

Módulo 3. Gráficos con Python (Matplotlib y Seaborn)

Introducción

En este módulo vamos a aprender cómo generar gráficos usando las librerías Matplotlib y Seaborn. Además, vamos a acompañar la explicación de los distintos tipos de gráficos con la revisión de algunos conceptos de estadística para facilitar su análisis. Comenzamos analizando algunos ejemplos sencillos para luego agregar más detalles que se pueden personalizar.

Es importante señalar que ambos paquetes permiten construir gráficos usando sintaxis distintas. No existe una única manera de expresarla. A lo largo de los ejemplos vamos a ver algunas variaciones de la forma de escribir código para generar gráficos.

Video de inmersión



Unidad 1. Matplotlib

Tema 1: Introducción a Matplotlib y gráficos de línea

Matplotlib es una librería de Python que permite construir visualizaciones estáticas, dinámicas e interactivas. Es muy flexible y da la posibilidad de personalizar muchos elementos de los gráficos. Además, puede trabajar con las distintas estructuras de datos de Numpy y Pandas que ya vimos (matrices y *dataframes*).

Vamos a trabajar con un conjunto de datos que contiene la temperatura diaria máxima y

mínima para tres estaciones meteorológicas: Aeroparque (Capital Federal), Las Lomitas (Formosa) y Río Grande (Tierra del Fuego). Los datos los obtenemos del Servicio Meteorológico Nacional. Vamos a analizar las diferencias en las temperaturas entre las estaciones.

Para comenzar, importamos las librerías con las que vamos a trabajar: además de Numpy y Pandas para procesar los datos, vamos a usar para graficar a Matplotlib y Seaborn (que importamos usando los alias **plt** y **sns** respectivamente).

Figura 1. Importamos los paquetes que vamos a usar y definimos la ruta en la que vamos a trabajar

```
In [1]: import numpy as np, pandas as pd

# Importamos los paquetes para graficar: Matplotlib y Seaborn
import matplotlib.pyplot as plt, seaborn as sns

# Importamos algunos paquetes auxiliares: datetime para trabajar con fechas, y os para cambiar la ruta
import datetime, os

# Usamos esta opción para que muestre los gráficos en la celda de output
%matplotlib inline

# Definimos la ruta en la que vamos a trabajar
os.chdir("C:/Users/Ignacio/Dropbox/TecLab/Modulo 3")
```

Fuente: elaboración propia.

Figura 2. Importamos el *input* (temperaturas.csv) y generamos los *dataframes* que vamos a usar

```
In [2]: # Importamos el archivo con los datos, y definimos un campo de fecha
temp = pd.read_csv("temperaturas.csv", parse_dates = ["fecha"])

# Generamos una variable de amplitud térmica
temp["amplitud"] = temp["maxima"] - temp["minima"]

# Generamos un dataframe que contenga solo los datos de Aeroparque, primero hacemos una copia del general
aep = temp.copy()
aep[aep["estacion"] == "AEROPARQUE AERO"]

# Generamos una variable que tiene el año y el mes, para poder agregar
aep["mes"] = pd.to_datetime(aep["fecha"]).dt.to_period("M")
```

Fuente: elaboración propia.

Vamos a generar un primer gráfico para la temperatura máxima a lo largo de los días, un **gráfico de línea** para esta serie de tiempo. Una forma de hacerlo es usar la función «plot», que corresponde a Pandas pero usa Matplotlib aplicada a un *dataframe* (en este caso **aep**). Usar esta función es un atajo, más adelante vamos a ver otras formas de escribir las mismas instrucciones.

Figura 3. Gráfico de línea

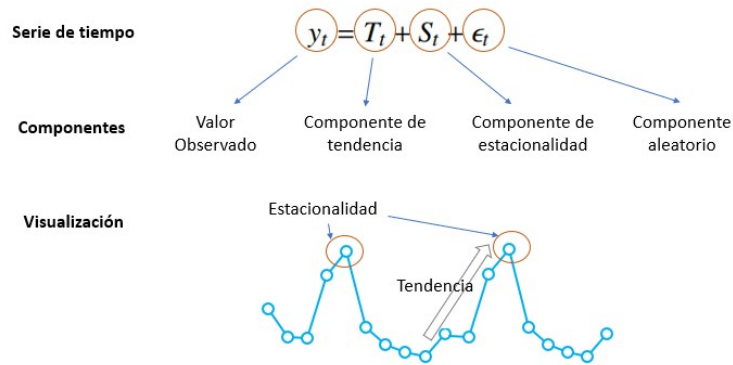
```
In [3]: # Hacemos un gráfico de línea
aep.plot(x = "fecha", y = "maxima", kind = "line")
```

Fuente: elaboración propia.

Especificamos también los argumentos de las variables a usar (máxima en el eje vertical y fecha en el horizontal) y el tipo de gráfico (*kind = line*). Al ejecutar podemos ver la salida y el gráfico de línea nos muestra cómo fluctúa la máxima diaria a lo largo del año.

Nos va a ayudar a analizar esta variable repasar algunos conceptos: podemos interpretar a las series de tiempo como una suma de distintos componentes: la tendencia, la estacionalidad y el ruido o componente aleatorio.

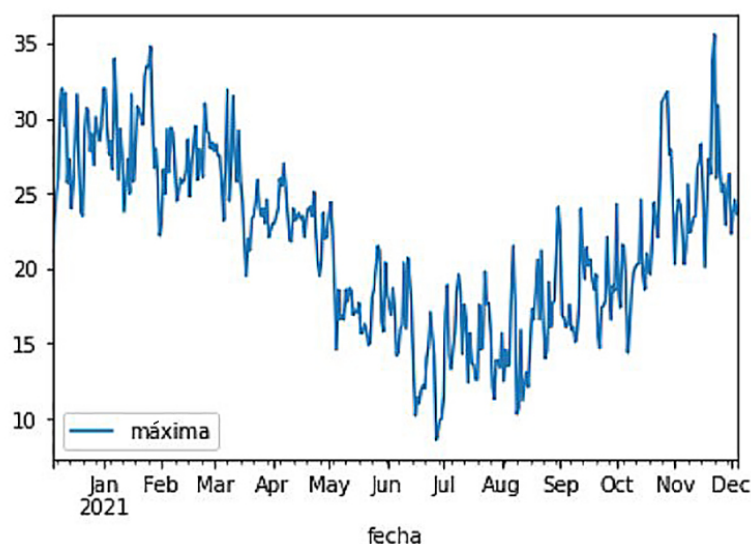
Figura 4. Series de tiempo



Fuente: [Imagen sin título sobre series de tiempo]. (2019). Recuperado de <https://imgur.com/g0PQASf>

- La **tendencia** comprende los movimientos de largo plazo (que se observan en períodos más largos).
- La **estacionalidad**: es el patrón estable y predecible que se repite dentro del período de un año por factores naturales como el clima o culturales como el calendario escolar. Por ejemplo: los que se observan en la producción agrícola (siembras, cosechas) o en los precios de la indumentaria (lanzamientos, liquidaciones).
- El **ruido o componente aleatorio**: por el contrario, incluye todos los movimientos que no pueden predecirse, son erráticos o irregulares.
- En **series económicas**: también podríamos incluir al componente cíclico ya que está relacionado con los movimientos de la economía en su conjunto (expansiones, recesiones) y tiene una duración mayor a un año pero menor a la de los movimientos de tendencia.

Figura 5. Componente estacional y aleatorio



Fuente: Elaboración propia

La estacionalidad de una serie de tiempo es el componente que captura los movimientos de largo plazo.

Verdadero.

Falso.

Justificación

Tema 2: Gráficos de barra, histogramas y de área

Otro de los gráficos más usados es el de **barras**. Estos representan una magnitud usando un atributo visual de forma, tamaño y longitud. Dichos atributos sirven para comparar entre distintas categorías.

Figura 6. Agregamos las observaciones de una frecuencia diaria a una mensual

```
In [4]: # Agregamos las observaciones (a nivel día) a los totales mensuales
aep_mes = aep.groupby(["mes"], as_index = False).agg({"maxima": "mean"})
aep_mes

Out[4]:
```

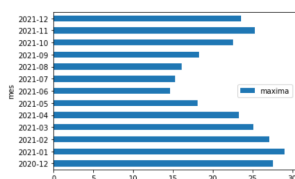
	mes	maxima
0	2020-12	27.592593
1	2021-01	29.016129
2	2021-02	27.153871
3	2021-03	25.148387
4	2021-04	23.303333
5	2021-05	18.083871
6	2021-06	14.623333
7	2021-07	15.287097
8	2021-08	16.119355
9	2021-09	18.263333
10	2021-10	22.532258
11	2021-11	25.260000
12	2021-12	23.525000

Fuente: elaboración propia.

Figura 7. Gráfico de barras horizontales

```
In [5]: # Gráfico de barras horizontales para La temperatura máxima en Aeroporque por mes
aep_mes.plot(x = "mes", y = "maxima", kind = "barh")

Out[5]: <AxesSubplot:ylabel='mes'>
```



mes	maxima
2021-12	23.525000
2021-11	25.260000
2021-10	22.532258
2021-09	18.263333
2021-08	16.119355
2021-07	15.287097
2021-06	14.623333
2021-05	18.083871
2021-04	23.303333
2021-03	25.148387
2021-02	27.153871
2021-01	29.016129
2020-12	27.592593

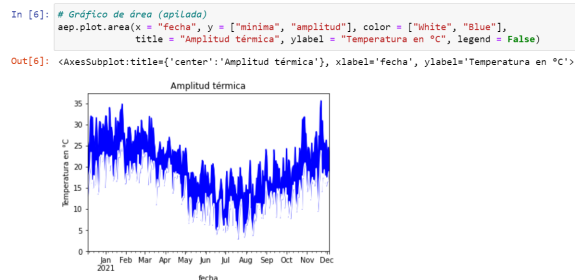
Fuente: elaboración propia.

En nuestro ejemplo podríamos querer visualizar el promedio de las temperaturas máximas a nivel mensual. Para esto, generaremos una tabla auxiliar agrupada por mes usando

«groupby» y «agg». Luego, podemos usar la función «plot», seleccionar las variables que vamos a representar (mes en el eje vertical y temperatura máxima promedio en el eje horizontal). Por último, el tipo de gráfico **barh** (que es un gráfico de barras horizontal).

Otro gráfico común es el de **área**, que es similar al de líneas pero incluye un relleno entre ellas y los ejes. Dentro de estos gráficos, una posibilidad es que las áreas estén apiladas. Vamos a hacer un gráfico de áreas apiladas para mostrar la evolución de la amplitud térmica usando las variables mínima y amplitud.

Figura 8. Gráfico de área



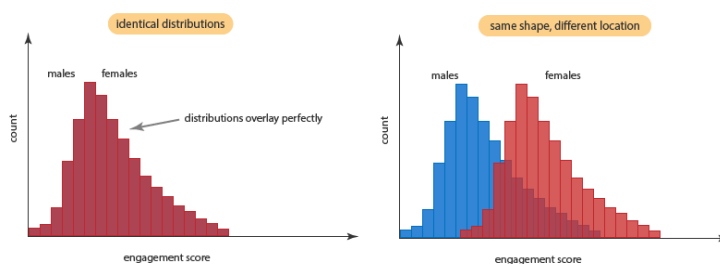
Fuente: elaboración propia.

Al especificar las opciones de plt (Matplotlib), vamos a probar una variación: usar la función «plot» y agregar el tipo de gráfico (área). Entre paréntesis, además de las variables (en este caso hay dos variables en el eje, entre corchetes y separadas por comas) definimos los colores de las series, el título, la etiqueta del eje vertical y la leyenda.

Un tipo de gráfico muy importante es el **histograma** ya que nos permite representar la distribución de una variable numérica continua: en él la frecuencia de los valores se muestra con barras de distinto largo y mismo ancho según intervalos de la variable (por ejemplo, de 10 en 10). Ellas no tienen espacio entre sí ya que son adyacentes.

Los histogramas nos revelan mucha información para poder analizar la distribución de una variable cuantitativa: su posición central, su dispersión y forma.

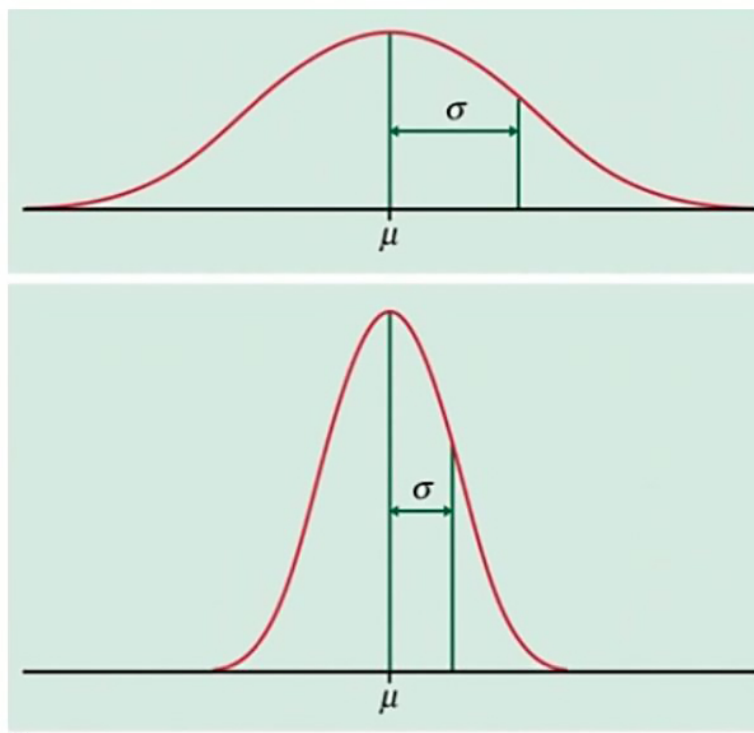
Figura 9. Posición central



Fuente: [Imagen sin título sobre histogramas]. (s.f.). Recuperado de <https://statistics.laerd.com/statistical-guides/mann-whitney-u-test-assumptions.php>

La **tendencia o posición central** nos señala un “centro” en torno al cual están distribuidos los valores de la variable que estamos analizando. Podemos calcular distintas medidas que pueden representar la tendencia central. Las más comunes son la media o promedio, la moda y la mediana (vamos a verla más adelante). En el gráfico de arriba (figura 9) las dos distribuciones son iguales excepto que tienen distinta posición central.

Figura 10. Gráfico: Dispersión

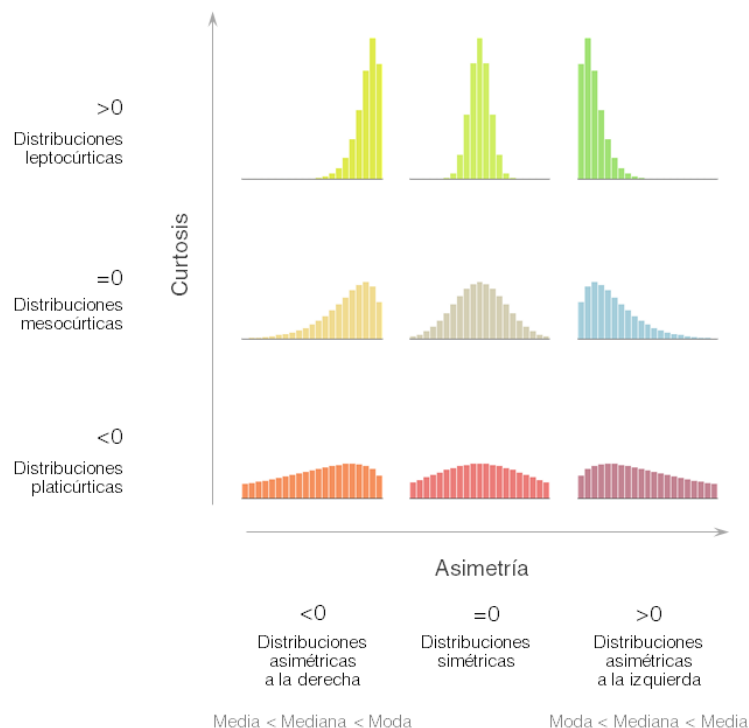


Fuente: [Imagen sin título sobre gráfico]. (s.f.). Recuperado de https://www.nlm.nih.gov/nichsr/stats_tutorial/section2/mod8_sd.html

Otra de las características importantes de una distribución es su **dispersión**, es decir, cuán concentrados o separados están los valores de la media. La medida más común es el desvío estándar. Asimismo, el rango y el rango intercuartílico también sirven para analizar esto.

Las medidas de **forma** son la **asimetría**, la **curtosis** (apuntamiento) y la **modalidad**. La asimetría nos indica cómo se distribuyen los valores a ambos lados del centro: una distribución puede ser simétrica o asimétrica a derecha o izquierda. La curtosis nos muestra si la distribución es más aplanada o más apuntada.

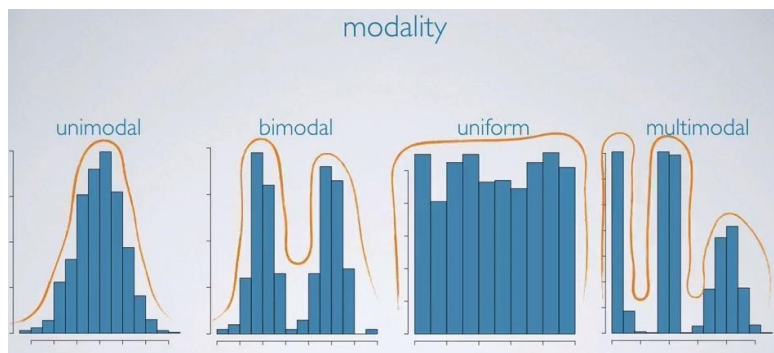
Figura 11. Medidas de forma: asimetría y curtosis



Fuente: elaboración propia.

La modalidad hace referencia a la cantidad de picos que tiene una distribución: puede tener uno solo (unimodal), dos (bimodal), muchos (multimodal) o ninguno (uniforme).

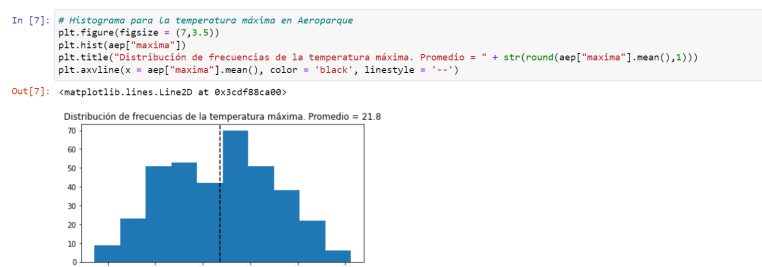
Figura 12. Modalidad



Fuente: [Imagen sin título sobre modalidad]. (s.f.). Recuperado de <https://mathematica.stackexchange.com/questions/173275/a-simple-fast-way-to-estimate-distribution-modality>

En nuestro ejemplo vamos a representar la distribución de frecuencias de la temperatura máxima en Aeroparque con la función «hist». También podemos cambiar el tamaño del gráfico con «figsize», el título con «title» y agregar una línea de referencia sobre el eje X con el valor del promedio en línea punteada.

Figura 13. Histograma



Fuente: elaboración propia.

¿Cuál de estas es una medida de forma?

1. Desvío estándar

2. Mediana

3. Rango intercuartílico

4. Asimetría

Justificación

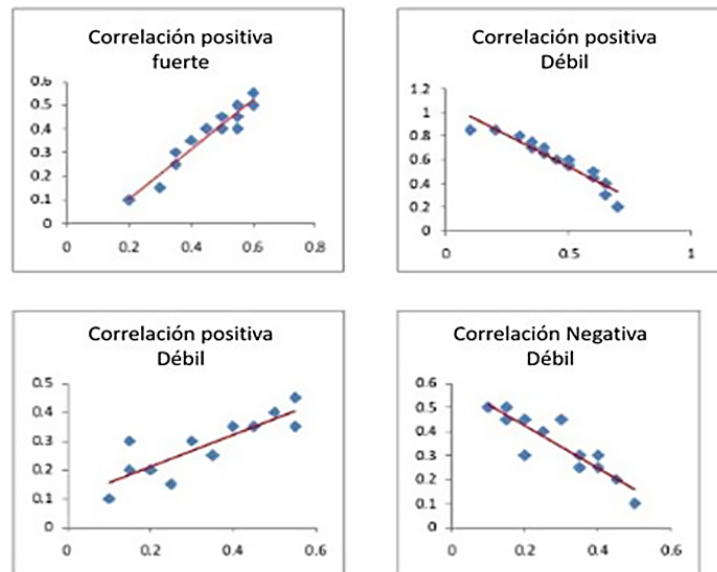
Tema 3: Gráficos de dispersión (*scatterplots*)

Ahora vamos a construir **diagramas o gráficos de dispersión**. Estos muestran una nube de puntos (en este caso en dos dimensiones: x , y) que nos permiten analizar si existe alguna relación estadística entre ambas, es decir, cómo es el comportamiento de una cuando varía la otra.

Podemos resumir esta información en el **coeficiente de correlación** que toma la covarianza entre las dos variables (cuánto varía una cuando varía la otra) y la divide por el producto de los desvíos de ambas variables. Este coeficiente toma valores entre -1 y 1 y es una medida de la **asociación lineal** entre dos variables numéricas. Según su dirección, la correlación puede ser positiva o negativa (lo cual vamos a poder observar en el signo del coeficiente); mientras que esta relación puede ser más fuerte o más débil, lo que va a

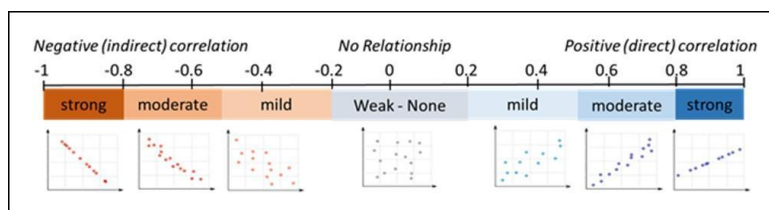
estar reflejado en la magnitud. Mientras más cerca del 1 ó -1 la relación lineal es más fuerte.

Figura 14. Correlaciones



Fuente: Enfermeriacelayane [nombre de usuario]. (2018). Tipos de correlación. Recuperado de <https://blogs.ugto.mx/enfermeriaenlinea/unidad-didactica-5-correlacion-y-regresion/>

Figura 15. Correlaciones



Fuente: [Imagen sin título sobre correlaciones]. (s.f.). Recuperado de https://booksite.elsevier.com/9780128017128/chapter1_5.php

Vamos a construir un gráfico de dispersión para las temperaturas máximas y mínimas en la estación de Aeroparque.

Figura 16. Diagrama de dispersión o scatterplot

```
In [10]: # Podemos cambiar el tamaño del gráfico
plt.figure(figsize = (4,4))

# Definimos el tipo de gráfico (scatter) y las variables
plt.scatter(aep["minima"], aep["maxima"])

# Definimos las etiquetas de los ejes
plt.xlabel("minima"); plt.ylabel("maxima")

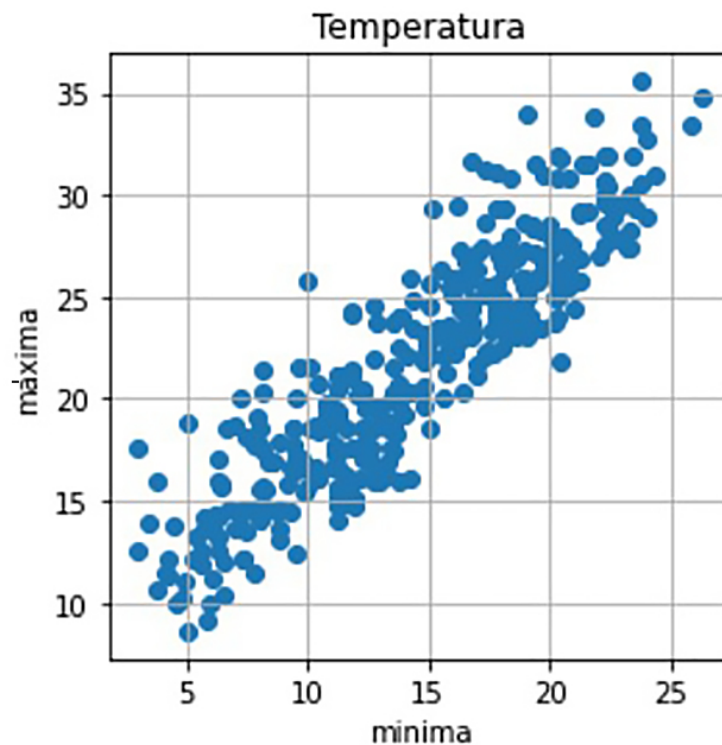
# Definimos el título del gráfico
plt.title("Temperatura")

# Podemos incluir las líneas de cuadrícula
plt.grid()
```

Fuente: elaboración propia.

En esta parte del código, en la opción «figsize», mostramos cómo podemos cambiar el tamaño de la figura. Luego, con la función «scatter» de Matplotlib vamos a generar el gráfico, señalando qué variables usar en el eje x (temperatura mínima) y en el eje y (temperatura máxima). En este ejemplo también definimos otras características del gráfico como las etiquetas de los ejes, el título y las líneas de la cuadrícula (*grid*).

Figura 17. Temperatura



Fuente: Elaboración propia

Una vez ejecutada esta instrucción, podemos observar la relación entre las variables. Como esperamos, existe una relación positiva y muy fuerte entre ellas: en los días en los que la temperatura mínima es más alta, la máxima tiende a ser más alta también, aunque existe mucha variabilidad en la amplitud térmica (la diferencia entre las dos) a lo largo de los días.

También podemos calcular el coeficiente de correlación usando **numpy**, la función «corrcoef» y listando las variables que queremos analizar. La salida es la **matriz de correlaciones**, que contiene los coeficientes entre cada par de variables con unos en la diagonal (las correlaciones de las variables consigo mismas) y los otros coeficientes fuera de la diagonal. En este caso la correlación es +0.896 (positiva y fuerte).

Figura 18. Coeficiente de correlación

```
In [11]: np.corrcoef(aep["minima"], aep["maxima"])  
Out[11]: array([[1., 0.89650901],  
               [0.89650901, 1.]])
```

Fuente: elaboración propia.

El coeficiente de correlación puede tomar valores entre 0 y 1.

Verdadero.

Falso.

Justificación

Tema 4: Gráficos *boxplot*

Los **gráficos de caja o boxplots** son un tipo de visualización que nos ayuda a entender algunos aspectos de la distribución de una variable numérica. Estos se basan en los percentiles que nos señalan los valores hasta los cuales se concentra un determinado porcentaje de la totalidad de las observaciones.

Por ejemplo, si tuviéramos 100 personas y las ordenáramos de menor a mayor por altura, el percentil 10 correspondería a la altura de la 10ª persona en orden de altura, mientras que el percentil 50 (la mediana) sería el valor de la persona en el orden 50; es decir, la que divide la muestra en dos (y así sucesivamente). A los percentiles también se los llama cuantiles.

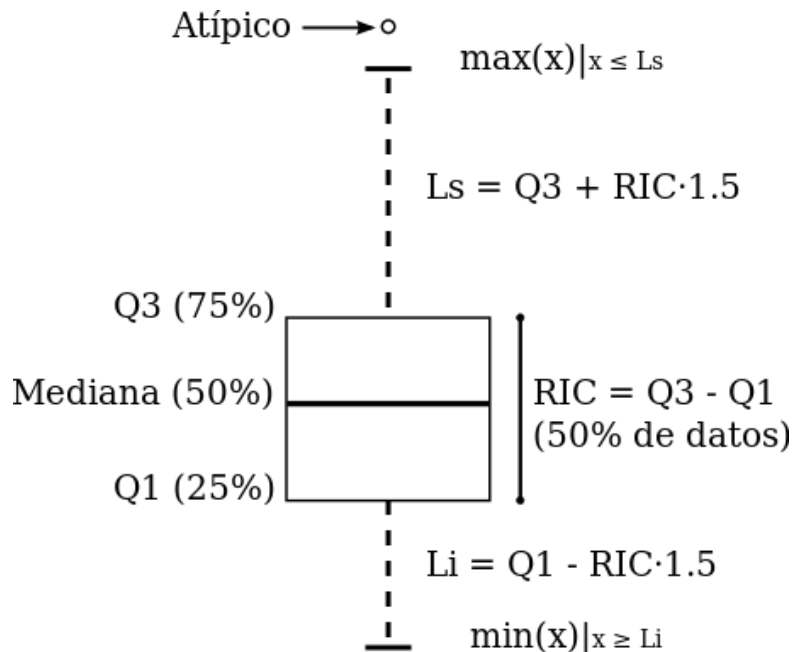
Una forma común de resumir esta información sobre la distribución de la variable es usando deciles (percentiles de 10 en 10), quintiles (dividen la distribución en 5, son percentiles de 20 en 20) o cuartiles (de 25 en 25).

Justamente el *boxplot* toma los cuartiles (expresados como Q1, Q2, etc.) de una variable y los representa con cajas y bigotes. Los límites superiores e inferiores de la caja estarán ubicados en los valores Q1 (percentil 25) y Q3 (percentil 75), mientras que la línea dentro de la caja va a corresponder a la mediana (Q2 o percentil 50). Los bigotes reflejan lo siguiente: se toma la diferencia entre Q3 y Q1, llamada rango intercuartílico o RIC (una medida de dispersión) y se lo multiplica por 1,5, que sumado a Q3 y restado a Q1 nos dan los extremos de los bigotes. Los valores que estén por fuera de estos límites pueden ser identificados como potenciales valores atípicos o extremos (como vimos anteriormente, no es el único criterio) y se los representa con puntos individuales. En algunas variaciones de

este gráfico también se puede agregar la media aritmética o promedio con otro símbolo especial.

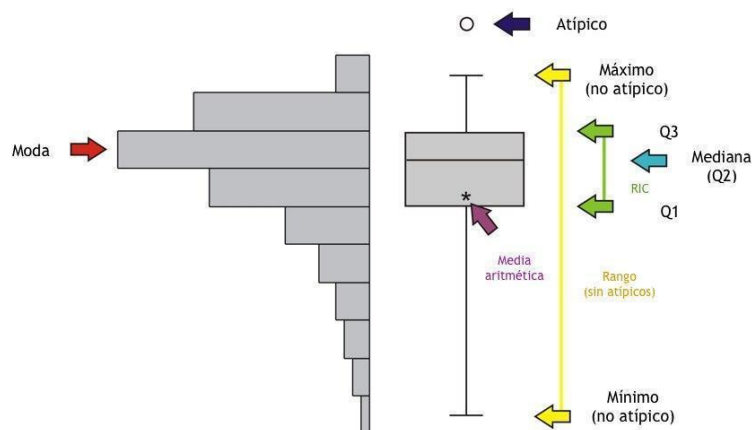
Como otros tipos de gráficos, los *boxplots* poseen ciertas desventajas. Por ejemplo, no permiten distinguir con mayor detalle de la distribución de los valores como la modalidad. La posible solución es emplear los gráficos de violín (los veremos en la siguiente unidad).

Figura 19. Boxplots



Fuente: [Imagen sin título sobre boxplots]. (s.f.). Recuperado de <https://www.qvision.es/blogs/manuel-rodriguez/2015/03/30/interpretacion-de-los-graficos-de-caja-en-el-analisis-descriptivo-e-inferencial/>

Figura 20. Diagrama de caja



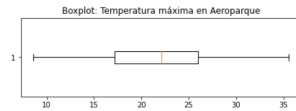
Fuente: [Imagen sin título sobre diagrama de caja]. (s.f.). Recuperado de https://commons.wikimedia.org/wiki/File:Diagrama_de_caja.jpg

En nuestro ejemplo vamos a generar un *boxplot* simple con la temperatura máxima. Usamos la función de Matplotlib «boxplot» y señalamos la variable que queremos usar.

Figura 21. Código

```
In [12]: # Boxplot
plt.figure(figsize = (7,2))
plt.boxplot(aep["maxima"], vert = False)
plt.title("Boxplot: Temperatura máxima en Aeroparque")

Out[12]: Text(0.5, 1.0, 'Boxplot: Temperatura máxima en Aeroparque')
```



Fuente: elaboración propia.

Como chequeo, podemos verificar los valores de los cuartiles que nos interesan: Q1, Q2 o mediana y Q3.

Figura 22. Cuartiles de la distribución

```
In [13]: print("Q1 = " + str(round( np.percentile(aep["maxima"], 25) ,2)))
print("Q2 o mediana = " + str(round( np.percentile(aep["maxima"], 50) ,2)))
print("Q3 = " + str(round( np.percentile(aep["maxima"], 75) ,2)))

Q1 = 17.2
Q2 o mediana = 22.1
Q3 = 26.0
```

Fuente: elaboración propia.

El rango intercuartílico se define como la diferencia entre el tercer y el primer cuartil.

Verdadero.

Falso.

Justificación

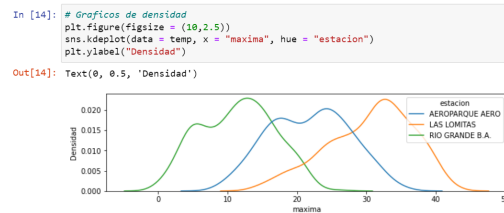
Unidad 2. Seaborn

Tema 1: Introducción a Seaborn, gráficos de densidad y de burbujas

Seaborn es una librería que también nos sirve para hacer visualizaciones en Python. Está construida sobre Matplotlib y una de sus ventajas es que su sintaxis es más sencilla. En esta unidad vamos a ver algunas variaciones de los gráficos más comunes que vimos en la primera parte (pero usando Seaborn).

Los gráficos de densidad, también conocidos como de densidad de *kernel*, son variaciones de los histogramas. Su diferencia es que “suavizan” la forma de la distribución, facilitando la remoción del posible ruido que tenga. Además, tiene la ventaja de no depender de la cantidad de intervalos o contenedores (*bins*) que es necesario definir en los histogramas.

Figura 23. Gráficos de densidad o de densidad de *kernel*

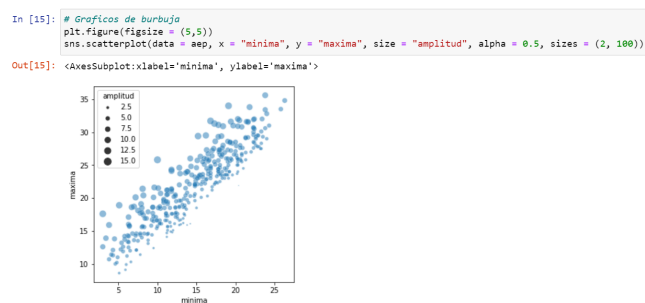


Fuente: elaboración propia.

Para implementar este tipo de gráfico vamos a usar la función «kdeplot» de Seaborn. Graficaremos la distribución de la temperatura máxima desagregándola por estación meteorológica. En el gráfico puede verse la diferencia entre los climas de los tres lugares, así como las distintas características de la distribución (por ejemplo, la bimodalidad en Aeroparque y Río Grande).

También vamos a generar un gráfico de burbujas (*bubbleplot*) que es una variación del diagrama de dispersión. La distinción es que incorpora otra variable o dimensión que está representada en el tamaño de los marcadores. Su beneficio es que permite representar una variable más. La desventaja es que en los círculos las diferentes magnitudes son más difíciles de discriminar.

Figura 24. Gráficos de burbujas



Fuente: elaboración propia.

En nuestro ejemplo vamos a modificar el gráfico de temperaturas mínimas y máximas, agregándole a cada marcador el tamaño por amplitud térmica. La función es la misma que para el *scatterplot* pero agregamos la opción de *size* (tamaño).

Los gráficos de burbujas son una variación de...

1. Gráficos de línea

2. Gráficos de dispersión (scatterplot)

3. Gráficos de área

4. Gráficos de caja y bigotes (boxplot)

Justificación

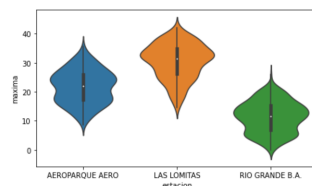
Tema 2: Gráficos *violin plot*

Los gráficos de violín combinan los elementos de los gráficos de caja (*boxplots*) y de densidad. Permiten analizar visualmente los cuartiles y *outliers*. También incluyen un gráfico de densidad en espejo sobre el mismo eje que muestra en más detalle la distribución de la variable.

Figura 25. Gráficos de violín

```
In [16]: # Violin plots
plt.figure(figsize = (7,4))
sns.violinplot(data = temp, x = "estacion", y = "maxima")

Out[16]: <AxesSubplot:xlabel='estacion', ylabel='maxima'>
```



Fuente: elaboración propia.

En nuestro ejemplo vamos a graficar las distribuciones de las temperaturas mínimas por estación meteorológica. Si observamos detenidamente, podremos notar las similitudes con el gráfico de densidades del tema anterior.

Los gráficos de violín (violin plots) combinan los diagramas de dispersión con los de densidad.

Verdadero.

Falso.

Justificación

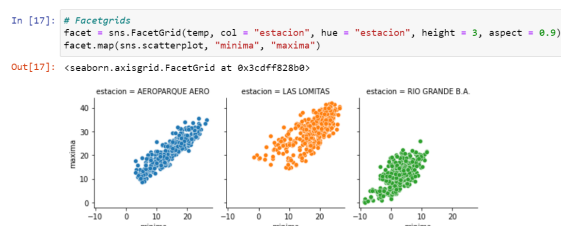
Tema 3: Paneles de gráficos (*facets*)

Muchas veces necesitamos analizar la distribución de alguna variable (pero agrupando sus valores de acuerdo a alguna dimensión). En estos casos es muy útil usar **paneles de gráficos** o *facets*. Ellos nos permiten generar múltiples visualizaciones, compartiendo los mismos ejes y tamaños, pero mostrando distintos subconjuntos de los datos originales. También reciben otros nombres como *small multiples* o *trellis charts*.

En nuestro caso, vamos a graficar paneles de **diagramas de dispersión** (*scatterplots*) para cada una de las estaciones meteorológicas que tenemos en el conjunto de datos original (Aeroparque, Las Lomas y Río Grande).

Vamos a usar la función «FacetGrid» de Seaborn. Definiremos la tabla de origen de los datos, la variable por la que vamos a segmentar estos gráficos (en este caso, la estación), la variable por la que van a tomar el color los marcadores del *scatterplot* (también la estación) y los parámetros que controlan el tamaño (*height* o altura y *aspect* o proporción).

Figura 26. Paneles de gráficos



Fuente: elaboración propia.

Otra opción que nos puede ser útil es la posibilidad de componer gráficos que tengan subgráficos. Estos pueden ser de distintos tipos de visualizaciones (por ejemplo, de líneas y de barras) o no compartir los mismos ejes. En nuestro caso vamos a construir uno con dos subgráficos: un *scatterplot* como los que vimos antes y otro con un histograma más una línea con la estimación de la función de densidad (*Kernel Density Estimation* o KDE).

Figura 27. Subgráficos



Fuente: elaboración propia.

Para esto, vamos a usar la función «subplot». En ella, los argumentos son el número de filas, de columnas y, para cada gráfico que construyamos, su orden. Los tres números van juntos, sin espacio. Luego generamos cada uno de los subgráficos, incluyendo la primera línea que indica a cuál de los subgráficos pertenece. En este caso, probamos la opción «savefig» que nos sirve para exportar el gráfico.

Los gráficos de panel contienen distintas variables y ejes.

Verdadero.

Falso.

Justificación

Tema 4: Gráficos marginales (histogramas, rugs y boxplots)

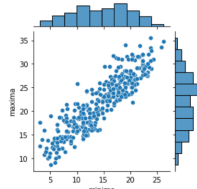
Como mencionamos en los módulos anteriores, una vez que logremos entender los distintos componentes de los gráficos, podemos ser flexibles y construir variaciones de estos según lo que necesitemos. Una forma de agregar información sin agregar mucha “tinta” es incorporar gráficos auxiliares sobre los márgenes para facilitar la lectura aprovechando el mismo eje. Estos pueden ser de distintos tipos, como histogramas, *boxplots* o gráficos de densidades. Vamos a ver cómo implementar estos ejemplos con Seaborn.

Primero vamos a ver la forma más sencilla pero menos flexible, que es usar «jointplot». Con los argumentos generales (*dataframe* y variables en el eje x, y) es suficiente. También podemos ajustar otros parámetros como el tamaño.

Figura 28. Gráfico de barra marginales usando «jointplot»

```
In [19]: # Scatterplot + Gráficos de barra marginales
sns.jointplot(data = aep, x = "minima", y = "maxima", height = 4)

Out[19]: <seaborn.axisgrid.JointGrid at 0x3ce0f5700>
```



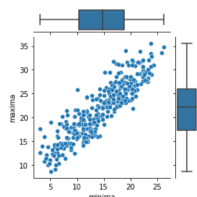
Fuente: elaboración propia.

Podemos también usar «JointGrid» ya que nos permitirá personalizar más características del gráfico. En la primera línea definimos las opciones generales y en las siguientes dos, el gráfico central y los gráficos marginales. Vamos a ver un ejemplo con *boxplots*.

Figura 29. Boxplots marginales

```
In [20]: # Scatterplot + Histogramas marginales
j = sns.JointGrid(data = aep, x = "minima", y = "maxima", height = 4)
j.plot_joint(sns.scatterplot)
j.plot_marginals(sns.boxplot)

Out[20]: <seaborn.axisgrid.JointGrid at 0x3ce01df9d0>
```



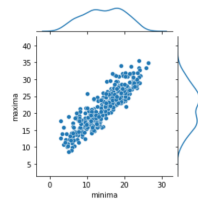
Fuente: elaboración propia.

Y otro con funciones de densidad:

Figura 30. Funciones de densidad marginales

```
In [21]: # Scatterplot + Gráficos de densidad marginales
j = sns.JointGrid(data = aep, x = "minima", y = "maxima", height = 4)
j.plot_joint(sns.scatterplot)
j.plot_marginals(sns.kdeplot)

Out[21]: <seaborn.axisgrid.JointGrid at 0x3ce0361280>
```



Fuente: elaboración propia.

Por último, una alternativa útil es incluir en los ejes los *rugs* o marcas para cada observación. Esto puede servir en casos en los que tengamos una cantidad no demasiado grande de registros para graficar o en los que sea necesario resaltar *outliers*, por ejemplo. En este caso, incluimos dos líneas, una en la que especificamos las opciones del *scatterplot* y otra para las opciones del *rugplot*.

Una de las ventajas de los gráficos auxiliares sobre los márgenes es poder agregar más información sin afectar demasiado al “ratio datos-tinta”.

Verdadero.

Falso.

Justificación

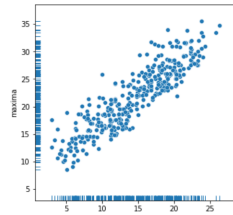
Figura 31. Rugplot

```
In [22]: # Rugplots
plt.figure(figsize=(5,5))

# Esta parte grafica el interior, los puntos del diagrama de dispersión
sns.scatterplot(data = aep, x = "minima", y = "maxima")

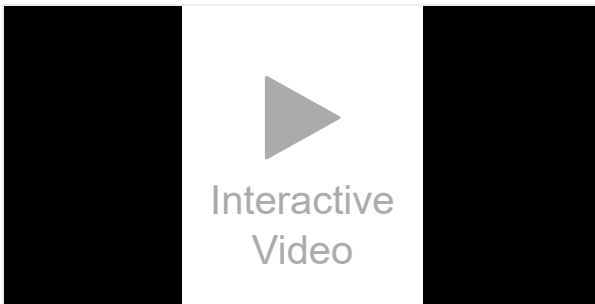
# Esta parte grafica los rugs, las marcas para las observaciones individuales
sns.rugplot(data = aep, x = "minima", y = "maxima")

Out[22]: <AxesSubplot: xlabel='minima', ylabel='maxima'>
```



Fuente: elaboración propia.

Video de habilidades



Cierre

En este módulo integramos lo que aprendimos anteriormente: lo que ya sabemos sobre visualización de datos y sobre cómo procesar datos en Python. Logramos combinarlo para construir visualizaciones usando las librerías Matplotlib y Seaborn.

Todo esto nos va a servir para el siguiente entorno de aprendizaje. Vamos a explorar cómo hacernos preguntas, formularnos hipótesis y comenzar a validarlas con los datos aplicando herramientas de análisis exploratorio.

Glosario



Referencias

[Imagen sin título sobre boxplots]. (s.f.). Recuperado de <https://www.qvision.es/blogs/manuel-rodriguez/2015/03/30/interpretacion-de-los-graficos-de-caja-en-el-analisis-descriptivo-e-inferencial/>

[Imagen sin título sobre correlaciones]. (s.f.). Recuperado de https://booksite.elsevier.com/9780128017128/chapter1_5.php

[Imagen sin título sobre diagrama de caja]. (s.f.). Recuperado de https://commons.wikimedia.org/wiki/File:Diagrama_de_caja.jpg

[Imagen sin título sobre gráfico]. (s.f.). Recuperado de https://www.nlm.nih.gov/nichsr/stats_tutorial/section2/mod8_sd.html

[Imagen sin título sobre histogramas]. (s.f.).

Recuperado de <https://statistics.laerd.com/statistical-guides/mann-whitney-u-test-assumptions.php>

[Imagen sin título sobre modalidad]. (s.f.). Recuperado de <https://mathematica.stackexchange.com/questions/173275/a-simple-fast-way-to-estimate-distribution-modality>

[Imagen sin título sobre series de tiempo]. (2019). Recuperado de <https://imgur.com/g0PQASf>

Enfermeriacelayane [nombre de usuario]. (2018). Tipos de correlación. Recuperado de <https://blogs.ugto.mx/enfermeriaenlinea/unidad-didactica-5-correlacion-y-regresion/>