Practice of statistics for data scientists

C. Guyeux

Introduction

Estimator, inferential statistics

The moment of a real random variable is an indicator of the dispersion of this variable (expectation, standard deviation...)

Estimator: function allowing to estimate a moment of a probability law.

It is used to infer certain characteristics of a total population from a sample (survey).

Inferential statistics is used to define and use such estimators.

Estimator of mean, variance

A good estimator of the population mean is the sample mean
It is unbiased, but not robust: an individual out of the ordinary will distort the estimate

On the other hand, the sample variance is a biased estimator of the population variance. The so-called empirical unbiased variance is to be preferred:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2$$

Estimator of mean, variance in python

The variance function of the statistics module (Python) is not the "usual" variance, but the unbiased empirical one: it is a statistics module. A "p" must be added as a prefix, to indicate that it is not a sample, but the complete population.

```
>>> from statistics import variance, pvariance, mean
>>> L=[1,2,3]
>>> sum([(L[i] - mean(L))**2 for i in range(n)])/n, pvariance(L)
(0.6666666666666666, 0.6666666666666666)
>>> sum([(L[i] - mean(L))**2 for i in range(n)])/(n-1), variance(L)
(1.0, 1)
>>>
>>> from statistics import stdev, pstdev
```

Confidence intervals

Confidence interval of a mean

- We have our estimator of the mean of a population, value obtained on a sample.

- We want to obtain a confidence interval where the true mean should probably be.

Confidence interval of a mean: example

We want to know the average height of 15-year-olds.

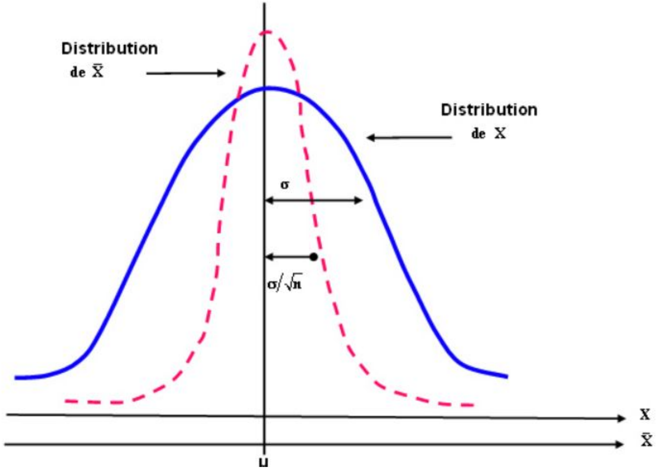
We have measured the height on a representative sample of these young people, and wish to deduce information on the whole population.

If we cannot find the exact average of the whole population, we can however provide an interval of values where it is likely to be found.

Distribution of the estimator

Let a variable follow a normal distribution $\mathcal{N}(\mu, \sigma)$

The estimator of the mean is also a random variable that follows a distribution $\mathcal{N}(\mu, \sigma/\sqrt{n})$

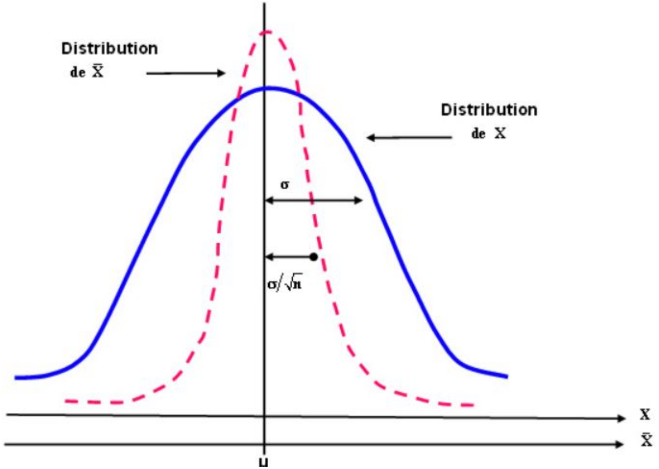


Distribution of the estimator

Let a variable follow a normal distribution $\mathcal{N}(\mu, \sigma)$

The estimator of the mean is also a random variable that follows a distribution $\mathcal{N}(\mu, \sigma/\sqrt{n})$

=> reducing the estimation error by a factor of 2 (10) requires acquiring 4 (100) times more observations in the sample



Distribution of the estimator

If the variable is normal and σ^2 known, the distribution of the mean is a Gaussian distribution $\mathcal{N}(\mu, \sigma\sqrt{n})$, regardless of the sample size n.

If the variable is normal and σ^2 unknown (estimated by s^2), which is the most frequent situation, the distribution of the mean depends on the sample size:

- If we have a large sample ($n>30$): the mean follows the law $N(\mu, s/\sqrt{n})$
- If we have a small sample: the mean follows a Student's t distribution with n-1 degrees of freedom

If the variable X is not normal but the sample size is large ($n>30$), the distribution of the mean is approximately described by :

- $N(\mu, \sigma/\sqrt{n})$ distribution if the variance is known
- and $N(\mu, s/\sqrt{n})$ if the variance is unknown, estimated by s^2.

Standard error of the sample mean (SEM): $\sigma/\sqrt{n}$, or $s/\sqrt{n}$ when estimated.

Confidence interval of a mean: python

The t-distribution approach is used when small samples (n<30) are involved:

```
>>> import scipy.stats as st
>>> from statistics import mean
>>> data = [10, 11, 10, 14, 16, 24, 10, 6, 8, 10, 11, 27, 28, 21, 13,
10, 6, 7, 8, 10]
>>> mean(data)
13
>>> st.t.interval(alpha=0.95, df=len(data)-1, loc=mean(data),
scale=st.sem(data))
(9.847638214228208, 16.15236178577179)
```

95% confidence interval of the mean of the population

Confidence interval of a mean: python

The normal distribution approach is used when larger samples (n>30) are involved:

```
>>> import numpy as np
>>> import scipy.stats as st
>>> data = np.random.randint(15, 20, 80)
>>> st.norm.interval(alpha=0.95,
...                  loc=np.mean(data),
...                  scale=st.sem(data))
(16.43912334233635, 17.06087665766365)
```

95% confidence interval of the mean of the population

General information on hypothesis testing

Hypothesis testing

An inference process that allows to check the validity of hypotheses related to one or more populations from the study of one or more random samples.

We want to determine (with a given probability) if the differences observed in the samples are either :

- attributable to chance,
- large enough to indicate that the samples are likely to be from different populations.

The null and alternative hypothesis

The null hypothesis H0 is the working hypothesis, which we wish to control.

It consists in saying that there is no difference between the compared parameters, or that the observed difference is not significant and is due to sampling fluctuations.

=> This is what we will reject (or not).

The alternative hypothesis H1: a hypothesis opposite or contrary to the null hypothesis.

Exemples of H0

We are trying to determine if the heart rate slows down during a blood draw.

H0: "Heart rate is the same before and after blood donation."

H1: "Heart rate is slower after blood donation".

We consider the education level of the pregnant women with their smoking status.

H0: "X and Y are independent".

H1: "X and Y are related".

Two types of errors are possible

Error of 1st kind: wrong rejection of H0

risk of rejecting the null hypothesis when it is true.

Error of the 2nd kind : wrong acceptance of H0

risk of accepting the null hypothesis when it is false.

Significance threshold α

H0 is chosen so that the error we are trying to avoid is the error of the first kind.

(The really interesting question is: should we reject H0?)

α: maximum first-species risk that one decides to accept.

The probability of falsely rejecting H0 must not exceed this threshold.

α is fixed by the user of the test, typically at 5%.

It must be small: we only want to reject the null hypothesis if we have enough evidence.

The value 1-α is called the confidence level of the test.

Example of the choice of H0

Not hospitalizing him when he is ill is more serious than hospitalizing him when he is not.

Hence...

H0: the patient needs to be hospitalized.
H1: the patient does not need to be hospitalized.

Power of a statistical test

=> probability of rejecting H0 right.

It is 1-β, where β is the 2nd species risk.

Test statistics

It is a function that depends on the sample and summarizes the information contained in it

Its value calculated on the sample, called the observed statistic, allows to accept or reject the null hypothesis:

- if the observed statistic belongs to a certain interval (critical zone) that depends on α, we reject H0,
- if it does not, we do not reject it.

The name of the tests

When the test statistic follows a given probability law under H0, this law is often the origin of the name of the test.

Since the same law can be found in several distinct tests, they partly have the same name:

- $\chi 2$ test of fit to a multinomial distribution, homogeneity, independence, Pearson's test;
- Fisher's F test of equality of two variances;
- Student's T test with one sample, with two independent samples...

These probability laws are often only useful for these tests.

p-value

- Rather than calculating the critical zone, we prefer to calculate a critical threshold called p-value (p), which is such that:
  - if $p < \alpha$ then we reject H0
  - otherwise we accept H0.
- It is the probability of obtaining test results at least as extreme as the result actually observed, assuming that the null hypothesis is correct.
- A very low p-value means that such an extreme observed result would be very unlikely under the null hypothesis.

p-value and rejection level

- p-value > 0.05:
  H0 is not rejected (~so it can be accepted)
- p-value < 0.05 :
  H0 is rejected
- p-value < 0.01 :
  this rejection of H0 is significant
- p-value < 0.001 :
  this rejection of H0 is very significant

Do not reject vs. accept

Why should we say "do not reject H0" rather than "accept H0"?

- If we reject H0 (at the risk of error α), it is because the observations are such that it is unlikely that H0 is true.
- If we do not reject H0, it is because we do not have sufficient criteria, no clear evidence to be able to say that H0 is false.

=> This does not mean that H0 is true.

The tests are not done to "prove" H0, but to reject it

Types of tests

If H0 corresponds to an equality pA=pB, then H1 can be...

- One-sided: an inequality pA > pB (or pA < pb)

  The question here is whether the effectiveness of treatment A is greater than that of treatment B

- Bilateral: an inequality pA ≠ pB

  The question here is whether the efficiencies are different

Types of tests

- **Paramétrique** : supposent que la population suit une distribution de probabilité (normale…).
  => cette hypothèse doit être vérifiée avant de se fier au résultat d'un test paramétrique.

- **Non-paramétrique** : ne supposent pas d'hypothèse sur la population. (Il n'y a donc pas d'hypothèses à vérifier au préalable avant de les utiliser.)

Types of tests

- Conformity test: compare a parameter calculated on the sample with a pre-established value

- Fit or adequacy test: to check the compatibility of the data with a distribution chosen a priori (normal distribution...)

- Homogeneity or comparison test: check that K samples come from the same population

- Independence or association test: to prove the existence of a link between 2 variables

Practice of hypothesis testing

Practice of hypothesis testing
*
The normality test

Normality test: Shapiro-Wilk

H0: "the sample is from a population with a Gaussian distribution".

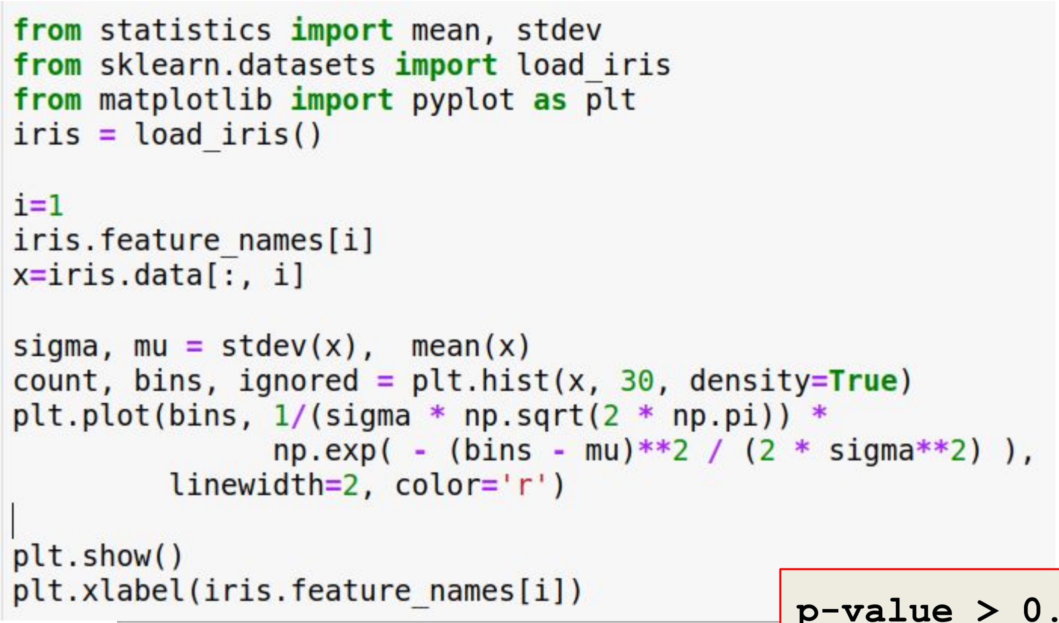
H1 : "the sample does not come from a population with a Gaussian distribution".

- p-value > 0.05 : H0 is not rejected

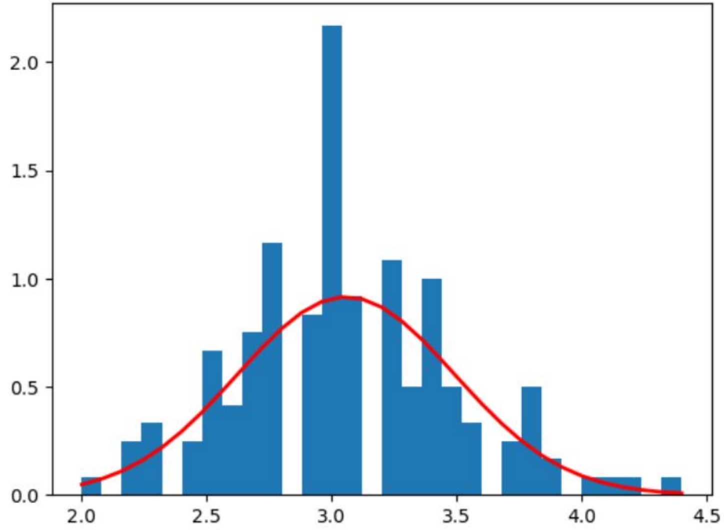
  => we can (dare to) admit normality

- p-value < 0.05 : the normality hypothesis is rejected
- p-value < 0.001 : the rejection of the normality of the data is highly significant

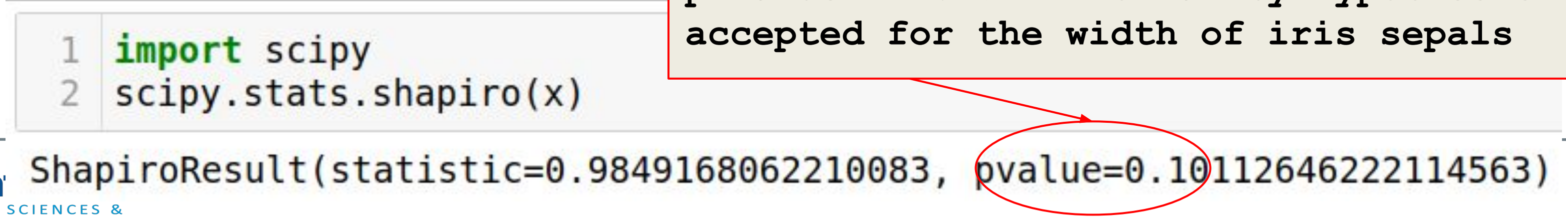
Normality test: Shapiro-Wilk



```
from statistics import mean, stdev
from sklearn.datasets import load_iris
from matplotlib import pyplot as plt
iris = load_iris()

i=1
iris.feature_names[i]
x=iris.data[:, i]

sigma, mu = stdev(x),  mean(x)
count, bins, ignored = plt.hist(x, 30, density=True)
plt.plot(bins, 1/(sigma * np.sqrt(2 * np.pi)) *
               np.exp( - (bins - mu)**2 / (2 * sigma**2) ),
         linewidth=2, color='r')

plt.show()
plt.xlabel(iris.feature_names[i])
```



p-value > 0.05 => normality hypothesis accepted for the width of iris sepals



```
import scipy
scipy.stats.shapiro(x)
```

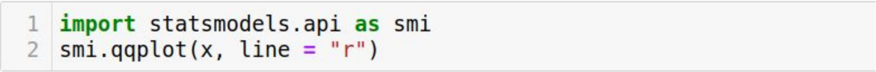
```
ShapiroResult(statistic=0.9849168062210083, pvalue=0.10112646222114563)
```

The quantile-quantile diagram

- Q-Q plot: graphical tool to evaluate the adequacy of the fit of a given distribution to a theoretical model.

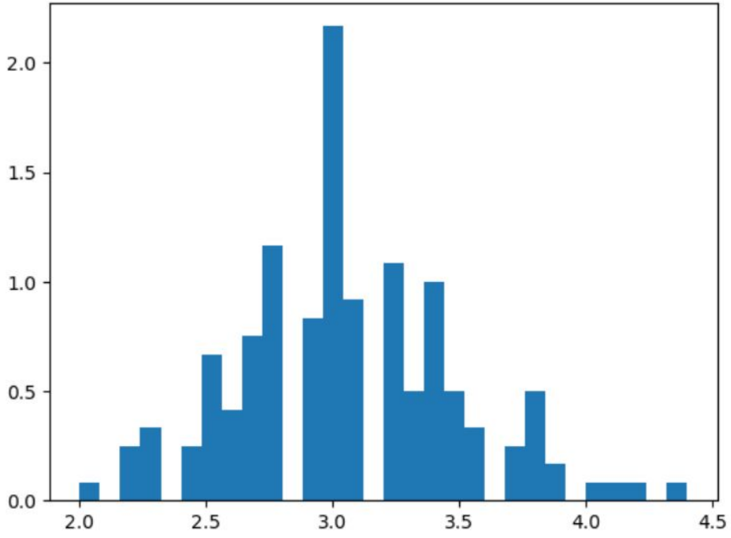
- The position of some quantiles in the observed population is compared with their position in the theoretical population

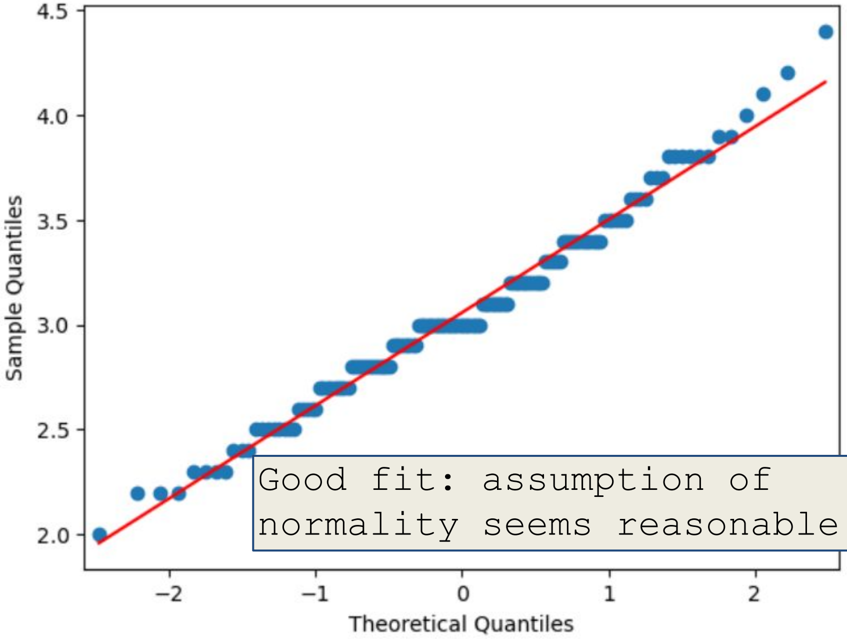
The quantile-quantile diagram



```python
import statsmodels.api as smi
smi.qqplot(x, line = "r")
```

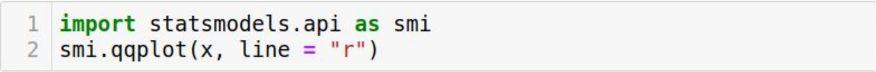
Width of iris sepals





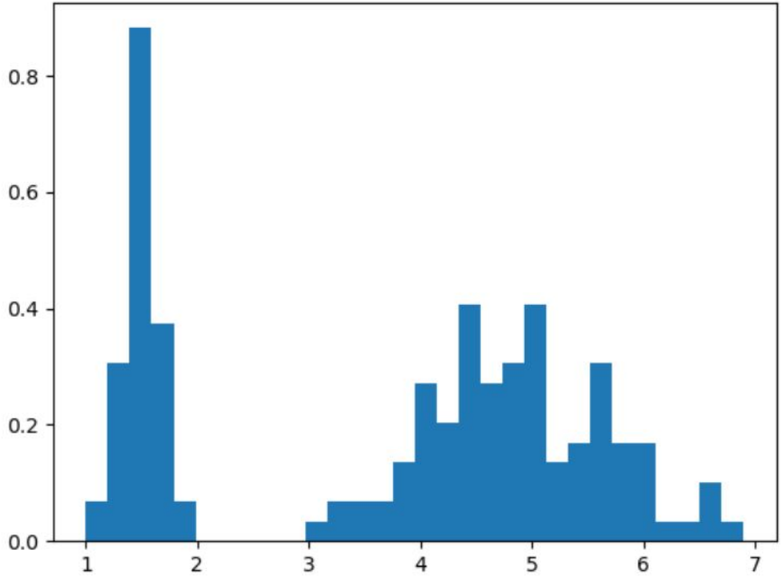
Good fit: assumption of normality seems reasonable

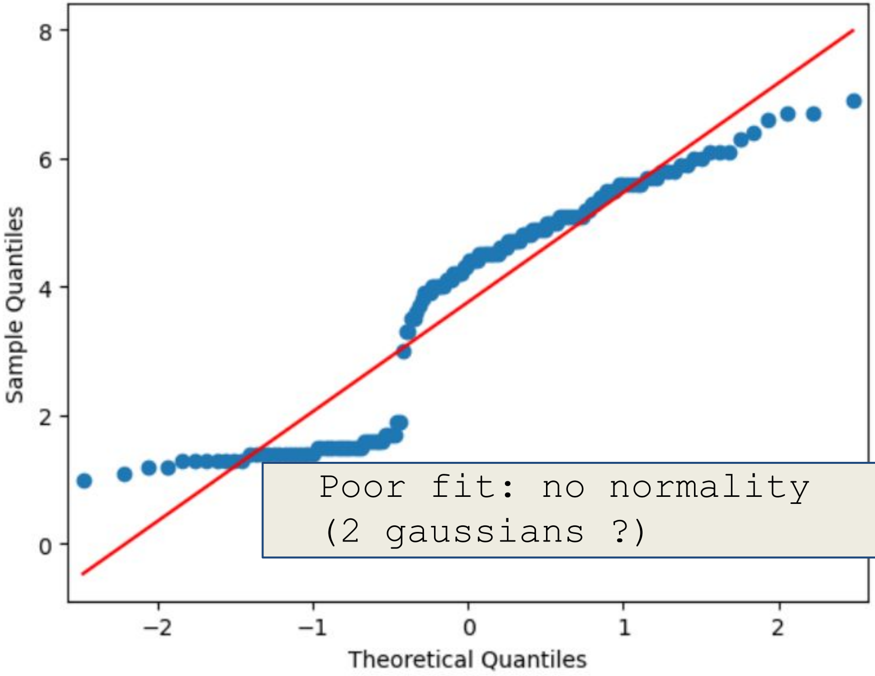
The quantile-quantile diagram



```
import statsmodels.api as smi
smi.qqplot(x, line = "r")
```

Length of the iris petals





Poor fit: no normality
(2 gaussians ?)

Practice of hypothesis testing

*

Tests of variance

Bartlett's test of equality of variances

Bartlett's test allows to estimate the null hypothesis that at least two normal populations of which we have the samples have the same variance.

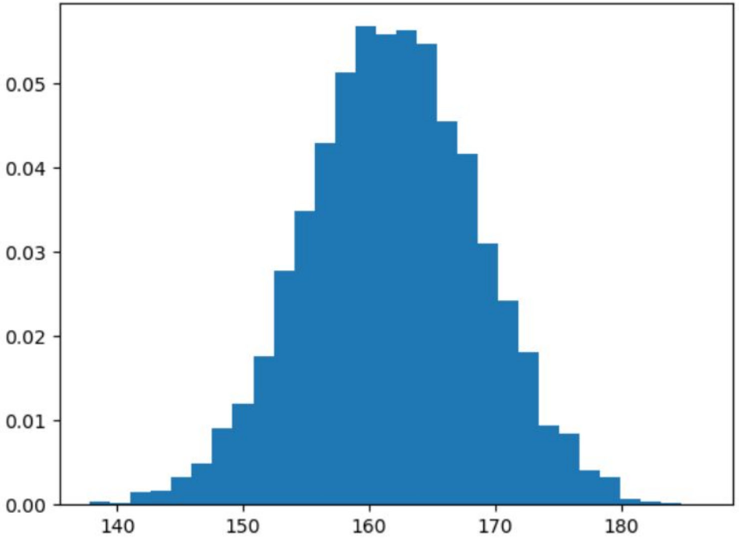
- H0: the variances of the populations are equal,
- H1: the variances of the populations are different.

(This is a parametric test.)

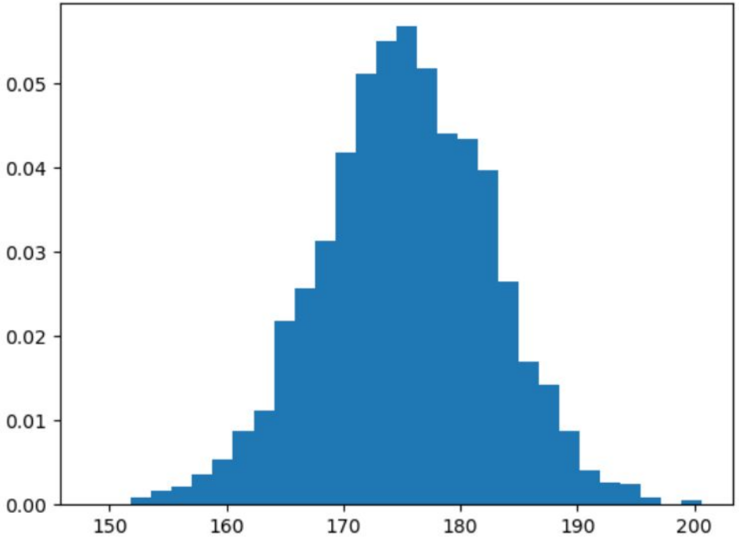
Bartlett's test generalizes Fisher's F test of equality of two variances.

Bartlett's test of equality of variances: example

Do the sizes of men and women have the same variance?



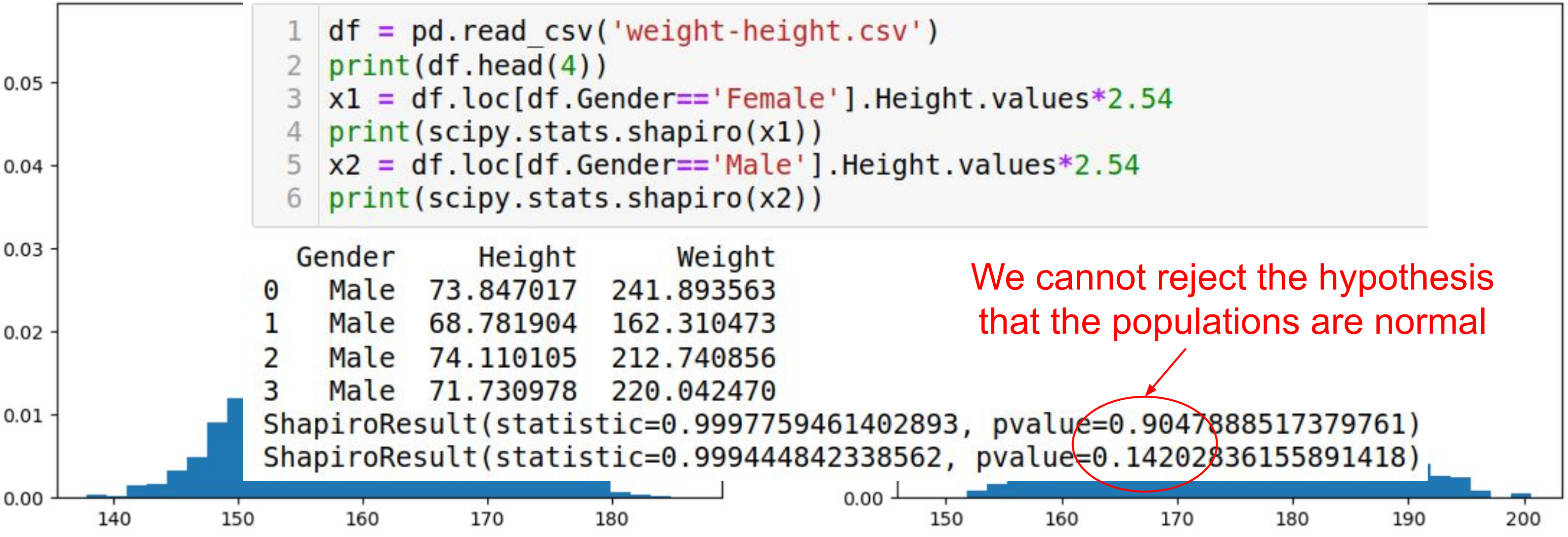
Women



Men

Bartlett's test of equality of variances: example

`Do the sizes of men and women have the same variance?`



```python
df = pd.read_csv('weight-height.csv')
print(df.head(4))
x1 = df.loc[df.Gender=='Female'].Height.values*2.54
print(scipy.stats.shapiro(x1))
x2 = df.loc[df.Gender=='Male'].Height.values*2.54
print(scipy.stats.shapiro(x2))
```

```
  Gender     Height      Weight
0   Male  73.847017  241.893563
1   Male  68.781904  162.310473
2   Male  74.110105  212.740856
3   Male  71.730978  220.042470
ShapiroResult(statistic=0.9997759461402893, pvalue=0.9047888517379761)
ShapiroResult(statistic=0.999444842338562, pvalue=0.14202836155891418)
```

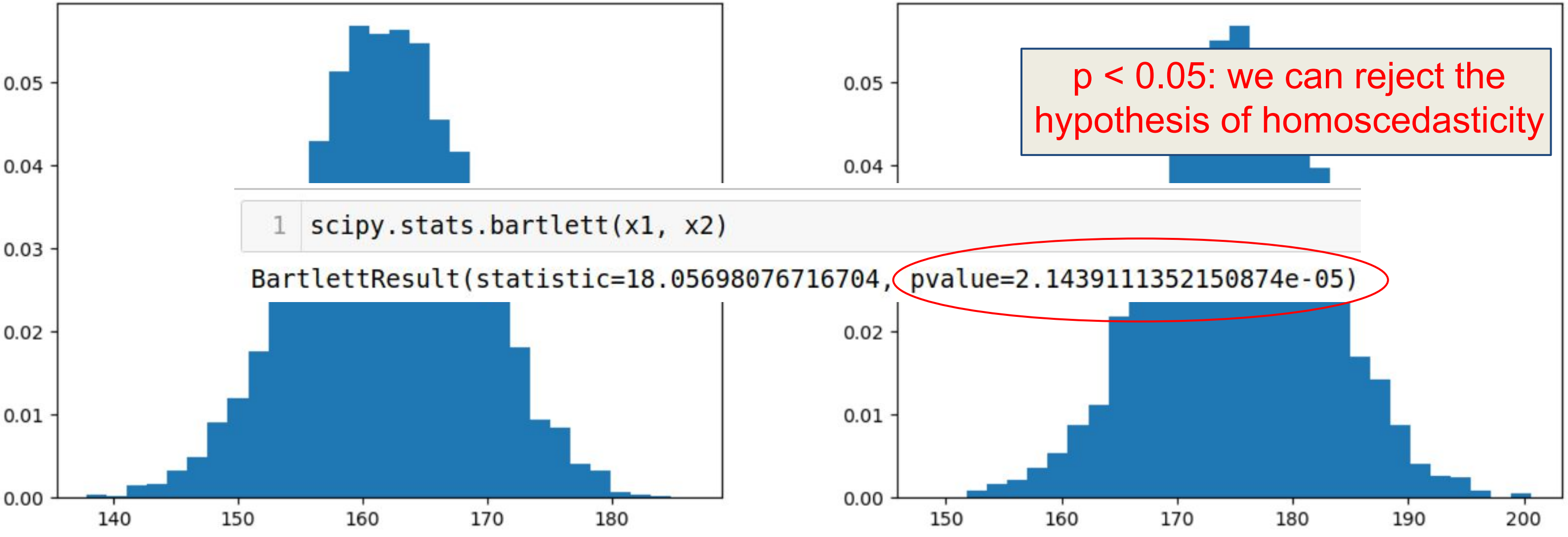
We cannot reject the hypothesis that the populations are normal

Women

Men

Bartlett's test of equality of variances: example

```
Do the sizes of men and women have the same variance?
```



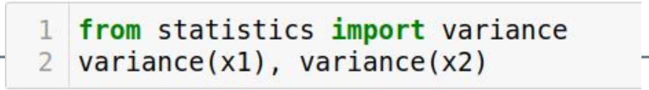
```
scipy.stats.bartlett(x1, x2)
```

```
BartlettResult(statistic=18.05698076716704, pvalue=2.1439111352150874e-05)
```

p < 0.05: we can reject the hypothesis of homoscedasticity

Women

Men



```
from statistics import variance
variance(x1), variance(x2)
```

```
(46.90279325016218, 52.895657127961044)
```

femto-st SCIENCES & TECHNOLOGIES

Levene's test of equality of variances

Levene's test allows to estimate the null hypothesis that at least two populations of which we have the samples have the same variance.

- H0: the variances of the populations are equal,
- H1: the variances of the populations are different.

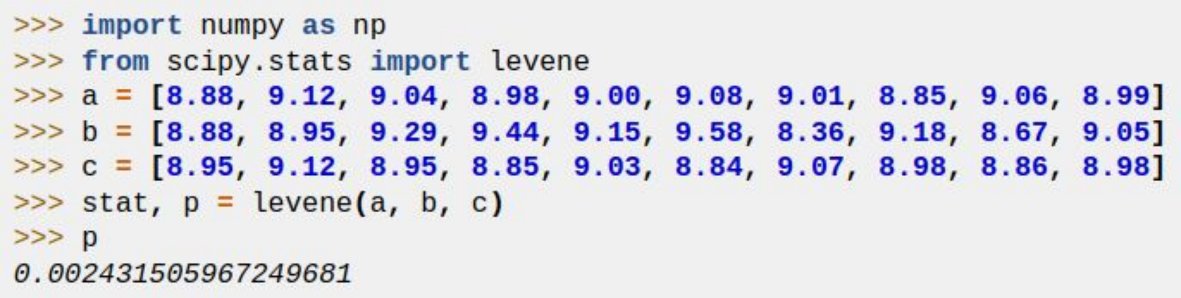
(It is no longer a parametric test).

Levene is an alternative to Bartlett in the case where there is no normality of the populations.

Levene's test of equality of variances

Levene's test allows to estimate the null hypothesis that at least two populations of which we have the samples have the same variance.

- H0: the variances of the populations are equal,
- H1: the variances of the populations are different.



```
>>> import numpy as np
>>> from scipy.stats import levene
>>> a = [8.88, 9.12, 9.04, 8.98, 9.00, 9.08, 9.01, 8.85, 9.06, 8.99]
>>> b = [8.88, 8.95, 9.29, 9.44, 9.15, 9.58, 8.36, 9.18, 8.67, 9.05]
>>> c = [8.95, 9.12, 8.95, 8.85, 9.03, 8.84, 9.07, 8.98, 8.86, 8.98]
>>> stat, p = levene(a, b, c)
>>> p
0.002431505967249681
```

=> The low p-value suggests that the populations do not have equal variances.

Practice of hypothesis testing

*

Tests of average

Student's t-test

The t-test, or Student's t-test, is used to measure the differences between the means :

- of two groups,
- or of a group compared to a standard value.

It is based on a probability law called Student's law.

Performing this test is used to understand if the differences are statistically significant (if they are not due to chance).

One sample Student's t-test

We assume that our sample is from a population with a normal distribution (Shapiro-Wilk test).

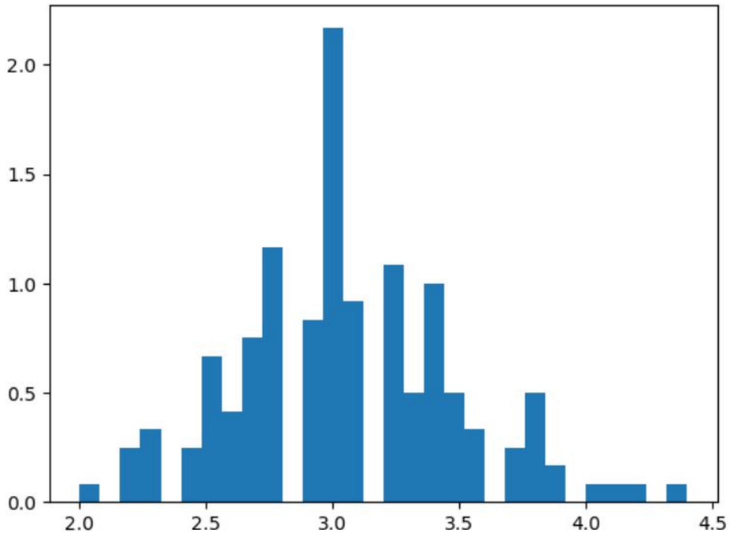
We want to test the hypothesis that the mean µ of the distribution from which we have the sample is worth a certain value m

- H0: $\mu = m$
- H1: $\mu \neq m$

e.g.: a sample of 1000 voters is asked whether they will vote right in the next election. We wonder if, from this sample, we can infer the results of the next elections (i.e. on the whole population)

One sample Student's t-test

Width of iris sepals

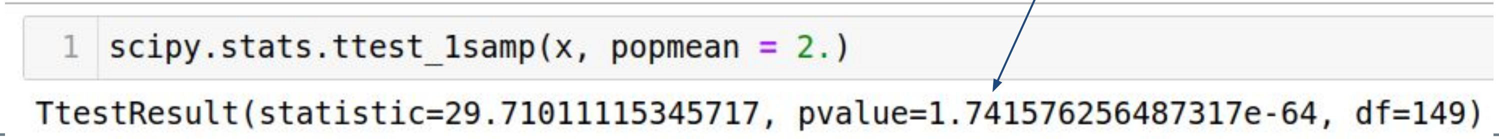


H0 : $\mu = 2$

H1 : $\mu \neq 2$

We reject H0 and this rejection is very significant:

the (normal) population of sepal widths does not have a mean of 2

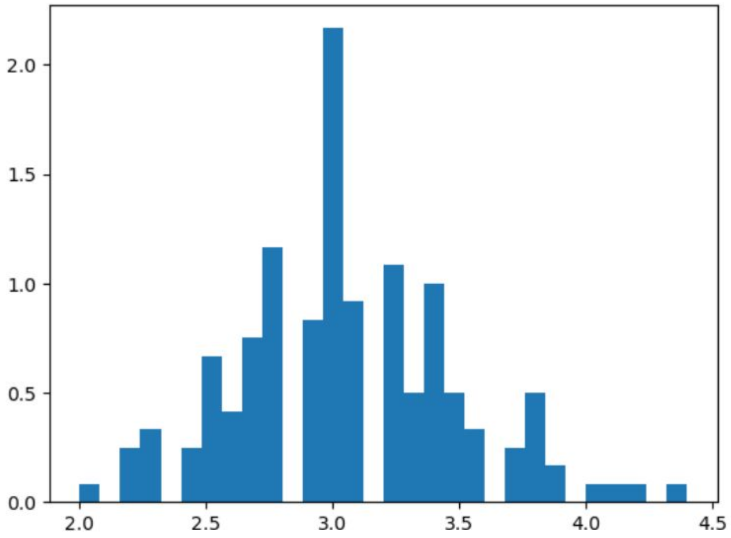


```
scipy.stats.ttest_1samp(x, popmean = 2.)
```

```
TtestResult(statistic=29.71011115345717, pvalue=1.741576256487317e-64, df=149)
```

One sample Student's t-test

Width of iris sepals

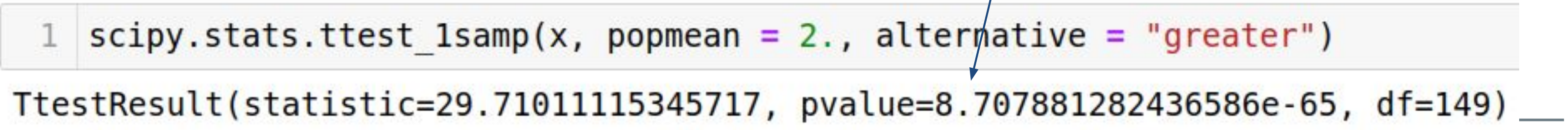


H0 : μ = 2

H1 : μ > 2

We reject H0 and this rejection is very significant

We can therefore state, with a small risk of being wrong, that the unknown mean is greater than 2



```
scipy.stats.ttest_1samp(x, popmean = 2., alternative = "greater")
```

```
TtestResult(statistic=29.71011115345717, pvalue=8.707881282436586e-65, df=149)
```

Student's t test with two independent samples

We want to test the equality of means of two populations from two independent samples:

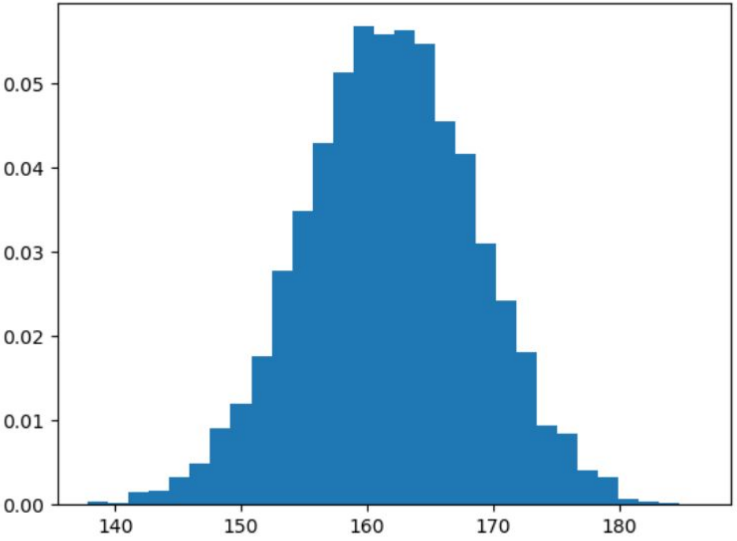
- $H0: \mu1 = \mu2$, vs.
- $H1: \mu1 \neq \mu2$,

(e.g., survival expectancy of individuals who have or have not taken a particular drug).

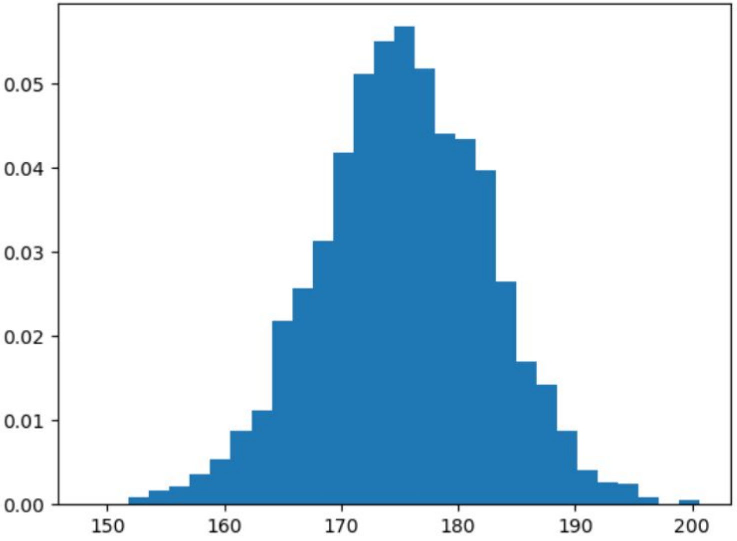
If the populations are normal and have the same variance, a two independent sample Student's t-test can be performed.

Two independent samples t-test

`Do men's and women's sizes have the same average?`



Women

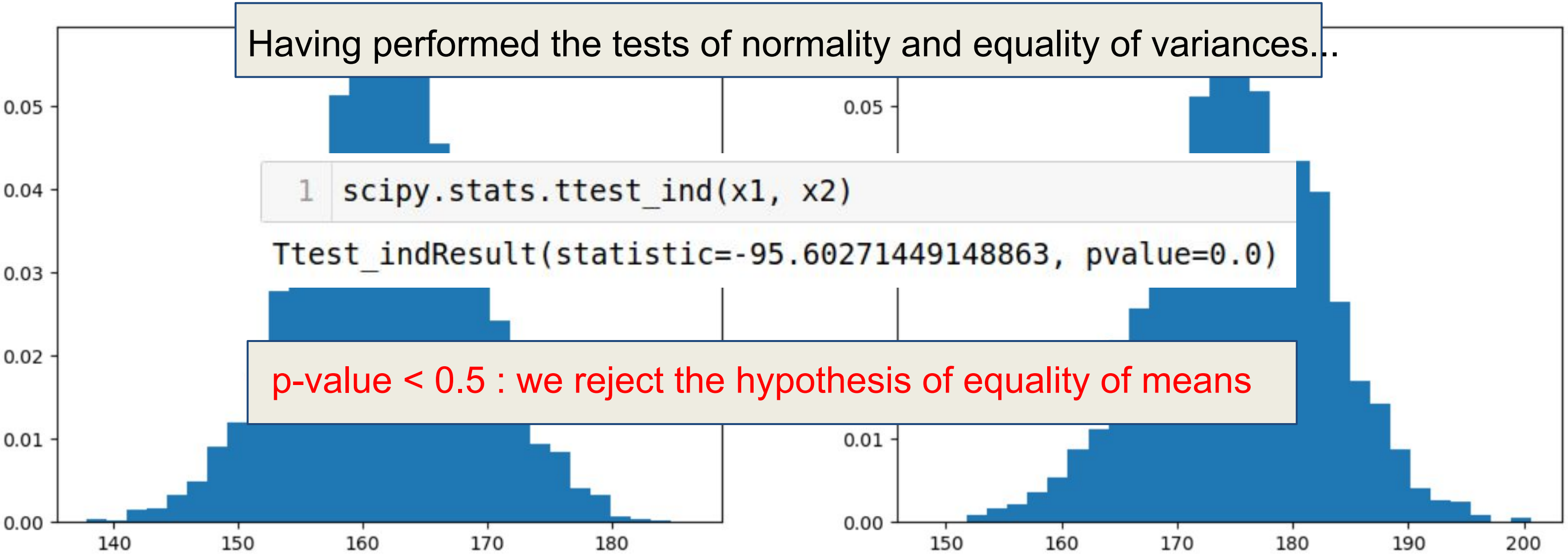


Men

Two independent samples t-test

Do men's and women's sizes have the same average?

Having performed the tests of normality and equality of variances...



```
scipy.stats.ttest_ind(x1, x2)
```

```
Ttest_indResult(statistic=-95.60271449148863, pvalue=0.0)
```

p-value < 0.5 : we reject the hypothesis of equality of means

Women

Men

Practice of hypothesis testing
*
χ2 test of independence

χ2 of independence of dichotomous factors

We want to test the independence of two (binary) characteristics in a given population.

For example, the occurrence (or not) of a given side effect following the use (or not) of a given drug.

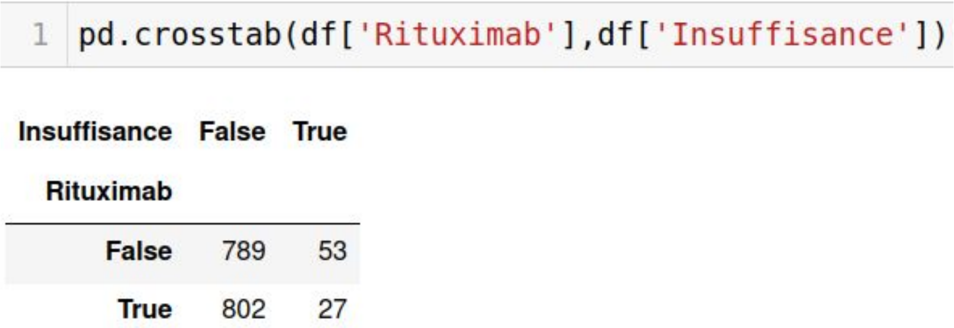
(Is there more heart failure in patients treated with Rituximab (lymphoma) than in those who are not?)

Or: having a boy as your first child, having a boy as your second child.

χ2 of independence of dichotomous factors

H0: "The two binary variables are independent"

Contingency table :

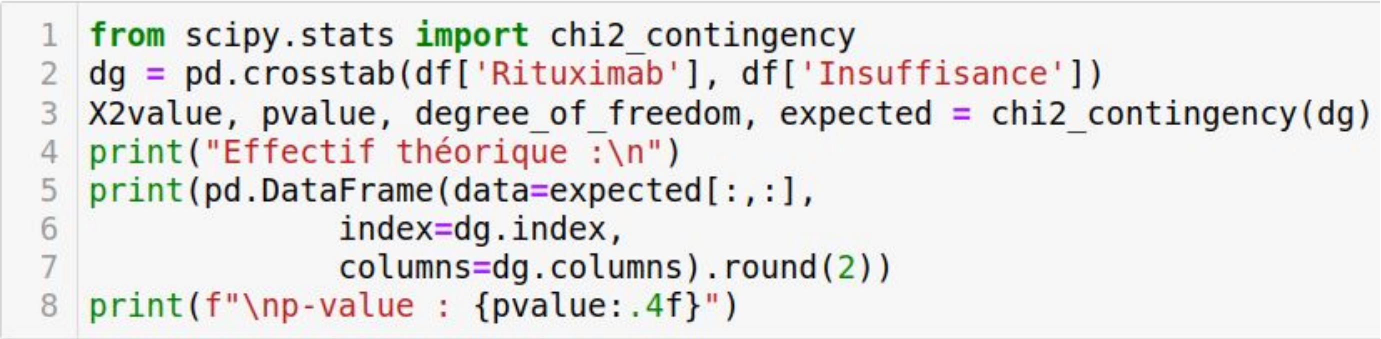


```
pd.crosstab(df['Rituximab'],df['Insuffisance'])
```

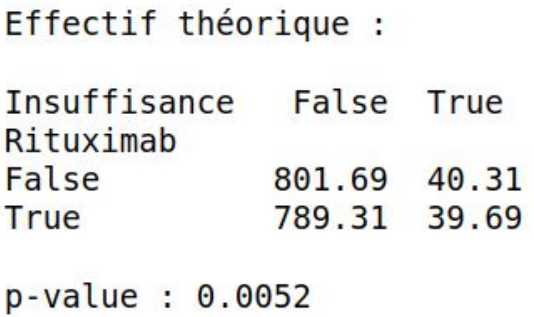
Insuffisance	False	True
Rituximab		
False	789	53
True	802	27

χ2 of independence of dichotomous factors

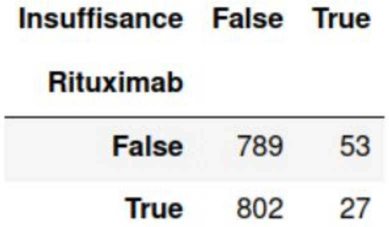
H0: "The two binary variables are independent"



```python
from scipy.stats import chi2_contingency
dg = pd.crosstab(df['Rituximab'], df['Insuffisance'])
X2value, pvalue, degree_of_freedom, expected = chi2_contingency(dg)
print("Effectif théorique :\n")
print(pd.DataFrame(data=expected[:,:],
             index=dg.index,
             columns=dg.columns).round(2))
print(f"\np-value : {pvalue:.4f}")
```



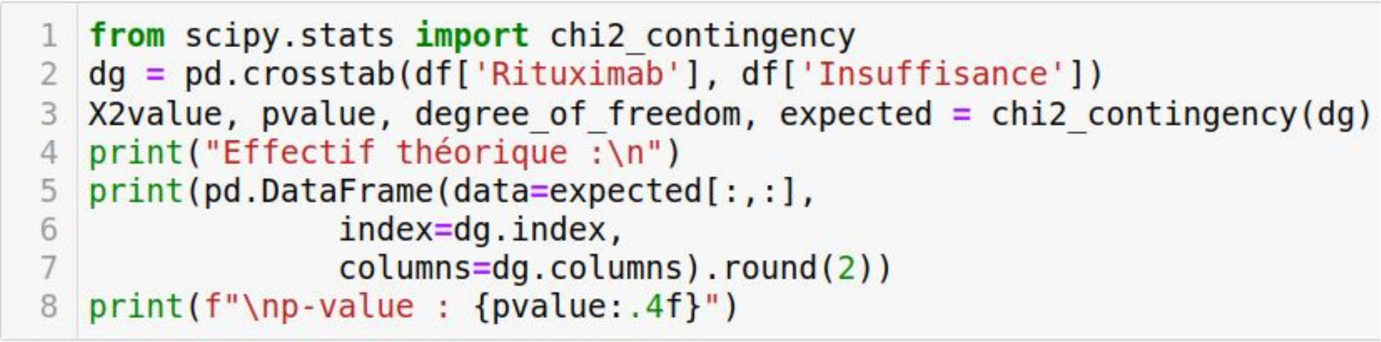
```
Effectif théorique :

Insuffisance   False   True
Rituximab
False         801.69  40.31
True          789.31  39.69

p-value : 0.0052
```



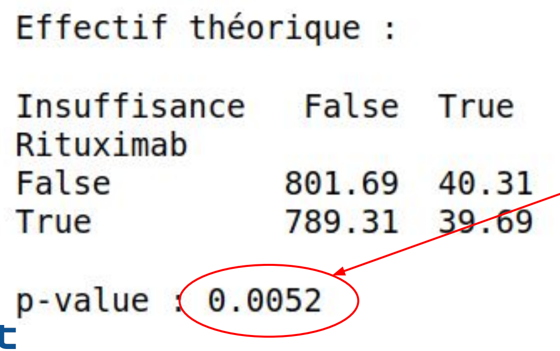
Insuffisance	False	True
Rituximab		
False	789	53
True	802	27

X2 of independence of dichotomous factors

H0: "The two binary variables are independent"

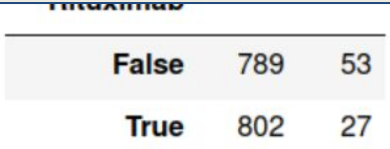


```python
from scipy.stats import chi2_contingency
dg = pd.crosstab(df['Rituximab'], df['Insuffisance'])
X2value, pvalue, degree_of_freedom, expected = chi2_contingency(dg)
print("Effectif théorique :\n")
print(pd.DataFrame(data=expected[:,:],
                   index=dg.index,
                   columns=dg.columns).round(2))
print(f"\np-value : {pvalue:.4f}")
```



```
Effectif théorique :

Insuffisance    False   True
Rituximab
False          801.69  40.31
True           789.31  39.69

p-value : 0.0052
```

Independence hypothesis rejected



Rituximab		
False	789	53
True	802	27

χ2 of independence of dichotomous factors

This test can at best establish dependency.

It does not provide any information :

- on the intensity of the relationship,
- on the causality.

Scope of application:

- total number > 30,
- the theoretical numbers are all ≥ 5.

Project

Lymphome project: objectives

- In lymphoma, patients were treated for 5 years with (or without) Rituximab (a drug), and with or without chemotherapy
  - These treatments may have introduced side effects, which had to be treated
  - If this is the case, it leads to an increase in the total cost of the treatment, and to the question of whether this money could not have been used in a better way
- You must answer the following questions:
  - what actually impacts the total cost?
  - Is the total cost significantly higher in patients treated with rituximab?
  - Are there any side effects associated with taking this drug?

Lymphome project: methodology

1. First, the data must be cleaned (don’t need to collect all features)
2. Then, various graphs and statistics must be produced to describe the data, keeping in mind the project objectives
3. Finally, use various statistical tests to provide some answers
4. All this, in a jupyter notebook, using pandas

The data: https://we.tl/t-KqHbbzEEbp

my email: cguyeux@femto-st.fr