

---

## **Clase 041 — Seaborn: distribuciones, relaciones, categóricas, facetas**

Parte: 0 — Prerrequisitos · Fuente: VanderPlas, cap. 4 § 4.13 Visualization with Seaborn · seaborn docs. · Duración estimada: 75 min.

# Clase 041 — Seaborn: distribuciones, relaciones, categóricas, facetas

Parte: 0 — Prerrequisitos · Fuente: VanderPlas, cap. 4 § 4.13 Visualization with Seaborn · seaborn docs.  
· Duración estimada: 75 min.

## Objetivo

Que el alumno use seaborn cuando aporta sobre matplotlib puro: defaults estéticos, API tipada para DataFrames (`x=`, `y=`, `hue=`, `col=`), distribuciones (`histplot`, `kdeplot`, `displot`), relaciones (`scatterplot`, `lmplot`), categóricas (`boxplot`, `violinplot`, `swarmplot`), y facetas (grilla automática por categoría).

## Resultados de aprendizaje

Al finalizar la clase, el alumno podrá:

1. Usar la API moderna (figure-level vs axes-level) y elegir la correcta.
2. Construir un pairplot para EDA rápido de un DataFrame.
3. Codificar 3 dimensiones con hue, style, size.
4. Hacer facetas con col= y row= para grillas automáticas.
5. Personalizar themes con `sns.set_theme(style=..., palette=...)`.

## Temas

#	Tema	Por qué importa
1	seaborn vs matplotlib	Seaborn es matplotlib + defaults + API tip
2	Figure-level (displot, relplot, catplot) v	Cuándo cada uno.
3	hue, style, size	Codificar dimensiones extra.
4	Facetas con col, row	Grilla automática por categoría.
5	pairplot para EDA	Matriz de scatters.
6	Themes y paletas	Defaults consistentes.

## Definiciones y características

Seaborn

: Wrapper sobre matplotlib con (1) defaults estéticos mejores, (2) API tipada para DataFrames (`x=`, `y=`, `hue=`), (3) plots estadísticos directos (regresión, distribuciones, facetas).

Figure-level (displot, relplot, catplot)

: Funciones que crean su propia figura, soportan facetas (`col=`, `row=` para grilla automática). Más alto nivel, menos control fino.

Axes-level (histplot, scatterplot, boxplot)

: Funciones que dibujan en un ax que tú pasas. Más bajo nivel, integran con layouts custom de matplotlib.

hue, style, size

: Codifican dimensiones extra: hue (color por categoría), style (marker por categoría), size (tamaño por valor continuo).

pairplot

: Matriz de scatterplots de todas las parejas de variables numéricas, diagonal con KDE/histograma. EDA visual en una línea.

Faceta (col/row)

: Una sub-figura por cada valor de una categórica. `relplot(..., col='species', row='sex')` produce grilla `n_species × n_sex` de plots automática.

## Dataset / recursos

Palmer Penguins (seaborn lo trae built-in via `sns.load_dataset('penguins')`).

## Ejercicios

1. Pairplot. Penguins, color por species. EDA en 1 línea.
2. Scatter con hue + size. `body_mass` vs `flipper`, hue por species, size por `bill_length`.
3. KDE distribución. `body_mass` por species (3 KDE en mismo plot).
4. Boxplot + swarm. Combinar boxplot con swarm para ver puntos individuales.
5. Facetas. `sns.relplot(...col='species', row='sex')` para  $3 \times 2 = 6$  subplots automáticos.

## Homework verificable

Notebook con penguins: (a) pairplot completo; (b) violin + swarm de `body_mass` por (species, sex); (c) faceta  $2 \times 3$  de scatter; (d) tema custom + paleta; (e) decisión documentada: cuándo usar figure-level vs axes-level.

Criterio de aceptación: Plots de EDA legibles. Decisiones de hue/style/col justificadas.

## Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
Mezclo <code>sns.set_theme()</code> con <code>plt.style.use()</code> .	Compiten por los <code>rcParams</code> . Fix: elige uno.
Figure-level no me deja agregar <code>ax.set_title</code>	Devuelve <code>FacetGrid</code> , no axes. Fix: <code>g.set_title</code>
pairplot con muchas columnas tarda eternid	$N^2$ scatters; con 20 cols son 400 plots. Fi
hue con muchas categorías → leyenda enorme	Default muestra todas las categorías. Fix:
<code>sns.scatterplot(...)</code> no aparece en noteboc	Falta capturar return o <code>plt.show()</code> . Fix: a

## Preguntas frecuentes

¿seaborn o matplotlib puro?

Seaborn para plots estadísticos rápidos con DataFrame (hue, facetas, defaults). Matplotlib para control fino,

customización extrema, o plots que no son estadísticos.

¿Cuándo figure-level vs axes-level?

Figure-level si quieres facetas o el plot es el output completo. Axes-level si necesitas integrar con layout custom (subplots manuales).

¿pairplot o corrmatrix?

pairplot muestra relaciones visualmente (no lineales, outliers, clusters). Heatmap de corr() muestra coeficiente Pearson (solo lineal). Complementarios.

¿Tema corporativo en seaborn?

sns.set\_theme(palette=['#0F766E', '#D9A441', '#7C3AED']) o palette custom: sns.color\_palette('husl', n\_colors=5). Combina con style='whitegrid'.

¿Seaborn maneja datasets grandes (>1M)?

Plots scatter/hist sí. Pairplot con muchas cols se vuelve lento. Para datasets enormes, considera datashader o downsample antes.

## Referencias

- VanderPlas, cap. 4 § 4.13.
- seaborn user guide
- Waskom, seaborn paper (JOSS, 2021)

## Siguiente clase

Clase 042 — Visualización geográfica (Plotly / folium)

## 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
import seaborn as sns

sns.set_theme(style='whitegrid', palette='deep')

try:
    peng = sns.load_dataset('penguins').dropna()
    print(f'penguins cargado: {peng.shape}')
except Exception as e:
    print(f'fallback sintético: {e}')
    rng = np.random.default_rng(42)
    peng = pd.DataFrame({
        'species': np.repeat(['Adelie', 'Chinstrap', 'Gentoo'], [50, 30, 40]),
        'sex' : np.tile(['Male', 'Female'], 60),
        'bill_length_mm' : np.concatenate([rng.normal(39, 2, 50), rng.normal(48, 3, 30), rng.normal(48, 3, 40)]),
        'bill_depth_mm' : np.concatenate([rng.normal(18, 1, 50), rng.normal(18, 1, 30), rng.normal(15, 1, 40)]),
        'flipper_length_mm': np.concatenate([rng.normal(190, 6, 50), rng.normal(196, 7, 30), rng.normal(217, 7, 40)]),
```

```
'body_mass_g' : np.concatenate([rng.normal(3700, 400, 50), rng.normal(3700, 400, 30), rng.normal(5050, 500, 40)],  
                                })
```

## Archivos complementarios

- notebook.ipynb