

CSE3006 - Data Visualization

Lab Assignment 3

Name: Jai Gaurav

Reg No: 21BCE7193

Importing libraries

```
In [1]: import pandas as pd
import numpy as np
from bokeh.plotting import figure, show
from bokeh.layouts import import gridplot, column
from bokeh.models import ColumnDataSource, HoverTool, Slider, CustomJS
from bokeh.sampledata.penguins import data as df
import seaborn as sns
import matplotlib.pyplot as plt
```

Question 1

```
In [2]: df.head()
```

Out[2]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	MALE
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	FEMALE
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	FEMALE
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	FEMALE

```
In [3]: source = ColumnDataSource(df)
```

First plot

```
In [4]: hover = HoverTool(tooltips=[("Species", "@species"), ("Bill Length (mm)", "@bill_length_mm"), ("Bill Depth (mm)", "@bill_depth_mm")])
plot1 = figure(title="Bill Length Vs Bill Depth (mm)",
               tools=[hover, 'pan', 'box_zoom', 'wheel_zoom', 'reset'])
scatter1 = plot1.circle('bill_length_mm',
                       'bill_depth_mm',
                       source=source, size=5)
```

Second plot

```
In [5]: hover = HoverTool(tooltips=[("Species", "@species"), ("Bill Length (mm)", "@bill_length_mm"), ("Flipper Length (mm)", "@flipper_length_mm")])
plot2 = figure(title="Bill Length Vs Flipper Length (mm)",
               tools=[hover, 'pan', 'box_zoom', 'wheel_zoom', 'reset'])
scatter2 = plot2.circle('bill_length_mm',
                       'flipper_length_mm',
                       source=source, size=5)
```

Make the ranges same

```
In [6]: plot1.x_range = plot2.x_range
```

Add slider for point size

```
In [7]: slider = Slider(start=1, end=10, value=5, step=0.5, title="Point Size")

callback = CustomJS(args=dict(source=source, scatter1=scatter1, scatter2=scatter2),
                    code="""
scatter1.glyph.size = {'value': cb_obj.value};
scatter2.glyph.size = {'value': cb_obj.value};
source.change.emit();
""")

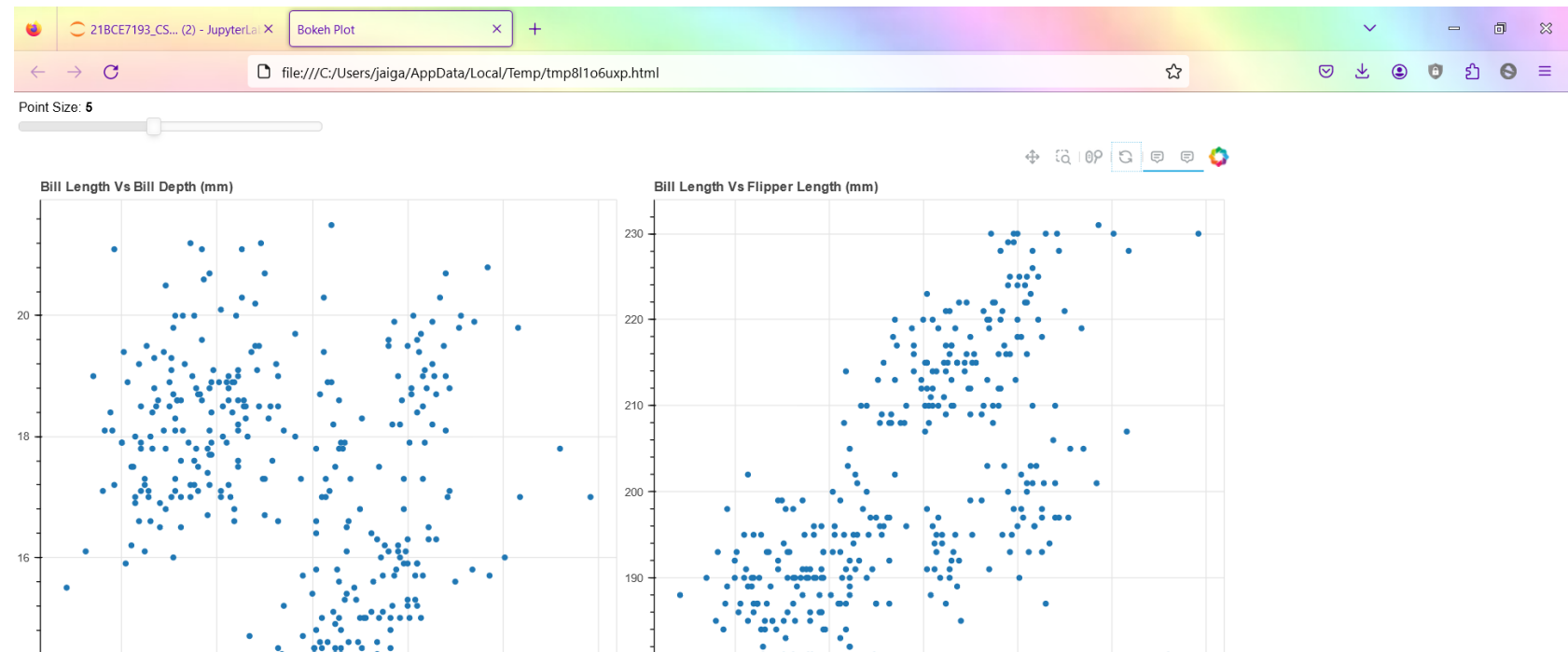
slider.js_on_change('value', callback)

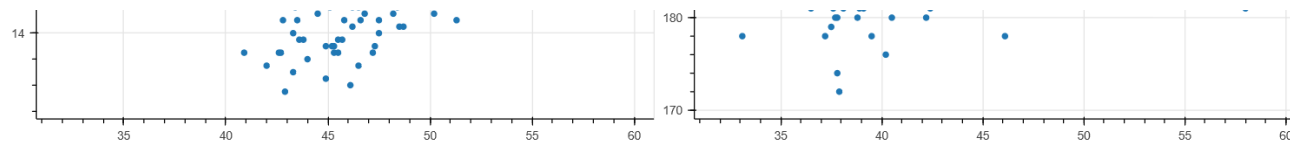
layout = column(slider, gridplot([[plot1, plot2]]))
```

Show linked view

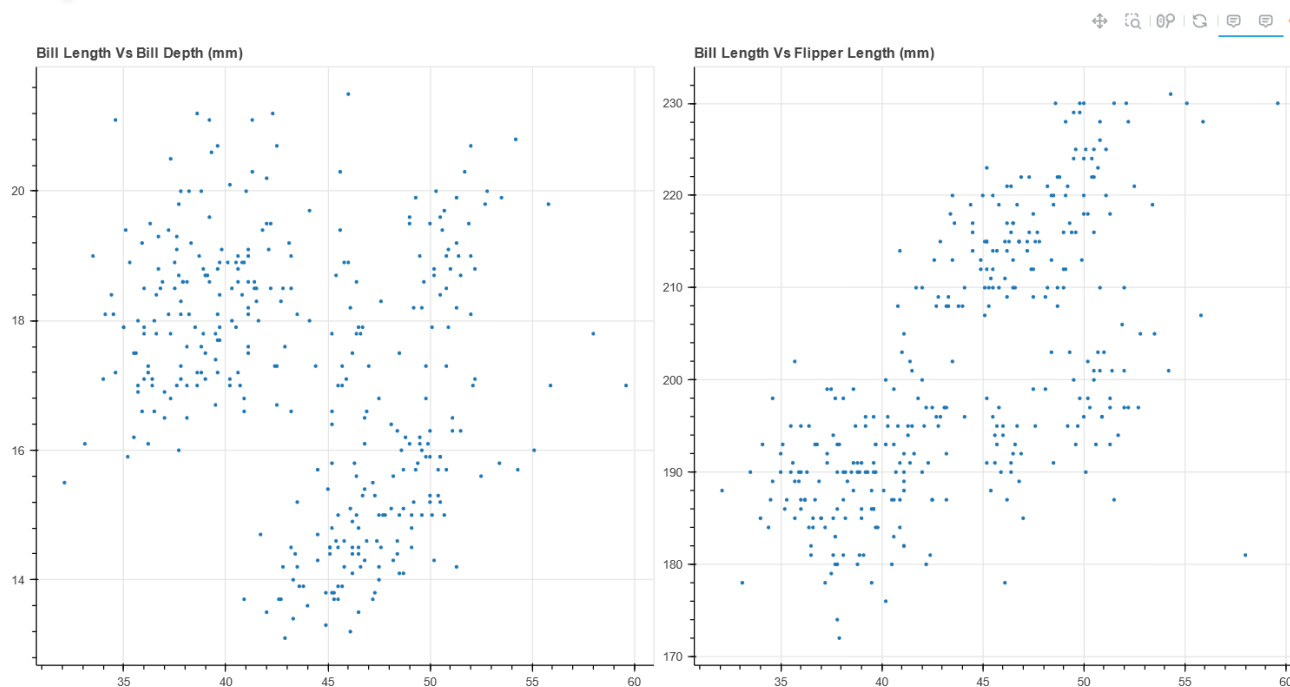
```
In [8]: show(layout)
```

Outputs



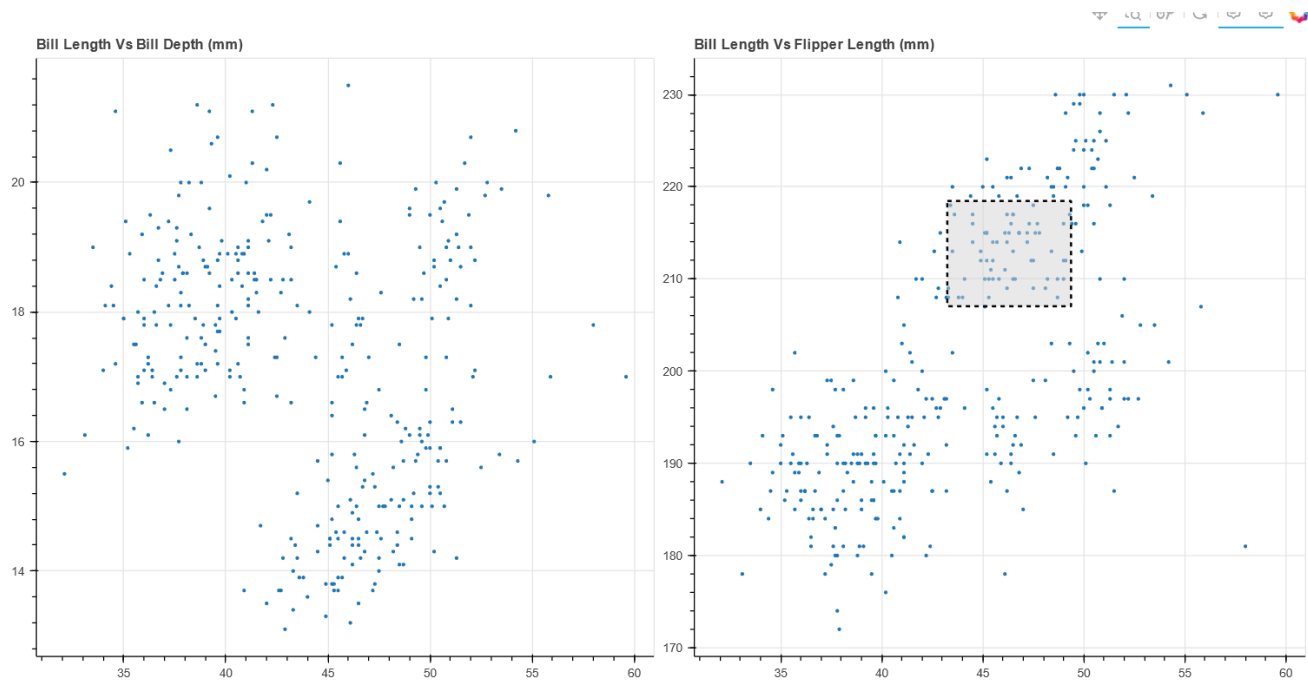


Point Size: 2.50

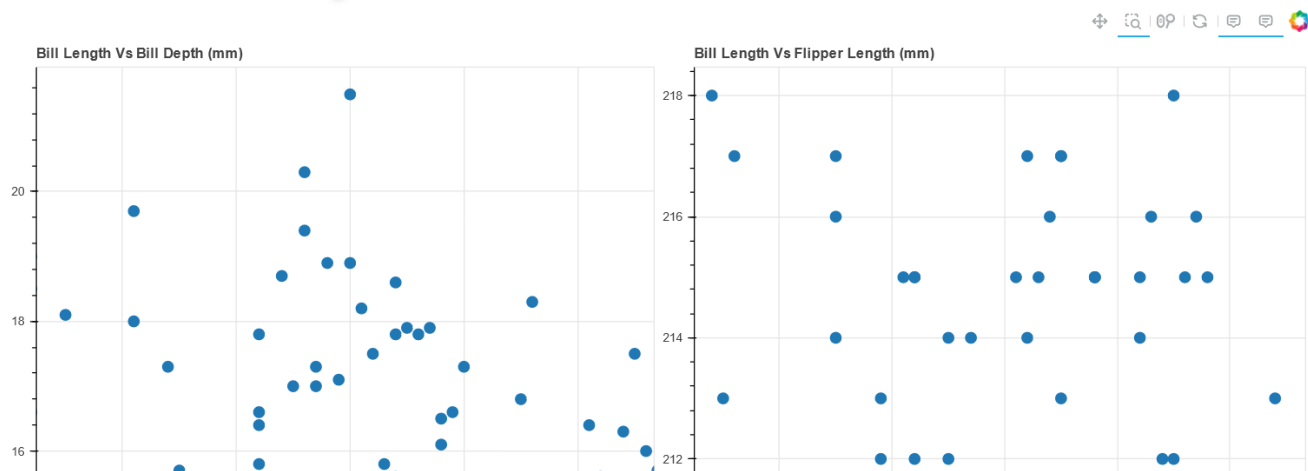


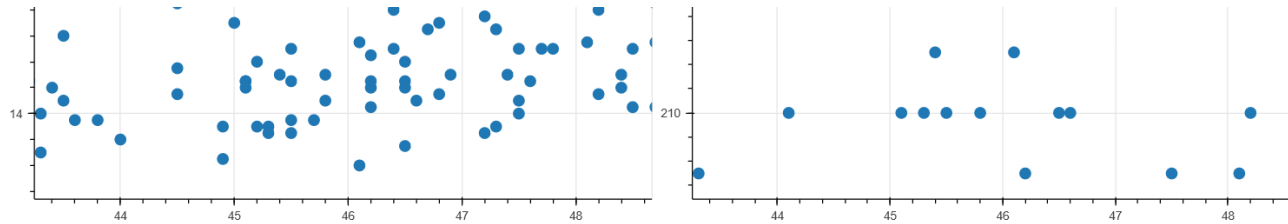
Point Size: 2.50





Point Size: 10





Question 2

```
In [9]: df = pd.read_csv('cloud.csv')
df.head()
```

Out[9]:

	Visible_mean	Visible_max	Visible_min	mean distribution	contrast	entropy	second angular momentum	IR_mean	IR_max	IR_min
0	3	140	43.5000	0.0833	862.8417	0.0254	3.8890	163	240	213.3555
1	3	135	41.9063	0.0790	690.3291	0.0259	3.8340	167	239	213.7188
2	2	126	21.0586	0.0406	308.3583	0.0684	3.1702	174	240	227.5859
3	4	197	77.4805	0.0890	874.4709	0.0243	3.9442	155	239	197.2773
4	7	193	88.8398	0.0884	810.1126	0.0223	3.9318	150	236	186.0195

```
In [10]: df.shape
```

Out[10]: (1024, 10)

```
In [11]: df.describe()
```

```
Out[11]:
```

	Visible_mean	Visible_max	Visible_min	mean distribution	contrast	entropy	second angular momentum	IR_mean	IR_max	IR_min
count	1024.000000	1024.000000	1024.000000	1024.000000	1024.000000	1024.000000	1024.000000	1024.000000	1024.000000	1024.000000
mean	6.367188	127.543945	40.536434	0.056755	538.81511	0.123346	3.139171	186.080078	243.142578	223.390078
std	7.181350	60.035897	30.852071	0.035038	474.42015	0.196783	1.014281	33.343255	8.593411	21.616078
min	1.000000	2.000000	1.027300	0.000000	0.000000	0.014500	0.000000	1.000000	148.000000	80.316078
25%	3.000000	88.000000	15.249975	0.028125	142.26875	0.026100	2.801275	165.000000	239.000000	211.640078
50%	5.000000	140.000000	34.593750	0.057800	453.79170	0.039550	3.528050	183.500000	245.000000	227.312078
75%	7.000000	173.250000	59.641600	0.084125	831.59270	0.100925	3.853300	207.000000	249.000000	240.669078
max	96.000000	255.000000	167.148400	0.174700	3211.47530	1.000000	4.382800	250.000000	255.000000	252.945078

Linked View for Cloud Multivariate Dataset: Visible VS IR Mean, Min and Max

```
In [12]: fig, axes = plt.subplots(3, 1, figsize=(10, 15))

sns.regplot(x='Visible_mean', y='IR_mean', data=df, ax=axes[0], scatter=False)
axes[0].set_title('Visible Mean vs IR Mean')

sns.regplot(x='Visible_min', y='IR_min', data=df, ax=axes[1], scatter=False)
axes[1].set_title('Visible Min vs IR Min')

sns.regplot(x='Visible_max', y='IR_max', data=df, ax=axes[2], scatter=False)
axes[2].set_title('Visible Max vs IR Max')

for ax in axes:
    ax.set_axisbelow(True)
    ax.grid(True)

plt.show()
```