Plotly 소개

d3.js를 이용하여 interactive하게 그래프를 보여준다.

사전 파이썬 버전 확인

- (base) C:\Users\toto>python --version
- Python 3.7.7

plotly를 pandas와 함께 사용하는 법

- cufflinks 설정과 .iplot()을 활용. pandas.plot()와 같이 판다스 데이터 시각화
- plotly.express 라이브러리 활용

cuffilinks 는 무엇

- 판다스 데이터 프레임과 Plotly를 연결하여 사용자가 판다스로부터 직접 시각화를 할 수 있는 라이브러리

01 시작하기 - 설치(Plotly and Cufflinks)

- pip install plotly
- pip install cufflinks

In [1]:

```
import plotly
import cufflinks as cf
import pandas as pd
import numpy as np
```

버전 확인

In [2]:

```
print(plotly.__version__)
print(cf.__version__)
print(pd.__version__)
print(np.__version__)
```

```
4.11.0
0.17.3
0.24.2
1.16.2
```

In [3]:

```
# 오프라인 모드에서도 인터렉티브한 그래픽을 가능하도록 하기
#Enabling the offline mode for interactive plotting locally
from plotly.offline import download_plotlyjs,init_notebook_mode,plot,iplot
init_notebook_mode(connected=True)
cf.go_offline()
```

데이터 생성 및 plot

In [4]:

```
#create Data
df = pd.DataFrame(np.random.randn(100,4), # 100개 4개 컬럼
 columns='A B C D'.split())
print(df.shape)
df.head()
```

```
(100, 4)
```

Out[4]:

	A	B	C	D
0	0.311031	-0.972142	0.445374	0.154822
1	0.879597	-0.061898	1.184329	-1.305192
2	0.778040	2.023429	-2.731564	0.845717
3	-1.888249	-2.398249	-0.999591	-0.760878
4	0.098424	1.194148	1.371483	0.757410

In [5]:

```
df2 = pd.DataFrame({'items':['bag','apple','cap'],'Values':[32,43,50,]})
df2
```

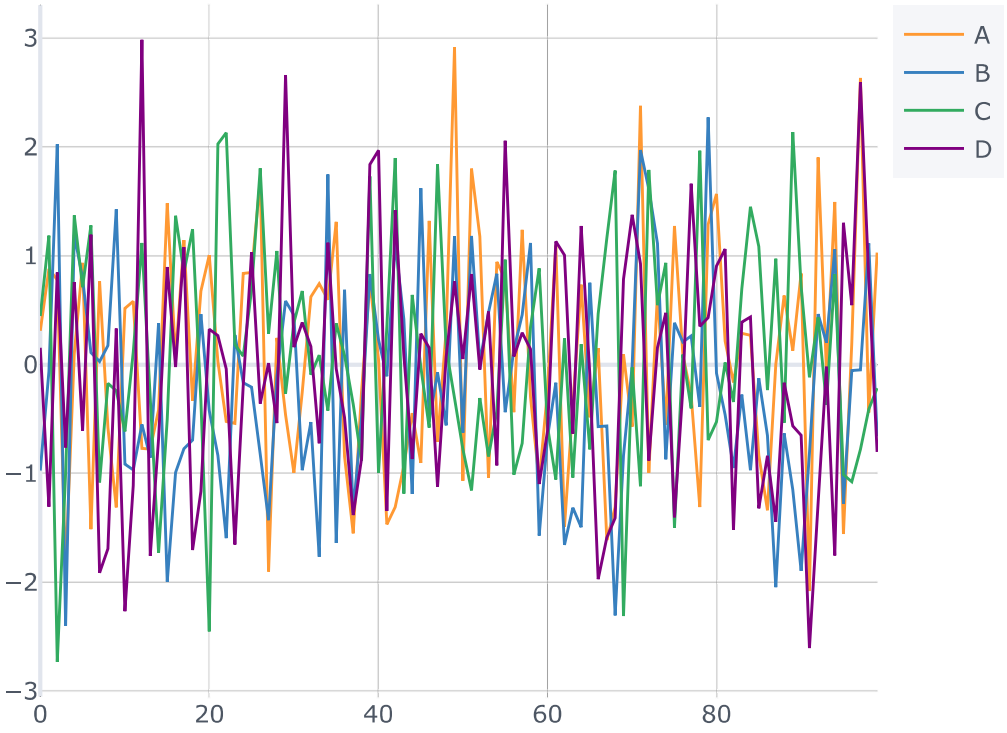
Out[5]:

	items	Values
0	bag	32
1	apple	43
2	cap	50

Line Plot

In [6]:

```
df.iplot()
```

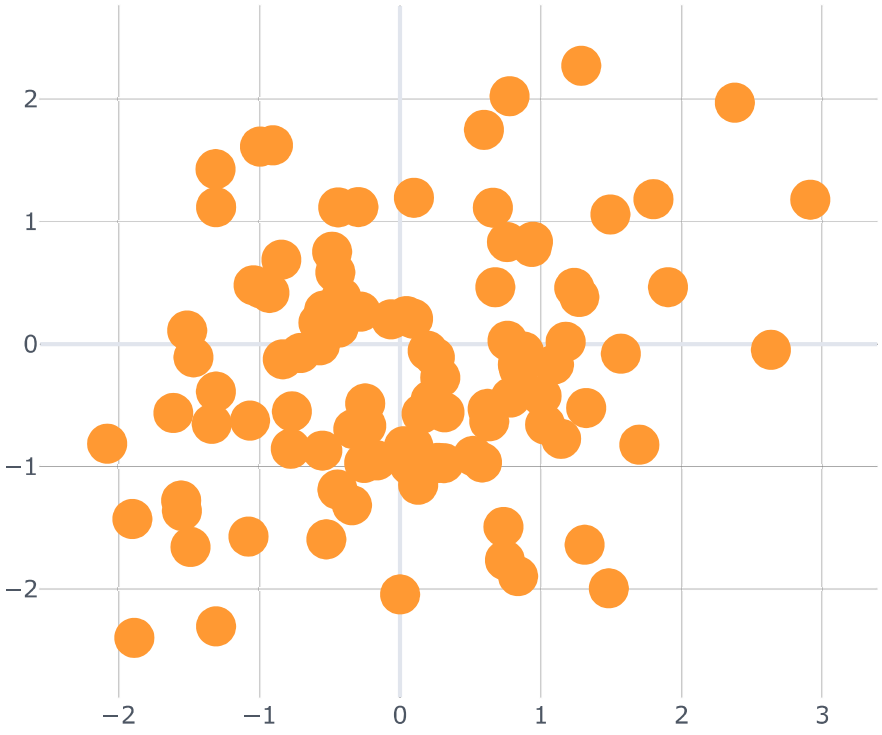


Export to plot.ly »

Scatter Plot

In [7]:

```
df.iplot(kind='scatter', x='A',y='B',mode='markers',size=20)
```

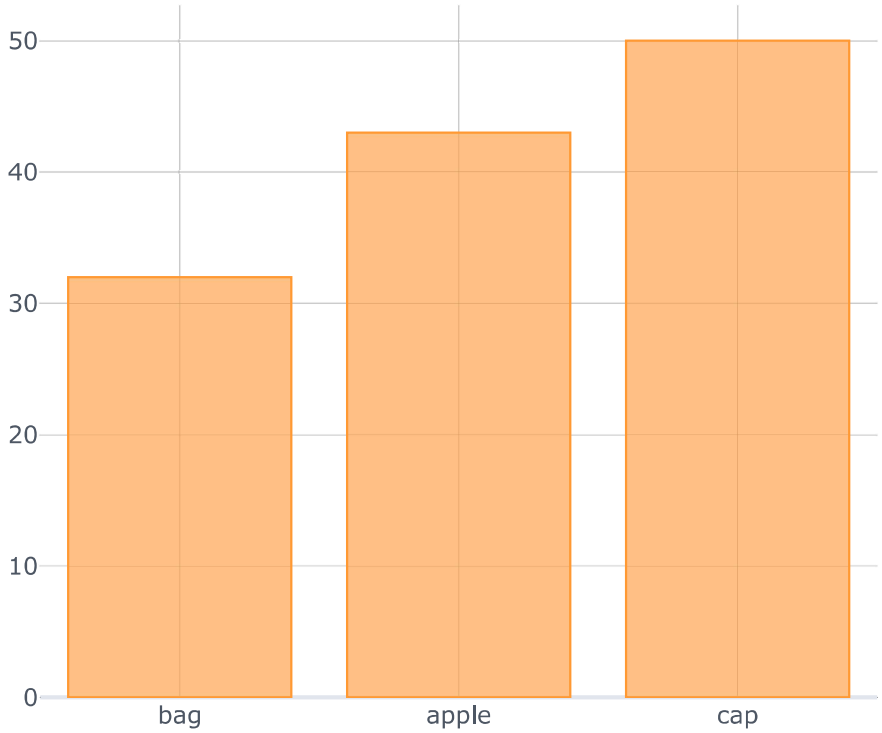


Export to plot.ly »

Bar Plot

In [8]:

```
df2.iplot(kind='bar',x='items',y='Values')
```



Export to plot.ly »

In [9]:

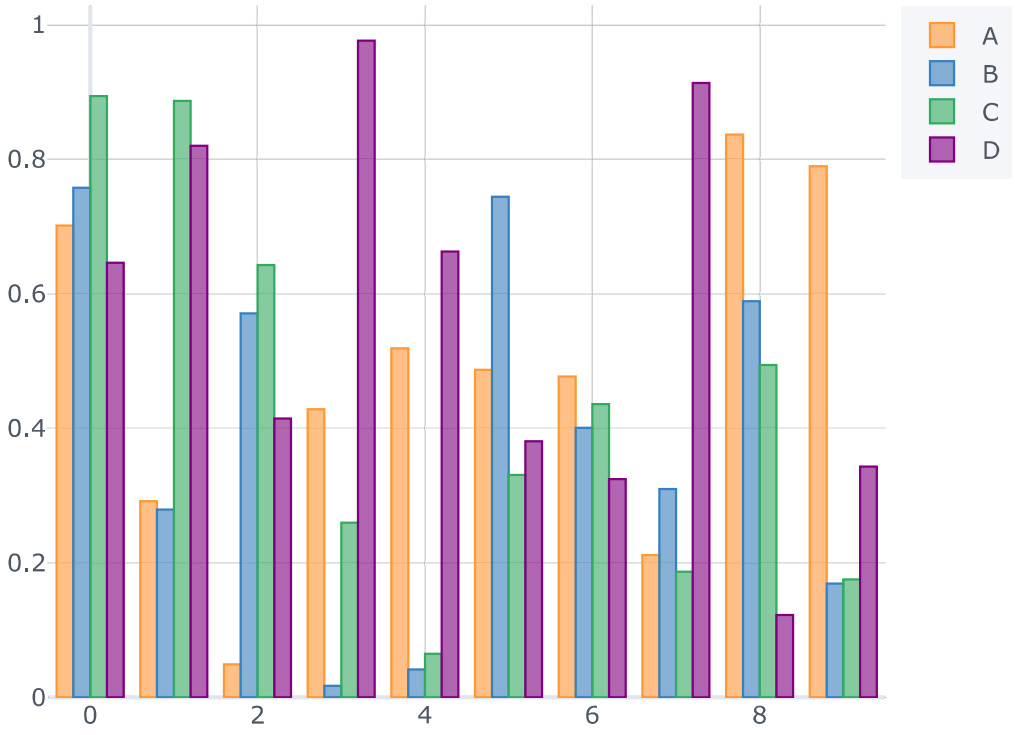
```
df = pd.DataFrame(np.random.rand(10,4),
                  columns=['A', 'B', 'C', 'D'])
df.head()
```

Out[9]:

	A	B	C	D
0	0.701231	0.757371	0.893604	0.646171
1	0.291714	0.279102	0.886785	0.820057
2	0.049156	0.570756	0.642743	0.414603
3	0.428128	0.017398	0.259880	0.976035
4	0.519040	0.041507	0.064945	0.662900

In [10]:

```
df.iplot(kind='bar')
```

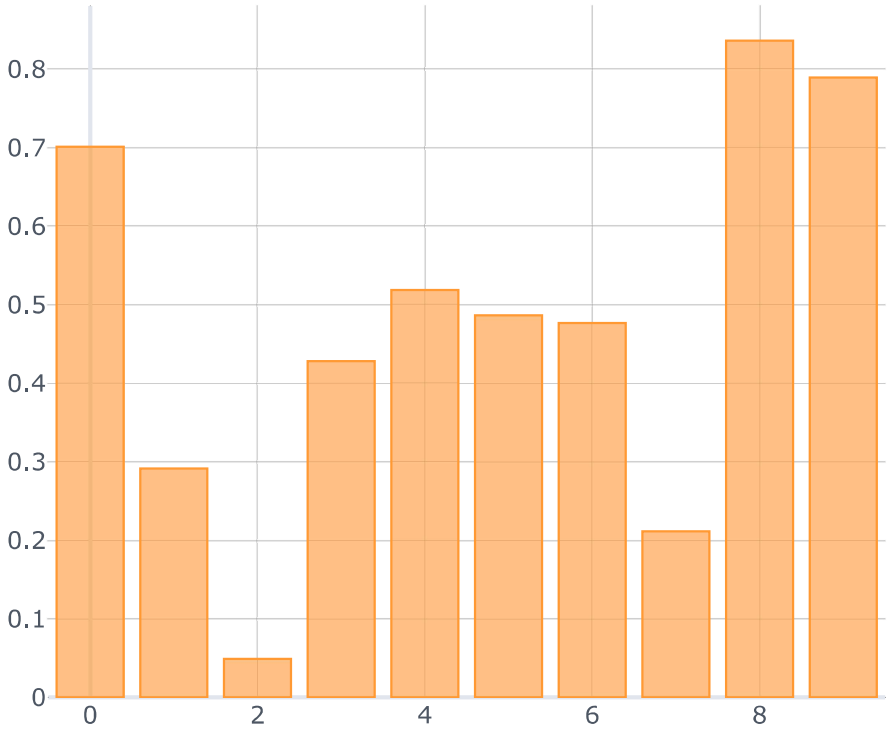


Export to plot.ly »

A컬럼만 보기

In [11]:

```
df['A'].iplot(kind='bar')
```

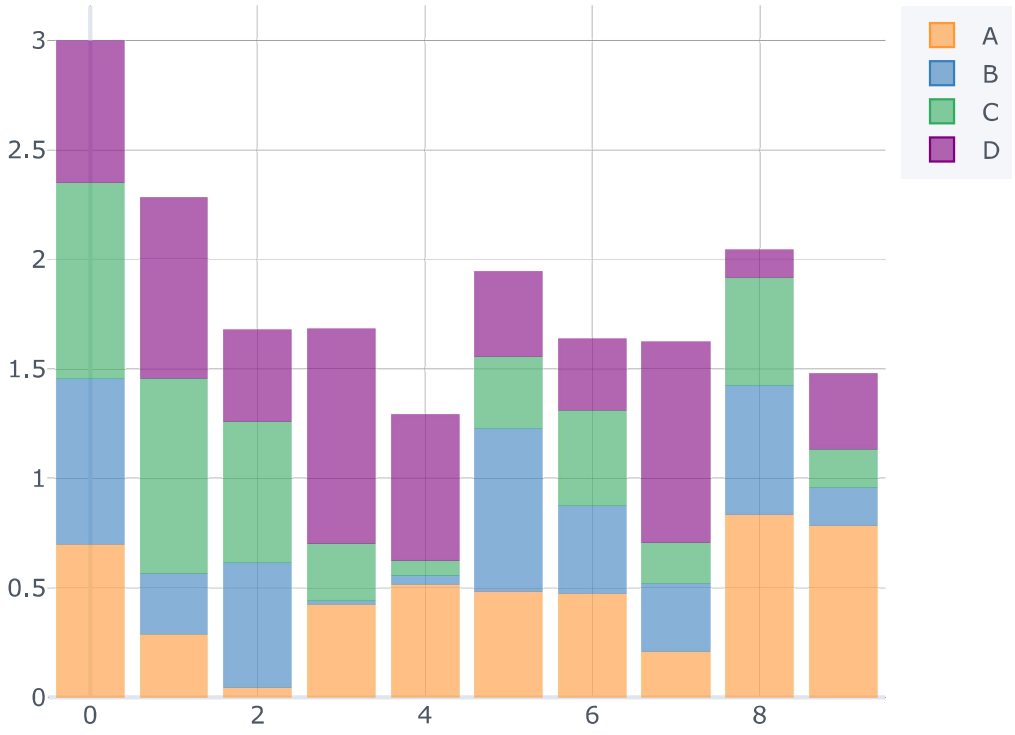


Export to plot.ly »

Stack plot

In [12]:

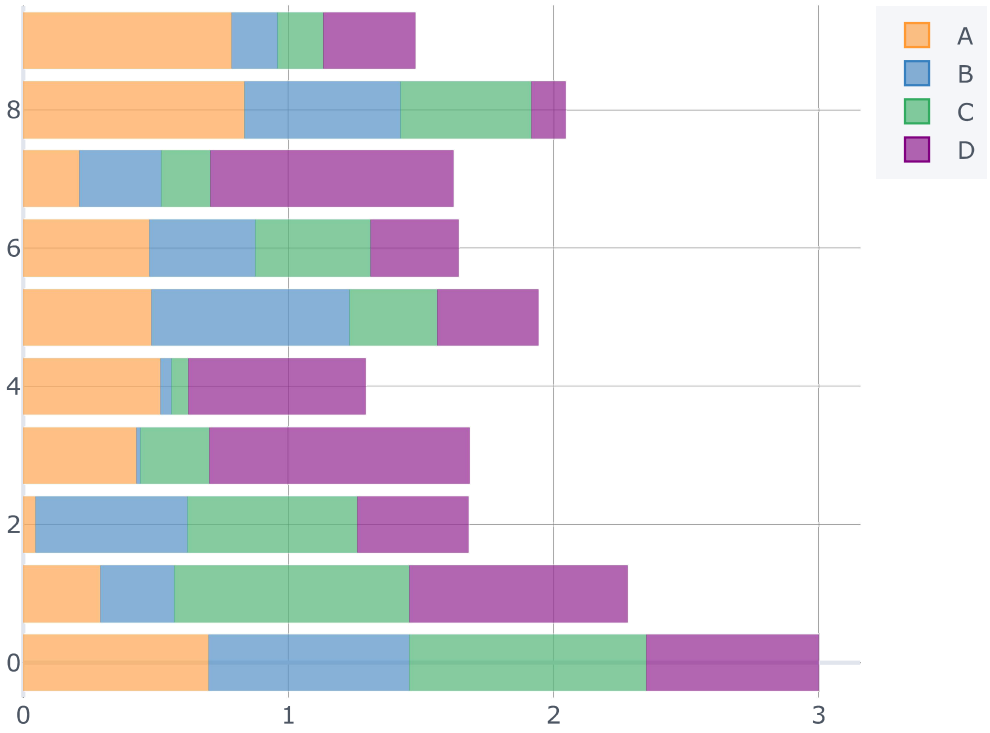
```
df.iplot(kind='bar', barmode='stack')
```



Export to plot.ly »

In [13]:

```
df.iplot(kind='barh', barmode='stack')
```

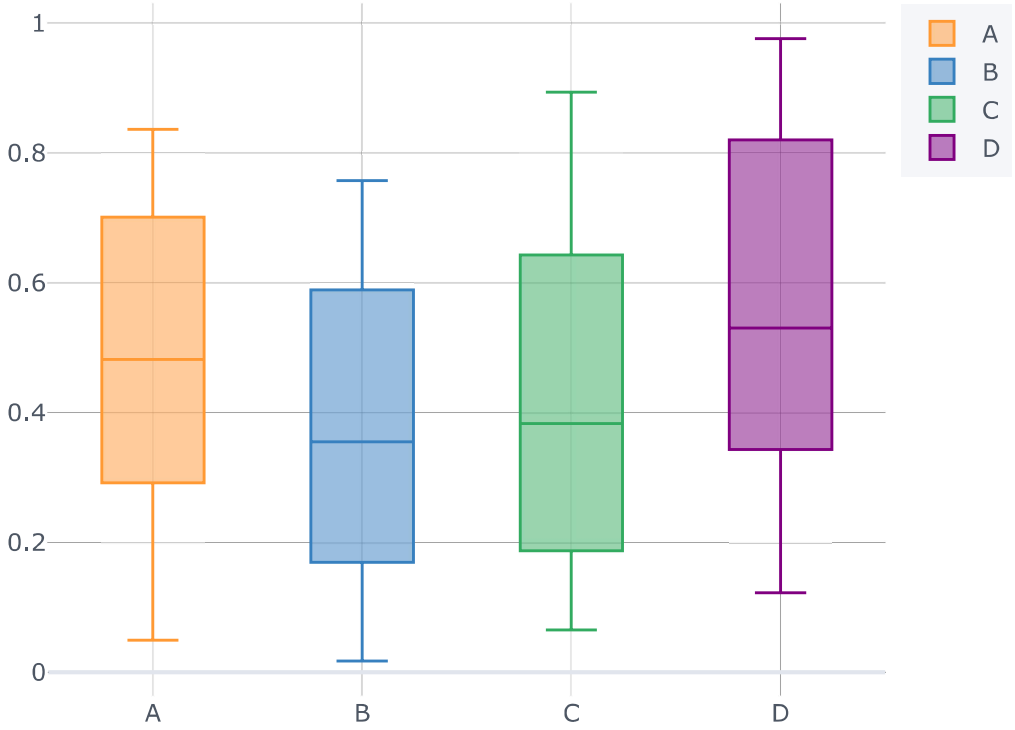


Export to plot.ly »

Boxplot

In [14]:

```
df.iplot(kind='box')
```



Export to plot.ly »

3D Surface Plot

In [15]:

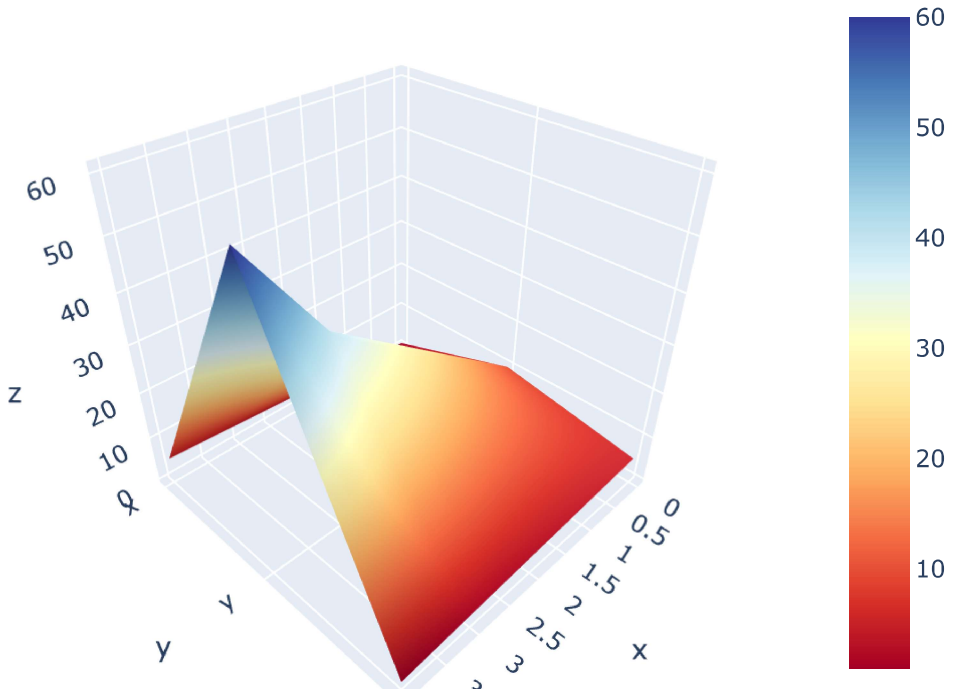
```
df3 = pd.DataFrame({'x':[1,2,3,4,5],
 'y':[10,20,30,40,60],
 'z':[5,4,3,2,1]})
df3
```

Out[15]:

	x	y	z
0	1	10	5
1	2	20	4
2	3	30	3
3	4	40	2
4	5	60	1

In [16]:

```
df3.iplot(kind='surface',colorscale='rdylbu')
```



Export to plot.ly »

Line Charts

In [17]:

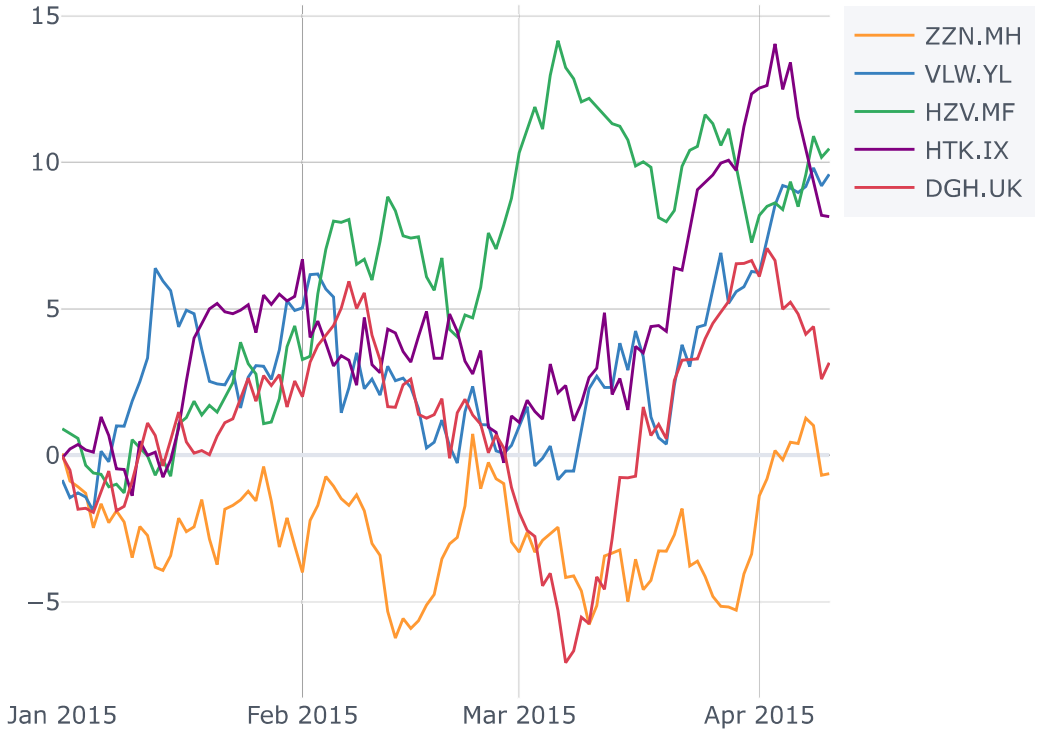
```
df = cf.datagen.lines()
df.head()
```

Out[17]:

	ZZN.MH	VLW.YL	HZV.MF	HTK.IX	DGH.UK
2015-01-01	0.060297	-0.846363	0.896352	-0.096500	-0.039194
2015-01-02	-0.880924	-1.451179	0.731229	0.209789	-0.508013
2015-01-03	-1.077331	-1.284934	0.576580	0.363297	-1.844349
2015-01-04	-1.298957	-1.432832	-0.350328	0.174084	-1.810260
2015-01-05	-2.475317	-1.912503	-0.612127	0.105041	-1.950671

In [18]:

```
df.iplot(kind='line')
```



Export to plot.ly »

In [19]:

```
print(df.shape)
df.head(10)
```

(100, 5)

Out[19]:

	ZZN.MH	VLW.YL	HZV.MF	HTK.IX	DGH.UK
2015-01-01	0.060297	-0.846363	0.896352	-0.096500	-0.039194
2015-01-02	-0.880924	-1.451179	0.731229	0.209789	-0.508013
2015-01-03	-1.077331	-1.284934	0.576580	0.363297	-1.844349
2015-01-04	-1.298957	-1.432832	-0.350328	0.174084	-1.810260
2015-01-05	-2.475317	-1.912503	-0.612127	0.105041	-1.950671
2015-01-06	-1.663735	0.139910	-0.651358	1.305280	-1.253633
2015-01-07	-2.308034	-0.223083	-1.075425	0.666434	-0.548868
2015-01-08	-1.894674	0.995168	-0.991331	-0.464184	-1.902325
2015-01-09	-2.279705	0.992705	-1.290524	-0.493019	-1.749270
2015-01-10	-3.497697	1.844919	0.528122	-1.388571	-0.881181

테마설정

In [20]:

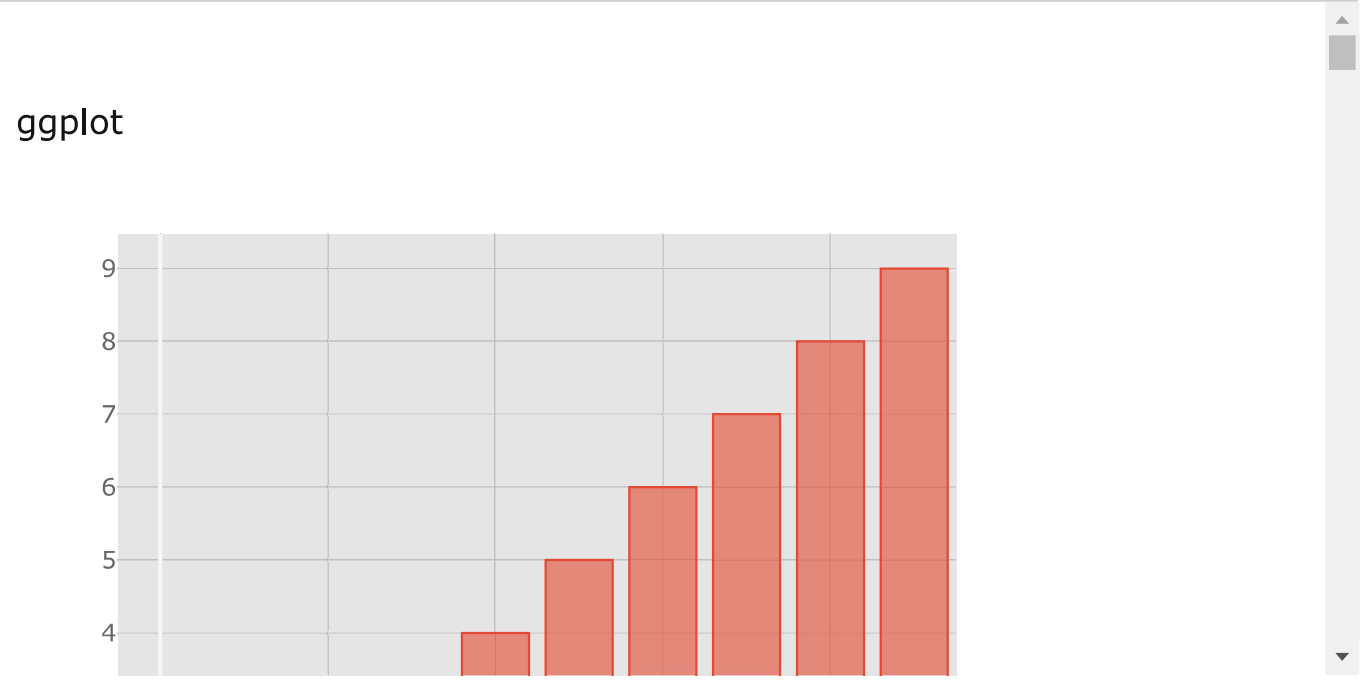
```
themes = cf.getThemes()
themes
```

Out[20]:

```
['ggplot', 'pearl', 'solar', 'space', 'white', 'polar', 'henanigans']
```

In [21]:

```
data = pd.Series(range(10))
for theme in themes:
 data.iplot(kind='bar', theme=theme, title=theme)
```



In [22]:

```
cf.set_config_file(theme='pearl')
```

Plot express 사용한 시각화

- cufflinks보다 좀 더 다양하며, 사용방법은 seaborn과 비슷함.
- plotly_express 이용. plotly 4.1 부터는 별도 설치 없어도 됨. 3.8.1의 경우 설치 필요

In [23]:

```
import plotly.express as px
```

In [24]:

```
# iris 데이터 불러오기
print(px.data.iris.__doc__)
px.data.iris().head()
```

```
Each row represents a flower.

https://en.wikipedia.org/wiki/Iris_flower_data_set (https://en.wikipedia.org/wiki/Ir
is_flower_data_set)

Returns:
    A `pandas.DataFrame` with 150 rows and the following columns:
    `['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'species', 'spec
ies_id']`.
```

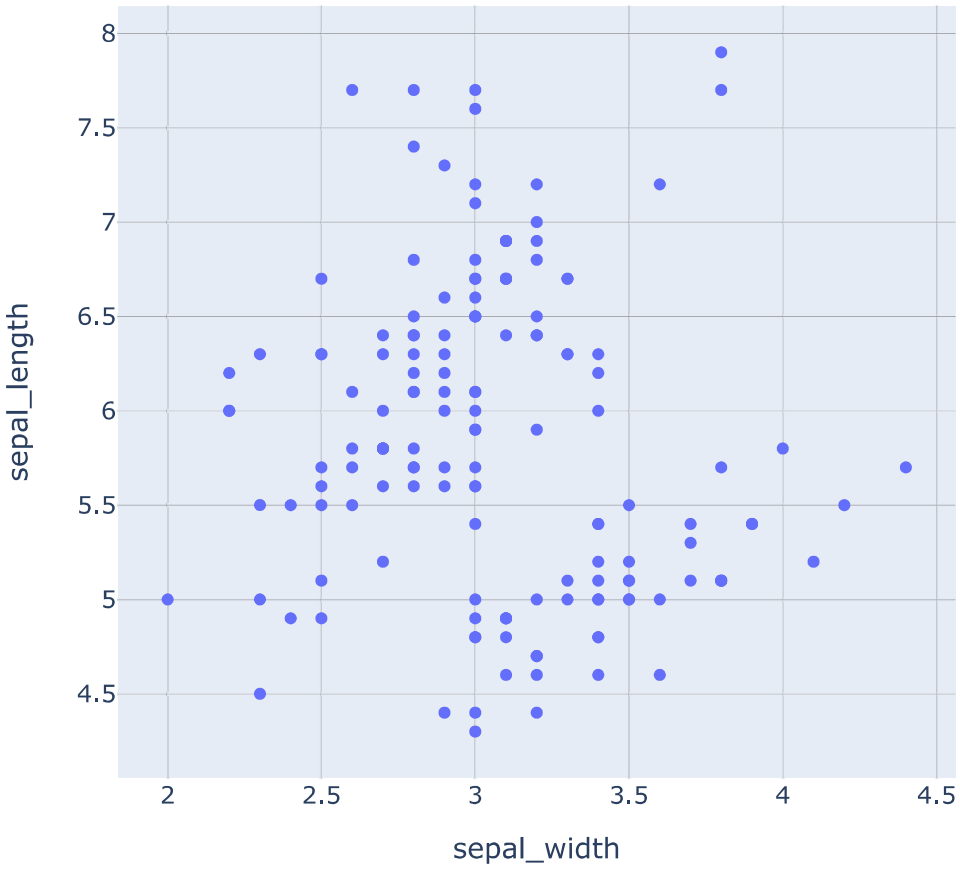
Out[24]:

	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

산점도 및 선 그래프

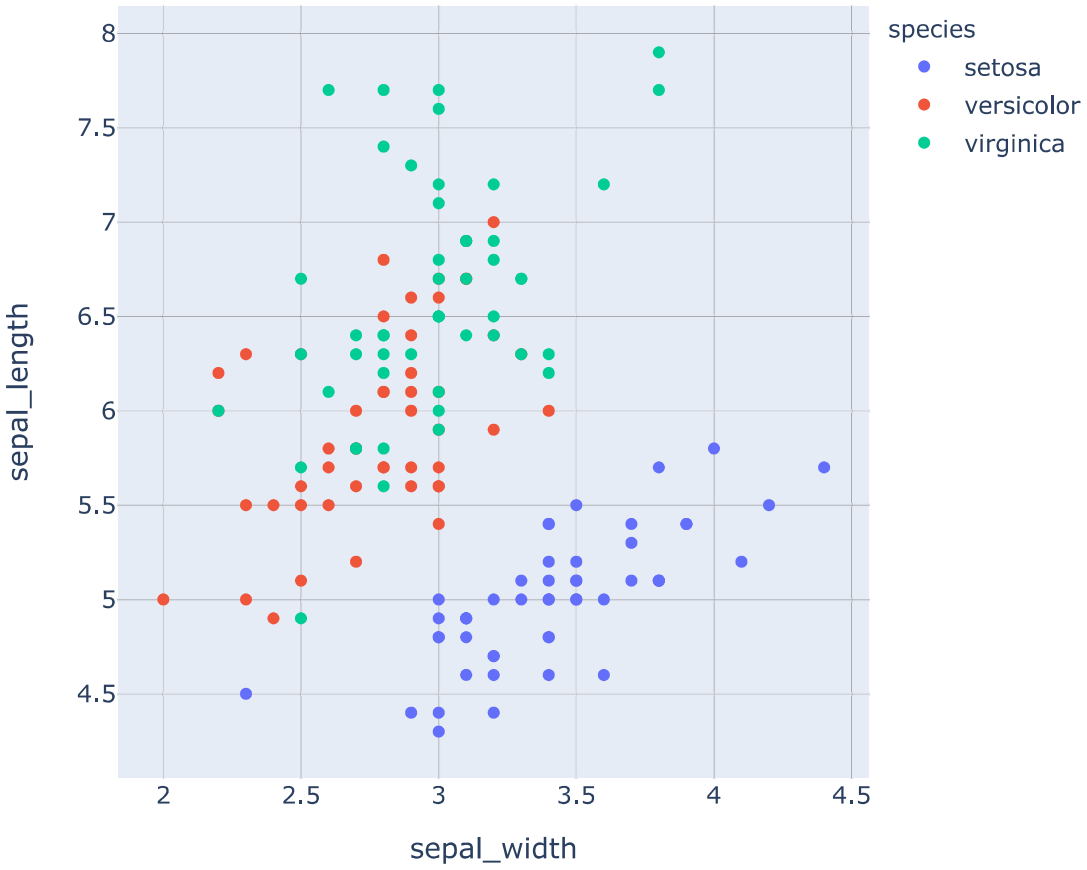
In [25]:

```
import plotly.express as px
df = px.data.iris()
fig = px.scatter(df, x="sepal_width", y="sepal_length")
fig.show()
```



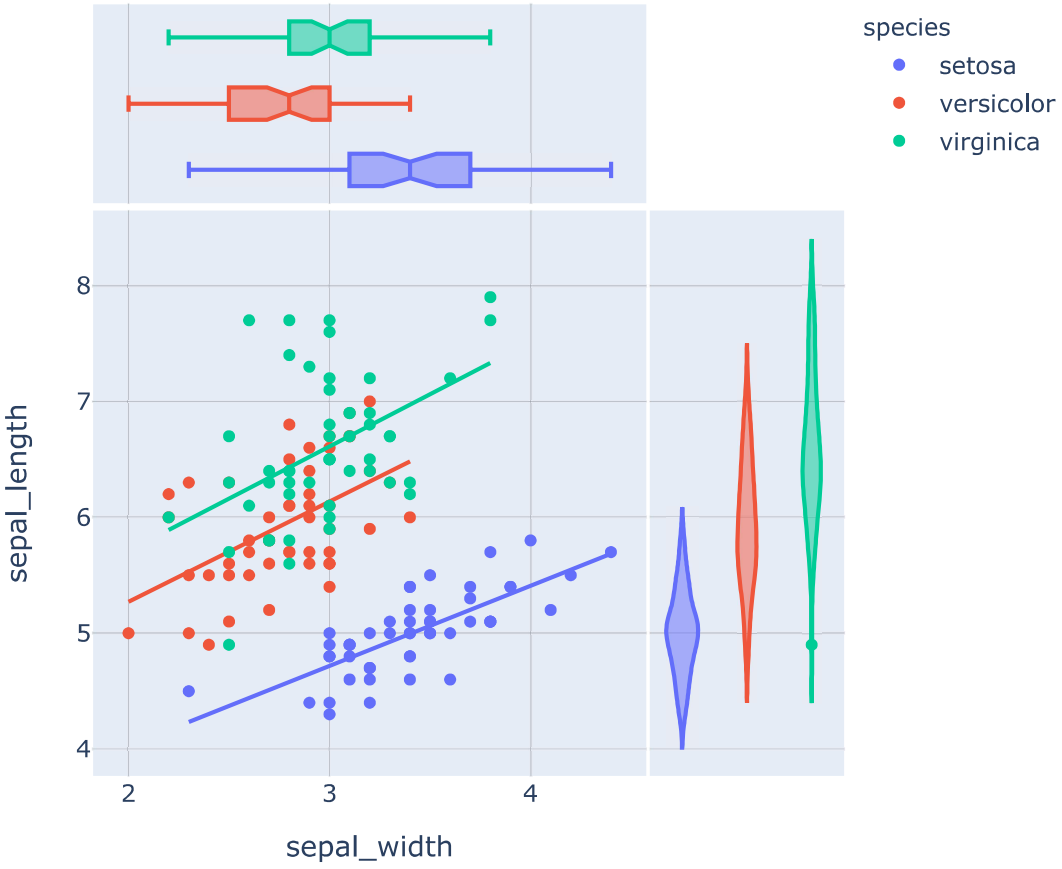
In [26]:

```
import plotly.express as px
df = px.data.iris()
fig = px.scatter(df, x="sepal_width", y="sepal_length", color="species")
fig.show()
```



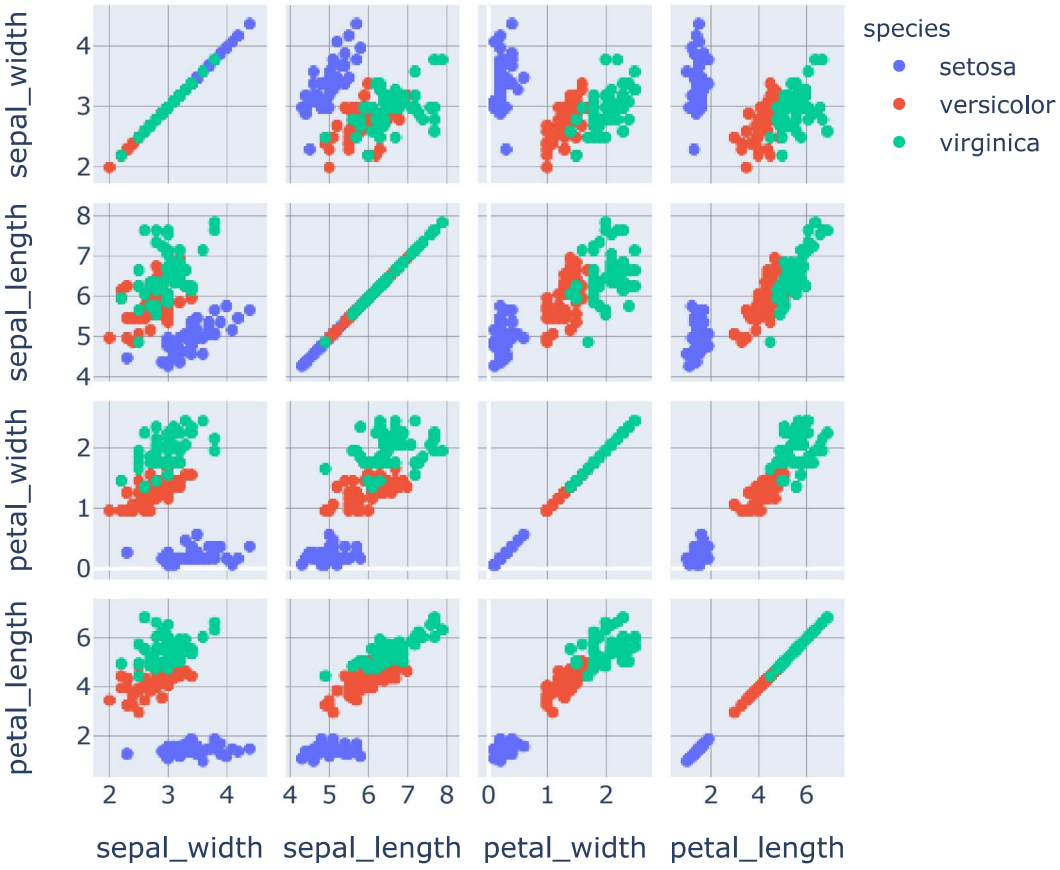
In [27]:

```
import plotly.express as px
df = px.data.iris()
fig = px.scatter(df,
         x="sepal_width", y="sepal_length",
         color="species", marginal_y="violin",
         marginal_x="box", trendline="ols")
fig.show()
```



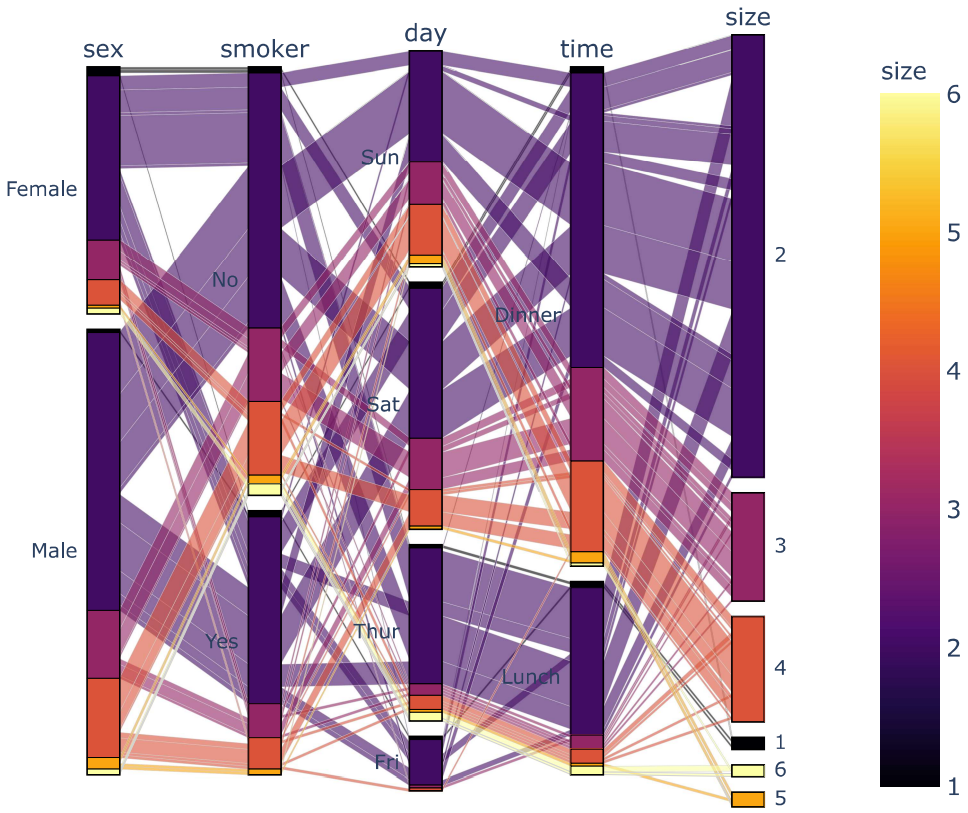
In [28]:

```
import plotly.express as px
df = px.data.iris()
fig = px.scatter_matrix(df, dimensions=["sepal_width",
                                        "sepal_length",
                                       "petal_width",
                                       "petal_length"],
                                       color="species")
fig.show()
```



In [29]:

```
import plotly.express as px
df = px.data.tips()
fig = px.parallel_categories(df, color="size",
                             color_continuous_scale=px.colors.sequential.Inferno)
fig.show()
```



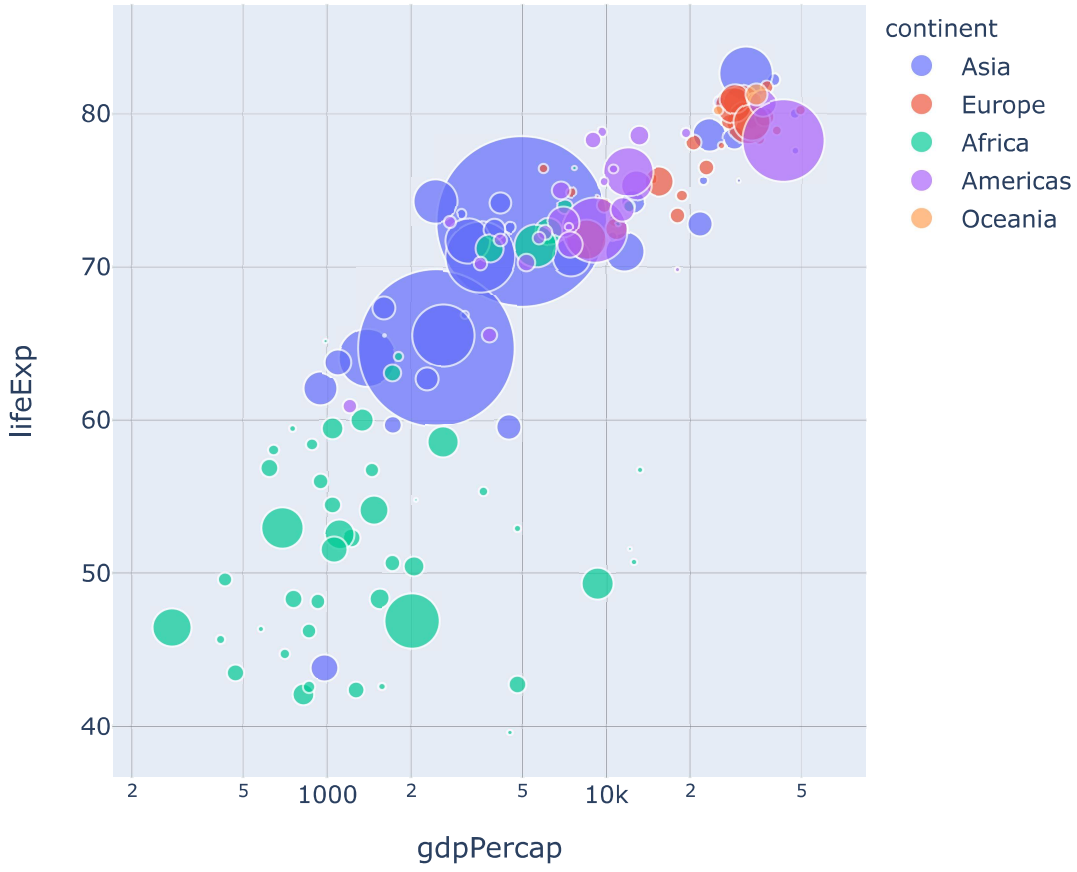
In [30]:

```
df = px.data.gapminder()
print(df.shape)
print(df.columns)
print(px.data.gapminder.__doc__)
```

```
(1704, 8)
Index(['country', 'continent', 'year', 'lifeExp', 'pop', 'gdpPercap',
       'iso_alpha', 'iso_num'],
      dtype='object')

Each row represents a country on a given year.

https://www.gapminder.org/data/ (https://www.gapminder.org/data/)

Returns:
    A `pandas.DataFrame` with 1704 rows and the following columns:
    `['country', 'continent', 'year', 'lifeExp', 'pop', 'gdpPercap',
    'iso_alpha', 'iso_num']`.
```

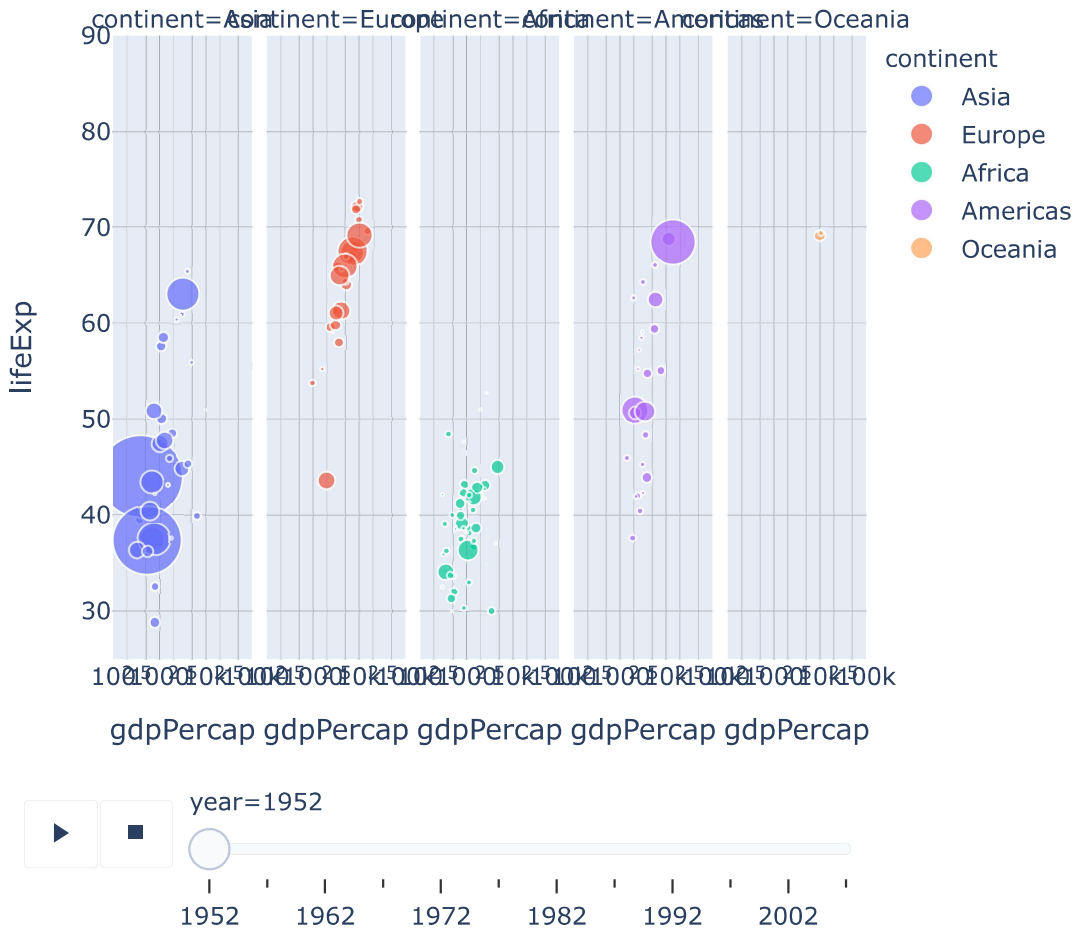
In [31]:

```
import plotly.express as px
df = px.data.gapminder()
fig = px.scatter(df.query("year==2007"),
         x="gdpPercap",
         y="lifeExp",
         size="pop",
         color="continent",
         hover_name="country", log_x=True, size_max=60)
fig.show()
```



In [32]:

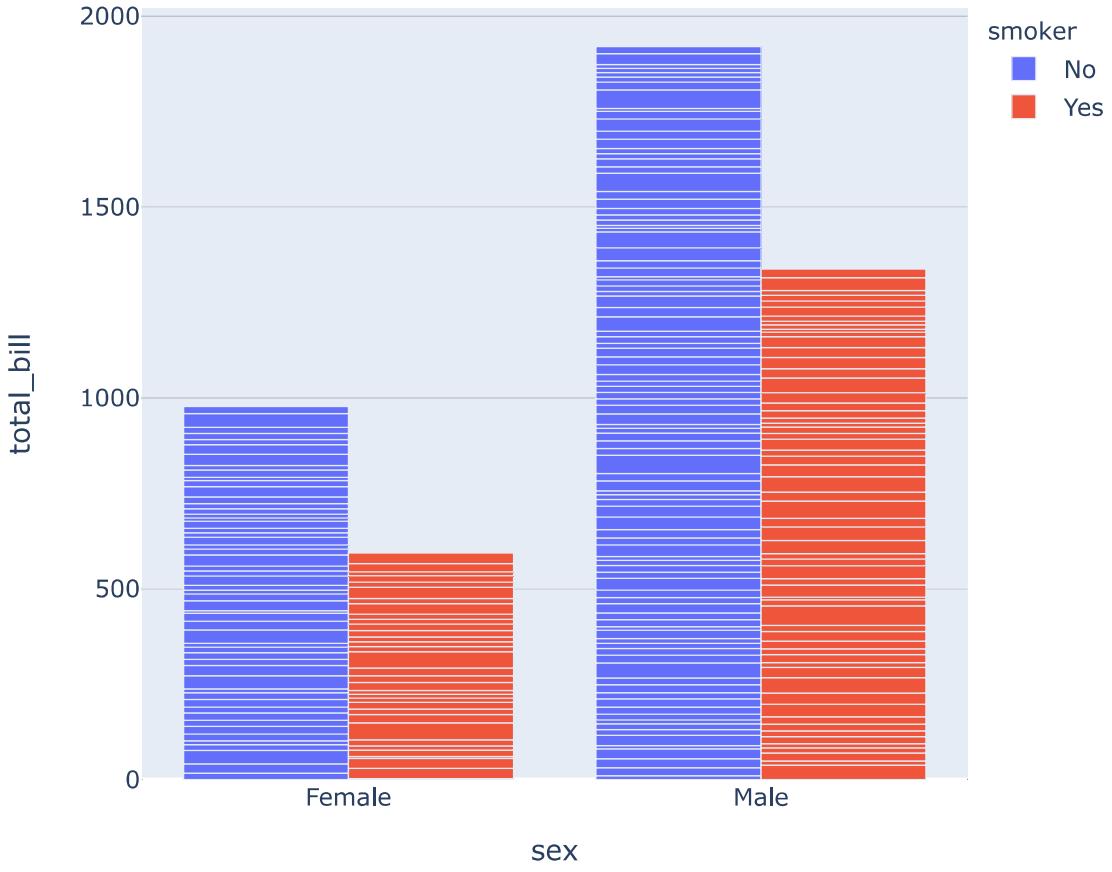
```
import plotly.express as px
df = px.data.gapminder()
fig = px.scatter(df, x="gdpPercap", y="lifeExp",
         animation_frame="year",
         animation_group="country",
         size="pop",
         color="continent",
         hover_name="country",
         facet_col="continent",
         log_x=True, size_max=45, range_x=[100,100000], range_y=[25,90])
fig.show()
```



막대 그래프

In [33]:

```
import plotly.express as px
df = px.data.tips()
fig = px.bar(df, x="sex", y="total_bill", color="smoker", barmode="group")
fig.show()
```



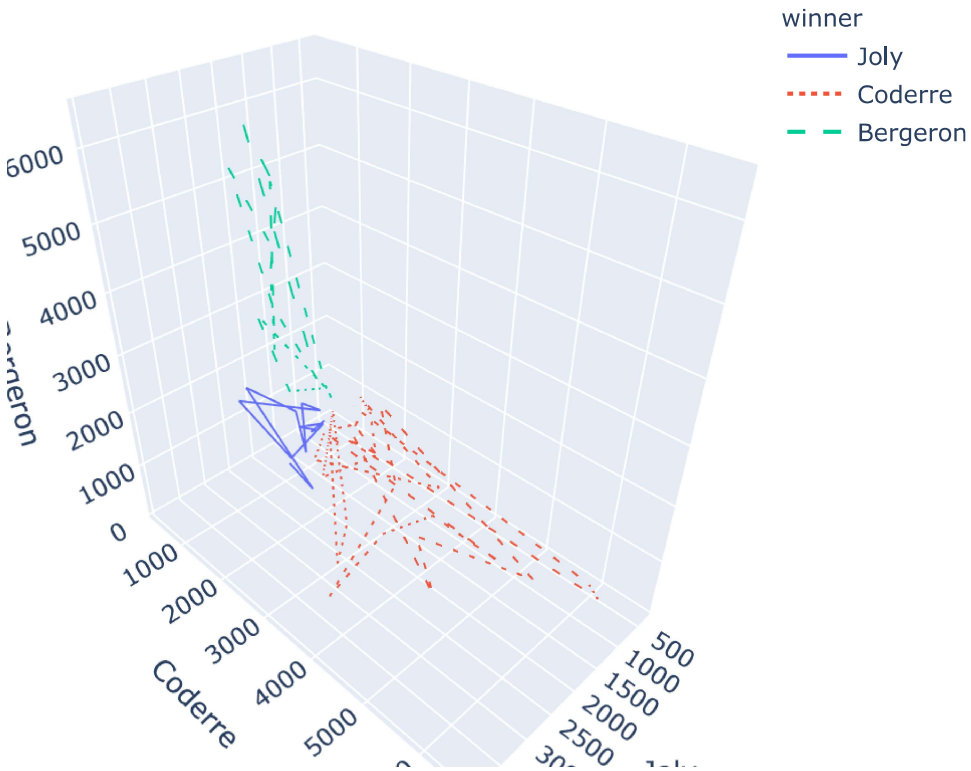
In [34]:

```
df = px.data.election()
print(df.shape)
print(df.head())
print(df.columns)
print(px.data.election.__doc__)
```

```
(58, 8)
                district  Coderre  Bergeron  Joly  total    winner     result  ₩
0     101-Bois-de-Liesse     2481      1829  3024   7334      Joly  plurality
1  102-Cap-Saint-Jacques     2525      1163  2675   6363      Joly  plurality
2   11-Sault-au-Récollet     3348      2770  2532   8650   Coderre  plurality
3           111-Mile-End     1734      4782  2514   9030  Bergeron   majority
4         112-DeLorimier     1770      5933  3044  10747  Bergeron   majority

   district_id
0          101
1          102
2           11
3          111
4          112
Index(['district', 'Coderre', 'Bergeron', 'Joly', 'total', 'winner', 'result',
       'district_id'],
      dtype='object')

Each row represents voting results for an electoral district in the 2013 Montreal
mayoral election.

Returns:
    A `pandas.DataFrame` with 58 rows and the following columns:
    `['district', 'Coderre', 'Bergeron', 'Joly', 'total', 'winner', 'result', 'distr
ict_id']`.
```

In [35]:

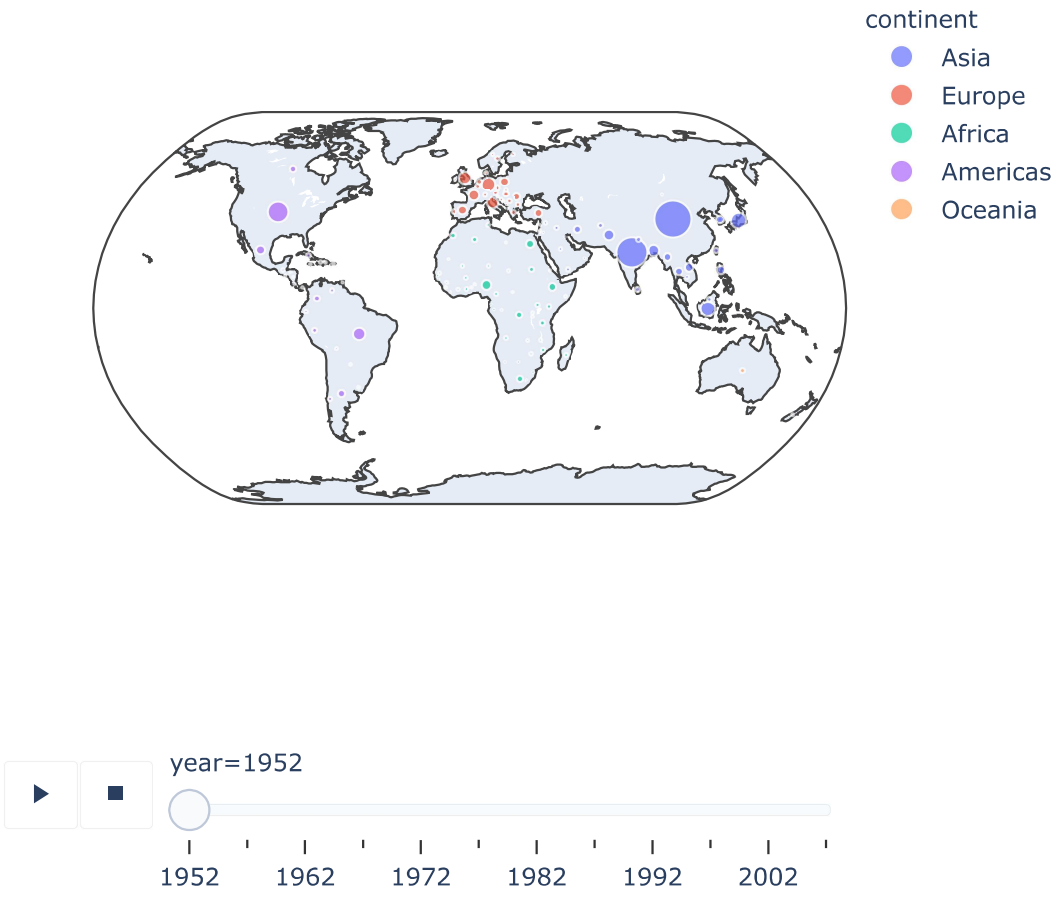
```
import plotly.express as px
df = px.data.election()
fig = px.line_3d(df,
                 x="Joly", y="Coderre", z="Bergeron",
                 color="winner", line_dash="winner")
fig.show()
```



Maps

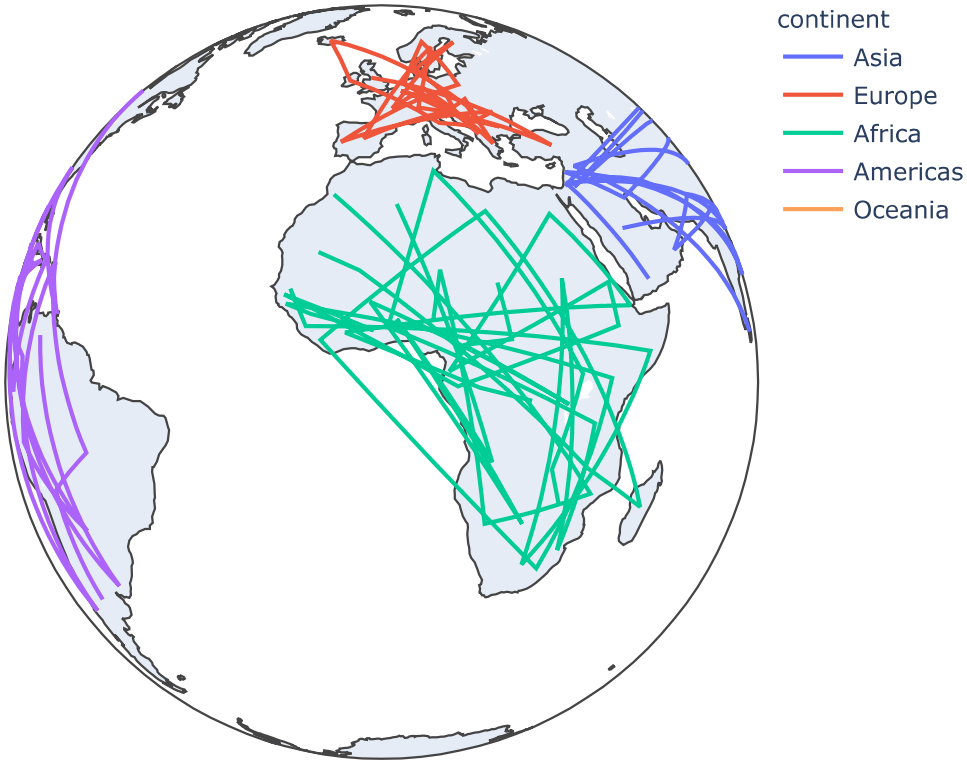
In [36]:

```
import plotly.express as px
df = px.data.gapminder()
fig = px.scatter_geo(df,
                     locations="iso_alpha",
                     color="continent",
                     hover_name="country",
                     size="pop",
                    animation_frame="year", projection="natural earth")
fig.show()
```



In [38]:

```
import plotly.express as px
df = px.data.gapminder().query("year == 2007")
fig = px.line_geo(df, locations="iso_alpha",
                  color="continent", # "continent" is one of the columns of gapminder
                  projection="orthographic")
fig.show()
```



REF

- cufflinks.datagen module
- https://jpoles1.github.io/cufflinks/html/cufflinks.datagen.html (https://jpoles1.github.io/cufflinks/html/cufflinks.datagen.html) (https://jpoles1.github.io/cufflinks/html/cufflinks.datagen.html (https://jpoles1.github.io/cufflinks/html/cufflinks.datagen.html))
- Plotly Express in Python
- https://plot.ly/python/plotly-express/#plotly-express (https://plot.ly/python/plotly-express/#plotly-express) (https://plot.ly/python/plotly-express/#plotly-express (https://plot.ly/python/plotly-express/#plotly-express))

In [ ]: