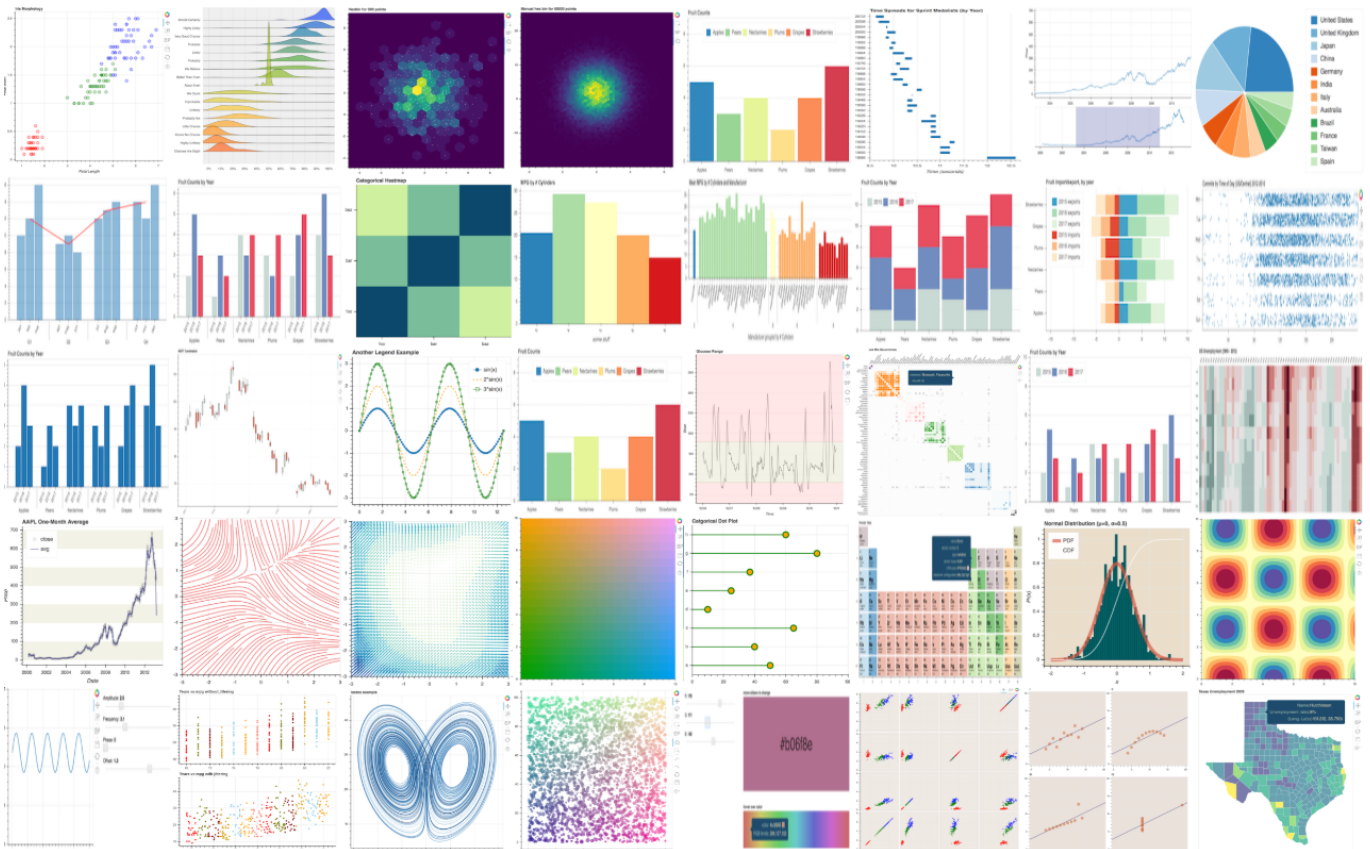


▼ Bokeh 한번에 제대로 배우기

bokeh



▼ Bokeh 특징

- 최신 브라우저의 인터랙티브 시각화
- 독립형 HTML 문서 또는 서버 지원
- 표현력이 뛰어나고 다양한 그래픽 지원
- 큰 동적 데이터 또는 스트리밍 데이터 지원
- 파이썬(또는 Scala, R, ...)에서 쉽게 사용
- 자바스크립트 불필요

```
import numpy as np
import pandas as pd
from bokeh.io import output_notebook, show
from bokeh.plotting import figure, show
```

```
from bokeh.plotting import figure, show
```

```
output_notebook()
```

- 샘플 데이터 다운로드

```
import bokeh.sampledata
```

```
bokeh.sampledata.download()
```

```
Creating /root/.bokeh directory
```

```
Creating /root/.bokeh/data directory
```

```
Using data directory: /root/.bokeh/data
```

```
Downloading: CGM.csv (1589982 bytes)
```

```
1589982 [100.00%]
```

```
Downloading: US_Counties.zip (3171836 bytes)
```

```
3171836 [100.00%]
```

```
Unpacking: US_Counties.csv
```

```
Downloading: us_cities.json (713565 bytes)
```

```
713565 [100.00%]
```

```
Downloading: unemployment09.csv (253301 bytes)
```

```
253301 [100.00%]
```

```
Downloading: AAPL.csv (166698 bytes)
```

```
166698 [100.00%]
```

```
Downloading: FB.csv (9706 bytes)
```

```
9706 [100.00%]
```

```
Downloading: GOOG.csv (113894 bytes)
```

```
113894 [100.00%]
```

```
Downloading: IBM.csv (165625 bytes)
```

```
165625 [100.00%]
```

```
Downloading: MSFT.csv (161614 bytes)
```

```
161614 [100.00%]
```

```
Downloading: WPP2012_SA_DB03_POPULATION_QUINQUENNIAL.zip (4816256 bytes)
```

```
4816256 [100.00%]
```

```
Unpacking: WPP2012_SA_DB03_POPULATION_QUINQUENNIAL.csv
```

```
Downloading: gapminder_fertility.csv (64346 bytes)
```

```
64346 [100.00%]
```

```
Downloading: gapminder_population.csv (94509 bytes)
```

```
94509 [100.00%]
```

```
Downloading: gapminder_life_expectancy.csv (73243 bytes)
```

```
73243 [100.00%]
```

```
Downloading: gapminder_regions.csv (7781 bytes)
```

```
7781 [100.00%]
```

```
Downloading: world_cities.zip (645274 bytes)
```

```
645274 [100.00%]
```

```
Unpacking: world_cities.csv
```

```
Downloading: airports.json (6373 bytes)
```

```
6373 [100.00%]
```

```
Downloading: movies.db.zip (5053420 bytes)
```

```
5053420 [100.00%]
```

```
Unpacking: movies.db
```

```
Downloading: airports.csv (203190 bytes)
```

```
203190 [100.00%]
```

```
Downloading: routes.csv (377280 bytes)
```

```
377280 [100.00%]
```

```
Downloading: haarcascade_frontalface_default.xml (930127 bytes)
```

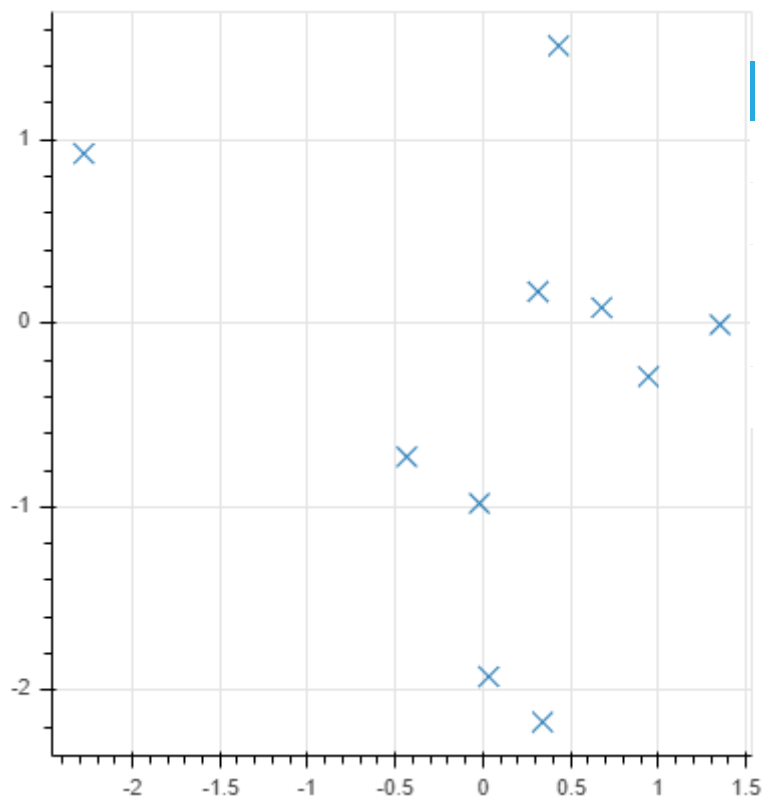
```
930127 [100.00%]
```

▼ 산점도(Scatter Plots)

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

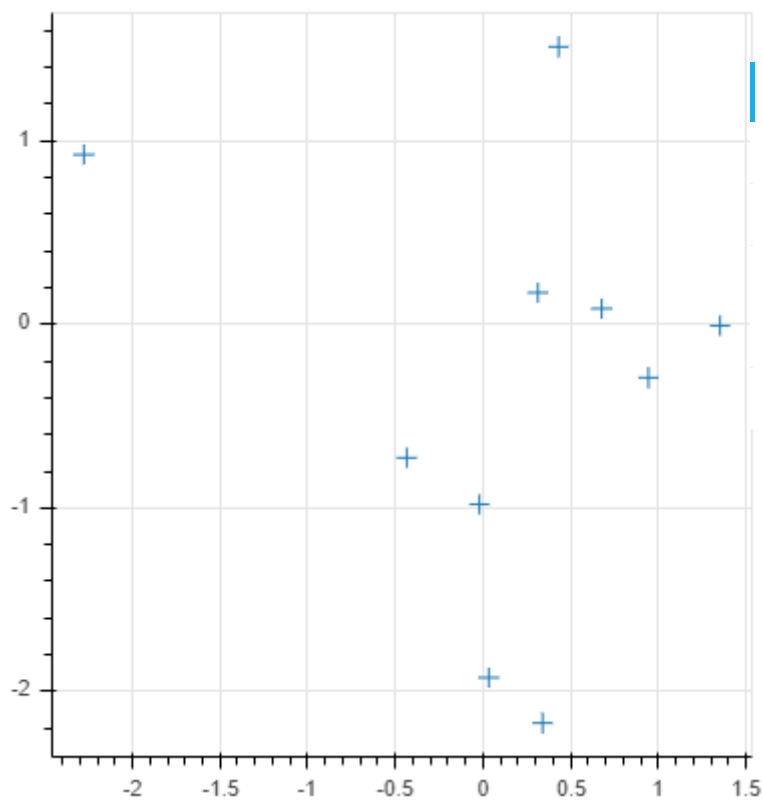
```
x = np.random.randn(10)
y = np.random.randn(10)
sz = np.random.randint(10, 40, size=10)
```

```
p = figure(plot_width=400, plot_height=400)
p.x(x, y, size=10)
show(p)
```

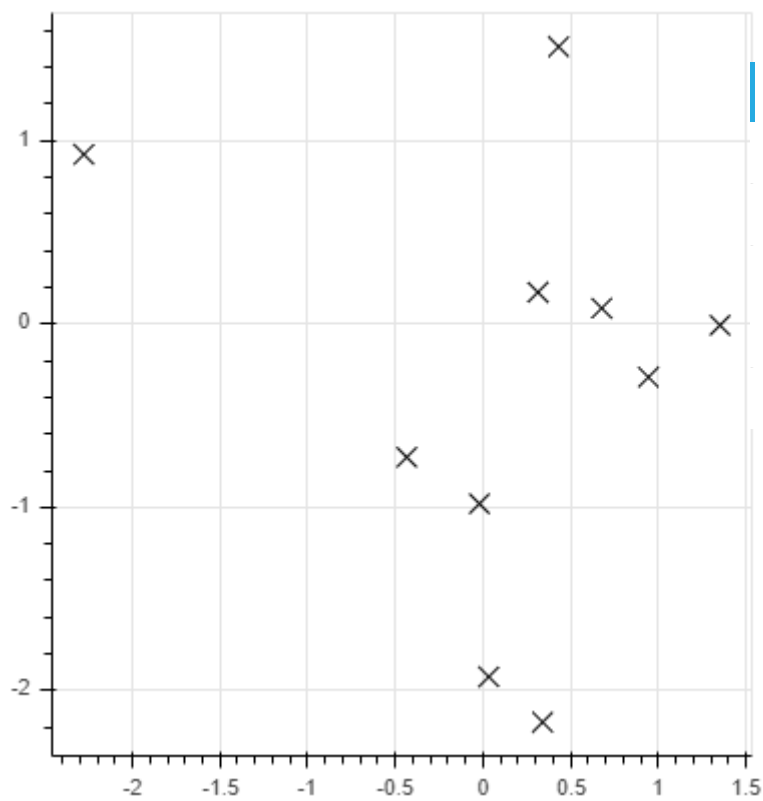


```
p = figure(plot_width=400, plot_height=400)
p.cross(x, y, size=10)
```

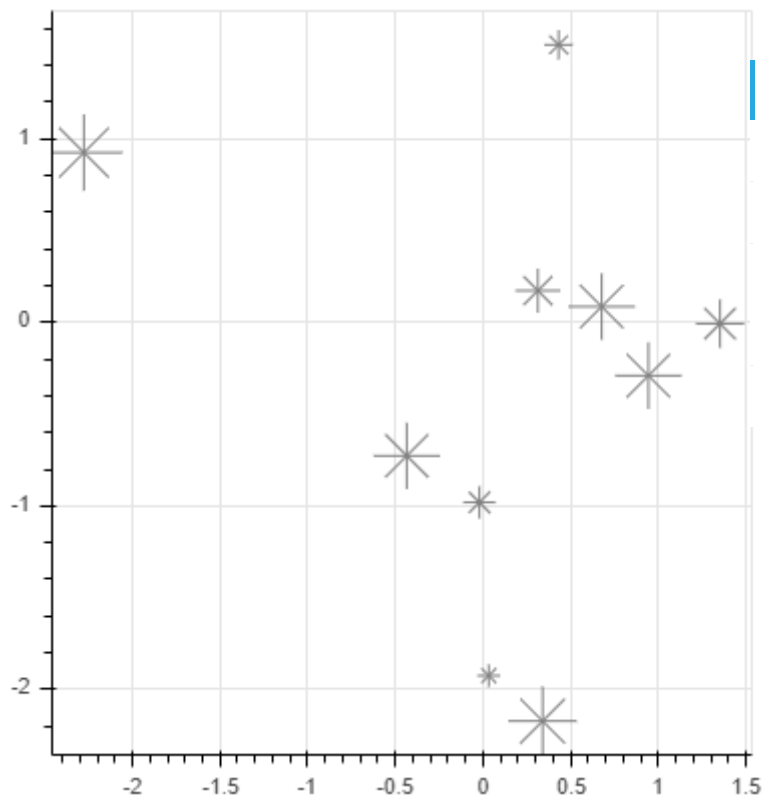
```
show(p)
```



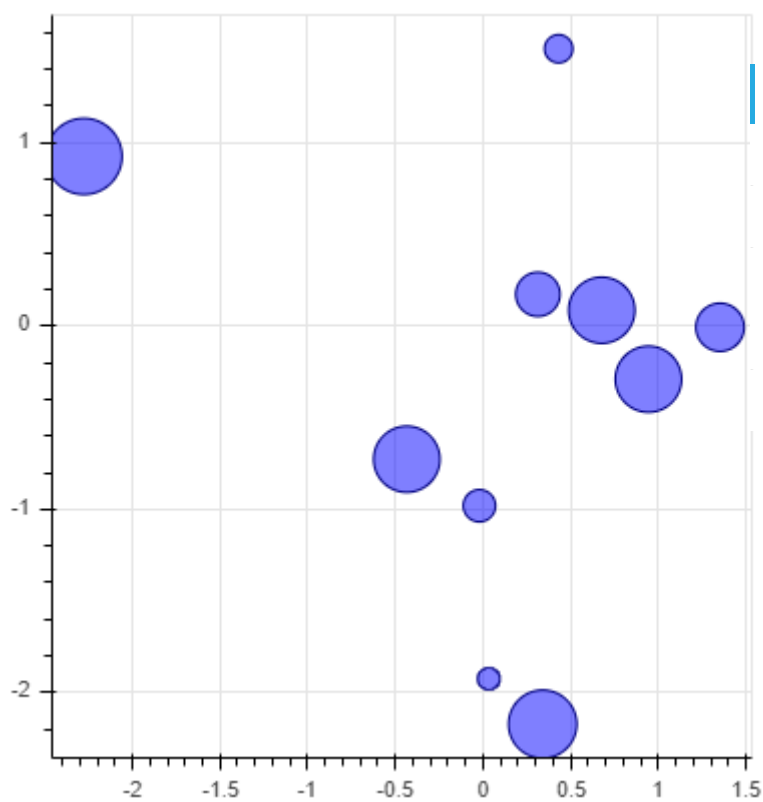
```
p = figure(plot_width=400, plot_height=400)
p.x(x, y, size=10, color="black")
show(p)
```



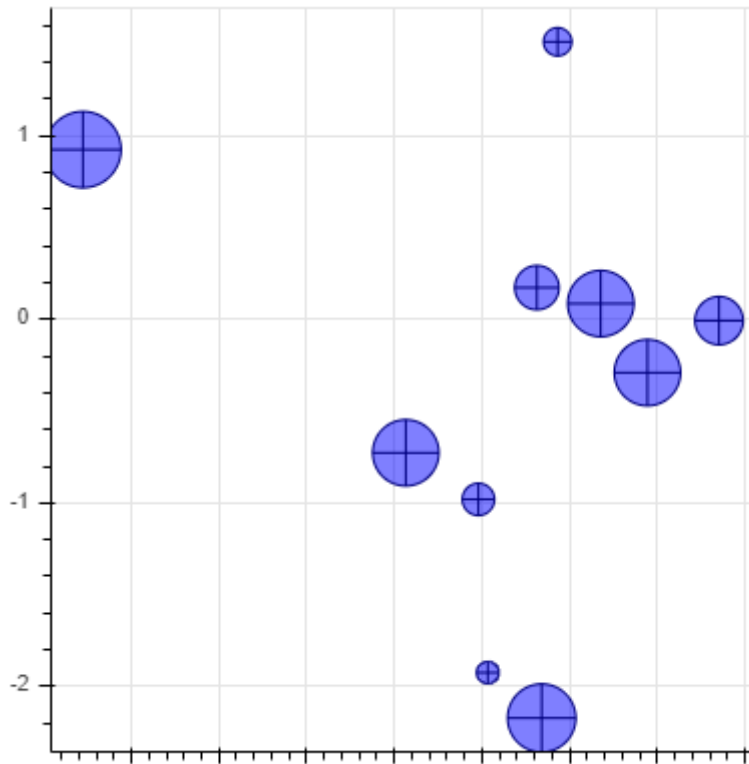
```
p = figure(plot_width=400, plot_height=400)
p.asterisk(x, y, size=sz, line_color="gray")
show(p)
```



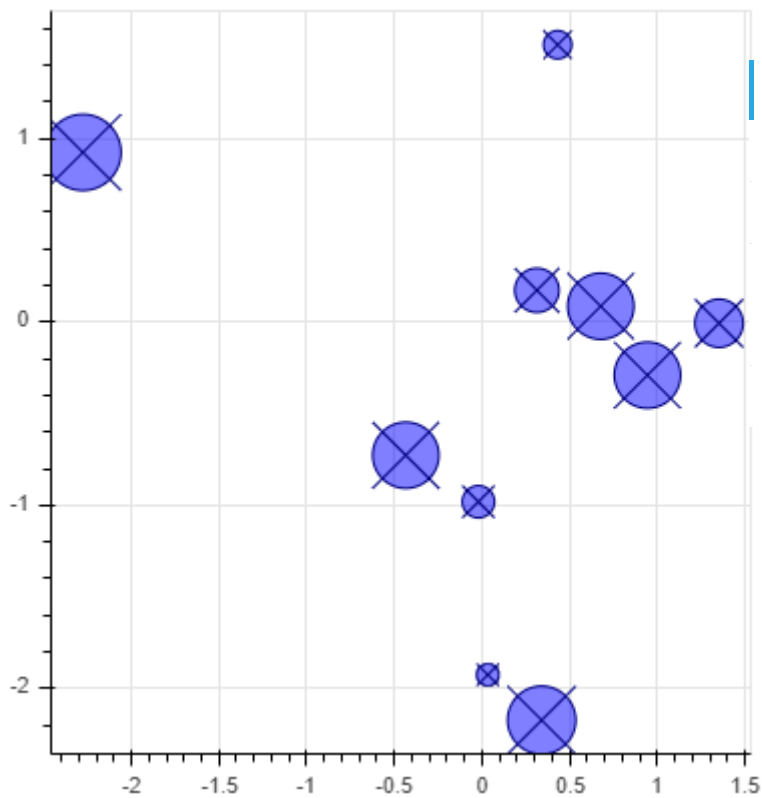
```
p = figure(plot_width=400, plot_height=400)
p.circle(x, y, size=sz, line_color="navy", fill_color="blue", fill_alpha=0.5)
show(p)
```



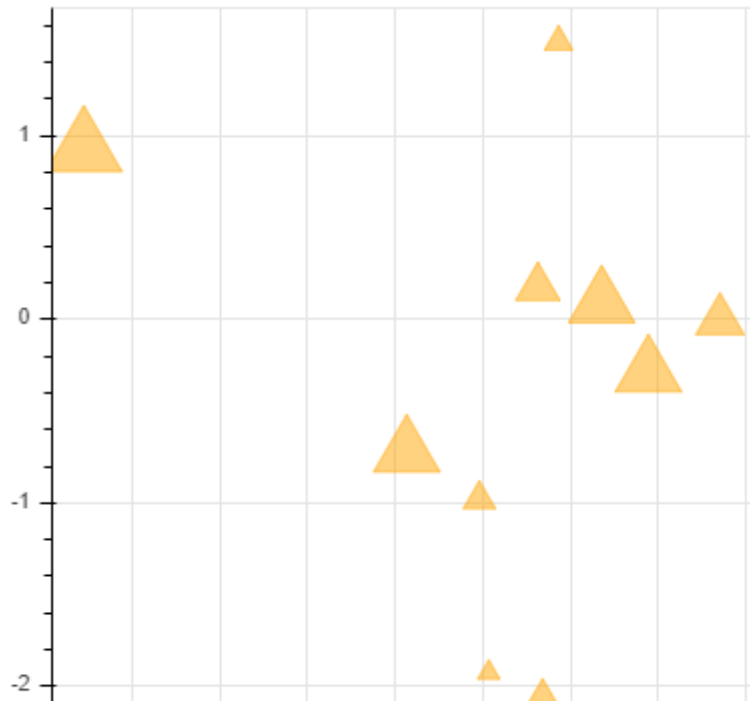
```
p = figure(plot_width=400, plot_height=400)
p.circle_cross(x, y, size=sz, line_color="navy", fill_color="blue", fill_alpha=0.5)
show(p)
```



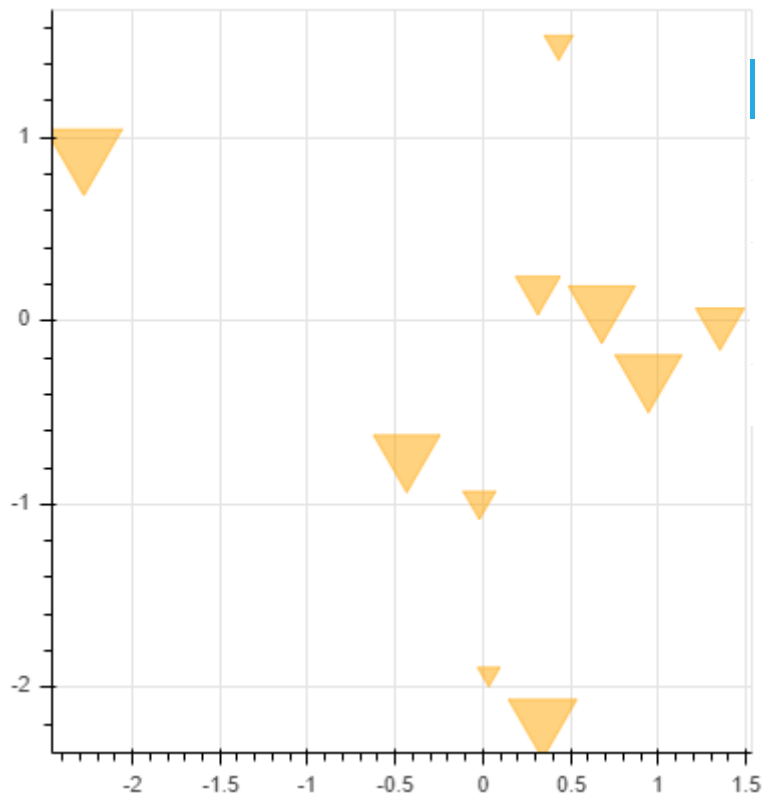
```
p = figure(plot_width=400, plot_height=400)
p.circle_x(x, y, size=sz, line_color="navy", fill_color="blue", fill_alpha=0.5)
show(p)
```



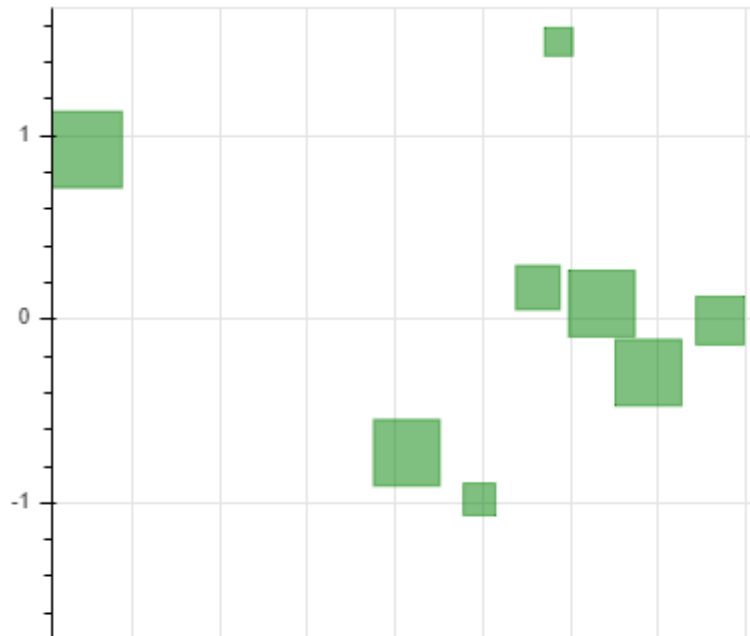
```
p = figure(plot_width=400, plot_height=400)
p.triangle(x, y, size=sz, color="orange", alpha=0.5)
show(p)
```



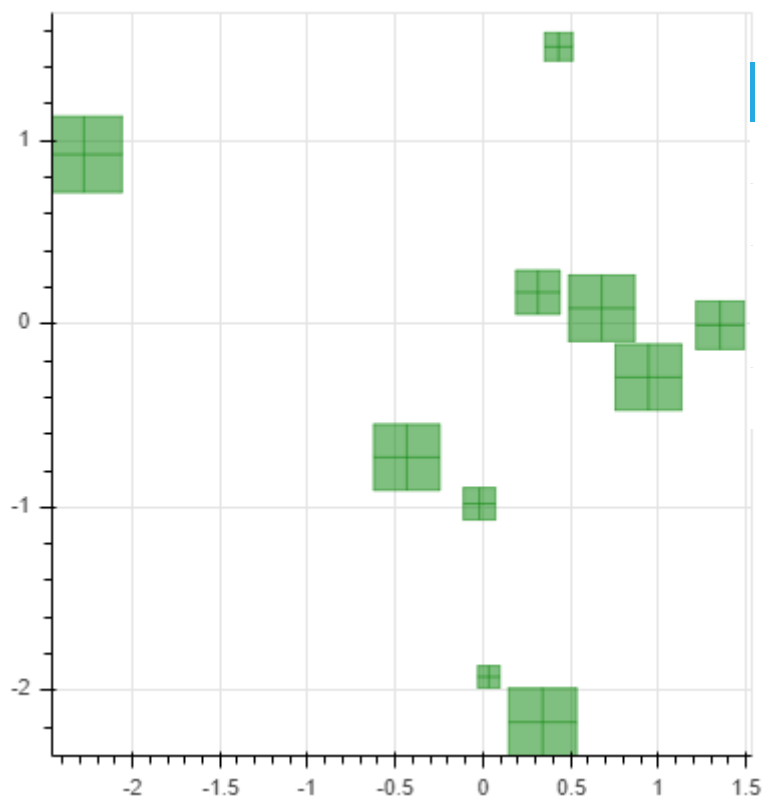
```
p = figure(plot_width=400, plot_height=400)
p.inverted_triangle(x, y, size=sz, color="orange", alpha=0.5)
show(p)
```



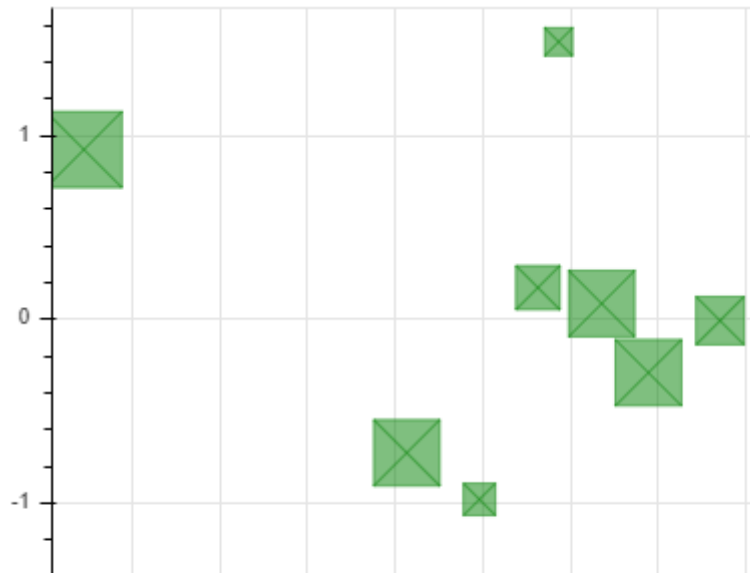
```
p = figure(plot_width=400, plot_height=400)
p.square(x, y, size=sz, color="green", alpha=0.5)
show(p)
```



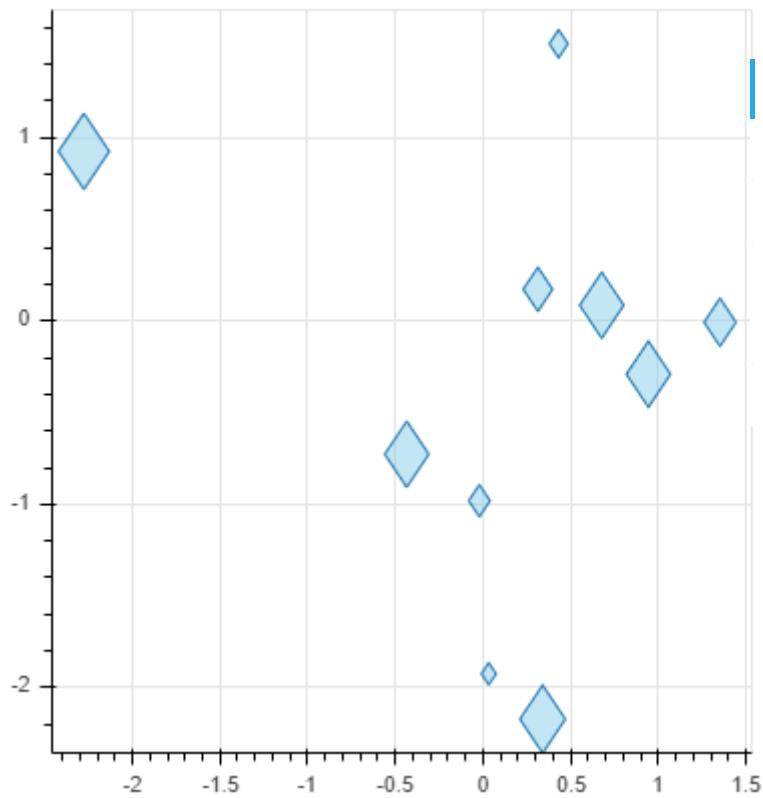
```
p = figure(plot_width=400, plot_height=400)
p.square_cross(x, y, size=sz, color="green", alpha=0.5)
show(p)
```



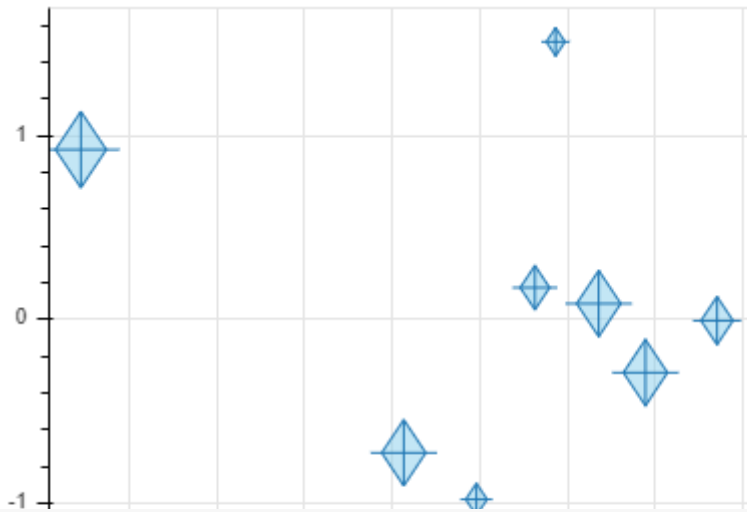
```
p = figure(plot_width=400, plot_height=400)
p.square_x(x, y, size=sz, color="green", alpha=0.5)
show(p)
```

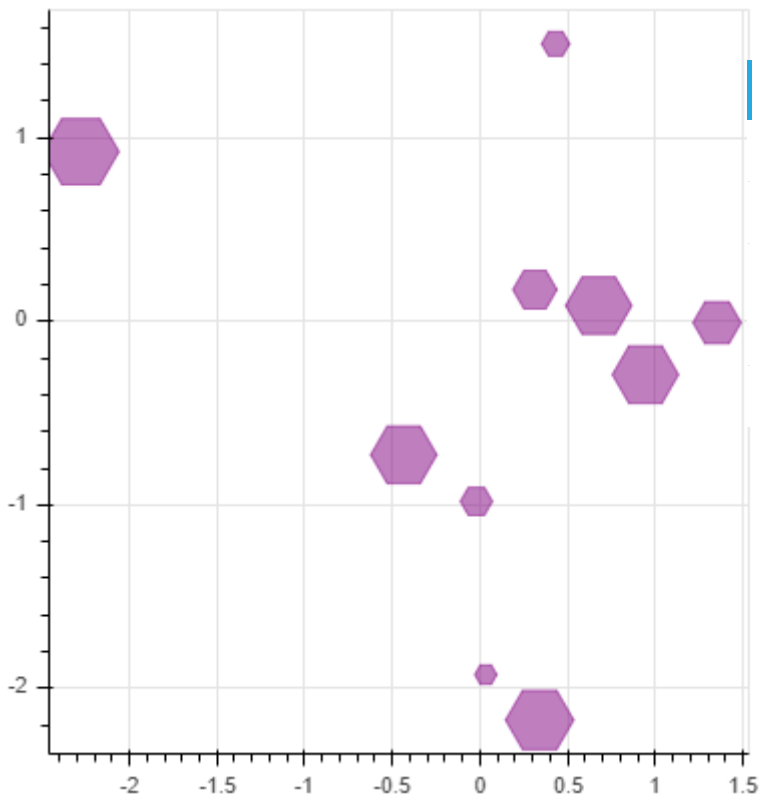
```
p = figure(plot_width=400, plot_height=400)
p.diamond(x, y, size=sz, fill_color="skyblue", fill_alpha=0.5)
show(p)
```



```
p = figure(plot_width=400, plot_height=400)
p.diamond_cross(x, y, size=sz, fill_color="skyblue", fill_alpha=0.5)
show(p)
```



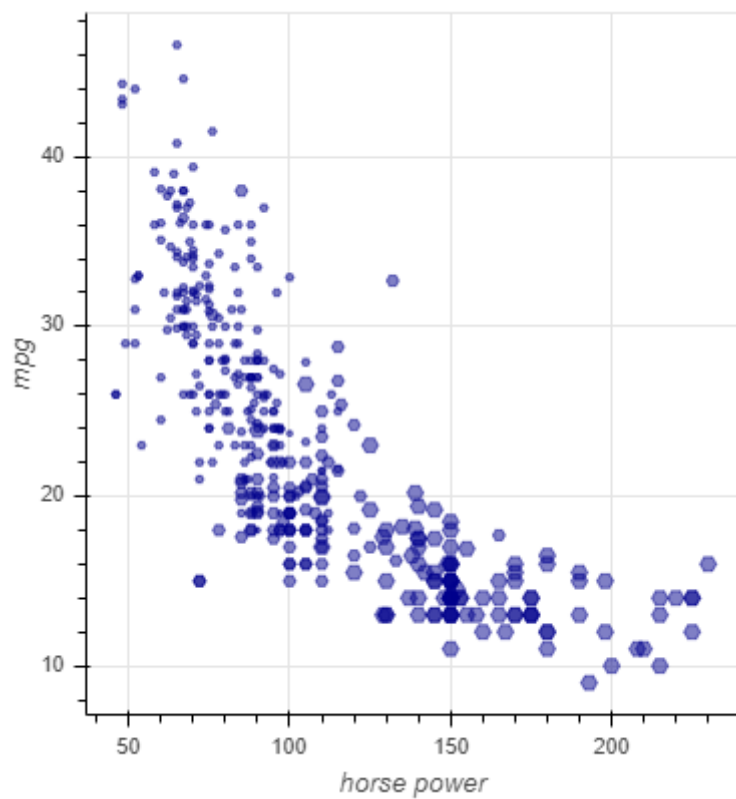
```
p = figure(plot_width=400, plot_height=400)
p.hex(x, y, size=sz, color="purple", alpha=0.5)
show(p)
```



```
from bokeh.sampledata.autompg import autompg
autompg.head()
```

	mpg	cyl	displ	hp	weight	accel	yr	origin	name
0	18.0	8	307.0	130	3504	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165	3693	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150	3436	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150	3433	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140	3449	10.5	70	1	ford torino

```
p = figure(plot_width=400, plot_height=400)
p.hex(autompghp, autompghmpg, size=autompghcyl, color="darkblue", alpha=0.5)
p.xaxis.axis_label = 'horse power'
p.yaxis.axis_label = 'mpg'
show(p)
```



```
p = figure(plot_width=400, plot_height=400)
p.hex(autompghyr, autompghhp, size=autompghcyl, color="darkblue", alpha=0.5)
p.xaxis.axis_label = 'year'
p.yaxis.axis_label = 'horse power'
show(p)
```



▼ 라인 플롯(Line Plots)



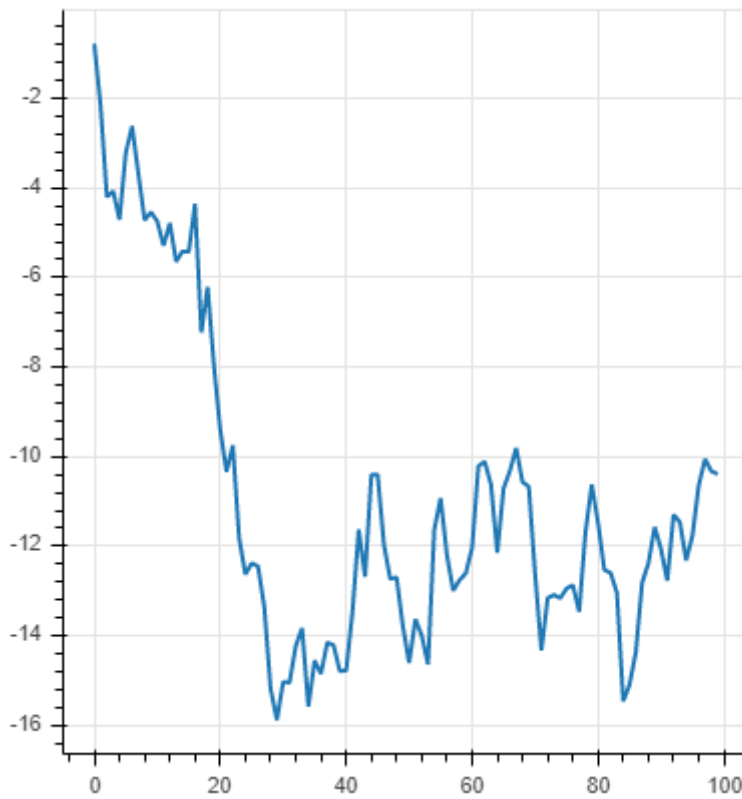
▼ 단일 라인(Single Lines)



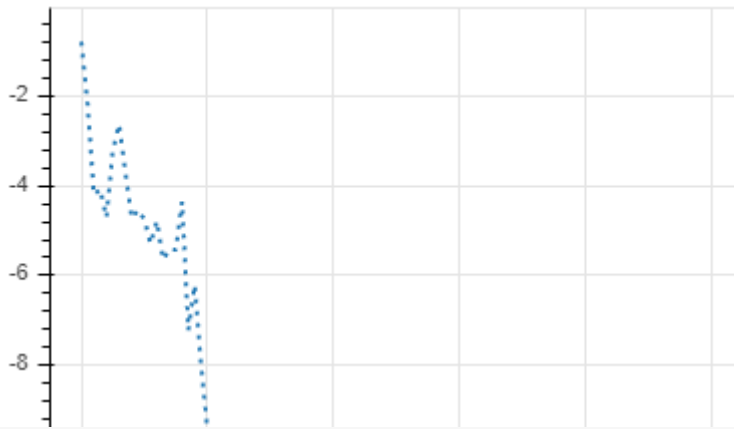
```
x = np.arange(100)
y = np.random.randn(100).cumsum()
```



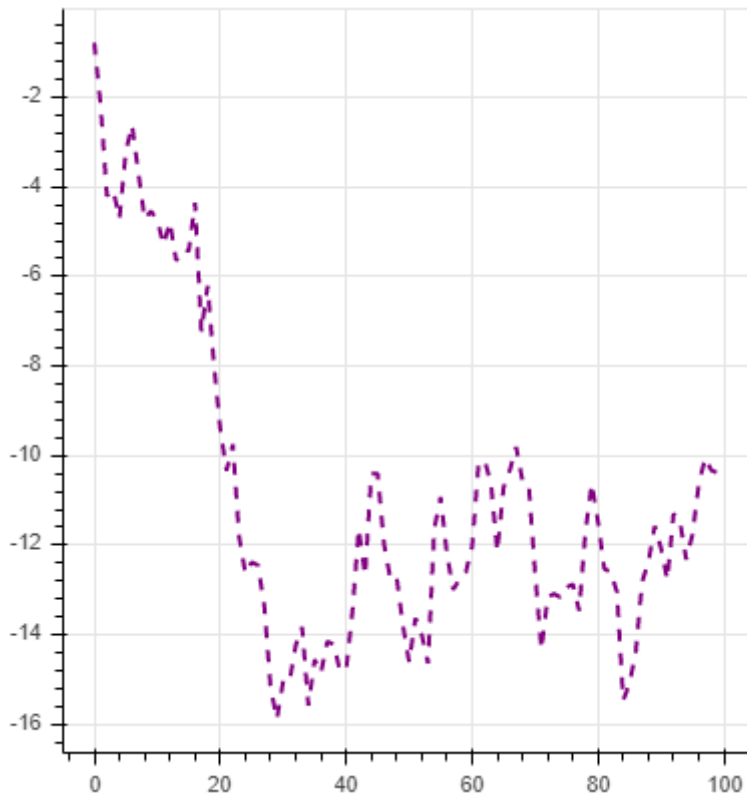
```
p = figure(plot_width=400, plot_height=400)
p.line(x, y, line_width=2)
show(p)
```



```
p = figure(plot_width=400, plot_height=400)
p.line(x, y, line_width=2, line_dash="dotted")
show(p)
```



```
p = figure(plot_width=400, plot_height=400)
p.line(x, y, line_width=2, line_dash="dashed", color="purple")
show(p)
```



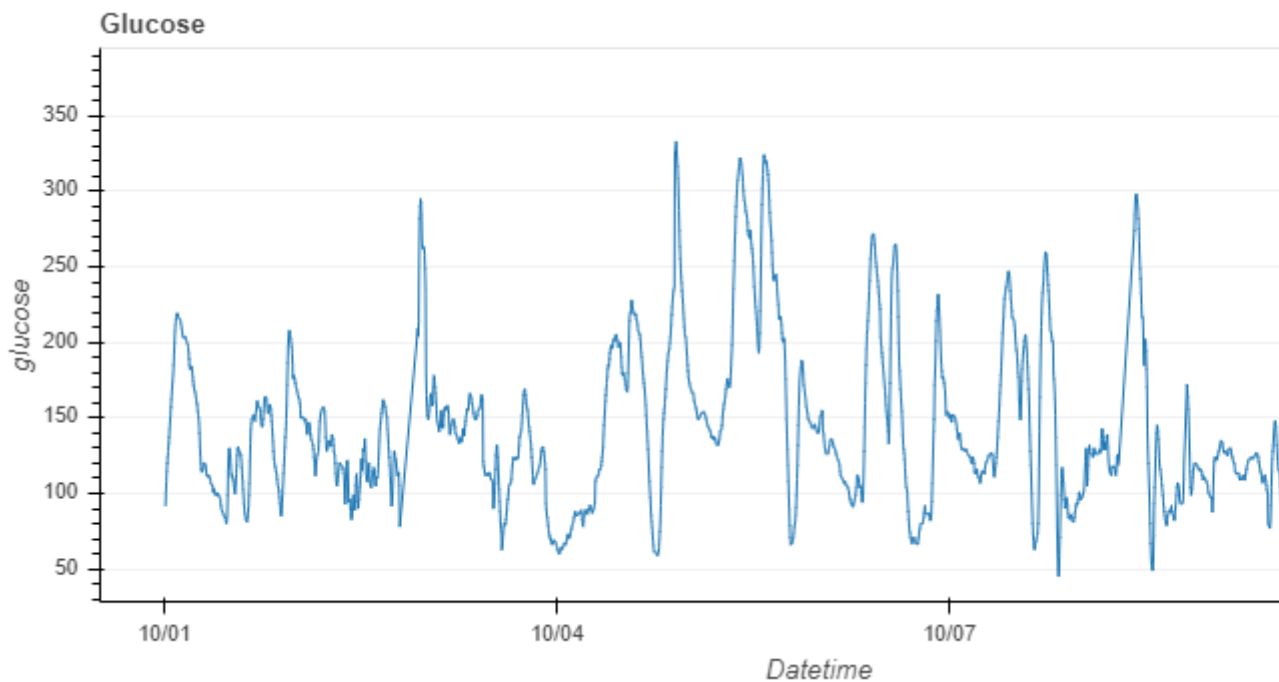
```
from bokeh.sampledata.glucose import data
data.tail()
```

	isig	glucose
datetime		
2010-10-10 23:37:00	29.46	160
2010-10-10 23:42:00	29.08	160
2010-10-10 23:47:00	29.06	160
2010-10-10 23:52:00	29.3	161
2010-10-10 23:57:00	30.8	165

```
days = data.loc['2010-10-01':'2010-10-10']
```

```
p = figure(x_axis_type="datetime", title="Glucose", plot_height=350, plot_width=800)
p.xgrid.grid_line_color=None
p.ygrid.grid_line_alpha=0.5
p.xaxis.axis_label = 'Datetime'
p.yaxis.axis_label = 'glucose'
p.line(days.index, days.glucose)

show(p)
```



```
from bokeh.sampledata.stocks import AAPL, GOOG
AAPL.keys()
```

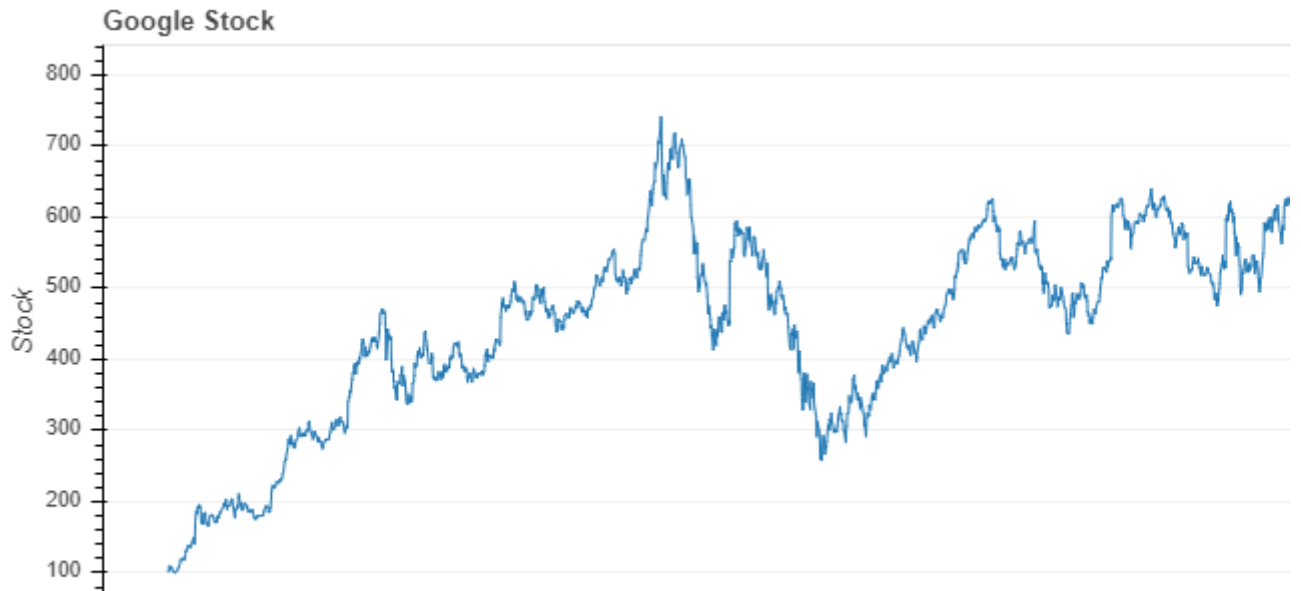
```
dict_keys(['date', 'open', 'high', 'low', 'close', 'volume', 'adj_close'])
```

```
dates = np.array(GOOG['date'], dtype=np.datetime64)
dates
```

```
array(['2004-08-19', '2004-08-20', '2004-08-23', ..., '2013-02-27',
       '2013-02-28', '2013-03-01'], dtype='datetime64[D]')
```

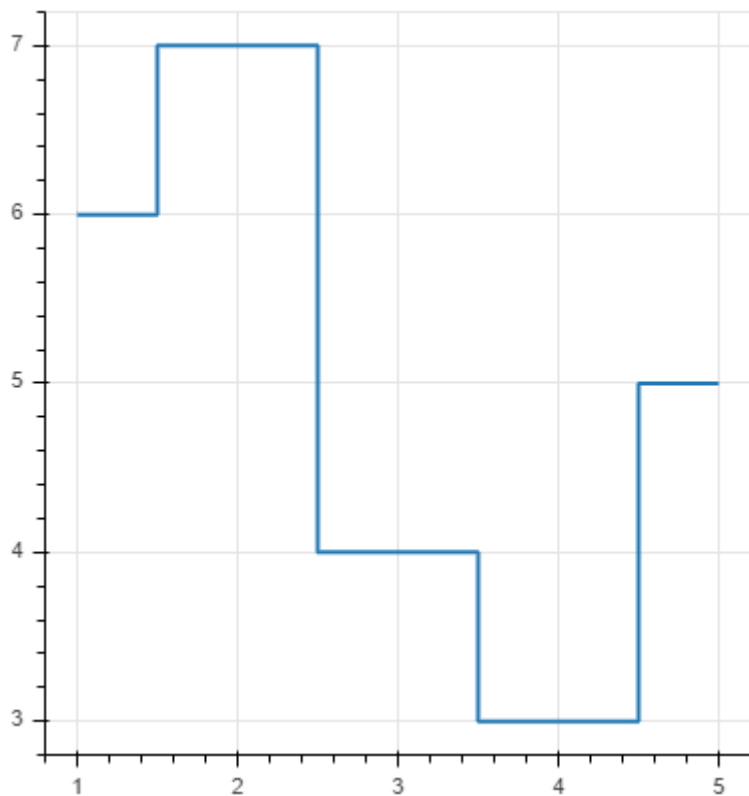
```
p = figure(x_axis_type="datetime", title="Google Stock", plot_height=350, plot_width=800)
p.xgrid.grid_line_color=None
p.ygrid.grid_line_alpha=0.5
p.xaxis.axis_label = 'Date'
p.yaxis.axis_label = 'Stock'
p.line(dates, GOOG['close'])

show(p)
```



▼ 스텝 라인(Step Lines)

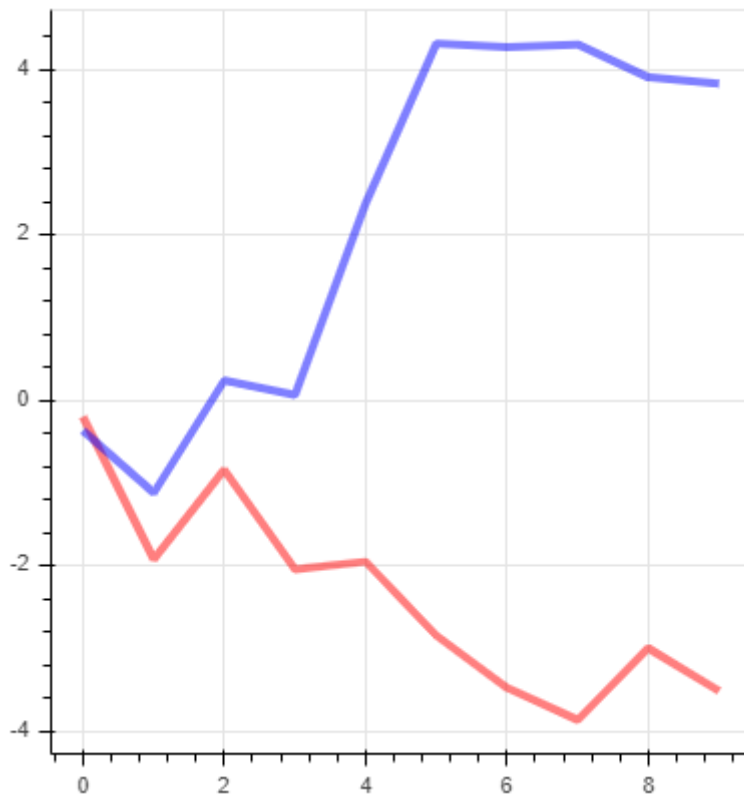
```
p = figure(plot_width=400, plot_height=400)
p.step([1, 2, 3, 4, 5], [6, 7, 4, 3, 5], line_width=2, mode="center")
show(p)
```



▼ 다중 라인(Multiple Lines)

```
p = figure(plot_width=400, plot_height=400)
p.multi_line([np.arange(10), np.arange(10)],
             [np.random.randn(10).cumsum(), np.random.randn(10).cumsum()],
             color=["red", "blue"], alpha=[0.5, 0.5], line_width=4)
show(p)
```

snow(p)



▼ 스택 라인(Stacked Lines)

```
from bokeh.models import ColumnDataSource

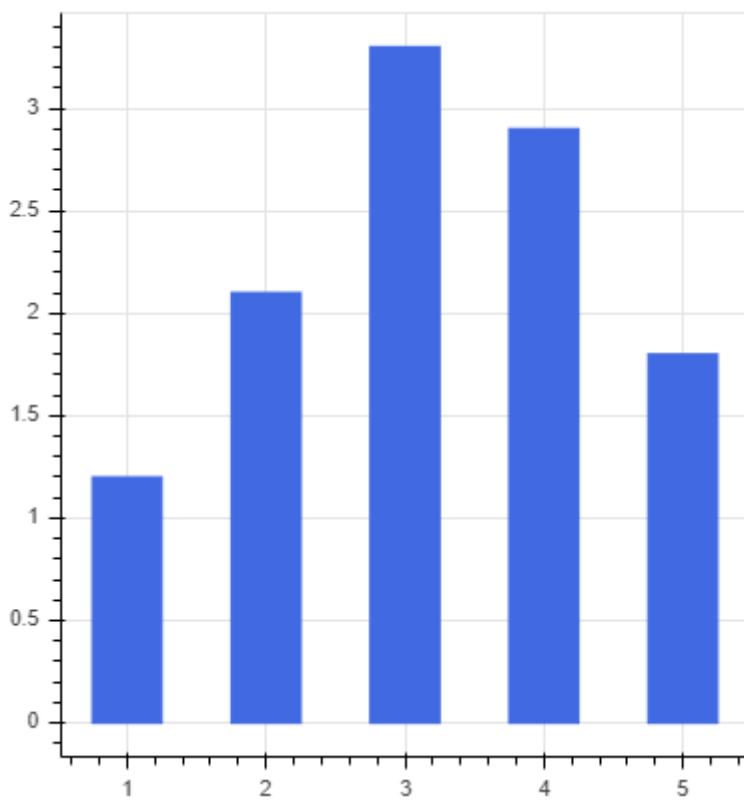
source = ColumnDataSource(data=dict(
    x = np.arange(50),
    y1 = np.random.randn(50).cumsum(),
    y2 = np.random.randn(50).cumsum(),
))

p = figure(plot_width=400, plot_height=400)
p.vline_stack(['y1', 'y2'], x='x', source=source)
show(p)
```

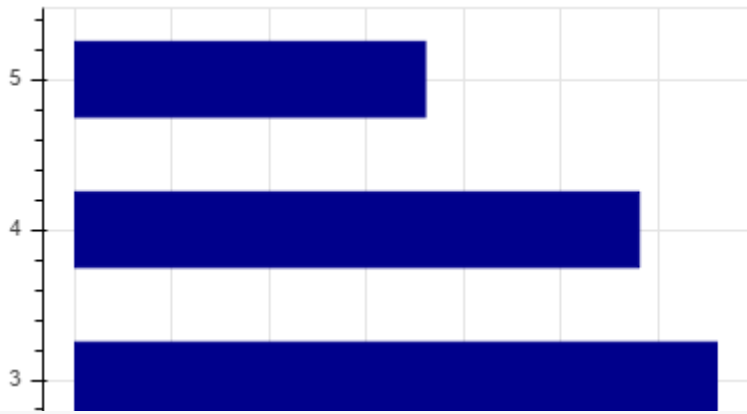



▼ 막대와 사각형(Bar & Rectangle)

```
p = figure(plot_width=400, plot_height=400)
p.vbar(x=[1, 2, 3, 4, 5], width=0.5, bottom=0,
       top=[1.2, 2.1, 3.3, 2.9, 1.8], color="royalblue")
show(p)
```



```
p = figure(plot_width=400, plot_height=400)
p.hbar(y=[1, 2, 3, 4, 5], height=0.5, left=0,
       right=[1.2, 2.1, 3.3, 2.9, 1.8], color="darkblue")
show(p)
```



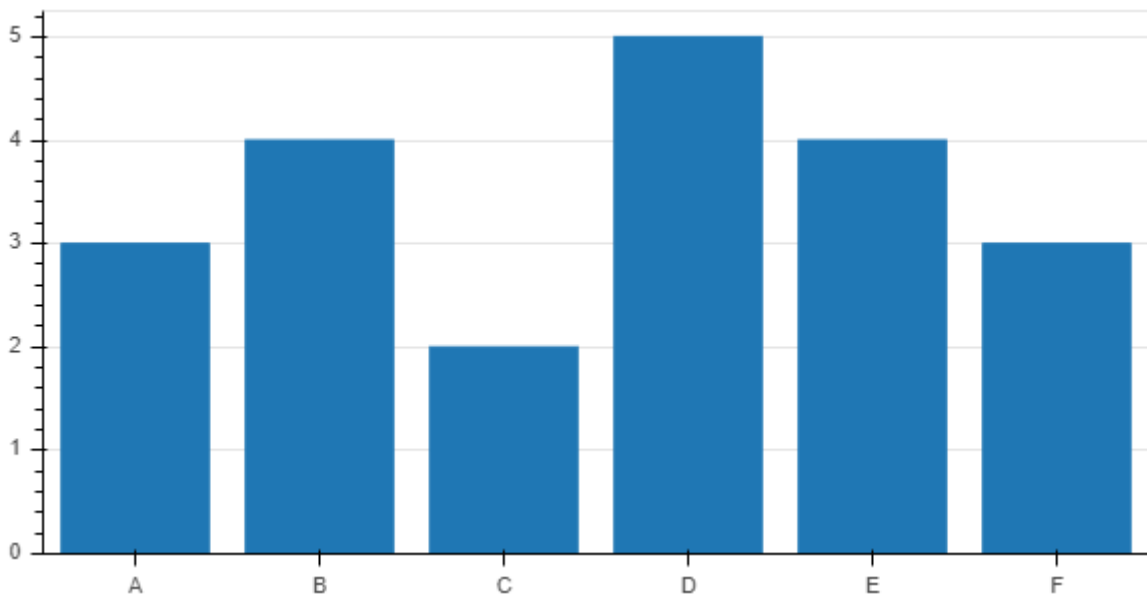
```

labels = ['A', 'B', 'C', 'D', 'E', 'F']

p = figure(x_range=labels, plot_height=300)
p.vbar(x=labels, top=[3, 4, 2, 5, 4, 3], width=0.8)
p.xgrid.grid_line_color = None
p.y_range.start = 0

show(p)

```



▼ 스택 막대(Stacked Bars)

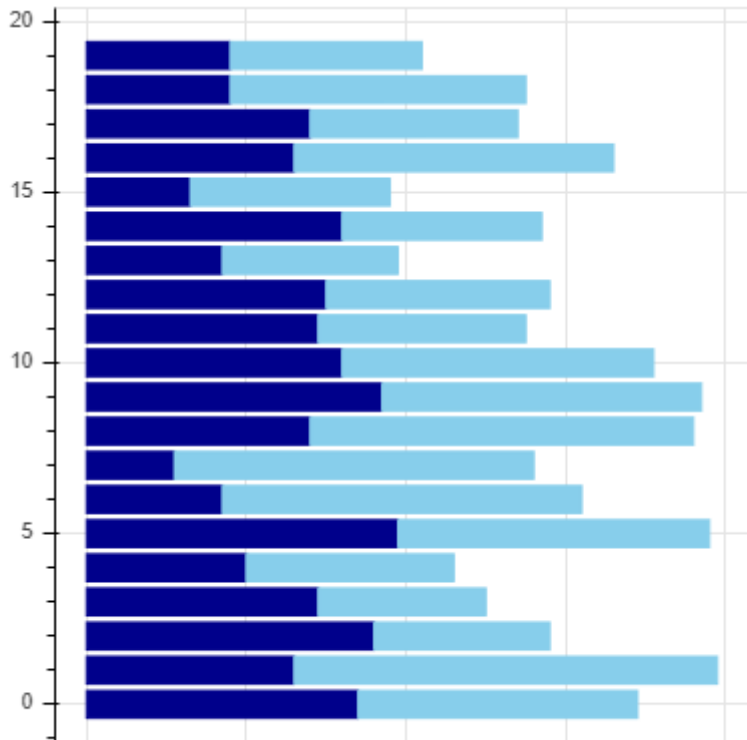
```

source = ColumnDataSource(data=dict(
    y = np.arange(20),
    x1 = np.random.randint(10, 40, 20),
    x2 = np.random.randint(20, 60, 20)
))

p = figure(plot_width=400, plot_height=400)
p.hbar_stack(['x1', 'x2'], y='y', height=0.8,
            color=('darkblue', 'skyblue'), source=source)

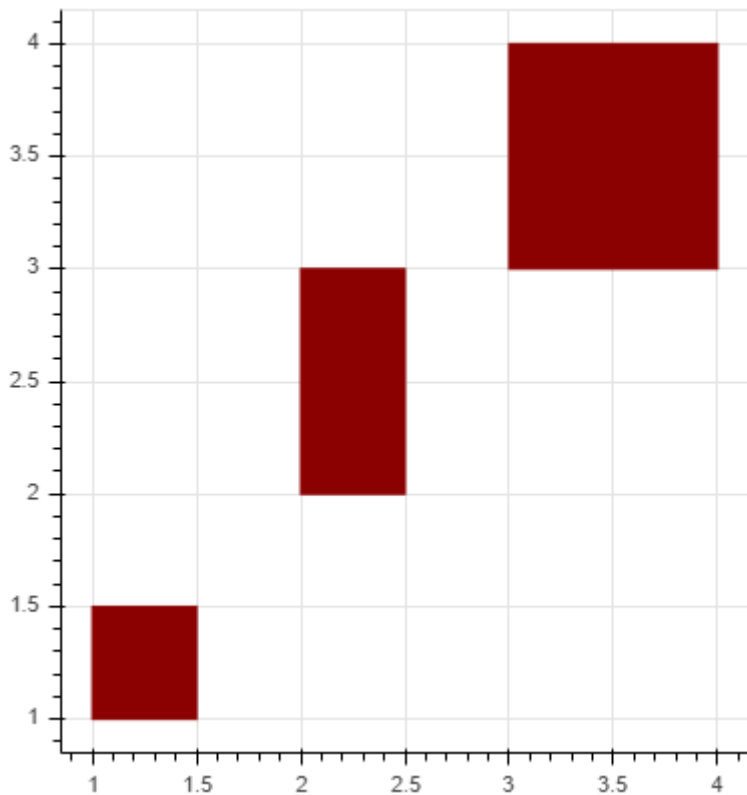
show(p)

```



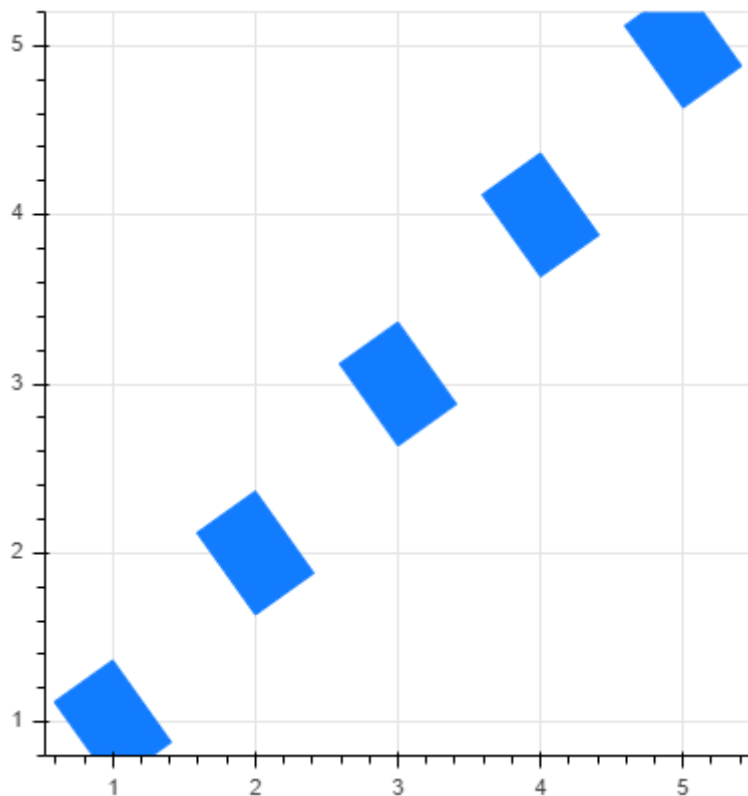
▼ 사각형(Rectangles)

```
p = figure(plot_width=400, plot_height=400)
p.quad(top=[1.5, 3, 4], bottom=[1, 2, 3],
        left=[1, 2, 3], right=[1.5, 2.5, 4], color="darkred")
show(p)
```



```
p = figure(plot_width=400, plot_height=400)
p.rect(x=[1, 2, 3, 4, 5], y=[1, 2, 3, 4, 5],
        width=0.5, height=50, color="#127CFF",
        stroke=100, height_units="screen")
```

```
angle=120, height_units= screen )
show(p)
```

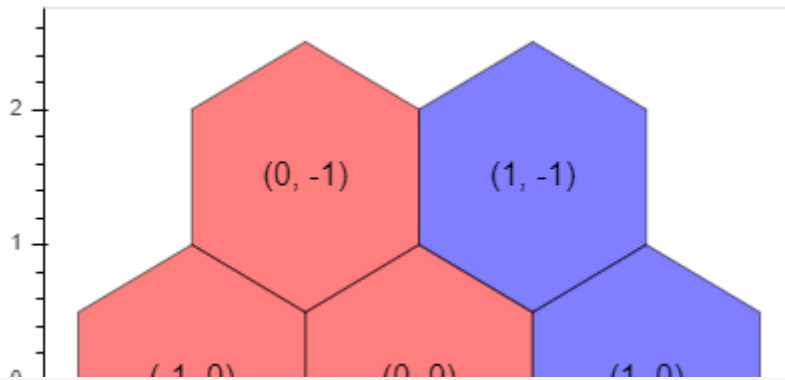


▼ 육각 타일(Hex Tiles)

```
from bokeh.util.hex import axial_to_cartesian

q = np.array([0, 0, 0, -1, -1, 1, 1])
r = np.array([0, -1, 1, 0, 1, -1, 0])

p = figure(plot_width=400, plot_height=400, toolbar_location=None)
p.grid.visible = False
p.hex_tile(q, r, size=1, fill_color=["red"]*4 + ["blue"]*3,
           line_color="black", alpha=0.5)
x, y = axial_to_cartesian(q, r, 1, "pointytop")
p.text(x, y, text=["(%d, %d)" % (q, r) for (q, r) in zip(q, r)],
       text_baseline="middle", text_align="center")
show(p)
```



```
from bokeh.palettes import Blues256
from bokeh.util.hex import hexbin

n = 10000
x = np.random.standard_normal(n)
y = np.random.standard_normal(n)

bins = hexbin(x, y, 0.1)
color = [Blues256[int(i)] for i in bins.counts/max(bins.counts) * 255]

p = figure(match_aspect=True, background_fill_color='#083573')
p.grid.visible = False
p.hex_tile(bins.q, bins.r, size=0.1, line_color=None, fill_color=color)

show(p)
```



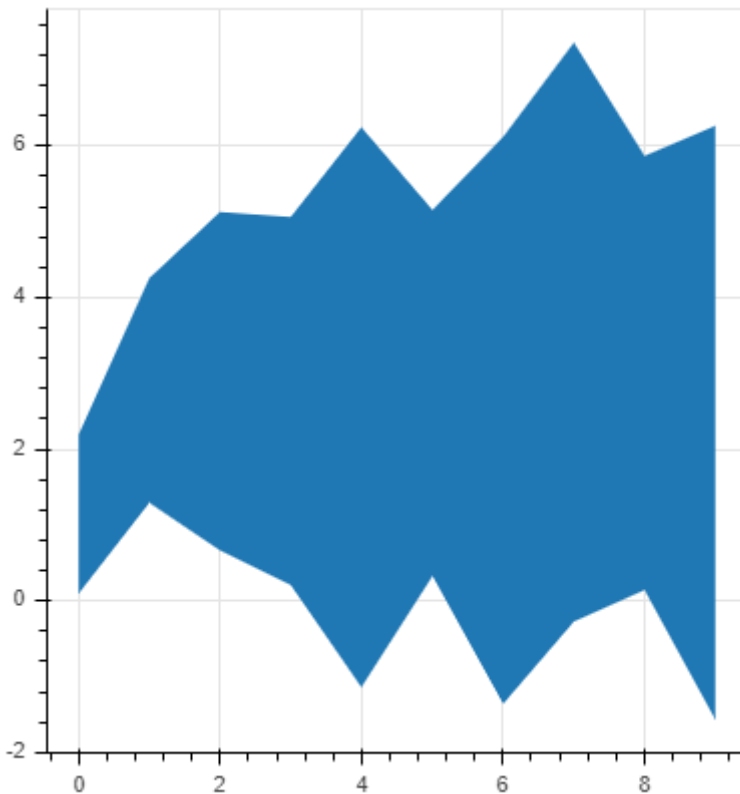
▼ 영역(Areas)



▼ 단일 영역(Single Areas)



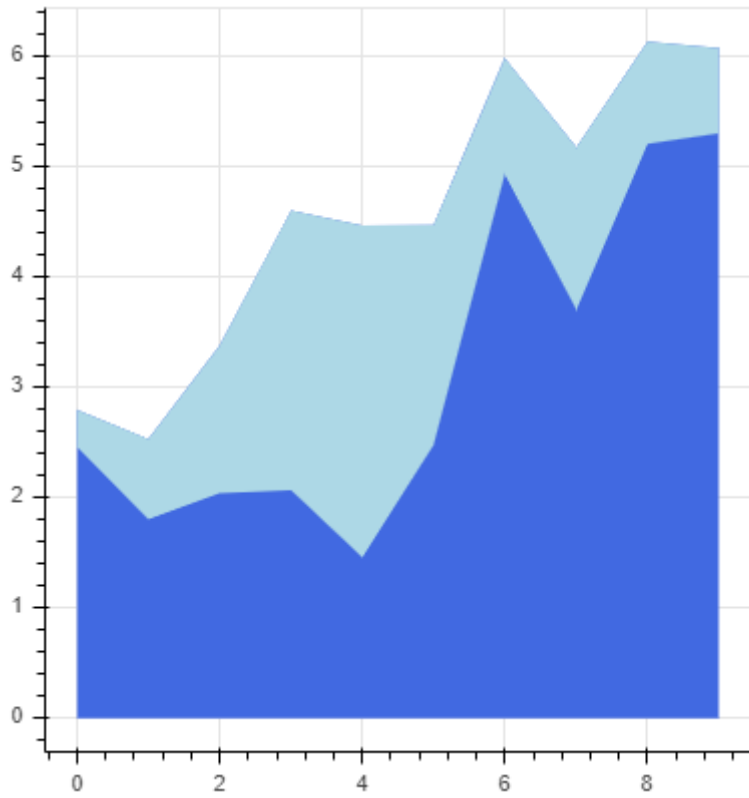
```
p = figure(plot_width=400, plot_height=400)
p.varea(x=np.arange(10),
        y1=np.random.randn(10).cumsum(),
        y2=np.random.randn(10).cumsum())
show(p)
```



▼ 스택 영역(Stacked Areas)

```
source = ColumnDataSource(data=dict(
    x=np.arange(10),
    y1=np.random.randn(10).cumsum()+3,
    y2=np.random.randn(10).cumsum()
))

p = figure(plot_width=400, plot_height=400)
p.varea_stack(['y1', 'y2'], x='x', color=("royalblue", "lightblue"), source=source)
show(p)
```



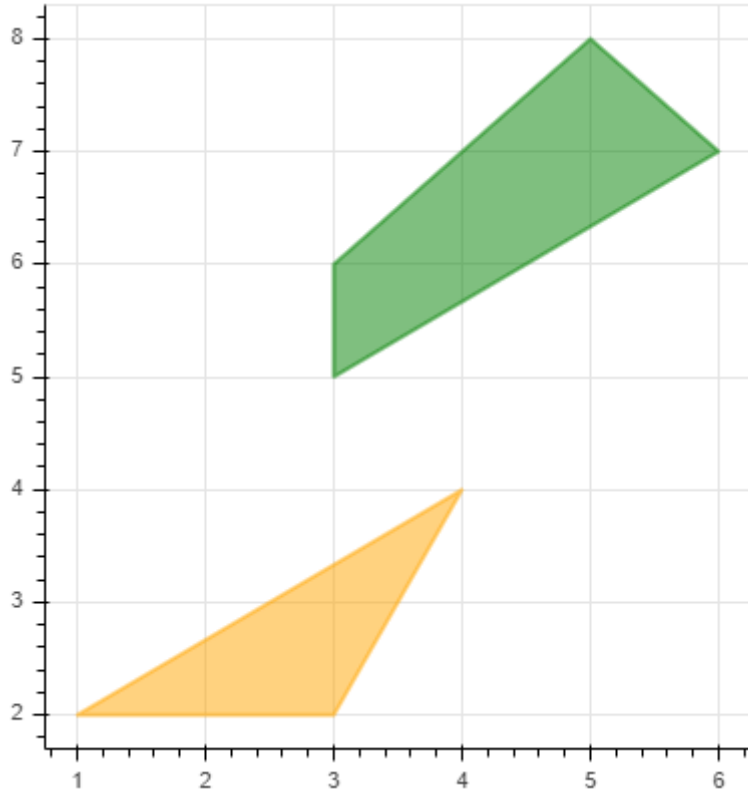
▼ 패치와 폴리곤(Patches and Polygons)

▼ 단일 패치(Single Patches)

```
p = figure(plot_width=400, plot_height=400)
p.patch([1, 2, 3, 4, 5], [5, 7, 8, 5, 2], alpha=0.5, line_width=1)
show(p)
```

▼ 다중 패치(Multiple Patches)

```
p = figure(plot_width=400, plot_height=400)
p.patches([[1, 3, 4], [3, 3, 5, 6]], [[2, 2, 4], [5, 6, 8, 7]],
          color=["orange", "green"], alpha=[0.5, 0.5], line_width=2)
show(p)
```

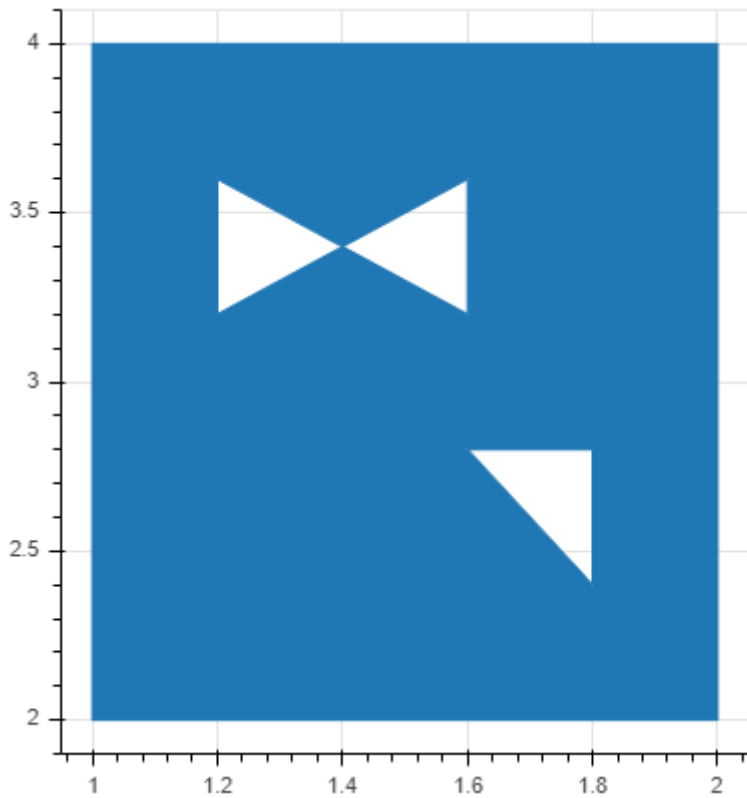


▼ 폴리곤(Polygons)

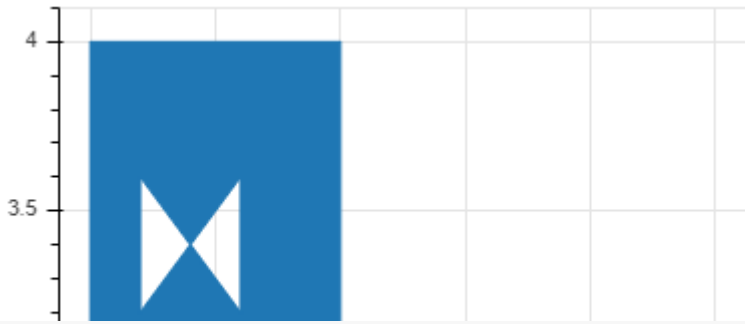
```
p = figure(plot_width=400, plot_height=400)
p.multi_polygons(xs=[[[1, 1, 2, 2]]],
                  ys=[[[2, 4, 4, 2]]])
show(p)
```




```
p = figure(plot_width=400, plot_height=400)
p.multi_polygons(xs=[[ [1, 1, 2, 2], [1.2, 1.6, 1.6, 1.2], [1.8, 1.8, 1.6] ]],
                 ys=[[ [2, 4, 4, 2], [3.2, 3.6, 3.2, 3.6], [2.4, 2.8, 2.8] ]])
show(p)
```



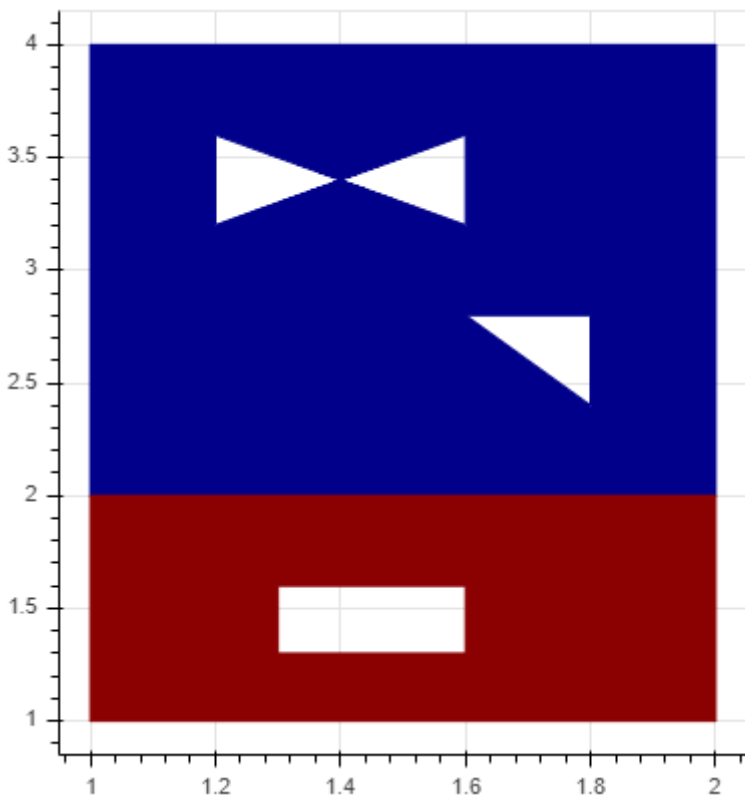
```
p = figure(plot_width=400, plot_height=400)
p.multi_polygons(xs=[[ [1, 1, 2, 2], [1.2, 1.6, 1.6, 1.2], [1.8, 1.8, 1.6] ], [ [2.5, 3.5, 3] ]],
                 ys=[[ [2, 4, 4, 2], [3.2, 3.6, 3.2, 3.6], [2.4, 2.8, 2.8] ], [ [2, 2, 3] ]])
show(p)
```



```

p = figure(plot_width=400, plot_height=400)
p.multi_polygons(xs=[[ [1, 1, 2, 2], [1.2, 1.6, 1.6, 1.2], [1.8, 1.8, 1.6] ]],
                 [[ [1, 2, 2, 1], [1.3, 1.3, 1.6, 1.6] ]]),
                 ys=[[ [2, 4, 4, 2], [3.2, 3.6, 3.2, 3.6], [2.4, 2.8, 2.8] ]],
                 [[ [1, 1, 2, 2], [1.3, 1.6, 1.6, 1.3] ]]),
                 color=['darkblue', 'darkred'])
show(p)

```

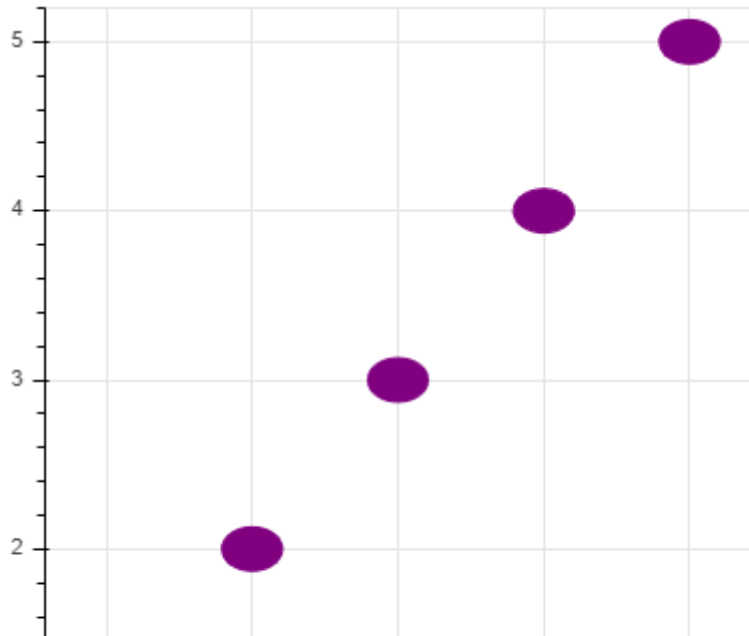


▼ 계란형과 타원형(Ovals and Ellipses)

```

p = figure(plot_width=400, plot_height=400)
p.oval(x=[1, 2, 3, 4, 5], y=[1, 2, 3, 4, 5],
       width=0.4, height=30, color="purple",
       angle=np.pi/2, height_units="screen")
show(p)

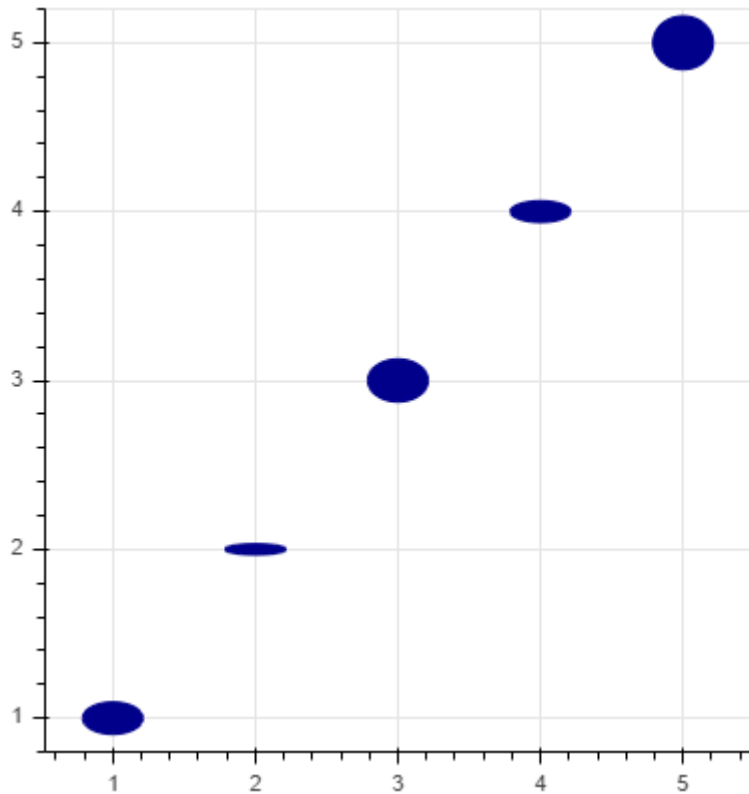
```



```

p = figure(plot_width=400, plot_height=400)
p.oval(x=[1, 2, 3, 4, 5], y=[1, 2, 3, 4, 5],
       width=[0.3, 0.1, 0.4, 0.2, 0.5], height=30, color="darkblue",
       angle=np.pi/2, height_units="screen")
show(p)

```



▼ 이미지(Images)

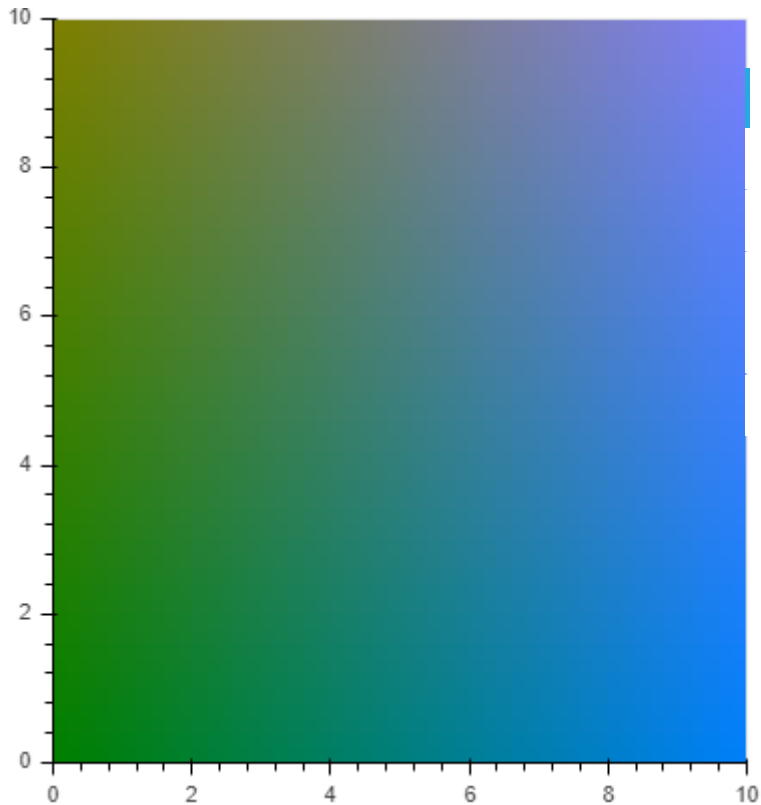
```

N = 50
img = np.empty((N, N), dtype=np.uint32)
view = img.view(dtype=np.uint8).reshape((N, N, 4))
for i in range(N):
    for i in range(N):

```

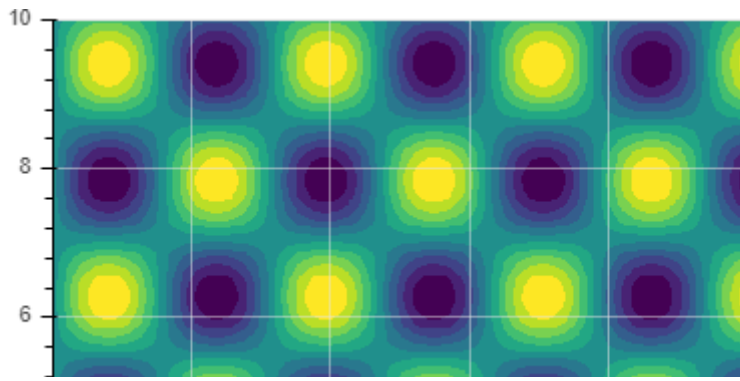
```
view[i, j, 0] = int(127 * i / N)
view[i, j, 1] = 127
view[i, j, 2] = int(255 * j / N)
view[i, j, 3] = 255
```

```
p = figure(plot_width=400, plot_height=400,
           x_range=(0, 10), y_range=(0, 10))
p.image_rgba(image=[img], x=[0], y=[0], dw=[10], dh=[10])
show(p)
```



```
l = np.linspace(0, 20, 400)
X, Y = np.meshgrid(l, l)
d = np.sin(X) * np.cos(Y)
```

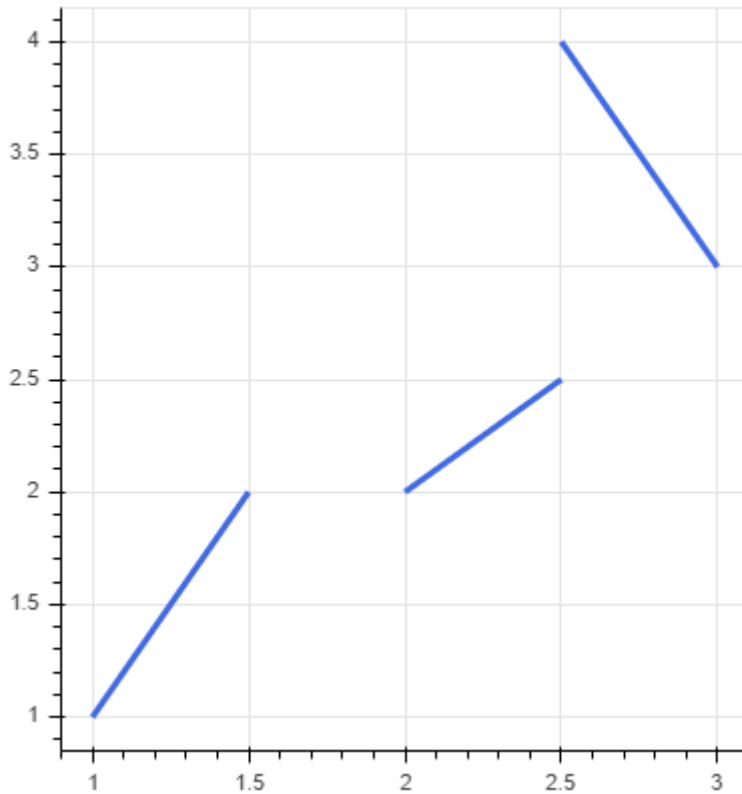
```
p = figure(plot_width=400, plot_height=400)
p.x_range.range_padding = p.y_range.range_padding = 0
p.image(image=[d], x=0, y=0, dw=10, dh=10, palette="Viridis11", level="image")
p.grid.grid_line_width = 0.5
show(p)
```



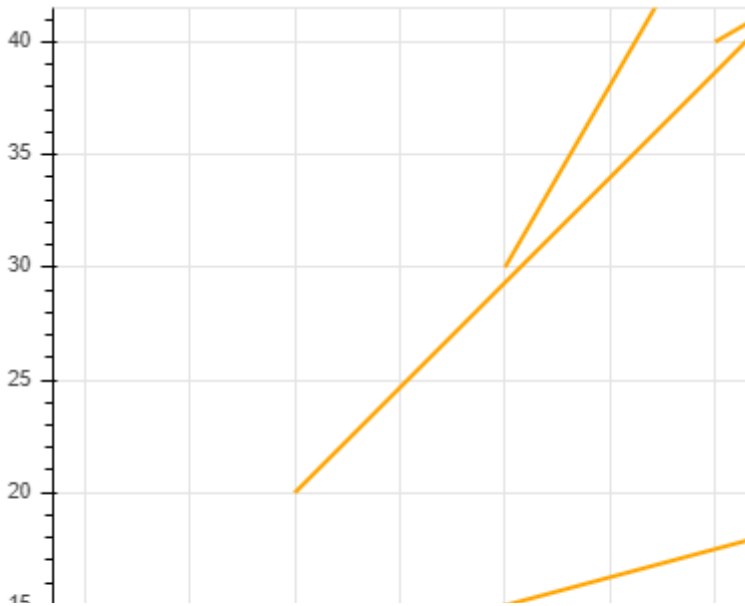
▼ 세그먼트와 광선(Segments and Rays)



```
p = figure(plot_width=400, plot_height=400)
p.segment(x0=[1, 2, 3], y0=[1, 2, 3],
          x1=[1.5, 2.5, 2.5], y1=[2, 2.5, 4],
          color="royalblue", line_width=3)
show(p)
```

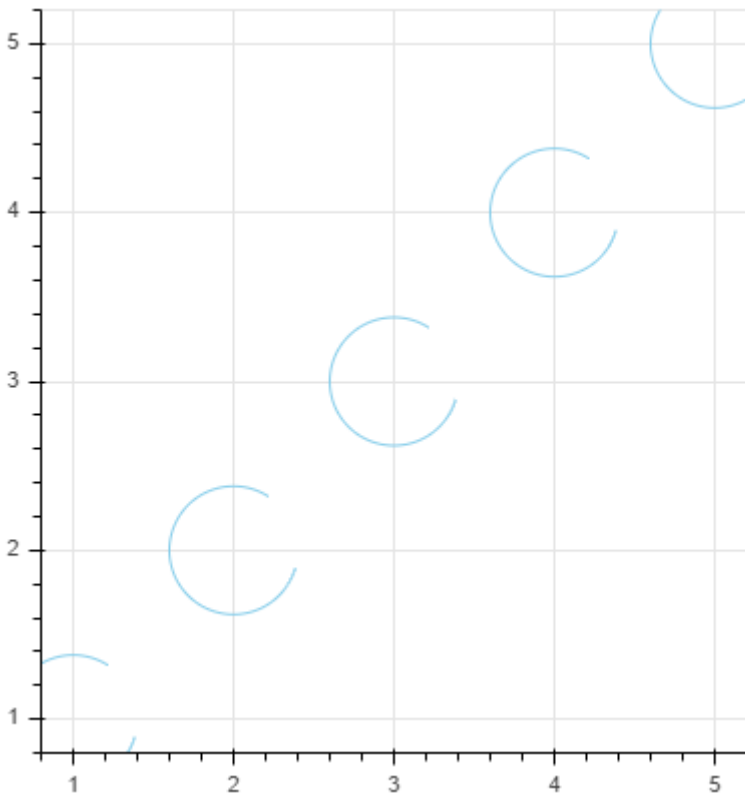


```
p = figure(plot_width=400, plot_height=400)
p.ray(x=[10, 20, 30, 40], y=[10, 20, 30, 40],
      length=50, angle=[15, 45, 60, 30],
      angle_units="deg", color="orange", line_width=2)
show(p)
```

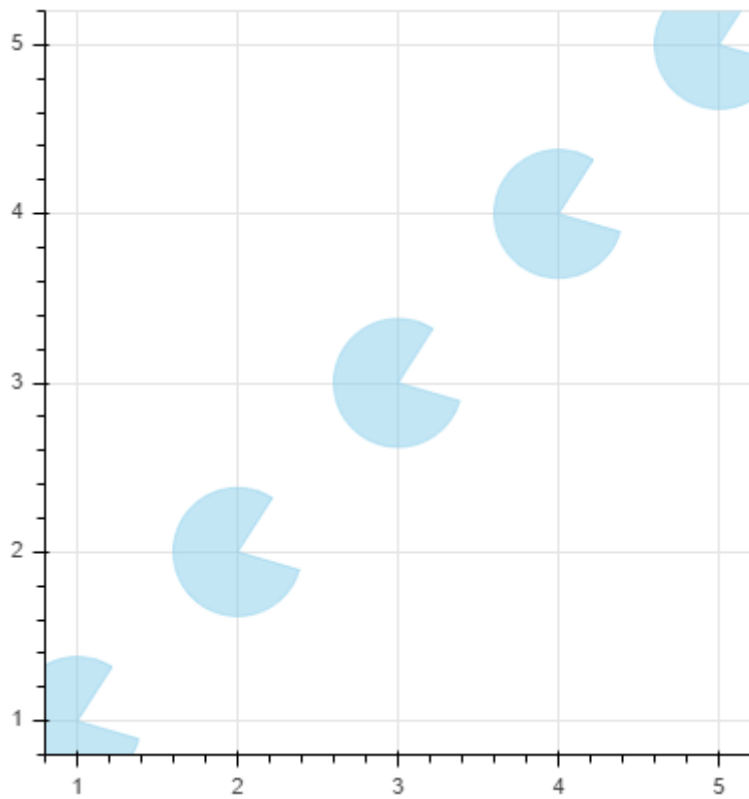


▼ 썩기와 원호(Wedges and Arcs)

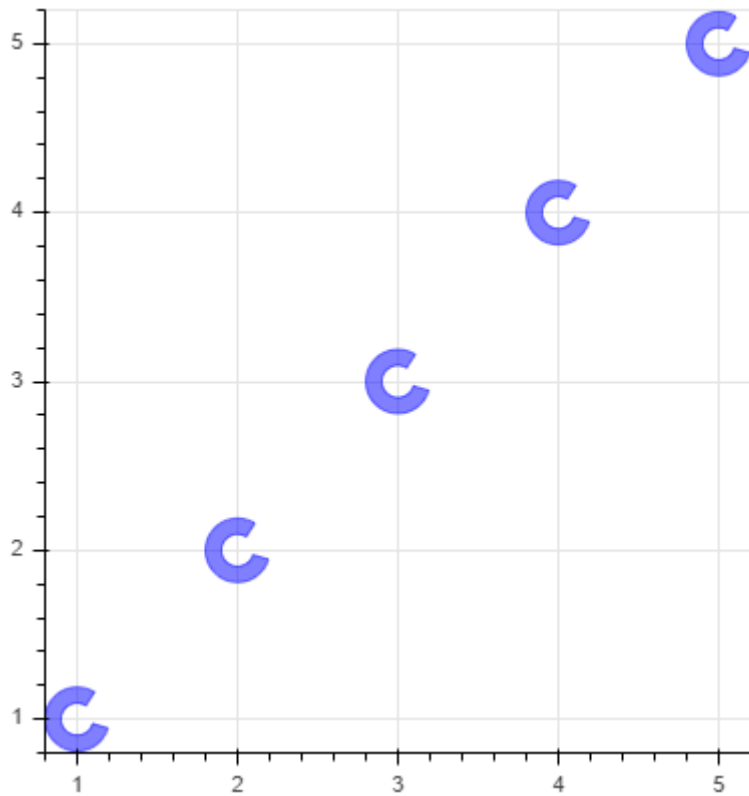
```
p = figure(plot_width=400, plot_height=400)
p.arc(x=[1, 2, 3, 4, 5], y=[1, 2, 3, 4, 5],
      radius=0.4, start_angle=1, end_angle=6, color="skyblue")
show(p)
```



```
p = figure(plot_width=400, plot_height=400)
p.wedge(x=[1, 2, 3, 4, 5], y=[1, 2, 3, 4, 5],
        radius=0.4, start_angle=6, end_angle=1,
        color="skyblue", alpha=0.5, direction="clock")
show(p)
```

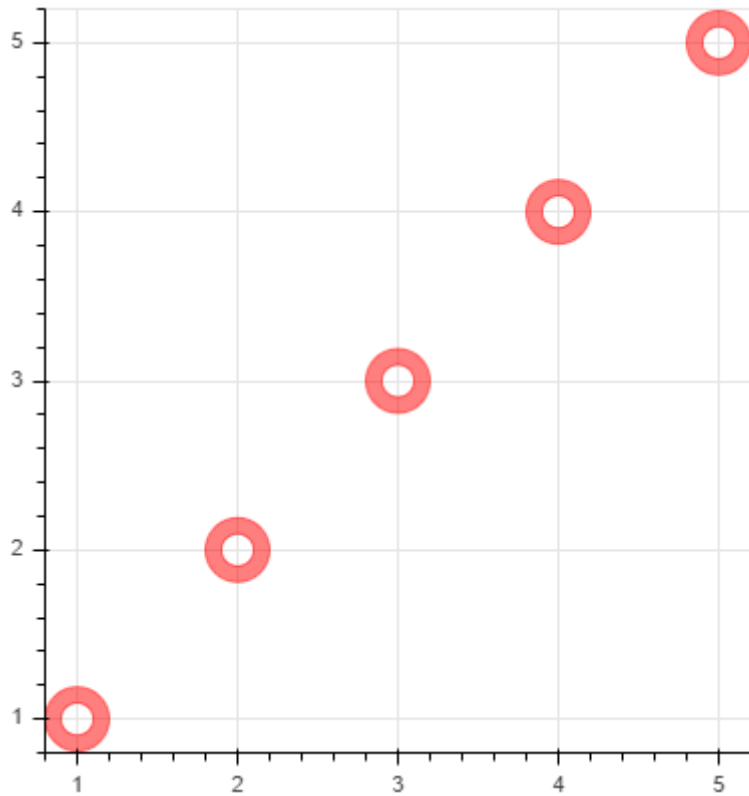


```
p = figure(plot_width=400, plot_height=400)
p.annular_wedge(x=[1, 2, 3, 4, 5], y=[1, 2, 3, 4, 5],
               inner_radius=0.1, outer_radius=0.2, start_angle=1, end_angle=6,
               color="blue", alpha=0.5)
show(p)
```



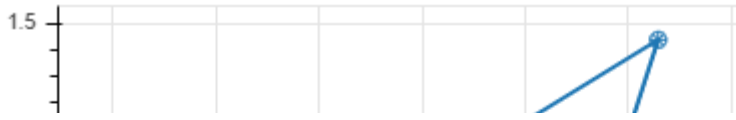
```
p = figure(plot_width=400, plot_height=400)
p.annulus(x=[1, 2, 3, 4, 5], y=[1, 2, 3, 4, 5],
          inner_radius=0.1, outer_radius=0.2,
          color="red", alpha=0.5)
```

```
show(p)
```



▼ 여러 도형 결합(Combining Multiple Glyphs)

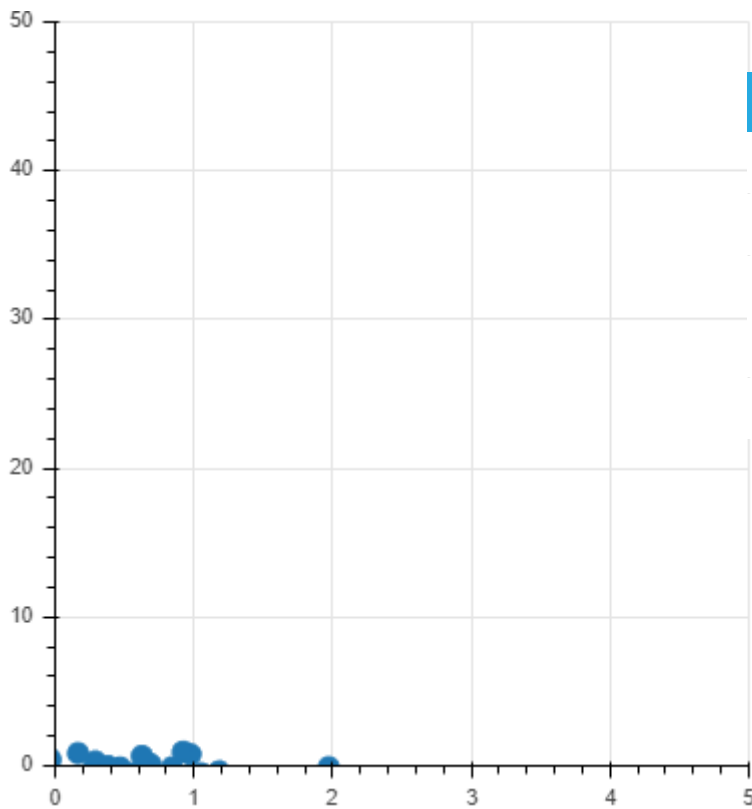
```
x = np.random.randn(5)
y = np.random.randn(5)
p = figure(plot_width=400, plot_height=400)
p.circle(x, y, fill_color="white", size=8)
p.asterisk(x, y, fill_color="white", size=8)
p.line(x, y, line_width=2)
show(p)
```

▼ 범위 지정(Setting Ranges)

```
from bokeh.models import Range1d

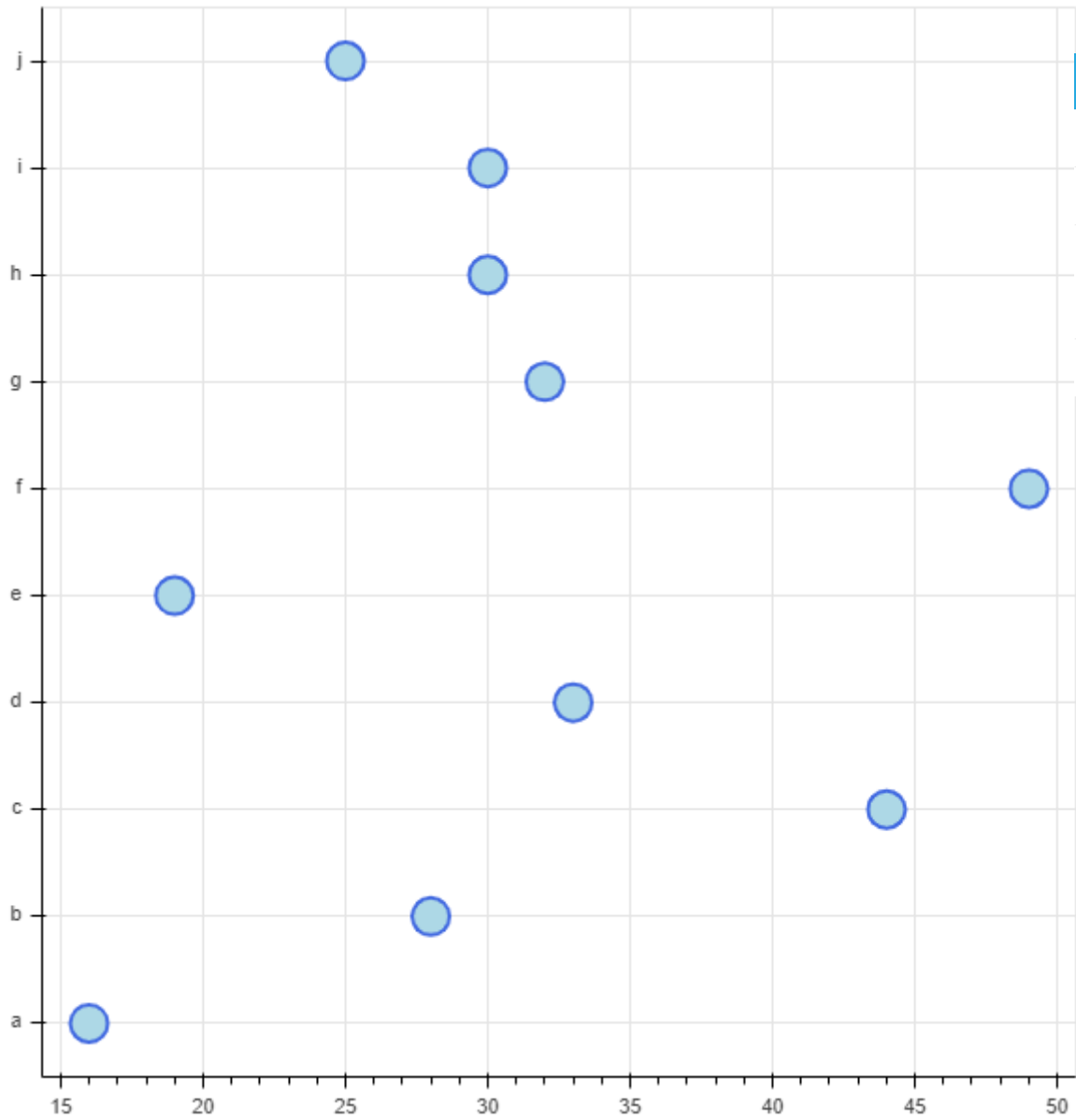
x = np.random.randn(50)
y = np.random.randn(50)
p.x_range = Range1d(0, 5)
p = figure(plot_width=400, plot_height=400, x_range=(0, 5))
p.y_range = Range1d(0, 50)
p.circle(x, y, size=10)
show(p)
```



▼ 축 유형 지정(Specifying Axis Types)

▼ 범주형 축(Categorical Axes)

```
factors = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
x = np.random.randint(10, 50, size=10)
p = figure(y_range=factors)
p.circle(x, factors, size=20, fill_color="lightblue", line_color="royalblue", line_width=2)
show(p)
```



▼ 날짜/시간 축(Datetime Axes)

```

from bokeh.sampledata.stocks import AAPL, GOOG

aapl = pd.DataFrame(AAPL)
goog = pd.DataFrame(GOOG)
aapl['date'] = pd.to_datetime(aapl['date'])
goog['date'] = pd.to_datetime(goog['date'])
p = figure(plot_width=800, plot_height=250, x_axis_type="datetime")
p.line(aapl['date'], aapl['close'], color='red', alpha=0.5)
p.line(goog['date'], goog['close'], color='blue', alpha=0.5)
show(p)

```



```

from bokeh.sampledata.stocks import AAPL, IBM, MSFT, GOOG
from bokeh.palettes import Spectral4

p = figure(plot_width=800, plot_height=250, x_axis_type="datetime")

for data, name, color in zip([AAPL, IBM, MSFT, GOOG], ["AAPL", "IBM", "MSFT", "GOOG"], Spectral4):
    datetime = np.asarray(data['date'], dtype=np.datetime64)
    value = np.asarray(data['close'])
    p.line(datetime, value, line_width=1, color=color, alpha=0.8, legend_label=name)

p.legend.location = "top_left"
p.legend.click_policy = "hide"

show(p)

```



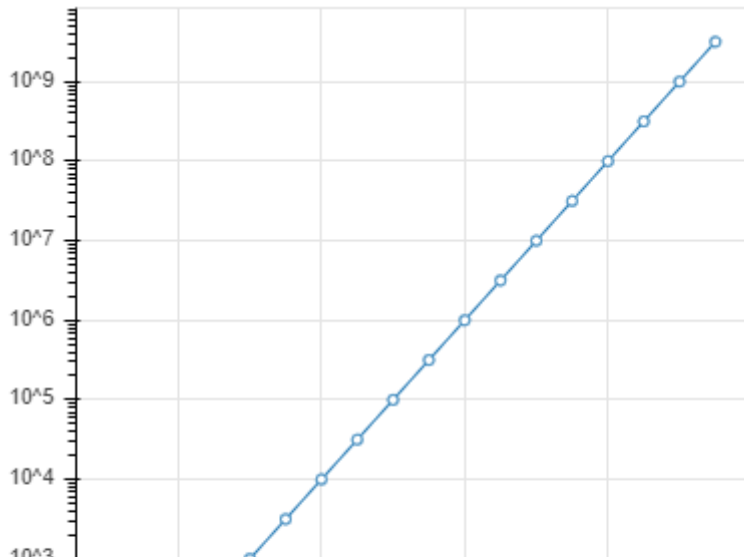
▼ 로그스케일(Log Scale Axes)

```

x = np.arange(1, 10, 0.5)
y = [10 ** xx for xx in x]

p = figure(plot_width=400, plot_height=400, y_axis_type="log")
p.line(x, y, line_width=1)
p.circle(x, y, fill_color="white", size=5)
show(p)

```



▼ 스타일(Style)



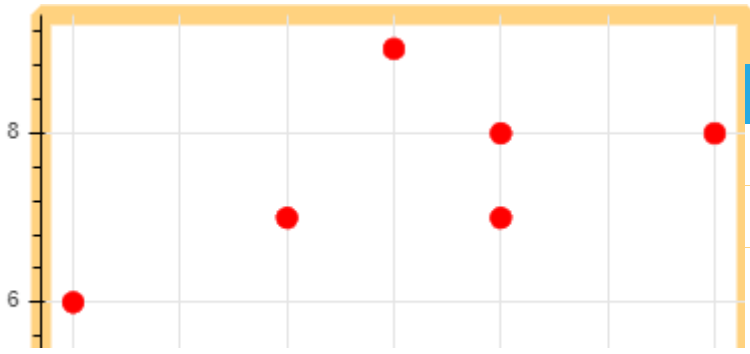
색상(Colors)

- HTML 색상
- RGB의 16진수 표현
- 0~255 정수값의 (r, g, b) 튜플
- 0~1 사이의 부동소수점 a가 추가된 (r, g, b, a) 튜플

▼ 플롯(Plots)

```
x = np.random.randint(1, 10, 10)
y = np.random.randint(1, 10, 10)

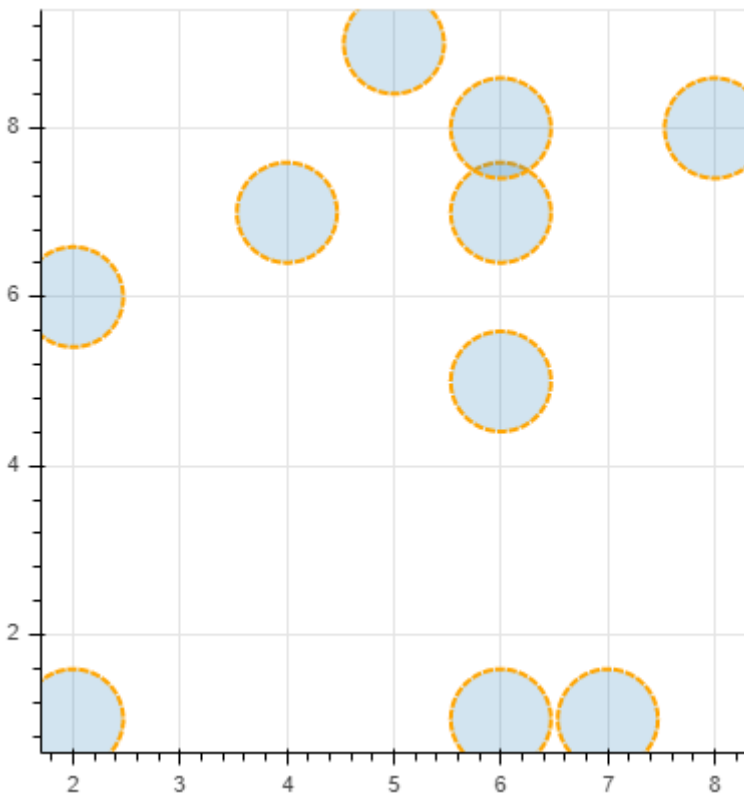
p = figure(plot_width=400, plot_height=400)
p.outline_line_width = 10
p.outline_line_alpha = 0.5
p.outline_line_color = "orange"
p.circle(x, y, color="red", size=10)
show(p)
```



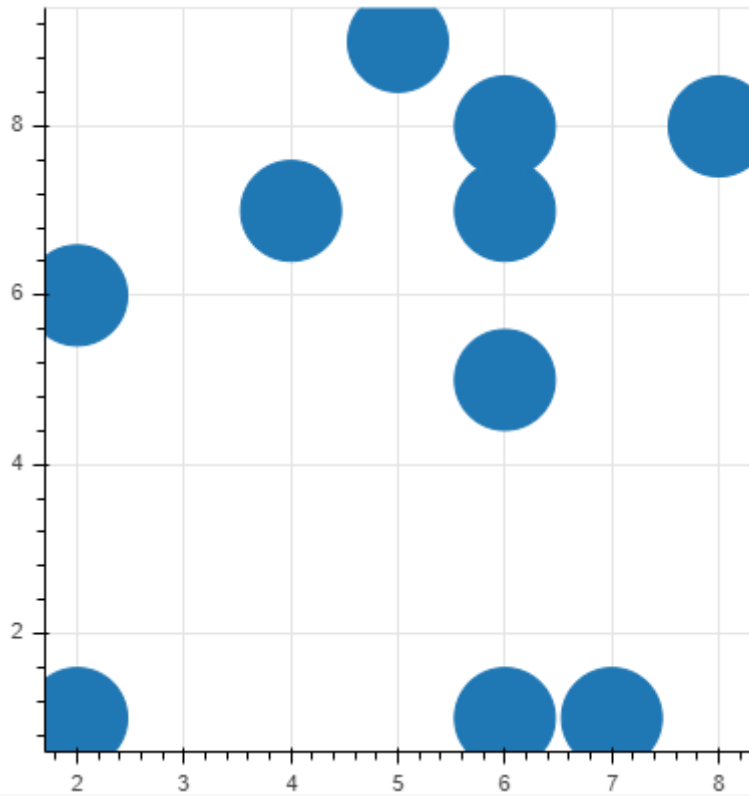
▼ 글리프(Glyphs)



```
p = figure(plot_width=400, plot_height=400)
r = p.circle(x, y)
r.glyph.size = 50
r.glyph.fill_alpha = 0.2
r.glyph.line_color = "orange"
r.glyph.line_dash = [5, 1]
r.glyph.line_width = 2
show(p)
```



```
p = figure(plot_width=400, plot_height=400, tools="tap")
r = p.circle(x, y, size=50,
             selection_color="royalblue",
             nonselection_fill_alpha=0.3,
             nonselection_fill_color="gray",
             nonselection_line_color="royalblue",
             nonselection_line_alpha=0.8)
show(p)
```



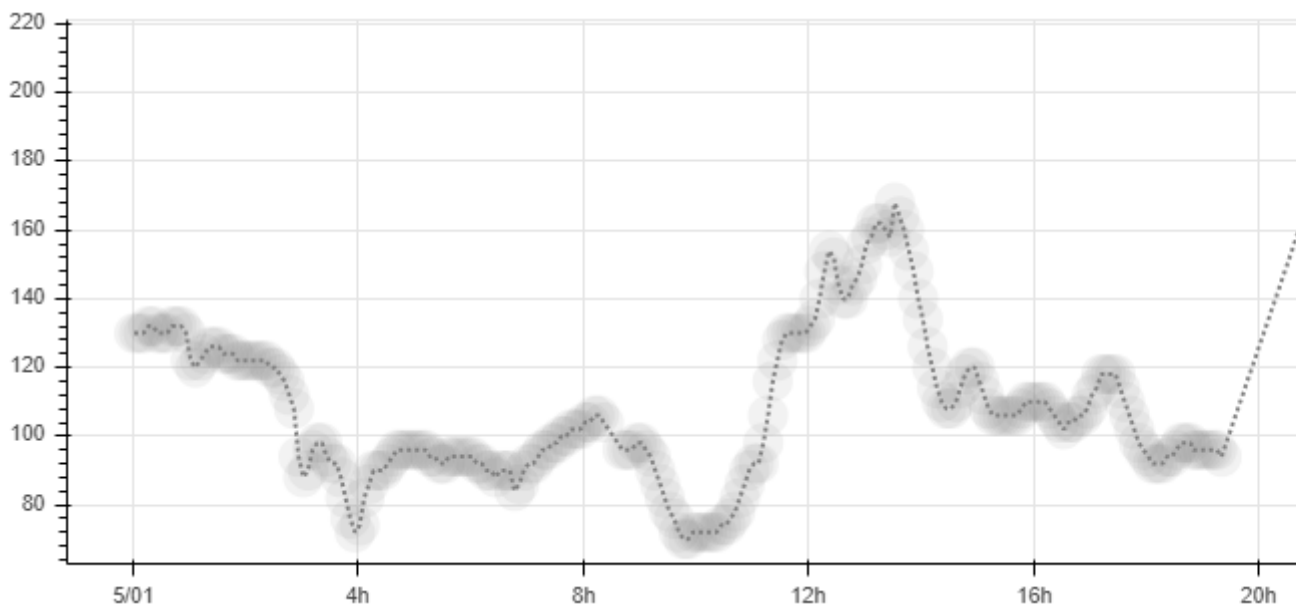
```

from bokeh.models.tools import HoverTool
from bokeh.sampledata.glucose import data

subset = data.loc['2010-05-01']
x, y = subset.index.to_series(), subset['glucose']

p = figure(width=800, height=300, x_axis_type="datetime")
p.line(x, y, line_dash="2 2", line_width=2, color="gray")
cr = p.circle(x, y, size=20,
              fill_color="gray", hover_fill_color="skyblue",
              fill_alpha=0.1, hover_alpha=0.4,
              line_color=None, hover_line_color="white")
p.add_tools(HoverTool(tooltips=None, renderers=[cr], mode="hline"))
show(p)

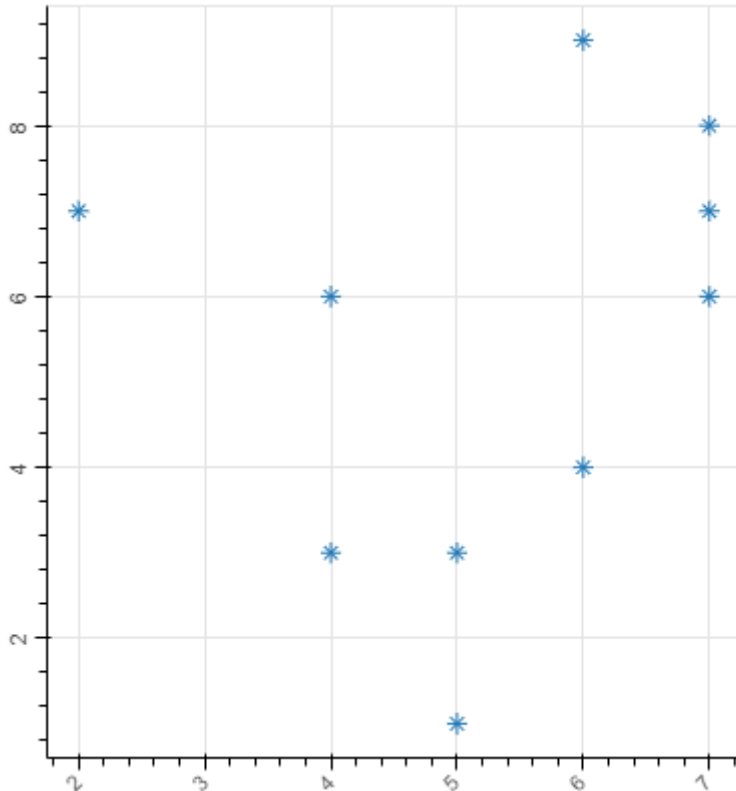
```



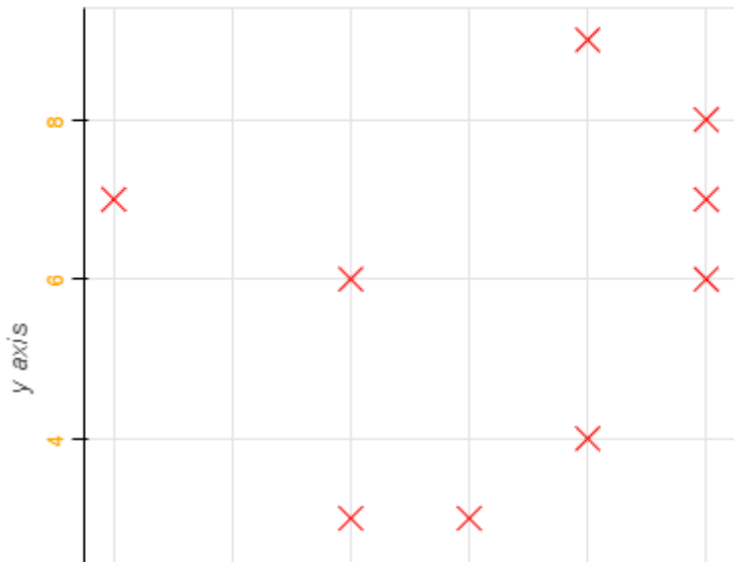
▼ 축(Axes)

```
x = np.random.randint(1, 10, 10)
y = np.random.randint(1, 10, 10)

p = figure(plot_width=400, plot_height=400)
p.asterisk(x, y, size=10, line_width=1)
p.xaxis.major_label_orientation = np.pi/4
p.yaxis.major_label_orientation = "vertical"
show(p)
```



```
p = figure(plot_width=400, plot_height=400)
p.x(x, y, size=12, color="red")
p.xaxis.axis_label = "x axis"
p.xaxis.axis_line_width = 2
p.xaxis.axis_line_color = "blue"
p.yaxis.axis_label = "y axis"
p.yaxis.major_label_text_color = "orange"
p.yaxis.major_label_orientation = "vertical"
p.axis.minor_tick_in = -5
p.axis.minor_tick_out = 5
show(p)
```



▼ 틱 라벨(Tick labels)

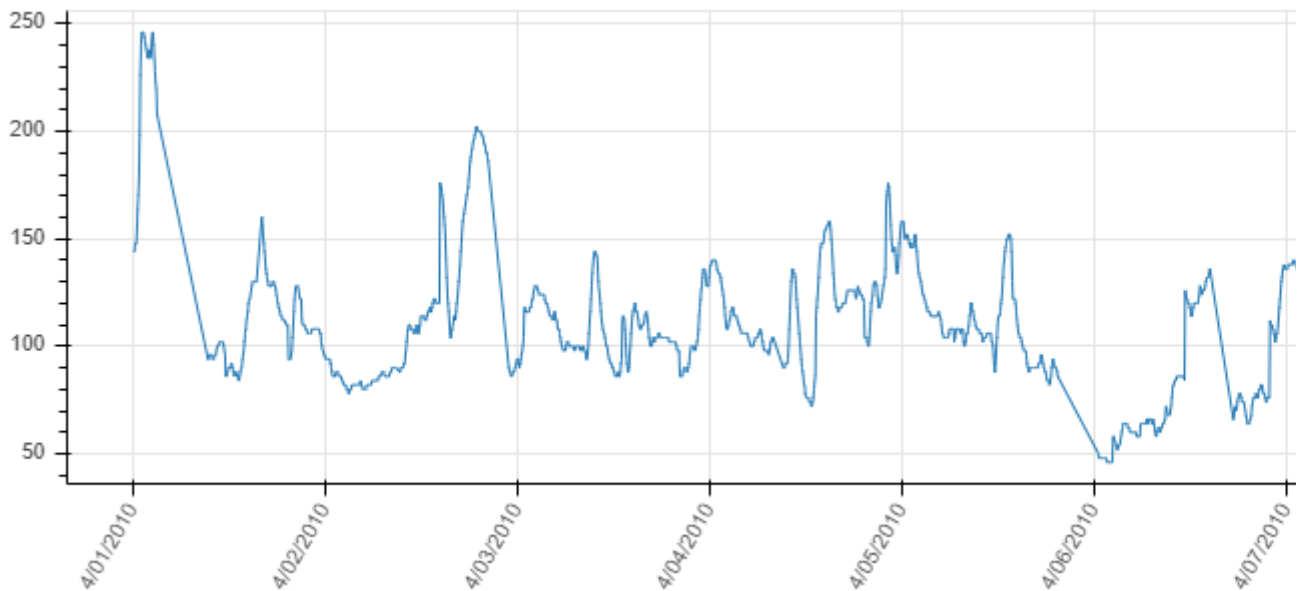
```

from bokeh.sampledata.glucose import data

week = data.loc['2010-04-01':'2010-04-07']

p = figure(x_axis_type="datetime", plot_height=300, plot_width=800)
p.xaxis.formatter.days = '%m/%d/%Y'
p.xaxis.major_label_orientation = np.pi/3
p.line(week.index, week.glucose)
show(p)

```



```

from bokeh.models import NumeralTickFormatter

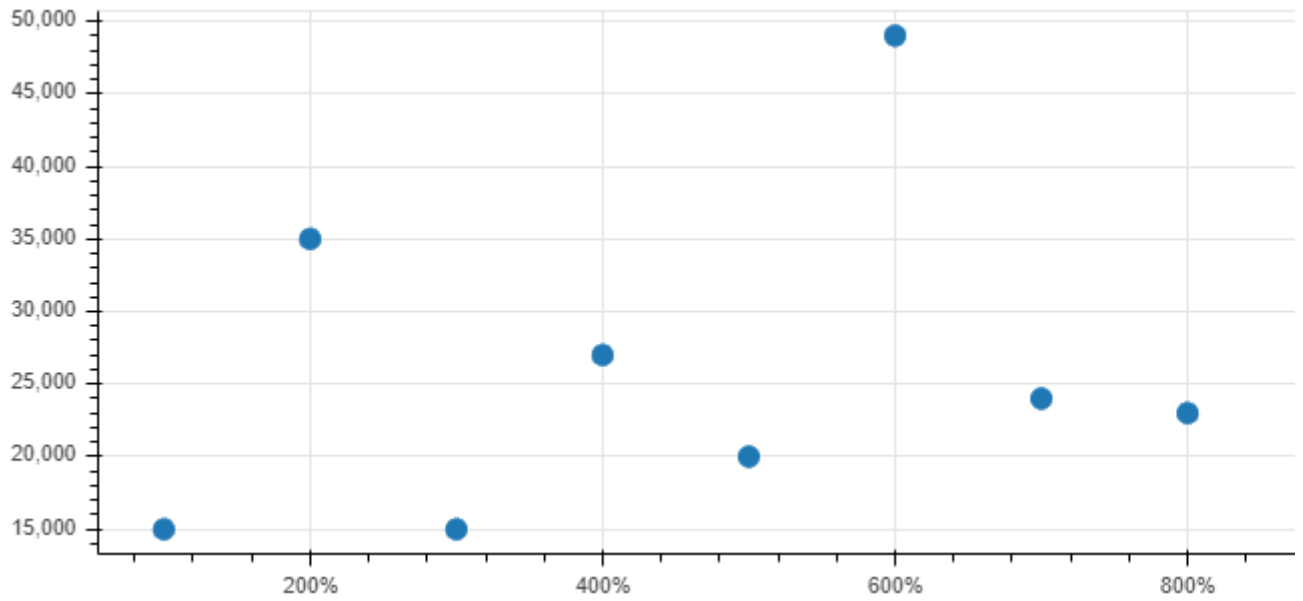
x = np.arange(1, 11)
y = [i * 1000 for i in np.random.randint(5, 50, 10)]

p = figure(plot_height=300, plot_width=800)
p.circle(x, y, size=10)
p.xaxis.formatter = NumeralTickFormatter(format="0%")

```

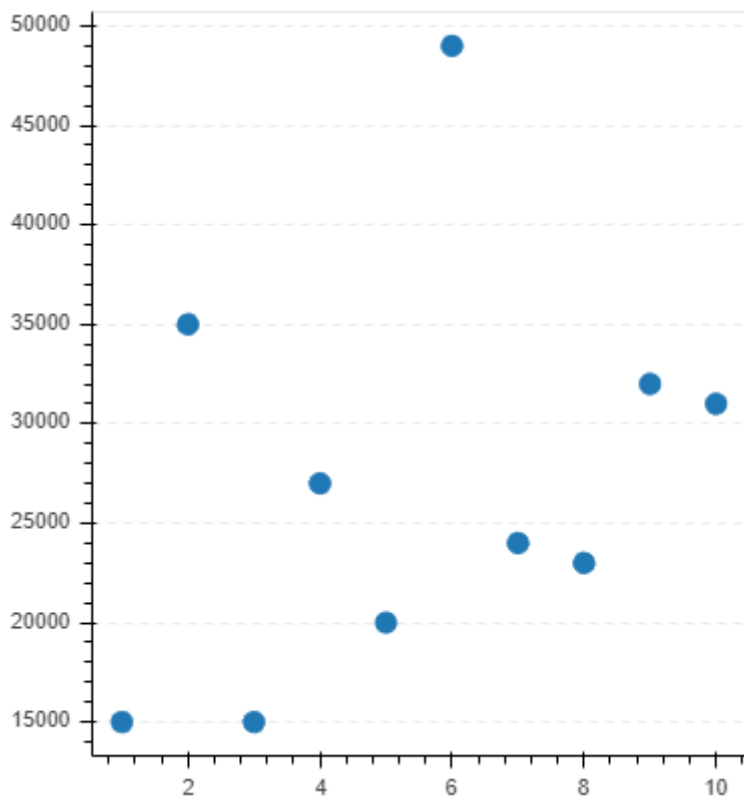


```
p.yaxis.formatter = NumeralltickFormatter(format="0,0")
show(p)
```



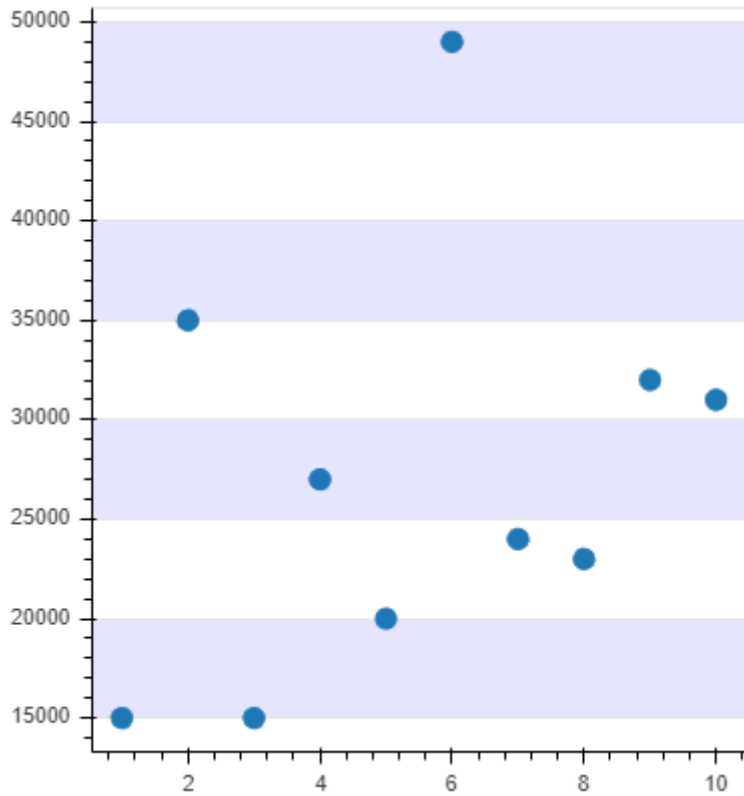
▼ 그리드(Grid)

```
p = figure(plot_width=400, plot_height=400)
p.circle(x, y, size=10)
p.xgrid.grid_line_color = None
p.ygrid.grid_line_alpha = 0.8
p.ygrid.grid_line_dash = [4, 4]
show(p)
```



```
p = figure(plot_width=400, plot_height=400)
p.circle(x, y, size=10)
```

```
p.scatter(x, y, size=10)
p.xaxis.grid(True)
p.ygrid.band_fill_alpha = 0.1
p.ygrid.band_fill_color = "blue"
show(p)
```



```
p = figure(plot_width=400, plot_height=400)
p.circle(x, y, size=10)
p.xaxis.grid(True)
p.ygrid.band_fill_alpha = 0.1
p.ygrid.band_fill_color = "green"
show(p)
```



▼ 데이터 제공(Providing Data)

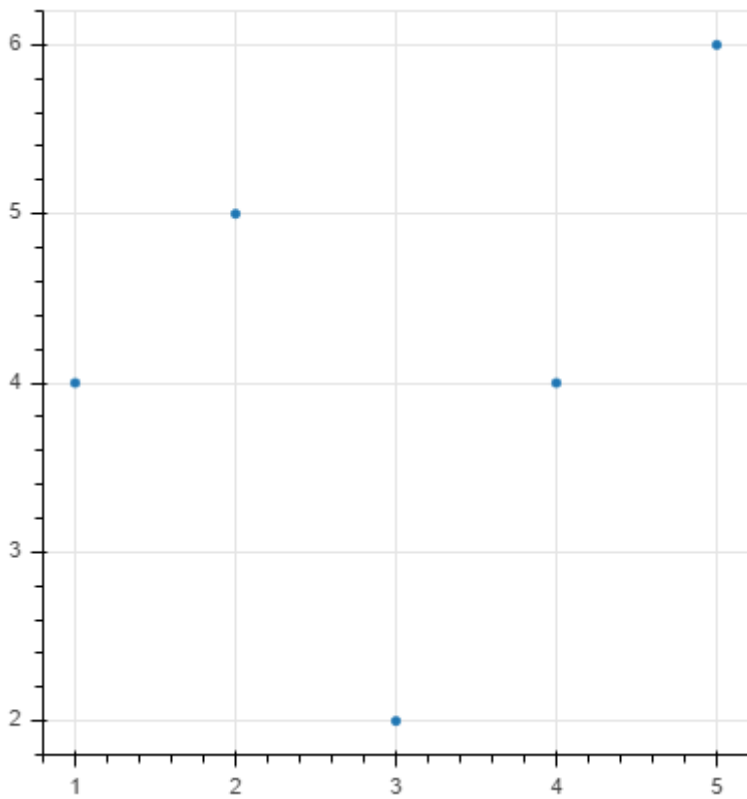


▼ 데이터 직접 제공



```
x = [1, 2, 3, 4, 5]
y = [4, 5, 2, 4, 6]
```

```
p = figure(plot_width=400, plot_height=400)
p.circle(x, y)
show(p)
```



▼ ColumnDataSource

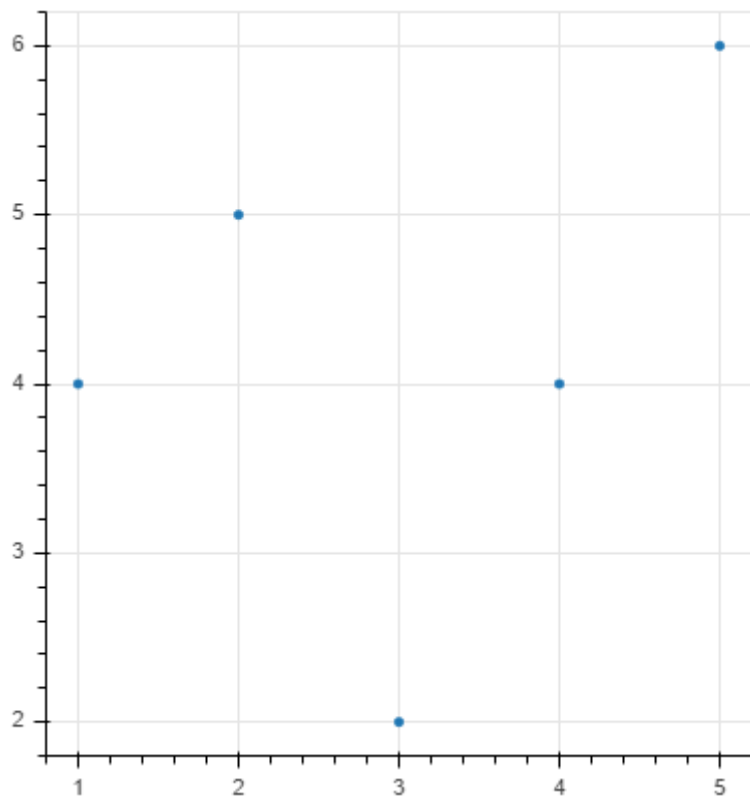
- 열 이름과 데이터 목록 사이의 매핑
- Bokeh 플롯의 핵심으로 플롯에서 글리프의 시각화된 데이터 제공
- DataTable과 같은 여러 플롯과 위젯간의 데이터를 쉽게 공유

```
from bokeh.models import ColumnDataSource

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

p = figure(plot_width=400, plot_height=400)
```

```
p.circle(x='x', y='y', source=source)
show(p)
```

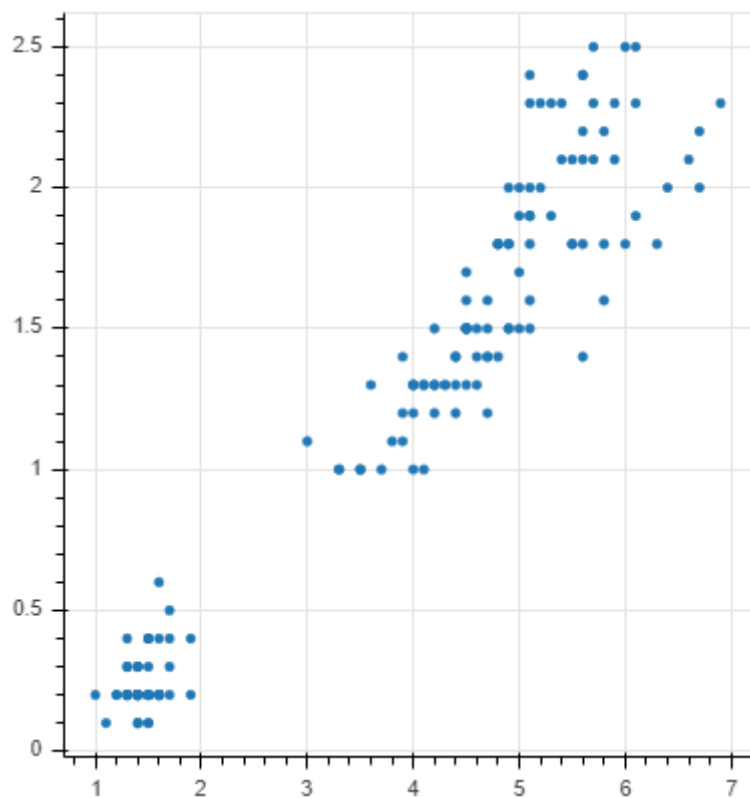


```
from bokeh.sampledata.iris import flowers
```

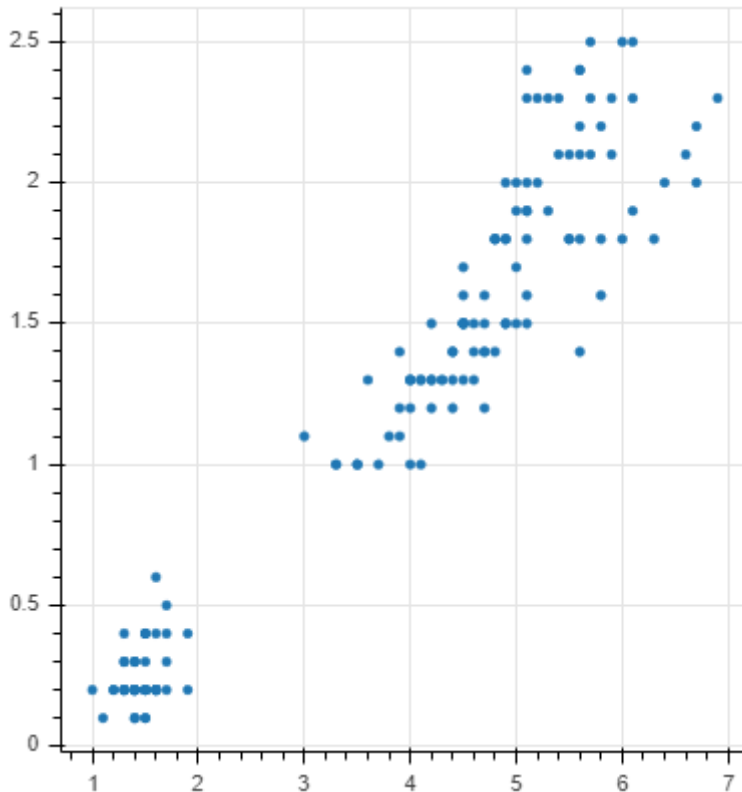
```
source = ColumnDataSource(flowers)
```

```
p = figure(plot_width=400, plot_height=400)
```

```
p.circle(x='petal_length', y='petal_width', source=source)
show(p)
```



```
p = figure(plot_width=400, plot_height=400)
p.circle(x='petal_length', y='petal_width', source=flowers)
show(p)
```

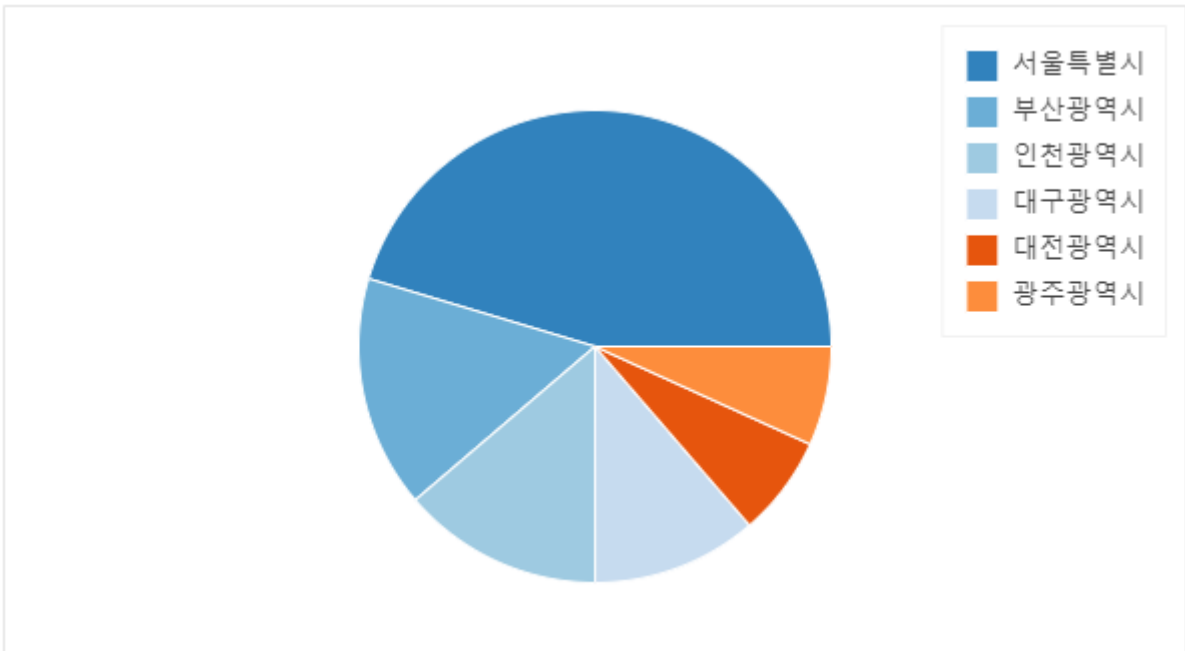


▼ 변환(Transformations)

```
from bokeh.palettes import Category20c
from bokeh.transform import cumsum

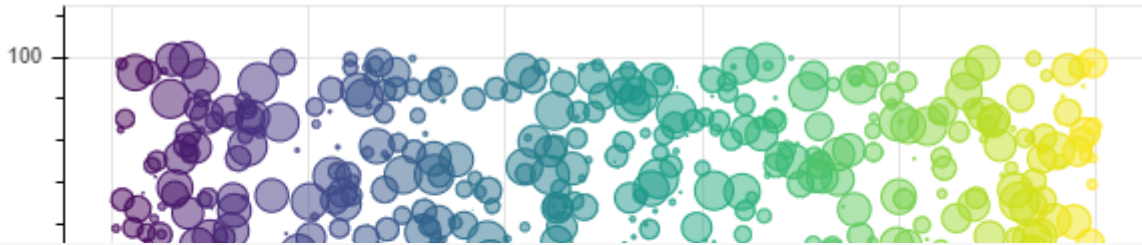
pop = {'서울특별시': 9720846,
       '부산광역시': 3404423,
       '인천광역시': 2947217,
       '대구광역시': 2427954,
       '대전광역시': 1471040,
       '광주광역시': 1455048}

data = pd.Series(pop).reset_index(name='population').rename(columns={'index': 'city'})
data['color'] = Category20c[len(pop)]
data['angle'] = data['population']/data['population'].sum() * 2 * np.pi
p = figure(plot_height=350, toolbar_location=None,
           tools="hover", tooltips="@city: @population")
p.wedge(x=0, y=1, radius=0.4,
        start_angle=cumsum('angle', include_zero=True), end_angle=cumsum('angle'),
        line_color="white", fill_color='color', legend_field='city', source=data)
p.axis.axis_label=None
p.axis.visible=False
p.grid.grid_line_color=None
show(p)
```



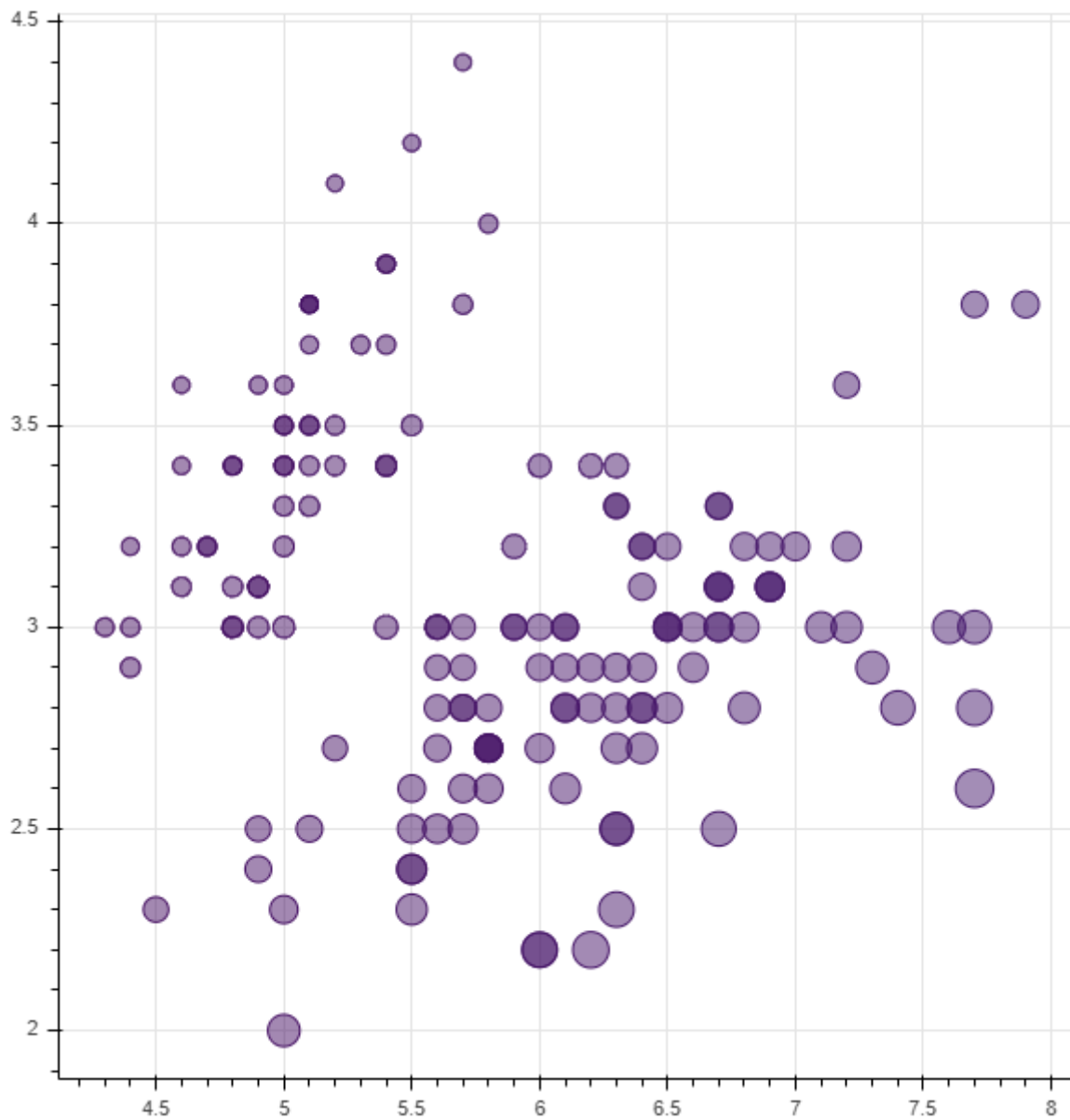
```
from bokeh.transform import linear_cmap

N = 2000
data = dict(x=np.random.random(size=N) * 100,
            y=np.random.random(size=N) * 100,
            r=np.random.random(size=N) * 2)
p = figure()
p.circle('x', 'y', radius='r', source=data, fill_alpha=0.5,
        color=linear_cmap('x', 'Viridis256', 0, 100))
show(p)
```



```
from bokeh.sampledata.iris import flowers
```

```
p = figure()
flowers['sepal_rate'] = (flowers['sepal_length'] / flowers['sepal_width']) / 40
p.circle('sepal_length', 'sepal_width', radius='sepal_rate',
        source=flowers, fill_alpha=0.5,
        color=linear_cmap('sepal_length', 'Viridis256', 0, 100))
show(p)
```



▼ 주식(Annotations)

▼ 스판(Span)

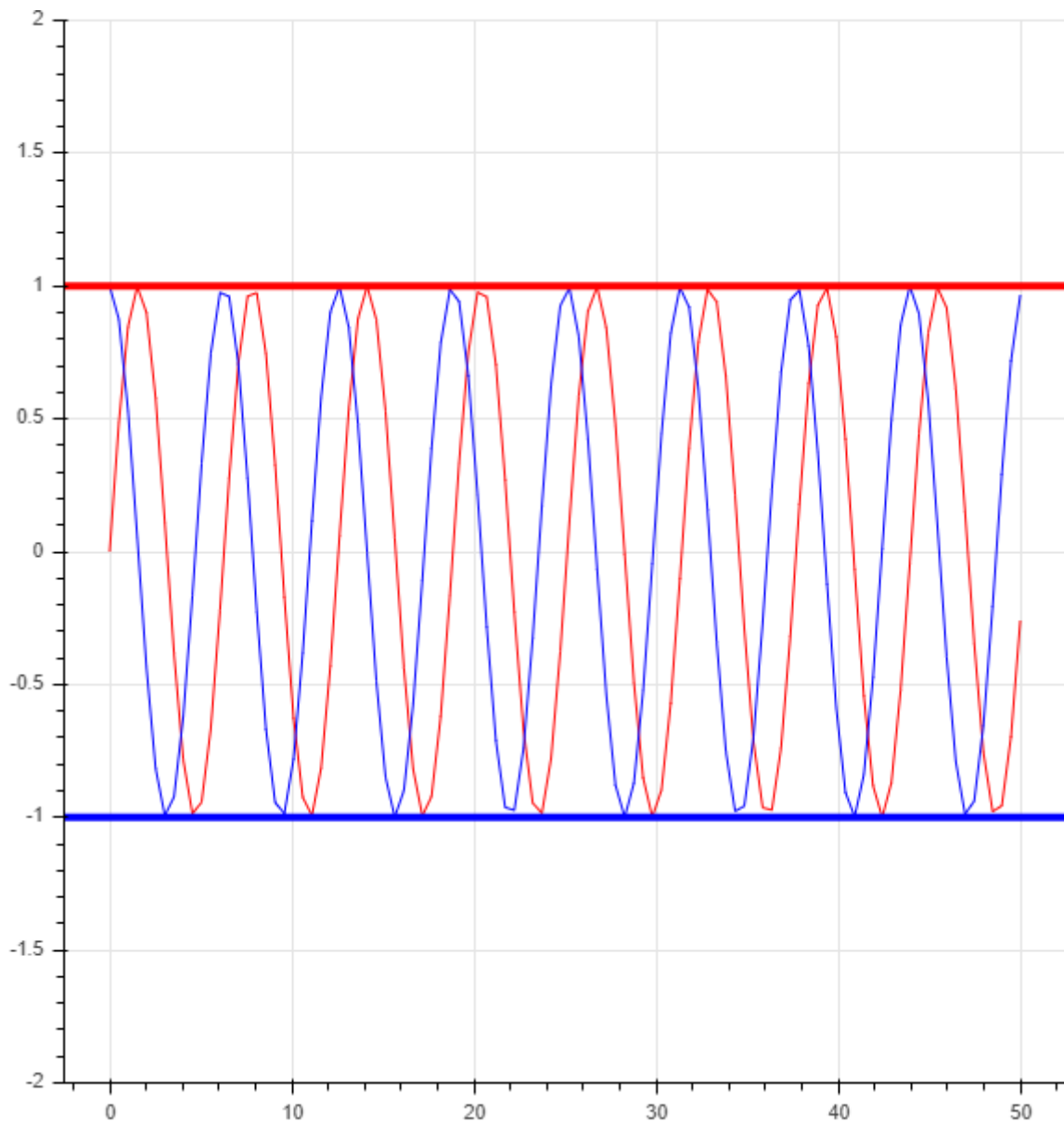
```
from bokeh.models.annotations import Span
```

```
x = np.linspace(0, 50, 100)  
y1 = np.sin(x)  
y2 = np.cos(x)
```

```
p = figure(y_range=(-2, 2))  
p.line(x, y1, color="red")  
p.line(x, y2, color="blue")
```

```
upper = Span(location=1, dimension='width', line_color='red', line_width=4)  
p.add_layout(upper)  
lower = Span(location=-1, dimension='width', line_color='blue', line_width=4)  
p.add_layout(lower)
```

```
show(p)
```



▼ 박스 주석(Box Annotations)


```

from bokeh.models.annotations import BoxAnnotation

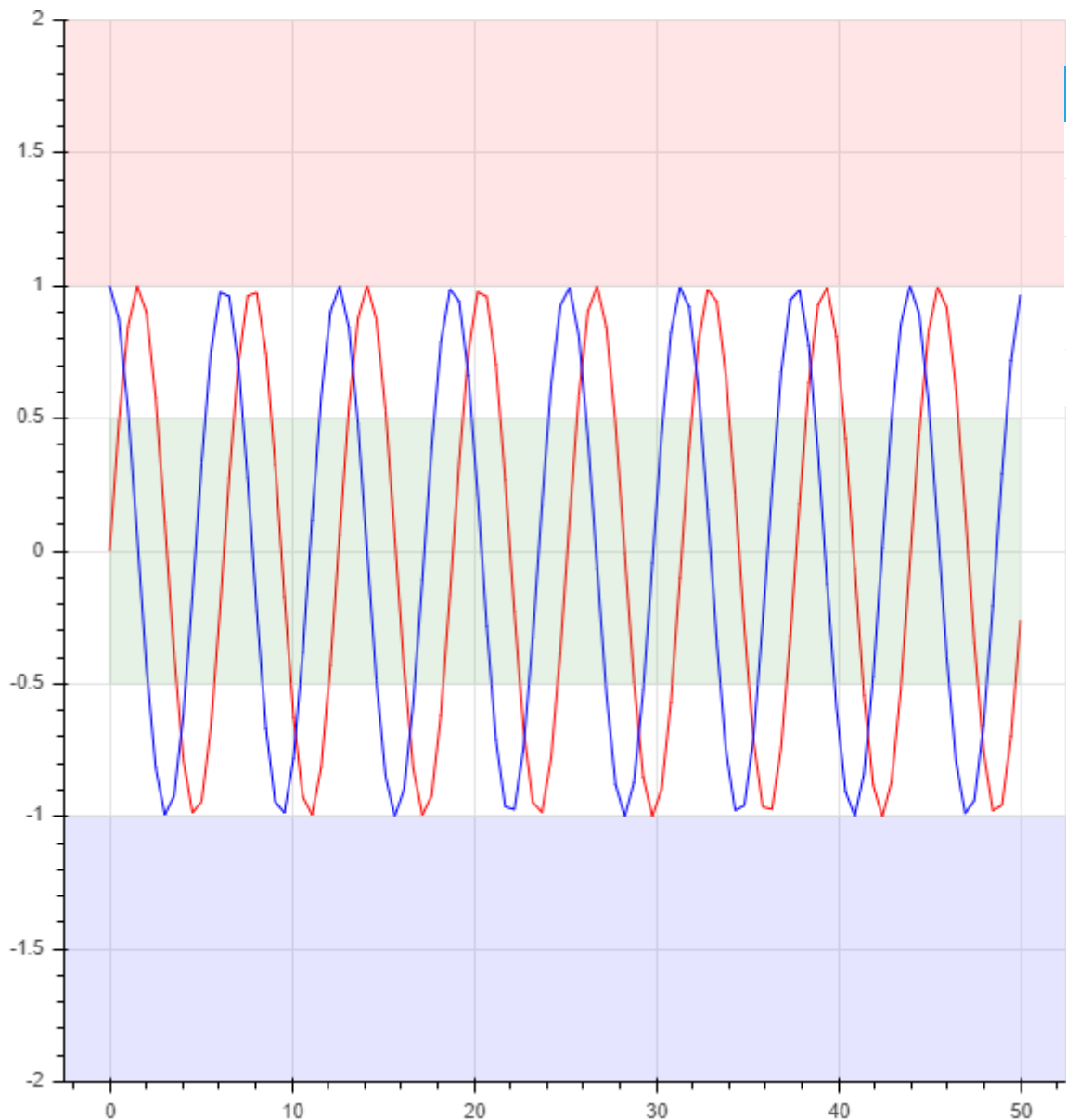
x = np.linspace(0, 50, 100)
y1 = np.sin(x)
y2 = np.cos(x)

p = figure(y_range=(-2, 2))
p.line(x, y1, color="red")
p.line(x, y2, color="blue")

upper = BoxAnnotation(bottom=1, fill_alpha=0.1, fill_color='red')
p.add_layout(upper)
lower = BoxAnnotation(top=-1, fill_alpha=0.1, fill_color='blue')
p.add_layout(lower)
center = BoxAnnotation(top=0.5, bottom=-0.5, left=0, right=50, fill_alpha=0.1, fill_color='green')
p.add_layout(center)

show(p)

```



▼ 라벨(Label)

```

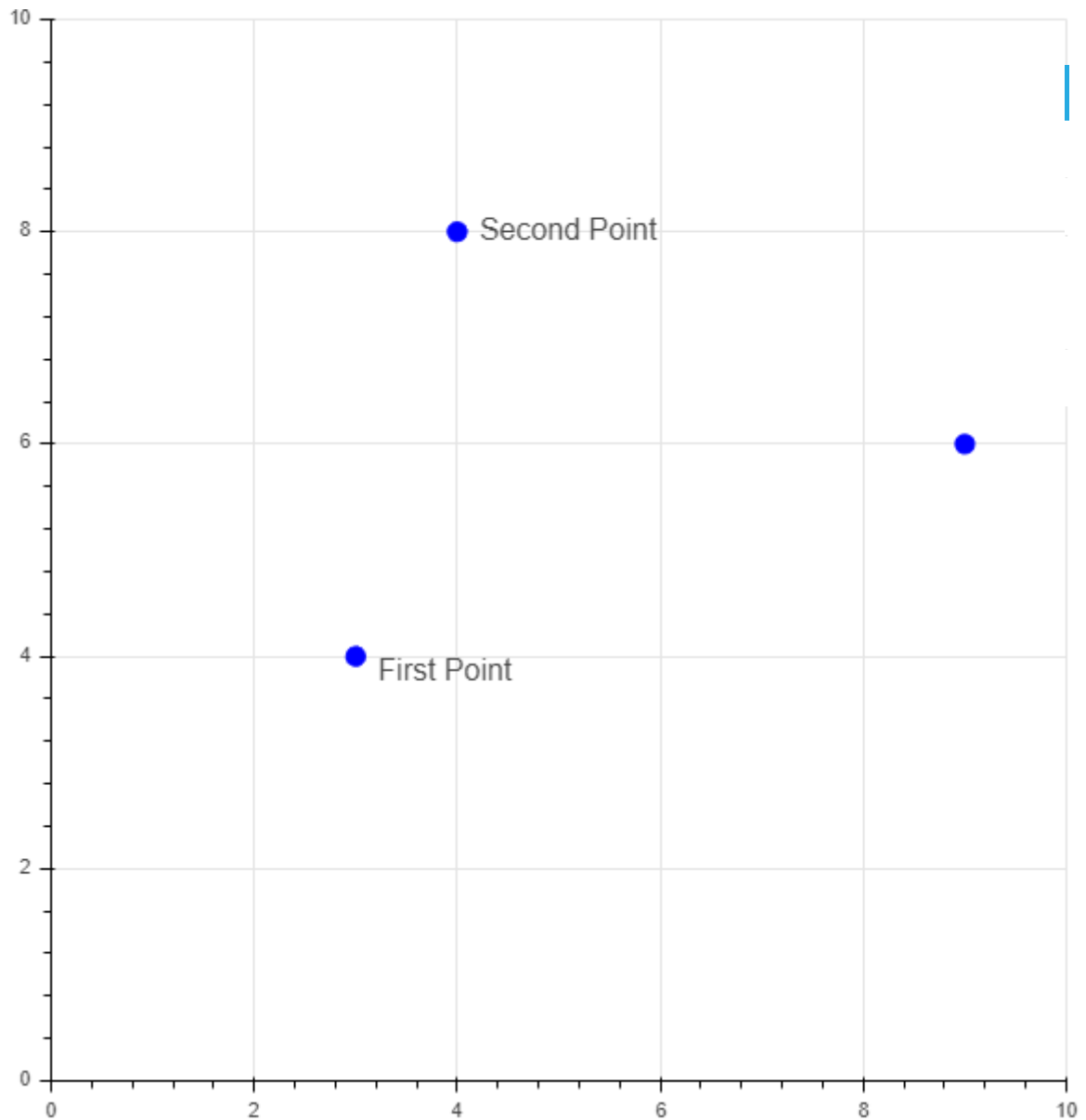
from bokeh.models.annotations import Label

p = figure(x_range=(0, 10), y_range=(0, 10))
p.circle([3, 4, 9], [4, 8, 6], color="blue", size=10)

label1 = Label(x=3, y=4, x_offset=12, text="First Point", text_baseline="top")
label2 = Label(x=4, y=8, x_offset=12, text="Second Point", text_baseline="middle")
p.add_layout(label1)
p.add_layout(label2)

show(p)

```



▼ 라벨셋(LabelSet)

```

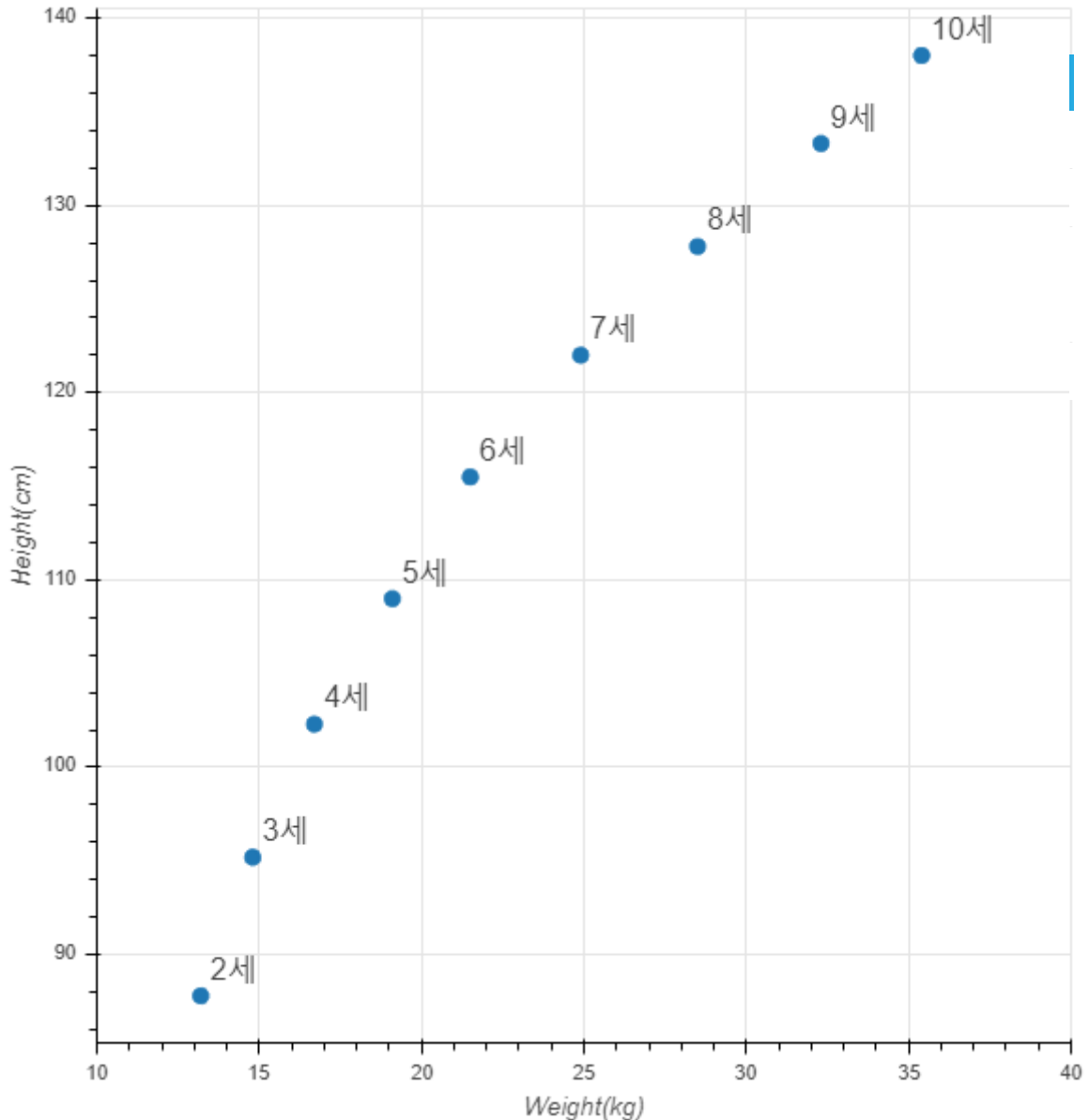
from bokeh.models import ColumnDataSource, LabelSet

source = ColumnDataSource(data=dict(
    weight=[13.2, 14.8, 16.7, 19.1, 21.5, 24.9, 28.5, 32.3, 35.4],
    height=[87.8, 95.2, 102.3, 109, 115.5, 122, 127.8, 133.3, 138]
))

```

```
height=[90, 95, 102, 109, 116, 122, 128, 133, 138, 143],
age=['2세', '3세', '4세', '5세', '6세', '7세', '8세', '9세', '10세'])
```

```
p = figure(x_range=(10, 40))
p.scatter(x='weight', y='height', size=8, source=source)
p.xaxis.axis_label = "Weight(kg)"
p.yaxis.axis_label = "Height(cm)"
labels = LabelSet(x='weight', y='height', text='age', level='glyph',
                  x_offset=5, y_offset=5, source=source, render_mode='canvas')
p.add_layout(labels)
show(p)
```



▼ 화살(Arrows)

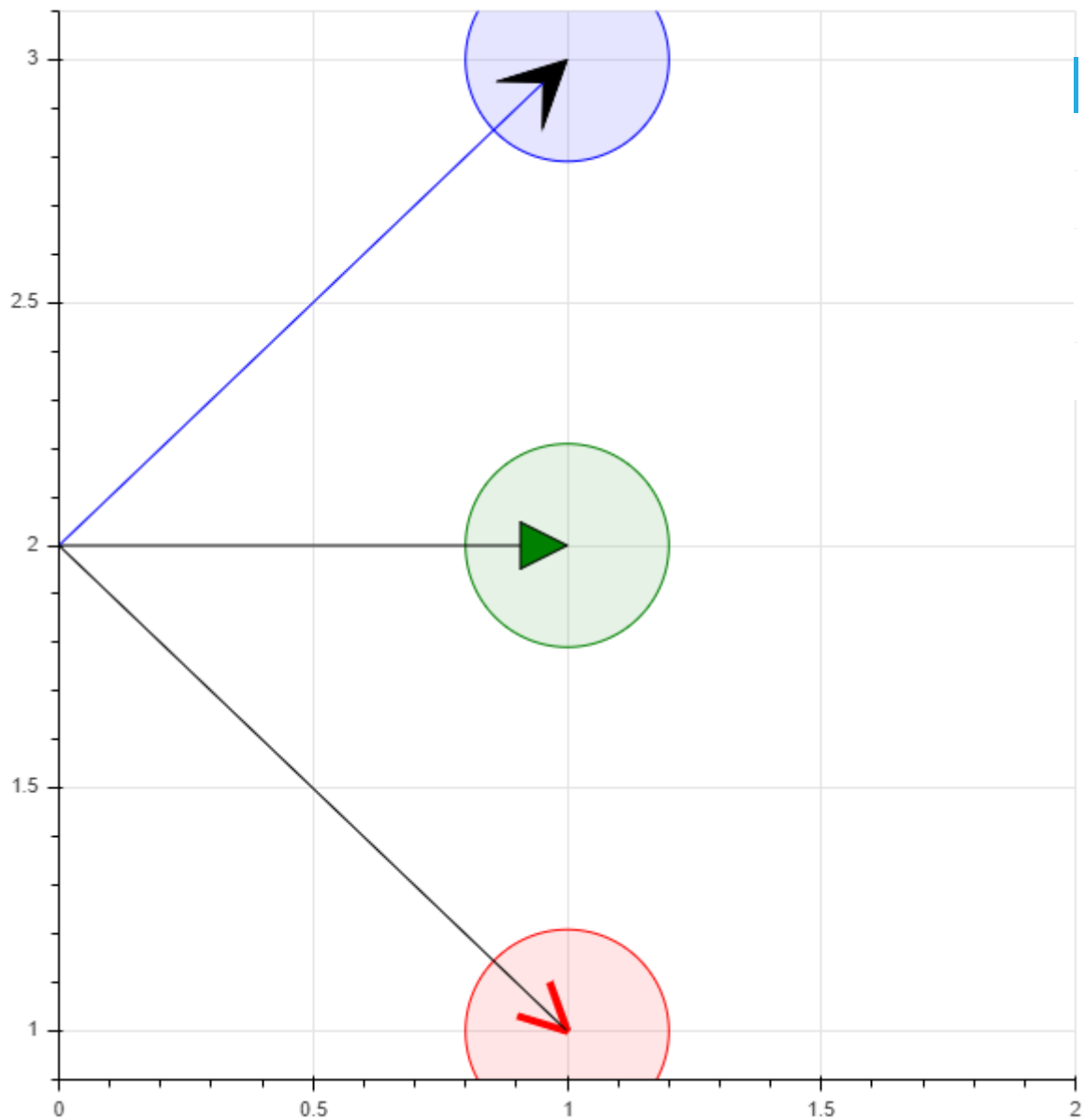
```
from bokeh.models.annotations import Arrow
from bokeh.models.arrow_heads import OpenHead, NormalHead, VeeHead

p = figure(plot_width=600, plot_height=600)
p.circle(x=[1, 1, 1], y=[1, 2, 3], radius=0.2,
         color=['red', 'green', 'blue'], fill_alpha=0.1)
p.add_layout(Arrow(end=OpenHead(line_color="red", line_width=4),
```

```

x_start=0, y_start=2, x_end=1, y_end=1))
p.add_layout(Arrow(end=NormalHead(fill_color="green"),
x_start=0, y_start=2, x_end=1, y_end=2))
p.add_layout(Arrow(end=VeeHead(size=35), line_color="blue",
x_start=0, y_start=2, x_end=1, y_end=3))
show(p)

```



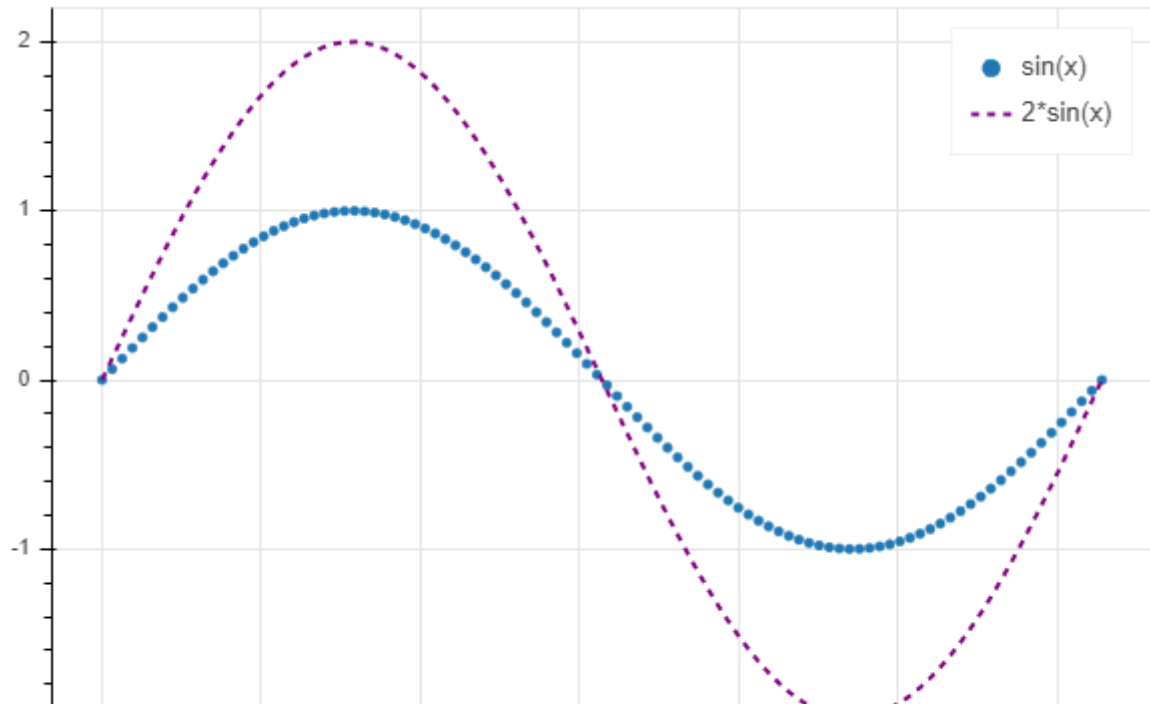
▼ 범례(Legends)

```

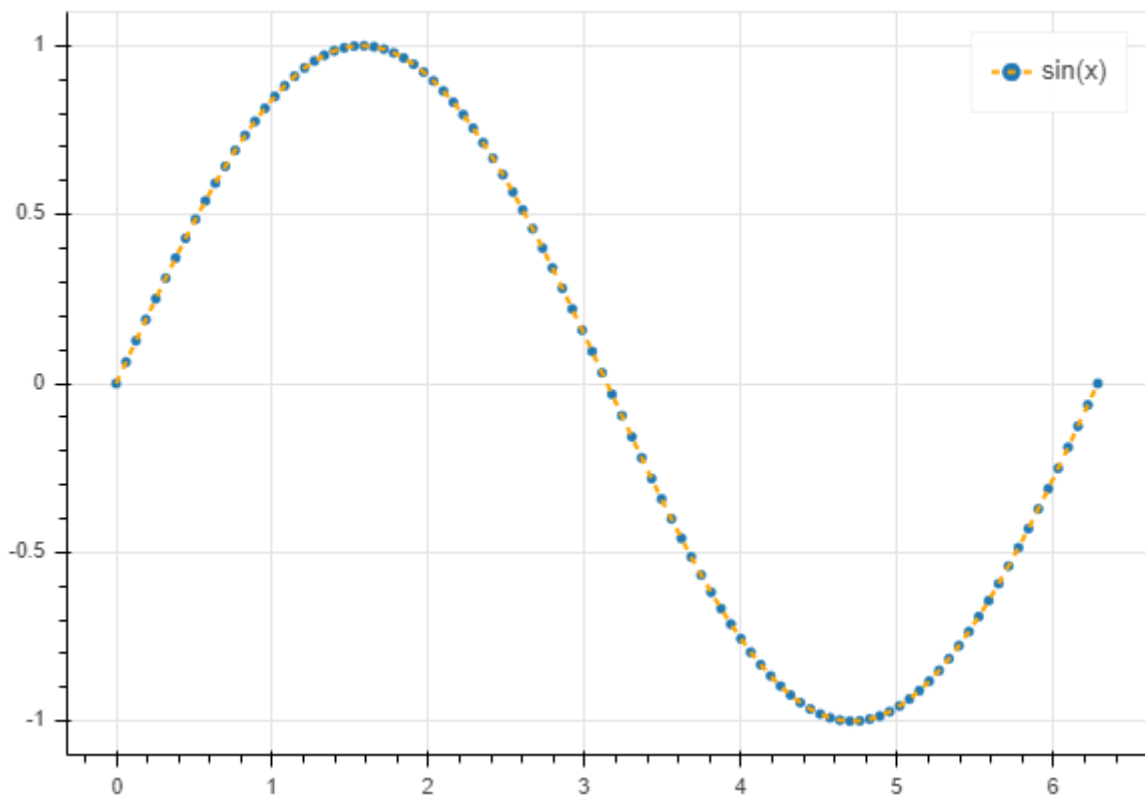
x = np.linspace(0, 2*np.pi, 100)
y = np.sin(x)

p = figure(height=400)
p.circle(x, y, legend_label="sin(x)")
p.line(x, 2*y, legend_label="2*sin(x)", line_dash=[4, 4], line_color="purple", line_width=2)
show(p)

```



```
p = figure(height=400)
p.circle(x, y, legend_label="sin(x)")
p.line(x, y, legend_label="sin(x)", line_dash=[4, 4], line_color="orange", line_width=2)
show(p)
```



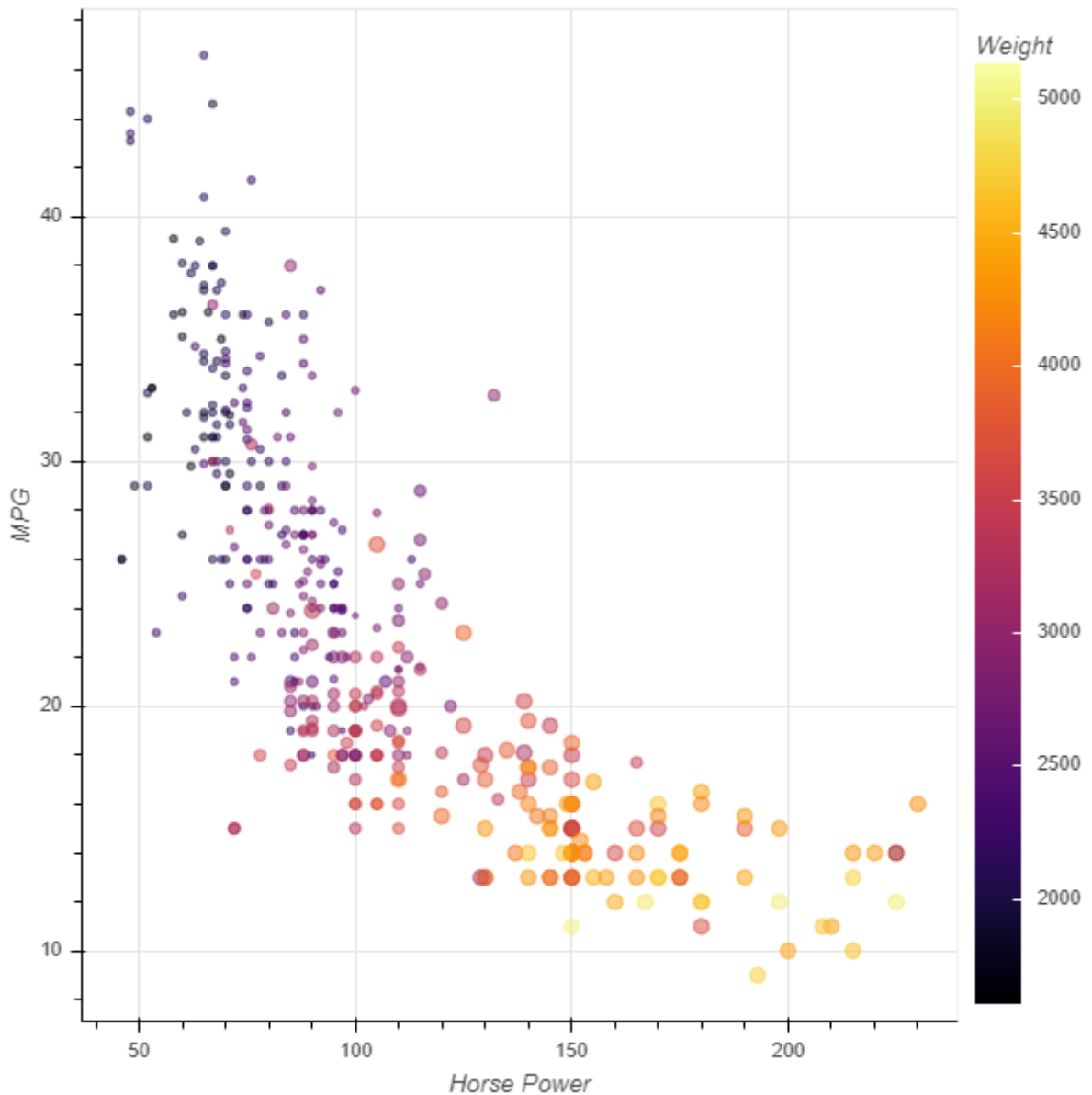
▼ 색상 막대(Color bars)

```
from bokeh.sampledata.autompg import autompg
from bokeh.models import LinearColorMapper, ColorBar
from bokeh.transform import transform
```

```

source = ColumnDataSource(autompg)
color_mapper = LinearColorMapper(palette="Inferno256",
                                low=autompg.weight.min(),
                                high=autompg.weight.max())
p = figure(x_axis_label='Horse Power', y_axis_label="MPG",
           tools='', toolbar_location=None)
p.circle(x='hp', y='mpg',
         color=transform('weight', color_mapper),
         size='cyl', alpha=0.5, source=autompg)
color_bar = ColorBar(color_mapper=color_mapper, label_standoff=10,
                    location=(0, 0), title='Weight')
p.add_layout(color_bar, 'right')
show(p)

```



▼ 레이아웃(Layout)

```

from bokeh.layouts import row

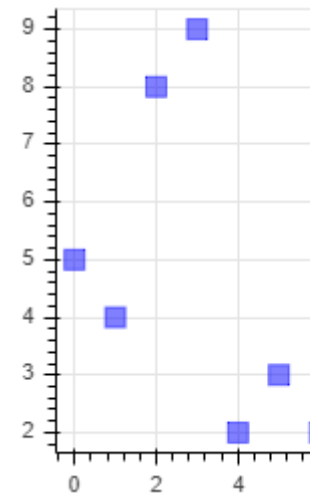
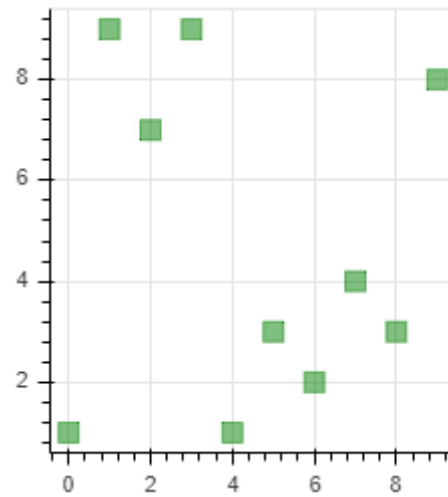
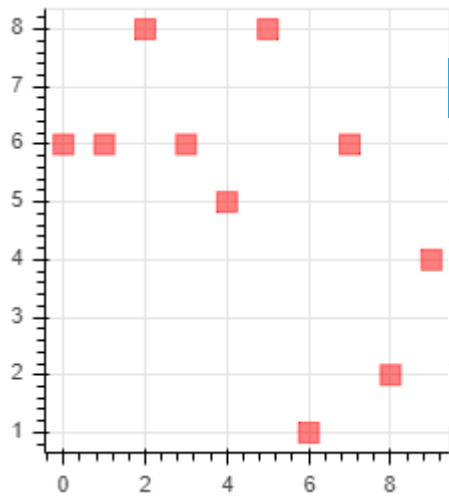
x = list(range(10))
y1 = np.random.randint(1, 10, 10)

```

```
y2 = np.random.randint(1, 10, 10)
y3 = np.random.randint(1, 10, 10)
```

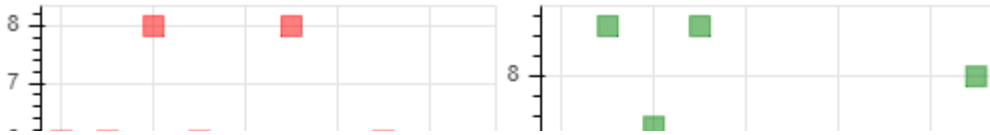
```
s1 = figure(width=250, plot_height=250)
s1.square(x, y1, size=10, color="red", alpha=0.5)
s2 = figure(width=250, plot_height=250)
s2.square(x, y2, size=10, color="green", alpha=0.5)
s3 = figure(width=250, plot_height=250)
s3.square(x, y3, size=10, color="blue", alpha=0.5)

show(row(s1, s2, s3))
```



```
from bokeh.layouts import gridplot
```

```
p = gridplot([[s1, s2], [s3, None]], toolbar_location=None)
show(p)
```



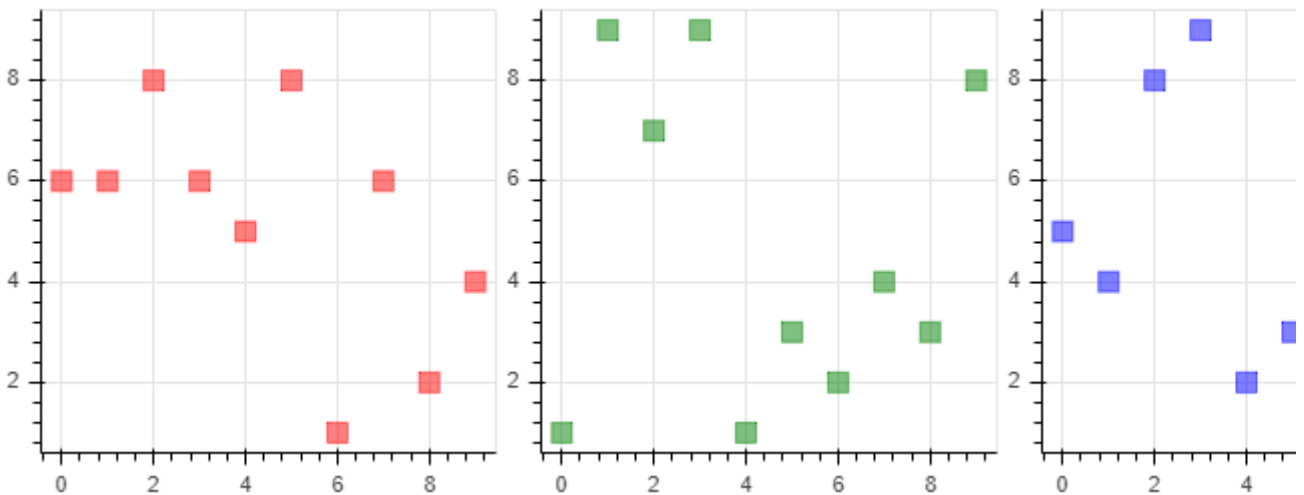
▼ 연결된 상호작용(Linked Interactions)



```
plot_options = dict(width=250, plot_height=250, tools='pan, wheel_zoom')
```

```
s1 = figure(**plot_options)
s1.square(x, y1, size=10, color="red", alpha=0.5)
s2 = figure(x_range=s1.x_range, **plot_options)
s2.square(x, y2, size=10, color="green", alpha=0.5)
s3 = figure(y_range=s1.y_range, **plot_options)
s3.square(x, y3, size=10, color="blue", alpha=0.5)
```

```
p = gridplot([[s1, s2, s3]])
show(p)
```

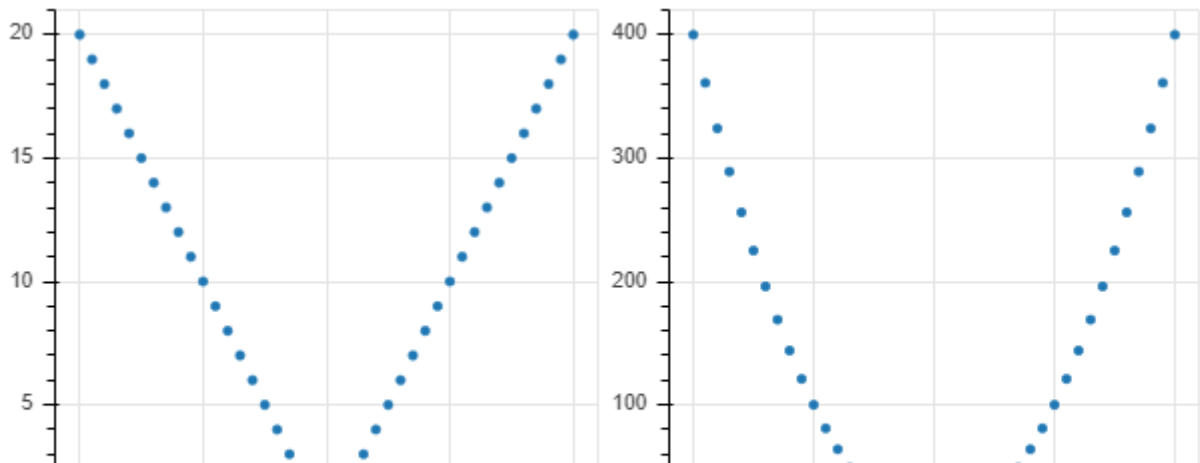


```
x = list(range(-20, 21))
y1, y2 = [abs(i) for i in x], [i ** 2 for i in x]

source = ColumnDataSource(data=dict(x=x, y1=y1, y2=y2))
tools = "box_select, lasso_select, help"

left = figure(tools=tools, width=300, height=300)
left.circle('x', 'y1', source=source)
right = figure(tools=tools, width=300, height=300)
right.circle('x', 'y2', source=source)

p = gridplot([[left, right]])
show(p)
```

```

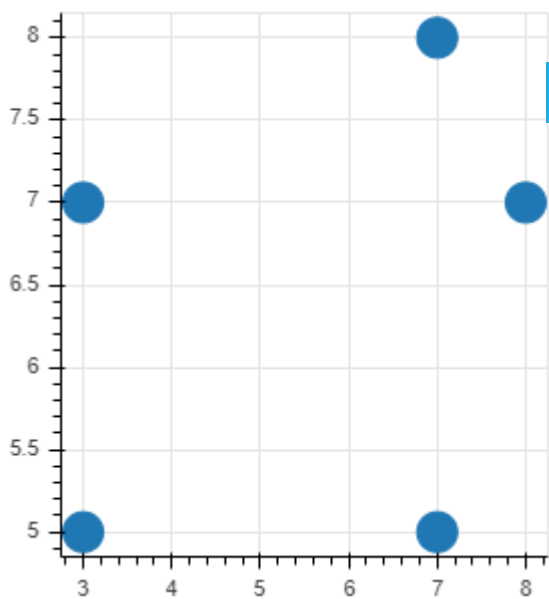
from bokeh.models import HoverTool

source = ColumnDataSource(
    data=dict(
        x=np.random.randint(1, 10, 5),
        y=np.random.randint(1, 10, 5),
        desc=['A', 'B', 'C', 'D', 'E'],
    )
)

hover = HoverTool(
    tooltips=[
        ("index", "$index"),
        ("(x, y)", "($x, $y)"),
        ("desc", "@desc"),
    ]
)

p = figure(plot_width=300, plot_height=300, tools=[hover])
p.circle('x', 'y', size=20, source=source)
show(p)

```



▼ 위젯(Widgets)

```
from bokeh.layouts import column
from bokeh.models import CustomJS, ColumnDataSource, Slider

x = [x*0.005 for x in range(0, 201)]
source = ColumnDataSource(data=dict(x=x, y=x))

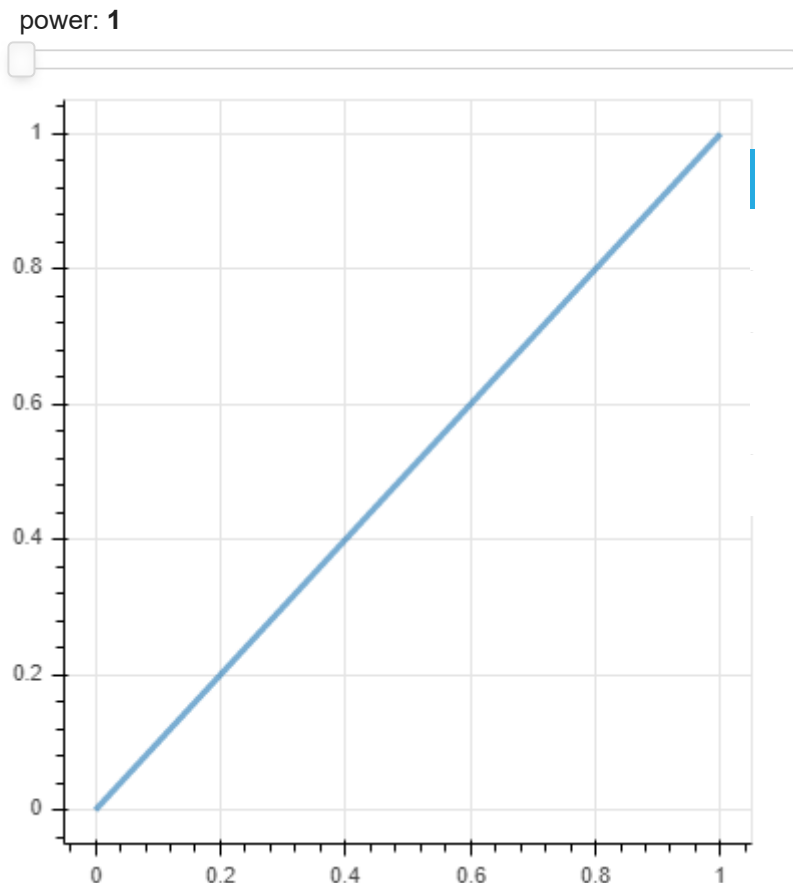
plot = figure(plot_width=400, plot_height=400)
plot.line('x', 'y', source=source, line_width=3, line_alpha=0.6)

slider = Slider(start=0.1, end=8, value=1, step=.1, title="power")

update_curve = CustomJS(args=dict(source=source, slider=slider), code="""
var data = source.data;
var f = slider.value;
var x = data['x'];
var y = data['y'];
for (var i = 0; i < x.length; i++) {
  y[i] = Math.pow(x[i], f)
}

source.change.emit();
""")
slider.js_on_change('value', update_curve)

show(column(slider, plot))
```



```

from bokeh.layouts import column, row
from bokeh.models import CustomJS, Slider
from bokeh.plotting import ColumnDataSource, figure, show

x = np.linspace(0, 10, 500)
y = np.sin(x)
source = ColumnDataSource(data=dict(x=x, y=y))

plot = figure(y_range=(-10, 10), plot_width=400, plot_height=400)
plot.line('x', 'y', source=source, line_width=3, line_alpha=0.6)

amp_slider = Slider(start=0.1, end=10, value=1, step=.1, title="Amplitude")
freq_slider = Slider(start=0.1, end=10, value=1, step=.1, title="Frequency")
phase_slider = Slider(start=0, end=6.4, value=0, step=.1, title="Phase")
offset_slider = Slider(start=-5, end=5, value=0, step=.1, title="Offset")

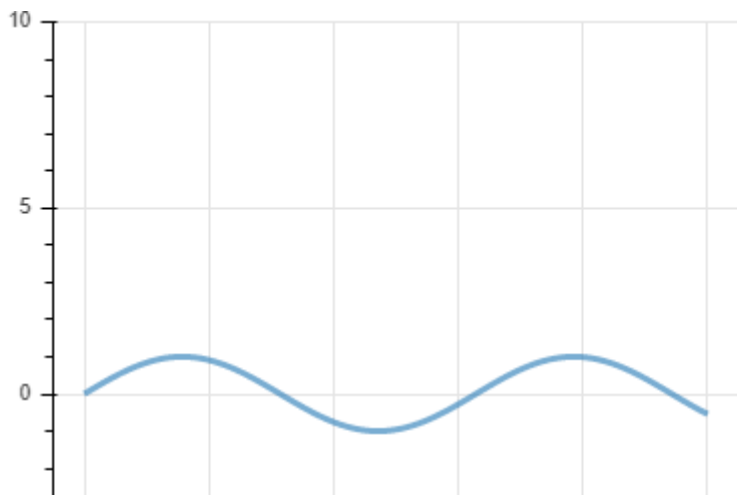
callback = CustomJS(args=dict(source=source,
                              amp=amp_slider, freq=freq_slider,
                              phase=phase_slider, offset=offset_slider),
                    code="""
const data = source.data;
const A = amp.value;
const k = freq.value;
const phi = phase.value;
const B = offset.value;
const x = data['x']
const y = data['y']
for (var i = 0; i < x.length; i++) {
  y[i] = B + A * Math.sin(k * x[i] + phi);
}
source.change.emit();
""")

amp_slider.js_on_change('value', callback)
freq_slider.js_on_change('value', callback)
phase_slider.js_on_change('value', callback)
offset_slider.js_on_change('value', callback)

layout = row(
    plot,
    column(amp_slider, freq_slider, phase_slider, offset_slider),
)

show(layout)

```



Amplitude: 1



Frequency: 1



Phase: 0



Offset: 0



```

from random import random

x = [random() for x in range(200)]
y = [random() for y in range(200)]
color = ["blue"] * len(x)

s1 = ColumnDataSource(data=dict(x=x, y=y, color=color))
p = figure(plot_width=400, plot_height=400, tools="lasso_select")
p.circle('x', 'y', color='color', size=8, alpha=0.4, source=s1,
         selection_color="orange", selection_alpha=0.4)

s2 = ColumnDataSource(data=dict(xm=[0, 1], ym=[0.5, 0.5]))
p.line(x='xm', y='ym', color='red', line_width=5, alpha=0.6, source=s2)

callback = CustomJS(args=dict(s1=s1, s2=s2), code="""
var inds = s1.selected.indices;
if (inds.length == 0)
  return;

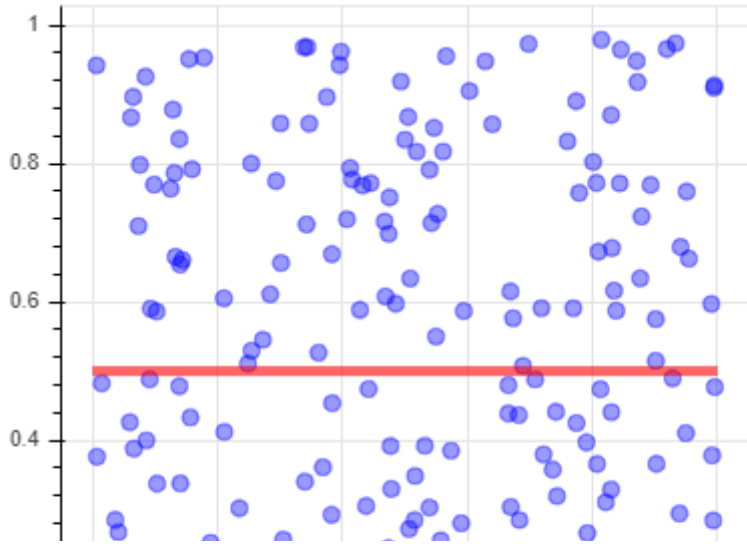
var ym = 0
for (var i = 0; i < inds.length; i++) {
  ym += s1.data.y[inds[i]]
}

ym /= inds.length
s2.data.ym = [ym, ym]
s2.change.emit();
""")

s1.selected.js_on_change('indices', callback)

show(p)

```



▼ 막대와 범주형 데이터 플롯(Bar and Categorical Data Plots)

```

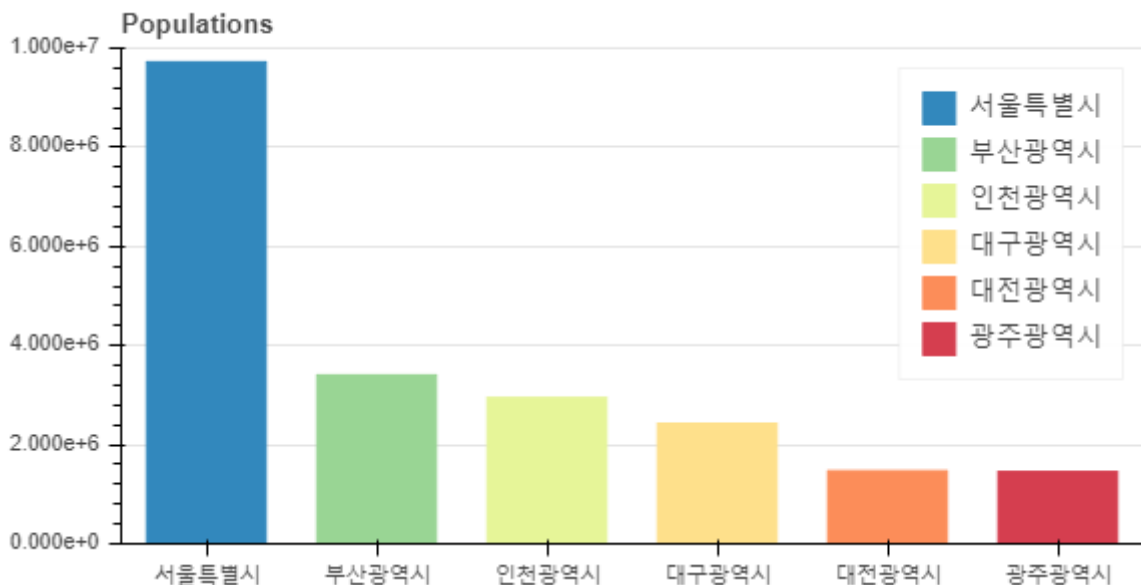
from bokeh.models import ColumnDataSource
from bokeh.palettes import Spectral6

cities = ['서울특별시', '부산광역시', '인천광역시',
         '대구광역시', '대전광역시', '광주광역시']
pops = [9720846, 3404423, 2947217, 2427954, 1471040, 1455048]
source = ColumnDataSource(data=dict(cities=cities, pops=pops, color=Spectral6))

p = figure(x_range=cities, plot_height=300, y_range=(0, 10000000), title="Populations")
p.vbar(x='cities', top='pops', width=0.7, color='color', legend_field='cities', source=source)
p.xgrid.grid_line_color=None
p.legend.orientation="vertical"
p.legend.location="top_right"

show(p)

```



```

from bokeh.palettes import GnBu3, PuBu3

```

```

years = ['2018', '2019', '2020']

```

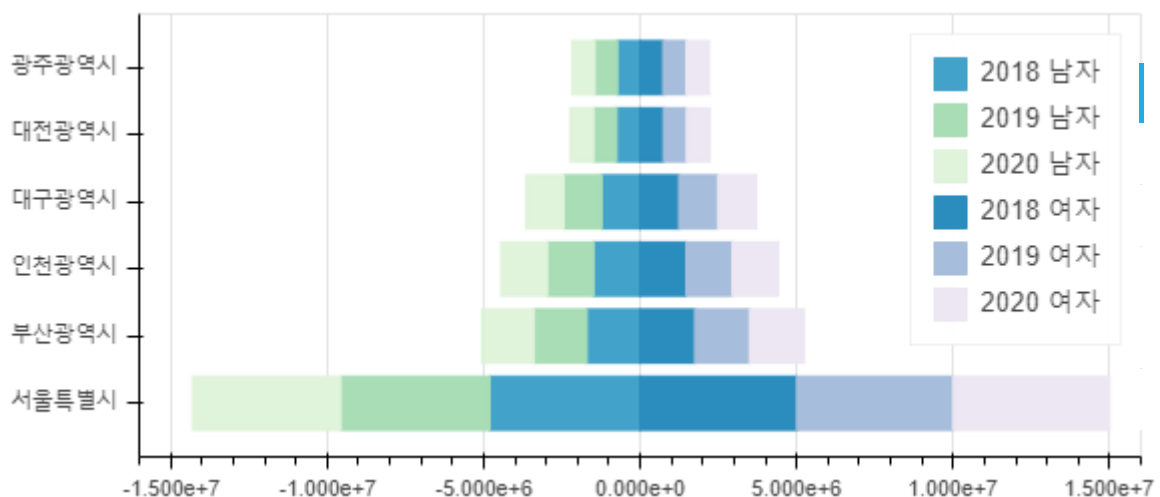
```

years = [ 2018 , 2019 , 2020 ]
cities = [ '서울특별시', '부산광역시', '인천광역시',
          '대구광역시', '대전광역시', '광주광역시' ]
male = { 'cities' : cities,
        '2018' : [-4802769, -1700822, -1481769, -1224126, -747071, -723562],
        '2019' : [-4762711, -1683596, -1482934, -1212450, -740834, -722559],
        '2020' : [-4732274, -1668618, -1476813, -1198815, -734441, -720060] }
female = { 'cities' : cities,
          '2018' : [5011280, 1754789, 1472114, 1245491, 747807, 737183],
          '2019' : [4994433, 1744036, 1474090, 1238028, 742104, 736465],
          '2020' : [4988571, 1735805, 1470404, 1229139, 736599, 734988] }

p = figure(y_range=cities, plot_height=250, x_range=(-16000000, 16000000))
p.hbar_stack(years, y='cities', height=0.8, color=GnBu3, source=ColumnDataSource(male),
             legend_label=["%s 남자" % x for x in years])
p.hbar_stack(years, y='cities', height=0.8, color=PuBu3, source=ColumnDataSource(female),
             legend_label=["%s 여자" % x for x in years])
p.y_range.range_padding = 0.1
p.ygrid.grid_line_color = None
p.legend.location = "top_right"

show(p)

```



```

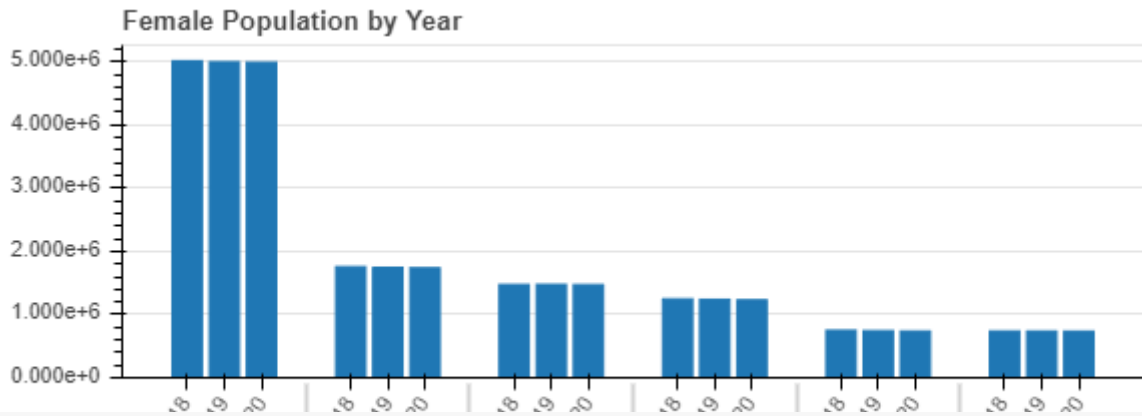
from bokeh.models import FactorRange

x = [ (city, year) for city in cities for year in years ]
pops = sum(zip(female['2018'], female['2019'], female['2020']), ())
source = ColumnDataSource(data=dict(x=x, pops=pops))

p = figure(x_range=FactorRange(*x), plot_height=250, title="Female Population by Year")
p.vbar(x='x', top='pops', width=0.8, source=source)
p.y_range.start = 0
p.x_range.range_padding = 0.1
p.xaxis.major_label_orientation = 1
p.xgrid.grid_line_color = None

show(p)

```



```

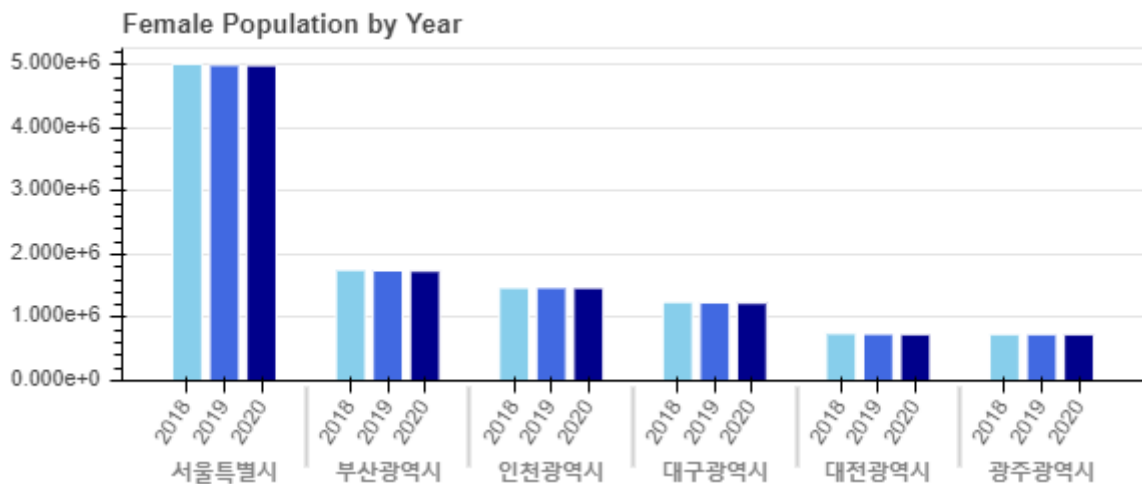
from bokeh.transform import factor_cmap

p = figure(x_range=FactorRange(*x), plot_height=250, title="Female Population by Year")
p.vbar(x='x', top='pops', width=0.8, source=source, line_color="white",
       fill_color=factor_cmap('x', palette=['skyblue', 'royalblue', 'darkblue'],
                             factors=years, start=1, end=2))

p.y_range.start = 0
p.x_range.range_padding = 0.1
p.xaxis.major_label_orientation = 1
p.xgrid.grid_line_color = None

show(p)

```



```

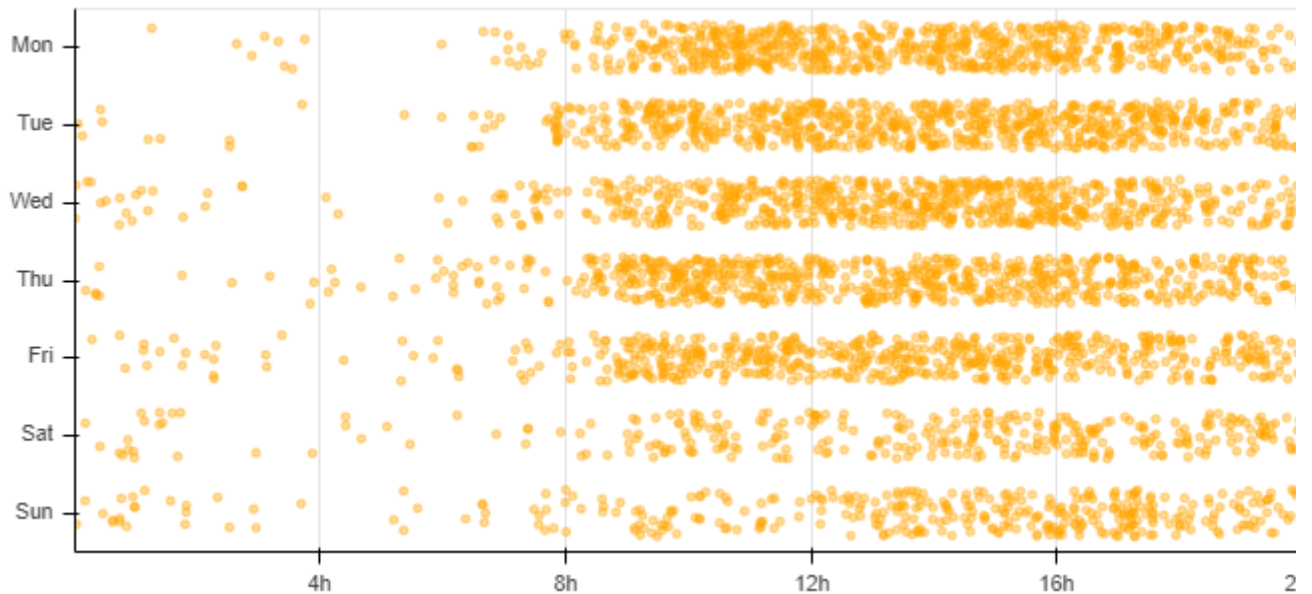
from bokeh.sampledata.commits import data
from bokeh.transform import jitter

days = ['Sun', 'Sat', 'Fri', 'Thu', 'Wed', 'Tue', 'Mon']
source = ColumnDataSource(data)

p = figure(plot_width=800, plot_height=300,
          y_range=days, x_axis_type='datetime')
p.circle(x='time', y=jitter('day', width=0.6, range=p.y_range),
        color='orange', source=source, alpha=0.4)
p.xaxis[0].formatter.days = ['%Hh']
p.x_range.range_padding = 0
p.ygrid.grid_line_color = None

show(p)

```



▼ 내보내기(Exporting)

```

from bokeh.sampledata.stocks import AAPL, IBM, MSFT, GOOG
from bokeh.palettes import Spectral4

p = figure(plot_width=800, plot_height=250, x_axis_type="datetime")

for data, name, color in zip([AAPL, IBM, MSFT, GOOG], ["AAPL", "IBM", "MSFT", "GOOG"], Spectral4):
    datetime = np.asarray(data['date'], dtype=np.datetime64)
    value = np.asarray(data['close'])
    p.line(datetime, value, line_width=1, color=color, alpha=0.8, legend_label=name)

p.legend.location='top_left'
p.legend.click_policy='hide'

show(p)

```



```

from bokeh.io import output_file, show

output_file("stocks.html")

```



```
show(p)
```



```
!!s
```

```
sample_data stocks.html
```

```
!cat stocks.html
```

```
<!DOCTYPE html>  
<html lang="en">
```

```
<head>
```

```
<meta charset="utf-8">  
<title>Bokeh Plot</title>
```

```
<script type="text/javascript" src="https://cdn.pydata.org/bokeh/release/bokeh-1.4  
<script type="text/javascript">  
  Bokeh.set_log_level("info");  
</script>
```

```
</head>
```

```
<body>
```

```
<div class="bk-root" id="f483b6d3-a8b2-4dcf-abc5-d7f484531a93" data-root-id=
```

```
<script type="application/json" id="123836">
  {"0ebe2291-076b-48d1-aa6f-ceb163d48afd":{"roots":{"references":[{"attributes":{"
</script>
<script type="text/javascript">
  (function() {
    var fn = function() {
      Bokeh.safely(function() {
        (function(root) {
          function embed_document(root) {

            var docs_json = document.getElementById('123836').textContent;
            var render_items = [{"docid":"0ebe2291-076b-48d1-aa6f-ceb163d48afd", "roc
            root.Bokeh.embed.embed_items(docs_json, render_items);

          }
          if (root.Bokeh !== undefined) {
```

참고 문헌

- Bokeh, <https://bokeh.org/>