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 [1]:

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

In [2]:

```
# 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.or
g/wiki/Iris_flower_data_set)

Returns:
    A `pandas.DataFrame` with 150 rows and the following columns:
    `['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'specie
s', 'species_id']`.
```

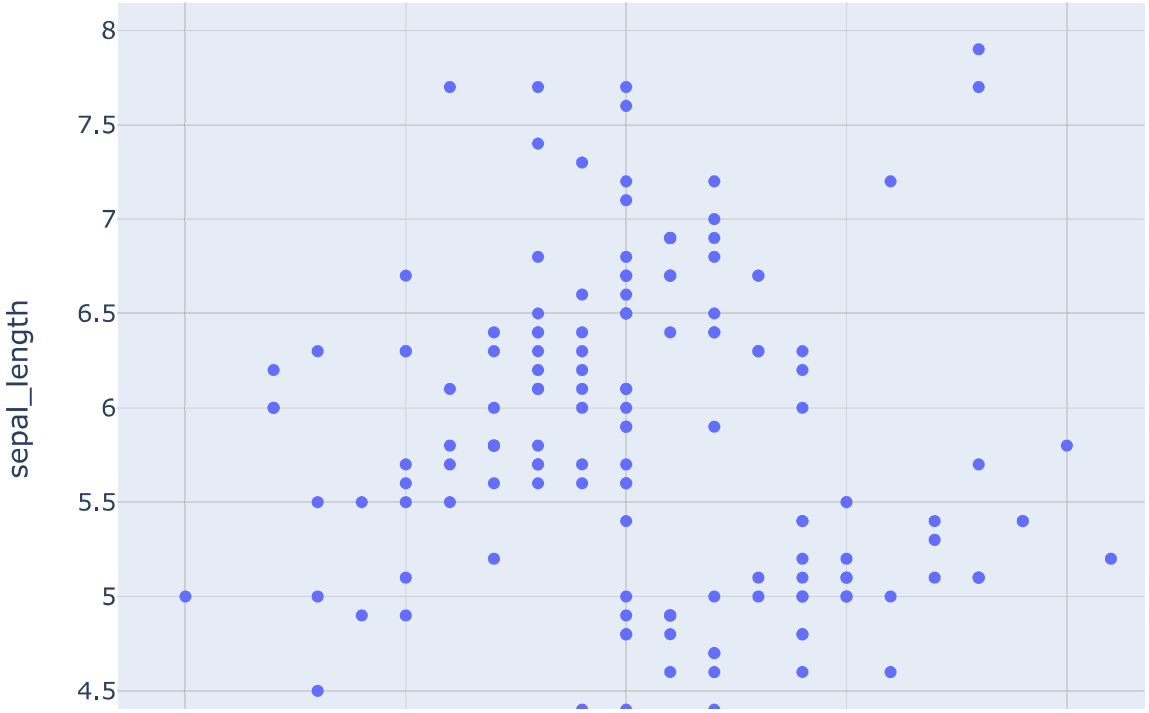
Out[2]:

	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

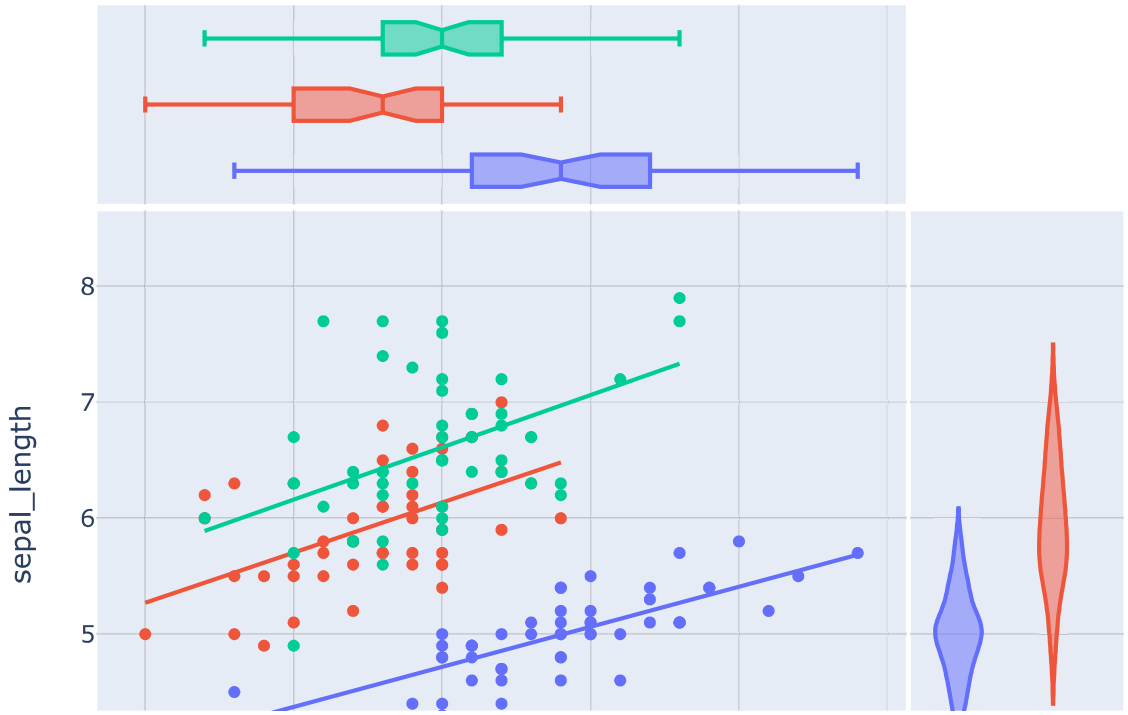
In [3]:

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



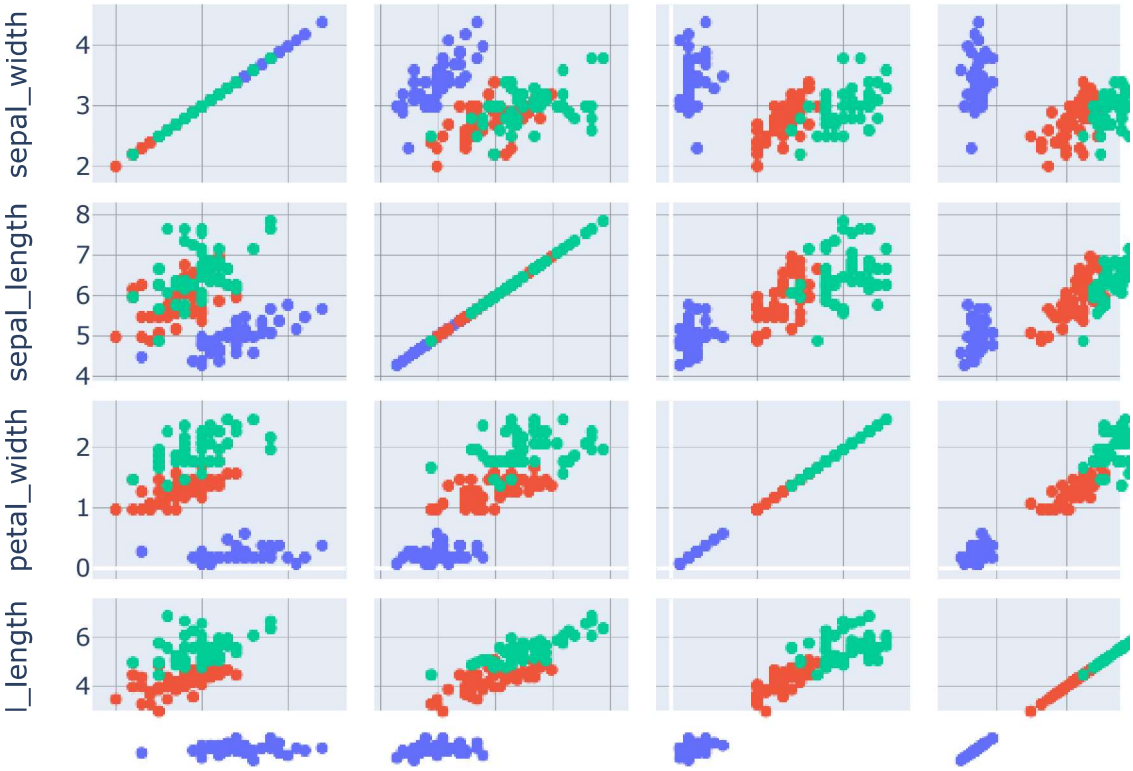
In [4]:

```
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()
```



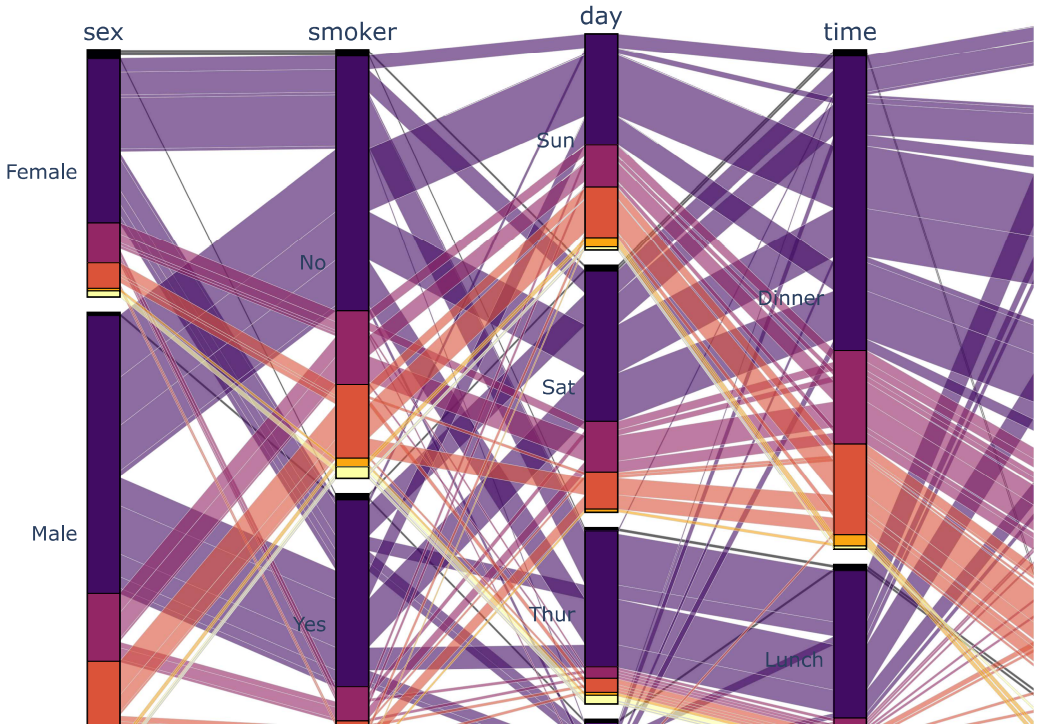
In [5]:

```
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 [6]:

```
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()
```



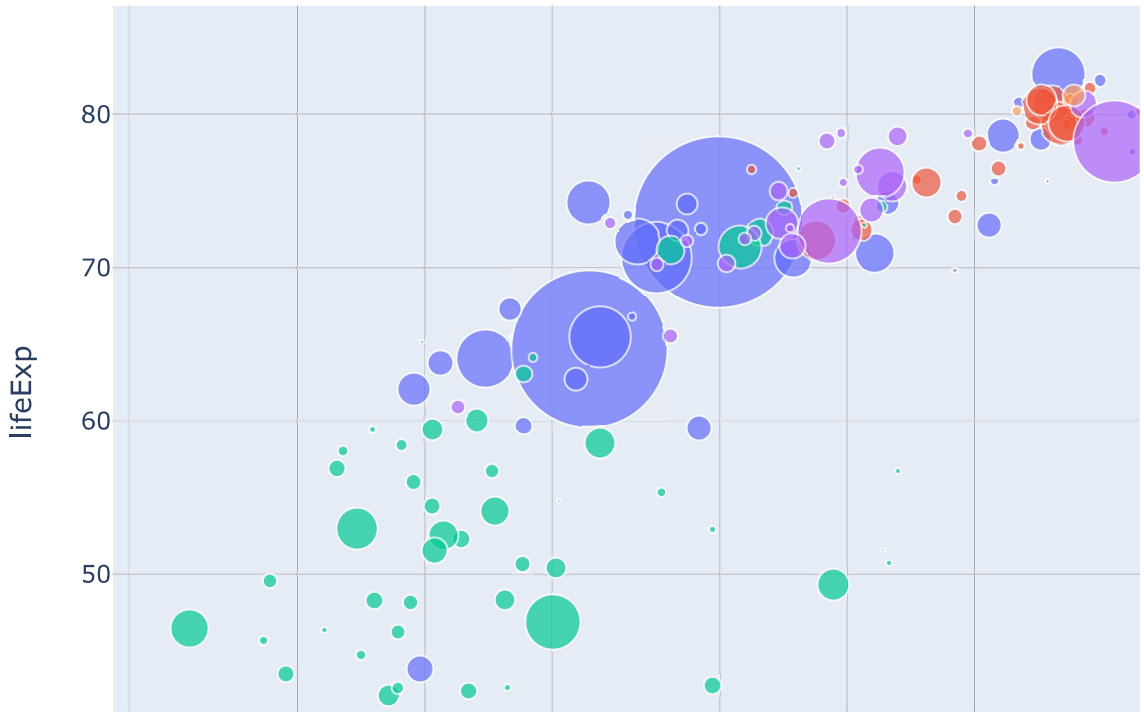
In [7]:

```
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']`.
```

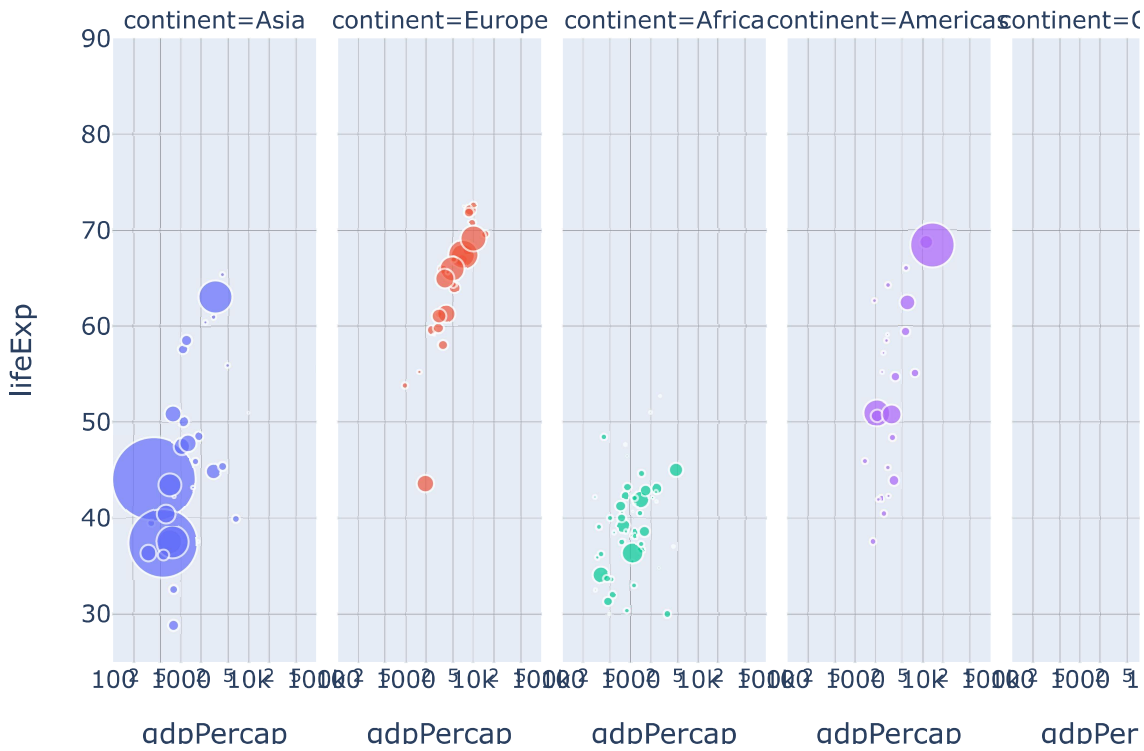
In [8]:

```
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 [9]:

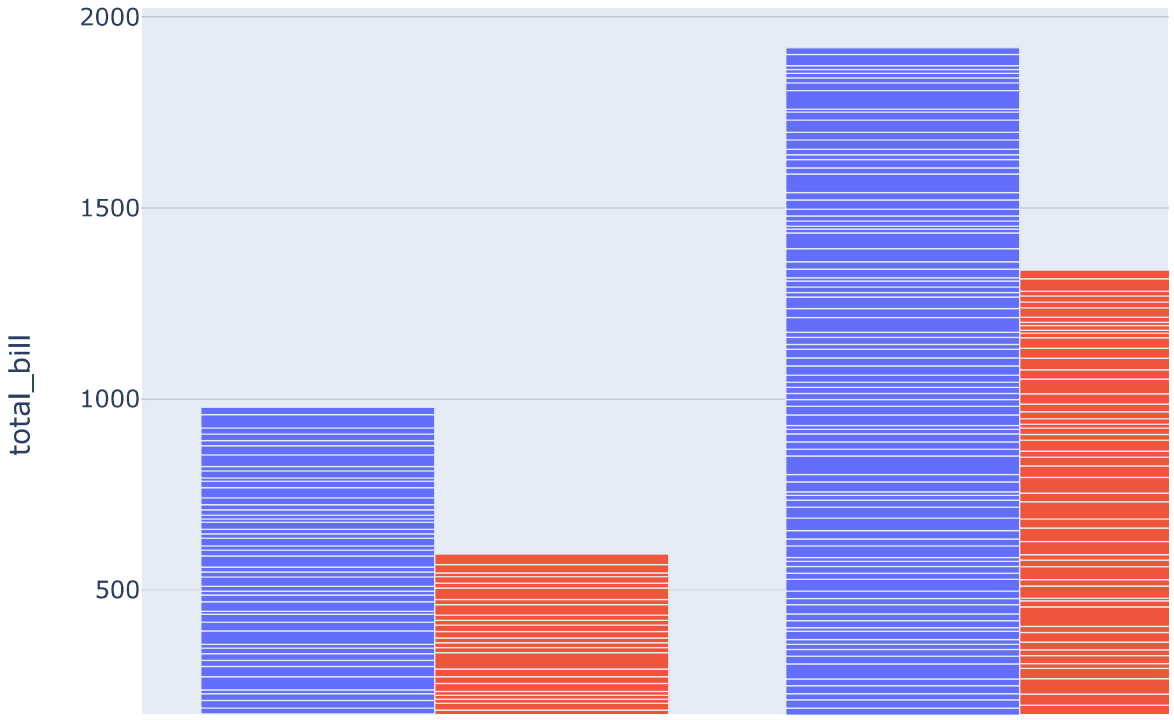
```
import plotly.express as px
df = px.data.gapminder()
fig = px.scatter(df, x="gdpPercap", y="lifeExp",
         animation_frame="year",
         animation_group="country",
         size="pop",
         color="continent",
         hover_name="country",
         facet_col="continent",
         log_x=True,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



막대 그래프

In [10]:

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



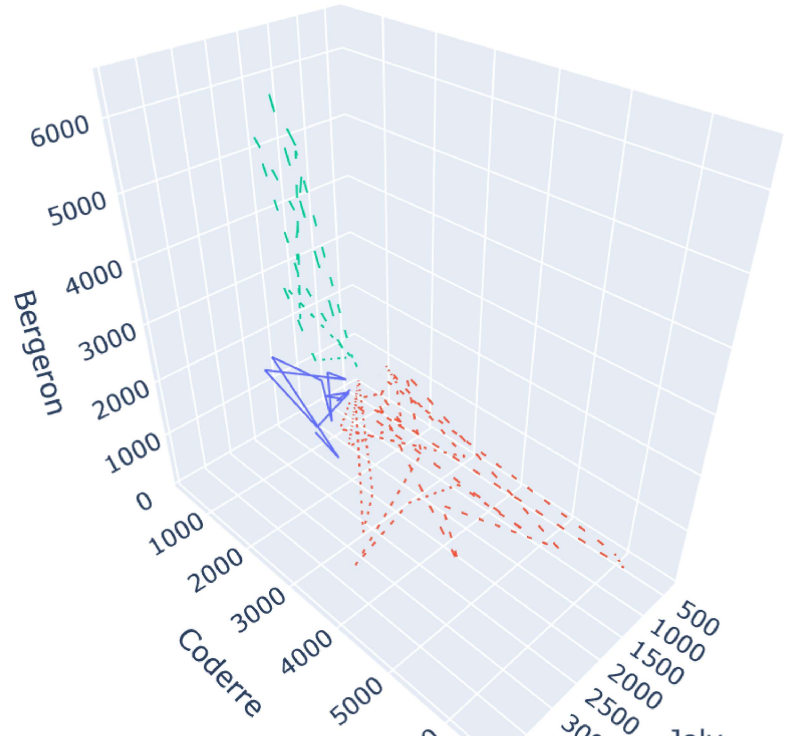
In [11]:

```
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     resul
t  \
0     101-Bois-de-Liesse     2481      1829  3024   7334      Joly  pluralit
y
1  102-Cap-Saint-Jacques     2525      1163  2675   6363      Joly  pluralit
y
2   11-Sault-au-Récollet     3348      2770  2532   8650   Coderre  pluralit
y
3           111-Mile-End     1734      4782  2514   9030  Bergeron   majorit
y
4         112-DeLorimier     1770      5933  3044  10747  Bergeron   majorit
y

   district_id
0          101
1          102
2           11
3          111
4          112
Index(['district', 'Coderre', 'Bergeron', 'Joly', 'total', 'winner', 'resul
t',
       'district_id'],
      dtype='object')

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

Returns:
    A `pandas.DataFrame` with 58 rows and the following columns:
    `['district', 'Coderre', 'Bergeron', 'Joly', 'total', 'winner', 'resul
t', 'district_id']`.
```

In [12]:

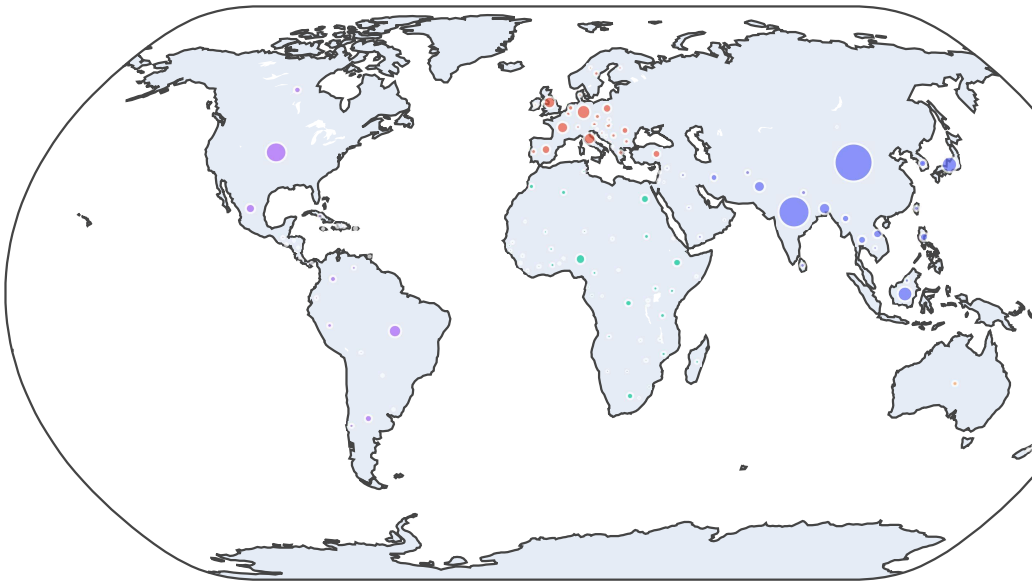
```
import plotly.express as px
df = px.data.election()
fig = px.line_3d(df,
                 x="Joly", y="Coderre", z="Bergeron",
                 color="winner", line_dash="winner")
fig.show()
```



MAPS

In [13]:

```
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 [14]:

```
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))