

Bits, pixeles y palabras:

# ¿Cómo hacer data storytelling con Python?

Sebastián Flores  
PyCon Chile 2023  
25 Noviembre 2023



PyCon Colombia 2020	PyCon Argentina 2021	PyCon Latam 2021	PyCon Chile 2021	PyCon Chile 2022	PyDay Chile 2022	PyCon Chile 2023
Presentaciones y encuestas interactivas en Jupyter Notebooks y RISE	Coding as a Service: Librería pypsider, aprendizajes y metodología	Presentaciones Interactivas con Jupyter Notebook + RISE	WebApps con Streamlit ¡Más fácil que la tabla del uno, po!	Usando Streamlit en la industria	Vizzualización de datos: Breve introducción al arte de visualizar datos	¿Cómo hacer Data Storytelling con Python?

Patrón común:

¿Cómo hacer que mis exposiciones más entretenidas y tengan mayor impacto?

“Escribe para tí mismo.  
Tendrás garantizado una audiencia de 1”



Episodios

01 En búsca del arte perdido

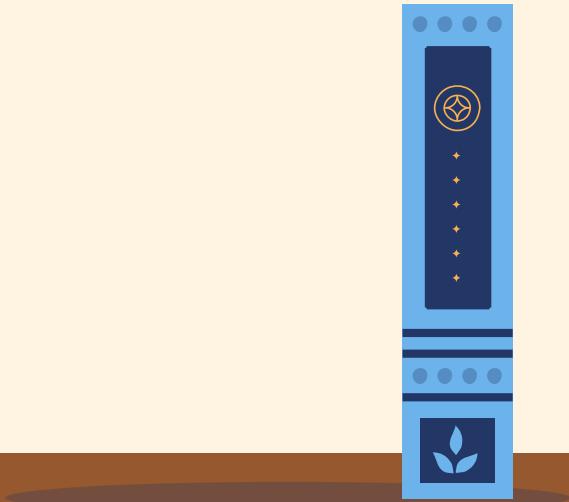
02 El código maldito

03 La última mirada

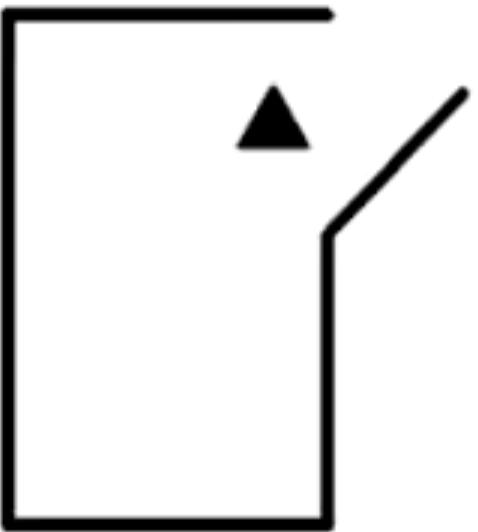


Episodio

# 01 En busca del arte perdido



## Test de Heider-Simmel (1944)

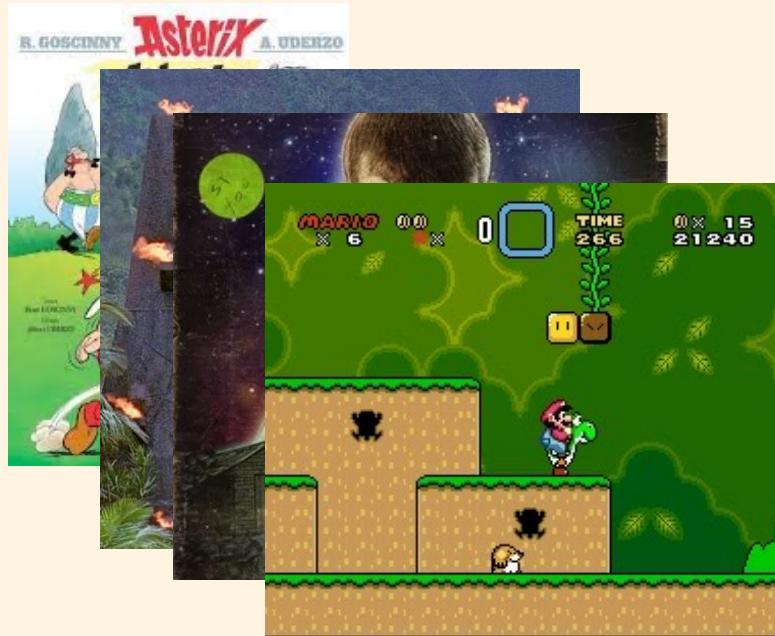


# Paraleidolia





# Contar, escuchar y vivir historias es profundamente humano



# Demostración empírica: recordamos historias no datos

← → C https://es.wikipedia.org/wiki/Métodos\_de\_integración G ↗ ☆ ↘

Contenidos [ocultar]

- Inicio
- Generalidades
- > Integración directa
- Funciones analíticas
- > Integración por cambio de variable
- > **Integración por Partes**
- > Integrales de funciones trigonométricas
- Integrales de funciones racionales
- Integración numérica
- Notas
- Véase también
- Bibliografía
- > Enlaces externos

**Integración por Partes** [editar]

En el cálculo y en general en el [análisis matemático](#), **integración por partes** es el proceso que encuentra la integral de un producto de funciones en términos de la integral de sus derivadas y antiderivadas. Frecuentemente usado para transformar la antiderivada de un producto de funciones en una antiderivada, por lo cual, una solución puede ser hallada más fácilmente.

El método de **integración por partes** es el que resulta de aplicar el siguiente teorema

**Teorema** [editar]

Si  $f'$  y  $g'$  son funciones continuas entonces

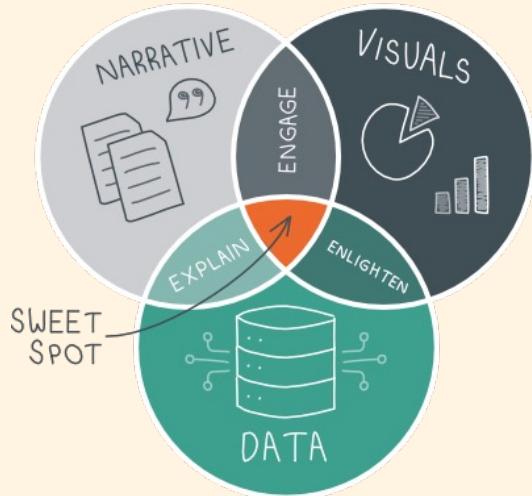
$$\int f(x)g'(x)dx = f(x)g(x) - \int f'(x)g(x)dx$$
$$\int_a^b f(x)g'(x)dx = f(x)g(x) \Big|_a^b - \int_a^b f'(x)g(x)dx$$

Típicamente se encuentra la fórmula como sigue:

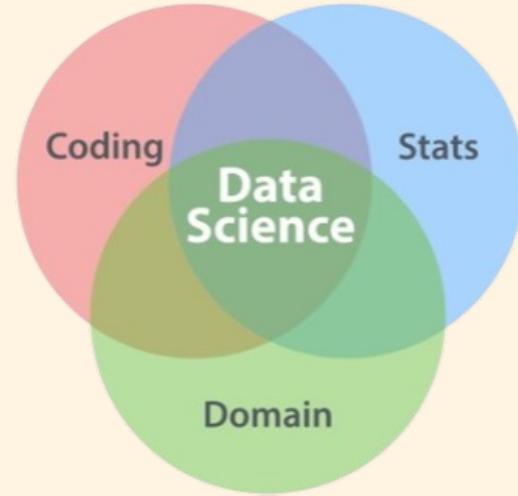
Si  $u = f(x)$  y  $v = g(x)$  entonces



# DS is the new DS

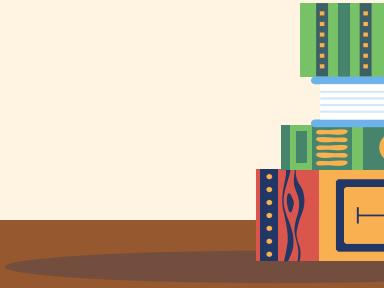
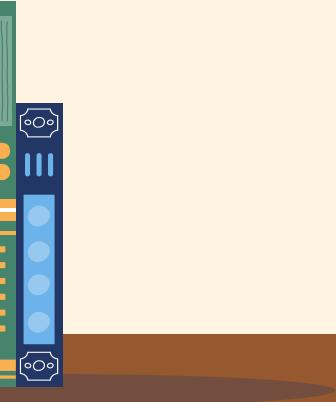


Data Storytelling



Data Science

# Nadie quiere más datos



# Data Storytelling

o

# Narrativa de Datos

**“Emplear datos, imágenes y narrativa para convencer a la audiencia y persuadirlos de actuar”**

# Ingrediente 1: Oratoria

- Particular
- Unidireccional
- Empatía

## **Importante:**

*Es una habilidad que normalmente no se enseña. Puede ser aprendida, pero requiere práctica y retroalimentación.*



# Ingrediente 2: Narrativa (Storytelling)

- Precisión
- Conflicto
- Sorpresa

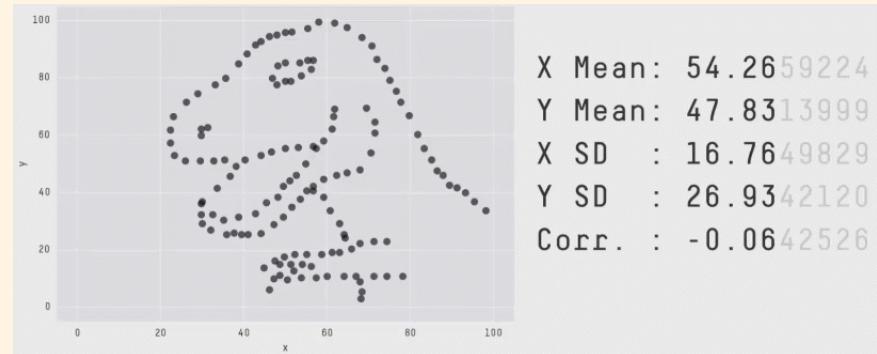
## Importante:

*Es una habilidad que normalmente no se enseña. Puede ser aprendida, pero requiere práctica y retroalimentación.*



# Ingrediente 3: Visualización de datos

- Comprender realmente
- Proceso incremental
- No es neutral



## Importante:

*Es una habilidad que normalmente no se enseña. Puede ser aprendida, pero requiere práctica y retroalimentación.*

# Mejor ejemplo de Data Storytelling:



Episodio

# 02 El código maldito



# Herramientas para presentación

- 01** PowerPoint (y equivalentes)
- 02** Prezi
- 03** Reveal.js
- 04** Jupyter Notebook + RISE
- 05** Streamlit

¡en Python!

¡en Python!



# Pero antes, un consejo...

**storytelling**

- historias
  - Cuentacuentos, historias de sus cuentos de maravillas → fantasía
  - Tintín - Heriberto - Simmler (figurativas) → fantasía
  - frases con estímulos → polémica actual
  - recordar la fermeza, fuerza-puntera
  - No importa que la historia sea cierta!
- brain works on stories, not data or logic
- claves históricas de la literatura? Historias (drama, comedia, romance)
- (fantasy & narrative)
- arte de storytelling → apunte de Matthew Dick.
  - closest to ordering, always in action
  - Don't split the ordering ("fragmented")

**Storytelling y visualización de datos**

- Storytelling
  - Cuentacuentos, historias de sus cuentos de maravillas → fantasía
  - Tintín - Heriberto - Simmler (figurativas) → fantasía
  - frases con estímulos → polémica actual
  - recordar la fermeza, fuerza-puntera
  - No importa que la historia sea cierta!
- brain works on stories, not data or logic
- claves históricas de la literatura? Historias (drama, comedia, romance)
- (fantasy & narrative)
- arte de storytelling → apunte de Matthew Dick.
  - closest to ordering, always in action
  - Don't split the ordering ("fragmented")

**Visualización**

- “Diseño de soft vs a mano”
- procedural
- Visualización → iterativa
- Viz → explícita
- explotar o animada

**librerías en C**

**librerías en R**

- librerías en JS
  - D3.js
  - vis.js
- librerías en Python
  - matplotlib lib
  - seaborn
  - altair
  - bokeh
  - viz3m
- maps:
  - ggplot lib
  - Folium

**oratoria**

- ideas en público es una forma de comunicación
- interactuar con las otras. Es importante una duración y intensidad constante de la charla
- Pausa dramática en una conversación = pausa, incomodidad, molestia
- importancia de lo retórico/matices para mejorar
- Frecuencia e buscar/pedir optimizaciones.
- (cuando algo que les apasiona, se hace mucho)

**Tipos de gráficos:**

- lineas
- barras
- ...
- ( todo tipo de Viz's )

**Reflexión**

- Nos enseñan soft, pero no a graficar sobre pensamientos.
- En aula no se educan delegados en los temas
- Hay una brecha en lo que el programa no cumple en.
- Sin motivación, no hay acci.

**Data Storytelling = logica + emoción**

- aprender narración a lo informativo, transformar datos en explicación de datos en algo interesante
- soft sin impacto
- Simplificar para datos

→ Presentar de Spotify

→ lógica

You have the power & reach  
in your tools! In your software!

# 01 PowerPoint (y equivalentes)

File Home Insert Design Transitions Animations Slide Show Review View Help

Paste New Reuse Slide Section

Clipboard Slides

Font Paragraph Drawing Editing Voice Designer

Find Replace Select

Share Comments

Dictate Design Ideas

Our Experiences

6 Our History

7 Our History

8 Our Philosophy

9

10 Our Experiences

Click to add notes

Slide 10 of 43 English (United States) Recovered

Notes

107%

# 02 Prezi



# 03 Reveal.js

The image shows a split-screen interface. On the left is a web browser window titled "Reveal Vite" displaying a blank slide with a light blue gradient background. On the right is a code editor window titled "index.html — reveal-vite" showing the following HTML code:

```
index.html > html > body > div.reveal > div.slides
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <link rel="icon" type="image/svg+xml" href="favicon.svg" />
6      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7      <title>Reveal Vite</title>
8    </head>
9    <body>
10      <div class="reveal">
11        <div class="slides"></div>
12      </div>
13      <script type="module" src="/main.js"></script>
14    </body>
15  </html>
16
```

The code editor has various icons on the right side, including file operations, search, and navigation. The status bar at the bottom of the code editor shows "main\* 0 0 0 7 html | index.html Formatting: ✓".

# 04 Jupyter Notebook + RISE

The screenshot shows a Jupyter Notebook interface with the RISE theme applied. The top navigation bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Logout, Trusted, and Python 3. The main area displays a presentation slide deck:

- Slide 1:** **0.4 Sobre hacer presentaciones con jupyter y RISE**  
Ok, pero... ¿porqué? ¿Porqué? ¿PORQUÉ?
  - Tener una presentación autocontenida: los ejemplos (código) queda en jupyter notebook.
  - Corregir directamente los errores cuando sea necesario.
  - Hacer participar a la audiencia.
- Slide 2:** **0.4 Sobre hacer presentaciones con jupyter y RISE**  
SI, incluso ejecutar código en python:

```
In [ ]:  
x = 200  
y = 200  
x is y
```

```
In [ ]:  
a = 300  
b = 300  
a is b
```

```
In [ ]:  
def cheeseburger(n):  
    "return ['bread'] + ['meat', 'cheese"]*n + ['bread']"  
cheeseburger("uno")
```
- Slide 3:** **0.5 Sobre hacer presentaciones con jupyter y RISE**  
¿Qué he aprendido?
  - Usa "Ctrl +\*" y "Ctrl -\*" para controlar aspecto.
  - Entrega parte del control de contenido al público. Usa un teclado+mouse inalámbrico.
  - Plantea desafíos de código incompleto pero entrega propuestas de solución.

pip install jupyter notebook rise

# 05 Streamlit

This is slide 1

Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by their place and supplies it with the necessary regelialia. It is a paradisematic country, in which roasted parts of sentences fly into your mouth. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas, wild Question Marks and devious Semikoli, but the Little Blind Text didn't listen. She packed her seven versalia, put her initial into the belt and made herself on the way. When she reached the first hills of the Italic Mountains, she had a last view back on the skyline of her hometown Bookmarksgrove, the headline of Alphabet Village and the subline of her own road, the Line Lane. Pityful a rhetoric question ran over her cheek, then she continued her way. On her way she met a copy. The copy warned the Little Blind Text, that where it came from it would have been rewritten a thousand times and everything that was left from its origin would be the word 'and' and the Little Blind Text should turn around and return to its own, safe country. But nothing the copy said could convince her and so it didn't take long until a few insidious Copy Writers ambushed her, made her drunk with Longe and Parole and dragged her into their agency, where they abused her for their projects again and again. And if she hasn't been rewritten, then they are still using her. Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by their place and supplies it with the necessary regelialia. It is a paradisematic country, in which roasted parts of sentences fly into your mouth. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem Ipsum decided to

pip install streamlit streamlit-slides

# Herramientas para visualización

¡en Python!

**01** matplotlib

**02** plotly

**03** seaborn

**04** plotnine (ggplot)

# Herramientas para visualización

¡en Python!

**01** matplotlib

**02** plotly

**03** seaborn

**04** plotnine (ggplot)

**05** pygal

**06** bokeh

**07** altair

**08** vizzu



# Setup para una comparación

```
# Import libraries
import pandas as pd

# Read the data
df = pd.read_excel('data/dataset.xlsx')
```

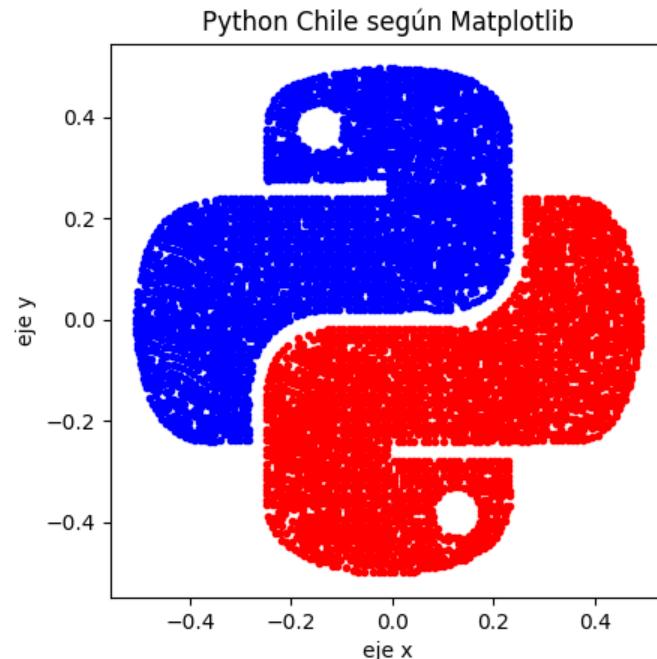
Graficar los 2800 puntos del conjunto A en rojo.

Graficar los 2800 puntos del conjunto B en azul.

group	x	y
A	0.061966	-0.33708
B	-0.0201	0.283512
A	-0.03784	-0.20666
B	-0.3245	0.042072
A	-0.20751	-0.16472
B	-0.19476	0.051977
A	0.412396	-0.04484
B	0.194486	0.078636
A	0.395207	0.17246
B	-0.05226	0.121314
A	0.172862	-0.07205
B	-0.18034	0.160935
A	0.335324	0.210461
B	-0.28181	0.002451
A	-0.15761	-0.18939
B	-0.18589	0.444304
A	0.033133	-0.05808
B	-0.32839	0.091598
A	0.042005	-0.36679

# 01 matplotlib

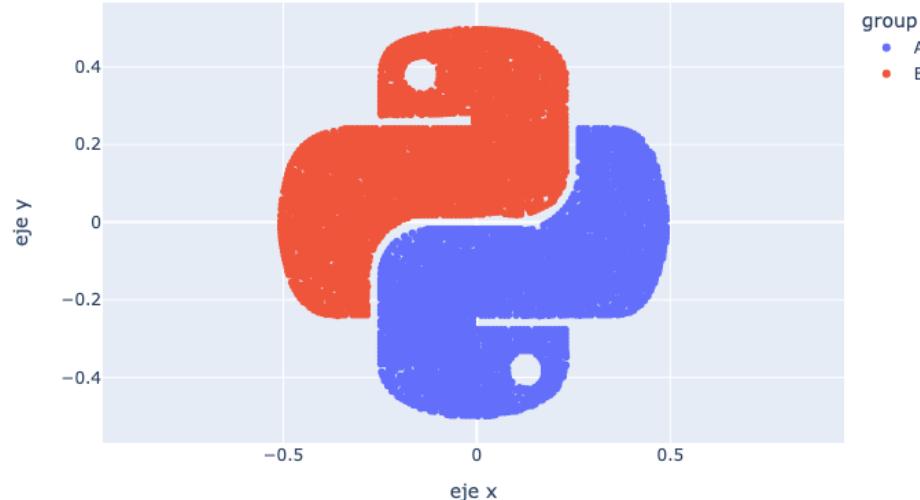
```
from matplotlib import pyplot as plt
plt.plot(df.x[df.group=="A"], df.y[df.group=="A"], ".r")
plt.plot(df.x[df.group=="B"], df.y[df.group=="B"], ".b")
plt.xlabel("eje x")
plt.ylabel("eje y")
plt.title("Python Chile según Matplotlib")
plt.axis('square')
```



# 02 plotly

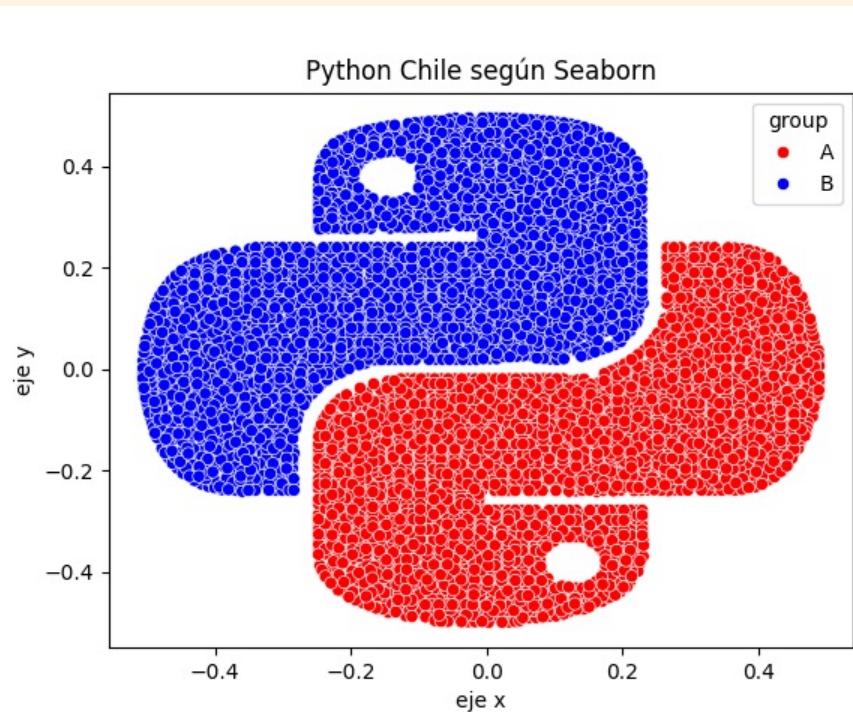
```
import plotly.express as px
fig = px.scatter(df, x="x", y="y", color='group',
                  labels={"x": "eje x", "y": "eje y"},
                  title="Python Chile según Plotly")
fig.update_yaxes(scaleanchor="x", scaleratio=1)
```

Python Chile según Plotly



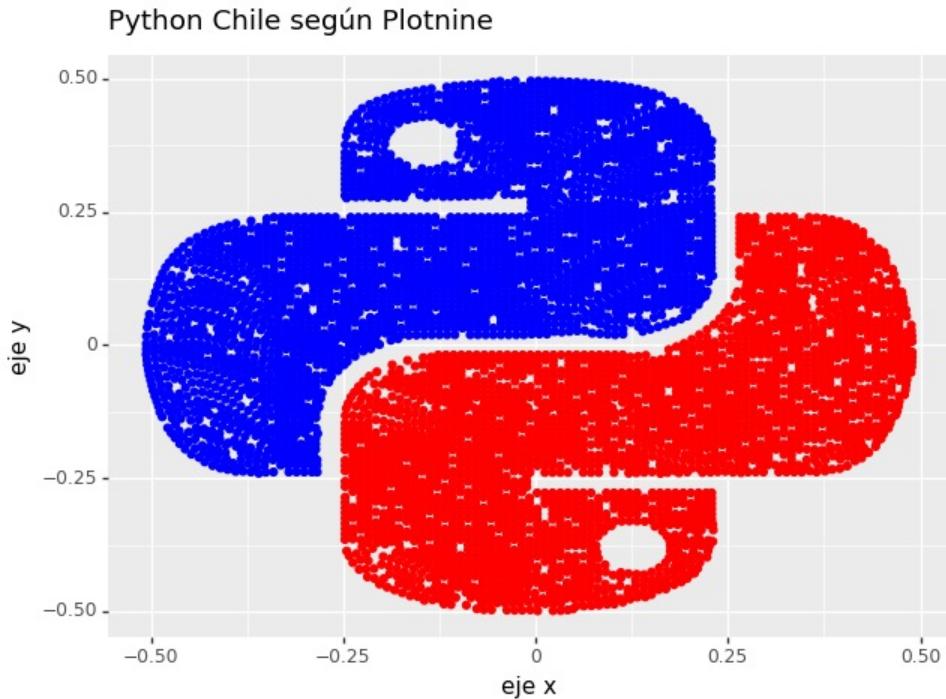
# 03 seaborn

```
import seaborn as sns
sns.scatterplot(x = "x",
                 y = "y",
                 data=df, hue="group",
                 palette=['red','blue']).\
set(title='Python Chile según Seaborn',
     xlabel='eje x',
     ylabel='eje y')
```



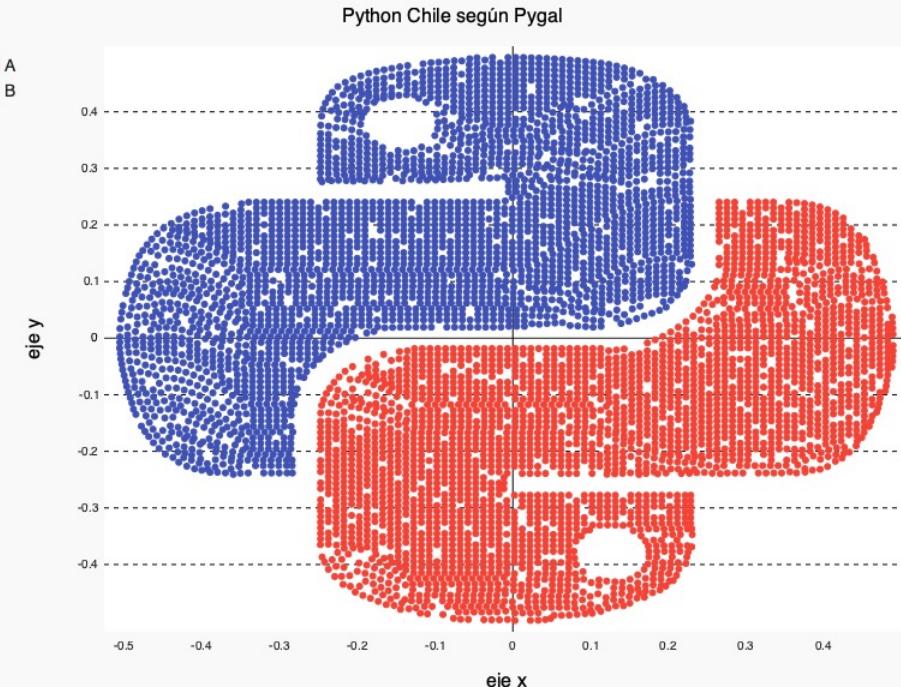
# 04 plotnine

```
import plotnine as p9
my_plot = (
    p9.ggplot(df)
    + p9.aes(x='x', y='y')
    + p9.geom_point(data=df[df.group=="A"], color='red')
    + p9.geom_point(data=df[df.group=="B"], color='blue')
    + p9.labs(title='Python Chile según Plotnine',
              x='eje x',
              y='eje y')
)
```



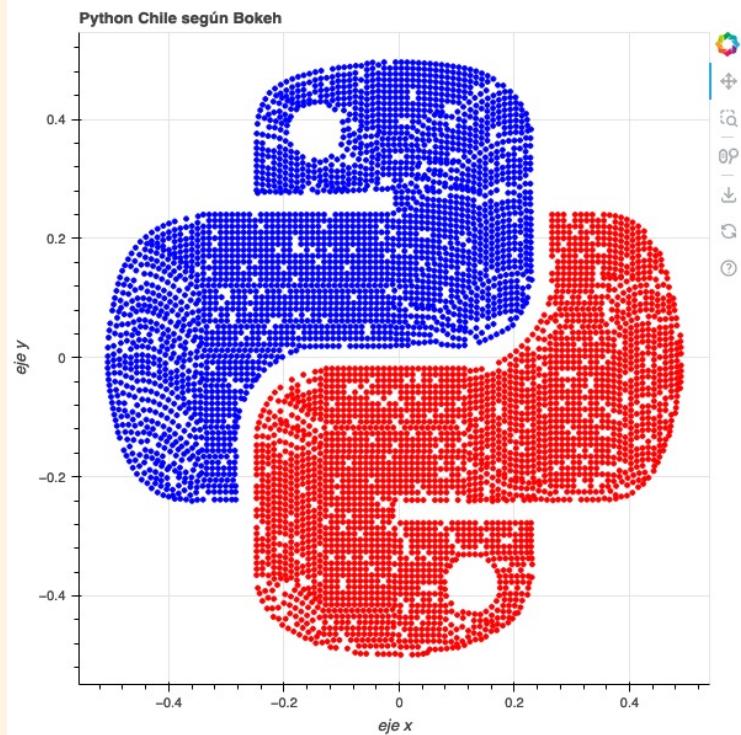
# 05 pygal

```
import pygal
xy_chart = pygal.XY()
data_A = df.loc[df.group=="A", ['x','y']].values
xy_chart.add('A', data_A, stroke=False, color='red')
data_B = df.loc[df.group=="B", ['x','y']].values
xy_chart.add('B', data_B, stroke=False, color='blue')
xy_chart.x_title = 'eje x'
xy_chart.y_title = 'eje y'
xy_chart.title = 'Python Chile según Pygal'
```



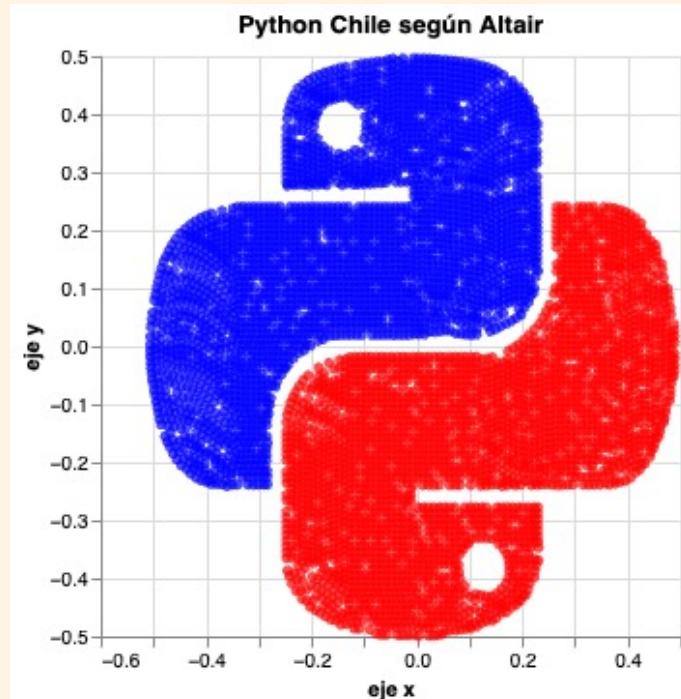
# 06 bokeh

```
import bokeh.plotting as bk
p = bk.figure(title="Python Chile según Bokeh",
               x_axis_label="eje x",
               y_axis_label="eje y")
m_A = df.group=="A"
p.circle(df.x[m_A], df.y[m_A], color="red")
m_B = df.group=="B"
p.circle(df.x[m_B], df.y[m_B], color="blue")
```



# 07 altair

```
import altair as alt
title = alt.TitleParams("Python Chile según Altair")
chart_1 = alt.Chart(df[df.group=="A"], title=title).\
    mark_circle().\
    encode(
        x=alt.X('x', title='eje x'),
        y=alt.Y('y', title='eje y'),
        color=alt.value('red')
    )
chart_2 = alt.Chart(df[df.group=="B"]).\
    mark_circle().\
    encode(
        x=alt.X('x'),
        y=alt.Y('y'),
        color=alt.value('blue')
    )
chart = chart_1 + chart_2
```



# 08 vizzu

```
data = Data()
data.add_df(df)

story = Story(data)
story.set_size("800px", "800px")
my_style = Style({
    "plot": {
        "marker": {"colorPalette": "#ff0000 #0000ff"}
    }
})

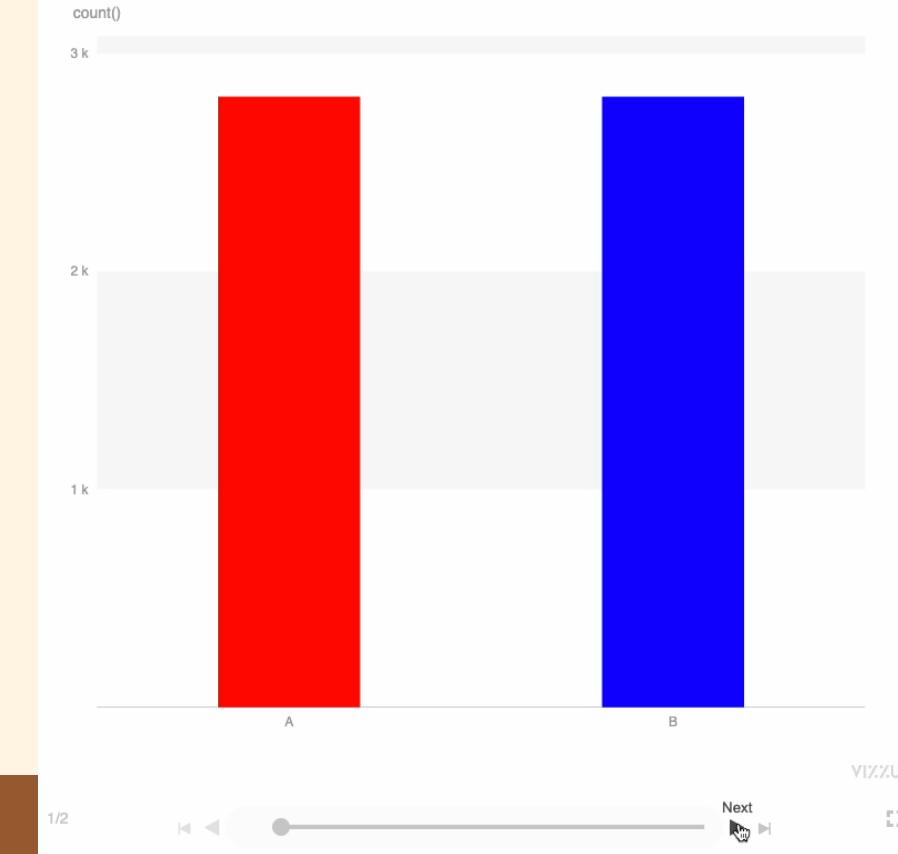
slide = Slide(Step(Config({
    "x": "group",
    "y": "count()", 
    "color": "group",
}), 
my_style))

story.add_slide(slide)

slide = Slide(Step(Config({
    "x": "x",
    "y": "y",
    "color": "aux_index",
    "title": "Python Chile según Vizzu",
    "geometry": "circle",
}), 
))

story.add_slide(slide)

story.play()
```

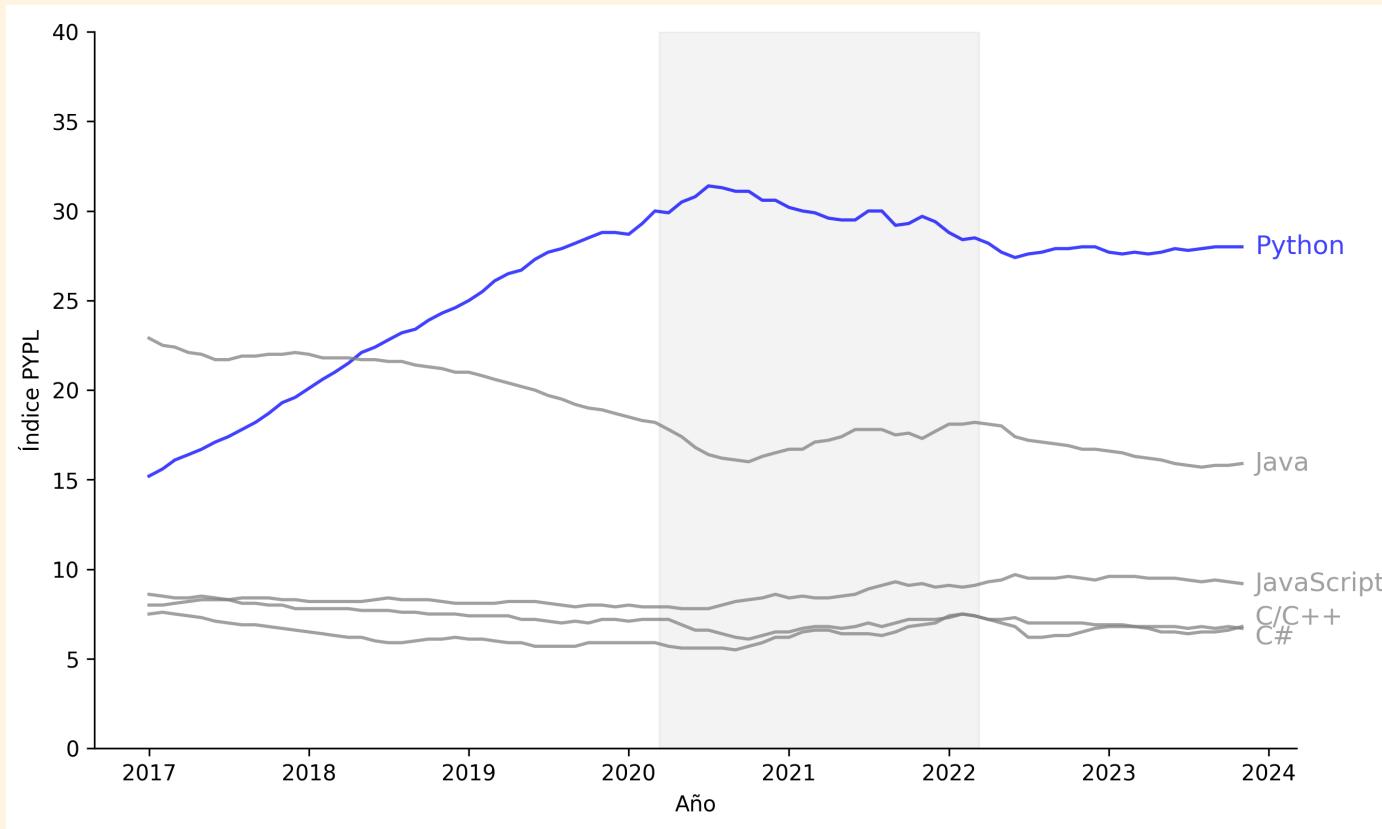


Episodio

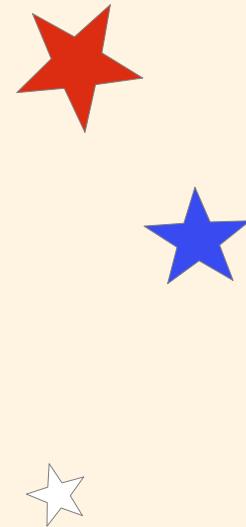
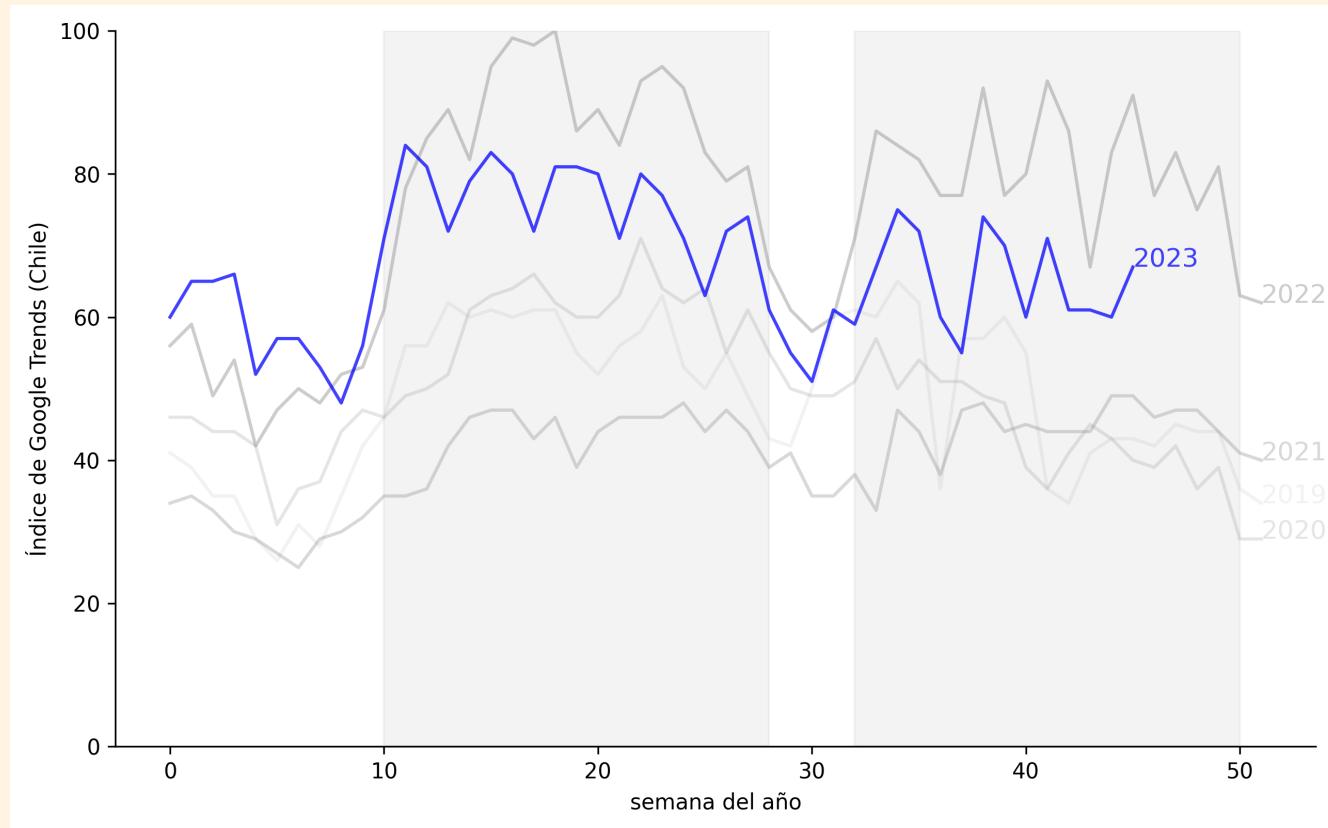
# 03 La última mirada



# ¿Vale la pena aprender Python y Data Storytelling?



# ¿Vale la pena aprender Python y Data Storytelling?



# ¿Vale la pena aprender Python y Data Storytelling?

Por cada **1000**  
páginas sobre “Python” (*en google*)

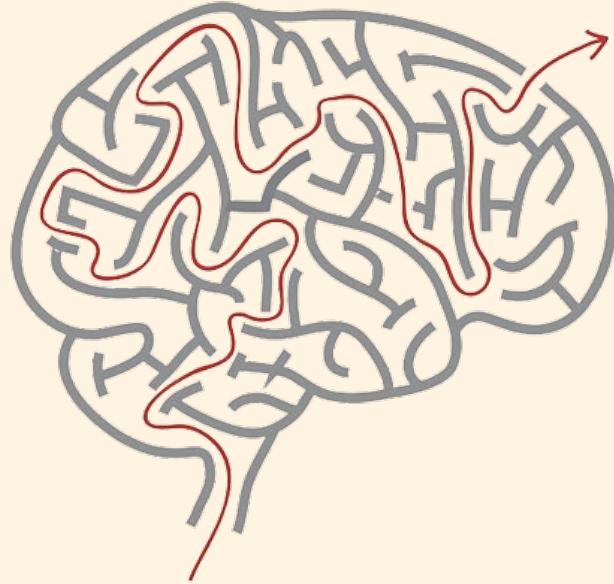
hay **1**

página sobre “Data Storytelling”

# Conclusión



# Data Storytelling o Narrativa de Datos



Funciona porque las historias son el lenguaje natural del cerebro

**Programación en Python:**  
Habilidad común y transferible a varias industrias.  
En transformación por avances de Inteligencia Artificial.

**Data Storytelling:**  
Habilidad **poco usual** y transferible a varias industrias.  
**Indiferente** a avances de Inteligencia Artificial.

**Python + Data Storytelling = Combo Ganador**

# Recursos recomendados

## Storytelling

- *Storyworthy*, de Matthew Dicks
- *Made to Stick*, de Chip Heath y Dan Heath

## Data Storytelling

- *Storytelling with Data*, de Cole Nussbaumer Knaflic
- *Data Visualization & Storytelling*, de Jose Berengueres y Marybeth Sandell

# ¡Gracias!

**CREDITS:** This presentation template was created by [Slidesgo](#), and includes icons by [Flaticon](#), and infographics & images by [Freepik](#)

Please keep this slide for attribution



Bits, pixeles y palabras:

# ¿Cómo hacer data storytelling con Python?

Sebastián Flores  
PyCon Chile 2023  
25 Noviembre 2023

## ¿Preguntas?



