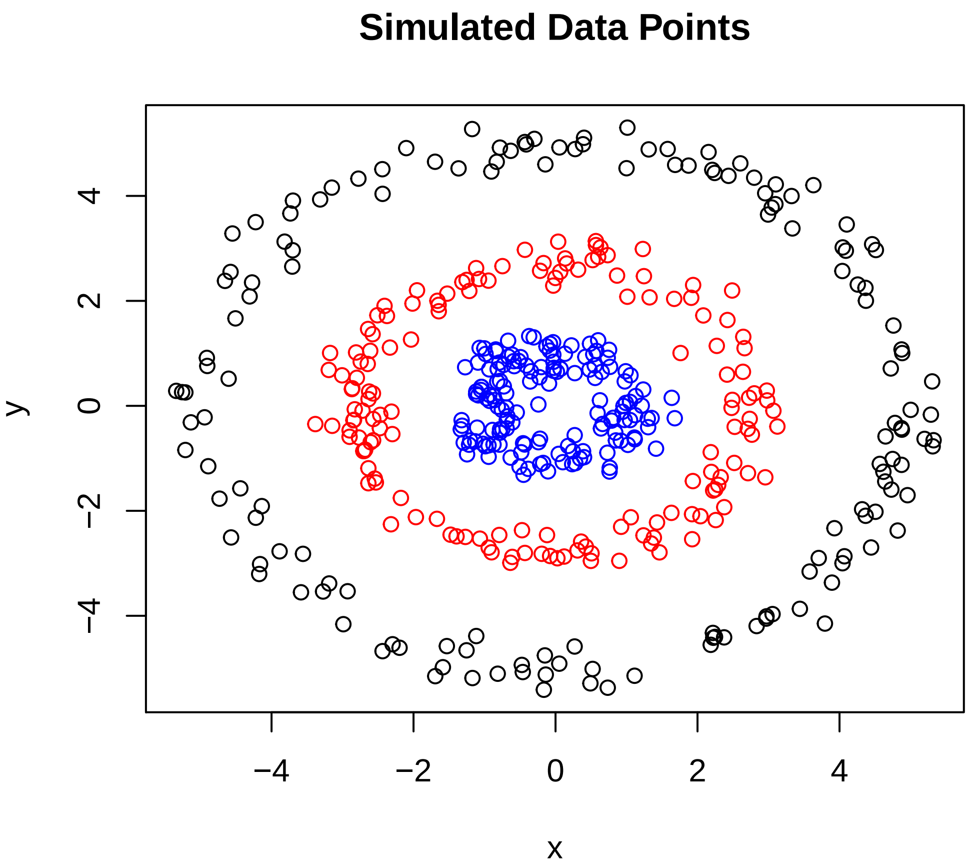
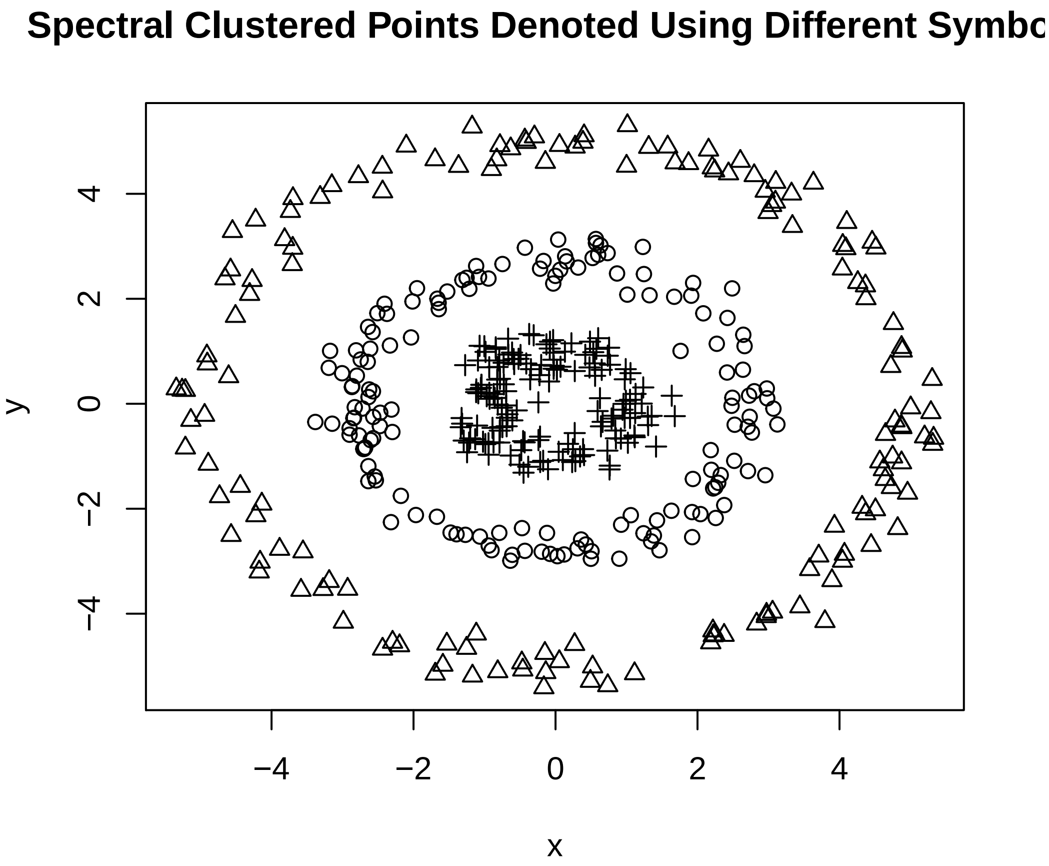
Problem 3

```
# We constructed a function called rgene to generate the of a circle with
# radius r
set.seed(0)
rgene <- function(r) {
    theta <- runif(150, min = -pi, max = pi)
    x <- r * cos(theta) + rnorm(150, mean = 0, sd = 0.25)
    y <- r * sin(theta) + rnorm(150, mean = 0, sd = 0.25)
    return(cbind(x = x, y = y))
}
X <- rgene(5)
Y <- rgene(2.8)
Z <- rgene(1)
# Draw the plot of graph
plot(X, main = "Simulated Data Points")
points(Y, col = "red")
points(Z, col = "blue")
```

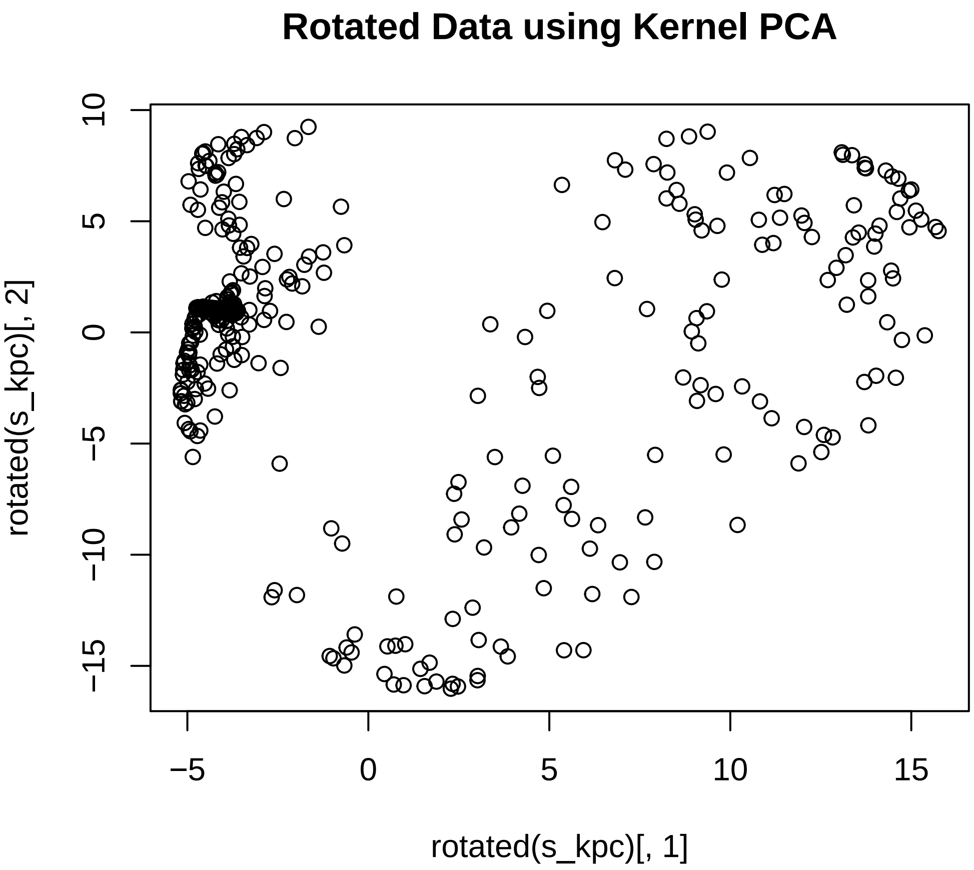


```
# Spectral Clustering
library(kernlab)
s_data <- rbind(X, Y, Z)
```

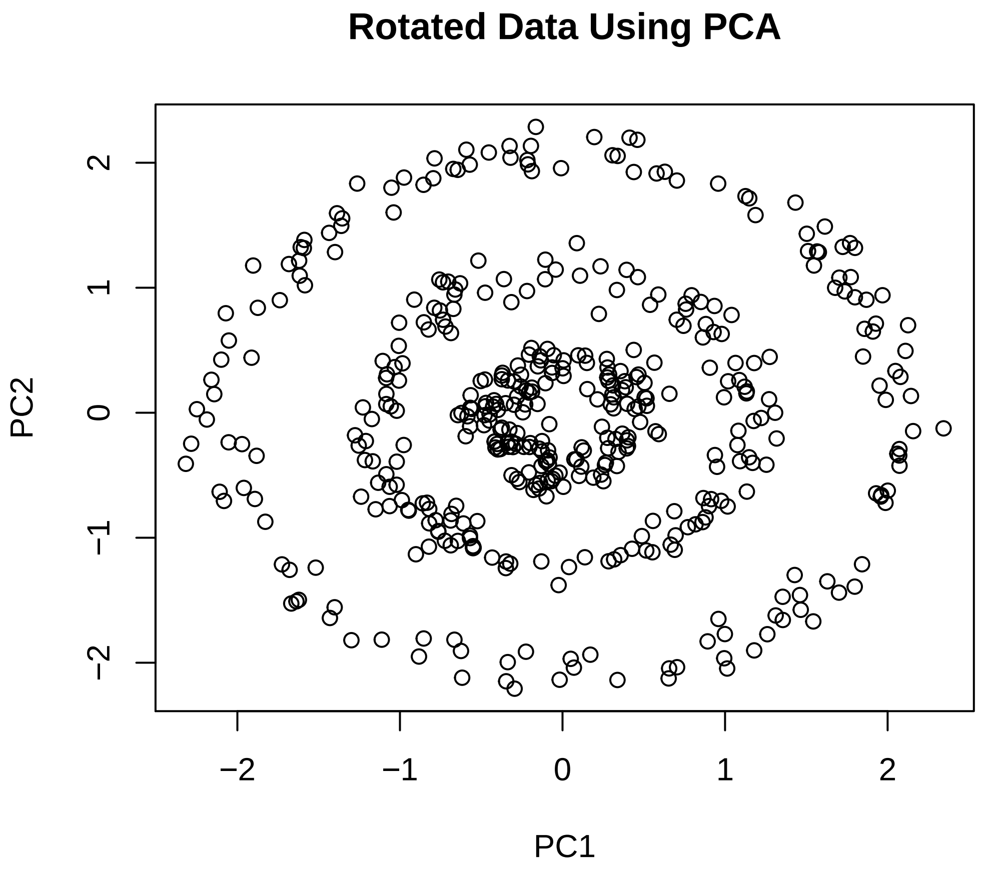
```r
spec_cl <- specc(s_data, centers = 3)
# It is a really accurate clustering
plot(s_data, pch = spec_cl, main = "Spectral Clustered Points Denoted Using Different Symbols")
```



```r
# Kernel PCA
s_kpc <- kpca(s_data, kernel = "rbfdot", list(sigma = 0.7))
plot(rotated(s_kpc)[, 1], rotated(s_kpc)[, 2], main = "Rotated Data using Kernel PCA")
```



```
# Comment: Using kernel PCA, we can see that there are three parts of
# points that can be grouped.
# PCA
s_pc <- prcomp(s_data, rtex = TRUE, scale = TRUE)
plot(s_pc$x, main = "Rotated Data Using PCA")
```



```
# Comment: Obviously, PCA cannot help us to cluster the data, which is
# reasonable since what we have done is only rotating the coordinates;
# however, it does not change anything since the data spread like
# circles.
```