

Resumen del capítulo: Validación de resultados

Ampliación de grupos

Los datos vienen en muchas formas diferentes y hay tipos de gráficos para prácticamente cualquier tipo de observación. Por ejemplo, si queremos visualizar fracciones del 100%, podemos hacer un **gráfico circular** (en el método `plot()`: `kind='pie'`). La columna que contiene datos para el diagrama se indica en el parámetro `y` del método.

```
data.plot(y='column', kind='pie');
```

Si vamos a estudiar un objeto separado que no está conectado de ninguna manera con sus vecinos en la clasificación, necesitaremos un **gráfico de barras**. Los gráficos de barras son adecuados para tales datos, ya que enfatizan la segmentación de los datos. Para trazar un gráfico de barras, usamos `plot()` y especificamos `kind='bar'`.

```
data.plot(y='column', kind='bar');
```

Sin embargo, a veces el gráfico puede ser un poco difícil de leer, ya que hay muchos objetos y solo una fracción de ellos son grandes. Es mejor combinar todos los objetos más pequeños y representarlos como una sola columna. De esa forma, no perderás los datos y podrás ilustrar la diferencia entre objetos más pequeños y más grandes.

Usamos el método `where()` para realizar un cambio de nombre selectivo. Se le pasan dos parámetros: una condición booleana y un nuevo valor. Si la condición es `True`, el valor correspondiente no se cambiará; si es `False`, el valor cambiará al nuevo valor.

```
data['column'].where(s > control_value, default_value)
```

Datos divididos en grupos

No siempre es práctico aplicar métodos estándar al trabajar con datos que dependen de ciertos valores en la columna. La forma más rápida es usar el método `groupby()` de pandas para hacer todos los bucles y segmentos por nosotros. Solo necesitamos pasarle el nombre de la columna y devolverá una secuencia de pares: valor de la columna - segmento correspondiente.

```
for column_value, column_slice in data.groupby('column'):  
    # hacer algo
```

En cada iteración del bucle, el método obtiene un nuevo par de esta agrupación. Luego se calcula el número de elementos en cada slice.

Si la operación que necesitamos realizar en grupos es un cálculo simple, se puede aplicar directamente a un resultado `groupby()`, sin el uso de un bucle.

Aplicar funciones al resultado de `groupby()` es otra forma de realizar el tipo de agregación que realiza el método `pivot_table()`.