10_LDA

Park Ju ho

2022 6 12

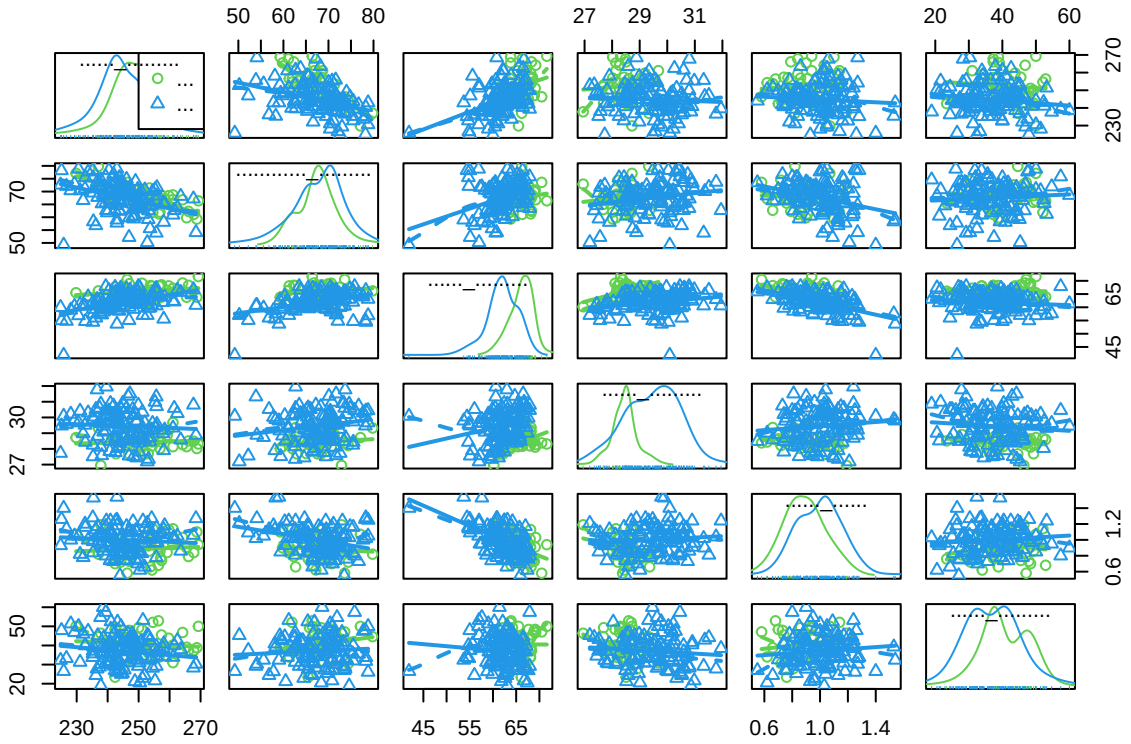
```
lpga<-read.table('D:/ /4-1 /  /R/lpga2008.txt', fileEncoding = 'utf-8',sep=",",header=T)
head(lpga)
```

```
##                    _     _    _    _
## 1     Ahn, Shi Hyun       249.4           64.6        61.2       27.44
## 2 Alfredsson, Helen       253.8           62.7        68.2       29.36
## 3 Ammaccapane, Dina       246.3           70.2        64.6       30.20
## 4       Bader, Beth       249.1           64.1        61.2       29.78
## 5       Bae, Kyeong       244.0           62.4        60.7       28.38
## 6     Baena, Marisa       254.2           64.7        60.9       29.21
##    _   _       _
## 1      1.10        34.5  6063           50
## 2      0.66        38.8 19343           74
## 3      0.74        40.5  1873           50
## 4      1.12        41.1  1212           65
## 5      1.02        43.9  2555           65
## 6      1.27        33.3  2282           52
```

```
lpga$  [rank(-lpga$ )<=40]<-' '
lpga$  [rank(-lpga$ )>40]<-' '
```

산점도를 통한 상관관계 분석

```
library(car)
scatterplotMatrix(~ _  +  _  + _  + _  + _ + _  |  ,data=lpga,col=c(3,4))
```



공분산 검정

귀무가설은 대부분 같은 걸로 잡음 귀무 가설 = 분산이 같다, 대립 가설 = 분산이 다르다

```
library(heplots)
boxM(lpga[,2:7],lpga$  )
```

```
##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  lpga[, 2:7]
## Chi-Sq (approx.) = 59.804, df = 21, p-value = 1.367e-05
```

p_value가 상당히 작은 것으로 보아 귀무 가설을 기각 할 수 있기에 분산이 다르다고 할 수 있다

MASS를 이용한 LDA

lda에서 Prior를 따로 입력하지 않으면 전체 데이터에서의 비율을 사용

```
library(MASS)
#
lpga.lda<-lda(  ~ _  +  _  + _  + _  + _ + _  ,data=lpga)


#
lpga.lda
```

```
## Call:
## lda(   ~  _   +   _   +  _   +
##       _   +  _   +  _  , data = lpga)
##
## Prior probabilities of groups:
##
## 0.2547771 0.7452229
##
## Group means:
##      _     _    _    _    _    _
##     252.4400        67.66750    66.10250    28.44000 0.8872500    40.59250
##     244.7521        67.54701    61.84188    29.45248 0.9995726    37.05128
##
## Coefficients of linear discriminants:
##                         LD1
##  _        0.00280365
##   _    0.02992010
##  _       -0.27874740
##  _        0.81317986
##  _         -0.85687947
##  _       -0.02655564
```

```
lpga.lda.p<-predict(lpga.lda)
```

Coefficients => 판별 분석의 판별식의 계수 (회귀분석 식과 똑같음 오차항이 없을 뿐) 결과값이 판별값이 아닌 사후 확률을 반환

예측값의 각 범주별 사후 확률

```
head(lpga.lda.p$posterior)
```

```
##
## 1 0.50899871 0.4910013
## 2 0.59300108 0.4069989
## 3 0.04121127 0.9587887
## 4 0.03470754 0.9652925
## 5 0.22497126 0.7750287
## 6 0.05776803 0.9422320
```

Confusion Matrix

```
lpga.lda.ct<-table(lpga$  ,lpga.lda.p$class)
prop.table(lpga.lda.ct,1) #   ,
```

```
##
##
##     0.77500000 0.22500000
##     0.04273504 0.95726496
```

```
sum(diag(prop.table(lpga.lda.ct))) #
```

```
## [1] 0.910828
```

0.91정도의 분류율을 보여주고 있음

사전확률을 부여한 LDA

```
lpga.lda<-lda(  ~ _  +  _  + _  + _  + _ + _  ,data=lpga, prior=c(0.2,0.8))
lpga.lda
```

```
## Call:
## lda(   ~  _   +   _   +  _   +
##       _   +  _  +  _  , data = lpga, prior = c(0.2,
##     0.8))
##
## Prior probabilities of groups:
##
## 0.2 0.8
##
## Group means:
##      _     _    _    _    _    _
##      252.4400        67.66750    66.10250    28.44000 0.8872500    40.59250
##      244.7521        67.54701    61.84188    29.45248 0.9995726    37.05128
##
## Coefficients of linear discriminants:
##                          LD1
##  _        0.00280365
##   _    0.02992010
##  _       -0.27874740
##  _        0.81317986
##  _         -0.85687947
##  _       -0.02655564
```

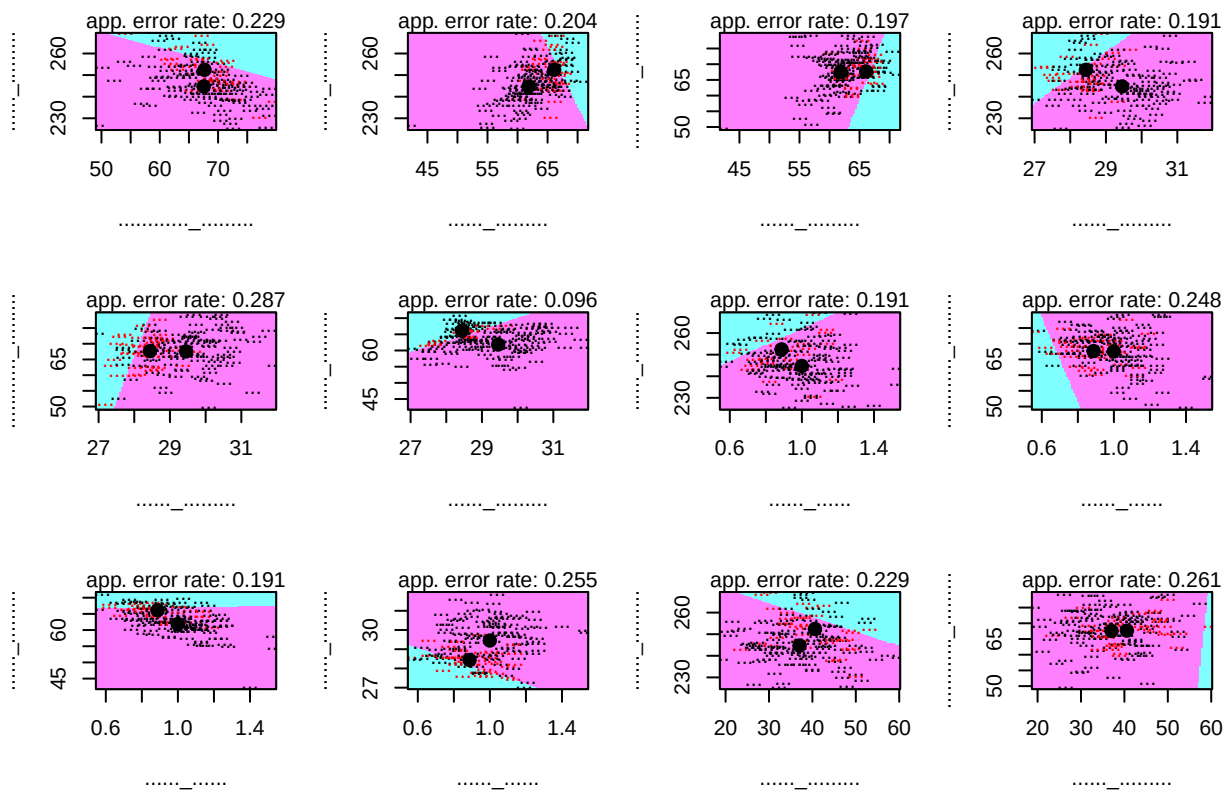
```
lpga.lda.p<-predict(lpga.lda)
head(lpga.lda.p$posterior)
```

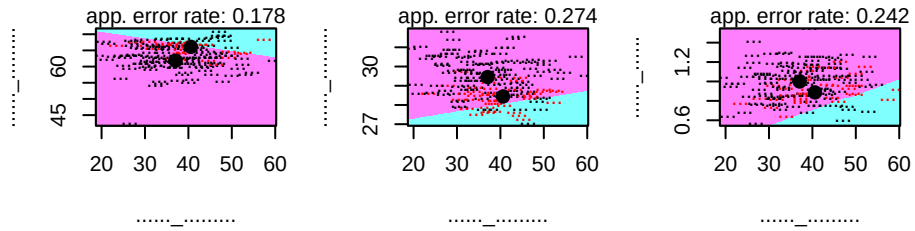
```
##
## 1 0.43118913 0.5688109
## 2 0.51584115 0.4841589
## 3 0.03047325 0.9695268
## 4 0.02561885 0.9743811
## 5 0.17509676 0.8249032
## 6 0.04290904 0.9570910
```

사전 확률을 부여하면 판별 규칙은 같으나 사후 확률이 달라짐

klaR을 이용한 LDA

```
library(klaR)
#
partimat(as.factor(  )~ _  +  _  + _  + _
         + _ + _  ,data=lpga,method='lda')
```



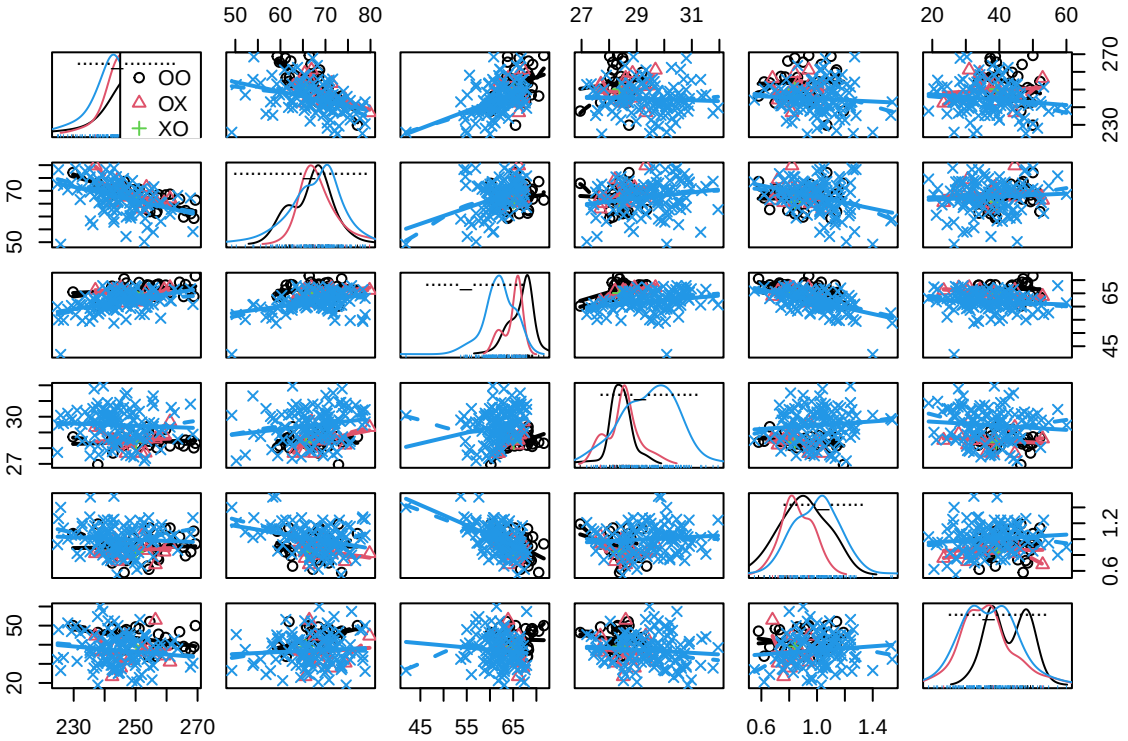



```
#
#       OO,       XX,          OX,          XO
lpga.lda.result<-cbind(lpga,lpga.lda.p$class)
lpga.lda.result$  [lpga.lda.result[,10]==lpga.lda.result[,11] &
                       lpga.lda.result[,10]==' ']<-'OO'
lpga.lda.result$  [lpga.lda.result[,10]==lpga.lda.result[,11] &
                       lpga.lda.result[,10]==' ']<-'XX'
lpga.lda.result$  [lpga.lda.result[,10]!=lpga.lda.result[,11] &
                       lpga.lda.result[,10]==' ']<-'OX'
lpga.lda.result$  [lpga.lda.result[,10]!=lpga.lda.result[,11] &
                       lpga.lda.result[,10]==' ']<-'XO'
head(lpga.lda.result)
```

```
##                    _     _    _    _
## 1     Ahn, Shi Hyun       249.4          64.6       61.2       27.44
## 2 Alfredsson, Helen       253.8          62.7       68.2       29.36
## 3 Ammaccapane, Dina       246.3          70.2       64.6       30.20
## 4       Bader, Beth       249.1          64.1       61.2       29.78
## 5       Bae, Kyeong       244.0          62.4       60.7       28.38
## 6     Baena, Marisa       254.2          64.7       60.9       29.21
##     _    _       _      lpga.lda.p$class
## 1      1.10        34.5  6063           50                          XX
## 2      0.66        38.8 19343           74                          OO
## 3      0.74        40.5  1873           50                          XX
## 4      1.12        41.1  1212           65                          XX
```

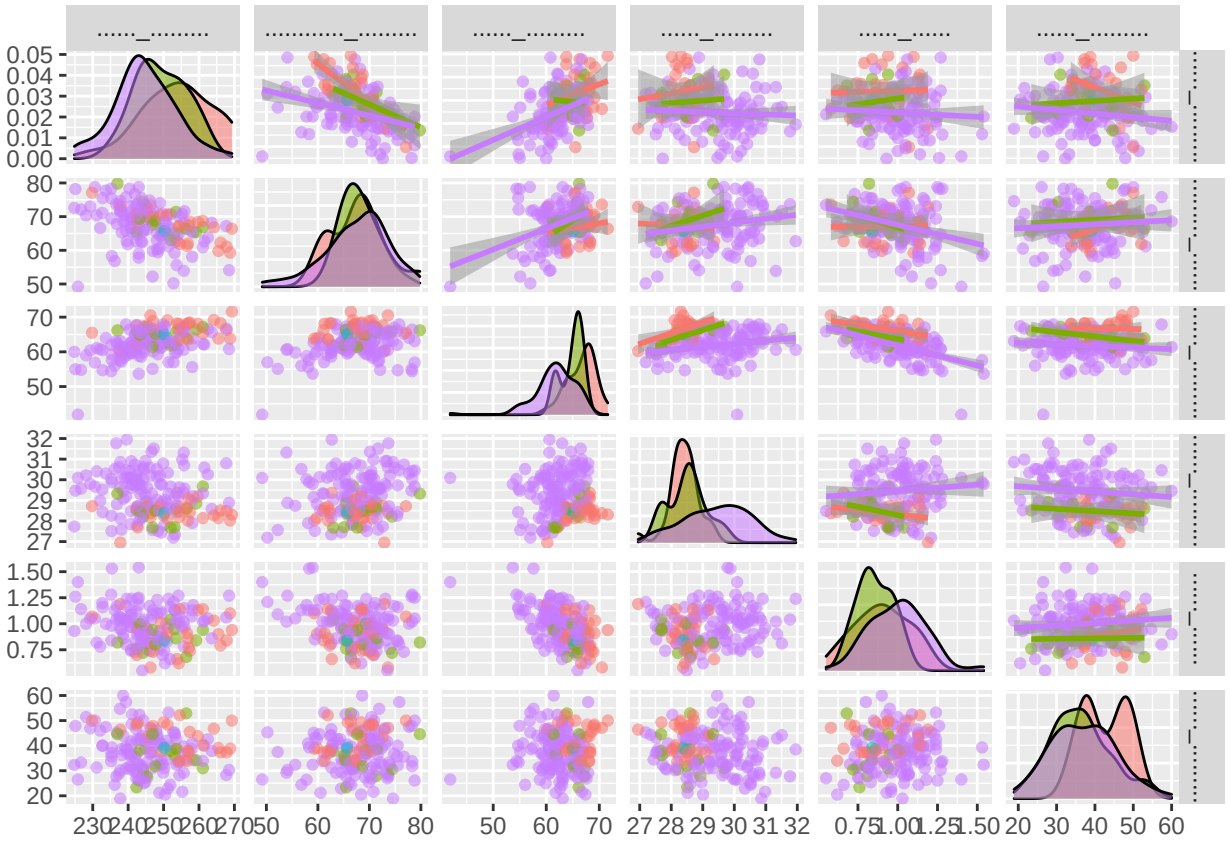
```
## 5       1.02         43.9  2555             65                              XX
## 6       1.27         33.3  2282             52                              XX
```

```
scatterplotMatrix(~ _  +  _  + _  + _  + _ + _  |  ,data=lpga.lda.result,col=c(1:4))
```



산점도로 표현

```
library(GGally)
#theme_update(text=element_text(family="AppleGothic"))#
ggpairs(lpga.lda.result[,2:7],aes(color=lpga.lda.result$  ,alpha=0.4),upper=list(continuous='smooth'))
```



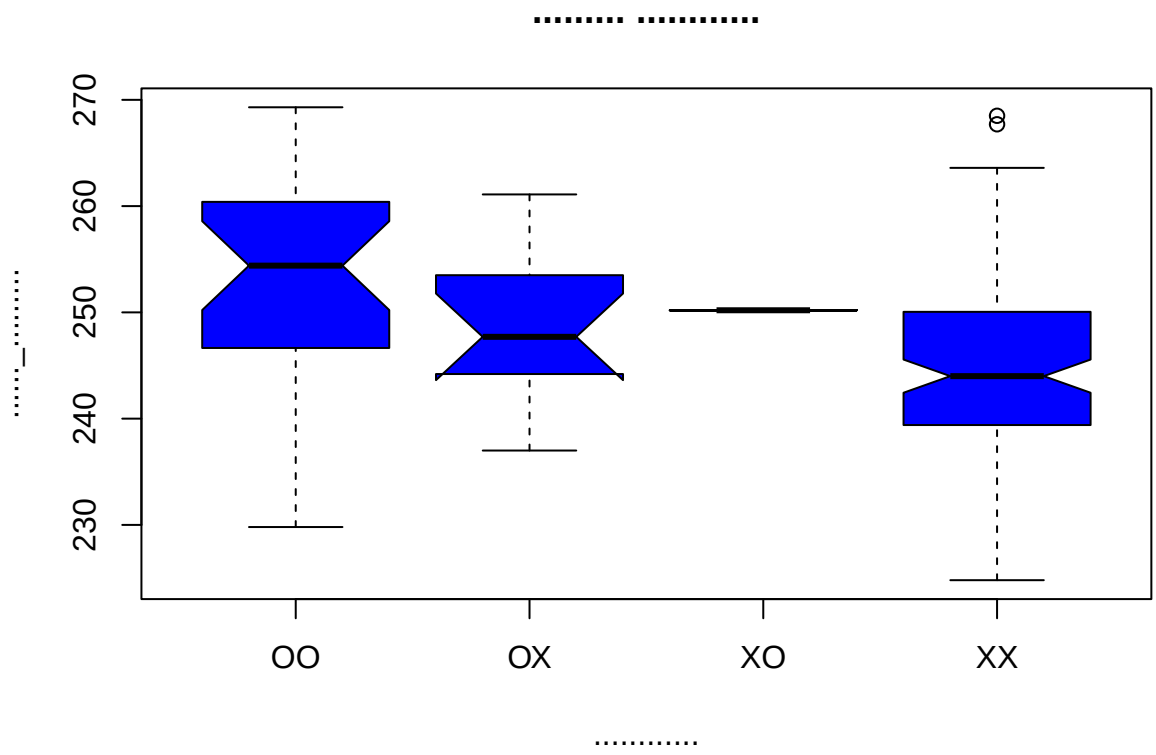
실선은 그 경향을 나타내고 분류결과가 총 4가지이기에 4가지 색을 보여줌

클래스를 기준으로 summary를 구해보기

```
library(doBy)
summaryBy( _  +  _  + _  + _  + _ + _  ~  , data=lpga.lda.result,FUN=c(mean,sd),na.rm=TRUE)
```

```
##        _  .mean    _  .mean   _  .mean
## 1        OO          253.9111              67.10741          66.68148
## 2        OX          249.3846              68.83077          64.90000
## 3        XO          250.2000              65.70000          65.10000
## 4        XX          244.7052              67.56293          61.81379
##     _  .mean   _ .mean   _  .mean   _  .sd
## 1          28.40296       0.9011111          42.65926       9.895700
## 2          28.51692       0.8584615          36.30000       7.147126
## 3          28.38000       0.8400000          39.10000             NA
## 4          29.46172       1.0009483          37.03362       8.728682
##      _  .sd   _  .sd   _  .sd   _ .sd   _  .sd
## 1           4.572907       2.642120       0.4745428     0.1692593       5.820802
## 2           4.529603       2.022787       0.6099506     0.1091517       7.898101
## 3                 NA             NA              NA            NA             NA
## 4           6.171837       3.794066       1.0438276     0.1783303       8.364365
```

```
boxplot( _  ~   ,data=lpga.lda.result,notch=TRUE,col='blue',main="     ", xlab="   ")
```




```
predict(lpga.lda,newdata=data.frame( _  =260,  _  =70, _  =65, _  =28, _ =1.5, _  =40))$posterior
```

```
##
## 1 0.8192054 0.1807946
```

oo와 xx의 summary차이를 보면 이 변수가 과연 분류에 효과적인지 확인 할 수 있음 => 즉 이 둘의 차이가 큰 변수가 분류를 잘 해준다고 할 수 있음

2차/비선형 판별 분석

선형과 해석 방법은 똑같음

```
lpga.qda<-qda(  ~ _  +  _  + _  + _  + _ + _  ,data=lpga)
lpga.qda
```

```
## Call:
## qda(   ~  _   +   _   +  _   +
##       _   +  _   +  _  , data = lpga)
##
```

```
## Prior probabilities of groups:
##
## 0.2547771 0.7452229
##
## Group means:
##      _     _    _    _    _    _
##      252.4400        67.66750    66.10250    28.44000 0.8872500    40.59250
##      244.7521        67.54701    61.84188    29.45248 0.9995726    37.05128
```

```
lpga.qda.p<-predict(lpga.qda) #  (posterior),   (x)

lpga.qda.ct<-table(lpga$  ,lpga.qda.p$class)
lpga.qda.ct
```

```
##
##
##      36   4
##       7 110
```

```
prop.table(lpga.qda.ct,1) #   ,
```

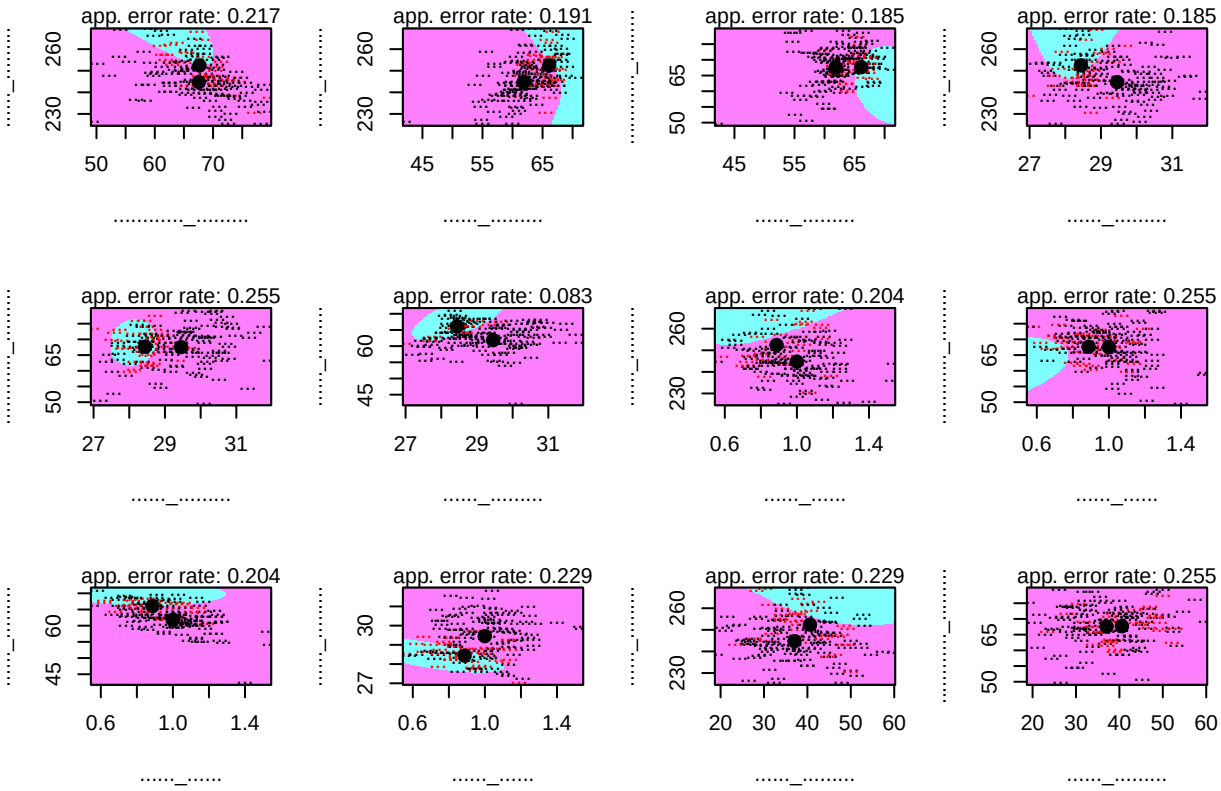
```
##
##
##     0.90000000 0.10000000
##     0.05982906 0.94017094
```

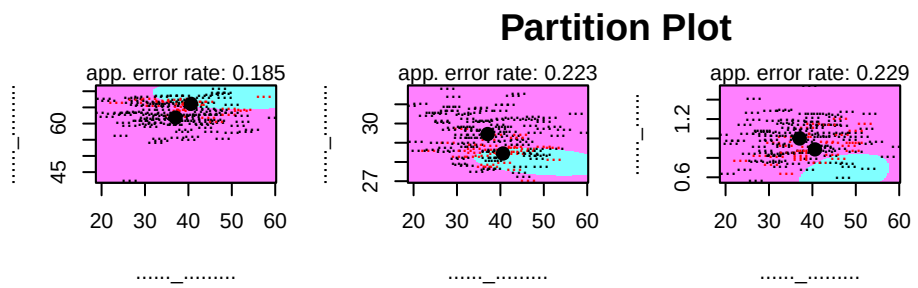
```
sum(diag(prop.table(lpga.qda.ct))) #
```

```
## [1] 0.9299363
```

그래프 그리기

```
library(klaR)
partimat(as.factor(  )~ _  +  _  + _  + _
         + _ + _  ,data=lpga,method='qda')
```


app. error rate: 0.217
app. error rate: 0.191
app. error rate: 0.185
app. error rate: 0.185
app. error rate: 0.255
app. error rate: 0.083
app. error rate: 0.204
app. error rate: 0.255
app. error rate: 0.204
app. error rate: 0.229
app. error rate: 0.229
app. error rate: 0.255



LDA와 달리 그 그래프 분류 경계가 곡선으로 나타나는 것을 확인 할 수 있음

```
lpga.qda.result<-cbind(lpga,lpga.qda.p$class)
lpga.qda.result$  [lpga.qda.result[,10]==lpga.qda.result[,11] &
                       lpga.qda.result[,10]==' ']<-'OO' #         !
lpga.qda.result$  [lpga.qda.result[,10]==lpga.qda.result[,11] &
                       lpga.qda.result[,10]==' ']<-'XX'
lpga.qda.result$  [lpga.qda.result[,10]!=lpga.qda.result[,11] &
                       lpga.qda.result[,10]==' ']<-'OX' #
lpga.qda.result$  [lpga.qda.result[,10]!=lpga.qda.result[,11] &
                       lpga.qda.result[,10]==' ']<-'XO'
table(lpga.qda.result[,12])
```

```
##
##  OO  OX  XO  XX
##  36   4   7 110
```

11개의 분류만 틀리고 모두 맞춘 것을 확인 할 수 있음

```
predict(lpga.qda,newdata=data.frame( _  =260,  _  =70, _  =65, _  =28, _ =1.5, _  =40))$posterior
```

```
##
## 1 0.9708559 0.02914412
```

사후확률

iris data를 이용한 LDA

공분산 동질성 테스트

```
boxM(iris[,1:4],iris$Species)
```

```
##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  iris[, 1:4]
## Chi-Sq (approx.) = 140.94, df = 20, p-value < 2.2e-16
```

p-value가 작으므로 공분산은 다르다고 할 수 있음

LDA 실행

```
library(MASS)
iris.qda<-qda(iris$Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width, data=iris)
iris.qda.p<-predict(iris.qda) #  ($posterior),   ($class)
iris.qda
```

```
## Call:
## qda(iris$Species ~ Sepal.Length + Sepal.Width + Petal.Length +
##     Petal.Width, data = iris)
##
## Prior probabilities of groups:
##     setosa versicolor  virginica
##  0.3333333  0.3333333  0.3333333
##
## Group means:
##            Sepal.Length Sepal.Width Petal.Length Petal.Width
## setosa            5.006       3.428        1.462       0.246
## versicolor        5.936       2.770        4.260       1.326
## virginica         6.588       2.974        5.552       2.026
```

해석은 똑같음

```
iris.qda.ct<-table(iris$Species,iris.qda.p$class)
iris.qda.ct
```

```
##
##              setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         48         2
##   virginica       0          1        49
```

```
prop.table(iris.qda.ct,1) #   ,
```

```
##
##              setosa versicolor virginica
##   setosa       1.00       0.00      0.00
##   versicolor   0.00       0.96      0.04
##   virginica    0.00       0.02      0.98
```

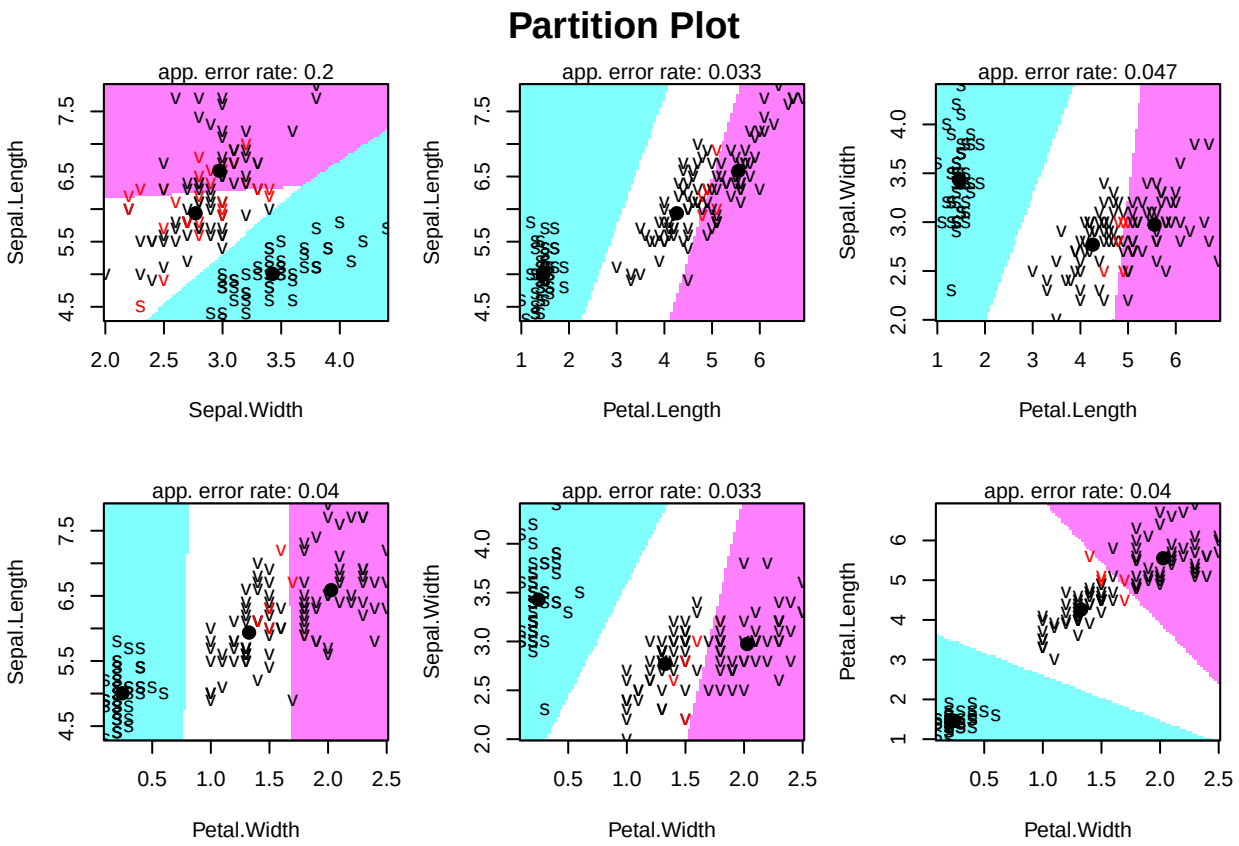
```
sum(diag(prop.table(iris.qda.ct))) #
```

```
## [1] 0.98
```

3개만 틀리고 모두 맞춘 결과를 보여주고 있다 정분류 0.98로 좋은 성적을 보여준다

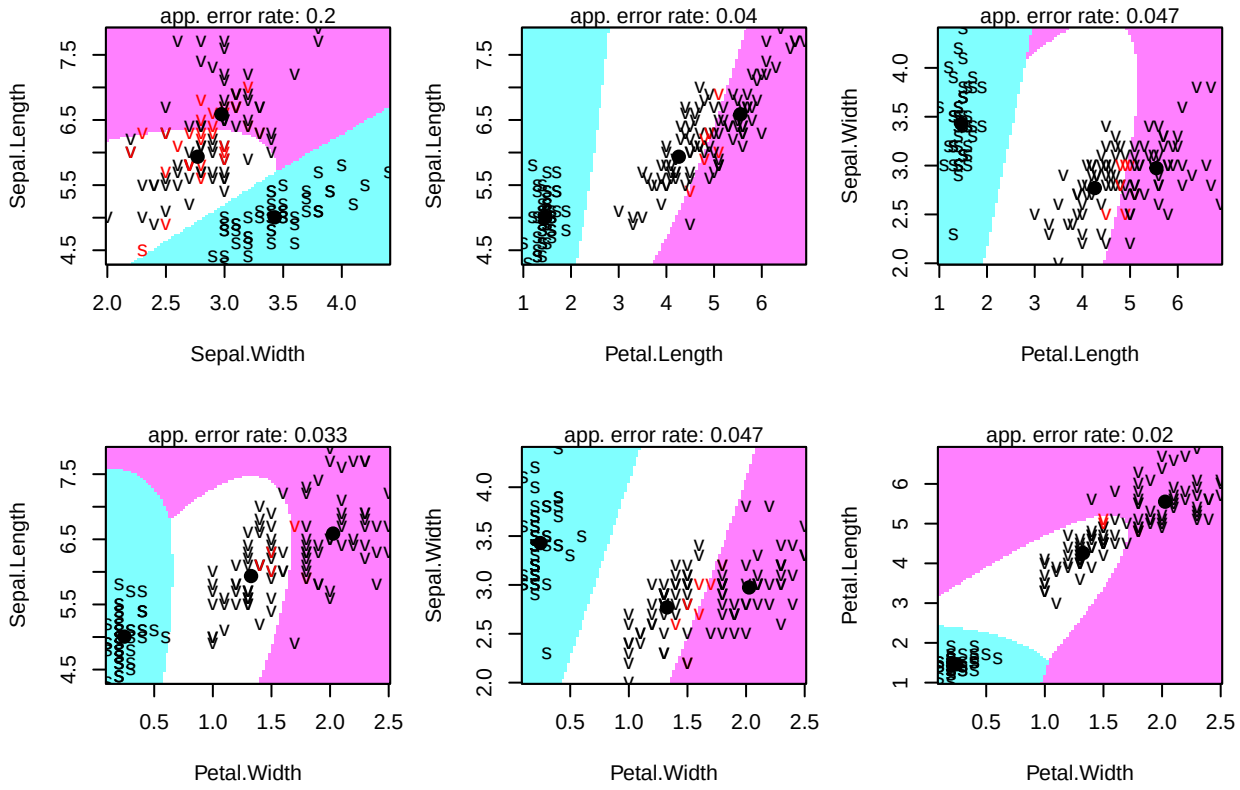
그래프 그리기

```
partimat(iris$Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,data=iris,method='lda')
```



비선형 판별 분석

```
partimat(iris$Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,data=iris,method='qda')
```



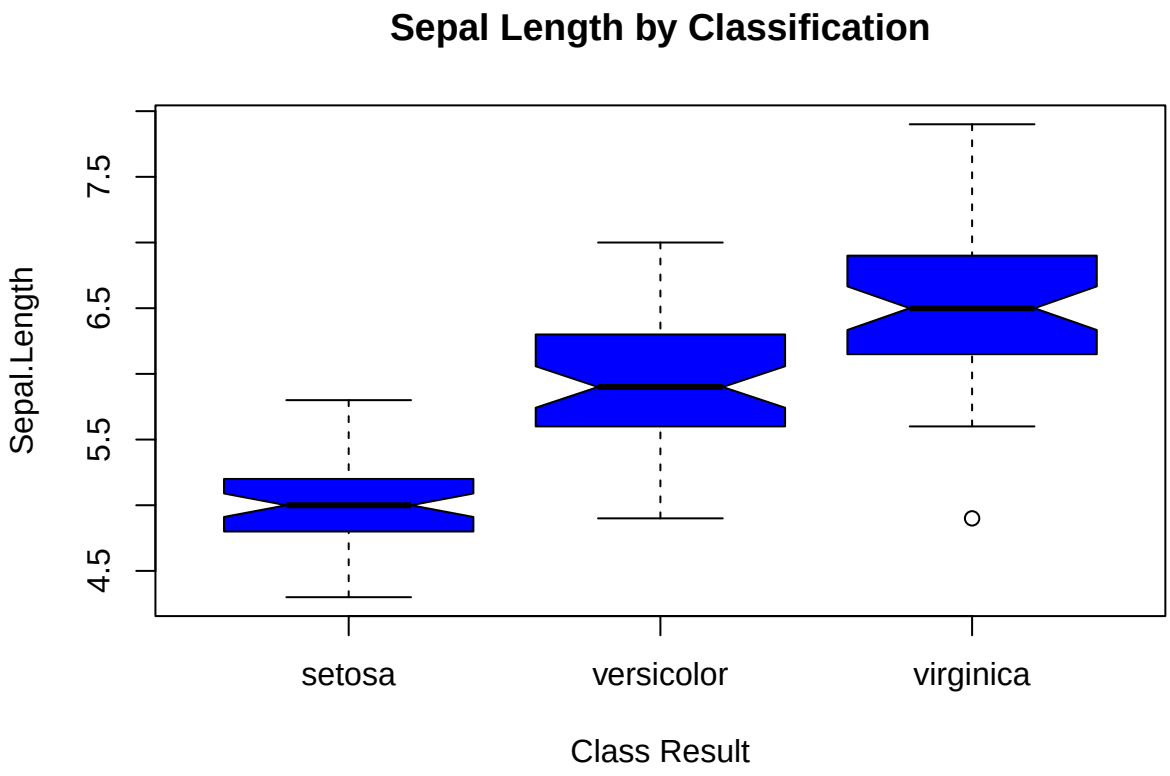

```
iris.qda.result<-cbind(iris,iris.qda.p$class)
#colnames       class
colnames(iris.qda.result) = c("Sepal.Length","Sepal.Width","Petal.Length","Petal.Width","Species","class
summaryBy(Sepal.Length+Sepal.Width+Petal.Length+Petal.Width~class,data=iris.qda.result,FUN=c(mean,sd),na
```

```
##        class Sepal.Length.mean Sepal.Width.mean Petal.Length.mean
## 1     setosa          5.006000         3.428000          1.462000
## 2 versicolor          5.942857         2.763265          4.248980
## 3  virginica          6.568627         2.976471          5.537255
##   Petal.Width.mean Sepal.Length.sd Sepal.Width.sd Petal.Length.sd
## 1         0.246000       0.3524897      0.3790644       0.1736640
## 2         1.314286       0.5240070      0.3107074       0.4682069
## 3         2.023529       0.6407777      0.3222348       0.5564030
##   Petal.Width.sd
## 1      0.1053856
## 2      0.1848423
## 3      0.2702504
```

```
names(iris.qda.result)
```

```
## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
## [6] "class"
```

```
boxplot(Sepal.Length~iris.qda.p$class,data=iris.qda.result,notch=TRUE,col='blue',main="Sepal Length by C
```



qda를 이용한 예측

```
predict(iris.qda,newdata=data.frame(Sepal.Length=45,Sepal.Width=30,Petal.Length=30,Petal.Width=15))$pos
```

```
##   setosa versicolor virginica
## 1      0          0         1
```