Introducing the Bokeh Server

INTERACTIVE DATA VISUALIZATION WITH BOKEH

Bryan Van de Ven
Core Developer of Bokeh

datacamp


mean: 0
stddev: 0.04
12-tick Moving Avg
160
140
120
100
80
60
5
0
-5
-10
1380
1400
1420
1440
1460


Google Chrome (beta)
Bokeh Applicati
localhost:8787/workers
Dask -- Documentation -- Scheduler Docs -- Tasks -- Workers
DASK
Processing and Pending
Memory Usage (%)
host cores processes memory cpu memory %
0 127.0.0.1 4 1 16 GiB 48.7 % 28.2 %
1 127.0.0.2 4 1 16 GiB 37.9 % 28.2 %
2 127.0.0.3 4 1 16 GiB 43.9 % 28.1 %
3 127.0.0.4 4 1 16 GiB 40.7 % 28.2 %
Bokeh Applicati
localhost:8787/status
Dask -- Documentation -- Scheduler Docs -- Tasks -- Workers
DASK
Memory
CPU
Network Send
Network Recv
MB/s
Time
Memory Use: 0.00 MB
Task Stream
Worker Core
Time
Progress -- total: 802, in-memory: 148, processing: 16, ready: 19, waiting: 481, failed: 0
compare 67 / 470
load 89 / 110
process 85 / 110
roll 43 / 106
finalize 1 / 2
load_from_sql 2 / 2
reduction 1 / 2

Basic app outline

`outline.py`

```
from bokeh.io import curdoc

# Create plots and widgets

# Add callbacks

# Arrange plots and widgets in layouts

curdoc().add_root(layout)
```

Running Bokeh applications

Running single module apps at the shell or Windows command prompt:

```
bokeh serve --show myapp.py
```

"Directory" style apps run similarly:

```
bokeh serve --show myappdir/
```

Let's practice!

INTERACTIVE DATA VISUALIZATION WITH BOKEH

Connecting sliders to plots

INTERACTIVE DATA VISUALIZATION WITH BOKEH

Bryan Van de Ven
Core Developer of Bokeh

A slider example

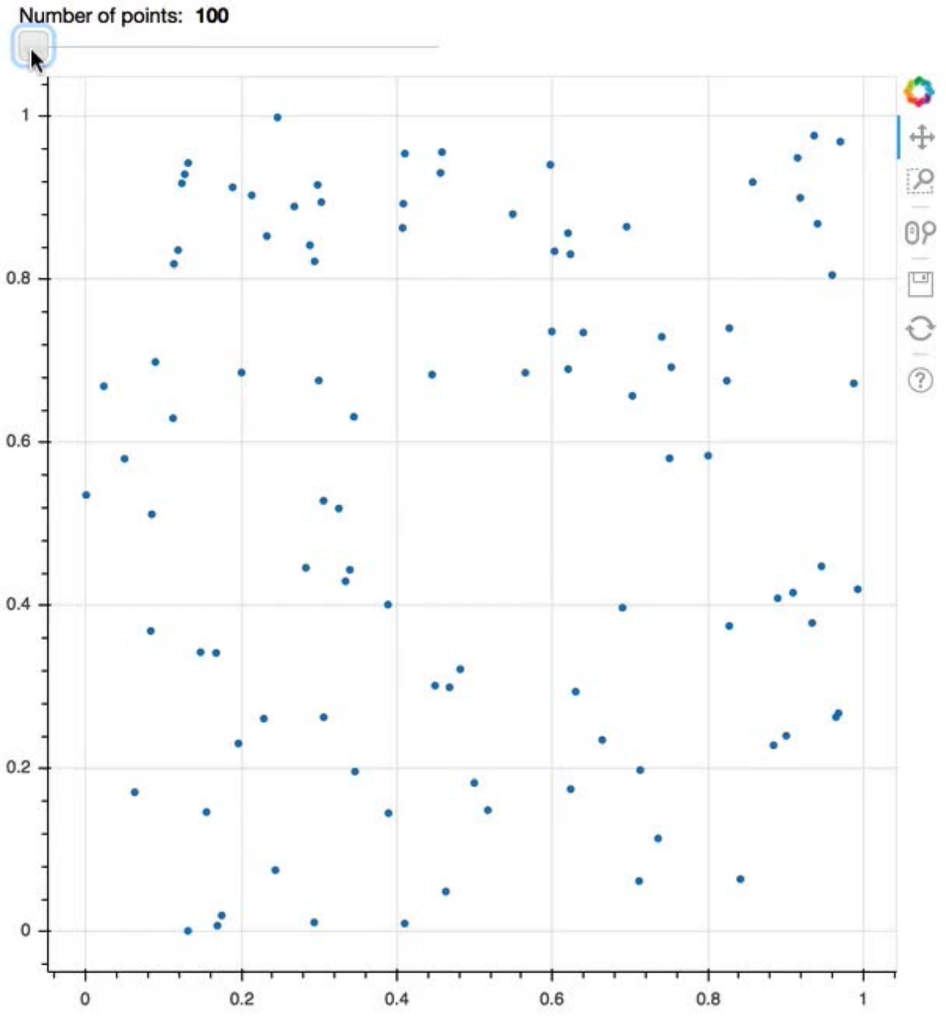
slider.py

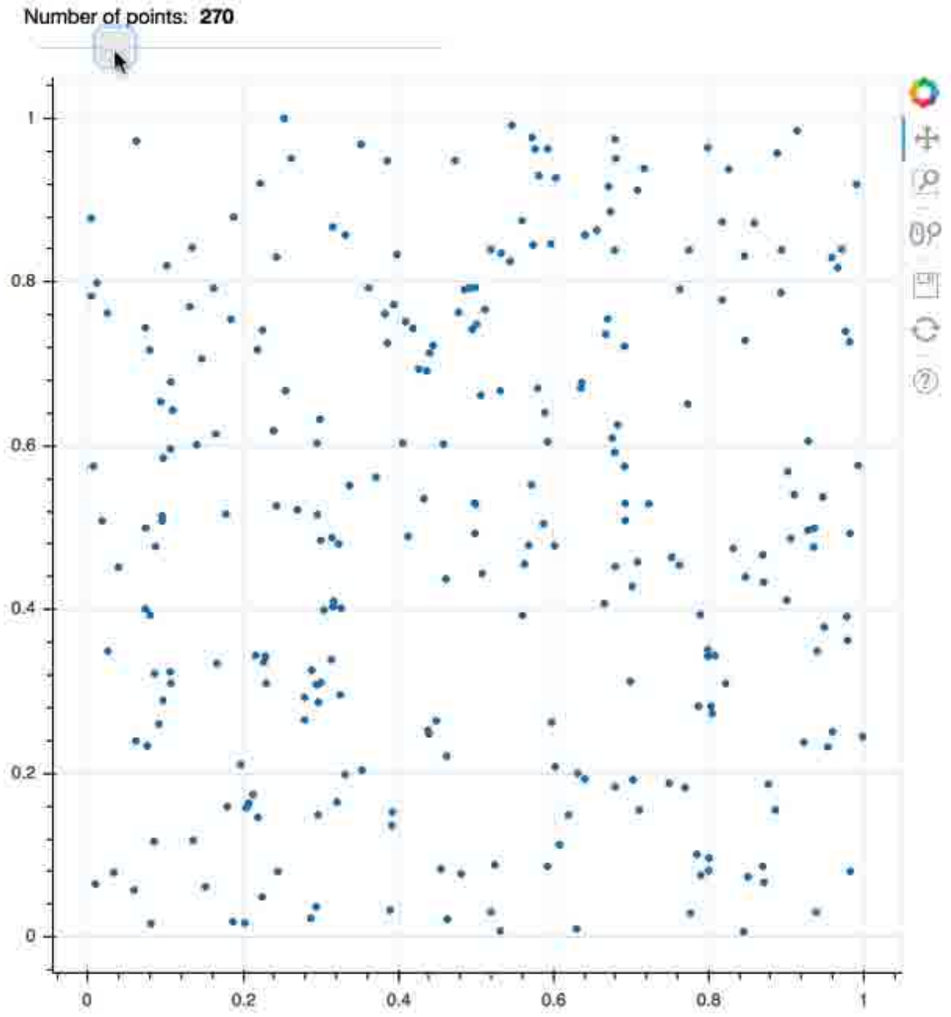
```
from bokeh.io import curdoc
from bokeh.layouts import column
from bokeh. models import ColumnDataSource, Slider
from bokeh.plotting import figure
from numpy.random import random

N = 300
source = ColumnDataSource(data={'x': random(N), 'y': random(N)})

# Create plots and widgets
plot = figure()
plot.circle(x='x', y='y', source=source)

slider = Slider(start=100, end=1000, value=N,
            step=10, title='Number of points')
```

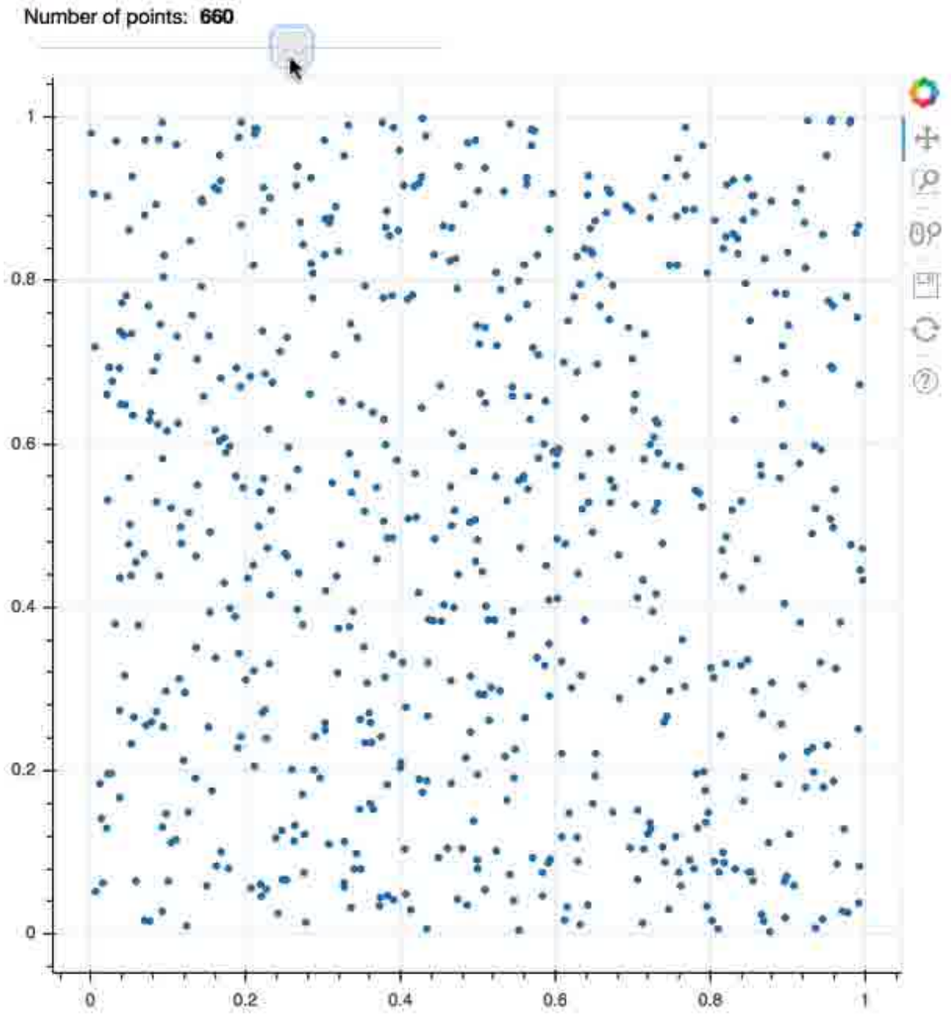
A slider example

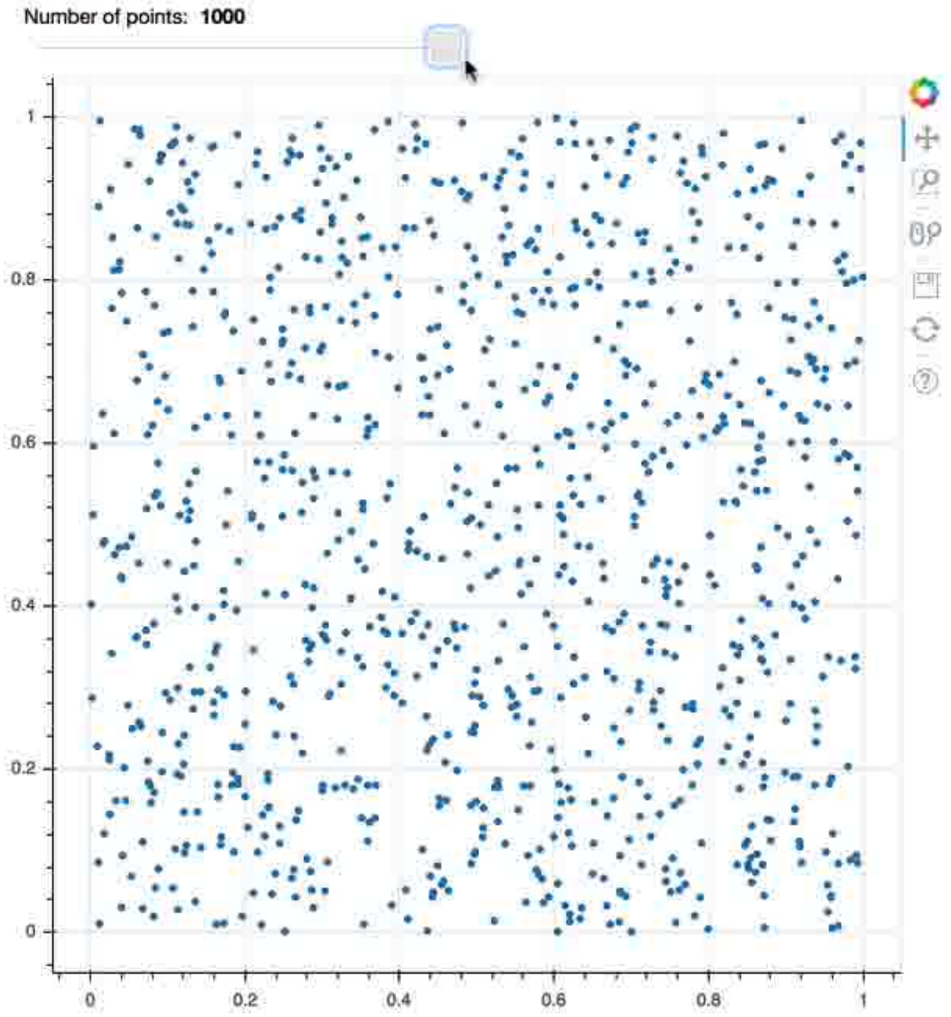
slider.py

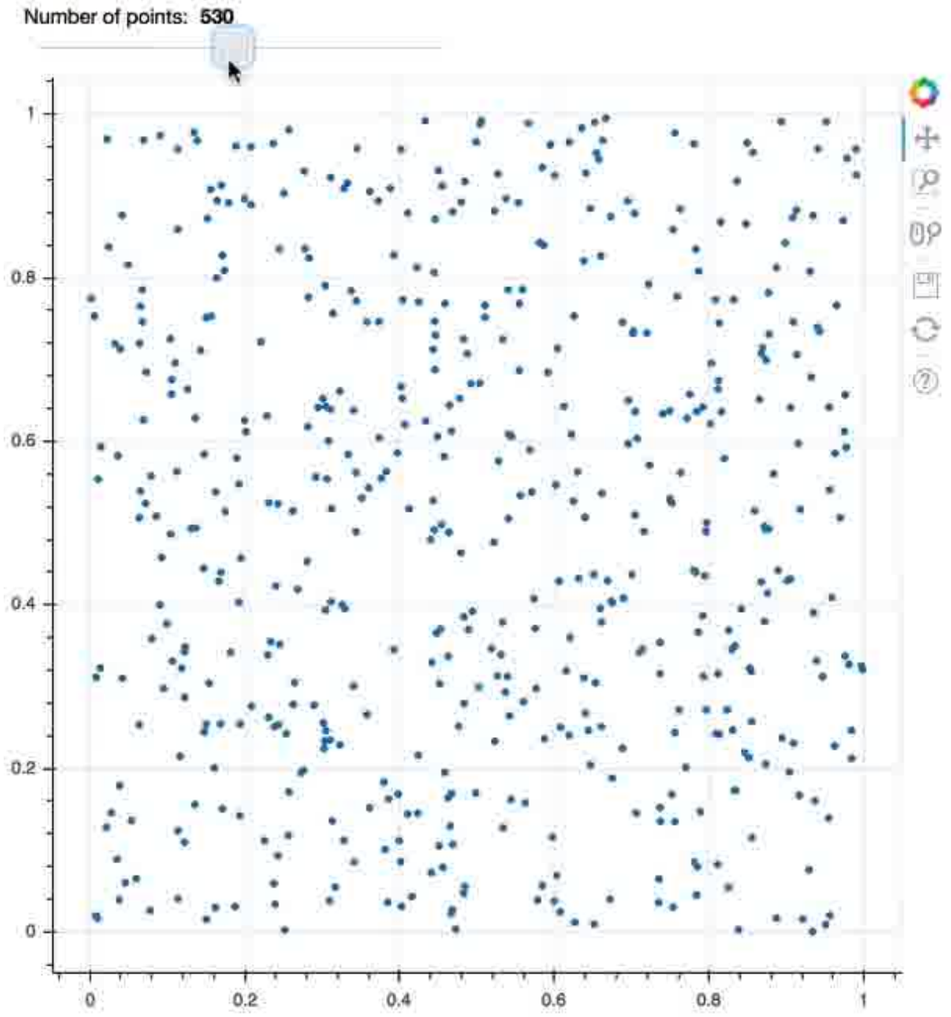
```
# (continued)

# Add callback to widgets
def callback(attr, old, new):
    N = slider.value
    source.data={'x': random(N), 'y': random(N)}

slider.on_change('value', callback)

# Arrange plots and widgets in layouts
layout = column(slider, plot)

curdoc().add_root(layout)
```


Number of points: 100
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


Number of points: 270
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


Number of points: 660
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


Number of points: 1000
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


Number of points: 530
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

Let's practice!

INTERACTIVE DATA VISUALIZATION WITH BOKEH

Updating plots from dropdowns

INTERACTIVE DATA VISUALIZATION WITH BOKEH

Bryan Van de Ven
Core Developer of Bokeh

A Select example

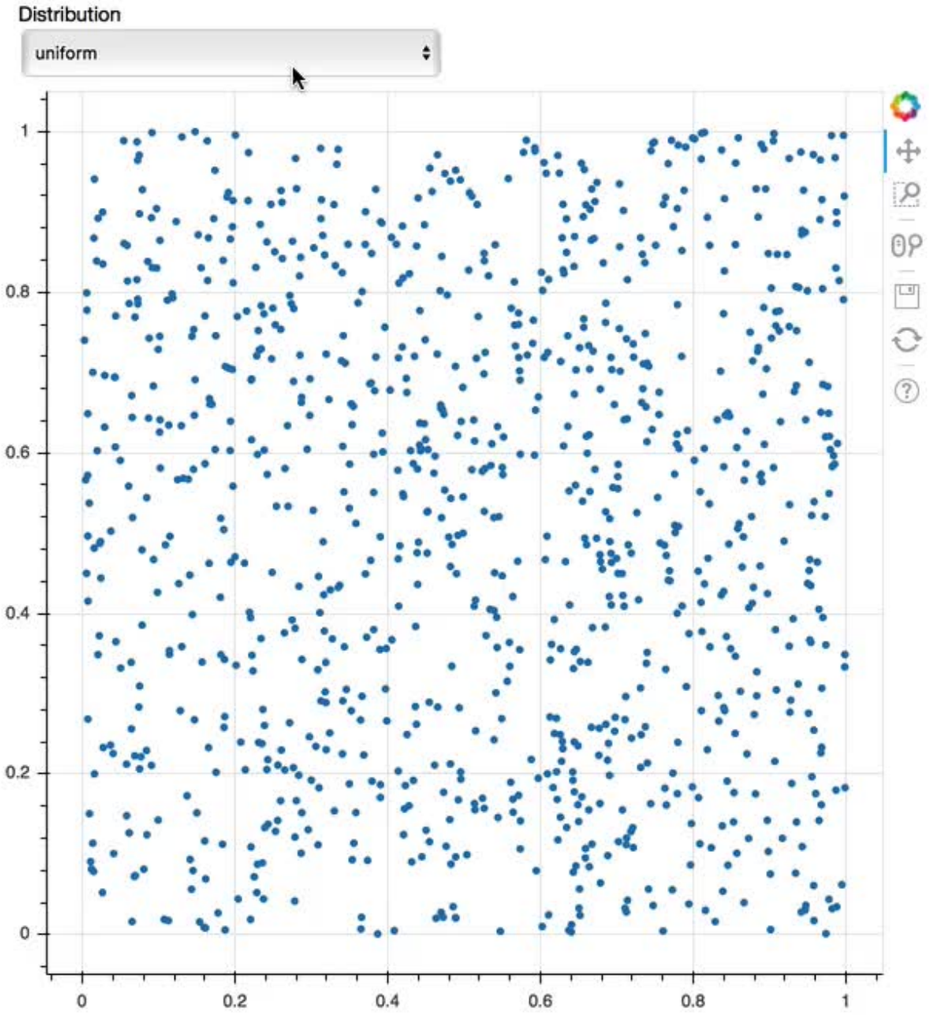
select.py

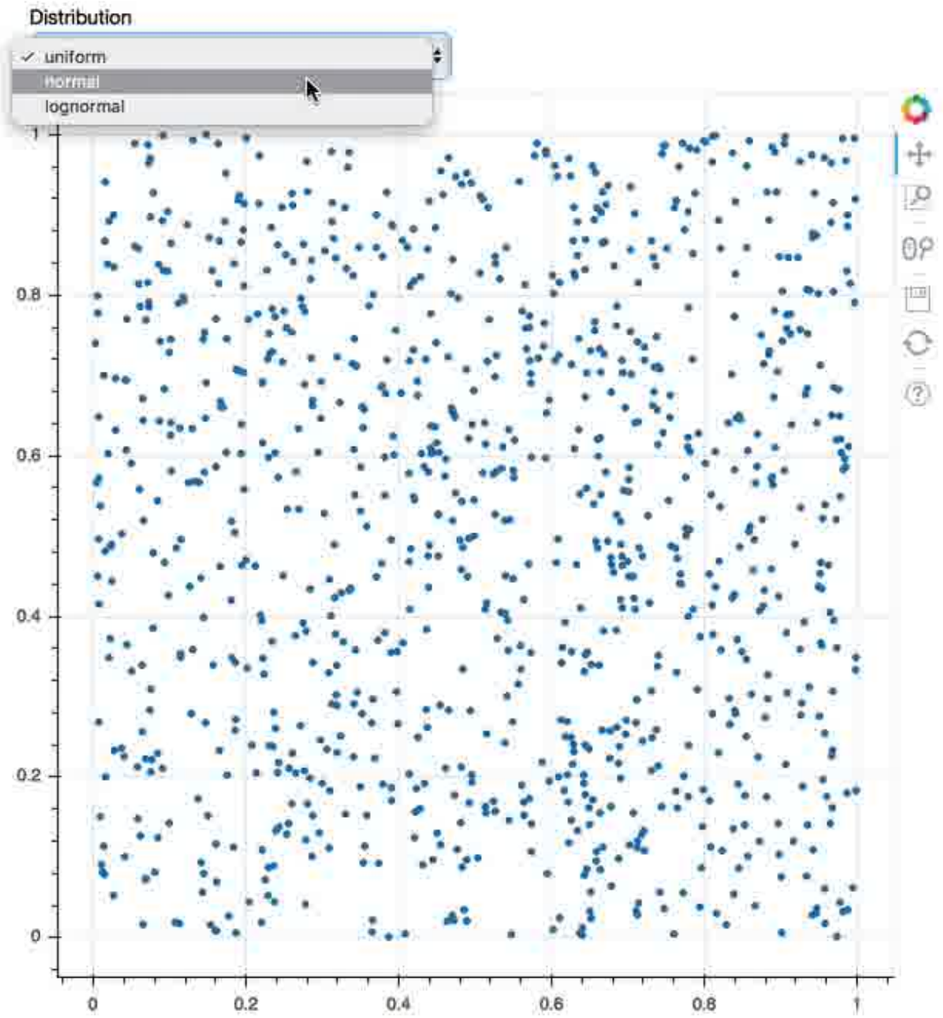
```
from bokeh.io import curdoc
from bokeh.layouts import column
from bokeh. models import ColumnDataSource, Select
from bokeh.plotting import figure
from numpy.random import random, normal, lognormal

N = 1000
source = ColumnDataSource(data={'x': random(N), 'y': random(N)})

# Create plots and widgets
plot = figure()
plot.circle(x='x', y='y', source=source)

menu = Select(options=['uniform', 'normal', 'lognormal'],
              value='uniform', title='Distribution')
```

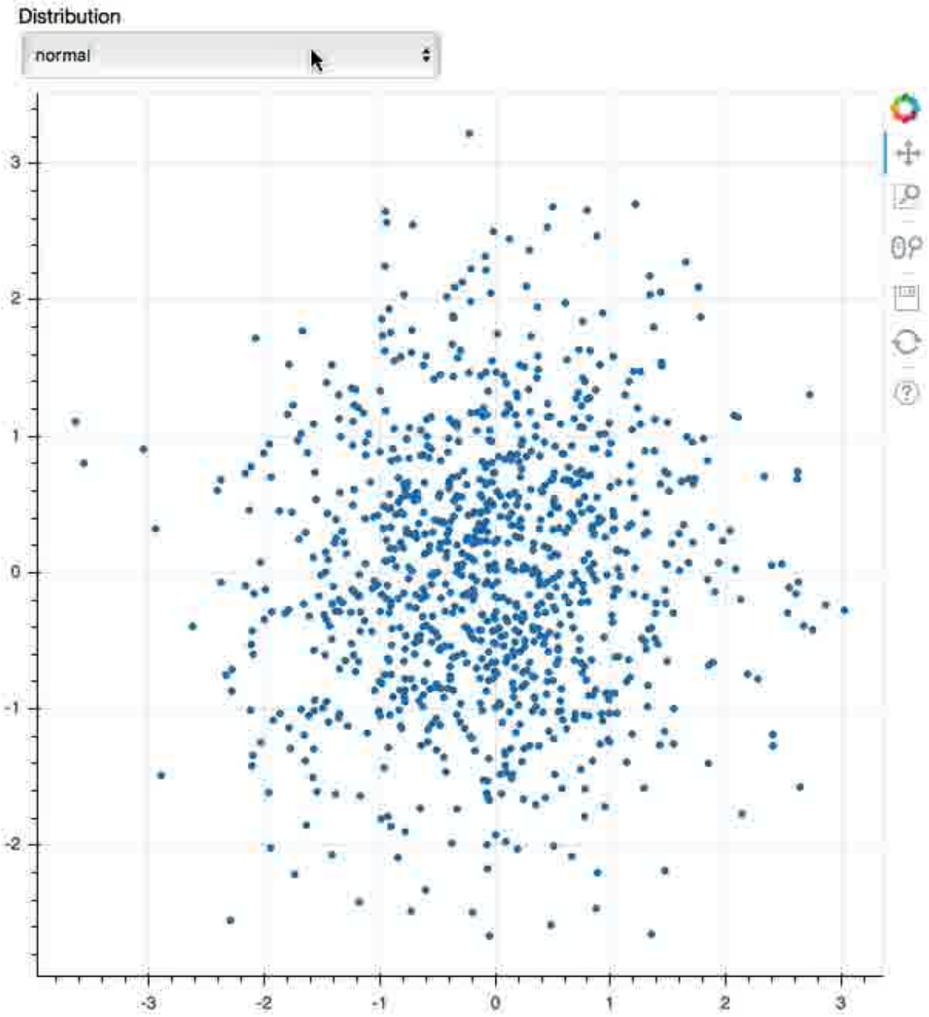
A Select example

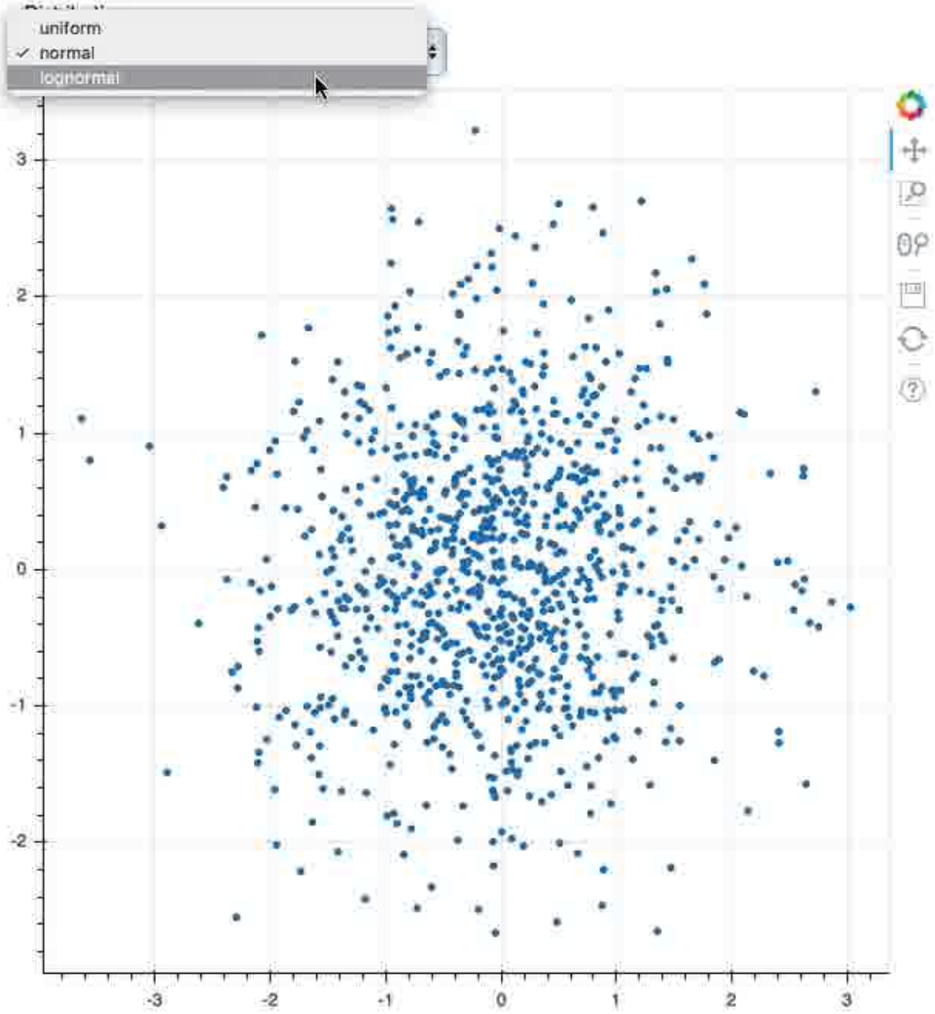
select.py

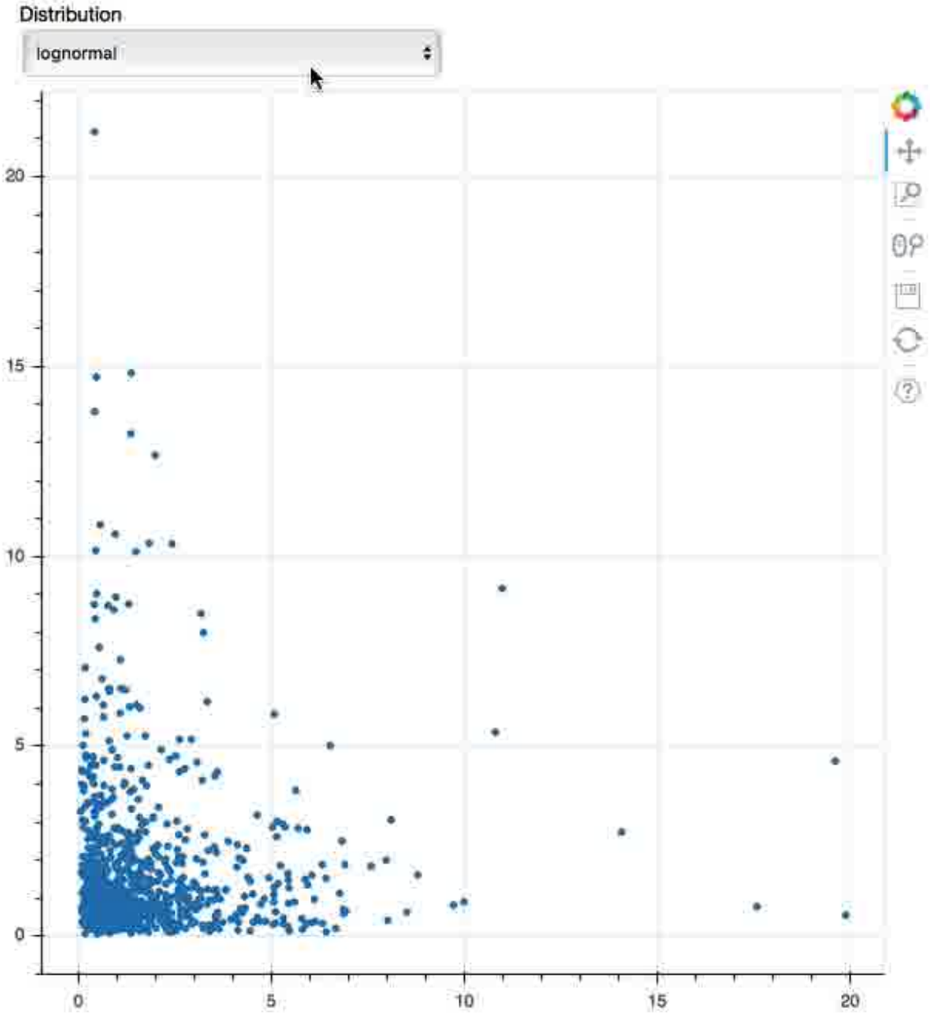
```
# (continued)

# Add callback to widgets
def callback(attr, old, new):
    if menu.value == 'uniform': f = random
    elif menu.value == 'normal': f = normal
    else: f = lognormal
    source.data={'x': f(size=N), 'y': f(size=N)}

menu.on_change('value', callback)

# Arrange plots and widgets in layouts
layout = column(menu, plot)

curdoc().add_root(layout)
```


Distribution
uniform
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


Distribution
uniform
normal
lognormal
0
0.2
0.4
0.6
0.8
1


Distribution
normal
3
2
1
0
-1
-2
-3
-2
-1
0
1
2
3


uniform
normal
lognormal
3
2
1
0
-1
-2
-3
-2
-1
0
1
2
3


Distribution
lognormal
20
15
10
5
0
0
5
10
15
20

Let's practice!

INTERACTIVE DATA VISUALIZATION WITH BOKEH

Buttons

INTERACTIVE DATA VISUALIZATION WITH BOKEH

Bryan Van de Ven
Core Developer of Bokeh

Button callbacks

select.py

```
from bokeh.models import Button

button = Button(label='press me')

def update():
    # Do something interesting

button.on_click(update)
```

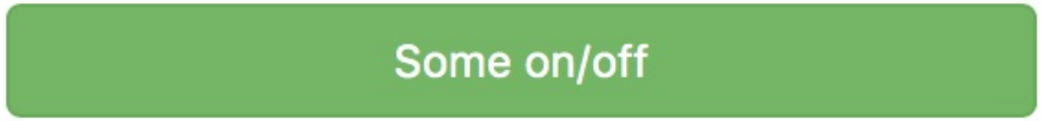
Button types

```
from bokeh.models import CheckboxGroup, RadioGroup, Toggle

toggle = Toggle(label='Some on/off', button_type='success')

checkbox = CheckboxGroup(labels=['foo', 'bar', 'baz'])

radio = RadioGroup(labels=['2000', '2010', '2020'])

def callback(active):
    # Active tells which button is active
```

Button types

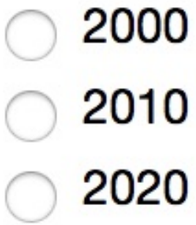
Plain button

Toggle



Radio Group

Checkbox Group



Let's practice!

INTERACTIVE DATA VISUALIZATION WITH BOKEH

Hosting applications for wider audiences

INTERACTIVE DATA VISUALIZATION WITH BOKEH

Bryan Van de Ven
Core Developer of Bokeh

datacamp

Bokeh Application Hosting

https://anaconda.org

Let's practice!

INTERACTIVE DATA VISUALIZATION WITH BOKEH