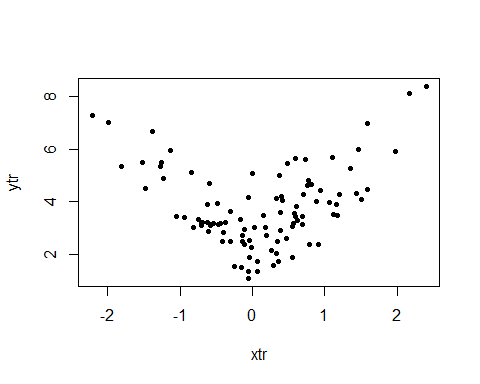
R-KNN

JeongHo LEE

## 1. 데이터 생성 ()

set.seed(1)  
xtr <- sort(rnorm(100))  
ytr <- 3 + xtr^2 + rnorm(100)  
library(FNN)  
eval.n <- 100  
eval.point = seq(-3,3, length= eval.n)  
plot(xtr, ytr, pch = 20)



시드를 1로 두고 , 를 100개 생성했다. x, y를 포인트로 찍어 plot을 그리면 위와 같다.

## 2. K-NN 적합 ()

a <- 100  
yhat <- matrix(0, eval.n, a)  
for(j in 1:a){  
 idx.mat <- knnx.index(xtr, eval.point , k = j)  
 for (i in 1:eval.n){  
 yhat[i,j]<-mean(ytr[idx.mat[i,]])  
 }  
}

knnx.index를 이용해 k에 따른 를 구해서 yhat의 각 열에 입력했다.

## 3. 새로운 데이터 생성 ()

xte <- list()  
yte <- list()  
for(i in 1:10){  
 set.seed(7\*i+140)  
 xte[[i]] <- sort(rnorm(100))  
 yte[[i]] <- 3 + xte[[i]]^2 + rnorm(100)  
}

여러번 시행으로 비교해 보기 위하여 시드를 20가지로 주어서 시드별로 생성된 데이터를 xte, yte 리스트에 저장하였다.

## 4. 구하기

xte.idx <- list()  
sse <- rep(0,a)  
sse.l <- list()  
for(j in 1:10){  
 xte.idx[[j]] <- as.numeric(cut(xte[[j]], breaks = eval.point, labels = 1:99))  
 for(i in 1:a){  
 sse[i] <- sum((yte[[j]]-yhat[xte.idx[[j]],i])^2)  
 }  
 sse.l[[j]] <- sse  
}

코드에서 yte와 yhat을 그냥 빼서 제곱하면, 인덱스가 어긋나므로 원하는 값이 나오지 않는다. 따라서 xtr의 점 사이의 구간을 나누어서 xte의 원소가 존재하는 각 구간을 인덱스(xte.idx)로 잡고, 이를 yhat의 인덱스로 지정하여 인덱스가 어긋나지 않게 계산할 수 있게 하였다.

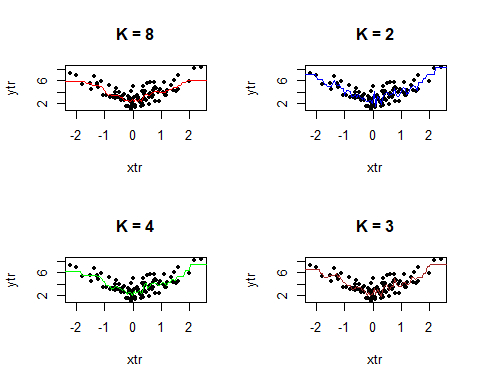
이런 식으로 k에 따른 를 각각 리스트 sse.l에 저장하였다.

## 5. k 선택하기

which.k <- sapply(sse.l, which.min)  
which.k

## [1] 8 2 8 2 8 3 4 2 8 4

par(mfrow = c(2,2))  
plot(xtr, ytr, main = "K = 8", pch = 20)  
lines(eval.point , yhat[ ,8], type= 'l', lty = 1, col = 'red')  
  
plot(xtr, ytr, main = "K = 2", pch = 20)  
lines(eval.point , yhat[ ,2], type= 'l', lty = 1, col = 'blue')  
  
plot(xtr, ytr, main = "K = 4", pch = 20)  
lines(eval.point , yhat[ ,4], type= 'l', lty = 1, col = 'green')  
  
plot(xtr, ytr, main = "K = 3", pch = 20)  
lines(eval.point , yhat[ ,3], type= 'l', lty = 1, col = 'brown')



각 시드별로 제곱합 값이 최소인 k를 구하면, 순서대로 8, 2, 8, 2, 8, 3, 4, 2, 8, 4로 8이 4번, 2가 3번, 4가 2번 3이 한 번 나왔다.

각 K에 따른 의 그래프와 KNN으로 를 구하는데 썼던 , 의 plot과 함께 비교해보면 위와 같고, 시행의 결과로 모형이 가장 좋은 k는 8이라고 선택할 수 있겠다.