Plotly入门

In [1]:
```
import numpy as np
import pandas as pd
import plotly
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

In [2]:
```
# import cufflinks as cf
# from plotly.offline import *
# init_notebook_mode(connected=True)
# cf.go_offline()
```

In [3]:
```
%matplotlib inline
```

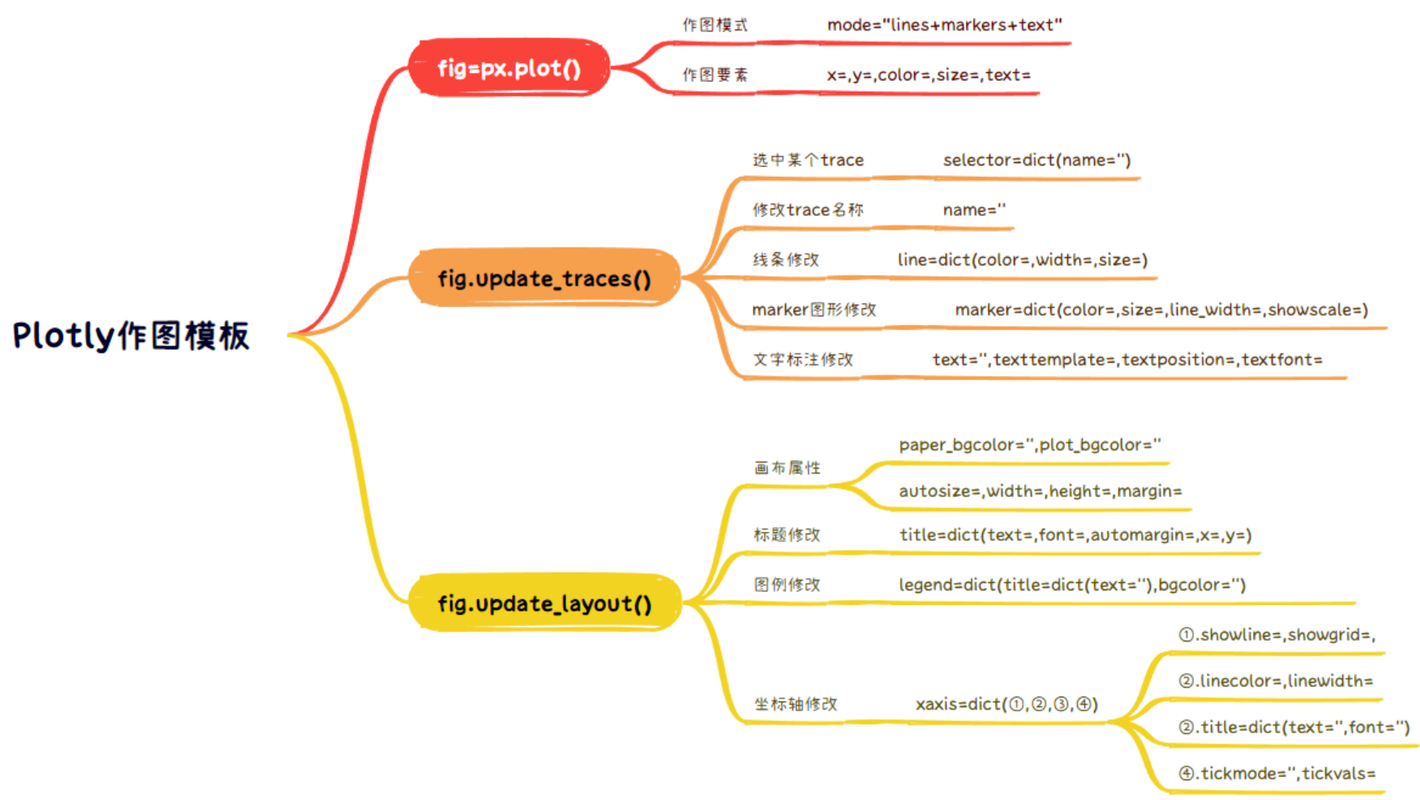
In [4]:
```
import warnings
warnings.filterwarnings("ignore")
```

主要元素一览

```
Figure({
    'data': [{'hovertemplate': 'x=%{x}<br>y=%{y}<extra></extra>',
              'legendgroup': '',
              'line': {'color': '#636efa', 'dash': 'solid'},
              'marker': {'symbol': 'circle'},
              'mode': 'lines',
              'name': '',
              'orientation': 'v',
              'showlegend': False,
              'type': 'scatter',
              'x': array(['a', 'b', 'c'], dtype=object),
              'xaxis': 'x',
              'y': array([1, 3, 2]),
              'yaxis': 'y'}],
    'layout': {'legend': {'tracegroupgap': 0},
               'template': '...',
               'title': {'text': 'sample figure'},
               'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'x'}},
               'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'y'}}}
})
```

参考链接:

examples: https://plotly.com/python/basic-charts
traces&layout: https://plotly.com/python/reference
Fundamentals: https://plotly.com/python/plotly-fundamentals
dataset：https://plotly.com/python-api-reference/generated/plotly.data.html#module-plotly.data
取色板：https://plotly.com/python/builtin-colorscales/



一、散点图 : Scatter Plot

step1：读取数据

再次重申一遍，plotly提供了很多可供实验的数据集，链接地址就在上面

In [5]:
```
df1=px.data.iris()
df1.head()
```

Out[5]:

	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

step2：px快速作图

In [6]:
```
fig1=px.scatter(df1,x="sepal_length",y="sepal_width",color="species",size="petal_length")
fig1
```

step3：trace元素(图形本体)修改

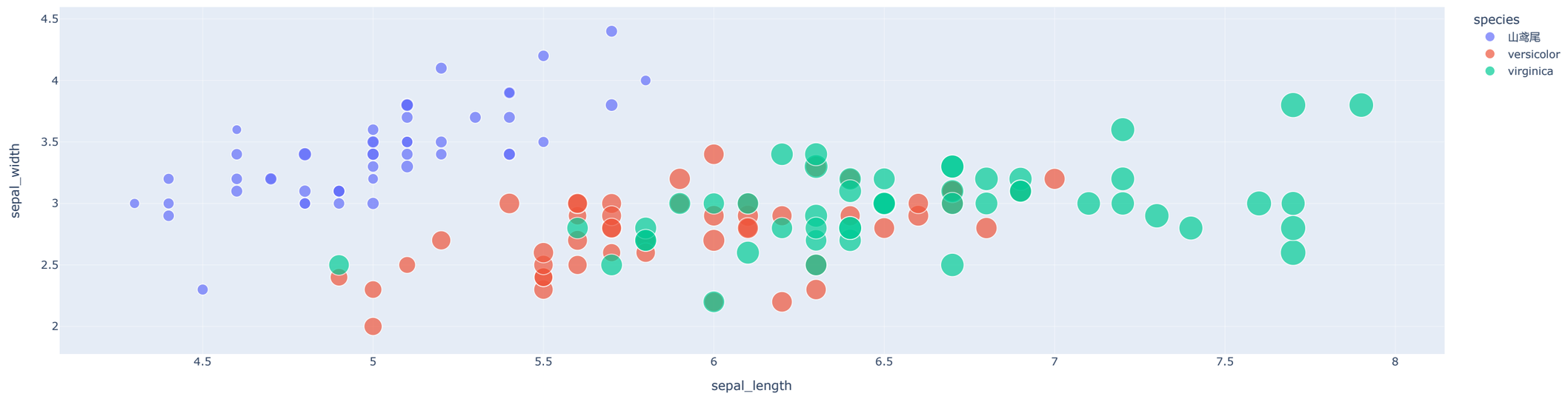
参考链接: https://plotly.com/python/reference/scatter/

这里的trace元素实际上就是画出来的这些点,这些点的主要属性设置如下:
1.selector=dict(name='Iris-setosa')选择对name='Iris-setosa'的trace操作

2.name="山鸢尾",覆盖刚刚选中的trace的name属性,由于name属性与legend关联,所以legend也会有相应改变
3.mode="lines+markers+text",作图改为用线+标记+文字标注
4.line=dict(),marker=dict(),text="",分别设置线条,marker,文字标注的属性

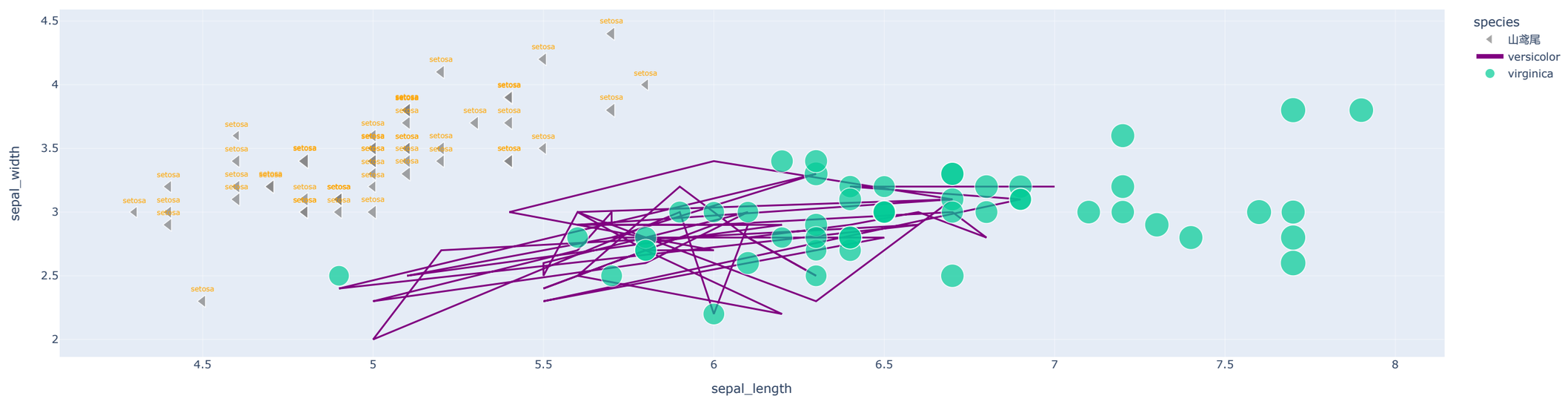
- 选中'Iris-setosa',然后改名为"山鸢尾"

In [7]:
```
fig1.update_traces(selector=dict(name='setosa'),name="山鸢尾")
```



- 作图模式改成线条+marker+文字标注
- 分别对线条,marker,文字标注进行修改

In [8]:
```
fig1.update_traces(selector=dict(name='山鸢尾'),mode="markers+text")
fig1.update_traces(selector=dict(name='versicolor'),mode="lines",line=dict(color='purple'))
fig1.update_traces(selector=dict(name='山鸢尾'),marker=dict(symbol='triangle-left',color="gray"))
fig1.update_traces(selector=dict(name='山鸢尾'),text=df1.species,textfont=dict(size=8,color='orange'),textposition='top center')
```

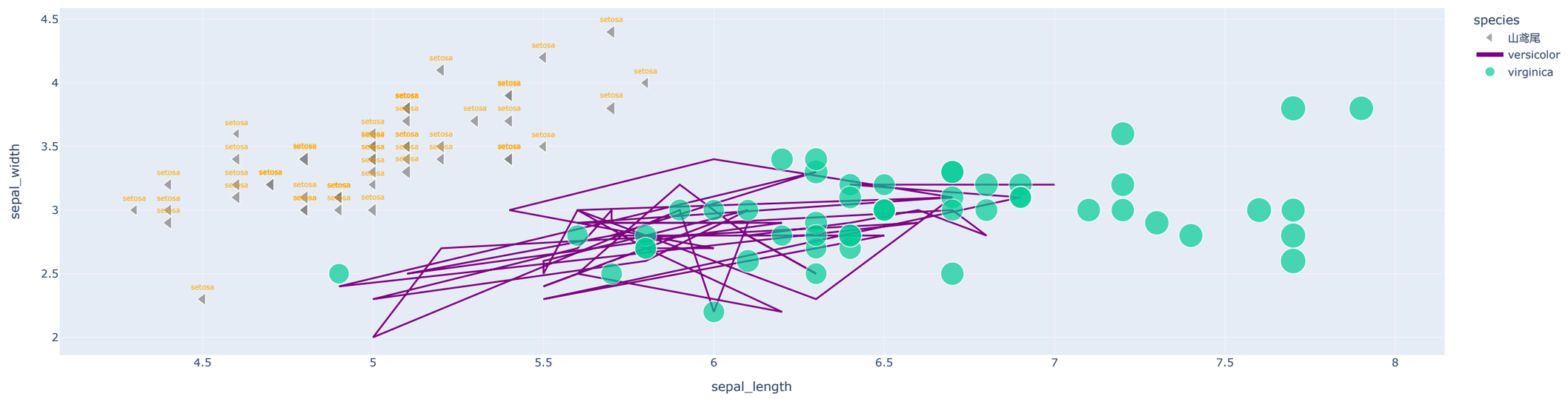


step4：layout元素(title,legend,xaxis,yaxis)修改

参考链接: https://plotly.com/python/reference/layout/

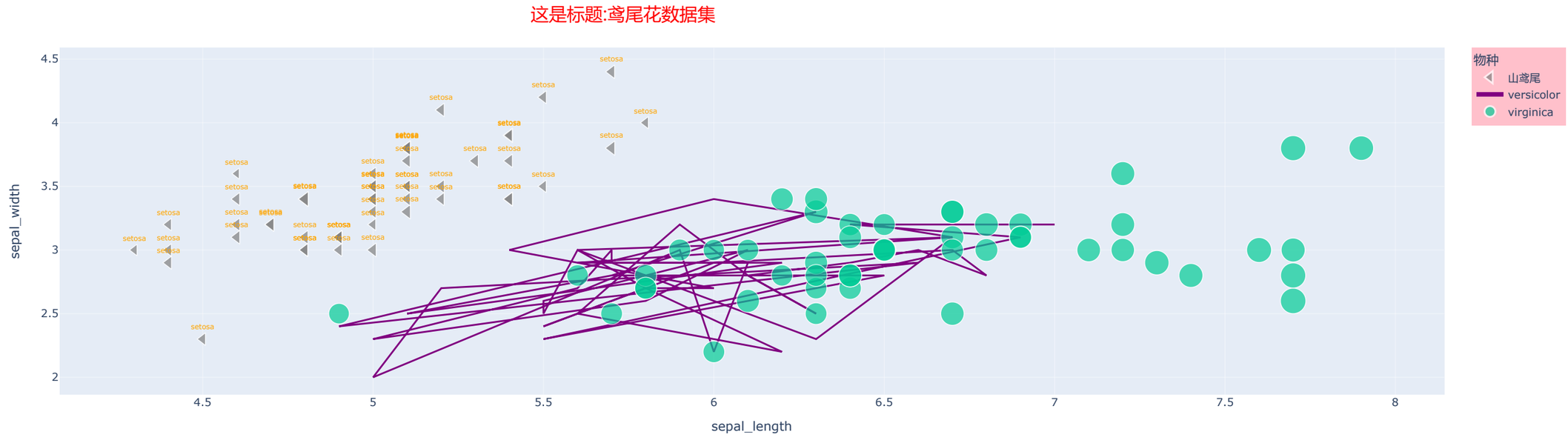
- 标题名称,字体,位置修改
  参考链接:https://plotly.com/python/reference/layout/

In [9]:
```
fig1.update_layout(title=dict(text="这是标题:鸢尾花数据集",font=dict(color="red",size=20,family="Balto"),x=0.4))
```



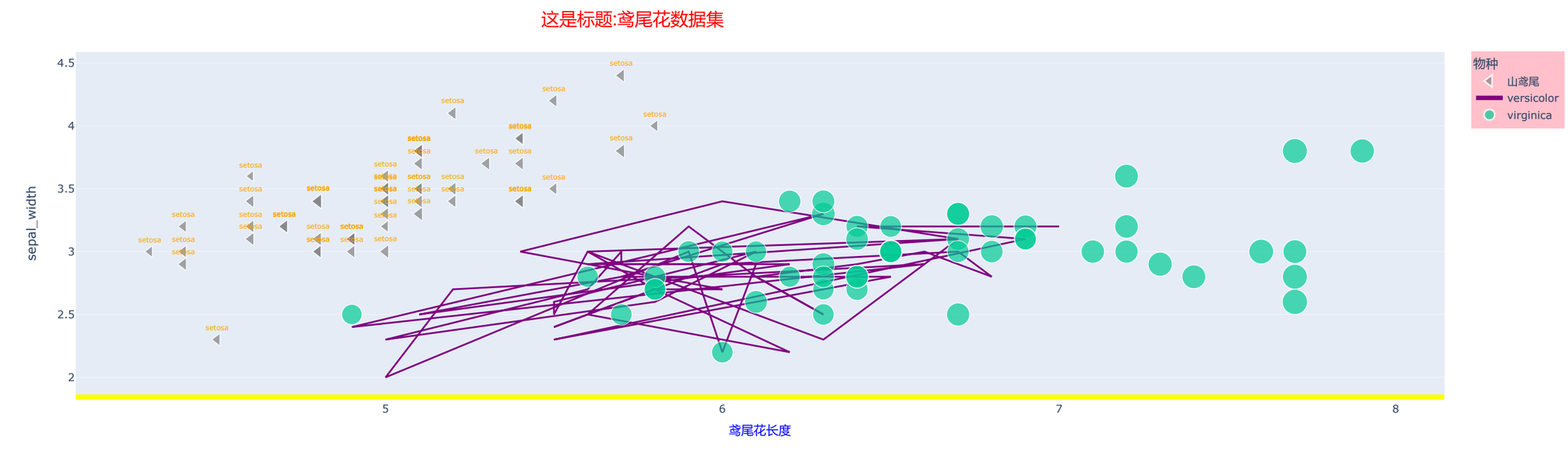
- 图例名称,字体,位置修改
  参考链接:https://plotly.com/python/reference/layout/#layout-legend
- 图例更多是与trace关联的,trace生成的时候会自动化处理legend
  所以有关legend的更多设置请参考:https://plotly.com/python/legend/

In [10]:
```
fig1.update_layout(legend=dict(bgcolor="pink",title=dict(text="物种")))
```

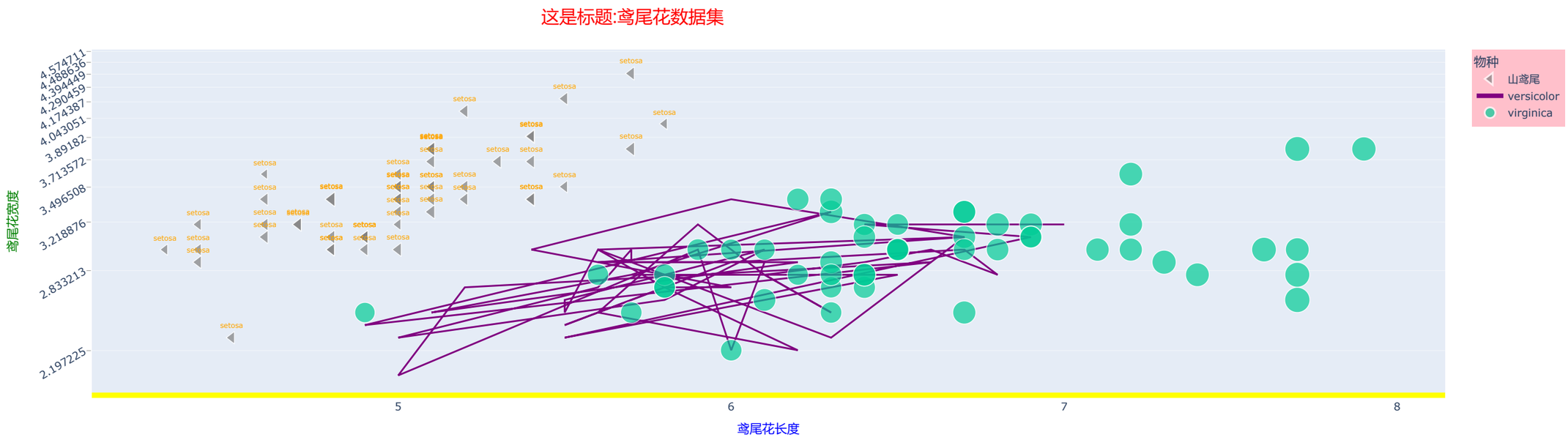


- 坐标轴属性(名称,轴线,刻度,网格)修改
  参考链接:https://plotly.com/python/reference/layout/xaxis/

In [11]:
```
fig1.update_layout(xaxis=dict(showgrid=False,showline=True))
fig1.update_layout(xaxis=dict(title=dict(text="鸢尾花长度",font=dict(color='blue'))))  # 设置title
fig1.update_layout(xaxis=dict(linecolor='yellow',linewidth=6))                         # 设置line
fig1.update_layout(xaxis=dict(tickmode="array",tickvals=np.arange(3,9)))               # 设置刻度
```



In [12]:
```
fig1.update_layout(yaxis=dict(showgrid=True,showline=False))
fig1.update_layout(yaxis=dict(title=dict(text="鸢尾花宽度",font=dict(color='green'))))  # 设置title
fig1.update_layout(yaxis=dict(linecolor='red',linewidth=3))                            # 设置line
fig1.update_layout(yaxis=dict(tickmode="array",tickvals=np.log(np.arange(1,100,8)),tickangle=-30,ticks='outside'))  # 设置刻度
```



- 上面的元素属性设置规律
  首先认清楚对哪个元素进行设置，这里元素分为：
  trace元素：lines，markers，text
  layout元素：title，legend，xaxis，yaxis
  然后针对每个元素，具体设置其属性，比如设置线条lines的属性，就是设置它的颜色，名称等
- 实际上还有另外一种方法设置属性
  就是元素+下划线+属性的方式，比如xaxis_title，
  但是这种有的可以，有的不行，所以尽量不要用这种
- 设置属性查询链接：
  https://plotly.com/python/reference
  一般每个元素到底能设置什么属性这个链接都比较全
  遗憾的是，里面对于每个属性的使用讲的不是很细
  所以有些属性放在那里，你也不知道它是用来干什么的
  后面会尽可能给出多的例子，来熟悉尽可能多的属性

小结：

正常来说，排版细节交给plotly自动化处理，我们只需要修改关键部分的数据属性即可，比如标题，轴的名称，图例的名称，默认排版是比较好看的，跟latex文字排版类似，我们只需要给它提供一个template模板即可。

二、曲线图：Line Plot

step1：读取数据

In [13]:
```
df2=px.data.stocks()
df2.head()
```

Out[13]:

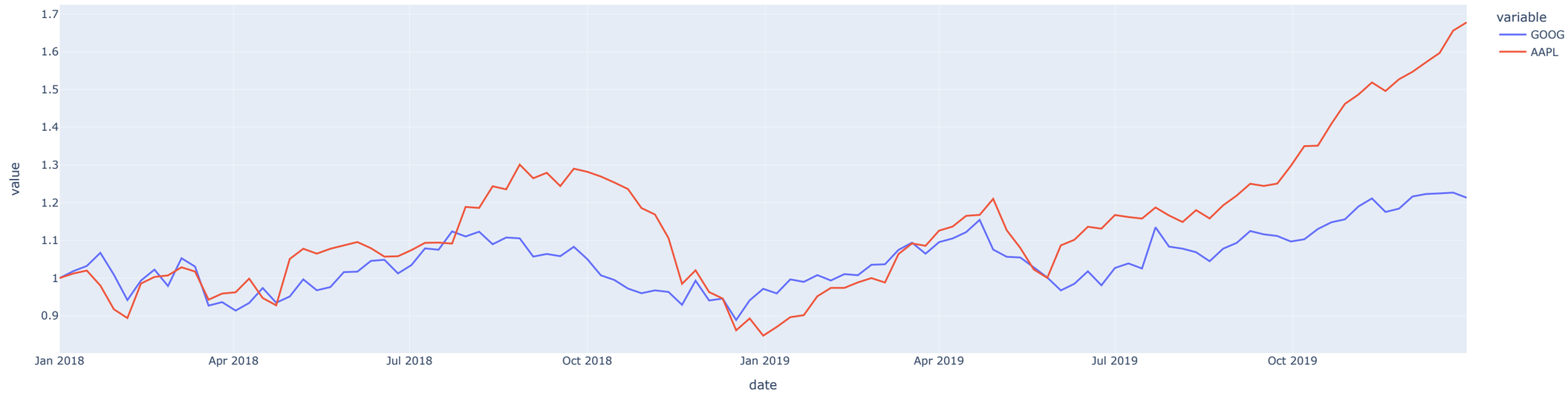
	date	GOOG	AAPL	AMZN	FB	NFLX	MSFT
0	2018-01-01	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1	2018-01-08	1.018172	1.011943	1.061881	0.959968	1.053526	1.015988
2	2018-01-15	1.032008	1.019771	1.053240	0.970243	1.049860	1.020524
3	2018-01-22	1.066783	0.980057	1.140676	1.016858	1.307681	1.066561
4	2018-01-29	1.008773	0.917143	1.163374	1.018357	1.273537	1.040708

step2：作图

- 下面提供了3种常见方式作图：
  第一种就是直接px作图
  第二种就是使用go.Scatter画出轨迹，然后使用已存在的figure添加这个轨迹，或者丢到新新创建的Figure对象中
- 关于go.Scatter和px可直接设置的元素属性
  px：trace和layout的所有元素属性都能设置
  go.Scatter：只能设置trace相关属性

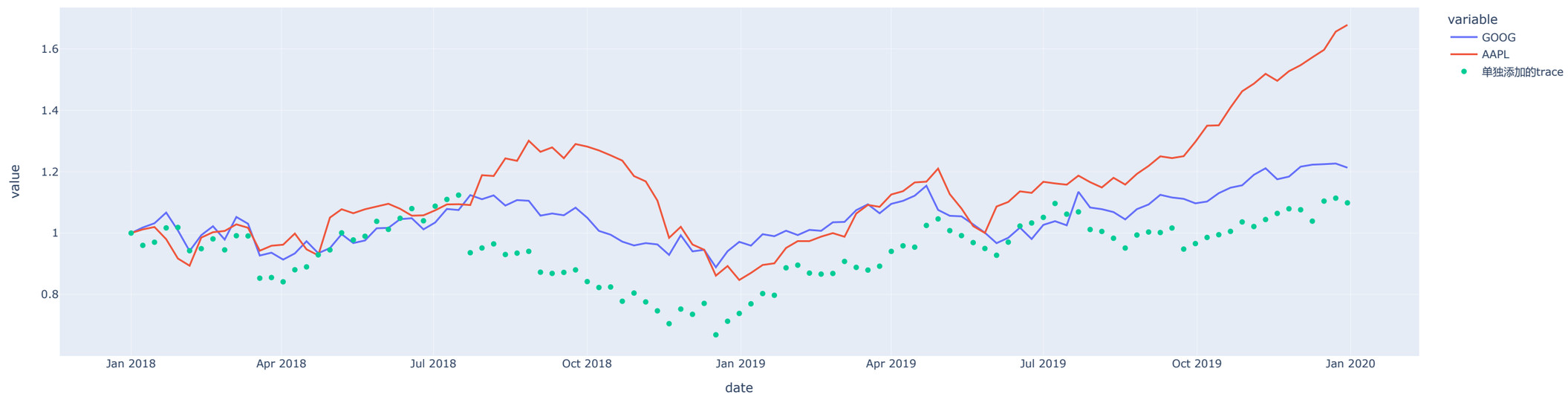
- px快速作图

In [14]:
```
fig2=px.line(df2,x='date',y=['GOOG','AAPL'])
fig2
```



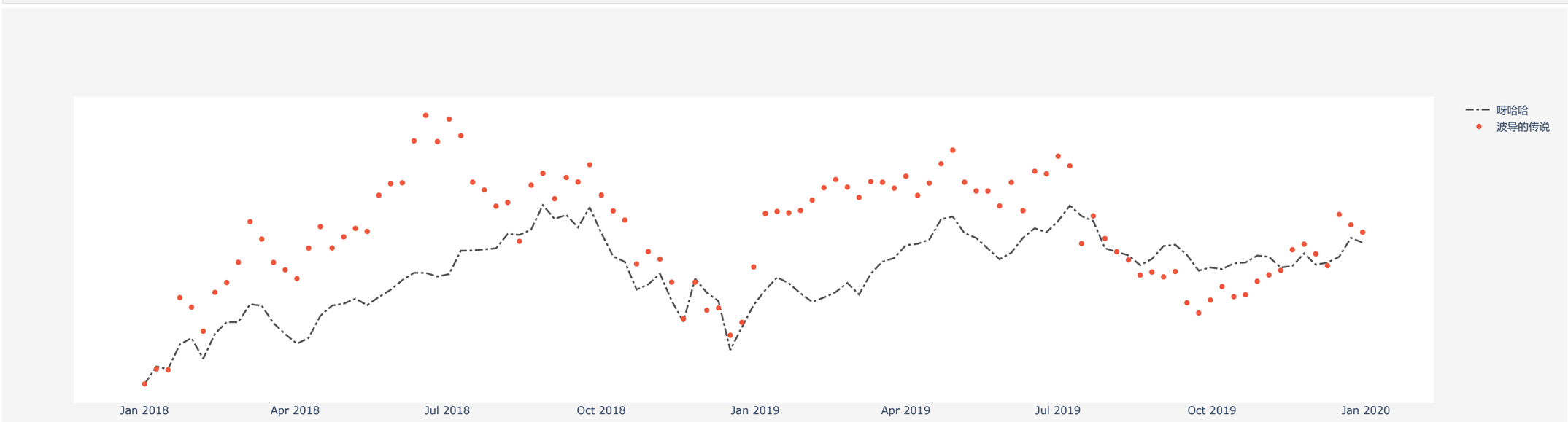
- go.Scatter作出轨迹，fig.add_trace添加此轨迹

In [15]:
```
trace=go.Scatter(x=df2.date,y=df2.FB,mode='markers',name='单独添加的trace')
fig2.add_trace(trace)
```



- go.Scatter作出轨迹，go.Figure根据轨迹作图

In [16]:
```
trace1=go.Scatter(x=df2.date,y=df2.AMZN,mode='lines',name='呀哈哈',line=dict(color='rgb(82,82,82)',dash='dashdot'))  # 这里dash参数是设置点划线
trace2=go.Scatter(x=df2.date,y=df2.NFLX,mode='markers',name='波导的传说')
fig3=go.Figure(data=[trace1,trace2])
fig3.update_layout(yaxis=dict(showline=False,showgrid=False,showticklabels=False)) # 这里取消显示y轴的轴线，网格线，刻度
fig3.update_layout(paper_bgcolor='rgb(245,245,245)',plot_bgcolor='white') # 设置背景颜色
```



- 介绍上面的原因：
  其一，是为了更加了解plotly的go和px作图原理
  其二，是为了更加灵活作图，比如前面作好的图，再添加一个trace
  使用建议，默认遵循px快速作图-->设置trace元素属性-->设置layout属性这个范式

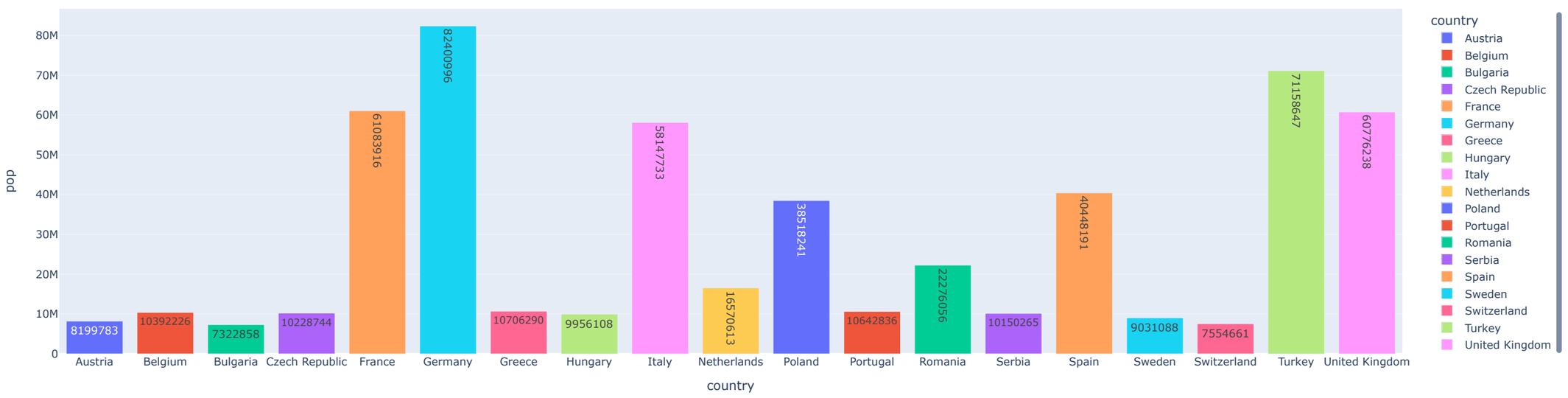
三、柱状图：Bar Plot

In [17]:
```
df3=px.data.gapminder().query("continent=='Europe' and year==2007 and pop>6e6")
df3.head()
```

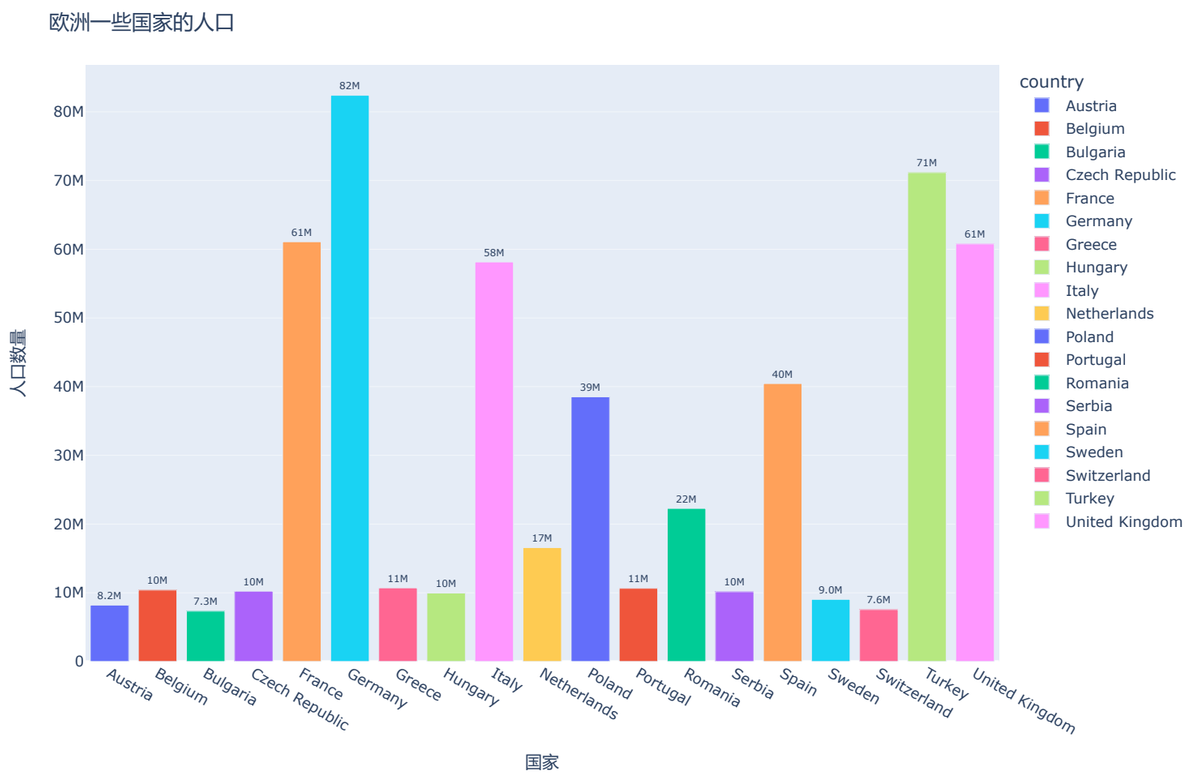
Out[17]:

	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
83	Austria	Europe	2007	79.829	8199783	36126.49270	AUT	40
119	Belgium	Europe	2007	79.441	10392226	33692.60508	BEL	56
191	Bulgaria	Europe	2007	73.005	7322858	10680.79282	BGR	100
407	Czech Republic	Europe	2007	76.486	10228744	22833.30851	CZE	203
539	France	Europe	2007	80.657	61083916	30470.01670	FRA	250

In [18]:
```
fig4=px.bar(df3,x='country',y='pop',text='pop',color='country')
fig4
```



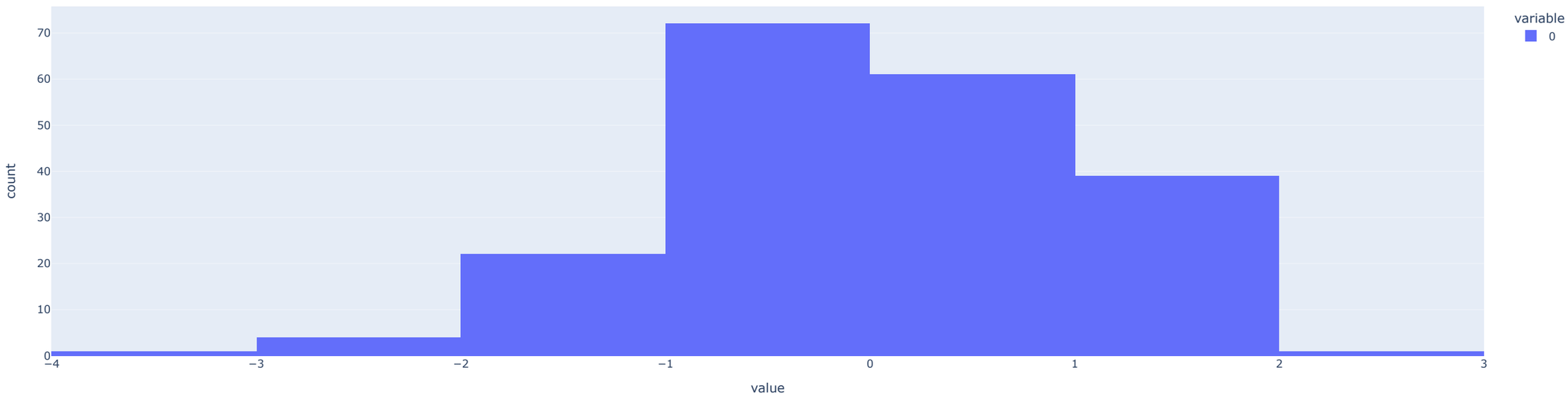
In [19]:
```
fig4.update_traces(texttemplate='%{text:0.2s}',textposition='outside',textfont=dict(size=8)) # 文字标注的格式控制
fig4.update_layout(title='欧洲一些国家的人口',autosize=False,width=1000,height=650,xaxis_title='国家',yaxis_title='人口数量')
```



四、直方图：Histogram Plot

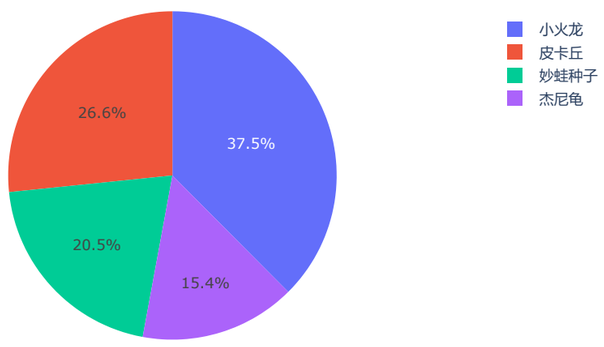
注意，上面的柱状图一般x轴是离散值，而这里的直方图x轴一般是连续值，需要提供nbins分割

In [20]:
```
data=np.random.randn(200)
px.histogram(data,nbins=7)
```



五、饼图：Pie Plot

In [21]:
```
trace=go.Pie(labels=['小火龙','妙蛙种子','杰尼龟','皮卡丘'],values=[110,60,45,78])
layout=go.Layout(autosize=False)
go.Figure(data=[trace],layout=layout)
```



案例：手势识别的论文分区分布图

一般脑子里想着取出哪一列或者哪两列出来作图

In [22]:
```
df=pd.read_csv("./dataset/slr/slr_research_preprocess.csv",index_col=0)
df.head()
```

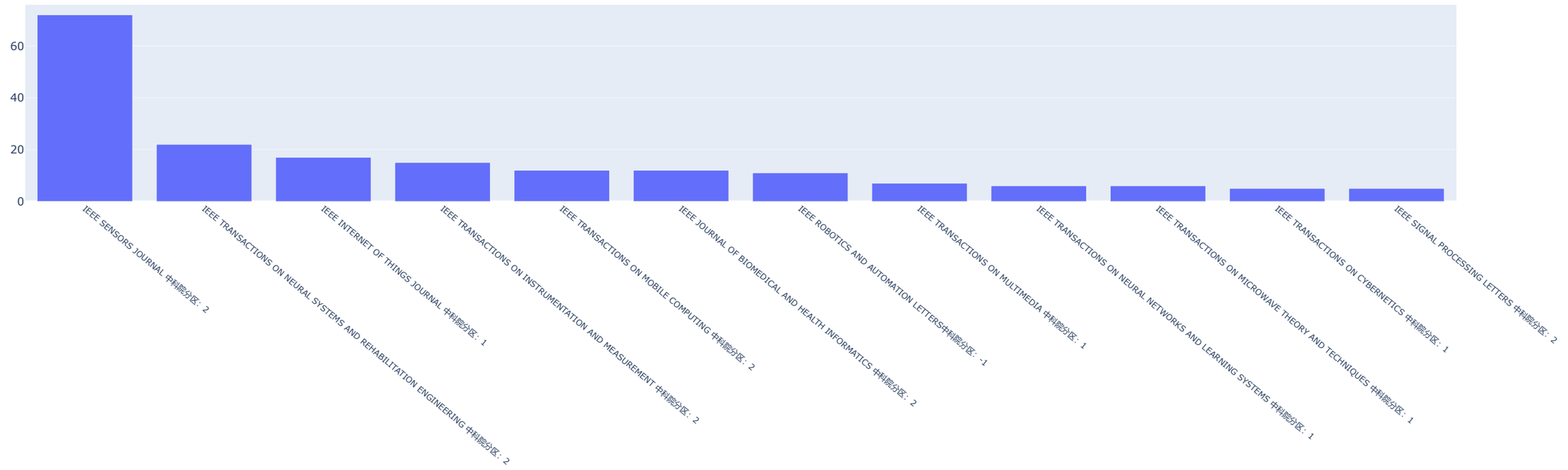
Out[22]:

	出版物类型	作者全名	文章标题	期刊名称	关键词（作者提供）	关键词（Web of Science 提供）	摘要	引用参考文献数目（Web of Science提供）	引用参考文献数目（作者提供）	引用次数（Web of Science提供）	总引用次数（Web of Science提供）	期刊ISO缩写名称，即期刊名ISO缩写。	期刊	分区
0	J	Guo, Zihui; Hou, Yonghong; Hou, Chunping; Yin,...	Locality-Aware Transformer for Video-Based Sig...	IEEE SIGNAL PROCESSING LETTERS	Videos; Assistive technologies; Gesture recogn...	RECOGNITION	Recently, the application of transformer makes...	Aoxiong Yin, 2021, MM '21: Proceedings of the ...	31	0	0	IEEE Signal Process. Lett.	IEEE SIGNAL PROCESSING LETTERS 中科院分区：2	2
1	J	Zhao, Jian; Qi, Weizhen; Zhou, Wengang; Duan, ...	Conditional Sentence Generation and Cross-Moda...	IEEE TRANSACTIONS ON MULTIMEDIA	Assistive technology; Videos; Gesture recognit...	RECOGNITION; FRAMEWORK	Sign Language Translation (SLT) aims to genera...	auslan, US; awhamburg, US; Bahdanau D, 2016, A...	67	5	5	IEEE Trans. Multimedia	IEEE TRANSACTIONS ON MULTIMEDIA 中科院分区：1	1
2	J	Tang, Shengeng; Guo, Dan; Hong, Richang; Wang,...	Graph-Based Multimodal Sequential Embedding fo...	IEEE TRANSACTIONS ON MULTIMEDIA	Continuous sign language translation; graph co...	RECOGNITION; FRAMEWORK	Sign language translation (SLT) is a challengi...	Beck D, 2018, PROCEEDINGS OF THE 56TH ANNUAL M...	74	7	7	IEEE Trans. Multimedia	IEEE TRANSACTIONS ON MULTIMEDIA 中科院分区：1	1
3	J	Xie, Pan; Zhao, Mengyi; Hu, Xiaohui	PiSLTRc: Position-Informed Sign Language Trans...	IEEE TRANSACTIONS ON MULTIMEDIA	Sign language recognition; sign language trans...	RECOGNITION; FRAMEWORK; NETWORK	Since the superiority of Transformer in learni...	Ba J.L., 2016, ARXIV160706450; Ba J.L., 2016, ...	52	2	2	IEEE Trans. Multimedia	IEEE TRANSACTIONS ON MULTIMEDIA 中科院分区：1	1
4	J	Hu, Jiwei; Liu, Yunfei; Lam, Kin-Man; Lou, Ping	STFE-Net: A Spatial-Temporal Feature Extractio...	IEEE ACCESS	Feature extraction; Assistive technologies; Ge...	NaN	The main challenge of continuous sign language...	Ariesta MC, 2018, 2018 INDONESIAN ASSOCIATION ...	36	0	0	IEEE Access	IEEE ACCESS 中科院分区：3	3

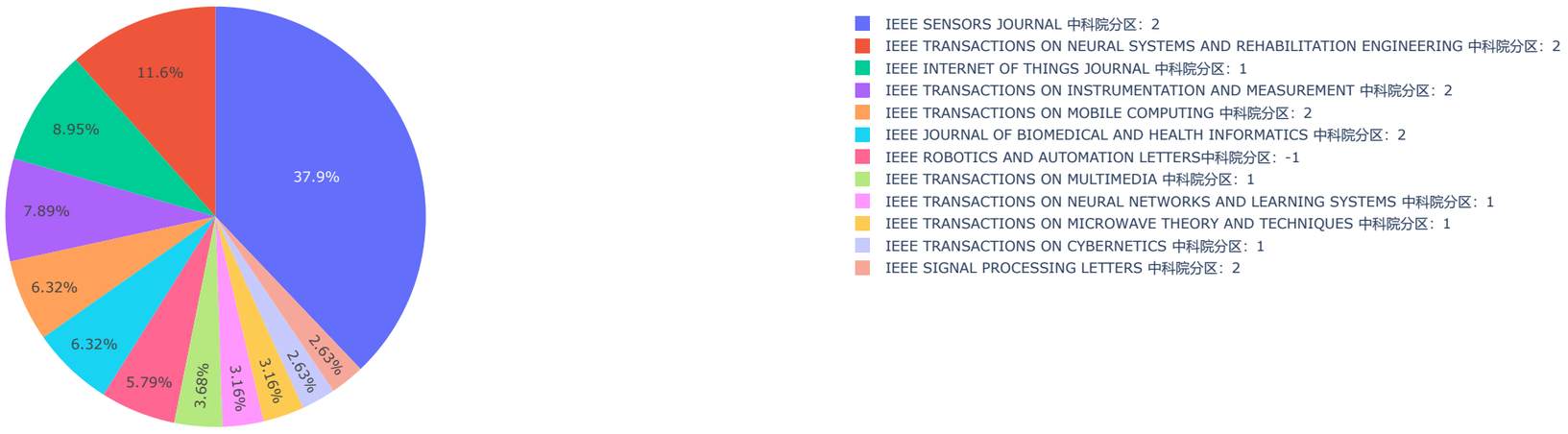
查看期刊分布和期刊饼图

In [23]:
```
df_=df.query("分区<3").期刊.value_counts()
df_=pd.DataFrame({'期刊':df_.index.values,'counts':df_.values}).iloc[:12,]
df_.期刊=df_.期刊.astype(str)
```

In [24]:
```
fig=px.bar(df_,x='期刊',y='counts')
fig.update_layout(showlegend=False)                          # 去掉图例
fig.update_layout(xaxis=dict(title=''),yaxis=dict(title='')) # 去掉坐标轴名称
fig.update_layout(xaxis=dict(tickfont=dict(size=9),tickangle=40)) # x轴文字设置
fig.update_layout(height=600)
```



In [25]:
```
fig=px.pie(names=df_.期刊,values=df_.counts)
fig.update_layout(height=500)
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: