Assignment 2 - Question 1

Toni Ciobanu (20910287)

2024-02-17

QUESTION 1: Implicit Attributes with Optimal Kernel Bandwidth

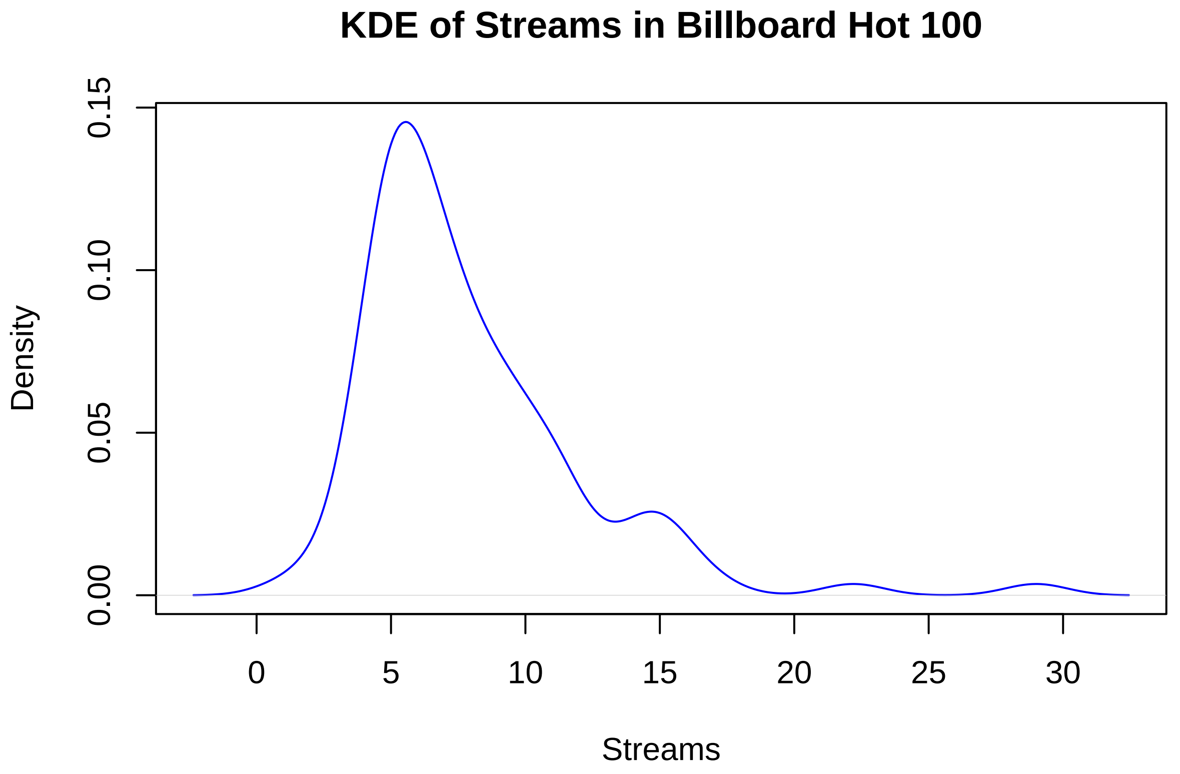
(a) [2 points] Create a plot of the kernel density estimate for the `Streams` variate using the Gaussian kernel and bandwidth selected by Silverman's rule of thumb. You *can* use the `density()` function on this assignment. Remember to include informative titles for the plot and both axes.

SOLUTION:

```r
# Read Data
bh100_jan20 <- read.csv("bh100_jan20.csv")

# Calculate KDE
bh100_KDE <- density(bh100_jan20$Streams, kernel = "gaussian")

# Plot KDE
plot(bh100_KDE, col = "blue", main = "KDE of Streams in Billboard Hot 100",
     xlab = "Streams", ylab = "Density")
```



(b) [5 points] Create a function `cost()` that returns the value of the above function for a given bandwidth `h` and the variates $\{y_1, \ldots, y_N\}$. Obtain the optimal bandwidth with respect to this cost function for the streams variate; you can use an optimizer like `optim()` to do so. Output the optimal bandwidth h you found.

SOLUTION:

```
# Define Cost Function
cost <- function(h, y) {
  N <- length(y)
  sum_sum <- 0
  # N-1 because we need a value to compare to
  for (i in 1:(N-1)) {
    for (j in (i+1):N) {
      numerator <- -(y[i] - y[j])^2
      sum_sum <- sum_sum + exp(numerator/(4*h^2)) -
        2 * sqrt(2) * exp(numerator/(2*h^2))
    }
  }
  return((1 / 2 * sqrt(pi) * N^2) *
           (N / h + (2 / h) * sum_sum))
}
```

```
# Calculate Optimal Bandwidth
opt_h <- optim(par = 1, fn = cost, y = bh100_jan20$Streams)$par
```

```
## Warning in optim(par = 1, fn = cost, y = bh100_jan20$Streams): one-dimensional optimization by Nelde
## use "Brent" or optimize() directly
```

```
opt_h
```

```
## [1] 0.6730469
```

(c) [4 points] Replicate your plot from part (a) and add a second density curve in a different colour for the streaming data created using a Gaussian kernel with the optimal bandwidth found in part (b).

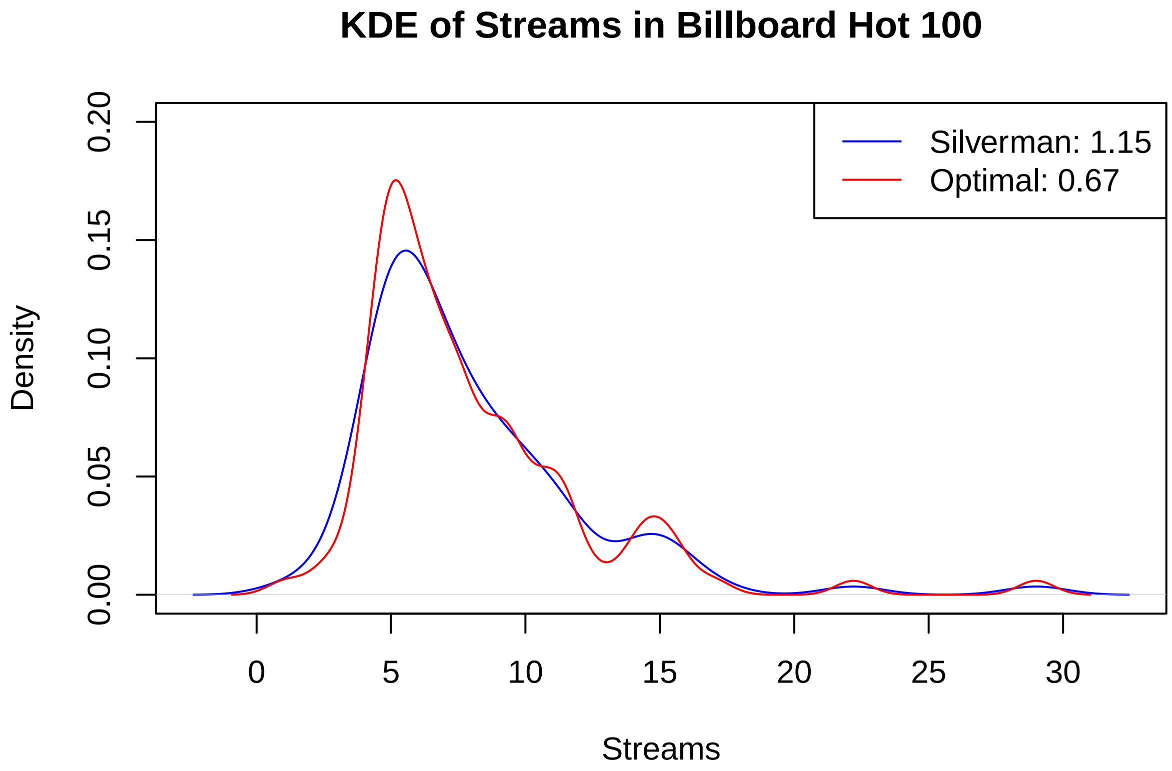
SOLUTION:

```
# Plot KDE with lim increased to account for Opt_h KDE
plot(bh100_KDE, col = "blue", main = "KDE of Streams in Billboard Hot 100",
     xlab = "Streams", ylab = "Density", ylim = c(0,0.2))

# Optimal h density
opt_h_KDE <- density(bh100_jan20$Streams, kernel = "gaussian", bw = opt_h)

# Plot Optimal h KDE
lines(opt_h_KDE, col = "red")

# Add Legend
legend("topright",
       legend = c(
         paste("Silverman:", round(bh100_KDE$bw,2)),
         paste("Optimal:", round(opt_h,2))),
       col = c("blue", "red"), lty = 1)
```



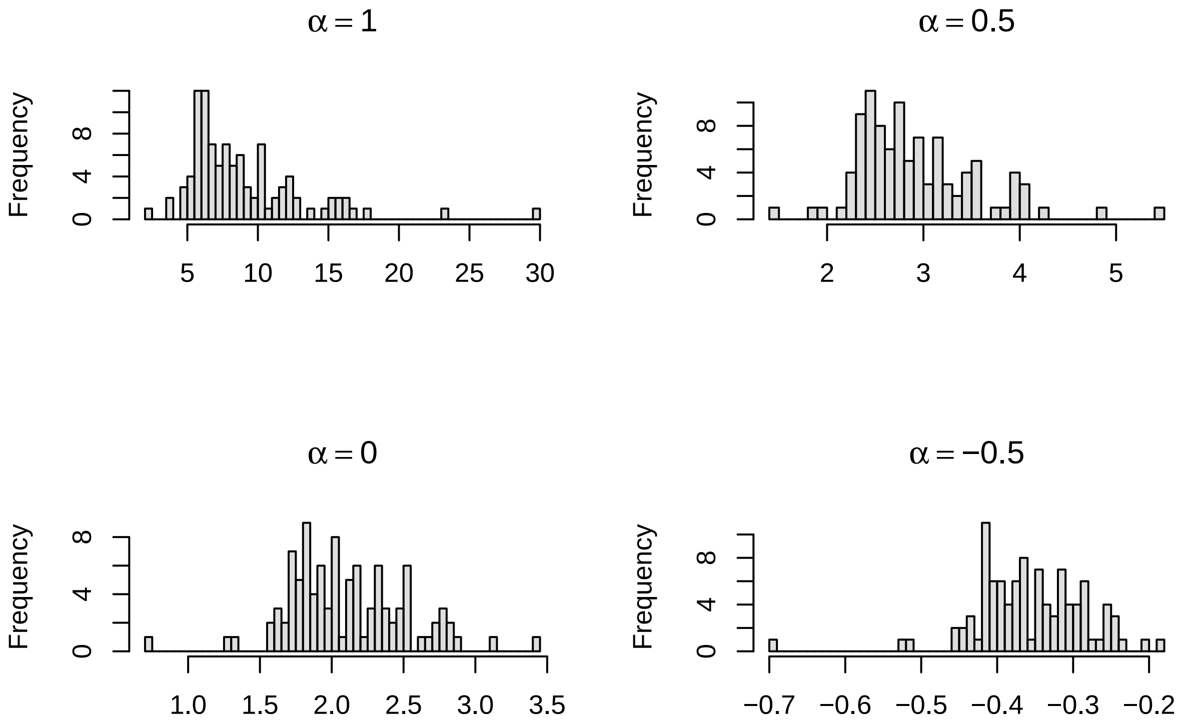
It appears that changing the bandwidth to the Optimal bandwidth using the formula in part b caused the density of approximately 5 to increase. It also seemed to vary more in the ranges between 12-17.

(d) [5 points] Implement the bump rule for histograms to make the streaming variate more symmetric.

SOLUTION:

```r
# Define Powerfun function from course notes
powerfun <- function(y, alpha) {
  if(sum(y <= 0) > 0) stop("y must be positive")
  if (alpha == 0)
    log(y)
  else if (alpha > 0) {
    y^alpha
  } else -y^alpha
}

# Varying the power on the number of Streams for the Billboard 100
par(mfrow=c(2,2))
a = seq(1, -0.5, length.out=4)
for (i in 1:4) {
hist( powerfun(bh100_jan20$Streams + 1, a[i]), col=adjustcolor("grey", alpha = 0.5),
     main= bquote(alpha == .(a[i])), xlab="", breaks=50 )
}
```



Starting at $\alpha = 1$, I tested the histogram plots by increasing alpha at first, then saw that the histograms kept getting more condensed. I then tested all alpha values below 1, and found that my preferred alpha transformation is 0 because it looks the most normal of all other histograms.

(e) [4 points] Use `cost()` from part (b) to find the optimal bandwidth $\hat{h}$ for the streaming data transformed using your preferred value of α from part (d). For the transformed data, create a plot that compares two Gaussian kernel density estimates: one with h selected via Silverman's rule of thumb and the other with $\hat{h}$ found in this question. This plot should be formatted as in part (c). What do you notice about the two density estimates for the transformed data?

SOLUTION:

```
# Transform the data using alpha = 0
new_streams <- log(bh100_jan20$Streams + 1)

# Calculate optimal bandwidth
opt_h_new <- optim(par = 1, fn = cost, y = new_streams)$par
```

```
## Warning in optim(par = 1, fn = cost, y = new_streams): one-dimensional optimization by Nelder-Mead i
## use "Brent" or optimize() directly
```

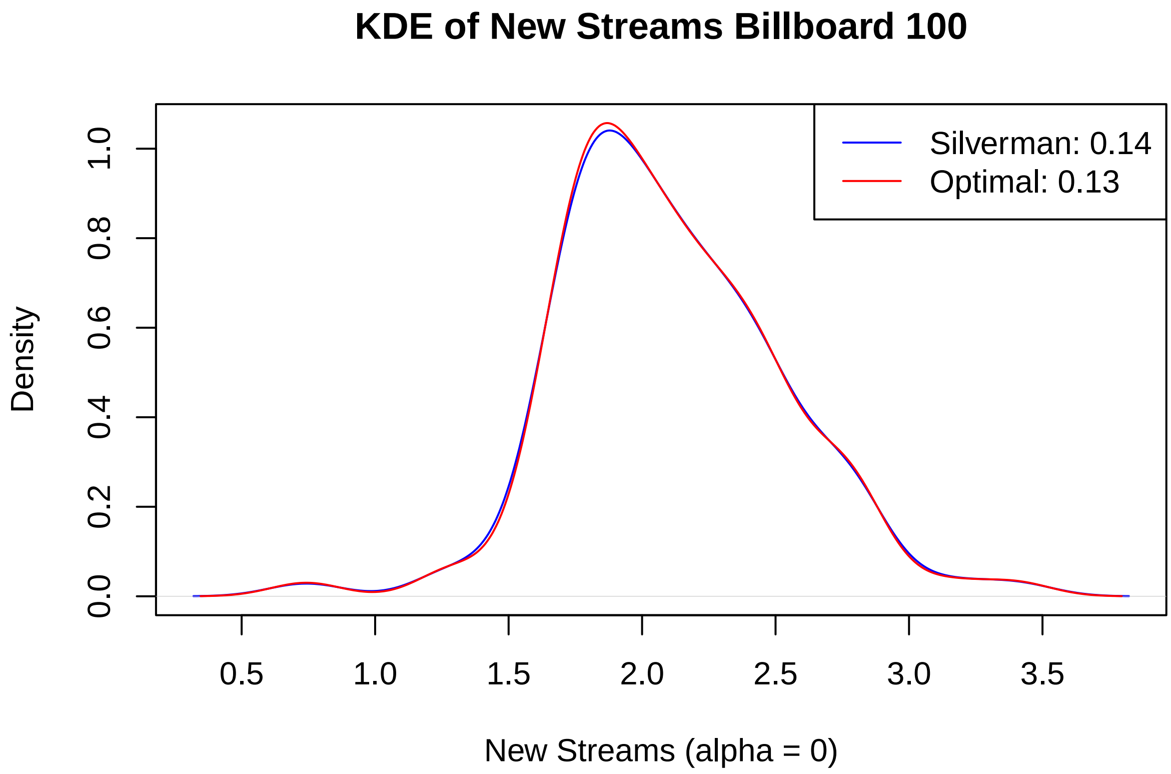
```
# KDE with Silverman's rule
bh100_KDE_new <- density(new_streams, kernel = "gaussian")
```

```r
# KDE with optimal h
opt_h_KDE_new <- density(new_streams, kernel = "gaussian", bw = opt_h_new)

# Plot new graph
plot(bh100_KDE_new, col = "blue", main = "KDE of New Streams Billboard 100",
     xlab = "New Streams (alpha = 0)", ylab = "Density",
     ylim = c(0, max(bh100_KDE_new$y, opt_h_KDE_new$y)))

lines(opt_h_KDE_new, col = "red")

legend("topright", legend = c(
  paste("Silverman:", round(bh100_KDE_new$bw, 2)),
  paste("Optimal:", round(opt_h_new, 2))),
  col = c("blue", "red"), lty = 1)
```



I notice that the 2 density estimates for the transformed data are more similar than before. They now both have relatively the same smoothness, and peak positions.