ACÁMICA

TEMA DEL DÍA

Visualizaciones

Visualizar datos es una parte fundamental del trabajo del Data Scientist. Hoy veremos cómo hacer gráficos en Python usando la librería Matplotlib.

Agenda

Daily

Explicación: Matplotlib

Break.

Hands-on training

Bonus Track: gráficos engañosos.

Cierre.

Daily

Daily

Sincronizando...

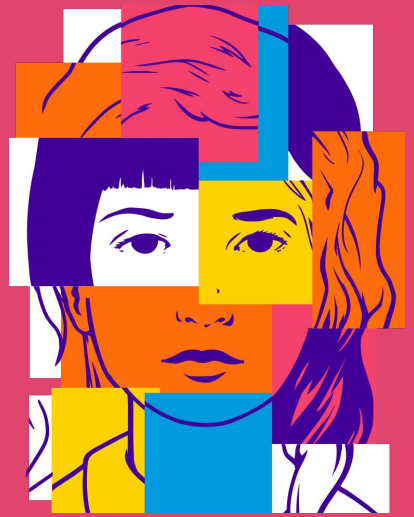
Toolbox

¿Cómo te ha ido?
¿Obstáculos?
¿Cómo seguimos?

Challenge

¿Cómo te ha ido?
¿Obstáculos?
¿Cómo seguimos?

Por qué visualizar



Visualizar los datos es una parte fundamental del análisis en ciencia de datos.

¿Por qué visualizar?

No solo sirve para **comunicar** (que es una parte fundamental del trabajo) sino que también es una herramienta esencial para **comprender el dataset** con el que estamos trabajando.

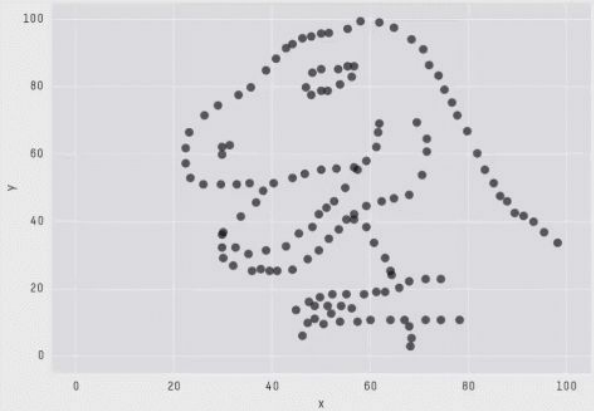
¿Por qué visualizar?

Hay veces que sólo indicadores numéricos no alcanzan para describir las características principales de nuestro dataset.

¿Por qué visualizar?

Hay veces que sólo indicadores numéricos no alcanzan para describir las características principales de nuestro dataset.

Conjuntos de datos con tres métricas en común: su media, su varianza y correlación.

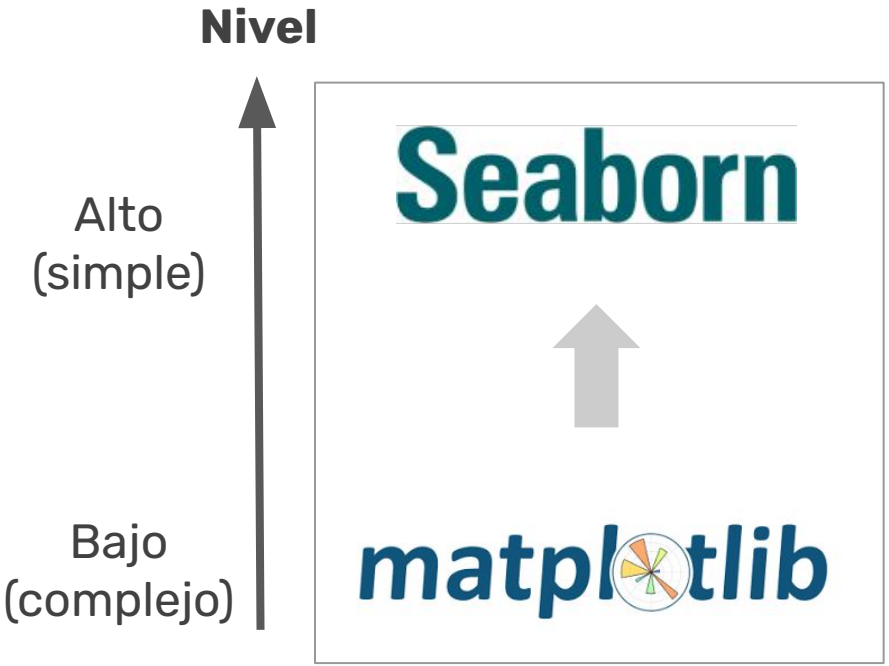


X Mean: 54.2659224
Y Mean: 47.8313999
X SD : 16.7649829
Y SD : 26.9342120
Corr. : -0.0642526

Matplotlib

Herramientas de Visualización

Herramientas de Visualización



plotly

Bokeh

¡Usaremos éstas!

antes de empezar...

Introducción a Matplotlib

Por si no la tienen instalada de meetings anteriores:

conda install matplotlib

Introducción a Matplotlib

La documentación de matplotlib es **excelente**, es importantísimo aprovecharla:

https://matplotlib.org/index.html

¡ahora sí!


matplotlib

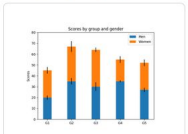
Documentación

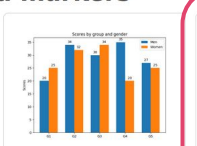
Gallery

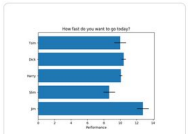
This gallery contains examples of the many things you can do with Matplotlib. Click on any image to see the full code.

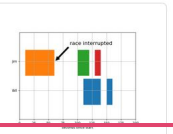
For longer tutorials, see our tutorials page. You can also find external resources and a FAQ in our user guide.

Lines, bars and markers

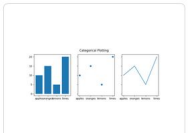

Stacked Bar Graph

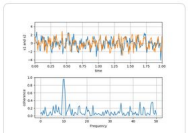

Grouped bar chart with labels

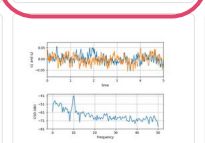

Horizontal bar chart

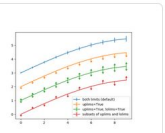


Broken Barh


Plotting

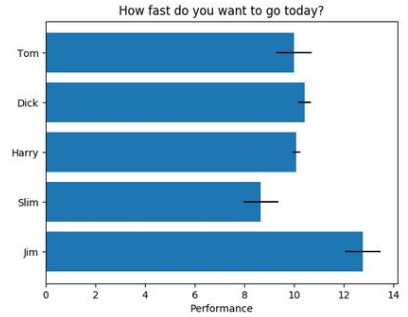

Plotting the


CSD Demo


Errorbar limit

Horizontal bar chart

This example showcases a simple horizontal bar chart.




```
import matplotlib.pyplot as plt
import numpy as np

# Fixing random state for reproducibility
np.random.seed(19680801)


plt.rcdefaults()
fig, ax = plt.subplots()

# Example data
people = ('Tom', 'Dick', 'Harry', 'Slim', 'Jim')
y_pos = np.arange(len(people))
performance = 3 + 10 * np.random.rand(len(people))
error = np.random.rand(len(people))

ax.barh(y_pos, performance, xerr=error, align='center')
ax.set_yticks(y_pos)
ax.set_yticklabels(people)
ax.invert_yaxis()  # labels read top-to-bottom
ax.set_xlabel('Performance')
ax.set_title('How fast do you want to go today?')

plt.show()
```

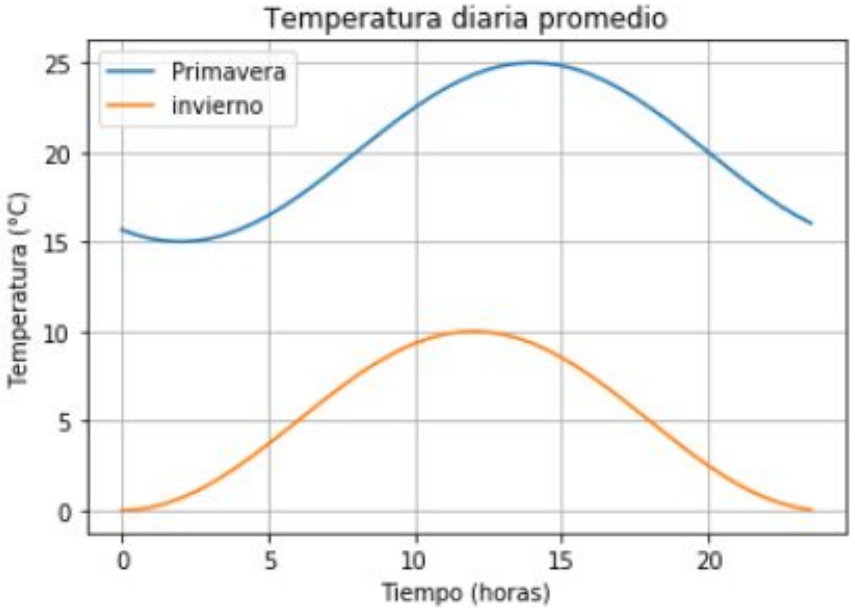
matplotlib

Graficar en un notebook

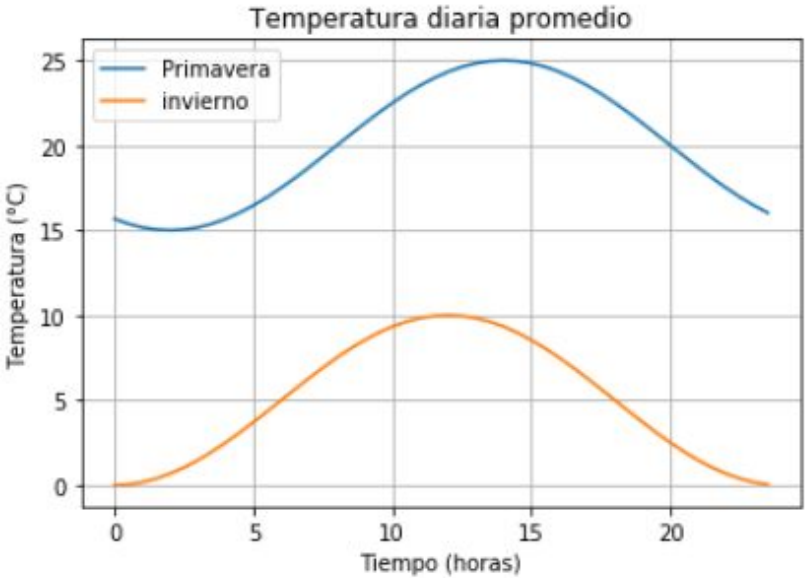
La parte de la librería que usaremos para graficar es **matplotlib.pyplot** y se suele importar con el nombre **plt**

```
[ ]: import matplotlib.pyplot as plt
```

Veamos cómo graficar este cuadro:



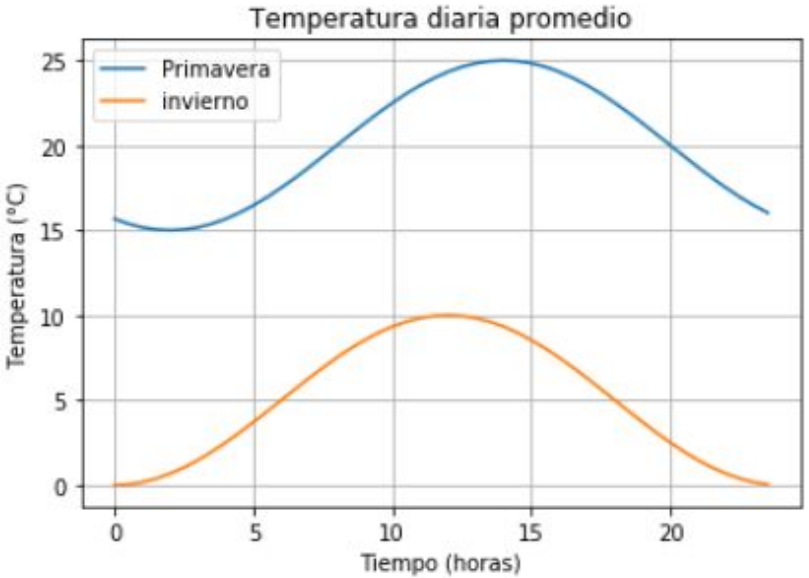
Veamos cómo graficar este cuadro:



Escribimos los valores a graficar

```
[8]: x = np.arange(0.0, 24, 0.5)
     y1 = 20 + 5*np.sin(2*np.pi*(x - 8)/24)
     y2 = 5 + 5*np.sin(2*np.pi*(x - 6)/24)
```

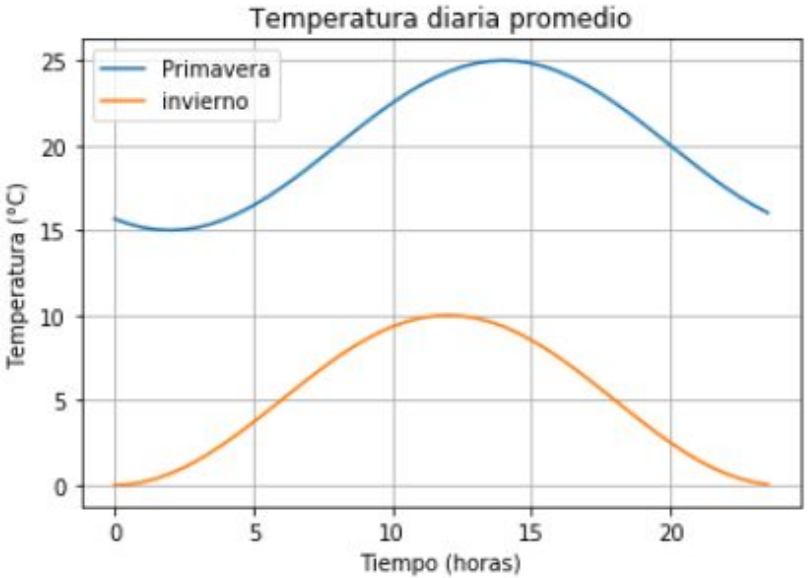
Veamos cómo graficar este cuadro:



Ploteamos las dos líneas, dándole un nombre a cada una mediante el argumento “label”.

```
plt.plot(x, y1, label = 'Primavera')
plt.plot(x, y2, label = 'invierno')
```

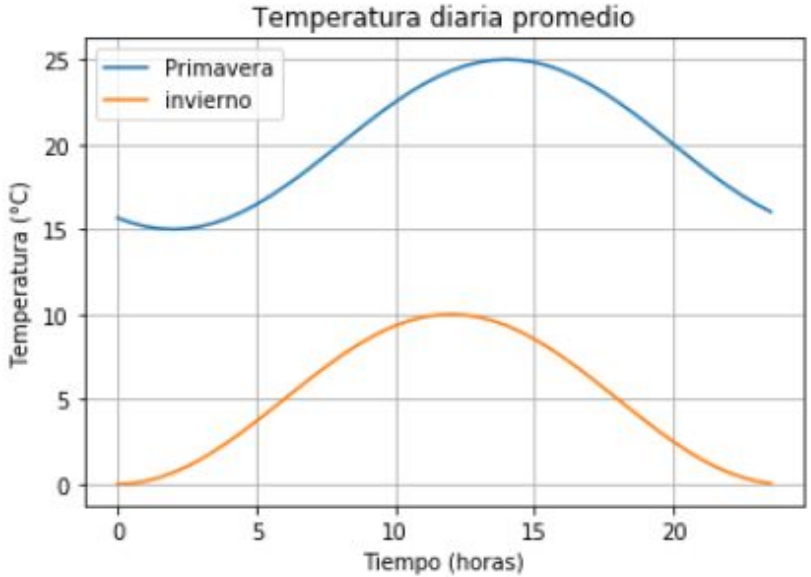
Veamos cómo graficar este cuadro:



Seteamos los títulos a cada eje.

```
plt.xlabel('Tiempo (horas)')
plt.ylabel('Temperatura (°C)')
plt.title('Temperatura diaria promedio')
```

Veamos cómo graficar este cuadro:




```
plt.grid()
```
← Activamos la grilla.

```
plt.legend()
```
← Agregamos la leyenda.

```
plt.show()
```
← ¡Mostramos el gráfico!

Veamos cómo graficar este cuadro:

[8]:
```
x = np.arange(0.0, 24, 0.5)
y1 = 20 + 5*np.sin(2*np.pi*(x - 8)/24)
y2 = 5 + 5*np.sin(2*np.pi*(x - 6)/24)

# Ploteamos las dos lineas, dandole un nombre a cada una mediante el parámetro 'label'.
# Notar que para agregar una curva, simplemente debemos poner una debajo de la otra
plt.plot(x, y1, label = 'Primavera')
plt.plot(x, y2, label = 'invierno')

plt.xlabel('Tiempo (horas)')
plt.ylabel('Temperatura (°C)')
plt.title('Temperatura diaria promedio')

# Este comando enciende la grilla de referencia
plt.grid()

# Agregamos la leyenda al gráfico
plt.legend() #loc='upper center', shadow=True, fontsize='x-large')
plt.show()
```

¡BREAK!

Ph. Credit: Drew Coffmann

Hands-on training

Hands-on training

DS_Toolbox_06_Matplotlib.ipynb

Visualizar datos también puede ser engañoso...

TED Ed

Lessons Worth Sharing

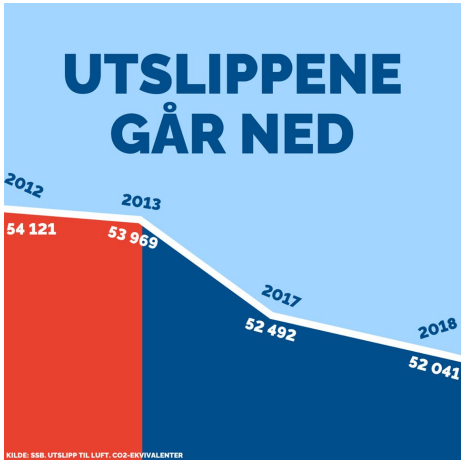
TED Ed

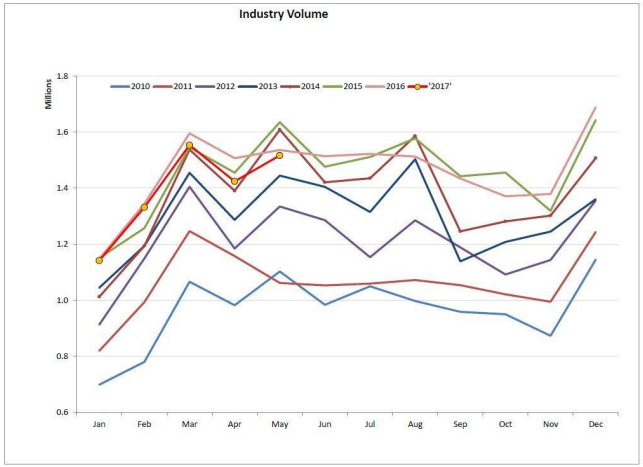
0:02 / 4:09

Cómo detectar un gráfico engañoso - Lea Gaslowitz

Ejercitación

¿Crees que los siguientes gráficos son engañosos o están mal hechos? Explica por qué.





Fuente: https://viz.wtf/

Ejercitación

En grupos de 3 personas, pensar y hacer una visualización con datos de la coyuntura actual que sea engañoso o esté malinterpretado.

Por ejemplo, pueden usar datos sobre encuestas, intención de votos en elecciones, datos sobre la pandemia de Covid-19, etc.

¡No hace falta programar! Pueden hacerlo en lápiz y papel.

Recursos

Probabilidad y Estadística

- Python Data Science Handbook - Capítulo 4, “Visualization with Matplotlib”.

Data Science

Sprint 1 (Meeting #6)

Nos tomamos unos minutos para completar esta encuesta.

Queremos saber cómo valoran mi tarea hasta acá. ¡Les va a llevar solo un minuto!

Para la próxima

- Termina el notebook de hoy.
- Lee la Toolbox 07
- Resuelve el Challenge.

ACÁMICA