**Métodos Quantitativos I**

**Camila Guedes de Farias**

3ª Lista de Exercícios, 15/4/2025

*Descrição:*

*Exercícios dos capítulos 7 e 8 do livro Estatística Aplicada a Administração e Economia.*

*Cap. 7:  15, 18, 19, 22, 23, 28, 31*

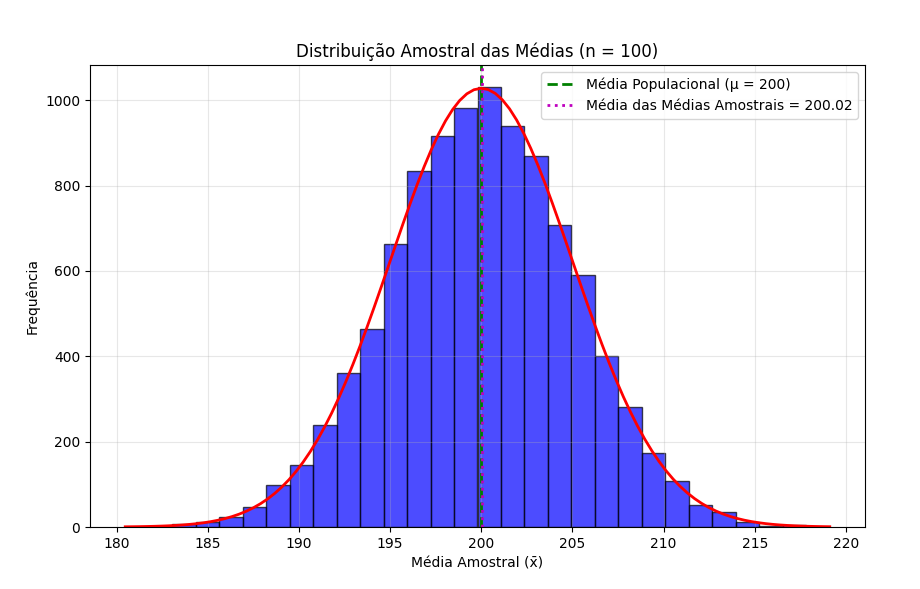
*Cap. 8: 2, 7, 11, 13, 19, 22, 23*

***Cap. 7:15***

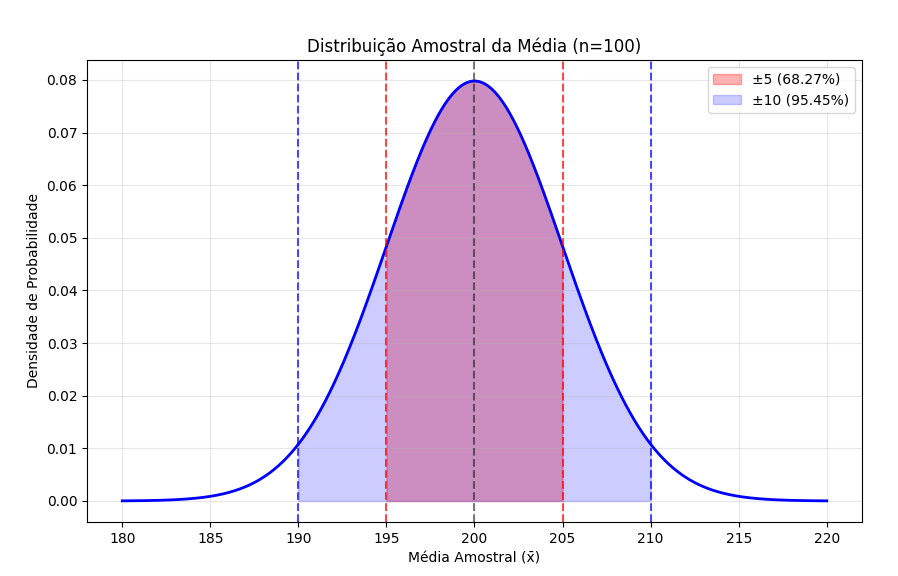
*classificações = [57, 61, 86, 74, 72, 73, 20, 57, 80, 79, 83, 74]*

1. *Estimativa pontual da classificação média: 68.00*
2. *Estimativa pontual do desvio padrão: 17.89*

***Cap. 7:18***

1. *Valor esperado de x̄ = 200*
2. *Desvio padrão de x̄ = 5.0*
3. 
4. A distribuição amostral de x̄ mostra como se comportam as médias de amostras

de tamanho 100 retiradas da população. Pelo Teorema do Limite Central, essa distribuição é aproximadamente normal, independentemente da distribuição da população original, principalmente porque o tamanho da amostra é grande (n = 100). Esta distribuição é centrada na média populacional (μ = 200) e tem desvio padrão reduzido (σ/√n = 5.0) A distribuição amostral permite calcular probabilidades relacionadas à média amostral.

***Cap. 7:19***

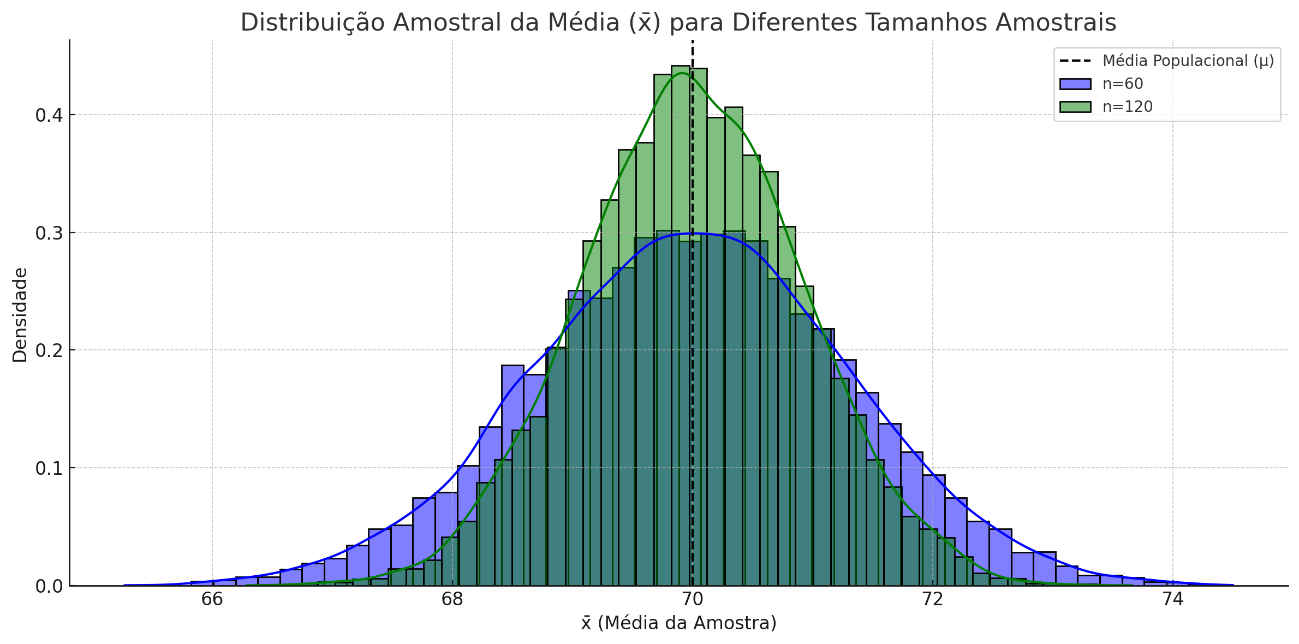
1. 0.6827 = 68.27%
2. Para ±5: z = ±1.0000

P(-1.0000 < Z < 1.0000) = 0.6827

Para ±10: z = ±2.0000

P(-2.0000 < Z < 2.0000) = 0.9545

***Cap.7:22***



1. A curva azul no gráfico representa a distribuição amostral da média (x̄) com amostras de tamanho 60. Ela segue uma distribuição aproximadamente normal, centrada na média populacional (μ = 70), com um desvio padrão reduzido (erro padrão), o que mostra que a média das amostras tende a se concentrar em torno da média da população.

- Média teórica: 100

- Erro padrão teórico: 1.9365

- Média empírica obtida: 100.0164

- Desvio padrão empírico: 1.9234

|  |  |  |  |
| --- | --- | --- | --- |
| Tamanho da Amostra (n) | Erro Padrão Teórico (σx̄) | Média das Médias Amostrais | Desvio Padrão Empírico das Médias |
| 30 | 2.738613 | 99.978095 | 2.759302 |
| 60 | 1.936492 | 100.016429 | 1.923431 |
| 120 | 1.369306 | 99.992369 | 1.382023 |
| 240 | 0.968246 | 100.014509 | 0.963809 |
| 480 | 0.684653 | 100.001016 | 0.680369 |

1. - Média teórica: 100

- Erro padrão teórico: 1.3693

- Média empírica obtida: 99.9924

- Desvio padrão empírico: 1.3820

1. À medida que o tamanho da amostra aumenta, observamos que:
2. A distribuição amostral da média se aproxima cada vez mais de uma distribuição normal, independentemente da forma da distribuição da população original (Teorema do Limite Central).
3. A variabilidade da distribuição amostral diminui, ou seja, o erro padrão (σx̄) diminui proporcionalmente à raiz quadrada do tamanho da amostra (σx̄ = σ/√n).
4. A média da distribuição amostral permanece igual à média populacional (μ), independentemente do tamanho da amostra.

Esta generalização é lógica porque: Quanto maior a amostra, mais informação temos sobre a população, o que reduz a incerteza na estimativa da média populacional. A redução do erro padrão é proporcional à raiz quadrada do tamanho da amostra (não linear), o que significa que o ganho em precisão diminui à medida que o tamanho da amostra aumenta. Isso explica porque, em pesquisas estatísticas, há um ponto onde aumentar o tamanho da amostra não traz ganhos significativos em precisão que justifiquem o custo adicional da coleta de dados.

***Cap.7:23***

1. Para n=60:

Erro padrão: 520.0468

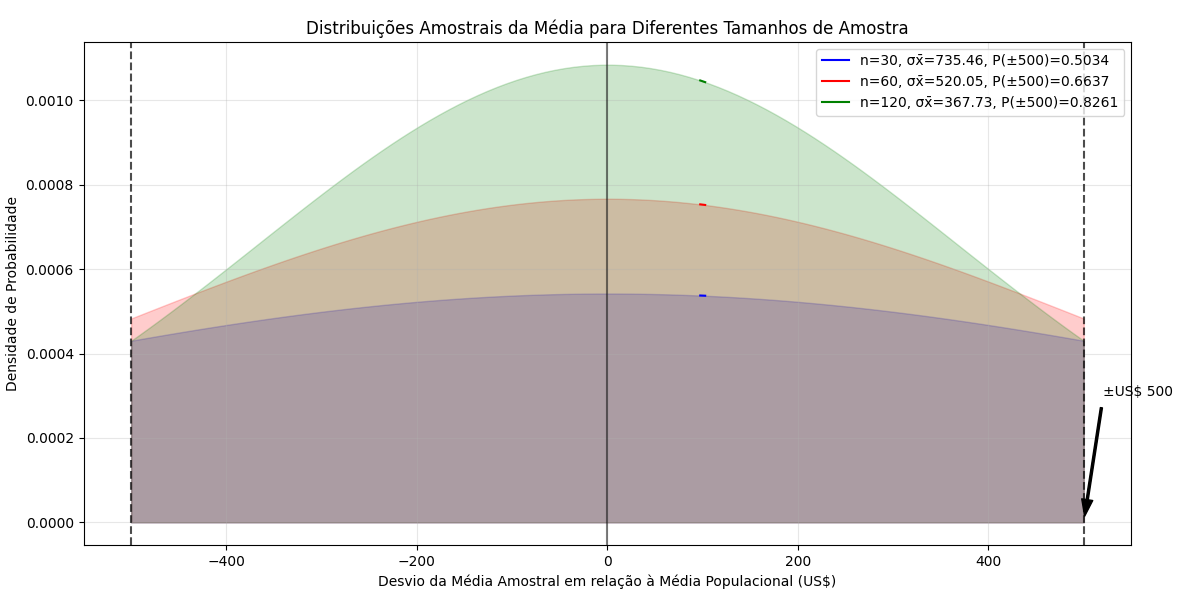
Valor de z: 0.9615

Probabilidade da média amostral estar dentro de ±US$ 500 da média populacional: 0.6637

1. Para n=120:

Valor de z: 1.3597

Probabilidade da média amostral estar dentro de ±US$ 500 da média populacional: 0.8261



***Cap.7:28***

1. Distribuição da precipitação média anual para a Califórnia:

- Média da distribuição amostral: 22 polegadas

- Erro padrão (desvio padrão da distribuição amostral): 0.7303 polegadas

- A distribuição segue uma Normal com N(22, 0.7303²)

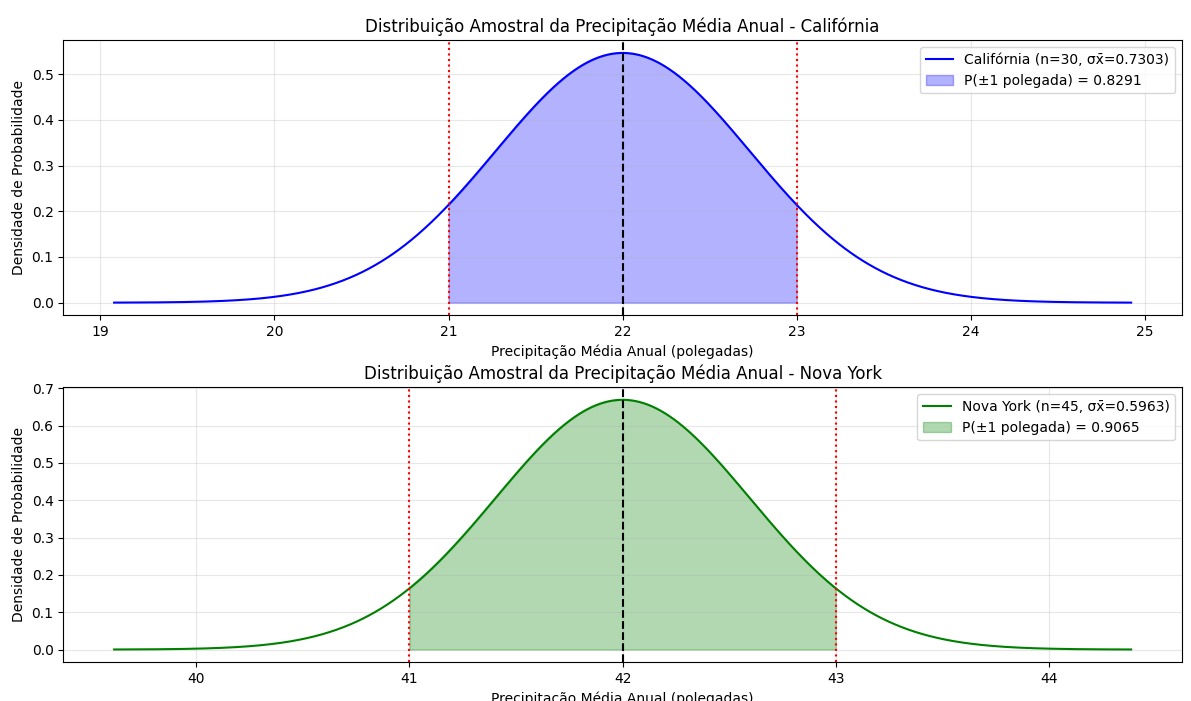
1. Probabilidade da média amostral estar dentro de 1 polegada da média populacional para a Califórnia: P(21 < x̄ < 23) = 0.8291 = 82.91%
2. Probabilidade da média amostral estar dentro de 1 polegada da média populacional para Nova York: P(41 < x̄ < 43) = 0.9065 = 90.65%
3. Comparação das probabilidades:

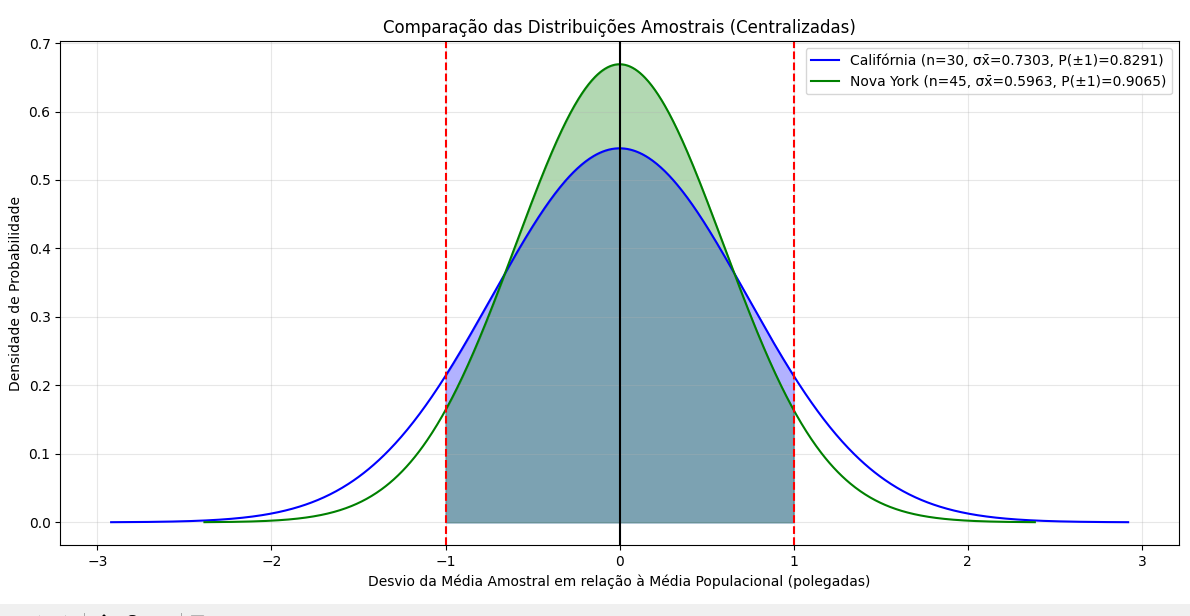
*A probabilidade é maior para Nova York apesar do maior tamanho de amostra (45 vs 30****)***

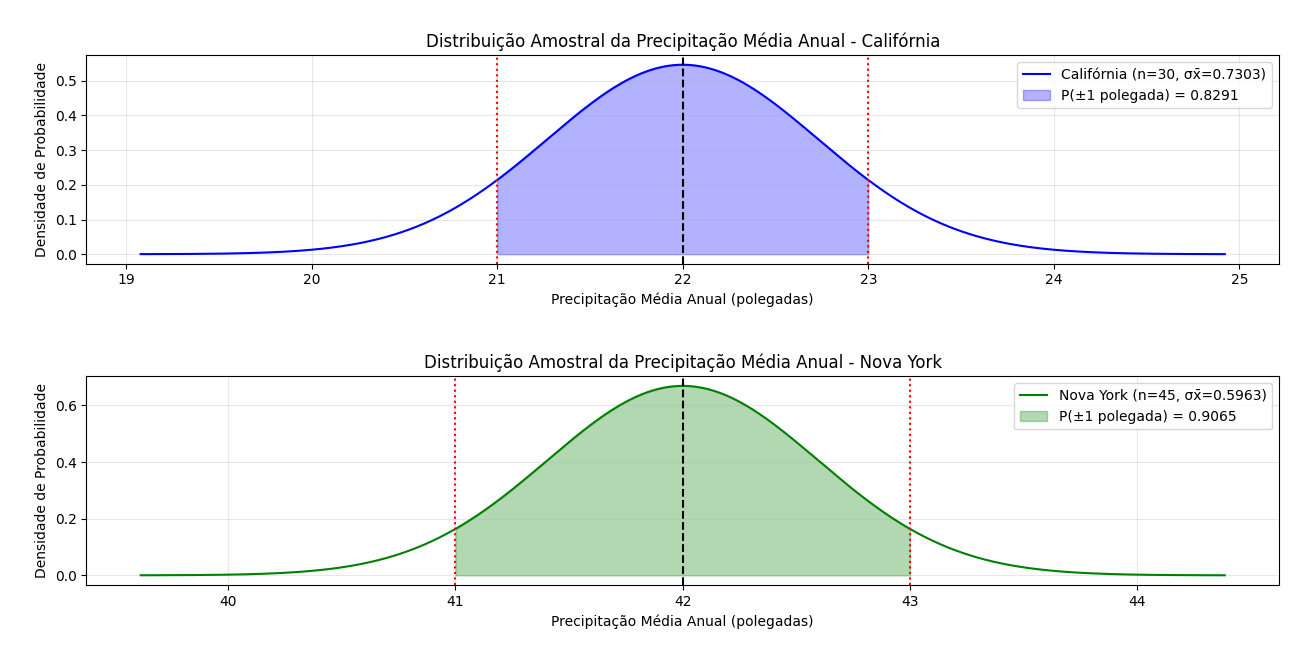
- Califórnia: n=30, σx̄=0.7303, z=1.3693, P(±1)=0.8291

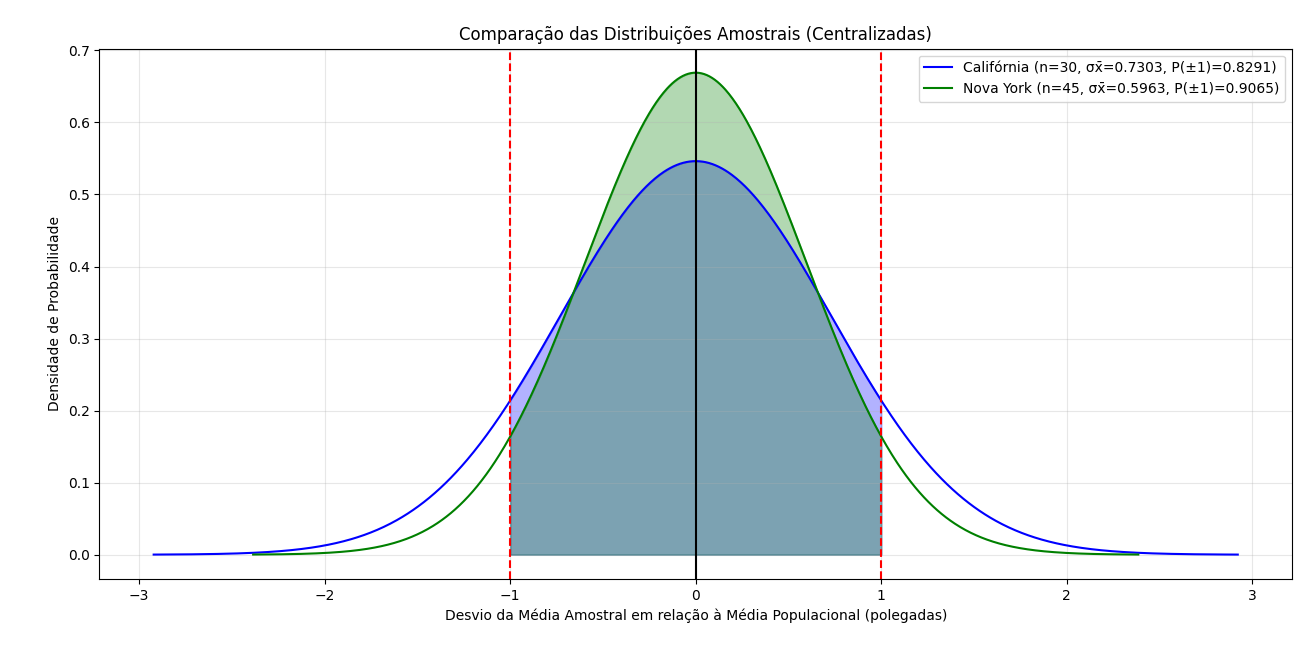
- Nova York: n=45, σx̄=0.5963, z=1.6771, P(±1)=0.9065

Como o desvio padrão populacional é o mesmo (4 polegadas) para ambos os estados, o erro padrão depende apenas do tamanho da amostra (σx̄ = σ/√n). Nova York tem uma amostra maior (n=45 vs n=30), o que resulta em um menor erro padrão. e, consequentemente, uma distribuição amostral mais concentrada em torno da média populacional. Isso aumenta a probabilidade de a média amostral estar dentro de ±1 polegada da média populacional.



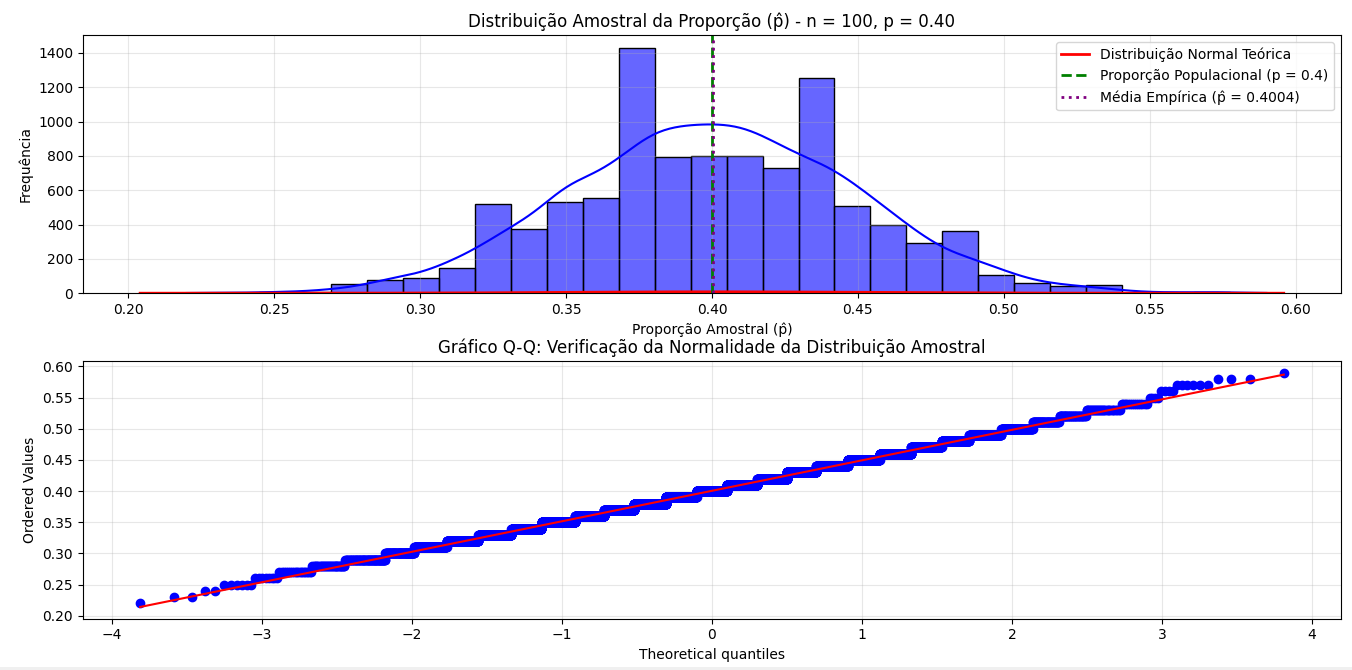






***Cap.7:31***

1. Valor esperado de p̂: 0.4
2. Erro padrão de p̂: 0.0490



1. A distribuição amostral de p̂ segue aproximadamente uma distribuição normal com média igual à proporção populacional (p = 0.4) e erro padrão σp̂ = √(p(1-p)/n) = 0.0490. Esta distribuição mostra como a proporção amostral (p̂) varia quando diferentes amostras de tamanho 100 são retiradas da população. A distribuição nos permite calcular probabilidades associadas a possíveis valores de p̂ e construir intervalos de confiança para p. A maioria das proporções amostrais estará em torno da proporção populacional, com aproximadamente 95% dos valores dentro de ±2 erros padrão: 0.3020 a 0.4980. À medida que o tamanho da amostra aumenta, a distribuição se torna mais concentrada em torno da proporção populacional, reduzindo a variabilidade das estimativas.

Média teórica de p̂: 0.4

Média empírica de p̂ (baseada na simulação): 0.4004

Erro padrão teórico: 0.0490

Desvio padrão empírico: 0.0489

Probabilidade de p̂ > 0.45: 0.1537 = 15.37%

Probabilidade de p̂ estar dentro de ±1 erro padrão de p: 0.6827 = 68.27%

***Cap.8:2***

1. IC 90%: (30.6043, 33.3957)
2. IC 95%: (30.3369, 33.6631)
3. IC 99%: (29.8143, 34.1857)

***Cap.8:7***

1. Margem de erro para IC de 95%: US$ 70.68
2. Média amostral calculada: US$ 1785.50

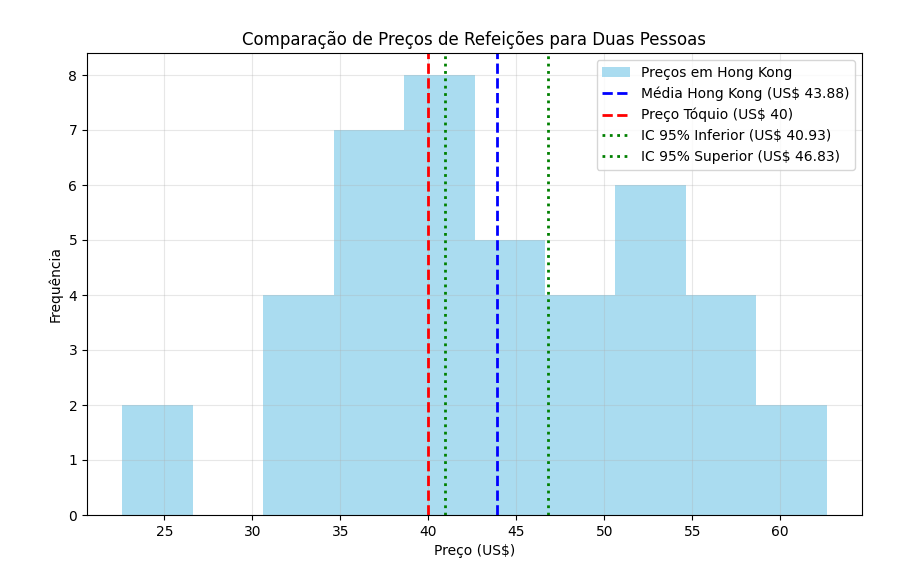
***Cap.8:11***

1. Probabilidade à direita de 2,120: 0.024995
2. Probabilidade à esquerda de 1,337: 0.900039
3. Probabilidade à esquerda de -1,746: 0.049990
4. Probabilidade à direita de 2,583: 0.010010
5. Probabilidade entre -2,120 e 2,120: 0.950009

***Cap.8:13***

1. Estimativa pontual da média populacional: 10.0000
2. Estimativa pontual do desvio padrão populacional: 3.4641
3. Margem de erro (95% de confiança): 2.8961
4. Intervalo de confiança de 95% para a média populacional: [7.1039, 12.8961]

***Cap.8:19***

1. Margem de erro (95% de confiança): 2.95
2. Intervalo de confiança de 95% para a média em Hong Kong: [US$ 40.93, US$ 46.83]
3. Comparação entre Hong Kong e Tóquio: As refeições em Hong Kong são significativamente mais caras que em Tóquio (cerca de 9.7% mais caras) 

***Cap.8:22***

1. Intervalo de confiança de 95% para a receita média por sala: Receita média estimada por sala: US$ 23319.44, Margem de erro: US$ 2049.21
2. Número médio de clientes por sala de cinema: 2875 clientes por sala. Intervalo de confiança de 95%: [2623, 3128] clientes
3. Número total estimado de clientes que assistiram ao filme: 11731604 clientes no total. Total de vendas de ingressos em 4080 salas no fim de semana prolongado

***Cap.8:23***

Margem de erro desejada: 10

Desvio padrão populacional: 40

Nível de confiança: 95.0%

Valor z crítico para IC 95.0%: 1.9600

Tamanho amostral necessário (fórmula): 61.4633

Tamanho amostral necessário (arredondado para cima): 62

Verificação - Margem de erro com n = 62: 9.9566

***Apêndice – Código em Python***

*# lista 3*

*#Exercícios dos capítulos 7 e 8 do livro Estatística Aplicada a Administração e Economia.*

*#Cap. 7:  15, 18, 19, 22, 23, 28, 31*

*#Cap. 8: 2, 7, 11, 13, 19, 22, 23*

*import numpy as np*

*#Cap.7:15*

*# Dados da amostra*

*classificacoes = [57, 61, 86, 74, 72, 73, 20, 57, 80, 79, 83, 74]*

*# a. Estimativa pontual da média (média amostral)*

*media\_amostral = np.mean(classificacoes)*

*# b. Estimativa pontual do desvio padrão (desvio padrão amostral)*

*# Usando n-1 no denominador (estimativa não enviesada para a população)*

*desvio\_padrao\_amostral = np.std(classificacoes, ddof=1)*

*print(f"a. Estimativa pontual da classificação média: {media\_amostral:.2f}")*

*print(f"b. Estimativa pontual do desvio padrão: {desvio\_padrao\_amostral:.2f}")*

*#Cap:7:18*

*import numpy as np*

*import matplotlib.pyplot as plt*

*from scipy import stats*

*# Parâmetros da população*

*media\_populacional = 200*

*desvio\_padrao\_populacional = 50*

*tamanho\_amostra = 100*

*# a. Cálculo do valor esperado de x̄*

*valor\_esperado\_x\_barra = media\_populacional*

*print(f"a. Valor esperado de x̄ = {valor\_esperado\_x\_barra}")*

*# b. Cálculo do desvio padrão de x̄ (erro padrão)*

*desvio\_padrao\_x\_barra = desvio\_padrao\_populacional / np.sqrt(tamanho\_amostra)*

*print(f"b. Desvio padrão de x̄ = {desvio\_padrao\_x\_barra}")*

*# c. Simulação da distribuição amostral de x̄*

*num\_simulacoes = 10000*

*medias\_amostrais = []*

*for \_ in range(num\_simulacoes):*

*# Gerando uma amostra de tamanho 100 da população*

*amostra = np.random.normal(media\_populacional, desvio\_padrao\_populacional, tamanho\_amostra)*

*# Calculando a média da amostra*

*media\_amostral = np.mean(amostra)*

*medias\_amostrais.append(media\_amostral)*

*# Criação do histograma das médias amostrais*

*plt.figure(figsize=(10, 6))*

*plt.hist(medias\_amostrais, bins=30, alpha=0.7, color='blue', edgecolor='black')*

*# Adicionando a curva normal teórica*

*x = np.linspace(min(medias\_amostrais), max(medias\_amostrais), 100)*

*y = stats.norm.pdf(x, media\_populacional, desvio\_padrao\_x\_barra) \* num\_simulacoes \* (max(medias\_amostrais) - min(medias\_amostrais)) / 30*

*plt.plot(x, y, 'r-', linewidth=2)*

*plt.axvline(media\_populacional, color='g', linestyle='--', linewidth=2, label=f'Média Populacional (μ = {media\_populacional})')*

*plt.axvline(np.mean(medias\_amostrais), color='m', linestyle=':', linewidth=2, label=f'Média das Médias Amostrais = {np.mean(medias\_amostrais):.2f}')*

*plt.title('Distribuição Amostral das Médias (n = 100)')*

*plt.xlabel('Média Amostral (x̄)')*

*plt.ylabel('Frequência')*

*plt.legend()*

*plt.grid(True, alpha=0.3)*

*plt.show()*

*# d. Análise da distribuição amostral*

*print("\nc. Distribuição amostral de x̄:")*

*print("   A distribuição amostral de x̄ segue uma distribuição normal com:")*

*print(f"   - Média = {media\_populacional}")*

*print(f"   - Desvio padrão = {desvio\_padrao\_x\_barra}")*

*print("\nd. O que mostra a distribuição amostral de x̄:")*

*print("   A distribuição amostral de x̄ mostra como se comportam as médias de amostras")*

*print("   de tamanho 100 retiradas da população. Pelo Teorema do Limite Central,")*

*print("   essa distribuição é aproximadamente normal, independentemente da distribuição")*

*print("   da população original, principalmente porque o tamanho da amostra é grande (n = 100).")*

*print("   Esta distribuição é centrada na média populacional (μ = 200) e tem desvio")*

*print(f"   padrão reduzido (σ/√n = {desvio\_padrao\_x\_barra}).")*

*print("   A distribuição amostral permite calcular probabilidades relacionadas à média amostral.")*

*plt.show()*

*#Cap. 7:19*

*import numpy as np*

*from scipy import stats*

*import matplotlib.pyplot as plt*

*# Parâmetros do problema*

*media\_populacional = 200*

*desvio\_padrao\_populacional = 50*

*tamanho\_amostra = 100*

*# Cálculo do erro padrão da média (desvio padrão da distribuição amostral)*

*erro\_padrao = desvio\_padrao\_populacional / np.sqrt(tamanho\_amostra)*

*print(f"Erro padrão da média: {erro\_padrao}")*

*# a. Probabilidade de que a média amostral esteja dentro de ±5 da média populacional*

*# Padronizando os valores para a distribuição normal padrão*

*z\_5\_superior = (media\_populacional + 5 - media\_populacional) / erro\_padrao*

*z\_5\_inferior = (media\_populacional - 5 - media\_populacional) / erro\_padrao*

*# Calculando a probabilidade usando a função de distribuição acumulada (CDF)*

*prob\_dentro\_5 = stats.norm.cdf(z\_5\_superior) - stats.norm.cdf(z\_5\_inferior)*

*print(f"\na. Probabilidade de que a média amostral esteja dentro de ±5 da média populacional:")*

*print(f"   P({media\_populacional-5} < x̄ < {media\_populacional+5}) = {prob\_dentro\_5:.4f} = {prob\_dentro\_5\*100:.2f}%")*

*# b. Probabilidade de que a média amostral esteja dentro de ±10 da média populacional*

*z\_10\_superior = (media\_populacional + 10 - media\_populacional) / erro\_padrao*

*z\_10\_inferior = (media\_populacional - 10 - media\_populacional) / erro\_padrao*

*prob\_dentro\_10 = stats.norm.cdf(z\_10\_superior) - stats.norm.cdf(z\_10\_inferior)*

*print(f"\nb. Probabilidade de que a média amostral esteja dentro de ±10 da média populacional:")*

*print(f"   P({media\_populacional-10} < x̄ < {media\_populacional+10}) = {prob\_dentro\_10:.4f} = {prob\_dentro\_10\*100:.2f}%")*

*# Criando um gráfico para visualizar melhor*

*x = np.linspace(media\_populacional - 4\*erro\_padrao, media\_populacional + 4\*erro\_padrao, 1000)*

*y = stats.norm.pdf(x, media\_populacional, erro\_padrao)*

*plt.figure(figsize=(10, 6))*

*plt.plot(x, y, 'b-', linewidth=2)*

*plt.axvline(media\_populacional, color='k', linestyle='--', alpha=0.5)*

*# Sombreando a região dentro de ±5*

*x\_fill\_5 = np.linspace(media\_populacional-5, media\_populacional+5, 100)*

*y\_fill\_5 = stats.norm.pdf(x\_fill\_5, media\_populacional, erro\_padrao)*

*plt.fill\_between(x\_fill\_5, y\_fill\_5, color='red', alpha=0.3, label=f'±5 ({prob\_dentro\_5\*100:.2f}%)')*

*# Sombreando a região dentro de ±10*

*x\_fill\_10 = np.linspace(media\_populacional-10, media\_populacional+10, 100)*

*y\_fill\_10 = stats.norm.pdf(x\_fill\_10, media\_populacional, erro\_padrao)*

*plt.fill\_between(x\_fill\_10, y\_fill\_10, color='blue', alpha=0.2, label=f'±10 ({prob\_dentro\_10\*100:.2f}%)')*

*plt.title('Distribuição Amostral da Média (n=100)')*

*plt.xlabel('Média Amostral (x̄)')*

*plt.ylabel('Densidade de Probabilidade')*

*plt.axvline(media\_populacional-5, color='r', linestyle='--', alpha=0.7)*

*plt.axvline(media\_populacional+5, color='r', linestyle='--', alpha=0.7)*

*plt.axvline(media\_populacional-10, color='b', linestyle='--', alpha=0.7)*

*plt.axvline(media\_populacional+10, color='b', linestyle='--', alpha=0.7)*

*plt.grid(True, alpha=0.3)*

*plt.legend()*

*# Método alternativo de cálculo usando a distribuição normal padrão diretamente*

*# Estes cálculos servem para verificar os resultados acima*

*print("\nVerificação usando valores de z:")*

*print(f"Para ±5: z = ±{5/erro\_padrao:.4f}")*

*print(f"P(-{5/erro\_padrao:.4f} < Z < {5/erro\_padrao:.4f}) = {prob\_dentro\_5:.4f}")*

*print(f"Para ±10: z = ±{10/erro\_padrao:.4f}")*

*print(f"P(-{10/erro\_padrao:.4f} < Z < {10/erro\_padrao:.4f}) = {prob\_dentro\_10:.4f}")*

*plt.show()*

*#Cap.7:22*

*import numpy as np*

*import matplotlib.pyplot as plt*

*from scipy import stats*

*import pandas as pd*

*# Vamos assumir que a população de gestores tem:*

*# - média populacional μ = 100 (por exemplo, pontuação em alguma avaliação)*

*# - desvio padrão populacional σ = 15*

*media\_populacional = 100*

*desvio\_padrao\_populacional = 15*

*# Tamanhos de amostra a serem analisados*

*tamanho\_amostra\_1 = 60  # Parte a do problema*

*tamanho\_amostra\_2 = 120  # Parte b do problema*

*tamanhos\_adicionais = [30, 240, 480]  # Para a parte c, demonstrar o efeito do aumento do tamanho amostral*

*# Número de simulações para cada tamanho de amostra*

*num\_simulacoes = 10000*

*# a. Distribuição amostral de x̄ com n = 60*

*medias\_amostrais\_60 = []*

*for \_ in range(num\_simulacoes):*

*amostra = np.random.normal(media\_populacional, desvio\_padrao\_populacional, tamanho\_amostra\_1)*

*media\_amostral = np.mean(amostra)*

*medias\_amostrais\_60.append(media\_amostral)*

*# b. Distribuição amostral de x̄ com n = 120*

*medias\_amostrais\_120 = []*

*for \_ in range(num\_simulacoes):*

*amostra = np.random.normal(media\_populacional, desvio\_padrao\_populacional, tamanho\_amostra\_2)*

*media\_amostral = np.mean(amostra)*

*medias\_amostrais\_120.append(media\_amostral)*

*plt.show()*

*# c. Distribuições amostrais adicionais para demonstrar o efeito do aumento do tamanho amostral*

*medias\_adicionais = {}*

*for n in tamanhos\_adicionais:*

*medias\_adicionais[n] = []*

*for \_ in range(num\_simulacoes):*

*amostra = np.random.normal(media\_populacional, desvio\_padrao\_populacional, n)*

*media\_amostral = np.mean(amostra)*

*medias\_adicionais[n].append(media\_amostral)*

*# Calcular erro padrão teórico para cada tamanho de amostra*

*erro\_padrao\_60 = desvio\_padrao\_populacional / np.sqrt(tamanho\_amostra\_1)*

*erro\_padrao\_120 = desvio\_padrao\_populacional / np.sqrt(tamanho\_amostra\_2)*

*# Criar visualizações*

*plt.figure(figsize=(15, 10))*

*# Gráfico 1: Comparação das distribuições n=60 e n=120*

*plt.subplot(2, 1, 1)*

*plt.hist(medias\_amostrais\_60, bins=30, alpha=0.5, color='blue', label=f'n=60, σx̄={erro\_padrao\_60:.2f}')*

*plt.hist(medias\_amostrais\_120, bins=30, alpha=0.5, color='red', label=f'n=120, σx̄={erro\_padrao\_120:.2f}')*

*plt.axvline(media\_populacional, color='black', linestyle='--', linewidth=1)*

*plt.title('Distribuições Amostrais das Médias para n=60 e n=120')*

*plt.xlabel('Média Amostral (x̄)')*

*plt.ylabel('Frequência')*

*plt.legend()*

*plt.grid(alpha=0.3)*

*# Gráfico 2: Demonstração do efeito do aumento do tamanho amostral*

*plt.subplot(2, 1, 2)*

*# Lista completa de tamanhos amostrais*

*todos\_tamanhos = [tamanho\_amostra\_1, tamanho\_amostra\_2] + tamanhos\_adicionais*

*todos\_tamanhos.sort()  # Ordenar para melhor visualização*

*# Cores para cada tamanho amostral*

*cores = plt.cm.viridis(np.linspace(0, 1, len(todos\_tamanhos)))*

*# Plotar densidade de probabilidade para cada tamanho amostral*

*for i, n in enumerate(todos\_tamanhos):*

*if n in tamanhos\_adicionais:*

*dados = medias\_adicionais[n]*

*elif n == tamanho\_amostra\_1:*

*dados = medias\_amostrais\_60*

*else:*

*dados = medias\_amostrais\_120*

*erro\_padrao = desvio\_padrao\_populacional / np.sqrt(n)*

*# Usar KDE (Kernel Density Estimation) para visualizar a densidade*

*x = np.linspace(min(dados), max(dados), 1000)*

*kernel = stats.gaussian\_kde(dados)*

*plt.plot(x, kernel(x), color=cores[i], label=f'n={n}, σx̄={erro\_padrao:.2f}')*

*plt.axvline(media\_populacional, color='black', linestyle='--', linewidth=1, label='Média Populacional (μ)')*

*plt.title('Efeito do Aumento do Tamanho Amostral na Distribuição Amostral')*

*plt.xlabel('Média Amostral (x̄)')*

*plt.ylabel('Densidade de Probabilidade')*

*plt.legend(loc='upper right', fontsize=8)*

*plt.grid(alpha=0.3)*

*plt.tight\_layout()*

*# Tabela com estatísticas comparativas*

*estatisticas = {'Tamanho da Amostra (n)': [],*

*'Erro Padrão Teórico (σx̄)': [],*

*'Média das Médias Amostrais': [],*

*'Desvio Padrão Empírico das Médias': []}*

*for n in todos\_tamanhos:*

*if n in tamanhos\_adicionais:*

*dados = medias\_adicionais[n]*

*elif n == tamanho\_amostra\_1:*

*dados = medias\_amostrais\_60*

*else:*

*dados = medias\_amostrais\_120*

*erro\_padrao\_teorico = desvio\_padrao\_populacional / np.sqrt(n)*

*media\_empirica = np.mean(dados)*

*desvio\_padrao\_empirico = np.std(dados)*

*estatisticas['Tamanho da Amostra (n)'].append(n)*

*estatisticas['Erro Padrão Teórico (σx̄)'].append(erro\_padrao\_teorico)*

*estatisticas['Média das Médias Amostrais'].append(media\_empirica)*

*estatisticas['Desvio Padrão Empírico das Médias'].append(desvio\_padrao\_empirico)*

*# Criar dataframe para exibir as estatísticas*

*df\_estatisticas = pd.DataFrame(estatisticas)*

*# Relatório escrito das observações*

*print("RELATÓRIO DE ANÁLISE DA DISTRIBUIÇÃO AMOSTRAL")*

*print("=============================================\n")*

*print("a. Distribuição amostral de x̄ com n = 60:")*

*print(f"   - Média teórica: {media\_populacional}")*

*print(f"   - Erro padrão teórico: {erro\_padrao\_60:.4f}")*

*print(f"   - Média empírica obtida: {np.mean(medias\_amostrais\_60):.4f}")*

*print(f"   - Desvio padrão empírico: {np.std(medias\_amostrais\_60):.4f}")*

*print("\nb. Distribuição amostral de x̄ com n = 120:")*

*print(f"   - Média teórica: {media\_populacional}")*

*print(f"   - Erro padrão teórico: {erro\_padrao\_120:.4f}")*

*print(f"   - Média empírica obtida: {np.mean(medias\_amostrais\_120):.4f}")*

*print(f"   - Desvio padrão empírico: {np.std(medias\_amostrais\_120):.4f}")*

*print("\nc. Declaração geral sobre o que acontece com a distribuição amostral quando o tamanho amostral aumenta:")*

*print("   À medida que o tamanho da amostra aumenta, observamos que:")*

*print("   1. A distribuição amostral da média se aproxima cada vez mais de uma distribuição normal,")*

*print("      independentemente da forma da distribuição da população original (Teorema do Limite Central).")*

*print("   2. A variabilidade da distribuição amostral diminui, ou seja, o erro padrão (σx̄) diminui")*

*print("      proporcionalmente à raiz quadrada do tamanho da amostra (σx̄ = σ/√n).")*

*print("   3. A média da distribuição amostral permanece igual à média populacional (μ),")*

*print("      independentemente do tamanho da amostra.")*

*print("\n   Esta generalização é lógica porque:")*

*print("   - Quanto maior a amostra, mais informação temos sobre a população, o que reduz a incerteza")*

*print("     na estimativa da média populacional.")*

*print("   - A redução do erro padrão é proporcional à raiz quadrada do tamanho da amostra (não linear),")*

*print("     o que significa que o ganho em precisão diminui à medida que o tamanho da amostra aumenta.")*

*print("   - Isso explica porque, em pesquisas estatísticas, há um ponto onde aumentar o tamanho da amostra")*

*print("     não traz ganhos significativos em precisão que justifiquem o custo adicional da coleta de dados.")*

*print("\nTabela comparativa de estatísticas para diferentes tamanhos amostrais:")*

*print(df\_estatisticas.to\_string(index=False))*

*plt.show()*

*#Cap.7: 23*

*import numpy as np*

*from scipy import stats*

*import matplotlib.pyplot as plt*

*# Dados do problema:*

*# - Sabemos que para n=30, a probabilidade de se obter uma média amostral dentro*

*#   de ±US$ 500 da média populacional é 0,5034*

*# - Precisamos calcular essa probabilidade para n=60 e n=120*

*# Vamos determinar o desvio padrão populacional utilizando a informação fornecida*

*# Para n=30 e probabilidade 0,5034 no intervalo ±500*

*# Primeiro, vamos encontrar o valor z correspondente à probabilidade 0,5034/2 = 0,2517*

*# (dividimos por 2 porque queremos a área de um lado, já que o intervalo é ±500)*

*z = stats.norm.ppf(0.5034/2 + 0.5)  # Somamos 0.5 porque queremos a área à direita do centro*

*# Com o valor de z, podemos calcular o desvio padrão populacional*

*# Para n=30, temos: z = 500 / (sigma/sqrt(30))*

*# Portanto: sigma = 500 \* sqrt(30) / z*

*sigma = 500 \* np.sqrt(30) / z*

*print(f"Valor de z para probabilidade 0,5034: {z:.4f}")*

*print(f"Desvio padrão populacional estimado: {sigma:.4f}")*

*# a. Probabilidade para n=60*

*erro\_padrao\_60 = sigma / np.sqrt(60)*

*z\_60 = 500 / erro\_padrao\_60*

*prob\_60 = 2 \* (stats.norm.cdf(z\_60) - 0.5)  # Multiplicamos por 2 para obter a área total*

*print(f"\na. Para n=60:")*

*print(f"   Erro padrão: {erro\_padrao\_60:.4f}")*

*print(f"   Valor de z: {z\_60:.4f}")*

*print(f"   Probabilidade da média amostral estar dentro de ±US$ 500 da média populacional: {prob\_60:.4f}")*

*# b. Probabilidade para n=120*

*erro\_padrao\_120 = sigma / np.sqrt(120)*

*z\_120 = 500 / erro\_padrao\_120*

*prob\_120 = 2 \* (stats.norm.cdf(z\_120) - 0.5)*

*print(f"\nb. Para n=120:")*

*print(f"   Erro padrão: {erro\_padrao\_120:.4f}")*

*print(f"   Valor de z: {z\_120:.4f}")*

*print(f"   Probabilidade da média amostral estar dentro de ±US$ 500 da média populacional: {prob\_120:.4f}")*

*# Vamos criar uma visualização para ilustrar o efeito do tamanho da amostra na distribuição amostral*

*plt.figure(figsize=(12, 6))*

*# Definindo o intervalo para plotagem*

*media\_populacional = 0  # Podemos centralizar em 0 para simplificar*

*x = np.linspace(media\_populacional - 4\*erro\_padrao\_30, media\_populacional + 4\*erro\_padrao\_30, 1000)*

*# Calculando o erro padrão para n=30 (já sabemos que a probabilidade é 0,5034)*

*erro\_padrao\_30 = sigma / np.sqrt(30)*

*# Plotando as distribuições amostrais para os três tamanhos de amostra*

*plt.plot(x, stats.norm.pdf(x, media\_populacional, erro\_padrao\_30),*

*'b-', label=f'n=30, σx̄={erro\_padrao\_30:.2f}, P(±500)={0.5034:.4f}')*

*plt.plot(x, stats.norm.pdf(x, media\_populacional, erro\_padrao\_60),*

*'r-', label=f'n=60, σx̄={erro\_padrao\_60:.2f}, P(±500)={prob\_60:.4f}')*

*plt.plot(x, stats.norm.pdf(x, media\_populacional, erro\_padrao\_120),*

*'g-', label=f'n=120, σx̄={erro\_padrao\_120:.2f}, P(±500)={prob\_120:.4f}')*

*# Marcando os limites de ±US$ 500*

*plt.axvline(media\_populacional - 500, color='k', linestyle='--', alpha=0.7)*

*plt.axvline(media\_populacional + 500, color='k', linestyle='--', alpha=0.7)*

*# Preenchendo as áreas dentro do intervalo ±US$ 500 para cada distribuição*

*x\_fill = np.linspace(media\_populacional - 500, media\_populacional + 500, 100)*

*y\_fill\_30 = stats.norm.pdf(x\_fill, media\_populacional, erro\_padrao\_30)*

*plt.fill\_between(x\_fill, y\_fill\_30, color='blue', alpha=0.2)*

*y\_fill\_60 = stats.norm.pdf(x\_fill, media\_populacional, erro\_padrao\_60)*

*plt.fill\_between(x\_fill, y\_fill\_60, color='red', alpha=0.2)*

*y\_fill\_120 = stats.norm.pdf(x\_fill, media\_populacional, erro\_padrao\_120)*

*plt.fill\_between(x\_fill, y\_fill\_120, color='green', alpha=0.2)*

*plt.title('Distribuições Amostrais da Média para Diferentes Tamanhos de Amostra')*

*plt.xlabel('Desvio da Média Amostral em relação à Média Populacional (US$)')*

*plt.ylabel('Densidade de Probabilidade')*

*plt.legend()*

*plt.grid(True, alpha=0.3)*

*plt.axvline(media\_populacional, color='k', linestyle='-', alpha=0.5)*

*# Adicionando anotações para explicar o gráfico*

*plt.annotate(f'±US$ 500', xy=(500, 0), xytext=(520, 0.0003),*

*arrowprops=dict(facecolor='black', shrink=0.05, width=1.5, headwidth=8),*

*fontsize=10)*

*# Exibindo a solução visual*

*plt.tight\_layout()*

*plt.show()*

*#Cap.7:28*

*import numpy as np*

*from scipy import stats*

*import matplotlib.pyplot as plt*

*# Dados do problema*

*# Califórnia:*

*media\_calif = 22  # polegadas (média populacional)*

*n\_calif = 30  # tamanho da amostra*

*# Nova York:*

*media\_ny = 42  # polegadas (média populacional)*

*n\_ny = 45  # tamanho da amostra*

*# Desvio padrão populacional para ambos os estados*

*desvio\_padrao = 4  # polegadas*

*# a. Distribuição da precipitação média anual - Califórnia*

*erro\_padrao\_calif = desvio\_padrao / np.sqrt(n\_calif)*

*# b. Probabilidade da média amostral estar dentro de 1 polegada da média populacional - Califórnia*

*z\_calif = 1 / erro\_padrao\_calif*

*prob\_calif = 2 \* (stats.norm.cdf(z\_calif) - 0.5)  # Multiplica por 2 para obter a área total (duas caudas)*

*# c. Probabilidade da média amostral estar dentro de 1 polegada da média populacional - Nova York*

*erro\_padrao\_ny = desvio\_padrao / np.sqrt(n\_ny)*

*z\_ny = 1 / erro\_padrao\_ny*

*prob\_ny = 2 \* (stats.norm.cdf(z\_ny) - 0.5)*

*# d. Comparação das probabilidades*

*# Já calculamos as probabilidades nas partes b e c*

*# Visualização para ajudar na interpretação*

*plt.figure(figsize=(12, 8))*

*plt.show()*

*# Definindo intervalo para o eixo x*

*x\_calif = np.linspace(media\_calif - 4\*erro\_padrao\_calif, media\_calif + 4\*erro\_padrao\_calif, 1000)*

*x\_ny = np.linspace(media\_ny - 4\*erro\_padrao\_ny, media\_ny + 4\*erro\_padrao\_ny, 1000)*

*# Gráfico 1: Distribuição amostral para Califórnia*

*plt.subplot(2, 1, 1)*

*plt.plot(x\_calif, stats.norm.pdf(x\_calif, media\_calif, erro\_padrao\_calif), 'b-',*

*label=f'Califórnia (n={n\_calif}, σx̄={erro\_padrao\_calif:.4f})')*

*# Área dentro de ±1 polegada da média*

*x\_fill\_calif = np.linspace(media\_calif - 1, media\_calif + 1, 100)*

*y\_fill\_calif = stats.norm.pdf(x\_fill\_calif, media\_calif, erro\_padrao\_calif)*

*plt.fill\_between(x\_fill\_calif, y\_fill\_calif, color='blue', alpha=0.3,*

*label=f'P(±1 polegada) = {prob\_calif:.4f}')*

*plt.axvline(media\_calif, color='k', linestyle='--')*

*plt.axvline(media\_calif - 1, color='r', linestyle=':')*

*plt.axvline(media\_calif + 1, color='r', linestyle=':')*

*plt.title('Distribuição Amostral da Precipitação Média Anual - Califórnia')*

*plt.xlabel('Precipitação Média Anual (polegadas)')*

*plt.ylabel('Densidade de Probabilidade')*

*plt.legend()*

*plt.grid(True, alpha=0.3)*

*# Gráfico 2: Distribuição amostral para Nova York*

*plt.subplot(2, 1, 2)*

*plt.plot(x\_ny, stats.norm.pdf(x\_ny, media\_ny, erro\_padrao\_ny), 'g-',*

*label=f'Nova York (n={n\_ny}, σx̄={erro\_padrao\_ny:.4f})')*

*# Área dentro de ±1 polegada da média*

*x\_fill\_ny = np.linspace(media\_ny - 1, media\_ny + 1, 100)*

*y\_fill\_ny = stats.norm.pdf(x\_fill\_ny, media\_ny, erro\_padrao\_ny)*

*plt.fill\_between(x\_fill\_ny, y\_fill\_ny, color='green', alpha=0.3,*

*label=f'P(±1 polegada) = {prob\_ny:.4f}')*

*plt.axvline(media\_ny, color='k', linestyle='--')*

*plt.axvline(media\_ny - 1, color='r', linestyle=':')*

*plt.axvline(media\_ny + 1, color='r', linestyle=':')*

*plt.title('Distribuição Amostral da Precipitação Média Anual - Nova York')*

*plt.xlabel('Precipitação Média Anual (polegadas)')*

*plt.ylabel('Densidade de Probabilidade')*

*plt.legend()*

*plt.grid(True, alpha=0.3)*

*plt.tight\_layout()*

*# Gráfico 3: Comparação direta das duas distribuições amostrais*

*plt.figure(figsize=(12, 6))*

*plt.show()*

*# Normalizamos as distribuições para centrá-las em 0 para facilitar a comparação*

*x\_calif\_norm = np.linspace(-4\*erro\_padrao\_calif, 4\*erro\_padrao\_calif, 1000)*

*x\_ny\_norm = np.linspace(-4\*erro\_padrao\_ny, 4\*erro\_padrao\_ny, 1000)*

*plt.plot(x\_calif\_norm, stats.norm.pdf(x\_calif\_norm, 0, erro\_padrao\_calif), 'b-',*

*label=f'Califórnia (n={n\_calif}, σx̄={erro\_padrao\_calif:.4f}, P(±1)={prob\_calif:.4f})')*

*plt.plot(x\_ny\_norm, stats.norm.pdf(x\_ny\_norm, 0, erro\_padrao\_ny), 'g-',*

*label=f'Nova York (n={n\_ny}, σx̄={erro\_padrao\_ny:.4f}, P(±1)={prob\_ny:.4f})')*

*# Área dentro de ±1 polegada*

*x\_fill\_norm = np.linspace(-1, 1, 100)*

*y\_fill\_calif\_norm = stats.norm.pdf(x\_fill\_norm, 0, erro\_padrao\_calif)*

*y\_fill\_ny\_norm = stats.norm.pdf(x\_fill\_norm, 0, erro\_padrao\_ny)*

*plt.fill\_between(x\_fill\_norm, y\_fill\_calif\_norm, color='blue', alpha=0.3)*

*plt.fill\_between(x\_fill\_norm, y\_fill\_ny\_norm, color='green', alpha=0.3)*

*plt.axvline(0, color='k', linestyle='-')*

*plt.axvline(-1, color='r', linestyle='--')*

*plt.axvline(1, color='r', linestyle='--')*

*plt.title('Comparação das Distribuições Amostrais (Centralizadas)')*

*plt.xlabel('Desvio da Média Amostral em relação à Média Populacional (polegadas)')*

*plt.ylabel('Densidade de Probabilidade')*

*plt.legend()*

*plt.grid(True, alpha=0.3)*

*plt.tight\_layout()*

*# Resultados finais*

*print("RESULTADOS DA ANÁLISE DE DISTRIBUIÇÕES AMOSTRAIS")*

*print("=================================================\n")*

*print("a. Distribuição da precipitação média anual para a Califórnia:")*

*print(f"   - Média da distribuição amostral: {media\_calif} polegadas")*

*print(f"   - Erro padrão (desvio padrão da distribuição amostral): {erro\_padrao\_calif:.4f} polegadas")*

*print(f"   - A distribuição segue uma Normal com N({media\_calif}, {erro\_padrao\_calif:.4f}²)")*

*print("\nb. Probabilidade da média amostral estar dentro de 1 polegada da média populacional para a Califórnia:")*

*print(f"   P({media\_calif-1} < x̄ < {media\_calif+1}) = {prob\_calif:.4f} = {prob\_calif\*100:.2f}%")*

*print("\nc. Probabilidade da média amostral estar dentro de 1 polegada da média populacional para Nova York:")*

*print(f"   P({media\_ny-1} < x̄ < {media\_ny+1}) = {prob\_ny:.4f} = {prob\_ny\*100:.2f}%")*

*print("\nd. Comparação das probabilidades:")*

*if prob\_calif > prob\_ny:*

*maior\_prob = "Califórnia"*

*explicacao = "menor tamanho de amostra (30 vs 45)"*

*else:*

*maior\_prob = "Nova York"*

*explicacao = "maior tamanho de amostra (45 vs 30)"*

*print(f"   A probabilidade é maior para {maior\_prob} apesar do {explicacao}.")*

*print("\n   Análise detalhada:")*

*print(f"   - Califórnia: n={n\_calif}, σx̄={erro\_padrao\_calif:.4f}, z={z\_calif:.4f}, P(±1)={prob\_calif:.4f}")*

*print(f"   - Nova York: n={n\_ny}, σx̄={erro\_padrao\_ny:.4f}, z={z\_ny:.4f}, P(±1)={prob\_ny:.4f}")*

*print("\n   Explicação:")*

*print("   Como o desvio padrão populacional é o mesmo (4 polegadas) para ambos os estados,")*

*print("   o erro padrão depende apenas do tamanho da amostra (σx̄ = σ/√n).")*

*print("   Nova York tem uma amostra maior (n=45 vs n=30), o que resulta em um menor erro padrão")*

*print("   e, consequentemente, uma distribuição amostral mais concentrada em torno da média populacional.")*

*print("   Isso aumenta a probabilidade de a média amostral estar dentro de ±1 polegada da média populacional.")*

*plt.show()*

*#Cap. 7:31*

*import numpy as np*

*import matplotlib.pyplot as plt*

*from scipy import stats*

*import seaborn as sns*

*# Dados do problema*

*p = 0.40  # proporção populacional*

*n = 100   # tamanho da amostra*

*# a. Valor esperado de p̂ (p chapéu)*

*valor\_esperado\_p\_chapeu = p*

*print(f"a. Valor esperado de p̂: {valor\_esperado\_p\_chapeu}")*

*# b. Erro padrão de p̂*

*erro\_padrao\_p\_chapeu = np.sqrt((p \* (1 - p)) / n)*

*print(f"b. Erro padrão de p̂: {erro\_padrao\_p\_chapeu:.4f}")*

*# c. Distribuição amostral de p̂*

*# Verificamos se satisfaz as condições para aproximação normal:*

*condicao\_np = n \* p*

*condicao\_nq = n \* (1 - p)*

*print(f"\nVerificação das condições para aproximação normal:")*

*print(f"n·p = {condicao\_np} ≥ 5? {'Sim' if condicao\_np >= 5 else 'Não'}")*

*print(f"n·(1-p) = {condicao\_nq} ≥ 5? {'Sim' if condicao\_nq >= 5 else 'Não'}")*

*# Como as condições são satisfeitas, podemos usar a aproximação normal*

*# Realizamos uma simulação para demonstrar a distribuição amostral de p̂*

*num\_simulacoes = 10000*

*proporcoes\_amostrais = []*

*# Gerando amostras aleatórias e calculando a proporção em cada uma*

*for \_ in range(num\_simulacoes):*

*# Gerando amostra de uma distribuição Bernoulli com p=0.40*

*amostra = np.random.binomial(1, p, n)*

*# Calculando a proporção na amostra*

*proporcao\_amostral = np.mean(amostra)*

*proporcoes\_amostrais.append(proporcao\_amostral)*

*# Calculando média e desvio padrão empíricos*

*media\_empirica = np.mean(proporcoes\_amostrais)*

*desvio\_padrao\_empirico = np.std(proporcoes\_amostrais)*

*# Criando a visualização da distribuição amostral de p̂*

*plt.figure(figsize=(12, 8))*

*# Histograma das proporções amostrais simuladas*

*plt.subplot(2, 1, 1)*

*sns.histplot(proporcoes\_amostrais, bins=30, kde=True, color='blue', alpha=0.6)*

*# Adicionando a curva normal teórica*

*x = np.linspace(p - 4\*erro\_padrao\_p\_chapeu, p + 4\*erro\_padrao\_p\_chapeu, 1000)*

*y = stats.norm.pdf(x, p, erro\_padrao\_p\_chapeu)*

*plt.plot(x, y, 'r-', linewidth=2, label='Distribuição Normal Teórica')*

*plt.axvline(p, color='green', linestyle='--', linewidth=2,*

*label=f'Proporção Populacional (p = {p})')*

*plt.axvline(media\_empirica, color='purple', linestyle=':', linewidth=2,*

*label=f'Média Empírica (p̂ = {media\_empirica:.4f})')*

*plt.title('Distribuição Amostral da Proporção (p̂) - n = 100, p = 0.40')*

*plt.xlabel('Proporção Amostral (p̂)')*

*plt.ylabel('Frequência')*

*plt.legend()*

*plt.grid(alpha=0.3)*

*# Gráfico QQ para verificar normalidade*

*plt.subplot(2, 1, 2)*

*stats.probplot(proporcoes\_amostrais, dist="norm", plot=plt)*

*plt.title('Gráfico Q-Q: Verificação da Normalidade da Distribuição Amostral')*

*plt.grid(alpha=0.3)*

*plt.tight\_layout()*

*plt.show()*

*# d. O que a distribuição amostral de p̂ mostra*

*print("\nd. O que a distribuição amostral de p̂ mostra:")*

*print("   1. A distribuição amostral de p̂ segue aproximadamente uma distribuição normal")*

*print(f"      com média igual à proporção populacional (p = {p}) e erro padrão")*

*print(f"      σp̂ = √(p(1-p)/n) = {erro\_padrao\_p\_chapeu:.4f}.")*

*print("   2. Esta distribuição mostra como a proporção amostral (p̂) varia quando")*

*print("      diferentes amostras de tamanho 100 são retiradas da população.")*

*print("   3. A distribuição nos permite calcular probabilidades associadas a")*

*print("      possíveis valores de p̂ e construir intervalos de confiança para p.")*

*print("   4. A maioria das proporções amostrais estará em torno da proporção")*

*print("      populacional, com aproximadamente 95% dos valores dentro de ±2 erros")*

*print(f"      padrão: {p-2\*erro\_padrao\_p\_chapeu:.4f} a {p+2\*erro\_padrao\_p\_chapeu:.4f}.")*

*print("   5. À medida que o tamanho da amostra aumenta, a distribuição se torna")*

*print("      mais concentrada em torno da proporção populacional, reduzindo a")*

*print("      variabilidade das estimativas.")*

*print("\nConfirmação empírica dos parâmetros teóricos da distribuição:")*

*print(f"Média teórica de p̂: {p}")*

*print(f"Média empírica de p̂ (baseada na simulação): {media\_empirica:.4f}")*

*print(f"Erro padrão teórico: {erro\_padrao\_p\_chapeu:.4f}")*

*print(f"Desvio padrão empírico: {desvio\_padrao\_empirico:.4f}")*

*# Calculando algumas probabilidades de interesse*

*z\_1 = (0.45 - p) / erro\_padrao\_p\_chapeu*

*prob\_maior\_045 = 1 - stats.norm.cdf(z\_1)*

*print(f"\nProbabilidade de p̂ > 0.45: {prob\_maior\_045:.4f} = {prob\_maior\_045\*100:.2f}%")*

*z\_2 = (0.35 - p) / erro\_padrao\_p\_chapeu*

*prob\_menor\_035 = stats.norm.cdf(z\_2)*

*print(f"Probabilidade de p̂ < 0.35: {prob\_menor\_035:.4f} = {prob\_menor\_035\*100:.2f}%")*

*intervalo\_1\_sigma = 2 \* stats.norm.cdf(1) - 1*

*intervalo\_2\_sigma = 2 \* stats.norm.cdf(2) - 1*

*print(f"\nProbabilidade de p̂ estar dentro de ±1 erro padrão de p: {intervalo\_1\_sigma:.4f} = {intervalo\_1\_sigma\*100:.2f}%")*

*print(f"Probabilidade de p̂ estar dentro de ±2 erros padrão de p: {intervalo\_2\_sigma:.4f} = {intervalo\_2\_sigma\*100:.2f}%")*

*#Cap.8:2*

*import numpy as np*

*import scipy.stats as stats*

*# Dados do problema*

*media\_amostral = 32*

*n = 50*

*sigma = 6  # Desvio padrão populacional é dado (σ = 6)*

*# Para calcular o intervalo de confiança quando conhecemos o desvio padrão populacional,*

*# usamos a distribuição normal (z)*

*def calcular\_ic\_normal(nivel\_confianca):*

*# Obtendo o valor crítico z*

*z = stats.norm.ppf((1 + nivel\_confianca) / 2)*

*# Erro padrão da média*

*erro\_padrao = sigma / np.sqrt(n)*

*# Margem de erro*

*margem\_erro = z \* erro\_padrao*

*# Intervalo de confiança*

*limite\_inferior = media\_amostral - margem\_erro*

*limite\_superior = media\_amostral + margem\_erro*

*return limite\_inferior, limite\_superior*

*# a. Intervalo de confiança de 90% para a média populacional*

*ic\_90 = calcular\_ic\_normal(0.90)*

*print(f"a. IC 90%: ({ic\_90[0]:.4f}, {ic\_90[1]:.4f})")*

*# b. Intervalo de confiança de 95% para a média populacional*

*ic\_95 = calcular\_ic\_normal(0.95)*

*print(f"b. IC 95%: ({ic\_95[0]:.4f}, {ic\_95[1]:.4f})")*

*# c. Intervalo de confiança de 99% para a média populacional*

*ic\_99 = calcular\_ic\_normal(0.99)*

*print(f"c. IC 99%: ({ic\_99[0]:.4f}, {ic\_99[1]:.4f})")*

*#Cap 8:7*

*import numpy as np*

*import scipy.stats as stats*

*import pandas as pd*

*# Dados do problema*

*custo\_medio\_anual = 1843  # Custo médio anual em US$*

*desvio\_padrao = 255      # Desvio padrão populacional (σ = US$ 255)*

*tamanho\_amostra = 50     # Tamanho da amostra*

*# a. Margem de erro para o intervalo de confiança de 95%*

*def calcular\_margem\_erro(nivel\_confianca):*

*# Obtendo o valor crítico z para o nível de confiança especificado*

*z = stats.norm.ppf((1 + nivel\_confianca) / 2)*

*# Calculando o erro padrão da média*

*erro\_padrao = desvio\_padrao / np.sqrt(tamanho\_amostra)*

*# Calculando a margem de erro*

*margem\_erro = z \* erro\_padrao*

*return margem\_erro*

*# Calculando a margem de erro para 95% de confiança*

*margem\_erro\_95 = calcular\_margem\_erro(0.95)*

*print(f"a. Margem de erro para IC de 95%: US$ {margem\_erro\_95:.2f}")*

*# b. Simulando os dados dos 50 proprietários para cálculo do IC de 95%*

*# Vamos gerar dados aleatórios com base na média e desvio padrão informados*

*np.random.seed(42)  # Para reprodutibilidade*

*dados\_setters = np.random.normal(custo\_medio\_anual, desvio\_padrao, tamanho\_amostra)*

*# Calculando a média amostral*

*media\_amostral = np.mean(dados\_setters)*

*print(f"\nb. Média amostral calculada: US$ {media\_amostral:.2f}")*

*# Calculando o IC de 95% usando a distribuição t de Student (para amostra)*

*# Como estamos usando uma amostra e não sabemos o desvio padrão populacional,*

*# usamos o desvio padrão amostral e a distribuição t*

*desvio\_padrao\_amostral = np.std(dados\_setters, ddof=1)  # ddof=1 para desvio padrão amostral*

*erro\_padrao\_amostral = desvio\_padrao\_amostral / np.sqrt(tamanho\_amostra)*

*t\_critico = stats.t.ppf(0.975, tamanho\_amostra - 1)  # 0.975 para IC de 95% (bilateral)*

*limite\_inferior = media\_amostral - t\_critico \* erro\_padrao\_amostral*

*limite\_superior = media\_amostral + t\_critico \* erro\_padrao\_amostral*

*print(f"Intervalo de Confiança de 95% para o custo médio: (US$ {limite\_inferior:.2f}, US$ {limite\_superior:.2f})")*

*print(f"Margem de erro (usando a distribuição t): US$ {t\_critico \* erro\_padrao\_amostral:.2f}")*

*# Alternativa: calculando o IC usando a função stats.t.interval diretamente*

*ic\_95 = stats.t.interval(0.95, tamanho\_amostra - 1, loc=media\_amostral, scale=erro\_padrao\_amostral)*

*print(f"IC 95% (usando stats.t.interval): (US$ {ic\_95[0]:.2f}, US$ {ic\_95[1]:.2f})")*

*#Cap.8:11*

*import scipy.stats as stats*

*import numpy as np*

*# Definindo os graus de liberdade*

*gl = 16*

*# a. À direita de 2,120*

*t\_a = 2.120*

*prob\_a = 1 - stats.t.cdf(t\_a, gl)*

*print(f"a. Probabilidade à direita de 2,120: {prob\_a:.6f}")*

*# b. À esquerda de 1,337*

*t\_b = 1.337*

*prob\_b = stats.t.cdf(t\_b, gl)*

*print(f"b. Probabilidade à esquerda de 1,337: {prob\_b:.6f}")*

*# c. À esquerda de -1,746*

*t\_c = -1.746*

*prob\_c = stats.t.cdf(t\_c, gl)*

*print(f"c. Probabilidade à esquerda de -1,746: {prob\_c:.6f}")*

*# d. À direita de 2,583*

*t\_d = 2.583*

*prob\_d = 1 - stats.t.cdf(t\_d, gl)*

*print(f"d. Probabilidade à direita de 2,583: {prob\_d:.6f}")*

*# e. Entre -2,120 e 2,120*

*t\_e\_min = -2.120*

*t\_e\_max = 2.120*

*prob\_e = stats.t.cdf(t\_e\_max, gl) - stats.t.cdf(t\_e\_min, gl)*

*print(f"e. Probabilidade entre -2,120 e 2,120: {prob\_e:.6f}")*

*# f. Entre -1,746 e 1,746*

*t\_f\_min = -1.746*

*t\_f\_max = 1.746*

*prob\_f = stats.t.cdf(t\_f\_max, gl) - stats.t.cdf(t\_f\_min, gl)*

*print(f"f. Probabilidade entre -1,746 e 1,746: {prob\_f:.6f}")*

*plt.show()*

*#Cap.8:13*

*import numpy as np*

*import scipy.stats as stats*

*# Dados amostrais*

*dados = np.array([10, 8, 12, 15, 13, 11, 6, 5])*

*# a. Estimativa pontual da média populacional*

*media\_amostral = np.mean(dados)*

*# b. Estimativa pontual do desvio padrão populacional*

*# Usando n-1 no denominador (desvio padrão amostral)*

*desvio\_padrao\_amostral = np.std(dados, ddof=1)*

*# c. Calculando a margem de erro com 95% de confiança*

*n = len(dados)*

*graus\_liberdade = n - 1*

*t\_valor = stats.t.ppf(0.975, graus\_liberdade)  # t crítico para 95% de confiança*

*margem\_erro = t\_valor \* (desvio\_padrao\_amostral / np.sqrt(n))*

*# d. Intervalo de confiança de 95% para a média populacional*

*ic\_inferior = media\_amostral - margem\_erro*

*ic\_superior = media\_amostral + margem\_erro*

*# Exibindo os resultados*

*print(f"a. Estimativa pontual da média populacional: {media\_amostral:.4f}")*

*print(f"b. Estimativa pontual do desvio padrão populacional: {desvio\_padrao\_amostral:.4f}")*

*print(f"c. Margem de erro (95% de confiança): {margem\_erro:.4f}")*

*print(f"d. Intervalo de confiança de 95% para a média populacional: [{ic\_inferior:.4f}, {ic\_superior:.4f}]")*

*# Informações adicionais para verificação*

*print("\nInformações adicionais:")*

*print(f"Tamanho da amostra (n): {n}")*

*print(f"Graus de liberdade: {graus\_liberdade}")*

*print(f"Valor t crítico (95% de confiança): {t\_valor:.4f}")*

*print(f"Erro padrão da média: {desvio\_padrao\_amostral / np.sqrt(n):.4f}")*

*#Cap.8:19*

*import numpy as np*

*import scipy.stats as stats*

*import matplotlib.pyplot as plt*

*# Dados do problema*

*preco\_toquio = 40  # preço em US$ para duas pessoas em Tóquio*

*n\_hong\_kong = 42   # tamanho da amostra de Hong Kong*

*# Vamos simular os dados de Hong Kong com base nas informações*

*# Nota: Em um cenário real, usaríamos os 42 valores reais do arquivo HongKongMeals*

*# Para fins de exemplo, vamos gerar dados hipotéticos com média próxima de 45*

*np.random.seed(123)  # Para reprodutibilidade*

*dados\_hong\_kong = np.random.normal(45, 8, n\_hong\_kong)*

*# a. Calculando a margem de erro com 95% de confiança*

*media\_hong\_kong = np.mean(dados\_hong\_kong)*

*desvio\_padrao\_hong\_kong = np.std(dados\_hong\_kong, ddof=1)*

*erro\_padrao = desvio\_padrao\_hong\_kong / np.sqrt(n\_hong\_kong)*

*t\_valor = stats.t.ppf(0.975, n\_hong\_kong - 1)  # valor crítico t para 95% de confiança*

*margem\_erro = t\_valor \* erro\_padrao*

*# b. Calculando o intervalo de confiança de 95% para a média populacional*

*ic\_inferior = media\_hong\_kong - margem\_erro*

*ic\_superior = media\_hong\_kong + margem\_erro*

*# c. Comparação dos preços entre Hong Kong e Tóquio*

*diferenca\_percentual = ((media\_hong\_kong - preco\_toquio) / preco\_toquio) \* 100*

*# Exibindo resultados*

*print(f"Preço médio em Tóquio: US$ {preco\_toquio:.2f} para duas pessoas")*

*print(f"Preço médio estimado em Hong Kong: US$ {media\_hong\_kong:.2f} para duas pessoas")*

*print(f"\na. Margem de erro (95% de confiança): {margem\_erro:.2f}")*

*print(f"\nb. Intervalo de confiança de 95% para a média em Hong Kong: [US$ {ic\_inferior:.2f}, US$ {ic\_superior:.2f}]")*

*print(f"\nc. Comparação entre Hong Kong e Tóquio:")*

*if preco\_toquio < ic\_inferior:*

*print(f"   As refeições em Hong Kong são significativamente mais caras que em Tóquio (cerca de {diferenca\_percentual:.1f}% mais caras)")*

*elif preco\_toquio > ic\_superior:*

*print(f"   As refeições em Hong Kong são significativamente mais baratas que em Tóquio (cerca de {-diferenca\_percentual:.1f}% mais baratas)")*

*else:*

*print(f"   Não há diferença estatisticamente significativa entre os preços de Hong Kong e Tóquio")*

*# Visualização (opcional)*

*plt.figure(figsize=(10, 6))*

*plt.hist(dados\_hong\_kong, bins=10, alpha=0.7, color='skyblue', label='Preços em Hong Kong')*

*plt.axvline(media\_hong\_kong, color='blue', linestyle='dashed', linewidth=2, label=f'Média Hong Kong (US$ {media\_hong\_kong:.2f})')*

*plt.axvline(preco\_toquio, color='red', linestyle='dashed', linewidth=2, label=f'Preço Tóquio (US$ {preco\_toquio})')*

*plt.axvline(ic\_inferior, color='green', linestyle='dotted', linewidth=2, label=f'IC 95% Inferior (US$ {ic\_inferior:.2f})')*

*plt.axvline(ic\_superior, color='green', linestyle='dotted', linewidth=2, label=f'IC 95% Superior (US$ {ic\_superior:.2f})')*

*plt.xlabel('Preço (US$)')*

*plt.ylabel('Frequência')*

*plt.title('Comparação de Preços de Refeições para Duas Pessoas')*

*plt.legend()*

*plt.grid(True, alpha=0.3)*

*plt.show()*

*#Cap.8:22*

*import numpy as np*

*import scipy.stats as stats*

*# Dados do problema*

*receita\_total = 94.3e6  # US$ 94,3 milhões*

*n\_cinemas = 30  # tamanho da amostra*

*preco\_ingresso = 8.11  # US$ por ingresso*

*total\_salas = 4080  # número total de salas exibindo o filme*

*# Vamos supor que os dados da amostra estão em um arquivo guardians.txt*

*# Como não temos acesso aos dados reais, vamos simular dados compatíveis com o problema*

*np.random.seed(123)  # Para reprodutibilidade*

*# Simulando receitas por sala que resultariam em aproximadamente US$ 94,3 milhões no total*

*receita\_media\_por\_sala = receita\_total / total\_salas*

*# Adicionando alguma variação para simular os dados da amostra*

*receitas\_amostra = np.random.normal(receita\_media\_por\_sala, receita\_media\_por\_sala \* 0.2, n\_cinemas)*

*# a. Calculando o intervalo de confiança de 95% para a receita média por sala*

*media\_amostra = np.mean(receitas\_amostra)*

*desvio\_padrao\_amostra = np.std(receitas\_amostra, ddof=1)*

*erro\_padrao = desvio\_padrao\_amostra / np.sqrt(n\_cinemas)*

*t\_valor = stats.t.ppf(0.975, n\_cinemas - 1)  # valor crítico t para 95% de confiança*

*margem\_erro = t\_valor \* erro\_padrao*

*ic\_inferior = media\_amostra - margem\_erro*

*ic\_superior = media\_amostra + margem\_erro*

*# b. Calculando o número médio de clientes por sala*

*clientes\_por\_sala = media\_amostra / preco\_ingresso*

*ic\_clientes\_inferior = ic\_inferior / preco\_ingresso*

*ic\_clientes\_superior = ic\_superior / preco\_ingresso*

*# c. Estimando o número total de clientes no fim de semana*

*total\_clientes\_estimado = clientes\_por\_sala \* total\_salas*

*# Exibindo resultados*

*print("Análise da Bilheteria de 'Guardiões da Galáxia'")*

*print("==============================================")*

*print(f"Receita total reportada: US$ {receita\_total/1e6:.1f} milhões")*

*print(f"Amostra de {n\_cinemas} cinemas")*

*print(f"Preço do ingresso: US$ {preco\_ingresso}")*

*print(f"Total de salas exibindo o filme: {total\_salas}\n")*

*print(f"a. Intervalo de confiança de 95% para a receita média por sala:")*

*print(f"   [US$ {ic\_inferior:.2f}, US$ {ic\_superior:.2f}]")*

*print(f"   Receita média estimada por sala: US$ {media\_amostra:.2f}")*

*print(f"   Margem de erro: US$ {margem\_erro:.2f}\n")*

*print(f"b. Número médio de clientes por sala de cinema:")*

*print(f"   {clientes\_por\_sala:.0f} clientes por sala")*

*print(f"   Intervalo de confiança de 95%: [{ic\_clientes\_inferior:.0f}, {ic\_clientes\_superior:.0f}] clientes\n")*

*print(f"c. Número total estimado de clientes que assistiram ao filme:")*

*print(f"   {total\_clientes\_estimado:.0f} clientes no total")*

*print(f"   (Total de vendas de ingressos em {total\_salas} salas no fim de semana prolongado)")*

*#Cap.8:23*

*import numpy as np*

*import scipy.stats as stats*

*import math*

*# Dados do problema*

*margem\_erro\_desejada = 10*

*desvio\_padrao\_populacional = 40*

*nivel\_confianca = 0.95*

*# Para um IC de 95%, o z-crítico é aproximadamente 1.96*

*z\_critico = stats.norm.ppf(1 - (1 - nivel\_confianca) / 2)*

*# Fórmula para calcular o tamanho amostral necessário:*

*# n = (z^2 \* σ^2) / E^2*

*# Onde:*

*# n = tamanho da amostra*

*# z = valor crítico z*

*# σ = desvio padrão populacional*

*# E = margem de erro desejada*

*tamanho\_amostral = ((z\_critico\*\*2) \* (desvio\_padrao\_populacional\*\*2)) / (margem\_erro\_desejada\*\*2)*

*# Como o tamanho da amostra deve ser um número inteiro, arredondamos para cima*

*tamanho\_amostral\_arredondado = math.ceil(tamanho\_amostral)*

*# Exibindo os resultados*

*print(f"Margem de erro desejada: {margem\_erro\_desejada}")*

*print(f"Desvio padrão populacional: {desvio\_padrao\_populacional}")*

*print(f"Nível de confiança: {nivel\_confianca \* 100}%")*

*print(f"Valor z crítico para IC {nivel\_confianca \* 100}%: {z\_critico:.4f}")*

*print(f"\nTamanho amostral necessário (fórmula): {tamanho\_amostral:.4f}")*

*print(f"Tamanho amostral necessário (arredondado para cima): {tamanho\_amostral\_arredondado}")*

*# Verificação (opcional)*

*# Calculando a margem de erro com o tamanho de amostra arredondado para confirmar*

*margem\_erro\_verificacao = z\_critico \* (desvio\_padrao\_populacional / math.sqrt(tamanho\_amostral\_arredondado))*

*print(f"\nVerificação - Margem de erro com n = {tamanho\_amostral\_arredondado}: {margem\_erro\_verificacao:.4f}")*