

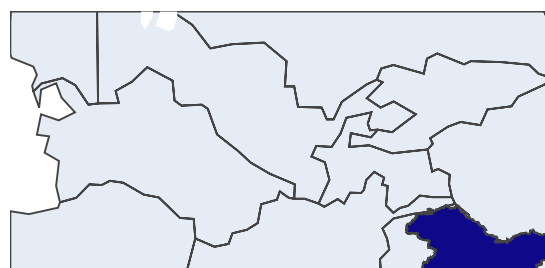
```
# 1> Display a Choropleth Map
# i> To show variation of Literacy in different Indian states
import json
import pandas as pd
import plotly.express as px
INDIA_MAP=json.load(open('/content/states_india.geojson','r'))
INDIA_DATA=pd.read_csv('India-Population1.csv')
MAP=px.choropleth(INDIA_DATA,geojson=INDIA_MAP,
locations='State',
featureidkey="properties.st_nm",
color='Literacy',scope='asia',fitbounds='locations')
MAP.show()
```

```
-----
FileNotFoundError                                Traceback (most recent call last)
<ipython-input-7-6f613819e1ac> in <cell line: 6>()
      4 import pandas as pd
      5 import plotly.express as px
----> 6 INDIA_MAP=json.load(open('/content/states_india.geojson','r'))
      7 INDIA_DATA=pd.read_csv('India-Population1.csv')
      8 MAP=px.choropleth(INDIA_DATA,geojson=INDIA_MAP,

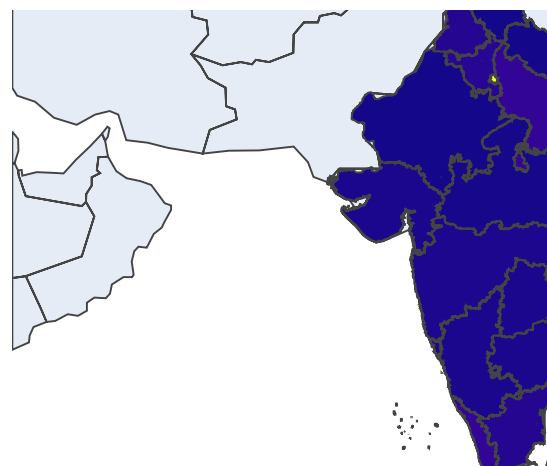
FileNotFoundError: [Errno 2] No such file or directory: '/content
/states_india.geojson'
```

SEARCH STACK OVERFLOW

```
# 1> Display a Choropleth Map
# i> To show variation of Density in different Indian states
import json
import pandas as pd
import plotly.express as px
INDIA_MAP=json.load(open('/content/states_india.geojson','r'))
INDIA_DATA=pd.read_csv('India-Population1.csv')
MAP=px.choropleth(INDIA_DATA,geojson=INDIA_MAP,
locations='State',
featureidkey="properties.st_nm",
color='Density',scope='asia',fitbounds='locations')
MAP.show()
```



! 0s completed at 10:07 PM



```
# 2. i> Load the above given dataset from plotly library using px.data.tips()
import plotly.express as px
DataTips = px.data.tips()
display(DataTips)
```

	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

```
# ii> Display a LINE plot for 'tip' Vs. 'total_bill' classified according to 'sex' and 'day'
# a. colour of the plots should be different for different type of 'sex'
import plotly.express as px
```

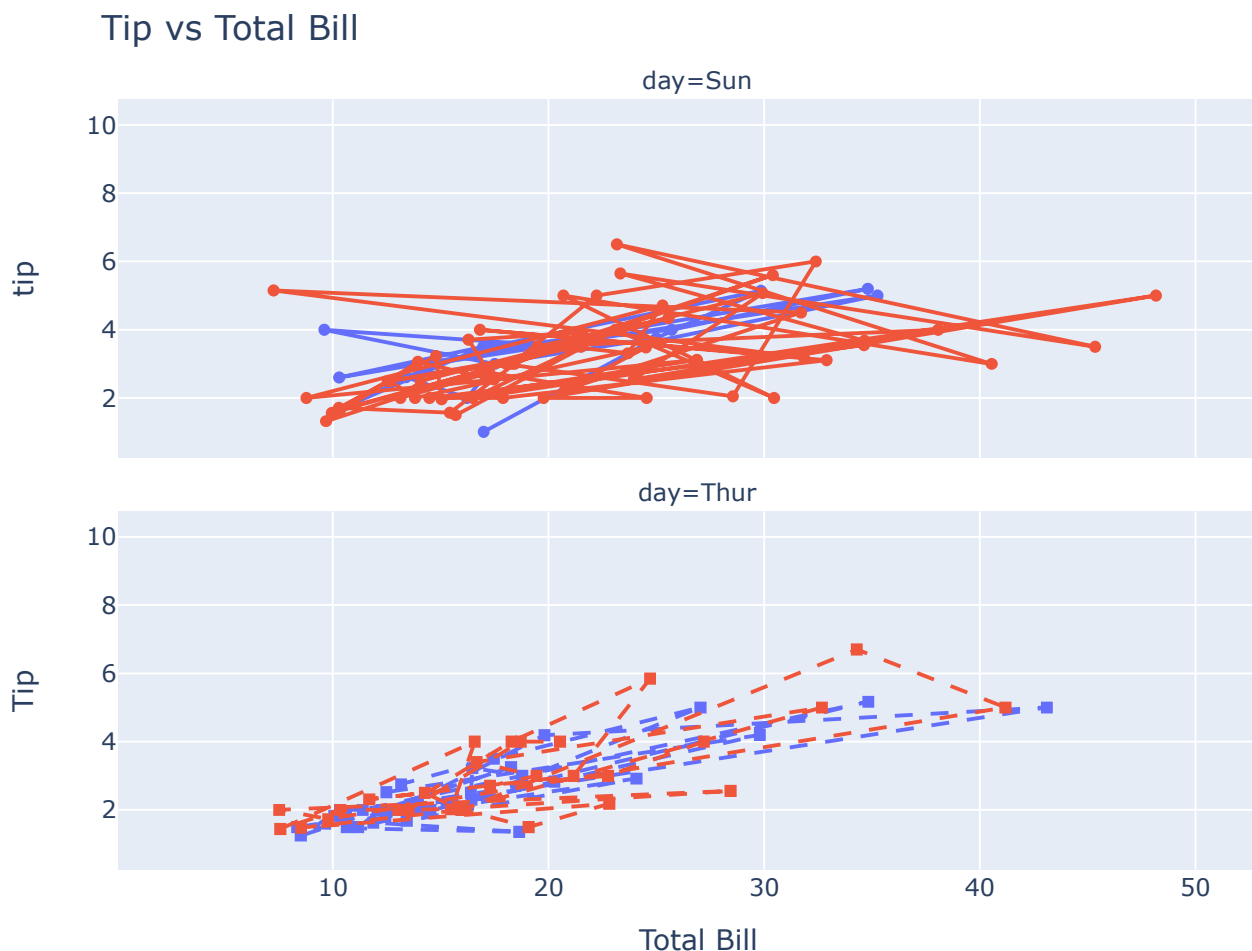
```

data = px.data.tips()
fig = px.line(data, x='total_bill', y='tip', color='sex', line_group='sex',
              facet_col='day', facet_col_wrap=2,
              markers=True, symbol='day', line_dash='day')

fig.update_layout(title='Tip vs Total Bill',
                  xaxis_title='Total Bill',
                  yaxis_title='Tip',
                  legend_title='Sex')

fig.show()

```



```

# ii> Display a LINE plot for 'tip' Vs. 'total_bill' classified according to 'sex' and 'day'
# b. marker type and plot style should be different for different 'day'
import plotly.express as px
data = px.data.tips()
fig = px.line(data, x='total_bill', y='tip', color='day', line_group='sex',
              facet_col='day', facet_col_wrap=2,
              markers=True, symbol='day', line_dash='day')

fig.update_layout(title='Tip vs Total Bill',
                  xaxis_title='Total Bill',

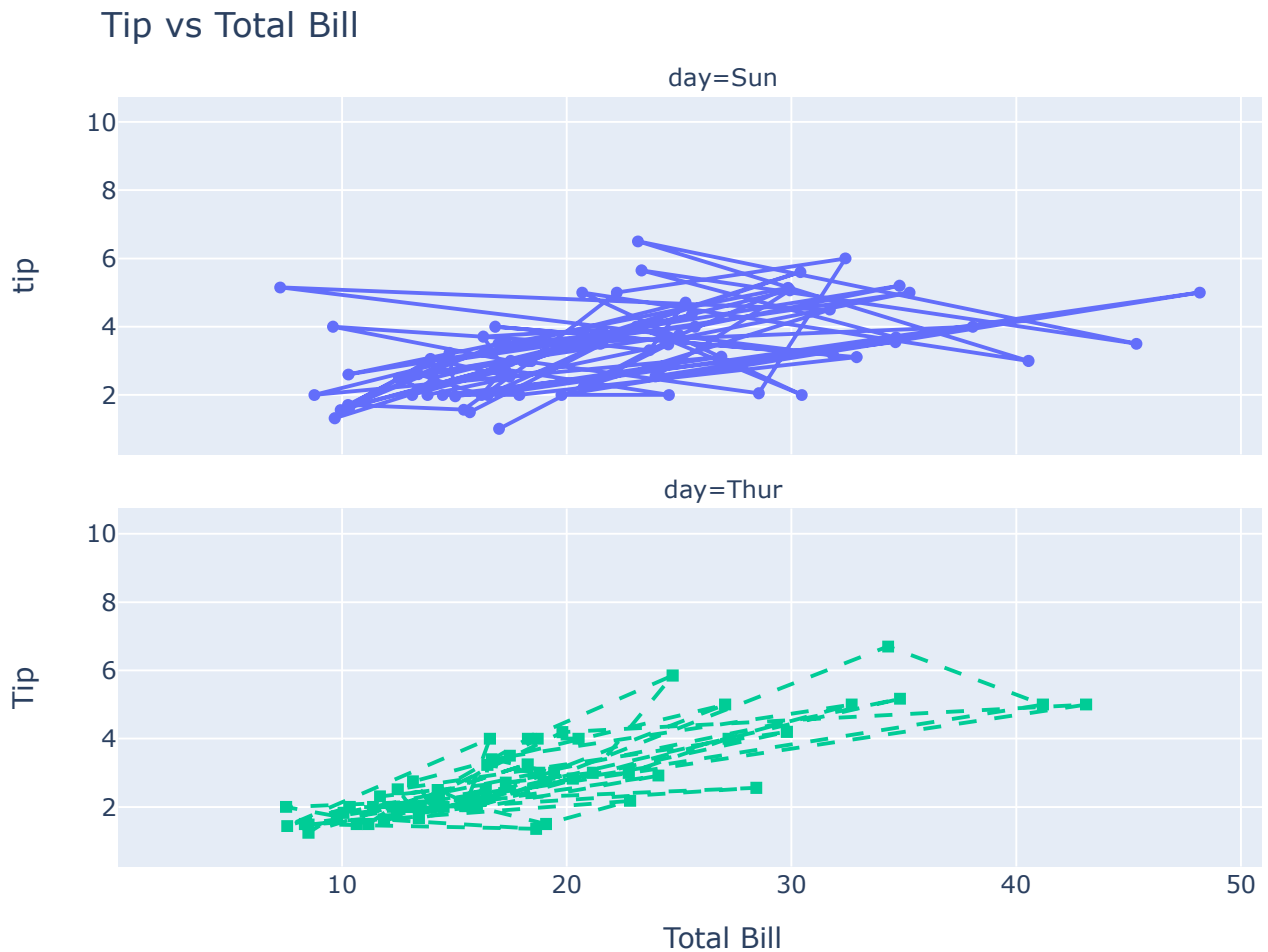
```

```

xaxis_title= 'Total Bill' ,
yaxis_title='Tip',
legend_title='day')

```

```
fig.show()
```



```

# iii. Display a bubble chart for 'total_bill' Vs. 'size'
# a. bubble colour should be classified according to 'day'

```

```

import plotly.express as px
data = px.data.tips()
fig = px.scatter(data, x='total_bill', y='size', color='day',
                 facet_col='day', facet_col_wrap=2,
                 symbol='day')

```

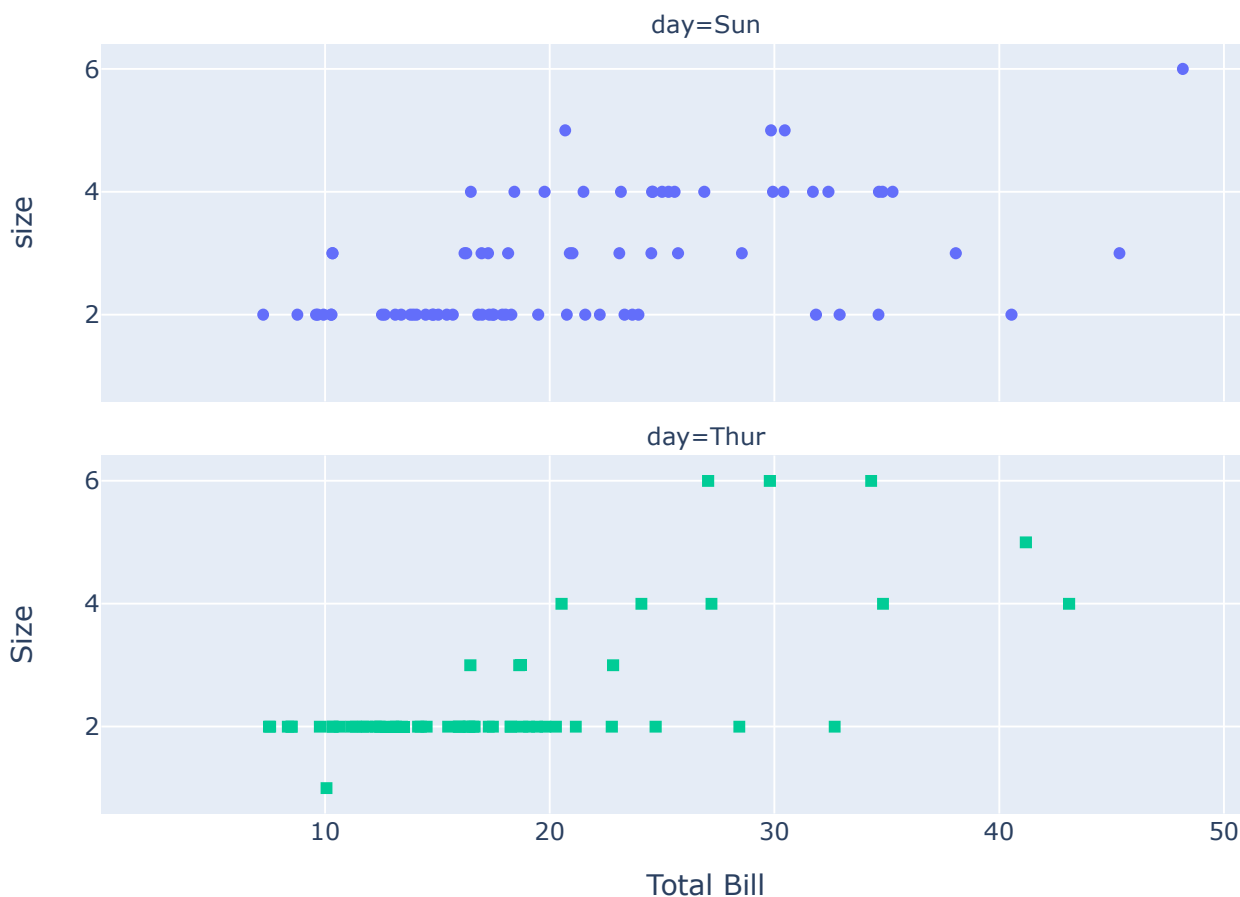
```

fig.update_layout(title='Size vs Total Bill',
                  xaxis_title='Total Bill',
                  yaxis_title='Size',
                  legend_title='Day')

```

```
fig.show()
```

Size vs Total Bill

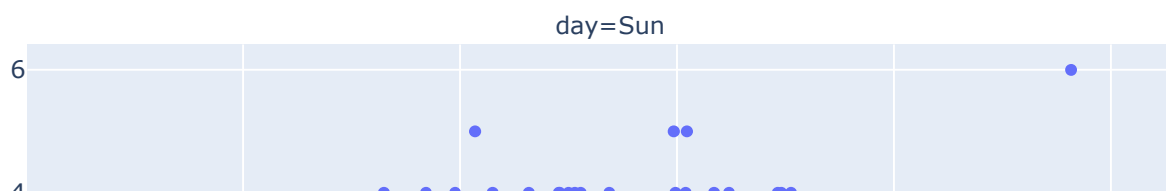


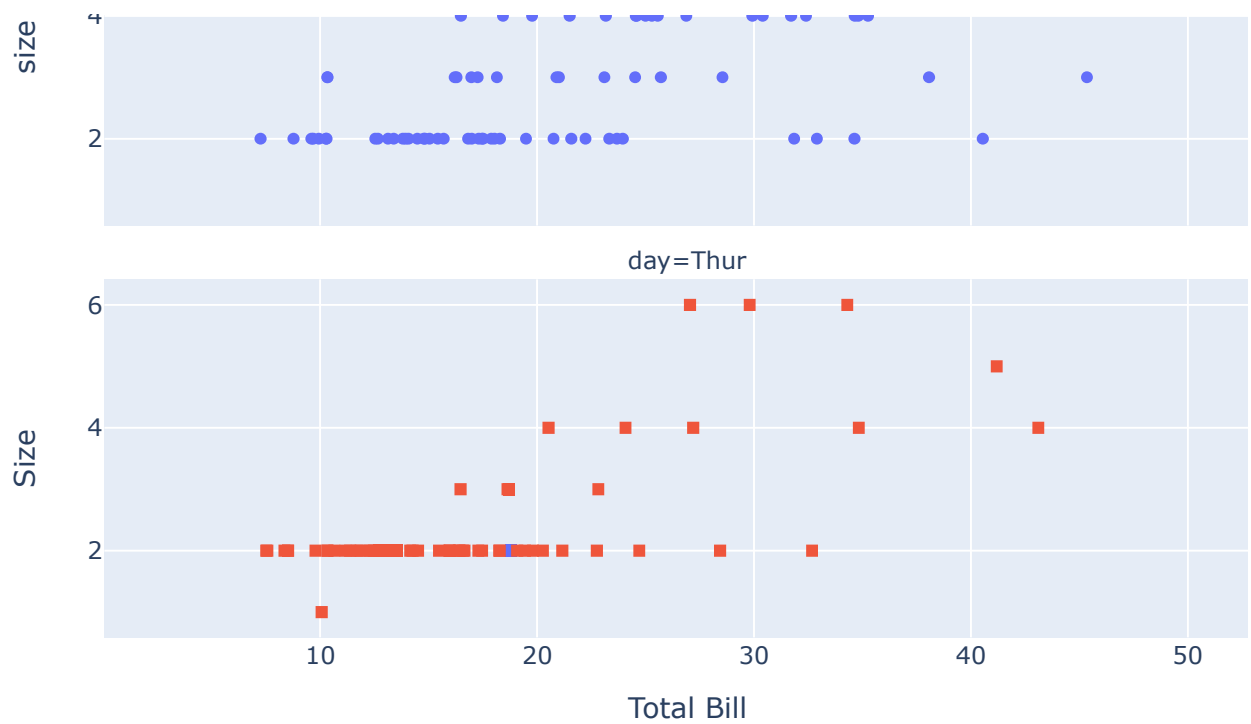
```
# iii. Display a bubble chart for 'total_bill' Vs. 'size'
# b. bubble size should be classified according to 'time'
import plotly.express as px
data = px.data.tips()
fig = px.scatter(data, x='total_bill', y='size', color='time',
                 facet_col='day', facet_col_wrap=2, symbol='day')
```

```
fig.update_layout(title='Total Bill vs Size',
                  xaxis_title='Total Bill',
                  yaxis_title='Size',
                  legend_title='day')
```

```
fig.show()
```

Total Bill vs Size

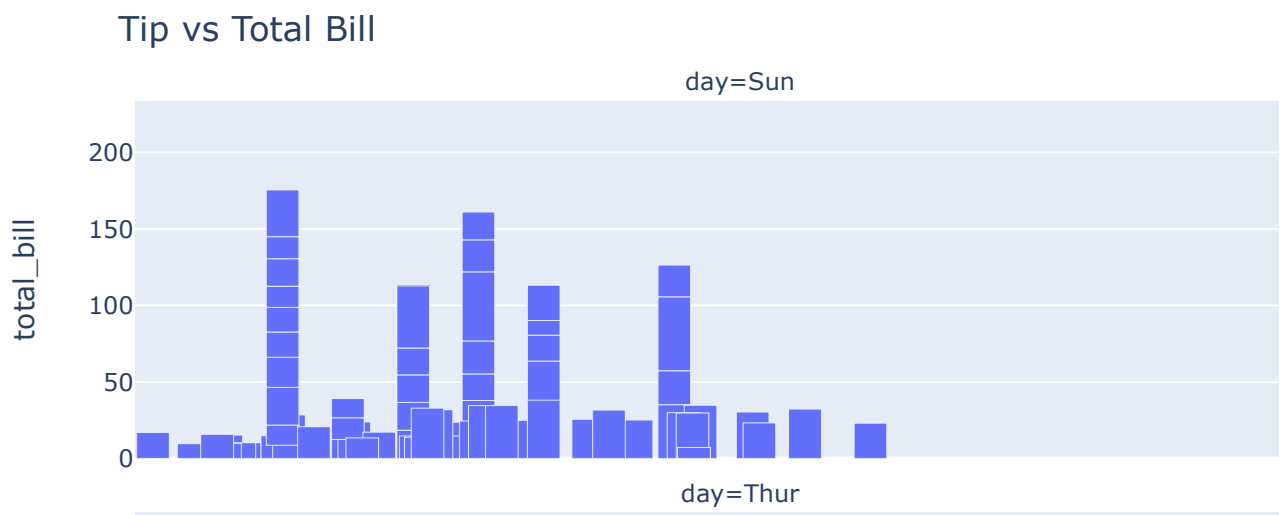


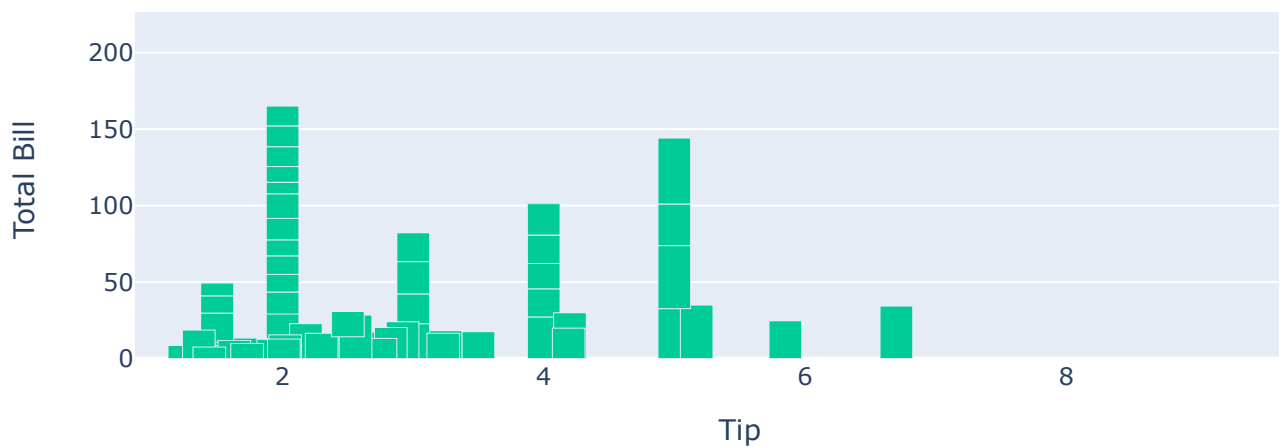


```
# iv. Display a BAR plot for tip Vs. 'total_bill' (classified according to 'day')
import plotly.express as px
data = px.data.tips()
fig = px.bar(data, x='tip', y='total_bill', color='day',
             facet_col='day', facet_col_wrap=2
             )

fig.update_layout(title='Tip vs Total Bill',
                  xaxis_title='Tip',
                  yaxis_title='Total Bill',
                  legend_title='day')
fig.update_traces(width=0.25)

fig.show()
```





```
# Draw graph for e^x vs x
import plotly.express as px
import numpy as np
x = [m for m in range(1,20)]
y = [np.exp(m) for m in range(1,20)]
Plot1 = px.line(x=x, y=y, title='e^x Vs. X',
labels=dict(x='x', y='e^x'),
range_x=(1,20), range_y=(1,90),
height=500, width=500, markers=True)

Plot1.update_traces(textposition='top left',
marker = dict(size =10, opacity = 0.8, color='red'),
line = dict(dash = 'dashdot', width = 2, color='red'))

Plot1.update_layout(plot_bgcolor='light grey',
paper_bgcolor='pink',
axis_showgrid=False, yaxis_showgrid=False,
axis_showline=True, axis_linecolor='black',
axis_showline=True, axis_linecolor='black',
axis_mirror=True, yaxis_mirror=True,
axis_ticks='inside', yaxis_ticks='inside',
axis_tickmode='linear', axis_dtick=1,
axis_tickmode='linear', axis_dtick=10,
showlegend=True,
legend_title_font_color="black",
legend_title_text='e^x Vs. X',
font_family="Courier New",
font_size=15,
font_color="black",
title_font_family="Times New Roman",
title_font_color="red",
title_x=0.5)

Plot1.show()
```

e^x Vs. X

