

# Teste T de Student

## 1. Introdução

- O teste t de Student é um método estatístico inferencial paramétrico utilizado para comparar as médias entre dois grupos diferentes (teste t de duas amostras) ou com um valor específico (teste t de uma amostra).
- No teste t, a estatística do teste segue a distribuição t (tipo de distribuição de probabilidade contínua) sob a hipótese nula.
- A distribuição t é primeiramente proposta por William Sealy Gosset e publicada sob o nome fictício de "Student" na revista científica Biometrika. Por conseguinte, a distribuição t é também conhecida como a distribuição t de Student.
- Em contraste com o teste Z, que requer um tamanho de amostra maior (as distribuições t e z são semelhantes para amostras maiores, digamos  $n \geq 30$ ), o teste t é especialmente desenvolvido para os dados de tamanho de amostra pequena ( $n \leq 30$ ).
- O teste t é também aplicável para dados de tamanho de amostra extremamente pequeno ( $n \leq 5$ ).
- O teste t fornece resultados mais conservadores para conjuntos de dados de amostra pequena.
- O teste t tem três tipos principais: Um teste t de amostra, dois testes t de amostra (não emparelhado ou independente), e teste t emparelhado.
- Num teste t de amostra, a média da amostra é comparada com os valores das hipóteses, enquanto que em dois testes t de amostra (teste t independente ou não emparelhado), são comparadas as médias de dois grupos independentes. No teste t emparelhado, as diferenças entre o par de variáveis dependentes são comparadas.

## 2. Tipos de teste-t

### 2.1. Teste-t para uma amostra (One Sample t-test)

#### 2.1.1. Motivação

- comparar a média de uma amostra com a média populacional.
- No teste t, a variância da população ( $\sigma^2$ ) é desconhecida e é estimada a partir da variância da amostra (s).

#### 2.1.2. Exemplos

- Um fabricante alega a vida média das pilhas AA é de 300 minutos. Se você suspeita-se que essa alegação não é válida, como poderia mostrar que ela é falsa?
- Verificar se o peso médio dos bebês de uma maternidade é similar ao esperado na população
- Testar se a frequência respiratória de atletas de corrida difere da de outros atletas
- Por exemplo, uma bola tem um diâmetro de 5 cm e queremos verificar se o diâmetro médio da bola da amostra aleatória (por exemplo, 50 bolas) colhida da linha de produção difere do tamanho conhecido.

### 2.1.3. Pressupostos

- Os dados são aleatórios e representativos da população
- A variável dependente deve ter uma distribuição aproximadamente normal (Shapiro-Wilks Test)
- As observações são independentes uma da outra
- A variável dependente deve ser contínua
- A variação da população ( $\sigma^2$ ) é desconhecida

### 2.1.4. Notas

Um teste t de amostra é relativamente robusta à hipótese de normalidade quando o tamanho da amostra é grande ( $n \geq 30$ )

### 2.1.5 Códigos

In [30]:

```
from scipy import stats as st
# pip install bioinfokit
from bioinfokit.analys import get_data
```

In [38]:

```
# Load dataset as pandas dataframe
df = get_data('t_one_samp').data
df.head(2)
```

Out[38]:

	size
0	5.739987
1	5.254042

In [36]:

```
a = df['size'].to_numpy()
a
```

Out[36]:

```
array([5.73998725, 5.25404227, 5.15238834, 4.87081905, 3.53625064,
        6.25717079, 5.67399212, 4.49975755, 4.32676488, 5.03792595,
        3.03686598, 5.18216044, 4.99427902, 3.91368411, 4.80401254,
        4.4536053 , 4.42941836, 4.10680117, 4.83134172, 6.12282451,
        5.00936774, 4.50611441, 5.29967781, 4.81613333, 3.33872827,
        6.20310606, 4.85366638, 5.41723274, 6.58384684, 5.10426761,
        4.02122049, 3.99308312, 5.41710494, 6.29843941, 3.80142355,
        5.34204192, 4.16098619, 4.78060505, 4.22507227, 7.37100455,
        6.55667429, 6.61408012, 5.86276128, 6.62984248, 5.91642598,
        6.25304554, 3.88957758, 5.6923657 , 4.85272771, 3.52936383])
```

In [33]:

```
# use parameter "alternative" for two-sided or one-sided test
st.ttest_1samp(a=a, popmean=5) # 5: Expected value in null hypothesis.
```

Out[33]:

```
Ttest_1sampResult(statistic=0.36789006583267403, pvalue=0.714539654336473)
```

Como não obtém resultados estatísticos detalhados (tais como intervalos de confiança, graus de liberdade, tamanho da amostra, etc.) da função `ttest_1samp()`, realizaremos um teste t de amostra usando a função `ttest()` no `bioinfokit`.

In [39]:

```
from bioinfokit.analys import stat
res = stat()
res.ttest(df=df, test_type=1, res='size', mu=5)
print(res.summary)
```

One Sample t-test

```
-----
Sample size      50
Mean             5.05128
t                0.36789
Df               49
p value (one-tail) 0.35727
p value (two-tail) 0.71454
Lower 95.0%      4.77116
Upper 95.0%      5.3314
-----
```

Para acessar valores t-value e p-value (two-tail)

In [35]:

```
t, p = res.result[1], res.result[3]
t, p
```

Out[35]:

```
(0.36789006583267403, 0.714539654336473)
```

O valor de p obtido do teste t de uma amostra não é significativo ( $p > 0,05$ ) e, portanto, concluímos que o diâmetro médio das bolas numa amostra aleatória é igual a 5 cm.

### Explicação do comando `ttest_1samp`

In [3]:

```
print (stats.ttest_1samp.__doc__)
```

Calculate the T-test for the mean of ONE group of scores.

This is a two-sided test for the null hypothesis that the expected value (mean) of a sample of independent observations `a` is equal to the given population mean, `popmean`.

Parameters

-----

`a` : array\_like

Sample observation.

`popmean` : float or array\_like

Expected value in null hypothesis. If array\_like, then it must have

the

same shape as `a` excluding the axis dimension.

`axis` : int or None, optional

Axis along which to compute test; default is 0. If None, compute over

r

the whole array `a`.

`nan_policy` : {'propagate', 'raise', 'omit'}, optional

Defines how to handle when input contains nan.

The following options are available (default is 'propagate'):

\* 'propagate': returns nan

\* 'raise': throws an error

\* 'omit': performs the calculations ignoring nan values

`alternative` : {'two-sided', 'less', 'greater'}, optional

Defines the alternative hypothesis.

The following options are available (default is 'two-sided'):

\* 'two-sided'

\* 'less': one-sided

\* 'greater': one-sided

.. versionadded:: 1.6.0

Returns

-----

`statistic` : float or array

t-statistic.

`pvalue` : float or array

Two-sided p-value.

Examples

-----

```
>>> from scipy import stats
```

```
>>> rng = np.random.default_rng()
```

```
>>> rvs = stats.norm.rvs(loc=5, scale=10, size=(50, 2), random_state=rn
```

g)

Test if mean of random sample is equal to true mean, and different mean. We reject the null hypothesis in the second case and don't reject it in the first case.

```
>>> stats.ttest_1samp(rvs, 5.0)
```

```
Ttest_1sampResult(statistic=array([-2.09794637, -1.75977004]), pvalue=array([0.04108952, 0.08468867]))
```

```
>>> stats.ttest_1samp(rvs, 0.0)
Ttest_1sampResult(statistic=array([1.64495065, 1.62095307]), pvalue=array([0.10638103, 0.11144602]))
```

Examples using axis and non-scalar dimension for population mean.

```
>>> result = stats.ttest_1samp(rvs, [5.0, 0.0])
>>> result.statistic
array([-2.09794637, 1.62095307])
>>> result.pvalue
array([0.04108952, 0.11144602])
```

```
>>> result = stats.ttest_1samp(rvs.T, [5.0, 0.0], axis=1)
>>> result.statistic
array([-2.09794637, 1.62095307])
>>> result.pvalue
array([0.04108952, 0.11144602])
```

```
>>> result = stats.ttest_1samp(rvs, [[5.0], [0.0]])
>>> result.statistic
array([[ -2.09794637, -1.75977004],
       [ 1.64495065, 1.62095307]])
>>> result.pvalue
array([[0.04108952, 0.08468867],
       [0.10638103, 0.11144602]])
```