

# Statistical Tests with Scipy



---

**Carlos Alfredo Torres Cubilla**

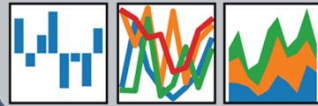
**Pedro Pablo Ropero de la Concepción**



Creación y  
manipulación  
de matrices

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



Manipulación  
de *DataFrames*

matplotlib

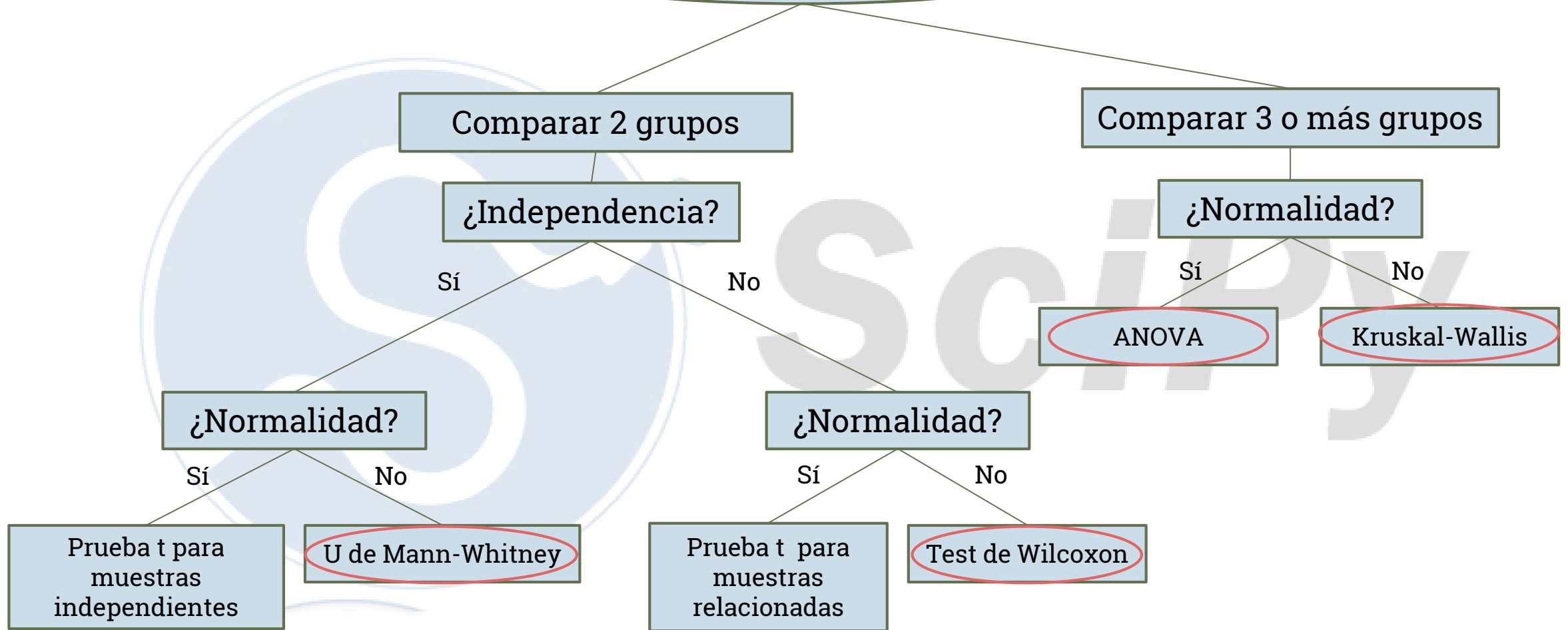
Creación de  
gráficos

Seaborn

Visualización  
de datos

Otras librerías utilizadas

# ¿Cómo elegir test?



# 1 | Test de Wilcoxon

## ¿QUÉ ES? / ¿PARA QUÉ SE UTILIZA?

Test no paramétrico utilizado para estudiar la relación entre dos variables dependientes (una variable cuantitativa y otra cualitativa).

## CONDICIONES

- Necesidad de variable ordinal y dependiente.
- No necesita una distribución específica.
- No necesita existencia de homocedasticidad.
- Preferible al t-test cuando hay valores atípicos, no hay normalidad o las muestras son pequeñas.

# 1 | Test de Wilcoxon

## CONTRASTE DE HIPÓTESIS

$$\left\{ \begin{array}{l} H_0 : Med_1 = Med_2 \\ H_1 : Med_1 \neq Med_2 \end{array} \right. \xrightarrow{\text{Se toman diferencias}} \left\{ \begin{array}{l} H_0 : Med_{dif} = 0 \\ H_1 : Med_{dif} \neq 0 \end{array} \right.$$

## EJEMPLO EXPLICATIVO

Clase 1	Clase 2	Diferencia
6	9	-3
5	6	-1
6	5	1
7	7	0
6	10	4

Se ordenan las diferencias sin tener en cuenta el signo

Si existen diferencias iguales se le asigna el rango medio

Diferencia	Rangos
0	-
1	1.5
1	1.5
3	3
4	4

# 1 | Test de Wilcoxon

## EJEMPLO EXPLICATIVO

Clase 1	Clase 2	Diferencia	Rangos
6	9	-3	3
5	6	-1	1.5
6	5	1	1.5
7	7	0	-
10	6	4	4

$$T(+) = 1,5 + 4 = 5,5$$

$$T(-) = 3 + 1,5 = 4,5$$

$$T = \min(T(+), T(-))$$

Por lo que el siguiente paso sería buscar el estadístico de contraste en la tabla de Wilcoxon, para un nivel de significación específico, y así ver si la diferencia entre clases es igual a 0 o significativamente distinta de 0..

# 1 | Test de Wilcoxon

## WILCOXON EN PYTHON

En primer lugar se importan las librerías necesarias para realizar el test:

```
import pandas as pd
import numpy as np
import scipy as sp
from scipy import stats
import math
```

Para  
Para  
Para  
Para  
Para

Posteriormente se cargan los datos :

```
w_data=pd.read_excel('Wilcoxon_Data.xlsx')
```

Se escribe la ruta que tiene que seguir el programa para encontrar el archivo

	valor	clase
28	3.4	Antes
29	2.4	Antes
...	...	...
190	6.1	Después
191	9.5	Después

# 1 | Test de Wilcoxon

## WILCOXON EN PYTHON

Se realiza el test utilizando el código:

```
W_statistic, Pvalue = sp.stats.wilcoxon(w_data['valor'][w_data['clase']=='Antes'],  
                                         w_data['valor'][w_data['clase']=='Después'],  
                                         zero_method='wilcox', correction=False)
```

Por lo tanto, tenemos el estadístico de contraste y el p-valor:

In [128]: Pvalue

Out[128]: 0.00012161475217272842

In [129]: W\_statistic

Out[129]: 1764.0

Al ser el p-valor  $< 0.05$ , se puede afirmar con tal nivel de significación que los valores que toma la variable '*valor*' dependen de la variable '*clase*'. Es decir, las muestras son significativamente distintas.



## 2 | Test U de Mann-Whitney

### ¿QUÉ ES? / ¿PARA QUÉ SE UTILIZA?

Test no paramétrico utilizado para estudiar la relación entre dos variables independientes (una variable cuantitativa y otra cualitativa).

### CONDICIONES

- Necesidad de variable ordinal e independiente.
- No necesita una distribución específica.
- Necesita existencia de homocedasticidad.
- Preferible al t-test cuando hay valores atípicos, no hay normalidad o las muestras son pequeñas.

### CONTRASTE

$$\begin{cases} H_0 : Med_1 = Med_2 \\ H_1 : Med_1 \neq Med_2 \end{cases}$$

## 2 | Test U de Mann-Whitney

### EJEMPLO EXPLICATIVO

Aplicando estos valores a las fórmulas:

Siendo  $n_1$  y  $n_2$  los tamaños muestrales de ambas poblaciones

$$\left. \begin{aligned} U_1 &= n_1 n_2 + \frac{n_1(n_1+1)}{2} - R_1 \\ U_2 &= n_1 n_2 + \frac{n_2(n_2+1)}{2} - R_2 \end{aligned} \right\} U = \min(U_1, U_2)$$

Se calcula el valor experimental  $U$ . Lo común es utilizar este valor para realizar una aproximación a una distribución normal estándar.

$$\left. \begin{aligned} U_1 &= 16 \\ U_2 &= 9 \end{aligned} \right\} U = 9$$

## 2 | Test U de Mann-Whitney

### EJEMPLO EXPLICATIVO

Pueblo 1	Pueblo 2
6	9
5	6
6	5
7	7
8	10

Se ordenan todos los valores de menor a mayor para asignarles un rango de orden.

Valor	5	5	6	6	6	7	7	8	9	10
Rango	1.5	1.5	4	4	4	6.5	6.5	8	9	10

Y sumando el rango asignado a los valores de los distintos pueblos se obtiene:

$$R_1 = 1,5 + 4 + 4 + 6,5 + 8 = 24$$

$$R_2 = 1,5 + 4 + 6,5 + 9 + 10 = 31$$

## 2 | Test U de Mann-Whitney

### EJEMPLO EXPLICATIVO

$$U = \min(U_1, U_2) \longrightarrow$$

Siendo:

$$m_u = n_1 n_2 / 2$$

$$\sigma_U = \sqrt{\frac{n_1 n_2 (n_1 + n_2 + 1)}{12}}$$

$$z = (U - m_U) / \sigma_U$$

Se obtiene el nuevo valor experimental Z. Comparando este valor con el estadístico de contraste (con una significación de 0.05, es 1.96) se aceptará o rechazará la hipótesis nula.

$$Z = -0,73 < 1,96$$

El valor experimental es menor que el estadístico de contraste, por lo que en este caso no hay evidencias para rechazar la hipótesis nula.

## 2 | Test U de Mann-Whitney

### MANN-WHITNEY EN PYTHON

Se cargan los datos desde la librería Scipy:

```
from sklearn.datasets import load_wine
```

linity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_diluted_wines	proline	class
19.4	107.0	2.95	2.97	0.37	1.78	4.500000	1.25	3.40	915.0	class_0
16.0	96.0	2.65	2.33	0.26	1.98	4.700000	1.04	3.59	1035.0	class_0
...	...	...	...	...	...	...	...	...	...	...
21.5	92.0	1.93	0.76	0.45	1.25	8.420000	0.55	1.62	650.0	class_1
21.5	113.0	1.41	1.39	0.34	1.14	9.400000	0.57	1.33	550.0	class_1

Test de normalidad:

```
In [116]: from scipy import stats
p_values = pd.DataFrame(stats.normaltest(wine.iloc[:, :-1]).pvalue, index = names[:-1]).T
p_values.style.applymap(lambda x: 'background-color : lightgreen' if x>=0.05 else 'background-color : salmon')
```

Out[116]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue
0	0.450497	0.00227458	0.0450865	0.954489	0.102813	7.46603e-05	0.923424	0.00352042	0.210936	0.00091475	5.65278e-11

Se quiere ver si la variable 'total\_phenols' depende de la variable 'clase'.

## 2 | Test U de Mann-Whitney

### MANN-WHITNEY EN PYTHON

Independencia: Se conoce de antemano que las muestras han sido recogidas de forma independiente.

Homocedasticidad: Se comprueba realizando el test de Levene:

```
In [120]: from scipy import stats
stats.levene(wine['total_phenols'][wine['class'] == 'class_0'],
             wine['total_phenols'][wine['class'] == 'class_1']).pvalue
```

Out[120]: 0.9630155799607799

No hay evidencias para decir que las varianzas son distintas

Por lo tanto, se realiza el test U de Mann-Whitney utilizando el código:

```
U_statistic, Pvalue = stats.mannwhitneyu(wine['total_phenols'][wine['class'] == 'class_0'],
                                          wine['total_phenols'][wine['class'] == 'class_1'])
```

```
In [134]: U_statistic
Out[134]: 44.0
```

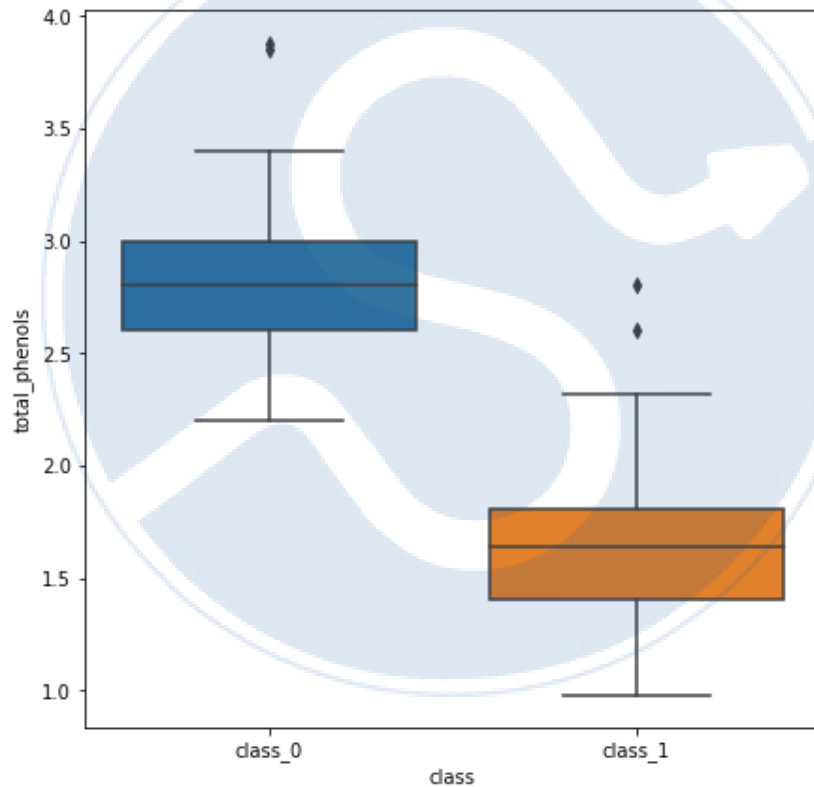
```
In [135]: Pvalue
Out[135]: 0.00012161475217272842
```

p-valor < 0.05, por lo que existen evidencias para rechazar la hipótesis nula. La variable *'total\_phenols'* depende de la variable cualitativa *'class'*

## 2 | Test U de Mann-Whitney

### MANN-WHITNEY EN PYTHON

Visualización del resultado mediante un gráfico Box-plot:



```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(7,7))
sns.boxplot(x="class", y="total_phenols", data=wine)
```

Se puede observar:

- Diferencia entre las medianas
- Los rangos intercuartílicos no se cruzan entre grupos.
- Existencia de outliers.

### 3 | Test de Kruskal-Wallis

#### ¿QUÉ ES? / ¿PARA QUÉ SE UTILIZA?


Test no paramétrico utilizado para comparar tres o más grupos independientes (una variable cuantitativa y otra cualitativa).

#### CONDICIONES

- No necesita existencia de normalidad.
- Necesita existencia de homocedasticidad.
- Misma distribución para todos los grupos.

#### CONTRASTE

$$\begin{cases} H_0 : Med_i = Med_j \\ H_1 : Med_i \neq Med_j \end{cases} \quad \forall i \neq j$$


$$\begin{cases} H_0 : \text{Todas las muestras provienen de la misma distribución} \\ H_1 : \text{Al menos una muestra proviene de una distribución distinta} \end{cases}$$



### 3 | Test de Kruskal-Wallis

#### EJEMPLO EXPLICATIVO

Grupo 1	Grupo 2	Grupo 3
6	9	10
5	10	11
6	5	12
7	7	8
8	8	14

$$R_1 = 1,5 + 3,5 + 3,5 + 5,5 + 8 = 22$$

$$R_2 = 1,5 + 5,5 + 8 + 10 + 11,5 = 36,5$$

$$R_3 = 8 + 11,5 + 13 + 14 + 15 = 61,5$$

Se ordenan todos los valores de menor a mayor para asignarles un rango de orden.

Valor	5	5	6	6	7	7	8	8	8	9	10	10	11	12	14
Rango	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Rango corregido	1.5	1.5	3.5	3.5	5.5	5.5	8	8	8	10	11.5	11.5	13	14	15

### 3 | Test de Kruskal-Wallis

#### EJEMPLO EXPLICATIVO

$$H = \frac{12}{N(N+1)} \sum_{i=1}^k \frac{R_i^2}{n_i} - 3(N+1) \longrightarrow H = 7,985$$

Si se presentan empates es necesario realizar una corrección en el estadístico de contraste calculado

$$H' = \frac{H}{1 - \frac{\sum_{j=1}^g (t_j^3 - t_j)}{N^3 - N}} \longrightarrow H' = 8,10 > 5,9915$$

Donde  $t_j$  es el número de datos empatados en el empate  $j$

El estadístico de contraste es mayor que el valor crítico, por lo que en este caso se rechaza la hipótesis nula. Al menos uno de los grupos es distinto

### 3 | Test de Kruskal-Wallis

#### KRUSKAL-WALLIS EN PYTHON

Se cargan los datos desde la librería Scipy:

```
from sklearn.datasets import load_wine
```

linity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_diluted_wines	proline	class
15.6	127.0	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065.0	class_0
11.2	100.0	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050.0	class_0
18.6	101.0	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185.0	class_0

Test de normalidad:

```
In [3]: 1 from scipy import stats
2 p_values = pd.DataFrame(stats.normaltest(wine.iloc[:, :-1]).pvalue ,index = names[: -1]).T
3 p_values.style.applymap(lambda x: 'background-color : lightgreen' if x>=0.05 else 'background-color : salmon')
```

```
Out[3]:
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity
0	0.000341632	3.17422e-06	0.036316	0.210043	4.65888e-09	0.000555667	0.000126933	0.0026647	0.00800081	5.31948e-05

Se quiere determinar si la variable '*magnesium*' depende de la variable '*class*'.

### 3 | Test de Kruskal-Wallis

#### KRUSKAL-WALLIS EN PYTHON

Independencia: Se conoce de antemano que las muestras han sido recogidas de forma independiente.

Homocedasticidad: Se comprueba realizando el test de Levene:

```
In [4]: 1 from scipy import stats
        2 stats.levene(wine['magnesium'][wine['class'] == 'class_0'],
        3                 wine['magnesium'][wine['class'] == 'class_1'],
        4                 wine['magnesium'][wine['class'] == 'class_2']).pvalue
```

Out[4]: 0.519719468148651

No hay evidencias para decir que las varianzas son distintas

Por lo tanto, se realiza el test Kruskal-Wallis utilizando el código:

```
In [17]: 1 stats.kruskal(wine['magnesium'][wine['class'] == 'class_0'],
        2                 wine['magnesium'][wine['class'] == 'class_1'],
        3                 wine['magnesium'][wine['class'] == 'class_2']).pvalue
```

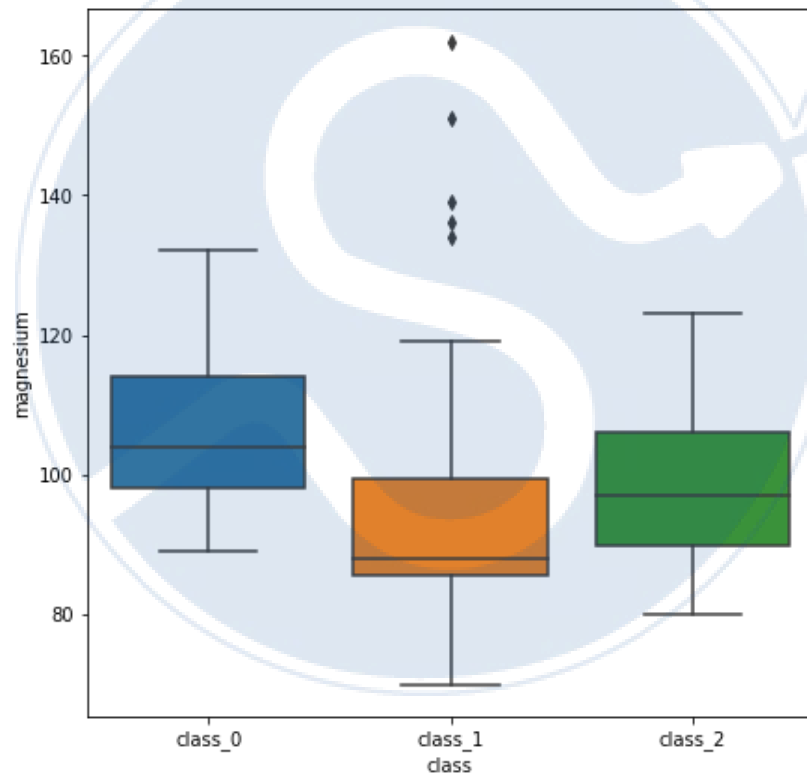
Out[17]: 1.5450460673846685e-09

p-valor < 0.05, por lo que existen evidencias para rechazar la hipótesis nula. La variable 'magnesium' depende de la variable cualitativa 'class'

### 3 | Test de Kruskal-Wallis

#### KRUSKAL-WALLIS EN PYTHON

Visualización del resultado mediante un gráfico Box-plot:



```
1 import matplotlib.pyplot as plt
2 import seaborn as sns
3
4 plt.figure(figsize=(7,7))
5 sns.boxplot(x="class", y="magnesium", data=wine)
```

Se puede observar:

- Diferencia entre las medianas
- Existencia de outliers.

## 4 | ANOVA de un factor

### ¿QUÉ ES? / ¿PARA QUÉ SE UTILIZA?

Test paramétrico utilizado para comparar tres o más grupos independientes (una variable cuantitativa y otra cualitativa).

#### CONDICIONES

- Los grupos a comparar deben ser independientes
- Asume existencia de normalidad.
- Asume presencia de homocedasticidad.

#### CONTRASTE

$$\begin{cases} H_0 : \mu_i = \mu_j \\ H_1 : \mu_i \neq \mu_j \end{cases} \quad \forall i \neq j$$

4

ANOVA de un factor

EXPLICACIÓN

La diferencia entre medias se detecta a través del estudio de la varianza entre grupos y dentro de grupos.

→

$$SS_{Total} = SS_{Treatment} + SS_{Residual}$$

*Variabilidad Total = Variabilidad explicada + variabilidad no explicada*

Fuente de variabilidad	Suma de cuadrados (SS)	Grados de libertad (df)	Cuadrados medios (MS)	F
Tratamientos	$SS_{Trat} = \sum_{i=1}^I n_i (\bar{y}_i - \bar{y})^2$	$I - 1$	$MS_{Trat} = \frac{SS_{Trat}}{I - 1}$	$F_{Trat} = \frac{MS_{Trat}}{MS_{Error}}$
Error	$SS_{Error} = \sum_{i=1}^I \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_i)^2$	$N - I$	$MS_{Error} = \frac{SS_{Error}}{I - 1}$	
Total	$SS_{Total} = \sum_{i=1}^I \sum_{j=1}^{n_i} (y_{ij} - \bar{y})^2$	$N - 1$		

↘

$$F_{Trat} \equiv F_{I-1, N-I}$$

## 4 | ANOVA de un factor

### EJEMPLO EXPLICATIVO

Método A	Método B	Método C
16	27	61
14	30	33
42	26	37
38	20	63
23	76	65

$$\begin{aligned} & \bar{y}_i \begin{cases} \bar{y}_A = 26.6 \\ \bar{y}_B = 35.8 \\ \bar{y}_C = 51.8 \end{cases} \\ & \bar{y} = 38.0667 \end{aligned}$$

Fuente de variabilidad	Suma de cuadrados	Grados de libertad	Cuadrados medios	F
Método	1626.133	2	813.067	2.651
Error	3680.800	12	306.733	
Total	5306.933	14		

El valor crítico es 3.885. Como el estadístico de contraste es menor que el valor crítico, no se detectan evidencias significativas para rechazar la hipótesis nula. Los tres métodos pueden considerarse iguales.



## 4 | ANOVA de un factor

### ANOVA EN PYTHON

Se cargan los datos desde la librería Scipy: `from sklearn.datasets import load_iris`

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	class
0	5.1	3.5	1.4	0.2	Iris-Setosa
1	4.9	3.0	1.4	0.2	Iris-Setosa
2	4.7	3.2	1.3	0.2	Iris-Setosa
3	4.6	3.1	1.5	0.2	Iris-Setosa
4	5.0	3.6	1.4	0.2	Iris-Setosa

Test de normalidad:

```
In [6]: 1 from scipy import stats
        2 p_values = pd.DataFrame(stats.normaltest(iris.iloc[:, :-1]).pvalue ,index = names[: -1]).T
        3 p_values.style.applymap(lambda x: 'background-color : lightgreen' if x>=0.05 else 'background-color : salmon')
```

```
Out[6]:
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	0.0568242	0.167241	8.67787e-49	1.99181e-30

Se quiere determinar si la variable *'sepal width'* depende de la variable *'class'*.

## 4 | ANOVA de un factor

### ANOVA EN PYTHON

Independencia: Se conoce de antemano que las muestras han sido recogidas de forma independiente.

Homocedasticidad: Se comprueba realizando el test de Levene:

```
In [8]: 1 from scipy import stats
        2 stats.levene(iris['sepal width (cm)'][iris['class'] == 'Iris-Setosa'],
        3               iris['sepal width (cm)'][iris['class'] == 'Iris-Versicolour'],
        4               iris['sepal width (cm)'][iris['class'] == 'Iris-Virginica']).pvalue
```

Out[8]: 0.5248269975064537

→ No hay evidencias para decir que las varianzas son distintas

Por lo tanto, se realiza el test ANOVA utilizando el código:

```
In [12]: 1 stats.f_oneway(iris['sepal width (cm)'][iris['class'] == 'Iris-Setosa'],
        2                  iris['sepal width (cm)'][iris['class'] == 'Iris-Versicolour'],
        3                  iris['sepal width (cm)'][iris['class'] == 'Iris-Virginica']).pvalue
```

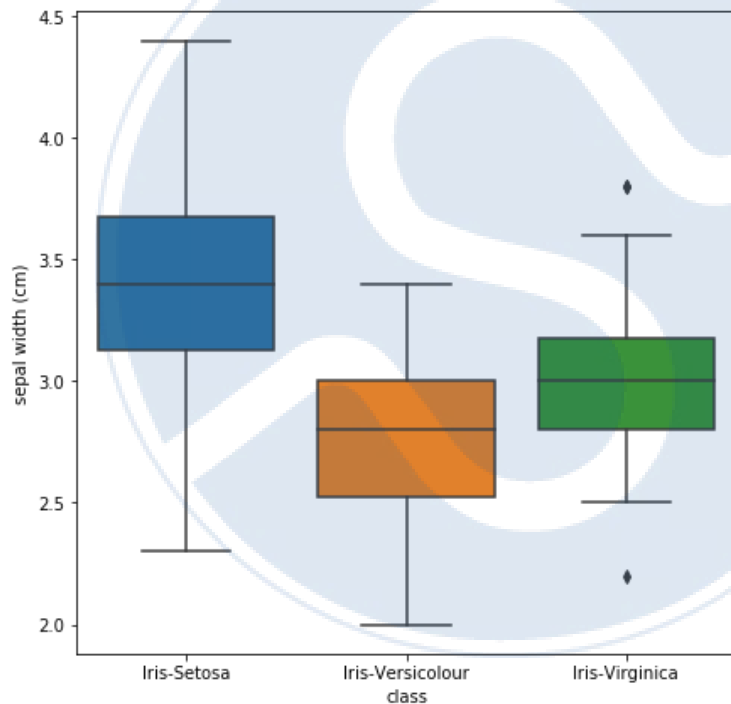
Out[12]: 1.3279165184572242e-16

→ p-valor < 0.05, por lo que existen evidencias para rechazar la hipótesis nula. La variable 'sepal width' depende de la variable cualitativa 'class'

### 3 | ANOVA de un factor

#### ANOVA EN PYTHON

Visualización del resultado mediante un gráfico Box-plot:



```
1 import matplotlib.pyplot as plt
2 import seaborn as sns
3
4 plt.figure(figsize=(7,7))
5 sns.boxplot(x="class", y="sepal width (cm)", data=iris)
```

Se puede observar:

- Diferencia entre las medianas
- Los rangos intercuartílicos no se cruzan entre grupos.
- Existencia de outliers.