

Streamlit Introduction

Content

1. Install and run a streamlit server
2. Editing static elements
(markdown, images)
3. Adding altair visualizations
4. Adding widgets

Get Started

See the streamlit documentation:

https://docs.streamlit.io/en/stable/getting_started.html

1. [optional] Set up virtual environment (optional – “conda create...”)

2. You’ll need Python 3.6+ and pip (type “pip” to check)

```
conda install pip
```

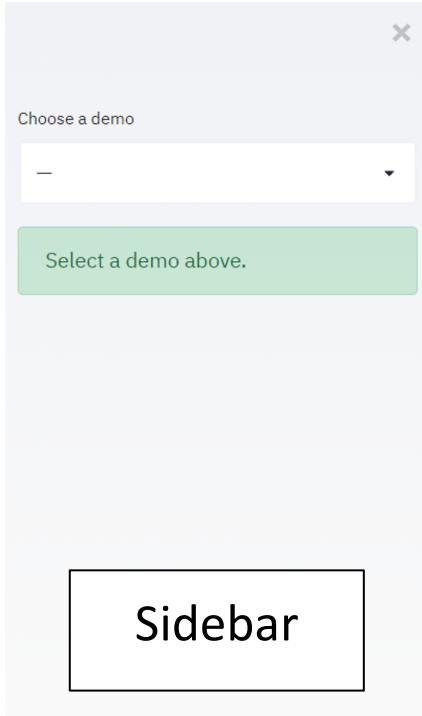
3. Install streamlit (the latest as of 9/30 is 0.67, so we’ll use that version)

```
pip install streamlit==0.67
```

4. Run the following command in the terminal:

```
streamlit hello
```

streamlit hello



Welcome to Streamlit! 🙌

Streamlit is an open-source app framework built specifically for Machine Learning and Science projects.

👉 **Select a demo from the dropdown on the left** to see some examples of what Streamlit can do!

Want to learn more?

- Check out [streamlit.io](#)
- Jump into our [documentation](#)
- Ask a question in our [community forums](#)

See more complex demos

- Use a neural net to analyze the [Udacity Self-driving Car Image Dataset](#)
- Explore a [New York City rideshare dataset](#)

Main
Section

Adding Static Elements

Start a server

1. Make a new python script (You can use any file editor)
2. Import streamlit

```
import streamlit as st
```

1. Open your terminal/command line/anaconda prompt, move (use cd <dir path>) to the directory where you store your python script
2. Run the following command:

```
streamlit run <filename>
```

```
(base) C:\>cd C:\licia\course\streamlit_lab
```

```
(base) C:\licia\course\streamlit_lab>streamlit run streamlit_basic_example.py
```

You can now view your Streamlit app in your browser.

Local URL: <http://localhost:8501>

Network URL: <http://192.168.1.4:8501>

This is where we are storing our python script (you can put it anywhere)

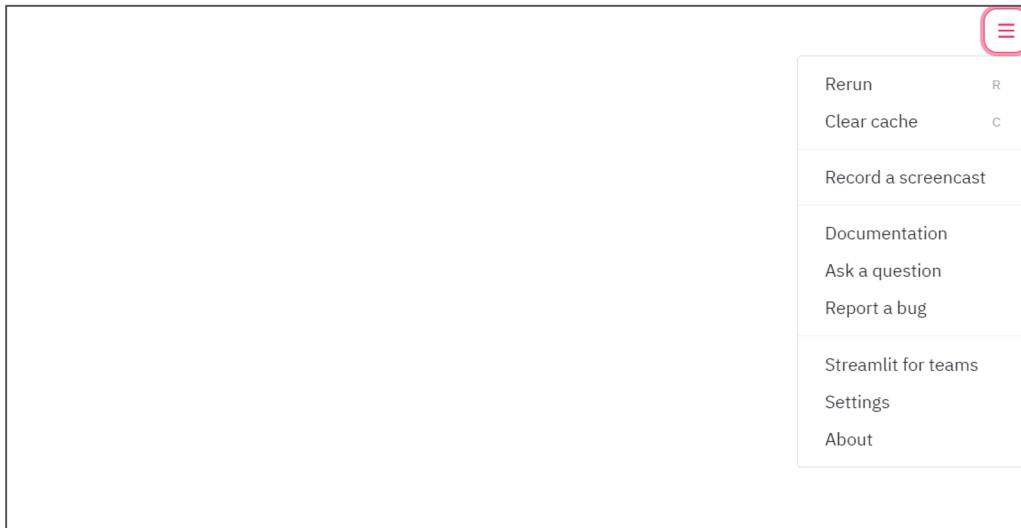
This is the name of our script (you can name it anything)

Open your browser and navigate to this address

Result of step 1

```
(base) C:\licia\course\streamlit_lab>streamlit run streamlit_basic_example.py  
You can now view your Streamlit app in your browser.  
Local URL: http://localhost:8501  
Network URL: http://192.168.1.4:8501
```

After navigating to the address, you will see a blank page with only a menu button on the top right



Adding a Title

Two methods that generate identical results

My Title

Markdown Title

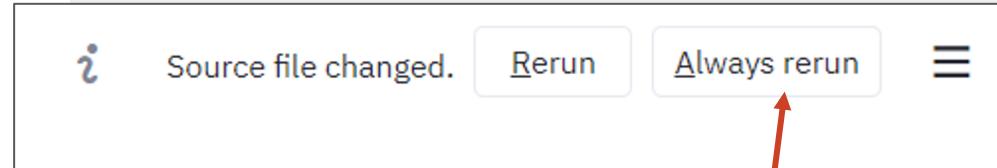
Method 1:

```
st.title("My Title")
```

Method 2:

```
st.write("# Markdown Title")
```

Whenever you change the script, you will see the following content on your webpage



Select this so that it automatically reruns for you.

Using st.write() to add markdown

st.write() is a powerful command. You can read more about it here:

<https://docs.streamlit.io/en/stable/api.html#streamlit.write>

Paste the following lines to your script and see what happens

```
st.write("I can format text content  
into different styles such as  
**bold**, *italics*, and  
~~strikethrough~~")
```

```
st.write("I am writing a link [to the  
streamlit documentation  
page] (https://docs.streamlit.io/en/stab  
le/api.html)")
```

```
st.write('![alt  
text] (https://upload.wikimedia.org/wiki  
pedia/commons/thumb/3/3e/Irises-  
Vincent_van_Gogh.jpg/314px-Irises-  
Vincent_van_Gogh.jpg)')
```

Result of step 3

I can format text content into different styles such as **bold**, *italics*, and ~~strikethrough~~

I am writing a link [to the streamlit documentation page](#)



Markdown is a language to format plain-text content. You can take a look at this cheatsheet:

<https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>

Making and displaying an altair visualization

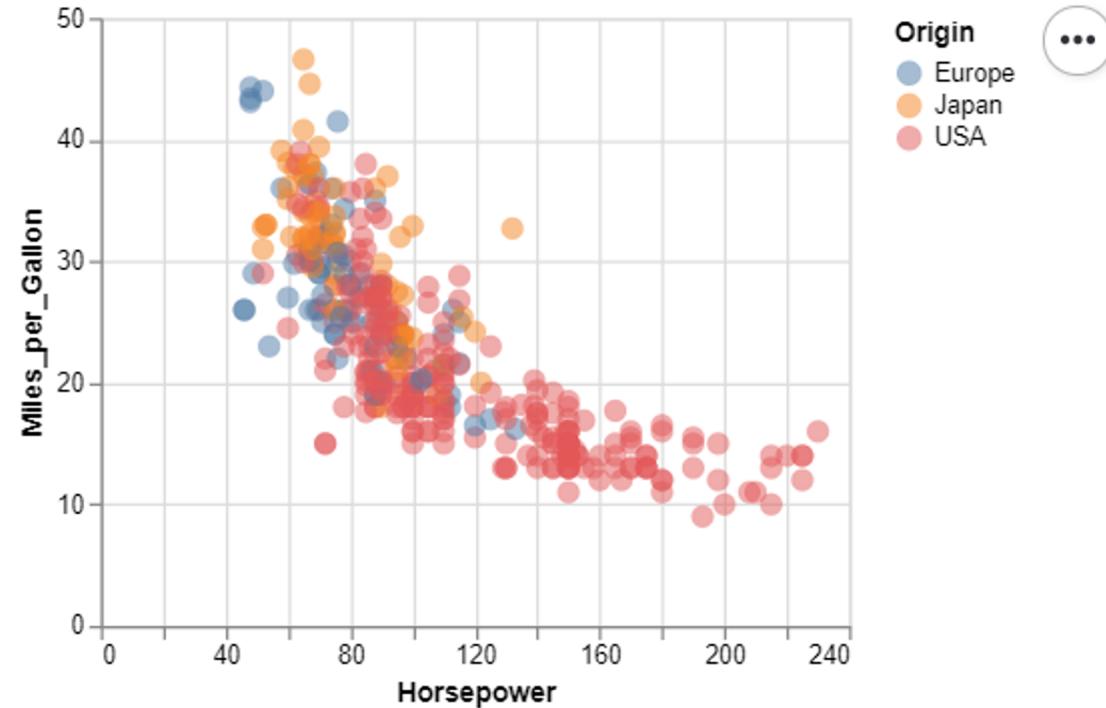
```
import pandas as pd
import altair as alt

car_url =
"https://raw.githubusercontent.com/altair-
viz/vega_datasets/master/vega_datasets/_dat-
a/cars.json"
cars = pd.read_json(car_url)

hp_mpg=alt.Chart(cars).mark_circle(size=80,
opacity=0.5).encode(
    x='Horsepower:Q',
    y='Miles_per_Gallon:Q',
    color="Origin"
)

st.write(hp_mpg)
```

Result of step 4



Magic Command

Write data and markdown without
the “st.write()”

Read more here:

<https://docs.streamlit.io/en/stable/api.html#magic-commands>

Delete all the st.write () commands and leave the command parameters in your script. It still works!

E.g., instead of

```
st.write("I can format text content  
into different styles such as  
**bold**, *italics*, and  
~~strikethrough~~")
```

```
st.write(hp_mpg)
```

Simply use

```
"I can format text content into  
different styles such as **bold**,  
*italics*, and ~~strikethrough~~"
```

```
hp_mpg
```

Add a button to toggle the display of the chart

Streamlit offers a number of built-in widgets such as button, checkbox, radio, selectbox. Read more here:

<https://docs.streamlit.io/en/stable/api.html#display-interactive-widgets>

```
#initialize a button
btn=st.button("display hp_mpg")
#if the button is clicked
if btn: #display the chart
    hp_mpg
else:
    #display the following text
    "click the button to display
    the hp_mpg chart"
```

Widgets hold users' inputs

Result of step 6

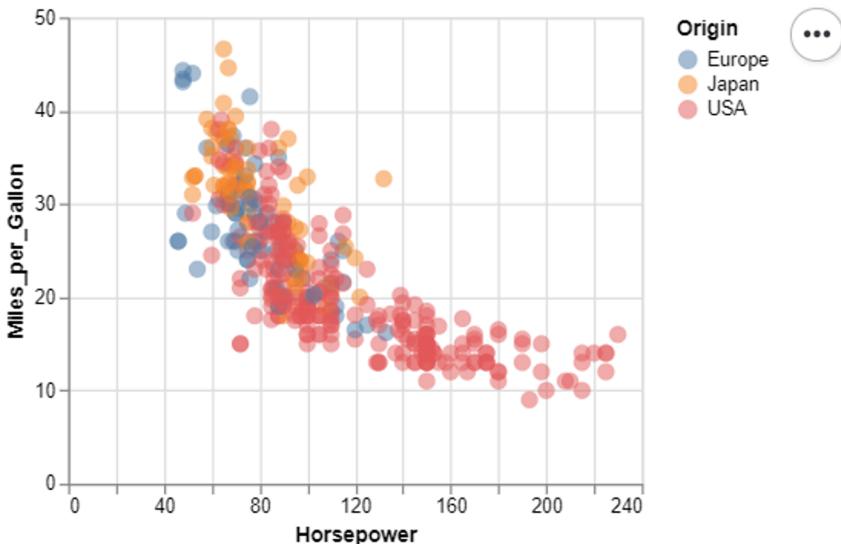
Before clicking

display hp_mpg

click the button to display the hp_mpg chart

After clicking

display hp_mpg



Make a select widget and use user input in a chart

The select widget creates a dropdown menu. Read more here:

<https://docs.streamlit.io/en/stable/api.html#display-interactive-widgets>

```
#step 8: make a select widget
##create a list of options
y_axis_options=["Acceleration","Miles_per_Gallon","Displacement"]
##create a select box
y_axis_selectbox=st.selectbox(
    label="Select the column to plot for the y-axis",
    options=y_axis_options
)
## create a chart that uses the select box
chart=alt.Chart(cars).mark_circle(size=80,
opacity=0.5).encode(
    x='Horsepower:Q',
    y=y_axis_selectbox,
    color="Origin"
)
#render the chart
chart
```

Widgets hold users' inputs

Result of step 7

Select the column to plot for the y-axis

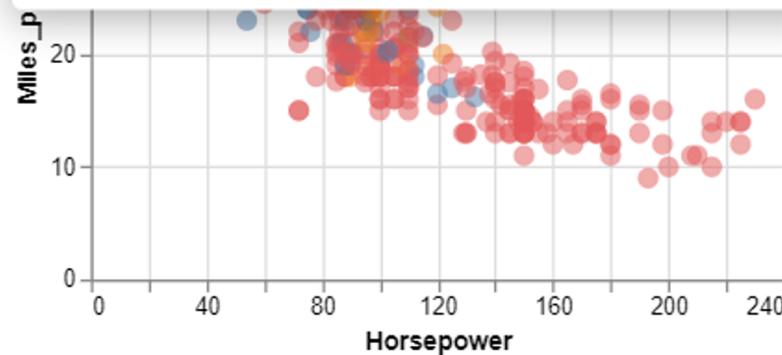
Miles_per_Gallon



Acceleration

Miles_per_Gallon

Displacement



Using the sidebar

Almost everything we have covered so far can be displayed in the sidebar by changing

`st.[element]` to
`st.sidebar.[element]` with the exception of `st.write()`

Read more:

<https://docs.streamlit.io/en/stable/api.html#add-widgets-to-sidebar>

E.g., instead of

`st.title("My Title")`

Write

`st.sidebar.title("My Title")`

Instead of

`st.write("# Markdown Title")`

Write

`st.sidebar.markdown("# Markdown Title")`

Result of step 8

x

My Title

Markdown Title

You can add almost any element to the sidebar: try moving your visualization and interactive widgets to the sidebar.

My Title

Markdown Title

I can format text content into different styles such as **bold**, *italics*, and ~~strikethrough~~

I am writing a link to the [streamlit documentation page](#)



display hp_mpg

click the button to display the hp_mpg chart

Select the column to plot for the y-axis

Miles_per_Gallon

Origin

Europe

Japan



Getting from Jupyter to Streamlit

```
import streamlit as st
```

Then:

just save your notebook as a .py file! Should work (more or less)