**Trabajo de investigación**

**Tecnico en Análisis de Datos**

**Modulo II**

**Librería Matplolib**

**Estudiante: Katherinne Viviana Sanchez Segura**

**Cedula:402290001**

**MATPLOTLIB**

**Introducción:**

Matplotlib es una librería de visualización de datos en Python ampliamente utilizada y altamente flexible la cual fue creada con el objetivo de proporcionar una herramienta que permitiera crear gráficos de alta calidad de manera sencilla y eficiente.

La estructura central de Matplotlib es pyplot el cual proporciona una interfaz que te permite crear y personalizar gráficos. Los gráficos se construyen a partir de objetos como figuras, ejes (x,y) y elementos gráficos individuales. Matplotlib también permite la incorporación de texto, anotaciones y etiquetas, lo que facilita la comprensión y la presentación de los datos visualizados.

**Matplotib- Python**

Para hacer uso de Matplotlib se requiere conocimientos de programación en Python y se debe tomar en cuenta que deende de la librería NumPy, su aplicación en Python nos permite el uso de la misma en ciencias de Datos y Machine Learning

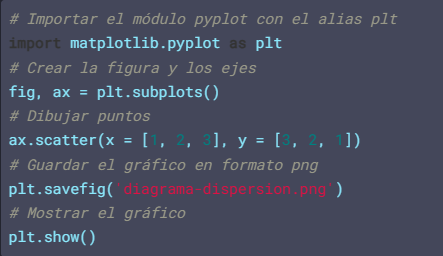
**Tipos de gráficos que podemos crear usando la librería Matplotlib**

* Diagramas de Barras: Se utilizan para comparar diferentes categorías o conjuntos de datos, nos permite visualizar relaciones entre variables y las diferencias entre valores.
* Histograma: Muestra la distribución de datos numéricos en intervalos (bins). Ayuda a comprender la frecuencia y densidad de los valores en un conjunto de datos continuos.
* Diagramas de Sectores: También conocidos como gráficos de pastel, representan proporciones de un todo. Son ideales para mostrar cómo una categoría contribuye a la totalidad. Se recomienda utilizarlos cuando no se tienen tantos valores que comparar.
* Diagramas de Caja y Bigotes: Se utilizan para representar estadísticas descriptivas de un conjunto de datos, como la mediana, cuartiles y posibles valores atípicos.
* Diagramas de Dispersión o Puntos: Muestran la relación entre dos variables numéricas. Cada punto en el gráfico representa una observación, lo que permite identificar patrones, tendencias y correlaciones.
* Diagramas de Líneas: Se utilizan para representar la evolución de una variable a lo largo del tiempo o algún otro orden secuencial. Son útiles para identificar tendencias a lo largo de un período.

**Pasos para la creación de gráficos**

1. Importar el módulo pyplot.
2. Definir la figura que contendrá el gráfico, donde se dibujará y los ejes sobre los que se dibujarán los datos. Para ello se utiliza la función subplots().
3. Dibujar los datos sobre los ejes, se utilizaran distintas funciones para definir la gráfica que queremos diseñar .
4. Personalizar el gráfico, se utilizaran funciones que nos permitirá añadir un título, una leyenda, cambiar colores o personalizar los ejes.
5. Guardar el gráfico -> savefig().
6. Mostrar el gráfico-> show().

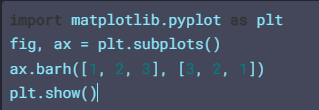
Ejemplo:



**Funciones de matplotlib**

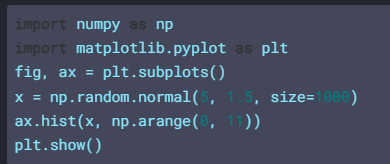
* plt.plot(): Crea gráficos de líneas y puntos para visualizar datos en 2D. Puede personalizarse con colores, estilos de línea y marcadores.
* plt.scatter(): Crea gráficos de dispersión, ideales para mostrar la relación entre dos conjuntos de datos. Permite personalizar los marcadores y los colores.
* plt.bar(): Genera gráficos de barras verticales / horizontales para comparar categorías diferentes. Puede ser útil para representar datos categóricos.

Verticales Horizontales

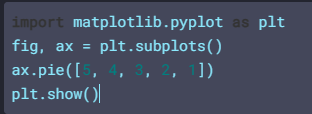
Texto

Descripción generada automáticamente

* plt.hist(): Crea histogramas para visualizar la distribución de datos numéricos en intervalos. Permite ajustar el número de intervalos (bins) y personalizar la apariencia.



* plt.pie(): Genera gráficos pastel (diagramas de sectores) para mostrar proporciones relativas. Permite destacar secciones y agregar etiquetas.



* plt.boxplot(): Crea diagramas de caja y bigotes para mostrar la distribución de un conjunto de datos y resaltar valores atípicos.

Texto

Descripción generada automáticamente

* plt.violinplot(): Genera diagramas de violín, que combinan diagramas de caja y gráficos de densidad para proporcionar una vista más completa de la distribución.

Texto

Descripción generada automáticamente

* plt.scatter(): Crea gráficos de dispersión para visualizar la relación entre dos variables numéricas. Puede ser útil para identificar correlaciones.
* plt.plot() : Crea gráficos de líneas para representar datos secuenciales a lo largo del tiempo u otro orden.
* plt.fill\_between() y plt.fill\_betweenx(): Rellena áreas entre líneas en gráficos de líneas o áreas para resaltar la contribución relativa de variables en el tiempo.
* plt.contour() y plt.contourf(): Genera gráficos de contorno y mapas de calor para representar datos 2D y 3D. En el módulo pyplot de la biblioteca matplotlib se utiliza para trazar contornos.

Texto

Descripción generada automáticamente

* plt.imshow(): Muestra imágenes o matrices como imágenes utilizando mapas de colores.

Texto

Descripción generada automáticamente

* plt.colorbar(): Agrega una barra de color a un gráfico para indicar la relación entre colores y valores en un mapa de calor, por ejemplo.
* plt.title(), plt.xlabel(), plt.ylabel(): Agrega títulos y etiquetas a los ejes para proporcionar contexto a las visualizaciones.

**Matplotlib y su importancia en Ciencia de Datos y Machine Learning**

En la ciencia de datos y machine learning nos permite desarrollar visualizaciones que resultan esenciales, además es una herramienta útil para limpieza de datos, análisis de datos y comunicación de resultados principalmente.

**Bibliografía**

Junio 8, 2023 por Miguel Sotaquirá

Alberca, A. S. (2020, 4 octubre). *La librería Matplotlib | Aprende con Alf*. Aprende con Alf. <https://aprendeconalf.es/docencia/python/manual/matplotlib/>

*MatplotLib — visualization with Python*. (s. f.). <https://matplotlib.org/>

KeepCoding, R. (2022, 30 noviembre). ¿Qué es Matplotlib y cómo funciona? | KeepCoding Bootcamps. *KeepCoding Bootcamps*. https://keepcoding.io/blog/que-es-matplotlib-y-como-funciona/