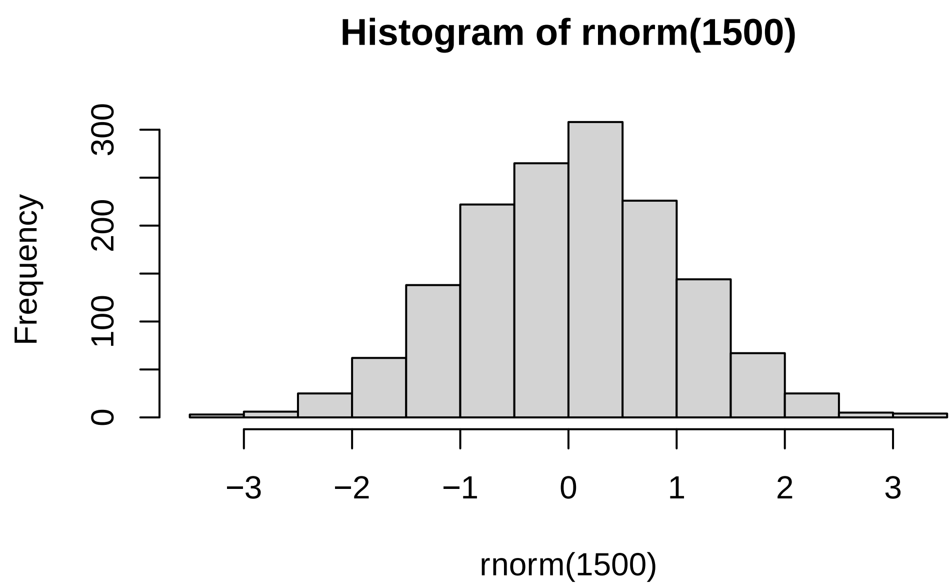
Class07

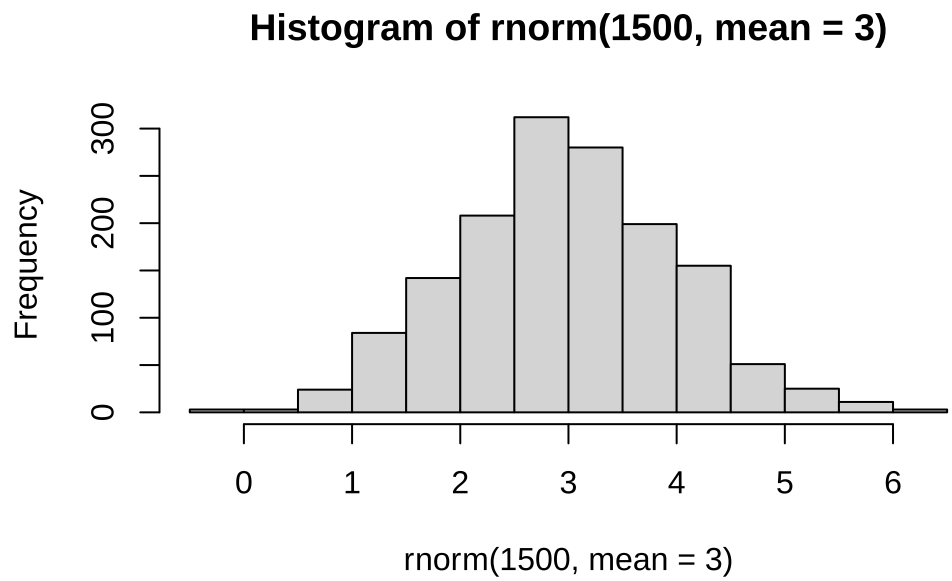
Dhruv

KMEANS() is used for PCA analysis. To learn about this, we can create a test dataset for ourselves. Before anything, we should start with the `rnorm()` function.

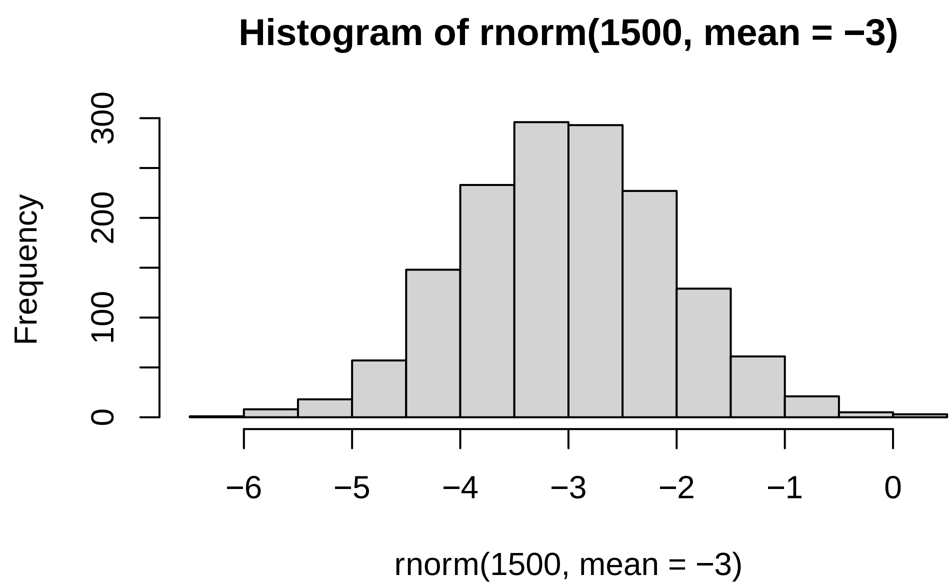
```
hist(rnorm(1500))
```



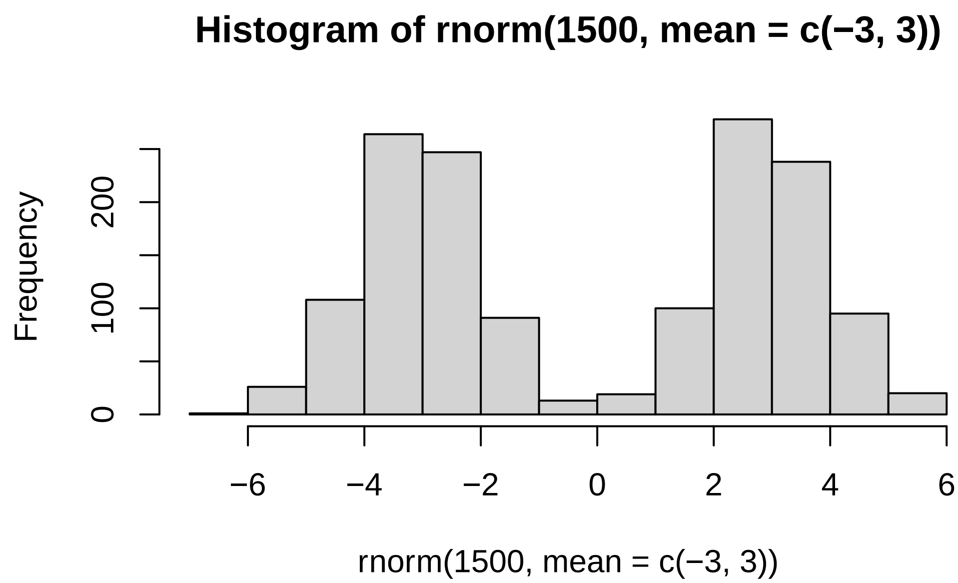

```
hist( rnorm(1500, mean = 3))
```



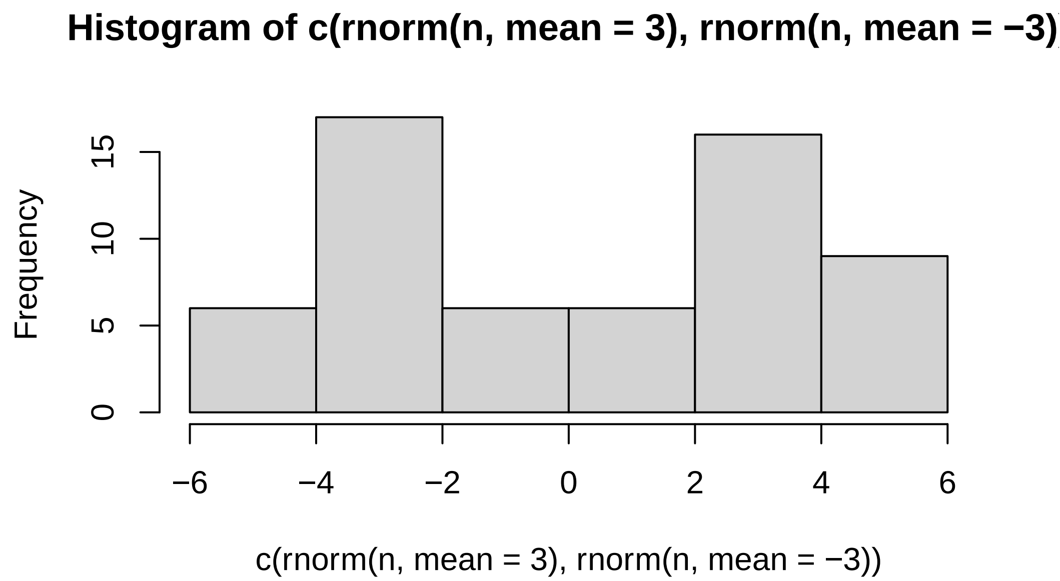

```r
hist( rnorm(1500, mean = -3))
```



```r
hist( rnorm(1500, mean = c(-3,3)))
```



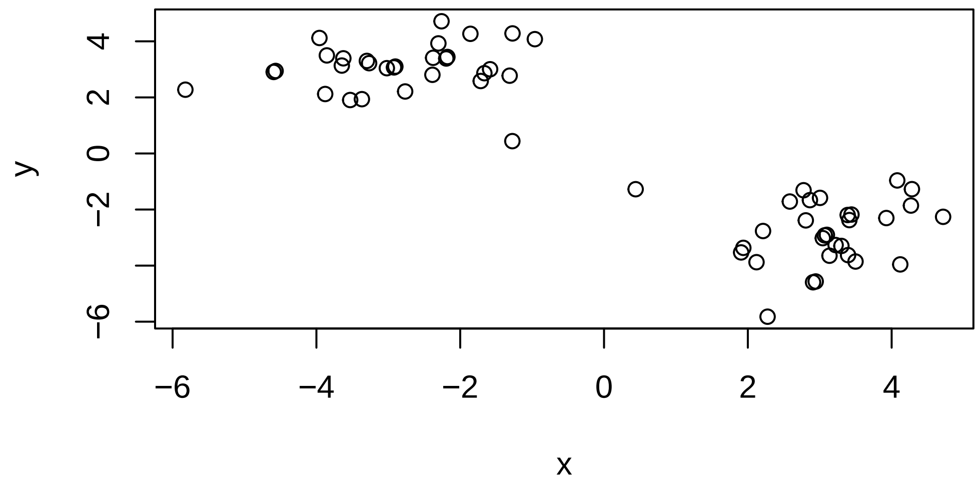

```r
n=30
hist(c(rnorm(n, mean =3), rnorm(n, mean = -3)))
```




```r
x <- c(rnorm(n, mean =3), rnorm(n, mean = -3))
#how to reverse x from y?
y <- rev(x)
y
```

```
 [1] -3.5300058 -1.7150085 -2.2598615 -2.3035355 -3.9575128 -1.3124003
 [7] -3.3671163 -3.6253032 -4.5939929 -3.2983892 -2.1944178 -2.7657015
[13] -4.5671485 -3.0203419 -1.8581458 -0.9620706 -1.6640641 -3.8779396
[19] -3.6467931 -1.2728020 -2.9021023 -1.2750205 -3.8549247 -2.1781451
[25] -2.3867031 -3.2671504 -2.3767540 -5.8221130 -1.5842519 -2.9211504
[31]  3.0700486  3.0038782  2.2746844  3.4117455  3.2205965  2.8059860
[37]  3.4413271  3.4977003  0.4396646  3.0993090  4.2821317  3.1354824
[43]  2.1204680  2.8634697  4.0777640  4.2682230  3.0421943  2.9438097
[49]  2.2113581  3.3883277  3.3013871  2.9068922  3.3939603  1.9372142
[55]  2.7750856  4.1197269  3.9261453  4.7157443  2.5839265  1.9068442
```

```r
z <- cbind(x,y)
plot(z)
```



##K-means clustering

The function for k-means clustering is called `kmeans()`. Run `kmeans()` and assign two centers.

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

Q1. Print out the club membership vector (our main answer)

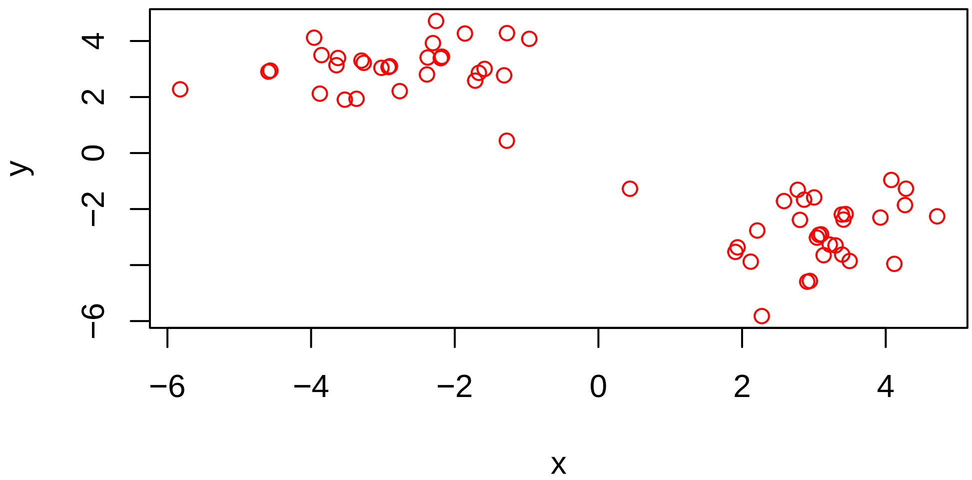
```
km$centers
```

```
          x         y
1  3.072170 -2.812029
2 -2.812029  3.072170
```

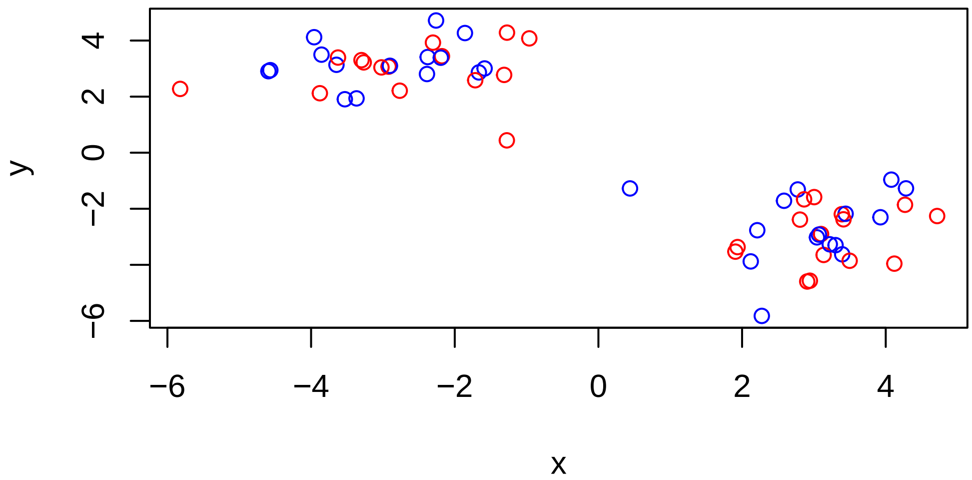
```
km$cluster
```

```
 [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 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
```

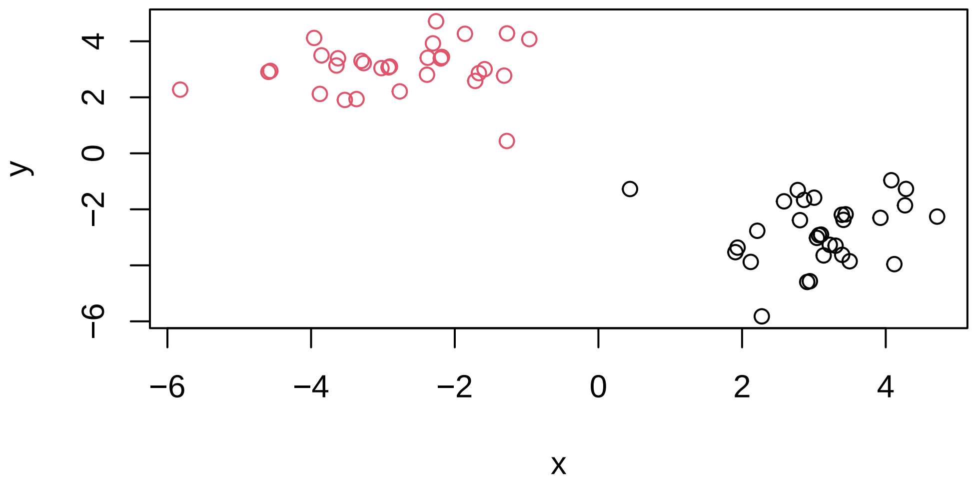
```
plot(z, col = "red")
```




```
plot(z, col = c("red", "blue")) #Think about R as vectors. One vector is two colors against c
```

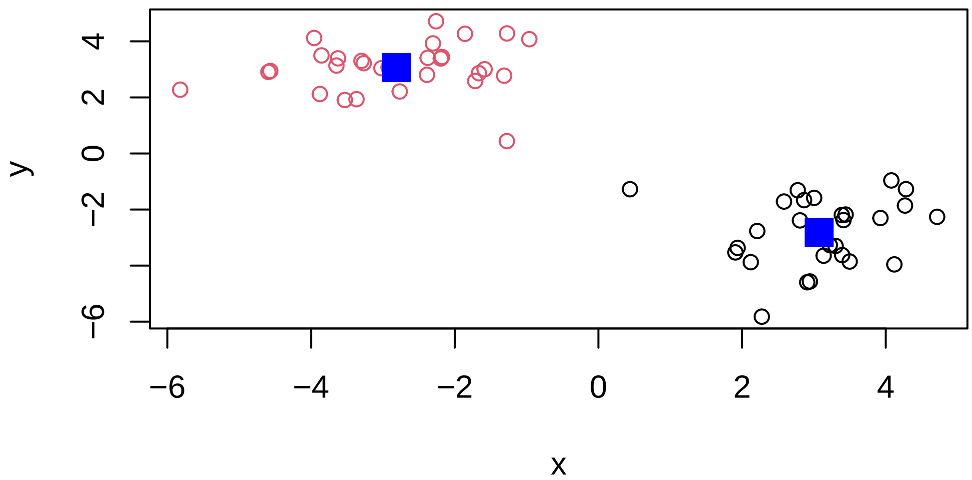


```r
plot(z, col = km$cluster)
```



plot with clustering result and add centers.

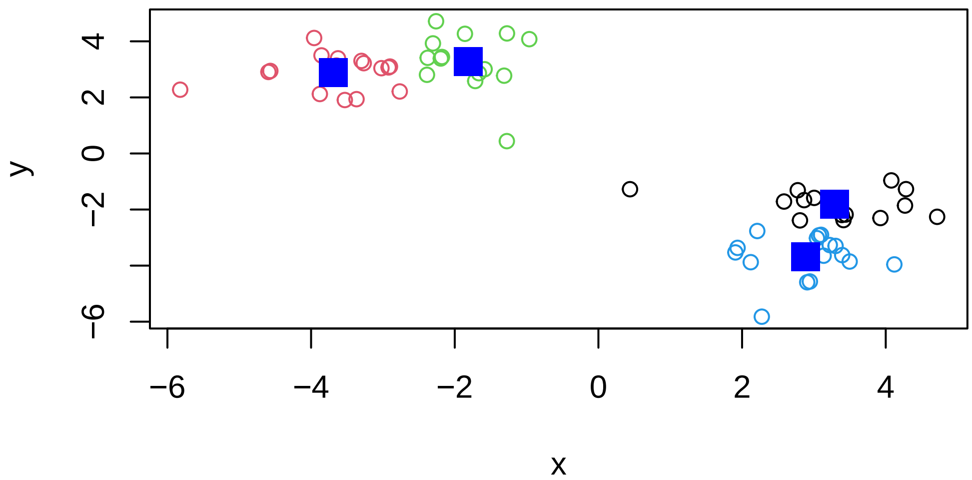
```r
plot(z, col = km$cluster)
points(km$centers, col = "blue", pch = 15, cex = 2)
```



Q. Can you cluster our data in `z` into four clusters

```
km_four <- kmeans(z, 4)
```

```
plot(z, col = km_four$cluster)
points(km_four$centers, col = "blue", pch = 15, cex = 2)
```



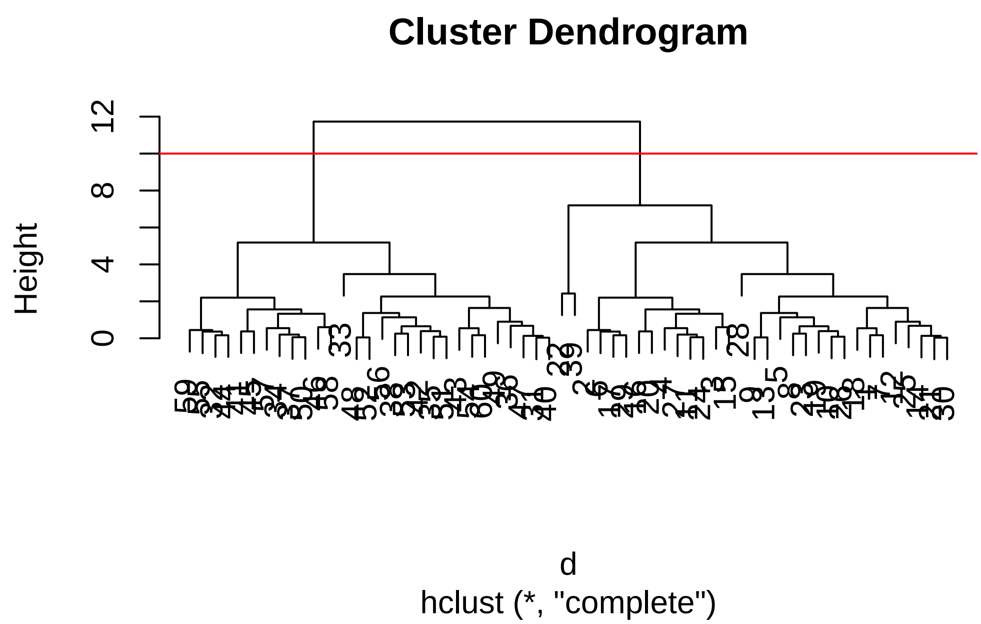

```
kmeans(z,4)
```

```
K-means clustering with 4 clusters of sizes 14, 16, 14, 16

Cluster means:
          x         y
1 -1.810227  3.284530
2 -3.688605  2.886355
3  3.284530 -1.810227
4  2.886355 -3.688605

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

Within cluster sum of squares by cluster:
[1] 17.4404 15.1317 17.4404 15.1317
 (between_SS / total_SS =  94.4 %)

Available components:

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

##Hierarchical Clustering

Function = `hclust()` For `hclust()` we first need a distance matrix from data.

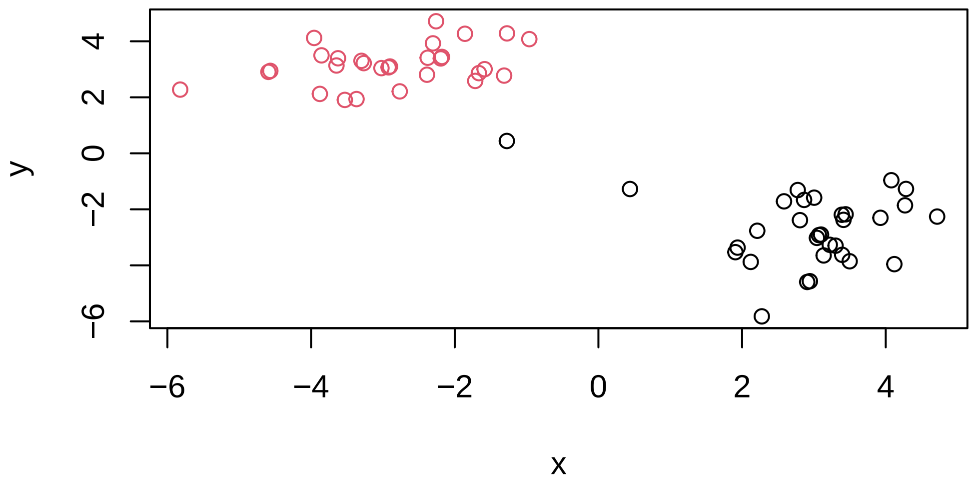
```
d <- dist(z)
hc <- hclust(d)
plot(hc) #gives a plot where data is divided on two main arms (1-30 and 31-60)
abline(h=10, col = "red")
```



To get main clustering results, can "cut" the tree and give height. to do this, use `cutree`.

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

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



##Principle component analysis (PCA)

Take original data, and choose path with most variance and assign it PC1. Then draw PC2 to capture more variance.

```
url <- "https://tinyurl.com/UK-foods"
x <- read.csv(url)
nrow(x) #Q.1 answer
```

```
[1] 17
```

```
ncol(x)# Q.1 answer
```

```
[1] 5
```

```
head(x) # Q.2 answer
```

```
               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
```

```r
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
```

```r
url <- "https://tinyurl.com/UK-foods"
x <- read.csv(url,row.names = 1)
nrow(x) #Q.1 answer
```

```
[1] 17
```

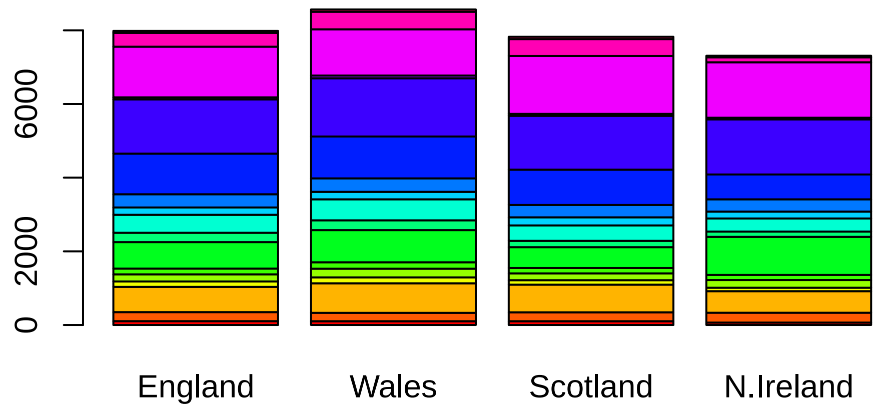
```r
ncol(x)# Q.1 answer
```

```
[1] 4
```

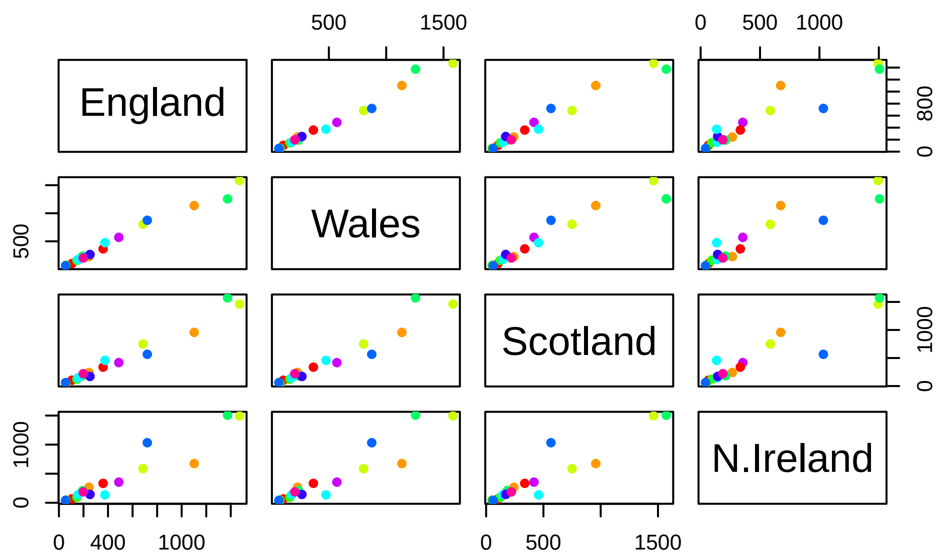
```r
dim(x)
```

```
[1] 17  4
```

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




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



17 variables is not close to how many dimensions we would normally look at. Using PCA

Function for PCA in base R is `prcomp()`

```r
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
```

```r
t(x)
```

```
          Cheese Carcass_meat  Other_meat  Fish Fats_and_oils  Sugars
England      105           245         685   147           193     156
Wales        103           227         803   160           235     175
Scotland     103           242         750   122           184     147
N.Ireland     66           267         586    93           209     139
          Fresh_potatoes  Fresh_Veg  Other_Veg  Processed_potatoes
England              720        253        488                 198
Wales                874        265        570                 203
Scotland             566        171        418                 220
N.Ireland           1033        143        355                 187
          Processed_Veg  Fresh_fruit  Cereals  Beverages Soft_drinks
England             360         1102     1472         57        1374
Wales               365         1137     1582         73        1256
Scotland            337          957     1462         53        1572
```

```
N.Ireland              334          674      1494         47          1506
          Alcoholic_drinks  Confectionery
England                375             54
Wales                  475             64
Scotland               458             62
N.Ireland              135             41
```

```r
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
```

What is inside our result from our object pca

```r
attributes(pca)
```

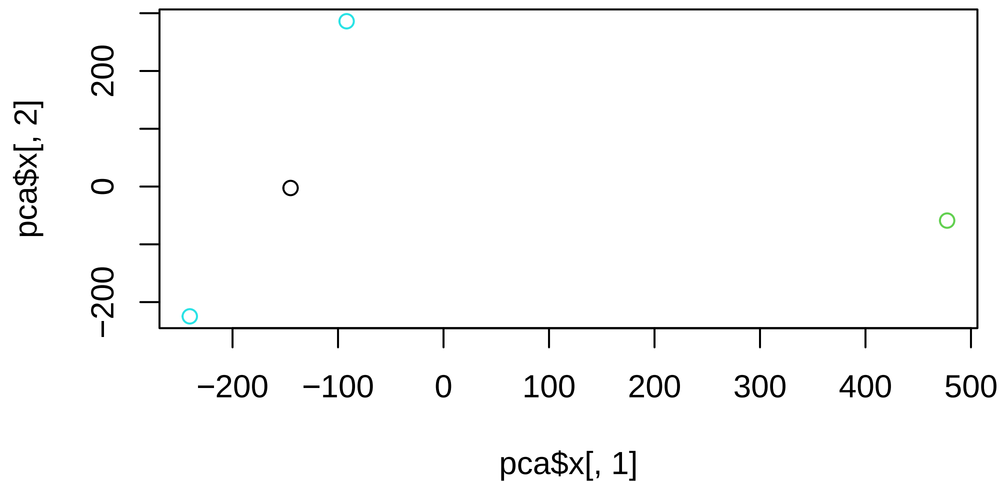
```
$names
[1] "sdev"     "rotation" "center"   "scale"    "x"

$class
[1] "prcomp"
```

```r
pca$x
```

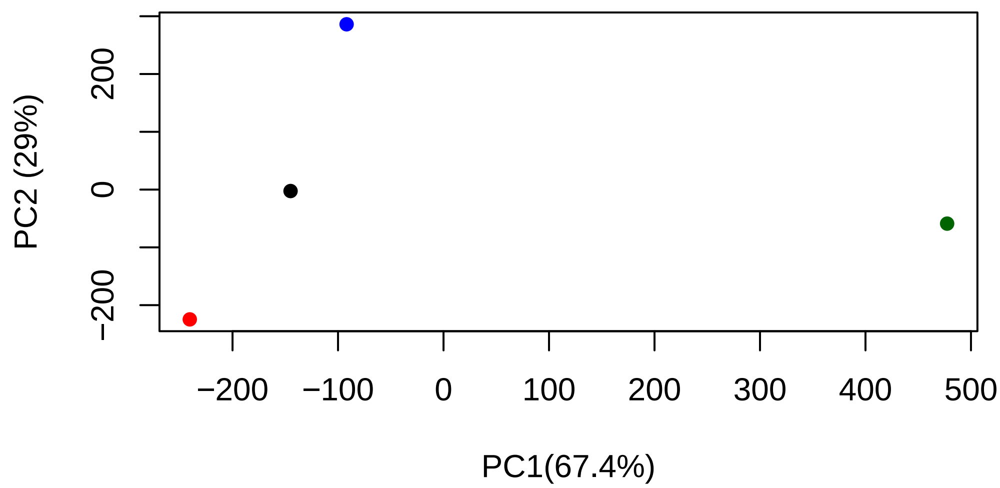
```
                PC1         PC2        PC3           PC4
England   -144.99315   -2.532999 105.768945 -9.152022e-15
Wales     -240.52915 -224.646925 -56.475555  5.560040e-13
Scotland   -91.86934  286.081786 -44.415495 -6.638419e-13
N.Ireland  477.39164  -58.901862  -4.877895  1.329771e-13
```

To make our PC plot/Score plot/ordination plot/PC1/2 plot.

```r
plot(pca$x[,1], pca$x[,2], col = x[,1])
```

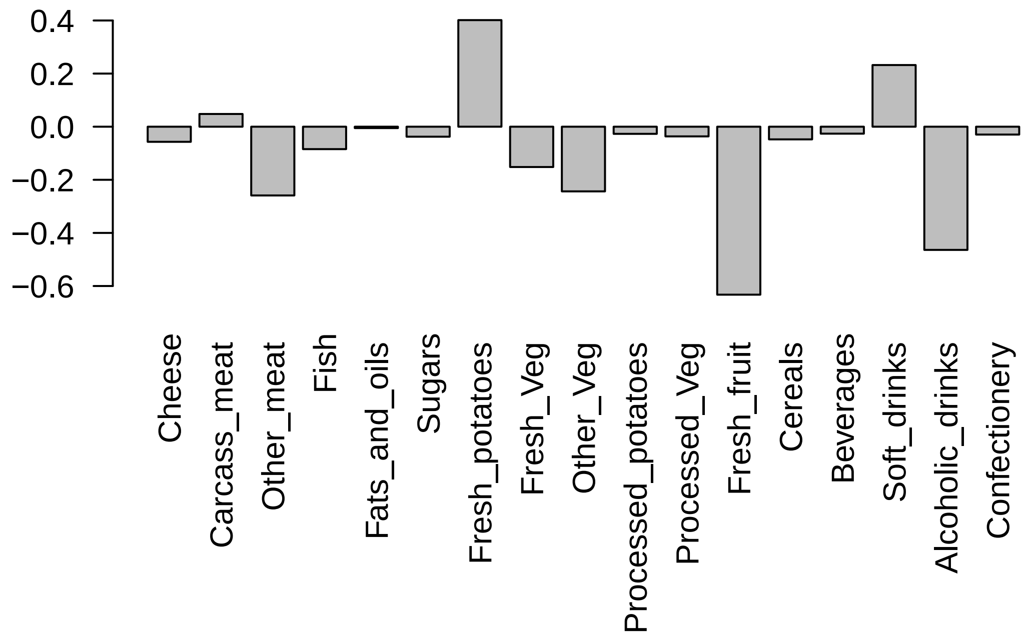



```
plot(pca$x[,1], pca$x[,2], col = c("black", "red", "blue", "darkgreen"), pch = 16, xlab = "PC
```

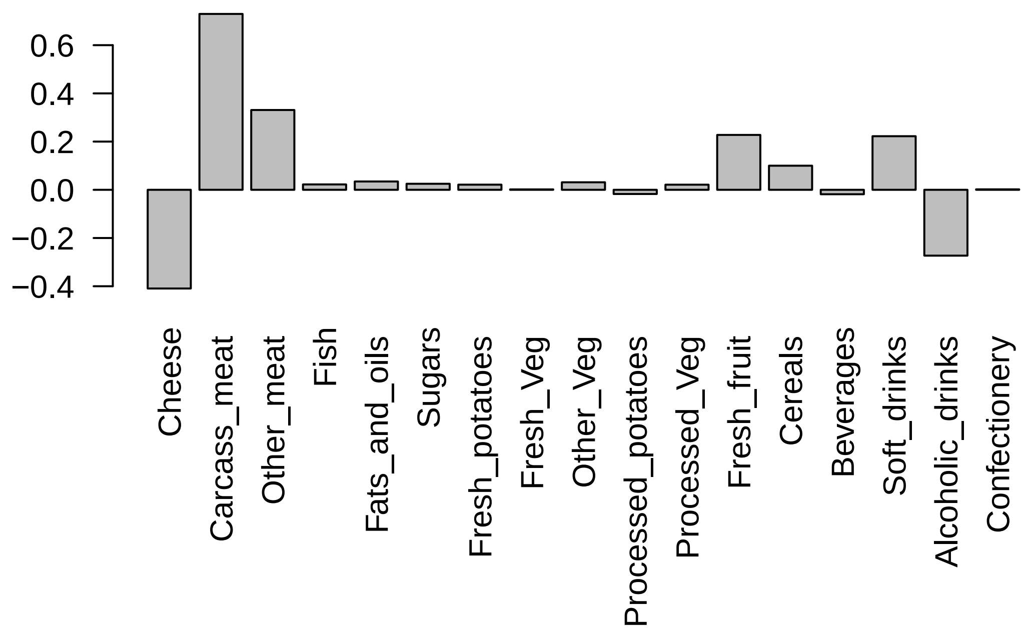


add loadings plot before submitting this.

```r
## 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 )
```




```r
barplot( pca$rotation[,4], las=2 )
```




```
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
```