

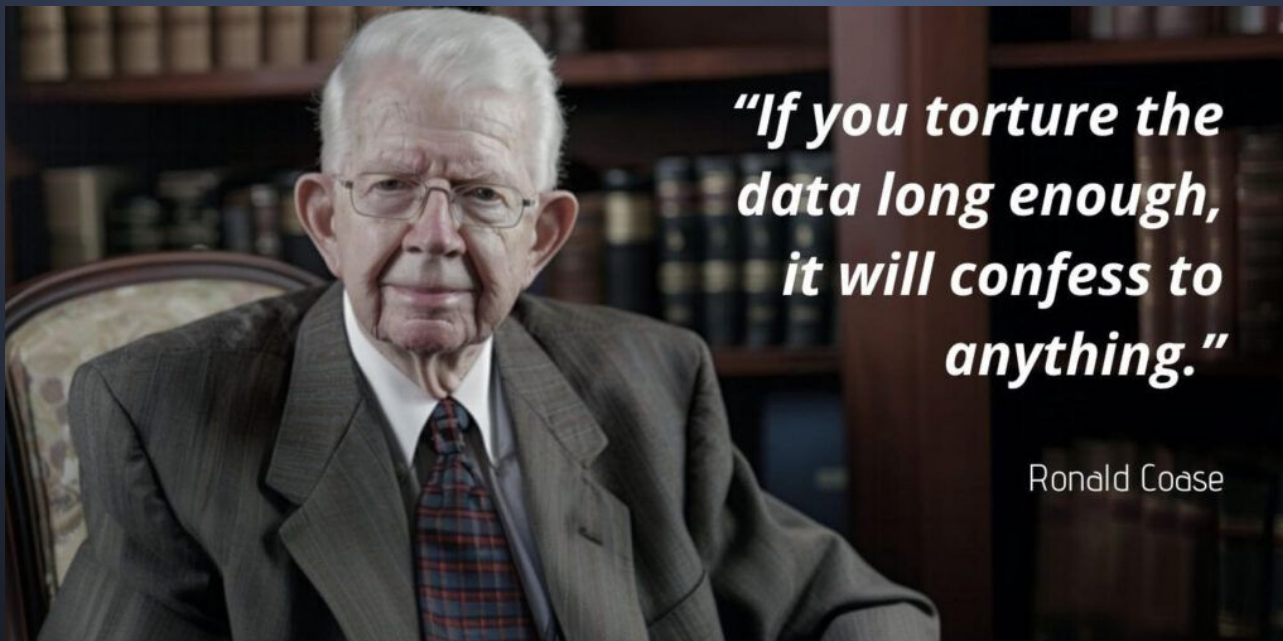


Aprendizaje Automático y Minería de Datos

Minería de datos
Adquisición y exploración

Motivación

- Antes de pasar los datos por la “picadora” hay que investigar meticulosamente lo que nos pueden contar *sin alterarlos*



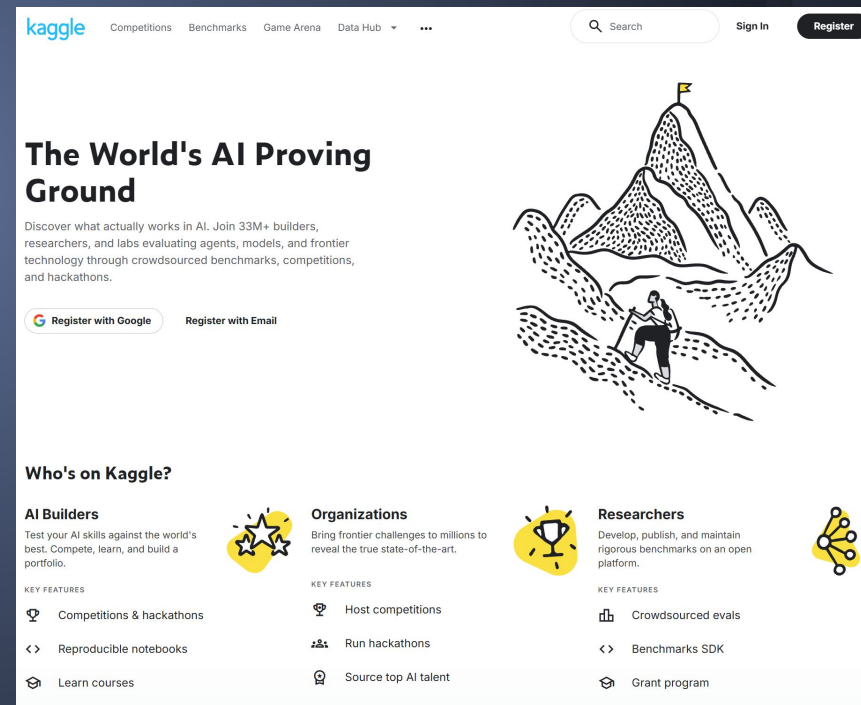
Adquisición

- Cada sistema o formato tiene sus peculiaridades
 - CSV y JSON son los más extendidos
 - XML estuvo de moda, pero es pesado y lento
 - LangExtract y otras herramientas de extracción
- Existen muchos tipos de Bases de Datos
 - Relacionales (tradicionales), centradas en registros individuales y de *procesamiento transaccional en línea*
ONLINE TRANSACTIONAL PROCESSING (OLTP)
 - Analíticas, “almacenes de datos” especializados en MD y de *procesamiento analítico en línea*
ONLINE ANALYTICAL PROCESSING (OLAP)
 - Ej. Google BigQuery o Amazon Redshift
 - No estructuradas (llamadas NoSQL), útiles para grafos o datos heterogéneos
 - Vectoriales, actualmente ideales para IA Generativa

Kaggle

- Repositorio de datos y proyectos relacionados (comprado por Google en 2017)

- Permite almacenar 200 GB por repositorio, más 20 GB de resultados temporales de un cuaderno interactivo que podemos ejecutar 12 h, etc.
- Ofrece competiciones de modelos predictivos (Aprendizaje Supervisado... y hasta algo de No Supervisado), competiciones de simulación (ej. Aprendizaje por Refuerzo... hasta con una Game Arena), hackatones, benchmarks y desafíos de investigación



Ejemplo

- Competición de ejemplo: *¿Quien sobrevivió al hundimiento del Titanic?*
 - Aquí nos dan *definición y ejemplos para cada característica...* en la vida real, nos suele tocar investigarlo y documentarlo a nosotros



Submit Prediction

Titanic - Machine Learning from Disaster

Start here! Predict survival on the Titanic and get familiar with ML basics

Overview

Data

Code

Models

Discussion

Leaderboard

Rules

Dataset Description

Overview

The data has been split into two groups:

- training set (train.csv)
- test set (test.csv)

The **training set** should be used to build your machine learning models. For the training set, we provide the outcome (also known as the "ground truth") for each passenger. Your model will be based on "features" like passengers' gender and class. You can also use **feature engineering** to create new features.

The **test set** should be used to see how well your model performs on unseen data. For the test set, we do not provide the ground truth for each passenger. It is your job to predict these outcomes. For each passenger in the test set, use the model you trained to predict whether or not they survived the sinking of the Titanic.

We also include **gender_submission.csv**, a set of predictions that assume all and only female passengers survive, as an example of what a submission file should look like.

Files

3 files

Size

93.08 kB

Type

csv

License

Subject to Competition Rules

Data Dictionary

Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

Variable Notes

pclass: A proxy for socio-economic status (SES)
1st = Upper
2nd = Middle
3rd = Lower

age: Age is fractional if less than 1. If the age is estimated, is it in the form of xx.5

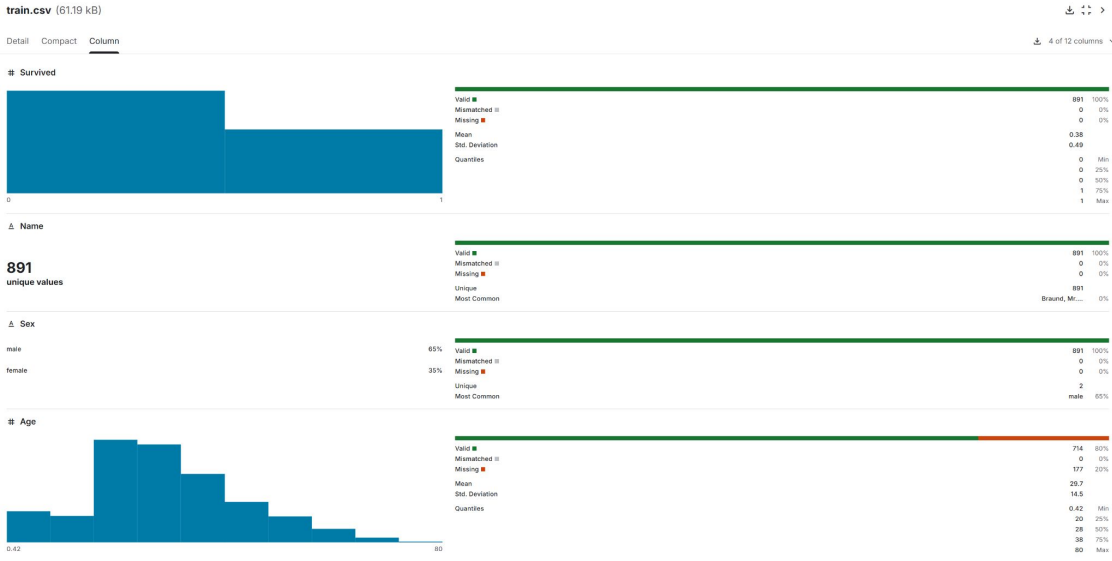
sibsp: The dataset defines family relations in this way...
Sibling = brother, sister, stepbrother, stepsister
Spouse = husband, wife (mistresses and fiancés were ignored)

parch: The dataset defines family relations in this way...
Parent = mother, father
Child = daughter, son, stepdaughter, stepson
Some children travelled only with a nanny, therefore parch=0 for them.

<https://www.kaggle.com/competitions/titanic/>

Ejemplo

- Datos en detalle, en compacto o por columna



train.csv (61.19 kB)

Detail Compact Column

12 of 12 columns

About this file

contains data

PassengerId	# Survived	# Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25	[null]	S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	G6	C
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.925	Other (200)	S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1	C123	S
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.85		S
6	0	3	Moran, Mr. James	male		0	0	330877	8.4583		Q
7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S
8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21.075		S
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0	2	347742	11.1333		S
10	1	2	Nasser, Mrs. Nicholas (Adele)	female	14	1	0	237736	38.0788		C

Selección

- No todos los datos nos interesan y algunos pueden aportar **ruido** al posterior modelado
 - **Relevancia**, la regla de oro es sólo usar datos orientados a nuestro objetivo (explicar, predecir, etc.)
 - **Representatividad**, cuando selecciono muestras / hago **SAMPLING** muestreo (**filas**), y **utilidad** (adecuada granularidad, mínima redundancia, etc.) cuando selecciono características (**columnas**)
 - **Ética y legalidad**, considerando temas de privacidad, anonimización, normas de **GDPR/LOPD**, disponibilidad...

Exploración

EXPLORATORY DATA ANALYSIS (EDA)

- El **análisis exploratorio de datos** consiste en examinar, resumir y visualizar de forma iterativa un conjunto de datos antes de aplicar cualquier modelo estadístico o de AA
 - Maximizar la **comprensión del conjunto de datos**: variables principales, tipo de datos, rango y distribución
 - Detectar **anomalías y errores**, **valores ausentes**, **valores atípicos** e inconsistencias de formato **MISSING VALUES**
 - Descubrir **patrones y relaciones**, **correlaciones**, **agrupamientos preliminares** y tendencias ocultas
 - Validar suposiciones como *normalidad*, *linealidad*... necesarias para aplicar algunos algoritmos después

OUTLIERS

Estadística

- Conviene recordar técnicas y métodos fundamentales de **análisis estadístico**

The screenshot shows the 'Estadística' page on the Narratech website. The header includes the Narratech logo and navigation links: Inicio, Proyectos, Herramientas, Equipo, and English. A search bar is also present. The main title 'Estadística' is prominently displayed. Below it, a 'Curso' tag is visible. The text describes statistics as a formal science for analyzing data and making decisions under uncertainty. It highlights the importance of statistics in data science and machine learning. The page also features sections for 'Análisis exploratorio de datos' and 'Análisis univariante'.

Narratech Laboratorios del Videojuego Narrativo

Inicio Proyectos Herramientas Equipo English

CIENCIA INVESTIGACIÓN UNIVERSITARIO

Estadística

Curso

La Estadística es la ciencia formal (originalmente una rama de las Matemáticas) que se encarga de recolectar, organizar, resumir, analizar e interpretar datos para convertir información numérica en conocimiento y tomar decisiones racionales bajo condiciones de incertidumbre.

Su importancia en Ciencia de Datos es indiscutible. La estadística es el cimiento teórico sobre el que se construye dicha Ciencia y sin ella, los algoritmos de Aprendizaje Automático serían cajas negras incomprensibles. La Estadística proporciona el lenguaje para entender la distribución de las variables, medir la incertidumbre, validar si un patrón descubierto es real o fruto del azar (ej. mediante contrastes de hipótesis) y garantizar que los modelos que construimos generalicen correctamente ante datos nuevos en lugar de limitarse a «alucinar» o memorizar el pasado.

Análisis exploratorio de datos

Como parte de la Minería de Datos es interesante no sólo visualizar y aplicar nuestra intuición sobre los datos que tenemos, sino saber analizarlos con rigor estadístico (estructura de los datos, relaciones entre sus variables, etc.).

Análisis univariante

Antes de analizar interacciones, es imprescindible caracterizar cada variable de forma aislada y evaluar si se cumplen los supuestos que luego permitirán usar (o no) los distintos métodos paramétricos.

<https://narratech.com/es/estadistica/>

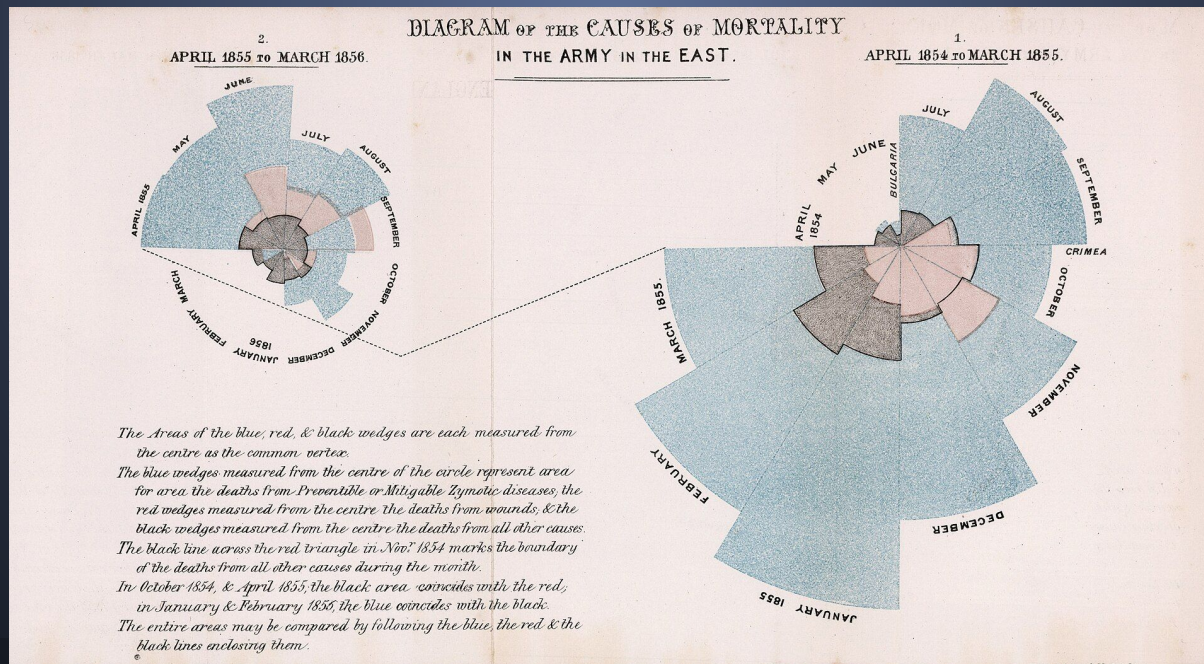
Visualización

- La **representación gráfica de información numérica y categórica mediante elementos visuales** (puntos, barras, curvas, superficies...) ayuda a los seres humanos a revelar patrones, estructuras y anomalías difíciles de ver en una *representación tabular*
- Tiene interés antes y después de tratar los datos para limpiarlos o transformarlos

Ejemplo

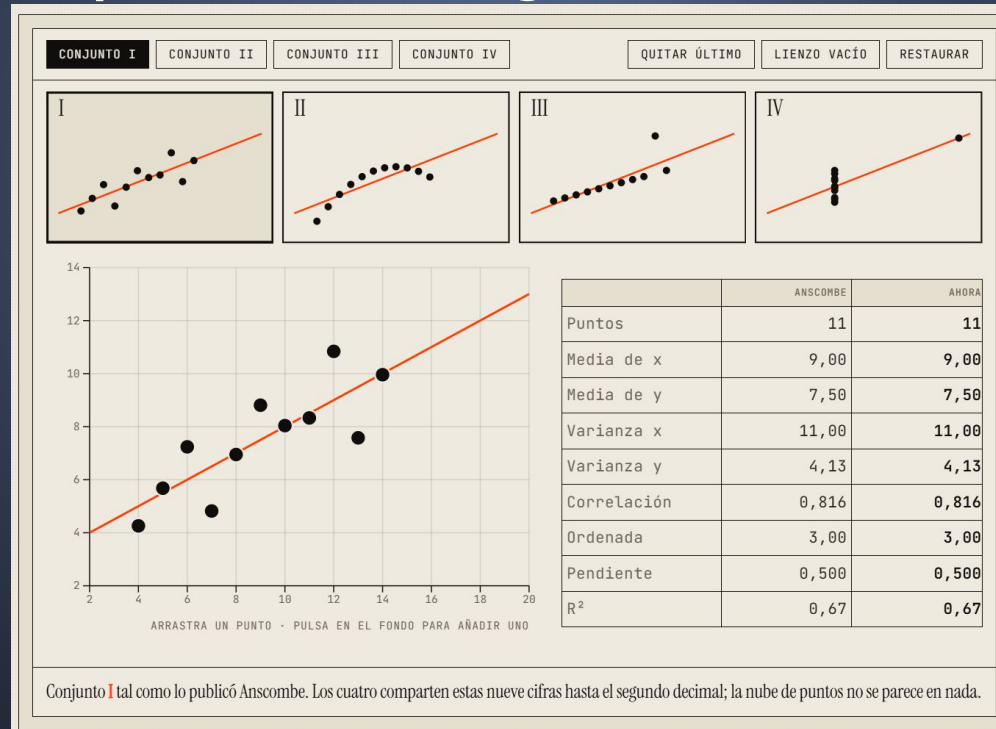
- La enfermera **Florence Nightingale** demostró en **1854** (poniendo “rostro” a la muerte) que los soldados de la Guerra de Crimea morían más por falta de higiene que por sus heridas

¡Las zonas azuladas son muertes que se podían haber evitado!



Ejemplo

- El cuarteto de Anscombe (1973), ¡los números pueden “engañarnos”!



<https://ikusmira.org/p/el-cuarteto-de-anscombe/>

Participación

- ¿Para que NO sirve el **análisis exploratorio**?
 - A. Crear modelos a partir de los datos
 - B. Detectar valores atípicos en los datos
 - C. Descubrir correlaciones entre variables
 - D. Comprender el conjunto de datos, en general
 - E. Validar suposiciones de normalidad en los datos



- Biblioteca fundamental para la manipulación, análisis y estructuración de datos en tablas
 - Construida sobre NumPy por Wes McKinney (2008), actualmente de código abierto
 - Facilidades para importar y exportar datos en formato CSV, JSON, SQL, Excel, Bases de Datos...



Matemático del MIT, speedrunner de Golden Eye 007 y “dictador benevolente vitalicio” de Pandas (¡nacido en plena Crisis Financiera de 2007-2008!)

```
import numpy as np
import pandas as pd

# 1. Definir los datos en un diccionario
datos = {
    "Título": ["The Witcher 3", "Elden Ring", "Hades", "Celeste", "Cyberpunk 2077"],
    "Género": ["RPG", "RPG", "Roguelike", "Plataformas", "RPG"],
    "Puntuación": [92, 96, 93, np.nan, 86], # Incluye un valor nulo (np.nan)
    "Ventas_M": [50.0, 25.0, 5.0, 1.5, 25.0], # Ventas en millones
    "Indie": [False, False, True, True, False],
}

# 2. Crear el DataFrame
df = pd.DataFrame(datos)

# 3. Mostrar el DataFrame por pantalla
print("--- DATAFRAME COMPLETO ---")
print(df)

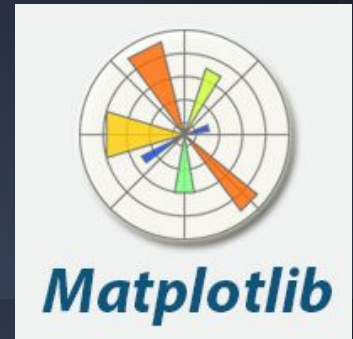
# 4. Inspección rápida (EDA básico)
print("\n--- INFORMACIÓN DE TIPOS Y NULOS ---")
print(df.info())

print("\n--- ESTADÍSTICAS DESCRIPTIVAS ---")
print(df.describe())
```

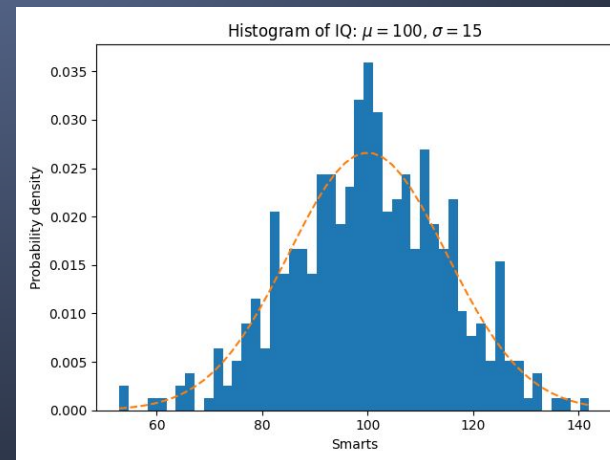
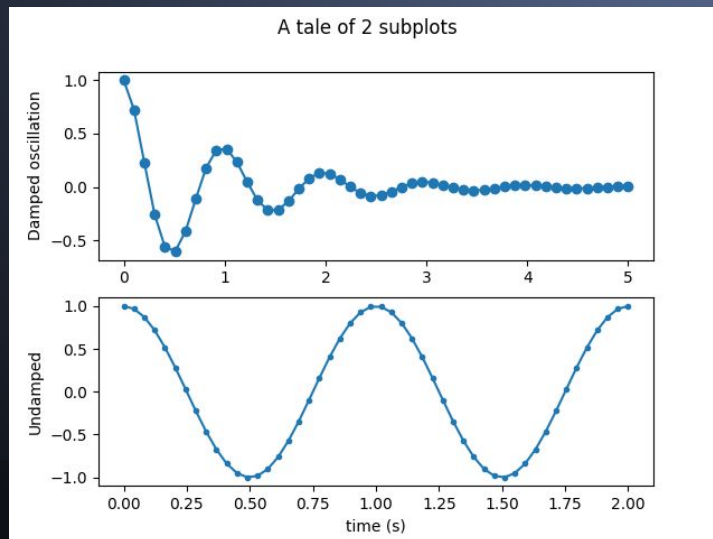
Serie y DataFrame

- **Serie**, array unidimensional etiquetado
 - Ej. Una **columna** con todos los valores de una misma característica para muchas muestras
- **DataFrame**, array bidimensional etiquetado en los dos ejes
 - Ej. Típica tabla con **filas y columnas**
 - Inspección rápida mediante **head**, **tail**, **info**, **describe...**
 - Indexación por etiquetas (**loc**) o posición (**iloc**)
 - Representación y tratamiento de **valores nulos** con **isna**, **fillna**, **dropna...** (tipo extensible **pd.NA**, que mejora el **np.nan** y el **None** de Python)
 - Operaciones complejas tipo SQL como **groupby** y **pivot_table**

Matplotlib



- Biblioteca / **Motor gráfico a bajo nivel** para Python de código libre, inspirado en **MATLAB**
 - **Figure** es el lienzo
 - **Axes** es el área de trazados con eje X, Y, leyendas...
 - Usarla “a pelo” (orientada a objetos) es duro, siendo mejor orientado a estados (módulo **pyplot**)



Ejemplo

```
from matplotlib.figure import Figure
from matplotlib.backends.backend_agg import FigureCanvasAgg

# 1. Definir los datos
x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]

# 2. Instanciar la Figura y asignarle un Canvas (Backend)
fig = Figure()
canvas = FigureCanvasAgg(fig)

# 3. Añadir los ejes a la figura
ax = fig.add_subplot(111)

# 4. Dibujar y personalizar sobre 'ax'
ax.plot(x, y, color='blue', marker='o', linestyle='--')
ax.set_title('Gráfico de Línea Sencillo')
ax.set_xlabel('Eje X')
ax.set_ylabel('Eje Y')

# 5. Guardar la imagen directamente (sin ventana emergente)
fig.savefig('grafico_oo.png')
```



```
import matplotlib.pyplot as plt

# 1. Datos
x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]

# 2. Inicialización limpia de Figura y Ejes
fig, ax = plt.subplots()

# 3. Trazo de datos
ax.plot(x, y, color='blue', marker='o', linestyle='--')

# 4. Configuración agrupada de metadatos
ax.set(
    title='Gráfico de Línea Sencillo',
    xlabel='Eje X',
    ylabel='Eje Y'
)

# 5. Renderizado
plt.show()
```

¡Mejor!

Visualización avanzada

- **Seaborn** se construye sobre **Matplotlib** y se integra con **Pandas** para dibujar distribuciones, mapas de calor de correlación, diagramas de caja (*box plots*), gráficos de dispersión (*pair plots*)... de manera **más sencilla y estética**
- **Plotly** crea **gráficas interactivas** (en **HTML** y **JavaScript**), teniendo **Plotly Express** como capa de alto nivel para facilitar su uso

Ejemplos

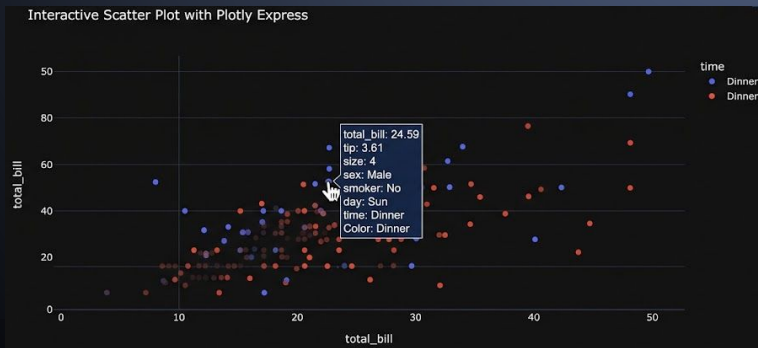
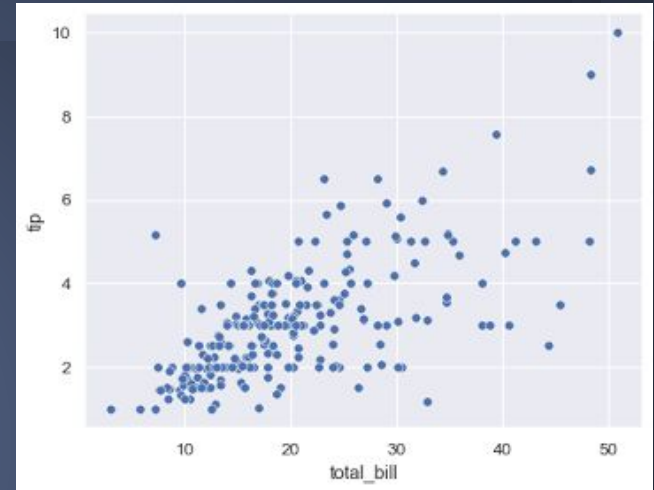
```
import seaborn as sns
import matplotlib.pyplot as plt

# 1. Cargar dataset de ejemplo de Seaborn
df = sns.load_dataset("tips")

# 2. Crear gráfico de dispersión (Scatter Plot)
sns.scatterplot(data=df, x="total_bill", y="tip", hue="time")

# 3. Personalizar título
plt.title("Relación entre Cuenta y Propina")

# 4. Mostrar gráfico
plt.show()
```



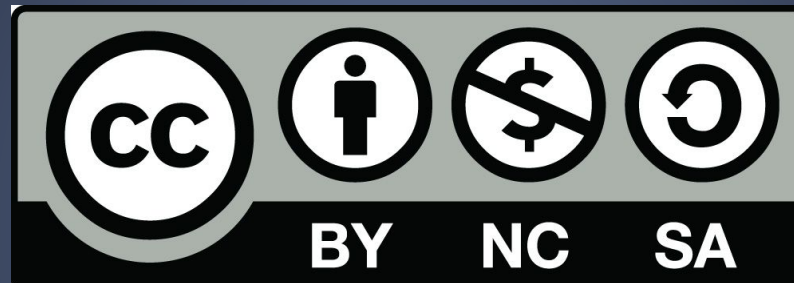
```
import plotly.express as px

# 1. Cargar dataset de ejemplo de Plotly
df = px.data.tips()

# 2. Crear gráfico interactivo de dispersión (retorna el objeto 'fig')
fig = px.scatter(
    df,
    x="total_bill",
    y="tip",
    color="time",
    title="Relación entre Cuenta y Propina"
)

# 3. Mostrar gráfico interactivo
fig.show()
```

Críticas, dudas, sugerencias...



* Excepto el contenido multimedia de terceros autores

Federico Peinado (2026)

www.federicopeinado.es

