Teste de Hipóteses

1. Introdução

O **Teste de Hipóteses** é a ferramenta base da inferência estatística, onde pode-se tirar informações a partir de uma dada amostra e extrapolar (ou inferir) conclusões sobre a população. Uma hipótese será uma declaração sobre um parâmetro da população, ou seja para dado μ, σ^2 e p da população usam-se estimadores como $\overline{X}$, S^2 e $\hat{p}$ para as amostras. Os cálculos desses estimadores são dados da seguinte forma:

$$\overline{X} = \sum_{i=1}^{n} \frac{X_i}{n}$$

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

$$\hat{p} = \frac{1}{n} \sum_{i=1}^{n} I_i$$

No caso para a proporção $\hat{p}$, o valor de I_i é igual a 1 se a observação tem a característica de interesse, ou seja $I_i \in \{0, 1\}$

Ao aplicar um teste de hipóteses o objetivo é testar alguma informação relevante sobre um atributo da população, de maneira a aceitar ou rejeitar esta informação. Para isto, é que formulam-se as chamadas **hipótese nula** (H_0) e a **hipótese alternativa** (H_a ou H_1), sendo a hipótese alternativa o complemento/negação da hipótese nula.

Ao testar essas hipóteses pode-se cair nos seguintes casos:

Hipóteses	**H_0 é verdadeira**	**H_0 é falsa**
Rejeitar H_0	Erro tipo I (α)	Sem erro
Aceitar H_0	Sem erro	Erro tipo II (β)

Usualmente aplicamos teste de hipótese para diminuir a chance de ocorrer o erro do Tipo **I**, ou seja, costuma-se trabalhar com o nível de significância α.

2. Aplicações de Teste de Hipóteses

Neste tópico, vão ser desenvolvidos alguns exemplo de aplicação de teste de hipóteses para tirar conclusões a respeito de amostras de dados populacional:

Exemplo 1: Uma fábrica anuncia que o índice de nicotina dos cigarros de uma dada marca é igual ou menor a 20 mg por cigarro. Um laboratório realiza 20 análises do índice obtendo: 22, 19, 21, 22, 20, 18, 27, 20, 21, 19, 20, 22, 17, 20, 21,18, 25, 16, 20, 21. Sabe-se que o índice de nicotina dos cigarros dessa marca se distribui normalmente com variância 4 mg2. Pode-se aceitar a afirmação do fabricante, ao nível de 5%?

$$H_0 : \mu \leq 20$$
$$H_1 : \mu > 20$$

Primeiramente, deve-se calcular o valor crítico (ou seja, o valor limítrofe para a avaliação do teste) para assim definir se o teste pode aceitar ou rejeitar H_0:

$$P\left(Z < \frac{\overline{X}_c - \mu}{\frac{\sigma}{\sqrt{n}}} | \mu = 20\right) = 1 - \alpha = 0.95$$

Implementando a avaliação do teste teórico utilizando o *Python*:

```
# Cálculo do Z crítico para a probabilidade de 95%
prob = st.norm.ppf(0.95)

# Mostra o valor da probabilidade para 95%
print("Z crítico para a probabilidade para 95%: ", prob)

# Parâmetros para o teste
mu = 20            # média
sigma = np.sqrt(4) # desvio padrão
n = 20             # tamanho da amostra

# Cálculo do valor crítico
Xc = mu + (sigma/np.sqrt(n))*prob

# Mostra o valor crítico
print("Valor crítico: ", Xc, " mg")

# Aplicando as observações
X = [22, 19, 21, 22, 20, 18, 27, 20, 21, 19, 20, 22, 17, 20, 21,18, 25, 16, 20, 21]

# Calculando a média das observações
Xobs = np.mean(X)

# Mostra o valor médio das observações
print("Valor médio observado: ", Xobs, " mg")

# Verificando se o valor médio ultrapassa o valor crítico
print("Teste de Hipótese: ")
if(Xobs < Xc):
    print("Aceitamos H0")
else:
    print("Rejeitamos H0")
```

O mesmo exercício pode ser resolvido pelo método de simulações (*Bootstrapping*) com a implementação em *Python*:

```
# Parâmetros da amostra
mu = 20    # média
sigma = 2  # desvio padrão
n = 20     # tamanho da amostra
```

```
# Número de simulações
Ns = 1000

# Vetor para salvar as médias amostrais
Xm = []

# Laço para gerar as médias
for s in range(1, Ns):
    x = np.random.normal(mu, sigma, n) # sorteia uma amostra de tamanho n
    Xm.append(np.mean(x))              # armazena a média da amostra sorteada

# Valores observados
X = [22, 19, 21, 22, 20, 18, 27, 20, 21, 19, 20, 22, 17, 20, 21,18, 25, 16, 20, 21]

# Média das observações
xobs = np.mean(X)

# Calcula o valor crítico e avalia o teste de hipótese
alpha = 95
xc = np.percentile(Xm, alpha)
print('Xc=',xc,'  Xobs = ', xobs)
if(xobs < xc):
    print("Aceitamos H0")
else:
    print("Rejeitamos H0")

# Cria uma gráfico da distribuição das simulações e compara o valor crítico e observado
plt.figure(figsize=(8,4))
a = plt.hist(x=Xm, bins=20, color='#0504aa', alpha=0.7, rwidth=0.85, density=True)
plt.axvline(x=xc, color='red', linestyle='--', label = 'xc1')
plt.axvline(x=xobs, color='green', linestyle='--', label = 'xobs')
plt.xlabel(r'$\bar{X}$', fontsize=20)
plt.ylabel(r'$P(\bar{X})$', fontsize=20)
plt.legend()
plt.show()
```

Exemplo 2 : Um pesquisador deseja estudar o efeito de certa substância no tempo de reação de seres vivos a um certo tipo de estímulo. Um experimento é desenvolvido com cobaias, que são inoculadas com a substância e submetidas a um estímulo elétrico, com seus tempos de reação (em segundos) anotados. Os seguintes valores foram obtidos: T = [9.1, 9.3, 7.2, 13.3, 10.9, 7.2, 9.9, 8.0, 8.6, 7.5]

Admite-se que, em geral, o tempo de reação tem distribuição Normal com média 8 segundos e desvio padrão 2 segundos. Entretanto, o pesquisador desconfia que o tempo médio sofra alteração por influência da substância. Verifique, a nível 6%, se o tempo de reação das cobaias submetidas à substância foi alterado.

$$H_0 : \mu = 8$$
$$H_1 : \mu \neq 8$$

Inicialmente deve-se definir definir a região crítica para o teste de hipótese:

$\alpha = \mathrm{P}(\text{rejeitar } H_0 \mid H_0$ é verdadeira) = 0.06, ou seja: $P\left(\overline{X} < \overline{X}_{c1} \cup \overline{X} > \overline{X}_{c2} | \mu = 8\right) = 0.06$

Para o caso do teste bicaudal, temos que a probabilidade para cada um dos casos será igual a $\alpha/2$:

$$P\left(\overline{X} < \overline{X}_{c1}|\mu = 8\right) = 0.03 \quad P\left(\overline{X} > \overline{X}_{c2}|\mu = 8\right) = 0.03$$

Por fim ajustando a probabilidade para o caso crítico 2:

$$P\left(\overline{X} < \overline{X}_{c1}|\mu = 8\right) = 0.03 \quad P\left(\overline{X} < \overline{X}_{c2}|\mu = 8\right) = 0.97$$

Implementando o teste de hipóteses utilizando o *Python*:

```
# Define o nível de significância
alpha = 0.06

# Definindo o caso crítico 1
print("Caso crítico 1")

# Cálculo do Z crítico para o caso crítico 1
Zc1 = st.norm.ppf(alpha/2)
print("Z crítico para o caso crítico 1: ", Zc1)

# Parâmetros do enunciado
mu = 8     # média
sigma = 2  # desvio padrão
n = 10     # tamanho da amostra

# Cálculo do valor crítico 1
Xc1 = mu + (sigma/np.sqrt(n))*Zc1
print("Valor crítico 1: ", Xc1, " s")
print("==================================================")

# Definindo o caso crítico 2
print("Caso crítico 2")

# Cálculo do Z crítico para o caso crítico 2
Zc2 = st.norm.ppf(1 - alpha/2)
print("Z crítico para o caso crítico 2: ", Zc2)

#  Cálculo do valor crítico 2
Xc2 = mu + (sigma/np.sqrt(n))*Zc2
print("Valor crítico 2: ", Xc2, " s")
print("==================================================")

# Valores observados
X = [9.1, 9.3, 7.2, 13.3, 10.9, 7.2, 9.9, 8.0, 8.6, 7.5]

# Valor médio das observações
Xobs = np.mean(X)
print("Valor médio observado: ", Xobs, " s")

# Aplicando o teste de hipótese
print("Teste de Hipótese: ")
if(Xobs < Xc1 or Xobs > Xc2):
    print("Rejeitamos H0")
```

```
else:
    print("Aceitamos H0")
```

Fazendo o mesmo processo de resolução do exemplo agora com o método de simulações:

```
# Parâmetros
mu = 8     # média
sigma = 2  # desvio padrão
n = 10     # tamanho da amostra

# Número de simulações
Ns = 10000

# Vetor para as médias amostrais
Xm = []

# Laço para as simulações
for s in range(1,Ns):
    x = np.random.normal(mu, sigma, n) # sorteia uma amostra de tamanho n
    Xm.append(np.mean(x))              # salva o valor das médias amostrais

# Observações
X = [9.1,9.3,7.2,13.3,10.9,7.2,9.9,8.0,8.6,7.5]

# Média das observações
xobs = np.mean(X)

# Aplicando o teste de hipótese a um nível de significância 3%
alpha = 3
xc1 = np.percentile(Xm, alpha)
xc2 = np.percentile(Xm, 100-alpha)
print('Xc1=',xc1, '  Xc2=', xc2, '  Xobs = ', xobs)
if(xobs < xc1 or xobs > xc2):
    print("Rejeitamos H0")
else:
    print("Aceitamos H0")

# Gera uma figura com a distribuição, os limites e média dos valores observados
plt.figure(figsize=(8,4))
a = plt.hist(x=Xm, bins=20, color='#0504aa', alpha=0.7, rwidth=0.85, density=True)
plt.axvline(x=xc1, color='red', linestyle='--', label = 'xc1')
plt.axvline(x=xc2, color='orange', linestyle='--', label = 'xc2')
plt.axvline(x=xobs, color='green', linestyle='--', label = 'xobs')
plt.xlabel(r'$\bar{X}$', fontsize=20)
plt.ylabel(r'$P(\bar{X})$', fontsize=20)
plt.legend()
plt.show()
```

3. Valor p (*p-value*)

Nos casos anteriores, para realizar o teste foi fixado a probabilidade do erro tipo I, dessa forma a aceitação ou não de H_0 depende do valor de α.

Uma outra forma de realizar o teste seria calculando a **probabilidade de significância** (nível descritivo) mais conhecido como **valor p** (*p-value*). A principal diferença com relação aos casos anteriores consiste em não construir a região crítica, o que se faz é indicar a probabilidade de ocorrer valores de estatística mais extremos do que o observado caso a hipótese nula seja verdadeira.

Um forma de avaliar o teste de acordo com diferentes valores p, seria gerando uma curva entre os valores α e o respectivo valor crítico e de acordo com a média das observações pode-se definir uma região onde tenho valores de significância que aceitem a hipótese H_0 e uma região que rejeita.

Como forma de exemplo, será gerada a curva dos valores α a partir do uma distribuição normal com $\mu = 10$ e $\sigma = 2$, onde serão feitas simulações com um tamanho de amostra de 50 observações:

```
# Parâmetros
mu = 10    # média
sigma = 2  # desvio padrão
n = 50     # tamanho das amostras

# Número de simulações
Ns = 10000

# Vetor com as médias amostrais
Xm = []

# Laço para as simulações
for s in range(1,Ns):
    x = np.random.normal(mu, sigma, n) # sorteia uma amostra de tamanho n
    Xm.append(np.mean(x))

# Gera um histograma das simulações
plt.figure(figsize=(8,4))
a = plt.hist(x=Xm, bins=20, color='#0504aa', alpha=0.7, rwidth=0.85, label = str(Ns), density=True)
plt.axvline(x=mu, color='r', linestyle='--', label = 'Média')
plt.xlabel(r'$\bar{X}$', fontsize=20)
plt.ylabel(r'$P(\bar{X})$', fontsize=20)
plt.show()
```

Definido o vetor de médias amostrais e supondo que é conhecido a média das observações $\bar{X_{obs}} = 9,8$, próximo passo seria calcular o valor crítico para valores de α variando entre 1 e 100:

```
# Parâmetros para gerar a curva
xobs = 9.8     # média das observações
xcs = []       # vetor de valores críticos
alphas = []    # vetor de valores de alpha

# Laço para determinar os valores críticos por alphas
for alpha in np.arange(1,100,2):
    xc = np.percentile(Xm, alpha)
    xcs.append(xc)
```

```
        alphas.append(alpha)

    pvalue = 0.23
    # Gera a curva de valores críticos por alpha
    plt.figure(figsize=(8,4))
    plt.plot(alphas,xcs)
    plt.axhline(y=xobs, color='red', linestyle='--', label = 'xobs')
    plt.axvline(x=pvalue*100, color='blue', linestyle='--', label = 'p-valor')
    plt.xlabel(r'$\alpha$', fontsize=20)
    plt.ylabel(r'$\bar{X}$', fontsize=20)
    plt.legend()
    plt.grid()
    plt.show()
```

O valor de α em que há a transição entre aceitar ou rejeitar H_0 é definido como o valor p. Calculando o valor p: $P(\bar{X} > \bar{x}_{obs}|\mu = \mu_0) = \alpha$ usando uma implementação em *Python*:

```
    # Define inicialmente o valor p como zero
    pvalue = 0

    # Laço para o cálculo do valor p
    for i in range(0, len(Xm)):
        if(Xm[i] < xobs):
            pvalue = pvalue + 1
    pvalue = pvalue/len(Xm)

    # Mostra o valor p definido
    print('Valor p: ', np.round(pvalue, 3))
```

Pode-se concluir então sobre o **valor p** é que ele representa o **menor nível de significância α no qual é possível rejeitar a hipótese nula** ou em outras palavras, a probabilidade de observar os dados se a hipótese nula for verdadeira.

Quanto maior o valor p, maior é a evidência contra a aceitação da hipótese nula. Vamos avaliar alguns exemplos diretos:

$p = 0.01$, significa que em 1 de cada 100 experimentos, a H_0 Será verdadeira;

- $p = 0.1$, significa que em 10 de cada 100 experimentos, a H_0 Será verdadeira;
- $p = 0.5$, significa que em 50 de cada 100 experimentos, a H_0 Será verdadeira;
- $p = 0.9$, significa que em 90 de cada 100 experimentos, a H_0 Será verdadeira.

Usualmente costuma-se avaliar o nível de significância de acordo com a tabela abaixo:

Nível de significância	Decisão
$p < 0.01$	Evidência muito forte contra H_0
$0.01 < p < 0.05$	Forte evidência contra H_0
$0.05 < p < 0.1$	Fraca evidência contra H_0
$p > 0.1$	Pouquíssima ou nenhuma evidência contra H_0

Exemplo : Uma companhia de ônibus planejou uma nova rota. Um estudo preliminar afirma que a duração da viagem pode ser considerada uma variável aleatória com distribuição normal com μ = 300 minutos e σ = 30 minutos. As 10 primeiras viagens resultaram em uma média igual a 314 minutos. Esse resultado comprova ou não o estudo preliminar?

$$H_0 : \mu \leq 300$$

$$H_1 : \mu > 300$$

$$P\left(\overline{X} > 314|\mu = 300\right) = P\left(\frac{\overline{X} - \mu}{\frac{\sigma}{\sqrt{n}}} > \frac{314 - \mu}{\frac{\sigma}{\sqrt{n}}}\right) = P\left(\frac{\overline{X} - \mu}{\frac{\sigma}{\sqrt{n}}} > \frac{314 - 300}{\frac{30}{\sqrt{10}}}\right) = P(Z > 1,48)$$

Lembrando que $P(Z > 1,48) = 1 - P(Z < 1,48)$:

```
# Parâmetros
Xobs = 314    # média das observações
mu = 300      # média da população
sigma = 30    # desvio padrão
n = 10        # quantidade de elementos na amostra

# Cálculo do Z-Score
z = (Xobs - mu)/(sigma/np.sqrt(n))

# Mostra o valor do Z-Score e o valor p
print('Z - Score: ', z)
print('Valor p: ', 1 - st.norm.cdf(z))
```

Após a simulação, como o valor p não é tão pequeno, pode-se dizer que há poucas evidências para aceitar H_0.

4. Comparação entre duas médias

Uma aplicação do teste de hipóteses voltado para Ciência de Dados seria na seleção de atributos, onde se compara as distribuições de uma determinada variável em relação a variável resposta e avalia se as médias destas distribuições são iguais:

$$H_0 : \mu_1 = \mu_2$$
$$H_a : \mu_1 \neq \mu_2$$

Se as médias forem iguais ou próximas, esta variável tem pouco poder de predição da variável resposta (ela não sabe diferenciar as classes da variável resposta). Quanto mais separadas forem as distribuições, melhor é aquela variável serve para classificar a variável resposta.

Para resolver a comparação de média, o teste a ser utilizado é o **t de Student**.

4.1 t de Student

O conceito por trás do teste **t de Student** é que a partir de diversas amostras de mesmo tamanho n, oriundas da mesma população e onde as médias amostrais seguem uma distribuição normal (garantido pelo Teorema Central do Limite para um valor de $n > 30$), as médias dessas amostras seguem uma distribuição t de Student para valores de n menores ou maiores que 30. A fórmula para calcular o valor da estatística t é definido pela seguinte relação:

$$t = \frac{\bar{X} - \mu_0}{\frac{\sigma}{\sqrt{n}}}$$

Onde dos valores apresentados na fórmula representam:

- $\bar{X}$ - média das amostras;
- μ_0 - média a ser comparada;
- σ - desvio padrão das amostras;
- n - número de elementos na amostra.

O formato da distribuição t de Student depende do número de graus de liberdade (número de elementos em uma amostra menos um), onde quanto maior o número de graus de liberdade, mais "concentrada" é a distribuição e mais próximo de se assemelhar a uma distribuição normal.

Exemplo de Aplicação: Compare duas distribuições de amostras e verifique se elas possuem a mesma média amostral, onde as amostras seguem respectivamente uma distribuição normal $N_1(0,2)$ e $N_2(5,2)$.

Identificando as hipóteses para o teste t de Student:

$$H_0 : \mu_1 = \mu_2$$

$$H_a : \mu_1 \neq \mu_2$$

Para gerar as curvas das distribuições conforme o enunciado, segue o código implementado em *Python* abaixo:

```
# Parâmetros da curva 1
n = 1000    # número de amostras
mu1 = 0     # média 1
sigma1 = 2  # sigma 1
x1 = np.random.normal(mu1, sigma1, n) # sorteia uma amostra de tamanho n

# Parâmetros da curva 2
mu2 = 5     # média 2
sigma2 = 2  # sigma 2
x2 = np.random.normal(mu2, sigma2, n) # sorteia uma amostra de tamanho n

# Histograma das distribuições
plt.figure(figsize=(8,4))
a1 = plt.hist(x=x1, bins=20, color='blue', alpha=0.7, rwidth=0.85, density=True)
a2 = plt.hist(x=x2, bins=20, color='red', alpha=0.7, rwidth=0.85, density=True)
plt.show(True)
```

Para aplicar o teste t de Student, utilize da função *ttest_ind* oriunda da biblioteca *SciPy*:

```
# Carrega a função ttest_ind
from scipy.stats import ttest_ind

# Aplicando o teste t-Student
t_stat, p = ttest_ind(x1, x2)

# Mostra o resultado do valor t de Student e o valor p
print("Valor t : ", t_stat)
print("Valor p : ", p)
```

Para este teste do exemplo, o valor p é muito pequeno, portanto rejeita-se a hipótese H_0, indicando que as médias populacionais são diferentes.

Materiais Complementares

Documentação do SciPy;

Artigo O que realmente um teste de hipóteses quer nos dizer? escrito por Marcelo Randolfo;

Referências

Pedro A. Morettin, Wilton O. Bussab, Estatística Básica, 8ª edição

Peter Bruce, Andrew Bruce & Peter Gedeck, Practical Statistics for Data Scientists, 50+ Essential Concepts Using R and Python, 2ª edition

Ron Larson & Betsy Farber, Estatística Aplicada, 6ª edição.