K-means Clustering

Matt Richey

04/23/2020

Adding cluster to a PCA plot

The usual libraries. You will probabilty need to install the factoextra package.

```
suppressMessages(library(tidyverse))
suppressMessages(library(MASS))
suppressMessages(library(factoextra))
```

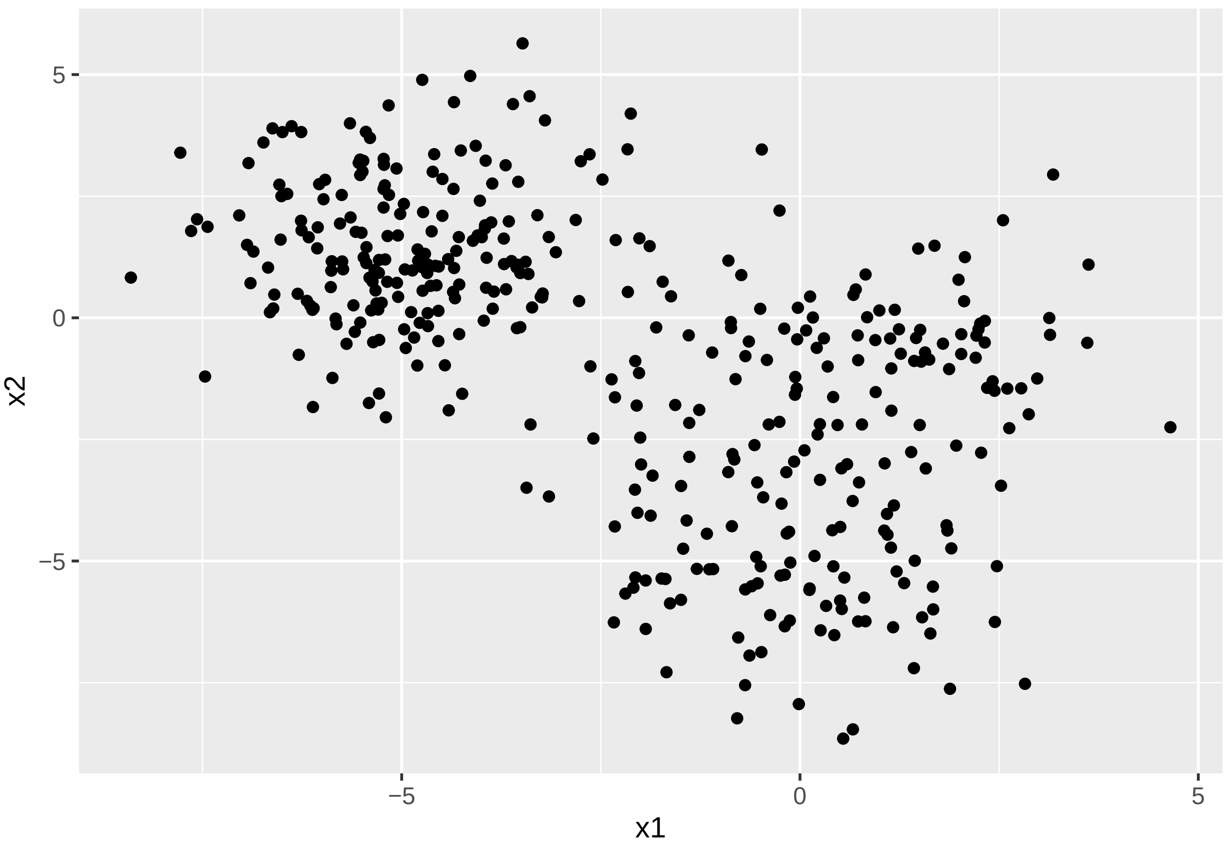
Data with more than 2 dimensions.

Create a data frame with 4 dimensions and

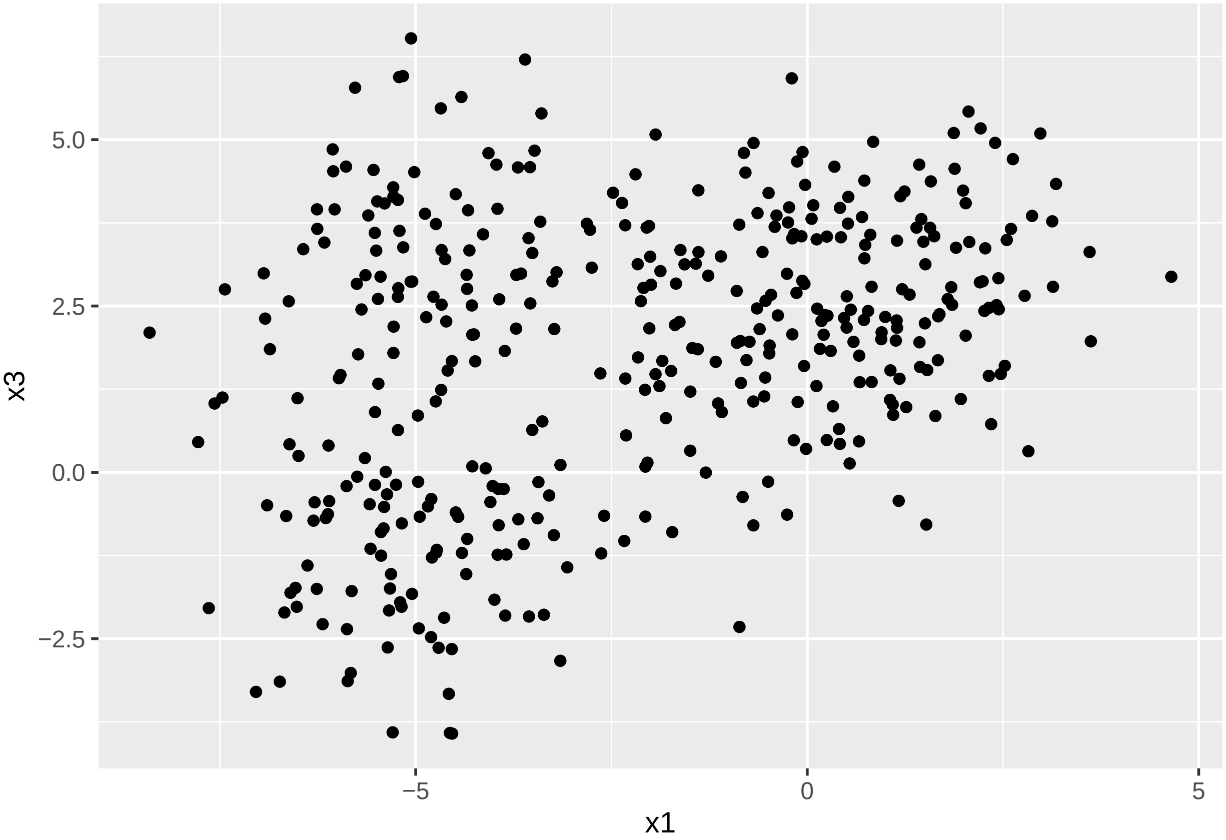
```
N <- 100
K <- 4
mu <- sample(-5:5,5*K,rep=T)
sd0 <- 2
dat <- c()
for(k in 1:K){
    dat <- c(dat,c(mvrnorm(N,c(mu[2*k-1],mu[2*k]),diag(c(1,1)*sd0)),
                   mvrnorm(N,c(mu[2*k+1],mu[2*k+2]),diag(c(1,1)*sd0))))
}
dat <- matrix(dat,byrow=F,ncol=K)
data.df <-
    data.frame(x1=dat[,1],
               x2=dat[,2],
               x3=dat[,3],
               x4=dat[,4])
```

Since we are in four dimensions, there is no simple visualization available. You could try some pairwise plots.

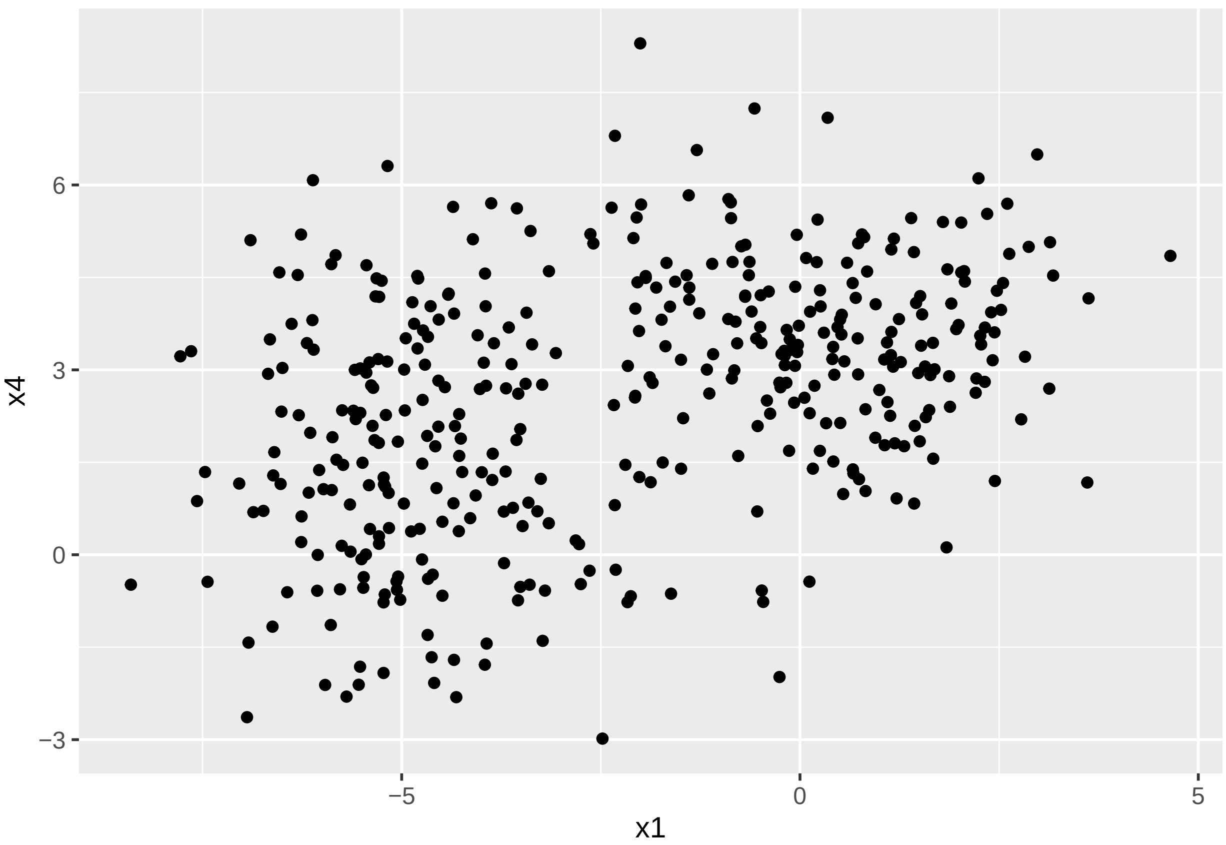
```
ggplot(data.df,aes(x1,x2))+geom_point()
```



```
ggplot(data.df,aes(x1,x3))+geom_point()
```



```
ggplot(data.df,aes(x1,x4))+geom_point()
```



Not too helpful.

However, we can still cluster. Just to be safe, scale the data and repack into data frame.

```
data.df <- scale(data.df)
data.df <- data.frame(data.df)
```

Apply kmeans with, say, K=5 means.

```
K<-5
mod.km <- kmeans(data.df,K,nstart=25)
data.df$cluster <- factor(mod.km$cluster)
```

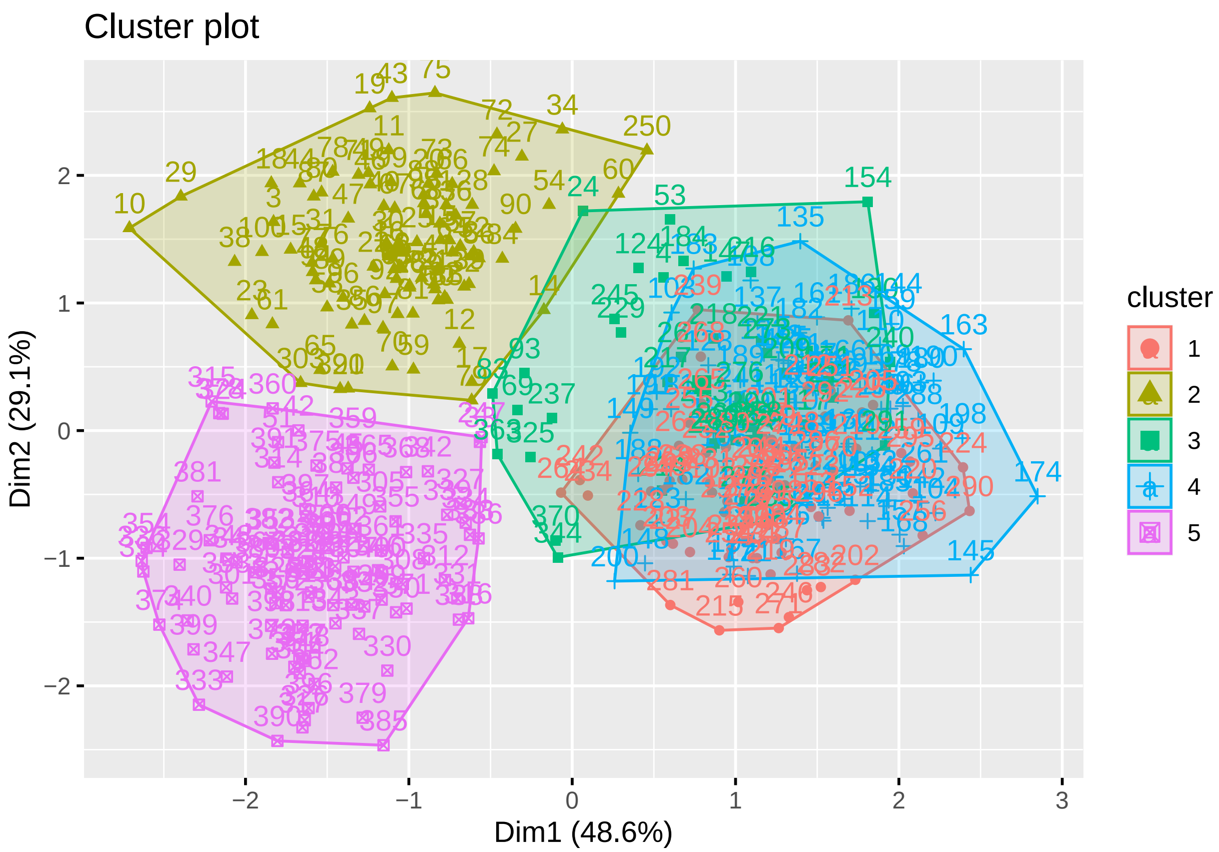
Ok..what do we do now, we have a clustering, but how does it look?

Here's the plan: Perform a Principal Component Analysis and project into 2-dimensional space. Carry the clusters along with the projection and see what we have.

The fivz_cluster function will do this.

I.e, fviz_cluster will project onto the "best" two dimensions. This is essentially the biplot with clustering information included.

```
## make sure we only use the original data!
fviz_cluster(mod.km,data=data.df[,1:4])
```



The boundaries are the "convex hulls" around each cluster. These are added to help visualize the clustering.

Note: We can build this ourselves (except the convex hulls).

Solution: Build it yourself

Cluster with k=4

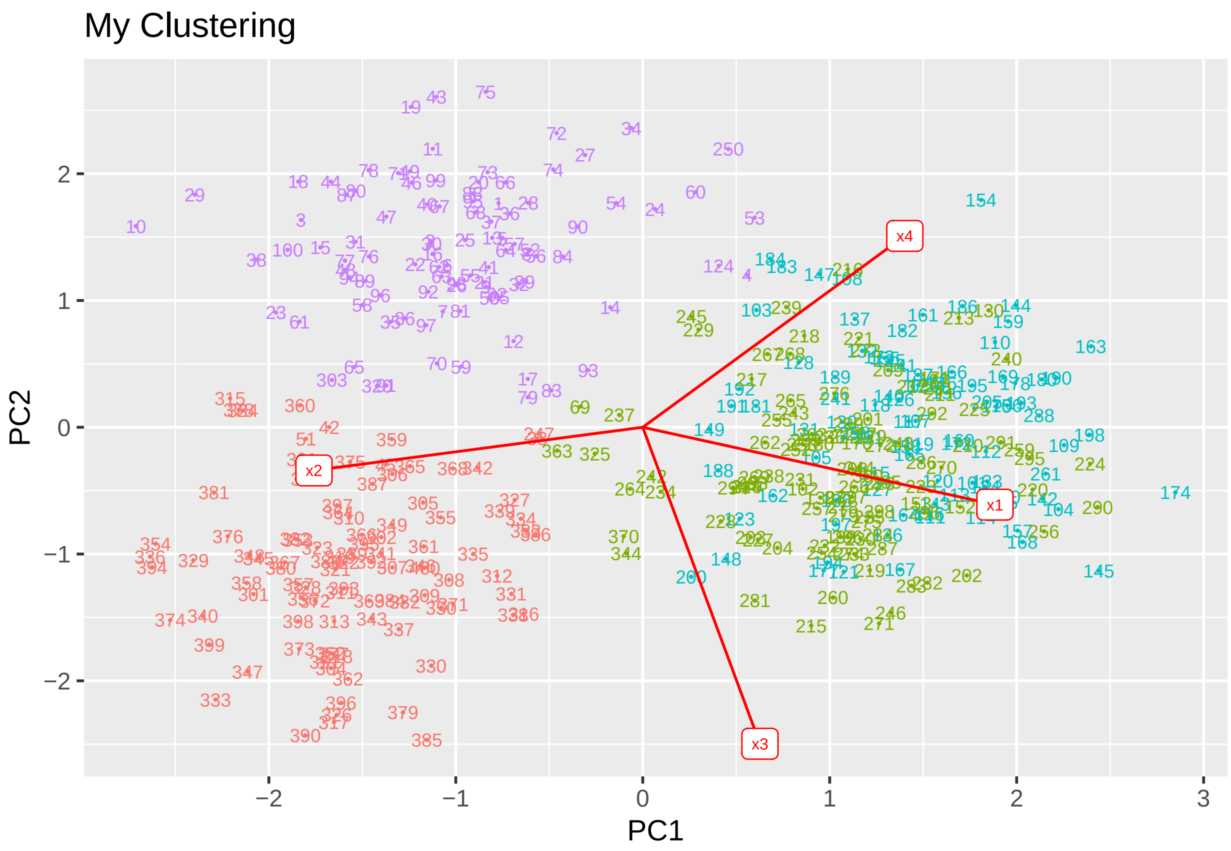
```
data.mat <- data.matrix(data.df[,1:4])
mod.km <- kmeans(data.mat,centers=4,nstart=10)
table(mod.km$cluster)
```

```
##
##   1   2   3   4
##  99 108  94  99
```

```
mod.pr <- prcomp(data.mat)
## Pull off the rotation matrix.
rot.mat <- mod.pr$rotation
## Rotate the data
dataRotate.mat <-  data.mat %*% rot.mat
dataRotate.df <- data.frame(dataRotate.mat)
dataRotate.df$cluster <- factor(mod.km$cluster)
dataRotate.df$id <- 1:nrow(data.df)
loading.df <- data.frame(rot.mat)
loading.df$pred <- rownames(rot.mat)
```

Now create the plot, include the loading vectors as well.

```
load.scale <- 3
dataRotate.df%>%
  ggplot()+
  geom_point(aes(PC1,PC2,color=cluster),size=.1)+
  geom_text(aes(PC1,PC2,color=cluster,label=id),
                  size=2.5)+
  geom_segment(data=loading.df,
               aes(x=0,xend=load.scale*PC1,
                   y=0,yend=load.scale*PC2),
               color="red")+
   geom_label(data=loading.df,
              aes(load.scale*PC1,
                  load.scale*PC2,
                  label=pred),
              color="red", size=2)+
  guides(color=F)+
    ggtitle("My Clustering")
```



The data are complete random so there isn't much to say about the loading directions.