Aula 9: Exemplos de Estimação de Densidade de Probabilidade

Prof. Dr. Eder Angelo Milani

29/05/2023

Estimação de Densidades pelo Método Kernel Gaussiano

A expressão do método Gaussiano é dado por

$$K(t) = \frac{1}{\sqrt{2\pi}} \exp\Big(-\frac{1}{2}t^2\Big).$$

A função é definida da seguinte forma.

```
kernel_Gaussiano = function(X, h, x){

  n <- length(X)
  t <- (x-X)/h
  w <- (2*pi)^(-0.5)*exp(-0.5*t^2)
  f <-  (1/n)*(1/h)*sum(w)
  return(f)
}
```

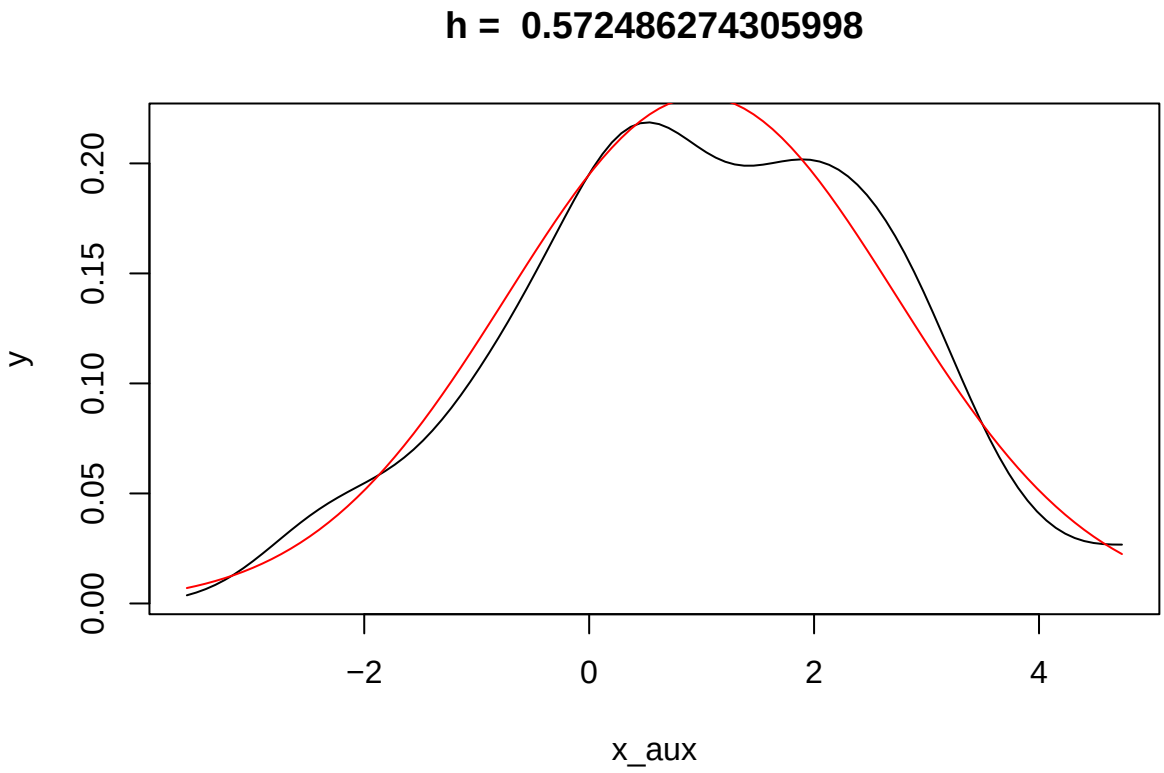
Dados simulados

Para a exemplificação, consideramos inicialmente uma amostra com 100 observações de uma N(1,3). Para a largura da banda, h, atribuiremos o sugerido por Silverman (1978).

```
set.seed(2023)

# Definindo a distribuição
n     <-  100
media <-  1
SD    <-  sqrt(3)
X     <-  rnorm(n = n, mean = media, sd = SD)

# Valores do eixo X
x_aux  <-  seq(min(X)-1, max(X)-1, length.out = 100)
y <-  numeric(100)

# definindo o valor de h
h <- 0.9*min(sd(X),IQR(X)/1.34)*n^(-1/5)
h
```

```
## [1] 0.5724863
```

```
for(i in 1:100) y[i] <- kernel_Gaussiano(X=X, h=h, x_aux[i])
plot(x_aux, y, type="l", main = paste("h = ", h))
lines(x_aux, dnorm(x_aux, media, SD), col = "red")
```

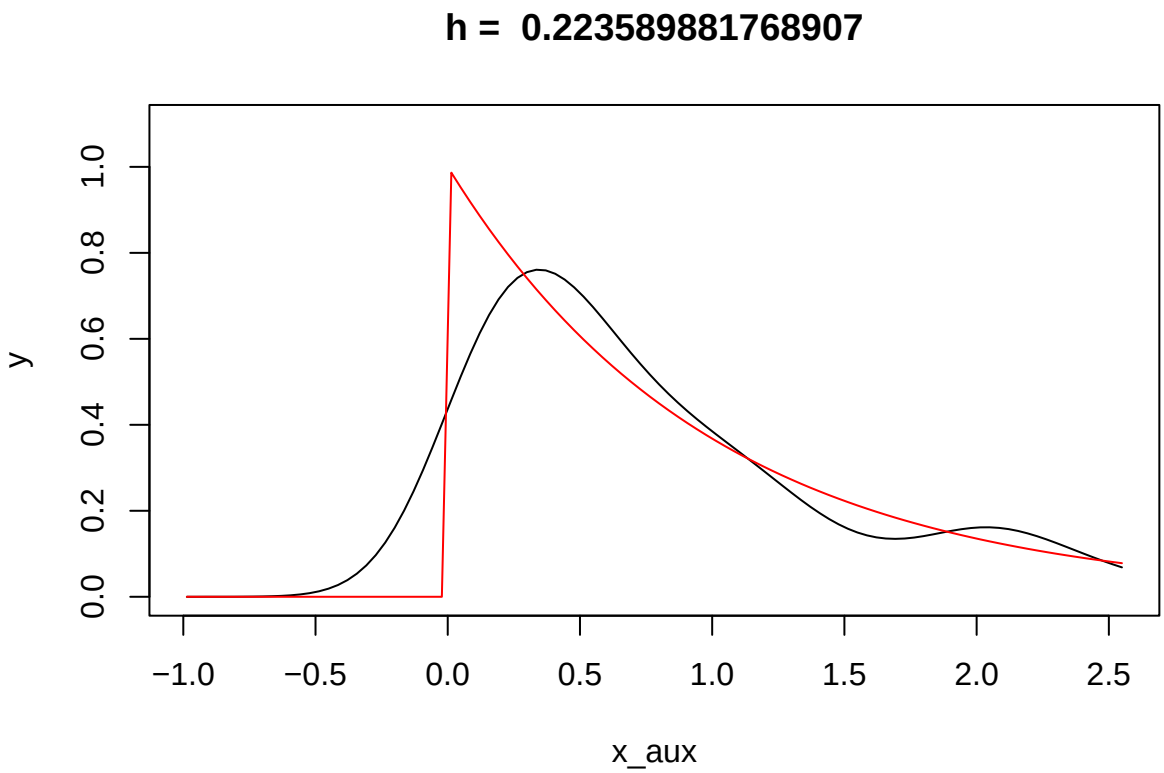


Agora, vamos exemplificar considerando uma amostra com 300 observações de uma Exp(1). Para a largura da banda, h, atribuiremos o sugerido por Silverman (1978).

```
set.seed(2023)

# Definindo a distribuição
n     <-  100
X     <-  rexp(n = n, 1)

# Valores do eixo X
x_aux  <-  seq(min(X)-1, max(X)-1, length.out = 100)
y <-  numeric(100)

# definindo o valor de h
h <- 0.9*min(sd(X),IQR(X)/1.34)*n^(-1/5)
h
```

```
## [1] 0.2235899
```

```r
for(i in 1:100) y[i] <- kernel_Gaussiano(X=X, h=h, x_aux[i])
plot(x_aux, y, type="l", main = paste("h = ", h), ylim=c(0, 1.1))
lines(x_aux, dexp(x_aux, 1), col = "red")
```



Utilizando o conjunto de dados geyser

Uma versão dos dados de erupções do gêiser 'Old Faithful' no Parque Nacional de Yellowstone, Wyoming. Esta versão vem de Azzalini e Bowman (1990) e é de medição contínua de 1º de agosto a 15 de agosto de 1985.

Algumas medidas de duração noturna foram codificadas como 2, 3 ou 4 minutos, tendo sido originalmente descritas como 'curta', 'média' ou 'longa'.

```r
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

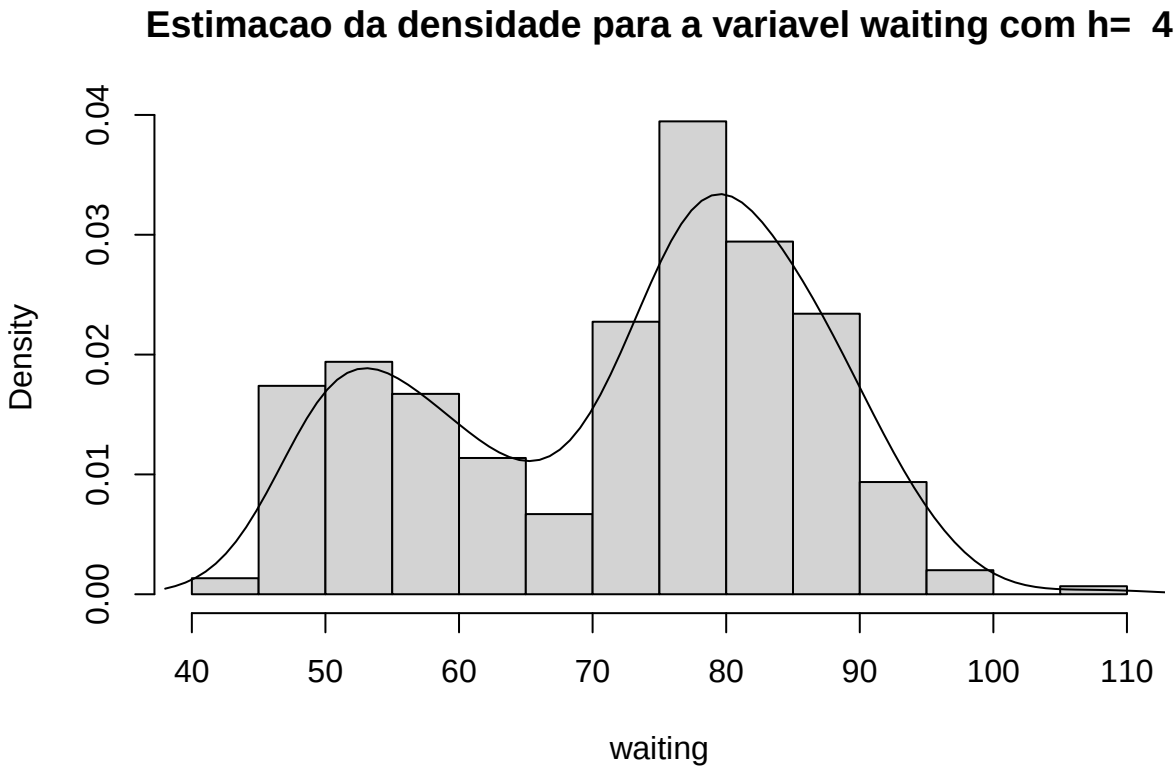
```
waiting <- geyser$waiting
duration <- geyser$duration
n <- length(waiting)
h_w <- 0.9*min(sd(waiting), IQR(waiting)/1.35)*n^(-1/5)
h_d <- 0.9*min(sd(duration), IQR(duration)/1.35)*n^(-1/5)

cat("h_w=", h_w, "\n")
```

```
## h_w= 3.997796
```

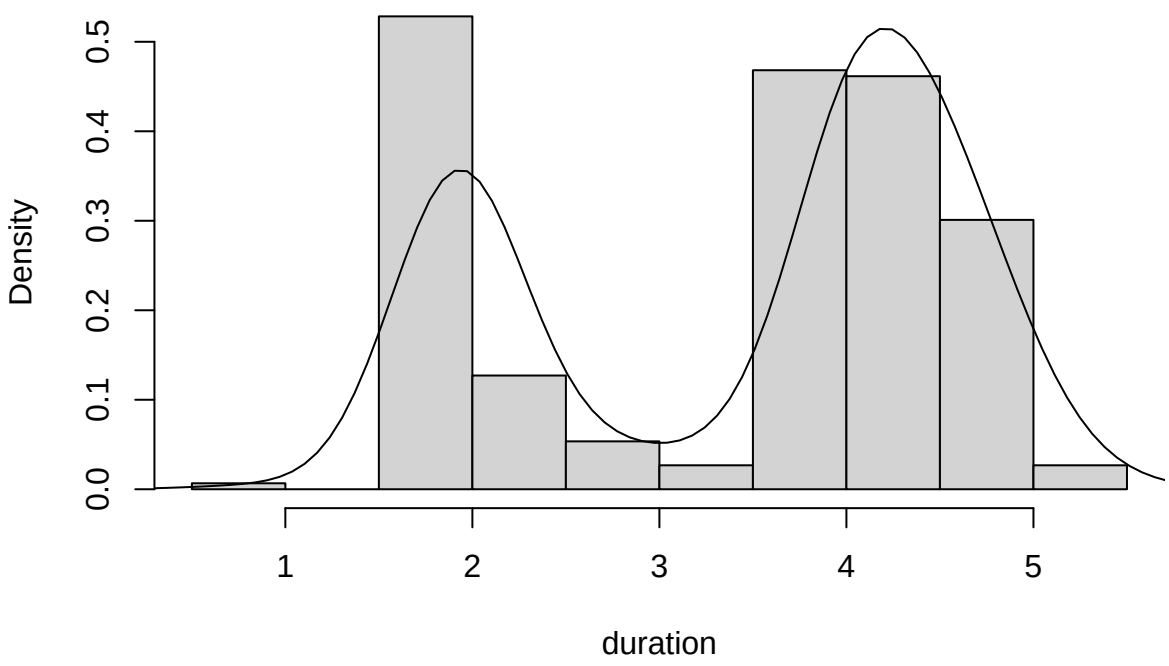
```
cat("h_d=", h_d, "\n")
```

```
## h_d= 0.33038
```

```
x_aux  <-  seq(min(waiting)-5, max(waiting)+5, length.out = 100)
y <-  numeric(100)

hist(waiting, freq = F, main = paste("Estimacao da densidade para a variavel waiting com h= ", round(h_
for( i in 1:100) y[i] <-  kernel_Gaussiano(X=waiting, h=h_w, x_aux[i])
points(x_aux, y, type="l")
```




```
x_aux  <-  seq(min(duration)-1, max(duration)+1, length.out = 100)
y <-  numeric(100)
hist(duration, freq = F, , main = paste("Estimacao da densidade para a variavel duration com h= ", roun
for( i in 1:100) y[i] <- kernel_Gaussiano(X=duration, h=h_d, x_aux[i])
points(x_aux, y, type="l")
```



Mas se fosse utilizar um outro kernel, como definir o h?

Para redimensionamento de kernel equivalente, a largura de banda h_2 pode ser redimensionada utilizando a expressão

$$h_2 \approx \frac{\sigma_{K_1}}{\sigma_{K_2}} h_1.$$

Assim, a partir do kernel Gaussiano podemos ter um ponto de partida para os demais utilizando a variância que foi apresentada na Tabela resumo da Aula 8.

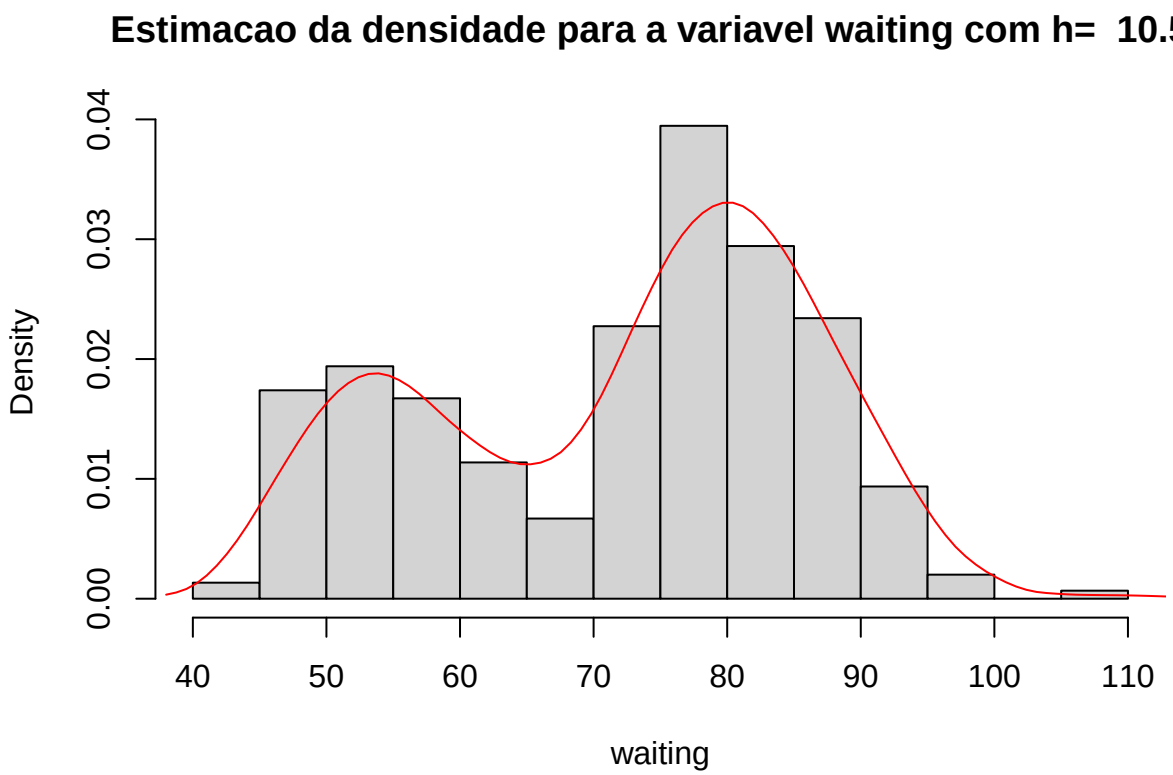
Por exemplo, a seguir utilizamos o kernel Biweight com a largura da banda definida de maneira equivalente.

O kernel biweight é dada por

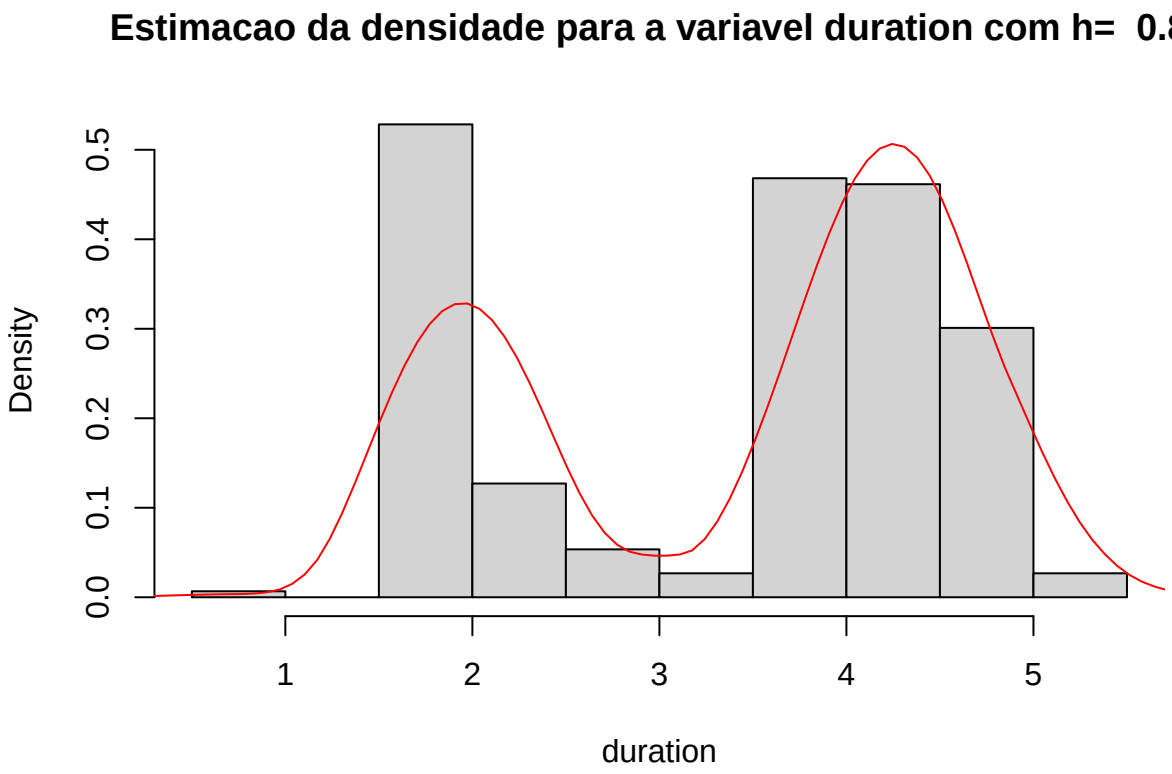
$$\frac{15}{16}(1-t^2)^2.$$

```
kernel_Biweight <- function(X, h, x){
  n <- length(X)
  t <- (x-X)/h
  w <- ifelse(abs(t)<1, 15*(1-t^2)^2/16, 0)
  f <- (1/n)*(1/h)*sum(w)
  return(f)
}
```

```r
waiting <- geyser$waiting
duration <- geyser$duration
n <- length(waiting)
h_w_ <- h_w/sqrt(1/7)
h_d_ <- h_d/sqrt(1/7)

x_aux  <-  seq(min(waiting)-5, max(waiting)+5, length.out = 100)
y <-  numeric(100)

hist(waiting, freq = F, main = paste("Estimacao da densidade para a variavel waiting com h= ", round(h_w
for( i in 1:100) y[i] <- kernel_Biweight(X=waiting, h=h_w_, x_aux[i])
lines(x_aux, y, col="red")
```




```r
x_aux  <-  seq(min(duration)-1, max(duration)+1, length.out = 100)
y <-  numeric(100)
hist(duration, freq = F, main = paste("Estimacao da densidade para a variavel duration com h= ", round(h
for( i in 1:100) y[i]  <-  kernel_Biweight(X=duration, h=h_d_, x_aux[i])
lines(x_aux, y, col="red")
```


Estimacao da densidade para a variavel duration com h= 0.87
Density
0.0
0.1
0.2
0.3
0.4
0.5
1
2
3
4
5
duration