

# Visualización de Datos: Matplotlib y Seaborn

sábado, 9 de julio de 2022 21:41

## MatPlotLib

"Visualization gives you answers to questions you didn't know you had."  
Ben Schneiderman

Las distribuciones de los datos pueden ser distintas, sin embargo poseer misma media, std.

Matplotlib emula a MatLab

`plt.plot(x,y, "colorpunterolinea") -> plt.plot(x,y, "bo-")` #grafica azul puntero de círculos y línea

## Subplot

`plt.subplot(filas, columnas, indice)`

`plt.plot(x,y)`

Indice: Para `subplot(1,2, 1)` se me crean 2 índices, en este caso voy a trabajar con el 1ero

`plt.plot(x,y)`

`subplot(1,2, 2)`

`plt.plot(x,y)`

## Metodo orientado a objeto

Se usa cuando se requiere un gran nivel de personalización de las figuras de una visualización.

Figure es el lienzo, en donde van a estar situadas todas las figuras.

Axes, es una figura en particular.

Cada axes es un objeto distinto que puede o no reposar en el mismo lienzo.

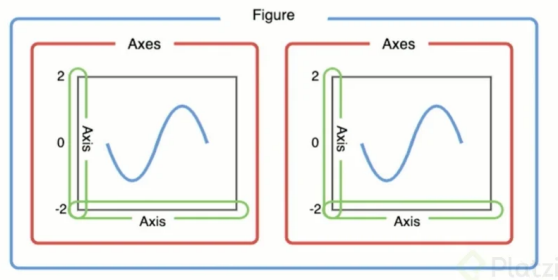
Axis, son los ejes coordenados de una o más figuras en particular.

`fig = plt.figure()`

`axes = fig.add_axes([posicionHorizontalEnLienzo,  
posicionVerticalEnLienzo,  
tamañoHorizontal,  
tamañoVertical])`

`axes = fig.add_axes([0.1,0.1,0.8,0.9])`

## Estructura



## Subplots

Trabaja con el metodo orientado a objetos

`fig, axes = plt.subplots(numeroFilas, numeroColumnas)`

`fig, axes = plt.subplots(1, 2) #1 Fila, 2 Columnas`

`axes[0].plot(x,y) #grafico algo en la fila 1 columna 1`

`axes[1].plot(x,y) #grafico algo en la fila 1 columna 2`

`fig, axes = plt.subplots(2,4) #2 filas, 4 Columnas`

`#axes[filas,columnas].plot(x,y)`

`axes[0,0].plot(x,y) #grafico algo en fila 1, columna 1`

##Leyendas, etiquetas, títulos, tamaños

`axes[filas, col].set_title("") #título de la figura [fil, col]`

`axes[filas, col].set_xlabel("") #título del eje X de la figura [fil, col]`

```

Axes[fil, col].set_ylabel("") #título del eje Y de la figura [fil, col]

#Tamaño del lienzo por medio de figsize
fig, (ax1,ax2) = plt.subplots(1,2, figsize= (tamañoHorizontal,tamañoVertical))

#Etiquetas
ax1.plot(x,y, 'b', label = "sin(x)")
Ax1.legend() #mostrar el label en la grafica

##Colores y estilos
Plt.style.available #estilos disponibles para uso
Plt.style.use("nombreEstilo") #usar un estilo de los disponibles

ax.plot(x,x+3,color = 'blue', linestyle = 'dashed', marker = '8', markersize = 10, markerfacecolor=
"#37D842") #cambiar tipo, tamaño y color del marker

##Grafico de barras
plt.bar(country, population, width=0.5, color= ["aqua", "grey", "teal", "crimson", "violet"]) #width =
grosor de las barras
plt.xticks(np.arange(5), ('India','Japon', 'Mexico', 'Colombia', 'Alemania'), rotation = 45) #xticks para
cambiar config del eje x de la fig, rotation para rotar lo nombre x angulo

plt.barh() # para grafico de barras horizontal

##Otros graficos

Plt.hist() # histograma: distribución de los datos del dataset
Plt.hist(beans = numeroDeBeans) #los beans son cuantas barras van a haber en el histograma
Plt.hist(histtype = step)#tipo de histograma a ver, se puede barras (bar), o step (pasos)

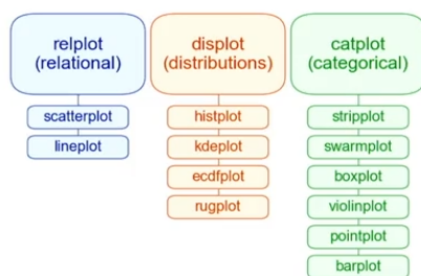
Plt.scatter(x,y, s=tamañoDeLosPuntos, marker = 'o', alpha = 0.5 ) #la S es el tamaño de cada punto,
internamente el programa lee 1 a 1 los valores y le asigna un tamaño al punto respectivo. Alpha es la
transparencia, y marker el marcador o forma de los puntos.

```

## Seaborn

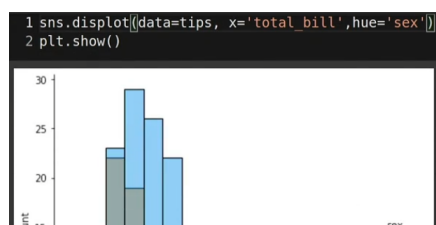
La base de seaborn es matplotlib, optimizada para funcionar con dataframes: pandas

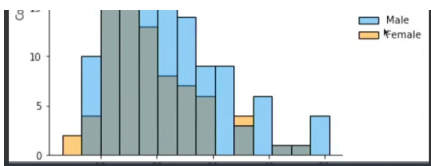
### Tipos de gráficas



Sns.set\_theme() #para usar un estilo predeterminado de seaborn

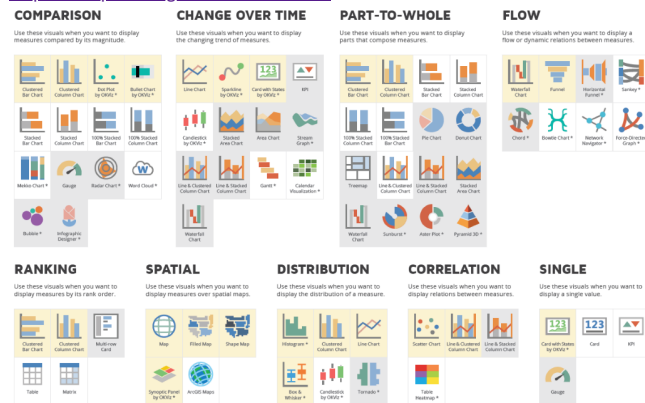
Sns.displot() #histograma  
 Sns.displot(hue = "variableARelacionar") #relacion entre variables 'x' y 'y' agrupados por hue  
 Sns.displot(kind = 'kde', palette = "paletaDeColor") #kind es el tipo de grafica que quiero, kde es para graficar por densidad





## Cheat Sheet

<https://deeplearning.fr/seaborn-cheats/>



### ##Distribuciones en Seaborn

`Sns.histplot(data = datosAGraficar, x,y=ejeX - ejeY,bins = numeroDeBarras, cumulative = True, hue, stat = 'probability', multiple = 'stack')` #histograma, cumulative para ver comportamiento acumulado de las frecuencias, hue para agrupar  
Stat para trabajar con conteo, probabilidad, porcentajes, densidad y demas.  
Multiple para configurar la presentacion de los datos: dodge, stack, fill

`Sns.kdeplot(data = datosAGraficar, x o y=ejeX o ejeY, shade = True, bw_adjust = 1)` #diagrama de densidad

Shade hace un área bajo la curva

`bw_adjust` ajusta la gráfica a los parámetros de los ejes

`Sns.ecdfplot()` #grafico escalonado

`Sns.distplot(kind = 'kde')` # Seaborn grafica el grafico que se adapte mejor al dataset con kind podemos cambiar el tipo de grafica

### ## Graficado de variables categóricas o de texto

`Sns.countplot()` #conteo de variables categoricas

`Sns.stripplot(dodge = true)` #grafico de dispersion, separar las distintas dispersiones

`Sns.swarmplot(dodge = true)` #grafico de dispersión, separar las distintas dispersiones, se tiene un eje horizontal donde hay más concentración de datos

`Sns.boxplot()` #cajas y bigotes

`Sns.violinplot(split = True)` #donde se concentran los datos, donde hay más datos crece horizontalmente, split combina los 2 violines en 1 solo

`Sns.catplot(kind = 'box', col = "Time")` #Seaborn define el mejor grafico para variables categoricas  
Kind define el tipo de gráfica, de cajas y bigotes o demás, col sirve para graficar / segmentar por otra dimensión.

### ## Graficado de relación entre variables

Markers = {"Lunch": "d", "Dinner": "s"} #diccionario para marcadores de la figura

`Sns.scatterplot(style = 'time', size = 'size', markers = markers)` #grafico de dispersion, style representa el marker en este caso van a haber 2 markers ya que en la columna time hay 2 valores: desayuno (x) y comida (\*), size es el tamaño de los markers, el marker tendra el tamaño según la columna size se lo indique

`Sns.lineplot()` #linea de relacion, mismos atributos de scatterplot

`Sns.relplot(col = 'time')` #Seaborn define el mejor gráfico, col sirve para graficar / segmentar por otra dimensión.

### ##Join Plot y Pair Plot

Para combinar más de un tipo de grafica

`sns.jointplot(data=tips, x='total_bill', y='tip', hue='sex', marginal_ticks=True)` #Seaborn grafica la mejor manera de representar la data, `marginal_ticks` es para ver los ejes de las gráficas de los lados  
Al acceder al marginal estoy accediendo a las gráficas de los lados

`Sns.pairplot(data=tips)` #graficar relación entre todas las variables numéricas del dataset

### ##HeatMap

`Tips.corr()` #correlacion de pearson entre variables

`Sns.heatmap(Tips.corr(), annot=True, linewidth = 5, vmin=ValorMinimoDeEscala, vmax=ValorMaximoDeEscala, cbar = True)` #mapa de calor, Para ver los valores de correlación, `linewidth` para separar cajas de matriz, `cbar False` para quitar la barra de la derecha