Chương 3. Phân tích thống kê mô tả

Nội dung

- 3.1. Tổng thể (population) và mẫu (sample)
- 3.2. Thống kê mô tả
- 3.3. Mô tả bằng bảng số liệu
- 3.4. Kiểm định xem một biến có phải phân bố chuẩn
- 3.5. Kiểm định t
  - 3.5.1 Kiểm định t một mẫu - Kiểm định giá trị trung bình của một nhóm
  - 3.5.2 Kiểm định t hai mẫu - so sánh hai giá trị trung bình của hai nhóm độc lập
  - 3.5.3 Kiểm định t – so sánh hai nhóm theo cặp
- 3.6. Kiểm định phi tham số cho hai nhóm
  - 3.6.1. Kiểm định Mann-Whitney U cho hai nhóm độc lập
  - 3.6.2. Kiểm định Wilcoxon cho hai nhóm theo cặp
- 3.7. So sánh phương sai
- 3.8. Kiểm định tỷ lệ

3.1. Tổng thể và mẫu

- Nhắc lại
  - Tính toán tham số của tổng thể: đo toàn bộ mẫu trong tổng thể thường là không khả thi, ví dụ: đo chiều cao của toàn bộ dân số.
  - Ước lượng thống kê của quần thể: chọn một số mẫu mang tính đại diện, đo và tính toán thống kê.
- Như vậy, mục tiêu chính của phân tích thống kê là sử dụng thống kê (của tập mẫu) để suy luận về tham số của tổng thể.

3.2. Thống kê mô tả (Descriptive statistic, summary)

Lý thuyết	Hàm Python
Trung bình	numpy.mean()
Phương sai	numpy.var()
Độ lệch chuẩn	numpy.std()
Trung bình sai số	scipy.stats.sem()
Giá trị nhỏ nhất	numpy.min()
Giá trị lớn nhất	numpy.max()

```
df = pd.DataFrame({'categorical': pd.Categorical(['d','e','f']),
...                    'numeric': [1, 2, 3],
...                    'object': ['a', 'b', 'c']
...                   })
>>> df.describe()
       numeric
count      3.0
mean       2.0
std        1.0
min        1.0
25%        1.5
50%        2.0
75%        2.5
max        3.0
```

- Pandas: pandas.DataFrame.describe()

3.3. Mô tả bằng bảng số liệu (tabular)

- Bảng số liệu là phương tiện tóm lược dữ liệu rất hữu ích, khi muốn xem xét theo nhóm.
- Python: kết hợp groupby(), pivot() và crosstab() của pandas.DataFrame.

```
import pandas as pd
import seaborn as sns

titanic = sns.load_dataset('titanic')
bin = [0,15,100]
titanic["adult"] = pd.cut(titanic.age, bin, labels=["kid","adult"])

pd.crosstab(titanic.survived, titanic.adult, dropna=False,  margins=True)

pd.pivot_table(titanic, index = ["sex","adult"], aggfunc=np.sum)
```

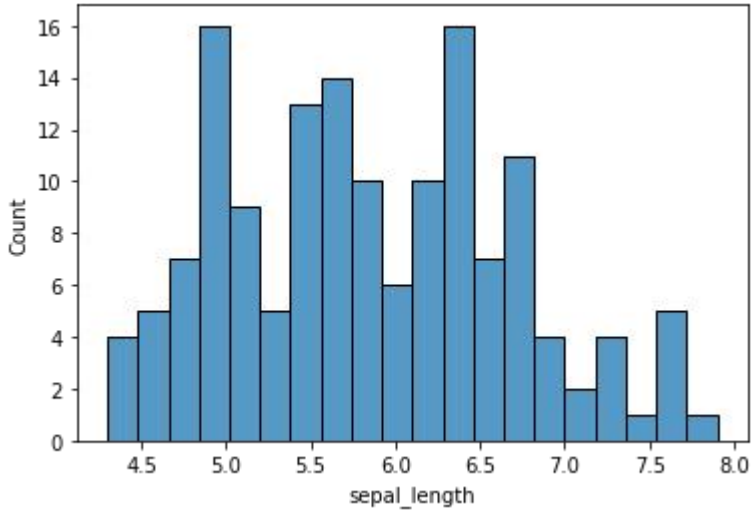
		adult_male	age	alone	fare	parch	pclass	sibsp	survived
sex	adult								
female	kid	0	310.00	5	1346.0001	51	113	55	28
	adult	0	6976.00	95	11073.1001	134	426	112	169
male	kid	0	218.67	1	1375.2209	54	105	89	21
	adult	413	13700.50	303	10977.5619	69	953	110	72

adult	kid	adult	All
survived			
0	34	390	549
1	49	241	342
All	83	631	891

3.4. Kiểm định một biến có phải phân bố chuẩn

- Trong phân tích thống kê, phần lớn phép tính dựa vào giả định biến số phải tuân theo phân bố chuẩn.
- Cần phải kiểm định xem biến có phải phân bố chuẩn hay không.
- Trong Python, sử dụng hàm scipy.stats.shapiro() để gọi Shapiro-Wilk test hoặc scipy.stats.normaltest()

```
import math
import numpy as np
from scipy.stats import shapiro

iris = sns.load_dataset('iris')
sns.histplot(iris.sepal_length, bins=20)
#perform Shapiro-Wilk test for normality
shapiro(iris.sepal_length)
```




```
ShapiroResult(statistic=0.9760897755622864, pvalue=0.010179840028285 98)
```

Vì p=0.0101 <0.05, nên ta kết luận rằng biến sepal_length không đáp ứng quy luật phân bố chuẩn

3.5. Kiểm định t

- Kiểm định t dựa vào giả thuyết phân bố chuẩn
- Có hai loại kiểm định t:
  - Kiểm định t một mẫu (one sample t-test)
  - Kiểm định t hai mẫu (two-sample t-test): trong đó có hai loại
    - Hai mẫu độc lập (independent samples t-test)
    - Nhóm theo từng cặp (paired samples t-test)

3.5.1. Kiểm định t một mẫu (One-sample t-test)

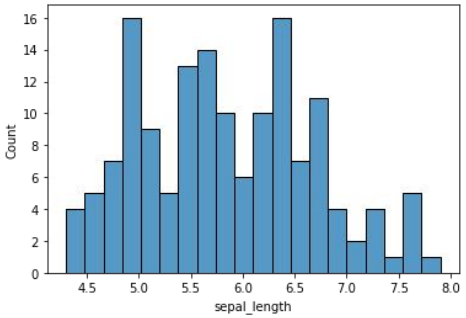
- Giả sử ta đã tính được giá trị trung bình của biến x của quần thể là $\bar{x}$, đôi lúc ta muốn kiểm định xem một giá trị μ có thể đại diện cho trung bình x hay không.
- Để trả lời câu hỏi này ta dung kiểm định t.

$$t = \frac{\bar{x} - \mu}{s/\sqrt{n}}$$

- Trong đó: $\bar{x}$ là giá trị trung bình, μ là giá trị cần kiểm định, s là độ lệch chuẩn, và n là cỡ mẫu.
- Nếu giá trị t cao hơn giá trị lý thuyết theo phân phối t ở một tiêu chuẩn có ý nghĩa thống kê (giả sử 5%) thì ta có lý do để phát biểu khác biệt có ý nghĩa thống kê.
- Python: scipy.stats.ttest_1samp

3.5.1. Kiểm định t một mẫu (One-sample t-test)

- Giả sử biết giá trị trung bình của iris.sepal_length là 5.84. Tuy nhiên, một người khác báo cáo giá trị trung bình này là 5.0. Vậy ta có thể bác bỏ giá trị này không?
  - H0: $\bar{x} = \mu = 5.0$: (giá trị trung bình của sepal_length bằng 5.0).




```
stats.ttest_1samp(iris.sepal_length, 5.0)
>>> Ttest_1sampResult(statistic=12.47325, pvalue=6.670742e-25)
```

  - => p<0.0.5: bác bỏ giả thuyết H0, tức giá trị 5.0 thấp hơn giá trị trung bình của sepal_length
- Thử test với giá trị 5.9:

```
stats.ttest_1samp(iris.sepal_length, 5.9)
>>> Ttest_1sampResult(statistic=-0.838123979992521, pvalue=0.4033053059421875)
```

  - => p>0.05, không thể bác bỏ H0, tức giá trị 5.9 có thể đại diện cho giá trị trung bình của sepal_lengh

- Ta thường có nhu cầu so sánh hai nhóm
  - VD: liệu SV Nam học giỏi toán hơn SV Nữ? Liệu SV Nữ học giỏi Anh văn hơn SV nam?
- Để trả lời cầu hỏi này, ta cần xem xét mức độ khác biệt trung bình giữa hai nhóm và độ lệch chuẩn của độ khác biệt

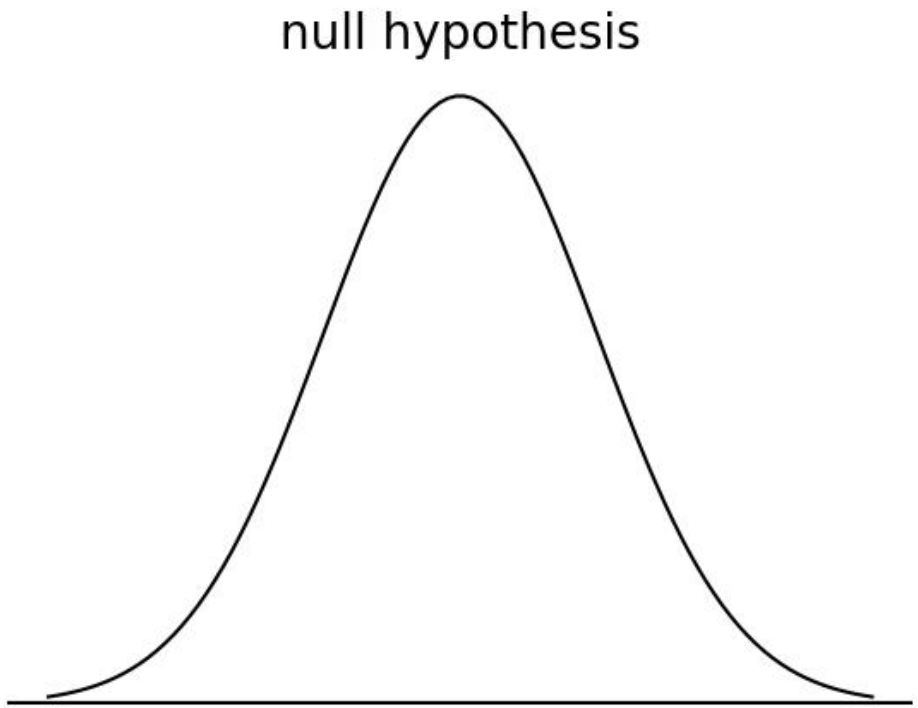
$$t = \frac{\overline{x_2} - \overline{x_1}}{SED}$$

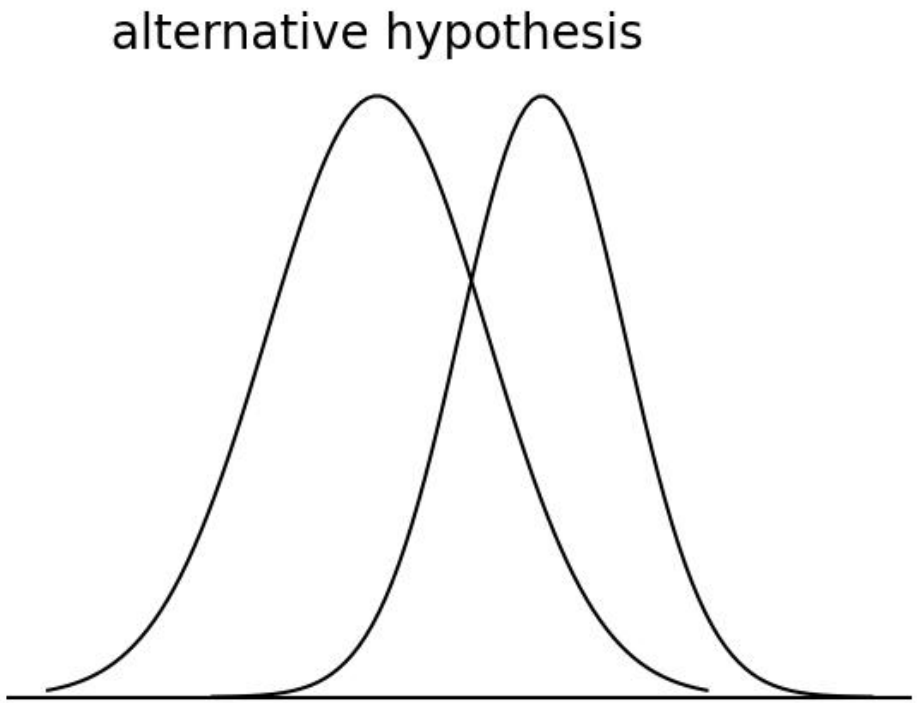
  - Trong đó: $\overline{x_1}$ và $\overline{x_2}$ là số trung bình của hai nhóm nam và nữ, SED là độ lệch chuẩn của $(\overline{x_2} - \overline{x_1})$. SED có thể ước tính bằng công thức $SED = \sqrt{SE_1^2 + SE_2^2}$, trong đó SE là sai số chuẩn của hai nhóm.
- t tuân theo luật phân bố t với bậc tự do $n_1 + n_2 - 2$, trong đó n_1 và n_2 là số mẫu của 2 nhóm.
- Trong Python, sử dụng: Scipy.stats.ttest_ind

Independent samples t-test

Giả thuyết vô hiệu giả định rằng cả hai nhóm có cùng giá trị trung bình

$$H_0: \quad \mu_1 = \mu_2$$
$$H_1: \quad \mu_1 \neq \mu_2$$





- Giả sử muốn so sánh sepal_length của 2 giống hoa lan Sentosa và Versicolor

```
setosa = iris.loc[iris.species=='setosa']
versicolor  = iris.loc[iris.species=='versicolor']

print('Mean values:', np.mean(setosa.sepal_length), np.mean(versicolor.sepal_length))
print('Variance: ',np.var(setosa.sepal_length), np.var(versicolor.sepal_length))
stats.ttest_ind(a=setosa.sepal_length, b=versicolor.sepal_length, equal_var=False)
```

```
>>> Mean values: 5.006 5.936
>>> Variance:  0.12176400000000002 0.261104
>>> Ttest_indResult(statistic=-10.52098626754911, pvalue=3.746742613983842e-17)
```

H0:?
Kết luận:?

3.5.3 Kiểm định t – so sánh hai nhóm theo cặp

- Kiểm định t cho hai nhóm độc lập không thể ứng dụng cho nghiên cứu theo thời gian, ví dụ, các chỉ số như huyết áp, mạch... trước và sau điều trị của một nhóm bệnh nhân.
- Như vậy chỉ số trước và sau điều trị trở thành “một cặp”. Điều ta cần làm là kiểm định giá trị trung bình theo cặp.

- Python: scipy.stats.ttest_rel(pre, post)

3.5.3 Kiểm định t – so sánh hai nhóm theo cặp

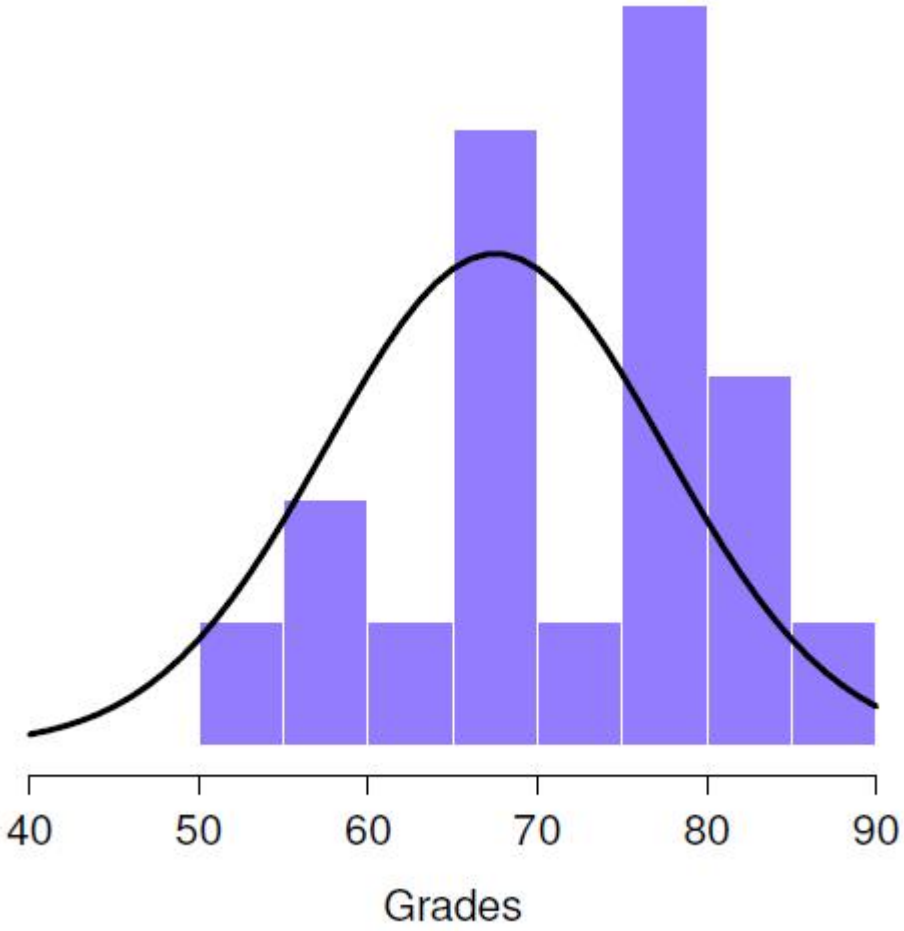
```
# Importing library
import scipy.stats as stats

# pre holds the mileage before applying the different engine oil
pre = [30, 31, 34, 40, 36, 35, 34, 30, 28, 29]

# post holds the mileage after applying the different engine oil
post = [30, 31, 32, 38, 32, 31, 32, 29, 28, 30]

# Performing the paired sample t-test
stats.ttest_rel(pre, post)
```

```
Ttest_relResult(statistic=2.584921310565987, pvalue=0.029457853822895275)
```

H0:?
Kết luận:?

3.6.2. Kiểm định phi tham số hai nhóm

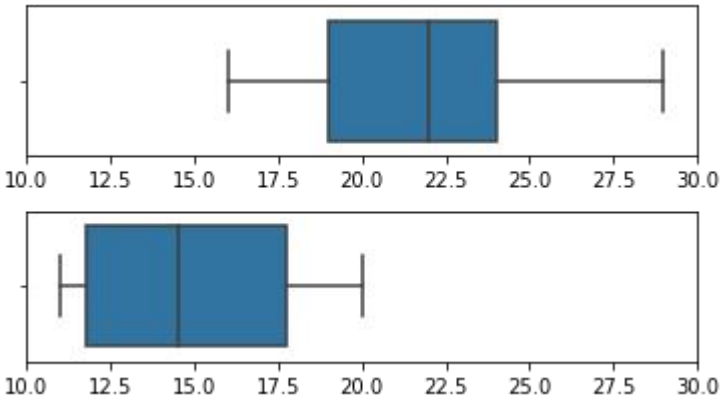
- Kiểm định t dựa vào giả thuyết một biến phải tuân theo phân phối chuẩn. Nếu không, kết quả của kiểm định t có thể không hợp lý.
- Vì thế, cần thực hiện kiểm định `Shapiro` trước khi kiểm định t.
- Nếu các biến không theo phân phối chuẩn, ta cần sử dụng phương pháp phi tham số (non-parametric) cho kiểm định.
- Kiểm định Wilcoxon không phụ thuộc vào phân phối chuẩn.



3.6.2 Kiểm định Mann-Whitney U cho hai nhóm độc lập

- Kiểm định Mann-Whitney U là một kiểm định phi tham số kiểm định giả thuyết H_0 rằng hai mẫu độc lập có có cùng phân phối hay không.
- Kiểm định Mann-Whitney U thường được dùng để kiểm định sự khác biệt về vị trí (giá trị trung bình) giữa hai phân bố.
- VD: So sánh Điểm của Nam và Nữ

```
from scipy.stats import mannwhitneyu
males = [19, 22, 16, 29, 24]
females = [20, 11, 17, 12]

U1, p = mannwhitneyu(males, females, method="exact")
print('U1={0}; p ={1}'.format( U1, p))

>>> U1=17.0; p = 0.1111111111111111
```

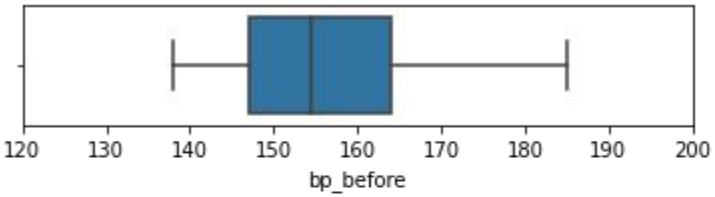


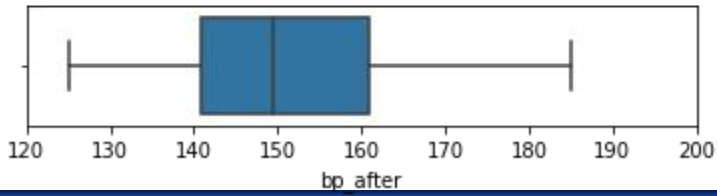
P>0.05, ta không thể bác bỏ giả thuyết H0 rằng 2 điểm của Nam và Nữ có cùng phân phối.

3.6.2 Kiểm định Wilcoxon cho hai nhóm theo cặp

- Kiểm định Wilcoxon sign-rank kiểm định giả thuyết H_0 rằng hai mẫu theo cặp có cùng sinh ra từ một phân phối hay không.
- Kiểm định Wilcoxon thường dùng cho so sánh trước sau (trong trường hợp biến không tuân theo phân phối chuẩn).

```
from scipy import stats
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("https://raw.githubusercontent.com/Opensourcefordata
science/Data-sets/master/blood_pressure.csv")

stats.wilcoxon(df['bp_before'], df['bp_after’])

>>>WilcoxonResult(statistic=2234.5, pvalue=0.0014107)
```





P<0.05, ta có thể bác bỏ giả thuyết H0 rằng 2 huyết áp của nhóm before và nhóm after là cùng phân phối

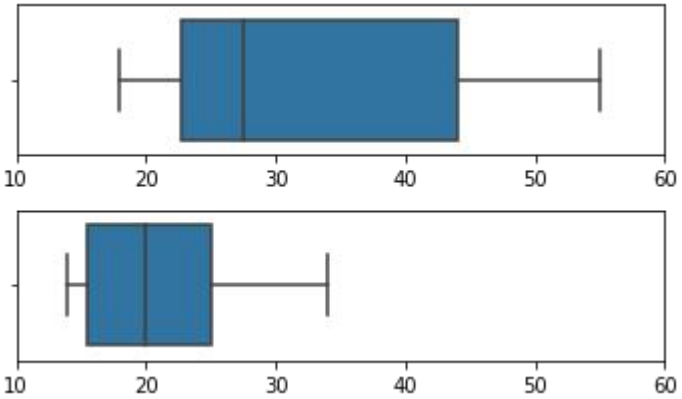
3.7. So sánh phương sai

- Đôi lúc, ta cần so sánh sự phân tán dữ liệu để đánh giá sự biến thiên hoặc độ ổn định của một biến nào đó trên hai tổng thể. Bài toán thường thấy khi so sánh chất lượng của 2 phương pháp. Khi đó ta hay xem xét tỉ lệ của hai phương sai.
- VD:
  - So sánh độ ổn định về sai số của 2 cái cân.
  - So sánh độ ổn định về kích thước sản phẩm của 2 cái máy.
- Giả thuyết vô hiệu: Không có sự khác biệt giữa hai phương sai $H_0: \frac{\sigma_1^2}{\sigma_2^2} = 1$
- Giả thuyết hữu hiệu: $H_a: \frac{\sigma_1^2}{\sigma_2^2} \neq 1$; $H_a: \frac{\sigma_1^2}{\sigma_2^2} < 1$; $H_a: \frac{\sigma_1^2}{\sigma_2^2} > 1$
- Sử dụng các kiểm định khác nhau tùy trường hợp:
  - F-test: sử dụng với giả định 2 nhóm mẫu có phân phối chuẩn.
  - Levene's test: sử dụng với giả định 2 nhóm mẫu là biến định lượng, tốt nhất cỡ mẫu dưới 20.

3.7. So sánh phương sai

- F-test

```
x = [18, 19, 22, 25, 27, 28, 41, 45, 51, 55]
y = [14, 15, 15, 17, 18, 22, 25, 25, 27, 34]

import numpy as np
import scipy.stats
#define F-test function
def f_test(x, y):
    x = np.array(x)
    y = np.array(y)
    f = np.var(x, ddof=1)/np.var(y, ddof=1) #calculate F test statistic
    print("Variance of sample 1: {}".format(np.var(x, ddof=1)))
    print("Variance of sample 2: {}".format(np.var(y, ddof=1)))
    dfn = x.size-1 #define degrees of freedom numerator
    dfd = y.size-1 #define degrees of freedom denominator
    p = 1-scipy.stats.f.cdf(f, dfn, dfd) #find p-value of F test statistic
    print("F-test:\n Statistics: {} \n p-value:{}".format(f,p))
    return f, p

#perform F-test
f_test(x, y)
```




```
Variance of sample 1: 186.988
Variance of sample 2: 42.622
F-test:
 Statistics: 4.3871
 p-value:0.019126
```

- Với $F = \frac{\sigma_1^2}{\sigma_2^2} = 4.3871$, p-value<0.05, ta có thể bác bỏ giả thuyết H0 và kết luận rằng phương sai của hai nhóm là không bằng nhau.
- Thực tế, F-value cũng cho biết tỷ lệ của hai phương sai là 4.3871

3.7. So sánh phương sai

- Levene's test:
- Ví dụ: So sánh về sai số của 3 cái cân khi cân cùng một vật thể 10 lần

```
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)
print("Levene's test:\n Statistics: {} \n p-value:{}".format(stat,p))
```

```
>>>Levene's test:
 Statistics: 7.584952754501659
 p-value:0.002431505967249681
```

- Vì p-value <0.05 nên ta kết luận phương sai của 3 phép đo là không bằng nhau