# Statistical Learning and Data mining

Midterm Examination

M052040003 鍾冠毅

1.a.

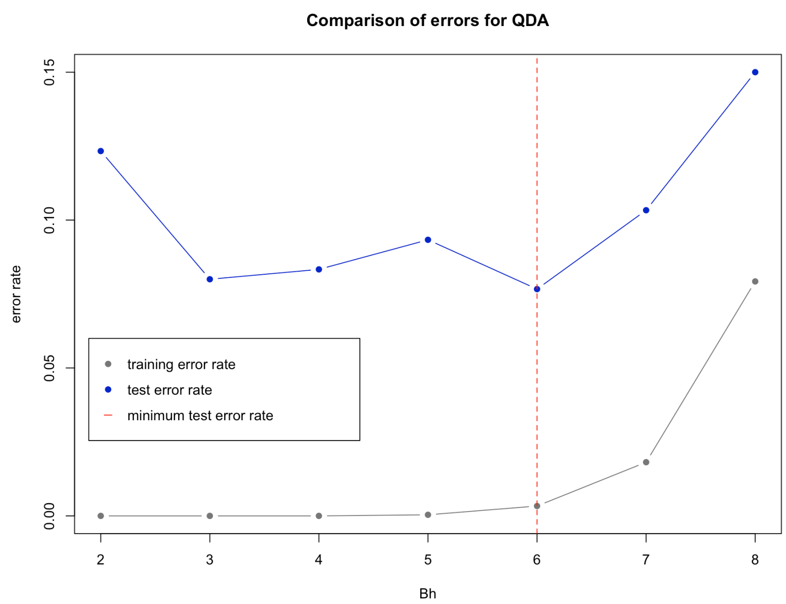
There are 185 features in B1. From the following table, we can observe that the training error increases in the B1 model from 0.07 to 0.086; yet, the test error decreases from 0.157 to 0.147. Since the test error rates of both models are larger than 10%, the LDA method seems not suitable of this data.

```
> err.compare
                original         B1
training error 0.0700000 0.08555556
test error     0.1566667 0.14666667
```

1.b.

In the QDA method, there is a term, $log|\Sigma_k|$, the logarithm of determinant of covariance matrix of the predictors of the response classes. If the value of the determinant equals 0, than its logarithm would be undefined. Thus, the original data and the B1 can't be derived QDA models. Yet, when we take B2 to B8, the determinants will be non-zero vector. The determinants are shown as the following table. The errors of  the QDA of B2 to B8 are shown in the following figure.

| | B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 | B8 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 1.903835e-238 | 3.840886e-187 | 4.671576e-119 | 9.474380e-60 |
| 2 | 0 | 0 | 3.965126e-301 | 8.390177e-246 | 1.604350e-210 | 1.680037e-169 | 5.176644e-135 | 7.653956e-89 | 4.618008e-44 |
| 3 | 0 | 0 | 4.179387e-305 | 1.013168e-247 | 3.844003e-216 | 1.159176e-173 | 1.948472e-136 | 4.978843e-90 | 2.112973e-46 |
| 4 | 0 | 0 | 2.155982e-316 | 3.773905e-256 | 2.385068e-224 | 1.516665e-178 | 2.078874e-141 | 3.016923e-94 | 2.467300e-48 |
| 5 | 0 | 0 | 3.217736e-306 | 1.997756e-247 | 1.120785e-214 | 3.021010e-172 | 2.601862e-136 | 4.915338e-91 | 2.730308e-47 |
| 6 | 0 | 0 | 0.000000e+00 | 7.960916e-275 | 9.044765e-232 | 8.028234e-186 | 5.277075e-147 | 3.580977e-96 | 3.048903e-49 |
| 7 | 0 | 0 | 0.000000e+00 | 8.056604e-296 | 3.595495e-253 | 1.517900e-192 | 1.065809e-151 | 4.962985e-100 | 3.885432e-53 |
| 8 | 0 | 0 | 1.538668e-297 | 1.723982e-235 | 1.525840e-204 | 1.957082e-160 | 4.572986e-126 | 7.475188e-83 | 3.710654e-43 |
| 9 | 0 | 0 | 0.000000e+00 | 9.001566e-268 | 1.557634e-230 | 1.511745e-178 | 3.841411e-140 | 3.147907e-93 | 1.126810e-48 |
| 10 | 0 | 0 | 1.392099e-318 | 8.524083e-263 | 1.077841e-230 | 2.269464e-189 | 1.502365e-153 | 1.388339e-105 | 1.149386e-58 |

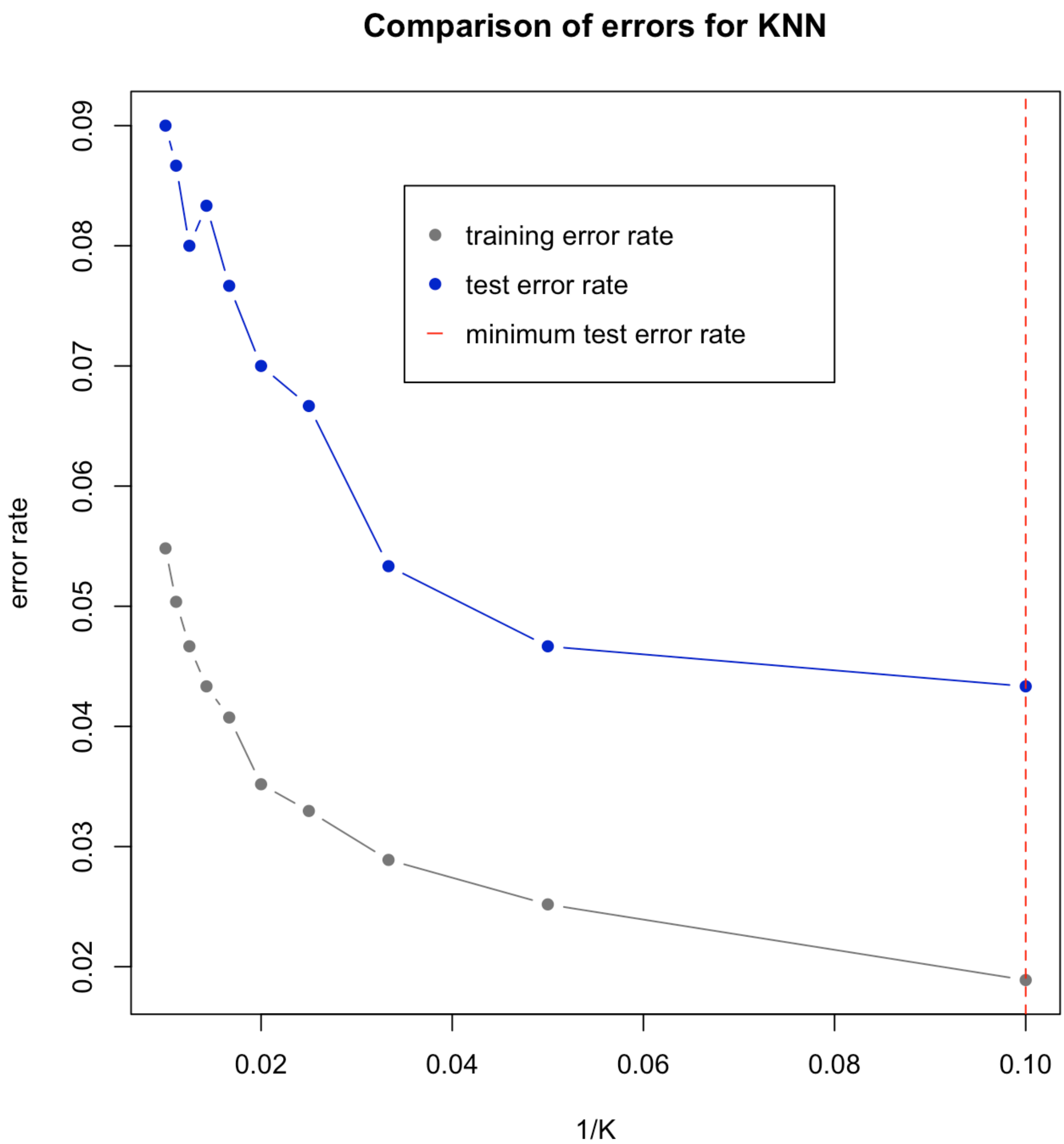


1.c.

The training error rate is 0.048. The test error rate is 0.267.

1.d.

The following figure shows that the test error rate reach its minimum while K is 10. Also, we can notice that the training error rate has its minimum at K=10 . It's reasonable that the flexibility goes up as the chosen K goes down.



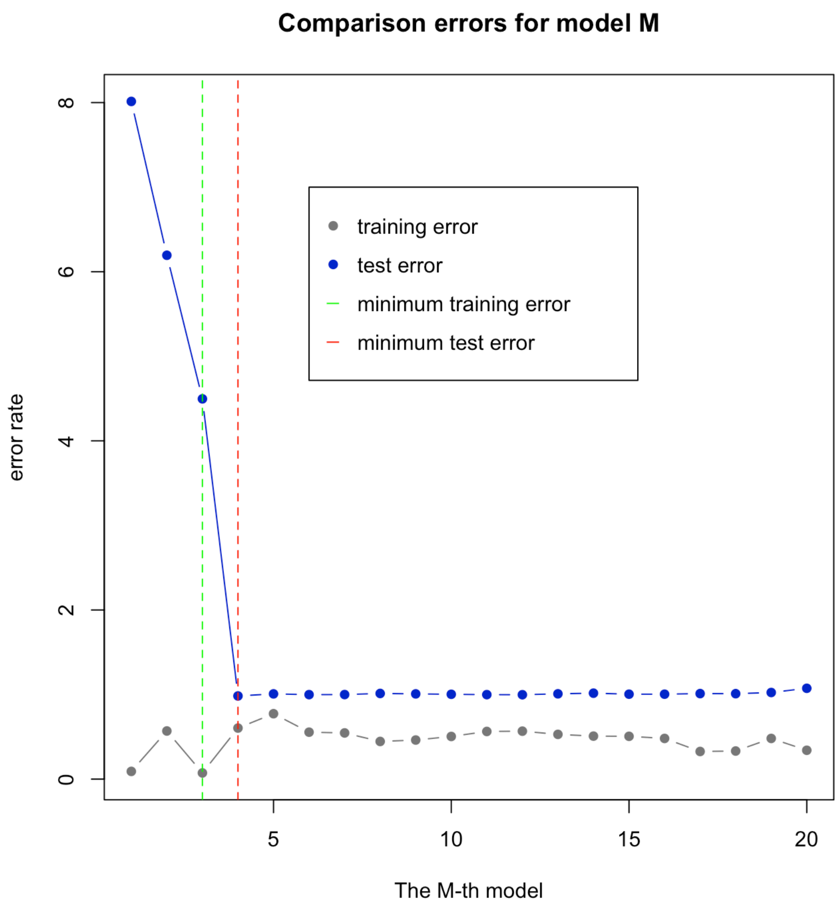

1.e.

In the QDA of B6, we can see that the test error rate is smallest among the models we built. Excluding the QDAs, the KNN method performs well as its minimum test error is smaller than 0.05.

2.a.

We can observe that the training error reach its minimum at the 3rd model. It's not reasonable because the more predictors a model contains, the less training error is produced. The 4th model is the most suitable due to its minimum test error.

```
> err.compare.2
        train      test
1  0.09133254 8.0137016
2  0.56923674 6.1945795
3  0.07305401 4.4966863
4  0.60374565 0.9835016
5  0.77353606 1.0080027
6  0.55517085 0.9991189
7  0.54617267 0.9998927
8  0.44545803 1.0129598
9  0.46228704 1.0079603
10 0.50447918 1.0031858
11 0.56348335 0.9990479
12 0.56705798 0.9979938
13 0.52930003 1.0086649
14 0.50847260 1.0163168
15 0.50626190 1.0044178
16 0.48113110 1.0044084
17 0.32689781 1.0113277
18 0.33226712 1.0103251
19 0.48118386 1.0242464
20 0.34120164 1.0740181
```




2.b.

The x1 and x8 are potential collinear predictors due to their VIF is larger than 10. We removed x8 for the model without x8 has the minimum test error, 1.04. Furthermore, this model improved the performance of model 20 with  test error, 1.07. Also, considering the model 1, we replace the response which' has the highest leverage. The percentage change of the slope estimator is −2.08%

```
> te.err.2.b
                  test error
model 20            1.074018
x1 removed          1.210301
x8 removed          1.039503
x1 and x8 removed  16.796160
```

3.a.

From following the summary table, x1, x7 and x10 significantly affect the model for their p-value lower than 0.05. The training error is 103.8187. The test error is 105.0835.

$$y = -35.56 + 0.03x_1 - 0.34x_2 + 1.09x_3 + 0.03x_4 - 0.38x_5 - 2.04x_6 + 39.69x_7 + -0.66x_8 - 3.09x_9(1) - 2.92x_9(2) - 0.09x_{10}$$

```
> summary(fit.3.a)

Call:
lm(formula = fmla, data = credit.train)

Residuals:
    Min      1Q  Median      3Q     Max 
-15.290  -6.952  -2.750   4.810  32.427 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)    -35.562615   6.696427  -5.311 4.26e-07 ***
x1               0.029681   0.006025   4.926 2.36e-06 ***
x2              -0.034301   0.088874  -0.386    0.700    
x3               1.090401   0.754971   1.444    0.151    
x4               0.027129   0.052355   0.518    0.605    
x5              -0.381480   0.287949  -1.325    0.187    
as.factor(x6)1  -2.038677   1.736931  -1.174    0.243    
as.factor(x7)1  39.693700   3.578362  11.093  < 2e-16 ***
as.factor(x8)1  -0.662897   1.723571  -0.385    0.701    
as.factor(x9)1  -3.086328   2.097064  -1.472    0.143    
as.factor(x9)2  -2.918328   2.379814  -1.226    0.222    
x10             -0.092220   0.004708 -19.590  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.19 on 138 degrees of freedom
Multiple R-squared:  0.8972,	Adjusted R-squared:  0.889 
F-statistic: 109.5 on 11 and 138 DF,  p-value: < 2.2e-16
```

3.b.

The training error and test error are 104.48 and 117.88, respectively. Model in (a) is more suitable for its smaller test error.

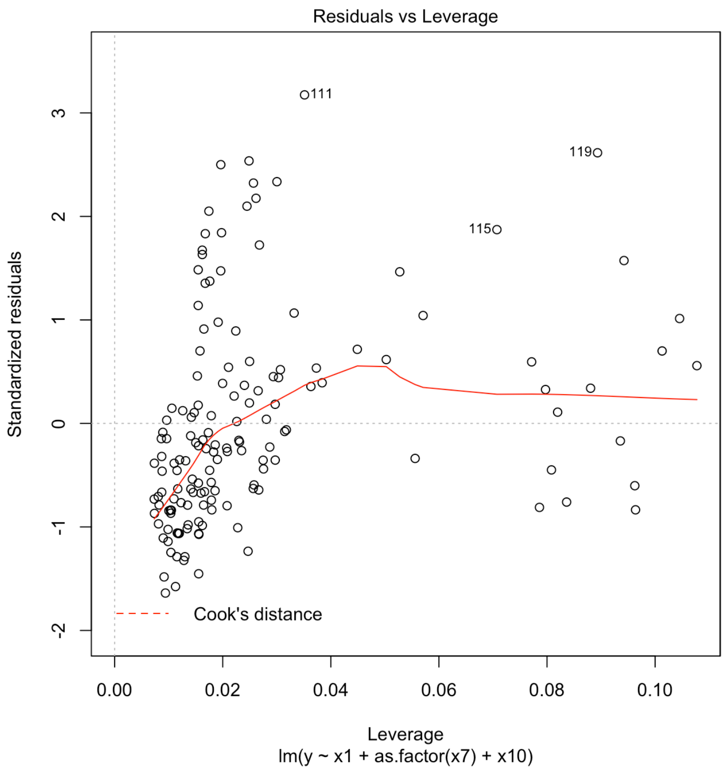
$$y = -40.85 + 0.03x_1 + 37.86x_7 - 0.09x_{10}$$

```
> confint(fit.3.b)
                     2.5 %      97.5 %
(Intercept)    -45.92443973 -35.78164874
x1               0.02540378   0.02887398
as.factor(x7)1  31.17627105  44.53746915
x10             -0.09835544  -0.08130796
```

```
> err.compare.3
        training error test error
model a       103.8187   105.0835
model b       104.4752   117.8766
```

3.c.

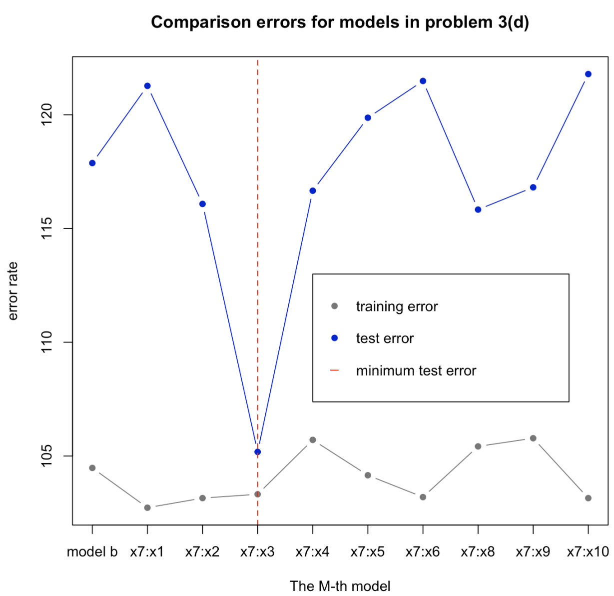
The standardized residual of sample 111 is larger than 3, so it is the outlier. There is no obvious high leverage shown on this plot.




3.d.

The model with interaction of x7 and x3 is the best one for its minimum test error among the nine models. Also it does improved the performance of model in (b).

```
> err.compare.3
             train     test
model a 103.8187 105.0835
model b 104.4752 117.8766
x7:x1   102.7279 121.2694
x7:x2   103.1494 116.0804
x7:x3   103.3188 105.1814
x7:x4   105.7079 116.6614
x7:x5   104.1531 119.8695
x7:x6   103.1939 121.4851
x7:x8   105.4229 115.8315
x7:x9   105.7812 116.8095
x7:x10  103.1486 121.7888
```



## Appendix (R code)

```
### 1 ###

# a #

# i

xnam <- paste0("x", 1:350)

fmla <- as.formula(paste("y ~", paste(xnam, collapse= "+")))

fit.1.a.i <- lda(fmla, data = disease.train)

tr.err.i <- length(disease.train[predict(fit.1.a.i, disease.train)$class != disease.train$y, 1])/2700

te.err.i <- length(disease.test[predict(fit.1.a.i, disease.test)$class != disease.test$y, 1])/300

# ii

choose <- 1:350

Bh <- function(h) choose[sapply(1:350, function(x) sum(abs(disease.train[,x ]))/ sum(abs(disease.train[, -351])) >= 0.001*h)]

B1 <- Bh(1)

length(B1)

# iii

xnam <- paste0("x", B1)

fmla <- as.formula(paste("y ~ ", paste(xnam, collapse= "+")))

fit.1.a.iii <- lda(fmla, data = disease.train)

tr.err.iii <- length(disease.train[predict(fit.1.a.iii, disease.train)$class != disease.train$y, 1])/2700

te.err.iii <- length(disease.test[predict(fit.1.a.iii, disease.test)$class != disease.test$y, 1])/300

# iv & v

err.compare <- data.frame(matrix(c(tr.err.i, te.err.i, tr.err.iii, te.err.iii), nrow = 2))

names(err.compare) <- c("original", "B1")

row.names(err.compare) <- c("training error", "test error")

err.compare



# b #

qda(fmla, data = disease.train[B.0.8[[4]]])

#  Error in qda.default(x, grouping, ...) :

#  some group is too small for 'qda'

B.0.8 <- sapply(0:8, function(h) Bh(h))

det.cov.m <- data.frame(matrix(NA, nrow = 10, ncol = 9))

for(h in 1:9){

  det.cov.m[, h] <- sapply(1:10, function(k) det(cov(disease.train[disease.train$y == k, B.0.8[[h]]] )))

}

names(det.cov.m) <- paste0("B", 0:8)



err.1.b <-

  sapply(2:8, function(x) {

    xnam <- paste0("x", Bh(x))

    fmla <- as.formula(paste("y ~ ", paste(xnam, collapse= "+")))

    fit.1.b <- qda(fmla, data = disease.train)

    c(length(disease.train[predict(fit.1.b, disease.train)$class != disease.train$y, 1])/2700,

      length( disease.test[predict(fit.1.b, disease.test)$class != disease.test$y, 1])/300)

  })

err.1.b <- data.frame(t(err.1.b))

err.1.b <- cbind(2:8, err.1.b)

names(err.1.b) <- c("Bh", "train", "test")

plot(err.1.b$train ~ err.1.b$Bh, type = "b", ylim = range(err.1.b[, 2:3]),

     xlab = "Bh", ylab = "error rate", main = "Comparison of errors for QDA",

     col = "gray47", pch = 16)

points(err.1.b$test  ~ err.1.b$Bh, type = "b", col = "blue3", pch = 16)

abline(v = err.1.b$Bh[which.min(err.1.b$test)], col = "red", lty = 2)

legend(x = 1.89, y = 0.06, legend = c("training error rate", "test error rate", "minimum test error rate"),

       pch = c(16,16,95), col = c("gray47", "blue3", "red"))



# c #

xnam <- paste0("x", choose);

fmla <- as.formula(paste("yi ~ ", paste(xnam, collapse= "+")))

predict.1.c.tr <- matrix(NA, nrow = 2700, ncol = 10)

predict.1.c.te <- matrix(NA, nrow = 300, ncol = 10)

for(i in 1:10){

    yi <- as.factor(ifelse(disease.train$y == i, 1, 0));

    fit.b <- glm(fmla, data = disease.train, family = "binomial");

    predict.1.c.tr[, i] <- predict(fit.b, disease.train, type = "response")

    predict.1.c.te[, i] <- predict(fit.b, disease.test , type = "response")

}

predict.1.c.tr <- as.data.frame(predict.1.c.tr)

class.1.c.tr <- apply(predict.1.c.tr, MARGIN = 1, function(x) which.max(x))

table(disease.train$y, class.1.c.tr)

tr.err.c <- 1 - sum(diag(table(disease.train$y, class.1.c.tr)))/2700
```

```
predict.1.c.te <- as.data.frame(predict.1.c.te)

class.1.c.te <- apply(predict.1.c.te, MARGIN = 1, function(x) which.max(x))

table(disease.test$y, class.1.c.te)

te.err.c <- 1 - sum(diag(table(disease.test$y, class.1.c.te)))/300



# d #

err.d <-

sapply(1:10, function(h) {

  predict.1.d.tr <- knn(disease.train, disease.train, disease.train$y, k=10*h);

  predict.1.d.te <- knn(disease.train, disease.test , disease.train$y, k=10*h);

  c(1 - sum(diag(table(disease.train$y, predict.1.d.tr)))/2700,

    1 - sum(diag(table(disease.test$y , predict.1.d.te)))/300)

})

err.d <- data.frame(cbind(1/((1:10)*10), t(err.d)))

names(err.d) <- c("1/K", "train", "test")

row.names(err.d) <- paste("k =", (1:10)*10)

plot(err.d$train ~ err.d$`1/K`, type = "b", ylim = range(err.d[, 2:3]),

     xlab = "1/K", ylab = "error rate", main = "Comparison of errors for KNN",

     col = "gray47", pch = 16)

points(err.d$test  ~ err.d$`1/K`, type = "b", col = "blue3", pch = 16)

abline(v = err.d$`1/K`[which.min(err.d$test)], col = "red", lty = 2)

legend(x = 0.035, y = 0.085, legend = c("training error rate", "test error rate", "minimum test error rate"),

       pch = c(16,16,95), col = c("gray47", "blue3", "red"))

### 2 ###

# a #

err.compare.2 <-

sapply(1:20, function(M){

  xnam <- paste0("x", 1:M);

  fmla <- as.formula(paste("y ~ 0 + ", paste(xnam, collapse= "+")));

  fit.2.a <- lm(fmla, data = data2.train);

  tr.err.2a <- (1/(150-M))*sum(data2.train$y - predict(fit.2.a ,data2.train))^2;

  te.err.2a <- (1/150     )*sum((data2.test$y - predict(fit.2.a ,data2.test))^2);

  c(tr.err.2a, te.err.2a)

})

err.compare.2 <- data.frame(t(err.compare.2))

names(err.compare.2) <- c("train", "test")

plot(err.compare.2$train , type = "b", ylim = range(err.compare.2),

     xlab = "The M-th model", ylab = "error rate", main = "Comparison errors for model M",

     col = "gray47", pch = 16)

points(err.compare.2$test, type = "b", col = "blue3", pch = 16)

abline(v = which.min(err.compare.2$test), col = "red", lty = 2)

abline(v = which.min(err.compare.2$train), col = "green", lty = 2)

legend(x = 6, y = 7,

       legend = c("training error", "test error",

                  "minimum training error", "minimum test error"),

       pch = c(16,16,95,95), col = c("gray47", "blue3", "green", "red"))




# b #

# i

choose2 <- 1:20

xnam <- paste0("x", choose2);

fmla <- as.formula(paste("y ~ 0 + ", paste(xnam, collapse= "+")));

fit.2.b <- lm(fmla, data = data2.train)

mse2.all <- mean((data2.test$y - predict(fit.2.b, data2.test))^2)

which(vif(fit.2.b)> 10)

xnam <- paste0("x", choose2[-1]);

fmla <- as.formula(paste("y ~ 0 + ", paste(xnam, collapse= "+")));

fit.2.b.v1 <- lm(fmla, data = data2.train)

mse2.1 <- mean((data2.test$y - predict(fit.2.b.v1, data2.test))^2)

xnam <- paste0("x", choose2[-8]);

fmla <- as.formula(paste("y ~ 0 + ", paste(xnam, collapse= "+")));

fit.2.b.v8 <- lm(fmla, data = data2.train)

mse2.8 <- mean((data2.test$y - predict(fit.2.b.v8, data2.test))^2)

xnam <- paste0("x", choose2[-c(1, 8)]);

fmla <- as.formula(paste("y ~ 0 + ", paste(xnam, collapse= "+")));

fit.2.b.v18 <- lm(fmla, data = data2.train)

mse2.18 <- mean((data2.test$y - predict(fit.2.b.v18, data2.test))^2)

te.err.2.b <- data.frame(c(mse2.all, mse2.1, mse2.8, mse2.18))

names(te.err.2.b) <- "test error"

row.names(te.err.2.b) <- c("model 20", "x1 removed", "x8 removed", "x1 and x8 removed")

row.names(te.err.2.b)[which.min(te.err.2.b$`test error`)]

# ii

fit.2.M1 <- lm(y ~ 0 + x1, data = data2.train)

hi <- (1/150) + (((data2.train$x1 - mean(data2.train$x1))^2)/(149*var(data2.train$x1)))
```

```
which.max(hi)

x1n <- data2.train$x1

yn  <- data2.train$y

yn[which.max(hi)] <- yn[which.max(hi)] + sd(yn)

fit.2.M1.n <- lm(yn ~ 0 + x1n)

slope.change.rate <- (fit.2.M1.n$coefficients[1] - fit.2.M1$coefficients[1])/fit.2.M1$coefficients[1]

names(slope.change.rate) <- "change rate of slope estimator"



### 3 ###

# a #

credit.train <- as.data.frame(credit.train)

sapply(6:9, function(x) {

  credit.train[, x] <- as.factor(credit.train[,x]);

  credit.test[, x]  <- as.factor(credit.test[,x] );

})

xnam <- paste0("x", 1:10)

names.3 <- names(credit.test)

names(credit.train) <- c("y", xnam)

names(credit.test)  <- c("y", xnam)

fmla <- as.formula(paste("y ~", paste(c(xnam[1:5], paste0("as.factor(", xnam[6:9], ")"), xnam[10]), collapse = "+")))

fit.3.a <- lm(fmla, data = credit.train)

summary(fit.3.a)

err.3.a <-

  c((1/(length(credit.train$y) - 12))*sum((credit.train$y - predict(fit.3.a, newdata = credit.train, type = "response"))^2),

    (1/(length(credit.test$y )      ))*sum((credit.test$y  - predict(fit.3.a, newdata = credit.test , type = "response"))^2))

names(err.3.a) <- c("training error", "test error")

err.3.a

# b #

fit.3.b <- lm(y ~ x1 + as.factor(x7) + x10, data = credit.train)

summary(fit.3.b)

confint(fit.3.b)

err.3.b <-

  c((1/(length(credit.train$y) - 4))*sum((credit.train$y - predict(fit.3.b, newdata = credit.train, type = "response"))^2),

    (1/(length(credit.test$y )      ))*sum((credit.test$y  - predict(fit.3.b, newdata = credit.test , type = "response"))^2))

names(err.3.b) <- c("train", "test")

err.compare.3 <- rbind(err.3.a, err.3.b)

row.names(err.compare.3) <- c("model a", "model b")



# c #

plot(fit.3.b)



# d #

xnam <- paste0("x", 1:10)

x7xi <- paste0("as.factor(x7):", c(xnam[1:5], paste0("as.factor(", xnam[c(6,8,9)], ")"), xnam[10]))

fmla <- unclass(sapply(1:9, function(x) as.formula(paste0("y ~ x1 + as.factor(x7) + x10 +", x7xi[x]))))

err.3.c <-

  sapply(1:9, function(x){

    fit.3.c <- lm(fmla[[x]], data = credit.train)

    c((1/(length(credit.train$y) - length(fit.3.c$coefficients) ))*sum((credit.train$y - predict(fit.3.c, newdata = credit.train, type = "response"))^2),

      (1/(length(credit.test$y ) ))*sum((credit.test$y  - predict(fit.3.c, newdata = credit.test , type = "response"))^2))

  })

err.3.c <- data.frame(t(err.3.c))

names(err.3.c) <- c("train", "test")

row.names(err.3.c) <- paste0("x7:", xnam[c(1:6, 8:10)])

which.min(err.3.c$test)

err.compare.3 <- rbind(err.3.a, err.3.b, err.3.c)

row.names(err.compare.3)[1:2] <- c("model a", "model b")

plot(err.compare.3$train[-1] , type = "b", ylim = range(err.compare.3),

     xlab = "The M-th model", ylab = "error rate", main = "Comparison errors for models in problem 3(d)",

     col = "gray47", pch = 16, xaxt = "n")

points(err.compare.3$test[-1], type = "b", col = "blue3", pch = 16)

abline(v = which.min(err.compare.3$test[-1]), col = "red", lty = 2)

legend(x = 5, y =113,

       legend = c("training error", "test error", "minimum test error"),

       pch = c(16,16,95), col = c("gray47", "blue3", "red"))

axis(1, 1:10, c("model b", paste0("x7:", xnam[c(1:6, 8:10)])))
```