

## EJERCICIOS DE MATPLOTLIB

Cargue el fichero `bmw.csv` y prepare `numpy` y `matplotlib` con el siguiente código:

```
import matplotlib as plt
import pandas as pd
import numpy as np

%matplotlib inline

# Conectar a Google Drive
from google.colab import drive
drive.mount('/content/drive')

df = pd.read_csv('/content/drive/MyDrive/my_data/bmw.csv')
```

### Ejercicio 1:

Representa la función  $f(x) = \sin(x) + \cos(x)$  en el intervalo  $[0, 2\pi]$ .

### Ejercicio 2:

Representa las funciones  $f(x) = \sin(x)$ ,  $g(x) = \log(1+x)$  en el intervalo  $[0, 2\pi]$ , tomando 100 puntos igualmente espaciados en dicho intervalo.

### Ejercicio 3:

Crea un gráfico de barras que muestre las frecuencias de los distintos valores del atributo `model` de la base de datos.

### Ejercicio 4:

Crea un gráfico de dispersión que muestre los precios (atributo `price`) en el eje horizontal, los kilometrajes (atributo `mileage`) en el eje vertical, y las eficiencias (atributo `mpg`) como colores.

### Ejercicio 5:

Crea un histograma que muestre los precios (atributo "price") de la base de datos.

### Ejercicio 6:

Considera la siguiente función, que calcula valores enteros entre 0 y 255 para dibujar el fractal de Mandelbrot, siendo a y b números en coma flotante que representan la parte real y la parte imaginaria de un número complejo, respectivamente:

```
MAXIMO_ITERACIONES = 80
```

```
def mandelbrot(a,b):
```

```
    c = complex(a,b)
```

```
    z = 0
```

```
    n = 0
```

```
    while abs(z) <= 2 and n < MAXIMO_ITERACIONES:
```

```
        z = z*z + c
```

```
        n += 1
```

```
    color_pixel = 255 - int(n * 255 / MAXIMO_ITERACIONES)
```

```
    return color_pixel
```

Utilizando el código anterior, dibuja un mapa de calor que represente una parte del fractal de Mandelbrot. El mapa debe tener 600 píxeles de ancho por 400 píxeles de alto, de manera que se consideren valores de la parte real entre -2 y 1, mientras que los valores de la parte imaginaria deben estar entre -1 y 1.