

UNIVERSIDAD SERGIO ARBOLEDA

Análisis de Datos
Nivel Integrador

PREPARACIÓN SEMANA 8

SEABORN ¿QUÉ ES?

Seaborn es una biblioteca de visualización de datos en Python que se basa en Matplotlib y proporciona una interfaz de alto nivel para la creación de gráficos estadísticos atractivos y informativos. Fue desarrollado para hacer que la creación de visualizaciones complejas y atractivas sea más accesible, permitiendo a los usuarios centrarse más en los aspectos analíticos de sus datos y menos en la gestión de detalles de visualización.

Características Principales:

1. **Integración con Pandas:** Seaborn está diseñado para trabajar de manera fluida con DataFrames de Pandas, facilitando la visualización de datos almacenados en estructuras tabulares.
2. **Estilo por Defecto Atractivo:** Seaborn tiene estilos visuales predefinidos que son más atractivos que los estilos predeterminados de Matplotlib. Esto permite que las visualizaciones tengan un aspecto más pulido sin esfuerzo adicional.

SEABORN ¿QUÉ ES?

Ventajas de Utilizar Seaborn:

- **Eficiencia:** Seaborn simplifica la creación de visualizaciones complejas, ahorrando tiempo y esfuerzo en comparación con el uso directo de Matplotlib.
- **Estética Mejorada:** La biblioteca proporciona estilos visuales atractivos y paletas de colores, mejorando la estética de las visualizaciones por defecto.
- **Fácil Integración con Pandas:** Dado que trabaja bien con DataFrames de Pandas, Seaborn es una opción natural para aquellos que ya están familiarizados con el análisis de datos en Python.
- **Abstracción de Detalles Técnicos:** Permite a los usuarios concentrarse en la interpretación y comunicación de sus datos, en lugar de preocuparse por detalles técnicos de visualización.

SEABORN ¿QUÉ ES?

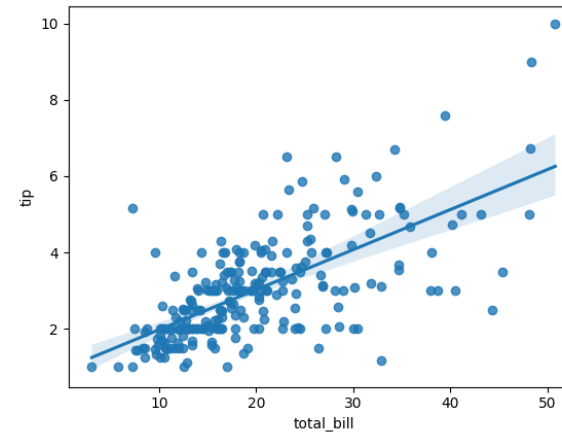
Ejemplo básico: Aquí Seaborn se encarga de detalles como la creación de un gráfico de dispersión con una línea de regresión, permitiendo a los usuarios enfocarse en la interpretación de la relación entre las variables "total_bill" y "tip".

```
import seaborn as sns
import matplotlib.pyplot as plt

# Cargar dataset de ejemplo
tips = sns.load_dataset("tips")

# Crear un gráfico de dispersión con línea de regresión
sns.regplot(x="total_bill", y="tip", data=tips)

# Mostrar el gráfico
plt.show()
```



El código de todos los ejemplos y explicaciones de ésta clase podrán encontrarlo en el siguiente enlace: <https://acortar.link/sli2ku>

CONFIGURACIÓN Y CARGA DE DATOS

Instalación de Seaborn: La instalación de Seaborn es sencilla y se puede realizar mediante la herramienta **pip**: ***pip install seaborn***. Con este comando se instalará la última versión de Seaborn junto con sus dependencias, incluyendo Matplotlib y NumPy

Carga de Datos: Antes de visualizar datos con Seaborn, es esencial cargar los datos en un formato que Seaborn pueda manejar fácilmente. Seaborn funciona de manera excelente con DataFrames de Pandas

Por ejemplo: Aquí se ha utilizado **pd.read_csv()** para cargar datos desde un archivo CSV. Se puede adaptar esto según el formato de tus datos, ya sea CSV, Excel, SQL, etc.

```
import seaborn as sns
import pandas as pd

# Ejemplo: Cargar datos de un archivo CSV
file_path = "ruta/del/archivo/datos.csv"
df = pd.read_csv(file_path)

# Visualizar la estructura del DataFrame
print(df.head())

# Ejemplo de visualización simple con Seaborn
sns.scatterplot(x="columna_x", y="columna_y", data=df)
```

CONFIGURACIÓN Y CARGA DE DATOS

Consejos adicionales:

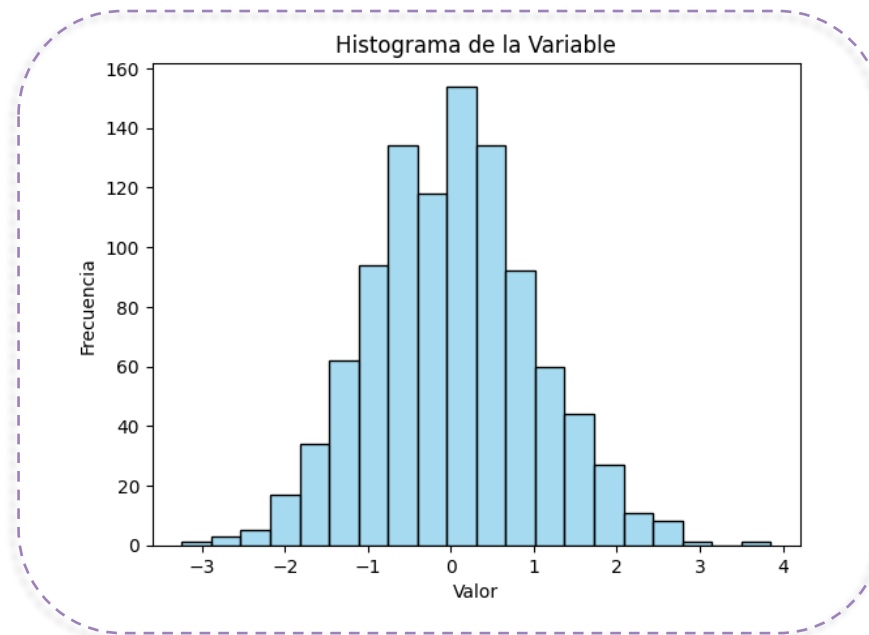
- **Exploración de Datos con Pandas:** Antes de utilizar Seaborn, es una buena práctica repasar `df.describe()`, `df.info()`, y `df.head()`.
- **Manipulación de Datos:** Realiza cualquier manipulación de datos necesaria antes de visualizarlos. Esto puede incluir limpieza de datos, transformaciones y filtrado.
- **Gestión de Datos Faltantes:** Aborda datos faltantes antes de la visualización. Pandas ofrece métodos como `dropna()` o `fillna()` que puedes usar para gestionar valores nulos.

GRÁFICOS BÁSICOS

Seaborn proporciona una interfaz sencilla y eficiente para crear gráficos estadísticos de alta calidad. A continuación, exploraremos dos tipos de gráficos básicos: los gráficos de distribución y los gráficos de relación.

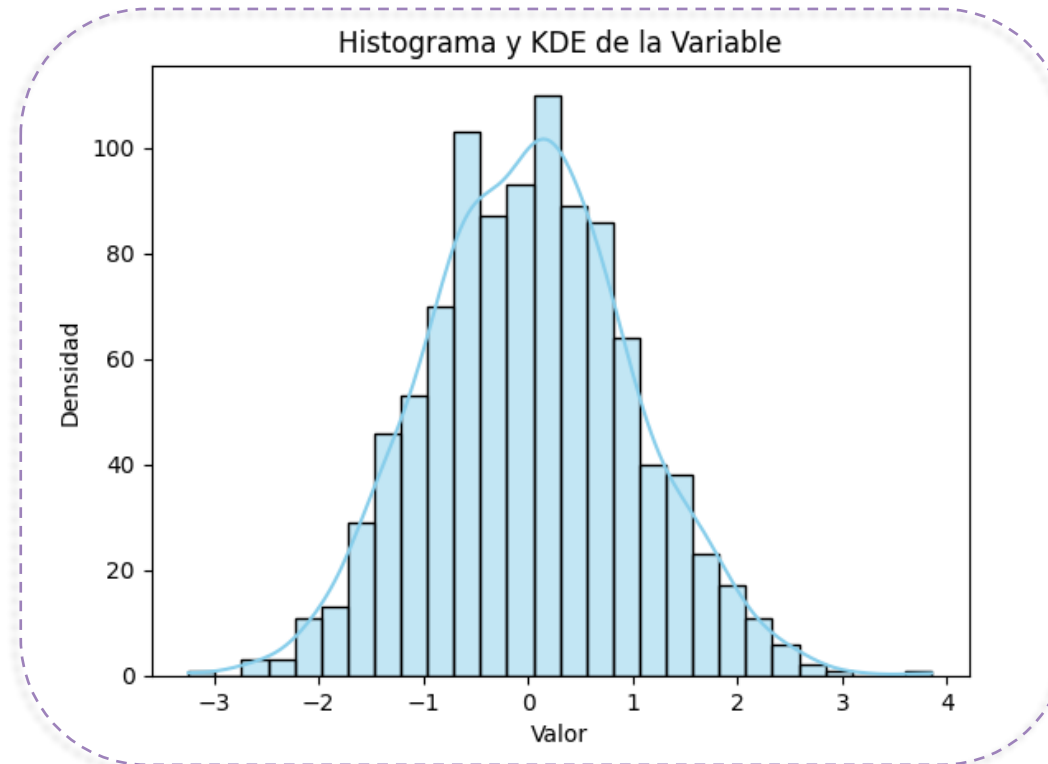
Gráficos de Distribución

Histogramas: Los histogramas son útiles para visualizar la distribución de una variable numérica. Seaborn simplifica la creación de histogramas con la función **sns.histplot()**.



GRÁFICOS BÁSICOS

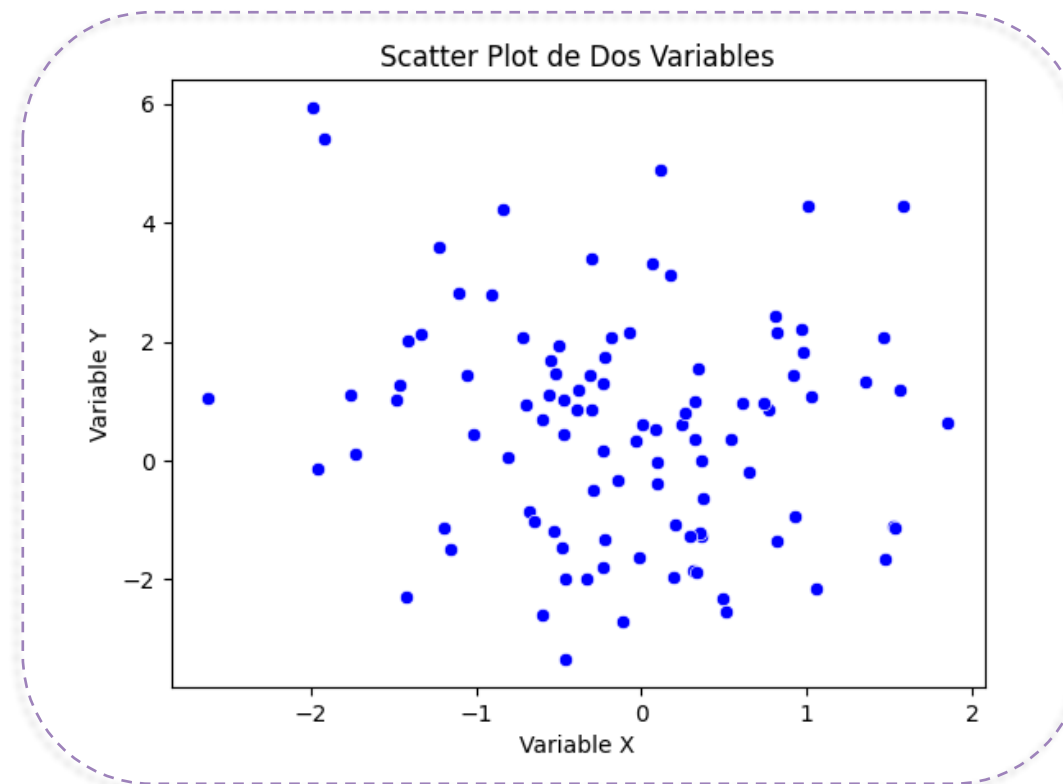
KDE (Kernel Density Estimation): El KDE proporciona una estimación suave de la distribución de los datos. Seaborn permite trazarlo junto con el histograma para obtener una visualización más completa.



GRÁFICOS BÁSICOS

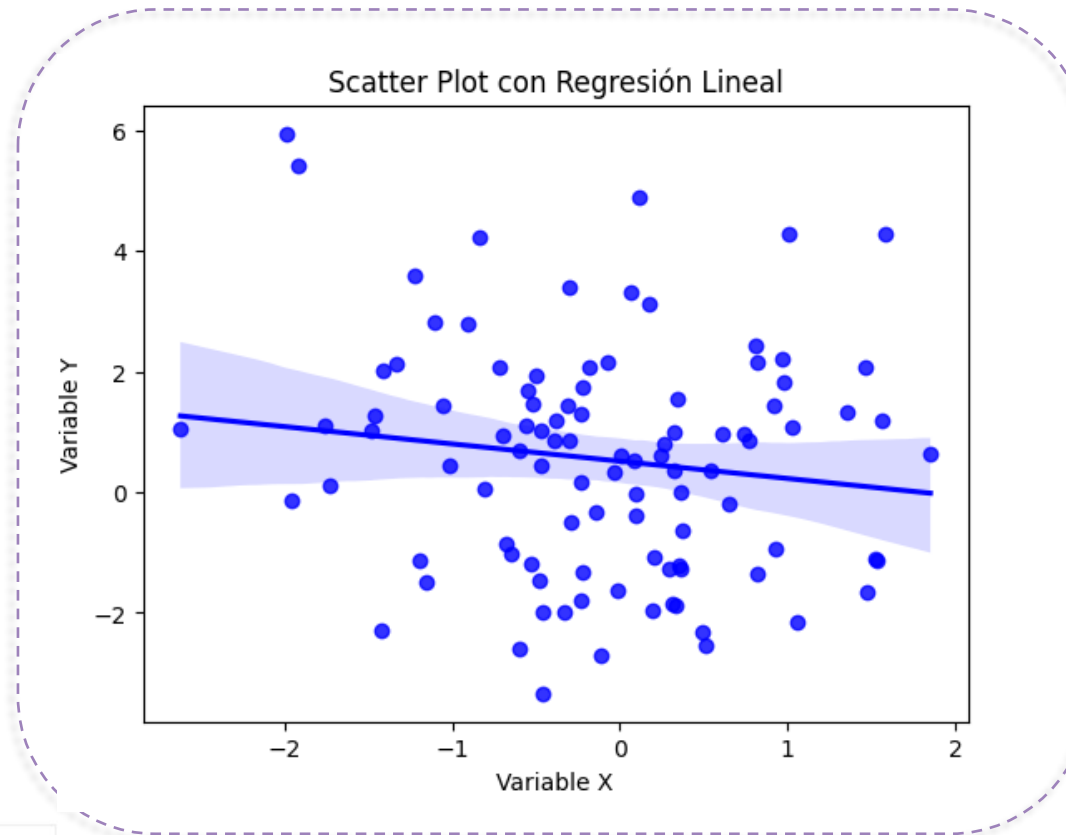
Gráficos de Relación

Scatter Plots: Los scatter plots son esenciales para visualizar la relación entre dos variables numéricas. Seaborn facilita su creación con **sns.scatterplot()**.



GRÁFICOS BÁSICOS

Regresiones Lineales: Para explorar tendencias y relaciones lineales, Seaborn proporciona **sns.regplot()**. Este gráfico también muestra una línea de regresión lineal junto con el scatter plot.



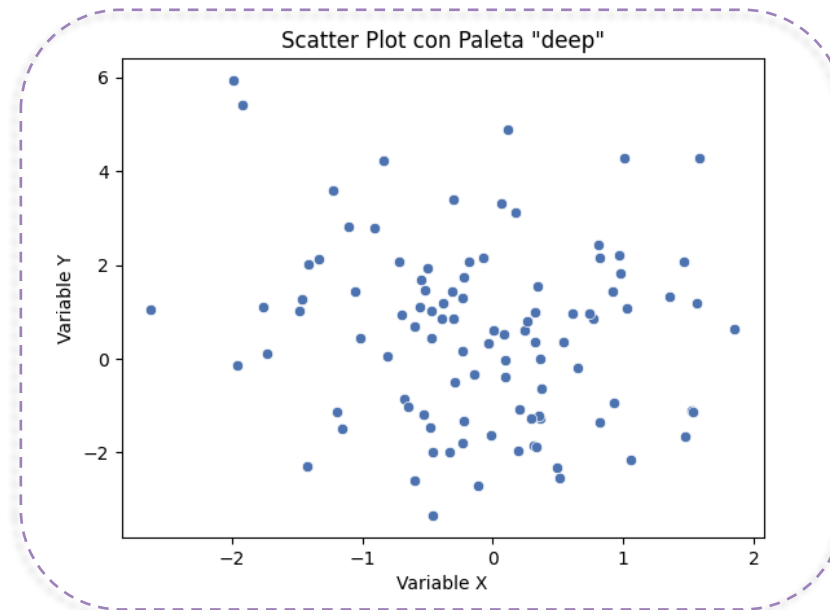
PERSONALIZACIÓN Y ESTILO

Seaborn ofrece varias opciones de personalización y estilos para adaptar tus visualizaciones a tus necesidades y preferencias específicas.

Personalización de Paletas de Colores

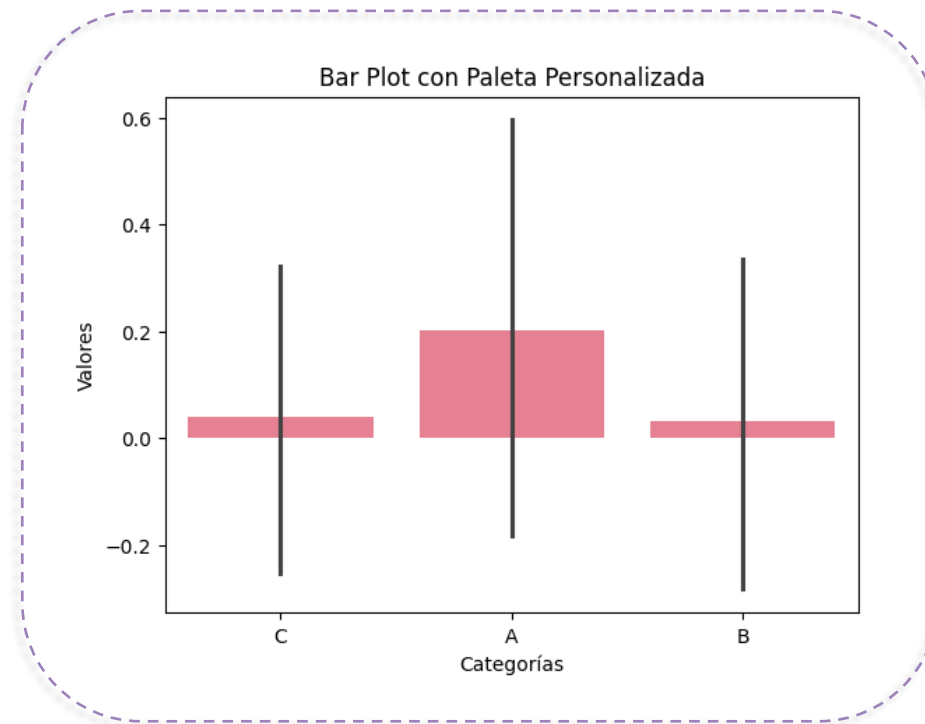
Paletas Predeterminadas

Seaborn proporciona paletas de colores predefinidas que puedes utilizar para personalizar el aspecto de tus gráficos. Algunas de las paletas comunes incluyen "deep", "muted", "bright", "pastel", entre otras.



PERSONALIZACIÓN Y ESTILO

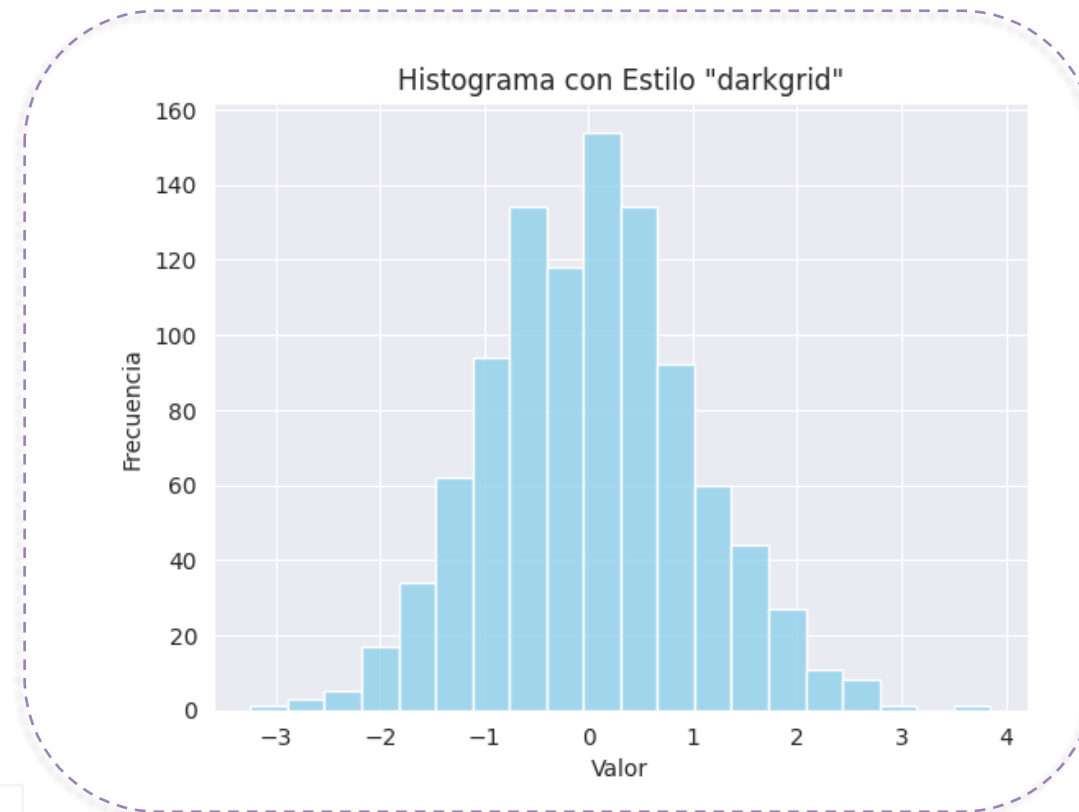
Personalización de Paletas: Puedes crear y personalizar tus propias paletas de colores para adaptarlas al contexto de tu visualización.



PERSONALIZACIÓN Y ESTILO

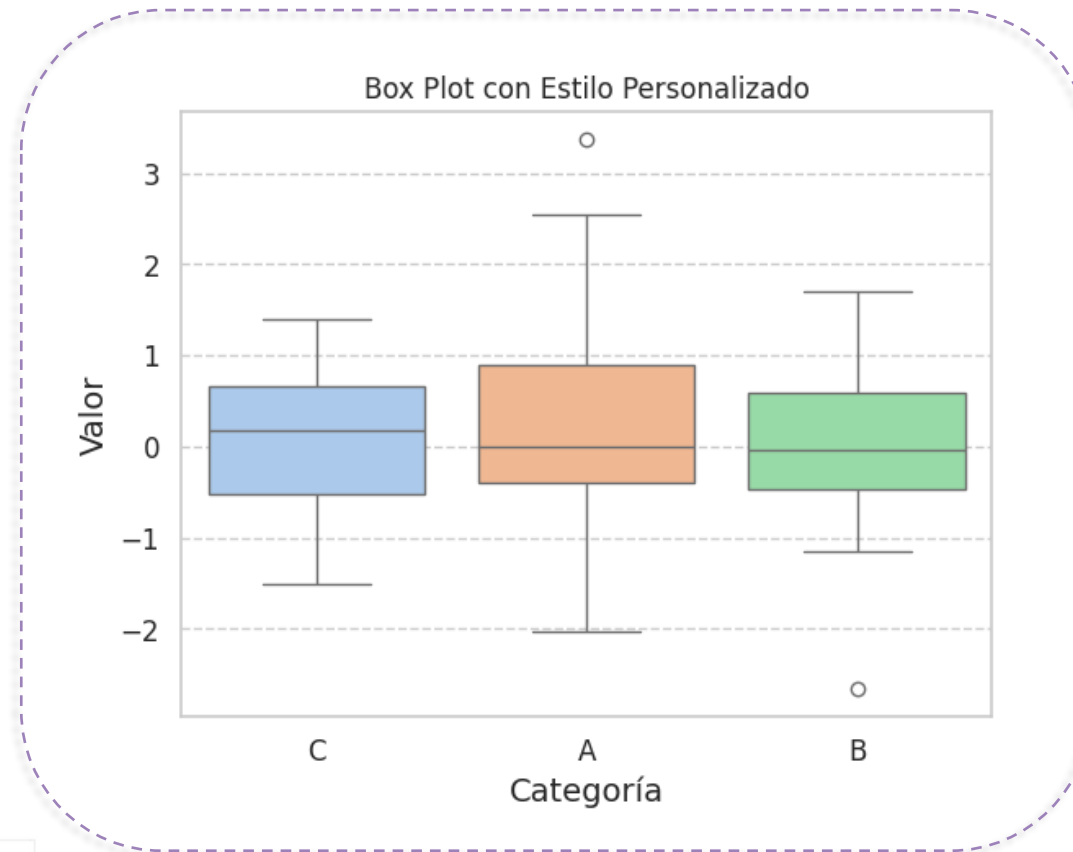
Ajuste de Estilos

Estilos Predeterminados: Seaborn también proporciona diferentes estilos predeterminados que afectan la apariencia general de los gráficos. Puedes elegir entre estilos como "darkgrid", "whitegrid", "dark", "white", y "ticks".



PERSONALIZACIÓN Y ESTILO

Personalización de Estilos: Además de los estilos predefinidos, puedes personalizar aspectos específicos del estilo, como el tamaño de la fuente, los colores de fondo y más.

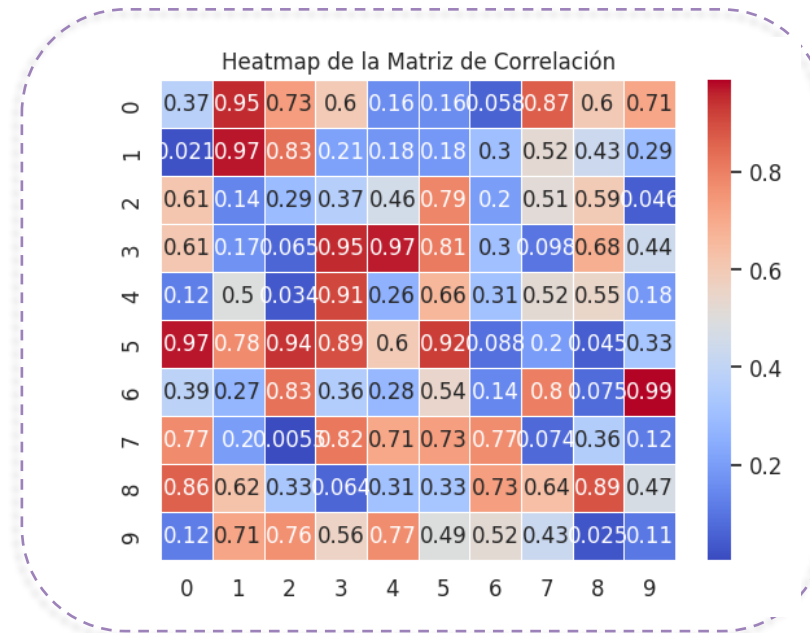


GRÁFICOS AVANZADOS

Seaborn ofrece herramientas avanzadas para visualizar datos de manera efectiva. A continuación, exploraremos Heatmaps, Gráficos Categóricos y Facet Grids.

Heatmaps

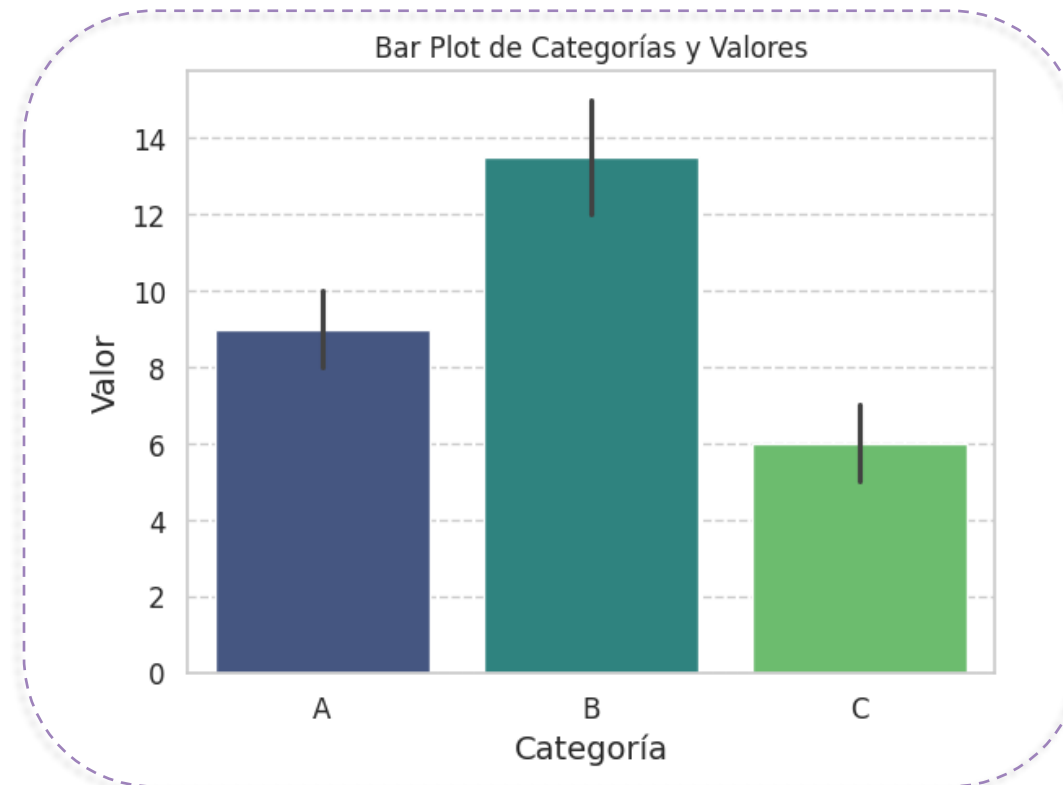
Matrices de Correlación: Los heatmaps son útiles para visualizar matrices de correlación. En el siguiente ejemplo, utilizaremos un DataFrame con datos ficticios, donde , **annot=True** agrega anotaciones con los valores de cada celda, y **cmap='coolwarm'** establece la paleta de colores.



GRÁFICOS AVANZADOS

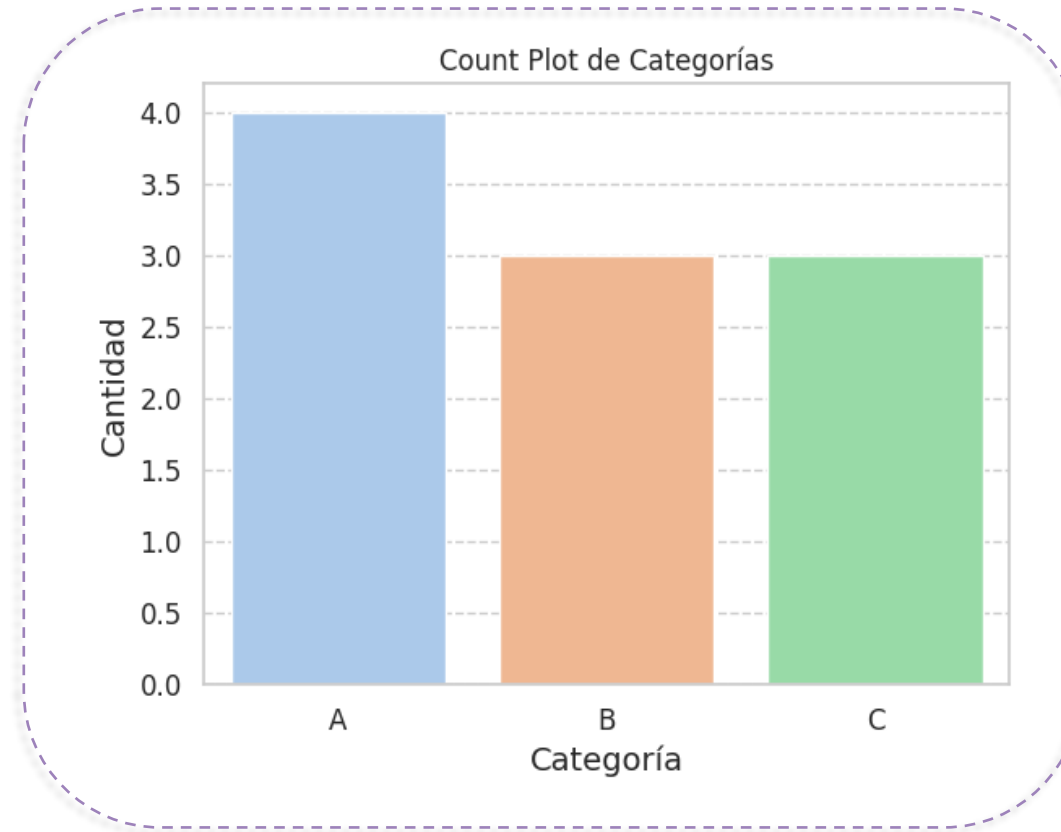
Gráficos Categóricos

Bar Plots: Los bar plots son útiles para comparar cantidades entre diferentes categorías.



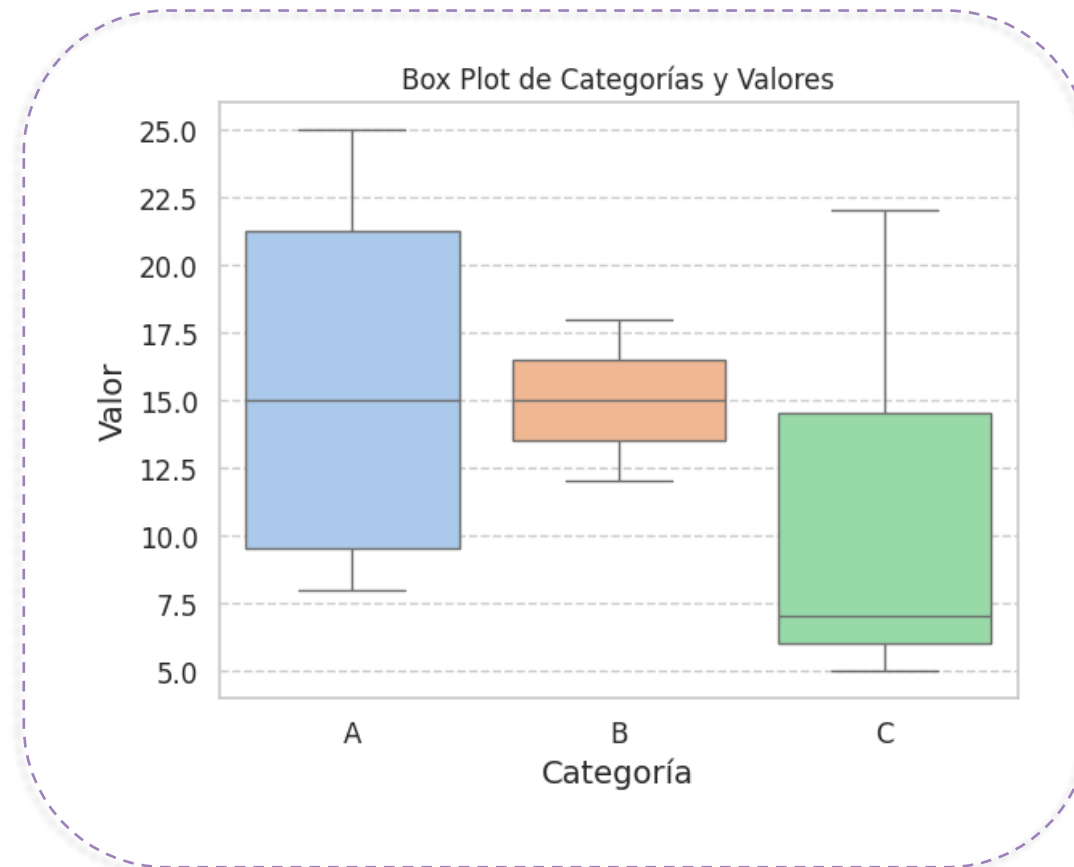
GRÁFICOS AVANZADOS

Count Plots: Los count plots son útiles para visualizar la distribución de datos categóricos.



GRÁFICOS AVANZADOS

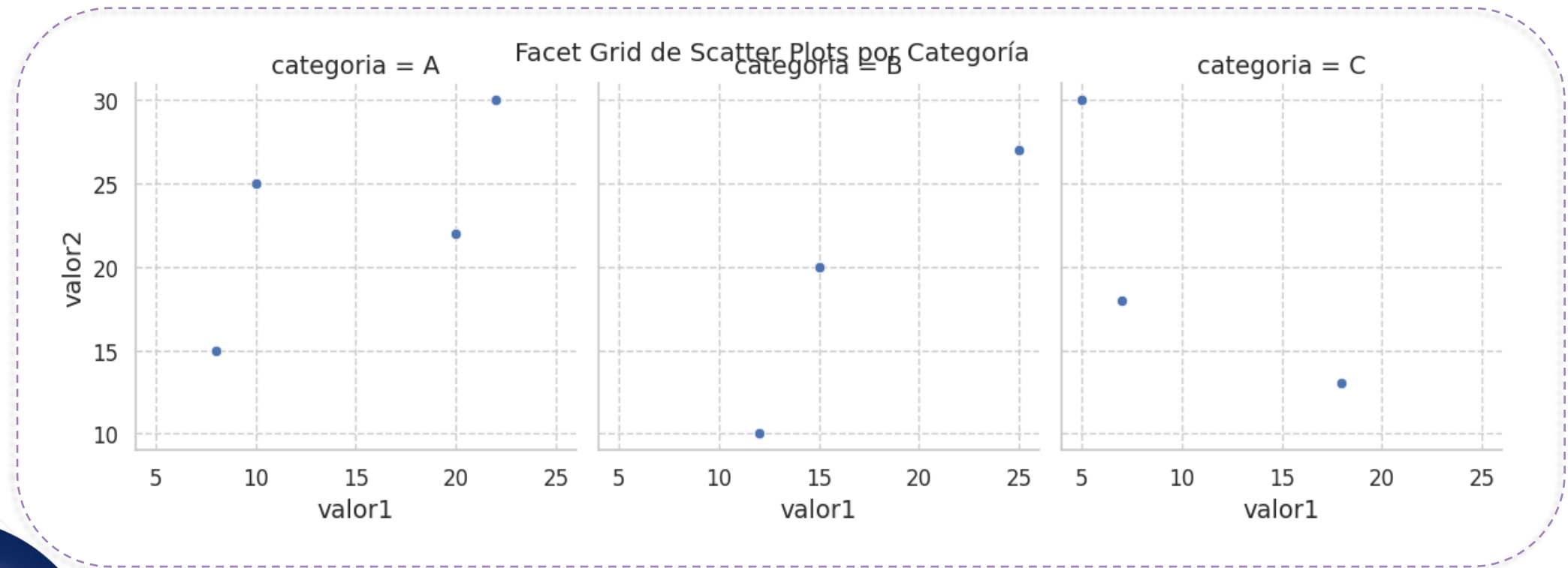
Box Plots: Los box plots son útiles para visualizar la distribución y los valores atípicos en datos categóricos.



GRÁFICOS AVANZADOS

Facet Grids

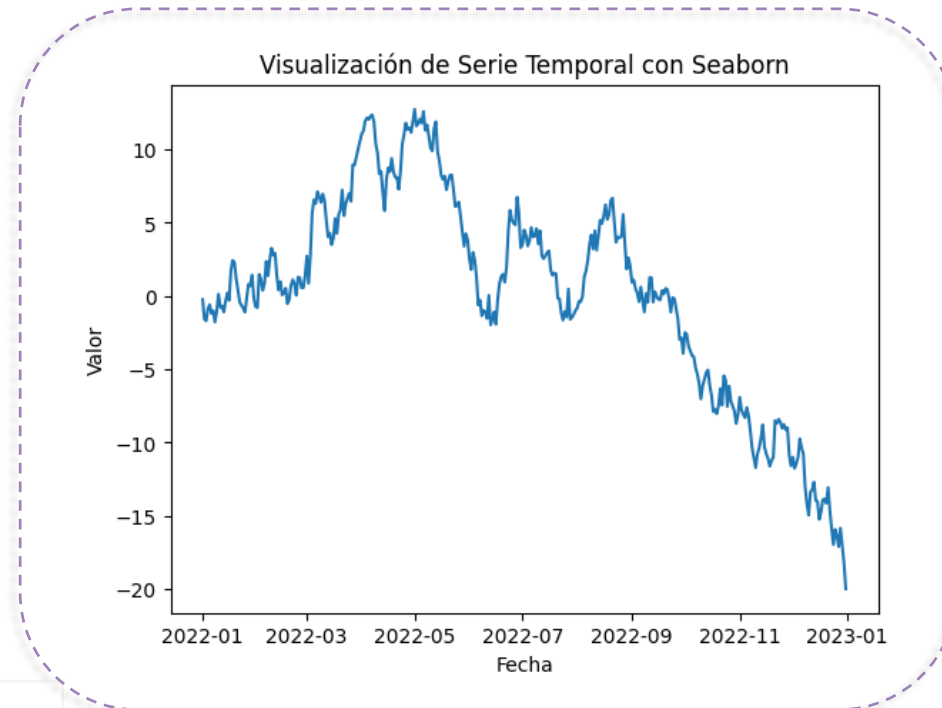
Creación de Múltiples Gráficos: Facet Grids permiten crear múltiples gráficos para comparaciones detalladas. En este ejemplo, crearemos un Facet Grid con subplots basados en una variable categórica.



VISUALIZACIONES ESPECÍFICAS

Uso de Seaborn para Representar Series Temporales

Seaborn es una biblioteca poderosa para la visualización de datos, pero su enfoque principal no es la manipulación de series temporales. Sin embargo, puede ser útil para representar tendencias y patrones a lo largo del tiempo. Aquí un ejemplo donde **lineplot** de Seaborn se utiliza para trazar la serie temporal. Asegúrate de tener tus fechas como índice en tu DataFrame o Serie temporal para que Seaborn pueda interpretarlas correctamente.

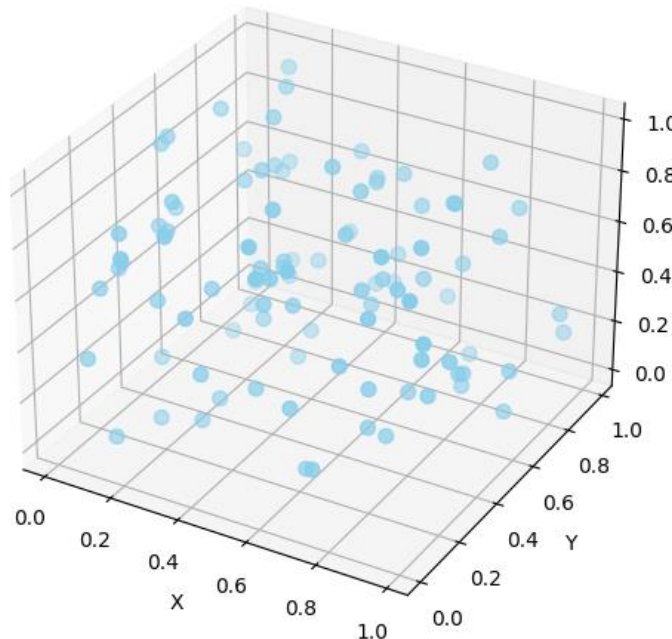


VISUALIZACIONES ESPECÍFICAS

Gráficos 3D con Seaborn

Seaborn no tiene funcionalidades específicas para visualizaciones tridimensionales como Matplotlib, pero puedes usar Matplotlib junto con Seaborn para crear gráficos 3D. Aquí hay un ejemplo básico:

Gráfico de Dispersión 3D con Seaborn y Matplotlib



¿Preguntas?