

Visualización con Matplotlib: Algunos elementos de apoyo

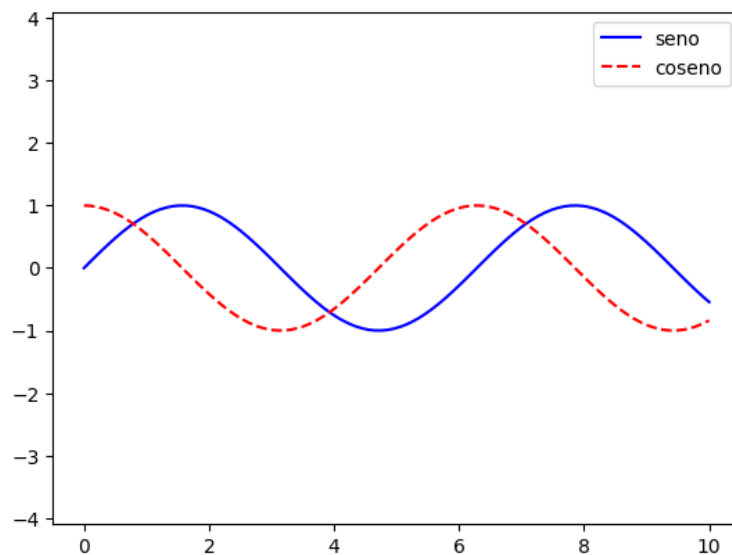
En esta sesión vamos a trabajar con las leyendas, los textos añadidos y en cómo marcar con flechas y señalar algunas zonas importantes de nuestras figuras y gráficas.

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

Manipulando leyendas

Recordemos como se creaba una leyenda, básicamente añadíamos una etiqueta a las gráficas y luego llamábamos a la función `plt.legend()`:

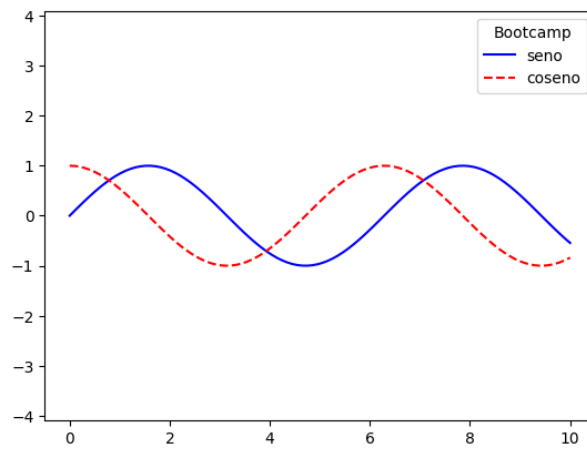
```
x=np.linspace(0,10,100)
fig,ax=plt.subplots()
ax.plot(x,np.sin(x),"-b",label="seno")
ax.plot(x,np.cos(x),"--r",label="coseno")
ax.axis("equal")
ax.legend();
```



Ahora la manipulamos:

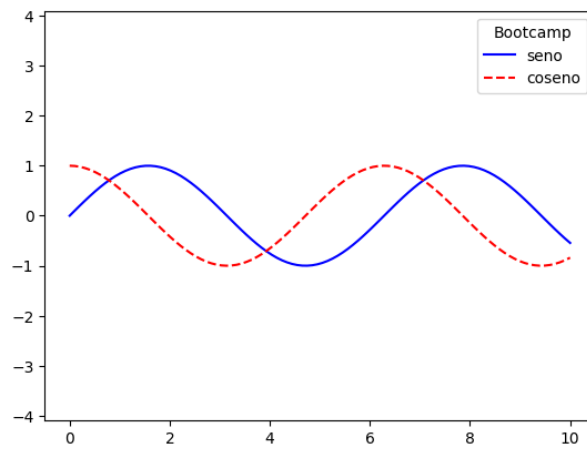
- Ponerle título, con el argumento `title`

```
ax.legend(loc="upper right",title="Bootcamp");
fig
```

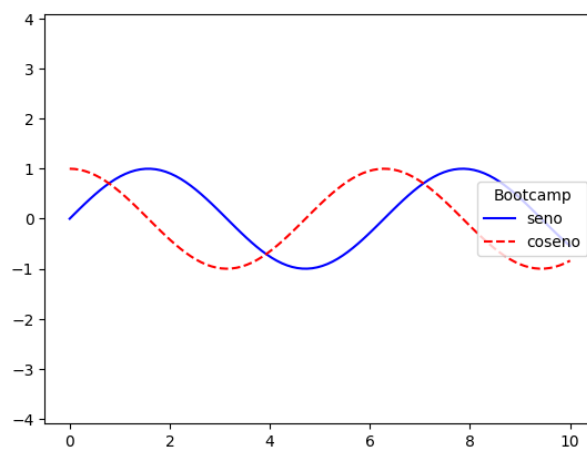


- Cambiar la posición de la leyenda, con el argumento loc

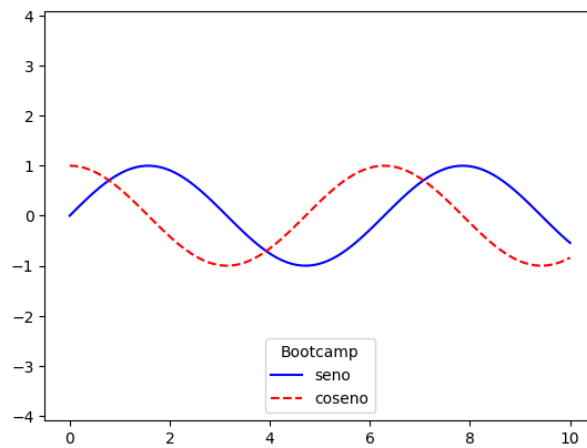
```
ax.legend(loc="upper right",title="Bootcamp");
fig
```



```
ax.legend(loc="center right", title="Bootcamp")
fig
```

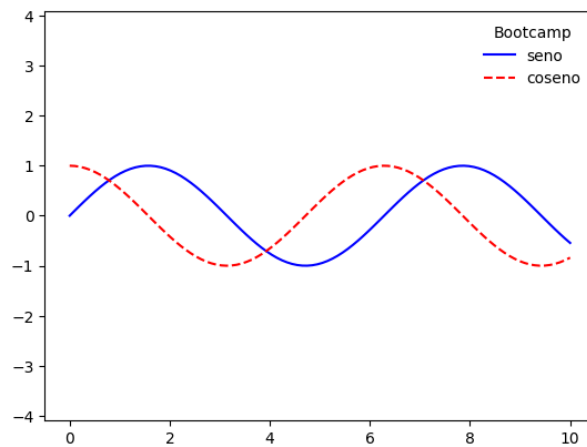


```
ax.legend(loc="lower center",title="Bootcamp")
fig
```



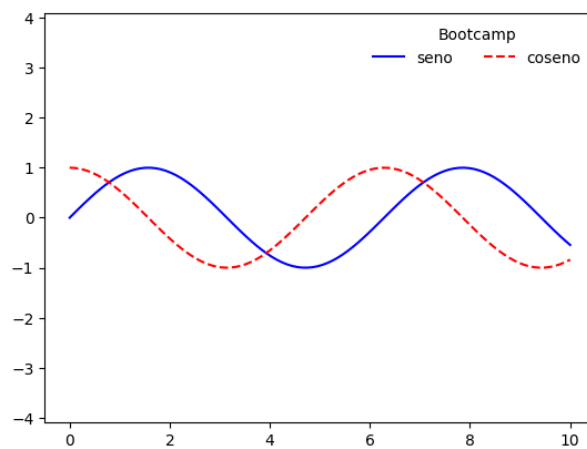
- Quitar el marco y ponerlo con el argumento frameon

```
ax.legend(loc="upper right",title="Bootcamp",frameon=False)
fig
```



- Indicar el número de columnas de la leyenda con ncol

```
ax.legend(loc="upper right",title="Bootcamp",frameon=False, ncol=2)
fig
```



Truquitos (I): Leyendas para el tamaño de los puntos

A veces, los valores predeterminados de la leyenda no son suficientes para la visualización dada. Por ejemplo, tal vez estés utilizando el tamaño de los puntos para marcar ciertas características de los datos, y quieras crear una leyenda que refleje esto. Además, también aprenderemos algo sobre la barra de colores o color bar.

Aquí tienes un ejemplo donde usaremos el tamaño de los puntos para indicar las poblaciones de las ciudades de California. Nos gustaría una leyenda que especifique la escala de los tamaños de los puntos, y lo lograremos trazando algunos datos etiquetados sin entradas:

```
cities=pd.read_csv("./data/california_cities.csv")
cities
```

```
[16]:
```

	Unnamed: 0	city	latd	longd	elevation_m	elevation_ft	population_total	area_total_sq_mi	area_land_sq_mi	area_water_sq_mi	area_total_km2	area_land_
0	0	Adelanto	34.576111	-117.432778	875.0	2871.0	31765	56.027	56.009	0.018	145.107	145
1	1	AgouraHills	34.153333	-118.761667	281.0	922.0	20330	7.822	7.793	0.029	20.260	20
2	2	Alameda	37.756111	-122.274444	NaN	33.0	75467	22.960	10.611	12.349	59.465	27
3	3	Albany	37.886944	-122.297778	NaN	43.0	18969	5.465	1.788	3.677	14.155	4
4	4	Alhambra	34.081944	-118.135000	150.0	492.0	83089	7.632	7.631	0.001	19.766	19
...
477	477	Yountville	38.403056	-122.362222	30.0	98.0	2933	1.531	1.531	0.000	3.966	3
478	478	Yreka	41.726667	-122.637500	787.0	2582.0	7765	10.053	9.980	0.073	26.036	26
479	479	YubaCity	39.134722	-121.626111	18.0	59.0	64925	14.656	14.578	0.078	37.959	37
480	480	Yucaipa	34.030278	-117.048611	798.0	2618.0	51367	27.893	27.888	0.005	72.244	72
481	481	YuccaValley	34.133333	-116.416667	1027.0	3369.0	20700	40.015	40.015	0.000	103.639	103

482 rows x 14 columns

Practiquemos además un poco con todo lo aprendido en las sesiones anteriores:

```
# Dispersion, barra de colores
```

```
lat,lon=cities["latd"],cities["longd"]
```

```
population,area=cities["population_total"],cities["area_total_km2"]
```

```
plt.figure(figsize=(20,10))
```

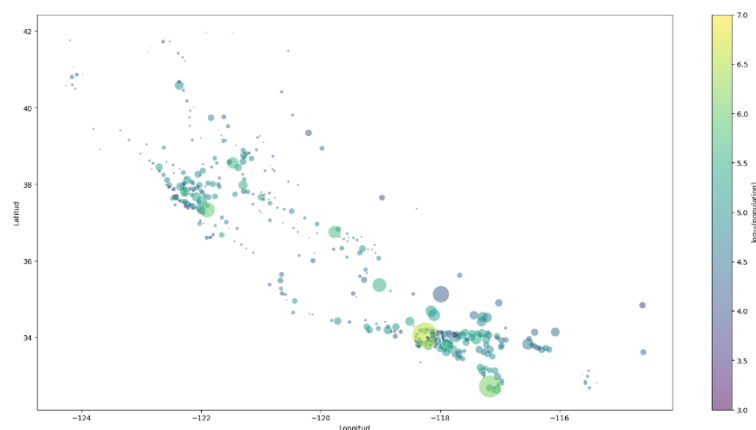
```
plt.scatter(lon,lat,c=np.log10(population),cmap="viridis", s=area, linewidth=0, alpha=0.5)
```

```
plt.xlabel("Longitud")
```

```
plt.ylabel("Latitud")
```

```
plt.colorbar(label="log$_{10}$ (population)")
```

```
plt.clim(3,7)
```

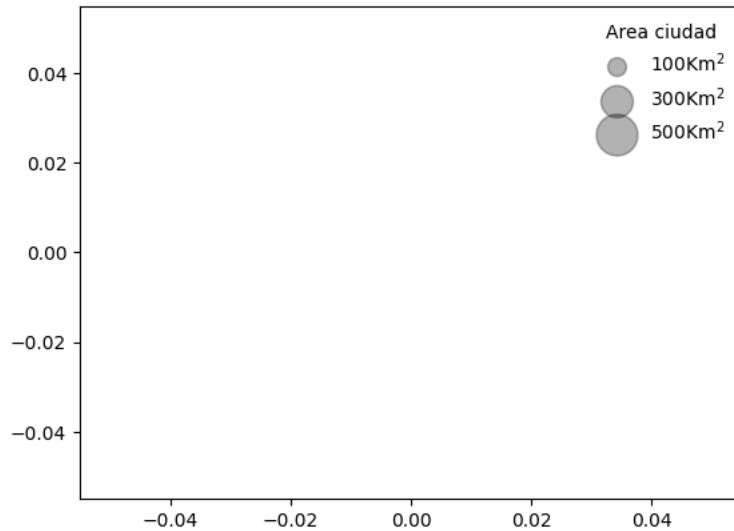


Fake legend

for area in [100,300,500]:

plt.scatter([],[],c="k",alpha=0.3,s=area,label=str(area)+"Km²")

plt.legend(frameon=**False**,title="Area ciudad");



Todo junto

lat,lon=cities["latd"],cities["longd"]

population,area=cities["population_total"],cities["area_total_km2"]

plt.figure(figsize=(20,10))

plt.scatter(lon,lat,c=np.log10(population),cmap="viridis", s=area, linewidth=0, alpha=0.5)

plt.xlabel("Longitud")

plt.ylabel("Latitud")

plt.colorbar(label="log₁₀(population)")

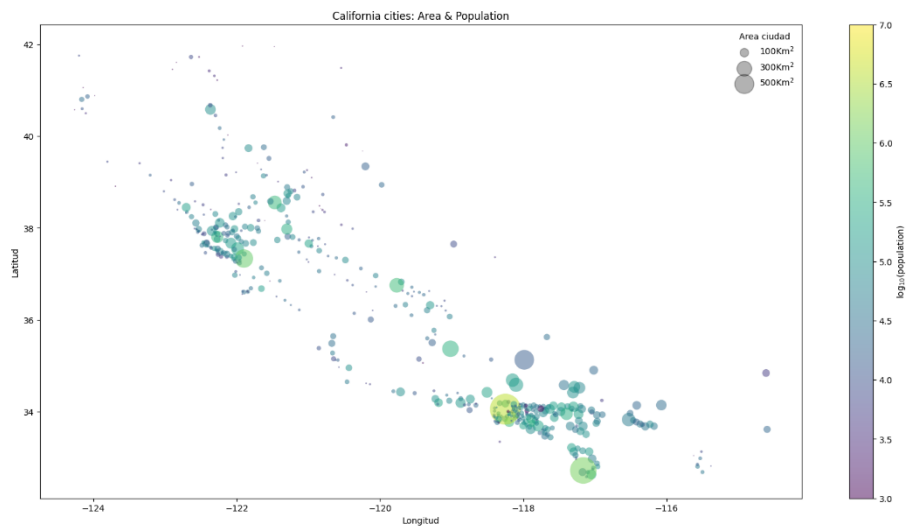
plt.clim(3,7)

for area in [100,300,500]:

plt.scatter([],[],c="k",alpha=0.3,s=area,label=str(area)+"Km²")

plt.legend(frameon=**False**,title="Area ciudad");

plt.title("California cities: Area & Population");

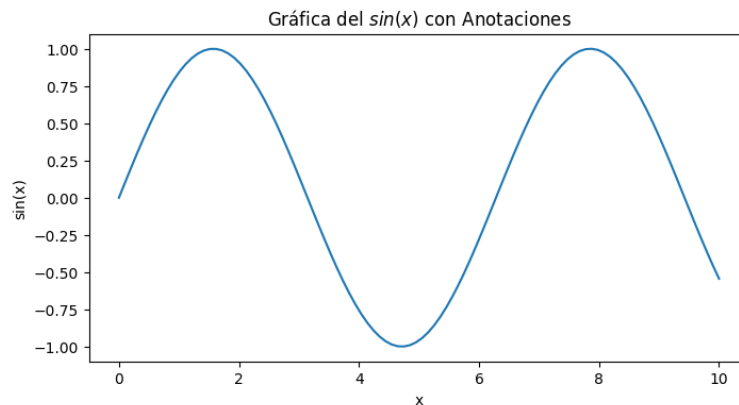


Textos, anotaciones y flechas

Texto

Ya lo vimos, de pasada, al hablar de subplots y marcar estos. Es sencillo:

```
# Generamos una gráfica simple
x = np.linspace(0, 10, 100)
y = np.sin(x)
min_x_value = pd.Series(y, index= x).idxmin()
min_y_value = pd.Series(y, index= x).min()
plt.figure(figsize=(8, 4))
plt.plot(x, y)
plt.title(r"Gráfica del  $\sin(x)$  con Anotaciones")
plt.xlabel("x")
plt.ylabel("sin(x)");
```



```
# Añadir texto en la gráfica
x = np.linspace(0, 10, 100)
y = np.sin(x)
min_x_value = pd.Series(y, index= x).idxmin()
min_y_value = pd.Series(y, index= x).min()
plt.figure(figsize=(8, 4))
plt.plot(x, y)
plt.title(r"Gráfica del  $\sin(x)$  con Anotaciones")
plt.xlabel("x")
plt.ylabel("sin(x)")

font_dict={
    "size":16,
    "family": "Calibri", No sé porqué no me la coge bien
    "color":"darkblue"
}
plt.text(min_x_value,min_y_value,"Valor mínimo", fontdict=font_dict);
```

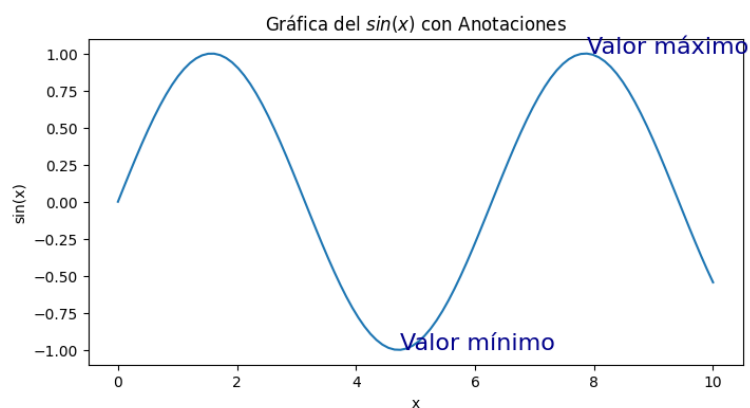


Añadir más texto en la gráfica

```
x = np.linspace(0, 10, 100)
y = np.sin(x)
min_x_value = pd.Series(y, index= x).idxmin()
min_y_value = pd.Series(y, index= x).min()
max_x_value = pd.Series(y, index= x).idxmax()
max_y_value = pd.Series(y, index= x).max()
plt.figure(figsize=(8, 4))
plt.plot(x, y)
plt.title(r"Gráfica del  $\sin(x)$  con Anotaciones")
plt.xlabel("x")
plt.ylabel("sin(x)")

font_dict={
    "size":16,
    "family": "Calibri",
    "color":"darkblue"
}

plt.text(min_x_value,min_y_value,"Valor mínimo", fontdict=font_dict)
plt.text(max_x_value,max_y_value,"Valor máximo", fontdict=font_dict);
```



[Aquí](#) puedes encontrar una lista de fuentes disponibles en Matplotlib (aunque puede que no te funcionen todas).

Y ahora veamos otra forma de añadir texto con flechas de diversa forma para poder marcar y anotar nuestras gráficas.

Flechas y anotaciones

Repetir la gráfica simple

```
x = np.linspace(0, 10, 100)
```

```
y = np.sin(x)
```

```
min_x_value = pd.Series(y, index=x).idxmin()
```

```
min_y_value = pd.Series(y, index=x).min()
```

```
plt.figure(figsize=(8, 4))
```

```
plt.plot(x, y)
```

```
plt.title(r"Gráfica del  $\sin(x)$  con Anotaciones")
```

```
plt.xlabel("x")
```

```
plt.ylabel("sin(x)")
```

#Añadir la flecha indicativa

```
props_flecha={
```

```
    "arrowstyle":"<->",
```

```
    "connectionstyle":"angle3"
```

```
}
```

```
plt.annotate("Valor
```

```
mínimo",xy=(min_x_value,min_y_value),xytext=(min_x_value,0.5),arrowprops=props_flecha)
```

