Clase 029 — Pandas: pivot tables y crosstab

Parte: 0 — Prerrequisitos · Fuente: VanderPlas, cap. 3 § 3.10 Pivot Tables. · Duración estimada: 60 min.

Clase 029 — Pandas: pivot tables y crosstab

> *Parte: 0 — Prerrequisitos · Fuente: VanderPlas, cap. 3 § 3.10 Pivot Tables. · Duración estimada: 60 min.*

Objetivo

Que el alumno construya tablas pivot (estilo Excel) con pivot_table y tablas de contingencia con crosstab. Son atajos sobre groupby pensados para resumen×visualización rápida.

Resultados de aprendizaje

Al finalizar la clase, el alumno podrá:

1. Usar pivot_table con index, columns, values, aggfunc.
2. Añadir totales con margins=True.
3. Construir tablas de contingencia con pd.crosstab y normalizar (normalize='all'/'index'/'columns').
4. Diferenciar pivot (sin agregar) vs pivot_table (con aggfunc, agrega duplicados).
5. Visualizar una pivot como heatmap básico para confirmar patrones.

Temas

#	Tema	Por qué importa
1	pivot vs pivot_table	pivot no acepta duplicados; pivot_table sí
2	Parámetros: index, columns, values, aggfun	Análogos a Excel.
3	margins=True: totales	Útil para verificar.
4	crosstab: tabla de contingencia	Counts entre dos categóricas.
5	normalize en crosstab	Proporciones por fila/col/total.
6	Pivot → heatmap	Detectar patrones visualmente.

Definiciones y características

pivot_table

: Resumen tabular estilo Excel: defines index, columns, values y aggfunc. Acepta duplicados (agrega). El atajo más usado para reportes.

pivot (sin _table)

: Variante que NO agrega — falla si hay duplicados en (index, columns). Más estricta; úsala solo cuando garantizas unicidad.

crosstab

: Tabla de contingencia entre 2 categóricas: counts cruzados. Con normalize='index'/'columns'/'all' muestra proporciones.

margins=True

: Añade fila/columna "Total" al pivot. Útil para verificar manualmente y para reportes ejecutivos.

Heatmap de pivot

: Renderizar el pivot como matriz coloreada (plt.imshow o seaborn.heatmap) — patrones visuales saltan a la vista.

Dataset / recursos

Palmer Penguins. Sin descarga adicional.

Ejercicios

1. Pivot básico. Penguins: índice species, columnas sex, valores body_mass mean.

2. Pivot con totales. Mismo con margins=True.

3. Crosstab counts. Counts species × island.

4. Crosstab normalizado. Mismo con normalize='index' (% por fila).

5. Pivot → heatmap. Toma un pivot table y plotéala con matplotlib imshow.

Homework verificable

Notebook con penguins: (a) pivot_table (species × island, mean body_mass); (b) crosstab species × island, count y normalizado; (c) verificación de totales con margins; (d) heatmap simple del pivot.

Criterio de aceptación: Pivot con shape correcto; sum de normalize='index' = 1.0 por fila.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
pivot() lanza ValueError: Index contains d	Hay duplicados en (index, columns). Fix: u
pivot_table da NaN donde no hay datos	Combinaciones (index × columns) sin filas.
crosstab cuenta cosas raras con muchos NaN	Crosstab cuenta filas no-NaN por default.
Pivot con cols numéricas float queda feo	Sin aggfunc explícito, pandas usa mean. Si
Plot del pivot rompe por MultiIndex	Pivot con múltiples niveles de columnas →

Preguntas frecuentes

¿pivot_table o groupby + unstack?

Equivalentes en resultado. pivot_table es más declarativo, mejor para reportes. groupby + unstack más componible, mejor en pipelines.

¿crosstab o pivot_table con aggfunc='count'?

Equivalentes para counts. crosstab tiene API más simple para 2 categóricas. pivot_table más flexible (varias values, varias funcs).

¿Cómo ordeno el pivot?

Por valores: pivot.sort_values('col_x', ascending=False). Por suma: pivot.loc[pivot.sum(axis=1).sort_values(ascending=False).index].

¿Reportes Excel-like exportables?

pivot.to_excel('reporte.xlsx') directo. O to_csv para CSV. Para formato fino (colores, formulas), usa openpyxl o xlsxwriter.

¿Cuándo no usar pivot?

Cuando los datos ya están en formato wide y solo necesitas plot/agregaciones — usar groupby directo. Pivot es para transformar long → wide.

Referencias

- VanderPlas, cap. 3 § 3.10.
- pandas Pivot guide

Siguiente clase

Clase 030 — Pandas: operaciones vectorizadas sobre strings

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

rng = np.random.default_rng(42)

df = pd.DataFrame({
    'species': np.repeat(['Adelie', 'Chinstrap', 'Gentoo'], [12, 8, 10]),
    'island' : rng.choice(['Biscoe', 'Dream', 'Torgersen'], 30),
    'sex'    : rng.choice(['M', 'F'], 30),
    'masa'   : rng.normal(4200, 600, 30),
})
print(df.head())
```

Archivos complementarios

- notebook.ipynb