Python For Data Science Cheat Sheet

Bokeh

Learn Bokeh Interactively at www.DataCamp.com, taught by Bryan Van de Ven, core contributor

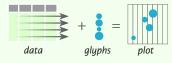


Plotting With Bokeh

The Python interactive visualization library Bokeh enables high-performance visual presentation of large datasets in modern web browsers.



Bokeh's mid-level general purpose bokeh.plotting interface is centered around two main components: data and glyphs.



The basic steps to creating plots with the bokeh.plotting interface are:

1. Prepare some data:

Python lists, NumPy arrays, Pandas DataFrames and other sequences of values

- 2. Create a new plot
- 3. Add renderers for your data, with visual customizations
- 4. Specify where to generate the output
- 5. Show or save the results

```
>>> from bokeh.plotting import figure
>>> from bokeh.io import output file, show
>>> x = [1, 2, 3, 4, 5]
>>> y = [6, 7, 2, 4, 5]
>>> p = figure(title="simple line example",
              x axis label='x',
              y axis label='y')
>>> p.line(x, y, legend="Temp.", line width=2) < Step 3
>>> output_file("lines.html") < Step 4
>>> show(p) < Step 5
```

Data

Also see Lists, NumPy & Pandas

Under the hood, your data is converted to Column Data Sources. You can also do this manually:

```
>>> import numpy as np
>>> import pandas as pd
>>> df = pd.DataFrame(np.array([[33.9,4,65, 'US'],
                                     [32.4,4,66, 'Asia'],
                                     [21.4,4,109, 'Europe']]),
                        columns=['mpg','cyl', 'hp', 'origin'],
index=['Toyota', 'Fiat', 'Volvo'])
>>> from bokeh.models import ColumnDataSource
>>> cds df = ColumnDataSource(df)
```

Plottina

```
>>> from bokeh.plotting import figure
>>> p1 = figure(plot width=300, tools='pan,box zoom')
>>> p2 = figure(plot width=300, plot height=300,
               x range=(0, 8), y range=(0, 8))
>>> p3 = figure()
```

Renderers & Visual Customizations

```
Glyphs
```

```
Scatter Markers
>>> p1.circle(np.array([1,2,3]), np.array([3,2,1]),
            fill color='white')
>>> p2.square(np.array([1.5,3.5,5.5]), [1,4,3],
            color='blue', size=1)
```

Line Glyphs

```
>>> p1.line([1,2,3,4], [3,4,5,6], line_width=2)
>>> p2.multi line(pd.DataFrame([[1,2,3],[5,6,7]]),
                 pd.DataFrame([[3,4,5],[3,2,1]]),
                 color="blue")
```

Rows & Columns Layout

>>>layout = row(column(p1,p2), p3)

>>> from bokeh.layouts import gridplot

>>> layout = gridplot([[p1,p2],[p3]])

>>> tab1 = Panel(child=p1, title="tab1")

>>> tab2 = Panel(child=p2, title="tab2")

>>> from bokeh.models.widgets import Panel, Tabs

```
Columns
>>> from bokeh.layouts import row|>>> from bokeh.layouts import columns
>>> layout = row(p1,p2,p3)
                                  >>> layout = column(p1,p2,p3)
Nesting Rows & Columns
```

Linked Plots

Linked Axes

>>> p2.x range = p1.x range >>> p2.y range = p1.y range

Linked Brushing

```
>>> p4 = figure(plot width = 100, tools='box select, lasso select')
>>> p4.circle('mpg', 'cyl', source=cds df)
>>> p5 = figure(plot width = 200, tools='box select, lasso select')
>>> p5.circle('mpg', 'hp', source=cds df)
>>> layout = row(p4,p5)
```

Customized Glyphs

Hover Glyphs

Colormapping

>>> p3.add tools(hover)

Selection and Non-Selection Glyphs

>>> p.circle('mpg', 'cyl', source=cds df,

>>> color mapper = CategoricalColorMapper(

>>> p3.circle('mpg', 'cyl', source=cds df,

selection color='red',

nonselection alpha=0.1)

>>> hover = HoverTool(tooltips=None, mode='vline')

color=dict(field='origin',

factors=['US', 'Asia', 'Europe'],

palette=['blue', 'red', 'green'])

transform=color mapper),

legend='Origin'))

>>> p = figure(tools='box select')

Leaends

Grid Lavout

>>> row2 = [p3]

>>> row1 = [p1,p2]

Tabbed Lavout

```
Inside Plot Area
>>> p.legend.location = 'bottom left'
```

>>> layout = Tabs(tabs=[tab1, tab2])

Outside Plot Area

>>> r1 = p2.asterisk(np.array([1,2,3]), np.array([3,2,1]) >>> r2 = p2.line([1,2,3,4], [3,4,5,6]) >>> legend = Legend(items=[("One", [p1, r1]),("Two", [r2])], location=(0, -30)) >>> p.add layout(legend, 'right')

Output

Output to HTML File

>>> from bokeh.io import output file, show >>> output file('my bar chart.html', mode='cdn')

Notebook Output

>>> from bokeh.io import output notebook, show >>> output notebook()

Embedding

Standalone HTML

>>> from bokeh.embed import file html >>> html = file html(p, CDN, "my plot")

>>> from bokeh.embed import components

>>> script, div = components(p)

Show or Save Your Plots

>>> show(p	1)	>>>	save(p1)
>>> show(l	ayout)	>>>	save(layout)

Legend Orientation

```
>>> p.legend.orientation = "horizontal"
>>> p.legend.orientation = "vertical"
```

Legend Background & Border

```
>>> p.legend.border line color = "navy"
>>> p.legend.background fill color = "white"
```

Statistical Charts With Bokeh

Bokeh's high-level bokeh. charts interface is ideal for quickly creating statistical charts

Bar Chart



>>> from bokeh.charts import Bar >>> p = Bar(df, stacked=True, palette=['red','blue'])

Box Plot



| >>> from bokeh.charts import BoxPlot

>>> p = BoxPlot(df, values='vals', label='cyl', legend='bottom right')

Histogram



>>> from bokeh.charts import Histogram >>> p = Histogram(df, title='Histogram')

Scatter Plot



>>> from bokeh.charts import Scatter >>> p = Scatter(df, x='mpg', y ='hp', marker='square', xlabel='Miles Per Gallon', ylabel='Horsepower')

DataCamp Learn Python for Data Science Interactively



Also see Data