Delivery 1: AMA

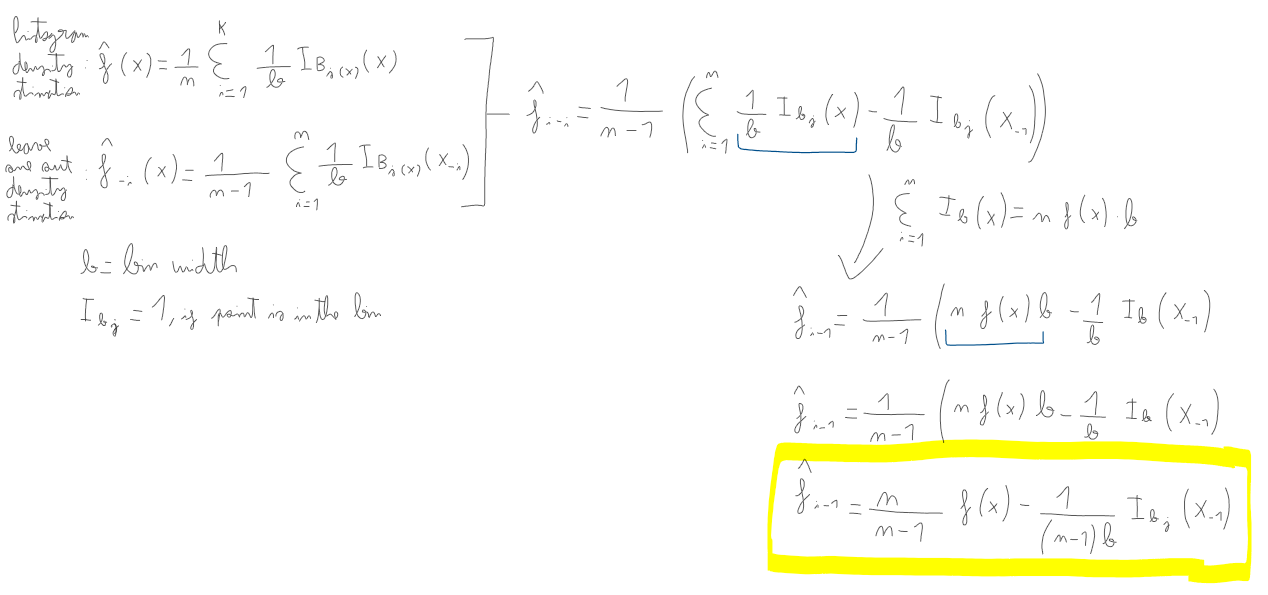
Victor Garcia Pizarro

2024-10-03

Table of Contents

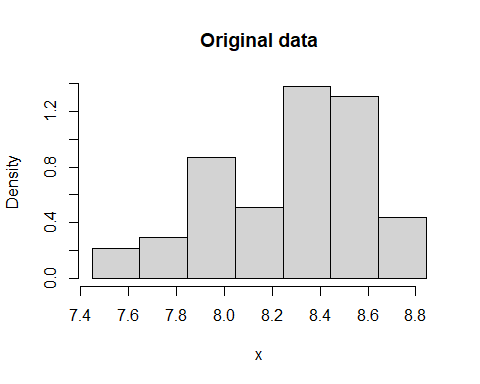
# Exercise 1

To calculate the relation we used the following approach:



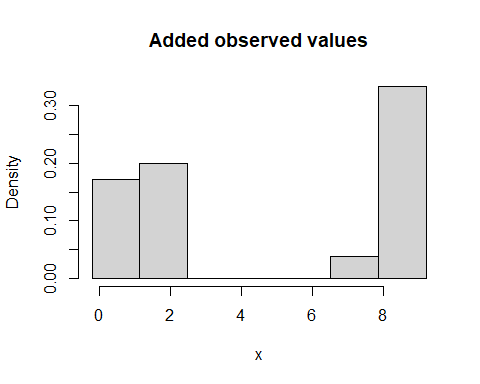
# Exercise 2

# Read and transform x to numeric  
df = read.table("cdrate.dat", col.names = c("x", "y"))  
x = df["x"]  
class(x) = "Numeric"  
x = x$x  
  
A <- min(x)-.05\*diff(range(x))  
Z <- max(x)+.05\*diff(range(x))  
nbr <- 7  
  
# Define histogram and the function  
hx <- hist(x,breaks=seq(A,Z,length=nbr+1),freq=F, main = "Original data")



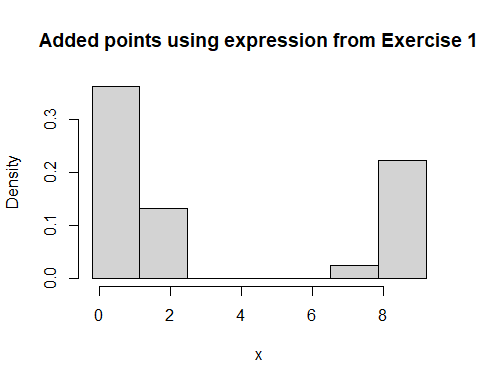
hx\_f <- stepfun(hx$breaks,c(0,hx$density,0))

# calculate observed values  
observed\_values = hx\_f(x)  
  
# add them to the histogram  
all\_data = append(x, observed\_values)  
  
A <- min(all\_data)-.05\*diff(range(all\_data))  
Z <- max(all\_data)+.05\*diff(range(all\_data))  
nbr <- 7  
  
hx <- hist(all\_data, breaks=seq(A,Z,length=nbr+1),freq=F, main = "Added observed values", xlab = "x")



# Exercise 3

leave\_one\_out\_density <- function(data, nbr) {  
 n <- length(data)  
 loo\_density <- numeric(n)   
   
 # Iterate over all points  
 for (i in 1:n) {  
   
 # Exclude one observation  
 data\_exclude <- data[-i]  
   
 # Fit a new histogram finding new number of breaks  
 A <- min(data\_exclude)-.05\*diff(range(data\_exclude))  
 Z <- max(data\_exclude)+.05\*diff(range(data\_exclude))  
   
 hist\_obj <- hist(data\_exclude, breaks=seq(A,Z,length=nbr+1), plot=F)  
   
 # calculate density: number of points (count) in each interval / width   
 loo\_density[i] <- sum(hist\_obj$counts) / (n - 1) \* (hist\_obj$mids[2] - hist\_obj$mids[1])  
 }  
   
 return(loo\_density)  
}  
  
num\_bins <- 7   
  
# Calculate leave-one-out density and add values the ones previously calculated  
loo\_density <- leave\_one\_out\_density(x, 2)  
all\_data2 = append(all\_data, loo\_density)  
  
# plot   
hx <- hist(all\_data2, breaks=seq(A,Z,length=num\_bins+1),freq=F, main = "Added points using expression from Exercise 1", xlab = "x")



# Exercise 4

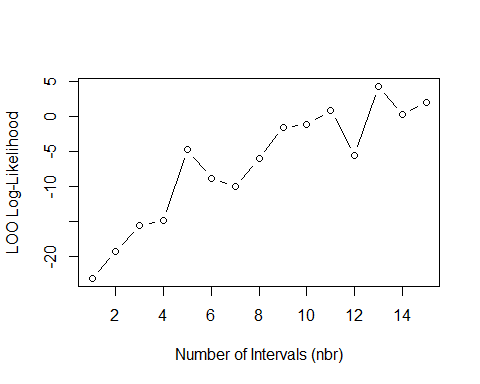
leave\_one\_out\_log\_likelihood <- function(data) {  
 n <- length(data)  
 log\_likelihoods <- numeric(n)   
  
 # remove one point, fit a normal distribution and calculate  
 # probability distribution. Then, the log-likelihood  
 for (i in 1:n) {  
 # Leave one out  
 data\_loo <- data[-i]   
   
 mean\_loo <- mean(data\_loo)  
 sd\_loo <- sd(data\_loo)  
   
 left\_out\_point <- data[i]  
 pdf\_value <- dnorm(left\_out\_point, mean = mean\_loo, sd = sd\_loo)  
   
 log\_likelihoods[i] <- log(pdf\_value)  
 }  
   
 # the log likelihood is the sum of all iterations  
 return(sum(log\_likelihoods))   
}  
  
# Compute the leave-one-out log-likelihood  
total\_log\_likelihood <- leave\_one\_out\_log\_likelihood(all\_data2)  
total\_log\_likelihood

## [1] -555.0707

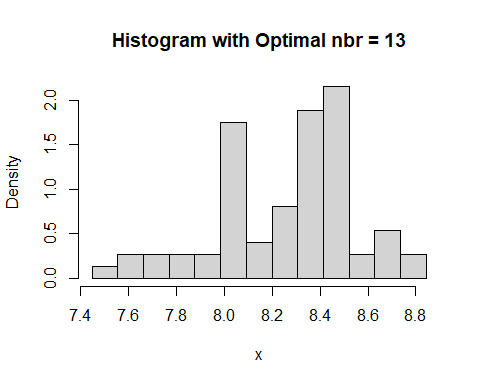
# Exercise 5

For exercise 5, we determined the optimal number of histogram intervals (nbr) by calculating the leave-one-out log-likelihood for different values of nbr ranging from 1 to 15. We then selected the value that maximized the log-likelihood as the optimal choice, and used this to plot the final histogram.

A <- min(x) - 0.05 \* diff(range(x))  
Z <- max(x) + 0.05 \* diff(range(x))  
  
nbr\_values <- seq(1, 15)  
looCV\_log\_lik <- numeric(length(nbr\_values))  
  
for (i in 1:length(nbr\_values)) {  
 nbr <- nbr\_values[i]  
 hx <- hist(x, breaks = seq(A, Z, length = nbr + 1), plot = FALSE)  
 hx\_f <- stepfun(hx$breaks, c(0, hx$density, 0))  
   
 # Compute the leave-one-out log-likelihood  
 loo\_log\_lik <- sum(log(hx\_f(x)[hx\_f(x) > 0]))  
 looCV\_log\_lik[i] <- loo\_log\_lik  
}  
  
plot(nbr\_values, looCV\_log\_lik, type = "b", xlab = "Number of Intervals (nbr)", ylab = "LOO Log-Likelihood")



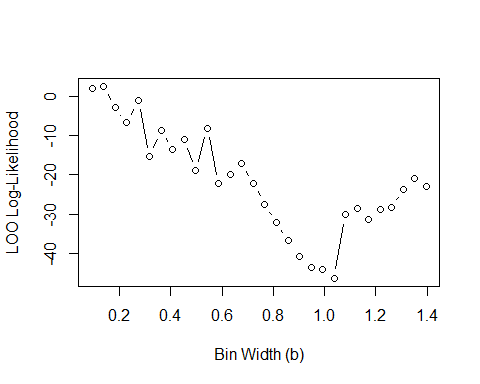
optimal\_nbr <- nbr\_values[which.max(looCV\_log\_lik)]  
  
hx\_optimal <- hist(x, breaks = seq(A, Z, length = optimal\_nbr + 1), plot = FALSE)  
plot(hx\_optimal, freq = FALSE, main = paste("Histogram with Optimal nbr =", optimal\_nbr))



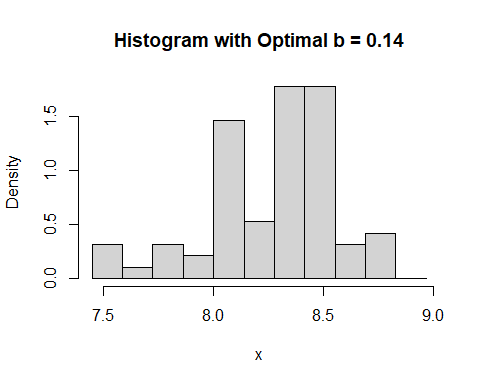
# Exercise 6

Here we followed a similar approach to exercise 5, but focused on finding the optimal bin width (b). We evaluated the leave-one-out log-likelihood for a range of possible bin widths and chose the one that maximized the log-likelihood. The histogram was then plotted using this optimal bin width.

b\_values <- seq((Z - A) / 15, (Z - A) / 1, length = 30)  
  
looCV\_log\_lik\_b <- numeric(length(b\_values))  
  
for (i in 1:length(b\_values)) {  
 b <- b\_values[i]  
 hx <- hist(x, breaks = seq(A, Z + b, by = b), plot = FALSE)  
 hx\_f <- stepfun(hx$breaks, c(0, hx$density, 0))  
   
 # Compute the leave-one-out log-likelihood  
 loo\_log\_lik <- sum(log(hx\_f(x)[hx\_f(x) > 0]))  
 looCV\_log\_lik\_b[i] <- loo\_log\_lik  
}  
  
  
plot(b\_values, looCV\_log\_lik\_b, type = "b", xlab = "Bin Width (b)", ylab = "LOO Log-Likelihood")



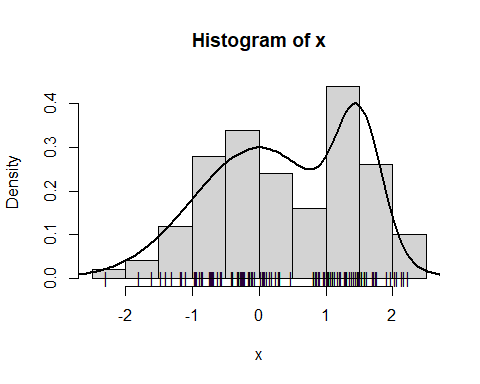
optimal\_b <- b\_values[which.max(looCV\_log\_lik\_b)]  
  
hx\_optimal\_b <- hist(x, breaks = seq(A, Z + optimal\_b, by = optimal\_b), plot = FALSE)  
plot(hx\_optimal\_b, freq = FALSE, main = paste("Histogram with Optimal b =", round(optimal\_b, 2)))



# Exercise 7

Generate data from

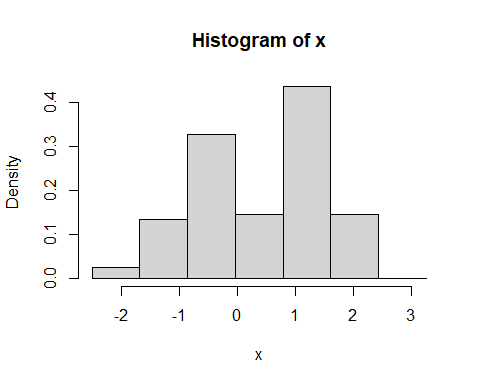
sim.mixt <- function(n=1,k=1,   
 mu=seq(-2\*(k-1),2\*(k-1),length=k),   
 sigma=seq(1,1,length=k),   
 alpha=seq(1/k,1/k,length=k), graphic=FALSE,...)  
{  
 csa<-cumsum(alpha)  
 x<-runif(n)  
   
 for (i in 1:n){  
 comp<-sum(csa<=x[i])+1  
 x[i]<-rnorm(1,mu[comp],sigma[comp])  
 }  
 if(graphic) {  
 out<-graph.mixt(k, mu, sigma, alpha, gr=FALSE)  
 hist(x,freq = FALSE,  
 ylim=c(0,max(c(max(out$fx),max(hist(x,plot=FALSE)$density)))))  
 lines(out$x,out$fx,lty=1,lwd=2)  
 }   
 return(x)  
}  
  
graph.mixt<-function(k=1, mu=seq(-2\*(k-1),2\*(k-1),length=k), sigma=seq(1,1,length=k), alpha=seq(1/k,1/k,length=k), graphic=TRUE,...)  
{  
 L<-min(mu-3\*sigma)  
 U<-max(mu+3\*sigma)  
   
 x<- seq(from=L,to=U,length=200)  
 fx<- 0\*x  
 Salpha<-sum(alpha)  
 for(i in 1:k){  
 p<-alpha[i]/Salpha  
# fx <- fx + p\*exp(-.5\*((x-mu[i])/sigma[i])^2)/(sqrt(2\*pi)\*sigma[i])  
 fx <- fx + p\*dnorm(x,mu[i],sigma[i])  
 }  
 if (graphic){  
 plot(x,fx,type="l",...)  
 }  
 return(list(L = L, U = U, x = x, fx = fx))  
}  
  
n <- 100  
m <- c(0, 3/2)  
s <- c(1, 1/3)  
a <- c(3/4, 1/4)  
  
x <- sim.mixt(n=n, k=2, mu=m, sigma=s, alpha=a, gr=T)  
points(x, 0\*x, pch="|")



sigma.mixt <- 1.095287  
scott <- 3.49 \* sigma.mixt \* length(x)^(-1/3)

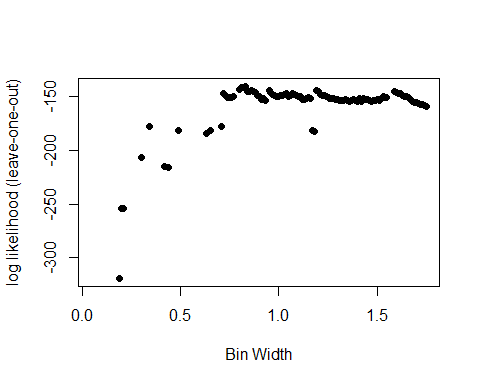
We use the Scott’s formula to find the value of b = scott (this was done in the density estimation script used in class). The follwing is the histogram result:

A <- min(x)-.05\*diff(range(x))   
Z <- max(x)+.05\*diff(range(x))  
  
hx <- hist(x,breaks=seq(A,Z+b,by=scott), plot=F)  
plot(hx,freq = FALSE)



Now taking the b value maximizing the leave-one-out log-likelihood function. The sequence of proposed b-values to select from is varied from 0.05 to 1.75 in steps of 0.01.

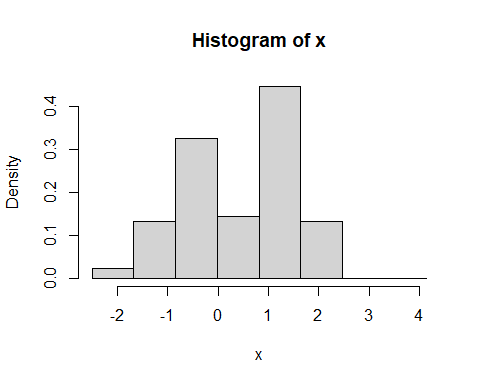
b\_seq = seq(0.05, 1.75, by=0.01)  
b\_loo\_log\_likelihood <- c()  
  
for (b in b\_seq) {  
 hx <- hist(x,breaks=seq(A,Z+b,by=b), plot=F)  
 hx\_f <- stepfun(hx$breaks,c(0,hx$density,0))  
 y\_hist <- hx\_f(x)  
 y\_loo\_hist <- (y\_hist-(1/(length(x)\*b)))\*(length(x)/(length(x)-1))  
 if (any(y\_loo\_hist<=0)) {  
 b\_loo\_log\_likelihood <- c(b\_loo\_log\_likelihood,-Inf)  
 }  
 else {  
 b\_loo\_log\_likelihood <- c(b\_loo\_log\_likelihood,sum(log(y\_loo\_hist)))  
 }  
}  
plot(b\_seq,b\_loo\_log\_likelihood, pch=16, xlab='Bin Width', ylab = 'log likelihood (leave-one-out)')



b\_ml <- b\_seq[which.max(b\_loo\_log\_likelihood)]

From this experiment, the leave-one-out cross-validation max log likelihood method selects b = rb\_ml` as the optimal value for b (out of the proposed values for b). The corresponding histogram can be found below.

hx <- hist(x,breaks=seq(A,Z+b,by=b\_ml), plot=F)  
plot(hx,freq = FALSE)



The two suggested values for 𝑏 differ significantly, as seen in the resulting histograms and kernel density estimates. These variations in bin width create distinct impressions of the data. When using Scott’s formula, the mixture of the two distributions (in this case, Gaussian distributions) appears less pronounced, making it harder to discern both distributions clearly. In contrast, the bin width derived from the maximum-likelihood method accentuates the presence of two separate patterns, making them more visually evident.

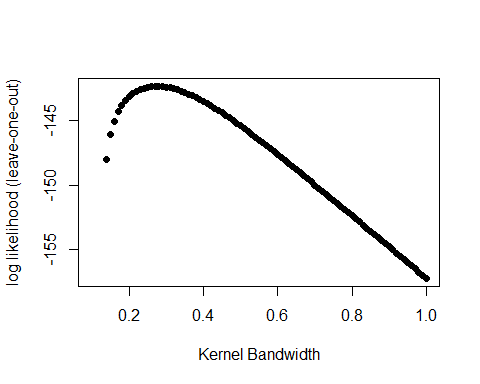
# Exercise 8

In this exercise, the Gaussian kernel is applied, with its bandwidth determined similarly to the bin width in prior exercises. A broad range of bandwidth values is tested (from 0.1 to 1 in increments of 0.01), and a plot of the leave-one-out log-likelihood is generated for each.

h\_seq <- seq(0.1,1, by=0.01)  
h\_loo\_log\_likelihood <- c()  
  
for (h in h\_seq) {  
 kx <- density(x,bw=h,kernel='gaussian')  
 kx\_f <- approxfun(x=kx$x, y=kx$y, method='linear', rule=2)  
 y\_density <- kx\_f(x)  
 y\_loo\_density <- (y\_density-(1/(sqrt(2\*pi)\*length(x)\*h)))\*(length(x)/(length(x)-1))  
   
 h\_loo\_log\_likelihood <- c(h\_loo\_log\_likelihood,sum(log(y\_loo\_density)))  
}

## Warning in log(y\_loo\_density): Se han producido NaNs  
## Warning in log(y\_loo\_density): Se han producido NaNs  
## Warning in log(y\_loo\_density): Se han producido NaNs  
## Warning in log(y\_loo\_density): Se han producido NaNs

plot(h\_seq,h\_loo\_log\_likelihood, pch=16, xlab='Kernel Bandwidth', ylab = 'log likelihood (leave-one-out)')



h\_ml <- h\_seq[which.max(h\_loo\_log\_likelihood)]

From this experiment, it becomes clear that the optimal max likelihood value for h is 0.27. The corresponding density function is plotted below.

kx <- density(x,bw=h\_ml,kernel='gaussian')  
kx\_f <- approxfun(x=kx$x, y=kx$y, method='linear', rule=2)  
plot(kx$x, kx$y, xlab='x', ylab = 'density', main = 'Gaussian Kernel Density Estimation')

