

Exercícios sobre Verossimilhança - V

1. Verossimilhanças para Distribuição Normal

Seja $X \sim N(\mu, \sigma^2)$ e considere escrever funções para obter gráficos da função de log-verossimilhança para os parâmetros $\theta = (\mu, \sigma)$ para as seguintes situações:

- log-verossimilhança conjunta para o par de parâmetros,
- log-verossimilhança para cada um considerando o outro fixo em um valor arbitrário,
- perfis de verossimilhança para cada parâmetro.

A função de log-verossimilhança é dada por:

$$\begin{aligned} l(\mu, \sigma) &= -\frac{n}{2} \left[\log(2\pi) + \log(\sigma^2) + \frac{\sum_i (x_i - \mu)^2}{n\sigma^2} \right] \\ &= -\frac{n}{2} \left[\log(2\pi) + \log(\sigma^2) + \frac{\sum_i x_i^2 - n\mu(2\bar{x} - \mu)}{n\sigma^2} \right] \end{aligned} \quad (1)$$

Os estimadores de máxima verossimilhança são:

$$\hat{\mu} = \bar{x} \quad (2)$$

$$\hat{\sigma} = \sqrt{\frac{\sum_i (x_i - \bar{x})^2}{n}} \quad (3)$$

Por vezes apenas um dos parâmetros é de interesse, sendo o outro então considerando *nuisance*. Diferentes abordagens são propostas para obter uma função de verossimilhança apenas para o parâmetro de interesse. Discutimos duas delas a seguir.

Se um dos parâmetros for fixado em um valor arbitrário a função passa a ser de apenas um parâmetro. Além disto, se o valor corresponder a uma estimativa (não necessariamente de máxima verossimilhança) deste parâmetro fixado, tem-se uma função de *log-verossimilhança estimada*. Alguns autores utilizam o termo *pseudo-verossimilhança* que preferimos não adotar por poder ter outras conotações ou definições. Neste caso obteremos a partir de função de verossimilhança:

$$l_e(\mu) = l(\mu; \sigma = \hat{\sigma}) \quad (4)$$

$$l_e(\sigma) = l(\sigma; \mu = \hat{\mu}) \quad (5)$$

A proposta anterior ignora a incerteza associada à estimação do parâmetro *nuisance*. Uma alternativa é construir a função verossimilhança onde o parâmetro *nuisance* é reestimado para cada valor do

parâmetro de interesse. As expressões dos perfis de verossimilhança são dadas por:

$$Pl(\mu) = l(\mu; \sigma = \hat{\sigma}_\mu) = -\frac{n}{2} \left[\log(2\pi) + \log(\hat{\sigma}_\mu^2) + \frac{\sum_i (x_i - \mu)^2}{n\hat{\sigma}_\mu^2} \right] \quad (6)$$

$$= -\frac{n}{2} \left[\log(2\pi) + \log\left(\sum_i (x_i - \mu)^2\right) - \log(n) + 1 \right]$$

$$= -\frac{n}{2} \left[\log(2\pi) + \log\left(\sum_i x_i^2 - n\mu(2\bar{x} - \mu)\right) - \log(n) + 1 \right]$$

$$Pl(\sigma) = l(\sigma; \mu = \hat{\mu}_\sigma) = -\frac{n}{2} \left[\log(2\pi) + 2\log(\sigma) + \frac{\sum_i (x_i - \bar{\mu})^2}{n\sigma^2} \right] \quad (7)$$

$$= -\frac{n}{2} \left[\log(2\pi) + 2\log(\sigma) + \frac{\sum_i x_i^2 - n\mu(2\bar{x} - \mu)}{n\sigma^2} \right]$$

Neste caso particular de parâmetros da distribuição normal note-se que $\hat{\mu}_\sigma = \hat{\mu} = \bar{x}$ para qualquer valor de σ . Portanto a verossimilhança perfilhada de σ coincide com o que se obtêm com a verossimilhança fixando o valor de μ no seu estimador de máxima verossimilhança. O mesmo já não ocorre para μ uma vez que $\hat{\sigma}_\mu$ depende do valor de μ .

Algumas formas alternativas de representar funções de (log)verossimilhança são:

- verossimilhança relativa: $L_R(\theta) = \frac{L(\theta)}{L(\hat{\theta})}$
- log-verossimilhança relativa: $l_R(\theta) = l(\theta) - l(\hat{\theta})$
- deviance: $D(\theta) = -2[l(\theta) - l(\hat{\theta})]$

Estas formas são convenientes pois "padronizam" de alguma forma os valores da verossimilhança que em geral não tem significado especial e permitem a sobreposição e comparação de curvas. A verossimilhança relativa possui valor máximo igual a 1 (um) no ponto correspondente ao estimador de máxima verossimilhança e portanto todos os demais valores pode ser interpretados como uma fração (percentagem) do valor máximo. Vale lembrar que os níveis de 0.04 (4%) e 0.15 (15%) correspondem aproximadamente a intervalos de 99% e 95%, respectivamente, *em funções de um único parâmetro*. A log-verossimilhança relativa tem a mesma idéia na escala logarítmica e portanto possui valor máximo igual a 0 (zero) e a exponencial dos valores da função são interpretados como frações do máximo. A deviance "inverte" o gráfico de forma que o estimador de máxima verossimilhança corresponde ao valor mínimo da função e a multiplicação pela constante 2 (dois) permite que os valores da função sejam comparáveis a quantis da distribuição assintótica χ^2 .

Para ilustrar a obtenção das funções vamos simular dados e definir faixas de valores para as quais serão obtidos os gráficos das funções. As estimativas de máxima verossimilhança pode ser calculadas pois

neste caso temos expressão analítica para os estimadores.

```
> set.seed(4132); x <- round(rnorm(15, m=70, sd=3), dig=1)
> x

[1] 67.2 67.5 67.6 67.4 72.4 69.4 69.8 70.5 75.9 75.5 70.7 67.2
[13] 64.7 67.2 74.0

> mu.vals <- seq(66, 74, len=201)
> sig.vals <- seq(1.8, 7.5, len=201)
> musig.vals <- expand.grid(mu.vals, sig.vals)
> mle <- c(mean(x), sqrt(var(x) * (length(x)-1)/length(x)))
> mle

[1] 69.800000  3.242838
```

Uma primeira alternativa para obter as funções de gráficos pedidos é então escrever uma função para cada situação. Começamos pela verossimilhança para os dois parâmetros e gráfico de contorno da log-verossimilhança relativa e da função deviance como mostrado na Figura 1. Os gráficos são idênticos em aparência diferindo apenas nos valores das funções calculadas o que não aparece pois optamos por colocar rótulos nos contornos a serem traçados para valores que correspondem a 0,0016; 0,01; 0,0225; 0,04; 0,15; 0,20; 0,50; 0,80; 0,90 % de verossimilhança relativa que são os valores indicamos nos gráficos. Como temos dois parâmetros consideramos os valores $0.0016 = 0.04^2$ e $0.0225 = 0.15^2$ que definem regiões de confiança de interesse especial. São acrescentados ainda isolinhas correspondentes a valores correspondentes a regiões de confiança assintótica de 95 e 99% baseadas na $\chi^2_{(2)}$. São indicados ainda os pontos correspondentes aos valores verdadeiros dos parâmetros e da estimativa de máxima verossimilhança para a amostra.

```
> ll <- function(pars, dados)
+   -(length(dados)/2)*(log(2*pi) + log(pars[2]^2) +
+     sum((dados-pars[1])^2)/(length(dados)*pars[2]^2))
> ll.max <- ll(mle, dados=x)
> ll.musig <- apply(musig.vals, 1, ll, dados=x)
> niveis <- c(0.0016, 0.0225, 0.04, 0.15, 0.20, 0.50, 0.80, 0.90)
> ## log-verossimilhança relativa
> contour(mu.vals, sig.vals, matrix(ll.musig, nc=length(sig.vals))-ll.max,
```

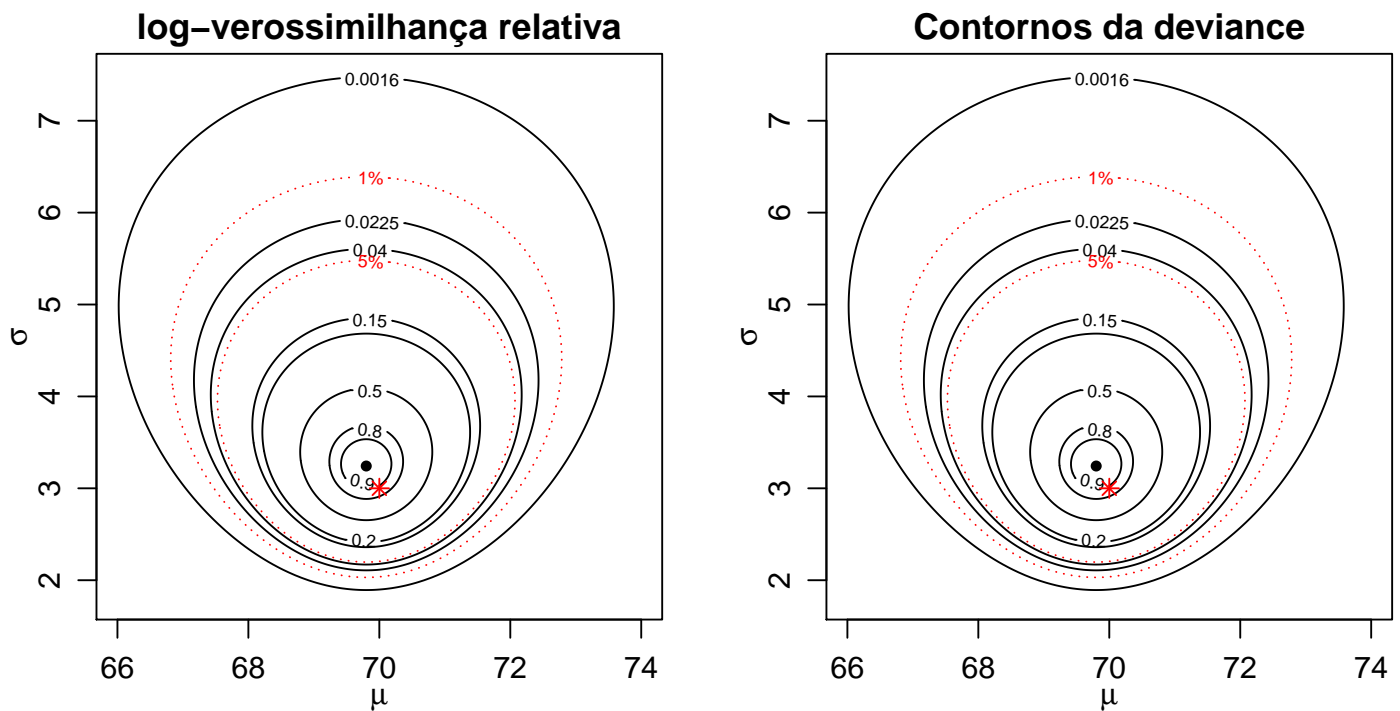


Figura 1: Log-verossimilhança relativa e deviance para função dos parâmetros da normal. Contornos indicam valores da verossimilhança relativa e no gráfico da esquerda quantis da $\chi^2_{(2)}$.

```

+         levels=log(niveis), labels=niveis, xlab=expression(mu), ylab=expression(sigma)
> contour(mu.vals, sig.vals, matrix(ll.musig, nc=length(sig.vals)) - ll.max,
+         levels=-qchisq(c(0.95, 0.99), df=2)/2, labels=c("5%", "1%"),
+         xlab=expression(mu), ylab=expression(sigma), lty=3, col=2, add=T)
> title("log-verossimilhança relativa")
> points(rbind(mle, c(70, 3)), pch=c(20,8), col=1:2)
> ## Deviance
> contour(mu.vals, sig.vals, -2*(matrix(ll.musig, nc=length(sig.vals))-ll.max),
+         levels=-2*log(niveis), labels= niveis, xlab=expression(mu), ylab=expression(sigma))
> contour(mu.vals, sig.vals, -2*(matrix(ll.musig, nc=length(sig.vals)) - ll.max),
+         levels=qchisq(c(0.95, 0.99), df=2), labels=c("5%", "1%"), xlab=expression(mu),
+         ylab=expression(sigma), lty=3, col=2, add=T)
> title("Contornos da deviance")
> points(rbind(mle, c(70, 3)), pch=c(20,8), col=1:2)

```

Pelo gráfico podemos verificar as diferenças entre três pares de regiões de confiança definidas por valores de corte $c = 0,04e0,15$, os valores de corte corrigidos para o número de parâmetros de $c^2 = 0,04^2 = 0.0016e0,15^2 = 0.0225$, e os valores correspondentes aos quantis 0,95 e 0,99 de distribuição $\chi^2_{(2)}$. Note-

se que valores c fornecem intervalos com intervalos/regiões mais estreitos e que não deverão atingir os níveis de cobertura nominais de 95 e 99%, c^2 definem intervalos/regiões bastante conservadoras e os valores baseados na aproximação assintótica χ^2 que definem regiões intermediárias entre as anteriores. A seguir são definidas funções para calcular valores das log-verossimilhanças relativas e perfilhadas de cada parâmetro.

```
> llest.mu <- function(mu, sigma, dados)
+   sapply(mu, function(m) -(length(dados)/2)*(log(2*pi) + log(sigma^2) + sum((dados-m)
> llest.sigma <- function(sigma, mu, dados)
+   sapply(sigma, function(s) -(length(dados)/2)*(log(2*pi) + log(s^2) + sum((dados-mu)
> llprof.mu <- function(mu, dados){
+   sapply(mu, function(m){
+     sigma2 <- sum((dados - m)^2)/length(dados)
+     -(length(dados)/2)*(log(2*pi) + log(sigma2) + sum((dados-m)^2)/(length(dados)*si
+   })
+ }
> llprof.sigma <- function(sigma, dados){
+   xbar <- mean(x)
+   sapply(sigma, function(s) -(length(dados)/2)*(log(2*pi) + log(s^2) +
+
+                                     sum((dados-xbar)^2)/(length(dados)
```

Para obtenção dos gráficos as funções acima são calculadas para uma sequência de valores de cada um dos parâmetros. Definimos também valores de corte relativos a verossimilhanças relativas e quantis da χ^2 . Embora nem todos os intervalos tenham interpretação probabilística, estão relacionados com o valor nominal de 95%.

```
> lle.mu <- llest.mu(mu.vals, sigma=mle[2], dados=x)
> llp.mu <- llprof.mu(mu.vals, dados=x)
> lle.sig <- llest.sigma(sig.vals, mu=mle[1], dados=x)
> llp.sig <- llprof.sigma(sig.vals, dados=x)
> cortes95 <- c(0.15, 0.0225, exp(-qchisq(0.95, df=1)/2), exp(-qchisq(0.95, df=2)/2))
> #cortes99 <- c(0.04, 0.04^2, exp(-qchisq(0.99, df=1)/2), exp(-qchisq(0.99, df=2)/2))
```

Na Figura 2 são mostradas log-verossimilhanças relativas (esquerda) e funções deviance (direita) para os parâmetros μ (em cima) e σ^2 (em baixo). Em cada gráfico são mostradas as verossimilhanças esti-

madadas e perfilhada e diferentes valores de corte para obtenção de intervalos de confiança que permitem comparações entre as formas de obtenção.

```

> ## logverossimilhanças relativas para mu
> plot(mu.vals, lle.mu - ll.max, xlab=expression(mu),
+      ylab=expression(ll[est/prof](mu)), type="l")
> lines(mu.vals, llp.mu - ll.max, lwd=2)
> abline(h=log(cortes95), lty=c(1,1,2,2), lwd=c(1,2,1,2))
> legend("bottomleft", expression(c, c^2, chi[1]^2, chi[2]^2), lty=c(1,1,2,2), lwd=1:2)
> legend("bottomright", c("estimada", "perfilhada"), lwd=1:2)
> ## deviance para mu
> plot(mu.vals, -2*(lle.mu - ll.max), xlab=expression(mu),
+      ylab=expression(D[est/prof](mu)), type="l")
> lines(mu.vals, -2*(llp.mu - ll.max), lwd=2)
> abline(h=-2*log(cortes95), lty=c(1,1,2,2), lwd=c(1,2,1,2))
> legend("topleft", expression(c,c^2,chi[1]^2,chi[2]^2), lty=c(1,1,2,2), lwd=1:2)
> legend("topright", c("estimada", "perfilhada"), lwd=1:2)
> ## logverossimilhanças relativas para sigma
> plot(sig.vals, lle.sig - ll.max, xlab=expression(sigma),
+      ylab=expression(ll[est/prof](sigma)), type="l")
> lines(sig.vals, llp.sig - ll.max, lwd=2)
> abline(h=log(cortes95), lty=c(1,1,2,2), lwd=c(1,2,1,2))
> legend("bottomleft", expression(c,c^2,chi[1]^2,chi[2]^2), lty=c(1,1,2,2), lwd=1:2)
> legend("bottomright", c("estimada", "perfilhada"), lwd=1:2)
> ## deviance para sigma
> plot(sig.vals, -2*(lle.sig - ll.max), xlab=expression(sigma),
+      ylab=expression(D[est/prof](sigma)), type="l")
> lines(sig.vals, -2*(llp.sig - ll.max), lwd=2)
> abline(h=-2*log(cortes95), lty=c(1,1,2,2), lwd=c(1,2,1,2))
> legend("topleft", expression(c, c^2, chi[1]^2, chi[2]^2), lty=c(1,1,2,2), lwd=1:2)
> legend("topright", c("estimada", "perfilhada"), lwd=1:2)

```

Nota-se nos gráficos inferiores o resultado comentado anteriormente de que as verossimilhanças estimadas e perfilhadas para o parâmetro σ não diferem pois as estimativas de μ não dependem de σ .

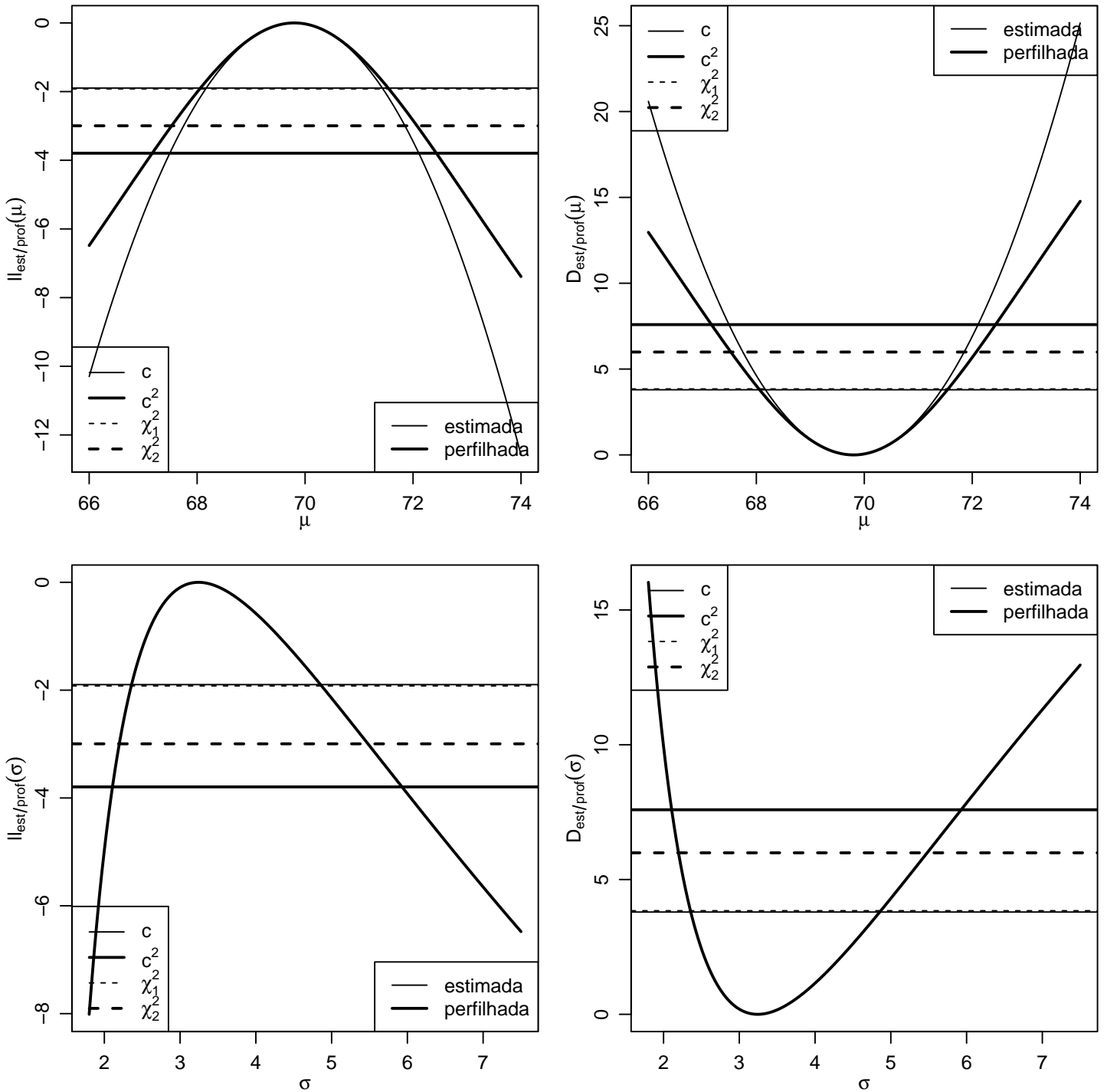


Figura 2: Verossimilhanças estimadas e perfilhadas: Verossimilhanças relativas (esquerda) e deviance (direita) para μ (superior) e σ (inferior). Linhas representam diferentes critérios para obtenção de intervalos.

Como uma extensão as funções acima poderiam ser adaptadas para fornecer intervalos e/ou regiões para os parâmetros.

Alternativas de código Os códigos anteriores poderiam ainda serem escritos utilizando a função do R para cálculo da densidade de normal.

```
> llR <- function(pars, dados) sum(dnorm(dados, mean=pars[1], sd=pars[2], log=TRUE))
> all.equal(ll.musig, apply(musig.vals, 1, llR, dados=x))
```

```
[1] TRUE
```

```
> lleR.mu <- function(mu, sigma, dados) sapply(mu, function(m) sum(dnorm(dados, mean=m, sd=sigma, log=TRUE)))
> all.equal(lle.mu, lleR.mu(mu.vals, sigma=mle[2], dados=x))
```

```
[1] TRUE
```

```
> lleR.sigma <- function(sigma, mu, dados) sapply(sigma, function(s) sum(dnorm(dados, mean=mu, sd=s, log=TRUE)))
> all.equal(lle.sig, lleR.sigma(sig.vals, mu=mle[1], dados=x))
```

```
[1] TRUE
```

```
> llpR.mu <- function(mu, dados){
+   sapply(mu, function(m){
+     sigma <- sqrt(sum((dados - m)^2)/length(dados))
+     sum(dnorm(dados, mean=m, sd=sigma, log=TRUE))})}
> all.equal(llp.mu, llpR.mu(mu.vals, dados=x))
```

```
[1] TRUE
```

```
> llpR.sigma <- function(sigma, dados){
+   xbar <- mean(dados)
+   sapply(sigma, function(s) sum(dnorm(dados, mean=xbar, sd=s, log=TRUE)))
+ }
> all.equal(llp.sig, llpR.sigma(sig.vals, dados=x))
```

```
[1] TRUE
```

Uma outra alternativa é escrever uma única função genérica, em que o resultado pode ser qualquer uma das alternativas pedidas, dependendo do que é fornecido no argumento da função. Note que não é preciso escrever as funções de verossimilhanças perfilhadas, bastando aplicar a definição substituindo as estimativas na função de verossimilhança original. Aproveitamos para escrever a função com opção de usar estatísticas suficientes, o que reduz os cálculos a serem feitos pelo computador quando fazendo o gráfico das funções.


```

> ll.N <- function(par, fixo, dados, suficientes = NULL)
+ {
+   if(!any(length(par) == 1:2) || is.null(names(par)) ||
+     !any(c("mu", "sigma") %in% names(par)))
+     stop("par deve ser um vetor ou lista com nomes \"mu\" e/ou \"sigma\"")
+   ## fixo é ignorado se length(par) == 2
+   ## fixo pode ser: "mle", "prof" ou um valor numérico
+   ## suficientes deve ser um vetor com 3 elementos:
+   ## (i) n (ii) média e (iii) somatório dos quadrados dos dados
+   if(!is.null(suficientes)){
+     suficientes <- unlist(suficientes)
+     n <- suficientes[1]; xbar <- suficientes[2]; sumx2 <- suficientes[3]
+   }
+   else{
+     n <- length(dados); xbar <- mean(dados); sumx2 <- sum(dados^2)
+   }
+   if(length(par) == 1){
+     if(names(par) == "mu"){
+       mu <- par
+       if(is.numeric(fixo)) sigma2 <- fixo^2
+       else sigma2 <- switch(fixo, mle=(sumx2-n*xbar^2)/n,
+                             prof=(sumx2-n*mu*(2*xbar-mu))/n)
+     }
+     if(names(par) == "sigma"){
+       sigma2 <- par^2
+       mu <- ifelse(any(fixo == c("mle", "prof")), xbar, fixo)
+     }
+   }
+   else{
+     mu <- par["mu"]
+     sigma2 <- (par["sigma"])^2
+   }
+   ll <- -(n/2)*(log(2*pi) + log(sigma2) + (sumx2-n*mu*(2*xbar-mu))/(n*sigma2))

```

```
+ return(ll)
+ }
```

Nesta função se `par` recebe dois valores, a log-verossimilhança bidimensional é retornada. Se recebe apenas um valor e `fixo` é fornecido então a log-verossimilhança de um parâmetro é calculada para um valor fixo do outro. Se `fixo="mle"` este valor fixo é o MLE e a verossimilhança obtida é a estimada. Finalmente `fixo="prof"` indica que valores da verossimilhança perfilhada são calculados

```
> ## definindo estísticas suficientes
> sufs <- list(n=length(x), xbar=mean(x), sumx2 = sum(x^2))
> ## bidimensional
> names(musig.vals) <- c("mu", "sigma")
> llN.musig <- apply(musig.vals, 1, ll.N, suf=sufs)
> all.equal(ll.musig, llN.musig)
```

```
[1] TRUE
```

```
> ## 1D para mu com sigma fixando no valor verdadeiro
> llN.mu0 <- apply(data.frame(mu=mu.vals), 1, ll.N, fixo=3, suf=sufs)
> ## 1D para mu com sigma fixando no MLE
> lleN.mu <- apply(data.frame(mu=mu.vals), 1, ll.N, fixo="mle", suf=sufs)
> all.equal(lle.mu, lleN.mu)
```

```
[1] TRUE
```

```
> ## perfilhada para mu
> llpN.mu <- apply(data.frame(mu=mu.vals), 1, ll.N, fixo="prof", suf=sufs)
> all.equal(lle.mu, lleN.mu)
```

```
[1] TRUE
```

```
> ##
> ## 1D para sigma com mu fixado no valor verdadeiro
> llN.sig0 <- apply(data.frame(sigma=sig.vals), 1, ll.N, fixo=70, suf=sufs)
> ## 1D para sigma com mu fixado no MLE
> lleN.sig <- apply(data.frame(sigma=sig.vals), 1, ll.N, fixo="mle", suf=sufs)
> all.equal(lle.sig, lleN.sig)
```

```
[1] TRUE
```

```
> ## perfilhada para sigma  
> llpN.sig <- apply(data.frame(sigma=sig.vals), 1, ll.N, fixo="prof", suf=sufs)  
> all.equal(lle.sig, lleN.sig)
```

```
[1] TRUE
```

2. Obter as funções de verossimilhança como no exemplo anterior para pelo menos dois modelos que envolvam ao menos dois parâmetros.