

## What is Plotly express?

Plotly Express is the easy-to-use, high-level interface to Plotly, which operates on a variety of types of data and produces easy-to-style figures. Plotly Express provides functions to visualize a variety of types of data. Most functions such as `px.bar` or `px.scatter` expect to operate on column-oriented data of the type you might store in a `DataFrame`.

## Plotly using express

```
In [ ]: !pip install moviepy
```

```
In [ ]: import plotly.graph_objects as go
import numpy as np
import pandas as pd
import plotly.express as px
import moviepy.editor as mpy
import io
from PIL import Image
```

### Import Dataset

```
In [ ]: tips = px.data.tips()
iris = px.data.iris()
gap = px.data.gapminder()
```

```
In [ ]: tips.head(5)
```

```
Out[ ]:
```

	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

```
In [ ]: iris.head(5)
```

```
Out[ ]:
```

	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

```
In [ ]: gap
```

```
Out[ ]:
```

	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

```
In [ ]: # scatter plot using plotly go
temp_df = gap[gap['year'] == 2007]
temp_df.head()
```

```
Out[ ]:
```

	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
11	Afghanistan	Asia	2007	43.828	31889923	974.580338	AFG	4
23	Albania	Europe	2007	76.423	3600523	5937.029526	ALB	8
35	Algeria	Africa	2007	72.301	33333216	6223.367465	DZA	12
47	Angola	Africa	2007	42.731	12420476	4797.231267	AGO	24
59	Argentina	Americas	2007	75.320	40301927	12779.379640	ARG	32

## Scatter Plot

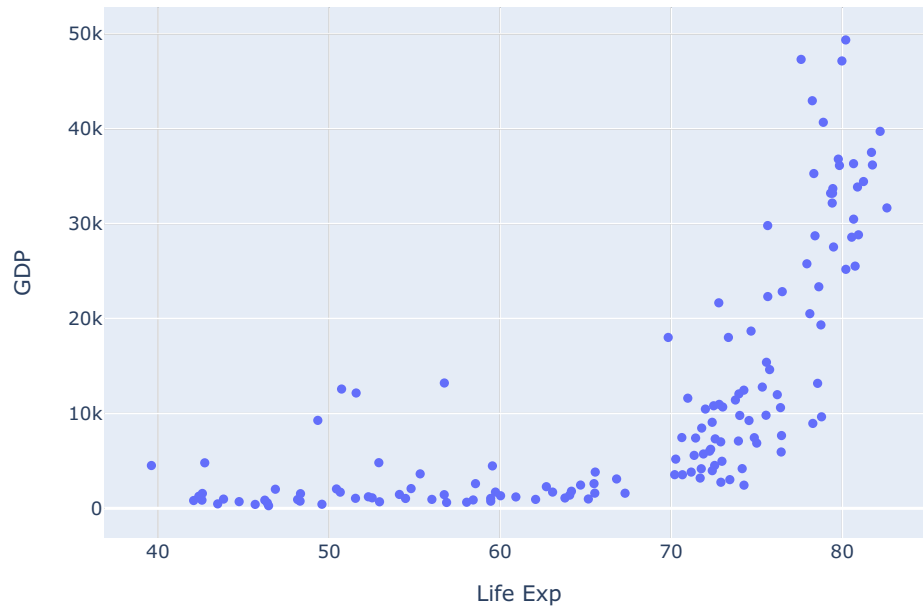
```
In [ ]: trace1 = go.Scatter(x=temp_df['lifeExp'],
                           y=temp_df['gdpPercap'], mode='markers')

data = [trace1]

layout = go.Layout(title='Life Expectancy vs percapita for 2007', xaxis={
                    'title': 'Life Exp'}, yaxis={'title': 'GDP'})
fig = go.Figure(data, layout)
fig.show("notebook")
```



Life Expectancy vs percapita for 2007

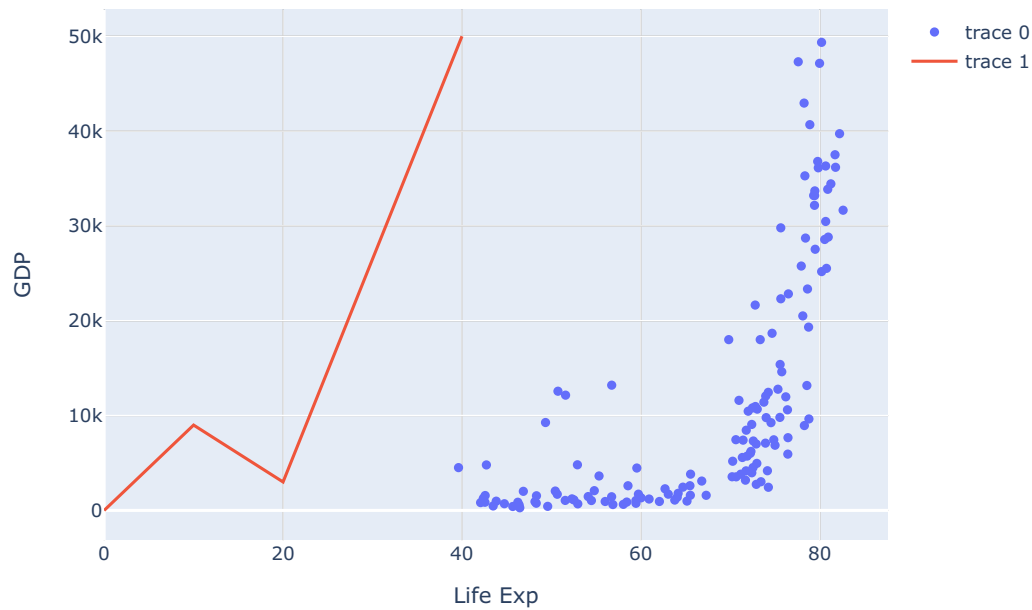


## Line and Scatter Plot

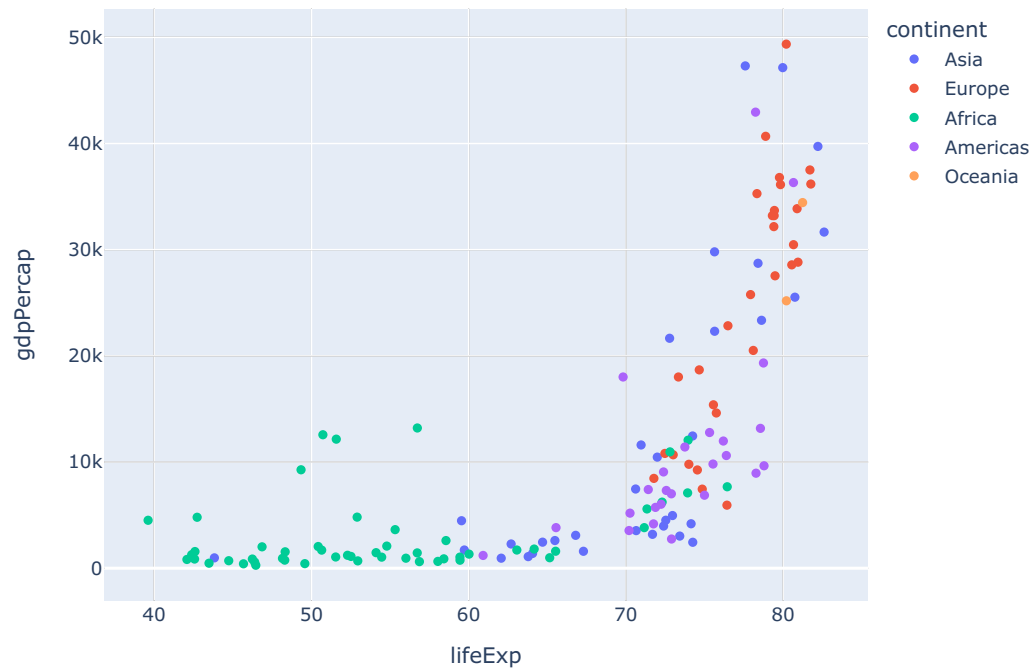
```
In [ ]: trace1 = go.Scatter(x=temp_df['lifeExp'],
                           y=temp_df['gdpPercap'], mode='markers')
trace2 = go.Scatter(x=[0, 10, 20, 40], y=[0, 9000, 3000, 50000], mode='lines')
data = [trace1, trace2]

layout = go.Layout(title='Life Expectancy vs percapita for 2007', xaxis={
                    'title': 'Life Exp'}, yaxis={'title': 'GDP'})
fig = go.Figure(data, layout)
fig.show("notebook")
```

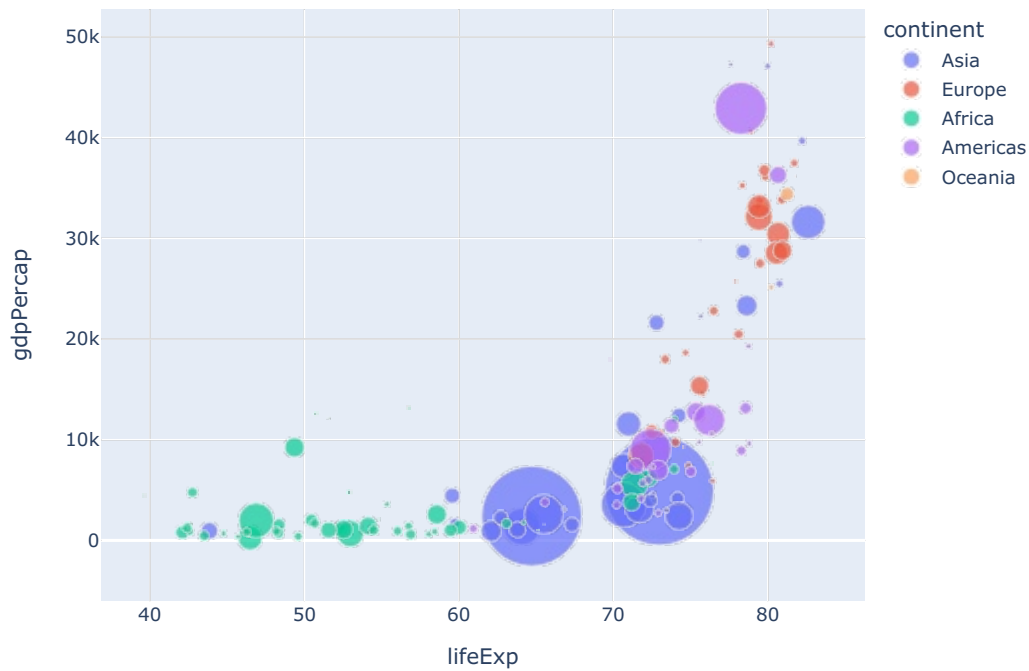
## Life Expectancy vs percapita for 2007



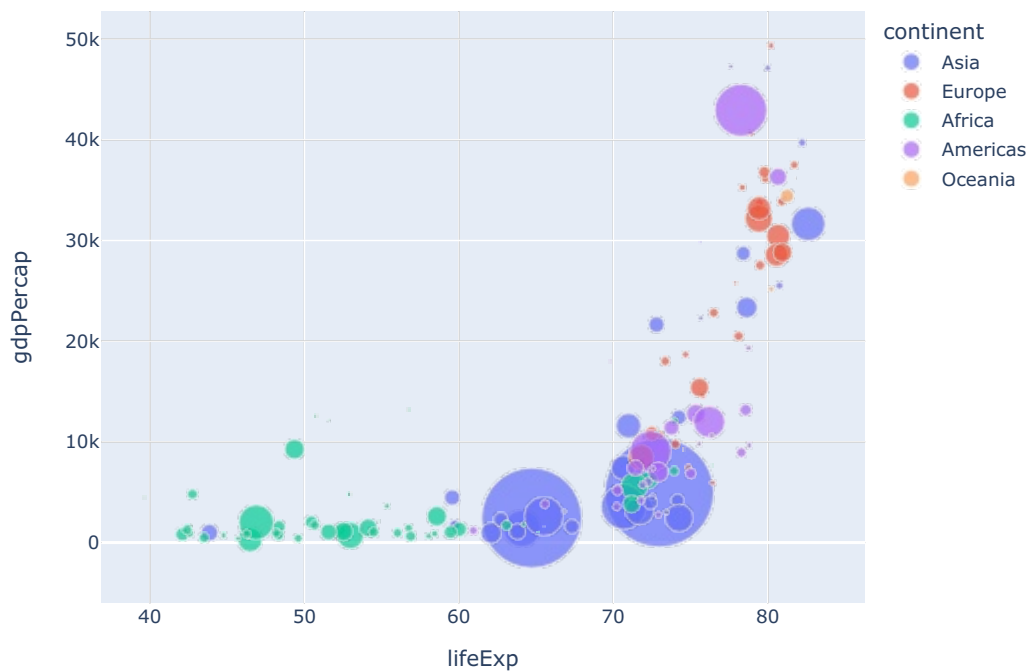
```
In [ ]: # plot life exp and gdp ->scatter
fig = px.scatter(temp_df, x='lifeExp', y='gdpPerCap', color='continent')
fig.show("notebook")
```



```
In [ ]: fig = px.scatter(temp_df, x='lifeExp', y='gdpPerCap',
                        color='continent', size='pop', size_max=50)
fig.show("notebook")
```

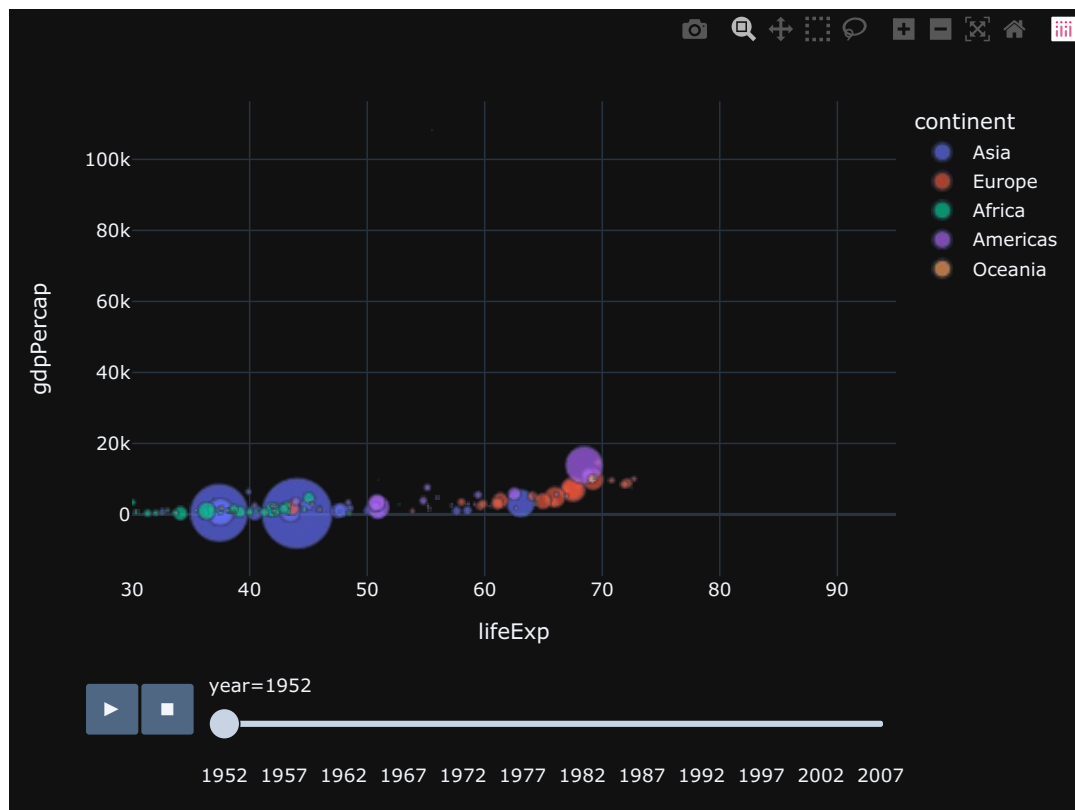


```
In [ ]: # hover to show country name
fig = px.scatter(temp_df, x='lifeExp', y='gdpPercap',
                 color='continent', size='pop', size_max=50, hover_name='country')
fig.show("notebook")
```



## Animation

```
In [ ]: # animation
# hover to show country name
fig = px.scatter(gap, x='lifeExp', y='gdpPercap',
                 color='continent', size='pop',
                 size_max=50, hover_name='country',
                 range_x=[30, 95],
                 animation_frame='year', animation_group='country', template='plotly_dark')
fig.show("notebook")
```



## Line Plot

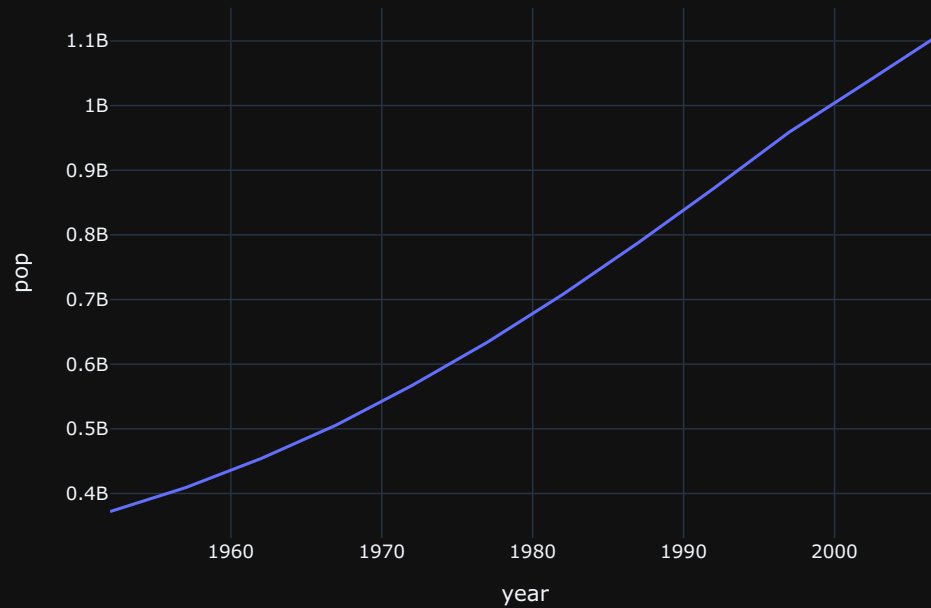
```
In [ ]: temp_df = gap[gap['country'] == 'India']
temp_df.head()
```

```
Out[ ]:
```

	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
696	India	Asia	1952	37.373	372000000	546.565749	IND	356
697	India	Asia	1957	40.249	409000000	590.061996	IND	356
698	India	Asia	1962	43.605	454000000	658.347151	IND	356
699	India	Asia	1967	47.193	506000000	700.770611	IND	356
700	India	Asia	1972	50.651	567000000	724.032527	IND	356

```
In [ ]: fig = px.line(temp_df, x='year', y='pop',
                    title='India population growth', template='plotly_dark')
fig.show("notebook")
```

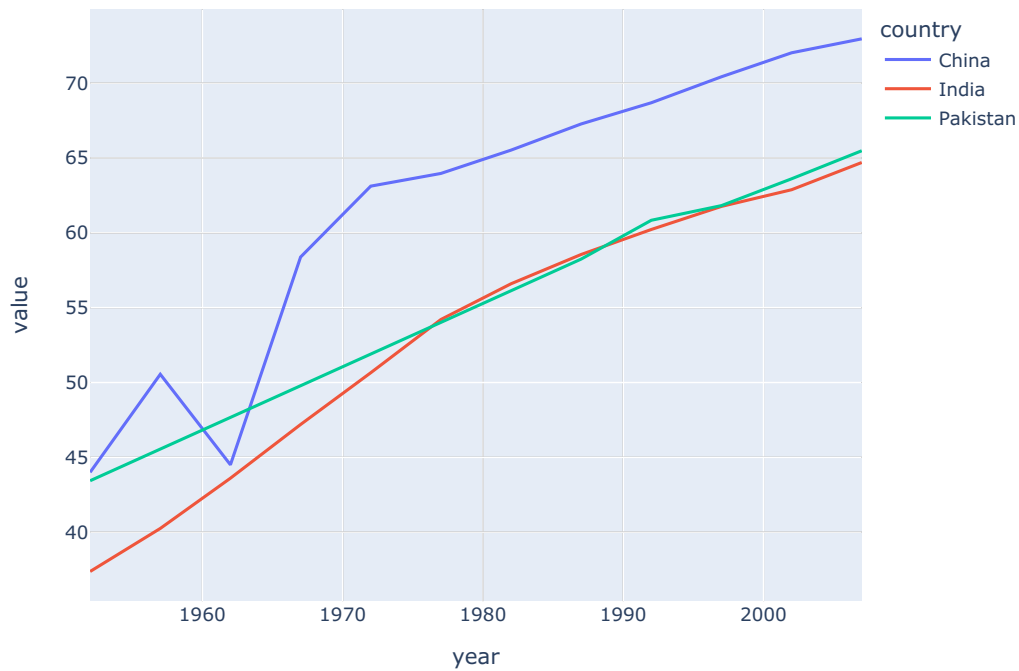
## India population growth



```
In [ ]: temp_df = gap[gap['country'].isin(['India', 'China', 'Pakistan'])].pivot(
        index='year', columns='country', values='lifeExp')
temp_df
```

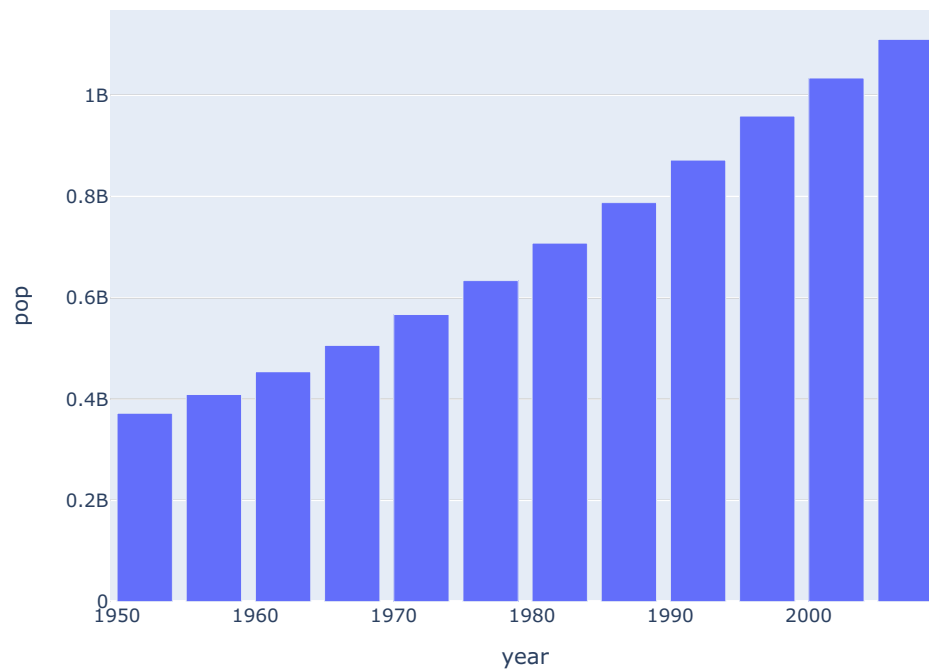
```
Out[ ]: country    China    India    Pakistan
year
1952    44.00000    37.373    43.436
1957    50.54896    40.249    45.557
1962    44.50136    43.605    47.670
1967    58.38112    47.193    49.800
1972    63.11888    50.651    51.929
1977    63.96736    54.208    54.043
1982    65.52500    56.596    56.158
1987    67.27400    58.553    58.245
1992    68.69000    60.223    60.838
1997    70.42600    61.765    61.818
2002    72.02800    62.879    63.610
2007    72.96100    64.698    65.483
```

```
In [ ]: fig = px.line(temp_df, x=temp_df.index, y=temp_df.columns)
fig.show("notebook")
```



## Bar Chart

```
In [ ]: # india's pop over the years
temp_df = gap[gap['country'] == 'India']
fig = px.bar(temp_df, x='year', y='pop')
fig.show("notebook")
```



```
In [ ]: gap
```



Out [ ]:

	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

In [ ]:

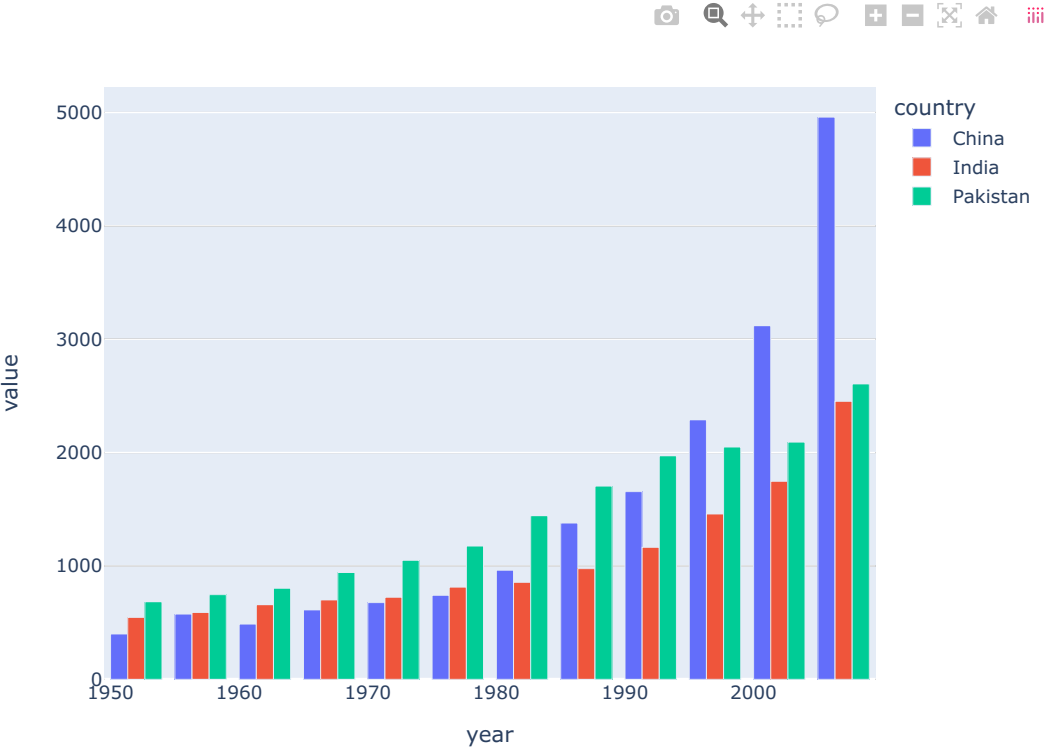
```
# group bar chart
temp_df = gap[gap['country'].isin(['India', 'China', 'Pakistan'])].pivot(
    index='year', columns='country', values='gdpPercap')
temp_df.head()
```

Out [ ]:

country	China	India	Pakistan
year			
1952	400.448611	546.565749	684.597144
1957	575.987001	590.061996	747.083529
1962	487.674018	658.347151	803.342742
1967	612.705693	700.770611	942.408259
1972	676.900092	724.032527	1049.938981

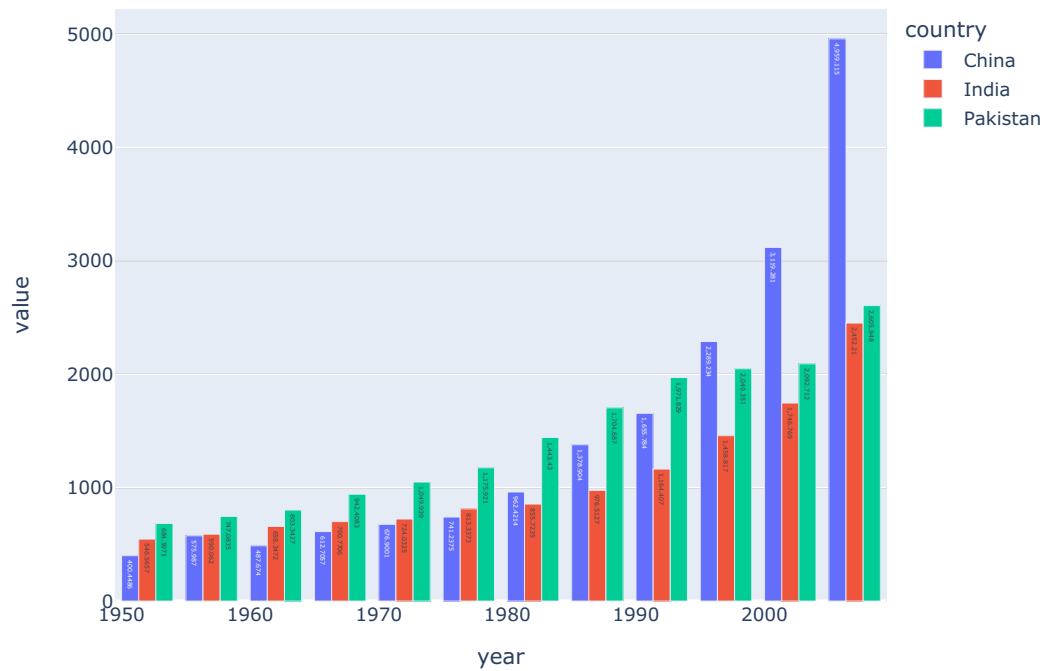
In [ ]:

```
fig = px.bar(temp_df, x=temp_df.index, y=temp_df.columns, barmode='group')
fig.show("notebook")
```

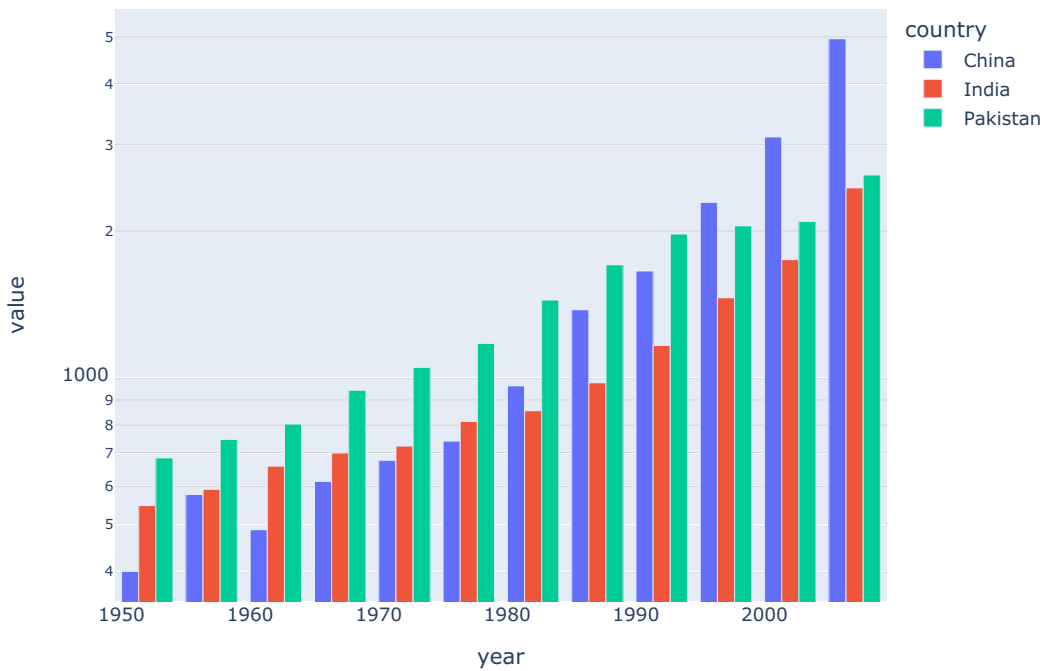


In [ ]:

```
fig = px.bar(temp_df, x=temp_df.index, y=temp_df.columns,
    barmode='group', text_auto=True)
fig.show("notebook")
```



```
In [ ]: fig = px.bar(temp_df, x=temp_df.index, y=temp_df.columns, barmode='group', log_y=True)
fig.show("notebook")
```



```
In [ ]: # stacked bar chart
# pop contribution per country to a continents pop s
temp_df = gap[gap['year'] == 2007]
temp_df
```

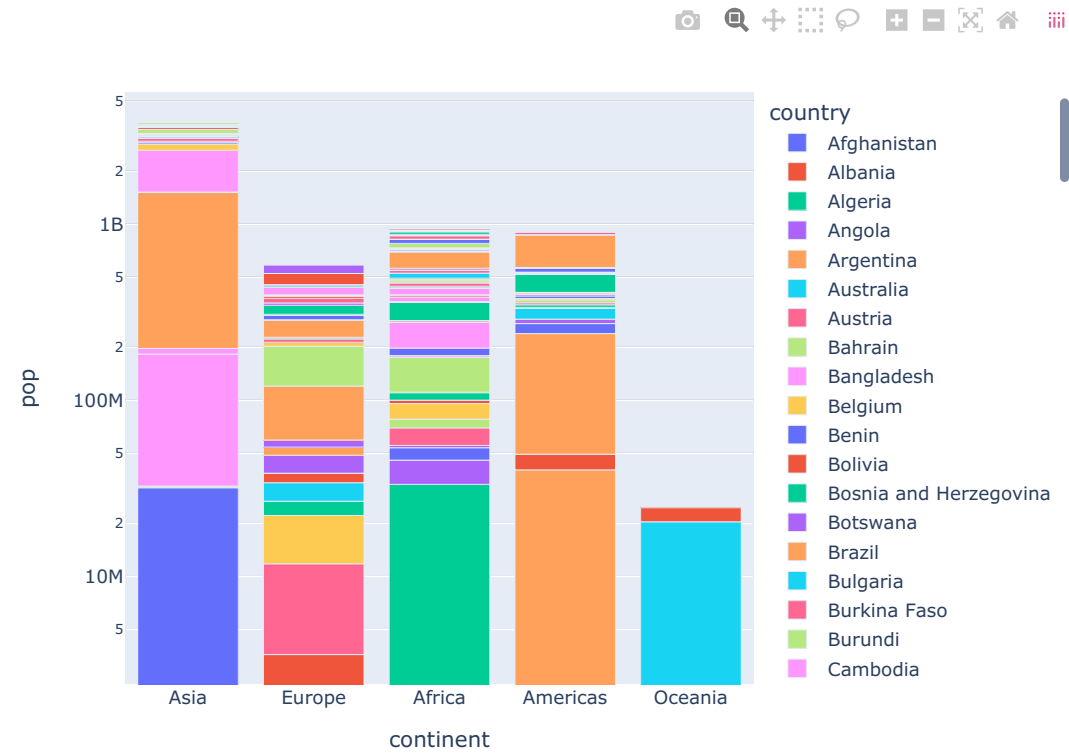
Out[ ]:

	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
11	Afghanistan	Asia	2007	43.828	31889923	974.580338	AFG	4
23	Albania	Europe	2007	76.423	3600523	5937.029526	ALB	8
35	Algeria	Africa	2007	72.301	33333216	6223.367465	DZA	12
47	Angola	Africa	2007	42.731	12420476	4797.231267	AGO	24
59	Argentina	Americas	2007	75.320	40301927	12779.379640	ARG	32
...	...	...	...	...	...	...	...	...
1655	Vietnam	Asia	2007	74.249	85262356	2441.576404	VNM	704
1667	West Bank and Gaza	Asia	2007	73.422	4018332	3025.349798	PSE	275
1679	Yemen, Rep.	Asia	2007	62.698	22211743	2280.769906	YEM	887
1691	Zambia	Africa	2007	42.384	11746035	1271.211593	ZMB	894
1703	Zimbabwe	Africa	2007	43.487	12311143	469.709298	ZWE	716

142 rows × 8 columns

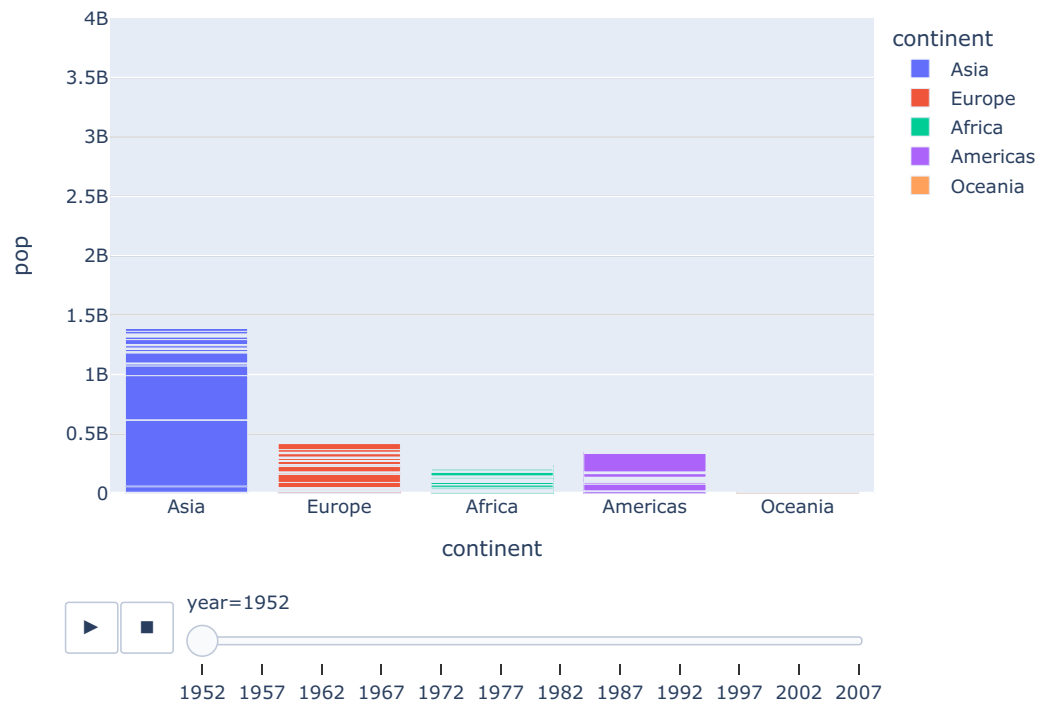
In [ ]:

```
fig = px.bar(temp_df, x='continent', y='pop', color='country', log_y=True)
fig.show("notebook")
```



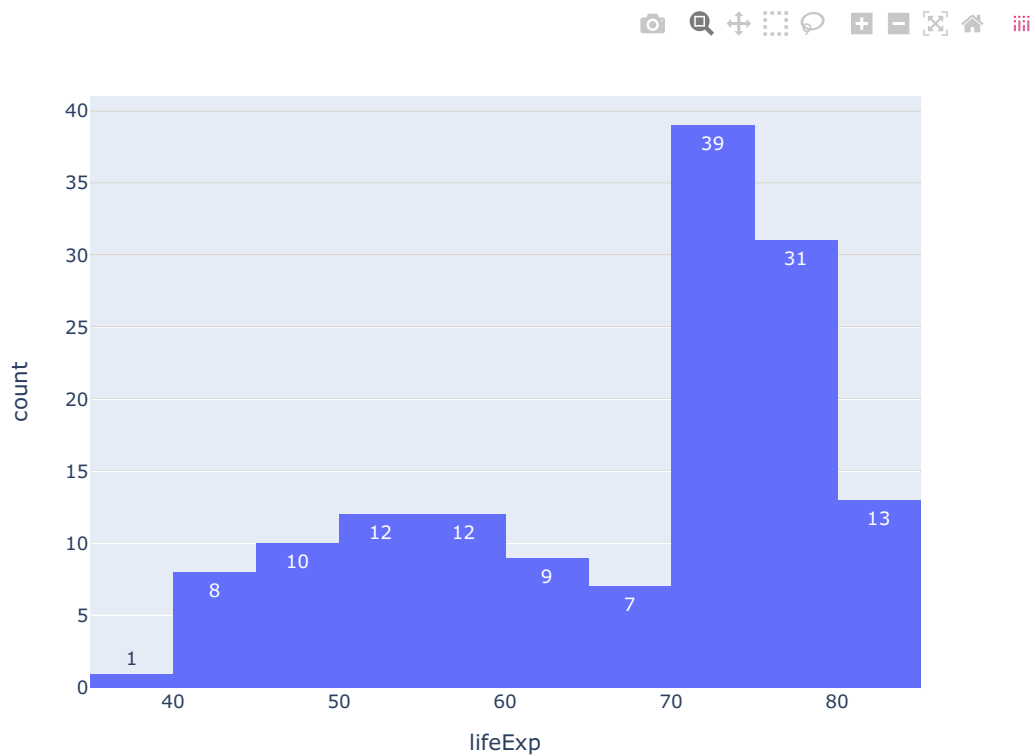
In [ ]:

```
# bar chart animation
fig = px.bar(gap, x='continent', y='pop', color='continent',
             animation_frame='year', animation_group='country', range_y=[0, 4000000000])
fig.show("notebook")
```

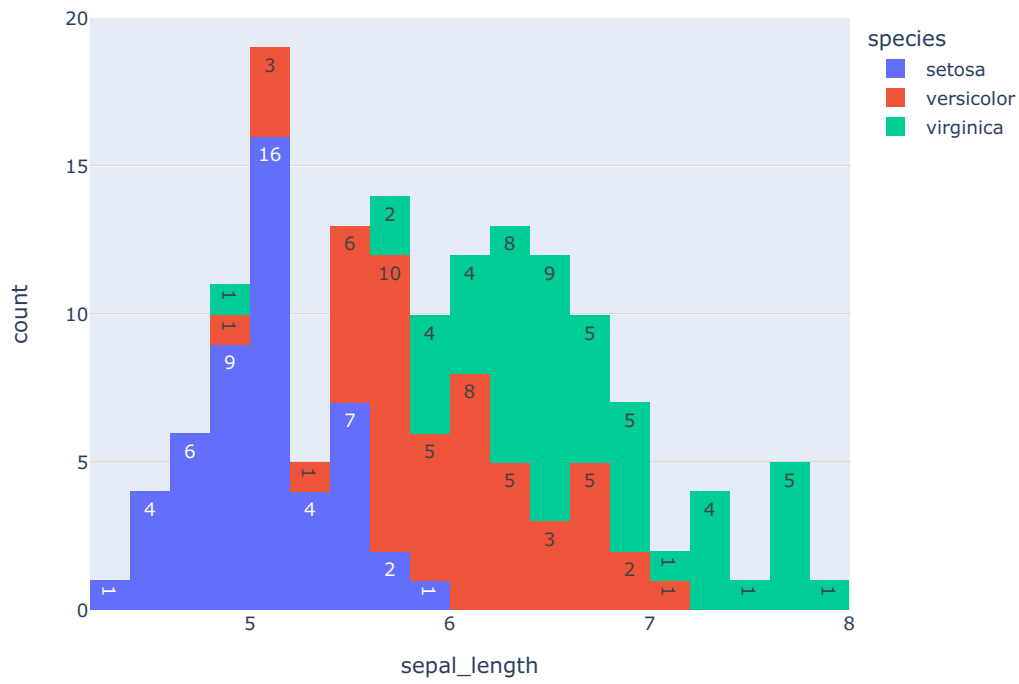


## Histogram

```
In [ ]: temp_df = gap[gap['year'] == 2007]
fig = px.histogram(temp_df, x='lifeExp', nbins=10, text_auto=True)
fig.show("notebook")
```



```
In [ ]: fig = px.histogram(iris, x='sepal_length', color='species', nbins=30, text_auto=True)
fig.show("notebook")
```



## Pie chart

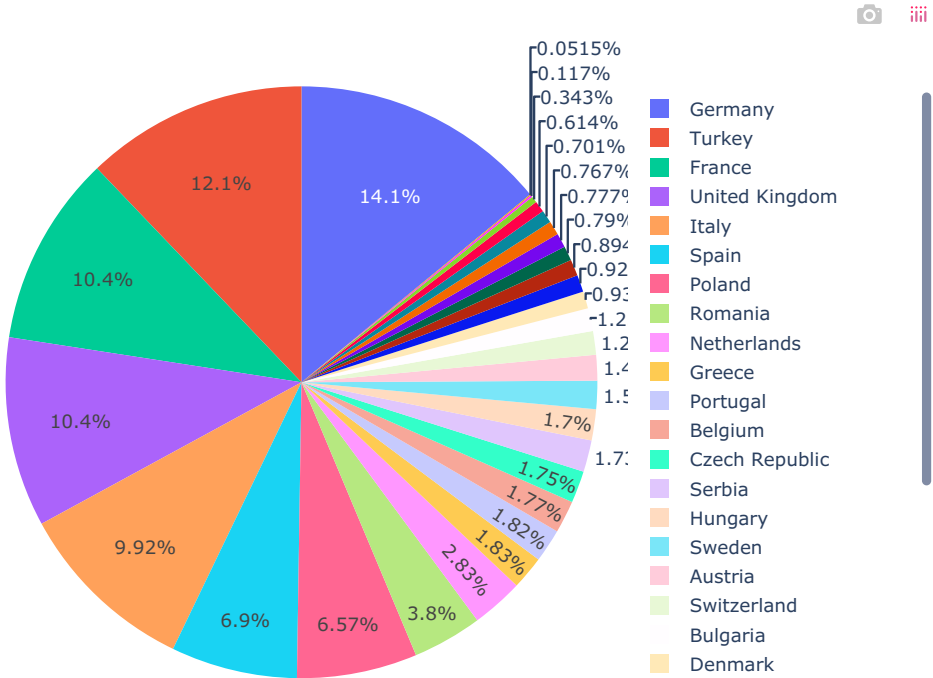
```

In [ ]: # population country in 2007
temp_df = gap[(gap['year'] == 2007) & (gap['continent'] == 'Europe')]
temp_df
    
```

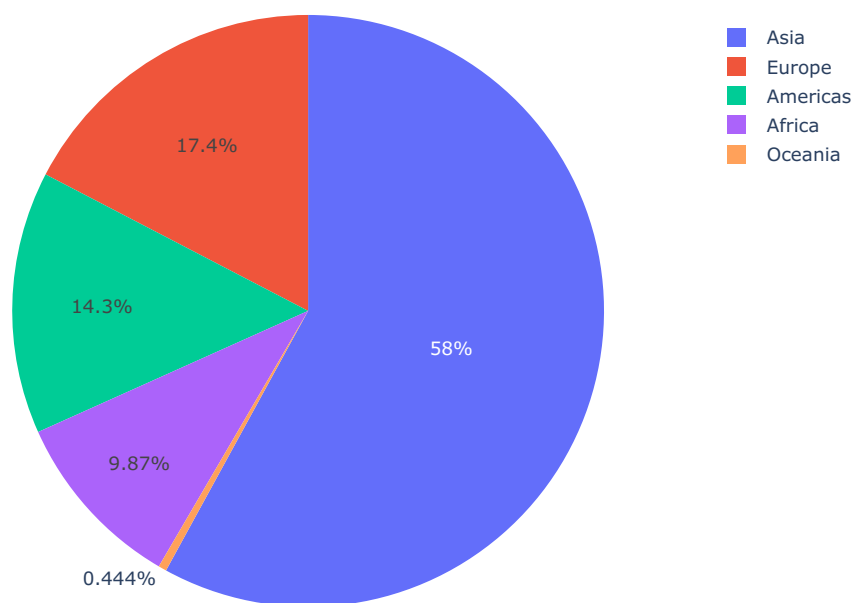
Out[ ]:

	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
23	Albania	Europe	2007	76.423	3600523	5937.029526	ALB	8
83	Austria	Europe	2007	79.829	8199783	36126.492700	AUT	40
119	Belgium	Europe	2007	79.441	10392226	33692.605080	BEL	56
155	Bosnia and Herzegovina	Europe	2007	74.852	4552198	7446.298803	BIH	70
191	Bulgaria	Europe	2007	73.005	7322858	10680.792820	BGR	100
383	Croatia	Europe	2007	75.748	4493312	14619.222720	HRV	191
407	Czech Republic	Europe	2007	76.486	10228744	22833.308510	CZE	203
419	Denmark	Europe	2007	78.332	5468120	35278.418740	DNK	208
527	Finland	Europe	2007	79.313	5238460	33207.084400	FIN	246
539	France	Europe	2007	80.657	61083916	30470.016700	FRA	250
575	Germany	Europe	2007	79.406	82400996	32170.374420	DEU	276
599	Greece	Europe	2007	79.483	10706290	27538.411880	GRC	300
683	Hungary	Europe	2007	73.338	9956108	18008.944440	HUN	348
695	Iceland	Europe	2007	81.757	301931	36180.789190	ISL	352
755	Ireland	Europe	2007	78.885	4109086	40675.996350	IRL	372
779	Italy	Europe	2007	80.546	58147733	28569.719700	ITA	380
1019	Montenegro	Europe	2007	74.543	684736	9253.896111	MNE	499
1091	Netherlands	Europe	2007	79.762	16570613	36797.933320	NLD	528
1151	Norway	Europe	2007	80.196	4627926	49357.190170	NOR	578
1235	Poland	Europe	2007	75.563	38518241	15389.924680	POL	616
1247	Portugal	Europe	2007	78.098	10642836	20509.647770	PRT	620
1283	Romania	Europe	2007	72.476	22276056	10808.475610	ROU	642
1343	Serbia	Europe	2007	74.002	10150265	9786.534714	SRB	688
1379	Slovak Republic	Europe	2007	74.663	5447502	18678.314350	SVK	703
1391	Slovenia	Europe	2007	77.926	2009245	25768.257590	SVN	705
1427	Spain	Europe	2007	80.941	40448191	28821.063700	ESP	724
1475	Sweden	Europe	2007	80.884	9031088	33859.748350	SWE	752
1487	Switzerland	Europe	2007	81.701	7554661	37506.419070	CHE	756
1583	Turkey	Europe	2007	71.777	71158647	8458.276384	TUR	792
1607	United Kingdom	Europe	2007	79.425	60776238	33203.261280	GBR	826

```
In [ ]: fig =px.pie(temp_df, values='pop', names='country')
fig.show("notebook")
```



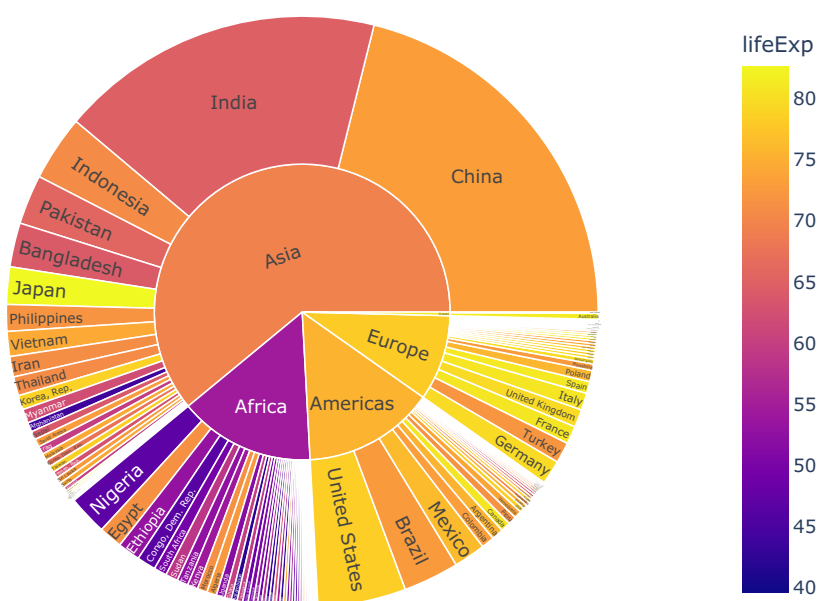
```
In [ ]: temp_df = gap[gap['year'] == 1952].groupby(
        'continent')['pop'].sum().reset_index()
fig = px.pie(temp_df, values='pop', names='continent')
fig.show("notebook")
```

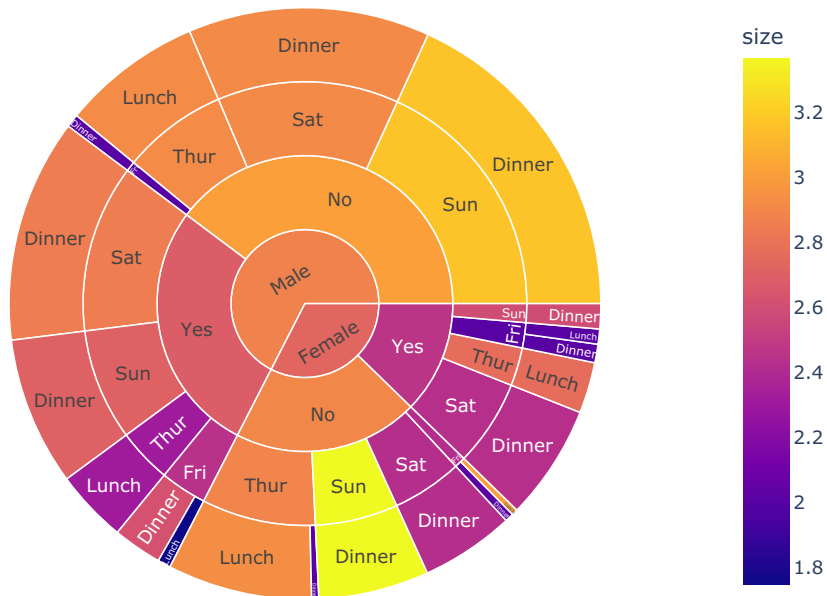


## Sunburst

```
In [ ]: # sunburst plot - sunburst plot visualize hierarchical data spanning outward radially from root to leave
temp_df = gap[gap['year'] == 2007]

fig = px.sunburst(temp_df, path=['continent', 'country'],
                  values='pop', color='lifeExp')
fig.show("notebook")
```

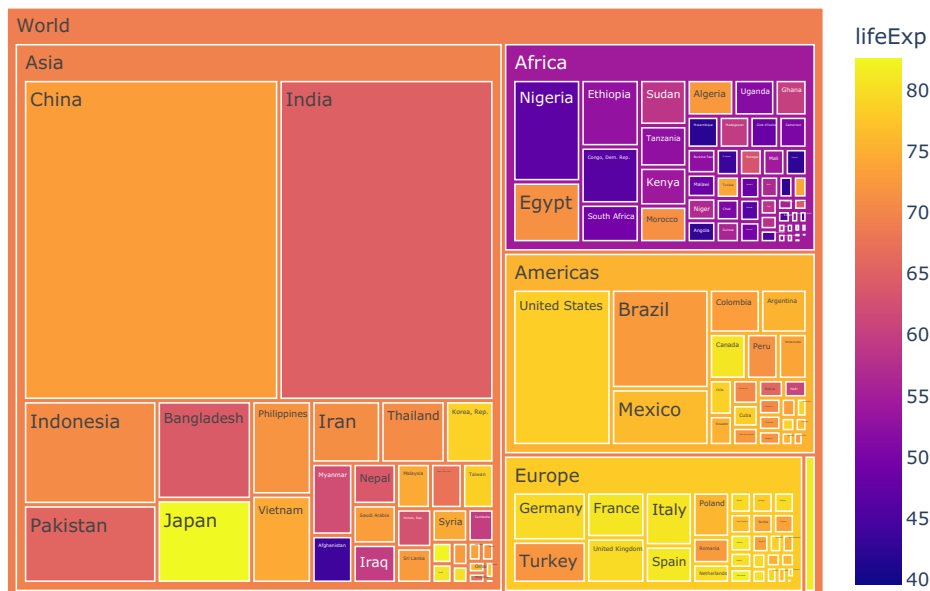




## Tree Map

```
In [ ]: temp_df = gap[gap['year'] == 2007]

fig = px.treemap(temp_df, path=[px.Constant('World'),
                                'continent', 'country'], values='pop', color='lifeExp')
fig.show("notebook")
```



## Heatmap

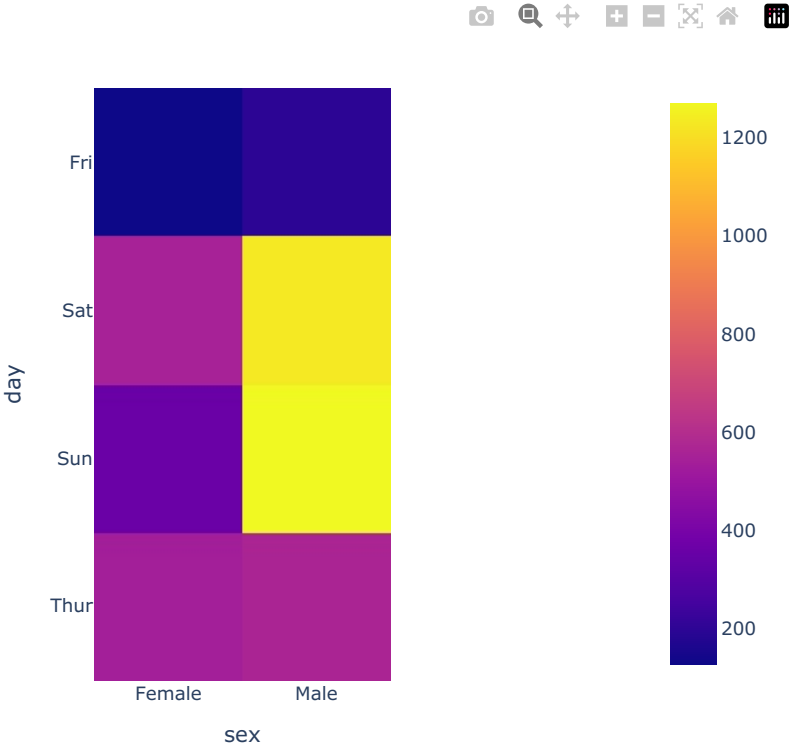
```
In [ ]: temp_df = tips.pivot_table(index='day', columns='sex',
                                    values='total_bill', aggfunc='sum')
temp_df
```



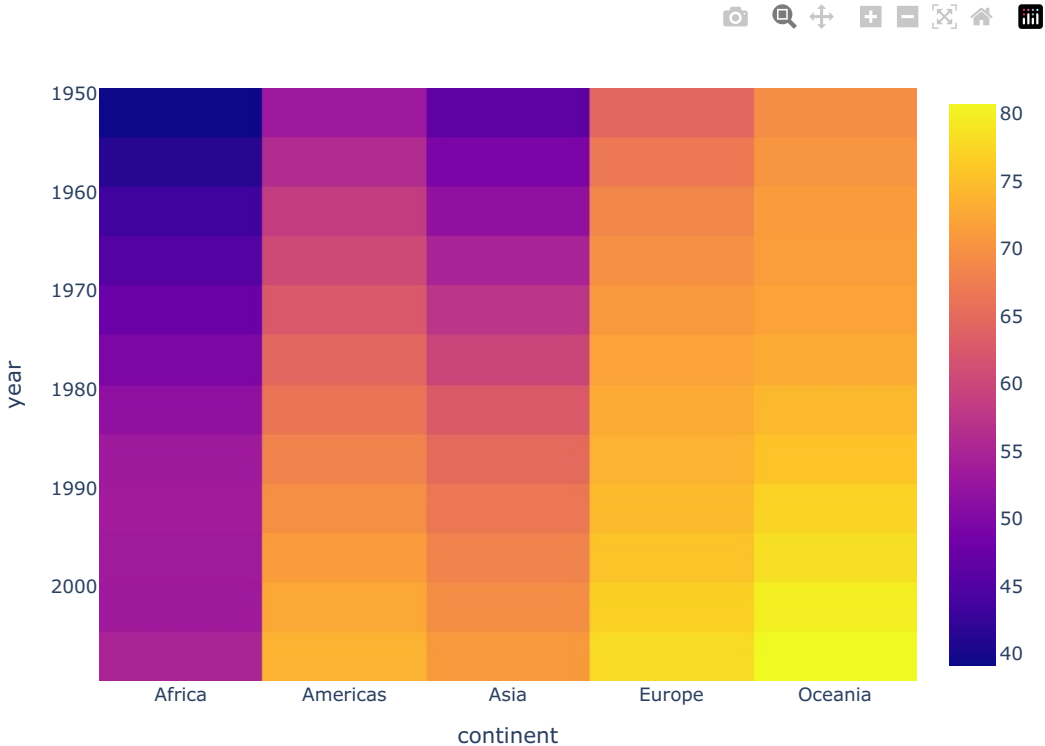
Out[ ]:

	sex	Female	Male
day			
Fri		127.31	198.57
Sat		551.05	1227.35
Sun		357.70	1269.46
Thur		534.89	561.44

```
In [ ]: fig = px.imshow(temp_df)
fig.show("notebook")
```



```
In [ ]: temp_df = gap.pivot_table(
    index='year', columns='continent', values='lifeExp', aggfunc='mean')
fig = px.imshow(temp_df)
fig.show("notebook")
```



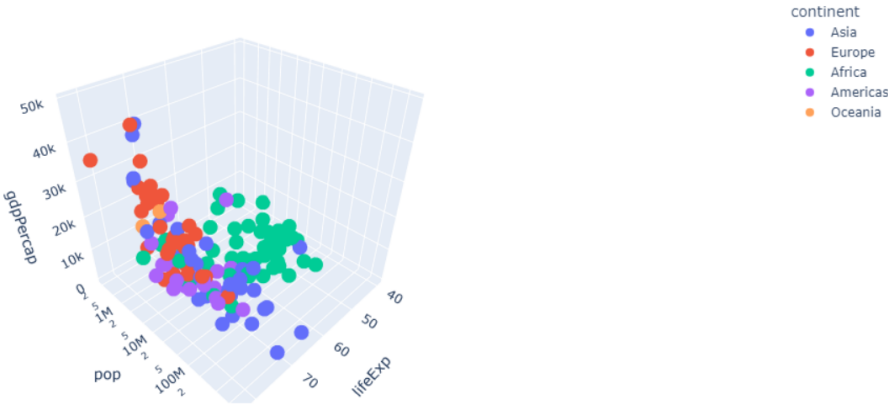
```
In [ ]: temp_df = gap[gap['year'] == 2007]
temp_df
```

Out[ ]:

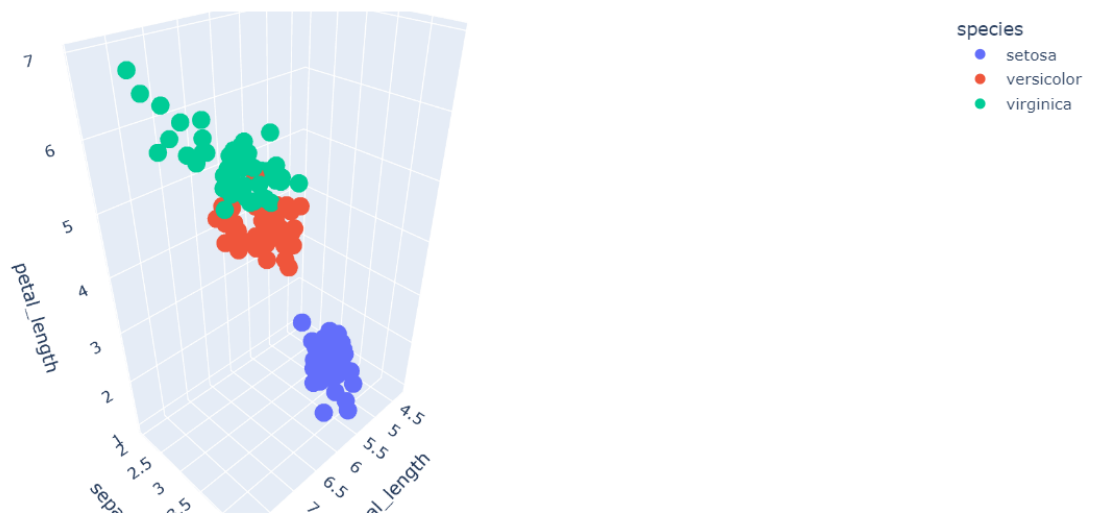
	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
11	Afghanistan	Asia	2007	43.828	31889923	974.580338	AFG	4
23	Albania	Europe	2007	76.423	3600523	5937.029526	ALB	8
35	Algeria	Africa	2007	72.301	33333216	6223.367465	DZA	12
47	Angola	Africa	2007	42.731	12420476	4797.231267	AGO	24
59	Argentina	Americas	2007	75.320	40301927	12779.379640	ARG	32
...	...	...	...	...	...	...	...	...
1655	Vietnam	Asia	2007	74.249	85262356	2441.576404	VNM	704
1667	West Bank and Gaza	Asia	2007	73.422	4018332	3025.349798	PSE	275
1679	Yemen, Rep.	Asia	2007	62.698	22211743	2280.769906	YEM	887
1691	Zambia	Africa	2007	42.384	11746035	1271.211593	ZMB	894
1703	Zimbabwe	Africa	2007	43.487	12311143	469.709298	ZWE	716

142 rows × 8 columns

```
In [ ]: fig = px.scatter_3d(temp_df, x='lifeExp', y='pop',
                           z='gdpPercap', log_y=True, color='continent', hover_name='country')
fig.show("notebook")
```



```
In [ ]: fig = px.scatter_3d(iris, x='sepal_length', y='sepal_width',
                           z='petal_length', color='species')
fig.show("notebook")
```



## Scatter metrix

```
In [ ]: iris.head(5)
```

```
Out[ ]:   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
```

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

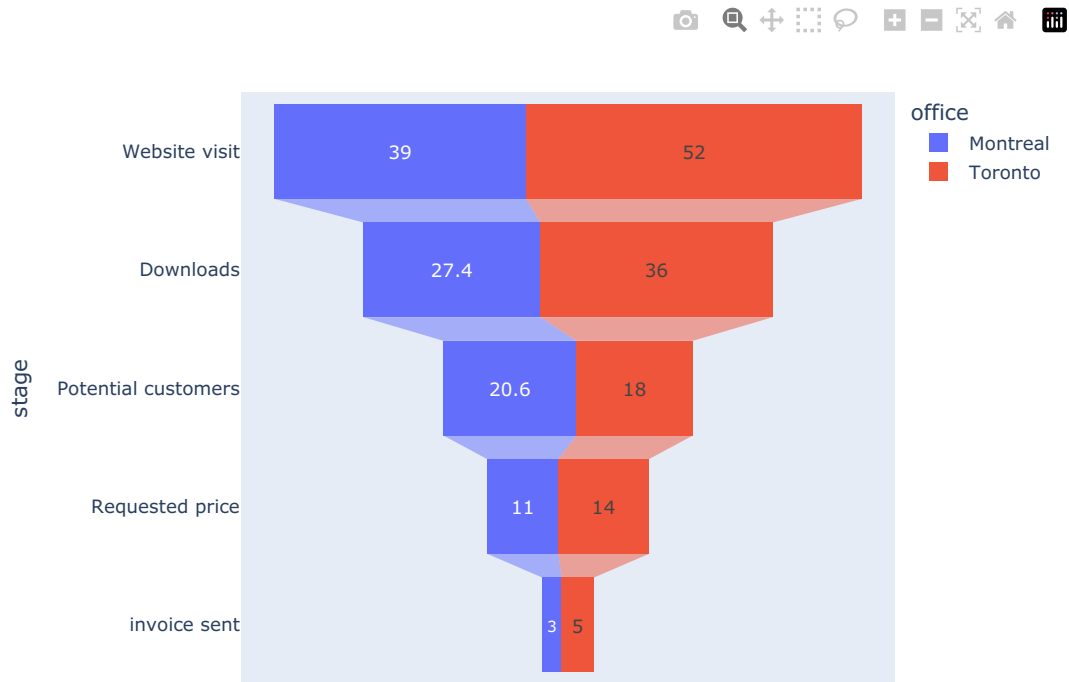
## Funnel

```
In [ ]: stages = ["Website visit", "Downloads",
                  "Potential customers", "Requested price", "invoice sent"]
df_mtl = pd.DataFrame(dict(number=[39, 27.4, 20.6, 11, 3], stage=stages))
df_mtl['office'] = 'Montreal'
df_toronto = pd.DataFrame(dict(number=[52, 36, 18, 14, 5], stage=stages))
df_toronto['office'] = 'Toronto'
df = pd.concat([df_mtl, df_toronto], axis=0)
# fig = px.funnel(df, x='number', y='stage', color='office')
# fig.show()
```

```
In [ ]: df
```

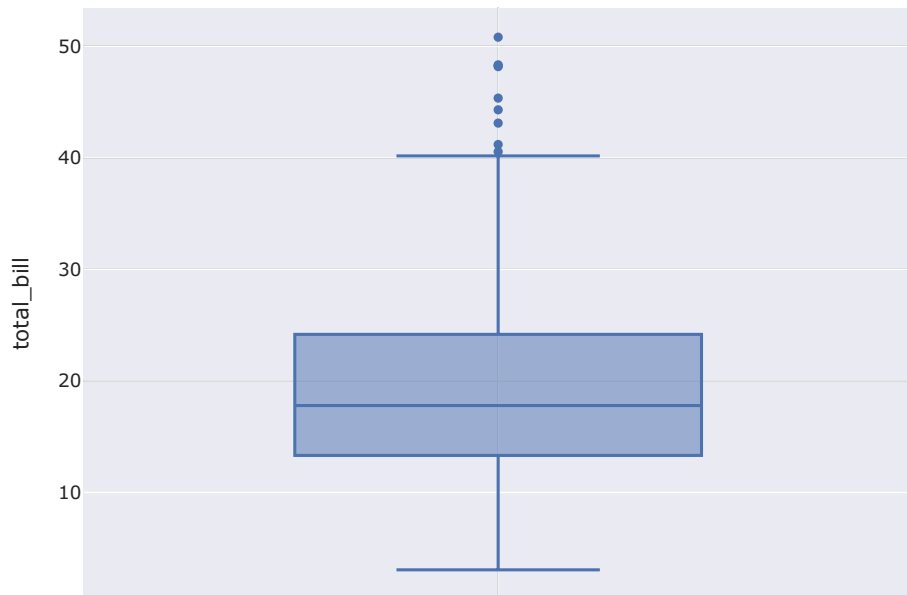
```
Out[ ]:   number  stage  office
0    39.0  Website visit  Montreal
1    27.4    Downloads  Montreal
2    20.6  Potential customers  Montreal
3    11.0  Requested price  Montreal
4     3.0    invoice sent  Montreal
0    52.0  Website visit   Toronto
1    36.0    Downloads   Toronto
2    18.0  Potential customers   Toronto
3    14.0  Requested price   Toronto
4     5.0    invoice sent   Toronto
```

```
In [ ]: fig = px.funnel(df, x='number', y='stage', color='office')
fig.show("notebook")
```

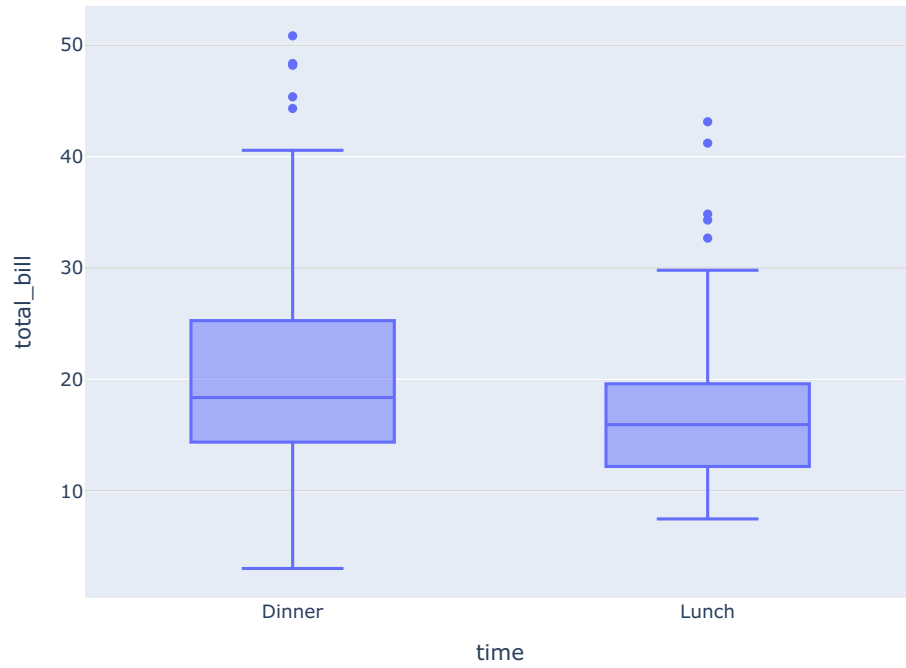


## Box

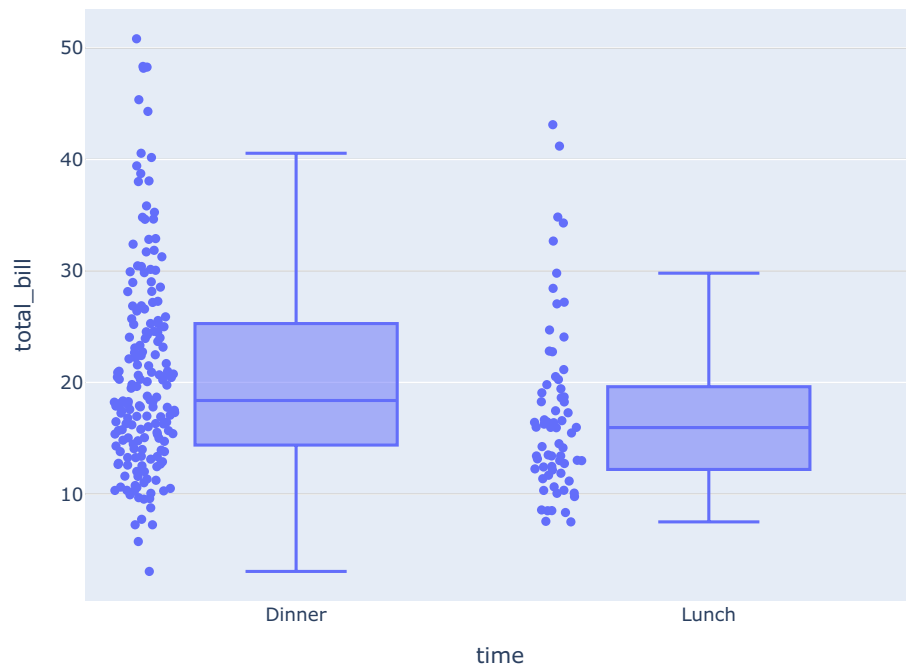
```
In [ ]: fig = px.box(tips, y='total_bill', template='seaborn')
fig.show("notebook")
```



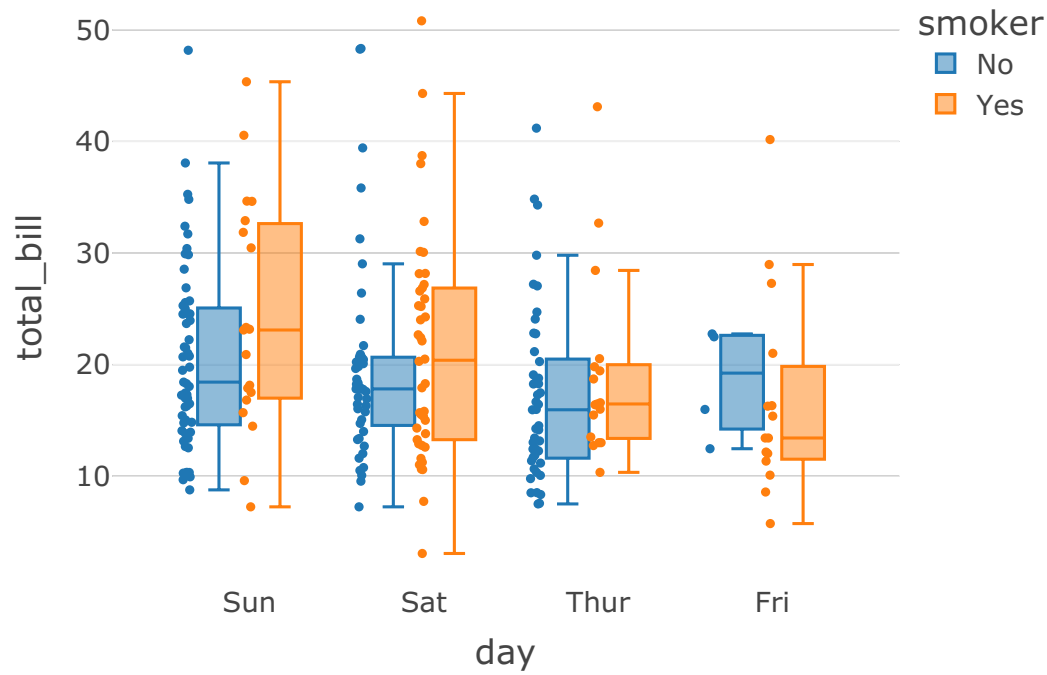
```
In [ ]: fig = px.box(tips, x='time', y='total_bill')
fig.show("notebook")
```



```
In [ ]: fig = px.box(tips, x='time', y='total_bill', points='all')
fig.show("notebook")
```

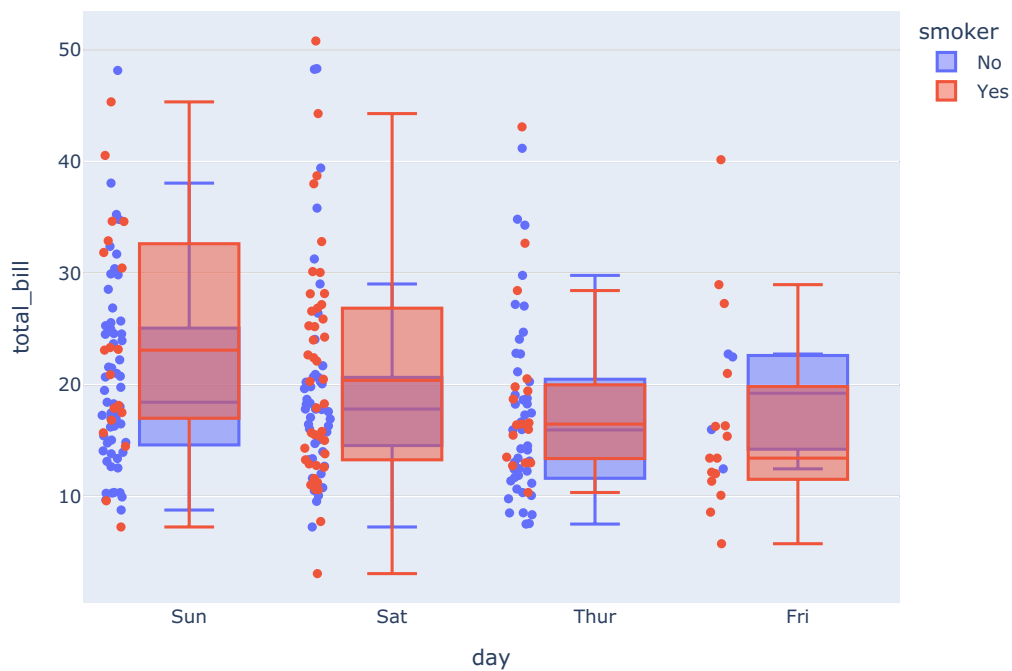


```
In [ ]: fig = px.box(tips, x='day', y='total_bill', color='smoker',
                    points='all', template='presentation')
fig.show("notebook")
```



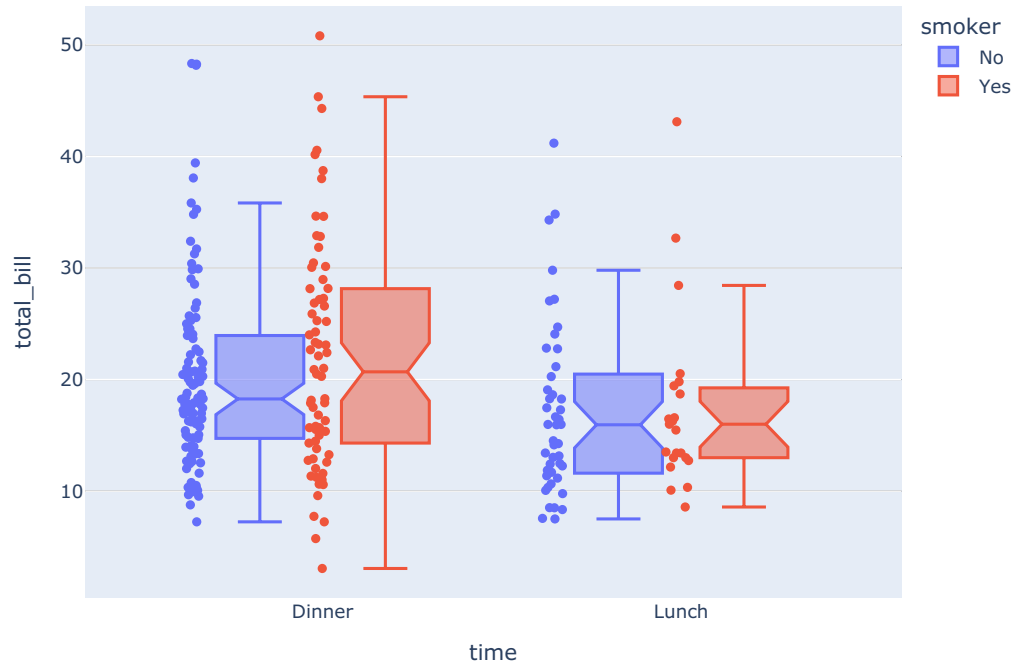
```
In [ ]: fig = px.box(tips, x='day', y='total_bill', color='smoker',
                  points='all', boxmode='overlay')

fig.show("notebook")
```



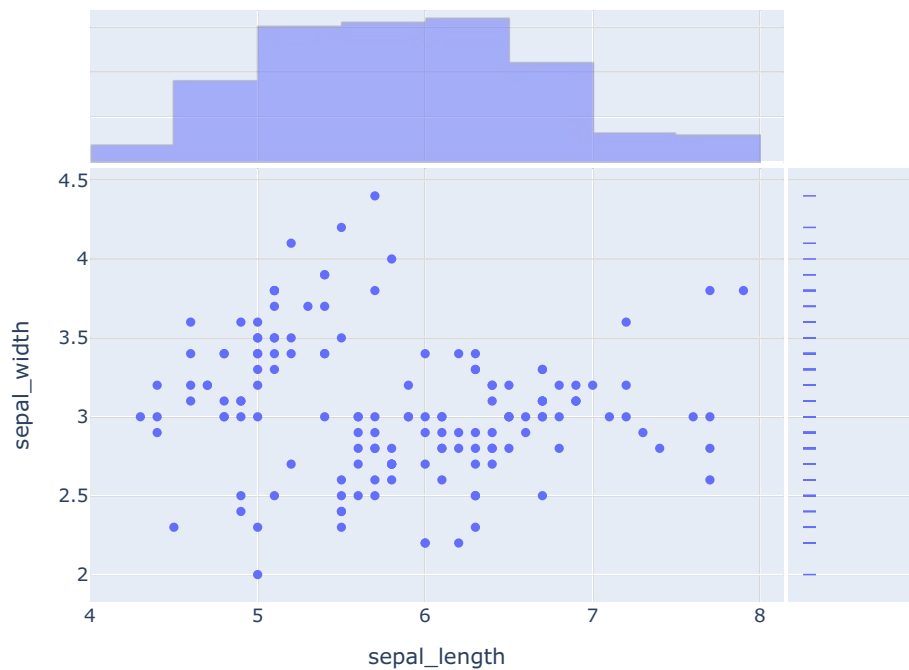
```
In [ ]: fig = px.box(tips, x='time', y='total_bill', color='smoker',
                  notched=True, points='all',
                  hover_data=['day'])

fig.show("notebook")
```



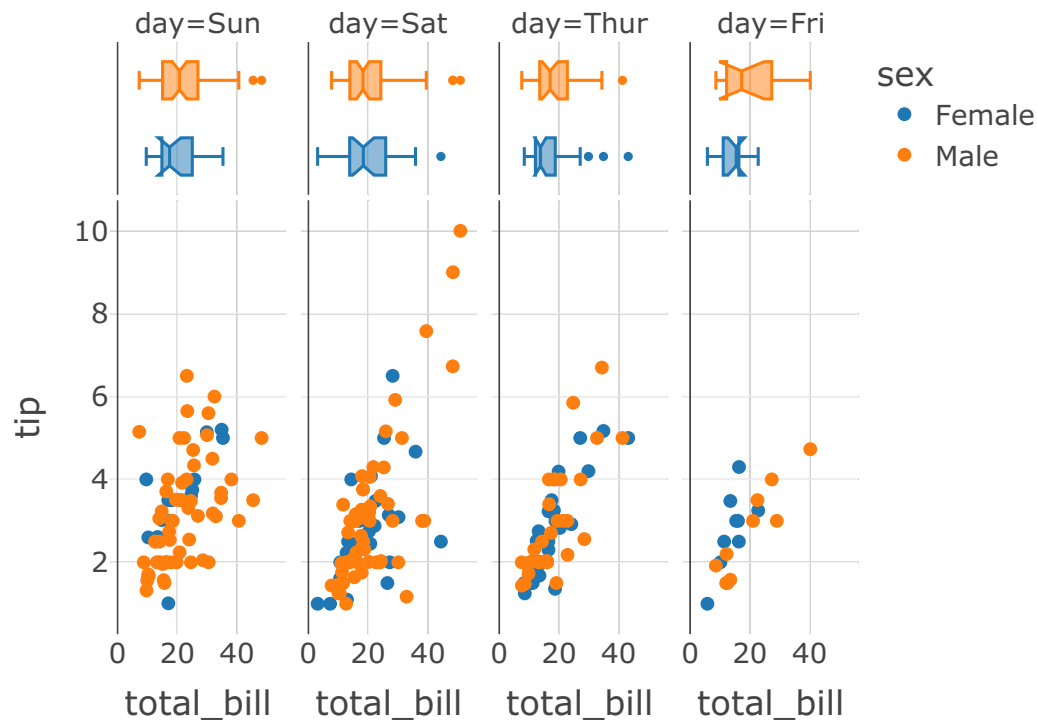
## Scatter Plot Marginals

```
In [ ]: fig = px.scatter(iris, x="sepal_length", y="sepal_width",
                        marginal_x="histogram", marginal_y="rug")
fig.show("notebook")
```

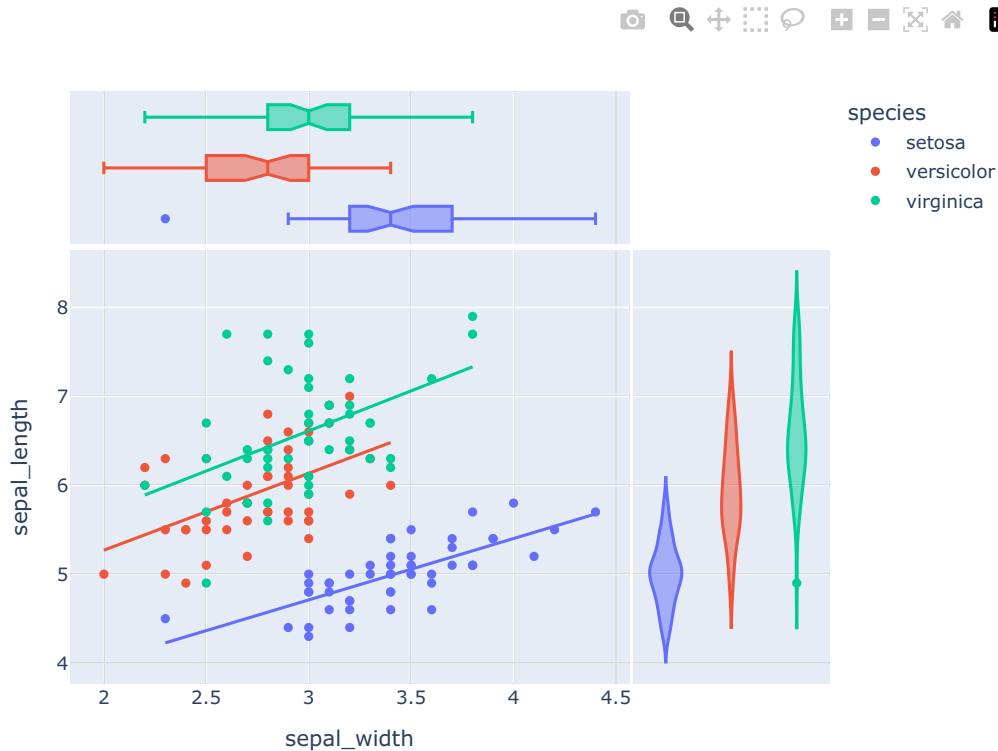


## Marginal Plots and Facets

```
In [ ]: fig = px.scatter(tips, x="total_bill", y="tip", color="sex", facet_col="day",
                        marginal_x="box", template='presentation')
fig.show("notebook")
```



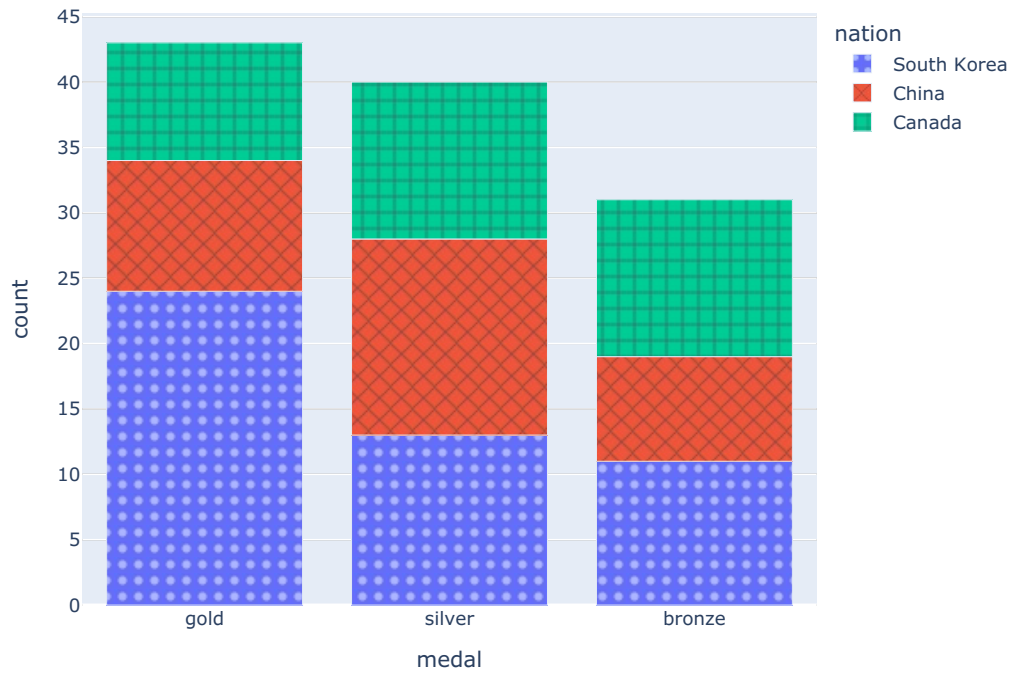
```
In [ ]: fig = px.scatter(iris, x="sepal_width", y="sepal_length", color="species", marginal_y="violin",
                      marginal_x="box", trendline="ols")
fig.show("notebook")
```



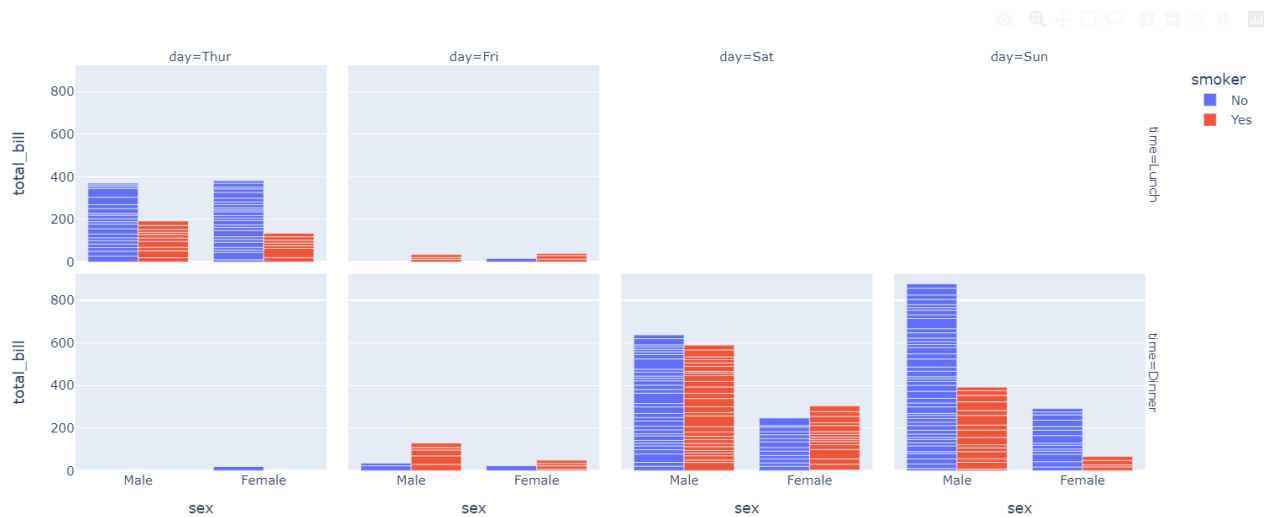
```
In [ ]: df = px.data.medals_long()

fig = px.bar(df, x="medal", y="count", color="nation",
             pattern_shape="nation", pattern_shape_sequence=[".", "x", "+"])
fig.show("notebook")
```



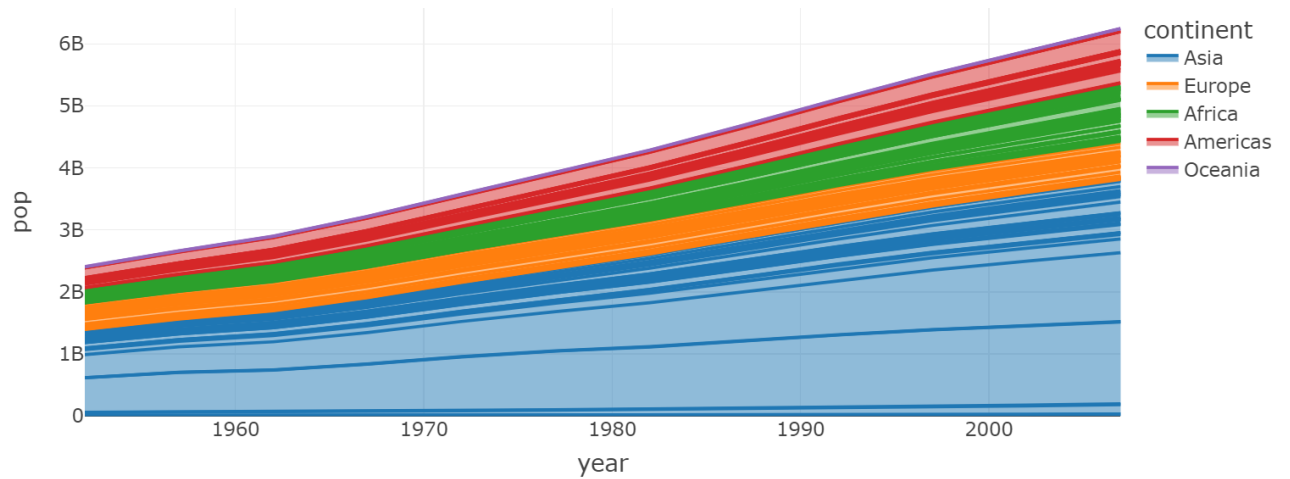


```
In [ ]: fig = px.bar(tips, x="sex", y="total_bill", color="smoker", barmode="group", facet_row="time", facet_col="day",
                  category_orders={
                      "day": ["Thur", "Fri", "Sat", "Sun"], "time": ["Lunch", "Dinner"]}
                  )
fig.show("notebook")
```



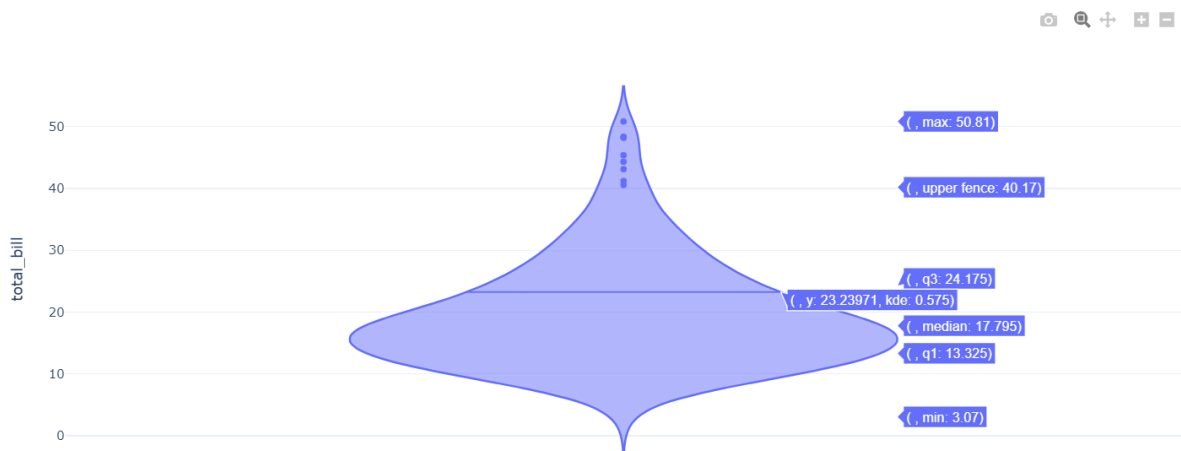
## Filled Area

```
In [ ]: fig = px.area(gap, x="year", y="pop", color="continent",
                    line_group="country", template='presentation')
fig.show("notebook")
```

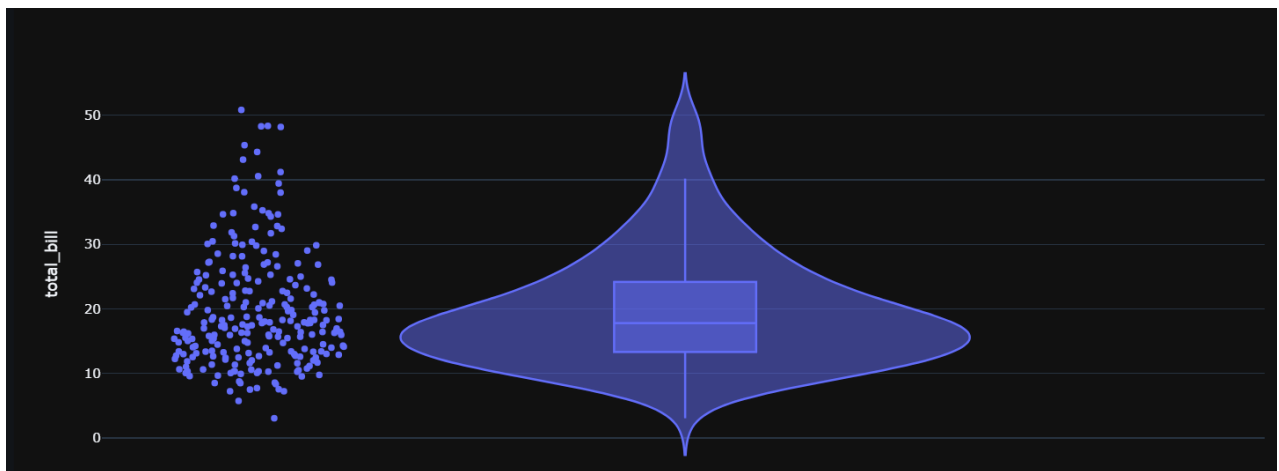


## Violin Plots

```
In [ ]: fig = px.violin(tips, y="total_bill", template='plotly_white')
fig.show("notebook")
```



```
In [ ]: fig = px.violin(tips, y="total_bill", box=True,
                        points='all', template='plotly_dark')
fig.show("notebook")
```



```
In [ ]: fig = px.scatter(gap, x="gdpPercap", y="lifeExp", animation_frame="year", animation_group="country",
                        size="pop", color="continent", hover_name="country", facet_col="continent",
                        log_x=True, size_max=45, range_x=[100, 100000], range_y=[25, 90])
fig.show("notebook")
```

