Homework 10

Your name

Honor code:

“I have neither given nor received unauthorized assistance on this assignment.” Type your initials here.

I receive help from “ ” and give help to “ “.

Problem 1

Load iris data in R. Apply PCA on this dataset. Draw a biplot with PC1 scores as x axis and PC2 scores as y axis. Show the proportion of variance explained (PVE) for the first and second PCs and explain their meanings.

```
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
## 7            4.6         3.4          1.4         0.3     setosa
## 8            5.0         3.4          1.5         0.2     setosa
## 9            4.4         2.9          1.4         0.2     setosa
## 10           4.9         3.1          1.5         0.1     setosa
## 11           5.4         3.7          1.5         0.2     setosa
## 12           4.8         3.4          1.6         0.2     setosa
## 13           4.8         3.0          1.4         0.1     setosa
## 14           4.3         3.0          1.1         0.1     setosa
## 15           5.8         4.0          1.2         0.2     setosa
## 16           5.7         4.4          1.5         0.4     setosa
## 17           5.4         3.9          1.3         0.4     setosa
## 18           5.1         3.5          1.4         0.3     setosa
## 19           5.7         3.8          1.7         0.3     setosa
## 20           5.1         3.8          1.5         0.3     setosa
## 21           5.4         3.4          1.7         0.2     setosa
## 22           5.1         3.7          1.5         0.4     setosa
## 23           4.6         3.6          1.0         0.2     setosa
## 24           5.1         3.3          1.7         0.5     setosa
## 25           4.8         3.4          1.9         0.2     setosa
## 26           5.0         3.0          1.6         0.2     setosa
## 27           5.0         3.4          1.6         0.4     setosa
## 28           5.2         3.5          1.5         0.2     setosa
## 29           5.2         3.4          1.4         0.2     setosa
## 30           4.7         3.2          1.6         0.2     setosa
## 31          4.8         3.1          1.6         0.2     setosa
## 32          5.4         3.4          1.5         0.4     setosa
## 33          5.2         4.1          1.5         0.1     setosa
## 34          5.5         4.2          1.4         0.2     setosa
## 35          4.9         3.1          1.5         0.2     setosa
## 36          5.0         3.2          1.2         0.2     setosa
## 37          5.5         3.5          1.3         0.2     setosa
## 38          4.9         3.6          1.4         0.1     setosa
## 39          4.4         3.0          1.3         0.2     setosa
## 40          5.1         3.4          1.5         0.2     setosa
## 41          5.0         3.5          1.3         0.3     setosa
## 42          4.5         2.3          1.3         0.3     setosa
## 43          4.4         3.2          1.3         0.2     setosa
## 44          5.0         3.5          1.6         0.6     setosa
## 45          5.1         3.8          1.9         0.4     setosa
## 46          4.8         3.0          1.4         0.3     setosa
## 47          5.1         3.8          1.6         0.2     setosa
## 48          4.6         3.2          1.4         0.2     setosa
## 49          5.3         3.7          1.5         0.2     setosa
## 50          5.0         3.3          1.4         0.2     setosa
## 51          7.0         3.2          4.7         1.4 versicolor
## 52          6.4         3.2          4.5         1.5 versicolor
## 53          6.9         3.1          4.9         1.5 versicolor
## 54          5.5         2.3          4.0         1.3 versicolor
## 55          6.5         2.8          4.6         1.5 versicolor
## 56          5.7         2.8          4.5         1.3 versicolor
## 57          6.3         3.3          4.7         1.6 versicolor
## 58          4.9         2.4          3.3         1.0 versicolor
## 59          6.6         2.9          4.6         1.3 versicolor
## 60          5.2         2.7          3.9         1.4 versicolor
## 61          5.0         2.0          3.5         1.0 versicolor
## 62          5.9         3.0          4.2         1.5 versicolor
## 63          6.0         2.2          4.0         1.0 versicolor
## 64          6.1         2.9          4.7         1.4 versicolor
## 65          5.6         2.9          3.6         1.3 versicolor
## 66          6.7         3.1          4.4         1.4 versicolor
## 67          5.6         3.0          4.5         1.5 versicolor
## 68          5.8         2.7          4.1         1.0 versicolor
## 69          6.2         2.2          4.5         1.5 versicolor
## 70          5.6         2.5          3.9         1.1 versicolor
## 71          5.9         3.2          4.8         1.8 versicolor
## 72          6.1         2.8          4.0         1.3 versicolor
## 73          6.3         2.5          4.9         1.5 versicolor
## 74          6.1         2.8          4.7         1.2 versicolor
## 75          6.4         2.9          4.3         1.3 versicolor
## 76          6.6         3.0          4.4         1.4 versicolor
## 77          6.8         2.8          4.8         1.4 versicolor
## 78          6.7         3.0          5.0         1.7 versicolor
## 79          6.0         2.9          4.5         1.5 versicolor
## 80          5.7         2.6          3.5         1.0 versicolor
## 81          5.5         2.4          3.8         1.1 versicolor
## 82          5.5         2.4          3.7         1.0 versicolor
## 83          5.8         2.7          3.9         1.2 versicolor
## 84          6.0         2.7          5.1         1.6 versicolor
## 85           5.4         3.0          4.5         1.5 versicolor
## 86           6.0         3.4          4.5         1.6 versicolor
## 87           6.7         3.1          4.7         1.5 versicolor
## 88           6.3         2.3          4.4         1.3 versicolor
## 89           5.6         3.0          4.1         1.3 versicolor
## 90           5.5         2.5          4.0         1.3 versicolor
## 91           5.5         2.6          4.4         1.2 versicolor
## 92           6.1         3.0          4.6         1.4 versicolor
## 93           5.8         2.6          4.0         1.2 versicolor
## 94           5.0         2.3          3.3         1.0 versicolor
## 95           5.6         2.7          4.2         1.3 versicolor
## 96           5.7         3.0          4.2         1.2 versicolor
## 97           5.7         2.9          4.2         1.3 versicolor
## 98           6.2         2.9          4.3         1.3 versicolor
## 99           5.1         2.5          3.0         1.1 versicolor
## 100          5.7         2.8          4.1         1.3 versicolor
## 101          6.3         3.3          6.0         2.5  virginica
## 102          5.8         2.7          5.1         1.9  virginica
## 103          7.1         3.0          5.9         2.1  virginica
## 104          6.3         2.9          5.6         1.8  virginica
## 105          6.5         3.0          5.8         2.2  virginica
## 106          7.6         3.0          6.6         2.1  virginica
## 107          4.9         2.5          4.5         1.7  virginica
## 108          7.3         2.9          6.3         1.8  virginica
## 109          6.7         2.5          5.8         1.8  virginica
## 110          7.2         3.6          6.1         2.5  virginica
## 111          6.5         3.2          5.1         2.0  virginica
## 112          6.4         2.7          5.3         1.9  virginica
## 113          6.8         3.0          5.5         2.1  virginica
## 114          5.7         2.5          5.0         2.0  virginica
## 115          5.8         2.8          5.1         2.4  virginica
## 116          6.4         3.2          5.3         2.3  virginica
## 117          6.5         3.0          5.5         1.8  virginica
## 118          7.7         3.8          6.7         2.2  virginica
## 119          7.7         2.6          6.9         2.3  virginica
## 120          6.0         2.2          5.0         1.5  virginica
## 121          6.9         3.2          5.7         2.3  virginica
## 122          5.6         2.8          4.9         2.0  virginica
## 123          7.7         2.8          6.7         2.0  virginica
## 124          6.3         2.7          4.9         1.8  virginica
## 125          6.7         3.3          5.7         2.1  virginica
## 126          7.2         3.2          6.0         1.8  virginica
## 127          6.2         2.8          4.8         1.8  virginica
## 128          6.1         3.0          4.9         1.8  virginica
## 129          6.4         2.8          5.6         2.1  virginica
## 130          7.2         3.0          5.8         1.6  virginica
## 131          7.4         2.8          6.1         1.9  virginica
## 132          7.9         3.8          6.4         2.0  virginica
## 133          6.4         2.8          5.6         2.2  virginica
## 134          6.3         2.8          5.1         1.5  virginica
## 135          6.1         2.6          5.6         1.4  virginica
## 136          7.7         3.0          6.1         2.3  virginica
## 137          6.3         3.4          5.6         2.4  virginica
## 138          6.4         3.1          5.5         1.8  virginica
## 139          6.0         3.0          4.8         1.8  virginica
## 140          6.9         3.1          5.4         2.1  virginica
## 141          6.7         3.1          5.6         2.4  virginica
## 142          6.9         3.1          5.1         2.3  virginica
## 143          5.8         2.7          5.1         1.9  virginica
## 144          6.8         3.2          5.9         2.3  virginica
## 145          6.7         3.3          5.7         2.5  virginica
## 146          6.7         3.0          5.2         2.3  virginica
## 147          6.3         2.5          5.0         1.9  virginica
## 148          6.5         3.0          5.2         2.0  virginica
## 149          6.2         3.4          5.4         2.3  virginica
## 150          5.9         3.0          5.1         1.8  virginica
```

```
iris.pca = prcomp(iris[ , -5])
iris.pca = princomp(iris[ , -5])
iris.pca.scaled = princomp(iris[,-5], cor=T)
library(ggbiplot)
```

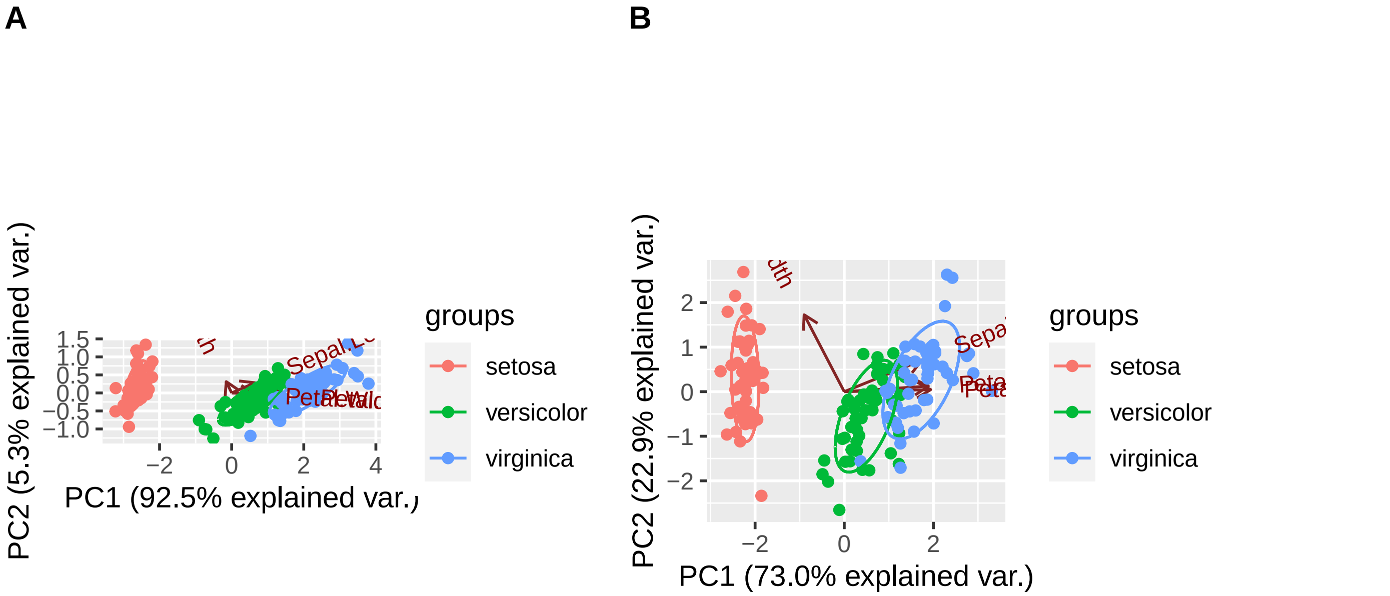
```
## Loading required package: ggplot2

## Loading required package: plyr

## Loading required package: scales

## Loading required package: grid
```

```
library(cowplot)
p1 = ggbiplot(iris.pca, groups =iris$Species,  ellipse = TRUE, obs.scale = 1, var.scale = 1)
p2 = ggbiplot(iris.pca.scaled, groups =iris$Species,  ellipse = TRUE, obs.scale = 1, var.scale = 1)
plot_grid(p1, p2, labels = c('A', 'B'), label_size = 12)
```



PVE is 92.5% for PC1 and 73% ofr PC2. This means PC1 is better as there is less unexplained variance within the model.

Problem 2

Load the MNIST data set.

```
library(keras)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```
mnist <- dataset_mnist()
# Use the following data if dataset_mnist() doesn't work
# mnist = readRDS("MNIST_data.rds")
idx_train = which(mnist$train$y%in% c("5", "6"))
idx_test = which(mnist$test$y%in% c("5", "6"))
x.train <- mnist$train$x[idx_train,,]
y.train <- as.factor(mnist$train$y[idx_train])
x.test <- mnist$test$x[idx_test,,]
y.test <- as.factor(mnist$test$y[idx_test])
# reshape
x.train <- array_reshape(x.train, c(nrow(x.train), 784))
x.test <- array_reshape(x.test, c(nrow(x.test), 784))
# rescale
x.train <- x.train / 255
x.test <- x.test / 255
```

(a) Apply PCA on this dataset. Use biplot to visualize the data. Is there any evidence to show that there are differences between digits 5 and 6?

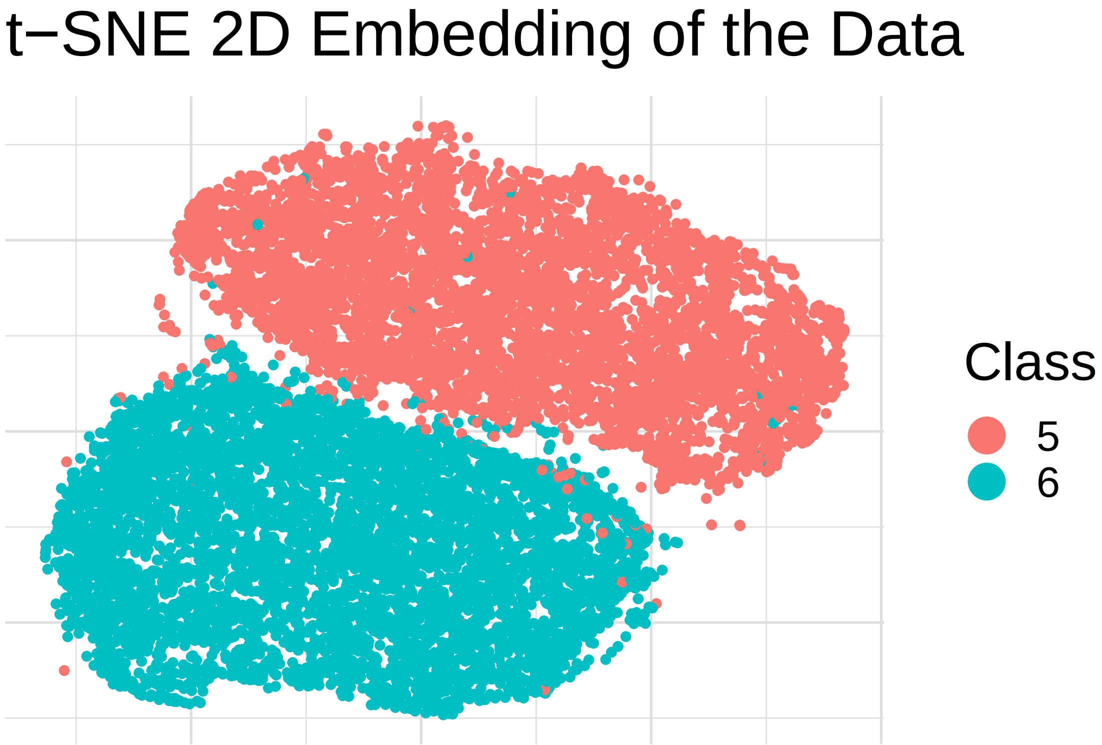
```
mnist.pca = princomp(x.train, cor=0)
total.var = sum(diag(cov(x.train)))
df = data.frame(npc = 1:ncol(x.train), cpve = cumsum(mnist.pca$sdev^2)/total.var)
```

```
library(Rtsne)
library(ggplot2)
mnist.tsne <- Rtsne(x.train, dims = 2, perplexity=30, verbose=TRUE, max_iter = 300)
```

```
## Performing PCA
## Read the 11339 x 50 data matrix successfully!
## Using no_dims = 2, perplexity = 30.000000, and theta = 0.500000
## Computing input similarities...
## Building tree...
##  - point 10000 of 11339
## Done in 11.61 seconds (sparsity = 0.010676)!
## Learning embedding...
## Iteration 50: error is 99.227454 (50 iterations in 1.58 seconds)
## Iteration 100: error is 90.660972 (50 iterations in 1.64 seconds)
## Iteration 150: error is 87.499179 (50 iterations in 1.61 seconds)
## Iteration 200: error is 87.045251 (50 iterations in 1.55 seconds)
## Iteration 250: error is 86.915493 (50 iterations in 1.60 seconds)
## Iteration 300: error is 3.425772 (50 iterations in 1.60 seconds)
## Fitting performed in 9.58 seconds.
```

```
embedding <- as.data.frame(mnist.tsne$Y)
embedding$Class <- as.factor(y.train)
p <- ggplot(embedding, aes(x=V1, y=V2, color=Class)) +
     geom_point(size=1.25) +
     guides(colour = guide_legend(override.aes = list(size=6))) +
     xlab("") + ylab("") +
```

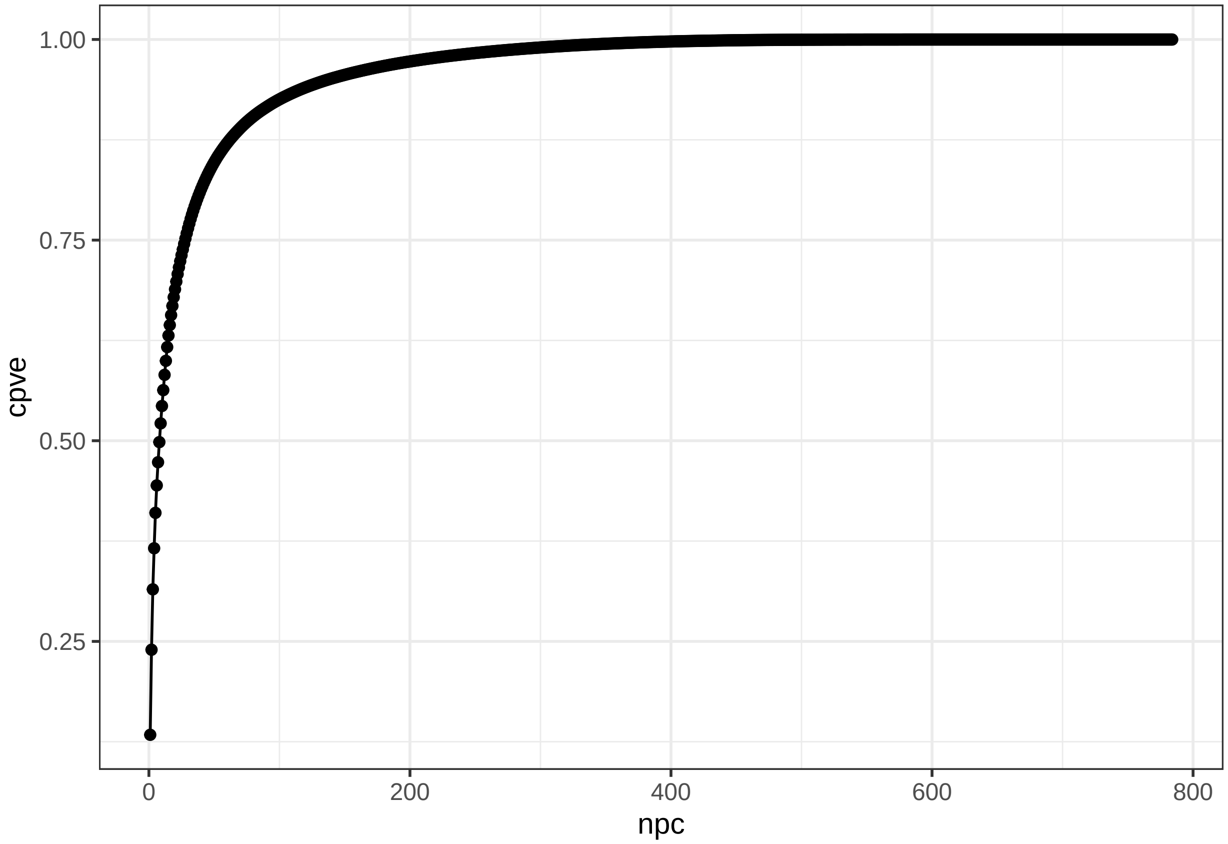
```r
    ggtitle("t-SNE 2D Embedding of the Data") +
    theme_light(base_size=20) +
    theme(strip.background = element_blank(),
          strip.text.x     = element_blank(),
          axis.text.x      = element_blank(),
          axis.text.y      = element_blank(),
          axis.ticks       = element_blank(),
          axis.line        = element_blank(),
          panel.border     = element_blank())
p
```



~~~
Yes, there is evidence to show that 5 is different from 6.


(b) Draw a scree plot (cPVE vs number of PCs) and select the number of PCs such that the cPVE is larger

```r
ggplot(df, aes(npc, cpve)) + geom_line() + geom_point() +theme_bw()
~~~



NPC such that cPVE is larger than 0.9 is somewhere around 100

(c) Use the PCs selected in (b) and project all observations on them. Calculate the projected PC scores. Use the scores as new features to fit a logistic regression model on training data. On test data, first project all test points on the PCs obtained from the training data. Then make predictions based on the projection scores and calculate the test accuracy on testing data.

```r
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
library(MASS)
npc = 100
z.train = as.data.frame(mnist.pca$scores[,1:npc])
z.train$y = y.train
qda.fit = qda(y~.,data = z.train)
z.test = as.data.frame(x.test %*% mnist.pca$loadings[,1:npc])
qda.pred = predict(qda.fit,as.data.frame(z.test))
qda_roc = roc(y.test,qda.pred$posterior[,2])
```

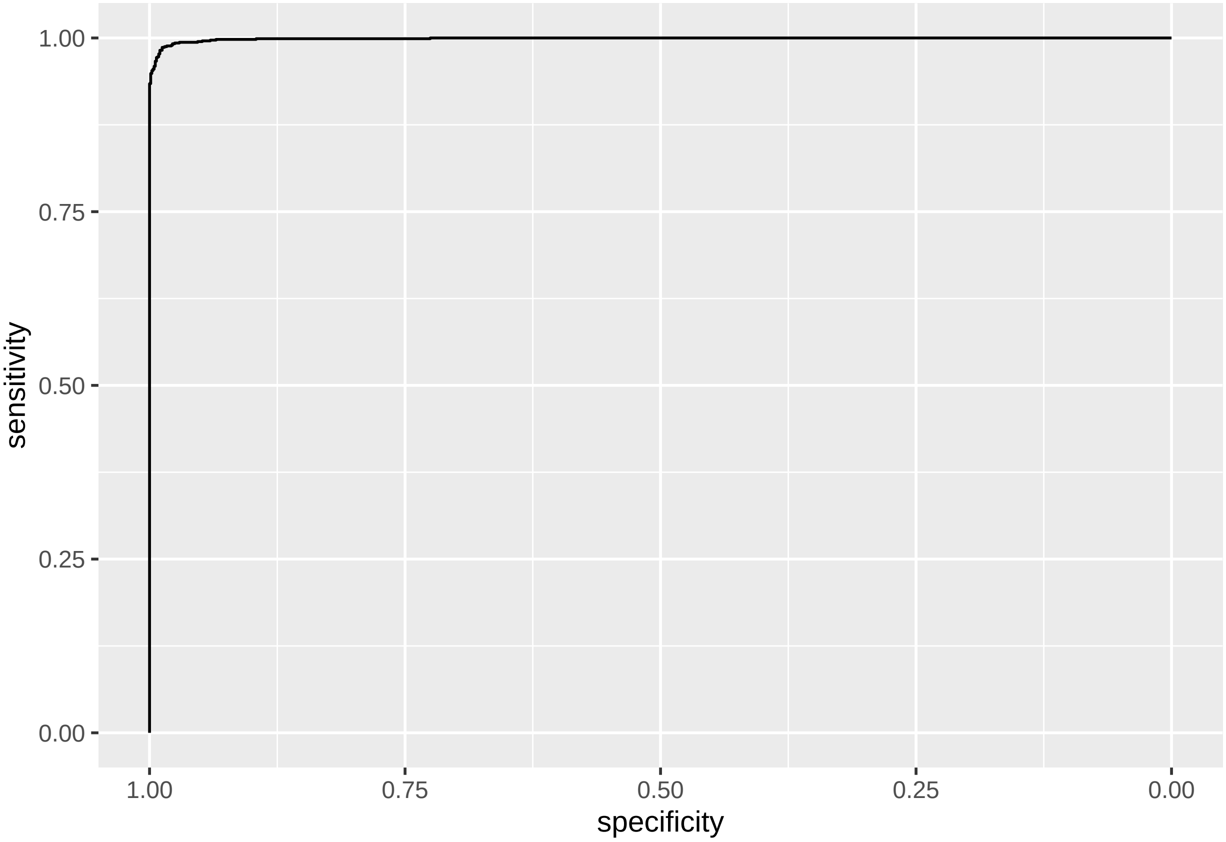
```
## Setting levels: control = 5, case = 6

## Setting direction: controls < cases
```

```
qda_roc
```

```
##
## Call:
## roc.default(response = y.test, predictor = qda.pred$posterior[,     2])
##
## Data: qda.pred$posterior[, 2] in 892 controls (y.test 5) < 958 cases (y.test 6).
## Area under the curve: 0.9989
```

```
ggroc(qda_roc)
```

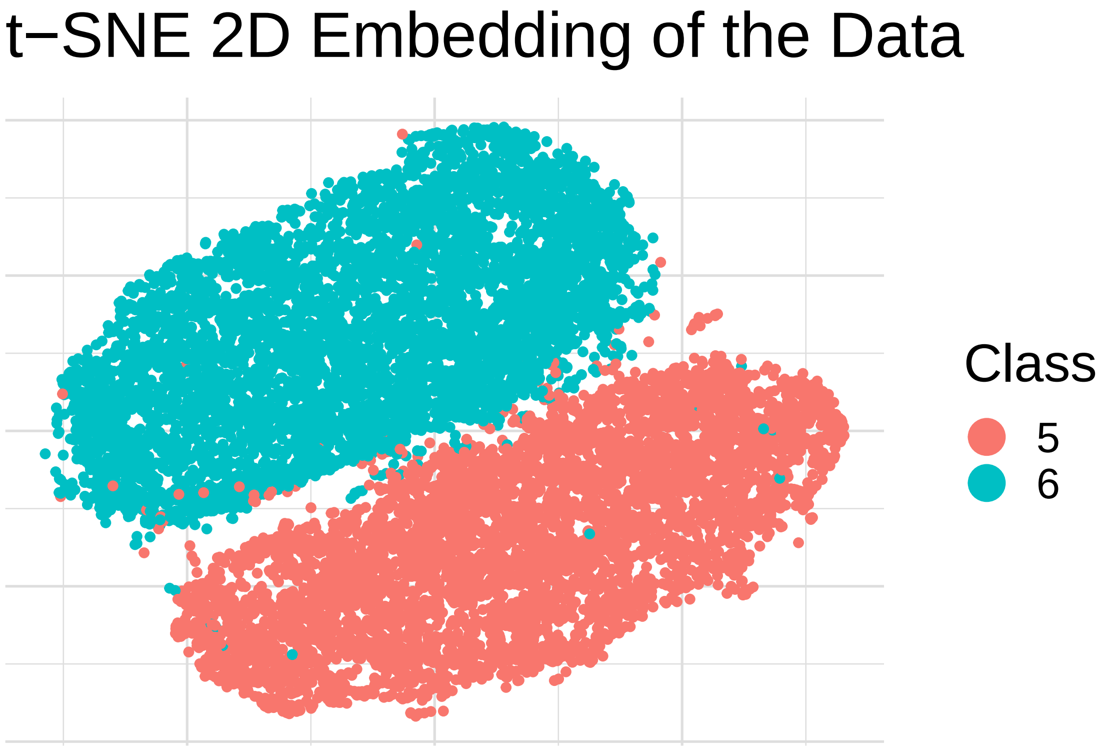


(d) Use tSNE method to reduce the dimension to two and draw a biplot.

```
library(Rtsne)
library(ggplot2)
mnist.tsne <- Rtsne(x.train, dims = 2, perplexity=30, verbose=TRUE, max_iter = 300)
```

```
## Performing PCA
## Read the 11339 x 50 data matrix successfully!
## Using no_dims = 2, perplexity = 30.000000, and theta = 0.500000
## Computing input similarities...
## Building tree...
##  - point 10000 of 11339
## Done in 13.21 seconds (sparsity = 0.010676)!
## Learning embedding...
## Iteration 50: error is 99.227454 (50 iterations in 1.66 seconds)
## Iteration 100: error is 90.877217 (50 iterations in 1.84 seconds)
## Iteration 150: error is 87.272738 (50 iterations in 1.79 seconds)
## Iteration 200: error is 87.010462 (50 iterations in 1.84 seconds)
## Iteration 250: error is 86.946926 (50 iterations in 1.93 seconds)
## Iteration 300: error is 3.393119 (50 iterations in 1.65 seconds)
## Fitting performed in 10.70 seconds.
```

```
embedding <- as.data.frame(mnist.tsne$Y)
embedding$Class <- as.factor(y.train)
p <- ggplot(embedding, aes(x=V1, y=V2, color=Class)) +
     geom_point(size=1.25) +
     guides(colour = guide_legend(override.aes = list(size=6))) +
     xlab("") + ylab("") +
     ggtitle("t-SNE 2D Embedding of the Data") +
     theme_light(base_size=20) +
     theme(strip.background = element_blank(),
           strip.text.x     = element_blank(),
           axis.text.x      = element_blank(),
           axis.text.y      = element_blank(),
           axis.ticks       = element_blank(),
           axis.line        = element_blank(),
           panel.border     = element_blank())
p
```


t−SNE 2D Embedding of the Data
Class
● 5
● 6