

Análisis de datos con Python

Clase 8: Visualización de datos II

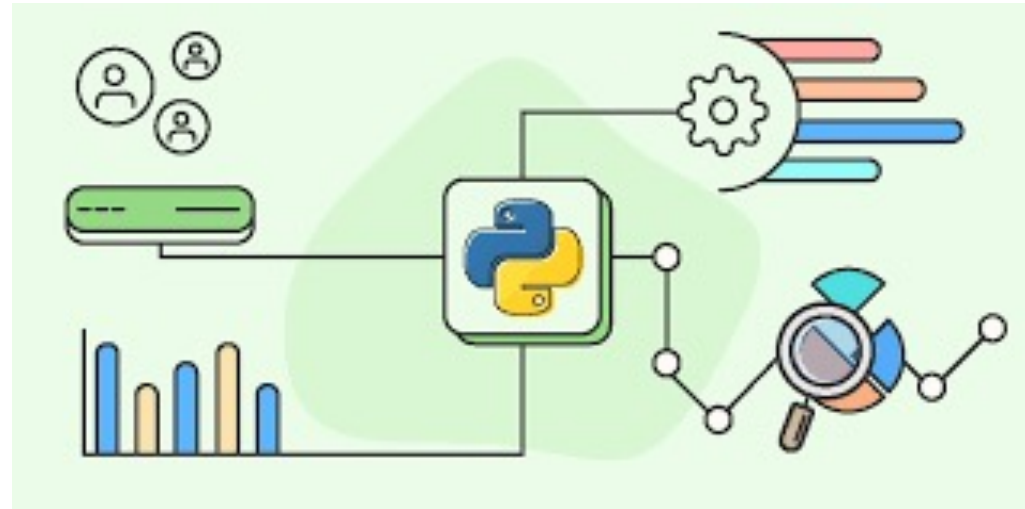
Silvia Salinas Ayaviri



Obejtivos

Aprender más sobre visualización de datos

Aprender sobre la librería seaborn python



Por qué usar Seaborn?

Seaborn es una librería de python usada para ejecutar tareas desafiantes de visualización de datos y está basada en [Matplotlib](#).

Seaborn permite la creación de gráficos a través de las siguientes funcionalidades:

- Una API que se basa en conjuntos de datos que permiten la comparación entre múltiples variables.
- Admite gráficos en cuadrículas múltiples que, a su vez, facilitan la creación de visualizaciones complejas
- Visualizaciones univariadas y bivariadas disponibles para comparar entre subconjuntos de datos
- Disponibilidad de diferentes paletas de colores para revelar varios tipos de patrones.
- Calcula y traza la regresión lineal automáticamente



Seaborn vs Matplotlib

“If Matplotlib “tries to make easy things easy and hard things possible”, Seaborn tries to make a well-defined set of hard things easy too” – Michael Waskom (Creator of Seaborn).

1. Matplotlib se puede personalizar, pero es difícil determinar qué configuraciones se requieren para hacer que los gráficos sean más atractivos. Por otro lado, Seaborn viene con numerosos temas personalizados e interfaces de alto nivel para resolver este problema.

2. Cuando se trabaja con Pandas, Matplotlib no funciona bien cuando se trabajan con DataFrames, mientras que las funciones de Seaborn realmente funcionan en DataFrames.



Instalando dependencias de Seaborn

Dependencias obligatorias para seaborn:

- [NumPy](#)
- [SciPy](#)
- [Matplotlib](#)
- [Pandas](#)

Hay una dependencia también recomendada:

- statsmodels

Para instalar Seaborn, puedes usar cualquiera de los comandos en la ventana de comandos:

```
conda install -c anaconda seaborn
```

o

```
conda install seaborn
```

Para importar:

```
import seaborn as sns
```



Visualizando relaciones estadísticas:

El proceso de comprender las relaciones entre las variables de un conjunto de datos y cómo estas relaciones, a su vez, dependen de otras variables se conoce como análisis estadístico.

relplot():

Esta es una función a nivel de figura que hace uso de otras dos funciones de ejes para visualizar relaciones estadísticas que son:

- scatterplot()
- lineplot()

EXAMPLE:

```
1 | f = sns.load_dataset("flights")  
2 | sns.relplot(x="passengers", y="month", data=f);
```

OUTPUT:



```
1 | import numpy as np  
2 | import pandas as pd  
3 | import matplotlib.pyplot as plt  
4 | import seaborn as sns  
5 | sns.set(style="darkgrid")
```



Visualizando relaciones estadísticas:

OUTPUT:

EXAMPLE:

```
1 | f = sns.load_dataset("flights")  
2 | sns.relplot(x="passengers", y="month", data=f);
```



Como puede ver, los puntos se trazan en 2 dimensiones. Sin embargo, puede agregar otra dimensión utilizando la semántica "tono"='hue'.

EXAMPLE:

```
1 | f = sns.load_dataset("flights")  
2 | sns.relplot(x="passengers", y="month", hue="year", data=f);
```

