IMD0105 - Special Issues in Information Technology VI

Interactive Data Visualization with Bokeh

Natal-RN
March 2017

Previously on last class (...)

Agenda

Part I

- Introduction about Bokeh
- Glyphs
- Lines
- Plot (numpy, pandas)
- Column data source

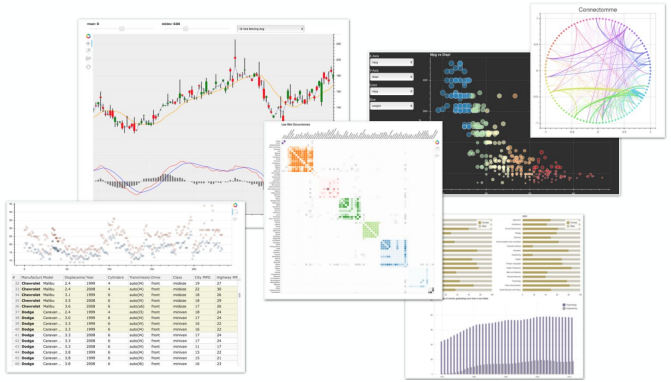
Part II

- Layout
- Rows of plot
- Columns of plot
- Nested plots
- Grid plots
- Tabbed plots
- Links selections
- Legends

What is Bokeh?

http://bokeh.pydata.org/en/latest/

- 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



Plotting with Glyphs

What are Glyphs

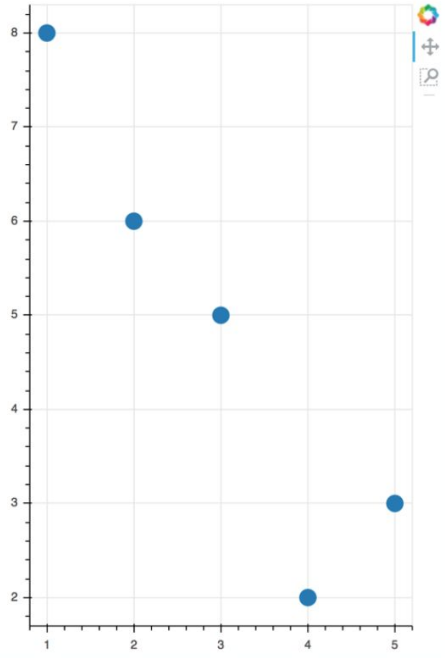
- visual shapes
- circles, squares, triangles
- rectangles, lines, wedges

With properties attached to data

- coordinates (x,y)
- size, color, transparency

Typical usage

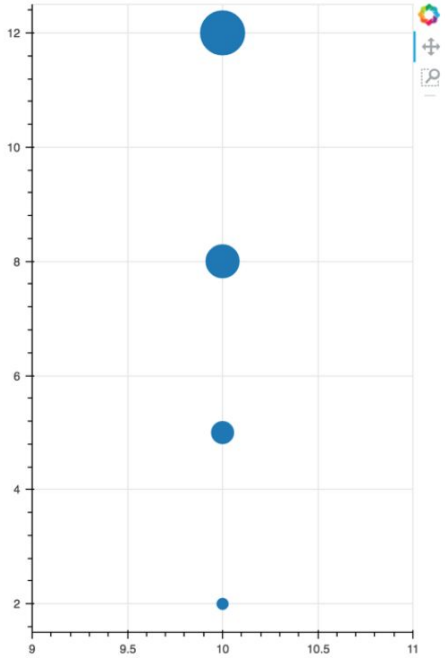
```
In [1]: from bokeh.io import output_file, show

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])
```

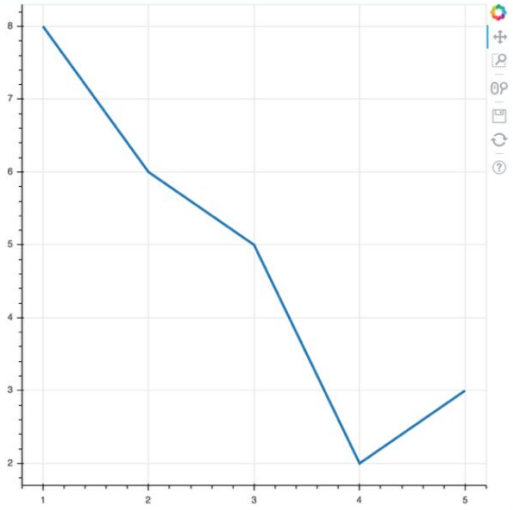


Markers

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

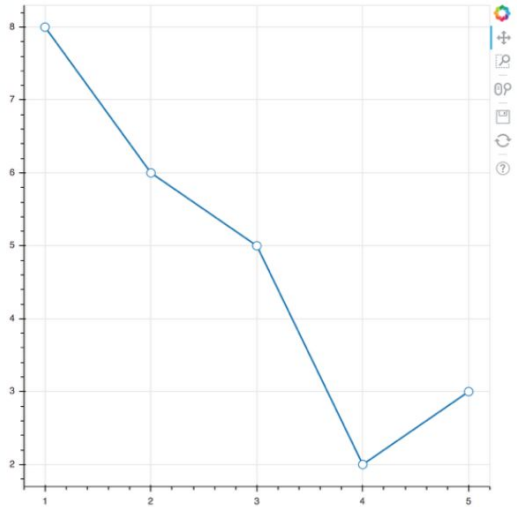
Lines

```
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)
```



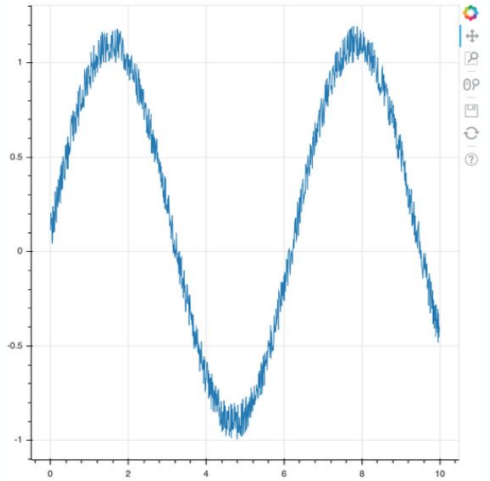
Lines and markers together

```
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)
```



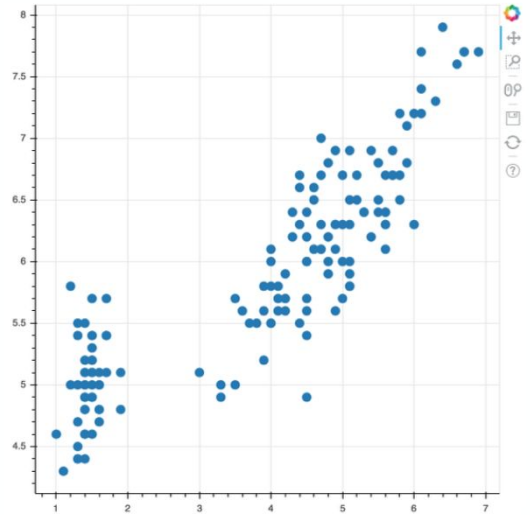
Numpy arrays

```
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]}
```

Introduction to Layout

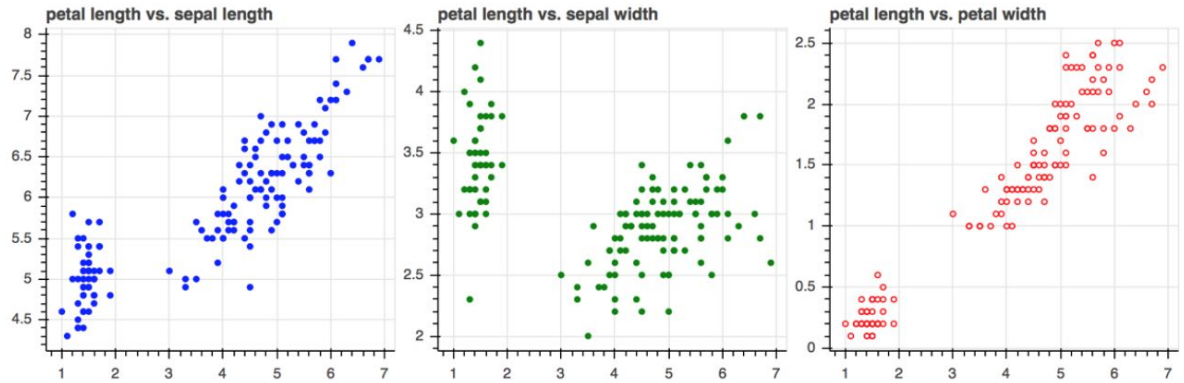
Arranging multiple plots

Arrange plots (and controls) visually on a page:

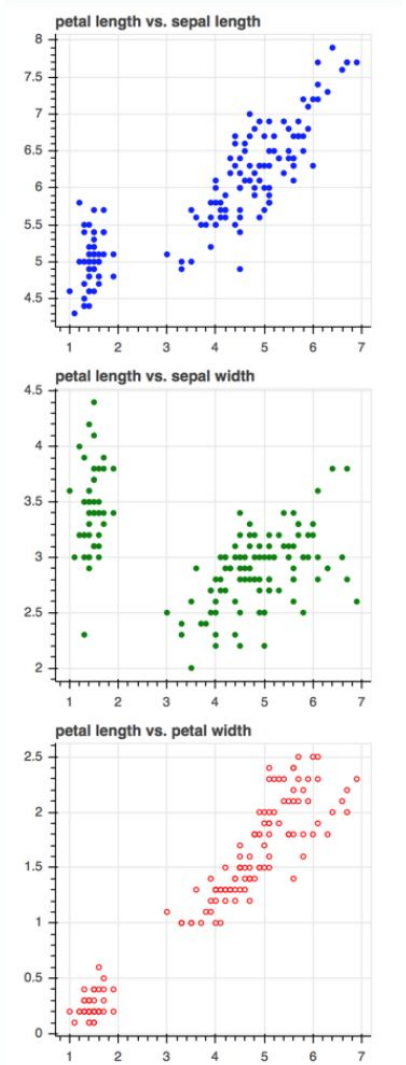
- rows, columns
- grid arrangements
- tabbed layouts

Rows of plot

```
In [1]: from bokeh.layouts import row

In [2]: layout = row(p1, p2, p3)

In [3]: output_file('row.html')

In [4]: show(layout)
```



Columns of plot

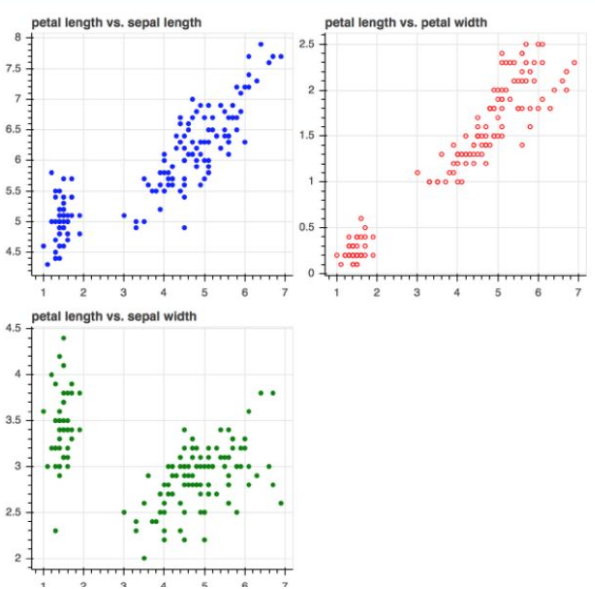



```
In [1]: from bokeh.layouts import column

In [2]: layout = column(p1, p2, p3)

In [3]: output_file('column.html')

In [4]: show(layout)
```

Nested plots

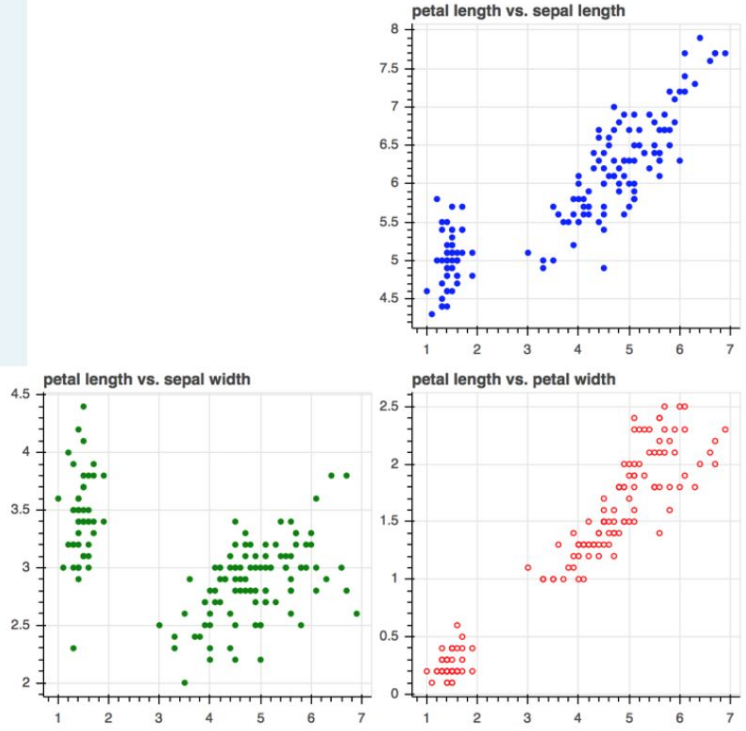
- Rows and column can be nested for more sophisticated layouts

```
In [1]: from bokeh.layouts import column, row

In [2]: layout = row(column(p1, p2), p3)

In [3]: output_file('nested.html')

In [4]: show(layout)
```



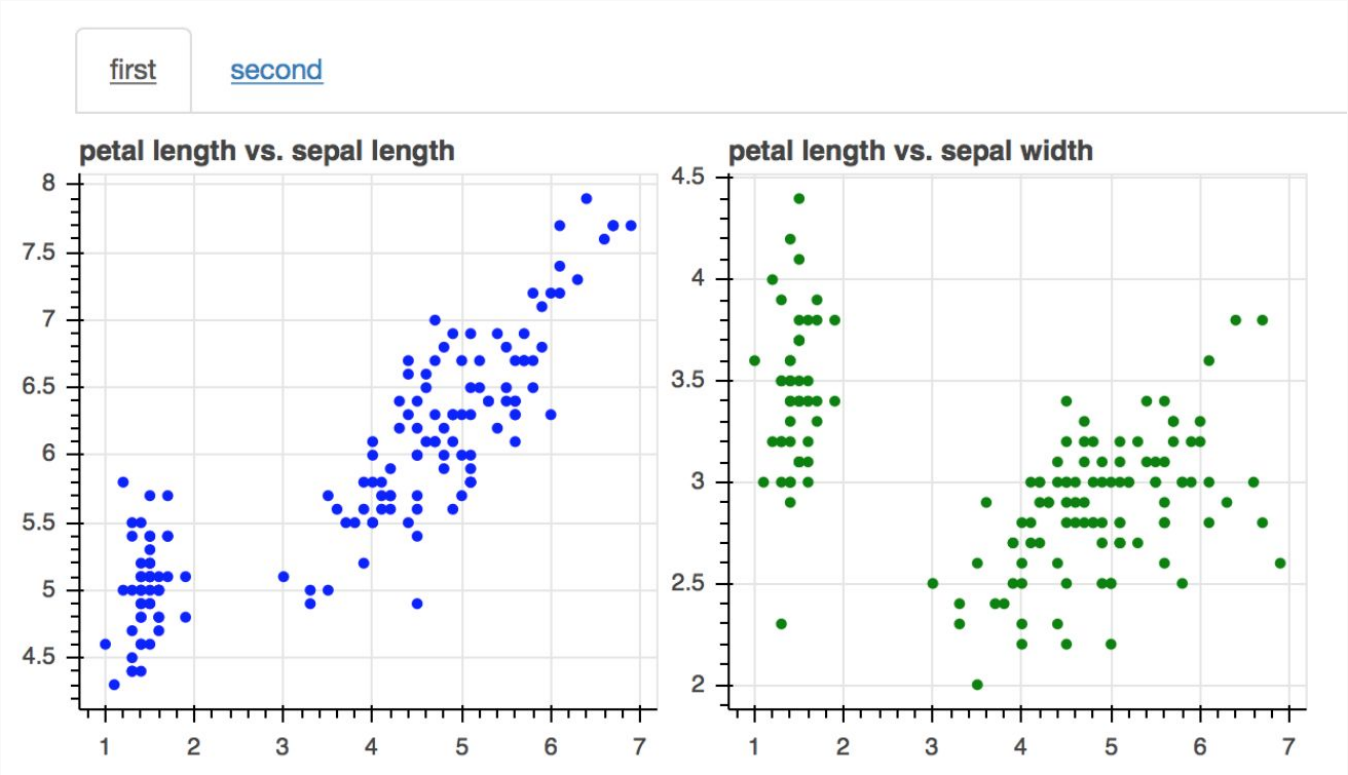
Gridplots

```
In [1]: from bokeh.layouts import gridplot

In [2]: layout = gridplot([[None, p1], [p2, p3]],
   ...:                   toolbar_location=None)

In [3]: output_file('nested.html')

In [4]: show(layout)
```



- Give a “list of rows” for layout
- can use None as a placeholder
- Accepts toolbar_location

Tabbed Layout

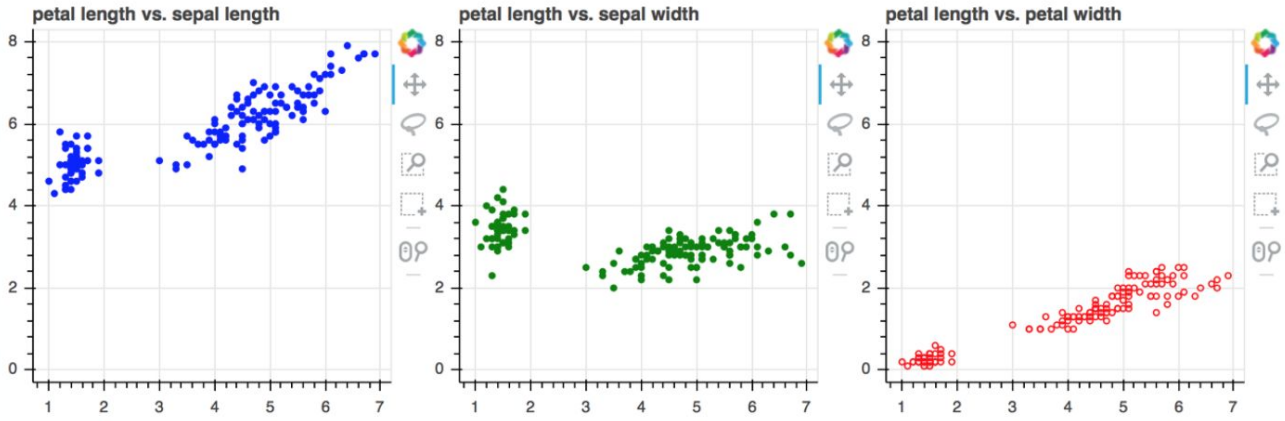


Tabbed Layout

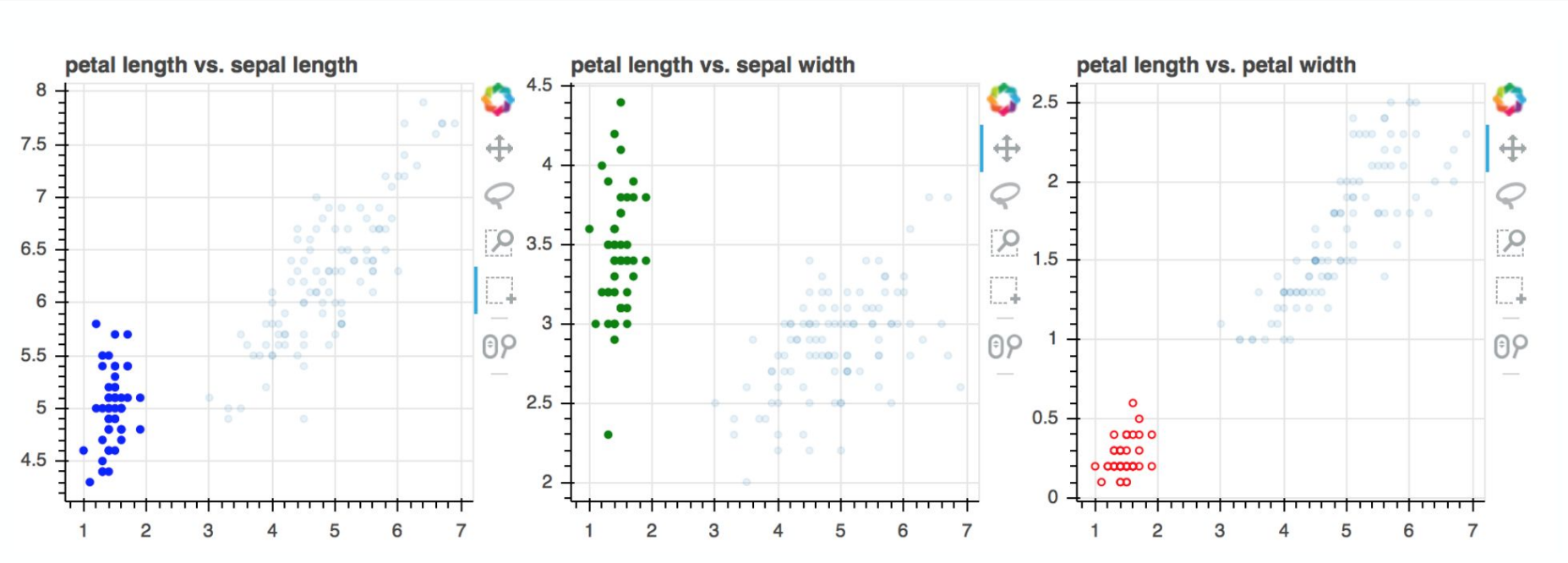
```
In [1]: from bokeh.models.widgets import Tabs, Panel

In [2]: # Create a Panel with a title for each tab

In [3]: first = Panel(child=row(p1, p2), title='first')

In [4]: second = Panel(child=row(p3), title='second')

In [5]: # Put the Panels in a Tabs object

In [6]: tabs = Tabs(tabs=[first, second])

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

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

Linking axes

```
In [1]: p3.x_range = p2.x_range = p1.x_range

In [2]: p3.y_range = p2.y_range = p1.y_range
```



Link selections

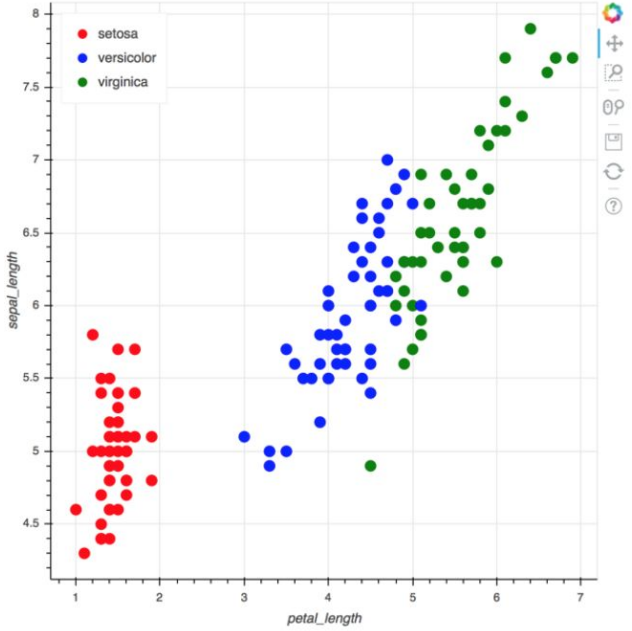


Link selections

```
In [1]: p1 = figure(title='petal length vs. sepal length')

In [2]: p1.circle('petal_length', 'sepal_length',
   ...:           color='blue', source=source)

In [3]: p2 = figure(title='petal length vs. sepal width')

In [4]: p2.circle('petal_length', 'sepal_width',
   ...:           color='green', source=source)

In [5]: p3 = figure(title='petal length vs. petal width')

In [6]: p3.circle('petal_length', 'petal_width',
   ...:           line_color='red', fill_color=None,
   ...:           source=source)
```

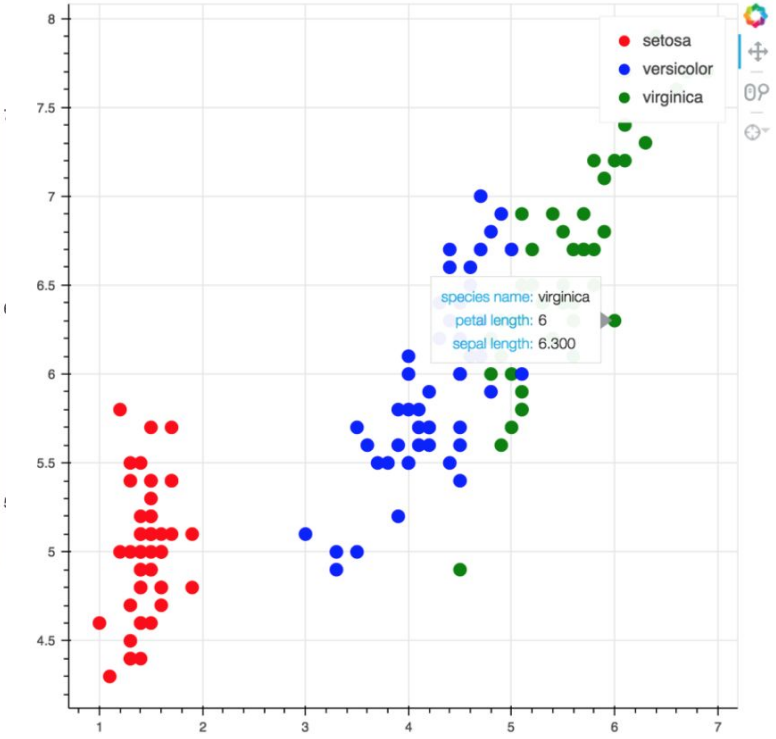
Legends

```
In [1]: plot.circle('petal_length', 'sepal_length',
   ...:             size=10, source=source,
   ...:             color={'field': 'species',
   ...:             'transform': mapper},
   ...:             legend='species')

In [2]: plot.legend.location = 'top_left'
```



Hover tips

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

In [2]: hover = HoverTool(tooltips=[
   ...:          ('species name', '@species'),
   ...:          ('petal length', '@petal_length'),
   ...:          ('sepal length', '@sepal_length'),
   ...:          ])

In [3]: plot = figure(tools=[hover, 'pan',
   ...:                 'wheel_zoom'])
```



References

http://bokeh.pydata.org/en/latest/