Visualizaciones interactivas con Plotly

Celia López | Customer Success Engineer

19 de Mayo de 2022

Copyright © 2022 Plotly. All rights reserved.

¿Qué es Plotly?

Plotly (empresa) ofrece:

- **Plotly:** librería de visualización de datos para varios lenguajes (Python, R, Julia)
- **Dash Open Source:** librería de Python y R para construir apps web interactivas sin necesidad de saber HTML, CSS o JS.
- **Dash Enterprise:** Dash para empresas, con más funcionalidades y servicios.

4M+ descargas mensuales de Plotly

300k+ descargas mensuales de Dash

Estructura de la sesión

1. Qué es Plotly y por qué podría interesarme.
2. Cómo se crean y modifican figuras con Plotly.
3. Cómo seguir aprendiendo plotly y resolver dudas/problemas.

plotly

Copyright © 2022 Plotly. All rights reserved.

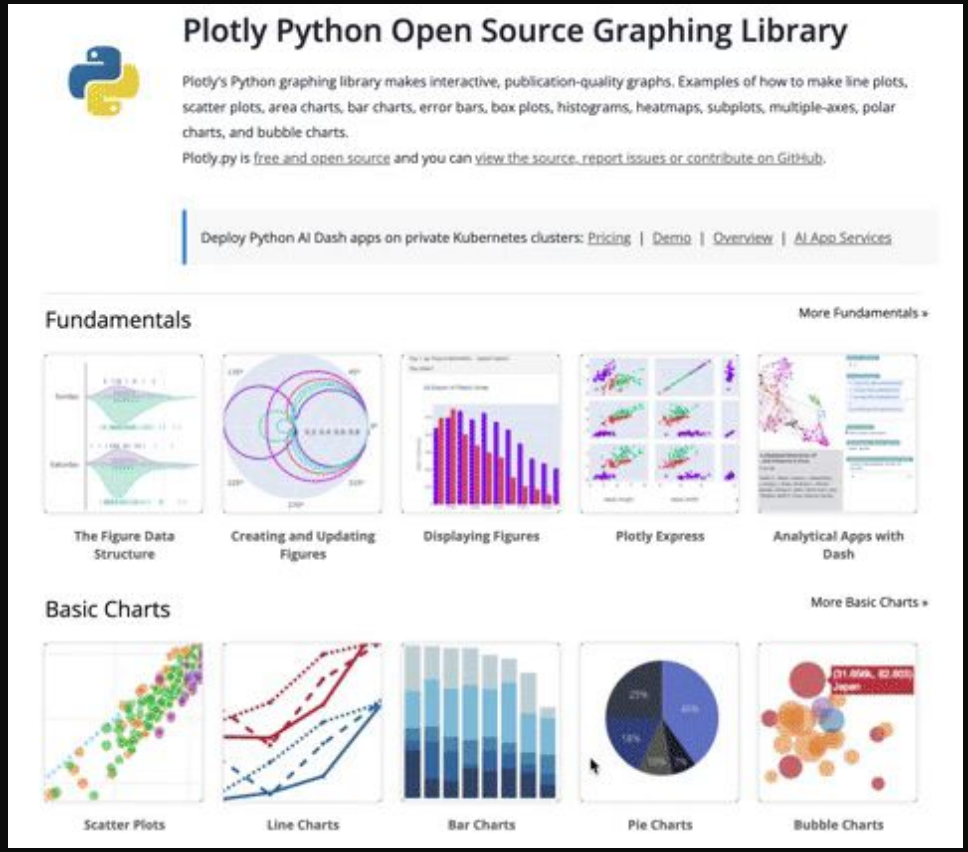
¿Qué ofrece Plotly?

Sus características:

- **Gráficos interactivo**
- Versatilidad
- Fácil de aprender

+100 tipos de gráficos:

- Gráficos 3D
- Mapas
- Subplots con gráficos de diferentes tipos



plotly

Copyright © 2022 Plotly. All rights reserved.

Crear y modificar figuras con Plotly

Los módulos de Plotly más utilizados son:

- **plotly.express (px):** crear gráficos de forma simple y rápida.
- **plotly.graph_objects (go):** crear y modificar gráficos de forma más elaborada.

```
!pip install plotly

import plotly.express as px
import plotly.graph_objects as go
import plotly.io as pio
from plotly.subplots import make_subplots
```

Los gráficos creados con **px** y con **go:**

- Tienen la misma estructura interna (son la misma clase de objeto).
- Se pueden modificar de las mismas formas: **`.update_layout()`**, **`.add_traces()`**
- Se pueden combinar. De hecho, lo más habitual es crear una figura con **`px`** y modificarla con **`go`**.

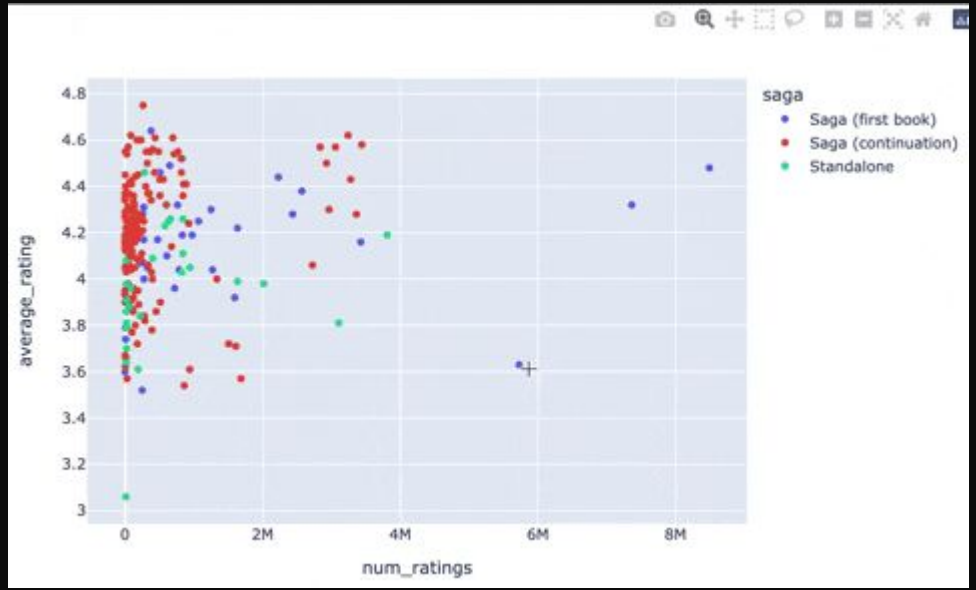
plotly

Copyright © 2022 Plotly. All rights reserved.

Crear gráficos: Scatter plot interactivo

```
import plotly.express as px
fig_scatter = px.scatter(
   books_df_scatter, x="num_ratings", y="average_rating", color="saga",
   hover_data=["title", "series_name", "series_n"])
fig_scatter.show()
```

- Información adicional del punto al situarse sobre él (**hover**).
- Menú con opciones: **descarga**, zoom, selección.
- Zoom en **región seleccionada**.
- **Reset de zoom** al hacer doble click o seleccionar 'reset axes'



plotly

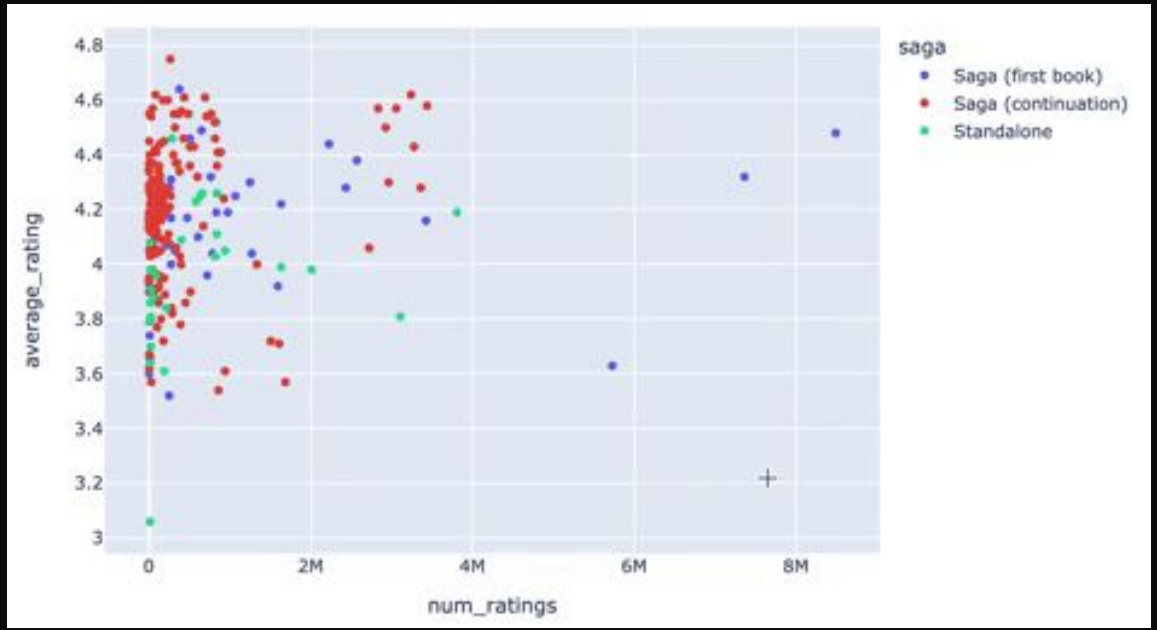
Copyright © 2022 Plotly. All rights reserved.

Crear gráficos: Leyenda interactiva + Hover data

```
fig_scatter = px.scatter(
   books_df_scatter, x="num_ratings", y="average_rating", color="saga",
   hover_data=["title", "series_name", "series_n"],
   )
fig_scatter.show()
```

- **Click** = Activar/desactivar grupo.
- **Doble click** = Aislar grupo.

- `hover_data` permite incluir información adicional del df que no aparece en otros ejes (x, y, color).



plotly

Copyright © 2022 Plotly. All rights reserved.

Crear gráficos: Estructura interna

Diccionarios anidados con dos elementos principales:

- `data` (lista de *traces*)
  - Datos y valores asociados.
  - Se modifica con `.update_traces()`
- `layout`
  - Aspecto.
  - Se modifica con `.update_layout()`

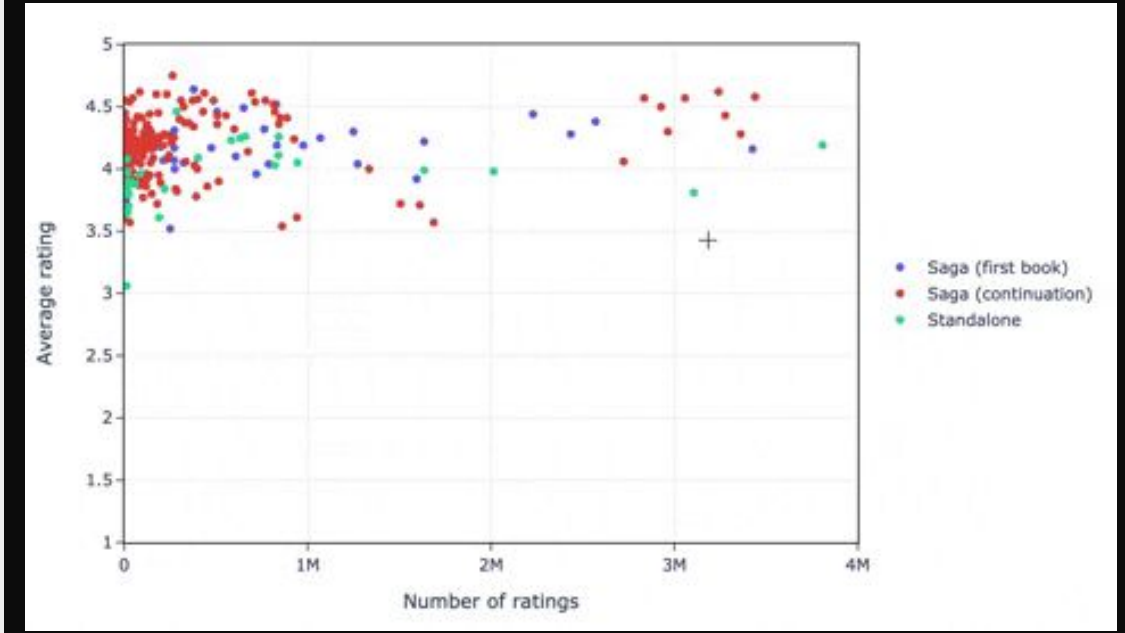
```
<class 'plotly.graph_objs._figure.Figure'>

Figure({
    'data': [{'hovertemplate': 'num_ratings=%{x}<br>average_rating=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'symbol': 'circle'},
              'mode': 'markers',
              'name': '',
              'orientation': 'v',
              'showlegend': False,
              'type': 'scatter',
              'x': array([8491079, 3277548, ...,   15518,  285067,   16896]),
              'xaxis': 'x',
              'y': array([4.48, 4.43, 4.58, ..., 3.64, 4.46, 3.86]),
              'yaxis': 'y'}],
   'layout': {'legend': {'tracegroupgap': 0},
              'margin': {'t': 60},
              'template': '...',
              'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0],
                        'title': {'text': 'num_ratings'}},
              'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0],
                        'title': {'text': 'average_rating'}}}
})
```

plotly

Copyright © 2022 Plotly. All rights reserved.

Modificar gráficos: layout y data/traces

```
# Cambiar aspecto
# no es necesario reasignarlo al objeto
fig_scatter.update_layout(
    template = 'plotly_white+borders',
    xaxis = {'title':'Number of ratings',
             'range':[0, 4000000]},
    yaxis = {'title':'Average rating',
             'range':[1,5]},
    legend = {'title': None,
              'x': 1.02,
              'y': 0.5}
)
# Especificar plantilla del texto de hover
fig_scatter.update_traces(
    hovertemplate = '<b>%{customdata[0]}</b>
<br>%{customdata[1]} %{customdata[2]}'
)

fig_scatter.show()
```

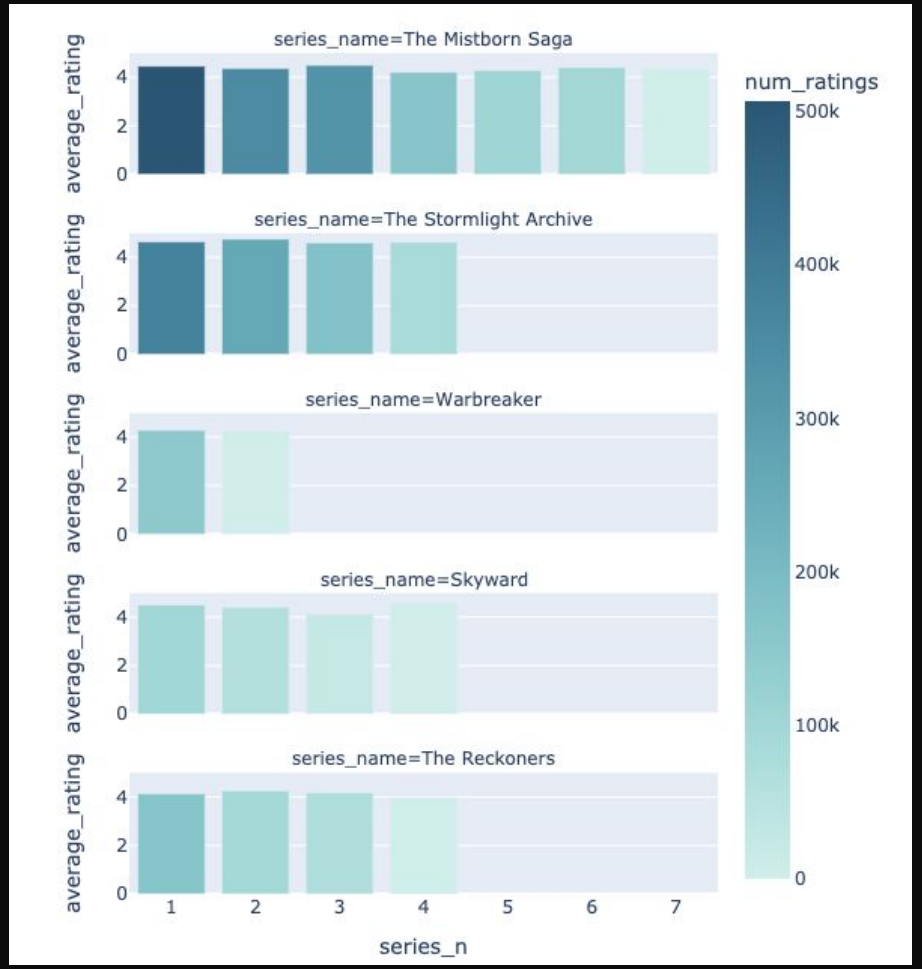


plotly

Copyright © 2022 Plotly. All rights reserved.

Crear gráficos: Gráficos múltiples

```
fig_sanderson = px.bar(
    sanderson_sagas_df,
    x='series_n', y='average_rating',
    # Generar un gráfico para cada valor de 'series_name'
    facet_col='series_name', facet_col_wrap=1,
    # Especificar var. continua para color y su paleta
    color = 'num_ratings',
    color_continuous_scale = px.colors.sequential.Teal,
    hover_data = ['title'],
    # Especificar anchura y altura
    width = 600, height = 700
)

fig_sanderson.show()
```

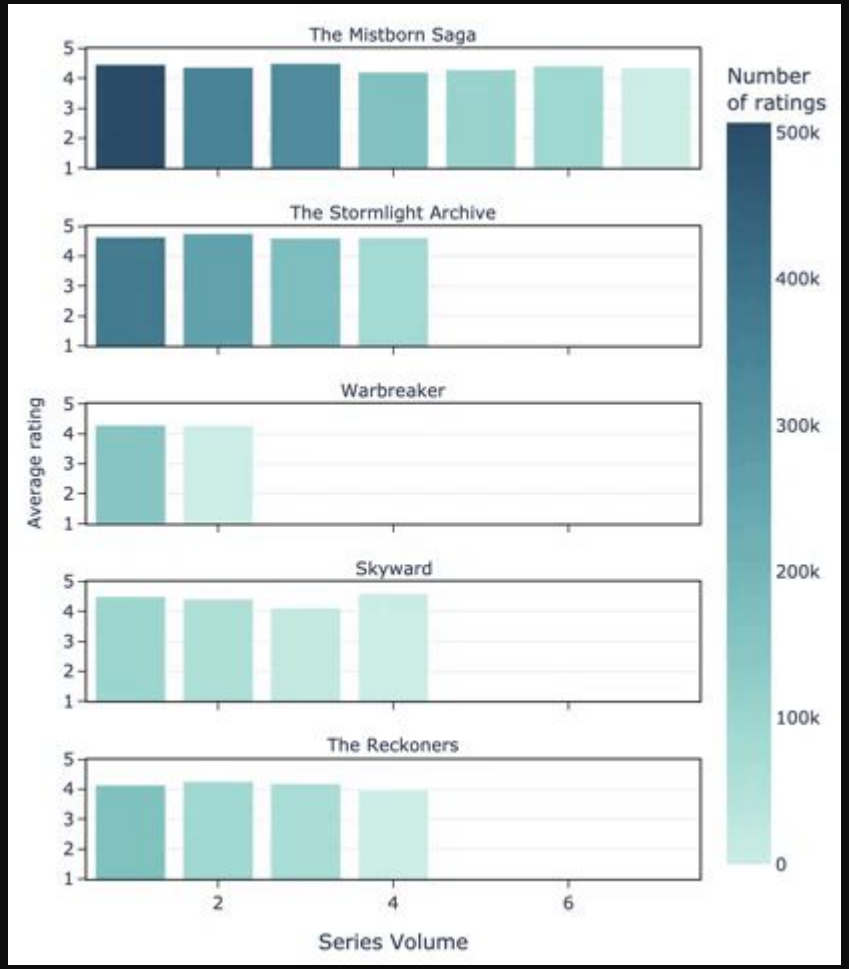


plotly

Copyright © 2022 Plotly. All rights reserved.

Modificar gráficos: Gráficos múltiples

```
# Cambiar estilo
fig_sanderson.update_layout(
   coloraxis_colorbar = {'title':'Number<br>of ratings'},
   xaxis = {'title':'Series Volume'},
   yaxis = {'range':[1,5]},
   template = 'plotly_white+borders'
  )

# Quitar las etiquetas delante del =
fig_sanderson.for_each_annotation(
   lambda a: a.update(text=a.text.split("=")[-1])
   )

# Quitar el título del eje Y de cada subplot
fig_sanderson.for_each_yaxis(
   lambda y: y.update(title=None)
  )

# Crear un título general/compartido para el eje Y
fig_sanderson.add_annotation(
   text='Average rating',
   x=-0.11, y=0.5,
   xref="paper", yref="paper",
   textangle=-90, showarrow=False
   )
```

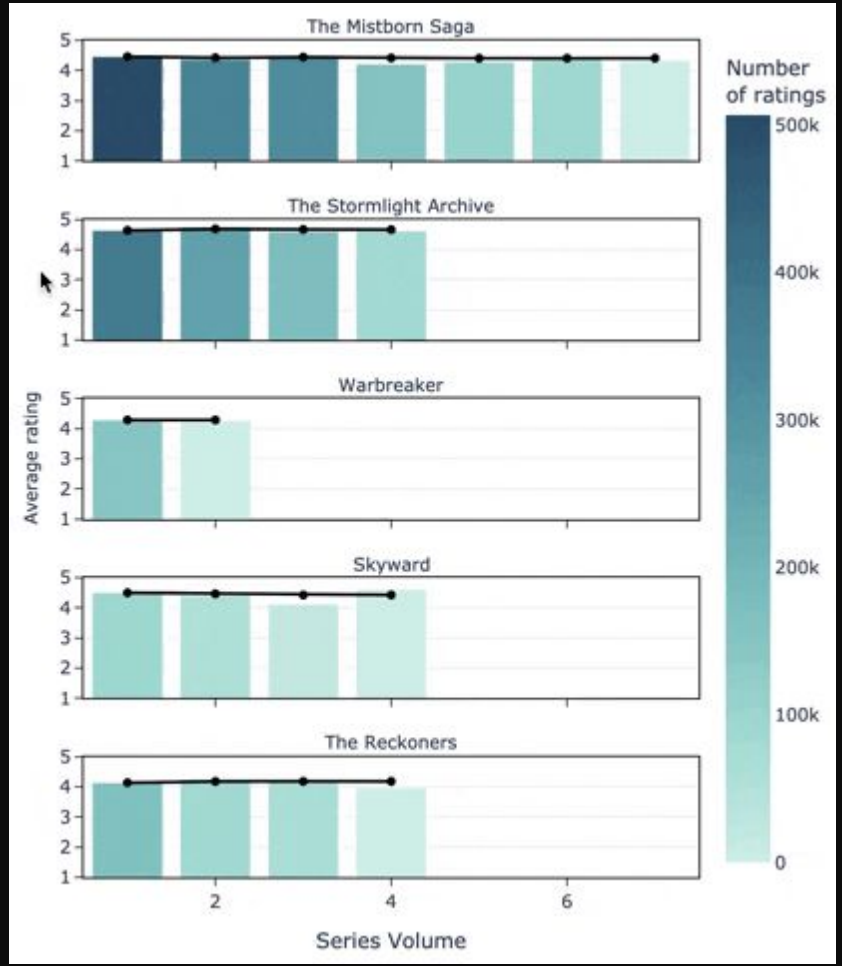


plotly

Copyright © 2022 Plotly. All rights reserved.

Modificar gráficos: Añadir traces

```
sanderson_sagas = sanderson_sagas_df.series_name.unique()

# Usamos `[::-1]` para revertir el orden de
sanderson_sagas porque en los facet plots, las filas
(row) empiezan a contar desde abajo (y desde 1, no 0)

for i, saga in enumerate(sanderson_sagas[::-1]):
    df_i=sanderson_sagas_df.query(f'series_name=="{saga}"')
    fig_sanderson.add_trace(
        go.Scatter(
            x=df_i['series_n'], y=df_i['hist_rating'],
            mode='lines+markers', line_color='black',
            showlegend=False, hoverinfo='none'
            ),
        row=i+1, col=1)

fig_sanderson.show()
```



plotly

Copyright © 2022 Plotly. All rights reserved.

Seguir aprendiendo: Dónde buscar

Google

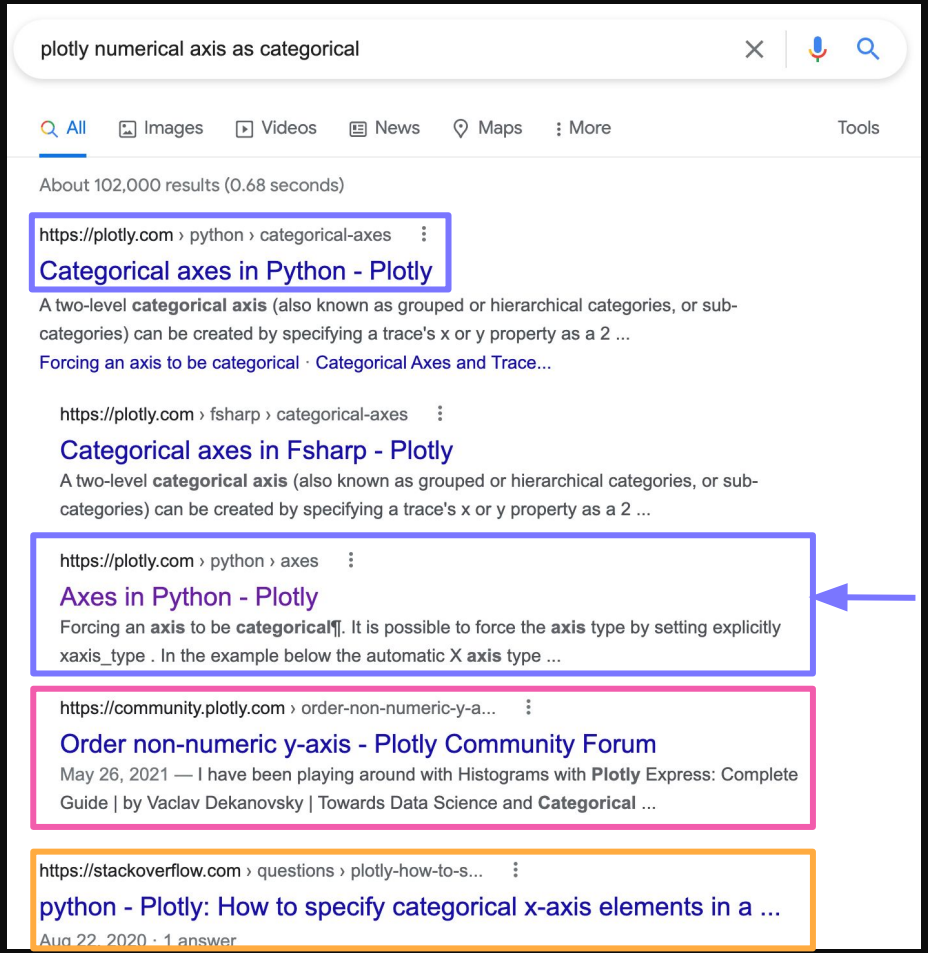
Documentación de Plotly para Python

- Ejemplos de gráficos con distintos aspectos y funcionalidades
- Referencia completa de todos los argumentos

Foro oficial de Plotly

Stackoverflow

Issues de GitHub



plotly

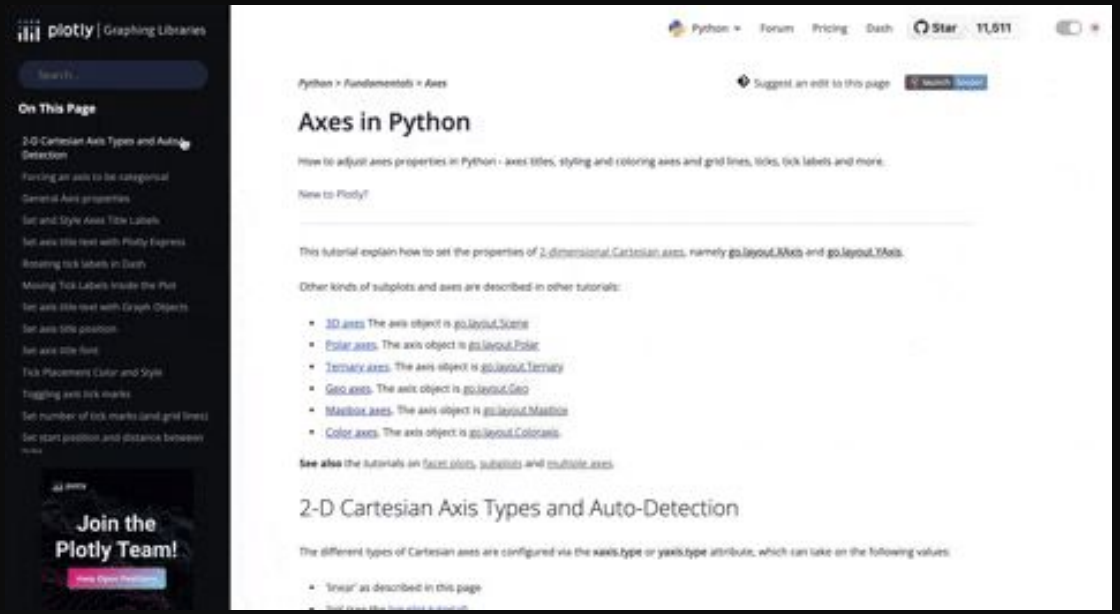
Copyright © 2022 Plotly. All rights reserved.

Seguir aprendiendo: Cómo usar la documentación

La forma más sencilla de aprender a hacer algo en Plotly es:

1. Buscar un ejemplo de gráfico con una funcionalidad/aspecto similar. Podemos encontrarlo en la web de **Plotly**, el **Foro** o **Stackoverflow**.
2. Acudir a la documentación completa (***Reference****) de esa función para adaptar los detalles a nuestro caso.
3. Preguntar en el **Foro** si seguimos sin encontrar respuesta.

¡! Incluir un MRE (*Minimal Reproducible Example*)



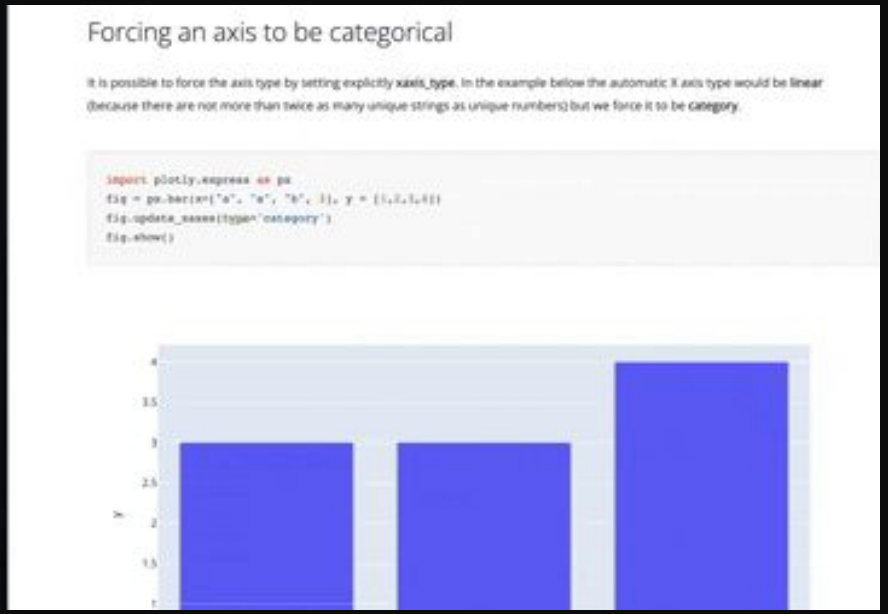

```
import plotly.express as px
fig = px.bar(x=["a", "a", "b", 3], y = [1,2,3,4])
fig.update_xaxes(type='category')

fig.show()
```

plotly

Copyright © 2022 Plotly. All rights reserved.

Seguir aprendiendo: Cómo usar la documentación

- El apartado ***Reference*** siempre está al final de la página donde están los casos de uso relacionados.
- Respecto a `.update_layout()` y sus métodos relacionados, pueden especificarse de dos formas:
  - Diccionarios anidados, con `dict()` o `{}`
  - Encadenando los nombres con underscores (_).




```
fig.update_xaxes(type='category')
```

```
fig.update_layout(xaxis = {'type':'category'})
```

plotly

Copyright © 2022 Plotly. All rights reserved.

Resumen

1. Crea un gráfico con `fig = px.Scatter` (o fig=px.Bar, fig=px.Bar, etc.)
2. Modifícalo con `fig.update_layout()` y `fig.update_traces()`.
3. Añade elementos con `fig.add_trace()`, `fig.add_annotation()`, `fig.add_hline()`, `fig.add_shape()`...
4. A interactuar!

Tips para seguir avanzando y resolver problemas

Alguien habrá tenido antes la misma pregunta que tú tienes ahora:

1. Busca en Google, la Documentación o el Foro un gráfico parecido (usa **palabras clave**).
2. Una vez hayas identificado el código clave (función/método) consulta la ***Referencia*** para adaptarlo a tu caso.

Si ya dominas Plotly para Python: Dash Open Source : https://dash.plotly.com/

plotly

Copyright © 2022 Plotly. All rights reserved.

Recursos útiles

Documentación de Plotly para Python: https://plotly.com/python/

Cómo usar la documentación de referencia: https://plotly.com/python/reference/index/

Foro oficial: https://community.plotly.com/c/plotly-python/

Qué es un MRE y cómo incluirlo en tus preguntas:
https://community.plotly.com/t/how-to-write-a-minimal-reproducible-example-mre/61502

Charming Data - Canal de tutoriales de Plotly y Dash Open Source:
https://www.youtube.com/channel/UCqBFsuAz41sqWcFjZkqmJqQ

GitHub de plotly.py: https://github.com/plotly/plotly.py/

Si ya dominas Plotly para Python: Dash Open Source : https://dash.plotly.com/

plotly

Copyright © 2022 Plotly. All rights reserved.

Thank You!

For more information about Plotly and Dash reach out to us at **info@plotly.com**!

: @plotly : @plotlygraphs

plotly

Copyright © 2022 Plotly. All rights reserved.