Solutions to HW 2

Shaoyi Zhang

April 15th, 2016

```r
library(ISLR)
library(ggplot2)
library(data.table)
```

Question 1.a

```r
# First, we should have a big picture of the iris data set.
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300  
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500  
##        Species  
##  setosa    :50  
##  versicolor:50  
##  virginica :50  
##                 
##                 
## 
```

```r
RawData = iris
RawData$Species = as.numeric(RawData$Species)
# Take a glance of the variance of each feature
apply(RawData,2,var)
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species 
##    0.6856935    0.1899794    3.1162779    0.5810063    0.6711409
```

```r
# See first elements of iris data set
head(RawData)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2       1
## 2          4.9         3.0          1.4         0.2       1
## 3          4.7         3.2          1.3         0.2       1
## 4          4.6         3.1          1.5         0.2       1
## 5          5.0         3.6          1.4         0.2       1
## 6          5.4         3.9          1.7         0.4       1
```

```r
# Partion feature matrix and label vector
predictorX = subset(RawData,select = -Species)
responseY = subset(RawData,select = Species)
# perform the princle component analysis
pr.out = prcomp(predictorX, center = T, scale=T)
```

```r
# Summary of the PCA
summary(pr.out)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.7084 0.9560 0.38309 0.14393
## Proportion of Variance 0.7296 0.2285 0.03669 0.00518
## Cumulative Proportion  0.7296 0.9581 0.99482 1.00000
```

```r
# The correlation between PCs and features
pr.out$rotation
```

```
##                     PC1         PC2        PC3        PC4
## Sepal.Length  0.5210659 -0.37741762  0.7195664  0.2612863
## Sepal.Width  -0.2693474 -0.92329566 -0.2443818 -0.1235096
## Petal.Length  0.5804131 -0.02449161 -0.1421264 -0.8014492
## Petal.Width   0.5648565 -0.06694199 -0.6342727  0.5235971
```

Question 1.b

For second PC, the loding of Sepal length is the greatest and the loading of Petal width and Petal length is near zero. Thus, we can say PC2 is the dimension of Sepal.

Question 1.c

```r
require(devtools)
```

```
## Loading required package: devtools
```

```r
install_github("vqv/ggbiplot")
```

```
## Skipping install for github remote, the SHA1 (7325e880) has not changed since last install.
##   Use `force = TRUE` to force installation
```

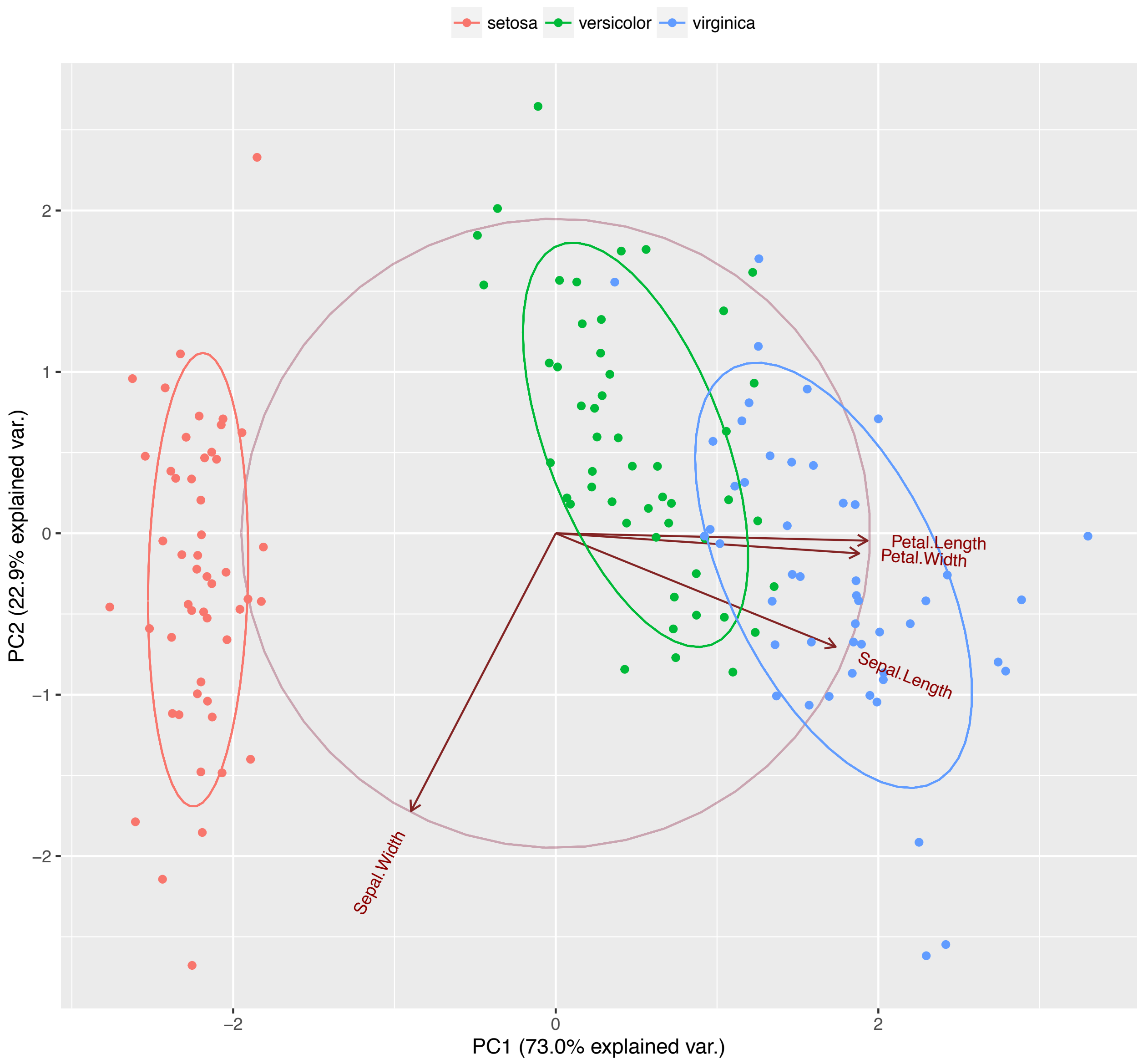
```r
library(ggbiplot)
```

```
## Loading required package: plyr

## Loading required package: scales

## Loading required package: grid
```

```
RawData = iris
RawData$Species = as.numeric(RawData$Species)
iris.pca <- prcomp(subset(RawData,select = -Species), scale. = TRUE)
# Use ggbiplot to create PCA graph
ggbiplot(iris.pca, obs.scale = 1, var.scale = 1,
  groups = iris$Species, ellipse = TRUE, circle = TRUE) +
  scale_color_discrete(name = '') +
  theme(legend.direction = 'horizontal', legend.position = 'top')
```



From the first principle component, we notice that Sepal.Length, Petal.Length and Petal.Width are highly correlated. From the second pricinple component, we notice that Sepal.Width contribute the most, with a light correlation with Sepal.Length. The remaining two priciple component are not very useful, since they only explained 3.67% and 0.518% of the total variance.

Setosa is different from other two species by Petal Length, Petal Width and Sepal.Length. It means that we can easily distinguish Setosa and the other two just by looking at the petal. Virginica and Versicolor can be distinguished by Sepal.Width. Virginica tends to have larger Sepal Width than Versicolor.

Question 1.d

```
pr.var = pr.out$sdev^2
# Calculation of PVE
pve = pr.var/sum(pr.var)
pve
```

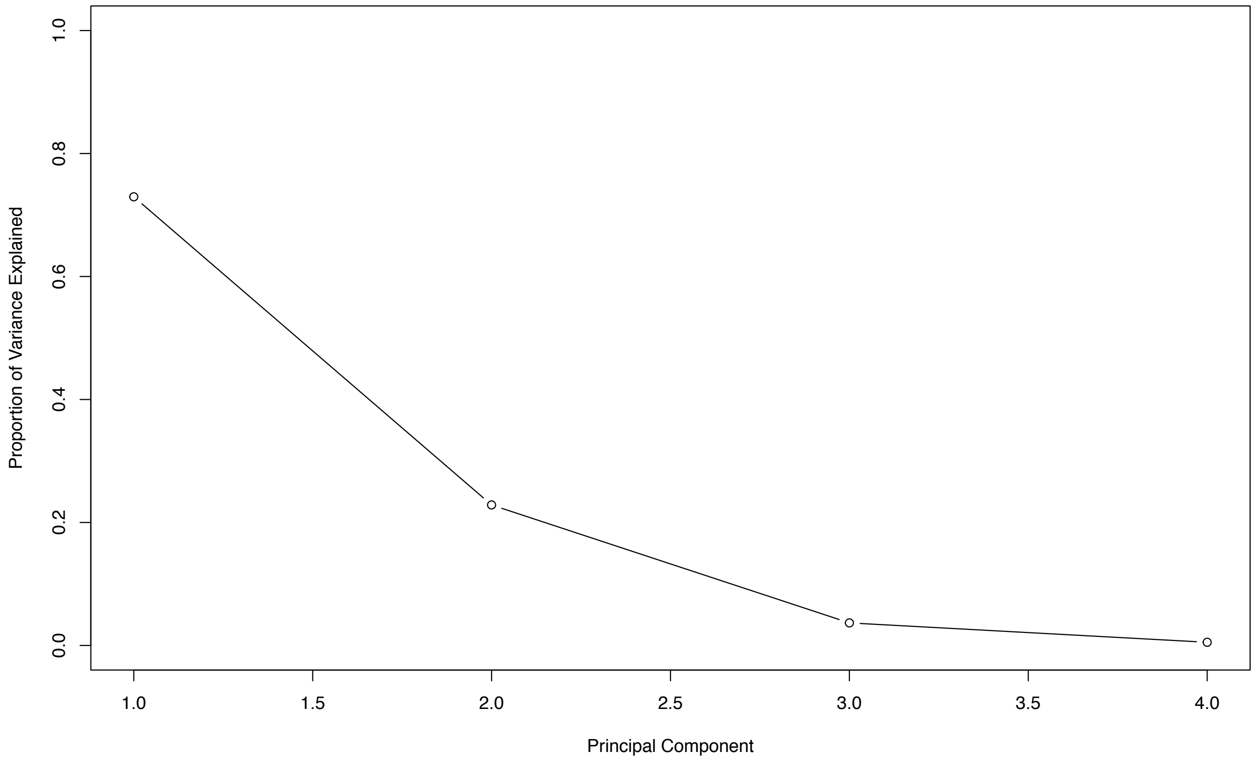
```
## [1] 0.729624454 0.228507618 0.036689219 0.005178709
```

The PVE of the first pricipal component is

```
## [1] 0.7296245
```

The PVE of the second pricipal component is

```
## [1] 0.2285076
```



Question 2.a

```
require(class)
```

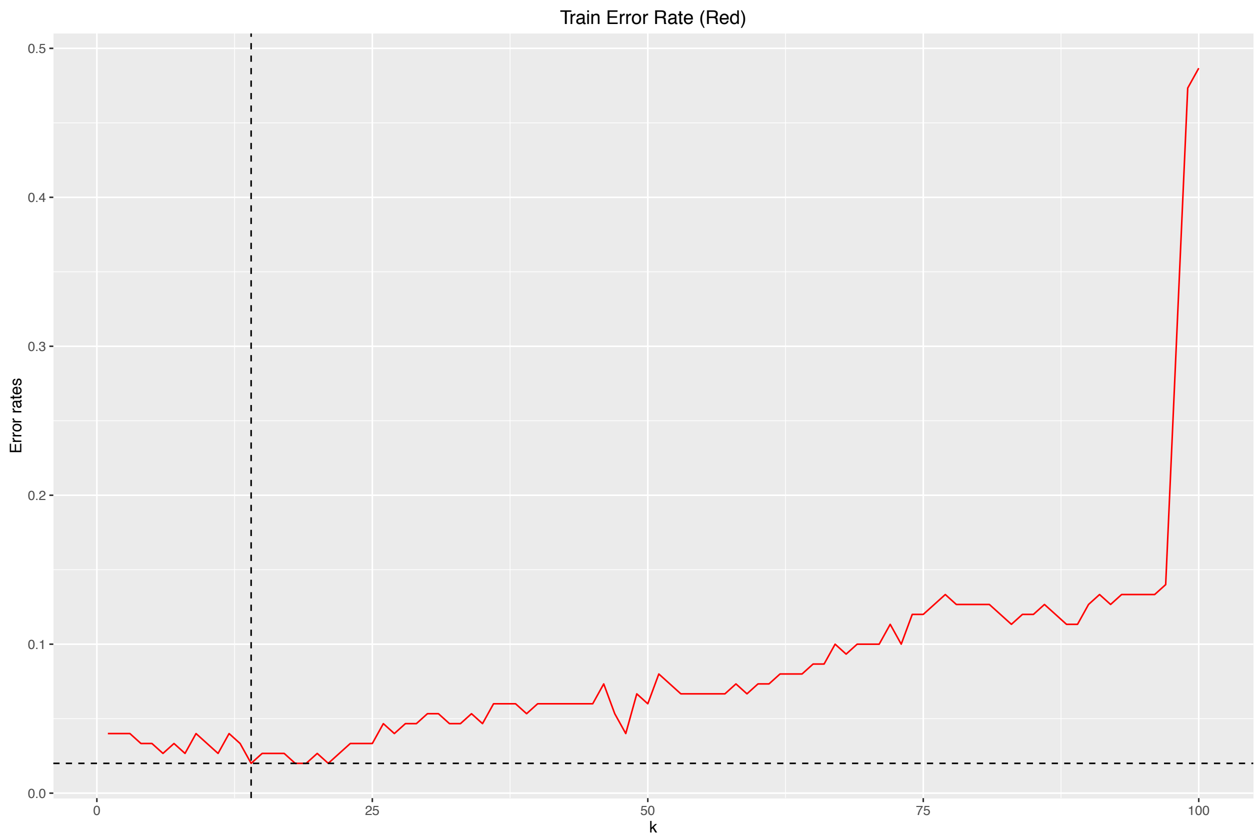
```
## Loading required package: class
```

```
require(boot)
```

```
## Loading required package: boot
```

```
df = data.frame(iris)
X = df[,1:4]
p.YTrain = NULL
train.error.rate = NULL
for(i in 1:100){
  set.seed(3)
  p.YTrain = knn.cv(train = X, cl = df$Species, k = i)
  train.error.rate[i] = mean(df$Species != p.YTrain)
}

gg4<-ggplot(data.frame(x = 1:100,y = train.error.rate))+geom_line(aes(x=x,y=y), color="Red")+xlab("k")+
gg4
```



The minimum error rate for KNN on iris is

```
## [1] 0.02
```

where k =

```
## [1] 14
```

Question 2.b

```r
require(MASS)
```

```
## Loading required package: MASS
```

```r
irisdf = data.frame(iris)
train.error.rate = NULL
lda.fit = lda(Species~.,data=irisdf,CV=T)
lda.error.rate = mean(irisdf$Species != lda.fit$class)
```

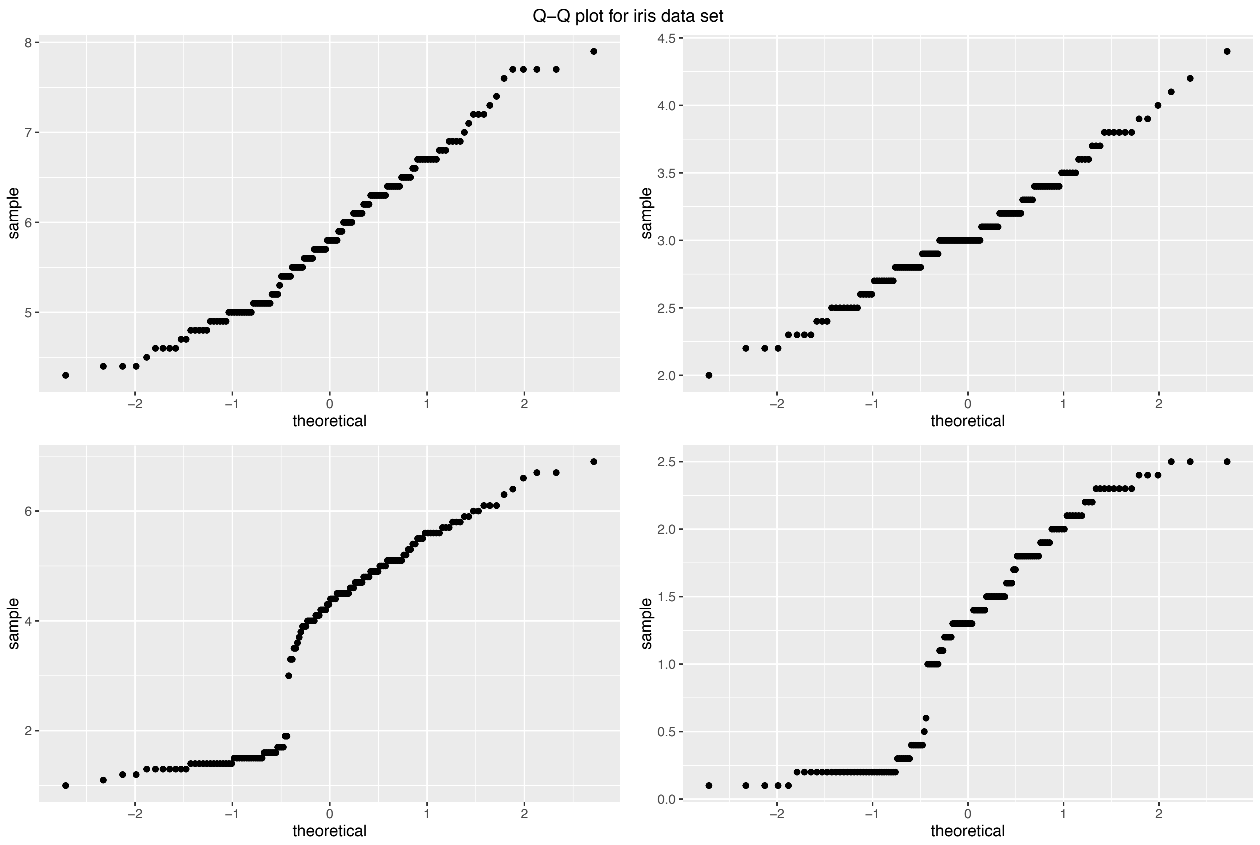
The error rate for LDA on iris is

```
## [1] 0.02
```

Question 2.c

Since minimum error rate for KNN and LDA are the same (2%) for iris data set, we will check the assumptions of these method. KNN has no assumption on data while LDA assumes the variables(feature) has a normal distribution. We need to take a look at the Q-Q plot of the iris data set.

```
## Loading required package: gridExtra
```



It is obvious that the variable "Petal.Length" and "Petal.Width" is NOT normally distributed. Thus, it is better to use KNN for iris data set.

Question 2.d

```r
df = data.frame(iris)
X = df[,1:4]
test = data.frame(rbind(c(4,2.5,3.0,0.5),c(6,4.0,1.8,1.5)))
colnames(test) = names(iris)[1:4]

# Peform KNN with optimal k = 14
knn(train=X, test = test, cl=df$Species, k = 14)
```

The prediciton of KNN is versicolor and setosa.

```
## [1] versicolor setosa
## Levels: setosa versicolor virginica
```

```r
test = data.frame(rbind(c(4,2.5,3.0,0.5),c(6,4.0,1.8,1.5)))
colnames(test) = names(iris)[1:4]

irisdf = data.frame(iris)
train.error.rate = NULL
lda.fit = lda(Species~.,data=irisdf,CV=F)

# Perform LDA
predict(lda.fit,test)$class
```

The prediciton of LDA is versicolor and setosa.

```
## [1] versicolor setosa
## Levels: setosa versicolor virginica
```

Additional Exercise PSTAT 231

Question 1

There is a need for fast and accurate face recognition system in airpots and hotels. However, previous face recogntion solutions are inaccurate and lacks scalability.

Question 2

Professor Turk (currently at UCSB CS department) used Principle Component Analysis to find a low dimensional representation of the complex, multidimensional pictures. In the paper, he only mentioned the brightness, the color of the pixels are variables in the analysis. I think there might be much more features used in Eigenface, but the importance of Eigenface is about PCA instead of various features.

Qustion 3

Since PCA transform the dimension from x*y pixels into the number of images in the training set, Eigenface is much quicker than other face recognition technologies. Eigenface evalute faces by the distnace between the feature vector of input images and the pre-accomplished face space. If they are far away, it means the input image is not a known face and vice versa.

Question 4

Eigenface approach requires specific orientation of the face image (can't work with upsidedown faces). This approach also requires a lot of images of the same person shooted from different angles so that the algorithmn can generate a "ghost" like face of that person.