Class_07_Lab

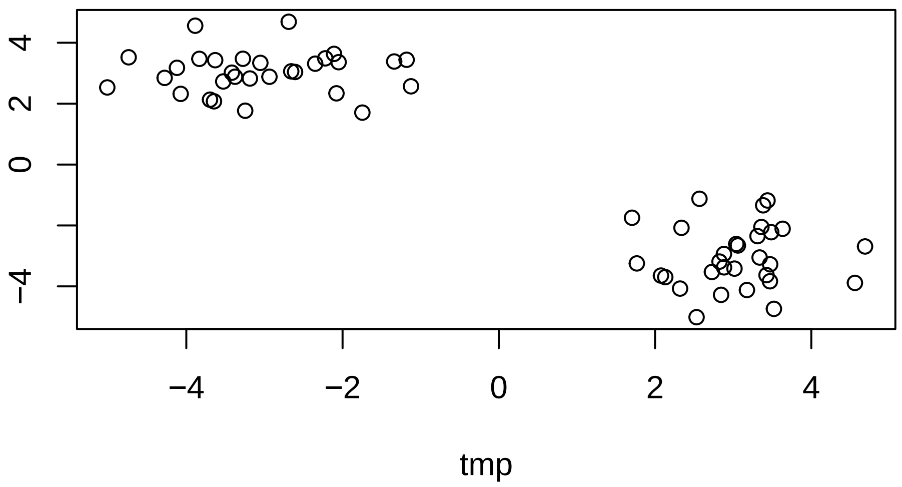
RUNQI ZHANG

Table of contents

```r
library(ggplot2)
```

K-means clustering

```r
# generate some example data for clustering
tmp<-c(rnorm(30,-3),rnorm(30,3))
x<-cbind(tmp, rev(tmp))
plot(x)
```



apply kmeans()

kmeans take two parameters - x and centers

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

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

Cluster means:
        tmp
1 -3.036068  3.033079
2  3.033079 -3.036068

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 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

Within cluster sum of squares by cluster:
[1] 43.49658 43.49658
 (between_SS / total_SS =  92.7 %)
```

```
Available components:

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

Q. How many points are in each cluster?

```
km$size
```

```
[1] 30 30
```

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

```
km$cluster
```

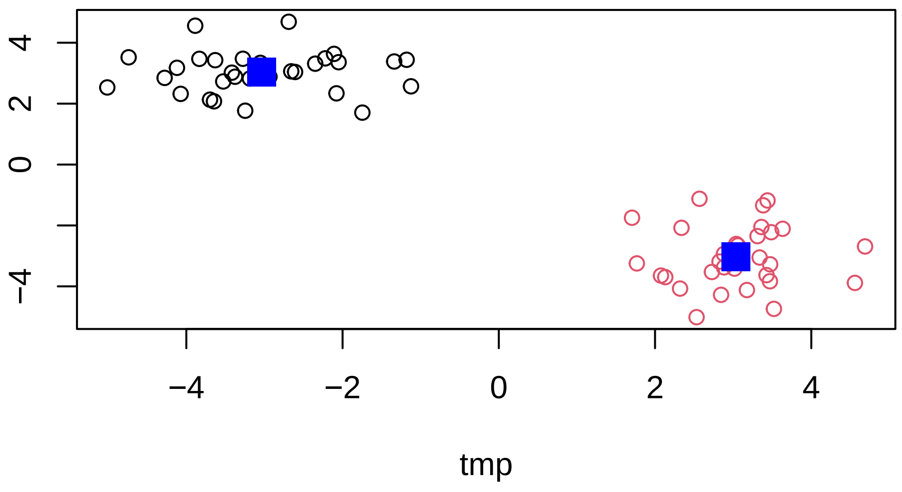
```
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
[39] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

```
km$centers
```

```
        tmp
1 -3.036068  3.033079
2  3.033079 -3.036068
```

Q. Plot x colored by the kmeans cluster assignment and add cluster centers as blue point

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



```
dist_matrix <- dist(x)
dim(dist_matrix)
```

```
NULL
```

```
View( as.matrix(dist_matrix) )
dim(x)
```

```
[1] 60  2
```

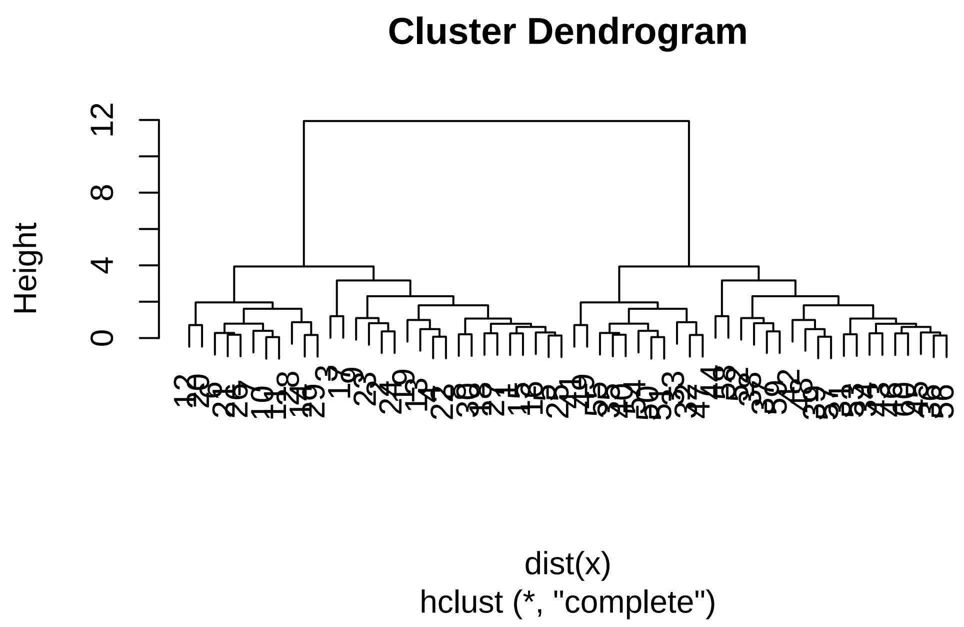
```
dim( as.matrix(dist_matrix) )
```

```
[1] 60 60
```

Hierarchical Clustering

The ‘hclust()’ function requires an input distance matrix

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



use ‘cutree()’ to separate dendrogram to get the cluster membership vector

cutree(hc, k, h) -> k:number of groups, h:height

```r
cutree(hc, h=8)
```

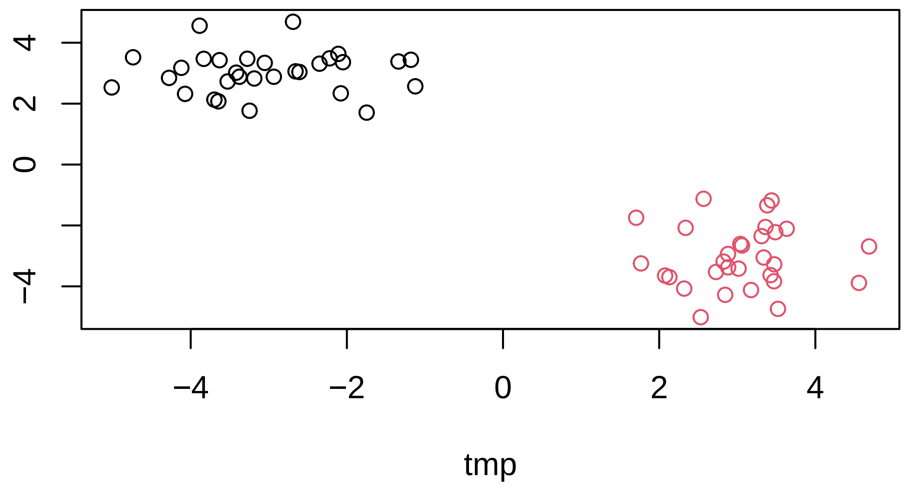
```
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
[39] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

cut into k groups

```r
grps <- cutree(hc,k=2)
```

A plot of our data colored by our hcluster groups

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



PCA

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

```
[1] 17  5
```

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

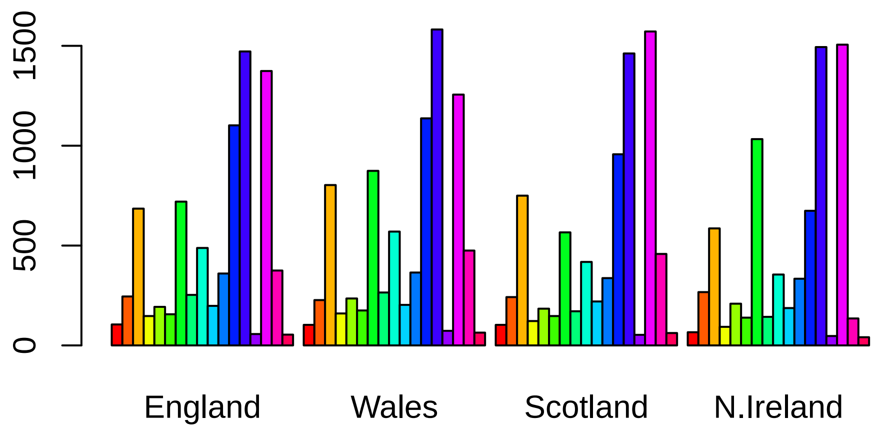
```
               X England Wales Scotland N.Ireland
1         Cheese     105   103      103        66
2  Carcass_meat      245   227      242       267
3    Other_meat      685   803      750       586
4           Fish     147   160      122        93
5 Fats_and_oils      193   235      184       209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

```r
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
```

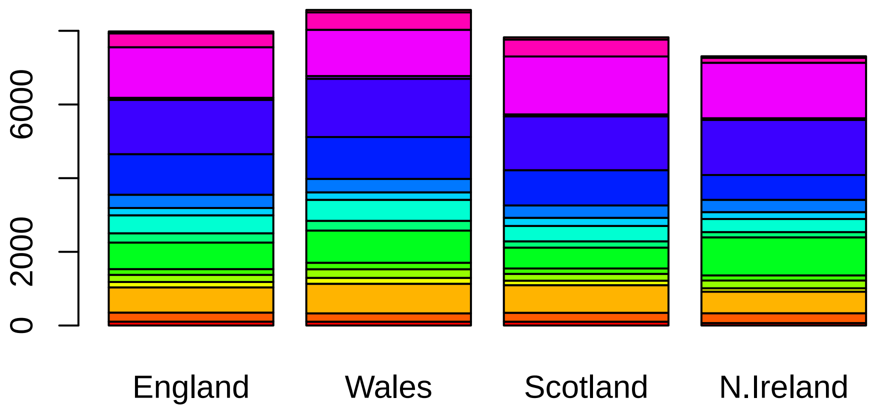
barplot

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

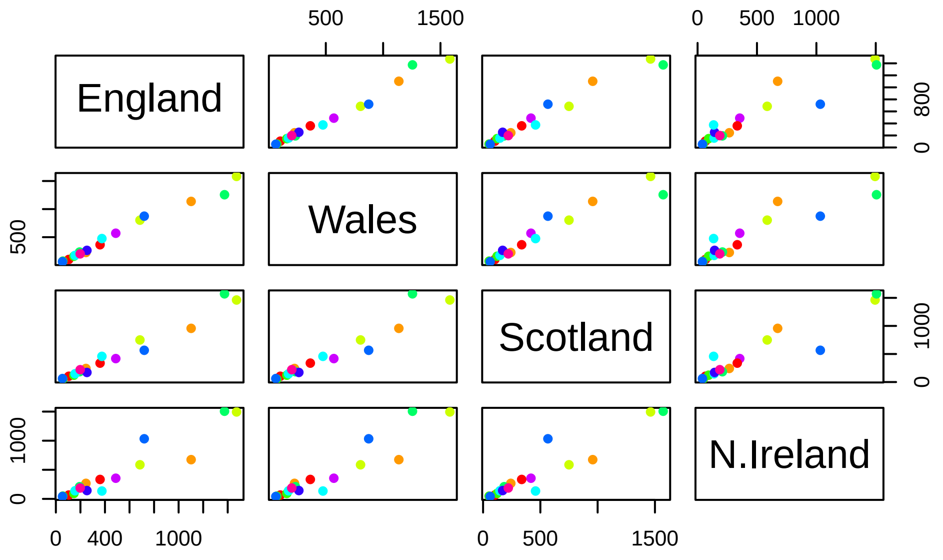


histogram change besdie=T to F

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



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



A#5: The pairs() function compares the datasets across different conditions. If a point falls on the diagonal lines, it is compared with itself.

A#6: People from N.Ireland comsume more whatever the sky-blue is representing and comsume less whatever the navy blue is representing as compared with people from other countries of UK.

PCA to the rescue

```r
# Use the prcomp() PCA function
pca <- prcomp( t(x) )
pca
```

```
Standard deviations (1, .., p=4):
[1] 3.241502e+02 2.127478e+02 7.387622e+01 4.188568e-14

Rotation (n x k) = (17 x 4):
                            PC1          PC2         PC3          PC4
Cheese             -0.056955380 -0.016012850 -0.02394295 -0.691718038
Carcass_meat        0.047927628 -0.013915823 -0.06367111  0.635384915
Other_meat         -0.258916658  0.015331138  0.55384854  0.198175921
Fish               -0.084414983  0.050754947 -0.03906481 -0.015824630
Fats_and_oils      -0.005193623  0.095388656  0.12522257  0.052347444
Sugars             -0.037620983  0.043021699  0.03605745  0.014481347
Fresh_potatoes      0.401402060  0.715017078  0.20668248 -0.151706089
Fresh_Veg          -0.151849942  0.144900268 -0.21382237  0.056182433
Other_Veg          -0.243593729  0.225450923  0.05332841 -0.080722623
Processed_potatoes -0.026886233 -0.042850761  0.07364902 -0.022618707
Processed_Veg      -0.036488269  0.045451802 -0.05289191  0.009235001
Fresh_fruit        -0.632640898  0.177740743 -0.40012865 -0.021899087
Cereals            -0.047702858  0.212599678  0.35884921  0.084667257
Beverages          -0.026187756  0.030560542  0.04135860 -0.011880823
Soft_drinks         0.232244140 -0.555124311  0.16942648 -0.144367046
Alcoholic_drinks   -0.463968168 -0.113536523  0.49858320 -0.115797605
Confectionery      -0.029650201 -0.005949921  0.05232164 -0.003695024
```

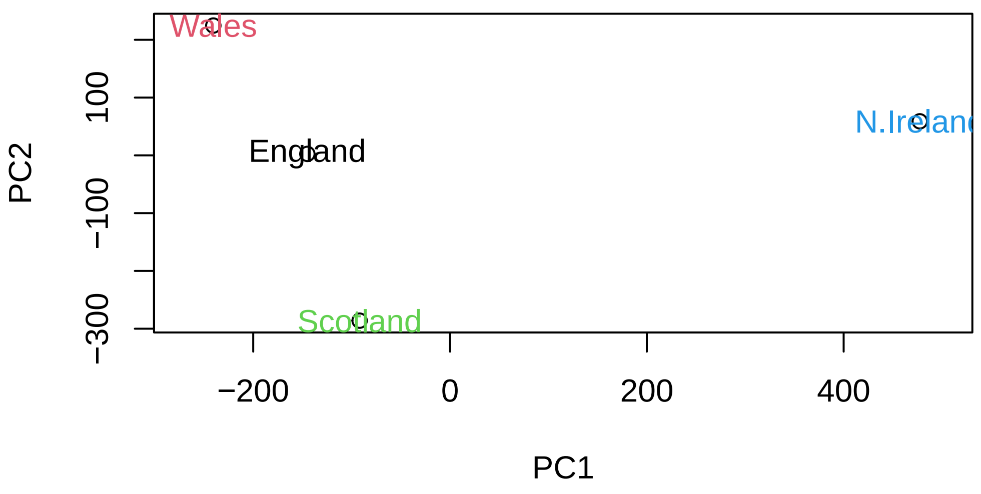
```r
summary(pca)
```

```
Importance of components:
```

```
                             PC1      PC2      PC3       PC4
Standard deviation     324.1502 212.7478 73.87622 4.189e-14
Proportion of Variance   0.6744   0.2905  0.03503 0.000e+00
Cumulative Proportion    0.6744   0.9650  1.00000 1.000e+00
```

Plot PC1 vs PC2

```
# 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), col=c(1,2,3,4))
```



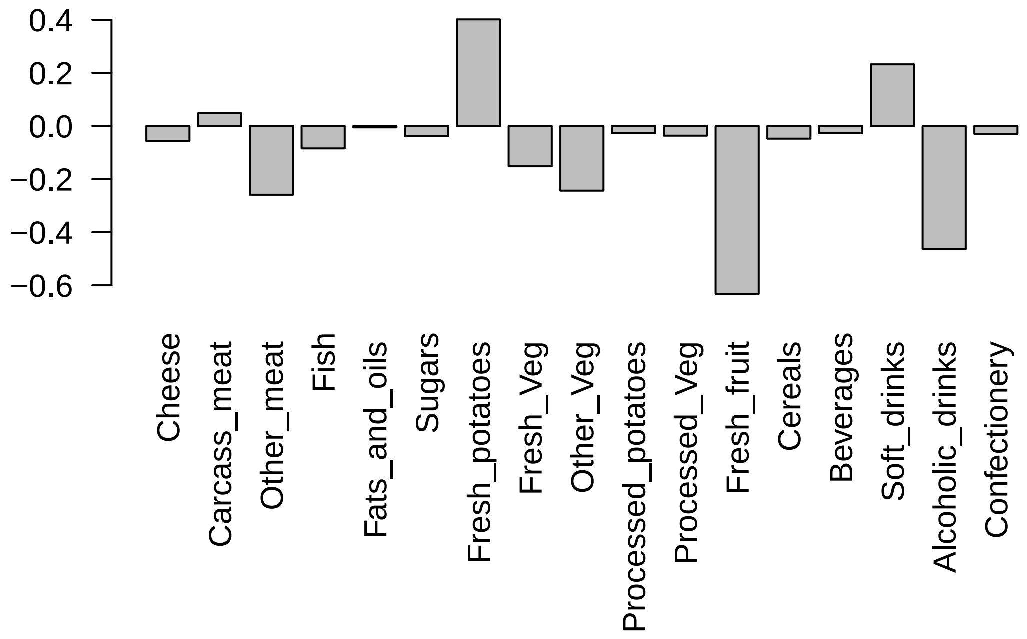
```
v <- round( pca$sdev^2/sum(pca$sdev^2) * 100 )
v
```

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

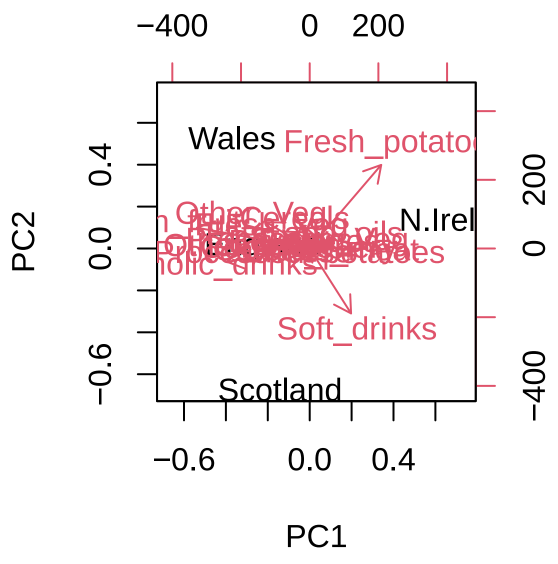
```
z <- summary(pca)
z$importance
```

```
                           PC1       PC2      PC3          PC4
Standard deviation     324.15019 212.74780 73.87622 4.188568e-14
Proportion of Variance   0.67444   0.29052  0.03503 0.000000e+00
Cumulative Proportion    0.67444   0.96497  1.00000 1.000000e+00
```

```
par(mar=c(10, 3, 0.35, 0))
barplot( pca$rotation[,1], las=2 )
```



```
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)
```

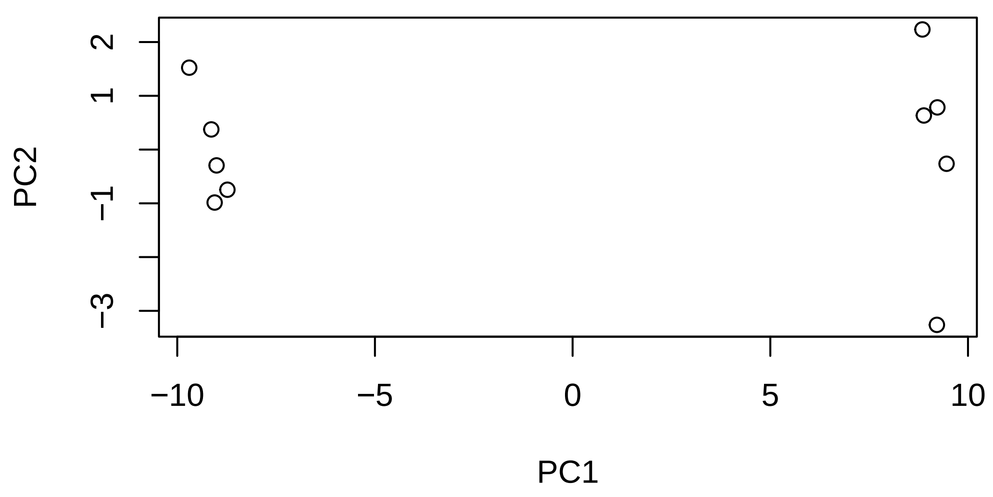
```
       wt1 wt2  wt3  wt4 wt5 ko1 ko2 ko3 ko4 ko5
gene1  439 458  408  429 420  90  88  86  90  93
gene2  219 200  204  210 187 427 423 434 433 426
gene3 1006 989 1030 1017 973 252 237 238 226 210
gene4  783 792  829  856 760 849 856 835 885 894
gene5  181 249  204  244 225 277 305 272 270 279
gene6  460 502  491  491 493 612 594 577 618 638
```

```
dim(rna.data)
```

```
[1] 100  10
```

A: 100 genes, 10 sampels

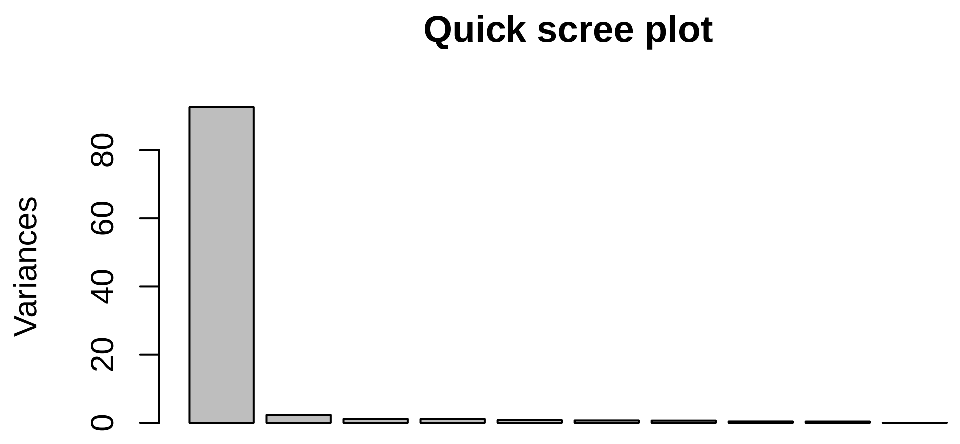
```r
## Again we have to take the transpose of our data
pca <- prcomp(t(rna.data), scale=TRUE)

## Simple un polished plot of pc1 and pc2
plot(pca$x[,1], pca$x[,2], xlab="PC1", ylab="PC2")
```



```r
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 3.348e-15
Proportion of Variance 0.00385 0.00364 0.000e+00
Cumulative Proportion  0.99636 1.00000 1.000e+00
```

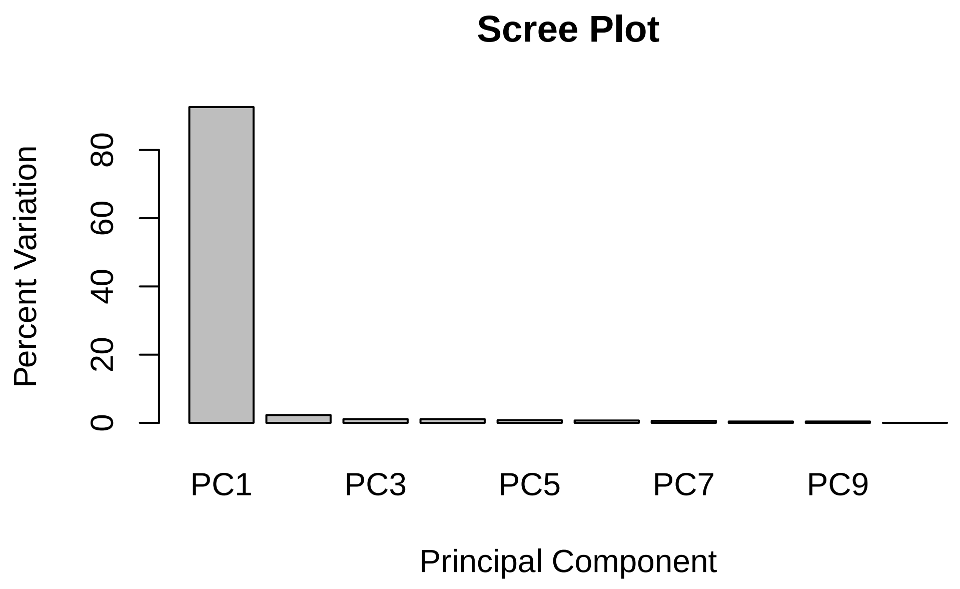
```r
plot(pca, main="Quick scree plot")
```



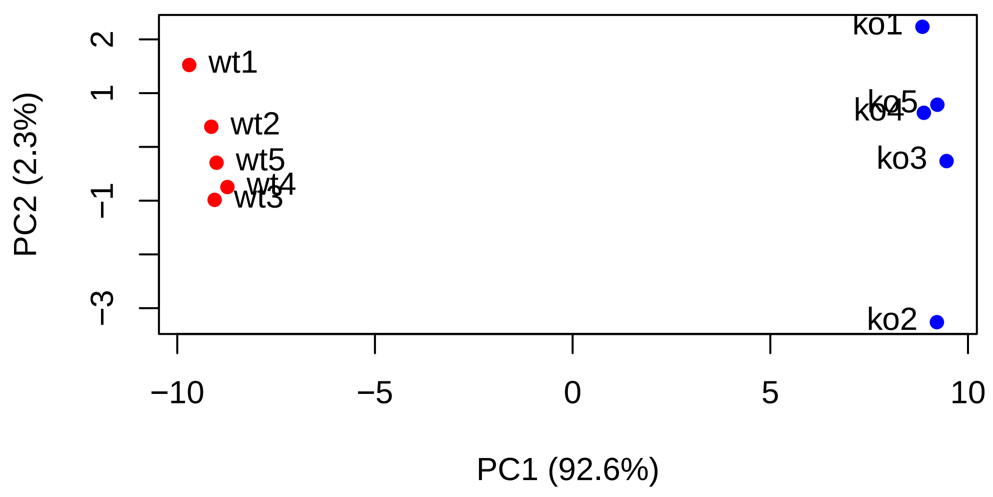
```r
## Variance captured per PC
pca.var <- pca$sdev^2

## Percent variance is often more informative to look at
pca.var.per <- round(pca.var/sum(pca.var)*100, 1)
pca.var.per
```

```
[1] 92.6  2.3  1.1  1.1  0.8  0.7  0.6  0.4  0.4  0.0
```

```r
barplot(pca.var.per, main="Scree Plot",
        names.arg = paste0("PC", 1:10),
        xlab="Principal Component", ylab="Percent Variation")
```



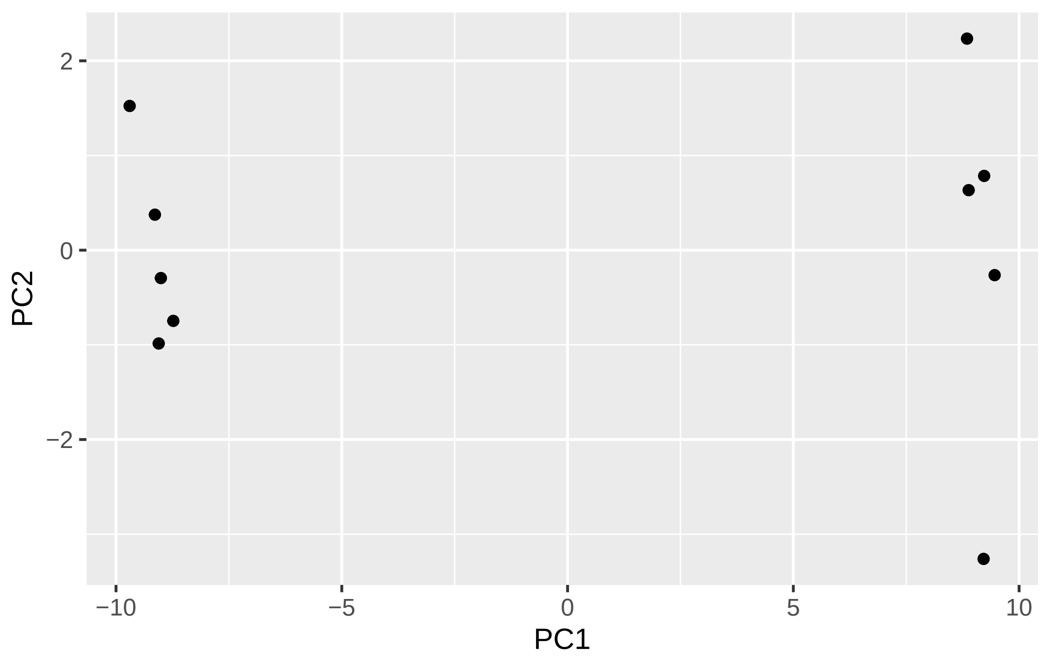
```
## A vector of colors for wt and ko samples
colvec <- colnames(rna.data)
colvec[grep("wt", colvec)] <- "red"
colvec[grep("ko", colvec)] <- "blue"

plot(pca$x[,1], pca$x[,2], col=colvec, pch=16,
     xlab=paste0("PC1 (", pca.var.per[1], "%)"),
     ylab=paste0("PC2 (", pca.var.per[2], "%)"))

text(pca$x[,1], pca$x[,2], labels = colnames(rna.data), pos=c(rep(4,5), rep(2,5)))
```



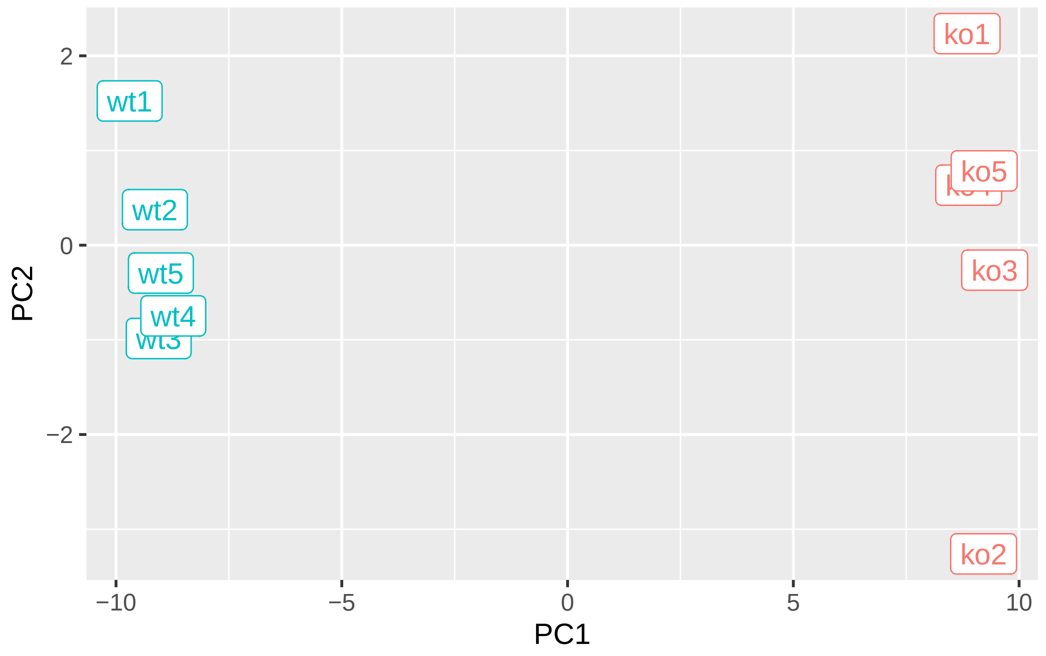
Using ggplot

convert list to dataframe

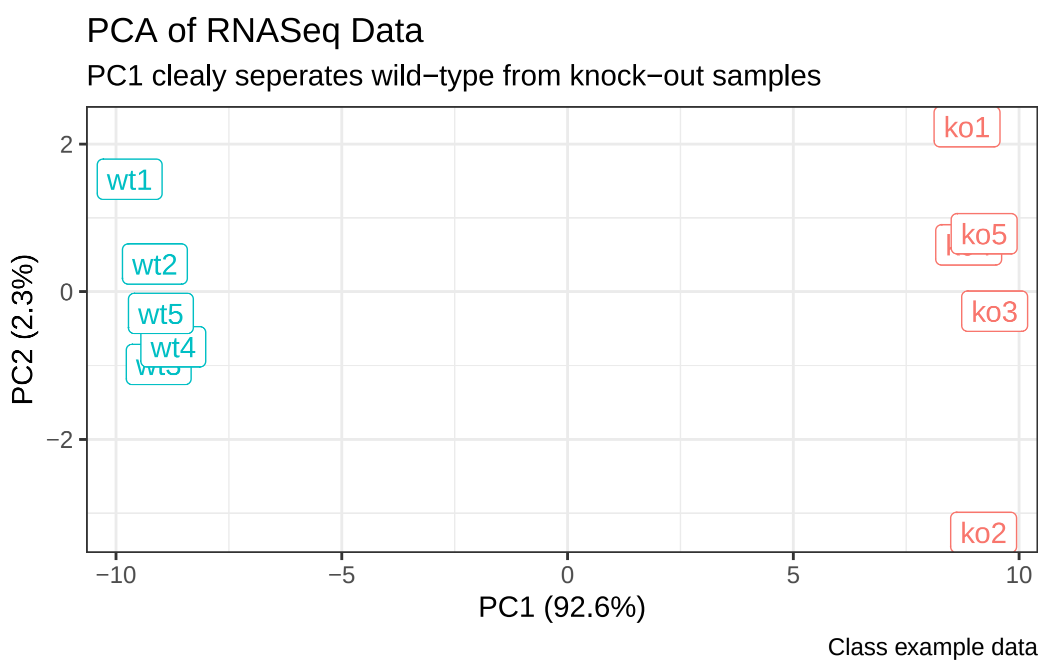
```
df <- as.data.frame(pca$x)

# Our first basic plot
ggplot(df) +
  aes(PC1, PC2) +
  geom_point()
```



```
df$samples <- colnames(rna.data)
df$condition <- substr(colnames(rna.data),1,2)

p <- ggplot(df) +
        aes(PC1, PC2, label=samples, col=condition) +
        geom_label(show.legend = FALSE)
p
```



```
p + labs(title="PCA of RNASeq Data",
       subtitle = "PC1 clealy seperates wild-type from knock-out samples",
       x=paste0("PC1 (", pca.var.per[1], "%)"),
       y=paste0("PC2 (", pca.var.per[2], "%)"),
       caption="Class example data") +
     theme_bw()
```


PCA of RNASeq Data
PC1 clealy seperates wild−type from knock−out samples
ko1
wt1
ko5
wt2
ko3
wt5
wt4
ko2
2
0
−2
−10
−5
0
5
10
PC2 (2.3%)
PC1 (92.6%)
Class example data