Machine Learning Lab 7

Sam Altshuler (PID: A59010373)

2/10/2022

PCA of UK food data

Import the UK foods dataset

```
url <- "https://tinyurl.com/UK-foods"
x <- read.csv(url)
# How many rows and columns does it have?
dim(x)
```

```
## [1] 17  5
```

The dataset has 17 rows and 5 columns. However the first column is the rownames. ## Check the data

```
head(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
```

Set the first column as rownames

```
rownames(x) <- x[,1]
#remove the first row since it was set to the rownames
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
```

Now see what `dim()` returns

```
dim(x)
```

```
## [1] 17  4
```

You could also rename the rownames in the initial read.csv() by specifying which columns are the names.

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

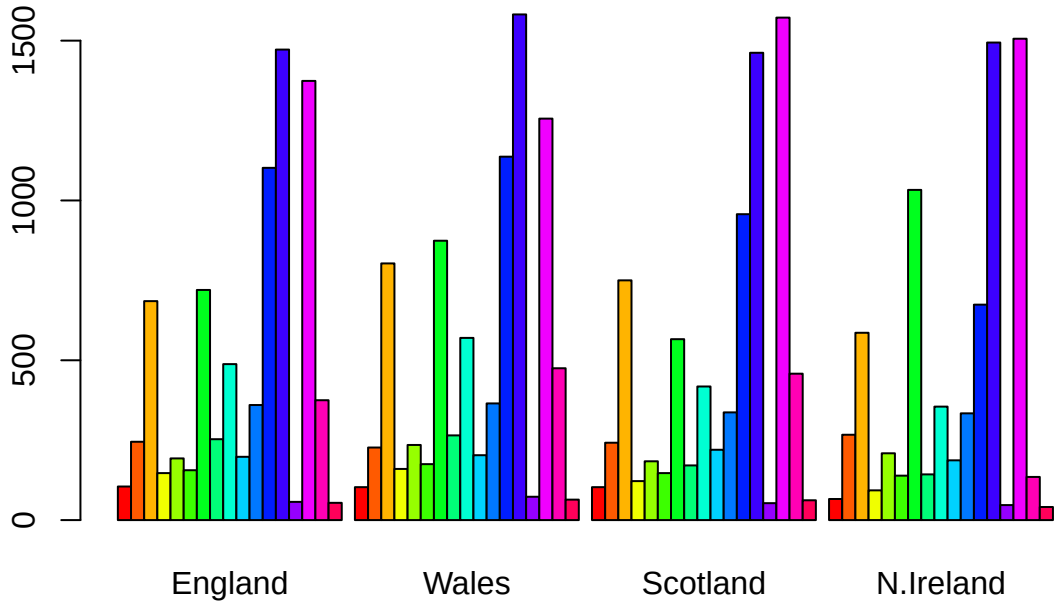
```
##                England Wales Scotland N.Ireland
## Cheese             105   103      103        66
## Carcass_meat       245   227      242       267
## Other_meat         685   803      750       586
## Fish               147   160      122        93
## Fats_and_oils      193   235      184 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

This way is prefered so that you don't run the risk of deleting any of the data when doing the first style of data transformation.

Spotting major differences and trends

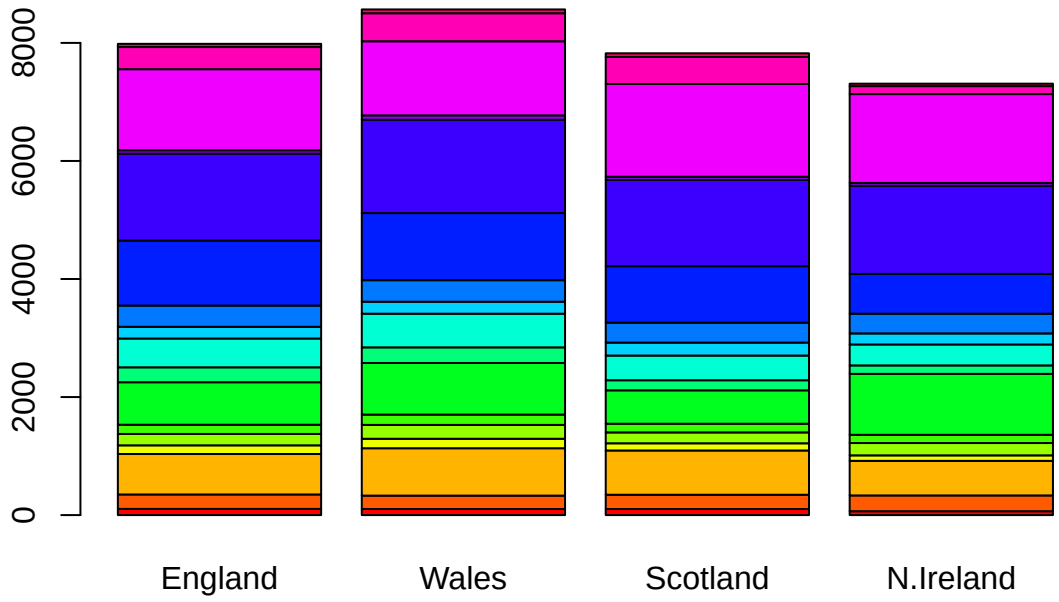
Barplot of the data, it's hard to get any valuable information between different components.

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



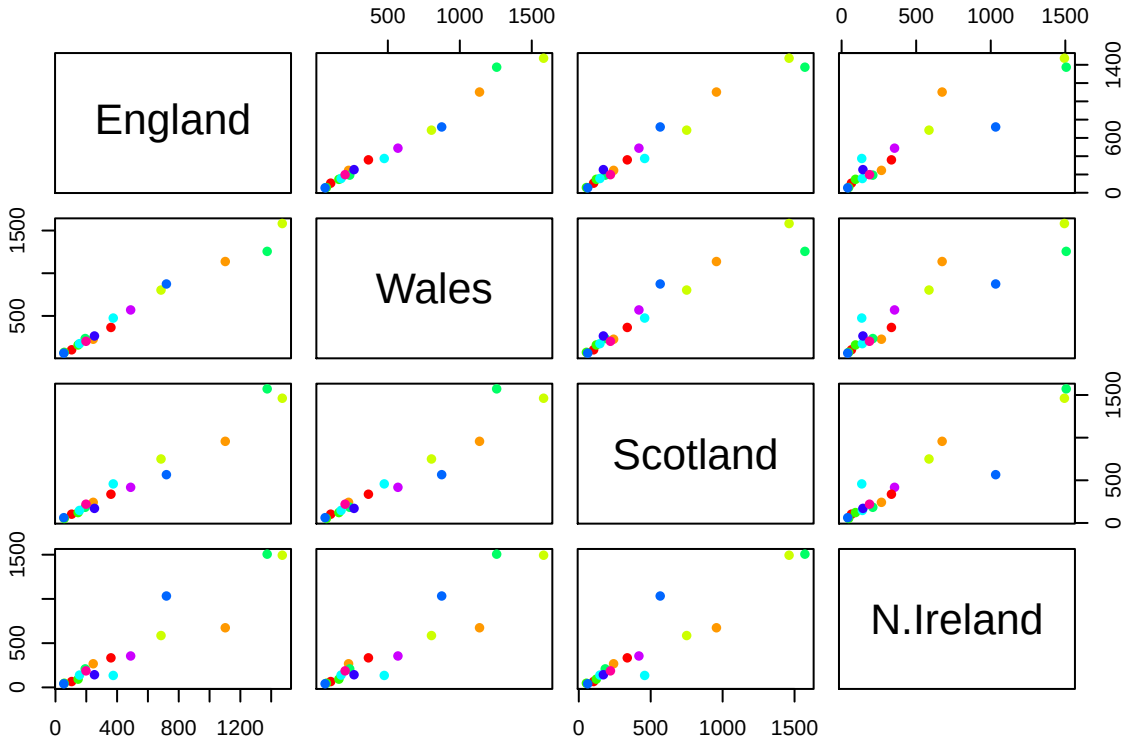
Change the barplot to have them stacked for each country.

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



Pairwise plots might be helpful

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



This plot compares between the two countries the amount of consumption per category. If a value is on the diagonal line, it means they have the same amount of consumption for that food category. This is a fold-change of zero. However, it's still hard to determine main differences between one country from the other countries.

PCA to the rescue

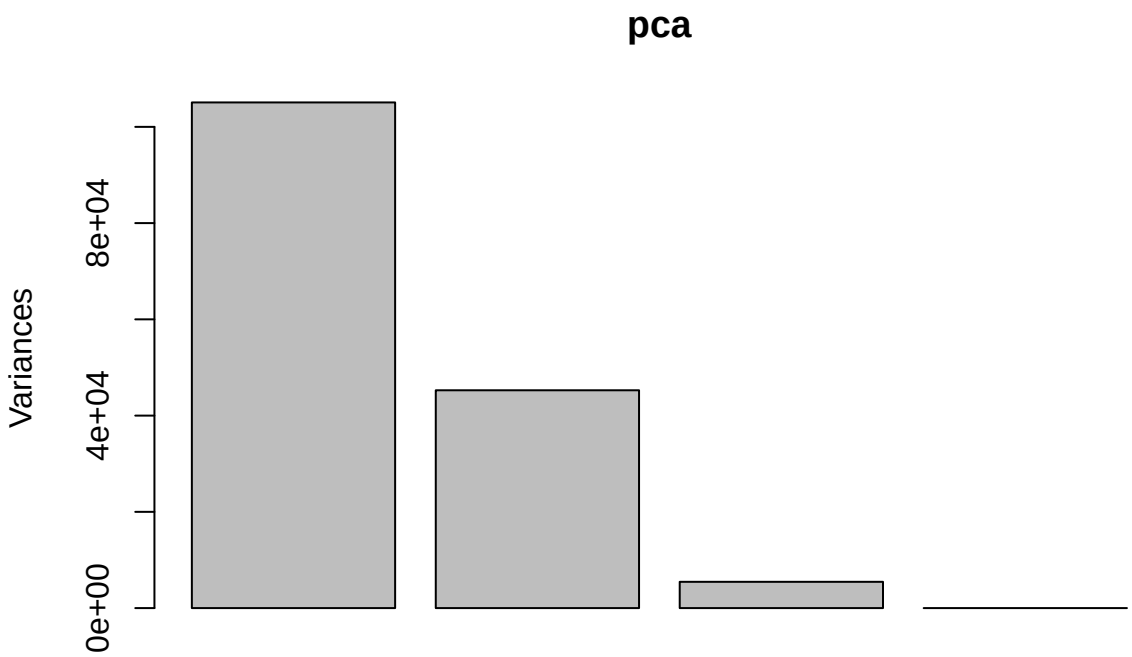
Do PCA of this 17 dimension UK food data. The main function in base R is called `prcomp()`

```
# Need to transpose x to make it in the correct format for prcomp()
pca <- prcomp(t(x))
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
```

The `prcomp()` function returns a list of objects

```
plot(pca)
```



The "PCA plot" is also known as a pca score plot. It is a plot of PC1 v PC2. Basically a new PCA axis to view the data.

```
attributes(pca)
```

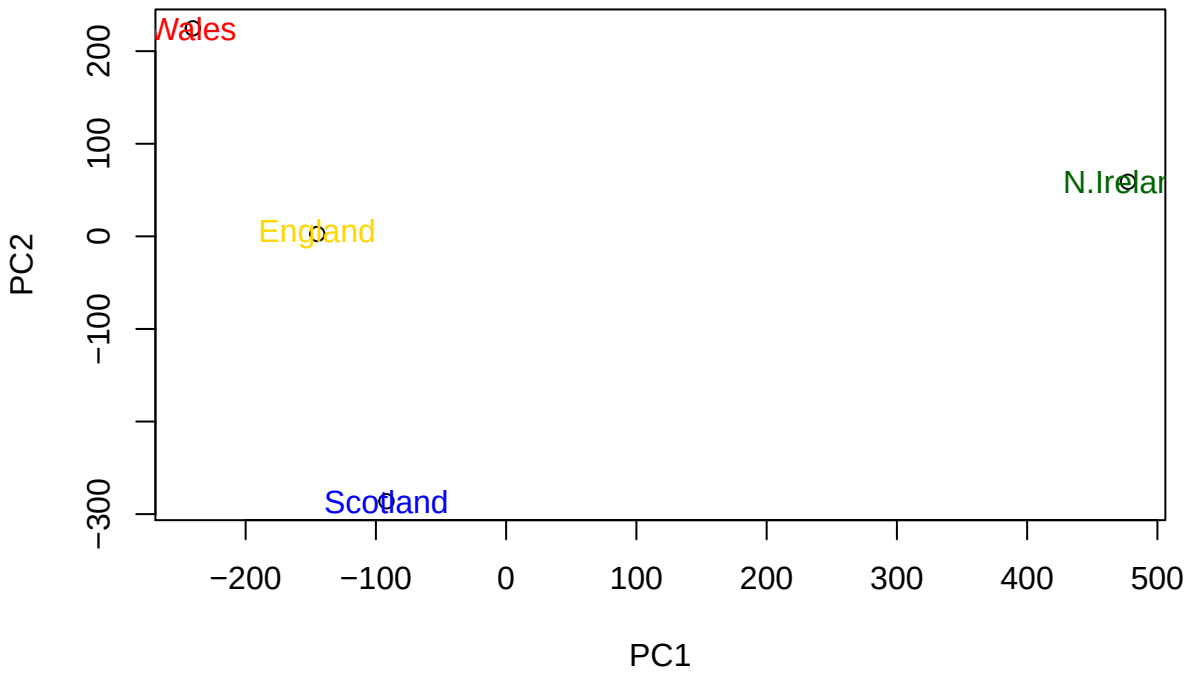
```
## $names
## [1] "sdev"     "rotation" "center"   "scale"    "x"
##
## $class
## [1] "prcomp"
```

Pay attention to the "x" attribute which is a matrix of the data (`pca$x`).

```
pca$x
```

```
##                  PC1         PC2         PC3           PC4
## England   -144.99315    2.532999 -105.768945  2.842865e-14
## Wales     -240.52915  224.646925   56.475555  7.804382e-13
## Scotland   -91.86934 -286.081786   44.415495 -9.614462e-13
## N.Ireland  477.39164   58.901862    4.877895  1.448078e-13
```

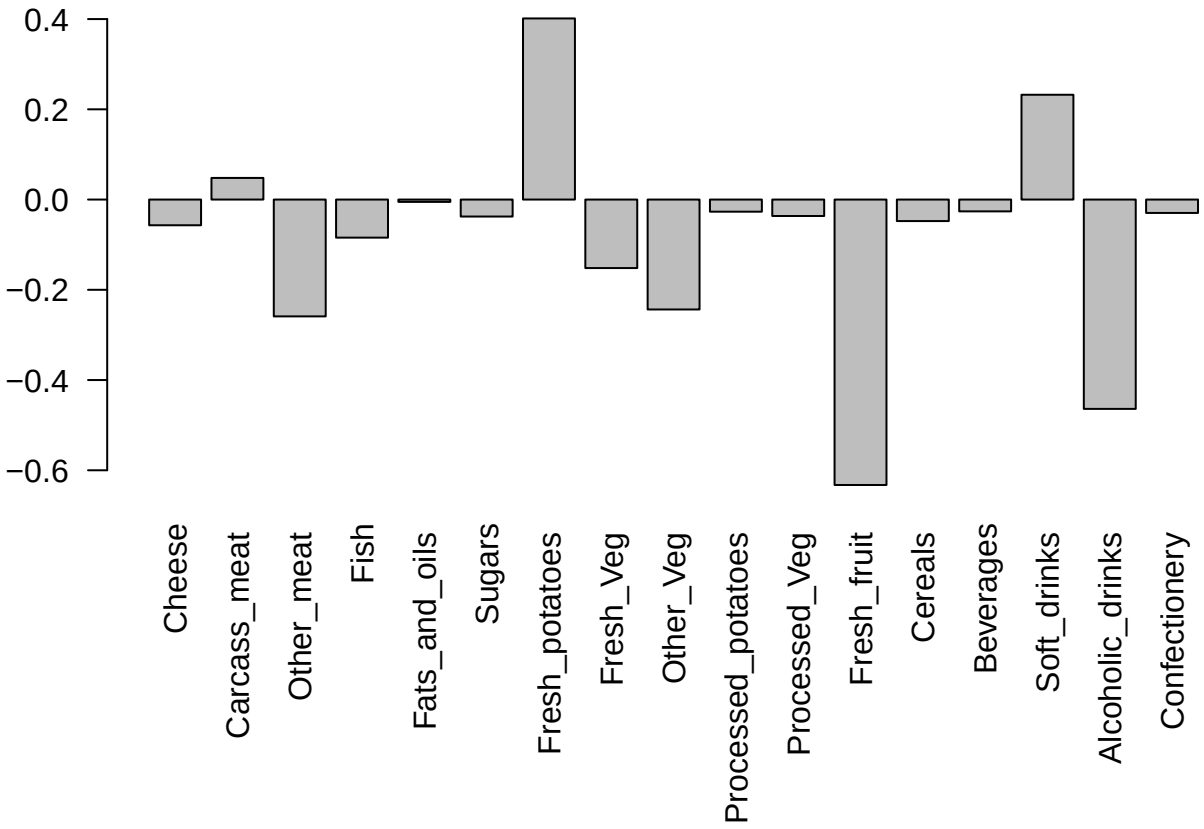
```
plot(pca$x[,1], pca$x[,2], xlab = "PC1", ylab = "PC2") + text(pca$x[,1], pca$x[,2], labels = colnames(x
```




```
## integer(0)
```

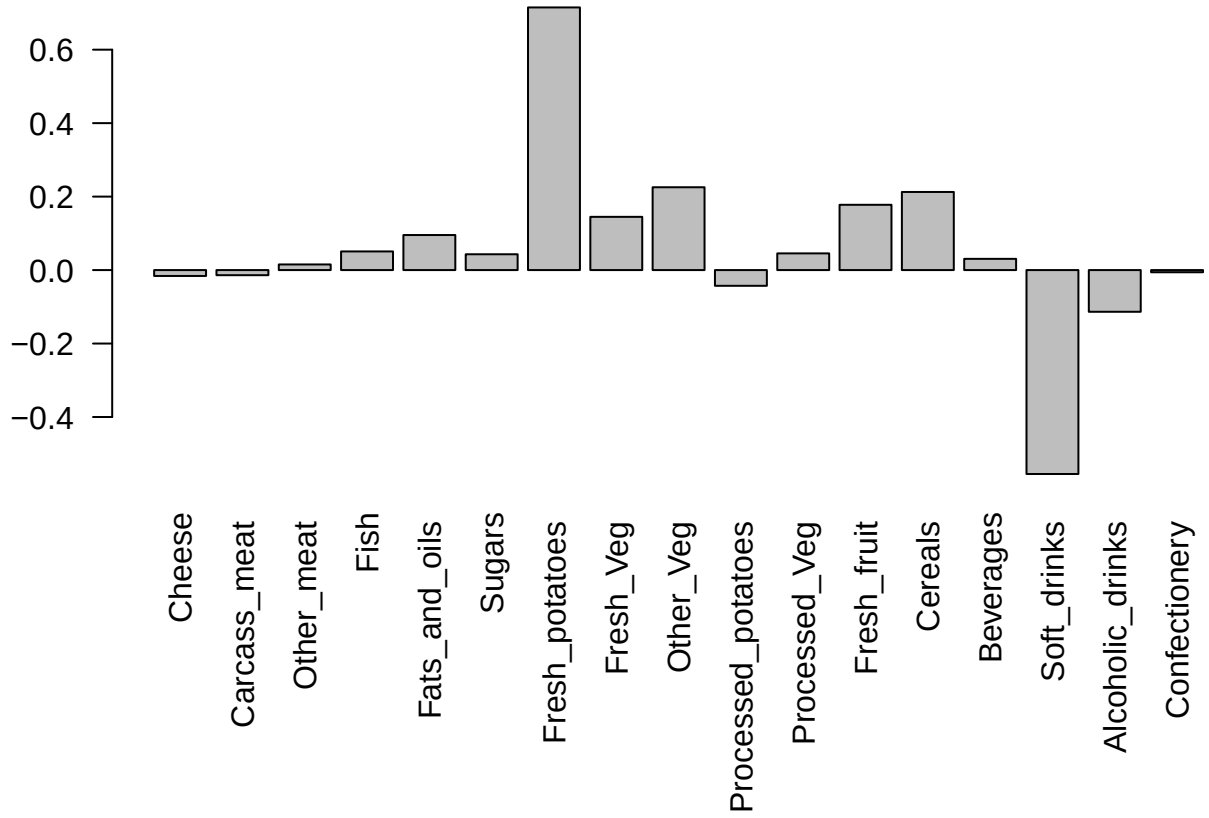
Digging Deeper

```
# focusing on PC1 since it accounts for >90% of all variance in the dataset
par(mar=c(10, 3, 0.35, 0))
barplot( pca$rotation[,1], las=2 )
```



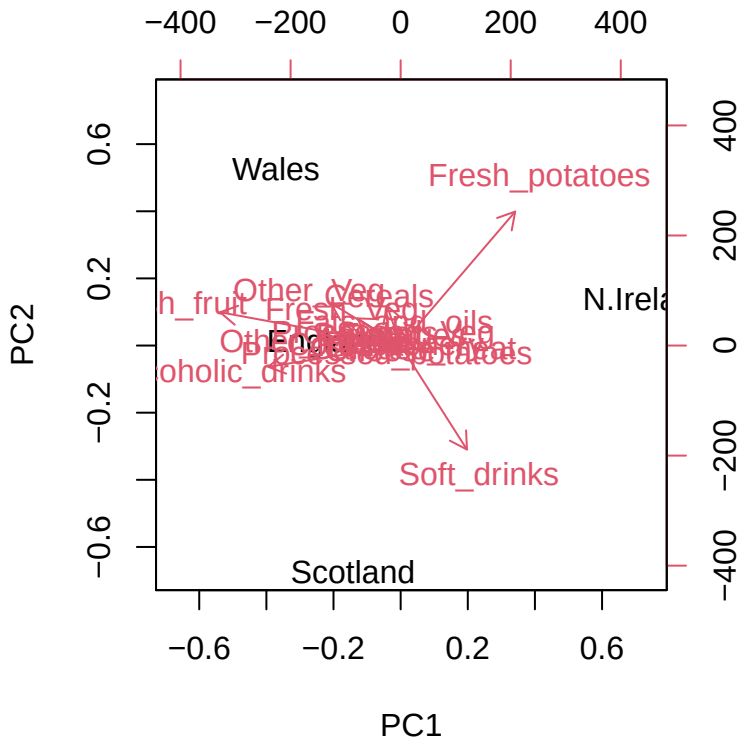
Now for PC2

```
# focusing on PC1 since it accounts for >90% of all variance in the dataset
par(mar=c(10, 3, 0.35, 0))
barplot( pca$rotation[,2], las=2 )
```



Biplots are another way to view this data. The arrows from the center show the amount of variance that single dimension is responsible for in each principal component.

```
biplot(pca)
```



#PCA of RNA-seq data

Import the expression 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
```

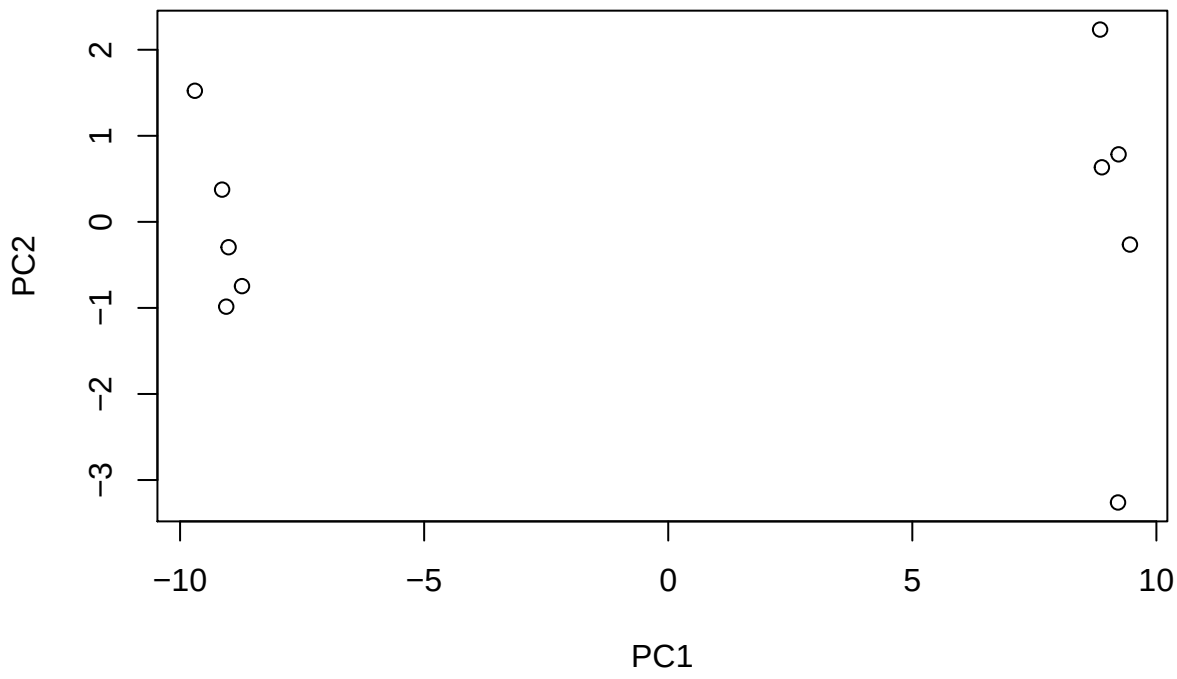
There are 100 genes and 10 samples in this data set (genes are rows and samples are columns).

Perform the PCA analysis and do a simple plot.

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

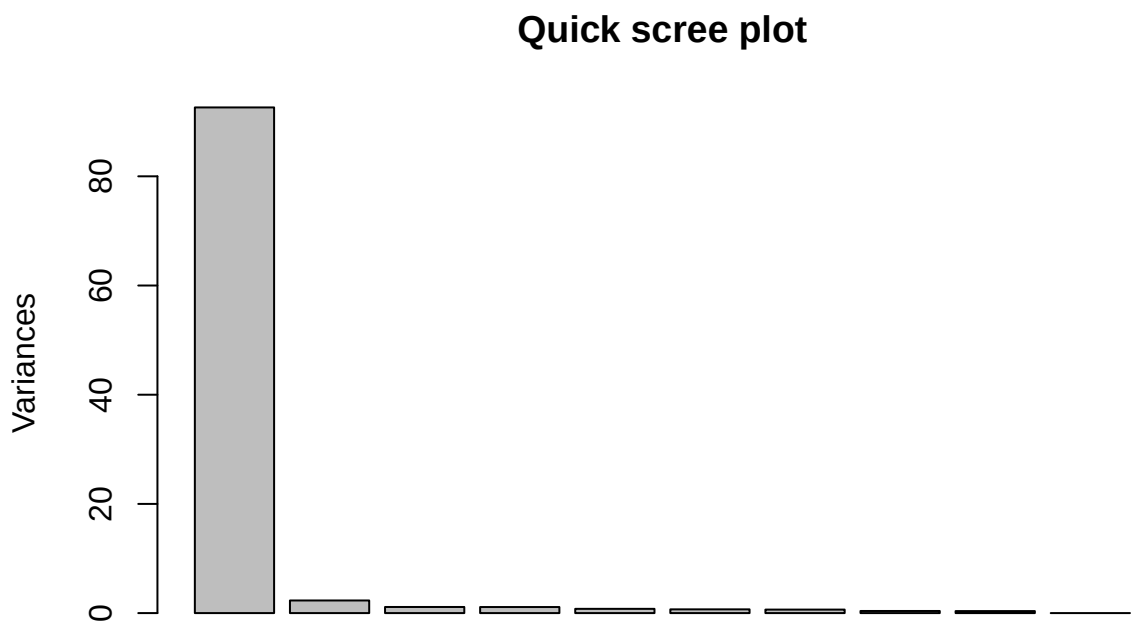
```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     9.6237 1.5198 1.05787 1.05203 0.88062 0.82545 0.80111
## Proportion of Variance 0.9262 0.0231 0.01119 0.01107 0.00775 0.00681 0.00642
## Cumulative Proportion  0.9262 0.9493 0.96045 0.97152 0.97928 0.98609 0.99251
##                           PC8     PC9      PC10
## Standard deviation     0.62065 0.60342 3.348e-15
## Proportion of Variance 0.00385 0.00364 0.000e+00
## Cumulative Proportion  0.99636 1.00000 1.000e+00
```

```
#simple plot
plot(pca$x[,1], pca$x[,2], xlab = "PC1", ylab = "PC2")
```



About 93% of the variance can be shown in the PC1 dimension. Plot the Scree-plot to show the elbow point.

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



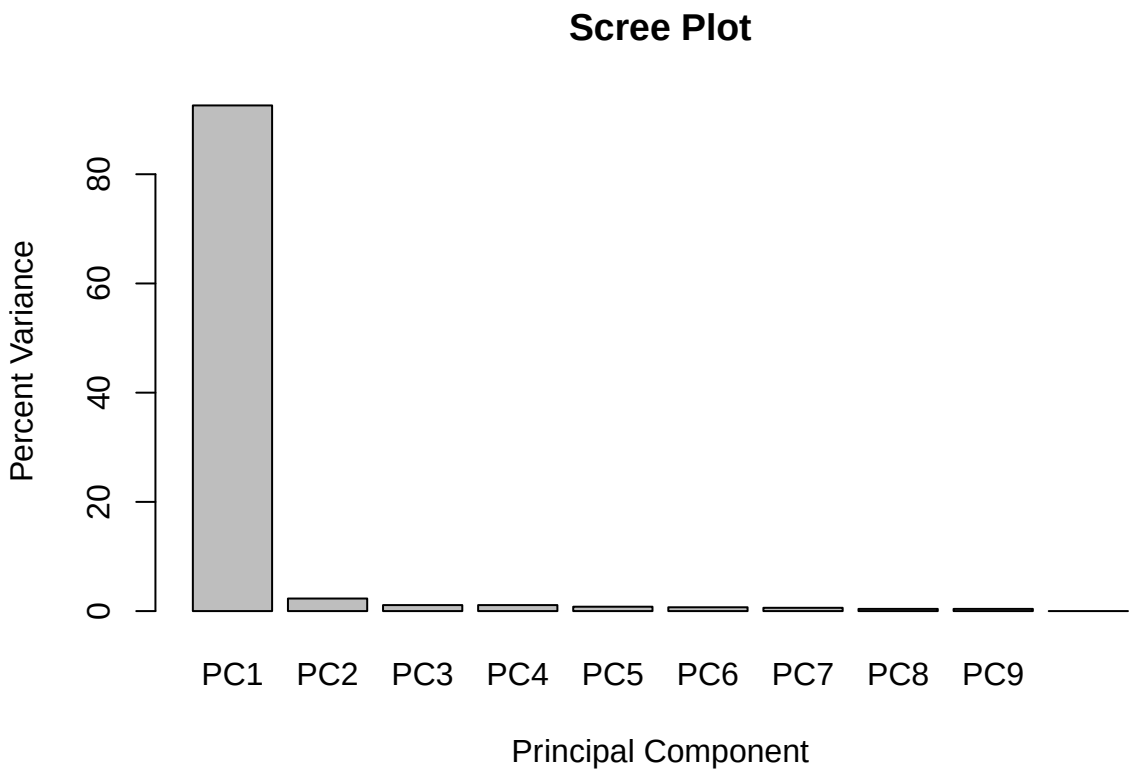
We can also generate our own Scree plot.

```
# Variance caught per principal component
pca.var <- pca$sdev^2

# percent variance is usually easier to look at than just variance
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
```

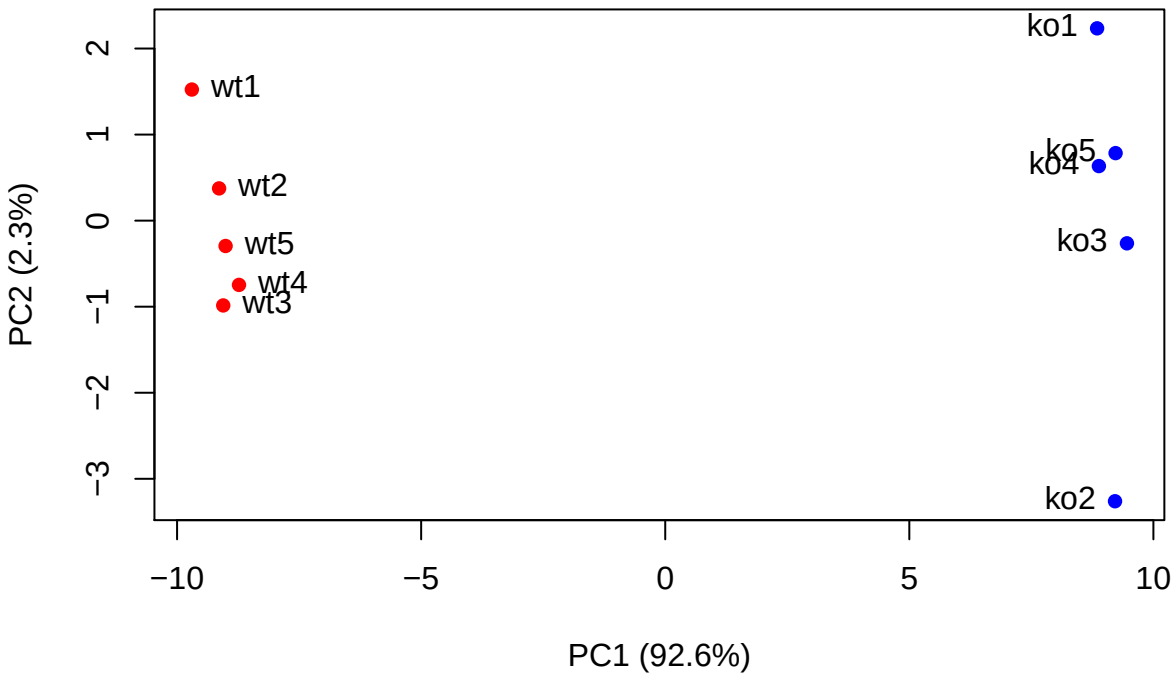
Now use this data to make our own scree plot.

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



Now we can make the PCA plot a bit easier to look at.

```
## 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)))
```

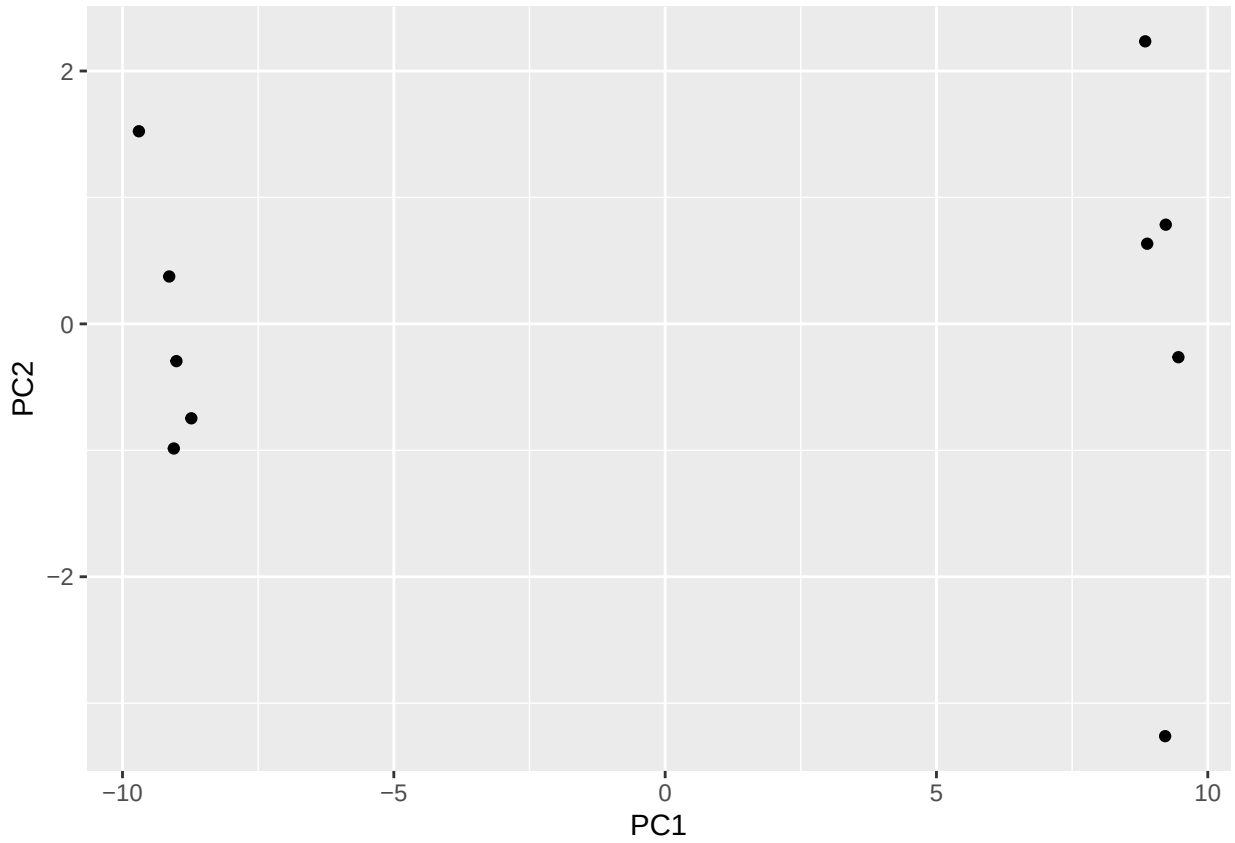


Use ggplot because it's more fun.

```
# Load in the ggplot package
library(ggplot2)
```

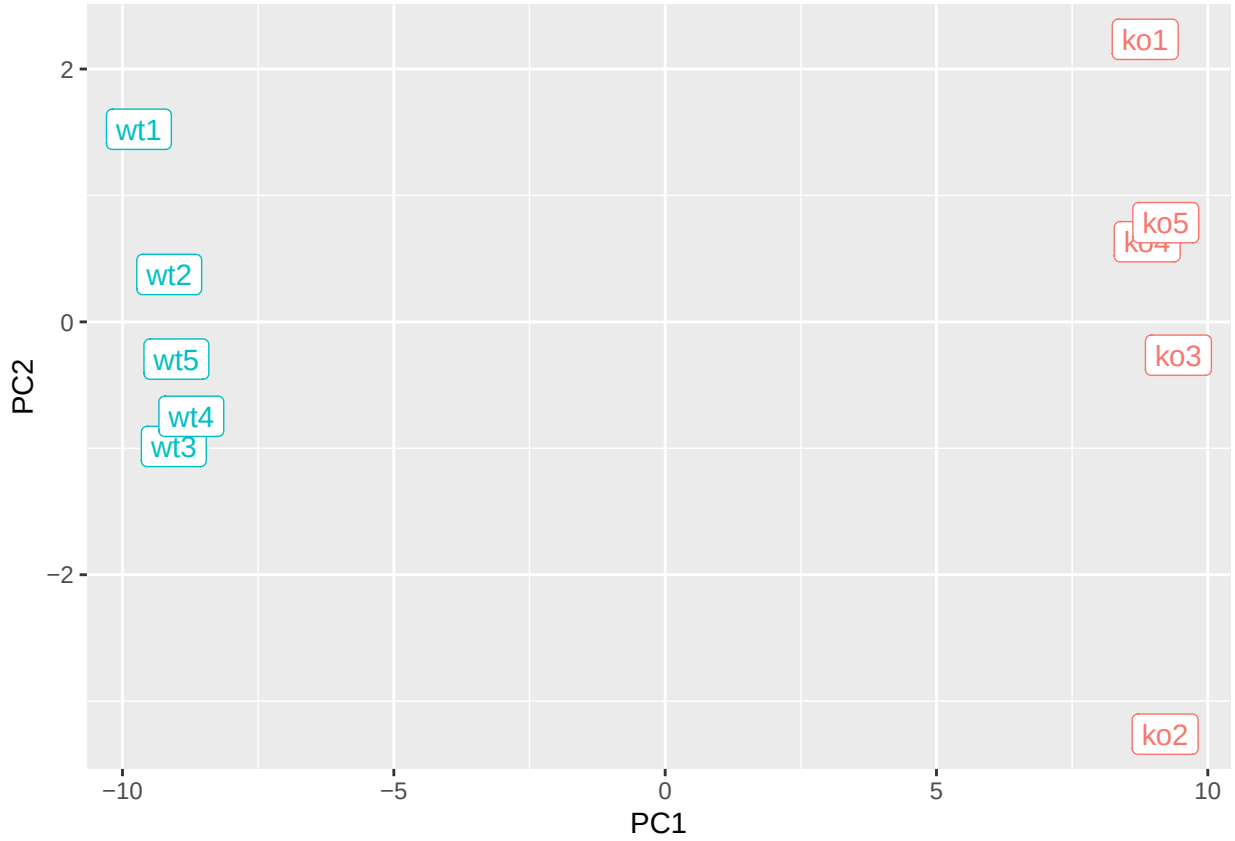
```
## Warning: package 'ggplot2' was built under R version 4.0.3
```

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

# Basic plot once again!
ggplot(df) +
  aes(PC1, PC2) +
  geom_point()
```



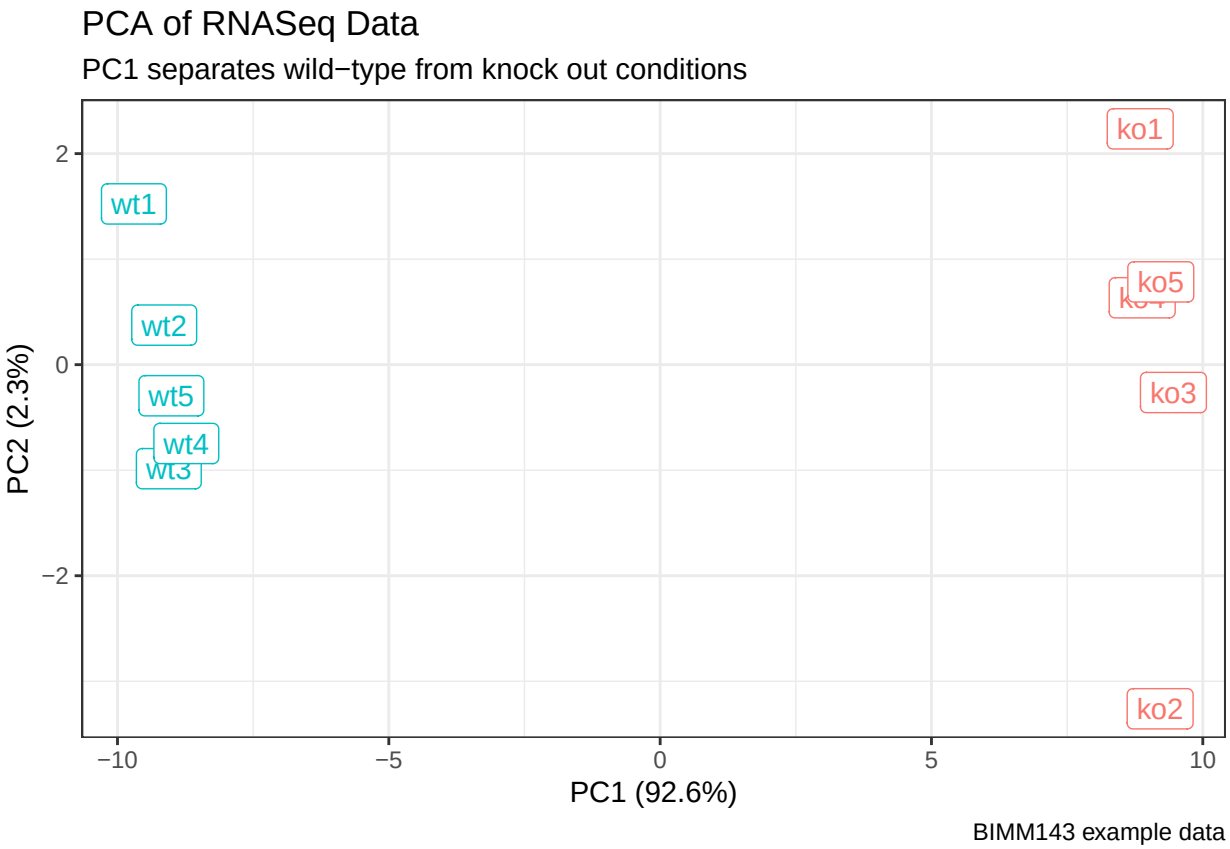
Add in some colors and other aesthetics.

```
# Add a wt and ko condition column
df$samples <- colnames(rna.data)
df$condition <- substr(colnames(rna.data), 1, 2)

#Plot with labels and colors
p <- ggplot(df) +
  aes(PC1, PC2, label = samples, col = condition) +
  geom_label(show.legend = FALSE)
p
```



Add some more labels

```
p + labs(title = "PCA of RNASeq Data",
         subtitle = "PC1 separates wild-type from knock out conditions",
         x = paste0("PC1 (", pca.var.per[1], "%)"),
         y = paste0("PC2 (", pca.var.per[2], "%)"),
         caption = "BIMM143 example data") +
  theme_bw()
```


PCA of RNASeq Data
PC1 separates wild−type from knock out conditions
ko1
wt1
ko5
wt2
ko3
wt5
wt4
wt3
ko2
2
0
−2
PC2 (2.3%)
−10
−5
0
5
10
PC1 (92.6%)
BIMM143 example data