

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.**
- ▶ 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 dataset tabular 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 fila por fila.
- ▶ Las tecnicas de visualización nos permiten encontrar patrones o tendencias en los datos.
- ▶ La visualización con gráficos nos permite responder a preguntas asociadas a los datos.

Matplotlib vs Seaborn

En esta clase trabajaremos con dos paquetes para visualizar datos.

- ▶ **Matplotlib:** Librería de uso general para construir todo tipo de gráficos.
 - ▶ Es muy flexible, pero requiere más código.
 - ▶ Recordar importar: `import matplotlib.pyplot as plt`
 - ▶ Las funciones de matplotlib reciben los vectores de datos a plotear como argumentos (no un DataFrame).
- ▶ **Seaborn:** Librería orientada al análisis de datos.
 - ▶ Está pensada para trabajar directamente con **DataFrames**.
 - ▶ Recordar importar: `import seaborn as sns`

Matplotlib vs Seaborn

En Matplotlib, los gráficos se construyen indicando:

- ▶ Directamente los **datos a graficar** (arreglos, listas o Series).
- ▶ No existe un argumento data que agrupe los datos.
- ▶ Los argumentos (x, y, etc.) corresponden a los **arreglos de valores** y no a los nombres de las columnas.

En Seaborn, los gráficos se construyen indicando:

- ▶ data: el **DataFrame** que contiene los datos.
- ▶ El resto de los argumentos (x, y, hue, etc.) son los **nombres de las columnas** de ese DataFrame que se quieren graficar.

Esto permite crear gráficos de forma simple, clara y eficiente, sin necesidad de extraer manualmente los datos como ocurre habitualmente en Matplotlib.

Gráficos de Línea

- ▶ Los ejes x e y representan los valores que adoptan las variables analizadas.
- ▶ Se usan para analizar tendencias temporales y comparación la evolución de variables.
- ▶ Las marcas son puntos que se conectan por líneas.
- ▶ La posición vertical expresa un valor cuantitativo, mientras la posición horizontal contiene las llaves ordenadas.

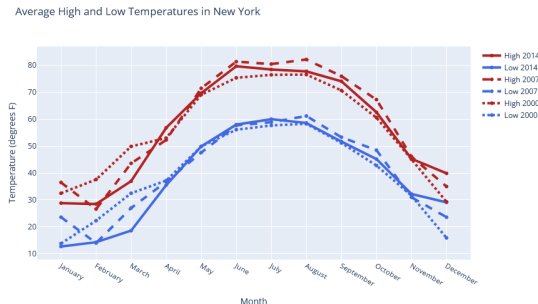


Figura 1: Gráfico de línea mostrando la evolución de la temperatura un año.

Ejemplo de gráfico de línea (Matplotlib)

Con `plt.plot`:

```
import matplotlib.pyplot as plt
>>> x = [0, 1, 2, 3, 4]
>>> y = [1, 3, 2, 5, 4]
>>> # plot (0,1), (1,3), ... (4,4)
>>> plt.plot(x,y)
>>> plt.ylabel("y")
>>> plt.show()
```

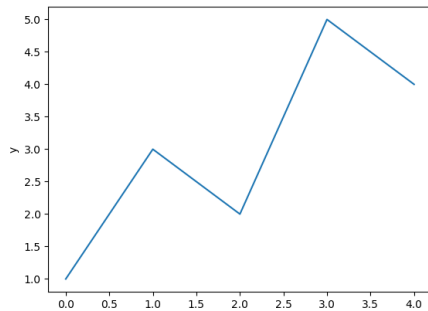


Figura 2: Gráfico de línea en matplotlib.

Mas ejemplos: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.plot.html

Ejemplo de gráfico de línea (Seaborn)

Con `sns.lineplot`:

```
>>> import seaborn as sns
>>> # DataFrame de vuelos
>>> df = sns.load_dataset("flights")
>>> sns.lineplot(data=df,
                  x="year",
                  y="passengers")
```

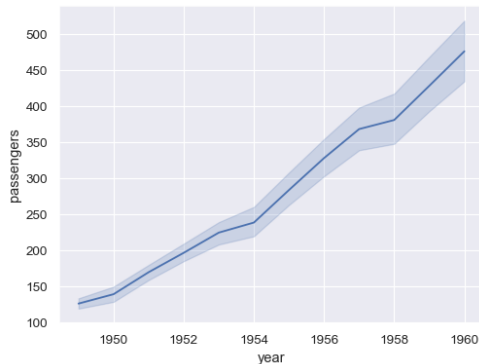


Figura 3: Gráfico de línea que muestra la evolución de vuelos según el año. El área sombreada representa un intervalo de confianza.

Mas ejemplos: <https://seaborn.pydata.org/generated/seaborn.lineplot.html>

Gráfico de Dispersión

- El gráfico de dispersión muestra la relación entre dos variables numéricas.
- Permite detectar correlaciones, identificar patrones, clusteres y encontrar outliers.
- Los canales son las posiciones: horizontal y vertical. Los ejes x e y representan los valores que adoptan las variables analizadas.

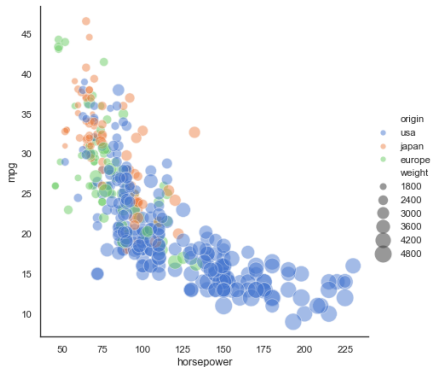


Figura 4: Gráfico de dispersión entre millas por galon (MPG) y caballos de fuerza (horsepower) en autos.

Ejemplo de gráfico de dispersión (Matplotlib)

Con `plt.scatter`:

```
>>> import matplotlib.pyplot as plt
>>> 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()
```

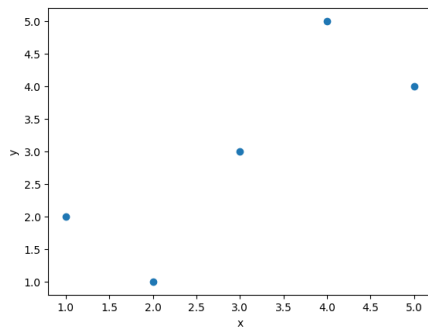


Figura 5: Gráfico de dispersión en Matplotlib.

Mas ejemplos: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.scatter.html

Ejemplo de gráfico de dispersión (Seaborn)

Con `sns.scatterplot`:

```
>>> import seaborn as sns
>>> #DataFrame de propinas
>>> df = sns.load_dataset("tips")
>>> sns.scatterplot(data=df,
                    x="total_bill",
                    y="tip")
```

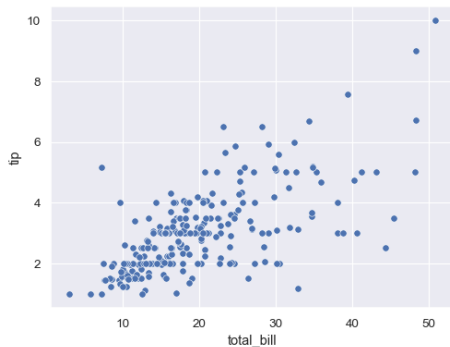


Figura 6: Gráfico de dispersión entre pago total (total bill, eje x) y propina (tip, eje y).

Mas ejemplos: <https://seaborn.pydata.org/generated/seaborn.scatterplot.html>

Grafico de Barras.

- ▶ Requiere 1 atributo categórico y 1 cuantitativo.
- ▶ Los canales que codifican información incluyen: el largo de la barra para expresar un valor cuantitativo y una separación en el espacio para representar otra categoría.
- ▶ Se puede usar para comparar categorías y encontrar casos extremos.

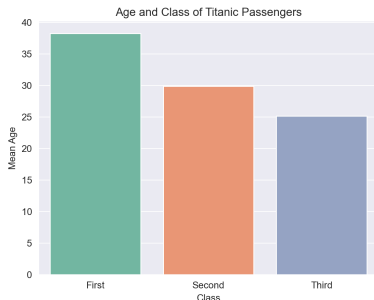


Figura 7: Edad promedio según la clase en el barco Titanic.

Ejemplo de gráfico de barras (Matplotlib)

Con `plt.bar`:

```
>>> import matplotlib.pyplot as plt
>>> categorias = ["A", "B", "C", "D"]
>>> valores = [23, 45, 12, 30]
>>> plt.bar(categorias, valores)
>>> plt.ylabel("Valor")
>>> plt.xlabel("Categoría")
>>> plt.show()
```

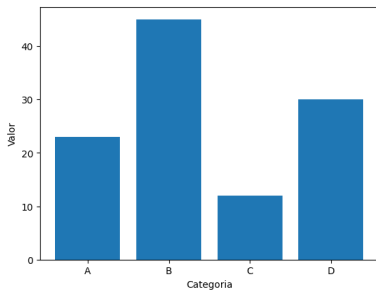


Figura 8: Gráfico de barras generado en Matplotlib.

Mas ejemplos: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.bar.html

Ejemplo de gráfico de barras (Seaborn)

Con `sns.barplot`:

```
>>> import seaborn as sns
>>> #DataFrame de pingüinos
>>> df = sns.load_dataset("penguins")
>>> sns.barplot(df,
                x="island",
                y="body_mass_g")
```

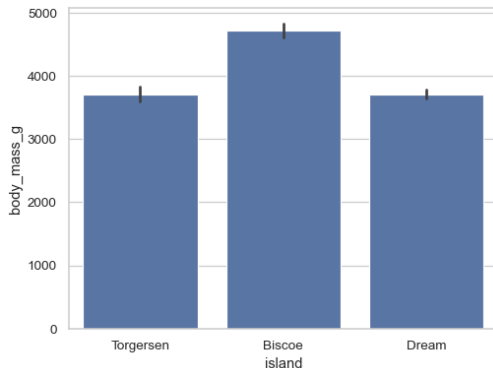


Figura 9: Gráfico de barras mostrando el peso promedio de pingüinos (variable numerica) en islas (variable categorica).

Mas ejemplos: <https://seaborn.pydata.org/generated/seaborn.barplot.html>

Histograma

- ▶ Un histograma representa una variable cuantitativa agrupando los datos en intervalos.
- ▶ Cada barra corresponde a un rango de valores del eje horizontal.
- ▶ Permite analizar la forma de la distribución, su asimetría y concentración.
- ▶ El eje y muestra la frecuencia (número de observaciones) en cada intervalo.

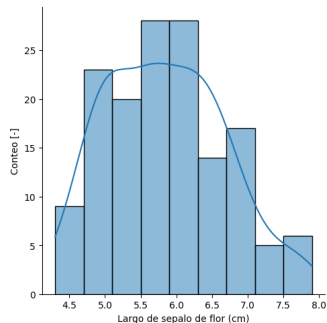


Figura 10: Histograma que muestra la distribución del largo del sépalo en dataset Iris.

Ejemplo de histograma (Matplotlib)

Con `plt.hist`:

```
>>> import matplotlib.pyplot as plt
>>> import numpy as np
>>> temperatura = [2,33,24,25,12,20,0,23]
>>> plt.hist(temperatura)
>>> plt.xlabel("Temperatura [°C]")
>>> plt.ylabel("Conteo [-]")
>>> plt.show()
```

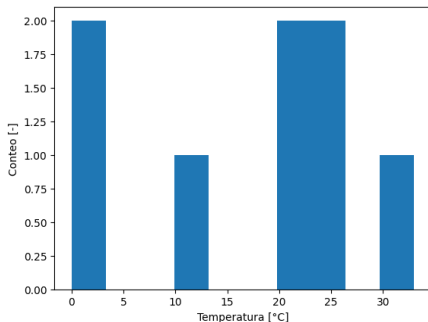


Figura 11: Histograma en matplotlib.

Más ejemplos: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.hist.html

Ejemplo de histograma (Seaborn)

Con `sns.histplot`:

```
>>> import seaborn as sns
>>> # DataFrame de pingüinos
>>> df = sns.load_dataset("penguins")
>>> sns.histplot(data=df,
                  x="flipper_length_mm")
```

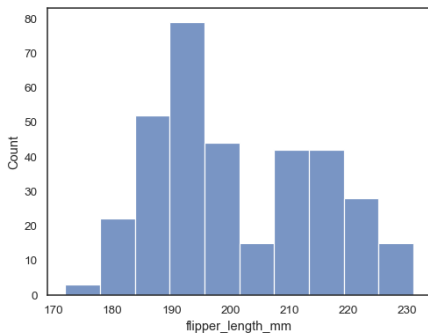


Figura 12: Histograma sobre largo de aleta de pingüinos (`flipper_length_mm`).

Más ejemplos: <https://seaborn.pydata.org/generated/seaborn.histplot.html>

Matplotlib: boxplot

- El boxplot resume una distribución considerando 5 valores obtenidos de la serie de datos: mínimo, cuartil 1 (Q1), cuartil 2 (mediana), cuartil 3 (Q3) y máximo.
- Se usa para comparar las distribuciones entre columnas de un DataFrame en función de sus cuartiles y outliers.
- Requiere un atributo categorico (eje x) y un atributo cuantitativo (eje y).
- El principal canal de codificación es el largo de la caja (rango intercuartil) y las líneas horizontales (asociadas a outliers).

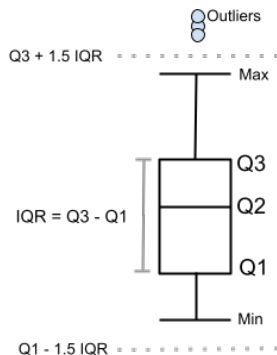


Figura 13: Componentes de boxplot. IQR es el rango intercuartil.

Matplotlib: boxplot

- El boxplot resume una distribución considerando 5 valores obtenidos de la serie de datos: mínimo, cuartil 1 (Q1), cuartil 2 (mediana), cuartil 3 (Q3) y máximo.
- Se usa para comparar las distribuciones entre columnas de un DataFrame en función de sus cuartiles y outliers.
- Requiere un atributo categorico (eje x) y un atributo cuantitativo (eje y).
- El principal canal de codificación es el largo de la caja (rango intercuartil) y las líneas horizontales (asociadas a outliers).

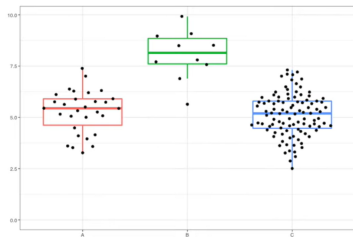


Figura 14: Ejemplo de boxplot para comparar categorías (A, B y C).

Ejemplo de boxplot en Matplotlib

Con `plt.boxplot`:

```
>>> import matplotlib.pyplot as plt
>>> temperatura = [2,33,24,25,12,20,0,23,50]
>>> plt.boxplot(temperatura)
>>> plt.ylabel("Temperatura")
```

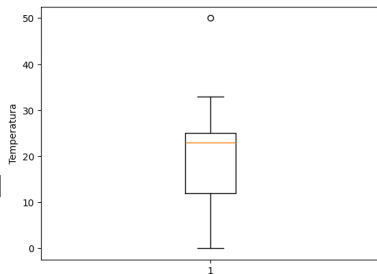


Figura 15: Boxplot generado considerando la variable temperatura.

Mas ejemplos: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.boxplot.html

Ejemplo de boxplot en Seaborn

Con `sns.boxplot`:

```
>>> import matplotlib.pyplot as plt
>>> import seaborn as sns
>>> df = sns.load_dataset('titanic')
>>> sns.boxplot(data=df, x="class", y="age")
>>> plt.ylabel("Edad")
>>> plt.xlabel("Clase")
```

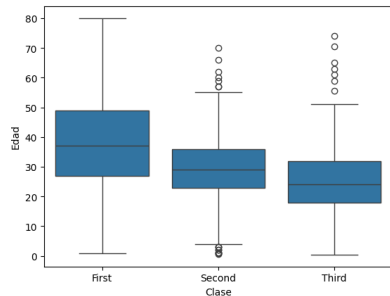


Figura 16: Distribuciones de edades según clase en embarcación Titanic.

Mas ejemplos: <https://seaborn.pydata.org/generated/seaborn.boxplot.html>

Matplotlib (Resumen)

```
import matplotlib.pyplot as plt # podemos importarla de esta forma
```

Función	Uso	Ejemplo
plot	Gráfico de línea	<code>plt.plot(valores)</code>
scatter	Gráfico de puntos	<code>plt.scatter(valores_x, valores_y)</code>
bar	Gráfico de barras	<code>plt.bar(etiquetas, valores)</code>
boxplot	Gráfico de caja	<code>plt.boxplot(valores)</code>
histogram	Histograma	<code>plt.hist(valores)</code>
imshow	Imshow	<code>plt.imshow(valores)</code>

Tabla 1: Funciones en matplotlib.

Seaborn (Resumen)

```
import seaborn as sns # podemos importarla de esta forma
```

Seaborn, a diferencia de matplotlib, recibe un dataframe como argumento. Adicionalmente se deben indicar que columnas del dataframe seran utilizadas en cada eje.

- ▶ **lineplot**: Gráfico de que muestra la relación entre la columna 'x' e 'y' de un dataframe.
`sns.lineplot(data=dataframe, x="x", y="y")`
- ▶ **scatterplot**: Gráfico de dispersión entre columna 'x' e 'y' de un dataframe.
`sns.scatterplot(data=dataframe, x="x", y="y")`
- ▶ **histplot**: Histograma de la columna 'x' de un dataframe.
`sns.histplot(data=dataframe, x="x")`
- ▶ **boxplot**: Comparación de distribuciones de la columna 'valor' en un valor.
`sns.boxplot(data=dataframe, x="grupo", y="valor")`
- ▶ **pairplot**: Relación entre todas las variables numéricas en un dataframe coloreadas segun la columna 'clase'.
`sns.pairplot(dataframe, hue="clase")`

Referencias:

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