Class 07 Lab: Machine Learning

Kira Jung

1. PCA of UK food data

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

Question 1.

There are 17 rows and 5 columns in the data frame.

```
nrow(x)
```

```
[1] 17
```

```
ncol(x)
```

```
[1] 5
```

Checking 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
```

Correcting the row-names:

```
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
```

Now checking the rows and columns again:

```
dim(x)
```

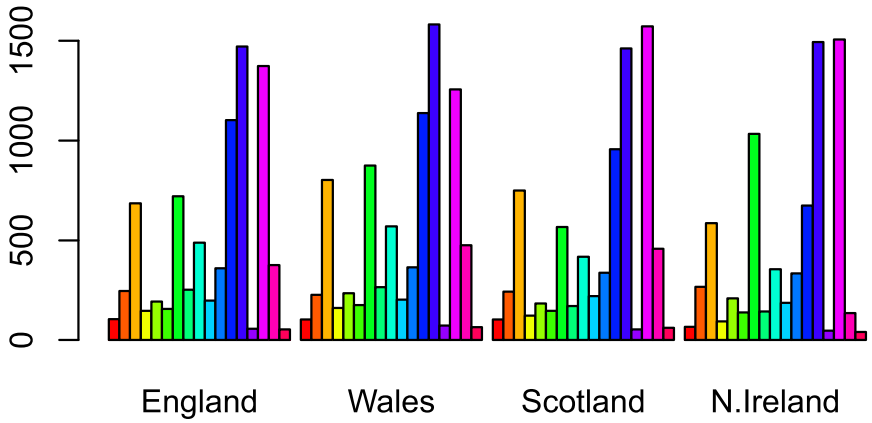
```
[1] 17  4
```

Question 2.

In this situation, running the code x <- read.csv(url, row.names=1) and head(x) seems simpler and more intuitive. With both approaches you would still be deleting the first column of the most recent data frame if you were to run the code block multiple times. So in that sense, I neither method seems to be more robust than the other.

Generating a regular bar-plot:

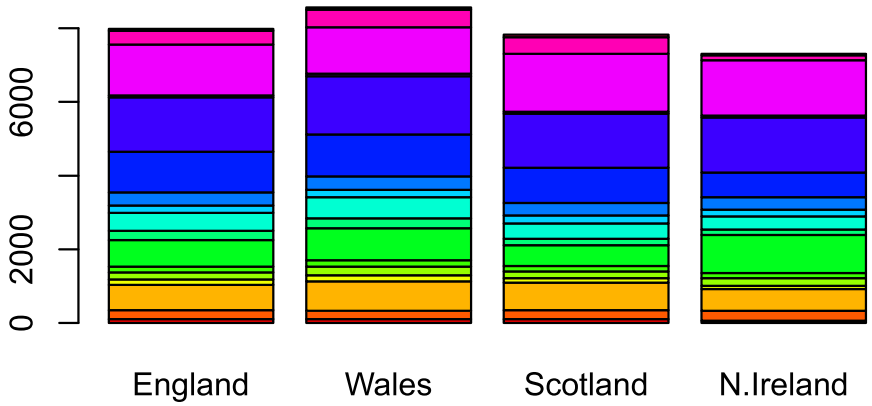
```
barplot(as.matrix(x), beside = T, col = rainbow(nrow(x)))
```



Question 3.

Removing the argument beside = T from the code block above results in the plot below.

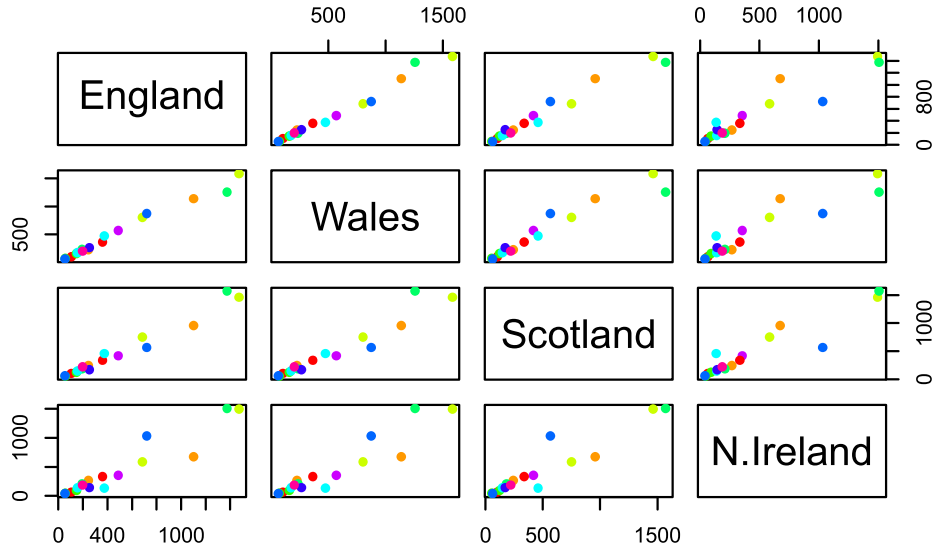
```
barplot(as.matrix(x), col = rainbow(nrow(x)))
```



Question 5.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?

This produces a plot of all possible combinations of countries against each other. the points on the diagonal correlate to if that particular food is eaten more frequently in a given country. Say one of the pairwise plots is comparing England and Wales (e.g. the plot on the first row, second column). Then the dot will be placed higher if that food is consumed more in England, and lower if the food is consumed more in Wales.

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



Question 6.

The main differences between N. Ireland and the other countries of the UK in this data-set is that N. Ireland seems to have higher & lower consumption of certain foods in comparison to other countries, in rates that are much different than other countries being compared. This creates what looks like outliers in the plots that are only so severe when N. Ireland is one of the countries being compared (can see this in column 4 or in row 4).

Performing PCA using prcomp():

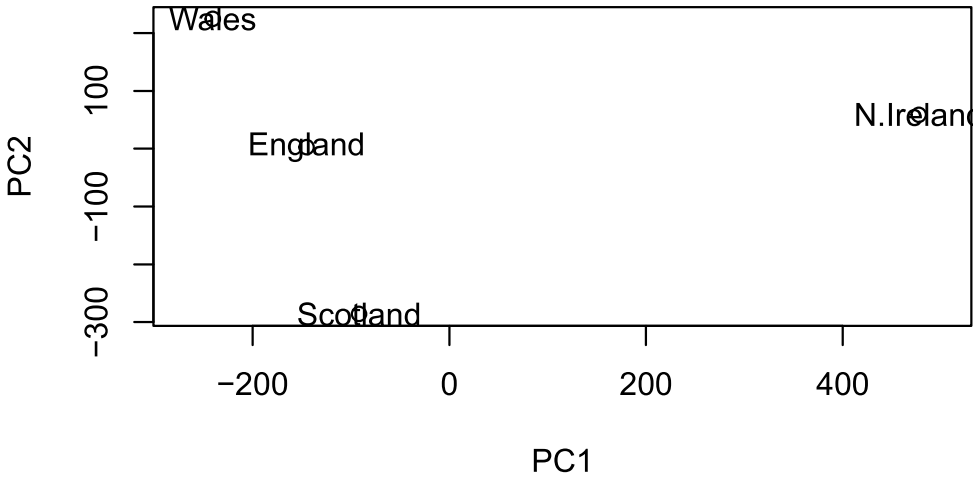
```
pca <- prcomp( t(x) )
summary(pca)
```

```
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
```

Question 7.

Generating a plot of PC1 vs PC2

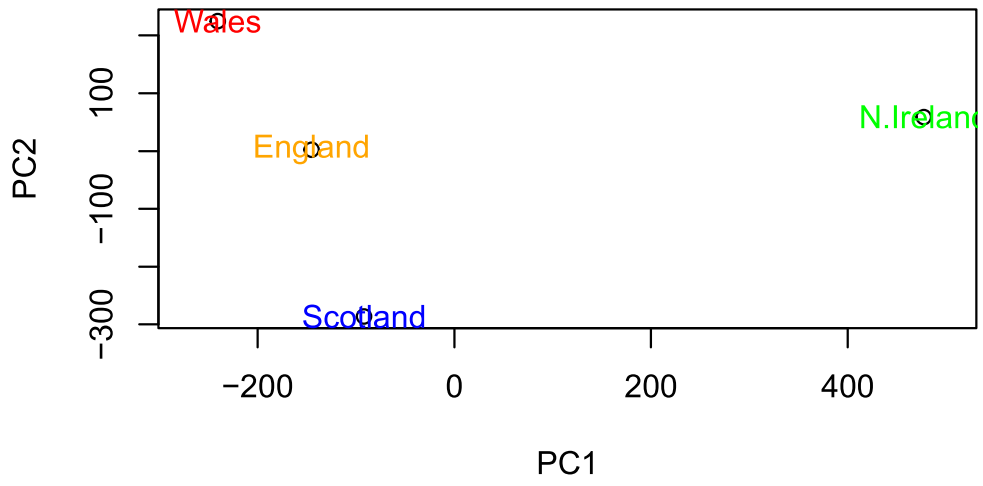
```
plot(pca$x[, 1], pca$x[, 2], xlab="PC1", ylab="PC2", xlim=c(-270,500))
text(pca$x[,1], pca$x[,2], colnames(x))
```



Question 8.

Customizing the plot so the colors of the country names match the colors in the UK and Ireland map and table:

```
country_cols <- c("orange", "red", "blue", "green")
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 = country_cols)
```



Calculating how much variation in the original data each PC accounts for:

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

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

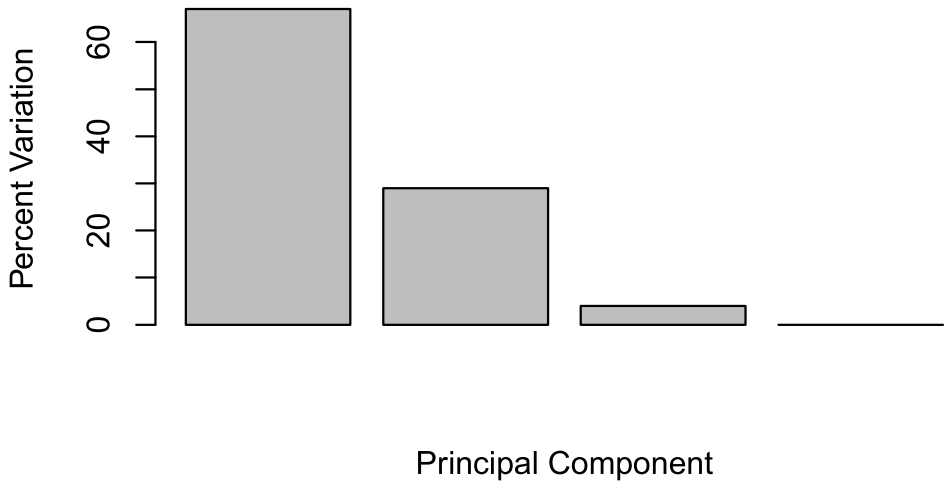
the second row:

```
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
```

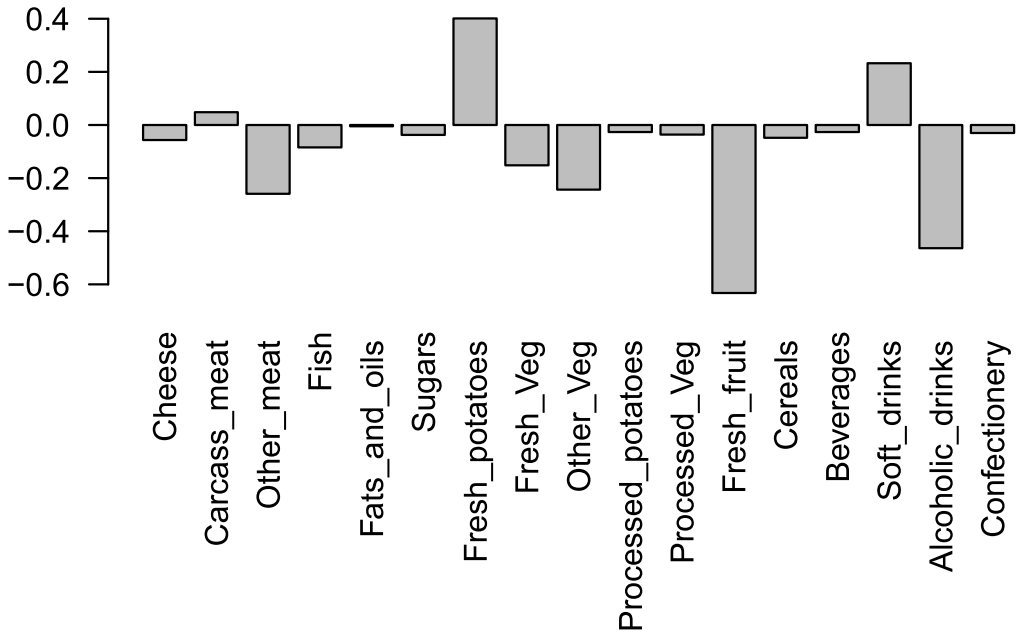
Summarizing the information above in a plot of the variances with respect to the principal component number:

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



Variable loadings, focusing on PC1:

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

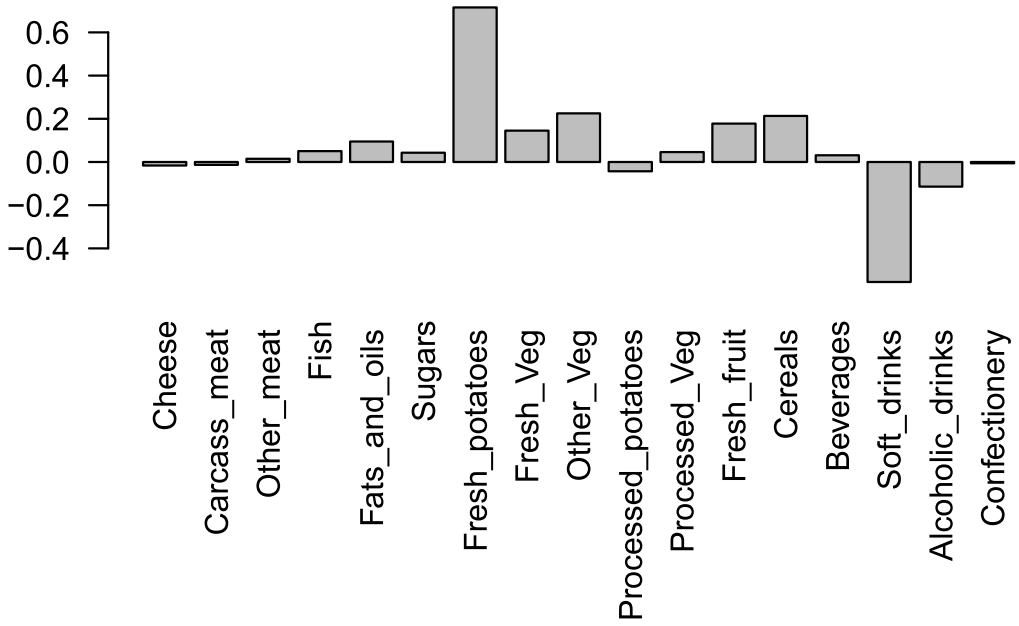


Question 9.

In the loadings plot for PC2, the largest positive loading score is fresh potatoes and the largest negative is soft drinks. We can also see changes in vegetables, fresh fruit, and alcoholic beverages between PC1 and PC2. Based on PC2, the other countries that are not N. Ireland drink more alcohol and eat more fresh fruit while N. Ireland consumes more potatoes and soft drinks.

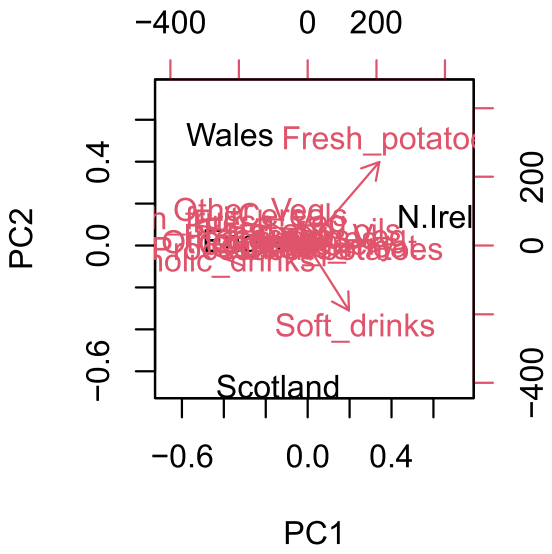
Generating a similar loadings plot for PC2:

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



Creating a biplot:

```
biplot(pca)
```



2. PCA of RNA-seq data

Loading the small RNA-seq count dataset:

```
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
```

Question 10.

There are 10 samples and 100 genes

```
ncol(rna.data)
```

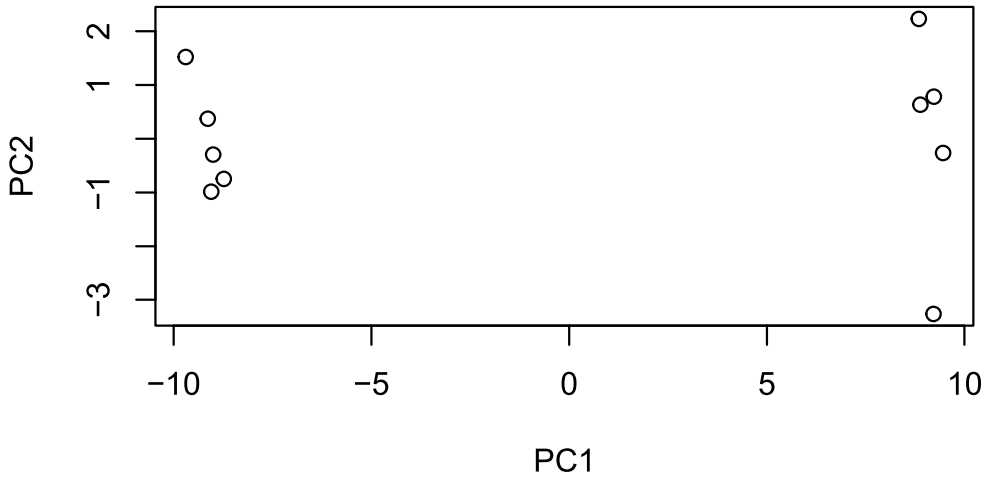
```
[1] 10
```

```
nrow(rna.data)
```

```
[1] 100
```

PCA:

```
pca <- prcomp(t(rna.data), scale=TRUE)
plot(pca$x[,1], pca$x[,2], xlab="PC1", ylab="PC2")
```



Summary of variation in the original data each PC accounts for:

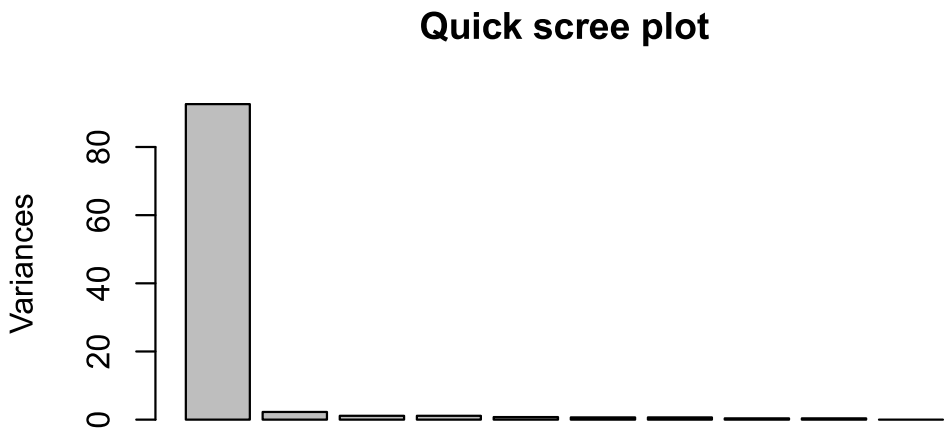
```
summary(pca)
```

```
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
```

Barplot summary of proportion of variance for each PC:

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



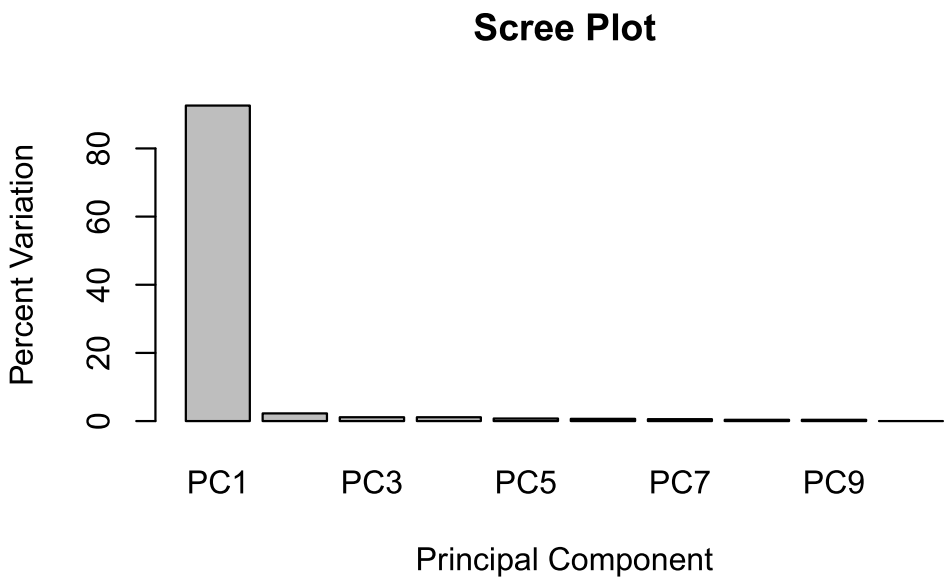
Calculating variance captured per PC:

```
pca.var <- pca$sdev^2
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
```

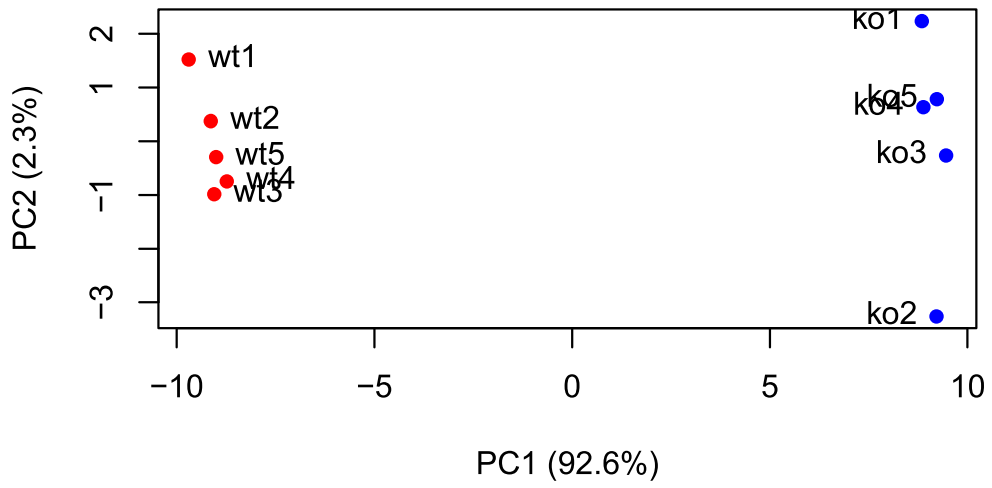
Generating a scree-plot:

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



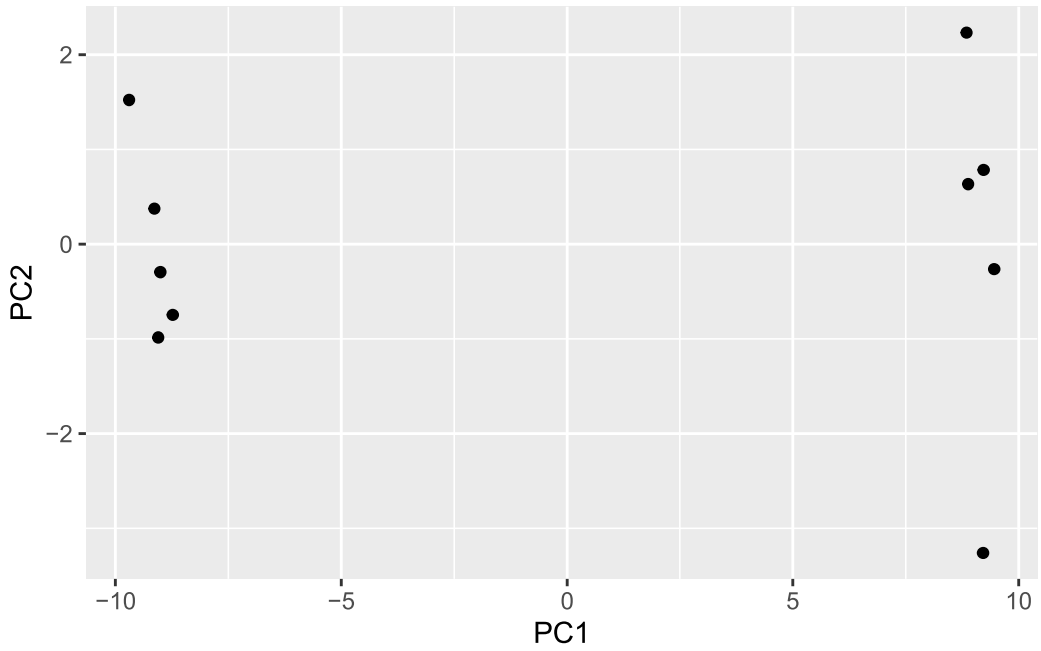
Further editing the main PCA plot:

```
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)))
```



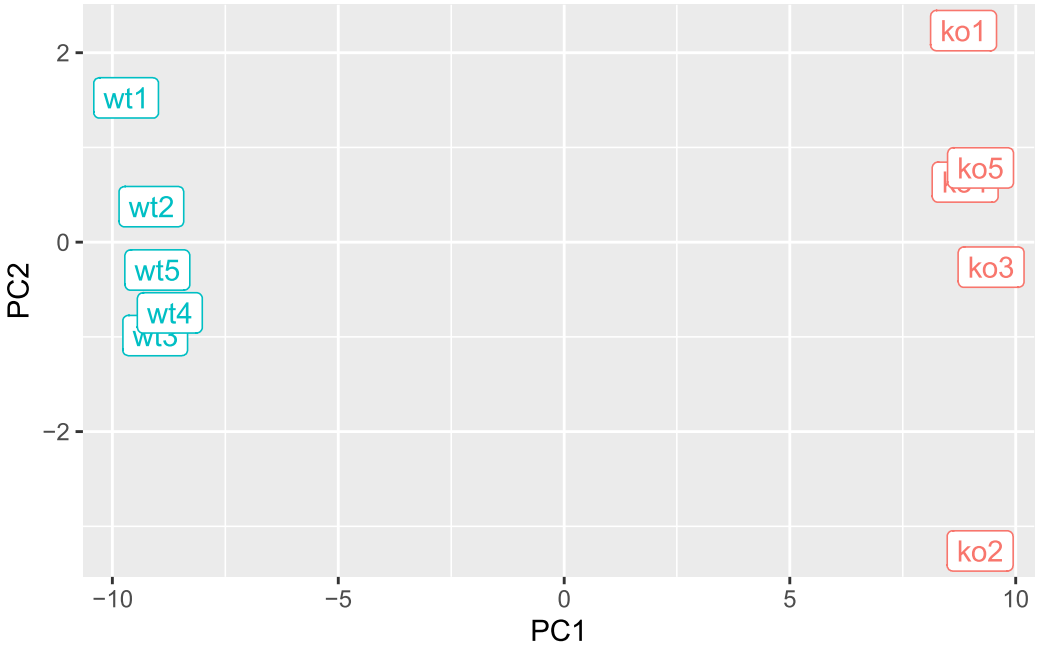
Using ggplot:

```
library(ggplot2)

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

ggplot(df) + aes(PC1, PC2) + geom_point()
```



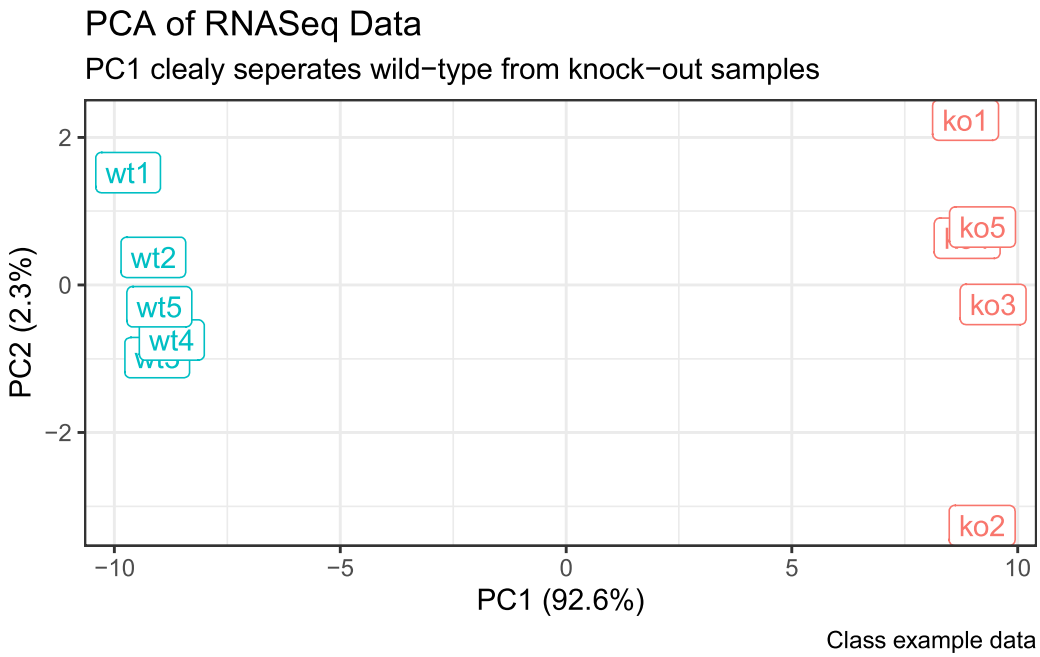
Adding colors and sample label aesthetics for WT and KO samples:

```
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 = F
p
```



Adding titles, subtitles, and axis titles:

```
p + labs(title="PCA of RNASeq Data", subtitle = "PC1 clealy seperates wild-type from knock
```



Gene loadings:

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

gene_scores <- abs(loading_scores)
gene_score_ranked <- sort(gene_scores, decreasing=TRUE)

top_10_genes <- names(gene_score_ranked[1:10])
top_10_genes
```

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

```
sessionInfo()
```

```
R version 4.2.3 (2023-03-15 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 22621)

Matrix products: default
```

```
locale:
[1] LC_COLLATE=English_United States.utf8
[2] LC_CTYPE=English_United States.utf8
[3] LC_MONETARY=English_United States.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.utf8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] ggplot2_3.4.2

loaded via a namespace (and not attached):
 [1] rstudioapi_0.14  knitr_1.43       magrittr_2.0.3   tidyselect_1.2.0
 [5] munsell_0.5.0    colorspace_2.1-0 R6_2.5.1         rlang_1.1.0
 [9] fastmap_1.1.1    fansi_1.0.4      dplyr_1.1.2      tools_4.2.3
[13] grid_4.2.3       gtable_0.3.3     xfun_0.39        utf8_1.2.3
[17] cli_3.6.1        withr_2.5.0      htmltools_0.5.5  yaml_2.3.7
[21] digest_0.6.31    tibble_3.2.1     lifecycle_1.0.3  farver_2.1.1
[25] vctrs_0.6.2      glue_1.6.2       evaluate_0.21    rmarkdown_2.21
[29] labeling_0.4.2   compiler_4.2.3   pillar_1.9.0     generics_0.1.3
[33] scales_1.2.1     jsonlite_1.8.4   pkgconfig_2.0.3
```