Resumen del capítulo: primeros gráficos y conclusiones

El ejercicio en cuestión

Los datos de los archivos CSV pueden estar separados por comas, puntos y comas, caracteres de tabulación u otros símbolos. Además, ten cuidado con los decimales escritos con comas.

Los parámetros de la función `read_csv()` indican qué símbolo usamos para separar decimales y columnas. Usamos el parámetro `sep` para indicar qué símbolo marca el final de una columna y el comienzo de la siguiente. El parámetro decimal indica el símbolo utilizado para los decimales.

```
file = pd.read_csv('file.csv',  sep=';' , decimal=',')
```

Calcular la media con tablas dinámicas

En Preprocesamiento de datos, utilizaste `pivot_table()`, un método para crear tablas dinámicas. Estableces el valor de `aggfunc` en `sum`, totalizando todos los elementos en la columna. Si no ingresas un parámetro `aggfunc`, el método `pivot_table()` calculará por defecto la media aritmética de los valores enumerados en el parámetro `values`.

Verificación básica de datos

Ten en cuenta que pueden surgir todo tipo de problemas cuando trabajas con datos:

- Datos incompletos o inexactos
- Errores en el registro de valores de tiempo
- Segundos que se confunden con minutos
- Hechos importantes que se pasan por alto.

Como analista, eres directamente responsable de la calidad de los datos y de las conclusiones que extraes. Cuando obtienes nuevos datos, necesitas tener una idea

de su fiabilidad. En este caso, puedes explorar algunas preguntas básicas. Luego, tú y tus compañeros de trabajo podéis ver si los resultados son razonables.

Una verificación básica como esta podría descubrir un problema en los datos. Por supuesto, también podría decirte que todo está bien, al menos por ahora.

Gráficos de barras

A veces usamos gráficos de barras para trazar datos cuantitativos. Cada barra en un gráfico así corresponde a un valor: cuanto mayor sea el valor, mayor será la barra. Las diferencias entre los valores son claramente visibles.

En pandas, los gráficos se trazan con el método `plot()`. Se pasan varios tipos de gráficos en el parámetro `kind`. Para trazar un gráfico de barras, indica `'bar'`.

Histogramas

Un histograma es un gráfico que muestra la frecuencia con la que aparecen diferentes valores en un conjunto de datos. Agrupa valores numéricos por rangos; es decir, encuentra la frecuencia con la que ocurren los valores dentro de cada intervalo. Los histogramas son algo similares a los gráficos de barras, pero para los primeros, la agrupación se realiza para rangos de valores definidos. Puedes modificar el ancho del intervalo para cambiar la apariencia de tu histograma.
En pandas, los histogramas se trazan con el método `hist()`.Se puede aplicar a una lista o una columna de un DataFrame, en cuyo caso la columna se pasa como argumento. El método `hist()` encuentra los valores más altos y más bajos en un conjunto y divide el rango resultante en 10 intervalos igualmente espaciados, o **contenedores**. Luego, el método encuentra el número de valores dentro de cada contenedor y lo representa en el gráfico.

Otra forma de trazar un histograma es llamar al método `plot()` con el parámetro `kind='hist'`. Esto te permite incluir más parámetros. El parámetro `bins` determina en cuántas secciones se dividirá un rango de datos con el valor predeterminado `bins=10`. También puedes establecer manualmente la escala usando el **parámetro** `range`: `range=(min_value, max_value)`.

```
import pandas as pd
import matplotlib.pyplot as plt # importar la biblioteca usando el nombre estándar plt

pd.Series(...).hist(bins=n_bins, range=(min_value, max_value)

plt.show() # dar el comando para mostrar el histograma
```

Diagramas de caja

Al describir una distribución, los analistas calculan la media o la mediana. Aprendiste sobre los métodos `mean()` y `median()` en el curso de Preprocesamiento de datos. Además de la media y la mediana, también es importante conocer la **dispersión**: qué valores y cuántos de ellos están lejos del promedio.

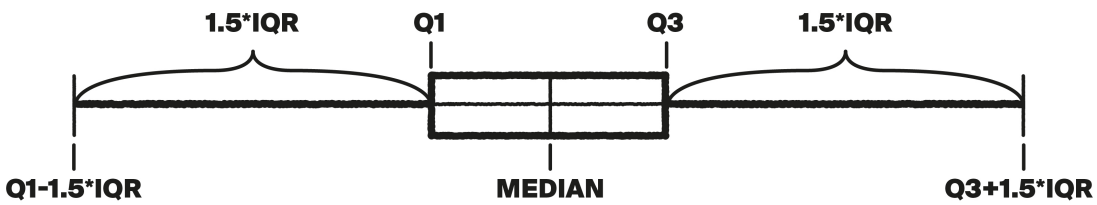
La forma más fácil de tener una idea de la dispersión es mirar los valores mínimo y máximo, pero si hay valores atípicos, esto no te dirá mucho. Es mucho mejor mirar el **rango intercuartílico.**

Los **cuartiles** (del latín *quartus*, o cuarto) dividen conjuntos ordenados de datos en cuatro partes. El primer cuartil, o Q1, marca el valor mayor que el 25% de los elementos del conjunto de datos y menor del 75%. La mediana es Q2; aquí los elementos se dividen en dos. Q3 es mayor que el 75% de los elementos y menor del 25%. El rango intercuartílico es todo lo que se encuentra entre Q1 y Q3.

Podemos encontrar la mediana y los cuartiles en Python usando un gráfico especial llamado **diagrama de caja**, o diagrama de caja y bigotes.

La caja se extiende desde el primer hasta el tercer cuartil, y la mediana se dibuja dentro de ella.

Los bigotes se extienden un máximo de 1,5 rangos intercuartílicos (IQR) a la izquierda y derecha de la caja. (Cada bigote va al valor más grande que se encuentra dentro de este rango.) Los valores típicos se encuentran dentro de los bigotes mientras que los valores atípicos se muestran como puntos fuera de ellos.



Python tiene un método `boxplot()` para crear estos gráficos:

```
import matplotlib.pyplot as plt
data.boxplot()
plt.show()
```

Podemos importar la librería **Matplotlib** y su módulo pyplot para hacer un trabajo más avanzado con histogramas y gráficos en general. Por ejemplo, podemos ajustar los nombres de los ejes especificando el rango de los ejes X e Y. Para hacerlo, necesitamos usar los métodos de Matplotlib: `ylim(y_min, y_max)` para el eje vertical, `xlim(x_min, x_max)` para el eje horizontal.

```
import matplotlib.pyplot as plt
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
```

Describir datos

Los histogramas y los diagramas de caja nos permiten describir conjuntos de datos visualmente. Además del método `quantile()`, se puede usar `describe()` para obtener valores de cuartiles y medianas. Es conveniente porque da **descripciones numéricas de datos**. Una descripción numérica es más que un simple accesorio de los gráficos; puede valerse por sí misma como una herramienta inicial para el análisis. Te informa sobre todas las columnas de una tabla de una sola vez.

La **desviación estándar** describe la forma en que se dispersan los valores y nos permite saber qué tan lejos tienden a estar de la media. La desviación estándar a menudo nos ayuda a comprender las distribuciones y descubrir qué tan uniformes son los conjuntos de datos.

```
data['column'].describe()
```