ILA1 - Interactive Data Visualization - Alex Beyer

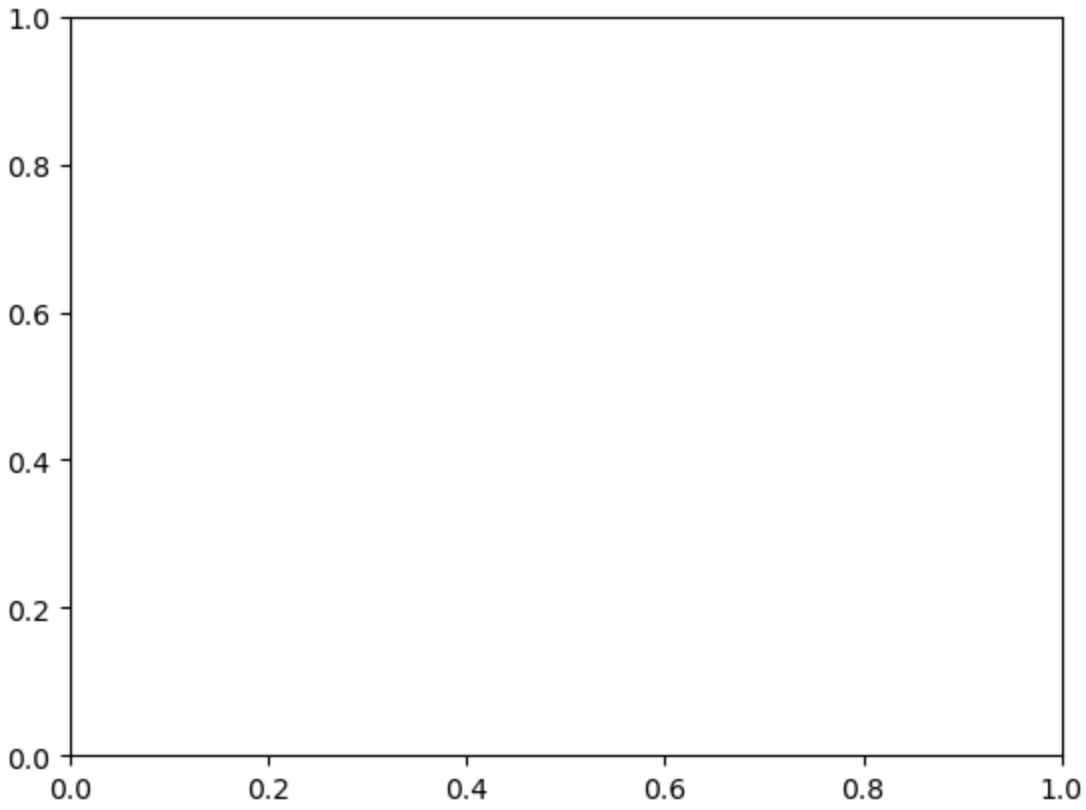
This should work for the most recent versions of all libraries used. I had consistent issues with node.js when running on my desktop that caused this to do strange things (usually crash and or generate infinite plots) but forcibly disabling any and all JavaScript in VSCode fixed that, somehow. If the issues persist on your end let me know and I will send a video of this working properly.

In [ ]:
```
#library imports
import numpy as np
import matplotlib.pyplot as plt
import ipywidgets as widg
from IPython.display import display, clear_output
from sklearn.datasets import fetch_california_housing

#get housing data
caHousingFrame = fetch_california_housing(data_home='.', as_frame = True)

#initialize plot object
fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)

#define an updater function to be called by sliders when moved
def updatePlot(e):
    with out:
        #get updated slider values
        inc = [incMinSlider.value, incSlider.value]
        age = [ageMinSlider.value, ageSlider.value]
        room = [roomMinSlider.value, roomSlider.value]
        bed = [bedMinSlider.value, bedSlider.value]
        pop = [popMinSlider.value, popSlider.value]
        occ = [occMinSlider.value, occSlider.value]
        lat = [latMinSlider.value, latSlider.value]
        long = [longMinSlider.value, longSlider.value]
        #clear out the old plot
        clear_output(True)
        ax.clear()
        #generate a new plot and mask out any data outside the user-set ranges from
        ax.scatter(np.array(caHousingFrame.data.Longitude)[\
                                                    np.logical_and(np.array(caH
                                                    np.array(caHousingFrame.dat
                                                    np.array(caHousingFrame.dat
                                                    np.array(caHousingFrame.dat
                                                    np.array(caHousingFrame.dat
                                                    np.array(caHousingFrame.dat
                                                    np.array(caHousingFrame.dat
                                                    np.array(caHousingFrame.dat
                                                    np.array(caHousingFrame.dat
                                                    np.array(caHousingFrame.dat
                                                    np.array(caHousingFrame.dat
                                                    np.array(caHousingFrame.dat
```

```
                                                    np.array(caHousingFrame.dat
                                                    np.array(caHousingFrame.dat
                                                    np.array(caHousingFrame.dat
                                                    np.logical_and(np.array(caH
                                                    np.array(caHousingFrame.dat
                                                    np.array(caHousingFrame.dat
                                                    np.array(caHousingFrame.dat
                                                    np.array(caHousingFrame.dat
                                                    np.array(caHousingFrame.dat
                                                    np.array(caHousingFrame.dat
                                                    np.array(caHousingFrame.dat
                                                    np.array(caHousingFrame.dat
                                                    np.array(caHousingFrame.dat
                                                    np.array(caHousingFrame.dat
                                                    np.array(caHousingFrame.dat
                                                    np.array(caHousingFrame.dat
                                                    np.array(caHousingFrame.dat
                                                    np.array(caHousingFrame.dat
        #draw, title and render the plot
        display(fig)
        ax.set_title("Filtered California Housing Data")
        plt.show()

#setup interactive widgets
#every one of these declarations follows the same 4 line format:
#declare max value slider (w/ args)
#define the observer using the updater from before
#declare min value slider (w/ args)
#define the observer for this slider the same way as before
#income
incSlider = widg.FloatSlider(description = "Max Median Income (*100k USD)", min = n
incSlider.observe(updatePlot)
incMinSlider = widg.FloatSlider(description = "Min Median Income (*100k USD)", min
incMinSlider.observe(updatePlot)
#house age
ageSlider = widg.FloatSlider(description = "Max House Age (years)", min = np.min(np
ageSlider.observe(updatePlot)
ageMinSlider = widg.FloatSlider(description = "Min House Age (years)", min = np.min
ageMinSlider.observe(updatePlot)
#num rooms
roomSlider = widg.FloatSlider(description = "Max Average # of Rooms", min = np.min(
roomSlider.observe(updatePlot)
roomMinSlider = widg.FloatSlider(description = "Min Average # of Rooms", min = np.m
roomMinSlider.observe(updatePlot)
#num bed rooms
bedSlider = widg.FloatSlider(description = "Max Average # of Bedrooms", min = np.mi
bedSlider.observe(updatePlot)
bedMinSlider = widg.FloatSlider(description = "Min Average # of Bedrooms", min = np
bedMinSlider.observe(updatePlot)
#popln
popSlider = widg.FloatSlider(description = "Max Population (100k people)", min = np
popSlider.observe(updatePlot)
popMinSlider = widg.FloatSlider(description = "Min Population (100k people)", min =
popMinSlider.observe(updatePlot)
#occupancy
```

```
occSlider = widg.FloatSlider(description = "Max Average Occupancy (# people)", min
occSlider.observe(updatePlot)
occMinSlider = widg.FloatSlider(description = "Min Average Occupancy (# people)", m
occMinSlider.observe(updatePlot)
#latitude
latSlider = widg.FloatSlider(description = "Max Latitude (deg)", min = np.min(np.ar
latSlider.observe(updatePlot)
latMinSlider = widg.FloatSlider(description = "Min Latitude (deg)", min = np.min(np
latMinSlider.observe(updatePlot)
#longitude
longSlider = widg.FloatSlider(description = "Max Longitude (deg)", min = np.min(np.
longSlider.observe(updatePlot)
longMinSlider = widg.FloatSlider(description = "Min Longitude (deg)", min = np.min(
longMinSlider.observe(updatePlot)

#define an output widget to contain the plot
out = widg.Output()

#finally, display everything onscreen
display(incMinSlider, incSlider, ageMinSlider, ageSlider, roomMinSlider, roomSlider
```

```
FloatSlider(value=0.4999, description='Min Median Income (*100k USD)', max=15.000
1, min=0.4999, step=0.0001)
FloatSlider(value=15.0001, description='Max Median Income (*100k USD)', max=15.000
1, min=0.4999, step=0.0001)
FloatSlider(value=1.0, continuous_update=False, description='Min House Age (year
s)', max=52.0, min=1.0, step=1…
FloatSlider(value=52.0, continuous_update=False, description='Max House Age (year
s)', max=52.0, min=1.0, step=…
FloatSlider(value=0.8461538461538461, continuous_update=False, description='Min Av
erage # of Rooms', max=141.9…
FloatSlider(value=141.9090909090909, continuous_update=False, description='Max Ave
rage # of Rooms', max=141.90…
FloatSlider(value=0.3333333333333333, continuous_update=False, description='Min Av
erage # of Bedrooms', max=34…
FloatSlider(value=34.06666666666667, continuous_update=False, description='Max Ave
rage # of Bedrooms', max=34.…
FloatSlider(value=3.0, continuous_update=False, description='Min Population (100k
people)', max=35682.0, min=3…
FloatSlider(value=35682.0, continuous_update=False, description='Max Population (1
00k people)', max=35682.0, m…
FloatSlider(value=0.6923076923076923, continuous_update=False, description='Min Av
erage Occupancy (# people)',…
FloatSlider(value=1243.3333333333333, continuous_update=False, description='Max Av
erage Occupancy (# people)',…
FloatSlider(value=32.54, continuous_update=False, description='Min Latitude (de
g)', max=41.95, min=32.54, step…
FloatSlider(value=41.95, continuous_update=False, description='Max Latitude (de
g)', max=41.95, min=32.54, step…
FloatSlider(value=-124.35, continuous_update=False, description='Min Longitude (de
g)', max=-114.31, min=-124.3…
FloatSlider(value=-114.31, continuous_update=False, description='Max Longitude (de
g)', max=-114.31, min=-124.3…
Output()
```



In [ ]:
```
from IPython.display import display
button = widg.Button(description="Click Me!")
output = widg.Output()

display(button, output)

def on_button_clicked(b):
    with output:
        print("Button clicked.")

button.on_click(on_button_clicked)
```

```
Button(description='Click Me!', style=ButtonStyle())
Output()
```