**Métodos Quantitativos I**

**Camila Guedes de Farias**

4ª Lista de Exercícios, 22/4/2025

*Descrição:*

*Exercícios do capítulo 9 do livro Estatística Aplicada a Administração e Economia.*

*Capítulo 9 - Testes de hipóteses: 4, 7, 9, 14, 17, 24, 32*

***Cap. 9:4***

1. *Hipóteses alternativas:*

*H0: μ >= 220 (o novo método não reduz os custos)*

*H1: μ < 220 (o novo método reduz os custos)*

1. *Conclusão de quando H0 não pode ser rejeitada:*

*H0 não pode ser rejeitada quando o valor-p ≥ α ou quando t\_calculado ≥ t\_crítico*

1. *Conclusão de quando H0 pode ser rejeitada:*

*H0 pode ser rejeitada quando o valor-p < α ou quando t\_calculado < t\_crítico*

***Cap. 9:7***

Análise do Programa de Incentivo de Vendas da Carpetland

Média atual de vendas: US$ 8.0 mil por semana

Média após programa de incentivo: US$ 8.54 mil por semana

Tamanho da amostra experimental: 30

Desvio padrão amostral: US$ 1.19 mil

Estatística t calculada: 2.5131

Valor-p: 0.0089

Valor crítico t (α = 0.05): 1.6991

Rejeitamos H0 com nível de significância 0.05.

Há evidência estatística de que o programa de incentivo aumentou a média de vendas.

1. Hipóteses formuladas:

H0: μ ≤ $8,000 (o programa de incentivo não aumentou a média de vendas)

H1: μ > $8,000 (o programa de incentivo aumentou a média de vendas)

1. Erro Tipo I nesta situação:

Definição: Rejeitar H0 quando H0 é verdadeira.

Contexto: Concluir que o programa de incentivo aumentou as vendas quando na realidade não aumentou.

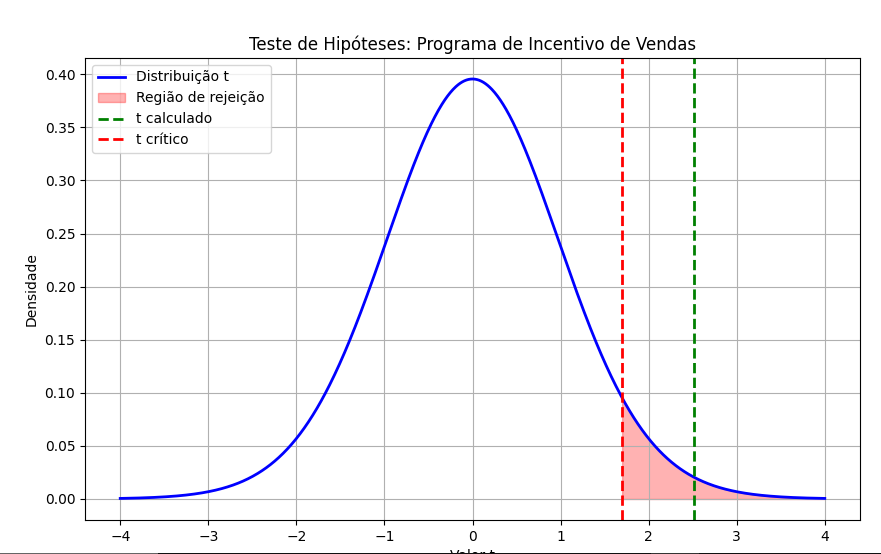
Consequências: A empresa implementaria o programa de incentivo acreditando em sua eficácia, mas estaria pagando mais aos vendedores sem obter o aumento esperado nas vendas. Isso resultaria em custos adicionais sem o retorno financeiro esperado.

1. Erro Tipo II nesta situação:

Definição: Não rejeitar H0 quando H0 é falsa.

Contexto: Concluir que o programa de incentivo não aumentou as vendas quando na realidade aumentou.

Consequências: A empresa deixaria de implementar um programa que realmente aumentaria as vendas. Isso resultaria em perda de oportunidade de aumento nas receitas e potencial vantagem competitiva.



***Cap. 9:9***

1. Estatística de teste: Z = -2.1213

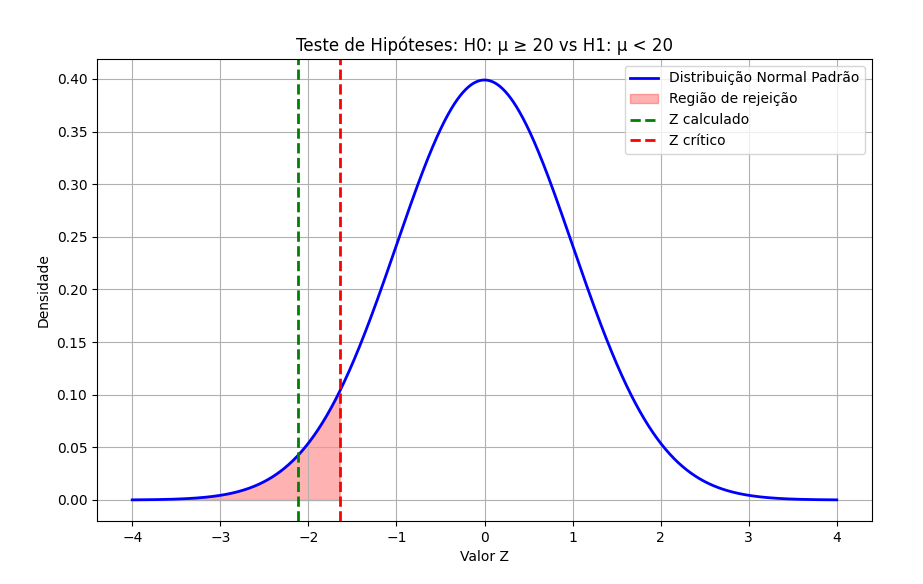
Valor-p = {p\_valor:.4f}

1. Valor-p = 0.0169

Conclusão com α = 0.05: {conclusao}

1. Conclusão com α = 0.05: Rejeitamos H0. Rejeitamos a hipótese nula.
2. Regra de rejeição: Rejeitar H0 se Z < -1.6449

Conclusão baseada na regra: Rejeitamos H0



***Cap. 9:14***

1. x̄ = 23

Estatística Z = 0.8660

Valor-p = 0.3865

Para α = 0.01, Não rejeitamos H0.

Não há evidência estatística suficiente para concluir que a média populacional é diferente de 22.

1. x̄ = 25.1

Estatística Z = 2.6847

Valor-p = 0.0073

Para α = 0.01, Rejeitamos H0.

Há evidência estatística de que a média populacional é diferente de 22.

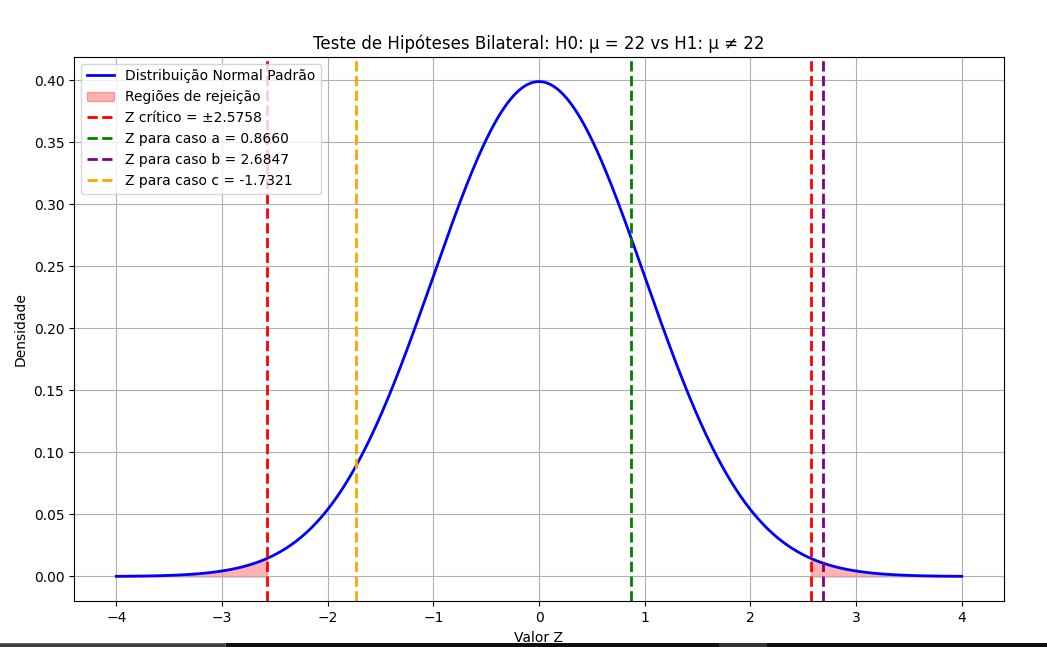
1. x̄ = 20

Estatística Z = -1.7321

Valor-p = 0.0833

Para α = 0.01, Não rejeitamos H0.

Não há evidência estatística suficiente para concluir que a média populacional é diferente de 22.



***Cap. 9:17***

1. Hipóteses para o teste:

H0: μ = $24.57 (salário médio na indústria manufatureira é igual ao reportado)

H1: μ ≠ $24.57 (salário médio na indústria manufatureira é diferente do reportado)

Justificativa: Como queremos verificar se o salário difere (podendo ser maior ou menor), usamos um teste bilateral.

1. Estatística de teste: Z = -1.5519

Valor-p = 0.1207

1. Usando α = 0.05:

Como valor-p (0.1207) ≥ α (0.05), Não rejeitamos H0.

Não há evidência estatística suficiente para concluir que o salário médio na indústria manufatureira é diferente de $24.57 por hora.

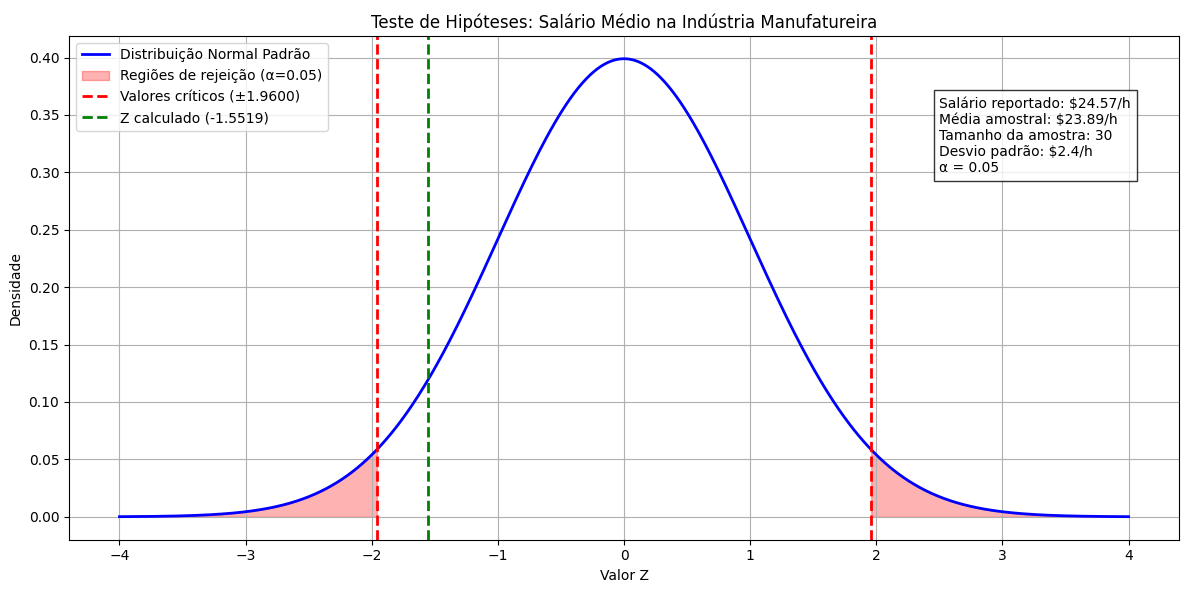
1. Usando o critério do valor crítico:

Valor crítico para α = 0.05 (teste bilateral): Z = ±1.9600

Regra de rejeição: Rejeitar H0 se |Z| > 1.9600

Como |Z| (1.5519) ≤ 1.9600, Não rejeitamos H0.

Não há evidência estatística suficiente para concluir que o salário médio na indústria manufatureira é diferente de $24.57 por hora.



***Cap. 9:24***

1. Estatística de teste t = -1.5396

Estatística de teste t: t = (x̄ - μ₀)/(s/√n) = (17 - 18)/(4,5/√48) = -1/(4,5/6,928) = -1/0,6495 = -1,5396

1. Para um teste bilateral com 47 graus de liberdade (n-1 = 48-1 = 47):

O valor-p é calculado como: valor-p = 2 × P(T₄₇ < -1,5396) = 2 × 0,0651 = 0,1302

Usando a tabela de distribuição t com 47 graus de liberdade, o valor mais próximo para t = -1,5396 nos dá um valor-p aproximado de 0,13.

Valor-p = 0.1304

1. Usando α = 0.05

Como o valor-p (0,1302) > α (0,05), não rejeitamos a hipótese nula.

Conclusão: Não há evidência estatística suficiente para concluir que a média populacional é diferente de 18.

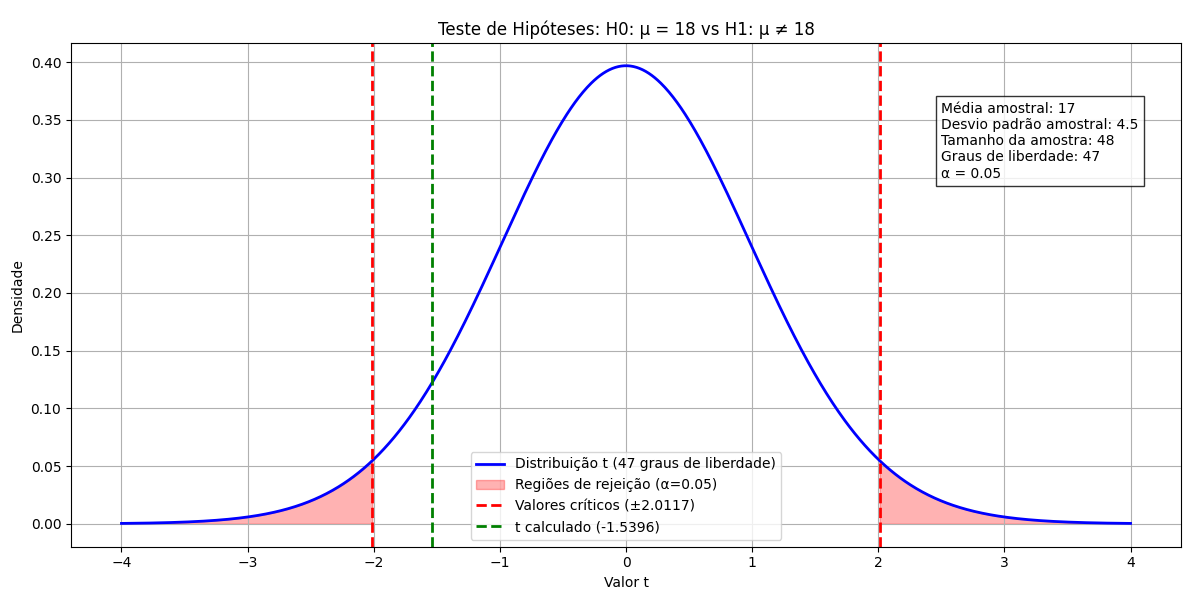
1. Para α = 0,05 em um teste bilateral com 47 graus de liberdade, o valor crítico é: t₀.₀₂₅,₄₇ = ±2,0117

Regra de rejeição: Rejeitar H₀ se |t| > 2,0117

Como |t| = |-1,5396| = 1,5396 < 2,0117, não rejeitamos H₀.

Conclusão usando o critério do valor crítico: Não há evidência estatística suficiente para concluir que a média populacional é diferente de 18.

Ambos os métodos (valor-p e valor crítico) chegam à mesma conclusão, como seria esperado.



***Cap. 9:32***

1. Hipóteses para o teste:

H0: μ = $10,192 (o preço médio na revendedora é igual à média nacional)

H1: μ ≠ $10,192 (o preço médio na revendedora é diferente da média nacional)

Justificativa: Como queremos determinar se existe uma diferença no preço médio (podendo ser maior ou menor), usamos um teste bilateral.

1. Estatística de teste: t = -4.1816

Graus de liberdade: 49

Valor-p = 0.0001

1. Com α = 0.05:

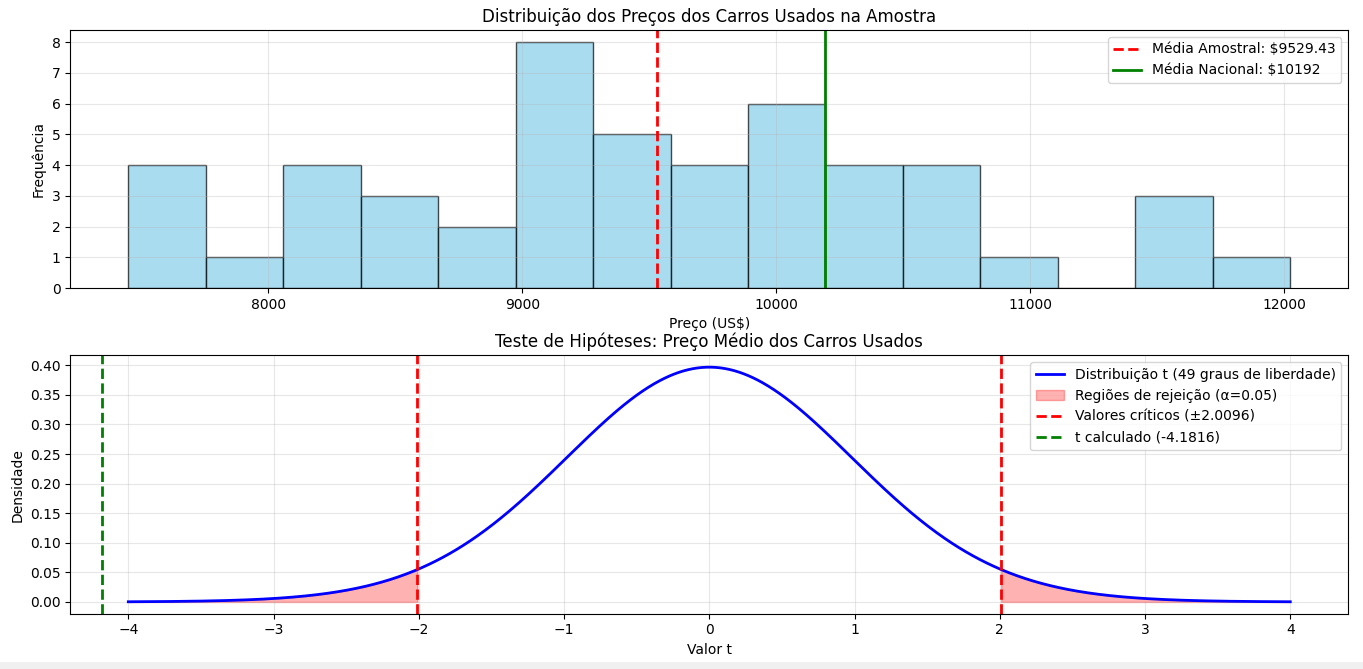
Como valor-p (0.0001) < α (0.05), Rejeitamos H0.

Há evidência estatística suficiente para concluir que o preço médio dos carros usados na revendedora de Kansas City é diferente da média nacional de $10,192.

Valor crítico para α = 0.05 (teste bilateral): t = ±2.0096

Regra de rejeição: Rejeitar H0 se |t| > 2.0096

Como |t| (4.1816) > 2.0096, rejeitamos H0.



***Apêndice:***

***Código em Python***

*# #Capítulo 9 - Testes de hipoteses: 4, 7, 9, 14, 17, 24, 32*

*#Cap. 9:4*

*import numpy as np*

*import scipy.stats as stats*

*import matplotlib.pyplot as plt*

*# Definindo o problema*

*custo\_atual = 220  # Custo médio do método atual (US$ por hora)*

*# Função para executar o teste de hipóteses*

*def teste\_hipoteses(dados\_novo\_metodo, nivel\_significancia=0.05):*

*# Tamanho da amostra*

*n = len(dados\_novo\_metodo)*

*# Média amostral do novo método*

*media\_novo = np.mean(dados\_novo\_metodo)*

*# Desvio padrão amostral do novo método*

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

*# Hipóteses:*

*# H0: μ >= 220 (o novo método não reduz os custos)*

*# H1: μ < 220 (o novo método reduz os custos)*

*# Estatística do teste t*

*t\_estatistica = (media\_novo - custo\_atual) / (desvio\_padrao / np.sqrt(n))*

*# Valor-p para teste unilateral à esquerda*

*p\_valor = stats.t.cdf(t\_estatistica, df=n-1)*

*# Valor crítico para rejeição de H0*

*t\_critico = stats.t.ppf(nivel\_significancia, df=n-1)*

*# Resultados*

*print(f"Dados do teste de hipóteses:")*

*print(f"Custo médio do método atual: US$ {custo\_atual} por hora")*

*print(f"Custo médio do novo método: US$ {media\_novo:.2f} por hora")*

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

*print(f"Desvio padrão amostral: {desvio\_padrao:.2f}")*

*print(f"Estatística t: {t\_estatistica:.4f}")*

*print(f"Valor-p: {p\_valor:.4f}")*

*print(f"Valor crítico t para α = {nivel\_significancia}: {t\_critico:.4f}")*

*# Conclusão sobre H0*

*print("\nConclusão:")*

*if p\_valor < nivel\_significancia:*

*print(f"Rejeitamos H0 com nível de significância {nivel\_significancia}.")*

*print("Há evidência estatística de que o novo método reduz os custos.")*

*else:*

*print(f"Não rejeitamos H0 com nível de significância {nivel\_significancia}.")*

*print("Não há evidência estatística suficiente de que o novo método reduz os custos.")*

*# Informações adicionais sobre quando rejeitar H0*

*print("\nInformações sobre rejeição de H0:")*

*print("a) H0 é rejeitada quando o valor-p < nível de significância α")*

*print(f"   No nosso caso, H0 é rejeitada quando o valor-p < {nivel\_significancia}")*

*print("b) H0 é rejeitada quando t\_calculado < t\_crítico (para teste unilateral à esquerda)")*

*print(f"   No nosso caso, H0 é rejeitada quando t\_calculado < {t\_critico:.4f}")*

*# Visualização*

*visualizar\_teste(dados\_novo\_metodo, t\_estatistica, t\_critico)*

*return p\_valor, t\_estatistica, t\_critico*

*# Função para visualizar o teste*

*def visualizar\_teste(dados, t\_estatistica, t\_critico):*

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

*# Definindo o intervalo para plotar a distribuição t*

*x = np.linspace(-4, 4, 1000)*

*df = len(dados) - 1*

*y = stats.t.pdf(x, df)*

*# Plotando a distribuição t*

*plt.plot(x, y, 'b-', lw=2, label='Distribuição t')*

*# Área de rejeição*

*x\_rej = np.linspace(-4, t\_critico, 100)*

*y\_rej = stats.t.pdf(x\_rej, df)*

*plt.fill\_between(x\_rej, y\_rej, color='red', alpha=0.3, label='Região de rejeição')*

*# Marcando o valor t calculado*

*plt.axvline(t\_estatistica, color='green', linestyle='--', lw=2, label='t calculado')*

*plt.axvline(t\_critico, color='red', linestyle='--', lw=2, label='t crítico')*

*plt.title('Teste de Hipóteses para Redução de Custos')*

*plt.xlabel('Valor t')*

*plt.ylabel('Densidade')*

*plt.legend()*

*plt.grid(True)*

*plt.show()*

*# Exemplo de uso:*

*# Vamos simular alguns dados para teste*

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

*# Simulando um novo método com média 210 (menor que o atual) e desvio padrão 15*

*dados\_simulados = np.random.normal(210, 15, 30)*

*# Executando o teste*

*teste\_hipoteses(dados\_simulados)*

*# Para responder às partes específicas da questão:*

*print("\nRespostas específicas para a questão:")*

*print("a) Hipóteses alternativas:")*

*print("   H0: μ >= 220 (o novo método não reduz os custos)")*

*print("   H1: μ < 220 (o novo método reduz os custos)")*

*print("\nb) Conclusão de quando H0 não pode ser rejeitada:")*

*print("   H0 não pode ser rejeitada quando o valor-p ≥ α ou quando t\_calculado ≥ t\_crítico")*

*print("\nc) Conclusão de quando H0 pode ser rejeitada:")*

*print("   H0 pode ser rejeitada quando o valor-p < α ou quando t\_calculado < t\_crítico")*

*#Cap.9:7*

*import numpy as np*

*import scipy.stats as stats*

*import matplotlib.pyplot as plt*

*# Configurações iniciais*

*media\_atual = 8000  # Média atual de vendas: US$ 8 mil por semana por vendedor*

*def analise\_programa\_incentivo(dados\_experimentais, alfa=0.05):*

*"""*

*Realiza análise estatística para determinar se o programa de incentivo aumentou as vendas.*

*Args:*

*dados\_experimentais: Dados de vendas após implementação do programa de incentivo*

*alfa: Nível de significância para o teste*

*"""*

*# Estatísticas descritivas*

*n = len(dados\_experimentais)*

*media\_nova = np.mean(dados\_experimentais)*

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

*# Hipóteses:*

*# H0: μ ≤ 8000 (o programa não aumentou a média de vendas)*

*# H1: μ > 8000 (o programa aumentou a média de vendas)*

*# Cálculo do teste t*

*t\_estatistica = (media\_nova - media\_atual) / (desvio\_padrao / np.sqrt(n))*

*# Valor-p para teste unilateral à direita*

*p\_valor = 1 - stats.t.cdf(t\_estatistica, df=n-1)*

*# Valor crítico*

*t\_critico = stats.t.ppf(1 - alfa, df=n-1)*

*# Resultados*

*print("=== Análise do Programa de Incentivo de Vendas da Carpetland ===")*

*print(f"Média atual de vendas: US$ {media\_atual/1000:.1f} mil por semana")*

*print(f"Média após programa de incentivo: US$ {media\_nova/1000:.2f} mil por semana")*

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

*print(f"Desvio padrão amostral: US$ {desvio\_padrao/1000:.2f} mil")*

*print(f"Estatística t calculada: {t\_estatistica:.4f}")*

*print(f"Valor-p: {p\_valor:.4f}")*

*print(f"Valor crítico t (α = {alfa}): {t\_critico:.4f}")*

*# Conclusão*

*print("\nConclusão do teste:")*

*if p\_valor < alfa:*

*print(f"Rejeitamos H0 com nível de significância {alfa}.")*

*print("Há evidência estatística de que o programa de incentivo aumentou a média de vendas.")*

*else:*

*print(f"Não rejeitamos H0 com nível de significância {alfa}.")*

*print("Não há evidência estatística suficiente de que o programa de incentivo aumentou a média de vendas.")*

*# Análise dos erros*

*print("\n=== Análise dos Erros Tipo I e Tipo II ===")*

*print("a) Hipóteses formuladas:")*

*print("   H0: μ ≤ $8,000 (o programa de incentivo não aumentou a média de vendas)")*

*print("   H1: μ > $8,000 (o programa de incentivo aumentou a média de vendas)")*

*print("\nb) Erro Tipo I nesta situação:")*

*print("   Definição: Rejeitar H0 quando H0 é verdadeira.")*

*print("   Contexto: Concluir que o programa de incentivo aumentou as vendas quando na realidade não aumentou.")*

*print("   Consequências: A empresa implementaria o programa de incentivo acreditando em sua eficácia,")*

*print("   mas estaria pagando mais aos vendedores sem obter o aumento esperado nas vendas.")*

*print("   Isso resultaria em custos adicionais sem o retorno financeiro esperado.")*

*print("\nc) Erro Tipo II nesta situação:")*

*print("   Definição: Não rejeitar H0 quando H0 é falsa.")*

*print("   Contexto: Concluir que o programa de incentivo não aumentou as vendas quando na realidade aumentou.")*

*print("   Consequências: A empresa deixaria de implementar um programa que realmente aumentaria as vendas.")*

*print("   Isso resultaria em perda de oportunidade de aumento nas receitas e potencial vantagem competitiva.")*

*# Visualização do teste*

*visualizar\_teste(t\_estatistica, t\_critico, n-1)*

*return p\_valor, t\_estatistica, t\_critico*

*def visualizar\_teste(t\_estat, t\_crit, df):*

*"""*

*Visualiza a distribuição t com região de rejeição e estatística calculada.*

*"""*

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

*# Intervalo para plotar*

*x = np.linspace(-4, 4, 1000)*

*y = stats.t.pdf(x, df)*

*# Plotando a distribuição t*

*plt.plot(x, y, 'b-', lw=2, label='Distribuição t')*

*# Área de rejeição (cauda direita)*

*x\_rej = np.linspace(t\_crit, 4, 100)*

*y\_rej = stats.t.pdf(x\_rej, df)*

*plt.fill\_between(x\_rej, y\_rej, color='red', alpha=0.3, label='Região de rejeição')*

*# Marcando o valor t calculado e crítico*

*plt.axvline(t\_estat, color='green', linestyle='--', lw=2, label='t calculado')*

*plt.axvline(t\_crit, color='red', linestyle='--', lw=2, label='t crítico')*

*plt.title('Teste de Hipóteses: Programa de Incentivo de Vendas')*

*plt.xlabel('Valor t')*

*plt.ylabel('Densidade')*

*plt.legend()*

*plt.grid(True)*

*plt.show()*

*# Exemplo de uso com dados simulados*

*# Simulando dados de vendas após implementação do programa*

*# Média de $8,500 (aumento de $500) com desvio padrão de $1,000*

*np.random.seed(123)*

*dados\_vendas\_pos\_programa = np.random.normal(8500, 1000, 30)*

*# Executando a análise*

*analise\_programa\_incentivo(dados\_vendas\_pos\_programa)*

*#Cap. 9:9*

*import numpy as np*

*import scipy.stats as stats*

*import matplotlib.pyplot as plt*

*# Dados do problema*

*media\_amostral = 19.4*

*desvio\_padrao\_populacional = 2*

*tamanho\_amostra = 50*

*media\_hipotese = 20*

*nivel\_significancia = 0.05*

*# a) Cálculo da estatística de teste*

*# Para médias com desvio padrão populacional conhecido, usamos a distribuição Z*

*z\_estatistica = (media\_amostral - media\_hipotese) / (desvio\_padrao\_populacional / np.sqrt(tamanho\_amostra))*

*print(f"a) Estatística de teste Z = {z\_estatistica:.4f}")*

*# b) Cálculo do valor-p*

*# Como é um teste unilateral à esquerda (H1: μ < 20), calculamos P(Z ≤ z\_estatistica)*

*p\_valor = stats.norm.cdf(z\_estatistica)*

*print(f"b) Valor-p = {p\_valor:.4f}")*

*# c) Conclusão usando α = 0.05*

*print("\nc) Usando α = 0.05:")*

*if p\_valor < nivel\_significancia:*

*conclusao = "Rejeitamos H0"*

*print(f"Como o valor-p ({p\_valor:.4f}) < α ({nivel\_significancia}), {conclusao}.")*

*print("Há evidência estatística de que a média populacional é menor que 20.")*

*else:*

*conclusao = "Não rejeitamos H0"*

*print(f"Como o valor-p ({p\_valor:.4f}) ≥ α ({nivel\_significancia}), {conclusao}.")*

*print("Não há evidência estatística suficiente para concluir que a média populacional é menor que 20.")*

*# d) Valor crítico e regra de rejeição*

*z\_critico = stats.norm.ppf(nivel\_significancia)*

*print(f"\nd) Valor crítico para α = 0.05: z\_crítico = {z\_critico:.4f}")*

*print("Regra de rejeição: Rejeitar H0 se z\_estatistica < z\_crítico")*

*if z\_estatistica < z\_critico:*

*conclusao\_regra = "Rejeitamos H0"*

*print(f"Como z\_estatistica ({z\_estatistica:.4f}) < z\_crítico ({z\_critico:.4f}), {conclusao\_regra}.")*

*print("Há evidência estatística de que a média populacional é menor que 20.")*

*else:*

*conclusao\_regra = "Não rejeitamos H0"*

*print(f"Como z\_estatistica ({z\_estatistica:.4f}) ≥ z\_crítico ({z\_critico:.4f}), {conclusao\_regra}.")*

*print("Não há evidência estatística suficiente para concluir que a média populacional é menor que 20.")*

*# Visualização do teste*

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

*# Definindo o intervalo para plotar a distribuição normal*

*x = np.linspace(-4, 4, 1000)*

*y = stats.norm.pdf(x)*

*# Plotando a distribuição normal padrão*

*plt.plot(x, y, 'b-', lw=2, label='Distribuição Normal Padrão')*

*# Área de rejeição*

*x\_rej = np.linspace(-4, z\_critico, 100)*

*y\_rej = stats.norm.pdf(x\_rej)*

*plt.fill\_between(x\_rej, y\_rej, color='red', alpha=0.3, label='Região de rejeição')*

*# Marcando o valor z calculado e crítico*

*plt.axvline(z\_estatistica, color='green', linestyle='--', lw=2, label='Z calculado')*

*plt.axvline(z\_critico, color='red', linestyle='--', lw=2, label='Z crítico')*

*plt.title('Teste de Hipóteses: H0: μ ≥ 20 vs H1: μ < 20')*

*plt.xlabel('Valor Z')*

*plt.ylabel('Densidade')*

*plt.legend()*

*plt.grid(True)*

*plt.show()*

*# Resumo das respostas*

*print("\nRespostas:")*

*print(f"a) Estatística de teste: Z = {z\_estatistica:.4f}")*

*print(f"b) Valor-p = {p\_valor:.4f}")*

*print(f"c) Conclusão com α = 0.05: {conclusao}. {'Rejeitamos a hipótese nula.' if p\_valor < nivel\_significancia else 'Não rejeitamos a hipótese nula.'}")*

*print(f"d) Regra de rejeição: Rejeitar H0 se Z < {z\_critico:.4f}")*

*print(f"   Conclusão baseada na regra: {conclusao\_regra}")*

*#Cap. 9:14*

*import numpy as np*

*import scipy.stats as stats*

*import matplotlib.pyplot as plt*

*# Dados do problema*

*media\_hipotese = 22*

*desvio\_padrao\_populacional = 10*

*tamanho\_amostra = 75*

*nivel\_significancia = 0.01*

*# Médias amostrais a serem testadas*

*medias\_amostrais = [23, 25.1, 20]*

*def teste\_hipotese\_bilateral(media\_amostral):*

*"""*

*Realiza teste de hipótese bilateral:*

*H0: μ = 22*

*H1: μ ≠ 22*

*"""*

*# Cálculo da estatística de teste Z*

*z\_estatistica = (media\_amostral - media\_hipotese) / (desvio\_padrao\_populacional / np.sqrt(tamanho\_amostra))*

*# Cálculo do valor-p para teste bilateral*

*# Para teste bilateral: valor-p = 2 \* min(P(Z ≤ z), P(Z ≥ z))*

*if z\_estatistica < 0:*

*p\_valor = 2 \* stats.norm.cdf(z\_estatistica)*

*else:*

*p\_valor = 2 \* (1 - stats.norm.cdf(z\_estatistica))*

*# Decisão usando o valor-p*

*if p\_valor < nivel\_significancia:*

*decisao = "Rejeitamos H0"*

*conclusao = f"Há evidência estatística de que a média populacional é diferente de {media\_hipotese}."*

*else:*

*decisao = "Não rejeitamos H0"*

*conclusao = f"Não há evidência estatística suficiente para concluir que a média populacional é diferente de {media\_hipotese}."*

*return z\_estatistica, p\_valor, decisao, conclusao*

*# Calcular valor crítico para α = 0.01 (teste bilateral)*

*z\_critico = stats.norm.ppf(1 - nivel\_significancia/2)  # Para teste bilateral dividimos alfa por 2*

*# Aplicar o teste para cada média amostral*

*resultados = []*

*for i, media in enumerate(medias\_amostrais, start=1):*

*letra = chr(96 + i)  # 'a', 'b', 'c'*

*z, p, decisao, conclusao = teste\_hipotese\_bilateral(media)*

*resultados.append((letra, media, z, p, decisao, conclusao))*

*print(f"\nCaso {letra}. x̄ = {media}")*

*print(f"Estatística Z = {z:.4f}")*

*print(f"Valor-p = {p:.4f}")*

*print(f"Para α = {nivel\_significancia}, {decisao}.")*

*print(conclusao)*

*# Visualização gráfica dos resultados*

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

*# Definindo o intervalo para plotar a distribuição normal*

*x = np.linspace(-4, 4, 1000)*

*y = stats.norm.pdf(x)*

*# Plotando a distribuição normal padrão*

*plt.plot(x, y, 'b-', lw=2, label='Distribuição Normal Padrão')*

*# Áreas de rejeição*

*x\_rej\_esq = np.linspace(-4, -z\_critico, 100)*

*y\_rej\_esq = stats.norm.pdf(x\_rej\_esq)*

*plt.fill\_between(x\_rej\_esq, y\_rej\_esq, color='red', alpha=0.3)*

*x\_rej\_dir = np.linspace(z\_critico, 4, 100)*

*y\_rej\_dir = stats.norm.pdf(x\_rej\_dir)*

*plt.fill\_between(x\_rej\_dir, y\_rej\_dir, color='red', alpha=0.3, label='Regiões de rejeição')*

*# Marcando os valores críticos*

*plt.axvline(-z\_critico, color='red', linestyle='--', lw=2, label=f'Z crítico = ±{z\_critico:.4f}')*

*plt.axvline(z\_critico, color='red', linestyle='--', lw=2)*

*# Marcando os valores Z calculados*

*cores = ['green', 'purple', 'orange']*

*for i, (letra, media, z, p, \_, \_) in enumerate(resultados):*

*plt.axvline(z, color=cores[i], linestyle='--', lw=2, label=f'Z para caso {letra} = {z:.4f}')*

*plt.title('Teste de Hipóteses Bilateral: H0: μ = 22 vs H1: μ ≠ 22')*

*plt.xlabel('Valor Z')*

*plt.ylabel('Densidade')*

*plt.legend()*

*plt.grid(True)*

*plt.show()*

*# Tabela resumo dos resultados*

*print("\nResumo dos resultados:")*

*print("-" \* 80)*

*print(f"{'Caso':<5}{'Média':<10}{'Estatística Z':<15}{'Valor-p':<15}{'Decisão':<20}{'Conclusão'}")*

*print("-" \* 80)*

*for letra, media, z, p, decisao, conclusao in resultados:*

*print(f"{letra}. {media:<10}{z:<15.4f}{p:<15.4f}{decisao:<20}{conclusao}")*

*print("-" \* 80)*

*print(f"Valor crítico para α = {nivel\_significancia}: Z = ±{z\_critico:.4f}")*

*print(f"Regra de rejeição: Rejeitar H0 se |Z| > {z\_critico:.4f}")*

*#Cap 9:17*

*import numpy as np*

*import scipy.stats as stats*

*import matplotlib.pyplot as plt*

*# Dados do problema*

*salario\_medio\_reportado = 24.57  # Salário médio reportado para indústrias produtoras de bens (US$/hora)*

*media\_amostral = 23.89  # Média amostral da indústria manufatureira (US$/hora)*

*desvio\_padrao\_populacional = 2.40  # Desvio padrão populacional (US$/hora)*

*tamanho\_amostra = 30  # Tamanho da amostra*

*nivel\_significancia = 0.05  # Nível de significância para o teste*

*def teste\_hipoteses\_salario():*

*"""*

*Realiza teste de hipóteses para verificar se o salário médio na indústria*

*manufatureira difere do salário médio reportado para indústrias produtoras de bens.*

*"""*

*# a) Declaração das hipóteses*

*print("a) Hipóteses para o teste:")*

*print("   H0: μ = $24.57 (salário médio na indústria manufatureira é igual ao reportado)")*

*print("   H1: μ ≠ $24.57 (salário médio na indústria manufatureira é diferente do reportado)")*

*print("\n   Justificativa: Como queremos verificar se o salário difere (podendo ser maior ou menor),")*

*print("   usamos um teste bilateral.\n")*

*# b) Cálculo da estatística de teste Z e valor-p*

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

*z\_estatistica = (media\_amostral - salario\_medio\_reportado) / erro\_padrao*

*# Cálculo do valor-p para teste bilateral*

*if z\_estatistica < 0:*

*p\_valor = 2 \* stats.norm.cdf(z\_estatistica)*

*else:*

*p\_valor = 2 \* (1 - stats.norm.cdf(z\_estatistica))*

*print(f"b) Estatística de teste: Z = {z\_estatistica:.4f}")*

*print(f"   Valor-p = {p\_valor:.4f}")*

*# c) Conclusão usando α = 0.05*

*print("\nc) Usando α = 0.05:")*

*if p\_valor < nivel\_significancia:*

*conclusao = "Rejeitamos H0"*

*explicacao = "Há evidência estatística suficiente para concluir que o salário médio na indústria manufatureira é diferente de $24.57 por hora."*

*else:*

*conclusao = "Não rejeitamos H0"*

*explicacao = "Não há evidência estatística suficiente para concluir que o salário médio na indústria manufatureira é diferente de $24.57 por hora."*

*print(f"   Como valor-p ({p\_valor:.4f}) {'<' if p\_valor < nivel\_significancia else '≥'} α ({nivel\_significancia}), {conclusao}.")*

*print(f"   {explicacao}")*

*# d) Teste usando o critério do valor crítico*

*z\_critico = stats.norm.ppf(1 - nivel\_significancia/2)  # Para teste bilateral*

*print(f"\nd) Usando o critério do valor crítico:")*

*print(f"   Valor crítico para α = 0.05 (teste bilateral): Z = ±{z\_critico:.4f}")*

*print(f"   Regra de rejeição: Rejeitar H0 se |Z| > {z\_critico:.4f}")*

*if abs(z\_estatistica) > z\_critico:*

*conclusao\_critico = "Rejeitamos H0"*

*explicacao\_critico = "Há evidência estatística suficiente para concluir que o salário médio na indústria manufatureira é diferente de $24.57 por hora."*

*else:*

*conclusao\_critico = "Não rejeitamos H0"*

*explicacao\_critico = "Não há evidência estatística suficiente para concluir que o salário médio na indústria manufatureira é diferente de $24.57 por hora."*

*print(f"   Como |Z| ({abs(z\_estatistica):.4f}) {'>' if abs(z\_estatistica) > z\_critico else '≤'} {z\_critico:.4f}, {conclusao\_critico}.")*

*print(f"   {explicacao\_critico}")*

*# Retornando os resultados para visualização*

*return z\_estatistica, p\_valor, z\_critico, conclusao*

*def visualizar\_teste(z\_estatistica, z\_critico):*

*"""*

*Cria uma visualização gráfica do teste de hipóteses.*

*"""*

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

*# Definindo o intervalo para plotar a distribuição normal*

*x = np.linspace(-4, 4, 1000)*

*y = stats.norm.pdf(x)*

*# Plotando a distribuição normal padrão*

*plt.plot(x, y, 'b-', lw=2, label='Distribuição Normal Padrão')*

*# Áreas de rejeição (bilateral)*

*x\_rej\_esq = np.linspace(-4, -z\_critico, 100)*

*y\_rej\_esq = stats.norm.pdf(x\_rej\_esq)*

*plt.fill\_between(x\_rej\_esq, y\_rej\_esq, color='red', alpha=0.3)*

*x\_rej\_dir = np.linspace(z\_critico, 4, 100)*

*y\_rej\_dir = stats.norm.pdf(x\_rej\_dir)*

*plt.fill\_between(x\_rej\_dir, y\_rej\_dir, color='red', alpha=0.3, label='Regiões de rejeição (α=0.05)')*

*# Marcando os valores críticos*

*plt.axvline(-z\_critico, color='red', linestyle='--', lw=2, label=f'Valores críticos (±{z\_critico:.4f})')*

*plt.axvline(z\_critico, color='red', linestyle='--', lw=2)*

*# Marcando o valor Z calculado*

*plt.axvline(z\_estatistica, color='green', linestyle='--', lw=2, label=f'Z calculado ({z\_estatistica:.4f})')*

*plt.title('Teste de Hipóteses: Salário Médio na Indústria Manufatureira')*

*plt.xlabel('Valor Z')*

*plt.ylabel('Densidade')*

*plt.legend()*

*plt.grid(True)*

*# Adicionando informações adicionais no gráfico*

*info\_text = (*

*f"Salário reportado: ${salario\_medio\_reportado}/h\n"*

*f"Média amostral: ${media\_amostral}/h\n"*

*f"Tamanho da amostra: {tamanho\_amostra}\n"*

*f"Desvio padrão: ${desvio\_padrao\_populacional}/h\n"*

*f"α = {nivel\_significancia}"*

*)*

*plt.text(2.5, 0.3, info\_text, bbox=dict(facecolor='white', alpha=0.8))*

*plt.tight\_layout()*

*plt.show()*

*# Executar o teste de hipóteses*

*z\_estatistica, p\_valor, z\_critico, conclusao = teste\_hipoteses\_salario()*

*# Criar visualização gráfica*

*visualizar\_teste(z\_estatistica, z\_critico)*

*#Cap.9:24*

*import numpy as np*

*import scipy.stats as stats*

*import matplotlib.pyplot as plt*

*# Dados do problema*

*media\_amostral = 17*

*desvio\_padrao\_amostral = 4.5*

*tamanho\_amostra = 48*

*media\_hipotese = 18*

*nivel\_significancia = 0.05*

*# a) Cálculo da estatística de teste t*

*t\_estatistica = (media\_amostral - media\_hipotese) / (desvio\_padrao\_amostral / np.sqrt(tamanho\_amostra))*

*print(f"a) Estatística de teste t = {t\_estatistica:.4f}")*

*# b) Calculando o valor-p usando a tabela de distribuição t*

*# Graus de liberdade = n - 1*

*graus\_liberdade = tamanho\_amostra - 1*

*# Para teste bilateral, o valor-p é:*

*if t\_estatistica < 0:*

*p\_valor = 2 \* stats.t.cdf(t\_estatistica, df=graus\_liberdade)*

*else:*

*p\_valor = 2 \* (1 - stats.t.cdf(t\_estatistica, df=graus\_liberdade))*

*print(f"b) Valor-p = {p\_valor:.4f}")*

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

*# c) Conclusão usando α = 0.05*

*print("\nc) Usando α = 0.05:")*

*if p\_valor < nivel\_significancia:*

*conclusao = "Rejeitamos H0"*

*explicacao = "Há evidência estatística suficiente para concluir que a média populacional é diferente de 18."*

*else:*

*conclusao = "Não rejeitamos H0"*

*explicacao = "Não há evidência estatística suficiente para concluir que a média populacional é diferente de 18."*

*print(f"   Como valor-p ({p\_valor:.4f}) {'<' if p\_valor < nivel\_significancia else '≥'} α ({nivel\_significancia}), {conclusao}.")*

*print(f"   {explicacao}")*

*# d) Valor crítico e regra de rejeição*

*t\_critico = stats.t.ppf(1 - nivel\_significancia/2, df=graus\_liberdade)*

*print(f"\nd) Valor crítico para α = 0.05 e {graus\_liberdade} graus de liberdade: t = ±{t\_critico:.4f}")*

*print(f"   Regra de rejeição: Rejeitar H0 se |t| > {t\_critico:.4f}")*

*if abs(t\_estatistica) > t\_critico:*

*conclusao\_criterio = "Rejeitamos H0"*

*explicacao\_criterio = "Há evidência estatística suficiente para concluir que a média populacional é diferente de 18."*

*else:*

*conclusao\_criterio = "Não rejeitamos H0"*

*explicacao\_criterio = "Não há evidência estatística suficiente para concluir que a média populacional é diferente de 18."*

*print(f"   Como |t| ({abs(t\_estatistica):.4f}) {'>' if abs(t\_estatistica) > t\_critico else '≤'} {t\_critico:.4f}, {conclusao\_criterio}.")*

*print(f"   {explicacao\_criterio}")*

*# Visualização gráfica do teste*

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

*# Definindo o intervalo para plotar a distribuição t*

*x = np.linspace(-4, 4, 1000)*

*y = stats.t.pdf(x, df=graus\_liberdade)*

*# Plotando a distribuição t*

*plt.plot(x, y, 'b-', lw=2, label=f'Distribuição t ({graus\_liberdade} graus de liberdade)')*

*# Áreas de rejeição*

*x\_rej\_esq = np.linspace(-4, -t\_critico, 100)*

*y\_rej\_esq = stats.t.pdf(x\_rej\_esq, df=graus\_liberdade)*

*plt.fill\_between(x\_rej\_esq, y\_rej\_esq, color='red', alpha=0.3)*

*x\_rej\_dir = np.linspace(t\_critico, 4, 100)*

*y\_rej\_dir = stats.t.pdf(x\_rej\_dir, df=graus\_liberdade)*

*plt.fill\_between(x\_rej\_dir, y\_rej\_dir, color='red', alpha=0.3, label='Regiões de rejeição (α=0.05)')*

*# Marcando os valores críticos*

*plt.axvline(-t\_critico, color='red', linestyle='--', lw=2, label=f'Valores críticos (±{t\_critico:.4f})')*

*plt.axvline(t\_critico, color='red', linestyle='--', lw=2)*

*# Marcando o valor t calculado*

*plt.axvline(t\_estatistica, color='green', linestyle='--', lw=2, label=f't calculado ({t\_estatistica:.4f})')*

*plt.title('Teste de Hipóteses: H0: μ = 18 vs H1: μ ≠ 18')*

*plt.xlabel('Valor t')*

*plt.ylabel('Densidade')*

*plt.legend()*

*plt.grid(True)*

*# Adicionando informações adicionais no gráfico*

*info\_text = (*

*f"Média amostral: {media\_amostral}\n"*

*f"Desvio padrão amostral: {desvio\_padrao\_amostral}\n"*

*f"Tamanho da amostra: {tamanho\_amostra}\n"*

*f"Graus de liberdade: {graus\_liberdade}\n"*

*f"α = {nivel\_significancia}"*

*)*

*plt.text(2.5, 0.3, info\_text, bbox=dict(facecolor='white', alpha=0.8))*

*plt.tight\_layout()*

*plt.show()*

*# Resumo final com todos os resultados*

*print("\nResumo do teste de hipóteses:")*

*print("-" \* 70)*

*print(f"Hipótese nula (H0): μ = {media\_hipotese}")*

*print(f"Hipótese alternativa (H1): μ ≠ {media\_hipotese}")*

*print(f"Média amostral: {media\_amostral}")*

*print(f"Desvio padrão amostral: {desvio\_padrao\_amostral}")*

*print(f"Tamanho da amostra: {tamanho\_amostra}")*

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

*print(f"Estatística t: {t\_estatistica:.4f}")*

*print(f"Valor-p: {p\_valor:.4f}")*

*print(f"Valor crítico (α = {nivel\_significancia}): ±{t\_critico:.4f}")*

*print(f"Conclusão: {conclusao}")*

*print("-" \* 70)*

*#Cap. 9:32*

*import numpy as np*

*import scipy.stats as stats*

*import pandas as pd*

*import matplotlib.pyplot as plt*

*# Dados do problema*

*preco\_medio\_nacional = 10192  # Preço médio nacional de carros usados (US$)*

*tamanho\_amostra = 50  # Número de carros na amostra da revendedora*

*# Simulando o arquivo UsedCars com dados fictícios (apenas para demonstração)*

*# Em um caso real, estes dados seriam lidos de um arquivo real*

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

*# Gerando preços com média um pouco diferente da nacional para ilustração*

*precos\_amostra = np.random.normal(loc=9800, scale=1200, size=tamanho\_amostra)*

*precos\_amostra = np.round(precos\_amostra, 2)  # Arredondando para 2 casas decimais*

*# Criando um DataFrame como se fosse o arquivo UsedCars*

*df = pd.DataFrame({'Preco': precos\_amostra})*

*# Calculando estatísticas da amostra*

*media\_amostral = df['Preco'].mean()*

*desvio\_padrao\_amostral = df['Preco'].std(ddof=1)  # ddof=1 para desvio padrão amostral*

*def teste\_hipoteses\_carros\_usados():*

*"""*

*Realiza teste de hipóteses para verificar se o preço médio dos carros usados*

*na revendedora de Kansas City difere da média nacional.*

*"""*

*print("Análise de Preços de Carros Usados - Kansas City vs. Média Nacional")*

*print("=" \* 70)*

*# a) Formulação das hipóteses*

*print("a) Hipóteses para o teste:")*

*print("   H0: μ = $10,192 (o preço médio na revendedora é igual à média nacional)")*

*print("   H1: μ ≠ $10,192 (o preço médio na revendedora é diferente da média nacional)")*

*print("\n   Justificativa: Como queremos determinar se existe uma diferença no preço")*

*print("   médio (podendo ser maior ou menor), usamos um teste bilateral.\n")*

*# b) Cálculo da estatística de teste t e valor-p*

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

*t\_estatistica = (media\_amostral - preco\_medio\_nacional) / erro\_padrao*

*# Graus de liberdade*

*gl = tamanho\_amostra - 1*

*# Cálculo do valor-p para teste bilateral*

*if t\_estatistica < 0:*

*p\_valor = 2 \* stats.t.cdf(t\_estatistica, df=gl)*

*else:*

*p\_valor = 2 \* (1 - stats.t.cdf(t\_estatistica, df=gl))*

*print(f"b) Estatística de teste: t = {t\_estatistica:.4f}")*

*print(f"   Graus de liberdade: {gl}")*

*print(f"   Valor-p = {p\_valor:.4f}")*

*# c) Conclusão usando α = 0.05*

*alpha = 0.05*

*print(f"\nc) Com α = {alpha}:")*

*if p\_valor < alpha:*

*conclusao = "Rejeitamos H0"*

*explicacao = "Há evidência estatística suficiente para concluir que o preço médio dos carros usados na revendedora de Kansas City é diferente da média nacional de $10,192."*

*else:*

*conclusao = "Não rejeitamos H0"*

*explicacao = "Não há evidência estatística suficiente para concluir que o preço médio dos carros usados na revendedora de Kansas City é diferente da média nacional de $10,192."*

*print(f"   Como valor-p ({p\_valor:.4f}) {'<' if p\_valor < alpha else '≥'} α ({alpha}), {conclusao}.")*

*print(f"   {explicacao}")*

*# Valor crítico e região de rejeição*

*t\_critico = stats.t.ppf(1 - alpha/2, df=gl)*

*print(f"\n   Valor crítico para α = {alpha} (teste bilateral): t = ±{t\_critico:.4f}")*

*print(f"   Regra de rejeição: Rejeitar H0 se |t| > {t\_critico:.4f}")*

*if abs(t\_estatistica) > t\_critico:*

*conclusao\_criterio = "rejeitamos H0"*

*else:*

*conclusao\_criterio = "não rejeitamos H0"*

*print(f"   Como |t| ({abs(t\_estatistica):.4f}) {'>' if abs(t\_estatistica) > t\_critico else '≤'} {t\_critico:.4f}, {conclusao\_criterio}.")*

*return t\_estatistica, p\_valor, gl, conclusao*

*def visualizar\_resultados(t\_estatistica, gl):*

*"""*

*Cria visualizações para melhor compreensão dos resultados.*

*"""*

*plt.figure(figsize=(14, 8))*

*# Gráfico 1: Distribuição dos preços da amostra*

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

*plt.hist(df['Preco'], bins=15, alpha=0.7, color='skyblue', edgecolor='black')*

*plt.axvline(media\_amostral, color='red', linestyle='--', linewidth=2, label=f'Média Amostral: ${media\_amostral:.2f}')*

*plt.axvline(preco\_medio\_nacional, color='green', linestyle='-', linewidth=2, label=f'Média Nacional: ${preco\_medio\_nacional}')*

*plt.title('Distribuição dos Preços dos Carros Usados na Amostra')*

*plt.xlabel('Preço (US$)')*

*plt.ylabel('Frequência')*

*plt.legend()*

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

*# Gráfico 2: Visualização do teste t*

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

*# Definindo o intervalo para plotar a distribuição t*

*x = np.linspace(-4, 4, 1000)*

*y = stats.t.pdf(x, df=gl)*

*# Plotando a distribuição t*

*plt.plot(x, y, 'b-', lw=2, label=f'Distribuição t ({gl} graus de liberdade)')*

*# Valor crítico para α = 0.05 bilateral*

*t\_critico = stats.t.ppf(0.975, df=gl)*

*# Áreas de rejeição*

*x\_rej\_esq = np.linspace(-4, -t\_critico, 100)*

*y\_rej\_esq = stats.t.pdf(x\_rej\_esq, df=gl)*

*plt.fill\_between(x\_rej\_esq, y\_rej\_esq, color='red', alpha=0.3)*

*x\_rej\_dir = np.linspace(t\_critico, 4, 100)*

*y\_rej\_dir = stats.t.pdf(x\_rej\_dir, df=gl)*

*plt.fill\_between(x\_rej\_dir, y\_rej\_dir, color='red', alpha=0.3, label='Regiões de rejeição (α=0.05)')*

*# Marcando os valores críticos*

*plt.axvline(-t\_critico, color='red', linestyle='--', lw=2, label=f'Valores críticos (±{t\_critico:.4f})')*

*plt.axvline(t\_critico, color='red', linestyle='--', lw=2)*

*# Marcando o valor t calculado*

*plt.axvline(t\_estatistica, color='green', linestyle='--', lw=2, label=f't calculado ({t\_estatistica:.4f})')*

*plt.title('Teste de Hipóteses: Preço Médio dos Carros Usados')*

*plt.xlabel('Valor t')*

*plt.ylabel('Densidade')*

*plt.legend()*

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

*plt.tight\_layout()*

*plt.show()*

*# Executar o teste de hipóteses*

*t\_estatistica, p\_valor, gl, conclusao = teste\_hipoteses\_carros\_usados()*

*# Visualizar os resultados*

*visualizar\_resultados(t\_estatistica, gl)*

*# Exibir resumo final*

*print("\nResumo dos dados:")*

*print(f"Média nacional: ${preco\_medio\_nacional}")*

*print(f"Média amostral na revendedora: ${media\_amostral:.2f}")*

*print(f"Desvio padrão amostral: ${desvio\_padrao\_amostral:.2f}")*

*print(f"Tamanho da amostra: {tamanho\_amostra}")*