

## ▼ Plotly 한번에 제대로 배우기



plotly

## ▼ Plotly 특징

- 인터랙티브 그래프 생성
- JSON 데이터 형식으로 저장
- 벡터 이미지, 래스터 이미지로 Export 가능

홈페이지: <https://plotly.com/python/>

```
import numpy as np
import pandas as pd
from urllib.request import urlopen
import json
import plotly.io as pio
import plotly.express as px
import plotly.graph_objects as go
import plotly.figure_factory as ff
from plotly.subplots import make_subplots
from plotly.validators.scatter.marker import SymbolValidator
```

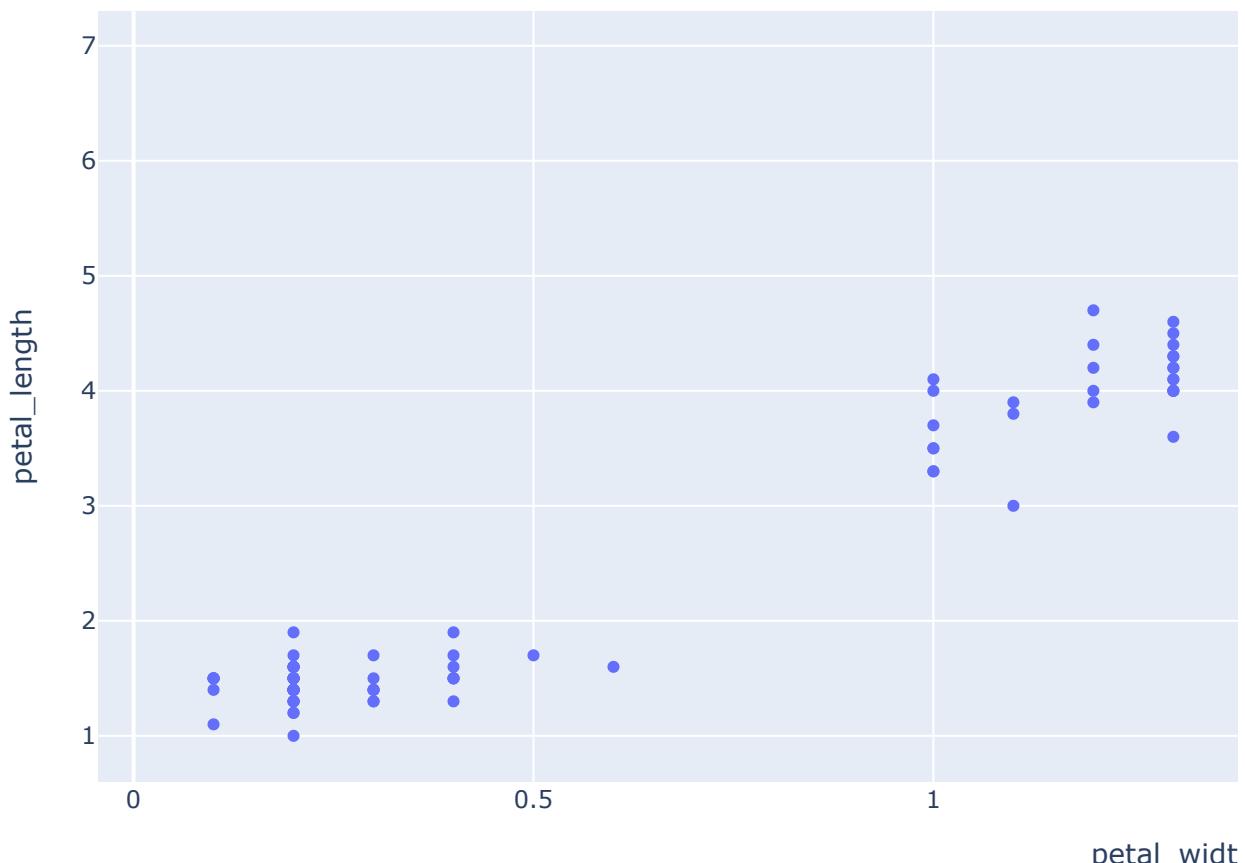
## ▼ Plotly 차트

### ▼ 산점도(Scatter Plots)

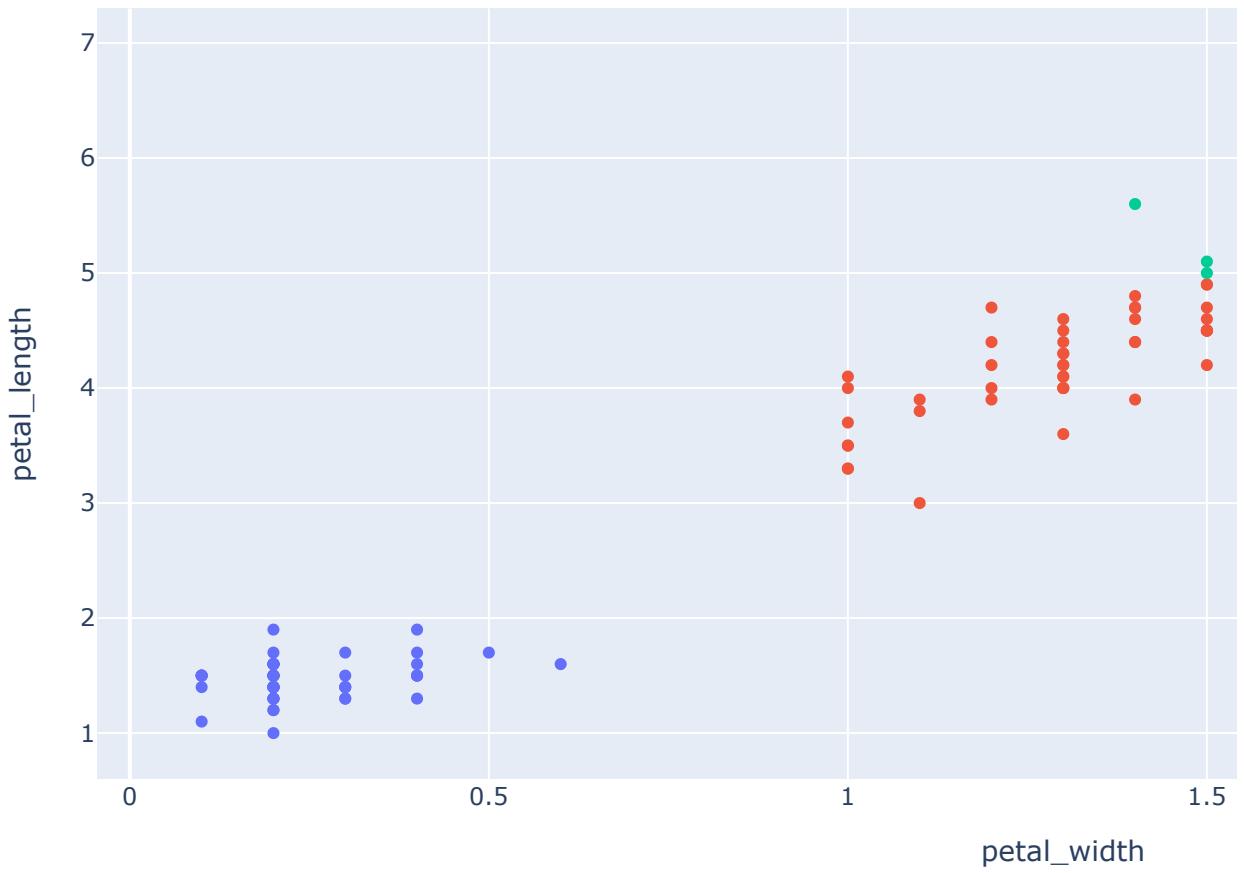
```
iris = px.data.iris()
iris
```

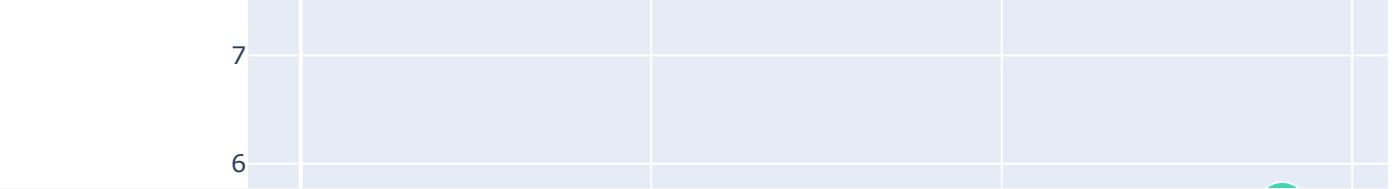
	sepal_length	sepal_width	petal_length	petal_width	species	species_id
0	5.1	3.5	1.4	0.2	setosa	1
1	4.9	3.0	1.4	0.2	setosa	1
2	4.7	3.2	1.3	0.2	setosa	1
3	4.6	3.1	1.5	0.2	setosa	1
4	5.0	3.6	1.4	0.2	setosa	1
...	...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica	3
146	6.3	2.5	5.0	1.9	virginica	3
147	6.5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

```
fig = px.scatter(iris, x='petal_width', y='petal_length')
fig.show()
```



```
fig = px.scatter(iris, x='petal_width', y='petal_length', color='species')
fig.show()
```



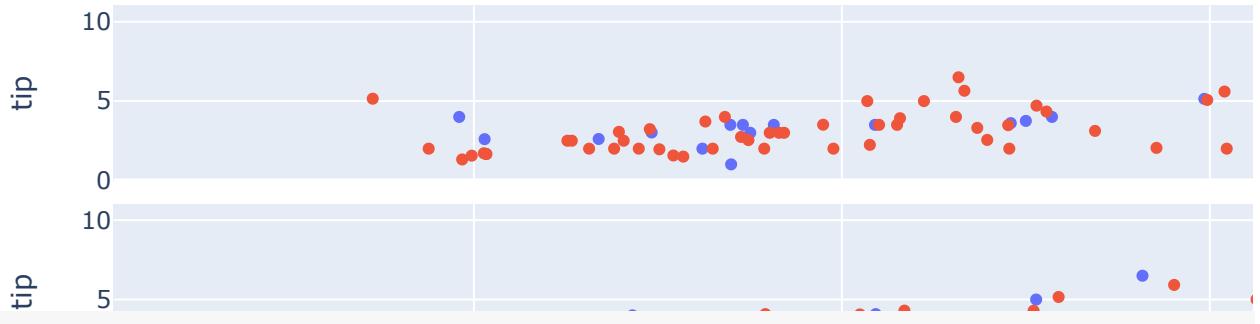


```
tips = px.data.tips()  
tips
```

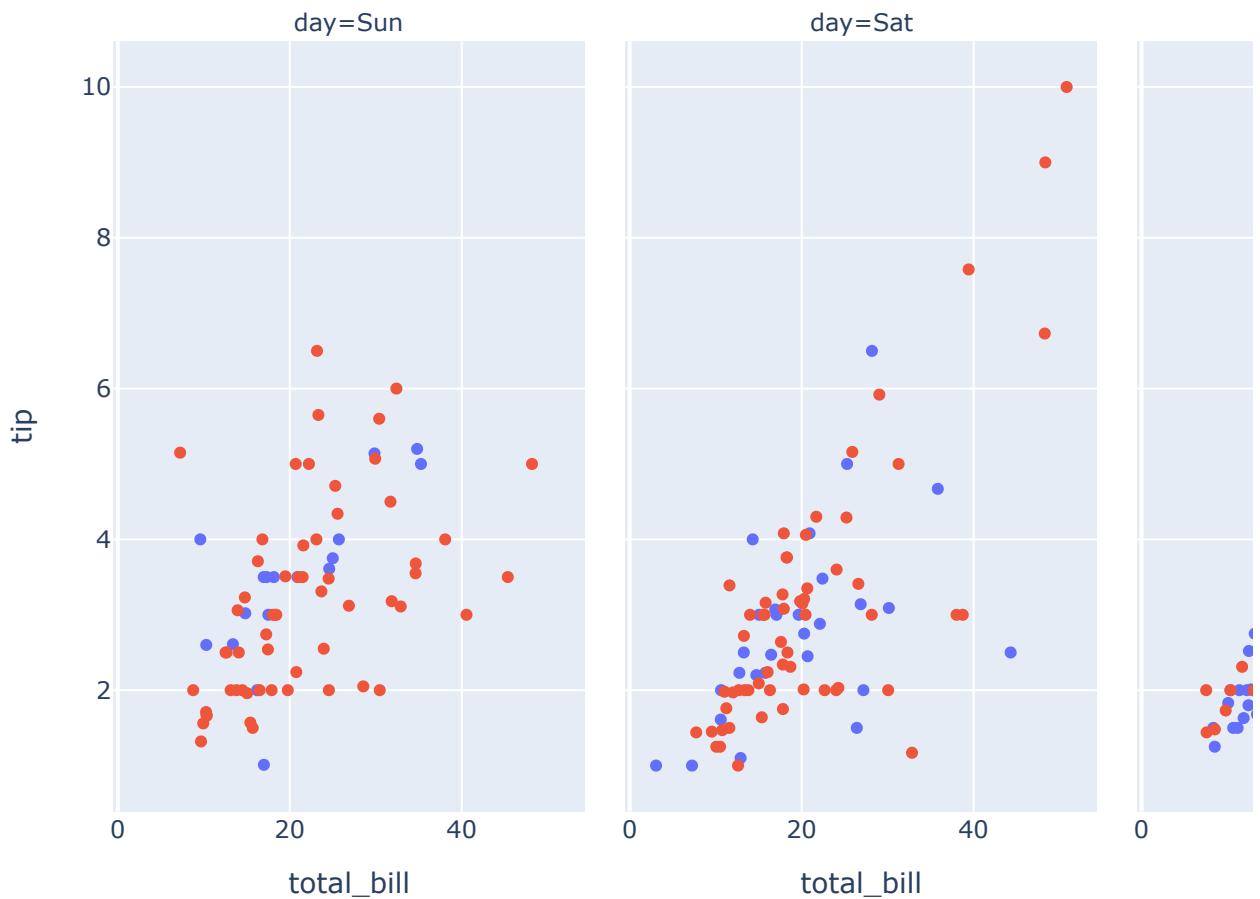
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female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

```
fig = px.scatter(tips, x='total_bill', y='tip', color='sex', facet_row='day')  
fig.show()
```

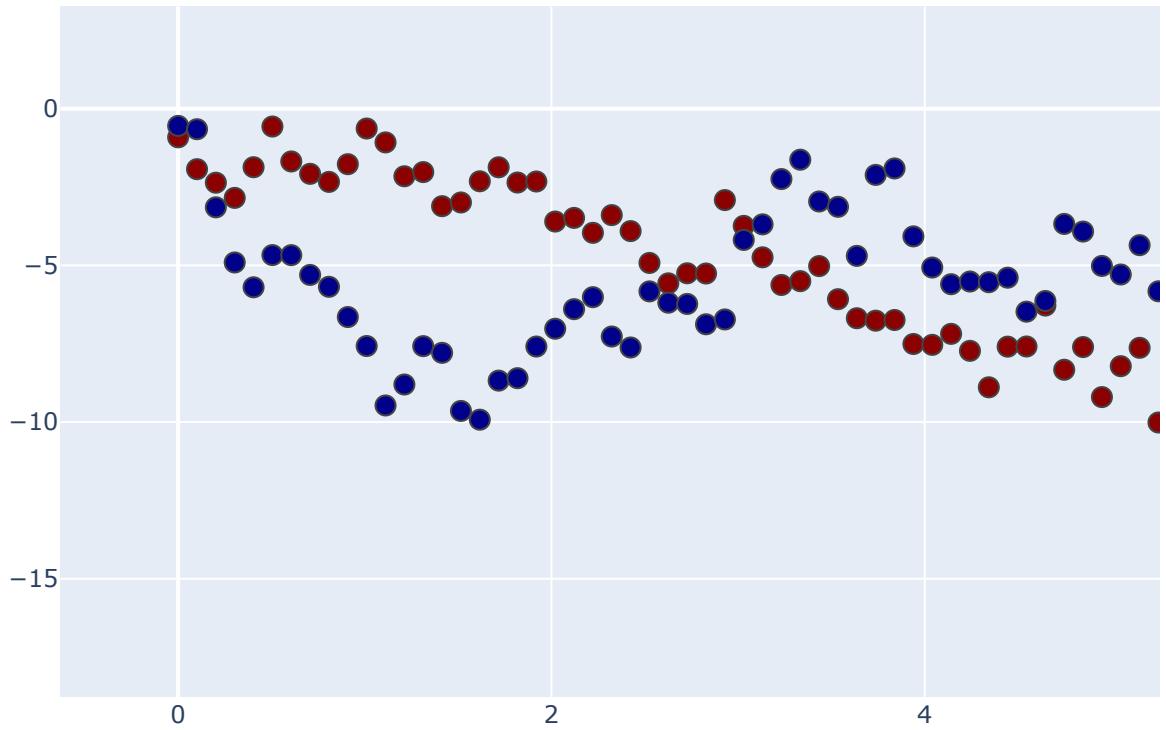


```
fig = px.scatter(tips, x='total_bill', y='tip', color='sex', facet_col='day')
fig.show()
```



```
t = np.linspace(0, 10, 100)
y1 = np.random.randn(100).cumsum()
y2 = np.random.randn(100).cumsum()

fig = go.Figure()
fig.add_trace(go.Scatter(x=t, y=y1, name='red', mode='markers', marker_color='darkred'))
fig.add_trace(go.Scatter(x=t, y=y2, name='blue', mode='markers', marker_color='darkblue'))
fig.update_traces(mode='markers', marker_line_width=1, marker_size=10)
fig.show()
```



```
fig = go.Figure(data=go.Scatter(x=[1, 2, 3, 4, 5],  
                                 y=[11, 12, 13, 14, 15],  
                                 mode='markers',  
                                 marker=dict(size=[20, 40, 60, 80, 100],  
                                             color=[1, 2, 3, 4, 5])))  
fig.show()
```

16

```
gapminder = px.data.gapminder()  
gapminder
```

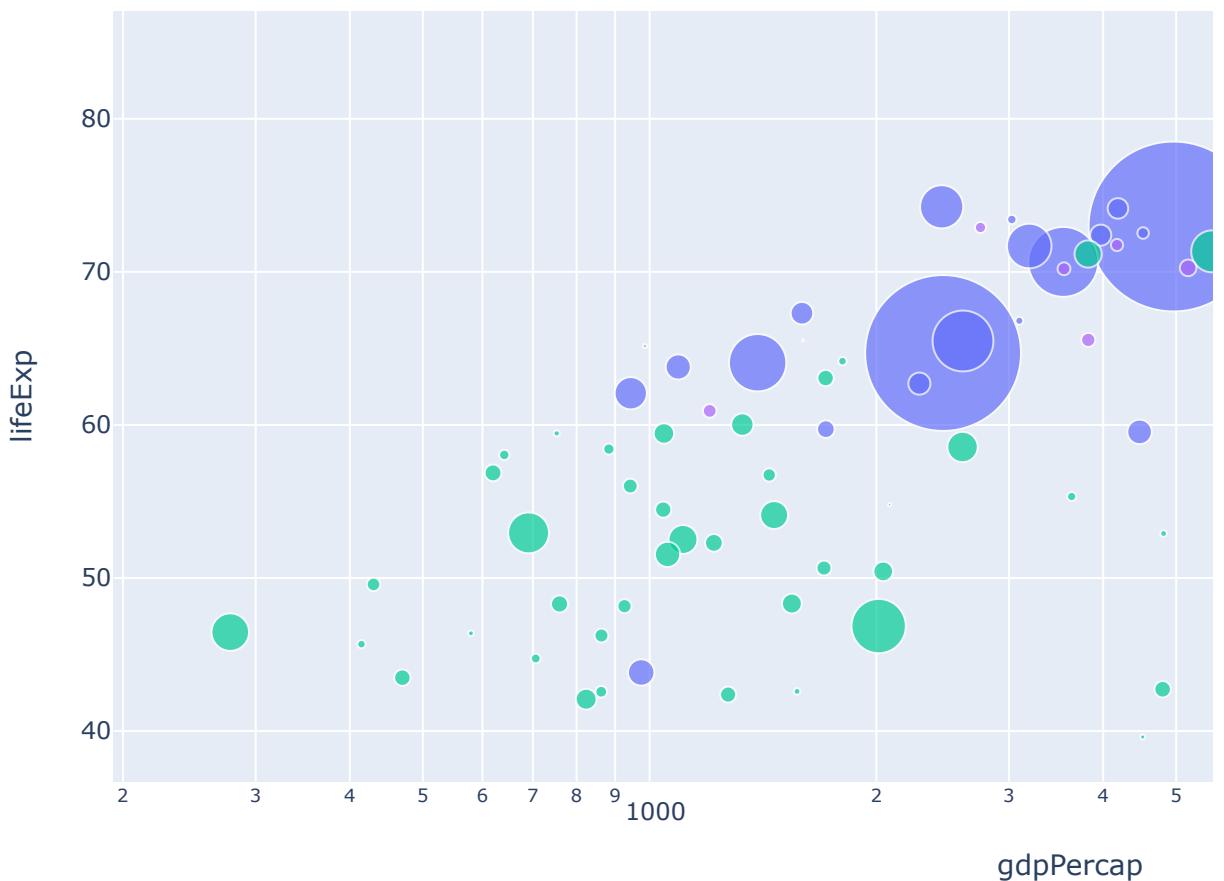
	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
0	Afghanistan	Asia	1952	28.801	8425333	779.445314	AFG	4
1	Afghanistan	Asia	1957	30.332	9240934	820.853030	AFG	4
2	Afghanistan	Asia	1962	31.997	10267083	853.100710	AFG	4
3	Afghanistan	Asia	1967	34.020	11537966	836.197138	AFG	4
4	Afghanistan	Asia	1972	36.088	13079460	739.981106	AFG	4
...	...	...	...	...	...	...	...	...
1699	Zimbabwe	Africa	1987	62.351	9216418	706.157306	ZWE	716
1700	Zimbabwe	Africa	1992	60.377	10704340	693.420786	ZWE	716
1701	Zimbabwe	Africa	1997	46.809	11404948	792.449960	ZWE	716
1702	Zimbabwe	Africa	2002	39.989	11926563	672.038623	ZWE	716
1703	Zimbabwe	Africa	2007	43.487	12311143	469.709298	ZWE	716

1704 rows × 8 columns

```
gapminder_2007 = gapminder.query("year == 2007")  
gapminder_2007
```

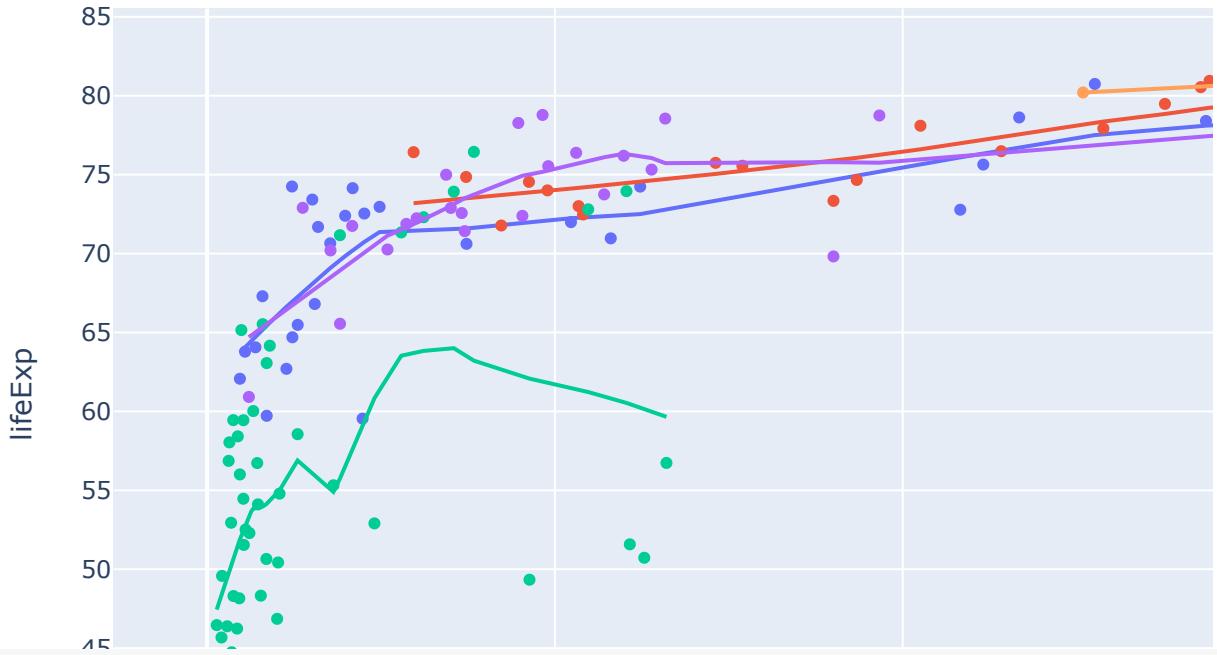
	country	continent	year	lifeExp	pop	gdpPerCap	iso_alpha	iso_nu
11	Afghanistan	Asia	2007	43.828	31889923	974.580338	AFG	
23	Albania	Europe	2007	76.423	3600523	5937.029526	ALB	

```
fig = px.scatter(gapminder_2007,
                 x='gdpPerCap', y='lifeExp',
                 size='pop', color='continent',
                 hover_name='country', log_x=True, size_max=60)
fig.show()
```



```
fig = px.scatter(gapminder_2007,
                 x='gdpPerCap', y='lifeExp',
                 color='continent', trendline='lowess')
fig.show()
```

```
/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning  
pandas.util.testing is deprecated. Use the functions in the public API at pandas.t
```



```
fig = px.scatter(gapminder,  
                  x='gdpPerCap', y='lifeExp',  
                  size='pop', color='continent',  
                  facet_col='year', facet_col_wrap=4)  
fig.show()
```



year=1952

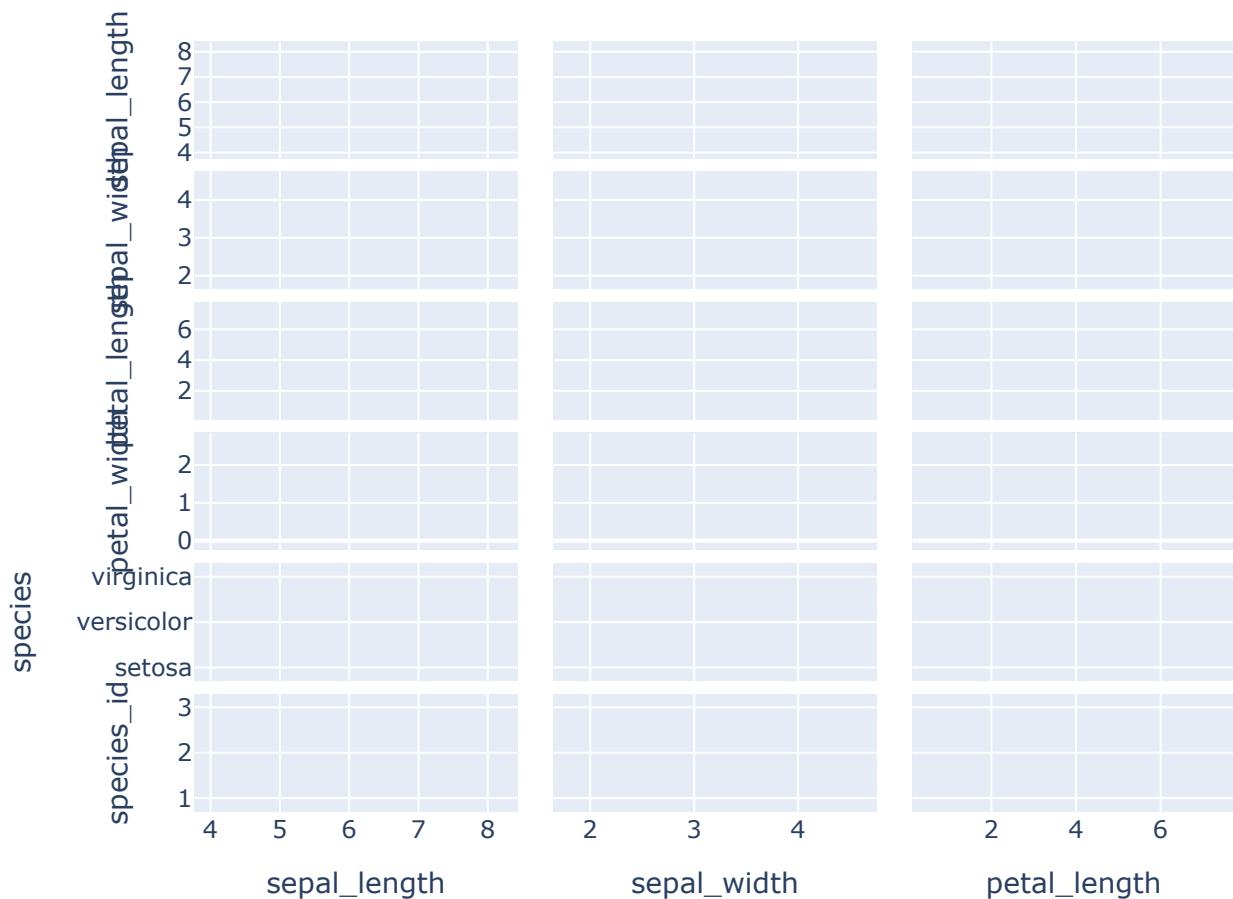
year=1957

~~

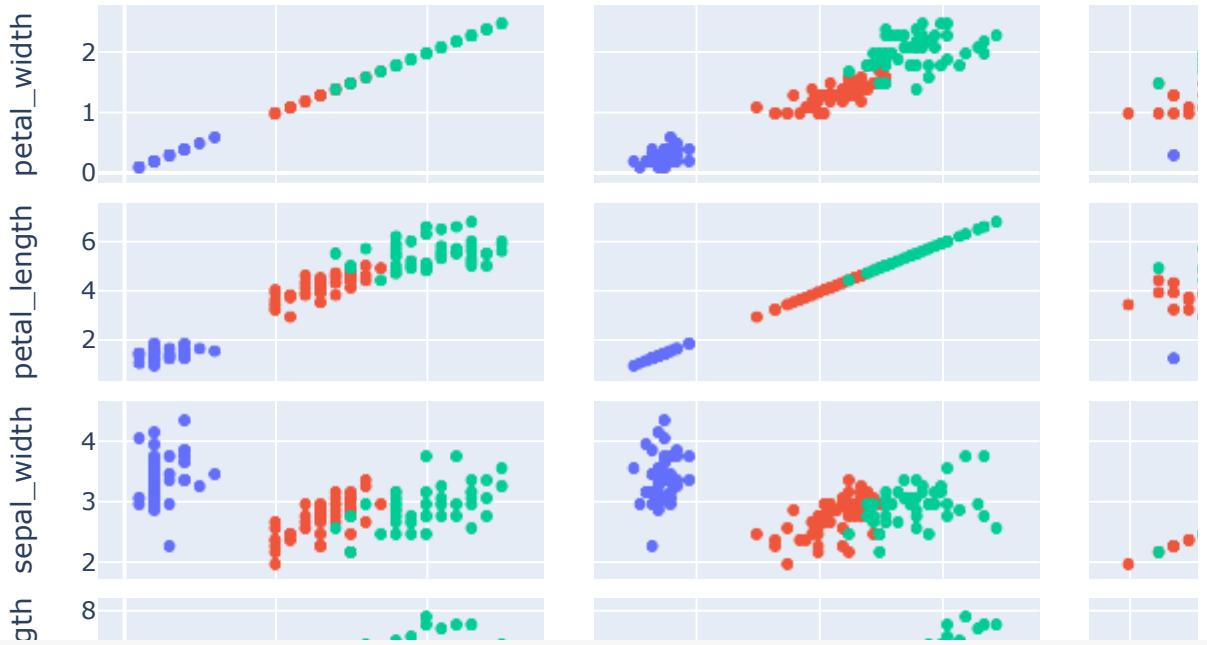
## ▼ 산점도 매트릭스(Scatter Matrix)

lit 40

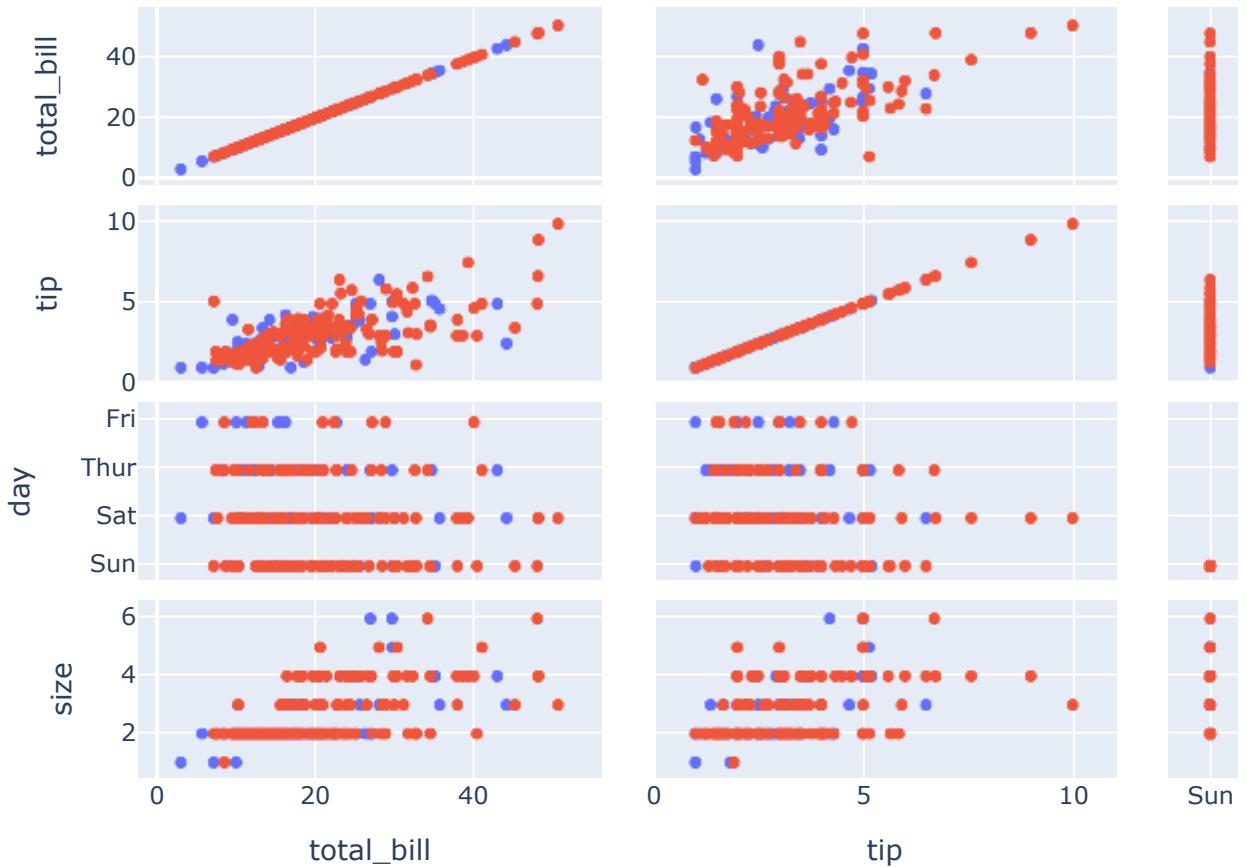
```
fig = px.scatter_matrix(iris)
fig.show()
```



```
fig = px.scatter_matrix(iris,
                        dimensions=['petal_width', 'petal_length',
                                    'sepal_width', 'sepal_length'],
                        color='species')
fig.show()
```

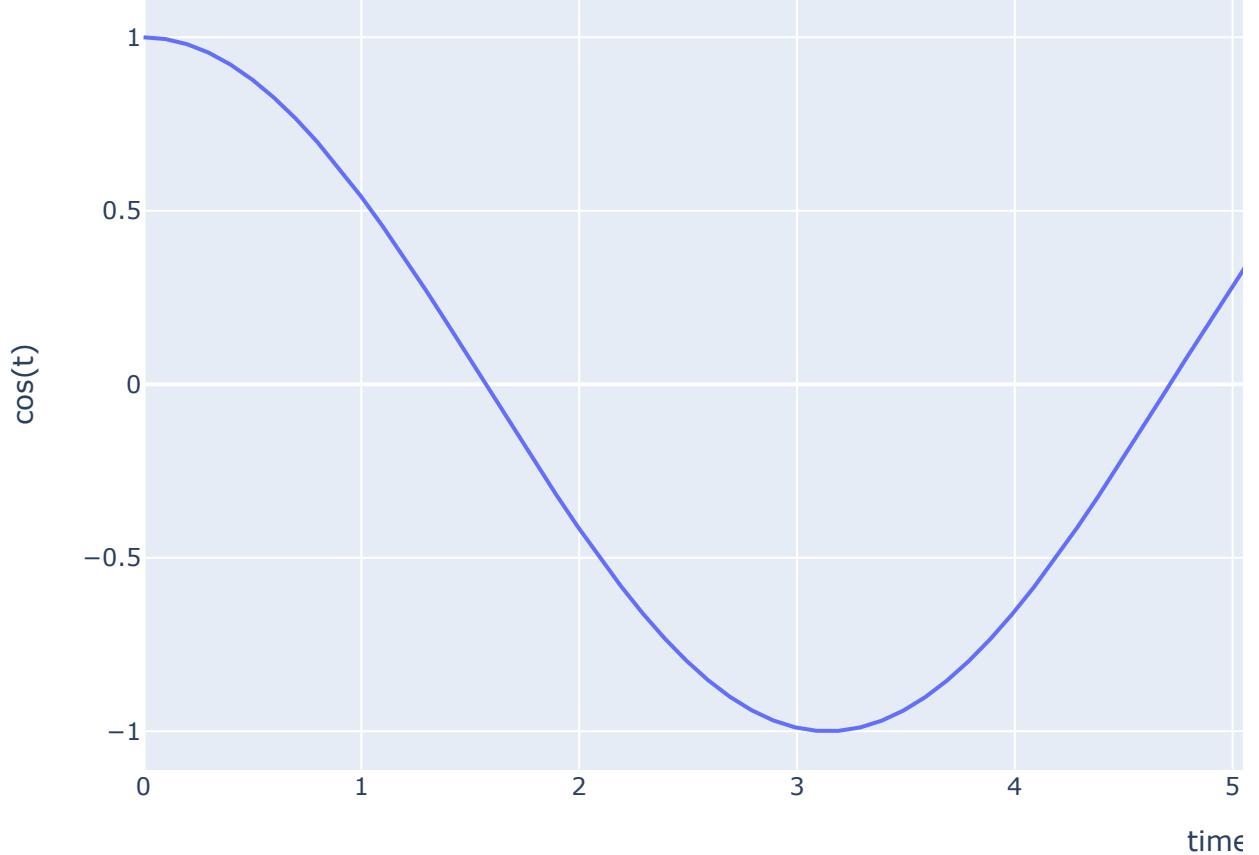


```
fig = px.scatter_matrix(tips,
                        dimensions=['total_bill', 'tip', 'day', 'size'],
                        color='sex')
fig.show()
```



## ▼ 라인 플롯(Line Plots)

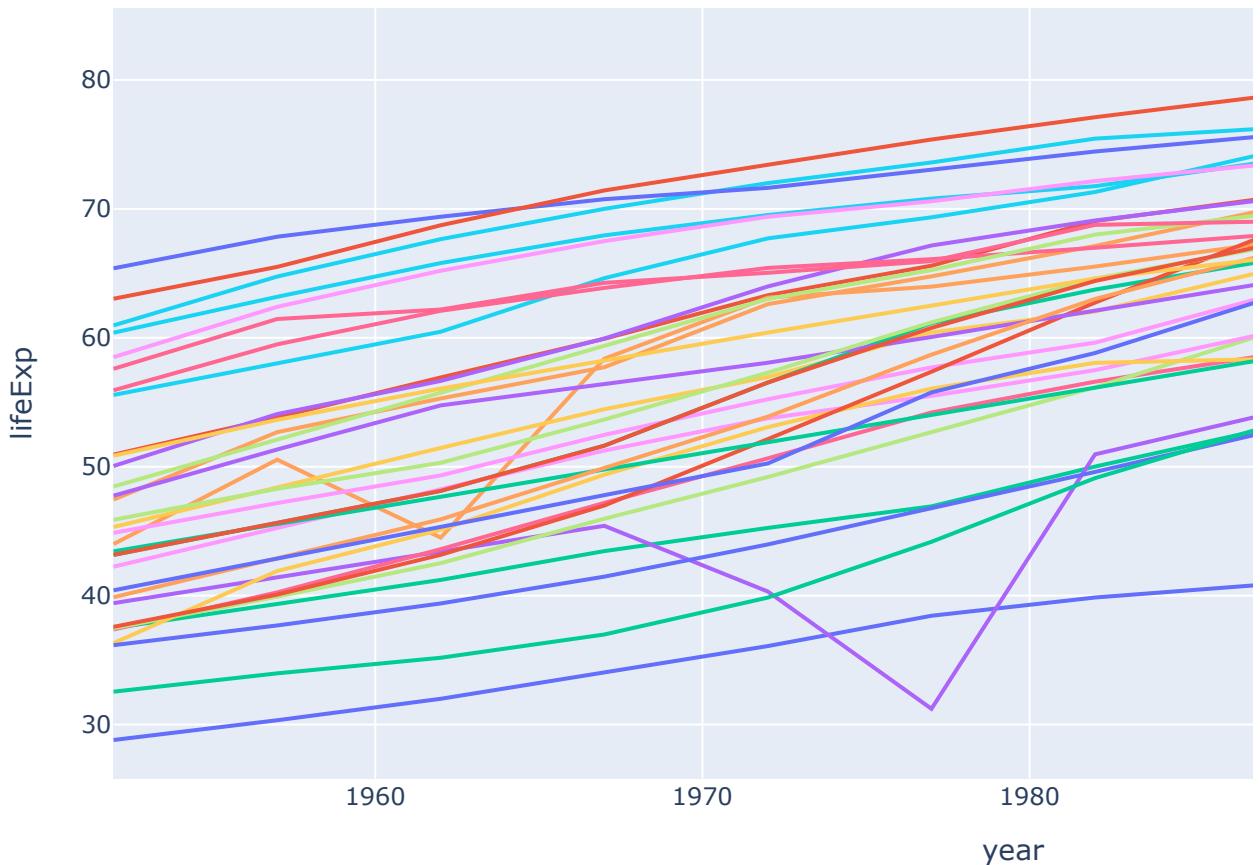
```
t = np.linspace(0, np.pi**2, 100)  
  
fig = px.line(x=t, y=np.cos(t), labels={'x':'time', 'y':'cos(t)'})  
fig.show()
```



```
gapminder_asia = gapminder.query("continent == 'Asia'")  
gapminder_asia
```

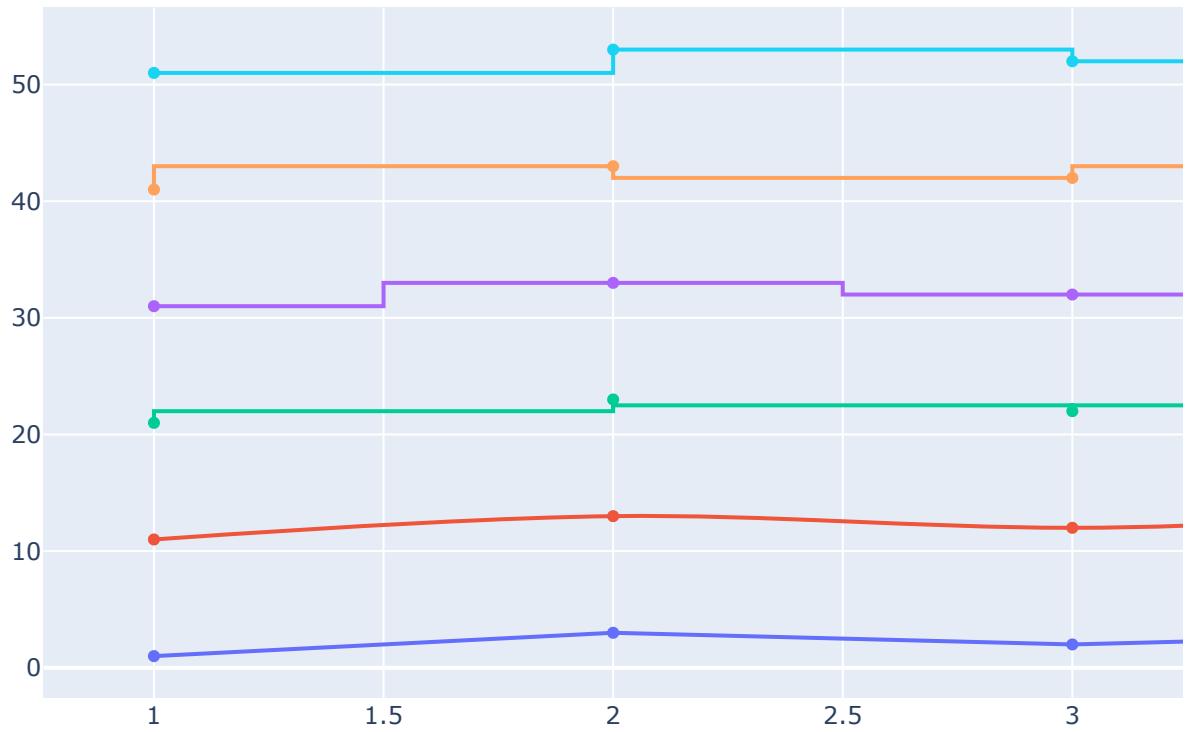
	country	continent	year	lifeExp	pop	gdpPerCap	iso_alpha	iso_i
0	Afghanistan	Asia	1952	28.801	8425333	779.445314	AFG	
1	Afghanistan	Asia	1957	30.332	9240934	820.853030	AFG	
2	Afghanistan	Asia	1962	31.997	10267083	853.100710	AFG	
3	Afghanistan	Asia	1967	34.020	11537966	836.197138	AFG	

```
fig = px.line(gapminder_asia, x='year', y='lifeExp', color='country')
fig.show()
```



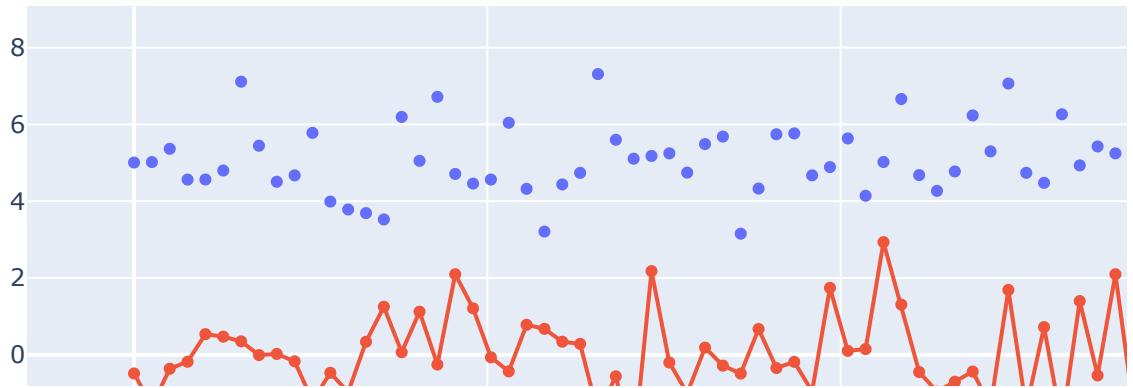
```
x = np.array([1, 2, 3, 4, 5])
y = np.array([1, 3, 2, 3, 1])

fig = go.Figure()
fig.add_trace(go.Scatter(x=x, y=y, name='linear', line_shape='linear'))
fig.add_trace(go.Scatter(x=x, y=y+10, name='spline', hoverinfo='text+name', line_shape='spline'))
fig.add_trace(go.Scatter(x=x, y=y+20, name='vhv', line_shape='vhv'))
fig.add_trace(go.Scatter(x=x, y=y+30, name='vhv', line_shape='vhv'))
fig.add_trace(go.Scatter(x=x, y=y+40, name='vh', line_shape='vh'))
fig.add_trace(go.Scatter(x=x, y=y+50, name='hv', line_shape='hv'))
fig.update_traces(hoverinfo='text+name', mode='lines+markers')
fig.update_layout(legend=dict(y=0.5, traceorder='reversed', font_size=16))
fig.show()
```



```
N = 100
x = np.linspace(0, 1, N)
y0 = np.random.randn(N) + 5
y1 = np.random.randn(N)
y2 = np.random.randn(N) - 5

fig = go.Figure()
fig.add_trace(go.Scatter(x=x, y=y0, mode='markers', name='markers'))
fig.add_trace(go.Scatter(x=x, y=y1, mode='lines+markers', name='lines+markers'))
fig.add_trace(go.Scatter(x=x, y=y2, mode='lines', name='lines'))
fig.show()
```



## ▼ 시계열(Time Series)

- <https://raw.githubusercontent.com/plotly/datasets/master/finance-charts-apple.csv>

```
aapl = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/finance-charts-apple.csv')
aapl
```

	Date	AAPL.Open	AAPL.High	AAPL.Low	AAPL.Close	AAPL.Volume	AAPL.Adjusted
0	2015-02-17	127.489998	128.880005	126.919998	127.830002	63152400	122.905000
1	2015-02-18	127.629997	128.779999	127.449997	128.720001	44891700	123.760000
2	2015-02-19	128.479996	129.029999	128.330002	128.449997	37362400	123.501000
3	2015-02-20	128.619995	129.500000	128.050003	129.500000	48948400	124.510000
4	2015-02-23	130.020004	133.000000	129.660004	133.000000	70974100	127.876000
...	...	...	...	...	...	...	...
501	2017-02-10	132.460007	132.940002	132.050003	132.119995	20065500	132.119000
502	2017-02-13	133.080002	133.820007	132.750000	133.289993	23035400	133.289000

```
fig = px.line(aapl, x='Date', y='AAPL.Close')
fig.show()
```



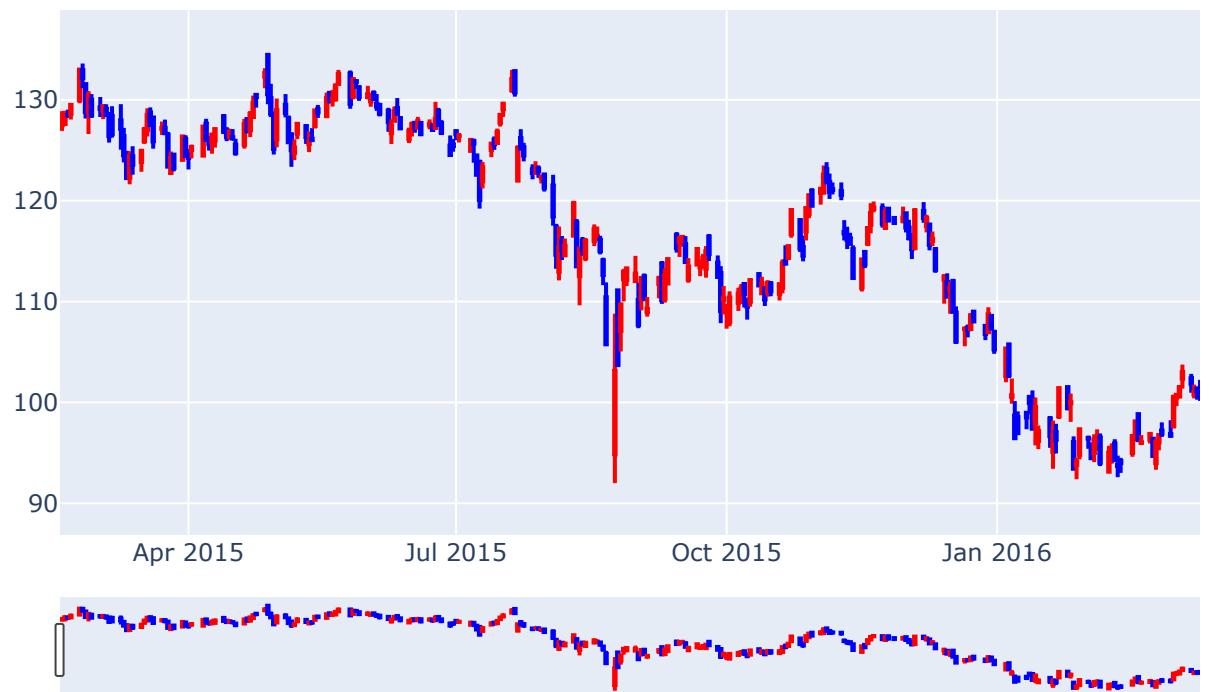
```
fig = px.line(aapl, x='Date', y='AAPL.Close', range_x=['2016-02-01','2017-02-27'])
fig.show()
```

```
fig = px.line(appl, x='Date', y='AAPL.Close')
fig.update_xaxes(rangeslider_visible=True)
fig.show()
```



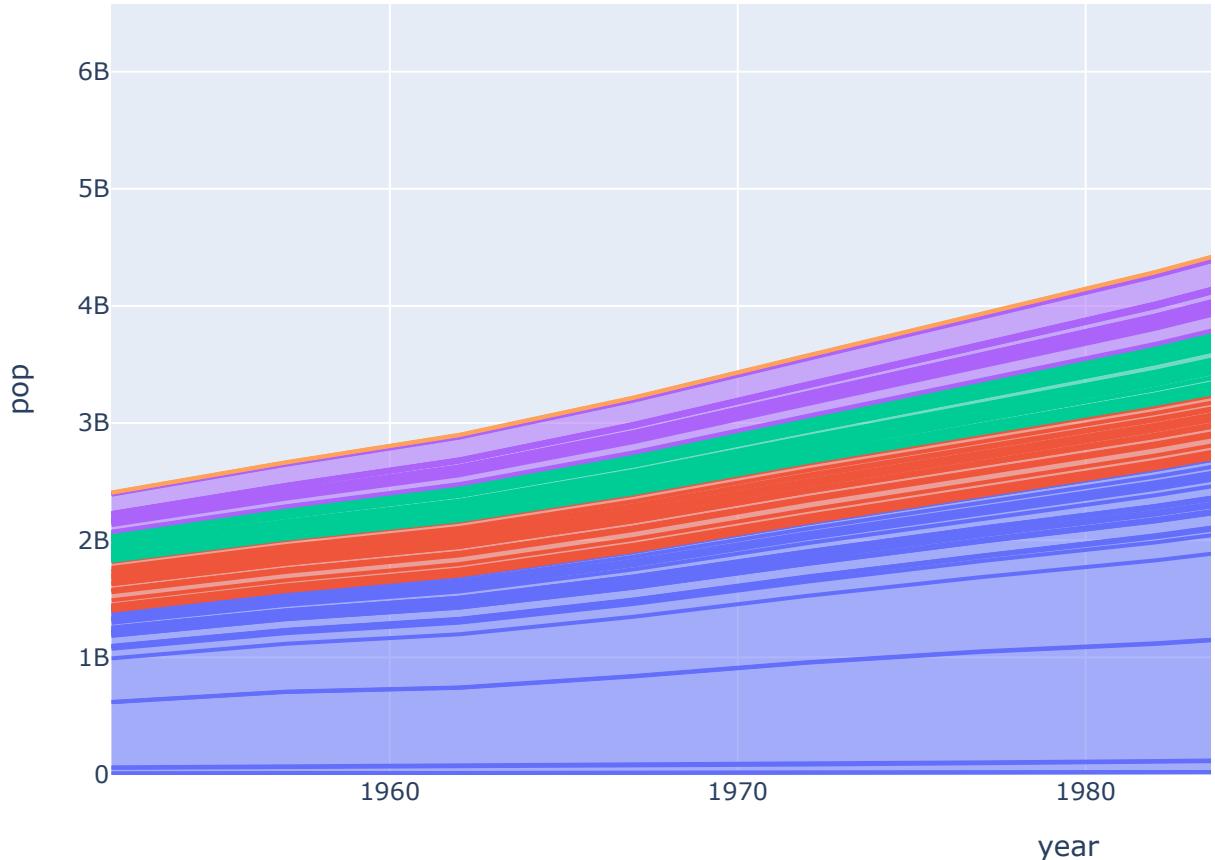


```
fig = go.Figure(data=[go.Candlestick(x=aapl['Date'],
                                      open=aapl['AAPL.Open'],
                                      high=aapl['AAPL.High'],
                                      low=aapl['AAPL.Low'],
                                      close=aapl['AAPL.Close'],
                                      increasing_line_color='red',
                                      decreasing_line_color='blue')])
fig.show()
```



## ▼ 면적도(Area Plots)

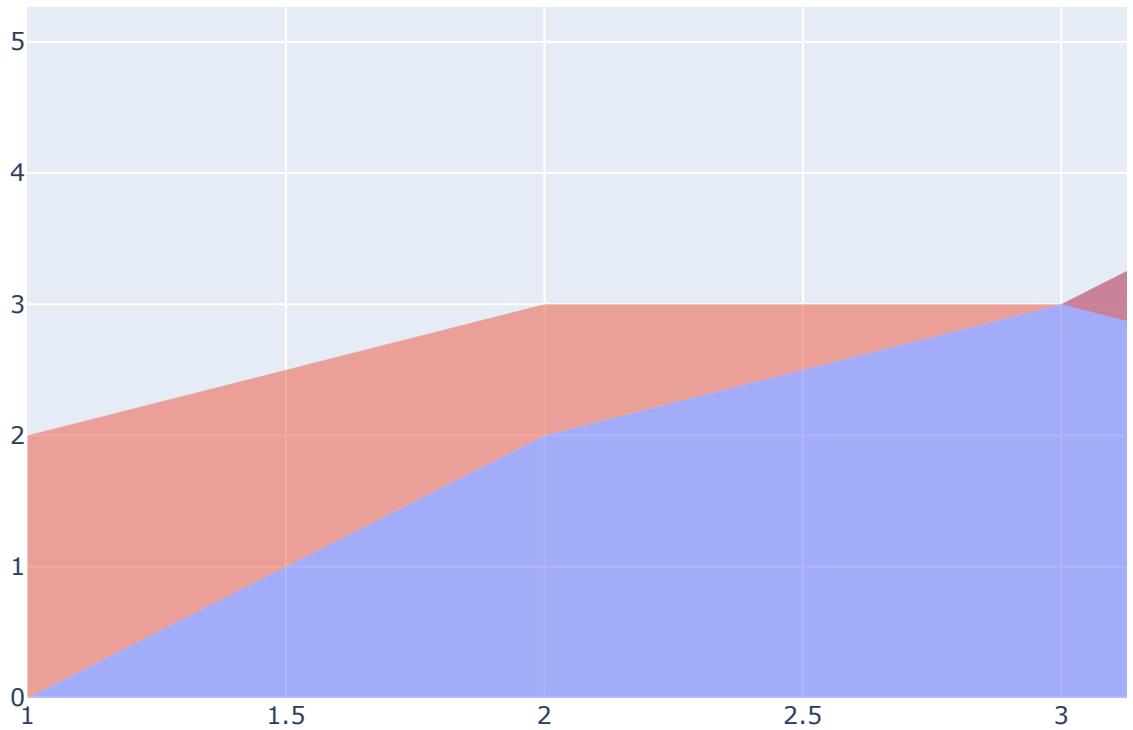
```
fig = px.area(gapminder, x='year', y='pop',
               color='continent', line_group='country')
fig.show()
```



```
fig = go.Figure()
fig.add_trace(go.Scatter(x=[1, 2, 3, 4, 5], y=[0, 2, 3, 5, 2], fill='tozeroy'))
fig.add_trace(go.Scatter(x=[1, 2, 3, 4, 5], y=[2, 3, 3, 2, 4], fill='tonexty'))
fig.show()
```



```
fig = go.Figure()
fig.add_trace(go.Scatter(x=[1, 2, 3, 4, 5], y=[0, 2, 3, 5, 2],
                         fill='tozeroY', mode='none'))
fig.add_trace(go.Scatter(x=[1, 2, 3, 4, 5], y=[2, 3, 3, 2, 4],
                         fill='tonextY', mode='none'))
fig.show()
```

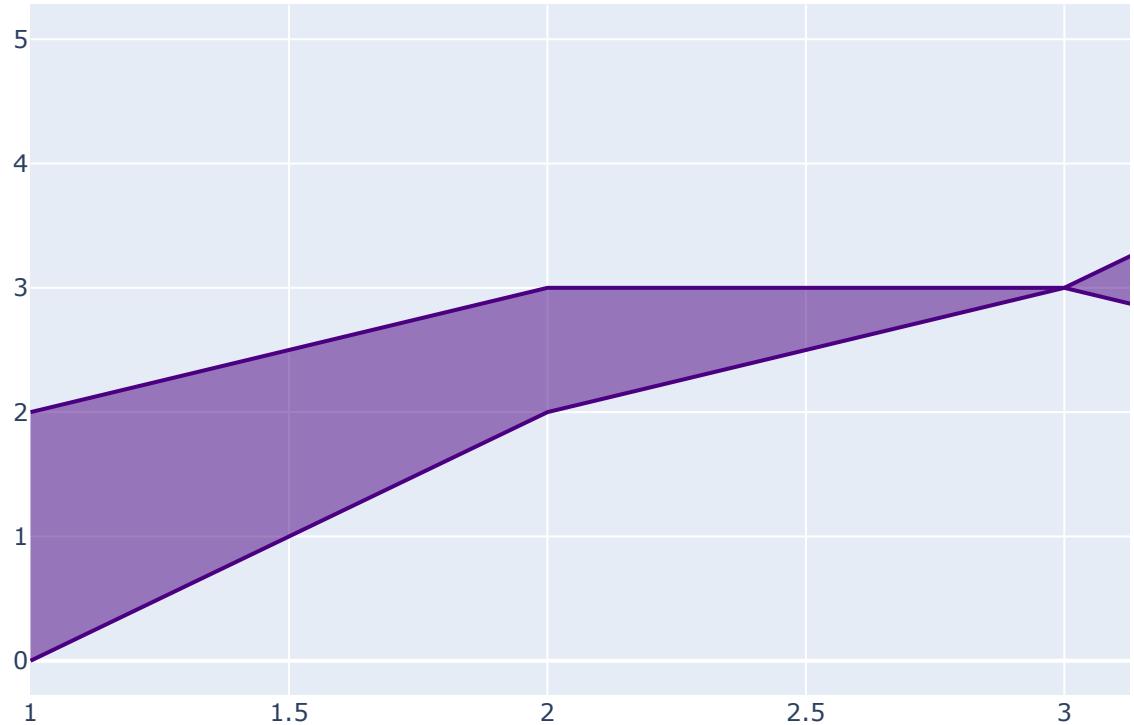


```
fig = go.Figure()
fig.add_trace(go.Scatter(x=[1, 2, 3, 4, 5], y=[0, 2, 3, 5, 2],
```

```

        fill='none', mode='lines', line_color='indigo'))
fig.add_trace(go.Scatter(x=[1, 2, 3, 4, 5], y=[2, 3, 3, 2, 4],
                         fill='tonexty', mode='lines', line_color='indigo'))
fig.show()

```

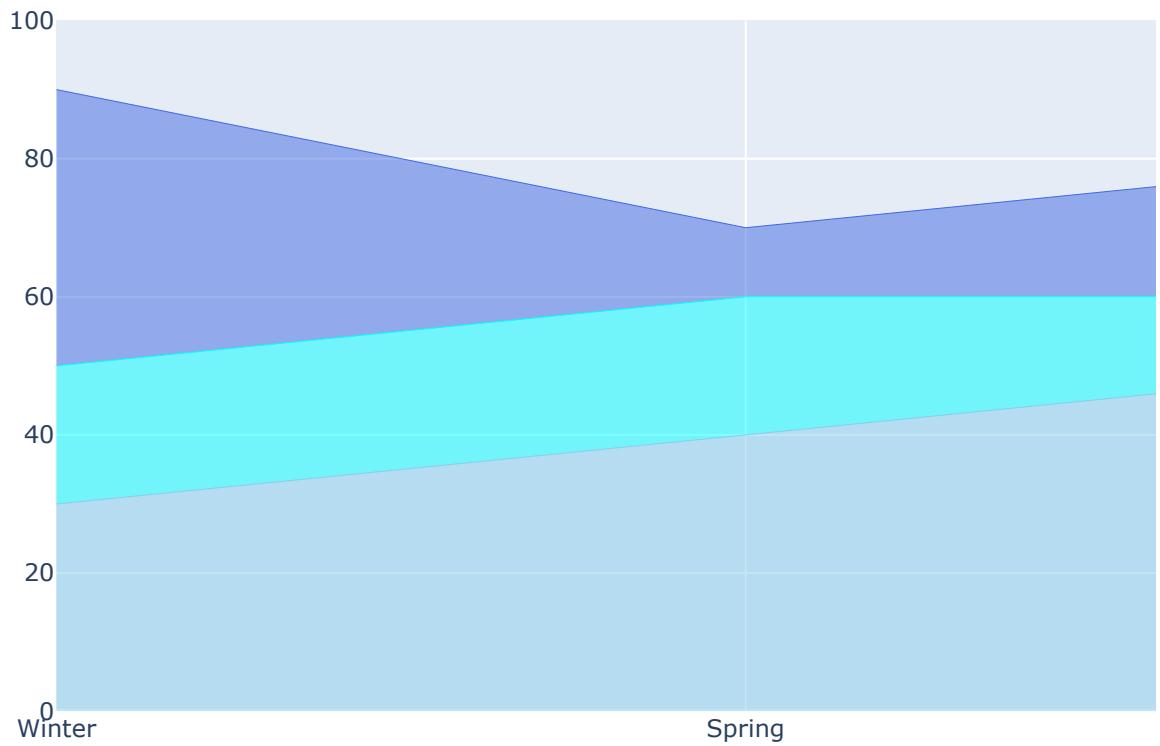


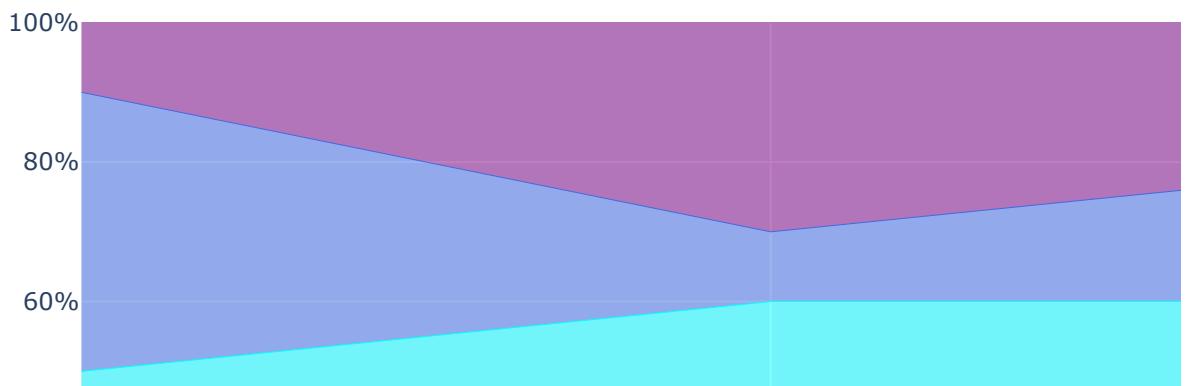
```

x = ['Winter', 'Spring', 'Summer', 'Fall']

fig = go.Figure()
fig.add_trace(go.Scatter(x=x, y=[30, 40, 50, 30],
                         hover info='x+y',
                         mode='lines',
                         line=dict(width=0.5, color='skyblue'),
                         stackgroup='one'))
fig.add_trace(go.Scatter(x=x, y=[20, 20, 10, 20],
                         hover info='x+y',
                         mode='lines',
                         line=dict(width=0.5, color='cyan'),
                         stackgroup='one'))
fig.add_trace(go.Scatter(x=x, y=[40, 10, 20, 10],
                         hover info='x+y',
                         mode='lines',
                         line=dict(width=0.5, color='royalblue'),
                         stackgroup='one'))
fig.update_layout(yaxis_range=(0, 100))
fig.show()

```



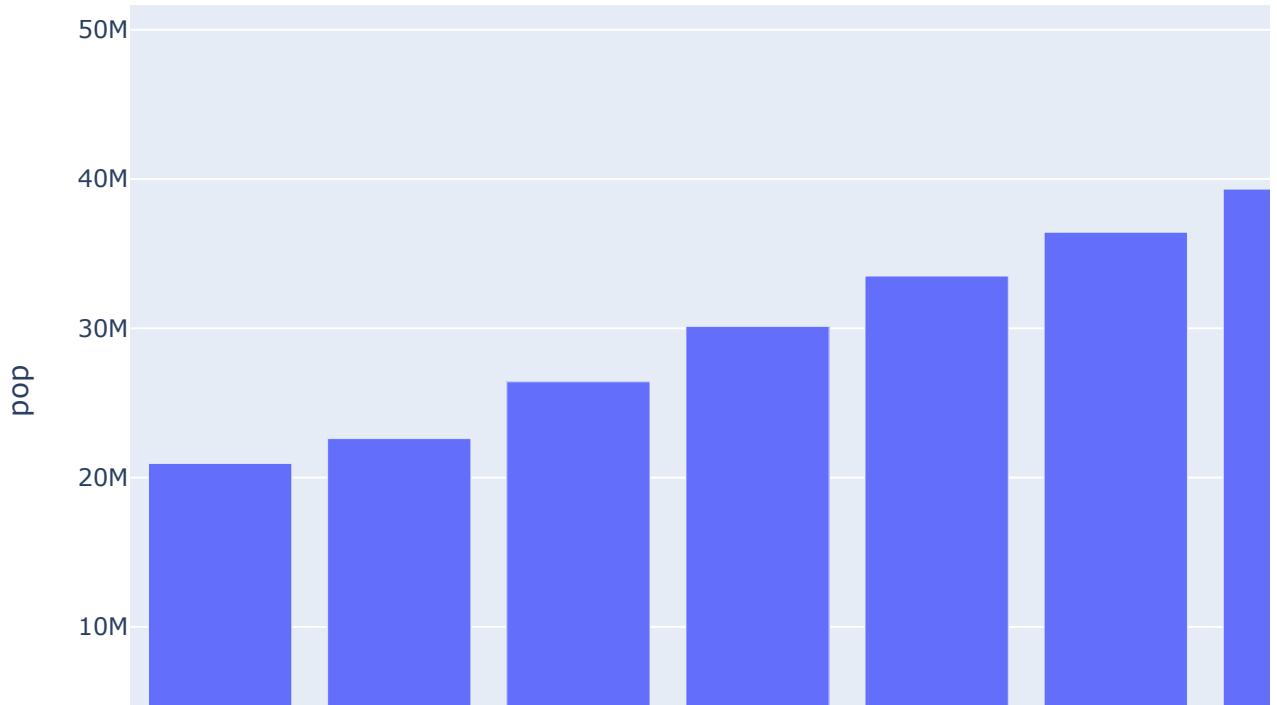


## ▼ 막대 차트(Bar Charts)

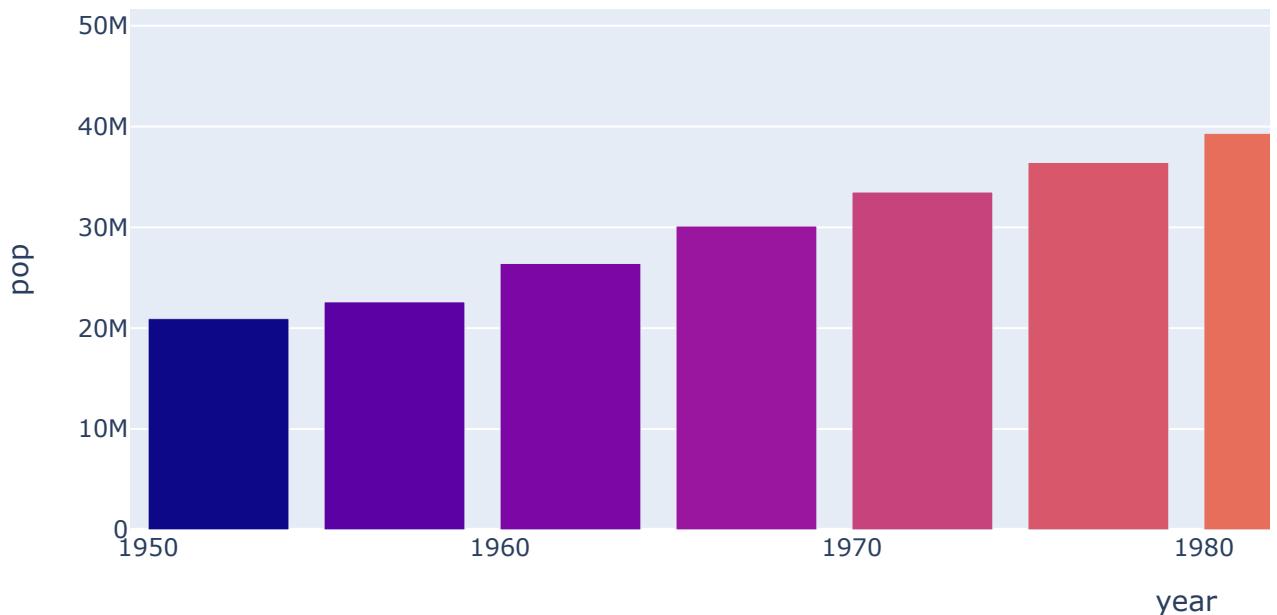
```
gapminder_korea = gapminder.query("country == 'Korea, Rep.'")
gapminder_korea
```

	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
840	Korea, Rep.	Asia	1952	47.453	20947571	1030.592226	KOR	410
841	Korea, Rep.	Asia	1957	52.681	22611552	1487.593537	KOR	410
842	Korea, Rep.	Asia	1962	55.292	26420307	1536.344387	KOR	410
843	Korea, Rep.	Asia	1967	57.716	30131000	2029.228142	KOR	410
844	Korea, Rep.	Asia	1972	62.612	33505000	3030.876650	KOR	410
845	Korea, Rep.	Asia	1977	64.766	36436000	4657.221020	KOR	410
846	Korea, Rep.	Asia	1982	67.123	39326000	5622.942464	KOR	410
847	Korea, Rep.	Asia	1987	69.810	41622000	8533.088805	KOR	410

```
fig = px.bar(gapminder_korea, x='year', y='pop')
fig.show()
```



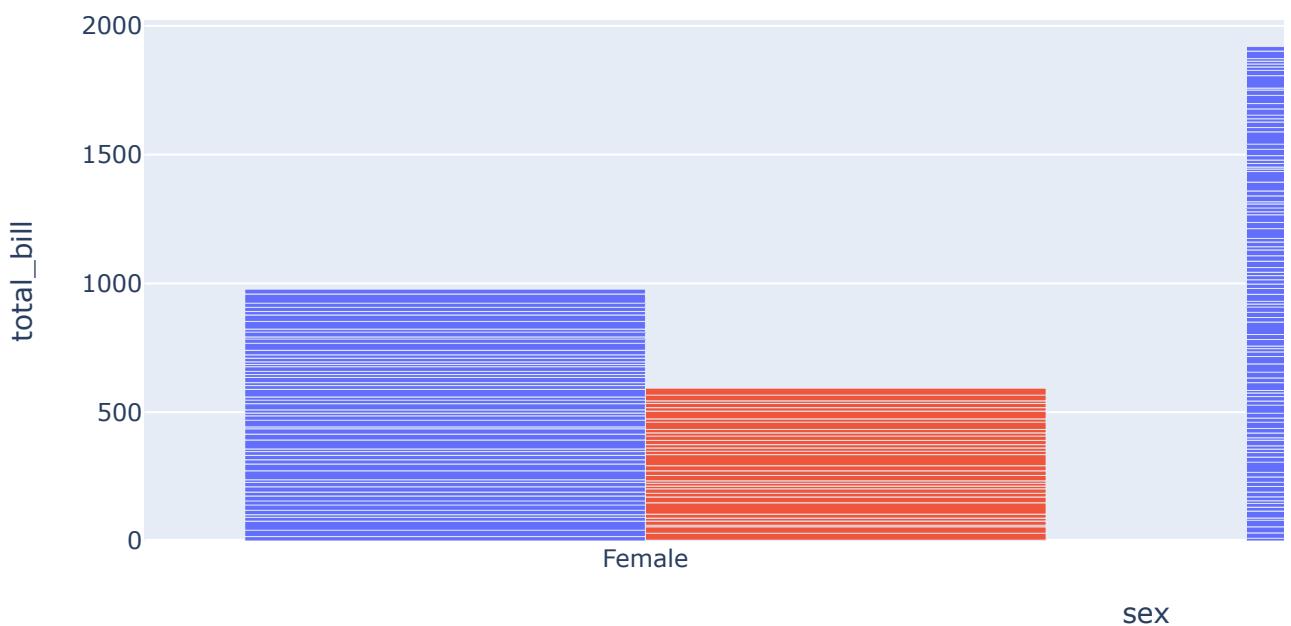
```
fig = px.bar(gapminder_korea, x='year', y='pop',
              hover_data=['lifeExp', 'gdpPer cap'],
              color='lifeExp', height=400)
fig.show()
```



```
fig = px.bar(tips, x='sex', y='total_bill', color='time')
fig.show()
```



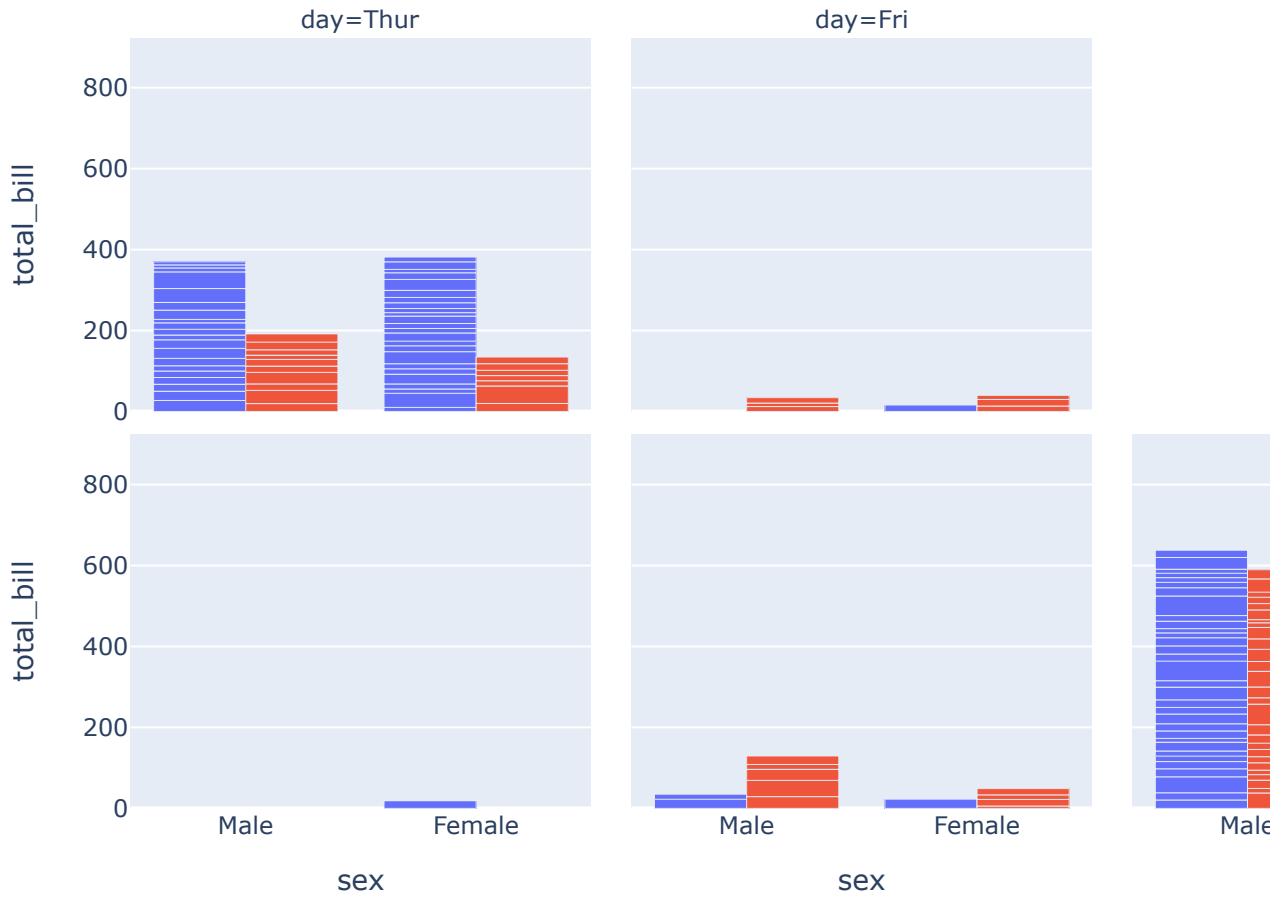
```
fig = px.bar(tips, x='sex', y='total_bill',
              color='smoker', barmode='group',
              height=400)
fig.show()
```



```

fig = px.bar(tips, x='sex', y='total_bill',
              color='smoker', barmode='group',
              facet_row='time', facet_col='day',
              category_orders={'day': ['Thur', 'Fri', 'Sat', 'Sun'],
                               'time': ['Lunch', 'Dinner']})
fig.show()

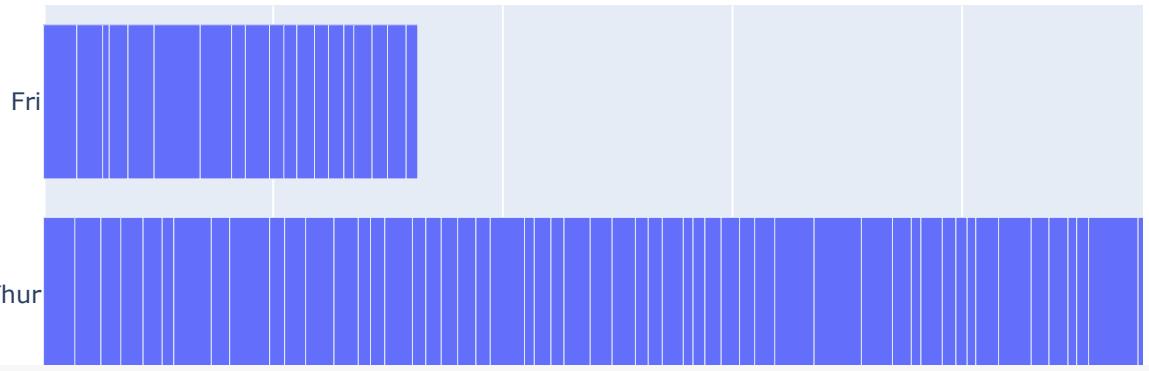
```



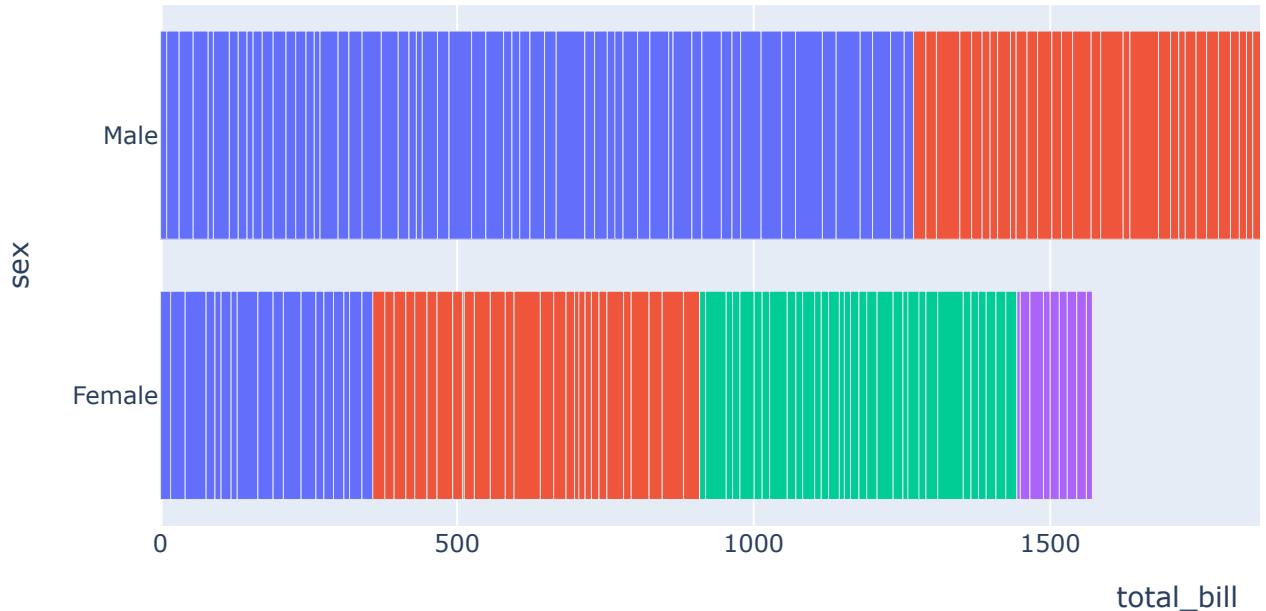
```

fig = px.bar(tips, x='total_bill', y='day', orientation='h')
fig.show()

```

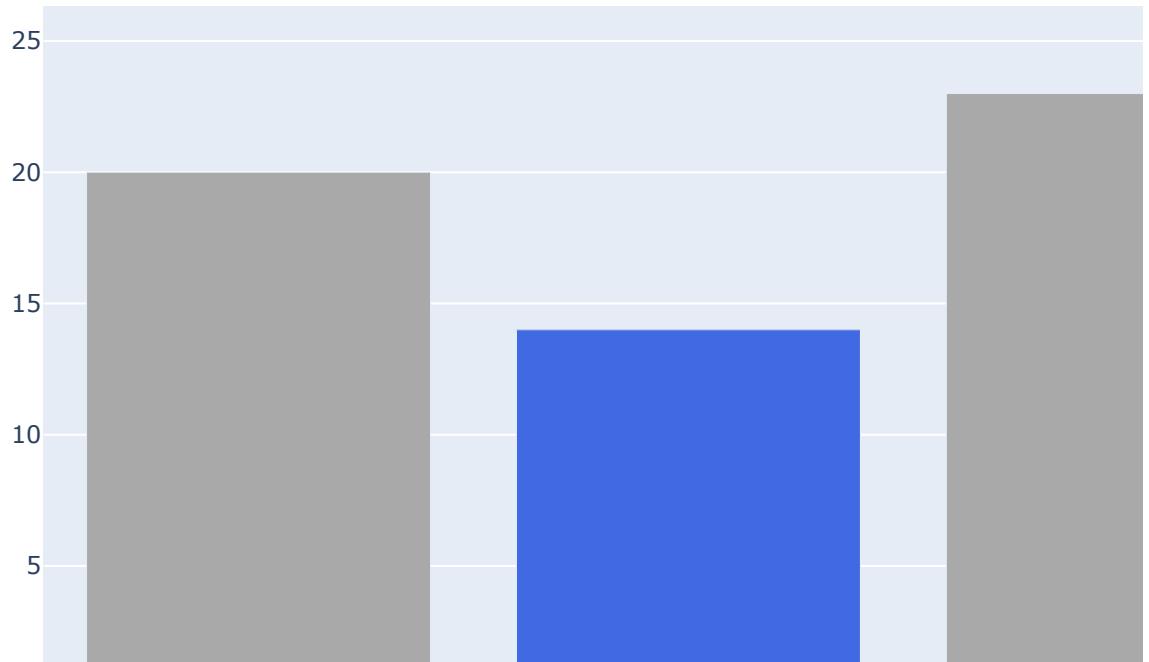


```
fig = px.bar(tips, x='total_bill', y='sex', color='day', orientation='h',
              hover_data=['tip', 'size'], height=400)
fig.show()
```



```
colors = ['darkgray'] * 5
colors[1] = 'royalblue'
colors[3] = 'limegreen'

fig = go.Figure(data=[go.Bar(x=['A', 'B', 'C', 'D', 'E'],
                             y=[20, 14, 23, 25, 22],
                             marker_color=colors)])
fig.update_layout()
```



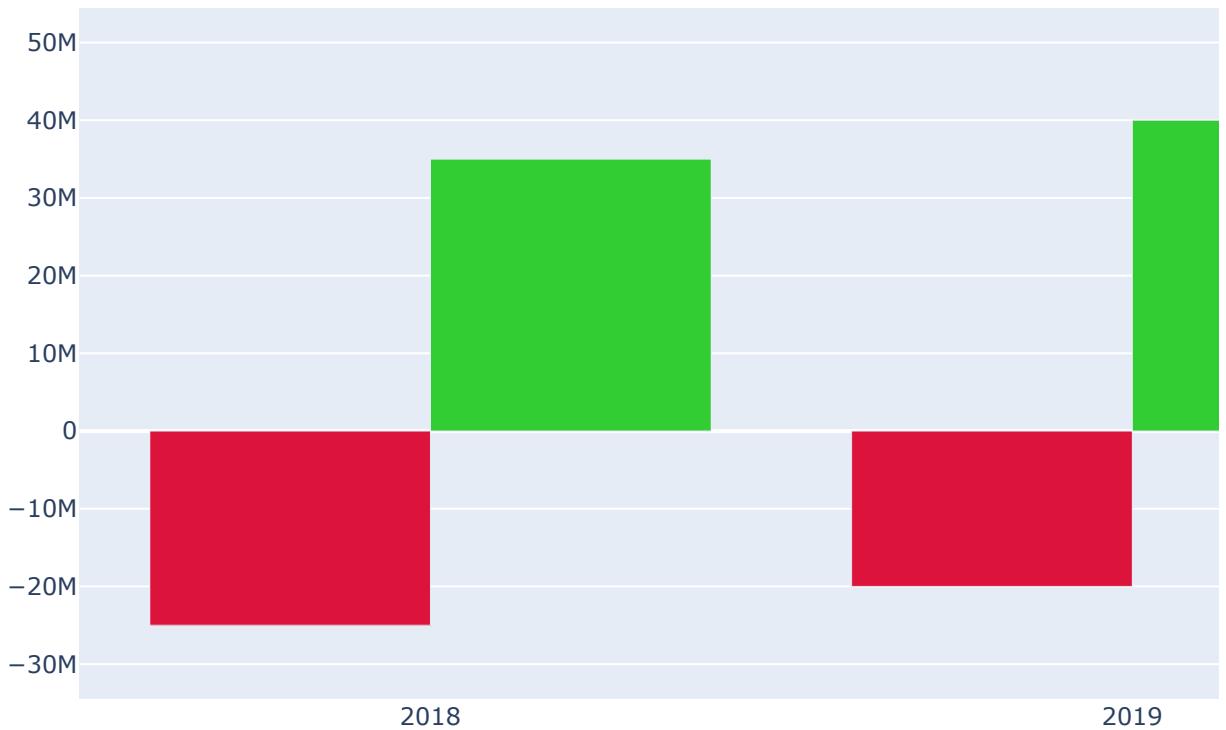
```
fig = go.Figure(data=[go.Bar(x=[1, 3, 5.1, 6.9, 9.5],  
                             y=[6, 8, 3, 5, 4],  
                             width=[1, 2.1, 0.8, 2.6, 1.4]))  
fig.show()
```

```

years = ['2018', '2019', '2020']

fig = go.Figure()
fig.add_trace(go.Bar(x=years, y=[25000000, 20000000, 30000000],
                     base=[-25000000, -20000000, -30000000],
                     marker_color='crimson',
                     name='지출'))
fig.add_trace(go.Bar(x=years, y=[35000000, 40000000, 50000000],
                     base=0,
                     marker_color='limegreen',
                     name='수입'))
fig.show()

```



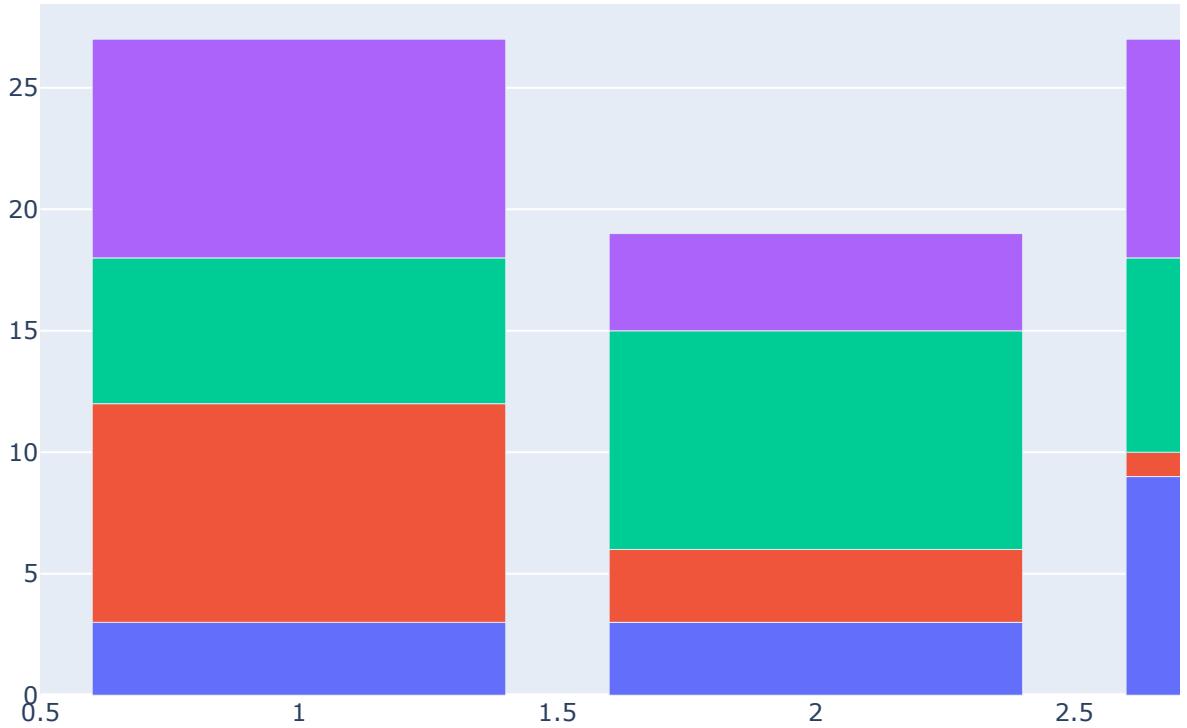
```

x = [1, 2, 3, 4]
y1 = np.random.randint(1, 10, 5)
y2 = np.random.randint(1, 10, 5)
y3 = np.random.randint(1, 10, 5)
y4 = np.random.randint(1, 10, 5)

fig = go.Figure()
fig.add_trace(go.Bar(x=x, y=y1))
fig.add_trace(go.Bar(x=x, y=y2))
fig.add_trace(go.Bar(x=x, y=y3))

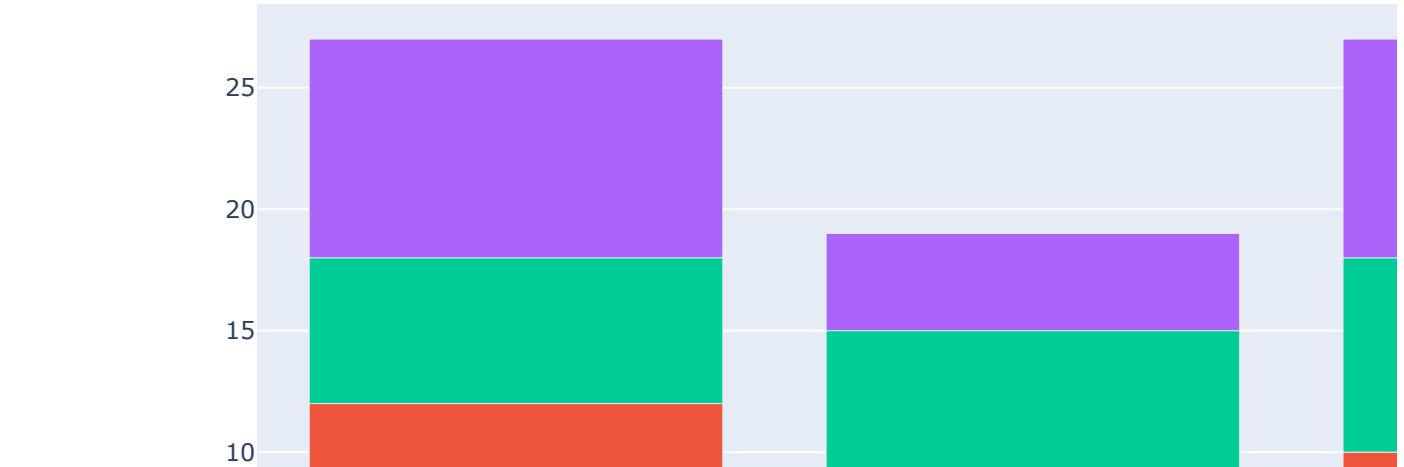
```

```
fig.add_trace(go.Bar(x=x, y=y4))
fig.update_layout(barmode='relative')
fig.show()
```



```
mx = [['A', 'A', 'B', 'B'],
       [5, 10, 15, 20]]

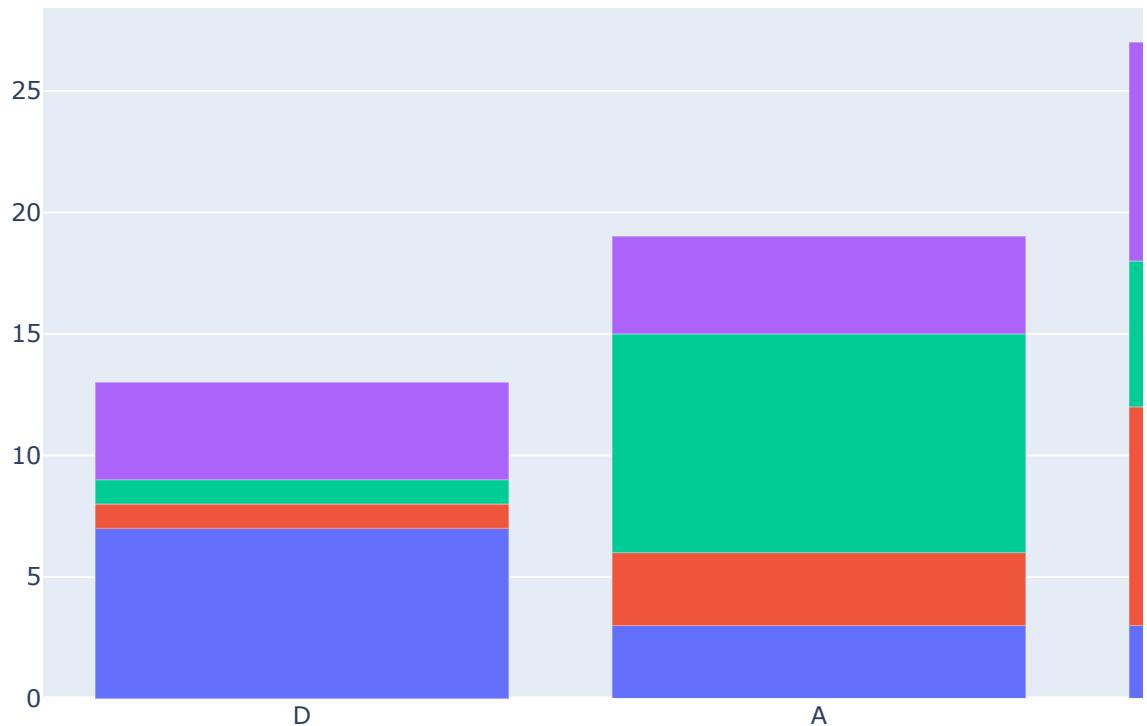
fig = go.Figure()
fig.add_trace(go.Bar(x=mx, y=y1))
fig.add_trace(go.Bar(x=mx, y=y2))
fig.add_trace(go.Bar(x=mx, y=y3))
fig.add_trace(go.Bar(x=mx, y=y4))
fig.update_layout(barmode='relative')
fig.show()
```



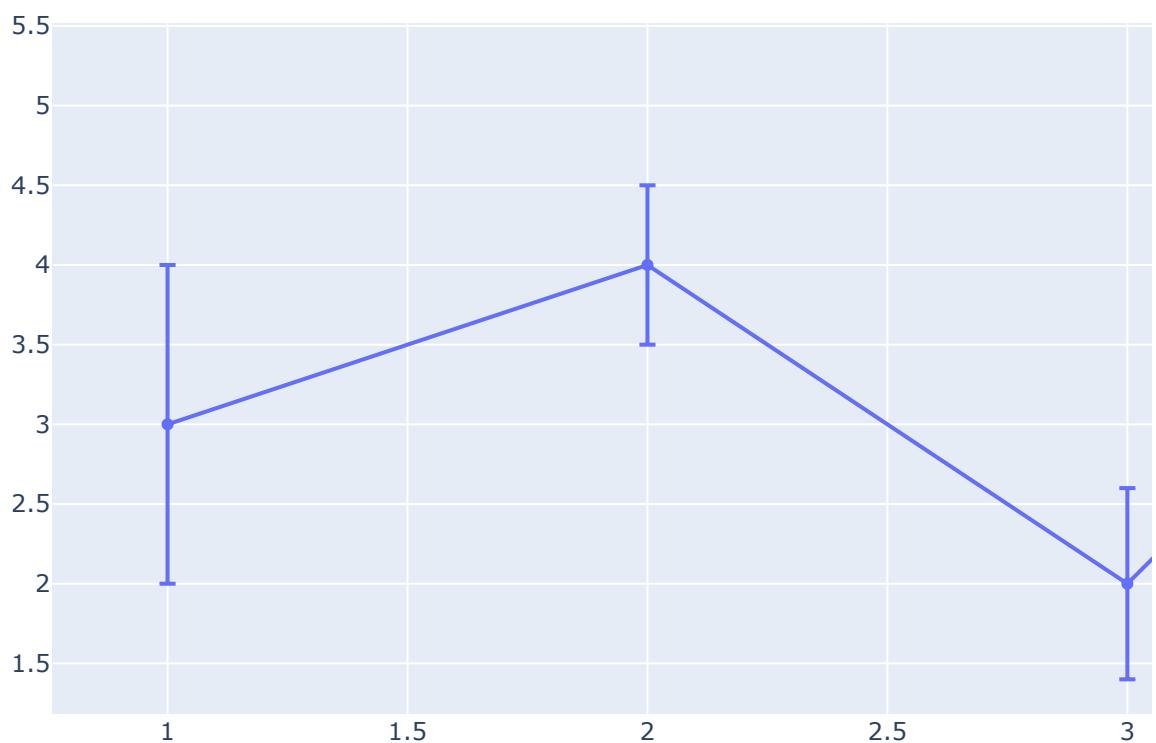
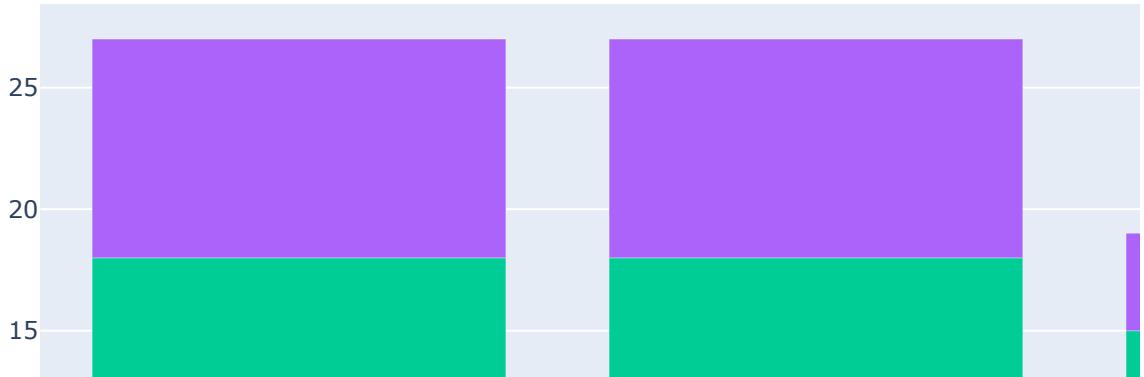
```
a = ['C', 'A', 'B', 'D']

fig = go.Figure()
fig.add_trace(go.Bar(x=a, y=y1))
fig.add_trace(go.Bar(x=a, y=y2))
fig.add_trace(go.Bar(x=a, y=y3))
fig.add_trace(go.Bar(x=a, y=y4))
fig.update_layout(barmode='stack',
                  xaxis={'categoryorder ':'category ascending'})
fig.show()
```

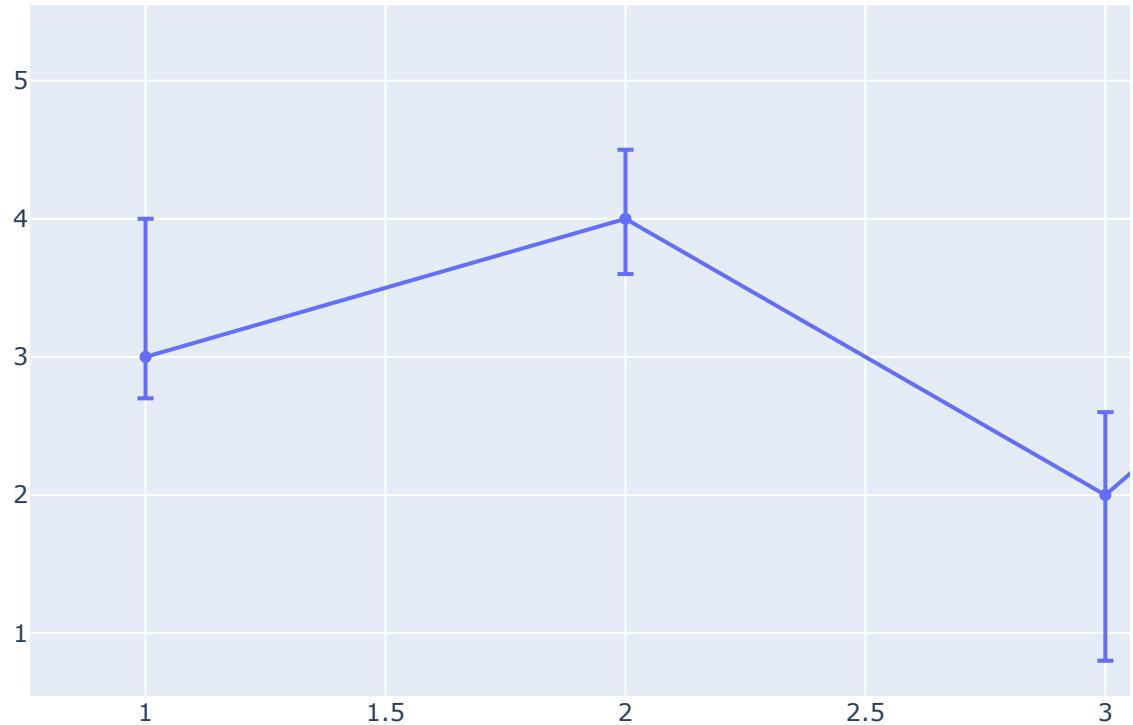
```
fig.update_layout(barmode='stack',
                  xaxis={'categoryorder':'array',
                         'categoryarray':['D', 'A', 'C', 'B']})
fig.show()
```



```
fig.update_layout(barmode='stack',
                  xaxis={'categoryorder':'total_descending'})
fig.show()
```

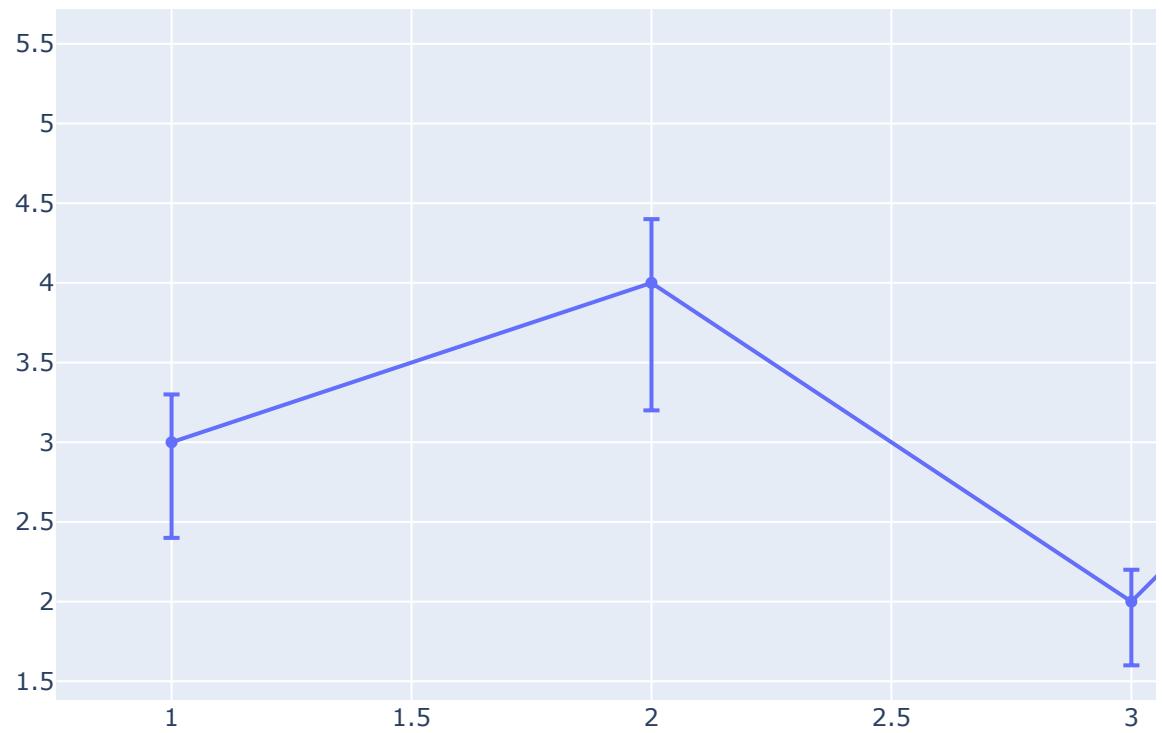


```
array=[1, 0.5, 0.6, 0.3, 0.5],  
arrayminus=[0.3, 0.4, 1.2, 0.4, 1],  
visible=True)))  
fig.show()
```



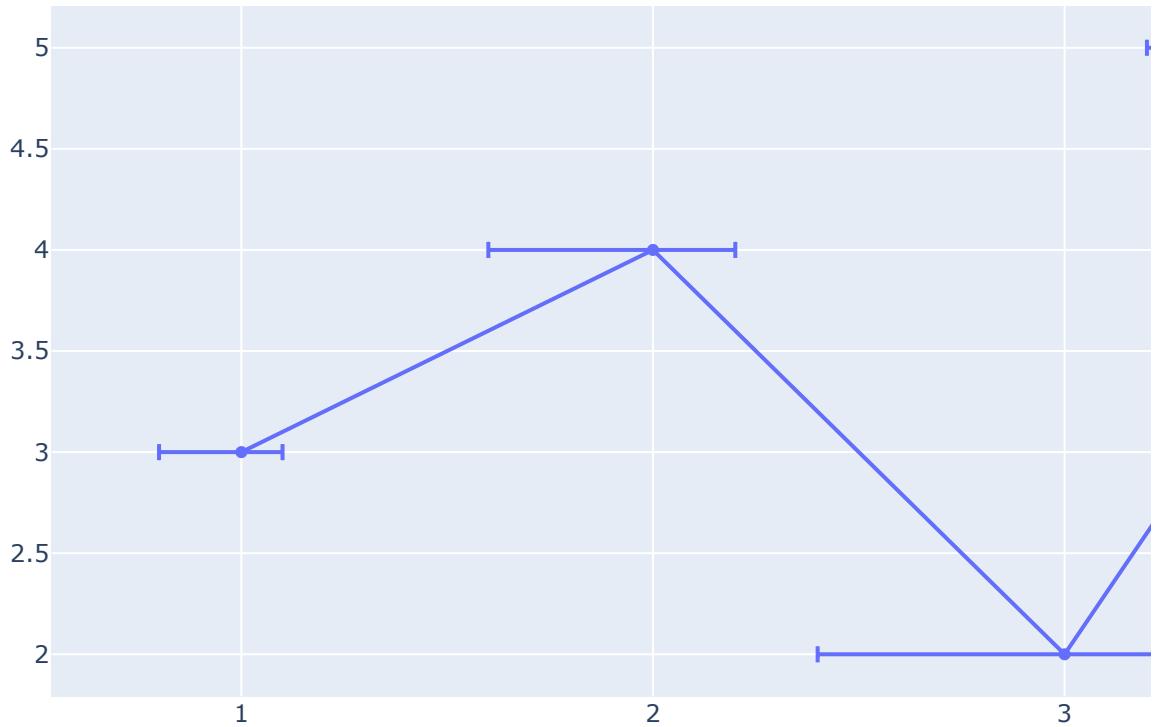


```
fig = go.Figure(data=go.Scatter(x=[1, 2, 3, 4, 5],  
                                 y=[3, 4, 2, 5, 2],  
                                 error_y=dict(type='percent',  
                                              symmetric=False,  
                                              value=10,  
                                              valueminus=20)))  
fig.show()
```

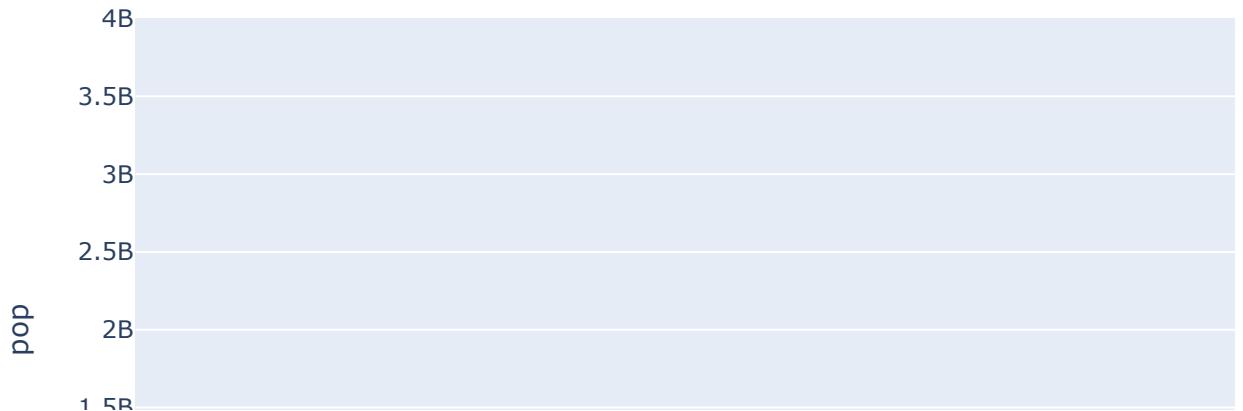


```
fig = go.Figure(data=go.Scatter(x=[1, 2, 3, 4, 5],  
                                 y=[3, 4, 2, 5, 2],
```

```
        error_x=dict(type='percent',
                      symmetric=False,
                      value=10,
                      valueminus=20)))
fig.show()
```

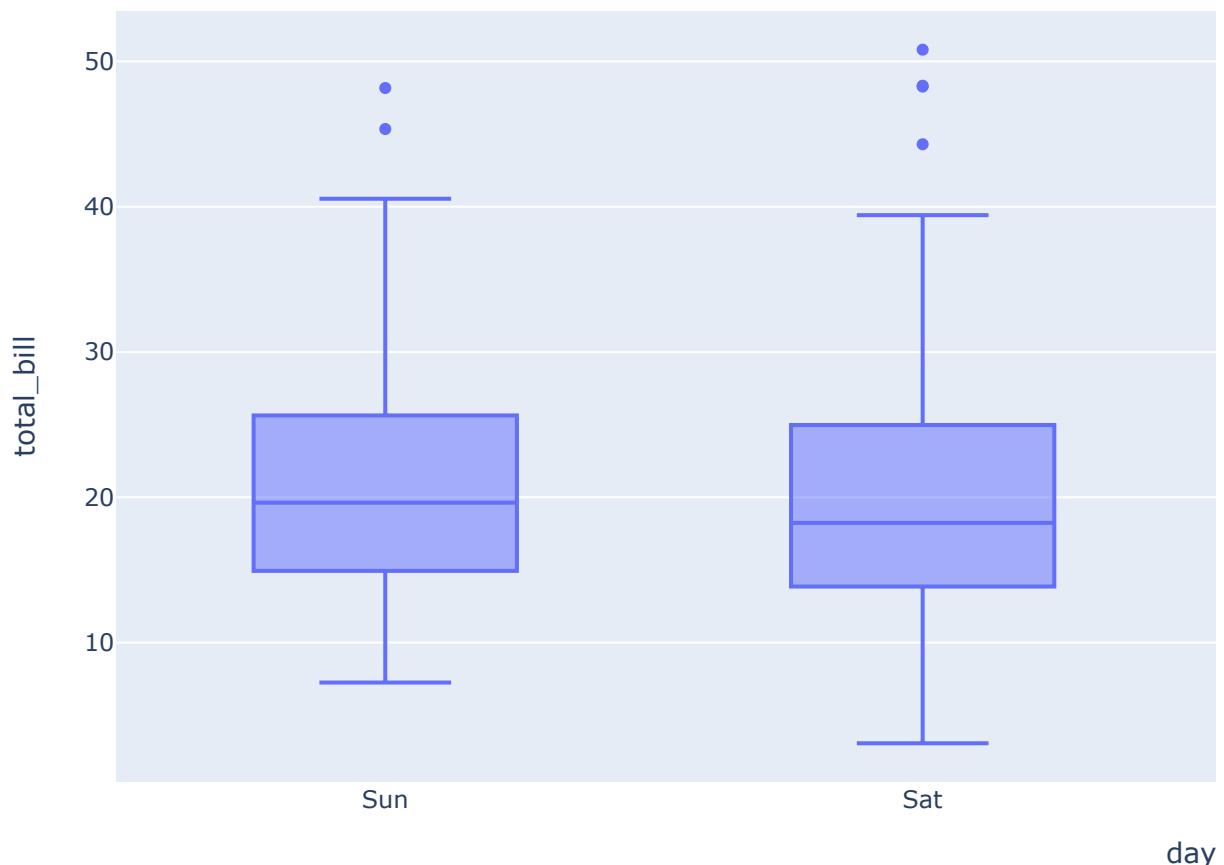


```
fig = px.bar(gapminder, x='continent', y='pop', color='continent',
              animation_frame='year', animation_group='country',
              range_y=[0, 4000000000])
fig.show()
```

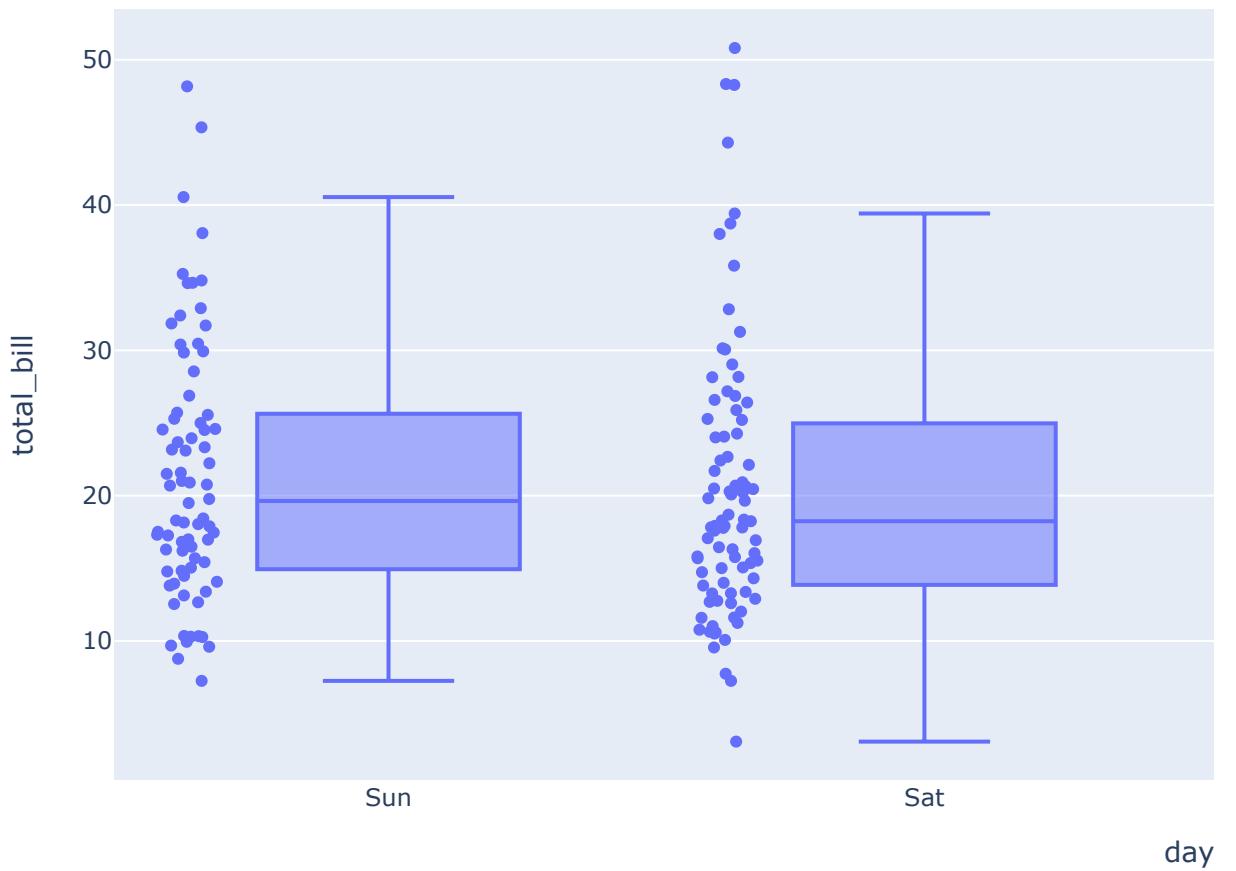


## ▼ 박스 플롯(Box Plots)

```
fig = px.box(tips, x='day', y='total_bill')
fig.show()
```

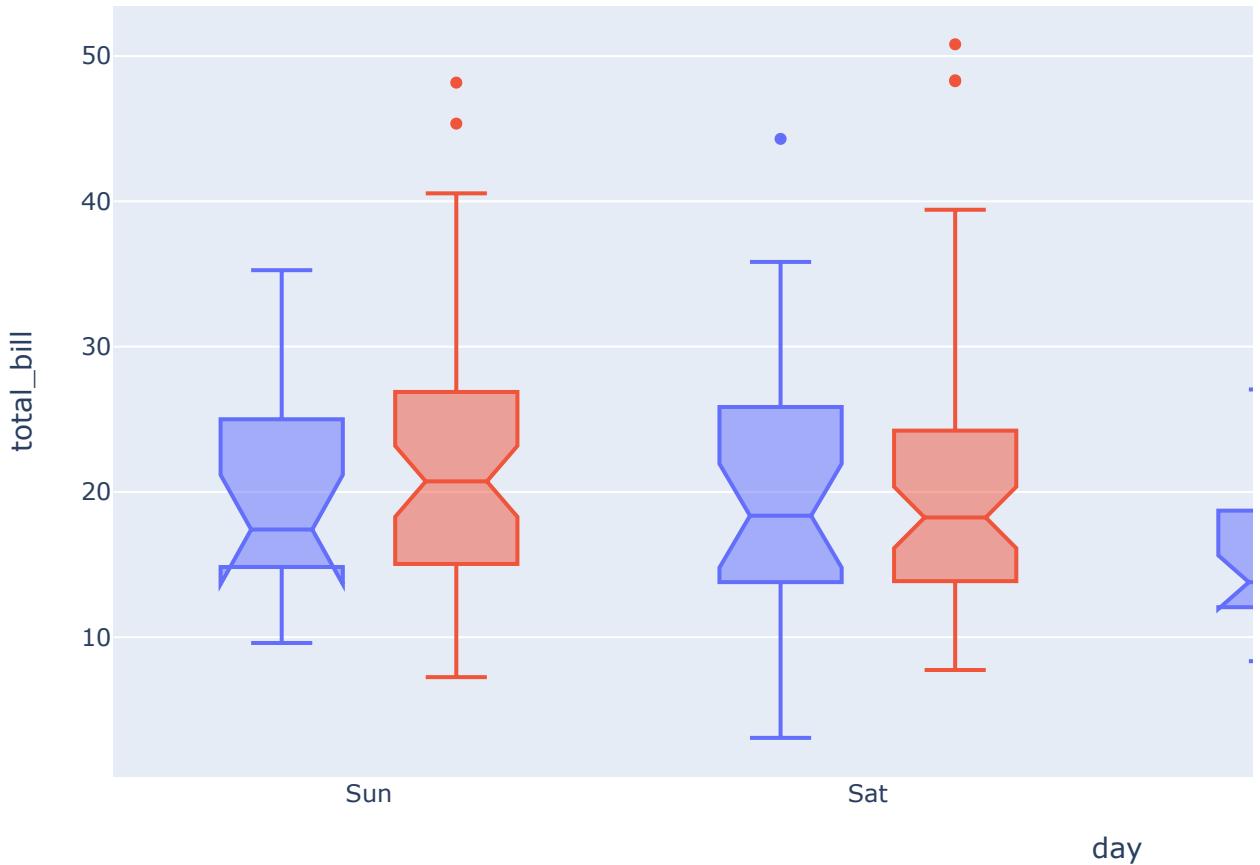


```
fig = px.box(tips, x='day', y='total_bill', points='all')
fig.show()
```



```
fig = px.box(tips, x='day', y='total_bill', color='sex')
fig.show()
```

```
fig = px.box(tips, x='day', y='total_bill', color='sex',
              notched=True, hover_data=['time'])
fig.show()
```



```
x1 = np.random.randn(50)
x2 = np.random.randn(50) + 5

fig = go.Figure()
fig.add_trace(go.Box(x=x1))
fig.add_trace(go.Box(x=x2))
fig.show()
```

trace 1

trace 0

```
fig = go.Figure()
fig.add_trace(go.Box(x=x1))
fig.add_trace(go.Box(x=x2, boxpoints='all', jitter=0.3, pointpos=-1.8))
fig.show()
```

trace 1

trace 0

-2

-1

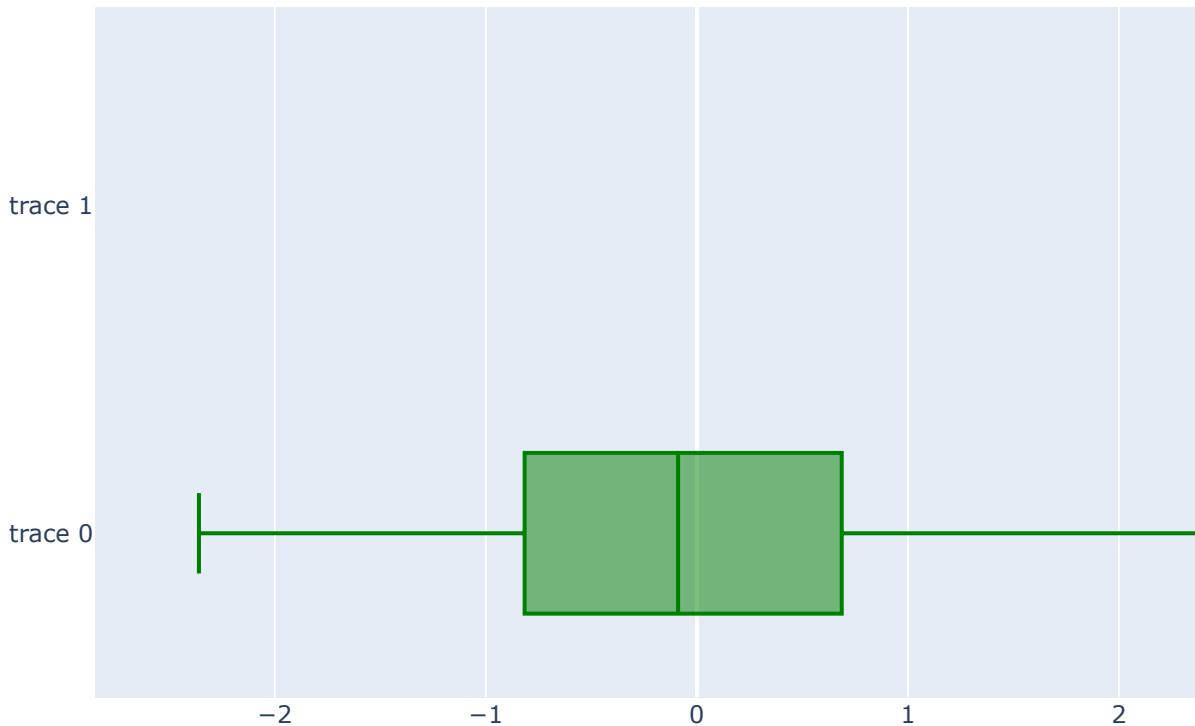
0

1

2

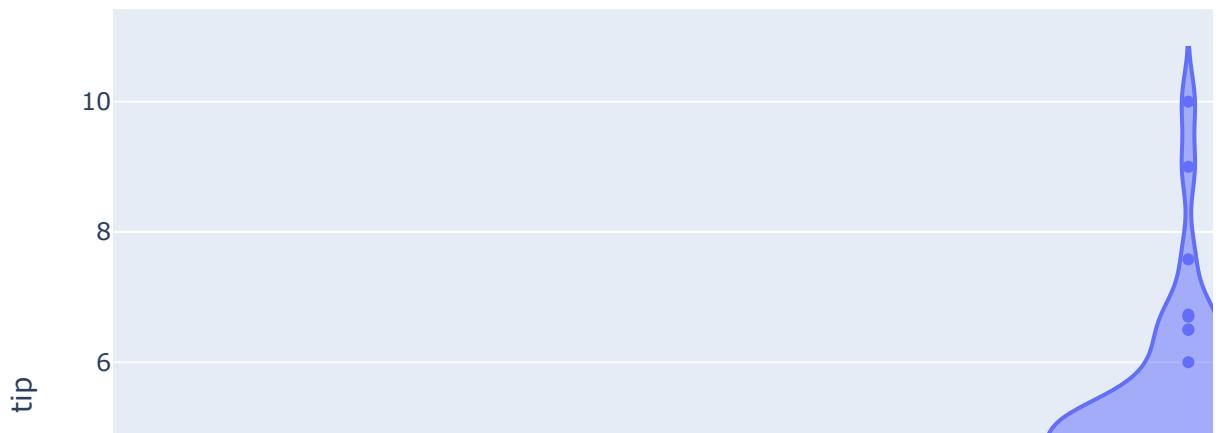
```
fig = go.Figure()
```

```
...  
fig.add_trace(go.Box(x=x1, marker_color='olive', line_color='green'))  
fig.add_trace(go.Box(x=x2, marker_color='skyblue', line_color='royalblue',  
                     boxpoints='all', jitter=0.3, pointpos=-1.8))  
fig.show()
```

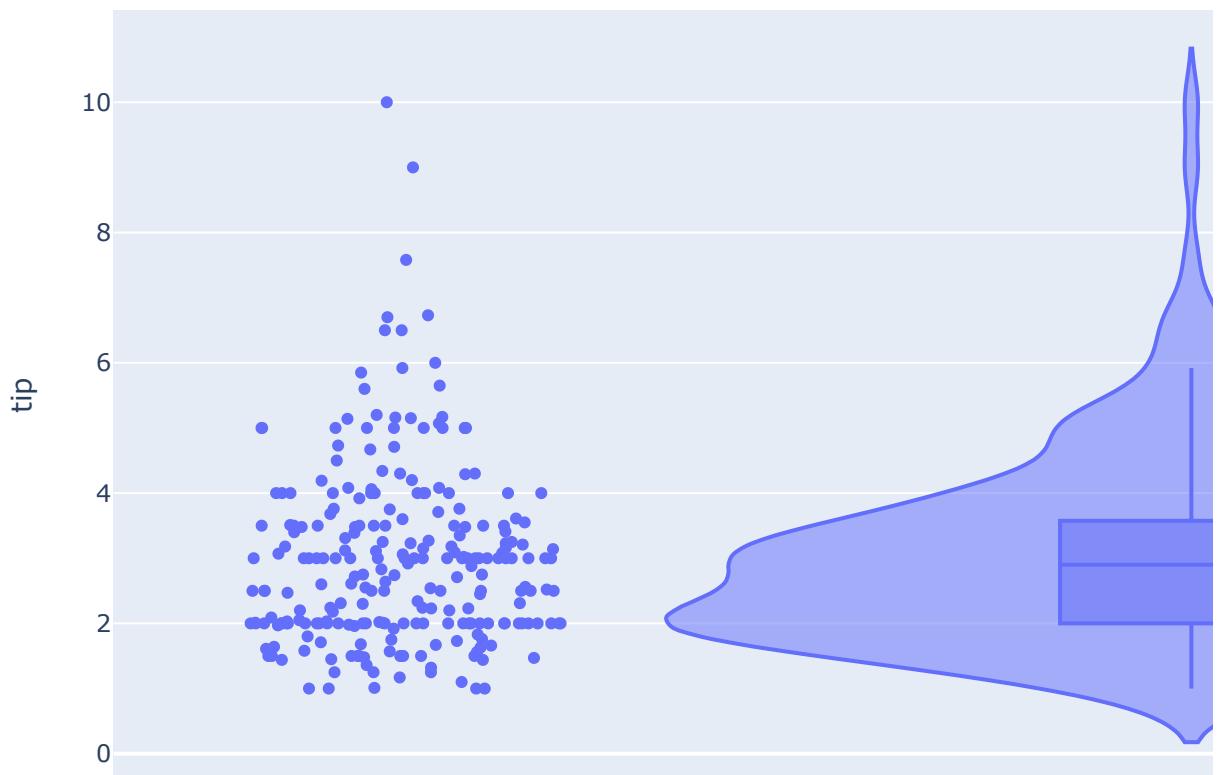


## ▼ 바이올린 플롯(Violin Plot)

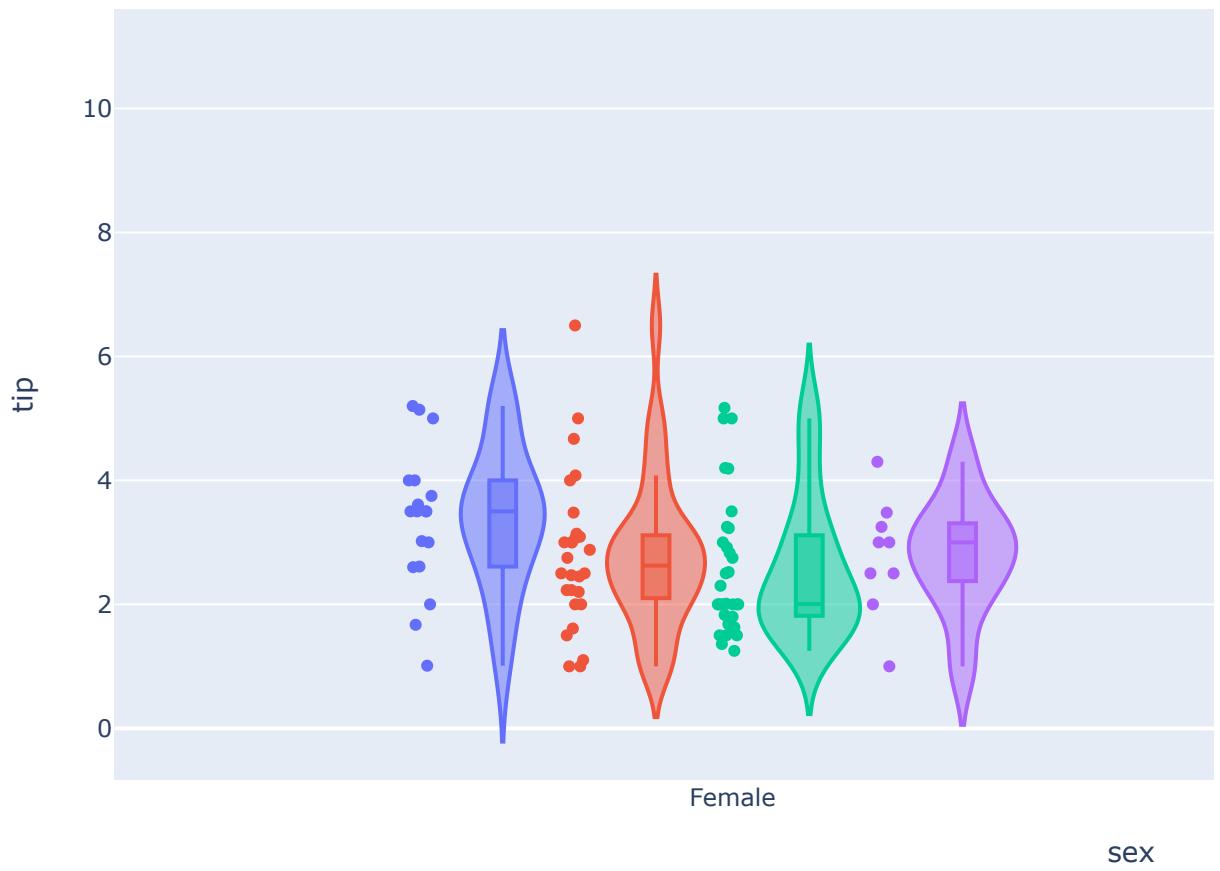
```
fig = px.violin(tips, y='tip')  
fig.show()
```



```
fig = px.violin(tips, y='tip', box=True, points='all')
fig.show()
```



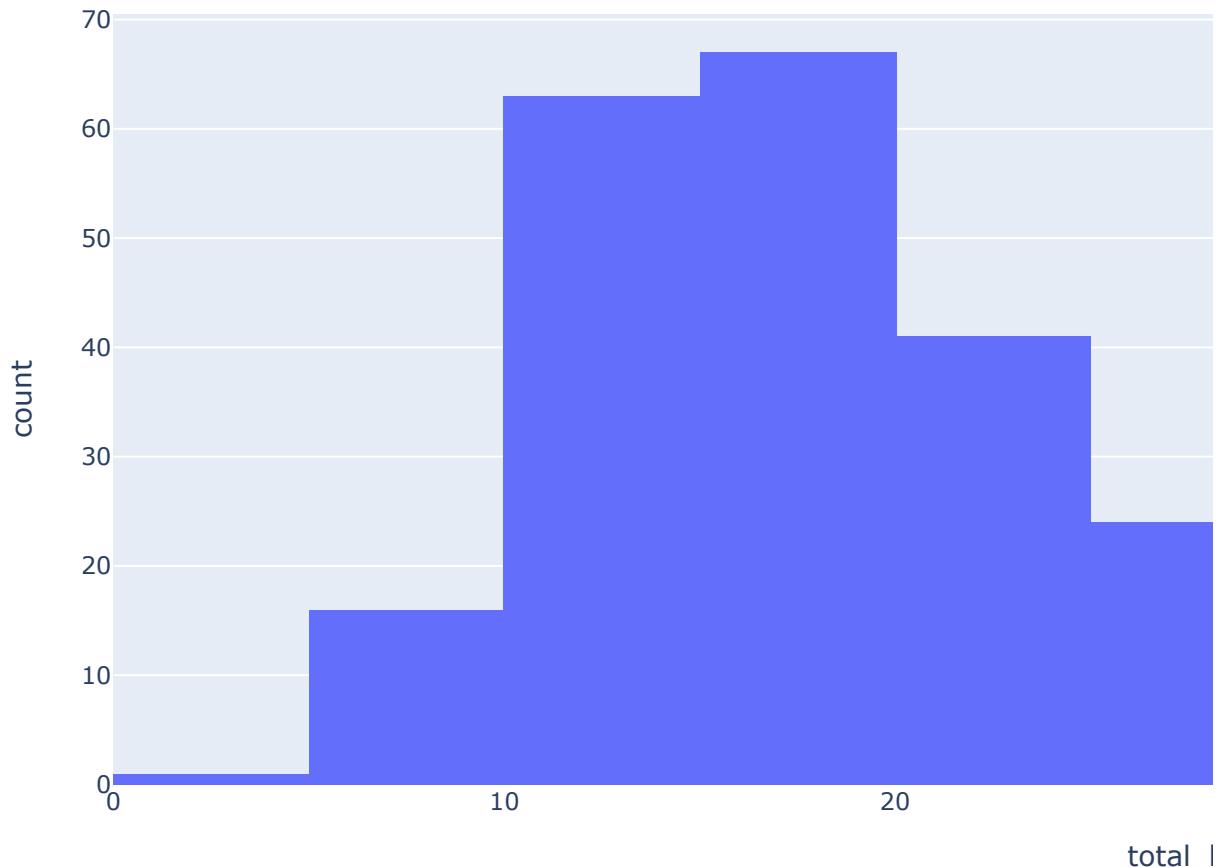
```
fig = px.violin(tips, y='tip', x='sex',
                  color='day', box=True, points='all',
                  hover_data=tips.columns)
fig.show()
```



## ▼ 히스토그램(Histogram)

```
fig = px.histogram(tips, x='total_bill')
fig.show()
```

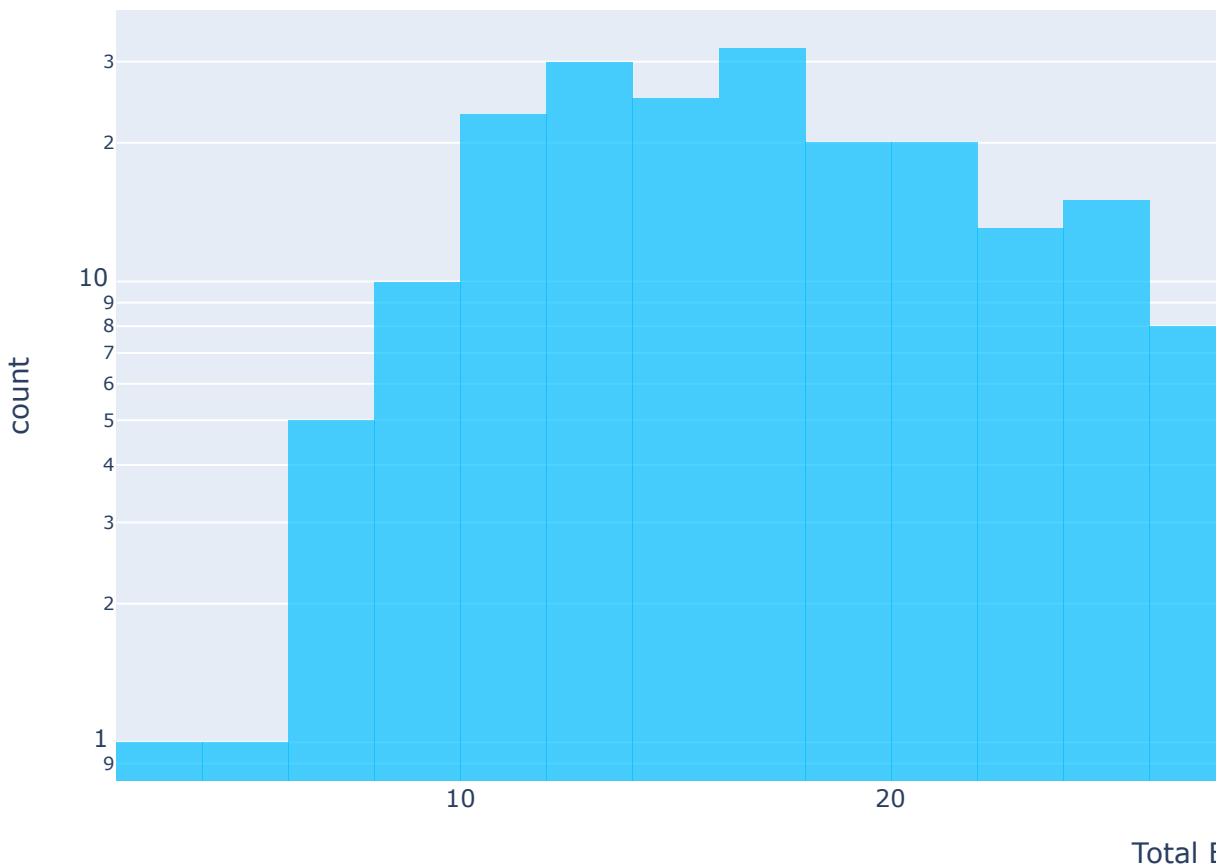
```
fig = px.histogram(tips, x='total_bill', nbins=20)  
fig.show()
```



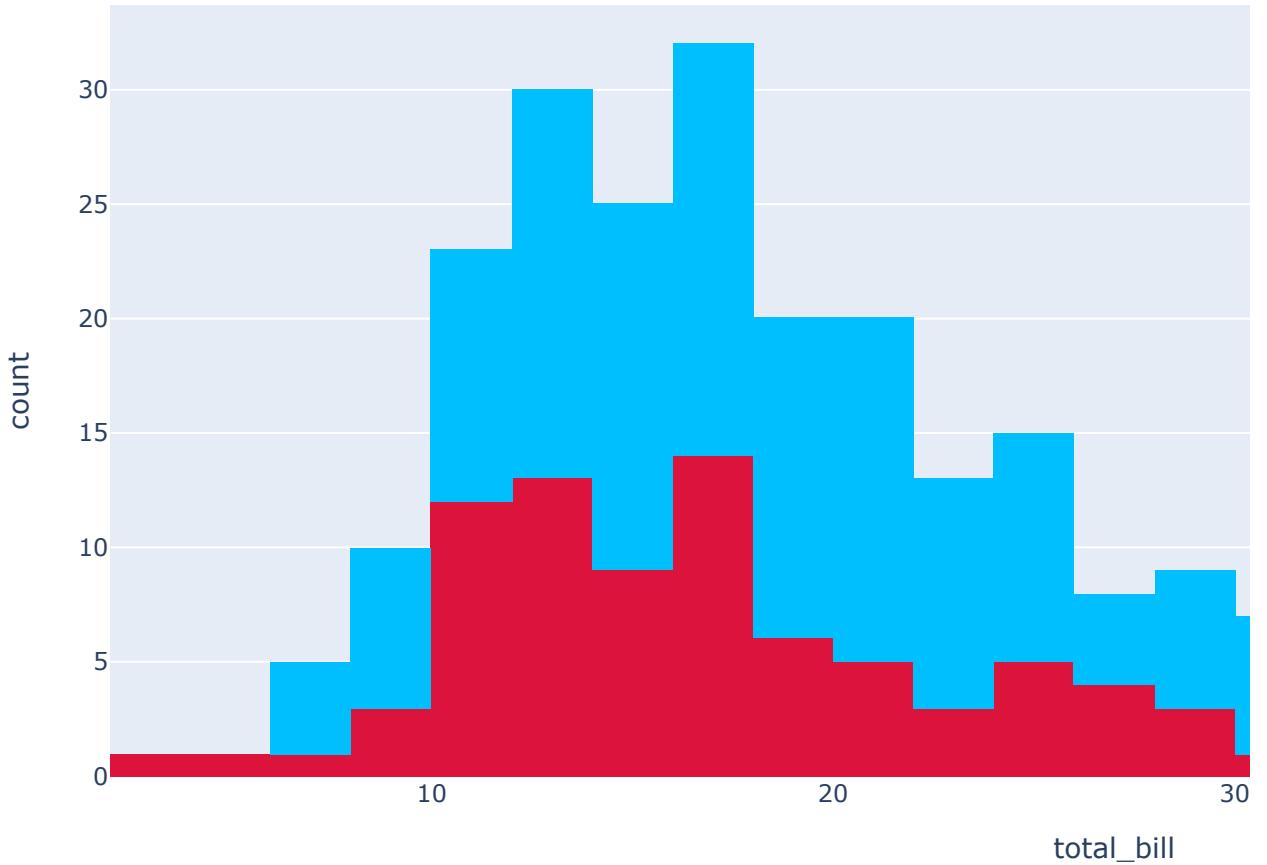
```
fig = px.histogram(tips, x='total_bill', histnorm='probability density')  
fig.show()
```



```
fig = px.histogram(tips, x='total_bill',
                   labels={'total_bill':'Total Bill'},
                   opacity=0.7, log_y=True,
                   color_discrete_sequence=[ 'deepskyblue' ])
fig.show()
```

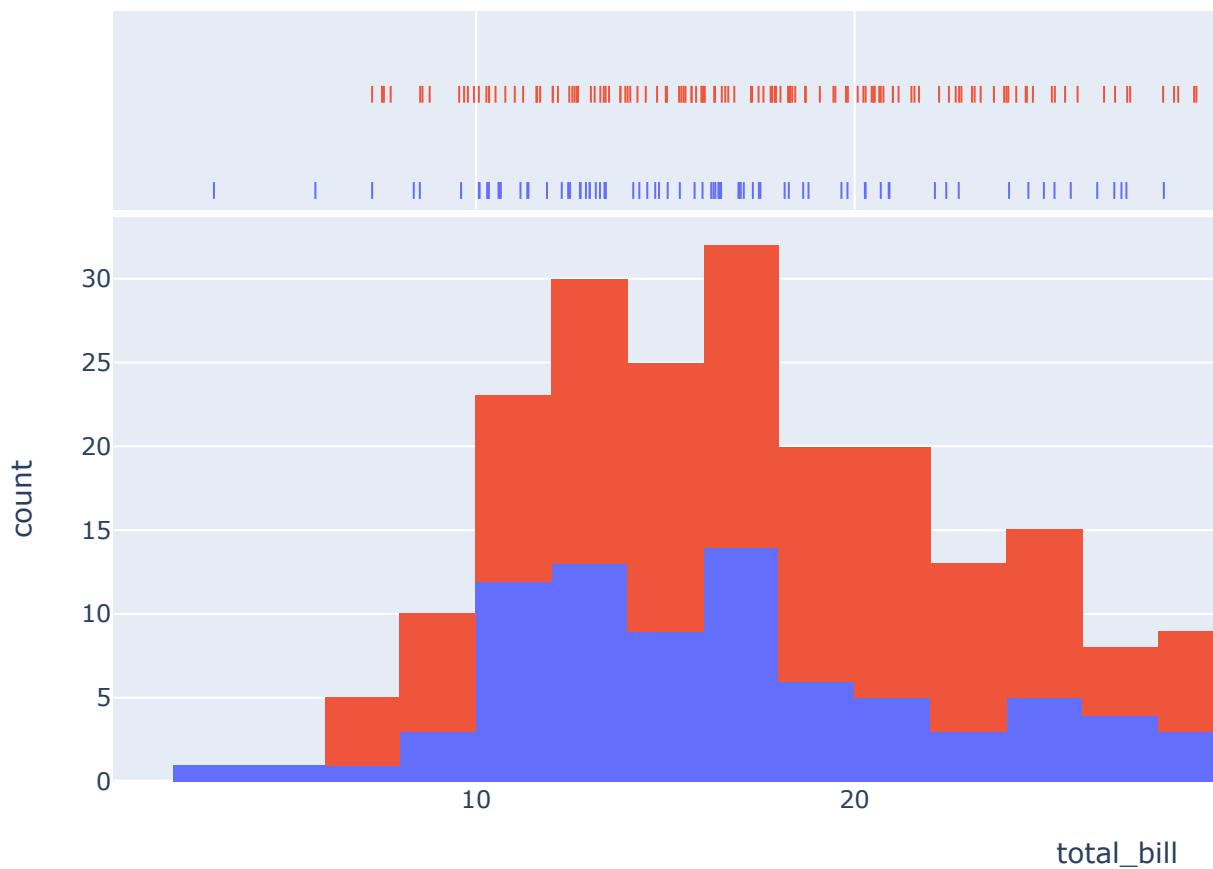


```
fig = px.histogram(tips, x='total_bill', color='sex',
                   color_discrete_sequence=[ 'crimson', 'deepskyblue' ])
fig.show()
```

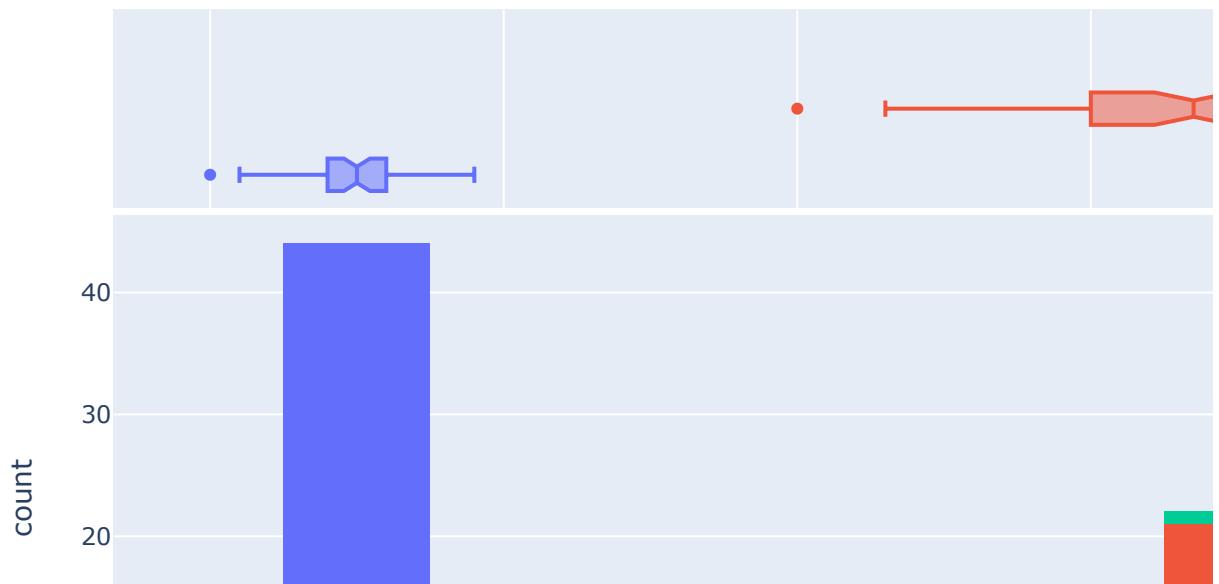


```
fig = px.histogram(tips, x='total_bill', y='tip', histfunc='avg')
fig.show()
```

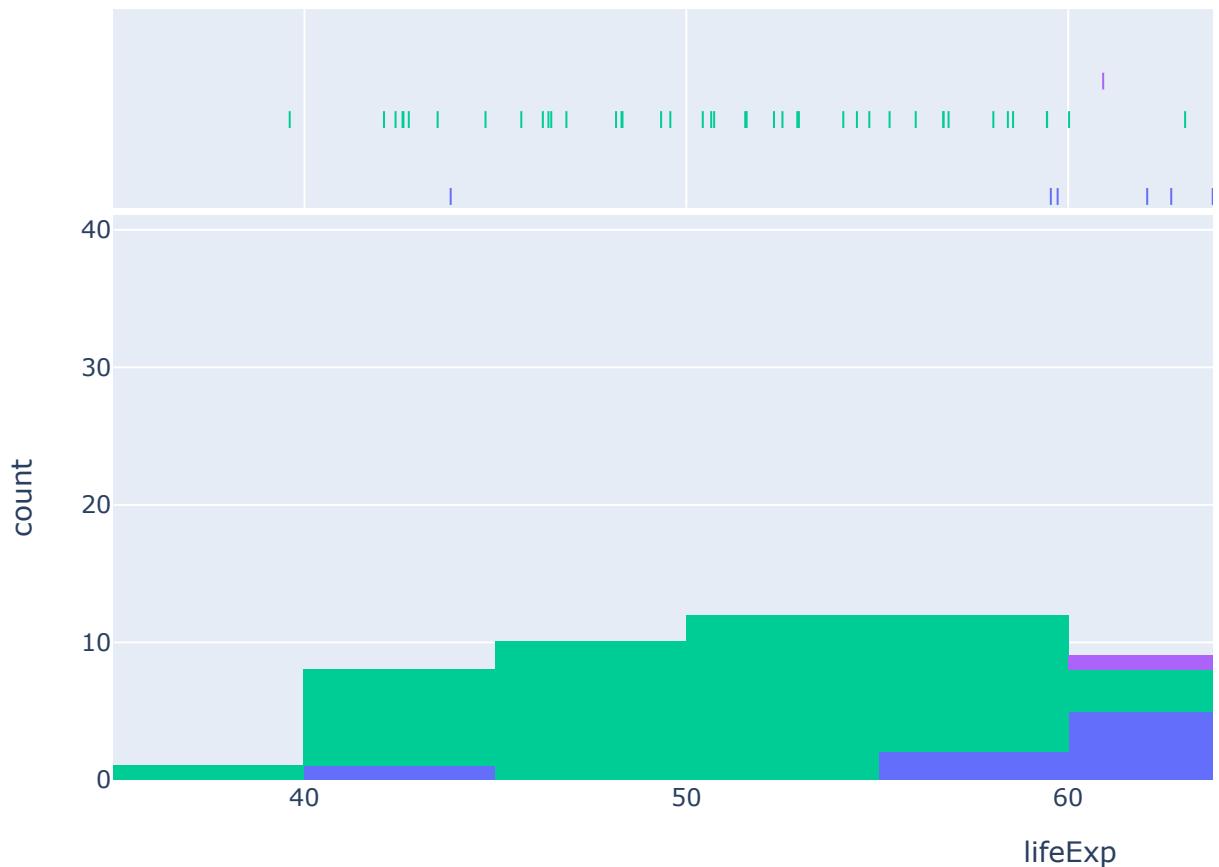
```
fig = px.histogram(tips, x='total_bill',
                    color='sex', marginal='rug',
                    hover_data=tips.columns)
fig.show()
```



```
fig = px.histogram(iris, x='petal_length', color='species', marginal='box')
fig.show()
```



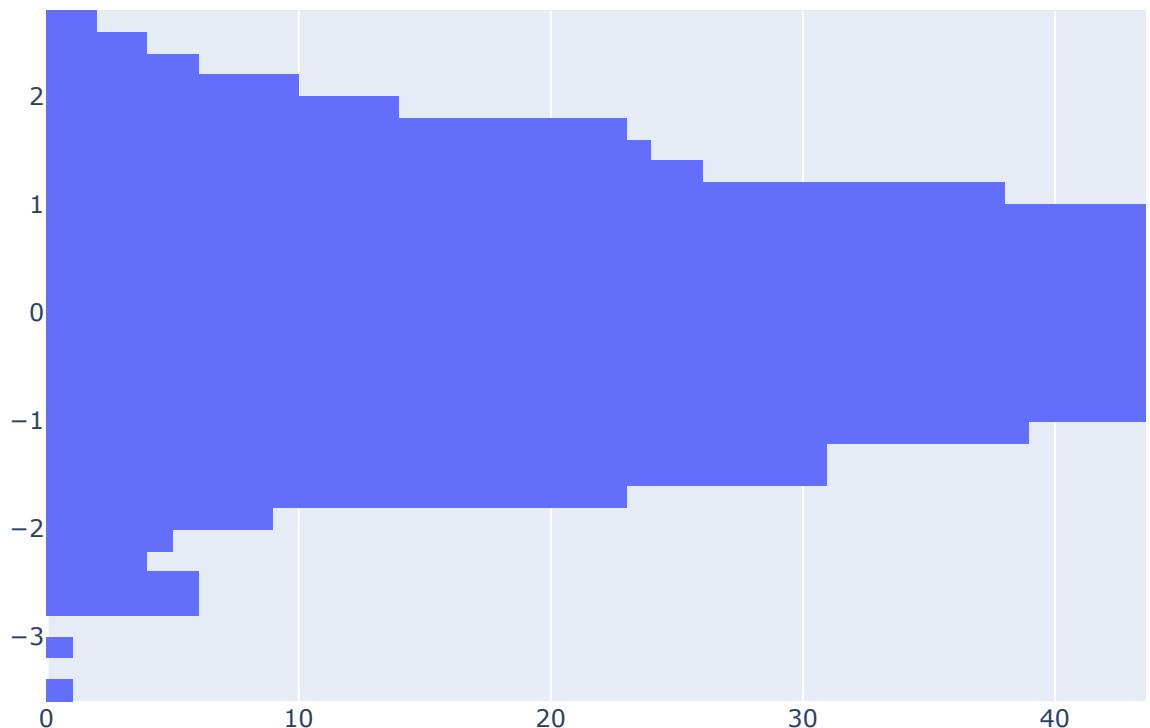
```
fig = px.histogram(gapminder_2007,
                    x='lifeExp', color='continent',
                    marginal='rug', hover_name='country')
fig.show()
```



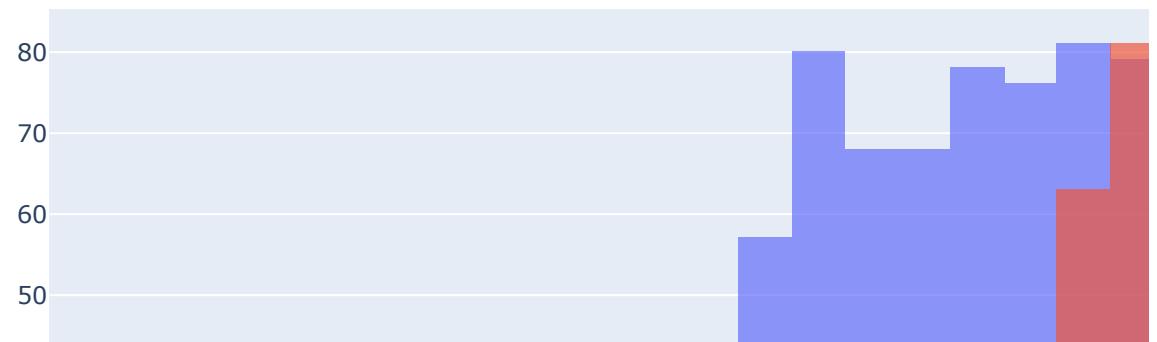
```
n1 = np.random.randn(1000)
n2 = np.random.randn(1000) + 1
```

```
n1 = np.random.ranint(0, 40)
```

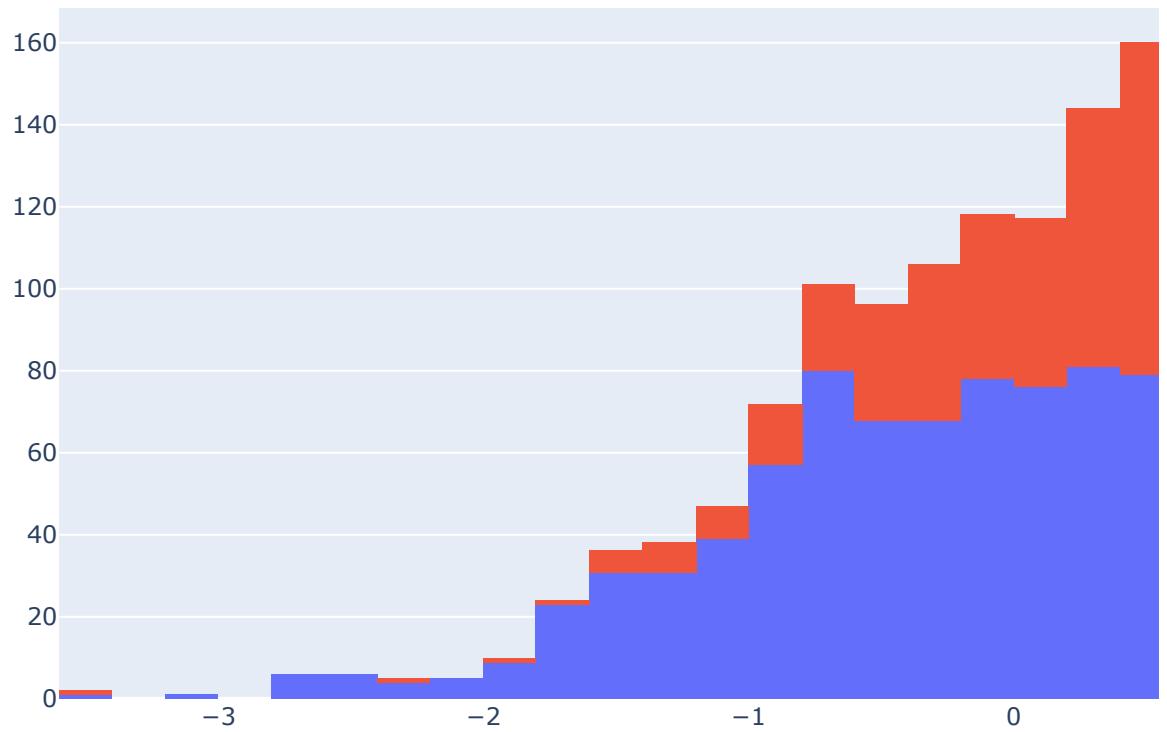
```
fig = go.Figure(data=[go.Histogram(y=n1)])
fig.show()
```



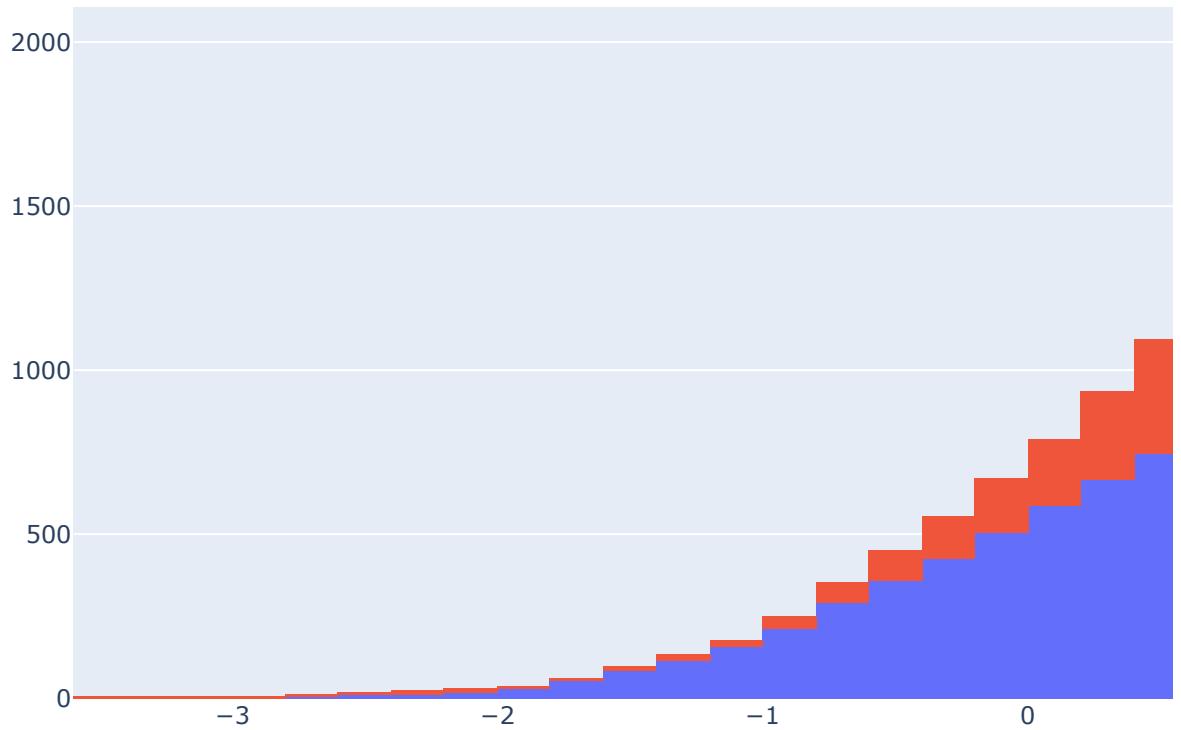
```
fig = go.Figure()
fig.add_trace(go.Histogram(x=n1))
fig.add_trace(go.Histogram(x=n2))
fig.update_layout(barmode='over lay')
fig.update_traces(opacity=0.7)
fig.show()
```



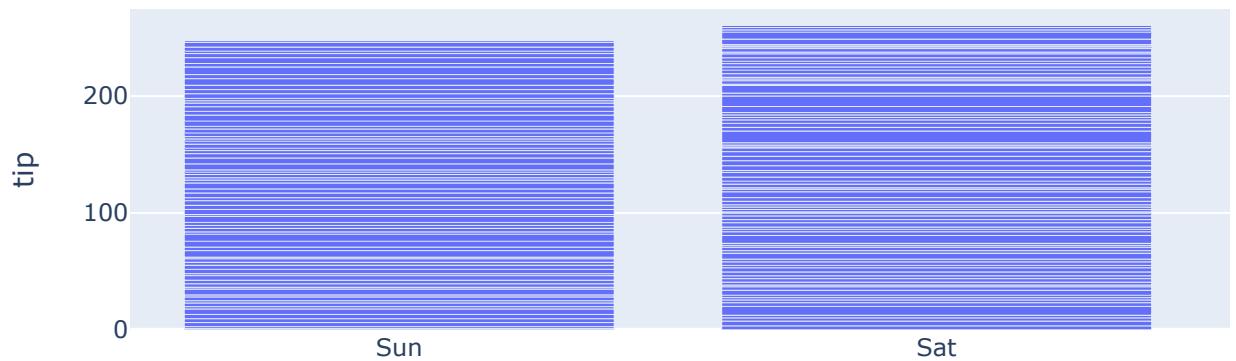
```
fig = go.Figure()
fig.add_trace(go.Histogram(x=n1))
fig.add_trace(go.Histogram(x=n2))
fig.update_layout(barmode='stack')
fig.show()
```



```
fig = go.Figure()
fig.add_trace(go.Histogram(x=n1, cumulative_enabled=True))
fig.add_trace(go.Histogram(x=n2, cumulative_enabled=True))
fig.update_layout(barmode='stack')
fig.show()
```



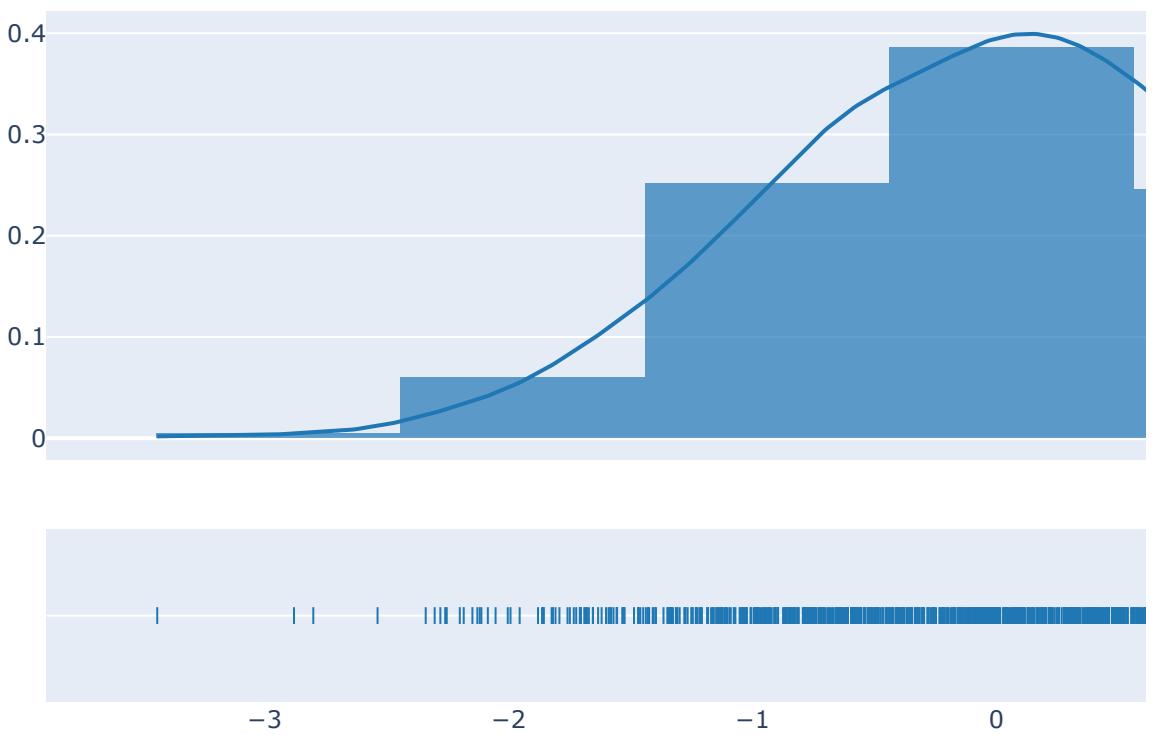
```
fig1 = px.bar(tips, x='day', y='tip', height=300)
fig1.show()
fig2 = px.histogram(tips, x='day', y='tip', histfunc='sum', height=300)
fig2.show()
```



## ▼ 분포도(Distribution Plot)

```
x = np.random.randn(1000)
hist_data = [x]
group_labels = ['distplot']

fig = ff.create_distplot(hist_data, group_labels)
fig.show()
```



```
x1 = np.random.randn(200) - 5
x2 = np.random.randn(200) - 2
```

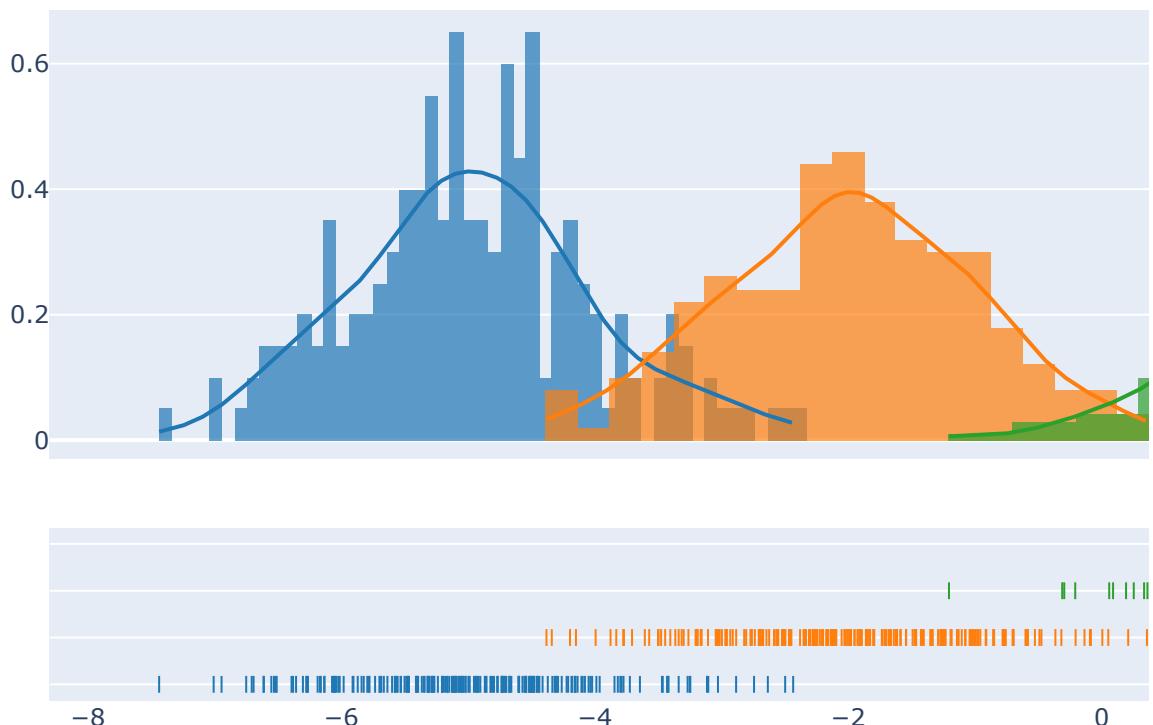
```

x1 = np.random.randn(200)
x2 = np.random.randn(200) + 1
x3 = np.random.randn(200) + 2
x4 = np.random.randn(200) + 5

hist_data = [x1, x2, x3, x4]
group_labels = ['Group A', 'Group B', 'Group C', 'Group D']

fig = ff.create_distplot(hist_data, group_labels,
                         bin_size=[.1, .25, .5, 1])
fig.show()

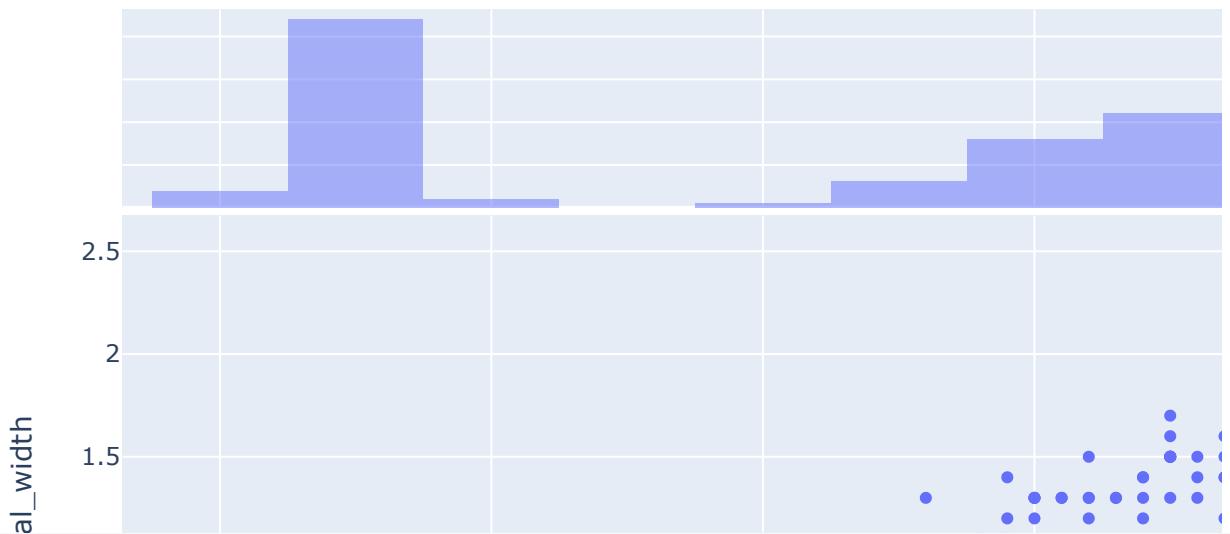
```



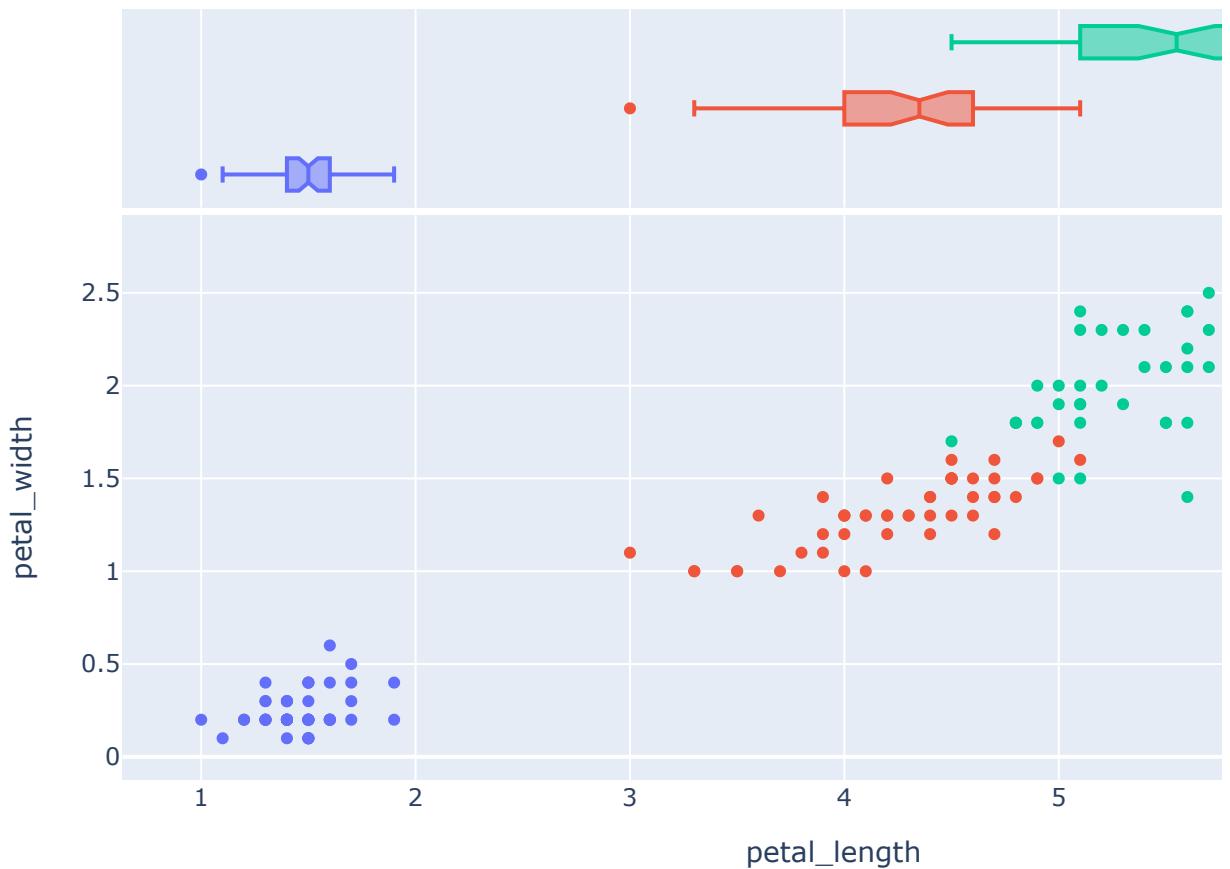
```

fig = px.scatter(iris, x='petal_length', y='petal_width',
                 marginal_x='histogram', marginal_y='rug')
fig.show()

```



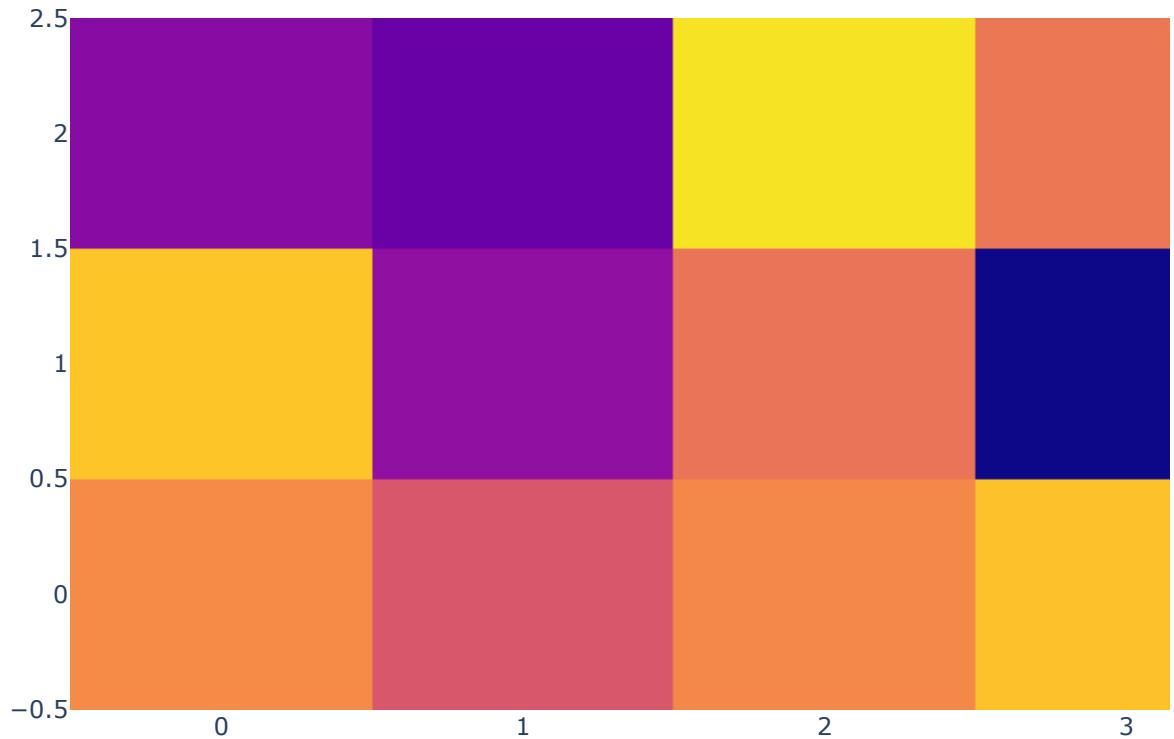
```
fig = px.scatter(iris, x='petal_length', y='petal_width', color='species',
                 marginal_x='box', marginal_y='violin')
fig.show()
```



## ▼ 히트맵(Heatmaps)

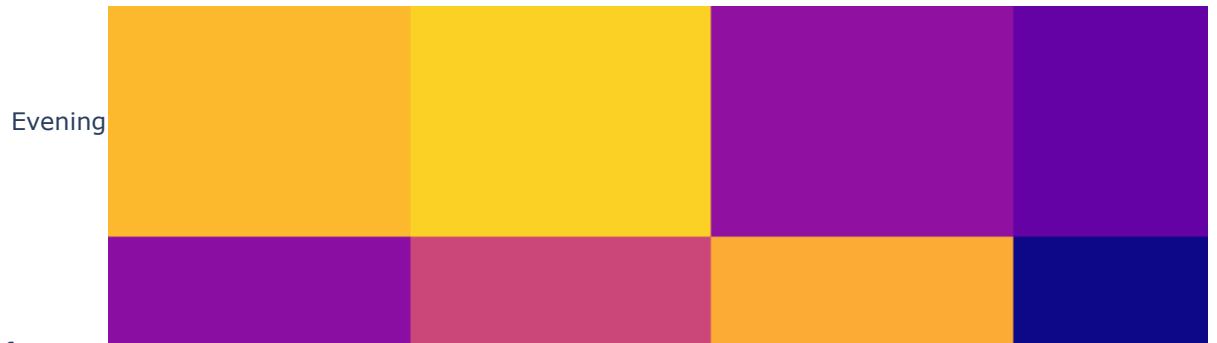
```
n = np.random.randint(1, 100, size=(3, 7))

fig = go.Figure(data=go.Heatmap(z=n))
fig.show()
```

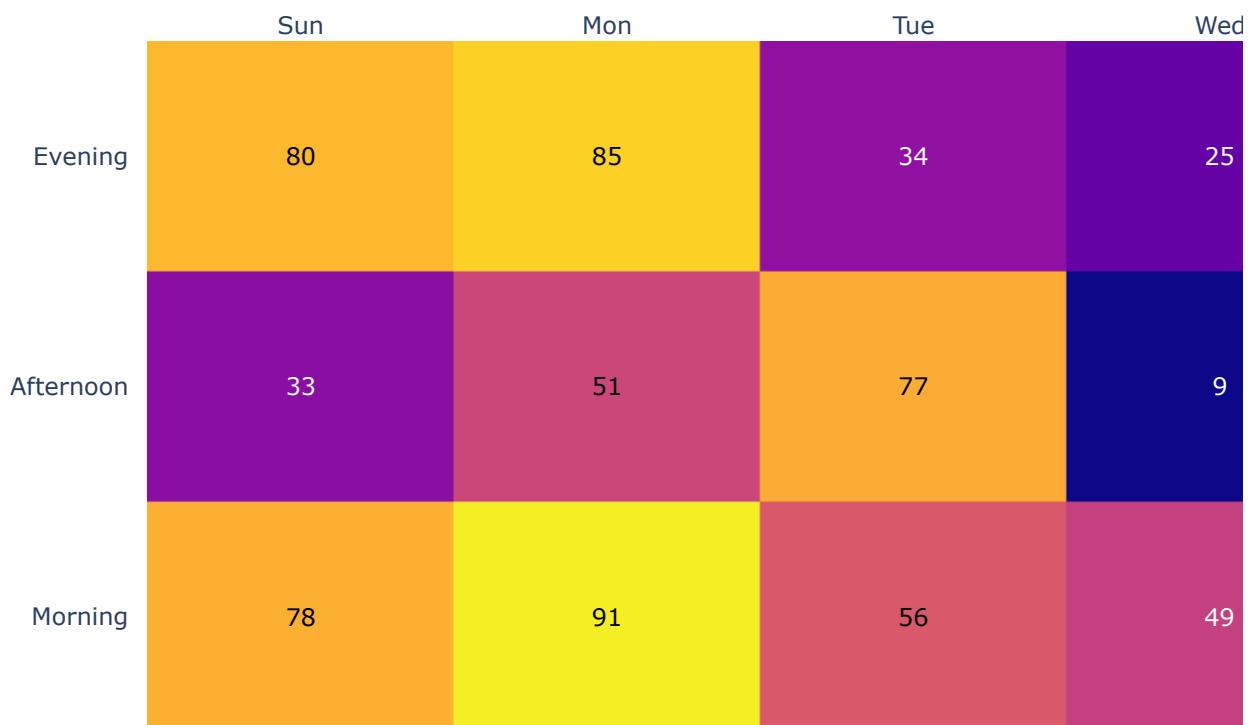


```
w = ['Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat']
t = ['Morning', 'Afternoon', 'Evening']
n = np.random.randint(1, 100, size=(3, 7))

fig = go.Figure(data=go.Heatmap(x=w, y=t, z=n))
fig.show()
```



```
fig = ff.create_annotated_heatmap(x=w, y=t, z=n)
fig.show()
```

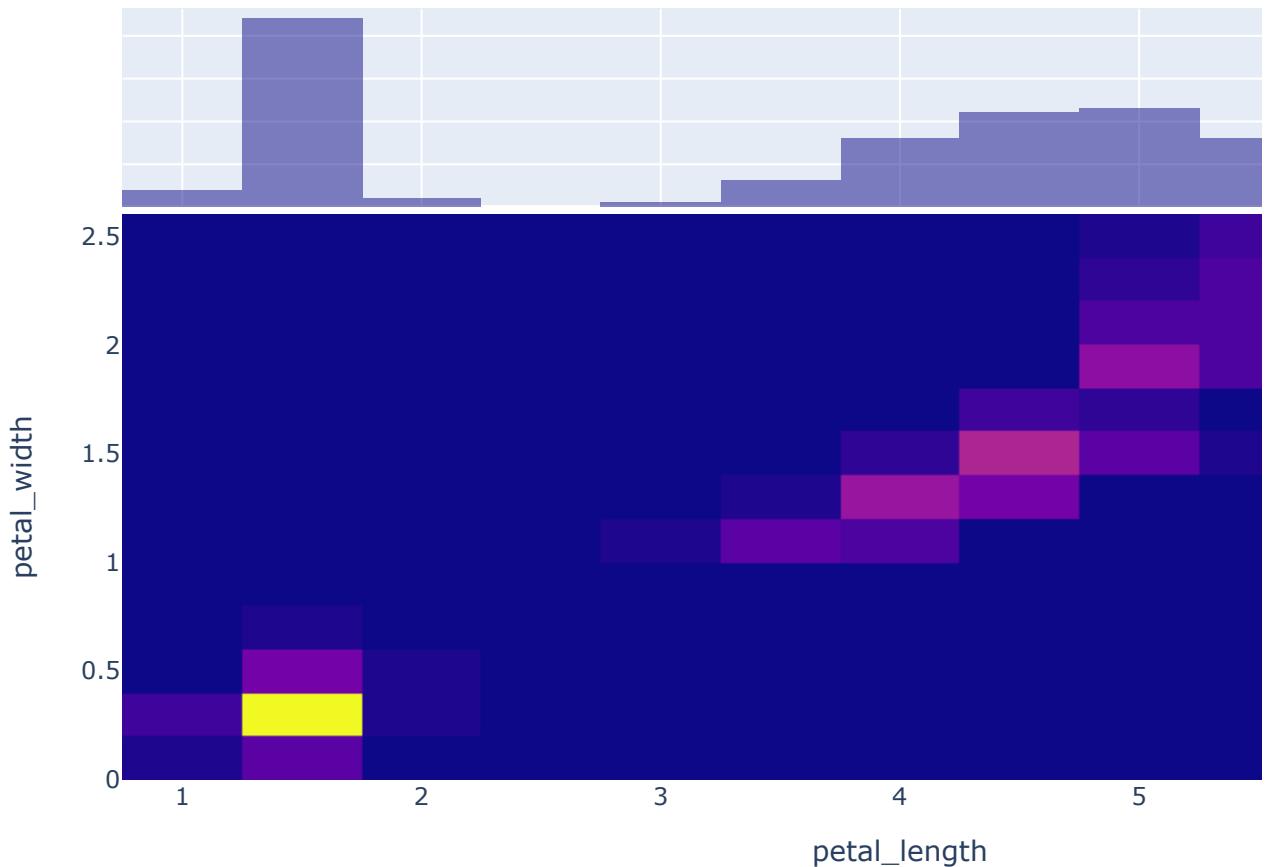


```
fig = px.density_heatmap(iris, x='petal_length', y='petal_width')
fig.show()
```



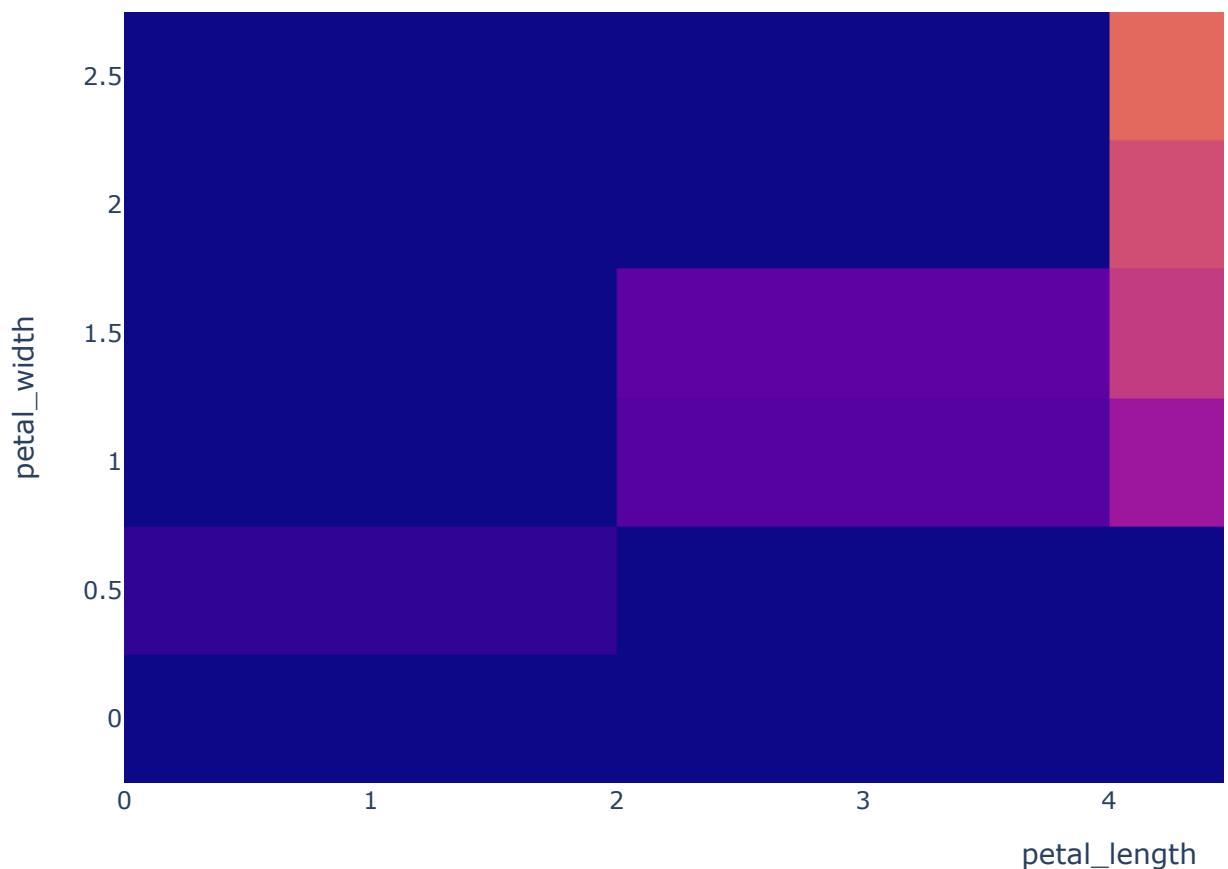
```
fig = px.density_heatmap(iris, x='petal_length', y='petal_width',
                         nbinsx=20, nbinsy=20,
                         color_continuous_scale='viridis')
fig.show()
```

```
fig = px.density_heatmap(iris, x='petal_length', y='petal_width',
                         nbinsx=20, nbinsy=20,
                         marginal_x='histogram', marginal_y='histogram')
fig.show()
```

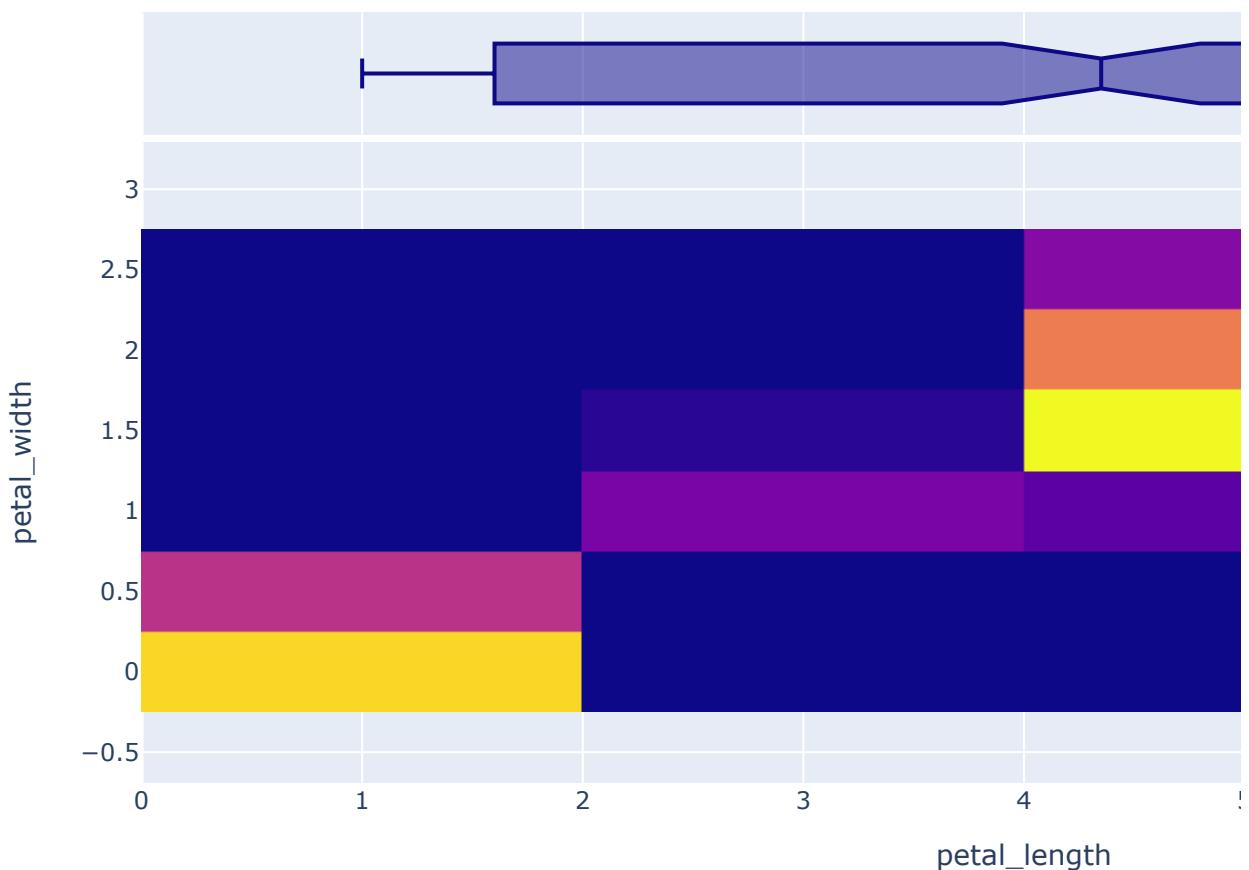




```
fig = px.density_heatmap(iris, x='petal_length', y='petal_width',
                         z='sepal_length', histfunc='avg')
fig.show()
```

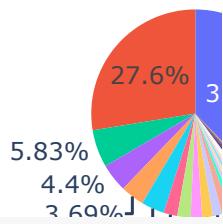


```
fig = px.density_heatmap(iris, x='petal_length', y='petal_width',
                         marginal_x='box', marginal_y='violin')
fig.show()
```

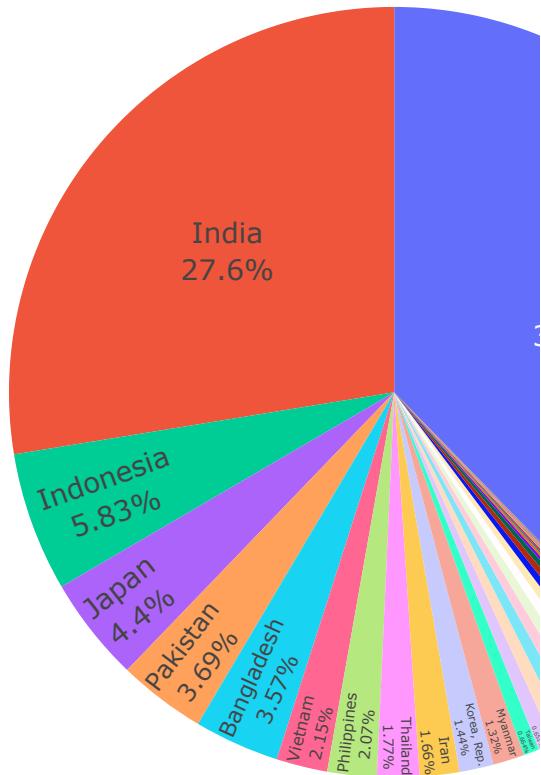


## ▼ 파이 차트(Pie Charts)

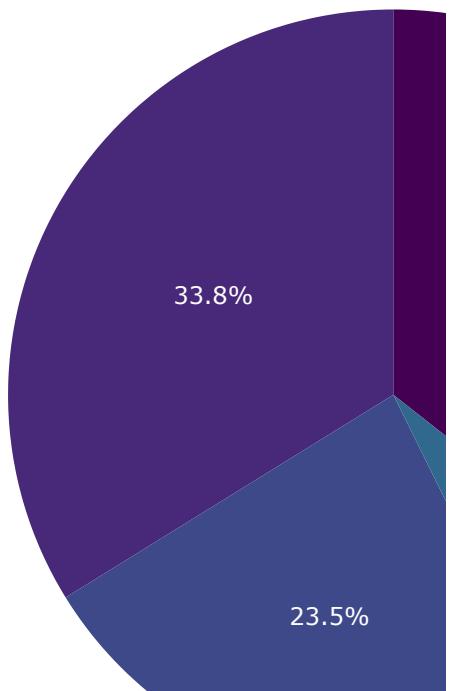
```
fig = px.pie(gapminder_asia, values='pop', names='country')
fig.show()
```



```
fig = px.pie(gapminder_asia, values='pop', names='country',
              hover_data=['lifeExp'], labels={'lifeExp':'life expectancy'})
fig.update_traces(textposition='inside', textinfo='percent+label', textfont_size=14)
fig.show()
```

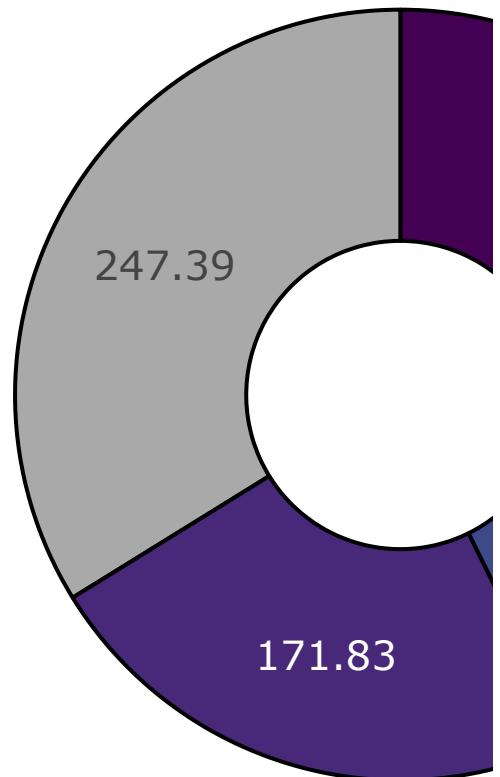


```
fig = px.pie(tips, values='tip', names='day',
              color_discrete_sequence=px.colors.sequential.Viridis)
fig.show()
```



```
fig = px.pie(tips, values='tip', names='day',
              color_discrete_sequence=px.colors.sequential.Viridis,
              hole=.4)
fig.show()
```

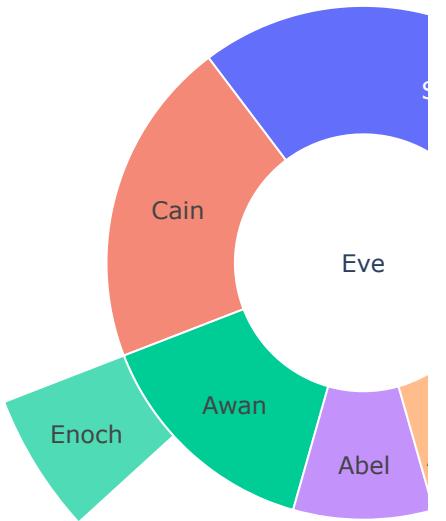
```
fig.update_traces(hoverinfo='label+percent',
                   textinfo='value', textfont_size=20,
                   marker=dict(colors=colors, line=dict(color='black', width=2)))
fig.show()
```



## ▼ Sunburst Chart

```
data = dict(
    character=['Eve', 'Cain', 'Seth', 'Enos', 'Noam', 'Abel', 'Awan', 'Enoch', 'Azura'],
    parent=['', 'Eve', 'Eve', 'Seth', 'Seth', 'Eve', 'Eve', 'Awan', 'Eve'],
    value=[10, 14, 12, 10, 2, 6, 6, 4, 4])

fig = px.sunburst(data, names='character',
                   parents='parent',
                   values='value')
fig.show()
```



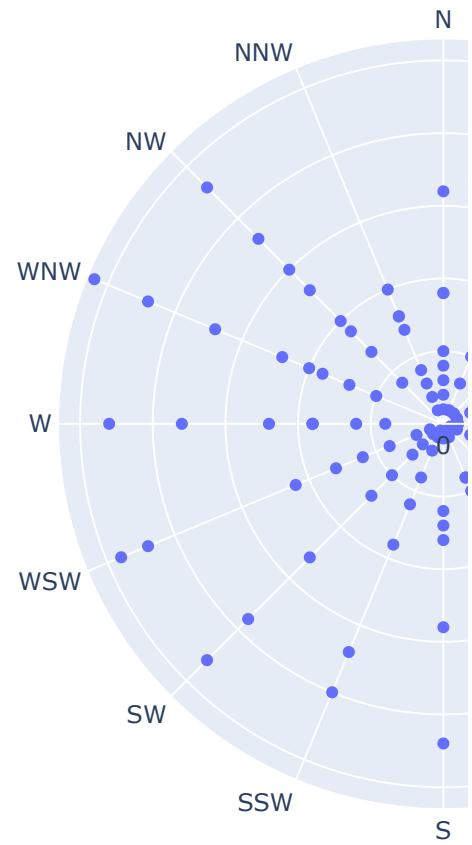
## ▼ 폴라 차트(Polar Charts)

```
wind = px.data.wind()  
wind
```

	direction	strength	frequency
0	N	0-1	0.5
1	NNE	0-1	0.6
2	NE	0-1	0.5
3	ENE	0-1	0.4
4	E	0-1	0.4
...	...	...	...
123	WSW	6+	0.1
124	W	6+	0.9
125	WNW	6+	2.2
126	NW	6+	1.5
127	NNW	6+	0.2

128 rows × 3 columns

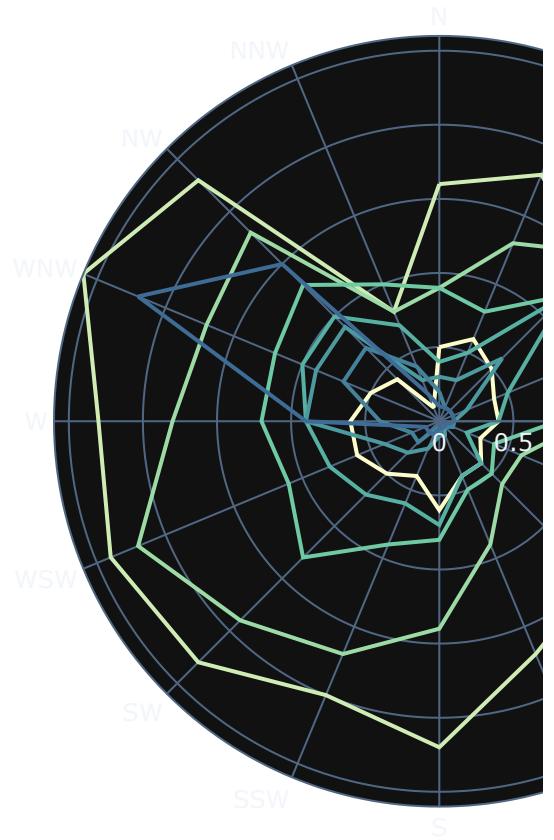
```
fig = px.scatter_polar(wind, r='frequency', theta='direction')  
fig.show()
```



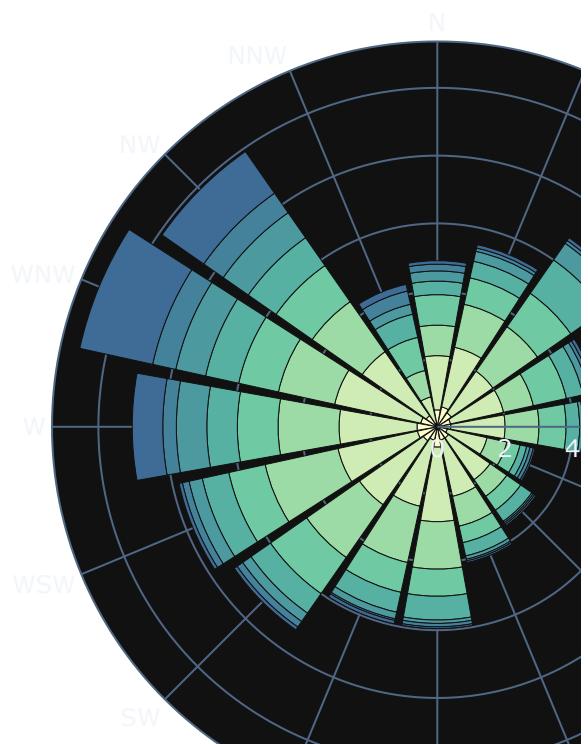
```
fig = px.scatter_polar(wind, r='frequency', theta='direction',
                      color='strength', symbol='strength', size='frequency',
                      color_discrete_sequence=px.colors.sequential.Blues_r)
fig.show()
```



```
fig = px.line_polar(wind, r='frequency', theta='direction',
                     color='strength', line_close=True, template='plotly_dark',
                     color_discrete_sequence=px.colors.sequential.deep)
fig.show()
```



```
fig = px.bar_polar(wind, r='frequency', theta='direction',
                    color='strength', template='plotly_dark',
                    color_discrete_sequence=px.colors.sequential.deep)
fig.show()
```



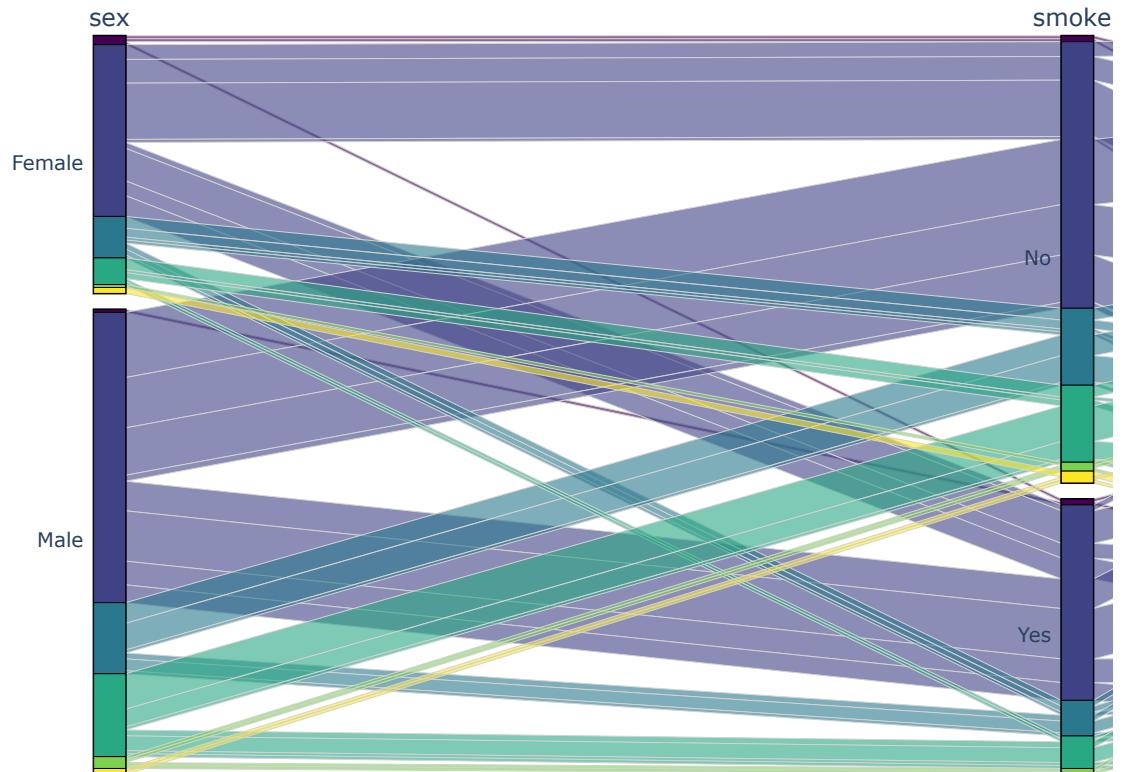
### ▼ 병렬 다이어그램(Parallel Diagram)

```
fig = px.parallel_categories(tips)  
fig.show()
```

```

fig = px.parallel_categories(tips,
                             dimensions=['sex', 'smoker', 'day'],
                             color='size',
                             color_continuous_scale=px.colors.sequential.Viridis)
fig.show()

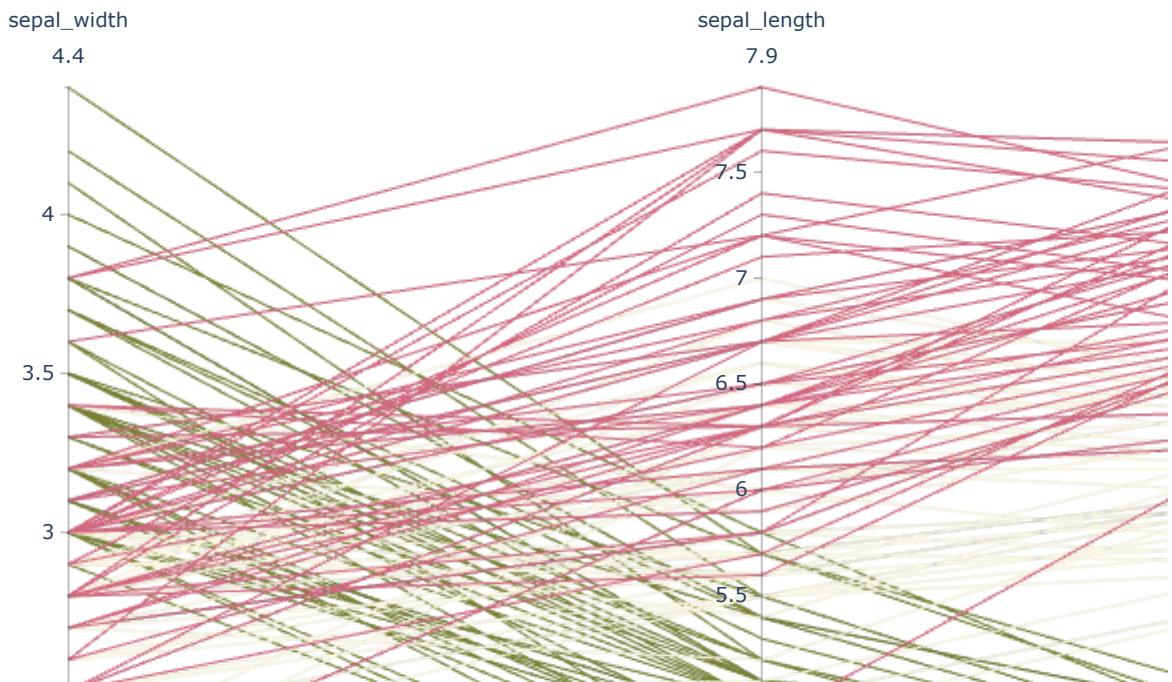
```



```

fig = px.parallel_coordinates(iris, color='species_id',
                             dimensions=['sepal_width', 'sepal_length',
                                         'petal_width', 'petal_length'],
                             color_continuous_scale=px.colors.diverging.Armyrose,
                             color_continuous_midpoint=2)
fig.show()

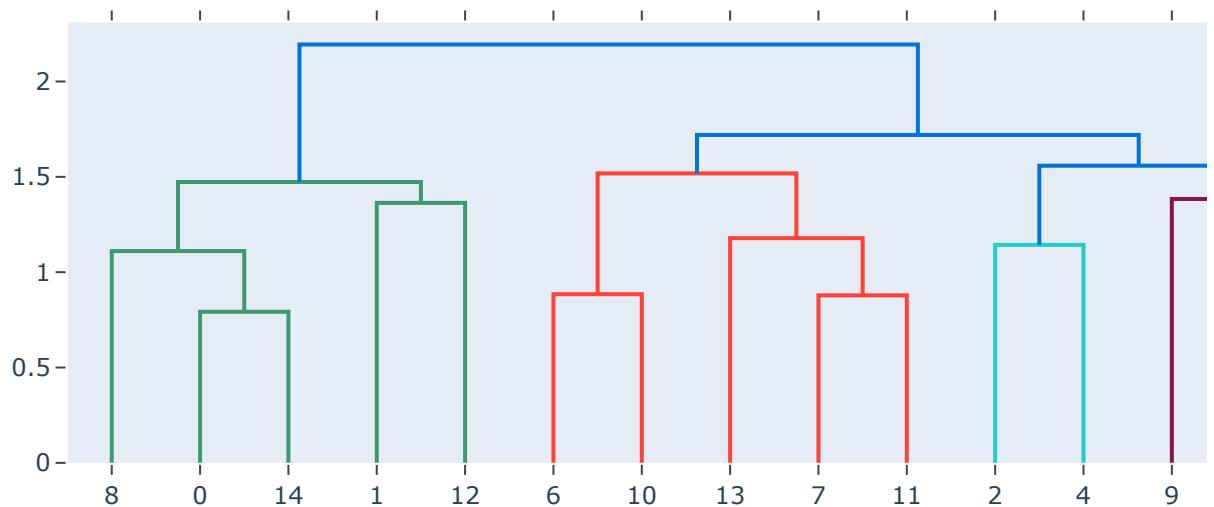
```



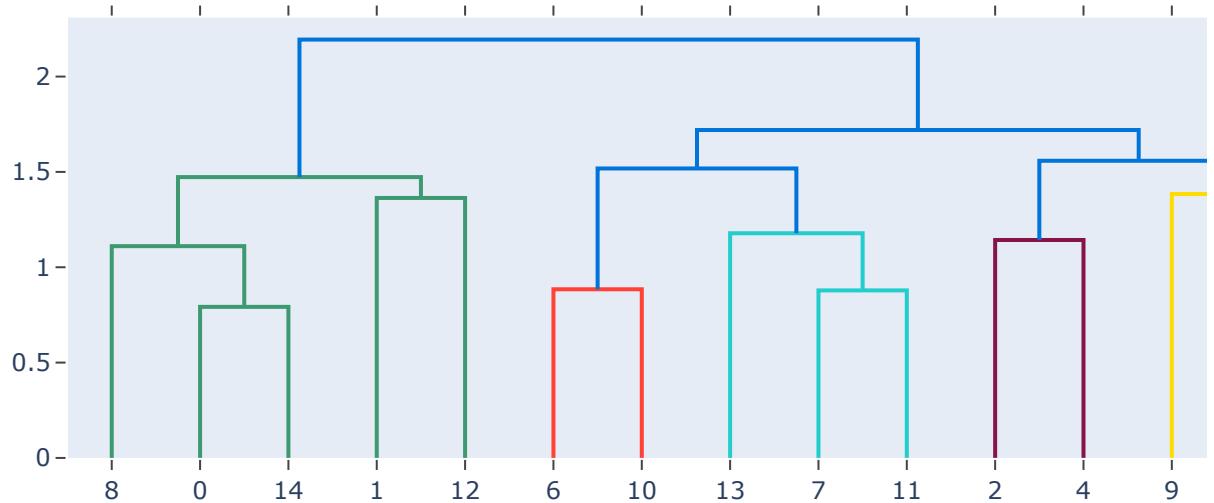
## ▼ 덴드로그램(Dendograms)

```
x = np.random.rand(15, 12)

fig = ff.create_dendrogram(x)
fig.update_layout(width=800, height=400)
fig.show()
```



```
fig = ff.create_dendrogram(x, color_threshold=1.5)
fig.update_layout(width=800, height=400)
fig.show()
```



## maps (Maps)

```
carshare = px.data.carshare()
carshare
```

	centroid_lat	centroid_lon	car_hours	peak_hour
0	45.471549	-73.588684	1772.750000	2
1	45.543865	-73.562456	986.333333	23
2	45.487640	-73.642767	354.750000	20
3	45.522870	-73.595677	560.166667	23
4	45.453971	-73.738946	2836.666667	19
...	...	...	...	...
244	45.547171	-73.556258	951.416667	3
245	45.546482	-73.574939	795.416667	2
246	45.495523	-73.627725	425.750000	8
247	45.521199	-73.581789	1044.833333	17
248	45.532564	-73.567535	694.916667	5

249 rows × 4 columns

```
fig = px.scatter_mapbox(carshare, lat='centroid_lat', lon='centroid_lon',
color='peak_hour', size='car_hours',
```

```
    color_continuous_scale=px.colors.cyclical.Edge,
    size_max=15, zoom=10,
    mapbox_style="carto-positron")
fig.show()
```

- <https://raw.githubusercontent.com/plotly/datasets/master/geojson-counties-fips.json>

```
with urlopen('https://raw.githubusercontent.com/plotly/datasets/master/geojson-counties-fips.json')
    counties = json.load(response)

counties["features"][0]
```

```
{'geometry': {'coordinates': [[[[-86.496774, 32.344437],  
    [-86.717897, 32.402814],  
    [-86.814912, 32.340803],  
    [-86.890581, 32.502974],  
    [-86.917595, 32.664169],  
    [-86.71339, 32.661732],  
    [-86.714219, 32.705694],  
    [-86.413116, 32.707386],  
    [-86.411172, 32.409937],  
    [-86.496774, 32.344437]]],  
    'type': 'Polygon'},  
    'id': '01001',  
    'properties': {'CENSUSAREA': 594.436,  
        'COUNTY': '001',
```

```
'GEO_ID': '0500000US01001',
'LSAD': 'County',
'NAME': 'Autauga',
'STATE': '01'},
'type': 'Feature'}
```

- <https://raw.githubusercontent.com/plotly/datasets/master/fips-unemp-16.csv>

```
unemp = pd.read_csv("https://raw.githubusercontent.com/plotly/datasets/master/fips-unemp-16.csv",
                     dtype={"fips": str})
unemp.head()
```

	fips	unemp
0	01001	5.3
1	01003	5.4
2	01005	8.6
3	01007	6.6
4	01009	5.5

```
fig = px.choropleth_mapbox(unemp, geojson=counties, locations='fips', color='unemp',
                           color_continuous_scale='blues',
                           range_color=(0, 12),
                           mapbox_style='carto-positron',
                           zoom=3, center={'lat': 37, 'lon': -95},
                           opacity=0.5,
                           labels={'unemp': 'unemployment rate'})
fig.update_layout(margin={'r':0, 't':0, 'l':0, 'b':0})
fig.show()
```

- <https://raw.githubusercontent.com/plotly/datasets/master/us-cities-top-1k.csv>

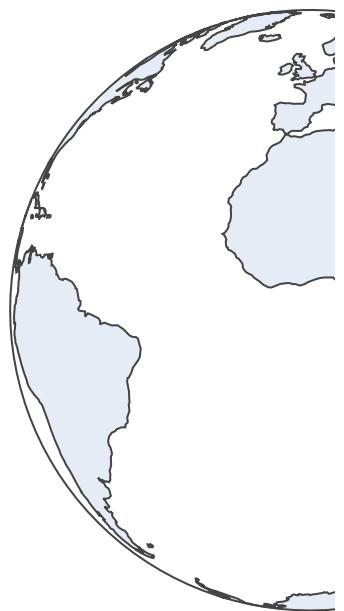
```
us_cities = pd.read_csv("https://raw.githubusercontent.com/plotly/datasets/master/us-cities-top-1k.csv")
us_cities = us_cities.query("State in ['California', 'New York']")
us_cities
```

	City	State	Population	lat	lon
1	Perris	California	72326	33.782519	-117.228648
11	Buena Park	California	82882	33.867514	-117.998118
19	Westminster	California	91739	33.751342	-117.993992
26	Rancho Palos Verdes	California	42448	33.744461	-118.387017
36	Buffalo	New York	258959	42.886447	-78.878369
...	...	...	...	...	...
981	Stanton	California	38623	33.802516	-117.993116
991	Santa Clara	California	120245	37.354108	-121.955236
992	Oceanside	California	172794	33.195870	-117.379483
993	Lake Forest	California	79312	33.646966	-117.689218
999	Compton	California	97877	33.895849	-118.220071

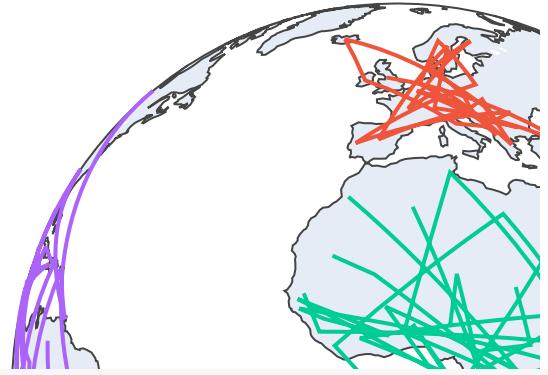
229 rows × 5 columns

```
fig = px.line_mapbox(us_cities, lat='lat', lon='lon',
                     color='State', zoom=2, height=400)
fig.update_layout(mapbox_style='stamen-terrain',
                  mapbox_zoom=2, mapbox_center_lat=40,
                  margin={'r':0, 't':0, 'l':0, 'b':0})
fig.show()
```

```
fig = go.Figure(go.Scattergeo())
fig.update_geos(projection_type='orthographic')
fig.update_layout(height=300, margin={'r':0, 't':0, 'l':0, 'b':0})
fig.show()
```



```
fig = px.line_geo(gapminder_2007, locations='iso_alpha',
                  color='continent', projection='orthographic')
fig.show()
```

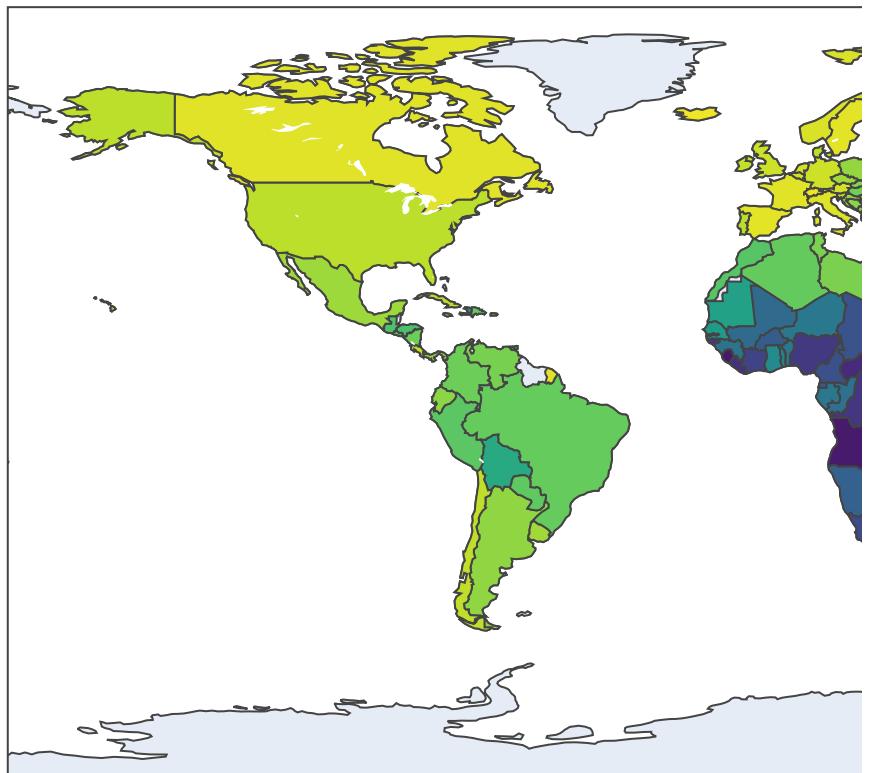


```
fig = go.Figure(go.Scattergeo())
fig.update_geos(projection_type='natural earth')
fig.update_layout(height=300, margin={'r':0, 't':0, 'l':0, 'b':0})
fig.show()
```

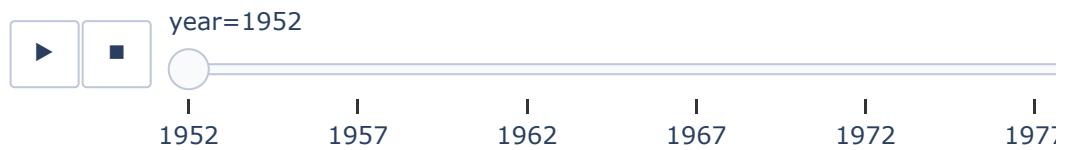
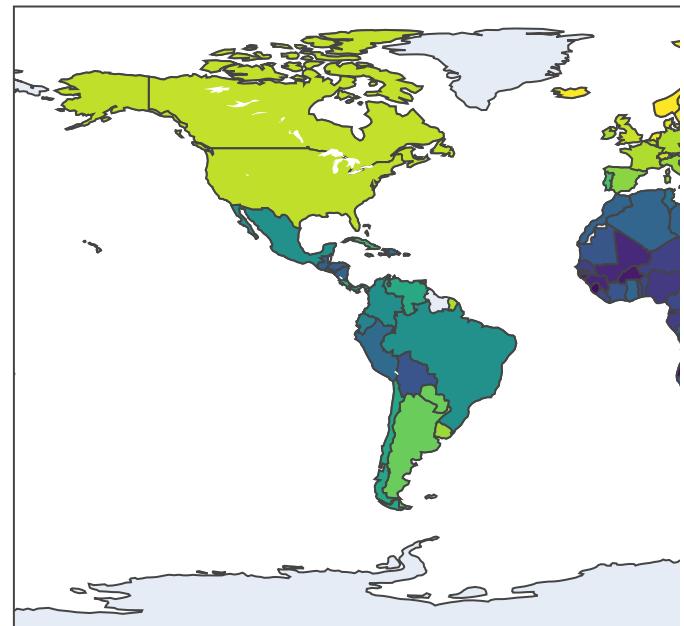




```
fig = px.choropleth(gapminder_2007, locations='iso_alpha',
                     color='lifeExp', hover_name='country',
                     color_continuous_scale=px.colors.sequential.Viridis)
fig.show()
```



```
fig.show()
```



- [https://raw.githubusercontent.com/plotly/datasets/master/2014\\_world\\_gdp\\_with\\_codes.csv](https://raw.githubusercontent.com/plotly/datasets/master/2014_world_gdp_with_codes.csv)

```
gdp = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/2014_world_gdp_with_codes.csv')
```

	COUNTRY	GDP (BILLIONS)	CODE
0	Afghanistan	21.71	AFG
1	Albania	13.40	ALB

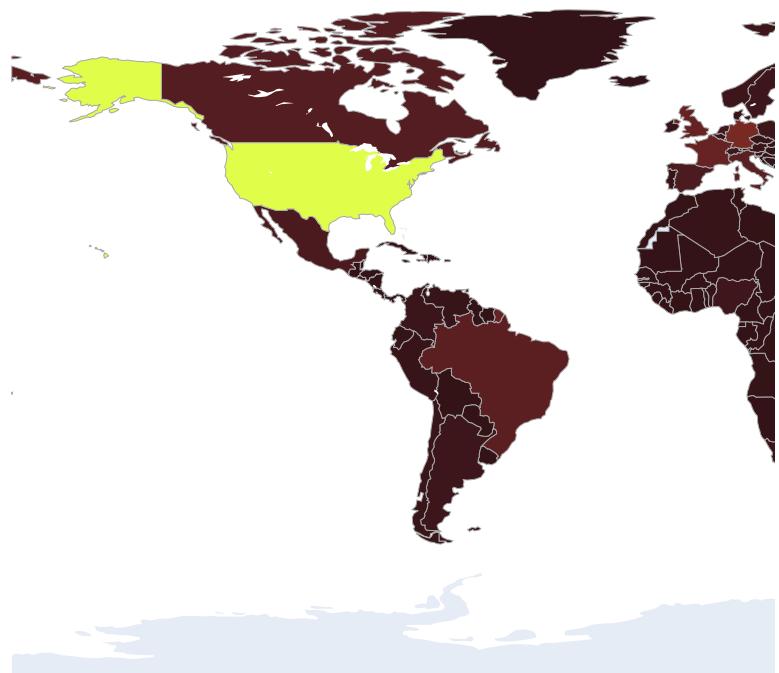
```

fig = go.Figure(data=go.Choropleth(
    locations = gdp['CODE'],
    z = gdp['GDP (BILLIONS)'],
    text = gdp['COUNTRY'],
    colorscale = 'solar_r',
    autocolorscale=False,
    reversescale=True,
    marker_line_color='darkgray',
    marker_line_width=0.5,
    colorbar_tickprefix = '$',
    colorbar_title = 'GDP<br>Billions US $'
))

fig.update_layout(
    title_text='2014 Global GDP',
    geo = dict(showframe=False,
               showcoastlines=False,
               projection_type='equirectangular')
)
fig.show()

```

2014 Global GDP



- <https://raw.githubusercontent.com/plotly/datasets/master/earthquakes-23k.csv>

```
ear thquakes = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/ear thquakes-23k.csv')
ear thquakes
```

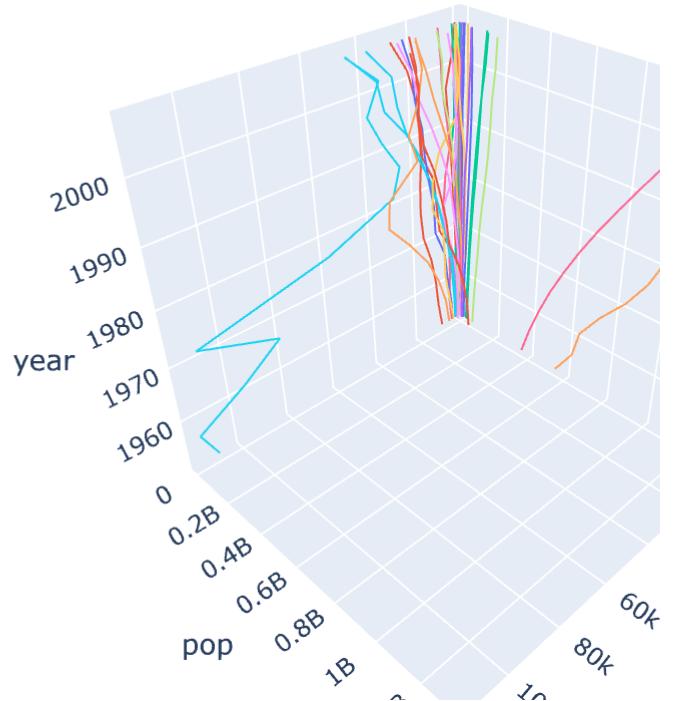
	Date	Latitude	Longitude	Magnitude
<b>0</b>	01/02/1965	19.2460	145.6160	6.0
<b>1</b>	01/04/1965	1.8630	127.3520	5.8
<b>2</b>	01/05/1965	-20.5790	-173.9720	6.2
<b>3</b>	01/08/1965	-59.0760	-23.5570	5.8
<b>4</b>	01/09/1965	11.9380	126.4270	5.8
...	...	...	...	...
<b>23407</b>	12/28/2016	38.3917	-118.8941	5.6
<b>23408</b>	12/28/2016	38.3777	-118.8957	5.5
<b>23409</b>	12/28/2016	36.9179	140.4262	5.9
<b>23410</b>	12/29/2016	-9.0283	118.6639	6.3
<b>23411</b>	12/30/2016	37.3973	141.4103	5.5

23412 rows × 4 columns

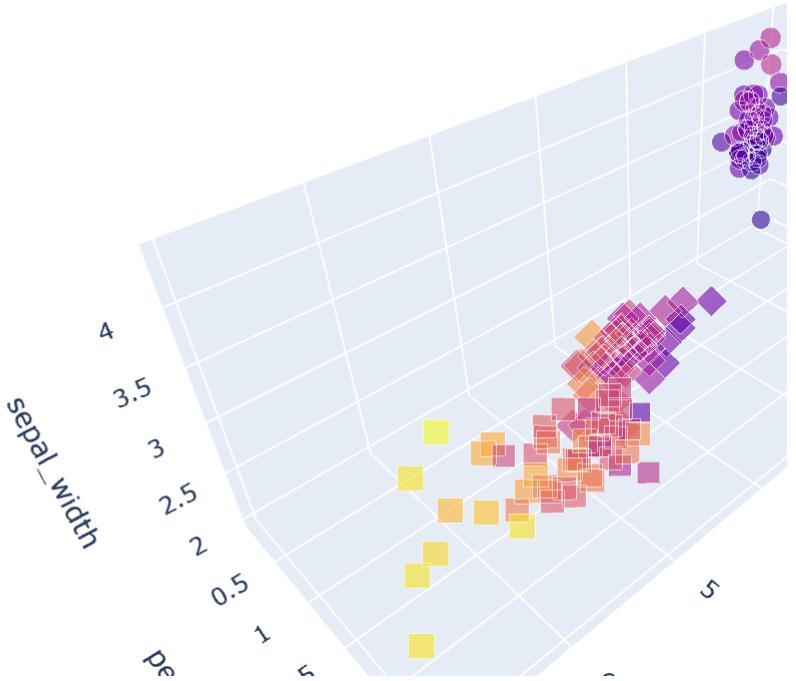
```
fig = px.density_mapbox(ear thquakes, lat='Latitude', lon='Longitude',
                        z='Magnitude', radius=10,
                        center=dict(lat=0, lon=180), zoom=0,
                        mapbox_style='stamen-terrain')
fig.show()
```

## ▼ 3D 차트(3D Charts)

```
fig = px.line_3d(gapminder_asia, x='gdpPerCap', y='pop', z='year', color='country')
fig.show()
```



```
fig = px.scatter_3d(iris, x='petal_length', y='petal_width', z='sepal_width',
                     color='sepal_length', symbol='species',
                     size='sepal_length', opacity=0.6)
fig.show()
```



```
fig = px.scatter_3d(gapminder, x='year', y='continent', z='pop',
                     size='gdpPerCap', color='lifeExp',
                     hover_data=['country'])
fig.update_layout(scene_zaxis_type='log')
fig.show()
```

## 기타 차트

- 간트 차트(Gantt Charts): <https://plotly.com/python/gantt/>
- 테이블(Tables): <https://plotly.com/python/table/>
- 생키 다이어그램(Sankey Diagram): <https://plotly.com/python/sankey-diagram/>
- 트리맵(Treemap): <https://plotly.com/python/treemaps/>
- 트리플롯(Tree-plots): <https://plotly.com/python/tree-plots/>
- 3차 플롯(Ternary Plots): <https://plotly.com/python/ternary-plots/>
- 3차 오버레이(Ternary Overlay): <https://plotly.com/python/ternary-scatter-contour/>
- 3차 등고선(Ternary Contours): <https://plotly.com/python/ternary-contour/>
- 이미지 쇼(Image Show): <https://plotly.com/python/imshow/>
- Quiver Plots: <https://plotly.com/python/quiver-plots/>
- 스트림라인 플롯(Streamline Plots): <https://plotly.com/python/streamline-plots/>
- 카펫 플롯(Carpet Plots): <https://plotly.com/python/carpet-plot/>
- 카펫 등고선(Carpet Contour Plot): <https://plotly.com/python/carpet-contour/>
- 카펫 산점도(Carpet Scatter Plot): <https://plotly.com/python/carpet-scatter/>
- 네트워크 그래프(Network Graphs): <https://plotly.com/python/network-graphs/>
- 깔대기 차트(Funnel Chart): <https://plotly.com/python/funnel-charts/>
- 등고선 플롯(Contour Plot): <https://plotly.com/python/contour-plots/>
- 2D 히스토그램 등고선(2D Histogram Contour): <https://plotly.com/python/2d-histogram-contour/>
- Trisurf Plots: <https://plotly.com/python/trisurf/>
- 3D Mesh Plots: <https://plotly.com/python/3d-mesh/>
- 3D Isosurface Plots: <https://plotly.com/python/3d-isosurface-plots/>
- 3D Volume Plots: <https://plotly.com/python/3d-volume-plots/>
- 3D Cone Plots: <https://plotly.com/python/cone-plot/>
- 3D Streamtube Plots: <https://plotly.com/python/streamtube-plot/>
- 3D Camera Controls: <https://plotly.com/python/3d-camera-controls/>

## ▼ Plotly 스타일

## ▼ 데이터 레이블(Data Label)

- [https://raw.githubusercontent.com/plotly/datasets/master/2014\\_usa\\_states.csv](https://raw.githubusercontent.com/plotly/datasets/master/2014_usa_states.csv)

```
data = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/2014_usa_states.csv')
```

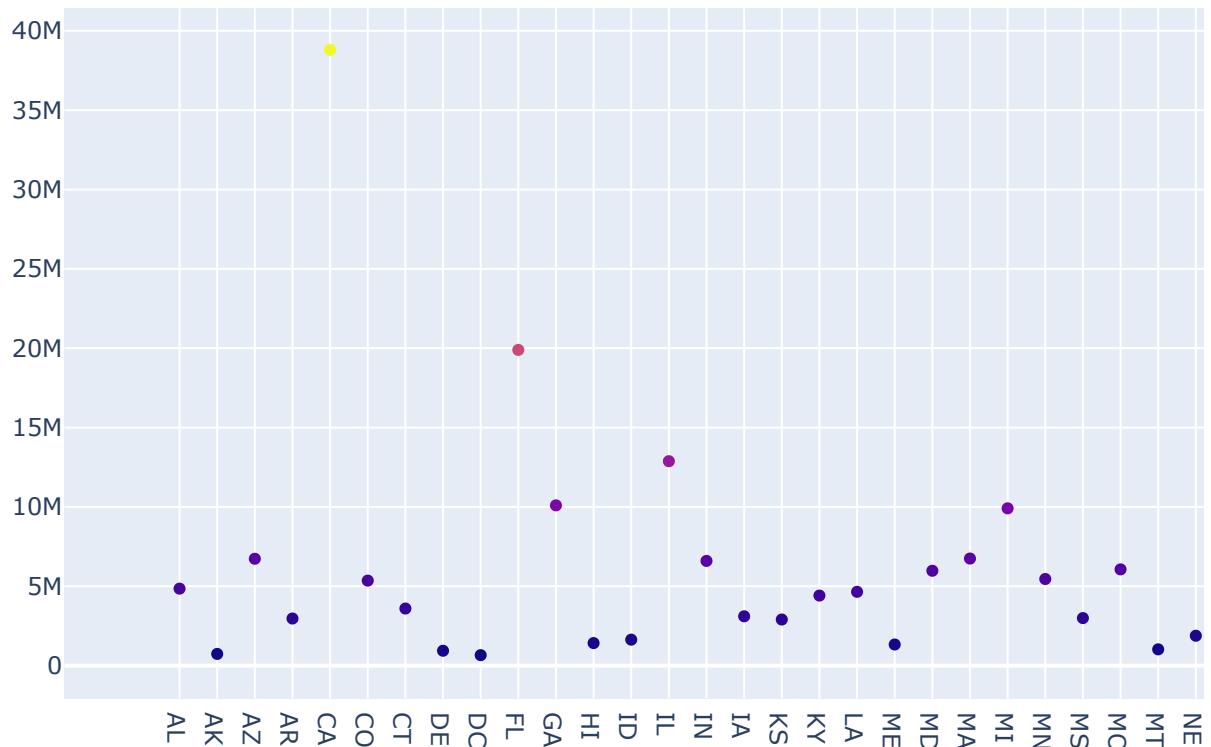
```
fia = go.Figure(data=go.Scatter(x=data['Postal'])
```

```

    fig = go.Figure(data=go.Scatter(x=x_data['State'],
                                      y=data['Population'],
                                      mode='markers',
                                      marker_color=data['Population'],
                                      text=data['State']))
fig.update_layout(title='Population of USA States')
fig.show()

```

## Population of USA States

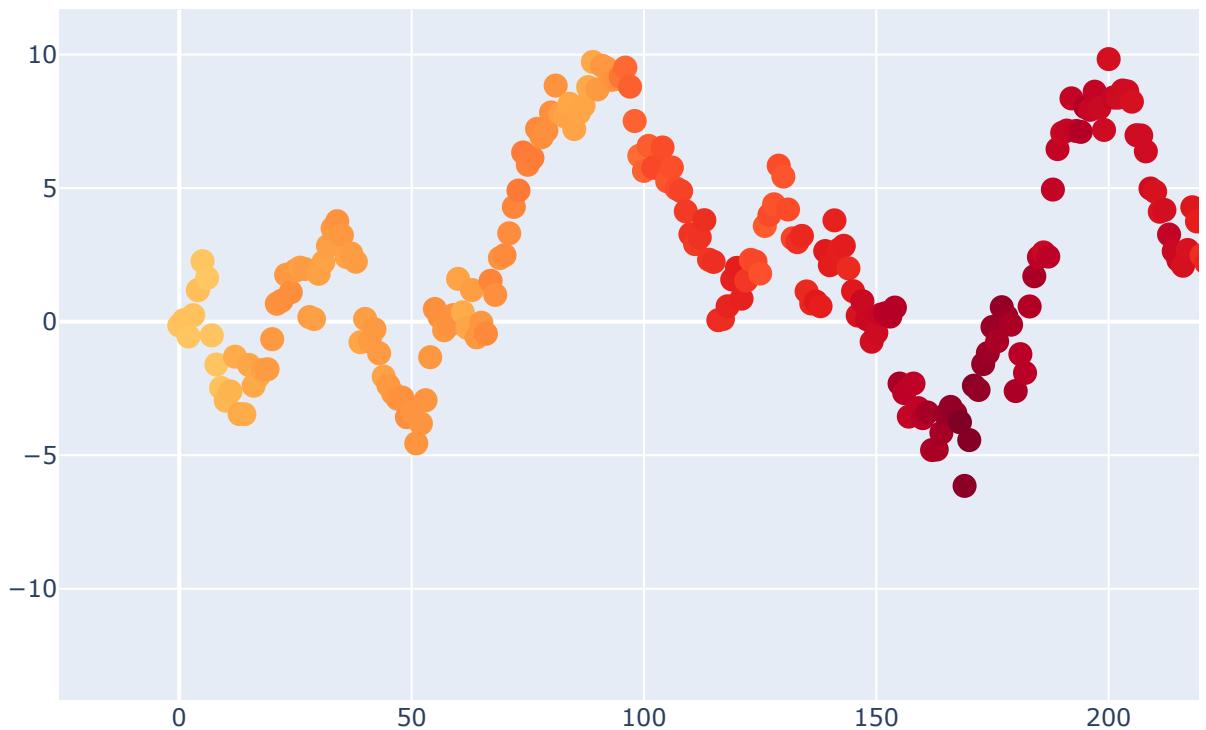


## ▼ 색상 차원(Color Dimension)

```

fig = go.Figure(data=go.Scatter(
    y = np.random.randn(400).cumsum(),
    mode='markers',
    marker=dict(
        size=12,
        color=np.random.randn(400).cumsum(),
        colorscale='YlOrRd',
        showscale=True
    )
))
fig.show()

```



## ▼ 템플릿(Templates)

```
pio.templates
```

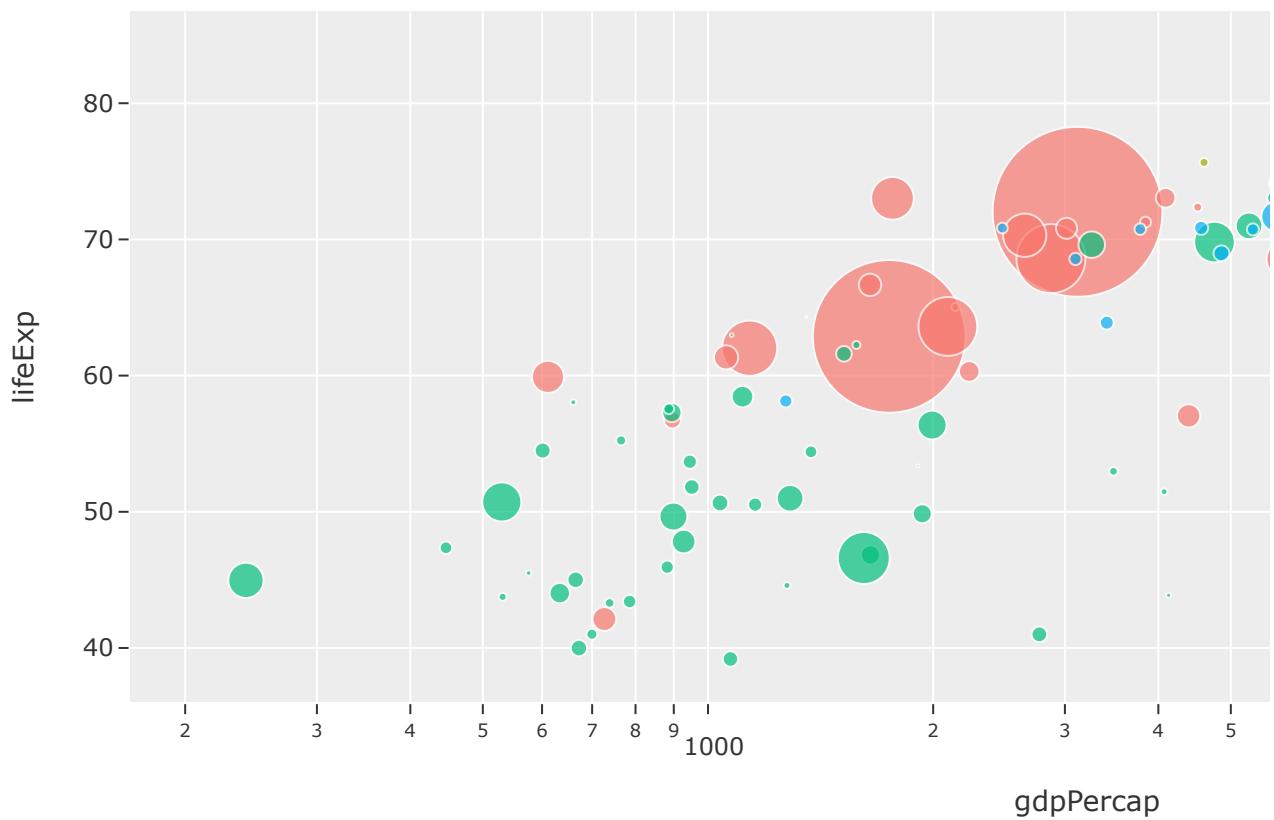
```
Templates configuration
```

```
Default template: 'plotly'  
Available templates:  
['ggplot2', 'seaborn', 'simple_white', 'plotly',  
 'plotly_white', 'plotly_dark', 'presentation', 'xgridoff',  
 'ygridoff', 'gridon', 'none']
```

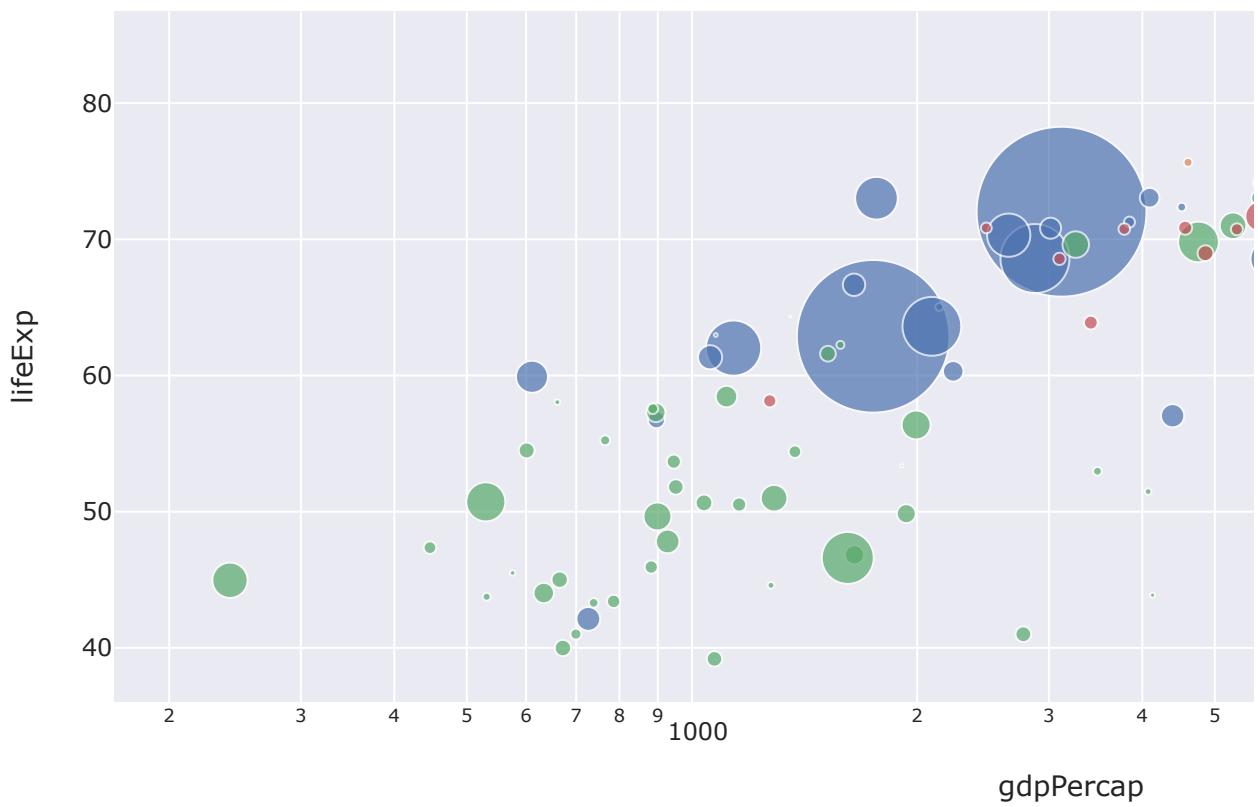
```
gapminder_2002 = gapminder.query("year == 2002")

for template in ['ggplot2', 'seaborn', 'simple_white', 'plotly', 'plotly_white', 'plotly_dark', 'pr
    fig = px.scatter(gapminder_2002,
                      x='gdpPerCap', y='lifeExp', size='pop', color='continent',
                      log_x=True, size_max=60,
                      template=template, title="Template: '%s'" % template)
fig.show()
```

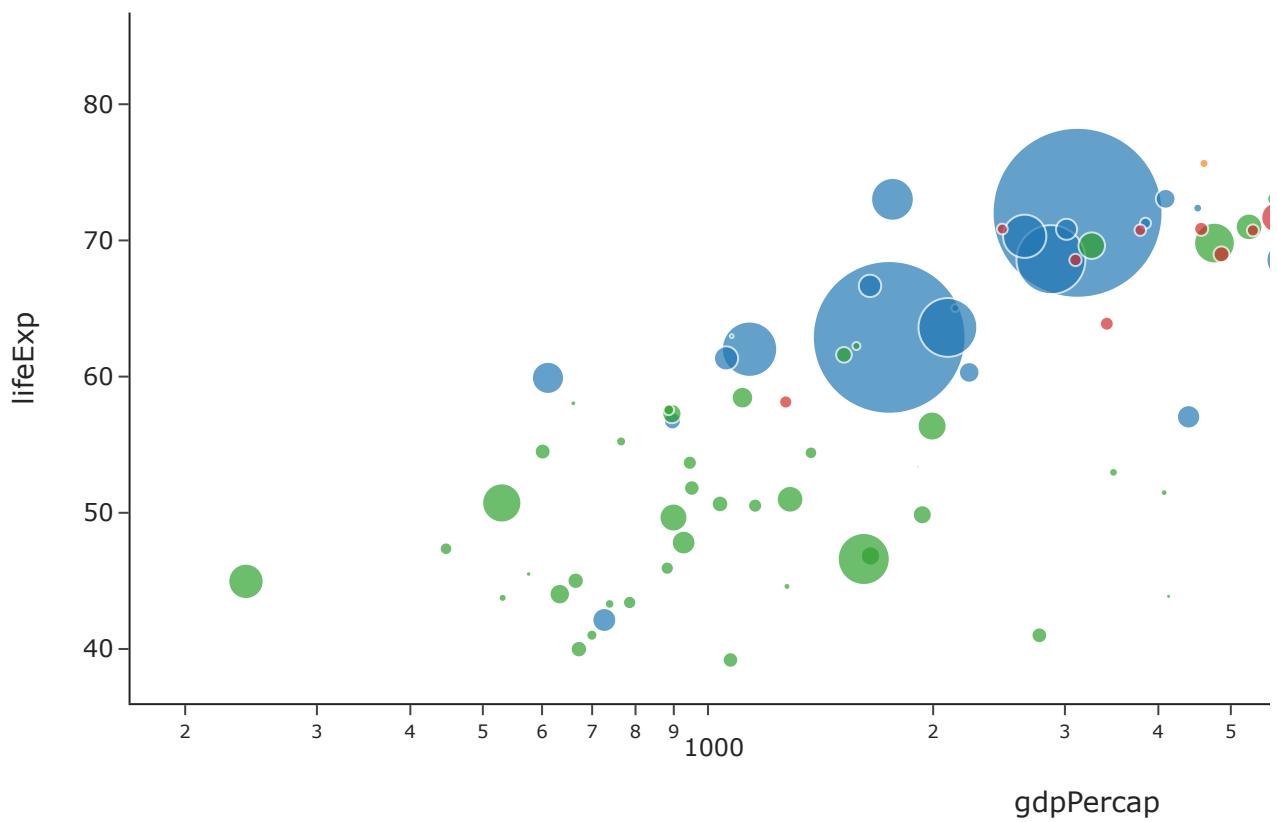
Template: 'gg



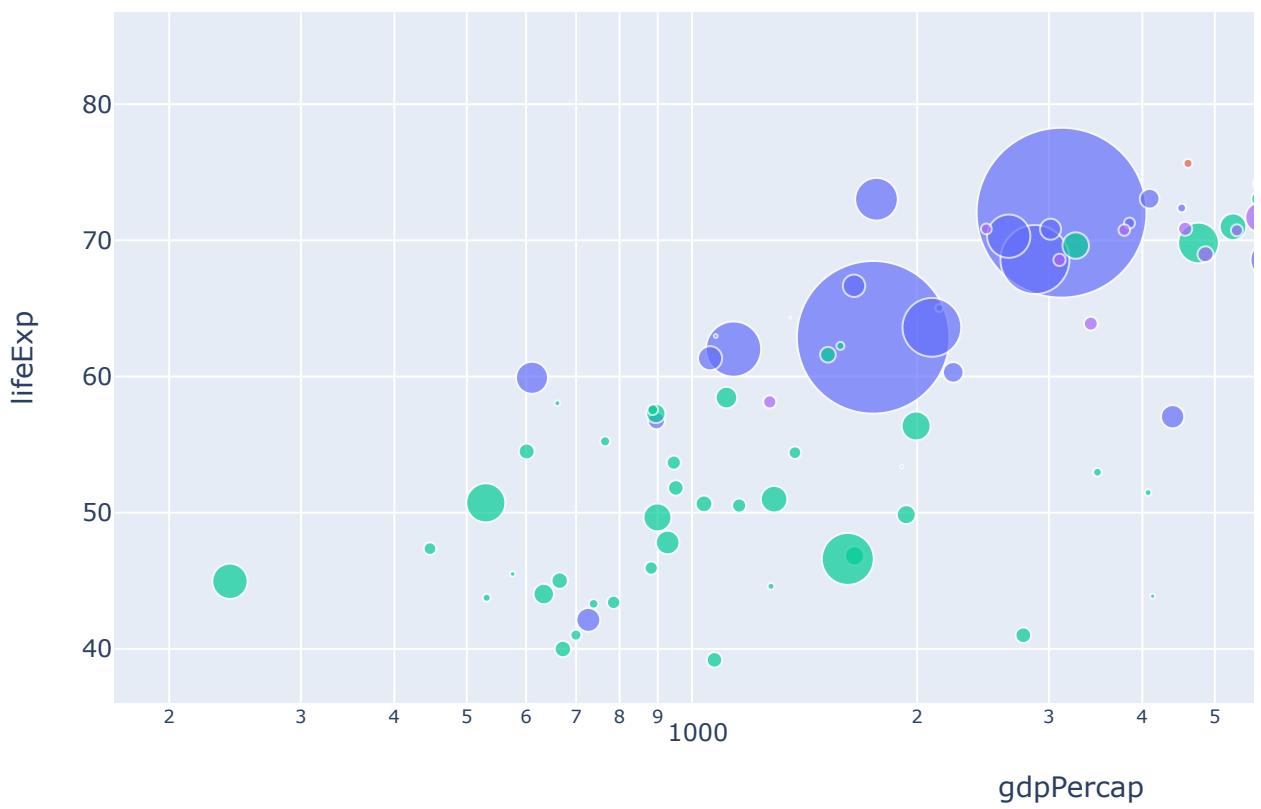
Template: 'se



Template: 'simple\_white'



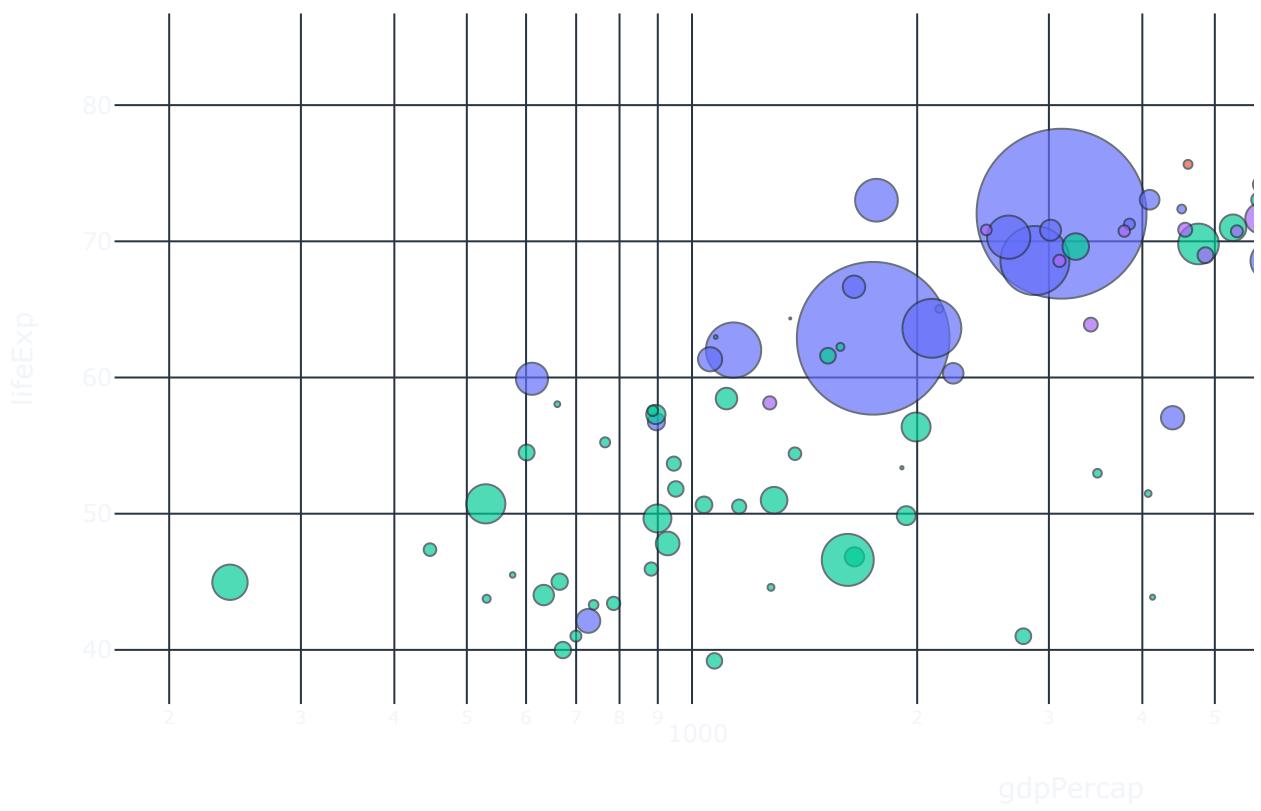
Template: 'plotly'



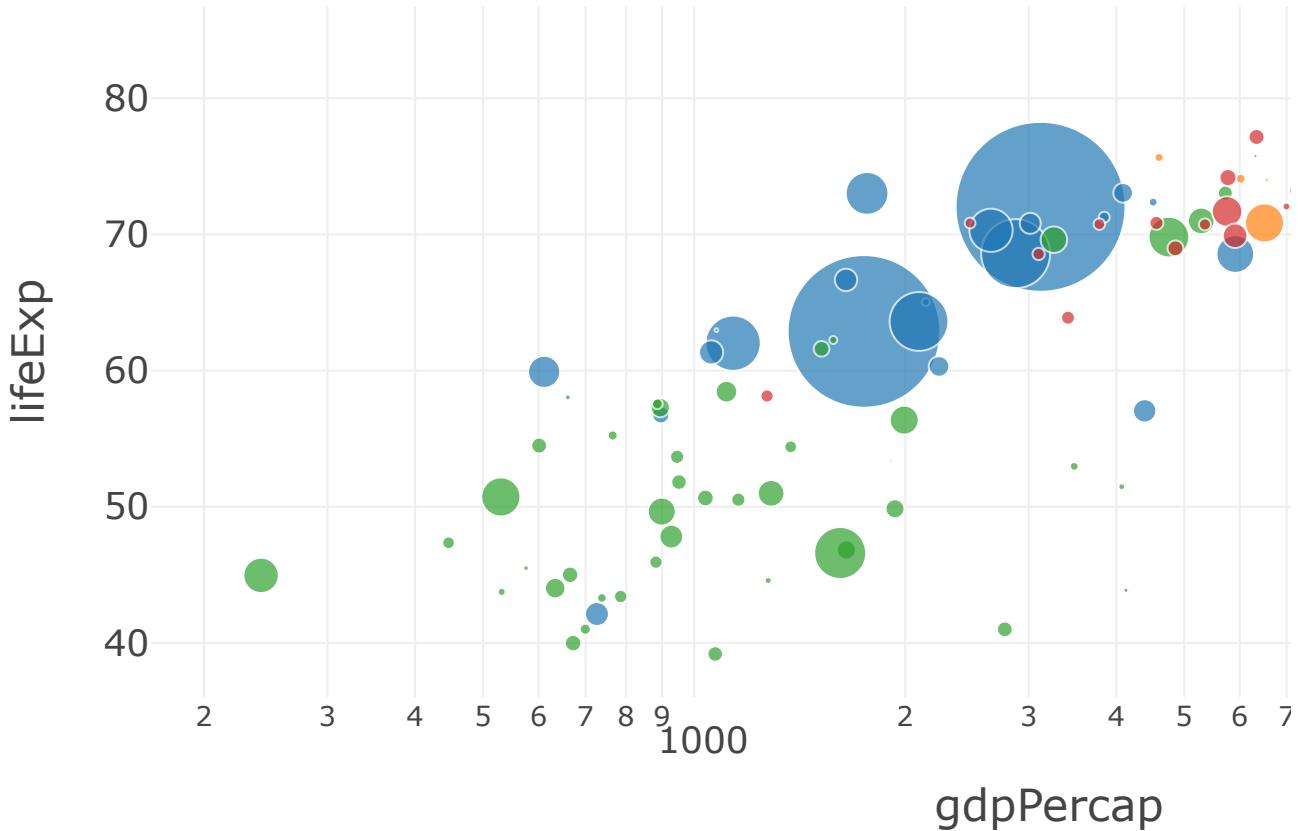
Template: 'plotly\_white'



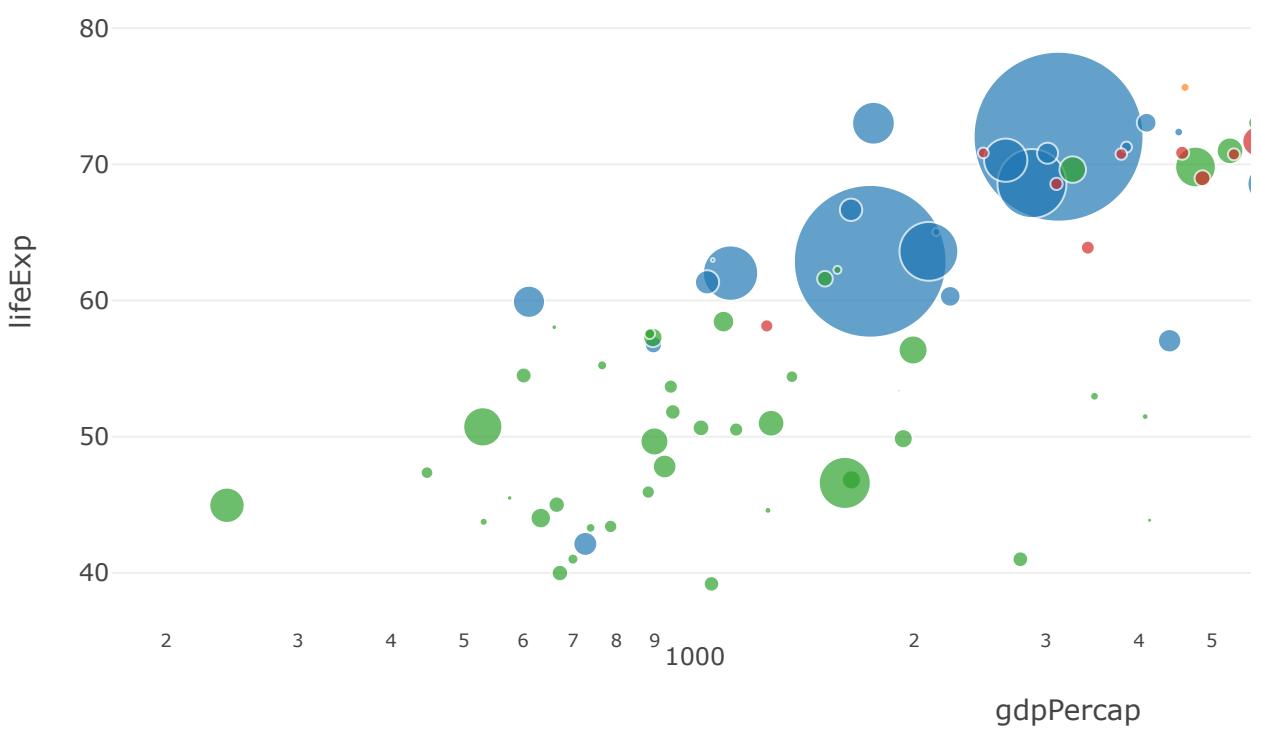
Template: 'plotly\_dark'



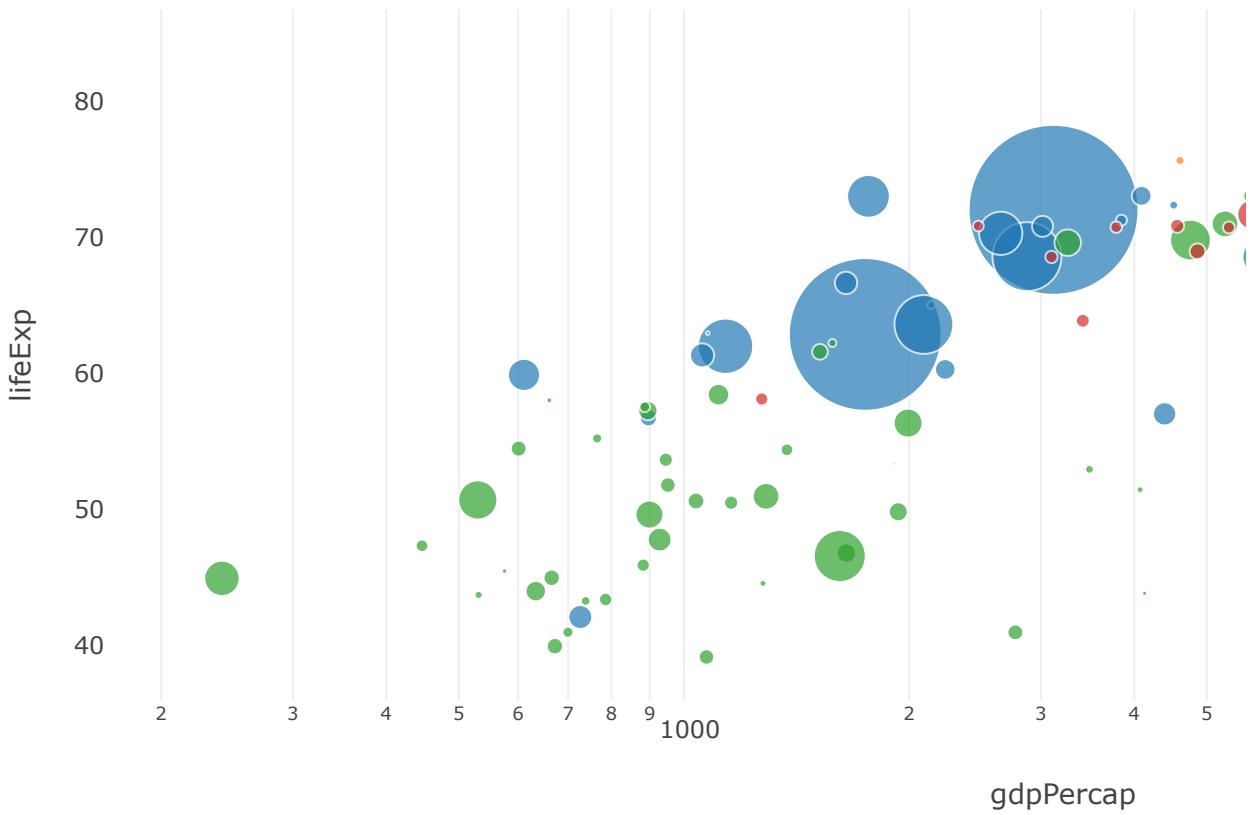
Template: 'pre'



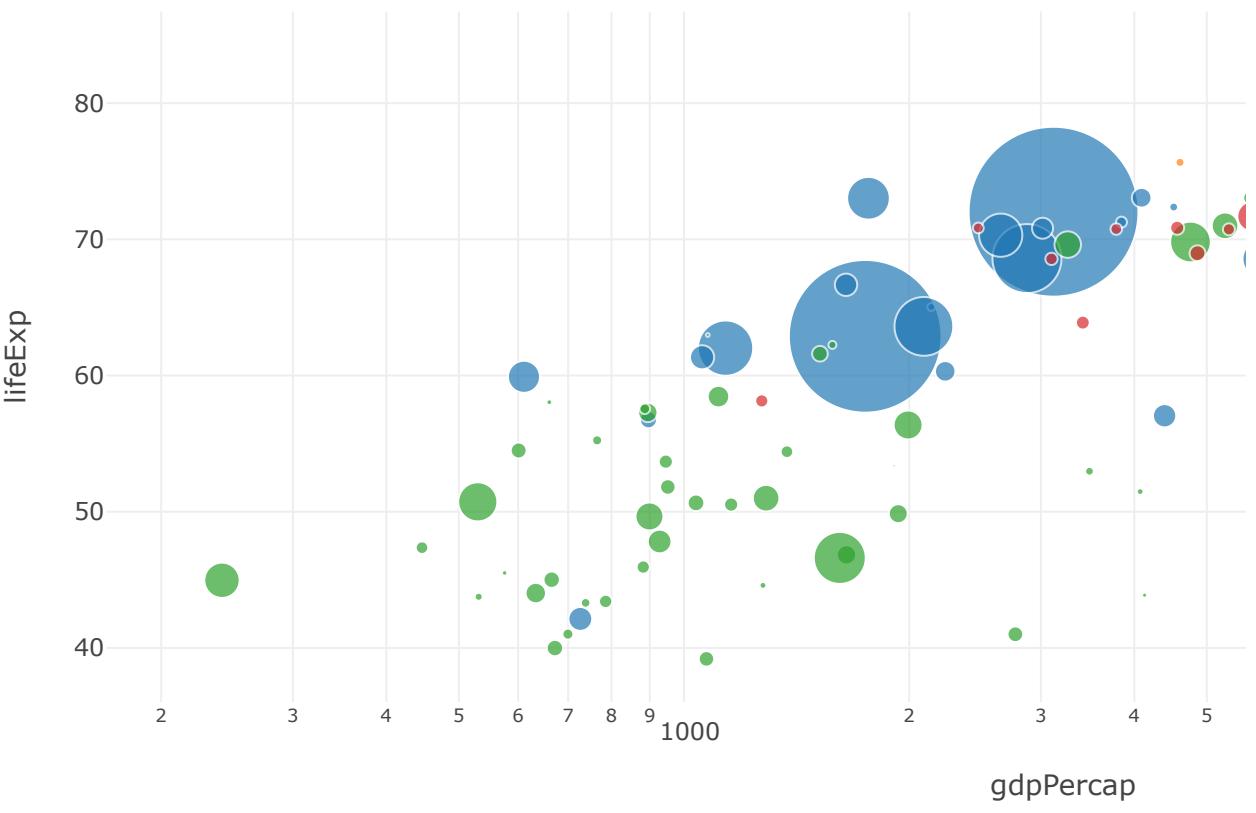
Template: 'xg'



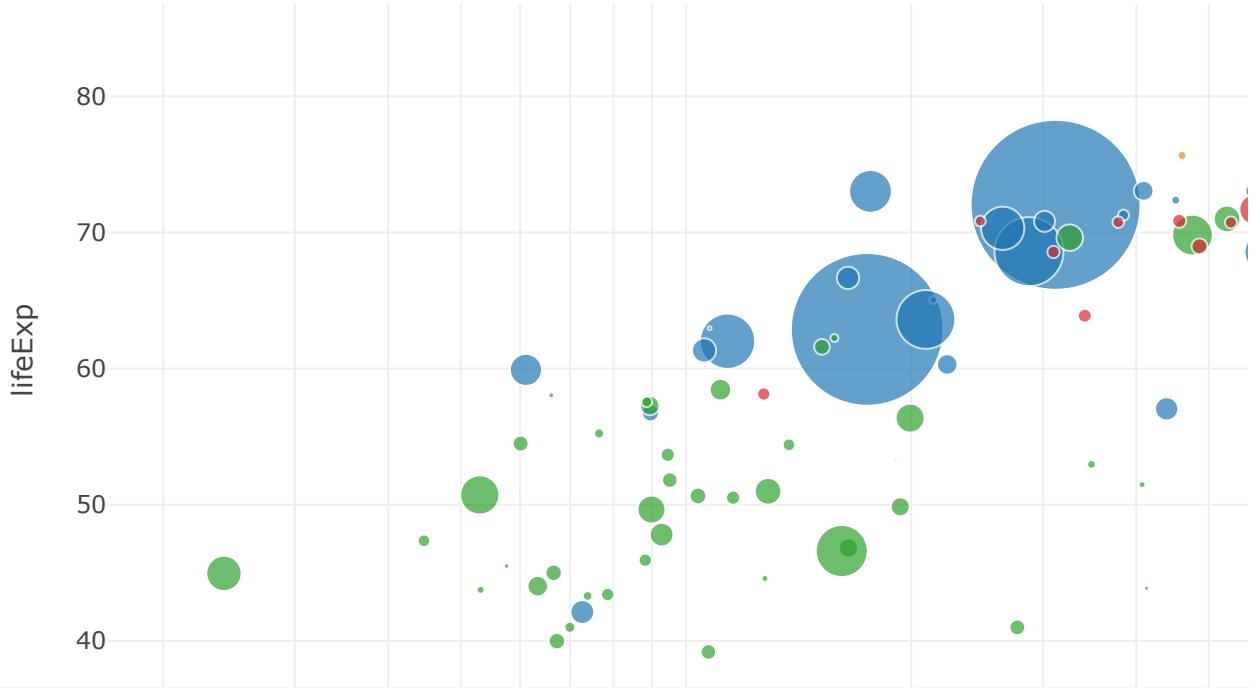
Template: 'yg



Template: 'gi



Template: 'r



```
pio.templates.default = "plotly_white"

fig = px.scatter(gapminder_2002,
                  x='gdpPerCap', y='lifeExp', size='pop', color='continent',
                  log_x=True, size_max=60,
                  title="Template: '%s'" % template)
fig.show()
```

Template: 'none'

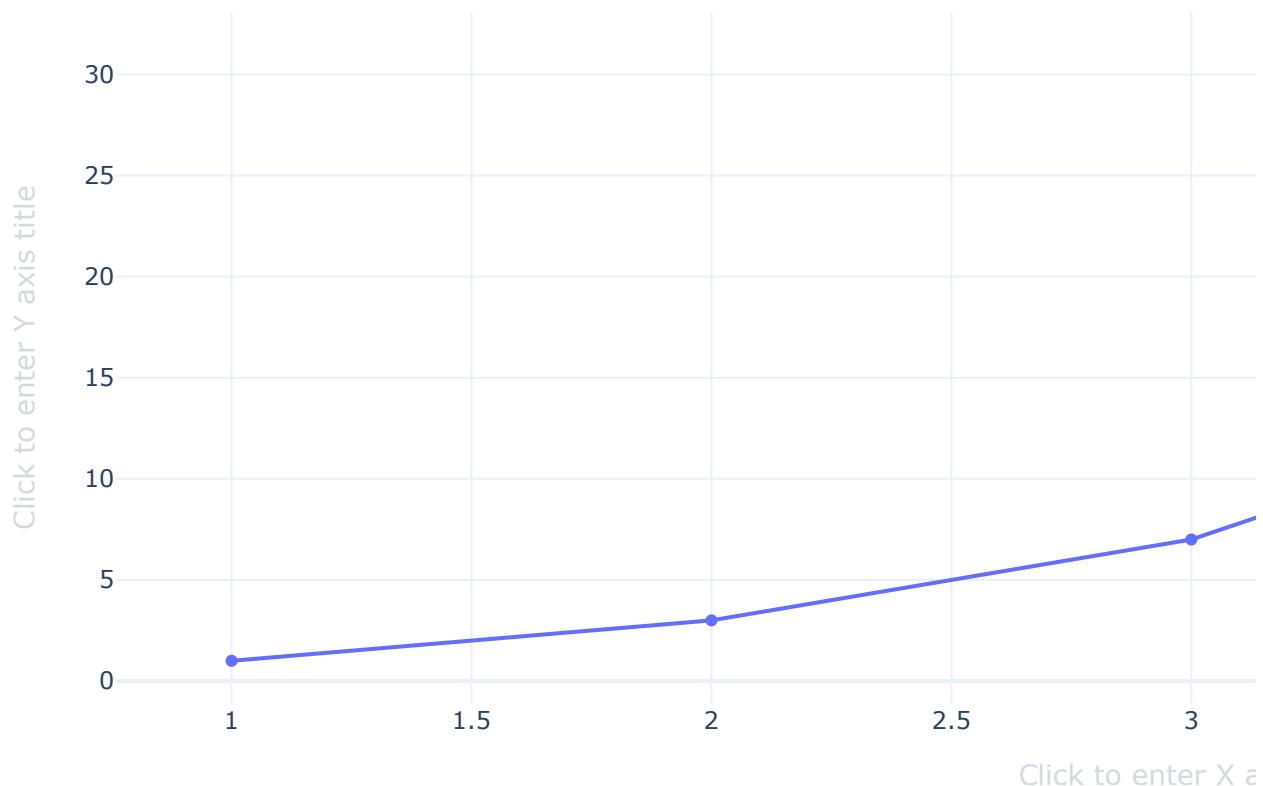
▼ 구성(Configuration)

```
fig = go.Figure()
config = dict({
    'scrollZoom': True,
    'displayModeBar': True,
    'editable': True
})

fig.add_trace(go.Scatter(x = [1, 2, 3, 4, 5],
                        y = [1, 3, 7, 15, 31]))
fig.show(config=config)
```

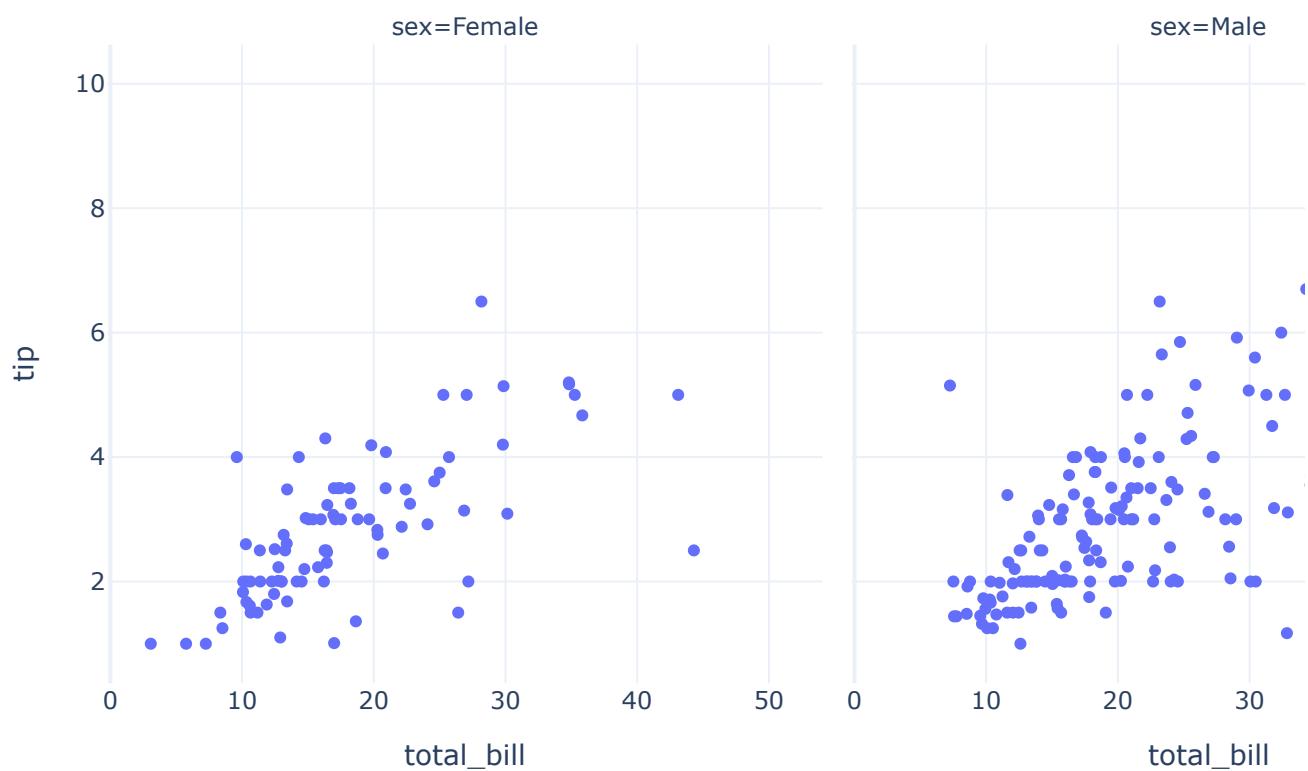


Click to enter Plot title



▼ 높이, 너비 및 여백 조정

```
tig = px.scatter(tips, x='total_bill', y='tip', facet_col='sex',
                  width=800, height=400)
fig.update_layout(margin=dict(l=10, r=20, t=30, b=40),
                  paper_bgcolor='skyblue')
fig.show()
```

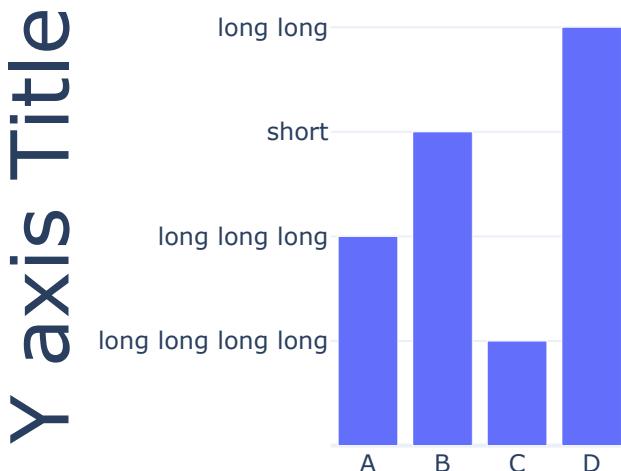


```
fig = go.Figure()
fig.add_trace(go.Scatter(x=np.arange(1, 10),
                         y=np.arange(10, 20)))
fig.update_layout(autosize=False, width=400, height=400,
                  margin=dict(l=10, r=20, t=30, b=40, pad=4),
                  paper_bgcolor='skyblue')
fig.show()
```

```

fig = go.Figure()
fig.add_trace(go.Bar(x=['A', 'B', 'C', 'D'], y=[2, 3, 1, 4]))
fig.update_layout(autosize=False, width=400, height=400,
                  yaxis=dict(
                      title_text='Y axis Title',
                      ticktext=['long long long long',
                                'long long long',
                                'short',
                                'long long'],
                      tickvals=[1, 2, 3, 4],
                      tickmode='array',
                      titlefont=dict(size=40)
                  ))
fig.update_yaxes(automargin=True)
fig.show()

```

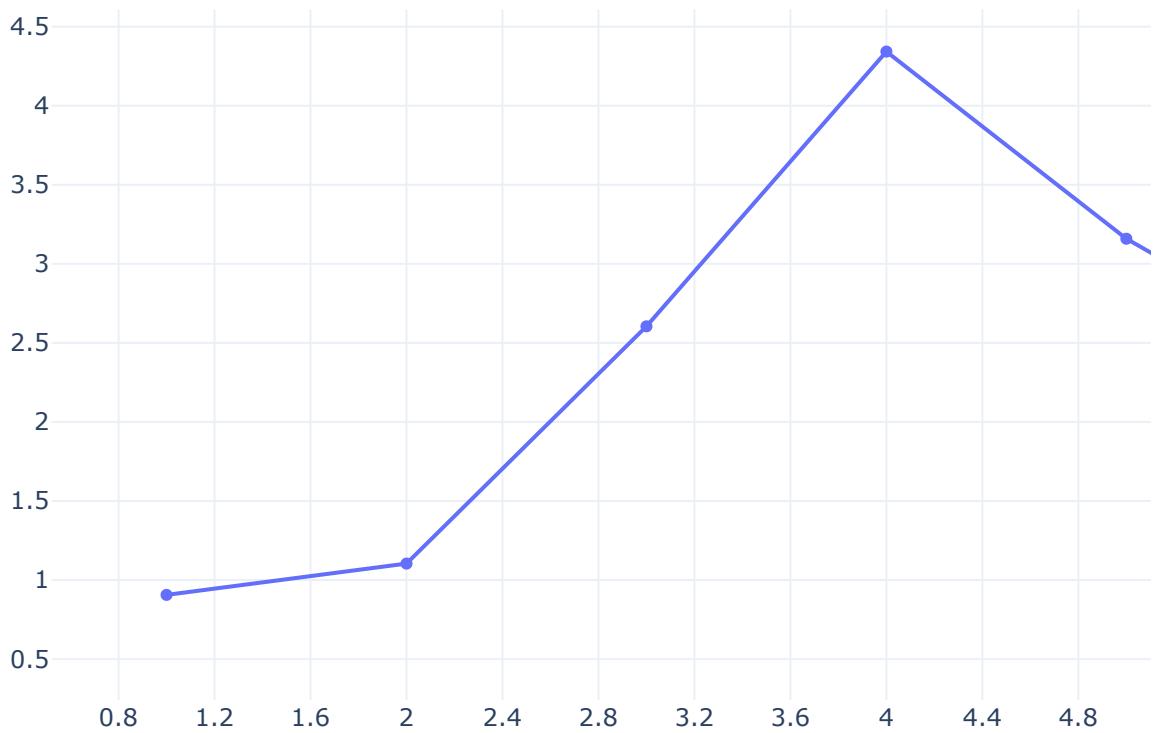


## ▼ 눈금(Tick) 형식 지정

```

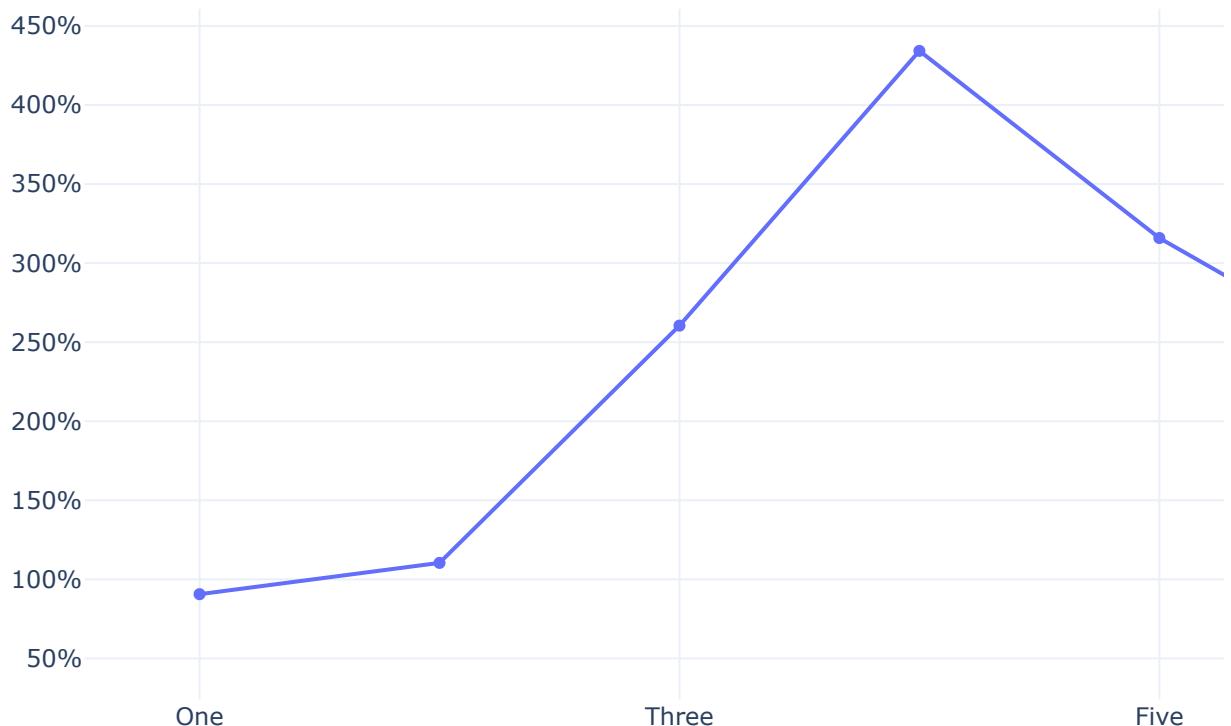
fig = go.Figure(go.Scatter(x = np.arange(1, 10),
                           y = np.random.randn(10).cumsum()))
fig.update_layout(xaxis = dict(tickmode = 'linear',
                               tick0 = 0.8, dtick = 0.4))
fig.show()

```

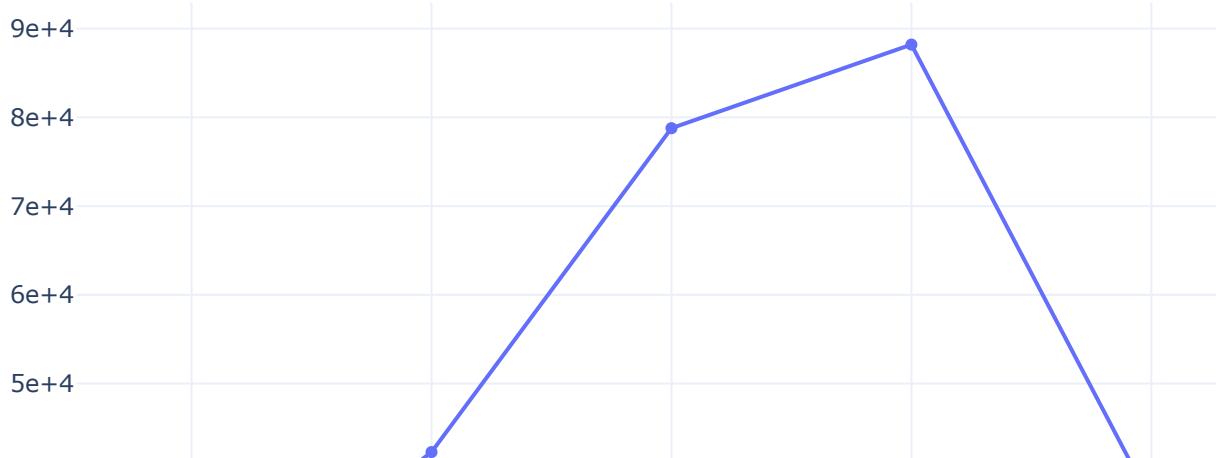


```
fig.update_layout(xaxis = dict(
    tickmode = 'array',
    tickvals = [1, 3, 5, 7, 9],
    ticktext = ['One', 'Three', 'Five', 'Seven', 'Nine']))
fig.show()
```

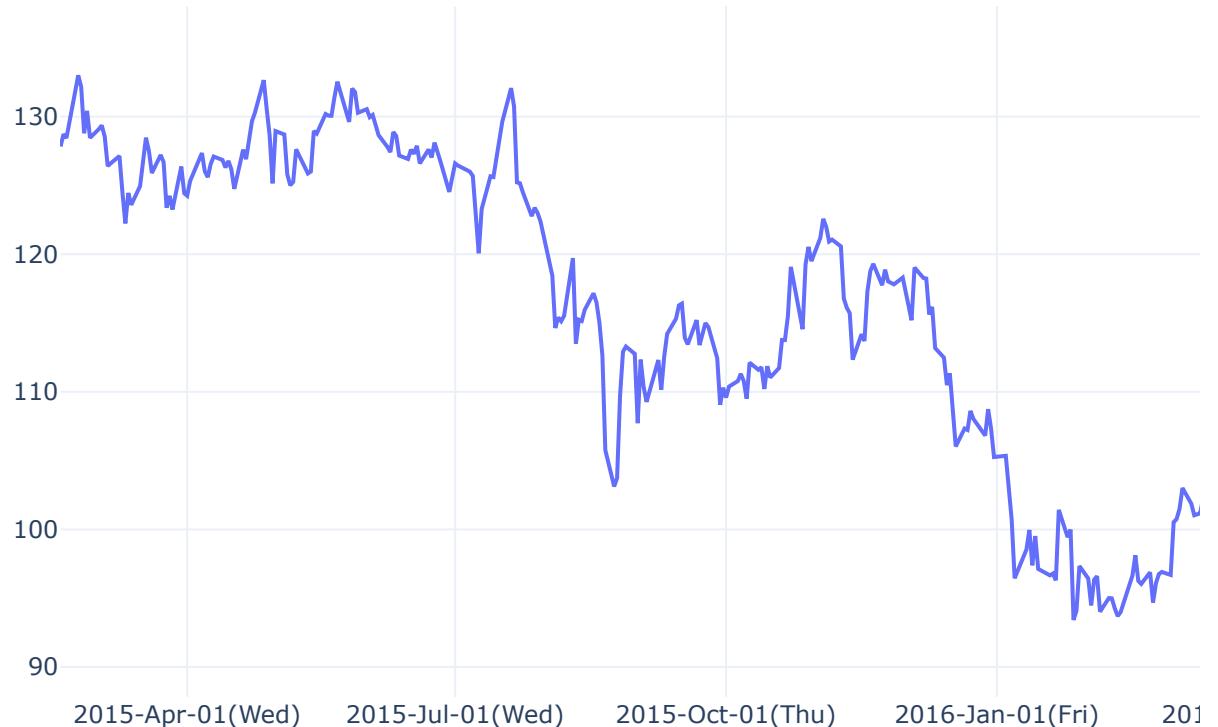
```
4 5  
fig.update_layout(yaxis_tickformat = '%')  
fig.show()
```



```
fig = go.Figure(go.Scatter(x = np.arange(1, 10),  
                           y = np.random.randint(10000, 100000, size=10)))  
fig.update_layout(yaxis=dict(showexponent='all', exponentformat='e'))  
fig.show()
```



```
fig = go.Figure(go.Scatter(x=aapl['Date'], y=aapl['AAPL.Close']))
fig.update_layout(xaxis_tickformat='%Y-%b-%d(%a)')
fig.show()
```



```
fig = go.Figure(go.Scatter(x=aapl['Date'], y=aapl['AAPL.Close']))
fig.update_layout(
    xaxis_tickformatstops=[
```

dict(dtickrange=[None, 1000], value="%H:%M:%S %I ")

```

        dict(dtickrange=[1000, 1000], value="%M-%d-%Y"),
        dict(dtickrange=[1000, 60000], value="%H:%M:%S"),
        dict(dtickrange=[60000, 3600000], value="%H:%M"),
        dict(dtickrange=[3600000, 86400000], value="%H:%M"),
        dict(dtickrange=[86400000, 604800000], value="%b-%e"),
        dict(dtickrange=[604800000, 'M1'], value="%Y-%b"),
        dict(dtickrange=["M12", None], value="%Y")
    ])
fig.show()

```



```

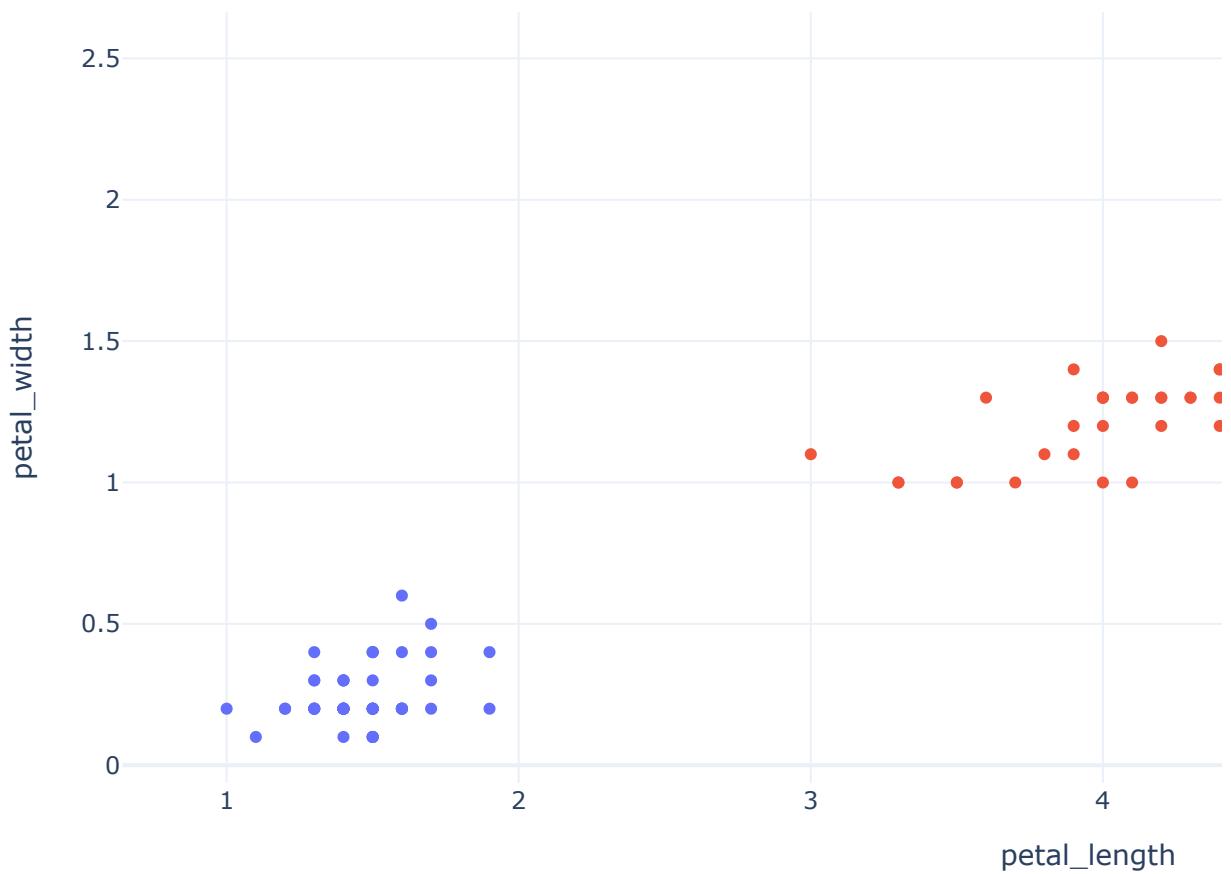
fig = go.Figure(go.Bar(x=['A', 'B', 'C'], y=[1, 2, 3]))
fig.update_xaxes(showgrid=True,
                  ticks='outside',
                  tickson='boundaries',
                  ticklen=20)
fig.show()

```



#### ▼ 글꼴, 제목, 범례 항목 및 축 제목 설정

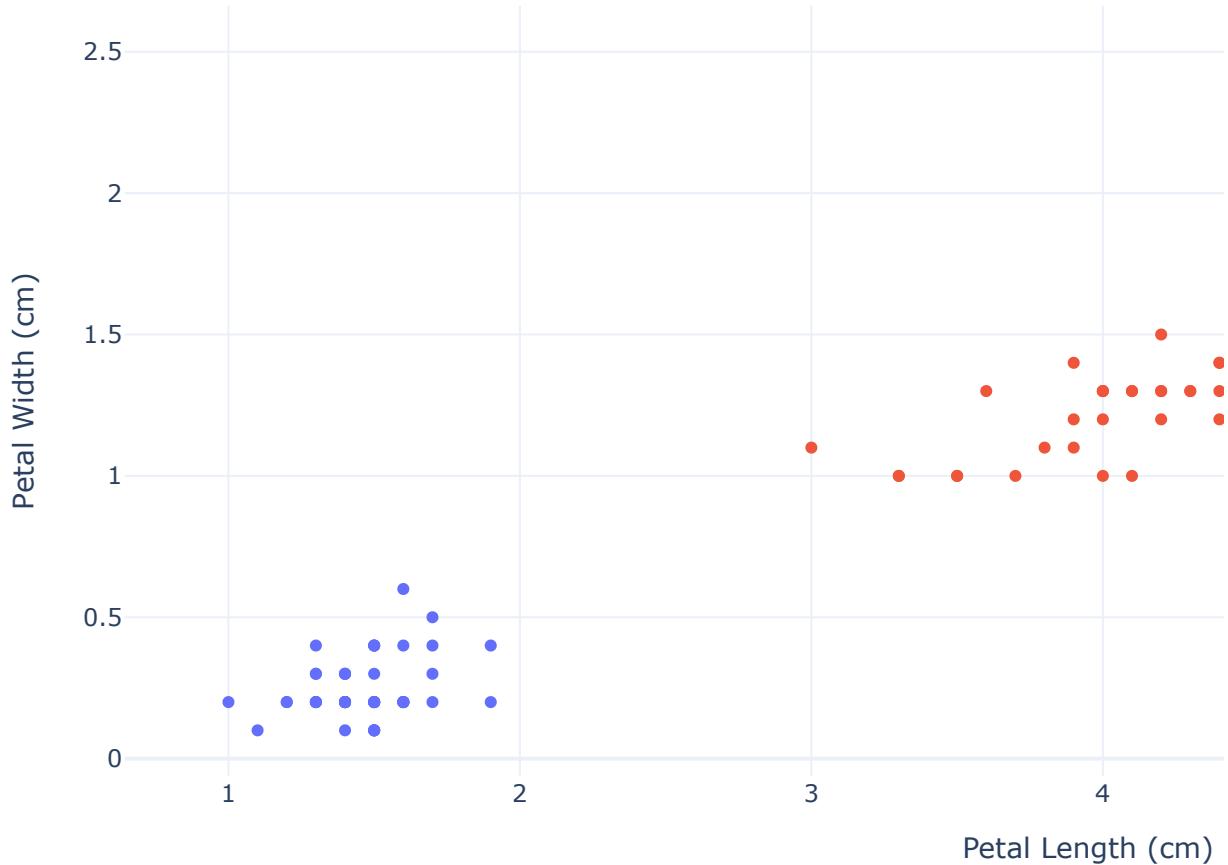
```
fig = px.scatter(iris, x='petal_length', y='petal_width', color='species')
fig.show()
```



```

fig = px.scatter(iris,
                  x='petal_length', y='petal_width', color='species',
                  labels={'petal_length': 'Petal Length (cm)',
                          'petal_width': 'Petal Width (cm)',
                          'species': "Species"})
fig.show()

```



```

fig = go.Figure(go.Scatter(y=[3, 1, 4, 2, 5],
                            x=['Mon', 'Tue', 'Wed', 'Thu', 'Fri']))
fig.update_layout(title={'text': 'Title',
                        'y':0.8, 'x':0.5,
                        'xanchor':'center', 'yanchor':'top'})
fig.show()

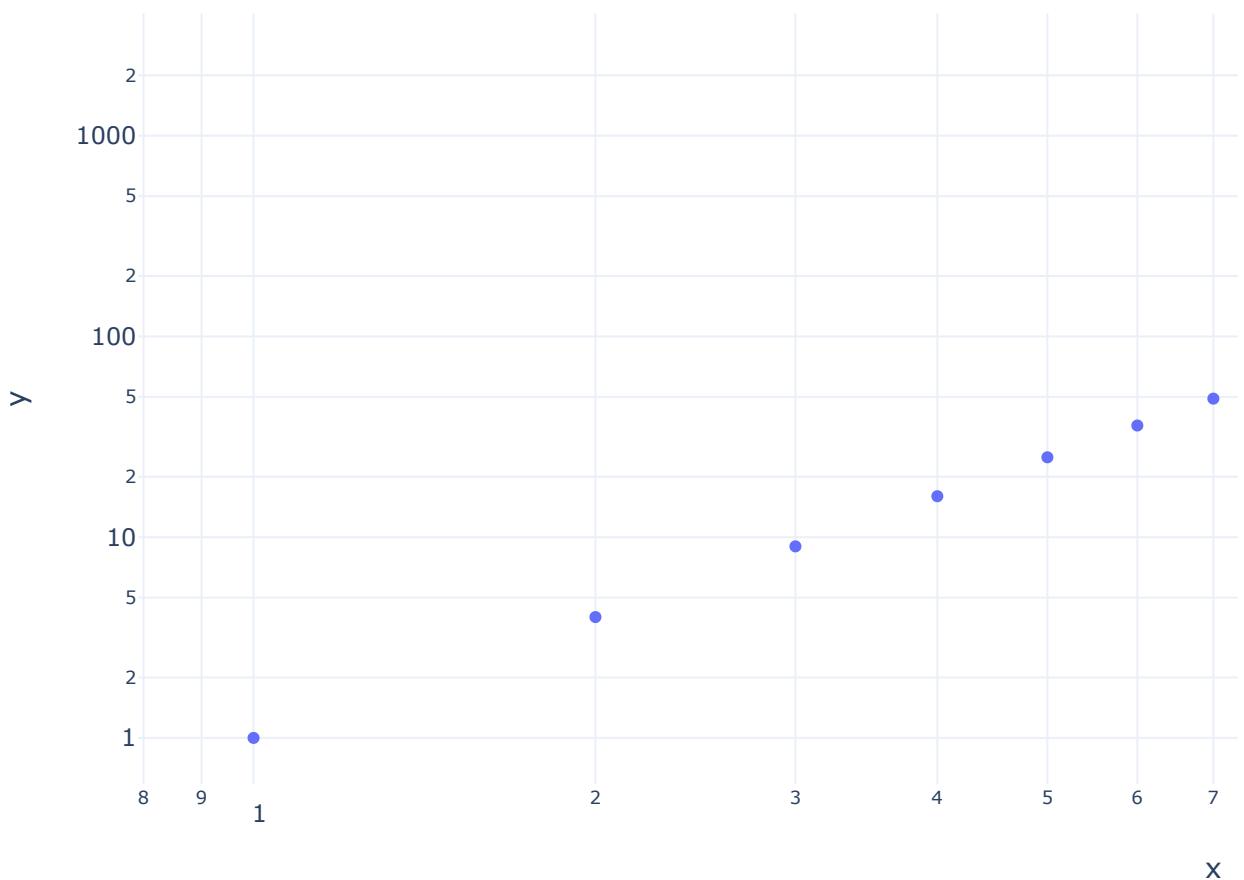
```

Title



▼ 축(Axes)

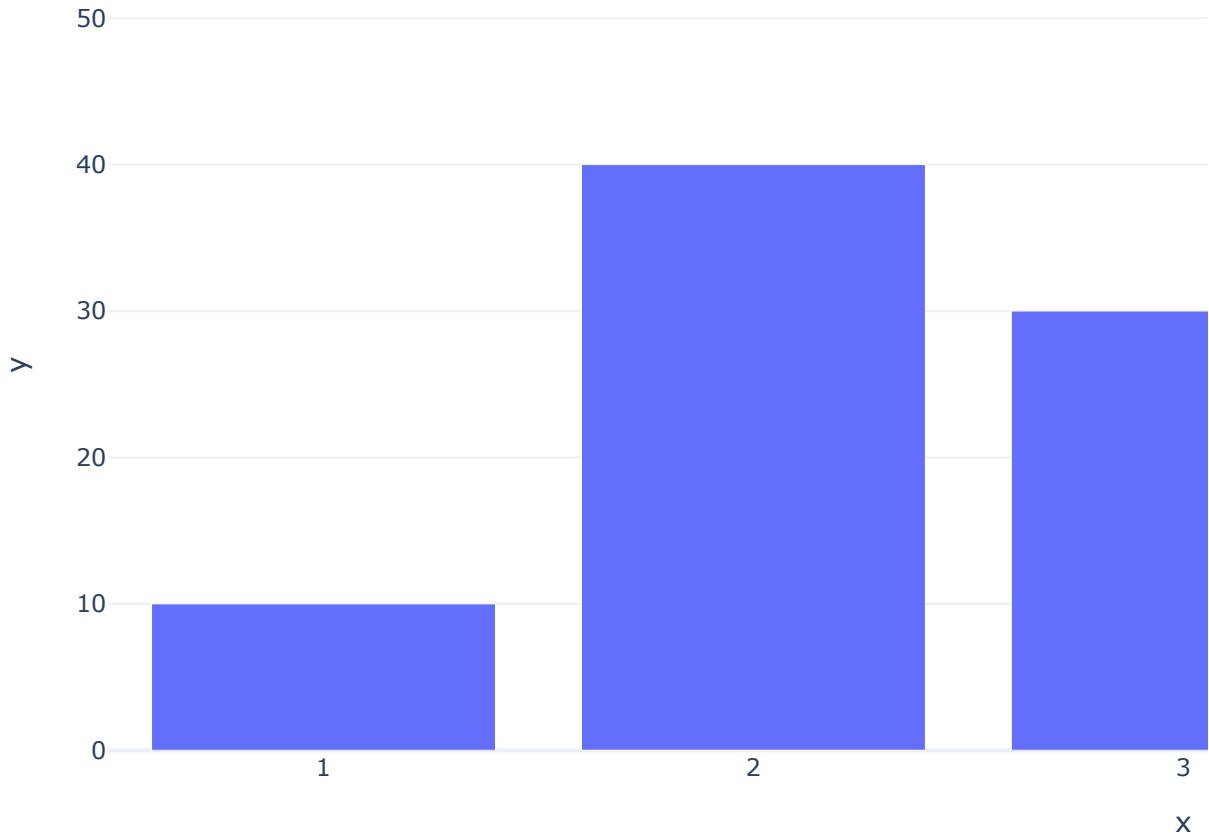
```
x = np.arange(50)
fig = px.scatter(x=x, y=x**2, log_x=True, log_y=True)
fig.show()
```



```

fig = px.bar(x=[1, 2, 3, 4, 5], y=[10, 40, 30, 20, 50])
fig.update_layout(xaxis_type='category')
fig.show()

```

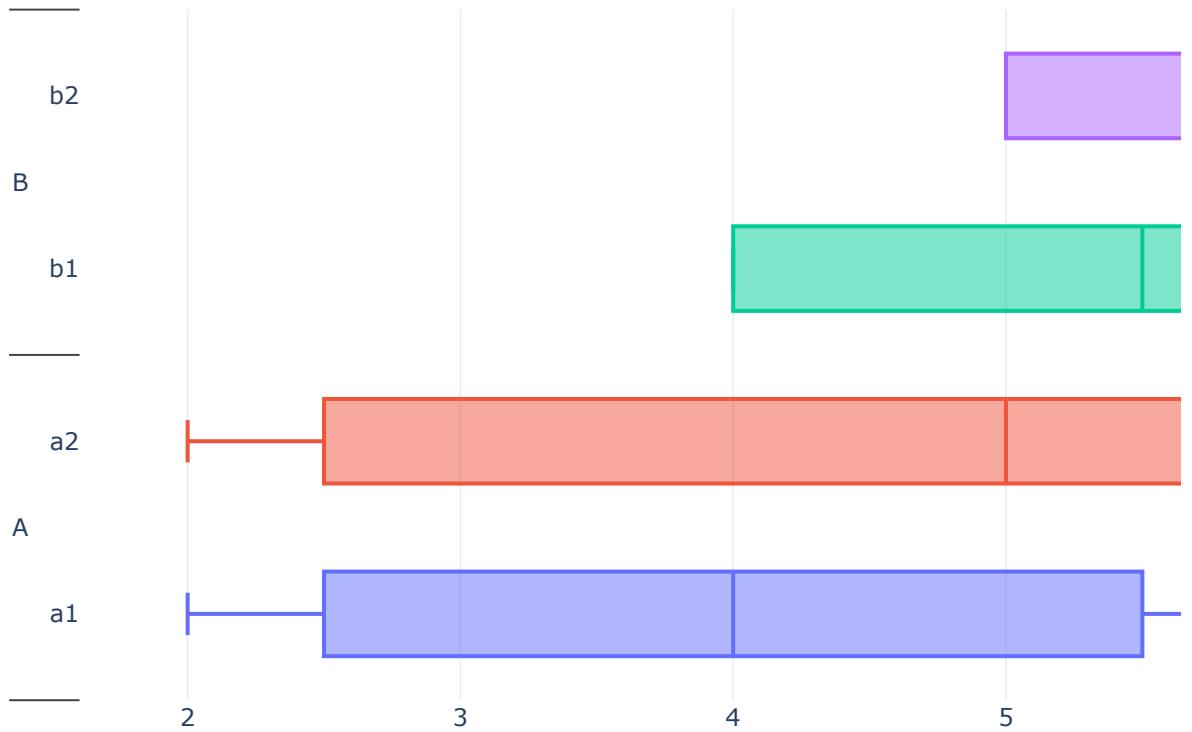


```

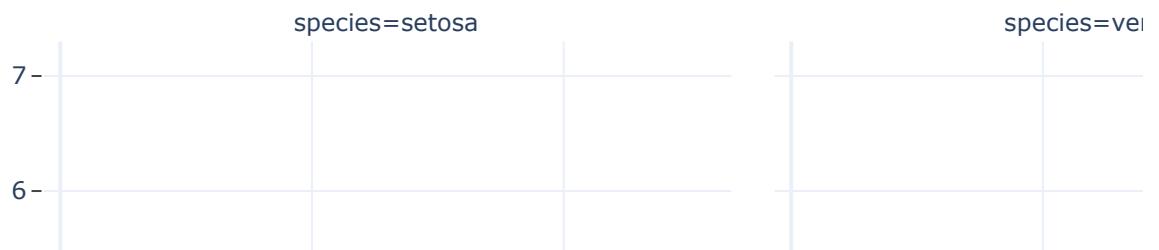
fig = go.Figure()
fig.add_trace(go.Box(x=np.random.randint(1, 10, 4),
                     y=[['A', 'A', 'A', 'A'],
                        ['a1', 'a1', 'a1', 'a1']],
                     name='a1',
                     orientation='h'))
fig.add_trace(go.Box(x=np.random.randint(1, 10, 4),
                     y=[['A', 'A', 'A', 'A'],
                        ['a2', 'a2', 'a2', 'a2']],
                     name='a2',
                     orientation='h'))
fig.add_trace(go.Box(x=np.random.randint(1, 10, 4),
                     y=[['B', 'B', 'B', 'B'],
                        ['b1', 'b1', 'b1', 'b1']],
                     name='b1',
                     orientation='h'))
fig.add_trace(go.Box(x=np.random.randint(1, 10, 4),
                     y=[['B', 'B', 'B', 'B'],
                        ['b2', 'b2', 'b2', 'b2']],
                     name='b2',
                     orientation='h'))
fig.update_layout()

```

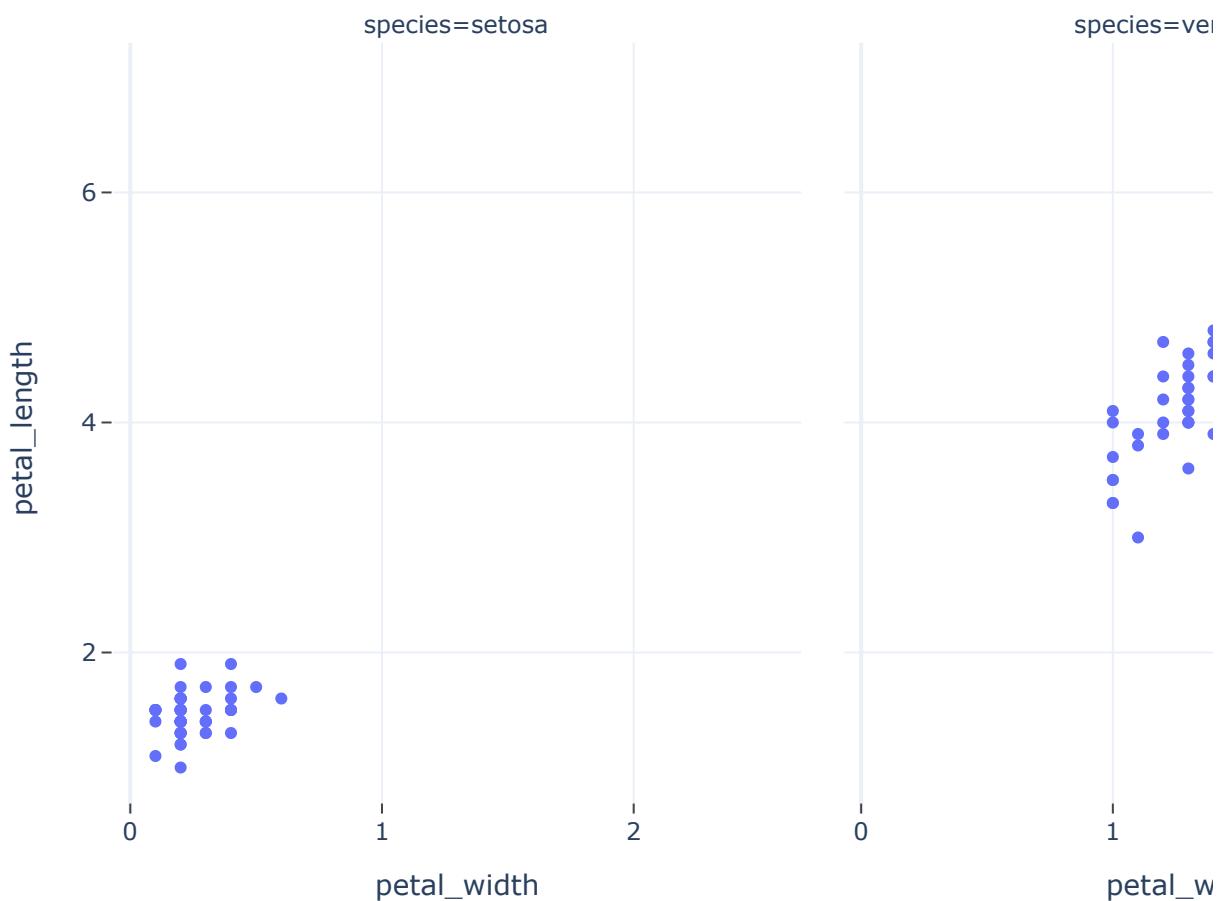
```
fig.show()
```



```
fig = px.scatter(iris, x='petal_width', y='petal_length', facet_col='species')
fig.update_xaxes(ticks='inside')
fig.update_yaxes(ticks='outside', col=1)
fig.show()
```



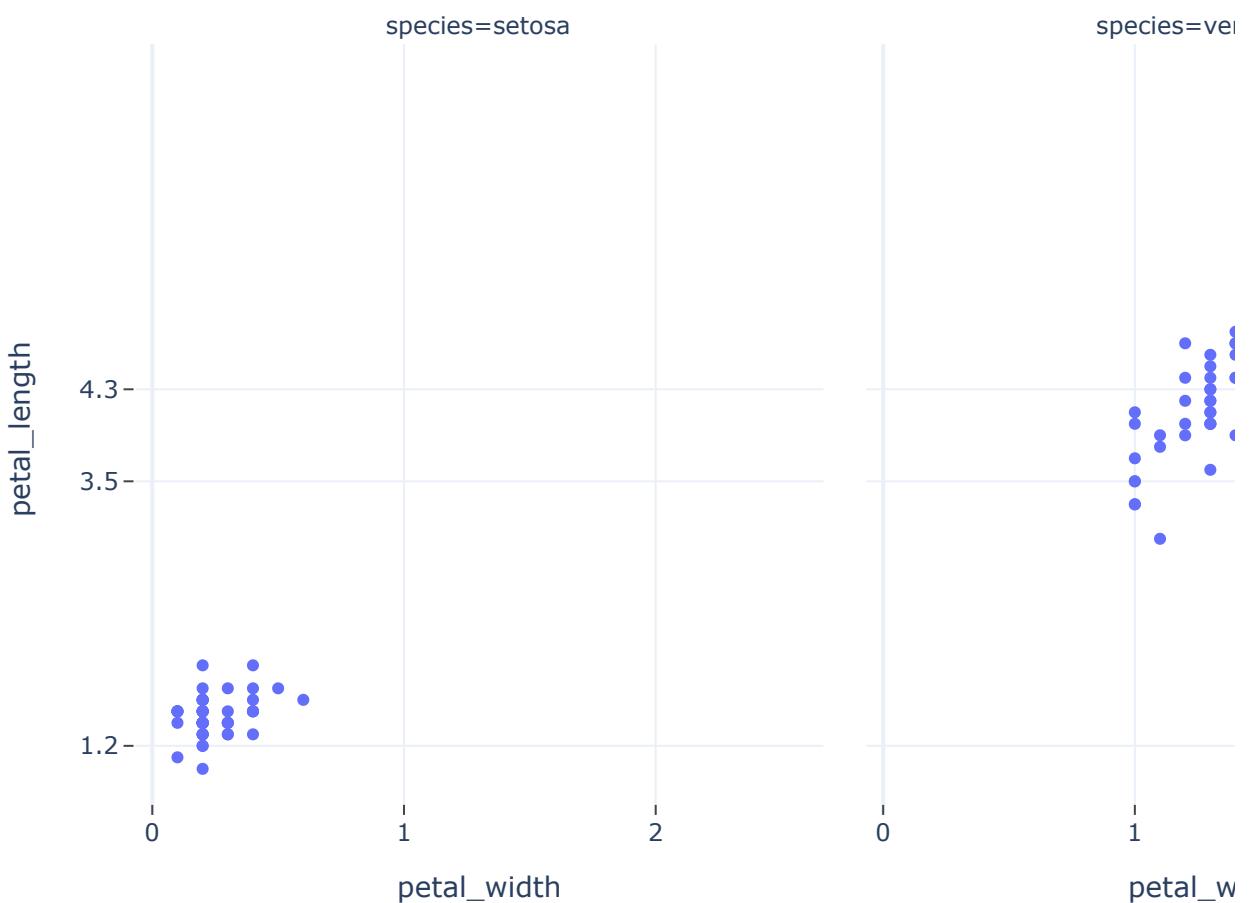
```
fig.update_yaxes(nticks=5)  
fig.show()
```



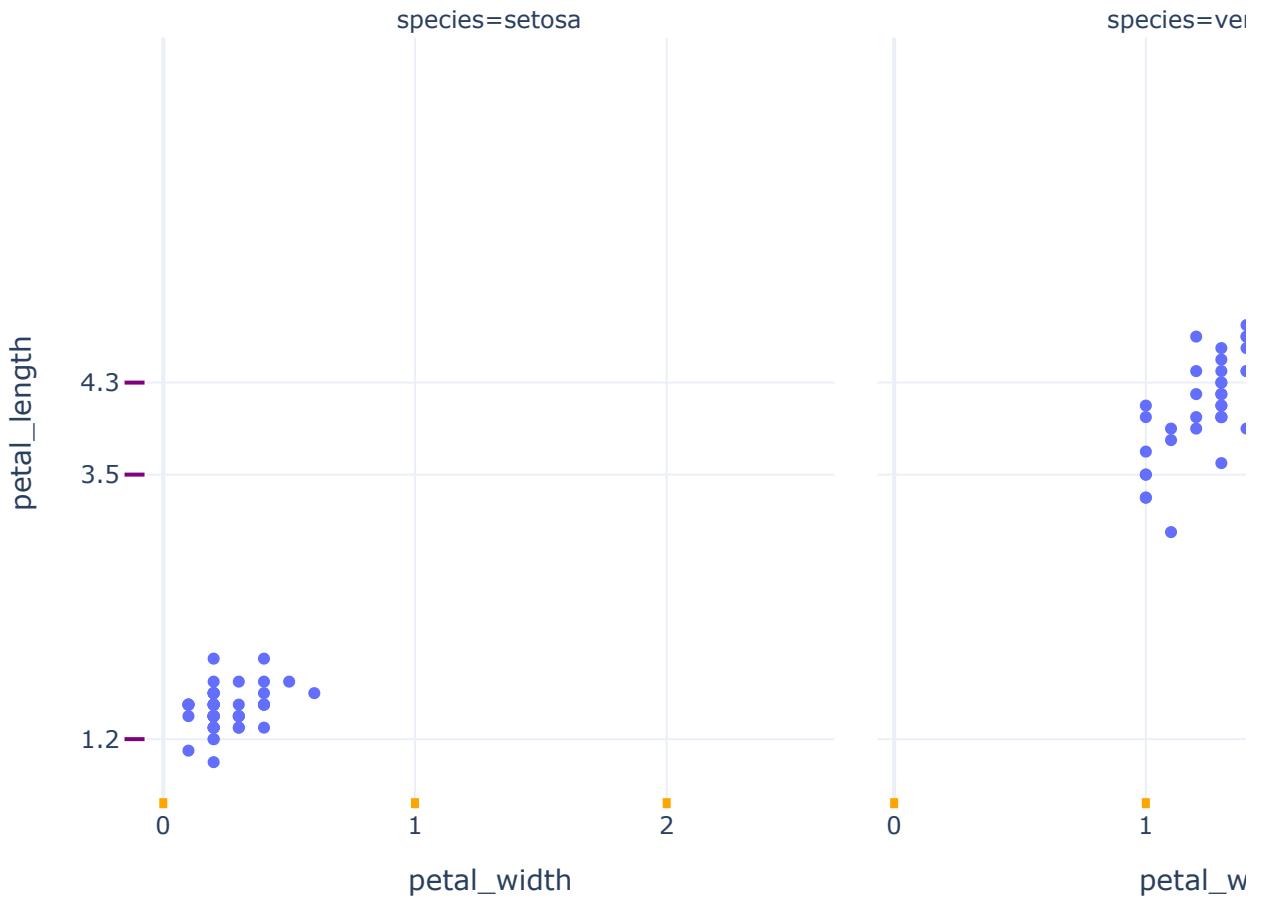
```
fig.update_yaxes(tick0=0.5, dtick=0.25)  
fig.show()
```



```
fig.update_yaxes(tickvals=[1.2, 3.5, 4.3, 7.5])
fig.show()
```

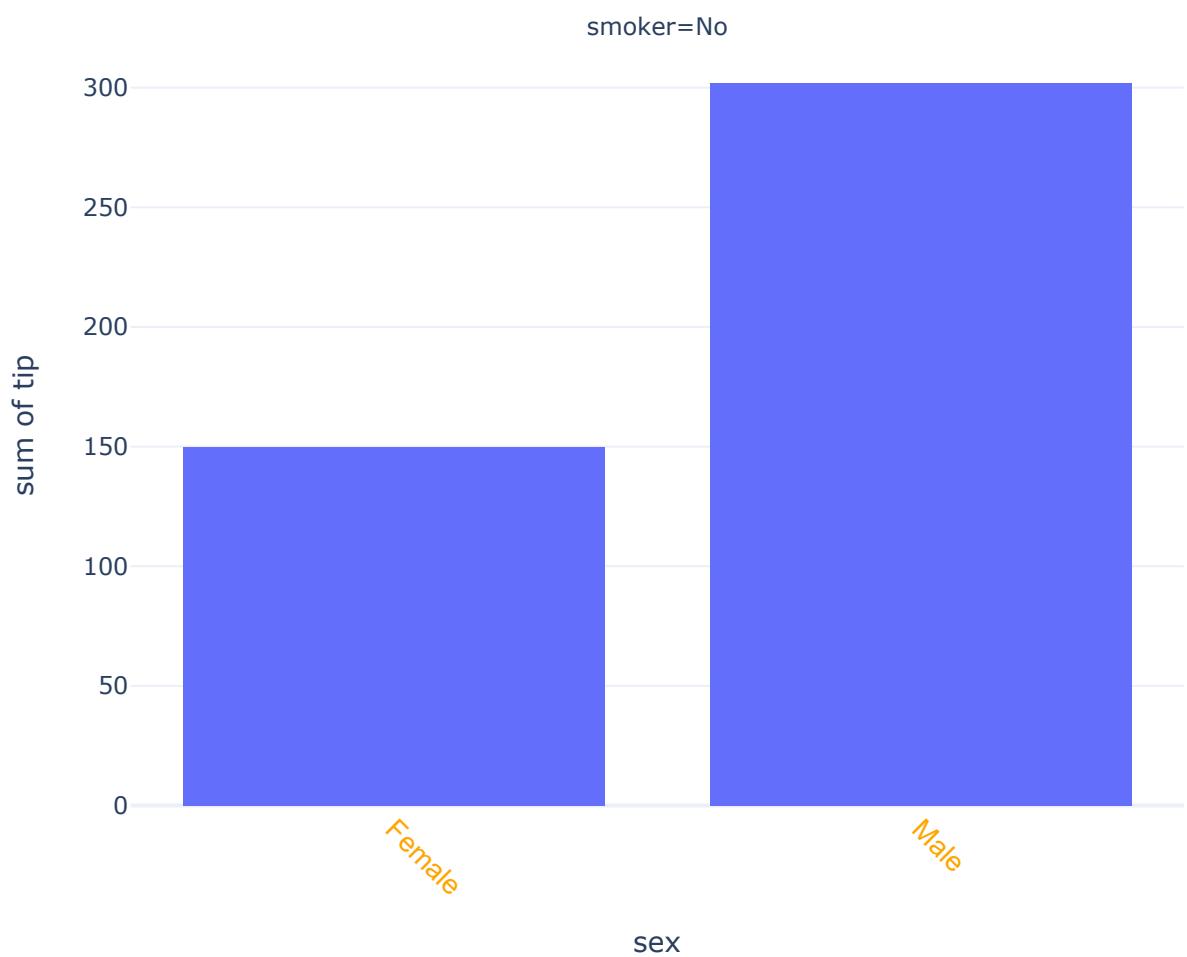


```
fig.update_xaxes(ticks='inside', tickwidth=4, tickcolor='orange', ticklen=5)
fig.update_yaxes(ticks='outside', tickwidth=2, tickcolor='purple', ticklen=10, col=1)
fig.show()
```

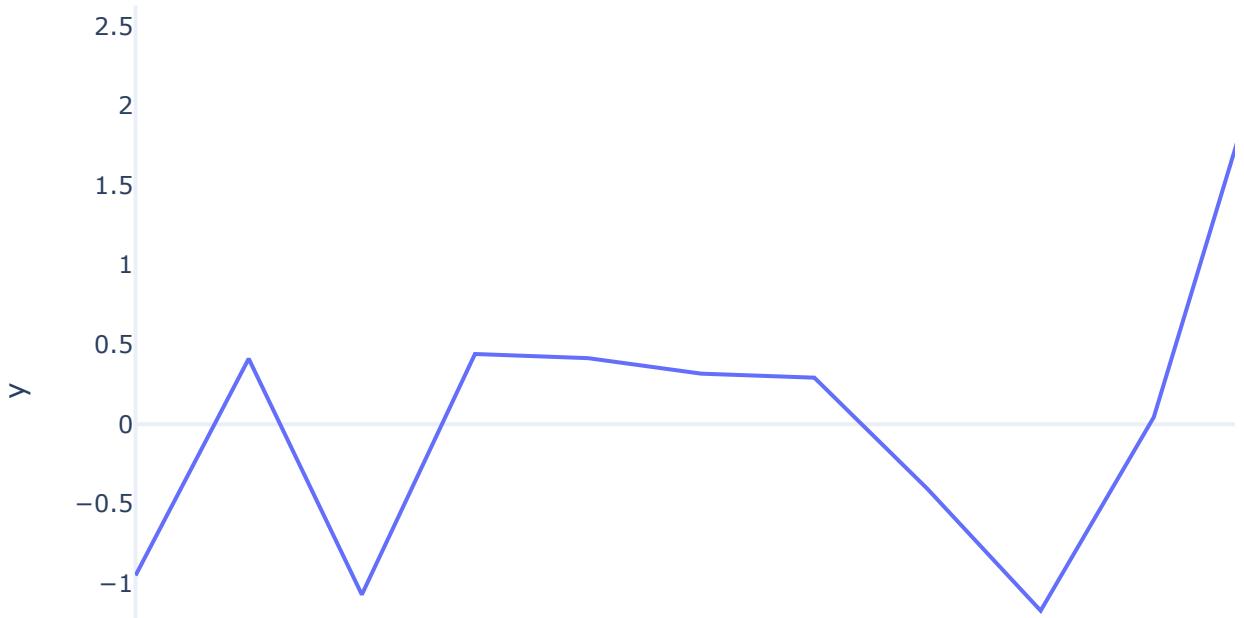


```
fig.update_xaxes(showticklabels=False)
fig.update_yaxes(showticklabels=True)
fig.show()
```

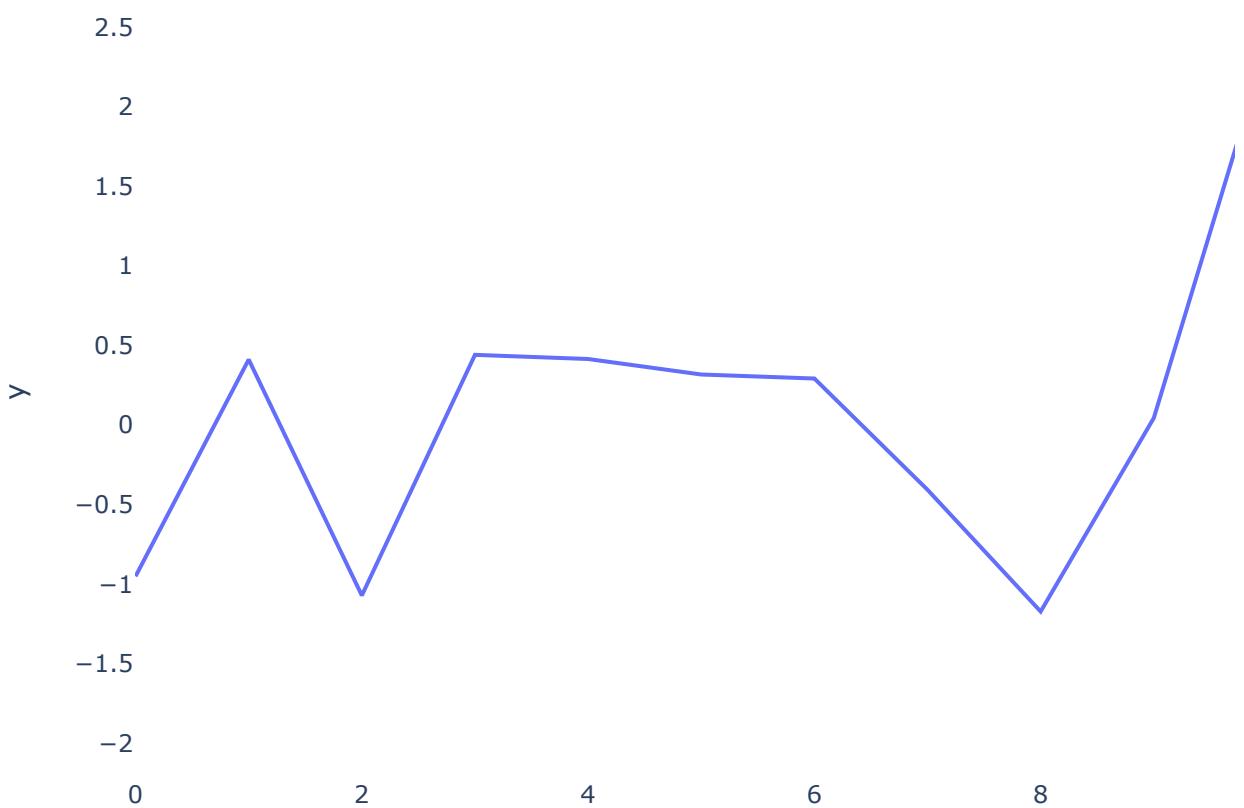
```
species=setosa          species=vei  
fig = px.histogram(tips, x='sex', y='tip', histfunc='sum', facet_col='smoker')  
fig.update_xaxes(tickangle=45, tickfont=dict(family='Arial', color='orange', size=14))  
fig.show()
```



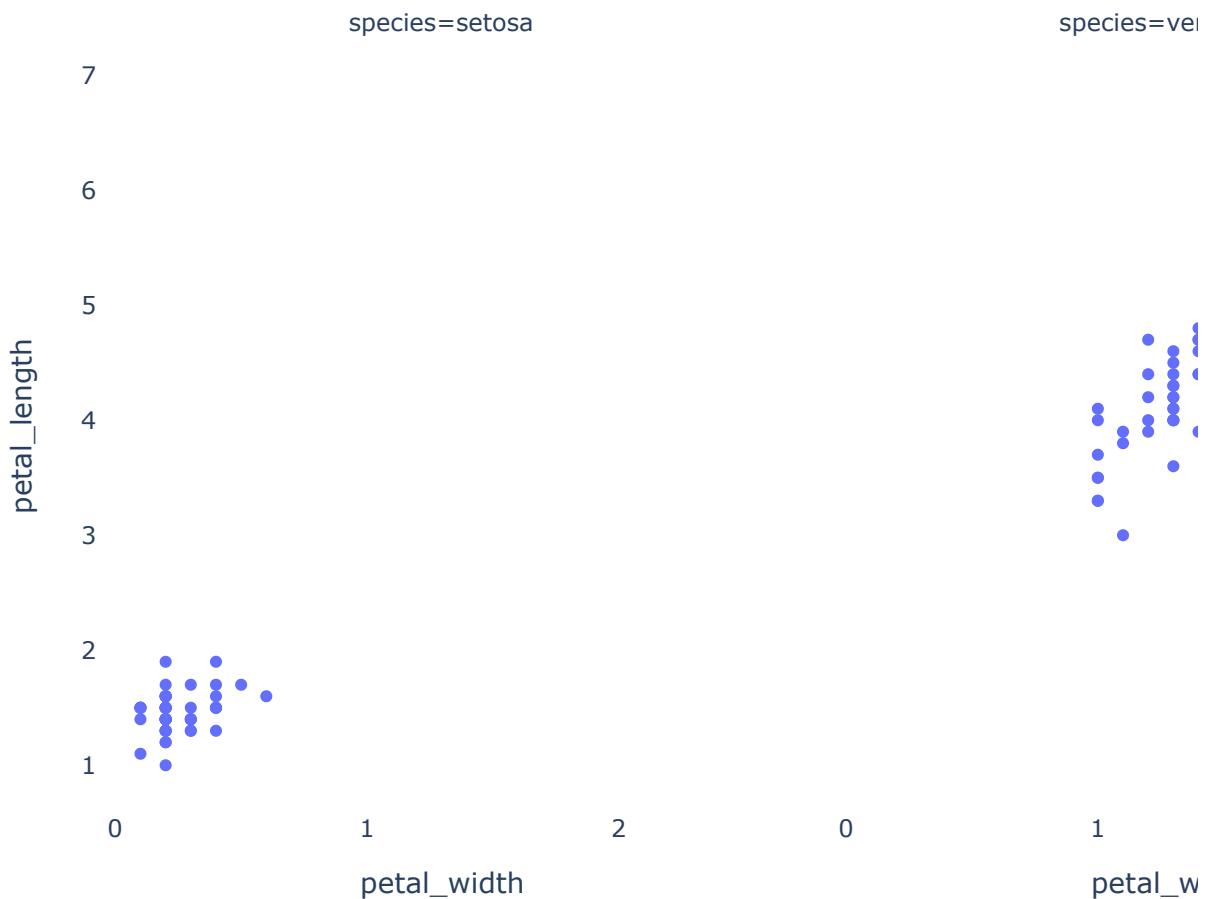
```
fig = px.line(y=np.random.randn(20))  
fig.update_layout(xaxis_showgrid=False, yaxis_showgrid=False)  
fig.show()
```



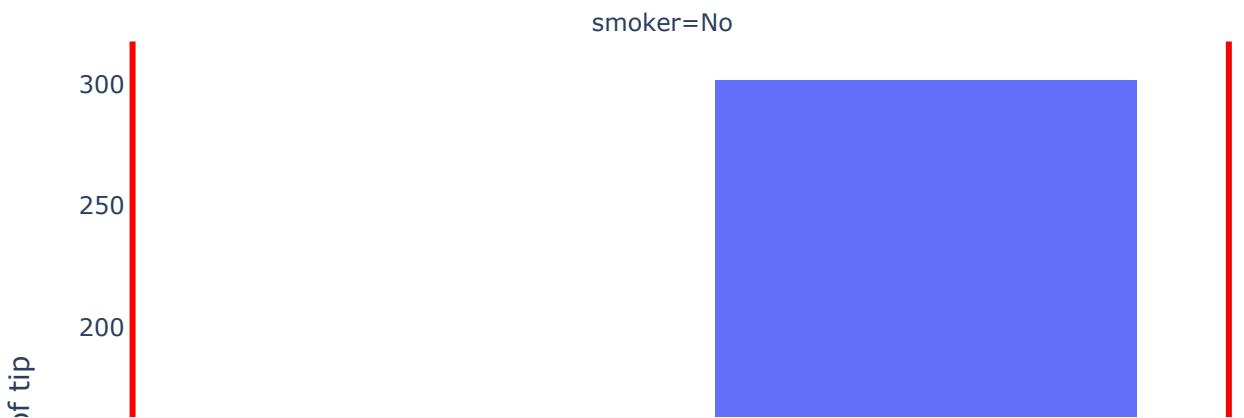
```
fig.update_layout(  
    xaxis=dict(showgrid=False, zeroline=False),  
    yaxis=dict(showgrid=False, zeroline=False),  
)  
fig.show()
```



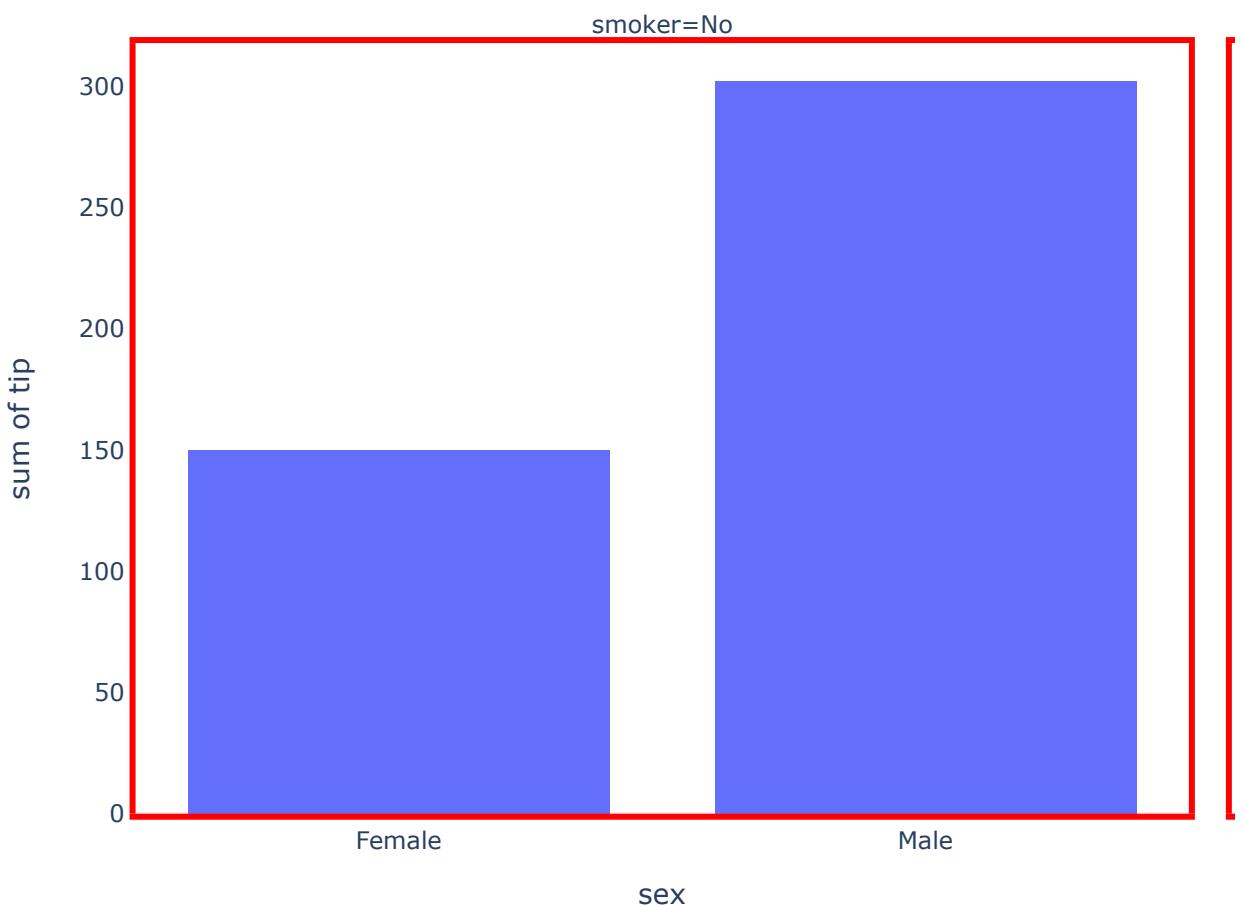
```
fig = px.scatter(iris, x='petal_width', y='petal_length', facet_col='species')
fig.update_xaxes(showgrid=False, zeroline=False)
fig.update_yaxes(showgrid=False, zeroline=False)
fig.show()
```



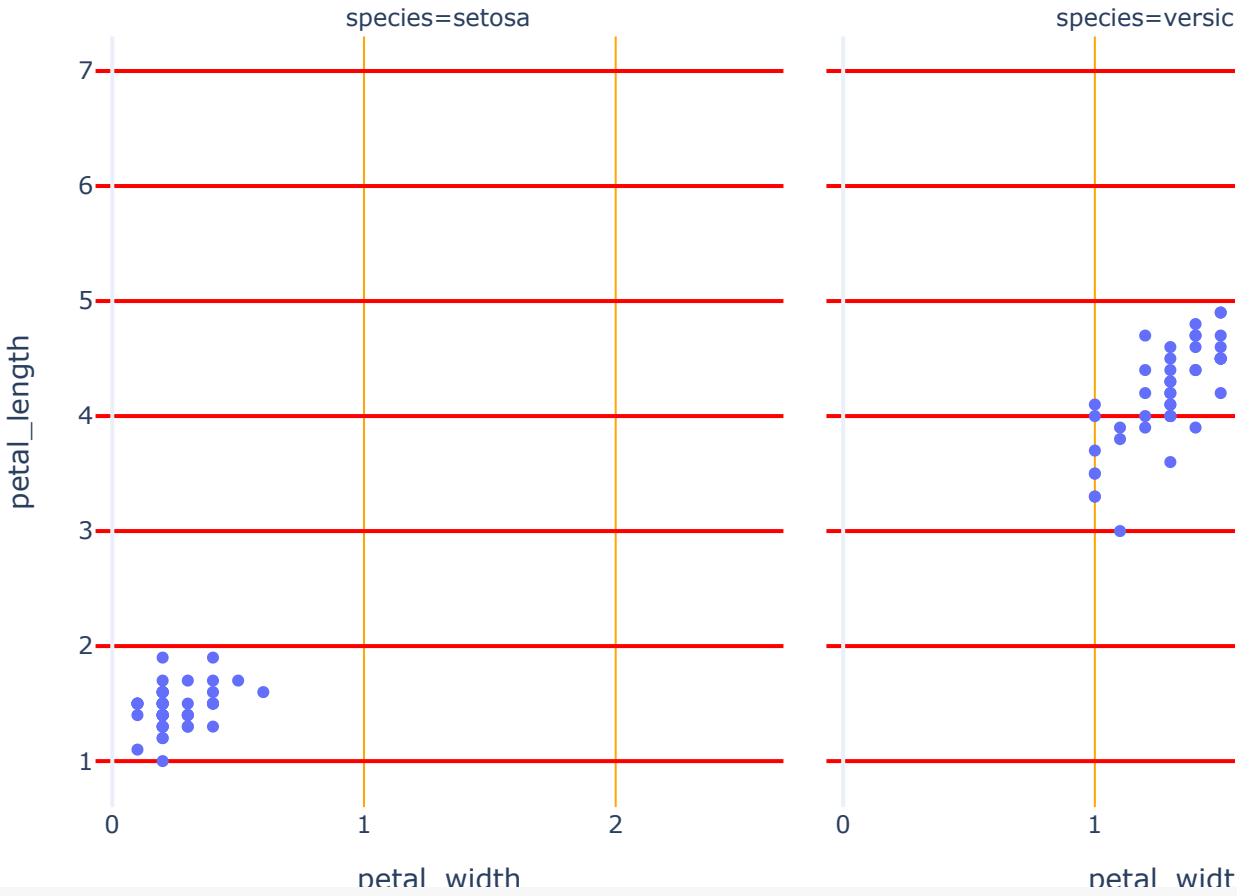
```
fig = px.histogram(tips, x='sex', y='tip', histfunc='sum', facet_col='smoker')
fig.update_xaxes(showgrid=False, linewidth=3, linecolor='red')
fig.update_yaxes(showgrid=False, linewidth=3, linecolor='red')
fig.show()
```



```
fig.update_xaxes(showgrid=False, linewidth=3, linecolor='red', mirror=True)
fig.update_yaxes(showgrid=False, linewidth=3, linecolor='red', mirror=True)
fig.show()
```

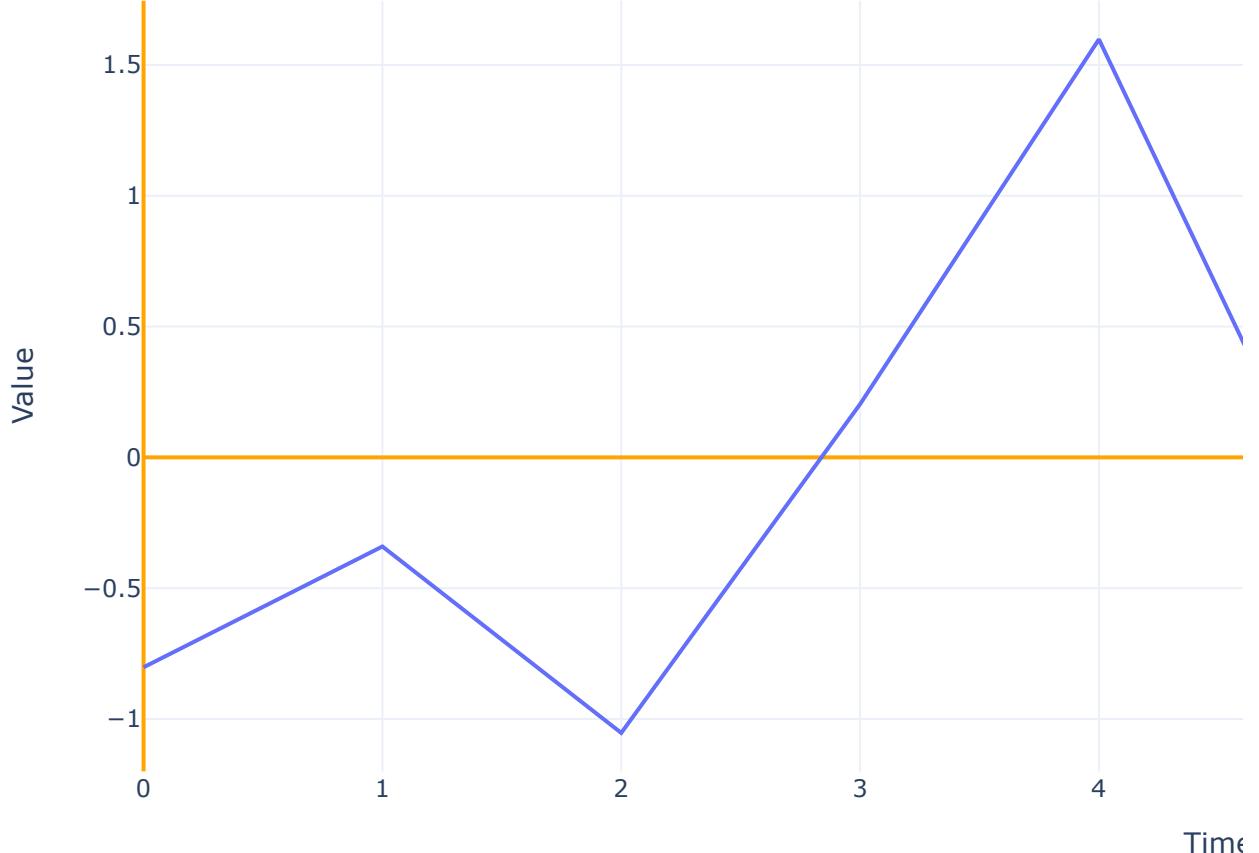


```
fig = px.scatter(iris, x='petal_width', y='petal_length', facet_col='species')
fig.update_xaxes(showgrid=True, gridwidth=1, gridcolor='orange')
fig.update_yaxes(showgrid=True, gridwidth=2, gridcolor='red')
fig.show()
```



```
fig = px.line(y=np.random.randn(10))
fig.update_xaxes(zeroline=True, zerolinewidth=2, zerolinecolor='orange')
fig.update_yaxes(zeroline=True, zerolinewidth=2, zerolinecolor='orange')
fig.show()
```

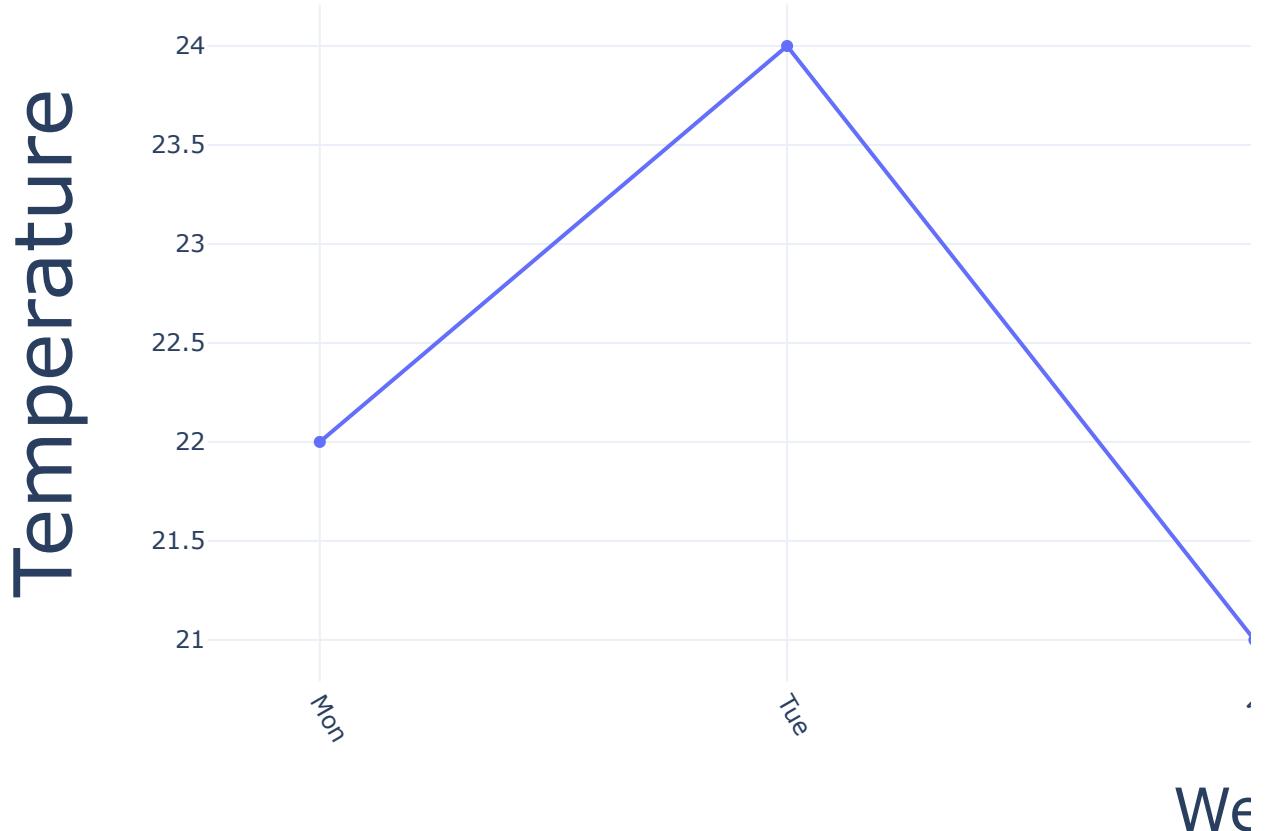
```
fig.update_xaxes(title_text = 'Time')
fig.update_yaxes(title_text = 'Value')
fig.show()
```



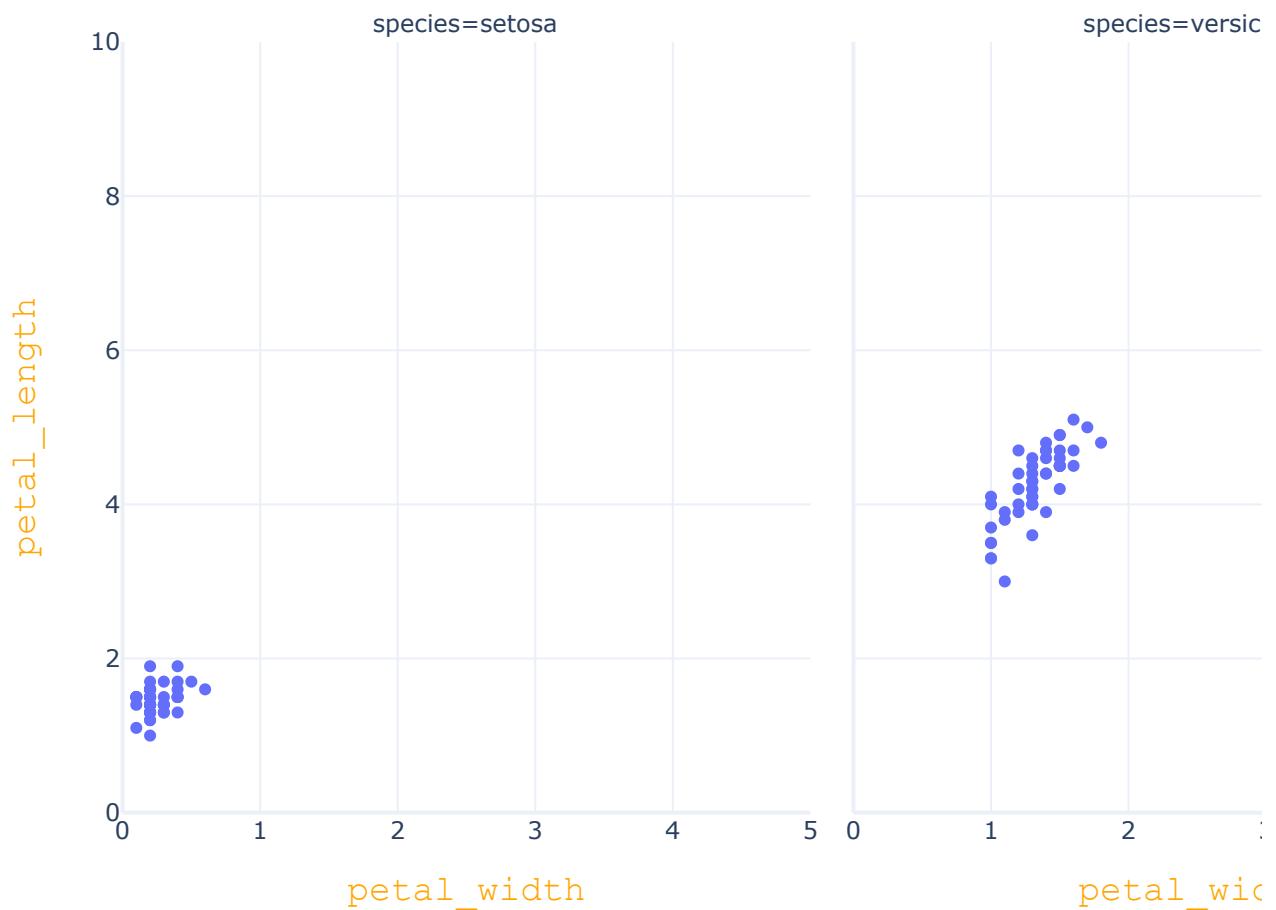
```
fig = go.Figure(go.Scatter(
    mode='lines+markers',
    y=[22, 24, 21, 22, 23],
    x=['Mon', 'Tue', 'Wed', 'Thu', 'Fri']
))
```

```
fig.update_layout(
    xaxis=dict(
        tickangle=60,
        title_text='Week',
        title_font={'size':30},
        title_standoff=20),
    yaxis=dict(
        title_text='Temperature',
        title_font={'size':40},
        title_standoff=40)
```

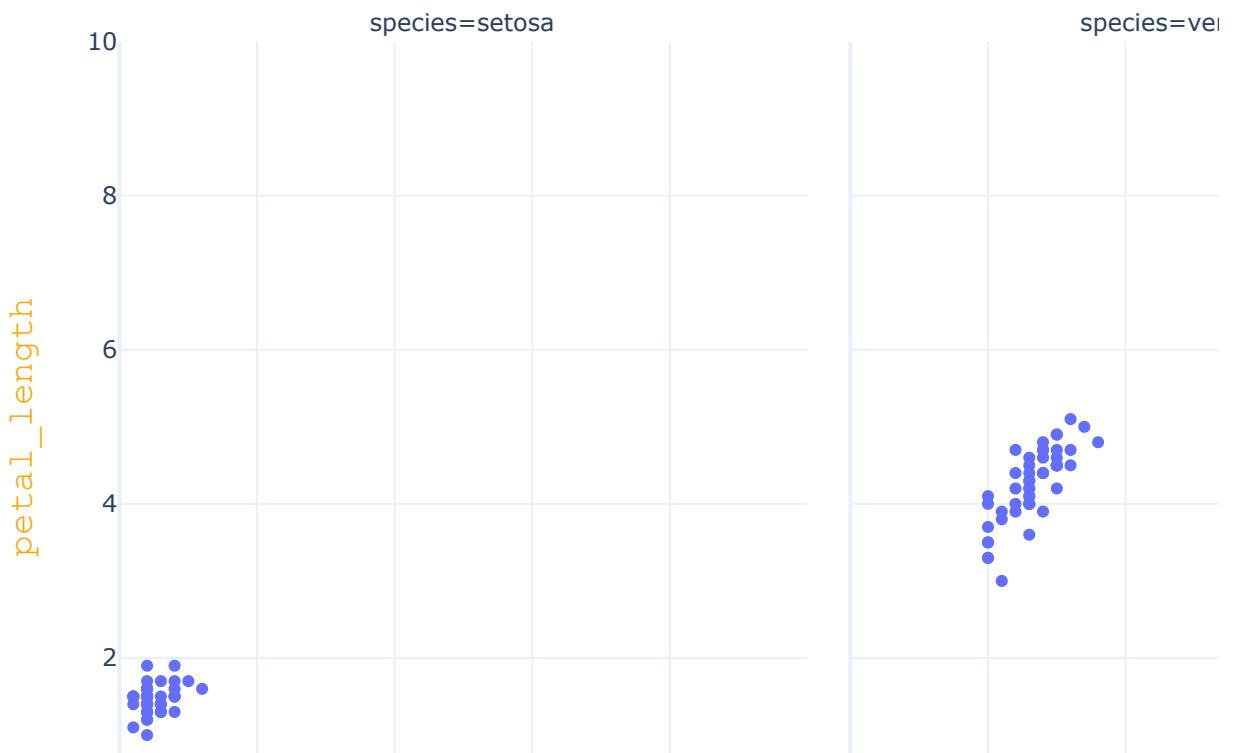
```
)  
fig.show()
```



```
fig = px.scatter(iris, x='petal_width', y='petal_length', facet_col='species')  
fig.update_xaxes(title_font=dict(size=18, family='Courier', color='orange'))  
fig.update_yaxes(title_font=dict(size=18, family='Courier', color='orange'))  
fig.show()
```

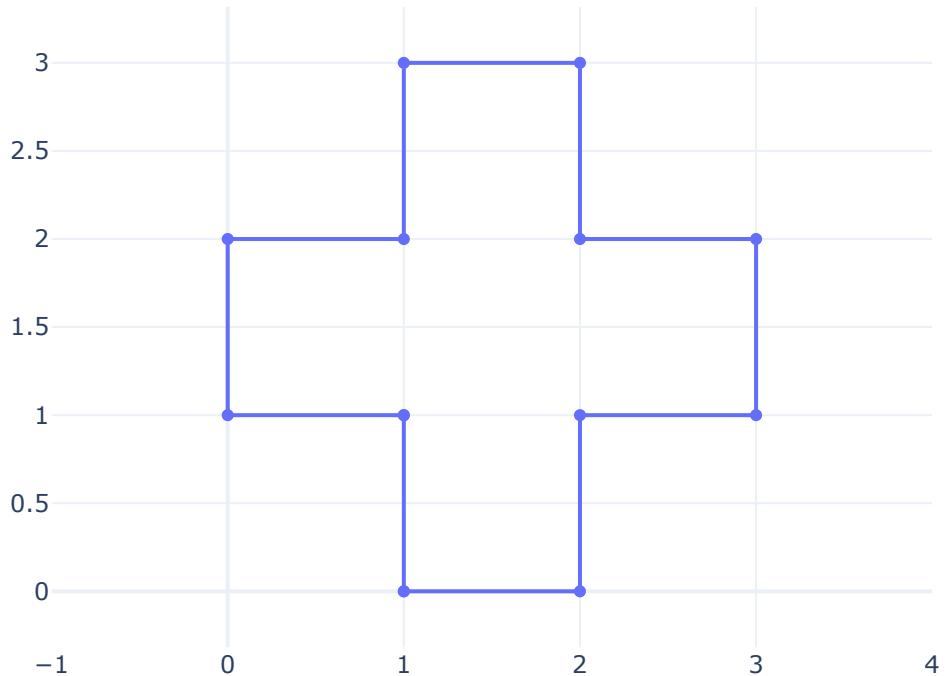


```
fig.update_xaxes(fixedrange=True)
fig.show()
```

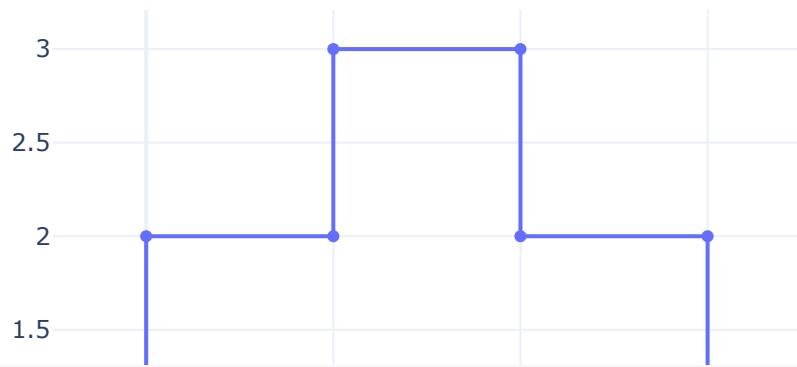


```
fig = go.Figure()
fig.add_trace(go.Scatter(x=[1, 1, 1, 0, 0, 1, 1, 2, 2, 3, 3, 2, 2, 1],
                         y=[0, 1, 1, 1, 2, 2, 3, 3, 2, 2, 1, 1, 0, 0]))
fig.update_layout(width=600, height=500,
                  yaxis=dict(scaleanchor='x', scaleratio=1))
fig.show()
```

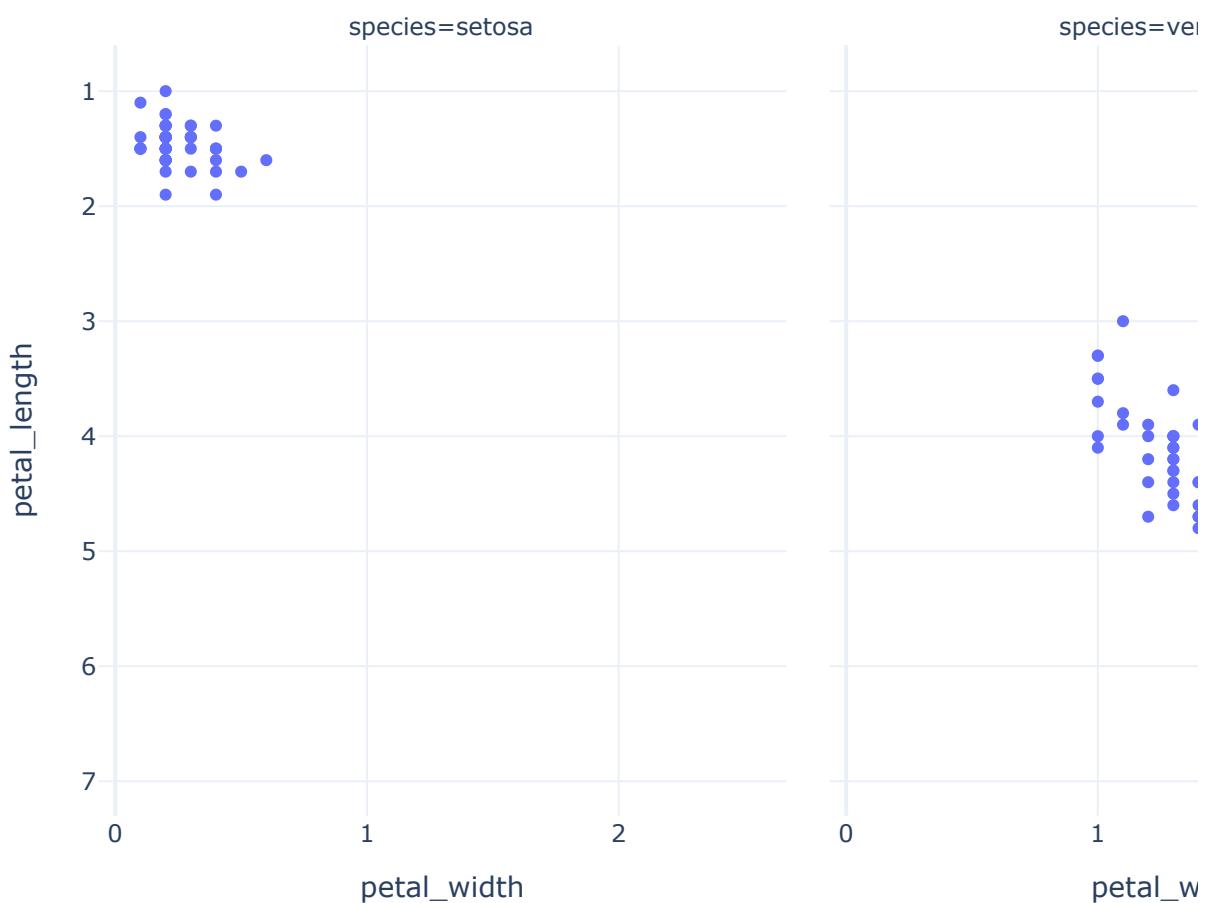
```
fig.update_layout(width=600, height=500,
                  xaxis=dict(range=[-1, 4],
                             constrain='domain'),
                  yaxis=dict(scaleanchor='x', scaleratio=1))
fig.show()
```



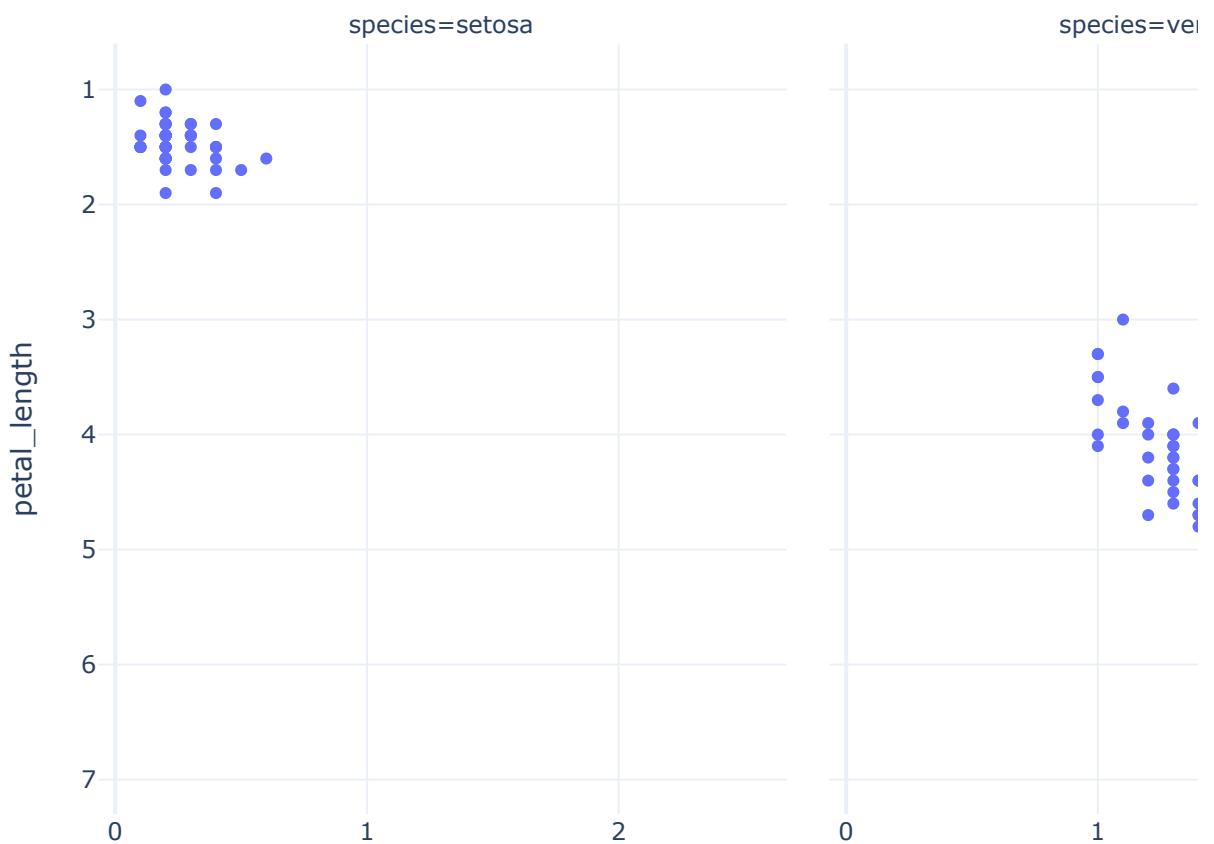
```
fig.update_layout(width=600, height=500,
                  xaxis=dict(range=[-0.5, 3.5],
                             constrain='domain'),
                  yaxis=dict(scaleanchor='x', scaleratio=1))
fig.show()
```



```
fig = px.scatter(iris, x='petal_width', y='petal_length', facet_col='species')
fig.update_yaxes(autorange='reversed')
fig.show()
```



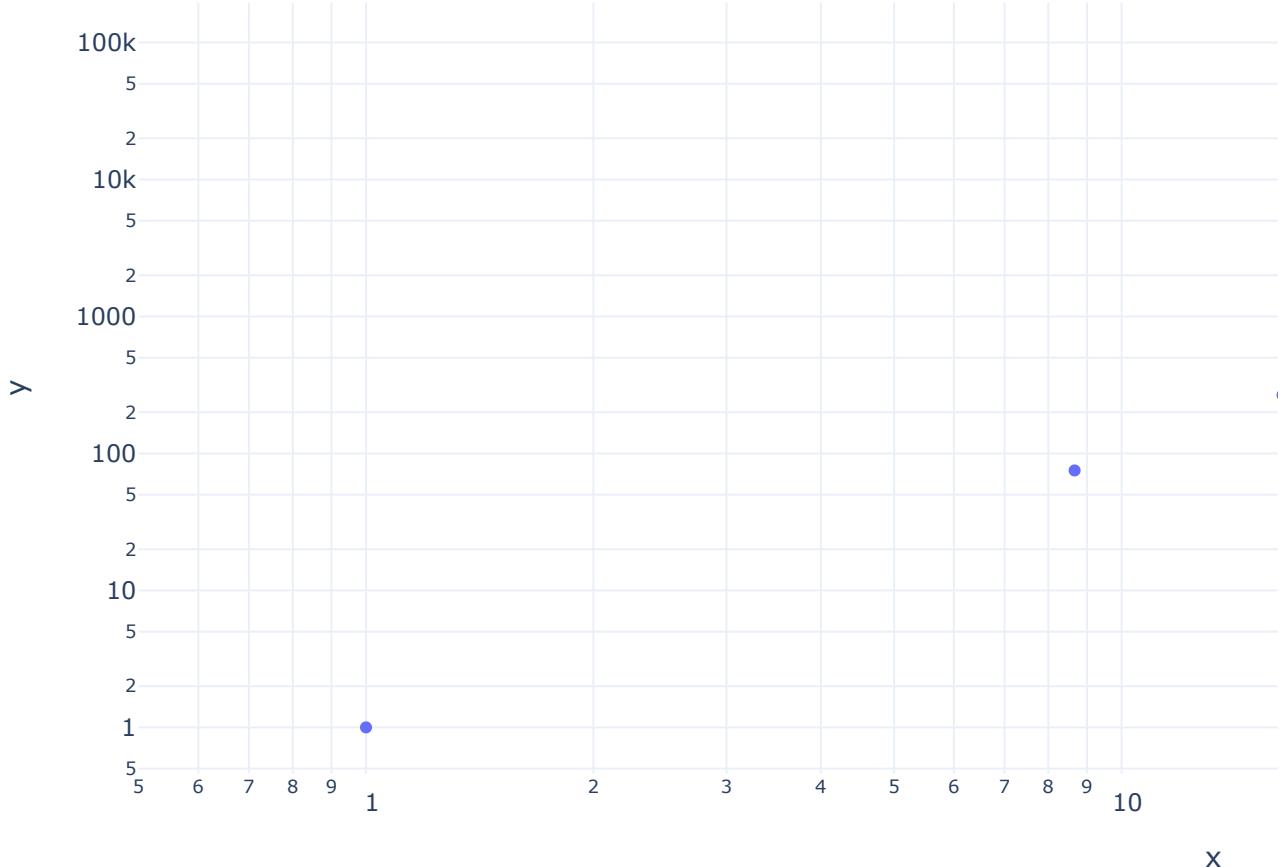
```
fig.update_yaxes(range=[9, 3])
fig.show()
```



```
x = np.linspace(1, 300, 40)

fig = px.scatter(x=x, y=x**2, log_x=True, log_y=True, range_x=[0.5, 350])
fig.show()
```

```
100k  
fig.update_xaxes(type='log', range=[np.log10(0.5), np.log10(350)])  
fig.update_yaxes(type='log')  
fig.show()
```

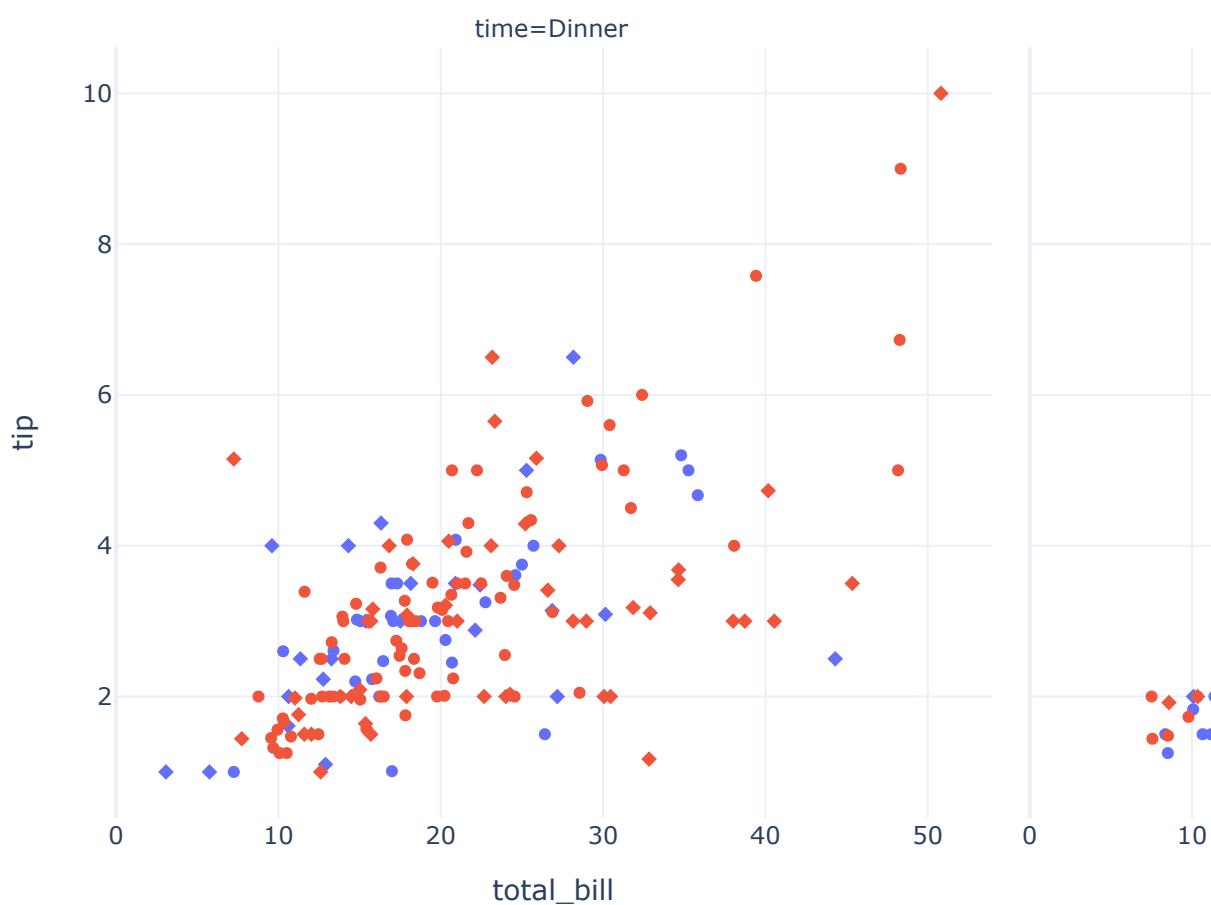


```
fig = px.scatter(iris, x='petal_width', y='petal_length', facet_col='species')  
fig.update_xaxes(rangemode='nonnegative')  
fig.update_yaxes(rangemode='tozero')  
fig.show()
```

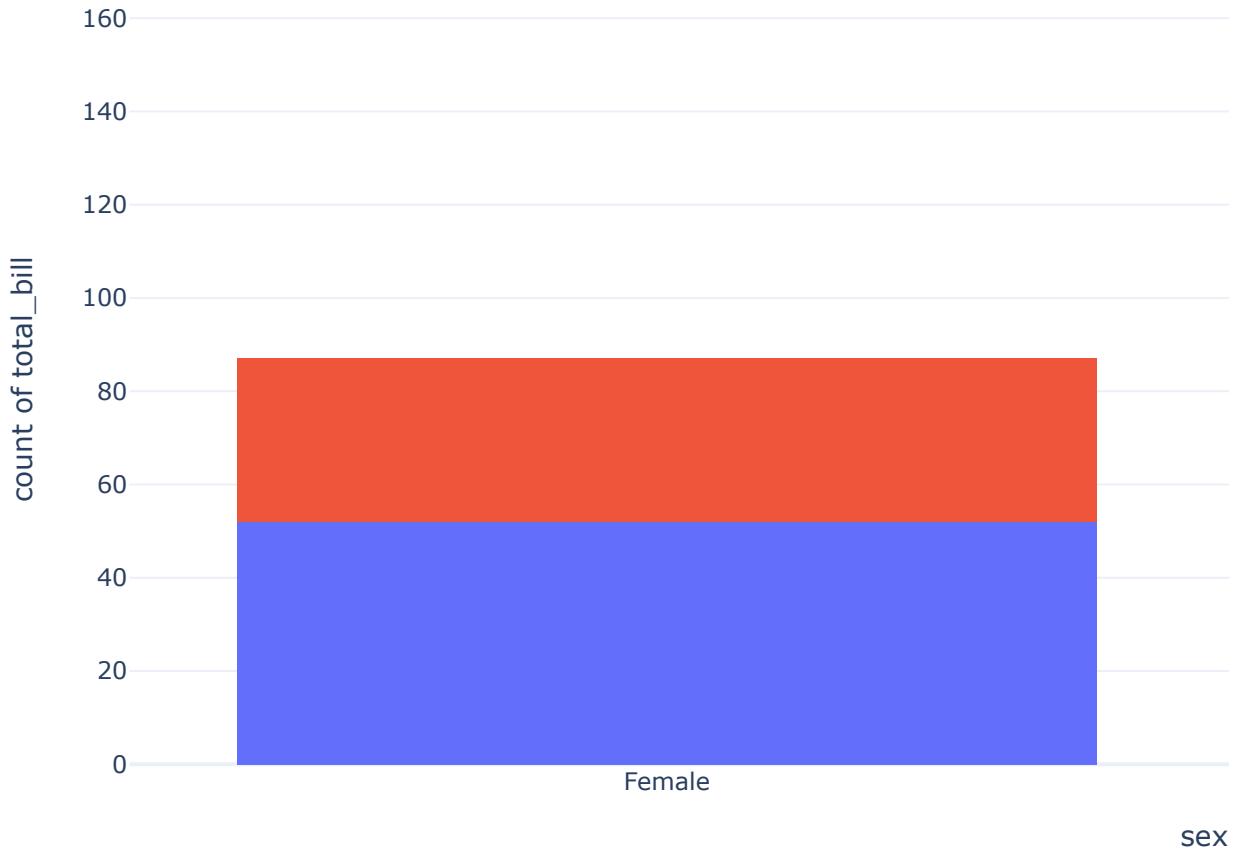


## ▼ 범례(Legends)

```
fig = px.scatter(tips, x='total_bill', y='tip',
                  color='sex', symbol='smoker', facet_col='time',
                  labels={'sex': 'Gender', 'smoker': 'Smokes'})
fig.show()
```

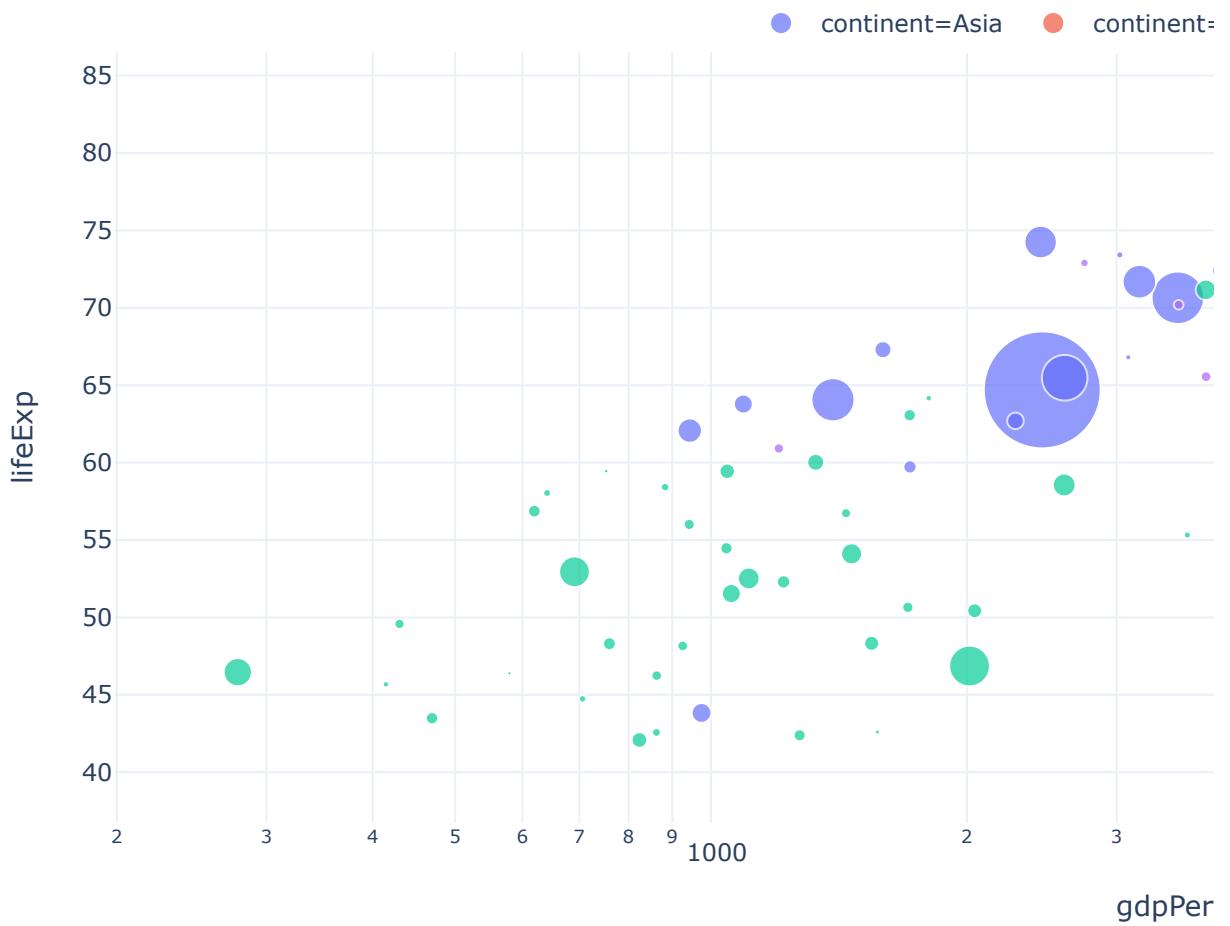


```
fig = px.histogram(tips, x='sex', y='total_bill', color='time')
fig.update_layout(showlegend=False)
fig.show()
```





```
fig.update_layout(legend=dict(orientation='h',
                             yanchor='bottom', y=1,
                             xanchor='right', x=1))
```

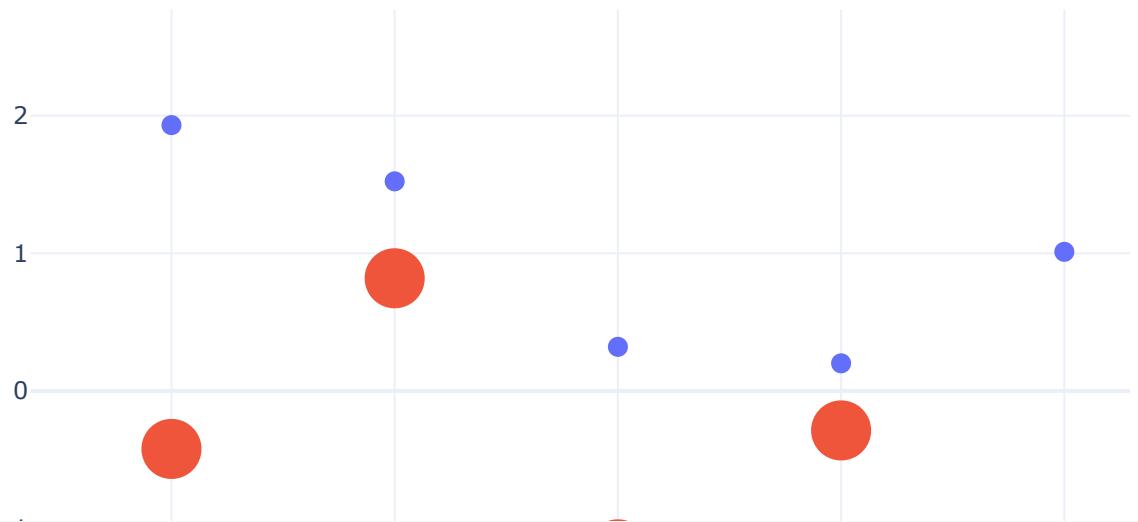


```
fig = go.Figure()
fig.add_trace(go.Scatter(x=np.arange(1, 10),
                         y=np.random.randn(10)+1,
                         #showlegend=False,
                         name="Positive"))
fig.add_trace(go.Scatter(x=np.arange(1, 10),
                         y=np.random.randn(10)-1,
```

```
#visible='legendonly',
name="Negative"))
fig.show()
```



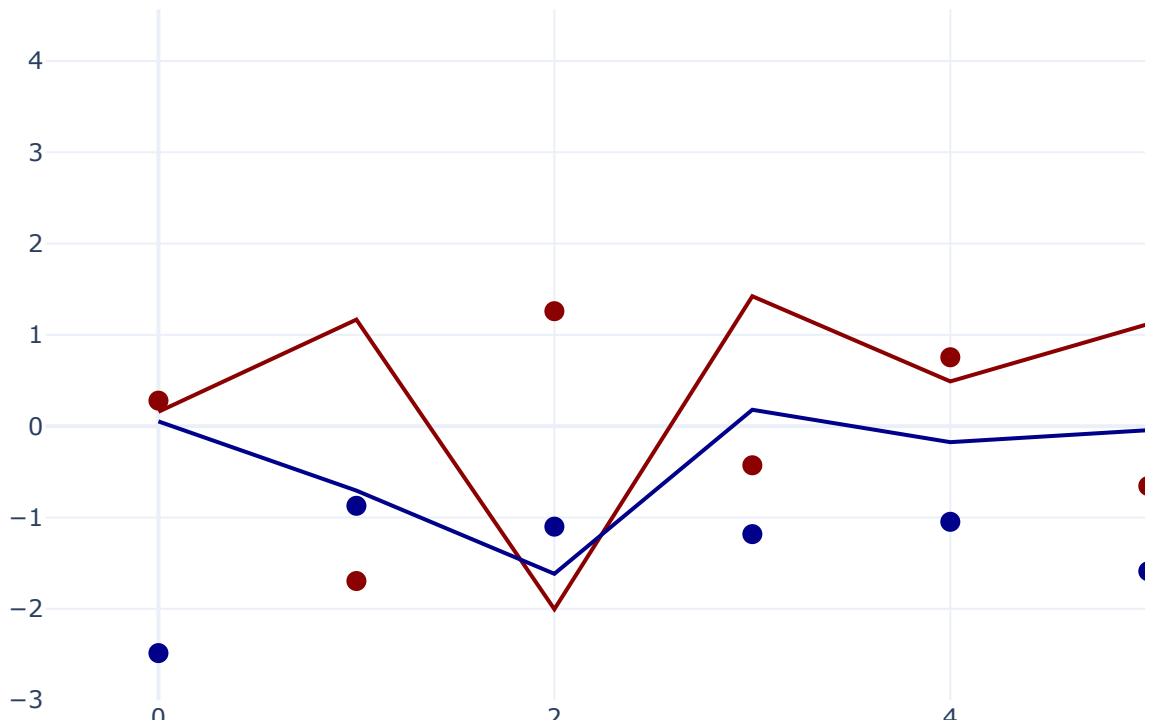
```
fig = go.Figure()
fig.add_trace(go.Scatter(x=np.arange(1, 10),
                         y=np.random.randn(10)+1,
                         mode='markers',
                         marker={'size':10}))
fig.add_trace(go.Scatter(x=np.arange(1, 10),
                         y=np.random.randn(10)-1,
                         mode='markers',
                         marker={'size':30}))
fig.update_layout(legend={'itemSizing': 'constant'})
fig.show()
```



```
x = np.arange(10)
y1 = np.random.randn(10)
y2 = np.random.randn(10)
y3 = np.random.randn(10)
y4 = np.random.randn(10)

fig = go.Figure()
fig.add_trace(go.Scatter(x=x, y=y1,
                        legendgroup='Group A',
                        name='A - markers',
                        mode='markers',
                        marker=dict(color='darkred', size=10)))
fig.add_trace(go.Scatter(x=x, y=y2,
                        legendgroup='Group A',
                        name='A - lines',
                        mode='lines',
                        marker=dict(color='darkred')))
fig.add_trace(go.Scatter(x=x, y=y3,
                        legendgroup='Group B',
                        name='B - markers',
                        mode='markers',
                        marker=dict(color='darkblue', size=10)))
fig.add_trace(go.Scatter(x=x, y=y4,
                        legendgroup='Group B',
                        name='B - lines',
                        mode='lines',
                        marker=dict(color='darkblue')))

fig.update_layout()
fig.show()
```



## ▼ 다중 축(Multiple Axes)

```

x = np.arange(1, 10, 5)
y1 = np.random.randn(5)
y2 = np.random.randn(5).cumsum()

fig = make_subplots(rows=2, cols=2,
                     specs=[[{'secondary_y':True}, {'secondary_y':True}],
                            [{"secondary_y":True}, {"secondary_y":True}]])

fig.add_trace(go.Scatter(x=x, y=y1, name='yaxis1 data'),
              row=1, col=1, secondary_y=False)
fig.add_trace(go.Scatter(x=x, y=y2, name='yaxis2 data'),
              row=1, col=1, secondary_y=True)

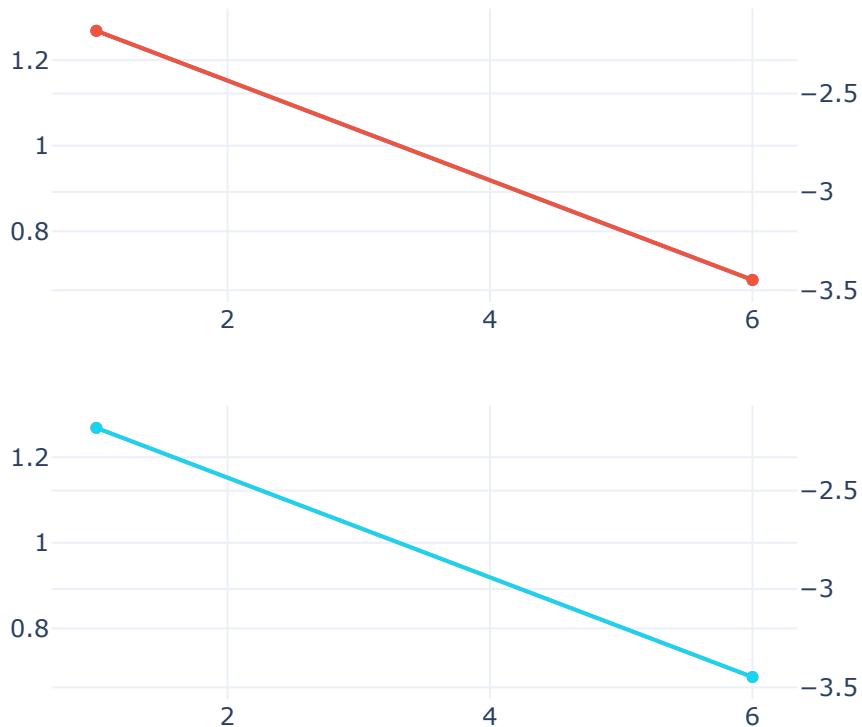
fig.add_trace(go.Scatter(x=x, y=y1, name='yaxis3 data'),
              row=1, col=2, secondary_y=False)
fig.add_trace(go.Scatter(x=x, y=y2, name='yaxis4 data'),
              row=1, col=2, secondary_y=True)

fig.add_trace(go.Scatter(x=x, y=y1, name='yaxis5 data'),
              row=2, col=1, secondary_y=False)
fig.add_trace(go.Scatter(x=x, y=y2, name='yaxis6 data'),
              row=2, col=1, secondary_y=True)

fig.add_trace(go.Scatter(x=x, y=y1, name='yaxis7 data'),
              row=2, col=2, secondary_y=False)
fig.add_trace(go.Scatter(x=x, y=y2, name='yaxis8 data'),
              row=2, col=2, secondary_y=True)

```

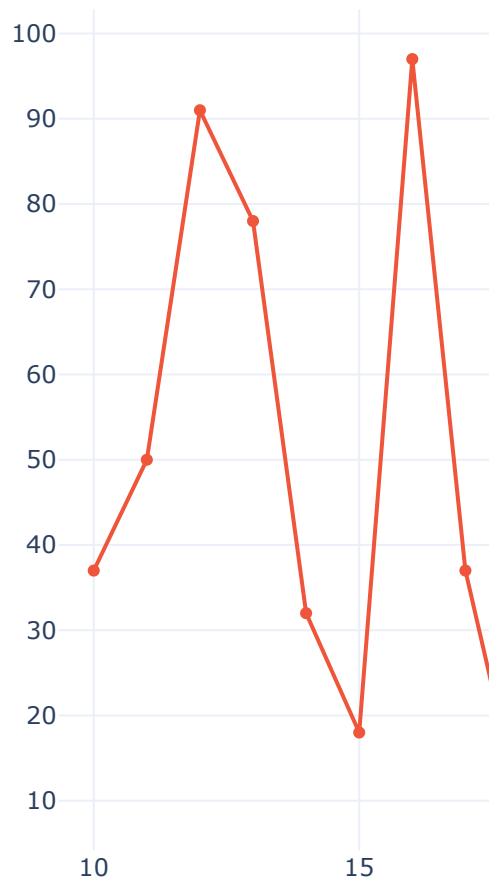
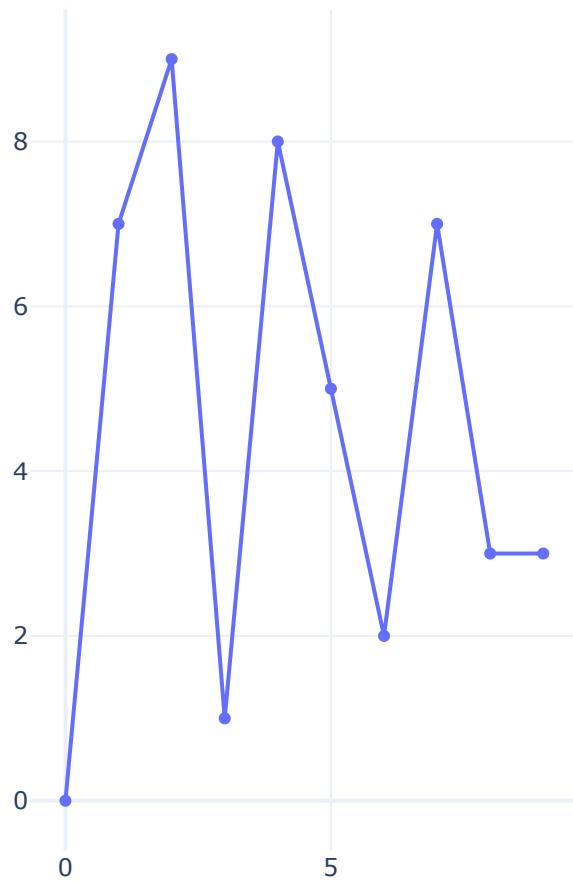
```
    row=2, col=2, secondary_y=True)
fig.show()
```



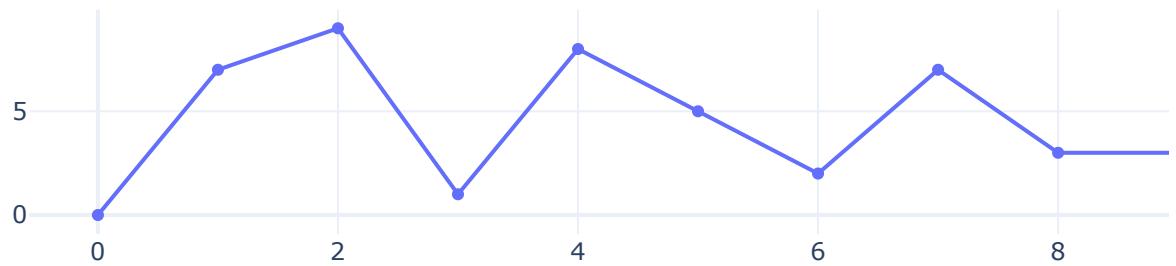
## ▼ 서브플롯(Subplots)

```
x1 = np.arange(0, 11)
y1 = np.random.randint(0, 10, 10)
x2 = np.arange(10, 21)
y2 = np.random.randint(10, 100, 10)
x3 = np.arange(20, 31)
y3 = np.random.randint(100, 1000, 10)
x4 = np.arange(30, 41)
y4 = np.random.randint(1000, 10000, 10)
```

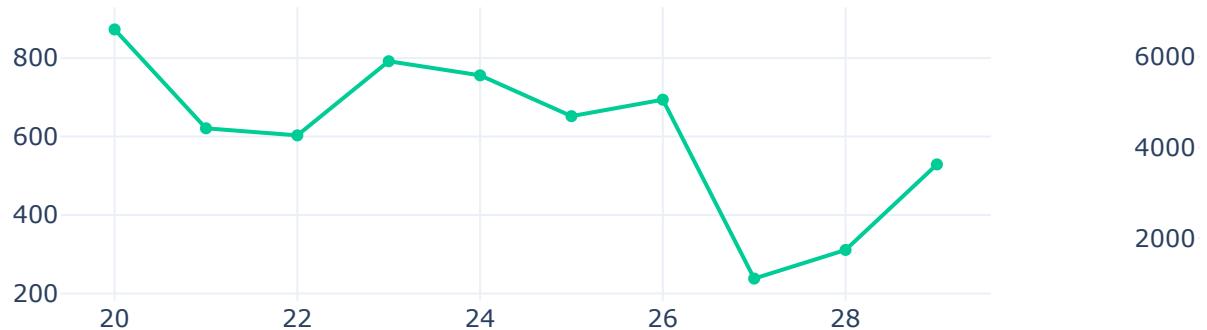
```
fig = make_subplots(rows=1, cols=2)
fig.add_trace(go.Scatter(x=x1, y=y1), row=1, col=1)
fig.add_trace(go.Scatter(x=x2, y=y2), row=1, col=2)
fig.update_layout(height=600, width=800)
fig.show()
```



```
fig = make_subplots(rows=3, cols=1)
fig.add_trace(go.Scatter(x=x1, y=y1), row=1, col=1)
fig.add_trace(go.Scatter(x=x2, y=y2), row=2, col=1)
fig.add_trace(go.Scatter(x=x3, y=y3), row=3, col=1)
fig.update_layout(height=600, width=800)
fig.show()
```



```
fig = make_subplots(rows=2, cols=2, start_cell='bottom-left')
fig.add_trace(go.Scatter(x=x1, y=y1), row=1, col=1)
fig.add_trace(go.Scatter(x=x2, y=y2), row=1, col=2)
fig.add_trace(go.Scatter(x=x3, y=y3), row=2, col=1)
fig.add_trace(go.Scatter(x=x4, y=y4), row=2, col=2)
fig.show()
```

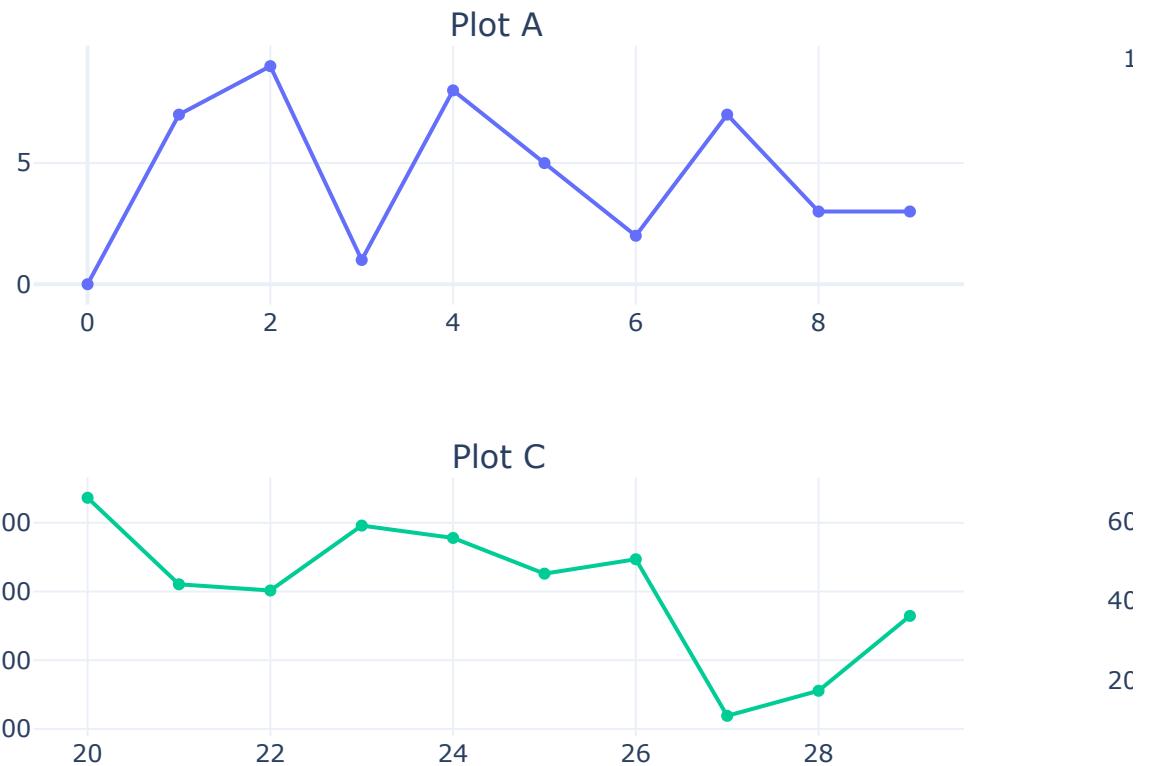


```
fig = make_subplots(rows=2, cols=2,
                     subplot_titles=('Plot A', 'Plot B', 'Plot C', 'Plot D'))
fig.add_trace(go.Scatter(x=x1, y=y1), row=1, col=1)
```

```

fig.add_trace(go.Scatter(x=x2, y=y2), row=1, col=2)
fig.add_trace(go.Scatter(x=x3, y=y3), row=2, col=1)
fig.add_trace(go.Scatter(x=x4, y=y4), row=2, col=2)
fig.show()

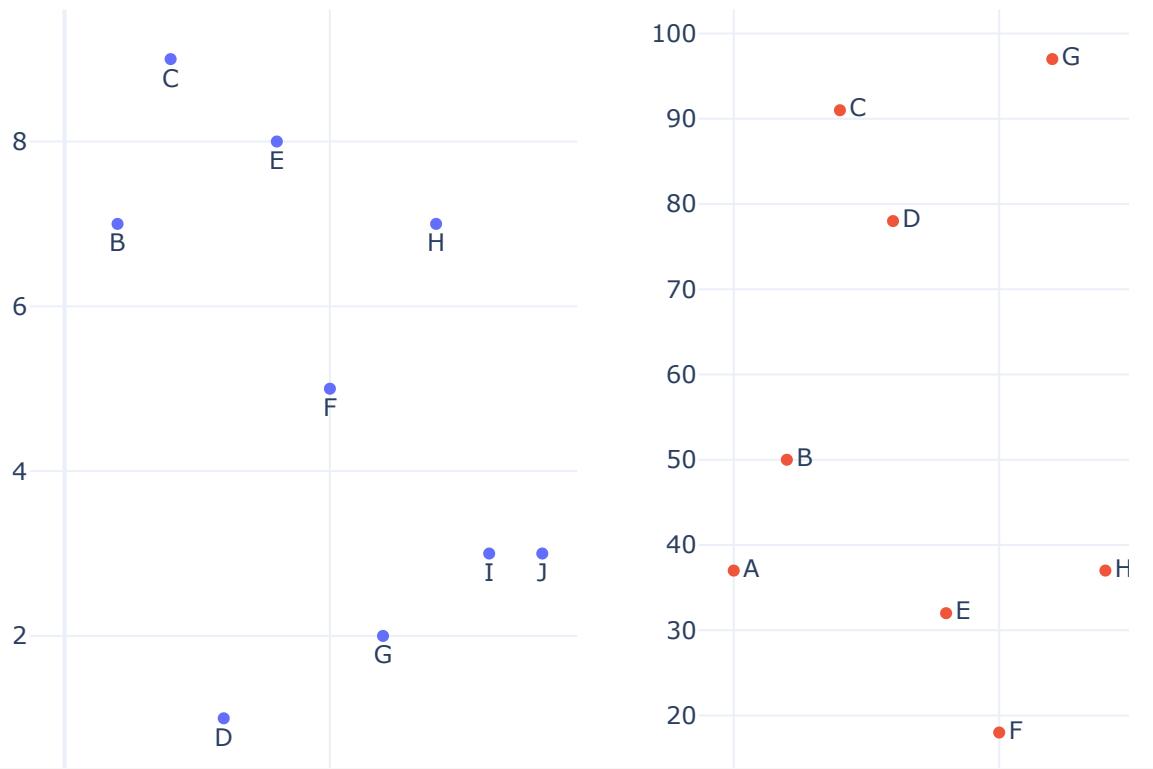
```



```

fig = make_subplots(rows=1, cols=2)
fig.add_trace(go.Scatter(x=x1, y=y1,
                        mode='markers+text',
                        text=['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J'],
                        textposition='bottom center'),
                        row=1, col=1)
fig.add_trace(go.Scatter(x=x2, y=y2,
                        mode='markers+text',
                        text=['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J'],
                        textposition='middle right'),
                        row=1, col=2)
fig.update_layout(height=600, width=800)
fig.show()

```

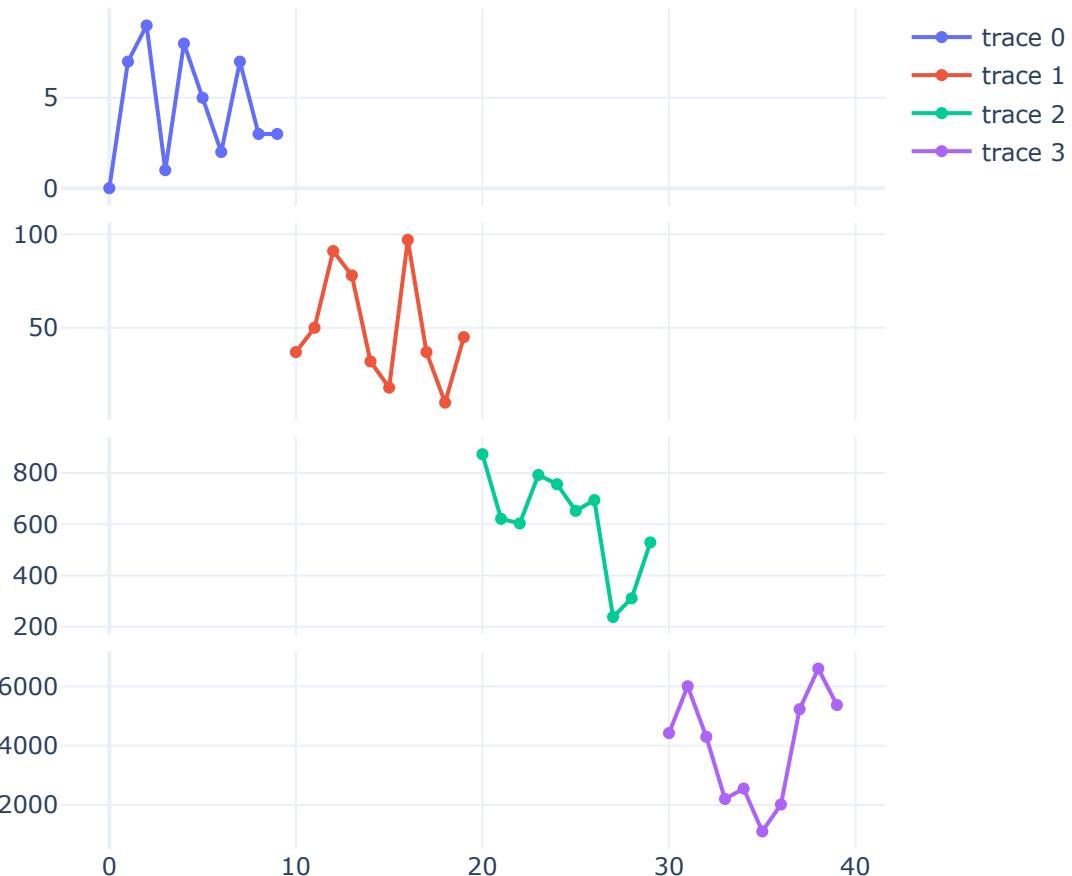


```
fig = make_subplots(rows=1, cols=2, column_width=[0.7, 0.3])
fig.add_trace(go.Scatter(x=x3, y=y3), row=1, col=1)
fig.add_trace(go.Scatter(x=x4, y=y4), row=1, col=2)
fig.show()
```

```
fig = make_subplots(rows=2, cols=2,
                     subplot_titles=['Plot A', 'Plot B', 'Plot C', 'Plot D'])
fig.add_trace(go.Scatter(x=x1, y=y1), row=1, col=1)
fig.add_trace(go.Scatter(x=x2, y=y2), row=1, col=2)
fig.add_trace(go.Scatter(x=x3, y=y3), row=2, col=1)
fig.add_trace(go.Scatter(x=x4, y=y4), row=2, col=2)
fig.update_xaxes(title_text='xaxes 1 title', row=1, col=1)
fig.update_xaxes(title_text='xaxes 2 title', range=[10, 50], row=1, col=2)
fig.update_xaxes(title_text='xaxes 3 title', showgrid=False, row=2, col=1)
fig.update_xaxes(title_text='xaxes 4 title', type='log', row=2, col=2)
fig.update_yaxes(title_text='yaxes 1 title', row=1, col=1)
fig.update_yaxes(title_text='yaxes 2 title', range=[10, 50], row=1, col=2)
fig.update_yaxes(title_text='yaxes 3 title', showgrid=False, row=2, col=1)
fig.update_yaxes(title_text='yaxes 4 title', row=2, col=2)
fig.update_layout(height=700)
fig.show()
```

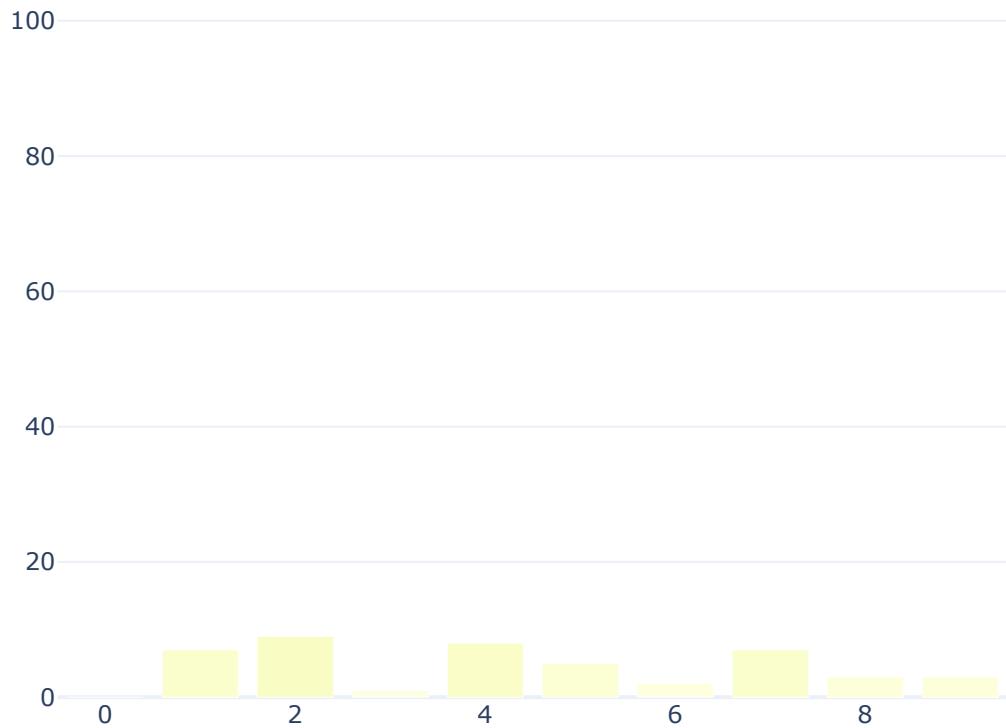
### Plot Δ

```
fig = make_subplots(rows=4, cols=1,
                     shared_xaxes=True,
                     vertical_spacing=0.02)
fig.add_trace(go.Scatter(x=x1, y=y1), row=1, col=1)
fig.add_trace(go.Scatter(x=x2, y=y2), row=2, col=1)
fig.add_trace(go.Scatter(x=x3, y=y3), row=3, col=1)
fig.add_trace(go.Scatter(x=x4, y=y4), row=4, col=1)
fig.update_layout(height=600, width=600)
fig.show()
```



```
fig = make_subplots(rows=1, cols=2, shared_yaxes=True)
fig.add_trace(go.Bar(x=x1, y=y1,
                     marker=dict(color=y1, coloraxis='coloraxis')), 1, 1)
fig.add_trace(go.Bar(x=x2, y=y2,
                     marker=dict(color=y2, coloraxis='coloraxis')), 1, 2)
fig.update_layout(coloraxis=dict(colorscale='YIGn'), showlegend=False)
```

```
fig.show()
```



```
fig = make_subplots(rows=2, cols=2, specs=[[{}, {}], [{"colspan":2}, None]],
                     subplot_titles=('Subplot A', 'Subplot B', 'Subplot C'))
fig.add_trace(go.Scatter(x=x1, y=y1), row=1, col=1)
fig.add_trace(go.Scatter(x=x2, y=y2), row=1, col=2)
fig.add_trace(go.Scatter(x=x3, y=y3), row=2, col=1)
fig.update_layout(showlegend=False)
fig.show()
```

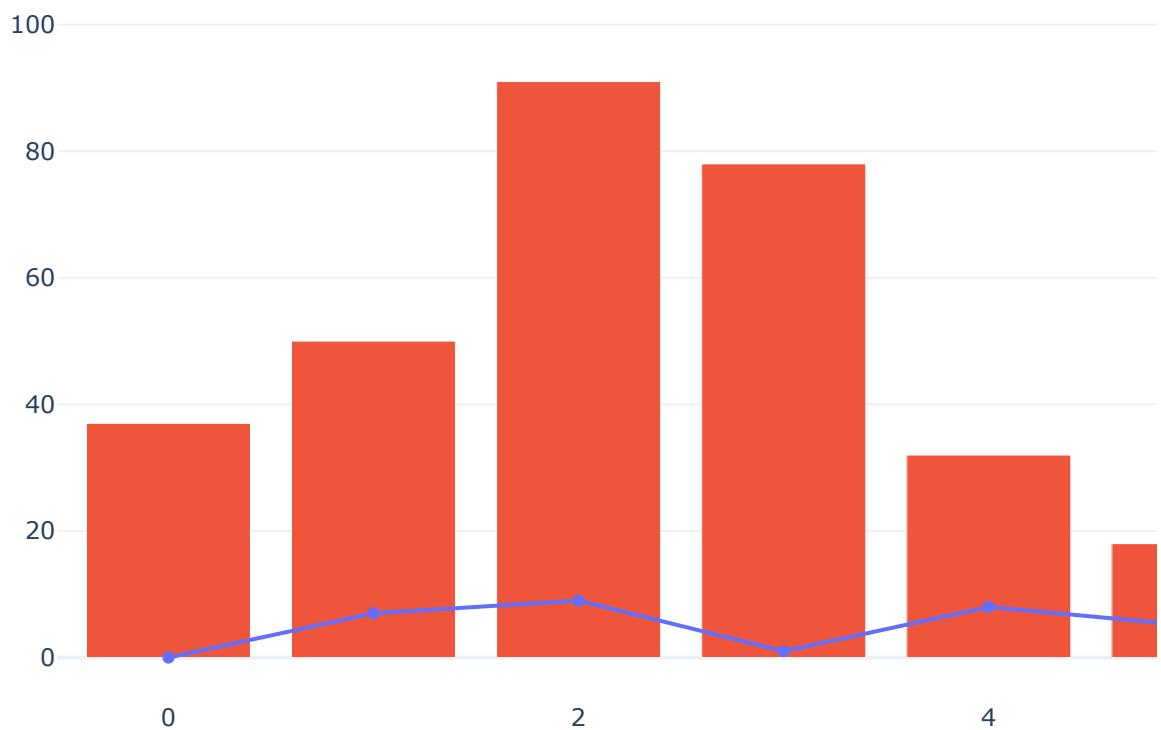
**Subplot A**



```
fig = make_subplots(rows=2, cols=2,
                     specs=[[{'type':'xy'}, {'type':'polar'}],
                            [{"type": "domain"}, {"type": "scene"}]])
fig.add_trace(go.Bar(y=y1), row=1, col=1)
fig.add_trace(go.Barpolar(theta=[0, 60, 120], r=y2), row=1, col=2)
fig.add_trace(go.Pie(values=y3), row=2, col=1)
fig.add_trace(go.Scatter3d(x=x4, y=y4, z=[0.5, 1, 2], mode='lines'), row=2, col=2)
fig.update_layout(height=700, showlegend=False)
fig.show()
```

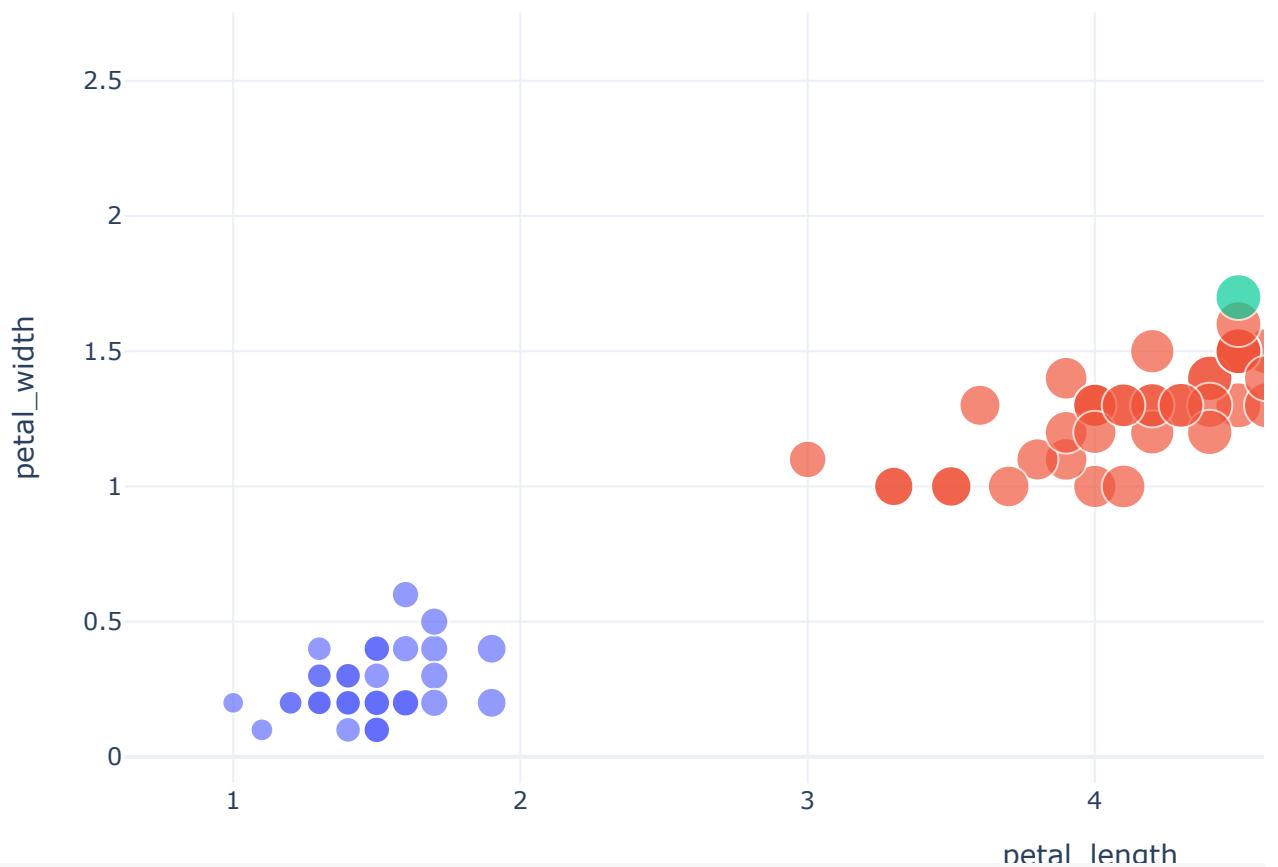
## ▼ 다중 차트 유형(Multiple Chart Types)

```
fig = go.Figure()
fig.add_trace(go.Scatter(x=x1, y=y1))
fig.add_trace(go.Bar(x=x1, y=y2))
fig.show()
```



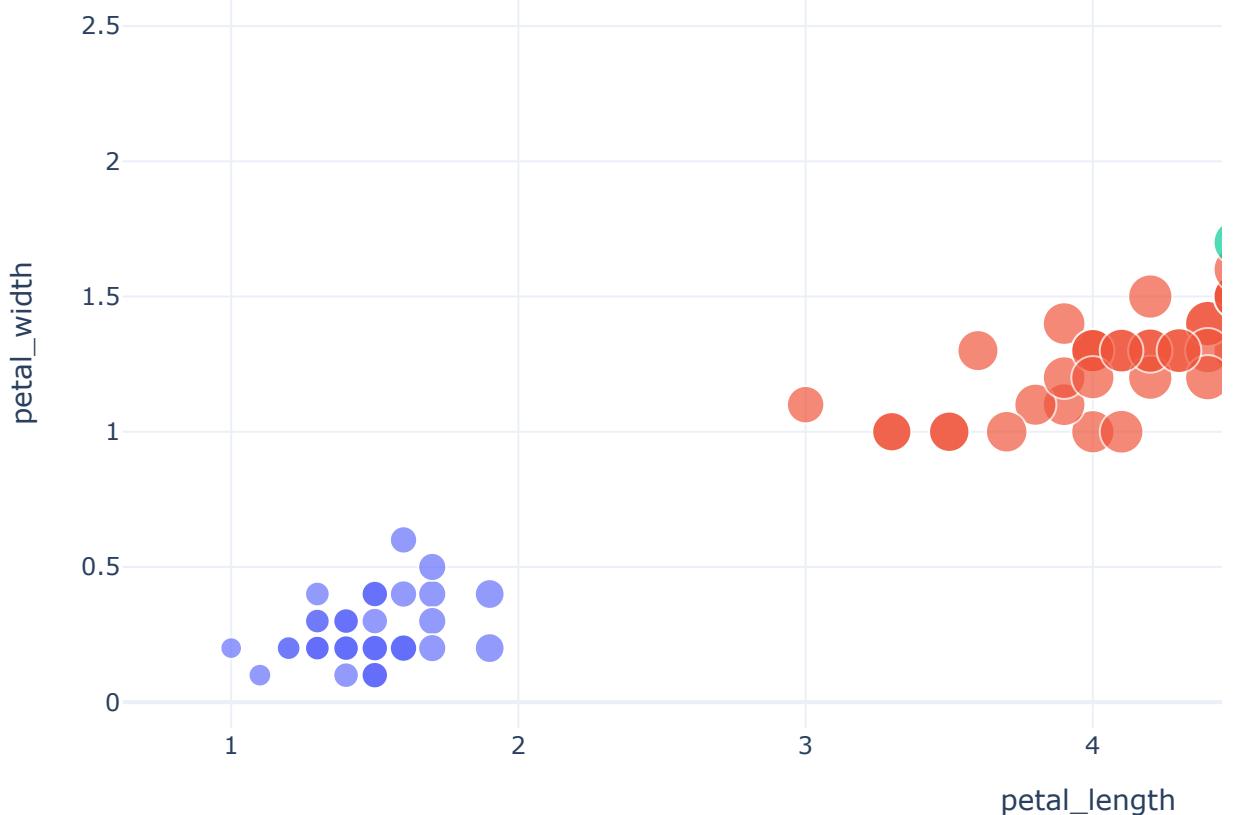
## ▼ 인수(Arguments)

```
fig = px.scatter(iris, x=iris.petal_length, y=iris.petal_width,
                 color=iris.species, size=iris.petal_length)
fig.show()
```

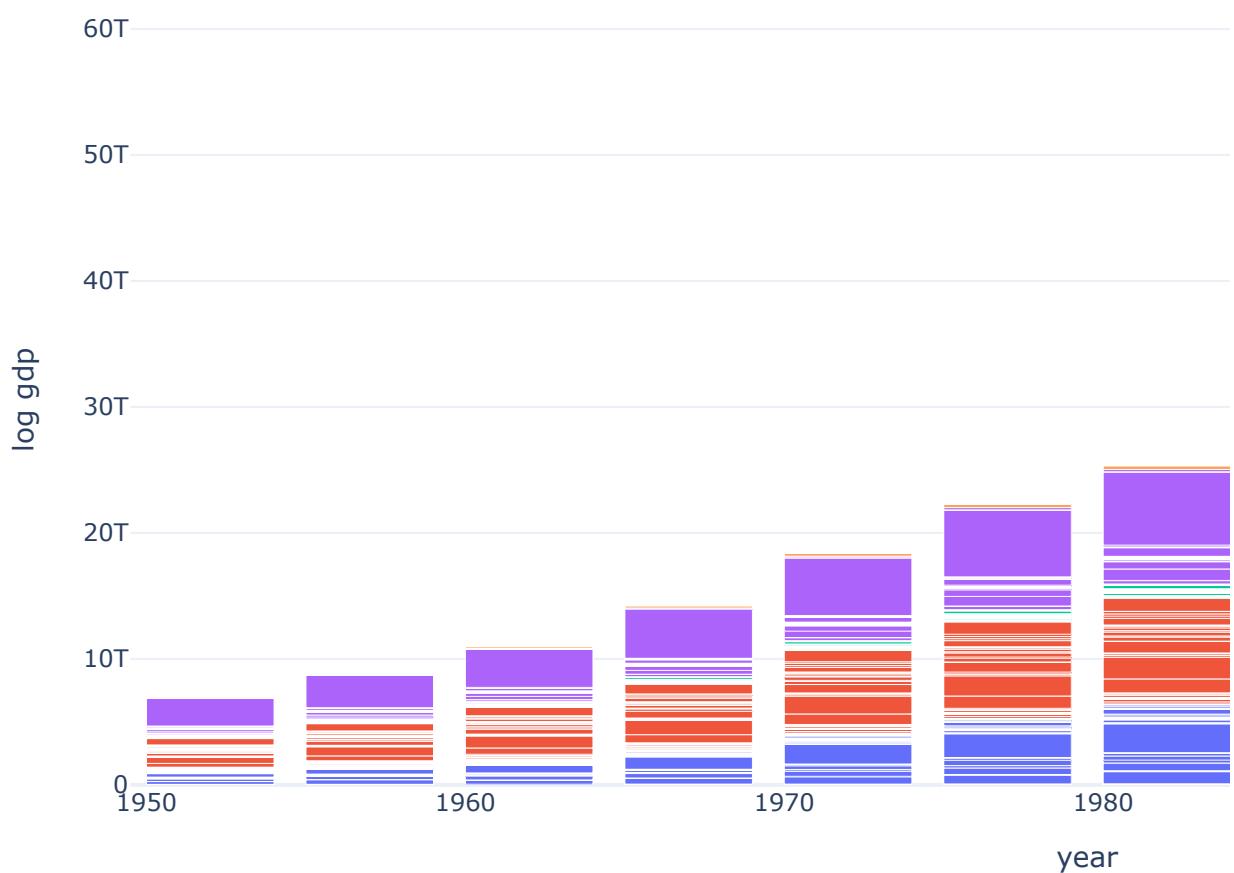


```
fig = px.scatter(iris, x='petal_length', y='petal_width',
                  color='species', size='petal_length')
fig.show()
```

```
fig = px.scatter(iris, x=iris.petal_length, y=iris.petal_width,  
                 color=iris.species, size=iris.petal_length,  
                 hover_data=[iris.index])  
fig.show()
```

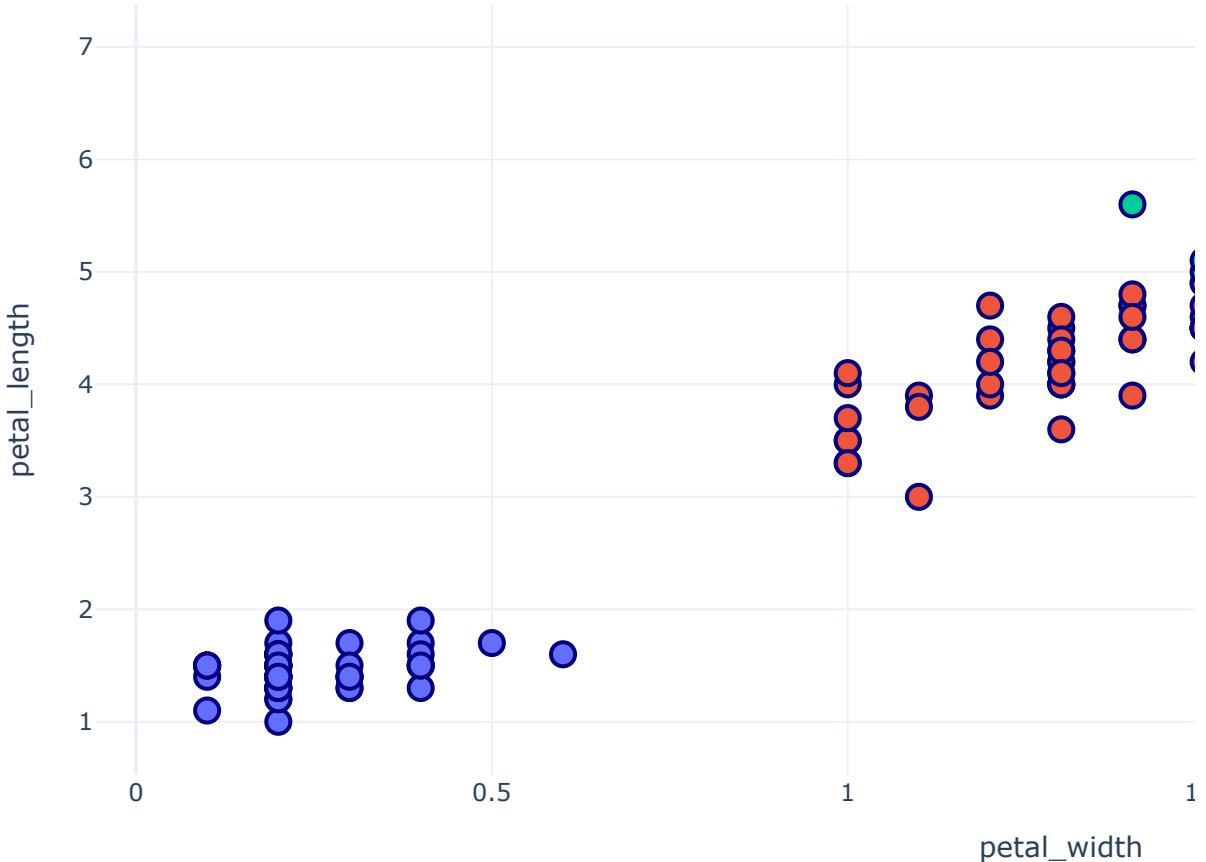


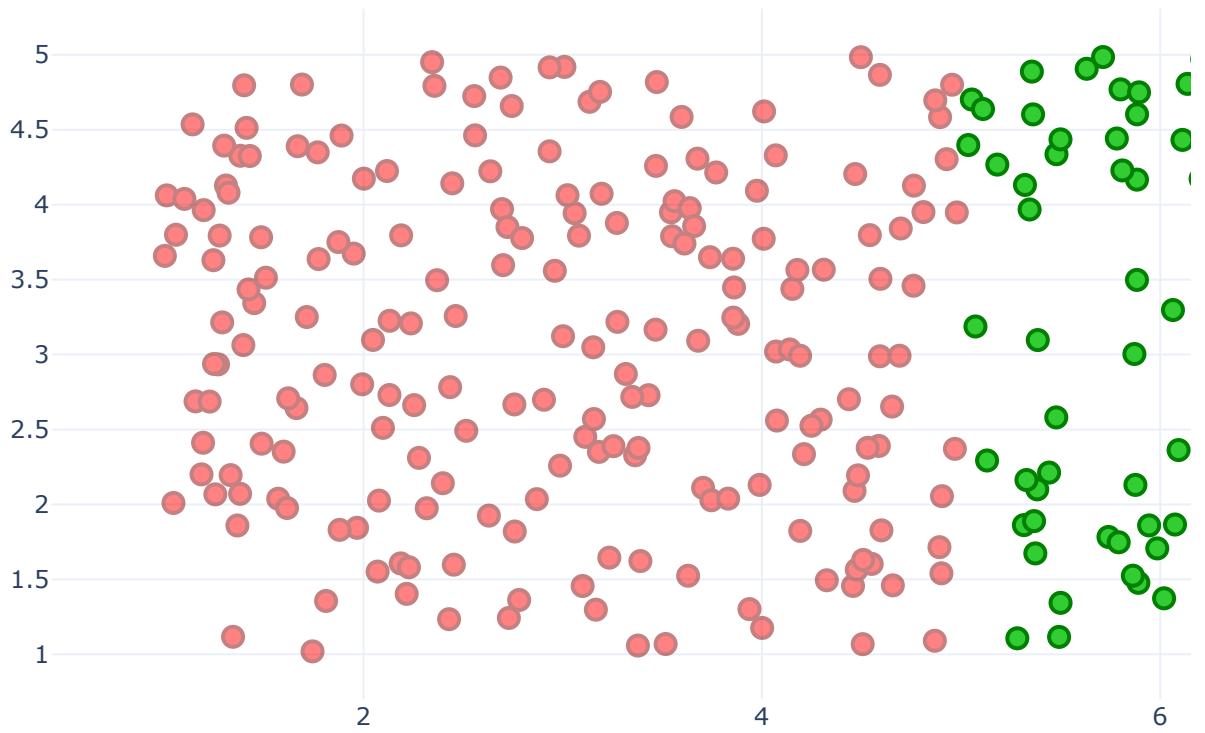
```
gdp = gapminder['pop'] * gapminder['gdpPerCap']  
fig = px.bar(gapminder, x='year', y=gdp, color='continent', labels={'y':'gdp'},  
             hover_data=['country'])  
fig.show()
```



## ▼ 마커 스타일(Styling Markers)

```
fig = px.scatter(iris, x='petal_width', y='petal_length', color='species')
fig.update_traces(marker=dict(size=12, line=dict(width=2, color='navy')),
                  selector=dict(mode='markers'))
fig.show()
```





```

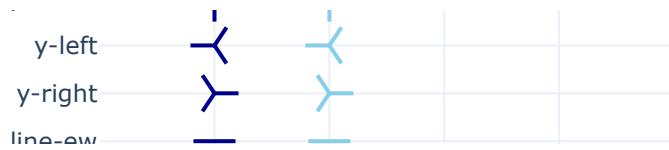
raw_symbols = SymbolValidator().values
namestems = []
namevariants = []
symbols = []

for i in range(0, len(raw_symbols), 2):
    name = raw_symbols[i+1]
    symbols.append(raw_symbols[i])
    namestems.append(name.replace("-open", "").replace("-dot", ""))
    namevariants.append(name[len(namestems[-1]):])

fig = go.Figure(go.Scatter(mode='markers', x=namevariants, y=namestems, marker_symbol=symbols,
                           marker_line_color='darkblue', marker_color='skyblue',
                           marker_line_width=2, marker_size=15,
                           hovertemplate="%{y}%{x}(num: %{marker.symbol}))")
fig.update_layout(xaxis_range=[-1, 4], yaxis_range=[len(set(namestems)), -1],
                  margin=dict(b=0, r=0), xaxis_side='top', height=1200, width=400)
fig.show()

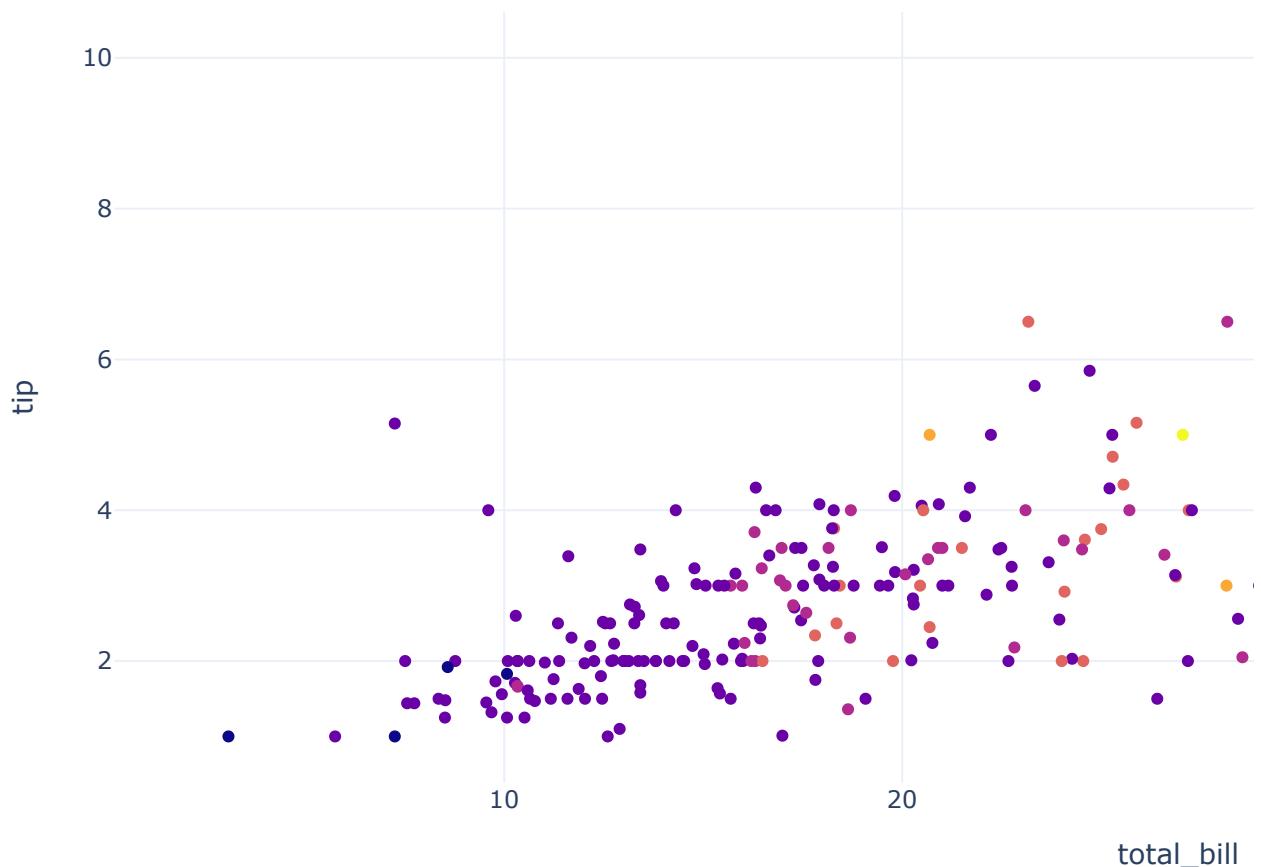
```

	-open	-dot	-open-dot
circle	●	○	○
square	■	□	□
diamond	◇	◆	◆
cross	+■	+□	+□
x	✗■	✗□	✗□
triangle-up	▲	▼	▼
triangle-down	▼	▲	▼
triangle-left	◀	▶	▶
triangle-right	▶	◀	◀
triangle-ne	◀▲	◀▼	◀▼
triangle-se	◀▼	◀▲	◀▼
triangle-sw	◀▼	◀▲	◀▼
triangle-nw	◀▲	◀▼	◀▼
pentagon	⬟	⬢	⬢
hexagon	⬡	⬢	⬢
hexagon2	⬢	⬡	⬡
octagon	⬢	⬢	⬢
star	★	★	★
hexagram	★★	★★	★★
star-triangle-up	▲★	▼★	▼★
star-triangle-down	▼★	▲★	▼★
star-square	▣★	▣★	▣★
star-diamond	◆★	◆★	◆★
diamond-tall	◆	◆	◆
diamond-wide	◆	◆	◆
hourglass	◐	◑	◑
bowtie	◐	◑	◑
circle-cross	⊕	⊕	⊕
circle-x	⊗	⊗	⊗
square-cross	▣	▣	▣
square-x	⊗	⊗	⊗
diamond-cross	◆	◆	◆
diamond-x	◆	◆	◆
cross-thin	+	+	+
x-thin	✗	✗	✗
asterisk	*	*	*
hash	#	#	#
y-up	⤒	⤒	⤒
y-down	⤓	⤓	⤓



## ▼ 색상 척도 및 색상 막대

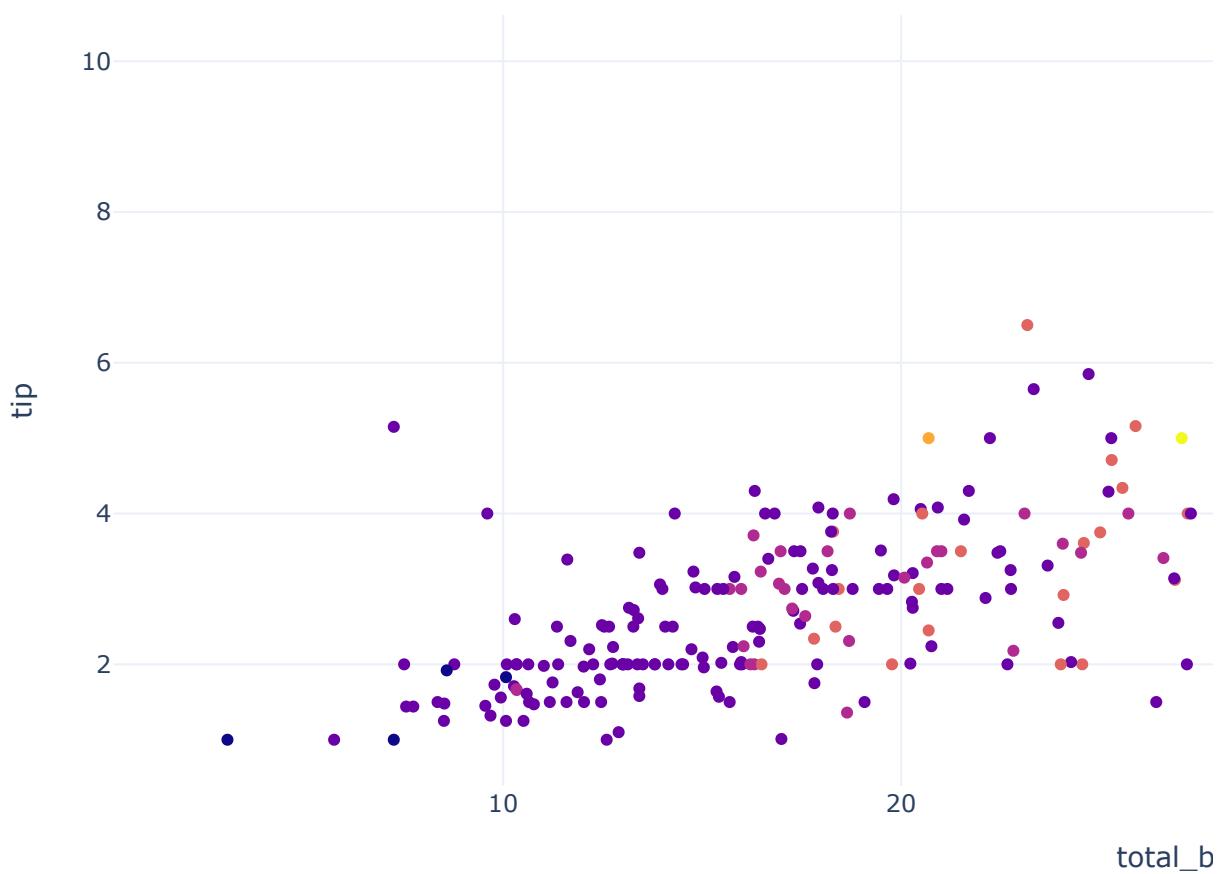
```
fig = px.scatter(tips, x='total_bill', y='tip', color='size')
fig.show()
```



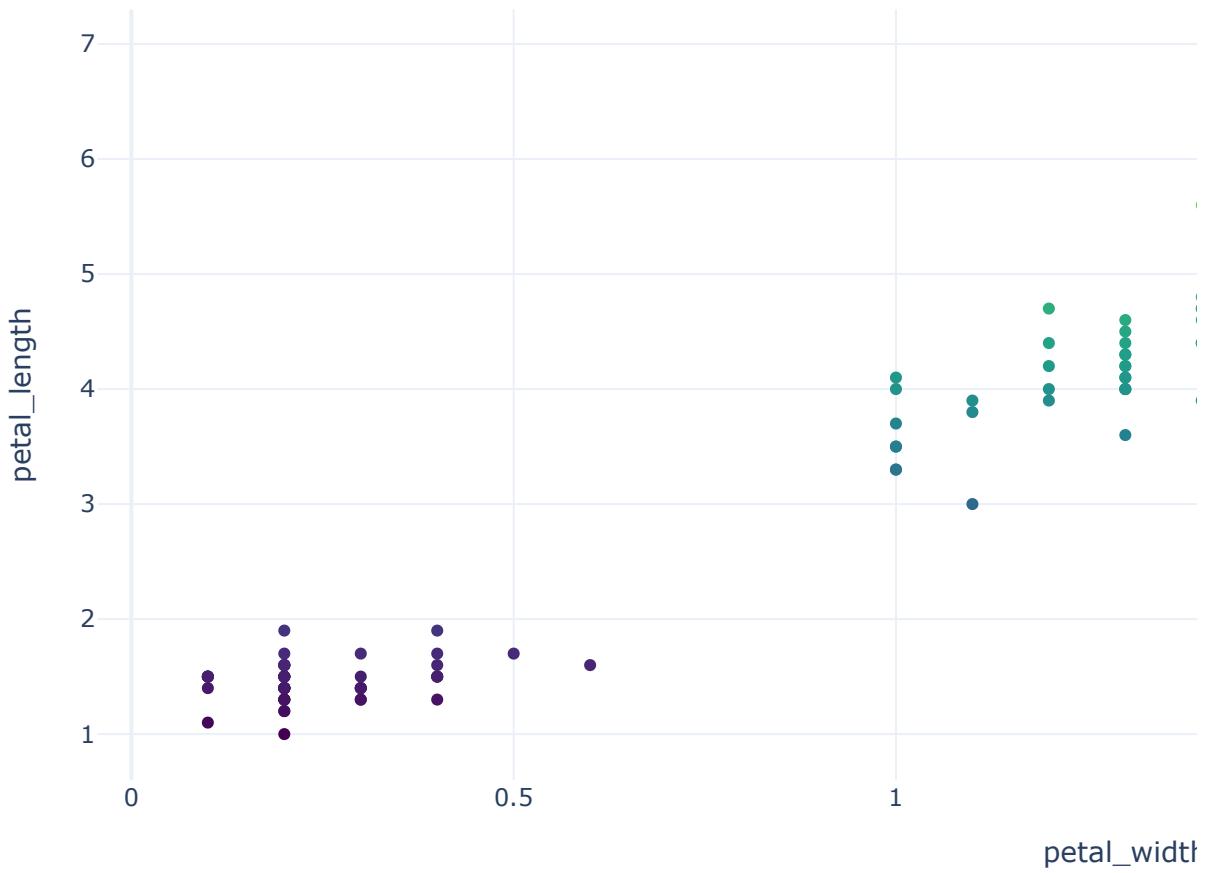
```
fig = px.scatter(tips, x='total_bill', y='tip', color=tips['size'].astype(str))
fig.show()
```



```
fig = px.scatter(tips, x='total_bill', y='tip', color=tips['size'].astype(float))
fig.show()
```

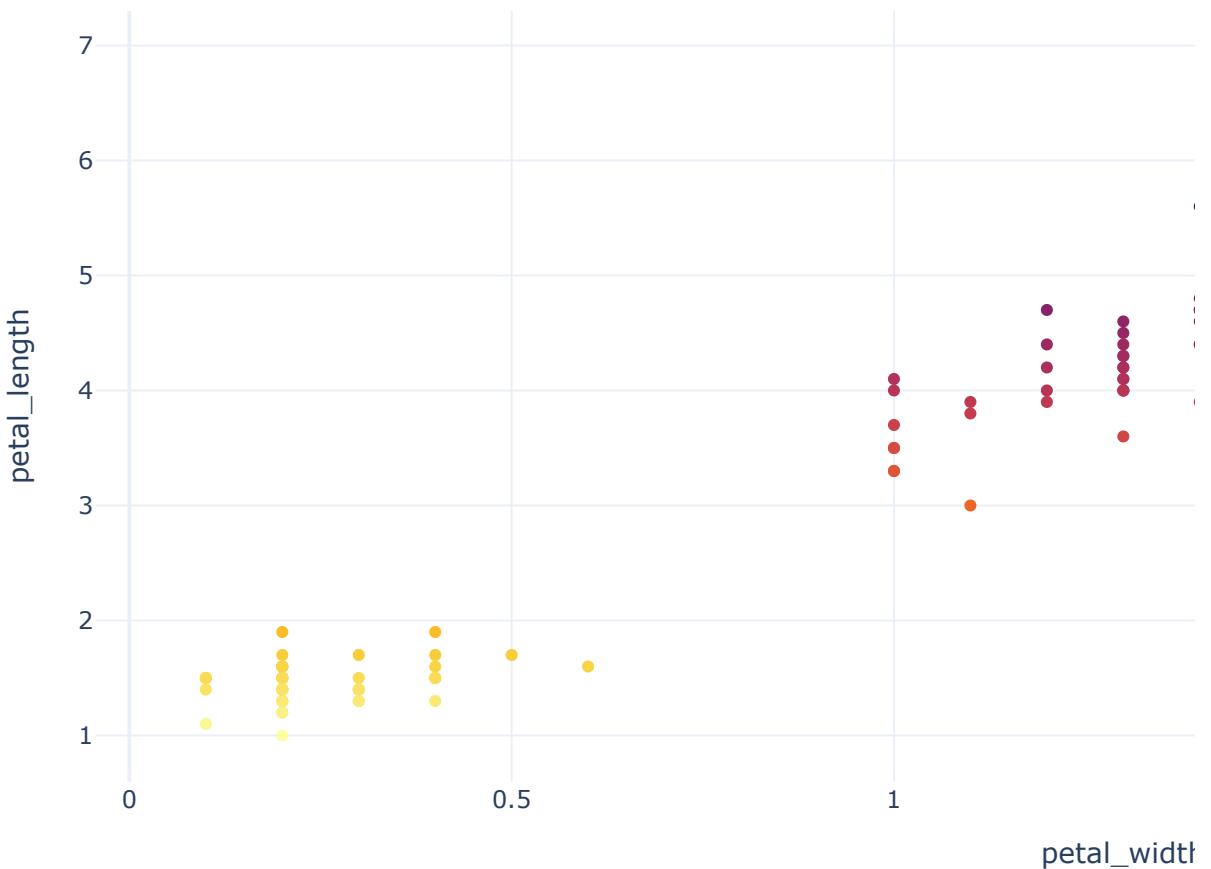


```
fig = px.scatter(iris, x='petal_width', y='petal_length',
                 color='petal_length',
                 color_continuous_scale=px.colors.sequential.Viridis)
fig.show()
```

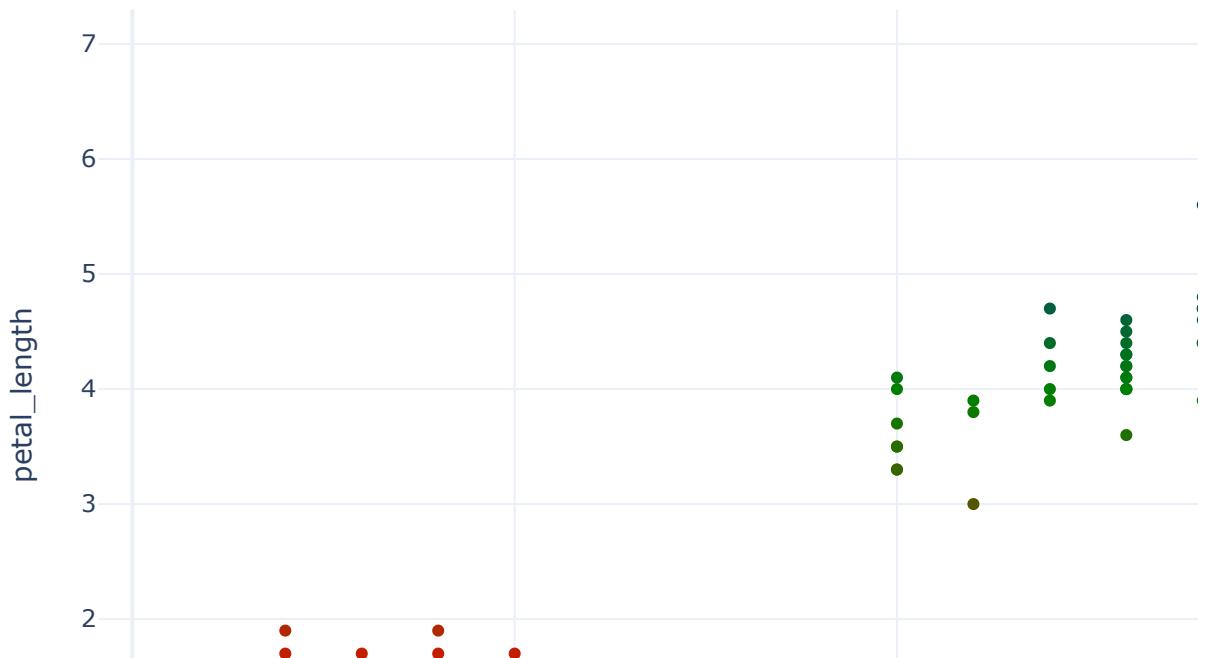


```
fig = px.scatter(iris, x='petal_width', y='petal_length',
                 color='petal_length',
                 color_continuous_scale='Inferno')
fig.show()
```

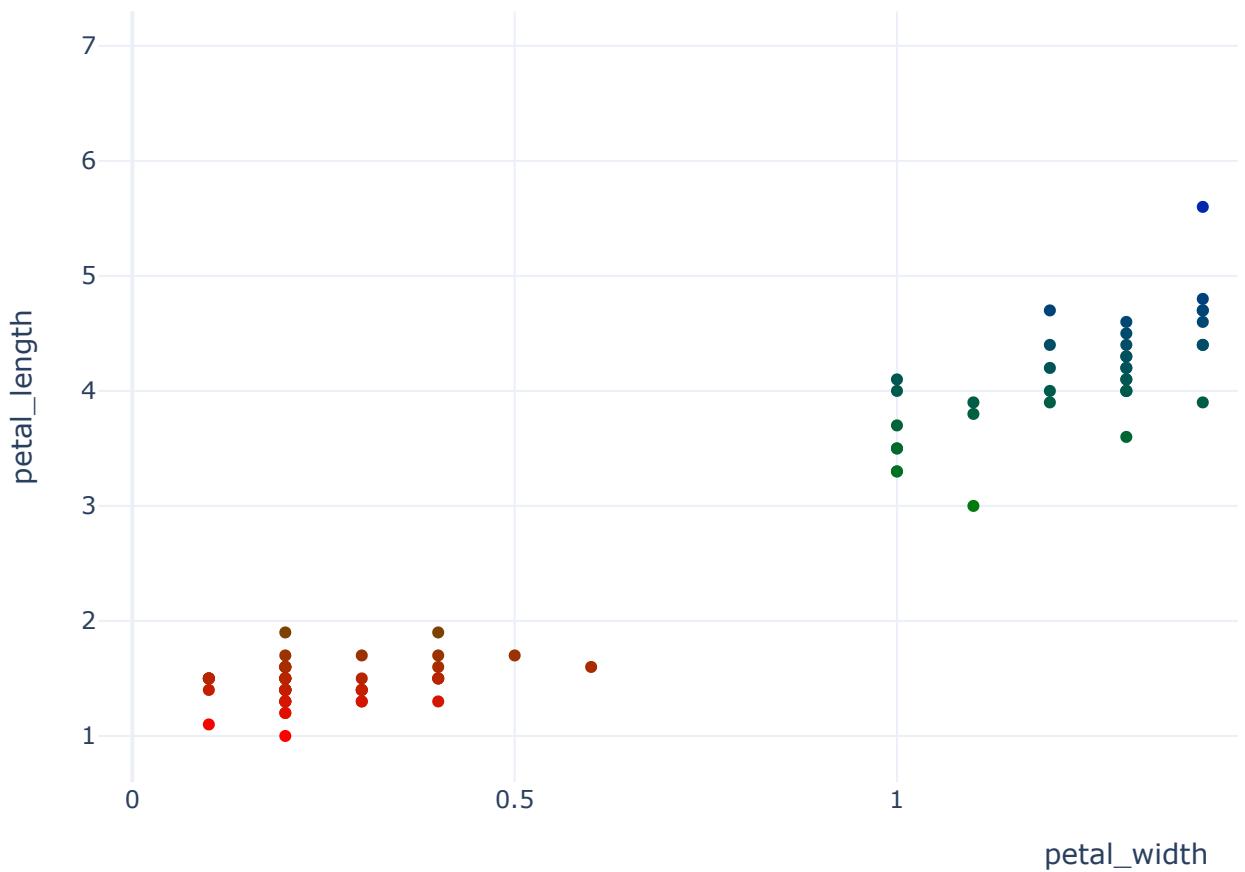
```
fig = px.scatter(iris, x='petal_width', y='petal_length',
                 color='petal_length',
                 color_continuous_scale='Inferno_r')
fig.show()
```



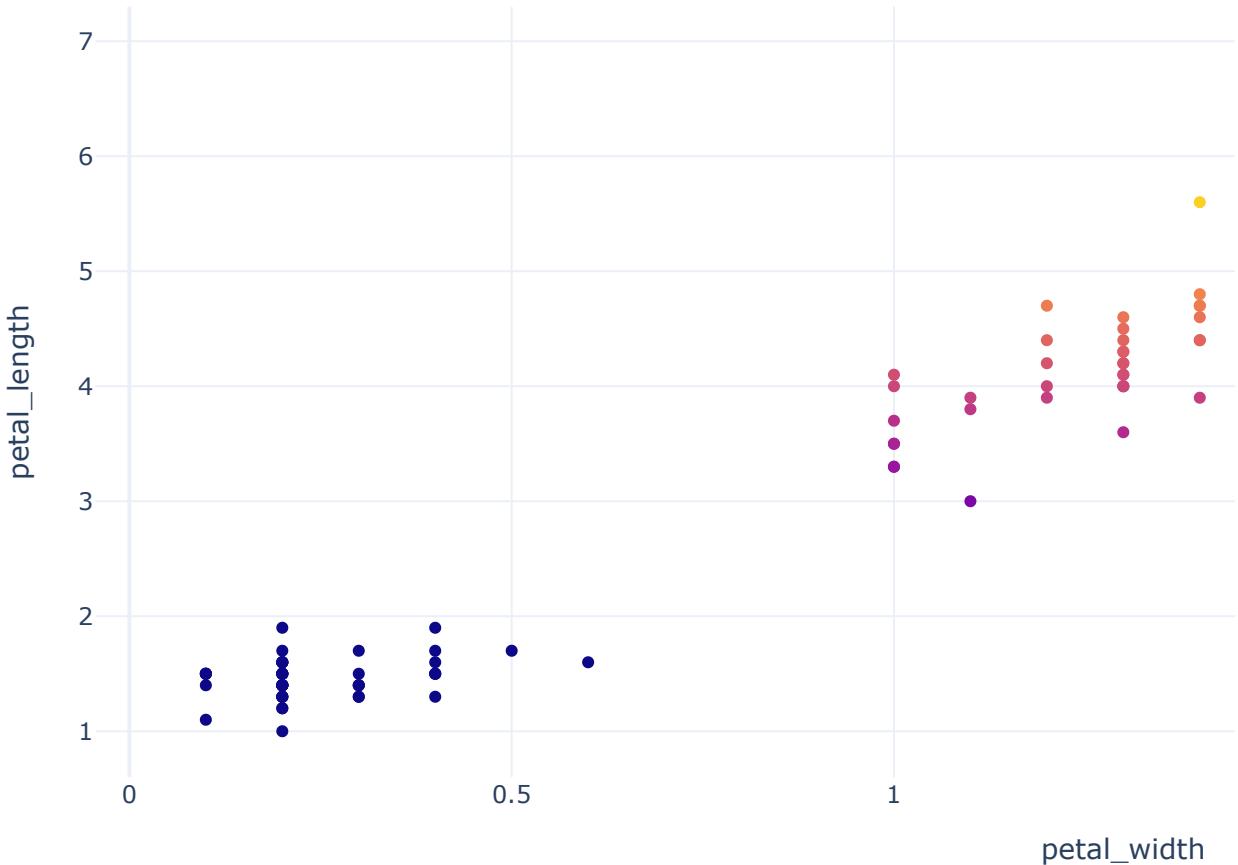
```
fig = px.scatter(iris, x='petal_width', y='petal_length',
                 color='petal_length',
                 color_continuous_scale=['red', 'green', 'blue'])
fig.show()
```



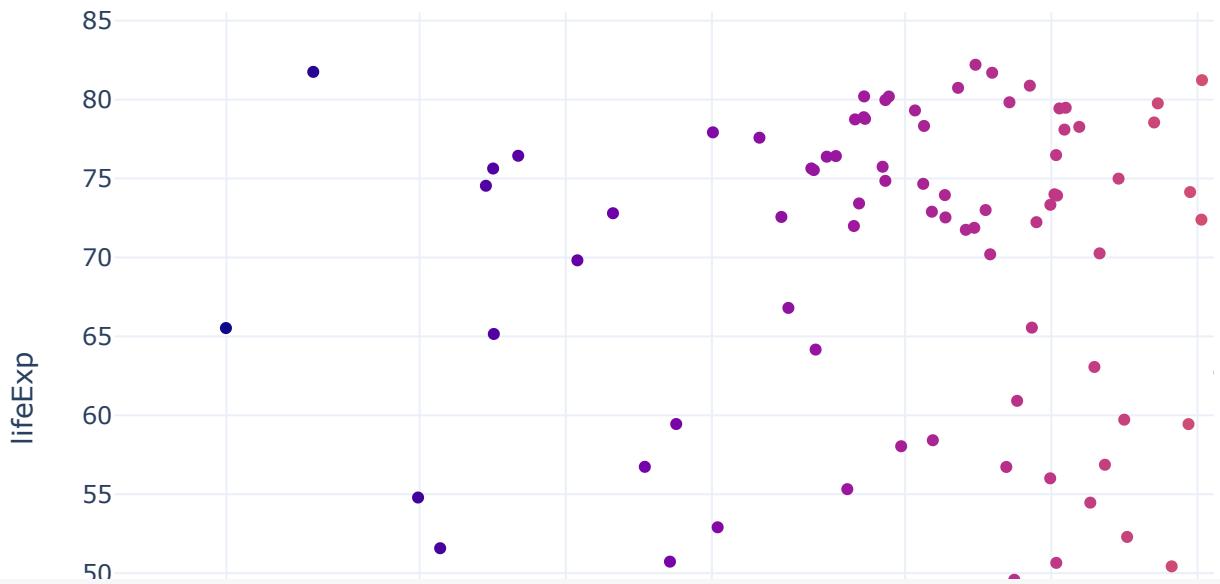
```
fig = px.scatter(iris, x='petal_width', y='petal_length',
                  color='petal_length',
                  color_continuous_scale=[(0, 'red'), (0.3, 'green'), (1, 'blue')])
fig.show()
```



```
fig = px.scatter(iris, x='petal_width', y='petal_length',
                 color='petal_length',
                 range_color=[2, 6])
fig.show()
```



```
fig = px.scatter(gapminder_2007, y='lifeExp', x='pop',
                 color=np.log10(gapminder_2007['pop']),
                 hover_name='country', log_x=True)
fig.update_layout(coloraxis_colorbar=dict(
    title='Population',
    tickvals=[6, 7, 8, 9],
    ticktext=['1M', '10M', '100M', '1B'],
))
fig.show()
```



```
fig = go.Figure()
values = list(range(40))
fig.add_trace(go.Scatter(x=values, y=values,
                        marker=dict(size=14,
                                    cmax=30, cmin=0,
                                    color=values,
                                    colorbar=dict(title='Colorbar'),
                                    colorscale='Inferno'),
                        mode='markers'))
fig.show()
```

## ▼ 텍스트 및 주석(Text and Annotation)

```
40  
gapminder_asia_2007 = gapminder.query("year == 2007 and continent == 'Asia'")  
gapminder_asia_2007
```

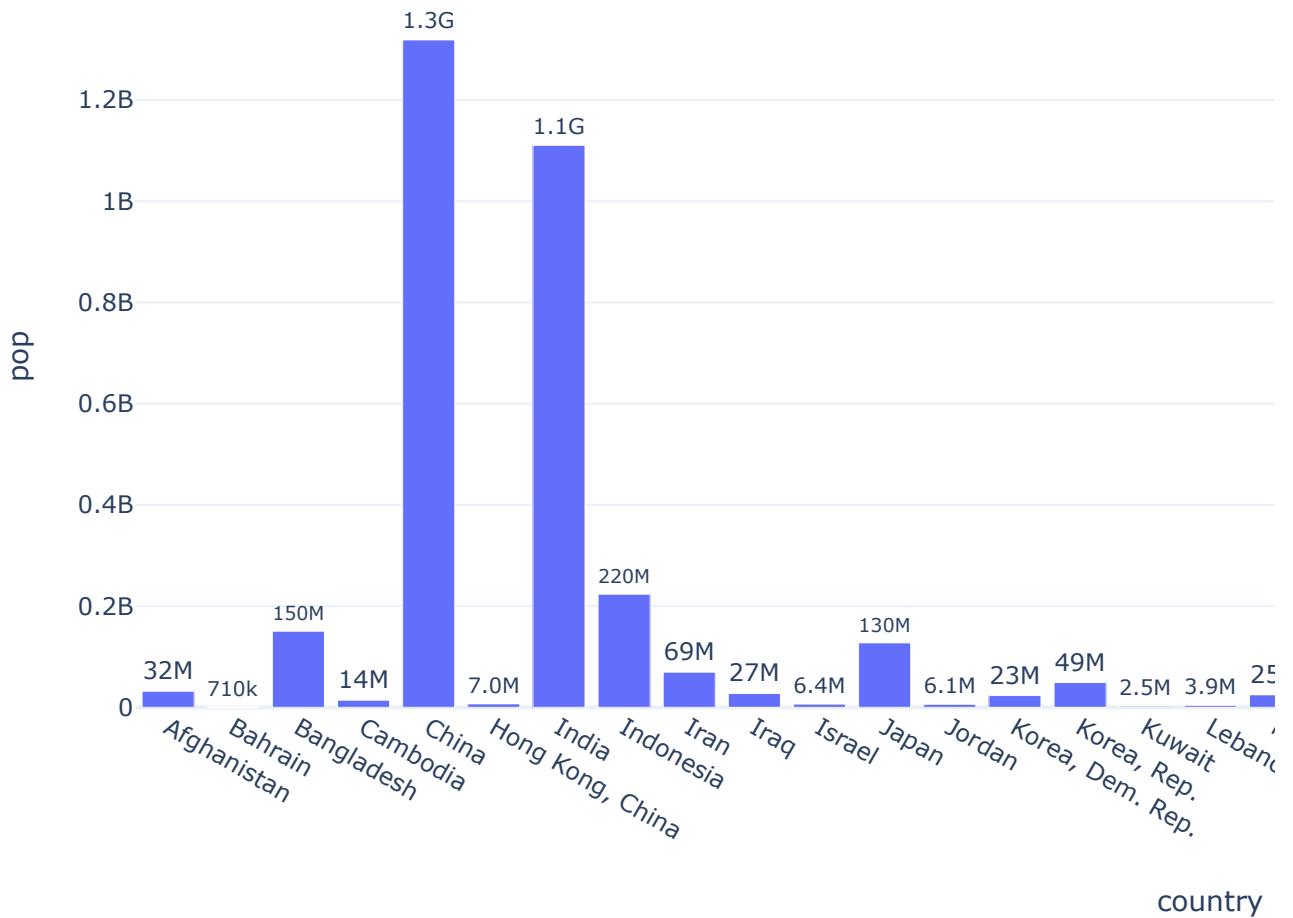
	country	continent	year	lifeExp	pop	gdpPerCap	iso_alpha	i:
11	Afghanistan	Asia	2007	43.828	31889923	974.580338	AFG	
95	Bahrain	Asia	2007	75.635	708573	29796.048340	BHR	
107	Bangladesh	Asia	2007	64.062	150448339	1391.253792	BGD	
227	Cambodia	Asia	2007	59.723	14131858	1713.778686	KHM	

```
fig = px.scatter(gapminder_asia_2007, x='gdpPerCap', y='lifeExp',
                  text='country', log_x=True, size_max=60)
fig.update_traces(textposition='top center')
fig.update_layout(height=800)
fig.show()
```

```

fig = px.bar(gapminder_asia_2007, y='pop', x='country', text='pop')
fig.update_traces(texttemplate='%{text:.2s}', textposition='outside')
fig.update_layout()
fig.show()

```



```

fig = go.Figure()
fig.add_trace(go.Scatter(x=[0, 1, 2, 3, 4, 5, 6, 7, 8],
                         y=[0, 1, 4, 2, 3, 3, 2, 5, 6]))
fig.add_trace(go.Scatter(x=[0, 1, 2, 3, 4, 5, 6, 7, 8],
                         y=[0, 2, 3, 1, 4, 2, 3, 4, 5]))
fig.add_annotation(x=2, y=4, text='Annotation 1')
fig.add_annotation(x=4, y=4, text='Annotation 1')
fig.update_annotations(dict(xref='x', yref='y',
                           showarrow=True, arrowhead=7, ax=20, ay=-40))
fig.update_layout(showlegend=False)
fig.show()

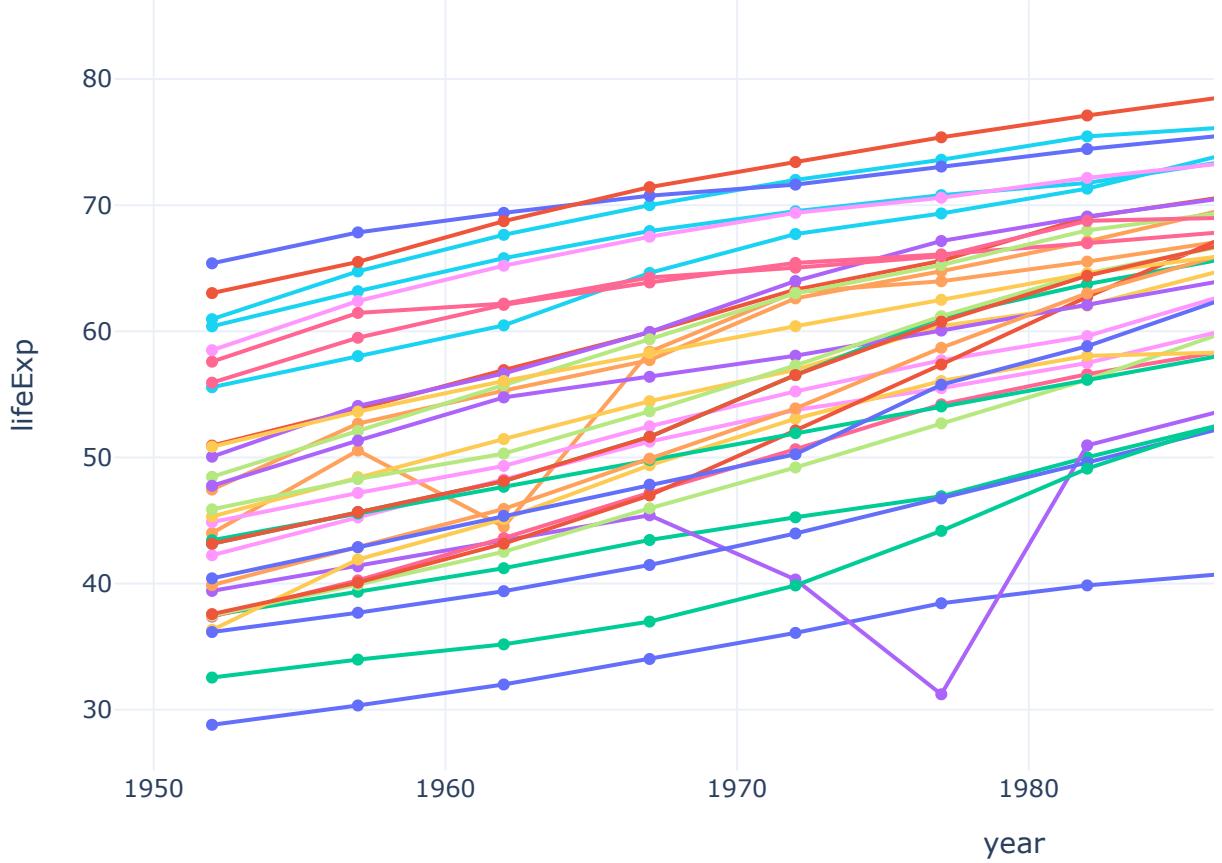
```



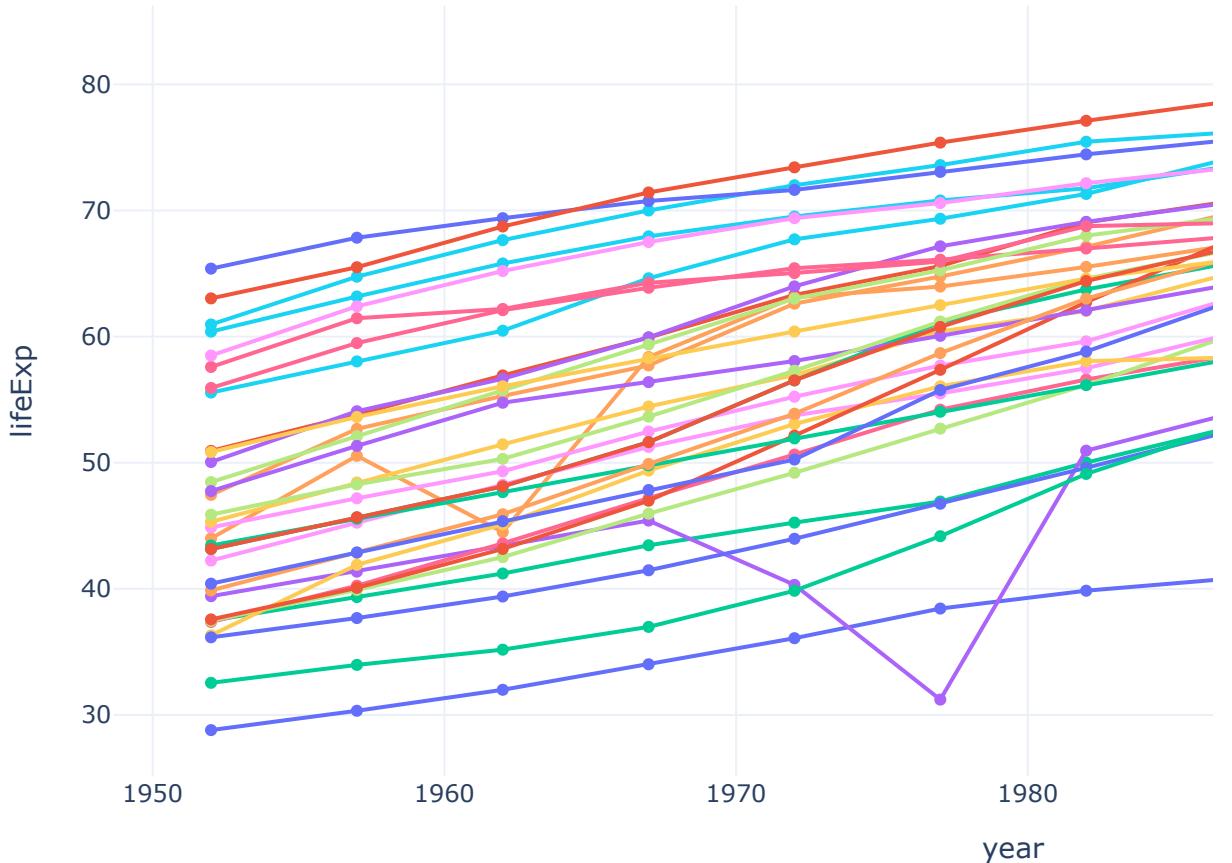
```
fig.add_annotation(x=5, y=3, xref='x', yref='y',
                   text='(5, 3)', showarrow=True,
                   font=dict(family='Courier New, monospace',
                             size=16,
                             color='blue'),
                   align='center',
                   arrowhead=2,
                   arrowsize=1,
                   arrowwidth=2,
                   arrowcolor='orange',
                   ax=20, ay=-40,
                   bordercolor='red',
                   borderwidth=2,
                   borderpad=4,
                   bgcolor='yellow',
                   opacity=0.8)
fig.update_layout(showlegend=False)
fig.show()
```

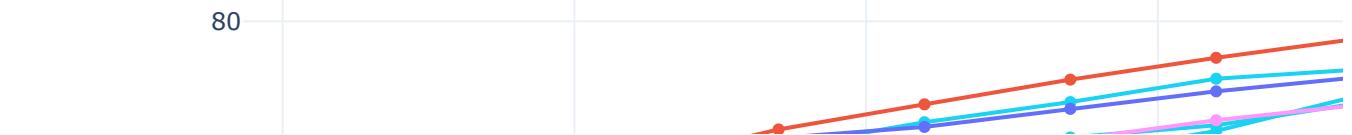


```
fig = px.line(gapminder_asia, x='year', y='lifeExp', color='country')
fig.update_traces(mode='markers+lines')
fig.show()
```

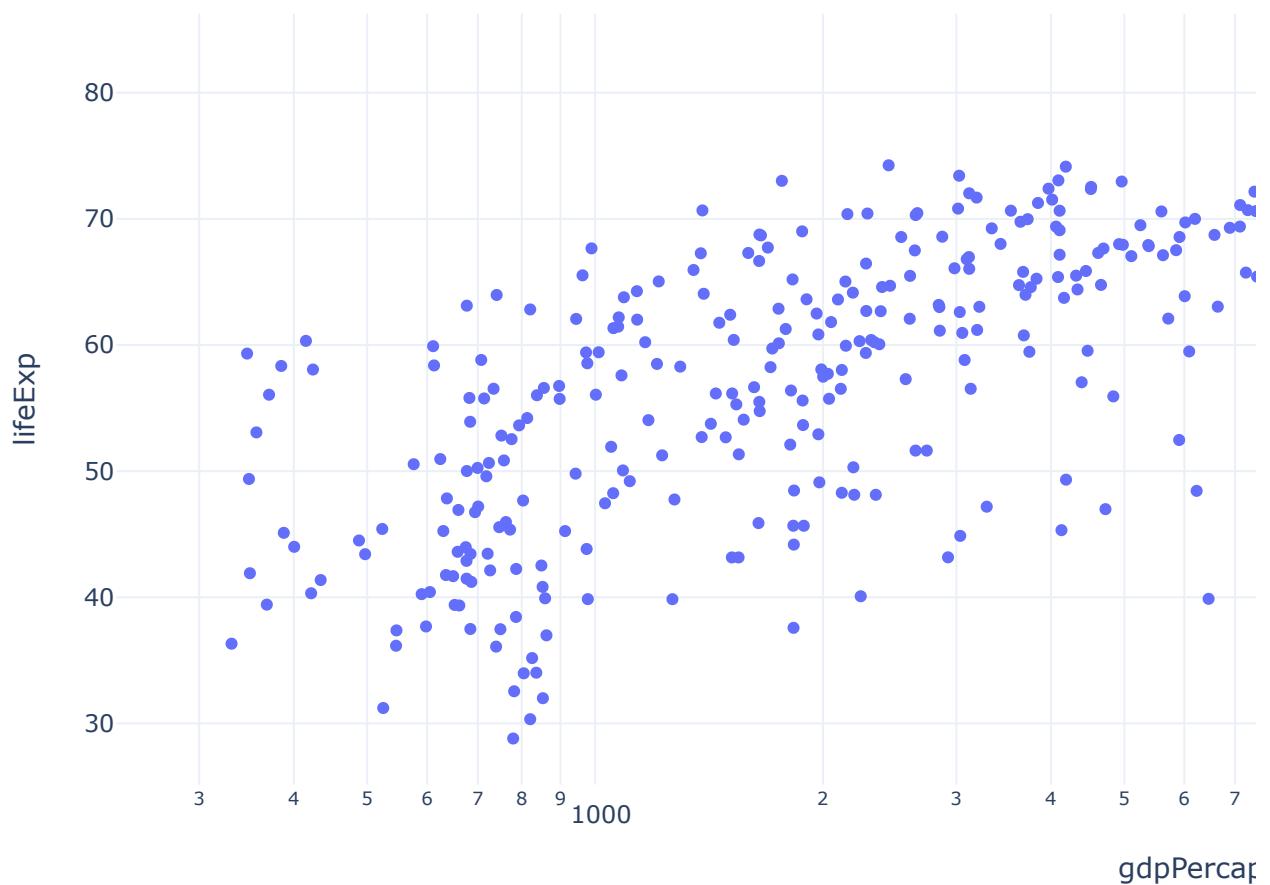


```
fig.update_traces(mode='markers+lines', hovertemplate=None)
fig.update_layout(hovermode='x')
fig.show()
```

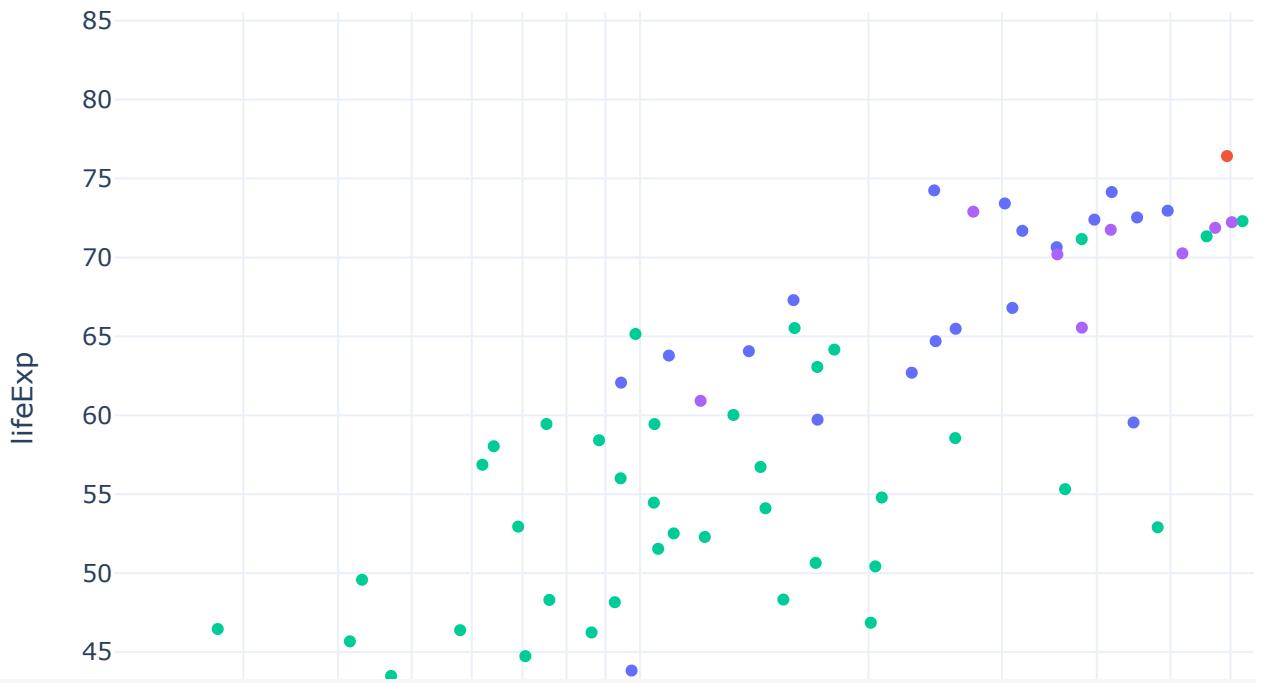




```
fig = px.scatter(gapminder_asia, x='gdpPerCap', y='lifeExp', log_x=True,
                 hover_name='country', hover_data=['continent', 'pop'])
fig.show()
```



```
fig = px.scatter(gapminder_2007, x='gdpPerCap', y='lifeExp', log_x=True,
                 color='continent')
fig.update_traces(hovertemplate='GDP: %{x} <br> Life Expectancy: %{y}')
fig.show()
```

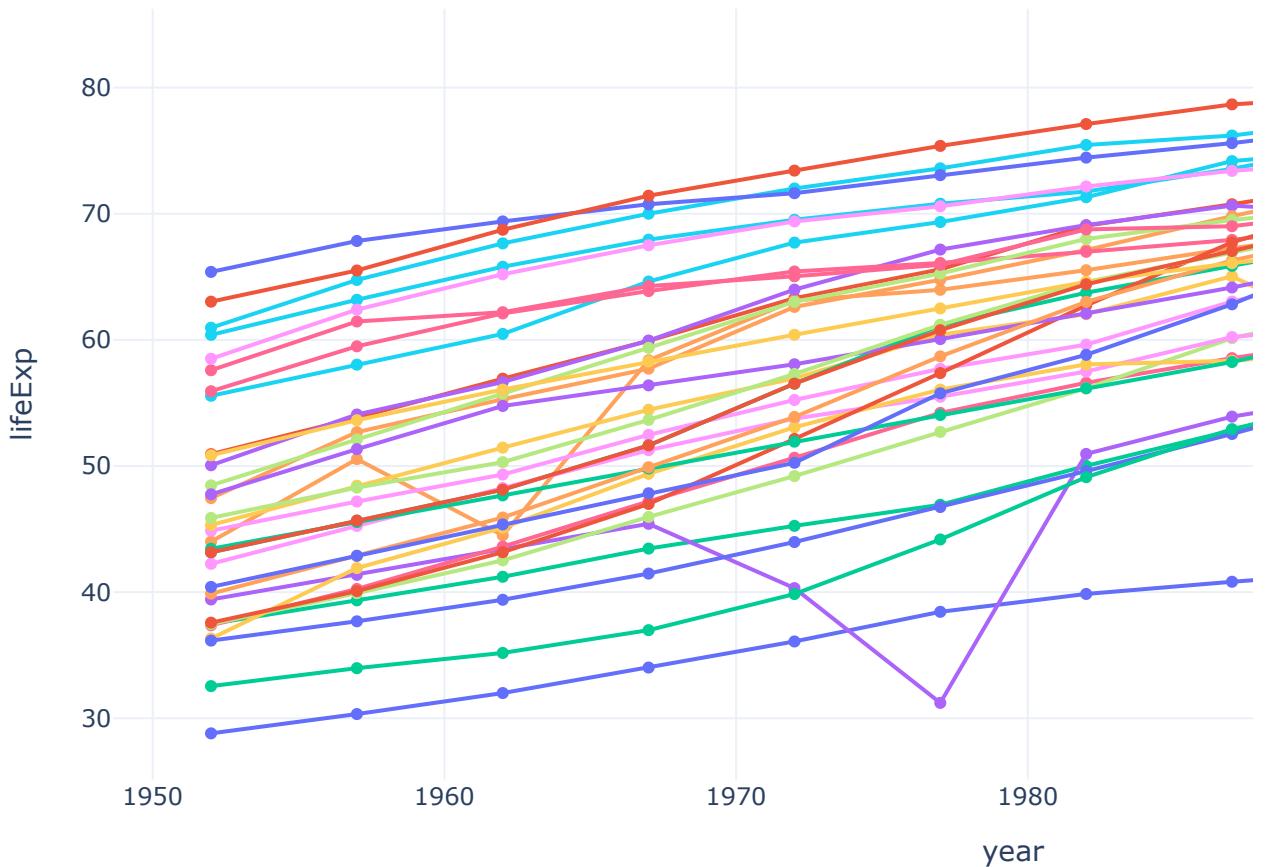


```
fig = px.line(gapminder_asia, x='year', y='lifeExp', color='country')
fig.update_traces(mode='markers+lines')
fig.update_xaxes(showspikes=True)
fig.update_yaxes(showspikes=True)
fig.show()
```

```

fig = px.line(gapminder_asia, x='year', y='lifeExp', color='country')
fig.update_traces(mode='markers+lines')
fig.update_xaxes(showspikes=True, spikecolor='green', spikesnap='cursor', spikemode='across')
fig.update_yaxes(showspikes=True, spikecolor='orange', spikethickness=2)
fig.update_layout(spikedistance=1000, hoverdistance=100)
fig.show()

```



## ▼ 모양(Shape)

```

fig = go.Figure()
fig.add_trace(go.Scatter(
    x=[2, 3.5, 6], y=[1, 1.5, 1],
    text=['Vertical Line',
          'Horizontal Dashed Line',
          'Diagonal Dotted Line'],
    mode='text',
))
fig.update_xaxes(range=[0, 7])
fig.update_yaxes(range=[0, 2.5])
fig.add_shape(dict(type='line', x0=1, y0=0, x1=1, y1=2,
                  line=dict(color='royalblue', width=3)))
fig.add_shape(dict(type='line', x0=2, y0=2, x1=5, y1=2,
                  line=dict(color='lightgreen', width=1, dash=[4, 4])))

```

```

    line=dict(color='lightseagreen', width=4, dash='dashdot'))
fig.add_shape(type='line', x0=4, y0=0, x1=6, y1=2,
              line=dict(color='mediumpurple', width=4, dash='dot'))
fig.update_shapes(dict(xref='x', yref='y'))
fig.show()

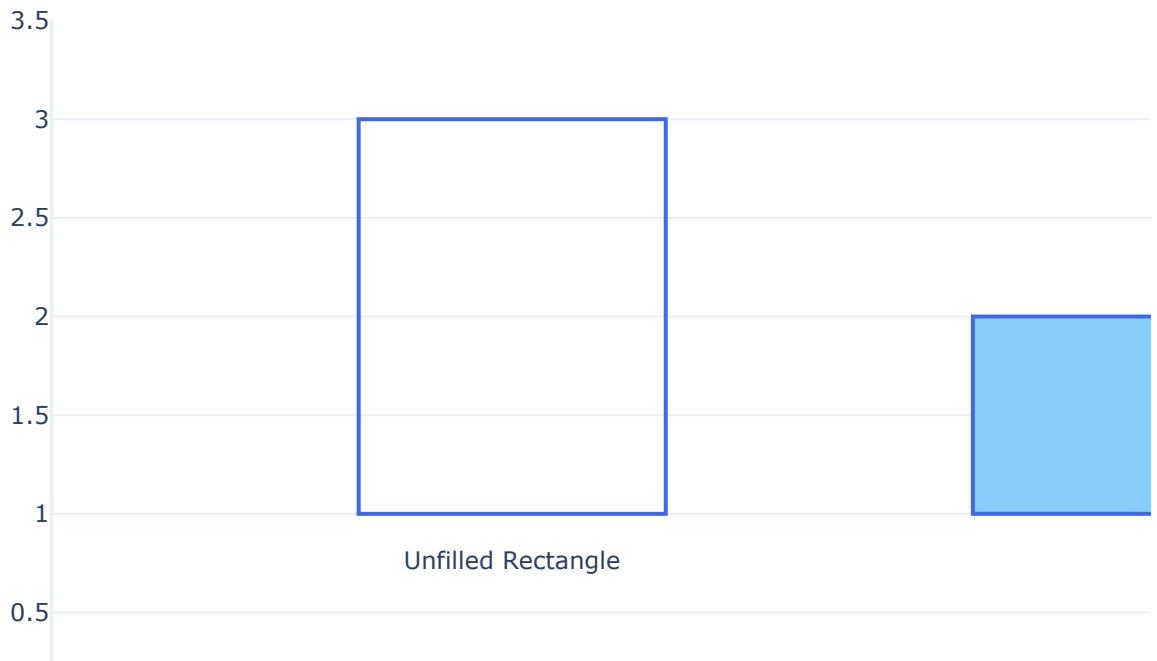
```



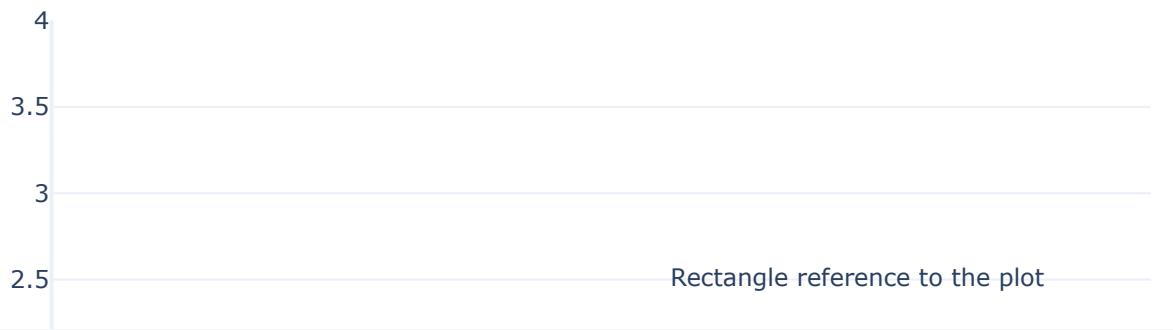
```

fig = go.Figure()
fig.add_trace(go.Scatter(x=[1.5, 4.5], y=[0.75, 0.75],
                        text=['Unfilled Rectangle', 'Filled Rectangle'],
                        mode='text'))
fig.update_xaxes(range=[0, 7], showgrid=False)
fig.update_yaxes(range=[0, 3.5])
fig.add_shape(type='rect', x0=1, y0=1, x1=2, y1=3,
              line=dict(color='royalblue'))
fig.add_shape(type='rect', x0=3, y0=1, x1=6, y1=2,
              line=dict(color='royalblue', width=2),
              fillcolor='lightskyblue')
fig.update_shapes(dict(xref='x', yref='y'))
fig.show()

```



```
fig = go.Figure()
fig.add_trace(go.Scatter(x=[1.5, 3], y=[2.5, 2.5],
                        text=['Rectangle reference to the plot',
                              'Rectangle reference to the axes'],
                        mode='text'))
fig.update_xaxes(range=[0, 4], showgrid=False)
fig.update_yaxes(range=[0, 4])
fig.add_shape(type='rect', xref='x', yref='y',
              x0=2.5, y0=0, x1=3.5, y1=2,
              line=dict(color='royalblue', width=3),
              fillcolor='lightskyblue')
fig.add_shape(type='rect', xref='paper', yref='paper',
              x0=0.25, y0=0, x1=0.5, y1=0.5,
              line=dict(color='royalblue', width=3),
              fillcolor='darkblue')
fig.show()
```



```
fig = go.Figure()
fig.add_trace(go.Scatter(x=[1.5, 3.5], y=[0.75, 2.5],
                        text=['Unfilled Circle',
                              'Filled Circle'],
                        mode='text'))
fig.update_xaxes(range=[0, 4.5], zeroline=False)
fig.update_yaxes(range=[0, 4.5])
fig.add_shape(type='circle', xref='x', yref='y',
              x0=1, y0=1, x1=3, y1=3,
              line_color='lightskyblue')
fig.add_shape(type='circle', xref='x', yref='y',
              x0=3, y0=3, x1=4, y1=4,
              line=dict(color='royalblue'),
              fillcolor='darkblue')
fig.update_layout(width=800, height=800)
fig.show()
```