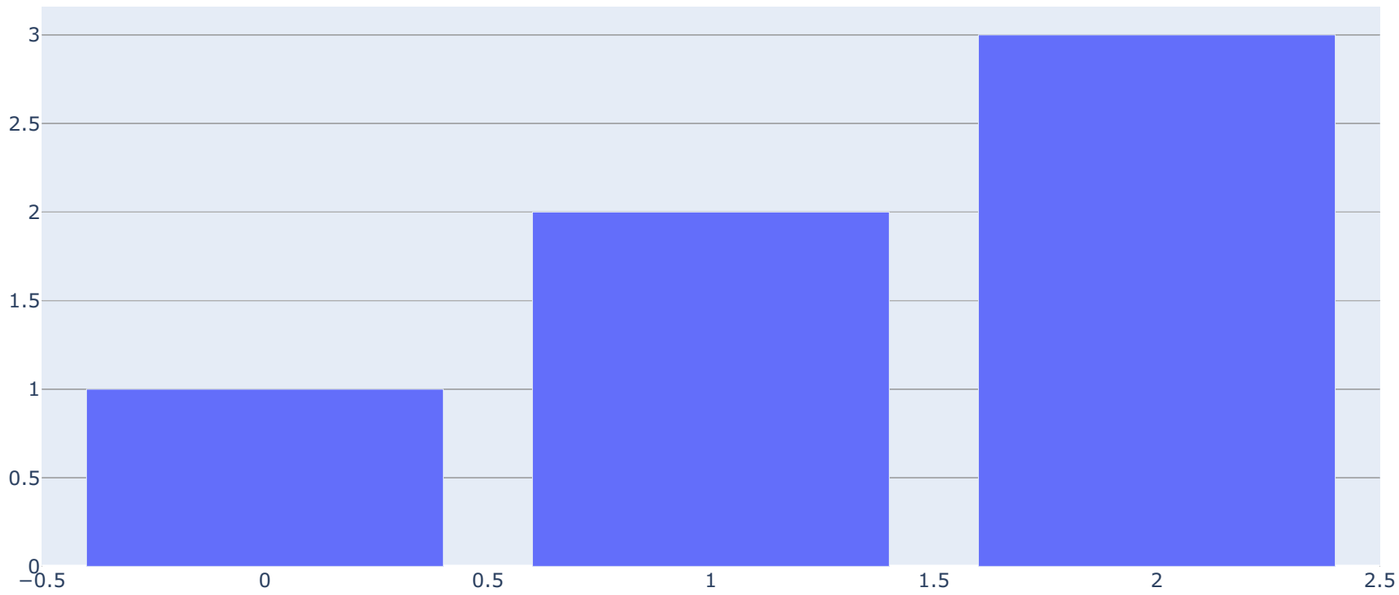
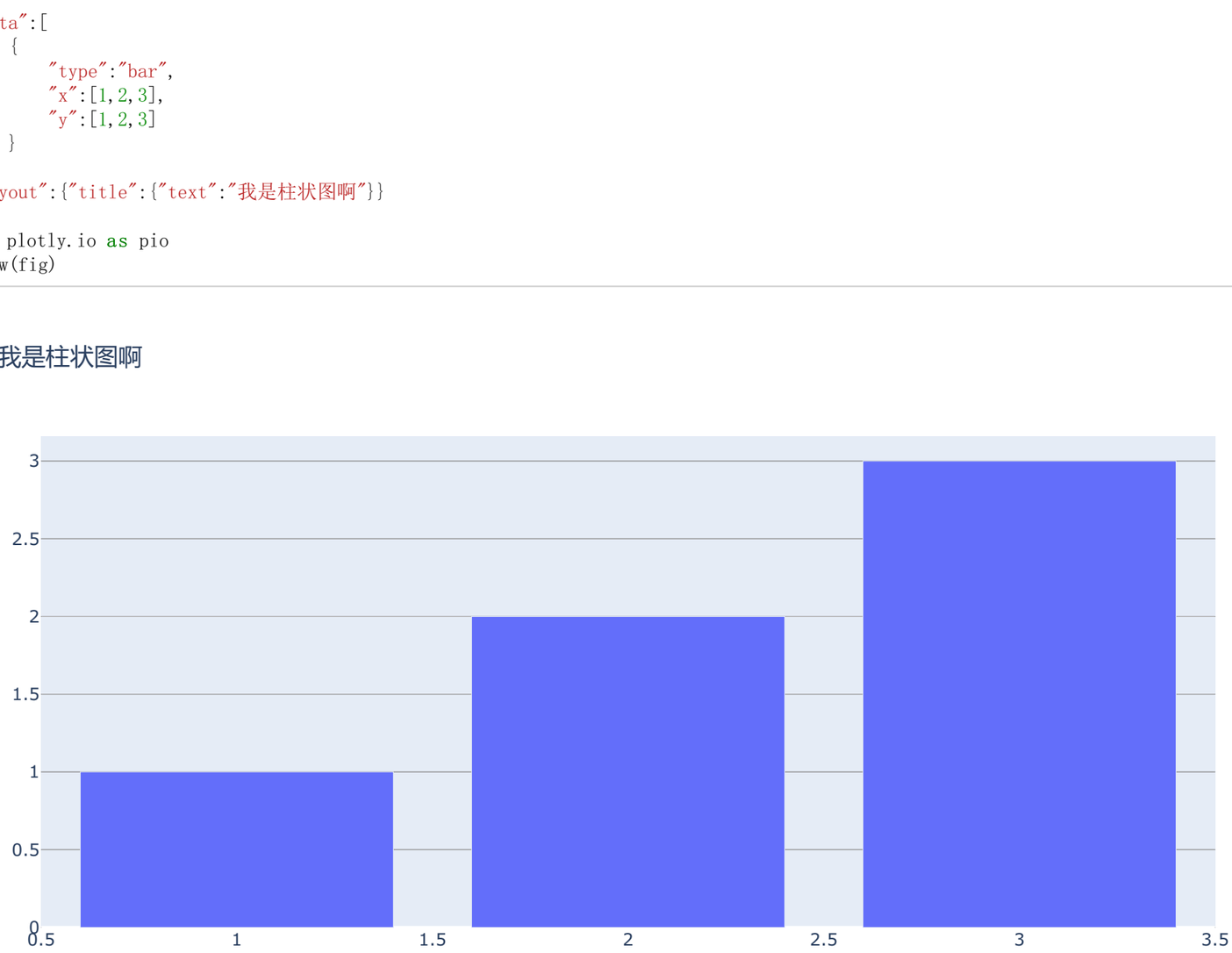
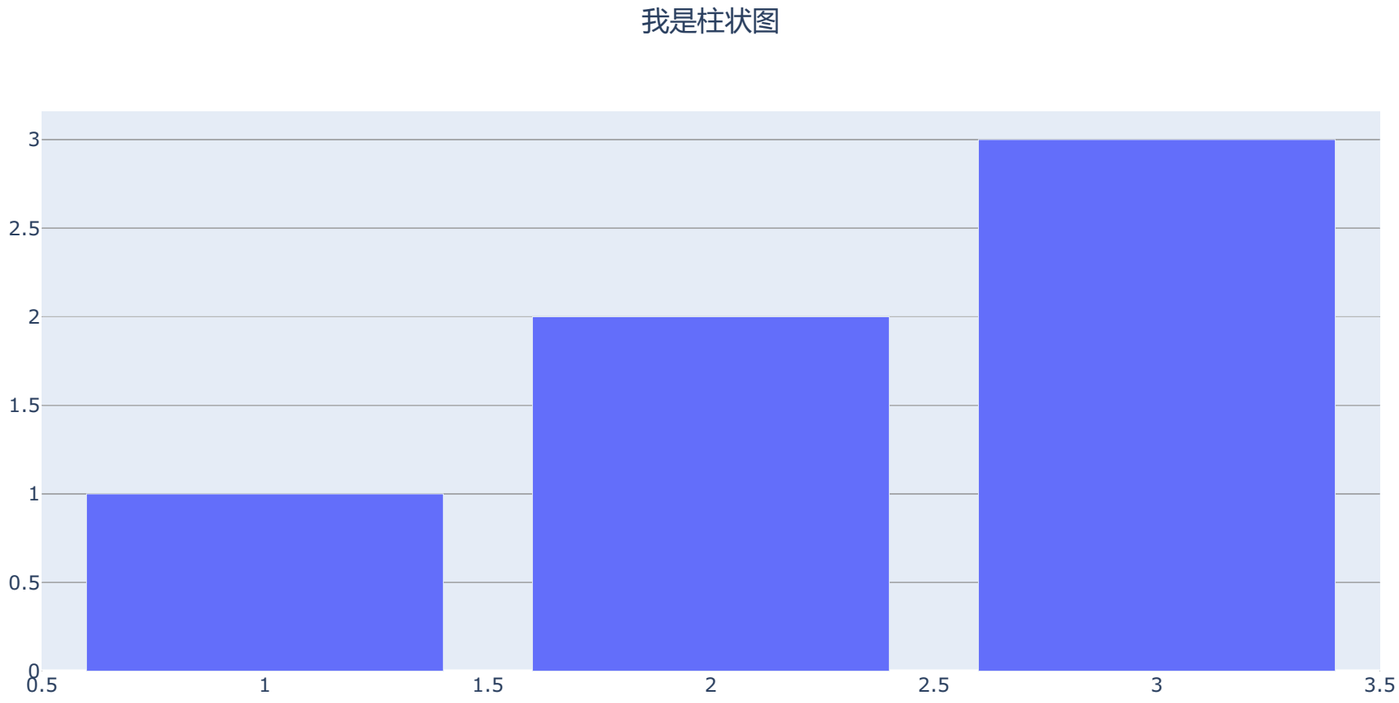
In [3]:
```
#pip install plotly==4.1.0
import plotly.graph_objects as go
fig = go.Figure(data=go.Bar(y = [1,2,3]))
fig.show()
```



In [4]:
```
fig = {
    "data":[
        {
            "type":"bar",
            "x":[1,2,3],
            "y":[1,2,3]
        }
    ],
    "layout":{"title":{"text":"我是柱状图啊"}}
}
import plotly.io as pio
pio.show(fig)
```



In [5]:
```
fig = go.Figure(
    data = [go.Bar(x=[1,2,3],y=[1,2,3])],
    layout = go.Layout(
        title = go.layout.Title(text="我是柱状图",x=0.5)
    )
)
fig.show()
```



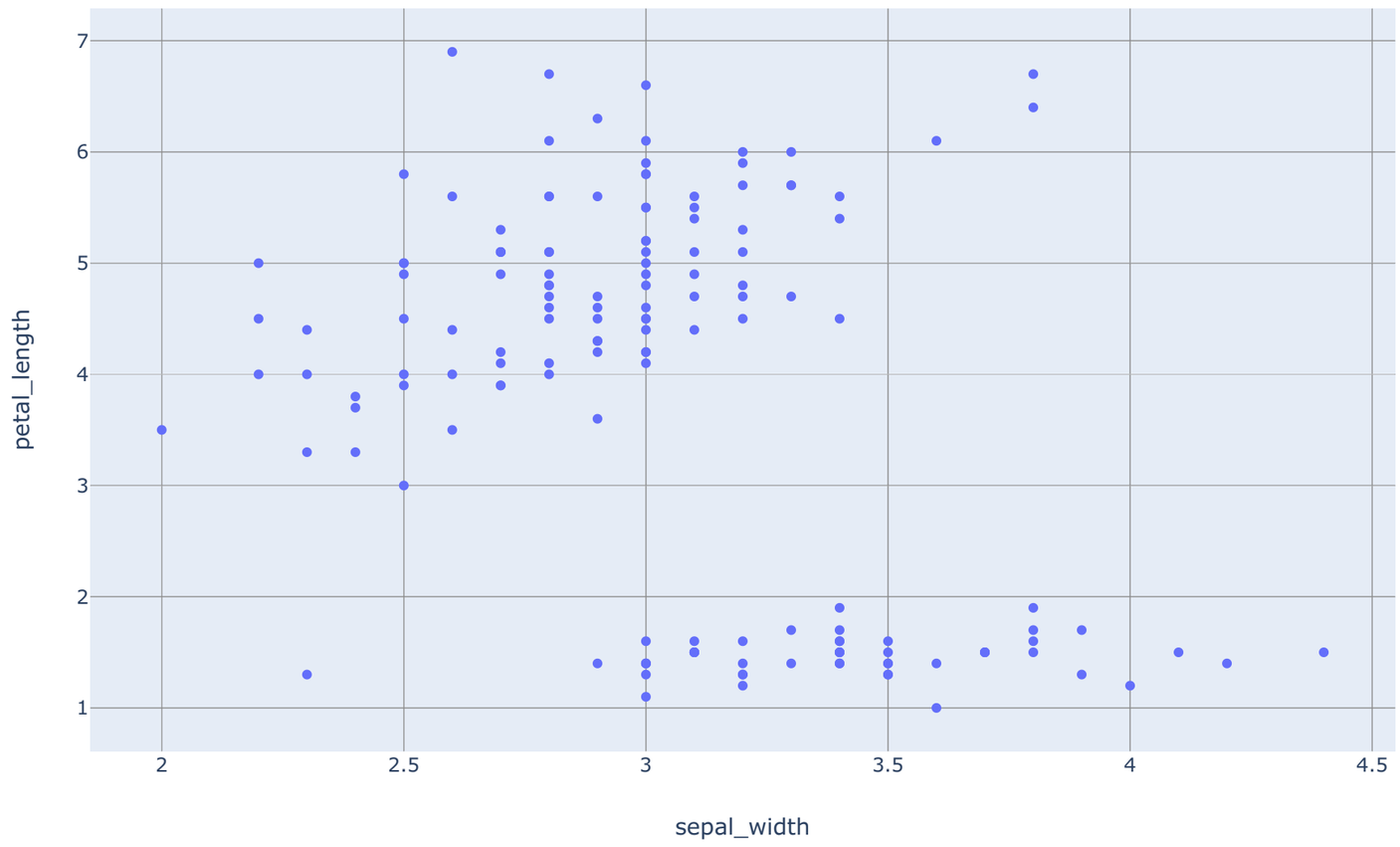
In [6]:
```
#散点图
import plotly.express as px
#导入数据
iris = px.data.iris()
print(iris)
print(type(iris))
```

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

     species_id
0             1
1             1
2             1
3             1
4             1
..          ...
145           3
146           3
147           3
148           3
149           3

[150 rows x 6 columns]
<class 'pandas.core.frame.DataFrame'>
```

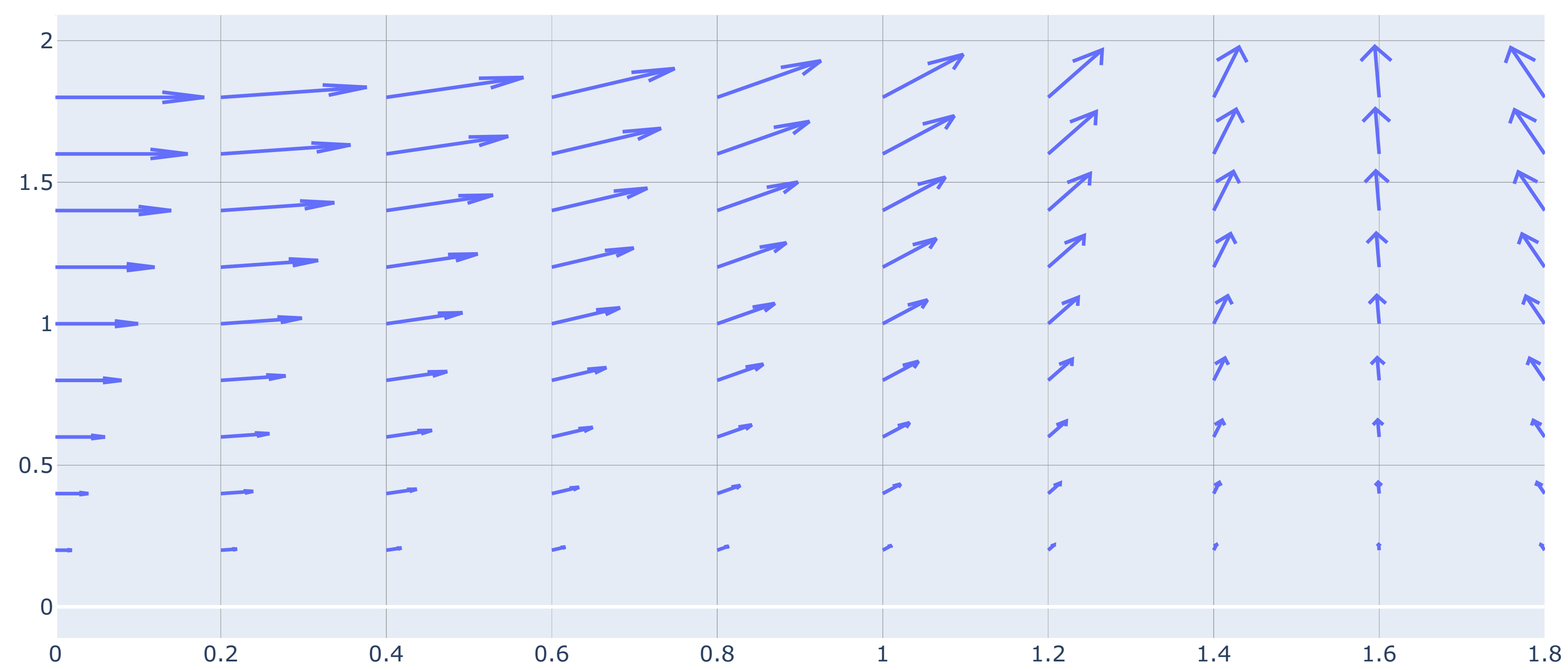
In [7]:
```
fig = px.scatter(iris,x="sepal_width",y="petal_length")
fig.show()
```



In [11]:

```python
#箭头图
import numpy as np
import plotly.figure_factory as ff
x1,y1 = np.meshgrid(np.arange(0,2,0.2),np.arange(0,2,0.2)) #https://blog.csdn.net/lllxxq141592654/article/details/81532855
print(np.arange(0,2,0.2))
u1 = np.cos(x1) * y1
v1 = np.sin(x1) * y1
print('x: \n',x1)
print('y: \n',y1)
print('u1: \n',u1)
print('v1: \n',v1)
ff.create_quiver(x1,y1,u1,v1).show()
```

```
[0.  0.2 0.4 0.6 0.8 1.  1.2 1.4 1.6 1.8]
x:
 [[0.  0.2 0.4 0.6 0.8 1.  1.2 1.4 1.6 1.8]
 [0.  0.2 0.4 0.6 0.8 1.  1.2 1.4 1.6 1.8]
 [0.  0.2 0.4 0.6 0.8 1.  1.2 1.4 1.6 1.8]
 [0.  0.2 0.4 0.6 0.8 1.  1.2 1.4 1.6 1.8]
 [0.  0.2 0.4 0.6 0.8 1.  1.2 1.4 1.6 1.8]
 [0.  0.2 0.4 0.6 0.8 1.  1.2 1.4 1.6 1.8]
 [0.  0.2 0.4 0.6 0.8 1.  1.2 1.4 1.6 1.8]
 [0.  0.2 0.4 0.6 0.8 1.  1.2 1.4 1.6 1.8]
 [0.  0.2 0.4 0.6 0.8 1.  1.2 1.4 1.6 1.8]
 [0.  0.2 0.4 0.6 0.8 1.  1.2 1.4 1.6 1.8]]
y:
 [[0.  0.  0.  0.  0.  0.  0.  0.  0.  0. ]
 [0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2]
 [0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4]
 [0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6]
 [0.8 0.8 0.8 0.8 0.8 0.8 0.8 0.8 0.8 0.8]
 [1.  1.  1.  1.  1.  1.  1.  1.  1.  1. ]
 [1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2]
 [1.4 1.4 1.4 1.4 1.4 1.4 1.4 1.4 1.4 1.4]
 [1.6 1.6 1.6 1.6 1.6 1.6 1.6 1.6 1.6 1.6]
 [1.8 1.8 1.8 1.8 1.8 1.8 1.8 1.8 1.8 1.8]]
u1:
 [[ 0.          0.          0.          0.          0.          0.
   0.          0.         -0.         -0.        ]
 [ 0.2         0.19601332  0.1842122   0.16506712  0.13934134  0.10806046
   0.07247155  0.03399343 -0.0058399  -0.04544042]
 [ 0.4         0.39202663  0.3684244   0.33013425  0.27868268  0.21612092
   0.1449431   0.06798686 -0.01167981 -0.09088084]
 [ 0.6         0.58803995  0.5526366   0.49520137  0.41802403  0.32418138
   0.21741465  0.10198029 -0.01751971 -0.13632126]
 [ 0.8         0.78405326  0.7368488   0.66026849  0.55736537  0.43224184
   0.2898862   0.13597371 -0.02335962 -0.18176168]
 [ 1.          0.98006658  0.92106099  0.82533561  0.69670671  0.54030231
   0.36235775  0.16996714 -0.02919952 -0.22720209]
 [ 1.2         1.17607989  1.10527319  0.99040274  0.83604805  0.64836277
   0.43482931  0.20396057 -0.03503943 -0.27264251]
 [ 1.4         1.37209321  1.28948539  1.15546986  0.97538939  0.75642323
   0.50730086  0.237954   -0.04087933 -0.31808293]
 [ 1.6         1.56810652  1.47369759  1.32053698  1.11473073  0.86448369
   0.57977241  0.27194743 -0.04671924 -0.36352335]
 [ 1.8         1.76411984  1.65790979  1.48560411  1.25407208  0.97254415
   0.65224396  0.30594086 -0.05255914 -0.40896377]]
v1:
 [[0.         0.         0.         0.         0.         0.
  0.         0.         0.         0.        ]
 [0.         0.03973387 0.07788367 0.11292849 0.14347122 0.1682942
  0.18640782 0.19708995 0.19991472 0.19476953]
 [0.         0.07946773 0.15576734 0.22585699 0.28694244 0.33658839
  0.37281563 0.39417989 0.39982944 0.38953905]
 [0.         0.1192016  0.23365101 0.33878548 0.43041365 0.50488259
  0.55922345 0.59126984 0.59974416 0.58430858]
 [0.         0.15893546 0.31153467 0.45171398 0.57388487 0.67317679
  0.74563127 0.78835978 0.79965888 0.7790781 ]
 [0.         0.19866933 0.38941834 0.56464247 0.71735609 0.84147098
  0.93203909 0.98544973 0.9995736  0.97384763]
 [0.         0.2384032  0.46730201 0.67757097 0.86082731 1.00976518
  1.1184469  1.18253968 1.19948832 1.16861716]
 [0.         0.27813706 0.54518568 0.79049946 1.00429853 1.17805938
  1.30485472 1.37962962 1.39940304 1.36338668]
 [0.         0.31787093 0.62306935 0.90342796 1.14776975 1.34635358
  1.49126254 1.57671957 1.59931776 1.55815621]
 [0.         0.3576048  0.70095302 1.01635645 1.29124096 1.51464777
  1.67767035 1.77380951 1.79923249 1.75292574]]
```

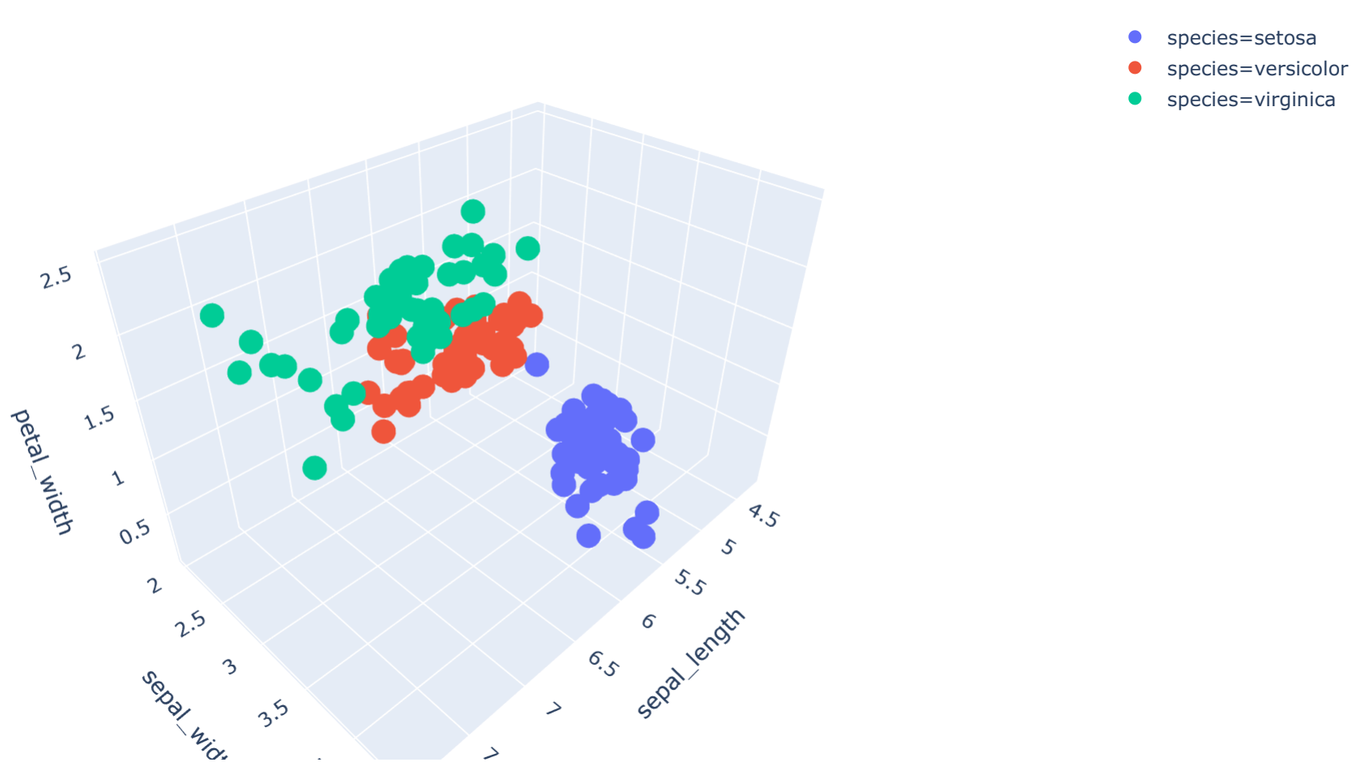


In [19]:
```
import plotly.express as px
iris = px.data.iris()
fig = px.scatter_3d(iris, x='sepal_length', y='sepal_width', z='petal_width',
              color='species')
print(iris)
fig.show()
```

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

     species_id
0             1
1             1
2             1
3             1
4             1
..          ...
145           3
146           3
147           3
148           3
149           3

[150 rows x 6 columns]
```



jupyter notebook 中查看源文件快捷键：

选中Shift+Tab（可以多按几次，就可以出来不同的选项）

In [10]:
```
#创建子图
from plotly.subplots import make_subplots
fig = make_subplots(rows=1,cols=3)
fig.add_trace(go.Scatter(y=[5,2,1],mode="lines"),row=1,col=1)
fig.add_trace(go.Scatter(x=[0,0.1,0.2],y=[5,2,1],mode="lines"),row=1,col=2)
fig.add_trace(go.Bar(y=[1,2,3]),row=1,col=3)
fig.add_trace(go.Scatter(y=[5,2,1],mode="lines"),row=1,col=3) #叠加在第三个图
fig.show()
```

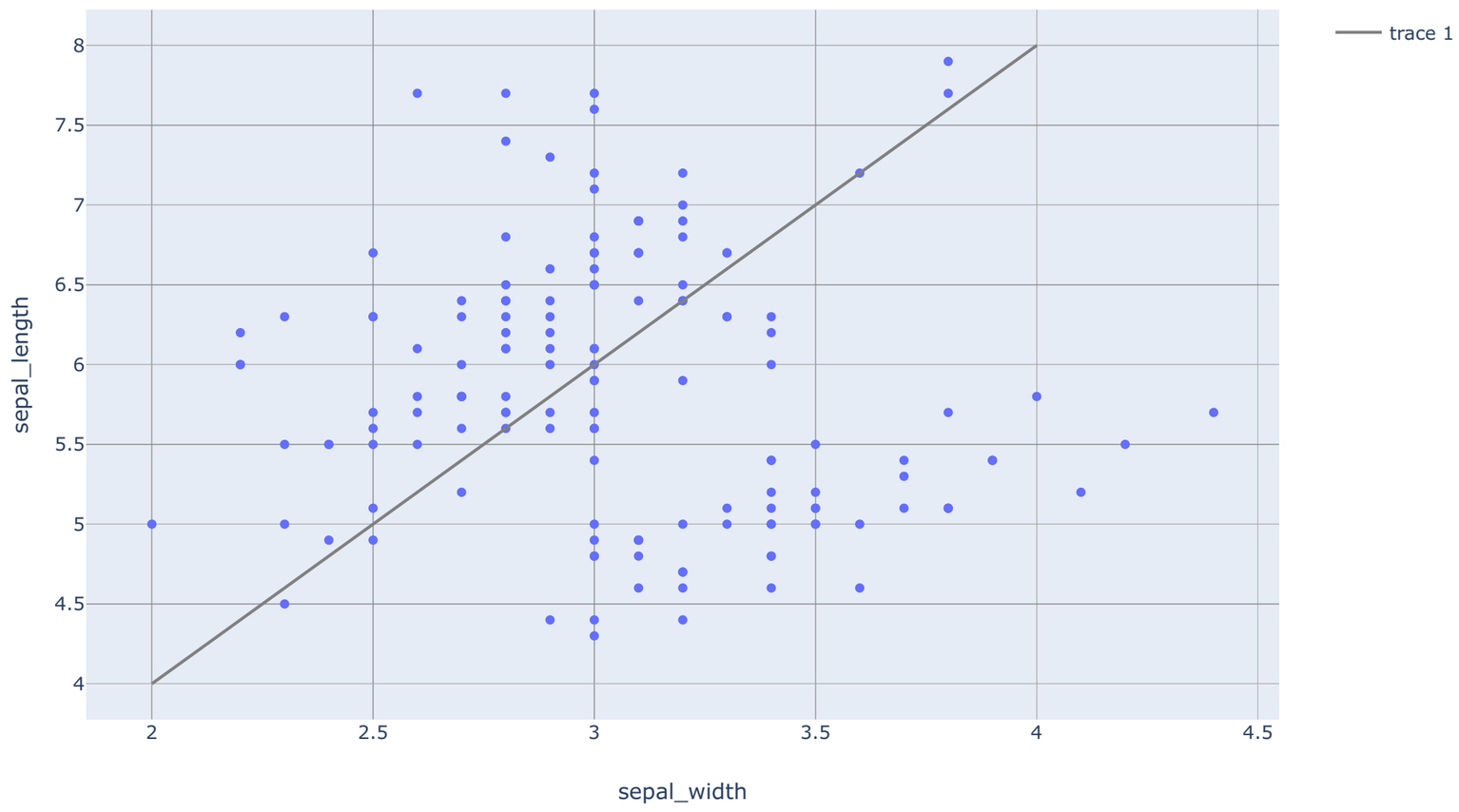
In [15]:
```
iris
```

Out[15]:

	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

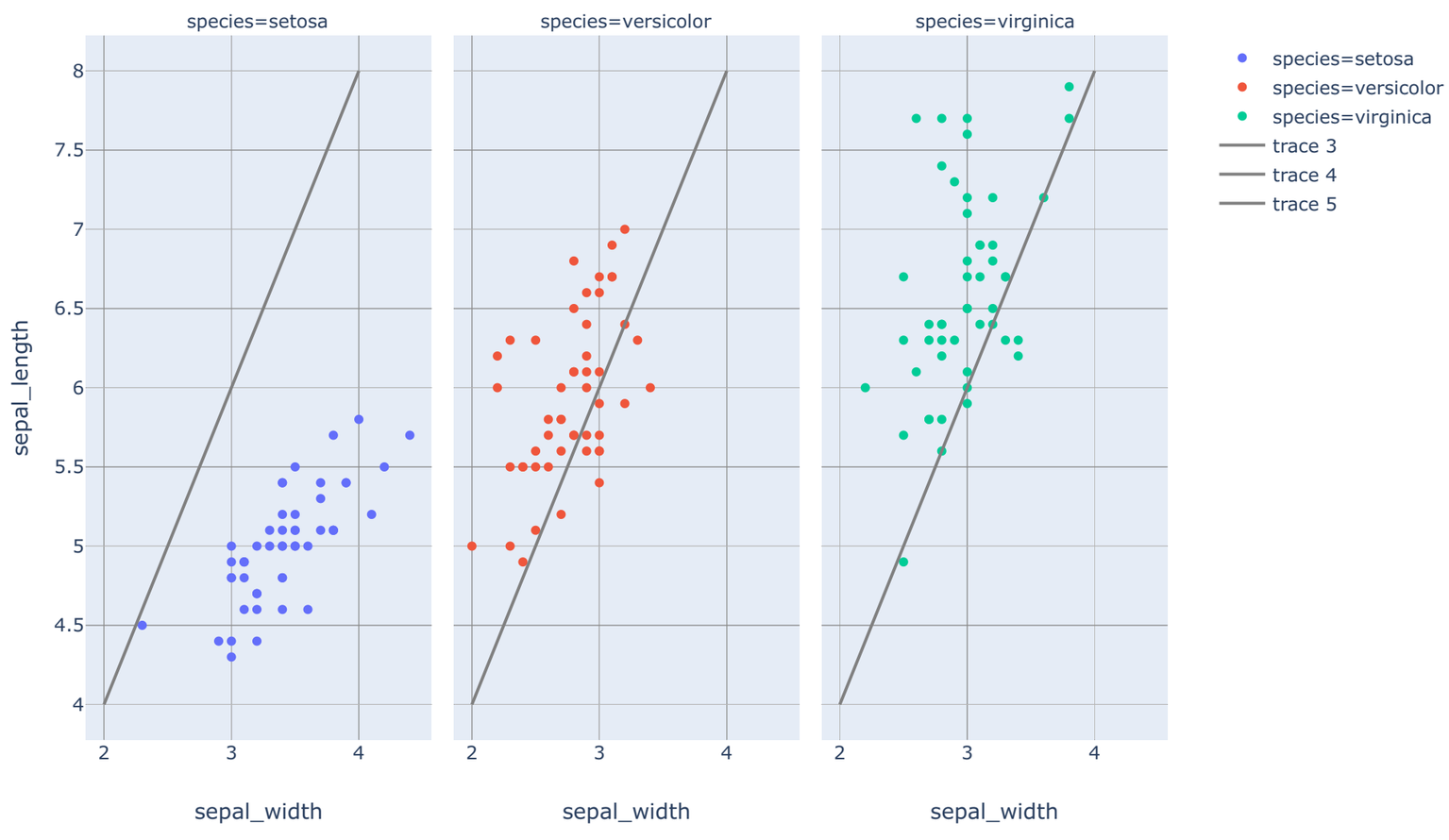
150 rows × 6 columns

In [24]:
```
fig = px.scatter(iris, y='sepal_length', x='sepal_width')
fig.add_trace(
    go.Scatter(
        x=[2,4],
        y=[4,8],
        mode = "lines",
        line=go.scatter.Line(color='gray')
    )
)
fig.show()
```



facet_col = "species" 与 fig = make_subplots(rows=1,cols=3)效果类似，但将iris数据拆开了

In [27]:
```
fig = px.scatter(iris, y='sepal_length', x='sepal_width',facet_col="species",color="species" )
reference_line = go.Scatter(x=[2,4],y=[4,8],mode='lines',line=go.scatter.Line(color='gray'))
fig.add_trace(reference_line,row=1,col=1)
fig.add_trace(reference_line,row=1,col=2)
fig.add_trace(reference_line,row=1,col=3)
fig.show()
```



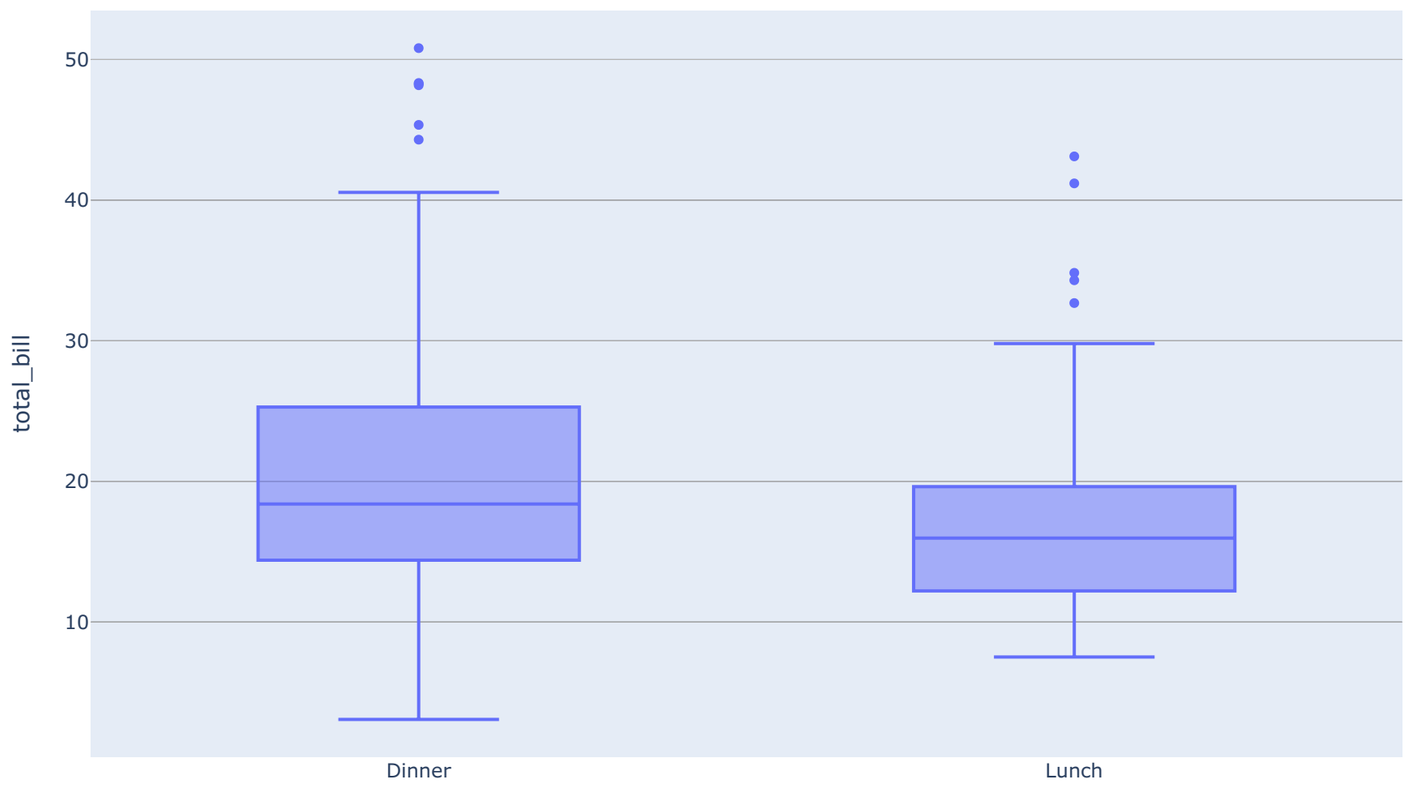
In [29]:
```
tips = px.data.tips()
tips
```

Out[29]:

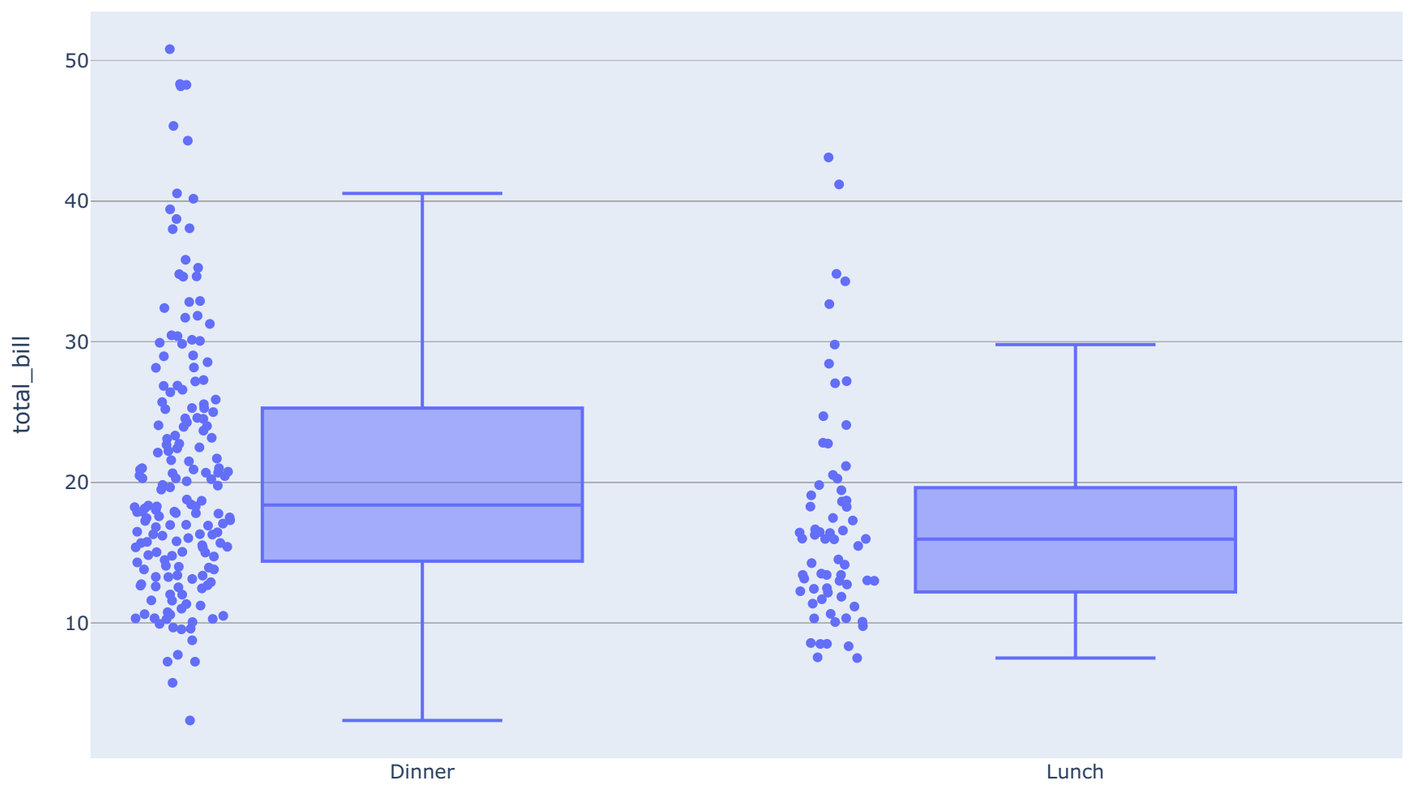
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female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

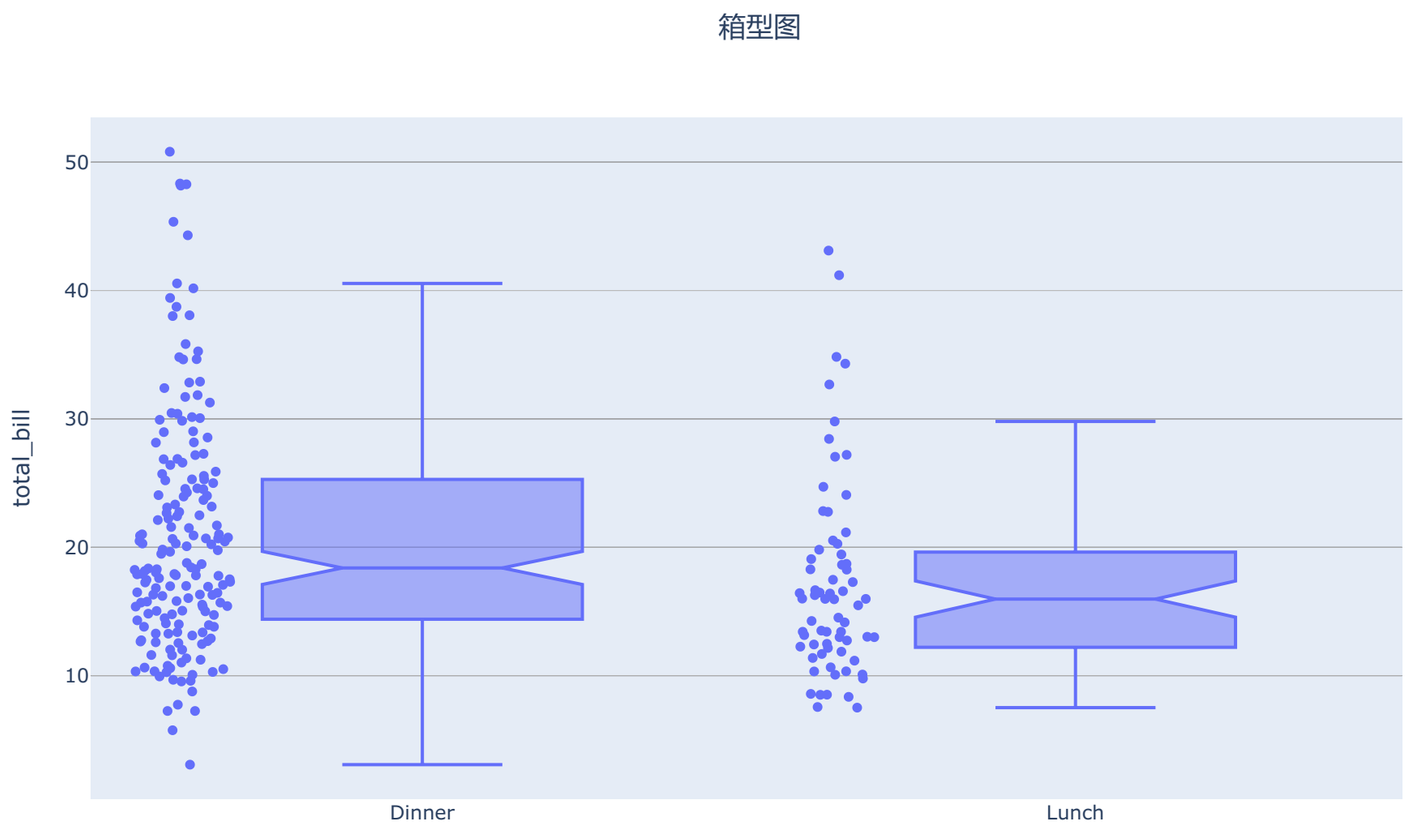
In [32]:
```
#箱型图
fig = px.box(tips,x='time',y='total_bill')
fig.show()
```



In [34]:
```
fig = px.box(tips,x='time',y='total_bill',points='all')#将分布打出来
fig.show()
```



In [42]:
```
fig = px.box(tips,x='time',y='total_bill',
             points='all',notched=True,
             title=go.layout.Title(text="箱型图",x=0.5))#
fig.show()
```

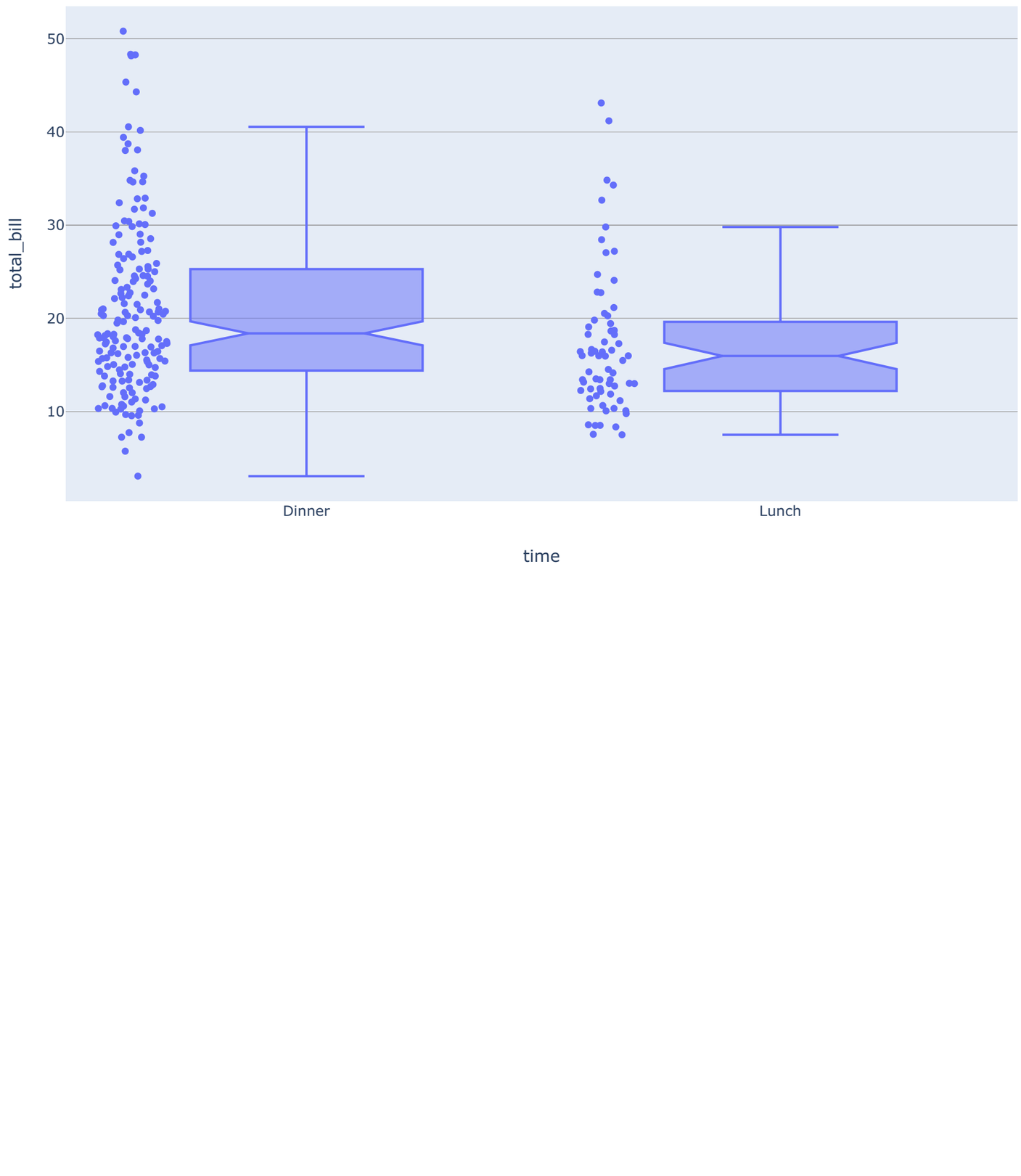


In [41]:
```
tips.head()
```

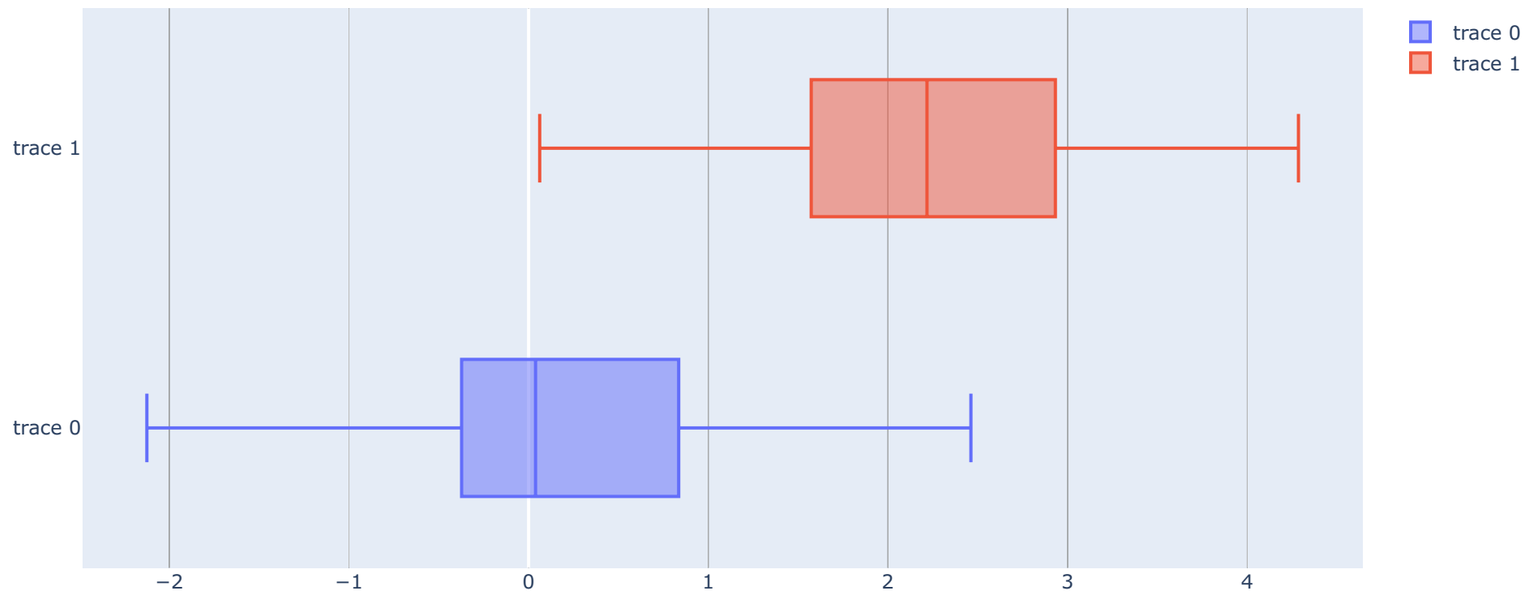
Out[41]:

	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 [45]:
```
fig = px.box(tips,x='time',y='total_bill',
             points='all',notched=True,
             title=go.layout.Title(text="箱型图",x=0.5),
             hover_data=["day"]      #把数据是哪一天的也显示出来（把day设为悬浮数据）
            )#
fig.show()
```



In [50]:
```
#没有y，箱型图就放倒了
x0 = np.random.randn(50)
x1 = np.random.randn(50) + 2
fig = go.Figure()
fig.add_trace(go.Box(x=x0))
fig.add_trace(go.Box(x=x1))
```



In [54]:
```
#饼图
fig = go.Figure(data=[
    go.Pie(labels=['a','b','c','d','e'],values=[20,30,50,10,40],hole=0.6)
])
fig.show()
```