

# Práctica Visualización de Datos II

Ainoa Iglesias Dasilva, 2024/25

[alu0101164403@ull.edu.es](mailto:alu0101164403@ull.edu.es)

Máster, Análisis de Datos Masivos

[GitHub de esta práctica](#)

---

## 1. Resumen: Why scientists need to be better at data visualization

En esta parte del artículo se habla del uso del color en las gráficas, destacando errores comunes en las visualizaciones, como percibimos los humanos los colores y cómo deberían usarse.

Los humanos percibimos los colores por su tono, saturación y luminosidad. Entre los errores de los que se habla están:

- Usar la escala de colores del arcoiris: ésta no representa adecuadamente los datos. Los colores no tienen un orden intuitivo, y puede interpretarse falsos picos por ejemplo, en los amarillos a pesar de estar en el centro de la escala.
- No tener en cuenta los colores como un conjunto, muy común en mapas de calor, esto provoca lo que se conoce como contraste simultáneo, un color puede verse distinto según los colores que lo rodean.
- Uso simultáneo de muchos colores, dificulta la lectura del gráfico y distrae.
- No tener en cuenta la cultura o sociedad.

Algunos científicos han intentado concienciar tanto a científicos como diseñadores de software sobre el uso de las escalas de color y su accesibilidad, consiguiendo que se hayan creado de manera matemática escalas como *viridis* o *cividis*. También es aconsejado usar escalas grises.

Sin embargo, el artículo destaca que lo más importante es resaltar los datos más relevantes de las visualizaciones, para mostrar de una vez al lector lo importante. Esto es un problema no solo de científicos, sino también de la escasa formación a futuros científicos, las editoriales y herramientas de software.

## 2. Diagrama, herramientas y tecnologías usadas

- **Python 3:** Núcleo de todo el procesamiento y la visualización.
- **Pandas:** Limpieza, transformación y agregación de datos.
- **Matplotlib / Seaborn / Plotly:** Creación de gráficos estáticos y estéticos.
- **Jupyter Notebook:** Interfaz de usuario con celdas, widgets y gráficos embebidos.
- **Patrón de diseño:** Strategy Pattern, para desacoplar visualización y fuentes de datos.
- **ipywidgets:** Selectores, sliders y botones para configurar gráficos on-the-fly.

### Módulos principales

- **datasource.py:** definición de AbstractDataSource y CSVDataSource
- **cleaner.py:** GenericCleaner para preprocesado (drop, imputación media/moda).
- **visualizer.py:** clase Visualizer que aplica una VisualizationStrategy al DataFrame.
- **strategies.py:** contiene todas las estrategias de trazado (LineChartStrategy, BarChartStrategy, ScatterStrategy, HistogramStrategy, MultiGroupStrategy).

