Tecnológico de Monterrey | Educación Continua

THE LEARNING GATE

Visualización de datos con Python

Explotar las ventajas que tiene Python y las plataformas de visualización Matplotlib, Pandas y Seaborn, para generar diversos tipos de gráficos que permitan analizar e interpretar resúmenes de datos del mundo real

Dra. Grettel Barceló Alonso
Diseñadora de la competencia

2

Tecnológico de Monterrey | Educación Continua

THE LEARNING GATE

Lecciones

2 Plataformas de visualización en Python
Reconoce el panorama de visualización de Python y algunas de las plataformas más empleadas para seleccionar la más adecuada a su contexto de aplicación

3 Estructura de los datos y tipos de gráficos
Resume la información usando representaciones gráficas acordes a la estructura subyacente de los datos para darles significado e identificar tendencias

4 Gráficas para exploración de datos
Selecciona las gráficas adecuadas de exploración de datos para reconocer la distribución, dispersión y correlación de las variables y puntos de datos a visualizar

5 Gráficas, ejes y figuras
Crea subgráficas y gráficas superpuestas que permitan hacer análisis comparativos y de variabilidad

6 Anotaciones en las gráficas
Integra texto, apuntadores y datos tabulares en las gráficas para obtener representaciones más descriptivas

3

Tecnológico de Monterrey | Educación Continua

THE LEARNING GATE

Plataformas de visualización

2

El panorama de visualización de Python puede parecer abrumador al principio.

Incluso se ha creado PyViz.org, un sitio para ayudar a los usuarios a decidir cuáles son las mejores herramientas de visualización de código abierto de Python para sus propósitos

https://pyviz.org/overviews/index.html

PyViz

4

Tecnológico de Monterrey | Educación Continua

THE LEARNING GATE

Matplotlib es una de las denominadas bibliotecas núcleo (*core*), sobre la que se construyen varias plataformas de nivel superior. Tiene una API completa y potente, que permite perzonalizar cualquier atributo de la figura

Las plataformas de alto nivel que ocupan Matplotlib, proporcionan una API más simple para cubrir las tareas más comunes de manera concisa y conveniente.

Una de las más antiguas es la API .plot () de Pandas. Esta interfaz renderiza gráficos estáticos en libretas de Jupyter o para exportar desde Python, con un comando que puede ser tan simple como df.plot () para un *dataframe* con dos columnas.

Seaborn es otra plataforma basada en matplotlib, para dibujar gráficos estadísticos atractivos e informativos. Ésta se integra estrechamente con las estructuras de datos de Pandas.

5

Tecnológico de Monterrey | Educación Continua

THE LEARNING GATE

Para utilizar estas plataformas debes importarlas en los scripts de Python

```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

Por ejemplo, la función de graficado básica es plot(x,y), que grafica valores de y contra x como líneas y/o marcadores

Con Matplotlib: `plt.plot(df.index, df[column])`
Con Pandas: `df.plot()`
Con Seaborn: `sns.lineplot(x=df.index,y=df[column])`

* Considerando un dataframe df de sólo dos columnas: una de índice y otra de valores

6

Tecnológico de Monterrey | Educación Continua

THE LEARNING GATE

Estructura de los datos y tipos de gráficos

- líneas
- barras
- circular o de sectores (pie)
- histograma
- caja y bigote (box plot)
- dispersión (scatter plot)

Tecnológico de Monterrey | Educación Continua

THE LEARNING GATE

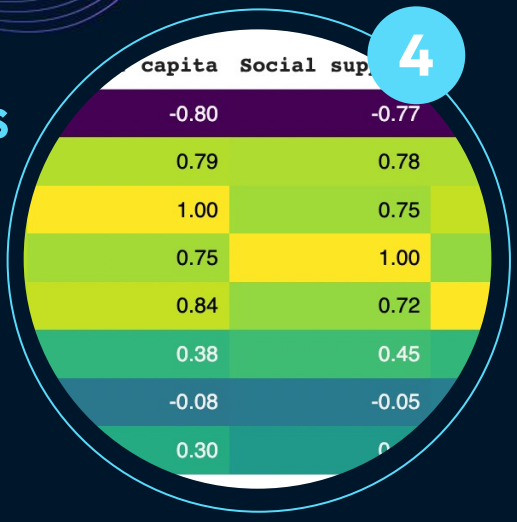
	Matplotlib plt	Pandas df	Seaborn sns
Líneas	`plot()`	`plot()`	`lineplot()`
Barras	`bar()`	`plot(kind = 'bar')` `plot.bar()`	`barplot()`
Circular	`pie()`	`plot(kind = 'pie')` `plot.pie()`	-

Tecnológico de Monterrey | Educación Continua

THE LEARNING GATE

Gráficas para exploración de datos

- líneas
- barras
- circular o de sectores (pie)
- histograma
- caja y bigote (box plot)
- dispersión (scatter plot)

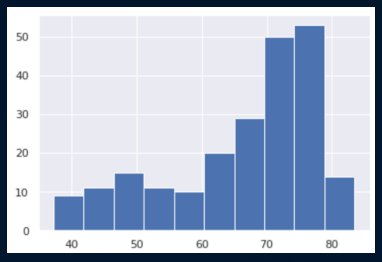


9

Tecnológico de Monterrey | Educación Continua

THE LEARNING GATE

Un histograma es una representación en barras de la distribución de los datos. En el eje horizontal se indican los valores o subrango de valores de las variables y en el vertical sus frecuencias.



Utiliza este tipo de diagrama cuando desees observar el grado de homogeneidad o variabilidad de las columnas cuantitativas continuas del dataframe.

Por defecto se utilizan 10 clases (número de bins) Sin embargo, las clases pueden especificarse explícitamente, cambiando el parámetro bins a un número entero (cantidad de clases igualmente espaciadas) o a una secuencia que define los bordes de las clases (y en este caso las clases pueden estar espaciadas de manera irregular)

```
plt.hist(countries['LifeExpectancy'], bins=[0,20,40,60,80,100])
```

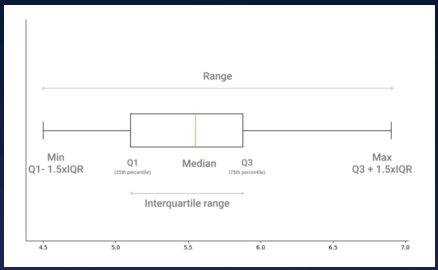
10

Tecnológico de Monterrey | Educación Continua

THE LEARNING GATE

Un diagrama de caja y bigote es una representación de los datos a través de sus cuartiles, mismos que se muestran en la caja junto con la mediana.

Los valores extremos (mínimo y máximo) se representan en los márgenes de los brazos o bigotes; y fuera de éstos, los valores atípicos, que se especifican mediante puntos.



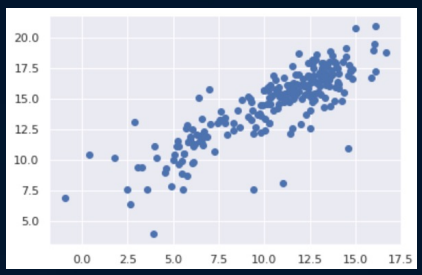
Utiliza este tipo de diagrama cuando desees evaluar la dispersión y la tendencia central de un conjunto de datos.

11

Tecnológico de Monterrey | Educación Continua

THE LEARNING GATE

Los diagramas de dispersión son un conjunto de puntos que muestran cómo dos variables se relacionan; es decir, a través de ellos es posible distinguir el grado de correlación entre dichas variables.



Utiliza este tipo de diagrama cuando desees identificar si hay correlación positiva, negativa o nula entre las columnas del dataframe.

Para cuantificar la correlación entre todos los pares de variables del dataframe, se pueden utilizar mapas de calor (*heatmap*)

12

Tecnológico de Monterrey | Educación Continua

THE LEARNING GATE

	Matplotlib plt	Pandas pd	Seaborn sns
Líneas	`plot()`	`plot()`	`lineplot()`
Barras	`bar()`	`plot(kind = 'bar')` `plot.bar()`	`barplot()`
Circular	`pie()`	`plot(kind = 'pie')` `plot.pie()`	-
Histograma	`hist()`	`plot(kind='hist')` `plot.hist()`	`histplot()`
Caja y bigote	`boxplot()`	`plot(kind='box')` `plot.box()`	`boxplot()`
Dispersión	`scatter()`	`plot(kind='scatter')` `plot.scatter()`	`scatterplot()`

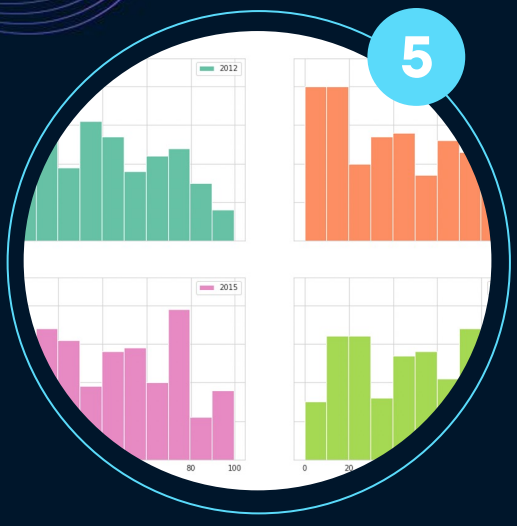
13

Tecnológico de Monterrey | Educación Continua

THE LEARNING GATE

Gráficas, ejes y figuras

- Compartiendo ejes
- Subgráficas
- Ejes duales

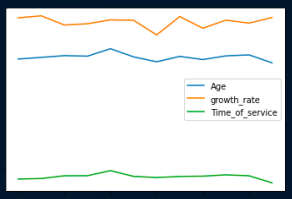


14

Tecnológico de Monterrey | Educación Continua

THE LEARNING GATE

Compartiendo ejes



Con Matplotlib: La función `plot()` grafica todas las series del *dataframe*, si éste se pasa como parámetro.

```
plt.plot(df)
```

Para mostrar en la leyenda los encabezados de cada serie, se utiliza el atributo columns

```
plt.legend(df.columns)
```

Con Pandas: La función plot() graficará todas las series que contenga el *dataframe*, así que basta con una llamada. La leyenda se muestra por defecto en el gráfico

```
df.plot()
```

Con Seaborn: La función *lineplot* hace el trazado de todas las series que contenga el *dataframe* que recibe en el argumento data

```
sns.lineplot(data=df)
```

15

Tecnológico de Monterrey | Educación Continua

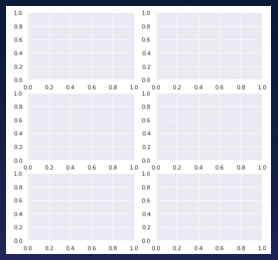
THE LEARNING GATE

Subgráficas

Cuando se ocupa la función `subplots()` de matplotlib sin argumentos, regresa una única figura, con una única gráfica, lo que sería equivalente a una llamada de la función `plot()`.

Pero `subplots()` puede recibir el número de filas y columnas, para acomodar matricialmente series que no comparten los valores de sus ejes.

Por defecto, la función de `subplots()` con un único parámetro hace un apilado vertical. Para dos direcciones es necesario especificar ambas dimesiones.



```
fig, axs = plt.subplots(3,2)
```

16

Tecnológico de Monterrey | Educación Continua

THE LEARNING GATE

Subgráficas

Dentro de la matriz puede generarse cada subgráfica de manera independiente y con configuración propia, especificando las coordenadas en que se ubicará:

Con Matplotlib: `axs[0,0].plot(...df...)`
Con Pandas: `df.plot(...,ax=axs[0,1])`
Con Seaborn: `sns.lineplot(...df..., ax=axs[1,0])`

Para personalizar los elementos de las subgráficas utiliza:

`axs[0,0].set_title()` en lugar de `plt.title()`
`axs[0,0].set_xlabel()` en vez de `plt.xlabel()`

Para el título de la figura utiliza la variable del contenedor fig

`fig.suptitle()`

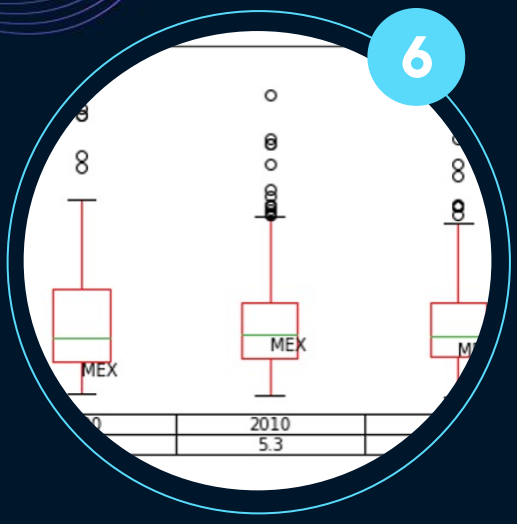
17

Tecnológico de Monterrey | Educación Continua

THE LEARNING GATE

Anotaciones en las gráficas

- Texto
- Anotaciones
- Tablas



18

Tecnológico de Monterrey | Educación Continua

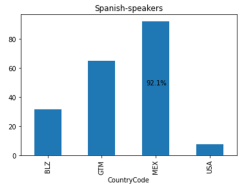
THE LEARNING GATE

text()

Puede usarse en cualquier gráfico, independientemente de la plataforma de trazado empleada

```
plt.text(coordenada en x,
         coordenada en y,
         texto a escribir)
```

```
plt.text(0.58,0.5,'92.1%')
```

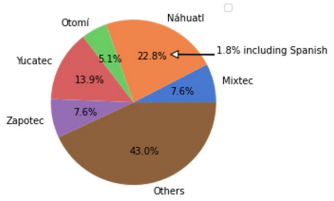


annotate()

Además de ubicar un texto, se puntualiza alguna característica

```
plt.annotate(texto a escribir
             coordenadas (x,y) de la anotación,
             coordenadas (x,y) del texto,
             características de la flecha)
```

```
plt.annotate('1.8% including Spanish',
xy=(0.45,0.58),xytext=(1,0.6),
arrowprops=props_arrow)
```

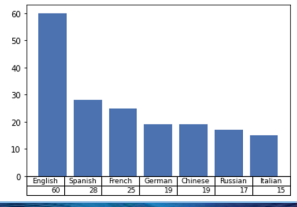


table()

Incluye información en formato tabular bajo el eje decoordenadas horizontal

```
plt.table(valores a incluir,
          nombre de las columnas)
```

```
plt.table(cellText=counts.values,
colLabels=counts.columns)
```



19

Tecnológico de Monterrey | Educación Continua

THE LEARNING GATE

Gracias.

20