Class 07: machine Learning 1

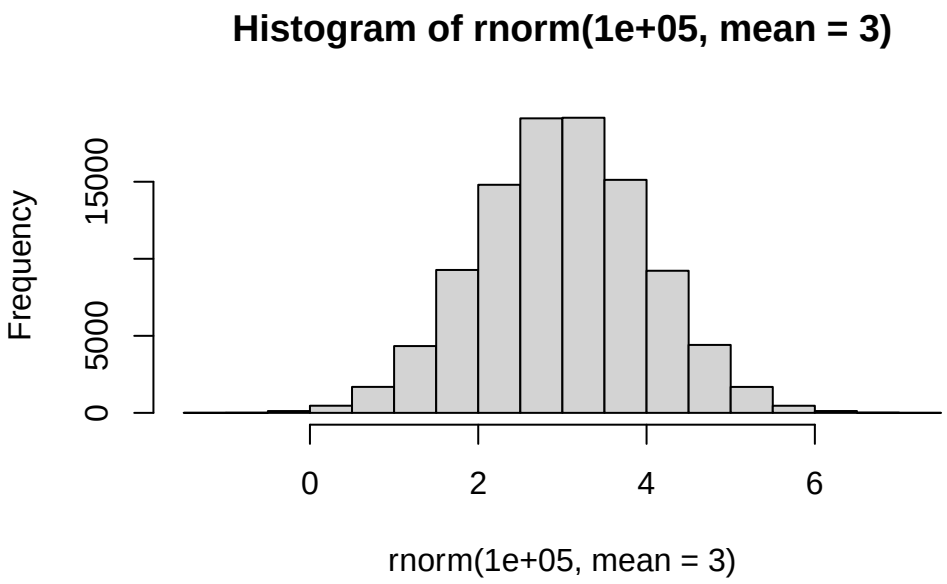
Pranati

Today we will start out multi-part exploration of some key machine learning methods. We will begin with clustering - finding groupings in data, and then dimensionallity reduction.

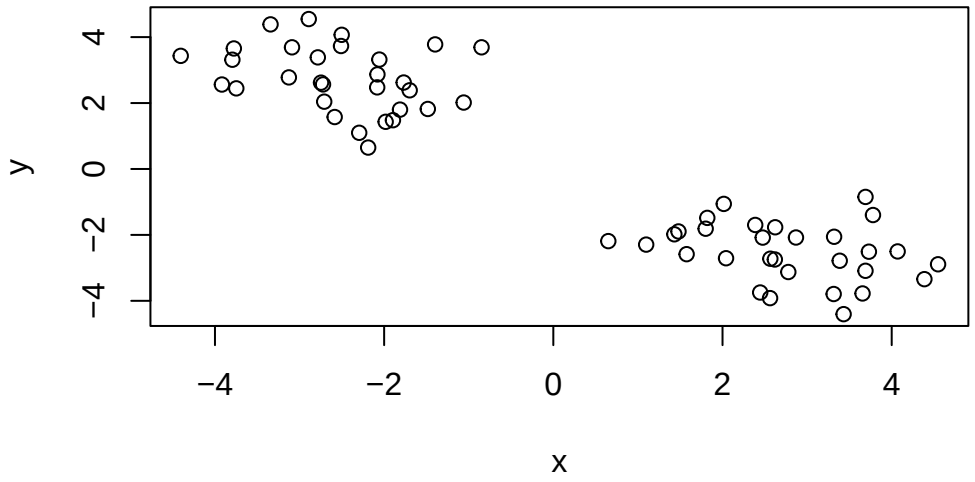
Clustering

Let's start with "k-means" clustering. The main function in base R for this `kmeans()`.

```
# Make up some data
hist(rnorm(100000, mean=3))
```



```r
tmp <- c(rnorm(30, -3), rnorm (30, +3))
x <- cbind(x=tmp, y=rev(tmp))
plot(x)
```



Now let's try out `kmeans()`

```r
km <- kmeans(x, centers=2)
km
```

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

Cluster means:
          x         y
1 -2.509966  2.741181
2  2.741181 -2.509966

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 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] 51.23545 51.23545
 (between_SS / total_SS =  89.0 %)

Available components:

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

Q. How many points in each cluster?

```
km$size
```

```
[1] 30 30
```

Q. What component of your result object details cluster assignment/membership?

```
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 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
```

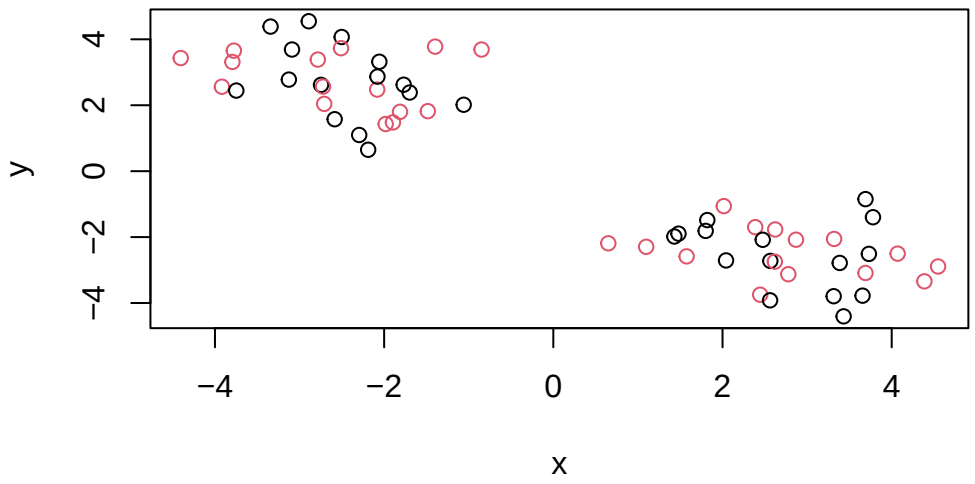
Q. What are centers/mean values of each cluster?

```
km$centers
```

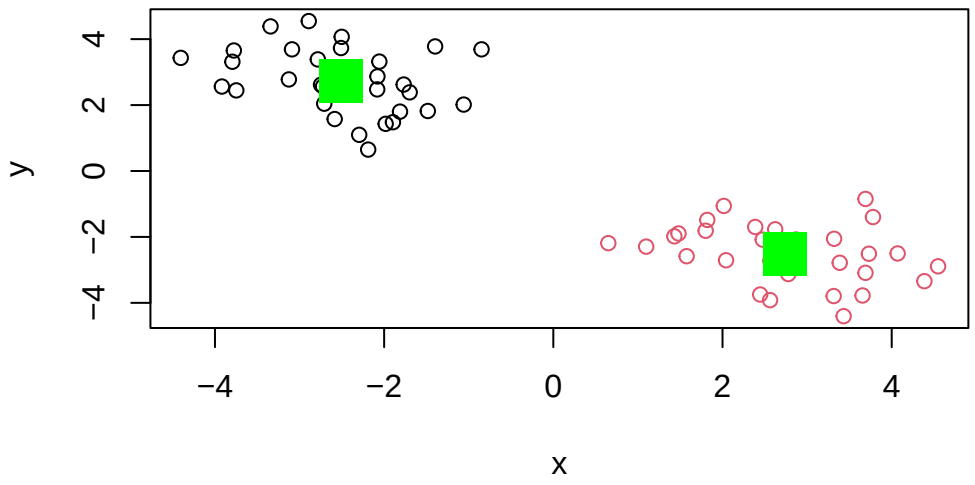
```
          x         y
1 -2.509966  2.741181
2  2.741181 -2.509966
```

Q. Make a plot of your data showing your clustering results (groupings/clusters and cluster centers).

```
plot(x, col = c(1, 2))
```




```
plot(x, col=km$cluster)
points(km$centers, col="green", pch=15, cex=3)
```

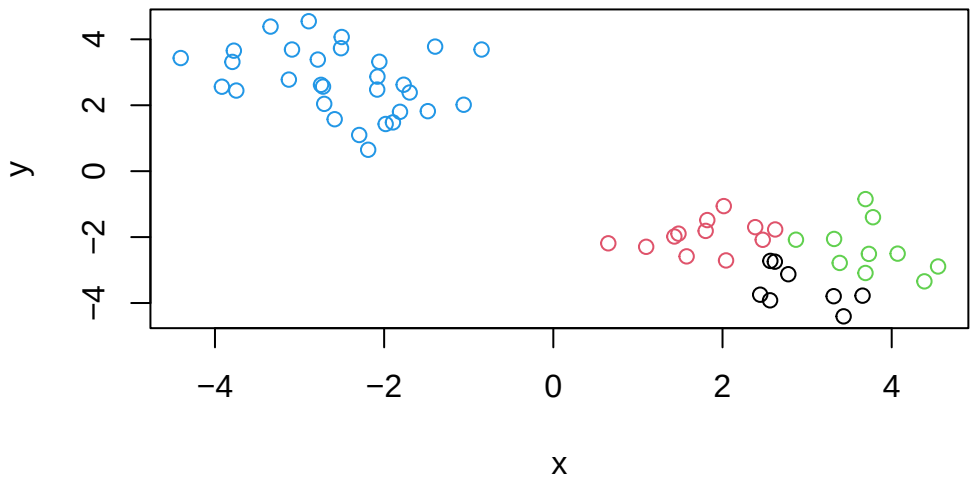


Q. Run `kmeans()` again and cluster in 4 groups and plot the results.

```
km4 <- kmeans(x, centers=4)
km4
```

```
K-means clustering with 4 clusters of sizes 8, 12, 10, 30

Cluster means:
          x         y
1  2.921380 -3.529634
2  1.783008 -1.963346
3  3.746829 -2.350176
4 -2.509966  2.741181

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

Within cluster sum of squares by cluster:
[1]  4.070966  5.982390  7.633254 51.235448
 (between_SS / total_SS =  92.6 %)

Available components:

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

```
plot(x, col=km4$cluster)
```



Hierarchical Clustering

This form of clustering aims to reveal the structure in your data by progressivey grouping points into a ever smaller number of clusters.

The main function in base R for this called `hclust()`. This function does not take our input data directly but wants a "distance matrix" that details how (dis)similar all our input points are to each other.

```
dist(x)
```

```
           1          2          3          4          5          6
2  1.1463103
3  1.6714465  0.7302767
4  1.1631793  1.0126227  1.7373420
5  0.4797790  1.1675160  1.4857383  1.5289284
6  3.2538568  2.1129177  1.8897818  2.6134733  3.2473244
7  0.8987317  0.7403961  0.8730379  1.4971568  0.6167044  2.6900526
8  2.5898605  1.9593062  1.2639784  2.9694897  2.2157356  2.5279723
9  3.2919683  2.2026439  1.6415654  3.0272513  3.1260764  1.0917449
10 2.2896105  1.1495241  0.8441929  1.9064676  2.2253176  1.0492681
11 3.9108925  2.7999588  2.6727227  3.1106490  3.9613474  0.8080543
```

12	3.24713263	2.10283021	1.84662670	2.63980521	3.22612568	0.09889291
13	2.91116817	1.82089795	1.82165877	2.11768815	2.98830614	0.61988309
14	3.00472409	1.85899310	1.53788791	2.49347688	2.95175176	0.40702431
15	1.97522089	0.87408948	0.45152026	1.77159788	1.86594403	1.44066096
16	1.06039883	0.44042207	0.61199997	1.34993485	0.90241287	2.35940704
17	2.36422673	1.26997599	0.75385126	2.13999110	2.22909927	1.21044113
18	0.84996954	1.02878998	1.75260463	0.34163368	1.25738642	2.85823275
19	0.74104091	0.43125005	1.09729290	0.79866304	0.88066203	2.51604945
20	2.03851195	1.44127283	0.80372151	2.44135423	1.68071620	2.41951185
21	2.58942433	1.47554616	0.99914186	2.29201601	2.46916647	0.99701467
22	1.91273658	1.40163289	2.00945709	0.76221544	2.23578273	2.24234865
23	3.45188458	2.33986797	2.23437579	2.67604707	3.50192759	0.45711886
24	2.42901432	1.34604269	1.43400016	1.67265211	2.51315364	0.94906642
25	1.98267291	1.34652345	1.90349907	0.86993826	2.27049028	2.03654406
26	1.42790051	1.62151115	2.35079143	0.62194868	1.88075414	3.14208579
27	1.86587871	0.76806128	0.98297115	1.25773170	1.93514874	1.41228479
28	1.92373787	0.82254108	1.00579798	1.30707119	1.98998201	1.35491989
29	1.62323764	0.69781600	1.19892863	0.85650117	1.78609200	1.76732928
30	3.16915292	2.03517619	1.60492924	2.75137932	3.07001123	0.63224677
31	7.81780003	6.70145932	6.17239240	7.38509427	7.65405612	4.79080293
32	9.69063227	8.56395781	8.05721546	9.20169832	9.54157192	6.58980053
33	9.24403079	8.11975690	7.60724828	8.76865356	9.09100555	6.15900256
34	9.29610017	8.17247657	7.65824732	8.82338739	9.14173820	6.21410753
35	11.09885256	9.96632732	9.47543176	10.57335224	10.96088167	7.96020438
36	9.98379430	8.84516044	8.37942118	9.42373364	9.86461599	6.81351756
37	8.90699099	7.77348801	7.28906390	8.38630448	8.77480202	5.77285523
38	8.02247320	6.88013167	6.43956367	7.44249568	7.92097455	4.83383764
39	10.19015449	9.05145021	8.58570027	9.62860194	10.07092262	7.01877174
40	8.35778988	7.25131484	6.70207520	7.95773843	8.17782474	5.37373121
41	9.17453407	8.12074100	7.50319964	8.91883935	8.93540190	6.41759612
42	10.26431537	9.15657665	8.60802149	9.84994316	10.08232196	7.25315838
43	10.75260499	9.63163501	9.10947845	10.28486247	10.59100821	7.67477266
44	8.57190145	7.47000328	6.91265463	8.18603845	8.38520044	5.60724967
45	9.87909070	8.78623157	8.21410068	9.51528777	9.67829799	6.94004649
46	8.96606863	7.86240591	7.30769454	8.57204247	8.78082851	5.98743363
47	8.05519776	6.93120188	6.41985979	7.58853645	7.90420682	4.98295469
48	8.53194588	7.39291714	6.93201777	7.97711507	8.41651635	5.36512408
49	7.92010475	6.78787811	6.30114904	7.41340897	7.78688704	4.80123287
50	7.69273254	6.54730494	6.13713296	7.07334706	7.61118146	4.47544660
51	8.70040640	7.58672243	7.05084467	8.27204999	8.53026001	5.67542370
52	7.64013497	6.54169245	5.97957949	7.27175522	7.45149746	4.70966344
53	8.76792260	7.73613689	7.09934557	8.56430701	8.51162885	6.11342606
54	10.11963208	9.03574271	8.45133928	9.78136074	9.90810003	7.21787886

55	7.95925727	6.82489395	6.34568588	7.43988254	7.83129437	4.82672519
56	10.73511505	9.65225473	9.06638731	10.39742216	10.52143099	7.83129437
57	10.54711561	9.42011047	8.91328497	10.05269773	10.39742216	7.43988254
58	9.26830804	8.17740243	7.60276435	8.91328497	9.06638731	6.34568588
59	9.83325252	8.72532717	8.17740243	9.42011047	9.65225473	6.82489395
60	10.93126091	9.83325252	9.26830804	10.54711561	10.73511505	7.95925727
	7	8	9	10	11	12
2						
3						
4						
5						
6						
7						
8	1.69637640					
9	2.51065591	1.68867616				
10	1.64918650	1.73149128	1.12115648			
11	3.43480045	3.32302920	1.77170296	1.82876312		
12	2.66056887	2.44647620	0.99298694	1.01303814	0.87999757	
13	2.49587954	2.73617964	1.58704214	1.03057078	1.00710680	0.69547740
14	2.37182629	2.12124341	0.78249509	0.72645205	1.21111030	0.33197523
15	1.27402966	1.48034416	1.32960577	0.39287628	2.22120270	1.40074750
16	0.33985545	1.66189080	2.24626139	1.32603500	3.09743787	2.33313594
17	1.62273567	1.40995310	0.93374936	0.34589078	2.01687413	1.14988439
18	1.34075089	2.92848471	3.17379800	2.06508396	3.39935080	2.87604299
19	0.69981174	2.24090764	2.63163007	1.57811670	3.17098533	2.51215393
20	1.14236491	0.55583347	1.79562727	1.47222704	3.22750095	2.35255743
21	1.86561715	1.55624026	0.73864136	0.39404311	1.80486813	0.92837060
22	2.06887882	3.27042205	2.91026420	1.83812847	2.57833001	2.29369735
23	2.97990222	2.96903561	1.53860518	1.39417421	0.46018286	0.55230978
24	2.03893519	2.48538809	1.64882598	0.76174577	1.48584744	0.98795368
25	2.04754746	3.15535370	2.72236263	1.66608836	2.37809231	2.08865538
26	2.00730847	3.56573098	3.63265415	2.51190856	3.55660376	3.17879615
27	1.47208927	2.17949393	1.79119560	0.67942136	2.04622161	1.42134553
28	1.52091003	2.18591741	1.75166958	0.64695310	1.98872041	1.36477432
29	1.43554044	2.45397151	2.20288700	1.08723923	2.32535651	1.78859894
30	2.46981027	1.97537094	0.46670861	0.88737986	1.36627817	0.53463779
31	7.03749643	5.82188529	4.53160627	5.55893899	4.54131890	4.75298215
32	8.92561569	7.72140702	6.41573659	7.41609353	6.23915343	6.56198146
33	8.47483137	7.26531148	5.96568299	6.97284045	5.82904870	6.12900236
34	8.52548482	7.31111206	6.01669864	7.02581318	5.88598974	6.18382028
35	10.34573026	9.15719974	7.83507615	8.81698205	7.55907407	7.93794027
36	9.25204757	8.13512803	6.74530101	7.69602467	6.38884090	6.79528198
37	8.16059480	7.02137695	5.65076388	6.62407681	5.40149771	5.74856240

38	7.31239690	6.29380569	4.81974862	5.73288030	4.41268548	4.81750894
39	9.45830489	8.33812752	6.95139827	7.90233665	6.59080258	7.00082049
40	7.56122493	6.28520173	5.06615695	6.11458186	5.14278017	5.33285726
41	8.32703895	6.87726333	5.92074288	7.02470959	6.30891967	6.36230906
42	9.46586418	8.15886544	6.97267201	8.01788180	6.96742731	7.21676876
43	9.97440373	8.72043500	7.46854293	8.48582425	7.33148235	7.64516150
44	7.76887481	6.46943021	5.27996131	6.33620524	5.38510051	5.56506343
45	9.06319081	7.70796639	6.58958678	7.65766711	6.70991889	6.89741381
46	8.16445467	6.86350404	5.67410384	6.72707066	5.74664523	5.94692883
47	7.28828170	6.10744261	4.77836631	5.78467383	4.68831766	4.95017003
48	7.80494889	6.72730180	5.30142924	6.24385714	4.96338516	5.34535952
49	7.17276572	6.05220302	4.66316562	5.63869813	4.46528804	4.77384901
50	7.00783149	6.06727252	4.53825907	5.40541921	4.01445490	4.46528804
51	7.91355948	6.66115082	5.41148693	6.44525420	5.40541921	5.63869813
52	6.83520280	5.54663744	4.34850171	5.41148693	4.53825907	4.66316562
53	7.90818633	6.41912126	5.54663744	6.66115082	6.06727252	6.05220302
54	9.29437469	7.90818633	6.83520280	7.91355948	7.00783149	7.17276572
55	7.21787886	6.11342606	4.70966344	5.67542370	4.47544660	4.80123287
56	9.90810003	8.51162885	7.45149746	8.53026001	7.61118146	7.78688704
57	9.78136074	8.56430701	7.27175522	8.27204999	7.07334706	7.41340897
58	8.45133928	7.09934557	5.97957949	7.05084467	6.13713296	6.30114904
59	9.03574271	7.73613689	6.54169245	7.58672243	6.54730494	6.78787811
60	10.11963208	8.76792260	7.64013497	8.70040640	7.69273254	7.92010475

	13	14	15	16	17	18
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14	0.80455484					
15	1.38775062	1.10032855				
16	2.15604159	2.05185723	0.96369412			
17	1.32705547	0.82272372	0.39606355	1.33297006		
18	2.39704685	2.70154710	1.87091796	1.26908919	2.25795321	
19	2.17378828	2.28042907	1.30208489	0.58090109	1.69812065	0.68824489
20	2.50259566	2.02090794	1.13766770	1.11366122	1.21451551	2.38267522

21	1.20152150	0.59680638	0.61434276	1.57002486	0.24621501	2.43528059
22	1.65855939	2.23916837	1.86392037	1.82977196	2.15000303	1.10234643
23	0.55922966	0.85290003	1.78375161	2.64171545	1.61301459	2.95613970
24	0.48215927	0.92783898	1.03849144	1.70030851	1.10434473	1.93526111
25	1.45232191	2.04004669	1.72108511	1.78430344	1.98679306	1.21025274
26	2.59938585	3.06273070	2.39258745	1.91833968	2.75726994	0.66672875
27	1.05525827	1.24128142	0.71190750	1.13673333	0.97724438	1.46187332
28	0.99927810	1.18885140	0.71089319	1.18429941	0.95542948	1.51626463
29	1.31835627	1.63737429	1.05080512	1.13484077	1.36629229	1.09239999
30	1.12779597	0.32915785	1.20569282	2.16861071	0.85158039	2.93710267
31	5.36318686	4.89730242	5.84356182	6.77773817	5.45366056	7.59844772
32	7.13216785	6.73406296	7.71554761	8.65894288	7.32956036	9.43164168
33	6.70841586	6.29513917	7.26881051	8.21001424	6.88176526	8.99476225
34	6.76428201	6.34901240	7.32088471	8.26137674	6.93353363	9.04891330
35	8.48326913	8.12471228	9.12532618	10.07406036	8.74240727	10.81342525
36	7.32434143	6.99419320	8.01561107	8.97125339	7.63829701	9.67093241
37	6.30555269	5.93198582	6.93432713	7.88531152	6.55288208	8.62292789
38	5.34247809	5.02425430	6.06236851	7.02279658	5.69187528	7.69194366
39	7.52828469	7.20029529	8.22200308	9.17762116	7.84466255	9.87635069
40	5.95265251	5.46580909	6.38741609	7.31074583	5.99474646	8.16388943
41	7.02377993	6.44115932	7.24665840	8.11427312	6.85093406	9.08932164
42	7.81919605	7.36161391	8.29360897	9.21700480	7.90115311	10.06226617
43	8.22137035	7.81073441	8.77764206	9.71450882	8.38897497	10.51077884
44	6.18895026	5.69312792	6.60400857	7.52236546	6.21043876	8.38897497
45	7.52173416	7.02188220	7.91718689	8.82533023	7.52236546	9.71450882
46	6.56487018	6.07997966	6.99726132	7.91718689	6.60400857	8.77764206
47	5.54201282	5.10991775	6.07997966	7.02188220	5.69312792	7.81073441
48	5.88396900	5.54201282	6.56487018	7.52173416	6.18895026	8.22137035
49	5.34535952	4.95017003	5.94692883	6.89741381	5.56506343	7.64516150
50	4.96338516	4.68831766	5.74664523	6.70991889	5.38510051	7.33148235
51	6.24385714	5.78467383	6.72707066	7.65766711	6.33620524	8.48582425
52	5.30142924	4.77836631	5.67410384	6.58958678	5.27996131	7.46854293
53	6.72730180	6.10744261	6.86350404	7.70796639	6.46943021	8.72043500
54	7.80494889	7.28828170	8.16445467	9.06319081	7.76887481	9.97440373
55	5.36512408	4.98295469	5.98743363	6.94004649	5.60724967	7.67477266
56	8.41651635	7.90420682	8.78082851	9.67829799	8.38520044	10.59100821
57	7.97711507	7.58853645	8.57204247	9.51528777	8.18603845	10.28486247
58	6.93201777	6.41985979	7.30769454	8.21410068	6.91265463	9.10947845
59	7.39291714	6.93120188	7.86240591	8.78623157	7.47000328	9.63163501
60	8.53194588	8.05519776	8.96606863	9.87909070	8.57190145	10.75260499

	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 1.69443109
21 1.90653041 1.42418787
22 1.39969230 2.79901731 2.22692408
23 2.71163138 2.82737617 1.42049022 2.18634741
24 1.69182132 2.17614441 1.06745413 1.31693120 1.03083076
25 1.40809427 2.70232253 2.05058166 0.20639208 1.98197736 1.11514515
26 1.34058189 3.02749099 2.90141542 0.99640806 3.14640641 2.19307502
27 1.12518549 1.77914439 1.07343702 1.17327913 1.58659803 0.57802544
28 1.18296381 1.79682164 1.04005866 1.19543275 1.52900016 0.52385776
29 0.91319570 1.99793991 1.48085170 0.82053602 1.87468166 0.84635674
30 2.46485363 1.96055230 0.60579284 2.54497980 1.08621348 1.24452338
31 7.13253101 6.16807140 5.23049050 7.02118138 4.85758740 5.73905683
32 8.99388767 8.06918963 7.10124688 8.78610102 6.60104805 7.53715045
33 8.55006609 7.61395214 6.65465304 8.36469682 6.18207431 7.10738388
34 8.60286674 7.66174461 6.70678692 8.42073983 6.23838321 7.16258428
35 10.39491127 9.50206635 8.51113287 10.12716661 7.94166270 8.90280356
36 9.27151318 8.45085922 7.40289184 8.96180220 6.77866091 7.75138452
37 8.20193071 7.34234034 6.32035796 7.95616659 5.76996936 6.71783232
38 7.30425619 6.56603885 5.45278724 6.98203794 4.79766877 5.76996936
39 9.47775166 8.65558586 7.60927946 9.16460490 6.98203794 7.95616659
40 7.68255547 6.65689261 5.77678401 7.60927946 5.45278724 6.32035796
41 8.54714767 7.31997389 6.65689261 8.65558586 6.56603885 7.34234034
42 9.58782608 8.54714767 7.68255547 9.47775166 7.30425619 8.20193071
43 10.06226617 9.08932164 8.16388943 9.87635069 7.69194366 8.62292789
44 7.90115311 6.85093406 5.99474646 7.84466255 5.69187528 6.55288208
45 9.21700480 8.11427312 7.31074583 9.17762116 7.02279658 7.88531152
46 8.29360897 7.24665840 6.38741609 8.22200308 6.06236851 6.93432713

47	7.36161391	6.44115932	5.46580909	7.20029529	5.02425430	5.93198582
48	7.81919605	7.02377993	5.95265251	7.52828469	5.34247809	6.30555269
49	7.21676876	6.36230906	5.33285726	7.00082049	4.81750894	5.74856240
50	6.96742731	6.30891967	5.14278017	6.59080258	4.41268548	5.40149771
51	8.01788180	7.02470959	6.11458186	7.90233665	5.73288030	6.62407681
52	6.97267201	5.92074288	5.06615695	6.95139827	4.81974862	5.65076388
53	8.15886544	6.87726333	6.28520173	8.33812752	6.29380569	7.02137695
54	9.46586418	8.32703895	7.56122493	9.45830489	7.31239690	8.16059480
55	7.25315838	6.41759612	5.37373121	7.01877174	4.83383764	5.77285523
56	10.08232196	8.93540190	8.17782474	10.07092262	7.92097455	8.77480202
57	9.84994316	8.91883935	7.95773843	9.62860194	7.44249568	8.38630448
58	8.60802149	7.50319964	6.70207520	8.58570027	6.43956367	7.28906390
59	9.15657665	8.12074100	7.25131484	9.05145021	6.88013167	7.77348801
60	10.26431537	9.17453407	8.35778988	10.19015449	8.02247320	8.90699099

	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	1.18491301					
27	1.01650788	1.84717015				
28	1.03216494	1.89202852	0.05809210			
29	0.70468087	1.43594905	0.41215649	0.45610125		

30	2.34891879	3.33614613	1.49381350	1.44591015	1.90143573	
31	6.81480842	7.93164213	6.13845792	6.08551523	6.52903663	4.67233080
32	8.58084024	9.72615852	7.96993829	7.91537714	8.34976441	6.52903663
33	8.15903601	9.29850145	7.53288897	7.47867411	7.91537714	6.08551523
34	8.21504722	9.35389872	7.58704560	7.53288897	7.96993829	6.13845792
35	9.92339649	11.08249352	9.35389872	9.29850145	9.72615852	7.93164213
36	8.75883923	9.92339649	8.21504722	8.15903601	8.58084024	6.81480842
37	7.75138452	8.90280356	7.16258428	7.10738388	7.53715045	5.73905683
38	6.77866091	7.94166270	6.23838321	6.18207431	6.60104805	4.85758740
39	8.96180220	10.12716661	8.42073983	8.36469682	8.78610102	7.02118138
40	7.40289184	8.51113287	6.70678692	6.65465304	7.10124688	5.23049050
41	8.45085922	9.50206635	7.66174461	7.61395214	8.06918963	6.16807140
42	9.27151318	10.39491127	8.60286674	8.55006609	8.99388767	7.13253101
43	9.67093241	10.81342525	9.04891330	8.99476225	9.43164168	7.59844772
44	7.63829701	8.74240727	6.93353363	6.88176526	7.32956036	5.45366056
45	8.97125339	10.07406036	8.26137674	8.21001424	8.65894288	6.77773817
46	8.01561107	9.12532618	7.32088471	7.26881051	7.71554761	5.84356182
47	6.99419320	8.12471228	6.34901240	6.29513917	6.73406296	4.89730242
48	7.32434143	8.48326913	6.76428201	6.70841586	7.13216785	5.36318686
49	6.79528198	7.93794027	6.18382028	6.12900236	6.56198146	4.75298215
50	6.38884090	7.55907407	5.88598974	5.82904870	6.23915343	4.54131890
51	7.69602467	8.81698205	7.02581318	6.97284045	7.41609353	5.55893899
52	6.74530101	7.83507615	6.01669864	5.96568299	6.41573659	4.53160627
53	8.13512803	9.15719974	7.31111206	7.26531148	7.72140702	5.82188529
54	9.25204757	10.34573026	8.52548482	8.47483137	8.92561569	7.03749643
55	6.81351756	7.96020438	6.21410753	6.15900256	6.58980053	4.79080293
56	9.86461599	10.96088167	9.14173820	9.09100555	9.54157192	7.65405612
57	9.42373364	10.57335224	8.82338739	8.76865356	9.20169832	7.38509427
58	8.37942118	9.47543176	7.65824732	7.60724828	8.05721546	6.17239240
59	8.84516044	9.96632732	8.17247657	8.11975690	8.56395781	6.70145932
60	9.98379430	11.09885256	9.29610017	9.24403079	9.69063227	7.81780003

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

1.90143573						
1.44591015	0.45610125					
1.49381350	0.41215649	0.05809210				
3.33614613	1.43594905	1.89202852	1.84717015			
2.34891879	0.70468087	1.03216494	1.01650788	1.18491301		
1.24452338	0.84635674	0.52385776	0.57802544	2.19307502	1.11514515	
1.08621348	1.87468166	1.52900016	1.58659803	3.14640641	1.98197736	
2.54497980	0.82053602	1.19543275	1.17327913	0.99640806	0.20639208	
0.60579284	1.48085170	1.04005866	1.07343702	2.90141542	2.05058166	
1.96055230	1.99793991	1.79682164	1.77914439	3.02749099	2.70232253	
2.46485363	0.91319570	1.18296381	1.12518549	1.34058189	1.40809427	
2.93710267	1.09239999	1.51626463	1.46187332	0.66672875	1.21025274	
0.85158039	1.36629229	0.95542948	0.97724438	2.75726994	1.98679306	
2.16861071	1.13484077	1.18429941	1.13673333	1.91833968	1.78430344	
1.20569282	1.05080512	0.71089319	0.71190750	2.39258745	1.72108511	
0.32915785	1.63737429	1.18885140	1.24128142	3.06273070	2.04004669	
1.12779597	1.31835627	0.99927810	1.05525827	2.59938585	1.45232191	
0.53463779	1.78859894	1.36477432	1.42134553	3.17879615	2.08865538	
1.36627817	2.32535651	1.98872041	2.04622161	3.55660376	2.37809231	
0.88737986	1.08723923	0.64695310	0.67942136	2.51190856	1.66608836	
0.46670861	2.20288700	1.75166958	1.79119560	3.63265415	2.72236263	
1.97537094	2.45397151	2.18591741	2.17949393	3.56573098	3.15535370	
2.46981027	1.43554044	1.52091003	1.47208927	2.00730847	2.04754746	
0.63224677	1.76732928	1.35491989	1.41228479	3.14208579	2.03654406	

```
56  3.07001123  1.78609200  1.98998201  1.93514874  1.88075414  2.27049028
57  2.75137932  0.85650117  1.30707119  1.25773170  0.62194868  0.86993826
58  1.60492924  1.19892863  1.00579798  0.98297115  2.35079143  1.90349907
59  2.03517619  0.69781600  0.82254108  0.76806128  1.62151115  1.34652345
60  3.16915292  1.62323764  1.92373787  1.86587871  1.42790051  1.98267291
            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  1.03083076
```

39	1.31693120	2.18634741				
40	1.06745413	1.42049022	2.22692408			
41	2.17614441	2.82737617	2.79901731	1.42418787		
42	1.69182132	2.71163138	1.39969230	1.90653041	1.69443109	
43	1.93526111	2.95613970	1.10234643	2.43528059	2.38267522	0.68824489
44	1.10434473	1.61301459	2.15000303	0.24621501	1.21451551	1.69812065
45	1.70030851	2.64171545	1.82977196	1.57002486	1.11366122	0.58090109
46	1.03849144	1.78375161	1.86392037	0.61434276	1.13766770	1.30208489
47	0.92783898	0.85290003	2.23916837	0.59680638	2.02090794	2.28042907
48	0.48215927	0.55922966	1.65855939	1.20152150	2.50259566	2.17378828
49	0.98795368	0.55230978	2.29369735	0.92837060	2.35255743	2.51215393
50	1.48584744	0.46018286	2.57833001	1.80486813	3.22750095	3.17098533
51	0.76174577	1.39417421	1.83812847	0.39404311	1.47222704	1.57811670
52	1.64882598	1.53860518	2.91026420	0.73864136	1.79562727	2.63163007
53	2.48538809	2.96903561	3.27042205	1.55624026	0.55583347	2.24090764
54	2.03893519	2.97990222	2.06887882	1.86561715	1.14236491	0.69981174
55	0.94906642	0.45711886	2.24234865	0.99701467	2.41951185	2.51604945
56	2.51315364	3.50192759	2.23578273	2.46916647	1.68071620	0.88066203
57	1.67265211	2.67604707	0.76221544	2.29201601	2.44135423	0.79866304
58	1.43400016	2.23437579	2.00945709	0.99914186	0.80372151	1.09729290
59	1.34604269	2.33986797	1.40163289	1.47554616	1.44127283	0.43125005
60	2.42901432	3.45188458	1.91273658	2.58942433	2.03851195	0.74104091

	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						

2.25795321						
1.26908919	1.33297006					
1.87091796	0.39606355	0.96369412				
2.70154710	0.82272372	2.05185723	1.10032855			
2.39704685	1.32705547	2.15604159	1.38775062	0.80455484		
2.87604299	1.14988439	2.33313594	1.40074750	0.33197523	0.69547740	
3.39935080	2.01687413	3.09743787	2.22120270	1.21111030	1.00710680	
2.06508396	0.34589078	1.32603500	0.39287628	0.72645205	1.03057078	
3.17379800	0.93374936	2.24626139	1.32960577	0.78249509	1.58704214	
2.92848471	1.40995310	1.66189080	1.48034416	2.12124341	2.73617964	
1.34075089	1.62273567	0.33985545	1.27402966	2.37182629	2.49587954	
2.85823275	1.21044113	2.35940704	1.44066096	0.40702431	0.61988309	
1.25738642	2.22909927	0.90241287	1.86594403	2.95175176	2.98830614	
0.34163368	2.13999110	1.34993485	1.77159788	2.49347688	2.11768815	
1.75260463	0.75385126	0.61199997	0.45152026	1.53788791	1.82165877	
1.02878998	1.26997599	0.44042207	0.87408948	1.85899310	1.82089795	
0.84996954	2.36422673	1.06039883	1.97522089	3.00472409	2.91116817	
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  0.87999757
51  1.01303814  1.82876312
52  0.99298694  1.77170296  1.12115648
53  2.44647620  3.32302920  1.73149128  1.68867616
54  2.66056887  3.43480045  1.64918650  2.51065591  1.69637640
55  0.09889291  0.80805429  1.04926806  1.09174492  2.52797230  2.69005257
56  3.22612568  3.96134737  2.22531762  3.12607641  2.21573563  0.61670436
57  2.63980521  3.11064897  1.90646763  3.02725134  2.96948968  1.49715680
58  1.84662670  2.67272266  0.84419292  1.64156543  1.26397835  0.87303787
59  2.10283021  2.79995884  1.14952412  2.20264388  1.95930624  0.74039606
60  3.24713263  3.91089248  2.28961052  3.29196834  2.58986051  0.89873169
              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
```

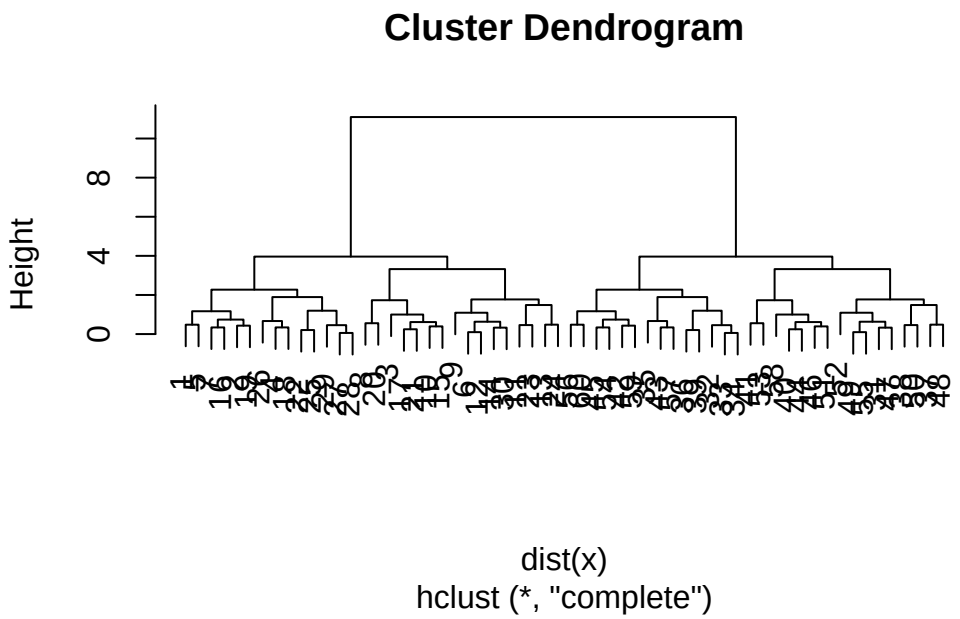
```
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56  3.24732437
57  2.61347333  1.52892845
58  1.88978178  1.48573832  1.73734196
59  2.11291767  1.16751603  1.01262272  0.73027674
60  3.25385678  0.47977901  1.16317929  1.67144646  1.14631034
```

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

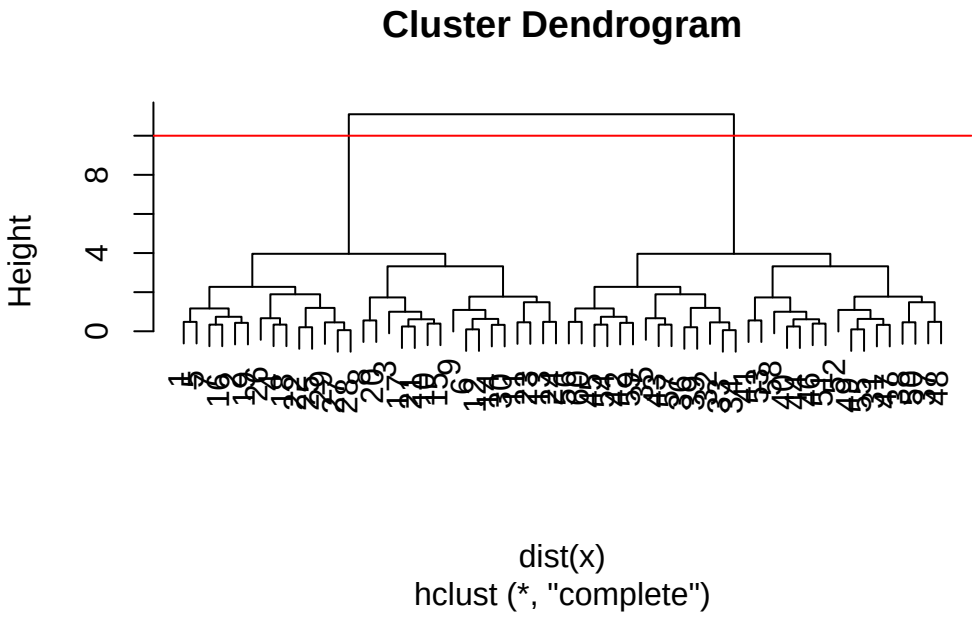
```
Call:
hclust(d = dist(x))

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

The print out above is not very useful (unlike that from kmeans) but there is a useful `plot()` method.

```
plot(hc)
```




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

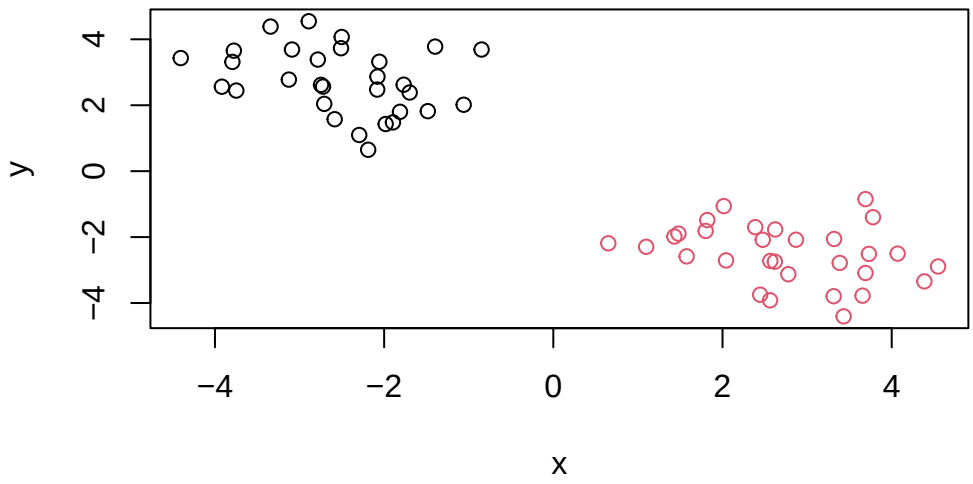


To gte my main result (my cluster membership vector) I need to "cut" my tree using the function `cutree()`

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

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

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



Principal Component Analysis (PCA)

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

Q1.

```
dim(x)
```

```
[1] 17  5
```

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

```
              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
```

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

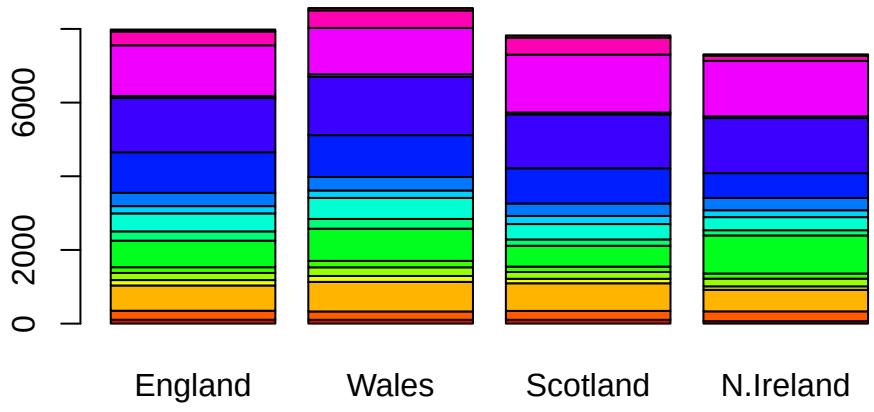
Q2. I prefer the second option because if I run the above code again, it will delete the name to the next column.

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

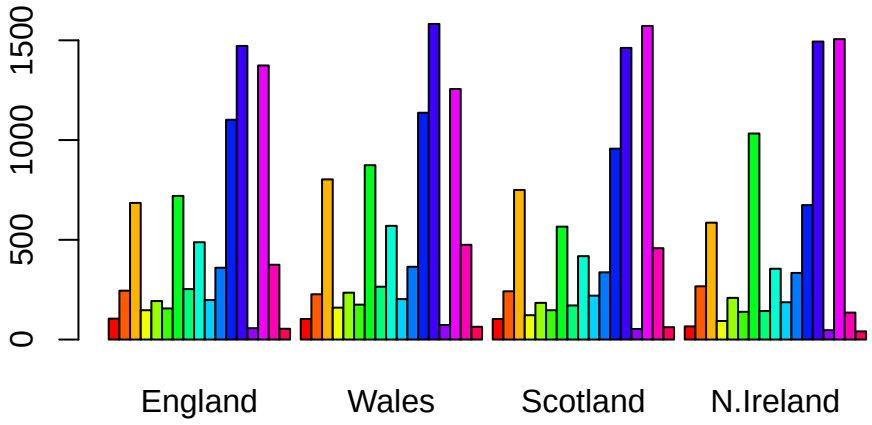
```
              England Wales Scotland N.Ireland
Cheese            105   103      103        66
Carcass_meat      245   227      242       267
Other_meat        685   803      750       586
Fish              147   160      122        93
Fats_and_oils     193   235      184       209
Sugars        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

Q3. Change beside=T to beside=F.

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

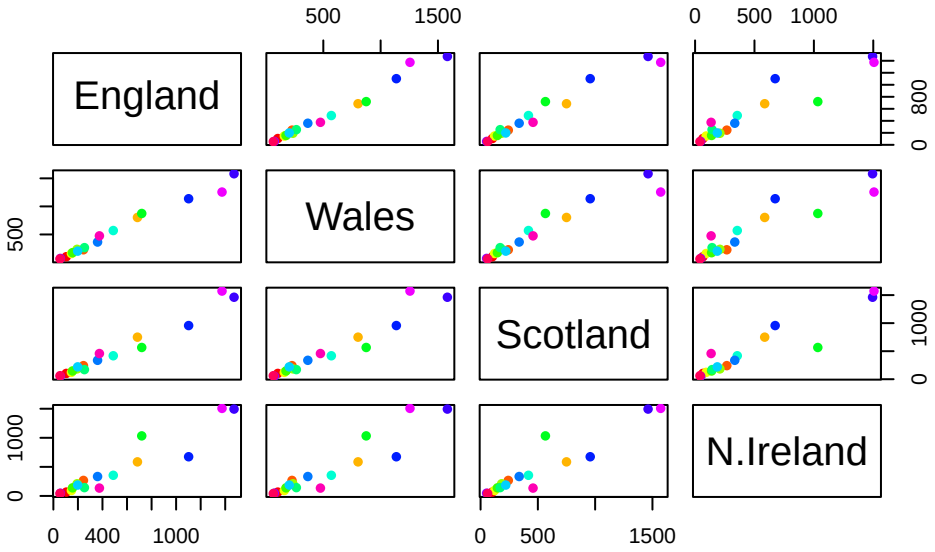


```
barplot(as.matrix(x), beside=T, col=rainbow(nrow(x)))
```



The so-called "pairs" plot can be useful for small datasets:

```
#rainbow(nrow(x))
pairs(x, col=rainbow(nrow(x)),pch=16)
```



Q5. The graph shows duplicate information, for example the plot next to the England box has England on the x axis and Wales on the y. While the plot directly below the England box has Wales on the x axis and England on the y. As a result, the two plots are the same, just flipped cause of the flipped axes.

Q6. Ireland seems to be to have more similar graphs across. But we can use PCA to clarify all the differences.

The pairs plot is useful for small datasets but it can be lots of work to interpret and gets untractable for larger datasets.

So PCA to the rescue…

The main function to do PCA is base R is called `prcomp()`.This function wants the trasnpose of our data in this case.

```
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
```

```
attributes(pca)
```

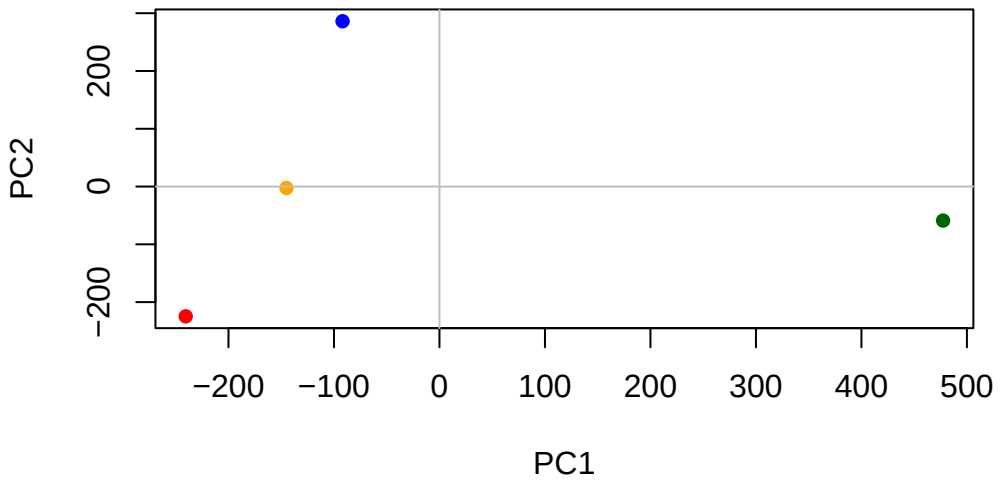
```
$names
[1] "sdev"     "rotation" "center"   "scale"    "x"

$class
[1] "prcomp"
```

```
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
```

a major PCA result viz is called a "PCA plot" (a.k.a. a score plot, biplot, PC1 vs PC2 plot, ordination plot)

```
mycols <- c("orange", "red", "blue", "darkgreen")
plot(pca$x[,1], pca$x[,2], col=mycols, pch=16,
     xlab="PC1", ylab="PC2")
abline(h=0, col="gray")
abline(v=0, col="gray")
```



Another important output from PCA is called the "loadings" vector or the "rotation" component - this tells us how much the original variables (the foods in this case) contribute to the new PCs.

```
pca$rotation
```

	PC1	PC2	PC3	PC4
Cheese	-0.056955380	0.016012850	0.02394295	-0.409382587
Carcass_meat	0.047927628	0.013915823	0.06367111	0.729481922
Other_meat	-0.258916658	-0.015331138	-0.55384854	0.331001134
Fish	-0.084414983	-0.050754947	0.03906481	0.022375878
Fats_and_oils	-0.005193623	-0.095388656	-0.12522257	0.034512161
Sugars	-0.037620983	-0.043021699	-0.03605745	0.024943337
Fresh_potatoes	0.401402060	-0.715017078	-0.20668248	0.021396007
Fresh_Veg	-0.151849942	-0.144900268	0.21382237	0.001606882
Other_Veg	-0.243593729	-0.225450923	-0.05332841	0.031153231
Processed_potatoes	-0.026886233	0.042850761	-0.07364902	-0.017379680
Processed_Veg	-0.036488269	-0.045451802	0.05289191	0.021250980
Fresh_fruit	-0.632640898	-0.177740743	0.40012865	0.227657348
Cereals	-0.047702858	-0.212599678	-0.35884921	0.100043319
Beverages	-0.026187756	-0.030560542	-0.04135860	-0.018382072
Soft_drinks	0.232244140	0.555124311	-0.16942648	0.222319484

```
Alcoholic_drinks    -0.463968168  0.113536523 -0.49858320 -0.273126013
Confectionery       -0.029650201  0.005949921 -0.05232164  0.001890737
```

PCA looks to be super useful method for gaining some insight into high dimensional data that is difficult to examine in other ways.

PCA of RNASeq 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
```

```
# Again we have to transpose our data
pca <- prcomp(t(rna.data), scale=TRUE)
```

```
summary(pca)
```

```
Importance of components:
                          PC1    PC2     PC3     PC4     PC5     PC6     PC7
Standard deviation     9.6237 1.5198 1.05787 1.05203 0.88062 0.82545 0.80111
Proportion of Variance 0.9262 0.0231 0.01119 0.01107 0.00775 0.00681 0.00642
Cumulative Proportion  0.9262 0.9493 0.96045 0.97152 0.97928 0.98609 0.99251
                          PC8     PC9      PC10
Standard deviation     0.62065 0.60342 3.345e-15
Proportion of Variance 0.00385 0.00364 0.000e+00
Cumulative Proportion  0.99636 1.00000 1.000e+00
```

Q10. How many genes in the dataset?

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

```
[1] 100
```

```
attributes(pca)
```

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

$class
[1] "prcomp"
```

```
pca$x
```

```
          PC1        PC2        PC3        PC4        PC5        PC6
wt1 -9.697374  1.5233313 -0.2753567  0.7322391 -0.6749398  1.1823860
wt2 -9.138950  0.3748504  1.0867958 -1.9461655  0.7571209 -0.4369228
wt3 -9.054263 -0.9855163  0.4152966  1.4166028  0.5835918  0.6937236
wt4 -8.731483 -0.7468371  0.5875748  0.2268129 -1.5404775 -1.2723618
wt5 -9.006312 -0.2945307 -1.8498101 -0.4303812  0.8666124 -0.2496025
ko1  8.846999  2.2345475 -0.1462750 -1.1544333 -0.6947862  0.7128021
ko2  9.213885 -3.2607503  0.2287292 -0.7658122 -0.4922849  0.9170241
ko3  9.458412 -0.2636283 -1.5778183  0.2433549  0.3654124 -0.5837724
ko4  8.883412  0.6339701  1.5205064  0.7760158  1.2158376 -0.1446094
ko5  9.225673  0.7845635  0.0103574  0.9017667 -0.3860869 -0.8186668
           PC7         PC8         PC9         PC10
wt1 -0.24446614  1.03519396  0.07010231 3.388516e-15
wt2 -0.03275370  0.26622249  0.72780448 2.996563e-15
wt3 -0.03578383 -1.05851494  0.52979799 3.329630e-15
wt4 -0.52795595 -0.20995085 -0.50325679 3.317526e-15
wt5  0.83227047 -0.05891489 -0.81258430 2.712504e-15
ko1 -0.07864392 -0.94652648 -0.24613776 2.768138e-15
ko2  0.30945771  0.33231138 -0.08786782 3.317091e-15
ko3 -1.43723425  0.14495188  0.56617746 3.299214e-15
ko4 -0.35073859  0.30381920 -0.87353886 3.000948e-15
ko5  1.56584821  0.19140827  0.62950330 2.785473e-15
```

```
head(pca$x)
```

```
          PC1        PC2        PC3        PC4        PC5        PC6
```

```
wt1 -9.697374  1.5233313 -0.2753567  0.7322391 -0.6749398  1.1823860
wt2 -9.138950  0.3748504  1.0867958 -1.9461655  0.7571209 -0.4369228
wt3 -9.054263 -0.9855163  0.4152966  1.4166028  0.5835918  0.6937236
wt4 -8.731483 -0.7468371  0.5875748  0.2268129 -1.5404775 -1.2723618
wt5 -9.006312 -0.2945307 -1.8498101 -0.4303812  0.8666124 -0.2496025
ko1  8.846999  2.2345475 -0.1462750 -1.1544333 -0.6947862  0.7128021
           PC7         PC8         PC9         PC10
wt1 -0.24446614  1.03519396  0.07010231 3.388516e-15
wt2 -0.03275370  0.26622249  0.72780448 2.996563e-15
wt3 -0.03578383 -1.05851494  0.52979799 3.329630e-15
wt4 -0.52795595 -0.20995085 -0.50325679 3.317526e-15
wt5  0.83227047 -0.05891489 -0.81258430 2.712504e-15
ko1 -0.07864392 -0.94652648 -0.24613776 2.768138e-15
```

I will make a main result figure using ggplot:

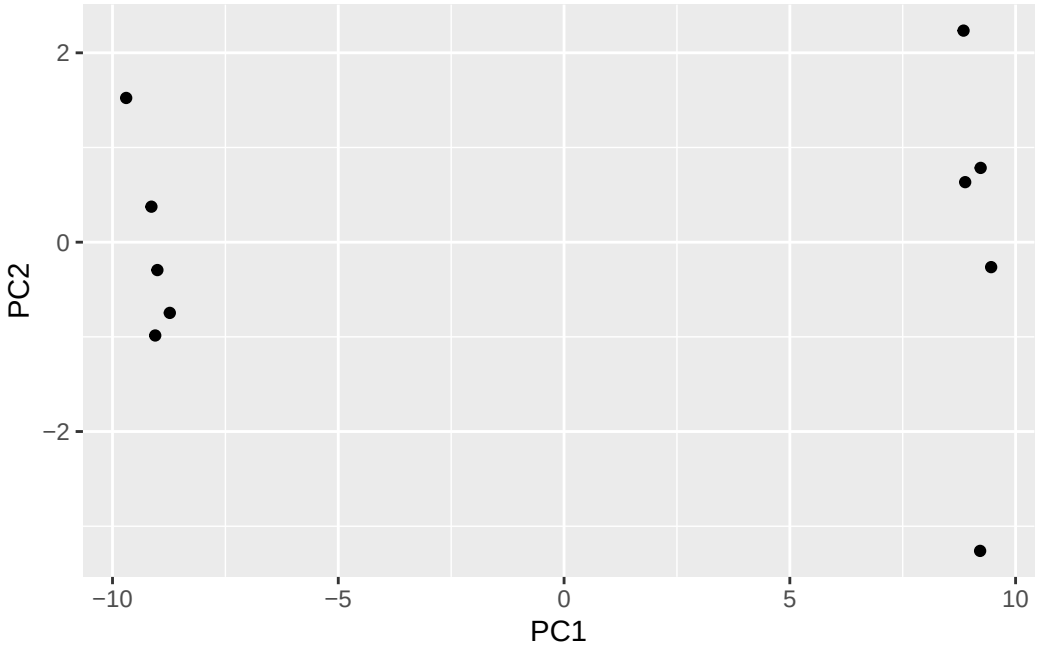
```
library(ggplot2)
```

```
res <- as.data.frame(pca$x)
```

```
head(res)
```

```
          PC1        PC2        PC3        PC4        PC5        PC6
wt1 -9.697374  1.5233313 -0.2753567  0.7322391 -0.6749398  1.1823860
wt2 -9.138950  0.3748504  1.0867958 -1.9461655  0.7571209 -0.4369228
wt3 -9.054263 -0.9855163  0.4152966  1.4166028  0.5835918  0.6937236
wt4 -8.731483 -0.7468371  0.5875748  0.2268129 -1.5404775 -1.2723618
wt5 -9.006312 -0.2945307 -1.8498101 -0.4303812  0.8666124 -0.2496025
ko1  8.846999  2.2345475 -0.1462750 -1.1544333 -0.6947862  0.7128021
           PC7         PC8         PC9         PC10
wt1 -0.24446614  1.03519396  0.07010231 3.388516e-15
wt2 -0.03275370  0.26622249  0.72780448 2.996563e-15
wt3 -0.03578383 -1.05851494  0.52979799 3.329630e-15
wt4 -0.52795595 -0.20995085 -0.50325679 3.317526e-15
wt5  0.83227047 -0.05891489 -0.81258430 2.712504e-15
ko1 -0.07864392 -0.94652648 -0.24613776 2.768138e-15
```

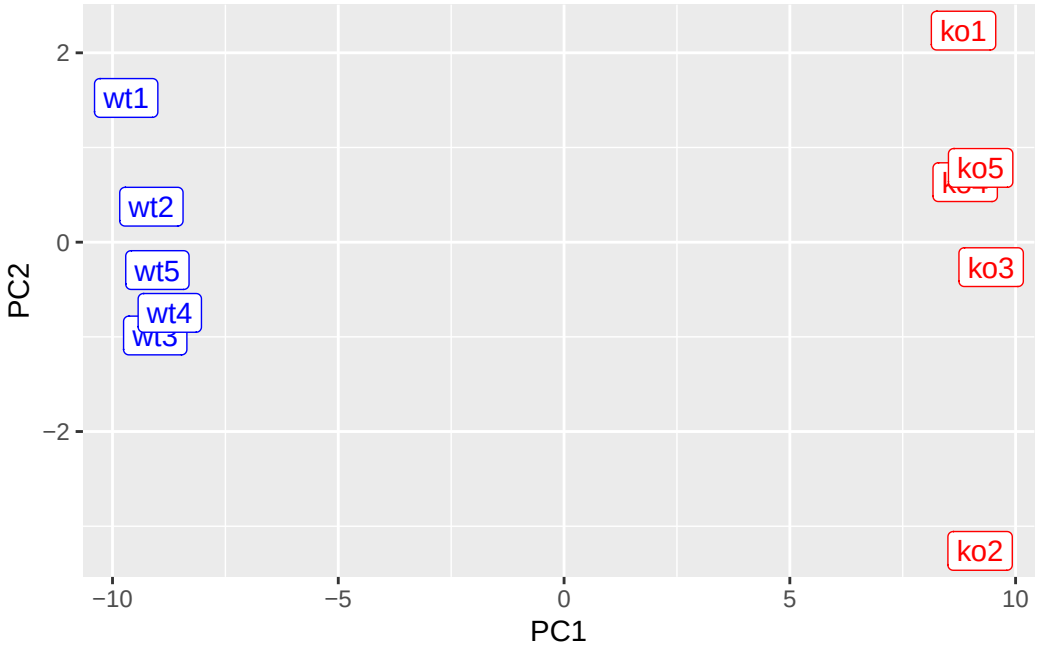
```
ggplot(res) +
  aes(PC1, PC2) +
  geom_point()
```




```
mycols <- c(rep("blue", 5), rep("red", 5))
mycols
```

```
[1] "blue" "blue" "blue" "blue" "blue" "red"  "red"  "red"  "red"  "red"
```

```
ggplot(res) +
  aes(x=PC1, y=PC2, label=row.names(res)) +
  geom_point(col=mycols) +
  geom_label(col=mycols)
```




```
colnames(rna.data)
```

```
 [1] "wt1" "wt2" "wt3" "wt4" "wt5" "ko1" "ko2" "ko3" "ko4" "ko5"
```

```
kmeans(pca$x[,1], centers = 2)
```

```
K-means clustering with 2 clusters of sizes 5, 5

Cluster means:
       [,1]
1 -9.125676
2  9.125676

Clustering vector:
wt1 wt2 wt3 wt4 wt5 ko1 ko2 ko3 ko4 ko5
  1   1   1   1   1   2   2   2   2   2

Within cluster sum of squares by cluster:
[1] 0.5017505 0.2648467
 (between_SS / total_SS =  99.9 %)
```

```
Available components:

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