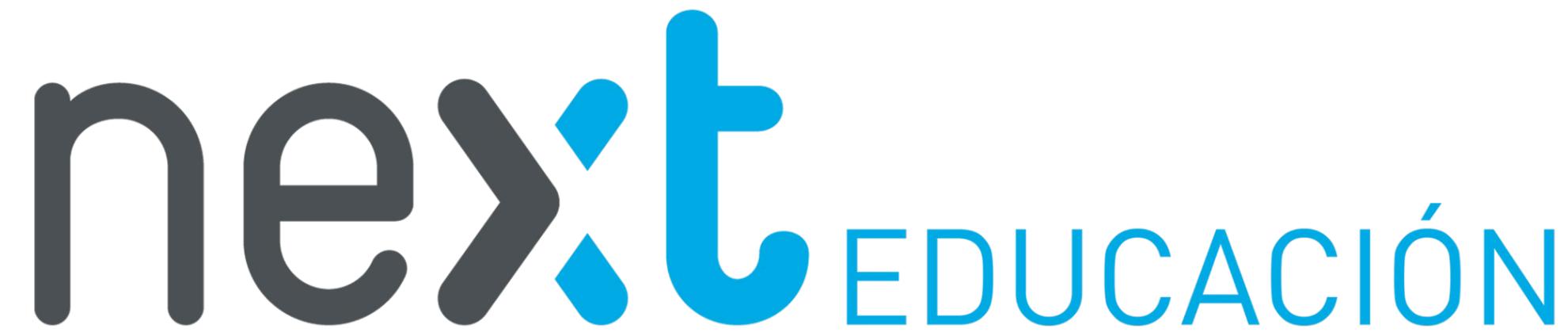




next EDUCACIÓN



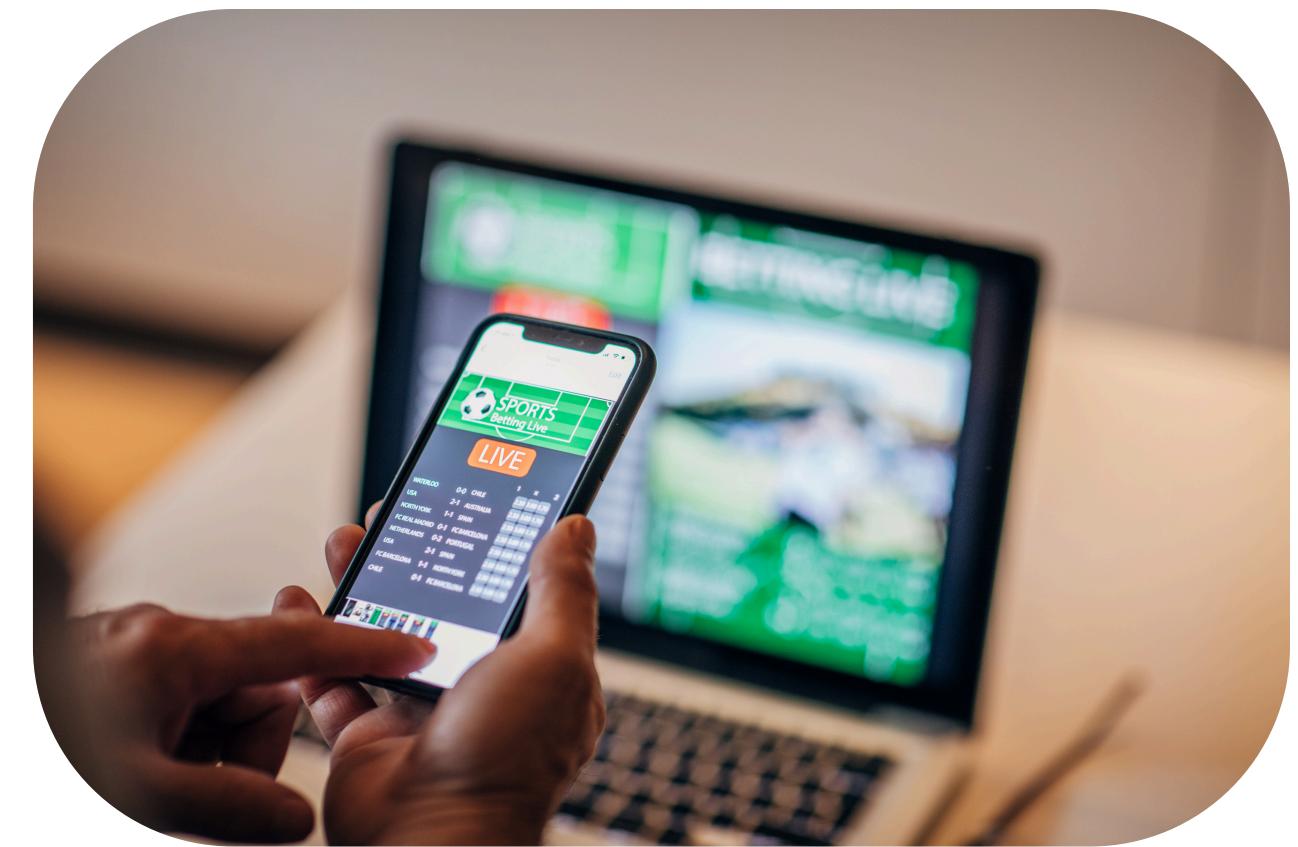
Streamlit

Docente: Isidre Munné-Bertran

Introduction to Streamlit

What is Streamlit?

- Streamlit is an open-source Python library that converts data scripts into shareable web apps with minimal code.
- It allows data scientists to build **interactive applications** in pure Python without needing front-end skills.
- Other frameworks: Dash (complex UI interactions & multi-page apps), Flask (to build an API or backend for web app), Django (full-stack)



1

Rapid Prototyping

2

Python-Centric

3

Growing Ecosystem



Steps

Install Streamlit

Run **pip install streamlit** in your terminal to install the package.

Create Your First App

Create a Python file and add your Streamlit code.

Run Your App

Execute **streamlit run main.py** to launch your web application.

Streamlit Basics

Basic Syntax

```
1 import streamlit as st  
2  
3 st.write('Hello, world!') # Basic body text  
4 hello = st.write('Hello, world (as variable)!')  
5 print(hello) # Display the variable hello
```



Hello, world!

Hello, world (as variable)!

Text Display

Streamlit provides multiple ways to display text, including **titles**, **headers**, **markdown**, and **LaTeX**.

```
11  ### Text Display ###
12  st.title('Text Display Basics') # title
13  st.header('Header') # Larger text
14  st.subheader('Subheader') # Smaller text
15  st.code('print("Hello, world!")') # Code block
16  st.markdown('**Markdown**') # Markdown
17  st.latex(r'\int_a^b f(x) dx') # LaTeX
18  st.write("Basic Text") # Horizontal rule
```

Text Display Basics

Header

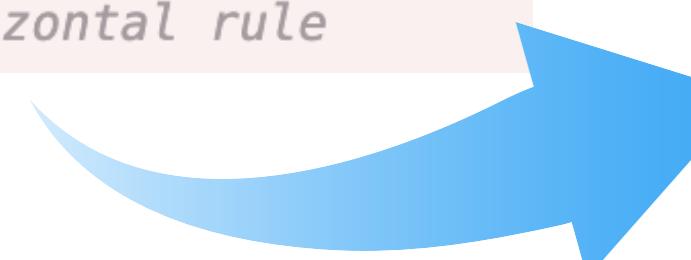
Subheader

```
print("Hello, world!")
```

Markdown

$$\int_a^b f(x) dx$$

Basic Text



Data Display

Streamlit makes it easy to display tables and structured data from Pandas and JSON.

```
20  ### Data Display ###
21  st.title('Data Display Basics') # title
22
23  # DataFrames (interactive)
24  df = pd.DataFrame({
25      'first column': [1, 2, 3, 4],
26      'second column': [10, 20, 30, 40]
27  })
28  st.dataframe(df)
29
30  # DataFrames (non-interactive)
31  st.table(df)
32
33  # JSON
34  st.json({
35      'first column': [1, 2, 3, 4],
36      'second column': [10, 20, 30, 40]
37  })
```

Data Display Basics

	first column	second column
0	1	10
1	2	20
2	3	30
3	4	40

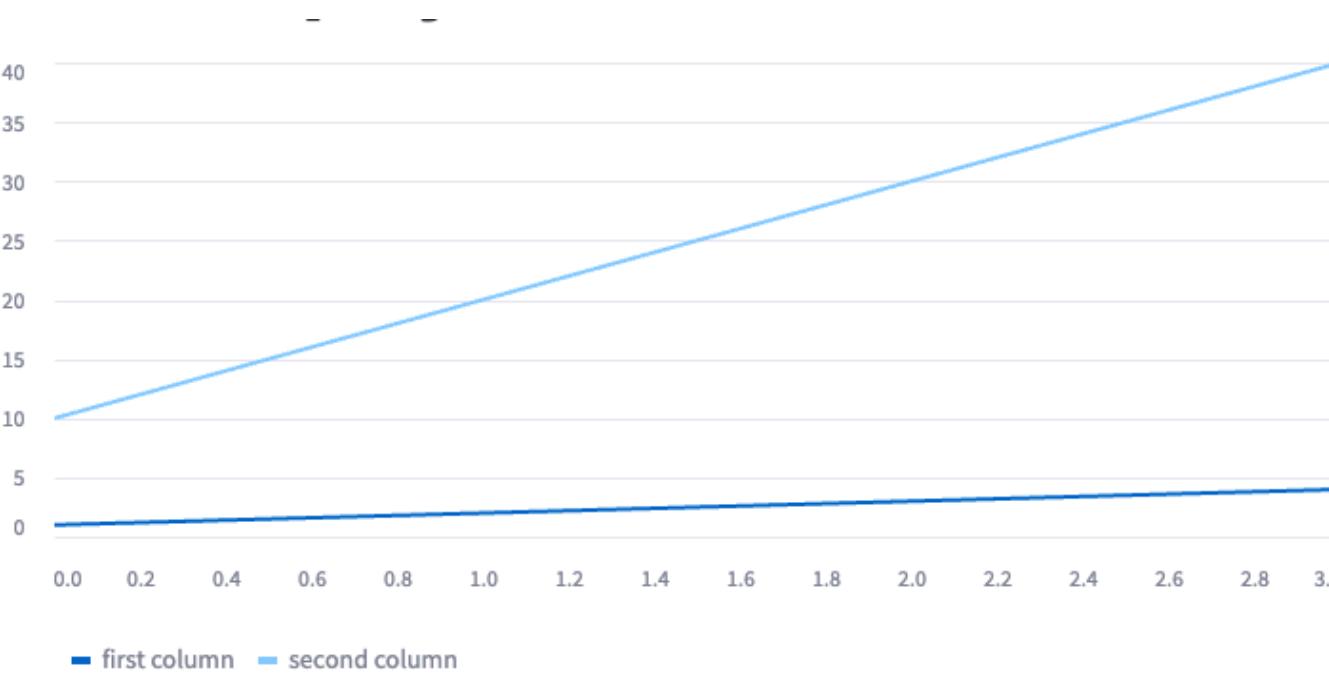
	first column	second column
0	1	10
1	2	20
2	3	30
3	4	40

```
{ "first column": [
    0: 1,
    1: 2,
    2: 3,
    3: 4
],
"second column": [
    0: 10,
    1: 20,
    2: 30,
    3: 40
]}
```

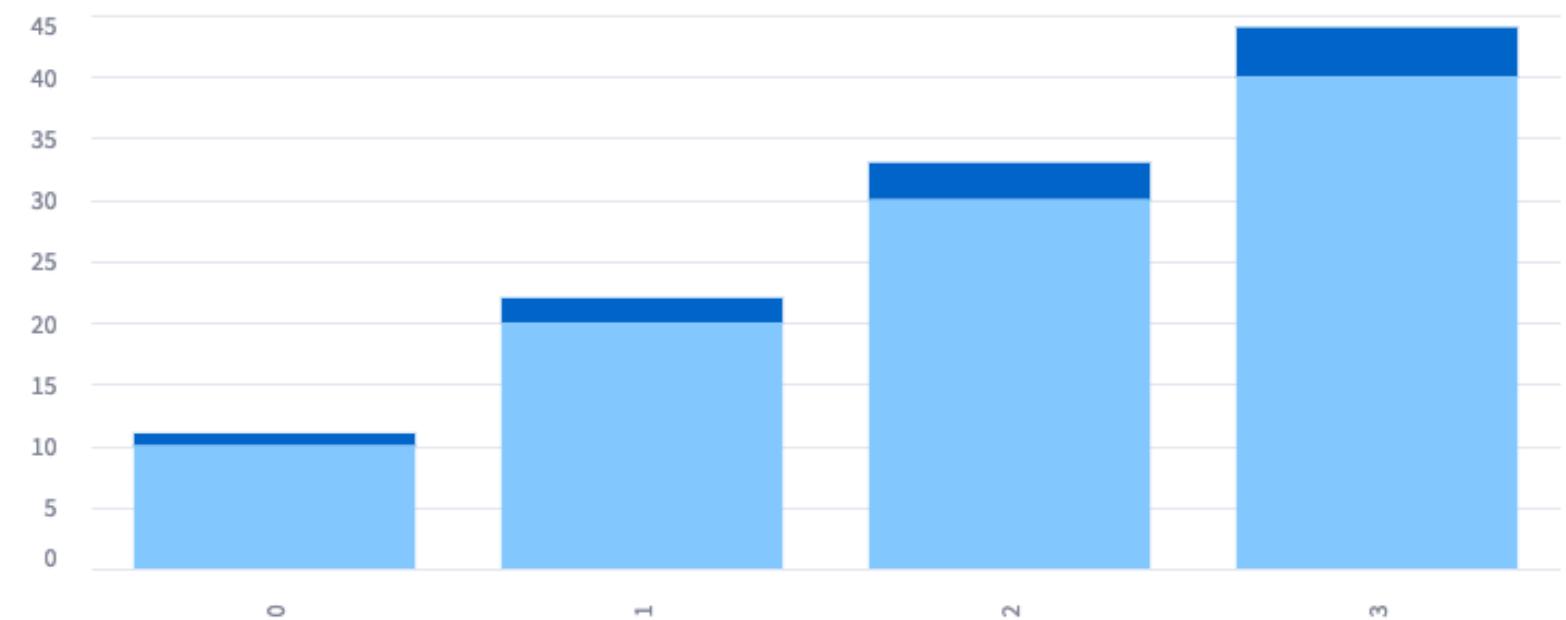


Data Display II: Built-In Viz

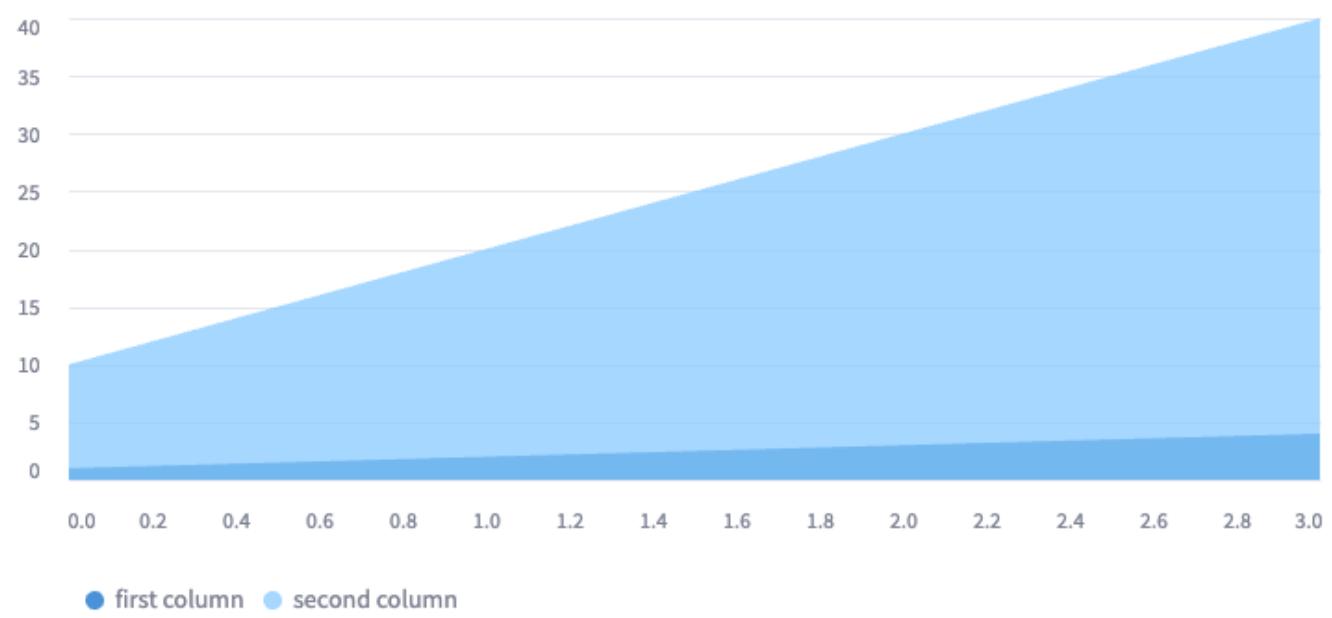
Streamlit offers built-in plotting functions for quick visualizations:



1 `st.line_chart()`



2 `bar_chart()`



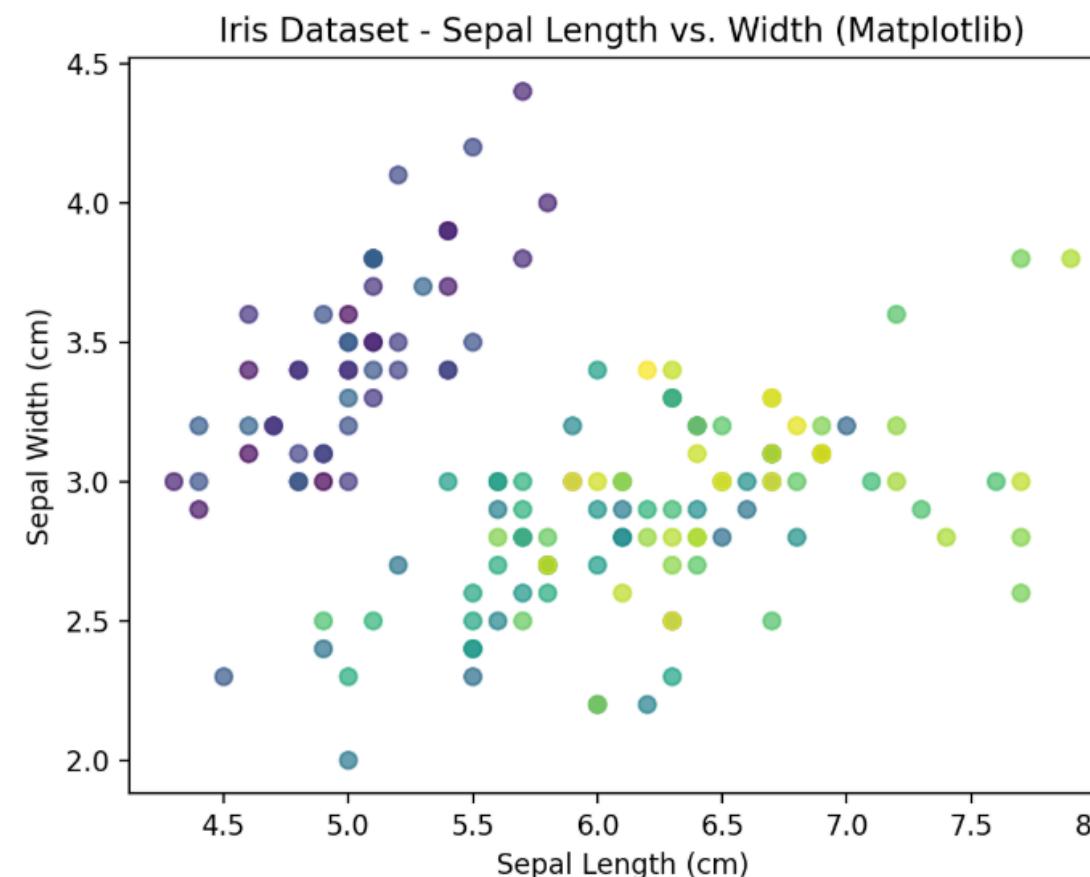
3 `st.area_chart()`

Data Display III: Using Viz Libraries

Streamlit supports external visualization libraries for more customization.

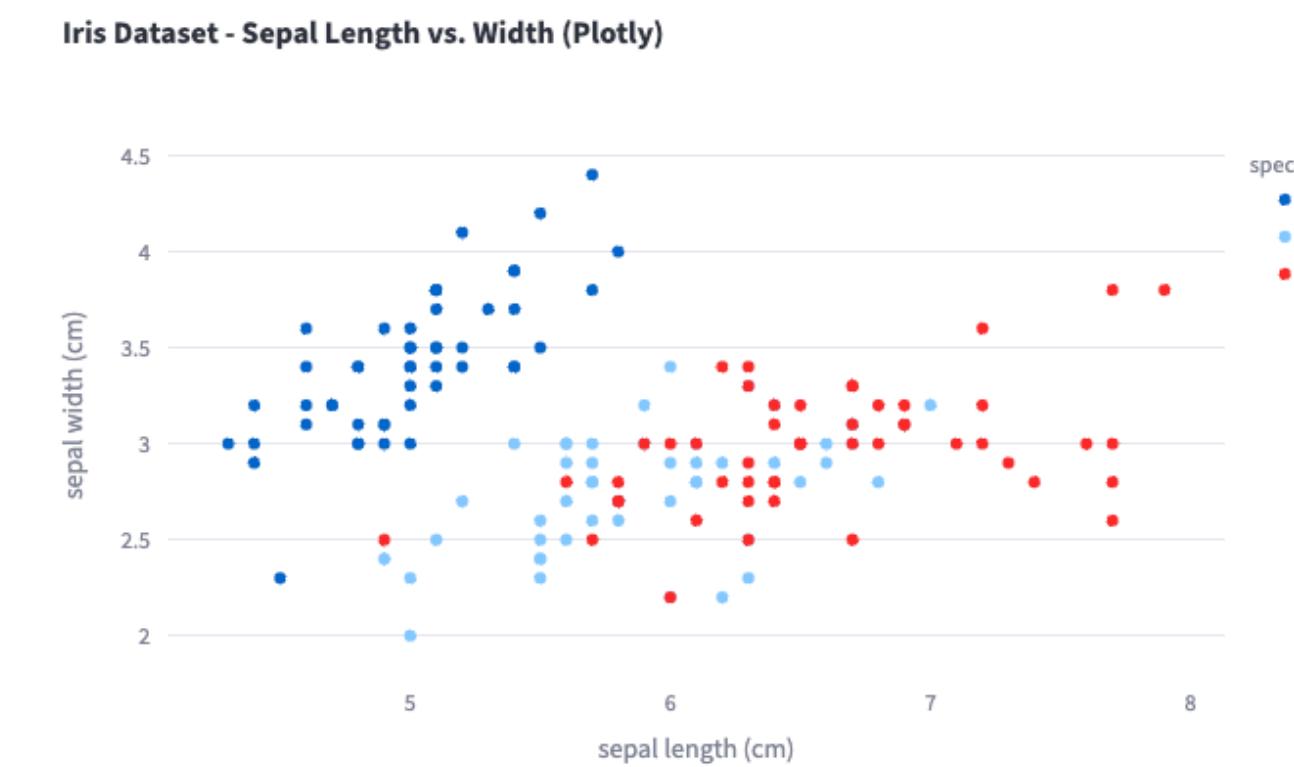
Matplotlib

Create Matplotlib figures in your normal workflow, then display with `st.pyplot(fig)`.



Plotly

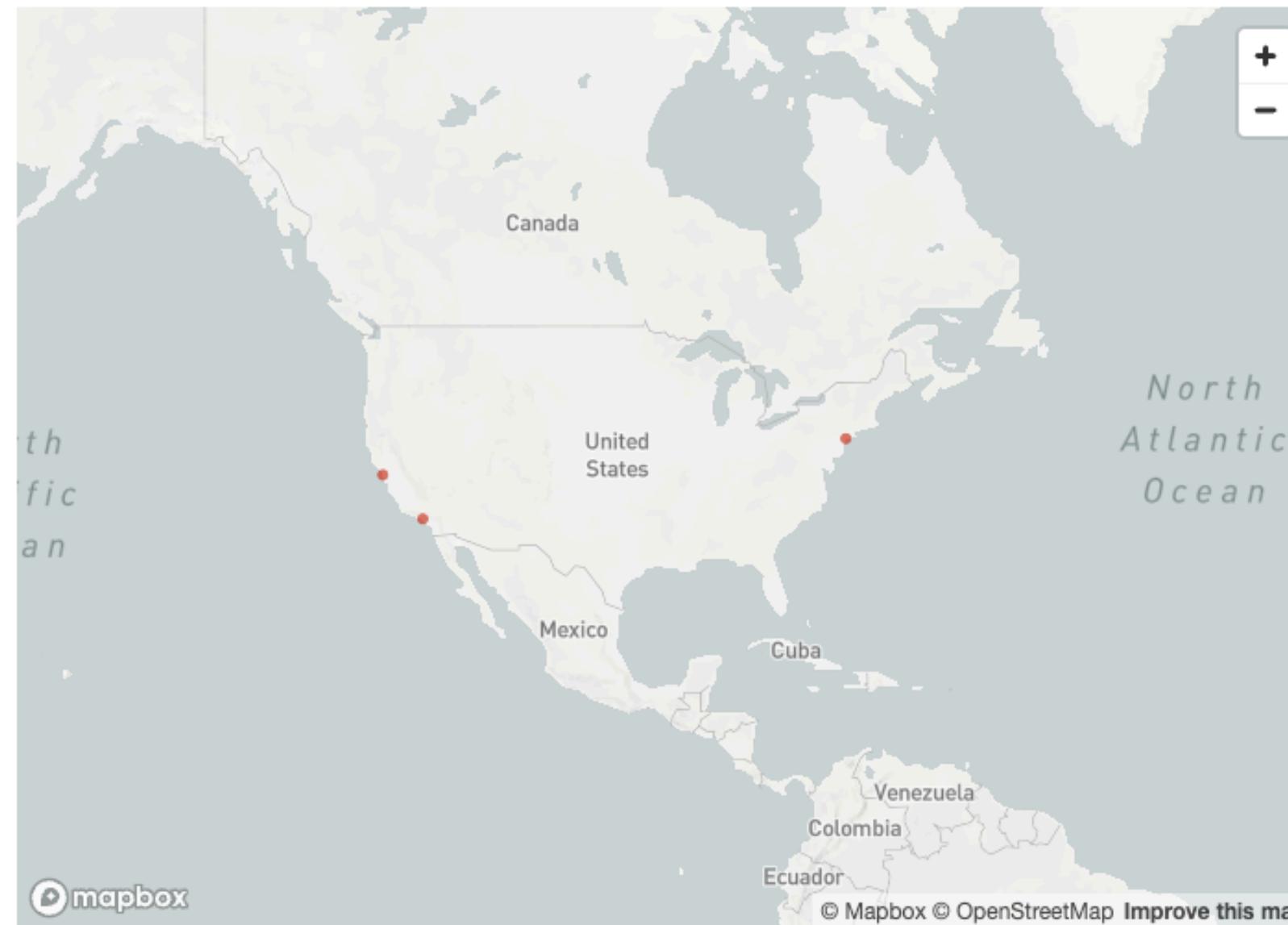
Build interactive Plotly charts and render them with `st.plotly_chart(fig)`.



Data Display IV: Maps and Geospatial Data

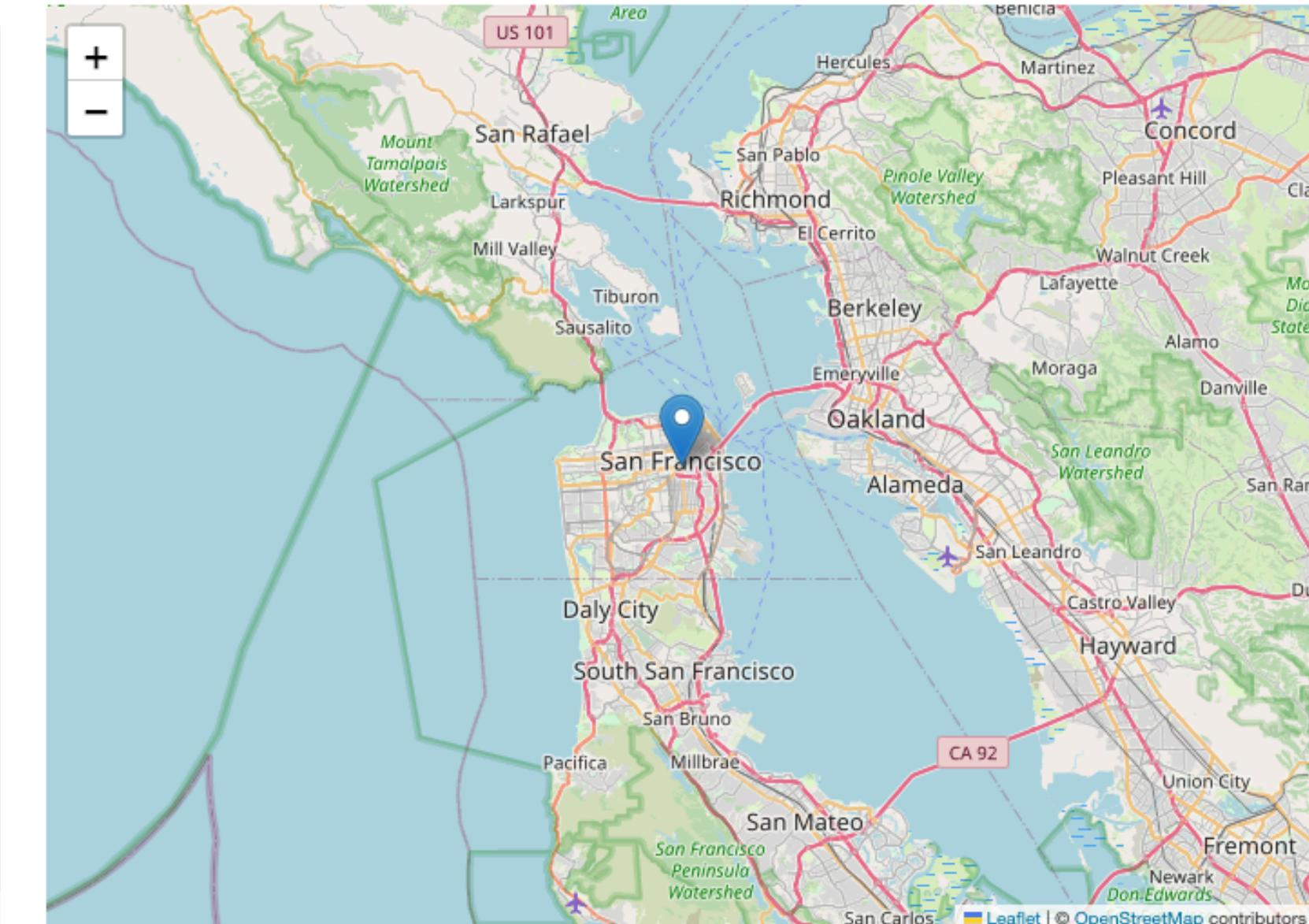
Basic Maps

Display geographic data using `st.map(data)` with a DataFrame containing lat/lon columns.



Custom Maps

Create advanced maps using Folium, Pydeck...

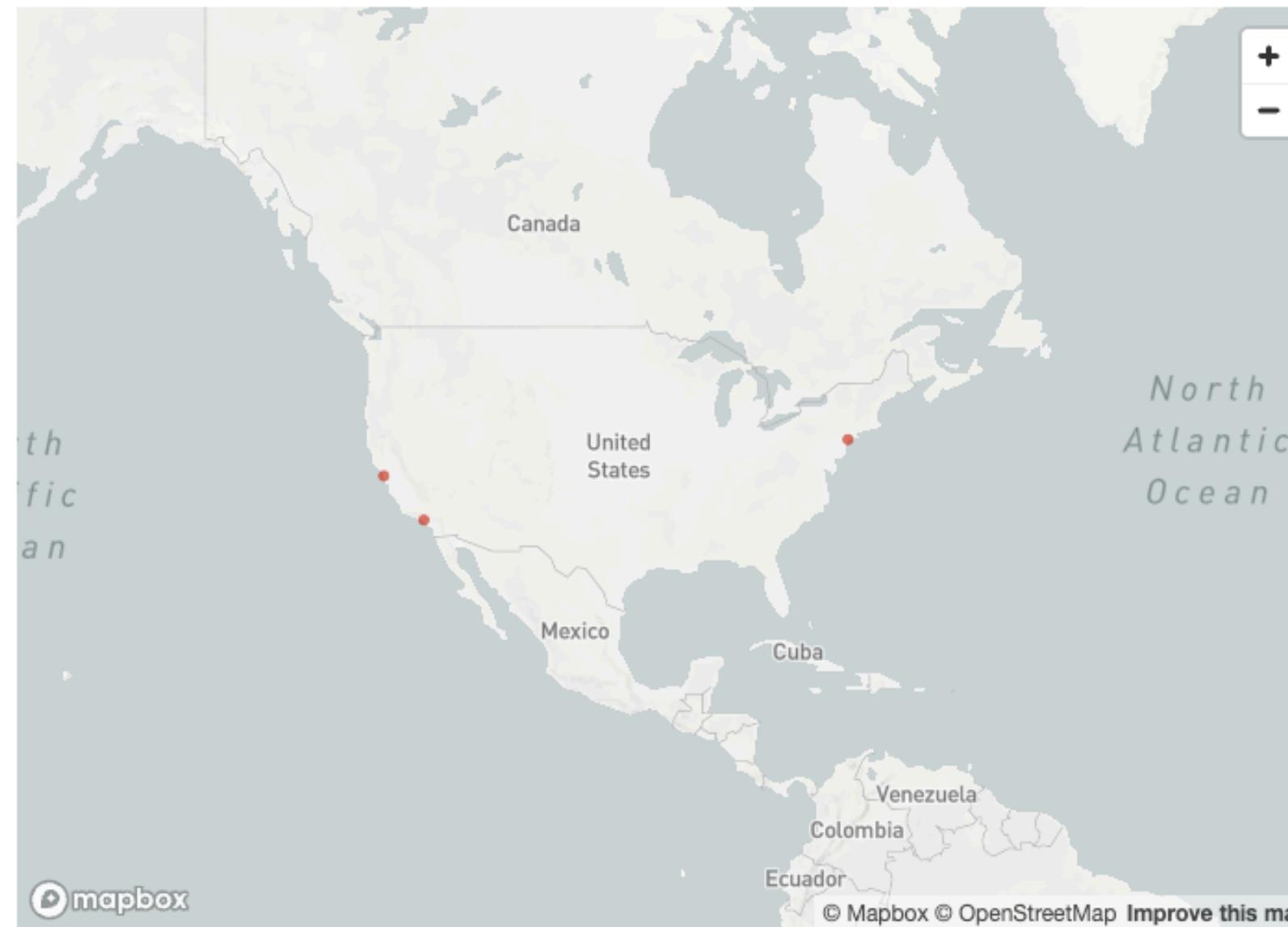


Streamlit Components

Data Display IV: Maps and Geospatial Data

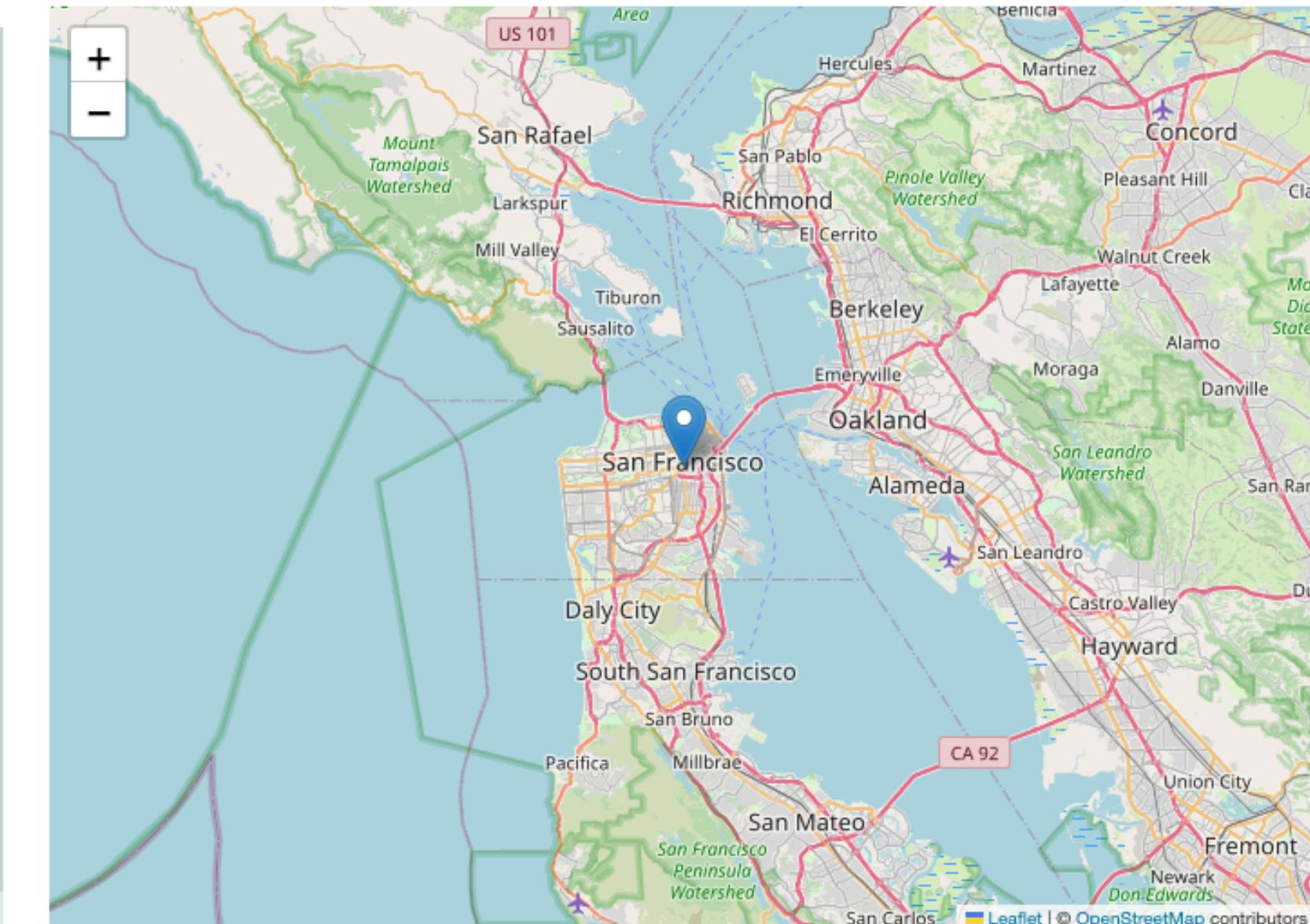
Basic Maps

Display geographic data using `st.map(data)` with a DataFrame containing lat/lon columns.



Custom Maps

Create advanced maps using Folium, Pydeck...



Media Elements: Images, Audio, Video

Streamlit supports multimedia elements for better user engagement.



Images

Display images with **st.image()**. Support for local files, URLs, and NumPy arrays.



Audio

Play audio files with **st.audio()**. Supports most common audio formats.



Video

Embed videos with **st.video()**. Works with files, URLs, and YouTube links.



Widgets & User Interactions

- Widgets allow users to interact with the app dynamically.
- Includes buttons, checkboxes, sliders, text inputs, and more.

Buttons

[Click Me!](#)

Checkbox

I agree

Radio

Select your age



You selected: 25

Input widgets - Streamlit Docs

Thanks for stopping by! We use cookies to help us understand how you interact with our website.

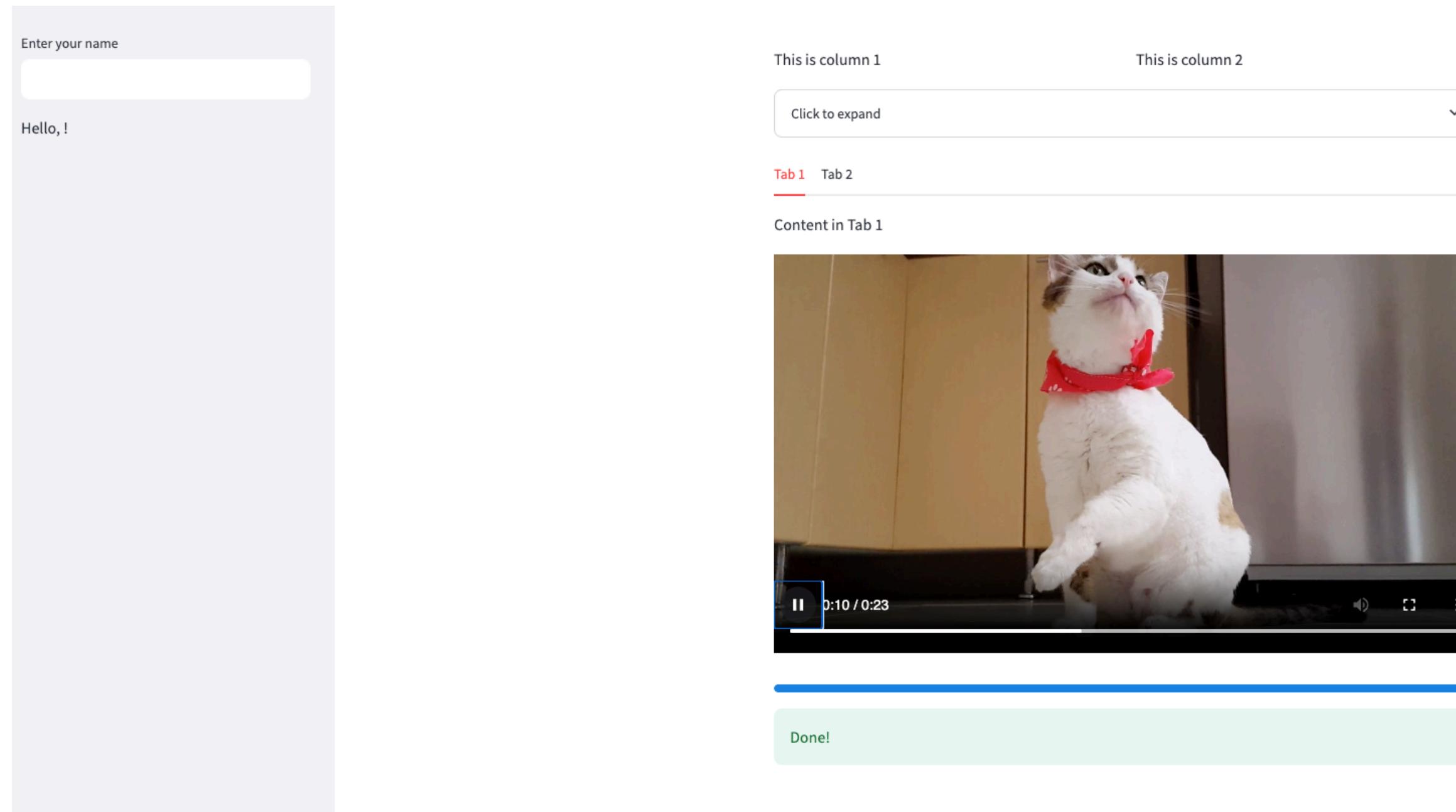
 streamlit.io



Streamlit Layout

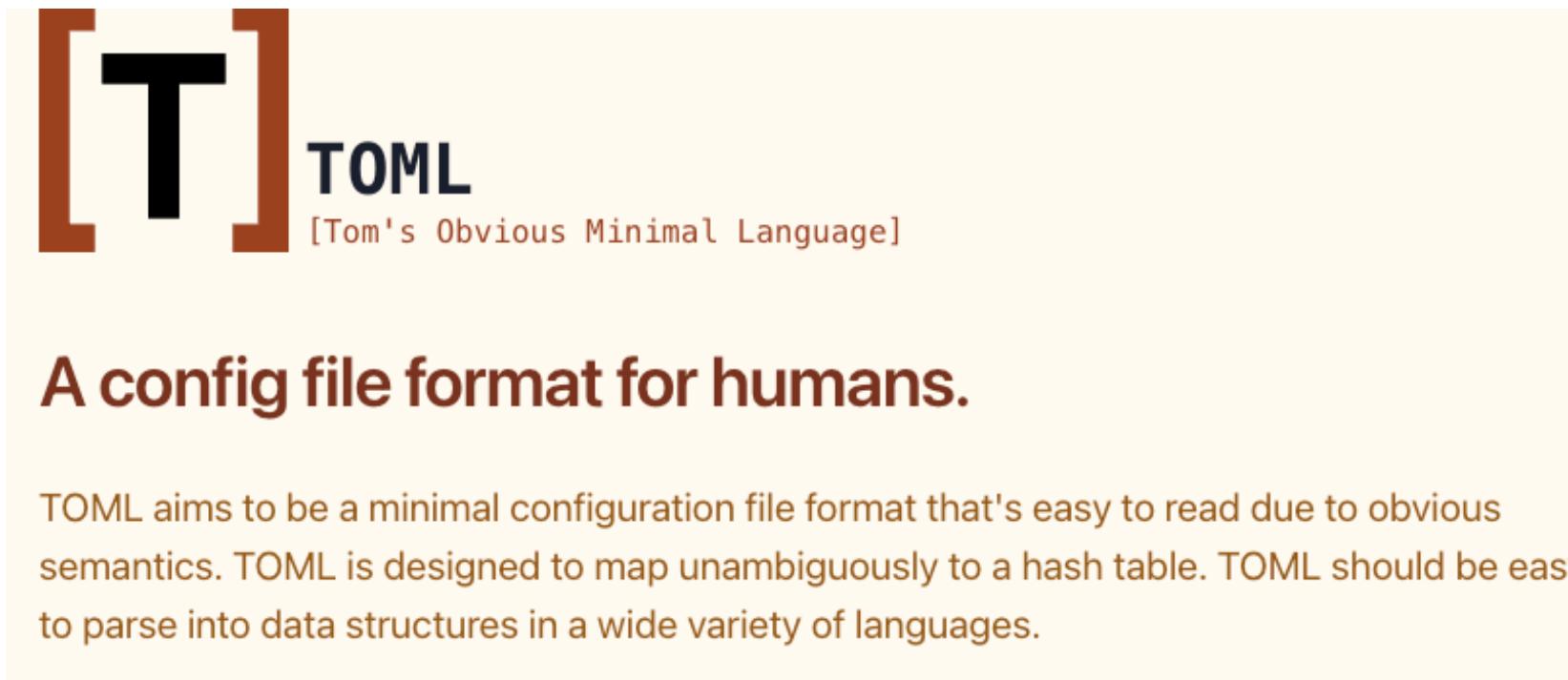
Introduction to Layouts

- Use columns, sidebars, tabs, and expanders to organize content.
- Use progress bars and status messages for better UX.



config.toml

- Streamlit allows theme customization through config.toml.



```
[theme]
primaryColor = "#FF4B4B"
backgroundColor = "#FFFFFF"
secondaryBackgroundColor = "#F0F2F6"
textColor = "#262730"
```

Streamlit Deployment

Deployment in Streamlit Cloud

- 1** Push your script to GitHub
- 2** Go to [Streamlit Cloud](#)
- 3** Connect GitHub repo & deploy
- 4** Share the public URL

What would you like to do?



Deploy a public app from GitHub

My code is ready on a GitHub repo, and it is totally awesome.

[Deploy now](#)



Deploy a public app from a template

I want to see what kind of amazing concoctions you have for me.

[Check out templates](#)



Deploy a private app in Snowflake

I want unlimited enterprise-grade apps, with the security of Snowflake.

[Start trial →](#)

Best Practices. When to use Streamlit?

- Model Testing & Debugging (ML & AI Prototyping).
- Fast Dashboards & Real-Time Data Apps
- Interactive Data Exploration & EDA (Exploratory Data Analysis)
- Deployment of Interactive Reports
- Real-Time Monitoring & Alerting

✖ When NOT to Use Streamlit

- For large-scale web apps – Streamlit isn't Flask or Django.
- For complex authentication – It lacks built-in user management.
- For performance-heavy tasks – Large datasets can slow down Streamlit if not optimized.
- For multi-user collaboration in real-time
- Streamlit is not designed for multi-user sessions like Dash or Shiny.



next EDUCACIÓN