Clustering and high dimensional visualization

STAT5003

Dr. Justin Wishart

For the first example, we'll inspect the Classical `iris` dataset and the clustering algorithm of k-means.

k-means based clustering algorithms

Apply k-means clustering on iris data

```
dim(iris)
```

```
## [1] 150   5
```

```
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

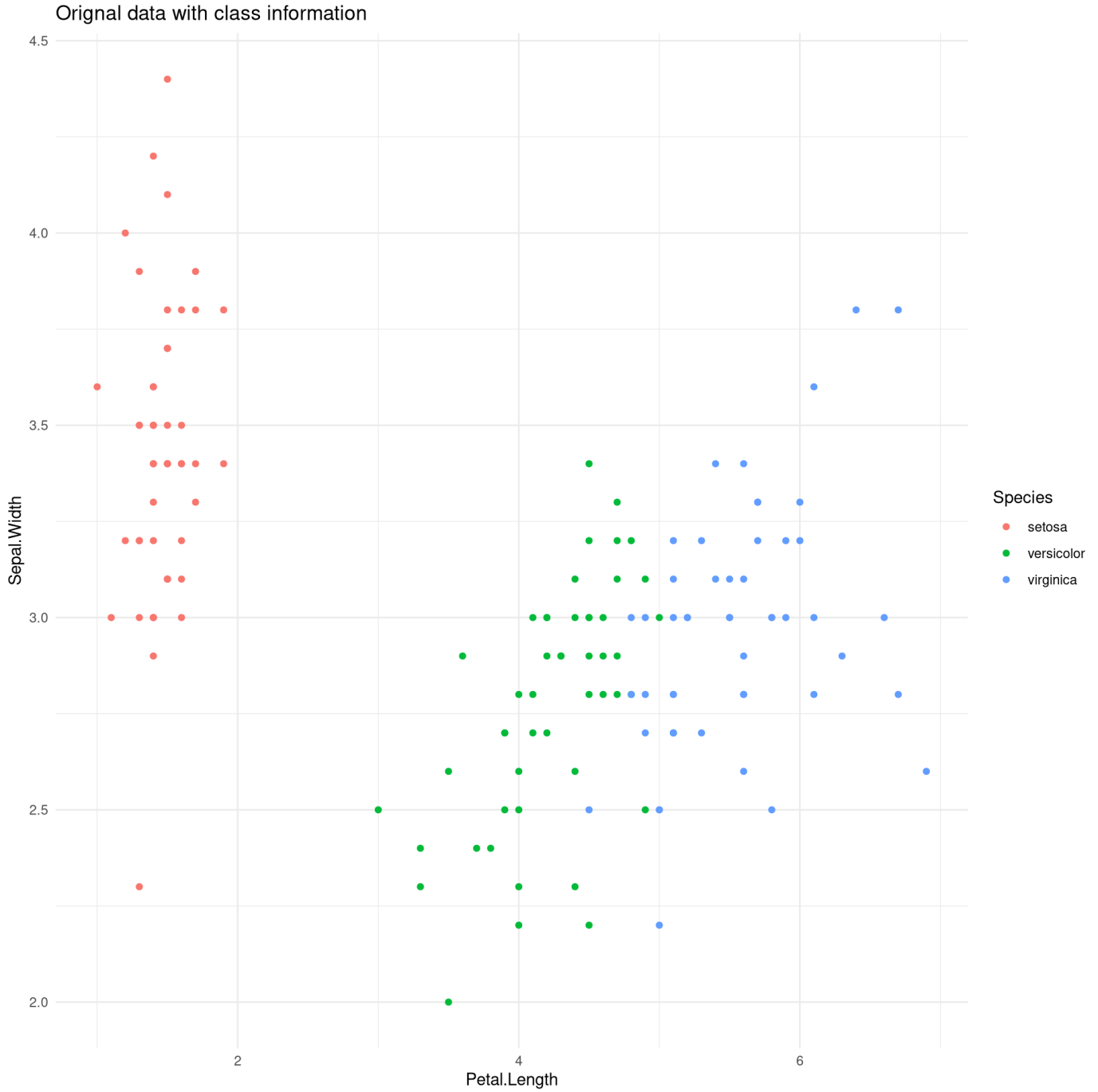
```
table(iris$Species)
```

```
##
##     setosa versicolor  virginica
##         50         50         50
```

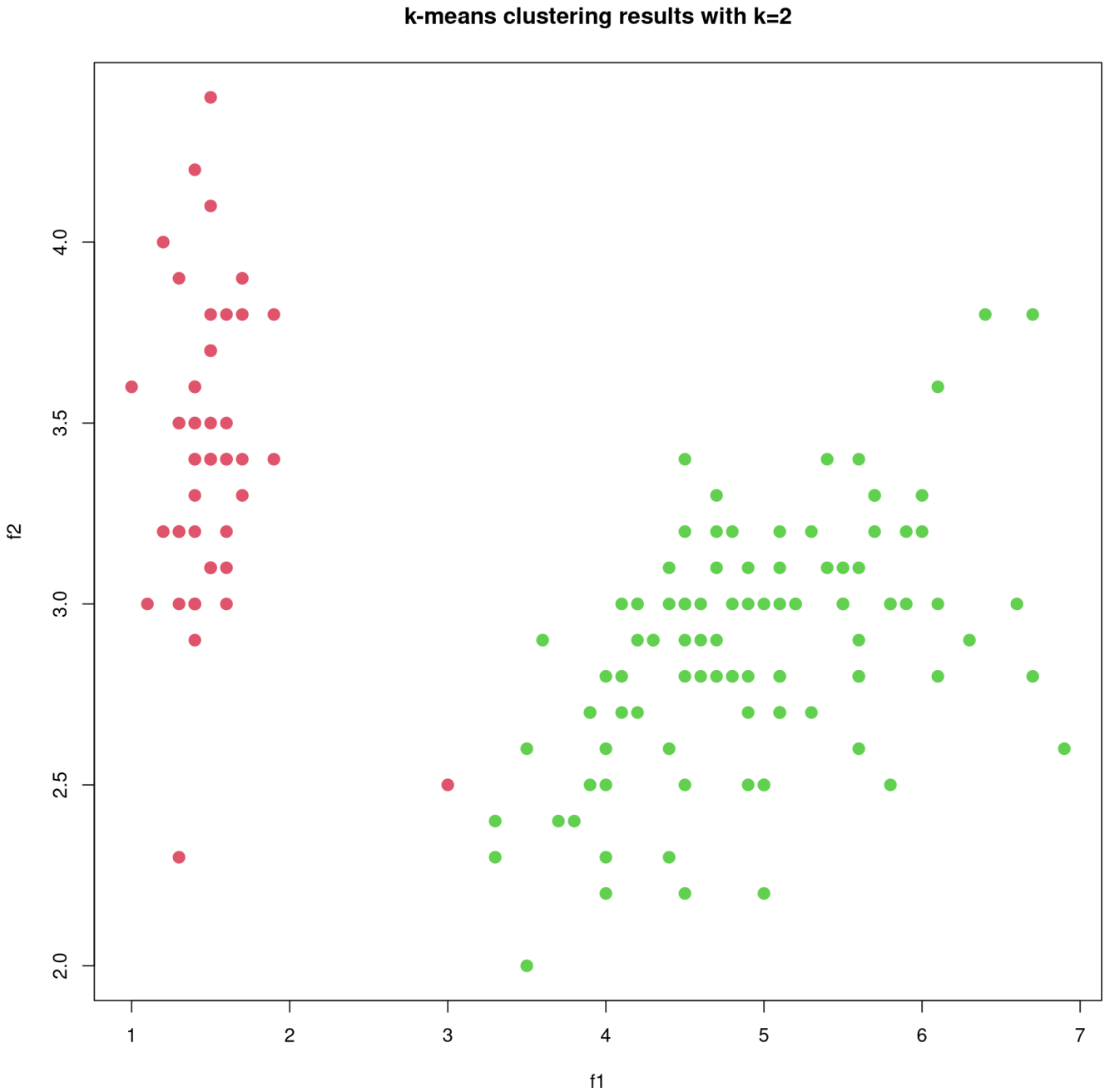
```
class(iris)
```

```
## [1] "data.frame"
```

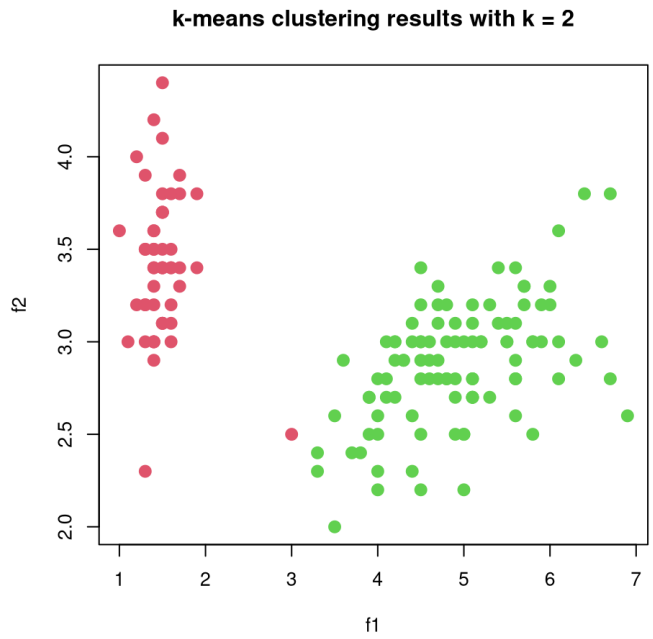
```
# set up plots
ggplot(data.frame(iris), aes(x = Petal.Length, y = Sepal.Width, colour = Species)) +
    geom_point() + ggtitle("Orignal data with class information") + theme_minimal()
```

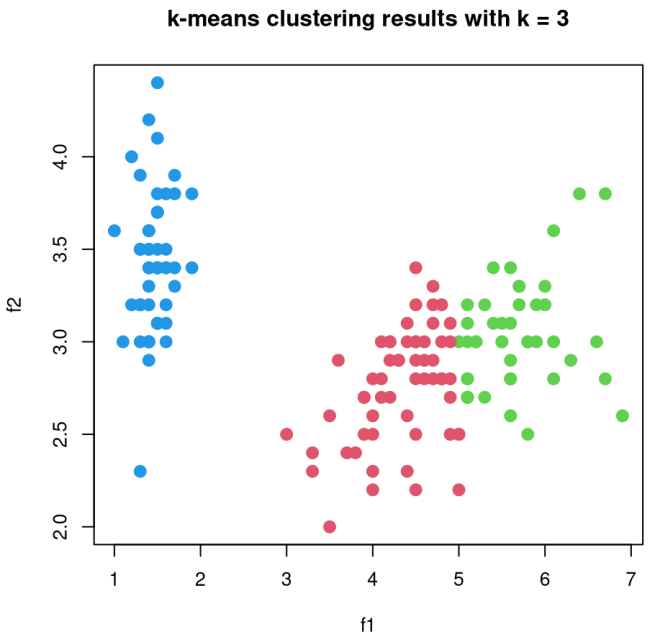


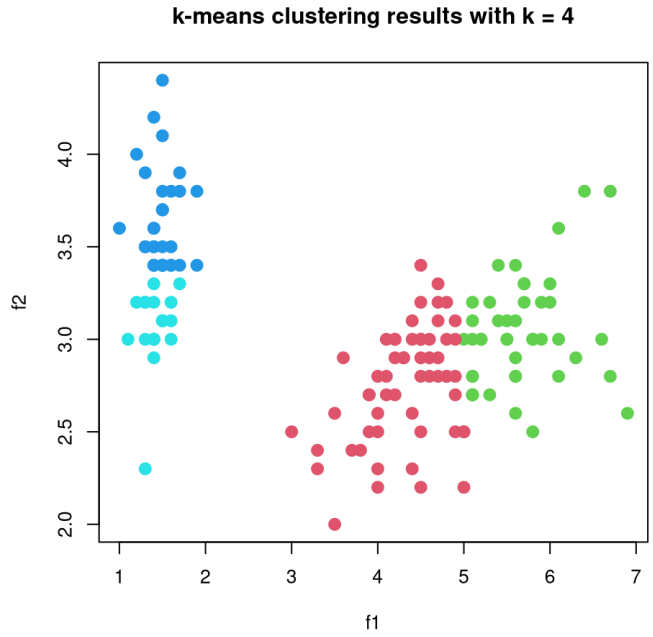

```
# apply k-means with k=2
set.seed(1)
data.mat <- cbind(iris$Petal.Length, iris$Sepal.Width)
km.out2 <- kmeans(data.mat, centers = 2)
plot(data.mat, col = (km.out2$cluster + 1),
     main = "k-means clustering results with k=2",
     xlab = "f1", ylab = "f2", pch = 20, cex = 2)
```

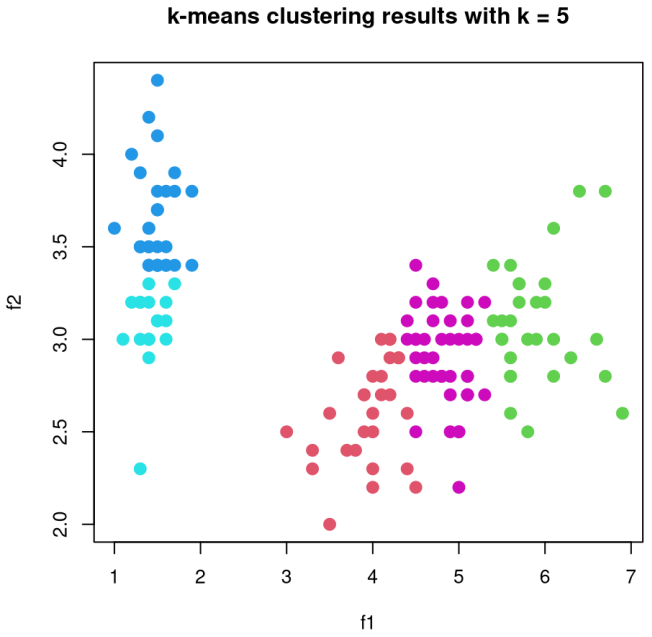



```
kmeans.out <- lapply(2:6, function(k){
    set.seed(1)
    kmeans(data.mat, centers = k)
})
par(mfrow = c(2, 2))
invisible(lapply(2:5, function(x) {
    plot(data.mat, col = (kmeans.out[[x - 1]]$cluster + 1),
         main = paste0("k-means clustering results with k = ", x),
         xlab = "f1", ylab = "f2", pch = 20, cex = 2)
}))
```









Cluster statistics

```
# what is the output from k-means clustering?
str(km.out2)
```

```
## List of 9
##  $ cluster     : int [1:150] 1 1 1 1 1 1 1 1 1 1 ...
##  $ centers     : num [1:2, 1:2] 1.49 4.93 3.41 2.88
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "1" "2"
##   .. ..$ : NULL
##  $ totss       : num 493
##  $ withinss    : num [1:2] 11.7 74.6
##  $ tot.withinss: num 86.3
##  $ betweenss   : num 406
##  $ size        : int [1:2] 51 99
##  $ iter        : int 1
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

```
# check between and within cluster sum of squares for all outputs
betweenss <- vapply(kmeans.out, "[[", numeric(1), "betweenss")
tot.withinss <- vapply(kmeans.out, "[[", numeric(1), "tot.withinss")

names(betweenss) <- names(tot.withinss) <- paste0("k = ", vapply(kmeans.out, function
(x) length(x$centers), numeric(1)))
betweenss
```

```
##    k = 4    k = 6    k = 8   k = 10   k = 12
## 406.3217 451.8953 456.0399 469.2922 473.0172
```

```
tot.withinss
```

```
##    k = 4    k = 6    k = 8   k = 10   k = 12
## 86.31065 40.73707 36.59246 23.34009 19.61514
```

Application of clustering algorithms to gene expression dataset

There is a dataset in the `dslabs` package called `tissue_gene_expression`. It contains gene expression data on 7 tissues.

```
# load data
library(dslabs)
data("tissue_gene_expression")
# extract expression matrix
gene.expr <- tissue_gene_expression[[1]]
# obtain the list of tissues
tissues.labs <- tissue_gene_expression[[2]]
length(tissues.labs)
```

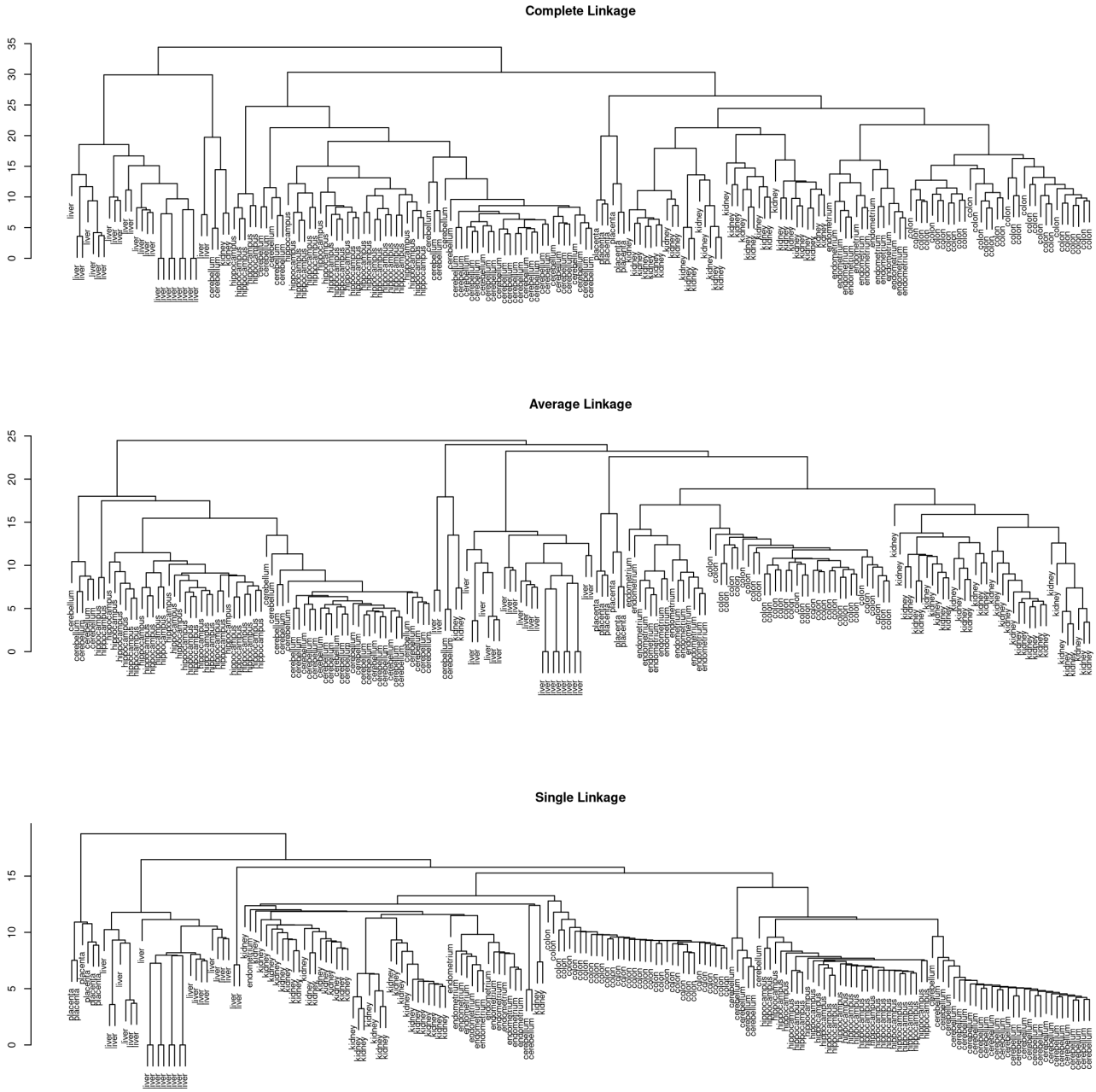
```
## [1] 189
```

```
dim(gene.expr)
```

```
## [1] 189 500
```

Hierarchical clustering analysis

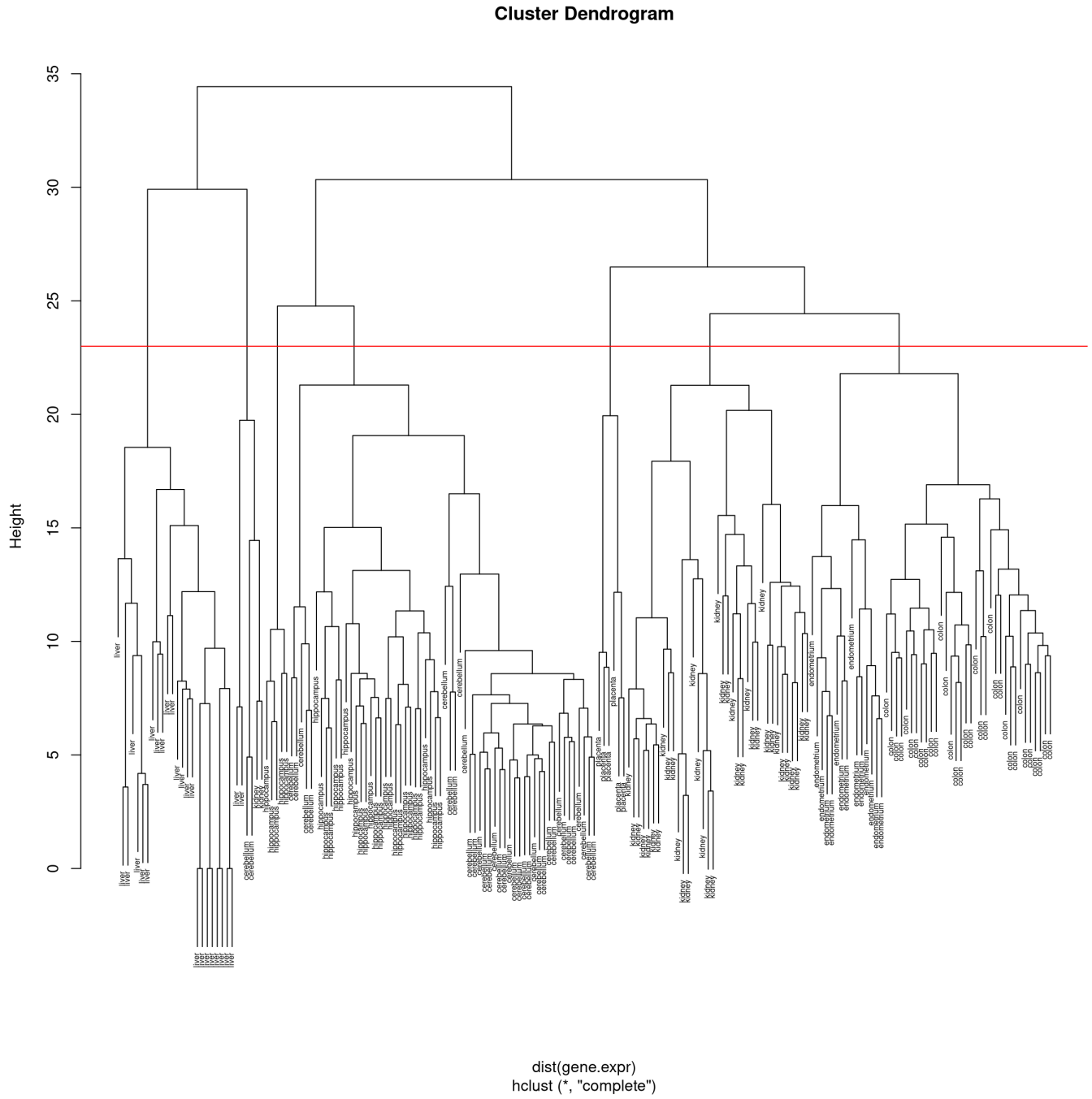
```
par(mfrow = c(3, 1))
data.dist = dist(gene.expr)
plot(hclust(data.dist), labels = tissues.labs,
     main = "Complete Linkage", xlab = "", sub = "", ylab = "", cex = 0.8)
plot(hclust(data.dist, method = "average"), labels = tissues.labs,
     main = "Average Linkage", xlab = "", sub = "", ylab = "", cex = 0.8)
plot(hclust(data.dist, method = "single"), labels = tissues.labs,
     main = "Single Linkage", xlab = "", sub = "", ylab = "", cex = 0.8)
```



```
hc.out = hclust(dist(gene.expr))
hc.clusters = cutree(hc.out, 7)
table(hc.clusters, tissues.labs)
```

```
##             tissues.labs
## hc.clusters cerebellum colon endometrium hippocampus kidney liver placenta
##           1         36     0           0          26      0     0         0
##           2          2     0           0           0      2     2         0
##           3          0    34          15           0      0     0         0
##           4          0     0           0           5      0     0         0
##           5          0     0           0           0     37     0         0
##           6          0     0           0           0      0    24         0
##           7          0     0           0           0      0     0         6
```

```
par(mfrow = c(1, 1))
plot(hc.out, labels = tissues.labs, cex = 0.5)
abline(h = 23, col = "red")
```




```
hc.out
```

```
##
## Call:
## hclust(d = dist(gene.expr))
##
## Cluster method   : complete
## Distance         : euclidean
## Number of objects: 189
```

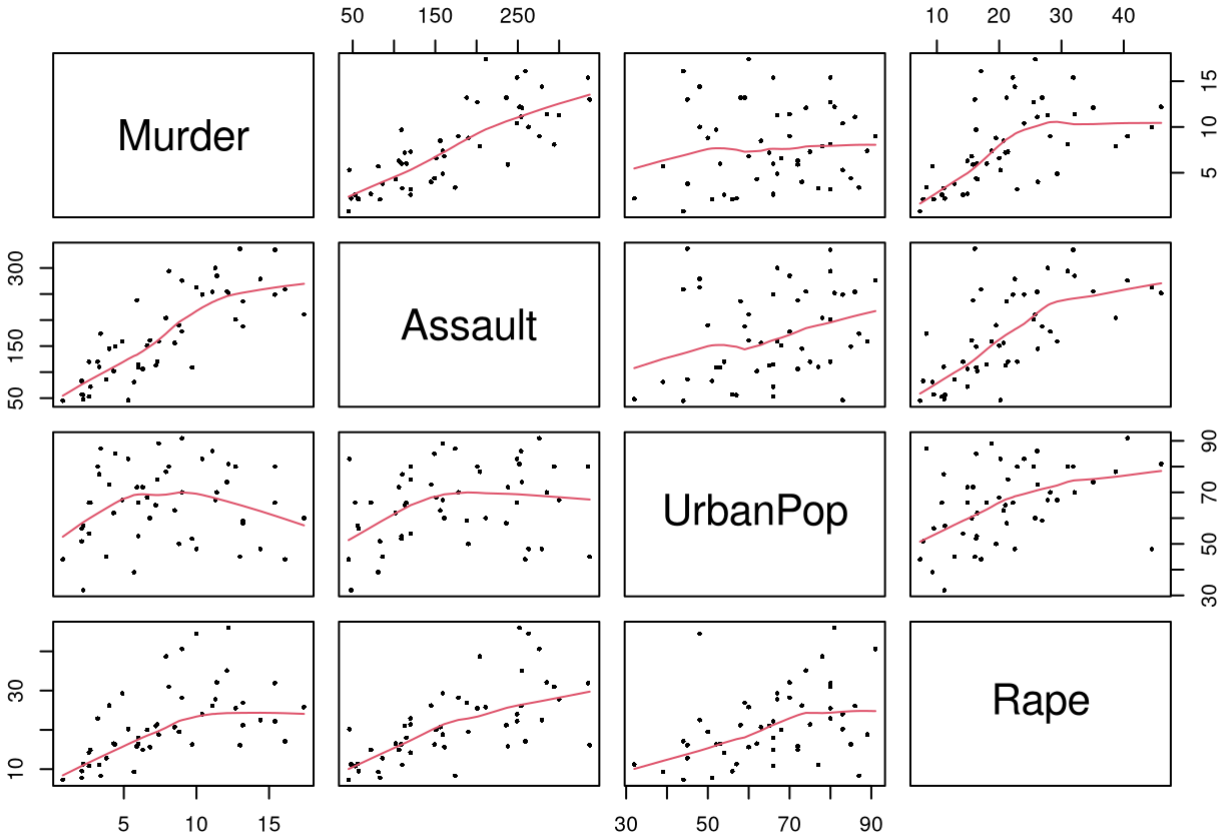
PCA

USArrests dataset

Let's firstly explore PCA with the USArrests data. This dataset, along with the function for PCA `prcomp()`, should be included in your base R installation.

```
data("USArrests")

# Make pairwise plots from the data
pairs(USArrests, pch=16, cex = 0.5, panel = panel.smooth)
```



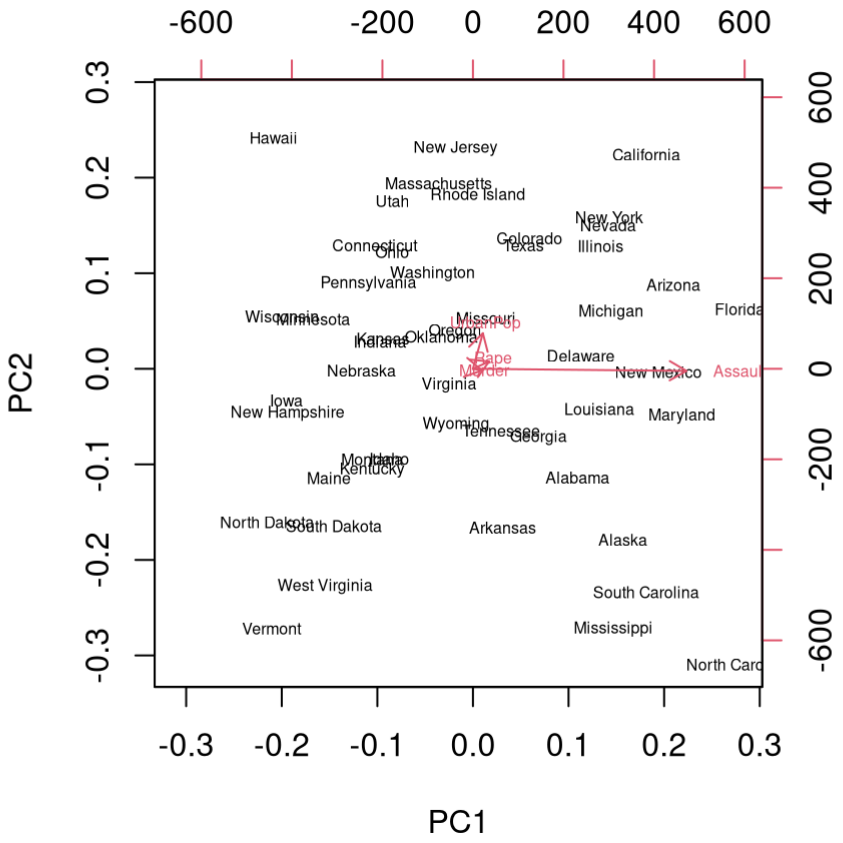

```
# Let's check what the data looks like
# You can see the variables have different scales
summary(USArrests)
```

```
##      Murder          Assault         UrbanPop          Rape
##  Min.   : 0.800   Min.   : 45.0   Min.   :32.00   Min.   : 7.30
##  1st Qu.: 4.075   1st Qu.:109.0   1st Qu.:54.50   1st Qu.:15.07
##  Median : 7.250   Median :159.0   Median :66.00   Median :20.10
##  Mean   : 7.788   Mean   :170.8   Mean   :65.54   Mean   :21.23
##  3rd Qu.:11.250   3rd Qu.:249.0   3rd Qu.:77.75   3rd Qu.:26.18
##  Max.   :17.400   Max.   :337.0   Max.   :91.00   Max.   :46.00
```

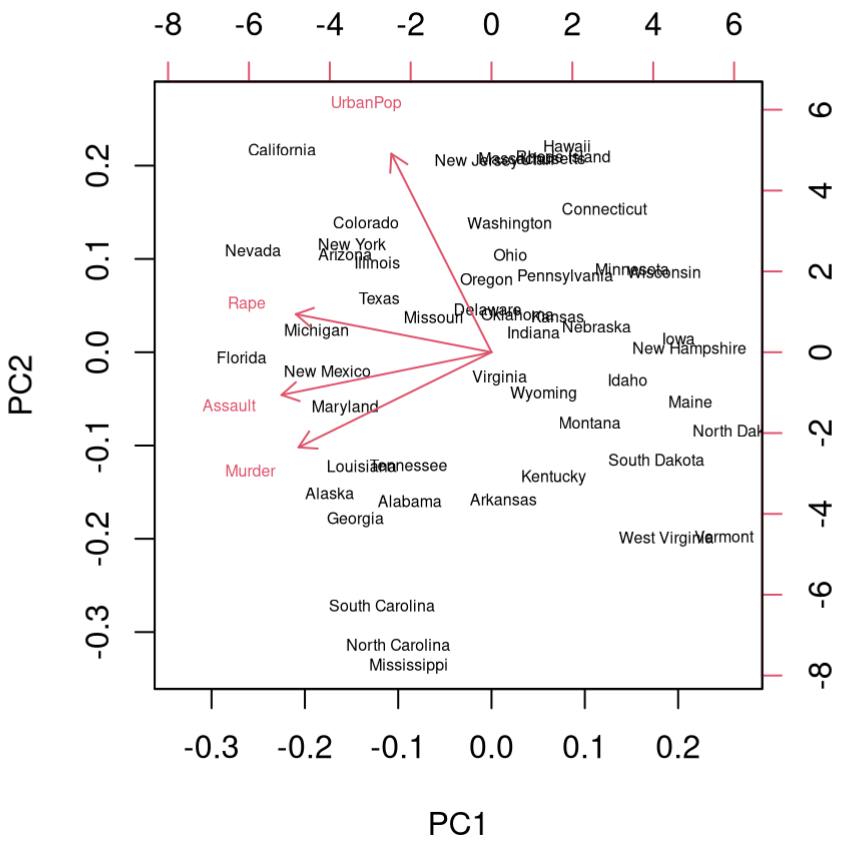
```
# Do PCA without scaling
usarrest.pca <- prcomp(USArrests)
usarrest.pca
```

```
## Standard deviations (1, .., p=4):
## [1] 83.732400 14.212402  6.489426  2.482790
##
## Rotation (n x k) = (4 x 4):
##                 PC1         PC2         PC3         PC4
## Murder   0.04170432 -0.04482166  0.07989066 -0.99492173
## Assault  0.99522128 -0.05876003 -0.06756974  0.03893830
## UrbanPop 0.04633575  0.97685748 -0.20054629 -0.05816914
## Rape     0.07515550  0.20071807  0.97408059  0.07232502
```

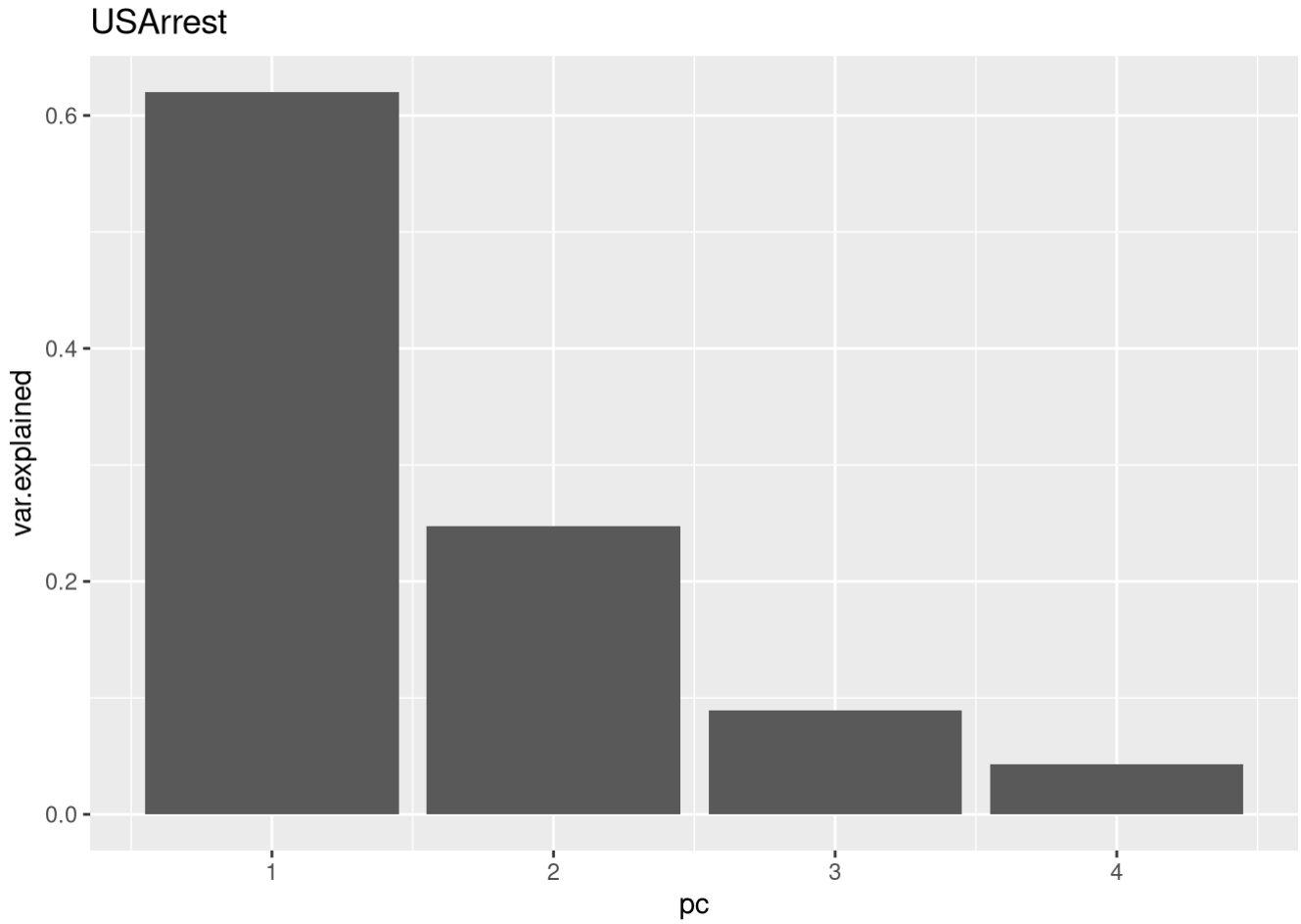
```
# Make a biplot
biplot(usarrest.pca, cex = 0.5)
```




```
# Now do pca again but scale the variable so they have mean of 0 and sd of 1
usarrest.pca.scale <- prcomp(USArrests, scale = TRUE)
biplot(usarrest.pca.scale, cex = 0.5)
```



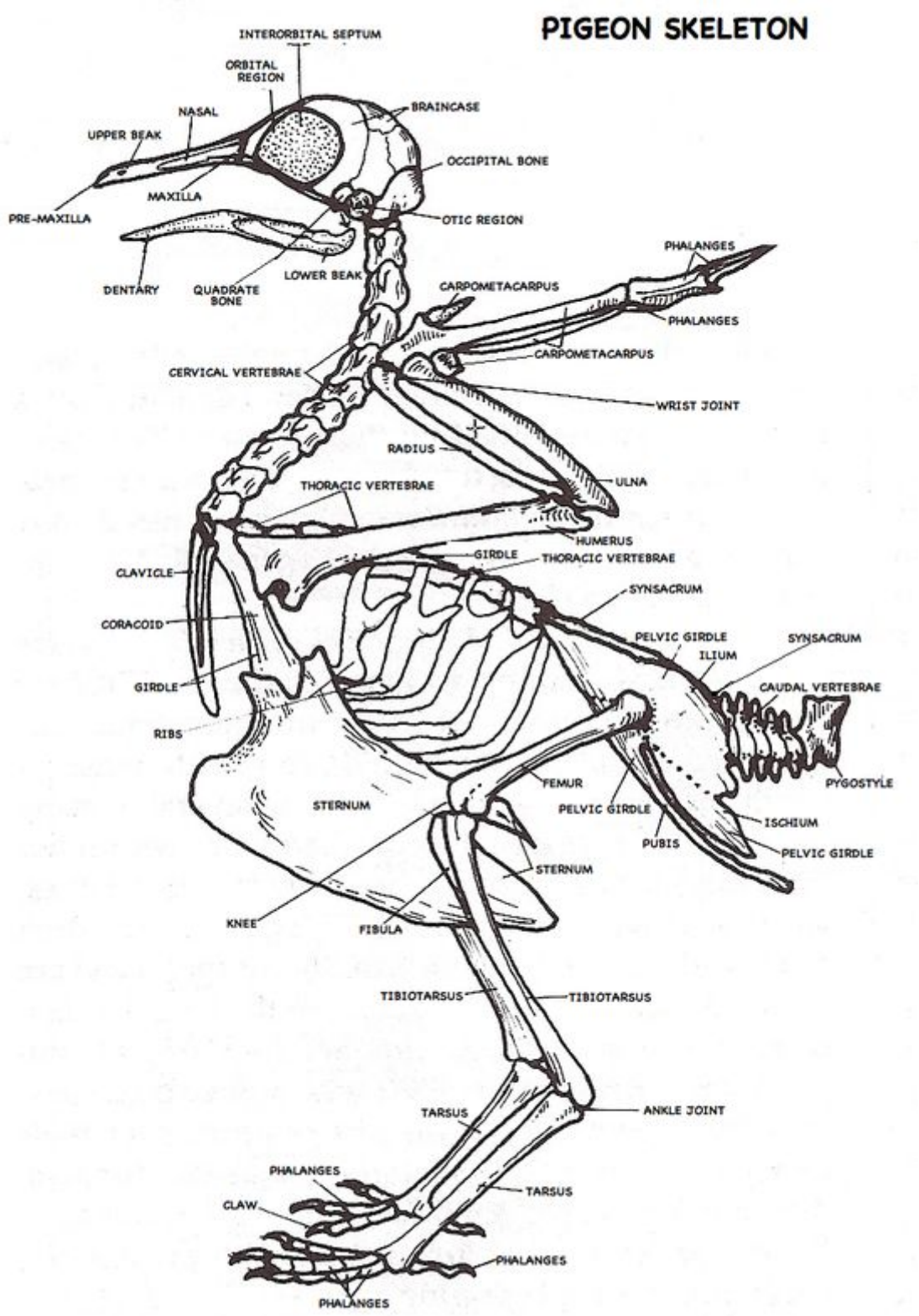

```
# Extract the variance explained
tot.var <- sum(usarrest.pca.scale$sdev^2)
var.explained <- data.frame(pc = seq(1:4), var.explained  = usarrest.pca.scale$sdev^2
/tot.var )
ggplot(var.explained, aes(pc, var.explained)) + geom_bar(stat = "identity") + ggtitle
("USArrest")
```



Birds dataset

The next dataset we will be exploring is a dataset that contains bone measures and living habits of birds provided by Dr. D. Liu of Beijing Museum of Natural History. The data contains ecological categories of the birds and measures of length and diameter of five bones from the bird's skeleton. The dataset was downloaded from Kaggle. There are 10 measurements of different bones (see image below) and the birds come from six ecological classes:

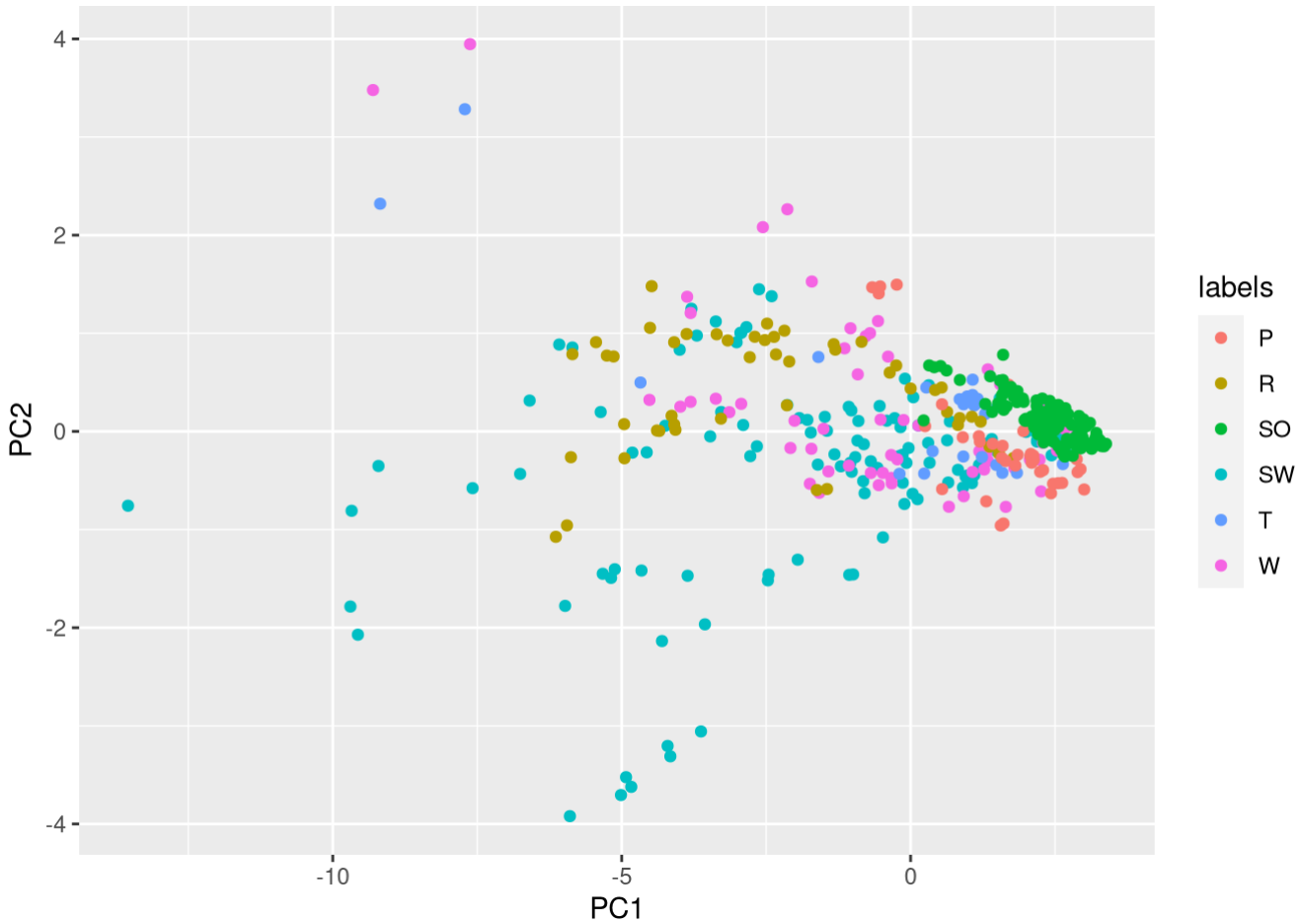
- Swimming Birds (S)
- Wading Birds (W)
- Terrestrial Birds (T)
- Raptors (R)
- Scansorial Birds (P)
- Singing Birds (SO)




```
birds <- read.csv("bird.csv", header = TRUE)
dim(birds)
```

```
## [1] 413  12
```

```
# PCA
# Do PCA with scaling
birds.pca <- prcomp(birds[,2:11], scale = TRUE)
birds.df <- data.frame(PC1 = birds.pca$x[,1], PC2 = birds.pca$x[,2], PC3 = birds.pca
$x[,3], labels = as.factor(birds$type))
ggplot(birds.df, aes(PC1, PC2, col = labels)) + geom_point()
```



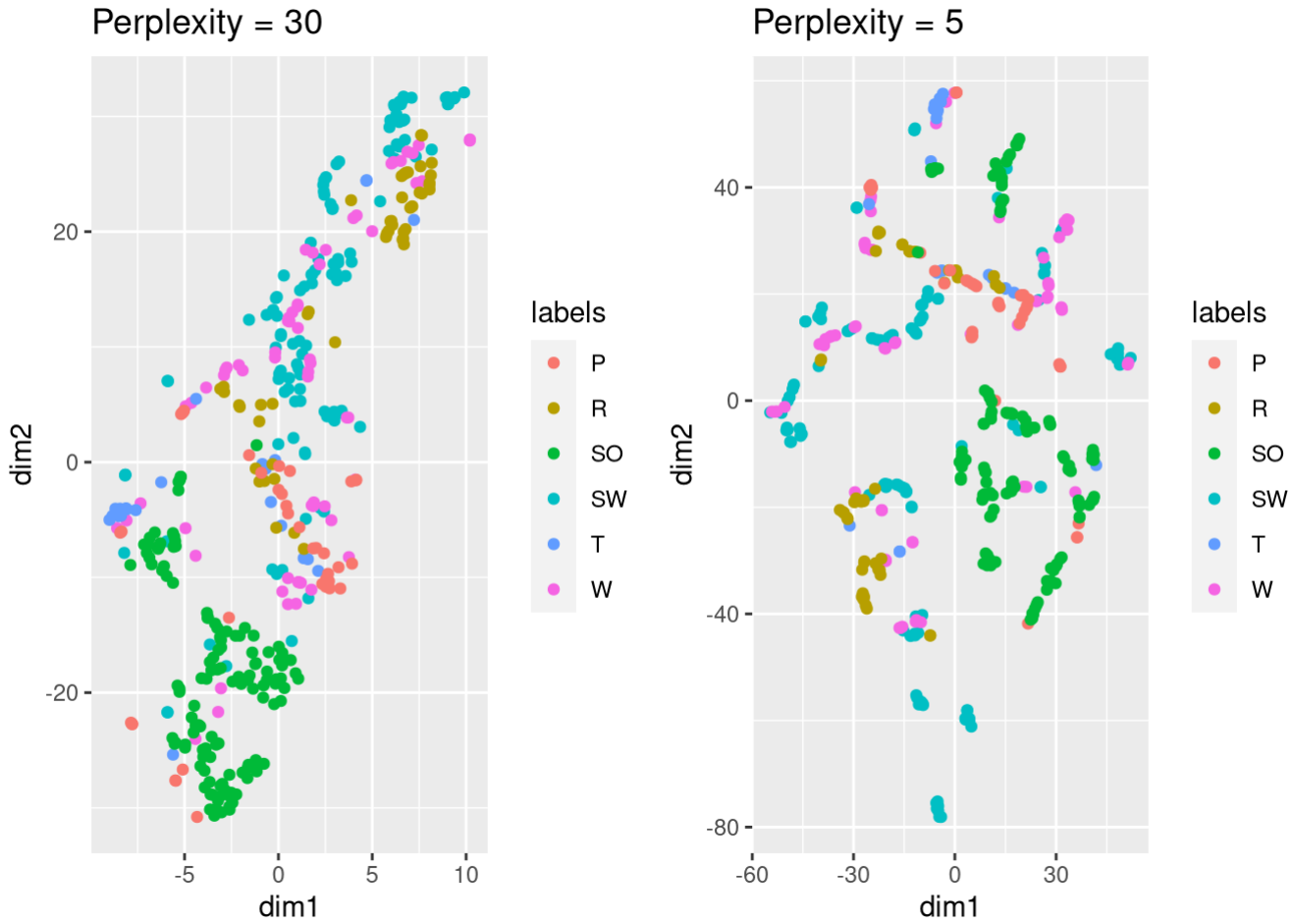
Let's now try to visualize the dataset with *t*-SNE. You can see if we make the perplexity parameter smaller, the size of the cluster appear to be smaller.

```
set.seed(5003)
birds.Rtsne <- Rtsne(birds[,2:11])
birds.rtsne.df <- data.frame(dim1 = birds.Rtsne$Y[,1], dim2 = birds.Rtsne$Y[,2], labe
ls = as.factor(birds$type))



birds.Rtsne.p5 <- Rtsne(birds[,2:11], perplexity = 5)
birds.rtsne.df.p5 <- data.frame(dim1 = birds.Rtsne.p5$Y[,1], dim2 = birds.Rtsne.p5$Y
[,2], labels = as.factor(birds$type))

p1 <- ggplot(birds.rtsne.df, aes(dim1, dim2, col = labels)) + geom_point() + ggtitle(
"Perplexity = 30")
p2 <- ggplot(birds.rtsne.df.p5, aes(dim1, dim2, col = labels)) + geom_point() + ggtit
le("Perplexity = 5")
grid.arrange(p1, p2, ncol = 2)
```



PCA and kmeans

We can use PCA as a preprocessing step. Here, we first run PCA on the birds dataset, extract the first 2 principal components and use them downstream to perform kmeans clustering.

```
set.seed(5003)
birds.km <- kmeans(birds[,2:11], centers = 6)
table(birds.km$cluster, birds$type)
```

```
##
##        P   R  SO  SW   T   W
##    1   0  15   0  24   1   7
##    2  20   1 102   9   4  12
##    3   0   0   0  14   0   0
##    4  18   8  22  12  15  13
##    5   0  14   0  19   2  10
##    6   0  10   0  38   1  22
```

```
birds.pca2.km <- kmeans(birds.pca$x[,1:2], centers = 6)
birds.pca3.km <- kmeans(birds.pca$x[,1:3], centers = 6)

birds.rtsne.df$kmeans2 <- as.factor(birds.pca2.km$cluster)
birds.rtsne.df$kmeans3 <- as.factor(birds.pca3.km$cluster)
birds.rtsne.df$kmeansall <- as.factor(birds.km$cluster)

# Let's make the plots again on the tSNE space
p1 <- ggplot(birds.rtsne.df, aes(dim1, dim2, col = labels)) + geom_point() + ggtitle(
"Original label")
p2 <- ggplot(birds.rtsne.df, aes(dim1, dim2, col = kmeans2)) + geom_point() + ggtitle
("Kmeans label (2 PC)")
p3 <- ggplot(birds.rtsne.df, aes(dim1, dim2, col = kmeans3)) + geom_point() + ggtitle
("Kmeans label (3 PC)")
p4 <- ggplot(birds.rtsne.df, aes(dim1, dim2, col = kmeansall)) + geom_point() + ggtit
le("Kmeans label (all)")
grid.arrange(p1, p2, p3, p4, ncol = 2)
```