

Graphs

Thota Naveen Babu

Follow me on LinkedIn :- <https://www.linkedin.com/in/naveen-babu-thota-45894b243/>
(<https://www.linkedin.com/in/naveen-babu-thota-45894b243/>)

Follow me on GitHub :- <https://github.com/naveen23049?tab=repositories>
(<https://github.com/naveen23049?tab=repositories>)

```
In [ ]: import cufflinks as cf
import numpy as np
import pandas as pd
import seaborn as sns
```

```
In [2]: cf.go_offline()
```

```
In [3]: df =pd.DataFrame(np.random.randn(100,3) , columns=["A" , "B" , "C"])
```

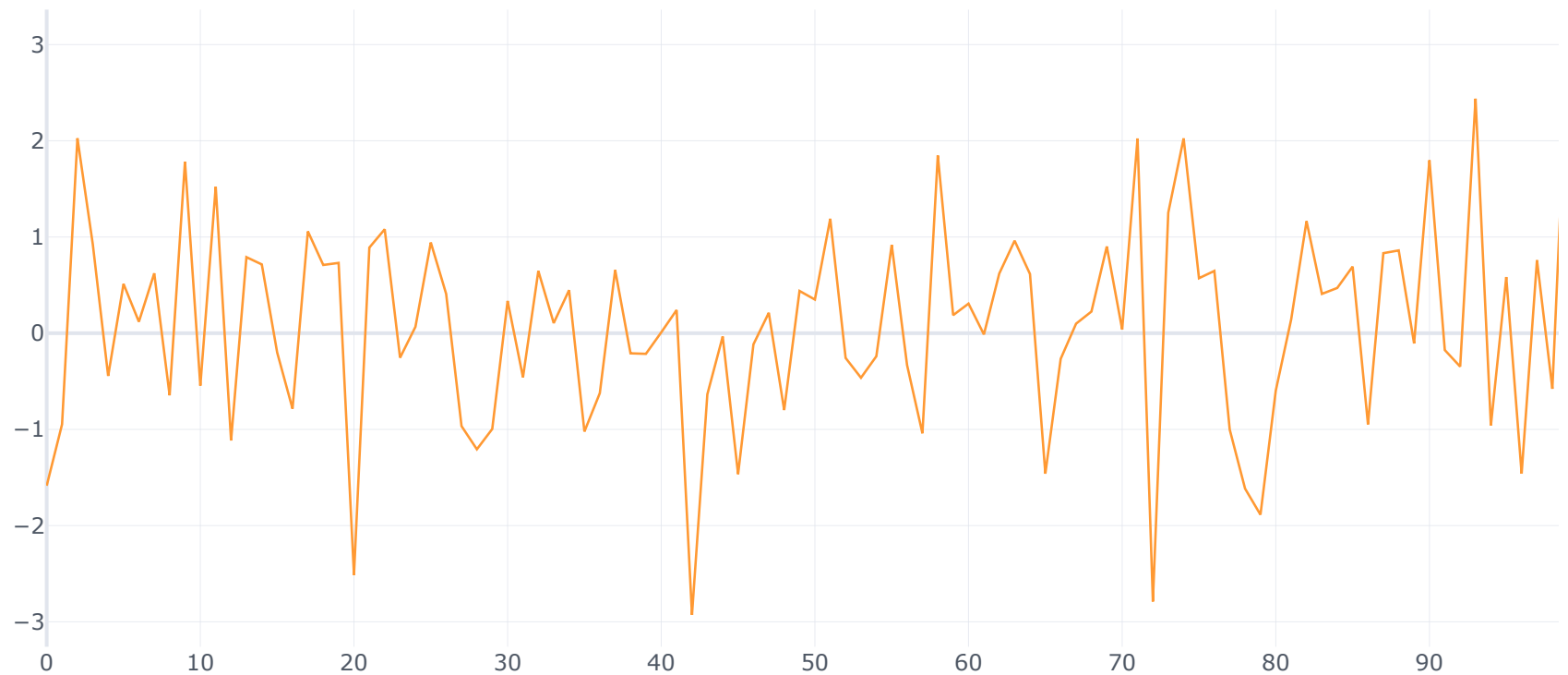
In [4]: df

Out[4]:

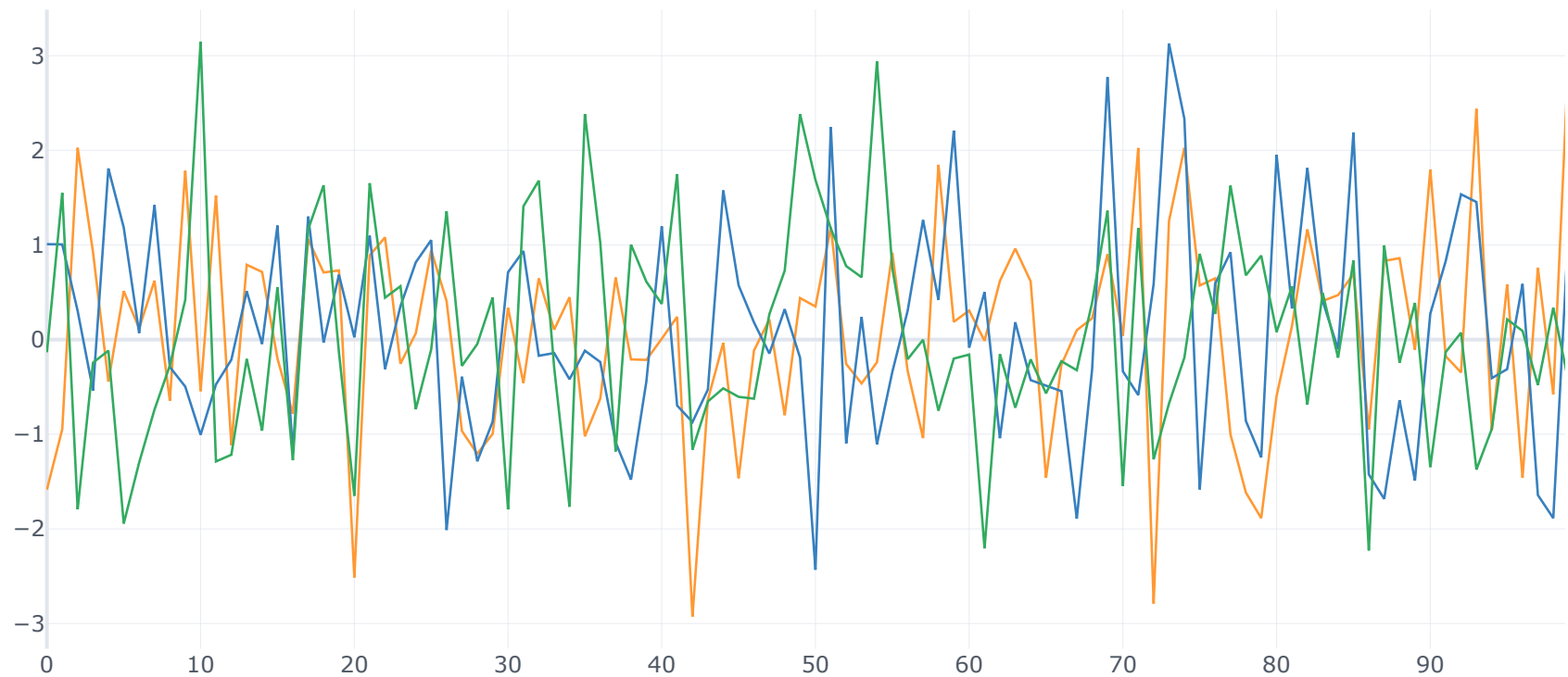
	A	B	C
0	-1.584950	1.008447	-0.134143
1	-0.947177	1.006018	1.550452
2	2.026504	0.302720	-1.792775
3	0.924137	-0.538951	-0.244098
4	-0.444107	1.808399	-0.117070
...
95	0.583358	-0.311901	0.214027
96	-1.459669	0.588080	0.091587
97	0.759547	-1.643115	-0.480081
98	-0.578149	-1.886409	0.336736
99	3.032333	1.193231	-0.448493

100 rows × 3 columns

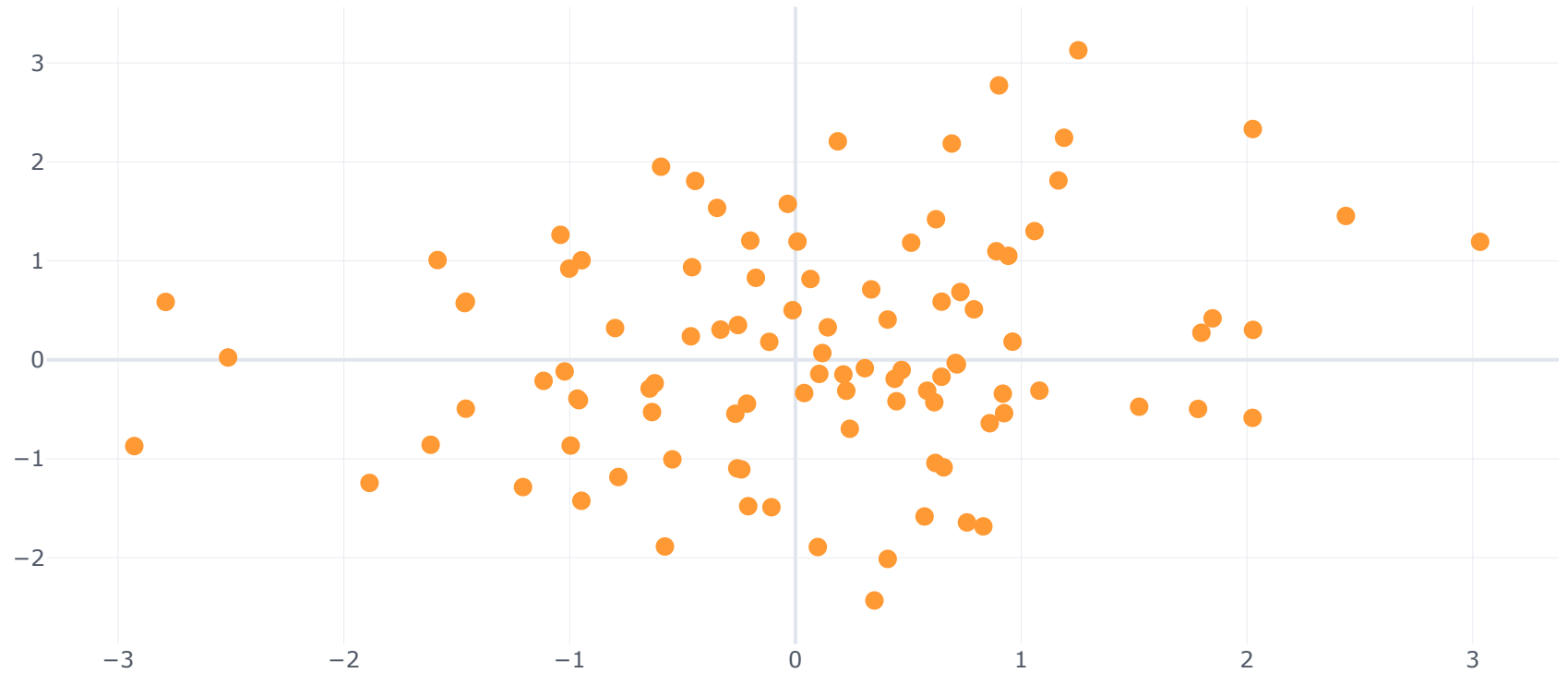
```
In [5]: df[["A"]].iplot()
```



```
In [6]: df.iplot()
```



```
In [7]: df.iplot( x = "A", y = "B", mode = "markers", size = 10)
```



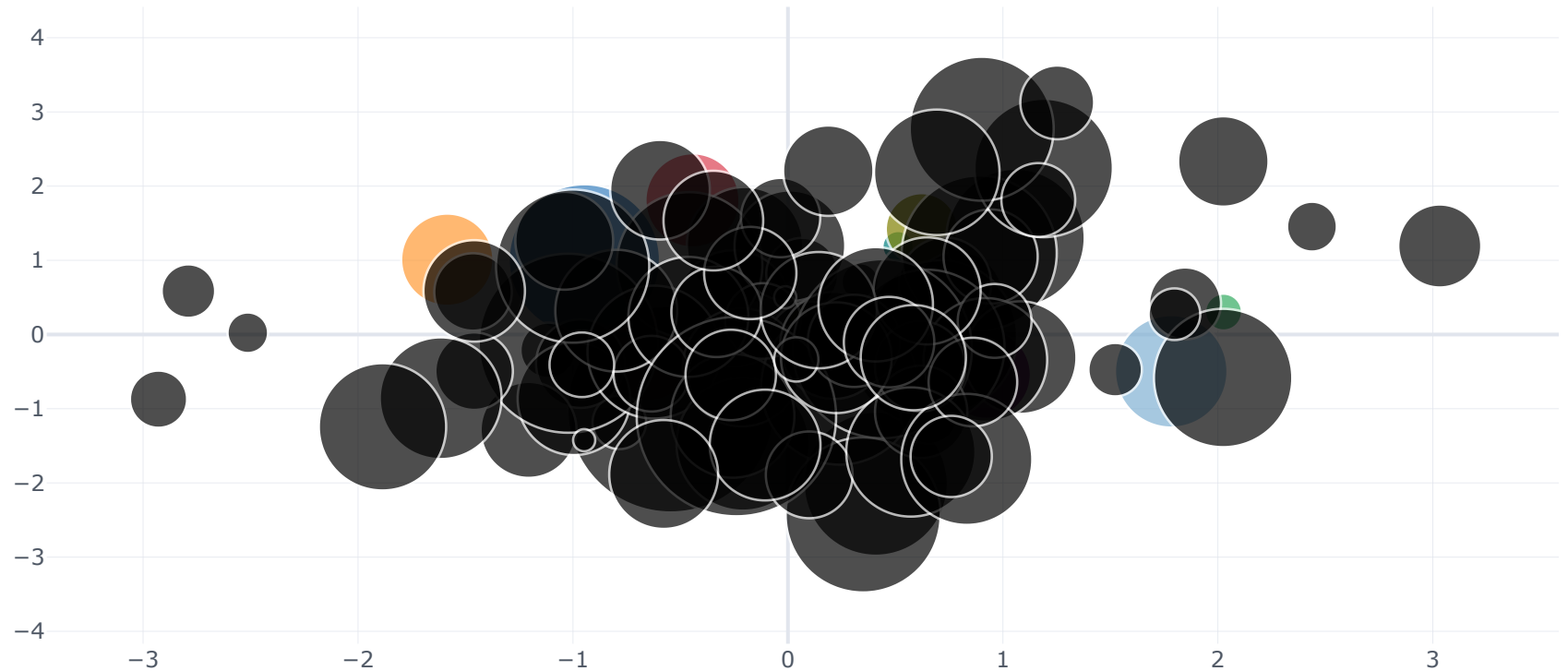
In [8]: df

Out[8]:

	A	B	C
0	-1.584950	1.008447	-0.134143
1	-0.947177	1.006018	1.550452
2	2.026504	0.302720	-1.792775
3	0.924137	-0.538951	-0.244098
4	-0.444107	1.808399	-0.117070
...
95	0.583358	-0.311901	0.214027
96	-1.459669	0.588080	0.091587
97	0.759547	-1.643115	-0.480081
98	-0.578149	-1.886409	0.336736
99	3.032333	1.193231	-0.448493

100 rows × 3 columns

```
In [9]: df.iplot(kind = 'bubble' , x = "A" , y = "B" ,size = "C")
```



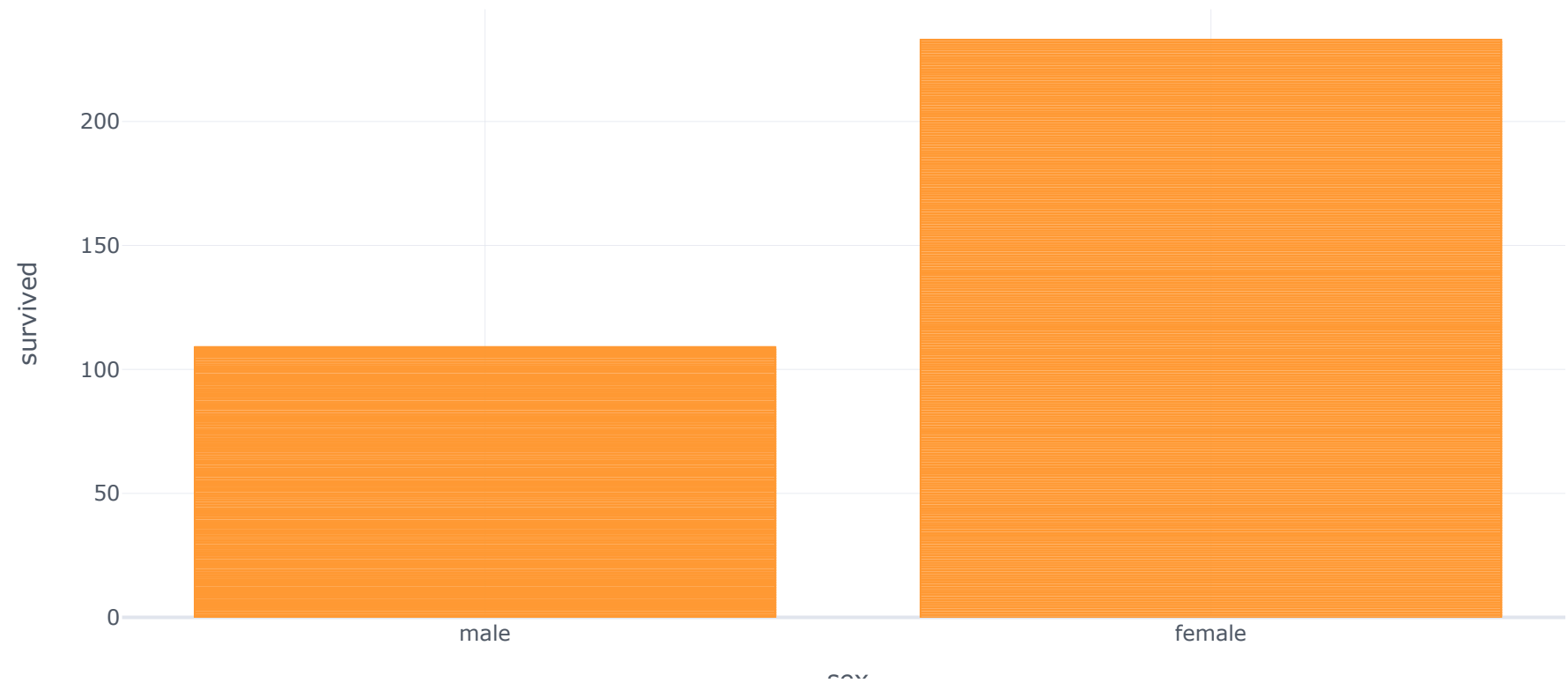
```
In [10]: df1 = sns.load_dataset("titanic")
```

```
In [11]: df1.head()
```

```
Out[11]:
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True


```
In [12]: # can you please show me the male,female vs survived column  
# simply, we can ask how many male & female are survived ?  
  
df1.iplot(x = "sex", y = "survived", kind = "bar", xTitle = "sex", yTitle = "survived")
```



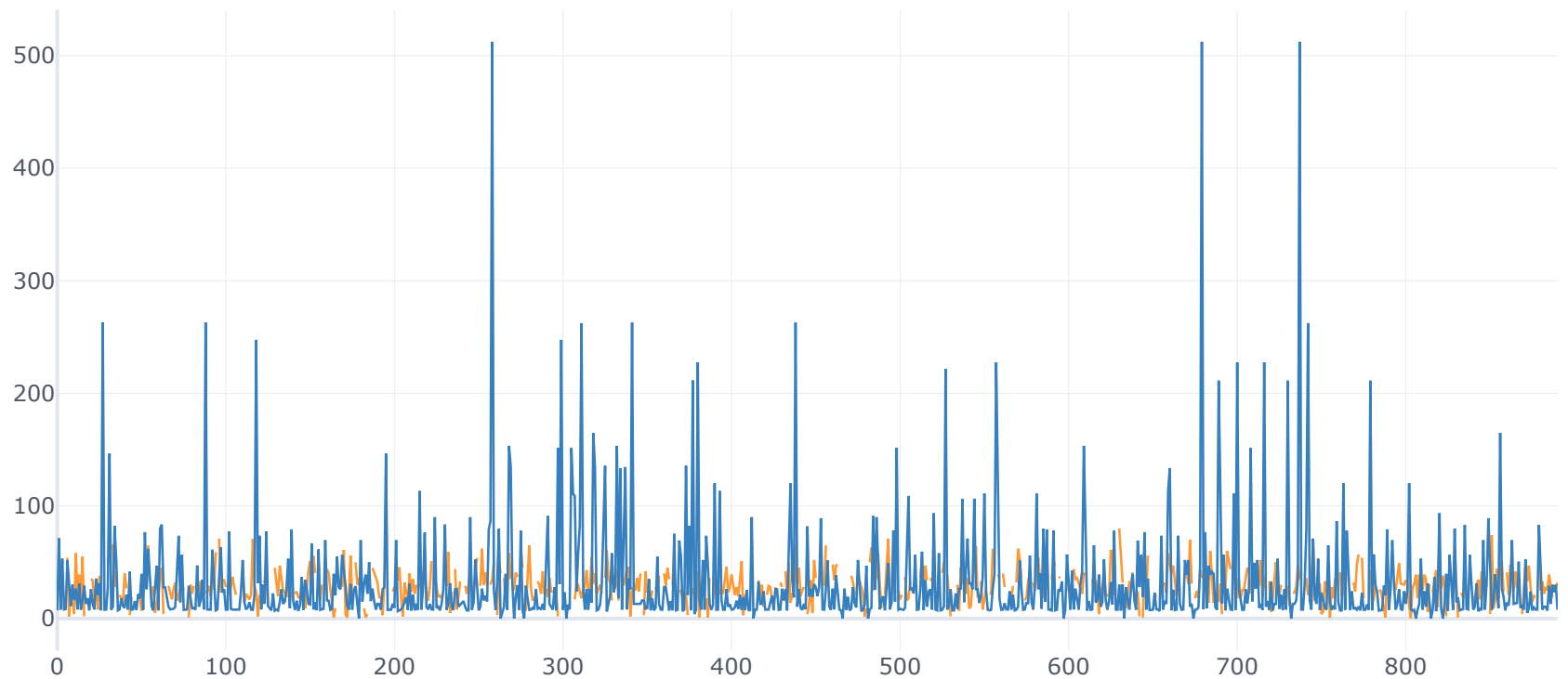
```
In [13]: df1[["age", "fare"]]
```

```
Out[13]:
```

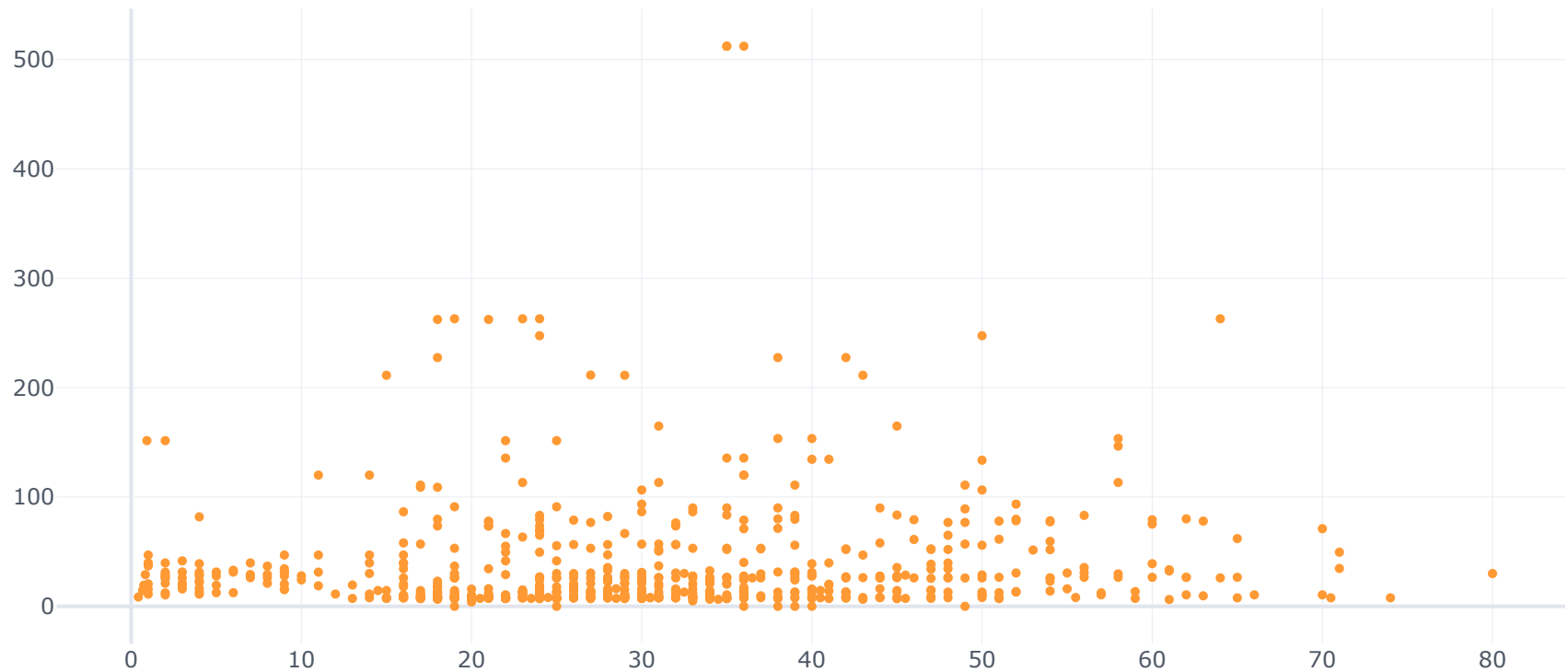
	age	fare
0	22.0	7.2500
1	38.0	71.2833
2	26.0	7.9250
3	35.0	53.1000
4	35.0	8.0500
...
886	27.0	13.0000
887	19.0	30.0000
888	NaN	23.4500
889	26.0	30.0000
890	32.0	7.7500

891 rows × 2 columns

```
In [14]: df1[["age", "fare"]].iplot()
```



```
In [15]: df1[["age", "fare"]].iplot(x = "age", y = "fare", mode = "markers", size = 5)
```



```
In [16]: # In the above graph we are not finding any relation between the age & fare column.
```

```
In [16]: tip = sns.load_dataset("tips")
```

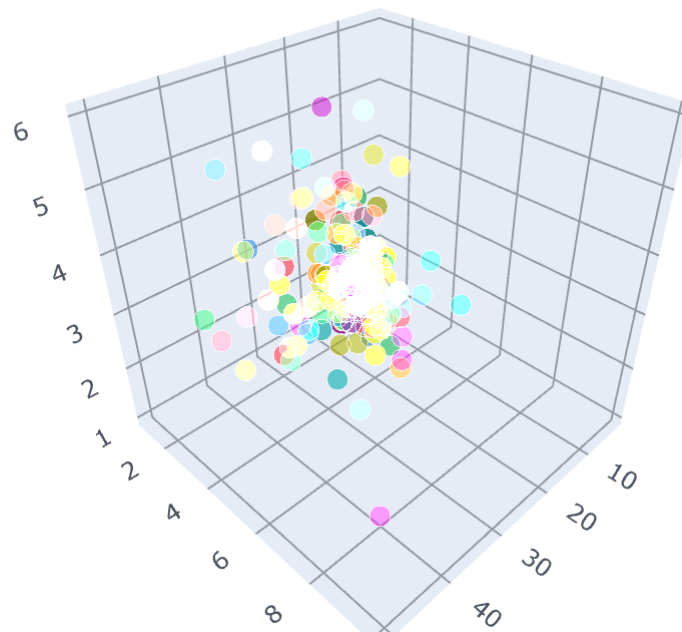
```
In [17]: tip
```

Out[17]:

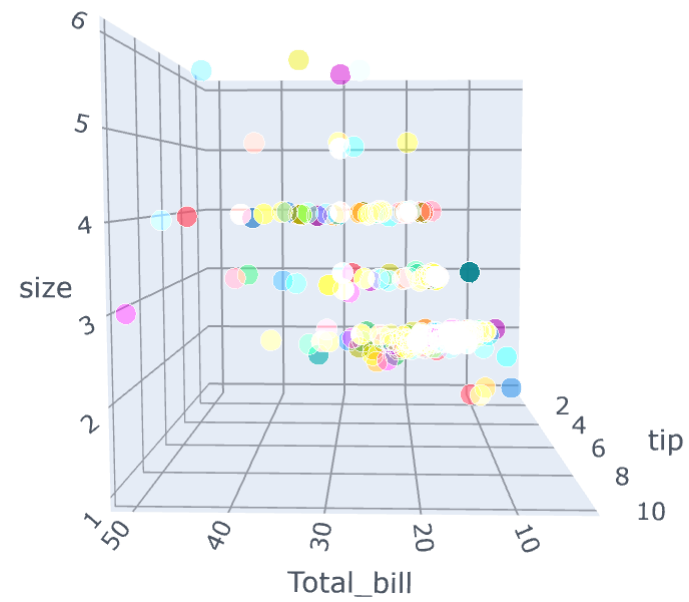
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67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

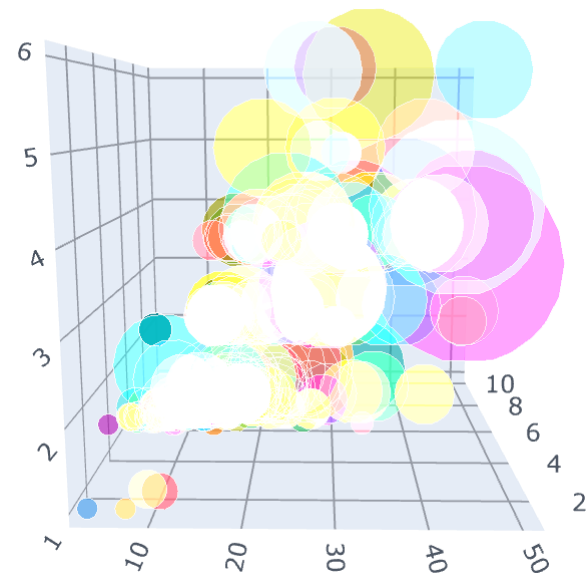
```
In [22]: tip.iplot(kind = 'scatter3d' , x = 'total_bill',y = 'tip',z = "size" )
```



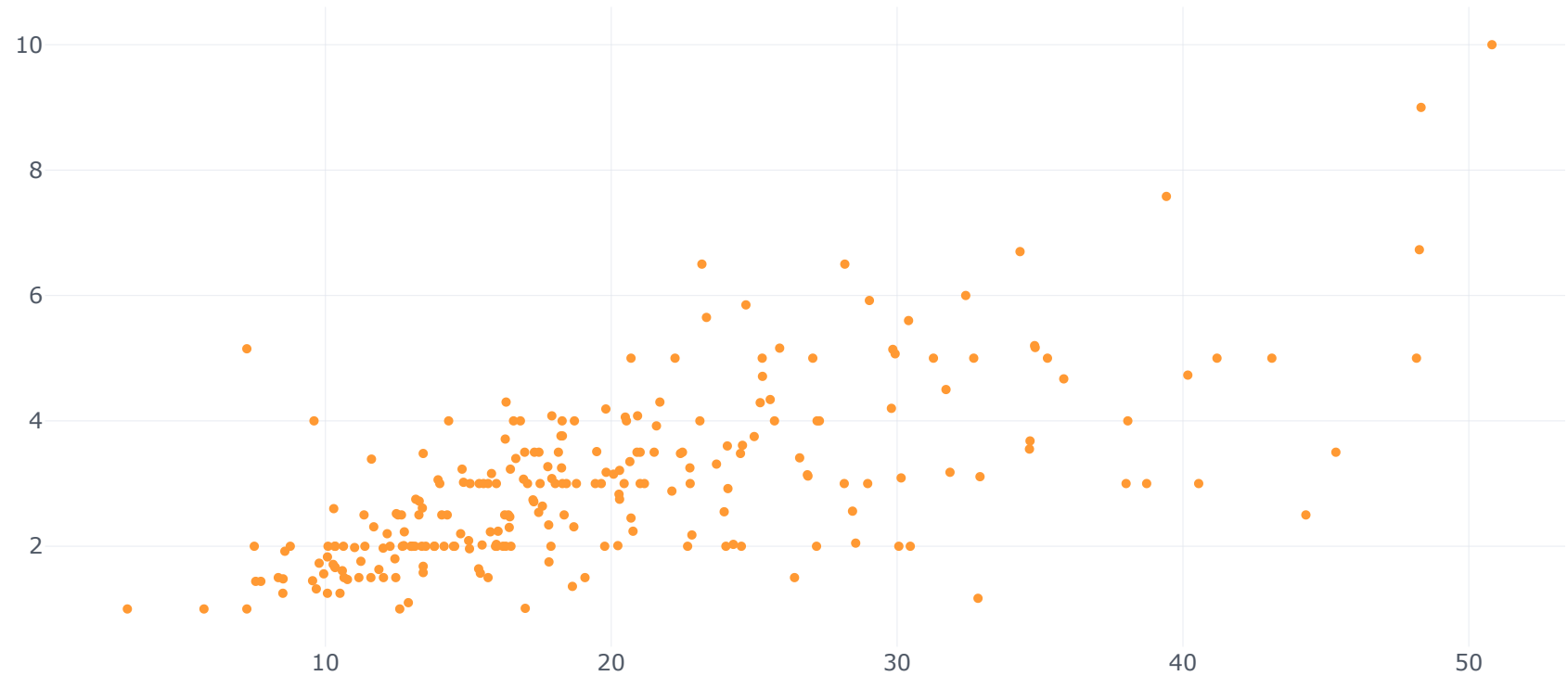
```
In [23]: tip.iplot(kind = 'scatter3d' , x = 'total_bill',y = 'tip' , z = 'size',xTitle="Total_bill",yTitle="tip",zTitle="size" )
```



```
In [24]: tip.iplot(kind = 'bubble3d' , x = 'total_bill',y = 'tip' , z = 'size' , size = 'tip' )
```




```
In [25]: tip[['total_bill' , 'tip']].iplot(x = "total_bill" , y = "tip" ,mode = "markers",size = 5)
```



In [26]: tip

Out[26]:

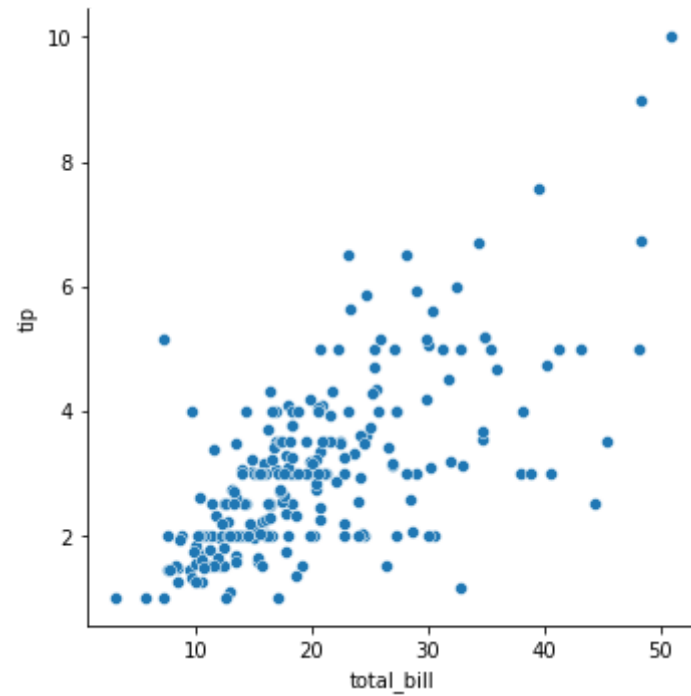
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	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 [27]: *# style gives a*

```
sns.relplot(x = "total_bill", y = "tip", data = tip)
```

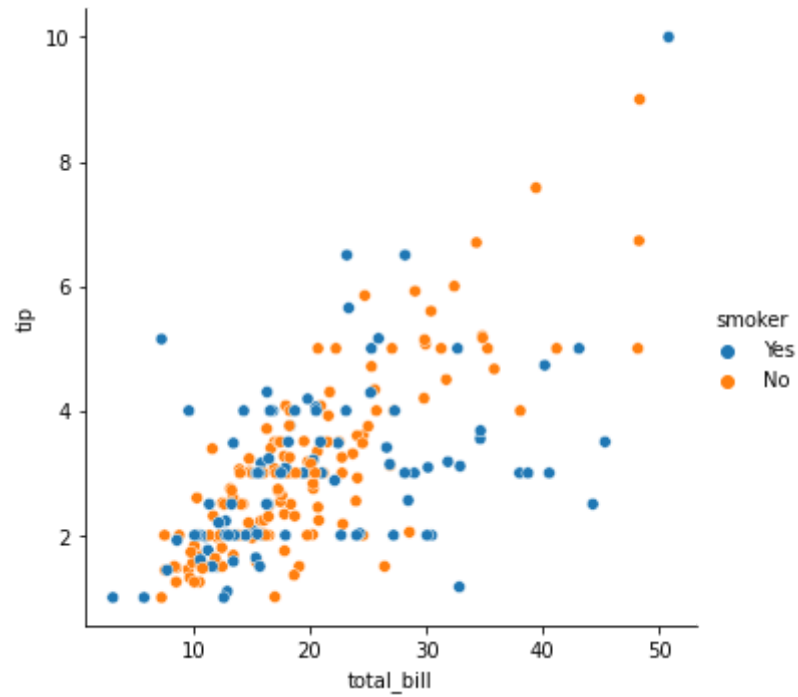
Out[27]: <seaborn.axisgrid.FacetGrid at 0x1727233f5e0>



In [28]: *# hue parameter shows a difference based on dataset*

```
sns.relplot(x = "total_bill", y = "tip", data = tip, hue = "smoker")
```

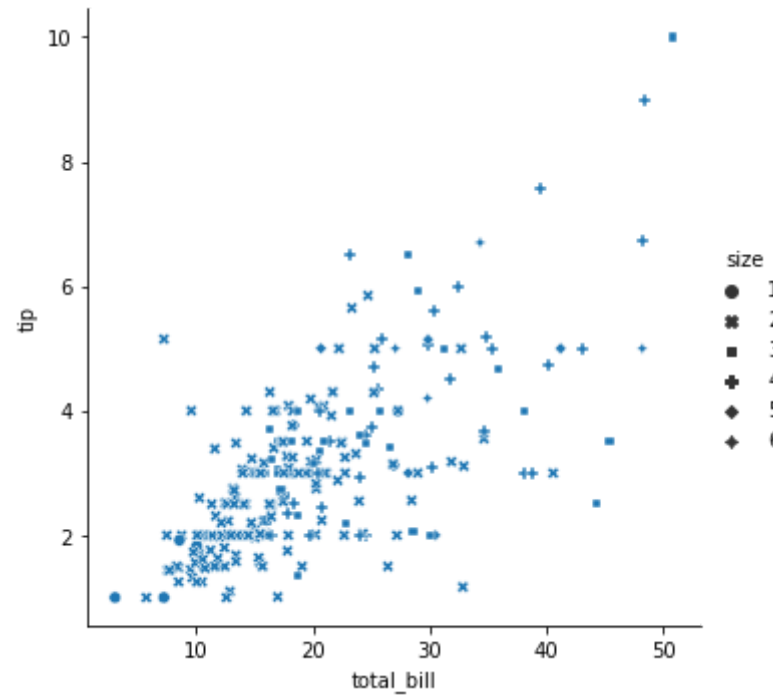
Out[28]: <seaborn.axisgrid.FacetGrid at 0x172726f9190>



```
In [30]: # style shows the shape of the every size
```

```
sns.relplot(x = "total_bill", y = "tip", data = tip, style = "size")
```

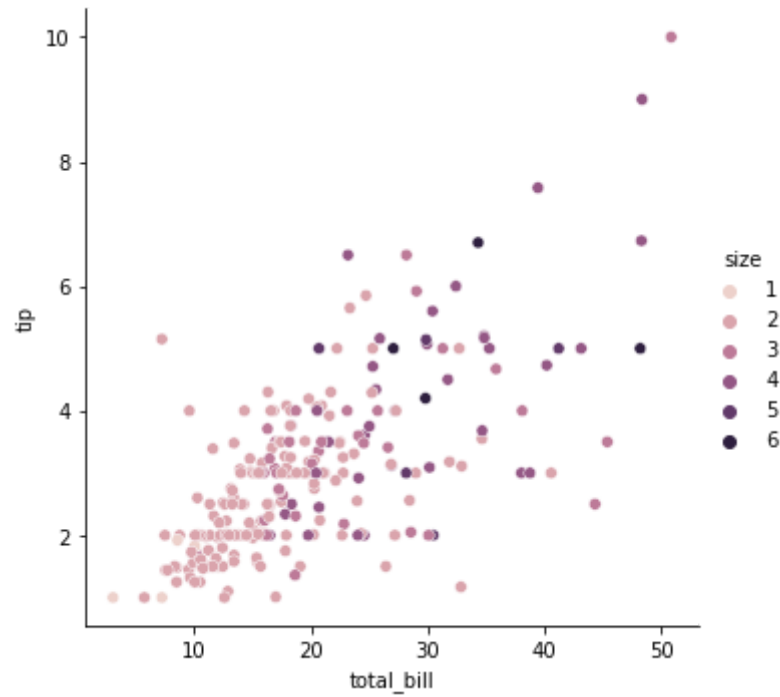
```
Out[30]: <seaborn.axisgrid.FacetGrid at 0x17272bf3eb0>
```



In [32]: *# hue gives a different color based on different size of data.*

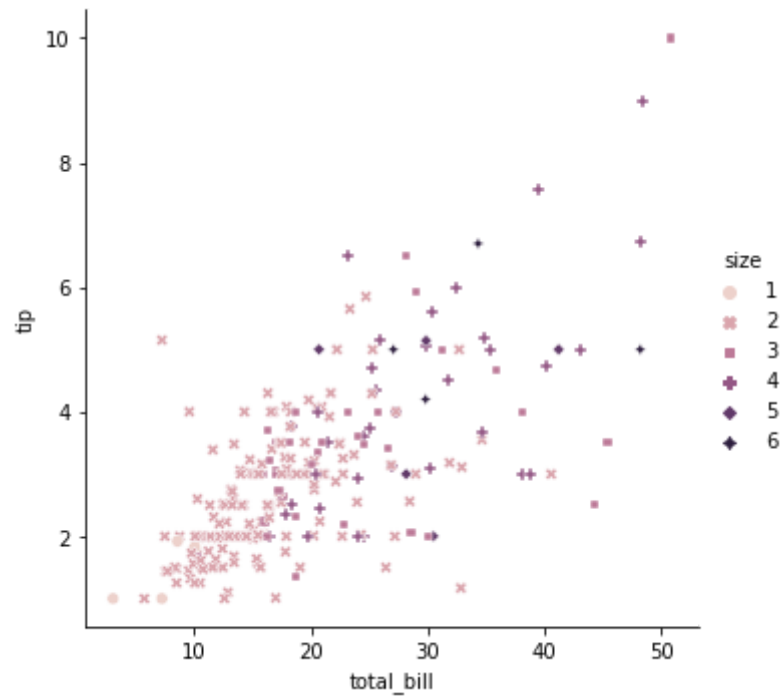
```
sns.relplot(x = "total_bill", y = "tip", data = tip, hue = "size")
```

Out[32]: <seaborn.axisgrid.FacetGrid at 0x17272fc64f0>



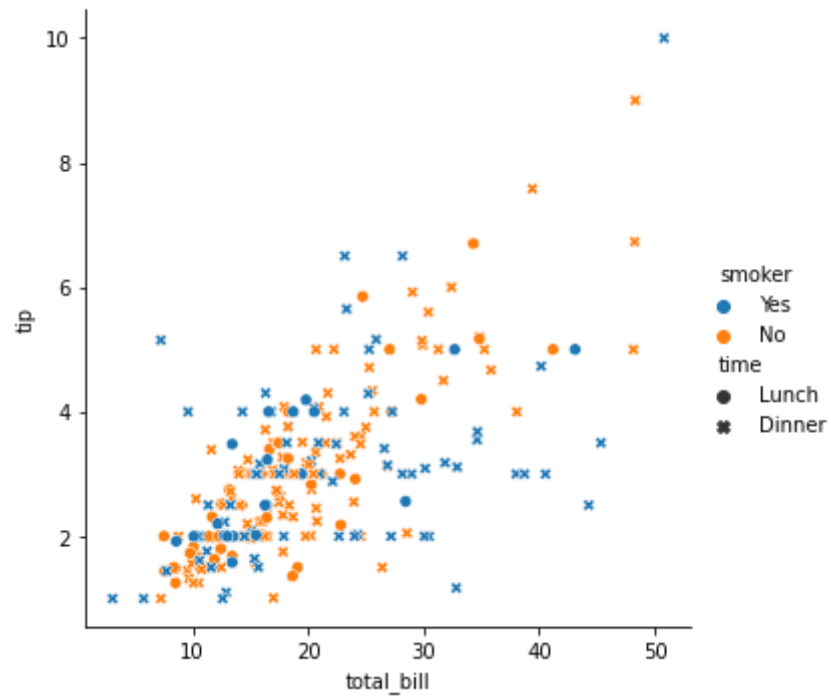
```
In [33]: sns.relplot(x = "total_bill", y = "tip", data = tip, hue = "size", style = "size")
```

```
Out[33]: <seaborn.axisgrid.FacetGrid at 0x17272fd3250>
```



```
In [34]: sns.relplot(x = "total_bill", y = "tip", data = tip , hue = "smoker", style = "time")
```

```
Out[34]: <seaborn.axisgrid.FacetGrid at 0x17272780c70>
```



```
In [37]: iris = sns.load_dataset("iris")
```


In [38]: iris

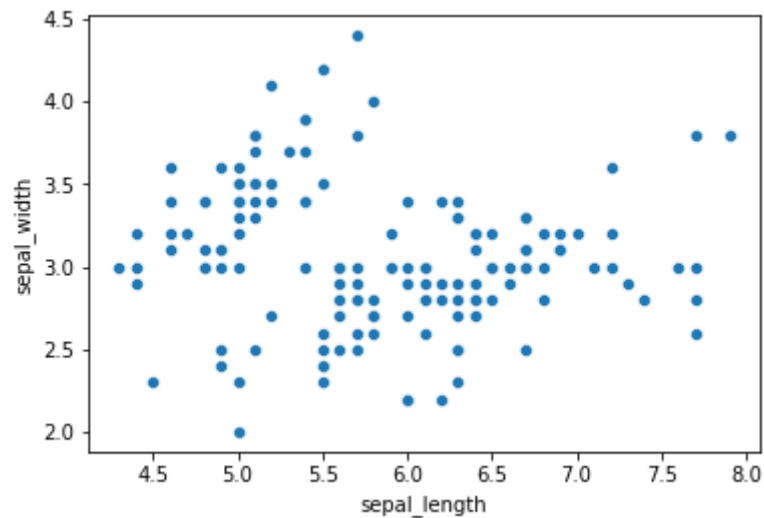
Out[38]:

	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

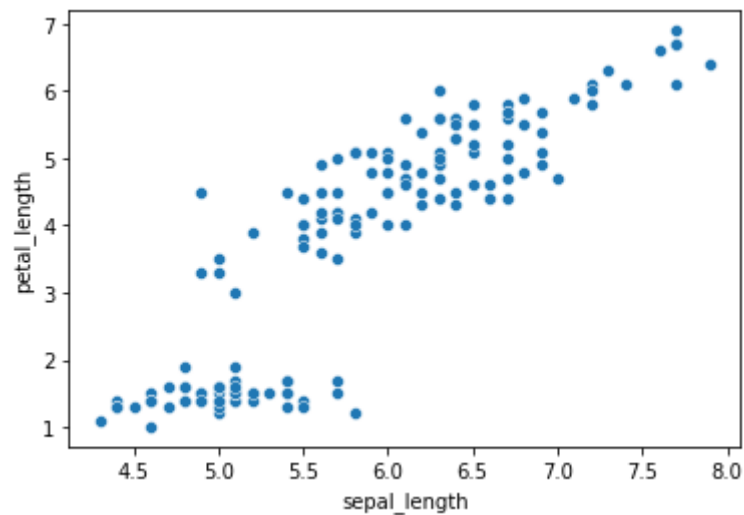
```
In [41]: sns.scatterplot(x = "sepal_length",y="sepal_width",data = iris)
```

```
Out[41]: <AxesSubplot:xlabel='sepal_length', ylabel='sepal_width'>
```



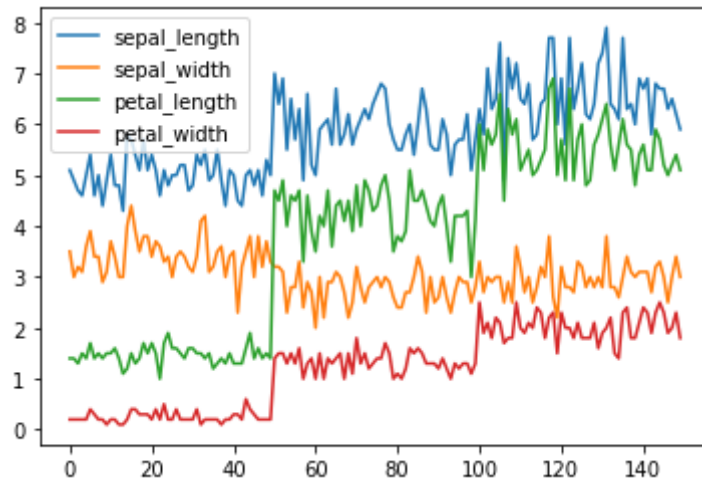
```
In [42]: sns.scatterplot(x = "sepal_length",y = "petal_length",data = iris)
```

```
Out[42]: <AxesSubplot:xlabel='sepal_length', ylabel='petal_length'>
```



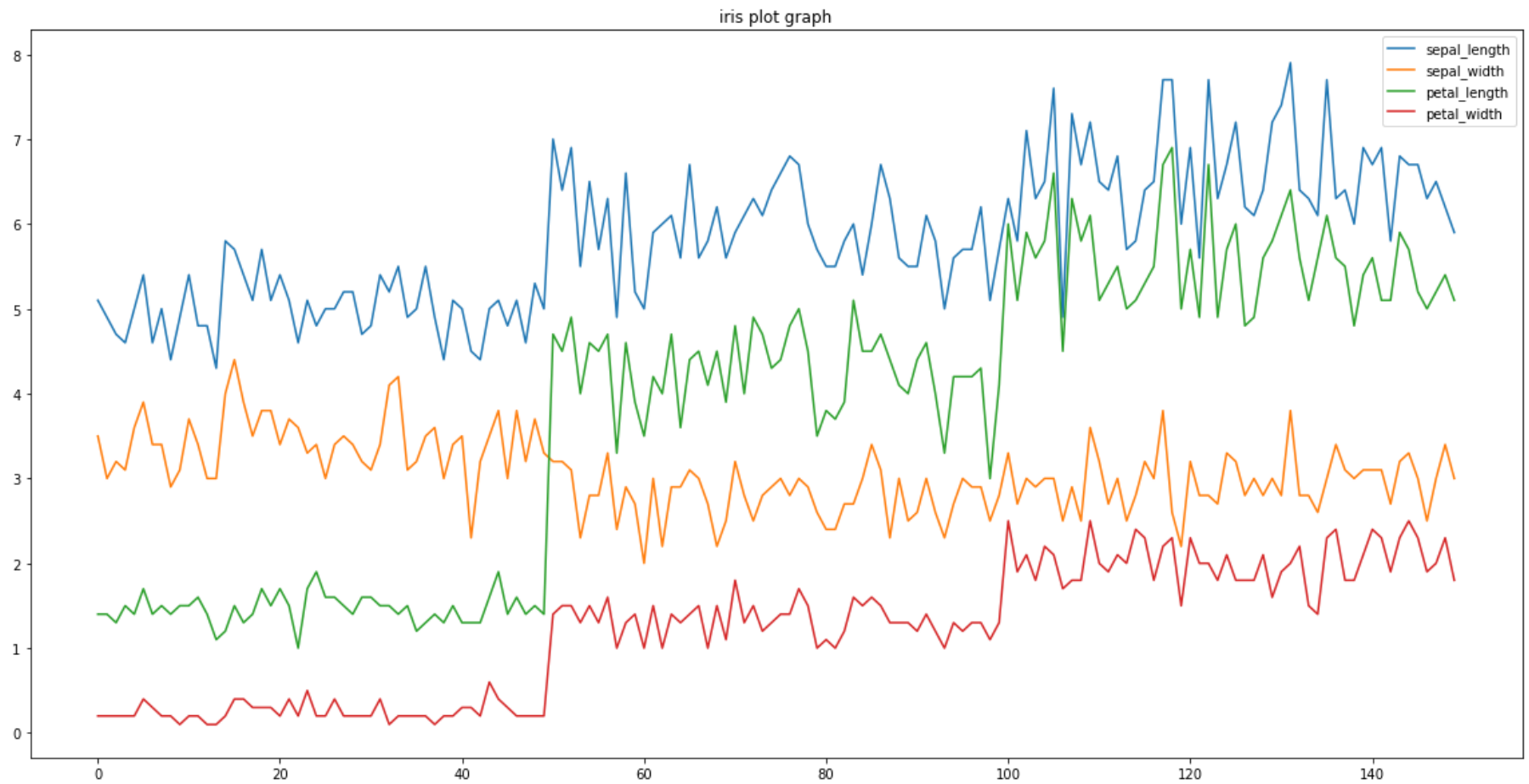
```
In [43]: # This is called pandas plot & iplot() is plotly plot  
  
iris.plot()
```

Out[43]: <AxesSubplot:>



```
In [45]: iris.plot(figsize = (20,10), title = "iris plot graph")
```

```
Out[45]: <AxesSubplot:title={'center':'iris plot graph'}>
```



In [46]: iris

Out[46]:

	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

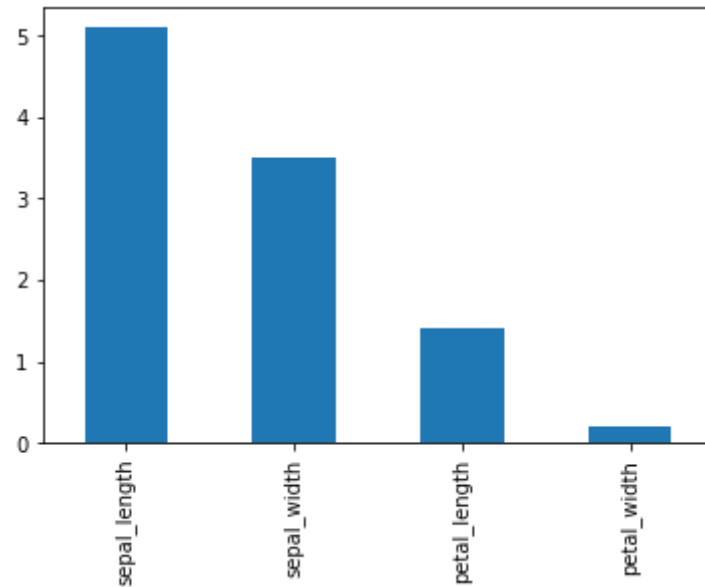
150 rows × 5 columns

```
In [47]: iris.iloc[0].plot(kind = 'bar')
```

```
-----  
TypeError                                Traceback (most recent call last)  
Input In [47], in <cell line: 1>()  
----> 1 iris.iloc[0].plot(kind = 'bar')  
  
File ~\anaconda3\lib\site-packages\pandas\plotting\_core.py:972, in PlotAccessor.__call__(self, *args, **kwargs)  
    969         label_name = label_kw or data.columns  
    970         data.columns = label_name  
--> 972 return plot_backend.plot(data, kind=kind, **kwargs)  
  
File ~\anaconda3\lib\site-packages\pandas\plotting\_matplotlib\__init__.py:71, in plot(data, kind, **kwargs)  
    69         kwargs["ax"] = getattr(ax, "left_ax", ax)  
    70 plot_obj = PLOT_CLASSES[kind](data, **kwargs)  
--> 71 plot_obj.generate()  
    72 plot_obj.draw()  
    73 return plot_obj.result  
  
File ~\anaconda3\lib\site-packages\pandas\plotting\_matplotlib\core.py:327, in MPLPlot.generate(self)  
    325 def generate(self):  
    326     self._args_adjust()  
--> 327     self._compute_plot_data()  
    328     self._setup_subplots()  
    329     self._make_plot()  
  
File ~\anaconda3\lib\site-packages\pandas\plotting\_matplotlib\core.py:506, in MPLPlot._compute_plot_data(self)  
    504 # no non-numeric frames or series allowed  
    505 if is_empty:  
--> 506     raise TypeError("no numeric data to plot")  
    508 self.data = numeric_data.apply(self._convert_to_ndarray)  
  
TypeError: no numeric data to plot
```

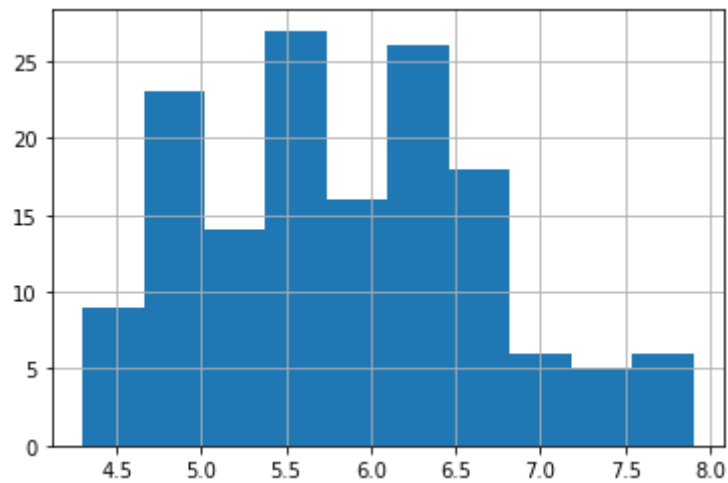
```
In [48]: # The above code is showing error because in the above dataset one column is having categorical data, so to avoid that  
iris.iloc[0,[0,1,2,3]].plot(kind = "bar")
```

Out[48]: <AxesSubplot:>



```
In [49]: iris["sepal_length"].hist()
```

```
Out[49]: <AxesSubplot:>
```



```
In [51]: data = iris.iloc[0,[0,1,2,3]]
```

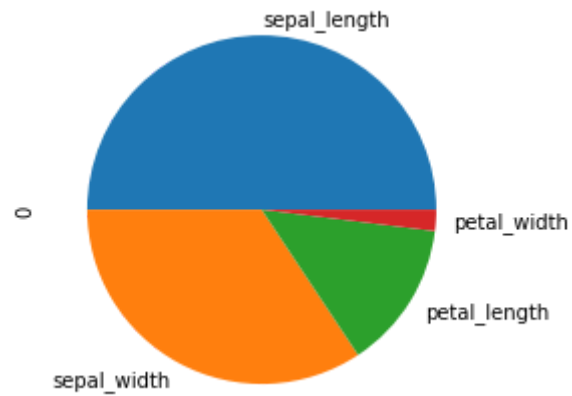
```
In [52]: data
```

```
Out[52]: sepal_length    5.1  
sepal_width    3.5  
petal_length    1.4  
petal_width    0.2  
Name: 0, dtype: object
```



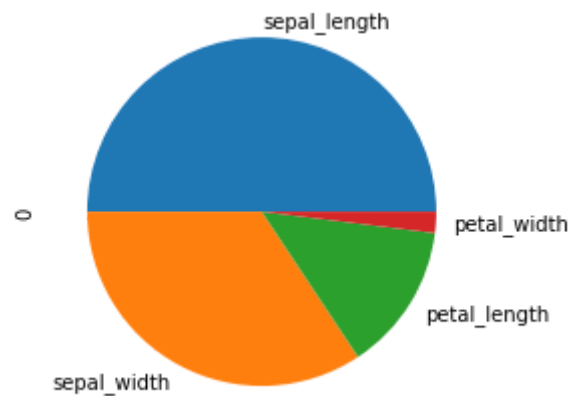
```
In [54]: data.plot.pie()
```

```
Out[54]: <AxesSubplot:ylabel='0'>
```



```
In [55]: # To save the graph and to use in any of the ppt's we can save the any the graph in this way.
```

```
a = data.plot.pie()  
b= a.get_figure()  
b.savefig('test.png')
```



In [56]: `ls`

Volume in drive C is OS
Volume Serial Number is 445D-2ABB

Directory of C:\Users\nav

26-02-2023	11:19 PM	<DIR>	.
05-10-2022	07:07 PM	<DIR>	..
19-02-2023	11:43 AM	<DIR>	.anaconda
26-08-2022	10:35 AM	<DIR>	.astropy
23-10-2022	01:03 PM		199 .bash_history
19-02-2023	11:52 AM	<DIR>	.conda
19-02-2023	11:43 AM		25 .condarc
19-02-2023	11:42 AM	<DIR>	.continuum
26-02-2023	09:13 PM	<DIR>	.cufflinks
25-08-2022	08:15 AM		61 .gitconfig
26-08-2022	02:15 PM	<DIR>	.gnupg
25-02-2023	04:01 PM	<DIR>	.ipynb_checkpoints
17-10-2022	02:34 PM	<DIR>	.ipython
15-09-2022	12:28 PM	<DIR>	.jupyter
26-08-2022	10:42 AM		20 .lessht
26-08-2022	10:45 AM	<DIR>	.matplotlib
24-08-2022	12:15 PM	<DIR>	.ssh
07-02-2023	09:53 PM		1,173 addresses.csv
10-02-2023	07:36 PM		16,384 airline.xls
10-02-2023	07:44 PM		32,256 airline1.xls
19-02-2023	11:49 AM	<DIR>	anaconda3
22-02-2023	05:40 PM		21,470 Assignment on 22nd february.ipynb
24-02-2023	10:51 AM		16,976 Assignment on 23rd feb.ipynb
05-10-2022	07:17 PM	<DIR>	Contacts
07-02-2023	09:56 PM		1,589 contacts.csv
15-01-2023	06:41 PM	<DIR>	Documents
24-02-2023	09:53 AM	<DIR>	Downloads
05-10-2022	07:17 PM	<DIR>	Favorites
24-02-2023	10:02 PM		52,043 Graphs.ipynb
10-02-2023	06:04 PM		277 holiday_schedules.csv
05-10-2022	07:17 PM	<DIR>	Links
10-02-2023	06:04 PM		18,015 locations.csv
10-02-2023	06:04 PM		1,610 mail_addresses.csv
05-10-2022	07:17 PM	<DIR>	Music
23-02-2023	10:01 AM		23,716 Numpy part -2.ipynb
23-02-2023	10:49 AM		27,170 Numpy.ipynb

```

26-02-2023 09:09 PM <DIR> OneDrive
10-02-2023 06:04 PM      2,281 organizations.csv
19-02-2023 10:33 AM    301,016 Pandas Part-2.ipynb
20-02-2023 06:34 PM    91,305 Pandas part-3.ipynb
22-02-2023 08:51 AM    86,906 Pandas practice.ipynb
10-02-2023 06:04 PM      2,092 phones.csv
22-02-2023 08:48 AM    81,552 players.csv
20-08-2022 03:30 PM <DIR> Postman
10-02-2023 06:04 PM      60 programs.csv
13-02-2023 02:28 PM <DIR> PycharmProjects
23-02-2023 05:45 PM    16,513 Python Assignments.innomatics.ipynb
09-02-2023 11:37 PM      3,210 python patterns practice.ipynb
20-02-2023 01:51 PM      3,827 Python patterns.ipynb
10-02-2023 06:04 PM      350 regular_schedules.csv
05-10-2022 07:17 PM <DIR> Saved Games
26-02-2023 10:43 PM <DIR> seaborn-data
05-10-2022 07:17 PM <DIR> Searches
10-02-2023 06:04 PM    19,128 services.csv
13-02-2023 01:07 AM    30,587 String practice linkedin.ipynb
10-02-2023 12:36 PM      1,278 Strings class.ipynb
10-02-2023 06:04 PM    14,618 taxonomy.csv
26-02-2023 11:19 PM      8,602 test.png
10-02-2023 08:39 PM      608 test1.csv
10-02-2023 08:41 PM    6,087 test2.xlsx
17-10-2022 02:38 PM      4,033 Untitled.ipynb
14-02-2023 02:47 PM    19,208 Untitled1.ipynb
21-02-2023 01:29 PM      7,883 Untitled2.ipynb
24-02-2023 01:02 PM    21,617 Untitled3.ipynb
24-02-2023 01:09 AM      2,072 Untitled4.ipynb
26-02-2023 11:13 PM    1,228,437 Untitled5.ipynb
25-02-2023 05:23 PM    57,330 Untitled6.ipynb
25-02-2023 05:03 PM    4,000,859 Untitled7.ipynb
20-11-2022 08:41 AM <DIR> Videos
      42 File(s)      6,224,443 bytes
      27 Dir(s)  146,494,144,512 bytes free

```

In [57]: iris

Out[57]:

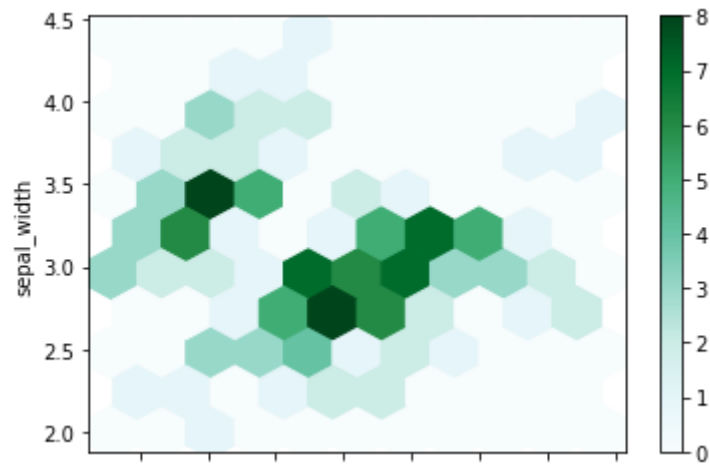
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica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

150 rows × 5 columns

In [58]: *# hexbin means nothing but simply we can say shape of the plot*

```
iris.plot.hexbin(x = 'sepal_length' , y ='sepal_width' ,gridsize=10)
```

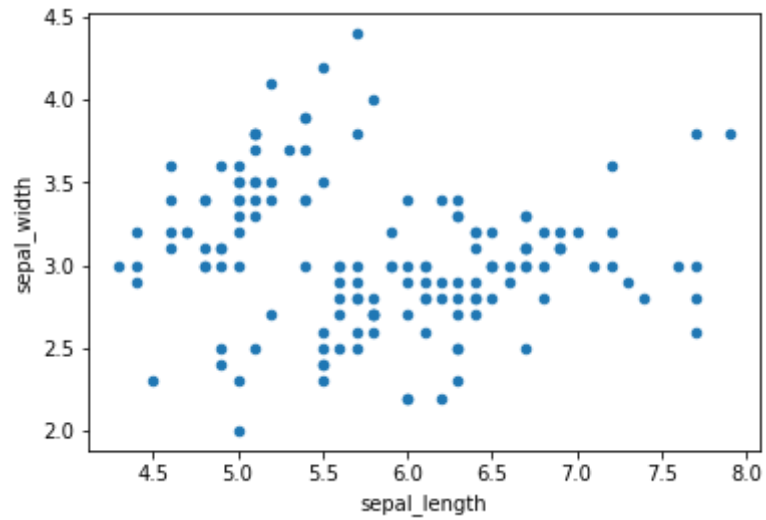
Out[58]: <AxesSubplot:xlabel='sepal_length', ylabel='sepal_width'>



In [64]: *# It is also same like hexbin but the difference is in the shape & size*

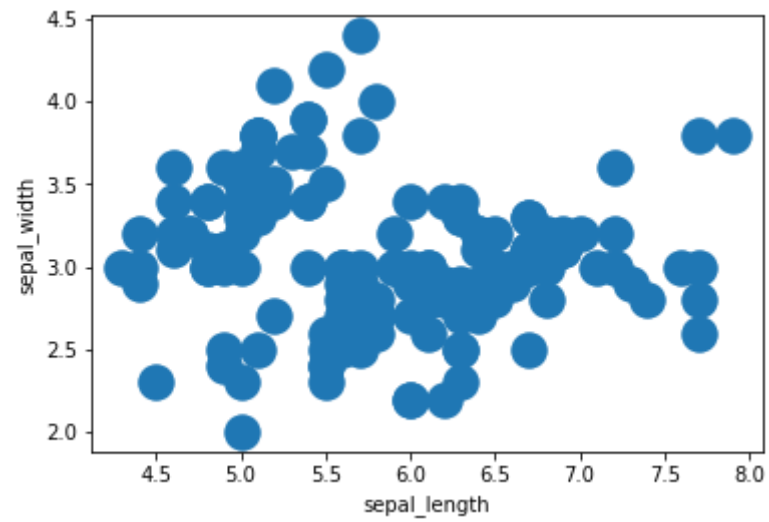
```
iris.plot.scatter(x = 'sepal_length' , y ='sepal_width' )
```

Out[64]: <AxesSubplot:xlabel='sepal_length', ylabel='sepal_width'>

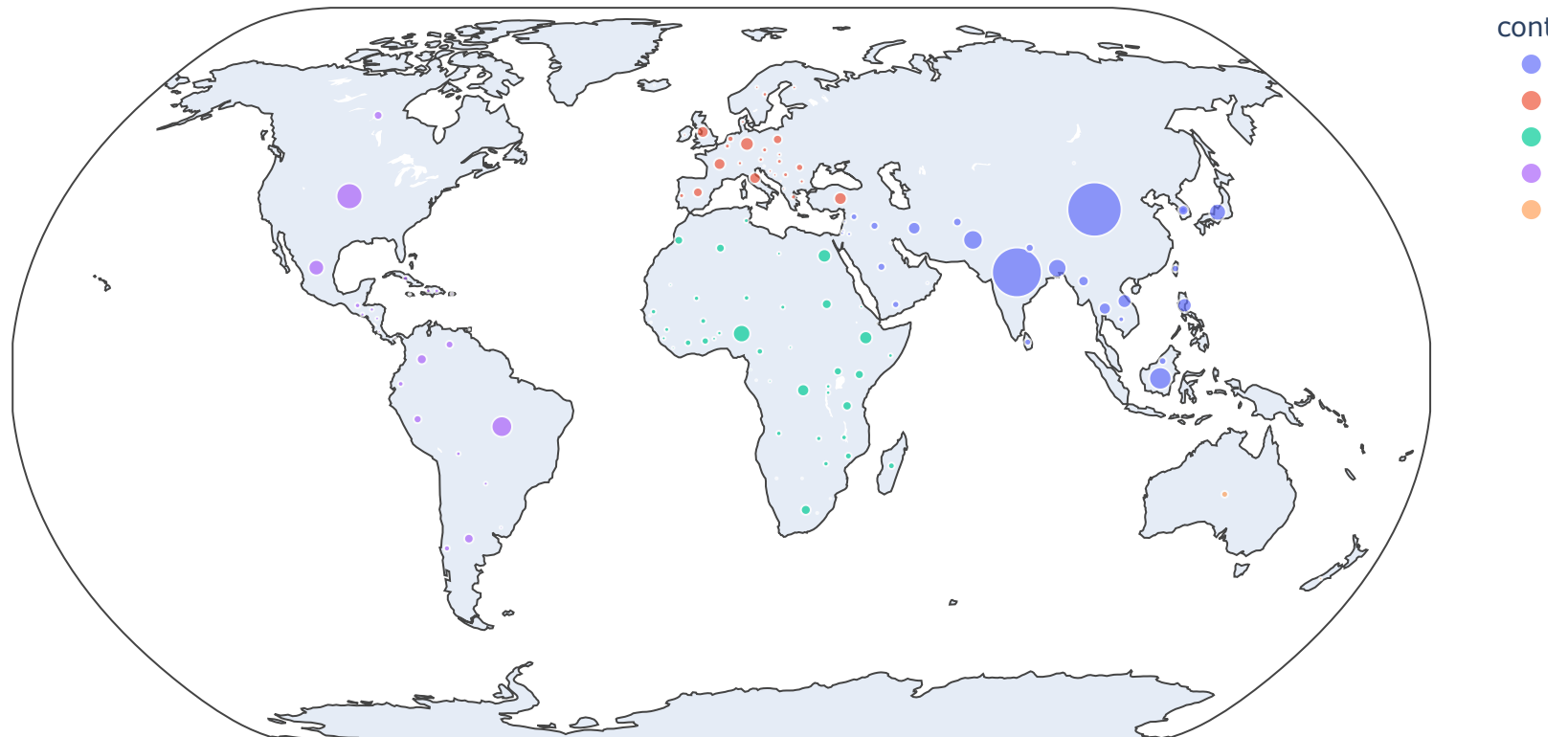


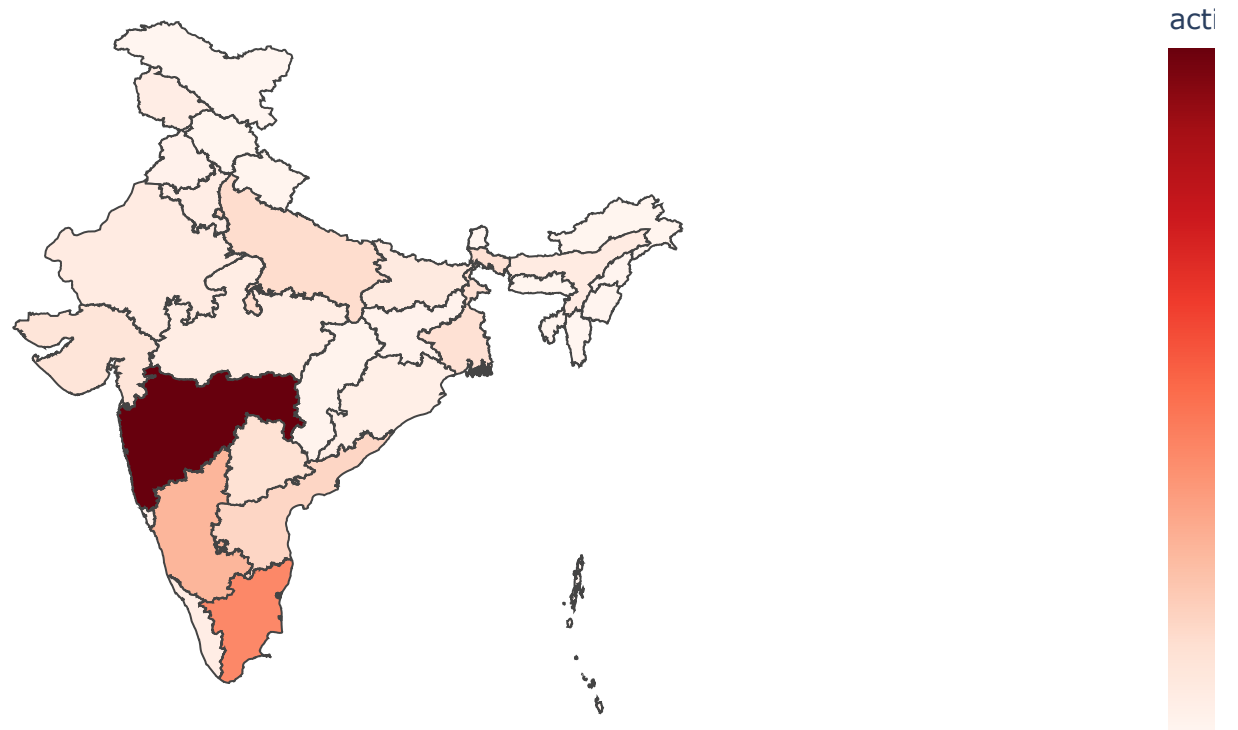
```
In [63]: iris.plot.scatter(x = 'sepal_length' , y ='sepal_width' , s = 300)
```

```
Out[63]: <AxesSubplot:xlabel='sepal_length', ylabel='sepal_width'>
```




```
In [65]: import plotly.express as px
df = px.data.gapminder().query("year == 2007")
fig = px.scatter_geo(df, locations="iso_alpha",
                    color="continent", # which column to use to set the color of markers
                    hover_name="country", # column added to hover information
                    size="pop", # size of markers
                    projection="natural earth")
fig.show()
```





```
In [68]: import plotly.graph_objects as go

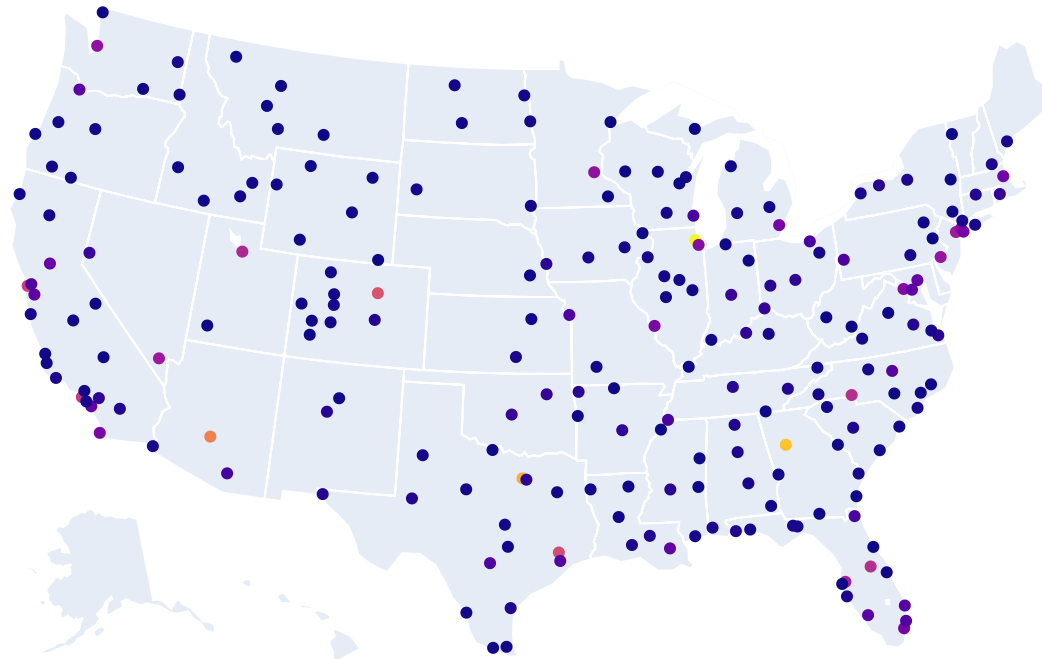
import pandas as pd

df = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/2011_february_us_airport_traffic.csv')
df['text'] = df['airport'] + ' ' + df['city'] + ', ' + df['state'] + ' ' + 'Arrivals: ' + df['cnt'].astype(str)

fig = go.Figure(data=go.Scattergeo(
    lon = df['long'],
    lat = df['lat'],
    text = df['text'],
    mode = 'markers',
    marker_color = df['cnt'],
))

fig.update_layout(
    title = 'Most trafficked US airports<br>(Hover for airport names)',
    geo_scope='usa',
)
fig.show()
```

Most trafficked US airports
(Hover for airport names)



In []: