

Plotly By Virat Tiwari -

Plotly - Plotly is used for data visualization and understanding the data simply and easily. We can visualize the data in the graphical form through which we can analyze the

- ▾ data because reading huge amount of data is not easy so we should convert that data into the graphical representation that enable us to understand huge amount of data easily, we also get the insights of data with the help of visualization.

1) Note - In Jupyter Notebook, plotly taking little bit more time for execution of code and sometimes it showing the glitch and lacking as well.

2) Note - In Jupyter Notebook, you firstly install the plotly package as " pip install plotly " after that you can import the plotly as " import plotly.graph_objects as go ".

3) Note - In Google Collab, plotly works smoothly without any lacking so i recommend you to use Google collab for Plotly and Bokeh as well for getting better result.

4) Note - In Google Collab, you can directly import plotly as " import plotly.graph_objects as go " without installing its package.

```
# We import plotly.graph_objects as go, go is our alias (representation for plotly, whenever we use plotly we call go)
```

```
import plotly.graph_objects as go
```

```
# fig=go.figure ( ) function is used for figure out the graph as a output
```

```
# fig.add_trace(go.Scatter( ) function is used for trace the graph and we also pass the go.Scatter ( ) inside the function for getting the output in Scatter Graph
```

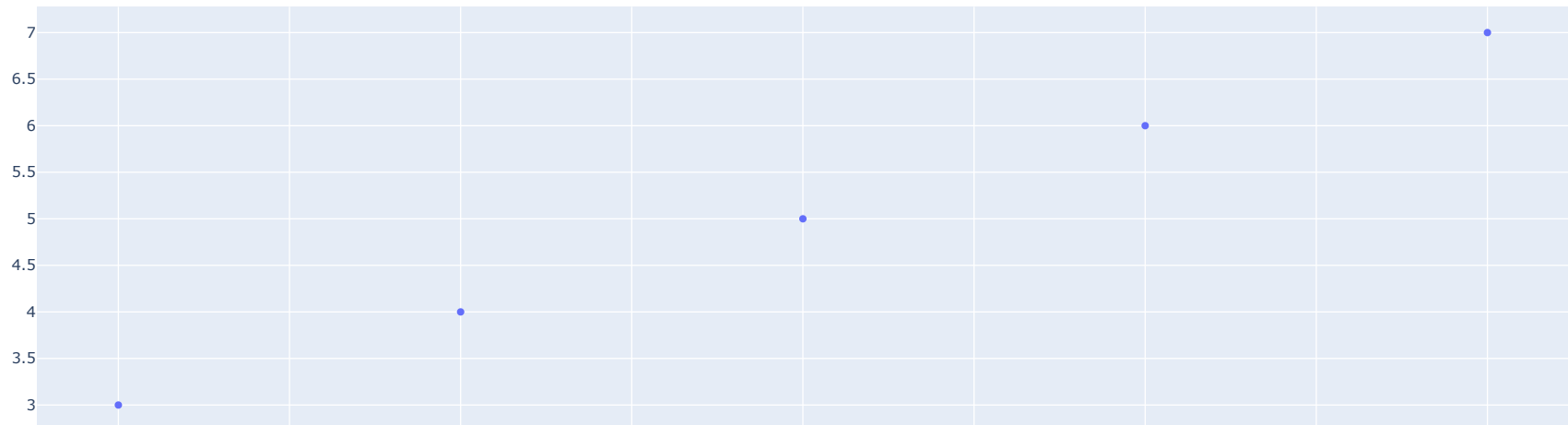
```
# We use " mode = markers " for getting dots inside the graph that shows the data
```

```
import plotly.graph_objects as go
```

```
fig=go.Figure()
```

```
fig.add_trace(go.Scatter(x=[1,2,3,4,5],y=[3,4,5,6,7],mode="markers"))
```

```
fig.show()
```

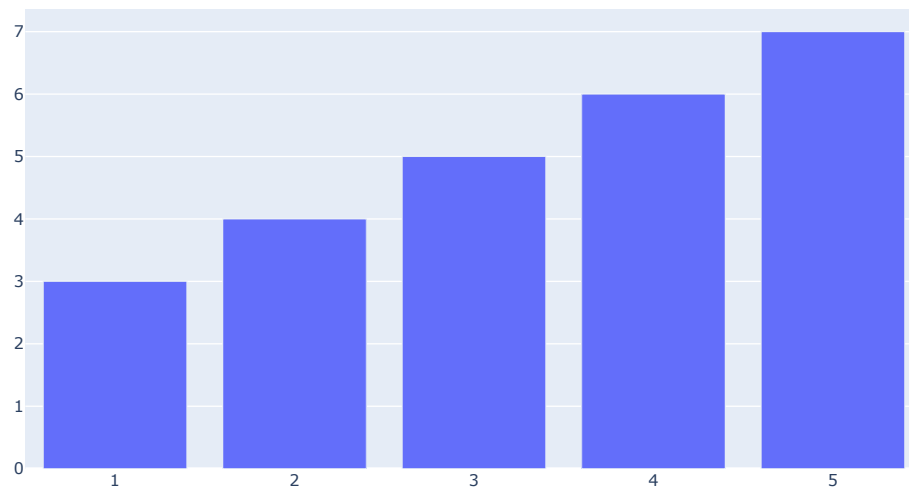


```
# fig=go.figure ( ) function is used for figure out the graph as a output
# fig.add_trace(go.Scatter( ) function is used for trace the graph and we also pass the go.Scatter ( ) inside the function for getting the output in Scatter Graph
# We use " mode = lines " for getting lines inside the graph that shows the data
```

```
import plotly.graph_objects as go
fig=go.Figure()
fig.add_trace(go.Scatter(x=[1,2,3,4,5],y=[3,4,5,6,7],mode="lines"))
fig.show()
```

```
# fig=go.figure ( ) function is used for figure out the graph as a output
# fig.add_trace(go.Bar ( ) function is used for trace the graph and we also pass the go.Bar ( ) inside the function for getting the output in a Bar format
# We use " mode = markers " for getting dots inside the graph that shows or data
```

```
import plotly.graph_objects as go
fig=go.Figure()
fig.add_trace(go.Bar(x=[1,2,3,4,5],y=[3,4,5,6,7]))
fig.show()
```

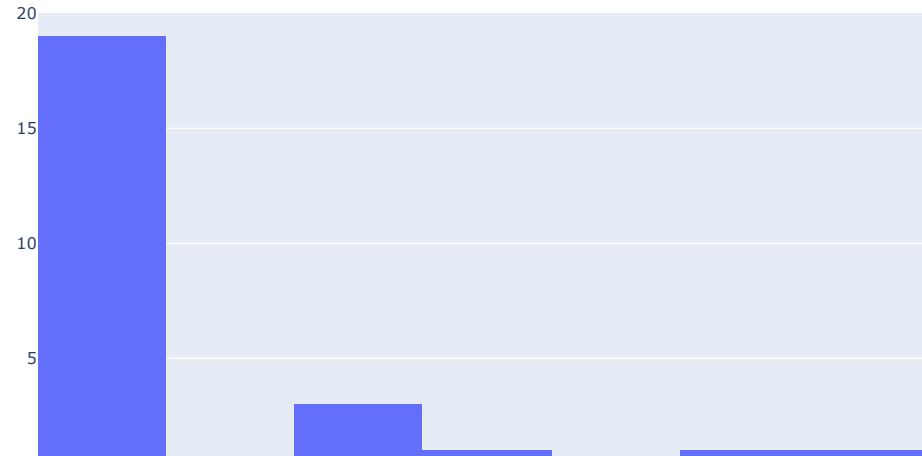


```
# Here we made " x = [ ] " list as a data
```

```
x=[1,2,3,4,5,6,7,8,8,5,4,6,201,6,9,7,201,1,3,201,3,366,14,622,563]
```

```
# fig=go.Figure(data=[go.Histogram( )]) - This function is used for making the presentation of data in histogram graph
# fig=go.Figure(data=[go.Histogram(x=x)]) in this function we pass our data " x " for getting result in Histogram format
```

```
fig=go.Figure(data=[go.Histogram(x=x)])
fig.show()
```



```
# Seaborn provided some built-in datasets like " tips " , " iris " etc
# We import searbor as sns for using its some built-in datasets

import seaborn as sns

# sns.load_dataset ( ) function is used for importing the searborn built-in datasets , we simply pass the name of dataset inside the load_dataset ( ) function

sns.load_dataset("tips")
```

	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

```
# We store the " tips " dataset in " tips "" variable
```

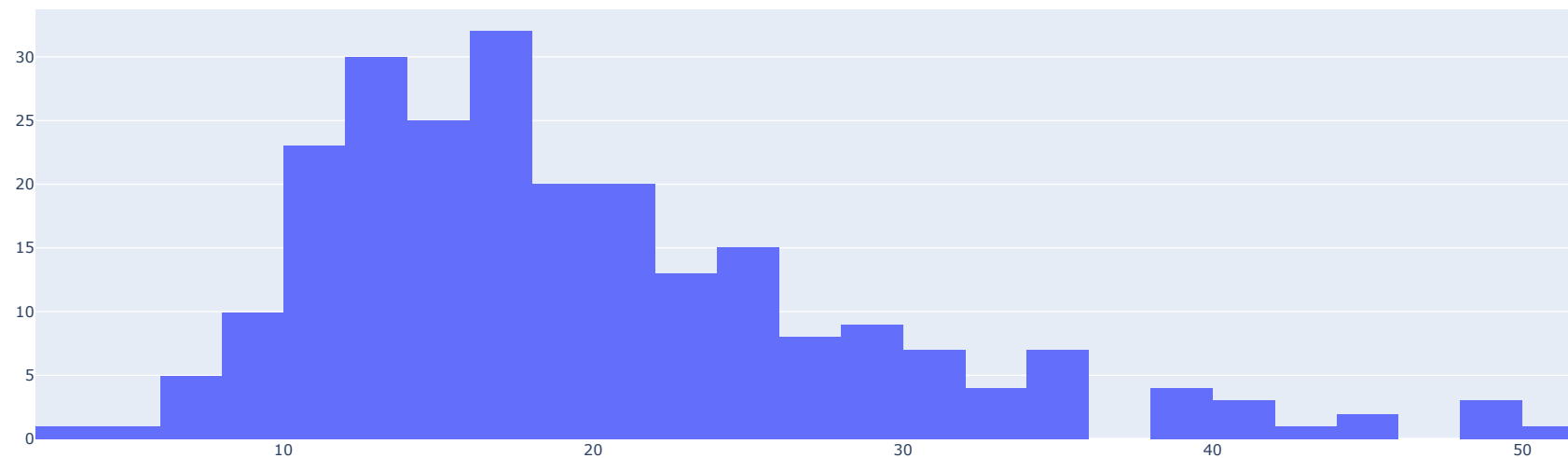
```
tips=sns.load_dataset("tips")
```

```
# Here we visualize the " total_bill " column of tips dataset in histogram format
```

```
# fig=go.Figure(data=[go.Histogram(x=tips.total_bill)]) - This is how we perform the visualization operation
```

```
fig=go.Figure(data=[go.Histogram(x=tips.total_bill)])
```

```
fig.show()
```

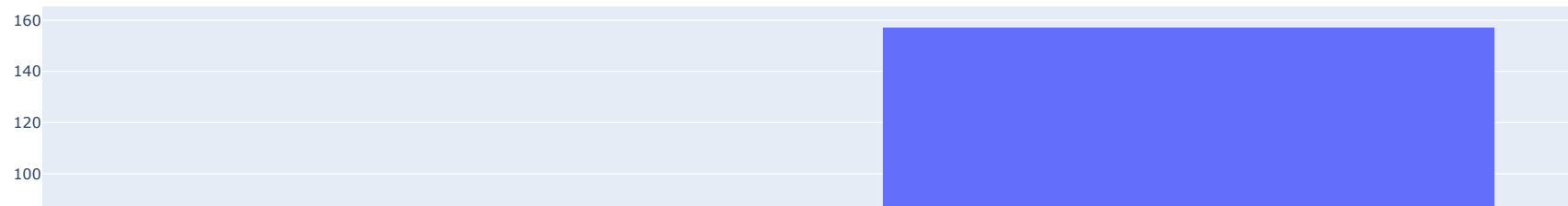


```
# Here we visualize the " sex " column of tips dataset in histogram format
```

```
# fig=go.Figure(data=[go.Histogram(x=tips.sex)]) - This is how we perform the visualization operation
```

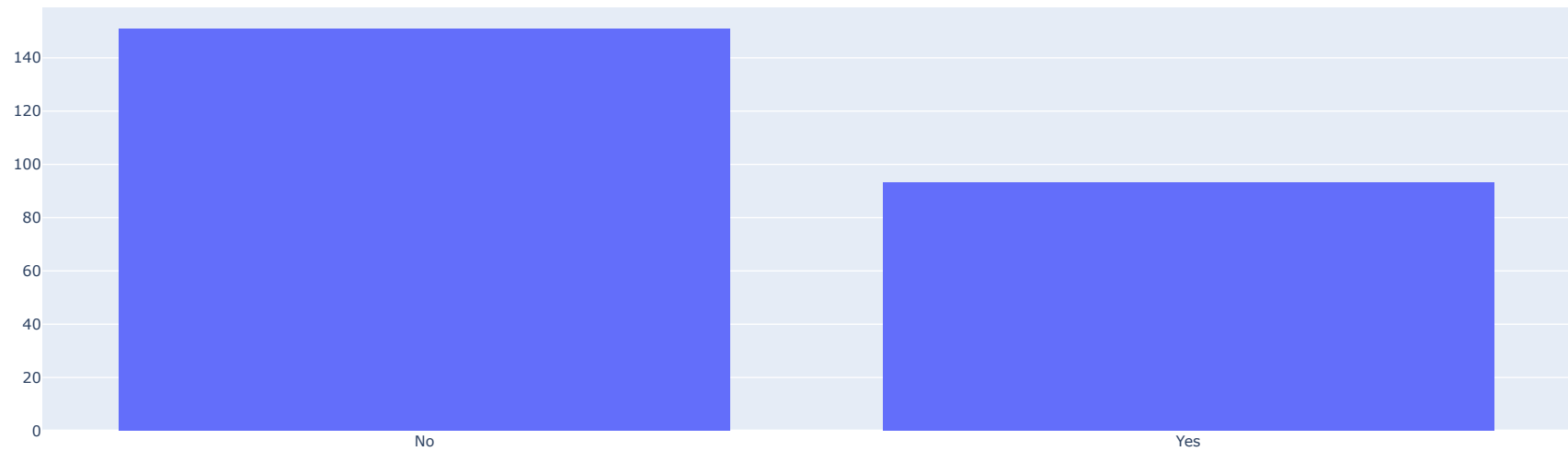
```
fig=go.Figure(data=[go.Histogram(x=tips.sex)])
```

```
fig.show()
```



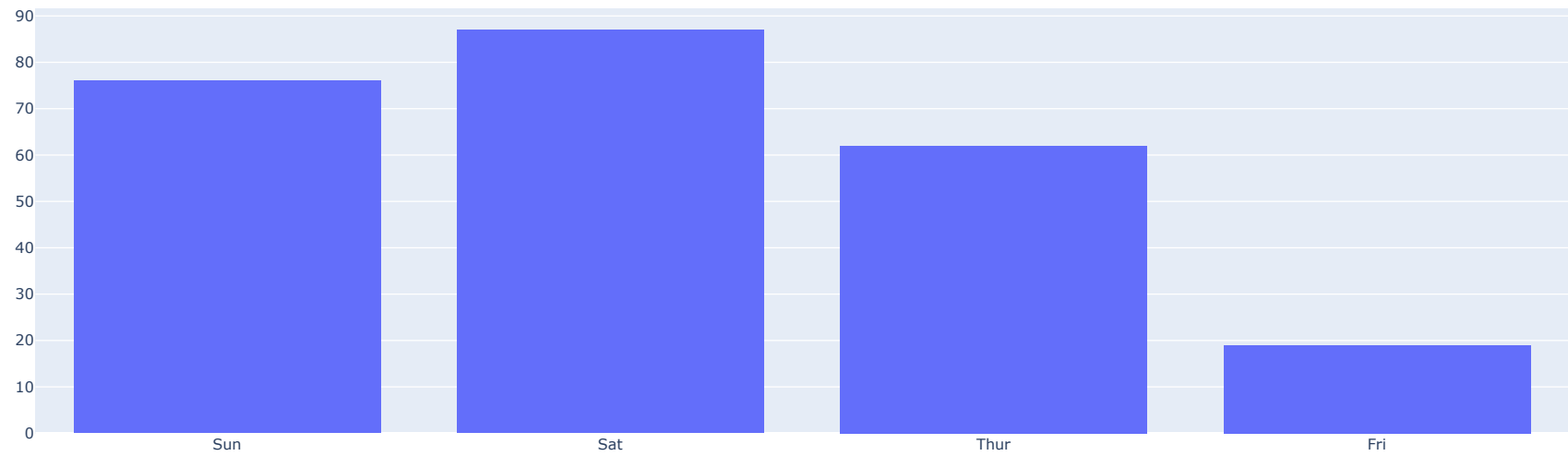
```
# Here we visualize the " smoker " column of tips dataset in histogram format
# fig=go.Figure(data=[go.Histogram(x=tips.smoker)]) - This is how we perform the visualization operation

fig=go.Figure(data=[go.Histogram(x=tips.smoker)])
fig.show()
```



```
# Here we visualize the " day " column of tips dataset in histogram format
# fig=go.Figure(data=[go.Histogram(x=tips.day)]) - This is how we perform the visualization operation

fig=go.Figure(data=[go.Histogram(x=tips.day)])
fig.show()
```

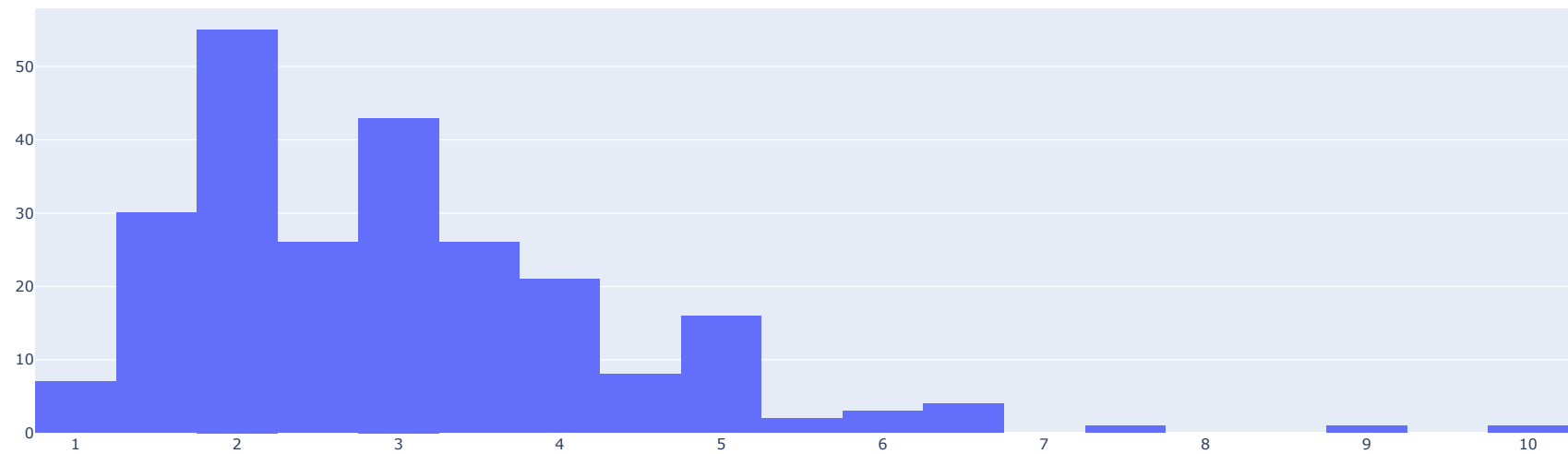


```
# Here we visualize the " time " column of tips dataset in histogram format
# fig=go.Figure(data=[go.Histogram(x=tips.time)]) - This is how we perform the visualization operation

fig=go.Figure(data=[go.Histogram(x=tips.time)])
fig.show()
```

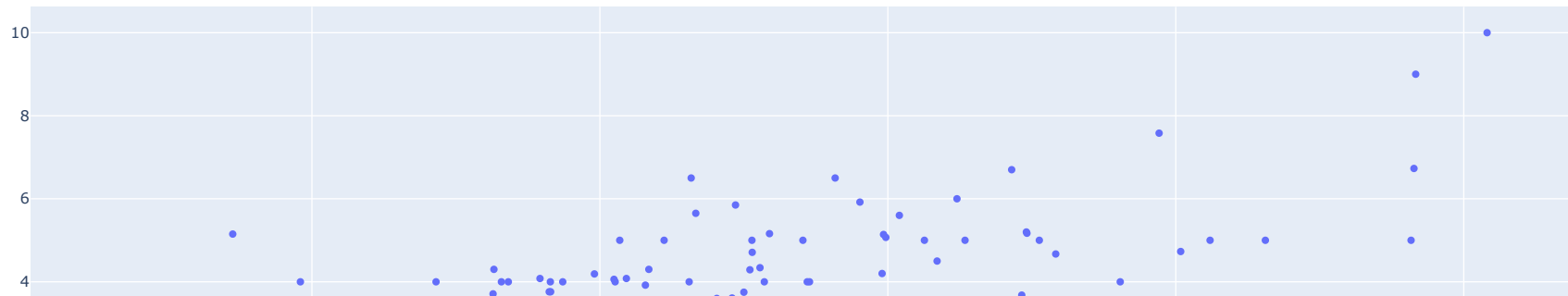
```
# Here we visualize the " tip " column of tips dataset in histogram format
# fig=go.Figure(data=[go.Histogram(x=tips.tip)]) - This is how we perform the visualization operation
```

```
fig=go.Figure(data=[go.Histogram(x=tips.tip)])
fig.show()
```



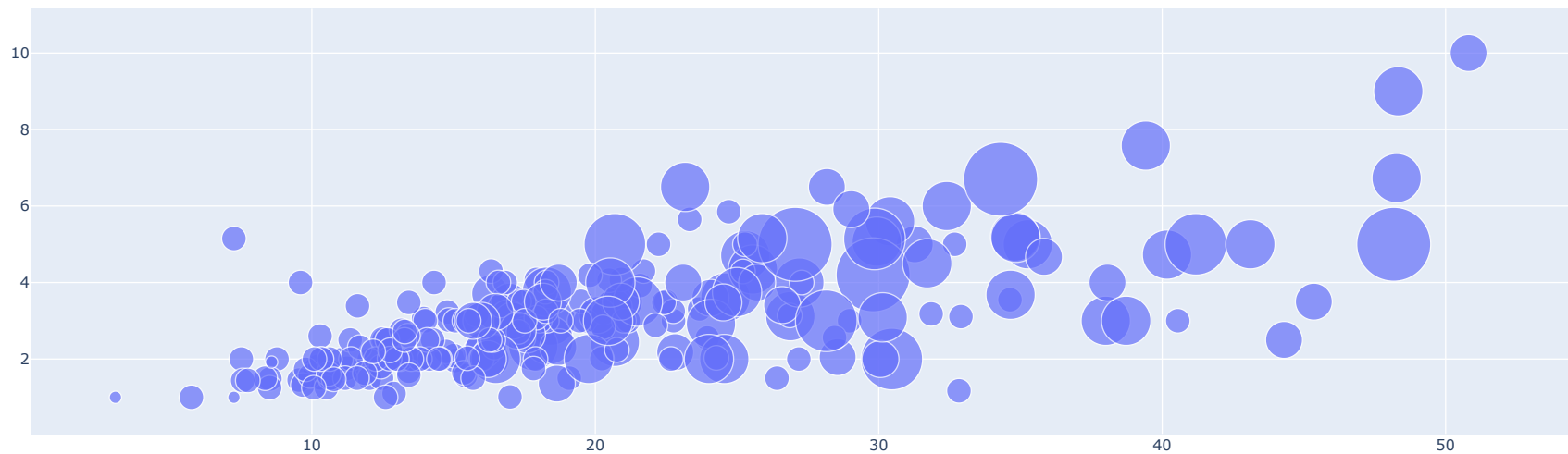
```
# Here we visualize the previous graph in scatter format
# fig=go.Figure( ) function is used for presentation the graph as as result of execution
# fig.show( ) function is showing the final output
```

```
import plotly.graph_objects as go
fig=go.Figure()
fig.add_trace(go.Scatter(x=tips.total_bill,y=tips.tip,mode="markers"))
fig.show()
```

```
# By giving the mode = "markers" we set the dots as a data inside the graph
# By giving the " marker_size=10*tips[tips] " we set the size of the dots inside the graph

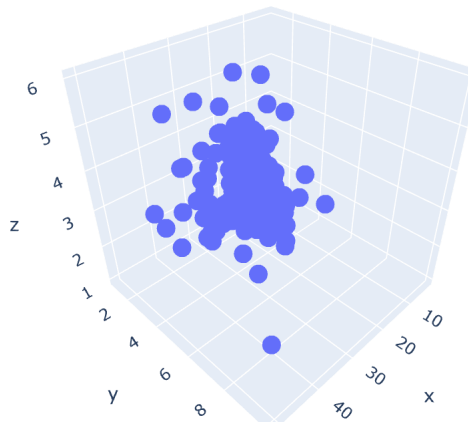
import plotly.graph_objects as go
fig=go.Figure()
fig.add_trace(go.Scatter(x=tips.total_bill,y=tips.tip,mode="markers",marker_size=10*tips["size"])))
fig.show()
```



```
# 3d Graph
# fig.add_trace( go.Scatter3d ) - This function enables the data into the 3d visualization
```

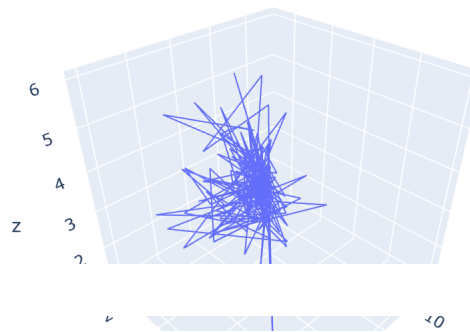
```
# We pass three coordinates "x" , "y" and "z" inside the function for 3d graph
# ex -: Here we pass these 3 co-ordinates as x , y and z - (x=tips.total_bill,y=tips.tip,mode="markers",z=tips["size"])

import plotly.graph_objects as go
fig=go.Figure()
fig.add_trace(go.Scatter3d(x=tips.total_bill,y=tips.tip,mode="markers",z=tips["size"]))
fig.show()
```



```
# 3d Grapgh
# fig.add_trace( go.Scatter3d ) - This function enables the data into the 3d visualization
# We pass three coordinates "x" , "y" and "z" inside the function for 3d graph
# ex -: Here we pass these 3 co-ordinates as x , y and z - (x=tips.total_bill,y=tips.tip,mode="lines",z=tips["size"])

import plotly.graph_objects as go
fig=go.Figure()
fig.add_trace(go.Scatter3d(x=tips.total_bill,y=tips.tip,mode="lines",z=tips["size"]))
fig.show()
```



THANK YOU SO MUCH !!
YOURS VIRAT TWIARI :)