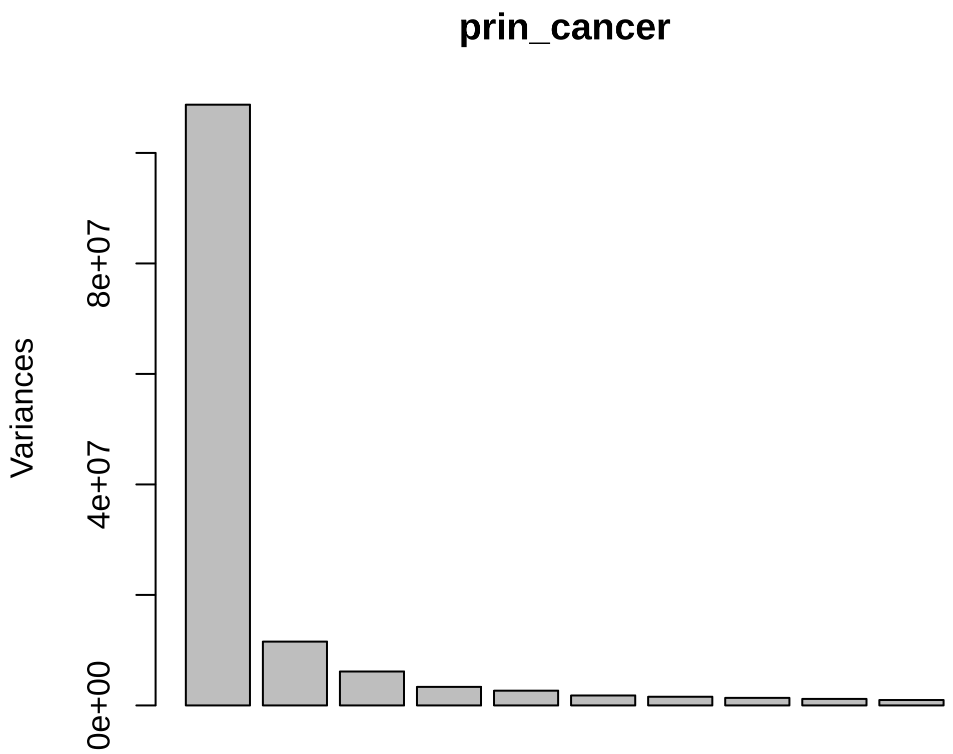
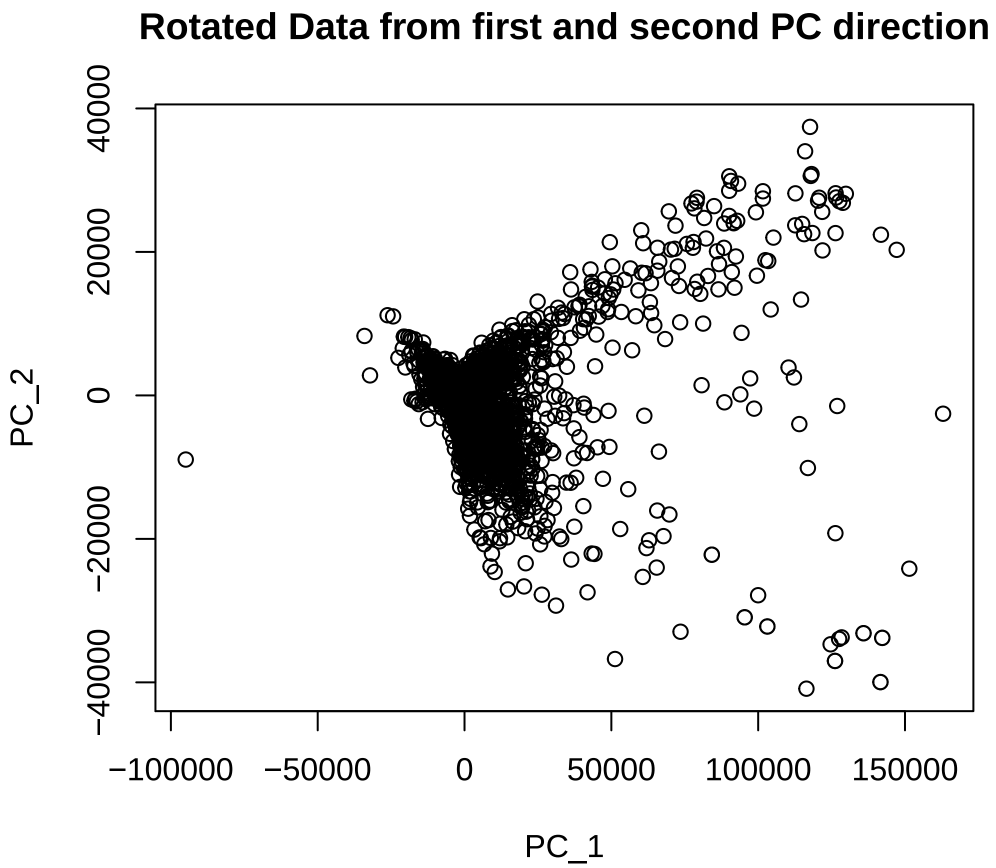
## Problem 2

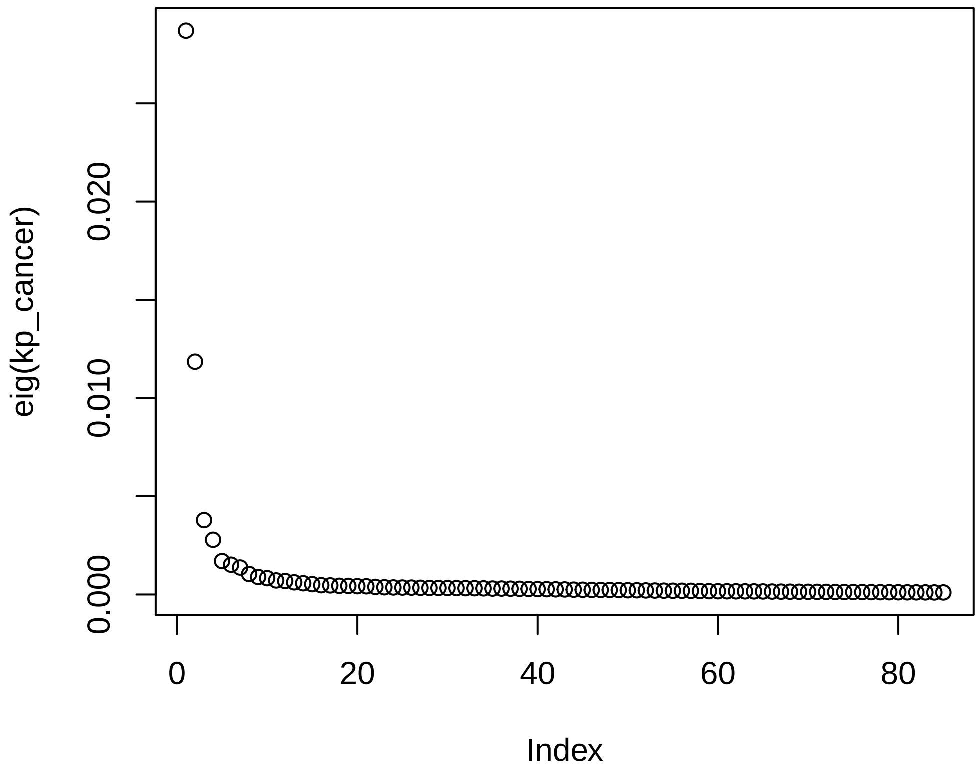
```
# 1). PCA and Kernel PCA to the data
cancerdata <- read.table("Cancer.txt", header = FALSE)
names(cancerdata) <- NULL
cancerdata <- as.matrix(cancerdata)
# Due to the computation limitation, I will choose use all of
# the data to do PCA and use part of data to do kernel PCA.
prin_cancer <- prcomp(cancerdata, rtex = TRUE)
# Check the screeplot of PCA result, it seems that the first
# principal components contain a large portion of
# information, which can be used to compress and cluster the
# data
screeplot(prin_cancer)
```



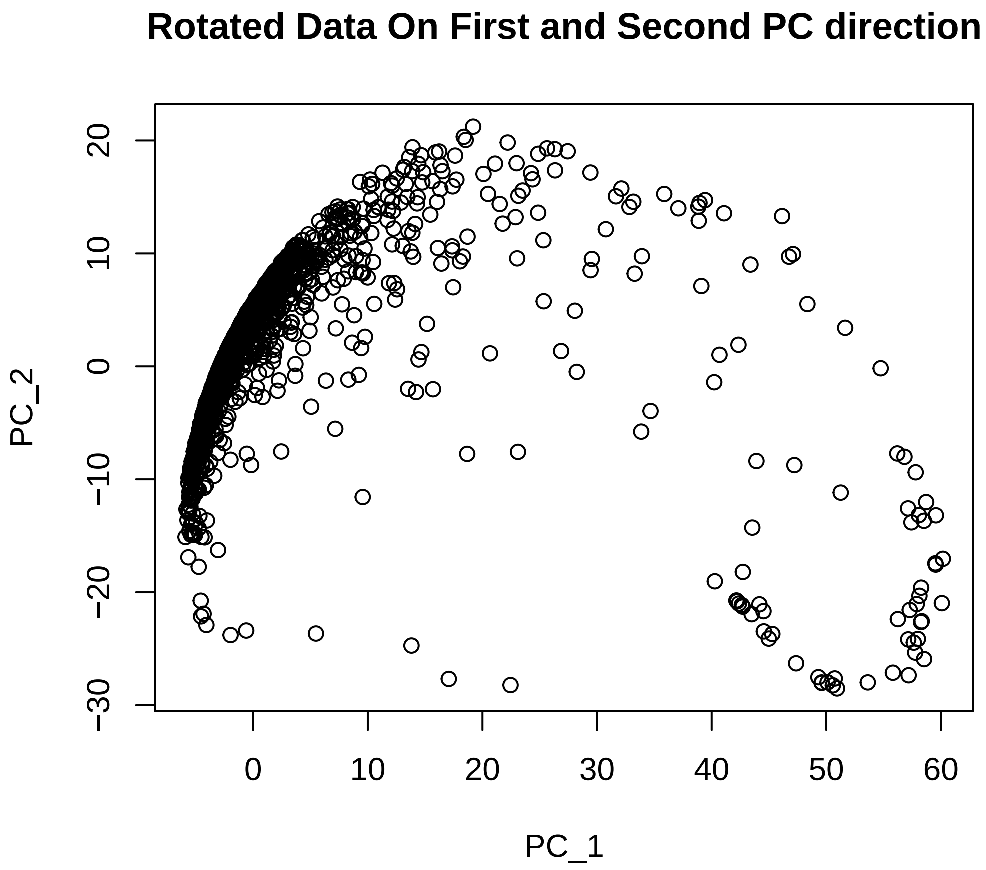
```
# Then I can plot the projection on the first and second PC
# loadings, and we can see an obvious clustering of a large
# portion of data. But we need further investigation in order
# to find out if we can cluster those points into 14 groups
# clearly.
plot(prin_cancer$x[, 1], prin_cancer$x[, 2], main = "Rotated Data from first and second PC direction",
    xlab = "PC_1", ylab = "PC_2")
```



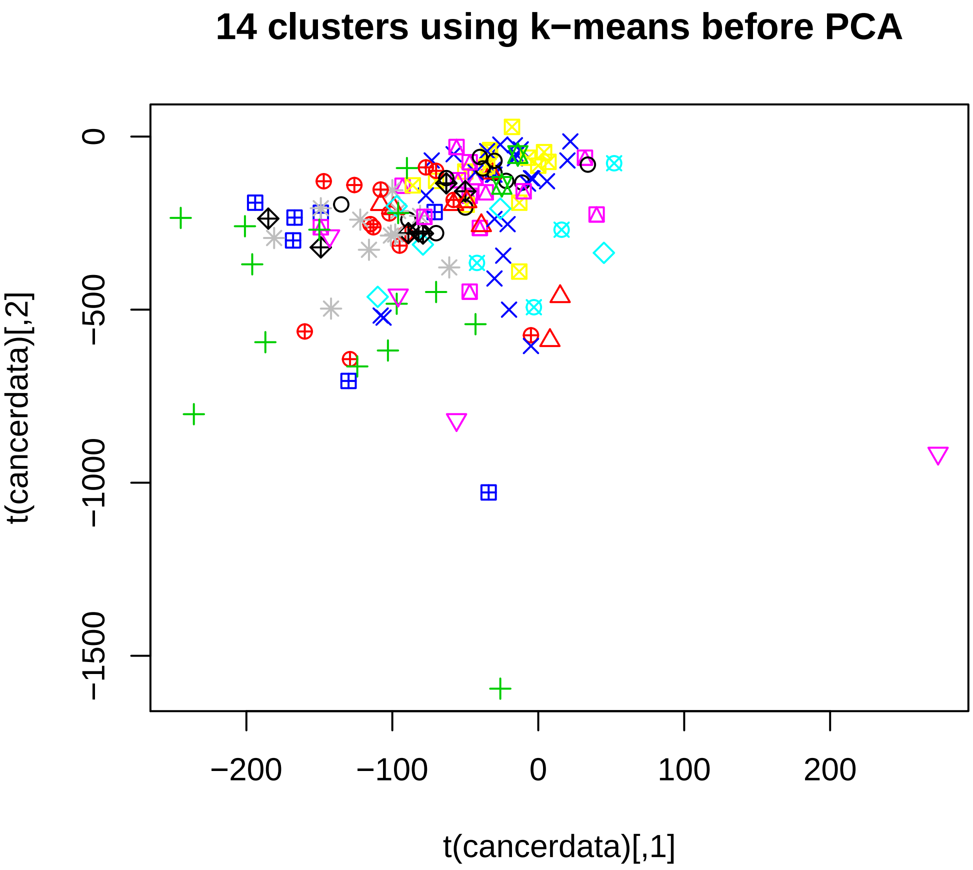
```
# To further illustrate, I can check the variance
# contribution of first several principal components.
explain_var_cancer <- sapply(1:144, function(i) sum(prin_cancer$sdev[1:i]^2)/sum(prin_cancer$sdev^2))
head(explain_var_cancer)  # The first PC contains almost 70% of information.

## [1] 0.6939 0.7676 0.8068 0.8282 0.8453 0.8568

###### Kernel PCA with Gaussian kernel function ######
library(kernlab)
rbf <- rbfdot(sigma = 0.001)
kern_cancer <- kernelMatrix(rbf, scale(cancerdata[1:3000, ]))  #Part of the data due to computation lim
kp_cancer <- kpca(kern_cancer)
# We can check the screeplot of the kernel PCA outcome, it
# seems we have a really good scree plot and the first and
# second principal component to project the data.
plot(eig(kp_cancer))
```



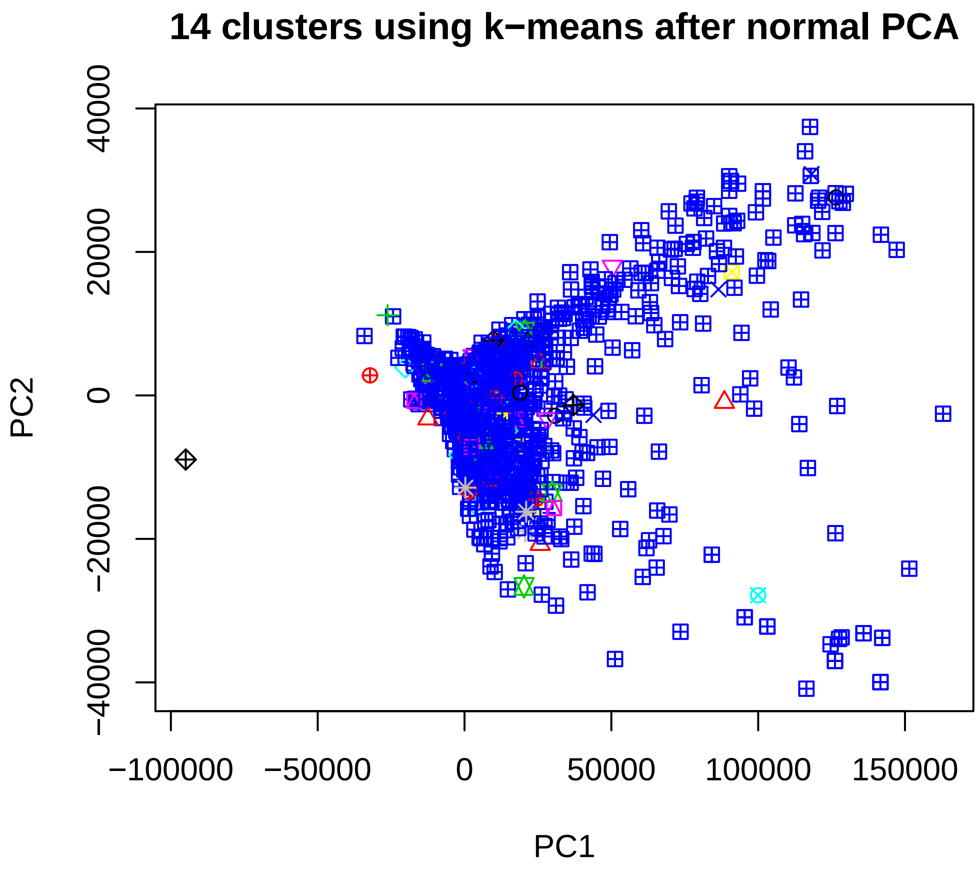
```
plot(rotated(kp_cancer)[, 1], rotated(kp_cancer)[, 2], main = "Rotated Data On First and Second PC direc
    xlab = "PC_1", ylab = "PC_2")
```



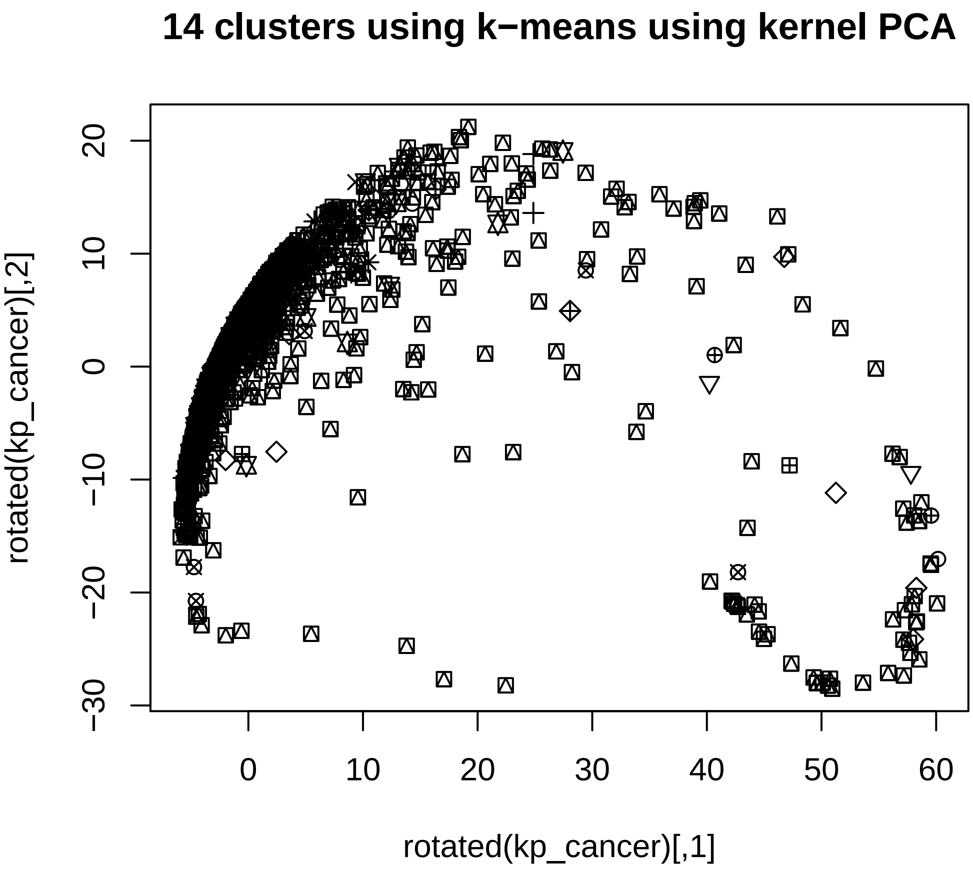
```
# Comment: we can observe that there are two groups of points
# emerging, and we would expect to have more groups if we use
# all of the data.
# 2).  K-means ###########
kmean <- kmeans(t(cancerdata), 14, nstart = 1)
plot(t(cancerdata), col = kmean$cluster, pch = kmean$cluster,
    main = "14 clusters using k-means before PCA")
```



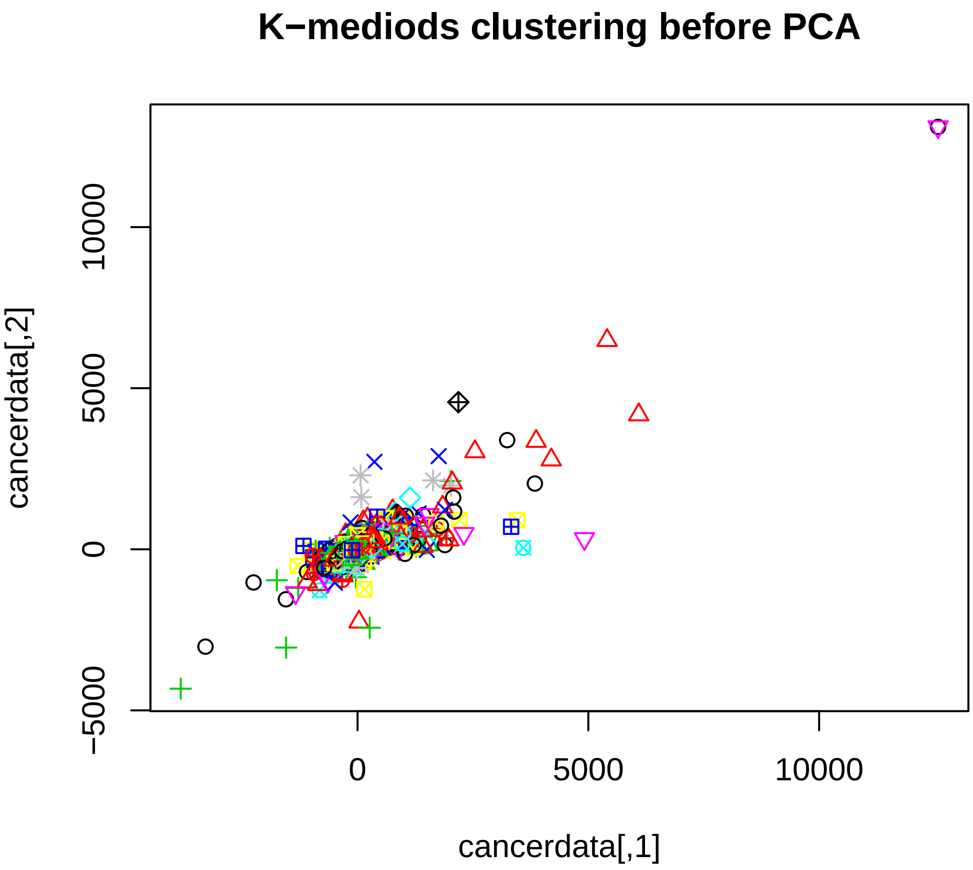
```
# Comment: K-means does not help us to clearly cluster the
# datapoints, and we need to figure out better clustering.
# I use the projected data in the first and second PCs and I
# expect to have a better clustering outcome.
kmeans_PCA <- kmeans(t(prin_cancer$x), 14, nstart = 1)
plot(prin_cancer$x, col = kmeans_PCA$cluster, pch = kmeans_PCA$cluster,
    main = "14 clusters using k-means after normal PCA")
```



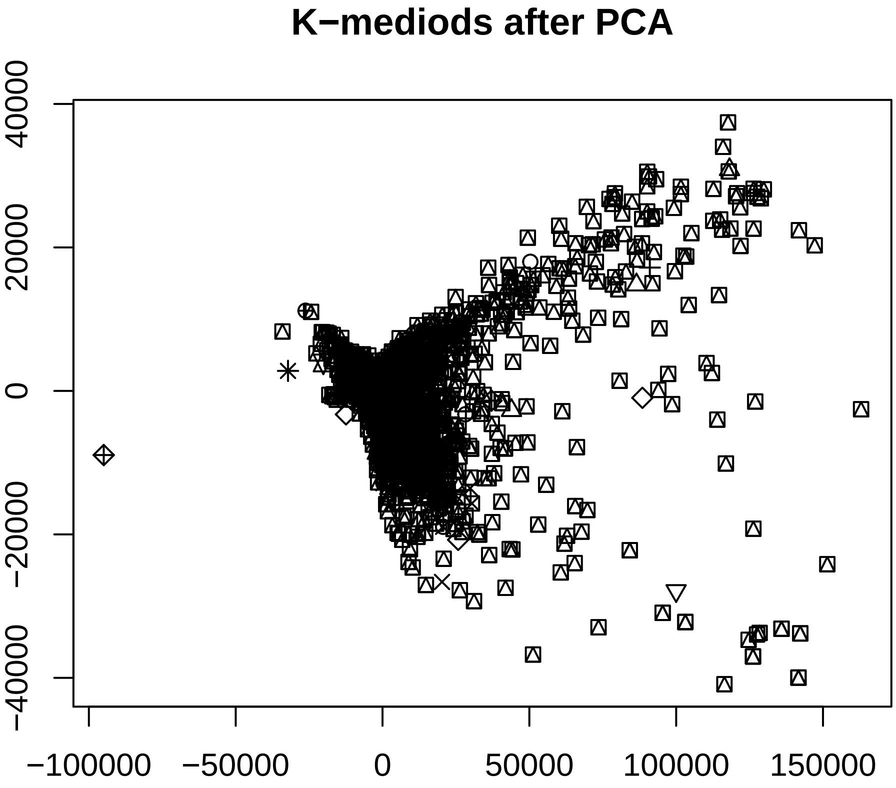
```
# Comment: we have a majority of points being classfied in
# one group, it means that PCA does not identify them into
# different groups, and they have similar variance in most of
# PC loading directions.
kmeans_kpca <- kmeans(t(rotated(kp_cancer)), 14, nstart = 1)
plot(rotated(kp_cancer), pch = kmeans_kpca$cluster, main = "14 clusters using k-means using kernel PCA")
```



```
# Comment: Obviously, the k-means method is biased by an
# outlier.
####### K-medoids #########
library(cluster)
diss <- dist(t(cancerdata))
kmedoids <- pam(diss, 14, do.swap = FALSE)
plot(cancerdata, col = kmedoids$cluster, pch = kmedoids$cluster,
    main = "K-mediods clustering before PCA")
```

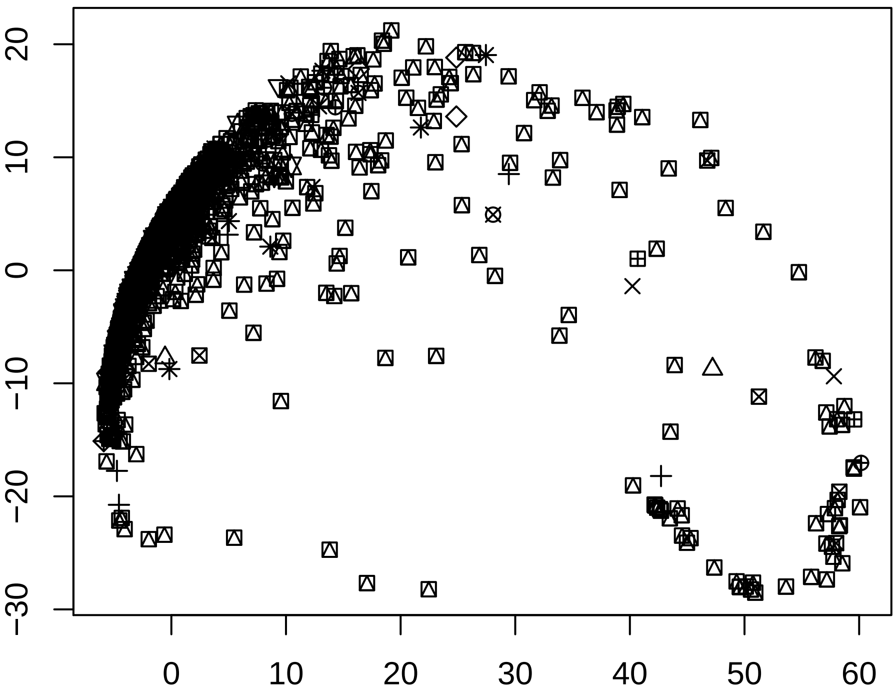


```
# Comment: It is not a good clustering since the points are
# not spead out and grouped clearly.
diss_pca <- dist(t(prin_cancer$x))
kmedoids_pca <- pam(diss_pca, 14, do.swap = FALSE)
plot(prin_cancer$x, pch = kmedoids_pca$cluster, xlab = "", ylab = "",
    main = "K-mediods after PCA")
```



```
# Comment: Still not good, I don't think PCA is helpful in
# this case or we need to figure out a combination of PCs to
# reflect clustering.
diss_kpca <- dist(t(rotated(kp_cancer)))
kmedoids_kpca <- pam(diss_kpca, 14, do.swap = FALSE)
plot(rotated(kp_cancer), pch = kmedoids_kpca$cluster, xlab = "",
    ylab = "", main = "K-mediods after kernel PCA")
```

## K−mediods after kernel PCA



```
# Comment: Probably it is due to the lack of useful genes or
# it is due to the fact that the principal component needs to
# be further revised in this case, we cannot cluter out the
# points.
```