Parametros_estimadores

May 24, 2021

[20]:
```
library(tidyverse)
library(gridExtra)
## define tamanho das figuras
options(repr.plot.width=6, repr.plot.height=4)
```

0.1 População e amostra. Amostragem e inferência

População estatística: todos os elementos que podem compor uma amostra. Podem ser medidas como comprimentos, temperaturas, velocidades, etc.

Unidade amostral: um único elemento da população.

Censo: o levantamento de *todos* os elementos da população.

Amostra: um subconjunto extraído da população.

Tamanho populacional (N): o número de elementos da população.

Tamanho amostral (n): o número de elementos da amostra.

0.2 Parâmetros e estimadores

Parâmetro: a medida que descreve uma característica da *população*. Ex.: a média (μ) ou a variância (σ^2) populacional.

Estimador ou **Estatística**: Uma medida que descreve uma característica da *amostra*. Ex.: a média amostral ($\overline{X}$) ou a variância amostral (s^2).

Estimativa: é o valor numérico assumido pelo estimador. Ex. o valor número da média ou variância amostral

0.2.1 Medidas de tendência central: média aritimética

$$\mu = \frac{X_1 + X_2 + X_3 + \cdots + X_N}{N} = \frac{\sum_{i=1}^{N} X_i}{N}$$

Quando nos referimos a uma amostra com **n** elementos, a média aritmética **amostral** ($\overline{X}$) é dada por:

$$\overline{X} = \frac{X_1 + X_2 + X_3 + \cdots + X_n}{n} = \frac{\sum_{i=1}^{n} X_i}{n}$$

0.2.2 Medidas de variação: variância, desvio padrão, desvio padrão

A variância e o desvio padrão populacional

$$\sigma^2 = \frac{\sum_{i=1}^{N} (X_i - \mu)^2}{N}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} (X_i - \mu)^2}{N}}$$

A variância e o desvio padrão amostral

$$s^2 = \frac{\sum_{i=1}^{N} (X_i - \overline{X})^2}{n - 1}$$

$$s = \sqrt{\frac{\sum_{i=1}^{N} (X_i - \overline{X})^2}{n - 1}}$$

[2]:
```
x = c(5.0, 4.1, 6.6, 4.5, 4.9, 7.7, 4.0, 5.0, 7.0, 7.9)
```

[3]:
```
n = length(x)
xb = mean(x)
xb
```

5.67

[4]:
```
var_a = sum((x - xb)^2)/(n - 1)
var_a
desv_a = sqrt(var_a)
desv_a
```

2.20455555555556

1.48477458072111

[5]:
```
sd(x)
```

1.48477458072111

0.3 Modelos probabilísticos e distribuições de frequência

0.3.1 O modelo normal de probabilidade

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

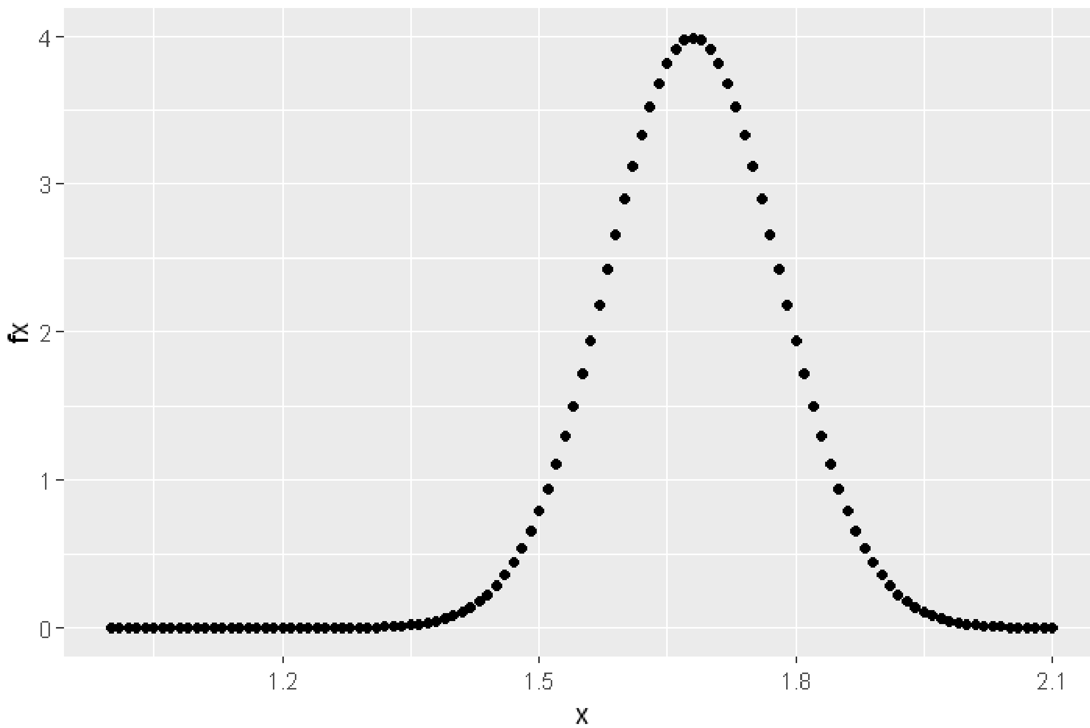
$$X \sim \mathcal{N}(\mu, \sigma^2)$$

[6]:
```
mi = 10
sg = 2
x = c(4, 10)
fx = (1/(sqrt(2 * pi * sg^2))) * exp((-1/2) * ((x - mi)/sg)^2)
fx
```

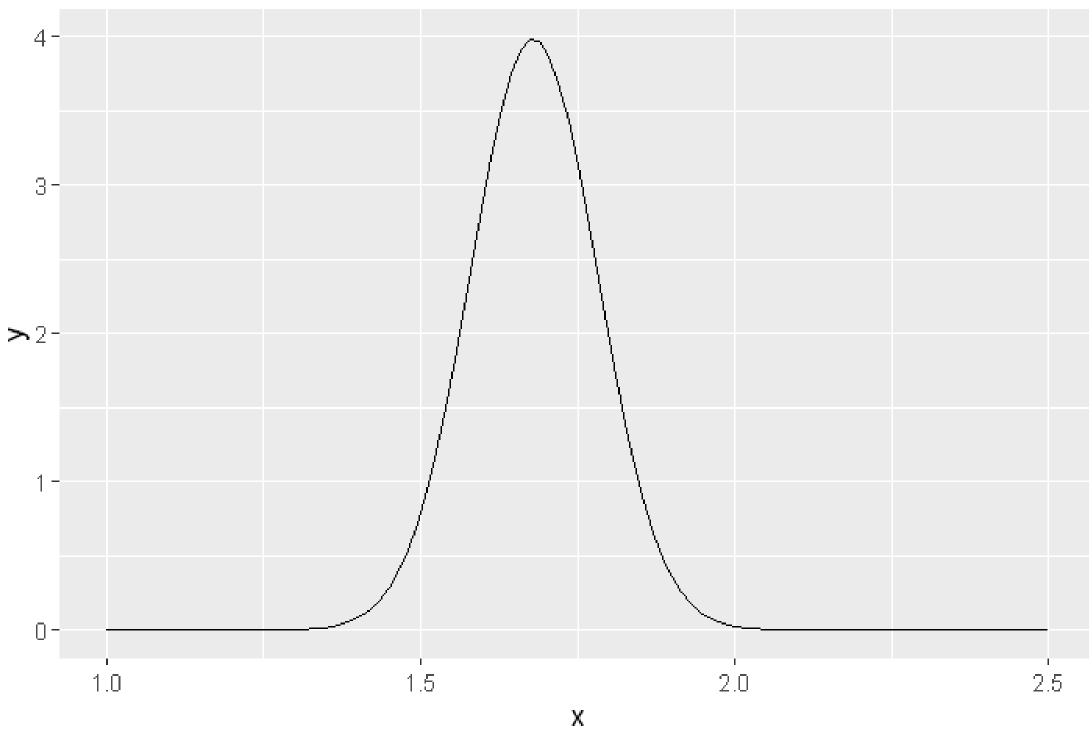
1. 0.002215924205969 2. 0.199471140200716

[7]:
```
x = seq(1, 2.1, 0.01)
fx = dnorm(x, mean = 1.68, sd = 0.10)
```

[8]:
```
ggplot(data.frame(x = x, fx = fx)) +
    geom_point(aes(x, fx))
```



[9]:
```
ggplot(data.frame(x = c(1,2.5))) +
  stat_function(aes(x = x), fun = dnorm,
                args = list(mean = 1.68, sd = 0.1))
```



0.3.2 Alguns (vários) fenômenos podem ser descritos por um modelo normal

[10]:
```
ie = read.csv("datasets/IE_BICT_2019.csv", sep = ";", dec = ',')
dim(ie)

temp <- data.frame(tm = datasets::nhtemp)
dim(temp)
```

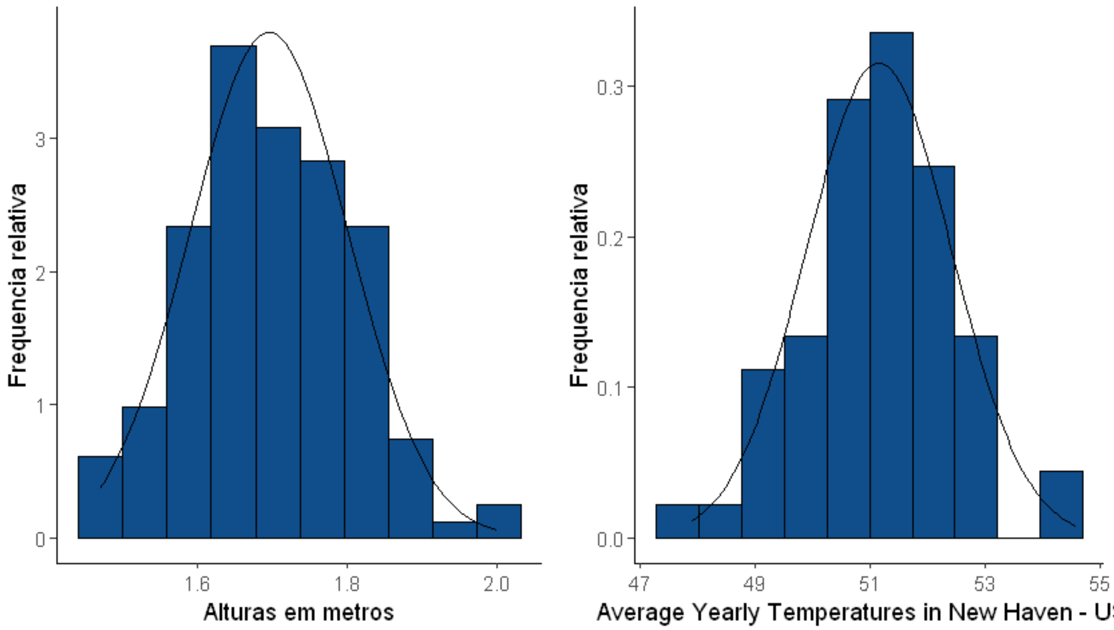
1. 139 2. 2

1. 60 2. 1

[11]:
```
alt_plt <- ggplot(ie, aes(x = Altura)) +
   geom_histogram(aes(y =..density..),
                  fill = 'dodgerblue4', color = 'black', bins = 10) +
   stat_function(fun = dnorm, args = list(mean = mean(ie$Altura, na.rm = T),
                                          sd = sd(ie$Altura, na.rm = T))) +
   labs(x = "Alturas em metros",
        y = "Frequencia relativa") +
   theme_classic()

temp_plt <- ggplot(temp, aes(x = tm)) +
   geom_histogram(aes(y =..density..),
                  fill = 'dodgerblue4', color = 'black', bins = 10) +
```

```
    stat_function(fun = dnorm, args = list(mean = mean(temp$tm, na.rm = T),
                                           sd = sd(temp$tm, na.rm = T))) +
    labs(x = "Average Yearly Temperatures in New Haven - USA",
         y = "Frequencia relativa") +
    theme_classic()

options(repr.plot.width=7, repr.plot.height=4)
grid.arrange(alt_plt, temp_plt, ncol = 2)
```

```
Warning message:
"Removed 1 rows containing non-finite values (stat_bin)."Don't know how to
automatically pick scale for object of type ts. Defaulting to continuous.
```



0.3.3 Fazendo predições com o modelo normal

Qual a probabilidade de encontrar *ao acaso* uma pessoa mais baixa que 1.80 m?

[12]:
```
xb = mean(ie$Altura, na.rm = T)
dp = sd(ie$Altura, na.rm = T)
xb; dp
```

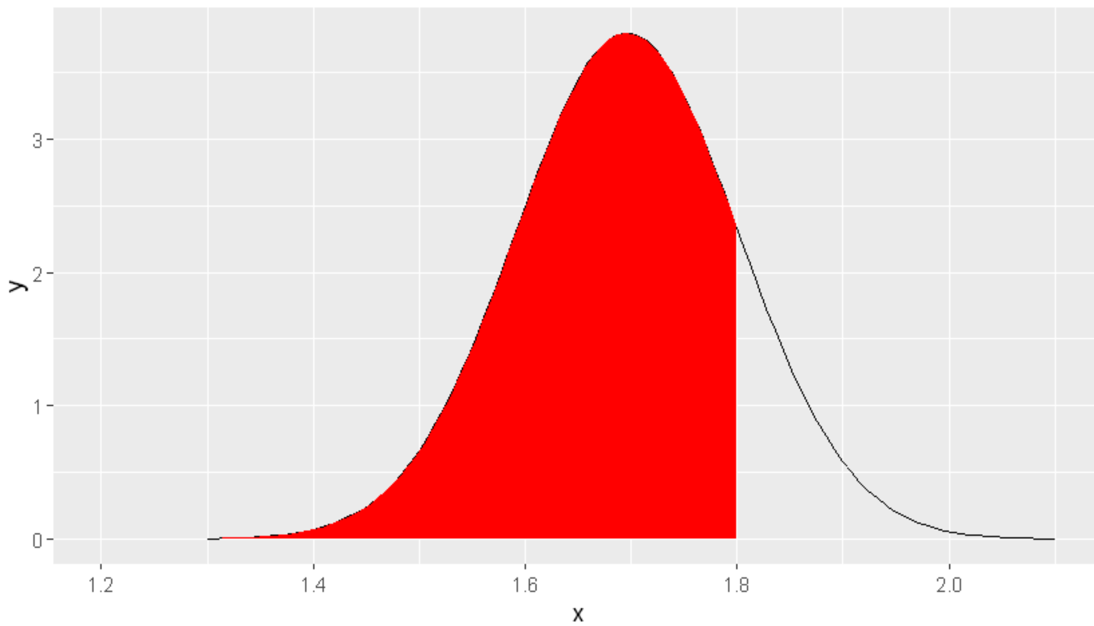
1.69630434782609

0.105048310658474

[13]:
```
px = pnorm(1.80, mean = xb, sd = dp, lower.tail = TRUE)
px
```

0.838208941574631

[14]:
```
ggplot(data.frame(x = c(1.3, 2.1))) +
  stat_function(aes(x = x), fun = dnorm,
                args = list(mean = xb, sd = dp)) +
  geom_area(stat = 'function', fun = dnorm, fill = 'red',
            args = list(mean = xb, sd = dp),
            xlim = c(1.2, 1.8))
```



0.4 Amostrando uma população estatística

0.4.1 O Teorema Central do Limite

> DEFINIÇÃO DO TCL: Seja uma popuilação estatística com média μ e desvio padrão σ. A distribuição das médias amostrais desta população tenderá a apresentar uma distribuição normal de probabilidades com média μ e desvio padrão $\frac{\sigma}{\sqrt{(n)}}$ à medida que o tamanho amostral n aumenta, ainda que a dsitribuição das observações originais **não possua** uma distribuição normal.

Segundo o TCL, as médias amostrais $\overline{X}$ de um experimento distribuem-se como:

$$\overline{X} \sim \mathcal{N}(\mu_{\overline{X}}, \sigma^2_{\overline{X}})$$

em que

$$\mu_{\overline{X}} = \mu$$

$$\sigma^2_{\overline{X}} = \frac{\sigma^2}{n}$$

0.5 Inferência estatística: o intervalo de confiança

DEFINIÇÃO: é um intervalo de valores associado a um determinado nível de significância (α). Quando dizemos que um intervalo foi calculado a um nível de confiança de 95% ($1-\alpha$), estamos dizendo que a probabilidade dos limites do IC conterem o valor da média populacional μ é de 95%.

Distribuiçao t de Student Quando não conhecemos μ e σ e as amostras são pequenas (ex. $n < 30$), a dsitribuiçao normal não é a melhor aproximação para o comportamento das médias amostrais. Nestes casos, substituímos a distribuição normal pela **Distribuição** t **de Student**, sendo o intervalo de confiança obtido por:

$$IC_{1-\alpha} = \overline{X} \pm t_{\alpha/2,gl} \times \frac{s}{\sqrt{n}}$$

Em que α continua sendo o nível de significância e gl é definido como os **graus de liberdade**. Neste caso, os graus de liberdade são dados por:

$$gl = n - 1$$

[15]:
```
x = c(1.97, 1.83, 1.62, 1.77, 1.63, 1.68, 1.63, 1.68, 1.69, 1.85)
(xb = mean(x))
(sd = sd(x))
(n = length(x))
```

1.735

0.11645218379709

10

[16]:
```
alfa = 0.05
t = qt(1-alfa/2, df = n - 1)
t
```

2.2621571627982

[17]:
```
(IC_sup = xb + t * sd/sqrt(n))
(IC_sup = xb - t * sd/sqrt(n))
```

1.81830487389461

1.65169512610539

[18]:
```
muX = round(xb,2)
sigmaX = round(dp,2)
x1 <- 1.8
n1 <- 2
n2 <- 10

tam <- 20
```

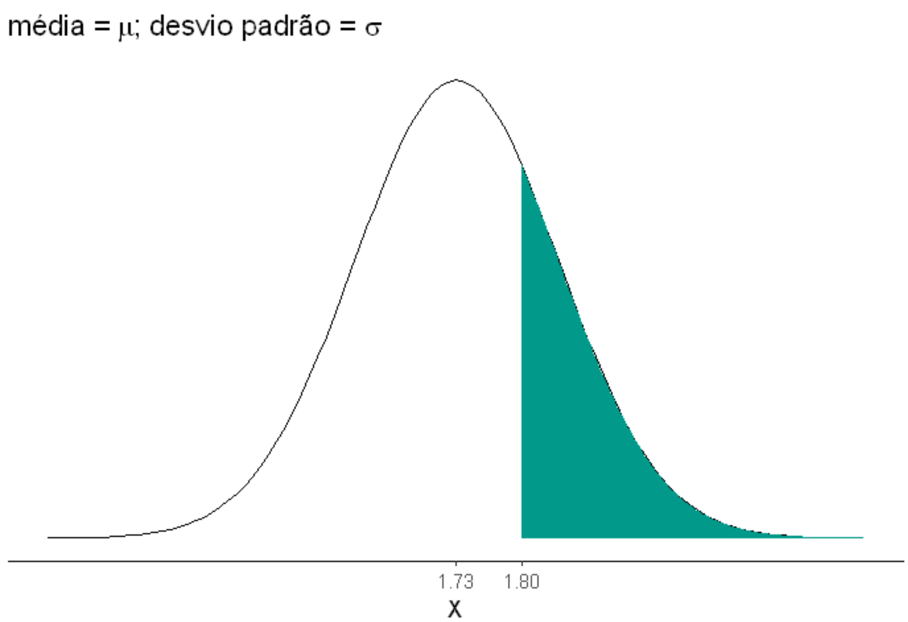
```r
lim_x <- muX + c(-4,4) * sigmaX
exprx <- expression(
   paste('média = ', mu, '; desvio padrão = ', sigma))
expr1 <- expression(
   paste('média = ', mu, '; desvio padrão = ', frac(sigma, sqrt('n'["1"]))))
expr2 <- expression(
   paste('média = ', mu, '; desvio padrão = ', frac(sigma, sqrt('n'["2"]))))

p1 <- ggplot(data.frame(x = lim_x), aes(x = x)) +
  stat_function(fun = dnorm,
                args = list(mean = muX,
                            sd = sigmaX)) +
   geom_area(stat = "function", fun = dnorm,
             args = list(mean = muX,
                         sd = sigmaX),
             fill = "#00998a",
             xlim = c(x1, lim_x[2])) +
   labs(x = "X", y = "") +
   scale_y_continuous(breaks = NULL) +
   scale_x_continuous(breaks = c(muX, x1)) +
   theme(axis.text.x = element_text(size = tam),
         axis.title.x = element_text(size = tam),
         plot.title = element_text(hjust = 0.5, size = tam)) +
   ggtitle(exprx) +
   theme_classic()

p2 <- ggplot(data.frame(x = lim_x), aes(x = x)) +
  stat_function(fun = dnorm,
                args = list(mean = muX,
                            sd = sigmaX/sqrt(n1))) +
   geom_area(stat = "function", fun = dnorm,
             args = list(mean = muX,
                         sd = sigmaX/sqrt(n1)),
             fill = "#00998a",
             xlim = c(x1, lim_x[2])) +
   labs(x = expression(bar("X")), y = "") +
   scale_y_continuous(breaks = NULL) +
   scale_x_continuous(breaks = c(muX, x1)) +
   theme(axis.text.x = element_text(size = tam),
         axis.title.x = element_text(size = tam),
         plot.title = element_text(hjust = 0.5, size = tam)) +
   ggtitle(expr1) +
   theme_classic()

p3 <- ggplot(data.frame(x = lim_x), aes(x = x)) +
  stat_function(fun = dnorm,
                args = list(mean = muX,
```

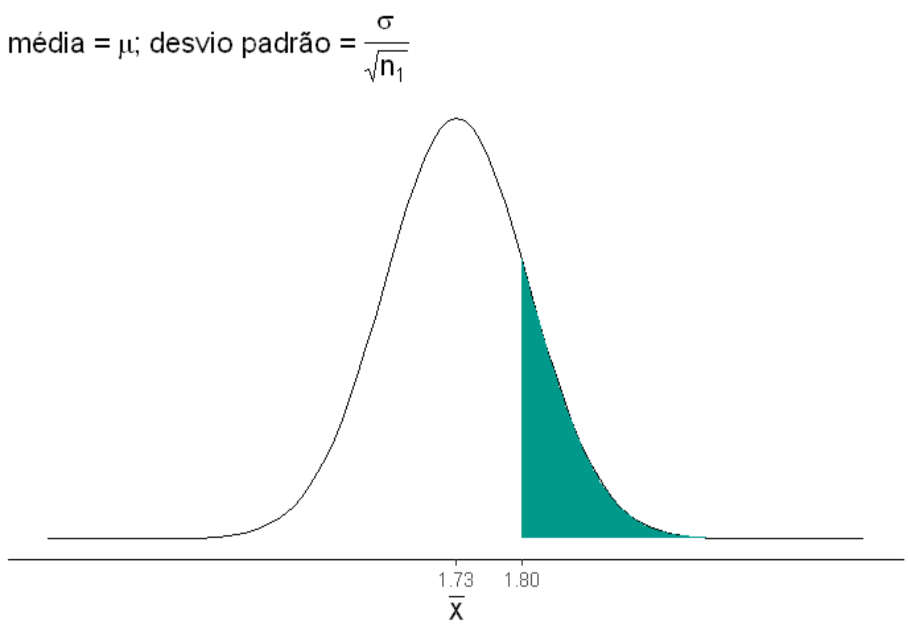
```
                              sd = sigmaX/sqrt(n2))) +
   geom_area(stat = "function", fun = dnorm,
             args = list(mean = muX,
                         sd = sigmaX/sqrt(n2)),
             fill = "#00998a",
             xlim = c(x1, lim_x[2])) +
   labs(x = expression(bar("X")), y = "") +
   scale_y_continuous(breaks = NULL) +
   scale_x_continuous(breaks = c(muX, x1)) +
   theme(axis.text.x = element_text(size = tam),
         axis.title.x = element_text(size = tam),
         plot.title = element_text(hjust = 0.5, size = tam)) +
   ggtitle(expr2) +
   theme_classic()

options(repr.plot.width=6, repr.plot.height=12)
pn <- grid.arrange(p1,p2,p3, ncol = 1)

(area <- round(pnorm(x1, mean = muX, sd = sigmaX, lower.tail = F), 3))
(area1 <- round(pnorm(x1, mean = muX, sd = sigmaX/sqrt(n1), lower.tail = F), 3))
(area2 <- round(pnorm(x1, mean = muX, sd = sigmaX/sqrt(n2), lower.tail = F), 3))
```

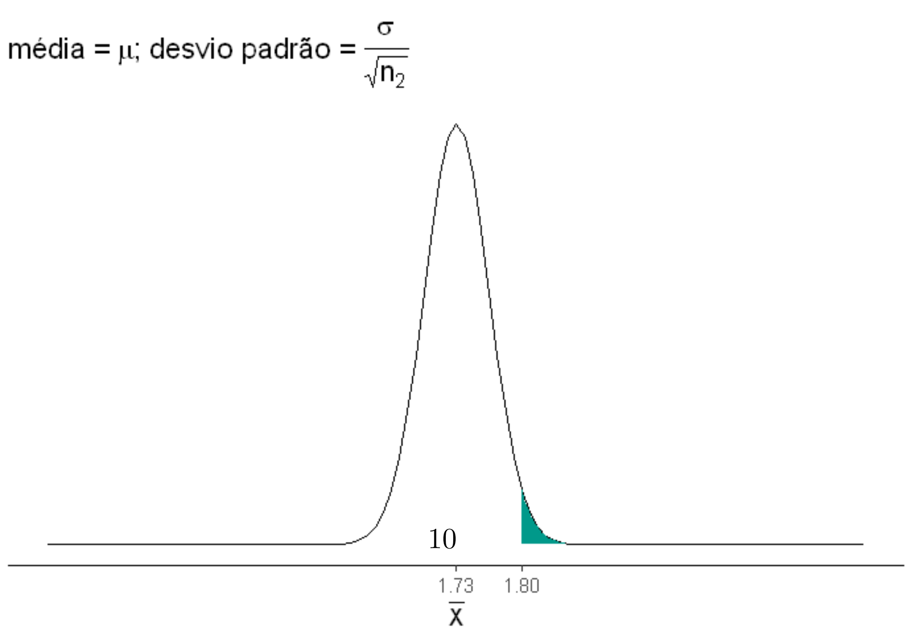
0.262

0.184

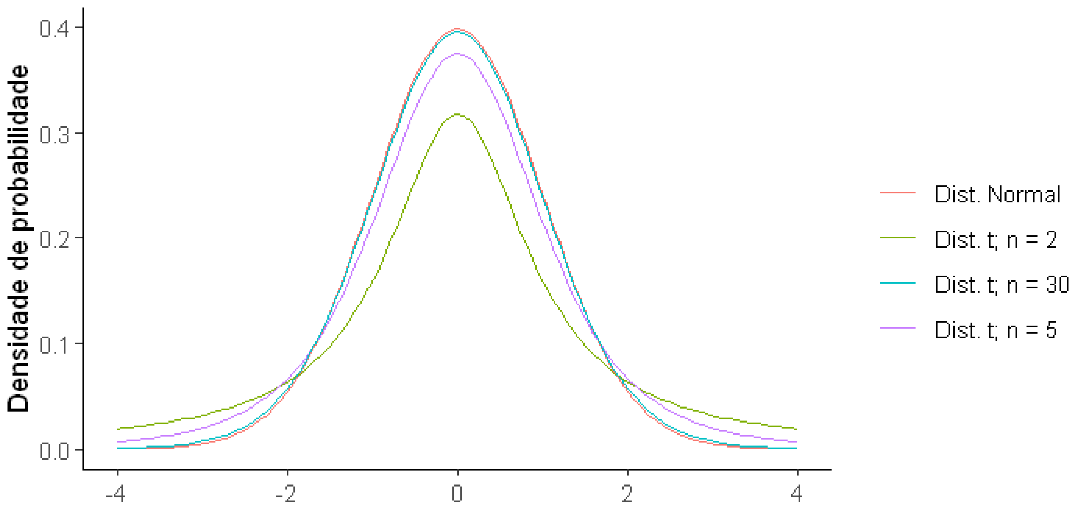
0.022


média = μ; desvio padrão = σ
1.73
1.80
X


média = μ; desvio padrão = $\frac{\sigma}{\sqrt{n_1}}$
1.73
1.80
$\bar{X}$


média = μ; desvio padrão = $\frac{\sigma}{\sqrt{n_2}}$
1.73
1.80
$\bar{X}$

[19]:
```r
options(repr.plot.width=6, repr.plot.height=3)
n1 <- 2
n2 <- 5
n3 <- 30
ggplot(data = data.frame(x = c(-4,4)),
       mapping = aes(x = x)) +
  stat_function(mapping = aes(color = "Dist. Normal"),
                fun = dnorm) +
  stat_function(mapping = aes(color = paste("Dist. t; n =",n1)),
                fun = dt,
                args = list(df = n1-1)) +
  stat_function(mapping = aes(color = paste("Dist. t; n =",n2)),
                fun = dt,
                args = list(df = n2-1)) +
  stat_function(mapping = aes(color = paste("Dist. t; n =",n3)),
                fun = dt,
                args = list(df = n3-1)) +
  labs(colour = "", y = "Densidade de probabilidade", x = "") +
  theme_classic()
```



1 Leitura

1. Princípios de Estatística em Ecologia.

- Capítulo 2. Tópico: **Variável aleatória normal**
- Capítulo 3. **Inteiro**

- Capítulo 4. **Inteiro**

Para a próxima aula

1. Princípios de Estatística em Ecologia.

- Capítulo 5. Três Estruturas para Análises Estatísticas. **Inteiro**
- Capítulo 6. Delineando Estudos de Campo com Sucesso. **Inteiro**
- Capítulo 6. Um Bestiário de Delçineamentos Experimentais e Amostrais. **Inteiro**