

Cuándo usar cada tipo de gráfico

Elegir el gráfico adecuado depende del tipo de datos y del mensaje que quieres transmitir. Aquí tienes una guía sobre cuándo usar cada tipo de gráfico en **Matplotlib y Seaborn**.

1 Gráficos de líneas (`plt.plot()` / `sns.lineplot()`)

✓ Cuándo usarlo:

- Para mostrar tendencias o cambios a lo largo del tiempo.
- Para representar series temporales.
- Para analizar la evolución de una métrica.

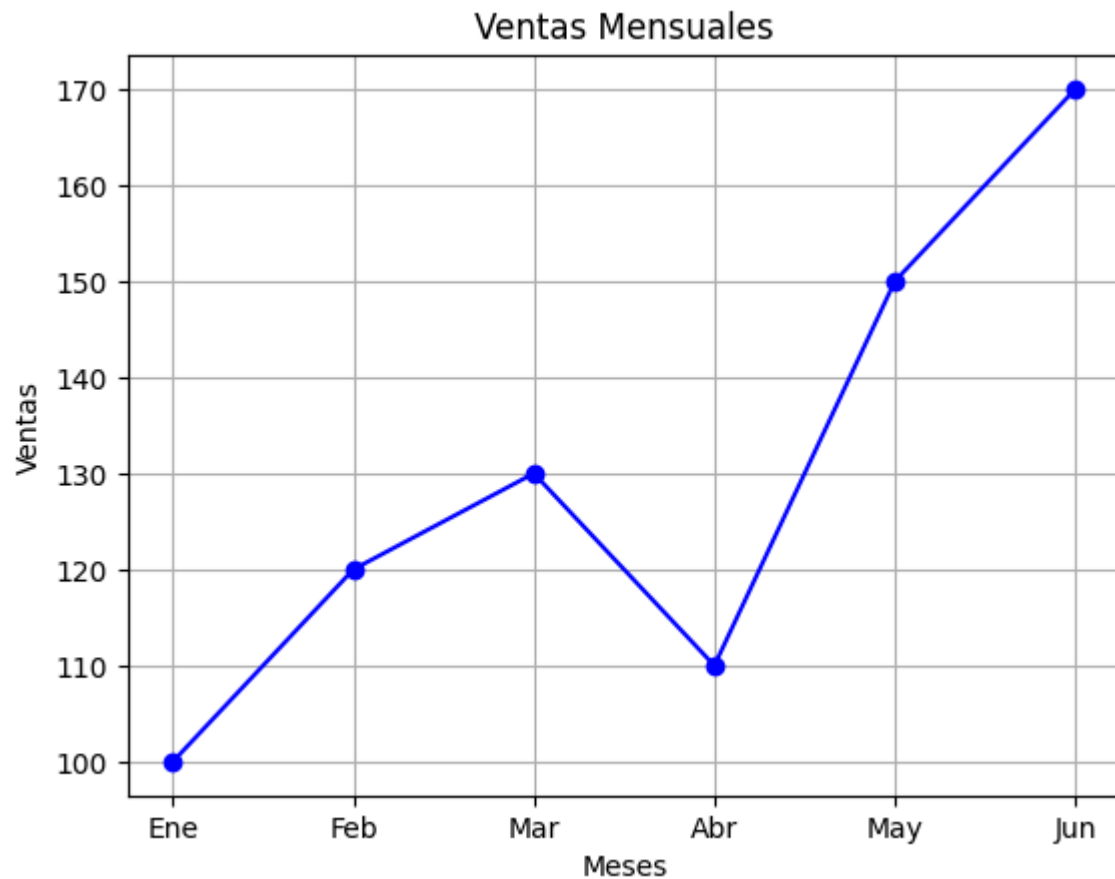
📌 Ejemplo de uso:

Ventas mensuales en un año.

```
import matplotlib.pyplot as plt

meses = ["Ene", "Feb", "Mar", "Abr", "May", "Jun"]
ventas = [100, 120, 130, 110, 150, 170]

plt.plot(meses, ventas, marker='o', linestyle='-', color='b')
plt.title('Ventas Mensuales')
plt.xlabel('Meses')
plt.ylabel('Ventas')
plt.grid(True)
plt.show()
```



2 Gráficos de barras (`plt.bar()` / `sns.barplot()`)

✓ Cuándo usarlo:

- Para comparar categorías o grupos.
- Para visualizar frecuencias o valores de variables discretas.

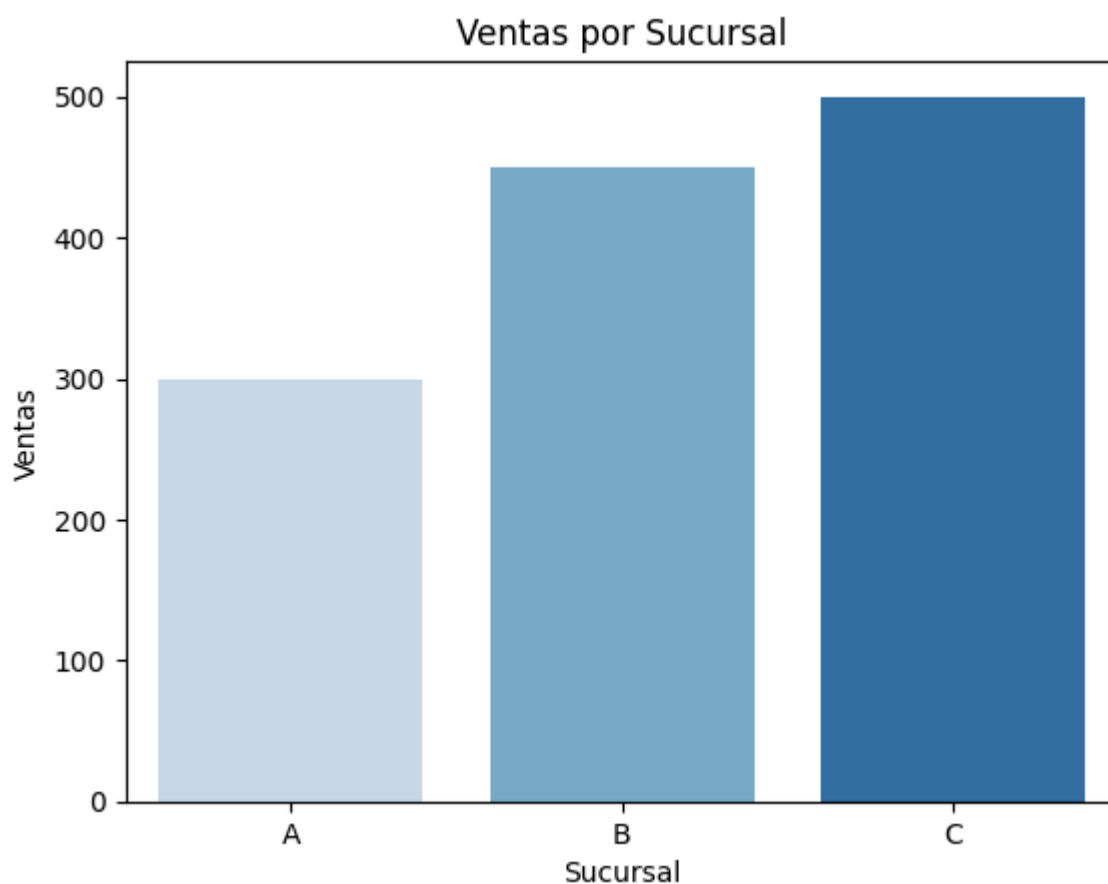
📌 Ejemplo de uso:

📊 Comparación de ventas por sucursal.

```
import seaborn as sns
import pandas as pd

data = pd.DataFrame({
    "Sucursal": ["A", "B", "C"],
    "Ventas": [300, 450, 500]
})
```

```
sns.barplot(x="Sucursal", y="Ventas", data=data, palette="Blues")  
plt.title("Ventas por Sucursal")  
plt.show()
```



3 Gráficos de dispersión (`plt.scatter()` / `sns.scatterplot()`)

✓ Cuándo usarlo:

- Para analizar la relación entre dos variables numéricas.
- Para detectar correlaciones o patrones.

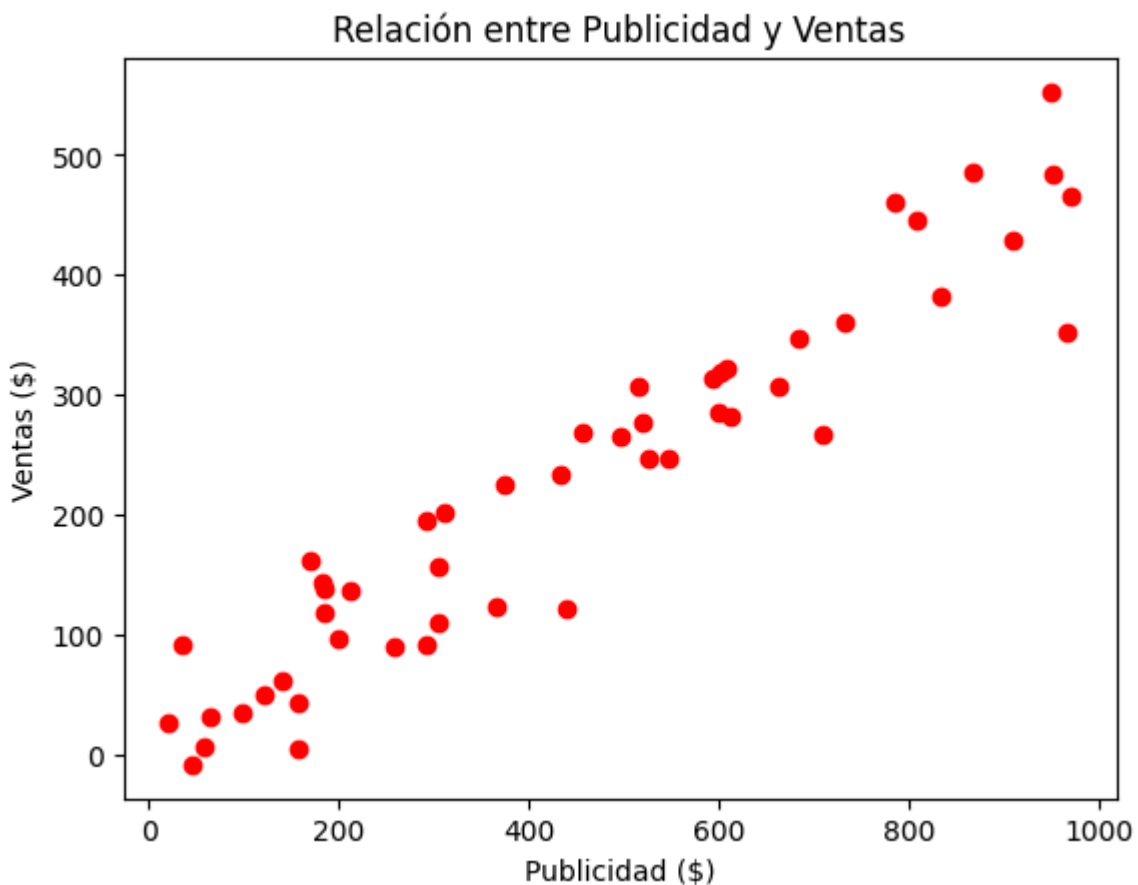
📌 Ejemplo de uso:

🔍 Relación entre publicidad y ventas.

```
import numpy as np
```

```
# Datos simulados
np.random.seed(42)
publicidad = np.random.rand(50) * 1000
ventas = publicidad * 0.5 + np.random.randn(50) * 50

plt.scatter(publicidad, ventas, color='r')
plt.title("Relación entre Publicidad y Ventas")
plt.xlabel("Publicidad ($)")
plt.ylabel("Ventas ($)")
plt.show()
```



4 Histogramas (`plt.hist()` / `sns.histplot()`)

✓ Cuándo usarlo:

- Para visualizar la distribución de una variable numérica.

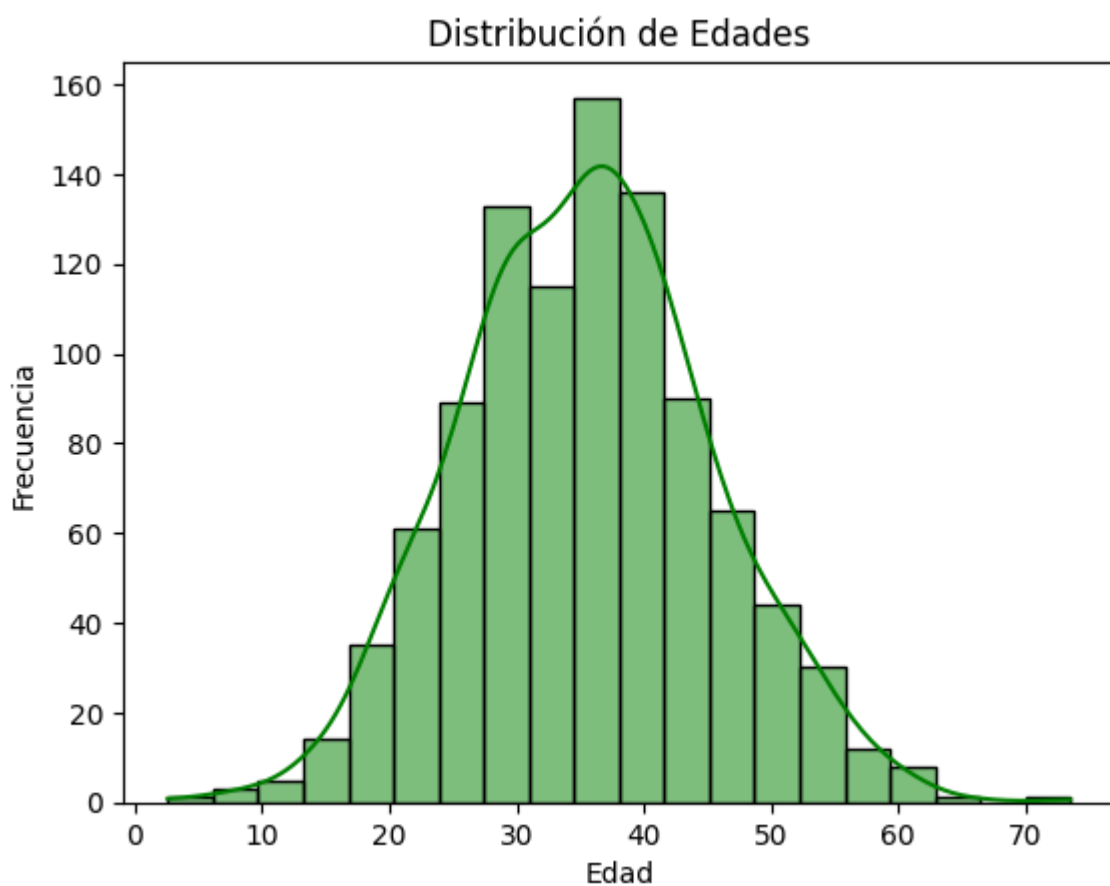
- Para detectar valores atípicos o patrones en los datos.

📌 Ejemplo de uso:

📊 Distribución de edades de clientes.

```
edades = np.random.normal(35, 10, 1000) # Media de 35 años, desviación estándar de 10
```

```
sns.histplot(edades, bins=20, kde=True, color='g')  
plt.title("Distribución de Edades")  
plt.xlabel("Edad")  
plt.ylabel("Frecuencia")  
plt.show()
```



5 Boxplots (`plt.boxplot()` / `sns.boxplot()`)

✅ Cuándo usarlo:

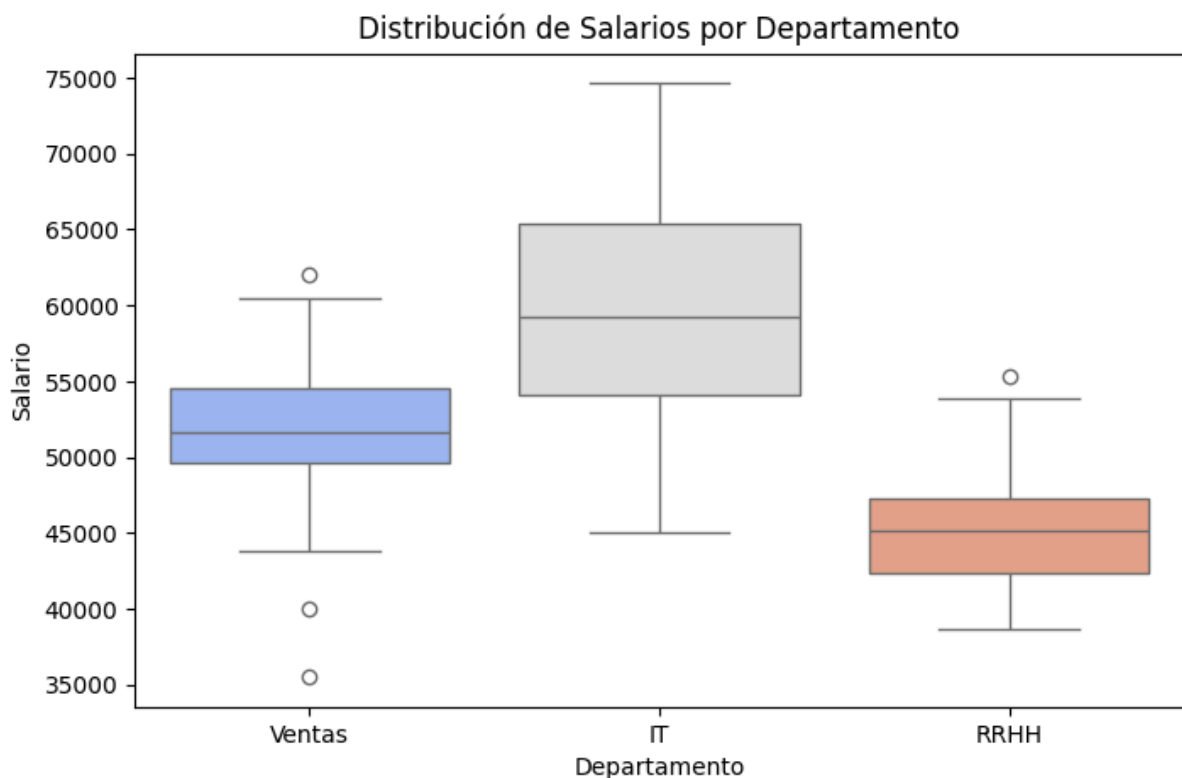
- Para visualizar la distribución y detectar valores atípicos.
- Para comparar varias categorías.

📌 Ejemplo de uso:

📦 Comparación de sueldos por departamento.

```
data = pd.DataFrame({
    "Departamento": ["Ventas"] * 50 + ["IT"] * 50 + ["RRHH"] * 50,
    "Salario": np.concatenate([np.random.normal(50000, 5000, 50),
                               np.random.normal(60000, 6000, 50),
                               np.random.normal(45000, 4000, 50)])
})

plt.figure(figsize=(8,5))
sns.boxplot(x="Departamento", y="Salario", data=data, palette="coolwarm")
plt.title("Distribución de Salarios por Departamento")
plt.show()
```



6 Mapas de calor (`sns.heatmap()`)

✓ Cuándo usarlo:

- Para visualizar correlaciones entre variables.
- Para analizar matrices de datos.

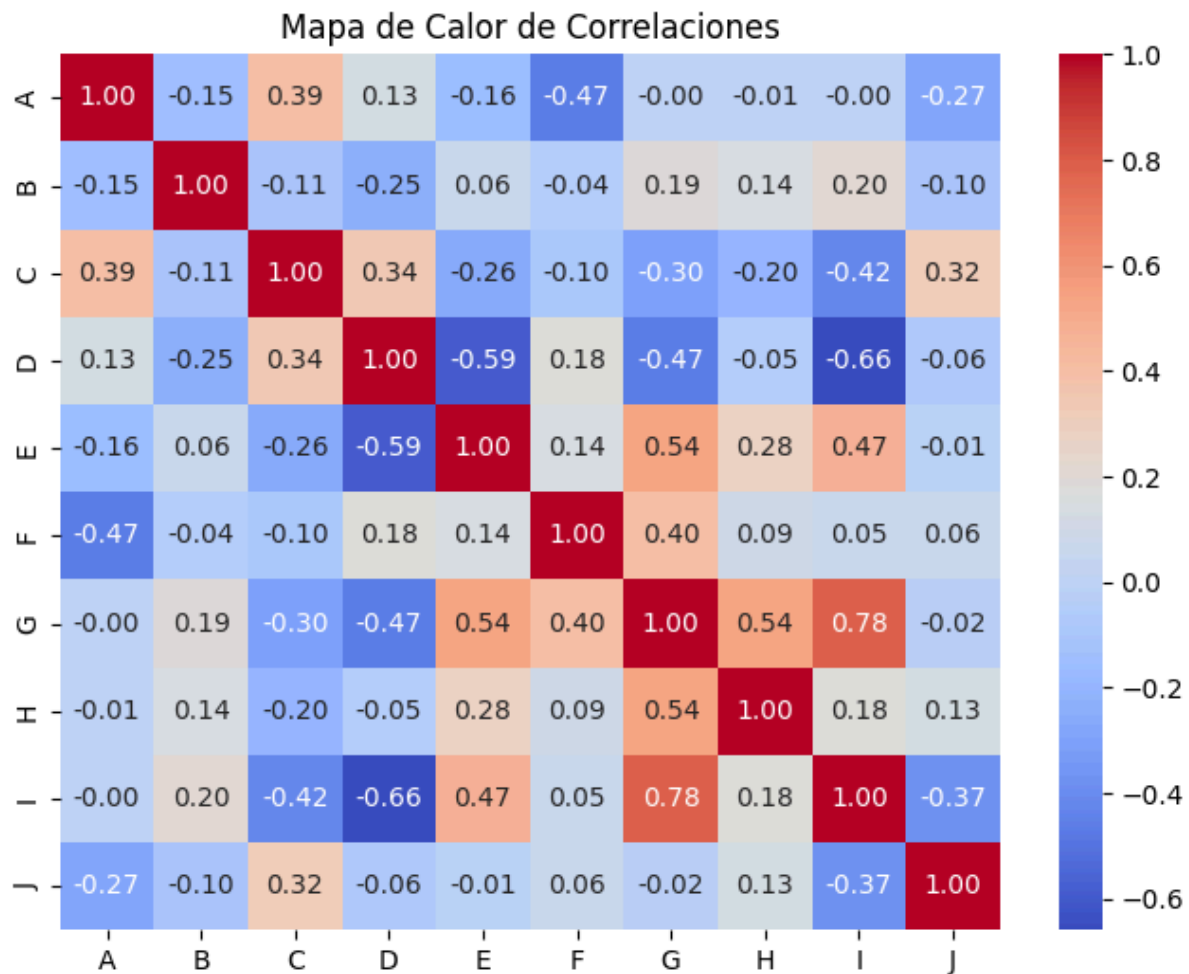
📌 Ejemplo de uso:

🔥 Correlación entre variables en un dataset.







```
import seaborn as sns

# Crear datos de correlación
data = pd.DataFrame(np.random.rand(10, 10), columns=list('ABCDEFGHJI
J'))

plt.figure(figsize=(8,6))
sns.heatmap(data.corr(), annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Mapa de Calor de Correlaciones")
plt.show()
```



Resumen: ¿Qué gráfico usar?

Tipo de gráfico	¿Cuándo usarlo?
 Líneas	Tendencias, series temporales.
 Barras	Comparación de categorías.
 Dispersión	Relación entre dos variables.
 Histograma	Distribución de una variable.
 Boxplot	Comparar distribuciones y detectar valores atípicos.
 Mapa de calor	Correlaciones entre variables.

Gráficos avanzados

1 Gráficos de Líneas Avanzados (Series Temporales con Estimaciones y Múltiples Líneas)

✓ Casos de uso avanzados:

- Visualizar tendencias con líneas de regresión suavizadas.
- Comparar múltiples series en un solo gráfico.
- Resaltar periodos de interés en una serie de tiempo.

📌 Ejemplo: Comparación de ventas de dos productos con tendencia suavizada

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

# Generar datos de ventas de dos productos
np.random.seed(42)
dias = np.arange(1, 31)
ventas_a = np.cumsum(np.random.randint(10, 50, size=30))
ventas_b = np.cumsum(np.random.randint(20, 60, size=30))

# Crear DataFrame
data = pd.DataFrame({"Día": dias, "Producto A": ventas_a, "Producto B": ventas_b})

# Graficar con líneas suavizadas
plt.figure(figsize=(10, 5))
sns.lineplot(x="Día", y="Producto A", data=data, label="Producto A", marker='o')
sns.lineplot(x="Día", y="Producto B", data=data, label="Producto B", marker='s')

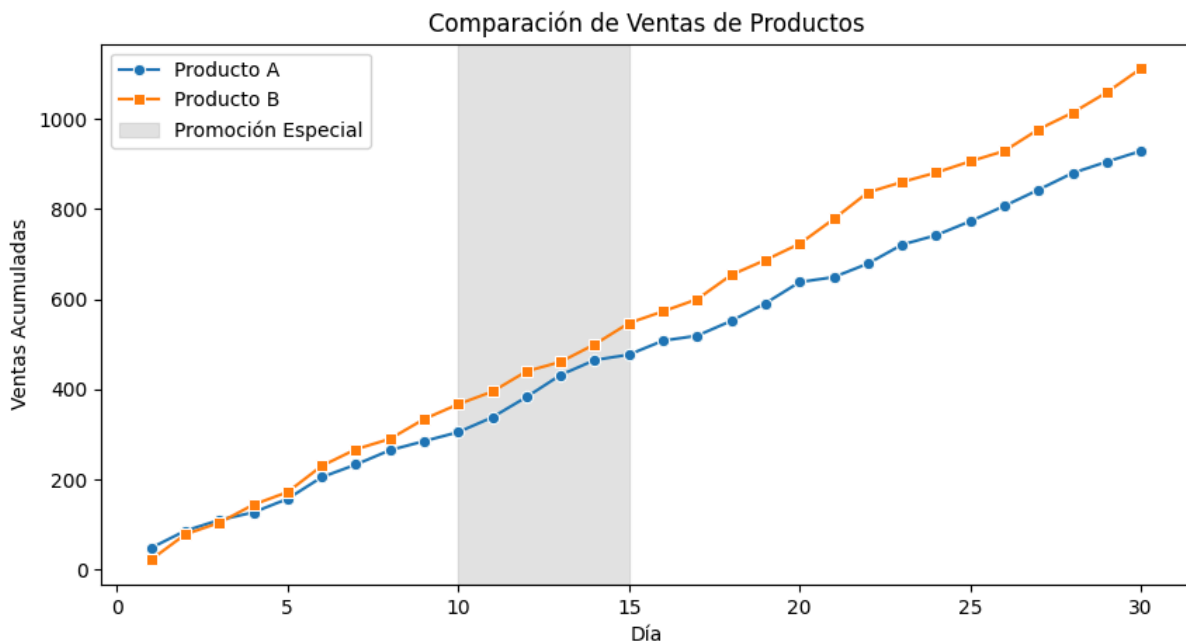
# Resaltar un periodo de interés
plt.axvspan(10, 15, color='gray', alpha=0.2, label="Promoción Especial")

plt.title("Comparación de Ventas de Productos")
plt.xlabel("Día")
```

```
plt.ylabel("Ventas Acumuladas")
plt.legend()
plt.show()
```

◆ ¿Qué se agregó?

- Se comparan dos productos en la misma gráfica.
- Se usa `sns.lineplot()` para trazar múltiples líneas con marcadores.
- Se añade `axvspan()` para resaltar un periodo específico.



2 Gráficos de Barras Avanzados (Barras Agrupadas y Apiladas)

✓ Casos de uso avanzados:

- Comparar múltiples categorías dentro de un mismo grupo.
- Mostrar cambios en una métrica en distintas condiciones.

📌 Ejemplo: Ventas de productos por trimestre con barras agrupadas

```
# Datos de ventas de 3 productos en diferentes trimestres
trimestres = ["Q1", "Q2", "Q3", "Q4"]
ventas = {
```

```

"Producto A": [200, 250, 300, 280],
"Producto B": [180, 220, 270, 260],
"Producto C": [210, 260, 310, 290],
}

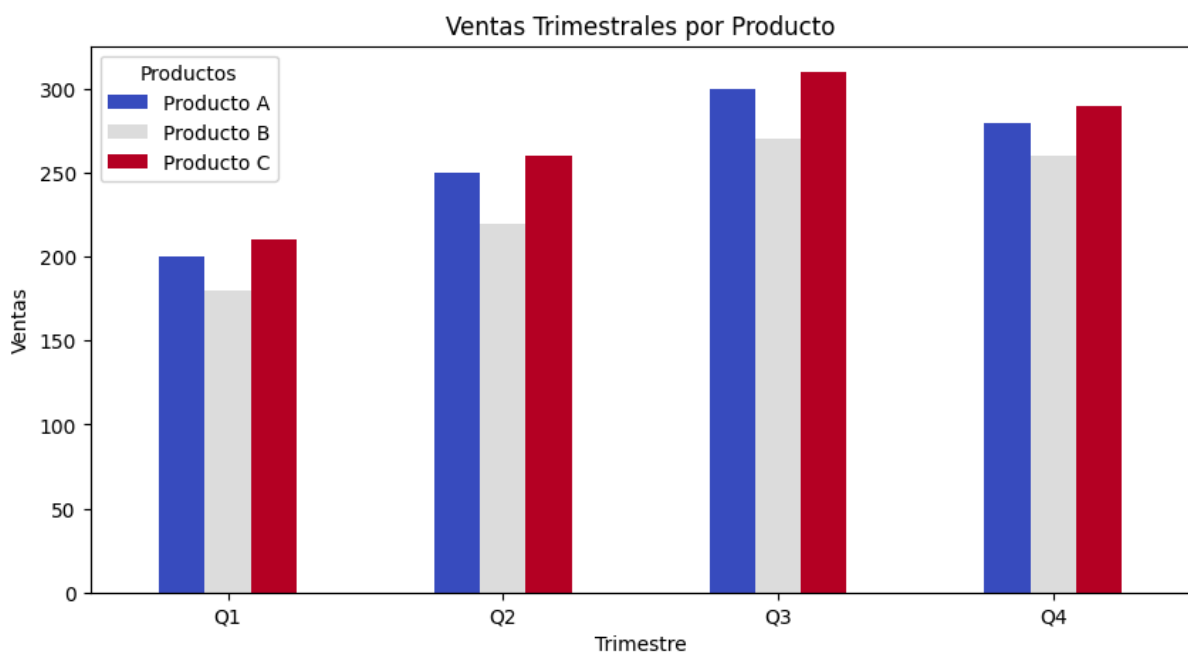
df = pd.DataFrame(ventas, index=trimestres)

# Crear gráfico de barras agrupadas
df.plot(kind='bar', figsize=(10, 5), colormap='coolwarm')
plt.title("Ventas Trimestrales por Producto")
plt.xlabel("Trimestre")
plt.ylabel("Ventas")
plt.xticks(rotation=0)
plt.legend(title="Productos")
plt.show()

```

◆ ¿Qué se agregó?

- `df.plot(kind='bar')` para barras agrupadas.
- Colores con `colormap='coolwarm'`.
- Se mantiene el texto horizontal (`rotation=0`) para mejor legibilidad.

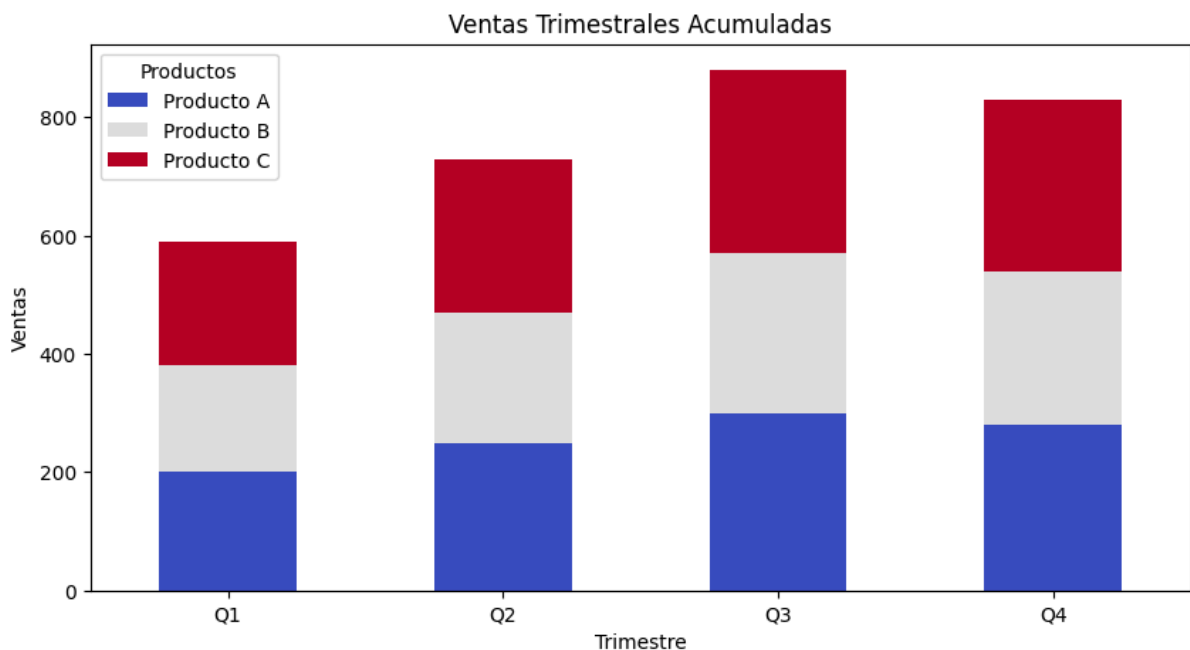


📌 Ejemplo: Barras Apiladas (Comparación de Ventas en un solo gráfico)

```
# Crear gráfico de barras apiladas
df.plot(kind='bar', stacked=True, figsize=(10, 5), colormap='coolwarm')
plt.title("Ventas Trimestrales Acumuladas")
plt.xlabel("Trimestre")
plt.ylabel("Ventas")
plt.xticks(rotation=0)
plt.legend(title="Productos")
plt.show()
```

◆ ¿Qué se agregó?

- `stacked=True` para hacer el gráfico de barras apiladas.



3 Gráficos de Dispersión Avanzados (Regresión y Densidad)

✓ Casos de uso avanzados:

- Detectar correlaciones con una línea de tendencia.
- Evaluar la densidad de puntos en un área específica.

📌 **Ejemplo: Relación entre presupuesto de marketing y ventas con regresión**

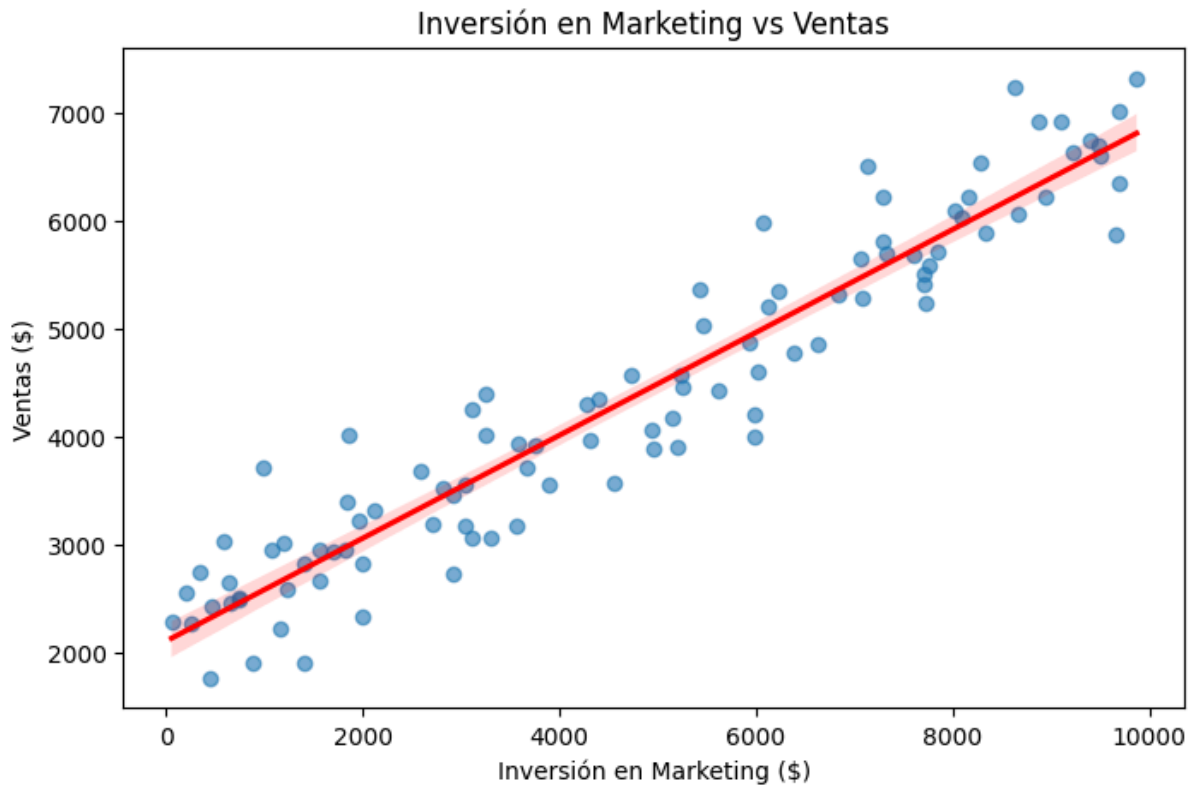
```
# Generar datos de marketing y ventas
np.random.seed(42)
marketing = np.random.rand(100) * 10000
ventas = marketing * 0.5 + np.random.randn(100) * 500 + 2000

df = pd.DataFrame({"Marketing": marketing, "Ventas": ventas})

# Gráfico de dispersión con línea de regresión
plt.figure(figsize=(8, 5))
sns.regplot(x="Marketing", y="Ventas", data=df, scatter_kws={"alpha": 0.6}, line_kws={"color": "red"})
plt.title("Inversión en Marketing vs Ventas")
plt.xlabel("Inversión en Marketing ($)")
plt.ylabel("Ventas ($)")
plt.show()
```

◆ ¿Qué se agregó?

- `sns.regplot()` agrega una línea de regresión.
- `scatter_kws={"alpha": 0.6}` para hacer los puntos semitransparentes.
- `line_kws={"color": "red"}` para personalizar la línea.

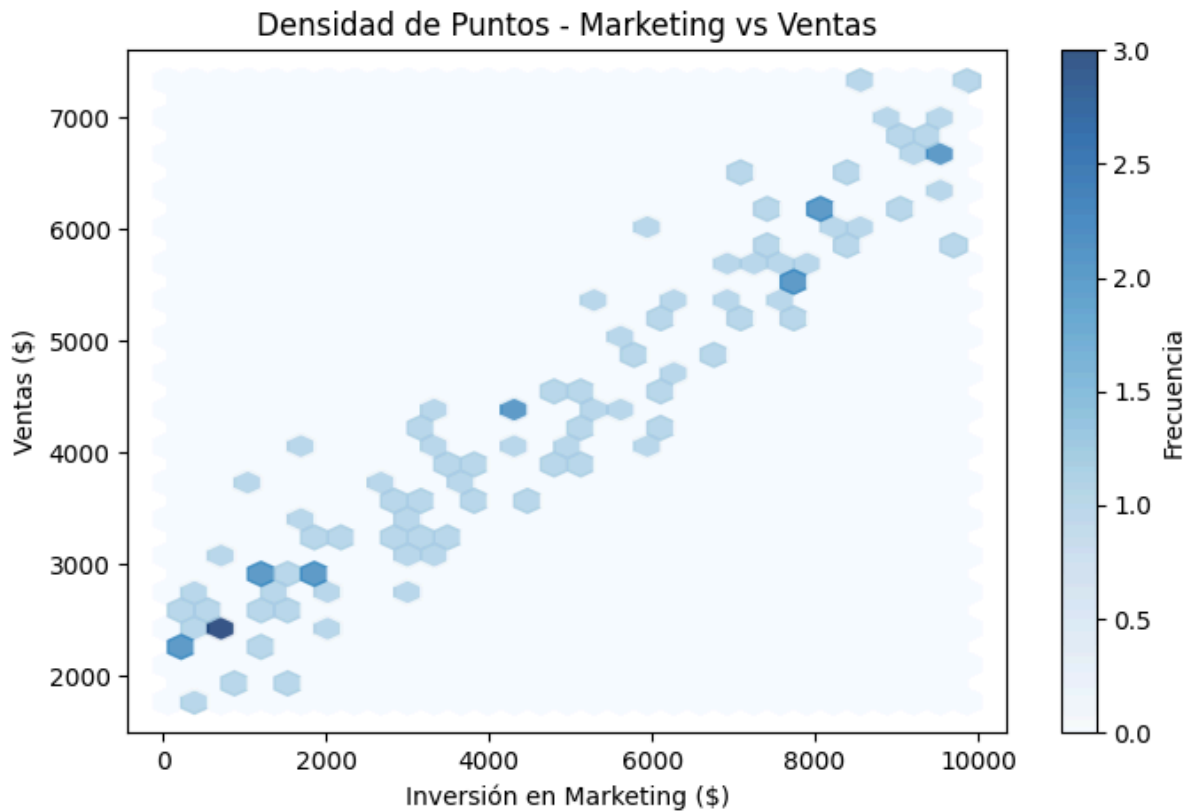


📌 Ejemplo: Mapa de Densidad en 2D (Hexbin Plot)

```
# Crear gráfico de hexágonos
plt.figure(figsize=(8, 5))
plt.hexbin(marketing, ventas, gridsize=30, cmap='Blues', alpha=0.8)
plt.colorbar(label="Frecuencia")
plt.title("Densidad de Puntos - Marketing vs Ventas")
plt.xlabel("Inversión en Marketing ($)")
plt.ylabel("Ventas ($)")
plt.show()
```

◆ ¿Qué se agregó?

- `plt.hexbin()` para mostrar densidad en hexágonos.
- `gridsize=30` para controlar el tamaño de los hexágonos.



4 Gráficos de Boxplot Avanzados (Comparación de Distribuciones)

✓ Casos de uso avanzados:

- Comparar distribuciones en múltiples grupos.
- Identificar valores atípicos en diferentes segmentos.

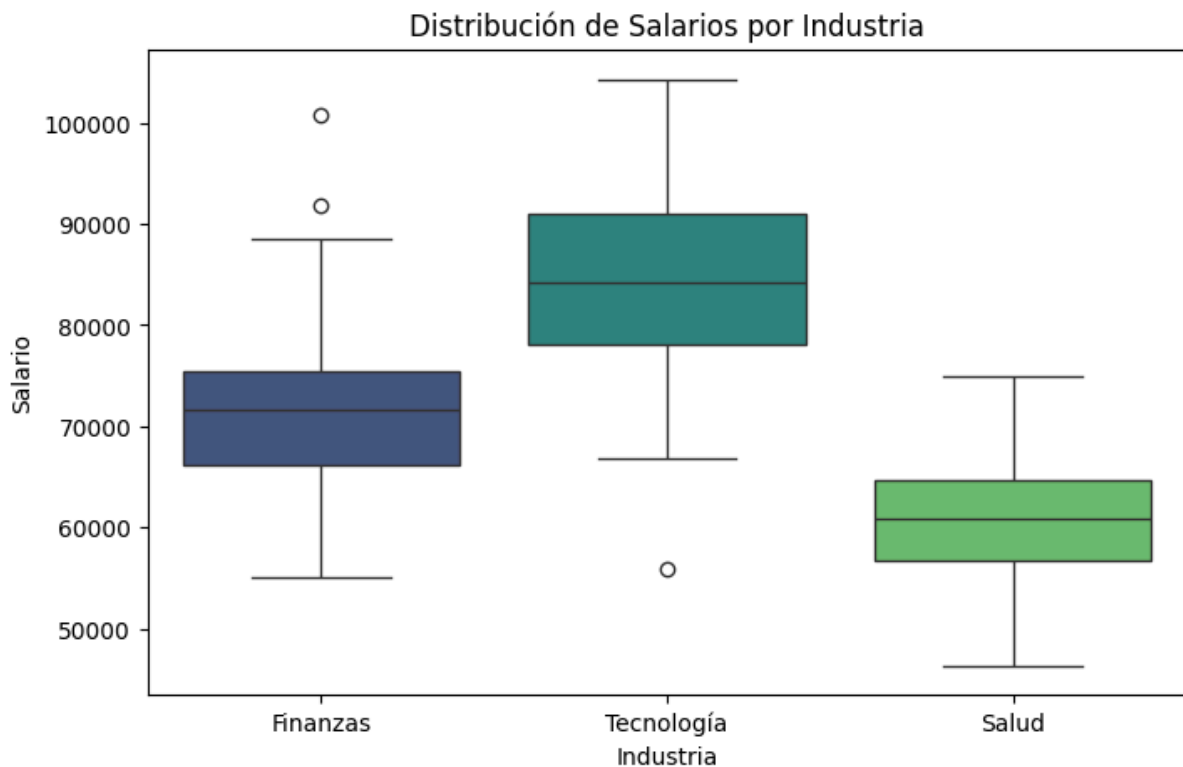
📌 Ejemplo: Comparación de salarios por industria

```
# Generar datos de salarios por industria
data = pd.DataFrame({
    "Industria": ["Finanzas"] * 50 + ["Tecnología"] * 50 + ["Salud"] * 50,
    "Salario": np.concatenate([
        np.random.normal(70000, 8000, 50),
        np.random.normal(85000, 9000, 50),
        np.random.normal(60000, 7000, 50)
    ])
})
```

```
plt.figure(figsize=(8,5))
sns.boxplot(x="Industria", y="Salario", data=data, palette="viridis")
plt.title("Distribución de Salarios por Industria")
plt.show()
```

◆ ¿Qué se agregó?

- `sns.boxplot()` para visualizar distribución.
- `palette="viridis"` para colores modernos.



5 Mapas de Calor Avanzados (Matriz de Correlación Mejorada)

✓ Casos de uso avanzados:

- Analizar correlaciones entre múltiples variables.
- Ajustar etiquetas y colores para mejorar la interpretación.

📌 Ejemplo: Matriz de correlación entre variables económicas


```
# Generar datos aleatorios
data = pd.DataFrame(np.random.rand(10, 10), columns=list('ABCDEFGHJI
J'))

plt.figure(figsize=(10,6))
sns.heatmap(data.corr(), annot=True, cmap="coolwarm", linewidths=0.5, f
mt=".2f")
plt.title("Matriz de Correlación Mejorada")
plt.show()
```

◆ ¿Qué se agregó?

- `linewidths=0.5` para separar las celdas.
- `fmt=".2f"` para mostrar solo 2 decimales.

