

Introdução ao Algoritmo EM

Erick Amorim

Departamento de Estatística
Universidade Federal de Minas Gerais

30 de Setembro de 2016

SUMÁRIO

Introdução

Um Exemplo para Motivar

Alguns Detalhes do Algoritmo EM

Aplicação em Exemplos

Expectation-Maximization

- ▶ O algoritmo EM (Expectation-Maximization) é uma técnica de otimização originalmente introduzida por Dempster, Laird e Rubin (1977).
- ▶ É um algoritmo usado para obter o estimador que maximiza a verossimilhança (EMV).
- ▶ Segundo Casella e Berge (2010), o EM é um algoritmo que seguramente converge para o EMV e tem como base substituir uma difícil maximização da verossimilhança por uma sequência de maximizações mais fáceis, cujo limite é o EMV.
- ▶ A demonstração original da Convergência do EM realizada por Dempster, Laird e Rubin (1977) tinha uma falha, mas provas válidas da convergência foram apresentadas por Boyles (1983) e Wu (1983).

EM

- ▶ O algoritmo EM tem grande utilidade em problemas de dados incompletos, onde a estimação de máxima verossimilhança é difícil devido a ausência de parte dos dados.
- ▶ O algoritmo é muito útil em problemas de: Clustering, reconhecimento de padrões, modelos de Markov oculto, entre outros.

O algoritmo consiste de duas etapas:

- Etapas**
- E** Calcula-se o valor esperado do logaritmo da verossimilhança;
 - M** Encontra-se o máximo deste valor esperado, iterando a esperança até a convergência ou critério de parada.

Aplicação 1

Este problema foi proposto por Rao (1973, p. 368-369). McLachlan and Krishnan (1997) também discutem este problema e apresentaremos o modelo deles.

Os dados abaixo se referem a $n = 197$ animais que se distribuem em 4 categorias (C_1, C_2, C_3, C_4) de modo que o vetor de frequência observado é $\mathbf{x} = (x_1, x_2, x_3, x_4)' = (125, 18, 20, 34)'$.

Neste caso temos uma distribuição Multinomial com 4 categorias e função de probabilidade dada por:

$$P(X_1 = x_1, X_2 = x_2, X_3 = x_3, X_4 = x_4) = g(\mathbf{x} \mid \theta)$$

$$\text{com } g(\mathbf{x} \mid \theta) = \frac{n!}{x_1!x_2!x_3!x_4!} \left(\frac{1}{2} + \frac{\theta}{4}\right)^{x_1} \left(\frac{1}{4} - \frac{\theta}{4}\right)^{x_2} \left(\frac{1}{4} - \frac{\theta}{4}\right)^{x_3} \left(\frac{\theta}{4}\right)^{x_4}$$

onde $n = x_1 + x_2 + x_3 + x_4$ e $0 \leq \theta \leq 1$.

Solução Analítica da Aplicação 1

A função de verossimilhança $L(\theta; \mathbf{x}) = g(\mathbf{x} | \theta)$

$$g(\mathbf{x} | \theta) = \frac{n!}{x_1!x_2!x_3!x_4!} \left(\frac{1}{2} + \frac{\theta}{4}\right)^{x_1} \left(\frac{1}{4}(1 - \theta)\right)^{x_2+x_3} \left(\frac{\theta}{4}\right)^{x_4}$$

$$g(\mathbf{x} | \theta) \propto (2 + \theta)^{x_1} (1 - \theta)^{x_2+x_3} \theta^{x_4}$$

e o logaritmo da verossimilhança será:

$$l(\theta) \propto x_1 \ln(2 + \theta) + (x_2 + x_3) \ln(1 - \theta) + x_4 \ln \theta$$

e além do mais:

$$\frac{d}{d\theta} l(\theta) = \frac{x_1}{2 + \theta} - \frac{x_2 + x_3}{1 - \theta} + \frac{x_4}{\theta}$$

fazendo $\frac{d}{d\theta} l(\theta) = 0$ chegaremos na equação

Solução Analítica da Aplicação 1

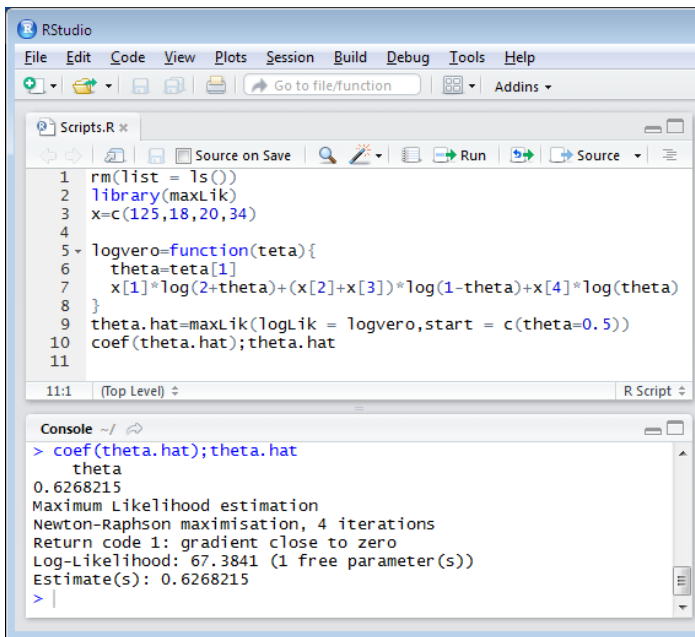
$n\theta^2 + (2x_2 + 2x_3 - x_1 + x_4)\theta - 2x_4 = 0$ cuja solução será:

$$\hat{\theta}_{MV} = \frac{-(2x_2 + 2x_3 - x_1 + x_4) + \sqrt{(2x_2 + 2x_3 - x_1 + x_4)^2 + 8nx_4}}{2n}$$

então para $\mathbf{x} = (125, 18, 20, 34)'$ teremos

$$\hat{\theta}_{MV} = \frac{15 + \sqrt{53809}}{394} = 0,626821498$$

Script da Aplicação 1



The screenshot shows the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Tools, and Help. Below the menu is a toolbar with icons for file operations and a search bar labeled 'Go to file/function'. The main editor window displays a script named 'Scripts.R' with the following R code:

```
1 rm(list = ls())
2 library(maxLik)
3 x=c(125,18,20,34)
4
5 logvero=function(teta){
6   theta=teta[1]
7   x[1]*log(2+theta)+(x[2]+x[3])*log(1-theta)+x[4]*log(theta)
8 }
9 theta.hat=maxLik(logLik = logvero,start = c(theta=0.5))
10 coef(theta.hat);theta.hat
11
```

The status bar at the bottom of the editor shows '11:1 (Top Level)' and 'R Script'. Below the editor is the Console window, which shows the output of the executed code:

```
> coef(theta.hat);theta.hat
      theta
0.6268215
Maximum Likelihood estimation
Newton-Raphson maximisation, 4 iterations
Return code 1: gradient close to zero
Log-Likelihood: 67.3841 (1 free parameter(s))
Estimate(s): 0.6268215
> |
```


Alguns detalhes do Método

Suponha que $\mathbf{X} = (X_1, \dots, X_n)$ sejam v.a. i.i.d. com distribuição conjunta $g(\mathbf{x} \mid \theta)$. Neste caso $g(\mathbf{x} \mid \theta) = L(\theta; \mathbf{x})$. E o que se deseja é $\hat{\theta} = \arg \max L(\theta; \mathbf{x})$.

Vamos considerar que os dados completos \mathbf{w} são provenientes de uma amostra aleatória $\mathbf{W} = (\mathbf{X}, \mathbf{Z})$. Neste caso a verossimilhança completa é:

$$f(\mathbf{w} \mid \theta) = f(\mathbf{x}, \mathbf{z} \mid \theta) = k(\mathbf{z} \mid \mathbf{x}, \theta)g(\mathbf{x} \mid \theta)$$

O objetivo é maximizar $L(\theta \mid \mathbf{x})$ usando a verossimilhança completa $L^c(\theta \mid \mathbf{x}, \mathbf{z}) = f(\mathbf{x}, \mathbf{z} \mid \theta)$ neste processo.

Como calculamos $L^c(\theta \mid \mathbf{x}, \mathbf{z})$ se não conhecemos \mathbf{z} ?

Alguns Detalhes do Método

Como não conhecemos \mathbf{z} em $L^c(\theta \mid \mathbf{x}, \mathbf{z})$ vamos considerar que $L^c(\theta \mid \mathbf{x}, \mathbf{z})$ é uma v.a.

Calcularemos uma média de funções de verossimilhanças sobre variáveis não observadas (latentes).

Há uma relação entre as verossimilhanças dada por:

$$L^c(\theta \mid \mathbf{x}, \mathbf{z}) = k(\mathbf{z} \mid \mathbf{x}, \theta) L(\theta \mid \mathbf{x})$$

$$\ln L^c(\theta \mid \mathbf{x}, \mathbf{z}) = \ln k(\mathbf{z} \mid \mathbf{x}, \theta) + \ln L(\theta \mid \mathbf{x})$$

Usando a igualdade $k(\mathbf{z} \mid \mathbf{x}, \theta) = \frac{f(\mathbf{x}, \mathbf{z} \mid \theta)}{g(\mathbf{x} \mid \theta)}$ e para um valor $\theta_0 \in \Theta$ fixado teremos:

Com Alguns Cálculos

$$\ln L(\theta \mid \mathbf{x}) = \int \ln [L(\theta \mid \mathbf{x})] k(z \mid \mathbf{x}, \theta_0) d\mathbf{z}$$

$$\ln L(\theta \mid \mathbf{x}) = \int \ln [g(\mathbf{x} \mid \theta)] k(z \mid \mathbf{x}, \theta_0) d\mathbf{z}$$

$$\ln L(\theta \mid \mathbf{x}) = \int \ln \left[\frac{f(\mathbf{x}, \mathbf{z} \mid \theta)}{k(\mathbf{z} \mid \mathbf{x}, \theta)} \right] k(z \mid \mathbf{x}, \theta_0) d\mathbf{z}$$

$$\ln L(\theta \mid \mathbf{x}) = \int [\ln f(\mathbf{x}, \mathbf{z} \mid \theta) - \ln k(\mathbf{z} \mid \mathbf{x}, \theta)] k(z \mid \mathbf{x}, \theta_0) d\mathbf{z}$$

$$\ln L(\theta \mid \mathbf{x}) = \int \ln f(\mathbf{x}, \mathbf{z} \mid \theta) k(z \mid \mathbf{x}, \theta_0) d\mathbf{z} - \int \ln k(\mathbf{z} \mid \mathbf{x}, \theta) k(z \mid \mathbf{x}, \theta_0) d\mathbf{z}$$

$$\underbrace{\ln L(\theta \mid \mathbf{x})}_{\text{Dados Observados}} = \underbrace{E_{\theta_0} [\ln L^c(\theta \mid \mathbf{x}, \mathbf{z}) \mid \theta_0, \mathbf{x}]}_{\text{Dados Completos}} - \underbrace{E_{\theta_0} [\ln k(\mathbf{z} \mid \mathbf{x}, \theta) \mid \theta_0, \mathbf{x}]}_{\text{Dados Ausentes}}$$

Alguns Detalhes do Método

Definimos a função Q do termo da equação anterior que será Maximizada no passo M como $Q(\theta \mid \theta_0, \mathbf{x}) = E_{\theta_0} [\ln L^c(\theta \mid \mathbf{x}, \mathbf{z}) \mid \theta_0, \mathbf{x}]$

No processamento do algoritmo, maximizar $Q(\theta \mid \theta_0, \mathbf{x})$ é equivalente a maximizar $\ln L(\theta \mid \mathbf{x})$

Denote por $\theta^{(0)}$ uma estimativa inicial de θ . Então $\theta^{(1)}$ é o argumento que maximiza $Q(\theta \mid \theta_0, \mathbf{x})$. Esse é o primeiro passo para estimar θ . Se nós procedermos desta forma teremos o algoritmo definido como:

Algoritmo EM

Seja $\theta^{(m)}$ a estimativa na m -ésima iteração. Para calcular a estimativa na iteração $(m + 1)$ proceda da seguinte maneira:

1. Etapa E: Calcule

$$Q(\theta \mid \hat{\theta}_m, \mathbf{x}) = E_{\hat{\theta}^{(m)}} \left[\ln L^c(\theta \mid \mathbf{x}, \mathbf{z}) \mid \hat{\theta}_m, \mathbf{x} \right]$$

2. Etapa M: Calcule $\hat{\theta}^{(m+1)} = \arg \max Q(\theta \mid \hat{\theta}_m, \mathbf{x})$

Sob fortes condições pode ser mostrado que $\hat{\theta}^{(m)}$ converge em probabilidade para o EMV quando $m \rightarrow \infty$. Isso não será mostrado aqui.

Mas pode-se mostrar que $\hat{\theta}^{(m+1)}$ sempre aumenta a verossimilhança em relação a $\hat{\theta}^{(m)}$.

Alguns Detalhes do Método

Teorema: A sequência de estimadores $\hat{\theta}^{(m)}$ satisfaz

$$L(\hat{\theta}^{(m+1)} \mid \mathbf{x}) \geq L(\hat{\theta}^{(m)} \mid \mathbf{x}).$$

Prova: Isso é devido a $\hat{\theta}^{(m+1)}$ maximizar $Q(\theta \mid \hat{\theta}^{(m)}, \mathbf{x})$, pois temos:

$$Q(\hat{\theta}^{(m+1)} \mid \hat{\theta}^{(m)}, \mathbf{x}) \geq Q(\hat{\theta}^{(m)} \mid \hat{\theta}^{(m)}, \mathbf{x})$$

que pode ser escrito como,

$$E_{\hat{\theta}^{(m)}} \left[\ln L^c(\hat{\theta}^{(m+1)} \mid \mathbf{x}, \mathbf{z}) \right] \geq E_{\hat{\theta}^{(m)}} \left[\ln L^c(\hat{\theta}^{(m)} \mid \mathbf{x}, \mathbf{z}) \right],$$

onde a esperança é em relação a $k(\mathbf{z} \mid \hat{\theta}^{(m)}, \mathbf{x})$.

A prova é completada mostrando que

$$E_{\hat{\theta}^{(m)}} \left[\ln k(\mathbf{z} \mid \hat{\theta}^{(m+1)}, \mathbf{x}) \right] \leq E_{\hat{\theta}^{(m)}} \left[\ln k(\mathbf{z} \mid \hat{\theta}^{(m)}, \mathbf{x}) \right]$$

Alguns Detalhes do Método

Aplicando a desigualdade de Jensen tem-se:

$$E_{\hat{\theta}^{(m)}} \left[\ln \frac{k(\mathbf{z} \mid \hat{\theta}^{(m+1)}, \mathbf{x})}{k(\mathbf{z} \mid \hat{\theta}^{(m)}, \mathbf{x})} \right] \leq \ln E_{\hat{\theta}^{(m)}} \left[\frac{k(\mathbf{z} \mid \hat{\theta}^{(m+1)}, \mathbf{x})}{k(\mathbf{z} \mid \hat{\theta}^{(m)}, \mathbf{x})} \right]$$

$$= \ln \int \frac{k(\mathbf{z} \mid \hat{\theta}^{(m+1)}, \mathbf{x})}{k(\mathbf{z} \mid \hat{\theta}^{(m)}, \mathbf{x})} k(\mathbf{z} \mid \hat{\theta}^{(m)}, \mathbf{x}) d\mathbf{z} = \ln(1) = 0.$$

Aplicação 1

Recordando o exemplo que apresentamos referente aos dados de $n = 197$ animais que se distribuem em 4 categorias, de modo que o vetor de frequência é $\mathbf{x} = (x_1, x_2, x_3, x_4)' = (125, 18, 20, 34)'$.

- ▶ A função de distribuição conjunta para os dados observados é

$$g(\mathbf{x} \mid \theta) = \frac{n!}{x_1!x_2!x_3!x_4!} \left(\frac{1}{2} + \frac{\theta}{4}\right)^{x_1} \left(\frac{1}{4} - \frac{\theta}{4}\right)^{x_2} \left(\frac{1}{4} - \frac{\theta}{4}\right)^{x_3} \left(\frac{\theta}{4}\right)^{x_4}$$

- ▶ Cujo núcleo é $g(\mathbf{x} \mid \theta) \propto (2 + \theta)^{x_1} (1 - \theta)^{x_2+x_3} \theta^{x_4}$.

Agora suponha que os dados da maior categoria seja proveniente das categorias z_1 e z_2 , com probabilidades $\frac{1}{2}$ e $\frac{\theta}{4}$, respectivamente.

Aplicação 1

Desta maneira se introduzem variáveis latentes ($x_1 = z_1 + z_2$) resultando em 5 categorias: $w = (z_1, z_2, x_2, x_3, x_4)'$ que representam os dados completos e $\mathbf{z} = (z_1, z_2)$ os dados perdidos.

- * A distribuição conjunta de \mathbf{w} é $f(\mathbf{w}) = f(\mathbf{x}, \mathbf{z} \mid \theta)$, onde

$$f(\mathbf{x}, \mathbf{z} \mid \theta) = \frac{n!}{z_1!z_2!x_2!x_3!x_4!} \left(\frac{1}{2}\right)^{z_1} \left(\frac{\theta}{4}\right)^{z_2} \left(\frac{1}{4} - \frac{\theta}{4}\right)^{x_2+x_3} \left(\frac{\theta}{4}\right)^{x_4}$$

- * E o núcleo da distribuição é

$$f(\mathbf{x}, \mathbf{z} \mid \theta) \propto \theta^{z_2+x_4} (1 - \theta)^{x_2+x_3}$$

Aplicação 1

Usando as distribuições dos dados completos e dos dados observados podemos calcular a distribuição de $[\mathbf{z} \mid \mathbf{x}, \theta]$ que é

$$k(\mathbf{z} \mid \mathbf{x}, \theta) = \frac{f(\mathbf{x}, \mathbf{z} \mid \theta)}{g(\mathbf{x} \mid \theta)} = \binom{x_1}{z_2} \left(\frac{\theta}{2 + \theta} \right)^{z_2} \left(1 - \frac{\theta}{2 + \theta} \right)^{x_1 - z_2}$$

Veja que $Z_2 \sim \text{Binomial} \left(x_1, \frac{\theta}{2 + \theta} \right)$ e sua média é $E[Z_2] = x_1 \left(\frac{\theta}{2 + \theta} \right)$.

Para o **passo E** precisamos calcular a esperança de $\ln L^c(\theta \mid \mathbf{x}, \mathbf{z})$ em relação a $k(\mathbf{z} \mid \theta^{(0)}, \mathbf{x})$.

Aplicação 1

$$\ln L^c(\theta \mid \mathbf{x}, \mathbf{z}) = \ln f(\mathbf{x}, \mathbf{z} \mid \theta) \propto (Z_2 + x_4) \ln \theta + (x_2 + x_3) \ln(1 - \theta)$$

$$E_{\hat{\theta}^{(m)}} [\ln L^c(\theta \mid \mathbf{x}, \mathbf{z})] \propto E_{\hat{\theta}^{(m)}} [(Z_2 + x_4) \ln \theta + (x_2 + x_3) \ln(1 - \theta)]$$

$$E_{\hat{\theta}^{(m)}} [\ln L^c(\theta \mid \mathbf{x}, \mathbf{z})] \propto E_{\hat{\theta}^{(m)}} [Z_2] \ln \theta + x_4 \ln \theta + (x_2 + x_3) \ln(1 - \theta)$$

$$Q(\theta \mid \hat{\theta}^{(m)}) \propto \frac{x_1 \theta^{(m)}}{2 + \theta^{(m)}} \ln \theta + x_4 \ln \theta + (x_2 + x_3) \ln(1 - \theta)$$

Para o **passo M** precisamos encontrar o maximo de $Q(\theta \mid \hat{\theta}^{(m)})$;

$$\frac{d}{d\theta} Q(\theta \mid \hat{\theta}^{(m)}) = 0$$

$$\text{chega-se a } \hat{\theta} = \frac{x_4 + x_1 \frac{\hat{\theta}^{(m)}}{2 + \hat{\theta}^{(m)}}}{x_2 + x_3 + x_4 + x_1 \frac{\hat{\theta}^{(m)}}{2 + \hat{\theta}^{(m)}}}$$

Aplicação 1

Substituindo os dados $x_1 = 125$, $x_2 = 18$, $x_3 = 20$ e $x_4 = 34$ tem-se

$$\hat{\theta} = \frac{34 + 125 \frac{\hat{\theta}^{(m)}}{2 + \hat{\theta}^{(m)}}}{72 + 125 \frac{\hat{\theta}^{(m)}}{2 + \hat{\theta}^{(m)}}}$$

Em cada iteração tem-se

$$\hat{\theta}^{(m+1)} = \frac{68 + 159\hat{\theta}^{(m)}}{144 + 197\hat{\theta}^{(m)}}$$

Veja o Script no R.

Aplicação 2

Vamos considerar um problema envolvendo misturas de normais. Suponha que $Y_1 \sim N(\mu_1, \sigma_1^2)$ e $Y_2 \sim N(\mu_2, \sigma_2^2)$. Considere também uma v.a. W sendo uma Bernoulli independente de Y_1 e Y_2 com probabilidade de sucesso $\epsilon = P(W = 1)$.

Suponha que a variável aleatória que observamos seja

$$X = (1 - W)Y_1 + WY_2.$$

Nesse caso o vetor de parâmetros é $\theta' = (\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \epsilon)$ e a função densidade de X é:

$$f(x) = (1 - \epsilon)f_1(x) + \epsilon f_2(x)$$

Aplicação 2

Em problemas de misturas os dados não observáveis são v.a. que identificam as componentes da mistura. Então para $i = 1, \dots, n$ definimos variáveis

$$W_i = \begin{cases} 0, & \text{se } X_i \text{ tem densidade } f_1(x) \\ 1, & \text{se } X_i \text{ tem densidade } f_2(x). \end{cases}$$

Desta forma os $W_{i's}$ são i.i.d. com probabilidade de sucesso ϵ .

A verossimilhança completa será:

$$L^c(\theta \mid \mathbf{x}, \mathbf{w}) = \prod_{W_i=0} f_1(x_i) \prod_{W_i=1} f_2(x_i)$$

Aplicação 2

O logaritmo da verossimilhança completa é:

$$\ln L^c(\theta \mid \mathbf{x}, \mathbf{w}) = \sum_{W_i=0} \ln f_1(x_i) + \sum_{W_i=1} \ln f_2(x_i)$$

$$\ln L^c(\theta \mid \mathbf{x}, \mathbf{w}) = \sum_{i=1}^n [(1 - W_i) \ln f_1(x_i) + W_i \ln f_2(x_i)].$$

Então calculando o valor esperado tem-se:

$$E_{\hat{\theta}^{(m)}} [\ln L^c(\theta \mid \mathbf{x}, \mathbf{w})] = \sum_{i=1}^n [(1 - E_{\hat{\theta}^{(m)}}[W_i]) \ln f_1(x_i) + E_{\hat{\theta}^{(m)}}[W_i] \ln f_2(x_i)]$$

Para o **passo E** precisamos da esperança condicional de W_i dado \mathbf{x} e $\theta^{(m)}$ que é $E[W_i \mid \theta^{(m)}, \mathbf{x}] = P[W_i = 1 \mid \theta^{(m)}, \mathbf{x}]$

Aplicação 2

Uma estimativa desse valor esperado é dado por:

$$\gamma_i = \hat{P} \left[W_i = 1 \mid \theta^{(m)}, \mathbf{x} \right] = \frac{\hat{\epsilon} f_{2,(m)}(x_i)}{(1 - \hat{\epsilon}) f_{1,(m)}(x_i) + \hat{\epsilon} f_{2,(m)}(x_i)},$$

detalhes sobre a expressão acima podem ser vistas em McLachlan and Krishnan (1997).

Substituindo γ_i na Esperança, temos a expressão do **passo E**:

$$Q(\theta \mid \hat{\theta}^{(m)}, \mathbf{x}) = \sum_{i=1}^n [(1 - \gamma_i) \ln f_1(x_i) + \gamma_i \ln f_2(x_i)]$$

Para o **passo M** basta derivar em relação a cada parâmetro por exemplo:

$$\frac{\partial}{\partial \mu_1} Q(\theta \mid \hat{\theta}^{(m)}, \mathbf{x}) = \sum_{i=1}^n (1 - \gamma_i) \left(-\frac{1}{2\sigma_1^2} \right) (-2)(x_i - \mu_1)$$

Aplicação 2

Fazendo o cálculo para cada μ e σ^2 obtemos as seguintes estimativas:

$$\hat{\mu}_1 = \frac{\sum_{i=1}^n (1 - \gamma_i) x_i}{\sum_{i=1}^n (1 - \gamma_i)}$$

$$\hat{\sigma}_1^2 = \frac{\sum_{i=1}^n (1 - \gamma_i) (x_i - \hat{\mu}_1)^2}{\sum_{i=1}^n (1 - \gamma_i)}$$

$$\hat{\mu}_2 = \frac{\sum_{i=1}^n \gamma_i x_i}{\sum_{i=1}^n \gamma_i}$$

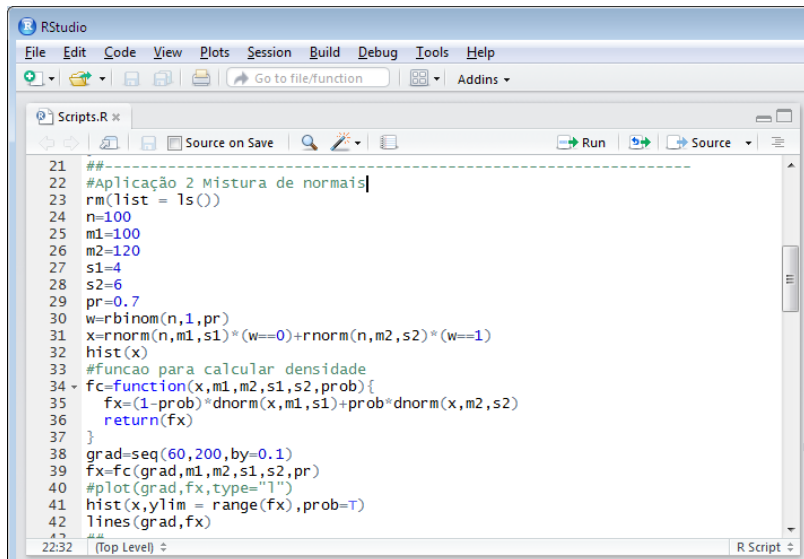
$$\hat{\sigma}_2^2 = \frac{\sum_{i=1}^n \gamma_i (x_i - \hat{\mu}_2)^2}{\sum_{i=1}^n \gamma_i}$$

Já que a estimativa de $P[W_i = 1 \mid \theta^{(m)}, \mathbf{x}]$ é γ_i , então para estimar $\epsilon = P(W_i = 1)$ usamos a média $\hat{\epsilon} = \frac{\sum_{i=1}^n \gamma_i}{n}$

Aplicação 2

Os seguintes dados foram gerados da v.a. $X = (1 - W)Y_1 + WY_2$ com $Y_1 \sim N(100, 4^2)$, $Y_2 \sim N(120, 6^2)$ e $W \sim \text{Bernoulli}(0.7)$.
Veja o Script no R.

Aplicação 2

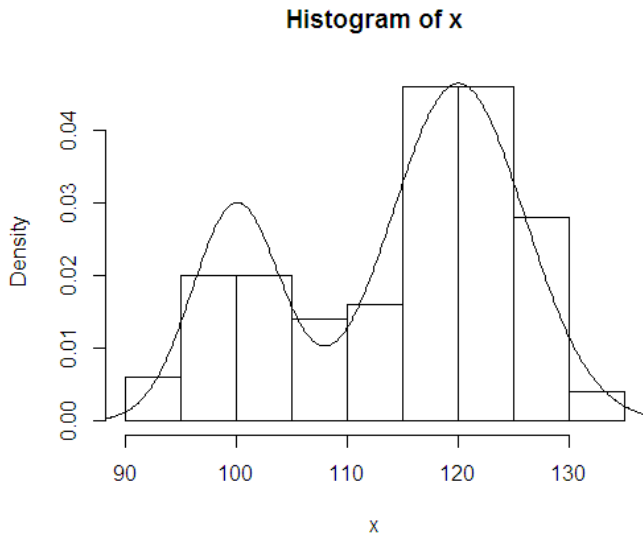


The screenshot shows the RStudio interface with a script editor open. The script is titled "Scripts.R" and contains R code for generating a mixture of two normal distributions. The code includes comments in Portuguese, variable assignments for sample size and parameters, random number generation, a custom density function, and plotting commands. The status bar at the bottom indicates the current line is 22:32 and the editor is at the Top Level.

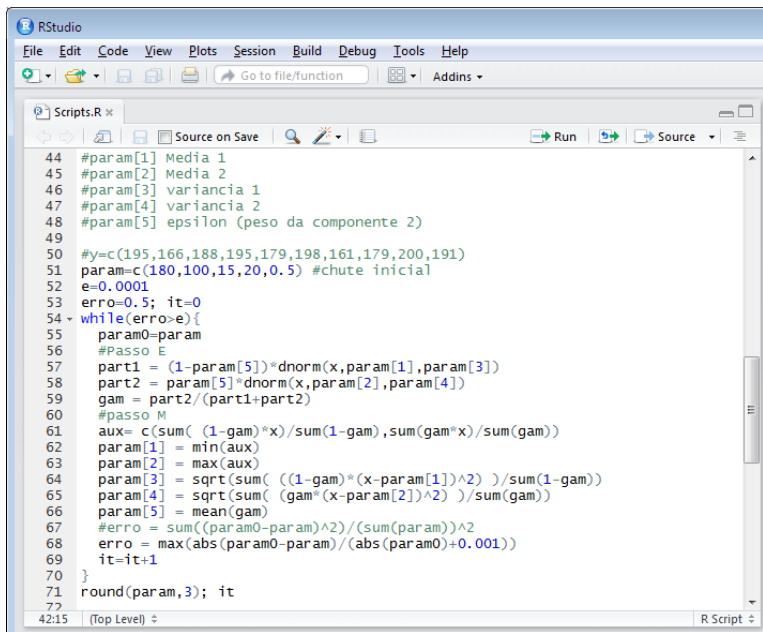
```
21 ##-----
22 #Aplicação 2 Mistura de normais|
23 rm(list = ls())
24 n=100
25 m1=100
26 m2=120
27 s1=4
28 s2=6
29 pr=0.7
30 w=rbinom(n,1,pr)
31 x=rnorm(n,m1,s1)*(w==0)+rnorm(n,m2,s2)*(w==1)
32 hist(x)
33 #funcao para calcular densidade
34 fc=function(x,m1,m2,s1,s2,prob){
35   fx=(1-prob)*dnorm(x,m1,s1)+prob*dnorm(x,m2,s2)
36   return(fx)
37 }
38 grad=seq(60,200,by=0.1)
39 fx=fc(grad,m1,m2,s1,s2,pr)
40 #plot(grad,fx,type="l")
41 hist(x,ylim = range(fx),prob=T)
42 lines(grad,fx)
43 ##
```

22:32 (Top Level) ↕ R Script ↕

Aplicação 2

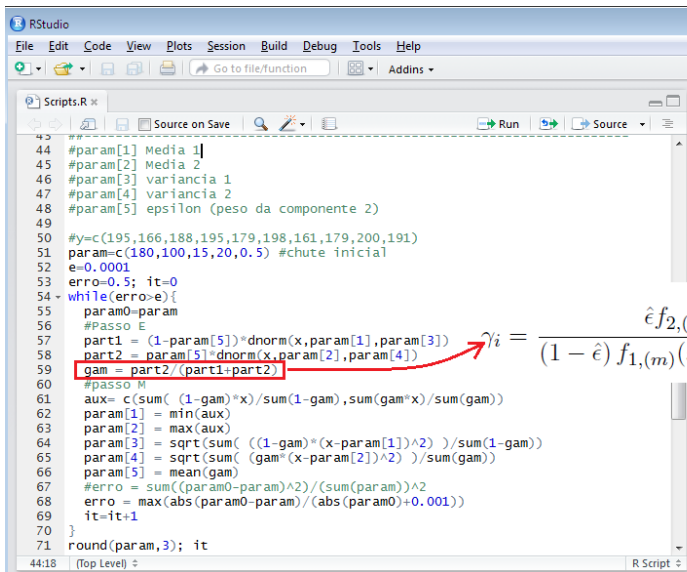


Aplicação 2





```
RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function
Scripts.R x
Source on Save Run Source
44 #param[1] Media 1
45 #param[2] Media 2
46 #param[3] variancia 1
47 #param[4] variancia 2
48 #param[5] epsilon (peso da componente 2)
49
50 #y=c(195,166,188,195,179,198,161,179,200,191)
51 param=c(180,100,15,20,0.5) #chute inicial
52 e=0.0001
53 erro=0.5; it=0
54 while(erro>e){
55   param0=param
56   #Passo E
57   part1 = (1-param[5])*dnorm(x,param[1],param[3])
58   part2 = param[5]*dnorm(x,param[2],param[4])
59   gam = part2/(part1+part2)
60   #passo M
61   aux= c(sum( (1-gam)*x)/sum(1-gam),sum(gam*x)/sum(gam))
62   param[1] = min(aux)
63   param[2] = max(aux)
64   param[3] = sqrt(sum( ((1-gam)*(x-param[1])^2 )/sum(1-gam)))
65   param[4] = sqrt(sum( (gam*(x-param[2])^2 )/sum(gam)))
66   param[5] = mean(gam)
67   #erro = sum((param0-param)^2)/(sum(param))^2
68   erro = max(abs(param0-param)/(abs(param0)+0.001))
69   it=it+1
70 }
71 round(param,3); it
72
42:15 (Top Level) R Script
```

Aplicação 2

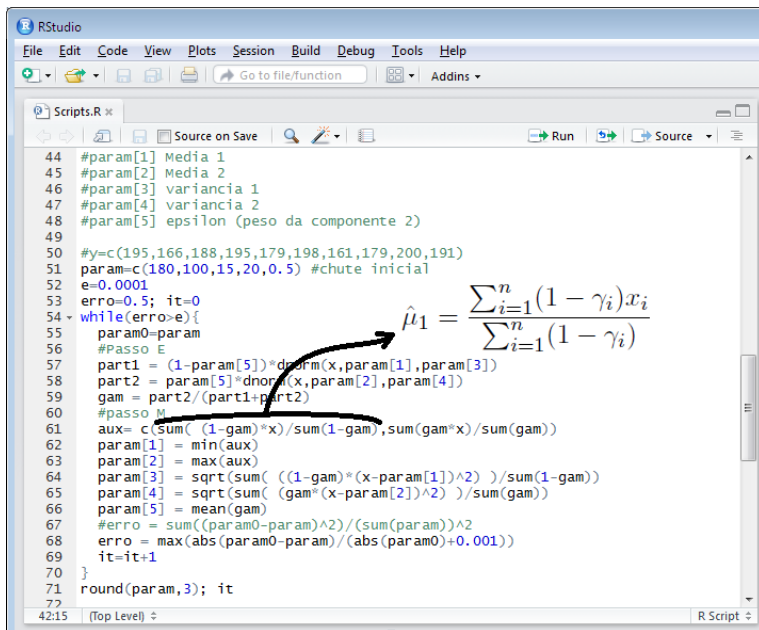


```
43 ##
44 #param[1] Media 1|
45 #param[2] Media 2
46 #param[3] variancia 1
47 #param[4] variancia 2
48 #param[5] epsilon (peso da componente 2)
49
50 #y=c(195,166,188,195,179,198,161,179,200,191)
51 param=c(180,100,15,20,0.5) #chute inicial
52 e=0.0001
53 erro=0.5; it=0
54 while(erro>e){
55   param0=param
56   #Passo E
57   part1 = (1-param[5])*dnorm(x,param[1],param[3])
58   part2 = param[5]*dnorm(x,param[2],param[4])
59   gam = part2/(part1+part2)
60   #passo M
61   aux= c(sum((1-gam)*x)/sum(1-gam),sum(gam*x)/sum(gam))
62   param[1] = min(aux)
63   param[2] = max(aux)
64   param[3] = sqrt(sum((1-gam)*(x-param[1])^2)/sum(1-gam))
65   param[4] = sqrt(sum(gam*(x-param[2])^2)/sum(gam))
66   param[5] = mean(gam)
67   #erro = sum((param0-param)^2)/(sum(param))^2
68   erro = max(abs(param0-param)/(abs(param0)+0.001))
69   it=it+1
70 }
71 round(param,3); it
```

44:18 (Top Level)  R Script 

$$\gamma_i = \frac{\hat{e}f_{2,(m)}(x_i)}{(1 - \hat{e})f_{1,(m)}(x_i) + \hat{e}f_{2,(m)}(x_i)}$$

Aplicação 2



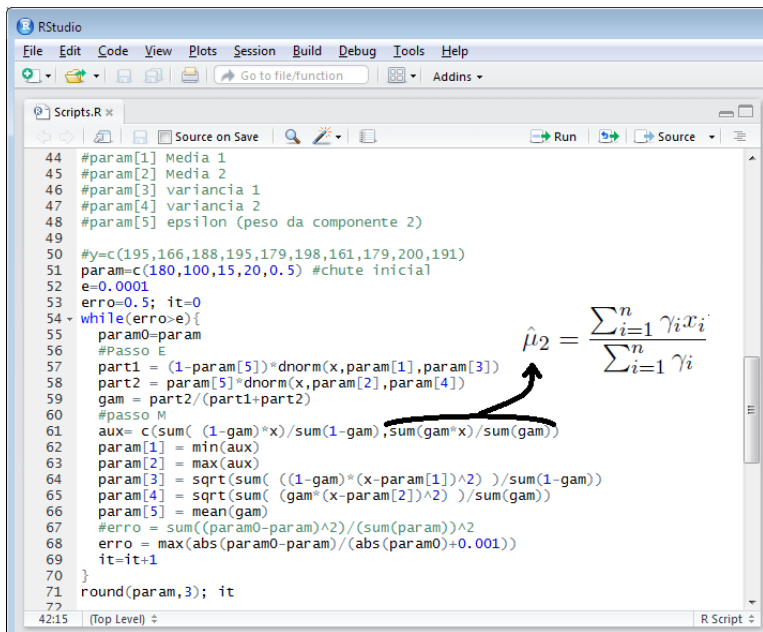
```
44 #param[1] Media 1
45 #param[2] Media 2
46 #param[3] variancia 1
47 #param[4] variancia 2
48 #param[5] epsilon (peso da componente 2)
49
50 #y=c(195,166,188,195,179,198,161,179,200,191)
51 param=c(180,100,15,20,0.5) #chute inicial
52 e=0.0001
53 erro=0.5; it=0
54 while(erro>e){
55   param0=param
56   #Passo E
57   part1 = (1-param[5])*dnorm(x,param[1],param[3])
58   part2 = param[5]*dnorm(x,param[2],param[4])
59   gam = part2/(part1+part2)
60   #passo M
61   aux= c(sum( (1-gam)*x)/sum(1-gam),sum(gam*x)/sum(gam))
62   param[1] = min(aux)
63   param[2] = max(aux)
64   param[3] = sqrt(sum( ((1-gam)*(x-param[1])^2 )/sum(1-gam))
65   param[4] = sqrt(sum( (gam*(x-param[2])^2 )/sum(gam))
66   param[5] = mean(gam)
67   #erro = sum((param0-param)^2)/(sum(param))^2
68   erro = max(abs(param0-param)/(abs(param0)+0.001))
69   it=it+1
70 }
71 round(param,3); it
72
```

Handwritten arrow pointing from line 59 (`gam = part2/(part1+part2)`) to the formula:

$$\hat{\mu}_1 = \frac{\sum_{i=1}^n (1 - \gamma_i) x_i}{\sum_{i=1}^n (1 - \gamma_i)}$$

42:15 (Top Level) ↕ R Script ↕

Aplicação 2

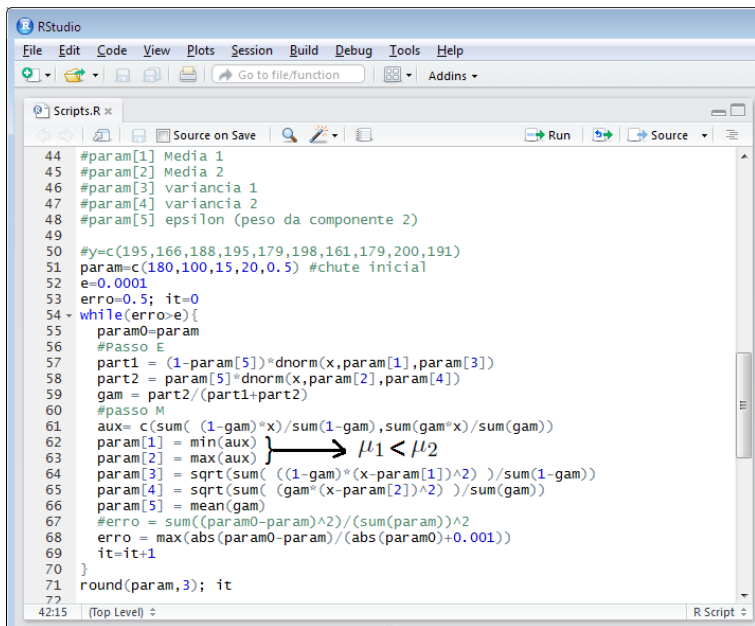


```
44 #param[1] Media 1
45 #param[2] Media 2
46 #param[3] variancia 1
47 #param[4] variancia 2
48 #param[5] epsilon (peso da componente 2)
49
50 #y=c(195,166,188,195,179,198,161,179,200,191)
51 param=c(180,100,15,20,0.5) #chute inicial
52 e=0.0001
53 erro=0.5; it=0
54 while(erro>e){
55   param0=param
56   #Passo E
57   part1 = (1-param[5])*dnorm(x,param[1],param[3])
58   part2 = param[5]*dnorm(x,param[2],param[4])
59   gam = part2/(part1+part2)
60   #passo M
61   aux= c(sum( (1-gam)*x)/sum(1-gam),sum(gam*x)/sum(gam))
62   param[1] = min(aux)
63   param[2] = max(aux)
64   param[3] = sqrt(sum( ((1-gam)*(x-param[1])^2 )/sum(1-gam)))
65   param[4] = sqrt(sum( (gam*(x-param[2])^2 )/sum(gam)))
66   param[5] = mean(gam)
67   #erro = sum((param0-param)^2)/(sum(param))^2
68   erro = max(abs(param0-param)/(abs(param0)+0.001))
69   it=it+1
70 }
71 round(param,3); it
72
```

$$\hat{\mu}_2 = \frac{\sum_{i=1}^n \gamma_i x_i}{\sum_{i=1}^n \gamma_i}$$

42:15 (Top Level) R Script

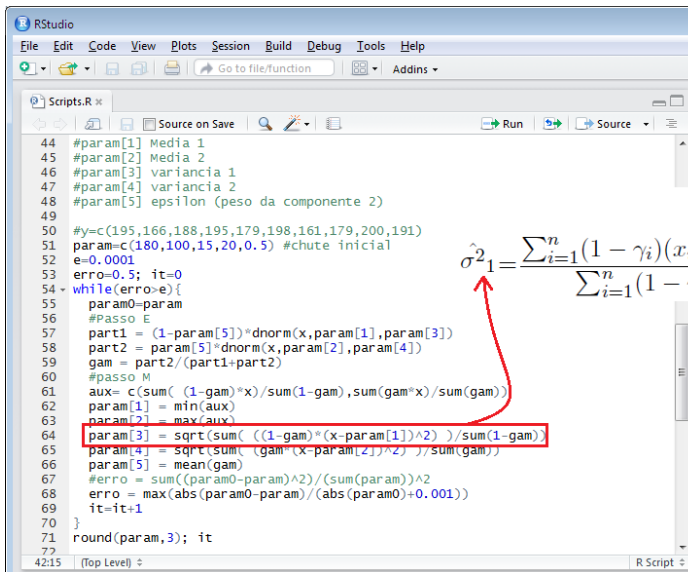
Aplicação 2



```
44 #param[1] Media 1
45 #param[2] Media 2
46 #param[3] variancia 1
47 #param[4] variancia 2
48 #param[5] epsilon (peso da componente 2)
49
50 #y=c(195,166,188,195,179,198,161,179,200,191)
51 param=c(180,100,15,20,0.5) #chute inicial
52 e=0.0001
53 erro=0.5; it=0
54 while(erro>e){
55   param0=param
56   #Passo E
57   part1 = (1-param[5])*dnorm(x,param[1],param[3])
58   part2 = param[5]*dnorm(x,param[2],param[4])
59   gam = part2/(part1+part2)
60   #passo M
61   aux= c(sum( (1-gam)*x)/sum(1-gam),sum(gam*x)/sum(gam))
62   param[1] = min(aux)
63   param[2] = max(aux) } →  $\mu_1 < \mu_2$ 
64   param[3] = sqrt(sum( ((1-gam)*(x-param[1])^2 )/sum(1-gam)))
65   param[4] = sqrt(sum( (gam*(x-param[2])^2 )/sum(gam)))
66   param[5] = mean(gam)
67   #erro = sum((param0-param)^2)/(sum(param))^2
68   erro = max(abs(param0-param)/(abs(param0)+0.001))
69   it=it+1
70 }
71 round(param,3); it
72
```

42:15 (Top Level) ↕ R Script ↕

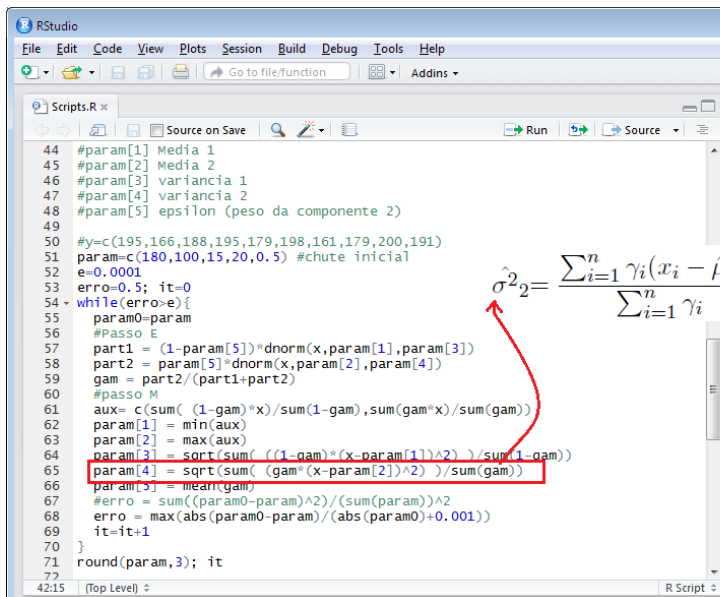
Aplicação 2



```
44 #param[1] Media 1
45 #param[2] Media 2
46 #param[3] variancia 1
47 #param[4] variancia 2
48 #param[5] epsilon (peso da componente 2)
49
50 #y=c(195,166,188,195,179,198,161,179,200,191)
51 param=c(180,100,15,20,0.5) #chute inicial
52 e=0.0001
53 erro=0.5; it=0
54 while(erro>e){
55   param0=param
56   #Passo E
57   part1 = (1-param[5])*dnorm(x,param[1],param[3])
58   part2 = param[5]*dnorm(x,param[2],param[4])
59   gam = part2/(part1+part2)
60   #passo M
61   aux= c(sum((1-gam)*x)/sum(1-gam),sum(gam*x)/sum(gam))
62   param[1] = min(aux)
63   param[2] = max(aux)
64   param[3] = sqrt(sum(((1-gam)*(x-param[1])^2))/sum(1-gam))
65   param[4] = sqrt(sum((gam*(x-param[2])^2))/sum(gam))
66   param[5] = mean(gam)
67   #erro = sum((param0-param)^2)/(sum(param))^2
68   erro = max(abs(param0-param)/(abs(param0)+0.001))
69   it=it+1
70 }
71 round(param,3); it
72
42:15 (Top Level) R Script
```

$$\hat{\sigma}_1^2 = \frac{\sum_{i=1}^n (1 - \gamma_i)(x_i - \hat{\mu}_1)^2}{\sum_{i=1}^n (1 - \gamma_i)}$$

Aplicação 2

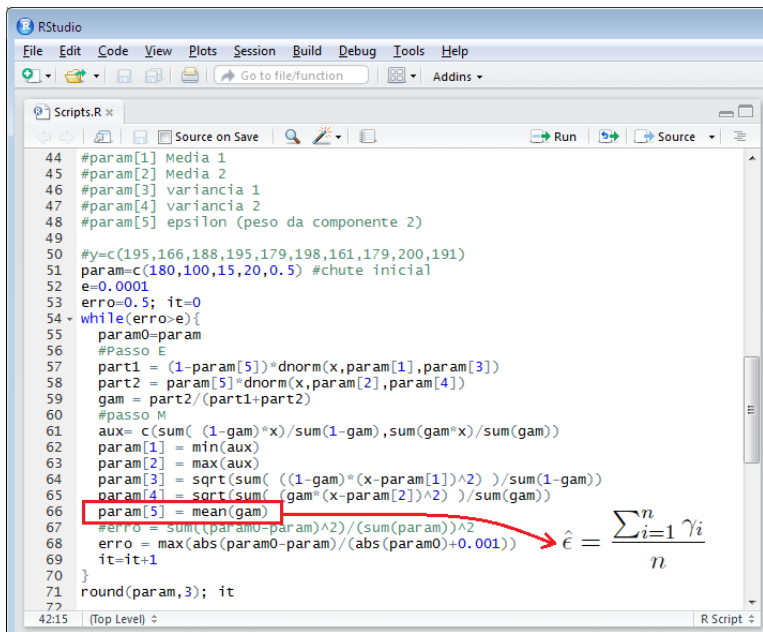


```
44 #param[1] Media 1
45 #param[2] Media 2
46 #param[3] variancia 1
47 #param[4] variancia 2
48 #param[5] epsilon (peso da componente 2)
49
50 #y=c(195,166,188,195,179,198,161,179,200,191)
51 param=c(180,100,15,20,0.5) #chute inicial
52 e=0.0001
53 erro=0.5; it=0
54 while(erro>e){
55   param0=param
56   #Passo E
57   part1 = (1-param[5])*dnorm(x,param[1],param[3])
58   part2 = param[5]*dnorm(x,param[2],param[4])
59   gam = part2/(part1+part2)
60   #passo M
61   aux= c(sum( (1-gam)*x)/sum(1-gam),sum(gam*x)/sum(gam))
62   param[1] = min(aux)
63   param[2] = max(aux)
64   param[3] = sqrt(sum( ((1-gam)*(x-param[1])^2 )/sum(1-gam))
65   param[4] = sqrt(sum( (gam*(x-param[2])^2 )/sum(gam))
66   param[5] = mean(gam)
67   #erro = sum((param0-param)^2)/(sum(param))^2
68   erro = max(abs(param0-param)/(abs(param0)+0.001))
69   it=it+1
70 }
71 round(param,3); it
72
```

$\hat{\sigma}_2^2 = \frac{\sum_{i=1}^n \gamma_i (x_i - \hat{\mu}_2)^2}{\sum_{i=1}^n \gamma_i}$

42:15 (Top Level) R Script

Aplicação 2

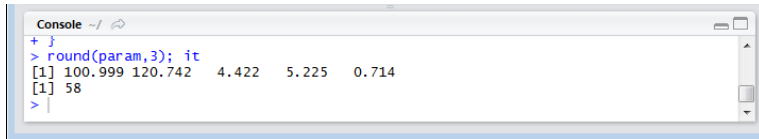



```
44 #param[1] Media 1
45 #param[2] Media 2
46 #param[3] variancia 1
47 #param[4] variancia 2
48 #param[5] epsilon (peso da componente 2)
49
50 #y=c(195,166,188,195,179,198,161,179,200,191)
51 param=c(180,100,15,20,0.5) #chute inicial
52 e=0.0001
53 erro=0.5; it=0
54 while(erro>e){
55   param0=param
56   #Passo E
57   part1 = (1-param[5])*dnorm(x,param[1],param[3])
58   part2 = param[5]*dnorm(x,param[2],param[4])
59   gam = part2/(part1+part2)
60   #passo M
61   aux= c(sum( (1-gam)*x)/sum(1-gam),sum(gam*x)/sum(gam))
62   param[1] = min(aux)
63   param[2] = max(aux)
64   param[3] = sqrt(sum( ((1-gam)*(x-param[1])^2 )/sum(1-gam)))
65   param[4] = sqrt(sum( (gam*(x-param[2])^2 )/sum(gam)))
66   param[5] = mean(gam)
67   #erro = sum((param0-param)^2)/(sum(param)^2)
68   erro = max(abs(param0-param)/(abs(param0)+0.001))
69   it=it+1
70 }
71 round(param,3); it
72
```

42:15 (Top Level) ↕ R Script ↕

$$\hat{\epsilon} = \frac{\sum_{i=1}^n \gamma_i}{n}$$

Aplicação 2



```
Console ~/ 
+ }
> round(param,3); it
[1] 100.999 120.742 4.422 5.225 0.714
[1] 58
> |
```

Aplicação 3

Vamos considerar nesta aplicação um modelo dicotômico sobre o enfoque de variáveis latentes.

Nesse exemplo existe uma variável latente contínua (Z) que não observamos diretamente e portanto não podemos mensurar.

O que observamos de fato é um indicador da variável não observada ou latente.

Sob este enfoque considere o modelo probit:

$$Y_i = \mathbf{1}_{[Z_i > 0]},$$

$$Z_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \epsilon_i,$$

$$\epsilon_i \sim N(0, 1)$$

Induzindo a variável latente conseguimos modelar a probabilidade.

$$P(Y_i = 1) = P(Z_i > 0) = P(Z_i - X_i' \beta > -X_i' \beta) = \Phi(X_i' \beta),$$

podemos escrever a distribuição de Y_i sob forma de funções indicadoras:

$$f(y_i | z_i) = \mathbf{1}_{[Y_i=1]} \mathbf{1}_{[Z_i>0]} + \mathbf{1}_{[Y_i=0]} \mathbf{1}_{[Z_i \leq 0]}.$$

Veja que a densidade de $[Z_i | Y_i, \beta]$ é:

$$f(z_i | y_i, \beta) \propto f(y_i | z_i) f(z_i | \beta),$$

$$f(z_i | y_i, \beta) \propto \{ \mathbf{1}_{[Y_i=1]} \mathbf{1}_{[Z_i>0]} + \mathbf{1}_{[Y_i=0]} \mathbf{1}_{[Z_i \leq 0]} \} N(X_i' \beta, 1)$$

$$f(z_i | y_i, \beta) = \begin{cases} N(X_i' \beta, 1) \mathbf{1}_{[Z_i>0]}, & \text{se } Y_i = 1 \\ N(X_i' \beta, 1) \mathbf{1}_{[Z_i \leq 0]}, & \text{se } Y_i = 0. \end{cases}$$

Aplicação 3

Escrevemos a verossimilhança completa da seguinte forma:

$$L^c(\beta \mid \mathbf{y}, \mathbf{z}) = \prod_{i=1}^n f(y_i \mid z_i) f(z_i \mid \beta)$$

$$\ln L^c(\beta \mid \mathbf{y}, \mathbf{z}) = \sum_{i=1}^n [\ln f(y_i \mid z_i) + \ln f(z_i \mid \beta)]$$

$$\ln L^c(\beta \mid \mathbf{y}, \mathbf{z}) \propto \sum_{i=1}^n \ln f(z_i \mid \beta)$$

$$\ln L^c(\beta \mid \mathbf{y}, \mathbf{z}) \propto -\frac{1}{2} \sum_{i=1}^n (Z_i - X_i' \beta)^2 = -\frac{1}{2} (\mathbf{Z} - \mathbf{X}\beta)' (\mathbf{Z} - \mathbf{X}\beta)$$

$$\ln L^c(\beta \mid \mathbf{y}, \mathbf{z}) \propto -\frac{1}{2} (\mathbf{Z}'\mathbf{Z} - 2\beta'\mathbf{X}'\mathbf{Z} + \beta'\mathbf{X}'\mathbf{X}\beta)$$

$$\underbrace{E_{\hat{\theta}^{(m)}} [\ln L^c(\beta \mid \mathbf{y}, \mathbf{z})]}_{Q(\beta \mid \beta^{(m)}, \mathbf{X})} \propto -\frac{1}{2} (E_{\hat{\theta}^{(m)}} [\mathbf{Z}'\mathbf{Z}] - 2\beta'\mathbf{X}'E_{\hat{\theta}^{(m)}} [\mathbf{Z}] + \beta'\mathbf{X}'\mathbf{X}\beta)$$

$$\frac{\partial}{\partial \beta} Q(\beta \mid \beta^{(m)}, \mathbf{X}) = -\frac{1}{2} (0 - 2\mathbf{X}'E_{\hat{\theta}^{(m)}} [\mathbf{Z}] + 2\mathbf{X}'\mathbf{X}\beta)$$

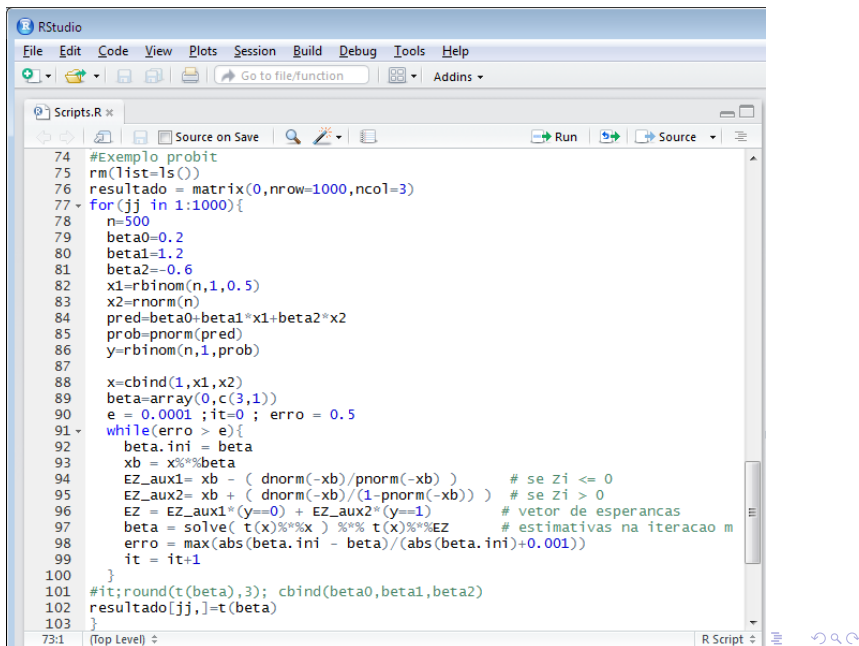
Aplicação 3

$$2\mathbf{X}'\mathbf{X}\beta = 2\mathbf{X}'E_{\hat{\theta}^{(m)}}[\mathbf{Z}]$$

$$\beta_{\text{est}} = (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'E_{\hat{\theta}^{(m)}}[\mathbf{Z}]$$

$$E\left[Z_i \mid \theta^{(m)}, \mathbf{y}\right] = \begin{cases} X_i'\beta - \frac{\phi(-X_i\beta)}{\Phi(-X_i\beta)}, & \text{se } Z_i \leq 0 \\ X_i'\beta + \frac{\phi(-X_i\beta)}{1 - \Phi(-X_i\beta)}, & \text{se } Z_i > 0. \end{cases}$$

Aplicação 3

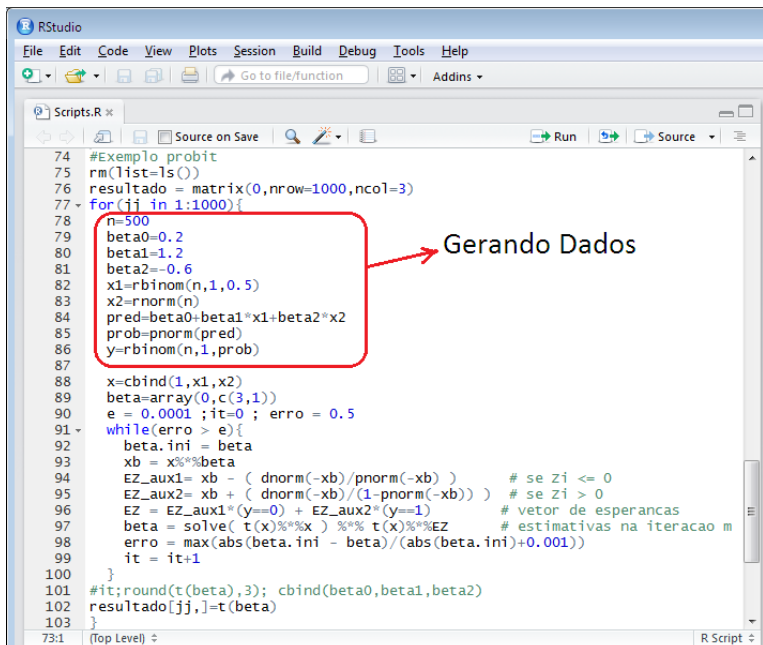


The screenshot shows the RStudio interface with a script editor open. The script defines a function to estimate a probit model using the Newton-Raphson method. It includes comments in Portuguese explaining the steps: generating random data, calculating the log-likelihood, and iteratively updating the parameter estimates until convergence.

```
74 #Exemplo probit
75 rm(list=ls())
76 resultado = matrix(0,nrow=1000,ncol=3)
77 for(jj in 1:1000){
78   n=500
79   beta0=0.2
80   beta1=1.2
81   beta2=-0.6
82   x1=rbinom(n,1,0.5)
83   x2=rnorm(n)
84   pred=beta0+beta1*x1+beta2*x2
85   prob=pnorm(pred)
86   y=rbinom(n,1,prob)
87
88   x=cbind(1,x1,x2)
89   beta=array(0,c(3,1))
90   e = 0.0001 ; it=0 ; erro = 0.5
91   while(erro > e){
92     beta.ini = beta
93     xb = x%%beta
94     EZ_aux1= xb - ( dnorm(-xb)/pnorm(-xb) )      # se Zi <= 0
95     EZ_aux2= xb + ( dnorm(-xb)/(1-pnorm(-xb)) )  # se Zi > 0
96     EZ = EZ_aux1*(y==0) + EZ_aux2*(y==1)         # vetor de esperancas
97     beta = solve( t(x)%%x ) %% t(x)%%EZ          # estimativas na iteracao m
98     erro = max(abs(beta.ini - beta)/(abs(beta.ini)+0.001))
99     it = it+1
100   }
101   #it;round(t(beta),3); cbind(beta0,beta1,beta2)
102   resultado[jj,]=t(beta)
103 }
73:1 (Top Level) ↕
```

R Script

Aplicação 3

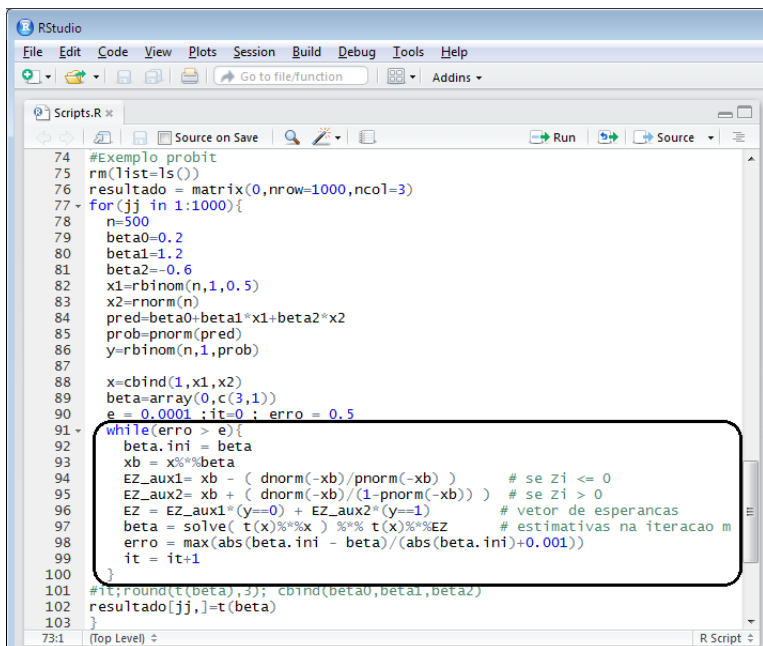


```
74 #Exemplo probit
75 rm(list=ls())
76 resultado = matrix(0,nrow=1000,ncol=3)
77 for(jj in 1:1000){
78   n=500
79   beta0=0.2
80   beta1=1.2
81   beta2=-0.6
82   x1=rbinom(n,1,0.5)
83   x2=rnorm(n)
84   pred=beta0+beta1*x1+beta2*x2
85   prob=pnorm(pred)
86   y=rbinom(n,1,prob)
87
88   x=cbind(1,x1,x2)
89   beta=array(0,c(3,1))
90   e = 0.0001 ; it=0 ; erro = 0.5
91   while(erro > e){
92     beta.ini = beta
93     xb = x%%beta
94     EZ_aux1= xb - ( dnorm(-xb)/pnorm(-xb) )      # se Zi <= 0
95     EZ_aux2= xb + ( dnorm(-xb)/(1-pnorm(-xb)) )  # se Zi > 0
96     EZ = EZ_aux1*(y==0) + EZ_aux2*(y==1)         # vetor de esperancas
97     beta = solve( t(x)%%x ) %% t(x)%%EZ          # estimativas na iteracao m
98     erro = max(abs(beta.ini - beta)/(abs(beta.ini)+0.001))
99     it = it+1
100   }
101   #it;round(t(beta),3); cbind(beta0,beta1,beta2)
102   resultado[jj,]=t(beta)
103 }
```

Gerando Dados

73:1 (Top Level) ↕ R Script

Aplicação 3



```
74 #Exemplo probit
75 rm(list=ls())
76 resultado = matrix(0,nrow=1000,ncol=3)
77 for(jj in 1:1000){
78   n=500
79   beta0=0.2
80   beta1=1.2
81   beta2=-0.6
82   x1=rbinom(n,1,0.5)
83   x2=rnorm(n)
84   pred=beta0+beta1*x1+beta2*x2
85   prob=pnorm(pred)
86   y=rbinom(n,1,prob)
87
88   x=cbind(1,x1,x2)
89   beta=array(0,c(3,1))
90   e = 0.0001 ; it=0 ; erro = 0.5
91   while(erro > e){
92     beta.ini = beta
93     xb = x%%beta
94     EZ_aux1= xb - ( dnorm(-xb)/pnorm(-xb) )      # se Zi <= 0
95     EZ_aux2= xb + ( dnorm(-xb)/(1-pnorm(-xb)) )  # se Zi > 0
96     EZ = EZ_aux1*(y==0) + EZ_aux2*(y==1)         # vetor de esperanças
97     beta = solve( t(x)%%x ) %% t(x)%%EZ          # estimativas na iteracao m
98     erro = max(abs(beta.ini - beta)/(abs(beta.ini)+0.001))
99     it = it+1
100  }
101  #it;round(t(beta),3); cbind(beta0,beta1,beta2)
102  resultado[jj,]=t(beta)
103 }
```

73:1 (Top Level) ↕ R Script ↕

Aplicação 3

RStudio interface showing a script for a probit model. The script includes data generation, parameter initialization, and an iterative estimation loop. A red box highlights the calculation of EZ_aux1 , and another red box highlights the mathematical formula for the conditional expectation of Z_i given $\theta^{(m)}$ and y .

```
74 #Exemplo probit
75 rm(list=ls())
76 resultado = matrix(0,nrow=1000,ncol=3)
77 for(jj in 1:1000){
78   n=500
79   beta0=0.2
80   beta1=1.2
81   beta2=-0.6
82   x1=rbinom(n,1,0.5)
83   x2=rnorm(n)
84   pred=beta0+beta1*x1+beta2*x2
85   prob=pnorm(pred)
86   y=rbinom(n,1,prob)
87
88   x=cbind(1,x1,x2)
89   beta=array(0,c(3,1))
90   e = 0.0001 ; it=0 ; erro = 0.5
91   while(erro > e){
92     beta.ini = beta
93     xb = x%%beta
94     EZ_aux1= xb - ( dnorm(-xb)/pnorm(-xb) ) # se Zi <= 0
95     EZ_aux2= xb + ( dnorm(-xb)/(1-pnorm(-xb)) ) # se Zi > 0
96     EZ = EZ_aux1*(y==0) + EZ_aux2*(y==1) # vetor de esperancas
97     beta = solve( t(x)%%x ) %% t(x)%%EZ # estimativas na iteracao m
98     erro = max(abs(beta.ini - beta)/(abs(beta.ini)+0.001))
99     it = it+1
100   }
101   #it;round(t(beta),3); cbind(beta0,beta1,beta2)
102   resultado[jj,]=t(beta)
103 }
```

73:1 (Top Level) ↕ R Script

Equation highlighted in red box:

$$E \left[Z_i \mid \theta^{(m)}, y \right] = X_i' \beta - \frac{\phi(-X_i \beta)}{\Phi(-X_i \beta)}$$

Aplicação 3

RStudio

File Edit Code View Plots Session Build Debug Tools Help

Go to file/function Addins

Scripts.R x

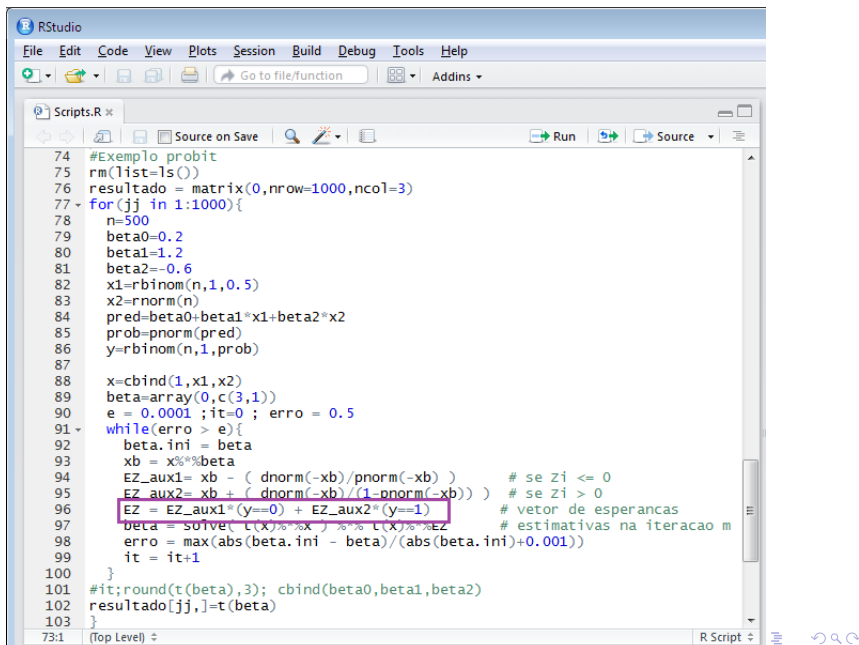
Source on Save Run Source

```
74 #Exemplo probit
75 rm(list=ls())
76 resultado = matrix(0,nrow=1000,ncol=3)
77 for(jj in 1:1000){
78   n=500
79   beta0=0.2
80   beta1=1.2
81   beta2=-0.6
82   x1=rbinom(n,1,0.5)
83   x2=rnorm(n)
84   pred=beta0+beta1*x1+beta2*x2
85   prob=pnorm(pred)
86   y=rbinom(n,1,prob)
87
88   x=cbind(1,x1,x2)
89   beta=array(0,c(3,1))
90   e = 0.0001 ; it=0 ; erro = 0.5
91   while(erro > e){
92     beta.ini = beta
93     xb = x%%beta
94     EZ_aux1= xb - ( dnorm(-xb)/pnorm(-xb) ) # se Zi <= 0
95     EZ_aux2= xb + ( dnorm(-xb)/(1-pnorm(-xb)) ) # se Zi > 0
96     EZ = EZ_aux1*(y==0) + EZ_aux2*(y==1) # vetor de esperanças
97     beta = solve( t(x)%%x ) %% t(x)%%EZ # estimativas na iteracao m
98     erro = max(abs(beta.ini - beta)/(abs(beta.ini)+0.001))
99     it = it+1
100   }
101   #it;round(t(beta),3); cbind(beta0,beta1,beta2)
102   resultado[jj,]=t(beta)
103 }
```

73:1 (Top Level) R Script

$$E[Z_i | \theta^{(m)}, y] = X_i' \beta + \frac{\phi(-X_i \beta)}{1 - \Phi(-X_i \beta)}$$

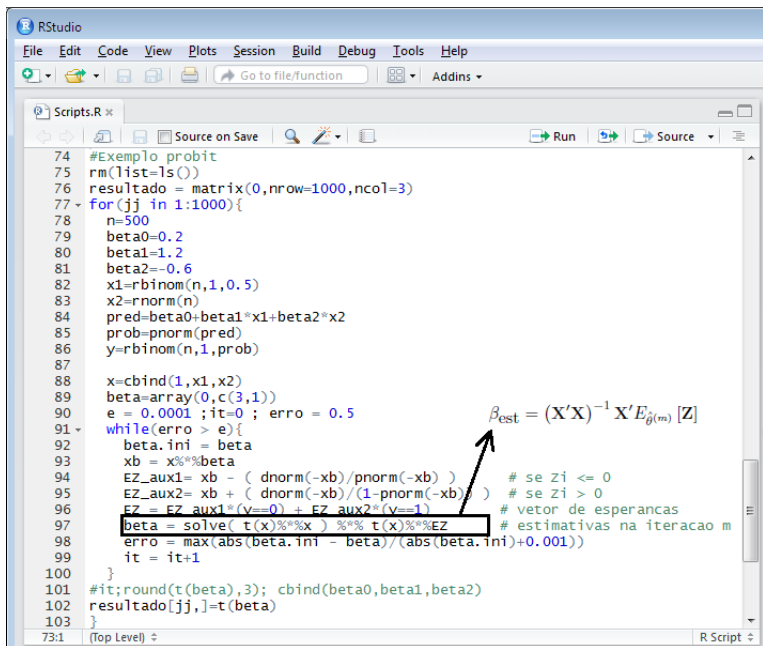
Aplicação 3



The image shows the RStudio interface with a script editor open. The script is a function to simulate a probit model. It includes comments in Portuguese and a loop to estimate the parameters. A red rectangle highlights the calculation of the expected values vector `EZ`.

```
74 #Exemplo probit
75 rm(list=ls())
76 resultado = matrix(0,nrow=1000,ncol=3)
77 for(jj in 1:1000){
78   n=500
79   beta0=0.2
80   beta1=1.2
81   beta2=-0.6
82   x1=rbinom(n,1,0.5)
83   x2=rnorm(n)
84   pred=beta0+beta1*x1+beta2*x2
85   prob=pnorm(pred)
86   y=rbinom(n,1,prob)
87
88   x=cbind(1,x1,x2)
89   beta=array(0,c(3,1))
90   e = 0.0001 ; it=0 ; erro = 0.5
91   while(erro > e){
92     beta.ini = beta
93     xb = x%%beta
94     EZ_aux1= xb - ( dnorm(-xb)/pnorm(-xb) ) # se Zi <= 0
95     EZ_aux2= xb + ( dnorm(-xb)/(1-pnorm(-xb)) ) # se Zi > 0
96     EZ = EZ_aux1*(y==0) + EZ_aux2*(y==1) # vetor de esperanças
97     beta = solve( t(x)%%x ) %% t(x)%%EZ # estimativas na iteracao m
98     erro = max(abs(beta.ini - beta)/(abs(beta.ini)+0.001))
99     it = it+1
100   }
101   #it;round(t(beta),3); cbind(beta0,beta1,beta2)
102   resultado[jj,]=t(beta)
103 }
73:1 (Top Level) ↕
```

Aplicação 3



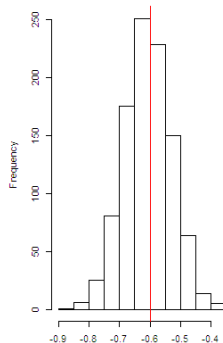
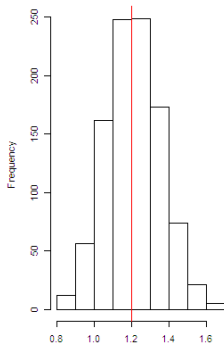
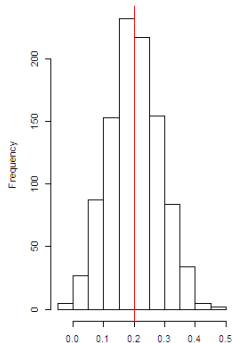
```
74 #Exemplo probit
75 rm(list=ls())
76 resultado = matrix(0,nrow=1000,ncol=3)
77 for(jj in 1:1000){
78   n=500
79   beta0=0.2
80   beta1=1.2
81   beta2=-0.6
82   x1=rbinom(n,1,0.5)
83   x2=rnorm(n)
84   pred=beta0+beta1*x1+beta2*x2
85   prob=pnorm(pred)
86   y=rbinom(n,1,prob)
87
88   x=cbind(1,x1,x2)
89   beta=array(0,c(3,1))
90   e = 0.0001 ; it=0 ; erro = 0.5
91   while(erro > e){
92     beta.ini = beta
93     xb = x%%beta
94     EZ_aux1= xb - ( dnorm(-xb)/pnorm(-xb) ) # se Zi <= 0
95     EZ_aux2= xb + ( dnorm(-xb)/(1-pnorm(-xb)) ) # se Zi > 0
96     EZ = EZ_aux1*(y==0) + EZ_aux2*(y==1) # vetor de esperanças
97     beta = solve( t(x)%%x ) %% t(x)%%EZ # estimativas na iteracao m
98     erro = max(abs(beta.ini - beta)/(abs(beta.ini)+0.001))
99     it = it+1
100   }
101   #it;round(t(beta),3); cbind(beta0,beta1,beta2)
102   resultado[jj,]=t(beta)
103 }
```

$\beta_{est} = (X'X)^{-1} X'E_{\hat{\theta}(m)} [Z]$

73:1 (Top Level) ↕ R Script ↕

Aplicação 3

```
Console ~ / ↺  
> mean(resultado[,1]);mean(resultado[,2]);mean(resultado[,2])  
[1] 0.200374  
[1] 1.21238  
[1] 1.21238  
> |
```



Veja que no **passo E**, na iteração $m + 1$ temos que calcular

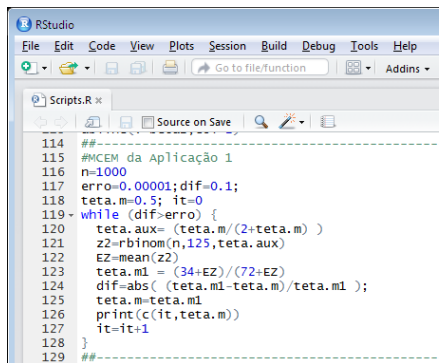
$$Q(\theta \mid \hat{\theta}^{(m)}, \mathbf{x}) = E_{\hat{\theta}^{(m)}} [\ln L^c(\theta \mid \mathbf{x}, \mathbf{z})]$$

E algumas vezes é difícil obter analiticamente o valor $E [Z_i \mid \theta^{(m)}, \mathbf{y}]$, mas sabemos gerar, via Monte Carlo, os valores da distribuição condicional de $[Z_i \mid \theta^{(m)}, \mathbf{y}]$.

Suponha que temos uma amostra $Z_i^{m,1}, Z_i^{m,2}, \dots, Z_i^{m,K}$ de $[Z_i \mid \theta^{(m)}, \mathbf{y}]$. Então aproxime $Q(\theta \mid \hat{\theta}^{(m)}, \mathbf{x})$ por:

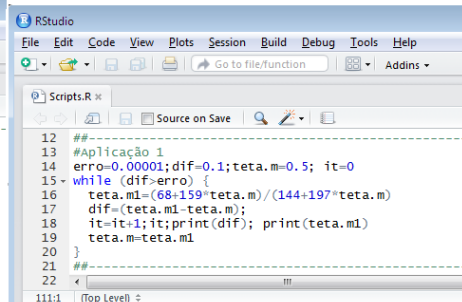
$$Q(\theta \mid \hat{\theta}^{(m)}, \mathbf{x}) \approx \frac{1}{K} \sum_{i=1}^K \{\ln L^c(\theta \mid \mathbf{x}, \mathbf{z})\}$$

MCEM



RStudio window showing R code for the MCEM algorithm. The code is in a script file named 'Scripts.R'. The code defines a function for the MCEM algorithm, starting with a comment '#MCEM da Aplicação 1'. It sets parameters: n=1000, erro=0.00001, dif=0.1, teta.m=0.5, and it=0. A while loop is used to iterate until the error is less than 0.00001. Inside the loop, a random variable z2 is generated from a binomial distribution with parameters n and teta.m/(2+teta.m). The expected value EZ is calculated as (34+EZ)/(72+EZ). The difference dif is calculated as the absolute value of (teta.m1 - teta.m)/teta.m1. The difference is printed, and teta.m is updated to teta.m1. The iteration counter it is incremented by 1. The code ends with a comment '##-----'.

```
114 ##-----
115 #MCEM da Aplicação 1
116 n=1000
117 erro=0.00001; dif=0.1;
118 teta.m=0.5; it=0
119 while (dif>erro) {
120   teta.aux= (teta.m/(2+teta.m) )
121   z2=rbinom(n,125,teta.aux)
122   EZ=mean(z2)
123   teta.m1 = (34+EZ)/(72+EZ)
124   dif=abs( (teta.m1-teta.m)/teta.m1 );
125   teta.m=teta.m1
126   print(c(it,teta.m))
127   it=it+1
128 }
129 ##-----
```



RStudio window showing R code for the MCEM algorithm. The code is in a script file named 'Scripts.R'. The code defines a function for the MCEM algorithm, starting with a comment '#Aplicação 1'. It sets parameters: erro=0.00001, dif=0.1, teta.m=0.5, and it=0. A while loop is used to iterate until the error is less than 0.00001. Inside the loop, teta.m1 is calculated as (68+159*teta.m)/(144+197*teta.m). The difference dif is calculated as (teta.m1 - teta.m). The difference is printed, and teta.m is updated to teta.m1. The iteration counter it is incremented by 1. The code ends with a comment '##-----'.

```
12 ##-----
13 #Aplicação 1
14 erro=0.00001; dif=0.1; teta.m=0.5; it=0
15 while (dif>erro) {
16   teta.m1=(68+159*teta.m)/(144+197*teta.m)
17   dif=(teta.m1-teta.m);
18   it=it+1; it; print(dif); print(teta.m1)
19   teta.m=teta.m1
20 }
21 ##-----
22
```

```

RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function Addins
Scripts.R x
Source on Save Run Source
130 #Exemplo probit
131 rm(list=ls())
132 library(msm)
133 n=500
134 m=c(rep(10,30),rep(50,20),rep(100,20),rep(500,15),rep(1000,15))
135 r=100
136 beta0=0.2
137 beta1=1.2
138 beta2=-0.6
139 x1=rbinom(n,1,0.5)
140 x2=rnorm(n)
141 pred=beta0+beta1*x1+beta2*x2
142 prob=pnorm(pred)
143 y=rbinom(n,1,prob)
144 x=cbind(1,x1,x2)
145 beta=array(0,c(3,1))
146 e = 0.0001 ; it=0 ; erro = 0.5 ; result=array(0,c(r,3))
147 # while(erro > e){
148 for(k in 1:r){
149   beta.ini = beta
150   xb = x%%beta
151   Z_aux1= t(sapply(1:n, function(i){rtnorm(m[k], mean=xb[i], sd=1, lower=-Inf, upper=0)})) # se zi <= 0
152   Z_aux2= t(sapply(1:n, function(i){rtnorm(m[k], mean=xb[i], sd=1, lower=0, upper= Inf)})) # se zi > 0
153   EZ_aux1=rowMeans(Z_aux1)
154   EZ_aux2=rowMeans(Z_aux2)
155   EZ = EZ_aux1*(y==0) + EZ_aux2*(y==1) # vetor de esperancas
156   beta = solve( t(x)%%x ) %% t(x)%%EZ # estimativas na iteracao m
157   #erro = max(abs(beta.ini - beta)/(abs(beta.ini)+0.001))
158   result[k,]=t(beta) ; #print(k)
159 }
160 # it;round(t(beta),3); cbind(beta0,beta1,beta2)
161
129:1 (Top Level)
R Script

```

Obrigado!

Referências I

Boyles, R. A., On the Convergence of the EM algorithm. J. Roy. Statist. Soc. Ser. B 45, 47-50, 1983.

Casella, G., Berger, R. L. Inferência estatística - tradução da 2a edição norte americana. Centage Learning, 2010. Página 147-151, 291-294, 329.

Dempster, A. P., Laird, N.M. e Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, série B, 39, 1-22.

McLachlan, G. J. and Krishnan, T. (1997), The EM Algorithm and Extensions, New York: John Wiley Sons.

Rao, C. R. (1973), Linear Statistical Inference and Its Applications, 2nd Ed., New York: John Wiley Sons.

Wu, C. F. J. (1983) On the Convergence of the EM algorithm. Ann. Statist. 11, 95-113.