jupyter Bokeh Last Checkpoint: a few seconds ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

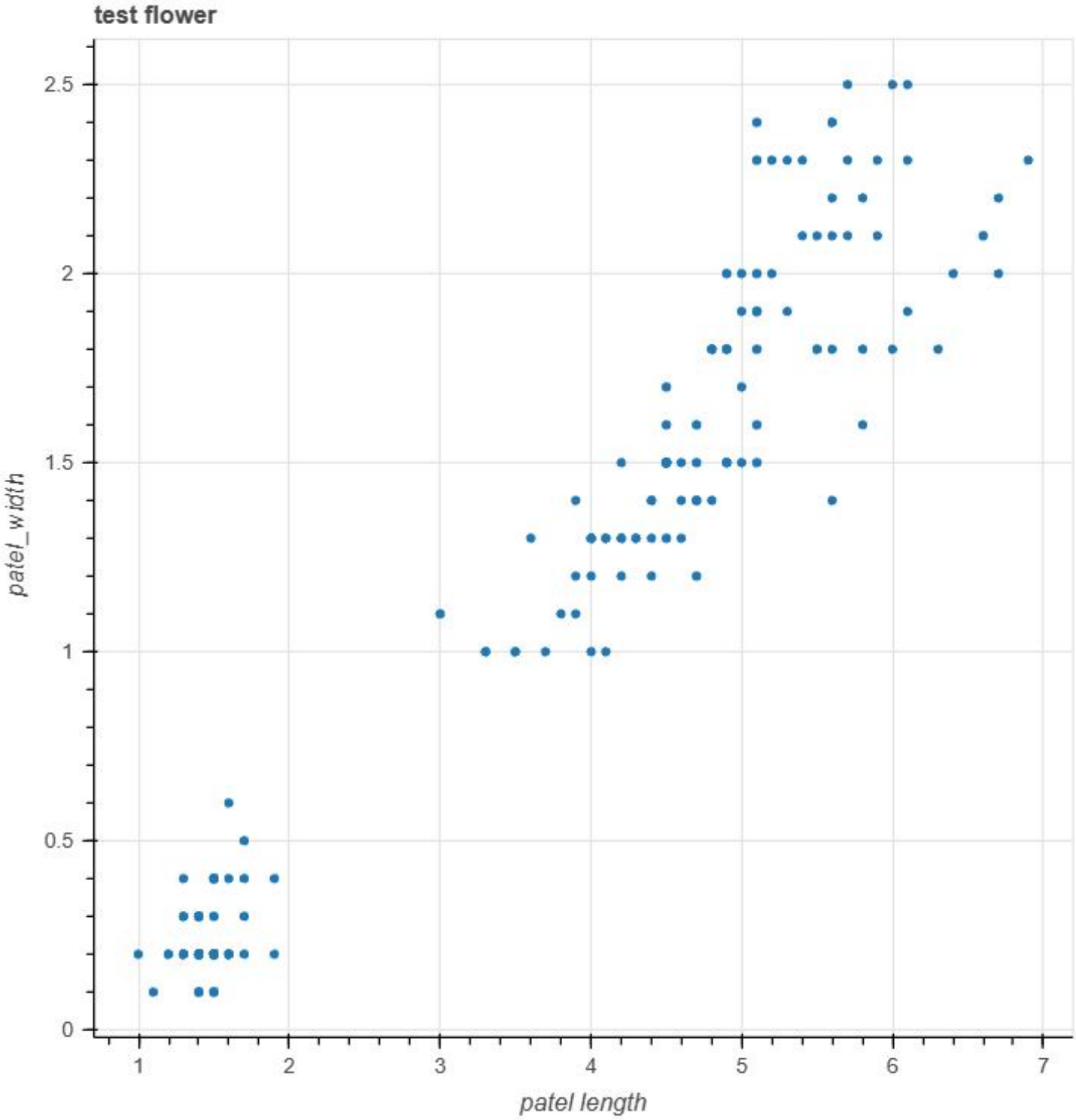
Python Bokeh is a Data Visualization library that provides interactive charts and plots. Bokeh renders its plots using HTML and JavaScript that uses modern web browsers for presenting elegant, concise construction of novel graphics with high-level interactivity.

In [3]:
```
# installation
#pip install bokeh
```

In [17]:
```
import bokeh.io
import bokeh.plotting
bokeh.io.output_notebook()
```

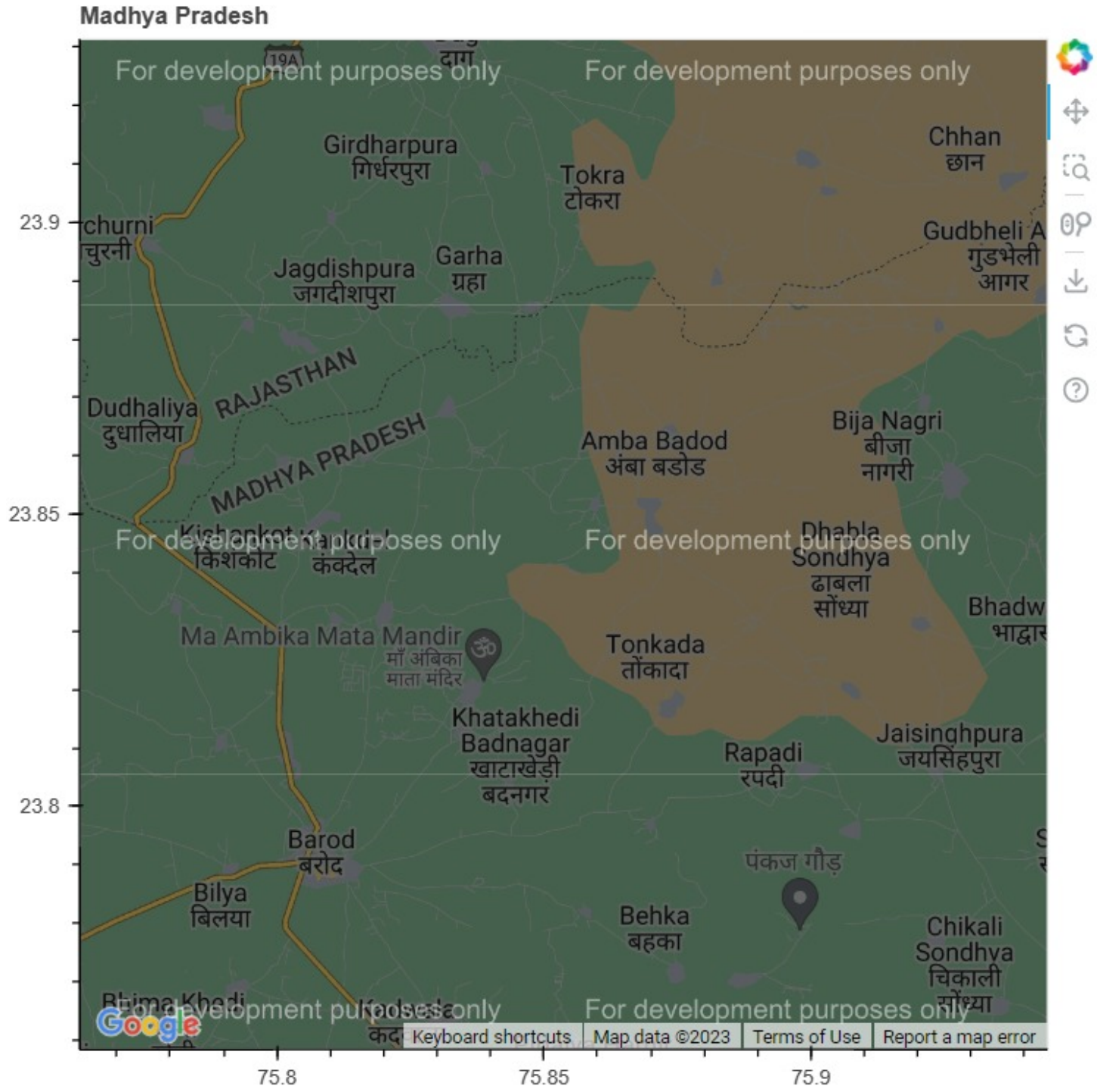
BokehJS 3.1.1 successfully loaded.

In [6]:
```
from bokeh.plotting import figure , output_file ,show
from bokeh.sampledata.iris import flowers

# give output html file
output_file('output_file.html ')
p = figure(title =  'test flower')
p.xaxis.axis_label = 'patel length'
p.yaxis.axis_label = ' patel_width'
p.circle(flowers['petal_length'],flowers['petal_width'])
show(p)
```


test flower
2.5
2
1.5
1
0.5
0
patel_width
1
2
3
4
5
6
7
patel length

Python Bokeh – Plot for all Types of Google Maps (roadmap, satellite, hybrid, terrain)

In [24]:
```
# importing the required modules
from bokeh.plotting import gmap
from bokeh.models import GMapOptions
from bokeh.io import output_file, show

# file to save the model
output_file("google_mapfile.html")

# configuring the Google map
lat = 23.9475213
lng = 75.7795103
map_type = "roadmap"
zoom = 12
google_map_options = GMapOptions(lat = lat,lng = lng,map_type = map_type,zoom = zoom)

# generating the Google map
google_api_key = ""
title = "Madhya Pradesh"
google_map = gmap(google_api_key,google_map_options,title = title)

# displaying the model
show(google_map)
```


Madhya Pradesh
For development purposes only
For development purposes only
Girdharpura
गिर्धरपुरा
Tokra
टोकरा
Chhan
छान
churni
चुरनी
Jagdishpura
जगदीशपुरा
Garha
ग्रहा
Gudbheli A
गुडभेली
आगर
23.9
RAJASTHAN
MADHYA PRADESH
Dudhaliya
दुधालिया
Amba Badod
अंबा बडोड
Bija Nagri
बीजा
नागरी
23.85
For development purposes only
For development purposes only
Dhabla
Sondhya
ढाबला
सोंध्या
Ma Ambika Mata Mandir
माँ अंबिका
माता मंदिर
Tonkada
तोंकादा
Bhadw
Khatakhedi
Badnagar
खाटाखेड़ी
बदनगर
Rapadi
रपदी
Jaisinghpura
जयसिंहपुरा
23.8
Barod
बरोद
पंकज गौड़
Bilya
बिलया
Behka
बहका
Chikali
Sondhya
चिकाली
सोंध्या
For development purposes only
For development purposes only
Google
Keyboard shortcuts
Map data ©2023
Terms of Use
Report a map error
75.8
75.85
75.9

In [25]:

```
# satellitee view in google map
# importing the required modules
from bokeh.plotting import gmap
from bokeh.models import GMapOptions
from bokeh.io import output_file, show

# file to save the model
output_file("satellite_file.html")

# configuring the Google map
lat = 23.9475213
lng = 75.7795103
map_type = "satellite"
zoom = 12
google_map_options = GMapOptions(lat = lat,lng = lng,map_type = map_type,zoom = zoom)

# generating the Google map
google_api_key = ""
title = "Madhya Pradesh"
google_map = gmap(google_api_key,google_map_options,title = title)

# displaying the model
show(google_map)
```


Madhya Pradesh
23.95
23.9
23.85
urposes only
For development purposes only
For development pur
urposes only
For development purposes only
For development pur
Google
Keyboard shortcuts
Imagery ©2023 TerraMetrics
Terms of Use
Report a map error
75.75
75.8
75.85
75.9

In [7]:
```
flowers
```

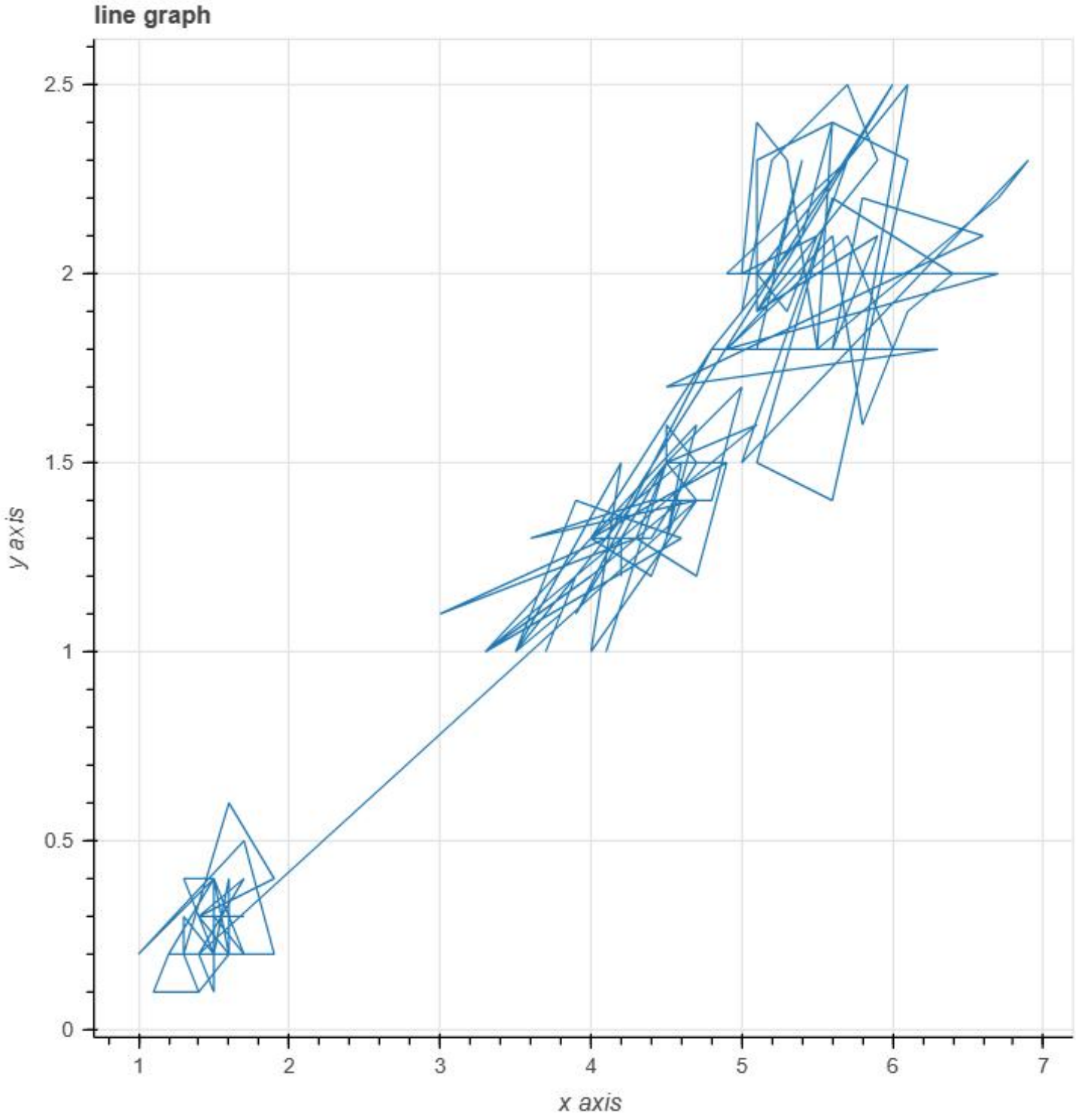
Out[7]:

	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
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

Line Graph

In [9]:
```
from bokeh.plotting import figure , output_file , show
from bokeh.sampledata.iris import flowers

output_file("line.html")

p = figure(title = 'line graph')
p.xaxis.axis_label = '  x axis'
p.yaxis.axis_label = 'y axis'
p.line(flowers['petal_length'] , flowers['petal_width'])
show(p)
```

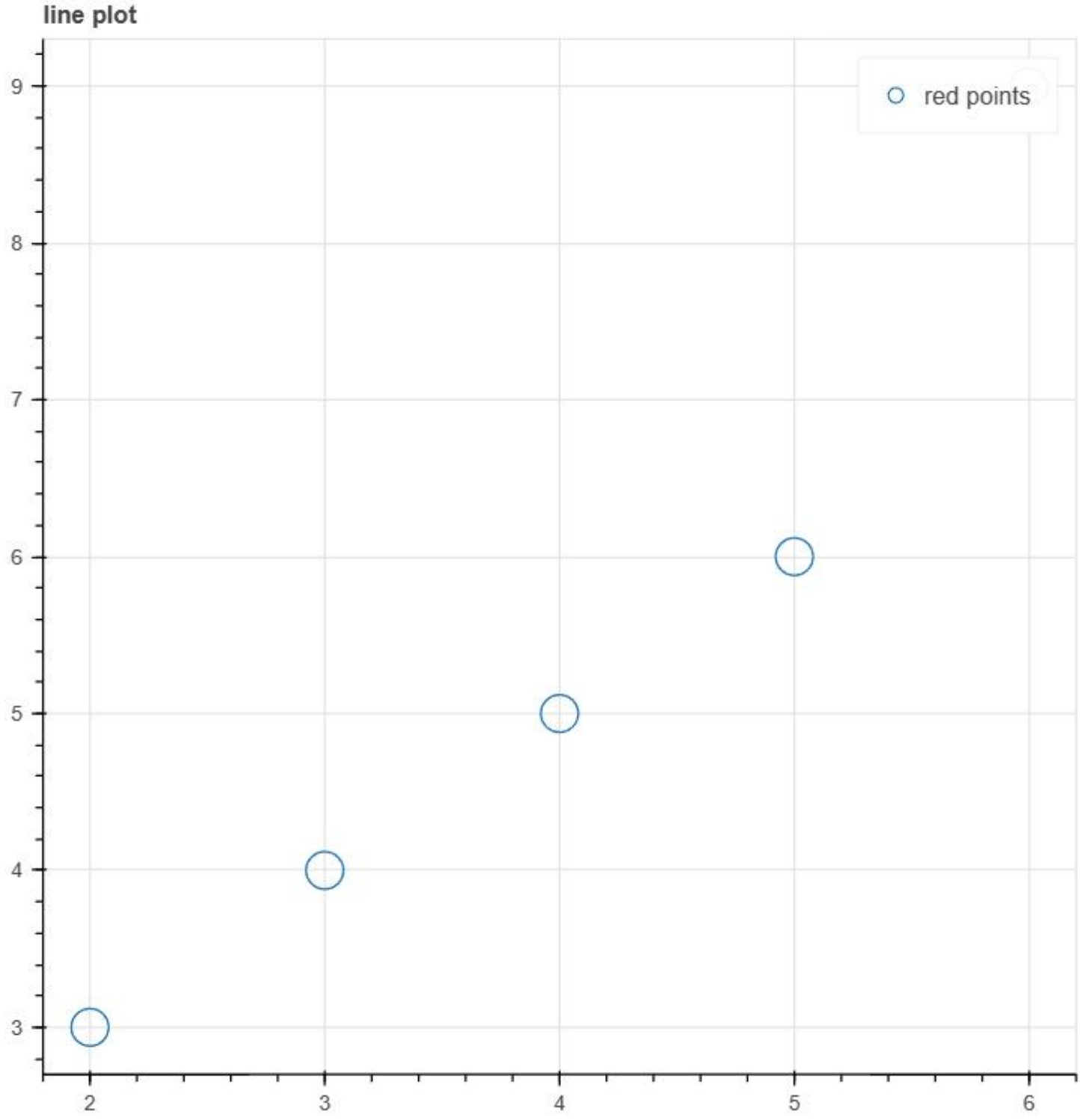

line graph
2.5
2
1.5
1
0.5
0
y axis
1
2
3
4
5
6
7
x axis

In [27]:
```
from bokeh.plotting import figure , output_file , show
from bokeh.sampledata.iris import flowers

output_file("line.html")

p = figure(title = 'line graph')
p.xaxis.axis_label = '  x axis'
p.yaxis.axis_label = 'y axis'
p.line(flowers['petal_length'] , flowers['petal_width'])
show(p)
```

In [11]:
```
x = [2,3,4,5,6]
y  = [3,4,5,6,9]

output_file("linegraph_bydataxy.html")
p = figure(title = 'line plot ')
p.line(x,y)
show(p)
```

In [15]:
```
# line plot by given data
x = [2,3,4,5,6]
y  = [3,4,5,6,9]

output_file("linegraph_bydataxy2.html")
p = figure(title = 'line plot ')
p.scatter(x,y , fill_color = 'p2086' , legend_label = 'red points ',size = 20)
show(p)
```

```
ERROR:bokeh.core.validation.check:E-1001 (BAD_COLUMN_NAME): Glyph refers to nonexistent column name. This could either be due t
o a misspelling or typo, or due to an expected column being missing. : fill_color='p2086' [no close matches] {renderer: GlyphRe
nderer(id='p2926', ...)}
```


line plot
red points
9
8
7
6
5
4
3
2
3
4
5
6

from bokeh.plotting import figure , output_file ,show from bokeh.sampledata.dataset_name import dataset

give output html file

output_file('output_file.html ')

title name

p = figure(title = 'test flower')

xlabel

p.xaxis.axis_label = 'patel length'

ylabel

p.yaxis.axis_label = ' patel_width'

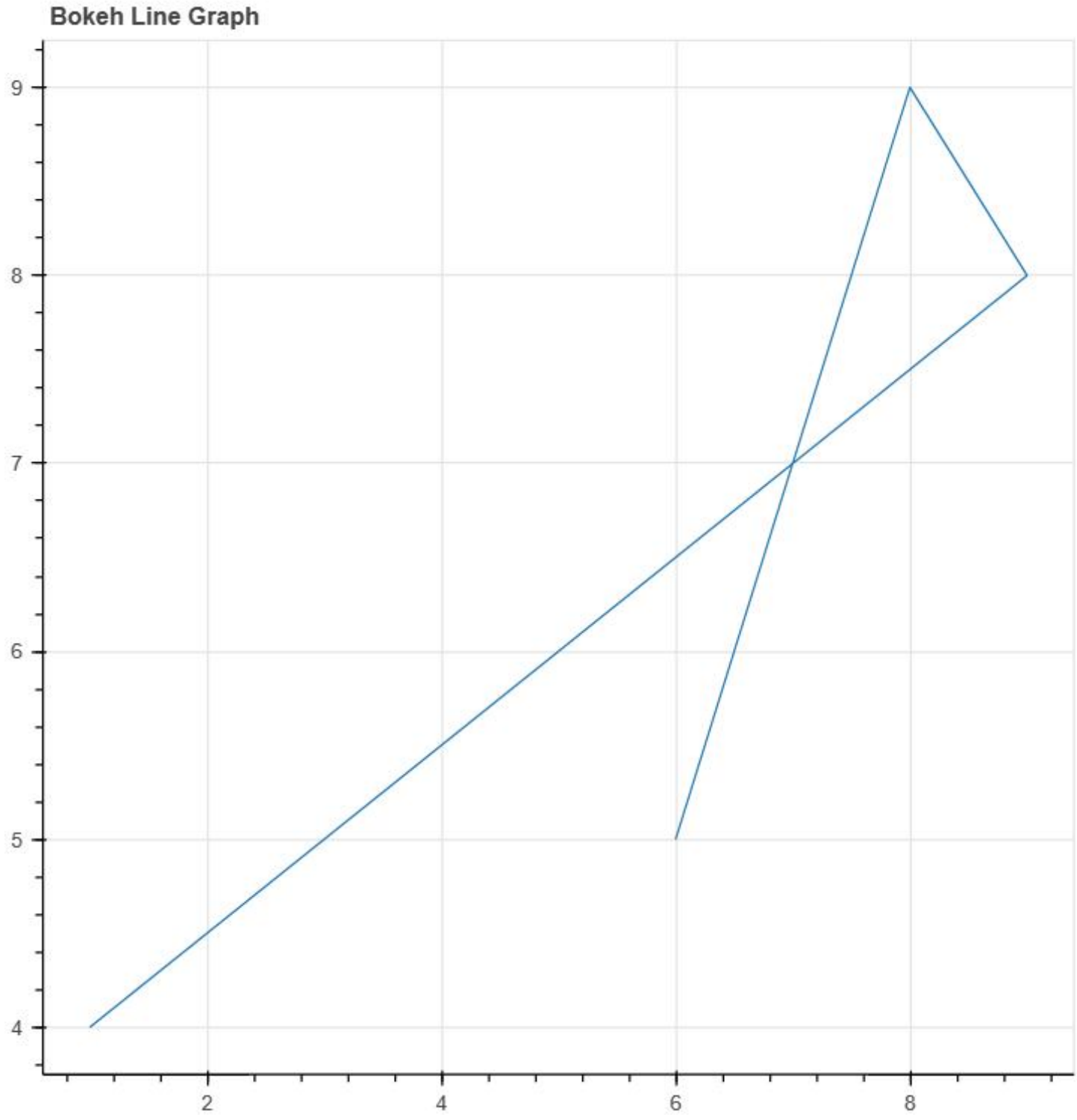
p.circle(x,y)

p.circle(flowers['petal_length'],flowers['petal_width'])

show()

show(p)

In [18]:
```
# Bokeh line graph
x = [1,9,8,7,6]
y = [4,8,9,7,5]
from bokeh.plotting import figure , output_file , show
graph = figure(title = ' Bokeh Line Graph')

graph.line(x,y)
show(graph)
```

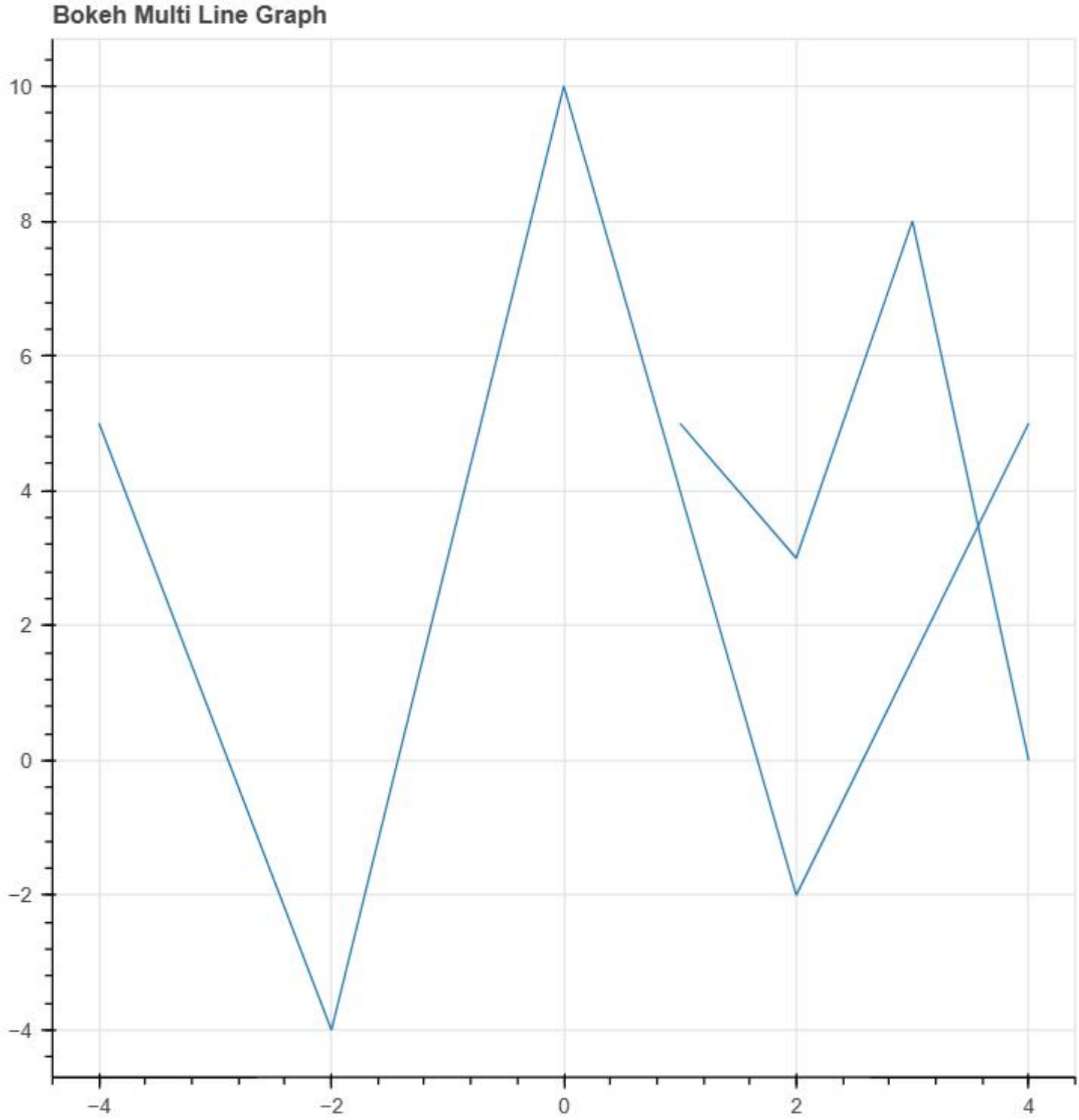

Bokeh Line Graph
9
8
7
6
5
4
2
4
6
8

In [3]:

```
# multiple lines
from bokeh.plotting import figure, output_file, show


output_file("multiple_lines.html")

# instantiating the figure object
graph = figure(title = "Bokeh Multi Line Graph")

# the points to be plotted
x = [[1, 2, 3, 4, 5], [-4, -2, 0, 2, 4]]
y = [[5, 3, 8, 0], [5, -4, 10, -2, 5]]

# plotting the graph
graph.multi_line(x, y)

# displaying the model
show(graph)
```

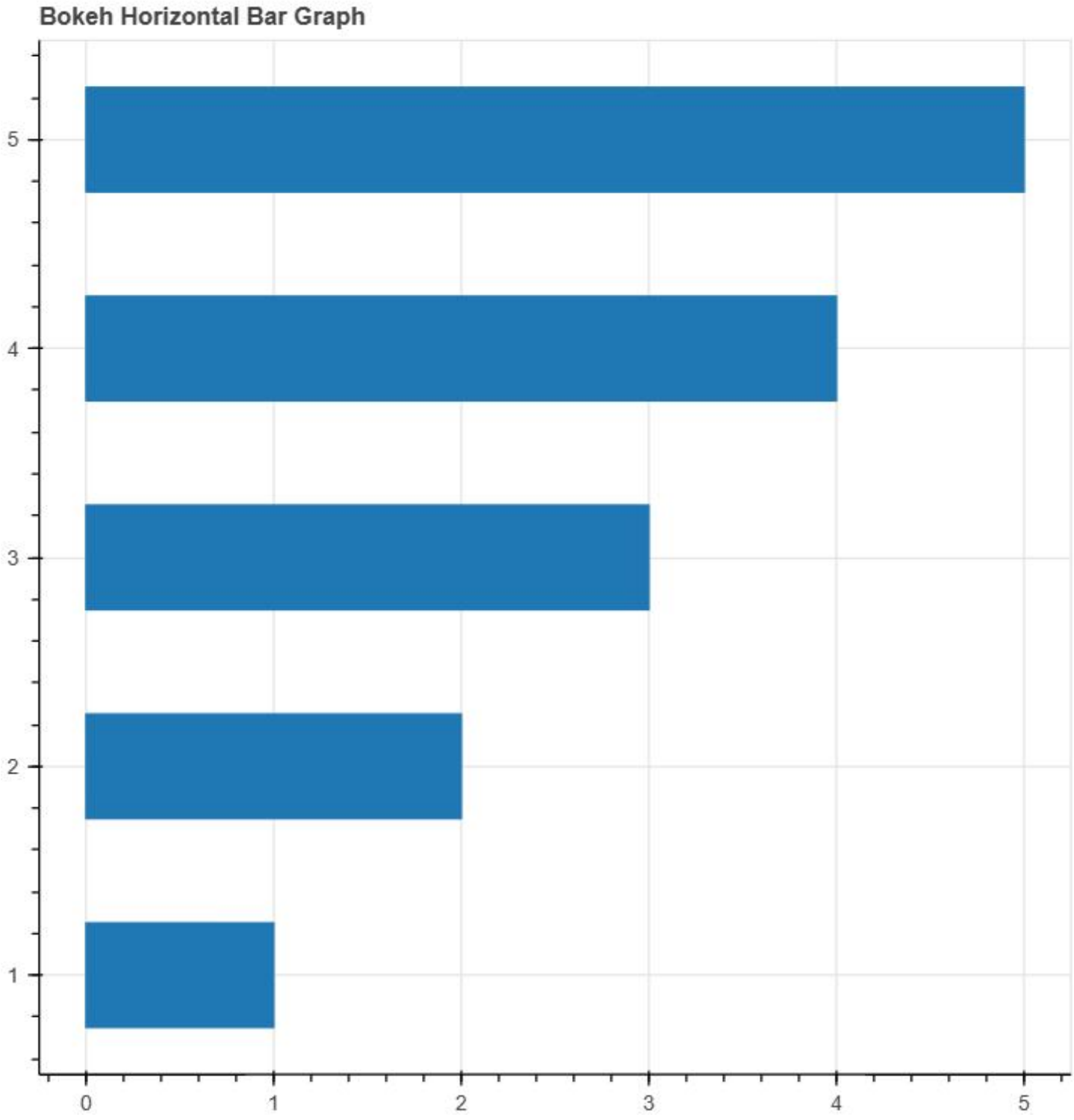
In [3]:

```
# multiple lines
from bokeh.plotting import figure, output_file, show


output_file("multiple_lines.html")

# instantiating the figure object
graph = figure(title = "Bokeh Multi Line Graph")

# the points to be plotted
x = [[1, 2, 3, 4, 5], [-4, -2, 0, 2, 4]]
y = [[5, 3, 8, 0], [5, -4, 10, -2, 5]]

# plotting the graph
graph.multi_line(x, y)

# displaying the model
show(graph)
```


Bokeh Multi Line Graph
10
8
6
4
2
0
−2
−4
−4
−2
0
2
4

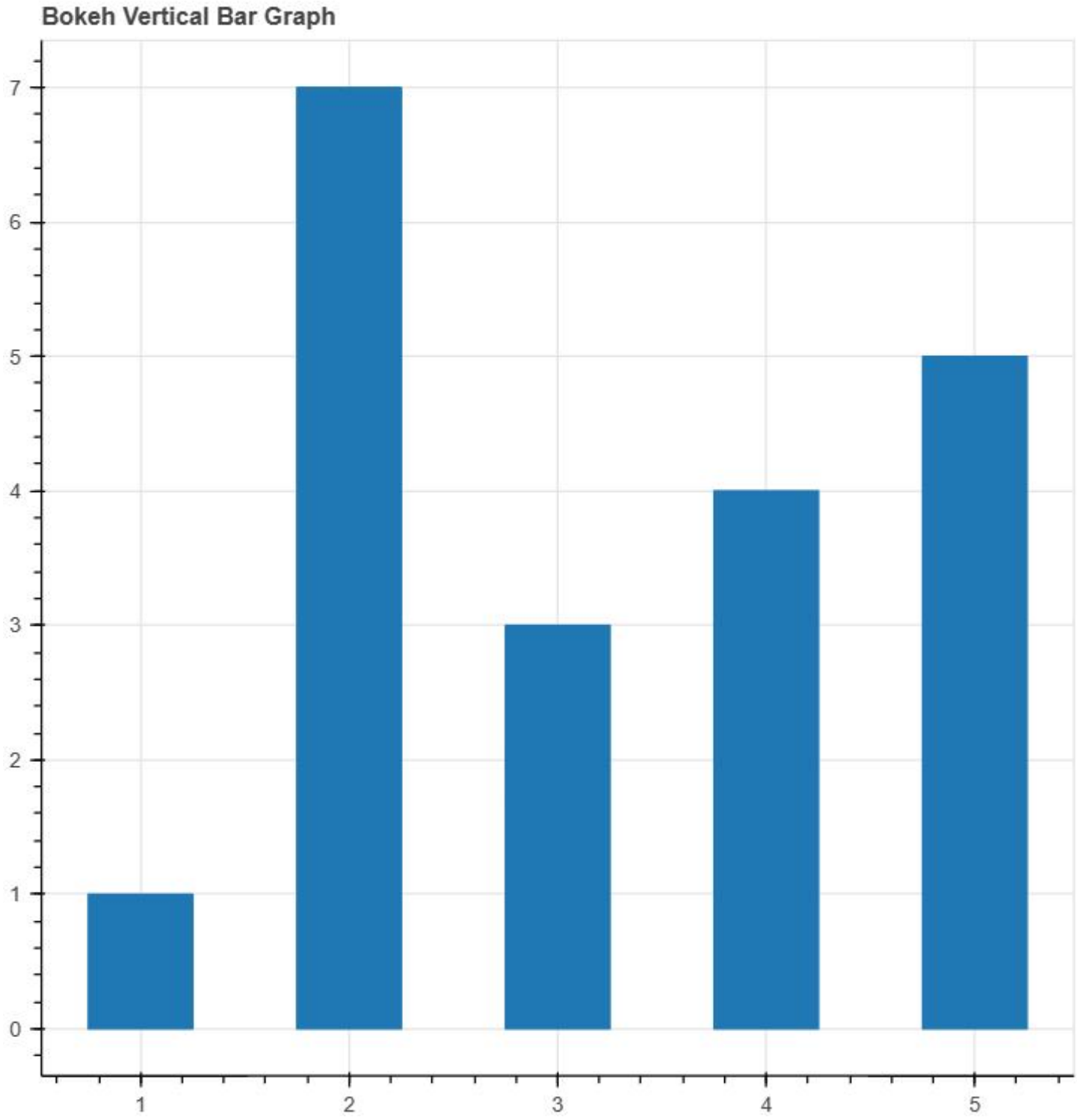
In [4]:

```
# horizontal bar graph
# importing the modules
from bokeh.plotting import figure, output_file, show

# file to save the model
output_file("horizontal_bar.html")
graph = figure(title = "Bokeh Horizontal Bar Graph")

# y-coordinates to be plotted
y = [1, 2, 3, 4, 5]

# x-coordinates of the right edges
right = [1, 2, 3, 4, 5]

# height / thickness of the bars
height = 0.5

# plotting the graph
graph.hbar(y,right = right,height = height)

# displaying the model
show(graph)
```


Bokeh Horizontal Bar Graph
5
4
3
2
1
0
1
2
3
4
5

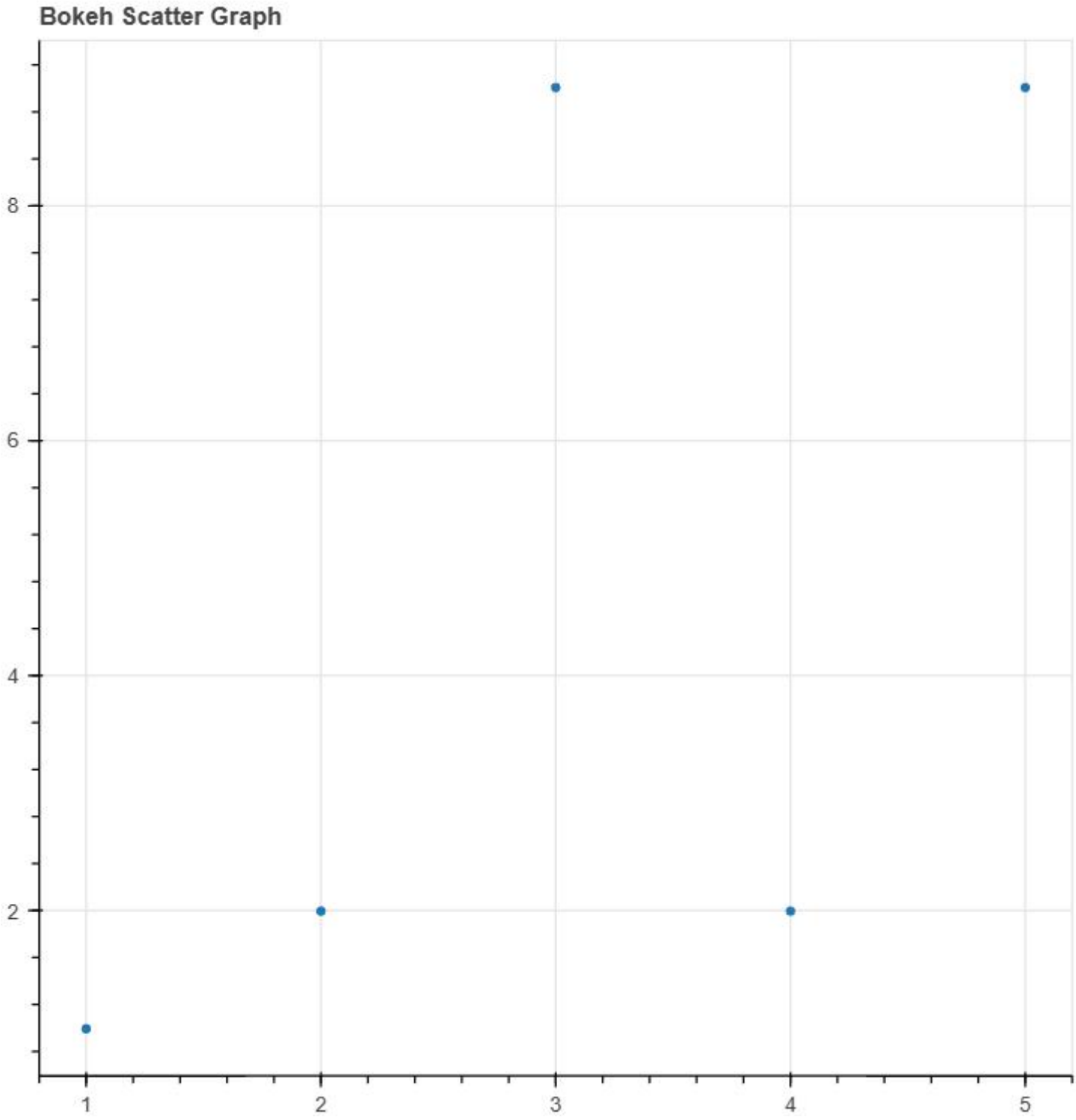
In [6]:

```
# vertical bar graph
# importing the modules
from bokeh.plotting import figure, output_file, show

# file to save the model
output_file("horizontal_bar.html")

graph = figure(title = "Bokeh Vertical Bar Graph")

# x-coordinates to be plotted
x = [1, 2, 3, 4, 5]

# x-coordinates of the top edges
top = [1, 7, 3, 4, 5]

# width / thickness of the bars
width = 0.5

# plotting the graph
graph.vbar(x,top = top,width = width)

# displaying the model
show(graph)
```


Bokeh Vertical Bar Graph
7
6
5
4
3
2
1
0
1
2
3
4
5

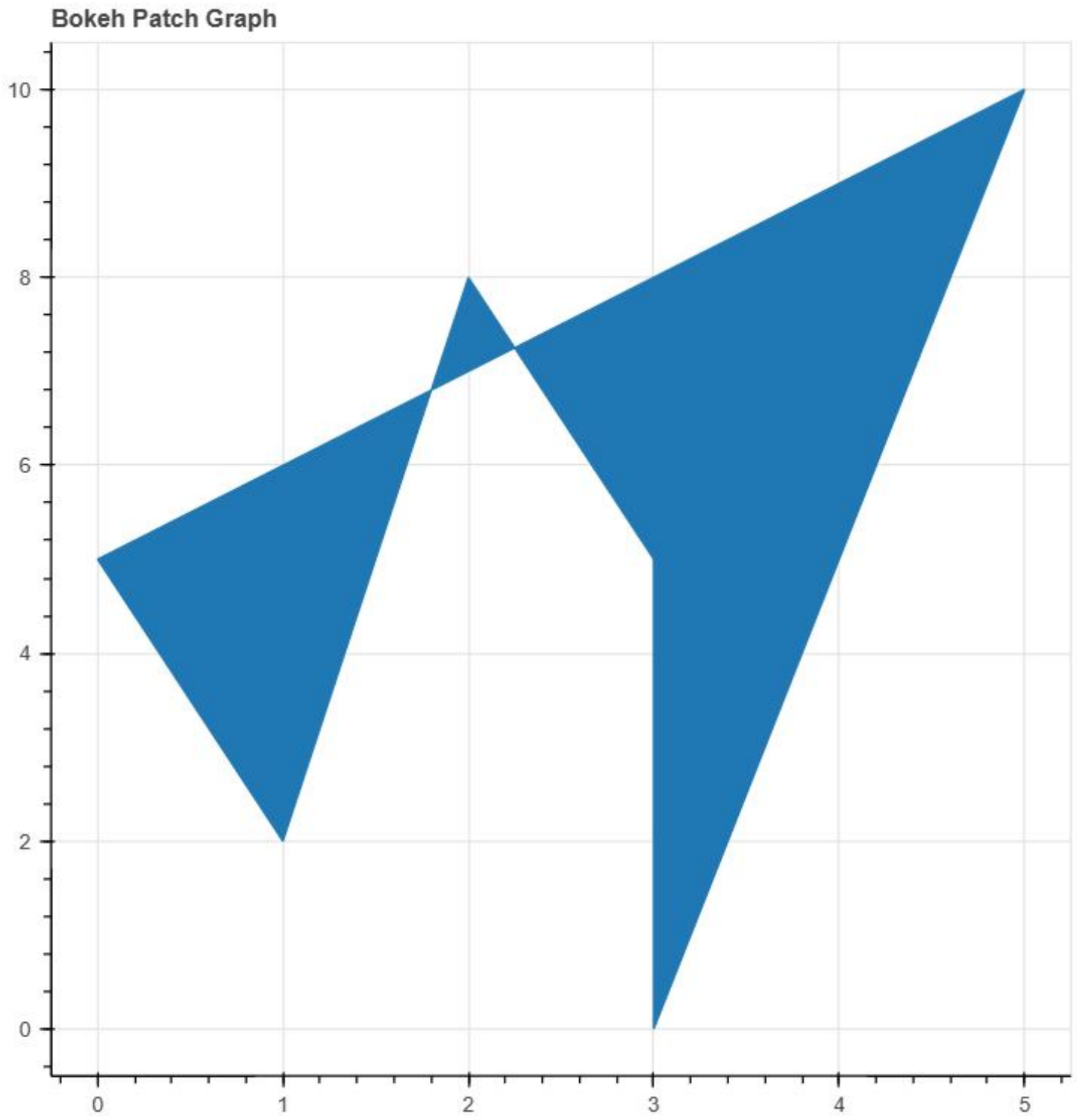
In [7]:

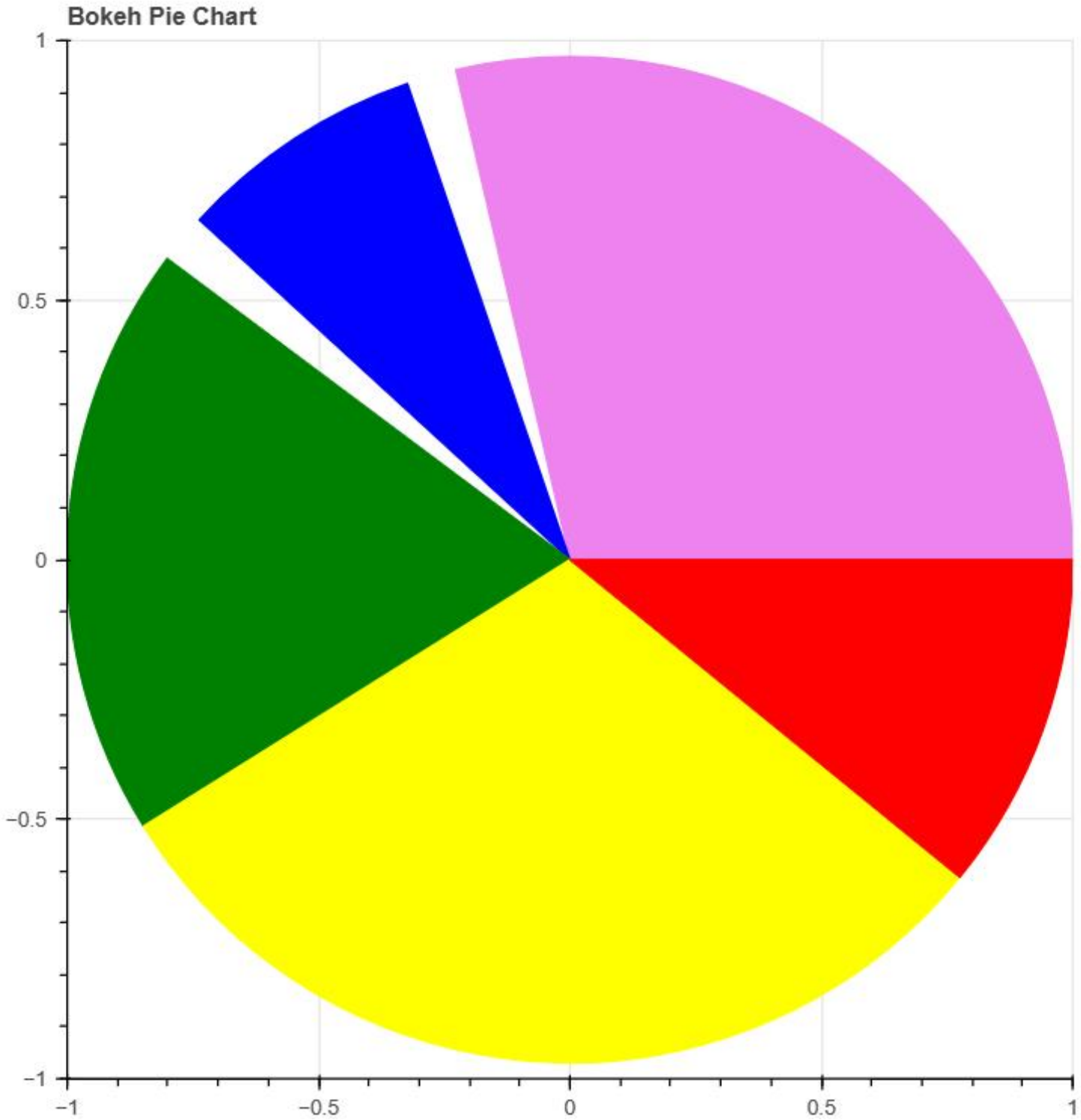
```
# scatter plot
# importing the modules
from bokeh.plotting import figure, output_file, show

# file to save the model
output_file("scatter_plot.html")

graph = figure(title = "Bokeh Scatter Graph")

# the points to be plotted
x = [1, 2, 3, 4, 5]
y = [1, 2, 9, 2, 9]

# plotting the graph
graph.scatter(x, y)

# displaying the model
show(graph)
```


Bokeh Scatter Graph
8
6
4
2
1
2
3
4
5

In [8]:
```
# plotting patches
# importing the modules
from bokeh.plotting import figure, output_file, show

output_file("gfg.html")

graph = figure(title = "Bokeh Patch Graph")

# the points to be plotted
x = [0, 1, 2, 3, 3, 5]
y = [5, 2, 8, 5, 0, 10]

# plotting the graph
graph.patch(x, y)

# displaying the model
show(graph)
```

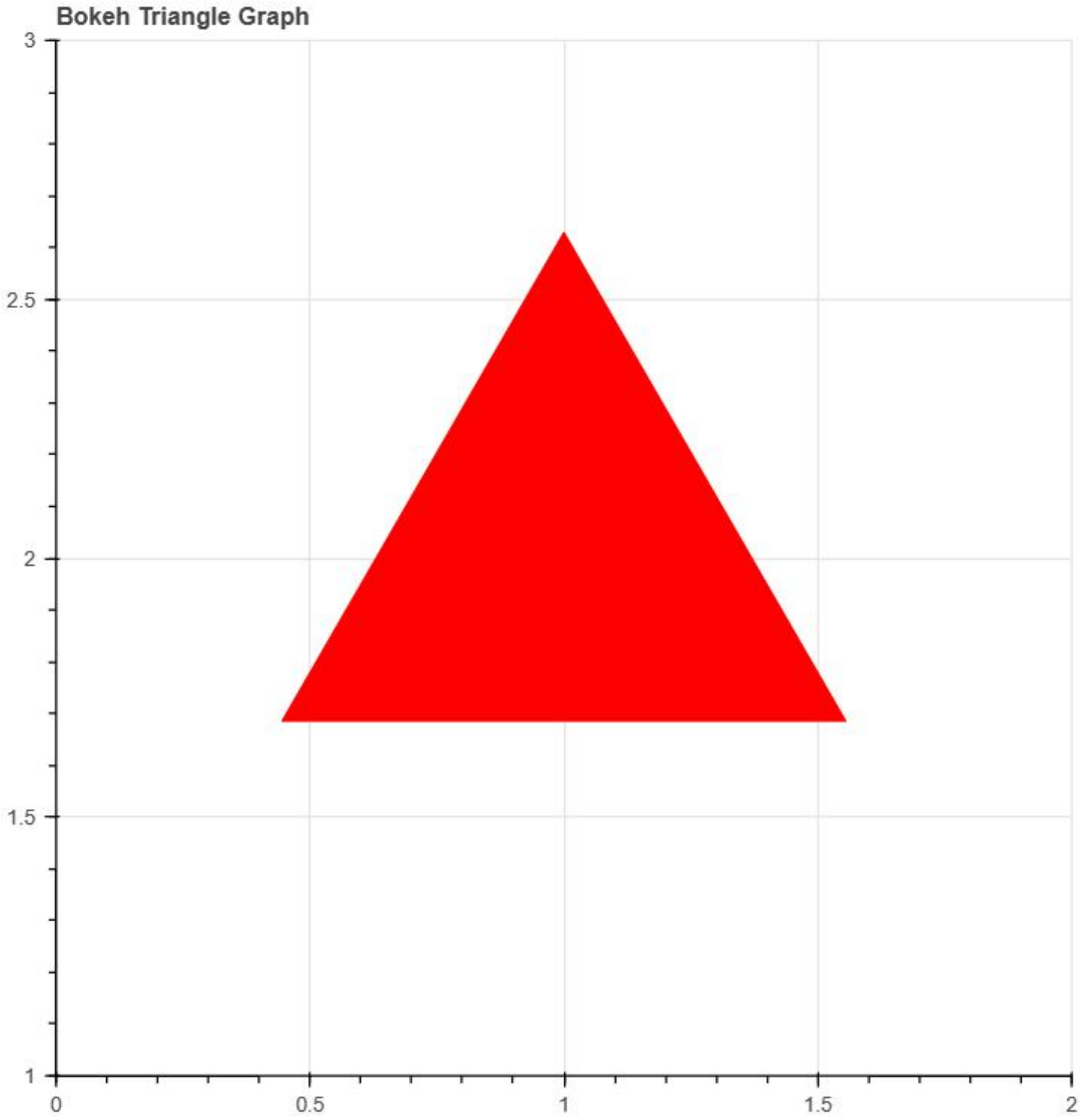
In [14]:
```
# pie chart
# importing the modules
from bokeh.plotting import figure, output_file, show

# file to save the model
output_file("pie_chart.html")

# instantiating the figure object
graph = figure(title = "Bokeh Pie Chart")

# center of the pie chart
x = 0
y = 0

# radius of the glyphs
radius = 1
```

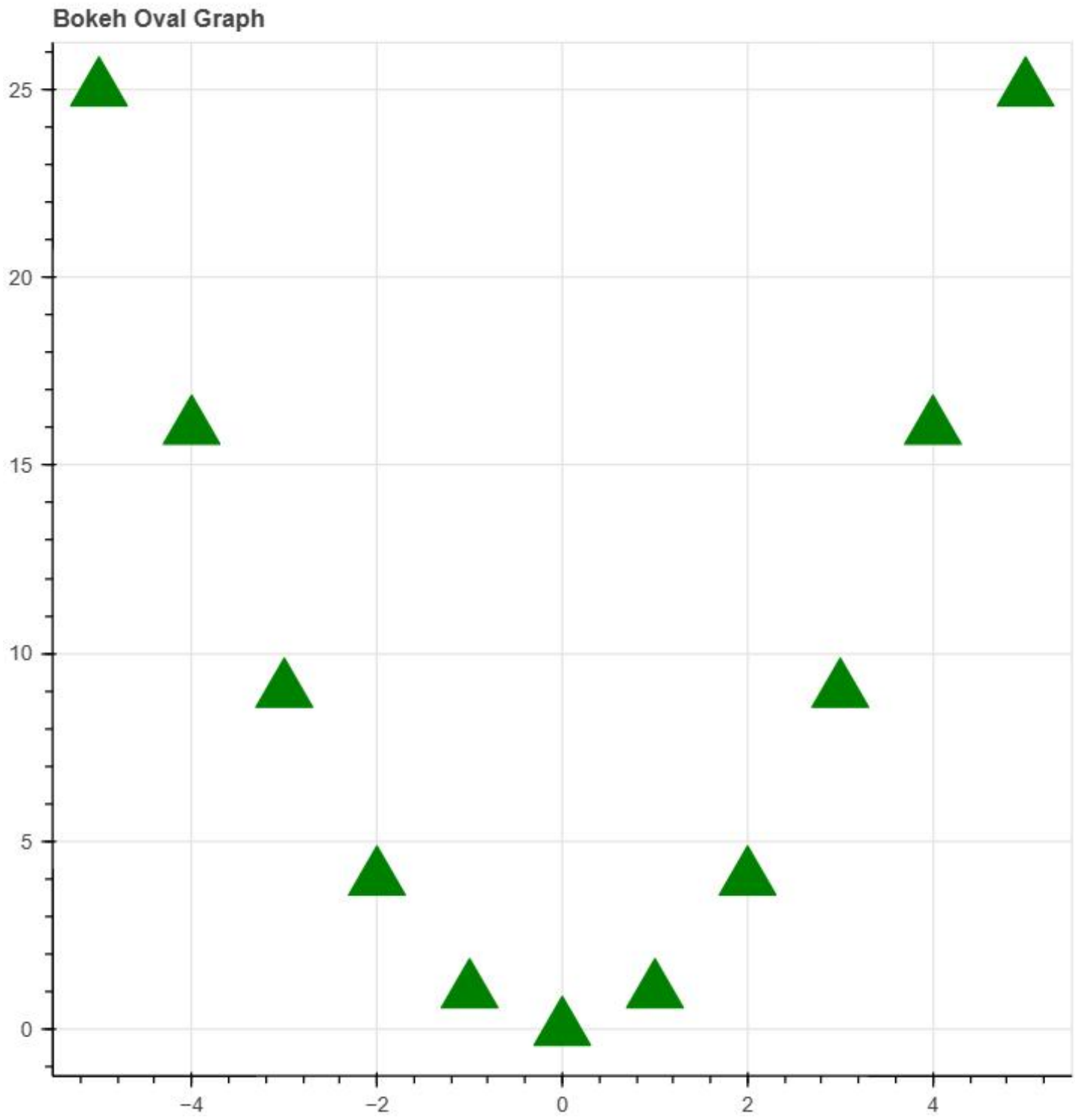

Bokeh Patch Graph
10
8
6
4
2
0
0
1
2
3
4
5


Bokeh Pie Chart
1
0.5
0
−0.5
−1
−1
−0.5
0
0.5
1

In [17]:
```
# plotting triangle on graph
# importing the modules
from bokeh.plotting import figure, output_file, show
# file to save the model
output_file("plot_triangle.html")
# instantiating the figure object
graph = figure(title = "Bokeh Triangle Graph")
# the points to be plotted
x = 1
y = 2
# plotting the graph
graph.triangle(x, y, size = 300 , color = 'red')
# displaying the model
show(graph)
```

In [21]:
```
# plotting oval
# importing the modules
from bokeh.plotting import figure, output_file, show
# file to save the model
output_file("plot_oval.html")

# instantiating the figure object
graph = figure(title = "Bokeh Oval Graph")

# the points to be plotted
x = [-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5]
y = [i ** 2 for i in x]

# plotting the graph
graph.triangle(x, y, size = 30, color = 'green')
# displaying the model
show(graph)
```


Bokeh Triangle Graph
3
2.5
2
1.5
1
0
0.5
1
1.5
2


Bokeh Oval Graph
25
20
15
10
5
0
−4
−2
0
2
4