

Import

```
In [1]: import pandas as pd
import numpy as np
import chart_studio.plotly as py
import seaborn as sns
import cufflinks as cf
import plotly.express as px

%matplotlib inline
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
init_notebook_mode(connected=True)
cf.go_offline()
```

Basics

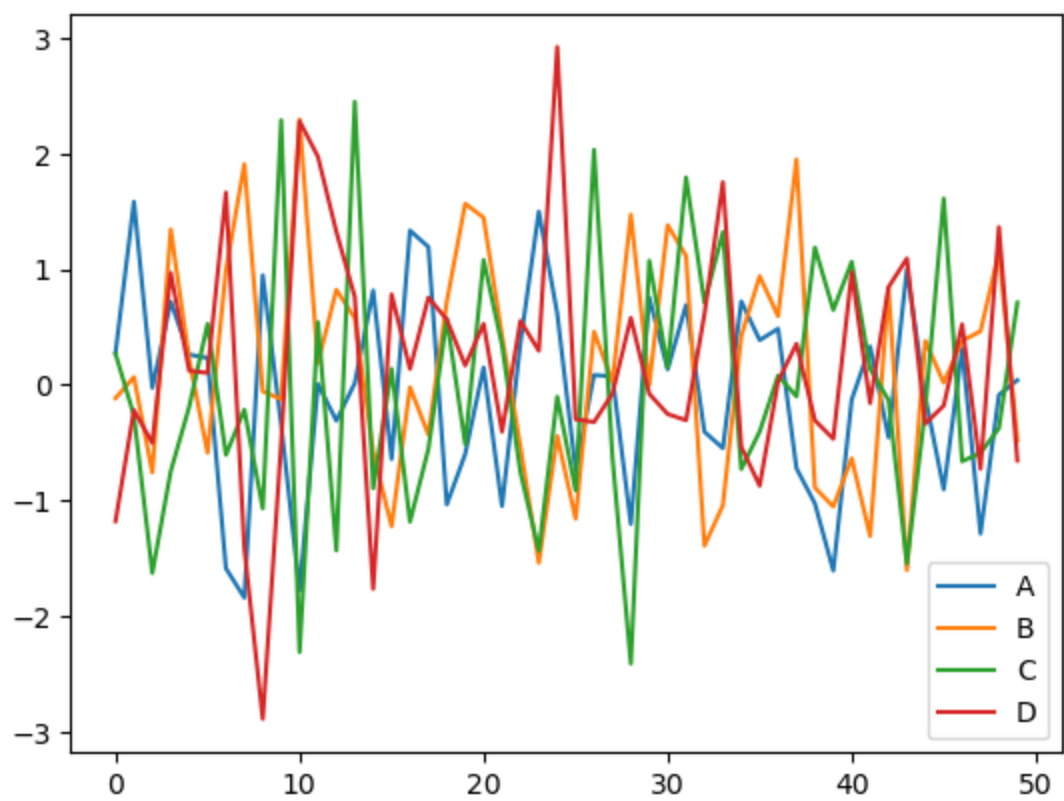
```
In [2]: arr_1 = np.random.randn(50, 4)
df_1 = pd.DataFrame(arr_1, columns=['A', 'B', 'C', 'D'])
df_1.head()
```

```
Out[2]:
```

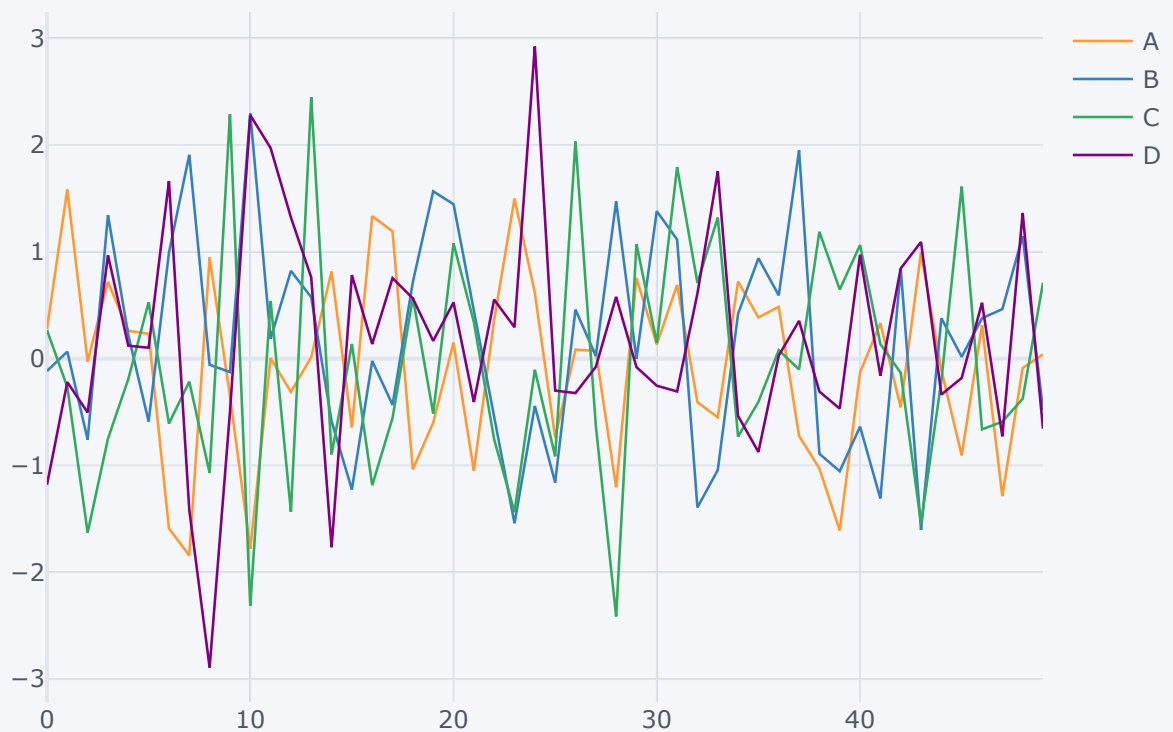
	A	B	C	D
0	0.276980	-0.115776	0.267245	-1.180181
1	1.583341	0.065915	-0.265451	-0.219958
2	-0.026691	-0.758970	-1.628598	-0.503853
3	0.719383	1.343952	-0.748580	0.968329
4	0.260953	0.221135	-0.192127	0.121407

```
In [3]: df_1.plot()
```

```
Out[3]: <AxesSubplot:>
```



```
In [4]: df_l.iplot()
```



[Export to plot.ly »](#)

Line Plots

```
In [5]: import plotly.graph_objects as go
```

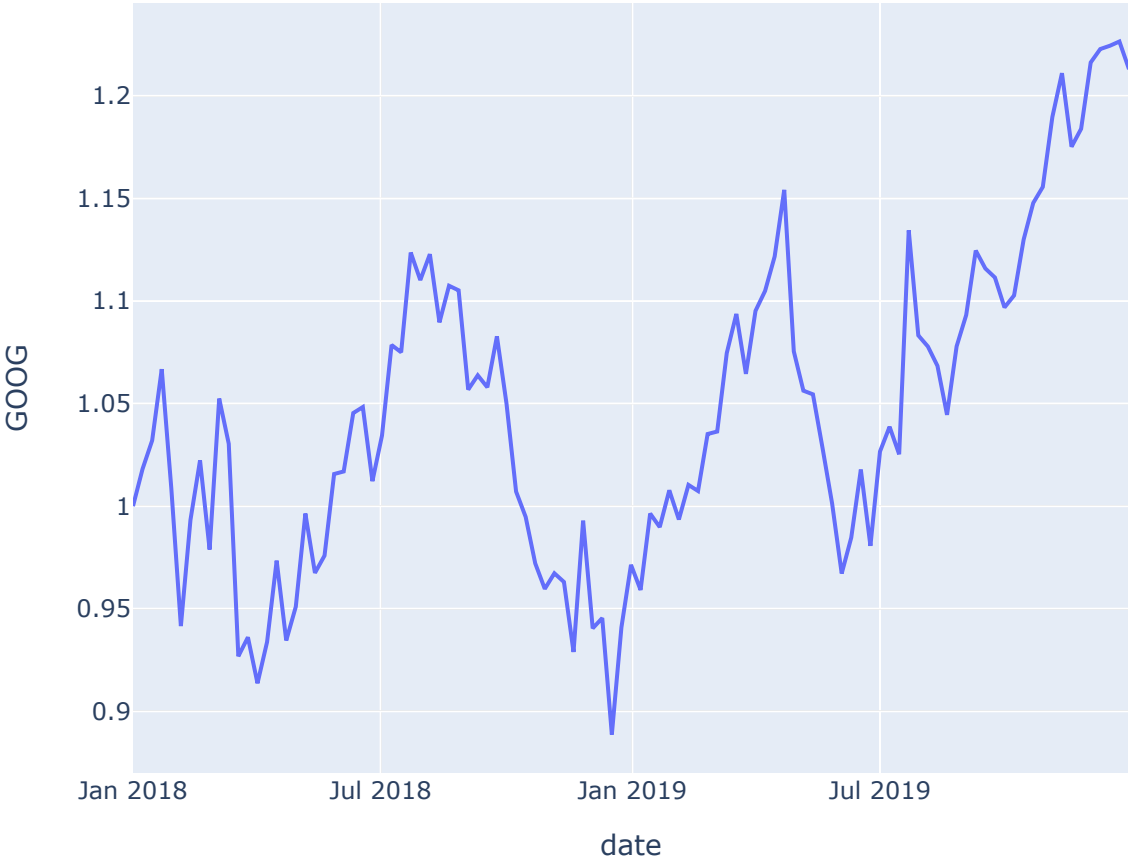
```
In [6]: df_stocks = px.data.stocks()  
df_stocks
```

Out[6]:

	date	GOOG	AAPL	AMZN	FB	NFLX	MSFT
0	2018-01-01	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1	2018-01-08	1.018172	1.011943	1.061881	0.959968	1.053526	1.015988
2	2018-01-15	1.032008	1.019771	1.053240	0.970243	1.049860	1.020524
3	2018-01-22	1.066783	0.980057	1.140676	1.016858	1.307681	1.066561
4	2018-01-29	1.008773	0.917143	1.163374	1.018357	1.273537	1.040708
...
100	2019-12-02	1.216280	1.546914	1.425061	1.075997	1.463641	1.720717
101	2019-12-09	1.222821	1.572286	1.432660	1.038855	1.421496	1.752239
102	2019-12-16	1.224418	1.596800	1.453455	1.104094	1.604362	1.784896
103	2019-12-23	1.226504	1.656000	1.521226	1.113728	1.567170	1.802472
104	2019-12-30	1.213014	1.678000	1.503360	1.098475	1.540883	1.788185

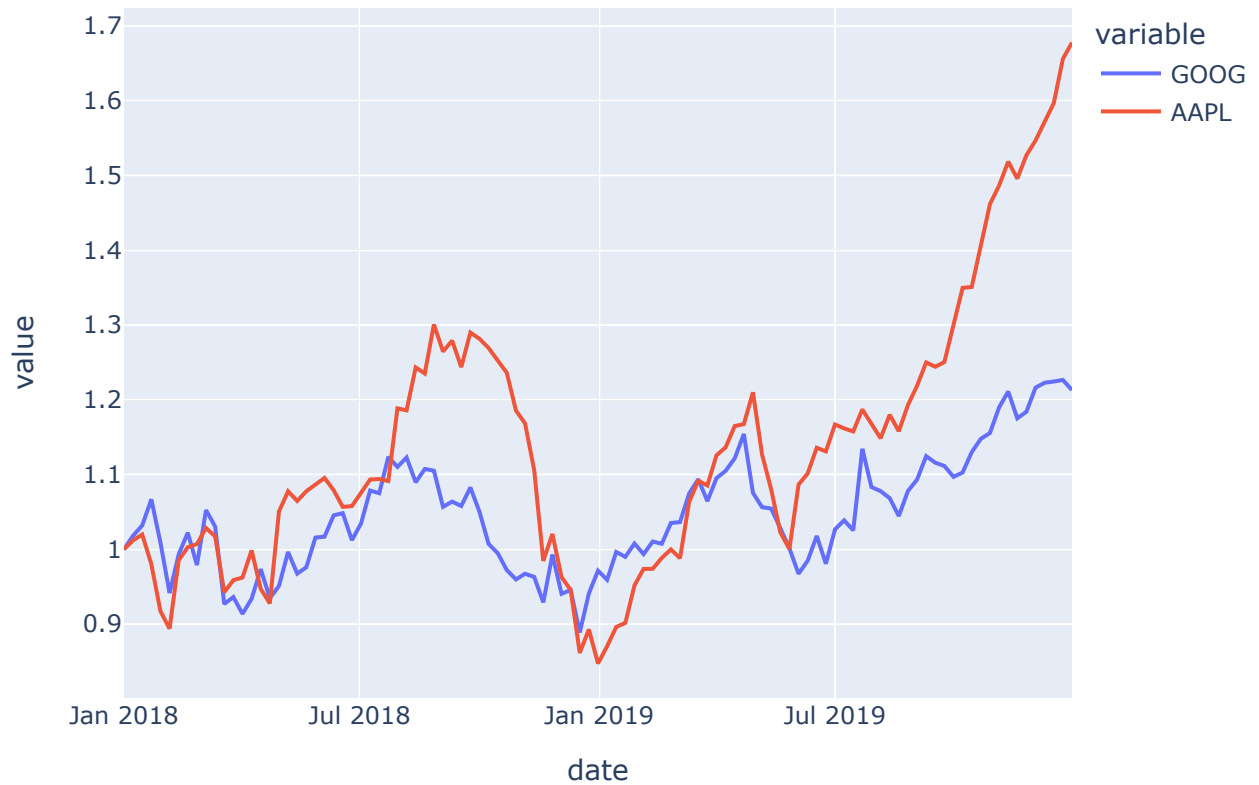
105 rows × 7 columns

```
In [7]: px.line(df_stocks, x='date', y='GOOG', labels = {'x': 'Date', 'y': 'Price'})
```



```
In [8]: px.line(df_stocks, x='date', y=['GOOG', 'AAPL'], labels = {'x':'Date', 'y':'Price'}, tit
```

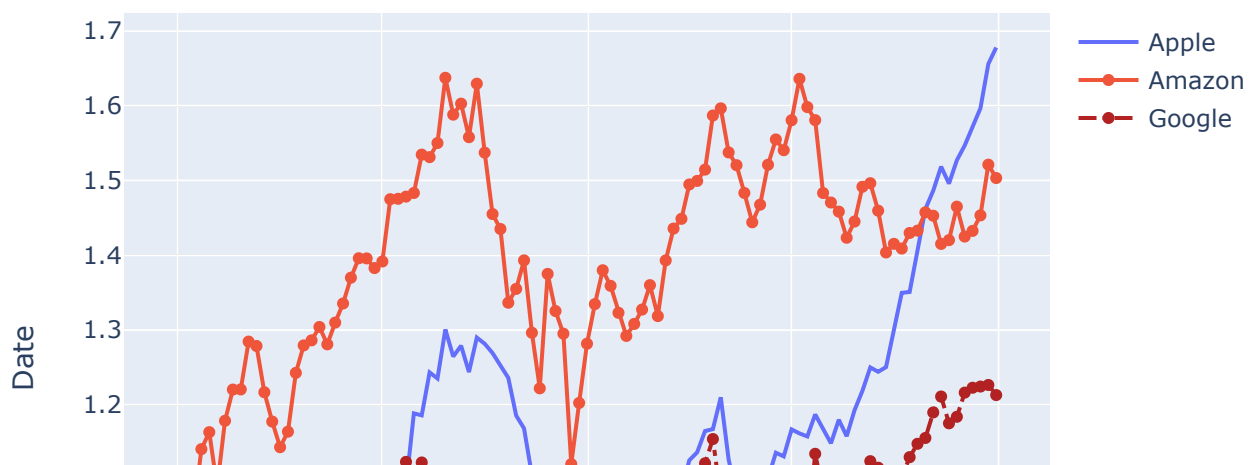
Apple vs Google



```
In [9]: fig = go.Figure()
fig.add_trace(go.Scatter(x=df_stocks.date, y=df_stocks.AAPL, mode='lines', name='Apple'))
fig.add_trace(go.Scatter(x=df_stocks.date, y=df_stocks.AMZN, mode='lines+markers', name='Amazon'))
fig.add_trace(go.Scatter(x=df_stocks.date, y=df_stocks.GOOG, mode='lines+markers', name='Google',
                        line = dict(color='firebrick', width=2, dash='dashdot'))))

fig.update_layout(title='Stock Price Data 2018 -2020', xaxis_title = 'Price', yaxis_titl
```

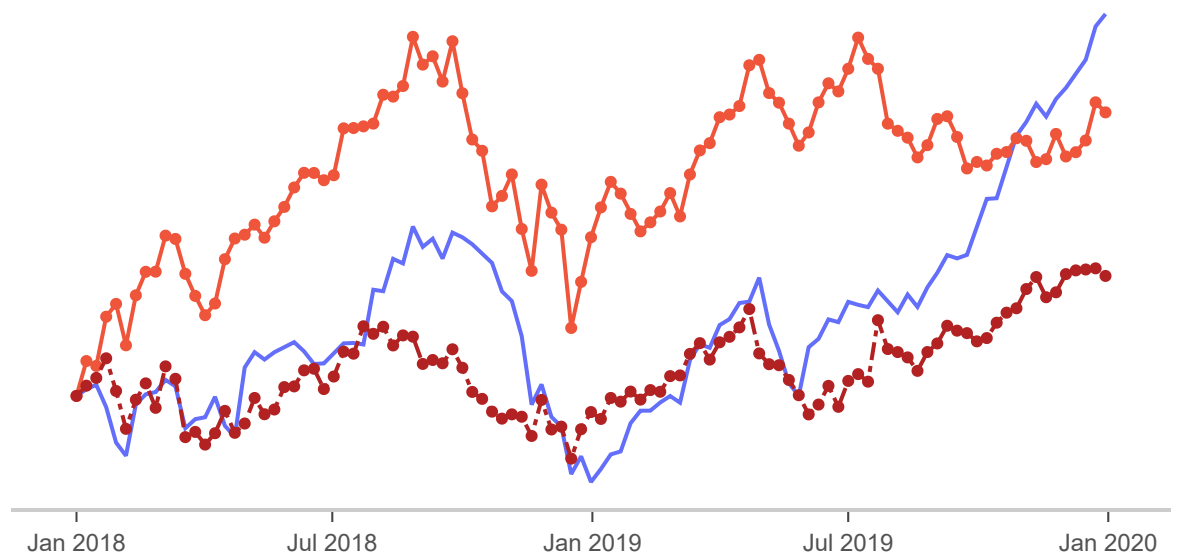
Stock Price Data 2018 -2020





```
In [10]: fig = go.Figure()
fig.add_trace(go.Scatter(x=df_stocks.date, y=df_stocks.AAPL, mode='lines', name='Apple'))
fig.add_trace(go.Scatter(x=df_stocks.date, y=df_stocks.AMZN, mode='lines+markers', name=
fig.add_trace(go.Scatter(x=df_stocks.date, y=df_stocks.GOOG, mode='lines+markers', name=
                    line = dict(color='firebrick', width=2, dash='dashdot'))))

fig.update_layout(
xaxis=dict(
showline=True, showgrid=False, showticklabels = True, linecolor='rgb(204, 204, 204)',
linewidth=2, ticks = 'outside',
tickfont = dict(family = 'Arial', size = 12, color = 'rgb(82, 82, 82)',
),
),
yaxis=dict(
showline=False, showgrid=False, showticklabels = False, zeroline = False),
autosize=False,
margin=dict(
autoexpand = False, l=100, r=20, t=110),
showlegend=False, plot_bgcolor='white')
```



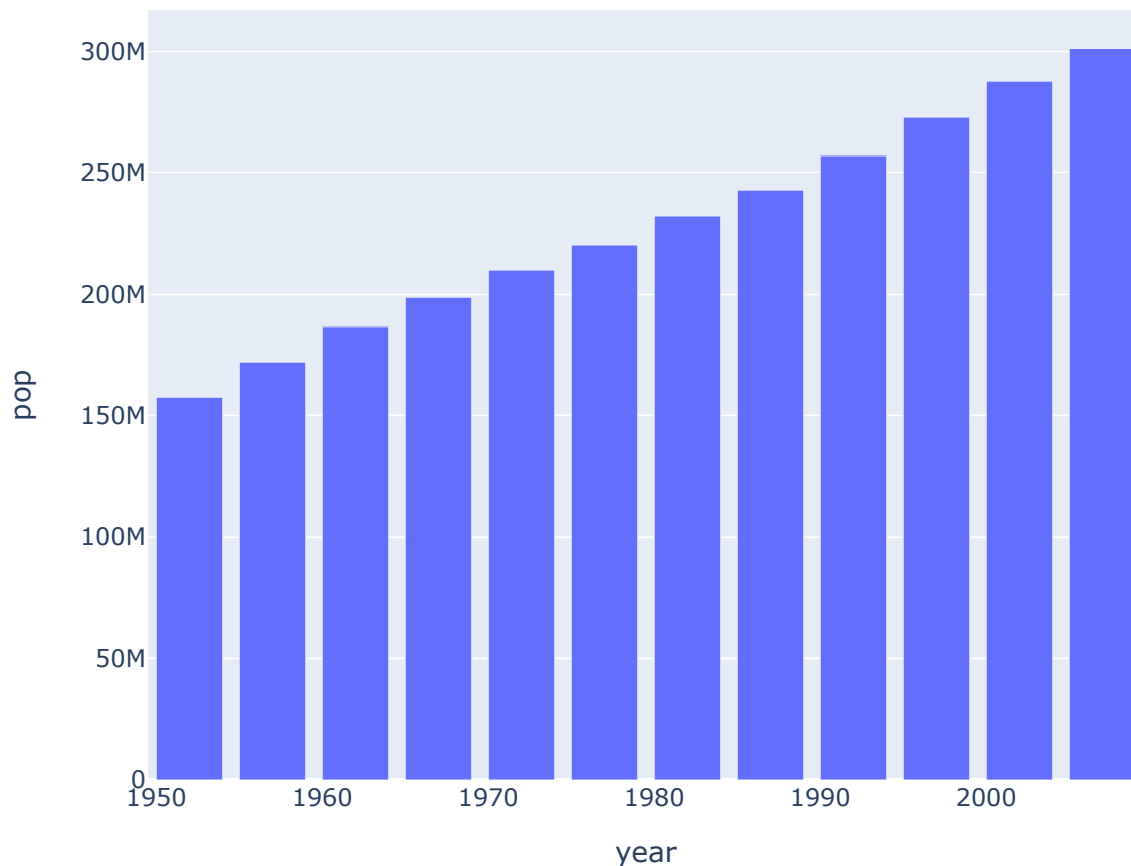
Bar Charts

```
In [11]: df_us = px.data.gapminder().query('country=="United States"')
df_us
```

Out[11]:

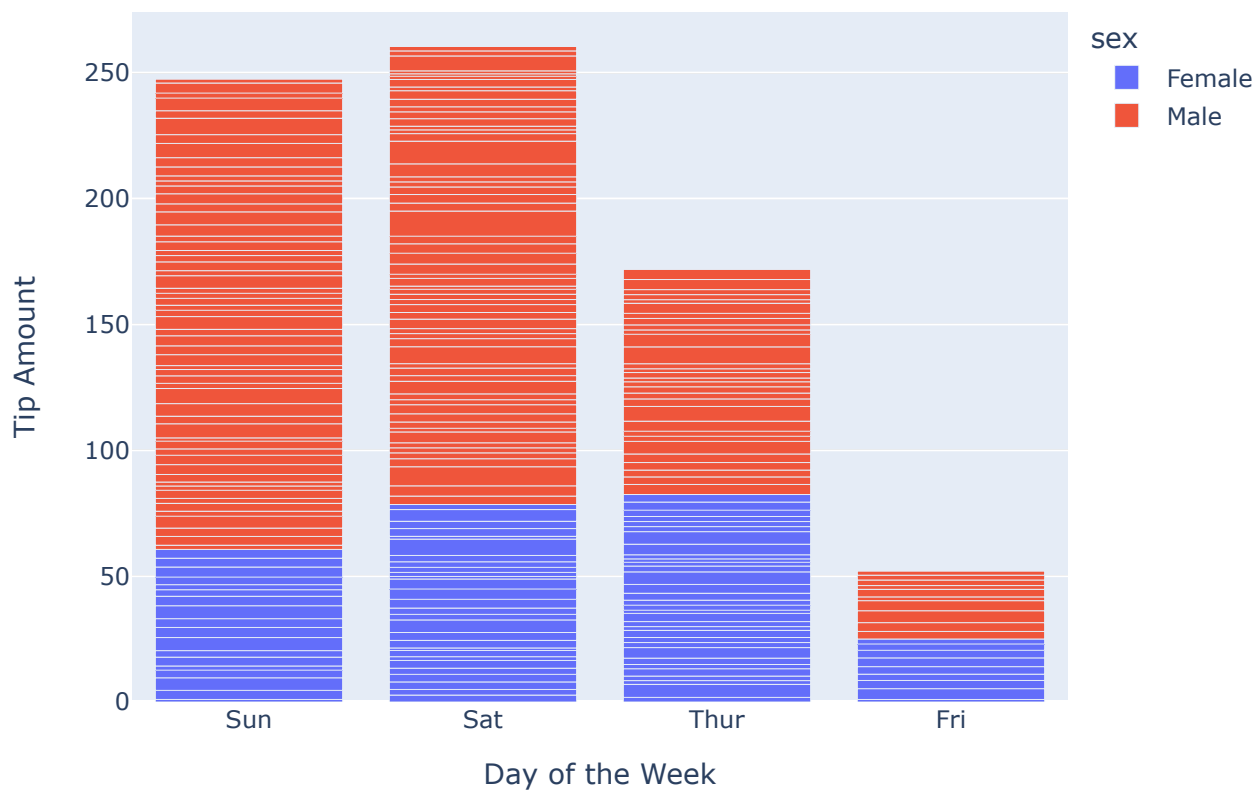
	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
1608	United States	Americas	1952	68.440	157553000	13990.48208	USA	840
1609	United States	Americas	1957	69.490	171984000	14847.12712	USA	840
1610	United States	Americas	1962	70.210	186538000	16173.14586	USA	840
1611	United States	Americas	1967	70.760	198712000	19530.36557	USA	840
1612	United States	Americas	1972	71.340	209896000	21806.03594	USA	840
1613	United States	Americas	1977	73.380	220239000	24072.63213	USA	840
1614	United States	Americas	1982	74.650	232187835	25009.55914	USA	840
1615	United States	Americas	1987	75.020	242803533	29884.35041	USA	840
1616	United States	Americas	1992	76.090	256894189	32003.93224	USA	840
1617	United States	Americas	1997	76.810	272911760	35767.43303	USA	840
1618	United States	Americas	2002	77.310	287675526	39097.09955	USA	840
1619	United States	Americas	2007	78.242	301139947	42951.65309	USA	840

```
In [12]: px.bar(df_us, x='year', y='pop')
```

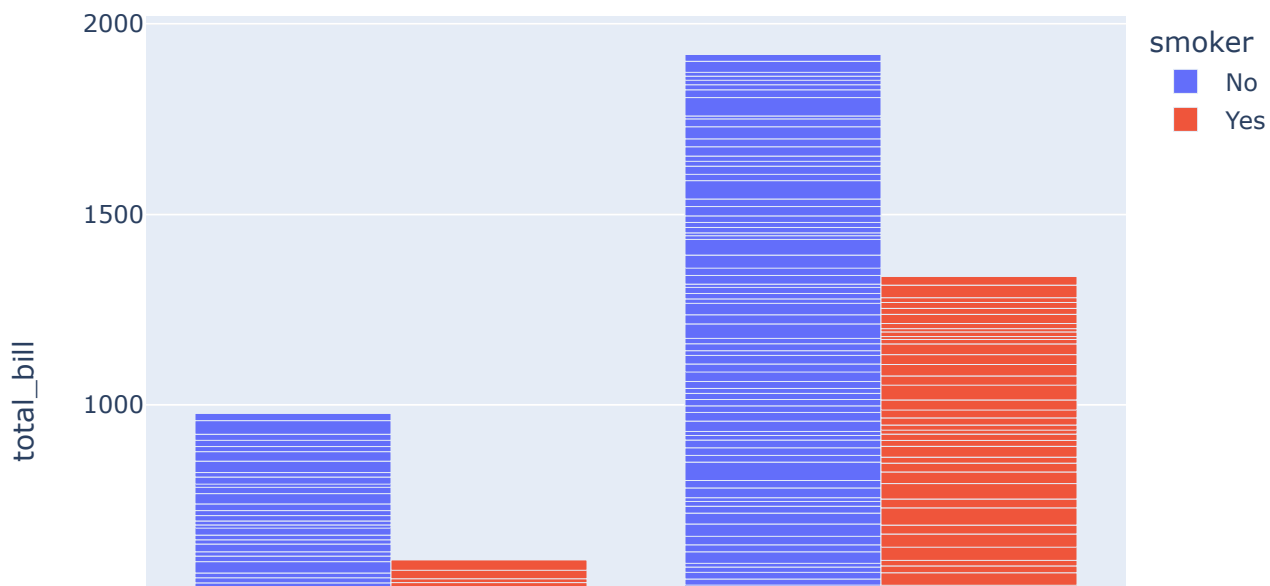


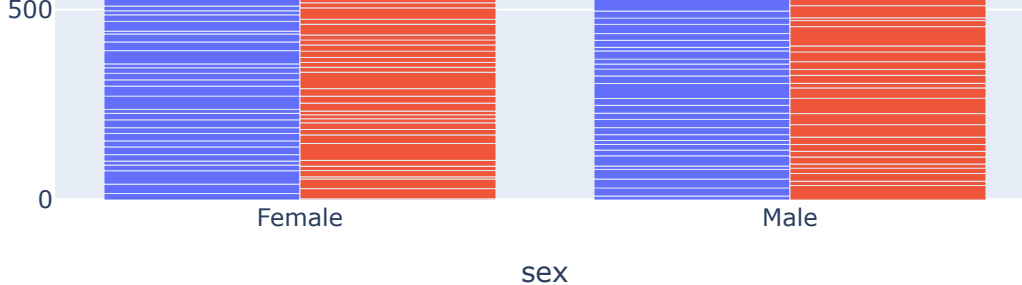
```
In [13]: df_tips = px.data.tips()
px.bar(df_tips, x='day', y='tip', color = 'sex', title = 'Tips by Sex on Each Day',
       labels={'tip': 'Tip Amount', 'day': 'Day of the Week'})
```

Tips by Sex on Each Day



```
In [14]: px.bar(df_tips, x='sex', y='total_bill', color = 'smoker',
               barmode = 'group')
```



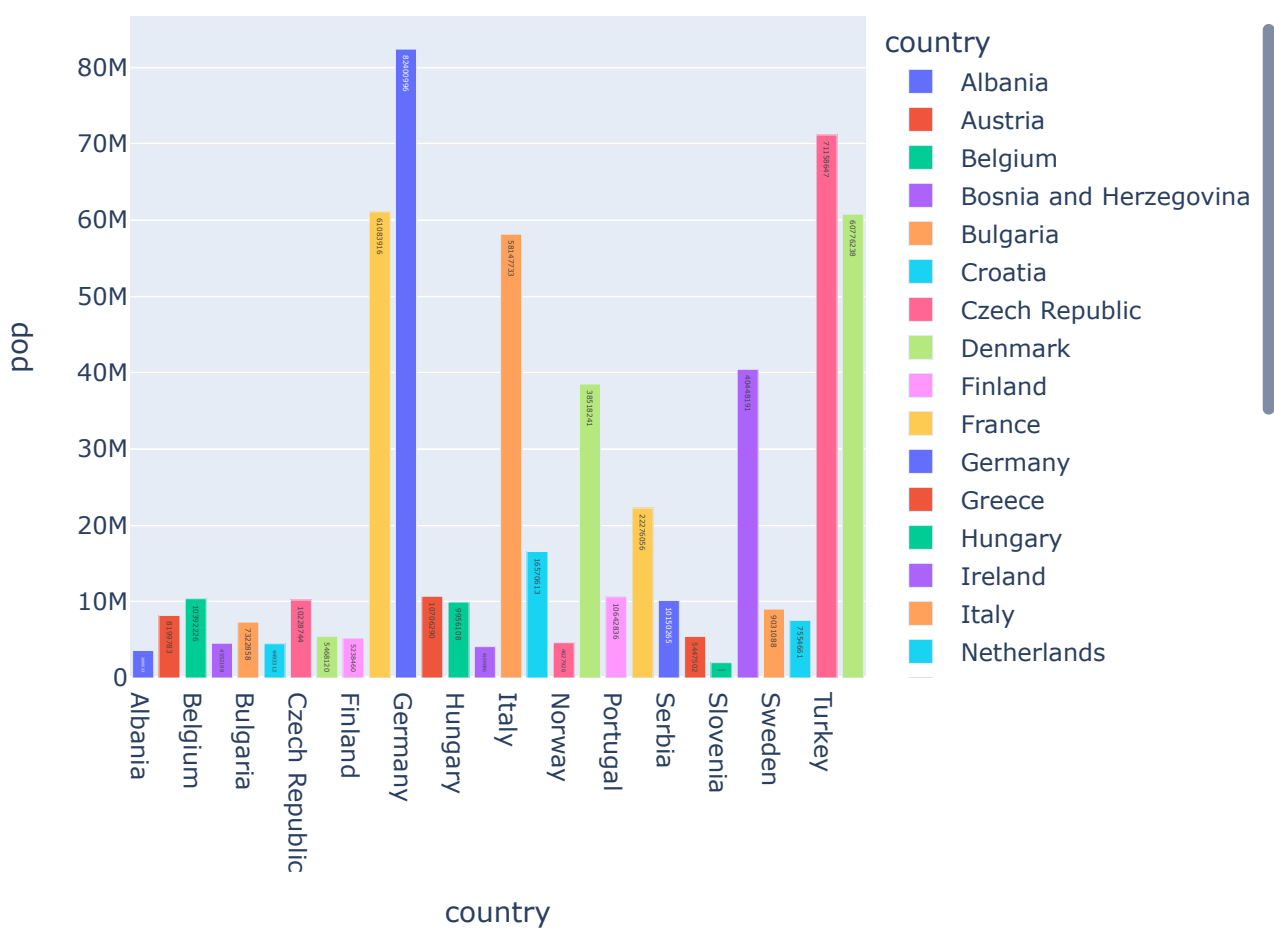


```
In [15]: df_europe = px.data.gapminder().query('continent == "Europe" and year==2007 and pop>2.e6')
df_europe
```

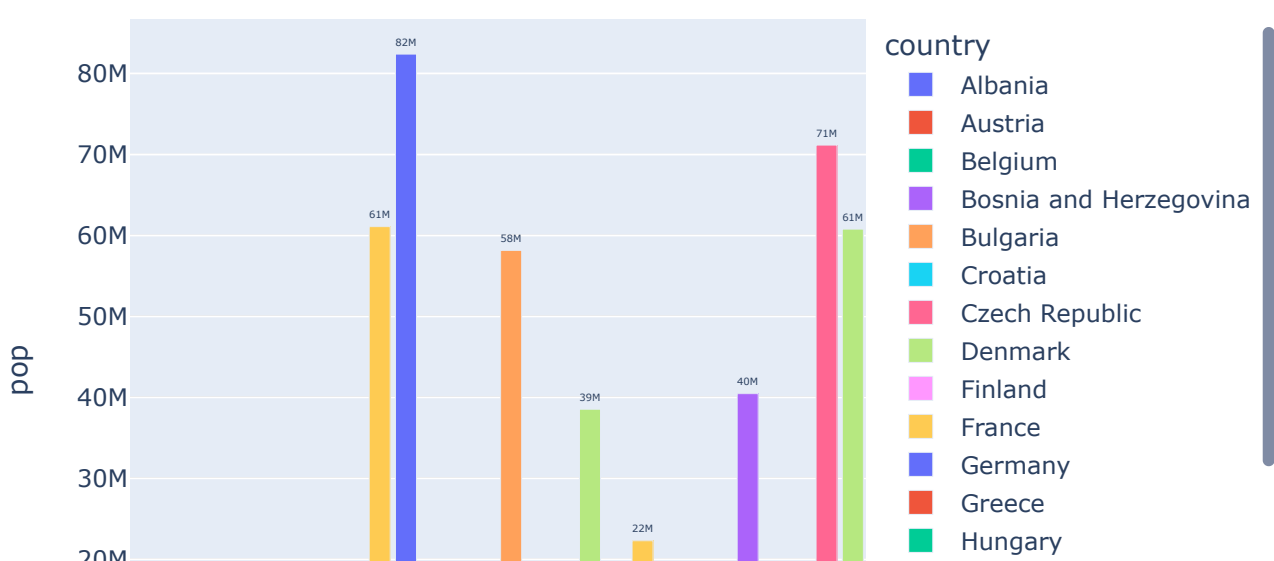
Out[15]:

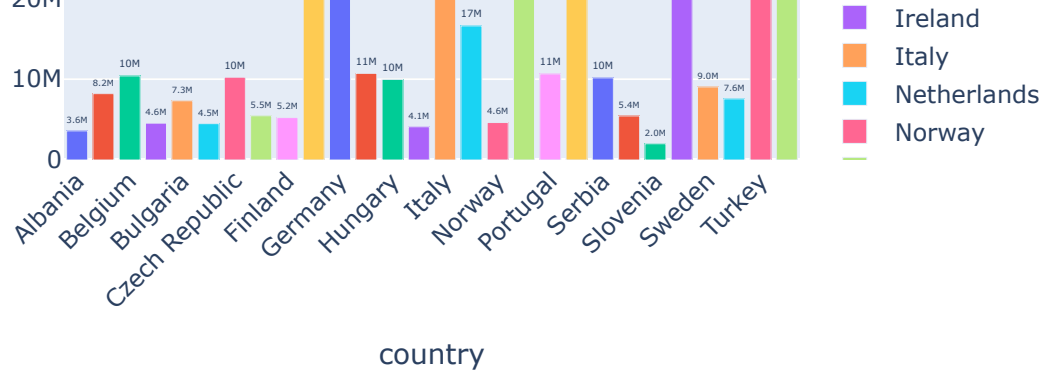
	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
23	Albania	Europe	2007	76.423	3600523	5937.029526	ALB	8
83	Austria	Europe	2007	79.829	8199783	36126.492700	AUT	40
119	Belgium	Europe	2007	79.441	10392226	33692.605080	BEL	56
155	Bosnia and Herzegovina	Europe	2007	74.852	4552198	7446.298803	BIH	70
191	Bulgaria	Europe	2007	73.005	7322858	10680.792820	BGR	100
383	Croatia	Europe	2007	75.748	4493312	14619.222720	HRV	191
407	Czech Republic	Europe	2007	76.486	10228744	22833.308510	CZE	203
419	Denmark	Europe	2007	78.332	5468120	35278.418740	DNK	208
527	Finland	Europe	2007	79.313	5238460	33207.084400	FIN	246
539	France	Europe	2007	80.657	61083916	30470.016700	FRA	250
575	Germany	Europe	2007	79.406	82400996	32170.374420	DEU	276
599	Greece	Europe	2007	79.483	10706290	27538.411880	GRC	300
683	Hungary	Europe	2007	73.338	9956108	18008.944440	HUN	348
755	Ireland	Europe	2007	78.885	4109086	40675.996350	IRL	372
779	Italy	Europe	2007	80.546	58147733	28569.719700	ITA	380
1091	Netherlands	Europe	2007	79.762	16570613	36797.933320	NLD	528
1151	Norway	Europe	2007	80.196	4627926	49357.190170	NOR	578
1235	Poland	Europe	2007	75.563	38518241	15389.924680	POL	616
1247	Portugal	Europe	2007	78.098	10642836	20509.647770	PRT	620
1283	Romania	Europe	2007	72.476	22276056	10808.475610	ROU	642
1343	Serbia	Europe	2007	74.002	10150265	9786.534714	SRB	688
1379	Slovak Republic	Europe	2007	74.663	5447502	18678.314350	SVK	703
1391	Slovenia	Europe	2007	77.926	2009245	25768.257590	SVN	705
1427	Spain	Europe	2007	80.941	40448191	28821.063700	ESP	724
1475	Sweden	Europe	2007	80.884	9031088	33859.748350	SWE	752
1487	Switzerland	Europe	2007	81.701	7554661	37506.419070	CHE	756
1583	Turkey	Europe	2007	71.777	71158647	8458.276384	TUR	792
1607	United Kingdom	Europe	2007	79.425	60776238	33203.261280	GBR	826


```
In [16]: fig = px.bar(df_europe, y='pop', x='country', text='pop', color='country')
fig
```



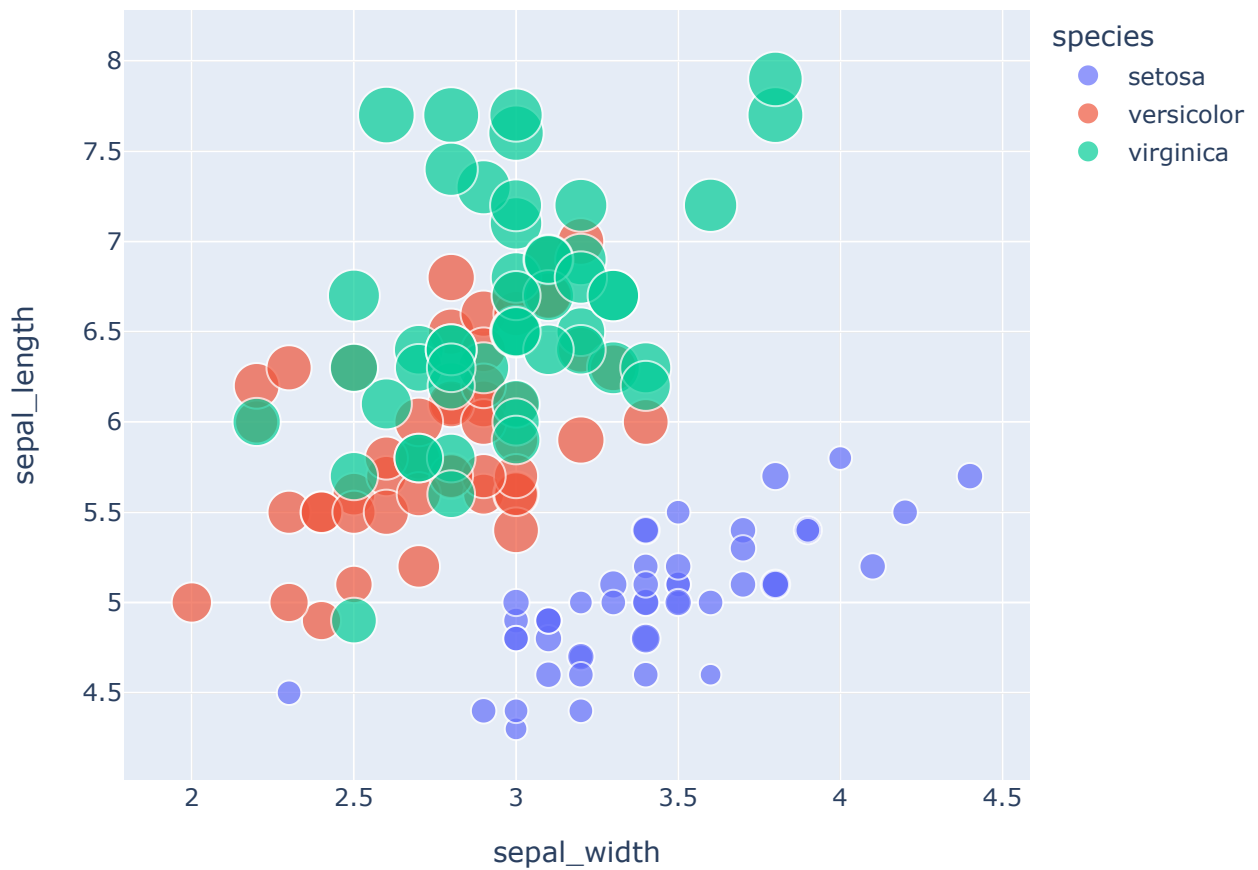
```
In [17]: fig = px.bar(df_europe, y='pop', x='country', text='pop', color='country')
fig.update_traces(texttemplate = '%{text:.2s}', textposition = 'outside')
fig.update_layout(uniformtext_minsize=8)
fig.update_layout(xaxis_tickangle=-45)
fig
```



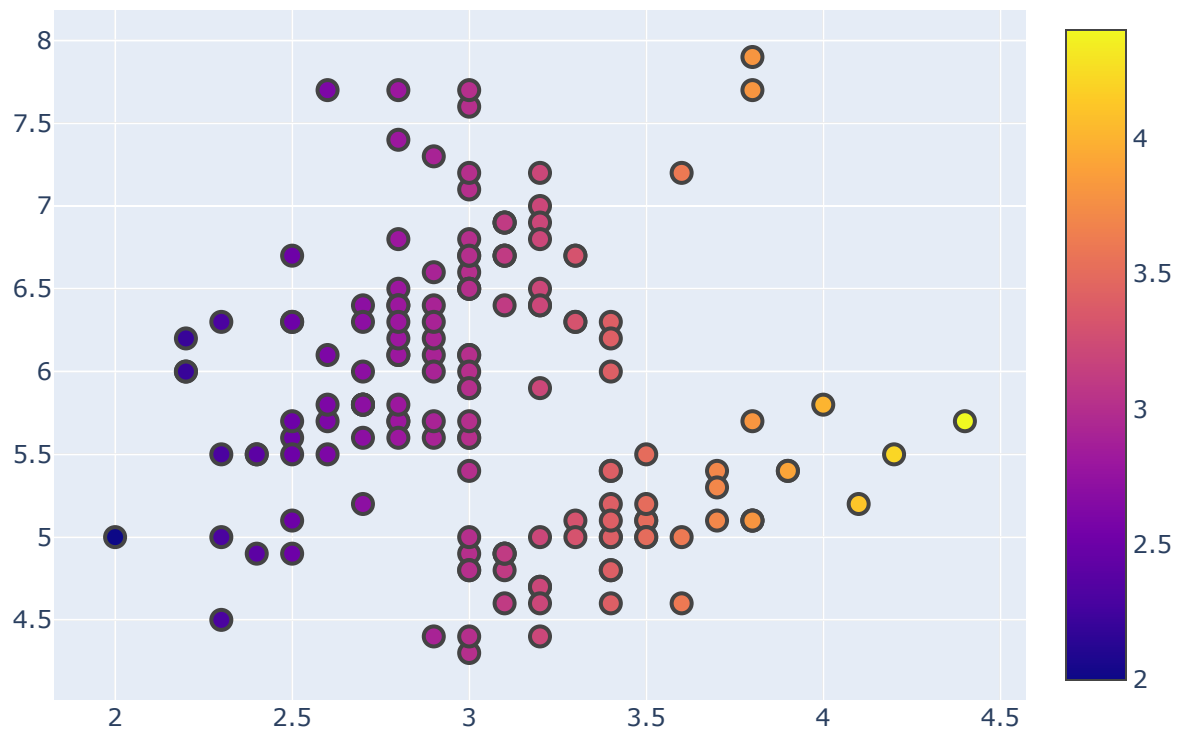


Scatter Plots

```
In [18]: df_iris = px.data.iris()
px.scatter(df_iris, x='sepal_width', y='sepal_length', color='species', size = 'petal_le
          hover_data = ['petal_width'])
```

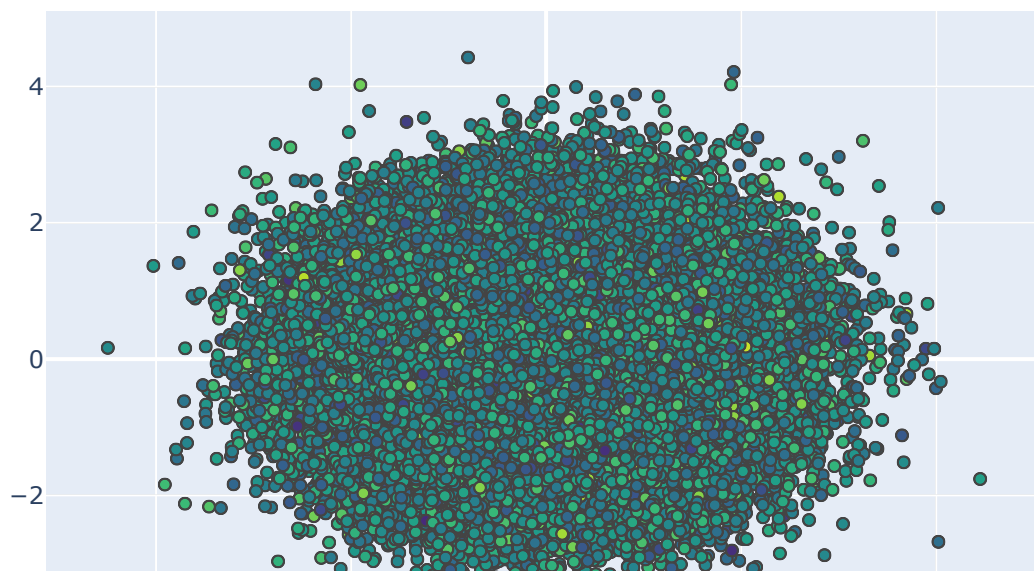


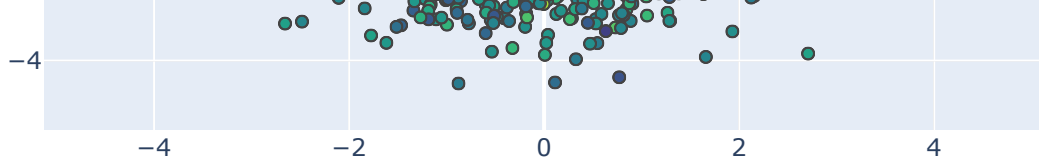
```
In [19]: fig = go.Figure()
fig.add_trace(go.Scatter(
x=df_iris.sepal_width, y=df_iris.sepal_length,
mode = 'markers',
marker_color = df_iris.sepal_width,
text = df_iris.species, marker=dict(showscale=True)))
fig.update_traces(marker_line_width=2, marker_size=10)
```



```
In [20]: fig = go.Figure(data=go.Scattergl(
x = np.random.randn(100000),
y = np.random.randn(100000),
mode = 'markers',
marker = dict(
color = np.random.randn(100000),
colorscale = 'Viridis',
line_width = 1)))
```

```
In [21]: fig
```





Pie Charts

```
In [22]: df_asia = px.data.gapminder().query("year==2007").query("continent=='Asia'")
df_asia
```

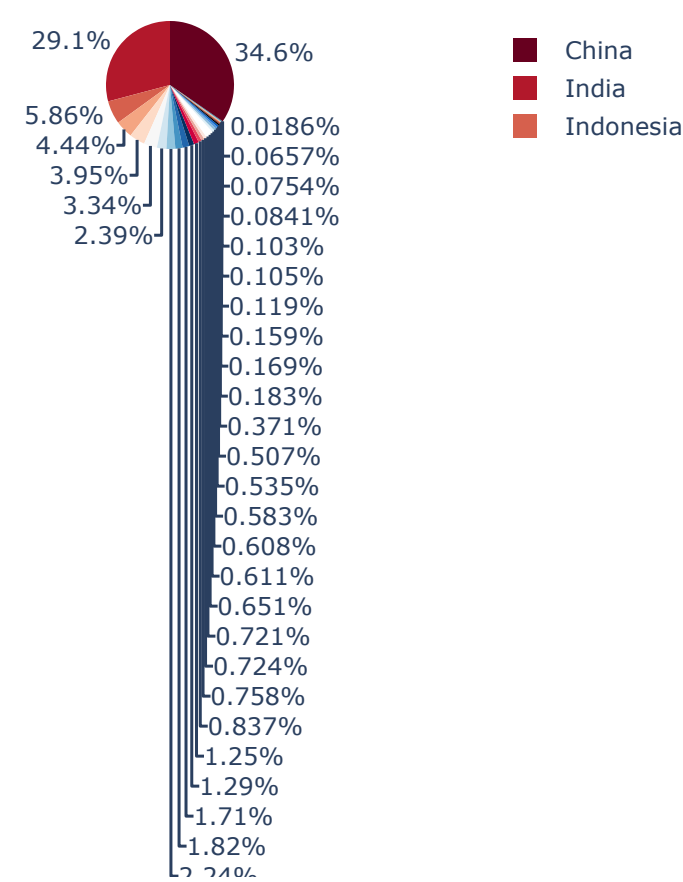
Out[22]:

	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
11	Afghanistan	Asia	2007	43.828	31889923	974.580338	AFG	4
95	Bahrain	Asia	2007	75.635	708573	29796.048340	BHR	48
107	Bangladesh	Asia	2007	64.062	150448339	1391.253792	BGD	50
227	Cambodia	Asia	2007	59.723	14131858	1713.778686	KHM	116
299	China	Asia	2007	72.961	1318683096	4959.114854	CHN	156
671	Hong Kong, China	Asia	2007	82.208	6980412	39724.978670	HKG	344
707	India	Asia	2007	64.698	1110396331	2452.210407	IND	356
719	Indonesia	Asia	2007	70.650	223547000	3540.651564	IDN	360
731	Iran	Asia	2007	70.964	69453570	11605.714490	IRN	364
743	Iraq	Asia	2007	59.545	27499638	4471.061906	IRQ	368
767	Israel	Asia	2007	80.745	6426679	25523.277100	ISR	376
803	Japan	Asia	2007	82.603	127467972	31656.068060	JPN	392
815	Jordan	Asia	2007	72.535	6053193	4519.461171	JOR	400
839	Korea, Dem. Rep.	Asia	2007	67.297	23301725	1593.065480	KOR	410
851	Korea, Rep.	Asia	2007	78.623	49044790	23348.139730	KOR	410
863	Kuwait	Asia	2007	77.588	2505559	47306.989780	KWT	414
875	Lebanon	Asia	2007	71.993	3921278	10461.058680	LBN	422
947	Malaysia	Asia	2007	74.241	24821286	12451.655800	MYS	458
1007	Mongolia	Asia	2007	66.803	2874127	3095.772271	MNG	496
1055	Myanmar	Asia	2007	62.069	47761980	944.000000	MMR	104
1079	Nepal	Asia	2007	63.785	28901790	1091.359778	NPL	524
1163	Oman	Asia	2007	75.640	3204897	22316.192870	OMN	512
1175	Pakistan	Asia	2007	65.483	169270617	2605.947580	PAK	586
1223	Philippines	Asia	2007	71.688	91077287	3190.481016	PHL	608
1319	Saudi Arabia	Asia	2007	72.777	27601038	21654.831940	SAU	682
1367	Singapore	Asia	2007	79.972	4553009	47143.179640	SGP	702
1439	Sri Lanka	Asia	2007	72.396	20378239	3970.095407	LKA	144

1499	Syria	Asia	2007	74.143	19314747	4184.548089	SYR	760
1511	Taiwan	Asia	2007	78.400	23174294	28718.276840	TWN	158
1535	Thailand	Asia	2007	70.616	65068149	7458.396327	THA	764
1655	Vietnam	Asia	2007	74.249	85262356	2441.576404	VNM	704
1667	West Bank and Gaza	Asia	2007	73.422	4018332	3025.349798	PSE	275
1679	Yemen, Rep.	Asia	2007	62.698	22211743	2280.769906	YEM	887

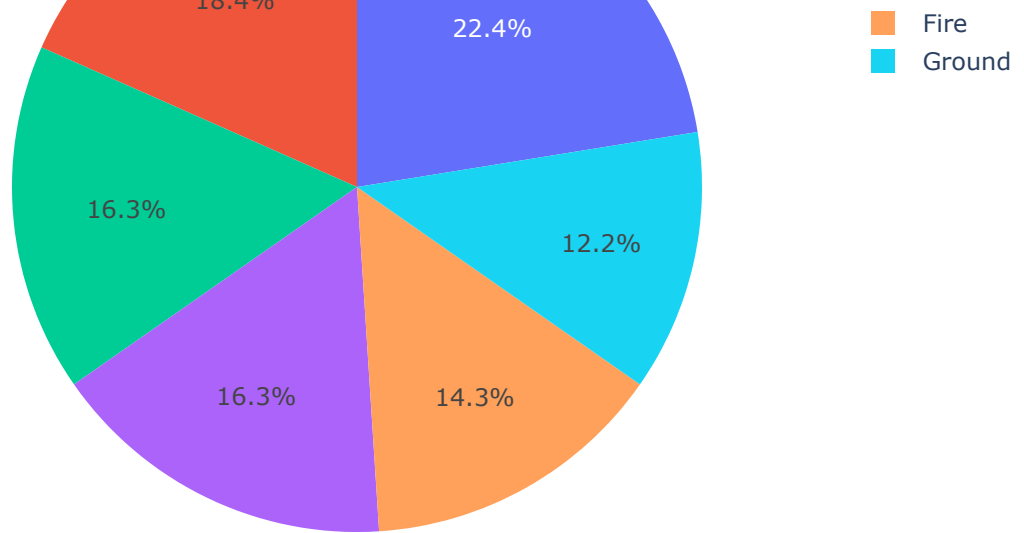
```
In [23]: px.pie(df_asia, values = 'pop', names='country',
              title = 'Population of Asian Continent',
              color_discrete_sequence = px.colors.sequential.RdBu)
```

Population of Asian Continent



```
In [24]: colors = ['blue', 'green', 'black', 'purple', 'red', 'brown']
fig = go.Figure(data=[go.Pie(labels = ['Water', 'Grass', 'Normal', 'Psychic', 'Fire', 'G
              values = [110, 90, 80, 80, 70, 60])])
fig
```

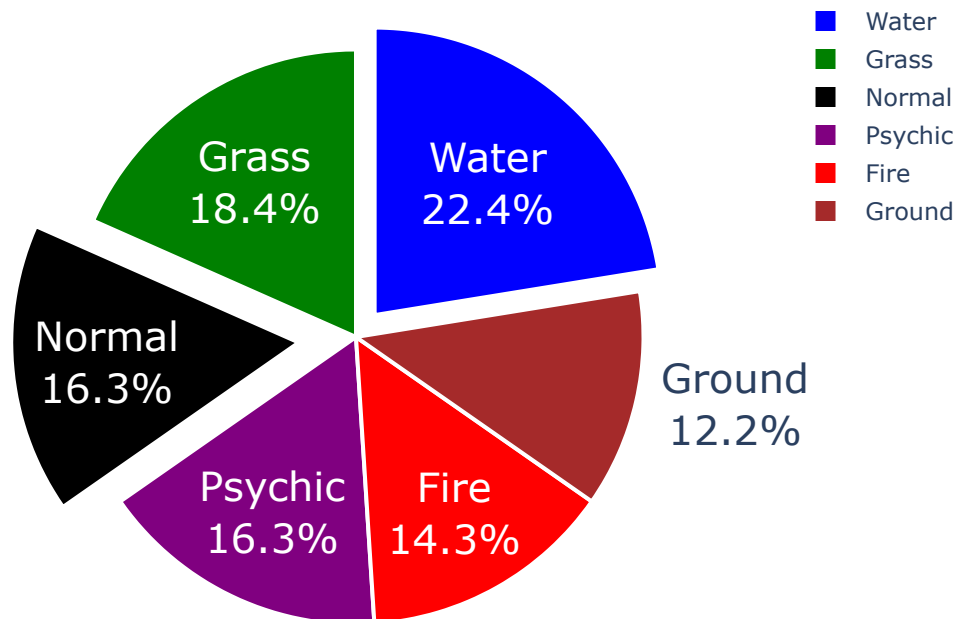




```
In [25]: colors = ['blue', 'green', 'black', 'purple', 'red', 'brown']
fig = go.Figure(data=[go.Pie(labels = ['Water', 'Grass', 'Normal', 'Psychic', 'Fire', 'Ground'],
                              values = [110, 90, 80, 80, 70, 60])])

fig.update_traces(hoverinfo='label+percent', textfont_size=20, textinfo = 'label+percent',
                  pull = [0.1, 0, 0.2, 0, 0, 0],
                  marker = dict(colors=colors, line=dict(color='#FFFFFF',
                                                         width=2)))

fig
```



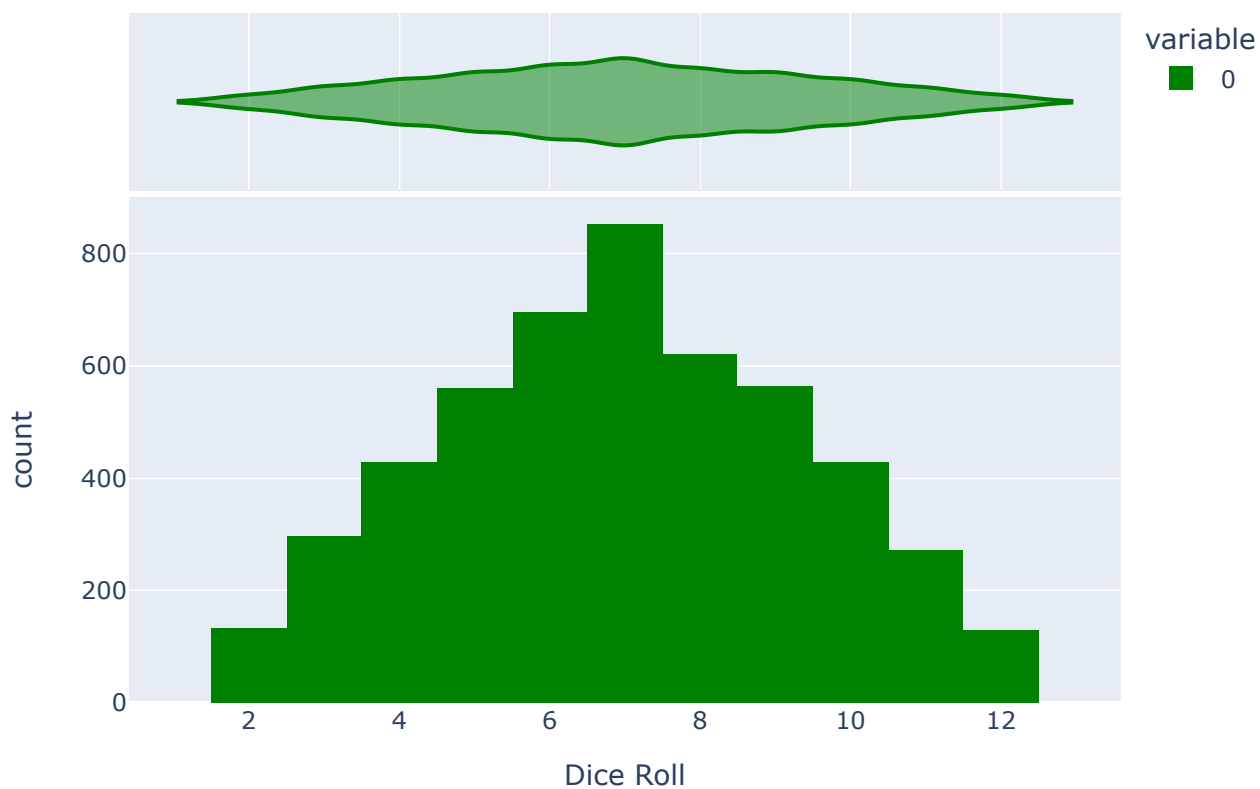
Histogram

```
In [26]: dice_1 = np.random.randint(1,7, 5000)
dice_2 = np.random.randint(1,7, 5000)
dice_sum = dice_1 + dice_2
```

```
In [27]: fig = px.histogram(dice_sum, nbins=11, labels={'value':'Dice Roll'}, title = '5000 Dice
        marginal = 'violin', color_discrete_sequence=['green'])
```

```
In [28]: fig
```

5000 Dice Roll Histogram

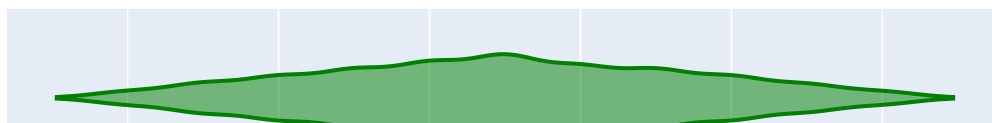


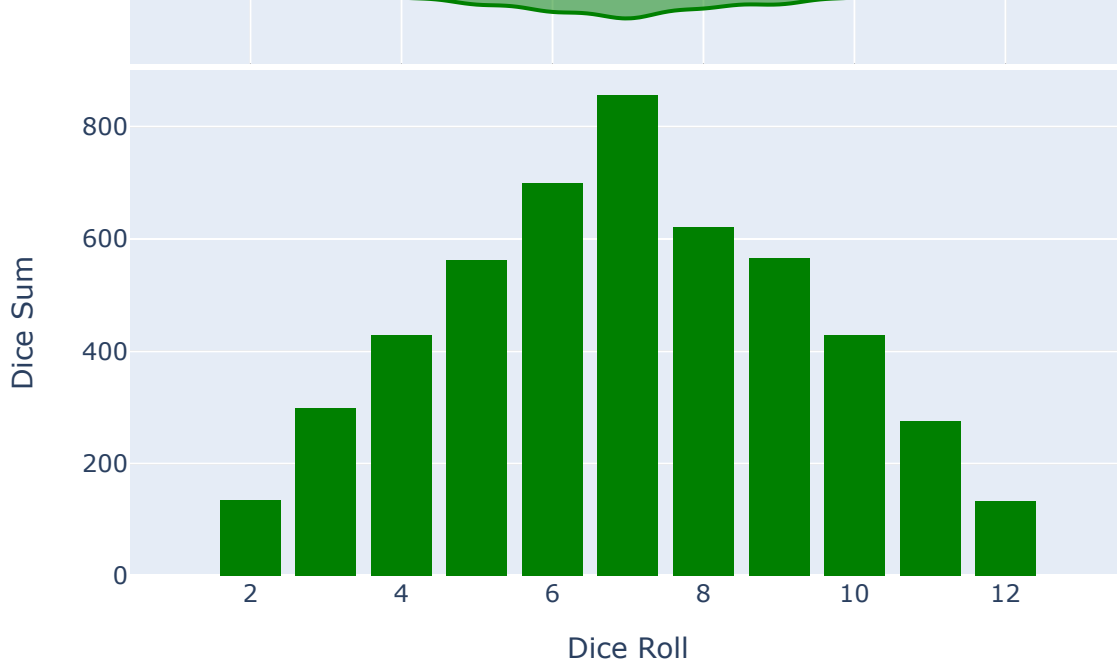
```
In [29]: fig = px.histogram(dice_sum, nbins=11, labels={'value':'Dice Roll'}, title = '5000 Dice
        marginal = 'violin', color_discrete_sequence=['green'])

fig.update_layout(axis_title_text='Dice Roll',
                  yaxis_title_text='Dice Sum',
                  bargap=0.2, showlegend=False)

fig
```

5000 Dice Roll Histogram





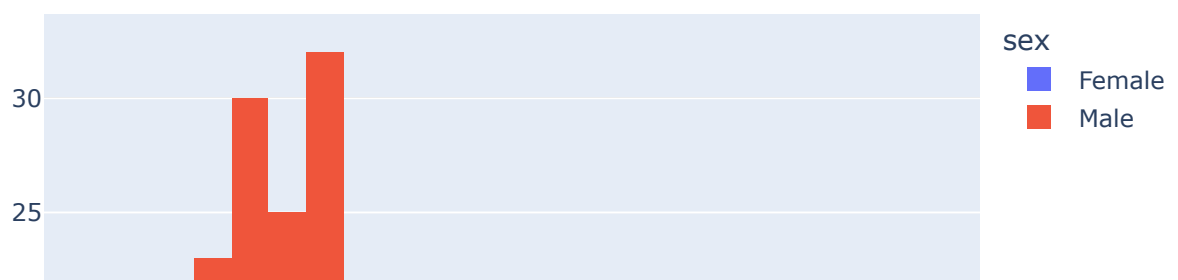
```
In [30]: df_tips = px.data.tips()
df_tips
```

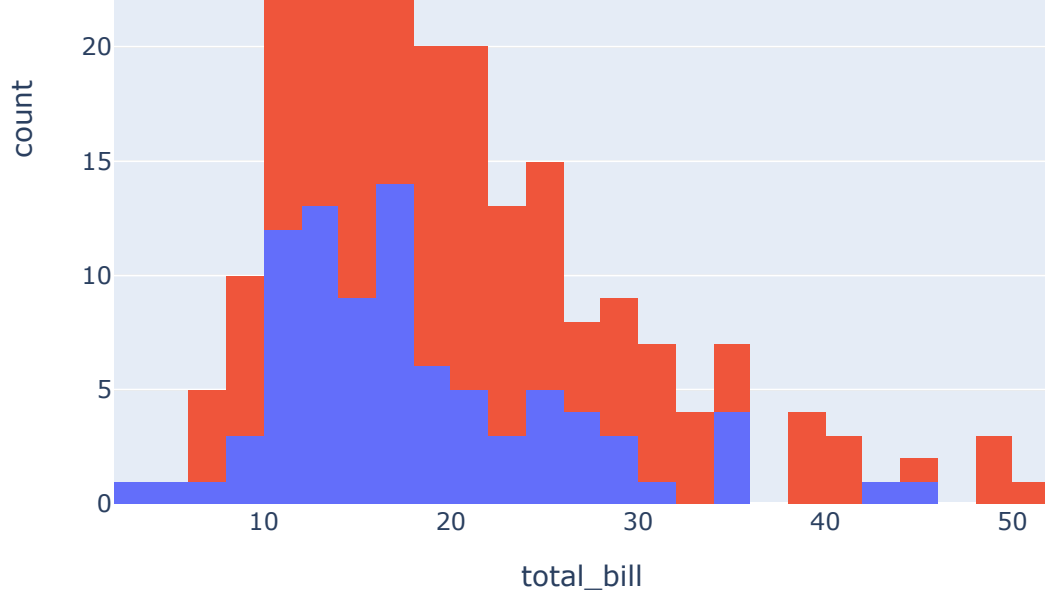
Out[30]:

	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 [31]: px.histogram(df_tips, x='total_bill', color='sex')
```





Box Plots

```
In [32]: df_tips = px.data.tips()
df_tips
```

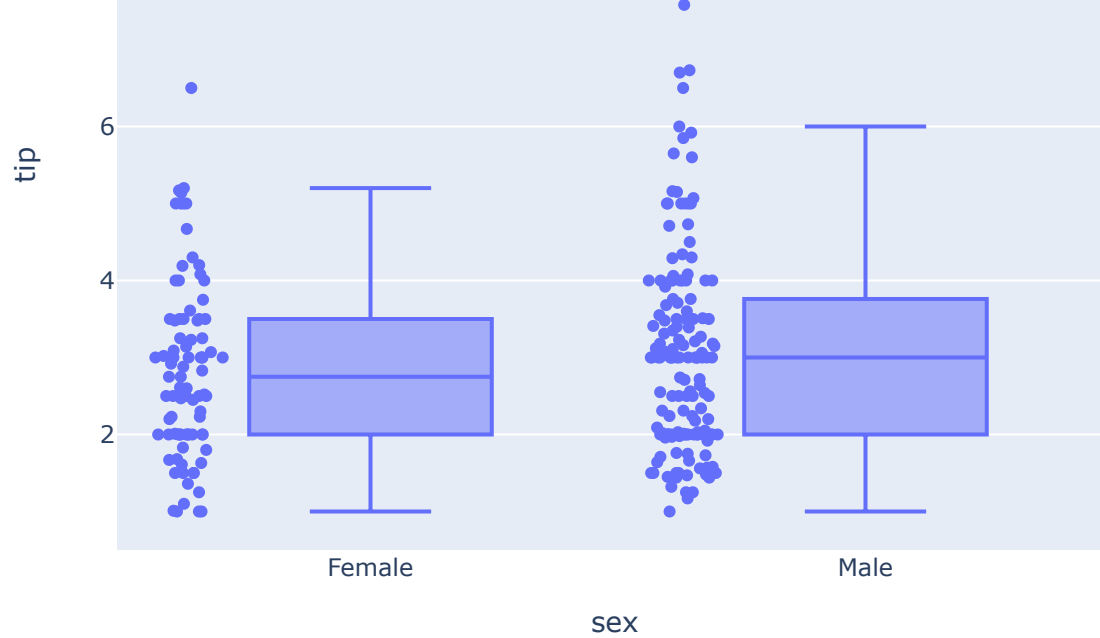
```
Out[32]:
```

	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

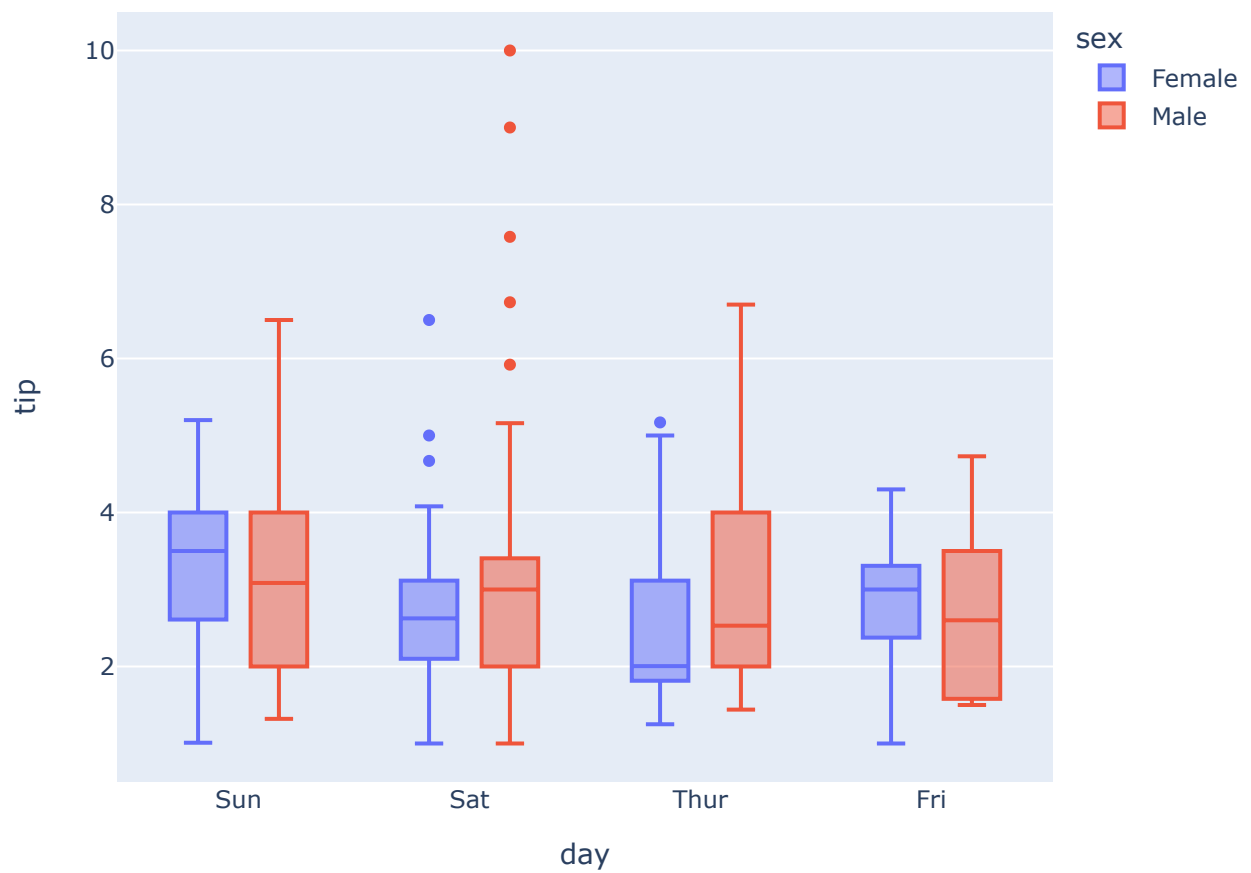
244 rows × 7 columns

```
In [33]: px.box(df_tips, x='sex', y='tip', points='all')
```

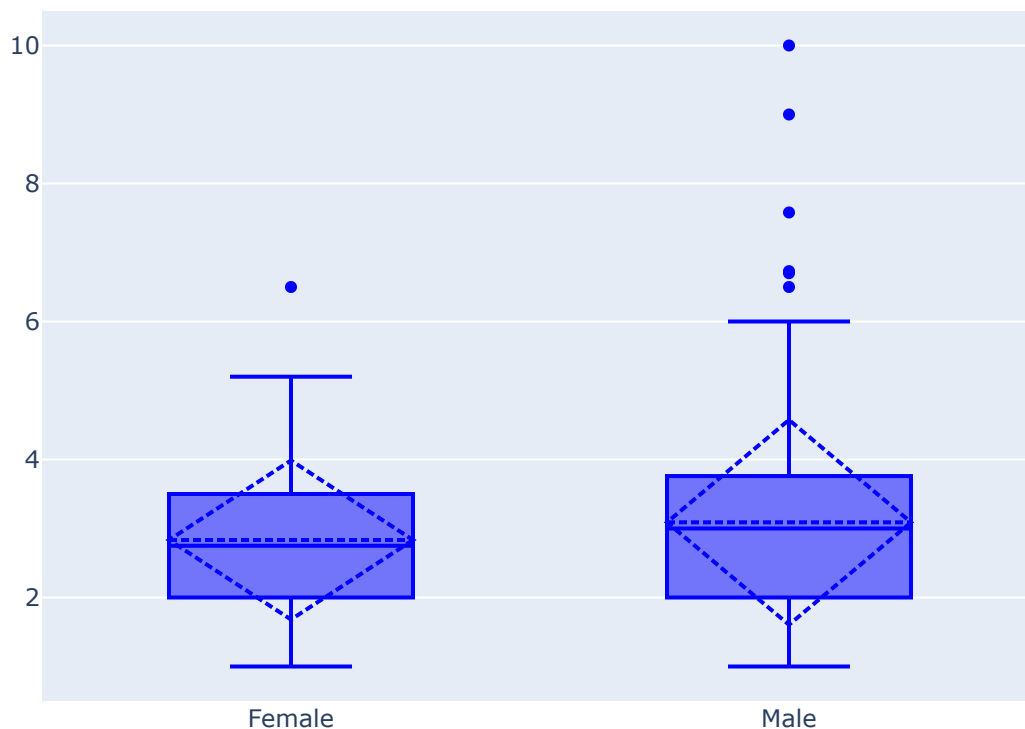




```
In [34]: px.box(df_tips, x='day', y='tip', color='sex')
```



```
In [35]: fig = go.Figure()
fig.add_trace(go.Box(x=df_tips.sex, y=df_tips.tip, marker_color='blue', boxmean='sd'))
```



```
In [36]: df_stocks = px.data.stocks()
df_stocks
```

```
Out[36]:
```

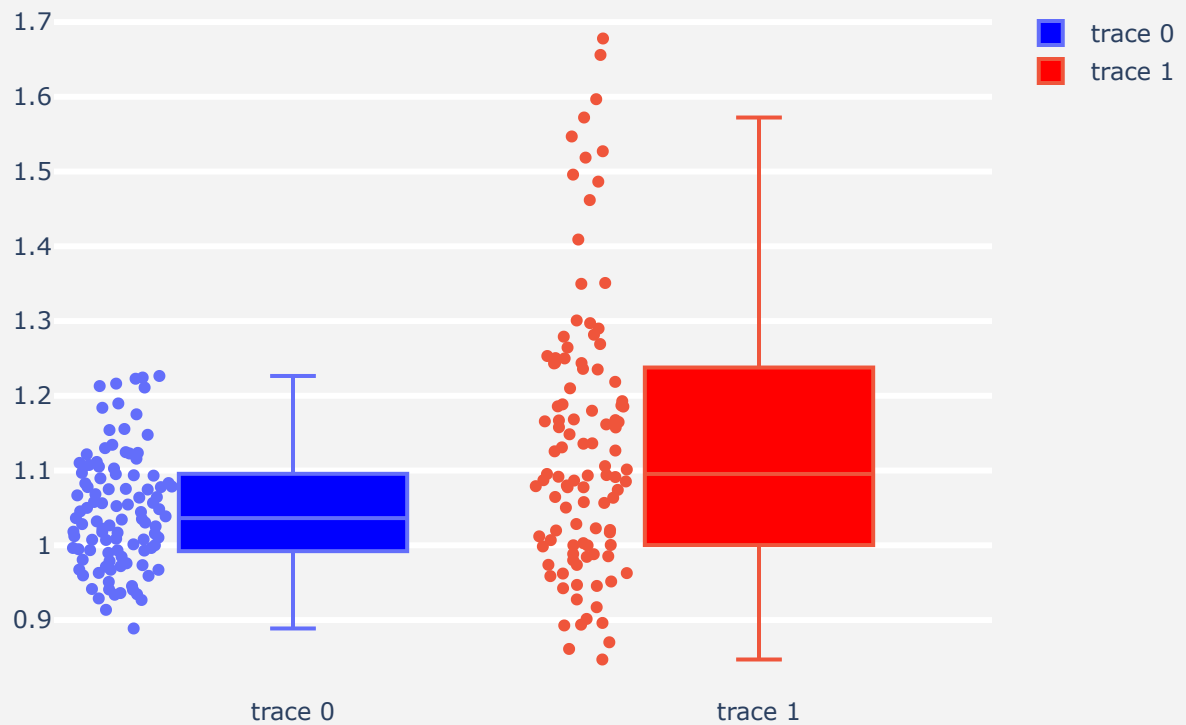
	date	GOOG	AAPL	AMZN	FB	NFLX	MSFT
0	2018-01-01	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1	2018-01-08	1.018172	1.011943	1.061881	0.959968	1.053526	1.015988
2	2018-01-15	1.032008	1.019771	1.053240	0.970243	1.049860	1.020524
3	2018-01-22	1.066783	0.980057	1.140676	1.016858	1.307681	1.066561
4	2018-01-29	1.008773	0.917143	1.163374	1.018357	1.273537	1.040708
...
100	2019-12-02	1.216280	1.546914	1.425061	1.075997	1.463641	1.720717
101	2019-12-09	1.222821	1.572286	1.432660	1.038855	1.421496	1.752239
102	2019-12-16	1.224418	1.596800	1.453455	1.104094	1.604362	1.784896
103	2019-12-23	1.226504	1.656000	1.521226	1.113728	1.567170	1.802472
104	2019-12-30	1.213014	1.678000	1.503360	1.098475	1.540883	1.788185

105 rows × 7 columns

```
In [37]: fig = go.Figure()
fig.add_trace(go.Box(y=df_stocks.GOOG, boxpoints='all', fillcolor='blue', jitter=0.5,
                    whiskerwidth=0.2))
fig.add_trace(go.Box(y=df_stocks.AAPL, boxpoints='all', fillcolor='red', jitter=0.5,
                    whiskerwidth=0.2))
fig.update_layout(title='Google vs Apple', yaxis = dict(gridcolor='rgb(255, 255, 255)',
                    gridwidth=3),
```

```
paper_bgcolor='rgb(243, 243, 243)',  
plot_bgcolor='rgb(243, 243, 243)')
```

Google vs Apple



Violin Plots

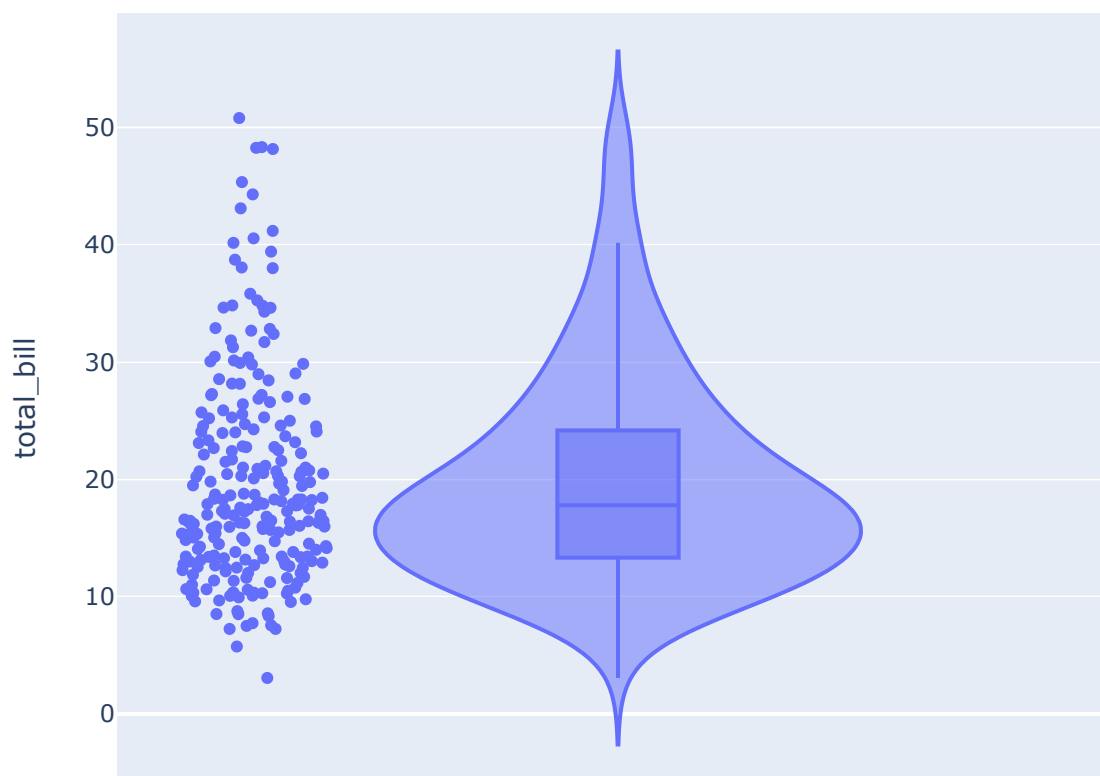
```
In [38]: df_tips = px.data.tips()  
df_tips
```

```
Out[38]:
```

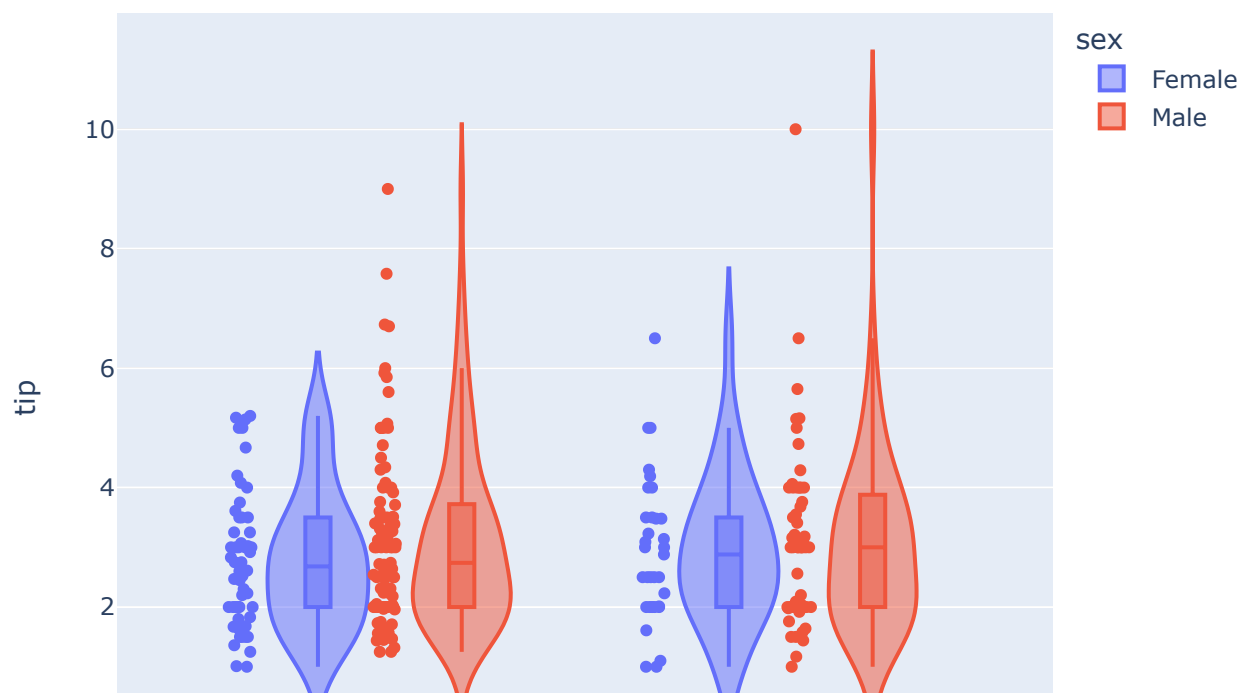
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67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

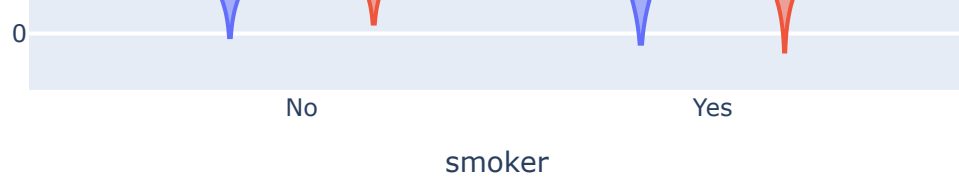
244 rows × 7 columns

```
In [39]: px.violin(df_tips, y='total_bill', box=True, points = 'all')
```

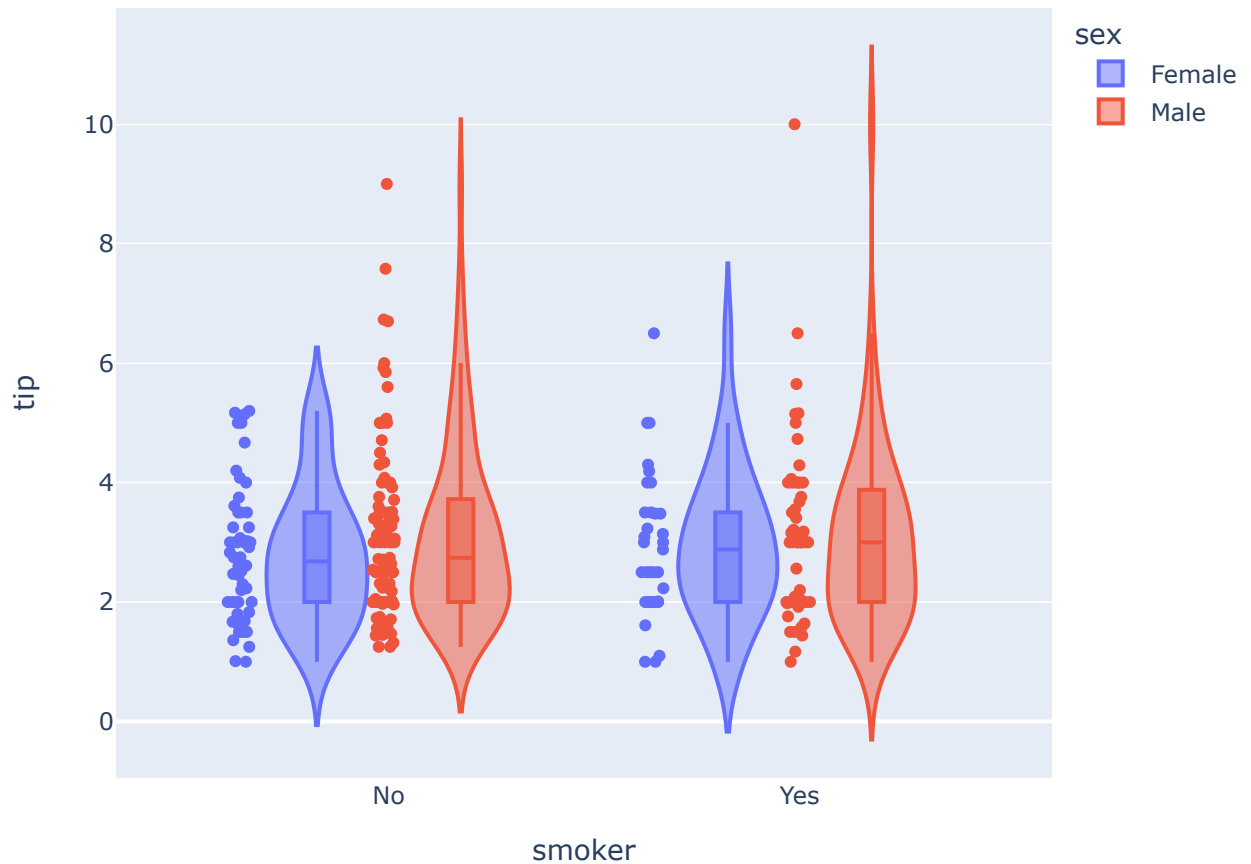


```
In [40]: px.violin(df_tips, y='tip', x='smoker', color='sex', box=True, points='all', hover_data=
```





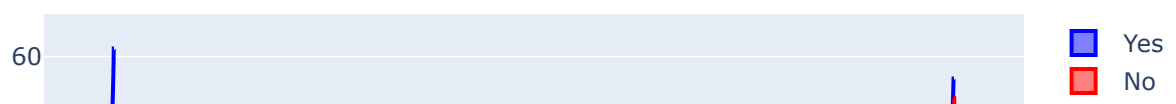
```
In [41]: px.violin(df_tips, y='tip', x='smoker', color='sex', box=True, points='all')
```

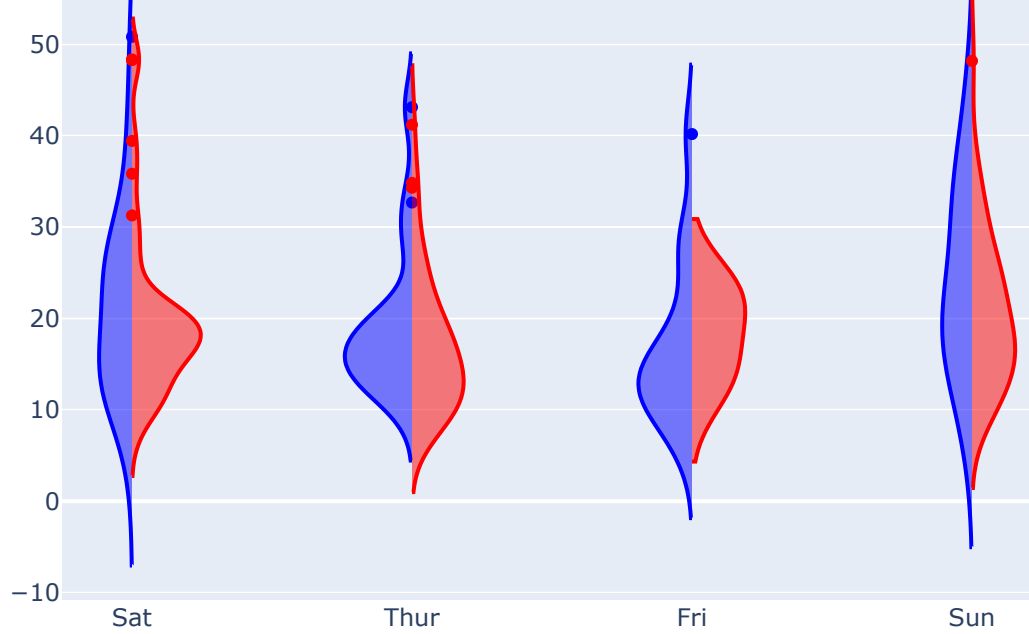


```
In [42]: fig = go.Figure()

fig.add_trace(go.Violin(x=df_tips['day'][df_tips['smoker']=='Yes'],
                        y=df_tips['total_bill'][df_tips['smoker']=='Yes'],
                        legendgroup = 'Yes', scalegroup='Yes', name = 'Yes', side = 'nega
                        line_color='blue'))

fig.add_trace(go.Violin(x=df_tips['day'][df_tips['smoker']=='No'],
                        y=df_tips['total_bill'][df_tips['smoker']=='No'],
                        legendgroup = 'Yes', scalegroup='Yes', name = 'No', side = 'posit
                        line_color='red'))
```





Density Heatmaps

```
In [43]: flights = sns.load_dataset('flights')
flights
```

```
Out[43]:
```

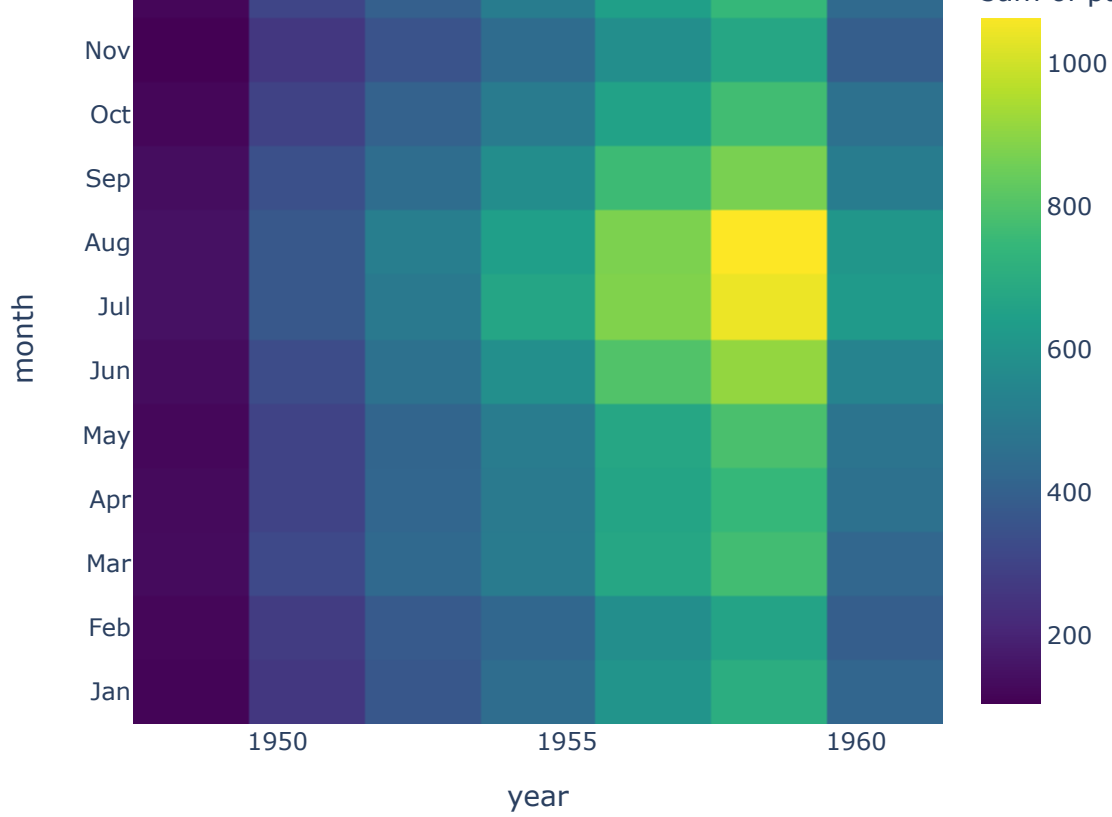
	year	month	passengers
0	1949	Jan	112
1	1949	Feb	118
2	1949	Mar	132
3	1949	Apr	129
4	1949	May	121
...
139	1960	Aug	606
140	1960	Sep	508
141	1960	Oct	461
142	1960	Nov	390
143	1960	Dec	432

144 rows × 3 columns

```
In [44]: fig = px.density_heatmap(flights, x='year', y='month', z='passengers', color_continuous_
```

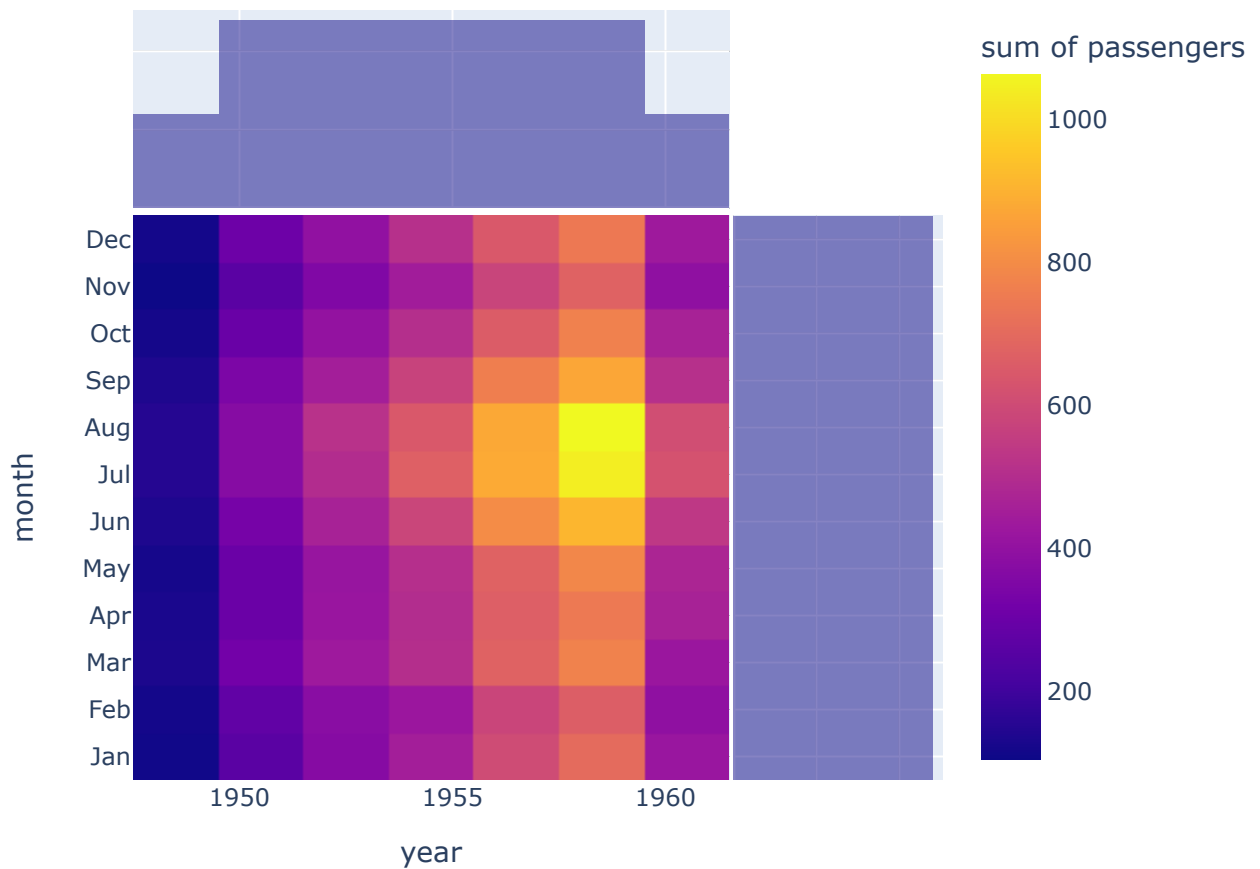
```
In [45]: fig
```





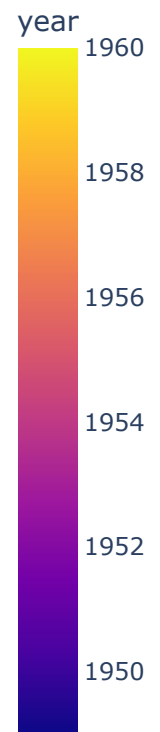
```
In [46]: fig = px.density_heatmap(flights, x='year', y='month', z='passengers',
                                marginal_x = 'histogram', marginal_y = 'histogram' )

fig
```



3D Scatter Plot

```
In [47]: fig = px.scatter_3d(flights, x='year', y='month', z='passengers', color='year', opacity=0.5)
fig
```



3D Line Plots

```
In [48]: fig = px.line_3d(flights, x='year', y='month', z='passengers', color='year')
fig
```



1957
1958
1959
1960

Scatter Matrix

```
In [49]: # https://www.youtube.com/watch?v=GGL6U0k8WYA
# Time: 1:04:00
```

```
In [50]: fig = px.scatter_matrix(flights, color='month')
fig
```

C:\Users\azizt\anaconda3\lib\site-packages\plotly\express_core.py:279: FutureWarning: iteritems is deprecated and will be removed in a future version. Use .items instead.

