**Medidas de Tendencia Central - Introducción a la estadística**

**→ PYTHON**: usaremos la [**librería “statistics”**](https://docs.python.org/3/library/statistics.html)(proporciona muchas funciones matemáticas predefinidas). Está dirigido **a nivel de gráficas** y **calculadoras científicas**.

import statistics as stats

Son compatibles para formatos **int**, **float**, **Decimal** y **Fraction**. Si los datos de entrada consta de tipo mixto, es posible que pueda usar **map()** para asegurar un resultado consistente → **map(float, input\_data)**

**Averages and measures of central location**

| [**mean()**](https://docs.python.org/3/library/statistics.html#statistics.mean) | Media aritmética ("promedio") de los datos. |
| --- | --- |
| [**fmean()**](https://docs.python.org/3/library/statistics.html#statistics.fmean) | Media aritmética de coma flotante rápida. |
| [**geometric\_mean()**](https://docs.python.org/3/library/statistics.html#statistics.geometric_mean) | Media geométrica de datos. |
| [**harmonic\_mean()**](https://docs.python.org/3/library/statistics.html#statistics.harmonic_mean) | Media armónica de datos. |
| [**median()**](https://docs.python.org/3/library/statistics.html#statistics.median) | Mediana (valor medio) de los datos. |
| [**median\_low()**](https://docs.python.org/3/library/statistics.html#statistics.median_low) | Mediana baja de datos. |
| [**median\_high()**](https://docs.python.org/3/library/statistics.html#statistics.median_high) | Alta mediana de datos. |
| [**median\_grouped()**](https://docs.python.org/3/library/statistics.html#statistics.median_grouped) | Mediana, o percentil 50, de datos agrupados. |
| [**mode()**](https://docs.python.org/3/library/statistics.html#statistics.mode) | Modo único (valor más común) de datos discretos o nominales. |
| [**multimode()**](https://docs.python.org/3/library/statistics.html#statistics.multimode) | Lista de modos (valores más comunes) de datos nominales o discretos. |
| [**quantiles()**](https://docs.python.org/3/library/statistics.html#statistics.quantiles) | Divida los datos en intervalos con la misma probabilidad. |

**Measures of spread** → de una variable (con valores dentro)

| [**pstdev()**](https://docs.python.org/3/library/statistics.html#statistics.pstdev) | Desviación estándar poblacional de los datos. |
| --- | --- |
| [**pvariance()**](https://docs.python.org/3/library/statistics.html#statistics.pvariance) | Varianza poblacional de datos. |
| [**stdev()**](https://docs.python.org/3/library/statistics.html#statistics.stdev) | Desviación estándar muestral de datos. |
| [**variance()**](https://docs.python.org/3/library/statistics.html#statistics.variance) | Varianza muestral de datos. |

**Statistics for relations between two inputs** → de varias variables (cada una con sus valores

| [**covariance()**](https://docs.python.org/3/library/statistics.html#statistics.covariance) | Covarianza muestral para dos variables. |
| --- | --- |
| [**correlation()**](https://docs.python.org/3/library/statistics.html#statistics.correlation) | Coeficiente de correlación de Pearson para dos variables. |
| [**linear\_regression()**](https://docs.python.org/3/library/statistics.html#statistics.linear_regression) | Pendiente e intersección para regresión lineal simple. |

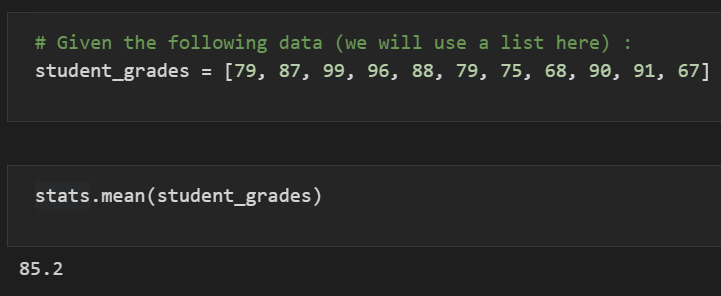
1. **MEAN:** La media o promedio de los datos es la suma de todos los valores de los de datos dividida por el número total de datos. Es importante tener en cuenta que es **sensible a valores atípicos** o valores que están muy lejos de la mayoría de los demás valores de los datos. Esto significa que la media mostrará una fluctuación relativamente grande cuando se agreguen o eliminen puntos periféricos del conjunto de datos.

import statistics as stats # `as` is a keyword that is used create an alias `stats` for the statistics library

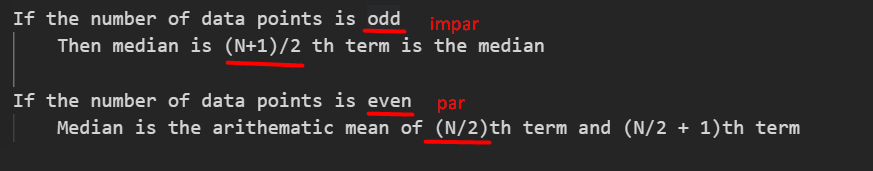
**stats.**mean(<argument>)

**stats.**median(<argument>)

→ usando un alias lo hacemos más “amigable” y así tenemos que escribir menos.



1. **MEDIAN:** es **el número del medio** cuando los datos se ordenan **en una lista ordenada**. *Si hay un número par de puntos de datos, la mediana es el promedio de los dos valores del medio*. A diferencia de la media, **la inclusión de valores atípicos no tiene un impacto muy grande** en la mediana (incluso si el valor atípico es muy grande, si el resto de los datos se distribuyen de manera más similar, agregar un valor atípico a los datos sólo moverá la mediana ligeramente a un lado de su posición anterior en el centro de los datos)



→ LAS LISTAS TIENEN QUE ESTAR ORDENADAS, aunque si ponemos directamente stats.median(list name) te devuelve el valor correcto.

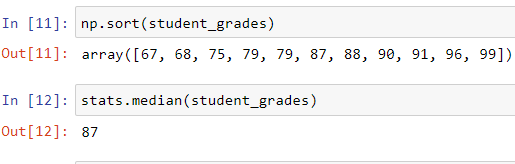
Para ordenar, podemos usar la librería **NUMPY** y el comando **sort**

student\_grades = [79, 87, 99, 96, 88, 79, 75, 68, 90, 91, 67]

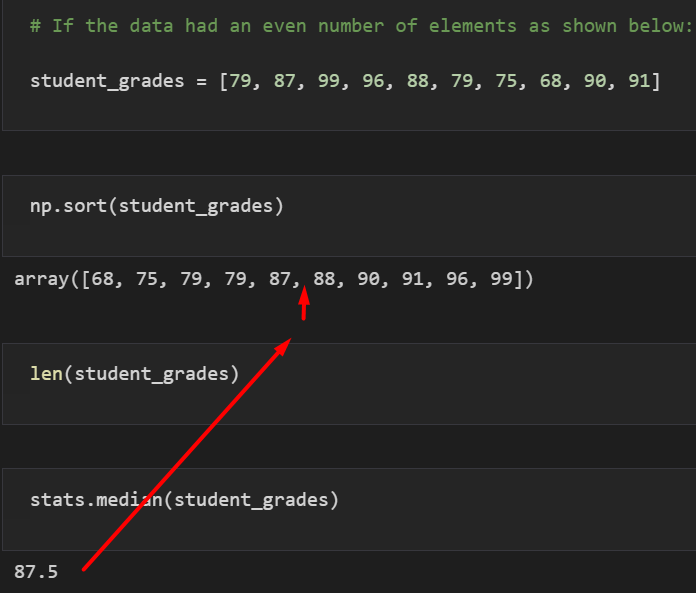
import numpy as np

np.sort(student\_grades)

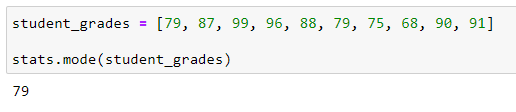
*array([68, 75, 79, 79, 87, 88, 90, 91, 96, 99])*



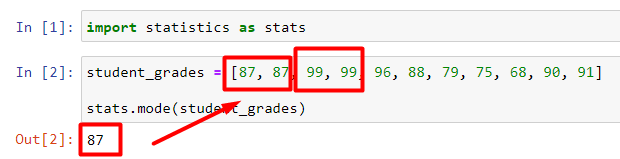
**→ Cuando** el número de puntos de **datos es par**, **la mediana se interpola tomando el promedio de los dos valores medios**:



1. **MODA:** es el **valor que aparece con mayor frecuencia** en los datos. Esta medida **se usa más para datos** categóricos (**cualitativos**), donde estaríamos más interesados en la categoría que se usa más como una forma de describir cuál es el valor "promedio" para esa categoría.



→ Si hay **varias modas** con la misma frecuencia, **devuelve el primero que se encuentra en los datos**

****

En este caso habría que usar **multimode** (devuelve una lista de los valores que ocurren con más frecuencia **en el orden en que se encontraron por primera vez**).



→ Si se desea el más pequeño o el más grande de ellos, hay que usar *min(multimode(data))* o **max(multimode(data))**.

EJERCICIO:

PseudoCode :

Define an empty dictionary. This dictionary will store a key-value pair as "Item in the list" and "Its frequency in the List"

Iterate on the list:

Find the frequency of each element using the count function

(Here is a simple demonstration : https://www.tutorialspoint.com/python3/list\_count.htm)

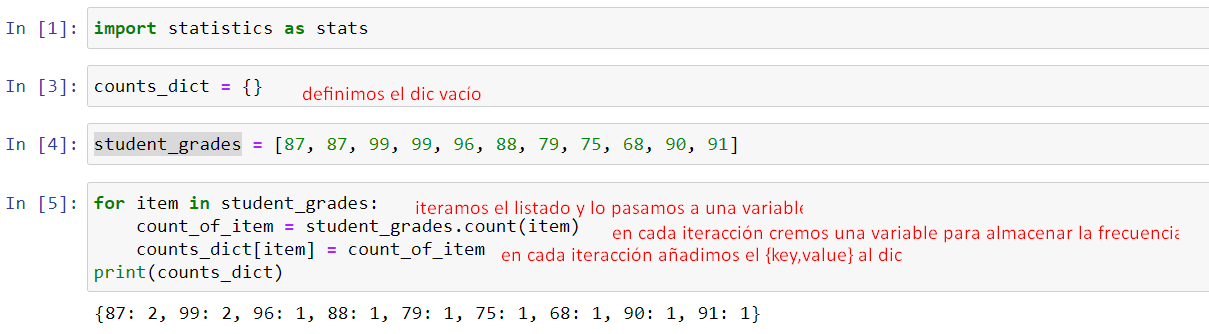
Add a new key-value pair with the element in the list as the key and frequency count as the value

Find the maximum of the values in the dictionary and store it in a new variable max\_value.

Now iterate on the items in the dictionary

if the value in the key-value pair matches the variable max\_variable:

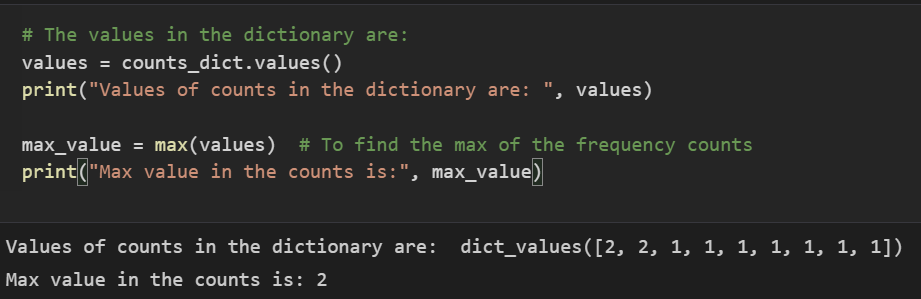
print the key for that key-value pair



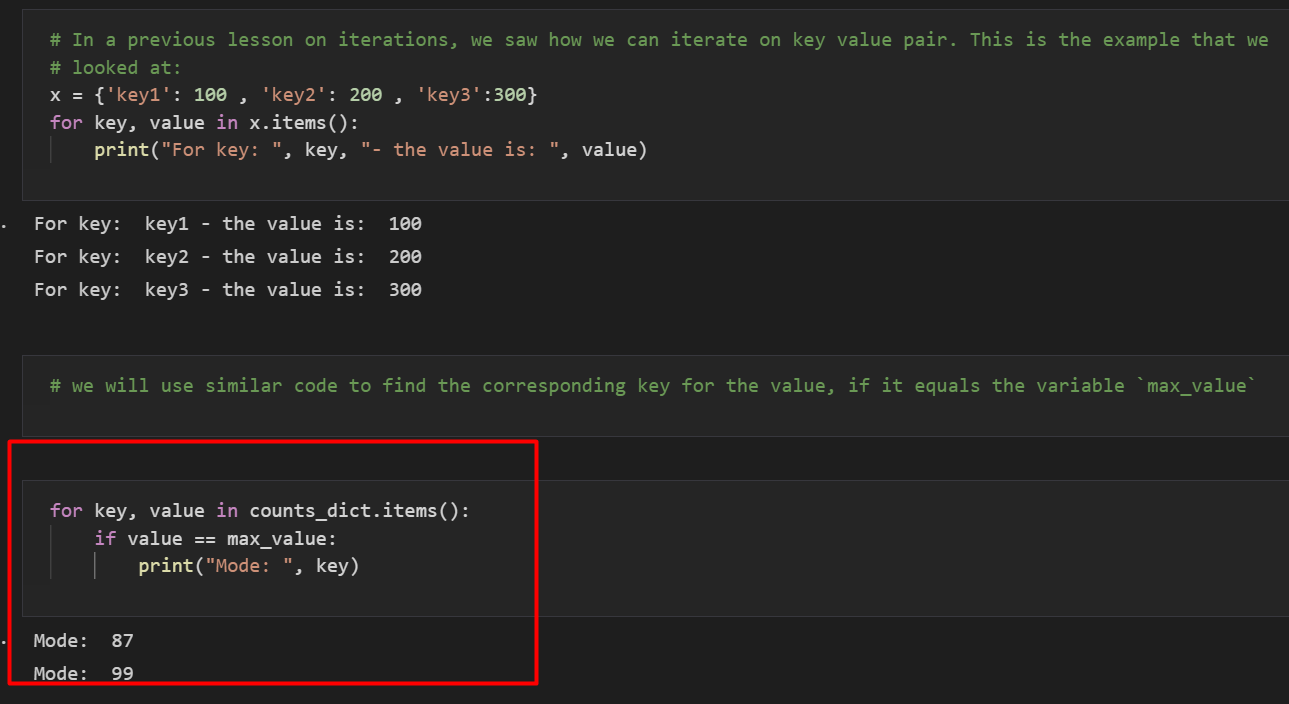
fruits = ['apple', 'banana', 'cherry']

x = fruits**.count**("cherry")

→ .**count()** es un método usado para buscar la frecuencia de una variable en una LISTA (se busca el nombre del elemento de la lista, por eso en el ejercicio anterior usábamos como nombre la variable iterada)



→ **max()** es un método usado para buscar el LISTAS DE INTEGERS (por eso en el ejercicio pasamos los valores del dic a una variable que es una lista, y hacemos el max(de esa lista))



→ como en un dic podemos buscar por el key pero no por el value, y como sabemos la moda, iteramos para decir que si la moda == al value que me pintes el nombre del key que contiene esa moda o value