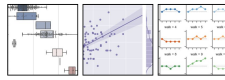


Seaborn



Aprende Python para Ciencia de Datos en [www.datademia.es](http://www.datademia.es)



## Visualización de datos estadísticos con Seaborn

La biblioteca de visualización de Python Seaborn se basa en Matplotlib y proporciona una interfaz de alto nivel para dibujar gráficos estadísticos atractivos.

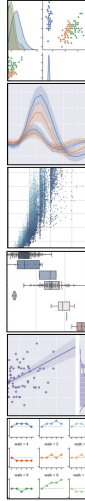
Usa la siguiente convención:

```
>>> import matplotlib.pyplot as plt
>>> import seaborn as sns
```

Los pasos básicos para crear gráficos con Seaborn son:

1. Prepara algunos datos
2. Controla la estética del gráfico
3. Traza con Seaborn
4. Personaliza aún más

```
>>> tips = sns.load_dataset("tips")
>>> sns.set_style("whitegrid")
>>> g = sns.lmplot(x="tip", y="total_bill",
                  data=tips, aspect=2)
>>> g = (g.set_axis_labels("Tip",
                          "Total bill (USD)"))
>>> g.set(xlim=(0,10),ylim=(0,100))
>>> plt.title("title")
>>> plt.show(g)
```



## Datos

```
>>> import pandas as pd
>>> import numpy as np
>>> uniform_data = np.random.rand(10, 12)
>>> data = pd.DataFrame({'x':np.arange(1,101),
                        'y':np.random.normal(0,4,100)})
```

Seaborn también ofrece conjuntos de datos integrados:

```
>>> titanic = sns.load_dataset("titanic")
>>> iris = sns.load_dataset("iris")
```

## Estética

```
>>> f, ax = plt.subplots(figsize=(5,6))
```

 Crea una figura y una subtrazado

### Estilos Seaborn

```
>>> sns.set()
```

 Restablecer los valores predeterminados de seaborn

```
>>> sns.set_style("whitegrid")
>>> sns.set_style("ticks",
                  {"xtick.major.size":8,
                   "ytick.major.size":8})
>>> sns.axes_style("whitegrid")
```

 Establecer los parámetros de matplotlib  
Establecer los parámetros de matplotlib  
Devuelve un dict de paráms o usa con with para configurar estilo

### Funciones de contexto

```
>>> sns.set_context("talk")
>>> sns.set_context("notebook",
                  font_scale=1.5,
                  rc={"lines.linewidth":2.5})
```

 Establecer contexto para "hablar"  
Establecer contexto a "cuaderno",  
escalar elementos de fuente y  
anular el mapeo de param

### Paleta de color

```
>>> sns.set_palette("husl",3)
>>> sns.color_palette("husl")
>>> flatui = ["#9b59b6", "#3498db", "#95a5a6", "#e74c3c", "#34495e", "#2ecc71"]
>>> sns.set_palette(flatui)
```

 Definir la paleta de colores.  
Usar with para configurar temporalmente la paleta  
Establece tu propia paleta de colores

## Rejillas de Eje

```
>>> g = sns.FacetGrid(titanic,
                    col="survived",
                    row="sex")
>>> g = g.map(plt.hist, "age")
>>> sns.factorplot(x="pclass",
                  y="survived",
                  hue="sex",
                  data=titanic)
>>> sns.lmplot(x="sepal_width",
              y="sepal_length",
              hue="species",
              data=iris)
>>> h = sns.PairGrid(iris)
>>> h = h.map(plt.scatter)
>>> sns.pairplot(iris)
>>> i = sns.JointGrid(x="x",
                    y="y",
                    data=data)
>>> i = i.plot(sns.regplot,
              sns.distplot)
>>> sns.jointplot("sepal_length",
                "sepal_width",
                data=iris,
                kind='kde')
```

 Cuadrícula de figuras para trazar relaciones condicionales  
Dibuja un diagrama categórico en un FacetGrid  
Dibuja un gráfico de datos y regresión  
Cuadrícula de figuras para trazar relaciones en parejas  
Cuadrícula de figuras para trazar distribuciones bivariadas en parejas  
Cuadrícula de figuras para trazar bivariado con marginal  
Trazado de distribución bivariada

## Gráficos Categoricos

### Gráfico de puntos/dispersión

```
>>> sns.stripplot(x="species",
                  y="petal_length",
                  data=iris)
```

 Gráfico de puntos con una variable categórica

```
>>> sns.swarmplot(x="species",
                  y="petal_length",
                  data=iris)
```

 Gráfico de puntos categórico sin superposición

### Gráfico de Barras

```
>>> sns.barplot(x="sex",
                y="survived",
                hue="class",
                data=titanic)
```

 Gráfico de barras con intervalos de confianza

### Gráfico de cuenta

```
>>> sns.countplot(x="deck",
                  data=titanic,
                  palette="Greens_d")
```

 Muestra recuento de observaciones

### Gráfico de punto

```
>>> sns.pointplot(x="class", y="survived",
                  hue="sex", data=titanic,
                  palette={"male":"g",
                           "female":"m"},
                  markers=["^","o"],
                  linestyle=["-", "--"])
```

 Mostrar estimaciones puntuales y intervalos de confianza

### Diagrama de caja y bigotes

```
>>> sns.boxplot(x="alive",
                y="age",
                hue="adult_male",
                data=titanic)
>>> sns.boxplot(data=iris, orient="h")
```

 Forma ancha

### Diagrama de violín

```
>>> sns.violinplot(x="age",
                  y="sex",
                  hue="survived",
                  data=titanic)
```

## Gráfico de regresión

```
>>> sns.regplot(x="sepal_width",
                y="sepal_length",
                data=iris,
                ax=ax)
```

 Gráfico de datos y regresión lineal

## Gráfico de distribución

```
>>> plot = sns.distplot(data.y,
                        kde=False,
                        color="b")
```

 Gráfico de distribución univariante

## Gráfico de matriz

```
>>> sns.heatmap(uniform_data, vmin=0, vmax=1)
```

 Mapa de calor

## Personalizaciones

### Objetos de cuadrícula del eje

```
>>> g.despine(left=True)
>>> g.set_ylabels("Survived")
>>> g.set_xticklabels(rotation=45)
>>> g.set_axis_labels("Survived",
                    "Sex")
>>> h.set(xlim=(0,5),
        ylim=(0,5),
        xticks=[0,2.5,5],
        yticks=[0,2.5,5])
```

 Retira la cuadrícula izquierda  
Etiquetas del eje y  
Marcas de eje de x  
Etiquetas de eje  
Límite de los ejes y marcas de los ejes

### Gráfico

```
>>> plt.title("A Title")
>>> plt.ylabel("Survived")
>>> plt.xlabel("Sex")
>>> plt.ylim(0,100)
>>> plt.xlim(0,10)
>>> plt.setp(ax, yticks=[0,5])
>>> plt.tight_layout()
```

 Agregar titulo  
Agregar etiqueta en el eje y  
Agregar etiqueta en el eje x  
Ajustar el límite de y  
Ajustar el límite de x  
Ajusta el trazado  
Ajusta los parámetros del subtrazado

## Muestra o guarda el gráfico

```
>>> plt.show()
>>> plt.savefig("foo.png")
>>> plt.savefig("foo.png",
                transparent=True)
```

 Muestra la figura  
Guarda la figura  
Guarda la figura transparente

## Cerrar y despejar

```
>>> plt.cla()
>>> plt.clf()
>>> plt.close()
```

 Despejar el eje  
Despejar la figura entera  
Cerrar la ventana