Resumen del capítulo: Trabajar con valores ausentes y duplicados

Métricas para evaluar fuentes de tráfico

En pandas puedes realizar **operaciones matemáticas** en las columnas: suma, resta, multiplicación y división. Por ejemplo:

```
data['column1'] = data['column12'] + data['column3']
```

IDs de usuarios y cookies

El método `unique()` se usa para buscar valores únicos en una columna: `data['column'].unique()`.

Para borrar las filas con valores ausentes, llama el método `dropna()`. Para volver a numerar, llama el método `reset_index()` con el argumento `drop=True`.

`NaN` y `None`

`NaN` y `None` indican que no hay valores en una celda. `NaN` significa "no es un número" ("not a number"), así que puedes realizar en él operaciones matemáticas. `None` es "ningún tipo", lo que significa que no puedes realizar operaciones matemáticas en él. Los valores `NaN` pueden llevarnos a resultados incorrectos al agrupar datos. No elimines filas con estos valores: con frecuencia, los valores ausentes pueden restablecerse.

En pandas, el método `value_counts()` devuelve valores únicos y sus conteos.

El método `isnull()` devuelve una lista booleana, en donde "true" significa que un valor está ausente en la columna.

Para sustituir un valor por uno ausente, utiliza el método `fillna()` con el argumento `value`.

Variables categóricas y cuantitativas

Existen dos tipos de variables: **categóricas** y **cuantitativas**. Las variables categóricas toman sus valores de un conjunto limitado, mientras que las variables cuantitativas toman valores numéricos de un rango limitado.

Las variables también pueden ser **lógicas (booleanas)**, lo que significa que indican si una sentencia es verdadera o falsa. Si una sentencia es verdadera, la variable toma el valor 1. Si una sentencia es falsa, es 0.

Valores ausentes en variables categóricas

Cuando tratamos con valores ausentes en variables categóricas:

1. Determina si los valores ausentes presentan un patrón, es decir, si su aparición en el conjunto de datos es aleatoria o no. Si no se puede detectar una correlación con otros valores en las filas en las que aparecen (por ejemplo, en el caso de los encuestados menores de 21 años, una pregunta sobre el alcohol no tiene respuesta), entonces probablemente sean aleatorios.
2. Dependiendo del patrón, decide cómo manejarlos:
    - Si los valores están ausentes de manera aleatoria, no hay un patrón, entonces puedes remplazarlos con valores predeterminados: un string vacío, una palabra en particular. Utiliza el método `loc[]` y la indexación booleana. El método `fillna()` también podría funcionar, pero solo si los valores ausentes son `NaN` o `None`.
    - Si los valores ausentes no son aleatorios, entonces hay un patrón. Este es el caso más complicado, y en este capítulo no nos enfocamos en él.

Aplicar funciones a columnas con diccionarios y `agg()`

El método `agg()` se utiliza para aplicar funciones a columnas particulares. El nombre de la columna y las funciones mismas se registran en una estructura de datos llamada **diccionario.** Los diccionarios se componen de **claves** y **valores**. La clave es el nombre de la columna en la que se deben usar las funciones mientras que el valor es la lista de nombres de funciones.

```
{'column':['function1','function2']}
```

Cuando estás utilizando el método `agg()`, los nombres de las columnas se vuelven complejos. Para hacer referencia al resultado de usar `['function1']` en `['column']`, simplemente escríbelos uno tras otro:

```
data['column']['function1']
```

Valores ausentes en variables cuantitativas

Digamos que sabes que hay valores ausentes para las variables cuantitativas. Utiliza valores representativos (la media o mediana) para completar los vacíos.

1. Determina si tus datos tienen valores atípicos significativos.
2. Si no hay valores atípicos significativos, calcula la media de tus datos: aplica el método `mean()` a la columna o al conjunto de datos completo.
3. Si tus datos tienen valores atípicos significativos, calcula la mediana de tus datos: aplica el método `median()` a la columna o al conjunto de datos completo.
4. Remplaza los valores ausentes con la media o mediana usando el método `fillna()`.

Buscar duplicados a mano

Al analizar datos, con frecuencia te encontrarás con entradas duplicadas. Si no las identificas a tiempo, podrías terminar con conclusiones erróneas.

Hay dos formas de buscar datos duplicados.

Método 1

Podemos usar el método `duplicated()` junto con `sum()` para obtener el número de duplicados. Recuerda que si llamas `duplicated()` sin calcular el total, se mostrará cada fila en la pantalla, y verás el valor `True` en donde esté un duplicado, y `False` donde no lo haya.

Método 2

Llama al método `value_counts()`. Este método analiza una columna, selecciona todos los valores únicos, y después calcula con qué frecuencia aparecen. Podemos aplicar este método a los objetos Series para obtener listas de pares valor-frecuencia en orden descendiente. Las entradas que se duplican con más frecuencia se encuentran en la parte superior de la lista.

Buscar duplicados a mano con distinción de mayúsculas y minúsculas

Los duplicados en datos de string requieren de una atención especial. Desde el punto de vista de Python, una `'A'` mayúscula y una `'a'` minúscula son símbolos diferentes.

Para detectar entradas duplicadas como estas, podemos cambiar todos los caracteres en el string a letras minúsculas llamando el método `lower()`.

En pandas, cambiamos los caracteres a letra **minúscula** usando un método que sigue una sintaxis parecida: `str.lower()`.