CE-209: Inferência Estatística I

Primeiro Semestre de 2003

Última atualização: 20 de janeiro de 2009

Sumário

Introdução

Nas aulas práticas deste curso vamos utilizar o programa estatístico R. Vamos começar "experimentando o R", para ter uma idéia de seus recursos e a forma de trabalhar. Para isto vamos rodar e estudar os comandos mostrados no texto e seus resultados para nos familiarizar com o programa. Nas sessões seguintes iremos ver com mais detalhes o uso do programa R. Siga os seguintes passos:

1. inicie o R em seu computador;
2. voce verá uma janela de comandos com o símbolo >,
   este é o *prompt* do R indicando que o programa está pronto para receber comandos;
3. a seguir digite (ou "recorte e cole") os comandos mostrados abaixo.

No restante deste texto vamos seguir as seguintes convenções.

- comandos do R são sempre mostrados em fontes do tipo `typewriter como esta`,
- linhas iniciadas pelo símbolo # são comentários e são ignoradas pelo R.

1 Distribuições de Probabilidade

O programa R inclui funcionalidade para operações com distribuições de probabilidades. Para cada distribuição há 4 operações básicas indicadas pelas letras:

`d` calcula a densidade de probabilidade $f(x)$ no ponto

`p` calcula a função de probabilidade acumulada $F(x)$ no ponto

`q` calcula o quantil correspondente a uma dada probabilidade

`r` amostra da distribuição

1.1 Distribuição Normal

A funcionalidade para distribuição normal é implementada por argumentos que combinam as letras acima com o termo `norm`. Vamos ver alguns exemplos com a distribuição normal padrão.

```
> dnorm(-1)
[1] 0.2419707

> (1/sqrt(2*pi)) * exp((-1/2)*(-1)^2)
[1] 0.2419707

> pnorm(-1)
[1] 0.1586553

> qnorm(0.975)
[1] 1.959964

> rnorm(10)
 [1] -0.0442493 -0.3604689  0.2608995 -0.8503701 -0.1255832  0.4337861
 [7] -1.0240673 -1.3205288  2.0273882 -1.7574165
```

O primeiro valor acima corresponde ao valor da densidade da normal

$$f(x) = \frac{1}{2\pi\sigma^2} \exp\{\frac{1}{2\sigma^2}(x-\mu)^2\}$$

com parâmetros ($\mu = 0, \sigma^2 = 1$) no ponto -1, como pode ser confirmado com o cálculo apresentado no segundo comando.
A função `pnorm(-1)` calcula a probabilidade $P(X < -1)$.
O comando `qnorm(0.975)` calcula o valor de a tal que $P(X < a) = 0.975$.
Finalmente o comando `rnorm(10)` gera uma amostra de 10 elementos da normal padrão.

Nota: cada vez que o comando `rnorm` é chamado diferentes elementos da amostra são produzidos, porque a *semente* do gerador é modificada. Para gerar duas amostras idênticas deve-se usar o comando `set.seed` como ilustrado abaixo.

```
> set.seed(214)      # define o valor da semente
> rnorm(5)           # amostra de 5 elementos
[1] -0.46774980  0.04088223  1.00335193  2.02522505  0.30640096
> rnorm(5)           # outra amostra de 5 elementos
```

```
[1]  0.4257775  0.7488927  0.4464515 -2.2051418  1.9818137
> set.seed(214)     # retorna o valor da semente ao valor inicial
> rnorm(5)          # gera novamente a primeira amostra de 5 elementos
[1] -0.46774980  0.04088223  1.00335193  2.02522505  0.30640096
```

As funções acima possuem argumentos adicionais, para os quais valores padrão (*default*) foram assumidos, e que podem ser modificados. Usamos `args` para ver os argumentos de uma função e `help` para visualisar a documentação detalhada:

```
> args(rnorm)
function (n, mean = 0, sd = 1)
```

As funções relacionadas à distribuição normal tem (entre outros) os argumentos `mean` e `sd` para definir média e desvio padrão da distribuição que podem ser modificados como nos exemplos a seguir.

```
> qnorm(0.975, mean = 100, sd = 8)
[1] 115.6797

> qnorm(0.975, m = 100, s = 8)
[1] 115.6797

> qnorm(0.975, 100, 8)
[1] 115.6797
```

Para informações mais detalhadas pode-se usar a função `help`. O comando

```
> help(rnorm)
```

irá exibir em uma janela a documentação da função que pode também ser chamada com `?rnorm`. Note que ao final da documentação são apresentados exemplos que podem ser rodados pelo usuário e que auxiliam na compreensão da funcionalidade.
Note que as 4 funções relacionadas à distribuição normal são documentadas conjuntamente, portanto `help(rnorm)`, `help(qnorm)`, `help(dnorm)` e `help(pnorm)` vão exibir a mesma documentação.

Estas funções aceitam também vetores em seus argumentos como ilustrado nos exemplo abaixo.

```
> qnorm(c(0.05, 0.95))
[1] -1.644854  1.644854
> rnorm(4, mean=c(0, 10, 100, 1000))
[1]   0.1599628   9.0957340 100.5595095 999.9129392
> rnorm(4, mean=c(10, 20, 30, 40), sd=c(2, 5))
[1] 10.58318 21.92976 29.62843 42.71741
```

Note que no último exemplo a *lei da reciclagem* foi utilizada no vetor de desvios padrão, i.e. os desvios padrão utilizados foram $(2, 5, 2, 5)$.

Cálculos de probabilidades usuais, para os quais utilizávamos tabelas estatísticas podem ser facilmente obtidos como no exemplo a seguir.

Seja X uma v.a. com distribuição $N(100, 100)$. Calcular as probabilidades:

1. $P[X < 95]$
2. $P[90 < X < 110]$

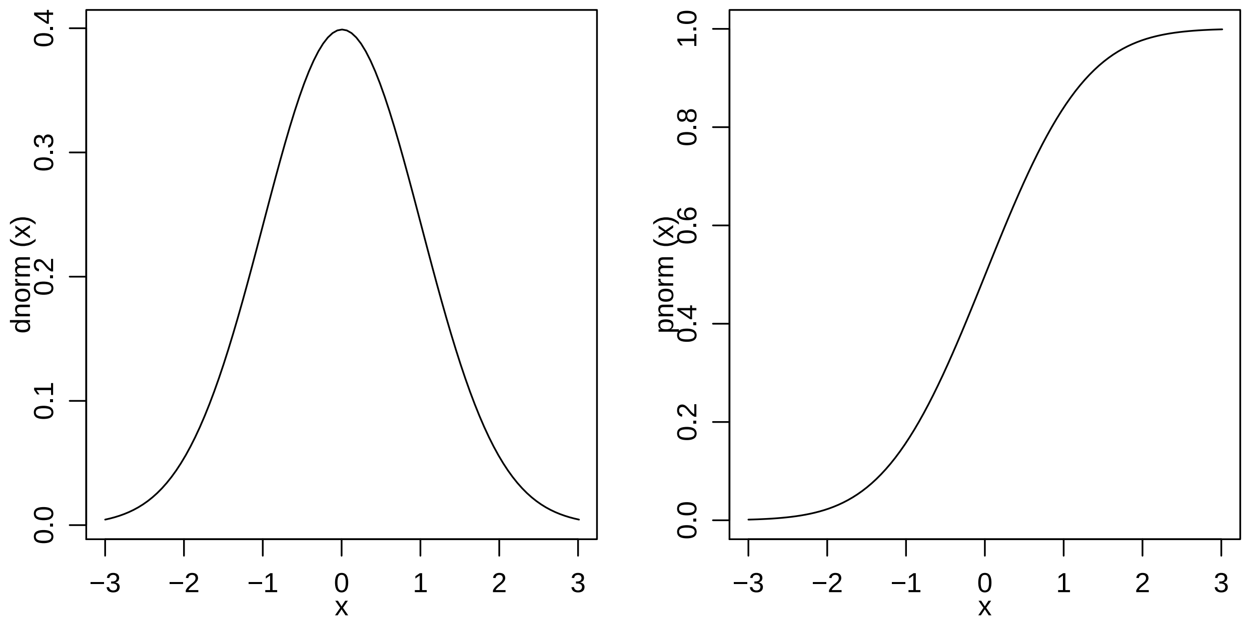

Figura 1: Funções de densidade e probabilidade da distribuição normal padrão.

3. $P[X > 95]$

Os comandos do R são:

```
> pnorm(95, 100, 10)
[1] 0.3085375\

> pnorm(110, 100, 10) - pnorm(90, 100, 10)
[1] 0.6826895

> 1 - pnorm(95, 100, 10)
[1] 0.6914625
> pnorm(95, 100, 10, lower=F)
[1] 0.6914625
```

Note a última probabilidade foi calculada de duas formas diferentes, sendo a segunda usando o argumento `lower` mais estável numericamente.

A seguir vamos ver comandos para fazer gráficos de distribuições de probabilidade. Vamos fazer gráficos de funções de densidade e de probabilidade acumulada. Estude cuidadosamente os comandos abaixo e verifique os gráficos por eles produzidos. A Figura 1 mostra gráficos da densidade (esquerda) e probabilidade acumulada da normal padrão produzidos com os comandos:

```
> plot(dnorm, -3, 3)
> plot(pnorm, -3, 3)
```

A Figura 2 mostra gráficos da densidade (esquerda) e probabilidade acumulada da $N(100, 64)$ produzidos com os comandos:

```
> plot(function(x) dnorm(x, 100, 8), 70, 130)
> plot(function(x) pnorm(x, 100, 8), 70, 130)
```

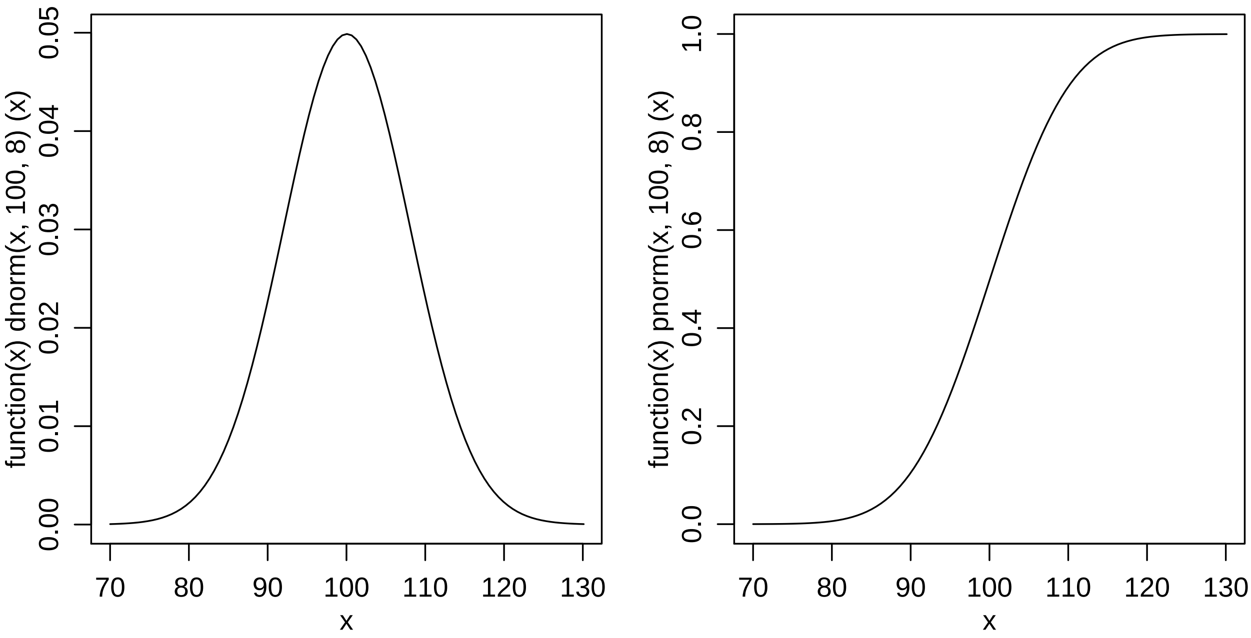

Figura 2: Funções de densidade e probabilidade da $N(100, 64)$.

Podemos incluir títulos e mudar texto dos eixos conforme mostrado na gráfico da esquerda da Figura 3 e nos dois primeiros comandos abaixo. Os demais comandos mostram como colocar diferentes densidades em um um mesmo gráfico como ilustrado à direita da mesma Figura.

```
> plot(dnorm, -3, 3, xlab='valores de X', ylab='densidade de probabilidade')
> title('Distribuicão Normal\nX ~ N(100, 64)')

> plot(function(x) dnorm(x, 100, 8), 60, 140, ylab='f(x)')
> plot(function(x) dnorm(x, 90, 8), 60, 140, add=T, col=2)
> plot(function(x) dnorm(x, 100, 15), 60, 140, add=T, col=3)
> legend(120, 0.05, c("N(100,64)","N(90,64)","N(100,225)"), fill=1:3)
```

1.2 Distribuição Binomial

Cálculos para a distribuição binomial são implementados combinando as *letras básicas* vistas acima com o termo `binom`. Vamos primeiro investigar argumentos e documentação com os comandos `args` e `binom`.

```
> args(dbinom)
function (x, size, prob, log = FALSE)

> help(dbinom)
```

Seja X uma v.a. com distribuição Binomial com $n = 10$ e $p = 0.35$. Vamos ver os comandos do R para:

1. fazer o gráfico das função de densidade
2. idem para a função de probabilidade
3. calcular $P[X = 7]$

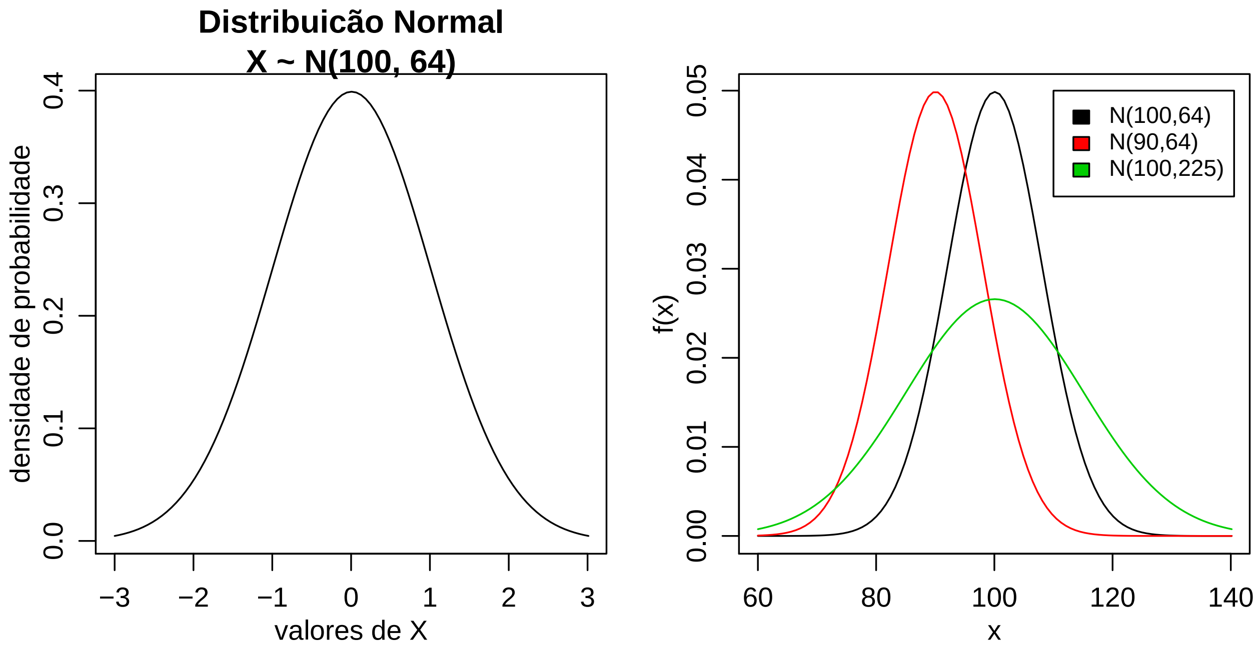

Figura 3: Gráfico com texto nos eixos e título (esquerda) e várias distribuições em um mesmo gráfico (direita).

4. calcular $P[X < 8] = P[X \leq 7]$
5. calcular $P[X \geq 8] = P[X > 7]$
6. calcular $P[3 < X \leq 6] = P[4 \geq X < 7]$

Note que sendo uma distribuição discreta de probabilidades os gráficos são diferentes dos obtidos para distribuição normal e os cálculos de probabilidades devem considerar as probabilidades nos pontos. Os gráficos das funções de densidade e probabilidade são mostrados na Figura 4.

```
> x <- 0:10

> fx <- dbinom(x, 10, 0.35)
> plot(x, fx, type='h')

> Fx <- pbinom(x, 10, 0.35)
> plot(x, Fx, type='S')

> dbinom(7, 10, 0.35)
[1] 0.02120302

> pbinom(7, 10, 0.35)
[1] 0.9951787
> sum(dbinom(0:7, 10, 0.35))
[1] 0.9951787

> 1-pbinom(7, 10, 0.35)
[1] 0.004821265
> pbinom(7, 10, 0.35, lower=F)
```

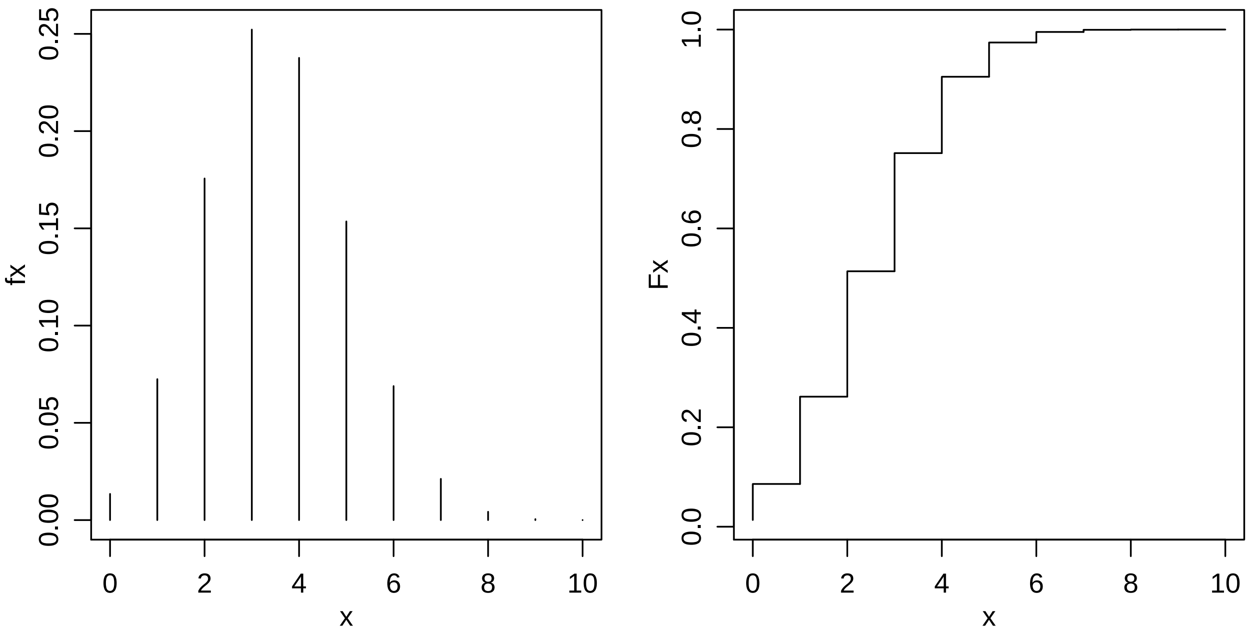

Figura 4: Funções de densidade e probabilidade da $B(10, 0.35)$.

```
[1] 0.004821265

> pbinom(6, 10, 0.35) - pbinom(3, 10, 0.35)
[1] 0.4601487
> sum(dbinom(4:6, 10, 0.35))
[1] 0.4601487
```

1.3 Exercícios

Nos exercícios abixo iremos também usar o R como uma calculadora estatística para resolver alguns exemplos/exercícios de probabilidade tipicamente apresentados em um curso de estatística básica.

Os exercícios abaixo com indicação de página foram retirados de:

Magalhães, M.N. & Lima, A.C.P. (2001) **Noções de Probabilidade e Estatística**. 3 ed. São Paulo, IME-USP. 392p.

1. (Ex 1, pag 67) Uma moeda viciada tem probabilidade de cara igual a 0.4. Para quatro lançamentos independentes dessa moeda, estude o comportamento da variável *número de caras* e faça um gráfico de sua função de distribuição.

2. (Ex 3.6, pag 65) Num estudo sobre a incidência de câncer foi registrado, para cada paciente com este diagnóstico o número de casos de câncer em parentes próximos (pais, irmãos, tios, filhos e sobrinhos). Os dados de 26 pacientes são os seguintes:

Paciente	1	2	3	4	5	6	7	8	9	10	11	12	13
Incidência	2	5	0	2	1	5	3	3	3	2	0	1	1

Paciente	14	15	16	17	18	19	20	21	22	23	24	25	26
Incidência	4	5	2	2	3	2	1	5	4	0	0	3	3

Estudos anteriores assumem que a incidência de câncer em parentes próximos pode ser modelada pela seguinte função discreta de probabilidades:

Incidência	0	1	2	3	4	5
p_i	0.1	0.1	0.3	0.3	0.1	0.1

- os dados observados concordam com o modelo teórico?
- faça um gráfico mostrando as frequências teóricas (esperadas) e observadas.

3. (Ex 5, pag 77) Sendo X uma variável seguindo o modelo Binomial com parâmetro $n = 15$ e $p = 0.4$, pergunta-se:
   - $P(X \geq 14)$
   - $P(8 < X \leq 10)$
   - $P(X < 2 \text{ ou } X \geq 11)$
   - $P(X \geq 11 \text{ ou } X > 13))$
   - $P(X > 3 \text{ e } X < 6)$
   - $P(X \leq 13 \mid X \geq 11)$

4. (Ex 8, pag 193) Para $X \sim N(90, 100)$, obtenha:
   - $P(X \leq 115)$
   - $P(X \geq 80)$
   - $P(X \leq 75)$
   - $P(85 \leq X \leq 110)$
   - $P(|X - 90| \leq 10)$
   - P valor de a tal que $P(90 - a \leq X \leq 90 + a) = \gamma,\ \gamma = 0.95$

5. Faça os seguintes gráficos:
   - da função de densidade de uma variável com distribuição de Poisson com parametro $lambda = 5$
   - da densidade de uma variável $X \sim N(90, 100)$
   - sobreponha ao gráfico anterior a densidade de uma variável $Y \sim N(90, 80)$ e outra $Z \sim N(85, 100)$
   - densidades de distribuições χ^2 com 1, 2 e 5 graus de liberdade.

2 Alguns recursos do R

2.1 O projeto R

O programa R é gratuito e de código aberto que propicia excelente ambiente para análises estatísticas e com recursos gráficos de alta qualidade. Detalhes sobre o projeto, colaboradores, documentação e diversas outras informações podem ser encontradas na página oficial do projeto em:
http://www.r-project.org.

O programa pode ser copiado livremente pela internet. Há um espelho (*mirror*) brasileiro da área de *downloads* do programa no *Departamento de Estatística da UFPR*:
http://www.est.ufpr.br/R
ou então via FTP:
ftp://est.ufpr.br/R

Será feita uma apresentação rápida da página do R durante o curso onde os principais recursos serão comentados assim como as idéias principais que governam o projeto e suas direções futuras.

2.2 Demos

O R vem com algumas demonstrações (*demos*) de seus recursos "embutidas" no programa. Para listar as demos disponíveis digite na linha de comando:

```
demo()
```

E para rodar uma delas basta colocar o nome da escolhida entre os parênteses. Por exemplo, vamos rodar a de recursos gráficos. Note que os comandos vão aparecer na janela de comandos e os gráficos serão automaticamente produzidos na janela gráfica. Voce vai ter que teclar ENTER para ver o próximo gráfico.

- inicie o programa R
- no "prompt" do programa digite:

  ```
  demo(graphics)
  ```

- Voce vai ver a seguinte mensagem na tela:

  ```
      demo(graphics)
      ---- ~~~~~~~~

      Type  <Return>   to start :
  ```

- pressione a tecla ENTER
- a "demo" vai ser iniciada e uma tela gráfica irá se abrir. Na tela de comandos serão mostrados comandos que serão utilizados para gerar um gráfico seguidos da mensagem:

  ```
      Hit <Return> to see next plot:
  ```

- inspecione os comandos e depois pressione novamente a tecla ENTER.
  Agora voce pode visualizar na janela gráfica o gráfico produzido pelos comandos mostrados anteriormente. Inspecione o gráfico cuidadosamente verificando os recursos utilizados (título, legendas dos eixos, tipos de pontos, cores dos pontos, linhas, cores de fundo, etc).

- agora na tela de comandos apareceram novos comandos para produzir um novo gráfico e a mensagem:

```
Hit <Return> to see next plot:
```

- inspecione os novos comandos e depois pressione novamente a tecla ENTER.
  Um novo gráfico surgirá ilustrando outros recursos do programa.
  Prossiga inspecionando os gráficos e comandos e pressionando ENTER até terminar a "demo".
  Experimente outras demos como `demo(pers)` e `demo(image)`, por exemplo.

2.3 Um tutorial sobre o R

Além dos materiais disponíveis na página do programa há também um *Tutorial de Introdução ao* R disponível em http://www.est.ufpr.br/Rtutorial.

2.4 RWeb

Este é um mecanismo que permite rodar o R pela web, sem que voce precise ter o R instalado no seu computador. Para usá-lo basta estar conectado na internet.

Para acessar o **RWeb** vá até a página do Re no menu à esquerda da página siga os links:
R GUIs . . . R Web
Nesta página selecione primeiro o link **R Web** e examine seu conteúdo.

Há ainda uma diversidade de recursos disponíveis na página do projeto. Os participantes do curso são estimulados a explorar detalhadamente ao final do curso os outros recursos da página.

2.5 Cartão de referência

Como voce já pode perceber, para utilizar o R é necessário conhecer e digitar comandos. Isto pode trazer alguma dificuldade no inicio até que o usuário de familiarize com os comandos mais comuns. Uma boa forma de aprender e memorizar os comandos básicos é utilizar o Cartão de Referência que contém os comandos mais frequentemente utilizados. Imprima o conteúdo deste arquivo (1 folha) e carregue sempre com voce.

3 Usando simulação para ilustrar resultados

Podemos utilizar recursos computacionais e em particular simulações para inferir distribuições amostrais de quantidades de interesse. Na teoria de estatística existem vários resultados que podem ser ilustrados via simulação, o que ajuda na compreensão e visualização dos conceitos e resultados. Veremos alguns exemplos a seguir.

Este uso de simulações é apenas um ponto de partida pois estas são especialmente úteis para explorar situações onde resultados teóricos não são conhecidos ou não podem ser obtidos.

3.1 Relações entre a distribuição normal e a χ^2

Resultado 1: Se $Y \sim \mathrm{N}(0,1)$ então $Y^2 \sim \chi^2_{(1)}$.

Vejamos como ilustrar este resultado. Vamos começar gerando uma amostra de 1000 números da distribuição normal padrão. A seguir vamos fazer um histograma dos dados obtidos e sobrepor a curva da distribuição teórica. Fazemos isto com os comando abaixo e o resultado está no gráfico da esquerda da Figura 5.

```
> y <- rnorm(1000)
> hist(y, prob=T)
> curve(dnorm(x), -4, 4, add=T)
```

Note que, para fazer a comparação do histograma e da curva teórica é necessário que o histograma seja de frequências relativas e para isto usamos o argumento `prob = T`.

Agora vamos estudar o comportamento da variável ao quadrado. O gráfico da direita da Figura 5 mostra o histograma dos quadrados do valores da amostra e a curva da distribuição de $\chi^2_{(1)}$.

```
> hist(y^2, prob=T)
> curve(dchisq(x, df=1), 0, 10, add=T)
```

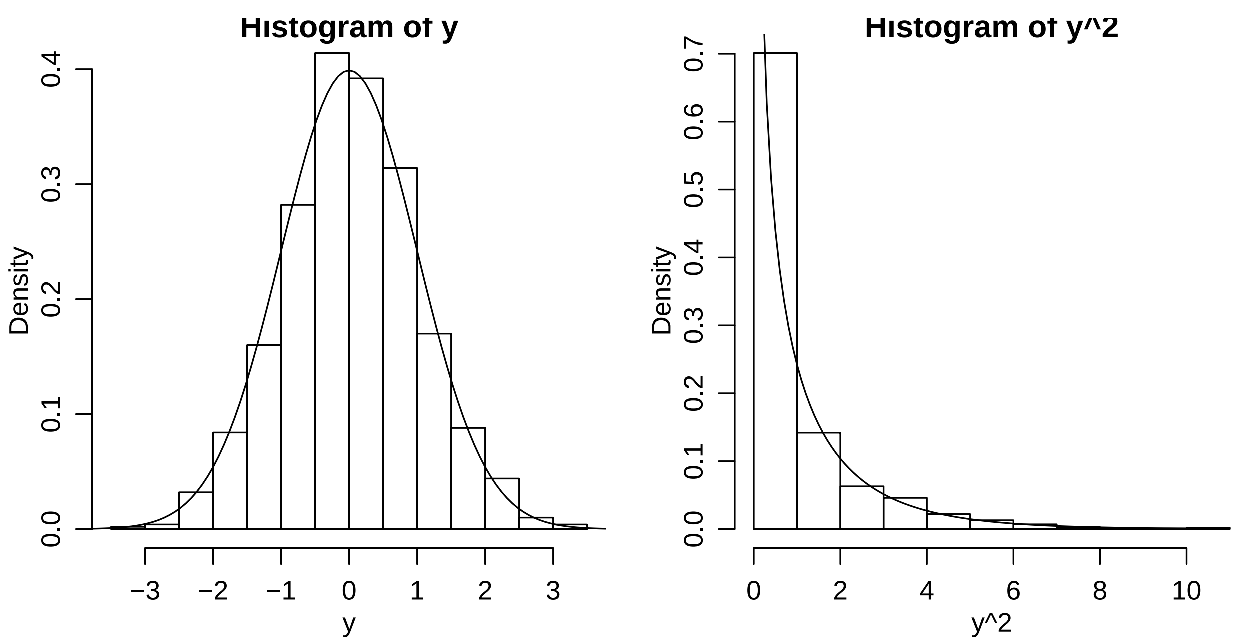

Figura 5: Histograma das amostra da e a curva teórica da distribuição normal padrão (esquerda) e histograma dos valores ao quadrado com a curva teórica da distribuição $\chi^2_{(1)}$ (direita).

Nos gráficos anteriores comparamos o histograma da distribuição empírica obtida por simulação com a curva teórica da distribuição. Uma outra forma e mais eficaz forma de comparar distribuições empíricas e teóricas é comparar os quantis das distribuições e para isto utilizamos o *qq-plot*. O *qq-plot* é um gráfico dos dados ordenados contra os quantis esperados de uma certa distribuição. Quanto mais próximo os pontos estiverem da reta 1-1 mais próximos os dados observados estão da distribuição considerada. Portanto para fazer o `qqplot` seguimos os seguintes passos:

1. obter os dados,
2. obter os quantis da distribuição teórica,
3. fazer um gráfico dos dados ordenados contra os quantis da distribuição.

Vamos ilustrar isto nos comandos abaixo. Primeiro vamos considerar como dados os quadrados da amostra da normal obtida acima. Depois obtemos os quantis da distribução χ^2 e por fim usamos a função `qqplot` para obter o gráfico mostrado na Figura 6. Adicionamos neste gráfico a linha 1-1.

```
> quantis <- qchisq(ppoints(length(y)), df=1)
> qqplot(quantis, y^2)
> abline(0,1)
```

Note que o comando `qchisq(ppoints(length(y)), df=1)` acima está concatenando 3 comandos e calcula os quantis da χ^2 a partir de uma sequência de valores de probabilidade gerada por `ppoints`. O número de elementos desta sequência deve igual ao número de dados e por isto usamos `length(y)`.

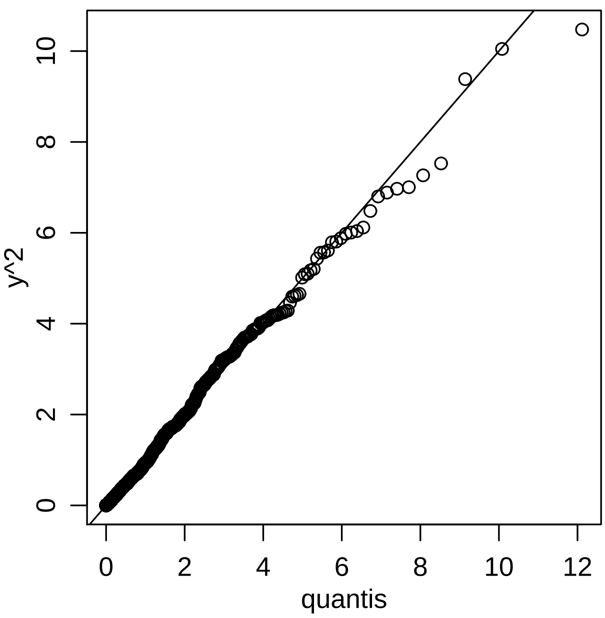

Figura 6: Comparando dados e quantis da χ^2 utilizando o *qq-plot*

Resultado 2: Se $Y_1, Y_2, \ldots Y_n \sim \mathrm{N}(0,1)$ então $\sum_1^n Y_i^2 \sim \chi^2_{(n)}$.

Para ilustrar este resultado vamos gerar 10.000 amostras de 3 elementos cada da distribuição normal padrão, elevar os valores ao quadrado e, para cada amostra, somar os quadrados dos três números. Na Figura 7 mostramos o histograma dos valores obtidos com a curva da distribuição esperada e o *qq-plot*.

```
> y <- matrix(rnorm(30000), nc=3)
> sy2 <- apply(y^2, 1, sum)
> hist(sy2, prob=T, main="")
> curve(dchisq(x, df=3), 0, 30, add=T)
> qqplot(qchisq(ppoints(length(sy2)), df=3), sy2)
> abline(0,1)
```

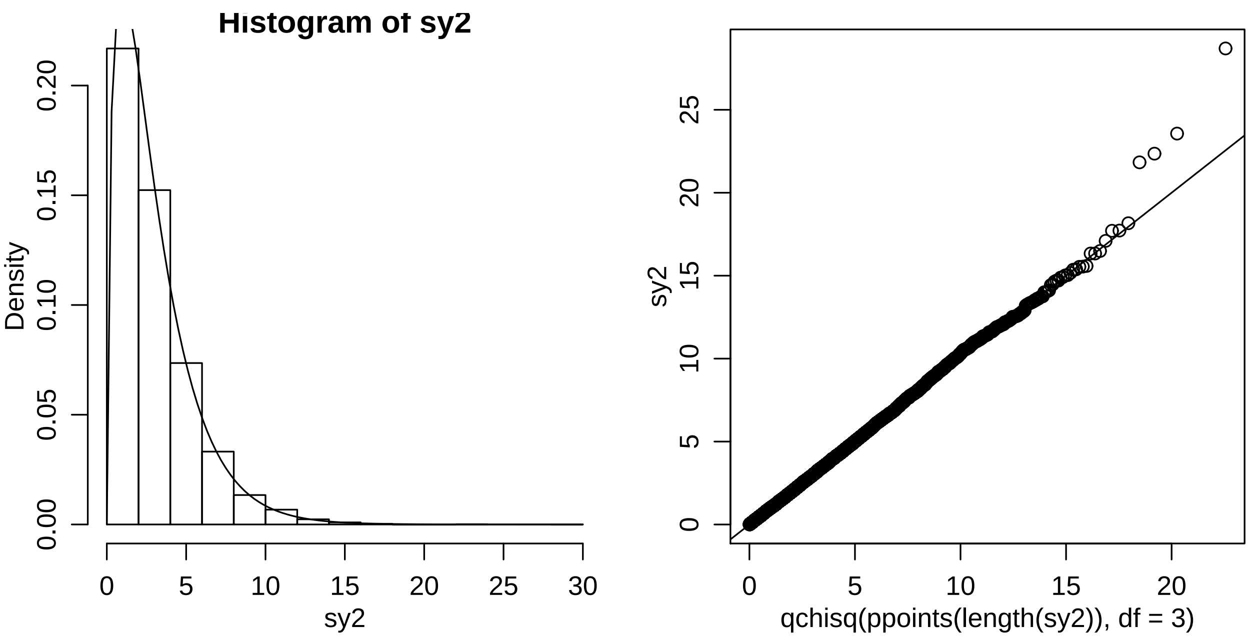

Figura 7: Histograma da uma amostra da soma dos quadrados de três valores da normal padrão e a curva teórica da distribuição de $\chi^2_{(3)}$ (esquerda) e o respectivo *qq-plot*.

3.2 Distribuição amostral da média de amostras da distribuição normal

Resultado 3: Se $Y_1, Y_2, \ldots Y_n \sim \mathrm{N}(\mu, \sigma^2)$ então $\bar{y} \sim \mathrm{N}(\mu, \sigma^2/n)$.

Neste exemplo vamos obter 1000 amostras de tamanho 20 de uma distribuição normal de média 100 e variância 30. Vamos organizar as amostras em uma matriz onde cada coluna corresponde a uma amostra. A seguir vamos calcular a média de cada amostra. Pelo **Resultado 3** acima esperamos que a média das médias amostrais seja 100 e a variância seja 1.5 (= 30/20), e que a distribuição das médias amostrais seja normal, valores bem próximos dos obtidos acima. Para completar vamos obter o gráfico com o histograma das médias das amostras e a distribuição teórica conforme Figura 8 e o respectivo `qq-plot`.

```
> y <- matrix(rnorm(20000, mean=100, sd=sqrt(30)), nc=1000)
> ybar <- apply(y, 2, mean)
> mean(ybar)
> [1] 99.96043
> var(ybar)
[1] 1.582839
> hist(ybar, prob = T)
> curve(dnorm(x, mean=100, sd=sqrt(30/20)), 95, 105, add=T)
> qqnorm(ybar)
```

```
> qqline(ybar)
```

Note que para obter o *qq-plot* neste exemplo utilizamos as funções `qqnorm` `qqline` já disponíveis no R para fazer *qq-plot* para distribuição normal.

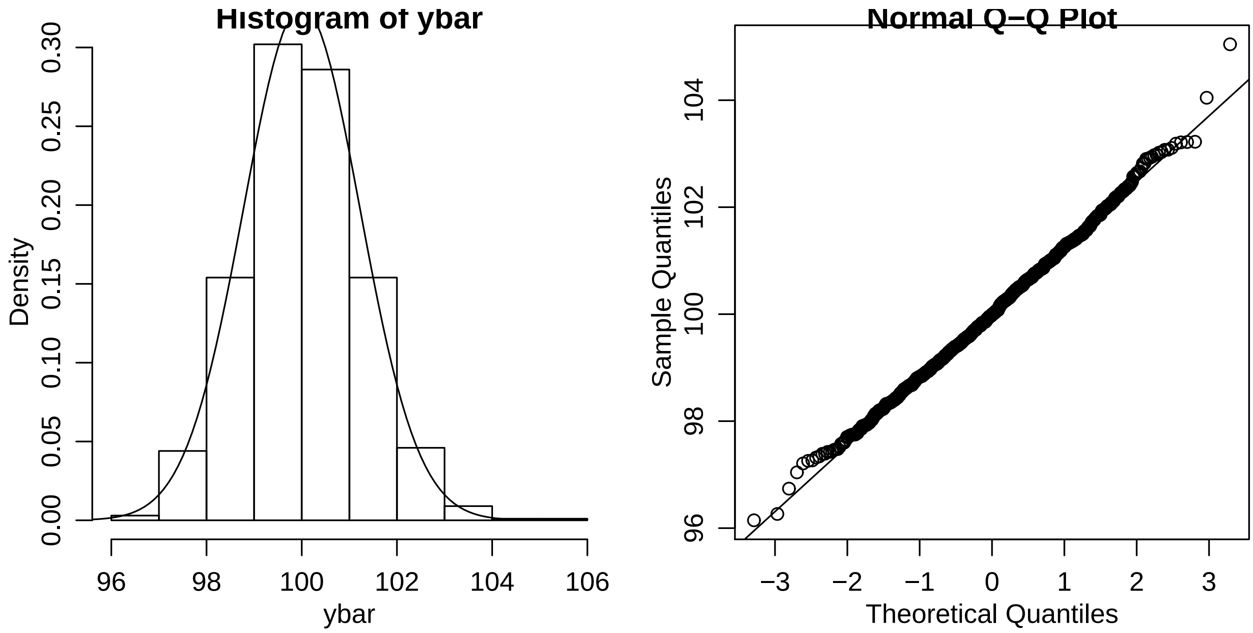

Figura 8: Histograma de uma amostra da distribuição amostral da média e a curva teórica da distribuição e o respectivo *qq-plot*.

3.3 Exercícios

1. Ilustrar usando simulação o resultado que afirma que o estimador $S^2 = \sum \frac{(x_i - \bar{x})^2}{n-1}$ da variância de uma distribuição normal tem distribuição χ^2_{n-1}.
   DICA: Voce pode começar pensando nos passos necessários para ilustrar este resultado:
   - escolha os parâmetros de uma distribuição normal,
   - escolha o tamanho de amostra n e o número de simulações N,
   - gere N amostras de tamanho n,
   - para cada amostra calcule S^2,
   - faça um histograma com os valores S^2 e compare com a curva de uma distrição χ^2_{n-1}.

2. Seja $X_1, \ldots, X_n$ a.a. de uma distribuição $\mathrm{N}(\mu, \sigma^2)$. Ilustrar o resultado que justifica o teste-t para média de uma amostra,

$$\frac{\bar{x} - \mu}{S/\sqrt{n}} \sim t_{n-1}$$

   onde S é o desvio padrão da amostra e n o tamanho da amostra.
   DICA: começe verificando passo a passo, como no exercício anterior, o que é necessário para ilustrar este resultado.

3. Ilustrar o resultado que diz que o quociente de duas variáveis independentes com distribuição χ^2 tem distribuição F.

4 Ilustrando o Teorema Central do Limite

Informalmente o *Teorema Central do Limite* generaliza o **Resultado 3** da Sessão3 diz que se uma váriável aleatória Y tem uma distribuição qualquer com média μ e variância σ^2 então a distribuição de $\bar{y}$ tende para $\mathrm{N}(\mu, \sigma^2/n)$ à medida que aumenta o tamanho da amostra n.

O objetivo desta aula é mostrar como escrever comandos para ilustrar este resultado usando simulação.

5 Ilustrando propriedades de estimadores

5.1 Consistência

Um estimador é consistente quando seu valor se aproxima do verdadeiro valor do parâmetro à medida que aumenta-se o tamanho da amostra. Vejamos como podemos ilustrar este resultado usando simulação. A idéia básica é a seguite:

1. escolher uma distribuição e seus parâmetros,
2. definir o estimador,
3. definir uma sequência crescente de valores de tamanho de amostras,
4. obter uma amostra de cada tamanho,
5. calcular a estatística para cada amostra,
6. fazer um gráfico dos valores das estimativas contra o tamanho de amostra, indicando neste gráfico o valor verdadeiro do parâmetro.

5.1.1 Média da distribuição normal

Seguindo os passos acima vamos:

1. tomar a distribuição Normal de média 10 e variância 4,
2. definir o estimador $\bar{x} = \sum_{i=1}^{n} \frac{x_i}{n}$,
3. escolhemos os tamanhos de amostra $n = 2, 5, 10, 15, 20, \ldots, 1000, 1010, 1020, \ldots, 5000$,
4. fazemos os cálculos e produzimos um gráfico com os comandos abaixo.

```
> ns <- c(2, seq(5, 1000, by=5), seq(1010, 5000, by=10))
> estim <- numeric(length(ns))
> for (i in 1:length(ns)){
>   amostra <- rnorm(ns[i], 10, 4)
>   estim[i] <- mean(amostra)
> }
> plot(ns, estim)
> abline(h=10)
```

5.2 Momentos das distribuições amostrais de estimadores

Para inferência estatística é necessário conhecer a distribuição amostral dos estimadores. Em alguns casos estas distribuições são derivadas analiticamente. Isto se aplica a diversos resultados vistos no curso de **Inferência I**. Por exemplo o resultado visto na Sessão 3: se $Y_1, Y_2, \ldots Y_n \sim \mathrm{N}(\mu, \sigma^2)$ então $\bar{y} \sim \mathrm{N}(\mu, \sigma^2/n)$. Resultados como estes podem ser ilustrados computacionalmente como visto na Sessão 3.

Além disto este procedimento permite investigar distribuições amostrais que não podem ser obtidas analiticamente. Vamos ver um exemplo: considere X uma v.a. com distribuição normal $N(\mu, \sigma^2)$ e seja um parâmetro de interesse $\theta = \mu/\sigma^2$. Obter a esperança e variância de um estimador de $T = \bar{X}/S^2$ onde $\bar{X}$ é a média e S^2 a variância de uma amostra.

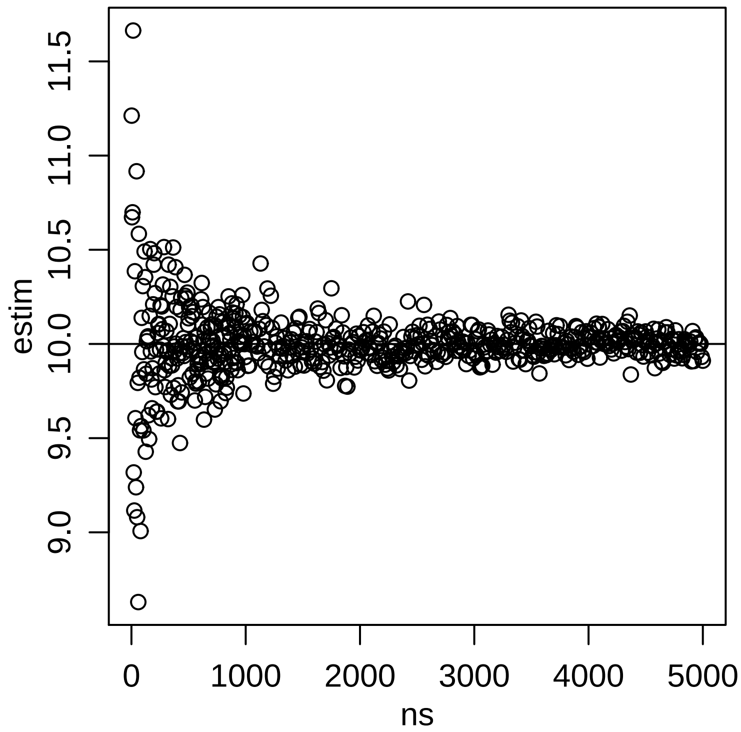

Figura 9: Médias de amostras de diferentes tamanhos.

1. escolher uma distribuição e seus parâmetros, no caso vamos escolher uma $N(180, 64)$,
2. definir um tamanho de amostra, no caso escolhemos $n = 20$,
3. obter por simulação um número N de amostras, vamos usar $N = 1000$,
4. calcular a estatística de interesse para cada amostra,
5. obter as estimativas $\hat{E}[T]$ e $\hat{\text{Var}}[T]$ usando as amostras.

Vamos ver agora comandos do R.

```
> amostras <- matrix(rnorm(20*1000, mean=180, sd=8), nc=1000)
> Tvals <- apply(amostras, 2, function(x) {mean(x)/var(x)})
> ET <- mean(Tvals)
> ET
[1] 3.134504
> VarT <- var(Tvals)
> VarT
[1] 1.179528
```

Nestes comandos primeiro obtemos 1000 amostras de tamanho 20 que armazenamos em uma matrix de dimensão 20×1000, onde cada coluna é uma amostra. A seguir usamos a função `apply` para calcular a quantidade desejada que definimos com `function(x) {mean(x)/var(x)}`. No meu caso acima obtive $\hat{E}[T] \approx 3.13$ e $\hat{\text{Var}}[T] \approx 1.18$. Se voce rodar os comandos acima deverá obter resultados um pouco diferentes (mas não muito!) pois nossas amostras da distribuição normal não são as mesmas.

5.3 Não-tendenciosidade

Fica como exercício, ver abaixo.

5.4 Variância mínima

Fica como exercício, ver abaixo.

5.5 Exercícios

1. Ilustre a consistência do estimador $\hat{\lambda} = 1/\bar{X}$ de uma distribuição exponencial $f(x) = \lambda \exp\{-\lambda x\}$.

2. No exemplo dos momentos das distribuições de estimadores visto em (5.2) ilustramos a obtenção dos momentos para um tamanho fixo de amostra $n = 20$. Repita o procedimento para vários tamanho de amostra e faça um gráfico mostrando o comportamento de $\hat{E}[T]$ e $\hat{\text{Var}}[T]$ em função de n.

3. Estime por simulação a esperança e variância do estimador $\hat{\lambda} = \bar{X}$ de uma distribuição de Poisson de parâmetro λ para um tamanho de amostra $n = 30$. Compara com os valores obtidos analiticamente. Mostre em um gráfico como os valores de $\hat{E}[\lambda]$ e $\hat{\text{Var}}[\lambda]$ variam em função de n.

4. Crie um exemplo para ilustrar a não tendenciosidade de estimadores. Sugestão: compare os estimadores $S^2 = \sum_{i=1}^{n}(x_1 - \bar{x})^2/(n-1)$ e $\sigma^2 = \sum_{i=1}^{n}(x_1 - \bar{x})^2/n$ do parâmetro de variância σ^2 de uma distribuição normal.

5. Crie um exemplo para comparar a variância de dois estimadores. Por exemplo compare por simulação as variâncias dos estimadores $T_1 = \bar{X}$ e $T_2 = (X_1 + X_n)/2$ do parâmetro μ de uma distribuição $\text{N}(\mu, \sigma^2)$, onde X_1 e X_n são os valores mínimo e máximo da amostra, respectivamente.

6 Funções de verossimilhança

A função de verossimilhança é central na inferência estatística. Nesta sessão vamos ver como traçar gráficos de funções de verossimilhança de um parâmetro utilizando o programa R. Também veremos como traçar a função *deviance*, obtida a partir da função de verossimilhança e conveniente em certos casos para representações gráficas, cálculos e inferências.

6.1 Definições e notações

Seja $L(\theta; y)$ a função de verossimilhança. A notação indica que o argumento da função é θ que pode ser um escalar ou um vetor de parâmetros. Nesta sessão consideraremos que é um escalar. O termo y denota valores realizados de uma variável aleatória Y, isto é os valores obtidos em uma amostra.

O valor que maximiza $L(\theta; y)$ é chamado do estimador de máxima verossimilhança e denotado por $\hat{\theta}$. A função de verossimilhança *relativa* ou *normatizada* $R(\theta; y)$ é dada pela razão entre a função de verossimilhança e o valor maximizado desta função, portanto $R(\theta; y) = L(\theta; y)/L(\hat{\theta}; y)$, assumindo valores no intervalo $[0, 1]$. Esta função é útil para comparar todos dos modelos dados pelos diferentes valores de θ com o modelo mais plausível (verossível) para a amostra obtida.

O valor que maximiza a função de verossimilhança é também o que maximiza a a função obtida pelo logarítimo da função de verossimilhança, chamada função de log-verossimilhança, uma vez que a função logarítimo é uma função monotônica. Denotamos a função de log-verossimilhança por $l(\theta; y)$ sendo $l(\theta; y) = \log(L(\theta; y))$. A função de log-verossimilhança é mais adequada para cálculos computacionais e permite que modelos possam ser comparados aditivamente, ao invés de multiplicativamente.

Aplicando-se o logarítimo à função padronizada obtemos $\log\{R(\theta; y)\} = l(\theta; y) - l(\hat{\theta}; y)$, que tem portanto um valor sempre não-positivo. Desta forma esta função pode ser multiplicada por um número negativo arbitrário, e sendo este número -2 obtemos a chamada *função deviance*, $D(\theta; y) = -2\left[l(\theta; y) - l(\hat{\theta}; y)\right]$, onde lembramos que $\hat{\theta}$ é o estimador de máxima verossimilhança de θ. Esta função tem portanto o seu mínimo em zero e quanto maior o seu valor, maior a diferença de plausibilidade entre o modelo considerado e o modelo mais plausível para os dados obtidos na amostra. Esta função combina as vantagens da verossimilhança relativa e da log-verossimilhança sendo portanto conveniente para cálculos computacionais e inferência.

6.2 Exemplo 1: Distribuição normal com variância conhecida

Seja o vetor $(12, 15, 9, 10, 17, 12, 11, 18, 15, 13)$ uma amostra aleatória de uma distribuição normal de média μ e variância conhecida e igual a 4. O objetivo é fazer um gráfico da função de log-verossimilhança.

Solução:

Vejamos primeiro os passos da solução analítica:

1. Temos que $X_1, \ldots, X_n$ onde, neste exemplo $n = 10$, é uma a.a. de $X \sim N(\mu, 4)$,
2. a densidade para cada observação é dada por $f(x_i) = \frac{1}{2\sqrt{2\pi}} \exp\{-\frac{1}{8}(x_i - \mu)^2\}$,
3. a verossimilhança é dada por $L(\mu) = \prod_1^{10} f(\mu; x_i)$,

4. e a log-verossimilhança é dada por

$$
\begin{aligned}
l(\mu) &= \sum_1^{10} \log(f(x_i)) \\
&= -5\log(8\pi) - \frac{1}{8}(\sum_1^{10} x_i^2 - 2\mu \sum_1^{10} x_i + 10\mu^2), \qquad (1)
\end{aligned}
$$

5. que é uma função de μ e portanto devemos fazer um gráfico de $l(\mu)$ *versus* μ tomando vários valores de μ e calculando os valores de $l(\mu)$.

Vamos ver agora uma primeira possível forma de fazer a função de verossimilhança no R.

1. Primeiro entramos com os dados que armazenamos no vetor `x`

```
> x <- c(12, 15, 9, 10, 17, 12, 11, 18, 15, 13)
```

2. e calculamos as quantidades $\sum_1^{10} x_i^2$ e $\sum_1^{10} x_i$

```
> sx2 <- sum(x^2)
> sx <- sum(x)
```

3. agora tomamos uma sequência de valores para μ. Sabemos que o estimador de máxima verossimilhança neste caso é $\hat{\mu} = 13.2$ (este valor pode ser obtido com o comando `mean(x)`) e portanto vamos definir tomar valores ao redor deste ponto.

```
> mu.vals <- seq(11, 15, l = 100)
```

4. e a seguir calculamos os valores de $l(\mu)$ de acordo com a equação acima

```
> lmu <- -5 * log(8 * pi) - (sx2 - 2 * mu.vals * sx + 10 * (mu.vals^2))/8
```

5. e finalmente fazemos o gráfico visto na Figura 10

```
> plot(mu.vals, lmu, type = "l", xlab = expression(mu), ylab = expression(l(mu)))
```

Entretanto podemos obter a função de verossimilhança no R de outras forma mais geral e menos trabalhosa. Sabemos que a função `dnorm()` calcula a densidade $f(x)$ da distribuição normal e podemos usar este fato para evitar a digitação da expressão acima.

- Primeiro vamos criar uma função que calcula o valor da log-verossimilhança para um certo valor do parâmetro e para um certo conjunto de dados,

```
> logvero <- function(mu, dados) {
+     sum(dnorm(dados, mean = mu, sd = 2, log = TRUE))
+ }
```

- a seguir criamos uma sequência adequada de valores de μ e calculamos $l(\mu)$ para cada um dos valores

```
> mu.vals <- seq(11, 15.5, l = 100)
> mu.vals[1:10]
```

```
 [1] 11.00000 11.04545 11.09091 11.13636 11.18182 11.22727 11.27273 11.31818 11.3636
[10] 11.40909
```

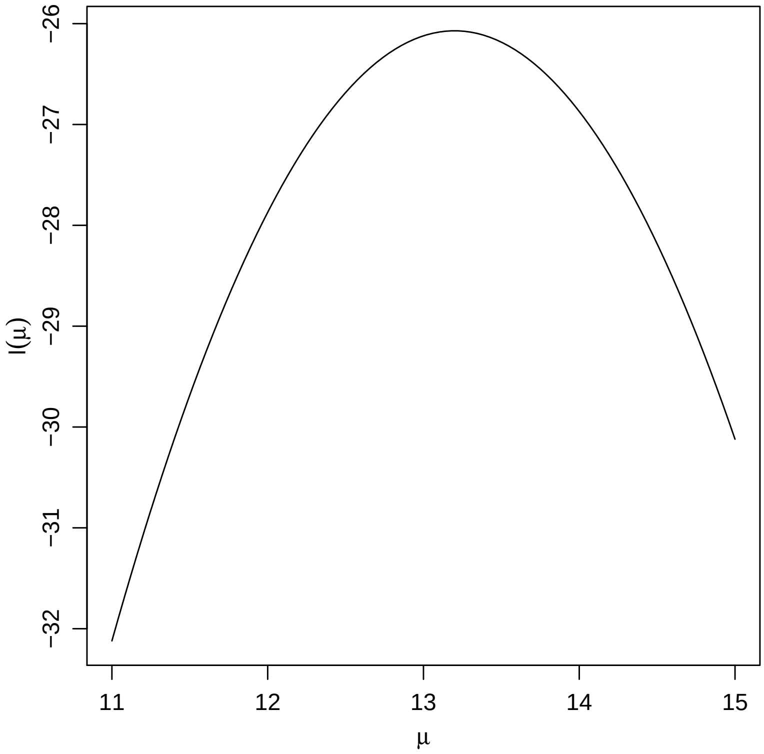

Figura 10: Função de verossimilhança para o parâmetro μ da distribuição normal com variância $\sigma^2 = 4$ com os dados do Exemplo 1.

```
> lmu <- sapply(mu.vals, logvero, dados = x)
> lmu[1:10]

 [1] -32.12086 -31.87344 -31.63119 -31.39410 -31.16218 -30.93542 -30.71383 -30.49741
 [9] -30.28615 -30.08005
```

Note na sintaxe acima que a função `sapply` aplica a função `logvero` anteriormente definida em cada elemento do vetor `mu.vals`.

- Finalmente fazemos o gráfico.

```
> plot(mu.vals, lmu, type = "l", xlab = expression(mu), ylab = expression(l(mu)))
```

Para encerrar este exemplo vamos apresentar uma solução ainda mais genérica que consiste em criar uma função que vamos chamar de `vero.norm.v4` para cálculo da verossimilhança de distribuições normais com σ^2=4. Esta função engloba os comandos acima e pode ser utilizada para obter o gráfico da log-verossimilhança para o parâmetro μ para qualquer amostra obtida desta distribuição.

```
> vero.normal.v4 <- function(mu, dados) {
+     logvero <- function(mu, dados) sum(dnorm(dados, mean = mu, sd = 2,
+         log = TRUE))
+     sapply(mu, logvero, dados = dados)
+ }
> curve(vero.normal.v4(x, dados = x), 11, 15, xlab = expression(mu),
+     ylab = expression(l(mu)))
```

6.3 Exemplo 2: Distribuição Poisson

Considere agora a amostra armazenada no vetor y:

```
> y <- c(5, 0, 3, 2, 1, 2, 1, 1, 2, 1)
```

de uma distribuição de Poisson de parâmetro λ. A função de verossimilhança pode ser definida por:

```
> lik.pois <- function(lambda, dados) {
+     loglik <- function(l, dados) {
+         sum(dpois(dados, lambda = l, log = TRUE))
+     }
+     sapply(lambda, loglik, dados = dados)
+ }
```

E podemos usar esta função para fazer o gráfico da função de verossimilhança como visto à esquerda da Figura 11

```
> lambda.vals <- seq(0, 10, l = 101)
> loglik <- sapply(lambda.vals, lik.pois, dados = y)
> plot(lambda.vals, loglik, ty = "l")
```

E o comando para gerar o gráfico poderia incluir o texto do eixos:

```
> plot(lambda.vals, loglik, type = "l", xlab = expression(lambda),
+     ylab = expression(l(lambda)))
```

ou simplesmente usar:

```
> curve(lik.pois(x, dados = y), 0, 10, xlab = expression(lambda),
+     ylab = expression(l(lambda)))
```

Alternativamente pode-se fazer um gráfico da função deviance, como nos comandos abaixo.

```
> dev.pois <- function(lambda, dados) {
+     lambda.est <- mean(dados)
+     lik.lambda.est <- lik.pois(lambda.est, dados = dados)
+     lik.lambda <- lik.pois(lambda, dados = dados)
+     return(-2 * (lik.lambda - lik.lambda.est))
+ }
> curve(dev.pois(x, dados = y), 0, 10, xlab = expression(lambda),
+     ylab = expression(D(lambda)))
```

Ou fazendo novamente em um intervalo menor

```
> curve(dev.pois(x, dados = y), 0.5, 5, xlab = expression(lambda),
+     ylab = expression(l(lambda)))
```

O estimador de máxima verossimilhança é o valor que maximiza a função de verossimilhança que é o mesmo que minimiza a função deviance. Neste caso sabemos que o estimador tem expressão analítica fechada $\lambda = \bar{x}$ e portanto pode ser obtido diretamente.

```
> lambda.est <- mean(y)
> lambda.est
```

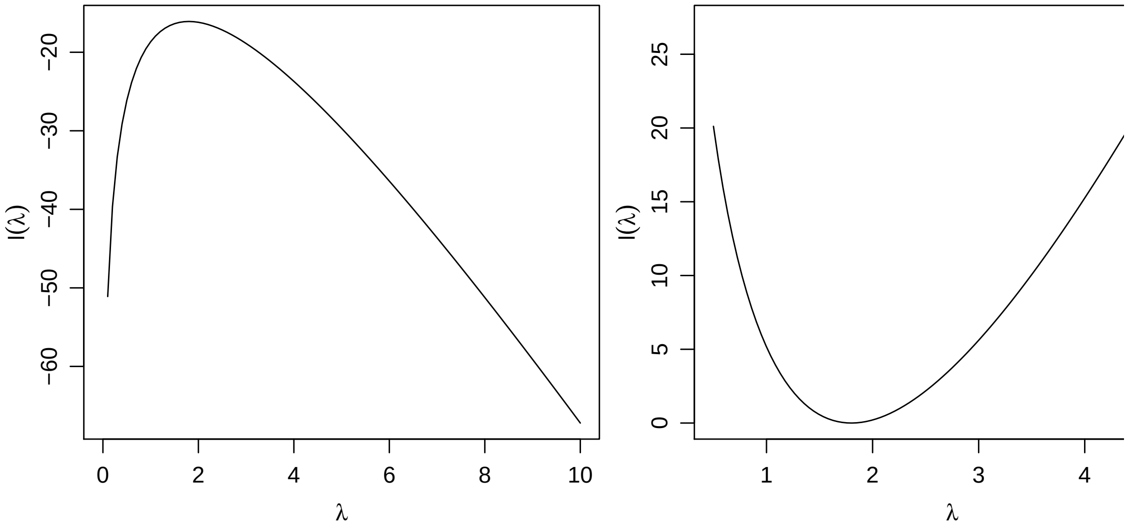

Figura 11: Função de verossimilhança (esquerda) e deviance (direita) para o parâmetro λ da distribuição Poisson.

```
[1] 1.8
```

Caso o estimador não tenha expressão fechada pode-se usar maximização (ou minimização) numérica. Para ilustrar isto vamos encontrar a estimativa do parâmetro da Poisson e verificar que o valor obtido coincide com o valor dado pela expressão fechada do estimador. Usamos o função `optimise()` para encontrar o ponto de mínimo da função deviance.

```
> optimise(dev.pois, int = c(0, 10), dados = y)

$minimum
[1] 1.800004

$objective
[1] 1.075264e-10
```

A função `optimise()` é adequada para minimizações envolvendo um único parâmetro. Para dois ou mais parâmetros deve-se usar a função `optim()` ou `nlminb()`.

Finalmente os comandos abaixo são usados para obter graficamente o intervalo de confiança (a 95%) baseado na função deviance.

```
> curve(dev.pois(x, dados = y), 0.8, 3.5, xlab = expression(lambda),
+     ylab = expression(l(lambda)))
> L.95 <- qchisq(0.95, df = 1)
> abline(h = L.95)
```

Os limites do intervalo são dados pela interseção dessa função com o valor do quantil da distribuição χ^2 para o nível de significância desejado.

```
> lim.fc <- function(lambda) dev.pois(lambda, dados = y) - L.95
> ic2.lambda <- c(inf = uniroot(lim.fc, c(0, lambda.est))$root, sup = uniroot(lim.fc,
+     c(lambda.est, max(y)))$root)
> ic2.lambda

     inf      sup
1.091267 2.764221
```

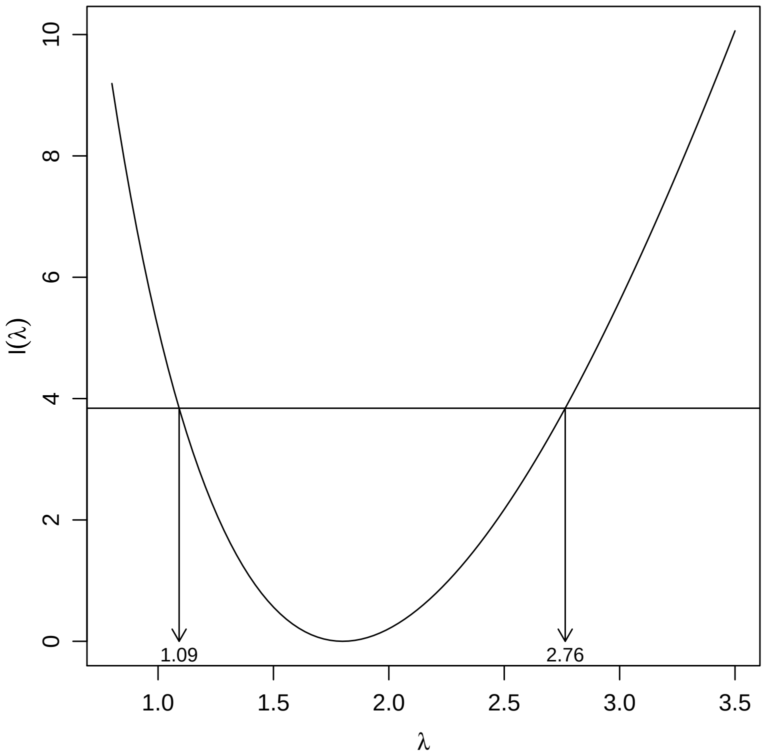

Figura 12: Intervalo de confiança baseado na deviance para o parâmetro λ da distribuição Poisson.

E adicionados ao gráfico com

```
> arrows(ic2.lambda, L.95, ic2.lambda, 0, len = 0.1)
> text(ic2.lambda, 0, round(ic2.lambda, dig = 2), pos = 1, cex = 0.8,
+     offset = 0.3)
```

6.4 Exemplo 3: Distribuição normal com variância desconhecida

Vamos agora revisitar o Exemplo 1 desta seção, usando os mesmos dados porém agora sem assumir que a variância é conhecida. Portanto temos agora dois parâmetros sobre os quais queremos fazer inferência: μ e σ . O objetivo é fazer um gráfico 3-D da função de log-verossimilhança de dois argumentos $l(\mu, \sigma)$.

Solução:
Vejamos primeiro os passos da solução analítica:

1. Temos que $X_1, \ldots, X_n$ onde, neste exemplo $n = 10$, é uma a.a. de $X \sim N(\mu, \sigma^2)$,
2. a densidade para cada observação é dada por $f(x_i) = \frac{1}{\sigma\sqrt{2\pi}} \exp\{-\frac{1}{2\sigma^2}(x_i - \mu)^2\}$,
3. a verossimilhança é dada por $L(\mu, \sigma) = \prod_1^{10} f(\mu, \sigma; x_i)$,
4. e a log-verossimilhança é dada por

$$\begin{aligned} l(\mu, \sigma) &= \sum_1^{10} \log(f(x_i)) \\ &= -5\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(\sum_1^{10} x_i^2 - 2\mu\sum_1^{10} x_i + 10\mu^2), \end{aligned} \quad (2)$$

5. que é uma função de μ e σ e portanto devemos fazer um gráfico tridimensional de $l(\mu, \sigma)$ *versus* μ e σ tomando vários valores de pares (μ, σ) e calculando os valores correspondentes de $l(\mu, \sigma)$.

Assim como no Exemplo 1 poderíamos calcular a verossimilhança fazendo as contas "passo a passo"da função acima, ou então usando a função `dnorm()`. Neste exemplo vamos fazer apenas da segunda forma, ficando a primeira como exercício para o leitor.

1. Primeiro entramos com os dados que armazenamos no vetor `x`. Vamos também calcular as estimativas de máxima verossimilhança.

```
> x <- c(12.1, 15.4, 9.8, 10.1, 17.4, 12.3, 11, 18.2, 15.4, 13.3,
+     13.8, 12.7, 15.2, 10.3, 9.9, 11.5, 14, 12.1, 11.2, 11.9, 11.1,
+     12.5, 13.5, 14.8, 12.1, 12.5, 9.7, 11.3, 8.6, 15.9, 12.8, 13.6,
+     13.8, 15.7, 15.5)
> pars.MV <- c(mu = mean(x), sd = sqrt(var(x) * (length(x) - 1)/length(x)))
> pars.MV

       mu        sd
12.885714  2.248954
```

2. a seguir vamos criar uma função que calcula o valor da log-verossimilhança para um certo par de valores dos parâmetros (média e desvio padrão, nesta ordem) e para um certo conjunto de dados,

```
> logveroN <- function(pars, dados) sum(dnorm(dados, mean = pars[1],
+     sd = pars[2], log = TRUE))
```

3. a seguir criamos uma sequência adequada de pares de valores de (μ, σ) e calculamos $l(\mu, \sigma)$ para cada um dos pares.

```
> par.vals <- expand.grid(mu = seq(5, 20, l = 100), sd = seq(1, 12.2,
+     l = 100))
> dim(par.vals)

[1] 10000     2

> head(par.vals)

        mu sd
1 5.000000  1
2 5.151515  1
3 5.303030  1
4 5.454545  1
5 5.606061  1
6 5.757576  1

> tail(par.vals)

            mu   sd
9995  19.24242 12.2
9996  19.39394 12.2
```

```
9997  19.54545 12.2
9998  19.69697 12.2
9999  19.84848 12.2
10000 20.00000 12.2

> par.vals$logL <- apply(par.vals, 1, logveroN, dados = x)
> head(par.vals)

        mu sd      logL
1 5.000000  1 -1208.903
2 5.151515  1 -1167.486
3 5.303030  1 -1126.873
4 5.454545  1 -1087.064
5 5.606061  1 -1048.058
6 5.757576  1 -1009.856
```

Note na sintaxe acima que a função `apply` aplica a função `logveroN` a cada par de valores em cada linha de `par.vals`. Ao final o objeto |par.vals| contém na terceira coluna os valores da log-verossimilhança correspondentes as valores dos parâmetros dados na primeira e segunda colunas.

4. O gráfico 3-D da função pode ser visualizado de três formas alternativas como mostrado na Figura 13: como uma superfície 3D gerada pela função `persp()`, como um mapa de curvas de isovalores obtido com `image()`, ou ainda como um mapa de cores correspondentes aos valores gerado por `image()`.

```
> with(par.vals, persp(unique(mu), unique(sd), matrix(logL, ncol = length(unique(sc
+     xlab = expression(mu), ylab = expression(sigma), zlab = expression(l(mu,
+         sigma)), theta = 30, phi = 30))
> with(par.vals, contour(unique(mu), unique(sd), matrix(logL, ncol = length(unique(
+     xlab = expression(mu), ylab = expression(sigma), levels = seq(-120,
+         -75, by = 5)), ylim = c(0, 12))
> points(pars.MV[1], pars.MV[2], pch = 4, cex = 1.5)
> with(par.vals, image(unique(mu), unique(sd), matrix(logL, ncol = length(unique(sc
+     xlab = expression(mu), ylab = expression(sigma), breaks = seq(-120,
+         -75, by = 5), col = gray(seq(0.3, 1, length = 9))))
> points(pars.MV[1], pars.MV[2], pch = 4, cex = 1.5)
```

Notas:

- a obtenção da função foi necessário especificar faixas de valores para μ e σ. A definição desta faixa foi feita após várias tentativas pois depende do problema, em especial do número e variabilidade dos dados.
- as funções gráficas utilizadas requirem: dois vetores de tamanhos n_1 e n_2 com os valores dos argumentos da função e os valores da função em uma matrix de dimensão $n_1 \times n_2$. Por isto usamos `unique()` para extrair os valores dos argumentos, sem repeti-los e `matrix()` para os valores da função.
- na função `perp()` as argumentos `theta` e `phi` são utilizados para rotacionar o gráfico a fim de se obter uma melhor visualização.

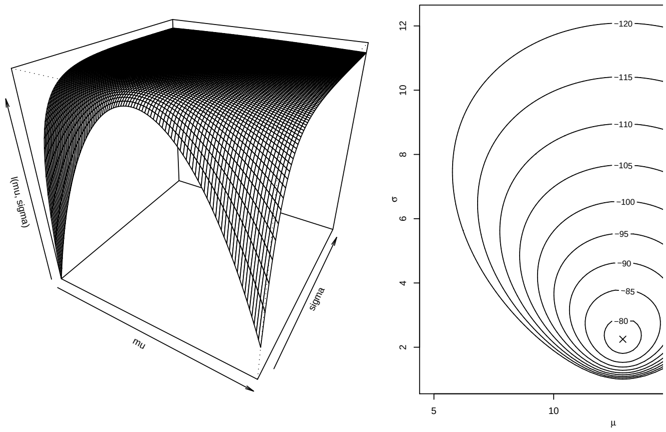

Figura 13: Função de verossimilhança para os parâmetros μ e σ da distribuição normal com os dados do Exemplo 1.

- o valor das estimativas de máxima verossimilhança são indicados por `x` nos dois últimos gráficos. Neste caso eles foram encontrados facilmente como mostrado acima no objeto `pars.MV` pois podem ser obtidos analiticamente. De forma mais geral, a função `fitdistr()` do pacote **MASS** poide ser usada para encontrar estimativas de máxima verossimilhança.

```
> require(MASS)
> MV <- fitdistr(x, "normal")
> MV

      mean          sd
  12.8857143    2.2489544
 ( 0.3801427) ( 0.2688015)
```

6.5 Exercícios

1. Seja a amostra abaixo obtida de uma distribuição Poisson de parâmetro λ.
   5 4 6 2 2 4 5 3 3 0 1 7 6 5 3 6 5 3 7 2
   Obtenha o gráfico da função de log-verossimilhança.

2. Seja a amostra abaixo obtida de uma distribuição Binomial de parâmetro p e com $n = 10$.
   7 5 8 6 9 6 9 7 7 7 8 8 9 9 9
   Obtenha o gráfico da função de log-verossimilhança.

3. Seja a amostra abaixo obtida de uma distribuição χ^2 de parâmetro ν.
   8.9 10.1 12.1 6.4 12.4 16.9 10.5 9.9 10.8 11.4
   Obtenha o gráfico da função de log-verossimilhança.