.1586984		
49	1.3566236	0.9064620	1.4876606	2.5574943	0.9820566	1.9824765	
50	1.2709049	0.7660927	1.3189857	2.3894959	0.8096204	1.8333736	0.1732845
51	2.7920548	2.1743801	1.1966443	1.2330949	1.7227972	2.4757790	2.2235726
52	1.7154414	1.3925113	0.7737932	0.8035279	1.1236159	0.9473607	2.1040428
53	1.8322190	1.8224250	1.6119493	1.4692834	1.7499745	0.8265574	2.6895284
54	0.9612758	0.4722677	0.6670527	1.5728700	0.2992533	0.8617148	1.2306546
55	1.2420704	1.3482220	1.4924325	1.7572155	1.4051524	0.2700237	2.2510199
56	1.5048252	0.8643759	0.4390915	1.4946250	0.4293631	1.4929085	1.0693548
57	3.2658893	2.7780129	1.7320323	0.8335143	2.3574836	2.5825414	3.1471112
58	1.4294061	1.0521938	0.5500481	1.0408815	0.7798172	0.8339257	1.7617244
59	2.4968575	1.8757201	1.6023757	2.4052677	1.5966015	2.7443169	1.3072818
60	3.9962463	3.9008382	3.2914002	2.4005381	3.6990809	2.9905147	4.6811270
	50	51	52	53	54	55	56

```
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  2.0991405
52  1.9309737  1.6689544
53  2.5259962  2.5336943  0.8825620
54  1.0632079  1.8513240  0.9272114  1.4669933
55  2.1004647  2.6014713  0.9838512  0.6095719  1.1059728
56  0.9067218  1.3203977  1.1733945  1.9484549  0.6594232  1.7125586
57  2.9918083  1.2072962  1.6363130  2.2418810  2.3320789  2.5848639  2.0901382
58  1.5890277  1.6449142  0.3473115  1.0727640  0.5828488  0.9642647  0.8826605
59  1.2948239  1.4732765  2.3746052  3.2026222  1.8855089  2.9750669  1.2677937
```

```
60  4.5085032  3.5744461  2.5798365  2.1659773  3.4706364  2.7705718  3.7243550
           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
```

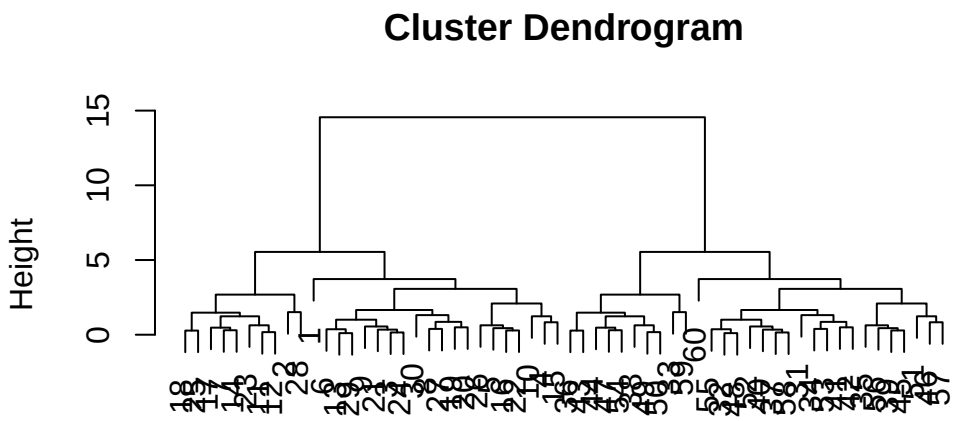
```
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58  1.8480453
59  2.6582343  2.1298627
60  2.5376224  2.9194817  4.7936531
```

ie distance between point 1 and 2 is x, distance between point 1 and 3 is y etc.

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

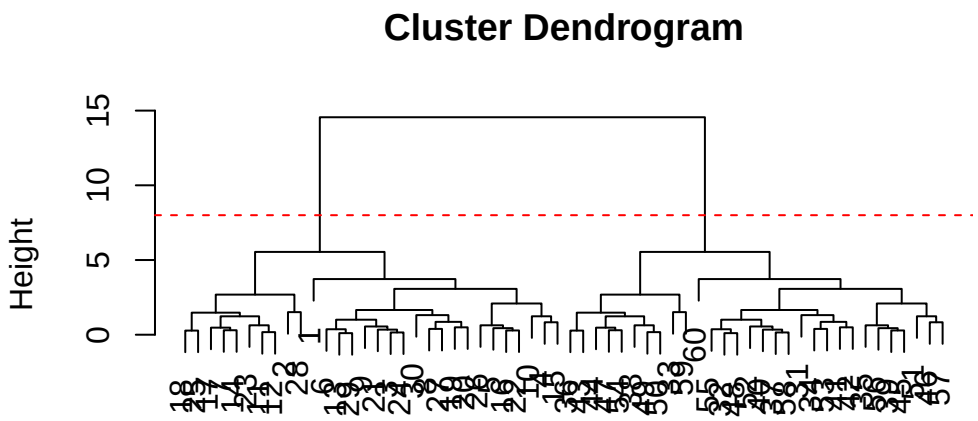
```
Call:
hclust(d = d)

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

```
plot(hc)
```



To extract our cluster membership vector from a `hclust()` result object we have to "cut" our tree at a given height to yield separate "groups/branches".

```
plot(hc)
abline(h=8, col="red", lty=2)
```



to do this we use the `cutree()` function on our `hclust()` object:

```
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 2 2
```

```
table(grps, km$cluster)
```

```
grps  1  2
   1 30  0
   2  0 30
```

PCA of UK food data

Import the dataset of food consumption in the UK

```
url <- "https://tinyurl.com/UK-foods"
x <- read.csv(url)
x
```

```
                    X England Wales Scotland N.Ireland
1              Cheese     105   103      103        66
2        Carcass_meat     245   227      242       267
3          Other_meat     685   803      750       586
4                Fish     147   160      122        93
5       Fats_and_oils     193   235      184       209
6               Sugars     156   175      147       139
7      Fresh_potatoes     720   874      566      1033
8           Fresh_Veg     253   265      171       143
9           Other_Veg     488   570      418       355
10 Processed_potatoes     198   203      220       187
11      Processed_Veg     360   365      337       334
12        Fresh_fruit    1102  1137      957       674
13            Cereals    1472  1582     1462      1494
14          Beverages      57    73       53        47
15        Soft_drinks    1374  1256     1572      1506
16   Alcoholic_drinks     375   475      458       135
17      Confectionery      54    64       62        41
```

Q.

```
dim(x)
```

```
[1] 17  5
```

One solution is to set the row names is to do it by hands..

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

```
 [1] "Cheese"             "Carcass_meat "      "Other_meat "
 [4] "Fish"               "Fats_and_oils "     "Sugars"
 [7] "Fresh_potatoes "    "Fresh_Veg "         "Other_Veg "
[10] "Processed_potatoes " "Processed_Veg "     "Fresh_fruit "
[13] "Cereals "           "Beverages"          "Soft_drinks "
[16] "Alcoholic_drinks "  "Confectionery "
```

(computer went wierd and had to find the script.. thats why I might miss some annotations)

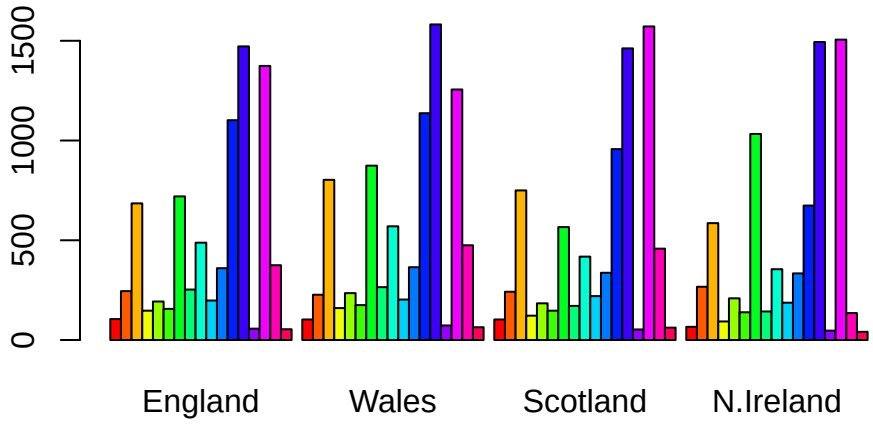
```
x <- read.csv(url, row.names=1)
x
```

```
                     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
```

Spotting major differences and trends

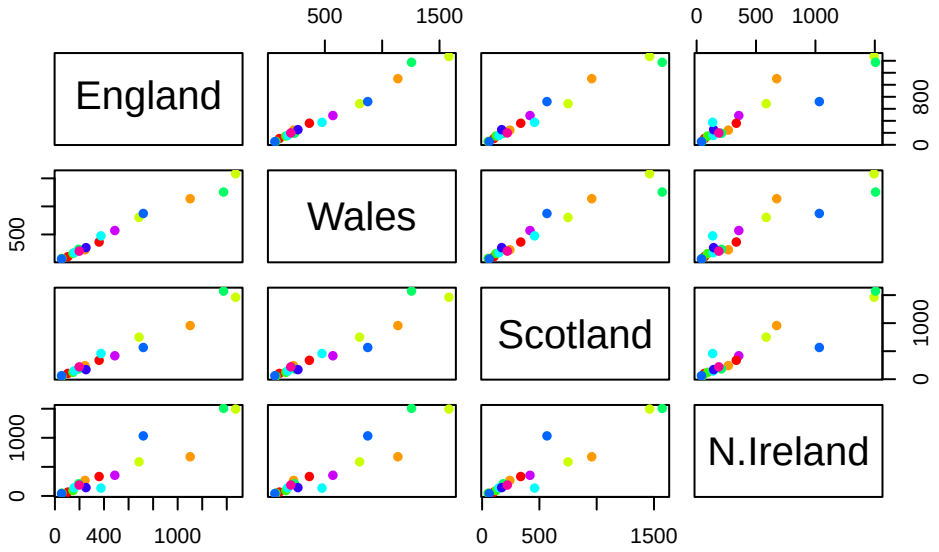
Is diffuclt even in this wee 17D dataset...

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



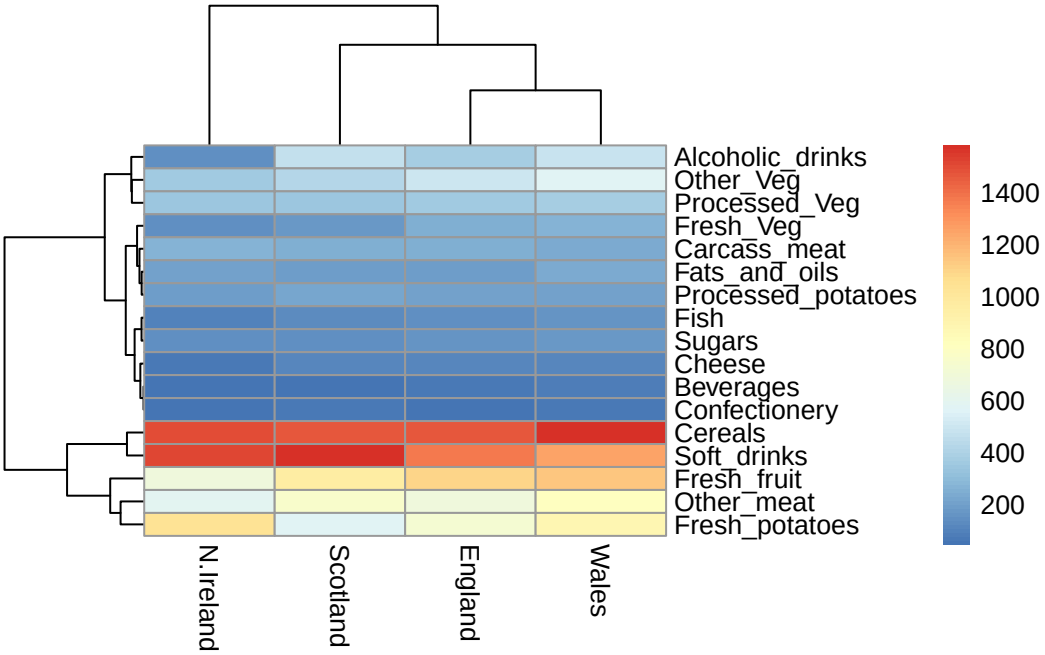
Pairs plot

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



Pheatmap- hotter is there is more stuff, colder means less

```
library(pheatmap)
pheatmap(as.matrix(x))
```



PCA to the rescue

The main PCA function in "base R" is called `prcomp()`. This function wants the transpose of our food data as input (ie the food as columsn and the countries as rows).

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

```
Standard deviations (1, .., p=4):
[1] 3.241502e+02 2.127478e+02 7.387622e+01 2.699876e-14

Rotation (n x k) = (17 x 4):
                            PC1           PC2         PC3           PC4
Cheese              -0.056955380  0.016012850  0.02394295  0.739145824
Carcass_meat         0.047927628  0.013915823  0.06367111  0.578851042
Other_meat          -0.258916658 -0.015331138 -0.55384854 -0.084756407
Fish                -0.084414983 -0.050754947  0.03906481  0.001282376
Fats_and_oils       -0.005193623 -0.095388656 -0.12522257  0.012073959
Sugars              -0.037620983 -0.043021699 -0.03605745  0.011712878
Fresh_potatoes       0.401402060 -0.715017078 -0.20668248  0.098706764
Fresh_Veg           -0.151849942 -0.144900268  0.21382237  0.067864113
Other_Veg           -0.243593729 -0.225450923 -0.05332841  0.017187324
Processed_potatoes  -0.026886233  0.042850761 -0.07364902  0.020275689
Processed_Veg       -0.036488269 -0.045451802  0.05289191 -0.013653986
Fresh_fruit         -0.632640898 -0.177740743  0.40012865  0.088466607
Cereals             -0.047702858 -0.212599678 -0.35884921  0.201601167
Beverages           -0.026187756 -0.030560542 -0.04135860 -0.004452115
Soft_drinks          0.232244140  0.555124311 -0.16942648  0.212426744
Alcoholic_drinks    -0.463968168  0.113536523 -0.49858320  0.032075763
Confectionery       -0.029650201  0.005949921 -0.05232164  0.035241822
```

```
summary(pca)
```

```
Importance of components:
                            PC1      PC2      PC3     PC4
Standard deviation     324.1502 212.7478 73.87622 2.7e-14
Proportion of Variance   0.6744   0.2905  0.03503 0.0e+00
Cumulative Proportion    0.6744   0.9650  1.00000 1.0e+00
```

```
attributes(pca)
```

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

$class
[1] "prcomp"
```

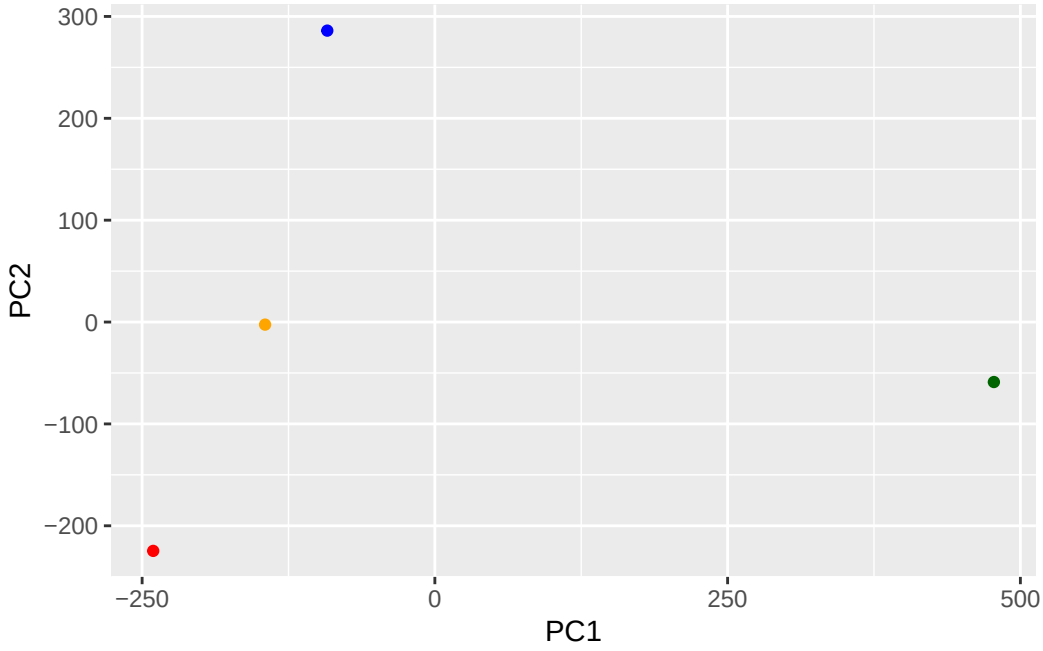
to make one of main PCA result figures we turn to `pca$x` the scores along our new PCs. This is called "PC plot" or "score plot" or "ordination plot"...

```
pca$x
```

```
                PC1         PC2        PC3           PC4
England   -144.99315   -2.532999 105.768945  1.612425e-14
Wales     -240.52915 -224.646925 -56.475555  4.751043e-13
Scotland   -91.86934  286.081786 -44.415495 -6.044349e-13
N.Ireland  477.39164  -58.901862  -4.877895  1.145386e-13
```

```
my_cols <- c("orange", "red", "blue", "darkgreen")
```

```
library(ggplot2)
ggplot(pca$x) +
  aes(PC1, PC2) +
  geom_point(col=my_cols)
```



the second major result figure is called a "loadings plot" of "variable contributions plot" or "weight plot"

```
pca$rotation
```

	PC1	PC2	PC3	PC4
Cheese	-0.056955380	0.016012850	0.02394295	0.739145824
Carcass_meat	0.047927628	0.013915823	0.06367111	0.578851042
Other_meat	-0.258916658	-0.015331138	-0.55384854	-0.084756407
Fish	-0.084414983	-0.050754947	0.03906481	0.001282376
Fats_and_oils	-0.005193623	-0.095388656	-0.12522257	0.012073959
Sugars	-0.037620983	-0.043021699	-0.03605745	0.011712878
Fresh_potatoes	0.401402060	-0.715017078	-0.20668248	0.098706764
Fresh_Veg	-0.151849942	-0.144900268	0.21382237	0.067864113
Other_Veg	-0.243593729	-0.225450923	-0.05332841	0.017187324
Processed_potatoes	-0.026886233	0.042850761	-0.07364902	0.020275689
Processed_Veg	-0.036488269	-0.045451802	0.05289191	-0.013653986
Fresh_fruit	-0.632640898	-0.177740743	0.40012865	0.088466607
Cereals	-0.047702858	-0.212599678	-0.35884921	0.201601167
Beverages	-0.026187756	-0.030560542	-0.04135860	-0.004452115
Soft_drinks	0.232244140	0.555124311	-0.16942648	0.212426744
Alcoholic_drinks	-0.463968168	0.113536523	-0.49858320	0.032075763
Confectionery	-0.029650201	0.005949921	-0.05232164	0.035241822

```
ggplot(pca$rotation) +
  aes(PC1, rownames(pca$rotation)) +
  geom_col()
```