Delivery 1: AMA

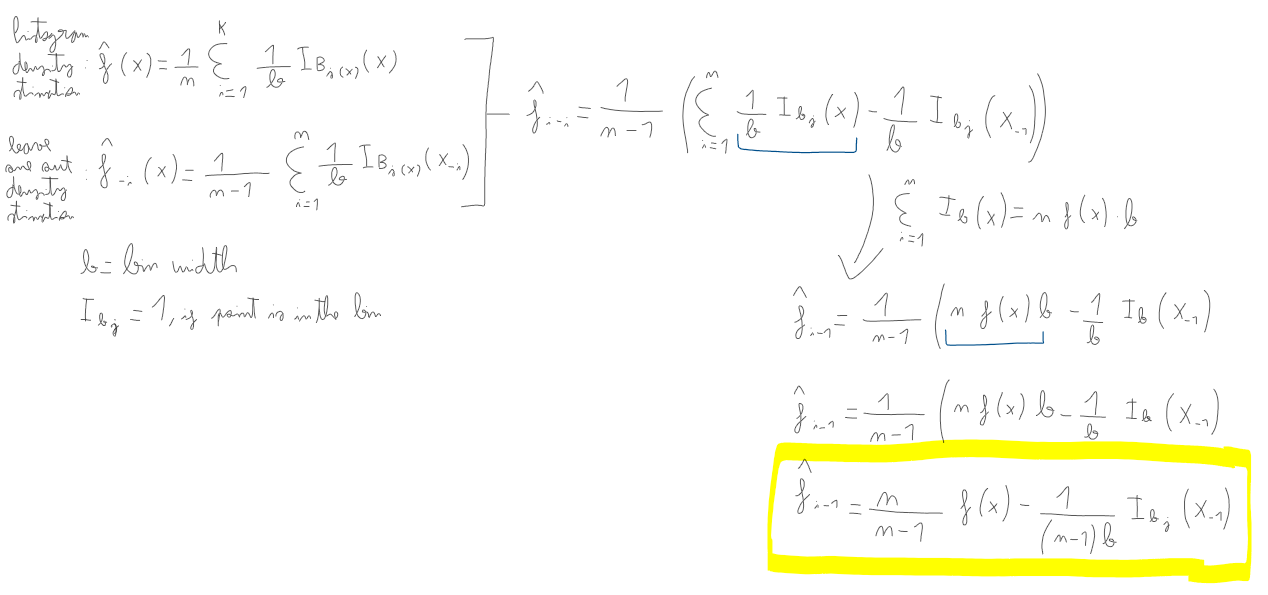
Victor Garcia Pizarro

2024-10-03

Table of Contents

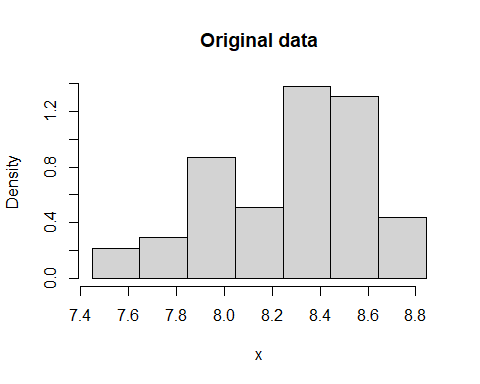
# Exercice 1

To calculate the relation we used the following approach:



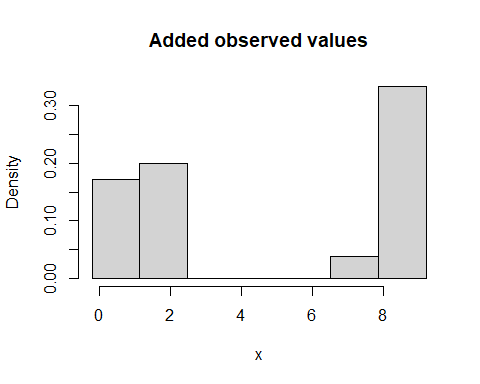
# Exercice 2

# 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")



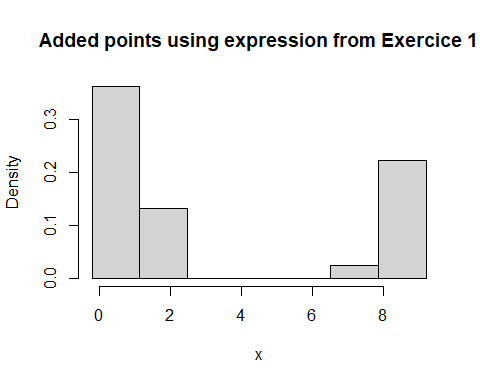
hx\_f <- stepfun(hx$breaks,c(0,hx$density,0))

# 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")



# Exercice 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 Exercice 1", xlab = "x")



# Exercice 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