DATA SOCIETY:

Interactive Visualization with Bokeh - Interactive Plots - 4

One should look for what is and not what he thinks should be. (Albert Einstein)

Module completion checklist

Objective	Complete
Integrate widgets to bokeh and plotly graphs	
Save your graphs	

Adding widgets to graphs

- `ipywidgets` library allows us to turn Jupyter Notebooks from static documents into interactive dashboards
- Widgets are handy when we wish to **change inputs** without needing to rewrite or rerun code
- You can read the documentation on `ipywidgets` ***here***

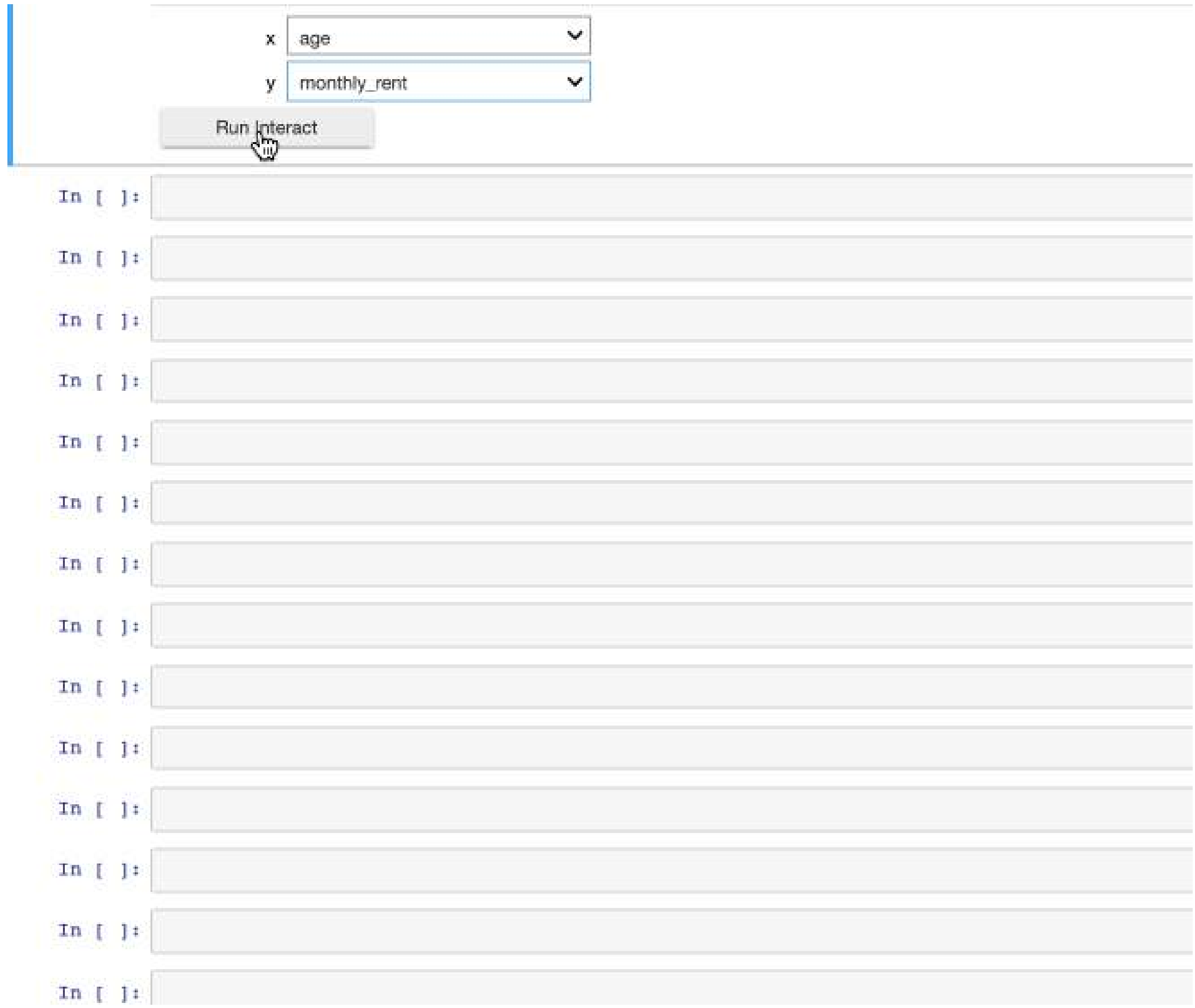
Adding widgets to graphs (cont'd)

- The Costa Rican dataset has many inputs which can slow the plot rendering, so let's subset the variables we wish to use

```
@interact_manual
def scatter_plot(x = list(costa_viz.columns), y = list(costa_viz.columns)):
    p = figure(title = f'{x} vs {y}',
               x_axis_label = x,
               y_axis_label = y)

    p.circle(x = x,
             y = y,
             source = costa_viz,
             size=20, color = "thistle", alpha = 0.2)
    show(p)
```

Adding widgets to graphs (cont’d)



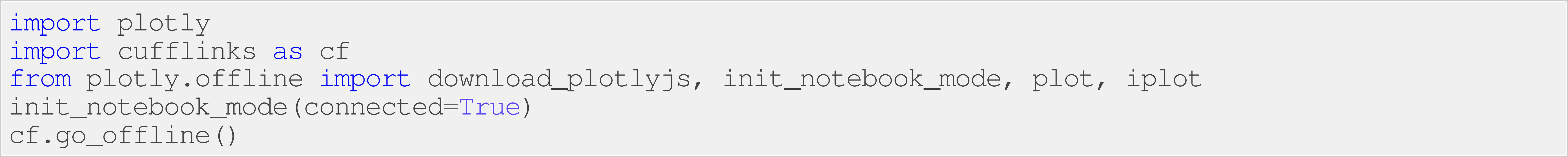
x age

y monthly_rent

```
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
```

Adding widgets to Plotly graphs

- We can add `colorscale` and `themes` to our widget in cufflinks graphs
- First, we need to run the initialization steps for offline plotting



```
import plotly
import cufflinks as cf
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
init_notebook_mode(connected=True)
cf.go_offline()
```

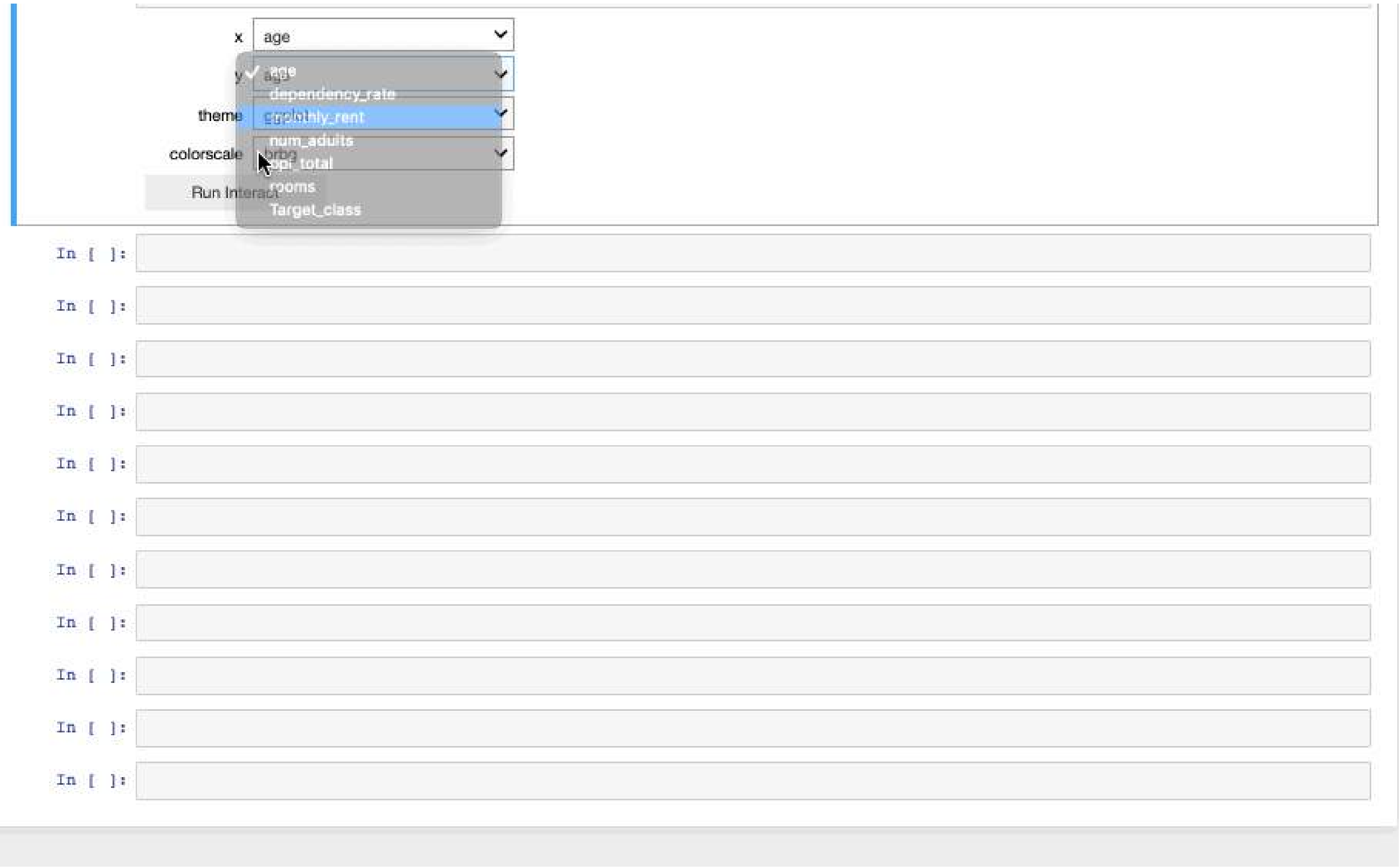
Adding widgets to Plotly graphs (cont’d)

- `@interact` is another common method which will automatically output the widgets and the plot, but may take longer to update
- We decided to use `@interact_manual` to prevent the notebook from freezing and to select all desired parameters before running the scatterplot

```
@interact_manual
def scatter_plot(x = list(costa_viz.columns), y = list(costa_viz.columns),
                 theme = list(cf.themes.THEMES.keys()),
                 colorscale = list(cf.colors._scales_names.keys())):

    costa_viz.iplot(kind = 'scatter', x = x, y = y, mode='markers',
             categories = 'Target_class',
             xTitle = x, yTitle = y,
             title = f'{x} vs {y}',
             theme = theme, colorscale = colorscale)
```

Adding widgets to Plotly graphs (cont'd)



x age

y age

theme ggplot

colorscale brbg

```
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
In [ ]:
```

Module completion checklist

Objective	Complete
Integrate widgets to bokeh and plotly graphs	✔
Save your graphs	

Saving your graph

- You will be saving all of your graphs as an `HTML` file in the plots folder

```
x_values = [1, 2, 3, 4, 5, 6]
y_values = [6, 7, 2, 3, 6, 4]

# Create figure.
p = figure(plot_width = 400, plot_height = 400)

# Add glyphs to it.
p.triangle(x_values, y_values, size = 20, color = "darkseagreen", alpha = 0.7)

# Save your plot.
output_file(plot_dir + "/bokeh-simple-plot.html", mode = 'inline')
save(p)
```

Plotly vs. Bokeh

- **Bokeh** is ideal for creating charts with multiple glyphs and has various options for customization
- But it undergoes a lot of development, hence the code we write today may change in the future
- **Plotly** syntax is also simple and can be embedded as an HTML in applications
- The main limitation of plotly is that the online community version plots are public and there is a limit of plots which can be created per day
- Choosing an interactive visualization package depends on your personal preference and ease of use!

Knowledge check

Link: ***Click here to complete the knowledge check***

Exercise

Module completion checklist

Objective	Complete
Integrate widgets to bokeh and plotly graphs	✔
Save your graphs	✔

Interactive Plots: Topic summary

In this part of the course, we have covered:

- Introduction to `bokeh` package
- Organize and visualize data with `bokeh`

This completes our module