IV. Regressão Não Paramétrica

Última atualização: 26 de janeiro de 2019.

Correlação e regressão estão intimamente relacionadas. Na correlação estamos principalmente preocupados com os aspectos qualitativos das relações possíveis. A regressão preocupa-se com os aspectos quantitativos das relações, tais como determinar a inclinação e intercepto de uma linha reta que fornece um ajuste adequado para dados fornecidos. Caso o ajuste considerado adequado seja um polinômio de grau p é necessário fornecer valores para as $p+1$ constantes que determinam um polinômio de melhor ajuste. Há sim equivalência entre alguns aspectos das duas abordagens, por exemplo, em regressão linear um teste de inclinação zero é equivalente a um teste de correlação zero. Valores de $+1$ ou -1 para o coeficiente de correlação do produto de momentos da Pearson nos diz que todos os pontos observados estão em uma linha reta.

Estudaremos aqui a regressão não paramétrica e para isso vamos considerar n pares de observações $(x_1, Y_1), \cdots, (x_n, Y_n)$ como nas figuras abaixo. A variável aleatória resposta Y está relacionada à covariável determinística x pelas equações

$$Y_i = r(x_i) + \epsilon_i, \qquad \mathrm{E}(\epsilon_i) = 0, \qquad i = 1, 2, \cdots, n,$$

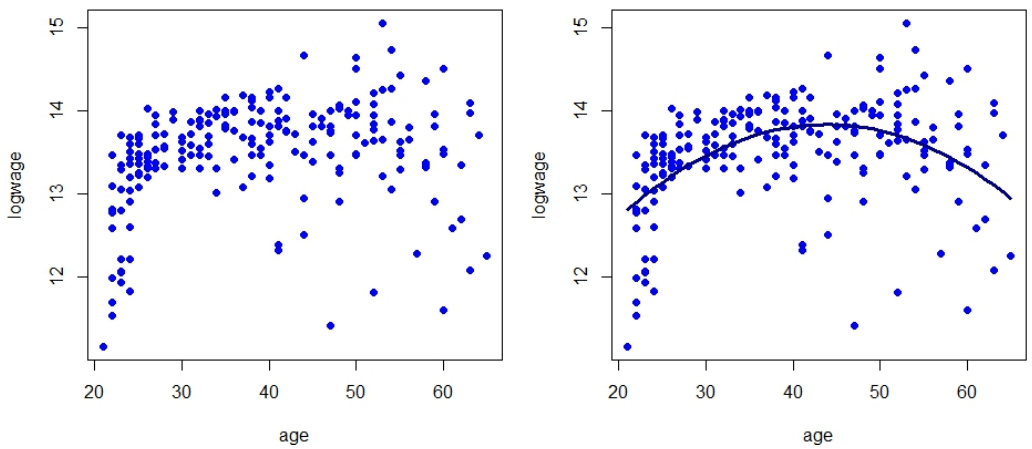
onde r é a função de regressão. A variável x também é chamada de recurso. Queremos estimar ou "aprender" a função r em hipóteses fracas. O estimador de $r(x)$ é denotado por $\hat{r}_n(x)$. Também nos referimos a $\hat{r}_n(x)$ como um suavizador. A princípio, faremos a suposição simplificadora de que a variância $\mathrm{Var}(\epsilon_i) = \sigma^2$, ou seja, a variância de ϵ_i não depende de x. Vamos relaxar essa suposição mais tarde.

Na equação acima, estamos tratando os valores da covariável x_i como fixos. Poderíamos tratá-los como aleatórios, em cujo caso escrevemos os dados como $(X_1, Y_1), \cdots, (X_n, Y_n)$ e $r(x)$ é então interpretado como a média de Y condicional em $X = x$, isto é,

$$r(x) = \mathrm{E}(Y \mid X = x).$$

Há pouca diferença nas duas abordagens e, na maioria das vezes, adotamos a abordagem "fixo x", exceto quando indicado.

Exemplo IV.1. Começamos com um conjunto de dados clássico retirado do livro de Pagan & Ullah (1999) que considera dados salariais canadenses de corte transversal consistindo de uma amostra aleatória retirada das Cópias de Uso Público do Censo Canadense de 1971 para indivíduos do sexo masculino com educação em comum (Grau 13). Existem $n = 205$ observações no total e 2 variáveis: o logaritmo do salário do indivíduo (logwage) e sua idade (age). A equação salarial tradicional é tipicamente modelada como uma idade quadrática. Os dados estão disponíveis no arquivo de dados **cps71**, no pacote de funções **np**. Este será um dos pacotes de funções **R** que utilizaremos para mostrar o ajuste dos modelos de regressão não paramétricos.



Estas figuras mostram a descrição dos dados a esquerda e o modelo de regressão paramétrico estimado a direita. Estas figuras foram obtidas com as linhas de comando mostradas a seguir.

```
> library(np)
> data("cps71")
> par(mfrow = c(1,1), mar = c(4, 4, 1, 1))
> plot(logwage ~ age, data = cps71, col = "blue", pch=19)
> model.par = lm(logwage ~ age + I(age^2), data = cps71)
> summary(model.par)

Call:
lm(formula = logwage ~ age + I(age^2), data = cps71)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-2.4041 -0.1711  0.0884  0.3182  1.3940

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 10.0419773  0.4559986  22.022  < 2e-16 ***
age          0.1731310  0.0238317   7.265 7.96e-12 ***
I(age^2)    -0.0019771  0.0002898  -6.822 1.02e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5608 on 202 degrees of freedom
Multiple R-squared:  0.2308,    Adjusted R-squared:  0.2232
F-statistic:  30.3 on 2 and 202 DF,  p-value: 3.103e-12

> par(mfrow = c(1,1), mar = c(4, 4, 1, 1))
> plot(logwage ~ age, data = cps71, col = "blue", pch=19)
> lines(cps71$age, fitted.values(model.par), col="darkblue", lwd = 3)
```

Nós temos medições ruidosas de Y_i e $r(x_i) = 10.0419 + 0.1731age - 0.0019age^2$. Nosso objetivo é estimar r de forma não paramétrica. A variância $\mathrm{Var}(\epsilon_i)$ definitivamente não é constante como uma função de x.

Os métodos que consideramos aqui são os métodos de regressão local e métodos de penalização. O primeiro inclui a regressão do kernel e a regressão polinomial local. Este último leva a métodos baseados em splines. Todos os estimadores considerados são suavizadores lineares.

Antes de mergulharmos na regressão não-paramétrica, primeiro revisamos brevemente a regressão linear ordinária e sua regressão logística relativa próxima.

IV.1 Revisão do modelo de regressão linear

Suponhamos temosos dados $(x_1, Y_1), \cdots, (x_n, Y_n)$ onde $Y_i \in \mathbb{R}$ e $x_i = (x_{i1}, \cdots, x_{ip})^\top \in \mathbb{R}^p$. O modelo de regressão linear assume que

$$Y_i \ = \ r(x_i) \ + \ \epsilon_i \ = \ \sum_{k=1}^{p} \beta_k x_{ik} \ + \ \epsilon_i, \qquad i = 1, \cdots, n,$$

onde $\mathrm{E}(\epsilon_i) = 0$ e $\mathrm{Var}(\epsilon_i) = \sigma^2$. Normalmente, queremos incluir um intercepto no modelo, então vamos adotar a convenção de que $x_{i1} = 1$.

A matriz de planejamento X é uma matriz de dimensão $n \times p$ definida por

$$X \ = \ \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}.$$

O conjunto $\mathcal{L}$ de vetores que podem ser obtidos como combinações lineares das colunas de X, é chamado de espaço de coluna de X.

Seja $Y_i = (Y_1, \cdots, Y_n)^\top$, $\epsilon = (\epsilon_1, \cdots, \epsilon_n)^\top$ e $\beta = (\beta_1, \cdots, \beta_p)^\top$. Podemos então escrever o modelo de regressão linear como

$$Y \ = \ X\beta \ + \ \epsilon.$$

Os estimadores de mínimos quadrados $\widehat{\beta} = (\widehat{\beta}_1, \cdots, \widehat{\beta}_p)^\top$ é o vetor que minimiza a soma de quadrados dos resíduos

$$(Y - X\beta)^\top (Y - X\beta) \ = \ \sum_{i=1}^{n} \left(Y_i - \sum_{k=1}^{p} x_{ik}\beta_k \right)^2.$$

Assumindo que $X^\top X$ seja inversível, o estimador de mínimos quadrados é

$$\widehat{\beta} \ = \ (X^\top X)^{-1} X^\top Y.$$

O comando **lm** avalia esta expresão e, é desta forma que obtemos $\widehat{\beta}_0 = 10.0419773$, $\widehat{\beta}_1 = 0.1731310$ e $\widehat{\beta}_2 = -0.0019771$ com os dados do exemplo acima.

O estimador de $r(x)$ no ponto $x = (x_1, \cdots, x_p)^\top$ é então

$$\hat{r}_n(x) \ = \ \sum_{k=1}^{n} \widehat{\beta}_k x_k \ = \ x^\top \widehat{\beta}.$$

Avaliando esta equação obtemos a curva estimada do logaritmo da renda anual segunda a idade do declarante, isto realizado com o auxílio do comando **fitted.values** aplicado no objeto **model.par**.

Segue-se que os valores ajustados $\hat{r}_n(x) = (\hat{r}_n(x_1), \cdots, \hat{r}_n(x_n))^\top$ podem ser escritos matricialmente como

$$\hat{r}_n(x) \ = \ X\widehat{\beta} \ = \ HY,$$

onde

$$H \ = \ X(X^\top X)^{-1} X^\top$$

é chamada de matriz hapéu. Os elementos do vetor $\hat{\epsilon} = Y - \hat{r}_n(x)$ são chamados de resíduos ordinários. A matriz chapéu é simétrica $H = H^\top$ e idempotente, $H^2 = H$. Segue-se que $\hat{r}_n(x)$ é a projeção de Y no espaço de colunas $\mathcal{L}$ de X. Pode-se mostrar que o número de parâmetros p está relacionado à matriz H pela equação

$$p \ = \ \mathrm{tr}(H)$$

onde $\mathrm{tr}(H)$ denota o traço da matriz H, isto é a soma dos seus elementos na disgonal principal. Na regressão não paramétrica, o número de parâmetros será substituído pelos graus de liberdade efetivos que serão definidos através de uma equação como $p \ = \ \mathrm{tr}(H)$.

Dado qualquer $x = (x_1, \cdots, x_p)^\top$, podemos escrever

$$\hat{r}_n(x) \ = \ l(x)^\top Y \ = \ \sum_{i=1}^{n} l_i(x) Y_i,$$

sendo que

$$l(x)^\top \ = \ x^\top (X^\top X)^{-1} X^\top.$$

Um estimador não viciado de σ^2 é

$$\widehat{\sigma}^2 \ = \ \frac{1}{n-p} \sum_{i=1}^{n} \big(Y_i - \hat{r}_n(x_i)\big)^2 \ = \ \frac{||\hat{\epsilon}||}{n-p}.$$

Em seguida, construímos uma banda de confiança para $\hat{r}_n(x)$. Queremos encontrar um par de funções $a(x)$ e $b(x)$ tais que

$$P\Big(\big\{x \in \mathbb{R}^p \ : \ a(x) \leq \hat{r}_n(x) \leq b(x), \, \forall \, x\big\}\Big) \geq 1 - \alpha.$$

Dado que $\hat{r}_n(x) \ = \ \sum_{i=1}^{n} l_i(x) Y_i$, segue que

$$\mathrm{Var}\big(\hat{r}_n(x)\big) \ = \ \sigma^2 \sum_{i=1}^{n} l_i^2(x) \ = \ \sigma^2 ||l||^2,$$

o qual sugere utilizarmos bandas de confiança da forma

$$I(x) \ = \ \big(a(x), b(x)\big) \ = \ \Big(\hat{r}_n(x) - c\widehat{\sigma}||l(x)||, \hat{r}_n(x) + c\widehat{\sigma}||l(x)||\Big),$$

para alguma constante c. O teorema a seguir pode ser encontrado em Scheffé (1959). Denotemos por $F(p, n-p)$ a variável aleatória com distribuição $F - Fisher$ com graus de liberdade p e $n-p$. Seja $F_\alpha(p, n-p)$ o α-quantil superior para essa variável aleatória, ou seja, $P\big(F(p, n-p) > F_\alpha(p, n-p)\big) = \alpha$

Teorema IV.1. Considere a amostra aleatória $(x_1, Y_1), \cdots, (x_n, Y_n)$ satisfazendo um modelo de regressaão linear. A banda de confiança

$$\Big(\hat{r}_n(x) - \sqrt{pF_\alpha(p, n-p)}\widehat{\sigma}||l(x)||, \hat{r}_n(x) + \sqrt{pF_\alpha(p, n-p)}\widehat{\sigma}||l(x)||\Big),$$

satisfaz que

$$P\Big(\big\{x \in \mathbb{R}^p \ : \ \hat{r}_n(x) - \sqrt{pF_\alpha(p, n-p)}\widehat{\sigma}||l(x)|| \leq \hat{r}_n(x) \leq \hat{r}_n(x) + \sqrt{pF_\alpha(p, n-p)}\widehat{\sigma}||l(x)||, \, \forall \, x\big\}\Big) \geq 1 - \alpha.$$

Demonstração. Ver Scheffé (1959).

IV.1.1 Regressão associada ao coeficiente de correlação τ_K de Kendall

Antes de passar para o procedimento anunciado aqui, olhamos brevemente para o modelo clássico de regressão de mínimos quadrados. Muitos leitores já estarão familiarizados com este modelo, mas enfatizamos aqui os aspectos que ajudam a entender a lógica por trás de muitas abordagens livres de distribuição. Na abordagem paramétrica clássica, assume-se que para cada um dos n conjuntos de x_i observados, que podem ser variáveis aleatórias ou um conjunto de valores fixos que podem ou não ser escolhidos antecipadamente, observamos algum valor y_i de uma variável aleatória Y_i, que tem as propriedades que sua média depende, ou seja, é condicionada ao valor de x_i de tal forma que

$$\mathrm{E}(Y_i|x_i) \,=\, \beta_0 + \beta_1 x_i,$$

enquanto a variância de Y_i é independente de x e para todos os x_i tem o valor

$$\mathrm{Var}(Y_i) \,=\, \sigma^2$$

onde β_0, β_1 e σ^2 são desconhecidos. A relação linear entre $\mathrm{E}(Y|x)$ e x da forma $\mathrm{E}(Y|x) = \beta_0 + \beta_1 x$ entre a média condicional de Y e x dado define a regressão de Y em x. A linha tem inclinação β_1 e intercepto β_0 no eixo y. A notação $Y_i|x_i$ é a notação convencional para um evento ou variável Y_i tendo alguma propriedade condicionada a um x_i especificado.

Para muitos fins de inferência, assume-se também que a distribuição condicional de $Y_i|x_i$ é $N(\beta_0 + \beta_1 x_i, \sigma^2)$. Um problema clássico de regressão é estimar β_0, β_1 e, às vezes, também σ^2 dado um conjunto de n observações emparelhadas $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$ onde para um dado x_i o y_i é um valor observado da variável aleatória Y_i. Estas condições são válidas se cada um (x_i, y_i) satisfizer

$$y_i \,=\, \beta_0 + \beta_1 x_i + \epsilon_i,$$

onde cada ϵ_i é um valor não observado de uma variável aleatória $N(0, \sigma^2)$ e os ϵ_i são independentes uns dos outros e também de x_i. A estimativa de mínimos quadrados busca valores $\widehat{\beta}_0$ e $\widehat{\beta}_1$ que minimizem

$$S = \sum_{i=1}^{n} \big(y_i - \beta_0 - \beta_i x_i\big)^2.$$

A linha reta $y = \widehat{\beta}_0 + \widehat{\beta}_1 x$ é chamada de regressão de mínimos quadrados de y em x. Se, para qualquer linha $y = \beta_0 + \beta_1 x$, denotamos a coordenada y correspondente a $x = x_i$ por $\hat{y}_i$, ou seja, $\hat{y}_i = \widehat{\beta}_0 + \widehat{\beta}_1 x_i$ então as diferenças entre os valores observados e previstos, isto é, $\hat{\epsilon}_i = y_i - \hat{y}_i, i = 1, 2, \cdots, n$ são chamados os resíduos absolutos com relação a essa linha.

Com os pressupostos acima da regressão clássica por mínimos quadrados é feita uma generalização de testes clássicos para comparação de médias de tratamento. Em particular, um teste de $H_0 : \beta_1 = 0$ equivale a um teste de igualdade de um conjunto de médias. Isso decorre de considerarmos os x_i como indicadores ou rótulos anexados às amostras. Se houver n observações emparelhadas (x_i, y_i), cada y_i correspondente a um x_i particular pode ser encarado como um valor observado a partir da amostra marcada por esse x_i. O número total de amostras m pode ser qualquer número entre 2 e n e o número de observações n_j, na amostra j, $j = 1, 2, \cdots, m$ estão sujeitos à restrição $n_1 + n_2 + \cdots + n_m = n$. Em particular, se não houver dois x_i iguais, existem n amostras, cada uma de uma observação e, no outro extremo, se houver apenas duas x_i distintas, há duas amostras com número observações n_1 e $n_2 = n - n_1$ respectivamente. Suponha que neste último marcamos a primeira amostra por $x = 0$ e a segunda categoria por $x = 1$ e os primeiros valores de amostra são

$$y_{11}, y_{12}, \cdots, y_{1n_1}$$

com média μ_0 e os valores da segunda amostra sejam

$$y_{21}, y_{22}, \cdots, y_{2n_2}$$

com média μ_1. Pode então ser mostrado que $\widehat{\beta}_1 = \mu_1 - \mu_0$, a diferença de média amostral utilizada no teste t para a igualdade das médias dos tratamentos. Assim, o teste de igualdade de médias é neste caso idêntico ao teste $H_0 : \beta_1 = 0$. No caso mais geral de m amostras, um teste para $\beta_1 = 0$ é um teste para a identidade de todas as m médias populacionais.

É bem conhecido que para o modelo clássico de mínimos quadrados o estimador de β_1 não depende do estimador de β_0, mas o de β_0 depende daquele de β_1 através de $\widehat{\beta}_1$, já que $\widehat{\beta}_0 = \overline{y} - \widehat{\beta}_1 \overline{x}$. A partir dessa expressão para $\widehat{\beta}_0$ é facilmente visto que a equação ajustada pode ser escrita sem referência ao intercepto na forma

$$y \,=\, \overline{y} - \widehat{\beta}_1 (x - \overline{x}),$$

implicando que a linha passa pelo ponto $(\overline{x}, \overline{y})$.

Uma interpretação gráfica disso é que a inclinação da linha ajustada é inalterada por uma mudança de origem. Em particular, se mudarmos a origem para a média bivariada $(\overline{x}, \overline{y})$ e escrevermos $\tilde{x} = x - \overline{x}$, $\tilde{y} = y - \overline{y}$ a equação acima se torna

$$\tilde{y} \,=\, \widehat{\beta}_1 \tilde{x}$$

e, então

$$\widehat{\beta}_1 \,=\, \frac{\sum\limits_{1=1}^{n} \tilde{x}_i \tilde{y}_i}{\sum\limits_{i=1}^{n} \tilde{x}_i^2}.$$

A expresão acima ainda é válida se apenas mudarmos a origem para a média dos x_i e usarmos o y_i original. Foi apontado que o coeficiente de correlação de Pearson permanece inalterada por transformações lineares de (x, y) da forma $x' = (x - k)/s$ e $y' = (y - m)/t$ onde s, t ambos são positivos. Acabamos de ver que, em regressão, embora a estimativa de β_0 seja afetada por uma mudança na origem, a de β_1 não é, isto é, transformações da forma $x' = (x - k)$, $y' = (y - m)$ não afetam a estimativa de β_1. Entretanto, é fácil estabelecer que a transformação $x' = (x - k)/s$ e $y' = (y - m)/t$ altera tanto o valor verdadeiro de β_1 quanto sua estimativa de mínimos quadrados pelo mesmo fator de escala s/t, alterando β_1 para $\beta_1' = s\beta_1/t$.

No entanto, faremos uso de outra propriedade do coeficiente de correlação de Pearson em testes permutacionais, ou seja, porque todas as outras quantidades na expressão da estimativa do coeficiente de correlação de Pearson permanecem constantes sob permutações, podemos basear testes de permutação na estatística $T = \sum_i x_i y_i$ no caso do coeficiente de Pearson como alternativa ao uso do coeficiente de correlação amostral r. Uma propriedade correspondente é transferida para o coeficiente de Spearman com postos substituindo os x_i, y_i.

Descartando a suposição de normalidade para o Y_i fazer inferências sobre β_0 e β_1 são difíceis, então consideramos primeiro a estimação apenas de β_1. O problema reduz-se então a fazer inferências sobre diferenças em medianas ou médias amostrais marcadas pelos diferentes x_i. Para qualquer Y_i, Y_j independentes, associados com x_i, x_j distintos, descartamos as suposições de normalidade nas distribuições condicionais de $Y|x$ e assumimos agora apenas que para qualquer x_i, x_j elas têm distribuições $F_i(Y_i|x_i)$, $F_j(Y_j|x_j)$ que diferem apenas na sua medida de centralidade que será tomada em geral como a mediana, mas que coincidirá com a média das distribuições condicionais que são assumidas simétricas desde que a média exista. Para o modelo de regressão linear, a mediana de $F_i(Y_i|x_i)$ é $\text{Mediana}(Y_i|x_i) = \beta_0 + \beta_1 x_i$ e aquela de $F_j(Y_j|x_j)$ é $\text{Mediana}(Y_j|x_j) = \beta_0 + \beta_1 x_j$, sendo estes os análogos da estimativa do coefficiente de correlaçã e a diferença entre as medianas é claramente

$$\text{Mediana}(Y_j - Y_i|x_i, x_j) = \beta_1(x_j - x_i)\cdot$$

Assumindo que as diferenças entre as distribuições estão confinadas a uma diferença mediana implica que, para todo i

$$D_i(\beta_1) = Y_i - \beta_1 x_i,$$

são distribuídas de forma idêntica e independente com mediana β_0. Como os D_i são, portanto, independentes do x_i, segue-se que eles não são correlacionados com o x_i. Como apontamos, o x_i pode ser alterado pela adição ou subtração de uma constante sem afetar a inclinação ou sua estimativa. Em particular, ele simplificará a álgebra se ajustarmos o x_i adicionando uma constante apropriada para fazer $\sum_i x_i = 0$, o que, em termos gráficos, implica mudar a origem para um ponto no eixo x correspondente à média $\overline{x}$. Escrevendo $\widehat{D}_i = yi - \beta x_i$ um teste intuitivamente razoável da hipótese $H_0 : \beta_1 = \beta$ contra uma alternativa de um ou dois lados é um teste para a correlação zero entre o x_i e o $\widehat{D}_i$, pois sabemos que se H_0 é válido então $D_i(\beta)$ não é correlacionado com o x_i e os $\widehat{D}_i$ são valores observados da variável $D_i(\beta)$. Se o teste mais apropriado é baseado em Pearson, Spearman, Kendall ou algum outro coeficiente dependerá de quais suposições sejam feitas sobre as características da distribuição $F(Y|x)$, por exemplo, cauda longa, simétrica ou assimétrica, etc.. Nós chamamos os $\widehat{D}_i$ de resíduos por conveniência, mas este é um uso não ortodoxo do termo que mais convencionalmente se refere ao $\hat{\epsilon}_i = y_i - \alpha - \beta x_i$ onde α é algum valor hipotético de β_0. Uma estatística apropriada baseada no coeficiente de Pearson para o teste de correlação zero é

$$T(\beta) \;=\; \sum_{i=1}^{n} x_i \widehat{D}_i \;=\; \sum_{i=1}^{n} x_i(y_i - \beta x_i)\cdot$$

Devemos lembrar que se tivermos medições em duas variáveis para uma amostra de n indivíduos, estas observações emparelhadas podem ser escritas como $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$. O coeficiente de correlação amostral de produto de momentos de Pearson r, é definido como

$$r \;=\; \frac{\sum_i \Big((x_i - \overline{x})(y_i - \overline{y})\Big)}{\sqrt{\sum_i (x_i - \overline{x})^2 \sum_i (y_i - \overline{y})^2}},$$

que, para fins computacionais, é geralmente reordenada e escrita como

$$r \;=\; \frac{C_{xy}}{\sqrt{C_{xx} C_{yy}}},$$

sendo $C_{xy} = \sum_i x_i y_i - (\sum_i x_i)(\sum_i y_i)/n$, $C_{xx} = \sum_i x_i^2 - (\sum_i x_i)^2/n$ e $C_{yy} = \sum_i y_i^2 - (\sum_i y_i)^2/n$.

Voltando ao nosso desenvolvimento. Observemos que, para estimar β, um procedimento de estimação intuitivamente razoável é resolver para β a equação

$$T(\beta) \;-\; \sum_{i=1}^{n} x_i \sum_{i=1}^{n} \widehat{D}_i/n \;=\; 0\cdot$$

Isso segue da forma de C_{xy}, já que equalizar C_{xy} a zero garante um valor zero para o coeficiente de correlação amostral, uma vez que torna o numerador do coeficiente de correlação amostral de Pearson zero. Lembrando que podemos ajustar o x_i para ter a média zero, ou seja, de modo que $\sum_i x_i = 0$ sem afetar nossa estimativa de β, assumimos que isso foi feito para que a equação acima simplifique para

$$T(\beta) \;=\; \sum_{i=1}^{n} x_i(y_i - \beta x_i) \;=\; 0,$$

com solução $\widehat{\beta} = (\sum_i x_i y_i)/(\sum_i x_i^2)$.

Não fizemos suposições sobre a distribuição do D_i exceto que eles são idênticos para todos os i, de modo que, para o teste de hipótese, um teste de permutação para o coeficiente de correlação de Pearson ser zero com base nos valores amostrais $(x_i, \widehat{D}_i)$ é apropriado. Como indicamos em outros casos, uma dificuldade com inferências baseadas em dados brutos usando um teste de permutação é que os resultados tendem a ser similares àqueles baseados na teoria equivalente, assumindo normalidade mesmo quando a suposição de normalidade é claramente violada. Em outras palavras, o método não tem robustez. Melhores procedimentos para lidar com dados são geralmente fornecidos por procedimentos de estimação baseados em classificação.

Vejamos como fazer inferências baseadas no coeficiente τ_K de Kendall, sabemos que τ_K depende apenas da ordem das observações e não da magnitude das diferenças entre os valores dos dados. Além disso, não há vantagem em ajustar o x_i para terem média zero. Simplifica a apresentação sem perda de generalidade se, quando os x_i são todos diferentes, assumimos que $x_1 < x_2 \cdots < x_n$.

A estatística usada para inferência sobre β é

$$T_t(\beta) = \sum_i \text{sgn}\big(\widehat{D}_{ij}(\beta)\big),$$

onde

$$\begin{aligned} \widehat{D}_{ij}(\beta) &= (y_j - \widehat{\beta}_0 - \widehat{\beta} x_j) - (y_i - \widehat{\beta}_0 - \widehat{\beta} x_i) = (y_j - \widehat{\beta} x_j) - (y_i - \widehat{\beta} x_i) \\ &= \widehat{D}_j - \widehat{D}_i = (y_j - y_i) - \widehat{\beta}(x_j - x_i) \end{aligned}$$

e e soma dos sinais dos $\widehat{D}_{ij}$ é sobre todo $i = 1, 2, \cdots, n-1$ e $j > i$. Como os x_i são todos diferentes e em ordem ascendente é claro que $T_t(\beta)$ é o numerador na expressão da estatística de teste para verificar nulidade do coeficiente τ_K composto dos números de concordâncias menos o número de discordâncias nos pares de dados $x_i, \widehat{D}_i)$.

O uso de uma estatística equivalente a $T_t(\beta)$ foi proposto pela primeira vez por Theil (1950) e o procedimento é amplamente conhecido como o método de Theil. Sen (1968) destacou a relação com o coeficiente τ_K de Kendall, então nos referimos a ele como o método **Theil-Kendall** ou estimador **Theil-Sen**.

Claramente $T_t(\beta)$ é uma função linear do estimador de amostral de τ_K e um estimador de pontual apropriado de β é obtido ajustando-se $T_t(\beta) = 0$. Como os x_i estão em ordem ascendente é fácil ver que $T_t(\beta)$ não é alterada se substituirmos $\widehat{D}_{ij}(\beta)$ por $\widehat{B}_{ij} - \beta$ onde, $\widehat{B}_{ij} = (y_j - y_i)/(x_j - x_i)$. Claramente, então, $T_t(\beta) = 0$, se escolhermos $\beta = mediana(\widehat{B}_{ij})$, então o número de positivos e o número de $\widehat{D}_{ij}$ negativos serão iguais.

Como foi o caso de $T_s(\beta)$, percebemos que apenas $T_t(\beta)$ muda de valor quando β passa por um valor de $\widehat{B}_{ij}$ e como a estatística é o numerador do coeficiente τ_K de Kendall, segue-se que se todos os $\widehat{B}_{ij}$ são distintos quando β aumenta de $-\infty$ para ∞ $T_t(\beta)$ é uma função degrau decrescente por passos de 2 desde ½$n(n-1)$ para -½$n(n-1)$. Claramente a divisão de $T_t(\beta)$ por ½$n(n-1)$ leva ao t_K de Kendall, que pode ser usado como uma estatística equivalente. Se os $\widehat{D}_{ij}$ não forem todos distintos, alguns dos passos serão múltiplos de 2. Em particular, se um $\widehat{D}_{ij}$ ocorre r vezes, ele induz um degrau de altura $2r$. O valor de $T_t(\beta)$ em qualquer $\widehat{D}_{ij}$ pode ser considerado como a m&eacudia;dia dos valores imediatamente acima e abaixo daquele $\widehat{D}_{ij}$.

Testes de hipóteses sobre β são diretos usando ou o t_K de Kendall ou o equivalente $T_t(\beta)$ como a estatística de teste e se sabemos a distribuição exata do coeficiente de Kendall amostral quando $\tau_K = 0$, um intervalo de confiança para β pode ser obtido. Os procedimentos estão ilustrados no Exemplo a seguir.

Exemplo IV.2. O fluxo de água em metros cúbicos por segundo y, em um ponto fixo de um córrego da montanha, é registrado em intervalos de uma hora x após um degelo de neve começando no tempo $x = 0$.

Horas a partir do início do degelo x	0	1	2	3	4	5	6
Fluxo em metros cúbicos/seg y	2.5	3.1	3.4	4.0	4.6	5.1	11.1

Durante os degelos anteriores, tem havido frequentemente uma relação quase linear entre o tempo e o fluxo. Use o método de Theil-Kendall para:

(i) testar a hipótese $H_0 : \beta_1 = 1$ contra a alternativa $H_1 : \beta_1 \neq 1$;

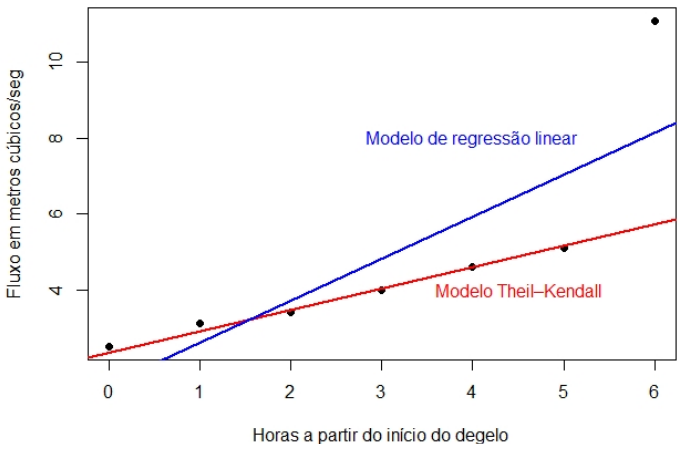
(ii) obter uma estimativa pontual de β_1 e um intervalo de confiança que dê pelo menos 95 por cento de cobertura.

Primeiro vamos ajustar o modelo de regressão de Theil-Kendall ou Theil-Sen, para isso utilizamos a função **mblm** no pacote homônimo.

```
> library(mblm)
> horas = c(0,1,2,3,4,5,6)
> fluxo = c(2.5,3.1,3.4,4,4.6,5.1,11.1)
> ajuste0 = mblm(fluxo ~ horas, repeated = FALSE)
> ajuste1 = lm(fluxo ~ horas)
> summary(ajuste0)
```

```
Call:
mblm(formula = fluxo ~ horas, repeated = FALSE)

Residuals:
       1        2        3        4        5        6        7
 0.16667  0.20000 -0.06667 -0.03333  0.00000 -0.06667  5.36667

Coefficients:
            Estimate     MAD V value Pr(>|V|)
(Intercept)  2.33333 0.09884      28   0.0156 *
horas        0.56667 0.09884     231 6.39e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.403 on 5 degrees of freedom

> par(mfrow = c(1,1), mar = c(4, 5, 2, 1), pch = 19)
> plot(fluxo ~ horas, pch = 19, xlab = "Horas a partir do início do degelo",
      ylab = "Fluxo em metros cúbicos/seg")
> abline(ajuste0, lty = 1, lwd = 2, col = "red")
> text(4.5,4, "Modelo Theil–Kendall", col = "red")
> abline(ajuste1, lty = 1, lwd = 2, col = "blue")
> text(4,8, "Modelo de regressão linear", col = "blue")
```

No gráfico a seguir mostramos o comportamento do modelo de regressão linear assim como do modelo não paramémetrico de Theil-Kendall ou Theil-Sen.



A regressão de Kendall-Theil é uma abordagem completamente não-paramétrica à regressão linear, onde existe uma variável independente e uma variável dependente. É robusto para outliers na variável dependente, como podemos perceber. Ele simplesmente calcula todas as linhas entre cada par de pontos e usa a mediana das inclinações dessas linhas. O método produz um declive e um intercepto para a linha de ajuste, assim como um p-valor para a inclinação também pode ser determinado. Normalmente, nenhuma medida análoga ao R^2 é relatada. Na Seção IV.2.1 Avaliando a qualidade do ajuste, estudaremos uma forma alternativa de escrever R^2 que poderemos utilizar nesta situação.

Para testar a hipótese em (i), observemos que inspeção mostra que a estimativa pontual de β_1, ou seja, $\text{mediana}(\widehat{B}_{ij})$ é 0.567, porque há dez maiores e dez menores $\widehat{B}_{ij}$.

```
> ajuste0$slopes
 [1] 0.6000000 0.4500000 0.5000000 0.5250000 0.5200000 1.4333333 0.3000000 0.4500000
 [9] 0.5000000 0.5000000 1.6000000 0.6000000 0.6000000 0.5666667 1.9250000 0.6000000
[17] 0.5500000 2.3666667 0.5000000 3.2500000 6.0000000
> median(ajuste0$slopes)
[1] 0.5666667
```

Para testar a hipótese $\beta_1 = 1$ subtraímos $\beta_1 = 1$ de cada $\widehat{B}_{ij}$

```
> ajuste0$slopes-1
 [1] -0.4000000 -0.5500000 -0.5000000 -0.4750000 -0.4800000  0.4333333 -0.7000000 -0.5500000
 [9] -0.5000000 -0.5000000  0.6000000 -0.4000000 -0.4000000 -0.4333333  0.9250000 -0.4000000
[17] -0.4500000  1.3666667 -0.5000000  2.2500000  5.0000000
```

e encontramos que isto implica 15 discordâncias ou valores negativos e 6 concordâncias, onde

$$T_t(1) = 6 - 15 = -9,$$

e quando $n = 7$, isto implica que, $t_K = -9/21 = 0.4286$.

```
> library(SuppDists)
> 2*pKendall(abs(-9/21), N = 7, lower.tail = FALSE)
[1] 0.1361111
```

O pacote **SuppDists** incorpora ao R diversas distribuições suplementares, entre elas, a distribuição exata do coefficiente de correlação τ_K de Kendall. Utilizamos as linhas de comando acima para avaliarmos evidências a favor ou contra H_0. Correspondendo a este valor de t_K com $N = 7$, os sete pares de dados, o $p - valor = 0.1361111$ é exatamente bilateral de modo que não há evidência convincente contra H_0.

Para obter um intervalo de confiança aproximado de 95 por cento para β_1, devemos primeiro determinar um valor de $|T_t|$ com cauda o mais próxima possível, mas não superior a 0.025.

```
> qKendall(0.025, N = 7)*21
[1] -13
> qKendall(0.975, N = 7)*21
[1] 13
```

Para obter um intervalo de confiança aproximado de 95 por cento para β_1, devemos primeiro determinar um valor de $|T_t|$ com uma cauda o mais próximo possível, mas não superior a 0.025. A distribuição exata quando $n = 7$ dado pelo R indica que para probabilidade na cauda de 0.025, $T_t = 13$. Obtemos o intervalo

```
> n = 7
> denom = 0.5*n*(n-1)
> Tt = 13
> Tt/denom
[1] 0.6190476
> 1 - 0.7142857 + qnorm(0.975)*(13/21)/n
[1] 0.4590444
> 1 + Tt/denom + qnorm(0.975)*(13/21)/n
[1] 1.792378
```

ou (0.4590444,1.792378).

A estreita relação entre o τ_K de Kendall e o T_t facilita o transporte de aproximações assintóticas entre eles. Usando os resultados para τ_K pode-se mostrar (Maritz, 1995) que

$$\mathrm{E}(T_t) = 0 \qquad \text{e que} \qquad \mathrm{Var}(T_t) = \frac{n(n-1)(2n+5)}{18}.$$

Para grandes n a distribuição de $Z = T_t / \sqrt{\mathrm{Var}(T_t)}$ é aproximadamente normal e inferências assintóticas podem então ser feitas da maneira usual. Mesmo para $n = 7$, a aproximação às vezes não é seriamente enganosa, mas recomenda-se cautela com uma amostra tão pequena.

A regressão de Kendall-Theil também chamada de regressão de Theil-Sen é uma robusta substituição não-paramétrica da abordagem tradicional de mínimos quadrados ao modelo de regressão de linha reta $Y = \beta_0 + \beta_1 x + \epsilon$ e também a alguns modelos de regressão linear mais complexos. Esta metodologia não exige normalidade dos erros aleatórios, sendo capaz de fornecer estimativas dos parâmetros, testes de hipóteses lineares sobre os parâmetros, bem como intervalos de confiança para os parâmetros. Ver, por exemplo, Hollander, M. et. all. (2014) para uma descrição mais detalhada do método.

IV.2 Suavizamento linear

Como observamos anteriormente, todos os estimadores não-paramétricos aqui considerados são suavizadores lineares. A definição formal é a seguinte.

Definição IV.1. Um estimador $\hat{r}_n$ de r é um suavizador linear se, para cada x, existe um vetor $l(x) = (l_1(x), \cdots, l_n(x))^\top$ tal que

$$\hat{r}_n(x) = \sum_{i=1}^{n} l_i(x) Y_i.$$

Definindo o vetor de valores ajustados

$$\hat{r}_n(x) = \big(\hat{r}_n(x_1), \cdots, \hat{r}_n(x_n)\big)^\top,$$

segue-se então que

$$\hat{r}_n(x) = LY,$$

onde L é uma matriz $n \times n$ cuja i-ésima linha é $l_i(x)^\top$, assim, $L_{ij} = l_j(x_i)$. As entradas da i-ésima linha mostram os pesos dados a cada Y_i na formação da estimativa $\hat{r}_n(x_i)$.

Definição IV.2. A matriz L é chamada matriz de suavização. A i-ésima linha de L é chamada de kernel efetivo para estimar $r(x_i)$. Definimos os graus de liberdade efetivos como

$$\nu = \operatorname{tr}(L).$$

O leitor não deve confundir suavizadores lineares com regressão linear, em que se assume que a função de regressão $r(x)$ é linear. Deemos observar que os pesos em todos os suavizadores que usaremos terão a propriedade que, para todo x, $\sum_{i=1}^{n} l_i(x) = 1$. Isto implica que o suavizador preserva curvas constantes, portanto, se $Y_i = c$ para todo i, então $\hat{r}_n(x) = c$.

Exemplo IV.3. (Regressograma) Suponhamos que $a \leq x_i \leq b$, para $i = 1, 2, \cdots, n$. Vamos dividir o intervalo (a, b) em m caixas igualmente espaçadas denotadas por $B_1, B_2, \cdots, B_m$. Definamos $\hat{r}_n(x)$ por

$$\hat{r}_n(x) = \frac{1}{n_j} \sum_{\{\, i\,:\, x \in B_i \,\}} Y_i, \qquad \forall x \in B_j,$$

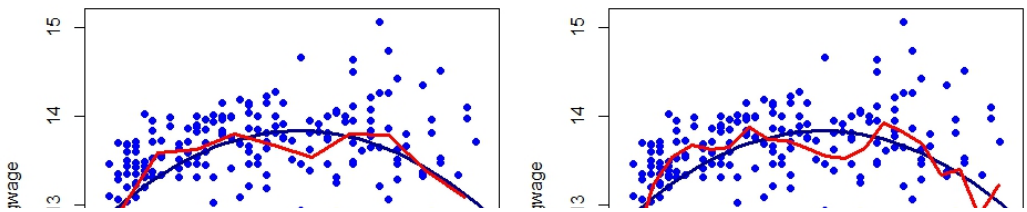
sendo que n_j é o número de pontos em B_j. Em outras palavras, a estimativa de $\hat{r}_n$ é uma função de passo obtida pela média do Y_i sobre cada caixa. Essa estimativa é chamada de **regressograma**. Um exemplo é apresentado nas Figuras abaixo. Para $x \in B_j$ definamos $l_i(x) = 1/n_j$ se $x_i \in B_j$ e $l_i(x) = 0$ caso contrário. Assim, $\hat{r}_n(x) = \sum_{i=1}^{n} Y_i l_i(x)$. O vetor de pesos $l(x)$ se parece com isso:

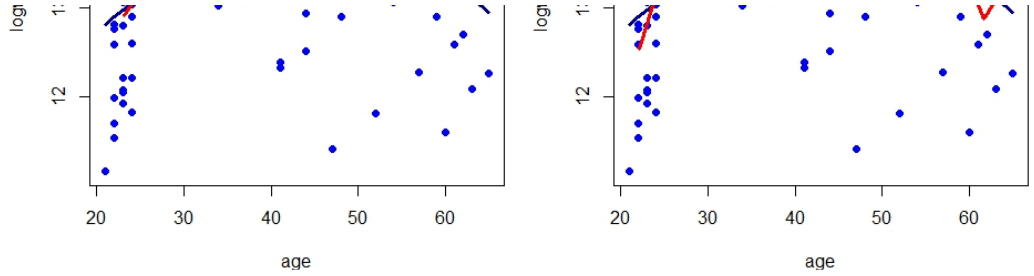
$$l(x)^\top = \Big(0, 0, \cdots, 0, \frac{1}{n_j}, \cdots, \frac{1}{n_j}, 0, \cdots, 0\Big).$$

Para ver como a matriz de suavização L se parece, suponha que $n = 9$, $m = 3$ e $n_1 = n_2 = n_3 = 3$. Então

$$L = \frac{1}{3} \times \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

Em geral, é fácil ver que existem $\nu = \operatorname{tr}(L) = m$ graus de liberdade efetivos. A largura de banda $h = (b - a)/m$ controla a suavidade da estimativa e a matriz de alisamento L tem a forma.





Vamos utilizar os dados do Exemplo IV.1 com os resultados do modelo de regressão linear. Para construir as figuras acima fizemos alterações ao comando **regressogram** disponível no pacote de funções **HoRM**. Como padrão a função **regressogram** devolve o gráfico de dispersão e a curva de regressão não paramétrica estimada pelo suavizamento linear. Como mostramos nas linhas de comando a seguir.

```
> library(HoRM)
> regressogram(cps71$age, cps71$logwage, x.lab = "age", y.lab = "logwage",
               nbins = 10, main = "Regressograma", show.bins = FALSE,
               show.means = FALSE, show.lines = TRUE)
> regressogram(cps71$age, cps71$logwage, x.lab = "age", y.lab = "logwage",
               nbins = 20, main = "Regressograma")
```

Com estes comandos obtemos duas figuras mostrando o ajuste de regressograma com $m = 10$ e $m = 20$ caixas, respectivamente. Isto foi obtido modificando o valor da opção $nbins$. Um problema é que gostariamos de apresentar conjuntamente com o regressograma o ajuste do modelo de regressão paramétrico. Como isso não é possível modificamos apropriadamente a função segundo nosso interesse, como mostramos a seguir.

```
> regressograma = function (x, y, nbins = 10) {
    xy <- data.frame(x = x, y = y)
    xy <- xy[order(xy$x), ]
    z <- cut(xy$x, breaks = seq(min(xy$x), max(xy$x), length = nbins +
                                1), labels = 1:nbins, include.lowest = TRUE)
    xyz <- data.frame(xy, z = z)
    MEANS <- c(by(xyz$y, xyz$z, FUN = mean))
    x.seq <- seq(min(x), max(x), length = nbins + 1)
    midpts <- (x.seq[-1] + x.seq[-(nbins + 1)])/2
    d2 <- data.frame(midpts = midpts, MEANS = MEANS)
    return(d2)
}
> model.par.1 = regressograma(cps71$age, cps71$logwage, nbins = 10)
> model.par.2 = regressograma(cps71$age, cps71$logwage, nbins = 20)
> par(mfrow = c(1,1), mar = c(4, 4, 1, 1))
> plot(logwage ~ age, data = cps71, col = "blue", pch=19)
> lines(cps71$age, fitted.values(model.par), col="darkblue", lwd = 3)
> lines(model.par.1$midpts, model.par.1$MEANS, col = "red", lwd = 3)
> par(mfrow = c(1,1), mar = c(4, 4, 1, 1))
> plot(logwage ~ age, data = cps71, col = "blue", pch=19)
> lines(cps71$age, fitted.values(model.par), col="darkblue", lwd = 3)
> lines(model.par.2$midpts, model.par.2$MEANS, col = "red", lwd = 3)
```

Podemos perceber com este exemplo que o regressograma aproxima-se do modelo paramétrico, embora o regressograma sirva mais como descrição dos dados e não como interpretação. O regressograma não é a única forma de encontrarmos um suavizamento linear. Vejamos no seguinte exemplo uma outra forma de definirmos um destes modelos.

Exemplo IV.4. (Médias locais) Fixemos $h > 0$ e definamos $B_x = \{i : |x_i - x| \leq h\}$. Seja n_x o número de pontos em B_x. Para qualquer x para o qual $n_x > 0$ definamos

$$\hat{r}_n(x) = \frac{1}{n_x} \sum_{i \in B_x} Y_i.$$

Este é o estimador médio local de $r(x)$, um caso especial do estimador kernel a ser discutido em breve. Nesse caso, $\hat{r}_n(x) = \sum_{i=1}^n Y_i l_i(x)$ onde $l_i(x) = 1/n_x$ se $|x_i - x| \leq h$ e $l_i(x) = 0$ caso

contrário. Como exemplo simples, suponha que $n = 9$, $x_i = i/9$ e $h = 1/9$. Então,

$$L = \begin{pmatrix} 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/3 & 1/3 & 1/3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \end{pmatrix}.$$

Maiores detalhes ver o lvro de Fan & Gijbels (1996).

IV.2.1 Avaliando a qualidade do ajuste

É desejável usar uma medida, livre de unidade, para mensurar a adequação de modelos de regressão não paramétricos que seja comparável àquela usada para modelos de regressão paramétrica, a saber, R^2. Note que esta será uma medida dentro da amostra da qualidade de ajuste. Dadas as desvantagens conhecidas de computar R^2 com base na decomposição da soma dos quadrados, há uma definição e um método alternativos para calcular R^2, que pode ser usado diretamente para qualquer modelo, linear ou não linear.

Definição IV.3. Seja Y_i a resposta e $\hat{r}_n(x_i)$ a resposta estimada para a i-ésima observação. Definimos R^2 como segue:

$$R^2 = \frac{\Big(\sum_{i=1}^n (y_i - \overline{y})(\hat{r}_n(x_i) - \overline{y})\Big)^2}{\sum_{i=1}^n (y_i - \overline{y})^2 \sum_{i=1}^n (\hat{r}_n(x_i) - \overline{y})^2},$$

onde $y_1, \cdots, y_n$ são as observações da resposta e $\overline{y}$ a estimativa da média.

Esta medida estará sempre no intervalo $[0, 1]$ com o valor 1 denotando um ajuste perfeito aos dados da amostra e 0 denotando nenhum poder preditivo acima daquele dado pela média incondicional da resposta. Pode ser demonstrado que este método de cálculo do R^2 é idêntico à medida padrão computada como $\sum_{i=1}^n (\widehat{\mu}_i - \overline{y})^2 / \sum_{i=1}^n (y_i - \overline{y})^2$ quando o modelo é linear, obtido por mínimos quadrados e incluindo o termo do intercepto. Nesta expressão $\widehat{\mu} = X\widehat{\beta}$ é o vetor de estimativas do preditor linear.

Esta medida útil permitirá a comparação direta entre a adequação da amostra e a qualificação óbvia de que este não é, de forma alguma, um critério de seleção de modelos e sim simplesmente uma medida resumida que podemos relatar.

Devemos observar que em situações nas quais agrupamos informações, como é o caso do regressograma, não é possível utilizarmos este coeficiente.

IV.2.2 Escolhendo o parâmetro de suavização

Os suavizadores que usaremos dependerão de algum parâmetro de suavização e precisaremos de alguma maneira de escolher h. Definamos o risco como o erro quadrático médio

$$R(h) = \mathrm{E}\Big(\frac{1}{n}\sum_{i=1}^n \big(\hat{r}_n(x_i) - r(x_i)\big)^2\Big).$$

Idealmente, gostaríamos de escolher h como aquele que minimizar $R(h)$, mas $R(h)$ depende da funç:ão desconhecida $r(x)$. Em vez disso, minimizaremos uma estimativa $\widehat{R}(h)$ de $R(h)$. Como primeiro palpite, podemos usar a média da soma de quadrados dos resíduos, também chamadas de erro de treinamento

$$\frac{1}{n}\sum_{i=1}^n \big(Y_i - \hat{r}_n(x_i)\big)^2$$

para estimar $R(h)$. Isto acaba por ser uma estimativa pobre de $R(h)$: é tendenciosa para baixo e tipicamente leva a subsuavizamento (sobreajuste). A razão é que estamos usando os dados duas vezes: para estimar a função e estimar o risco. A estimativa de função é escolhida para fazer $\sum_{i=1}^n \big(Y_i - \hat{r}_n(x_i)\big)^2$ pequeno, o que tenderá a subestimar o risco.

Estimaremos o risco usando validação cruzada, que é definida da seguinte forma.

Definição IV.4. O estimador de validação cruzada do risco é definido por

$$\widehat{R}(h) \;=\; \frac{1}{n}\sum_{i=1}^{n}\big(Y_i - \hat{r}_{-i}(x_i)\big)^2,$$

onde $\hat{r}_{-i}$ é o estimador obtido omitindo o i-ésimo par (x_i, Y_i).

Como dito acima, esta definição está incompleta. Nós não dissemos o que queremos dizer precisamente por $\hat{r}_{-i}$. Vamos definir

$$\hat{r}_{-i}(x) \;=\; \sum_{j=1}^{n} Y_j l_{j,(-i)}(x),$$

onde

$$l_{j,(-i)}(x) \;=\; \begin{cases} 0, & \text{se } j = i \\ \dfrac{l_j(x)}{\sum_{k\neq i} l_k(x)}, & \text{se } j \neq i \end{cases}.$$

Em outras palavras, definimos o peso em x_i para 0 e renormalizamos os outros pesos para somar um. Para todos os métodos a serem considerados: regressão kernel, polinômios locais e suavização por splines, essa forma para $\hat{r}_{-i}$ pode realmente ser derivada como uma propriedade do método, em vez de uma questão de definição. Mas é mais simples tratar isso como uma definição.

A intuição da validação cruzada é a seguinte. Observe que

$$\begin{aligned} \mathrm{E}\big(Y_i - \hat{r}_{-i}(x_i)\big)^2 &= \mathrm{E}\big(Y_i - r(x_i) + r(x_i) - \hat{r}_{-i}(x_i)\big)^2 \\ &= \sigma^2 \;+\; \mathrm{E}\big(r(x_i) - \hat{r}_{-i}(x_i)\big)^2 \;\approx\; \sigma^2 \;+\; \mathrm{E}\big(r(x_i) - \hat{r}_{n}(x_i)\big)^2, \end{aligned}$$

e, portanto,

$$\mathrm{E}(\widehat{R}) \;\approx\; R \;+\; \sigma^2 \;=\; \text{risco preditivo}\cdot$$

Assim, o escore de validação cruzada é uma estimativa quase imparcial do risco.

Parece que pode ser demorado avaliar $\widehat{R}(h)$, já que aparentemente precisamos recomputar o estimador após abandonar cada observação. Felizmente, existe uma fórmula de atalho para calcular $\widehat{R}$ para suavizadores lineares e apresentada no seguinte teorema.

Teorema IV.2. Seja $\hat{r}_n$ um suavizador linear. Então, o estimador de validação cruzada do risco $\widehat{R}(h)$ pode ser escrito como

$$\widehat{R}(h) \;=\; \frac{1}{n}\sum_{i=1}^{n}\Big(\frac{Y_i - \hat{r}_n(x_i)}{1 - L_{ii}}\Big)^2,$$

onde $L_{ii} = l_i(x_i)$ é o i-ésimo elemento da diagonal principal da matriz de suavização L.

Demonstração. Ver Fan and Gijbels (1996).∎

O parâmetro de suavização h pode ser escolhido minimizando $\widehat{R}(h)$. É importante notar que não podemos supor que $\widehat{R}(h)$ sempre tenha um mínimo bem definido. Devemos sempre traçar $\widehat{R}(h)$ como uma função de h. Embora interessante, não será desta maneira que encntrarems o valor estimado do parâmetro de suavização.

Em vez de minimizar o escore de validação cruzada, uma alternativa é usar uma aproximação chamada validação cruzada generalizada na qual cada L_{ii} na equação do Teorema IV.2 é substituída por sua média $\frac{1}{n}\sum_{i=1}^{n} L_{ii} = \nu/n$ onde $\nu = \mathrm{tr}(L)$ são os graus efetivos de liberdade. Assim, nós minimizaríamos

$$GCV(h) \;=\; \frac{1}{n}\sum_{i=1}^{n}\frac{1}{n}\sum_{i=1}^{n}\Big(\frac{Y_i - \hat{r}_n(x_i)}{1 - \nu/n}\Big)^2.$$

Geralmente, a largura de banda $\widehat{h}$ que minimiza o escore de validação cruzada generalizada está próximo da largura de banda que minimiza a validação cruzada.

Utilizando a aproximação $(1-x)^{-2} \approx 1 + 2x$ vemos que

$$GCV(h) \;\approx\; \frac{1}{n}\sum_{i=1}^{n}\big(Y_i - \hat{r}_n(x_i)\big)^2 \;+\; \frac{2\nu\widehat{\sigma}^2}{n} \;=\; C_p,$$

onde $\widehat{\sigma}^2 = n^{-1}\sum_{i=1}^{n}\big(Y_i - \hat{r}_n(x_i)\big)^2$. A equação acima é conhecida como a estatística C_p a qual foi originalmente proposta por Colin Mallows como critério para selecionar variáveis nos modelos de regressão linear. Mais geralmente, muitos critérios comuns de seleçãa de largura de banda podem ser escritos na forma

$$B(h) \;=\; \mathcal{D}(n,h) \times \frac{1}{n}\sum_{i=1}^{n}\big(Y_i - \hat{r}_n(x_i)\big)^2,$$

para diferentes escolhas de $\mathcal{D}(n,h)$. Veja Härdle et al. (1988) para detalhes. Além disso, Härdle et al. (1988) provou, sob condições adequadas, os seguintes fatos sobre o valor de $\widehat{h}$ que minimiza $B(h)$. Seja $\widehat{h}_0$ o valor de $\widehat{h}$ que minimiza a perda $L(\widehat{h}) = n^{-1}\sum_{i=1}^{n}\big(\hat{r}_n(x_i) - r(x_i)\big)^2$, e seja h_0 risco mínimo. Então para todo $\widehat{h}$, $\widehat{h}_0$ e h_0 tendem a 0 na taxa $n^{-1/5}$. Além disso, para certas constantes positivas C_1, C_2, σ_1 e σ_2, temos que

$$\begin{aligned} n^{3/10}\big(\widehat{h} - \widehat{h}_0\big) &\approx N(0,\sigma_1^2), & n\big(L(\widehat{h}) - L(\widehat{h}_0)\big) &\approx C_1\chi_1^2 \\ n^{3/10}\big(h_0 - \widehat{h}_0\big) &\approx N(0,\sigma_2^2), & n\big(L(h_0) - L(\widehat{h}_0)\big) &\approx C_2\chi_1^2. \end{aligned}$$

Assim, a taxa relativa de convergência de $\widehat{h}$ é

$$\frac{\widehat{h} - \widehat{h}_0}{\widehat{h}_0} \;=\; O_P\Big(\frac{n^{3/10}}{n^{1/5}}\Big) \;=\; O_P(n^{-1/10}).$$

Essa taxa lenta mostra que é difícil estimar a largura de banda. Essa taxa é intrínseca ao problema de seleção de largura de banda, já que também é verdade que

$$\frac{\widehat{h}_0 - h_0}{h_0} \;=\; O_P\Big(\frac{n^{3/10}}{n^{1/5}}\Big) \;=\; O_P(n^{-1/10}).$$

IV.3 Regressão local

Agora nos voltamos para a regressão não paramétrica local. Suponha que $x_i \in \mathbb{R}$ seja escalar e considere o modelo de regressão no qual a varaiável resposta Y_i está relacionada com a covariável pela equação

$$Y_i \;=\; r(x_i) \;+\; \epsilon_i,$$

onde $\mathrm{E}(\epsilon_i) = 0$, $i = 1, 2, \cdots, n$. Nesta seção, consideramos os estimadores de $r(x)$ obtidos pela média ponderada dos Y_i, dando maior peso àqueles pontos próximos de x. Começamos com o estimador de regressão Kernel.

Definição IV.5. Seja $h > 0$ um número positivo chamado a largura de banda. O estimador kernel Nadaraya–Watson é definido por

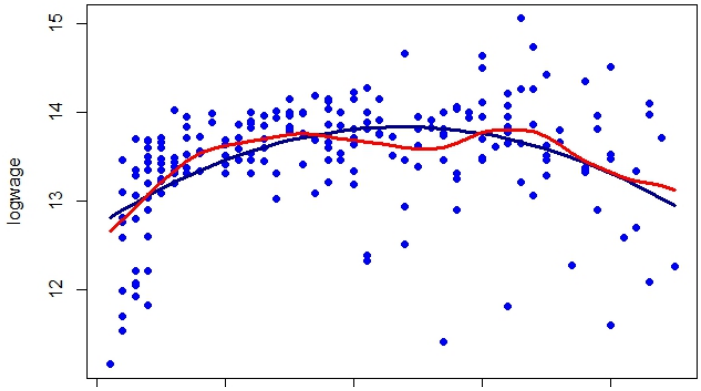
$$\hat{r}_n(x) \;=\; \sum_{i=1}^{n} l_i(x) Y_i,$$

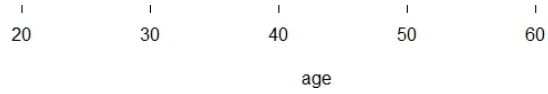
onde K é a função kernel e os pesos $l_i(x)$ são dados por

$$l_i(x) \;=\; \frac{K\Big(\frac{x - x_i}{h}\Big)}{\sum_{j=1}^{n} K\Big(\frac{x - x_j}{h}\Big)}.$$

O estimador de média local no Exemplo IV.3 é um particular estimador Kernel Nadaraya–Watson. Segundo esta definição para encontrarmos estes estimadores somente faltaria encontrar qual deve ser o valor da largura de banda h, mas fixando-a temos completamente definidos os pesos $l_i(x)$ e, portanto, o estimador $\hat{r}_n$.

Exemplo IV.5. (Continuação do Exemplo IV.1) Vamos utilizar novamente os dados deste exemplo para mostrar comparativamente a curva de regressão obtida pelo estimador kernel Nadaraya–Watsonm com o resultado do modelo de regressão linear paramétrico. A largura de banda é $h = 1.892157$, valor obtido por validação cruzada.






```
> library(np)
> model.npar = npregbw(logwage ~ age, data = cps71, regtype = "lc")
> par(mfrow = c(1,1), mar = c(4, 4, 1, 1))
> plot(logwage ~ age, data = cps71, col = "blue", pch=19)
> lines(cps71$age, fitted.values(model.par), col="darkblue", lwd = 3)
> lines(cps71$age, predict(model.npar)$mean, col = "red", lwd = 3)
> summary(model.npar)

Regression Data (205 observations, 1 variable(s)):

Regression Type: Local-Constant
Bandwidth Selection Method: Least Squares Cross-Validation
Formula: logwage ~ age
Bandwidth Type: Fixed
Objective Function Value: 0.316055 (achieved on multistart 1)

Exp. Var. Name: age Bandwidth: 1.892157 Scale Factor: 0.4487742

Continuous Kernel Type: Second-Order Gaussian
No. Continuous Explanatory Vars.: 1
Estimation Time: 0.12 seconds

> predict(model.npar)$R2
[1] 0.3108675

> summary(model.par)$adj.r.squared
[1] 0.2231529
```

Obseve que o R^2 no caso paramémetrico foi 0.2231529, aumentando um pouco no caso não paramétrico para 0.3108675.

A escolha do kernel não é muito importante. As estimativas obtidas usando kernels diferentes são geralmente numericamente muito semelhantes. Esta observação é confirmada por cálculos teóricos que mostram que o risco é muito insensível à escolha do kernel; veja Seção 6.2.3 de Scott (1992). Então mostramos o resultado utilizando o kernel gaussiano, que é padrão do comando. Com esta observação vemos que especificar ou não o kernel não é muito importante.

O que importa muito mais é a escolha da largura de banda h, que controla a quantidade de suavização. Larguras de banda pequenas fornecem estimativas muito aproximadas, enquanto larguras de banda maiores fornecem estimativas mais suaves. Em geral, vamos deixar a largura de banda depender do tamanho da amostra, então às vezes escrevemos h_n.

O teorema a seguir mostra como a largura de banda afeta o estimador. Para declarar estes resultados, precisamos fazer alguma suposição sobre o comportamento de $x_1, \cdots, x_n$ quando n aumenta. Para os propósitos do teorema, assumiremos que estes são escolhidos aleatoriamente de alguma densidade f.

Teorema IV.3. O risco, usando perda de erro quadrada integrada, do estimador kernel Nadaraya-Watson é

$$\begin{aligned} R(\hat{r}_n, r) &= \frac{h_n^4}{4}\left(\int x^2 K(x)\mathrm{d}x\right)^2 \int \left(r''(x) + 2r'(x)\frac{f'(x)}{f(x)}\right)^2 \mathrm{d}x \\ &\quad + \frac{\sigma^2 \int K^2(x)\mathrm{d}x}{nh_n} \int \frac{1}{f(x)}\mathrm{d}x + o(nh_n^{-1}) + o(h_n^4), \end{aligned}$$

quando $h_n \to 0$ e $nh_n \to \infty$.

Demonstração. Ver Scott (1992).∎

O primeiro termo no Teorema IV.3 é o viés ao quadrado e o segundo termo é a variância. O que é especialmente notável é a presença do termo

$$2r'(x)\frac{f'(x)}{f(x)}$$

no viés. Chamamos esta expressão de viés de design, pois depende do design, ou seja, a distribuição dos x_i. Isso significa que o viés é sensível à posição dos x_i. Além disso, pode-se mostrar que os estimadores de kernel também possuem alta tendência perto dos limites. Isso é conhecido como viés de limite. Veremos que podemos reduzir esses vieses usando um refinamento chamado regressão polinomial local.

Se diferenciarmos a expresão no Teorema IV.3 e definirmos o resultado igual a 0, veremos que a largura de banda ideal h_* é

$$h_* = \left(\frac{1}{n}\right)^{1/5} \left(\frac{\sigma^2 \int K^2(x)\mathrm{d}x \int \frac{1}{f(x)}\mathrm{d}x}{\left(\int x^2 K(x)\mathrm{d}x\right)^2 \int \left(r''(x) + 2r'(x)\frac{f'(x)}{f(x)}\right)^2 \mathrm{d}x} \right)^{1/5}.$$

Assim, $h_* = O(n^{-1/5})$. Colocando h_* de volta na expresão do Teorema IV.3, vemos que o risco diminui na taxa $O(n^{-4/5})$. Na maioria dos modelos paramétricos, o risco do estimador de máxima verossimilhança diminui para 0 na taxa $1/n$. A taxa mais lenta $n^{-4/5}$ é o preço de usar métodos não paramétricos. Na prática, não podemos usar a largura de banda dada acima, já que h_* depende da função desconhecida r. Em vez disso, usamos a validação cruzada, conforme descrito no Teorema IV.2.

IV.3.1 Polinômios locais

Os estimadores de kernel sofrem de viés de limite e viés de design. Esses problemas podem ser aliviados usando uma generalização da regressão kernel chamada de regressão polinomial local.

Para motivar este estimador, primeiro considere escolher um estimador $a = \hat{r}_n(x)$ para minimizar as somas de quadrados $\sum_{i=1}^n (Y_i - a)^2$. A solução é a função constante $\hat{r}_n(x) = \overline{Y}$ que obviamente não é um bom estimador de $r(x)$. Agora defina a função de peso $w_i(x) = K\big((x_i - x)/h\big)$ e escolha um $a = \hat{r}_n(x)$ para minimizar as somas de quadrados ponderadas

$$\sum_{i=1}^n w_i(x)\big(Y_i - a\big)^2.$$

Do cálculo, vemos que a solução é

$$\hat{r}_n(x) = \frac{\sum_{i=1}^n w_i(x) Y_i}{\sum_{i=1}^n w_i(x)},$$

que é exatamente o estimador de regressão kernel. Isso nos dá uma interpretação interessante do estimador kernel: é um estimador localmente constante, obtido a partir de mínimos quadrados ponderados localmente.

Isso sugere que podemos melhorar o estimador usando um polinômio local de grau p em vez de uma constante local. Seja x algum valor fixo no qual queremos estimar $r(x)$. Para valores u em uma vizinhança de x, defina o polinômio

$$P_x(u;a) = a_0 + a_1(u-x) + \frac{a_2}{2!}(u-x)^2 + \cdots + \frac{a_p}{p!}(u-x)^p.$$

Podemos aproximar uma função de regressão suave $r(u)$ em uma vizinhança do valor alvo x pelo polinômio:

$$r(u) = P_x(u;a).$$

Estimamos $a = (a_0, \cdots, a_p)^\top$ escolhendo $\widehat{a} = (\widehat{a}_0, \cdots, \widehat{a}_p)^\top$ que minimize a soma de quadrados ponderadas localmente

$$\sum_{i=1}^n w_i(x)\big(Y_i - P_x(X_i;a)\big)^2.$$

O estimador $\widehat{a}$ depende do valor alvo x, então escrevemos $\widehat{a}(x) = (\widehat{a}_0(x), \cdots, \widehat{a}_p(x))^\top$ se quisermos tornar essa dependência explícita. A estimativa local de r é

$$\hat{r}_n(x) = P_x(u;\widehat{a}).$$

Em particular, no valor alvo $u = x$ temos

$$\hat{r}_n(x) = P_x(x;\widehat{a}) = \widehat{a}_0(x).$$

Embora $\hat{r}_n(x)$ dependa apenas de $\widehat{a}_0(x)$, isso não é equivalente a simplesmente ajustar uma constante local.

Por exemplo, caso $p = 0$ retornamos ao estimador kernel. O caso especial em que $p = 1$ é chamado de regressão linear local e esta é a versão que recomendamos como uma opção padrão. Como veremos, estimadores polinomiais locais e, em particular, estimadores lineares locais possuem algumas propriedades notáveis como mostrado por Fan (1992) e Hastie e Loader (1993). Muitos dos resultados que se seguem são desses documentos.

Para encontrar $\widehat{a}(x)$, é útil re-expressar o problema em notação vetorial. Definamos

$$X_x = \begin{pmatrix} 1 & (x_1 - x) & \cdots & \frac{(x_1-x)^p}{p!} \\ 1 & (x_2 - x) & \cdots & \frac{(x_2-x)^p}{p!} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & (x_n - x) & \cdots & \frac{(x_n-x)^p}{p!} \end{pmatrix}$$

e seja W_x a matriz diagonal $n \times n$ cujo componente (i,i) é $w_i(x)$. Podemos então reescrever a soma de quadrados ponderadas localmente como

$$(Y - X_x a)^\top W_x (Y - X_x a) \cdot$$

Minimizando esta expressão fornece-nos o estimador de mínimos quadrados ponderados

$$\widehat{a}(x) = (X_x^\top W_x X_x)^{-1} X_x W_x Y \cdot$$

Em particular, $\hat{r}_n(x) = \widehat{a}_0(x)$ é o produto interno da primeira linha de $(X_x^\top W_x X_x)^{-1} X_x W_x Y$ com Y. Assim nós temos o seguinte teorema.

Teorema IV.4. A estimativa de regressão polinomial local é

$$\hat{r}_n(x) = \sum_{i=1}^n l_i(x) Y_i,$$

onde $l(x)^\top = (l_1(x), \cdots, l_n(x))$,

$$l^\top = e_1^\top (X_x^\top W_x X_x)^{-1} X_x^\top W_x,$$

$e_1 = (1, 0, \cdots, 0)^\top$ e X_x e W_x como definidos anteriormente. Este estimador têm por média

$$\mathrm{E}\big(\hat{r}_n(x)\big) = \sum_{i=1}^n l_i(x) r(x_i)$$

e variância

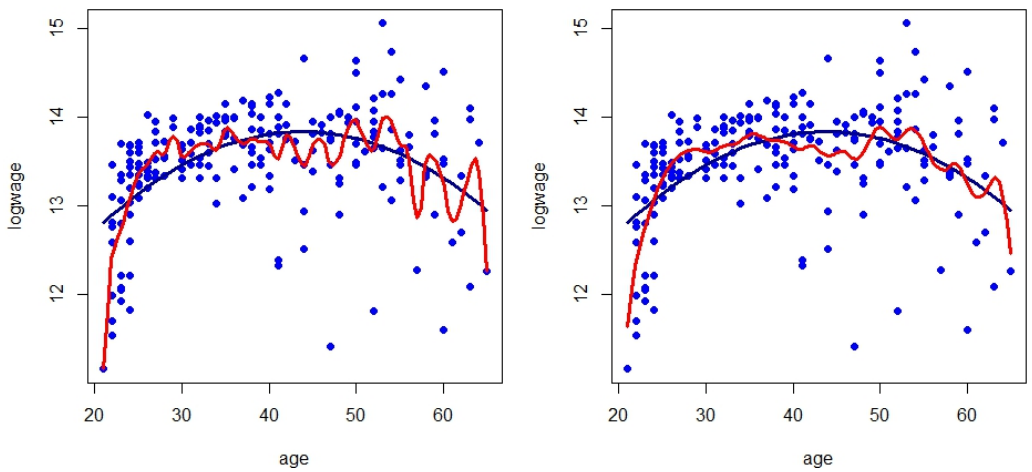
$$\mathrm{Var}\big(\hat{r}_n(x)\big) = \sigma^2 \sum_{i=1}^n l_i(x)^2 = \sigma^2 ||l(x)||^2 \cdot$$

Demonstração. Ver Scott (1992).

Mais uma vez, nossa estimativa é linear e podemos escolher a largura de banda por validação cruzada.

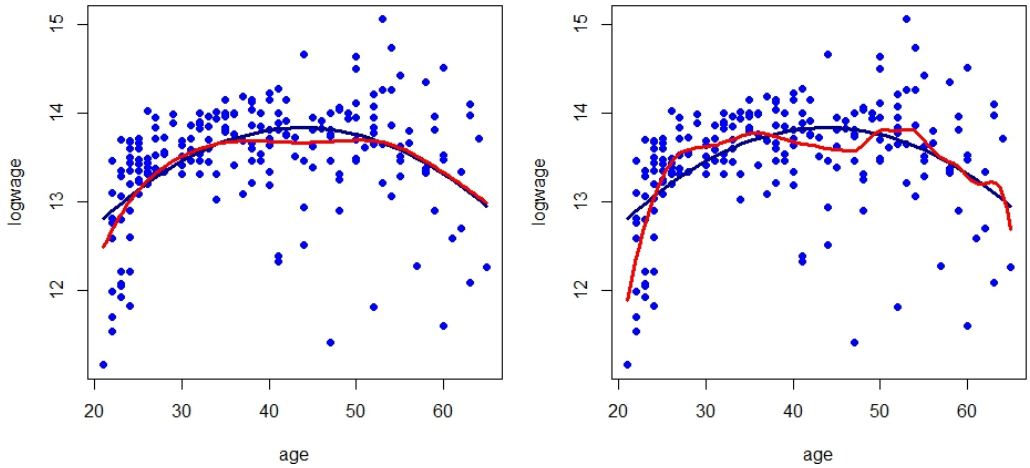
Exemplo IV.6. (Continuação do Exemplo IV.1) Vamos utilzar novamente os dados deste exemplo assim como o resultado do modelo de regressão linear paramétrico para mostrar, escolhendo diversos valores da largura de banda h, a curva de regressão obtida estimativa da regressão polinomial local.

Os dois primeiros gráficos são baseados em pequenas larguras de banda: a esquerda $h = 0.5$ e a direita $h = 1.0$, para isto utilizamos a opção bw no comando **locpol** (Local Polynomial estimation.) no pacote de funções homônimo.



```
> library(locpol)
> model.npar01 = locpol(logwage ~ age, data = cps71, kernel = gaussK, bw = 0.5)
> par(mfrow = c(1,1), mar = c(4, 4, 1, 1))
> plot(logwage ~ age, data = cps71, col = "blue", pch=19)
> lines(cps71$age, fitted.values(model.par), col="darkblue", lwd = 3)
> lines(model.npar01$lpFit[,1], model.npar01$lpFit[,2], col = "red", lwd = 3)
> model.npar02 = locpol(logwage ~ age, data = cps71, kernel = gaussK, bw = 1.0)
> par(mfrow = c(1,1), mar = c(4, 4, 1, 1))
> plot(logwage ~ age, data = cps71, col = "blue", pch=19)
> lines(cps71$age, fitted.values(model.par), col="darkblue", lwd = 3)
> lines(model.npar02$lpFit[,1], model.npar02$lpFit[,2], col = "red", lwd = 3)
```

Agora essolhemos as larguras de banda $h = 5.0$, no gráfico esquerdo, e $\widehat{h} = 1.49995$ no gráfico direito. Este valor obtido por validação cruzada. Observemos que, a medida que a largura de banda h aumenta, a função estimada passa de áspera demais para suave demais.




```
> model.npar03 = locpol(logwage ~ age, data = cps71, kernel = gaussK, bw = 5.0)
> par(mfrow = c(1,1), mar = c(4, 4, 1, 1))
> plot(logwage ~ age, data = cps71, col = "blue", pch=19)
> lines(cps71$age, fitted.values(model.par), col="darkblue", lwd = 3)
> lines(model.npar03$lpFit[,1], model.npar03$lpFit[,2], col = "red", lwd = 3)
> h = regCVBwSelC(cps71$age, cps71$logwage, deg = 1, kernel = gaussK)
> h
[1] 1.49995
> model.npar04 = locpol(logwage ~ age, data = cps71, kernel = gaussK)
> par(mfrow = c(1,1), mar = c(4, 4, 1, 1))
> plot(logwage ~ age, data = cps71, col = "blue", pch=19)
> lines(cps71$age, fitted.values(model.par), col="darkblue", lwd = 3)
> lines(model.npar04$lpFit[,1], model.npar04$lpFit[,2], col = "red", lwd = 3)
```

Nos últimos gráficos o da esquerda é baseado numa grande largura de banda $h = 5.0$ e o ajuste é muito suave. O gráfico do canto direito está correto, é aquele onde a largura de banda foi escolhida por validação cruzada. A função para obtermos $\widehat{h}$ é **regCVBwSelC**. Foi utilizada para mostrar o procedimento mas não é necessário em situações práticas, é padrão, bastando não dizer nada como em **model.npar04**. O gráfico esquerdo também mostra a presença de viés perto dos limites. Como veremos, esta é uma característica geral da regressão kernel.

O teorema a seguir apresenta o comportamento amostral em amostras grandes do risco do estimador linear local e mostra por que a regressão linear local é melhor que a regressão kernel.

Teorema IV.5. Seja $Y_i = r(X_i) + \sigma(X_i)\epsilon_i$, para $i = 1, \cdots, n$ e $a \leq X_i \leq b$. Assumamos que $X_1, \cdots, X_n$ seja uma amostra aleatória com densidade f e que

(i) $f(x) > 0$,
(ii) f, r'' e σ^2 sejam contínuos numa vizinhança de x,

(iii) $h_n \to 0$ e $nh_n \to \infty$.

Seja $x \in (a,b)$. Dado $X_1, \cdots, X_n$ temos o seguinte: o estimador linear local e o estimador kernel ambos têm variância

$$\frac{\sigma^2(x)}{f(x)nh_n}\int K^2(u)\mathrm{d}u + o_P\Big(\frac{1}{nh_n}\Big).$$

O estimador de kernel Nadaraya - Watson tem viés

$$h_n^2\left(\frac{1}{2}r''(x) + \frac{r'(x)f'(x)}{f(x)}\right)\int u^2K(u)\mathrm{d}u + o_P(h^2)$$

enquanto que o estimador linear local tem um viés assintótico

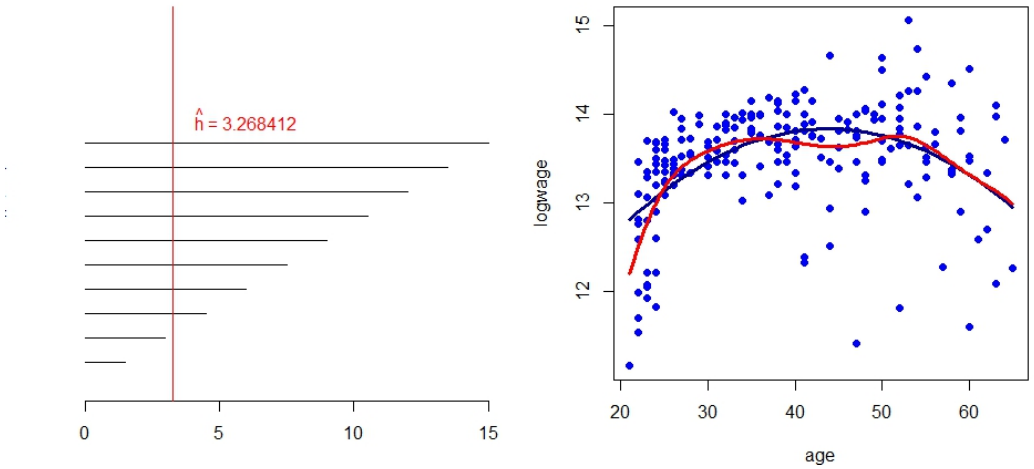
$$h_n^2\frac{1}{2}r''(x)\int u^2K(u)\mathrm{d}u + o_P(h^2).$$

Demonstração. Fan (1992) e Fan e Gijbels (1996).

Assim, o estimador linear local é livre de viés de projeto, é livre de f. Nos pontos de fronteira a e b, o estimador kernel de Nadaraya-Watson tem um viés assintótico de ordem h_n, enquanto o estimador linear local tem viés de ordem h_n^2. Nesse sentido, a estimativa linear local elimina o viés nos limites.

Sabemos que o procedimento mais adequado, dentre os vistos até agora, é o de polinômios locais. Dediquemos atenção agora a um detalhe na escolha do parâmetro de alisamento h por validação cruzada. Dizemos que dentre os gráficos mostrados no Exemplo IV.5, o último deles a direita é o correto e isso foi afirmado pelo fato de utilizarmos a validação cruzada para encontrarmos a estimativa do parâmetro de alisamento. Mas, não é bem assim, a seleção por validação cruzada é um pouco mais complexa, como veremos no próximo exemplo.

Exemplo IV.7. Queremos entender a dependência entre a estimativa $\widehat{h}$ da largura de banda e o intervalo onde procurar-la, também queremos obter o valor do R^2 na regressão local. Para isso vamos explorar mais dois detalhes do procedimento escolhido para estimarmos a regressão polinomial local.



O gráfico à esquerda dedica-se a pesquisar a relação entre o intervalo onde procurar a largura de banda e o valor da estimativa desta, ou seja, a estimativa de h. Para isso definimos dois vetores do mesmo comprimento, sup e hn. No primeiro deles indicaremos o extremo superior dos intervalos onde procuraremos a estimativa da largura de banda, no segundo guardamos a estimativa da largura de banda obtida.

```
> hn = sup = rep(0, 10)
> for(i in 1:10) {sup[i] = 1.5*i}
> for(i in 1:10){
        hn[i] = regCVBwSelC(cps71$age, cps71$logwage, deg = 1,
                                                    kernel = gaussK, interval = c(0,sup[i]))
  }
```

Mostramos os valores escolhidos para os extremos dos intervalos e as estimativas de h encontradas.

```
> sup
 [1]  1.5  3.0  4.5  6.0  7.5  9.0 10.5 12.0 13.5 15.0
> hn
 [1] 1.499950 2.999952 3.268412 3.268412 3.268417 3.268412 3.268411 3.268422 3.268414
[10] 3.268419
```

Os comandos a seguir permitem-nos obter o gráfico acima a esquerda.

```
> par(mfrow = c(1,1), mar = c(3, 3, 1, 1))
> plot(seq(0,15), seq(0,15), col = "blue", pch=19, type = "n", axes = FALSE)
> segments(0,1,sup[1],1)
> axis(1)
> for(i in 1:10) segments(0,i,sup[i],i)
> abline(v=hn[3], col= "red")
> text(6,11,expression(paste(hat(h)," = 3.268412")), col="red")
```

Podemos perceber que, caso não especifiquemos o intervalo de busca, podemos obter uma estimativa de h que não seja a correta. Conforme ampliamos o intervalo desta busca percebemos que os primeiros dois valores estimados correspondem ao limite superior, sendo isto um indicativo que devemos modificar o extremo superior do intervalo de busca. Uma vez feito, encontramos o valor de $\widehat{h}$ correto, ou seja, não importa quão mais amplo seja o intervalo de busca que a estimativa encontrada não muda. Encontramos então o polinômio local mais adequado à nossos dados.

```
> model.npar05 = locpol(logwage ~ age, data = cps71, kernel = gaussK, bw = hn[3],
                                                                xevalLen = length(cps71$age))
> par(mfrow = c(1,1), mar = c(4, 4, 1, 1))
> plot(logwage ~ age, data = cps71, col = "blue", pch=19)
> lines(cps71$age, fitted.values(model.par), col="darkblue", lwd = 3)
> lines(model.npar05$lpFit[,1], model.npar05$lpFit[,2], col = "red", lwd = 3)
```

Resolvemos o primeiro problema, como termos certeza de que o valor obtido de $\widehat{h}$ seja o correto. Queremos agora encontrar o valor do R^2 nesta situação. Aconte que, por padrão, o comando **locpol** somente avalia em **xevalLen = 100**, ou seja, avalia em no máximo 100 pontos a curva estimada. Isto caso sejam mais do que 100 os valores da variável regressora.

Em nosso exemplo temos um total de

```
> length(cps71$age)
[1] 205
```

pontos amostrais. Isso implica que seja necessário informar ao comando **locpol** que queremos tantos pontos estimados quantos pontos amostrais. Para isso utilizamos a opção **xevalLen = length(cps71$age)** e, dessa forma, indicamos que queremos tantas estimativas de $\hat{r}$ quantos dados amostrais. Podemos então avaliar na função **R2**.

```
> R2 = function(Y,r){
  media = mean(Y)
  soma1 = sum((Y-media)^2)
  soma2 = sum((r-media)^2)
  soma0 = sum((Y-media)*(r-media))
  R2 = (soma0^2)/(soma1*soma2)
  return(R2)
}
> R2(cps71$logwage,model.npar05$lpFit[,2])
[1] 0.2623865
```

A função **R2** foi criada por nás e tem por entrada os vatores de resposta Y e de estimativas $\hat{r}$. Por resultado temos que $R^2 = 0.26$ é o valor do coeficiente de determinação para a regressão polinomial local, valor ligeiramente superior àquele obtido com o modelo de regressão lineal.

IV.4 Splines

Considere mais uma vez o modelo de regressão

$$Y_i = r(x_i) + \epsilon_i$$

e suponha que estimamos r escolhendo $\hat{r}_n(x)$ que minimiza as somas de quadrados

$$\sum_{i=1}^{n} \left(Y_i - \hat{r}_n(x_i)\right)^2.$$

A minimização de todas as funções lineares, isto é, das funções da forma $\beta_0 + \beta_1 x$ produz o estimador de mínimos quadrados. Minimizar todas as funções produz uma função que interpola os dados. Anteriormente evitamos essas duas soluções extremas, substituindo as somas de quadrados por somas de quadrados ponderadas localmente. Uma maneira alternativa de obter soluções entre esses extremos é minimizar as **somas de quadrados penalizadas**

$$M(\lambda) = \sum_{i=1}^{n} \left(Y_i - \hat{r}_n(x_i)\right)^2 + \lambda J(r),$$

onde $J(r)$ é uma penalidade de aspereza. Adicionar um termo de penalidade ao critério que estamos otimizando é às vezes chamado de **regularização**.

Vamos nos concentrar no caso especial

$$J(r) = \int \left(r''(x)\right)^2 \mathrm{d}x.$$

O parâmetro λ controla a troca entre o ajuste e a penalidade. Vamos denotar por $\hat{r}_n$ a função que minimiza $M(\lambda)$. Quando $\lambda = 0$, a solução é a função de interpolação. Quando $\lambda \to \infty$, $\hat{r}_n$ converge para a linha de mínimos quadrados. O parâmetro λ controla a quantidade de suavização. Como é a aparência do $\hat{r}_n$ para $0 < \lambda < \infty$? Para responder a essa pergunta, precisamos definir splines.

Spline é um polinômio especial por partes, mais detalhes sobre splines podem ser encontrados em Wahba (1990). Os splines mais utilizadas são splines cúbicos por partes.

Definição IV.6. Seja $\xi_1 < \xi_2 < \cdots < \xi_k$ um conjunto de pontos ordenados, chamados nodos, contidos em algum intervalo (a, b). Um spline cúbico é uma função contínua r tal que:

(i) r é um polinômio cúbico sobre $(\xi_1, \xi_2), (\xi_2, \xi_3), \cdots$ e
(ii) r tem primeira e segunda derivadas contínuas nos nós.

Geralmente mais, um spline de ordem m é um polinômio de grau $m - 1$ por partes com $m - 2$ derivadas contínuas nos nós. Um spline que é linear além dos nós limítrofes é chamado de spline natural.

Splines cúbicos, ou seja, com $m = 4$ são os splines mais comuns usados na prática. Eles surgem naturalmente na estrutura de regressão penalizada, como mostra o seguinte teorema.

Teorema IV.6. A função $\hat{r}_n(x)$ que minimiza

$$M(\lambda) = \sum_{i=1}^{n} \left(Y_i - \hat{r}_n(x_i)\right)^2 + \lambda \int \left(r''(x)\right)^2 \mathrm{d}x$$

é um spline cúbico natural com nós nos pontos de dados. O estimador $\hat{r}_n$ é chamado de spline de suavização.

Demonstração. Wahba (1990).∎

O teorema acima não fornece uma forma explícita para $\hat{r}_n$. Para fazer isso, vamos construir uma base para o conjunto de splines.

Teorema IV.7. Sejam $\xi_1 < \xi_2 < \cdots < \xi_k$ nós contidos em um intervalo (a, b). Definimos $h_1(x) = 1$, $h_2(x) = x$, $h_3(x) = x^2$, $h_4(x) = x^3$ e $h_j(x) = (x - \xi_{j-4})_+^3$ para $j = 5, \cdots, k + 4$. As funções $\{h_1, \cdots, h_{k+4}\}$ formam uma base para o conjunto de splines cúbicos nesses nós, chamada de **base do poder truncado**. Assim, qualquer spline cúbico $r(x)$ com esses nós pode ser escrito como

$$r(x) = \sum_{j=1}^{k+4} \beta_j h_j(x).$$

Demonstração. Wahba (1990).

Agora, introduzimos uma base diferente para o conjunto de splines naturais, chamada de base B-spline, que é particularmente adequada para computação. Estes são definidos da seguinte maneira.

Seja $\xi_0 = a$ e $\xi_{k+1} = b$. Definimos novos nós $\tau_1, \cdots, \tau_m$ tais que

$$\tau_1 \leq \tau_2 \leq \tau_3 \leq \cdots \leq \tau_m \leq \xi_0,$$

$\tau_{j+m} = \xi_j$ para $j = 1, \cdots, k$ e

$$\xi_{k+1} \leq \tau_{k+m+1} \leq \cdots \leq \tau_{k+2m}.$$

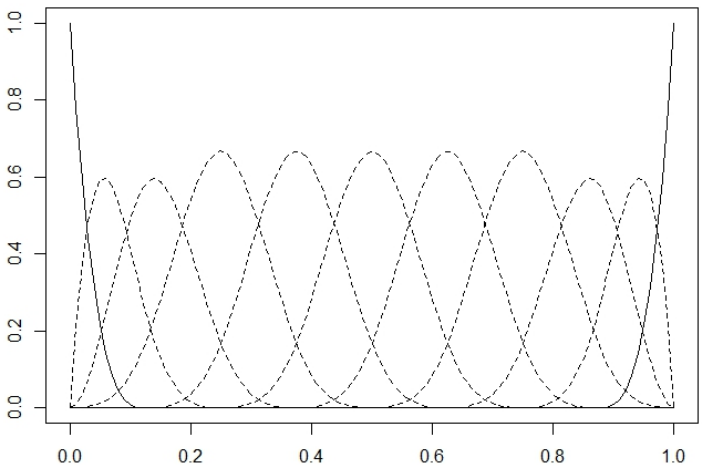
A escolha dos nós extras é arbitrária; geralmente escolhemos $\tau_1 = \cdots = \tau_m = \xi_0$ e $\xi_{k+1} = \tau_{k+m+1} = \cdots = \tau_{k+2m}$. Definimos as funções na base recursivamente do seguinte modo. Primeiro definimos

$$B_{i,1}(x) = \begin{cases} 1, & \text{caso } \tau_i \leq x < \tau_{i+1} \\ 0, & \text{caso contrário} \end{cases},$$

para $i = 1, 2, \cdots, k + 2m - 1$. Em seguida, para $r \leq m$ definimos

$$B_{i,r}(x) = \frac{x - \tau_i}{\tau_{i+r-1} - \tau_i} B_{i,r-1}(x) + \frac{\tau_{i+r} - x}{\tau_{i+r} - \tau_{i+1}} B_{i+1,r-1}(x),$$

para $i = 1, \cdots, k + 2m - r$. Entende-se que, se o denominador é 0, a função é definida como 0.



Base cúbica B-spline usando nove nós igualmente espaçados em (0,1).

O gráfico acima foi obtido pelos comandos a seguir.

```
> library(splines)
> x = seq(0,1, by = 0.01)
> curvas = bs(x, knots = seq(0,1,length.out = 9), degree = 3)
> par(mfrow = c(1,1), mar = c(2, 2, 1, 1))
> plot(seq(0,1,length.out = 101), curvas[,1], type = "l", ylim = c(0,1))
> for(i in 2:10) lines(seq(0,1,length.out = 101), curvas[,i], lty = 2)
> lines(seq(0,1,length.out = 101), curvas[,11], lty = 1)
```

Por padrão, os splines são cúbicos, ou seja, com $m = 4$ ou $degree = 3$ não sendo necessário especificar.

Teorema IV.8. As funções $\{B_{i,4},\ i = 1, \cdots, k + 4\}$ são uma base para o conjunto de splines cúbicos. Eles são chamados de

funções de base B-spline.

Demonstração. Hastie et al. (2001).

A vantagem das funções de base B-spline é que elas possuem suporte compacto que torna possível acelerar os cálculos. Veja Hastie et al. (2001) para detalhes. A figura acima mostra a base cúbica B-spline usando nove nós igualmente espaçados em (0,1).

Estamos agora em condições de descrever o estimador spline em mais detalhes. De acordo com o Teorema IV.6, $\hat{r}$ é um spline cúbico natural. Portanto, podemos escrever

$$\hat{r}_n(x) = \sum_{i=1}^{N} \widehat{\beta}_j B_j(x),$$

onde $B_1, \cdots, B_N$ são uma base para os splines naturais, como os B-splines com $N = n + 4$. Assim, precisamos apenas encontrar os coeficientes $\widehat{\beta} = (\widehat{\beta}_1, \cdots, \widehat{\beta}_N)^\top$. Ao expandir r na base, podemos agora reescrever a minimização da seguinte forma:

$$\text{minimizar:} \qquad (Y - B\beta)^\top (Y - B\beta) + \lambda \beta^\top \Omega \beta,$$

onde $B_{i,j} = B_j(X_i)$ e $\Omega_{jk} = \int B_j''(x) B_k''(x) \mathrm{d}x$.

Teorema IV.9. O valor de β que minimiza $(Y - B\beta)^\top (Y - B\beta) + \lambda \beta^\top \Omega \beta$ é

$$\widehat{\beta} = (B^\top B + \lambda \Omega)^{-1} B^\top Y.$$

Demonstração. Hastie et al. (2001).

Deste teorema concluímos que os splines são outro exemplo de suavizadores lineares.

Teorema IV.10. O spline de suavização $\hat{r}_n(x)$ é um suavizador linear, ou seja, existem pesos $l(x)$ tais que $\hat{r}_n(x) = \sum_{i=1}^n Y_i l_i(x)$. Em particular, a matriz de alisamento L é

$$L = B(B^\top B + \lambda \Omega)^{-1} B^\top,$$

e o vetor r de valores ajustados é dado por

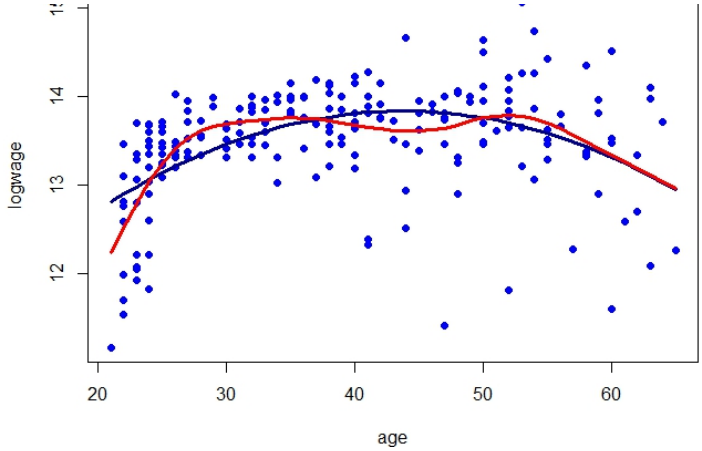
$$r = LY.$$

Demonstração. Hastie et al. (2001).

Se tivéssemos feito uma regressão linear ordinária de Y em B, a matriz chapéu seria $L = B(B^\top B)^{-1} B^\top$ e os valores ajustados interpolariam os dados observados. O efeito do termo $\lambda\Omega$ é reduzir os coeficientes de regressão para um subespaço, o que resulta em um ajuste mais suave. Como antes, nós definimos os graus de liberdade efetivos por $\nu = \mathrm{tr}(L)$ e escolhemos o parâmetro de suavização λ por validação cruzada.

Exemplo IV.8. Novamente utilizamos os dados do Exemplo IV.1 assim como o resultado do modelo de regressão linear paramétrico para mostrar a curva de regressão não paramétrica obtida por splines. As funções **smooth.spline** e **bs**, esta última utilizada anteriormente, estão disponíveis no pacote **splines**.

```
> model.npar06 = with(cps71, smooth.spline(age, logwage))
> model.npar06
Call:
smooth.spline(x = age, y = logwage)

Smoothing Parameter  spar= 0.6265389  lambda= 0.001355428 (11 iterations)
Equivalent Degrees of Freedom (Df): 7.803765
Penalized Criterion (RSS): 8.241849
GCV: 0.2925978
```



A figura mostra o spline de suavização obtido por validação cruzada para os dados em **cps71**. O número efetivo de graus de liberdade é 7.8. O ajuste é tão suave quanto o obtido com o estimador de regressão local. A diferença entre os dois ajustes é pequena comparada com a largura das bandas de confiança que iremos calcular mais tarde.

```
> par(mfrow = c(1,1), mar = c(4, 4, 1, 1))
> plot(logwage ~ age, data = cps71, col = "blue", pch=19)
> lines(cps71$age, fitted.values(model.par), col="darkblue", lwd = 3)
> lines(model.npar06, col = "red", lwd = 3)
```

Para podermos clacular o valor do R^2 nesta situação procedemos como mostrado a seguir.

```
> R2(cps71$logwage,fitted(model.npar06))
[1] 0.3290406
```

Observa-se que o valor do $R^2 = 0.33$ é muito maior do que o obtido por outros modelos apresentados.

IV.5 Estimação da variância

Consideramos vários métodos para estimar σ^2. Para suavizadores lineares, existe uma estimativa simples e quase isenta de σ^2.

Teorema IV.11 Seja $\hat{r}_n$ um suavizador linear. Então

$$\widehat{\sigma}^2 = \frac{\sum_{i=1}^{n} \left(Y_i - \hat{r}_n(x_i)\right)^2}{n - 2\nu + \tilde{\nu}},$$

onde

$$\nu = \mathrm{tr}(L) \qquad \text{e} \qquad \tilde{\nu} = \mathrm{tr}(L^\top L) = \sum_{i=1}^{n} ||l(x_i)||^2.$$

Se r é suficientemente suave, $\nu = o(n)$ e $\tilde{\nu} = o(n)$, então $\widehat{\sigma}^2$ é um estimador consistente de σ^2.

Demonstração. Lembremos que se Y é um vetor aleatório e Q é uma matriz simétrica, então $T^\top QY$ é chamado de forma quadrática e é bem conhecido que

$$\mathrm{E}(T^\top QY) = \mathrm{tr}(QV) + \mu^\top Q\mu,$$

onde $V = \mathrm{Var}(Y)$ é a matriz de covariancias de Y e $\mu = \mathrm{E}(Y)$ é a média do vetor. Agora

$$Y - r = Y - LY = (I - L)Y$$

e também

$$\widehat{\sigma}^2 \;=\; \frac{Y^\top \Lambda Y}{\mathrm{tr}(\Lambda)},$$

onde $\Lambda = (I-L)^\top (I-L)$. Conseqüentemente

$$\mathrm{E}(\widehat{\sigma}^2) \;=\; \frac{\mathrm{E}(T^\top \Lambda Y)}{\mathrm{tr}(\Lambda)} \;=\; \sigma^2 \;+\; \frac{r^\top \Lambda r}{n - 2\nu + \tilde{\nu}}.$$

Assumindo que ν e $\widehat{\nu}$ não cresçam muito rapidamente e que r seja suave, o último termo é pequeno para n grande e, portanto, $\mathrm{E}(\widehat{\sigma}^2) \approx \sigma^2$. Da mesma forma, pode-se mostrar que $\lim_{n\to\infty} \mathrm{Var}(\widehat{\sigma}^2) = 0$. ∎

Aqui mostramos um outro estimador, devido a Rice (1984). Suponha que os x_i estejam ordenados. Definimos

$$\widehat{\sigma}^2 \;=\; \frac{1}{2(n-1)} \sum_{i=1}^{n-1} \big(Y_{i+1} - Y_i\big)^2.$$

A motivação para este estimador é a seguinte. Assumindo que $r(x)$ é suave, temos que $r(x_{i+1}) - r(x_i) \approx 0$ e portanto

$$Y_{i+1} - Y_i \;=\; \big(r(x_{i+1}) + \epsilon_{i+1}\big) \;-\; \Big(r(x_i) + \epsilon_i\Big) \;\approx\; \epsilon_{i+1} - \epsilon_i$$

e, portanto $\big(Y_{i+1} - Y_i\big)^2 \approx \epsilon_{i+1}^2 + \epsilon_i^2 - 2\epsilon_{i+1}\epsilon_i$. Assim sendo,

$$\begin{aligned} \mathrm{E}\big(Y_{i+1} - Y_i\big)^2 \;&\approx\; \mathrm{E}(\epsilon_{i+1}^2) \;+\; \mathrm{E}(\epsilon_i^2) \;-\; 2\mathrm{E}(\epsilon_{i+1})\mathrm{E}(\epsilon_i) \\ &=\; \mathrm{E}(\epsilon_{i+1}^2) \;+\; \mathrm{E}(\epsilon_i^2) \;=\; 2\sigma^2. \end{aligned}$$

Assim, $\mathrm{E}(\widehat{\sigma}^2) \approx \sigma^2$. Uma variação deste estimador, devido a Gasser et al. (1986) é

$$\widehat{\sigma}^2 = \frac{1}{n-2} \sum_{i=2}^{n-1} c_i^2 \delta_i^2,$$

onde

$$\begin{aligned} \delta_i \;&=\; a_i Y_{i-1} \;+\; b_i Y_{i+1} \;-\; Y_i, \qquad & a_i \;&=\; (x_{i+1} - x_i)/(x_{i+1} - x_{i-1}), \\ b_i \;&=\; (x_i - x_{i-1})/(x_{i+1} - x_{i-1}), & c_i^2 \;&=\; (a_i^2 \;+\; b_i^2 \;+\; 1)^{-1}. \end{aligned}$$

A intuição deste estimador é que é a média dos resíduos que resultam da montagem de uma linha ao primeiro e terceiro ponto de cada triplo consecutivo de pontos.

Exemplo IV.9. Vamos utilizar novamente os dados do Exemplo IV.1. Nessa situação avaliamos os diferentes estimadores da variância apresentados até o momentos. O valor do estimador da variâcia no Teorema IV.11 é saída padrão do comando **locpol**. Assim, podemos observar a estimativa da seguinte forma:

```
> resumo = summary(model.npar04)

Kernel =
        dnorm(x, 0, 1)

    n deg      bw       ase
  205   1 1.49995 0.2612706
> resumo$ase
[1] 0.2612706
```

Para avaliarmos o estimador de Rice (1984) podemos executar os seguintes comandos:

```
> s2 = sum(diff(cps71$logwage)^2)/(2*(resumo$n-1))
> s2
[1] 0.3233995
```

O estimador proposto por Gasser et al. (1986) é um pouco mais complexo e nesta situação não pode ser calculado. Isto acontece porque temos muitos valores iguais na variável explicativa $cps71\$age$.

Até agora assumimos a homocedasticidade, significando que $\sigma^2 = \mathrm{Var}(\epsilon_i)$ não varia com x. No Exemplo IV.1 isso é flagrantemente falso. Claramente, σ^2 aumenta com x, então os dados são heterocedásticos. A estimativa da função $\hat{r}_n(x)$ é relativamente insensível à heterocedasticidade. No entanto, quando se trata de fazer bandas de confiança para $r(x)$, devemos levar

em conta a variância não constante.

Vamos assumir a seguinte abordagem. Veja Yu and Jones (2004) e referências para outras abordagens. Suponha que

$$Y_i = r(x_i) + \sigma(x_i)\epsilon_i.$$

Seja $Z_i = \log\big(Y_i - r(x_i)\big)^2$ e $\delta_i = \log(\epsilon_i^2)$. Então,

$$Z_i = \log\big(\sigma^2(x_i)\big) + \delta_i.$$

Isto sugere obter a estimativa de $\log\big(\sigma^2(x)\big)$ pela regressão do logaritmo dos resíduos ao quadrado em x.

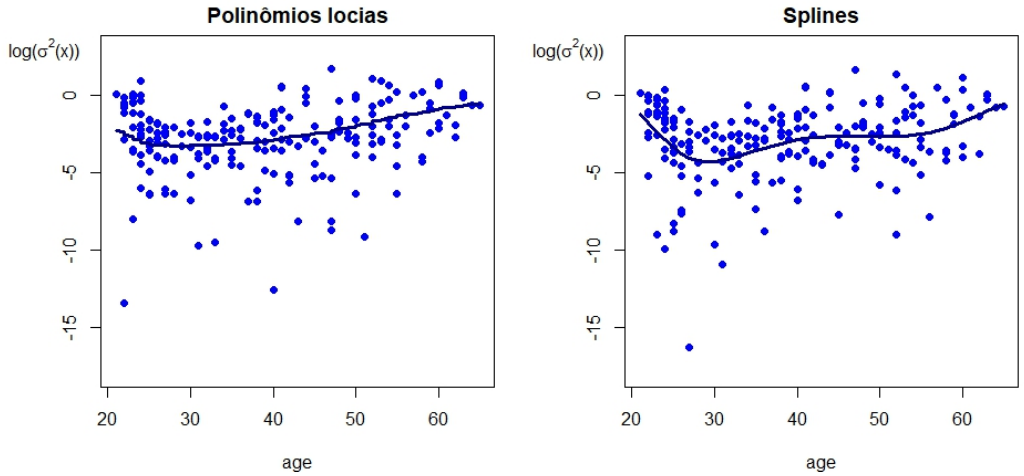
Prossegimos como segue:

1. Estimar $r(x)$ com qualquer método não paramétrico para obter a estimativa $\hat{r}_n(x)$.
2. Definir $Z_i = \log\big(Y_i - \hat{r}_n(x_i)\big)^2$.
3. Obter uma curva de regressão de Z nos x, novamente usando qualquer método não paramétrico, para obter uma estimativa $\hat{q}(x)$ do $\log\big(\sigma^2(x)\big)$ e calcular

$$\widehat{\sigma}^2(x) = e^{\hat{q}(x)}.$$

Uma desvantagem dessa abordagem é que o logaritmo de um resíduo muito pequeno será um grande outlier. Uma alternativa é suavizar diretamente os resíduos quadrados.

Exemplo IV.10. Utilizaremos os dados do Exemplo IV.1. O objetivo agora é mostrar a curva $\log\big(\sigma^2(x)\big)$ para cada um dos modelos estudados e utilizados nos dados deste exemplo, ou seja, mostrarmos a curva heterocedástica do logaritmo da variância para os modelos de regressão não paramétrico de polinômios locais e splines. Procedemos como descrito anteriormente e as curvas obtidas apresentm-se nos gráficos a seguir.



Em ambos gráficos mantivemos a mesma escala nos eixos, isto com o ojetivo de melhor visualizar as diferenças entre as curvas e entre os pontos, que são o logaritmo dos resíduos ao quadrado. Percedemos que tanto os resíduos quanto as curvas são bem diferentes entre estes modelos.

```
> # Polinomio local
> Z = log((cps71$logwage-model.npar05$lpFit[,2])^2)
> hn = regCVBwSelC(cps71$age, Z, deg = 1, kernel = gaussK, interval = c(0,10))
> model.npar05.Z = locpol(Z ~ age, data = cps71, kernel = gaussK, bw = hn,
                          xevalLen = length(cps71$age))
> sigma05 = exp(model.npar05.Z$lpFit[,2])
> par(mfrow = c(1,1), mar = c(4, 5, 2, 1))
> plot(Z ~ age, data = cps71, col = "blue", pch=19, ylab = "", ylim = c(-18,3),
     main = "Polinômios locias")
> mtext(expression(paste("log(",sigma^2,"(x))")), side = 2, adj = 1,
      las = 1, at = 3, line = 1)
> lines(cps71$age, model.npar05.Z$lpFit[,2], col="darkblue", lwd = 3)
> # Spline
> Z1 = log((cps71$logwage-fitted(model.npar06))^2)
> model.npar06.Z1 = with(cps71, smooth.spline(age, Z1))
> sigma06 = exp(fitted(model.npar06.Z1))
> par(mfrow = c(1,1), mar = c(4, 5, 2, 1))
> plot(Z1 ~ age, data = cps71, col = "blue", pch=19, ylab = "", ylim = c(-18,3),
     main = "Splines")
> mtext(expression(paste("log(",sigma^2,"(x))")), side = 2, adj = 1,
```

```
      las = 1, at = 3, line = 1)
> lines(cps71$age, fitted(model.npar06.Z1), col="darkblue", lwd = 3)
```

Com as linhas acima fizemos os gráficos mostrados e ainda obtivemos a variáncia de cada modelo armazenadas nos objetos **sigma05** e **sigma06**, respectivamente.

IV.6 Bandas de confiança

Nesta seção, vamos construir faixas de confiança para $r(x)$. Normalmente, essas bandas são da forma

$$\hat{r}_n(x) \pm c\,\mathrm{se}(x),$$

onde $\mathrm{se}(x)$ é uma estimativa do desvio padrão de $\hat{r}_n(x)$ e $c > 0$ é alguma constante. Antes de prosseguirmos, discutimos um problema pernicioso que surge sempre que fazemos o alisamento, ou seja, o problema do viés.

O problema do viés

Bandas de confiança não são realmente bandas de confiança para $r(x)$, ao contrário, elas são bandas de confiança para $\hat{r}_n(x) = \mathrm{E}\big(\hat{r}_n(x)\big)$ que podemos imaginar como uma versão suavizada de $r(x)$. Obter um conjunto de confiança para a verdadeira função $r(x)$ é complicado por razões que agora explicamos.

Denote a média e o desvio padrão de $\hat{r}_n(x)$ por $\overline{r}_n(x)$ e $s_n(x)$. Então,

$$\begin{aligned}\frac{\hat{r}_n(x) - r(x)}{s_n(x)} &= \frac{\hat{r}_n(x) - \overline{r}_n(x)}{s_n(x)} + \frac{\overline{r}_n(x) - r(x)}{s_n(x)} \\ &= Z_n(x) + + \frac{\mathrm{bias}\big(\hat{r}_n(x)\big)}{\sqrt{\mathrm{variância}\big(\hat{r}_n(x)\big)}},\end{aligned}$$

onde $Z_n(x) = \big(\hat{r}_n(x) - \overline{r}_n(x)\big)/s_n(x)$. Tipicamente, o primeiro termo $Z_n(x)$ converge para uma Normal padrão do qual derivam as faixas de confiança. O segundo termo é o viés dividido pelo desvio padrão. Na inferência paramétrica, o viés é geralmente menor que o desvio padrão do estimador, portanto, esse termo vai para zero à medida que o tamanho da amostra aumenta. Na inferência não paramétrica, vimos que o alisamento ótimo corresponde ao balanceamento do viés e do desvio padrão. O segundo termo não desaparece mesmo com amostras grandes.

A presença desse segundo termo não invasivo introduz um viés no limite Normal. O resultado é que o intervalo de confiança não será centrado em torno da verdadeira função r devido ao viés de suavização $\overline{r}_n(x) - r(x)$.

Existem várias coisas que podemos fazer sobre esse problema. A primeira é: viva com isso. Em outras palavras, basta aceitar o fato de que a faixa de confiança é para $\overline{r}_n(x)$ não para r. Não há nada de errado com isso, contanto que tenhamos cuidado quando reportarmos os resultados para deixar claro que as inferências são para $\overline{r}_n$ não r). Uma segunda abordagem é estimar a função de viés $\overline{r}_n(x) - r(x)$. Isso é difícil de fazer. De fato, o termo principal do viés é $r''(x)$ e a estimativa da segunda derivada de r é muito mais difícil do que a estimativa de r. Isso requer a introdução de condições de suavidade extras que, em seguida, colocam em questão o estimador original que não usou essa suavidade extra. Isso tem uma certa circularidade desagradável para ele. Uma terceira abordagem é a falta de bom senso. Se suavizarmos menos do que a quantidade ideal, a tendência diminuirá assintoticamente em relação à variação. Infelizmente, não parece haver uma regra prática simples para escolher a quantidade certa de sub alisamento.Adotaremos a primeira abordagem e nos contentaremos em encontrar uma banda de confiança para $\overline{r}_n(x)$.

Construindo Bandas de Confiança

Assuma que $\hat{r}_n(x)$ é um suavizador linear, de modo que $\hat{r}_n(x) = \sum_{i=1}^n l_i(x) Y_i$. Então,

$$\overline{r}_n(x) = \mathrm{E}\big(\hat{r}_n(x)\big) = \sum_{i=1}^n l_i(x) r(x_i).$$

Por enquanto, vamos assumir que $\sigma^2(x) = \sigma^2 = \mathrm{Var}(\epsilon_i)$ seja constante. Então,

$$\mathrm{Var}\big(\hat{r}_n(x)\big) = \sigma^2 ||l(x)||^2.$$

Vamos considerar uma banda de confiança para $\overline{r}_n(x)$ da forma

$$\mathrm{I}(x) = \Big(\hat{r}_n(x) - c\widehat{\sigma}||l||,\, \hat{r}_n(x) + c\widehat{\sigma}||l||\Big),$$

para algum $c > 0$ e $a \leq x \leq b$.

Seguiremos a abordagem em Sun and Loader (1994). Primeiro suponha que σ é conhecido. Então,

$$P\Big(\overline{r}(x) \notin \mathrm{I}(x) \text{ para algum } x \in [a,b]\Big) = P\left(\max_{x\in[a,b]} \frac{|\hat{r}_n(x) - \overline{r}_n(x)|}{\sigma||l||} > c\right) =$$

$$= P\left(\max_{x\in[a,b]} \frac{|\sum_{i=1}^n \epsilon_i l_i(x)|}{\sigma||l||} > c\right) = P\left(\max_{x\in[a,b]} |W(x)| > c\right),$$

onde $W(x) = \sum_{i=1}^n Z_i T_i(x)$, $Z_i = \epsilon_i/\sigma \sim N(0,1)$ e $T_i(x) = l_i(x)/||l(x)||$. Agora, $W(x)$ é um processo gaussiano. Para encontrar c, precisamos ser capazes de calcular a distribuição do máximo de um processo gaussiano. Felizmente, esse é um problema bem estudado. Em particular, Sun and Loader (1994) mostraram que, a chamada de fórmula tubo, é dada por

$$P\left(\max_x \Big|\sum_{i=1}^n Z_i T_i(x)\Big| > c\right) \approx 2\big(1-\Phi(c)\big) + \frac{\kappa_0}{\pi} e^{-c^2/2},$$

para c grande, onde

$$\kappa_0 = \int_a^b ||\,T'(x)\,||\mathrm{d}x,$$

$T'(x) = (T_1'(x), \cdots, T_n'(x))$ e $T_i'(x) = \partial T_i(x)/\partial x$. Se escolhermos c como o valor que resolve a equação

$$2\big(1-\Phi(c)\big) + \frac{\kappa_0}{\pi} e^{-c^2/2} = \alpha,$$

então obtemos a banda de confiança simultánea desejada. Caso σ seja desconhecido, usamos uma estimativa $\widehat{\sigma}$. Sun and Loader sugerem substituir o lado direito da fórmula tubo com

$$P(|T_m| > c) + \frac{\kappa_0}{\pi}\left(1 + \frac{c^2}{m}\right)^{-m/2}$$

onde T_m têm distribuição $t - Student$ com $m = n - \mathrm{tr}(L)$ graus de liberdade. Para n grande, continua a ser uma aproximação adequada.

Agors suponhamos que $\sigma(x)$ seja uma função de x. Então,

$$\mathrm{Var}\big(\hat{r}_n(x)\big) = \sum_{i=1}^n \sigma^2(x_i) l_i^2(x)\cdot$$

Neste caso escolhemos que

$$I(x) = \hat{r}_n(x) \pm c\, s(x),$$

sendo que

$$s(x) = \sqrt{\sum_{i=1}^n \widehat{\sigma}^2(x_i) l_i^2(x)},$$

onde $\widehat{\sigma}(x)$ é um estimador de $\sigma(x)$ e c é a constanre definida acima. Caso $\widehat{\sigma}(x)$ varie lentamente com x, então $\sigma_(x_i) \approx \sigma(x)$ para aqueles i tais que $l_i(x)$ é grande e então

$$s(x) \approx \widehat{\sigma}(x)||l(x)||\cdot$$

Assim, uma banda de confiança aproximada é

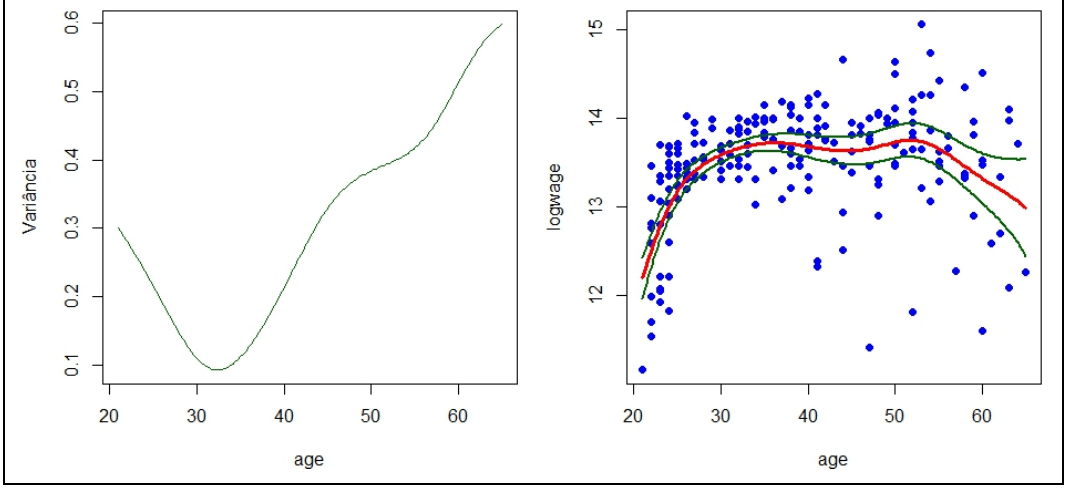
$$I(x) = \hat{r}_n(x) \pm c\widehat{\sigma}(x)||l(x)||\cdot$$

Para mais detalhes sobre esses métodos, veja Faraway and Sun (1995). Esta será a banda de confiança que utilizaremos de preferência.

Exemplo IV.11. Utilizaremos os dados do Exemplo IV.1. Para o modelo no objeto R **model.npar05** vamos mostrar a variância estimada e a banda de confiança. Existe uma saída padrão do comando **locpol**, mas não gostamos do resultado. Então, utilizamos nossa própria função. Mostramos o resultado para o modelo ajustado utilizando polinômios locais.

```
> plot(model.npar05$lpFit[, model.npar05$X], model.npar05$lpFit$var, type = "l",
      xlab = model.npar05$X, ylab = "Variância", col = "darkgreen")
> par(mfrow = c(1,1), mar = c(4, 4, 1, 1))
> plot(logwage ~ age, data = cps71, col = "blue", pch=19)
> lines(model.npar05$lpFit[,1], model.npar05$lpFit[,2], col = "red", lwd = 3)
> Intervalo.Conf = function (x)
 {
   dev <- sqrt(x$CIwidth * x$lpFit$var/x$lpFit$xDen)
   points(x$lpFit[, x$X], x$lpFit[, x$Y] + 2 * dev, type = "l", lwd = 2,
        col = "darkgreen")
   points(x$lpFit[, x$X], x$lpFit[, x$Y] - 2 * dev, type = "l", lwd = 2,
```

```
        col = "darkgreen")
 }
> Intervalo.Conf(model.npar05)
```



Agora o caso do modelo em **model.npar06**, nesta situação foram utilizado splines. Vamos investigar uma outra forma de encontrarmos faixas de confiança para o spline. Desta vez, precisamos fazer o bootstrap, e podemos fazê-lo reamostrando os resíduos ou reamostrando os todos os dados. Vamos escolher a última abordagem, que pressupõe menos sobre os dados. Precisamos de um simulador.

```
> simulador = function(dados){
   n = nrow(dados)
   resample.rows = sample(1:n, size = n, replace = TRUE)
   return(dados[resample.rows,])
 }
```

Tratamos assim os pontos no gráfico de dispersão como uma população completa e, em seguida, extraimos uma amostra deles, com substituição, tão grande quanto o original. Também precisamos de um estimador. O que queremos fazer é obter um monte de curvas de spline, uma em cada conjunto de dados simulado. Mas, como os valores da variável de entrada mudam de uma simulação para outra, para tornar tudo comparável, avaliamos cada função spline em uma grade fixa de pontos, que percorre o intervalo dos dados.

```
> spline.estimator = function(dados, m=300){
   # Ajuste por spline aos dados, com validação cruzada para selecionar lambda
   ajuste = smooth.spline(x=dados[,1], y=dados[,2], cv=TRUE)
   # Configurando uma grade de pontos com espaçamento uniforme para avaliar o spline
   eval.grid = seq(from=min(dados[,1]), to=max(dados[,1]), length.out=m)
   # Um pouco ineficiente para redefinir a mesma grade toda vez que chamamos isso,
   # mas não uma grande sobrecarga
   # Fazemos a previsão e retornamos os valores previstos
   return(predict(ajuste, x=eval.grid)$y) # Somente queremos os valores previstos
 }
```

Isso define o número de pontos de avaliação como 300, que é grande o suficiente para dar curvas visualmente suaves, mas não tão grandes a ponto de serem computacionalmente incômodas. Agora juntemo-os para obter bandas de confiança.

```
> spline.cis = function(dados, B, alpha, m=300) {
   spline.main = spline.estimator(dados, m=m)
   # Desenhe B amostras de bootstrap, ajustemos o spline para cada
   spline.boots = replicate(B, spline.estimator(simulador(dados),m=m))
   # O resultado tem m linhas e B colunas
   cis.lower = 2*spline.main - apply(spline.boots,1,quantile,probs=1-alpha/2)
   cis.upper = 2*spline.main - apply(spline.boots,1,quantile,probs=alpha/2)
   return(list(main.curve = spline.main,lower.ci=cis.lower,upper.ci=cis.upper,
               x=seq(from=min(dados[,1]),to=max(dados[,1]),length.out=m)))
 }
```

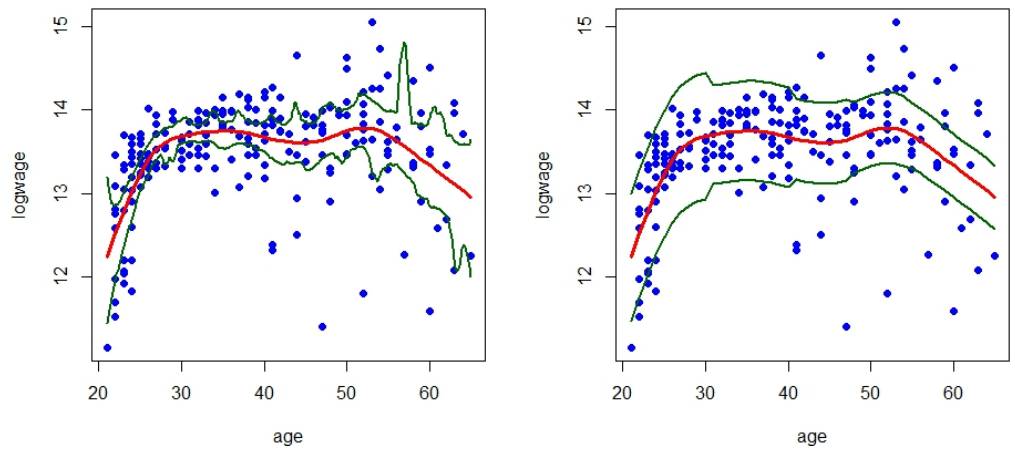
O valor de retorno aqui é uma lista que inclui a curva ajustada original, os limites de confiança inferior e superior e os pontos nos quais todas as funções foram avaliadas.

Exemplo IV.12. Continuando o Exemplo IV.11, mostramos agora o rssultado do ajuste utilizando spline.

```
> dados = data.frame(age = cps71$age, logwaget = cps71$logwage)
> head(cps71)
  logwage age
1 11.1563  21
2 12.8131  22
3 13.0960  22
4 11.6952  22
5 11.5327  22
6 12.7657  22
> head(dados)
  age logwaget
1  21  11.1563
2  22  12.8131
3  22  13.0960
4  22  11.6952
5  22  11.5327
6  22  12.7657
```

Construímos a nova base de dados **dados** porque nas funções **spline.estimaotr** e **spline.cis** assumimos a primeira coluna como **x** e a segunda coluna da base dados como **y**. Observe que na base de dados original estas variáveis estavam trocadas de posição.

```
> dados.cis = spline.cis(dados, B=1000, alpha=0.05)
> par(mfrow = c(1,1), mar = c(4, 5, 2, 1))
> plot(logwage ~ age, data = cps71, col = "blue", pch=19)
> lines(model.npar06, col = "red", lwd = 3)
> lines(x=dados.cis$x, y=dados.cis$lower.ci, lwd = 2, col = "darkgreen")
> lines(x=dados.cis$x, y=dados.cis$upper.ci, lwd = 2, col = "darkgreen")
```



Na Figura acima mostramos dois gráficos, à esquerda os limites de confiança de 95% resultantes, com base nas B = 1000 replicações bootstrap. Estas bandas são claramente assimétricos da mesma forma que a curva se ajusta a todos os dados, mas perceba como eles são largas e como eles se ampliam à medida que vamos do centro dos dados em qualquer direção. No caso do gráfico à direita utilizamos o resultado das bandas de confiança mostradas nesta seção e utilizamos o resultado guardado em **sigma06**, para isso utilizamos os comandos listados abaixo.

```
> norma = sum(model.npar06$lev^2)
> par(mfrow = c(1,1), mar = c(4, 5, 2, 1))
> plot(logwage ~ age, data = cps71, col = "blue", pch=19)
> lines(model.npar06, col = "red", lwd = 3)
```

```
> lines(model.npar06$x, model.npar06$y +
                      2*sqrt(sigma06[model.npar06$x])*norma, lwd = 2, col = "darkgreen")
> lines(model.npar06$x, model.npar06$y -
                      2*sqrt(sigma06[model.npar06$x])*norma, lwd = 2, col = "darkgreen")
```

IV.7 Cobertura média

Pode-se argumentar que exigir que as bandas de confiança cubram a função em todos os x é muito rigoroso. Wahba (1983), Nychka (1988) e Cummins et al. (2001) introduziram um tipo diferente de cobertura a que nos referimos como cobertura média. Aqui vamos discutir um método para construir bandas de cobertura média com base na ideia em Juditsky e Lambert-Lacroix (2003).

Suponha que estamos estimando $r(x)$ no intervalo $[0,1]$. Defina a cobertura média de uma banda (l,u) por

$$\mathcal{C} = \int_0^1 P\big(r(x) \in [l(x), u(x)]\big)\mathrm{d}x.$$

Vamos construir bolas de confiança para r, os quais são conjuntos $\mathcal{B}_n(\alpha)$, da forma

$$\mathcal{B}_n(\alpha) = \big\{r : ||\hat{r}_n - r|| \leq s_n(\alpha)\big\},$$

tais que

$$P\big(r \in \mathcal{B}_n(\alpha)\big) \geq 1-\alpha.$$

Dada uma bola de confiança, sejam

$$l(x) = \hat{r}_n(x) - s_n(\alpha/2)\sqrt{\frac{2}{\alpha}}, \qquad u(x) = \hat{r}_n(x) + s_n(\alpha/2)\sqrt{\frac{2}{\alpha}}.$$

Nós agora vamos mostrar que essas bandas têm uma cobertura média de pelo menos $1-\alpha$. Primeiro, observe que $\mathcal{C} = P\big(r(U) \in [l(U), u(U)]\big)$ onde $U \sim U(0,1)$ é independente dos dados. Seja A o evento que $r \in \mathcal{B}_n(\alpha/2)$. No evento A, $||\hat{r}_n - r|| \leq s_n(\alpha)$. Escrevendo s_n para $s_n(\alpha/2)$ temos,

$$\begin{aligned}
1-\mathcal{C} &= P\Big(r(U) \notin [l(U), u(U))]\Big) = P\Big(|\hat{r}_n(U) - r(U)| > s_n\sqrt{\tfrac{2}{\alpha}}\Big) \\
&= P\Big(|\hat{r}_n(U) - r(U)| > s_n\sqrt{\tfrac{2}{\alpha}}, A\Big) + P\Big(|\hat{r}_n(U) - r(U)| > s_n\sqrt{\tfrac{2}{\alpha}}, A^c\Big) \\
&\leq P\Big(|\hat{r}_n(U) - r(U)| > s_n\sqrt{\tfrac{2}{\alpha}}, A\Big) + P(A^c) \\
&\leq \frac{\mathrm{E}\big(\mathrm{I}_A||\hat{r}_n - r||^2\big)}{s_n^2\dfrac{2}{\alpha}} + \frac{\alpha}{2} \leq \frac{s_n^2}{s_n^2\dfrac{2}{\alpha}} + \frac{\alpha}{2} \leq \alpha.
\end{aligned}$$

IV.8 Regressão múltipla

Suponhamos agora que a covariável seja de dimensão d, ou seja,

$$x_i = (x_{i1}, x_{i2}, \cdots, x_{id})^\top.$$

O modelo de regressão assume a forma

$$Y = r(x_1, x_2, \cdots, x_d) + \epsilon.$$

Em princípio, todos os métodos que discutimos passam facilmente para este caso. Infelizmente, o risco de um estimador de regressão não paramétrico aumenta rapidamente com a dimensão d. Em um problema unidimensional, a taxa ótima de convergência de um estimador não paramétrico é $n^{-4/5}$ se r é assumido como tendo uma segunda derivada integrável. Em d dimensões, a taxa ótima de convergência é $n^{-4/(4+d)}$. Assim, o tamanho da amostra m requerido para um problema d-dimensional ter a mesma precisão que um tamanho de amostra n em um problema unidimensional é $m \propto n^{cd}$ onde $c = (4+d)/(5d) > 0$. Isto implica que para manter um determinado grau de precisão de um estimador, o tamanho da amostra deve aumentar exponencialmente com a dimensão d. Em outras palavras, as bandas de confiança ficam muito grandes à medida que a dimensão d aumenta. No entanto, vamos continuar e ver como podemos estimar a função de regressão.

IV.9.1 Regressão local

Considere a regressão linear local. A função kernel K é agora uma função de d variáveis. Dada uma matriz $d \times d$ de largura de banda definida positiva não-singular H, definimos

$$K_H \;=\; \frac{1}{|H|^{1/2}} K\big(H^{1/2}x\big)\cdot$$

Muitas vezes, a escala de cada covariada tem a mesma média e variância e então usamos o kernel

$$h^{-k}K\big(||x||/h\big),$$

onde K é qualquer kernel unidimensional. Então há um único parâmetro de largura de banda h. Num valor alvo $x = (x_1, \cdots, x_d)^\top$ a soma dos quadrados local é dada por

$$\sum_{i=1}^{n} \omega_i(x)\Big(Y_i \;-\; a_0 \sum_{j=1}^{d} a_j(x_{ij} - x_j)\Big)^2$$

onde

$$\omega_i(x) \;=\; K(||x_i - x||/h)\cdot$$

O estimador é

$$\hat{r}_n(x) \;=\; \widehat{a}_0,$$

onde $\widehat{a} = (\widehat{a}_0, \cdots, \widehat{a}_d)^\top$ é o valor de $a = (a_0, \cdots, a_d)^\top$ que minimiza a soma de quadrados ponderads. A solução $\widehat{a}$ é

$$\widehat{a} \;=\; (X_x^\top W_x X_x)^{-1} X_x^\top W_x Y$$

onde

$$X_x \;=\; \begin{pmatrix} 1 & x_{11} - x_1 & \cdots & x_{1d} - x_d \\ 1 & x_{21} - x_1 & \cdots & x_{2d} - x_d \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} - x_1 & \cdots & x_{nd} - x_d \end{pmatrix}$$

e W_x é a matriz diagonal cujo elemento (i, i) é $w_i(x)$.

As propriedades teóricas da regressão polinomial local em dimensões maiores são discutidas no seguinte teorema.

Teorema IV.12. Seja $\hat{r}_n(x)$ o estimador linear local multivariado com matriz de largura de banda H e assumimos satisfeitas as condições de regularidade:

(a) O kernel K é limitado, de suporte compacto de tal modo que $\int uu^\top K(u)\mathrm{d}u = \mu_2(K)\mathrm{I}$, onde $\mu_2(K) \neq 0$ é escalar e I é matriz identidade $d \times d$. Além disso, todos os momentos de ordem ímpar de K somem, isto é, $\int u_1^{l_1} \cdots u_d^{l_d} K(u)\mathrm{d}u = 0$ para todos os inteiros não negativos $l_1, \cdots, l_d$, de tal forma que sua soma é ímpar. Esta úútima condição é satisfeita por kernels esféricos simétricos e kernels de produtos baseados em kernels univariados simétricos.

(b) O ponto x está no suporte de f. Em x, ν é contínua, f é continuamente diferenciável e todas as derivadas de segunda ordem de r são contínuas. Além disso, $f(x) > 0$ e $\nu(x) > 0$.

(c) A sequência de matrizes de largura de banda $H^{1/2}$ é tal que $n^{-1}|H|$ e cada entrada de H tende a zero quando $n \to \infty$ com H permanecendo simétrica e definida positiva. Além disso, existe uma constante fixa L tal que a razão de seu maior e menor autovalor é no máximo L para todo n.

Suponha que x seja um ponto não-fronteira. Condicionado em $X_1, \cdots, X_n$ temos os seguintes resultados:

(*i*) O viés de $\hat{r}_n(x)$ é

$$\frac{1}{2}\mu_2(K)\mathrm{tr}\big(H\mathcal{H}\big) + o_P(\mathrm{tr}(H)),$$

onde $\mathcal{H}$ é a matriz das segundas derivadas parciais de r avaliada em x e $\mu_2(K)$ é o escalar definido pela condição de regularidade (a).

(*ii*) A variância de $\hat{r}_n(x)$ é

$$\frac{\sigma^2(x)\int K(u)^2\mathrm{d}u}{n|H|^{1/2}f(x)}\big(1 + o_P(1)\big)\cdot$$

Além disso, o viés no limite é da mesma ordem que no interior, ou seja, $O_P(\mathrm{tr}(H))$.

Demonstração. Ruppert and Wand (1994).∎

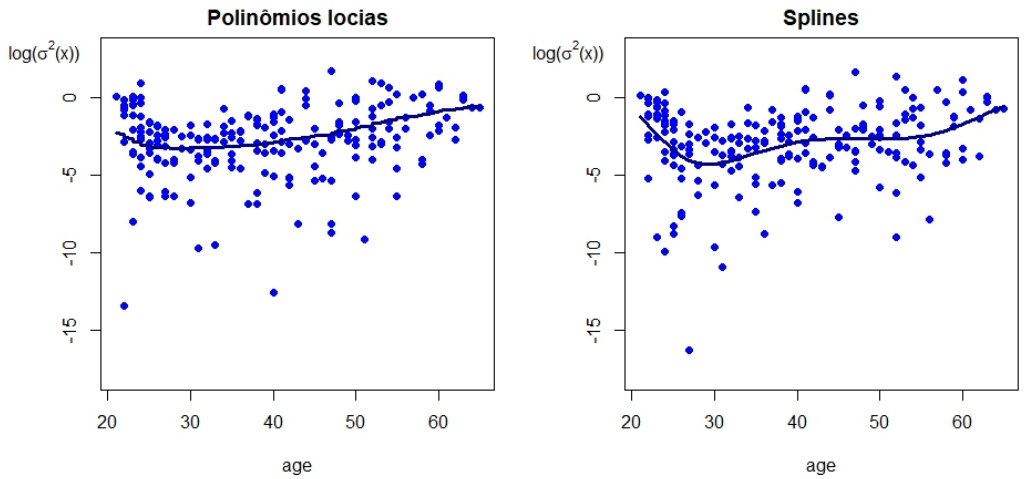
Ao estudarmos a regressão local conhecemos duas formas de avaliarmos estes estimadores: o estimador kernel Nadaraya–Watson e os polinômios locais. No Teorema IV.5 mostramos que a regressão linear local é melhor que a regressão kernel. Acontece que, pelo menos até o nosso conhecimento, os procedimentos implementados no **R** somente permitem a estimação do modelo de regressão por polinômios locais até ordem 2, ou seja, somente aceitam no máximo duas covariáveis contínuas. Nos referimos especificamente ao

pacote de funções **locpol**. Significa que, para aplicarmos modelos de regressão náo paramétrica a situações mais complexas devemos recorrer ao estimador kernel, muito bem implementado no pacote de funções **np** e descrito no artigo de Hayfield and Racine (2008).

O pacote **np** implementa uma variedade de estimadores não paramétricos e semiparamétricos baseados no estimador kernel Nadaraya–Watson Existem também procedimentos para testes de significância não paramétricos e testes consistentes de especificação de modelo. Este pacote foca nos métodos apropriados para a combinação de dados contínuos, discretos e categóricos, frequentemente encontrados em problemas aplicados. Os métodos de seleção de largura de banda orientados por dados são enfatizados, embora tenhamos cautela já que estes prodedimentos podem ser exigentes em termos computacionais.

Exemplo IV.13. Dados britânicos de corte transversal, consistindo de uma amostra aleatória extraída do Inquérito às Famílias da Família Britânica de 1995. Os agregados familiares consistem em casais casados com um chefe de família empregado entre os 25 e os 55 anos de idade. Existem 1655 observações ao nível do agregado familiar no total.

```
> library(np)
> data(Engel95)
> print(head(Engel95), digits = 4)
     food catering alcohol    fuel  motor   fares leisure logexp logwages nkids
1 0.14949  0.10460 0.00000 0.07555 0.1940 0.03446 0.16236  4.878    5.533     0
2 0.36461  0.03262 0.01533 0.09762 0.1994 0.03679 0.13521  5.094    5.371     0
3 0.21042  0.06525 0.07919 0.03367 0.2526 0.01002 0.17814  5.782    6.001     0
4 0.07869  0.11527 0.07014 0.05842 0.0000 0.24974 0.02726  5.756    5.861     0
5 0.28244  0.05508 0.07505 0.15707 0.2178 0.00000 0.03754  5.280    6.569     0
6 0.16636  0.15928 0.17427 0.02050 0.0907 0.00000 0.14901  5.822    5.879     0
```



Esta aqui é uma ilustração simples para ajudá-lo a começar com a regressão do Kernel multivariada e a plotagem atrav&eacudia;s da função de plotagem do R que chama **npplot**, no mesmo pacote np.

```
> Engel95 = Engel95[order(Engel95$logexp),]
> attach(Engel95)
> modelo.iv = npregiv(y=food, z=logexp, w=logwages, method="Landweber-Fridman")
> phihat = modelo.iv$phi
```

Calculando a regressão IV (ou seja, regressão de y em z)

```
> ghat <- npreg(food~logexp, regtype="ll")
```

Para os gráficos, restrinjimos a atenção à maior parte dos dados, isto é, para a área de plotagem, cortamos 1/4 de um por cento de cada cauda de y em z.

```
> trim <- 0.0025
> plot(logexp,food,
     ylab="Compartilhamento Orçamentário de Alimentos",
     xlab="log(Despesa total)",
```

```
     xlim=quantile(logexp,c(trim,1-trim)),
     ylim=quantile(food,c(trim,1-trim)),
     main="Regressão Não Paramétrica Kernel Instrumental",
     type="p",
     cex=.5,
     col="lightgrey")
> lines(logexp,phihat,col="blue",lwd=2,lty=2)
> lines(logexp,fitted(ghat),col="red",lwd=2,lty=4)
> legend(quantile(logexp,trim),quantile(food,1-trim),
       c(expression(paste("Não Paramétrica IV: ",hat(varphi)(logexp))),
         "Regressão Não Paramétrica : E(food | logexp)"),
       lty=c(2,4),
       col=c("blue","red"),
       lwd=c(2,2))
```

Assim, vemos que em dimensões mais altas, a regressão linear local ainda evita excessivo viés de limite e viés de design.

IV.9.2 Modelos aditivos

Interpretar e visualizar um ajuste de alta dimensão é difícil. À medida que o número de covariáveis aumenta, a carga computacional torna-se proibitiva. Ás vezes, uma abordagem mais frutífera é usar um modelo aditivo. Um modelo aditivo é um modelo da forma

$$Y = \mu + \sum_{j=1}^{d} r_j(x_i) + \epsilon,$$

onde $r_1, \cdots, r_d$ são funções suaves. Acontece que este modelo não é identificável, pois podemos adicionar qualquer constante a μ e subtrair a mesma constante de um dos r_j sem alterar a funçção de regressão. Este problema pode ser corrigido de várias maneiras, talvez o mais fácil seja definir $\widehat{\mu} = \overline{Y}$ e, em seguida, considerar os r_j como desvios de $\overline{Y}$. Neste caso, exigimos que

$$\sum_{i=1}^{n} r_j(x_i) = 0,$$

para cada j.

O modelo aditivo claramente não é tão geral quanto estimar $r(x_1, \cdots, x_d)$, mas é muito mais simples de ser computado e interpretado e, portanto, é geralmente um bom ponto de partida. Apresentamos um algoritmo simples para transformar qualquer regressão unidimensional mais suave em um método para ajustar modelos aditivos, chamado de backfitting ou adaptação.

ALGORITMO DE ADAPTAÇÃO

Passo 1: Inicialização: seja $\widehat{\mu} = \overline{Y}$, e considere valores inciais para $\hat{r}_1, \cdots, \hat{r}_d$.
Pssso 2: Iterar até a convergência: para $j = 1, \cdots, d$

1. Calcular

$$\widetilde{Y}_i = Y_i - \widehat{\mu} - \sum_{k \neq j} r_k(x_i),$$

   para $i = 1, \cdots, n$.
2. Aplicar o alisamento a $\widetilde{Y}_i$ no x_j para obter $\hat{r}_j$.
3. Atribuir $\hat{r}_j$ igual a

$$\hat{r}_j(x) - \frac{1}{n} \sum_{i=1}^{n} \hat{r}_j(x_i).$$

Uma forma de implementar este algoritmo está disponível para modelos de regressão linear múltipla aplicando splines a somente uma covarável, isso não significa que poderemos aplicar splines a somente uma covariável por vez. Significa que aplicaremos splines undimensionais a uma ou várias covariáveis isoladamente. Veja no seguinte exemplo a forma de utilizarmos o comando **ns** dentro do pacote de funções **splines**..

Exemplo IV.14.

Os dados em **mtcars**, disponíveis na base do R, foram extraídos da revista Motor Trend nos Estados Unidos de 1974 e compreendem o consumo de combustível e 10 aspectos do projeto e desempenho de 32 automóveis modelos de 1973 a 1974.

Interéssa-nos como reposta mpg ou milhas/galão em dólares americanos, como covariáveis temos gear ou a engrenagem contando o número de marchas para a frente, a qual é discreta ou um fator; wt o peso em 1000 libras e hp a potência bruta do motor.

```
> library(splines)
> mult.spline1 = lm(mpg ~ factor(gear) + ns(wt) + hp, data = mtcars)
> summary(mult.spline1)

Call:
lm(formula = mpg ~ factor(gear) + ns(wt) + hp, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max
-3.3025 -1.9307 -0.3722  1.0243  5.9784

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    29.97256    1.84066  16.284 1.74e-15 ***
factor(gear)4   1.26490    1.34084   0.943   0.3539
factor(gear)5   1.87356    1.86662   1.004   0.3244
ns(wt)        -15.79711    4.28188  -3.689   0.0010 **
hp             -0.03497    0.01260  -2.775   0.0099 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.619 on 27 degrees of freedom
Multiple R-squared:  0.8356,    Adjusted R-squared:  0.8112
F-statistic:  34.3 on 4 and 27 DF,  p-value: 3.196e-10
```

IV.9.2.1 Splines bidimensionais

Nesta abordagem precisamos definir splines em dimensões maiores. Para $d = 2$, minimizamos

$$\sum_i \left(Y_i - \hat{r}_n(x_{i1}, x_{i2})\right)^2 + \lambda J(r),$$

onde

$$J(r) = \int\int \left[\left(\frac{\partial^2 r(x)}{\partial x_1^2}\right) + 2\left(\frac{\partial^2 r(x)}{\partial x_1 \partial x_2}\right) + \left(\frac{\partial^2 r(x)}{\partial x_2^2}\right)\right] \mathrm{d}x_i \mathrm{d}x_2.$$

O minimizedor $\hat{r}_n$ é chamado spline de placa fina. É difícil descrever e ainda mais difícil, mas certamente não impossível, de ajustar. Veja Green and Silverman (1994) para mais detalhes.

Exemplo IV.15.(Continuação do Exemplo IV.14)

Utilizando os mesmos dados, podemos considerar interações entre as variáveis **wt** e **hp**, agora do tipo contínuas e aplicados aplines undimensionais a cada uma separadamente.

```
> library(splines)
> mult.spline2 = lm(mpg ~ factor(gear) + ns(wt)*ns(hp), data = mtcars)
> summary(mult.spline2)

Call:
lm(formula = mpg ~ factor(gear) + ns(wt) * ns(hp), data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max
-2.7324 -1.7140 -0.5176  1.4920  4.5603

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)     32.103      1.758  18.266 2.36e-16 ***
factor(gear)4    0.829      1.118   0.741  0.46505
factor(gear)5    1.816      1.548   1.174  0.25114
ns(wt)         -29.871      5.245  -5.695 5.43e-06 ***
ns(hp)         -28.974      5.867  -4.938 3.95e-05 ***
ns(wt):ns(hp)   47.236     12.961   3.645  0.00117 **
---
```

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.171 on 26 degrees of freedom
Multiple R-squared:  0.8912,    Adjusted R-squared:  0.8702
F-statistic: 42.58 on 5 and 26 DF,  p-value: 1.035e-11
```

Ou podemos utilizar o comando **s** no pacote **mgcv** com o qual podemos ajutar splines bidimensionais.

```
> library(mgcv)
> mult.spline2 = gam(mpg ~ factor(gear) + s(wt,hp), data = mtcars)
> summary(mult.spline2)

Family: gaussian
Link function: identity

Formula:
mpg ~ factor(gear) + s(wt, hp)

Parametric coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    19.6247     0.7288  26.926   <2e-16 ***
factor(gear)4   0.4072     1.2821   0.318    0.754
factor(gear)5   2.0046     1.6312   1.229    0.231
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
           edf Ref.df     F  p-value
s(wt,hp) 5.059  6.299 16.79 4.79e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.868   Deviance explained = 89.8%
GCV = 6.4313  Scale est. = 4.8116    n = 32
```

Perceba que, desta última forma, tivemos um ganho real no ajuste, o R^2 ajustado é de 0.898. Lembremos que estamos trabalhando com modelos de regressão, com isso a melhor maneirra de descubrirmos se estamos realizaddo um ajuste adequado é analisando os resíduos. Para isso, qualquer seja o modelos escolhido, sempre é possível utilizarmos o comando **residuals** e mostrar gráficos e testes adequados.

IV.9.3 Prospeção de Projeção

Vamos supor que o vetor de variáveis explicativas $X = (x_1, \cdots, x_p)$ seja de alta dimensão. O modelo aditivo

$$Y = \mu + \sum_{j=1}^{p} f_j(x_j) + \epsilon$$

pode ser muito flexível, pois permite alguns graus de liberdade por cada covariável, mas não cobre o efeito de interações entre as variáveis independentes, explicativas ou covariáveis. A regressão por **prospeção de projeção**, proposto por Friedman and Stuetzle (1981), aplica um modelo aditivo às variáveis projetadas. A ideia é aproximar a função de regressão $r(x_1, x_2, \cdots, x_p)$ com uma função da forma

$$\mu + \sum_{m=1}^{M} r_m(z_m),$$

onde

$$z_m = \alpha_m^\top X$$

e cada α_m é um vetor unitário, ou seja, de comprimento um, para $m = 1, \cdots, M$.

Assim, ele usa um modelo aditivo nas variáveis preditoras que são formadas projetando X em M direções cuidadosamente escolhidas. Mesmo sendo M muito grande tais modelos podem aproximar, uniformemente em conjuntos compactos e em muitos outros sentidos, funções contínuas arbitrárias de X. Os termos $r_m(z_m)$ são chamados de funções de crista, uma vez que são

constantes em todas as direções, exceto uma.

Note que cada z_m é a projeção de X num subespaço. O vetor de direção α é escolhido em cada etapa para minimizar a fração de variância inexplicada. Em mais detalhes, denotemos por $S(\cdot)$ o mapeamento que produz n valores ajustados de algun método de alisamento, dados os Y_i e alguns valores das covariáveis unidimensionais $z_1, \cdots, z_n$. Seja também $\widehat{\mu} = \overline{Y}$ e substitua Y_i por $Y_i - \overline{Y}$. Assim, os Y_i agora têm média 0. Da mesma forma, modificamos as covariáveis para que cada uma tenha a mesma variância. Então siga os seguintes passos:

Passo 1: Inicialize os resíduos $\hat{\epsilon}_i = Y_i, i = 1, \cdots, n$ e seja $m = 0$.
Pssso 2: Encontrar a direção, vetor unitário, α que maximiza

$$I(\alpha) \, = \, 1 - \frac{\sum_{i=1}^{n} \left(\hat{\epsilon}_i - S(\alpha^\top x_i)\right)^2}{\sum_{i=1}^{n} \hat{\epsilon}_i^2},$$

e seja $z_{mi} = \alpha^\top x_i$, $\hat{r}_m(z_{mi}) = S(z_{mi})$.
Pssso 3: Defina $m = m + 1$ e atualize os resíduos:

$$\hat{\epsilon}_i - \hat{r}_m(z_{mi}) \, \rightarrow \, \hat{\epsilon}_i.$$

Se $m = M$ paramos, caso contrário, voltamos para o **Passo 2**.

Exemplo IV.16. A função **ppr** (Projection Pursuit Regression) ajusta $\mu + \sum_{m=1}^{M} r_m(\alpha_m^\top X)$ por mínimos quadrados e restringe os vetores α_m a serem de comprimento unitário. Primeiramente, ajusta os termos M_{max} (max.terms) sequencialmente, depois remove de volta a M (nterms) em cada estágio, eliminando o termo menos eficiente e o reajusta. A função retorna a proporção da variância explicada por todos os ajustes de $M, \cdots, M_{max}$.

```
> library(np)
> data(wage1)
> modelo.ppr1 = ppr(lwage ~ female+educ+exper, data = wage1, nterms = 2, max.terms = 5)
> R2(wage1$lwage, fitted.values(modelo.ppr1))
[1] 0.447875
> modelo.ppr2 = ppr(lwage ~ female+educ+exper, data = wage1, nterms = 2, max.terms = 6)
> R2(wage1$lwage, fitted.values(modelo.ppr2))
[1] 0.45218
> modelo.ppr3 = ppr(lwage ~ female+educ+exper, data = wage1, nterms = 2, max.terms = 7)
> R2(wage1$lwage, fitted.values(modelo.ppr3))
[1] 0.4691726
> modelo.ppr4 = ppr(lwage ~ female+educ+exper, data = wage1, nterms = 2, max.terms = 8)
> R2(wage1$lwage, fitted.values(modelo.ppr4))
[1] 0.4697552
> modelo.ppr5 = ppr(lwage ~ female+educ+exper, data = wage1, nterms = 2, max.terms = 9)
> R2(wage1$lwage, fitted.values(modelo.ppr5))
[1] 0.46254
```

O maior valor de R^2 foi encontrado com o modelo no objeto **modelo.ppr4**. Mostramos o resultado.

```
> summary(modelo.ppr4)
Call:
ppr(formula = lwage ~ female + educ + exper, data = wage1, nterms = 2,
    max.terms = 8)

Goodness of fit:
 2 terms  3 terms  4 terms  5 terms  6 terms  7 terms  8 terms
78.65109 82.16414 71.81014 73.06259 70.76169 70.57399 72.15356

Projection direction vectors ('alpha'):
             term 1      term 2
femaleFemale -0.96783665 -0.58241848
femaleMale   -0.25074206 -0.80764297
educ         -0.01054856  0.08969389
exper        -0.01758883  0.02136698

Coefficients of ridge terms ('beta'):
   term 1    term 2
0.3836857 0.3625912
```

As informações adicionadas são os vetores de direção α_m e os coeficientes β_{im} em

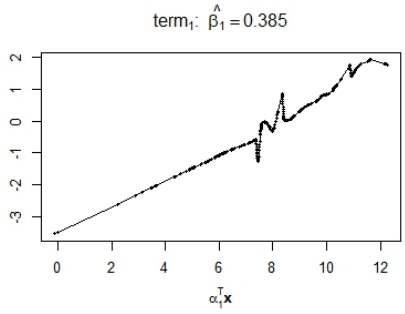
$$Y_i = \mu_i + \sum_{m=1}^{M} \beta_{im} r_m(\alpha_m^\top X) + \epsilon.$$

Note que esta é uma extensão do modelo de regressão por prospeção de projeção para múltiplas respostas e, então, separamos os escalonamentos das funções suaves r_m, que são escalonadas para ter média zero e variância unitária sobre as projeções do conjunto de dados.

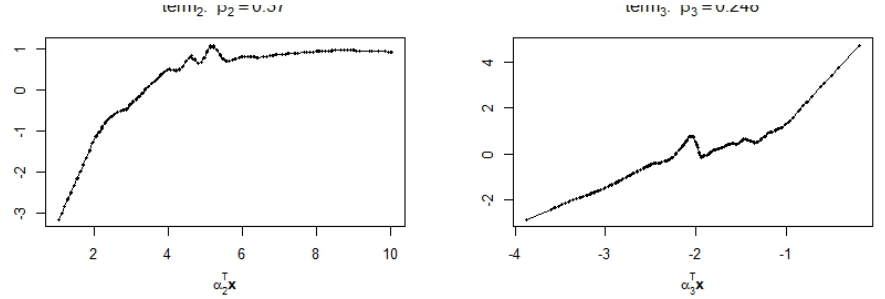
O algoritmo primeiro acrescenta os M_{max} termos (max.terms) um de cada vez; ele usará menos se não conseguir encontrar um termo para adicionar que faça diferença suficiente. Em seguida, ele remove o termo menos importante em cada etapa até que os M (nterms) termos sejam deixados.

Podemos perceber que o valor de M é arbitrário, por isso acompanhamos o ajuste de cada modelo com o valor correspondente do valor do R^2. Por isso escolhemos o modelo em **modelo.ppr4**, mas percebemos que o maior valor de variância explicada acontece quando a quantidade de termos é $M = 3$.

```
> modelo.ppr11 = ppr(lwage ~ female+educ+exper, data = wage1, nterms = 3, max.terms = 5)
> R2(wage1$lwage, fitted.values(modelo.ppr11))
[1] 0.4752204
> modelo.ppr21 = ppr(lwage ~ female+educ+exper, data = wage1, nterms = 3, max.terms = 6)
> R2(wage1$lwage, fitted.values(modelo.ppr21))
[1] 0.4721492
> modelo.ppr31 = ppr(lwage ~ female+educ+exper, data = wage1, nterms = 3, max.terms = 7)
> R2(wage1$lwage, fitted.values(modelo.ppr31))
[1] 0.4764152
> modelo.ppr41 = ppr(lwage ~ female+educ+exper, data = wage1, nterms = 3, max.terms = 8)
> R2(wage1$lwage, fitted.values(modelo.ppr41))
[1] 0.4460973
> modelo.ppr51 = ppr(lwage ~ female+educ+exper, data = wage1, nterms = 3, max.terms = 9)
> R2(wage1$lwage, fitted.values(modelo.ppr51))
[1] 0.4861771
> summary(modelo.ppr51)
Call:
ppr(formula = lwage ~ female + educ + exper, data = wage1, nterms = 3,
    max.terms = 9)

Goodness of fit:
 3 terms  4 terms  5 terms  6 terms  7 terms  8 terms  9 terms
76.30390 74.10947 73.00246 71.62785 70.10163 70.17658 68.77666

Projection direction vectors ('alpha'):
             term 1      term 2      term 3
femaleFemale -0.22908816  0.98099440  0.14632600
femaleMale    0.74008895  0.05253875  0.97618737
educ          0.63226754  0.08195210 -0.15525979
exper         0.00497110  0.16784969 -0.03925963

Coefficients of ridge terms ('beta'):
   term 1    term 2    term 3
0.3850895 0.3699895 0.2478915
```



term2: $\hat{\beta}_2 = 0.37$ term3: $\hat{\beta}_3 = 0.248$




```
> Preditos = predict(modelo.ppr51)
> par(mfrow=c(2,2))
> plot(wage1$lwage, Preditos)
> plot(modelo.ppr51)
```

A ideia da **prospeção de projeção** não é nova. A interpretação de dados de alta dimensão através do uso de projeções de dimensões inferiores bem escolhidas é um procedimento padrão na análise de dados multivariados. A escolha de uma projeção é geralmente guiada por uma figura de mérito apropriada. Se o objetivo é preservar as distâncias entre pontos tão bem quanto possível, a figura de mérito apropriada é a variância dos dados projetados, levando à projeção no maior componente principal. Se o propósito é separar duas amostras gaussianas com matrizes de covariância iguais, a figura de mérito é a taxa de erro de uma regra de classificação unidimensional na projeção, levando à Análise Linear Discriminante. Em ambos os casos, a figura de mérito é especialmente simples e a solução pode ser encontrada pela álgebra linear.

A regressão por **prospeção de projeção** segue uma receita semelhante. Ela constrói um modelo da superfície de regressão com base em projeções dos dados nos planos abrangidos pela resposta Y e uma combinação linear αX das preditoras. Aqui, a figura de mérito para uma projeção é a fração de variância explicada por um alisamento de Y versus αX. A estrutura é removida formando os resíduos do alisamento e substituindo-os pela resposta. O modelo em cada iteração é a soma das suavizações que foram subtraídas anteriormente e, portanto, incorpora a estrutura até agora encontrada.

IV.9.4 Árvores de regressão

Uma **árvore de regressão** é um modelo da forma

$$r(x) = \sum_{m=1}^{M} c_m I(x \in R_m),$$

onde $c_1, \cdots, c_M$ são constantes e $R_1, \cdots, R_M$ são retângulos disjuntos que dividem o espaço das covariáveis. Os modelos de árvores foram introduzidos por Morgan and Sonquist (1963) e Breiman et al. (1984). O modelo é ajustado de uma maneira recursiva que pode ser representada como uma árvore; daí o nome. Nossa descrição segue Hastie et al. (2001).

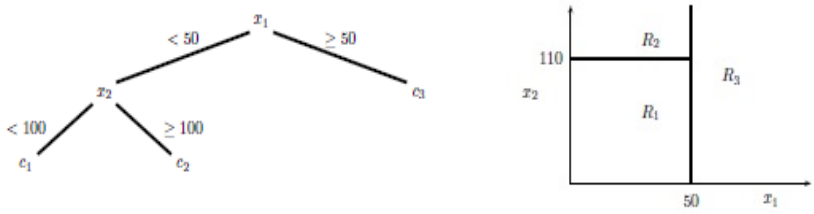
Denote um valor genérico das covariáveis por $x = (x_1, \cdots, x_j, \cdots, x_d)$. A covariável para a i-ésima observação é $x_i = (x_{i1}, \cdots, x_{ij}, \cdots, x_{id})$. Dada a covariável j e um ponto de divisão s definimos os retângulos

$$R_1 = R_1(j, s) = \{x : x_j \leq s\} \qquad e \qquad R_2 = R_2(j, s) = \{x : x_j > s\}$$

onde, nesta expressão, x_j se refere à j-ésima covariável não a observação j.

Escolhemos então c_1 como a média de todos o Y_i tais que $x_i \in R_1$ e c_2 como a média de todos os Y_i de tal forma que $x_i \in R_2$. Observe que c_1 e c_2 minimizam as somas de quadrados $\sum_{x_i \in R_1}(Y_i - c_1)^2$ e $\sum_{x_i \in R_2}(Y_i - c_2)^2$. A escolha de qual covariável x_j dividir e qual o ponto de divisão s para usar é baseado na minimização das somas de quadrados residuais. O processo de divisão é repetido em cada retângulo R_1 e R_2.

A figura mostra um exemplo simples de uma árvore de regressão; também são mostrados os retângulos correspondentes. A estimativa da função é constante sobre os retângulos.



Exemplo de uma árvore de regressão para duas covariáveis x_1 e x_2. A estimativa da função é

$$r(x) = c_1 I(x \in R_1) + c_2 I(x \in R_2) + c_3 I(x \in R_3)$$

onde R_1, R_2 e R_3 são os retângulos mostrados ao lado.

Geralmente, uma árvore cresce muito e, em seguida, a árvore é podada para formar uma subárvore, colapsando as regiões juntas. O tamanho da árvore é um parâmetro de ajuste escolhido da seguinte maneira. Seja N_m o número de pontos em um retângulo R_m de uma subárvore T e defina

$$c_m = \frac{1}{N_m} \sum_{x_i \in R_m} Y_i \qquad e \qquad Q_m(T) = \frac{1}{N_m} \sum_{x_i \in R_m} (Y_i - c_m)^2.$$

Definamos a complexidade de T por

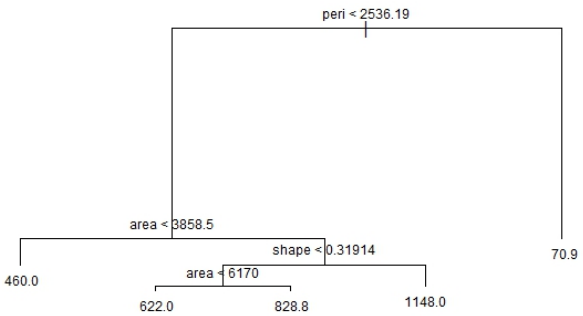
$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha |T|,$$

onde $\alpha > 0$ e $|T|$ é o número de nós terminais da árvore. Seja T_α a menor subárvore que minimize o C_α. O valor $\widehat{\alpha}$ de α pode ser escolhido por validação cruzada. A estimativa final é baseada na árvore $T_{\widehat{\alpha}}$.

Exemplo IV.17. Este exemplo, de Venables and Ripley (2002), envolve três covariáveis e uma variável de resposta. Os dados são 48 amostras de rochas de um reservatório de petróleo. A resposta é a **permeabilidade** em milli-Darcies. As covariáveis são: **área de poros**, em pixels de 256 por 256, **perímetro** em pixels e **forma**, medida em perímetro/$\sqrt{área}$. O objetivo é prever a permeabilidade utilizando as três covariáveis. Um modelo não paramétrico é

$$permeabilidade = r(área, perímetro, forma) + \epsilon.$$

```
> rock.dat = read.table("rock.dat", header = TRUE)
> library(tree)
> rock.model1 = tree(perm ~ area + peri + shape, data = rock.dat)
> par(mfrow = c(1,1), xpd = NA) # caso contrário, em alguns dispositivos, o texto é recortado
> plot(rock.model1)
> text(rock.model1, cex=.75)
```



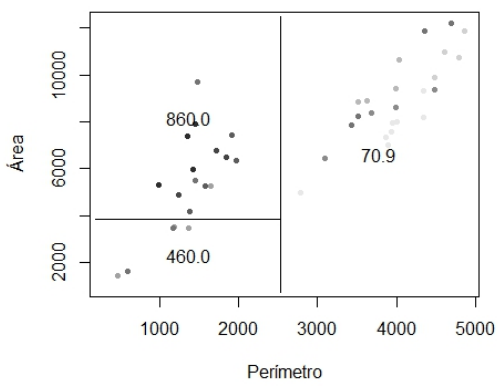
Podemos comparar as previsões com o conjunto de dados.

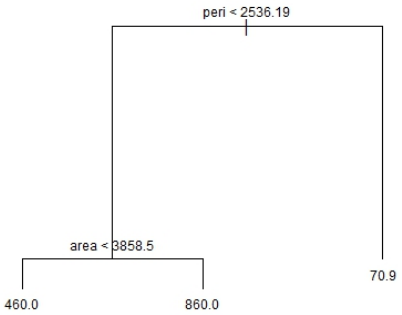
```
> R2(rock.dat$perm,predict(rock.model1))
[1] 0.790755
```

O qual indica um bom ajuste aos dados. Podemos comparar as previsões com o conjunto de dados sendo que, quanto mais escuro maior permeabilidade. Na figura abaixo à esquerda. Observe os valores médios de permeabilidade segundo as diferentes classificações.

```
> mean(rock.dat$perm[rock.dat$area>4000 & rock.dat$peri<2500])
[1] 860
> mean(rock.dat$perm[rock.dat$area<4000 & rock.dat$peri<2500])
[1] 460
```

```
> mean(rock.dat$perm[rock.dat$peri>2500])
[1] 70.9
```





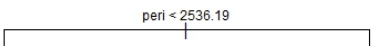

```
> rock.model1
node), split, n, deviance, yval
      * denotes terminal node

 1) root 48 9009000  415.4
   2) peri < 2536.19 24 3252000  760.0
     4) area < 3858.5 6 1123000  460.0 *
     5) area > 3858.5 18 1409000  860.0
      10) shape < 0.31914 13  659500  749.2
        20) area < 6170 5  409700  622.0 *
        21) area > 6170 8  118300  828.8 *
      11) shape > 0.31914 5  175100 1148.0 *
   3) peri > 2536.19 24   58880   70.9 *
> rock.model11 = snip.tree(rock.model1, nodes = 5)
> perm.deciles = quantile(rock.dat$perm, 0:10/10)
> cut.perm = cut(rock.dat$perm, perm.deciles, include.lowest=TRUE)
> plot(rock.dat$peri, rock.dat$area, col=grey(10:2/11)[cut.perm], pch=20, xlab="Perímetro",ylab="Área")
> partition.tree(rock.model11, ordvars = c("peri", "area"), add = TRUE)
```

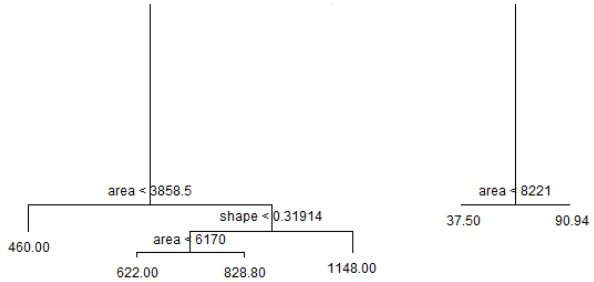
Os gráficos são em duas dimensões, por isso, devemos escolher uma subárvore em **snip.tree** que contenha somente duas variáveis e com esse objetivo eliminamos, por exemplo, o nodo 5 obténdo-se o gráfico a direita acima.

```
> summary(rock.model1)

Regression tree:
tree(formula = perm ~ area + peri + shape, data = rock.dat)
Number of terminal nodes:  5
Residual mean deviance:  43840 = 1885000 / 43
Distribution of residuals:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -522.0   -64.6    11.5     0.0    71.1   840.0
```

No resumo do modelo **Residual mean deviance** significa o erro quadrático médio residual.

A flexibilidade de uma árvore de regressãão é basicamente controlada por quantas folhas elas têm, já que são quantas células elas particionam. A função de ajuste da árvore tem um número de configurações de controles que limitam o quanto crescerá, cada nó deve conter um certo número de pontos e adicionar um nó deve reduzir o erro em pelo menos uma certa quantidade. O padrão para o último, **min.dev**, é 0:01; vamos desligá-lo e ver o que acontece:






```
> rock.model2 = tree(perm ~ area + peri + shape, data = rock.dat, mindev = 0.001)
> plot(rock.model2)
> text(rock.model2, cex=.75)
> summary(rock.model2)

Regression tree:
tree(formula = perm ~ area + peri + shape, data = rock.dat, mindev = 0.001)
Number of terminal nodes:  6
Residual mean deviance:  44500 = 1869000 / 42
Distribution of residuals:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-522.00  -32.34   -8.54    0.00   87.25  840.00
```

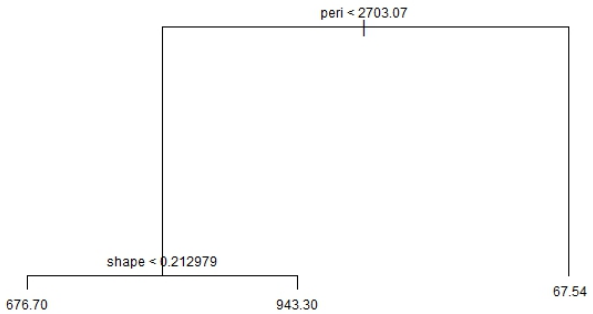
As árvores de classificação geram a classe prevista para uma determinada amostra. Vamos usar aqui o conjunto de dados rock.dat dividido em dois: de treinamento e teste. O conjunto de treinamento será de aproximadamente 70% do conjunto original.

```
> set.seed(101)
> alpha     = 0.7 # percentagem do conjunto de treino
> inTrain   = sample(1:nrow(rock.dat), alpha * nrow(rock.dat))
> train.set = rock.dat[inTrain,]
> test.set  = rock.dat[-inTrain,]
```

Existem duas opções para a saída: a previsão pontual, na qual simplesmente se fornece a predição da classe e a previsão da distribuição, neste fornece-se uma probabilidade para cada classe. Nosso caso a resposta é contínua, então temos somente a previsão pontual.

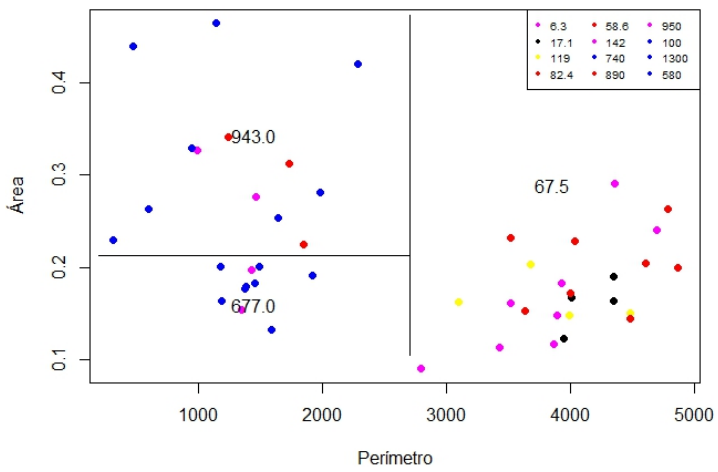
```
> # Ajuste do modelo para a base de treinamento
> rock.model3 = tree(perm ~ area + peri + shape, data = train.set)
> rock.model3
node), split, n, deviance, yval
      * denotes terminal node

1) root 33 6073000 417.10
  2) peri < 2703.07 15 1192000 836.70
    4) shape < 0.212979 6  499300 676.70 *
    5) shape > 0.212979 9  436400 943.30 *
  3) peri > 2703.07 18   41790  67.54 *
> summary(rock.model3)

Regression tree:
tree(formula = perm ~ area + peri + shape, data = train.set)
Variables actually used in tree construction:
[1] "peri"  "shape"
Number of terminal nodes:  3
Residual mean deviance:  32580 = 977500 / 30
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-576.700  -53.330   -8.944    0.000   63.330  356.700
```

```
> # Previsão pontual
> my.prediction = predict(rock.model3, test.set) # gives the probability for each class
> my.prediction
       1        5       10       12       15       24       28       31       38
 67.54444 67.54444 67.54444 67.54444 67.54444 67.54444 676.66667 943.33333 943.33333
      39       40       41       44       47       48
676.66667 943.33333 943.33333 943.33333 676.66667 676.66667
```



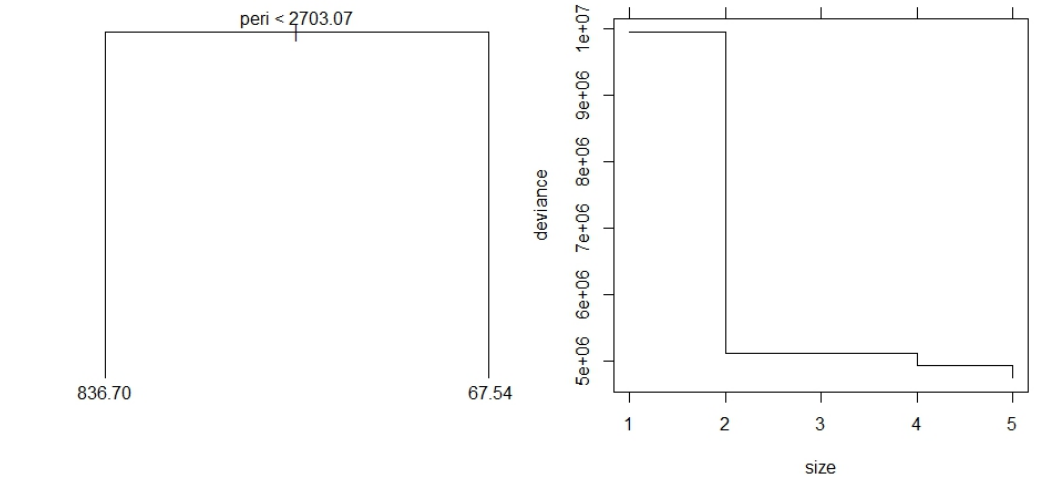
Observe que a variável **área** não aparece na árvore. Isso significa que essa variável nunca foi a covariável ideal para dividir o algoritmo nessa base de dados de treinamento. O resultado é que a árvore depende apenas **área de poros**e do **perímetro**. Isso ilustra uma característica importante da árvode de regressão: ela executa automaticamente a seleção de variáveis no sentido de que uma covariada x_j não aparecerá na árvore se o algoritmo achar que a variável não é importante.

```
> par(mar=c(4,4,1,1))
> plot(rock.dat$peri, rock.dat$shape, pch=19, col=as.numeric(rock.dat$perm),
      xlab = "Perímetro", ylab = "Área")
> partition.tree(rock.model3, label="Perímetro", add=TRUE)
> legend("topright",legend=unique(rock.dat$perm), col = unique(as.numeric(rock.dat$perm)),
        pch=19, cex = 0.6, horiz = FALSE, ncol = 3)
```



Podemos podar a árvore para evitar overfitting. A próxima função **prune.tree()** nos permite escolher quantas folhas queremos que a árvore tenha e ela retorna a melhor árvore com esse tamanho. O argumento **newdata** aceita novas entradas para tomar a decisão de podar. Se novos dados não forem fornecidos, o método usará o conjunto de dados original a partir do qual o modelo de árvore foi criado. Para árvores de classificação, ou seja, de resposta multinomial também podemos usar o método **method = misclass**, de modo que a medida de poda seja o número de erros de classificação.

```
> pruned.rock = prune.tree(rock.model3, best = 2)
> plot(pruned.rock)
> text(pruned.rock)
```



Este pacote também podemos utilizar validação cruzada para encontrar a melhor árvore, usando **cv.tree()**. Aqui, vamos usar todas as variáveis e todas as amostras. Na figura acima mostramos dois gráficos, a árvore com somente duas folha (best = 2) e o resultado da árvore obtida por validação cruzada.

```
> rock.model4 = tree(perm ~ ., data = rock.dat)
> summary(rock.model4)

Regression tree:
tree(formula = perm ~ ., data = rock.dat)
Number of terminal nodes:  5
Residual mean deviance:  43840 = 1885000 / 43
Distribution of residuals:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -522.0   -64.6    11.5     0.0    71.1   840.0
> cv.model = cv.tree(rock.model4)
> plot(cv.model)
```

Mostramos o desvio para cada árove segundo o número de folha, quanto menor melhor.

```
> cv.model$dev
[1] 4745226 4930241 5121875 5123366 9939503
```

Como é muito difícil decidir assim, perguntamos então qual tamanho é melhor?

```
> best.size = cv.model$size[which(cv.model$dev == min(cv.model$dev))]
> best.size
[1] 5
```

e temos por resposta a árove com 5 folhas. Vamos refazer o modelo da árvore com o número de folhas não sendo maior que o melhor tamanho.

```
> cv.model.pruned = prune.tree(rock.model4, best = best.size)
> summary(cv.model.pruned)

Regression tree:
tree(formula = perm ~ ., data = rock.dat)
Number of terminal nodes:  5
Residual mean deviance:  43840 = 1885000 / 43
```

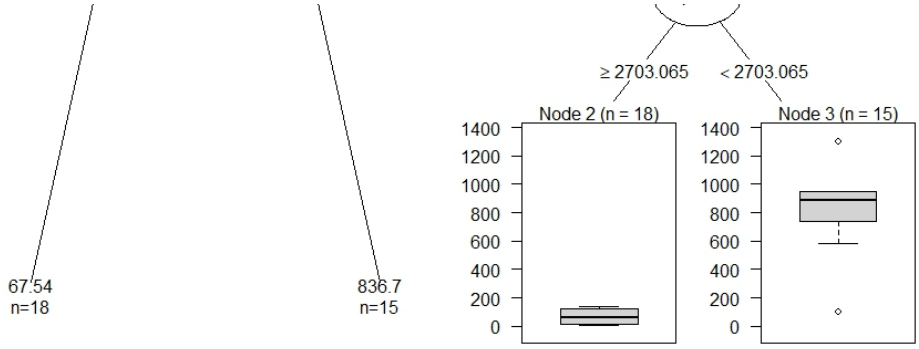
```
Distribution of residuals:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -522.0   -64.6    11.5     0.0    71.1   840.0
```

Podemos fazer mais bonito ainda. O desenvolvimento teórico permanece o mesmo porém o pacote **rpart** é mais rápido do que o **tree** e a qualidade de plotagem e funções de texto são melhores utilizando o pacote **partykit**.

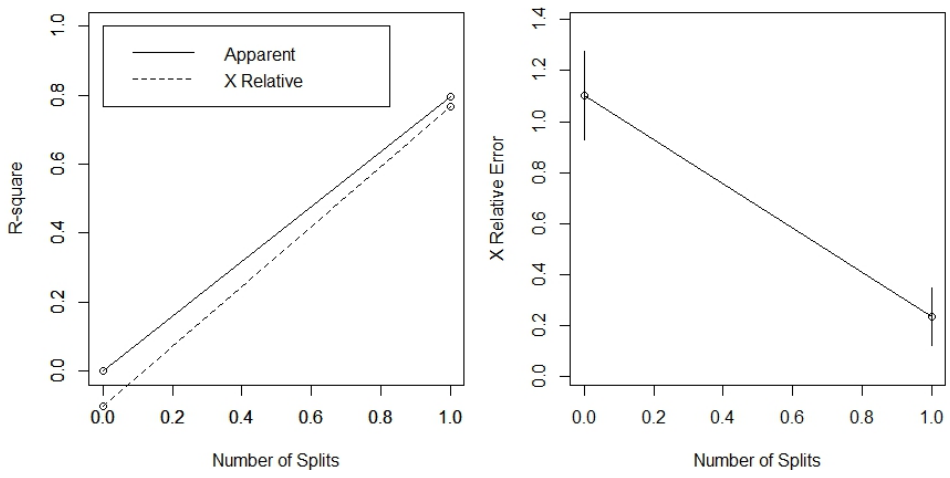
Exemplo IV.18. (Continuação do Exemplo IV.17)

```
> library(rpart)
> rock.rpart = rpart(perm ~ ., data = train.set)
> plot(rock.rpart, uniform=TRUE, branch=0.6, margin=0.05)
> text(rock.rpart, all=TRUE, use.n=TRUE)
> title("Árvore no Conjunto de Treinamento")
> predictions = predict(rock.rpart, test.set)
> table(test.set$perm, predictions)
      predictions
       67.5444444444444 836.666666666667
  6.3                 1                 0
  17.1                1                 0
  82.4                1                 0
  100                 0                 3
  119                 2                 0
  142                 1                 0
  580                 0                 2
  740                 0                 1
  890                 0                 1
  1300                0                 2
> prune.rpart = prune(rock.rpart, cp=0.02) # podando a árvore
> plot(prune.rpart, uniform=TRUE, branch=0.6)
> text(prune.rpart, all=TRUE, use.n=TRUE)
> summary(prune.rpart)
Call:
rpart(formula = perm ~ ., data = train.set)
  n= 33

         CP nsplit rel error    xerror      xstd
1 0.7968991      0 1.0000000 1.1467430 0.1734081
2 0.0100000      1 0.2031009 0.2349355 0.1118926

Variable importance
 peri  area shape
   48    35    16

Node number 1: 33 observations,    complexity param=0.7968991
  mean=417.1455, MSE=184044.7
  left son=2 (18 obs) right son=3 (15 obs)
  Primary splits:
      peri  < 2703.065  to the right, improve=0.7968991, (0 missing)
      area  < 7936.5    to the right, improve=0.4677176, (0 missing)
      shape < 0.2693715 to the left,  improve=0.2583957, (0 missing)
  Surrogate splits:
      area  < 6755.5    to the right, agree=0.879, adj=0.733, (0 split)
      shape < 0.251364  to the left,  agree=0.697, adj=0.333, (0 split)

Node number 2: 18 observations
  mean=67.54444, MSE=2321.938

Node number 3: 15 observations
  mean=836.6667, MSE=79448.89
```

Apresentamos o resultado do modelo quando podamos a árove mas não o gráfico porque é o mesmo modelo.




```
> # criando gráficos adicionais
> par(mfrow = c(1,2)) # dois gráficos em uma página
> rsq.rpart(rock.rpart1) # visualiza resultados de validação cruzada

Regression tree:
rpart(formula = perm ~ ., data = train.set, method = "anova")

Variables actually used in tree construction:
[1] peri

Root node error: 6073475/33 = 184045

n= 33

      CP nsplit rel error  xerror    xstd
1 0.7969      0    1.0000 1.10050 0.17373
2 0.0100      1    0.2031 0.23452 0.11226
```



IV.9 Exercícios

1. Obter os dados sobre fragmentos de vidro coletados em trabalhos forenses em **glass.dat**. Considere que RI seja o índice de refracção e que Al seja o teor de alumínio. Realize uma regressão não paramétrica para ajustar o modelo $RI = r(Al) + \epsilon$. Utilize os seguintes estimadores:
   - i) regressograma,
   - ii) kernel,
   - iii) regressão linear local,
   - iv) spline.

   Em cada caso, utilizar validação cruzada para escolher o parâmetro de alisamento. Estime a variância. Construir bandas de confiança de 95% para as suas estimativas.

2. O número de laranjas podres y em 10 caixas selecionadas aleatoriamente de uma grande remessa é contada depois de terem sido armazenadas por um número determinado de dias x. Use o método Theil-Kendall para calcular a inclinação de uma linha reta ajustada a esses dados e obtenha uma estimativa apropriada do intercepto.

x	3	5	8	11	15	18	20	25	27	30
y	2	4	7	10	17	23	29	45	59	73

Plote estes dados e encontre o modelo ajustado. O ajuste parece razoável?

3. Os seguintes dados têm como base no censo de agricultura dos EUA, que fornece, em intervalos de aproximadamente 10 anos, de 1920 a 1980, as porcentagens de fazendas dos EUA com tratores e fazendas com cavalos. Explique por que seria inútil ou errado ajustar uma regressão linear para porcentagem de tratores versus porcentagem de cavalos utilizando esses dados. Sugira que tipo alternativo de regressão pode ser mais apropriado.

Percentagem de tratores	9.2	30.9	58.8	72.7	89.9	88.7	90.2
Percentagem de cavalos	91.8	88.0	80.6	43.6	16.7	14.4	10.5

4. Dados simulados de acidentes de motocicleta: os dados em **motor.dat** possuem 94 linhas e 4 colunas. A covariável é o tempo, chamado de **times**, medido em milissegundos e a resposta é a aceleração **accel** no momento do impacto. Use validação cruzada para ajustar uma curva suave usando a regressão linear local.

5. Em 1976 dois pesquisadores mediram a concentração de amônia y em mg/l em várias profundidades x, em metros, no Mar Morto. Ajustar uma regressão linear para a concentração segundo as profundidades usando o método de Kendall-Theil e obter um intervalo de confiança aproximado de 95 por cento para β.

x	25	50	100	150	155	187	200	237	287	290	300
y	6.13	5.51	6.18	6.70	7.22	7.28	7.22	7.48	7.38	7.38	7.64

6. Numa pesquisa em 1965 fornece-nos dados para peso do alimento ingerido x e ganho de peso y para 10 suínos alimentados com um tipo de alimento A e para 10 alimentados com um segundo tipo B. Use o método adequado para ajustar as regressões lineares a cada uma das situações e testar se a hipótese de que as inclinações são iguais é suportada.

A	x	575	585	628	632	637	638	661	674	694	713
	y	130	146	156	164	158	151	159	165	167	170
B	x	625	646	651	678	710	722	728	754	763	831
	y	147	164	149	160	184	173	193	189	200	201

7. Sejam $Y_i \sim N(\mu_i, 1)$ para $i = 1, 2, \cdots, n$ observações independentes. Encontre os estimadores que minimizam cada uma das seguintes somas de quadrados penalizadas:

(a) $\sum_{i=1}^{n}(Y_i - \widehat{\mu}_i)^2 \, + \, \lambda \sum_{i=1}^{n} \widehat{\mu}_i^2$.

(b) $\sum_{i=1}^{n}(Y_i - \widehat{\mu}_i)^2 \, + \, \lambda \sum_{i=1}^{n} |\widehat{\mu}_i|$.

(c) $\sum_{i=1}^{n}(Y_i - \widehat{\mu}_i)^2 \, + \, \lambda \sum_{i=1}^{n} I\big(\widehat{\mu}_i = 0\big)$.

8. Seja $\hat{r}_n(x_1, x_2) = \sum_{i=1}^{n} Y_i l_i(x_1, x_2)$ um estimador linear da função de regressão múltipla $r\big(x_1, x_2\big)$. Suponha que queremos testar a hipótese de que a covariável x_2 pode ser descartada da regressão. Uma possibilidade é formar um estimador linear da forma $\tilde{r}_n(x_1) = \sum_{i=1}^{n} Y_i \tilde{l}_i(x_1)$ e em seguida, calcular

$$T \, = \, \sum_{i=1}^{n} \Big(\hat{r}_n(x_{1i}, x_{2i}) - \tilde{r}_n(x_{1i}) \Big)^2 .$$

(a) Suponha que o verdadeiro modelo seja $Y_i = r\big(x_{1i}\big) + \epsilon_i$, onde $\epsilon_i \sim N(0, \sigma^2)$. Por simplicidade assuma σ conhecido. Encontre uma expressão para a distribuição de T.

(b) A distribuição nula na parte (a) depende da função desconhecida $r\big(x_{1i}\big)$. Como você pode estimar a distribuição nula?

(c) Crie dados simulados do modelo em (a), use qualquer função $r\big(x_1\big)$ desejada, e veja se o método proposto em (b) aproxima a distribuição nula.