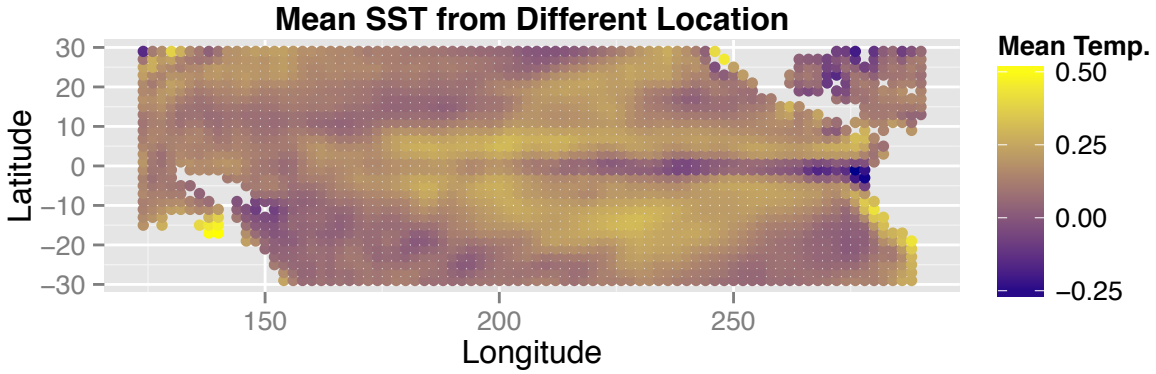
STAT 8330 FALL 2015 ASSIGNMENT 8

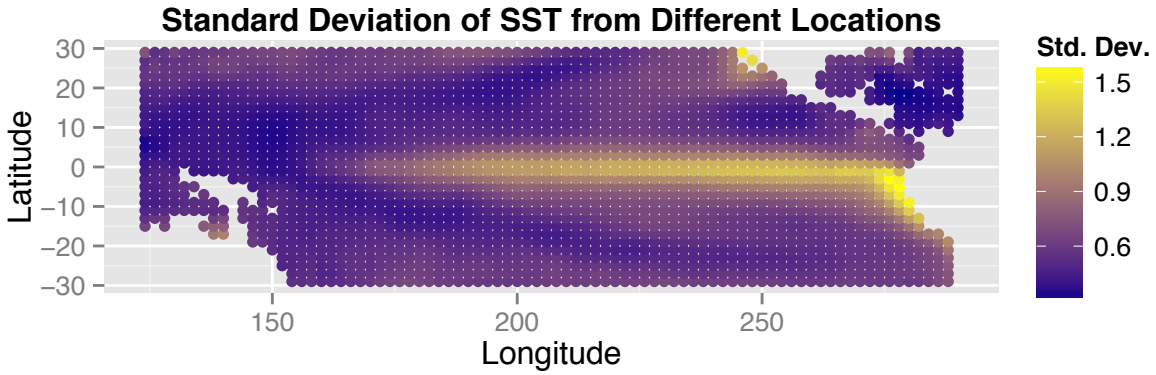
Peng Shao

November 24, 2015

▶ 1. Solution.

From the two plots, the mean temperatures of the area does not show so much differerces. The only abnormal phenomenon is that the the loacations which have relatively low temperature are around the equator and eastern Pacific Ocean, where people usually think the weather might be hot. Moreover, these locations also have very high variation, while the temperatures of other locations of Pacific Ocean are much more stable.





▶ 2. Solution.

The variance accounted for by the first 4 principal components when the data are not standardized and when they are standardized are

```
pr.out.nonsd.var[1:4]
```

```
## [1] 363.23203  86.83395  79.65743  39.02647
```

```
pr.out.sd.var[1:4]
```

```
## [1] 600.06905 288.08273 177.82343  89.85436
```

And the percentages of them are

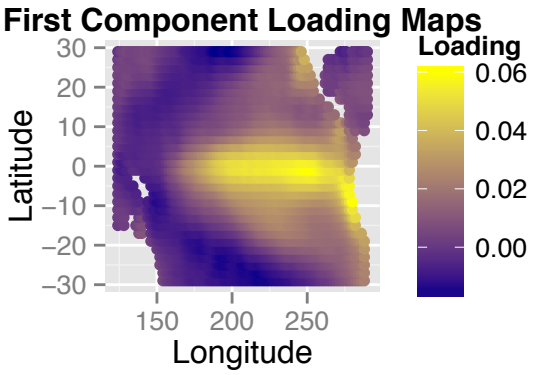
```
pr.out.nonsd.var[1:4]/sum(pr.out.nonsd.var)
```

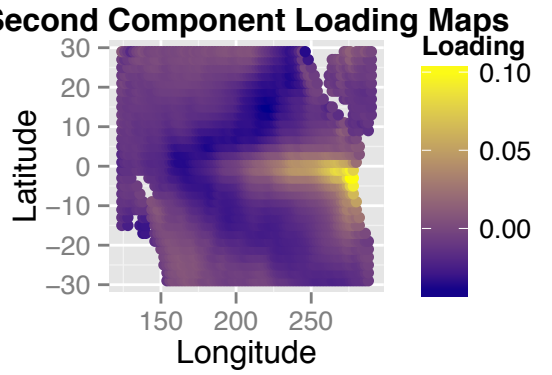
```
## [1] 0.38814831 0.09279042 0.08512162 0.04170353
```

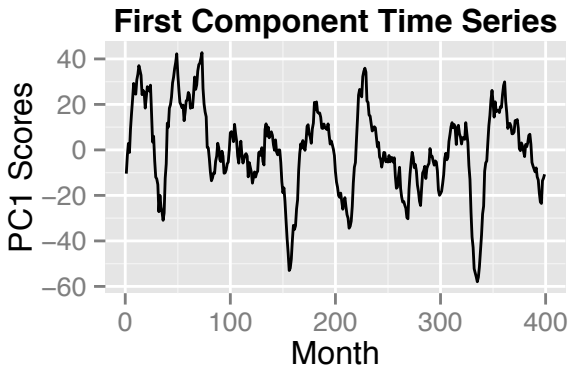
```
pr.out.sd.var[1:4]/sum(pr.out.sd.var)
```

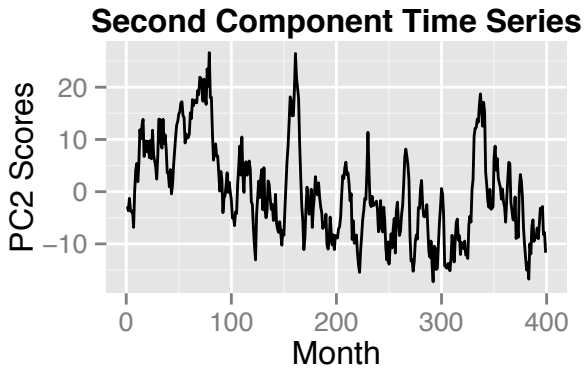
```
## [1] 0.26539985 0.12741386 0.07864813 0.03974098
```

Both map plots show that the locations around the equator and eastern Pacific Ocean have high loading, which means that they play very important roles in the first principle component and the second principle component. This is in accordance with the result in question 1. And from the time series plots, there are several unusual low temperatures in the trend of PC1 and several unusual high temperatures in the trend of PC2. To sum up, based on PCA, the first two key features which enfluence the "Tropical Pacific Sea Surface Temperature (SST)", are the unusual low temperatures and usual high temperatures of the area between 180 W ~ 90 W, 5 S ~ 5 N(about Nino Region 3), which may be the well-known La Niña and El Niño.



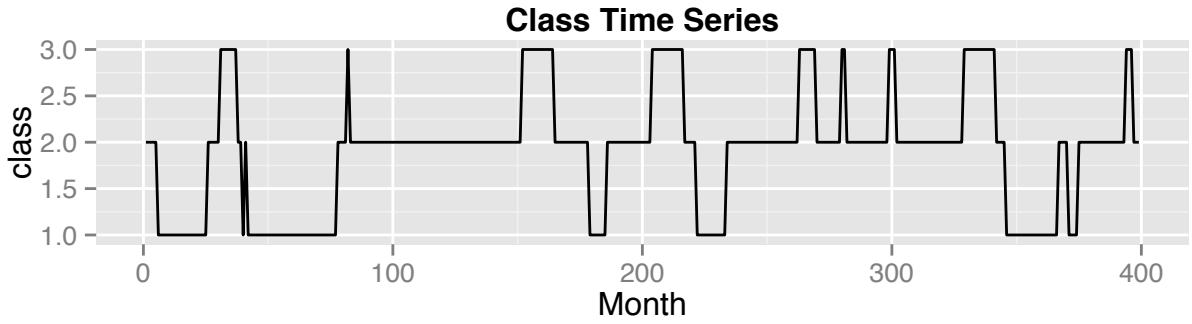


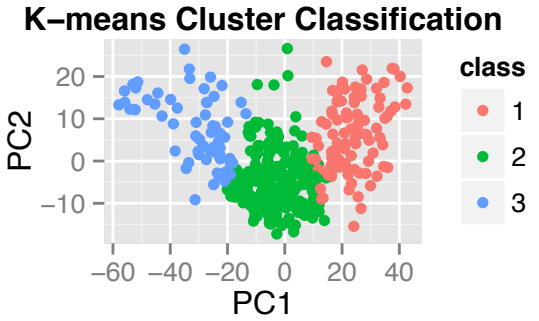


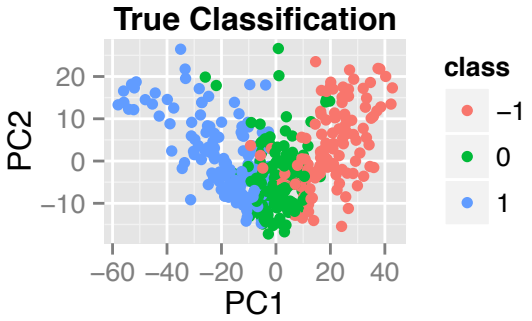


▶ 3. Solution.

To be clear, I use the No.3 random seed. The reason of my choice is the classification results of K-means cluster can be the same order with the Nion 3.4 Index, which means class 1 v.s La Niña, class 2 v.s Normal, and class 3 v.s El Niño.







From the classification plots, we can see that most data are correctly classified, so the K-means cluster based on PCA is fairly good. To see the performance more precisely, we can create the confusion matrix.

```
confusionMatrix(km.out$cluster-2, indexes[, 1])$table
```

```
##            Reference
## Prediction  -1   0   1
##         -1  95   6   0
##         0   35 128  73
##         1    0   2  60
```

```
confusionMatrix(km.out$cluster-2, indexes[, 1])$overall["Accuracy"]
```

```
##  Accuracy
## 0.7092732
```

► **4. Solution.**

For the average linkage methods

```
confusionMatrix(hc.average.cluster-2, indexes[, 1])$table
```

```
##            Reference
## Prediction  -1   0   1
##         -1  87   8   0
##         0   43 126 112
##         1    0   2  21
```

```
confusionMatrix(hc.average.cluster-2, indexes[, 1])$overall["Accuracy"]
```

```
##  Accuracy
## 0.5864662
```

For the complete linkage methods

```
confusionMatrix(hc.complete.cluster-2, indexes[, 1])$table
```

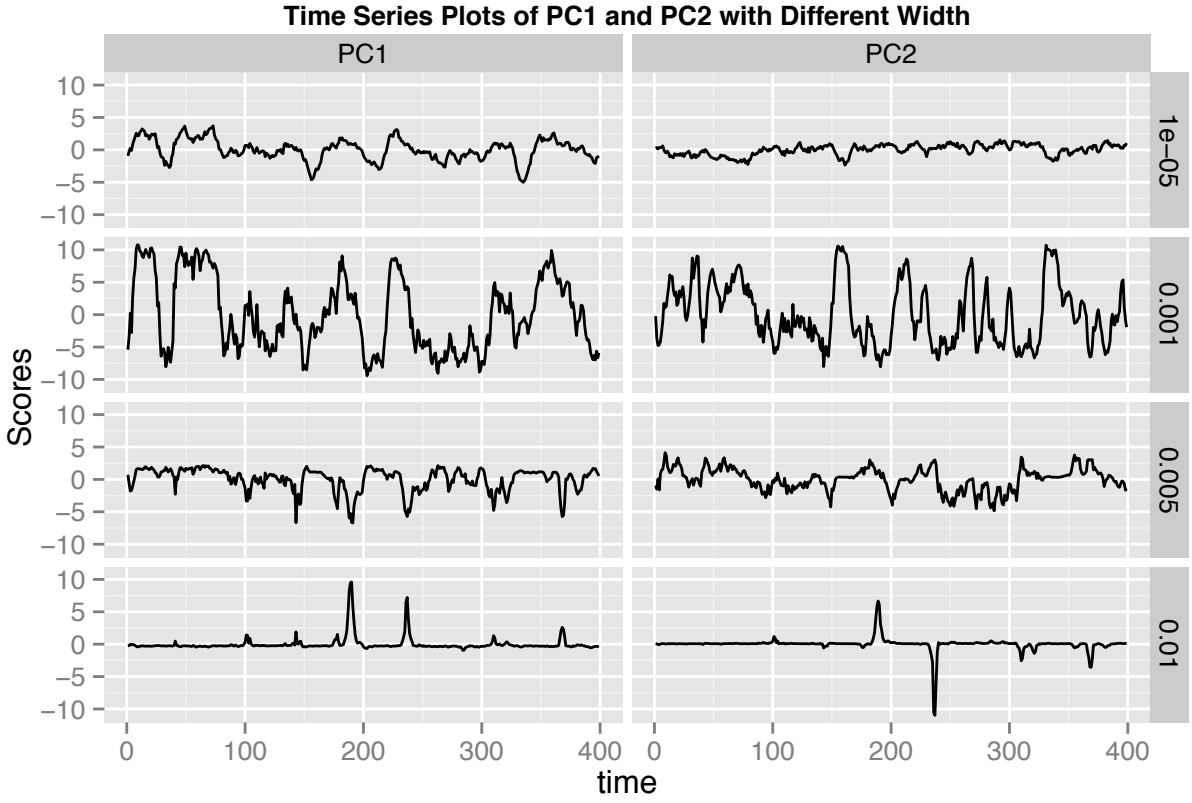
```
##            Reference
## Prediction  -1   0   1
##         -1 102  51   0
##         0   28  83  89
##         1    0   2  44
```

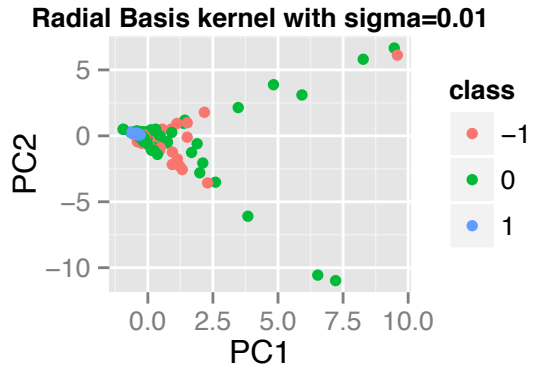
```
confusionMatrix(hc.complete.cluster-2, indexes[, 1])$overall["Accuracy"]
```

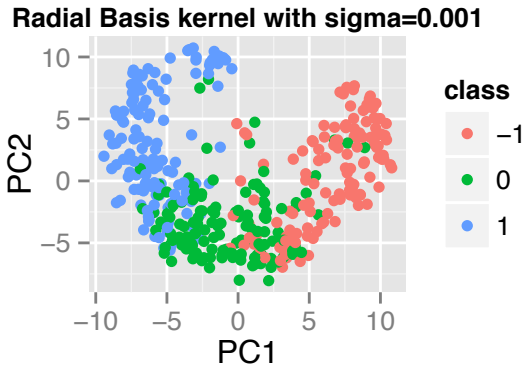
```
##  Accuracy
## 0.5739348
```

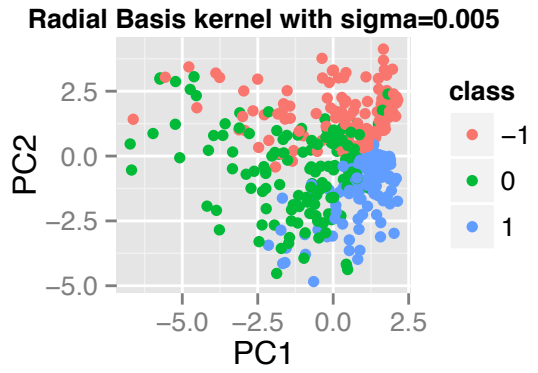
Both methods are worse than the K-means cluster, but the the complete linkage method is a little better than the other, not because the accuracy, but the high sensitivity, which make it possible to detect the anomalies more precisely.

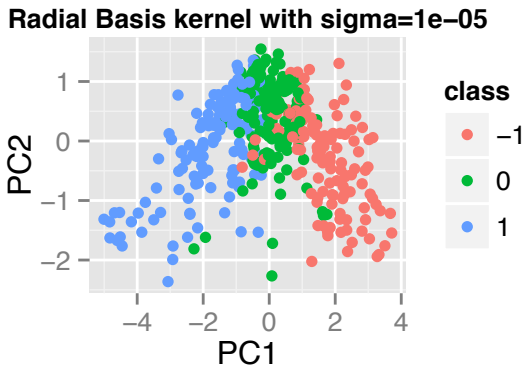
► **5. Solution.**









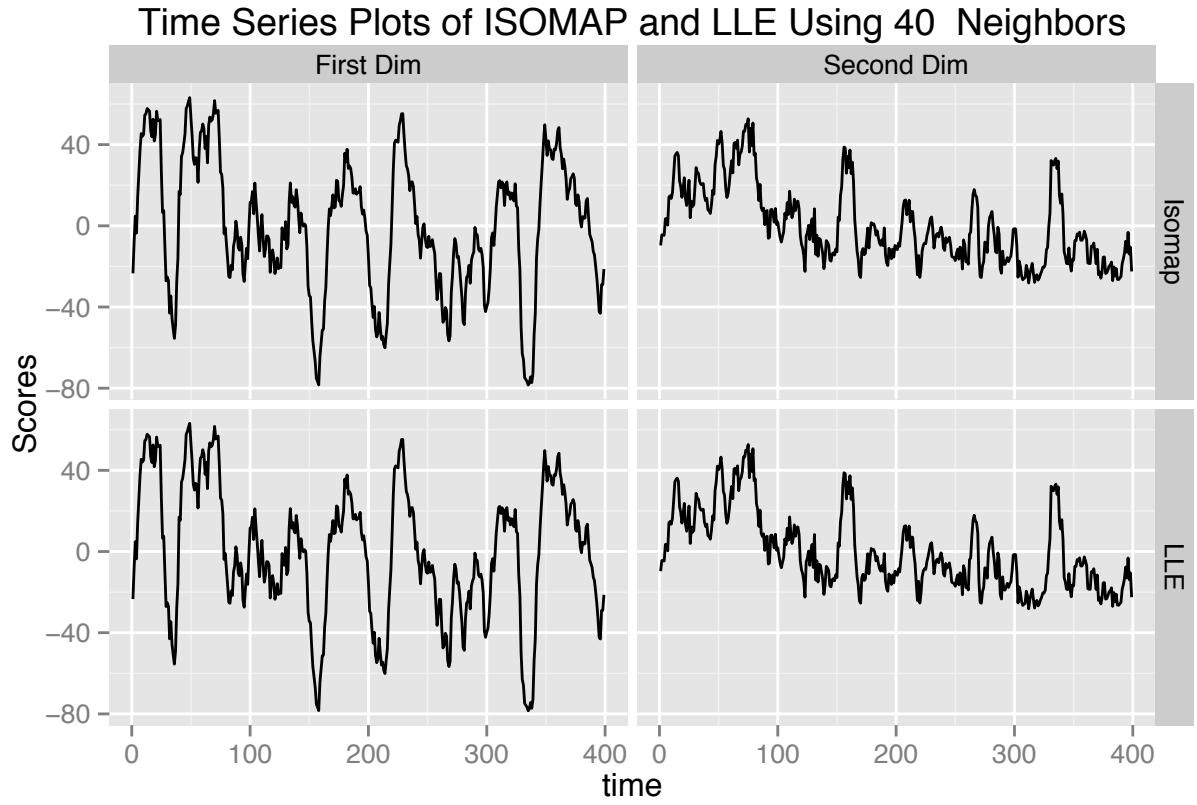


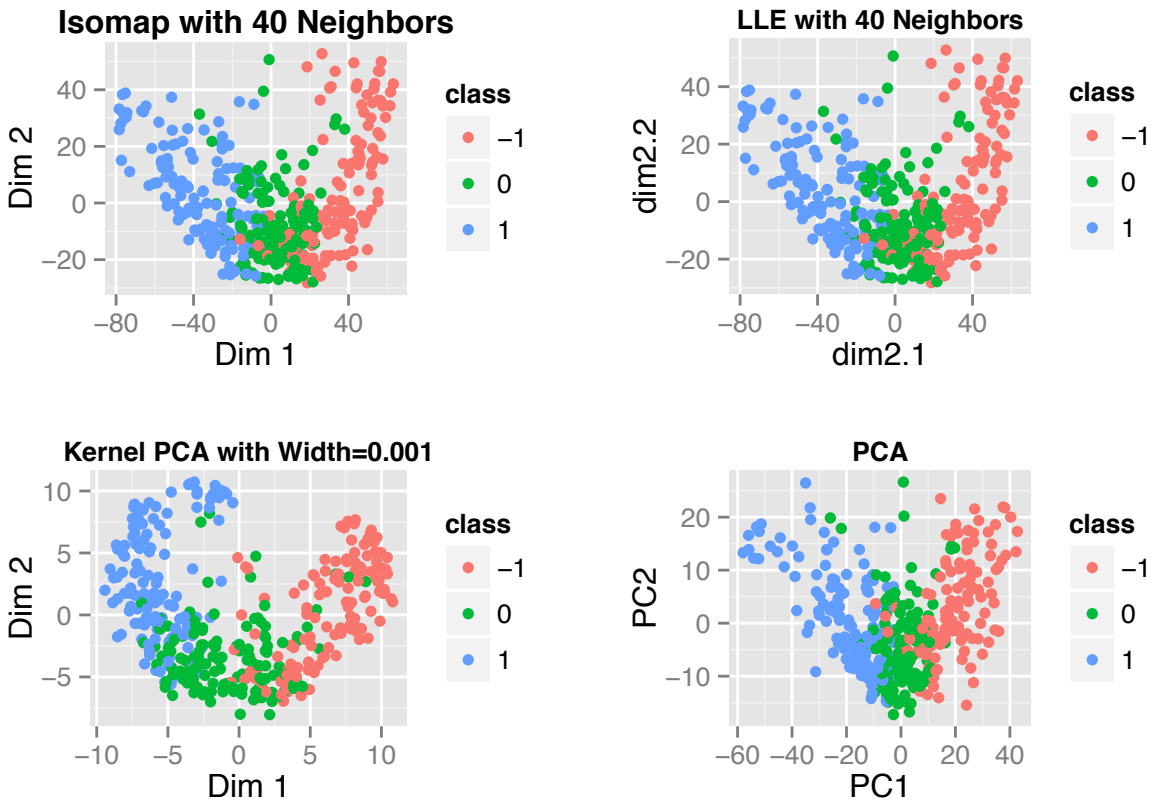

```
library(kernlab)
library(ggplot2)
sigma.grid <- c(0.01, 0.005, 0.001, 0.00001)
kpca.out <- c()
kpca.p <- list()
for (i in 1:length(sigma.grid)){
        kpca.out <- c(kpca.out, kpca( ~ ., data = data.frame(pr.input),
                                      kpar = list(sigma = sigma.grid[i])))
        kpca.p[[i]] <- ggplot(data = data.frame(kpca.out[[i]]@rotated[, 1:2],
                                                class = factor(indexes[, 1])),
                              aes(x = X1, y = X2, colour = class)) +
                geom_point() +
                xlab("PC1") +
                ylab("PC2") +
                ggtitle(paste("Radial Basis kernel with sigma=",
                              sigma.grid[i], sep = "")) +
                theme(plot.title = element_text(lineheight = 1,
                                                face = "bold",
                                                size = 10))
}
multiplot(plotlist = kpca.p, cols = 2)
four_result <- data.frame(matrix(NA, ncol = 4, nrow = 0))
names(four_result) <- c("time", "kernel_width", "PC", "Scores")
for (i in 1:length(sigma.grid)){
        if (exists("d")){
                rm(d)
        }
        if (exists("temp")){
```

```
                rm(temp)
        }
        d = data.frame(time = 1:nrow(kpca.out[[i]]@rotated[, 1:2]),
                       kernel_width = rep(sigma.grid[i], nrow(kpca.out[[i]]@rotated[, 1:2])),
                       kpca.out[[i]]@rotated[, 1:2])
        temp <- reshape(data = d,
                        direction = "long",
                        varying = list(names(d)[3:4]),
                        v.names = "Scores",
                        idvar = c("time", "kernel_width"),
                        timevar = "PC",
                        times = c("PC1", "PC2"))
        rownames(temp) <- NULL
        four_result <- rbind(four_result, temp)
}

multi.time.plot <- ggplot(data = four_result,
                          aes(x = time, y = Scores)) +
        geom_line() +
        facet_grid(kernel_width ~ PC)
```

► **6. Solution.**






```
library(RDRToolbox)
k = c(5, 10, 20, 40)
isomap.out <- list()
LLE.out <- list()
for (i in 1:length(k)){
        isomap.out[[i]] <- Isomap(data = pr.input, k = k[i])
        LLE.out[[i]] <- Isomap(data = pr.input, k = k[i])
}
isomap.out <- lapply(isomap.out,
                     data.frame,
                     class = factor(indexes[, 1]))
LLE.out <- lapply(LLE.out,
                  data.frame,
                  class = factor(indexes[, 1]))
isomap.p <- list()
for (i in 1:length(k)){
        isomap.p[[i]] <- ggplot(data = isomap.out[[i]],
                                aes(x = dim2.1, y = dim2.2, colour = class)) +
                geom_point()
}
multiplot(plotlist = isomap.p, cols = 2)
LLE.p <- list()
for (i in 1:length(k)){
        LLE.p[[i]] <- ggplot(data = LLE.out[[i]],
                             aes(x = dim2.1, y = dim2.2, colour = class)) +
                geom_point()
}
multiplot(plotlist = LLE.p, cols = 2)
```

```
iso.lle.time.series <- rbind(cbind(time = 1:nrow(isomap.out[[4]]),
                                   method = rep("Isomap",
                                                nrow(isomap.out[[4]])),
                                   isomap.out[[4]][, -3]),
                             cbind(time = 1:nrow(LLE.out[[4]]),
                                   method = rep("LLE",
                                                nrow(LLE.out[[4]])),
                                   LLE.out[[4]][, -3])
)
iso.lle.time.series <- reshape(data = iso.lle.time.series,
                               direction = "long",
                               varying = list(names(iso.lle.time.series)[3:4]),
                               v.names = "Scores",
                               idvar = c("time", "method"),
                               timevar = "Dim",
                               times = c("First Dim", "Second Dim")
)
rownames(iso.lle.time.series) <- NULL
iso.lle.time.plot <- ggplot(data = iso.lle.time.series,
                            aes(x = time, y = Scores)) +
        geom_line() +
        facet_grid(method ~ Dim)

pp.1 <- isomap.p[[4]]
pp.2 <- LLE.p[[4]]
pp.3 <- kpca.p[[3]]
pp.4 <- ggplot(data = data.frame(pr.out.nonsd$x[, 1:2],
                                 class = factor(indexes[, 1])),
               aes(x = PC1, y = PC2, colour = class)) +
        geom_point()
multiplot(pp.1, pp.3, pp.2, pp.4, cols = 2)
```