2.(a)

```
> library(MASS)
> library(msos)
> data("crabs")
>
x=cbind(rep(1,200),c(rep(0,50),rep(1,150)),c(rep(0,100),rep(1,10
0)),c(rep(0,150),rep(1,50)))
> bsm = bothsidesmodel.mle(x,as.matrix(crabs[,4:8]))
> bsm$SigmaR
           FL        RW       CL       CW        BD
FL  9.398456  6.650697 20.22712 22.77981  9.214437
RW  6.650697  5.081823 14.51954 16.39779  6.605713
CL 20.227123 14.519542 44.21318 49.74515 20.101845
CW 22.779807 16.397789 49.74515 56.15912 22.630463
BD  9.214437  6.605713 20.10185 22.63046  9.269161
```

2.(b)

```
> data("crabs")
> x.crabs = as.matrix(crabs[,4:8])
> y.crabs = rep(1:4,c(50,50,50,50))
> ld.crabs = lda(x.crabs,y.crabs)
> ld.crabs$a
         [,1]       [,2]      [,3] [,4]
FL -10.604987 -9.4798743 -3.318539    0
RW  -8.446403 -2.8831177 -6.827564    0
CL   1.774743 -0.3718635  4.912037    0
CW   8.204105  7.2892849 -1.751311    0
BD  -7.563446 -5.9232972  1.754568    0
> ld.crabs$c
[1] 23.48487 16.95592 21.11483  0.00000
```

2.(c)

```
> disc = x.crabs%*%ld.crabs$a
> disc = sweep(disc,2,ld.crabs$c,'+')
> imax = function(z) ((1:length(z))[z==max(z)])[1]
> yhat = apply(disc,1,imax)
> table(yhat,y.crabs)
    y.crabs
yhat  1  2  3  4
   1 45  0  0  0
   2  5 50  0  0
   3  0  0 50  3
   4  0  0  0 47
> ClassError = (5+3)/200
> ClassError
[1] 0.04
```

From the table, 0 crab had their species misclassified and 8 had their sex misclassified. The overall observed misclassification rate is 4%.

2.(d)

```
> yhat.cv = NULL
> varin=1:5
> n = nrow(x.crabs)
> for(i in 1:n) {
+   dcv = lda(x.crabs[-i,varin],y.crabs[-i])
+   dxi = x.crabs[i,varin]%*%dcv$a+dcv$c
+   yhat.cv = c(yhat.cv,imax(dxi))
+ }
> sum(yhat.cv!=y.crabs)/n
[1] 0.05
```

The overall misclassification rate is 0.05 and it is higher than the observed rate in part (c)

3.(a)

```
> data("SAheart")
> heartfull = glm(chd~.,data=SAheart,family=binomial)
```

3.(b)

```
> heartstepb = step(heartfull,scope=list(upper= ~.,lower =
~1),k=log(nrow(SAheart)),trace=0)
> heartstepb$formula
chd ~ tobacco + ldl + famhist + typea + age
```

From the result, tobacco, ldl, famhist, typea, age are included in the best model.

3.(c)

```
> heartfactor = glm(chd ~ tobacco+ldl+adiposity+obesity+alcohol,
+                   data=SAheart,family=binomial)
```

3.(d)

The result is summarized by the following table with code provided below.

	BIC	Obs. error	CV error
Full	533.4957	0.2662338	0.2813853
BIC	512.499	0.2575758	0.2640693
Factor	557.6685	0.2835498	0.2943723

```
> heartfull.BIC = heartfull$deviance + log(462)*10
> heartfull.BIC
[1] 533.4957
> heartfull.yhat = ifelse(predict(heartfull)>0,1,0)
> sum(heartfull.yhat!=SAheart[,'chd'])/462
[1] 0.2662338
> heartfull.err = NULL
> for(i in 1:462) {
+   yfiti = glm(chd ~., family = binomial,data =
SAheart,subset=(1:462)[-i])
+   dhati = predict(yfiti,newdata=SAheart[i,])
+   yhati = ifelse(dhati>0,1,0)
+   heartfull.err = c(heartfull.err,sum(yhati!
=SAheart[i,'chd']))
+ }
> mean(heartfull.err)
[1] 0.2813853

> heartstepb.BIC = heartstepb$deviance+log(462)*6
> heartstepb.BIC
[1] 512.499
> heartstepb.yhat = ifelse(predict(heartstepb)>0,1,0)
> sum(heartstepb.yhat!=SAheart[,'chd'])/462
[1] 0.2575758
> heartstepb.err = NULL
> for(i in 1:462) {
+   yfiti = glm(chd ~., family = binomial,data =
SAheart,subset=(1:462)[-i])
+   stepi = step(yfiti,scope=list(upper= ~.,lower =
~1),k=log(462),trace=0)
+   dhati = predict(stepi,newdata=SAheart[i,])
```

```
+   yhati = ifelse(dhati>0,1,0)
+   heartstepb.err = c(heartstepb.err,sum(yhati!
=SAheart[i,'chd']))
+ }
> mean(heartstepb.err)
[1] 0.2640693

> heartfactor.BIC = heartfactor$deviance + log(462)*6
> heartfactor.BIC
[1] 557.6685
> heartfactor.yhat = ifelse(predict(heartfactor)>0,1,0)
> sum(heartfactor.yhat!=SAheart[,'chd'])/462
[1] 0.2835498
> heartfactor.err = NULL
> for(i in 1:462) {
+   yfiti = glm(chd ~tobacco+ldl+adiposity+obesity+alcohol,
+               family = binomial,data = SAheart,subset=(1:462)
[-i])
+   dhati = predict(yfiti,newdata=SAheart[i,])
+   yhati = ifelse(dhati>0,1,0)
+   heartfactor.err = c(heartfactor.err,sum(yhati!
=SAheart[i,'chd']))
+ }
> mean(heartfactor.err)
[1] 0.2943723
```

3.(e)
i> FALSE. Step with BIC criterion has the lowest observed error rate
ii> FALSE. factoranalysis-based model is generally worst.
iii> TRUE
iv> TRUE
v> TRUE
vi> FALSE. Adiposity and obesity are not included in the stepwise procedure with BIC criterion, which is the best model.

4.(a)

```
> library(tree)
> district = read.csv("district115cong.csv",header = T)
> basetree = tree(as.factor(party)~pctMale + medAge + pctAge65 +
pctWhite + pctBlack +
+                   pctHisp + avgHouseSize + pctHighSch +
pctBach +
+                   pctVet + pctNativeBorn + pctUnemp +
medHouseIncome +
```

```
+                    medFamIncome + pctUnins + pctFamPov +
pctIndivPov,data=district)
> plot(basetree);text(basetree)
> summary(basetree)

Classification tree:
tree(formula = as.factor(party) ~ pctMale + medAge + pctAge65 +
    pctWhite + pctBlack + pctHisp + avgHouseSize + pctHighSch +
    pctBach + pctVet + pctNativeBorn + pctUnemp + medHouseIncome
+
    medFamIncome + pctUnins + pctFamPov + pctIndivPov, data =
district)
Variables actually used in tree construction:
 [1] "pctWhite"       "pctMale"        "pctNativeBorn"
"avgHouseSize"   "pctAge65"
 [6] "pctVet"         "medHouseIncome" "pctUnemp"
"pctUnins"       "pctBlack"
[11] "pctHisp"        "pctFamPov"
Number of terminal nodes:  26
Residual mean deviance:  0.3257 = 133.6 / 410
Misclassification error rate: 0.08257 = 36 / 436

> sniptree = snip.tree(basetree,nodes=c(2,12,216))
> summary(sniptree)

Classification tree:
snip.tree(tree = basetree, nodes = c(2L, 12L, 216L))
Variables actually used in tree construction:
 [1] "pctWhite"      "pctNativeBorn" "avgHouseSize"  "pctVet"
"pctMale"
 [6] "pctAge65"      "pctUnemp"      "pctUnins"      "pctBlack"
"pctHisp"
[11] "pctFamPov"
Number of terminal nodes:  22
Residual mean deviance:  0.3882 = 160.7 / 414
Misclassification error rate: 0.08257 = 36 / 436
```

From the result, basetree has 26 leaves. Snipped tree has 22 leaves. The observed misclassification rate is 0.08257.

4.(b)

```
> bictree = prune.tree(sniptree,k=2*log(436))
> summary(bictree)

Classification tree:
snip.tree(tree = sniptree, nodes = c(58L, 13L, 59L))
```

```
Variables actually used in tree construction:
[1] "pctWhite"      "pctNativeBorn" "avgHouseSize"  "pctUnins"
"pctBlack"
[6] "pctHisp"
Number of terminal nodes:  9
Residual mean deviance:  0.6539 = 279.2 / 427
Misclassification error rate: 0.1674 = 73 / 436
> deviance(bictree)
[1] 279.1951
> dimension = 9*2-1
> dimension
[1] 17
> bictree.BIC = deviance(bictree)+log(436)*dimension
> bictree.BIC
[1] 382.515
> plot(bictree);text(bictree)
```

From the result, it has 9 leaves. The deviance is 279.1951 and dimension is 17. BIC is 382.515.

4.(c)
The observed misclassification rate is 0.1674. It has following 6 variables:
pctWhite, pctNativeBorn, avgHouseSize, pctUnins, pctBlack and pctHisp.
pctWhite is the most prominent among them based on the graph.