

Matplotlib y Seaborn

Matplotlib

Gráficas

```
import matplotlib.pyplot as plt

plt.rcParams["figure.figsize"] = (10,8)
plt.figure(figsize = (n,m))
```

inicia una grafica dibujando el marco de la figura; n es la anchura y m es la altura, en pulgadas

```
plt.show()
```

 muestra la figura

Gráficas básicas

Bar plot

```
plt.bar(df["columna1"], df["columna2"])
```

 crea un diagrama de barras donde los ejes son: columna1 – x, columna2 – y

Horizontal bar plot

```
plt.barh(df["columna1"], df["columna2"])
```

 crea una diagramma de barras horizontales donde los ejes son: columna1 – x, columna2 – y

Stacked bar plot

```
plt.bar(x, y, label = 'etiqueta')
plt.bar(x2, y2, bottom = y, label = 'etiqueta2')
```

 crea una diagrama de barras apiladas para visualizar dos variables juntas; y indica la barra de referencia

Scatter plot

```
plt.scatter(df["columna1"], df["columna2"])
```

 crea una gráfica de dispersión donde los ejes son: columna1 – x, columna2 – y

Gráficas estadísticas

Histogram

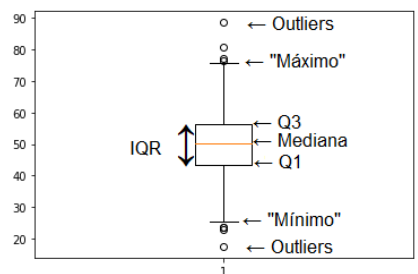
```
plt.hist(x = df['columna1'], bins = n)
```

 crea una histograma que muestra la frecuencias de una distribución de datos; donde x es la variable de interés y n es el número de barras

Box Plot

```
plt.boxplot(x = df['columna1'])
```

 crea un diagrama de cajas para estudiar las características de una variable numerica; x es la variable de interés - el mínimo es lo mismo que Q1 - 1.5 * IQR - el máximo es lo mismo que Q3 + 1.5 * IQR



Pie Chart

```
plt.pie(x, labels = categorias, radius = n)
```

 crea un gráfico de sectores donde x es la variable de interés (debe esta agrupado por categorias); n es el tamaño

Violin Plot

```
plt.violinplot(x, showmedians = True, showmeans = True)
```

 crea un diagrama de violin donde x es la variable de interés y muestra la mediana y la media

Seaborn gráficas

Line plot

```
fig = sns.lineplot(x = 'columna1', y = 'columna2', data = df, ci = None)
```

 crea una gráfica lineal donde los ejes son: columna1 – x, columna2 – y
ci = None para que no muestra el intervalo de confianza de los datos
hue = columna opcional; muestra lineas en diferentes colores por categorias segun una variable

Scatter plot

```
fig = sns.scatterplot(x = 'columna1', y = 'columna2', data = df, hue = 'columna')
```

 crea una gráfica de dispersión donde los marcadores no se solapan

Swarm plot

```
fig = sns.swarmplot(x = 'columna1', y = 'columna2', data = df, hue = 'columna')
```

 crea una gráfica de dispersión donde los marcadores no se solapan

Count plot

```
fig = sns.countplot(x = 'columna1', data = df, hue = 'columna')
```

 crea una gráfica de barras con la cuenta de una variable categórica; se puede especificar solo una variable en la eje x o y, mas una variable opcional con hue

Histogram

```
fig = sns.histplot(x = 'columna1', data = df, hue = 'columna3', kde = True, bins = n)
```

 crea una histograma que muestra la frecuencias de una distribución de datos; donde x es la variable de interés y n es el número de barras
kde = True muestra una curva de la distribucion

Box Plot

```
fig = sns.boxplot(x = 'columna1', data = df, hue = 'columna')
```

 crea un diagrama de cajas; x es la variable de interés; por defecto se muestra con orientación horizontal – usar eje y para orientación vertical

Catplot

```
fig = sns.catplot(x = 'columna1', y = 'columna2', data = df, hue = 'columna', kind = 'tipo')
```

 crea una gráfica que muestra la relacion entre una variable categorica y una variable numerica
kind = 'box' | 'bar' | 'violin' | 'boxen' | 'point' por defecto es strip plot

Pairplot

```
fig = sns.pairplot(data = df, hue = 'columna', kind = 'tipo')
```

 crea los histogramas y diagramas de dispersión de todas las variables numéricas de las que disponga el dataset con el que estemos trabajando; hue es opcional
kind = 'scatter' | 'kde' | 'hist' | 'reg' | 'point' por defecto es scatter

Heatmap

```
sns.heatmap(df.corr(), cmap = 'color_palette', annot = True, vmin = -1, vmax = 1)
```

 crea un heatmap con una escala de colores que refleja los valores de correlacion
annot = True para que aparezcan los valores
vmin/vmax establecen la escala de color

Regplot

```
fig = sns.regplot(x = 'columna1', y = 'columna2', data = df, scatter_kws = {'color': 'blue'}, line_kws = {'color': 'blue'})
```

crea un scatterplot mas la línea de regresión; nos permite encontrar la mejor función de la recta que permite predecir el valor de una variable sabiendo los valores de otra variable

Jointplot

```
sns.jointplot(x = 'columna1', y = 'columna2', data = df, color = 'blue', kind = 'tipo')
```

 crea un scatterplot o regplot con histogramas pegados en los lados para cada variable

Exportar figuras

```
plt.savefig('nombre_de_la_figura.extension')
```

Multigráficas

```
fig, ax = plt.subplots(numero_filas, numero_columnas)
```

 crear una figura con multiples graficas; fig es la figura y ax es un array con subplots como elementos
se establece como es cada grafica con los indices:

```
ax[indice].tipo_grafica(detalles de la grafica)
ax[indice].set_title('titulo')
ax[indice].set_xlabel('xlabel')
ax[indice].set_ylabel('ylabel')
ax[indice].set_xlim(min, max)
ax[indice].set_ylim(min, max)
ax[indice].set_xticklabels(labels = df['column'], rotation = n)
```

 para cambiar los nombres y/o la rotacion de las etiquetas de los valores en los ejes
Crear subplots en un for loop

```
fig, axes = plt.subplots(numero_filas, numero_columnas, figsize = (n, m))
axes = axes.flatten()
for col in df.columns:
    fig = sns.plot(x=col, data=df, ax=axes[i])
```

Crear subplots en un for loop

```
fig, axes = plt.subplots(numero_filas, numero_columnas, figsize = (n, m))
```

Usos de los tipos de gráficas

Datos categóricos

Barras

- muestra la relación entre una variable numérica y categórica
- barplot si tienes una variable numérica
- countplot para contar registros/filas por categoría

Pie chart/quesitos

- determinación de frecuencias

Datos numéricos

Líneas

- tendencias/evolución de una o más variables numéricas (normalmente sobre un período de tiempo)

Histograma

- distribución de una variable numérica

Boxplot

- representación de las medidas de posición más usadas: mediana, IQR, outliers

Scatterplot

- muestra la relación entre dos variables numéricas

Regplot

- scatterplot con una línea de regresión

Swarmplot

- tipo de gráfica de dispersión para representar variables categóricas; evita que se solapan los marcadores

Violinplot

- para visualizar la distribución de los datos y su densidad de probabilidad

Pairplot

- para representar múltiples relaciones entre dos variables

Heatmap

- evaluar la correlación entre las variables en una matriz de correlación

Personalización

Títulos

```
plt.title(label = "titulo")
```

 asignar un titulo a la gráfica

Ejes

```
plt.xlabel("etiqueta_eje_x")
```

 asignar nombre al eje x

```
plt.ylabel("etiqueta_eje_y")
```

 asignar nombre al eje y

```
plt.xlim([n,m])
```

 establece el rango del eje x; donde n es el mínimo y m es el máximo

```
plt.ylim([n,m])
```

 establece el rango del eje y; donde n es el mínimo y m es el máximo

```
fig.set(xlabel = 'etiqueta_eje_x', ylabel = 'etiqueta_eje_y')
```

 asignar nombre a los ejes

```
fig.set_title('titulo')
```

 asignar un titulo a la gráfica

```
fig.set_xlabel(xlabel = "etiqueta_eje_x", fontsize = n)
fig.set_ylabel(ylabel = "etiqueta_eje_y", fontsize = n)
```

```
fig.set(xticks = [1, 2, 3])
fig.set(yticks = [1, 2, 3, 4, 5])
fig.set(xticklabels = ['0%', '20%', '40%', '60%', '80%', '100%'])
fig.set(yticklabels = ['cat1', 'cat2', 'cat3'])
```

```
fig.set_xticklabels(labels = [0, 500, 1000, 1500], size=n)
fig.set_yticklabels(labels = fig.get_yticklabels(), size=n)
```

Para poner etiquetas encima de las barras

```
for indice, valor in enumerate(df ["col"]):
    plt.text(valor+1, indice, valor,
             horizontalalignment='left', fontsize= 16)
```

```
order = df.sort_values('columnay', ascending=False)
['columnax']
sns.set(font_scale=2)
plt.rcParams.update({'font.size': 22})
```

 font size general

Legendas

```
plt.legend(labels = ['label1', 'label2', etc])
```

 muestra la leyenda cuando mostramos la figura

```
plt.legend(bbox_to_anchor = (1, 1))
```

 coloca la leyenda en relación con los ejes

Quitar bordes

```
fig.spines[["top", "right"]].set_visible(False)
```

Linea de tres desviaciones estandares:

```
fig.axvline(x=valor, c='color', label='valor')
fig.axvline(x=valor, c='color', label='valor')
```

Cuadrícula

```
plt.grid()
```

 crea una cuadrícula al fondo de la figura; coge los parámetros:
color = "color"
linestyle = "solid" | "dashed" | "dashdot" | "dotted"
linewidth = n establece la anchura de la linea

Personalización

Colores

```
color = "color"
```

 establece el color de la grafica

```
facecolor = "color"
```

 establece el color del relleno

```
edgecolor = "color"
```

 establece el color de los bordes

Colores en Scatter Plots:

```
c= df['columna'].map(diccionario)
```

diccionario = {"valor1": "color1", "valor1": "color1"}
lista de colores

Paletas Seaborn:

Accent', 'Accent_r', 'Blues', 'Blues_r', 'BrBG', 'BrBG_r', 'BuGn', 'BuGn_r', 'BuPu', 'BuPu_r', 'CMRmap', 'CMRmap_r', 'Dark2', 'Dark2_r', 'GnBu', 'GnBu_r', 'Greens', 'Greens_r', 'Greys', 'Greys_r', 'OrRd', 'OrRd_r', 'Oranges', 'Oranges_r', 'PRGn', 'PRGn_r', 'Paired', 'Paired_r', 'Pastell', 'Pastell_r', 'Pastel2', 'Pastel2_r', 'PiYG', 'PiYG_r', 'PuBu', 'PuBuGn', 'PuBuGn_r', 'PuBu_r', 'PuOr', 'PuOr_r', 'PuRd', 'PuRd_r', 'Purples', 'Purples_r', 'RdBu', 'RdBu_r', 'RdGy', 'RdGy_r', 'RdPu', 'RdPu_r', 'RdYlBu', 'RdYlBu_r', 'RdYlGn', 'RdYlGn_r', 'Reds', 'Reds_r', 'Set1', 'Set1_r', 'Set2', 'Set2_r', 'Set3', 'Set3_r', 'Spectral', 'Spectral_r', 'Wistia', 'Wistia_r', 'YlGn', 'YlGnBu', 'YlGnBu_r', 'YlGn_r', 'YlOrBr', 'YlOrBr_r', 'YlOrRd', 'YlOrRd_r', 'afmhot', 'afmhot_r', 'autumn', 'autumn_r', 'binary', 'binary_r', 'bone', 'bone_r', 'brg', 'brg_r', 'bwr', 'bwr_r', 'cividis', 'cividis_r', 'cool', 'cool_r', 'coolwarm', 'coolwarm_r', 'copper', 'copper_r', 'crest', 'crest_r', 'cubehelix', 'cubehelix_r', 'flag', 'flag_r', 'flare', 'flare_r', 'gist_earth', 'gist_earth_r', 'gist_gray', 'gist_gray_r', 'gist_heat', 'gist_heat_r', 'gist_ncar', 'gist_ncar_r', 'gist_rainbow', 'gist_rainbow_r', 'gist_stern', 'gist_stern_r', 'gist_yarg', 'gist_yarg_r', 'gnuplot', 'gnuplot2', 'gnuplot2_r', 'gnuplot_r', 'gray', 'gray_r', 'hot', 'hot_r', 'hsv', 'hsv_r', 'icefire', 'icefire_r', 'inferno', 'inferno_r', 'jet', 'jet_r', 'magma', 'magma_r', 'mako', 'mako_r', 'nipy_spectral', 'nipy_spectral_r', 'ocean', 'ocean_r', 'pink', 'pink_r', 'plasma', 'plasma_r', 'prism', 'prism_r', 'rainbow', 'rainbow_r', 'rocket', 'rocket_r', 'seismic', 'seismic_r', 'spring', 'spring_r', 'summer', 'summer_r', 'tab10', 'tab10_r', 'tab20', 'tab20_r', 'tab20b', 'tab20b_r', 'tab20c', 'tab20c_r', 'terrain', 'terrain_r', 'turbo', 'turbo_r', 'twilight', 'twilight_r', 'twilight_shifted', 'twilight_shifted_r', 'viridis', 'viridis_r', 'vlag', 'vlag_r', 'winter', 'winter_r'

palette='light:nombre_paleta'|'dark:nombre_paleta'

Marcadores

```
marker = 'tipo'
```

 establece el tipo de marcador; se usa con plt.scatter y plt.plot

"."	Punto	"P"	Más (relleno)
"*"	Pixel	"**"	Estrella
"o"	Círculo	"h"	Hexágono 1
"v"	Triángulo abajo	"H"	Hexágono 2
"^"	Triángulo arriba	"+"	Más
"<"	Triángulo izquierda	"x"	x
">"	Triángulo derecha	"X"	x (relleno)
"8"	Octágono	"D"	Diamante
"s"	Cuadrado	"d"	Diamante fino
"p"	Pentágono		