# Mini Project #3

Names of group members: Junmei Fan, Xi Cui
Contribution of each group member: 100% for Junmei Fan, 100% for Xi Cui

## Question1

**a)** The full logistic model is fitted as below:

```
> fit.full<-gl(chd~sbp+tobacco+ldl+adiposity+famhist+typea+obesity+alcohol+age, family = binomial, data = SAheart)
> summary.glm(fit.full)
Call:
glm(formula = chd ~ sbp + tobacco + ldl + adiposity + famhist +
   typea + obesity + alcohol + age, family = binomial, data = SAheart)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.7781  -0.8213  -0.4387   0.8889   2.5435

Coefficients:
                  Estimate     Std. Error    z value   Pr(>|z|)
(Intercept)    -6.1507209  1.3082600  -4.701 2.58e-06 ***
sbp             0.0065040  0.0057304   1.135 0.256374
tobacco         0.0793764  0.0266028   2.984 0.002847 **
ldl             0.1739239  0.0596617   2.915 0.003555 **
adiposity       0.0185866  0.0292894   0.635 0.525700
famhistPresent  0.9253704  0.2278940   4.061 4.90e-05 ***
typea           0.0395950  0.0123202   3.214 0.001310 **
obesity        -0.0629099  0.0442477  -1.422 0.155095
alcohol         0.0001217  0.0044832   0.027 0.978350
age             0.0452253  0.0121298   3.728 0.000193 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)
   Null deviance: 596.11  on 461  degrees of freedom
   Residual deviance: 472.14  on 452  degrees of freedom
AIC: 492.14

Number of Fisher Scoring iterations: 5
```

The p values for the following predictors: sbp, adiposity, obesity and alcohol are large. So there is no clear evidence of a real association between chd and sbp(or adiposity, obesity or alcohol).

The hypothesis test is as below:

```
> fit.re<-glm(chd~tobacco+ldl+famhist+typea+age,family=binomial,data=SAheart)
> anova(fit.re,fit.full,test = "Chisq") #Note: The dropped predictors are not significant
Analysis of Deviance Table

Model 1: chd ~ tobacco + ldl + famhist + typea + age
Model 2: chd ~ sbp + tobacco + ldl + adiposity + famhist + typea + obesity +
   alcohol + age
   Resid. Df Resid. Dev  Df      Deviance Pr(>Chi)
1     456    475.69
2     452    472.14       4          3.5455   0.471
```

Hypothesis test:
H0: The coefficients for sbp, adiposity, obesity and alcohol are all zero.
Ha: at least one of the coefficients for sbp, adiposity, obesity and alcohol is not zero at significance level of 5%.

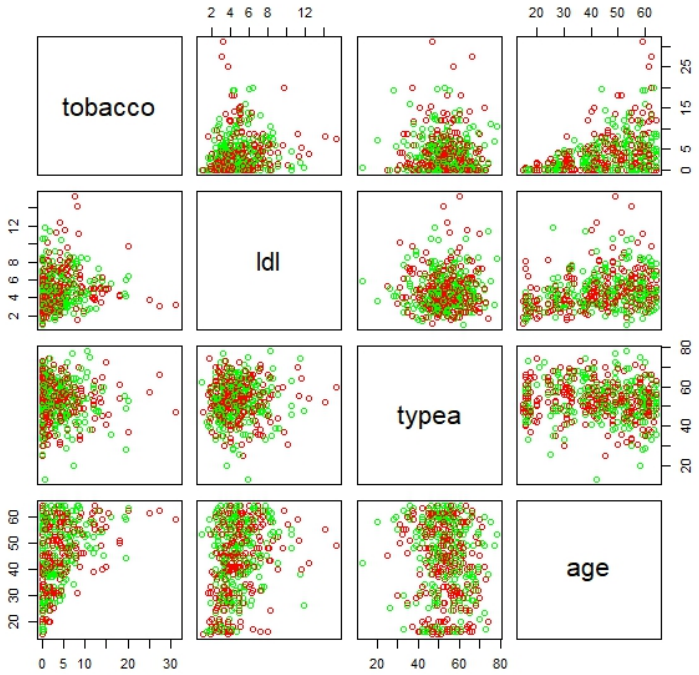
Test statistics: $F^*=\frac{3.5455/4}{472.14/452}\sim F(0.05, 4, 452)$, if $F^* > F(0.05, 4, 452)$ or p-value<0.05, then reject H0, otherwise fail to reject null hypothesis.
Conclusions: Since p-value 0.471>0.05, so we fail to reject H0 and conclude that the following 4 predictors: sbp, adiposity, obesity, and alcohol are not associated with chd.

```
> fit.re$coefficients
  (Intercept)      tobacco          ldl     famhistPresent       typea         age
-6.44644451   0.08037533   0.16199164   0.90817526   0.03711521   0.05046038
```

The logistic regression coefficients give the change in the log odds of the outcome for a one unit increase in the predictor variable.

- For every one unit change in tobacco while keeping all other predictors constant, the log odds of chd (presence of coronary heart disease versus absence of coronary heart disease) increases by 0.08037533.
- For additional one unit increase in ldl, the log odds of presence of coronary heart disease increases by 0.16199164.
- For additional one unit increase in typea, the log odds of presence of coronary heart disease increases by 0.03711521.
- The indicator variables for famhistPresent have a slightly different interpretation. Having family history of coronary heart disease, versus no family history of coronary heart disease, changes the log odds of chd by 0.90817526.

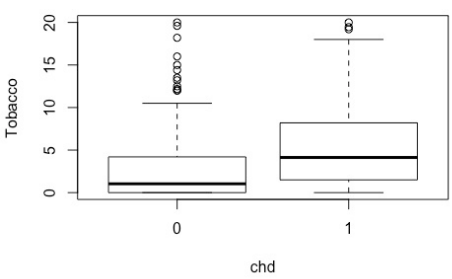
**b)**

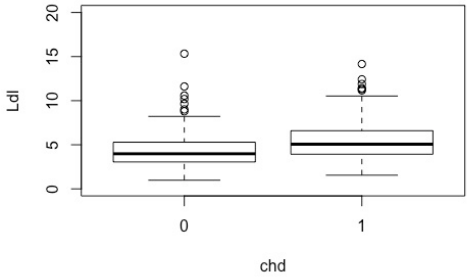


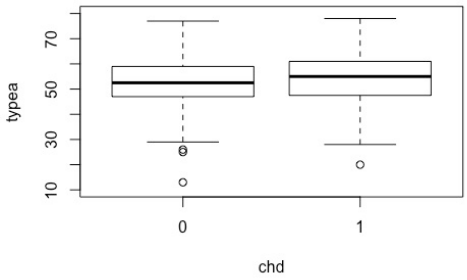

From the scatter plot, points belong to different class of "chd" are overlapped, it seems that neither of these four predictors can strongly predict the class of "chd".

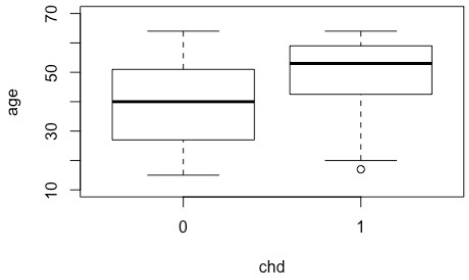
> cor(SAheart2)

| | tobacco | ldl | typea | age | chd |
|---|---|---|---|---|---|
| tobacco | 1.00000000 | 0.15890546 | -0.01460788 | 0.4503302 | 0.2997175 |
| ldl | 0.15890546 | 1.00000000 | 0.04404758 | 0.3117992 | 0.2630527 |
| typea | -0.01460788 | 0.04404758 | 1.00000000 | -0.1026063 | 0.1031558 |
| age | 0.45033016 | 0.31179923 | -0.10260632 | 1.0000000 | 0.3729733 |
| chd | 0.29971754 | 0.26305268 | 0.10315583 | 0.3729733 | 1.0000000 |




From the correlation coefficients and boxplots, we know that all four of the predictors have some positive correlations with chd. Probably age has the most correlation with chd, tobacco and ldl are the second and typea has the least correlation with chd.

**c)**Fit the logistic regression model:

```
glm(formula = chd ~ tobacco + ldl + typea + age, family = binomial,
   data = SAheart)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-2.0444  -0.8718  -0.4645   0.9741   2.5118

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.334452   0.897809  -7.055 1.72e-12 ***
tobacco      0.075031   0.025699   2.920  0.00350 **
ldl          0.179891   0.055027   3.269  0.00108 **
typea        0.037914   0.011885   3.190  0.00142 **
age          0.055040   0.009948   5.533 3.15e-08 ***
---
```

(1) Logistic regression decision boundary equation:

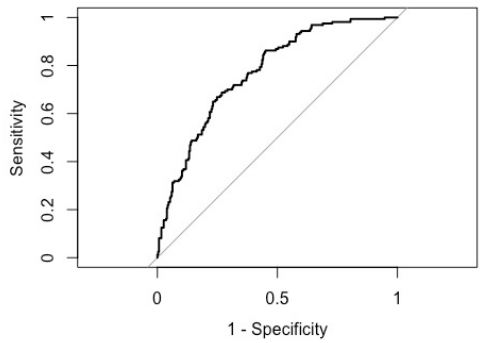
$\{x: \hat{p}(x) = 0.5\}$ or beta0 + beta1 *tobacco + beta2 * ldl+beta3 * typea+beta4 * age = 0

=> -6.334452+0.075031*tobacco + 0.179891* ldl+0.037914 * typea+0.055040 * age = 0

(2) Confusion matrix:

| lr.pred | True chd 0 | 1 |
|---|---|---|
| 0 | 260 | 85 |
| 1 | 42 | 75 |

(3) Sensitivity: 0.4687500; Specificity: 0.8609272. So this logistic model set the high threshold to detect chd, although specificity is relatively high, this model might miss a lot of positive chd cases.
(4) Overall misclassification rate: 0.2748918
(5) ROC plot:

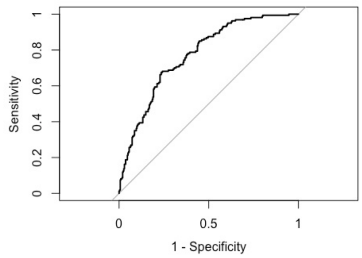



(6) Cross-validation error rate:
> cv.est [1] 0.2835338
The test error rate using 10-fold cross-validation is 0.2835338.

**d)**Fit the LDA model
(1) Decision boundary equation:
-0.0893448*tobacco -0.1879042* ldl-0.03371824 * typea-0.04818174 * age = 5.872758
(2) Confusion matrix:

| pred | Ture chd 0 | 1 |
|---|---|---|
| 0 | 257 | 87 |
| 1 | 45 | 73 |

(3) Sensitivity: 0.4562500; Specificity: 0.8509934, sensitivity and specificity are both worth than logistic model.
(4) Overall misclassification rate: 0.2857143
(5) ROC plot:




(6) Cross-validation error rate:
> cv.err.lda [1] 0.2943571
The test error rate using 10-fold cross-validation is 0.2943571.

**e)** Fit the QDA model:
(1) Decision boundary equation:

$$-\frac{1}{2}x^T\begin{bmatrix}0.0630 & -0.0256 & -0.0039 & -0.0048\\ -0.0256 & 0.1134 & -0.0003 & -0.0044\\ -0.0039 & -0.00027 & 0.0012 & -0.0008\\ -0.0048 & -0.0044 & -0.00084 & -0.0041\end{bmatrix}x+\begin{bmatrix}-0.3824\\ 0.0572\\ -0.0284\\ -0.3373\end{bmatrix}x=13.1587$$

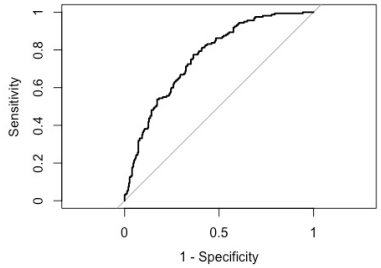
(2) Confusion matrix:

| pred | Ture chd 0 | 1 |
|---|---|---|
| 0 | 261 | 88 |
| 1 | 41 | 72 |

(3) Sensitivity: 0.4500000; Specificity: 0.8642384, comparing to logistic and LDA model, specificity increased a tiny bit.

(4) Overall misclassification rate: 0.2792208

(5) ROC plot:




(6) Cross-validation error rate:

> cv.err.qda [1] 0.2898242

The test error rate using 10-fold cross-validation is 0.2898242.

**f)** Fit the KNN model:

(1) Find the optimal K=18.

```
> result[err.knn == min(result$err.knn), ]
  ks   err.knn
18 18 0.2834875
```

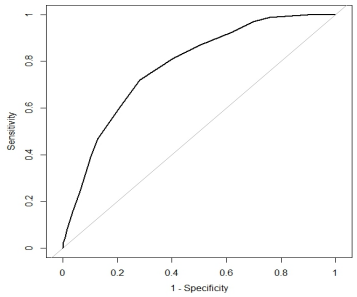
(2) Confusion matrix:

| knn.fit | Ture chd 0 | 1 |
|---|---|---|
| 0 | 267 | 93 |
| 1 | 35 | 67 |

(3) Sensitivity: 0.418750; Specificity: 0.884106 specificity increased but sensitivity decreased even more.

(4) Overall misclassification rate: 0.2770563

(5) ROC plot:

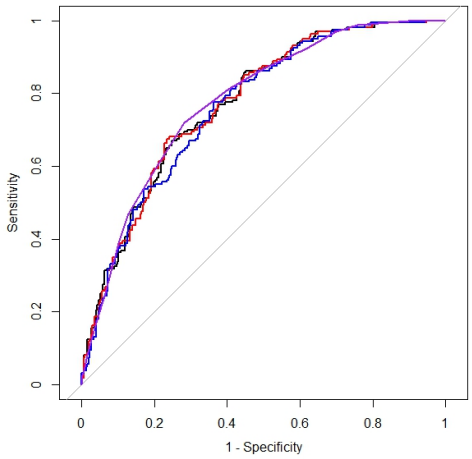



(6) Cross-validation error rate: The test error rate using 10-fold cross-validation is 0.2834875.

**g)**

Comparation of four models:

ROC curve:




AUC:

| | AUC |
|---|---|
| Logistic Regression | 0.7719 |
| LDA | 0.7721 |
| QDA | 0.7641 |
| KNN | 0.779 |

Test error rate by 10folds cross-validation:

| | |
|---|---|
| Logistic Regression | 0.2791859 |
| LDA | 0.2943571 |
| QDA | 0.2898242 |
| KNN | 0.2834875 |

The difference between these four models is small for this "SAheart" dataset. ROC curve and AUC shows the specificity and sensitivity for models, the performances of these four models are similar. However, the cross-validation error rate shows the logistic regression has the lowest test error rate. The response value we aimed to predict is a binary class, the logistic regression perform better is reasonable, so we recommend the logistic regression to do classification for this dataset.

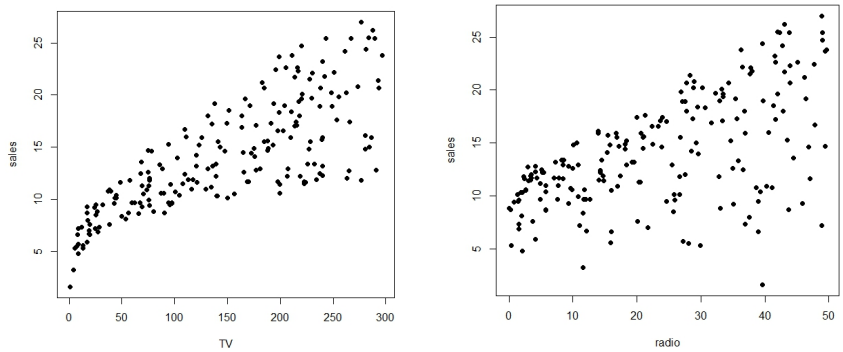
**Question2**

Fit the QDA by using 10 folds cross-validation:
Test error rate for Model1:"Derection~Lag1+Lag2": 0.4824
Test error rate for Model2:"Derection~Lag1+Lag2+Lag3+Lag4+Lag5+Volume": 0.5072
The test error rate is higher after adding "Lag3, Lag4, Lag5, Volume" to the model, which means the additional predictors could not improve the predictive power of QDA.

T-test of determining the difference of the 10 folds error rate:
H0: the error rate of Model 1 and Model 2 is equal.
Ha: the error rate of Model 1 and Model 2 is not equal. (True difference in means is not equal to 0)
data: cv.err.qda1.10 and cv.err.qda2.10
Test statistics: t = 0.7102 p-value = 0.4881
Conclusion: P-value >0.05, failed to reject H0, and we conclude that the error rate of these two models are the same, the additional predictors don't give any help for QDA.

**Question3**

(1) Scatter plot:




The scatter plot shows a linear relationship between sales and TV, also sales and radio. The slop shows their positive relationship, and the linear relationship is more obvious for sale and TV.

(2) Point Estimate of Population correlation:
Sales and TV => Rho1: 0.7822244
Sales and radio=> Rho2: 0.5762226
(3) Bootstrap Estimates
Sales and TV: Original= 0.7822244; Bias=-0.0001563068;
Standard error of point estimate=0.02689515
Sales and radio: Original = 0.5762226; Bias=0.0002223246;
Standard error of point estimate=0.05243057
(4) 95% confidence interval:
Sales and TV: ( 0.7283, 0.8305 )
Sales and radio: ( 0.4734, 0.6748 )

The original point estimates of Rho1 and Rho2 are included in the 95% confidence interval from bootstrap, so the bootstrap method here could reliably predict point estimation and its distribution in the population.

R-Code:

```
install.packages("ElemStatLearn")
library(ElemStatLearn)

#data prepared
attach(SAheart)
summary(SAheart)
SAheart$chd<-as.factor(SAheart$chd)

#1.a
fit.full<-glm(chd~sbp+tobacco+ldl+adiposity+famhist+typea+obesity+alcohol+age, family = binomial, data = SAheart)
summary.glm(fit.full)
fit.re<-glm(chd~tobacco+ldl+famhist+typea+age,family=binomial,data=SAheart)
#hypothesis test
anova(fit.re,fit.full,test = "Chisq") #Note: The dropped predictors are not significant
#coefficients
summary.glm(fit.re)

#b
#scatter plot
train<-SAheart[,c(2,3,6,9)]
plot(train,col=(c("red","green")))

SAheart2=SAheart[,c(2,3,6,9,10)]
str(SAheart2)
#pair correlation
cor(chd)
#plot correlation
pairs(SAheart2)
boxplot(SAheart2[, "tobacco"] ~ SAheart2$chd, ylab = "Tobacco",xlab = "chd", ylim = c(0, 20))
boxplot(SAheart$ldl ~ SAheart2$chd, ylab = "Ldl", xlab = "chd",ylim = c(0, 20))
boxplot(SAheart$typea ~ SAheart2$chd, ylab = "typea", xlab = "chd",ylim = c(10, 80))
boxplot(SAheart$age ~ SAheart2$chd, ylab = "age", xlab = "chd",ylim = c(10, 70))
par(mfrow = c(1, 1))

#c logistic regression model
fit<-glm(chd~tobacco+ldl+typea+age,family=binomial,data=SAheart)
lr.prob <- predict(fit,SAheart, type = "response")
#classification
lr.pred <- ifelse(lr.prob >= 0.5, "1", "0")
#confusion matrix
```

```
table(lr.pred, SAheart[, "chd"])
#sensitivity and specificity
cm<-matrix(table(lr.pred, SAheart[, "chd"]),nrow=2,ncol = 2,byrow=FALSE)
c(cm[2,2]/sum(cm[,2]),cm[1,1]/sum(cm[,1]))
#overall misclassification rate
1 - mean(lr.pred == SAheart[, "chd"])
#ROC curve
install.packages('pROC')
library(pROC)
roc.lr <- roc(SAheart[, "chd"], lr.prob, levels = c("0", "1"))
plot(roc.lr, legacy.axes = T)

#10 fold cross validation (Method2)
#Create 10 equally size folds
SAheart<-SAheart[sample(nrow(SAheart)),]
folds <- cut(seq(1,nrow(SAheart)),breaks=10,labels=FALSE)
K<-10
cv.err.lr.10<-sapply(1:K, FUN=function(i){
  #Segement your data by fold using the which() function
  testIndexes <- which(folds==i,arr.ind=TRUE)
  testData <- SAheart[testIndexes, ]
  trainData <- SAheart[-testIndexes, ]
  train.X<-trainData[,c(2,3,6,9)]
  train.y<-trainData$chd
  test.X<-testData[,c(2,3,6,9)]
  test.y<-testData[,c(10)]
  lr.fit<-glm(chd~tobacco+ldl+typea+age, family=binomial, data=trainData)
  # Estimated probabilities for test data
  lr.prob <- predict(lr.fit, testData, type = "response")

  # Predicted classes (using 0.5 cutoff)
  lr.pred <- ifelse(lr.prob >= 0.5, "1", "0")
  cv.test <-1-sum(lr.pred == test.y)/length(test.y)
  return(cv.test)}
)
cv.err.lr=mean(cv.err.lr.10)
cv.err.lr

#d,e
train.X<-SAheart[,c(2,3,6,9)]
train.y<-SAheart[,c(10)]
#LDA
#find decision boundary
s6340.lda <- function(y, X) {
  # y = training data response vector (a factor)
  # X = training data predictor matrix
  N <- length(y) # no of observations
  K <- nlevels(y) # no of classes
  p <- ncol(X) # no of predictors
  n <- as.numeric(table(y)) # class frequencies
  names(n) <- levels(y)
  pi <- n/N # class proportions
  # mean vector
  mu <- matrix(unlist(by(X, y, colMeans)), byrow = T, ncol = p)
  rownames(mu) <- levels(y)
  colnames(mu) <- colnames(X)
  # pooled covariance matrix
  S <- by(X, y, cov)
  Sigma <- Reduce("+", lapply(1:K, FUN = function(k) {(n[k] - 1) * S[[k]]}))/(N - K)
  # its inverse
  Sigma.inv <- solve(Sigma)
  # delta functions
```

```
  delta <- t(sapply(1:K, FUN = function(k) {
    c(-(1/2) * drop(t(mu[k, ]) %*% Sigma.inv %*% mu[k, ]) +
        log(pi[k]), t(mu[k, ]) %*% Sigma.inv)}))
  rownames(delta) <- levels(y)
  colnames(delta) <- c("(Intercept)", colnames(X))
  # pairwise difference of delta functions
  idx.pair <- combn(K, 2)
  delta.diff <- t(apply(idx.pair, MAR = 2, FUN = function(pair) {
    delta[pair[1], ] - delta[pair[2], ]}))
  rownames(delta.diff) <- apply(idx.pair, MAR = 2, FUN = function(pair) {
    paste0(levels(y)[pair[1]], "-", levels(y)[pair[2]])})
  # multiply intecept difference by 1 to get the cutoff c
  delta.diff[, 1] <- -delta.diff[, 1]
  colnames(delta.diff)[1] <- "Cutoff"
  # result
  result <- list(N = N, n = n, pi = pi, mu = mu, Sigma = Sigma,
                 delta = delta, disc = delta.diff)
  return(result)
}
our.lda.fit <- s6340.lda(train.y,train.X)
our.lda.fit$disc

#fit by package
library(MASS)
lda.fit <- lda(chd~tobacco+ldl+typea+age,data=SAheart)
lda.pred<- predict(lda.fit, SAheart)
#fit by function
train.X<-as.matrix(as.data.frame(train.X))
coeff <- our.lda.fit$disc[1, -1]
coeff
cutoff <- our.lda.fit$disc[1, 1]
score.test <- train.X %*% coeff
#predict class
pred.test <- ifelse(score.test >= cutoff, "0", "1")
#confusion matrix
table(pred.test, train.y)
#sensitivity and specificity
cm<-matrix(table(pred.test, train.y),nrow=2,ncol = 2,byrow=FALSE)
c(cm[2,2]/sum(cm[,2]),cm[1,1]/sum(cm[,1]))
#overall misclassification rate
1 - mean(pred.test == SAheart[, "chd"])
#ROC curve
lda.prob<-lda.pred$posterior[,2]
roc.lda <- roc(SAheart[, "chd"], lda.prob, levels = c("0", "1"))
plot(roc.lda, legacy.axes = T)
#k=10 folds cross-validation
SAheart<-SAheart[sample(nrow(SAheart)),]
cv.err.lda.10<-sapply(1:K, FUN=function(i){
  testIndexes <- which(folds==i,arr.ind=TRUE)
  testData <- SAheart[testIndexes, ]
  trainData <- SAheart[-testIndexes, ]
  train.X<-trainData[,c(2,3,6,9)]
  train.y<-trainData$chd
  test.X<-testData[,c(2,3,6,9)]
  test.y<-testData$chd
  lda.fit<-lda(chd~tobacco+ldl+typea+age, data=trainData)
  lda.pred<-predict(lda.fit,test.X)
  cv.est<-mean(lda.pred$class==test.y)
  return(cv.est)}
)
cv.err.lda=1-mean(cv.err.lda.10)
cv.err.lda
```

```
#QDA
#find decision boundary
s6340.qda <- function(y, X) {
  # y = training data response vector (a factor)
  # X = training data predictor matrix
  N <- length(y) # no of observations
  K <- nlevels(y) # no of classes
  p <- ncol(X) # no of predictors
  n <- as.numeric(table(y)) # class frequencies
  names(n) <- levels(y)
  pi <- n/N # class proportions
  # mean vector
  mu <- matrix(unlist(by(X, y, colMeans)), byrow = T, ncol = p)
  rownames(mu) <- levels(y)
  colnames(mu) <- colnames(X)
  #covariance matrix
  S <- by(X, y, cov)
  # its inverse
  Sigma.inv <- lapply(1:K, FUN=function(k){solve(S[[k]])})
  # delta functions
  delta <- t(sapply(1:K, FUN = function(k) {c(-(1/2) * drop(t(mu[k, ]) %*% Sigma.inv[[k]] %*% mu[k, ]) +
       log(pi[k])-(1/2)*log(drop(det(S[[k]]))),drop(t(mu[k, ]) %*% Sigma.inv[[k]]),drop(Sigma.inv[[k]]))}))

  rownames(delta) <- levels(y)
  idx.pair <- combn(K, 2)
  delta.diff <- t(apply(idx.pair, MAR = 2, FUN = function(pair) {
    delta[pair[1], ] - delta[pair[2], ]}))
  #rownames(delta.diff) <- apply(idx.pair, MAR = 2, FUN = function(pair) {
  #  paste0(levels(y)[pair[1]], "-", levels(y)[pair[2]])})
  # multiply intecept difference by 1 to get the cutoff c
  delta.diff[, 1] <- -delta.diff[, 1]
  colnames(delta.diff)[1] <- "say"
  result <- list(N = N, n = n, pi = pi, mu = mu, Sigma = S,
                 delta = delta,disc = delta.diff)
  return(result)
}
our.qda.fit <- s6340.qda(train.y, train.X)
our.qda.fit$disc[,c(1:5)]
sigma1_sigma2<-matrix(our.qda.fit$disc[,c(6:21)],nrow=4,ncol=4,byrow = TRUE)
sigma1_sigma2

#fit qda model and predict class
qda.fit <- qda(chd~tobacco+ldl+typea+age,data=SAheart)
qda.pred<- predict(qda.fit, SAheart)
#confusion matrix
table(qda.pred$class, SAheart[, "chd"])
#sensitivity and specificity
cm<-matrix(table(qda.pred$class, SAheart[, "chd"]),nrow=2,ncol = 2,byrow=FALSE)
c(cm[2,2]/sum(cm[,2]),cm[1,1]/sum(cm[,1]))
#overall misclassification rate
1 - mean(qda.pred$class == SAheart[, "chd"])
#ROC curve
qda.prob<-qda.pred$posterior[,2]
roc.qda <- roc(SAheart[, "chd"], qda.prob, levels = c("0", "1"))
plot(roc.qda, legacy.axes = T)
#10fold cross validation for qda
SAheart<-SAheart[sample(nrow(SAheart)),]
#Create 10 equally size folds
folds <- cut(seq(1,nrow(SAheart)),breaks=10,labels=FALSE)
K<-10
cv.err.qda.10<-sapply(1:K, FUN=function(i){
```

```
  #Segement your data by fold using the which() function
  testIndexes <- which(folds==i,arr.ind=TRUE)
  testData <- SAheart[testIndexes, ]
  trainData <- SAheart[-testIndexes, ]
  #Use the test and train data partitions however you desire...
  train.X<-trainData[,c(2,3,6,9)]
  train.y<-trainData$chd
  test.X<-testData[,c(2,3,6,9)]
  test.y<-testData[,c(10)]
  qda.fit<-qda(chd~tobacco+ldl+typea+age, data=trainData)
  qda.pred<-predict(qda.fit,test.X)
  cv.est<-mean(qda.pred$class == test.y)
  return(cv.est)}
)
cv.err.qda=1-mean(cv.err.qda.10)
cv.err.qda

#f
#10fold cross validation
library(boot)
library(class)
#SAheart<-SAheart[sample(nrow(SAheart)),]
#Create 10 equally size folds
folds <- cut(seq(1,nrow(SAheart)),breaks=10,labels=FALSE)
#Perform 10 fold cross validation

  # Fit KNN for several values of K

  ks <- c(seq(1, 50, by = 1), seq(40, 200, by = 10))

  nks <- length(ks)
  err.rate.train <- numeric(length = nks)
  err.rate.test <- numeric(length = nks)
  err.knn<-numeric(length = nks)
  names(err.rate.train) <- names(err.rate.test) <- ks

  for (j in seq(along = ks)) {

    cv.err.knn.10<-sapply(1:10, FUN=function(i){
      #Segement your data by fold using the which() function
      testIndexes <- which(folds==i,arr.ind=TRUE)
      testData <- SAheart[testIndexes, ]
      trainData <- SAheart[-testIndexes, ]
      #Use the test and train data partitions however you desire...
      train.X<-trainData[,c(2,3,6,9)]
      train.y<-trainData$chd
      test.X<-testData[,c(2,3,6,9)]
      test.y<-testData$chd

    set.seed(1)
    mod.train <- knn(train.X, train.X, train.y, k = ks[j])
    set.seed(1)
    mod.test <- knn(train.X, test.X, train.y, k = ks[j])

    err.rate.train[j] <- 1 - sum(mod.train == train.y)/length(train.y)
    err.rate.test[j] <- 1 - sum(mod.test == test.y)/length(test.y)
    return (err.rate.test[j])
  })
err.knn[j]<-(mean(cv.err.knn.10))
  #result <- data.frame(ks, err.rate.train, err.rate.test)
  }
  result <- data.frame(ks, err.knn)
```

```
result[err.knn == min(result$err.knn), ]

# Optimal KNN (K = 18) --- from 10 fold cross validation
train.X<-SAheart[,c(2,3,6,9)]
train.y<-SAheart$chd
library(class)
set.seed(1)
knn.fit <- knn(train.X, train.X, train.y, k = 18, prob = T)

knn.prob <- attr(knn.fit, "prob") # prob is voting fraction for winning class
knn.prob <- ifelse(knn.fit == "1", knn.prob, 1 - knn.prob) # now it is voting fraction for Direction == "1"

#confusion matrix
table(knn.fit, SAheart[, "chd"])
#sensitivity and specificity
cm<-matrix(table(knn.fit, SAheart[, "chd"]),nrow=2,ncol = 2,byrow=FALSE)
c(cm[2,2]/sum(cm[,2]),cm[1,1]/sum(cm[,1]))
#misclassification rate
1 - mean(knn.fit == train.y)
#roc plot for knn
roc.knn <- roc(train.y, knn.prob, levels = c("0", "1"))
plot(roc.knn,add = T,legacy.axes = T)

#g
plot(roc.lr, legacy.axes = T)
plot(roc.lda, add = T, col="red")
plot(roc.qda, add = T, col = "blue")
plot(roc.knn,add = T,col = "purple")
auc(roc.lr)
auc(roc.lda)
auc(roc.qda)
auc(roc.knn)

#Q2
library(ISLR)
attach(Smarket)

#QDA for Derection~Lag1+Lag2
qda1.fit <- qda(Direction ~ Lag1 + Lag2, data = Smarket)
qda1.pred<-predict(qda1.fit, Smarket)
1 - mean(qda1.pred$class == Smarket[, "Direction"])
qda2.fit <- qda(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data=Smarket)
qda2.pred<-predict(qda2.fit, Smarket)
1 - mean(qda2.pred$class == Smarket[, "Direction"])
#10 fold cross-validation
Smarket<-Smarket[sample(nrow(Smarket)),]
folds <- cut(seq(1,nrow(Smarket)),breaks=10,labels=FALSE)
K<-10
#QDA for Derection~Lag1+Lag2
cv.err.qda1.10<-sapply(1:K, FUN=function(i){
  testIndexes <- which(folds==i,arr.ind=TRUE)
  testData <- Smarket[testIndexes, ]
  trainData <- Smarket[-testIndexes, ]
  train.X<-trainData[,c("Lag1","Lag2")]
  train.y<-trainData$Direction
  test.X<-testData[,c("Lag1","Lag2")]
  test.y<-testData$Direction
  qda.fit <- qda(Direction ~ Lag1 + Lag2, data=trainData)
  qda.pred<-predict(qda.fit,test.X)
  cv.est<-mean(qda.pred$class==test.y)
  return(cv.est)}
)
```

```
1-mean(cv.err.qda1.10)
#QDA for Derection~Lag1+Lag2+Lag3+Lag4+Lag5+Volume
cv.err.qda2.10<-sapply(1:K, FUN=function(i){
  testIndexes <- which(folds==i,arr.ind=TRUE)
  testData <- Smarket[testIndexes, ]
  trainData <- Smarket[-testIndexes, ]
  train.X<-trainData[,c("Lag1","Lag2","Lag3","Lag4","Lag5","Volume")]
  train.y<-trainData$Direction
  test.X<-testData[,c("Lag1","Lag2","Lag3","Lag4","Lag5","Volume")]
  test.y<-testData$Direction
  qda.fit <- qda(Direction ~ Lag1 + Lag2 + Lag3+Lag4+Lag5+Volume, data=trainData)
  qda.pred<-predict(qda.fit,test.X)
  cv.est<-mean(qda.pred$class==test.y)
  return(cv.est)}
)
1-mean(cv.err.qda2.10)
#t test
t.test(cv.err.qda1.10,cv.err.qda2.10)

#Q3
install.packages('boot')
library(boot)
#Data preparing
ad<-read.table("C:/Users/xicui/Desktop/Advertising.csv", header = TRUE, sep=",")
plot(sales ~ TV, data = ad,pch=19)
plot(sales ~ radio, data = ad,pch=19)

#point estimate
rho1<-cor(ad$sales,ad$TV)
rho2<-cor(ad$sales,ad$radio)
c(rho1,rho2)
#bootstrap estimate
fit1.fn <- function(data, index) {
  data=data[index,]
  result<-cor(data$sales,data$TV)
  return(result)}
n <- nrow(ad)
fit1.fn(ad, 1:n)
reg1.boot <- boot(ad, fit1.fn, R = 1000)
reg1.boot

fit2.fn <- function(data, index) {
  data=data[index,]
  result<-cor(data$sales,data$radio)
  return(result)}
fit2.fn(ad, 1:n)
reg2.boot <- boot(ad, fit2.fn, R = 1000)
reg2.boot

#95% confidence interval
boot.ci(reg1.boot,type="perc")
boot.ci(reg2.boot,type="perc")
```