Delivery 1: AMA

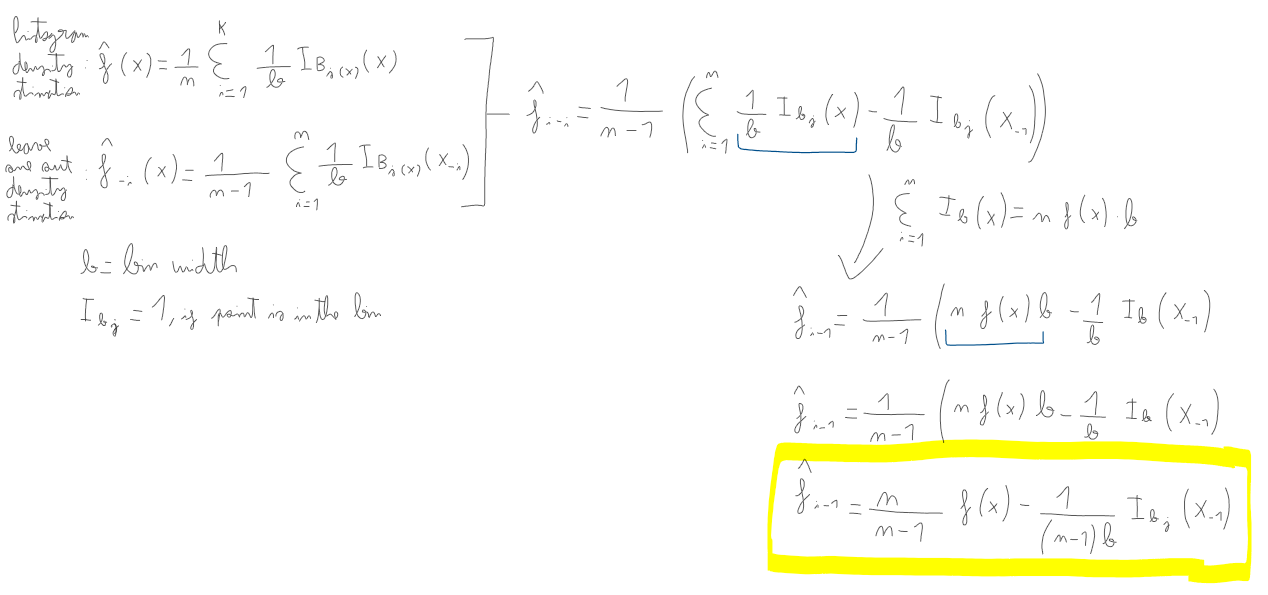
Marc Falcón Barau, Julian Fransen,Victor Garcia Pizarro

2024-10-03

Table of Contents

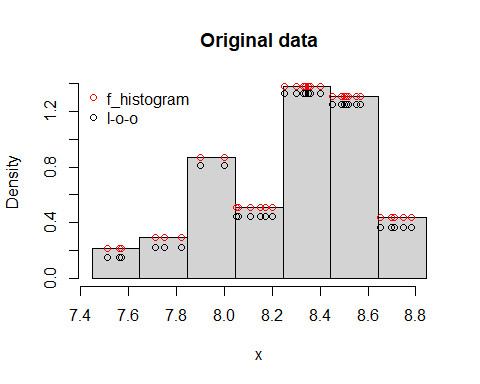
# Exercise 1

To calculate the relation we used the following approach:



# Exercise 2-3

# 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))  
binwidth <- hx$breaks[2]-hx$breaks[1]  
points(x,hx\_f(x),col="red", pch=1)  
  
y\_loo\_hist <- (hx\_f(x)-(1/(length(x)\*binwidth)))\*(length(x)/(length(x)-1))  
points(x,y\_loo\_hist,col="black", pch=1)  
  
legend("topleft", legend = c("f\_histogram", "l-o-o"), col = c("red", "black"), pch = 1, bty = "n")



# Exercise 4

The leave-one-out log-likelihood is calculated by the function .

l <- sum(log(y\_loo\_hist))  
print("The leave-one-out log-likelihood is:")

## [1] "The leave-one-out log-likelihood is:"

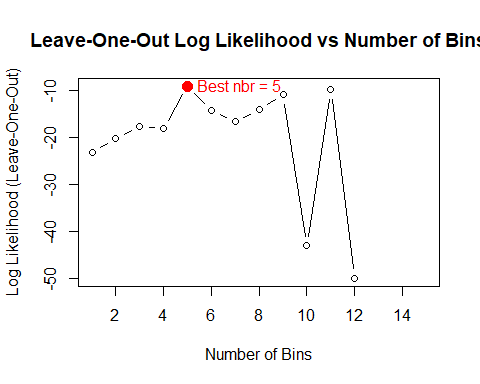
print(l)

## [1] -16.58432

# 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.

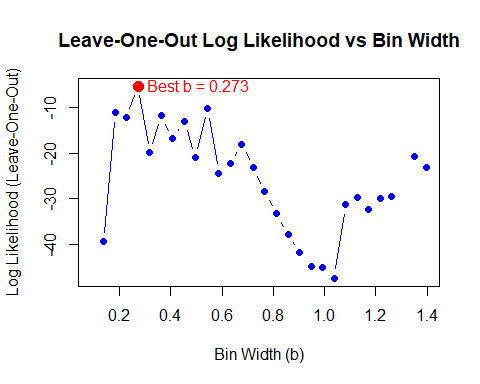
nbr\_seq <- seq(1, 15)  
looCV\_log\_lik <- numeric(length(nbr\_seq)) # Pre-allocate for efficiency  
  
# Loop through each value of nbr  
for (i in seq\_along(nbr\_seq)) {  
 nbr <- nbr\_seq[i]  
   
 # Compute the histogram  
 hx <- hist(x, breaks = seq(A, Z, length = nbr + 1), plot = FALSE)  
 binwidth <- hx$breaks[2] - hx$breaks[1] # Calculate bin width  
   
 # Convert histogram into a step function and evaluate the leave-one-out estimate  
 hx\_f <- stepfun(hx$breaks, c(0, hx$density, 0))  
 y\_loo\_hist <- (hx\_f(x) - (1 / (length(x) \* binwidth))) \* (length(x) / (length(x) - 1))  
   
 # Calculate the log-likelihood and handle zero values  
 if (any(y\_loo\_hist <= 0)) {  
 looCV\_log\_lik[i] <- -Inf  
 } else {  
 looCV\_log\_lik[i] <- sum(log(y\_loo\_hist))  
 }  
}  
  
# Identify the optimal number of bins  
optimal\_nbr <- nbr\_seq[which.max(looCV\_log\_lik)]  
optimal\_log\_lik <- max(looCV\_log\_lik)  
  
# Plot the leave-one-out log-likelihood with a marker for the optimal bin count  
plot(nbr\_seq, looCV\_log\_lik, type = "b", xlab = 'Number of Bins', ylab = 'Log Likelihood (Leave-One-Out)',  
 main = "Leave-One-Out Log Likelihood vs Number of Bins")  
  
# Highlight the optimal number of bins on the plot  
points(optimal\_nbr, optimal\_log\_lik, col = "red", pch = 19, cex = 1.5)  
text(optimal\_nbr, optimal\_log\_lik, labels = paste("Best nbr =", optimal\_nbr), pos = 4, col = "red")



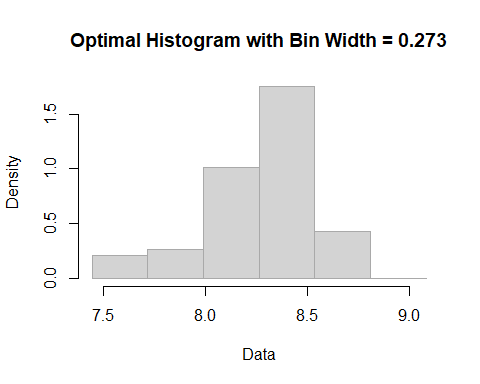
# Exercise 6

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\_seq <- seq((Z - A) / 15, (Z - A), length = 30)  
looCV\_log\_lik <- numeric(length(b\_seq)) # Pre-allocate for efficiency  
  
# Loop through each bin width b  
for (i in seq\_along(b\_seq)) {  
 b <- b\_seq[i]  
   
 # Compute the histogram with the specified bin width b  
 hx <- hist(x, breaks = seq(A, Z + b, by = b), plot = FALSE)  
 binwidth <- diff(hx$breaks)[1] # Confirming the bin width  
   
 # Convert histogram into a step function and evaluate at data points  
 hx\_f <- stepfun(hx$breaks, c(0, hx$density, 0))  
 y\_hist <- hx\_f(x)  
   
 # Compute the leave-one-out histogram estimate  
 y\_loo\_hist <- (y\_hist - (1 / (length(x) \* binwidth))) \* (length(x) / (length(x) - 1))  
   
 # Calculate log-likelihood, handling cases with zero or negative values  
 if (any(y\_loo\_hist <= 0)) {  
 looCV\_log\_lik[i] <- -Inf  
 } else {  
 looCV\_log\_lik[i] <- sum(log(y\_loo\_hist))  
 }  
}  
  
# Identify the optimal bin width  
optimal\_b <- b\_seq[which.max(looCV\_log\_lik)]  
optimal\_log\_lik <- max(looCV\_log\_lik)  
  
# Plot the leave-one-out log-likelihoods for different bin widths  
plot(b\_seq, looCV\_log\_lik, type = "b", xlab = 'Bin Width (b)', ylab = 'Log Likelihood (Leave-One-Out)',  
 main = "Leave-One-Out Log Likelihood vs Bin Width", col = "blue", pch = 19)  
  
# Highlight the optimal bin width on the plot  
points(optimal\_b, optimal\_log\_lik, col = "red", pch = 19, cex = 1.5)  
text(optimal\_b, optimal\_log\_lik, labels = paste("Best b =", round(optimal\_b, 3)), pos = 4, col = "red")



# Plot the histogram using the optimal bin width  
hist(x, breaks = seq(A, Z + optimal\_b, by = optimal\_b), freq = FALSE,   
 main = paste("Optimal Histogram with Bin Width =", round(optimal\_b, 3)),   
 xlab = "Data", ylab = "Density", col = "lightgray", border = "darkgray")



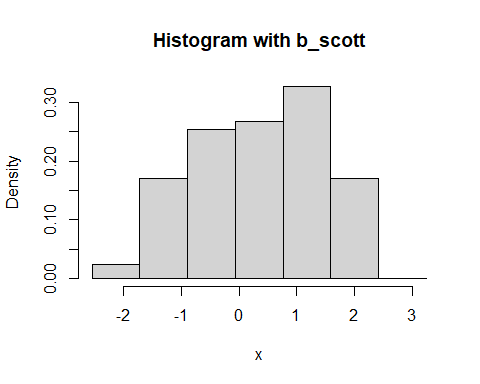
# Exercise 7

Generate data from

par(mfrow=c(1,1))  
  
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))  
}  
  
set.seed(123)  
n <- 100  
x <- sim.mixt(n = n, k = 2, mu = c(0, 1.5), sigma = c(1, 1/3), alpha = c(3/4, 1/4))  
  
A <- min(x) - 0.05 \* diff(range(x))  
Z <- max(x) + 0.05 \* diff(range(x))  
  
sigma.mixt <- 1.095287  
scott\_b <- 3.49 \* sigma.mixt \* length(x)^(-1/3)

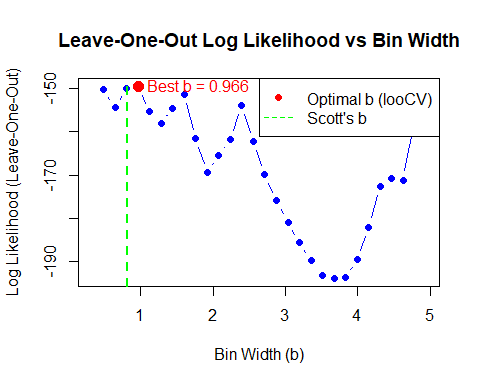
We use the Scott’s formula to find the value of b = rscott\_b` (this was done in the density estimation script used in class). The following is the histogram result:

hx <- hist(x,breaks=seq(A,Z+b,by=scott\_b), plot=F)  
plot(hx,freq = FALSE, main="Histogram with b\_scott")

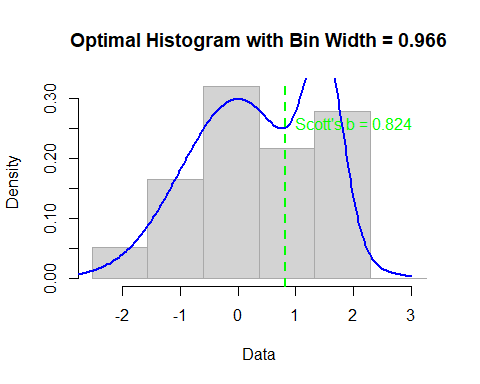


Now taking the b value maximizing the leave-one-out log-likelihood function. The sequence of proposed b-values to select from is varied from 0.05 to 1.75 in steps of 0.01.

b\_seq <- seq((Z - A) / 15, (Z - A), length = 30)  
looCV\_log\_lik <- numeric(length(b\_seq))  
  
for (i in seq\_along(b\_seq)) {  
 b <- b\_seq[i]  
   
 # Compute the histogram with the specified bin width b  
 hx <- hist(x, breaks = seq(A, Z + b, by = b), plot = FALSE)  
 binwidth <- diff(hx$breaks)[1] # Confirm the bin width  
   
 # Convert histogram into a step function and evaluate at data points  
 hx\_f <- stepfun(hx$breaks, c(0, hx$density, 0))  
 y\_hist <- hx\_f(x)  
   
 # Compute the leave-one-out histogram estimate  
 y\_loo\_hist <- (y\_hist - (1 / (n \* binwidth))) \* (n / (n - 1))  
   
 # Calculate log-likelihood, handling cases with zero or negative values  
 if (any(y\_loo\_hist <= 0)) {  
 looCV\_log\_lik[i] <- -Inf  
 } else {  
 looCV\_log\_lik[i] <- sum(log(y\_loo\_hist))  
 }  
}  
  
par(mfrow=c(1,1))  
  
optimal\_b <- b\_seq[which.max(looCV\_log\_lik)]  
optimal\_log\_lik <- max(looCV\_log\_lik)  
  
plot(b\_seq, looCV\_log\_lik, type = "b", xlab = 'Bin Width (b)', ylab = 'Log Likelihood (Leave-One-Out)',  
 main = "Leave-One-Out Log Likelihood vs Bin Width", col = "blue", pch = 19)  
  
points(optimal\_b, optimal\_log\_lik, col = "red", pch = 19, cex = 1.5)  
text(optimal\_b, optimal\_log\_lik, labels = paste("Best b =", round(optimal\_b, 3)), pos = 4, col = "red")  
abline(v = scott\_b, col = "green", lwd = 2, lty = 2)  
legend("topright", legend = c("Optimal b (looCV)", "Scott's b"), col = c("red", "green"), pch = c(19, NA), lty = c(NA, 2))

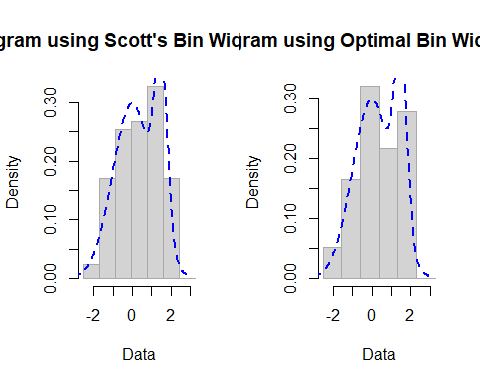


hist(x, breaks = seq(A, Z + optimal\_b, by = optimal\_b), freq = FALSE,   
 main = paste("Optimal Histogram with Bin Width =", round(optimal\_b, 3)),   
 xlab = "Data", ylab = "Density", col = "lightgray", border = "darkgray")  
  
# Overlay the mixture density from graph.mixt  
mixture\_density <- graph.mixt(k = 2, mu = c(0, 1.5), sigma = c(1, 1/3), alpha = c(3/4, 1/4), graphic = FALSE)  
lines(mixture\_density$x, mixture\_density$fx, col = "blue", lwd = 2)  
  
# Annotate Scott's bin width on the histogram  
abline(v = scott\_b, col = "green", lwd = 2, lty = 2)  
text(scott\_b, max(hist(x, breaks = seq(A, Z + optimal\_b, by = optimal\_b), plot = FALSE)$density) \* 0.8,  
 labels = paste("Scott's b =", round(scott\_b, 3)), col = "green", pos = 4)



From this experiment, the leave-one-out cross-validation max log likelihood method selects b = roptimal\_b` as the optimal value for b (out of the proposed values for b). And we give the different histograms that are produced with that bin value.

par(mfrow=c(1,2))  
  
hist(x, breaks = seq(A, Z + scott\_b, by = scott\_b), freq = FALSE,   
 main = paste("Histogram using Scott's Bin Width =", round(scott\_b, 3)),  
 xlab = "Data", ylab = "Density", col = "lightgray", border = "darkgray")  
lines(mixture\_density$x, mixture\_density$fx, col = "blue", lwd = 2, lty = 2) # Overlay the theoretical density  
  
hist(x, breaks = seq(A, Z + optimal\_b, by = optimal\_b), freq = FALSE,   
 main = paste("Histogram using Optimal Bin Width =", round(optimal\_b, 3)),  
 xlab = "Data", ylab = "Density", col = "lightgray", border = "darkgray")  
lines(mixture\_density$x, mixture\_density$fx, col = "blue", lwd = 2, lty = 2) # Overlay the theoretical density



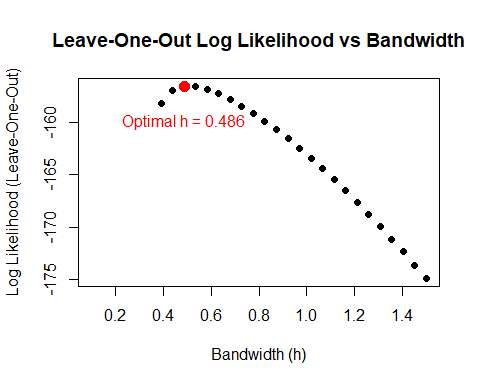
The two suggested values for 𝑏 differ significantly, as seen in the resulting histograms and density overlays. These variations in bin width create distinct impressions of the data. When using Scott’s formula, the mixture of the two distributions (in this case, Gaussian distributions) appears less pronounced, 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. This difference illustrates how bin width can greatly influence the ability to distinguish features within the data, particularly when analyzing complex or multimodal distributions.

# 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.

par(mfrow=c(1,1))  
  
h\_seq <- seq(0.1, 1.5, length.out = 30)  
looCV\_log\_lik <- numeric(length(h\_seq))  
K0 <- 1  
  
for (i in seq\_along(h\_seq)) {  
 h <- h\_seq[i]  
   
 # Estimate the kernel density with the current bandwidth  
 kx <- density(x, bw = h)  
 kx\_f <- approxfun(x = kx$x, y = kx$y, method = 'linear', rule = 2)  
   
 # Calculate the leave-one-out estimate for each data point  
 y\_loo <- (kx\_f(x)-(K0/(n\*h)))\*(n/(n-1))  
   
 # Calculate log-likelihood, handling cases with zero or negative values  
 if (any(y\_loo <= 0)) {  
 looCV\_log\_lik[i] <- -Inf  
 } else {  
 looCV\_log\_lik[i] <- sum(log(y\_loo))  
 }  
}  
  
optimal\_h <- h\_seq[which.max(looCV\_log\_lik)]  
optimal\_log\_lik <- max(looCV\_log\_lik)  
  
plot(h\_seq, looCV\_log\_lik, type = "b", xlab = "Bandwidth (h)", ylab = "Log Likelihood (Leave-One-Out)",  
 main = "Leave-One-Out Log Likelihood vs Bandwidth", pch = 19)  
points(optimal\_h, optimal\_log\_lik, col = "red", pch = 19, cex = 1.5)  
text(optimal\_h, optimal\_log\_lik, labels = paste("Optimal h =", round(optimal\_h, 3)), pos = 1, offset = 1.5, col = "red")



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

kx <- density(x,bw=optimal\_h,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')

