Machine Learning 1

Kaito Tanaka

10/21/2021

Principal Component Analysis (PCA)

PCA of UK food data

Read data from website and try a few visualizations

```
url <- "https://tinyurl.com/UK-foods"
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
```

> Q1. How many rows and columns are in your new data frame named x? What R functions could you use to answer this questions?

Use dim(x) to find the number of rows and colums in the new data

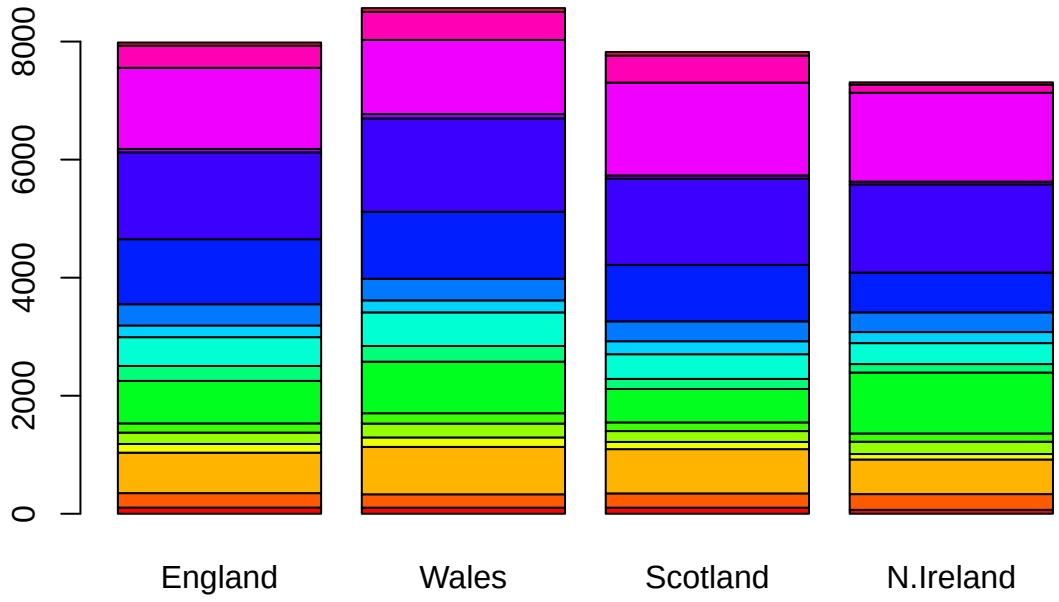
```
dim(x)
```

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

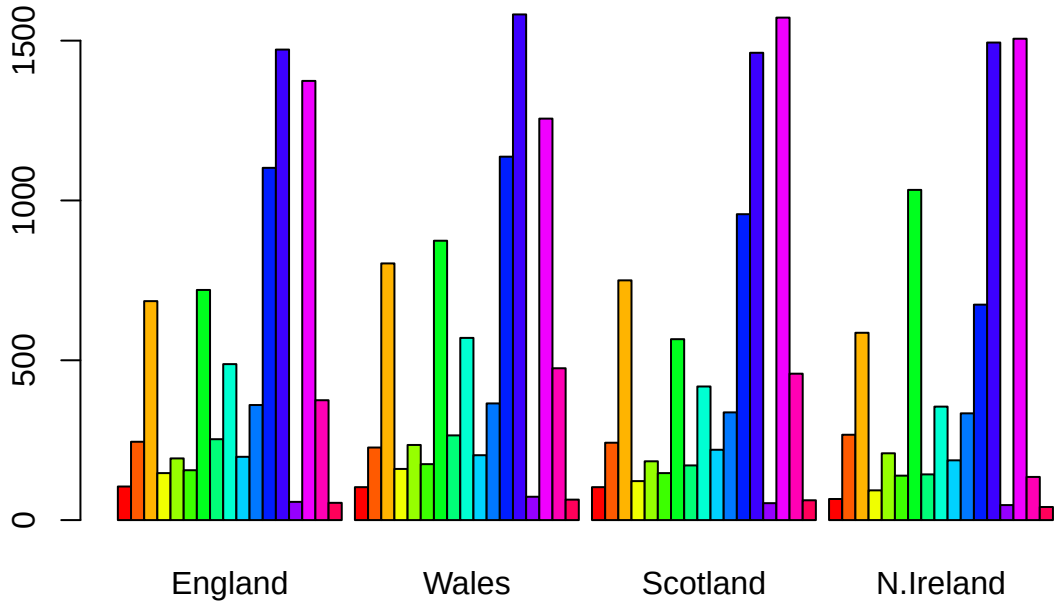
> Q2. Which approach to solving the 'row-names problem' mentioned above do you prefer and why? Is one approach more robust than another under certain circumstances?

I prefer using the first column as the argument of the table. Using "x <- x[,-1]" multiple times can cause you to lose arguments that you shouldn't be using.

```
cols<-rainbow(nrow(x))
barplot(as.matrix(x), col=cols)
```




```
barplot(as.matrix(x), beside=TRUE, col=cols)
```

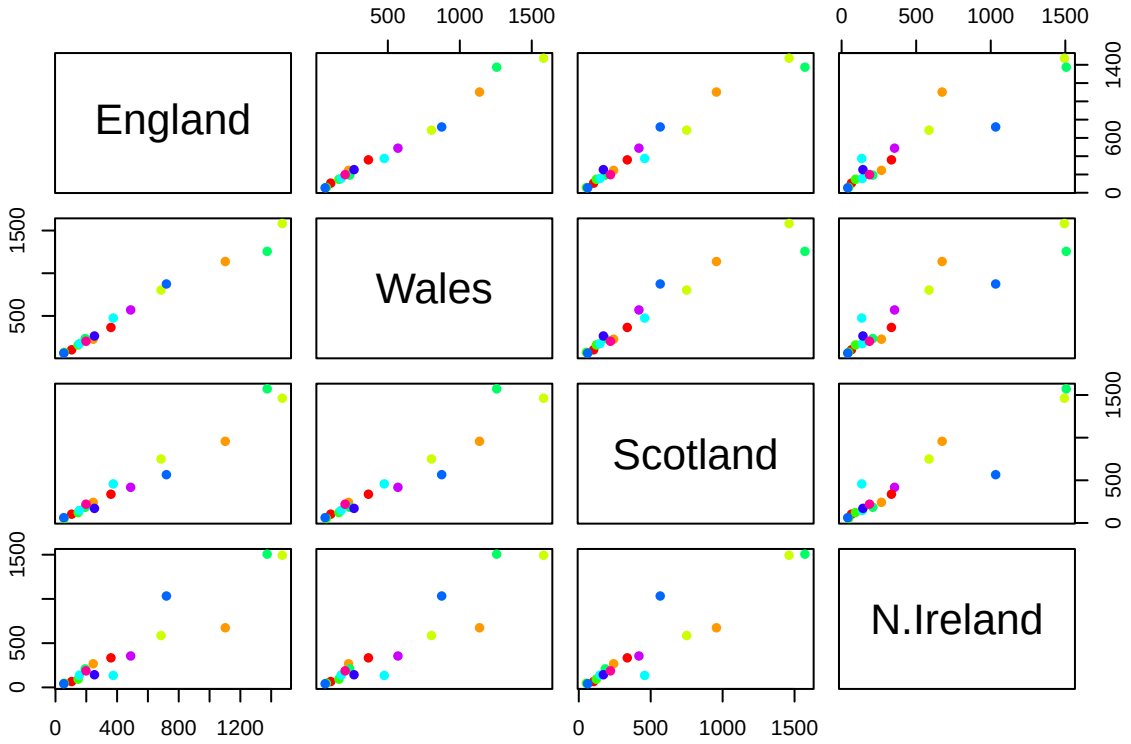


>Q3: Changing what optional argument in the above barplot() function results in the following plot?

Changing the argument of "beside". Setting it to beside=TRUE will result in the change to the plot.

Convert the data into a pairwise plot

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



>Q5: Generating all pairwise plots may help somewhat. Can you make sense of the following code and resulting figure? What does it mean if a given point lies on the diagonal for a given plot?

If a given point lies on the diagonal for a given plot, that means that (in our case) the consumption for that country follows the trend between countries. The closer a point lies on the diagonal, the similar the amount of the variables they have for the x and y parts.

> Q6. What is the main differences between N. Ireland and the other countries of the UK in terms of this data-set?

The main difference between N. Ireland and the other countries is that N. Ireland has a much more varied diet compared to the countries, as evident through the larger spread of points. N. Ireland has much less consumption of fresh fruit, as well as more consumption of alcoholic drinks than the other countries, as evident in the blue and orange dots that are distant from the diagonal.

PCA to the rescue The main base R PCA function is called 'prcomp()' and we will need to give it the transpose of our input data

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

```
attributes(pca)
```

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

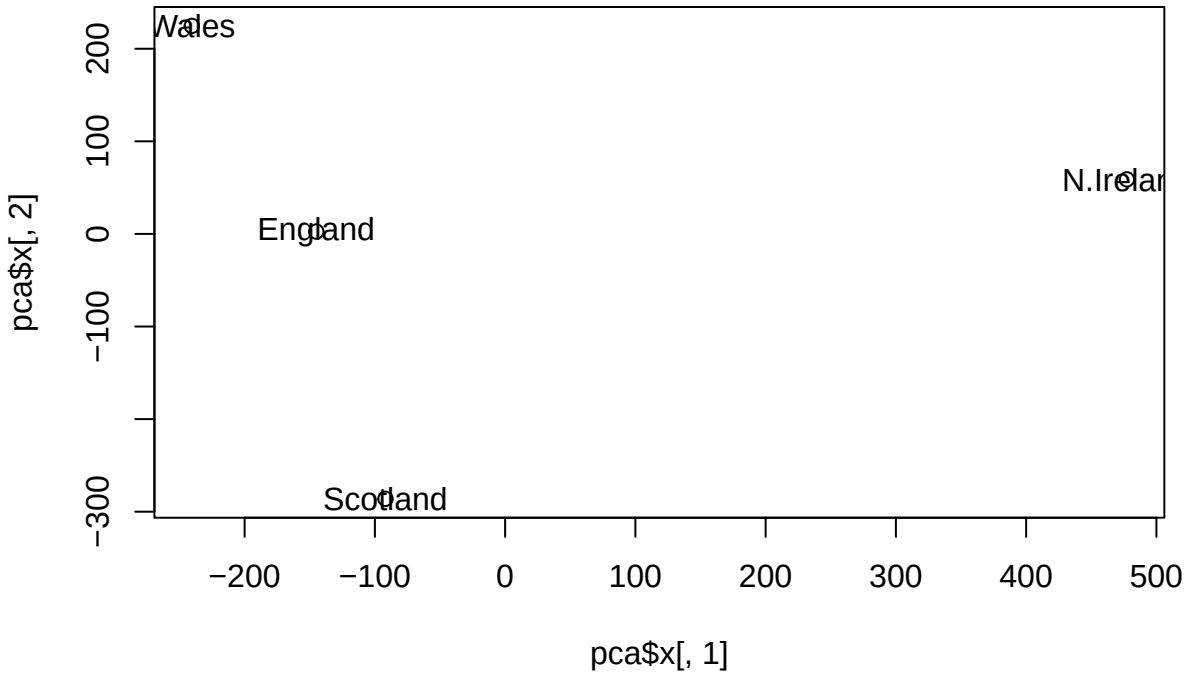
To make our new PCA plot (aka. PCA score plot) we access "pca$x''

```
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
```

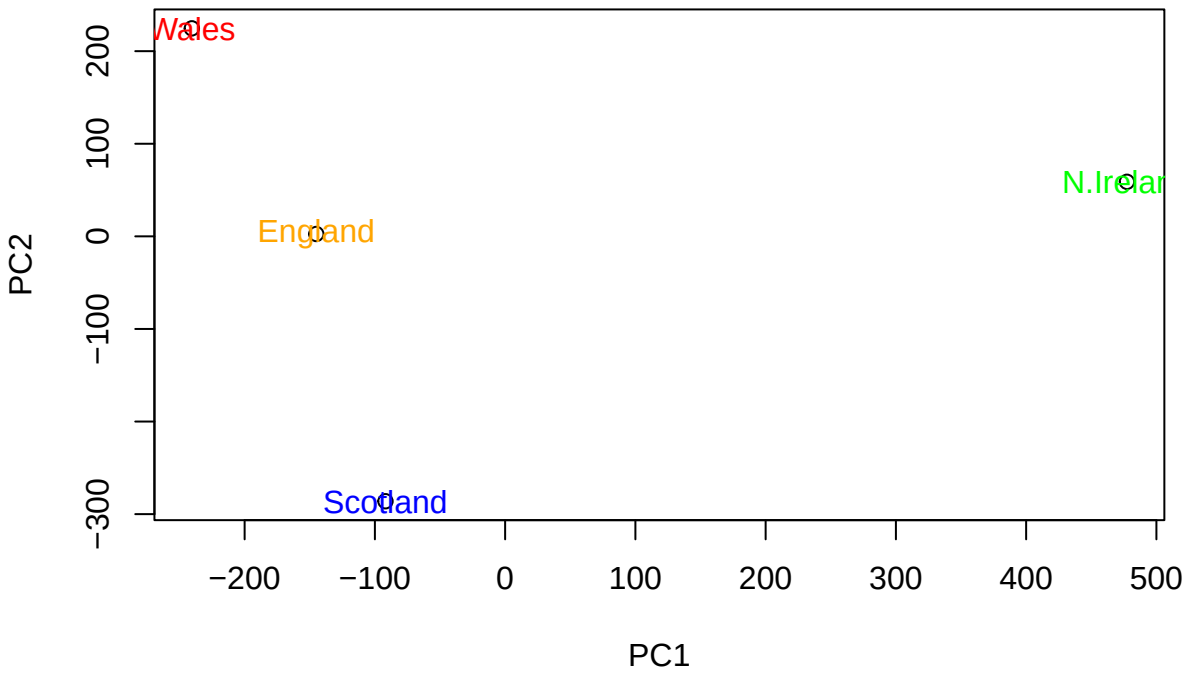
Now plot it, such that each point represents a country in our dataset and add labels as well

```
plot(pca$x[,1],pca$x[,2])
text(pca$x[,1],pca$x[,2], colnames(x))
```



Q7 and Q8: Now color up the plot and add text of PC1 vs PC2

```
country_cols<- c("orange","red","blue","green")
plot(pca$x[,1],pca$x[,2], xlab="PC1", ylab="PC2")
text(pca$x[,1],pca$x[,2], colnames(x), col=country_cols)
```

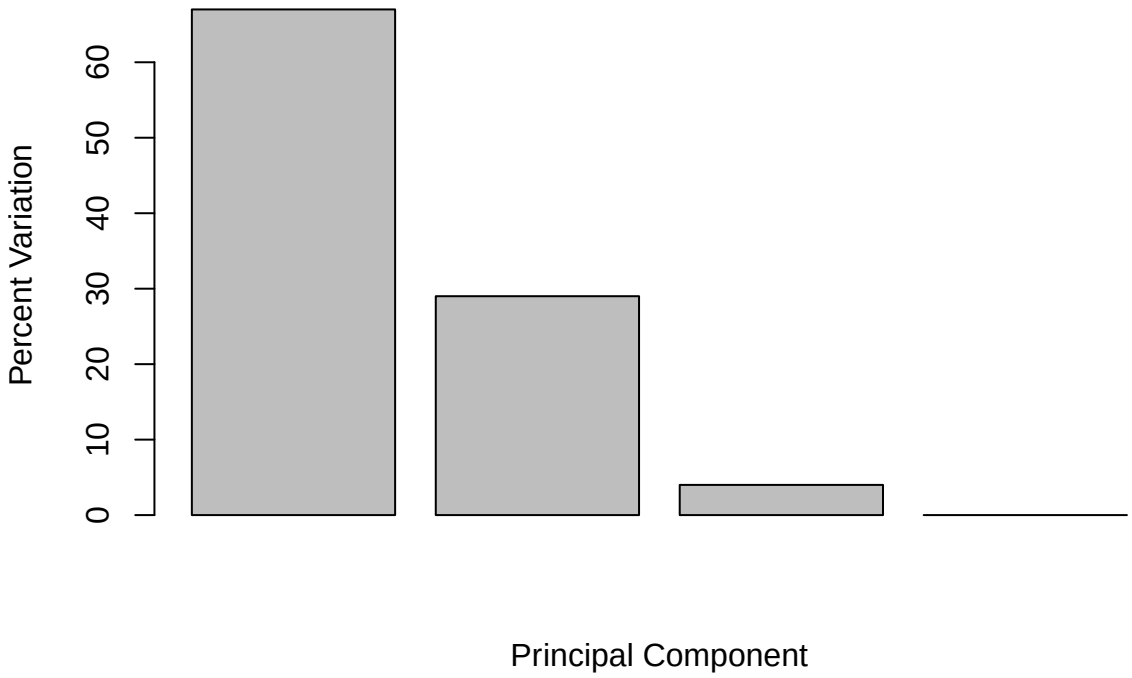


Use the square of pca$sdev (standard deviation) to calculate how much variation in the original data each PC accounts for, then summarize in a barplot

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

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

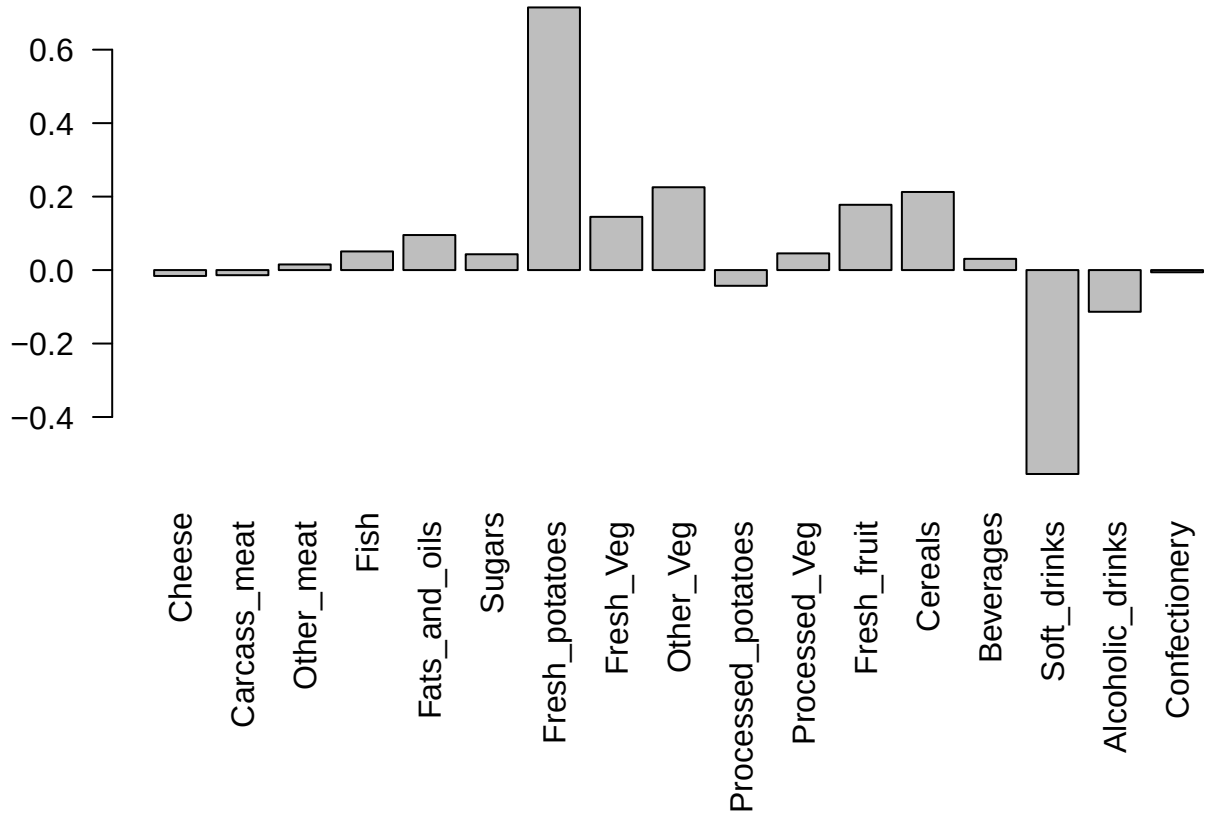
```
barplot(v, xlab="Principal Component", ylab="Percent Variation")
```



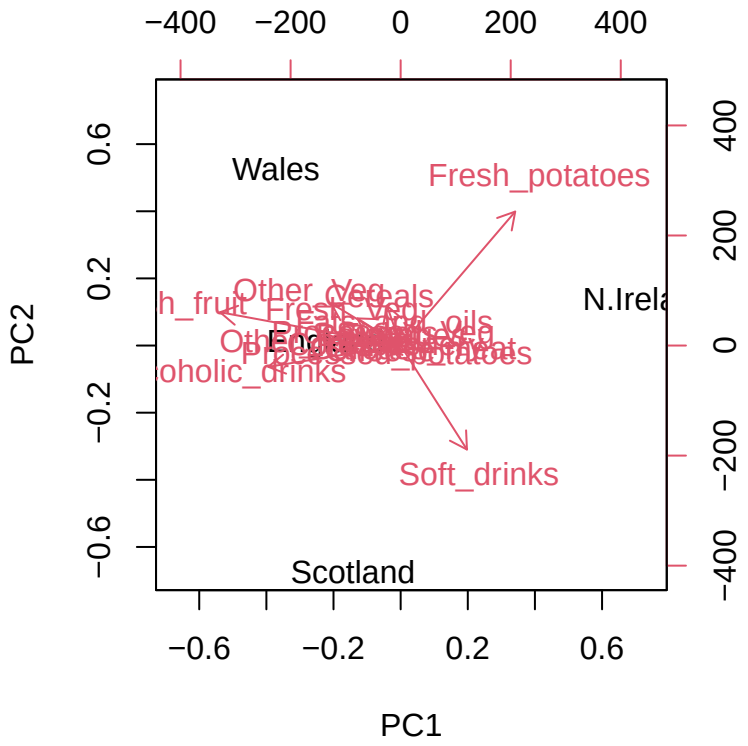
>Q9: Generate a similar 'loadings plot' for PC2. What two food groups feature prominantely and what does PC2 maninly tell us about?

Fresh potatoes and soft drinks are featured prominently. PC2 tells us that they eat more fresh potatoes, but consume less soft drinks, and that those variables greatly vary in N. Ireland compared to other UK countries.

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


0.6
0.4
0.2
0.0
−0.2
−0.4
Cheese
Carcass_meat
Other_meat
Fish
Fats_and_oils
Sugars
Fresh_potatoes
Fresh_Veg
Other_Veg
Processed_potatoes
Processed_Veg
Fresh_fruit
Cereals
Beverages
Soft_drinks
Alcoholic_drinks
Confectionery

```
biplot(pca)
```



##PCA of RNA-Seq data

Read in data from website

```
url2 <- "https://tinyurl.com/expression-CSV"
rna.data <- read.csv(url2, row.names=1)
head(rna.data)
```

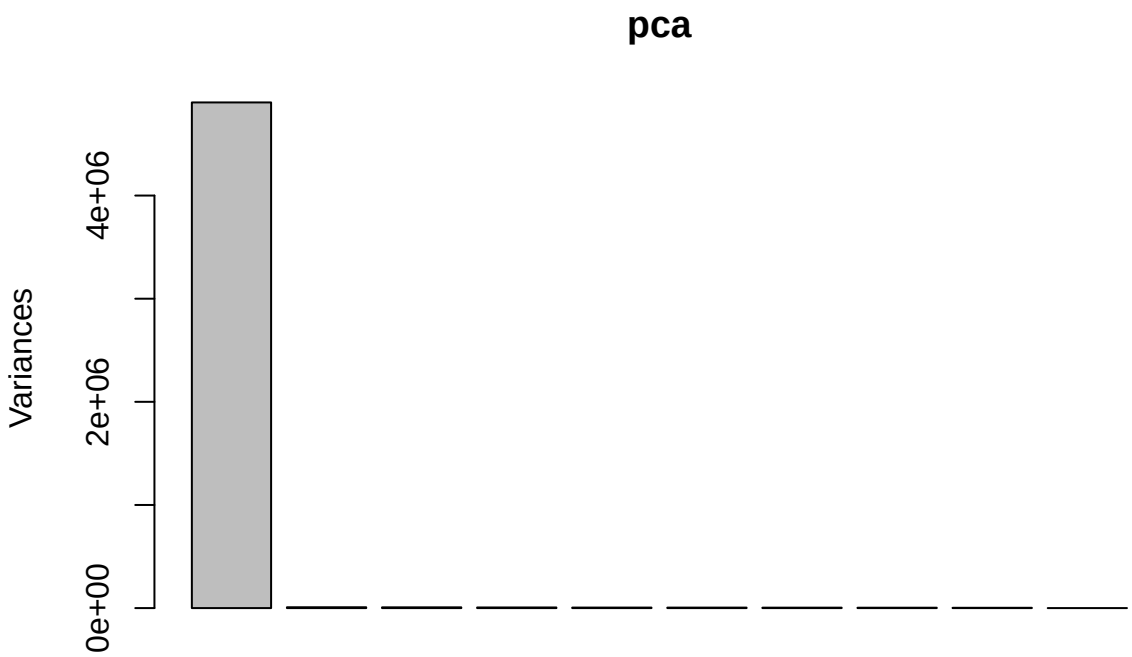
```
##        wt1 wt2  wt3  wt4 wt5 ko1 ko2 ko3 ko4 ko5
## gene1  439 458  408  429 420  90  88  86  90  93
## gene2  219 200  204  210 187 427 423 434 433 426
## gene3 1006 989 1030 1017 973 252 237 238 226 210
## gene4  783 792  829  856 760 849 856 835 885 894
## gene5  181 249  204  244 225 277 305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

Q10: How many genes and samples are in this data set?

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

```
## [1] 100  10
```

```
pca<-prcomp(t(rna.data))
plot(pca)
```



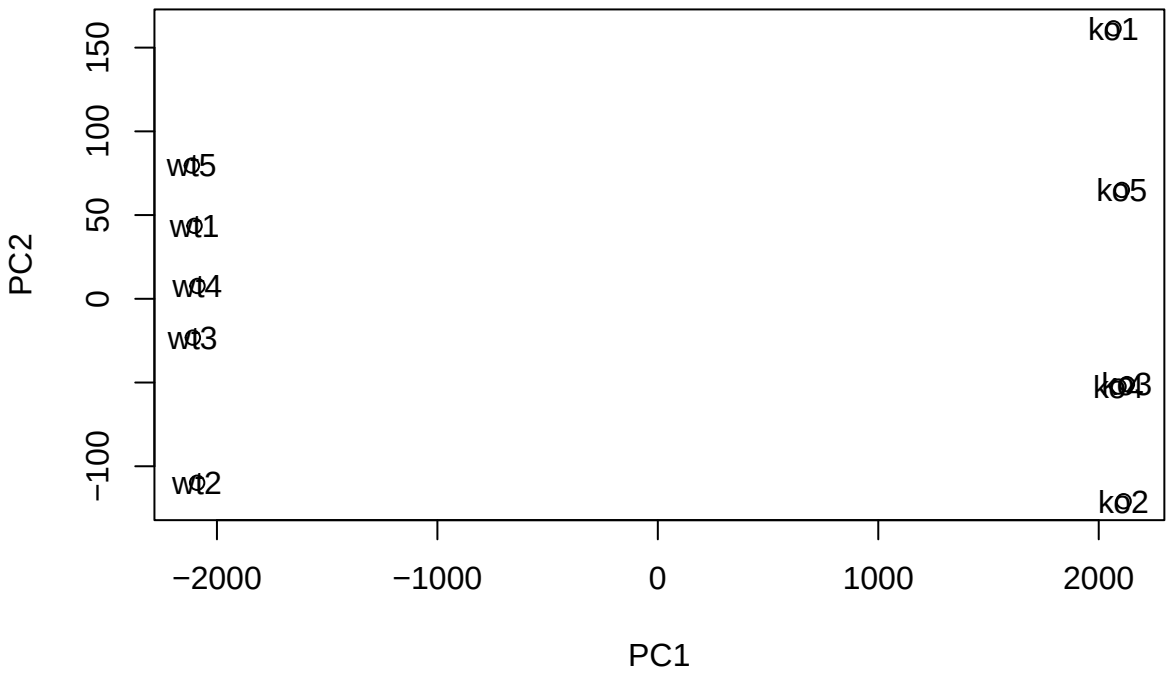
We can also view a summary of how well PCA is doing

```
summary(pca)
```

```
## Importance of components:
##                             PC1     PC2      PC3      PC4      PC5      PC6
## Standard deviation     2214.2633 88.9209 84.33908 77.74094 69.66341 67.78516
## Proportion of Variance    0.9917  0.0016  0.00144  0.00122  0.00098  0.00093
## Cumulative Proportion     0.9917  0.9933  0.99471  0.99593  0.99691  0.99784
##                            PC7      PC8      PC9      PC10
## Standard deviation     65.29428 59.90981 53.20803 3.142e-13
## Proportion of Variance  0.00086  0.00073  0.00057 0.000e+00
## Cumulative Proportion   0.99870  0.99943  1.00000 1.000e+00
```

Do our PCA plot of this RNA-Seq data with texts

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

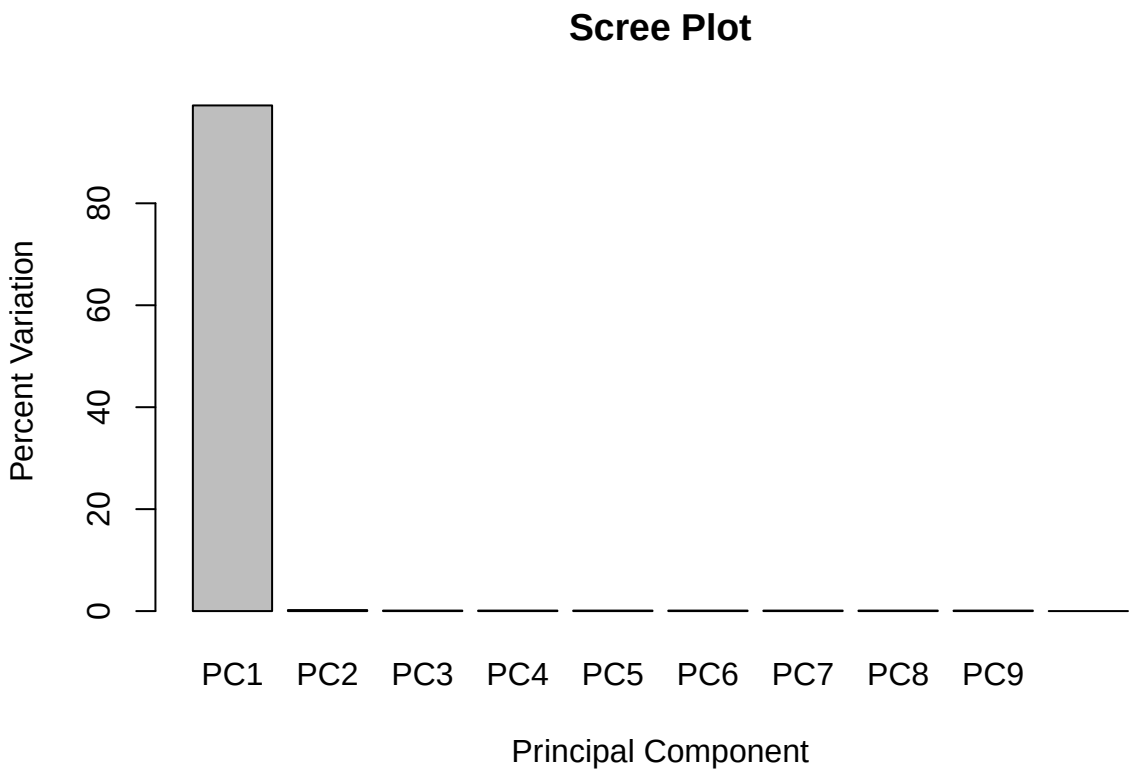


Generate a screen-plot

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

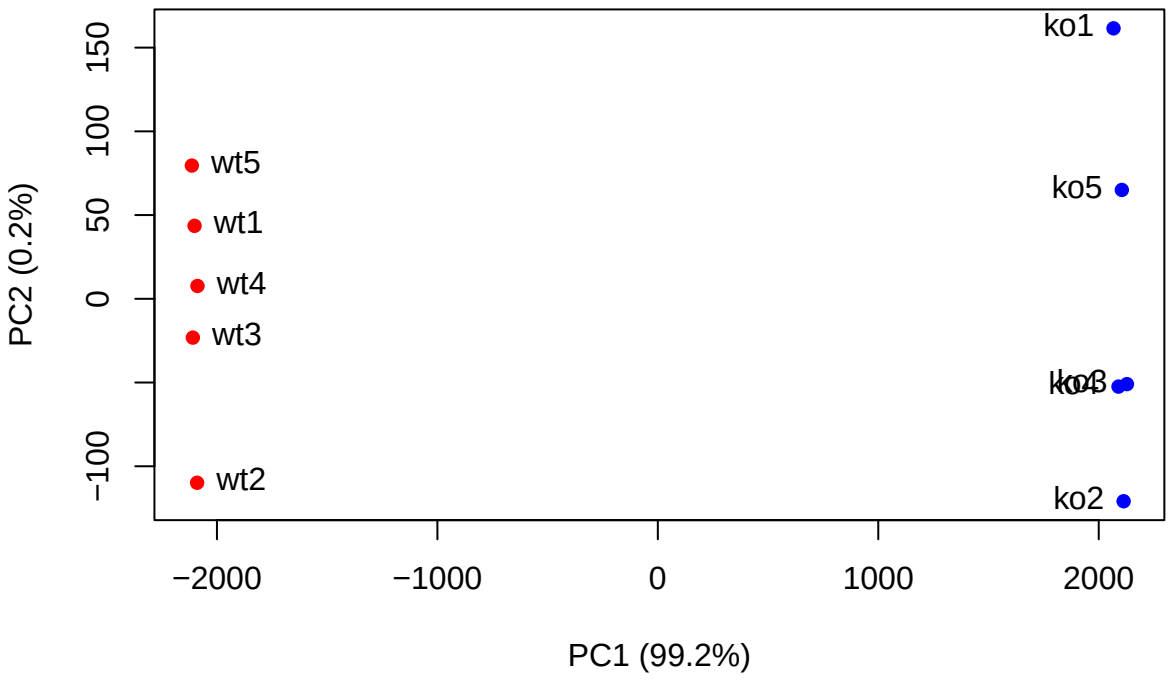
```
##  [1] 99.2  0.2  0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.0
```

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



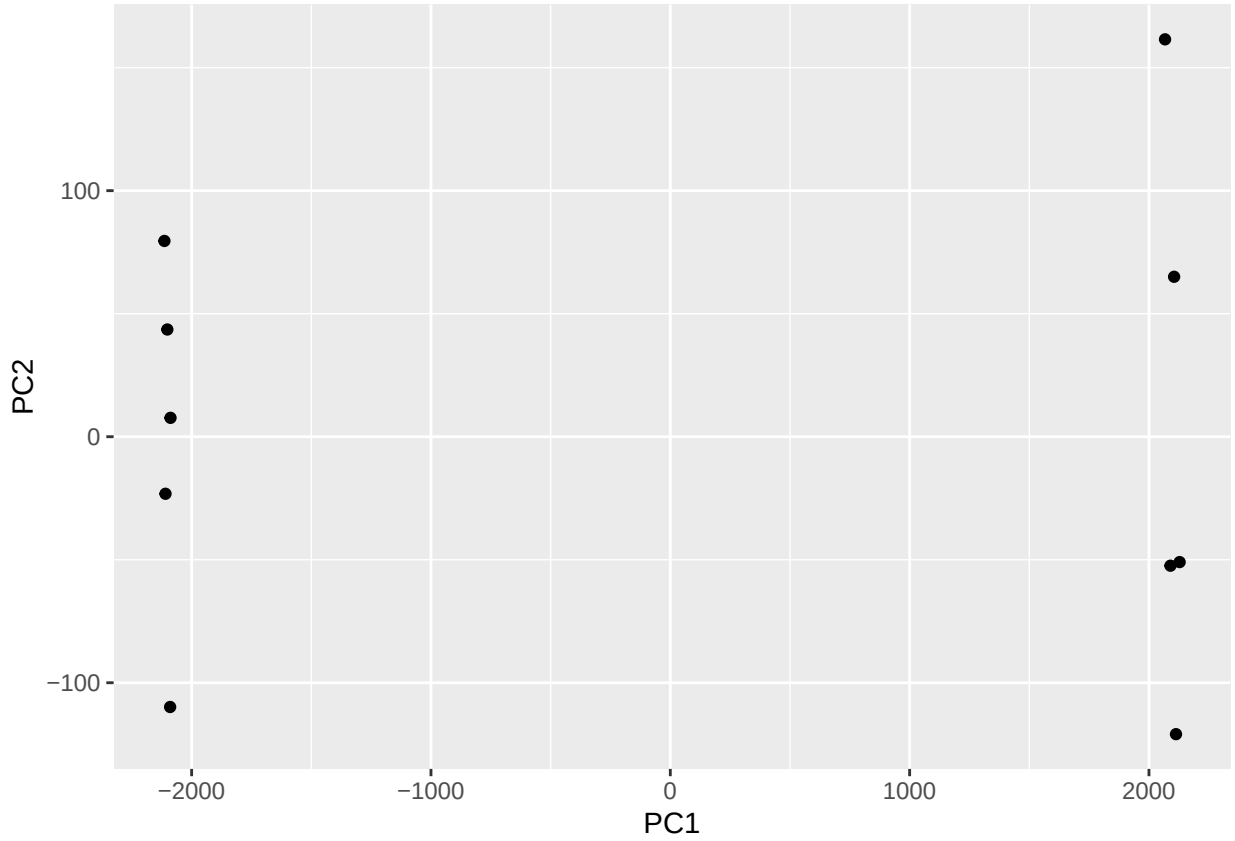
Now make the main PCA plot more attractive and useful

```
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)))
```



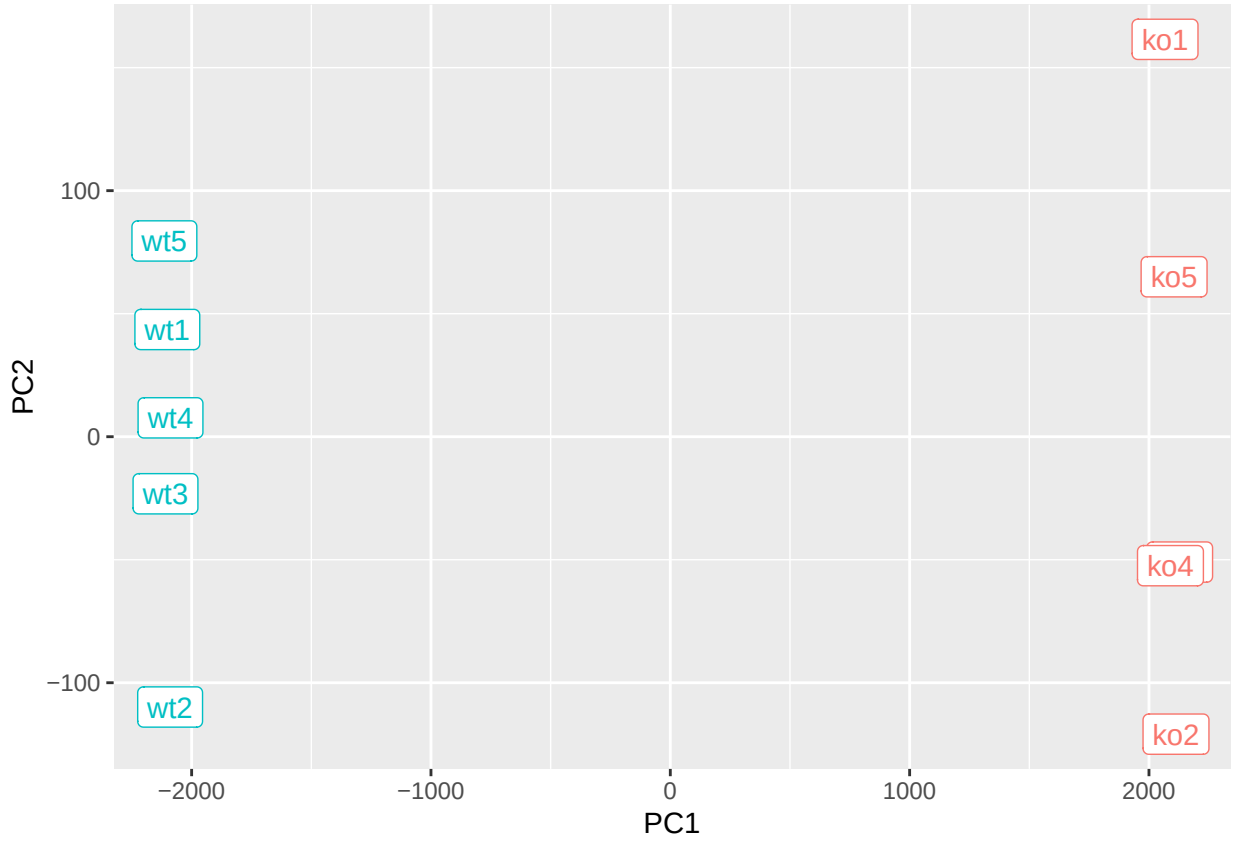
Now use ggplot2 package

```
library(ggplot2)

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

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



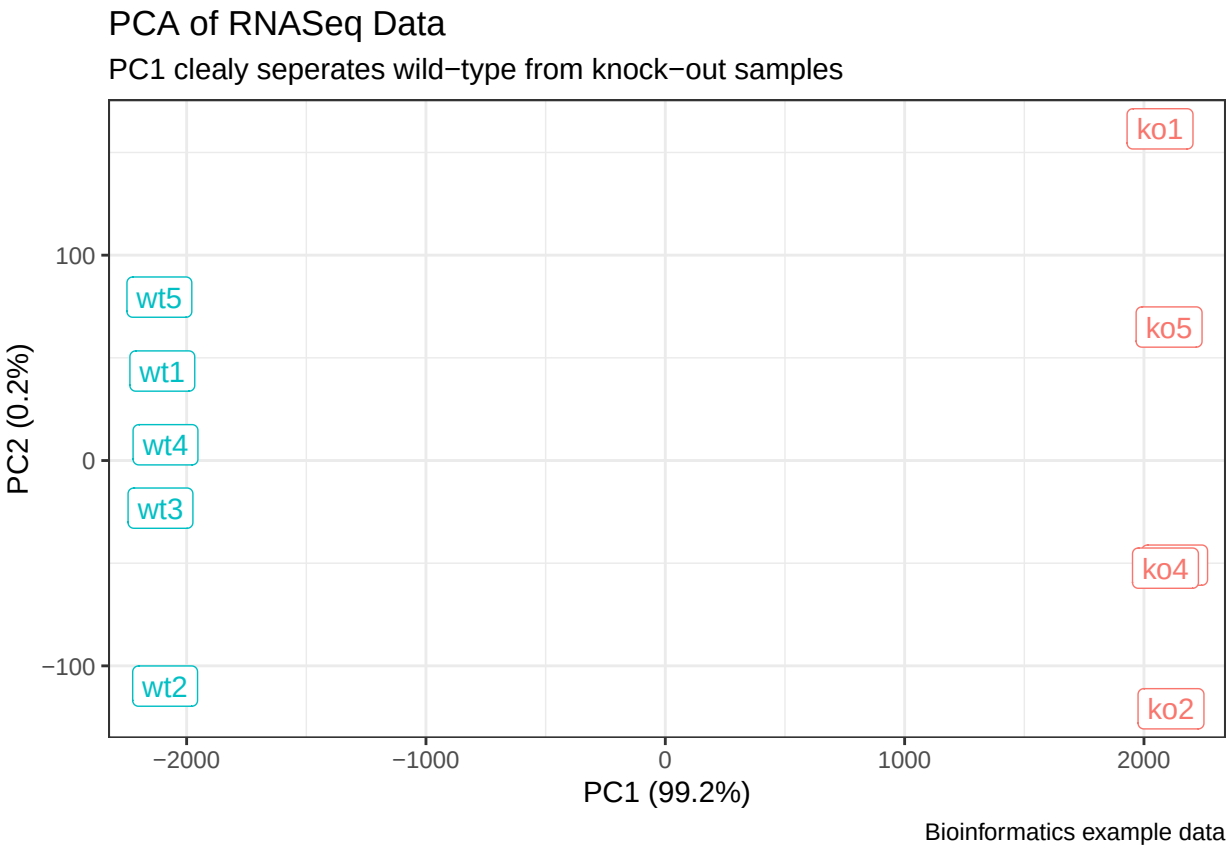
Now add condition specific colors to the ggplot

```
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
```



Finally, polish your ggplot

```
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="Bioinformatics example data") +
     theme_bw()
```


PCA of RNASeq Data
PC1 clealy seperates wild−type from knock−out samples
ko1
wt5
ko5
wt1
wt4
wt3
ko4
wt2
ko2
100
0
−100
PC2 (0.2%)
−2000
−1000
0
1000
2000
PC1 (99.2%)
Bioinformatics example data