# Hypothesis testing

Dhafer Malouche

## Contents

## 1 Introduction

Hypothesis testing is a data analysis method conducted to test one hypothesis (call null hypothesis, $H_0$) against another hypothesis (the alternative hypothesis, $H_1$).

We will use the random sample $X_1, \ldots, X_n$ (the data) to help decide between the hypothesis $H_0$ or $H_1$ with a fixed level of significance $\alpha$ which is the error to reject $H_0$ knowing that $H_0$ is correct.

$$\alpha = \mathbb{P}\left(H_0 \text{ rejected } \mid H_0 \text{ true}\right)$$

Any Hypothesis testing procedure should derive the following

- The test statistic $T$: It's a random variable computed from the random sample and where the probability distribution of $T$ is known when $H_0$ is true.
- The observed statistics $t_{\text{obs}}$ from $T$ computed from the observed random sample $x_1, \ldots, x_n$:

$$t_{\text{obs}=T(x_1,\ldots,x_n)}$$

- The **p-value**: It's the largest probability to reject $H_0$ assuming that $H_0$ correct. A smaller p-value means stronger evidence in favor of the alternative hypothesis.

There are two types of Hypothesis testing procedures: parametric and non-parametric testing.

- Parametric hypothesis testing is a testing procedure used when the hypothesis is based on comparing a population parameter to given values.
- Non-parametric hypothesis testing is used when the hypothesis is not based on a population parameter. It's about testing one assumption against its opposite.
- Parametric tests are more powerful and reliable than non-parametric tests.
- The hypothesis is developed on the parameters of the population distribution.
- We will see in this chapter how to perform Hypothesis testing in the following cases:
  - The mean: comparing to a given value, comparing between two means
  - The proportion: comparing to a given value, comparing between two means

# 2 Testing the mean

## 2.1 The theory

We would like to test the following null hypothesis

$$H_0 : \mu = \mu_0$$

versus

$$H_1 : \mu \neq \mu_0$$

where $\mu_0$ is given from a random sample $X_1, \ldots, X_n$ assumed to be generated from a Normal distribution with mean $\mu$ and unknown variance $\sigma^2$.

The test statistic of the test is

$$T = \sqrt{n}\,\frac{\overline{X} - \mu}{S}$$

where $\overline{X}$ is the sample mean and $S$ is the sample standard deviation.

Under $H_0$, $T$ is equal to

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

and follows a $t-$distribution with $n-1$ degrees of freedom.

## 2.2 Practice with Python

Let's generate a random sample of size 15, mean $\mu = -2$, and standard deviation $\sigma = 2$ (then with variance $\sigma^2 = 4$).

[1]:

```python
import numpy as np
np.random.seed(7654567)
x=np.random.normal(size=15,loc=-2,scale=2)
```

[2]:

```python
x
```

```
[2]: array([-3.89395351, -2.55250403, -2.39746865,  0.98703995, -2.8607237 ,
            -1.39481847, -0.74237156, -4.63886095, -3.81258843, -2.89188048,
            -1.50048335, -3.31429764, -3.28882943, -0.96256977, -2.0684448 ])
```

We assume that the random sample `x` is generated from a Normal probability distribution with unknown mean and variance. We will test now the following hypothesis

$$H_0: \ \mu = -2$$

versus

$$H_1: \ \mu \neq -2$$

```
[3]: from scipy.stats import ttest_1samp
```

```
[4]: test_mean=ttest_1samp(x, -2, alternative='two-sided')
```

The output above shows that $t_{\text{obs}}$ is

```
[5]: test_mean.statistic
```

```
[5]: -0.948081719359044
```

and the pvalue is

```
[6]: test_mean.pvalue
```

```
[6]: 0.3591669622802579
```

Since this p-value is greater than 0.05 (5%), we can conclude that we can accept the hypothesis $H_0$.

**How can the test statistc and the p-value are computed?**

```
[7]: m=np.mean(x)
     m
```

```
[7]: -2.3555169880044056
```

```
[8]: std_error=np.std(x)/np.sqrt(len(x)-1)
     std_error
```

```
[8]: 0.3749855953817513
```

Under $H\_0 : \mu = -2$, the test statistic is then

```
[9]: (m+2)/std_error
```

```
[9]: -0.948081719359044
```

```
[10]: test_mean.statistic
```

```
[10]: -0.948081719359044
```

The p-value is then computed as follows

```
[11]: from scipy import stats
      X = stats.t(len(x)-1)
```

```
[12]: 2*X.cdf(test_mean.statistic)
```

```
[12]: 0.3591669622802579
```

And the p-value is

```
[13]: test_mean.pvalue
```

```
[13]: 0.3591669622802579
```

We can also perform the following hypothesis testing. It's called the lower-tail alternative test:

$$H_0 : \ \mu = -2$$

versus

$$H_1 : \ \mu < -2$$

```
[14]: test_mean1=ttest_1samp(x, -2, alternative='less')
```

```
[15]: test_mean1.statistic
```

```
[15]: -0.948081719359044
```

```
[16]: test_mean1.pvalue
```

```
[16]: 0.17958348114012895
```

It's computed as follows

```
[17]: X.cdf(test_mean.statistic)
```

```
[17]: 0.17958348114012895
```

We can also perform the following hypothesis testing. It's called the **upper-tail** alternative test:

$$H_0 : \ \mu = -2$$

versus

$$H_1 : \ \mu > -2$$

```
[18]: test_mean2=ttest_1samp(x, -2, alternative='greater')
```

```
[19]: test_mean2.statistic
```

```
[19]: -0.948081719359044
```

[20]:

```
test_mean2.pvalue
```

[20]:

```
0.8204165188598711
```

The p-value is computed as follows

[21]:

```
1-X.cdf(test_mean.statistic)
```

[21]:

```
0.8204165188598711
```

## 3 Comparing two means

We have two types of hypothesis testing comparing two means:

- A paired sample t-test is a dependent sample t-test, which is used to decide whether the mean difference between two observations of the same group is zero.

**Example:** Compare the difference in blood pressure level for a group of patients before and after some drug treatment.

- A two-sample t-test is used for comparing the significant difference between two independent groups. This test is also known as an independent samples t-test.

**Example:** Comparing between the salaries of a sample of men and women employees.

### 3.1 Paired sample t-test

We are going to the following hypothesis:

- $H_0$ : Mean difference between the two dependent samples is 0.
- $H_1$: Mean difference between the two dependent samples is not 0.

**Example:** we're comparing the grades between the Quiz 1 and the Quiz 2

[22]:

```
import pandas as pd
```

[23]:

```
df=pd.read_csv('student_grades.csv')
```

[24]:

```
df.head()
```

[24]:

```
    Student  Section    Quiz 1   Quiz 2  Midterm 1
0         1  Section 1    17.5  13.4575       22.5
1         2  Section 1    16.5  14.2125       25.0
2         3  Section 2    12.5  14.6150       23.5
3         4  Section 2    10.5  15.2100       25.0
```

We remove the missing values from the data

[25]:

```
df=df.dropna()
```

```
[26]: df.head()
```

```
[26]:    Student  Section   Quiz 1   Quiz 2  Midterm 1
      0        1  Section 1   17.5  13.4575       22.5
      1        2  Section 1   16.5  14.2125       25.0
      2        3  Section 2   12.5  14.6150       23.5
      3        4  Section 2   10.5  15.2100       25.0
```

```
[27]: from scipy.stats import ttest_rel
      test_diffmean1=ttest_rel(df['Quiz 1'],df['Quiz 2'])
```

```
[28]: test_diffmean1.statistic
```

```
[28]: -1.1156139485204906
```

```
[29]: test_diffmean1.pvalue
```

```
[29]: 0.2701415971643917
```

We have tested here the following hypothesis:

$$H_0: \text{ the averages of the grades in Q1 and Q2 are equal}$$

versus

$$H_0: \text{ the averages of the grades in Q1 and Q2 are different}$$

We used a paired t-test and we can conclude that $H_0$ can't be rejected since the p-value is greater than 0.05 (5%) (the given level of significance).

We can test also if the mean $\mu_1$ of the grades of the Quiz 1 is higher than the mean $\mu_2$ of the grades of the Quiz 2:

$$H_0: \mu_1 \geq \mu_2$$

versus

$$H_1: \mu_1 < \mu_2$$

```
[30]: test_diffmean1a=ttest_rel(df['Quiz 1'],df['Quiz 2'],alternative='less')
```

```
[31]: test_diffmean1a.statistic
```

```
[31]: -1.1156139485204906
```

```
[32]: test_diffmean1a.pvalue
```

```
[32]: 0.13507079858219584
```

Conclusion: $H_0$ can't be rejected

[33]:

```
test_diffmean1b=ttest_rel(df['Quiz 1'],df['Quiz 2'],alternative='greater')
```

[34]:

```
test_diffmean1b.pvalue
```

[34]:

```
0.8649292014178042
```

A two-sample t-test

We will compare now the mean of the Quiz 1 between Section 1 and 2

[35]:

```
x1=df['Quiz 1'].loc[df['Section  ']=='Section 1']
x1
```

[35]:

```
0     17.5
1     16.5
5     14.0
9     14.0
11    10.5
12    16.5
13    14.5
14    14.5
15     8.0
16    14.5
18    11.0
20    18.5
22    10.5
23    10.0
28    15.0
29    18.0
32    14.0
36    11.0
37    15.0
41    14.0
44    16.0
45    10.5
46    16.5
48    17.5
50    14.5
51    16.5
Name: Quiz 1, dtype: float64
```

[36]:

```
x2=df['Quiz 1'].loc[df['Section  ']=='Section 2']
x2
```

[36]:

```
2     12.5
3     10.5
4      5.5
7     15.0
```

```
8     17.0
10    11.0
17    14.0
19    17.0
21    14.0
24    11.0
25    15.0
26    16.0
27    19.5
30    14.5
31    18.5
33    11.5
34    10.0
35    20.0
38    18.5
40     9.5
43    13.0
47    10.5
49    13.5
Name: Quiz 1, dtype: float64
```

[37]:

```
from scipy.stats import ttest_ind
test_diffmean2=ttest_ind(x1,x2)
```

[38]:

```
test_diffmean2.statistic
```

[38]:

```
0.4200033340360798
```

[39]:

```
test_diffmean2.pvalue
```

[39]:

```
0.6763969243532744
```

We conclude that both sections have the same means of the grades in Quiz 1.

# 4 Testing the proportion

## 4.1 The theory

Assume that we would like to test the following Hypothesis:

$$H_0 \,:\, p = p_0, \quad \text{versus } H_1 \,:\, p \neq p_0$$

where $p$ is a parameter proportion and $p_0$ is a given value of $p$. The hypothesis $H_0$ (null hypothesis) and $H_1$ will be tested from a given data on a random sample $X_1, \ldots, X_n$ with a Bernoulli distribution. The random variables $X_1, \ldots, X_n$ are binary variable with values 1 and 0 where each random variable $X_i$ takes the value 1 with probability $p$.

In most of the cases the data reported in this type of tests is the number $X$ of success among the $n$ trials. This later random variable $X$ is defined as follows:

$$X = X_1 + \ldots + X_n = \sum_{k=1}^{n} X_k$$

and has a Binomial distribution with size $n$ and probability $p$.

The probability $p$ is often estimated using the random variable $\widehat{p}$:

$$\widehat{p} = \frac{X}{n}.$$

When $n$ is large ($\geq 30$), the random variable $\widehat{p}$ follows approximatively a Normal distribution:

$$\widehat{p} \sim \mathcal{N}\left(p, \frac{p(1-p)}{n}\right).$$

Hence the random variable $Z = \sqrt{n}\frac{\widehat{p} - p}{\sqrt{\widehat{p}(1-\widehat{p})}}$ follows approximately a standard normal distribution.

The random variable $Z$ will be the test statistic of the the proportion hypothesis testing.

**Example:** According to the Washington Post, nearly 45% of all Americans are born with brown eyes, although their eyes don't necessarily stay brown. A random sample of 80 adults found 32 with brown eyes. Is there sufficient evidence at the .01 level to indicate that the proportion of brown-eyed adults differs from the proportion of Americans who are born with brown eyes?

In this example the sample size $n = 80$, the random sample is $X_1, \ldots, X_n$ where each $X_i$ is representing an American adult, $X_i$ is 1 if this adult has brown eyes and 0 elsewhere. The random variable $X$ is the number of Americans with brown eyes among the 80 surveyed adults. In this example the observed of $X$ is 32.

We are testing here the following two hypothesis:

- Null Hypothesis: $H_0 \, : \, p = .45$
- Alternative Hypothesis $H_1 \, : \, p \neq .45$

Under the hypothesis $H_0$ ($p = .45$), the random variable $Z$ is equal to:

$$Z = \sqrt{80}\frac{\widehat{p} - .45}{\sqrt{.45 \times .55}}$$

The observed value of $\widehat{p}$ from the data is:

$$\widehat{p}_{\text{obs}} = \frac{32}{80} = .4$$

Then the observed value of $Z$ is

$$Z_{\text{obs}} = \sqrt{80}\frac{.4 - .45}{\sqrt{.4 \times .6}} = -0.913$$

Since the alternative hypothesis is $H_1 : p \neq .45$. The test is called two-sided test and the p-value of the test is computed as follows:

$$\begin{aligned}
\text{p-value} &= \mathbb{P}\left(|Z| \geq |z_{\text{obs}||H_0} \text{ true}\right) \\
&= \mathbb{P}\left(|Z| \geq 0.913 \mid p = .45\right) \\
&= 2 \times (1 - F_Z(0.913)) \\
&= 0.361
\end{aligned}$$

where $F_Z$ is a cumulative probability function (CDF) of a standard normal distribution.

Since $1\% = 0.01$ is the level of significance of the test, and the p-value is greater than 1%, we can decide to not reject the null hypothesis $H_0$.

## 4.2 Practice with Python

[40]:

```
import numpy as np
from statsmodels.stats.proportion import proportions_ztest
```

[41]:

```
stat, pvalue=proportions_ztest(count=32, nobs=80, value=.45,␣
 ↪alternative='two-sided')
```

[42]:

```
stat
```

[42]:

```
-0.9128709291752767
```

[43]:

```
pvalue
```

[43]:

```
0.3613104285261789
```

[44]:

```
phat=32/80
phat
```

[44]:

```
0.4
```

[45]:

```
Zscore=np.sqrt(80)*(phat-.45)/np.sqrt(.4*.6)
Zscore
```

[45]:

```
-0.9128709291752768
```

[46]:

```
from scipy.stats import norm,t
```

[47]:

```
2*(1-norm.cdf(np.abs(Zscore),loc=0,scale=1))
```

[47]:

```
0.361310428526179
```

### 4.3 Other types of alternative hypothesis

The alternative hypothesis $H_1$ can be also one the following: + $H_1 : p < p_0$: `alternatine='smaller'` + $H_1 : p > p_0$: `alternatine='larger'`

In case of `alternatine='less'` we proceed as follows:

```
[48]: stat, pvalue=proportions_ztest(count=32, nobs=80, value=.45,␣
      ↪alternative='smaller')
```

```
[49]: stat
```

```
[49]: -0.9128709291752767
```

```
[50]: pvalue
```

```
[50]: 0.18065521426308945
```

The pvalue is computed as follows

```
[51]: norm.cdf(stat,0,1)
```

```
[51]: 0.18065521426308945
```

In case of `alternatine='larger'` we proceed as follows:

```
[53]: stat, pvalue=proportions_ztest(count=32, nobs=80, value=.45,␣
      ↪alternative='larger')
```

```
[54]: stat
```

```
[54]: -0.9128709291752767
```

```
[55]: pvalue
```

```
[55]: 0.8193447857369105
```

The pvalue is computed as follows

```
[56]: 1-norm.cdf(stat,0,1)
```

```
[56]: 0.8193447857369105
```

**Example:** Pizza-Hut claims that 90% of its order are delivered within 10 minutes of the time the order is placed. A sample of 100 order revealed that 82 were delivered within the promised time. At 10% significance level, can we conclude that at least 90% of the orders are delivered in less than 10 minutes?

We are testing in this example the following hypothesis testing:

- $H_0 : p \geq .90$
- $H_1 : p < .90$ where $p$ is the proportion of the orders that are delivered within 10 minutes.

We can also conclude that sample size is $n = 100$ and the observed value for $X = 92$.

```
[57]: stat, pvalue=proportions_ztest(count=82,nobs=100,value=.9,alternative='smaller')
```

```
[58]: stat
```

```
[58]: -2.0823168251814157
```

```
[59]: pvalue
```

```
[59]: 0.018656770036782476
```

The pvalue is computed as follows:

```
[60]: norm.cdf(stat,0,1)
```

```
[60]: 0.018656770036782476
```

**Example:** Of a sample of 361 owners of retail service and business firms that had gone into bankruptcy, 105 reported having no professional assistance prior to opening the business. It is claimed that at most 25% of all members of this population had no professional assistance before opening the business. Test the aforementioned claim at $\alpha = 0.01$

We're testing in this example the following hypothesis:

- $H_0 \,:\, p \geq .25$
- $H_1,:, p <.25 $ where $p$ is the proportion of the bankrupted retail service owners who claim that they have no professional assistance.

We can also conclude that sample size is =361 and the observed value for $X = 105$.

```
[61]: stat, pvalue=proportions_ztest(count=105,nobs=361,value=.
      ↪25,alternative='smaller')
```

```
[62]: stat
```

```
[62]: 1.709349971523915
```

```
[63]: pvalue
```

```
[63]: 0.9563069289956099
```

## 5 Comparing two proportions

Here we have two samples, defined by a proportion, and we want to see if we can make an assertion about whether the overall proportions of one of the underlying populations is greater than / less than / different to the other.

**Example:** we want to compare two different populations to see how their tests relate to each other:

- We have two samples A and B. Our null hypothesis is that the proportions from the two populations are the same

$$H_0 \,:\, p_A = p_B$$

- Our alternative hypothesis is that the proportions from the two populations are different

$$H_1 \,:\, p_A \neq p_B$$

- From the population $A$ we sampled $n_A = 500$ tests and found $X_A = 410$ passed
- From the other population $B$, we sampled $n_B = 400$ tests and found $X_B = 379$ passed
- We use a 2-sample z-test to check if the sample allows us to accept or reject the null hypothesis

We will use for this test statistic the following hypothesis testing:

$$Z = \frac{\widehat{p}_A - \widehat{p}_B}{S_p}$$

Where $S_p$, called the **pooled standard error**, is computed as follows:

$$S_p = \sqrt{\widehat{p}(1-\widehat{p}) \times \left(\frac{1}{n_A} + \frac{1}{n_B}\right)}$$

and

$$\widehat{p} = \frac{n_A \widehat{p}_A + n_B \widehat{p}_B}{n_A + n_B}$$

is the pooled proportion.

Let $z_{\text{obs}}$ be the observed value of $Z$ under $H_0$. The pvalue associated to the two-sided alternative test can be computed as follows:

$$\text{pvalue} = \mathbb{P}\left(|Z| \geq |z_{\text{obs}||H_0} \text{ true}\right)$$

and the test statistic follows approximately the standard normal distribution.

In Python we proceed as follows:

[5]:

```
from statsmodels.stats.proportion import proportions_ztest
import numpy as np
from scipy.stats import norm,t
```

[6]:

```
sample_success_a, sample_size_a = (410, 500)
sample_success_b, sample_size_b = (379, 400)
```

We create Python arrays for the number of successes and for the sample sizes.

[7]:

```
successes = np.array([sample_success_a, sample_success_b])
samples = np.array([sample_size_a, sample_size_b])
```

[8]:

```
successes
```

```
[8]: array([410, 379])
```

```
[9]: samples
```

```
[9]: array([500, 400])
```

```
[10]: stat, pvalue = proportions_ztest(count=successes, nobs=samples, ␣
       ↪alternative='two-sided')
```

```
[11]: stat
```

```
[11]: -5.7802476568050825
```

```
[12]: pvalue
```

```
[12]: 7.459074060078635e-09
```

Let's see now how the test statistic and the pvalue are computed

```
[13]: p_pooled=successes.sum()/samples.sum()
      p_pooled
```

```
[13]: 0.8766666666666667
```

```
[14]: Sp=np.sqrt(p_pooled*(1-p_pooled)*(np.sum(1/samples)))
      Sp
```

```
[14]: 0.02205787841112558
```

```
[15]: zscore = (successes[0]/samples[0]-successes[1]/samples[1])/Sp
      zscore
```

```
[15]: -5.7802476568050825
```

```
[16]: stat
```

```
[16]: -5.7802476568050825
```

```
[19]: pv=2*(1-norm.cdf(np.abs(zscore),0,1))
      pv
```

```
[19]: 7.459074025106815e-09
```

```
[20]: pvalue
```

```
[20]: 7.459074060078635e-09
```

**Example:** Let's consider the `Titanic` data

[21]:
```
import numpy as np
import pandas as pd
import scipy.stats.distributions as dist
```

Importing the data

[34]:
```
df= pd.read_csv('titanic.csv')
```

[27]:
```
df.head()
```

[27]:
```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0      1
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                           Allen, Mr. William Henry    male  35.0      0

   Parch            Ticket     Fare Cabin Embarked
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
```

[28]:
```
df.shape
```

[28]:
```
(1309, 12)
```

We encode then the variable Survived

[29]:
```
df['Survived'].value_counts()
```

[29]:
```
0    815
1    494
Name: Survived, dtype: int64
```

[43]:
```
df['Survived'] = df['Survived'].map({1:'Survived',0:'Not Survived'})
```

[44]:
```
df['Survived'].value_counts()
```

[44]:
```
Not Survived    815
Survived        494
```

```
Name: Survived, dtype: int64
```

We select Sex and Survived variables.

```
[45]: df1=df[['Sex','Survived']]
```

```
[46]: df1.shape
```

```
[46]: (1309, 2)
```

```
[47]: df1=df1.dropna()
```

```
[48]: df1.shape
```

```
[48]: (1309, 2)
```

```
[73]: tab=pd.crosstab(df1.Survived,df1.Sex)
      tab
```

```
[73]: Sex           female  male
      Survived
      Not Survived      81   734
      Survived         385   109
```

```
[74]: tab=np.array(tab)
```

```
[75]: tab
```

```
[75]: array([[ 81, 734],
             [385, 109]], dtype=int64)
```

```
[76]: tab[1]
```

```
[76]: array([385, 109], dtype=int64)
```

```
[77]: counts=tab[1]
```

```
[78]: counts
```

```
[78]: array([385, 109], dtype=int64)
```

```
[83]: nobs=tab.sum(axis=0)
```

```
[84]: nobs
```

```
[84]: array([466, 843], dtype=int64)
```

Probability of surviving by Gender

```
[85]: counts/nobs
```

```
[85]: array([0.82618026, 0.12930012])
```

Comparing both probabilities

```
[87]: stat, pvalue = proportions_ztest(count=counts, nobs=nobs, ␣
       ↪alternative='two-sided')
```

```
[88]: stat
```

```
[88]: 24.905334404307723
```

```
[89]: pvalue
```

```
[89]: 6.513244629920008e-137
```

**Exercise:** Compare the surviving probability between different other groups.