Resumen del capítulo: relaciones entre datasets

Gráficos de dispersión

Trazar puntos de datos discretos puede ayudarnos a identificar las relaciones entre los datos. Cuando buscamos una relación entre dos parámetros, podemos usar un **gráfico de dispersión**. Llamamos a la función `plot()` y especificamos `kind='scatter`'.

```
data.plot(x='column_x', y='column_y', kind='scatter')
```

Tal gráfico permite ver la relación entre los dos valores y comprender qué puntos de datos son típicos y cuáles son anormales.

Correlación

Una limitación obvia de los gráficos de dispersión es que puede haber tantos puntos agrupados que son indistinguibles. Hay dos formas de hacer un gráfico más claro:

- hacer los puntos semitransparentes modificando el parámetro **alpha**
- trazar un gráfico binning hexagonal

El gráfico se divide en celdas y se calculan los puntos en cada celda. Entonces las celdas se colorean: cuantos más puntos hay, más denso es el color.

Para trazar dicho gráfico, pasa `hexbin` (diagrama de contenedores hexagonales) al parámetro `kind`.

El número de celdas a lo largo del eje horizontal se establece con el parámetro `gridsize`, que es similar a `bins` para `hist()`.

```
data.plot(x='column_x', y='column_y', kind='hexbin', gridsize=our_gridsize, sharex=False, grid=True)
```

Como en un histograma, este gráfico muestra la frecuencia. Sin embargo, un histograma muestra solo un valor, mientras que aquí tenemos dos. La alta frecuencia de ciertas combinaciones indica que hay tendencias claras.

A menudo, el objetivo del análisis de datos es mostrar la relación entre dos valores. La interdependencia de dos valores se conoce como **correlación**. La altura y peso están **positivamente correlacionadas** porque un incremento en una de ellas generalmente significa un incremento en la otra. Un ejemplo de **correlación negativa** sería la altura y la voz;

generalmente (de nuevo, no siempre), cuanto más alto/a seas, menor será la frecuencia de tu voz.

Una cosa es mirar el gráfico, pero también necesitamos una forma numérica para describir la correlación. Para esto tenemos el **coeficiente de correlación de Pearson**, que nos dice cuánto cambia un valor cuando el otro cambia. Toma valores desde -1 a 1.

- Si uno de los valores incrementa junto con el otro, el coeficiente de correlación de Pearson es positivo.
- Si uno permanece igual mientras el otro cambia, el coeficiente es 0.
- Si uno se reduce mientras el otro incrementa, el coeficiente es negativo.

Cuanto más cerca esté el coeficiente de -1 o 1, más fuerte será la dependencia. Un valor cercano a 0 significa que hay una conexión débil, mientras que un valor de 0 puede significar que, o no hay, o que hay una compleja conexión no lineal que el coeficiente no puede reflejar.

En pandas, el coeficiente de correlación Pearson se calcula con el método **corr()**. Se aplica a la columna que contiene el primer valor, y la columna con el segundo se pasa como un parámetro. No importa cuál es cuál.

```
print(data['column_1'].corr(data['column_2']))
print(data['column_2'].corr(data['column_1']))
```

Distribuciones conjuntas para múltiples valores

Desafortunadamente, es imposible trazar un gráfico coherente para múltiples parámetros de inmediato. Sin embargo, podemos construir gráficos de dispersión para cada posible par de parámetros. En pandas, no lo hacemos con `df.plot()`, sino con un método diferente `pd.plotting.scatter_matrix(df)`.

```
pd.plotting.scatter_matrix(data)
```

Además, podemos encontrar los coeficientes de correlación para todos los pares de parámetros. Simplemente podemos llamar al método `corr()` sin ningún parámetro. Este tipo de tabla se llama **matriz de correlación:**

```
data.corr()
```