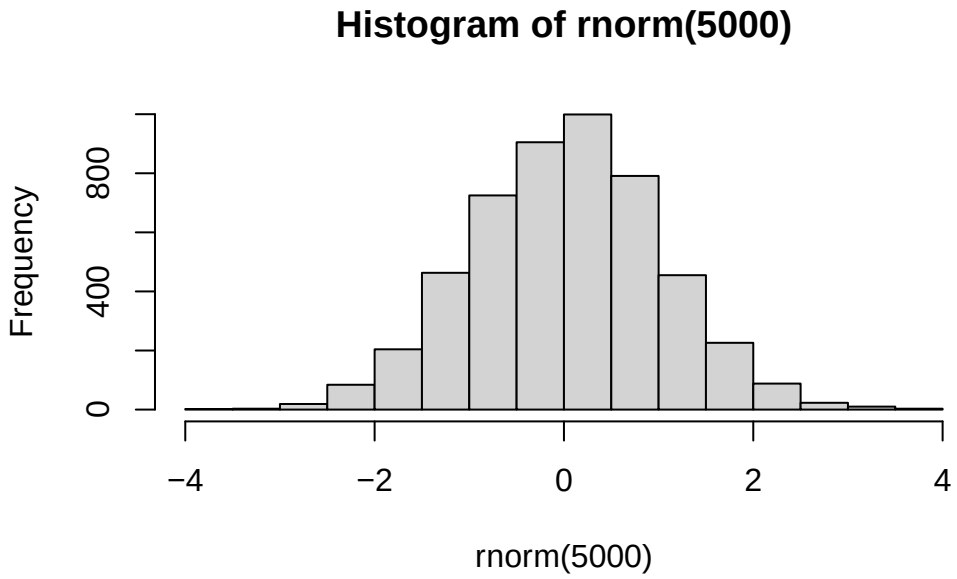
class 7: machine learning

sylvia ho a18482382

##bg clustering, dimensionality reduction

##k means clustering "clusters" `rnorm()`

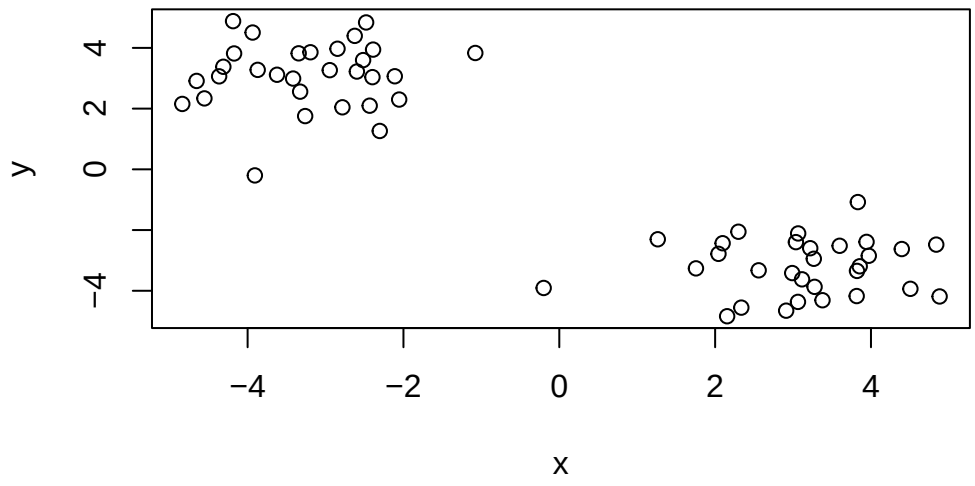
```
hist(rnorm(5000))
```




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

```
 [1] 3.6664574 3.0171237 4.5207654 0.5704894 2.4170378 2.8148909 3.7740016
 [8] 2.7567373 1.9051900 2.2958380 2.2411753 4.3655513 3.4285454 2.7247873
[15] 3.0201837 1.5774233 3.7012486 2.2486771 4.0910954 4.5882391 3.5329253
[22] 3.0239822 4.1709375 3.7265683 2.1094200 3.0735996 2.1221220 1.7193812
[29] 3.9888972 3.1733041
```

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



##k means clustering `kmeans()`

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

```r
km$size
```

```
[1] 30 30
```

centers

```r
km$centers
```

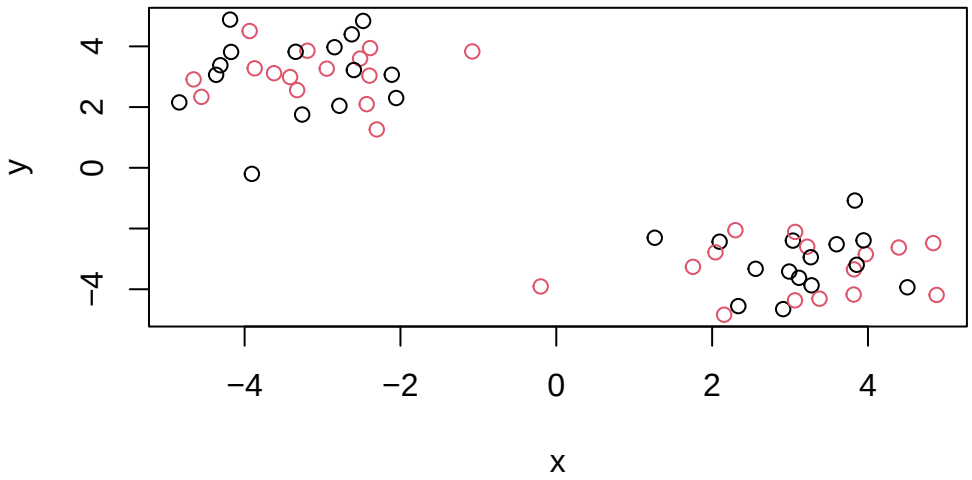
```
          x         y
1  3.102726 -3.217674
2 -3.217674  3.102726
```

membership

```
km$cluster
```

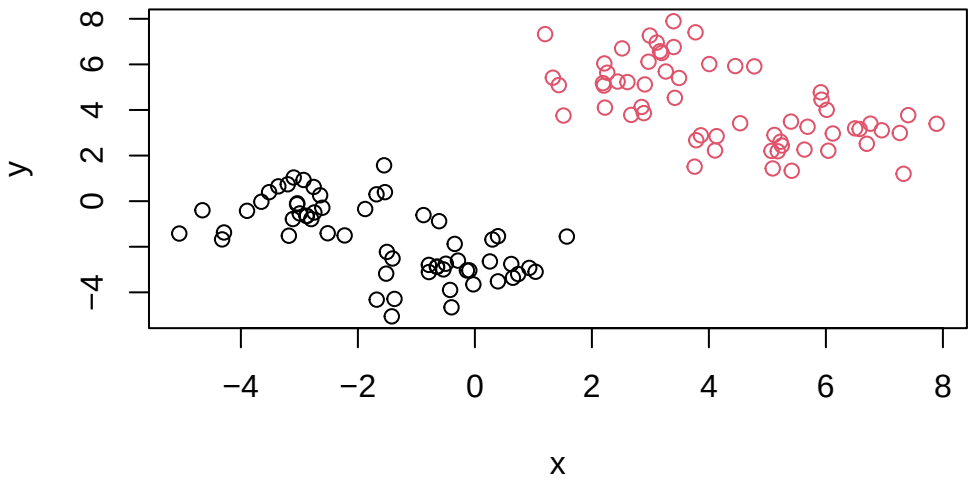
```
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
[39] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

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



`kmeans()` 4 clusters and figure

```
tmp <- c(rnorm(30, mean = 3),
         rnorm(30, mean = -3),
         rnorm(30, mean = 0),
         rnorm(30, mean = 6))
x <- cbind(x = tmp, y = rev(tmp))
k4 <- kmeans(x, centers = 2)
plot(x, col = k4$cluster)
```




```
km$tot.withinss
```

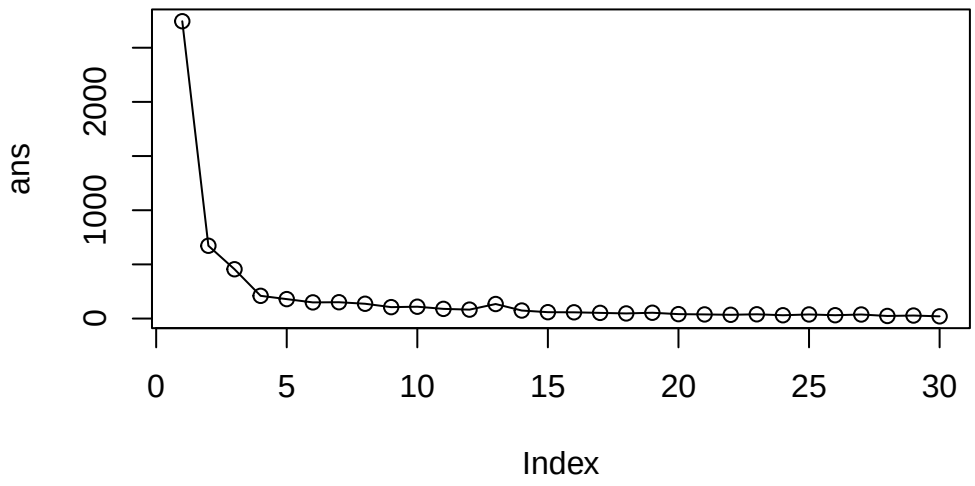
```
[1] 118.3162
```

```
k4$tot.withinss
```

```
[1] 672.0438
```

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

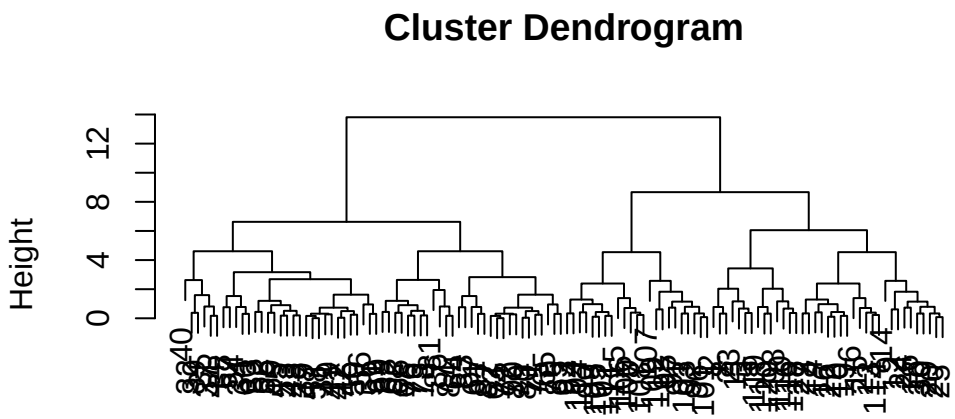
```
plot(ans,typ="o")
```



key pt k means will impose clustering structure on ata even if not there

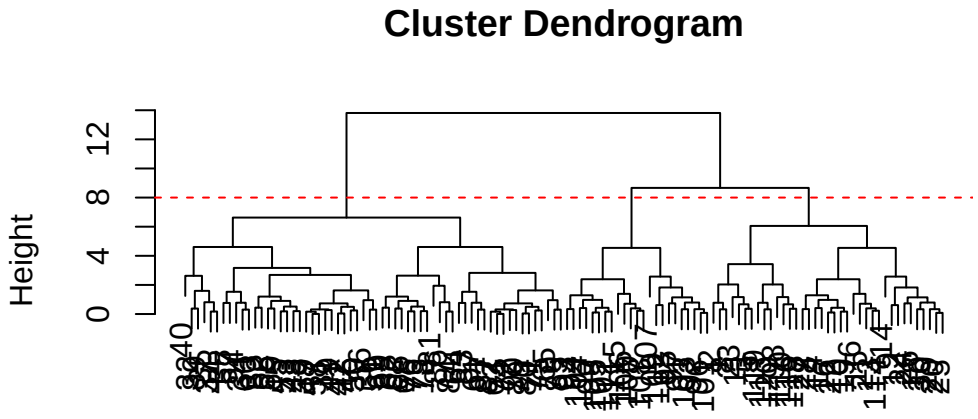
##hierarchical clusteral main fn `hclust()` unlike kmeans (does allw ork for u) hclust needs a "distance matrix" like returned from `dist()` fn could be seq id, anything

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



cut tree to uilt drp grps / branches to extract cluster membershp vector

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



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

```
  [1] 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 3
 [38] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [75] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2
[112] 2 1 1 2 1 2 1 1 1
```

pca of uk food data

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

```
dim(x)
```

```
[1] 17  5
```

```
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

better read csv

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

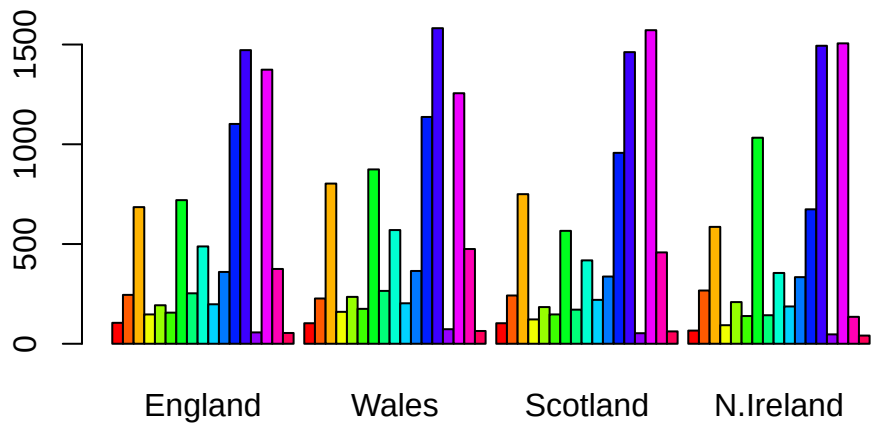
	England	Wales	Scotland	N.Ireland
Cheese	105	103	103	66
Carcass_meat	245	227	242	267
Other_meat	685	803	750	586
Fish	147	160	122	93

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

```
rainbow(4)
```

```
[1] "#FF0000" "#80FF00" "#00FFFF" "#8000FF"
```

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

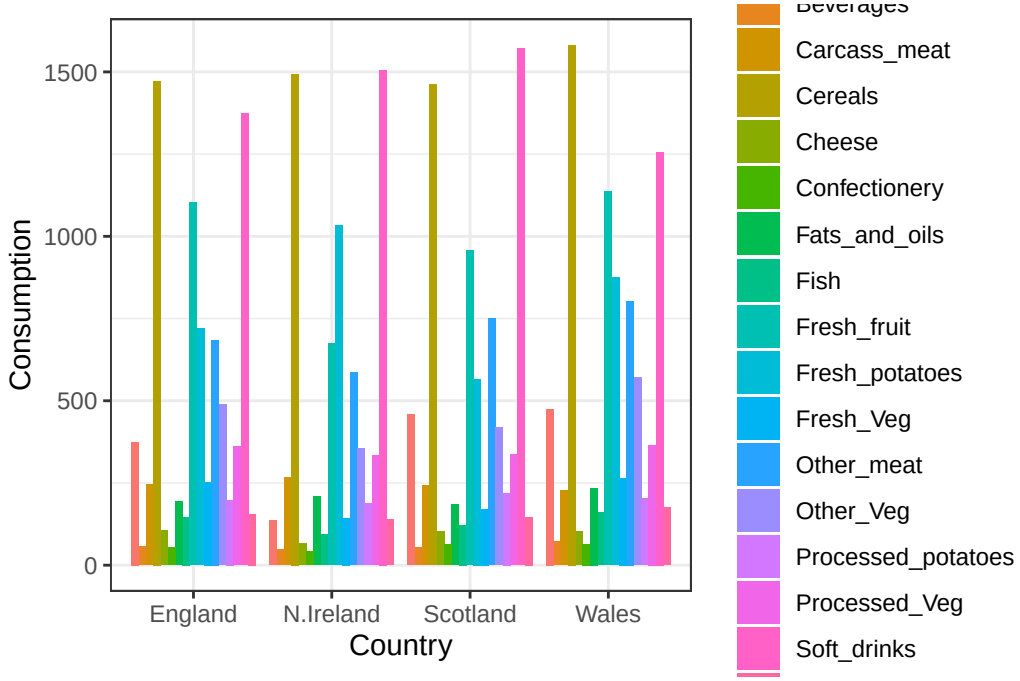


```
library(tidyr)

# Convert data to long format for ggplot with `pivot_longer()`
x_long <- x |>
          tibble::rownames_to_column("Food") |>
          pivot_longer(cols = -Food,
                       names_to = "Country",
                       values_to = "Consumption")

dim(x_long)
```

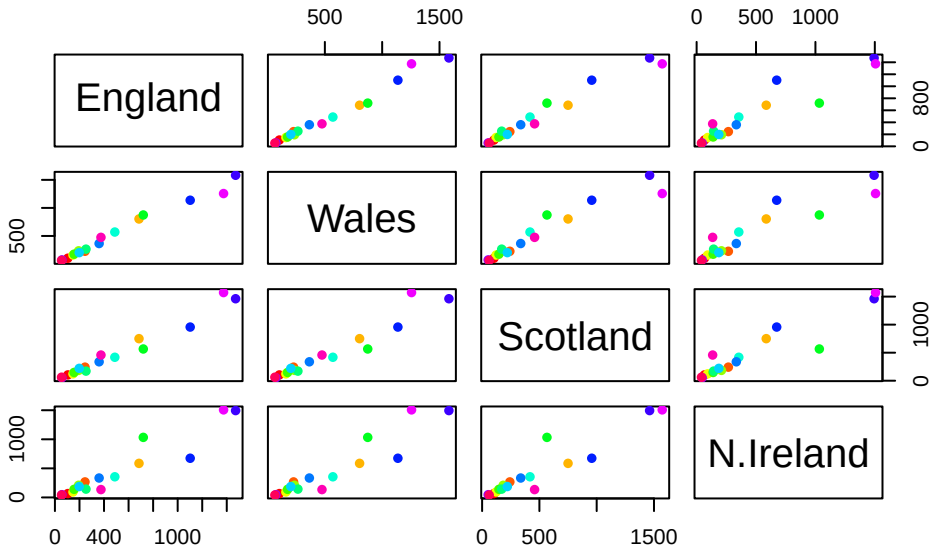
```
[1] 68  3
```

```
library(ggplot2)
```

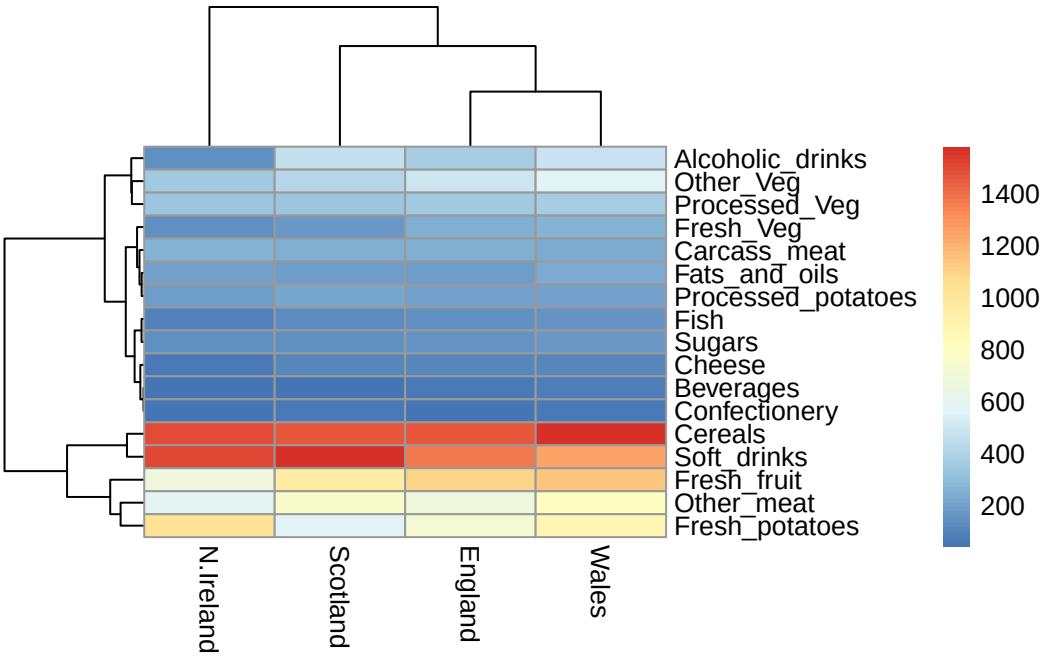
```
ggplot(x_long) +
  aes(x = Country, y = Consumption, fill = Food) +
  geom_col(position = "dodge") +
  theme_bw()
```



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




```
library(pheatmap)

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



main pca fn in base r is `prcomp()` fn wants transpose of food data asinput (food as cols and countries as rows)

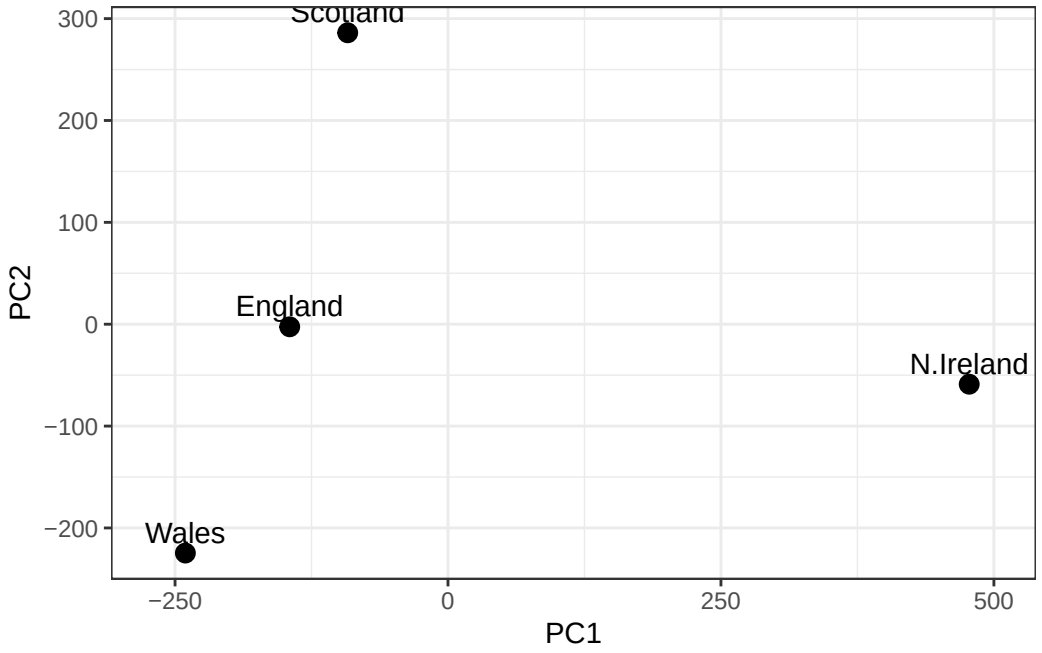
```
pca <- prcomp( t(x) )
summary(pca)
```

```
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
```

`pca$x` scores along w new pcs, called pc plot or score plot, ordination plot

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

ggplot(pca$x) +
  aes(x = PC1, y = PC2, label = rownames(pca$x)) +
  geom_point(size = 3) +
  geom_text(vjust = -0.5) +
  xlim(-270, 500) +
```

```
  xlab("PC1") +
  ylab("PC2") +
  theme_bw()
```




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

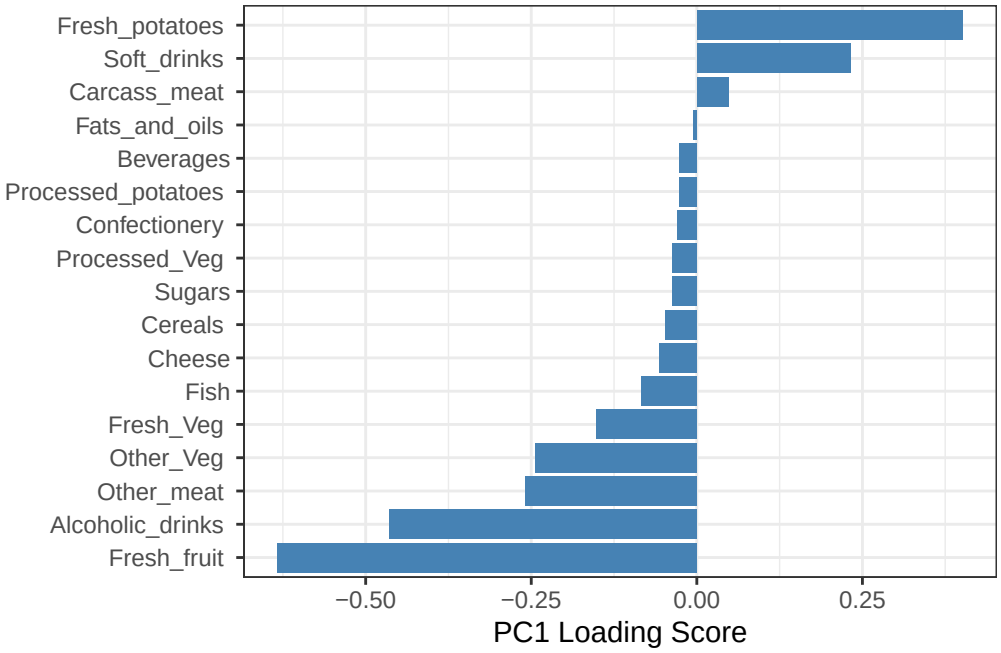
```
[1] 67 29  4  0
```

```
attributes(pca)
```

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

$class
[1] "prcomp"
```

loadings plot

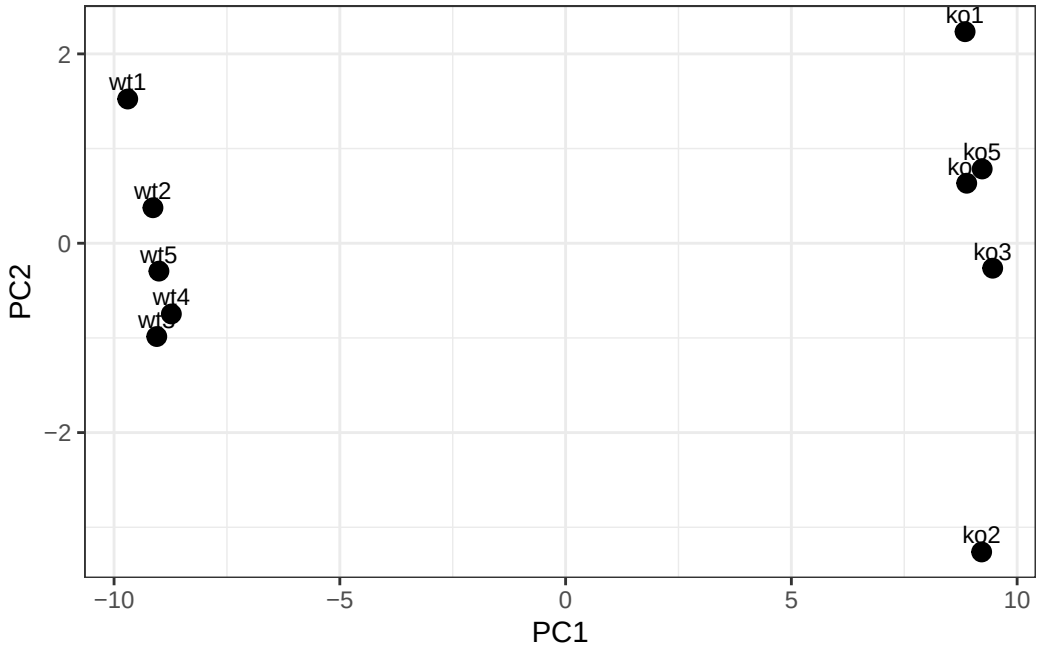
```
ggplot(pca$rotation) +
  aes(x = PC1,
      y = reorder(rownames(pca$rotation), PC1)) +
  geom_col(fill = "steelblue") +
  xlab("PC1 Loading Score") +
  ylab("") +
  theme_bw() +
  theme(axis.text.y = element_text(size = 9))
```




```
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
```

```r
pca <- prcomp(t(rna.data), scale=TRUE)

# Create data frame for plotting
df <- as.data.frame(pca$x)
df$Sample <- rownames(df)

## Plot with ggplot
ggplot(df) +
  aes(x = PC1, y = PC2, label = Sample) +
  geom_point(size = 3) +
  geom_text(vjust = -0.5, size = 3) +
  xlab("PC1") +
  ylab("PC2") +
  theme_bw()
```



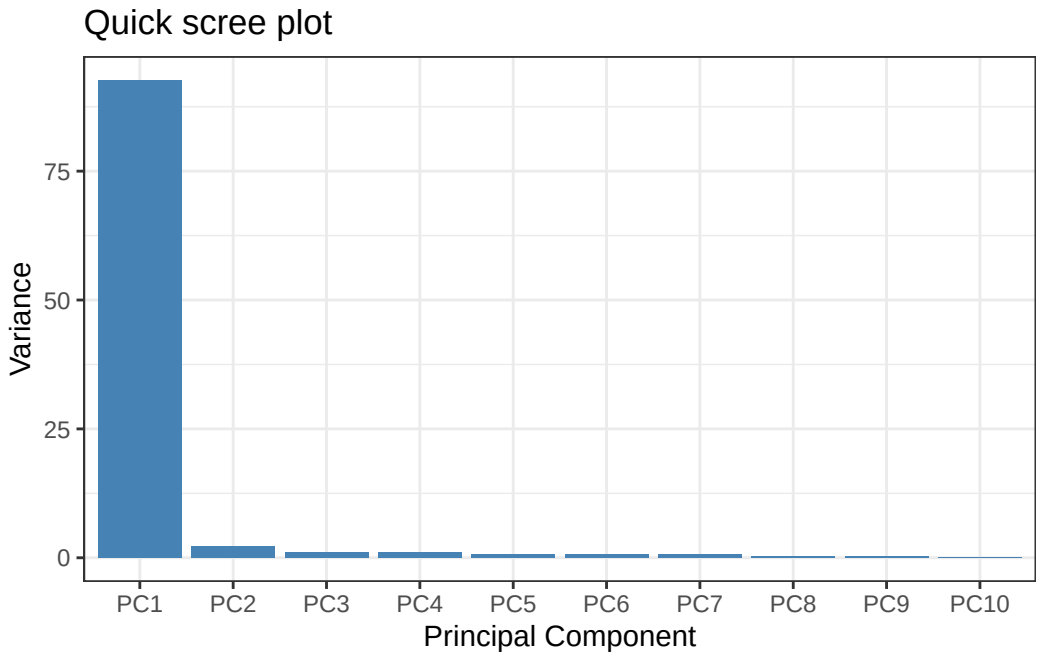

```r
summary(pca)
```

```
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
```

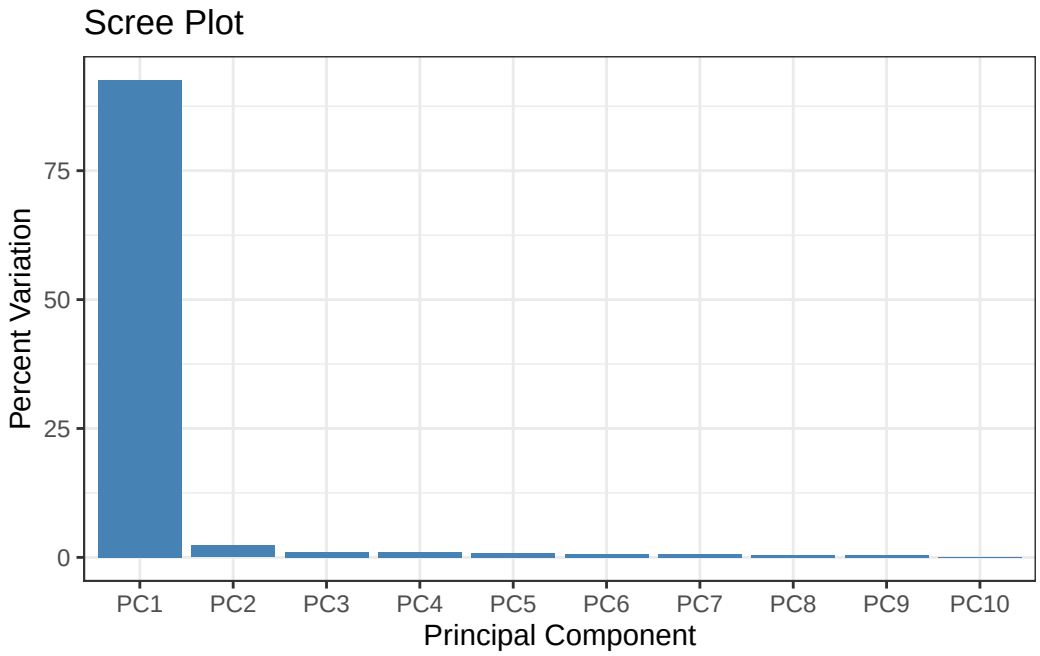
```
                            PC8     PC9     PC10
Standard deviation     0.62065 0.60342 3.39e-15
Proportion of Variance 0.00385 0.00364 0.00e+00
Cumulative Proportion  0.99636 1.00000 1.00e+00
```

percent var

```r
pca.var <- pca$sdev^2
pca.var.per <- round(pca.var/sum(pca.var)*100, 1)

# Create scree plot data
scree_df <- data.frame(
  PC = factor(paste0("PC", 1:10), levels = paste0("PC", 1:10)),
  Variance = pca.var[1:10]
)

ggplot(scree_df) +
  aes(x = PC, y = Variance) +
  geom_col(fill = "steelblue") +
  ggtitle("Quick scree plot") +
  xlab("Principal Component") +
  ylab("Variance") +
  theme_bw()
```



```
scree_pct_df <- data.frame(
  PC = factor(paste0("PC", 1:10), levels = paste0("PC", 1:10)),
  PercentVariation = pca.var.per[1:10]
)

ggplot(scree_pct_df) +
  aes(x = PC, y = PercentVariation) +
  geom_col(fill = "steelblue") +
  ggtitle("Scree Plot") +
  xlab("Principal Component") +
  ylab("Percent Variation") +
  theme_bw()
```




```
colvec <- colnames(rna.data)
colvec[grep("wt", colvec)] <- "red"
colvec[grep("ko", colvec)] <- "blue"

# Add condition to data frame
df$condition <- substr(df$Sample, 1, 2)
df$color <- colvec

ggplot(df) +
```

```
  aes(x = PC1, y = PC2, color = color, label = Sample) +
  geom_point(size = 3) +
  geom_text(vjust = -0.5, hjust = 0.5, show.legend = FALSE) +
  scale_color_identity() +
  xlab(paste0("PC1 (", pca.var.per[1], "%)")) +
  ylab(paste0("PC2 (", pca.var.per[2], "%)")) +
  theme_bw()
```