class07

AUTHOR
Jo Bautista - A10684919

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



#Q1. How many rows and columns are in your new data frame named x? What R functions could you use

#give dimensions --> number of rows, number of columns
dim(x)
```

```
[1] 17  5
```

```
#preview first six row
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
```

```
#reset first column to be name of rows instead of included as a column
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
```

```
#Q2. Which approach to solving the ‘row-names problem’ mentioned above do you prefer and why? Is

#A2. I prefer the second option (read.csv(url, row.names=1)) because it is more robust in that it
```

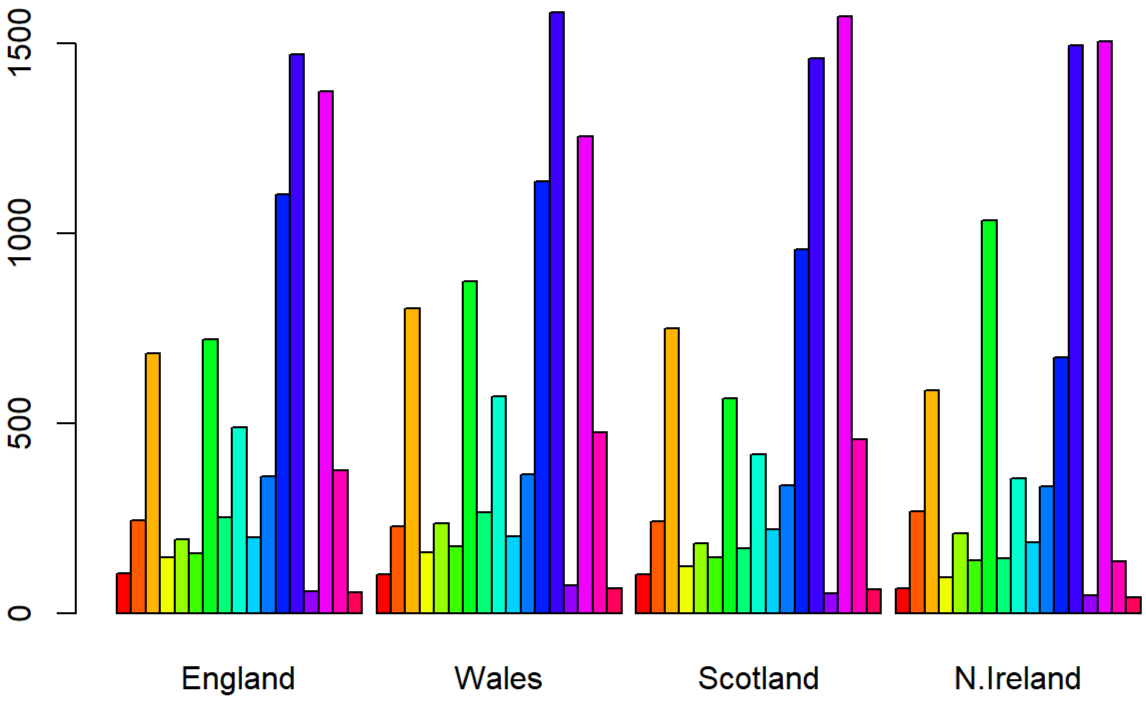
```
#check dimensions again (number of rows, number of columns)
dim(x)
```

```
[1] 17  4
```

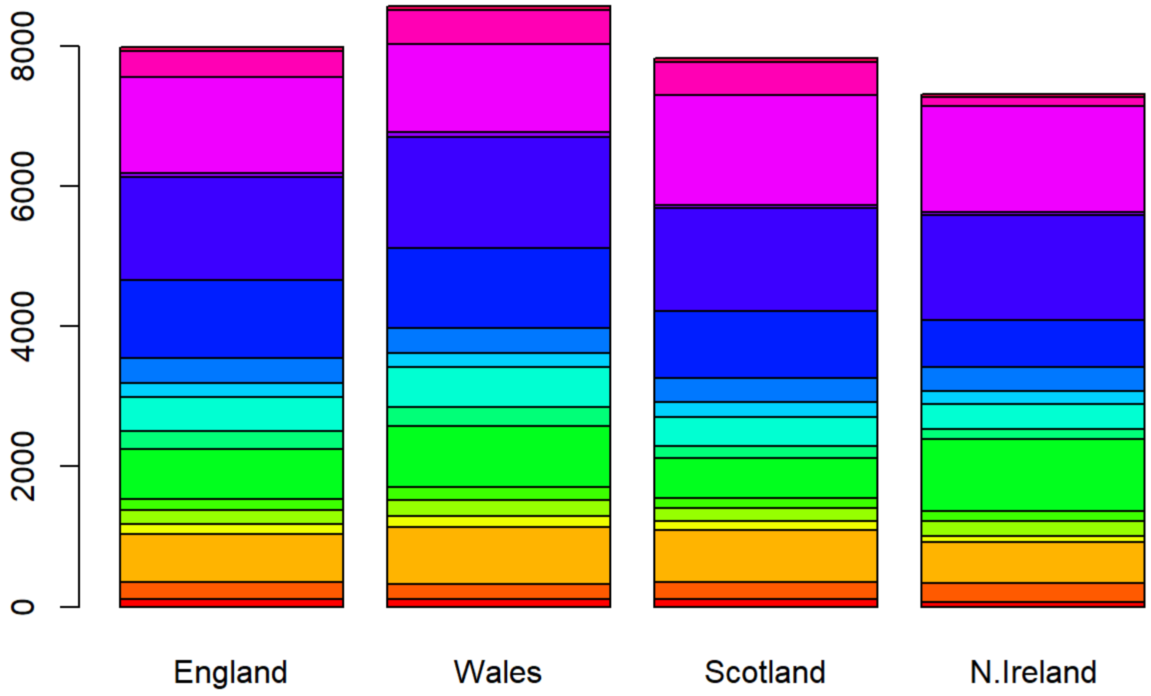
```
#another way of avoiding rownames as first column
x <- read.csv(url, row.names=1)
head(x)
```

```
               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
```

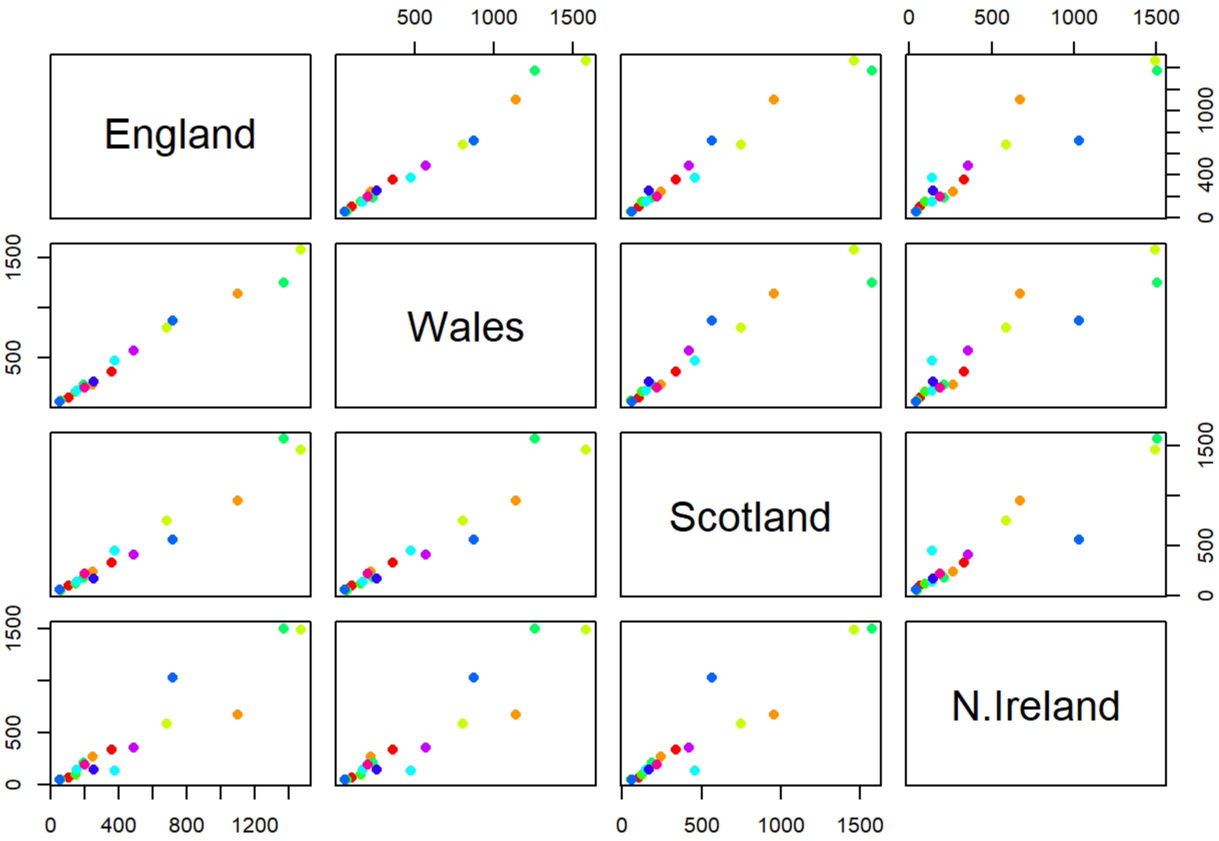
```
#barplot of x with bars displayed side by side
barplot(as.matrix(x), beside=T, col=rainbow(nrow(x)))
```



```
#Q3: Changing what optional argument in the above barplot() function results in the following plo

#A3. Change beside=T to beside=F.

#barplot of x with bars displayed stacked
barplot(as.matrix(x), beside=F, col=rainbow(nrow(x)))
```



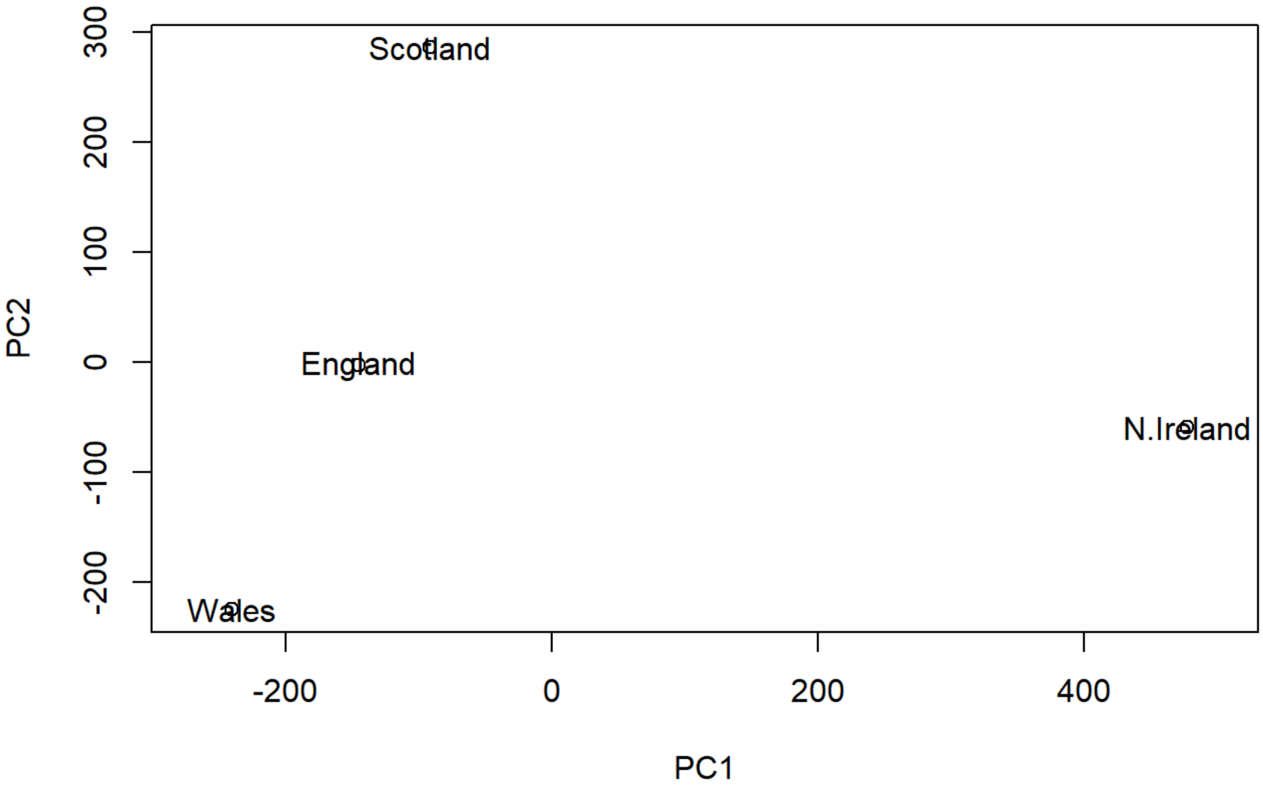
```
#Q5: Generating all pairwise plots may help somewhat. Can you make sense of the following code an

#A5: The diagonal shows the distribution of each variable (like a histogram).

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



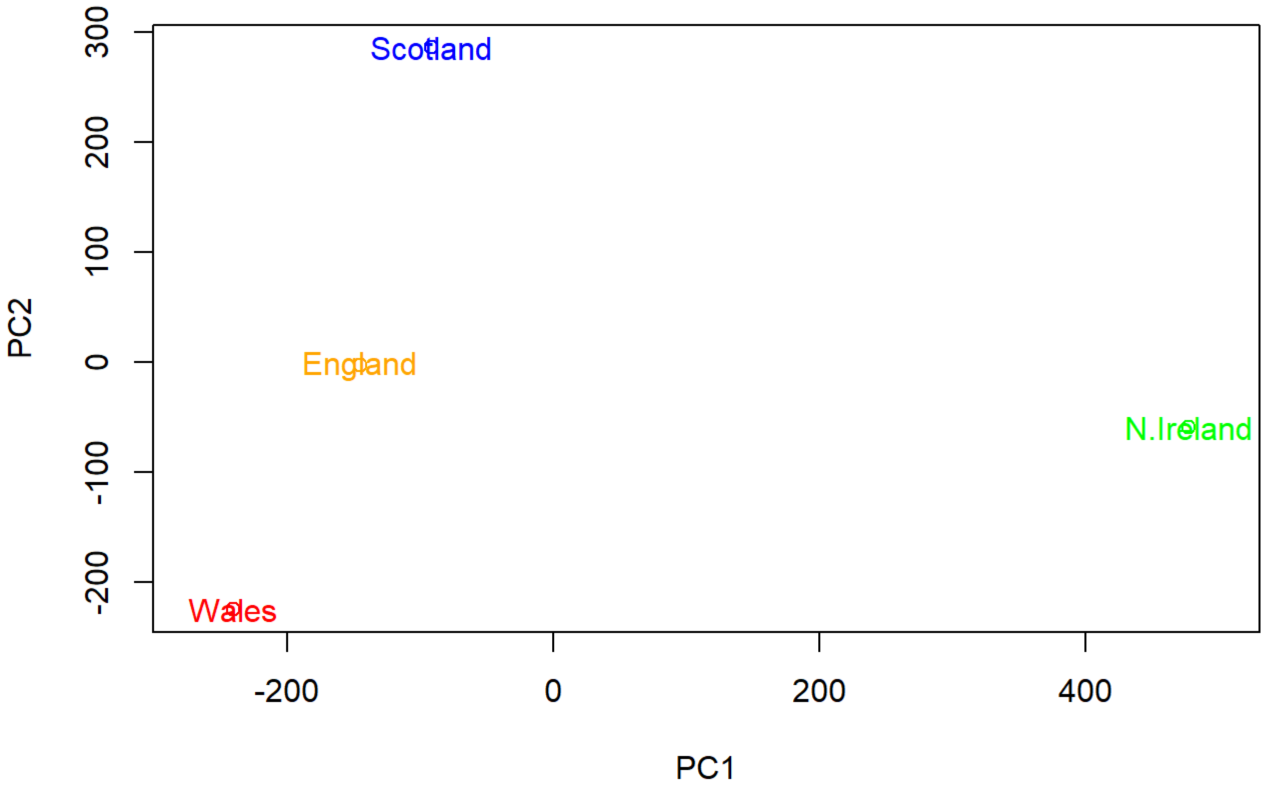
```
# Use the prcomp() PCA function
pca <- prcomp(t(x))
summary(pca)
```

```
Importance of components:
                            PC1      PC2      PC3       PC4
Standard deviation     324.1502 212.7478 73.87622 3.176e-14
Proportion of Variance   0.6744   0.2905  0.03503 0.000e+00
Cumulative Proportion    0.6744   0.9650  1.00000 1.000e+00
```

```
#Q7. Complete the code below to generate a plot of PC1 vs PC2. The second line adds text labels o


# 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))
```



```
#Q8. Customize your plot so that the colors of the country names match the colors in our UK and I

countries <- colnames(x)
colors <- ifelse(countries == "Wales", "red",
                 ifelse(countries == "England", "orange",
                        ifelse(countries == "Scotland", "blue",
                               ifelse(countries == "N. Ireland", "green", "green"))))

plot(pca$x[, 1], pca$x[, 2], xlab="PC1", ylab="PC2", xlim=c(-270, 500), col=colors)
text(pca$x[, 1], pca$x[, 2], countries, col=colors)
```



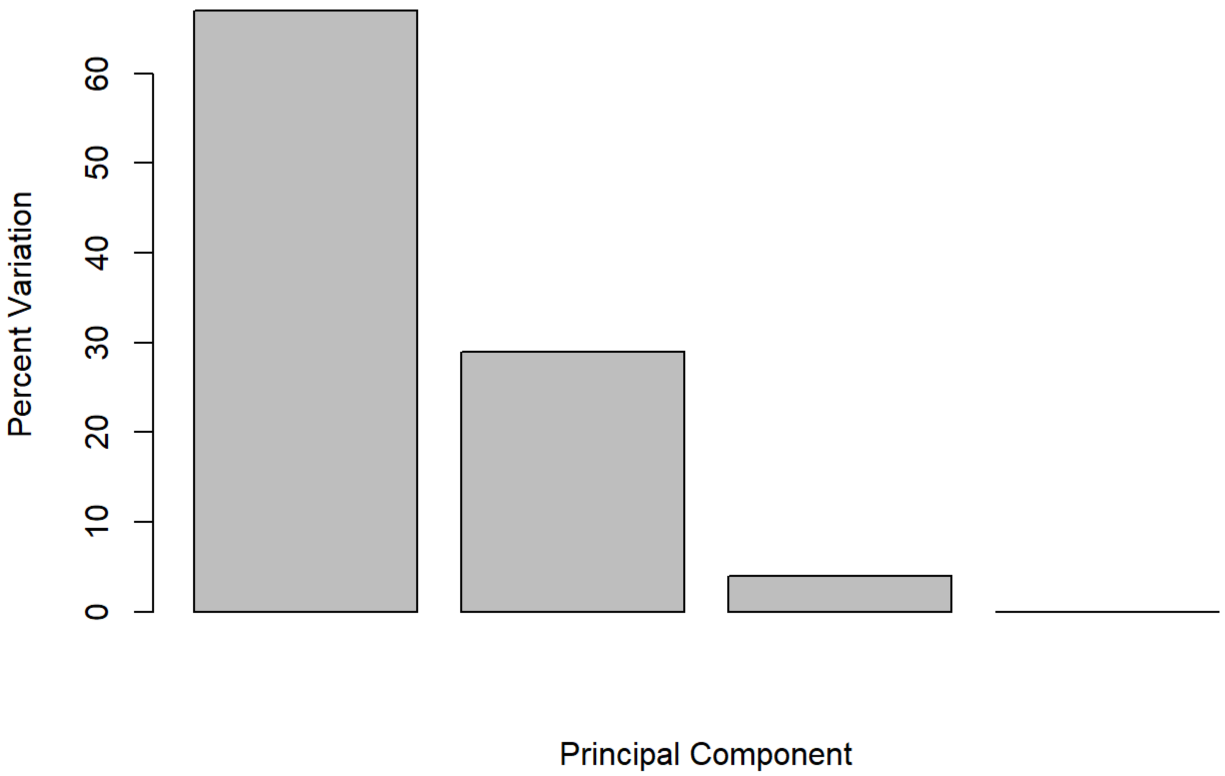
```
v <- round( pca$sdev^2/sum(pca$sdev^2) * 100 )
v
```

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

```
## second row
z <- summary(pca)
z$importance
```

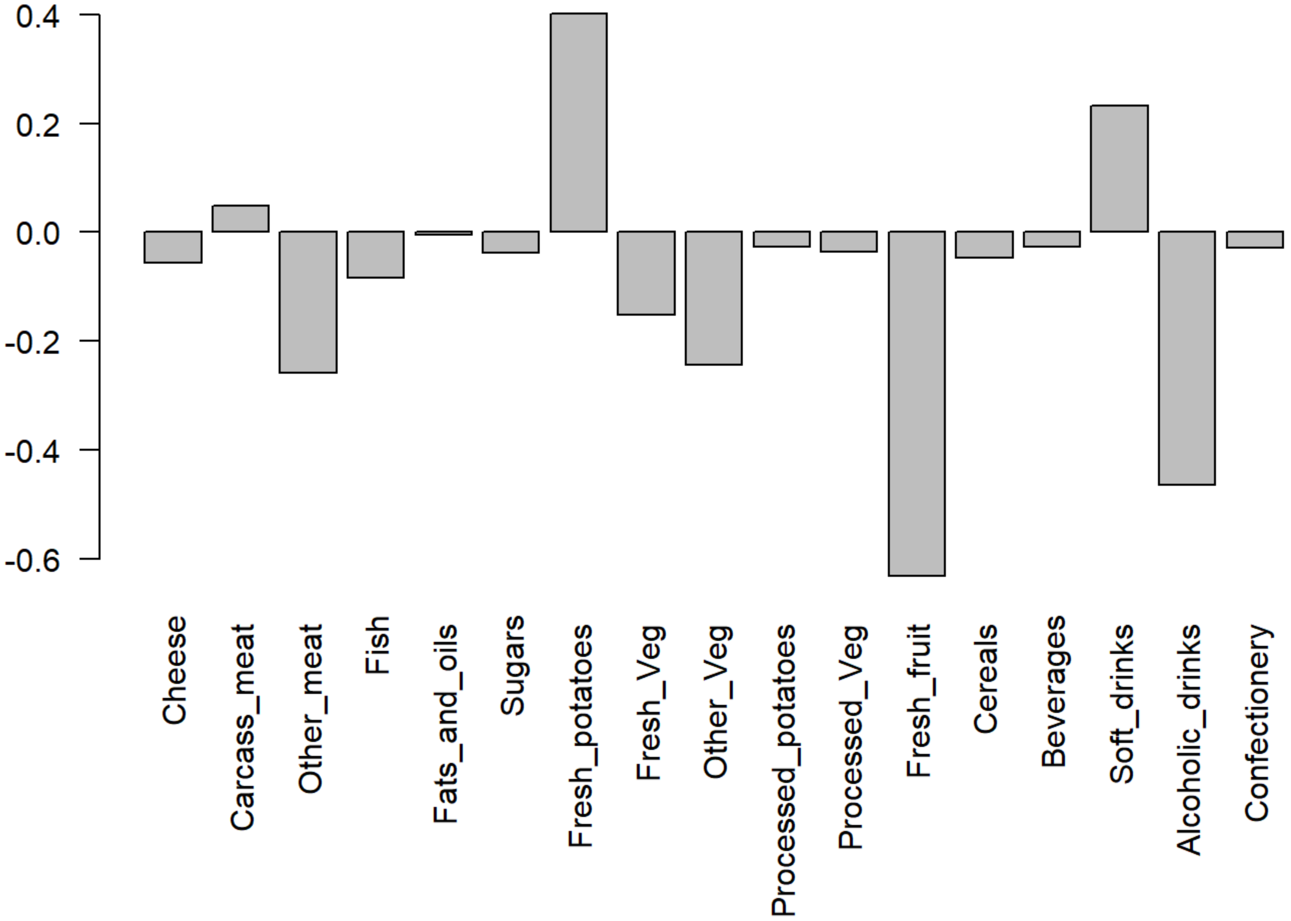
```
                             PC1       PC2      PC3          PC4
Standard deviation     324.15019 212.74780 73.87622 3.175833e-14
Proportion of Variance   0.67444   0.29052  0.03503 0.000000e+00
Cumulative Proportion    0.67444   0.96497  1.00000 1.000000e+00
```

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

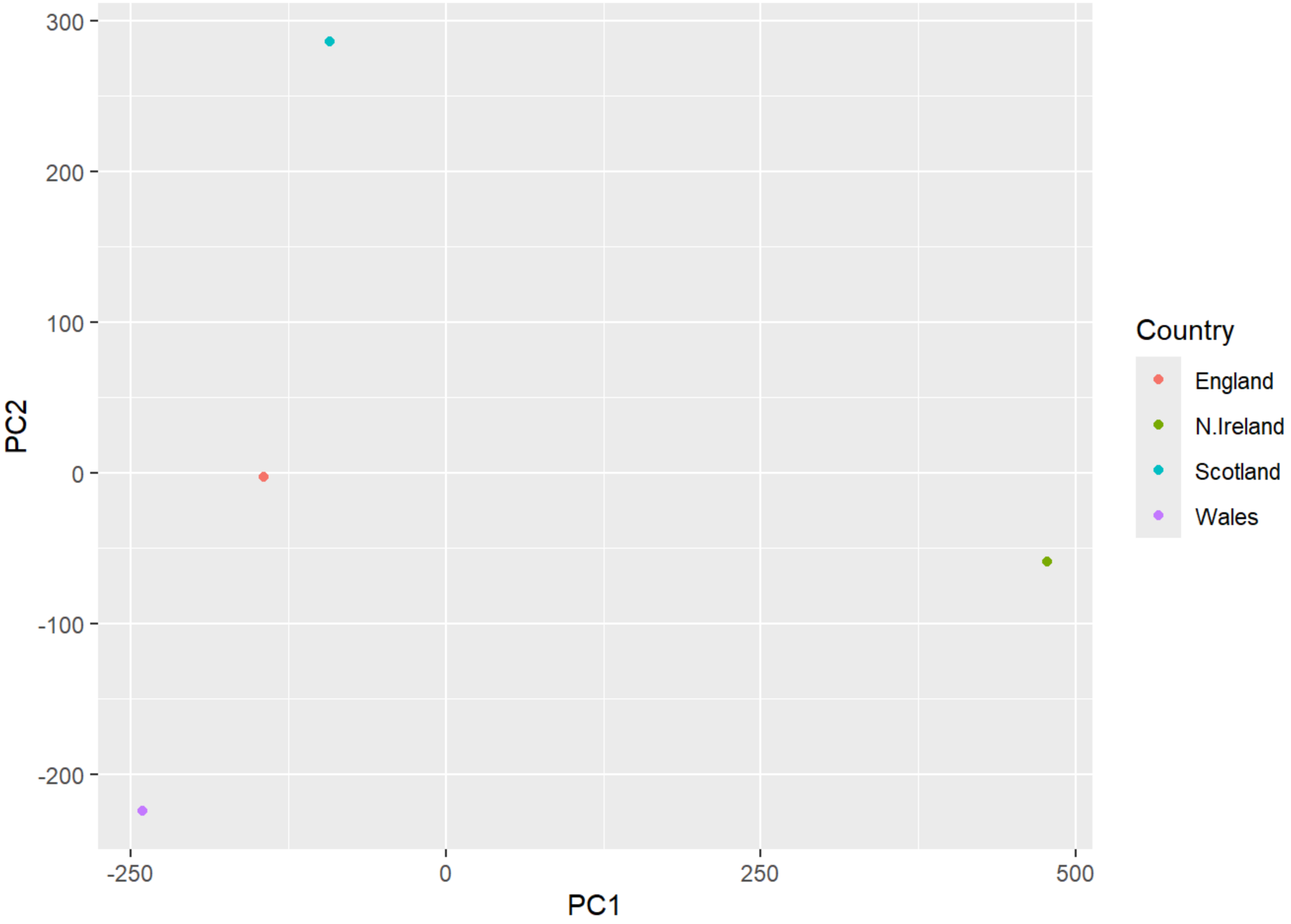


```r
##  PC1 -  accounts for > 90% of variance
par(mar=c(10, 3, 0.35, 0))
barplot( pca$rotation[,1], las=2 )

library(ggplot2)
```

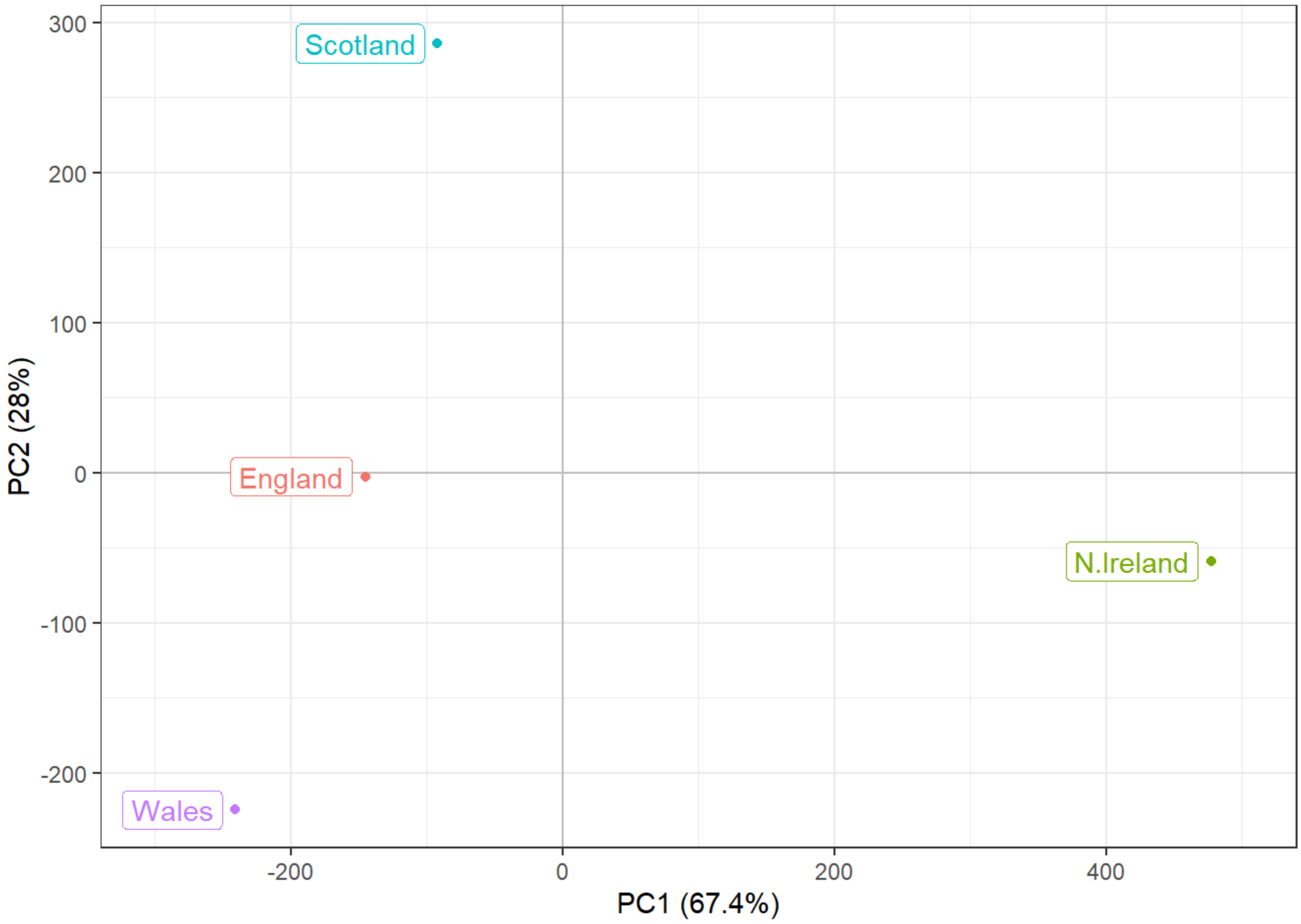
```
Warning: package 'ggplot2' was built under R version 4.3.3
```



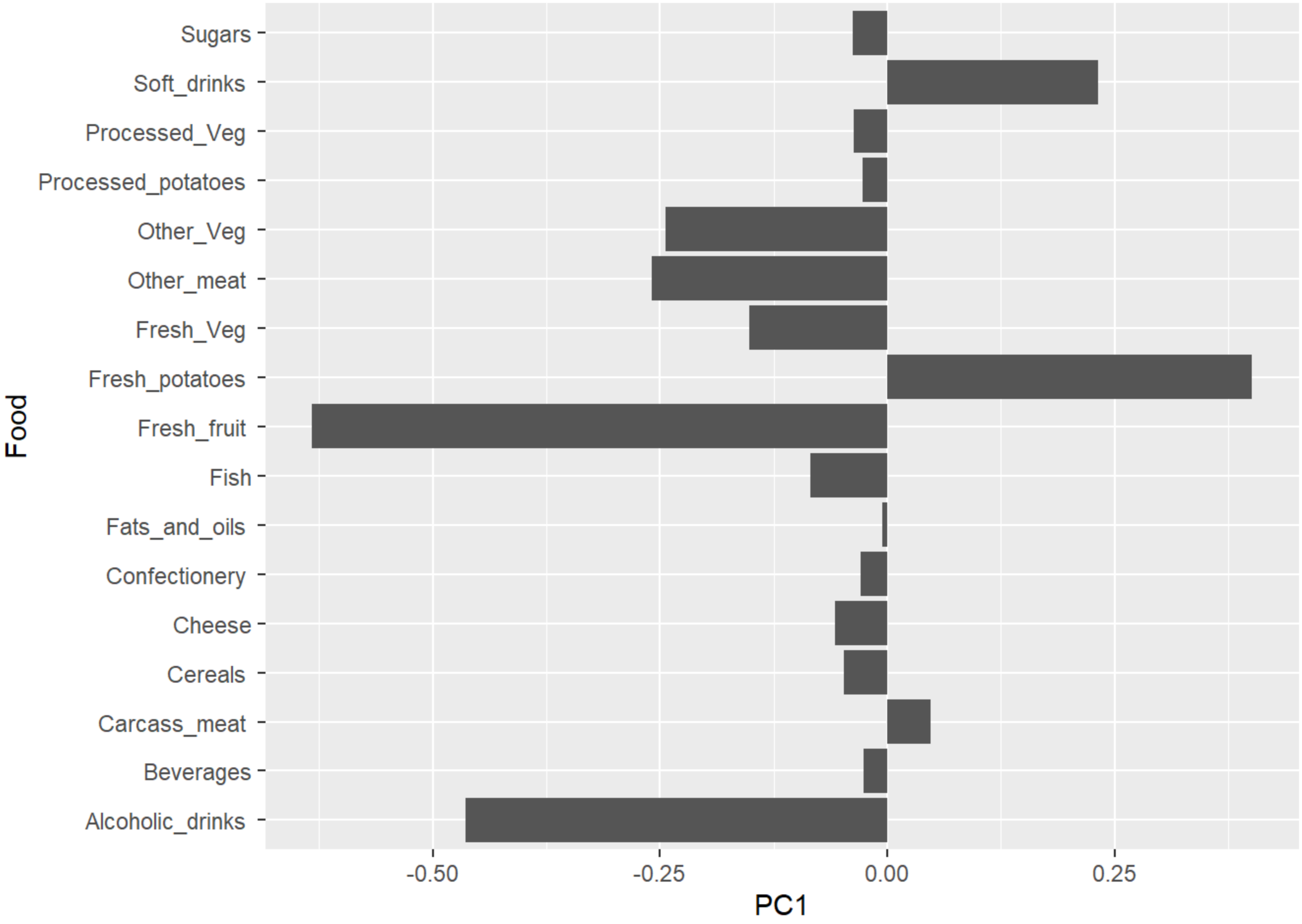
```r
df <- as.data.frame(pca$x)
df_lab <- tibble::rownames_to_column(df, "Country")

# first_plot
ggplot(df_lab) +
  aes(PC1, PC2, col=Country) +
  geom_point()
```



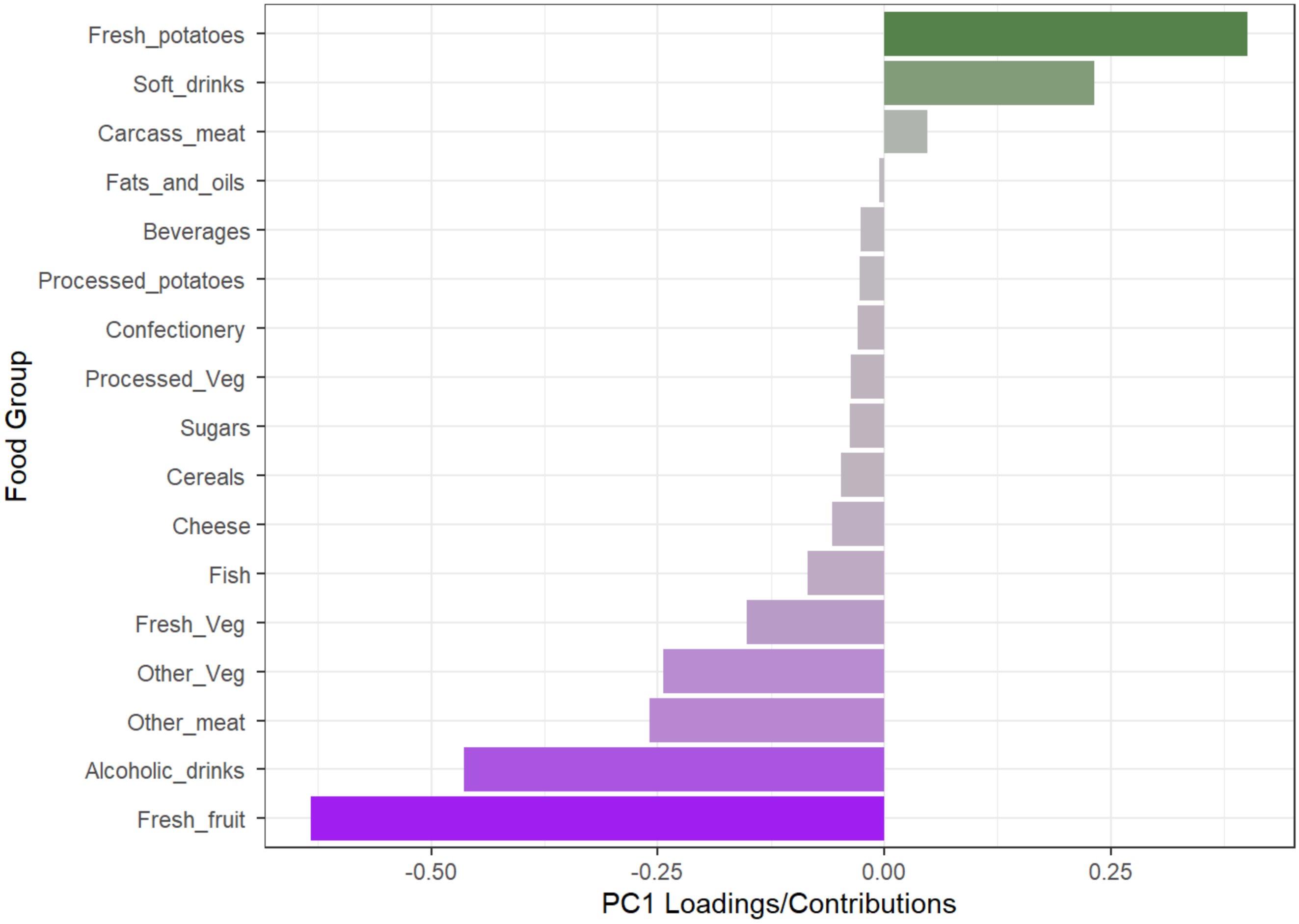
```
ggplot(df_lab) +
  aes(PC1, PC2, col=Country, label=Country) +
  geom_hline(yintercept = 0, col="gray") +
  geom_vline(xintercept = 0, col="gray") +
  geom_point(show.legend = FALSE) +
  geom_label(hjust=1, nudge_x = -10, show.legend = FALSE) +
  expand_limits(x = c(-300,500)) +
  xlab("PC1 (67.4%)") +
  ylab("PC2 (28%)") +
  theme_bw()
```



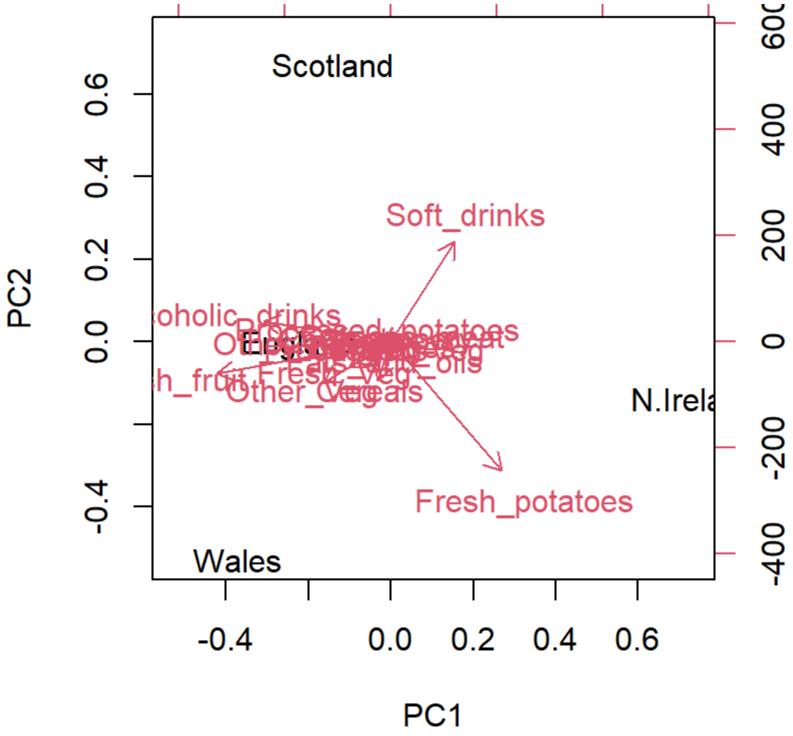
```r
ld <- as.data.frame(pca$rotation)
ld_lab <- tibble::rownames_to_column(ld, "Food")

ggplot(ld_lab) +
  aes(PC1, Food) +
  geom_col()
```



```
#add color
ggplot(ld_lab) +
  aes(PC1, reorder(Food, PC1), bg=PC1) +
  geom_col() +
  xlab("PC1 Loadings/Contributions") +
  ylab("Food Group") +
  scale_fill_gradient2(low="purple", mid="gray", high="darkgreen", guide=NULL) +
  theme_bw()
```



```
## biplot() - small datasets
biplot(pca)
```



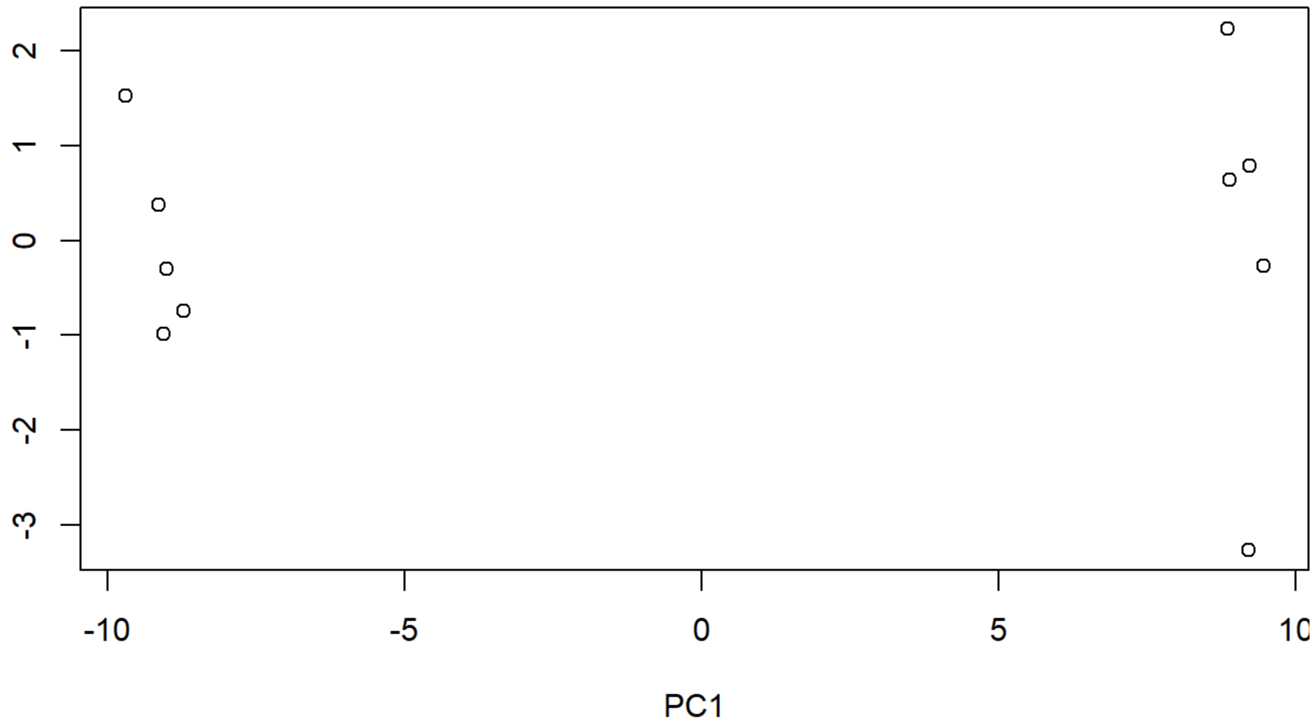
```
url2 <- "https://tinyurl.com/expression-CSV"
rna.data <- read.csv(url2, row.names=1)
head(rna.data)
```

```
       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?

#A10: 100 genes, 10 samples.

str(rna.data)
```

```
'data.frame':	100 obs. of  10 variables:
 $ wt1: int  439 219 1006 783 181 460 27 175 658 121 ...
 $ wt2: int  458 200 989 792 249 502 30 182 669 116 ...
 $ wt3: int  408 204 1030 829 204 491 37 184 653 134 ...
 $ wt4: int  429 210 1017 856 244 491 29 166 633 117 ...
 $ wt5: int  420 187 973 760 225 493 34 180 657 133 ...
 $ ko1: int  90 427 252 849 277 612 304 255 628 931 ...
 $ ko2: int  88 423 237 856 305 594 304 291 627 941 ...
 $ ko3: int  86 434 238 835 272 577 285 305 603 990 ...
 $ ko4: int  90 433 226 885 270 618 311 271 635 982 ...
 $ ko5: int  93 426 210 894 279 638 285 269 620 934 ...
```

```
## Take the transpose of our data
pca <- prcomp(t(rna.data), scale=TRUE)

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

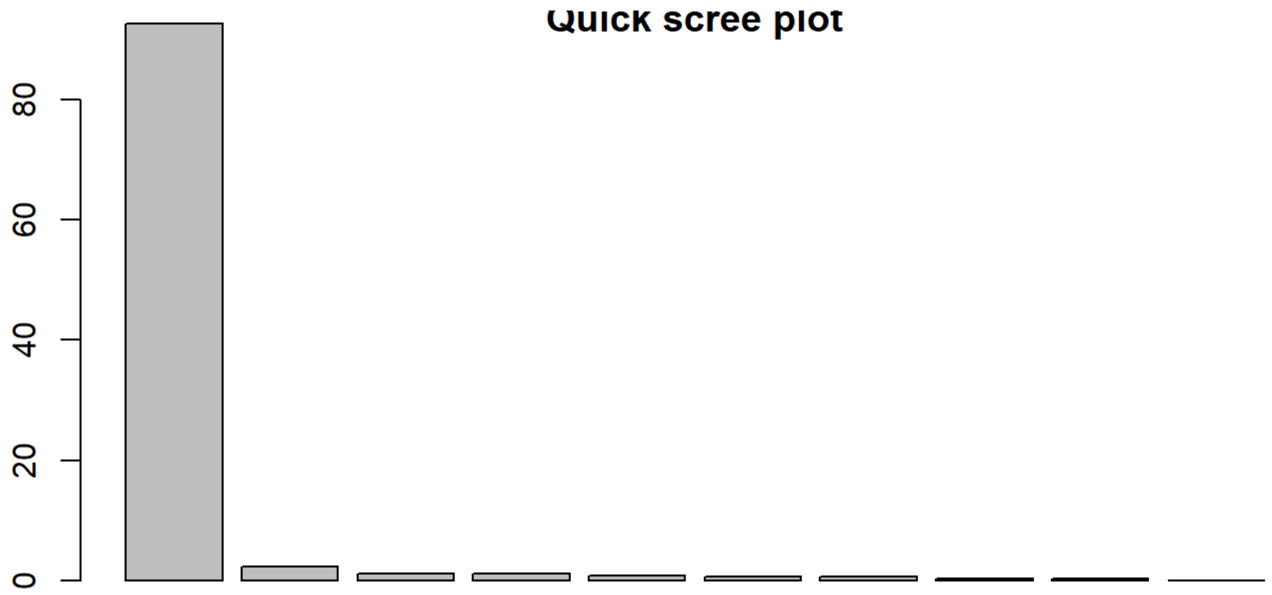


```
#summary
summary(pca)
```

```
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 3.457e-15
```

```
Proportion of Variance 0.00385 0.00364 0.000e+00
Cumulative Proportion  0.99636 1.00000 1.000e+00
```

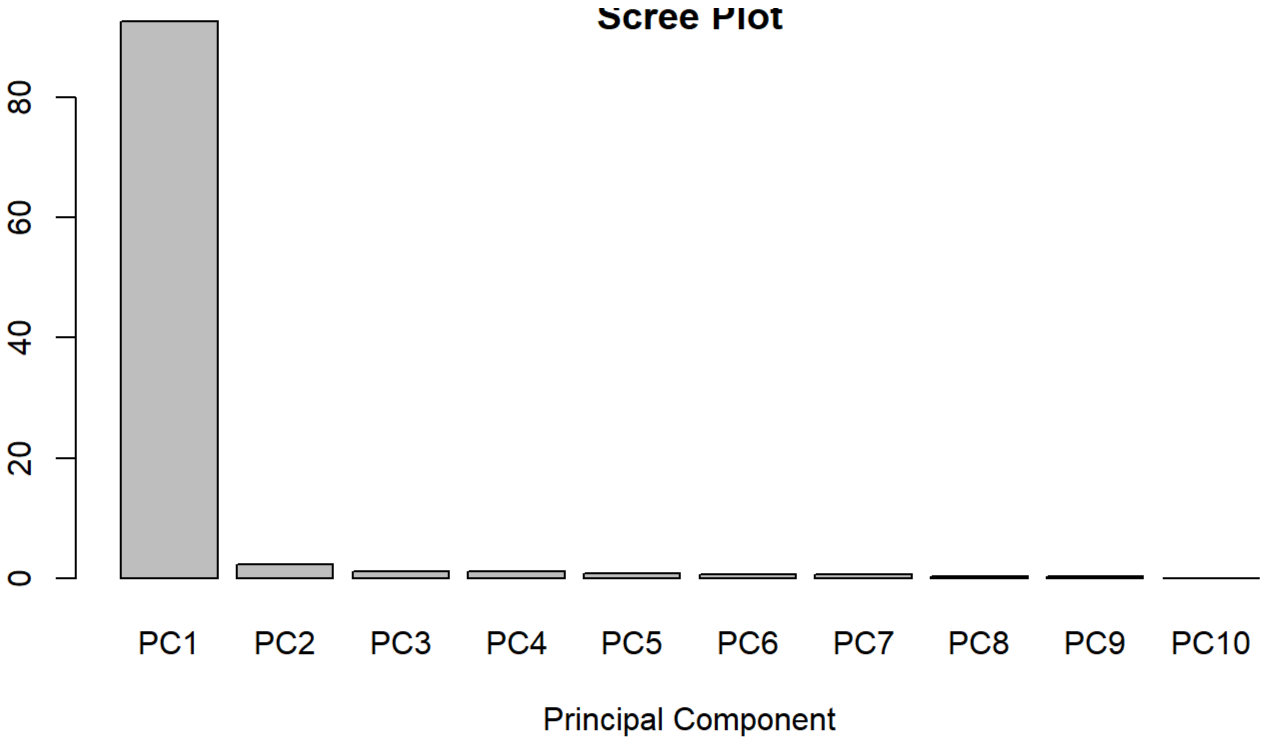
```
plot(pca, main="Quick scree plot")
```



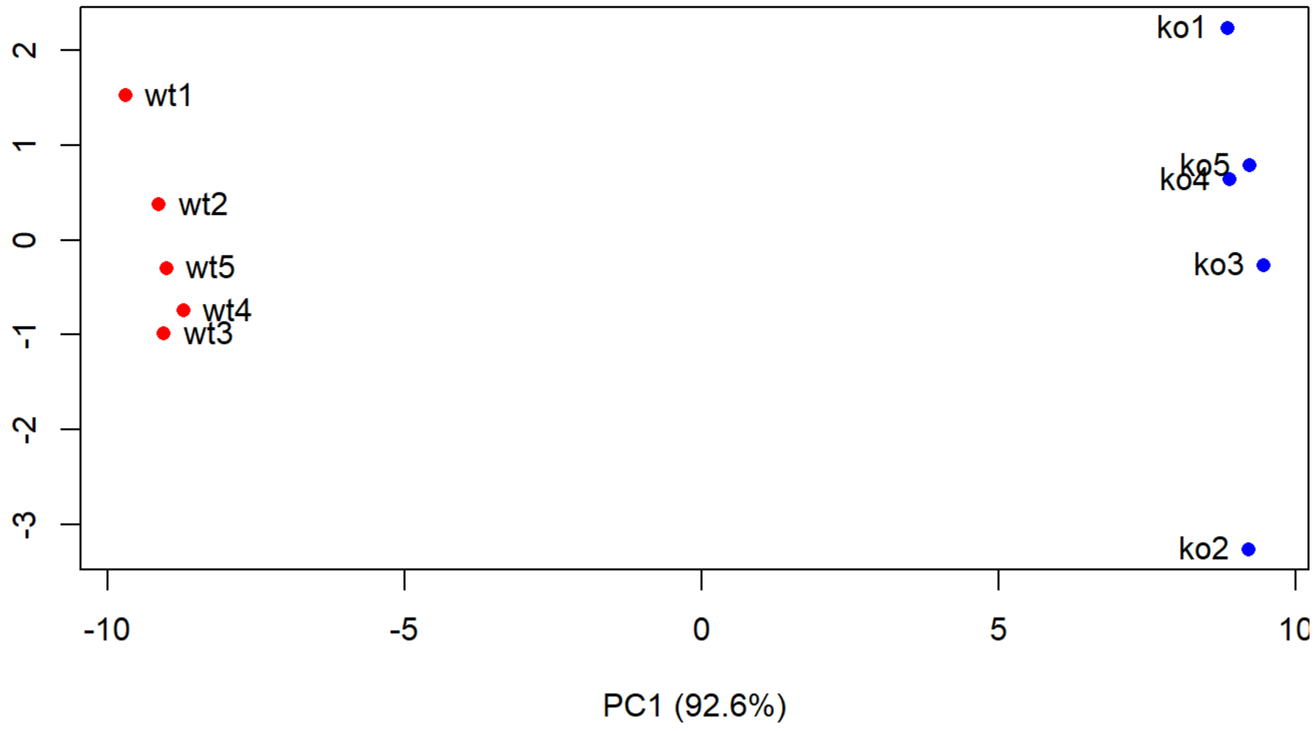
```
## Variance captured per PC
pca.var <- pca$sdev^2

## Percent variance - more informative visually
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
```

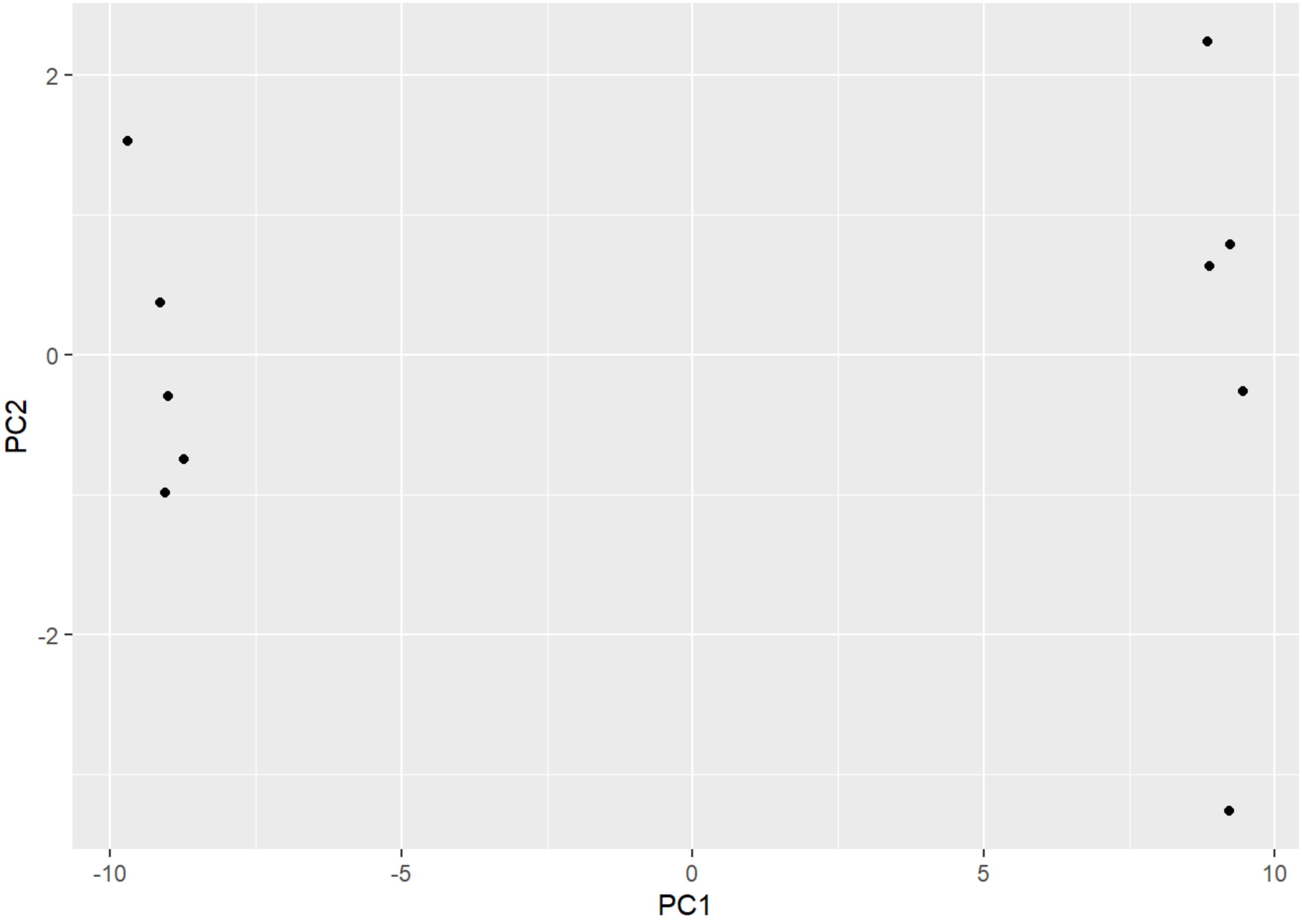
```
barplot(pca.var.per, main="Scree Plot",
        names.arg = paste0("PC", 1:10),
        xlab="Principal Component", ylab="Percent Variation")
```



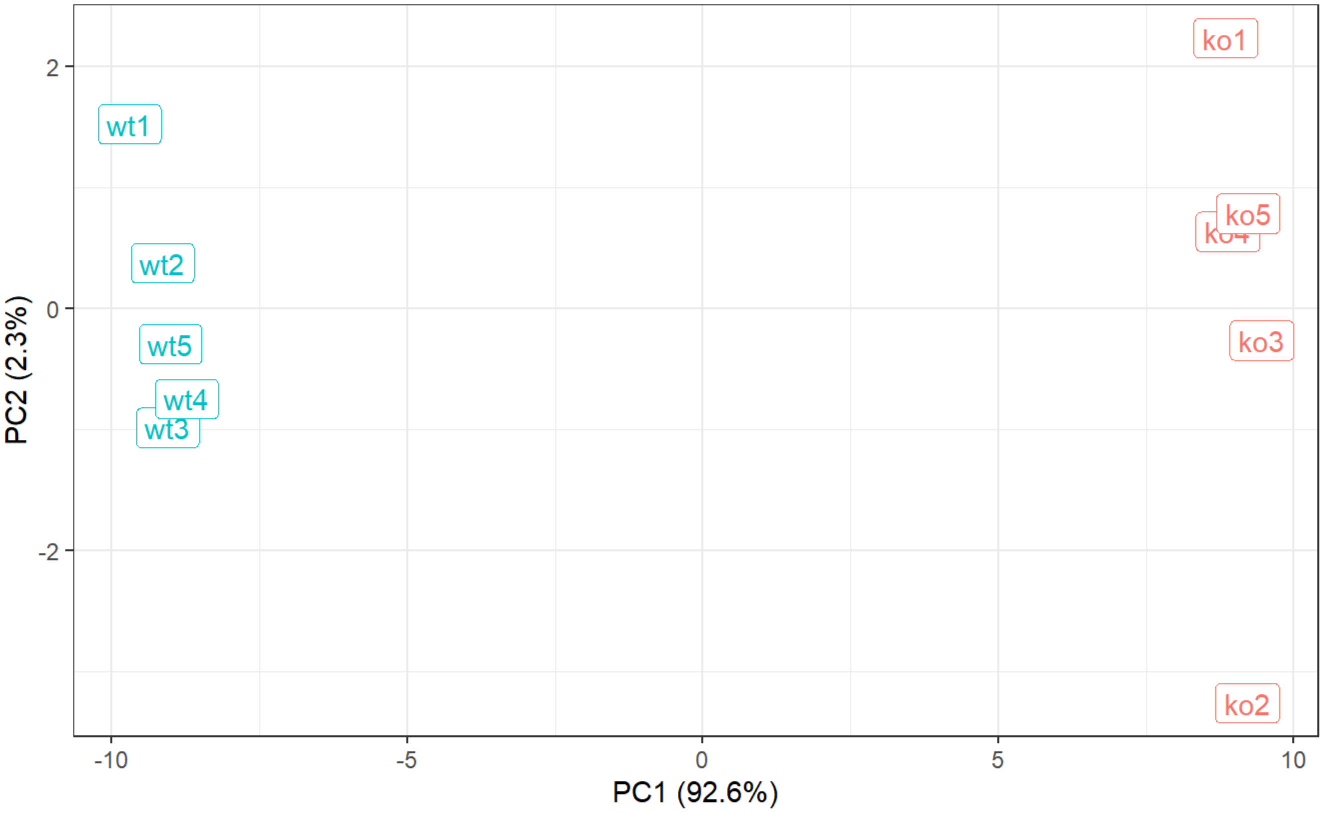
```r
## A vector of colors for wt and ko samples
colvec <- colnames(rna.data)
colvec[grep("wt", colvec)] <- "red"
colvec[grep("ko", colvec)] <- "blue"

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)))
```



```
library(ggplot2)

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

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



```
# Add a 'wt' and 'ko' "condition" column
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 + 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

Class example data

```r
loading_scores <- pca$rotation[,1]

## Find the top 10 measurements (genes) that contribute
## most to PC1 in either direction (+ or -)
gene_scores <- abs(loading_scores)
gene_score_ranked <- sort(gene_scores, decreasing=TRUE)

## show the names of the top 10 genes
top_10_genes <- names(gene_score_ranked[1:10])
top_10_genes
```

```
[1] "gene100" "gene66"  "gene45"  "gene68"  "gene98"  "gene60"  "gene21" 
[8] "gene56"  "gene10"  "gene90" 
```