Homework01

JongCheolLee

2024-09-30

Data

```r
library(MASS)
data(Boston)
```

```r
head(Boston)
```

```
##      crim zn indus chas   nox    rm  age    dis rad tax ptratio  black lstat
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90  4.98
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90  9.14
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83  4.03
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63  2.94
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90  5.33
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12  5.21
##   medv
## 1 24.0
## 2 21.6
## 3 34.7
## 4 33.4
## 5 36.2
## 6 28.7
```

```r
y <- Boston[, 1]
x <- Boston[, -c(1,4,9)]
x <- as.matrix(scale(x))
```

- 506 observations with 14 variables
- **crim**: response variable (11 variables are scaled predictors)

```r
dim(x)
```

```
## [1] 506  11
```

```r
length(y)
```

```
## [1] 506
```

```r
class(x)
```

```
## [1] "matrix" "array"
```

```r
colnames(x)
```

```
##  [1] "zn"      "indus"   "nox"     "rm"      "age"     "dis"     "tax"
##  [8] "ptratio" "black"   "lstat"   "medv"
```

Housing Values in Suburbs of Boston

- **crim**: 1 .
- **zn**: 25,000 .
- **indus**: .
- **nox**: (10).
- **rm**: .
- **age**: 1940 .
- **dis**: 5 .
- **tax**: 10,000 .
- **ptratio**: - .
- **black**: 1000(Bk−0.63)^2 Bk .
- **lstat**: ( ).
- **medv**: (: 1,000).

Goal Boston 1

```r
apply(x, 2, function(t) sum(is.na(t)))
```

```
##      zn   indus     nox      rm     age     dis     tax ptratio   black   lstat
##       0       0       0       0       0       0       0       0       0       0
##    medv
##       0
```

```r
head(x, 3)
```

```
##           zn      indus        nox        rm        age      dis        tax
## 1  0.2845483 -1.2866362 -0.1440749 0.4132629 -0.1198948 0.140075 -0.6659492
## 2 -0.4872402 -0.5927944 -0.7395304 0.1940824  0.3668034 0.556609 -0.9863534
## 3 -0.4872402 -0.5927944 -0.7395304 1.2814456 -0.2655490 0.556609 -0.9863534
##      ptratio     black      lstat       medv
## 1 -1.4575580 0.4406159 -1.0744990  0.1595278
## 2 -0.3027945 0.4406159 -0.4919525 -0.1014239
## 3 -0.3027945 0.3960351 -1.2075324  1.3229375
```

Question 1.

```r
test <- x[1,]
train <- x[-1,]

te <- 1
tran <- (1:nrow(x))[-te]

dist_1 <- function(train, test) {
  diff <- train - matrix(rep(test, nrow(train)),
                         nrow=nrow(train), byrow=T)
  dists <- apply(abs(diff), 1, sum)
  dists <- as.numeric(dists)
  return(dists)
}
```

```r
dist_2 <- function(train, test) {
  diff <- train - matrix(rep(test, nrow(train)),
                         nrow=nrow(train), byrow=T)
  dists <- sqrt(apply(diff^2, 1, sum))
  dists <- as.numeric(dists)
  return(dists)
}
```

```r
dist_3 <- function(train, test) {
  diff <- train - matrix(rep(test, nrow(train)),
                         nrow=nrow(train), byrow=T)
  numer <- abs(diff)
  denom <- abs(train) + abs(matrix(rep(test, nrow(train)),
                                   nrow=nrow(train), byrow=T))
  dists <- apply((numer/denom), 1, sum)
  dists <- as.numeric(dists)
  return(dists)
}
```

```r
fhat <- function(dist_func, train, test, target, K) {
  dist_vector <- dist_func(train, test)
  closest_K <- order(dist_vector,
                     decreasing = F)[1:K]
  fhat <- mean(target[closest_K])
  return(fhat)
}

fhat_1 <- fhat(dist_func=dist_1,
               train=train,
               test=test,
               target=y[tran],
               K=10)
fhat_2 <- fhat(dist_func=dist_2,
               train=train,
               test=test,
               target=y[tran],
               K=10)
fhat_3 <- fhat(dist_func=dist_3,
```

```r
               train=train,
               test=test,
               target=y[tran],
               K=10)

data.frame(l=1:3,fhat=c(fhat_1, fhat_2, fhat_3))
```

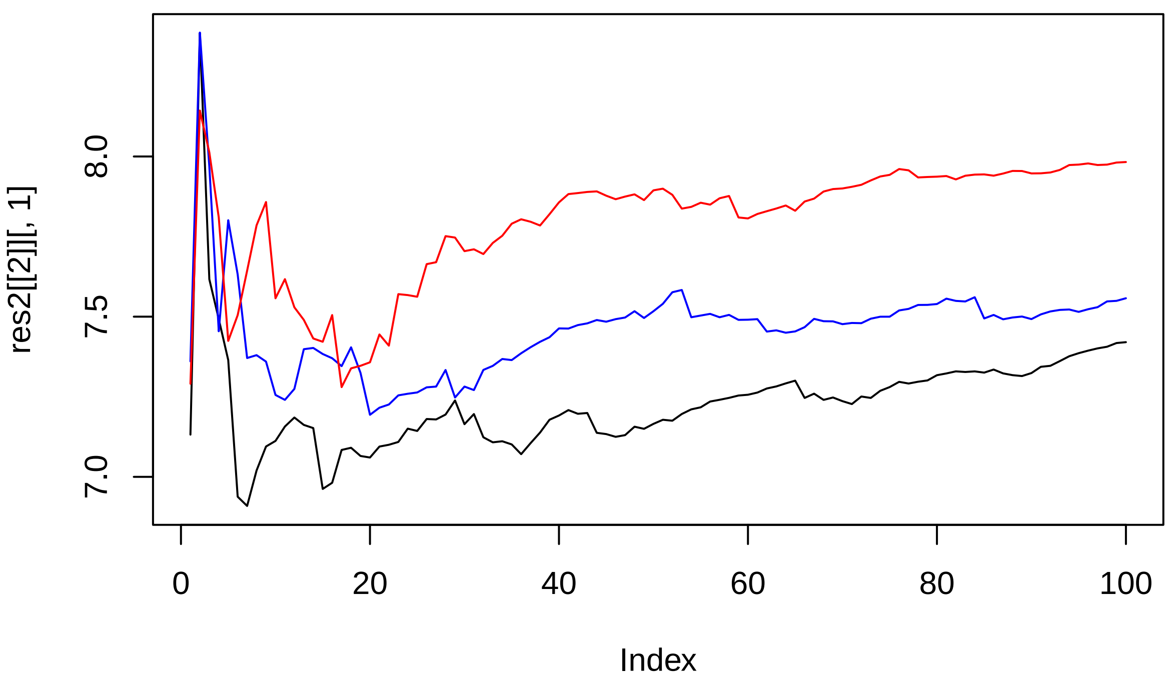
```
##   l     fhat
## 1 1 0.115894
## 2 2 0.201866
## 3 3 0.074659
```

```r
# [1] 0.115894 0.201866 0.074659
```

Question 2.

```r
set.seed(12345)
tran <- sample(nrow(x), 400) # Randomly selected 400 samples
te <- c(1:nrow(x))[-tran]

PE <- function(x, y) {
  train <- x[tran,]
  testset <- x[-tran,]
  y_te <- y[te]

  K_list <- seq(1,100)

  f1_mat <- matrix(0, nrow=nrow(testset), ncol=100)
  f2_mat <- matrix(0, nrow=nrow(testset), ncol=100)
  f3_mat <- matrix(0, nrow=nrow(testset), ncol=100)

  for (i in 1:nrow(testset)) {
    test <- testset[i,]
    y_i <- y_te[i]
    for (k in K_list) {
      fhat_1 <- fhat(dist_1, train, test, y[tran], K=k)
      fhat_2 <- fhat(dist_2, train, test, y[tran], K=k)
      fhat_3 <- fhat(dist_3, train, test, y[tran], K=k)
      f1_mat[i,k] <- (y_i - fhat_1)^2
      f2_mat[i,k] <- (y_i - fhat_2)^2
      f3_mat[i,k] <- (y_i - fhat_3)^2
    }
  }

  PE1 <- apply(f1_mat, 2, function(t) sqrt(mean(t)))
  PE2 <- apply(f2_mat, 2, function(t) sqrt(mean(t)))
  PE3 <- apply(f3_mat, 2, function(t) sqrt(mean(t)))

#  plot(PE1, type="l")
#  lines(PE2, col="blue")
#  lines(PE3, col="red")
```

```r
  result_df <- data.frame(K_opt=c(which.min(PE1),
                                   which.min(PE2),
                                   which.min(PE3)),
                          PE_value=c(min(PE1),
                                     min(PE2),
                                     min(PE3)))

  return(list(result_df, cbind(PE1, PE2, PE3)))
}


res2 <- PE(x,y)

plot(res2[[2]][,1], type="l")
lines(res2[[2]][,2], col="blue")
lines(res2[[2]][,3], col="red")
```



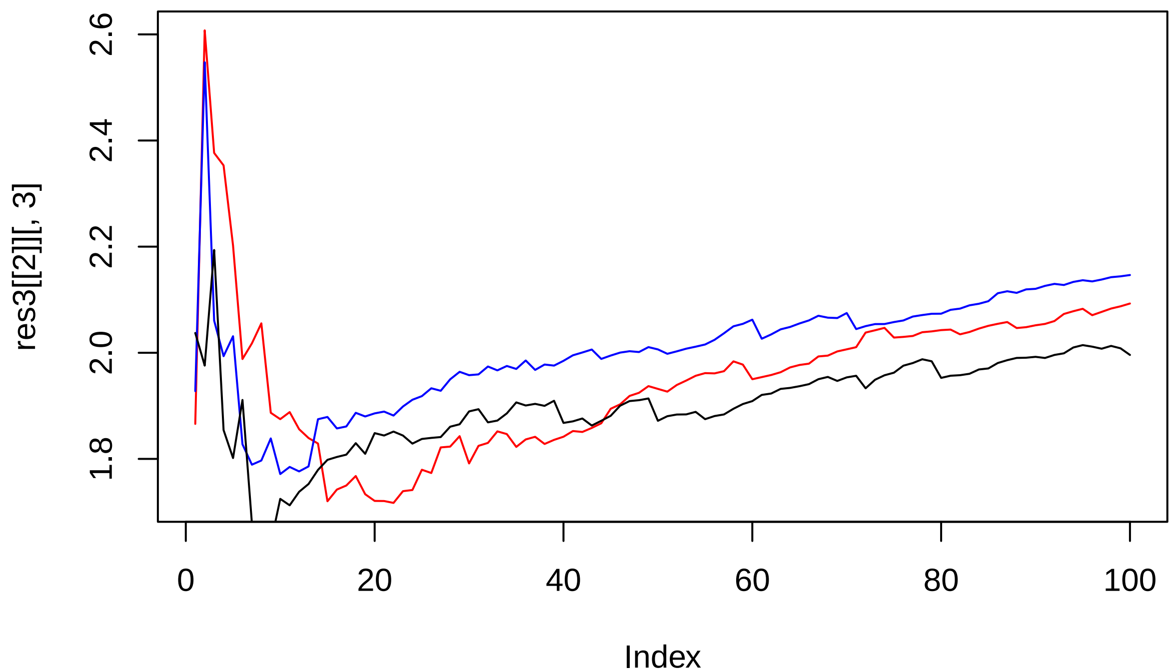

```r
res2[[1]]
```

```
##   K_opt PE_value
## 1     7 6.909187
## 2    20 7.193706
## 3    17 7.280037
```

Question 3.

```r
set.seed(1234)
foldID <- sample(rep(1:10, length=nrow(x)))

CVE <- function(x, y){
  for (id in 1:10) {
    res_mat <- matrix(0, nrow=100, ncol=3)

    train <- x[foldID!=id,]
    testset <- x[foldID==id,]
    y_te <- y[foldID==id]

    K_list <- 1:100

    f1_mat <- matrix(0, nrow=nrow(testset), ncol=100)
    f2_mat <- matrix(0, nrow=nrow(testset), ncol=100)
    f3_mat <- matrix(0, nrow=nrow(testset), ncol=100)

    for (i in 1:nrow(testset)) {
      test <- testset[i,]
      y_i <- y_te[i]
      for (k in K_list) {
        fhat_1 <- fhat(dist_1, train, test, y[foldID!=id], K=k)
        fhat_2 <- fhat(dist_2, train, test, y[foldID!=id], K=k)
        fhat_3 <- fhat(dist_3, train, test, y[foldID!=id], K=k)
        f1_mat[i,k] <- (y_i - fhat_1)^2
        f2_mat[i,k] <- (y_i - fhat_2)^2
        f3_mat[i,k] <- (y_i - fhat_3)^2
      }
    }
    mkPE1 <- apply(f1_mat, 2, sum)
    mkPE2 <- apply(f2_mat, 2, sum)
    mkPE3 <- apply(f3_mat, 2, sum)

    res_mat <- res_mat + cbind(mkPE1, mkPE2, mkPE3)
  }

  CVE_mat <- sqrt(res_mat/506)

#  plot(CVE_mat[,1], type="l")
#  lines(CVE_mat[,2], col="blue")
#  lines(CVE_mat[,3], col="red")

  K_opt <- apply(CVE_mat, 2, which.min)
  CVE_value <- apply(CVE_mat, 2, min)

  result_df <- data.frame(K_opt=c(K_opt[1],
                                  K_opt[2],
                                  K_opt[3]),
                          CVE_value=c(CVE_value[1],
                                      CVE_value[2],
                                      CVE_value[3]))
```

```r
  rownames(result_df) = c("1","2","3")

  return(list(result_df, CVE_mat))
}

res3 <- CVE(x,y)

plot(res3[[2]][,3], type="l", col="red")
lines(res3[[2]][,2], col="blue")
lines(res3[[2]][,1])
```




```r
res3[[1]]
```

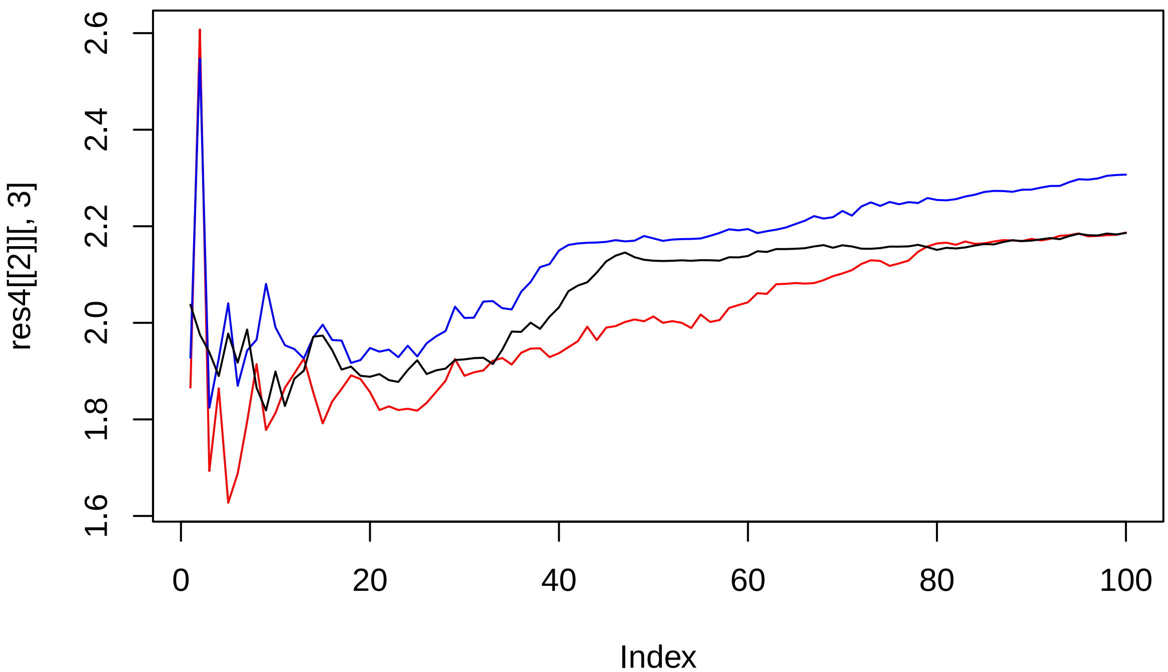
```
##   K_opt CVE_value
## 1     8  1.610041
## 2    10  1.771461
## 3    22  1.717081
```

Question 4.

```r
ghat <- function(dist_func, train, test, target, K) {
  dist_vector <- dist_func(train, test)
  closest_K <- order(dist_vector,
```

```
                      decreasing = F)[1:K]
  ghat <- median(target[closest_K])
  return(ghat)
}

set.seed(1234)
foldID <- sample(rep(1:10, length=nrow(x)))

CVE_g <- function(x, y){
  for (id in 1:10) {
    res_mat <- matrix(0, nrow=100, ncol=3)

    train <- x[foldID!=id,]
    testset <- x[foldID==id,]
    y_te <- y[foldID==id]

    K_list <- 1:100

    g1_mat <- matrix(0, nrow=nrow(testset), ncol=100)
    g2_mat <- matrix(0, nrow=nrow(testset), ncol=100)
    g3_mat <- matrix(0, nrow=nrow(testset), ncol=100)

    for (i in 1:nrow(testset)) {
      test <- testset[i,]
      y_i <- y_te[i]
      for (k in K_list) {
        ghat_1 <- ghat(dist_1, train, test, y[foldID!=id], K=k)
        ghat_2 <- ghat(dist_2, train, test, y[foldID!=id], K=k)
        ghat_3 <- ghat(dist_3, train, test, y[foldID!=id], K=k)
        g1_mat[i,k] <- (y_i - ghat_1)^2
        g2_mat[i,k] <- (y_i - ghat_2)^2
        g3_mat[i,k] <- (y_i - ghat_3)^2
      }
    }
    mkPE1 <- apply(g1_mat, 2, sum)
    mkPE2 <- apply(g2_mat, 2, sum)
    mkPE3 <- apply(g3_mat, 2, sum)

    res_mat <- res_mat + cbind(mkPE1, mkPE2, mkPE3)
  }

  CVE_mat <- sqrt(res_mat/506)

#  plot(CVE_mat[,1], type="l")
#  lines(CVE_mat[,2], col="blue")
#  lines(CVE_mat[,3], col="red")

  K_opt <- apply(CVE_mat, 2, which.min)
  CVE_value <- apply(CVE_mat, 2, min)

  result_df <- data.frame(K_opt=c(K_opt[1],
                                  K_opt[2],
                                  K_opt[3]),
```

```r
                          CVE_value=c(CVE_value[1],
                                      CVE_value[2],
                                      CVE_value[3]))
  rownames(result_df) <- 1:3

  return(list(result_df, CVE_mat))
}

res4 <- CVE_g(x,y)

plot(res4[[2]][,3], type="l", col="red")
lines(res4[[2]][,2], col="blue")
lines(res4[[2]][,1])
```



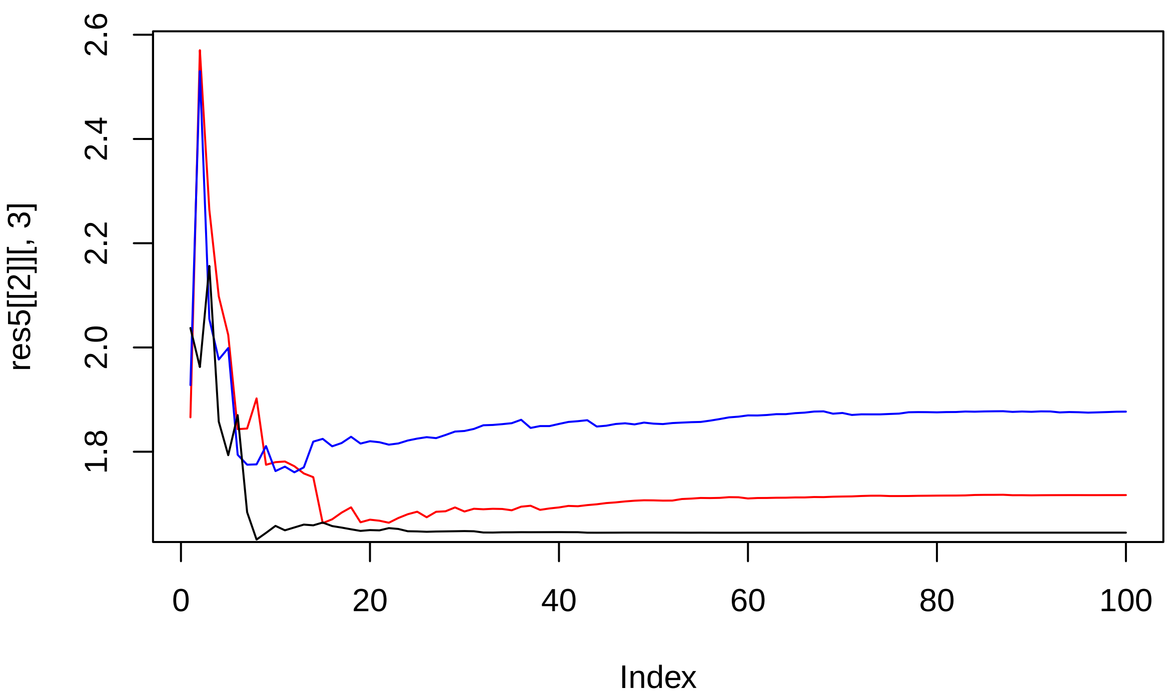

```r
res4[[1]]
```

```
##   K_opt CVE_value
## 1     9  1.818608
## 2     3  1.824093
## 3     5  1.627458
```

Question 5.

```r
hhat <- function(dist_func, train, test, target, K) {
  dist_vector <- dist_func(train, test)
  closest_K <- order(dist_vector,
                     decreasing = F)[1:K]

  min_dist <- min(dist_vector)

  delta <- exp(-(dist_vector[closest_K] - min_dist)^2)
  D_lk <- sum(delta)

  hhat <- sum(delta*target[closest_K])/(D_lk)
  return(hhat)
}

set.seed(1234)
foldID <- sample(rep(1:10, length=nrow(x)))

CVE_h <- function(x, y){
  for (id in 1:10) {
    res_mat <- matrix(0, nrow=100, ncol=3)

    train <- x[foldID!=id,]
    testset <- x[foldID==id,]
    y_te <- y[foldID==id]

    K_list <- 1:100

    h1_mat <- matrix(0, nrow=nrow(testset), ncol=100)
    h2_mat <- matrix(0, nrow=nrow(testset), ncol=100)
    h3_mat <- matrix(0, nrow=nrow(testset), ncol=100)

    for (i in 1:nrow(testset)) {
      test <- testset[i,]
      y_i <- y_te[i]
      for (k in K_list) {
        hhat_1 <- hhat(dist_1, train, test, y[foldID!=id], K=k)
        hhat_2 <- hhat(dist_2, train, test, y[foldID!=id], K=k)
        hhat_3 <- hhat(dist_3, train, test, y[foldID!=id], K=k)
        h1_mat[i,k] <- (y_i - hhat_1)^2
        h2_mat[i,k] <- (y_i - hhat_2)^2
        h3_mat[i,k] <- (y_i - hhat_3)^2
      }
    }
    mkPE1 <- apply(h1_mat, 2, sum)
    mkPE2 <- apply(h2_mat, 2, sum)
    mkPE3 <- apply(h3_mat, 2, sum)

    res_mat <- res_mat + cbind(mkPE1, mkPE2, mkPE3)
  }

  CVE_mat <- sqrt(res_mat/506)
```

```
#  plot(CVE_mat[,1], type="l")
#  lines(CVE_mat[,2], col="blue")
#  lines(CVE_mat[,3], col="red")

  K_opt <- apply(CVE_mat, 2, which.min)
  CVE_value <- apply(CVE_mat, 2, min)

  result_df <- data.frame(K_opt=c(K_opt[1],
                                  K_opt[2],
                                  K_opt[3]),
                          CVE_value=c(CVE_value[1],
                                      CVE_value[2],
                                      CVE_value[3]))
  rownames(result_df) <- 1:3

  return(list(result_df, CVE_mat))
}

res5 <- CVE_h(x,y)

plot(res5[[2]][,3], type="l", col="red")
lines(res5[[2]][,2], col="blue")
lines(res5[[2]][,1])
```



```r
res5[[1]]
```

```
##   K_opt CVE_value
## 1     8  1.631452
## 2    12  1.760485
## 3    15  1.663037
```

Question 6.

	$\hat{f}$		$\hat{g}$		$\hat{h}$	
	K	CVE	K	CVE	K	CVE
d_1	8	1.610	9	1.819	8	1.631
d_2	10	1.771	3	1.824	12	1.760
d_3	22	1.717	5	1.627	15	1.663

Figure 1: My Image