**Visualización con Matplotlib: Diagrama de barras y márgenes de errores**

Iportamos librerías y además, nos apoyaremos en el dataset del titanic para los ejemplos:

**import** matplotlib.pyplot **as** plt

**import** numpy **as** np

**import** pandas **as** pd

df\_titanic **=** pd.read\_csv("./data/titanic.csv")

df\_titanic

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

**Gráficos de Barras y Count plots**

Los gráficos de barras se emplean para comparar datos de forma más general y cuando se emplean para comparar frecuencias también reciben el nombre de Count plots. La sintaxis es muy sencilla:

*# Datos de ejemplo*

categorias **=** ['Categoría A', 'Categoría B', 'Categoría C']

valores **=** [23, 45, 56]

plt.bar(categorias, valores);

Gráfico, Gráfico de barras

Descripción generada automáticamente

Si queremos obtener un gráfico de frecuencias se las tenemos que dar calculadas. Veamos cómo sería el gráfico de frecuencias absolutas y de frecuencias relativas para la columna "who" del titanic:

categorias **=** df\_titanic.who.value\_counts().index.to\_list()

frecuencias **=** df\_titanic.who.value\_counts().to\_list()

frec\_relativas **=** (df\_titanic.who.value\_counts(normalize **=** **True**) **\*** 100).to\_list()

fig,ax **=** plt.subplots(1,2, figsize **=** (4.5,3))

fig.subplots\_adjust(wspace **=** 0.5)

ax[0].bar(categorias, frecuencias)

ax[0].set\_title("Frecuencias Absolutas")

ax[0].set\_xlabel("Valores Who")

ax[0].set\_ylabel("Num. personas")

ax[1].bar(categorias, frec\_relativas, color **=** ["brown","red","steelblue"])

ax[1].set\_title("Frecuencias Relativas")

ax[1].set\_xlabel("Valores Who")

ax[1].set\_ylabel("% del total");

Gráfico, Gráfico de barras

Descripción generada automáticamente

**Visualizando Márgenes de Errores**

En general, si no vas a hacer un análisis de datos de mediciones no vas a necesitar este tipo de gráficas, pero son muy comunes en ámbitos científicos y para representar medidas que por las condiciones de medición o de los aparatos de medida quieren presentar el posible rango de error cometido al tomarlas. Por ejemplo, si necesitamos mucha precisión en la medida de la velocidad de un objeto en un experimento de física y nuestro aparato de medida dice 5 pero también que comete un error de 0.05, lo normal es representar ese 5 como un punto y una barra, denominada barra de error, centrada en la medida y de extremos el resultado de restar y sumar el error (4.95 y 5.05).

En la visualización de datos y resultados, mostrar estos errores de manera efectiva puede hacer que una gráfica transmita mucha más información completa.

x **=** np.linspace(0, 10, 50)

dy **=** 0.8 *# el error de nuestro aparato de medida*

y **=** np.sin(x) **+** dy **\*** np.random.randn(50) *#una función inventada con ese error (aleatorio por medida)*

plt.errorbar(x, y, yerr**=**dy, fmt**=**'.g');

Un dibujo de una persona

Descripción generada automáticamente con confianza baja

Además de estas opciones básicas, la función errorbar tiene muchas opciones para ajustar finamente las salidas. Usando estas opciones adicionales, puedes personalizar fácilmente la estética de tu gráfico de barras de error. A menudo encuentro útil, especialmente en gráficos concurridos, hacer que las barras de error sean más claras que los propios puntos:

plt.errorbar(x, y, yerr**=**dy, fmt**=**'o', color**=**'black', ecolor**=**'lightgray', elinewidth**=**3, capsize**=**3);

Gráfico, Gráfico de cajas y bigotes

Descripción generada automáticamente

Existen muchas más opciones, que puedes encontrar en la documentación de plt.errorbar.