Data Visualization with Streamlit

Introducing Streamlit

Streamlit

Introducing Streamlit

- Streamlit is a framework for building interactive, data-driven applications in Python.
- Streamlit allows you to quickly and easily create interactive dashboards and applications.
- Streamlit abstracts all the javascript and other web components so that a data analyst or data scientist can focus on the data and not web components.

Streamlit

You don't run Streamlit in Jupyter

Starting a Streamlit Application

- You must write your code in a standard python file.
- When you are ready to launch, simply execute the following command. It will launch a web server to host your application.
- From the command line, run:

```
streamlit run your_script.py
```

You will be able to access your application at http://localhost:8501

There are various command line options which you can find in the streamlit docs here: https://docs.streamlit.io/

Basic Streamlit Concepts

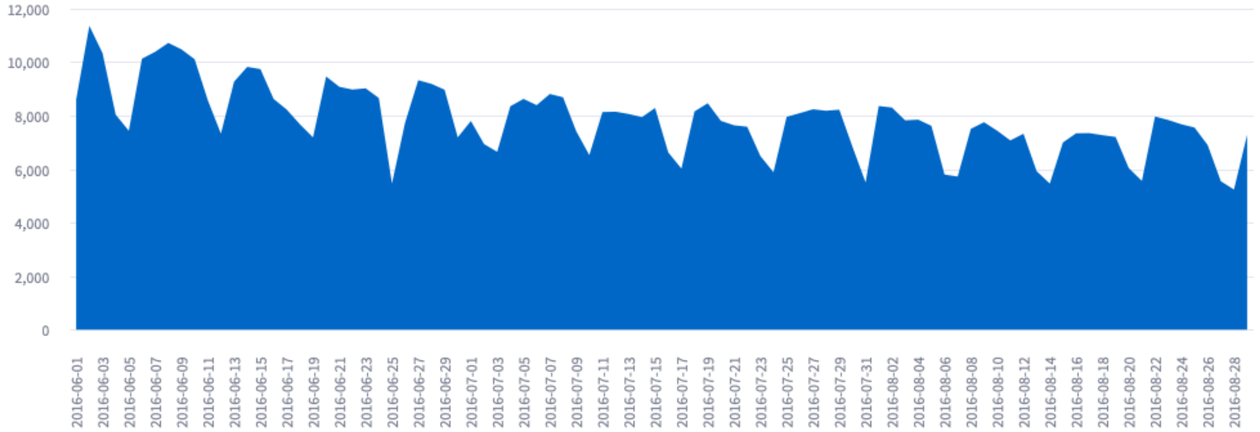
```
import streamlit as st
import pandas as pd


# Display some text...
st.markdown("""# Basic Application
This basic application will demonstrate how to display data, both as a table and as a chart."""
)


# Data ingest skipped for brevity
# Display the line chart
st.area_chart(line_chart_data)
st.dataframe(data, hide_index=True)
```

localhost

Deploy

Basic Application

This basic application will demonstrate how to display data, both as a table and as a chart.



date	botfam	industry	hosts	orgs
2016-06-03	Zusy	Retail	527	5
2016-06-04	Bedep	Education	24	16
2016-06-04	Bedep	Finance	360	15
2016-06-04	Bedep	Government/Politics	259	18
2016-06-04	Bedep	Healthcare/Wellness	80	11
2016-06-04	Bedep	Manufacturing	60	10
2016-06-04	Bedep	Retail	544	8
2016-06-04	ConfickerAB	Education	420	84
2016-06-04	ConfickerAB	Finance	233	45

Presenting Data with Streamlit

- You can see from the previous example how easy it is to present data using Streamlit
- Complete charting docs are available here: https://docs.streamlit.io/develop/api-reference/charts
- Each chart can be customized extensively.
- Complete code for the basic example is in the streamlit_examples folder in the course github repo.

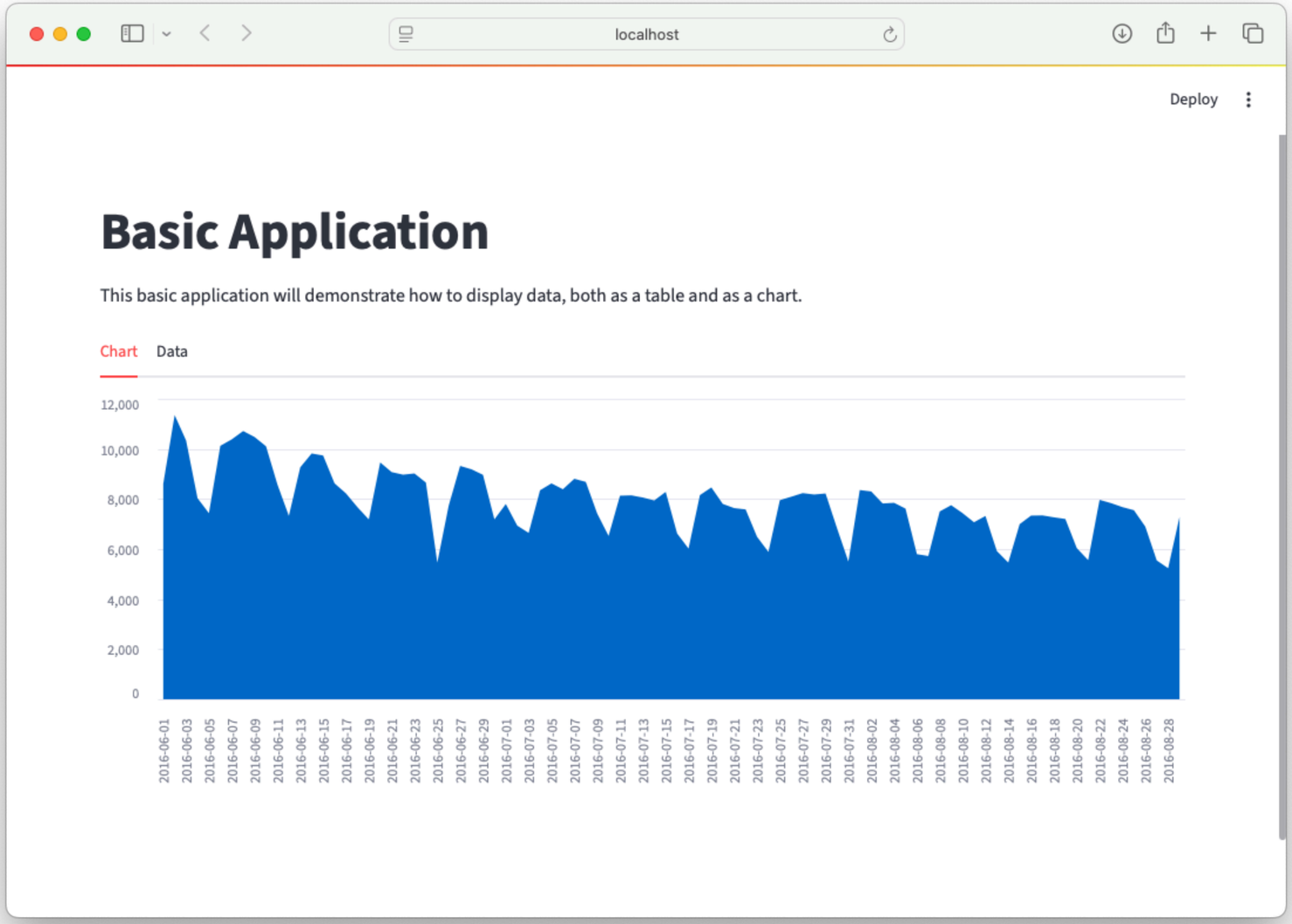
Laying Out Your Application

- Streamlit has a variety of layout components including tabs, pages, containers, expanders, popovers, modal dialogues and more.
- Complete list is available here: https://docs.streamlit.io/develop/api-reference/layout
- You can use the with notation to create a layout option, then add your streamlit elements in that with block

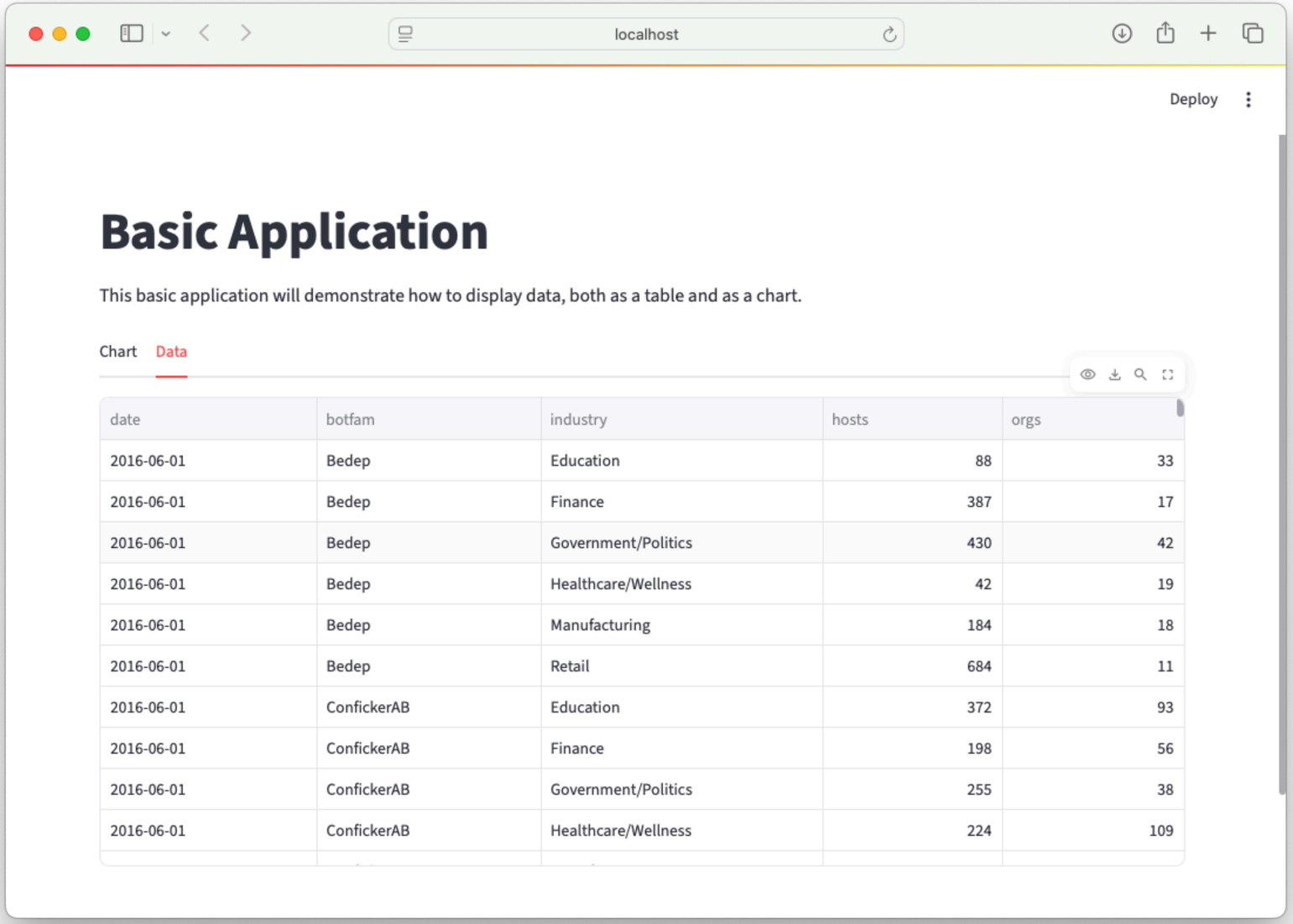
Laying Out Your Data

```
# Create the tabs
chart_tab, table_tab = st.tabs(["Chart", "Data"])

# Display the line chart
with chart_tab:
    st.area_chart(line_chart_data)

# Display the table
with table_tab:
    st.dataframe(raw_data, hide_index=True)
```

Laying Out Your Data



Laying Out Your Data



date	botfam	industry	hosts	orgs
2016-06-01	Bedep	Education	88	33
2016-06-01	Bedep	Finance	387	17
2016-06-01	Bedep	Government/Politics	430	42
2016-06-01	Bedep	Healthcare/Wellness	42	19
2016-06-01	Bedep	Manufacturing	184	18
2016-06-01	Bedep	Retail	684	11
2016-06-01	ConfickerAB	Education	372	93
2016-06-01	ConfickerAB	Finance	198	56
2016-06-01	ConfickerAB	Government/Politics	255	38
2016-06-01	ConfickerAB	Healthcare/Wellness	224	109

Taking Input from a User

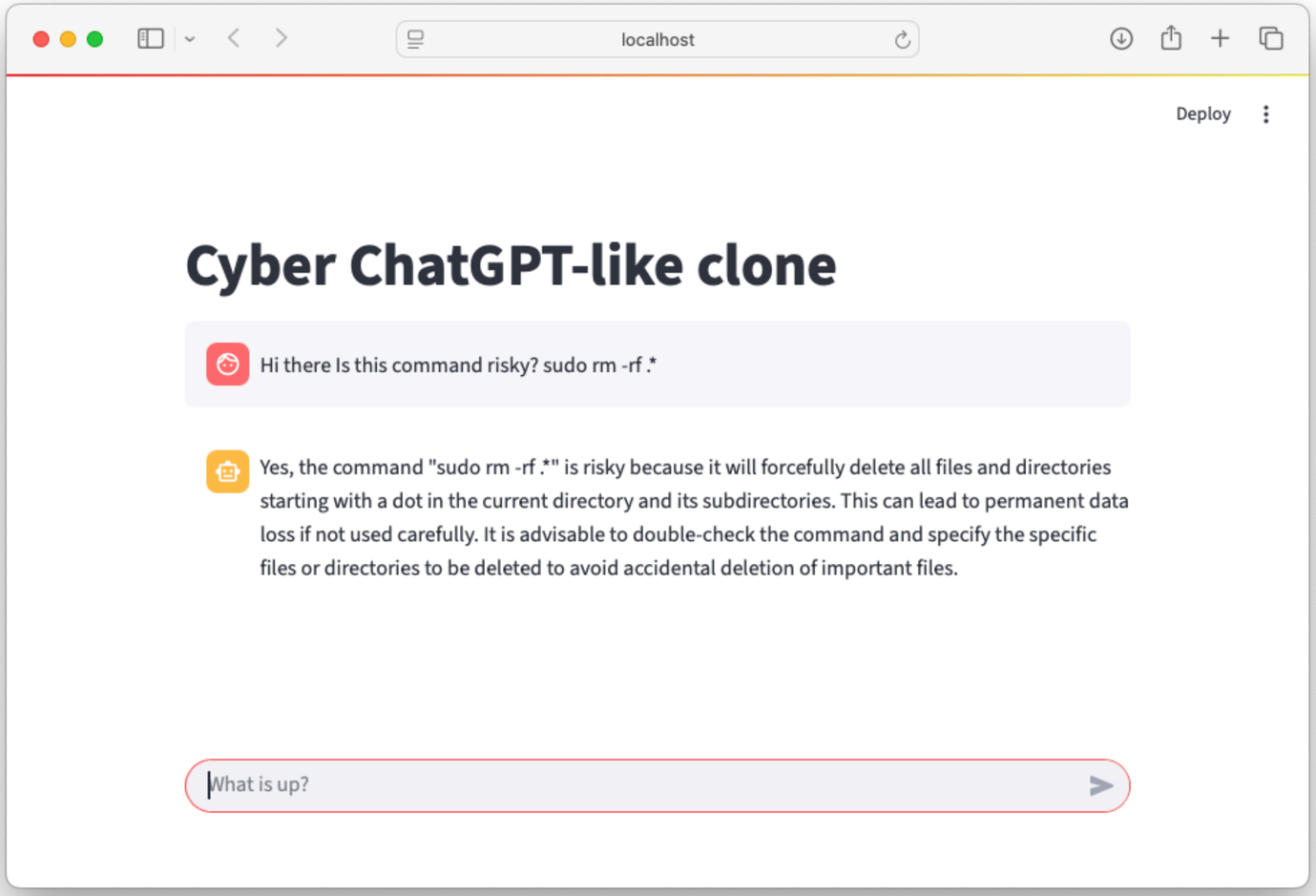
- Streamlit has a large collection of input widgets available here: https://docs.streamlit.io/develop/api-reference/widgets
- When the input element is updated, it creates an event which causes the page (and data) to be reloaded.
- The code below adds a textbox which will collect a value from the user and display it:
- Remember to validate user input!

```
name = st.text_input("Enter your name")
if len(name) > 0:
    st.write(f"Hello, {name}")
```

Building AI Applications with Streamlit

- Streamlit has a chat_input and chat_message widget which are built for interacting with LLMs.
- Full docs are available here: https://docs.streamlit.io/develop/api-reference/chat/st.chat_input

Building AI Applications with Streamlit



Building AI Applications with Streamlit

```
client = OpenAI(api_key=os.getenv("OPENAI_KEY"))
st.session_state["openai_model"] = "gpt-3.5-turbo"

if "messages" not in st.session_state:
    st.session_state.messages = []

for message in st.session_state.messages:
    with st.chat_message(message["role"]):
        st.markdown(message["content"])

       response = st.write_stream(stream)
    st.session_state.messages.append({"role": "assistant", "content": response})
```

Building AI Applications with Streamlit

```
if prompt := st.chat_input("What is up?"):
    st.session_state.messages.append({"role": "user", "content": prompt})
    with st.chat_message("user"):
        st.markdown(prompt)


    with st.chat_message("assistant"):
        system_message = {
            "role": "system",
            "content": "You are a helpful security bot that provides concise and accurate answers to cybersecurity questions. You may only answer cybersecurity questions.",
        }
        stream = client.chat.completions.create(
            model=st.session_state["openai_model"],
            messages=[system_message] + [
                {"role": m["role"], "content": m["content"]}
                for m in st.session_state.messages
            ],
            stream=True,
        )
```

Questions?

Lab

Please take 30 min to complete Interactive Visualization Worksheet (Not in Jupyter)