Capítulo 3: Describir datos con estadísticas

### **Encontrar la moda y crear una tabla de frecuencias**

En lugar de hallar la media o la mediana de un conjunto de números, ¿qué pasaría si quisieras encontrar el número que aparece con más frecuencia? Este número se llama *moda*. Por ejemplo, considera las puntuaciones de un examen de matemáticas (sobre 10 puntos) en una clase de 20 alumnos: 7, 8, 9, 2, 10, 9, 9, 9, 9, 4, 5, 6, 1, 5, 6, 7, 8, 6, 1 y 10. La moda de esta lista te indicaría qué puntuación era la más común en la clase. En la lista puedes ver que la puntuación de 9 es la más frecuente, por lo que 9 es la moda de esta lista de números. No hay una fórmula simbólica para calcular la moda: simplemente cuentas cuántas veces aparece cada número único y encuentras el que aparece más veces.

Para escribir un programa que calcule la moda, tendremos que hacer que Python cuente cuántas veces aparece cada número dentro de una lista e imprima el que aparece con más frecuencia. La clase Counter del módulo collections, que forma parte de la biblioteca estándar, nos lo pone muy fácil.

#### ***Encontrar los elementos más comunes***

Encontrar el número más común de un conjunto de datos puede considerarse como un subproblema de encontrar un número arbitrario de números más comunes. Por ejemplo, en lugar de la puntuación más común, ¿qué pasaría si quisieras conocer las cinco puntuaciones más comunes? El método most\_common() de la clase Counter nos permite responder fácilmente a estas preguntas. Veamos un ejemplo:

>>> simplelist = [4, 2, 1, 3, 4]  
>>> from collections import Counter  
>>> c = Counter(simplelist)  
>>> c.most\_common()  
[(4, 2), (1, 1), (2, 1), (3, 1)]

Partimos de una lista de cinco números e importamos Counter del módulo de colecciones. A continuación, creamos un objeto Counter, utilizando c para referirnos al objeto. A continuación, llamamos al método most\_common(), que devuelve una lista ordenada por los elementos más comunes.

Cada miembro de la lista es una tupla. El primer elemento de la primera tupla es el número que aparece con más frecuencia, y el segundo elemento es el número de veces que aparece. La segunda, tercera y cuarta tuplas contienen los demás números junto con el recuento del número de veces que aparecen. Este resultado nos dice que el 4 es el que más veces aparece (dos veces), mientras que los demás sólo aparecen una vez. Ten en cuenta que los números que aparecen el mismo número de veces son devueltos por el método most\_common() en un orden arbitrario.

Cuando llames al método most\_common(), también puedes proporcionarle un argumento que le indique el número de elementos más comunes que quieres que devuelva. Por ejemplo, si sólo quisiéramos encontrar el elemento más común, lo llamaríamos con el argumento 1:

>>> c.most\_common(1)  
[(4, 2)]

Si vuelves a llamar al método con 2 como argumento, verás esto

>>> c.most\_common(2)  
[(4, 2), (1, 1)]

Ahora el resultado devuelto por el método most\_common es una lista con dos tuplas. La primera es el elemento más común, seguido del segundo más común. Por supuesto, en este caso, hay varios elementos empatados en el más común, por lo que el hecho de que la función devuelva aquí 1 (y no 2 ó 3) es arbitrario, como ya se ha indicado.

El método most\_common() devuelve tanto los números como el número de veces que aparecen. ¿Y si sólo queremos los números y no nos importa el número de veces que aparecen? He aquí cómo podemos recuperar esa información:

➊ >>> mode = c.most\_common(1)  
>>> mode  
[(4, 2)]  
➋ >>> mode[0]  
(4, 2)  
➌ >>> mode[0][0]  
4

En ➊, utilizamos la etiqueta mode para referirnos al resultado devuelto por el método most\_common(). Recuperamos el primer (y único) elemento de esta lista con mode[0] ➋, lo que nos da una tupla. Como sólo queremos el primer elemento de la tupla, podemos recuperarlo con mode[0][0] ➌. Esto devuelve 4- el elemento más común, o el modo.

Ahora que sabemos cómo funciona el método most\_common(), lo aplicaremos para resolver los dos problemas siguientes.

#### ***Encontrar el modo***

Ya estamos preparados para escribir un programa que encuentre el modo de una lista de números:

'''  
Calculating the mode  
'''  
  
from collections import Counter  
  
def calculate\_mode(numbers):  
➊ c = Counter(numbers)  
➋ mode = c.most\_common(1)  
➌ return mode[0][0]  
  
if \_\_name\_\_=='\_\_main\_\_':  
scores = [7,8,9,2,10,9,9,9,9,4,5,6,1,5,6,7,8,6,1,10]  
mode = calculate\_mode(scores)  
  
print('The mode of the list of numbers is: {0}'.format(mode))

La función calculate\_mode() encuentra y devuelve el modo de los números que se le pasan como parámetro. Para calcular el modo, primero importamos la clase Counter del módulo collections y la utilizamos para crear un objeto Counter en ➊. Después, en ➋, utilizamos el método most\_common(), que, como vimos antes, nos da una lista que contiene una tupla con el número más común y el número de veces que aparece. Asignamos a esa lista la etiqueta mode. Por último, utilizamos mode[0][0] ➌ para acceder al número que queremos: el número más frecuente de la lista, que es el modo.

El resto del programa aplica la función calculate\_mode a la lista de puntuaciones de exámenes que vimos antes. Cuando ejecutes el programa, deberías ver la siguiente salida:

The mode of the list of numbers is: 9

¿Qué ocurre si tienes un conjunto de datos en el que dos o más números aparecen el mismo número máximo de veces? Por ejemplo, en la lista de números 5, 5, 5, 4, 4, 4, 9, 1 y 3, tanto el 4 como el 5 están presentes tres veces. En estos casos, se dice que la lista de números tiene modos múltiples, y nuestro programa debe encontrar e imprimir todos los modos. El programa modificado es el siguiente:

'''  
Calculating the mode when the list of numbers may  
have multiple modes  
'''  
  
  
from collections import Counter  
  
  
def calculate\_mode(numbers):  
  
c = Counter(numbers)  
➊ numbers\_freq = c.most\_common()  
➋ max\_count = numbers\_freq[0][1]  
  
modes = []  
for num in numbers\_freq:  
➌ if num[1] == max\_count:  
modes.append(num[0])  
return modes  
  
if \_\_name\_\_ == '\_\_main\_\_':  
scores = [5, 5, 5, 4, 4, 4, 9, 1, 3]  
modes = calculate\_mode(scores)  
print('The mode(s) of the list of numbers are:')  
➍ for mode in modes:  
print(mode)

En ➊, en lugar de encontrar sólo el elemento más común, recuperamos todos los números y el número de veces que aparece cada uno. A continuación, en ➋, hallamos el valor de la cuenta máxima, es decir, el número máximo de veces que aparece cualquier número. Luego, para cada uno de los números, comprobamos si el número de veces que aparece es igual a la cuenta máxima ➌. Cada número que cumple esta condición es un modo, y lo añadimos a la lista modes y devolvemos la lista.

En ➍, iteramos sobre la lista devuelta por la función calculate\_mode() e imprimimos cada uno de los números.

Cuando ejecutes el programa anterior, deberías ver la siguiente salida:

The mode(s) of the list of numbers are:  
4  
5

¿Y si quisieras encontrar el número de veces que aparece cada número en lugar de sólo la moda? Una *tabla* de frecuencias, como su nombre indica, es una tabla que muestra cuántas veces aparece cada número dentro de una colección de números.

#### ***Crear una tabla de frecuencias***

Consideremos de nuevo la lista de puntuaciones del examen: 7, 8, 9, 2, 10, 9, 9, 9, 9, 4, 5, 6, 1, 5, 6, 7, 8, 6, 1 y 10. La tabla de frecuencias de esta lista se muestra en [la Tabla 3-1](ch03.html#ch3tab1). Para cada número, indicamos el número de veces que aparece en la segunda columna.

**Tabla 3-1:** Tabla de frecuencias

| **Puntuación** | **Frecuencia** |
| --- | --- |
| 1 | 2 |
| 2 | 1 |
| 4 | 1 |
| 5 | 2 |
| 6 | 3 |
| 7 | 2 |
| 8 | 2 |
| 9 | 5 |
| 10 | 2 |

Observa que la suma de las frecuencias individuales de la segunda columna suma el número total de puntuaciones (en este caso, 20).

Utilizaremos una vez más el método most\_common() para imprimir la tabla de frecuencias de un conjunto dado de números. Recuerda que cuando no proporcionamos ningún argumento al método most\_common(), éste devuelve una lista de tuplas con todos los números y el número de veces que aparecen. Podemos imprimir simplemente cada número y su frecuencia de esta lista para mostrar una tabla de frecuencias.

Éste es el programa:

'''  
Frequency table for a list of numbers  
'''  
  
from collections import Counter  
  
def frequency\_table(numbers):  
➊ table = Counter(numbers)  
print('Number\tFrequency')  
➋ for number in table.most\_common():  
print('{0}\t{1}'.format(number[0], number[1]))  
  
if \_\_name\_\_=='\_\_main\_\_':  
scores = [7,8,9,2,10,9,9,9,9,4,5,6,1,5,6,7,8,6,1,10]  
frequency\_table(scores)

La función frequency\_table() imprime la tabla de frecuencias de la lista de números que se le pasa. En ➊, primero creamos un objeto Counter y creamos la etiqueta table para referirnos a él. A continuación, utilizando un bucle for ➋, recorremos cada una de las tuplas, imprimiendo el primer miembro (el número en sí) y el segundo miembro (la frecuencia del número correspondiente). Utilizamos \t para imprimir una tabulación entre cada valor para espaciar la tabla. Cuando ejecutes el programa, verás la siguiente salida:

Number Frequency  
9 5  
6 3  
1 2  
5 2  
7 2  
8 2  
10 2  
2 1  
4 1

Aquí puedes ver que los números aparecen en orden decreciente de frecuencia porque la función most\_common() devuelve los números en este orden. Si, por el contrario, quieres que tu programa imprima la tabla de frecuencias ordenada por valor de menor a mayor, como se muestra en [la Tabla 3-1](ch03.html#ch3tab1), tendrás que volver a ordenar la lista de tuplas.

El método sort() es todo lo que necesitamos para modificar nuestro anterior programa de tabla de frecuencias:

'''  
Frequency table for a list of numbers  
Enhanced to display the table sorted by the numbers  
'''  
  
from collections import Counter  
  
def frequency\_table(numbers):  
table = Counter(numbers)  
➊ numbers\_freq = table.most\_common()  
➋ numbers\_freq.sort()  
  
print('Number\tFrequency')  
➌ for number in numbers\_freq:  
print('{0}\t{1}'.format(number[0], number[1]))  
  
if \_\_name\_\_ == '\_\_main\_\_':  
scores = [7,8,9,2,10,9,9,9,9,4,5,6,1,5,6,7,8,6,1,10]  
frequency\_table(scores)

Aquí, almacenamos la lista devuelta por el método most\_common() en numbers\_freq en ➊, y luego la ordenamos llamando al método sort() ➋. Por último, utilizamos el bucle for para repasar las tuplas ordenadas e imprimir cada número y su frecuencia ➌. Ahora, cuando ejecutes el programa, verás la siguiente tabla, que es idéntica a [la Tabla 3-1](ch03.html#ch3tab1):

Number Frequency  
1 2  
2 1  
4 1  
5 2  
6 3  
7 2  
8 2  
9 5  
10 2

En esta sección hemos tratado la media, la mediana y la moda, que son tres medidas habituales para describir una lista de números. Cada una de ellas puede ser útil, pero también pueden ocultar otros aspectos de los datos cuando se consideran aisladamente. A continuación, veremos otras medidas estadísticas más avanzadas que pueden ayudarnos a sacar más conclusiones sobre una colección de números.

[anterior](ch03_3.html)[Subtema 4 de 10: (Ver todo)](ch03.html)[siguiente](ch03_5.html)