

Clase 6: Visualización de información

FM849 - Programación Científica para Proyectos de Inteligencia Artificial (IA)

7 de enero de 2026

Visualización de información

- ▶ La información puede almacenarse en diferentes formatos: **grafos, arboles, tablas, diccionarios, entre otros.**

Visualización de información

- ▶ La información puede almacenarse en diferentes formatos: **grafos, arboles, tablas, diccionarios, entre otros.**
- ▶ En este curso nos enfocaremos en datos tabulares (tablas). Este es el formato mas simple y utilizado.

Visualización de información

- ▶ La información puede almacenarse en diferentes formatos: **grafos, arboles, tablas, diccionarios, entre otros.**
- ▶ En este curso nos enfocaremos en datos tabulares (tablas). Este es el formato mas simple y utilizado.
- ▶ En la clase anterior vimos como abrir un datasets de este tipo usando Pandas. Ahora, la idea es visualizar estos datos.

Visualización de información

- ▶ La información puede almacenarse en diferentes formatos: **grafos, arboles, tablas, diccionarios, entre otros.**
- ▶ En este curso nos enfocaremos en datos tabulares (tablas). Este es el formato mas simple y utilizado.
- ▶ En la clase anterior vimos como abrir un datasets de este tipo usando Pandas. Ahora, la idea es visualizar estos datos.

Pero antes de eso, por que nos gustaria visualizar los datos ?

- ▶ Como vimos en la clase de estadística, los datasets pueden ser bastante complejos para analizar dato a dato.

Visualización de información

- ▶ La información puede almacenarse en diferentes formatos: **grafos, arboles, tablas, diccionarios, entre otros.**
- ▶ En este curso nos enfocaremos en datos tabulares (tablas). Este es el formato mas simple y utilizado.
- ▶ En la clase anterior vimos como abrir un datasets de este tipo usando Pandas. Ahora, la idea es visualizar estos datos.

Pero antes de eso, por que nos gustaria visualizar los datos ?

- ▶ Como vimos en la clase de estadística, los datasets pueden ser bastante complejos para analizar dato a dato.
- ▶ Las tecnicas de visualización nos permiten encontrar patrones o tendencias en los datos.

Visualización de información

- ▶ La información puede almacenarse en diferentes formatos: **grafos, arboles, tablas, diccionarios, entre otros.**
- ▶ En este curso nos enfocaremos en datos tabulares (tablas). Este es el formato mas simple y utilizado.
- ▶ En la clase anterior vimos como abrir un datasets de este tipo usando Pandas. Ahora, la idea es visualizar estos datos.

Pero antes de eso, por que nos gustaria visualizar los datos ?

- ▶ Como vimos en la clase de estadística, los datasets pueden ser bastante complejos para analizar dato a dato.
- ▶ Las tecnicas de visualización nos permiten encontrar patrones o tendencias en los datos.
- ▶ Se puede decir que la visualización de datos nos permite contar una historia de estos.

Visualización de información

- ▶ La información puede almacenarse en diferentes formatos: **grafos, arboles, tablas, diccionarios, entre otros.**
- ▶ En este curso nos enfocaremos en datos tabulares (tablas). Este es el formato mas simple y utilizado.
- ▶ En la clase anterior vimos como abrir un datasets de este tipo usando Pandas. Ahora, la idea es visualizar estos datos.

Pero antes de eso, por que nos gustaria visualizar los datos ?

- ▶ Como vimos en la clase de estadística, los datasets pueden ser bastante complejos para analizar dato a dato.
- ▶ Las tecnicas de visualización nos permiten encontrar patrones o tendencias en los datos.
- ▶ Se puede decir que la visualización de datos nos permite contar una historia de estos.
- ▶ Esto nos ayuda a tomar decisiones sobre los datos.

Matplotlib

Matplotlib es la librería mas popular para visualizar datos en Python. Esta libreria tiene funciones acopladas al paquete de Pandas.

```
import matplotlib.pyplot as plt
```

Función	Uso	Ejemplo
plot	Gráfico de linea	plt.plot(valores)
scatter	Gráfico de puntos	plt.scatter(valores)
bar	Gráfico de barras	plt.bar(etiquetas, conteos)
boxplot	Gráfico de caja	plt.boxplot(valores)
histogram	Histograma	plt.hist(valores)
imshow	Histograma	plt.imshow(valores)

Tabla 1: Funciones en matplotlib.

Analizaremos cada una de estas funciones y veremos como aplicarlas en un dataset.

Matplotlib: plot

El gráfico de líneas se usa para visualizar la evolución de una variable continua, generalmente respecto al tiempo o al orden de observación.

- ▶ Se usan para analizar tendencias temporales y comparación la evoluación de variables.

Los ejes x e y representan los valores que adoptan las variables analizadas.

```
>>> y = [1, 3, 2, 5, 4]  
>>> plt.plot(y)  
>>> plt.ylabel("y")  
>>> plt.show()
```

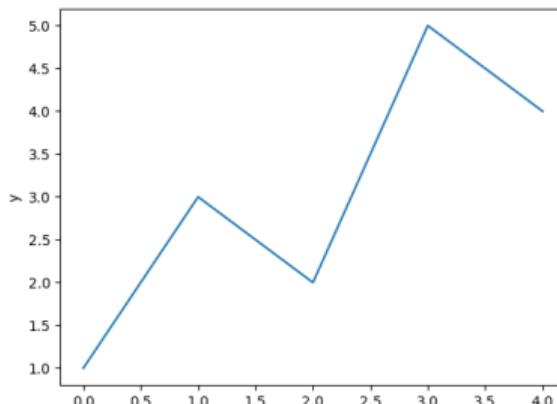


Figura 1: Grafico de línea en matplotlib.

Matplotlib: scatter

El gráfico de dispersión muestra la relación entre dos variables numéricas.

- ▶ Permite detectar correlaciones, identificar patrones, clusteres y encontrar outliers.

Los ejes x e y representan los valores que adoptan las variables analizadas.

```
x = [1, 2, 3, 4, 5]  
y = [2, 1, 3, 5, 4]
```

```
# plot de (1,2), (2,1), ... , (5,4)  
plt.scatter(x, y)  
plt.ylabel("y")  
plt.xlabel("x")  
plt.show()
```

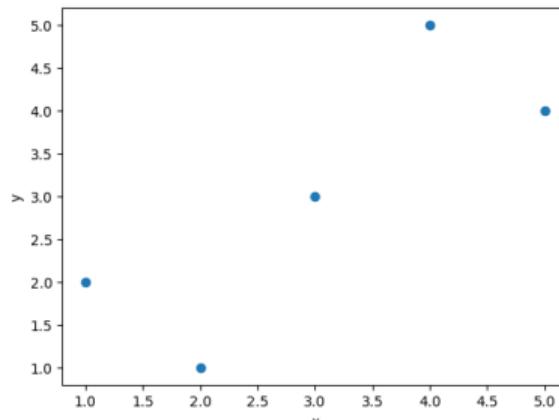


Figura 2: Grafico de dispersión en matplotlib..

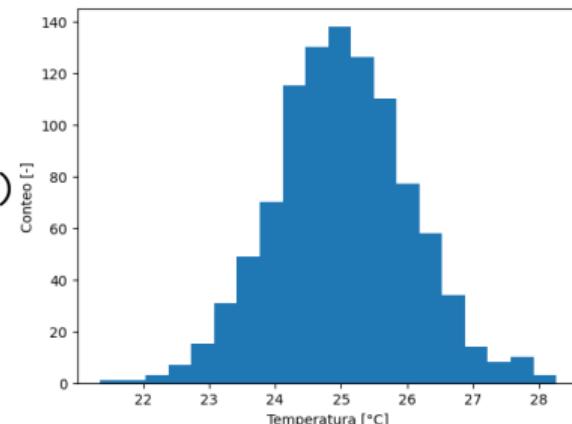
Matplotlib: hist

El histograma muestra la distribución de una variable numérica.

- ▶ Se utiliza para analizar la forma de la distribución, asimetría y concentración de valores.

En general, el eje x representa la variable analizada y el eje y el numero de veces que ocurre esa variable (frecuencia).

```
import numpy as np
temperatura_santiago = np.random.randn(1000)
plt.hist(temperatura_santiago, bins=20)
plt.xlabel("Temperatura [°C]")
plt.ylabel("Conteo [-]")
plt.show()
```



Matplotlib: boxplot

El boxplot resume una distribución mediante cuartiles y valores extremos.

- ▶ Se usa para comparar las distribuciones de diversas categorías en función de sus cuartiles y outliers.

Requiere un atributo categorico (eje x) y un atributo cuantitativo (eje y).

```
temperatura_santiago = np.random.randn(100)
plt.boxplot(temperatura_santiago)
plt.ylabel("Valor [°C]")
plt.xlabel("Temperatura en Santiago")
plt.show()
```

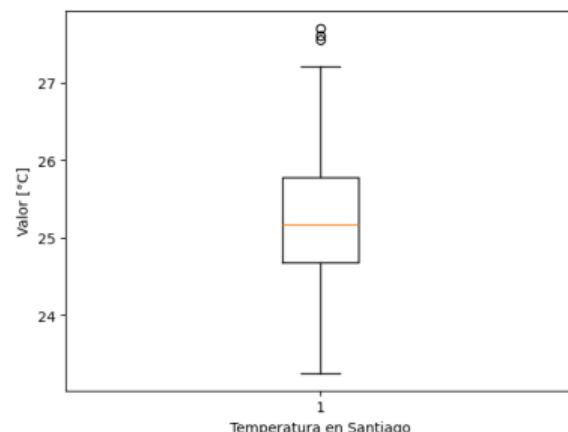


Figura 4: Boxplot en matplotlib.

Matplotlib: imshow

imshow se utiliza para visualizar datos matriciales como imágenes.

- Es util cuando queremos visualizar imágenes, matrices, mapas, entre otros.

Recibe una matriz con valores numericos como argumento.

```
import numpy as np
# matrix de 10 filas x 10 columnas
matriz = np.random.rand(10, 10)
plt.imshow(matriz)
plt.colorbar()
plt.show()
```

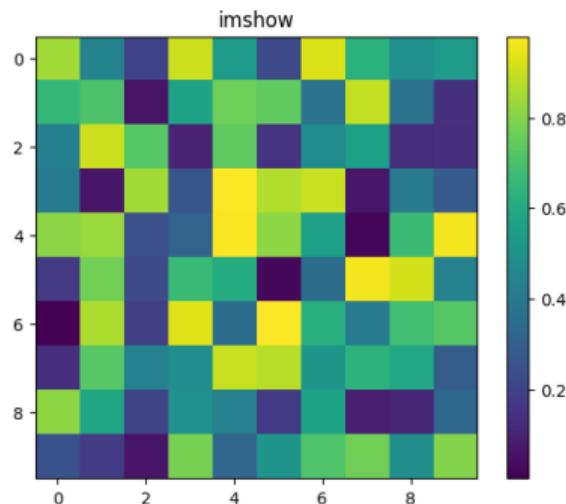


Figura 5: Imshow plot en matplotlib.

Aplicación sobre Iris Dataset

- ▶ Dataset con 150 datos tabulares.
- ▶ Cada instancia (fila) representa una flor.
- ▶ En este dataset tiene 5 columnas: **sepallength**, **sepalwidth**, **petallength**, **petalwidth**, **class**.
- ▶ Las clases (class) son categoricas: **setosa**, **versicolor**, **virginica**.

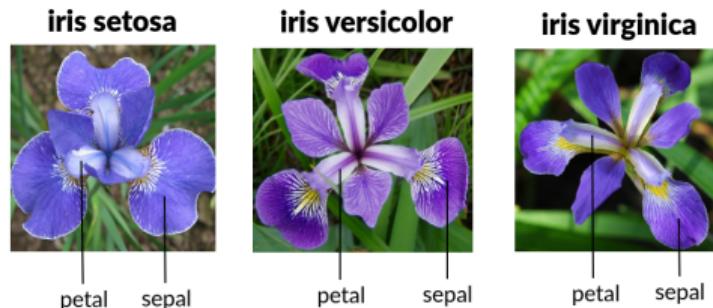


Figura 6: Features en Iris Dataset.

Seaborn

Seaborn es una librería de visualización que permite generar gráficos de manera simple, rápida y con muchos estilos. Con Matplotlib podemos lograr lo mismo pero requiere mayor trabajo.

```
import seaborn as sns
```

- ▶ **lineplot**: Gráfico de líneas con estimación estadística e intervalos de confianza.

```
sns.lineplot(data=df, x="x", y="y")
```

- ▶ **scatterplot**: Gráfico de dispersión con agrupación por color, tamaño o estilo.

```
sns.scatterplot(data=df, x="x", y="y", hue="clase")
```

- ▶ **histplot**: Histograma y densidad de una variable.

```
sns.histplot(data=df, x="x", kde=True)
```

- ▶ **boxplot**: Comparación de distribuciones mediante cajas.

```
sns.boxplot(data=df, x="grupo", y="valor")
```

- ▶ **pairplot**: Relación entre todas las variables numéricas.

```
sns.pairplot(df, hue="clase")
```

Referencias:

- ▶ Wes McKinney. (2022). Python for Data Analysis. Third Edition.
- ▶ https://seaborn.pydata.org/tutorial/function_overview.html