

★ Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)

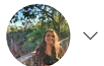


Testes de hipótese: tipos, exemplos e equívocos

Open in app ↗



12 min read · Jan 13



Share



More

Texto escrito por: Julio Manuel e Daniel Scaramelli.



Fonte: [Vecteezy](#).

Introdução

Sejam bem-vindos a mais uma edição do Turing Talks! Hoje, daremos continuidade à nossa série de textos sobre testes de hipótese, destacando os principais tipos de

testes, implementações deles no Python e cuidados que devemos ter na análise e interpretação dos modelos construídos.

Simplificando (muito) a definição, o teste de hipóteses tem por objetivo verificar se é possível rejeitar uma determinada hipótese estatística, que denotamos por hipótese nula. Podemos usá-lo para verificar, por exemplo, se a média dos dados de uma amostra é compatível com a média da população ou se é possível rejeitar essa possibilidade.

Para entender de fato como funcionam os testes de hipóteses, é fortemente recomendada a leitura do [primeiro Turing Talks da série](#), no qual explicamos detalhadamente a lógica por trás dos testes de hipóteses. Os conceitos nele apresentados serão essenciais para essa segunda parte.

Principais testes de hipóteses

Sem mais delongas, vamos aos principais tipos de testes de hipóteses:

À princípio, existem dois tipos de testes de hipótese: testes paramétricos e testes não-paramétricos. Testes paramétricos são aqueles em que fazemos uma suposição sobre a disposição dos dados sobre os quais testamos as hipóteses (isto é, podemos dizer que os dados seguem uma distribuição normal, uniforme, etc.). Os não-paramétricos são, então, aqueles em que não fazemos suposições sobre a distribuição dos dados. Sabendo disso, podemos falar do primeiro teste de hipótese: o Z-test (ou teste Z).

Z-test

O Z-test é usado quando queremos comparar a média de uma amostra com a média populacional. Para performar o Z-test, devemos conhecer a variância populacional e a população deve seguir uma distribuição normal. Considere a seguinte receita para realizar um Z-test:

1º passo: defina a média populacional μ_0 que deve corresponder à hipótese nula H_0 e defina o tipo de hipótese alternativa H_a sobre a média amostral X (unilateral ou bilateral). Além disso, escolha o nível de significância.

2º passo: calcule a estatística do teste Z-score, dada por

$$Z_s = \frac{\bar{X} - \mu_0}{\frac{\sigma}{\sqrt{n}}} \text{ Normalizar}$$

onde σ é o desvio-padrão populacional e n o tamanho da amostra. Pense, intuitivamente, que o numerador é uma forma simples de expressar a diferença entre os valores das hipóteses e o denominador é uma forma de padronizar essa diferença (ou normalizar).

3º passo: calcule o valor-p. Pelo fato de supormos que a estatística de teste segue uma distribuição normal, os valores-p serão, respectivamente, para testes laterais à direita, para testes laterais à esquerda e para testes bilaterais dados por $P(Z < Z_s)$, $P(Z > Z_s)$ e $2\min\{P(Z < Z_s), P(Z > Z_s)\}$. Também, $Z \sim N(0,1)$ (i.e., Z é a distribuição normal padrão).

4º passo: compare o valor-p com o nível de significância (caso o valor-p seja menor ou igual a , rejeite H_0 — caso contrário — o teste falhou a rejeitar H_0).

Agora, vamos fazer um exemplo em Python, usando a biblioteca scipy.

Exemplo 1: Considere uma população de estudantes de diferentes escolas com média 7.5 com desvio-padrão de 0.5 num dado exame. Uma turma de 15 alunos tem média 7.75 nesse exame. Queremos saber se a média dessa turma é significativamente maior do que a populacional. Tome $\alpha = 0.05$.

Input:

```
1  # Importando as ferramentas necessárias
2  import numpy as np
3  from scipy.stats import norm
4  # Criando as variáveis
5  alpha = 0.05
6  media_amostral = 7.75
7  media_populacional = 7.5
8  dp_populacional = 0.50
9  n = 15
10 # Calculando o z-score
11 z_score = (media_amostral - media_populacional)/(dp_populacional/np.sqrt(n))
12 # Calculando o valor-p
13 p_value = 1 - norm.cdf(z_score, loc = 0, scale = 1)
14 # Checando o valor-p e vendo se podemos rejeitar a hipótese nula
15 if alpha <= p_value:
16     print(f"O valor-p é {round(p_value,4)} e não podemos rejeitar a hipótese nula")
17 else:
18     print(f"O valor-p é {round(p_value,4)} e podemos rejeitar a hipótese nula")
```

tt1 hosted with ❤ by GitHub

[view raw](#)

Output:

O valor-p é 0.0264 e podemos rejeitar a hipótese nula

Aqui, `norm.cdf()` é uma função que calcula a distribuição acumulada para uma normal com média dada pelo parâmetro *loc* e variância dada pelo parâmetro *scale*. Note, então, que quando quisermos fazer um teste lateral à esquerda, nosso p-valor será dado por `norm.cdf(z_score, loc = 0, scale = 1)`, enquanto será 1 — `norm.cdf(z_score, loc = 0, scale = 1)` quando tivermos um teste lateral à direita.

T-test

Como foi dito anteriormente, precisamos conhecer de antemão a variância populacional para realizarmos um Z-test. Contudo, muitas vezes essa informação não nos é dada de antemão (na verdade, é bem raro conhecer a variância populacional). Sendo assim, precisaremos de uma estratégia para contornar esse problema — e a estratégia será usar um **estimador** para a variância populacional, que será a **variância amostral** (usamos a notação S^2). Usamos a variância amostral pois ela é um **estimador não-enviesado e consistente**. Assim, ao invés de usarmos a variável padronizada Z_s , usaremos a variável T definida por

$$T = \frac{\bar{X} - \mu_0}{\frac{S}{\sqrt{n}}}$$

S → dp amostral

onde \bar{X} é a estatística em questão, μ_0 é o valor da estatística que corresponde a H_0 e S é o desvio-padrão amostral ($S = S^2$). Agora, note que T não terá distribuição normal como ocorria com nossa estatística de teste no caso do Z-test. Na verdade, a distribuição de T é chamada de **t-Student**. A distribuição t-Student possui um parâmetro chamado de *graus de liberdade*, que corresponde ao número de dados menos um — usamos a notação t_{n-1} .

O T-test tem duas aplicações importantes: *testar uma hipótese sobre a média amostral quando a variância populacional é desconhecida e testar uma hipótese sobre as médias de populações diferentes usando amostras tiradas de ambas*. Como fizemos para o Z-test, definiremos uma receita para fazer um T-test no contexto da primeira aplicação:

1º passo: defina a média populacional μ_0 que deve corresponder à hipótese nula H_0 e defina o tipo de hipótese alternativa H_a sobre a média amostral \bar{X} (unilateral ou bilateral). Além disso, escolha o nível de significância.

2º passo: calcule o valor T já definido anteriormente

3º passo: calcule o valor-p. No caso unilateral, o valor-p será dado por $P(t_{n-1} > T)$ ou $P(t_{n-1} < T)$. No caso bilateral, o valor-p será dado por $2\min\{P(t_{n-1} > T), P(t_{n-1} < T)\}$.

4º passo: compare o valor-p com o nível de significância (caso o valor-p seja menor ou igual a α , rejeite H_0 — caso contrário — o teste não trouxe evidências suficientes para rejeitar H_0).

Agora, vamos fazer um exemplo em Python, usando a biblioteca Scipy.

Exemplo 2: considere uma população cuja média de horas gastas em redes sociais por dia é de 5 horas. Toma-se uma amostra de 10 indivíduos com horas gastas em

redes sociais com os seguintes valores (em horas):

4.5, 6.7, 4.3, 3.4, 6.3, 7.5, 5.4, 8.0, 5.8, 7.1. Deseja-se saber se a média de horas gastas dessa amostra é significativamente diferente da média populacional, considerando um nível de significância estatística = 0.05.

Input:

```
1  # Importando a ferramentas necessárias
2  from scipy.stats import ttest_1samp
3  # Criando as variáveis
4  alpha = 0.05
5  amostra = [4.5, 6.7, 4.3, 3.4, 6.3, 7.5, 5.4, 8.0, 5.8, 7.1]
6  # Calculando o valor-p
7  statistic, p_value = ttest_1samp(amostra, popmean = 5.0,
8                                  alternative = "two-sided")
9  # Checando o valor-p e verificando se podemos ou não rejeitar a hipótese nula
10 if p_value >= alpha:
11     print(f"A estatística T é {round(statistic,4)}. O p-valor é {round(p_value,4)}, e não podemos rejeitar a hipótese nula"
12 else:
13     print(f"A estatística T é {round(statistic,4)}. O p-valor é {round(p_value,4)}, e podemos rejeitar a hipótese nula")
```

tt2 hosted with ❤ by GitHub

[view raw](#)

Output:

A estatística T é 1.8978. O p-valor é 0.0902, e não podemos rejeitar a hipótese nula

A função `ttest_1samp()` toma uma amostra, uma média populacional e o tipo de hipótese que queremos testar, e nos retorna uma tupla com a estatística T e o valor-p.

Agora, vamos estudar a segunda aplicação do T-test: o teste para a média de duas populações diferentes e independentes.

Considere duas populações A e B diferentes e independentes, e tomamos uma amostra de tamanho n_A da população A e n_B da população B. Essas amostras vão ter médias \bar{X}_A e \bar{X}_B , enquanto suas populações vão ter médias μ_A e μ_B e as variâncias amostrais serão S^2_A e S^2_B , respectivamente. Então, formulamos nossas

hipóteses $H_0: \mu_A = \mu_B$ e H_a será uma hipótese unilateral ou bilateral como para outros testes. A receita para fazer um T-test para teste de hipótese sobre duas médias amostrais de duas populações diferentes e independentes é:

1º passo: defina a hipótese nula H_0 (tipicamente será tal que $H_0: \mu_A = \mu_B$) e defina o tipo de hipótese alternativa H_a sobre as médias de A e B (unilateral ou bilateral). Além disso, escolha o nível de significância .

2º passo: calcule a estatística T dada por:

$$T = \frac{\bar{X}_A - \bar{X}_B}{\sqrt{\frac{S_A^2}{n_A} + \frac{S_B^2}{n_B}}}$$

A rigor, o numerador é dado por $(\bar{X}_A - \bar{X}_B - (\mu_A - \mu_B))$. Mas, como dissemos, nossa hipótese nula é, tipicamente, que a média populacional das duas populações são iguais — então, para o cálculo do valor-p (que assume H_0), $\mu_A - \mu_B = 0$. Daí, $(\bar{X}_A - \bar{X}_B - (\mu_A - \mu_B)) = \bar{X}_A - \bar{X}_B$.

3º passo: calcule os graus de liberdade da distribuição t correspondente e, posteriormente, calcule o valor-p. Quando temos duas amostras, os graus de liberdade são dados pela soma dos graus de liberdade das duas populações. Isto é, $(n - 1) = (n_A - 1) + (n_B - 1) = n_A + n_B - 2$.

4º passo: compare o valor-p com o nível de significância (caso o valor-p seja menor ou igual a , rejeite H_0 — caso contrário — o teste falhou a rejeitar H_0).

Agora, vamos fazer um exemplo em Python, usando a biblioteca Statsmodels.

Exemplo 3: divide-se um conjunto de indivíduos em dois grupos para o teste de um xarope, e mede-se a quantidade de tosse observadas por indivíduo em 6 horas. O

primeiro grupo, denotado aqui por A (grupo que recebeu o tratamento) teve as seguintes observações:

1.0, 2.0, 0.0, 3.0, 0.0, 2.0, 0.0, 1.0, 0.0, 4.0, 1.0, 3.0, 0.0, 2.0, 0.0, 1.0, 0.0

O segundo grupo, denotado por B (grupo de controle) teve as seguintes observações:

0.0, 2.0, 1.0, 5.0, 6.0, 3.0, 4.0, 5.0, 3.0, 1.0, 0.0, 2.0, 1.0

Vamos testar se $\mu_B > \mu_A$. Considere um nível de significância de 0.05.

```
1  # Importando a ferramenta necessária
2  from statsmodels.stats.weightstats import ttest_ind
3  # Criando as variáveis
4  grupo_A = [1.0, 2.0, 0.0, 3.0, 0.0, 2.0, 0.0, 1.0, 0.0, 4.0, 1.0, 3.0, 0.0, 2.0, 0.0, 1.0, 0.0]
5  grupo_B = [0.0, 2.0, 1.0, 5.0, 6.0, 3.0, 4.0, 5.0, 3.0, 1.0, 0.0, 2.0, 1.0]
6  alpha = 0.05
7  # Calculando o p-valor
8  tstat, pvalue, df = ttest_ind(grupo_A, grupo_B, alternative = "smaller", usevar = "unequal", va
9  # Checando o valor-p e verificando se podemos ou não rejeitar a hipótese nula
10 if pvalue >= alpha:
11     print(f"A estatística T é {round(tstat,4)}. O p-valor é {round(pvalue,4)}, e não podemos re
12 else:
13     print(f"A estatística T é {round(tstat,4)}. O p-valor é {round(pvalue,4)}, e podemos rejeit
```

tt3 hosted with ❤ by GitHub

[view raw](#)

Output:

A estatística T é -2.1533. O p-valor é 0.022, e podemos rejeitar a hipótese nula

Aqui, a alternativa testada foi a de que a diferença entre as médias era menor do que o valor fornecido pelo parâmetro value. Testamos esta alternativa pelo simples fato de que a nossa alternativa $H_a: \mu_B > \mu_A$ equivale a $H_a: \mu_A - \mu_B < 0$. Também, fizemos o nosso teste relaxando a hipótese de que a variância das amostras é igual. O teste retorna uma tupla com três elementos: o valor da estatística T, o valor-p e os graus de liberdade usados no cálculo do valor-p.

ANOVA



Por enquanto, fizemos testes para a média de duas populações. Mas poderíamos querer testar:

$$H_0: \mu_1 = \mu_2 = \dots = \mu_k$$

HA: pelo menos uma igualdade em H_0 não vale

onde $1, \dots, K$ são populações e μ_1, \dots, μ_k são suas respectivas médias (o ANOVA é um teste paramétrico, e assumimos que as populações assumem uma distribuição normal). Para podermos resolver esse problema, devemos usar o teste ANOVA (ou análise de variância). Podemos entender a intuição por trás do ANOVA pensando sobre o problema que ele quer resolver. Imagine que temos K populações (chamaremos essas populações de grupos) e queremos testar a hipótese que formulamos acima. Uma primeira tentativa de resolver o problema seria aplicar um t-test para cada par de populações (isto é, performar um t-test comparando a população 1 com a 2, com a 3, até a K -ésima; daí a 2 com a 3, até a K -ésima e assim por diante), mas note que isso seria muito ineficiente e a probabilidade de cometermos um erro do tipo I em pelo menos um desses testes seria altíssima. Para resolver esse problema é que usamos o ANOVA. O ANOVA usa uma estatística de teste chamada F , e, informalmente, a F é dada por:

$$F = (\text{variação entre os grupos}) / (\text{variação dentro dos grupos})$$

Basicamente, o ANOVA quebra a variação total nos dados disponíveis em duas partes: a variação entre os grupos (isto é, o quanto que os dados variam entre os grupos) e a variação dentro dos grupos (isto é, o quanto que os dados variam dentro de cada grupo). Note que quando a estatística F for “grande”, a variação de todos os dados disponíveis vai ser mais explicada pela variação entre os grupos do que dentro dos grupos: o que indica, então, que os dados entre os grupos variam mais significativamente e as médias são mais improváveis de serem iguais. Claro que o contrário também vale: quando a estatística F for “pequena”, a variação total vai ser mais explicada entre a variação dentro dos grupos e não entre os grupos: então os dados variam menos significativamente entre os grupos e as médias são mais prováveis de serem iguais.

A operacionalidade por trás do ANOVA é um pouco complexa e está além do escopo deste texto. Sendo assim, podemos pular para o nosso exemplo de aplicação do ANOVA em Python, usando a biblioteca `scipy`. Usaremos a função `f_oneway()` da

biblioteca, que simplesmente recebe as amostras que temos como input e retorna-nos a estatística F e o valor-p do teste de hipótese.

Exemplo 4: considere um mamífero que está presente em 5 países de continentes diferentes (Brasil, Austrália, África do Sul, Portugal e Indonésia). Uma equipe de biólogos mede o tamanho de um osso específico para amostras de 6 indivíduos da espécie tomadas nos 5 países indicados. Considere as observações, em cm, para cada país:

Brasil: 3.0, 4.9, 3.4, 4.5, 4.4, 4.3

África do Sul: 4.5, 4.5, 2.9, 4.3, 4.1, 3.2

Austrália: 4.3, 3.9, 3.6, 2.8, 3.1, 4.0

Portugal: 3.2, 2.4, 2.7, 3.8, 3.9, 3.5

Indonésia: 3.9, 3.4, 3.6, 4.1, 4.5, 2.7

Deseja-se testar a hipótese de que as médias das populações desse mamífero nos países indicados não são todas iguais. Considere um nível de significância de 0.05.

Input:

```
1  # Importando a ferramenta necessária
2  from scipy.stats import f_oneway
3  # Criando as variáveis do problema
4  alpha = 0.05
5  brasil = [3.0, 4.9, 3.4, 4.5, 4.4, 4.3]
6  africa_do_sul = [4.5, 4.5, 2.9, 4.3, 4.1, 3.2]
7  australia = [4.3, 3.9, 3.6, 2.8, 3.1, 4.0]
8  portugal = [3.2, 2.4, 2.7, 3.8, 3.9, 3.5]
9  indonesia = [3.9, 3.4, 3.6, 4.1, 4.5, 2.7]
10 # Calculando a estatística F e o valor-p
11 fstat, pvalue = f_oneway(brasil, africa_do_sul, australia, portugal, indonesia)
12 # Checando a estatística F e o valor-p e vendo se podemos ou não rejeitar a hipótese nula
13 if alpha <= pvalue:
14     print(f"A estatística F é {round(fstat,4)}, o valor-p é {round(pvalue,4)} e não podemos rejeitar a hipótese nula"
15 else:
16     print(f"A estatística F é {round(fstat,4)}, o valor-p é {round(pvalue,4)} e podemos rejeitar a hipótese nula")
```

Output:

A estatística F é 1.4483, o valor-p é 0.2476 e não podemos rejeitar a hipótese nula

Agora que já vimos como aplicar na prática testes de hipóteses, é interessante conhecermos os erros de interpretação associados a esses testes e como eles se relacionam com alguns exemplos do cotidiano.

A primeira falha associada a testes de hipóteses (e a outros testes estatísticos) é a falta de cuidado no processo de amostragem, seja selecionando uma amostra que não represente a população ou comparando amostras enviesadas por determinado fator (altura, idade etc). Por exemplo, suponha que uma empresa farmacêutica quer testar se existe diferença de eficácia entre dois remédios com base no tempo de recuperação dos pacientes. Para tanto, é formulada a hipótese nula de que essa diferença de eficácia não existe e o experimento é testado em dois grupos: o grupo A toma o primeiro remédio e o grupo B o segundo. Porém, o experimento não se deu ao trabalho de selecionar adequadamente a amostra, fazendo com que o grupo A fosse majoritariamente composto por jovens e o grupo B por idosos. Com isso, o estudo foi enviesado pela variável de idade, o que pode levar a falsas conclusões (ou não levar à conclusão verdadeira).

Um segundo problema associado ao teste de hipóteses é a escolha do modelo a ser adotado. Normalmente, o caso que mais gera dúvidas é quando desejamos verificar se há diferença significativa na média de duas amostras. Para tanto, dois testes podem ser executados: o z-test e o t-test. Como o t-test é capaz de fazer essa verificação sem a utilização dos dados de variância da população, que normalmente são de difícil acesso, o modelo acaba sendo usado de maneira generalizada, mesmo quando o desvio-padrão é conhecido, caso em que o z-test mostra-se mais adequado e preciso. Além disso, costuma ser negligenciado o fato de que o z-test é baseado em uma distribuição normal da população e o t-test em uma distribuição t de Student; logo, para usar esses modelos, a distribuição das amostras também deve ser compatível com essas distribuições.

O terceiro e principal erro associado aos testes de hipóteses é a interpretação equivocada do p-valor e sua relação com o nível de significância α . Conforme vimos neste Turing Talks, o α costuma variar entre 1% e 10%. Porém, esse intervalo não

deve ser levado à ferro e fogo, dado que o fator principal para definição de α é a gravidade do erro. Suponha, por exemplo, que você é um médico e que seu paciente pode estar com câncer. Para verificar essa hipótese, testa a hipótese nula dele não ter câncer. Você sabe que, se entrar imediatamente com a medicação, o paciente irá se recuperar. Sendo assim, **o falso positivo é muito mais grave que o falso negativo, pois, enquanto o primeiro leva à morte, o segundo gera apenas efeitos colaterais leves**. Sendo assim, o nível de significância a ser adotado deve ser muito baixo, de modo que mesmo que o p-valor seja baixo, ainda faça sentido optar pela prevenção.

Ainda no tema de interpretação equivocada do p-valor, outro problema comum é tirar conclusões quando o p-valor é maior que o nível de significância. Quando isso ocorre, **a única conclusão possível é que a hipótese nula não pode ser rejeitada. Isso não quer dizer, por exemplo, que a hipótese alternativa pode ser rejeitada ou que a hipótese nula é verdadeira**, duas **falhas de interpretação** muito recorrentes no estudo de testes de hipóteses.

Conclusão

Neste Turing Talks, você pôde aprender um pouco mais sobre os principais modelos e aplicações do Teste de Hipóteses e os cuidados que devem ser tomados na hora de trabalhar com esse tipo de teste.

Esperamos que você tenha gostado e agradecemos por chegar até aqui. Se quiser conhecer um pouco mais sobre o que fazemos no **Turing-USP**, não deixe de nos seguir nas redes sociais: [Facebook](#), [Instagram](#), [LinkedIn](#) e, claro, acompanhar nossos posts no [Medium](#). Para acompanhar também mais de perto e participar de nossas discussões e eventos, entre no nosso servidor do [Discord](#).

Por hoje é só, até a próxima!

[Turing Usp](#)[Statistics](#)[Python](#)[Data Science](#)[Hypothesis Testing](#)