

# Tutorial em Métodos Não Paramétricos para Estimação de Curvas

Ronaldo Dias

*Departamento de Estatística.*

*Universidade Estadual de Campinas. São Paulo, Brasil*

e-mail address: `dias@ime.unicamp.br`

## 1 Estimação de Densidades: Método Kernel

### 1.1 Histograma

Histograma é um dos métodos não paramétricos mais usados (talvez o primeiro a ser usado) em estimação de densidades. Empíricamente, a idéia é dividir o intervalo de variação dos dados em intervalos de comprimento  $h$  e contar o número de observações que caem em cada intervalo. Sem perda de generalidade, considere o intervalo  $[-h/2, h/2)$ . Daí a probabilidade de uma observação cair em  $[-h/2, h/2)$  é dada por

$$P(X \in [-h/2, h/2)) = \int_{-h/2}^{h/2} f(x)dx,$$

onde  $f$  é a densidade de  $X$ . Naturalmente uma estimativa da densidade  $f$  pode ser pensada como contar o número de observações que caem em cada intervalo e dividi-lo pelo número total de observações. Em outras palavras, dado um conjunto de observações,  $X_1, \dots, X_n$ , temos:

$$P(X \in [-h/2, h/2)) \approx \frac{1}{n} \#\{X_i \in [-h/2, h/2)\}.$$

Ou seja,

$$P(X \in [-h/2, h/2)) = \int_{-h/2}^{h/2} f(x)dx = f(\xi)h,$$

com  $\xi \in [-h/2, h/2)$ . Daí uma estimativa para  $f$  seria,

$$\hat{f}_h(x) = \frac{1}{nh} \#\{X_i \in [-h/2, h/2)\},$$

para todo  $x \in [-h/2, h/2)$ .

Formalmente, suponha que observamos  $X_1, \dots, X_n$  i.i.d.  $f$  desconhecida. Seja  $k$  o número de intervalos de comprimento  $h$  e defina  $C_j = [x_0 + (j-1)h, x_0 + jh)$ ,  $j = 1, \dots, k$ . Tome,  $n_j = \sum_{i=1}^n I(X_i \in C_j)$ , tal que,  $\sum_{j=1}^k n_j = n$ . Então,

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{j=1}^k n_j I(x \in C_j),$$

Note que a função  $I(x \in A)$  é

$$I(x \in A) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases}$$

É fácil ver que a estimativa  $\hat{f}_h(x)$  depende fortemente da escolha de  $h$ , chamdo de **parâmetro de suavização**.

Exemplo da função R para histograma

```
x<-rnorm(100)
hist(x,nclass=6,prob=T)
hist(x,nclass=30,prob=T)
hist(x,nclass=50,rpb=T)
```

## 1.2 Estimação pelo método Kernel ou funções núcleo

As idéias e motivações do Histograma sugerem, naturalmente, uma generalização considerando funções do tipo

$$K(x) = \begin{cases} \frac{1}{2}, & \text{if } |x| < 1 \\ 0, & \text{otherwise} \end{cases}$$

Consequentemente, pode-se sugerir uma nova estimativa para  $f$  como sendo

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right).$$

Claramente, esta nova estimativa generaliza aquela baseada no Histograma. Também não é difícil ver que  $\hat{f}$  não é uma função contínua e tem derivada zero em todos os pontos exceto nos pontos de salto  $X_i \pm h$ . A figura 1.1 mostra esta característica de não suavidade deste estimador “ingênuo”

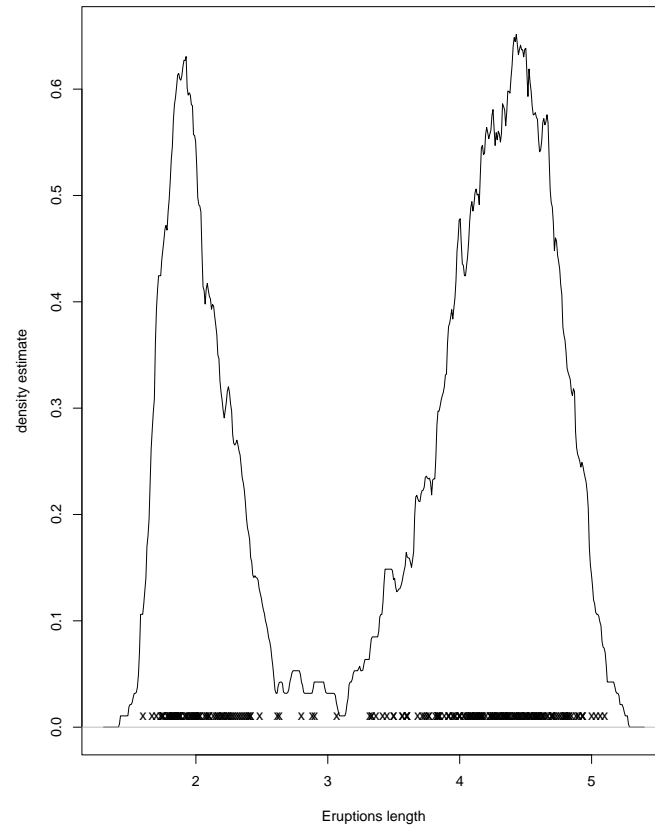


Figure 1.1: Estimativa “ingênuo” usando os dados *old faithful geyser*  $h = 0.1$

Para superar o fato de não suavidade faz-se suposições bastantes razoáveis sobre a função kernel  $K$ . Por exemplo, podemos assumir que  $K$  é uma função de densidade de probabilidade. Em particular quando se faz a escolha de  $K$  como sendo uma Gaussiana,  $\hat{f}$  será uma curva suave com derivadas de todas as ordens.

Figura 1.2 mostra como as funções de núcleo Gaussianas atuam na estimativa final da densidade.

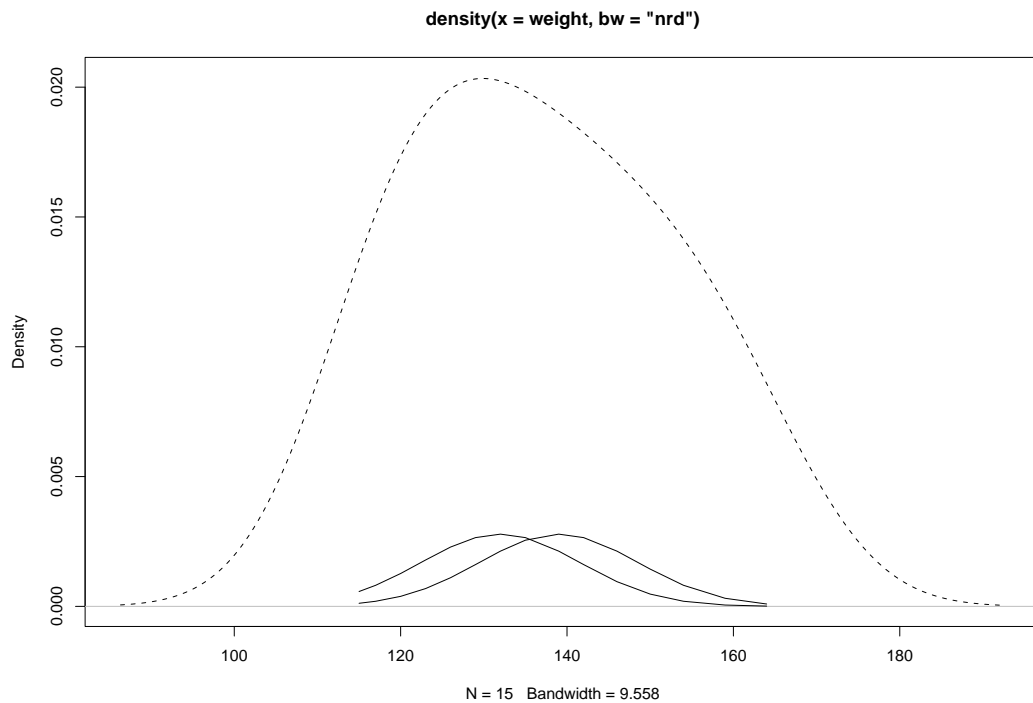


Figure 1.2: Estimativa pelo método de Kernel com algumas funções Kernel

Figura 1.3 mostra as propriedades (curva suave) de  $\hat{f}$  quando kernel Gaussiano é usado.

A estimativa é fortemente influenciada pela escolha de  $h$ . Quanto menor for o tamanho de  $h$  menos suave será a estimativa. De maneira oposta, quanto maior for  $h$  mais suave será a estimativa final.

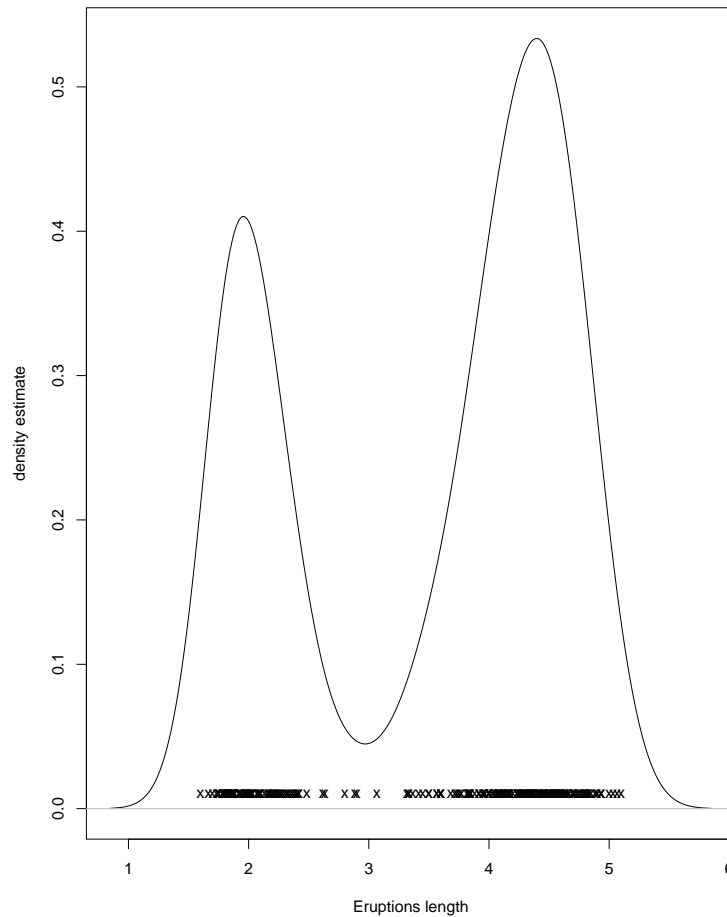


Figure 1.3: Estimativa de densidade por Kernel construída usando os dados Old faithful. Kernel Gaussiano e  $h = 0.25$

### 1.3 Estimativa pelo Vizinho Mais Próximo

Pode-se observar que previamente obtemos estimativas cuja suavidade depende de  $h$  **fixo**, não levando-se a maneira como os dados estão dispersos localmente. Então para a estimativa se ajustar a específicas regiões, seríamos forçados a aumentar ou diminuir o valor de  $h$  e conseqüentemente a estimativa final será afetada.

A idéia por detrás do método do vizinho mais próximo é adaptar a quantidade de suavização às características locais dos dados. Basicamente, o estimador obtido pelo método do vizinho mais próximo, calcula a distância de  $x$  em  $f(x)$  a cada ponto amostral. Por exemplo, seja  $d(x_1, x)$  a distância do ponto amostral  $x_1$  e para cada  $x$  denote  $d_k(x)$  como sendo a distância entre o  $x$  e o  $k$ -ésimo vizinho mais próximo

na amostra  $x_1, \dots, x_n$ . A estimativa pode ser pensada da seguinte forma. Suponha que a densidade no ponto  $x$  é dada por  $f(x)$ . Então, espera-se que cerca de  $2rn f(x)$  observações estarão no intervalo  $[x-r, x+r]$  para cada  $r > 0$ . Por definição, obteremos  $k$  pontos no intervalo  $[x - d_k(x), x + d_k(x)]$ . Assim uma estimativa de  $f$  no ponto  $x$  pode ser obtida por

$$k = 2d_k(x)n\hat{f}(x).$$

Ou seja,

$$\hat{f}(x) = \frac{k}{2d_k(x)n}$$

Um programa para calcular estimativa por K-nn

```
function(x,k)
x <- sort(x)
n <- length(x)
grid<- seq(min(x),max(x),length=n)
d <-rep(0, n)
dens<-rep(0,n)
for(i in 1:n)
d <- abs(grid[i] - x)
xk <- sort(d)
yk <- xk[k + 1]
dens[i] <- k/(2 * n * yk)

result <- matrix(c(grid, dens), n, 2)
result
plot(result,type="l")
```

Generalizando o estimador K-nn de maneira semelhante ao que foi feito com o Histograma obtemos

$$\hat{f}(x) = \frac{1}{nd_k(x)} \sum_{i=1}^n K\left(\frac{x - X_i}{d_k(x)}\right).$$

Observe que a quantidade the suavização é determinada pela escolha de  $k$ .

## 1.4 A escolha do Parâmetro de Suavização

Existem várias maneiras para se fazer uma boa escolha de  $h$ . As que tem sido aceitas comumente pelos os usuários são: A regra prática de Silverman (Silverman 1986) e validação cruzada por máxima verossimilhança. (Ver Härdle (1990))

Para a regra prática de Silverman:

$$\begin{aligned}\hat{h}_{opt} &= 1.06 \min\left(\hat{\sigma}, \frac{\hat{R}}{(\Phi(3/4) - \Phi(1/4))}\right) \\ &= 1.06 \min\left(\hat{\sigma}, \frac{\hat{R}}{1.349}\right),\end{aligned}\tag{1.1}$$

onde  $\Phi$  é a densidade de uma normal padrão.

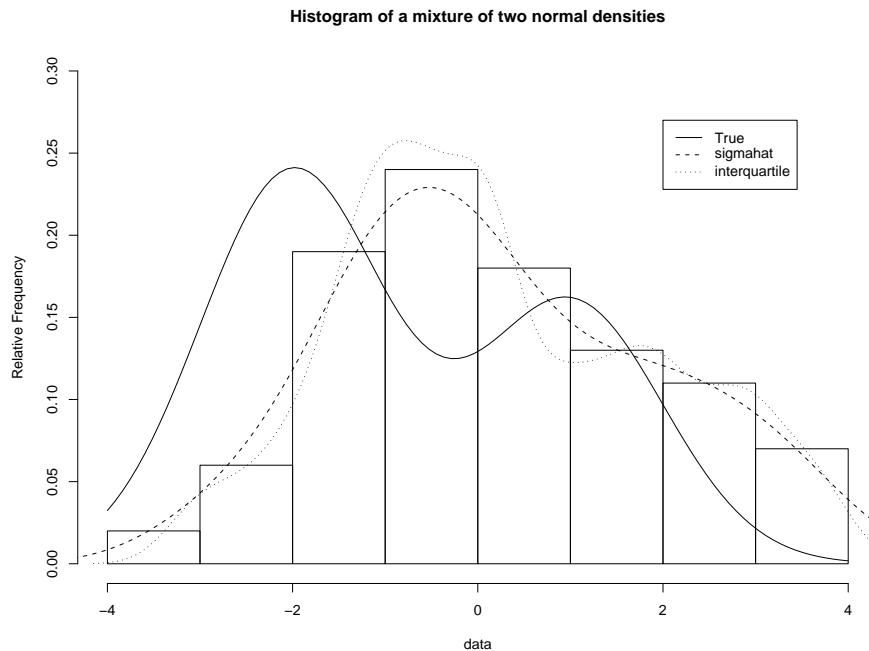


Figure 1.4: Comparação de dois “bandwidth”,  $\hat{\sigma}$  e  $\hat{R}$  (interquartil amostral) para a mistura  $0.7 \times N(-2, 1) + 0.3 \times N(1, 1)$ .

Validação Cruzada (*leaving-one-out*). I idéia básica é retirar uma observação e calcular cada estimativa da densidade a observação retirada. Assim termos varias estimativas de densidade, especificamente:

$$\hat{f}(X_i) = (n-1)^{-1}h^{-1} \sum_{j \neq i} K\left(\frac{X_i - X_j}{h}\right).$$

daí,

$$\prod_{i=1}^n \hat{f}_i(X_i) = (n-1)^{-n}h^{-n} \prod_{i=1}^n \sum_{j \neq i} K\left(\frac{X_i - X_j}{h}\right). \quad (1.2)$$

Ou equivalentemente

$$\begin{aligned} CV_{KL}(h) &= \frac{1}{n} \sum_{i=1}^n \log[f_{h,i}(X_i)] \\ &= \frac{1}{n} \sum_{i=1}^n \log\left[\sum_{j \neq i} K\left(\frac{X_i - X_j}{h}\right)\right] - \log[(n-1)h] \end{aligned} \quad (1.3)$$

De maneira natural escolhemos  $h$  por

$$h_{KL} = \arg \max_h CV_{KL}(h). \quad (1.4)$$

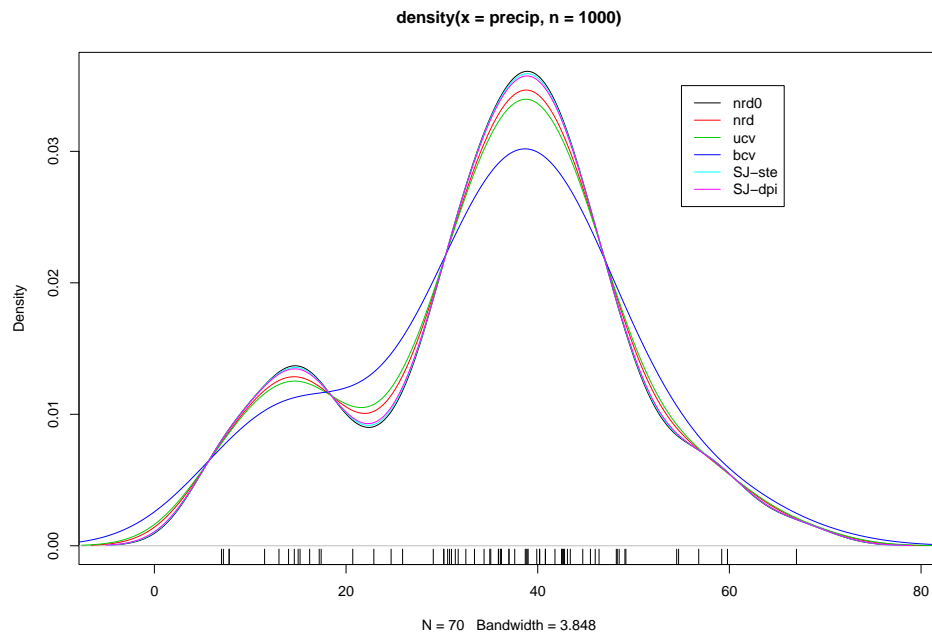


Figure 1.5: Comparação de várias escolhas de  $h$

```

Exemplo de programa para estudo de várias escolhas de  $h$ 
data(precip)
plot(density(precip, n = 1000))
rug(precip)
lines(density(precip, bw="nrd"), col = 2)
lines(density(precip, bw="ucv"), col = 3)
lines(density(precip, bw="bcv"), col = 4)
lines(density(precip, bw="SJ-ste"), col = 5)
lines(density(precip, bw="SJ-dpi"), col = 6)
legend(55, 0.035, legend = c("nrd0", "nrd", "ucv", "bcv", "SJ-ste", "SJ-dpi"),
col = 1:6, lty = 1)

```

## 2 Regressão Não Paramétrica: Método Kernel

Suponha que temos um conjunto de observações  $\{(X_i, Y_i)\}_{i=1}^n$  e acredita-se que exista um relacionamento entre as variáveis  $X_i$  e  $Y_i$  dado por

$$Y_i = g(x_i) + \varepsilon_i \quad i = 1, \dots, n,$$

onde  $\varepsilon$ 's são variáveis não correlacionadas com média zero e independentes de  $Y_i$ . Considerando o fato de que

$$g(x) = \mathbb{E}[Y|X = x].$$

(Nadaraya (1964) and Watson (1964)) sugeriram o seguinte estimador para a curva de regressão  $g(x)$ .

$$g_h(x) = \frac{\sum_{i=1}^n K_h(x - X_i) Y_i}{\sum_{j=1}^n K_h(x - X_j)} \quad (2.1)$$

A função que calcula a estimativa de Nadaraya-Watson

```

ksmooth(x, y, kernel = c("box", "normal"), bandwidth = 0.5, range.x =
range(x), n.points = max(100, length(x)), x.points)

```

Exemplo

```
data(cars)
with(cars, plot(speed, dist)
lines(ksmooth(speed, dist, "normal", bandwidth=2), col=2)
lines(ksmooth(speed, dist, "normal", bandwidth=5), col=3) )
```

Observe que a escolha de  $h$  continua sendo fundamental para determinar a forma final do ajuste.

Uma maneira de tornar este procedimento mais adaptativo aos dados é o usar o princípio do vizinho mais próximo.

$$g_K(x) = \frac{1}{n} \sum_{i=1}^n W_{K_i}(x) Y_i, \quad (2.2)$$

where,

$$W_{K_i}(x) = \begin{cases} n/K & \text{if } i \in J_x \\ 0 & \text{otherwise,} \end{cases} \quad (2.3)$$

com  $J_x = \{i : X_i \text{ um dos } K \text{ vizinho mais próximo a } x\}$

Exemplo de um programa R. De fato este programa aparece primeiramente no livro de Härdle (1990) e pode-se notar ao executar esta rotina que esta contém um erro de programação. Você seria capaz de consertar este programa? Explique sua resposta detalhadamente.

```
function(x,y,k) {
missing.flag<-is.na(x)+is.na(y)
x<-x[!missing.flag]
y<-y[!missing.flag]
#selection of missings
n<-length(x)
order.x<-order(x)
y<-y[order.x]
x<-x[order.x]
#sorting x and y
```

```

start<-1+floor(k/2)
mk<-rep(NA,n)
#fill the vector of estimates with missings
mk[start]<-sum(y[1:k])/k
#at x[start] the first k observations are the k nearest neighbors.
difference<-diff(y,k)
#computes differences of y to the lag k
mk[(start+1):(n-start)]<-cumsum(c(mk[start],difference))
#compute cumulative sum over differences
#result<-list(m=mk,x=x,y=y)
list(m=mk,x=x,y=y)
xx<-seq(min(x),max(x),length=length(mk))
plot(x,y)
lines(xx,mk)
}

```

Um procedimento bastante usado por analistas de dados é o procedimento conhecido como LOWESS. Cleveland (1979) propôs o algoritmo LOWESS (locally weighted scatter plot smoothing). A idéia é começar o procedimento por ajustar por mínimos quadrados um polinômio local e então robustificar o procedimento para obter o ajuste final. Especificamente, pode-se ajustar um polinômio local numa vizinhança de  $x$  resolvendo o seguinte problema de mínimos quadrados

$$n^{-1} \sum_{i=1}^n W_{ki} \left( y_i - \sum_{j=0}^p \beta_j x^j \right)^2, \quad (2.4)$$

onde  $W_{ki}$  denota os pesos de k-NN . Compute os resíduos  $\hat{\epsilon}_i$  e o parâmetro de escala  $\hat{\sigma} = \text{median}(\hat{\epsilon}_i)$ . Defina os pesos robustificados  $\delta_i = K(\hat{\epsilon}_i/6\hat{\sigma})$ , onde  $K(u) = (15/16)(1 - u)^2$ , if  $|u| \leq 1$  and  $K(u) = 0$ , caso contrário. Daí, ajuste uma regressão polinomial como em (2.4) mas com os pesos  $(\delta_i W_{ki}(x))$ . Cleveland sugere que  $p = 1$  dá um bom balanço entre o custo computacional e a necessidade de ajuste flexível para reproduzir a estrutura dos dados. O parâmetro de suavização pode ser determinado por validação cruzada.

A função R que executa LOWESS é

```
lowess(x, y = NULL, f = 2/3, iter=3, delta = 0.01 * diff(range(xy$x[0])))
```

Exemplo:

```
data(cars) plot(cars, main = "lowess(cars)")
lines(lowess(cars), col = 2)
lines(lowess(cars, f=.2), col = 3)
legend(5, 120, c(paste("f = ", c("2/3", ".2"))), lty = 1, col = 2:3)
```

Observe que  $f$  funciona como parâmetro de suavização. A relação entre  $f$  e  $K$  (viz. mais prox.) é dada por

$$K = \lceil n \times f \rceil, \quad f \in (0, 1),$$

e  $n$  é o tamanho da amostra.

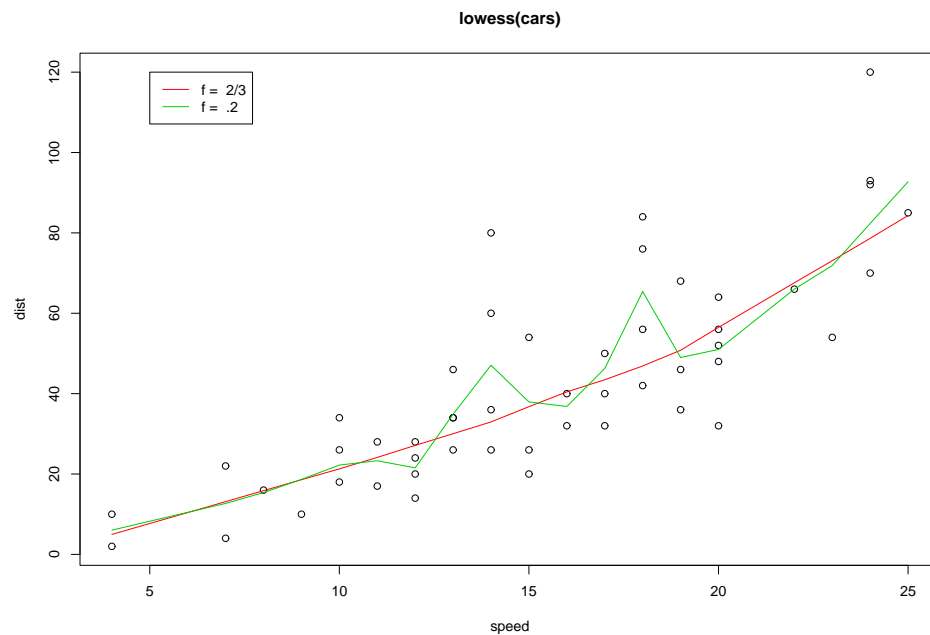


Figure 2.6: Comparação de várias escolhas de  $h$

### 3 Estimação de Densidades por Splines

Um dos métodos de estimação mais eficiente e mais rápido sob o ponto de vista computacional é conhecido como *Log spline* e pode ser calculado como

$$c(\boldsymbol{\theta}) = \log\left(\int_{\mathbb{R}} \exp\left(\sum_{j=1}^{K-1} \theta_j B_j(x) dx\right)\right)$$

e

$$f(x; \boldsymbol{\theta}) = \exp\left\{\sum_{j=1}^{K-1} \theta_j B_j(x) - c(\boldsymbol{\theta})\right\},$$

onde  $B_j$  são os conhecidos B-splines.

O método Log spline pode ser usado em R da seguinte maneira;

```
library(logspline)
y <- rnorm(100)
fit <- logspline(y)
plot(fit)
# as (4 == length(-2, -1, 0, 1, 2) -1), this forces these initial knots,
and does no knot selection
fit <- logspline(y, knots = c(-2, -1, 0, 1, 2), maxknots = 4, penalty =
0)
```

### 3.1 Regressão não paramétrica por Smoothing Splines

Considere o modelo de regressão dado por

$$Y_i = g(x_i) + \varepsilon_i \quad i = 1, \dots, n,$$

onde  $\varepsilon$ 's são variáveis não correlacionadas com média zero e independentes de  $Y_i$ .

O método de regressão que resolve o problema de mínimos quadrados penalizados é denominado *Smoothing Splines*. Ou seja, a estimativa é a solução do problema de otimização dado por:

$$L_\lambda(g) = \|\mathbf{y} - \mathbf{g}\|^2 + \lambda \int (g'')^2, \quad (3.1)$$

onde  $\mathbf{y} = (y_1, \dots, y_n)^T$  e  $\mathbf{g} = (g(x_1), \dots, g(x_n))^T$ .

Smoothing splines é obtido através da função

```
smooth.spline(x, y = NULL, w = NULL, df, spar = NULL, cv = FALSE, all.knots  
= FALSE, nknots = NULL, df.offset = 0, penalty = 1, control.spar = list())
```

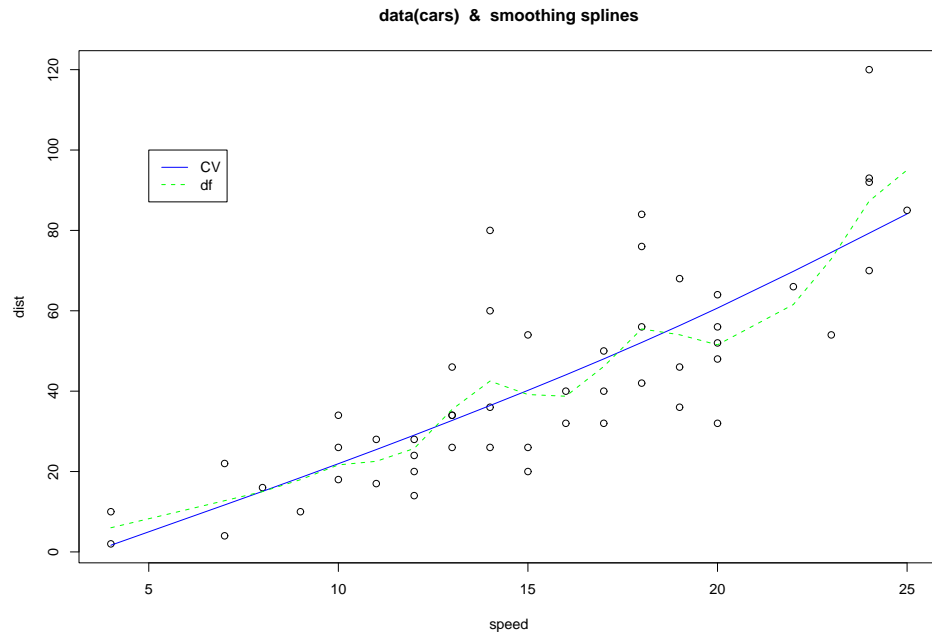


Figure 3.7: Comparação de várias escolhas do parâmetro de suavização

Figura (3.7) mostra estimativas usando o método de smoothing splines com duas opções de escolha do parâmetro de suavização.

## References

- Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots, *J. Amer. Statist. Assoc.* **74**(368): 829–836.
- Härdle, W. (1990). *Smoothing Techniques With Implementation in S*, Springer-Verlag (Berlin, New York).
- Nadaraya, E. A. (1964). On estimating regression, *Theory of probability and its applications* **10**: 186–190.
- Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*, Chapman and Hall (London).
- Watson, G. S. (1964). Smooth regression analysis, *Sankya A* **26**: 359–372.