Homework 2

linshuangquan

2017.03.12

No.3

Because$X_k \sim N(\mu_k.\sigma_k)$,
Thus

$$Pr(Y = k|X = x) = \frac{\Pi_k f_k(c)}{\sum_{l=1}^{k} \pi_l f_l(x)} \propto \Pi_k f_k(c)$$

$$log(Pr) \propto log(\pi_k) + log(\frac{1}{\sqrt{(2\pi)}\sigma_k} \exp(-\frac{(x-\mu_k)^2}{2\sigma_k^2}))$$

$$log(Pr) \propto log(\pi_k) - log(\sigma_k) - \frac{(x-\mu_k)^2}{2\sigma_k^2}$$

Thereforce the discriminant function:

$$\delta_k(x) = -\frac{1}{2\sigma_k} x^2 + x \cdot \frac{\mu_k}{\sigma_k^2} - \frac{\mu_k^2}{2\sigma_k^2} + log(\frac{\pi_k}{\sigma_k})$$

So the it is not linear, means quadratic.

No.5

a

If the Bayes decision boundary is linear, on the training set, QDA is more flexiblity so perform better. On the test set, LDA perform better than QDA because QDA wiil cause overfitting now.

b

If the Bayes decision bounary is non-linear, we expect QDA to perform better both on the training and test sets.

c

Yes, when the the sample size n increases, the boundary will be more complicate due to variance of data. So QDA will has better fit effect because it is more flexiblity than LDA.

d

False. QDA will cause overfitting when the Bayes boundary is linear, thus will has higher test rate than LDA.

No.11

```
library(ISLR)
attach(Auto)
```

a

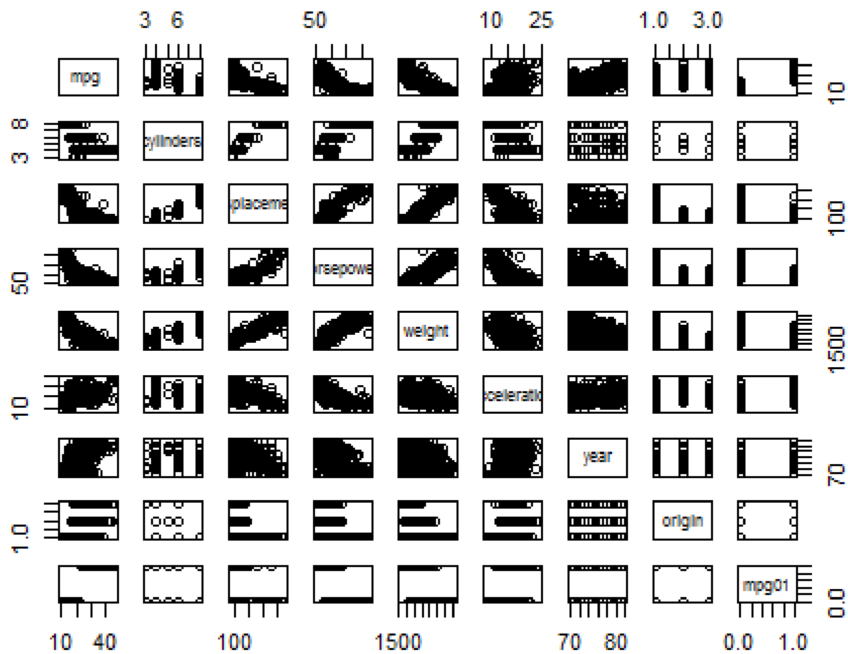
```
Auto$mpg01 <- ifelse(mpg>=median(mpg),1,0)
```

b

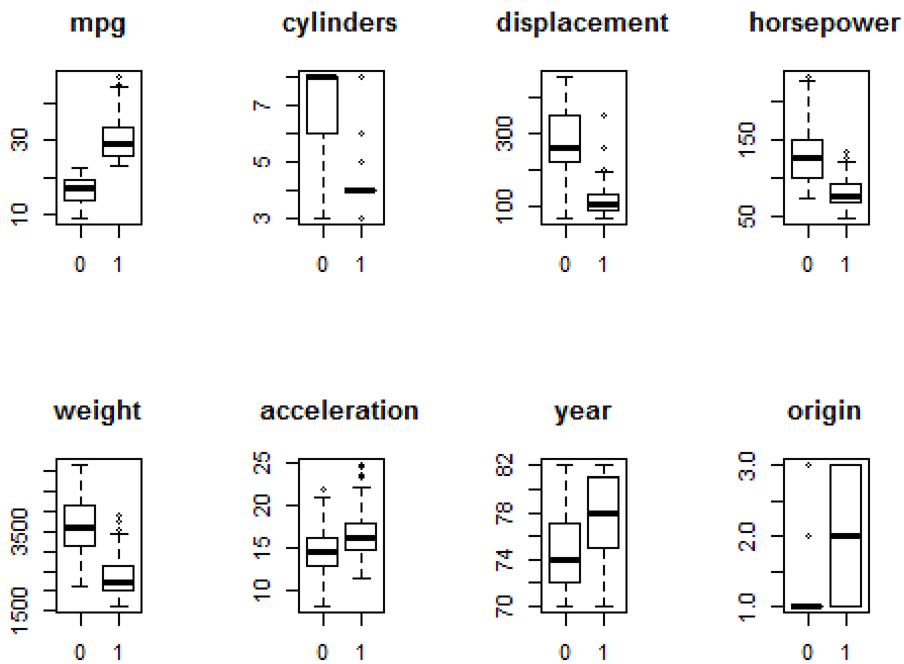
```
cor(Auto$mpg01,Auto[,1:8])

##            mpg  cylinders displacement horsepower     weight acceler
ation
## [1,] 0.8369392 -0.7591939   -0.7534766 -0.6670526 -0.7577566    0.34
68215
##           year    origin
## [1,] 0.4299042 0.5136984

pairs(Auto[,-9])
```




```
par(mfrow=c(2,4))
for(i in 1:8){
        boxplot(Auto[,i]~Auto$mpg01,main=colnames(Auto)[i])
}
```



From the correlation and the picture, we can find that these variables can be used for predicting "mpg01":mpg,cylinders,displacement,horsepower,weight,

c

```
set.seed(123)
sample <- sample(rep(c(1:4),length=dim(Auto)[1]))
trainset<- Auto[sample!=4,]
testset <- Auto[sample==4,]
```

d

```
library(MASS)
lda.fit = lda(mpg01 ~ cylinders + weight + displacement + horsepower, d
ata = trainset)
lda.pred = predict(lda.fit, testset)
mean(lda.pred$class != testset$mpg01)

## [1] 0.05102041
```

e

```
qda.fit = qda(mpg01 ~ cylinders + weight + displacement + horsepower, d
ata = trainset)
qda.pred = predict(qda.fit, testset)
mean(qda.pred$class != testset$mpg01)

## [1] 0.06122449
```

f

```r
glm.fit <- glm(mpg01 ~ cylinders + weight + displacement + horsepower,
data = trainset,family="binomial")
summary(glm.fit)

##
## Call:
## glm(formula = mpg01 ~ cylinders + weight + displacement + horsepower,

##     family = "binomial", data = trainset)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1555  -0.2520   0.1314   0.3949   3.2805
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  11.5346829  1.9126948   6.031 1.63e-09 ***
## cylinders     0.1435739  0.3856374   0.372  0.70967
## weight       -0.0019627  0.0007424  -2.644  0.00820 **
## displacement -0.0127788  0.0089016  -1.436  0.15113
## horsepower   -0.0474604  0.0159074  -2.984  0.00285 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 406.90  on 293  degrees of freedom
## Residual deviance: 170.24  on 289  degrees of freedom
## AIC: 180.24
##
## Number of Fisher Scoring iterations: 7

#remove the variable that p-value bigger than 0.05
glm.fit2 <- glm(mpg01 ~ displacement + horsepower, data = trainset,fami
ly="binomial")
summary(glm.fit2)

##
## Call:
## glm(formula = mpg01 ~ displacement + horsepower, family = "binomial",

##     data = trainset)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1116  -0.3315   0.1649   0.4749   3.3755
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)   9.049458   1.326893   6.820 9.10e-12 ***
## displacement -0.022111   0.003843  -5.754 8.74e-09 ***
## horsepower   -0.056033   0.015131  -3.703 0.000213 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 406.90  on 293  degrees of freedom
## Residual deviance: 177.64  on 291  degrees of freedom
## AIC: 183.64
##
## Number of Fisher Scoring iterations: 7
```

```
glm.probs = predict(glm.fit2, testset, type = "response")
glm.pred = ifelse(glm.probs>0.5,1,0)
mean(glm.pred != testset$mpg01)
```

```
## [1] 0.07142857
```

g

```
library(class)
errorrate <- rep(0,5)
set.seed(1234)
for(i in 1:10){
        knn.pred=knn(trainset[,2:5],testset[,2:5],trainset$mpg01,k=i)
        errorrate[i]<- mean(knn.pred != testset$mpg01)
}
names(errorrate) <- 1:10
errorrate
```

```
##          1          2          3          4          5          6
## 0.10204082 0.09183673 0.06122449 0.07142857 0.06122449 0.07142857
##          7          8          9         10
## 0.07142857 0.06122449 0.06122449 0.08163265
```

```
detach(Auto)
```

So when K=3 performes the best.

```
```