Graphs using Python

Topic's

1.MatplotLib

2.Plotly

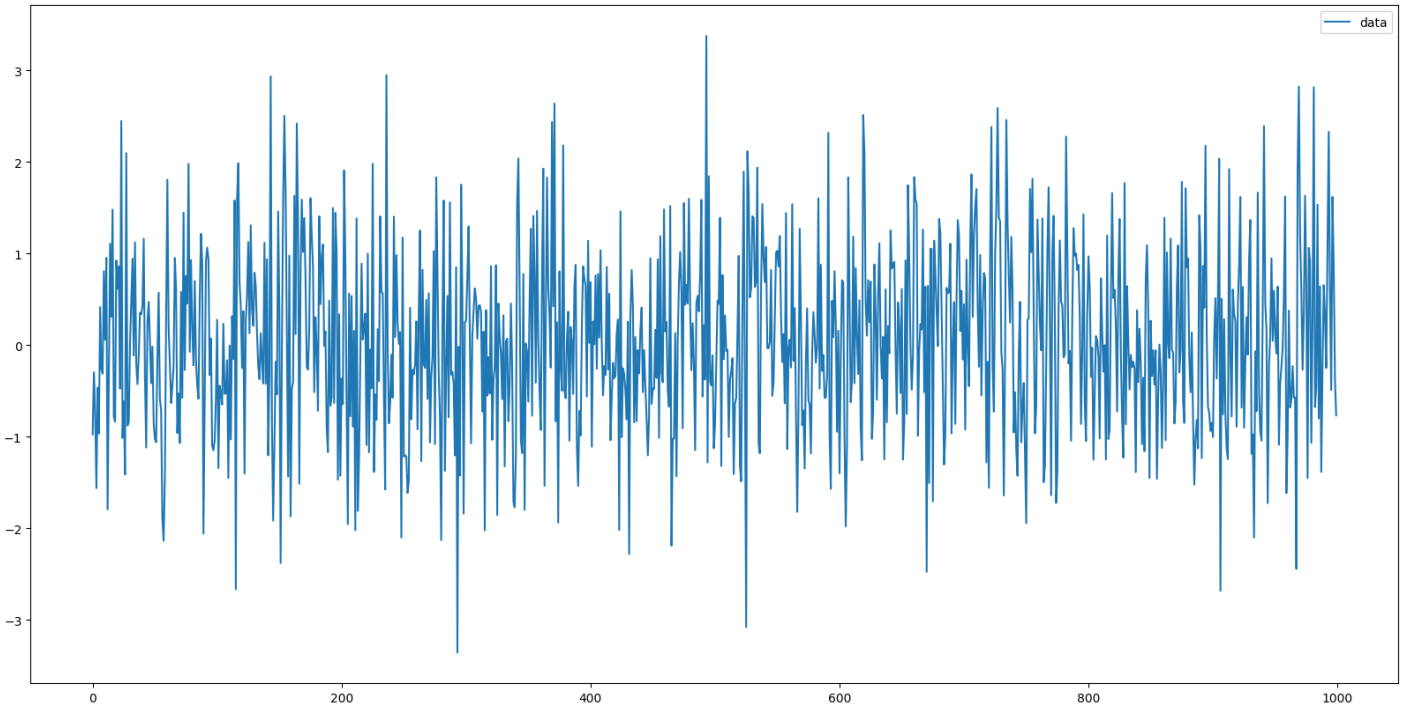
3.Seaborn

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib as plt
```

In [2]:
```
df=pd.DataFrame(np.random.randn(1000),columns=['data'])
```

In [3]:
```
#using plot to plot the data in graph  in pandas dataframe.
#using figsize parameter to set the length and width of represented graph
df.plot(figsize=(20,10))
```

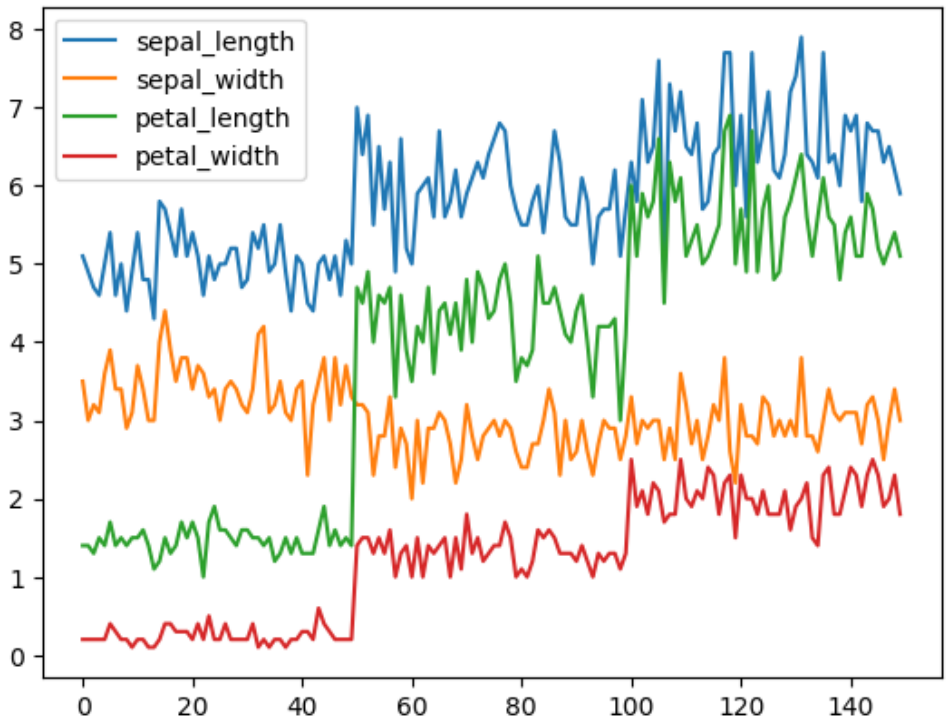
Out[3]: <AxesSubplot:>



In [4]:
```
df=pd.read_csv("https://gist.githubusercontent.com/curran/a08a1080b88344b0c8a7/raw/0e7a9b0a5d22642a
```

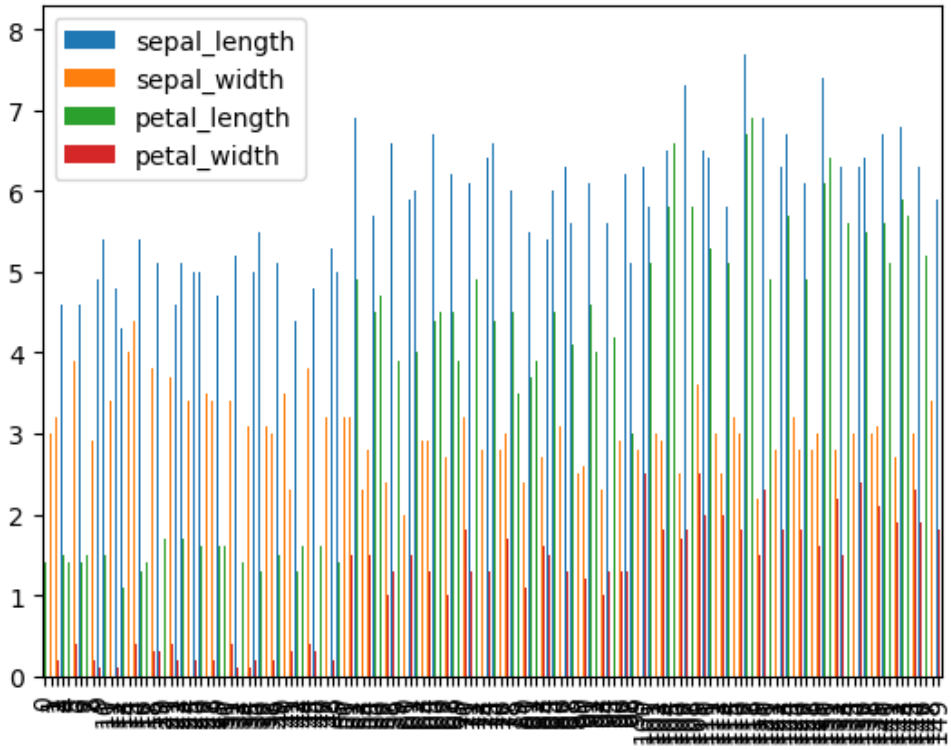
In [5]:
```
df.plot()
```

Out[5]: <AxesSubplot:>



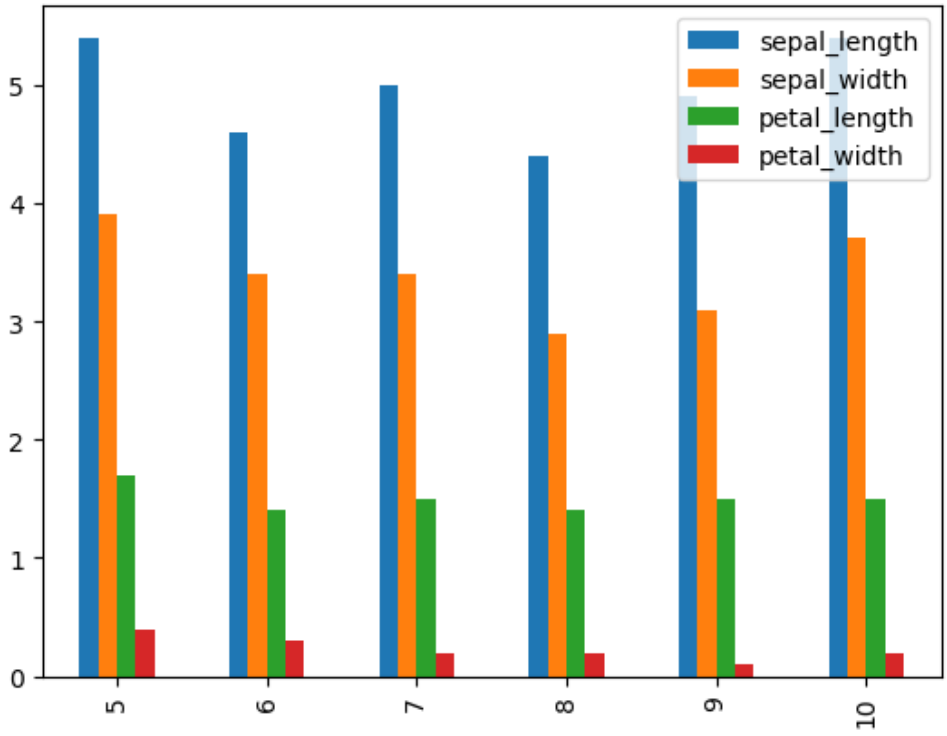
In [6]:
```
#we use kind argument to select the type of graph
df.plot(kind='bar')
```

Out[6]: <AxesSubplot:>



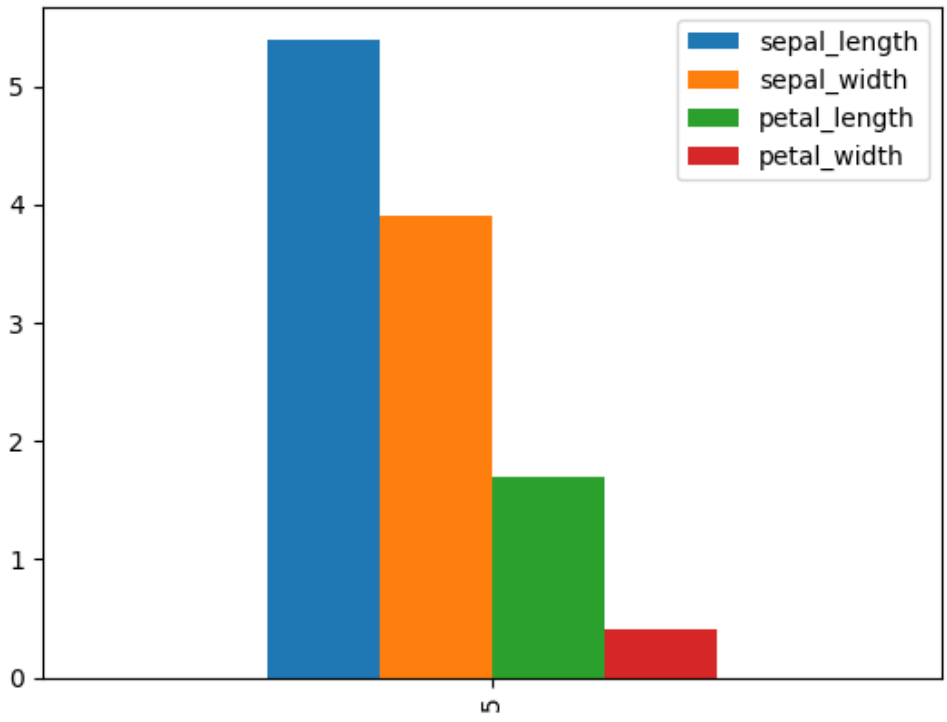
In [7]:
```
df.iloc[5:11].plot(kind='bar')
```

Out[7]: <AxesSubplot:>



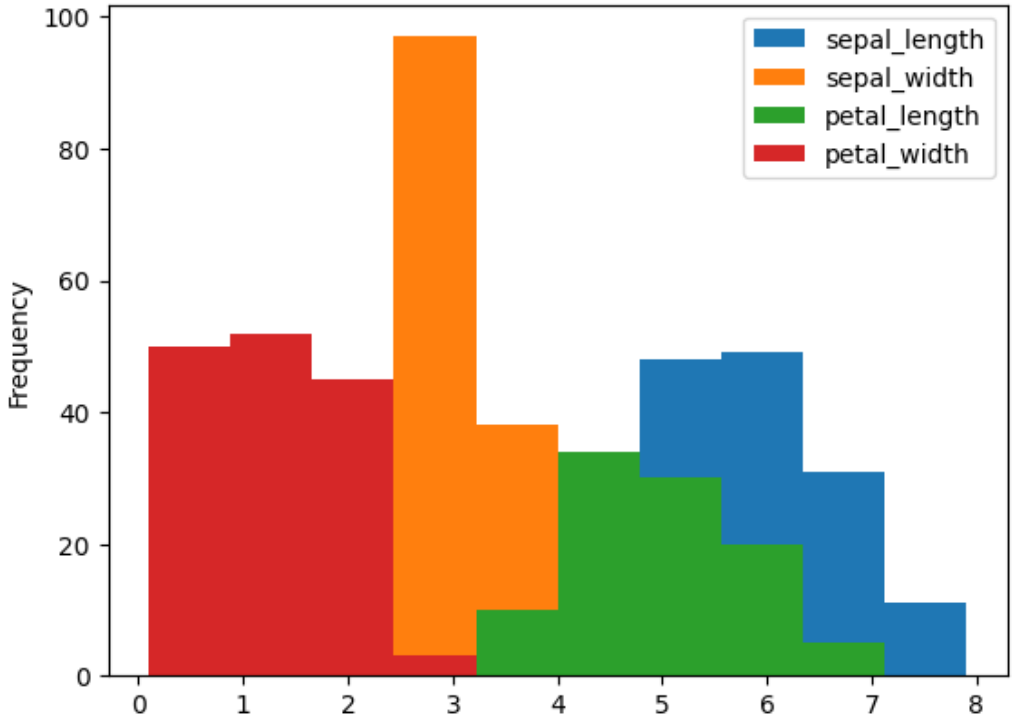
In [8]:
```
#ploting bar graph for particular row using iloc

df.iloc[[5]].plot(kind='bar')
```

Out[8]: <AxesSubplot:>



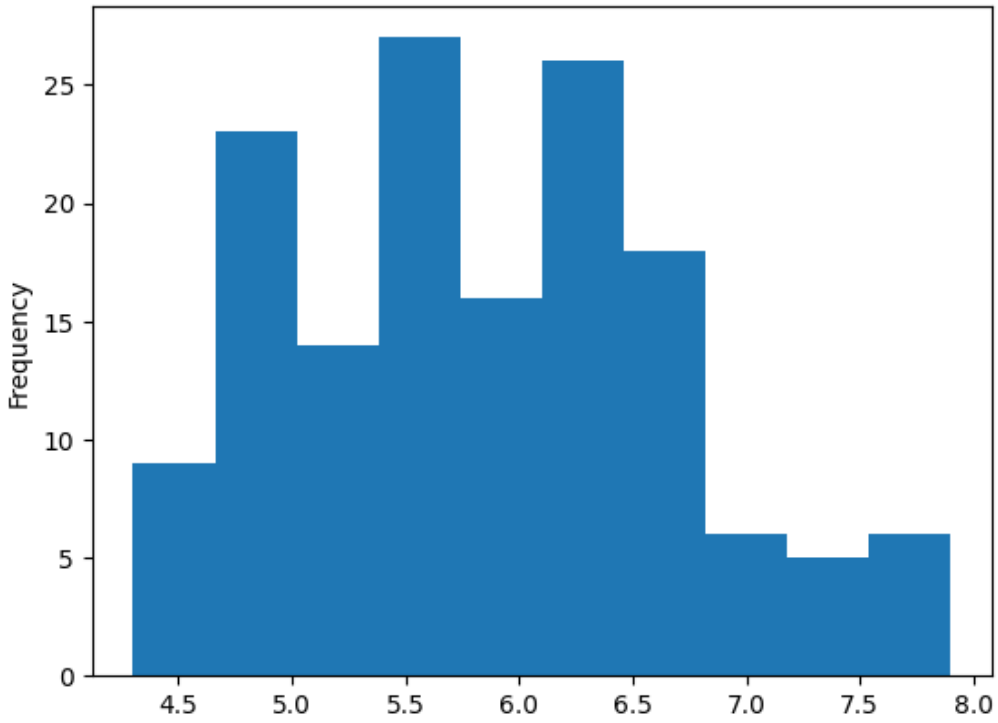
In [9]:
```
#ploting histogranm chart
'''
in hist graph X
    x -axis represents range
    y -axis represdents frequency
'''
#it represents data in form of bins
df.plot(kind='hist')
```

Out[9]: <AxesSubplot:ylabel='Frequency'>



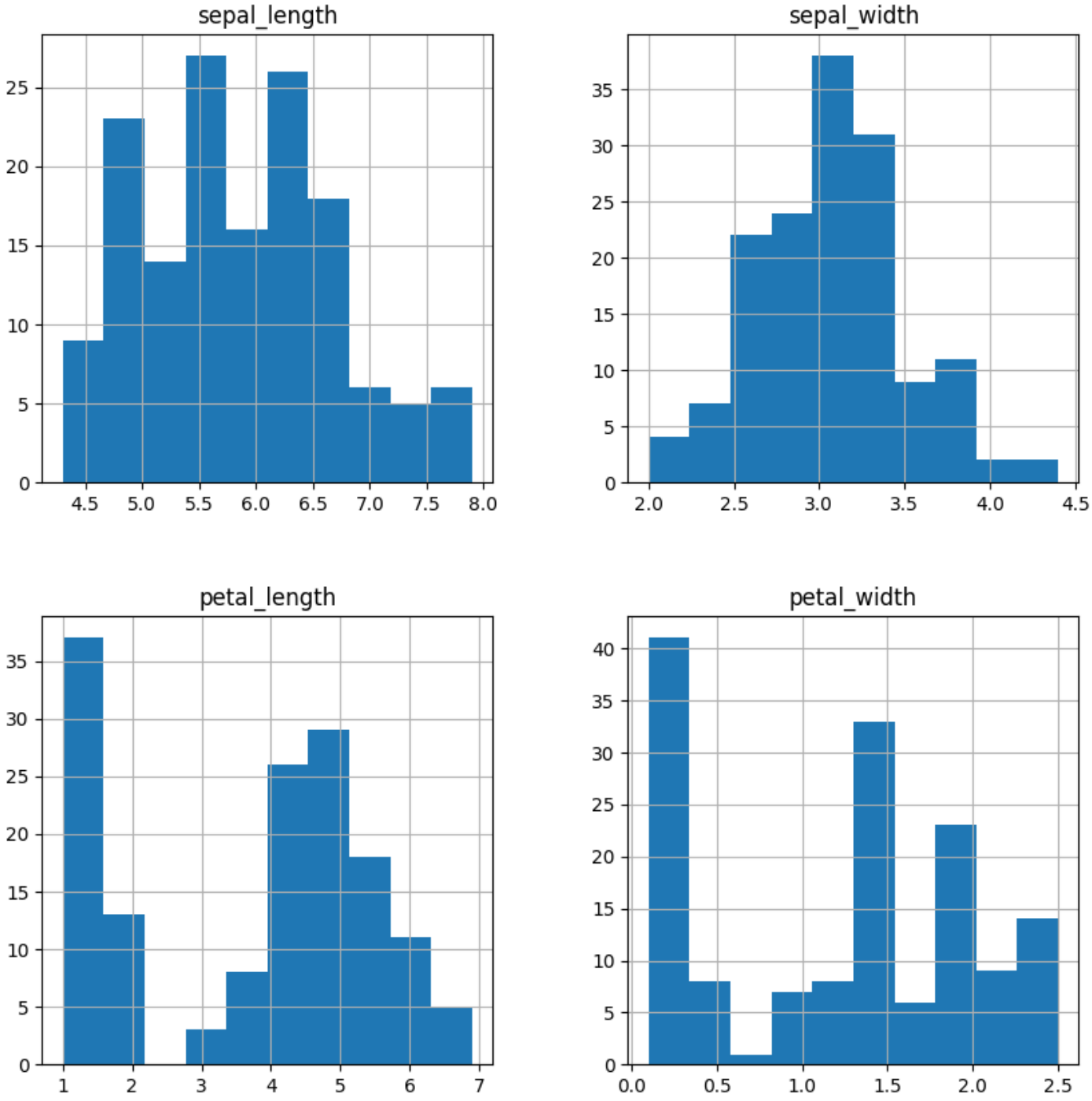
In [10]:
```
#ploting histogranm chart for particular column
df['sepal_length'].plot(kind='hist')

# to rotate the graph we use orientation parameter.
#df['sepal_length'].plot(kind='hist',orientation='horizontal')
```

Out[10]:
```
<AxesSubplot:ylabel='Frequency'>
```



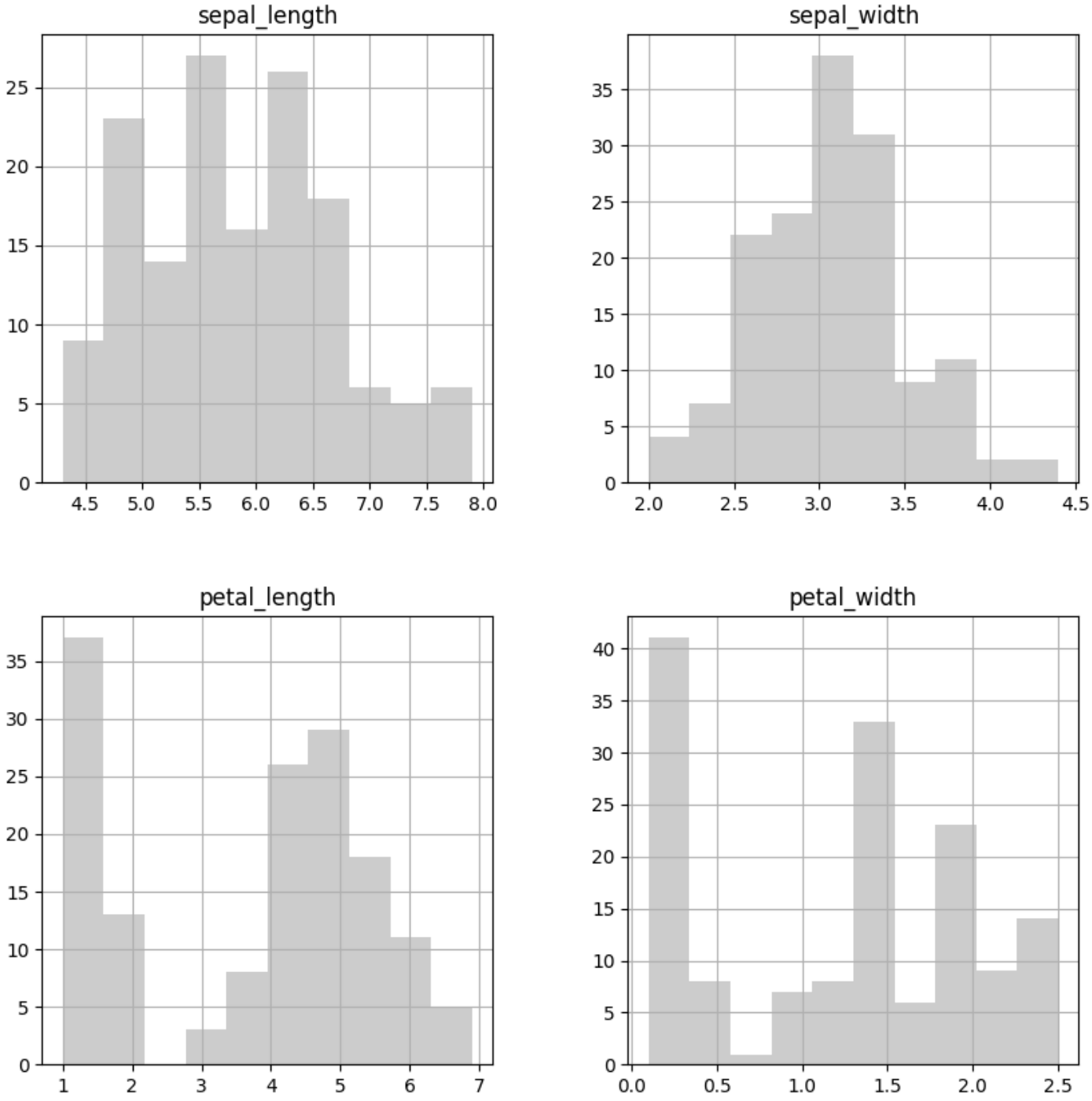
In [11]:
```
#using direct hist() function to plot instead of "kind" parameter in plot
df.hist(figsize=(10,10))
```

Out[11]:
```
array([[<AxesSubplot:title={'center':'sepal_length'}>,
        <AxesSubplot:title={'center':'sepal_width'}>],
       [<AxesSubplot:title={'center':'petal_length'}>,
        <AxesSubplot:title={'center':'petal_width'}>]], dtype=object)
```



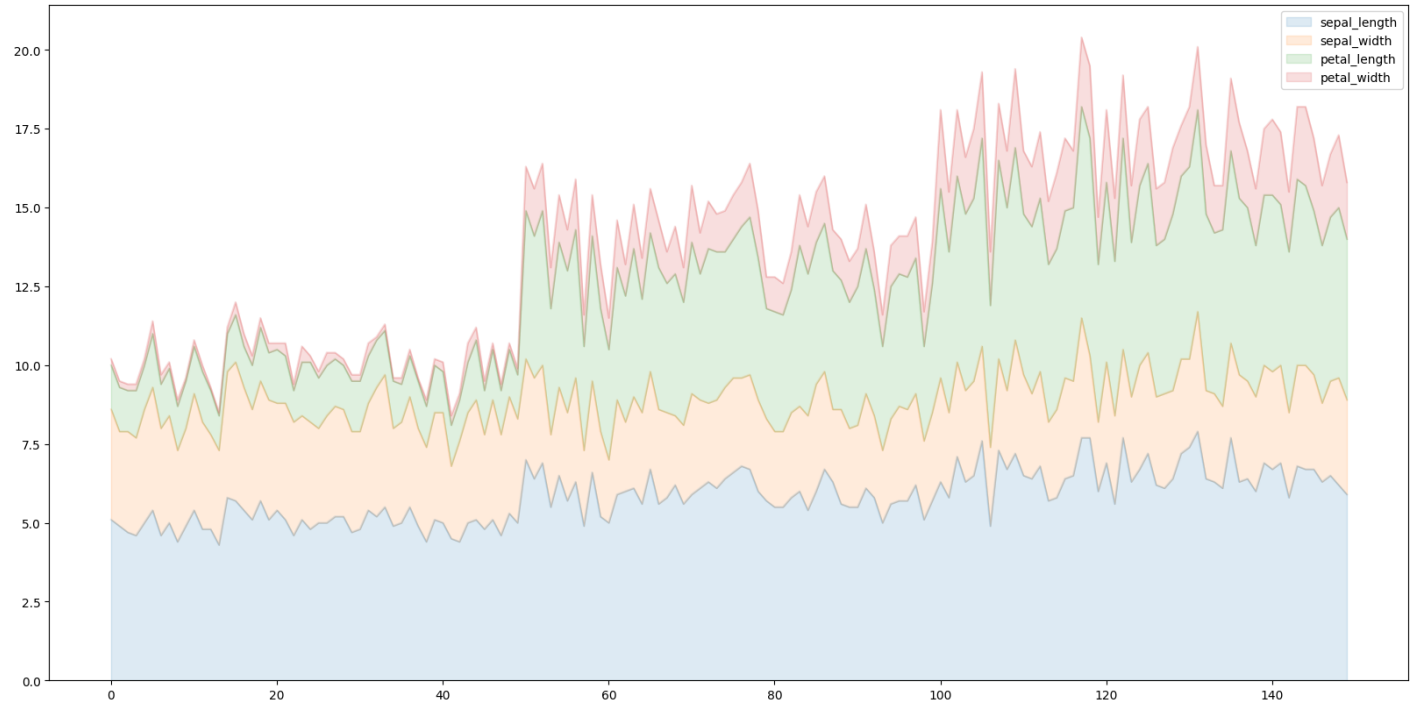
In [12]:
```
#color parameter- to set color of graph
#alpha parameter- to set density or density of color

df.hist(figsize=(10,10),color='black',alpha=.20)
```

Out[12]:
```
array([[<AxesSubplot:title={'center':'sepal_length'}>,
        <AxesSubplot:title={'center':'sepal_width'}>],
       [<AxesSubplot:title={'center':'petal_length'}>,
        <AxesSubplot:title={'center':'petal_width'}>]], dtype=object)
```



In [13]:
```
#area plot
#same like line plot
df.plot(kind='area',figsize=(20,10),alpha=.15)
```

Out[13]:
```
<AxesSubplot:>
```



In [14]:
```
df
```

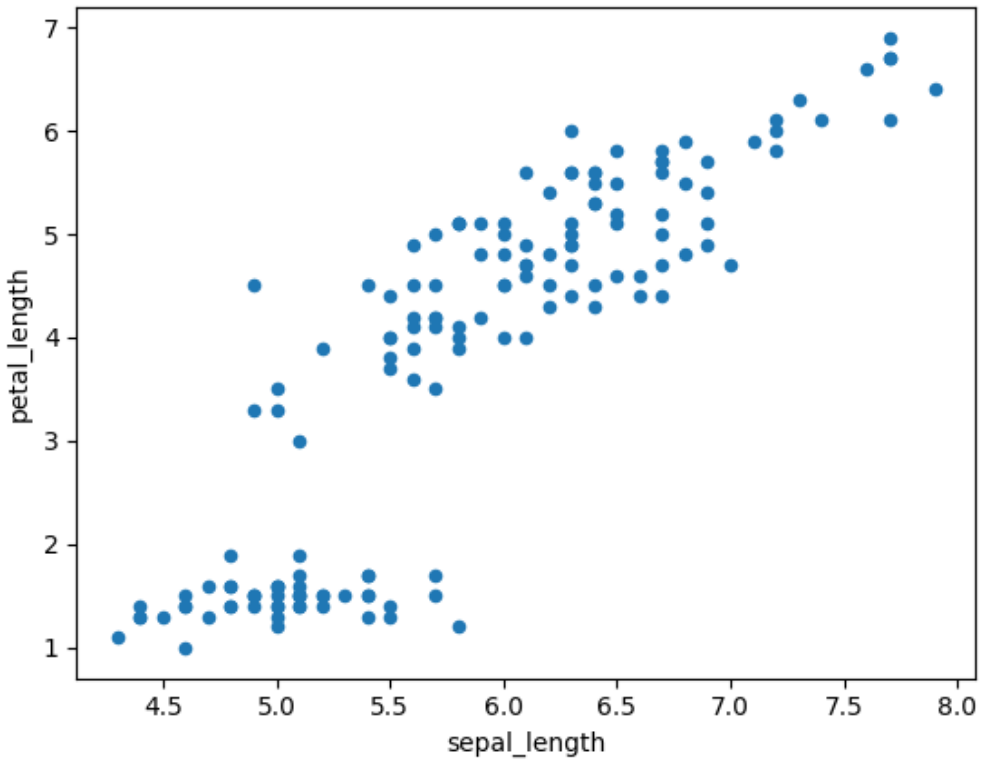
Out[14]:

	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

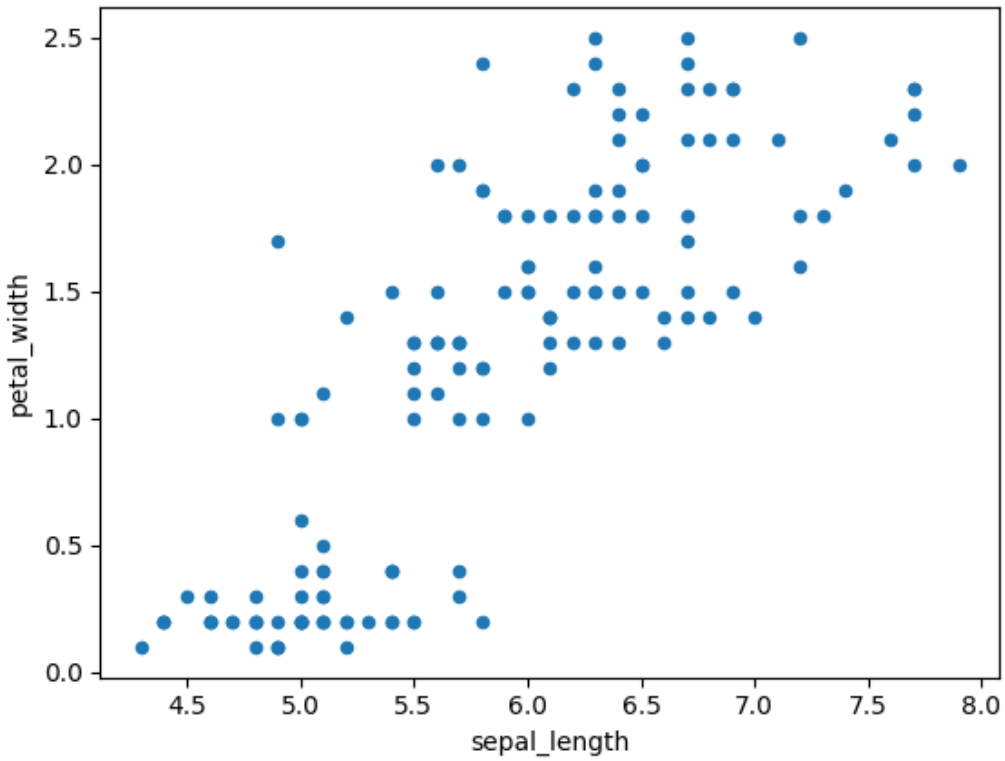
In [15]:
```
#scatter plot
#we use scatter plot to find the relation ship between two variables
#in this graph the relationship between 'sepal_length' and 'petal_length' is strong
df.plot(kind='scatter',x='sepal_length',y='petal_length')
```

Out[15]:
```
<AxesSubplot:xlabel='sepal_length', ylabel='petal_length'>
```



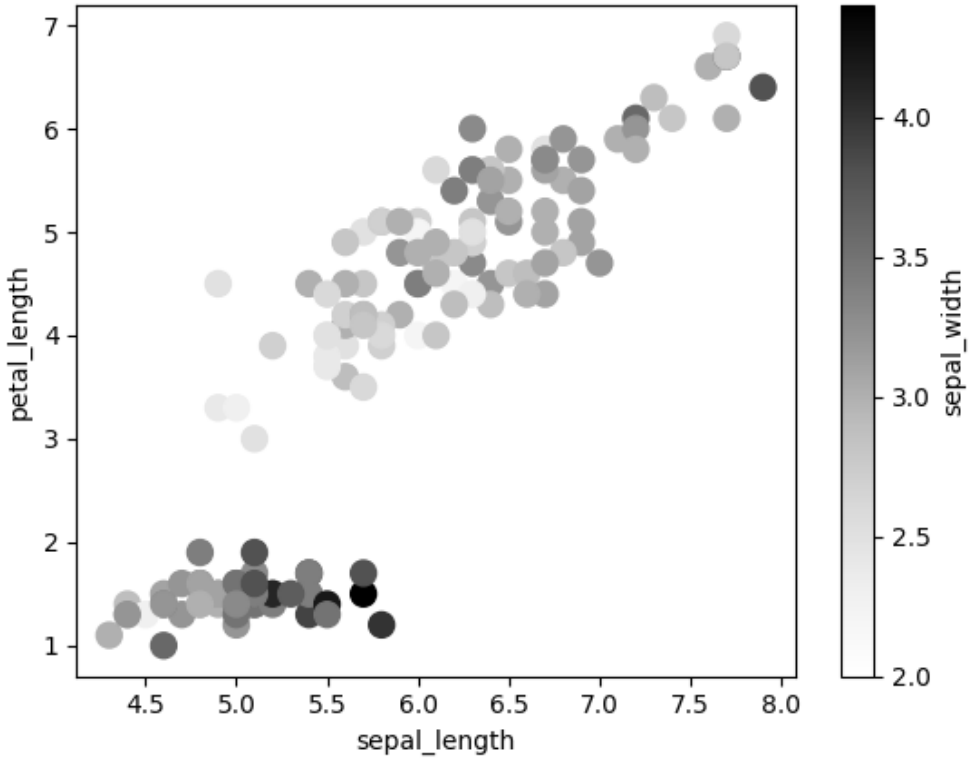
In [16]:
```
#in this graph the graph is more scattered i.e 'sepal_length' and 'petal_width' relationshiop is nc
df.plot(kind='scatter',x='sepal_length',y='petal_width')
```

Out[16]:
```
<AxesSubplot:xlabel='sepal_length', ylabel='petal_width'>
```



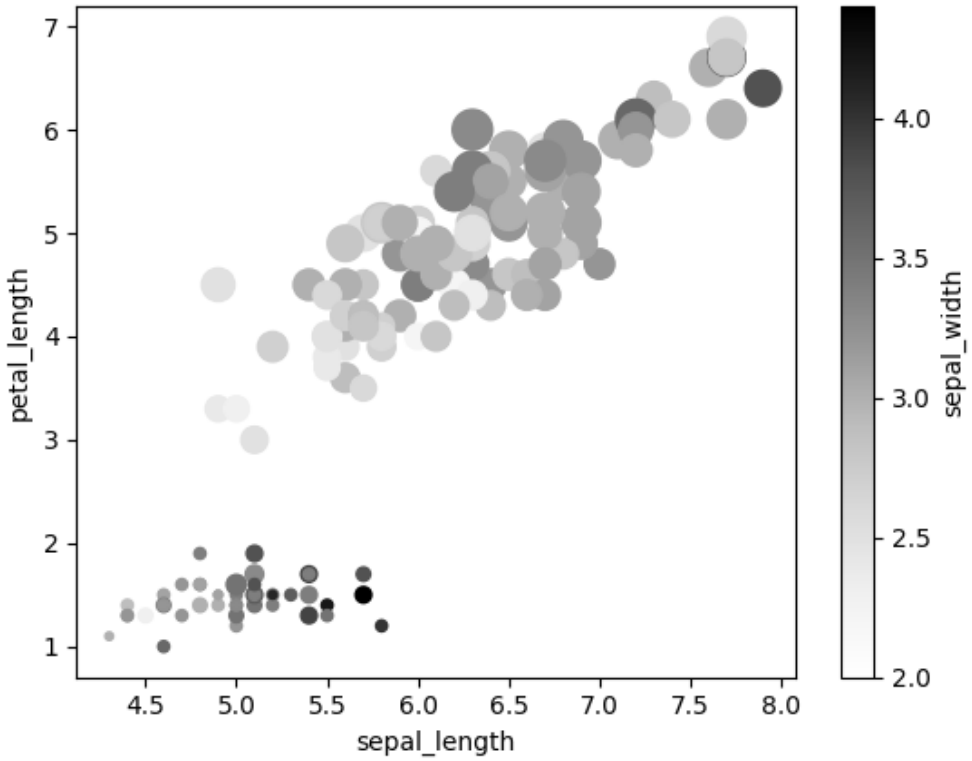
In [17]:
```
#scatterplot
#Here we are using c(color) parameter to represent another variable based on range of that column t
#s-paramter--increase the size of bubble

df.plot(kind='scatter',x='sepal_length',y='petal_length',c='sepal_width',s =100)
```

Out[17]:
```
<AxesSubplot:xlabel='sepal_length', ylabel='petal_length'>
```



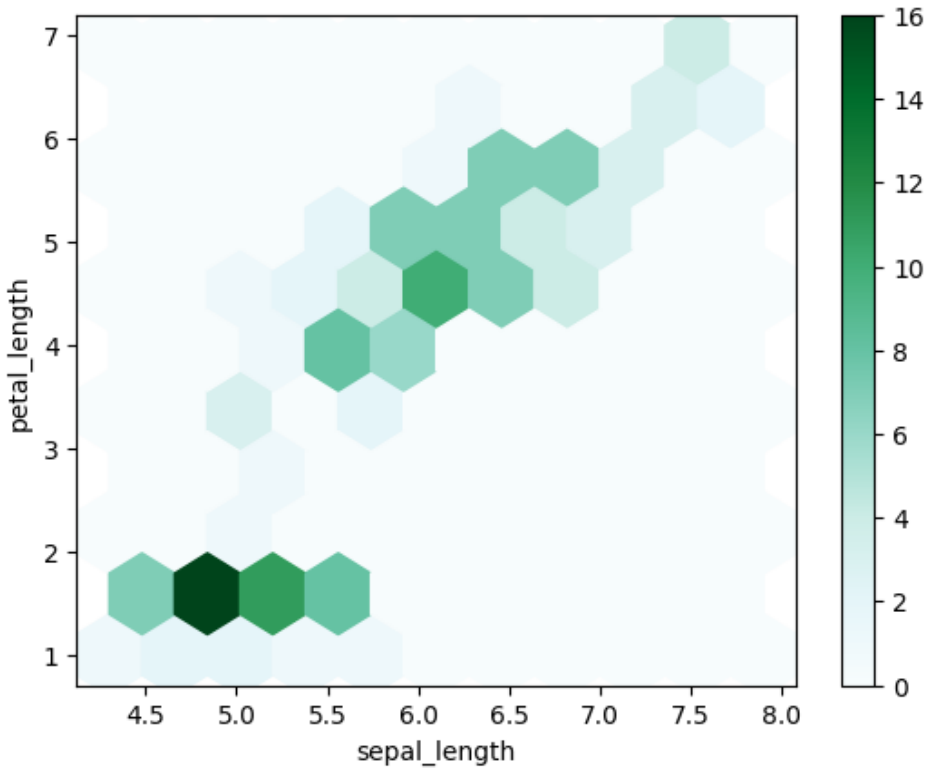
In [18]:
```
#scatterplot
#Here we are using c(color) parameter to represent another variable based on range of that column t
#s-paramter--or we can represent size based on particular variable i.e here we are using 'petal_wid

df.plot(kind='scatter',x='sepal_length',y='petal_length',c='sepal_width',s =df['petal_width']*100)
```

Out[18]:
```
<AxesSubplot:xlabel='sepal_length', ylabel='petal_length'>
```



In [19]:
```
#same as scatter plot but difference is style of representing i.e bubble into hexa shape
#gridsize parameter---is used to set the size of hexa shape.
df.plot(kind='hexbin',x='sepal_length',y='petal_length',gridsize=10)
```

Out[19]:
```
<AxesSubplot:xlabel='sepal_length', ylabel='petal_length'>
```



In [20]:
```
df
```

Out[20]:

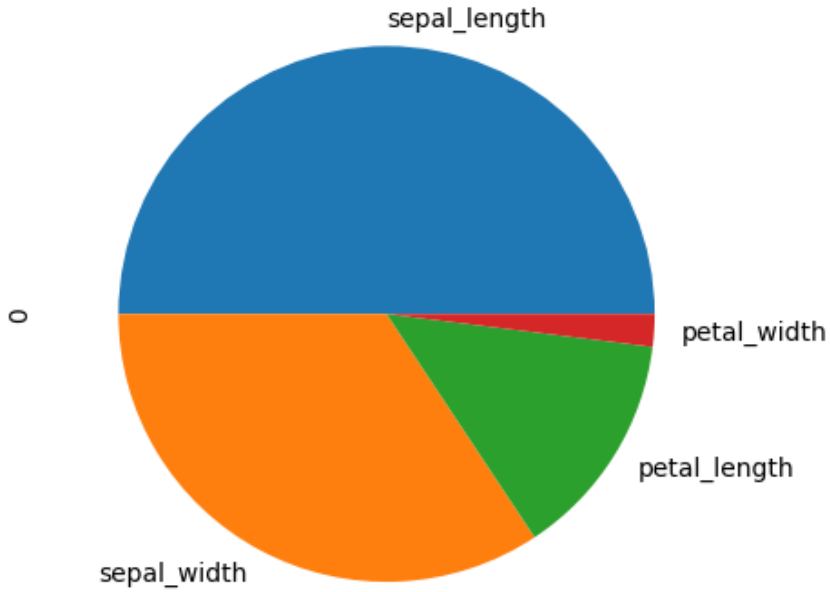
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

In [21]:
```
df1=df.drop(columns='species')
```

In [22]:
```
#piplot
df1.iloc[0].plot(kind='pie')
```

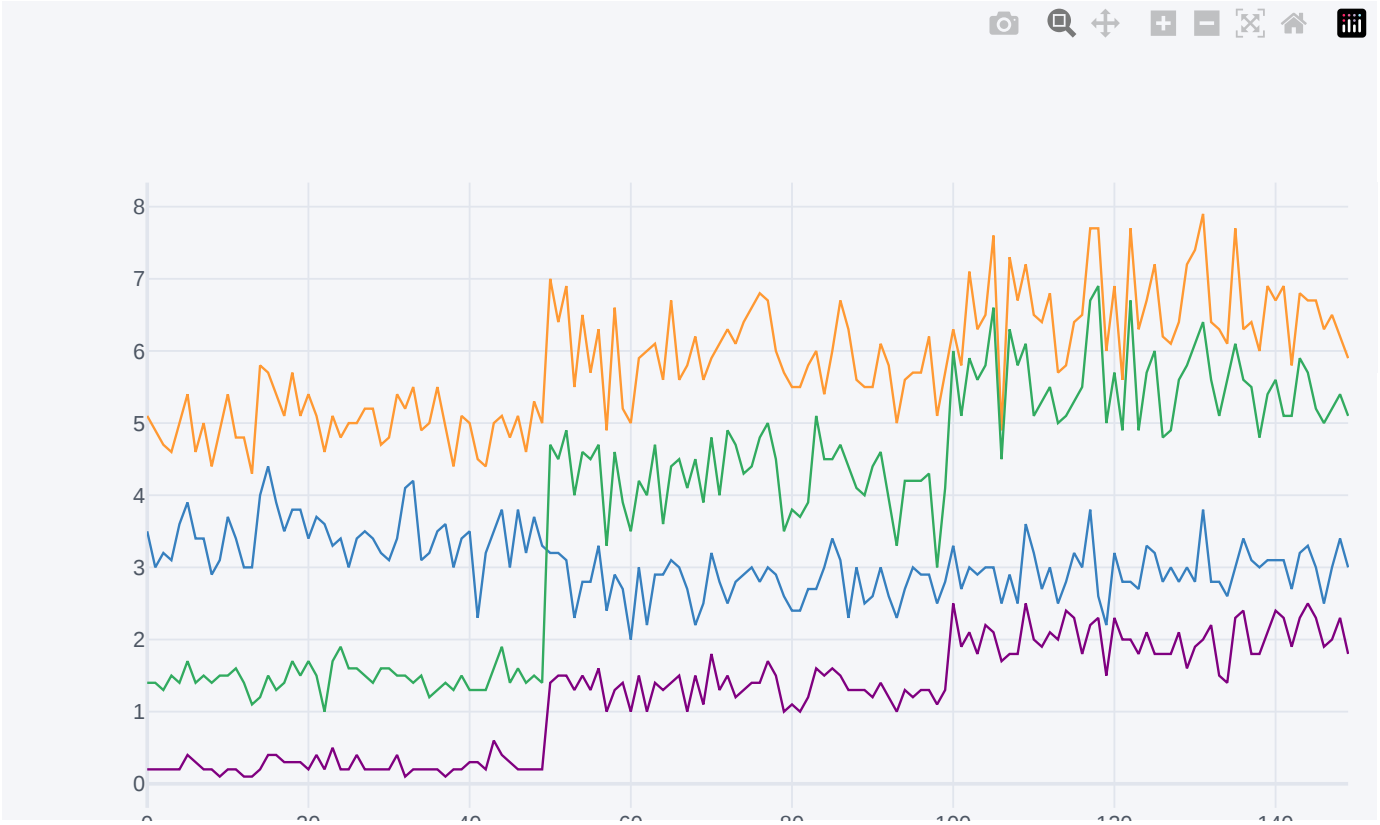
Out[22]:
```
<AxesSubplot:ylabel='0'>
```



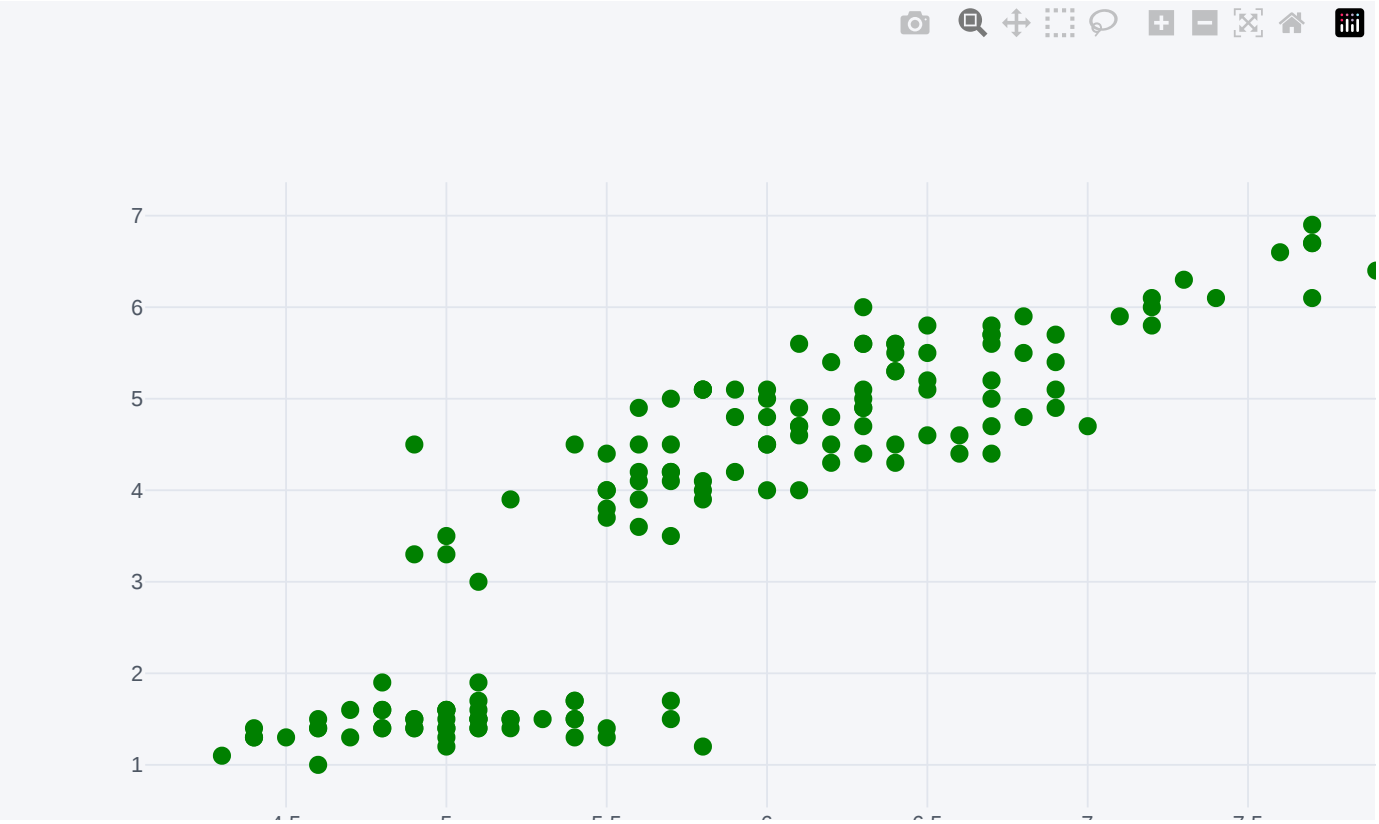
Graphs using Plotly

In [38]:
```
import cufflinks as cf

#Since it takes time to represent the graph we use the below method to make it fast representartion
cf.go_offline()
```

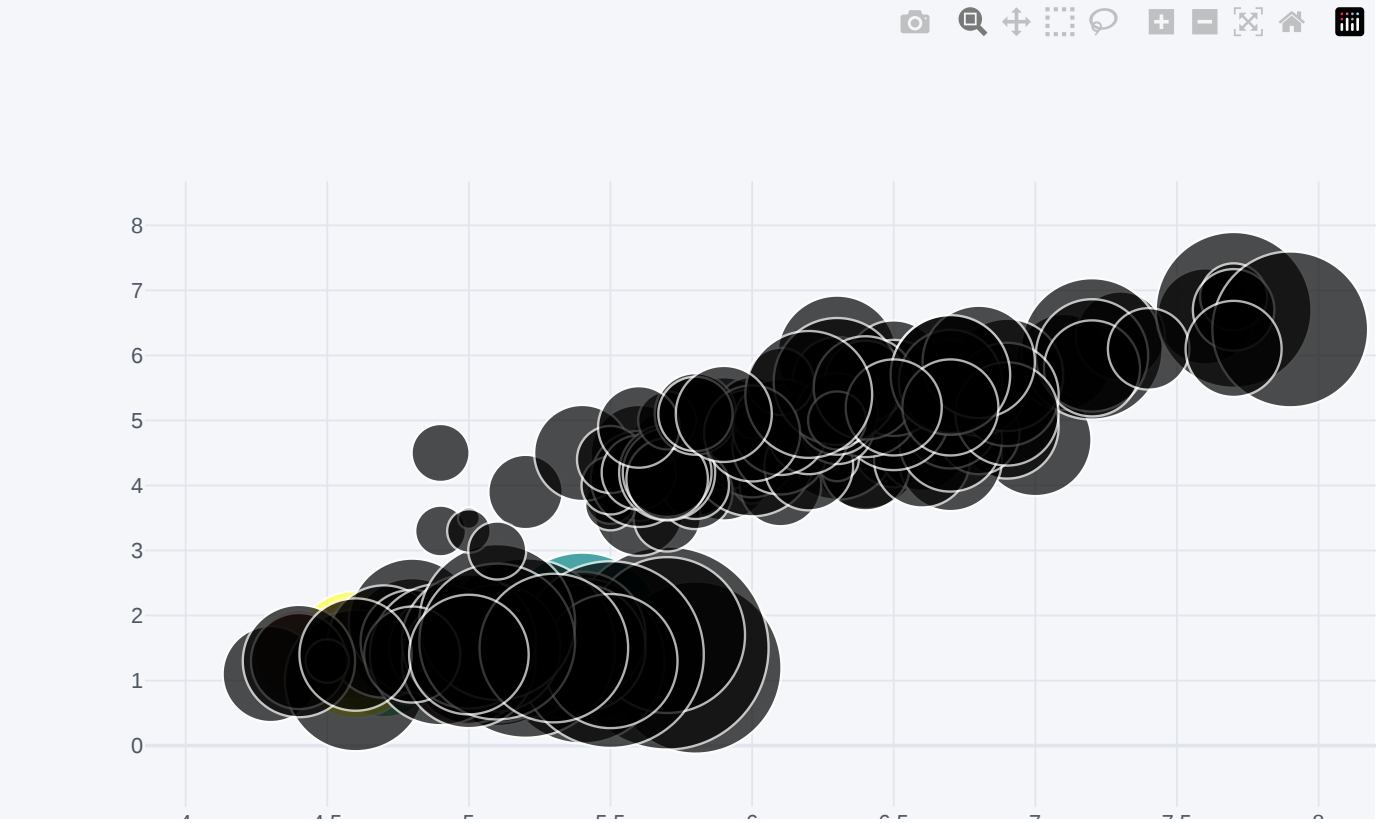
In [39]:
```
df.iplot()
```



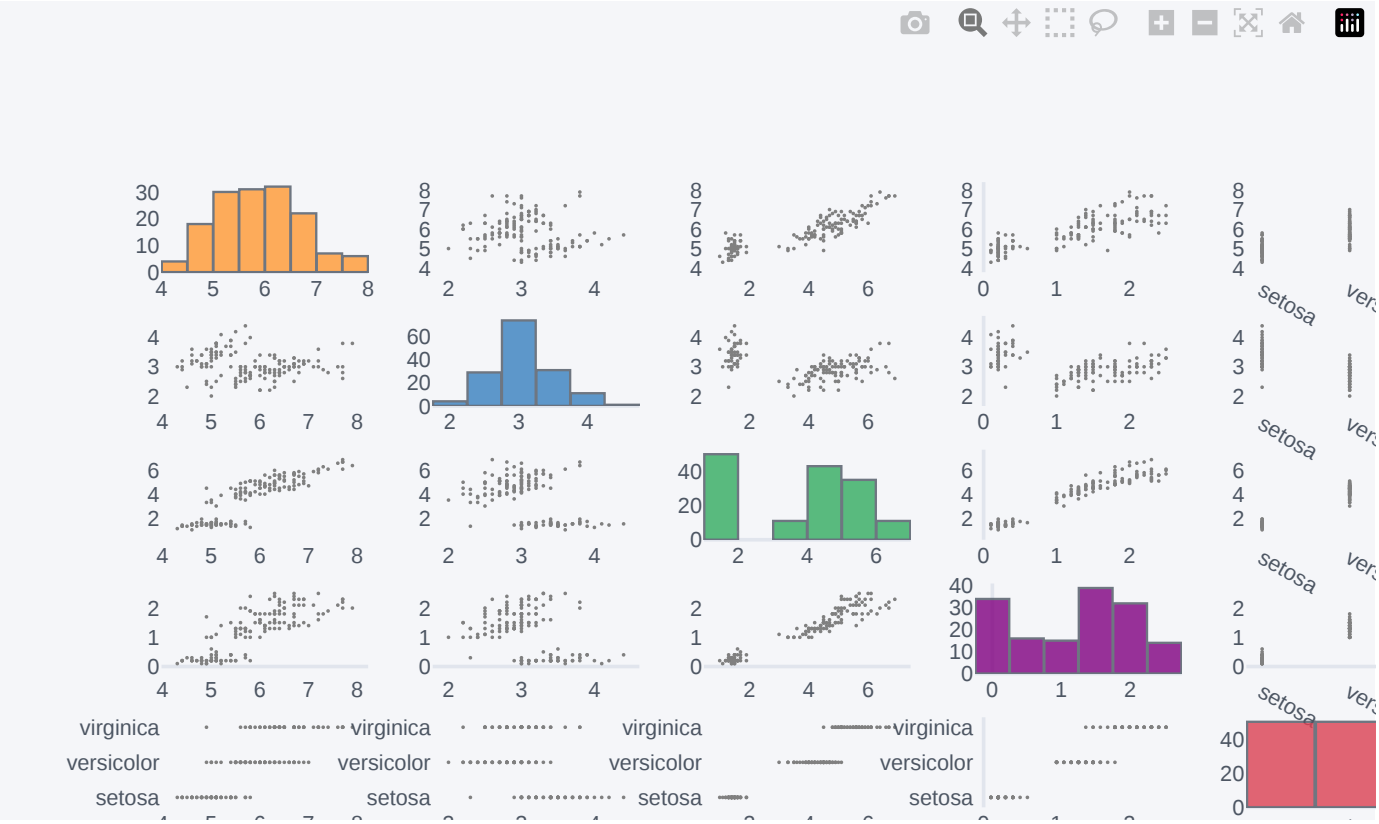
In [40]:
```
#scatter plot
#in plotly "kind" parameter wont give scatterplot itg goives scatter graph with lines
#so we use "mode" parameter to represent scatter plot in circles.
df.iplot(x='sepal_length',y='petal_length',kind='scatter',mode='markers',color='green',size=10)
```



In [41]:
```
#bubble plot
df.iplot(kind='bubble',x='sepal_length',y='petal_length',size='sepal_width')
```

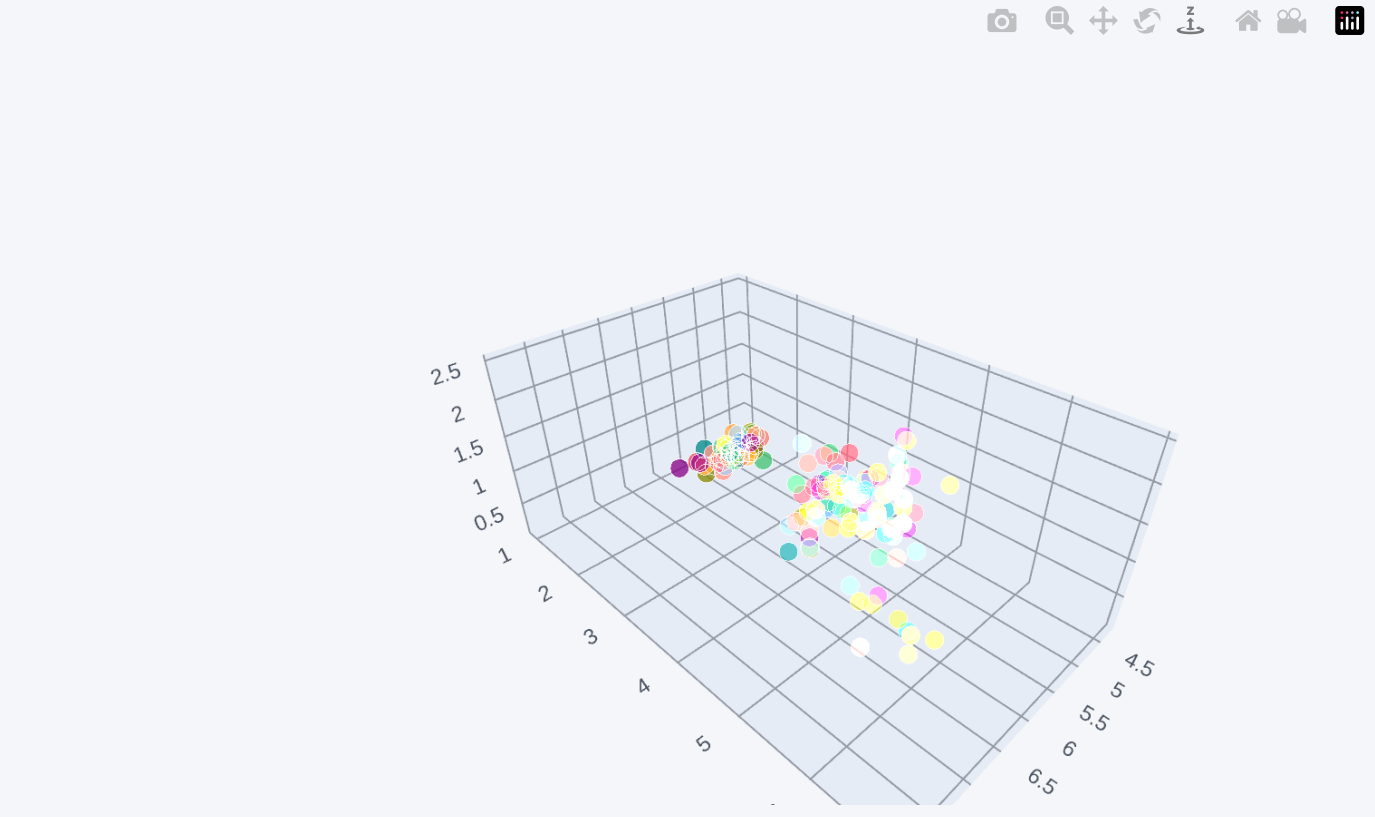


In [42]:
```
#scatter_matrix

df.scatter_matrix()
```

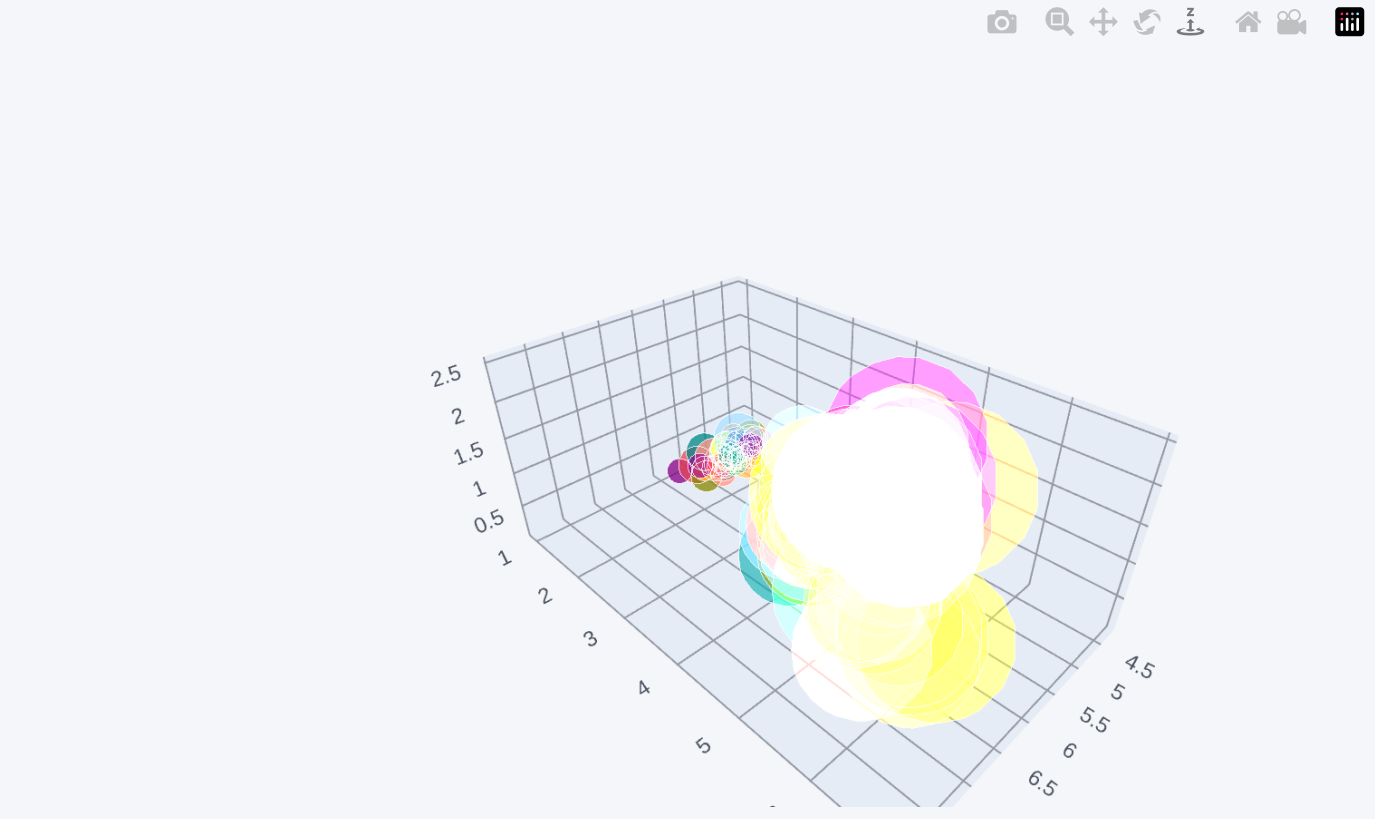


3D plot in plotly

In [43]:
```
'''
in below grpah the tooltip values are
x-given value in parameter
y-given value in parameter
z-given value in parameter
and last value represents index number in df
'''
df.iplot(kind='scatter3d',x='sepal_length',y='petal_length',z='petal_width')
```



In [44]:
```
#bubble3d graph
#in bubble 3d graph you have to add parameter "size" along with the three axis parameter for sure t
#or it will throw error.

df.iplot(kind='bubble3d',x='sepal_length',y='petal_length',z='petal_width',size='petal_width')
```



Seaborn topic notes

1.can represent data by group using **hue(color)** and **style(shape)** parameter

2.Can plot seperate graphs at a time based on column values using **col** parameter

3.can represent graph based on catagory using **catplot** function

In [1]:
```
import seaborn as sns
```

In [2]:
```
df=sns.load_dataset('iris')
```

In [3]:
```
df
```

Out[3]:

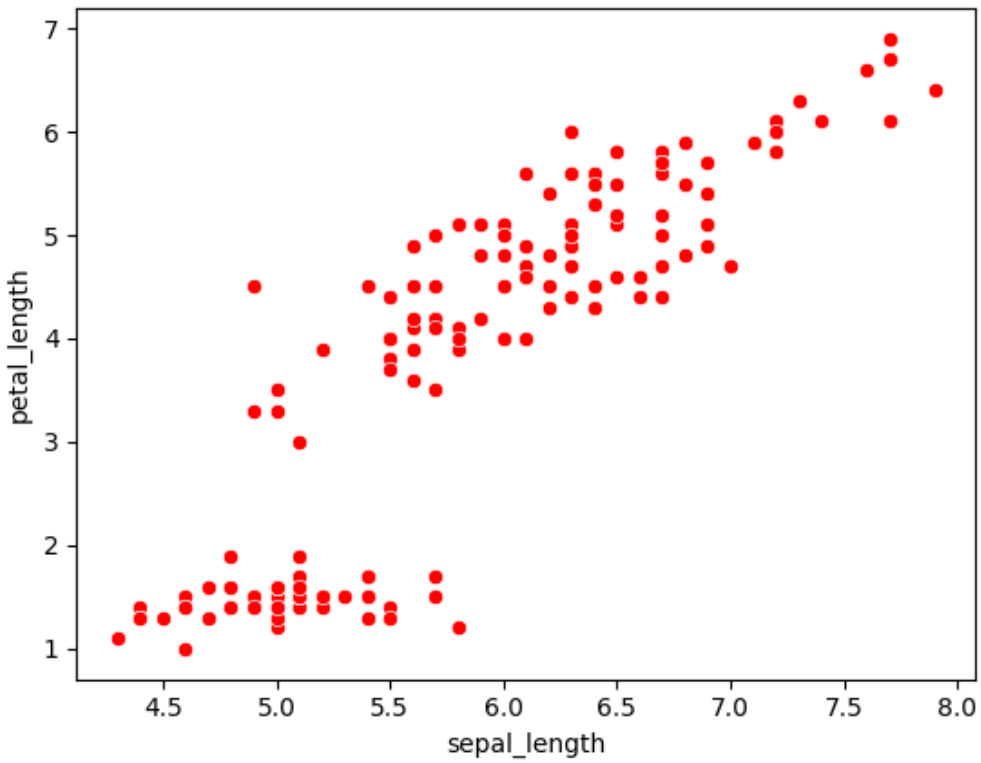
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

In [4]:
```
#scatter plot using seaborn

sns.scatterplot(df.sepal_length,df.petal_length,color='red')
```

```
C:\Users\seruvuri\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\_decorators.p
y:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the on
ly valid positional argument will be `data`, and passing other arguments without an explicit keywo
rd will result in an error or misinterpretation.
  warnings.warn(
```

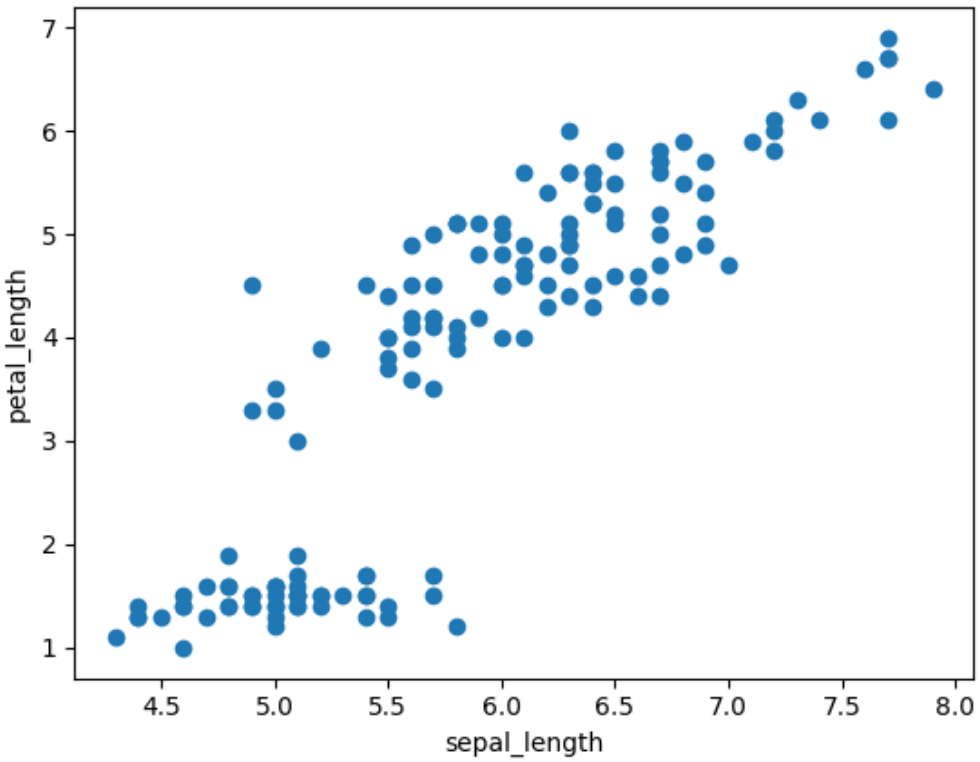
Out[4]: <AxesSubplot:xlabel='sepal_length', ylabel='petal_length'>



In [5]:
```
import matplotlib.pyplot as plt
```

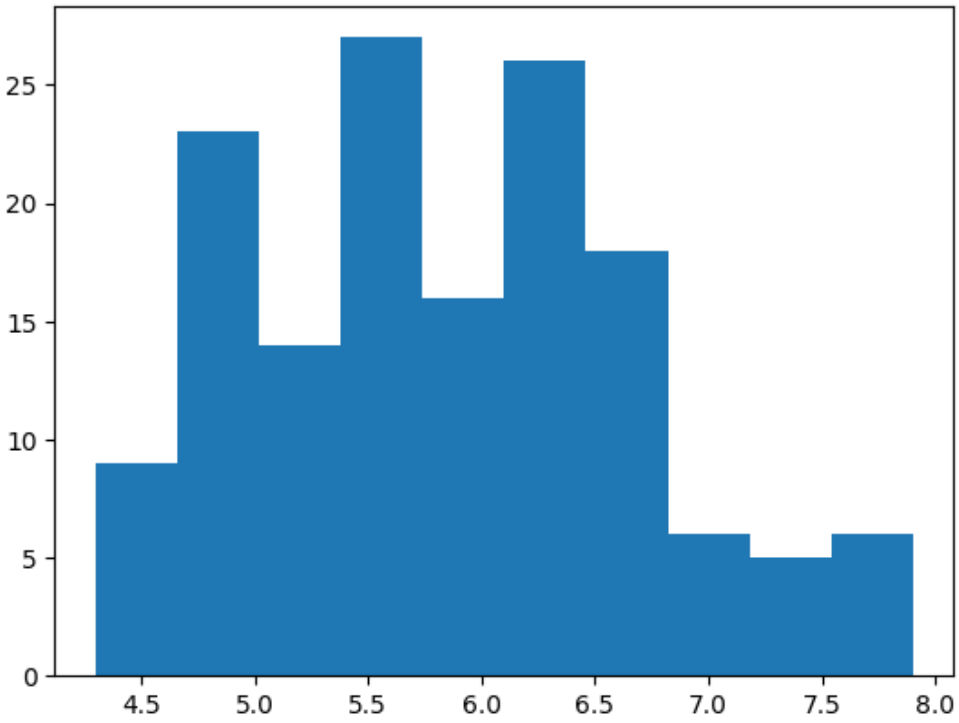
In [6]:
```
#scatter plot using matplotlib
plt.plot(df.sepal_length,df.petal_length,'o')
plt.xlabel("sepal_length")
plt.ylabel("petal_length")
```

Out[6]: Text(0, 0.5, 'petal_length')



In [7]:
```
#histogram plot using matplotlib
plt.hist(df['sepal_length'])
```

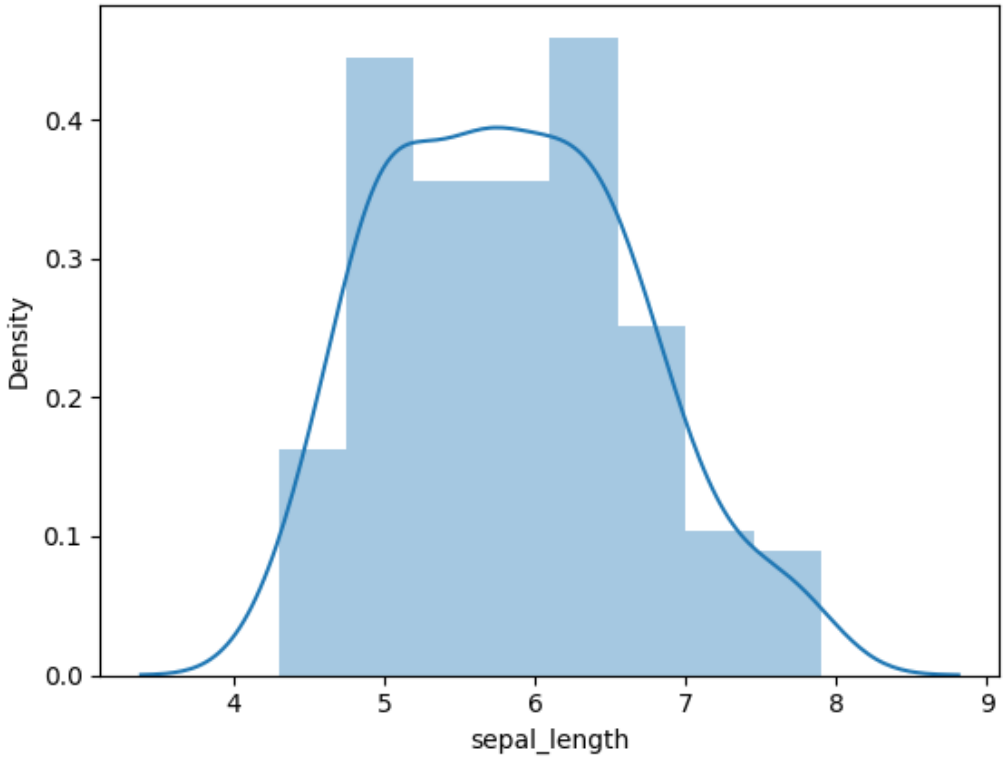
Out[7]:
```
(array([ 9., 23., 14., 27., 16., 26., 18.,  6.,  5.,  6.]),
 array([4.3 , 4.66, 5.02, 5.38, 5.74, 6.1 , 6.46, 6.82, 7.18, 7.54, 7.9 ]),
 <BarContainer object of 10 artists>)
```



In [8]:
```
#histogram using seaborn
#in seaborn we use "distplot" function yo plot histogram
sns.distplot(df['sepal_length'])
```

```
C:\Users\seruvuri\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\distributions.
py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future versio
n. Please adapt your code to use either `displot` (a figure-level function with similar flexibilit
y) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[8]: <AxesSubplot:xlabel='sepal_length', ylabel='Density'>



In [9]:
```
df1=sns.load_dataset('tips')
```

In [10]:
```
df1
```

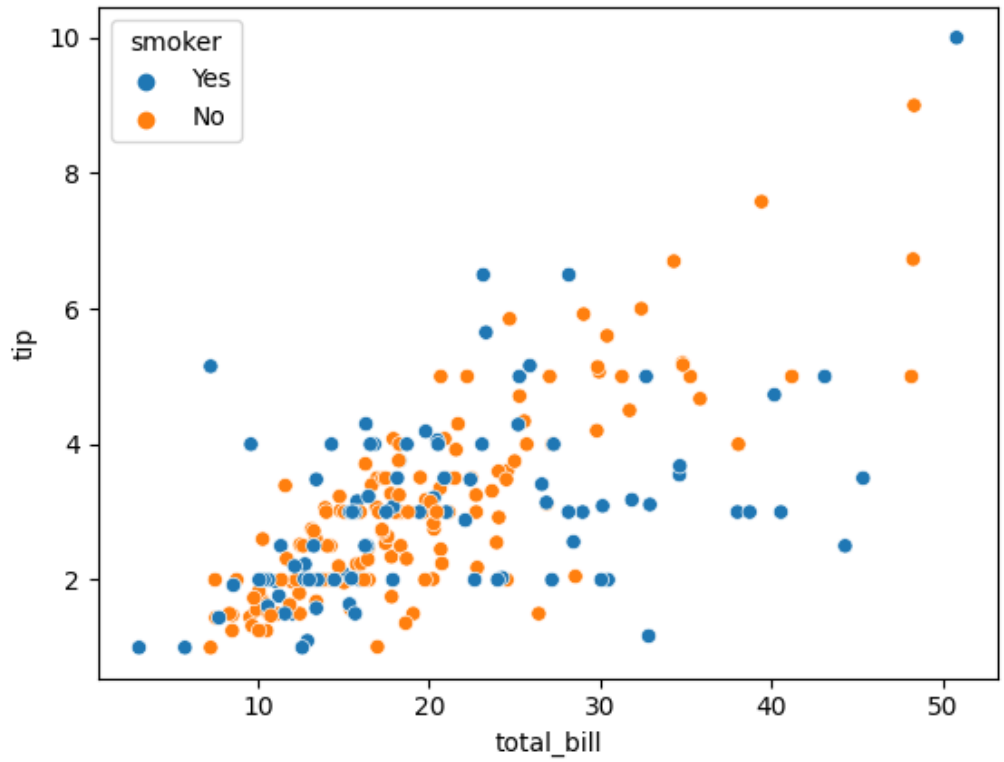
Out[10]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

244 rows × 7 columns

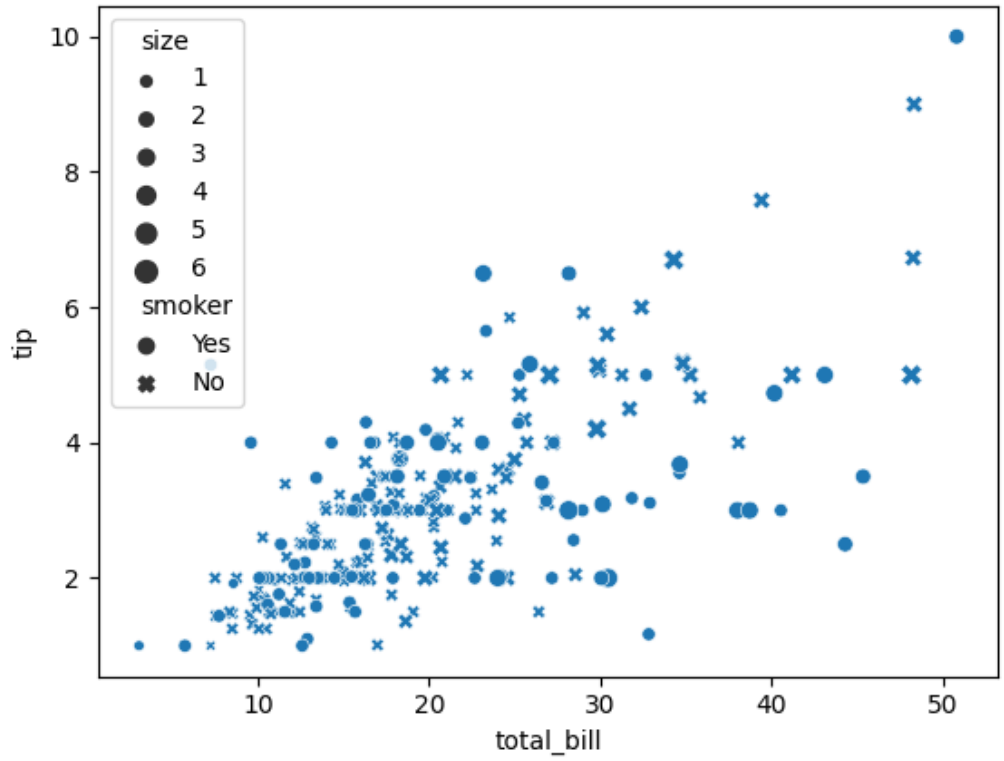
In [11]:
```
#hue parameter----- will group the data based on the variable passed.


sns.scatterplot(x='total_bill',y='tip',hue='smoker',data=df1)
```

Out[11]: <AxesSubplot:xlabel='total_bill', ylabel='tip'>



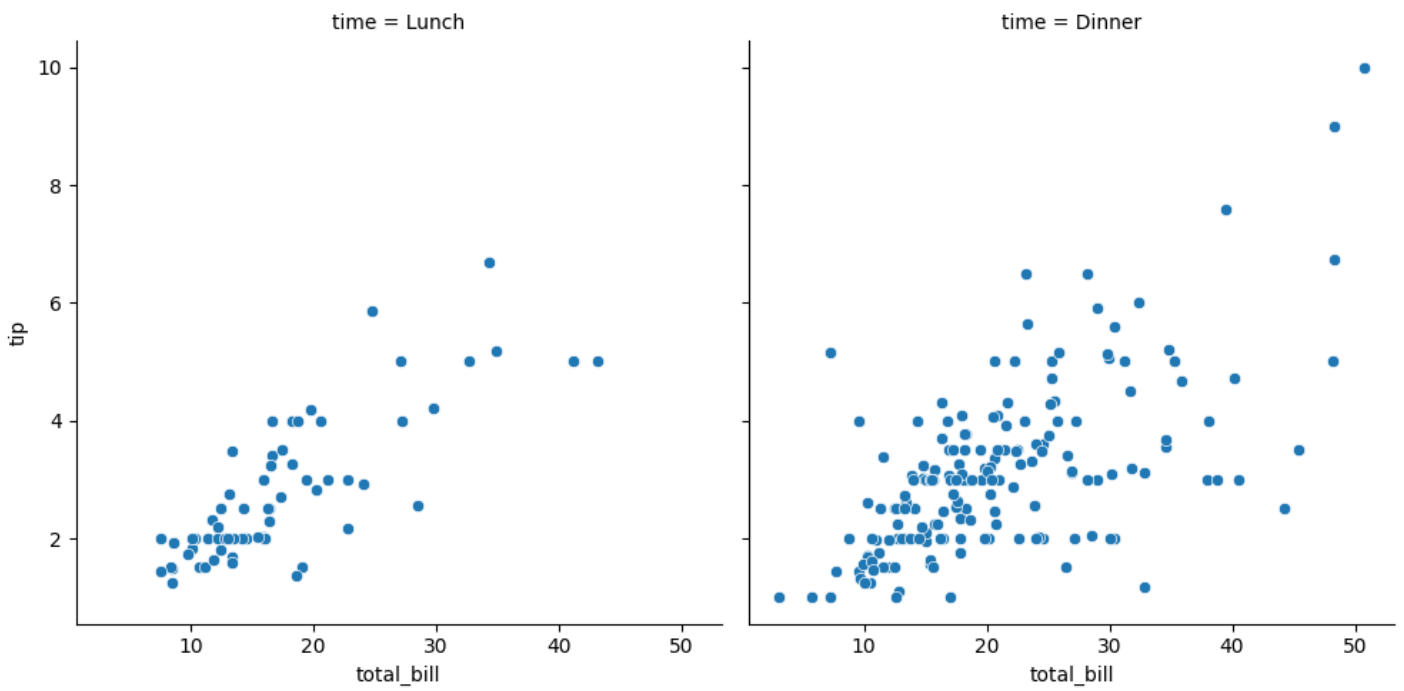
In [12]:
```
#Style parameter ------will group the data based on the variable passed in different shape for each

sns.scatterplot(x='total_bill',y='tip',style='smoker',data=df1,size='size')
```

Out[12]:
```
<AxesSubplot:xlabel='total_bill', ylabel='tip'>
```



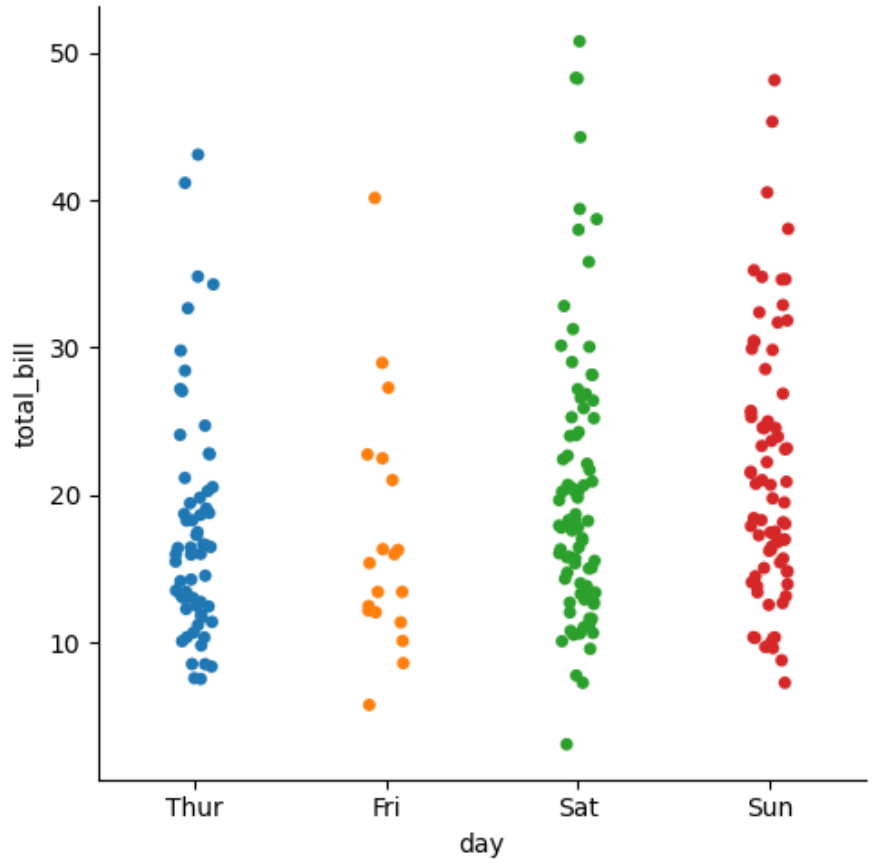
In [13]:
```
#col paramter-----plotting seperate graphs based on column value

sns.relplot(x='total_bill',y='tip',col='time',data=df1)
```

Out[13]:
```
<seaborn.axisgrid.FacetGrid at 0x1f42b31e4d0>
```



In [14]:
```
#catogary plot in seaborn

sns.catplot(x='day',y='total_bill',data=df1)
```

Out[14]:
```
<seaborn.axisgrid.FacetGrid at 0x1f42b3a6080>
```



In [ ]: