Métodos de Estatística Aplicada com Python

Aula 7

Carlos Góes[1]

[1]Pós-Graduação em Ciência de Dados
Instituto de Educação Superior de Brasília

2017

Sumário

Sumário

Erro padrão e intervalo de confiança

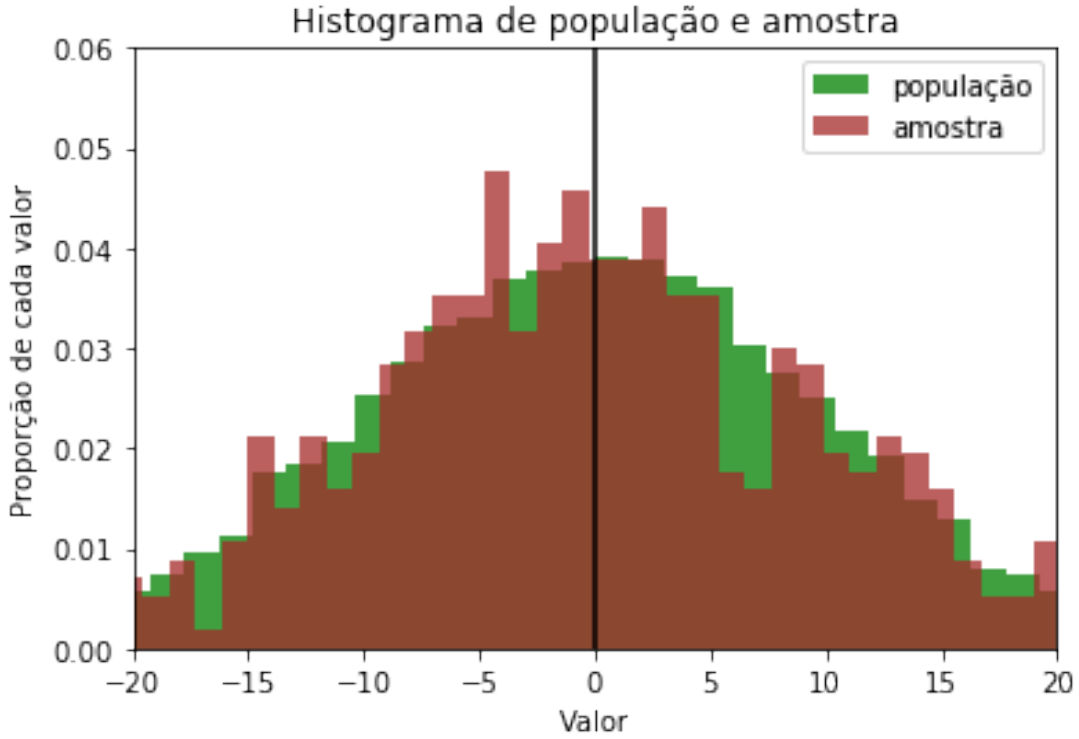
Intuição

- Um dos objetivos de experimentos estatísticos é conseguir descrever, por meio de estatísticas observadas, os parâmetros não-observados de uma população
- Por isso, é importante ter amostras aleatórias que sejam representativas e não-viesadas da população de interesse

Erro padrão e intervalo de confiança

Intuição

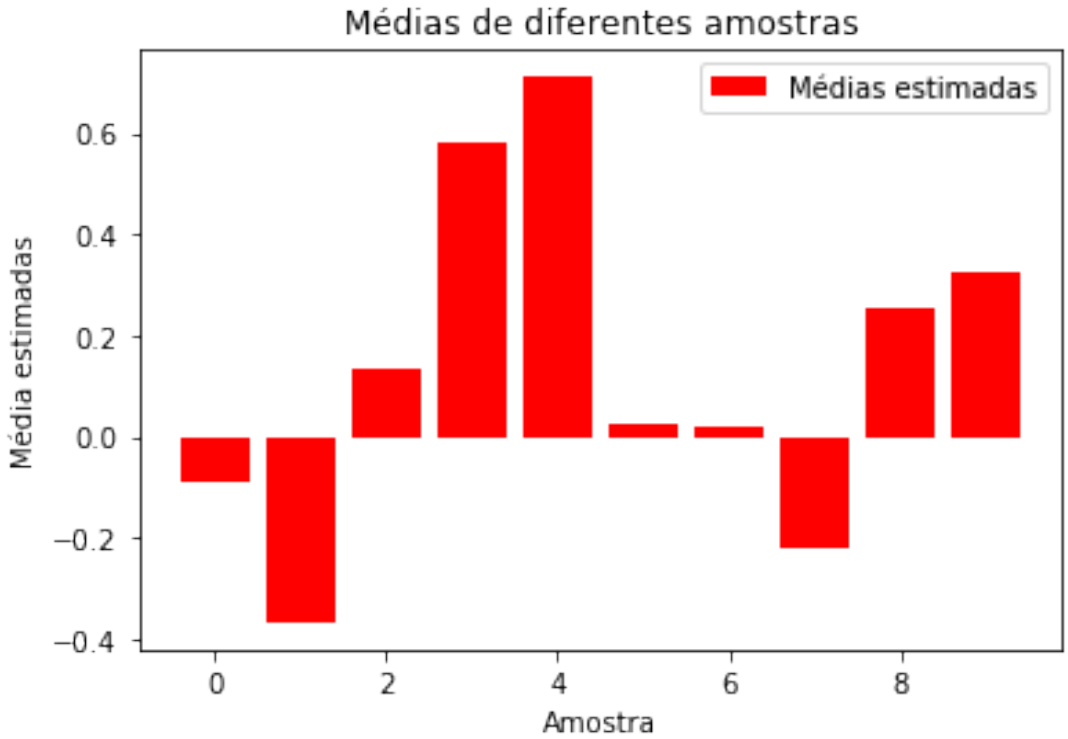
- Como já vimos, mesmo uma amostra representativa não é exatamente idêntica à população.



Erro padrão e intervalo de confiança

Intuição

- Existe, portanto, uma incerteza própria do processo de amostragem, de tal modo que se tirarmos médias de amostras representativas diferentes, elas vão ser um pouco diferentes



Erro padrão e intervalo de confiança

Teorema do Limite Central

- A variabilidade de médias de amostras representativas seguem um padrão interessante: elas aproximam uma distribuição normal
- E a média das médias de amostras representativas tendem a ser muito próximas da média da população: é o que se chama de teorema do limite central
- Vamos, por exemplo, repetir o experimento de tirar médias de jogar um dado muitas vezes e ver a distribuição de médias

Erro padrão e intervalo de confiança

Teorema do Limite Central

```
import numpy as np
import random
import matplotlib.pyplot as plt
from scipy import stats

amostra_dados = lambda tamanho: [random.randint(1,6)
for i in range(0,tamanho)]

tamanho, n_amostras = 100, 1000

medias = sorted([np.mean(amostra_dados(tamanho))
for i in range(0,n_amostras)])
pdf = list(stats.norm.pdf(medias, loc=np.mean(medias),
scale=np.std(medias)))
```

Erro padrão e intervalo de confiança

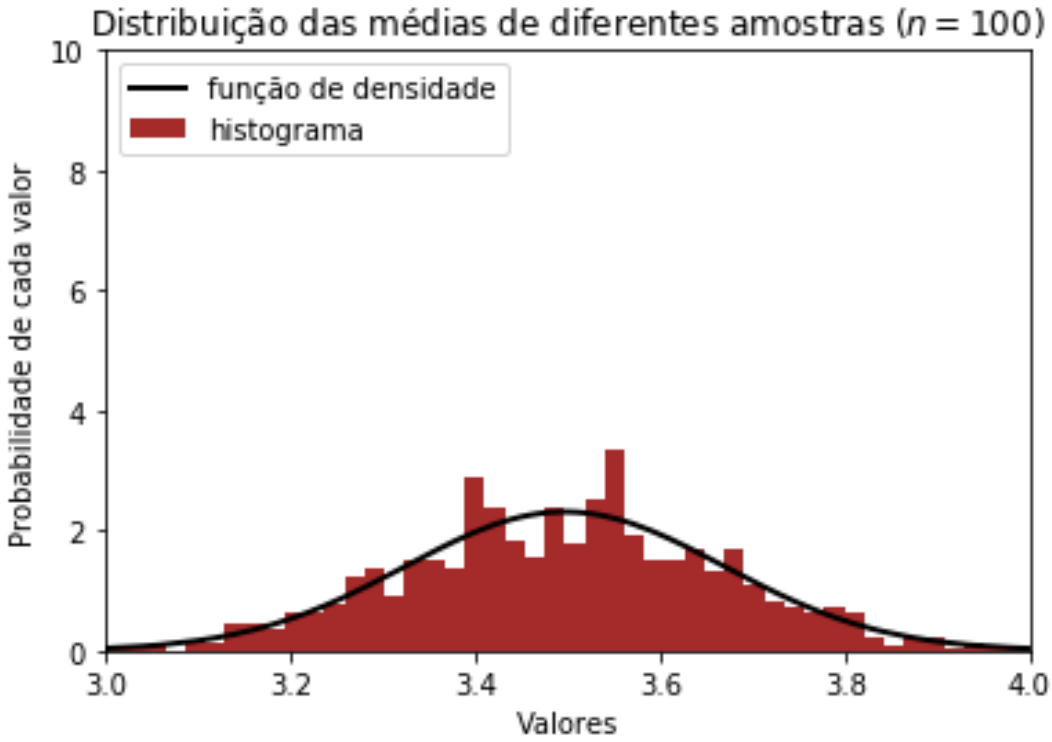
Teorema do Limite Central

```
plt.hist(medias, bins=50, color='brown',
         label='histograma', normed=True)

plt.plot(medias, pdf, color='black',
         linewidth=2, label='função de densidade')

plt.legend(loc="upper left")
plt.xlabel('Valores')
plt.ylabel('Probabilidade de cada valor')
plt.axis([3, 4, 0, 10])
plt.title(r'Distribuição das médias de diferentes amostras
plt.show()
```

Erro padrão e intervalo de confiança

Teorema do Limite Central

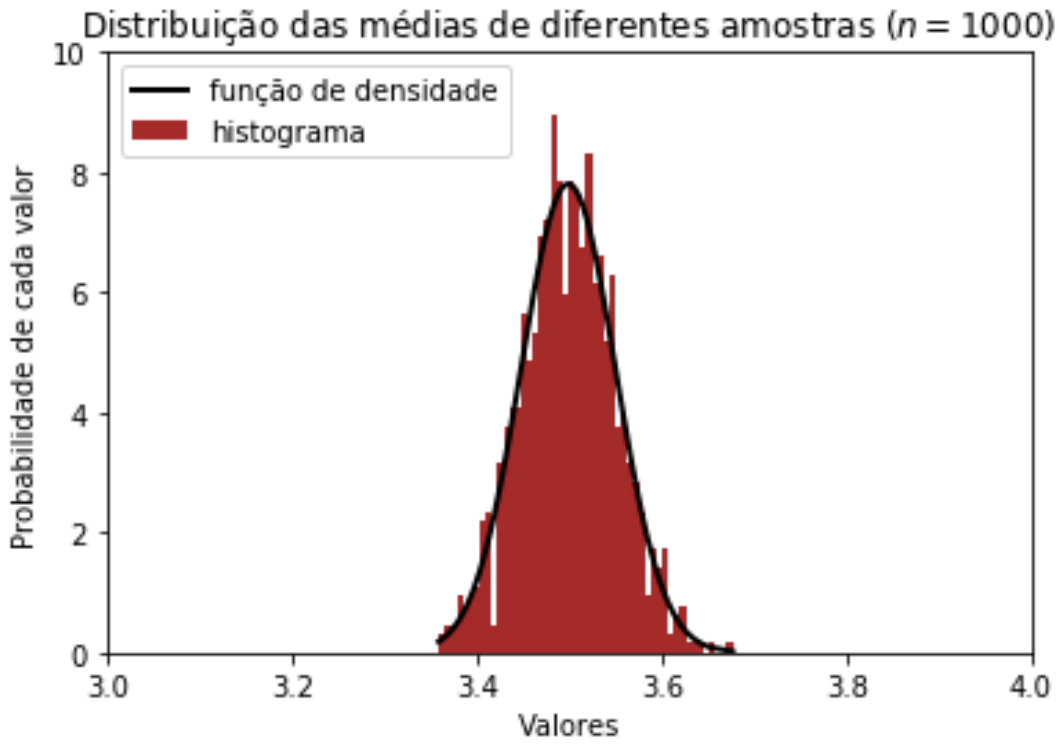
- Note que a média das médias é próxima de $3,5$ e parece uma distribuição normal.



Erro padrão e intervalo de confiança

Teorema do Limite Central

- E se o tamanho de cada amostra fosse maior, com $n = 1000$, o que acontece?
- Note que a variabilidade foi reduzida.



Erro padrão e intervalo de confiança

Teorema do Limite Central

- Teorema do limite central:
  - A média das médias de amostras representativas aproximam-se da média da população:

$$\mu_{\bar{X}} \equiv E(\bar{X}) \equiv \frac{1}{K}\sum_{k=1}^{K}\bar{x}_k = \frac{\bar{x}_1 + \ldots + \bar{x}_K}{K} \approx \mu \tag{1}$$

  - A desvio padrão das médias estimadas de diferentes amostras representativas reduz-se com o número de observações em cada amostra, seguindo um padrão:

$$\sigma_{\bar{X}} \equiv \frac{\sigma}{\sqrt{n}} \tag{2}$$

  - Para um número suficiente grande de observações e amostragens, a distribuição de médias aproxima-se de uma distribuição normal:

$$\bar{X} \sim \mathcal{N}(\mu_{\bar{X}}, \sigma^2_{\bar{X}}) \tag{3}$$

Erro padrão e intervalo de confiança

Média

- Vimos que:

$$\sigma_{\bar{X}} \equiv \frac{\sigma}{\sqrt{n}}$$

- O problema: o parâmetro σ é, na maioria das vezes, desconhecido.
- Por isso, substituímos o parâmetro pela estatística s_X: o desvio padrão amostral.

$$s_\mu = \frac{s_X}{\sqrt{n}} \tag{4}$$

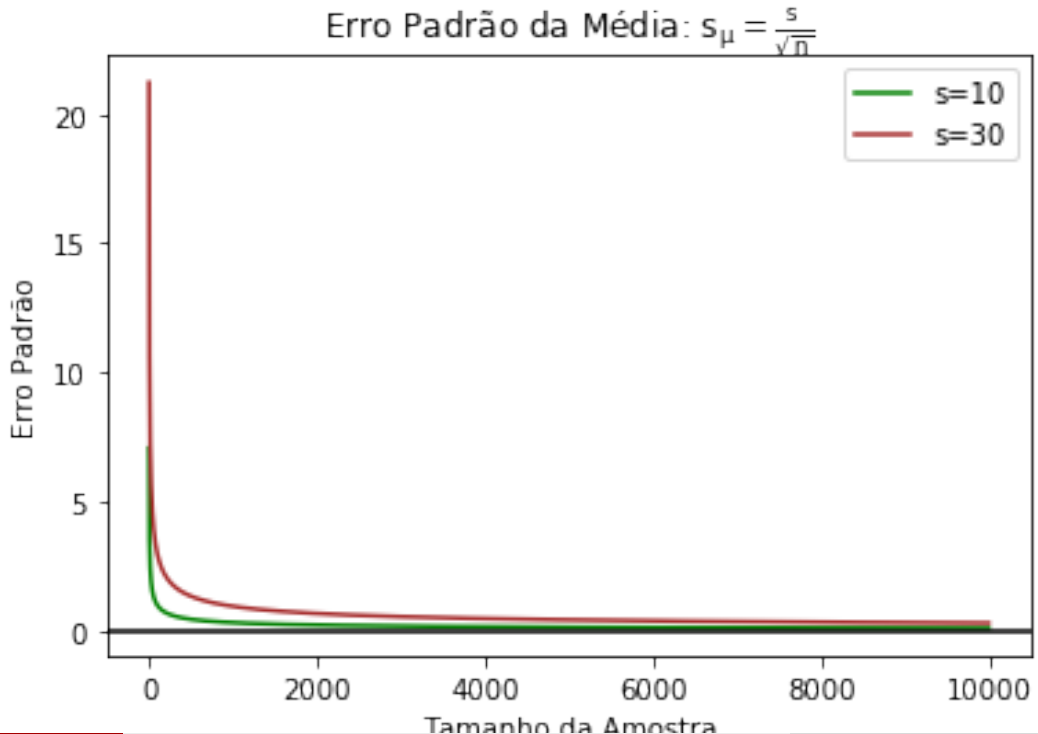
Erro padrão e intervalo de confiança

Média

Intuição

Quanto maior o desvio padrão, maior o erro padrão. Quanto maior a amostra, menor o erro padrão.

- Observando:



Erro padrão e intervalo de confiança

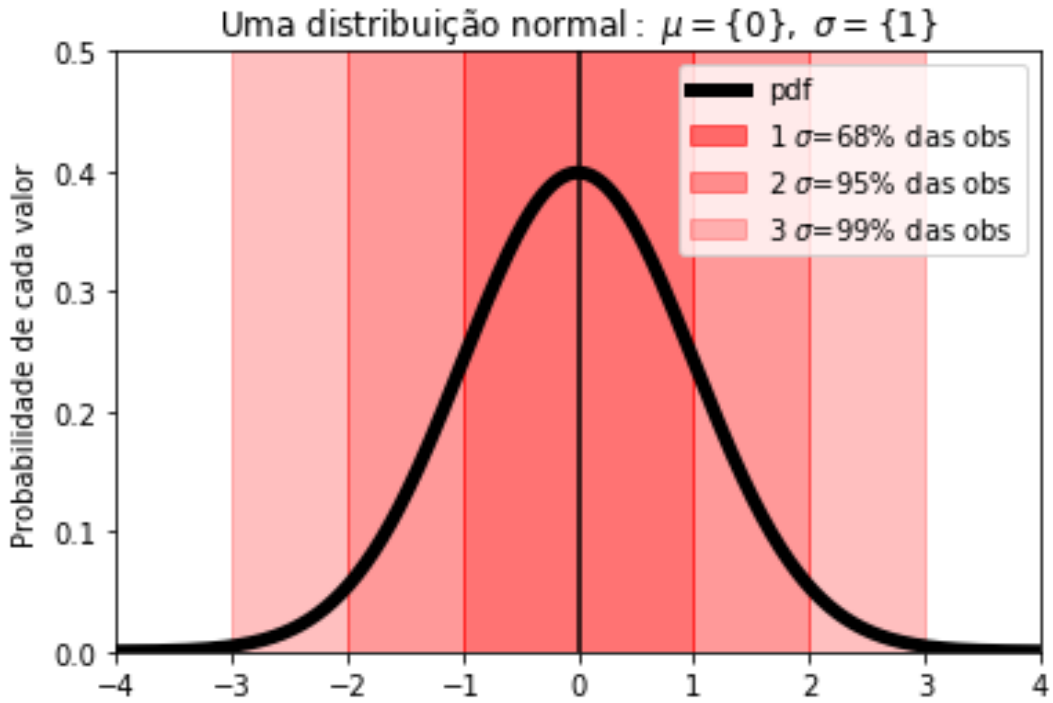
Média

```
sigma = 10
nn = np.linspace(2,10000,num=10000)

se1,se2 = [],[]
for n in nn:
    se1.append(sigma / np.sqrt(n))
    se2.append(3*sigma / np.sqrt(n))

seplot = plt.figure()

plt.plot(nn, se1, color='green', label='s=10')
plt.plot(nn, se2, color='brown', label='s=30')

plt.axhline(y=0, color='black')

plt.legend(loc=1)
plt.xlabel('Tamanho da Amostra')
plt.ylabel('Erro Padrão')
plt.title('Erro Padrão da Média: '
r'$\mathrm{ s_\mu = \frac{s}{\sqrt{n}}}$')
plt.show()
```

Erro padrão e intervalo de confiança

Média

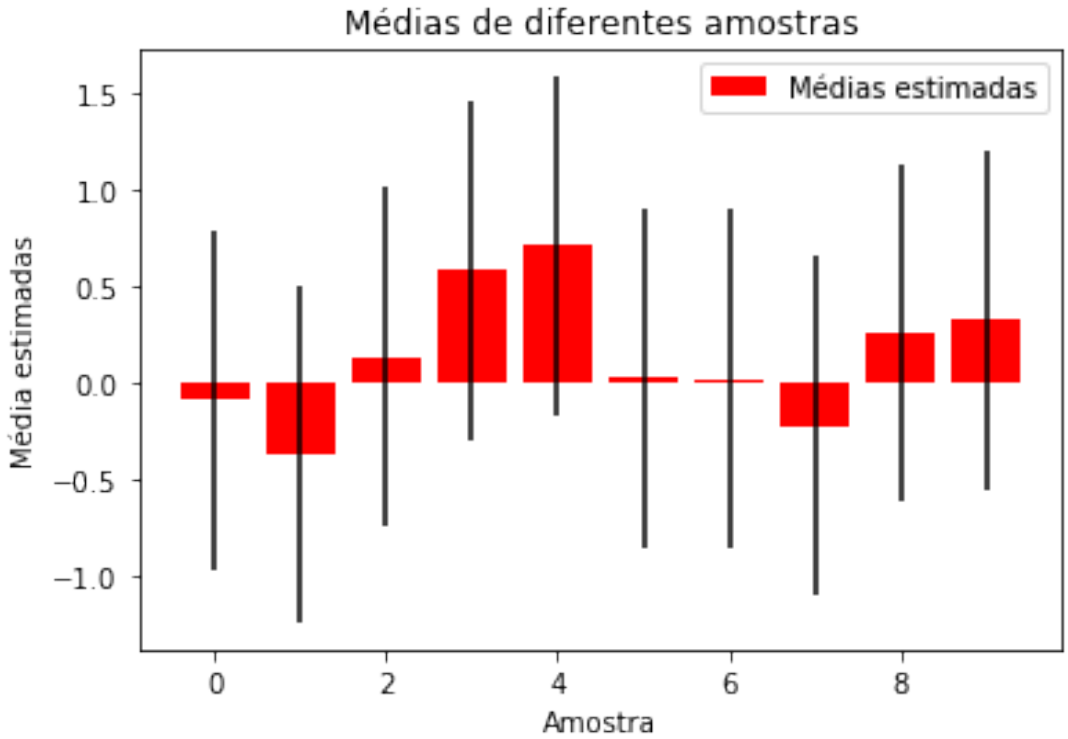
- Como sabemos que a distribuição do erro das médias segue uma distribuição normal, podemos utilizar nosso conhecimento da distribuição normal para criar um intervalo de confiança (ou intervalo de incerteza) ao redor de nossas médias estimadas, com base em nosso erro padrão



Erro padrão e intervalo de confiança

Média

- Se quisermos um intervalo de confiança que inclua $\approx 95\%$ das observações da distribuição de médias, podemos descrever um intervalo de 2 erros padrões ao redor da média!



Erro padrão e intervalo de confiança

Média

```
mu, sigma, = 0, 10
n_amostras, amostra_tam, pop_tam = 10, 500, 100
erros_padrao = 2

pop = np.random.normal(mu, sigma, pop_tam)

amostra = np.matrix([[0 for x in range(n_amostras)]
for y in range(amostra_tam)])

erros, medias = [], []
for i in range(n_amostras):
    s = np.random.choice(pop, size=amostra_tam)
    amostra[:,i] = np.transpose(np.matrix(s))
    media = s.mean()
    medias.append(media)
    erro = erros_padrao * (np.std(s) / np.sqrt(amostra_tam))
    erros.append(erro)
    print("Amostra " + str(i+1) + ", média: {:0.2f}; erro-padrão: {:0.2f}"
    .format(media, erro) )
```

Erro padrão e intervalo de confiança

Média

```
barras = plt.figure()

plt.bar(range(n_amostras), medias, color='red',
    label='Médias estimadas', yerr=erros)

plt.legend(loc=1)
plt.xlabel('Amostra')
plt.ylabel('Média estimadas')
plt.title('Médias de diferentes amostras')

plt.show()
```

Erro padrão e intervalo de confiança

Proporção amostral

- A mesma lógica vale para proporções de populações
- O cálculo do desvio padrão é um pouco diferente, porque com proporções segue-se a lógica de variáveis discretas (normalmente respostas sim/não, cara/coroa, verdadeiro/falso).
- Por exemplo: no primeiro turno, você votaria no candidato X? Sim/Não
- A distribuição das respostas segue um distribuição binomial

Erro padrão e intervalo de confiança

Proporção amostral

- O erro padrão de uma proporção amostral se dá por

$$s_{\hat{p}} = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \tag{5}$$

- Por exemplo, se perguntarmos para uma amostra aleatória de 1000 brasilienses se eles vão votar para o candidato x e 27% disserem que sim, qual é o erro padrão?

$$s_{\hat{p}} = \sqrt{\frac{27\% * 73\%}{1000}} = 1,4\% \tag{6}$$

- Qual o intervalo de ± 2 erros padrão?
- $27\% \pm 2 * 1,4 \approx \{24\%, 30\%\}$

Erro padrão e intervalo de confiança

Diferenças

- O erro padrão é simplesmente o desvio padrão de estimativas diferentes de médias ou proporções
- O desvio padrão da diferença de duas variáveis independentes é:

$$s_{(x_1 - x_2)} = \sqrt{s_{x_1}^2 + s_{x_2}^2} \tag{7}$$

- Portanto, o erro padrão de duas variáveis aleatórias independentes é

$$s_{(\bar{x}_1 - \bar{x}_2)} = \sqrt{s_{\bar{x}_1}^2 + s_{\bar{x}_2}^2} \tag{8}$$

Erro padrão e intervalo de confiança

Diferenças

- De forma similar, o erro padrão de duas proporções diferentes é:

$$s_{(\hat{p}_1-\hat{p}_2)} = \sqrt{s^2_{\hat{p}_1} + s^2_{\hat{p}_2}} = \sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n_1} + \frac{\hat{p}_2(1-\hat{p}_2)}{n_2}} \tag{9}$$

- Por exemplo, digamos que foi estimada a proporção de pessoas que ingerem álcool numa amostra de católicos ($\hat{p}_1 = 52\%, n_1 = 139$) e evangélicos ($\hat{p}_2 = 29\%, n_2 = 378$). Qual o erro padrão da diferença entre essas estimativas independentes (isto é, amostras independentes populações diferentes)?

$$s_{(\hat{p}_1-\hat{p}_2)} = \sqrt{\frac{0,52 * 0,48}{139} + \frac{0,29 * 0,71}{378}} = 4,83\% \tag{10}$$

Erro padrão e intervalo de confiança

Diferenças

- Qual é a diferença entre essas estimativas com dois desvios padrões como intervalo de confiança?

$$\begin{aligned}(\hat{p}_1 - \hat{p}_2) \pm 2 * s_{(\hat{p}_1 - \hat{p}_2)} &= (0,52 - 0,26) \pm 2 * 0,483 \\ &= 23\% \pm 2 * 9,66\% \approx \{13\%, 33\%\}\end{aligned}$$

- Podemos dizer com confiança que a diferença existe (é diferente de zero). Por que?

$$\frac{0,23 - 0}{4,83\%} \approx 4,75 e.p. \tag{11}$$

- O valor zero de $(\hat{p}_1 - \hat{p}_2)$ está afastado de nossa estimativa por aproximadamente 4,75 erros padrão

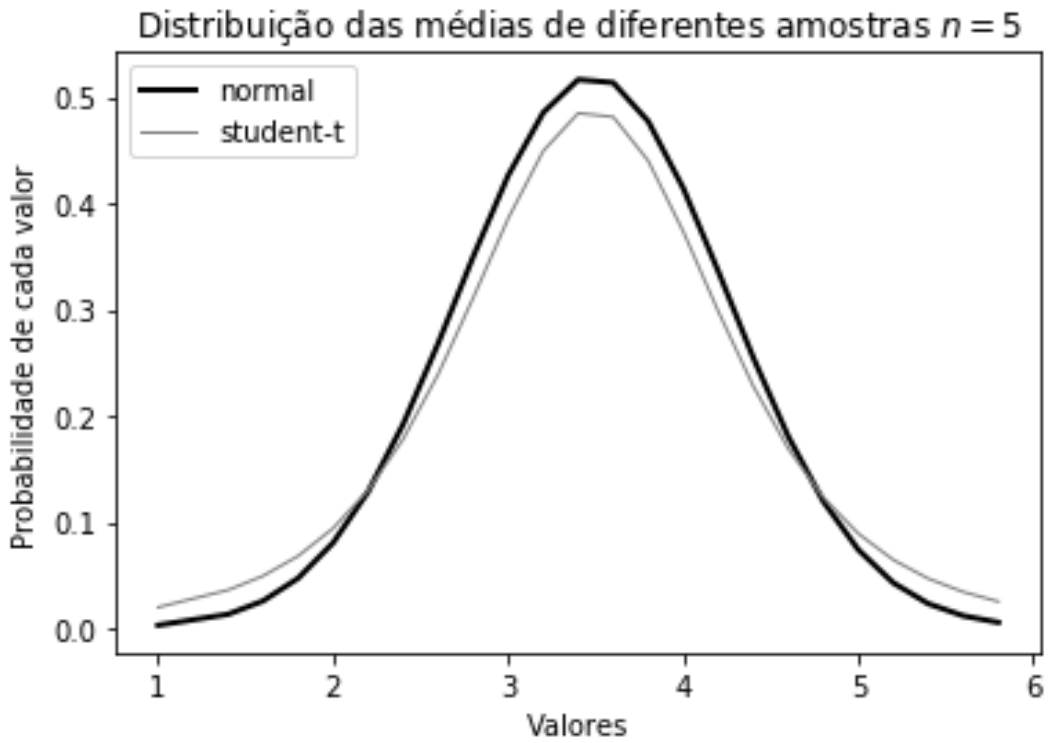
Erro padrão e intervalo de confiança

Distribuição-t

- A distribuição de erros padrão se aproxima de uma distribuição normal. Mas isso é verdade quando o número de observações em cada amostra é grande o suficiente.
- Ou seja, quando $n \to \infty$, $t - student \to \mathcal{N}$

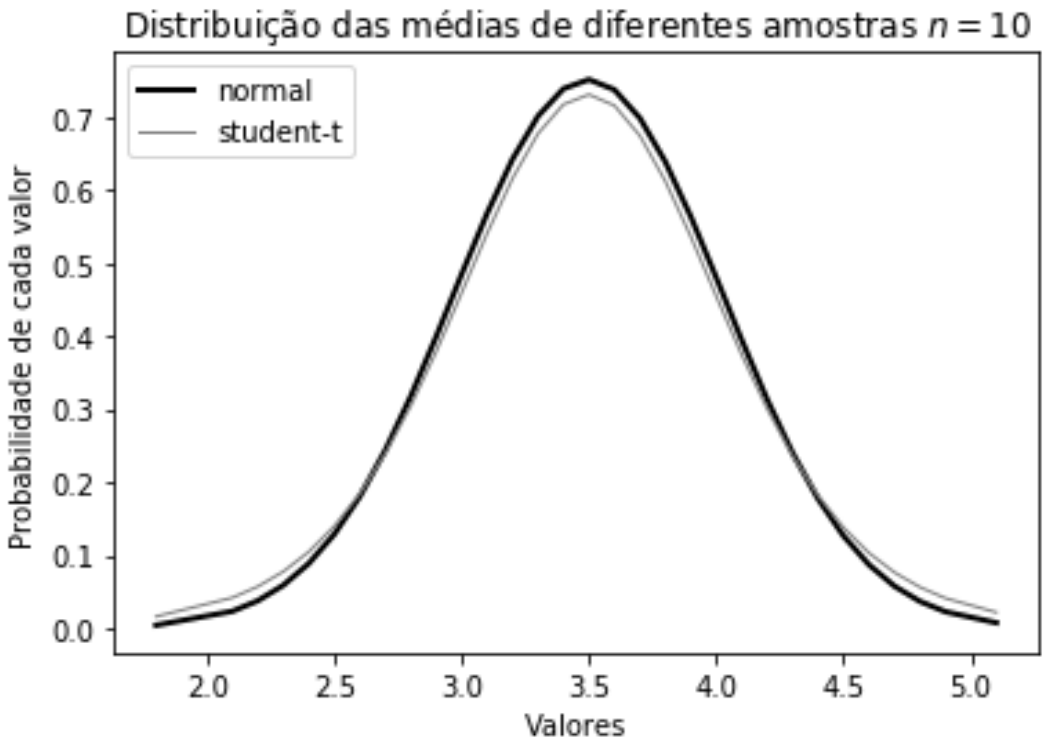
Erro padrão e intervalo de confiança

Distribuição-t



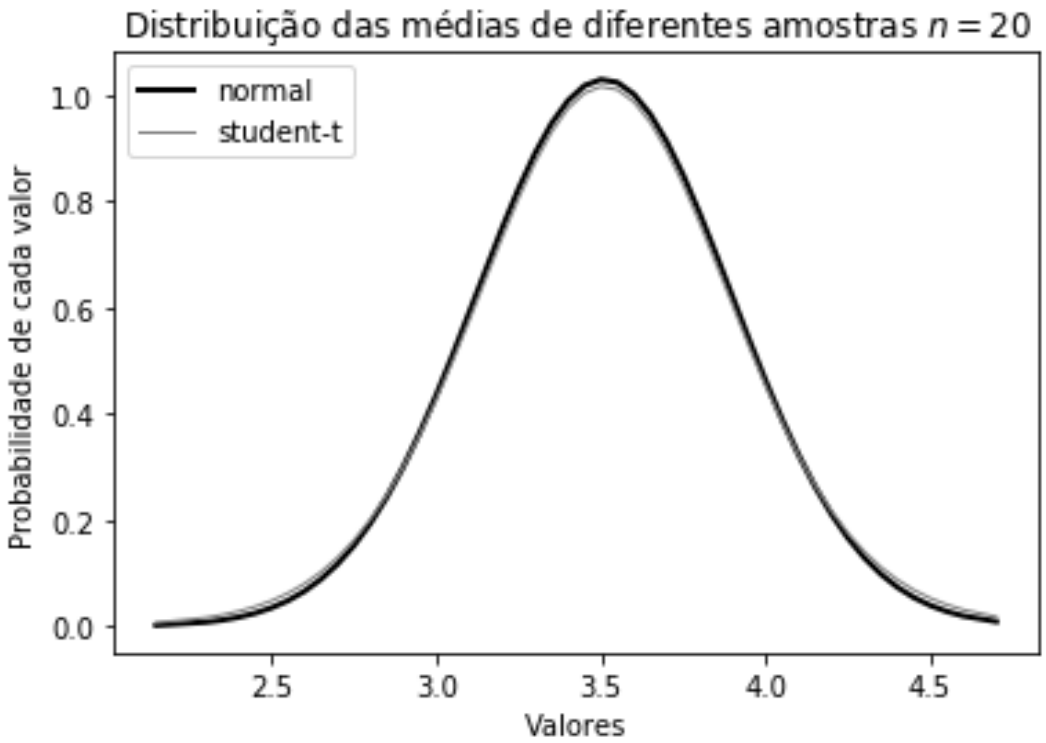
Erro padrão e intervalo de confiança

Distribuição-t



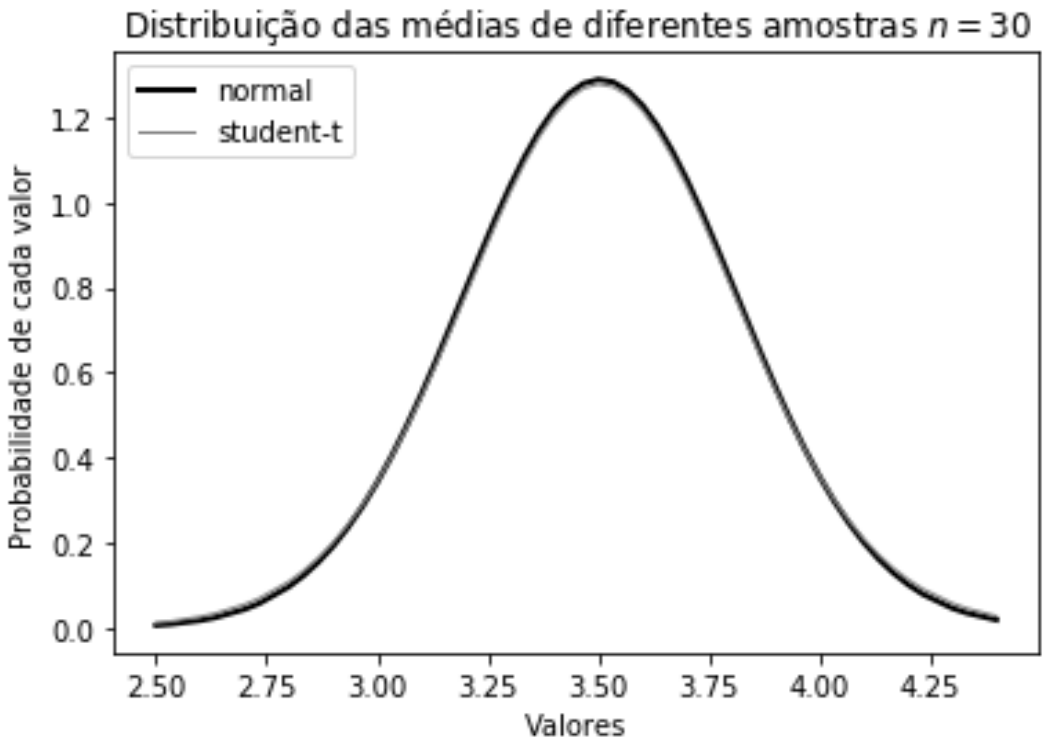
Erro padrão e intervalo de confiança

Distribuição-t



Erro padrão e intervalo de confiança

Distribuição-t



Erro padrão e intervalo de confiança

Distribuição-t

- Se a amostra for grande o suficiente, usar a distribuição normal ou a distribuição student-t para calcular intervalos de confiança chega no mesmo resultado
- No Python, você pode usar outra família da classe scipy.stats para traçar distribuições para a distribuição student-t:

```
x1 = 30
m1 = sorted([np.mean(amostra_dados(x1)) for i in range(0,1000)])
pdf1 = list(stats.t.pdf(m1, df=(x1-1), loc=np.mean(m1), scale=np.std(m1)))

scipy.stats.t.ppf(0.975, df=(x1-1))
scipy.stats.norm.ppf(0.975)
```

Sumário

Introdução ao teste de hipótese

Intuição

- Em estatística, muitas vezes queremos testar uma hipótese
- Podemos, por exemplo, comparar a performance média de dois grupos e, sabando da variabilidade dos estimadores, entender o quão confiável é a diferença entre eles

Introdução ao teste de hipótese

Intuição

- Em estatística, muitas vezes queremos testar uma hipótese
- Podemos, por exemplo, comparar a performance média de dois grupos e, sabendo da variabilidade dos estimadores, entender o quão confiável é a diferença entre eles
- Os passos essenciais, portanto, são (1) definir uma hipótese a ser testada; (2) definir qual o nosso grau de incerteza tolerável para aceitar tal hipótese

Introdução ao teste de hipótese

Intuição

- A notação que utilizamos é:

$$H_0 \quad : \quad \textit{hipótese nula}$$
$$H_a \quad : \quad \textit{hipótese alternativa}$$

- H_0 é a hipótese que vamos testar. Se a rejeitamos, aceitamos a hipótese alternativa.

Introdução ao teste de hipótese

estatística-t

- Para saber o quão distante o valor de corte está, em erros padrão, da nossa estimativa, podemos calcular a "estatística t":

$$t_0 = \frac{\hat{\theta} - \theta_0}{s_{\hat{\theta}}} = \frac{estimativa \quad - \quad valor \quad de \quad corte}{erro \quad padrão} \tag{12}$$

Introdução ao teste de hipótese

estatística-t

- Uma hipótese pode ser: "O peso médio de homens (p_h) é maior do que o peso médio de mulheres (p_m)"

$$\begin{aligned} H_0 &: p_h - p_m > 0 \\ H_A &: p_h - p_m \leq 0 \end{aligned}$$

- Como calcular isso?

Introdução ao teste de hipótese

estatística-t

- Primeiro, vamos importar um arquivo do excel:

```
import pandas as pd

dfiq = pd.read_excel('https://github.com/omercadopopular/
cgoes/blob/master/StatsPython/
data/brain_size.xlsx?raw=true')

print(dfiq)
```

Quais são os problemas com essa importação?

- Valores faltando
- Nomes e unidades em inglês

Introdução ao teste de hipótese

p-valor

- Primeiro, vamos importar um arquivo do excel, dizendo para o pandas que os valores faltantes estão sinalizados com um ponto:

```
import pandas as pd

dfiq = pd.read_excel('https://github.com/omercadopopular/
cgoes/blob/master/StatsPython/
data/brain_size.xlsx?raw=true', na_values=".")
```

- Depois, alteramos os nomes das colunas:

```
dfiq.columns = ['sexo', 'FSIQ', 'VIQ', 'PIQ', 'peso', 'altura', 'MRI_Count']
```

- Transformamos as escalas

```
lb_para_kg = lambda x: x / 2.2
in_para_cm = lambda x: x * 2.54

dfiq['peso'] = [lb_para_kg(pes) for pes in dfiq['peso']]
dfiq['altura'] = [in_para_cm(alt) for alt in dfiq['altura']]
```

Introdução ao teste de hipótese

estatística-t

- E alteramos os rótulos dos grupos:

```
dfiq['sexo'] = [string.replace("Female", "Feminino").replace("Male", "Masculin
```

- A partir daí, usamos groupby para criar sumarizações com base nos cortes por "sexo":

```
grupos = dfiq.groupby('sexo')

print(grupos.mean(), grupos.median())

print(grupos.describe().T)

grupos.boxplot(column=['peso'])
```

Introdução ao teste de hipótese

estatística-t

- Vamos calcular a diferença de média, o erro padrão da diferença e a estatística t

```
diff = grupos['peso'].mean()['Masculino'] - grupos['peso'].mean()['Feminino']

erro_padrao = ( (grupos['peso'].std()['Masculino'] /
                 (grupos['peso'].count()['Masculino']) ** (1/2)) +
                 (grupos['peso'].std()['Feminino'] /
                 (grupos['peso'].count()['Feminino']) ** (1/2)) )

t_stat = (diff - 0) / erro_padrao

print(t_stat)
```