



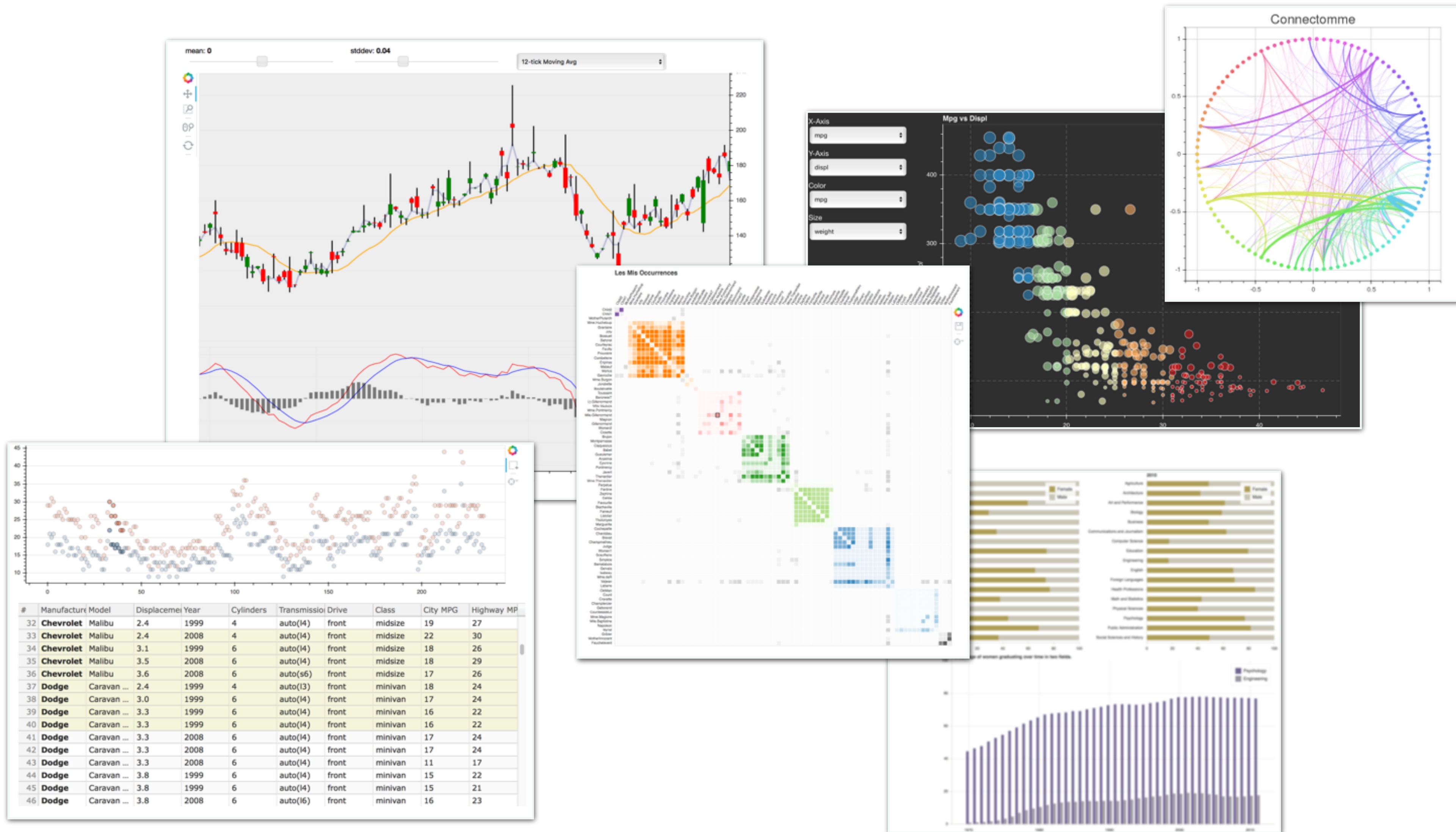
INTERACTIVE DATA VISUALIZATION WITH BOKEH

# **Interactive Data Visualization with Bokeh**



# What is Bokeh?

- Interactive visualization, controls, and tools
- Versatile and high-level graphics
- High-level statistical charts
- Streaming, dynamic, large data
- For the browser, with or without a server
- *No JavaScript*





# What you will learn

- Basic plotting with `bokeh.plotting`
- Layouts, interactions, and annotations
- Statistical charting with `bokeh.charts`
- Interactive data applications in the browser
- Case Study: A Gapminder explorer



INTERACTIVE DATA VISUALIZATION WITH BOKEH

**See you in  
the course!**



INTERACTIVE DATA VISUALIZATION WITH BOKEH

# Plotting with Glyphs



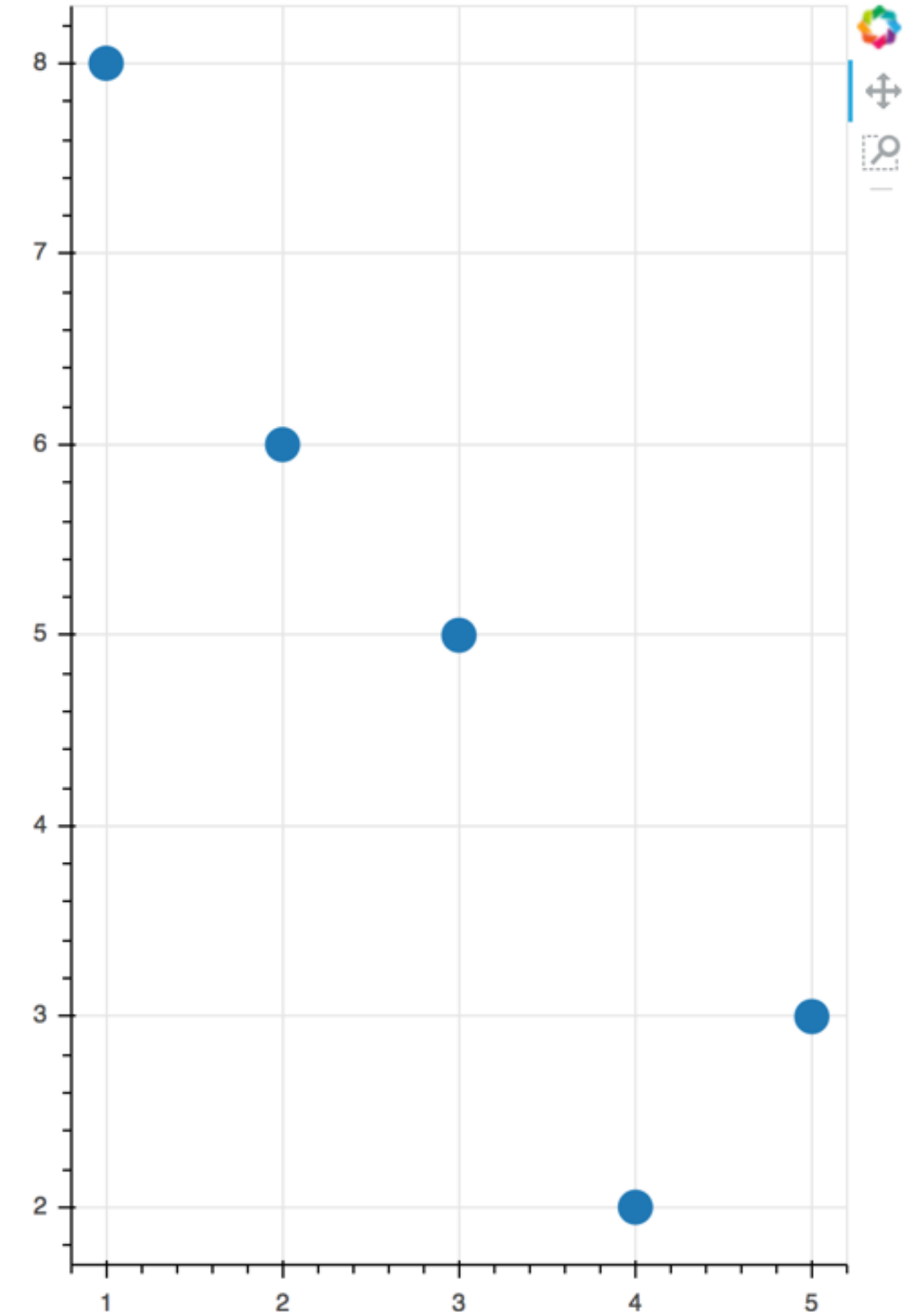


# What are Glyphs

- Visual shapes
  - circles, squares, triangles
  - rectangles, lines, wedges
- With properties attached to data
  - coordinates (x,y)
  - size, color, transparency

# Typical usage

```
In [1]: from bokeh.io import output_file, show  
  
In [2]: from bokeh.plotting import figure  
  
In [3]: plot = figure(plot_width=400, tools='pan,box_zoom')  
  
In [4]: plot.circle([1,2,3,4,5], [8,6,5,2,3])  
  
In [5]: output_file('circle.html')  
  
In [6]: show(plot)
```



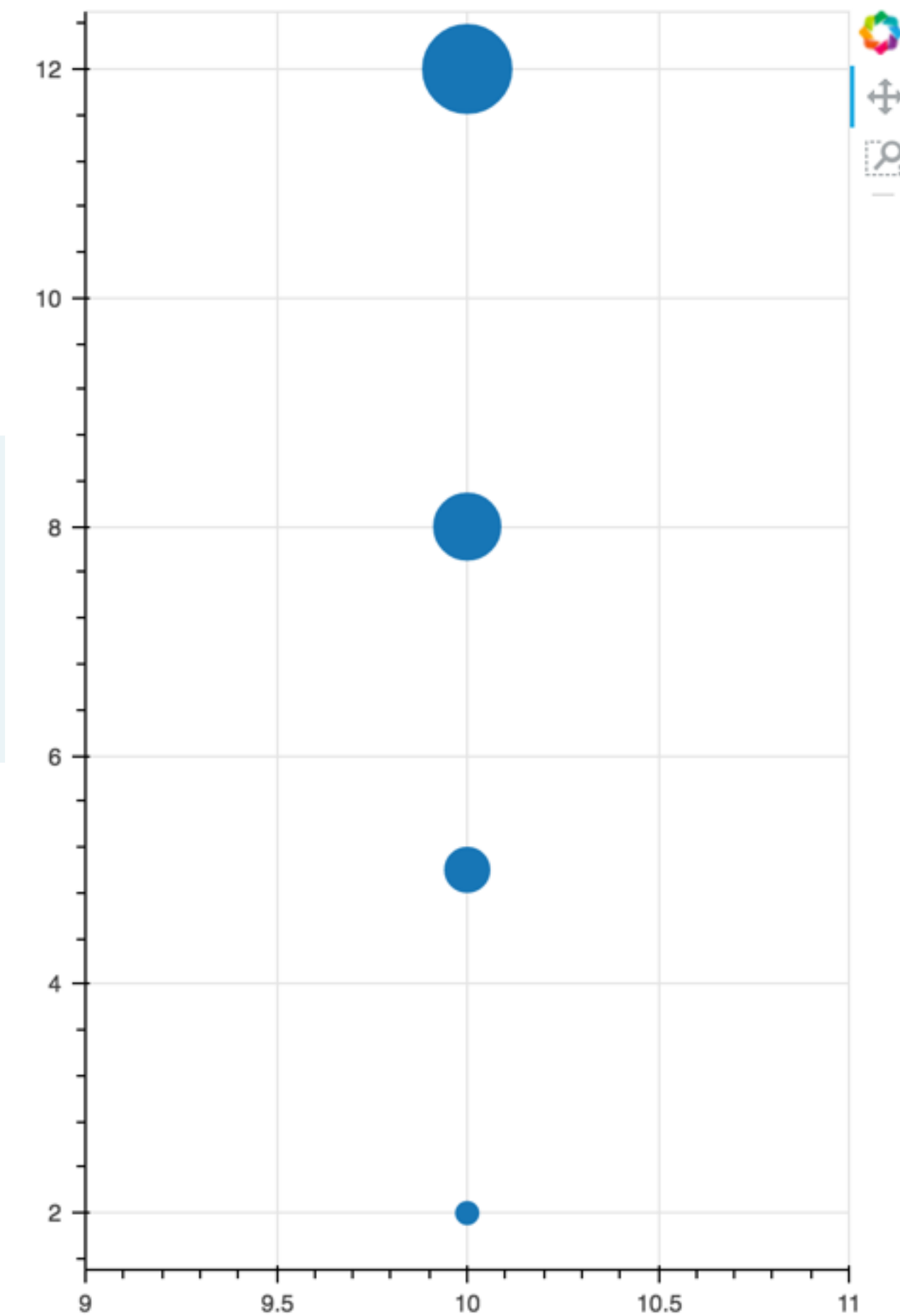


# Glyph properties

- Lists, arrays, sequences of values
- Single fixed values

```
In [1]: plot = figure()
```

```
In [2]: plot.circle(x=10, y=[2,5,8,12], size=[10,20,30,40])
```





# Markers

- `asterisk()`
- `circle()`
- `circle_cross()`
- `circle_x()`
- `cross()`
- `diamond()`
- `diamond_cross()`
- `inverted_triangle()`
- `square()`
- `square_cross()`
- `square_x()`
- `triangle()`
- `x()`



INTERACTIVE DATA VISUALIZATION WITH BOKEH

**Let's practice!**



INTERACTIVE DATA VISUALIZATION WITH BOKEH

# **Additional Glyphs**

# Lines

```
In [1]: from bokeh.io import output_file, show
```

```
In [2]: from bokeh.plotting import figure
```

```
In [3]: x = [1,2,3,4,5]
```

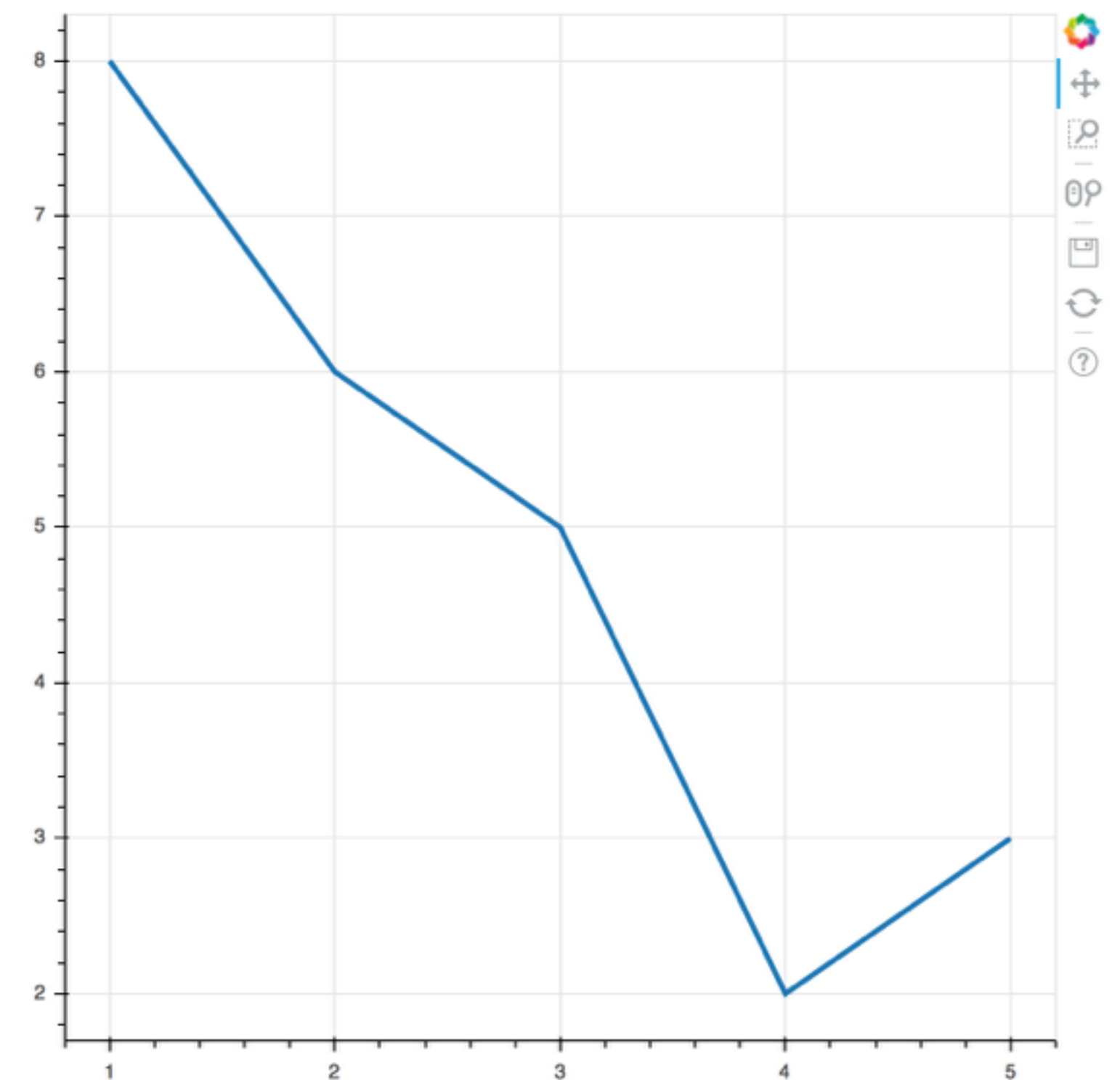
```
In [4]: y = [8,6,5,2,3]
```

```
In [5]: plot = figure()
```

```
In [6]: plot.line(x, y, line_width=3)
```

```
In [7]: output_file('line.html')
```

```
In [8]: show(plot)
```



# Lines and Markers Together

```
In [1]: from bokeh.io import output_file, show
```

```
In [2]: from bokeh.plotting import figure
```

```
In [3]: x = [1,2,3,4,5]
```

```
In [4]: y = [8,6,5,2,3]
```

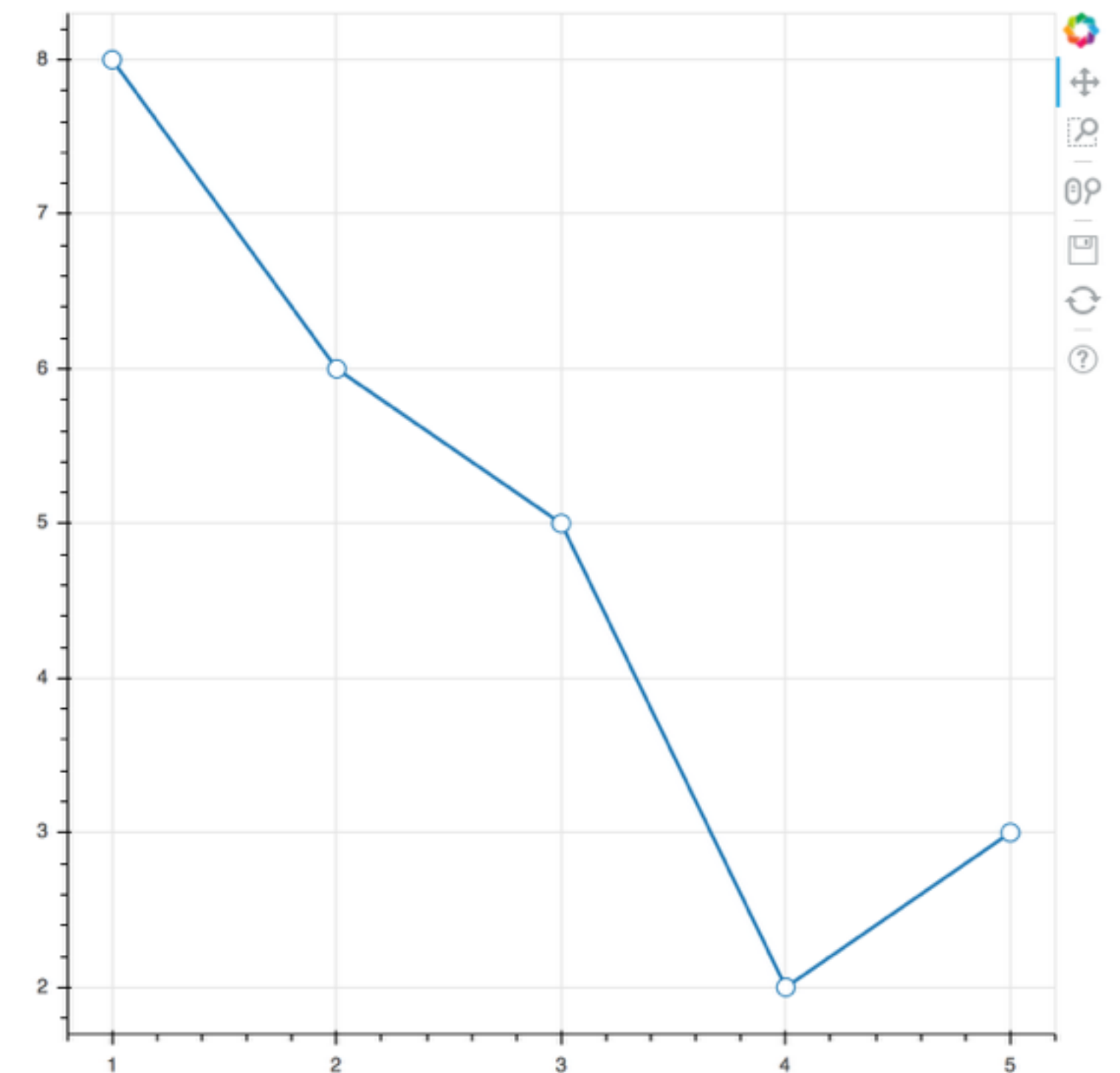
```
In [5]: plot = figure()
```

```
In [6]: plot.line(x, y, line_width=2)
```

```
In [7]: plot.circle(x, y, fill_color='white', size=10)
```

```
In [8]: output_file('line.html')
```

```
In [9]: show(plot)
```





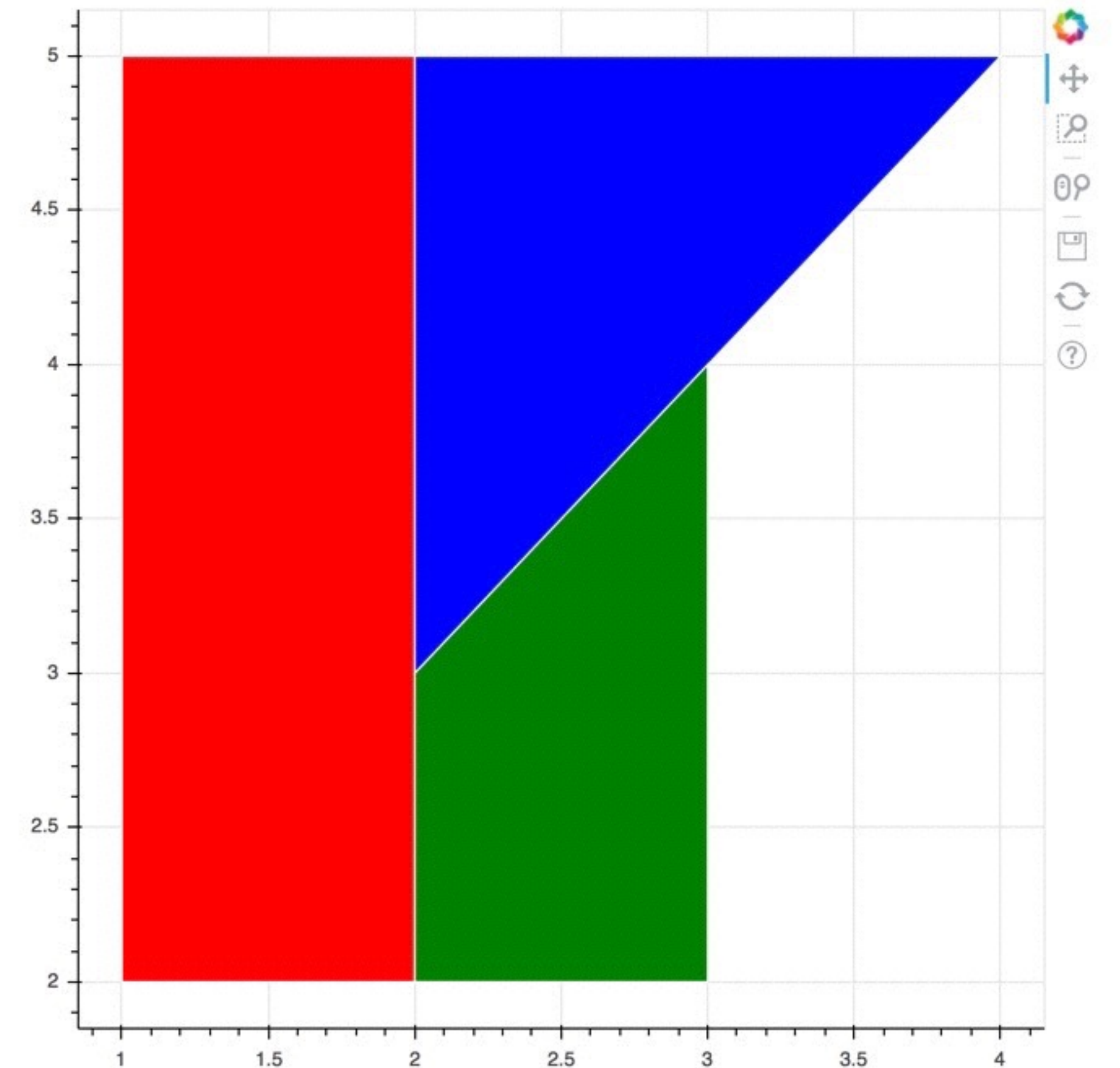
# Patches

- Useful for showing geographic regions
- Data given as “list of lists”



# Patches

```
In [1]: from bokeh.io import output_file, show
In [2]: from bokeh.plotting import figure
In [3]: xs = [ [1,1,2,2], [2,2,4], [2,2,3,3] ]
In [4]: ys = [ [2,5,5,2], [3,5,5], [2,3,4,2] ]
In [5]: plot = figure()
In [6]: plot.patches(xs, ys,
...:                 fill_color=
...:                 ['red', 'blue', 'green'],
...:                 line_color='white')
In [7]: output_file('patches.html')
In [8]: show(plot)
```





# Other glyphs

- `annulus()`
- `annular_wedge()`
- `wedge()`
- `rect()`
- `quad()`
- `vbar()`
- `hbar()`
- `image()`
- `image_rgba()`
- `image_url()`
- `patch()`
- `patches()`
- `line()`
- `multi_line()`
- `circle()`
- `oval()`
- `ellipse()`
- `arc()`
- `quadratic()`
- `bezier()`



INTERACTIVE DATA VISUALIZATION WITH BOKEH

**Let's practice!**



INTERACTIVE DATA VISUALIZATION WITH BOKEH

# Data Formats

# Python Basic Types

```
In [1]: from bokeh.io import output_file, show

In [2]: from bokeh.plotting import figure

In [3]: x = [1,2,3,4,5]

In [4]: y = [8,6,5,2,3]

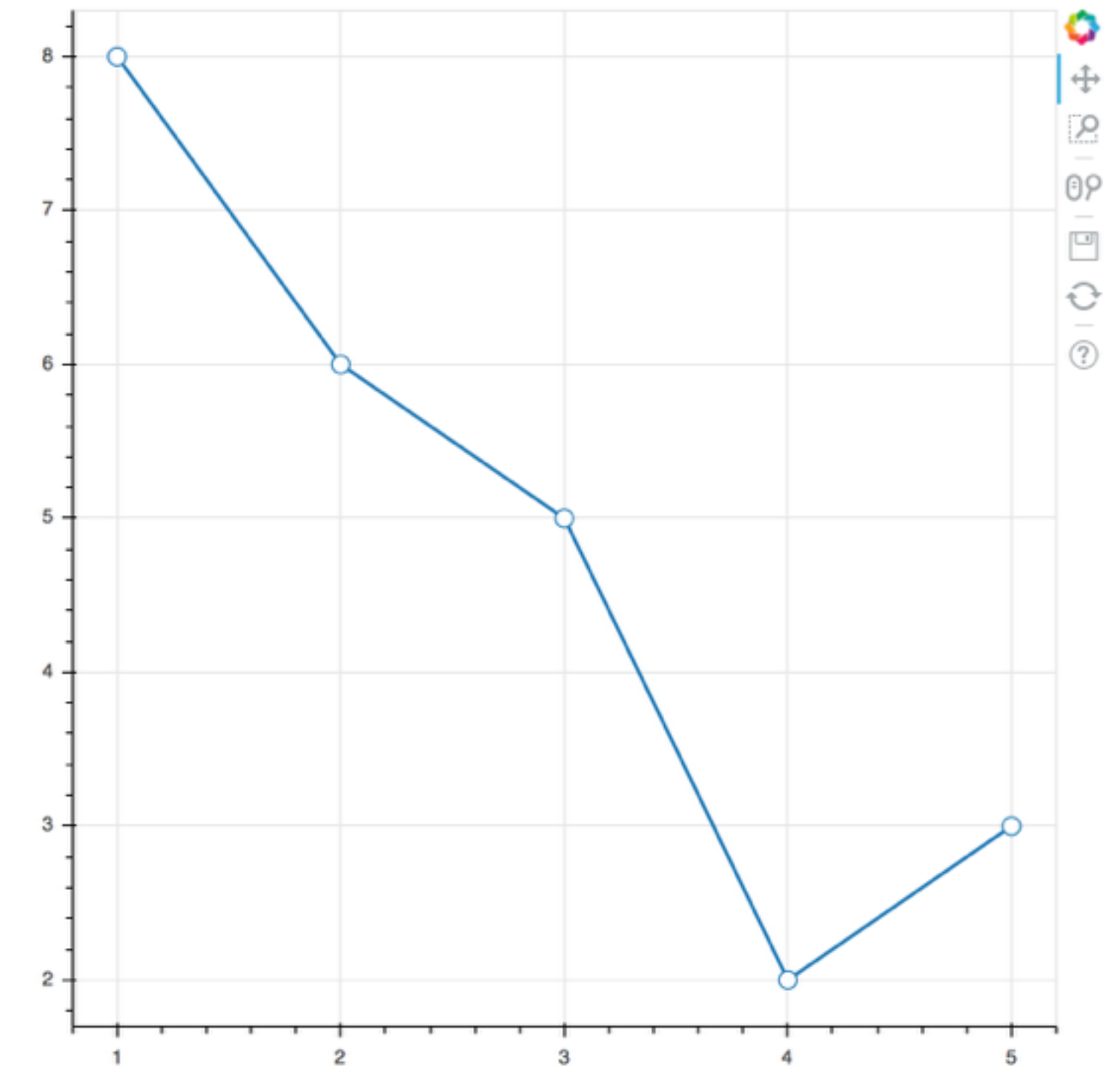
In [5]: plot = figure()

In [6]: plot.line(x, y, line_width=3)

In [7]: plot.circle(x, y, fill_color='white', size=10)

In [8]: output_file('basic.html')

In [9]: show(plot)
```



# NumPy Arrays

```
In [1]: from bokeh.io import output_file, show
```

```
In [2]: from bokeh.plotting import figure
```

```
In [3]: import numpy as np
```

```
In [4]: x = np.linspace(0, 10, 1000)
```

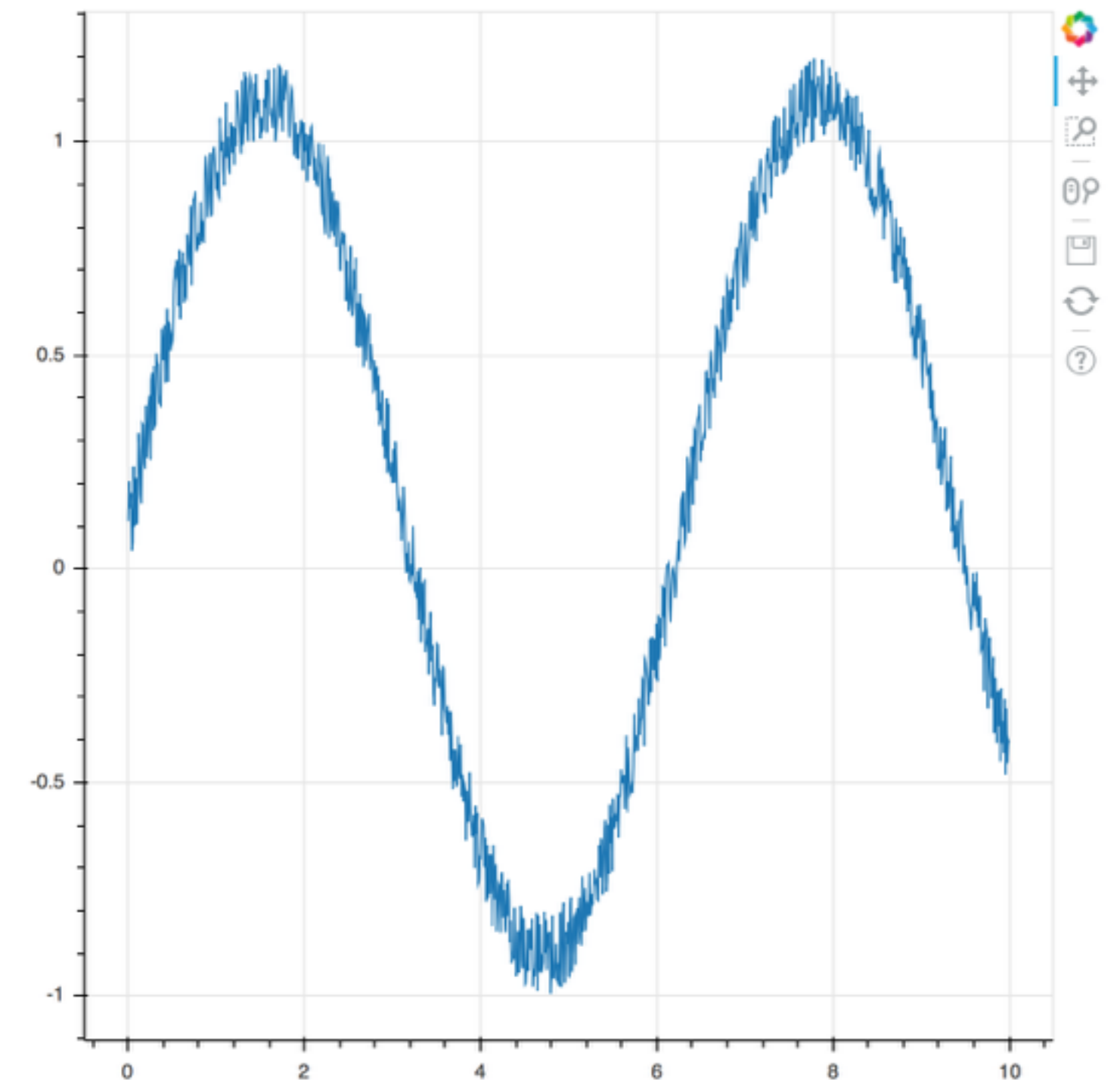
```
In [5]: y = np.sin(x) + np.random.random(1000) * 0.2
```

```
In [6]: plot = figure()
```

```
In [7]: plot.line(x, y)
```

```
In [8]: output_file('numpy.html')
```

```
In [9]: show(plot)
```



# Pandas

```
In [1]: from bokeh.io import output_file, show

In [2]: from bokeh.plotting import figure

In [3]: # Flowers is a Pandas DataFrame

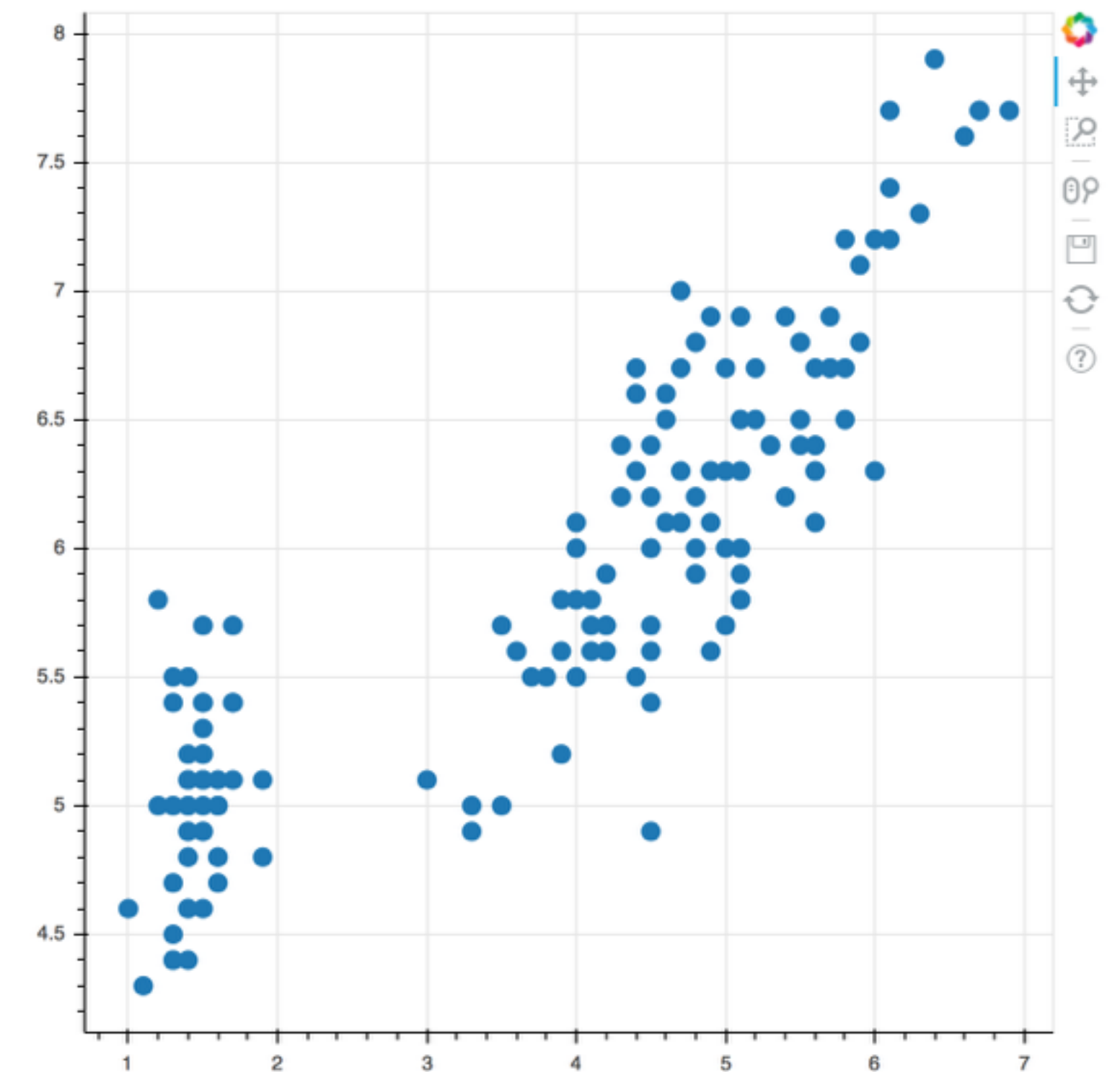
In [4]: from bokeh.sampledata.iris import flowers

In [5]: plot = figure()

In [6]: plot.circle(flowers['petal_length'],
...:                flowers['sepal_length'],
...:                size=10)

In [7]: output_file('pandas.html')

In [8]: show(plot)
```







# Column Data Source

- Common fundamental data structure for Bokeh
- Maps string column names to sequences of data
- Often created automatically for you
- Can be shared between glyphs to link selections
- Extra columns can be used with hover tooltips



# Column Data Source

```
In [1]: from bokeh.models import ColumnDataSource
```

```
In [2]: source = ColumnDataSource(data={  
...:                                     'x': [1,2,3,4,5],  
...:                                     'y': [8,6,5,2,3]})
```

```
In [3]: source.data
```

```
Out[3]: {'x': [1, 2, 3, 4, 5], 'y': [8, 6, 5, 2, 3]}
```



# Column Data Source

```
In [1]: from bokeh.models import ColumnDataSource
```

```
In [2]: from bokeh.sampledata.iris import flowers as df
```

```
In [3]: df.head()
```

```
Out[3]:
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

```
In [4]: source = ColumnDataSource(df)
```



INTERACTIVE DATA VISUALIZATION WITH BOKEH

**Let's practice!**



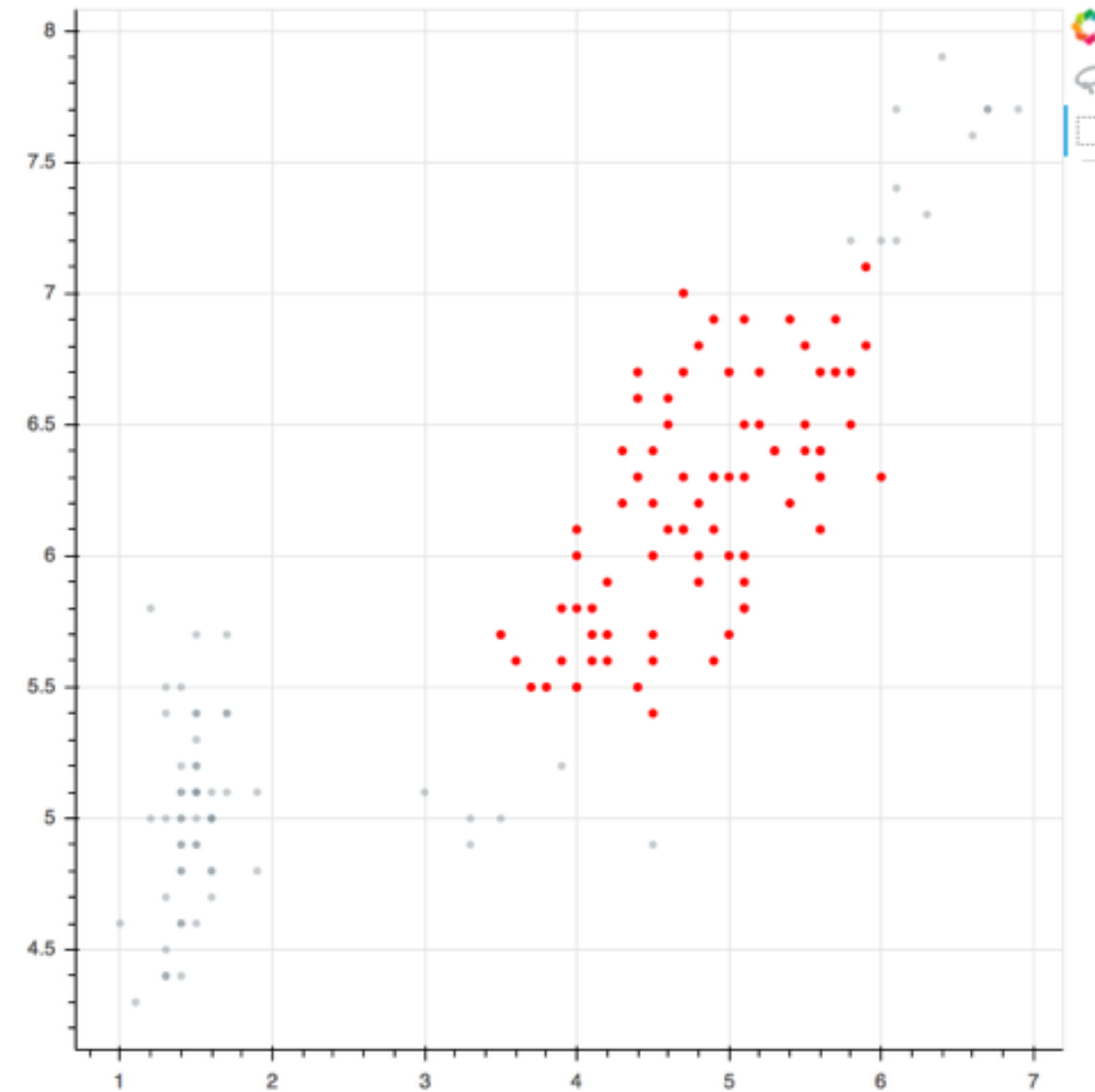
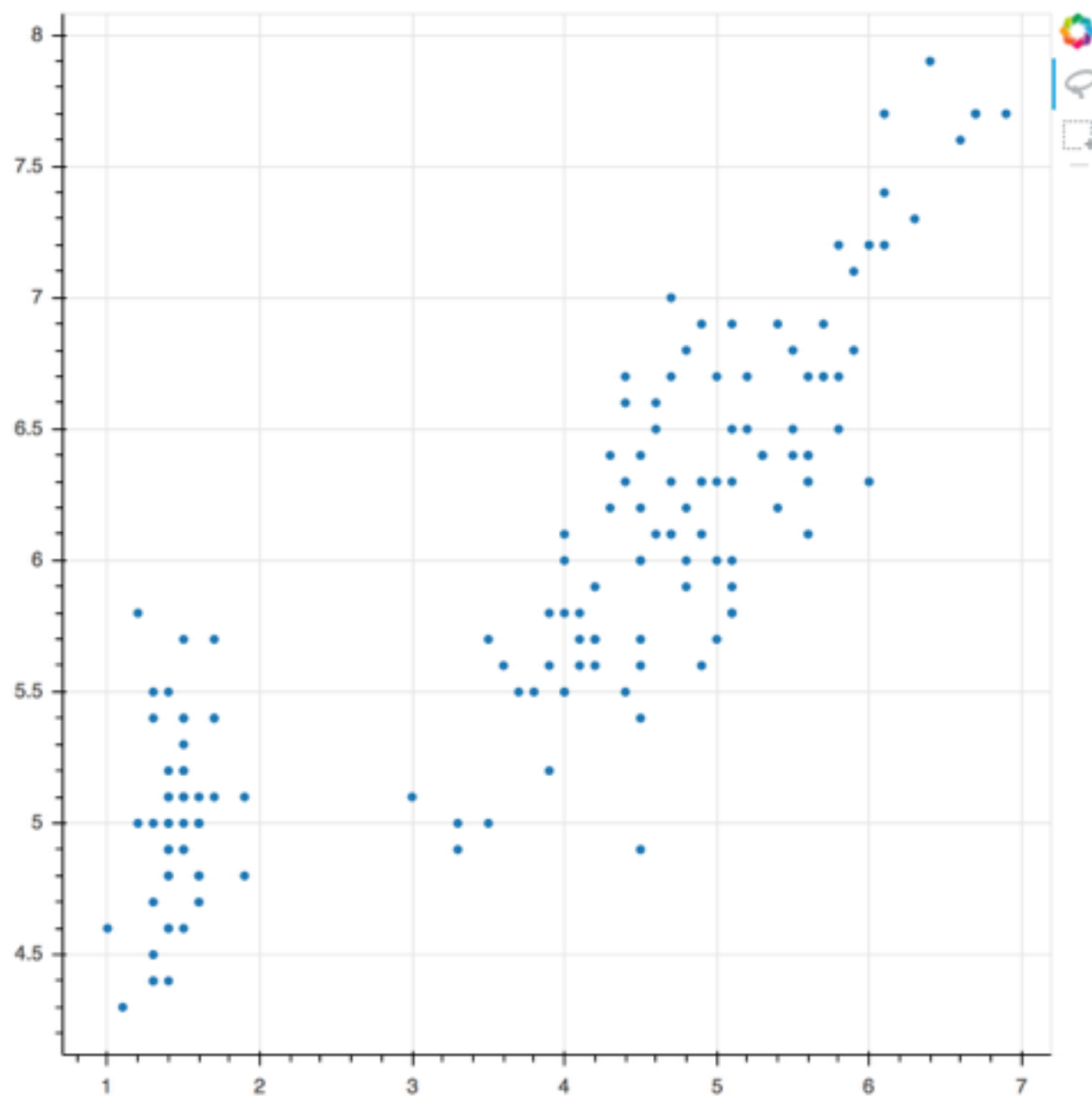
INTERACTIVE DATA VISUALIZATION WITH BOKEH

# Customizing Glyphs

# Selection appearance

```
In [1]: plot = figure(tools='box_select, lasso_select')
```

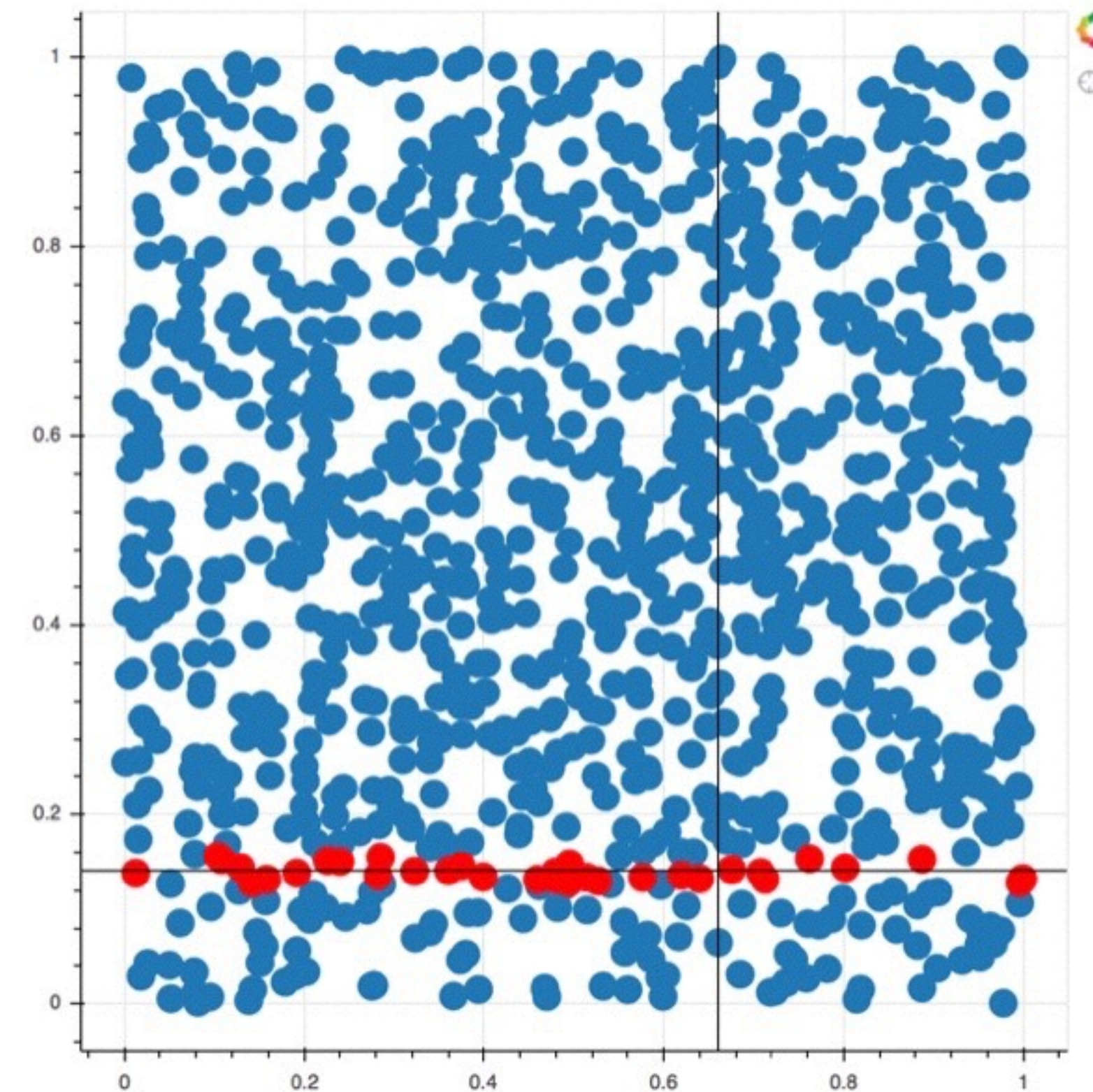
```
In [2]: plot.circle(petal_length, sepal_length,  
...:               selection_color='red',  
...:               nonselection_fill_alpha=0.2,  
...:               nonselection_fill_color='grey')
```





# Hover appearance

```
In [1]: from bokeh.models import HoverTool  
In [2]: hover = HoverTool(tooltips=None, mode='hline')  
In [3]: plot = figure(tools=[hover, 'crosshair'])  
In [4]: # x and y are lists of random points  
In [5]: plot.circle(x, y, size=15, hover_color='red')
```







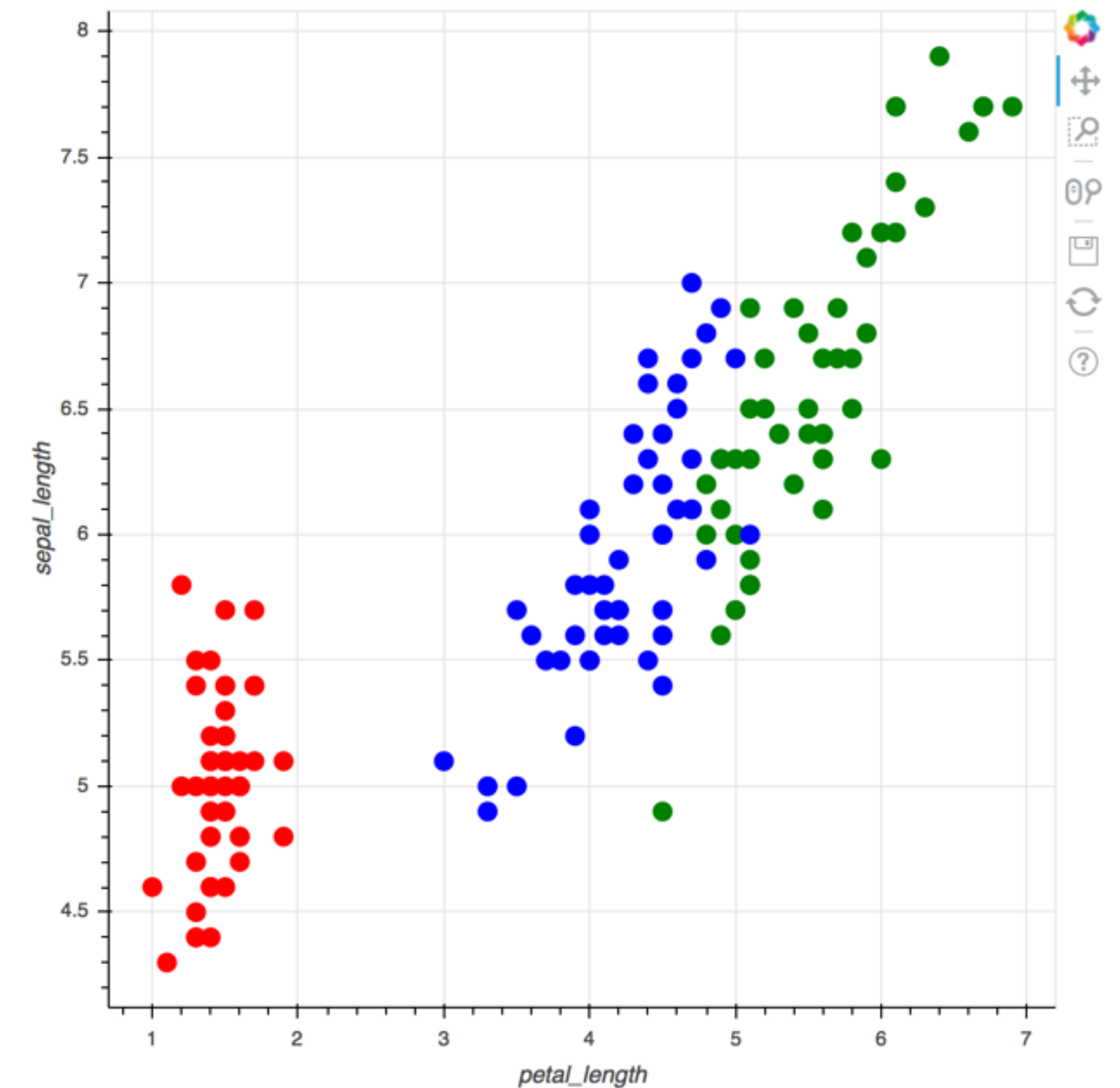
# Color mapping

```
In [1]: from bokeh.models import CategoricalColorMapper
```

```
In [2]: mapper = CategoricalColorMapper(  
    ...:     factors=['setosa', 'virginica',  
    ...:             'versicolor'],  
    ...:     palette=['red', 'green', 'blue'])
```

```
In [3]: plot = figure(x_axis_label='petal_length',  
    ...:               y_axis_label='sepal_length')
```

```
In [4]: plot.circle('petal_length', 'sepal_length',  
    ...:             size=10, source=source,  
    ...:             color={'field': 'species',  
    ...:                   'transform': mapper})
```





INTERACTIVE DATA VISUALIZATION WITH BOKEH

**Let's practice!**