**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();

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Ahora la manipulamos:

* Ponerle título, con el argumento title

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

fig

Gráfico, Gráfico de líneas

Descripción generada automáticamente

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

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

fig

Gráfico, Gráfico de líneas

Descripción generada automáticamente

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

fig

Gráfico, Gráfico de líneas

Descripción generada automáticamente

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

fig

Gráfico, Gráfico de líneas

Descripción generada automáticamente

* Quitar el marco y ponerlo con el argumento frameon

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

fig

Gráfico, Gráfico de líneas

Descripción generada automáticamente

* Indicar el número de columnas de la leyenda con ncol

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

fig

Gráfico, Gráfico de líneas

Descripción generada automáticamente

**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

Pantalla de computadora

Descripción generada automáticamente con confianza media

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)

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

*# Fake legend*

**for** area **in** [100,300,500]:

plt.scatter([],[],c**=**"k",alpha**=**0.3,s**=**area,label**=**str(area)**+**"Km$^2$")

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

Gráfico

Descripción generada automáticamente

*# 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**=**"log$\_{10}$(population)")

plt.clim(3,7)

**for** area **in** [100,300,500]:

plt.scatter([],[],c**=**"k",alpha**=**0.3,s**=**area,label**=**str(area)**+**"Km$^2$")

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

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

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

**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)");

Gráfico, Gráfico de líneas

Descripción generada automáticamente

*# 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);

Gráfico

Descripción generada automáticamente

*# 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);

Gráfico, Gráfico de líneas

Descripción generada automáticamente

[Aquí](https://jonathansoma.com/lede/data-studio/matplotlib/list-all-fonts-available-in-matplotlib-plus-samples/) puedes encontrar una lista de fuentes disponibles en Matplotlib (aunque puede que no te fucionen 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)

Gráfico

Descripción generada automáticamente