绘图

In [1]:
```
import pandas as pd
import cufflinks as cf
from chinese_calendar import is_workday

import plotly
import plotly.express as px
import plotly.graph_objects as go
import plotly.figure_factory as ff
plotly.offline.init_notebook_mode()

import scipy
import scipy.cluster.hierarchy as sch

from sklearn.metrics import *

import chart_studio
import chart_studio.plotly as py
from chart_studio.plotly import plot, iplot
chart_studio.tools.set_credentials_file(username='xxx', api_key='yyy')  # 这里改成自己的用户名和 API key！！！

import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False

from IPython.display import HTML
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = 'all'
InteractiveShell.ast_node_interactivity = 'last'

import pylatex
import latexify
```

In [2]:
```
print(cf.getThemes())

cf.set_config_file(
    offline=True,
    world_readable=True,  #
    theme='white',        # 设置绘图风格
```

```
    # offline=False,        # 离线
)
```

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

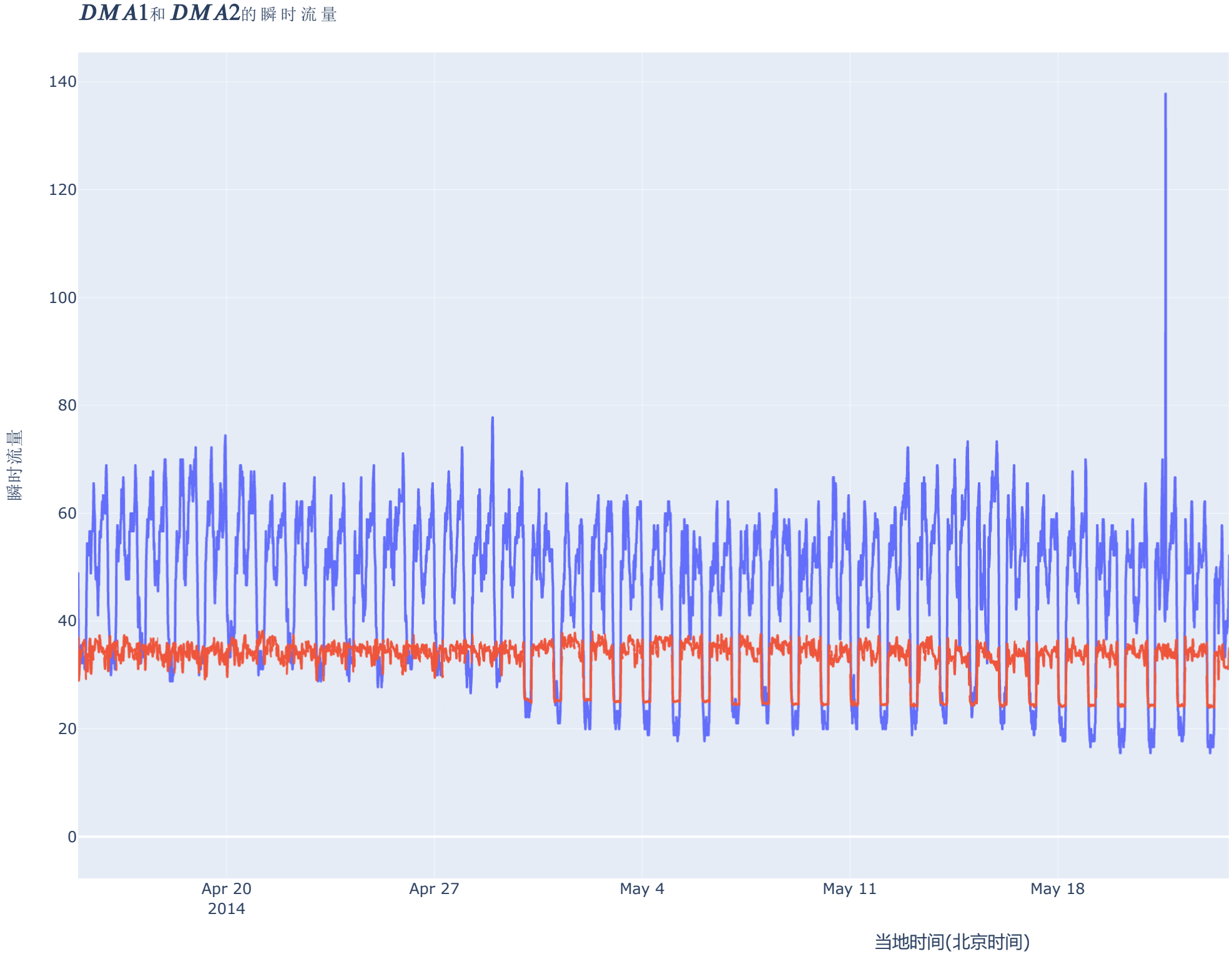
基本绘图

DMA1和DMA2的瞬时流量

In [3]:

```
InteractiveShell.ast_node_interactivity = 'last'

columns_name = ["当地时间(北京时间)", "DMA1", "DMA2"]
path = 'B1题附件.xls'

data = pd.read_excel(path)
data = pd.DataFrame(data.values, columns=columns_name)
data_time = data.set_index("当地时间(北京时间)")
# data.head()
# data_time.head()
fig = px.line(data, x="当地时间(北京时间)", y=["DMA1", 'DMA2'])
# layout = dict(
#     title=r'$DMA1和DMA2的瞬时流量$',
#     yaxis=dict(showticklabels=True,domain=[0, 0.85]),  # showticklables用来决定是否显示每个bar的旁注，domain用来设置y轴长度
#     yaxis2=dict(showline=True,showticklabels=False,linecolor='rgba(102, 102, 102, 0.8)',linewidth=2,domain=[0, 0.85]),
#     xaxis = dict(title = 'yourtitle',tickmode = 'array',tickvals = np.arange(1,16),ticktext=text)
#     xaxis=dict(zeroline=False,showline=False,showticklabels=True,showgrid=True,domain=[0, 0.42]),
#     xaxis2=dict(zeroline=False,showline=False,showticklabels=True,showgrid=True,domain=[0.47, 1],side='top',dtick=25),
#     legend=dict(x=0.029,y=1.038,font=dict(size=10) ),  #设置图例标志的大小和位置
#     margin=dict(l=200, r=20,t=70,b=70), # 设置bar旁注的长度、大小等
#     paper_bgcolor='rgb(248, 248, 255)', # 设置整个面板的背景色
#     plot_bgcolor='rgb(248, 248, 255)',  # 设置图像部份的背景色
# )
fig.update_layout(
    title='$DMA1和DMA2的瞬时流量$',
    yaxis=dict(title='$瞬时流量$'),
    showlegend=True,
    legend_title_text='',
#     xaxis=dict(title='yourtitle',tickmode = 'array',tickvals = np.arange(1,16),ticktext=text)
)
fig.update_layout(width=1600, height=800)
fig.write_image(r'./img/png/DMA1和DMA2的瞬时流量.png')
fig.write_image(r'./img/svg/DMA1和DMA2的瞬时流量.svg')
```

```
fig.write_html(r'./img/html/DMA1和DMA2的瞬时流量.html')
fig.show()
```

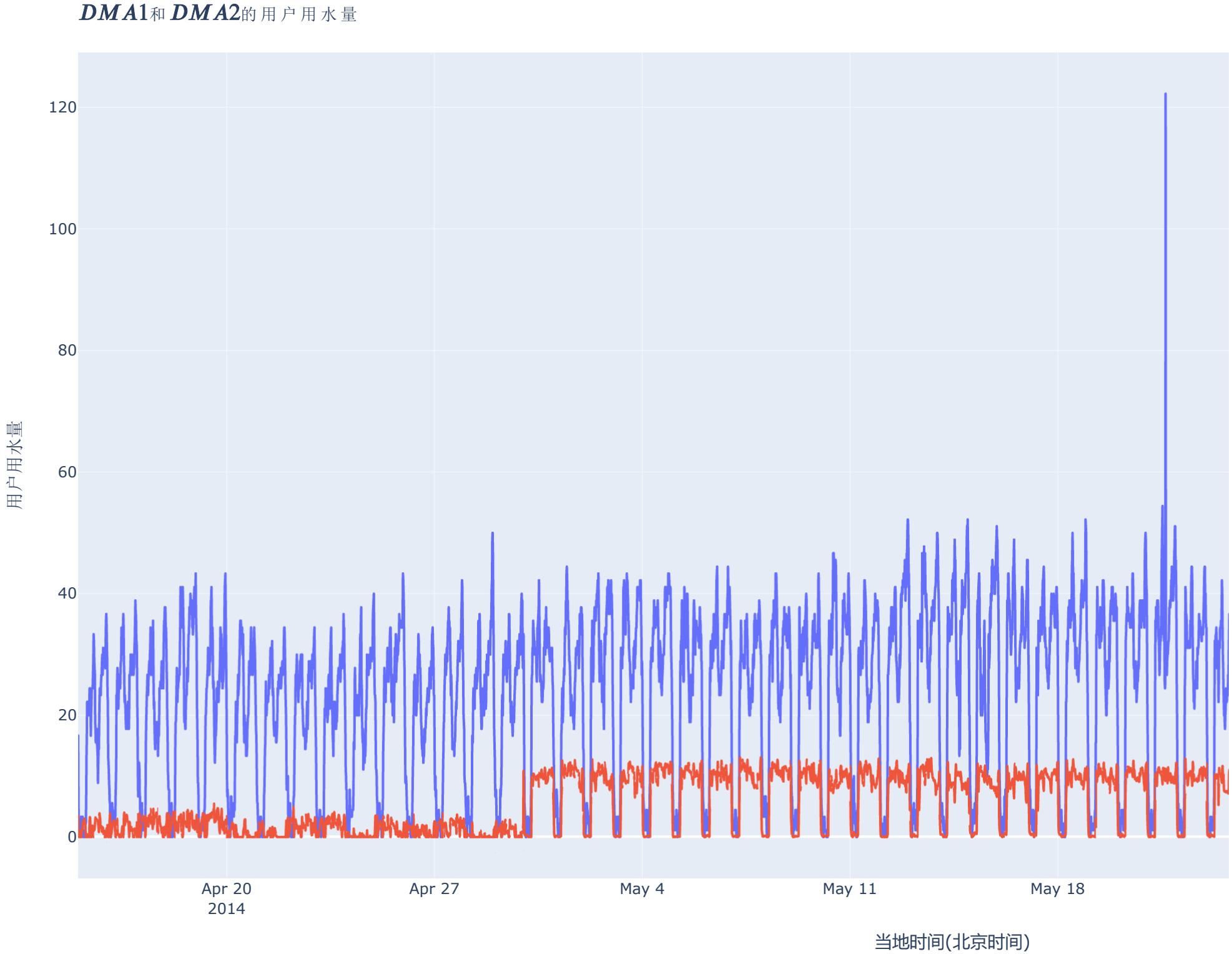

DMA1和DMA2的瞬时流量
瞬时流量
140
120
100
80
60
40
20
0
Apr 20
2014
Apr 27
May 4
May 11
May 18
当地时间(北京时间)

In [ ]:

DMA1和DMA2的用户用水量

In [ ]:

In [4]:
```
InteractiveShell.ast_node_interactivity = 'last'

path = '用户用水量数据.xlsx'
data_user = pd.read_excel(path)

fig = px.line(data_user, x="当地时间(北京时间)", y=["DMA1", 'DMA2'])
fig.update_layout(
    title='$DMA1和DMA2的用户用水量$',
    yaxis=dict(title='$用户用水量$'),
    showlegend=True,
    legend_title_text='',
)
fig.update_layout(width=1600, height=800)
fig.write_image(r'./img/png/DMA1和DMA2的用户用水量.png')
fig.write_image(r'./img/svg/DMA1和DMA2的用户用水量.svg')
fig.write_html(r'./img/html/DMA1和DMA2的用户用水量.html')
fig.show()
```

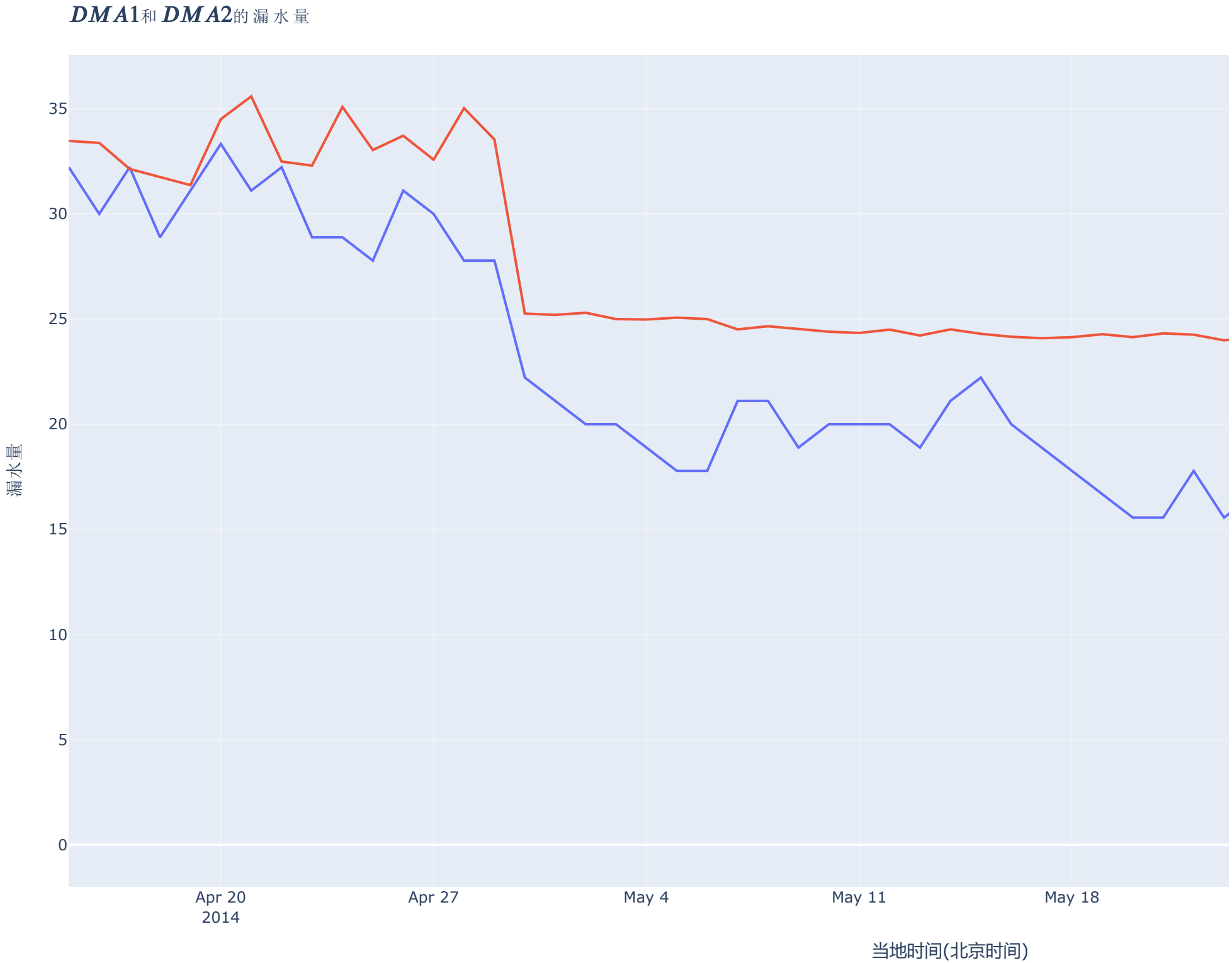

DMA1和DMA2的用户用水量
用户用水量
120
100
80
60
40
20
0
Apr 20
2014
Apr 27
May 4
May 11
May 18
当地时间(北京时间)

In [ ]:

DMA1和DMA2的漏水量

In [ ]:

In [5]:
```
InteractiveShell.ast_node_interactivity = 'last'

path = '按照日期处理后的数据.xlsx'
data_leaking = pd.read_excel(path, sheet_name='DMA1和DMA2的漏水量')

fig = px.line(data_leaking, x="当地时间(北京时间)", y=["DMA1", 'DMA2'])
fig.update_layout(
    title='$DMA1和DMA2的漏水量$',
    yaxis=dict(title='$漏水量$'),
    showlegend=True,
    legend_title_text='',
)
fig.update_layout(width=1600, height=800)
fig.write_image(r'./img/png/DMA1和DMA2的漏水量.png')
fig.write_image(r'./img/svg/DMA1和DMA2的漏水量.svg')
fig.write_html(r'./img/html/DMA1和DMA2的漏水量.html')
fig.show()
```


DMA1和DMA2的漏水量
漏水量
35
30
25
20
15
10
5
0
Apr 20
2014
Apr 27
May 4
May 11
May 18
当地时间(北京时间)

In [ ]:

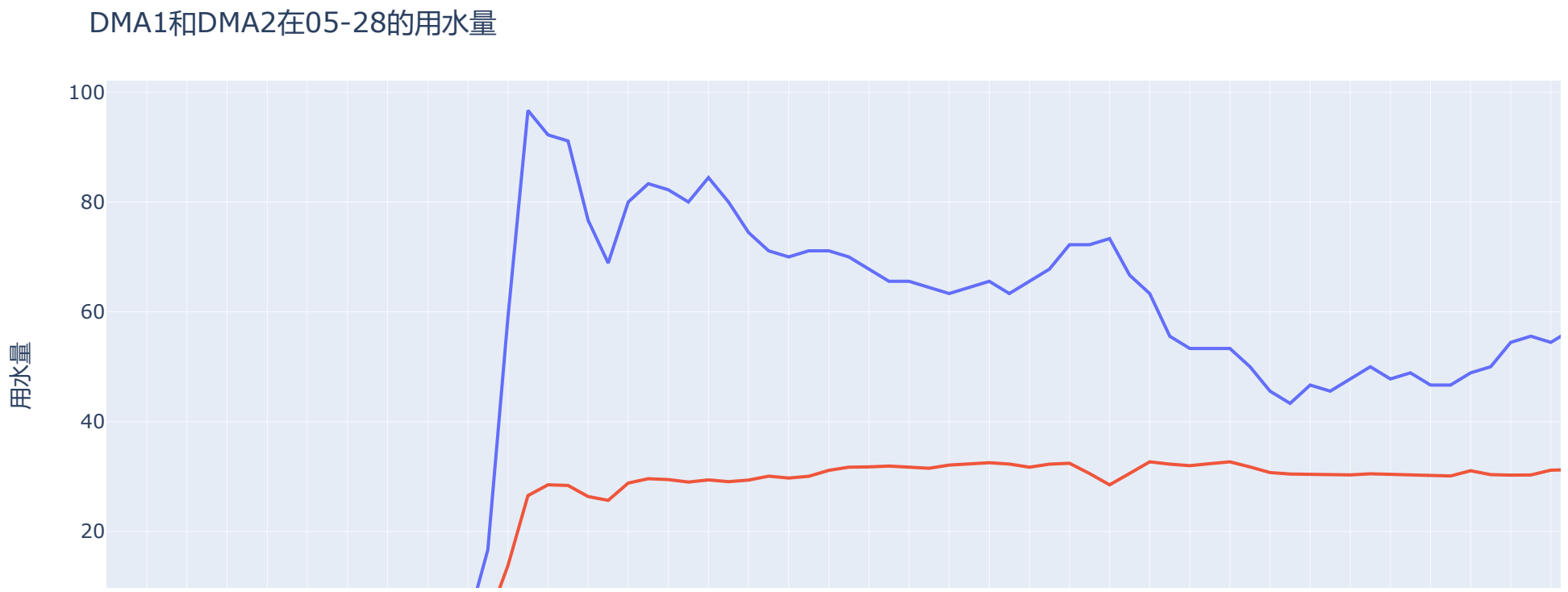
绘图 -- 日期

DMA1、2的5-28号的用水量

In [ ]:

In [6]:
```
# InteractiveShell.ast_node_interactivity = 'all'
InteractiveShell.ast_node_interactivity = 'last'

path = "按照日期处理后的数据.xlsx"
sheet = 'DMA1的用户用水量'
DMA1_user = pd.read_excel(path, sheet_name=sheet, index_col=0)
index=list(DMA1_user.index)
columns=list(DMA1_user.columns)
DMA1_user_5_28 = pd.DataFrame(DMA1_user.iloc[43, :]).T
DMA1_user_5_28

path = "按照日期处理后的数据.xlsx"
sheet = 'DMA2的用户用水量'
DMA2_user = pd.read_excel(path, sheet_name=sheet, index_col=0)
index=list(DMA2_user.index)
columns=list(DMA2_user.columns)
DMA2_user_5_28 = pd.DataFrame(DMA2_user.iloc[43, :]).T
DMA2_user_5_28

user_5_28 = pd.concat([DMA1_user_5_28, DMA2_user_5_28], ignore_index=True)
user_5_28.rename({0: "DMA1", 1: "DMA2"}, inplace=True)
user_5_28

fig = px.line(user_5_28.T)
fig.update_layout(
    title='DMA1和DMA2在05-28的用水量',
#     autosize=False,
#     width=800, height=750,
#     margin=dict(l=80, r=80, b=30, t=0),
    xaxis={"title": "时间"},
    yaxis={"title": "用水量"},
    legend_title_text='',
)
```

```
fig.write_html('./img/svg/DMA1和DMA2在05-28的用水量.html')
fig.write_image('./img/svg/DMA1和DMA2在05-28的用水量.svg')
fig.show()
```



In [ ]:

DMA1的瞬时流量

In [ ]:

In [7]:
```
InteractiveShell.ast_node_interactivity = 'last'

path = r"./按照日期处理后的数据.xlsx"
```

```
sheet = 'DMA1的瞬时流量'
z_data = pd.read_excel(path, sheet_name=sheet, index_col=0)

fig = go.Figure(data=[go.Surface(
    x=list(z_data.columns),
    y=list(z_data.index),
    z=z_data.values,
    colorbar=dict(title=r'DMA1的<br>瞬时流量'),
)])

fig.update_layout(
#     title='$DMA1的瞬时流量$',
    autosize=False,
    width=800, height=750,
    margin=dict(l=80, r=80, b=30, t=0),
    xaxis={"title": "日期"},
    yaxis={"title": "时间"},
)
fig.write_image('./img/png/DMA1的瞬时流量（月份）.png')
fig.write_image('./img/svg/DMA1的瞬时流量（月份）.svg')
fig.write_html('./img/html/DMA1的瞬时流量（月份）.html')
# fig.show()
```

3D曲线图从 `.ipynb` 文件导出 `.html` 文件有问题，建议到以下网址查看该图像 DMA1的瞬时流量（月份）.html，或者直接打开 `.ipynb` 文件，取消注释 `fig.show()` 运行查看

DMA2的瞬时流量

In [ ]:

In [8]:
```
InteractiveShell.ast_node_interactivity = 'last'

path = r"./按照日期处理后的数据.xlsx"
sheet = 'DMA2的瞬时流量'
z_data = pd.read_excel(path, sheet_name=sheet, index_col=0)

fig = go.Figure(data=[go.Surface(
    x=list(z_data.columns),
    y=list(z_data.index),
    z=z_data.values,
    colorbar=dict(title=r'DMA2的<br>瞬时流量'),
)])

fig.update_layout(
```

```
#     title='$DMA2的瞬时流量$',
    autosize=False,
    width=800, height=750,
    margin=dict(l=80, r=80, b=30, t=0),
    xaxis=dict(title='$日期$'),
    yaxis=dict(title='$时间$'),
#     zaxis=dict(title='$用户用水量$'),
)

fig.write_image('./img/png/DMA2的瞬时流量（月份）.png')
fig.write_image('./img/svg/DMA2的瞬时流量（月份）.svg')
fig.write_html('./img/html/DMA2的瞬时流量（月份）.html')
# fig.show()
```

3D曲线图从 `.ipynb` 文件导出 `.html` 文件有问题，建议到以下网址查看该图像 DMA2的瞬时流量（月份）.html，或者直接打开 `.ipynb` 文件，取消注释 `fig.show()` 运行查看

DMA1的用户用水量

In [ ]:

In [9]:
```
InteractiveShell.ast_node_interactivity = 'last'

path = r"./按照日期处理后的数据.xlsx"
sheet = 'DMA1的用户用水量'
z_data = pd.read_excel(path, sheet_name=sheet, index_col=0)

fig = go.Figure(data=[go.Surface(
    x=list(z_data.columns),
    y=list(z_data.index),
    z=z_data.values,
    colorbar=dict(title=r'DMA1的<br>用户用水量'),
)])

fig.update_layout(
#     title='$DMA1的用户用水量$',
    autosize=False,
    width=800, height=750,
    margin=dict(l=80, r=80, b=30, t=0),
    xaxis=dict(title='$日期$'),
    yaxis=dict(title='$时间$'),
#     zaxis=dict(title='$用户用水量$'),
)
```

```
fig.write_image('./img/png/DMA1的用户用水量（月份）.png')
fig.write_image('./img/svg/DMA1的用户用水量（月份）.svg')
fig.write_html('./img/html/DMA1的用户用水量（月份）.html')
# fig.show()
```

3D曲线图从 `.ipynb` 文件导出 `.html` 文件有问题，建议到以下网址查看该图像 DMA1的用户用水量（月份）.html，或者直接打开 `.ipynb` 文件，取消注释 `fig.show()` 运行查看

DMA2的用户用水量

In [ ]:

In [10]:
```
InteractiveShell.ast_node_interactivity = 'last'

path = r"./按照日期处理后的数据.xlsx"
sheet = 'DMA2的用户用水量'
z_data = pd.read_excel(path, sheet_name=sheet, index_col=0)

fig = go.Figure(data=[go.Surface(
    x=list(z_data.columns),
    y=list(z_data.index),
    z=z_data.values,
    colorbar=dict(title=r'DMA2的<br>用户用水量'),
)])

fig.update_layout(
#     title='$DMA2的用户用水量$',
    autosize=False,
    width=800, height=750,
    margin=dict(l=80, r=80, b=30, t=0),
    xaxis=dict(title='$日期$'),
    yaxis=dict(title='$时间$'),
#     zaxis=dict(title='$用户用水量$'),
)
fig.write_image('./img/png/DMA2的用户用水量（月份）.png')
fig.write_image('./img/svg/DMA2的用户用水量（月份）.svg')
fig.write_html('./img/html/DMA2的用户用水量（月份）.html')
# fig.show()
```

3D曲线图从 `.ipynb` 文件导出 `.html` 文件有问题，建议到以下网址查看该图像 DMA2的用户用水量（月份）.html，或者直接打开 `.ipynb` 文件，取消注释 `fig.show()` 运行查看

绘图 -- 星期

DMA1-星期瞬时流量

In [ ]:

In [11]:
```
InteractiveShell.ast_node_interactivity = 'last'

path = r"按照星期处理后的数据.xlsx"
sheet = 'DMA1的瞬时流量'
z_data = pd.read_excel(path, sheet_name=sheet, index_col=0)

fig = go.Figure(data=[go.Surface(
    x=list(z_data.columns),
    y=list(z_data.index),
    z=z_data.values,
    colorbar=dict(title=r'DMA1的<br>瞬时流量'),
)])

fig.update_layout(
#     title='$DMA1的用户用水量$',
    autosize=False,
    width=800, height=750,
    margin=dict(l=80, r=80, b=30, t=0),
    xaxis=dict(title='$日期$'),
    yaxis=dict(title='$时间$'),
#     zaxis=dict(title='$用户用水量$'),
)

fig.write_image('./img/png/DMA1的瞬时流量（星期）.png')
fig.write_image('./img/svg/DMA1的瞬时流量（星期）.svg')
fig.write_html('./img/html/DMA1的瞬时流量（星期）.html')
# fig.show()
# py.iplot(fig)
```

3D曲线图从 `.ipynb` 文件导出 `.html` 文件有问题，建议到以下网址查看该图像 DMA1的瞬时流量（星期）.html，或者直接打开 `.ipynb` 文件，取消注释 `fig.show()` 运行查看

DMA2-星期瞬时流量

In [ ]:

In [12]:
```
InteractiveShell.ast_node_interactivity = 'last'

path = r"按照星期处理后的数据.xlsx"
sheet = 'DMA2的瞬时流量'
z_data = pd.read_excel(path, sheet_name=sheet, index_col=0)

fig = go.Figure(data=[go.Surface(
    x=list(z_data.columns),
    y=list(z_data.index),
    z=z_data.values,
    colorbar=dict(title=r'DMA2的<br>瞬时流量'),
)])

fig.update_layout(
#     title='$DMA2的用户用水量$',
    autosize=False,
    width=800, height=750,
    margin=dict(l=80, r=80, b=30, t=0),
    xaxis=dict(title='$日期$'),
    yaxis=dict(title='$时间$'),
#     zaxis=dict(title='$用户用水量$'),
)
fig.write_image('./img/png/DMA2的瞬时流量（星期）.png')
fig.write_image('./img/svg/DMA2的瞬时流量（星期）.svg')
fig.write_html('./img/html/DMA2的瞬时流量（星期）.html')
# fig.show()
# py.iplot(fig)
```

3D曲线图从 `.ipynb` 文件导出 `.html` 文件有问题，建议到以下网址查看该图像 DMA2的瞬时流量（星期）.html，或者直接打开 `.ipynb` 文件，取消注释 `fig.show()` 运行查看

DMA1-星期用户用水流量

In [ ]:
```
```

In [13]:
```
InteractiveShell.ast_node_interactivity = 'last'

path = r"按照星期处理后的数据.xlsx"
sheet = 'DMA1的用户用水量'
z_data = pd.read_excel(path, sheet_name=sheet, index_col=0)

fig = go.Figure(data=[go.Surface(
    x=list(z_data.columns),
    y=list(z_data.index),
```

```
    z=z_data.values,
    colorbar=dict(title=r'DMA1的<br>用户用水量'),
)])

fig.update_layout(
#     title='$DMA1的用户用水量$',
    autosize=False,
    width=800, height=750,
    margin=dict(l=80, r=80, b=30, t=0),
    xaxis=dict(title='$日期$'),
    yaxis=dict(title='$时间$'),
#     zaxis=dict(title='$用户用水量$'),
)
fig.write_image('./img/png/DMA1的用户用水量（星期）.png')
fig.write_image('./img/svg/DMA1的用户用水量（星期）.svg')
fig.write_html('./img/html/DMA1的用户用水量（星期）.html')
# fig.show()
# py.iplot(fig)
```

3D曲线图从 `.ipynb` 文件导出 `.html` 文件有问题，建议到以下网址查看该图像 DMA1的用户用水量（星期）.html，或者直接打开 `.ipynb` 文件，取消注释 `fig.show()` 运行查看

DMA2-星期用户用水流量

In [ ]:

In [14]:
```
InteractiveShell.ast_node_interactivity = 'last'

path = r"按照星期处理后的数据.xlsx"
sheet = 'DMA2的用户用水量'
z_data = pd.read_excel(path, sheet_name=sheet, index_col=0)

fig = go.Figure(data=[go.Surface(
    x=list(z_data.columns),
    y=list(z_data.index),
    z=z_data.values,
    colorbar=dict(title=r'DMA2的<br>用户用水量'),
)])

fig.update_layout(
#     title='$DMA2的用户用水量$',
    autosize=False,
    width=800, height=750,
    margin=dict(l=80, r=80, b=30, t=0),
```

```
    xaxis=dict(title='$日期$'),
    yaxis=dict(title='$时间$'),
#     zaxis=dict(title='$用户用水量$'),
)
fig.write_image('./img/png/DMA2的用户用水量（星期）.png')
fig.write_image('./img/svg/DMA2的用户用水量（星期）.svg')
fig.write_html('./img/html/DMA2的用户用水量（星期）.html')
# fig.show()
# py.iplot(fig)
```

3D曲线图从 `.ipynb` 文件导出 `.html` 文件有问题，建议到以下网址查看该图像 DMA2的用户用水量（星期）.html，或者直接打开 `.ipynb` 文件，取消注释 `fig.show()` 运行查看

绘图 -- 工作日

DMA1-工作日瞬时流量

无

DMA2-工作日瞬时流量

无

DMA1-工作日用户用水量

In [15]:
```
# InteractiveShell.ast_node_interactivity = 'all'
InteractiveShell.ast_node_interactivity = 'last'

path = './按照星期处理后的数据.xlsx'
sheet = 'DMA1的用户用水量'
data_week_DMA1 = pd.read_excel(path, sheet, index_col=0)

data_weekday_DMA1 = data_week_DMA1.iloc[:5, :]
fig = go.Figure(data=[go.Surface(
    x=list(data_weekday_DMA1.columns),
    y=list(data_weekday_DMA1.index),
    z=data_weekday_DMA1.values,
    colorbar=dict(title=r'DMA1工作日<br>的用户用水量'),
)])
fig.update_layout(
    autosize=False,
```

```
    width=800, height=750,
    margin=dict(l=80, r=80, b=30, t=0),
    xaxis=dict(title='$日期$'),
    yaxis=dict(title='$时间$'),
)
fig.write_image('./img/png/DMA1工作日的用户用水量.png')
fig.write_image('./img/svg/DMA1工作日的用户用水量.svg')
fig.write_html('./img/html/DMA1工作日的用户用水量.html')
# py.iplot(fig)
# fig.show()

data_weekend_DMA1 = data_week_DMA1.iloc[5:, :]
fig = go.Figure(data=[go.Surface(
    x=list(data_weekend_DMA1.columns),
    y=list(data_weekend_DMA1.index),
    z=data_weekend_DMA1.values,
    colorbar=dict(title=r'DMA1非工作日<br>的用户用水量'),
)])

fig.update_layout(
    autosize=False,
    width=800, height=750,
    margin=dict(l=80, r=80, b=30, t=0),
    xaxis=dict(title='$日期$'),
    yaxis=dict(title='$时间$'),
)
fig.write_image('./img/png/DMA1非工作日的用户用水量.png')
fig.write_image('./img/svg/DMA1非工作日的用户用水量.svg')
fig.write_html('./img/html/DMA1非工作日的用户用水量.html')
# py.iplot(fig)
# fig.show()
```

3D曲线图从 `.ipynb` 文件导出 `.html` 文件有问题，建议到以下网址查看该图像 DMA1工作日的用户用水量.html、DMA1非工作日的用户用水量.html，或者直接打开 `.ipynb` 文件，取消注释 `fig.show()` 运行查看

DMA2-工作日用户用水量

In [ ]:

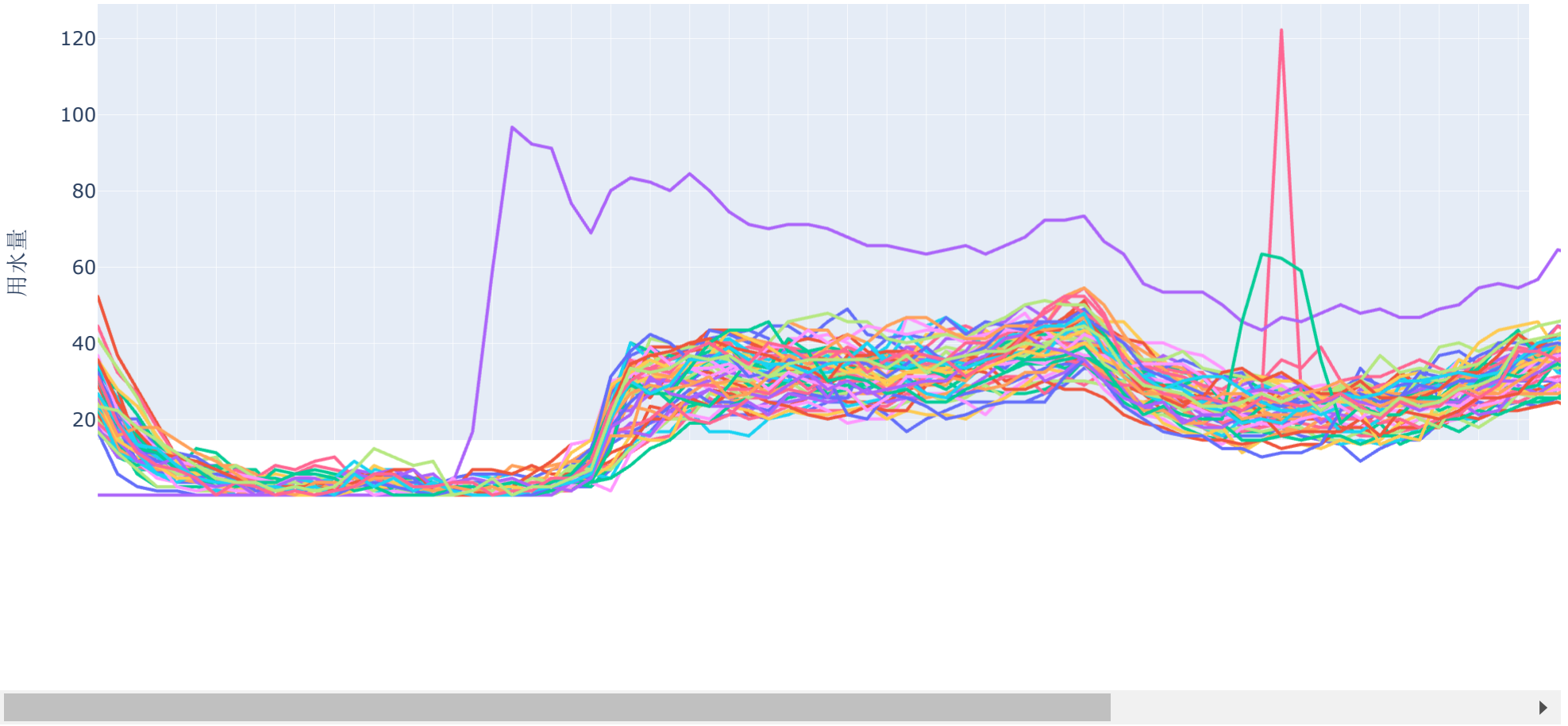
In [16]:
```
# InteractiveShell.ast_node_interactivity = 'all'
InteractiveShell.ast_node_interactivity = 'last'

path = './按照星期处理后的数据.xlsx'
sheet = 'DMA2的用户用水量'
```

```
data_week_DMA2 = pd.read_excel(path, sheet, index_col=0)

data_weekday_DMA2 = data_week_DMA2.iloc[:5, :]
fig = go.Figure(data=[go.Surface(
    x=list(data_weekday_DMA2.columns),
    y=list(data_weekday_DMA2.index),
    z=data_weekday_DMA2.values,
    colorbar=dict(title=r'DMA2工作日<br>的用户用水量'),
)])
fig.update_layout(
    autosize=False,
    width=800, height=750,
    margin=dict(l=80, r=80, b=30, t=0),
    xaxis=dict(title='$日期$'),
    yaxis=dict(title='$时间$'),
)
fig.write_image('./img/png/DMA2工作日的用户用水量.png')
fig.write_image('./img/svg/DMA2工作日的用户用水量.svg')
fig.write_html('./img/html/DMA2工作日的用户用水量.html')
# py.iplot(fig)
# fig.show()

data_weekend_DMA2 = data_week_DMA2.iloc[5:, :]
fig = go.Figure(data=[go.Surface(
    x=list(data_weekend_DMA2.columns),
    y=list(data_weekend_DMA2.index),
    z=data_weekend_DMA2.values,
    colorbar=dict(title=r'DMA2非工作日<br>的用户用水量'),
)])

fig.update_layout(
    autosize=False,
    width=800, height=750,
    margin=dict(l=80, r=80, b=30, t=0),
    xaxis=dict(title='$日期$'),
    yaxis=dict(title='$时间$'),
)
fig.write_image('./img/png/DMA2非工作日的用户用水量.png')
fig.write_image('./img/svg/DMA2非工作日的用户用水量.svg')
fig.write_html('./img/html/DMA2非工作日的用户用水量.html')
# py.iplot(fig)
# fig.show()
```

3D曲线图从 `.ipynb` 文件导出 `.html` 文件有问题，建议到以下网址查看该图像 DMA2工作日的用户用水量.html、DMA2非工作日的用户用水量.html，或者直接打开 `.ipynb` 文件，取消注释 `fig.show()` 运行查看

绘图 -- 问题2：类的趋势图

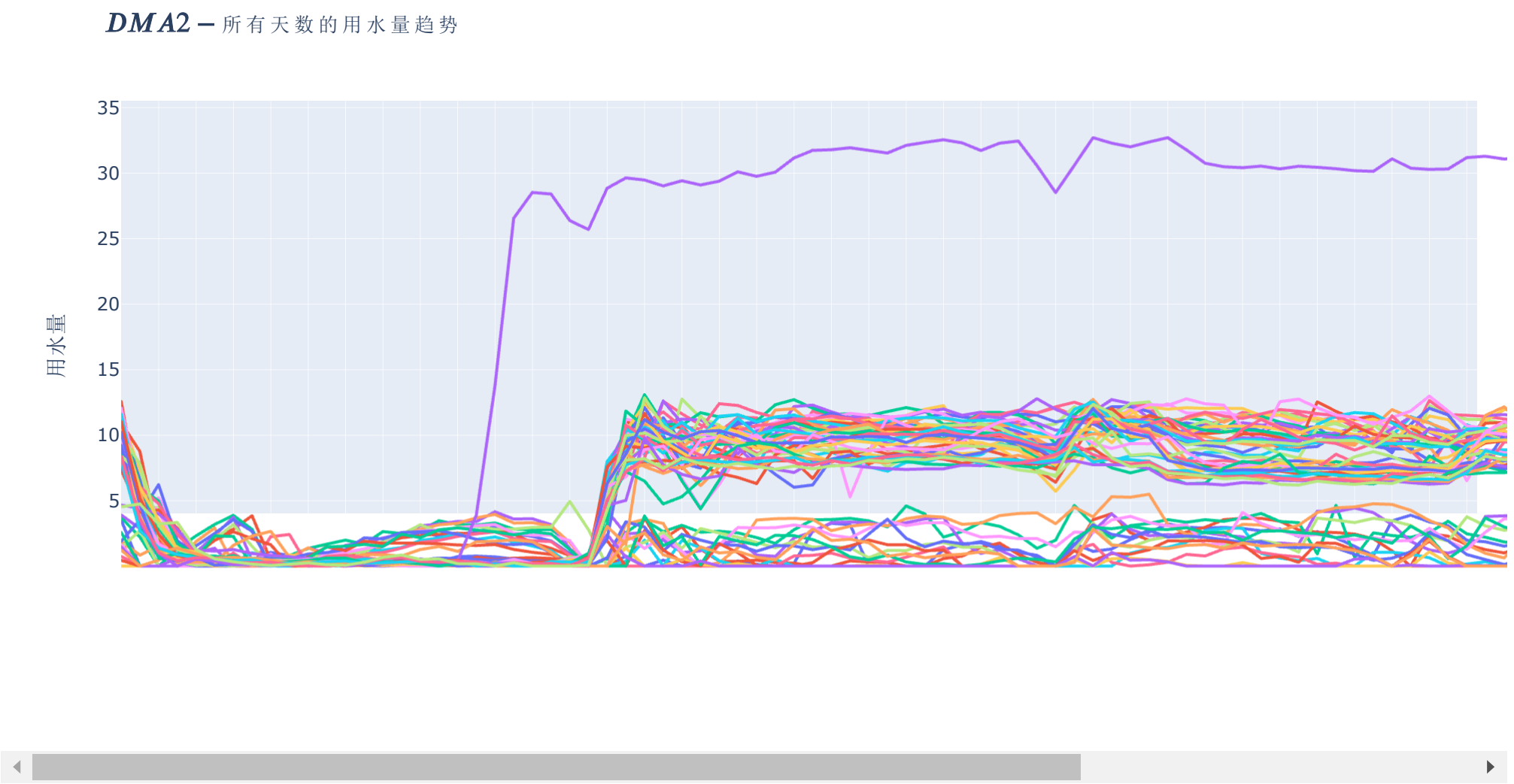
In [17]:
```
path = './按照日期处理后的数据.xlsx'
sheet = 'DMA1的用户用水量'
DMA1_data = pd.read_excel(path, sheet_name=sheet, index_col=0)
DMA1_data.index = DMA1_data.index.strftime("%m-%d")
DMA1_data
# DMA1_data.T.iplot(title='$DMA1-所有天数的用水量趋势$')

fig = px.line(
    DMA1_data.T,
    title='$DMA1-所有天数的用水量趋势$'
)
fig.update_layout(
    xaxis={"title": "$时间$"},
    yaxis={"title": "$用水量$"},
    legend_title_text='$日期$',
)
fig.write_image('./img/svg/DMA1-所有天数的用水量趋势.svg')
fig.show()
```



In [ ]:

In [18]:
```
path = './按照日期处理后的数据.xlsx'
sheet = 'DMA2的用户用水量'
DMA2_data = pd.read_excel(path, sheet_name=sheet, index_col=0)
DMA2_data.index = DMA2_data.index.strftime("%m-%d")
DMA2_data
# DMA2_data.T.iplot(title='$DMA2-所有天数的用水量趋势$')

fig = px.line(
    DMA2_data.T,
    title='$DMA2-所有天数的用水量趋势$'
```

```
)
fig.update_layout(
    xaxis={"title": "$时间$"},
    yaxis={"title": "$用水量$"},
    legend_title_text='$日期$',
)
fig.write_image('./img/svg/DMA2-所有天数的用水量趋势.svg')
fig.show()
```



DMA1-类1

In [19]:
```
DMA1_class1_April = [20, 27, 21, 23, 15, 24, 22, 26, 16, 17, 25, 29, 19, 28]
```

```
DMA1_class1_May = [1]
DMA1_class1 = [f'04-{i}' for i in DMA1_class1_April] + [f'05-{i}' for i in DMA1_class1_May]

DMA1_data_class1_mask = DMA1_data.apply(lambda x: x.name in DMA1_class1, axis=1)
DMA1_data_class1 = DMA1_data[DMA1_data_class1_mask]
DMA1_data_class1
```

Out[19]:

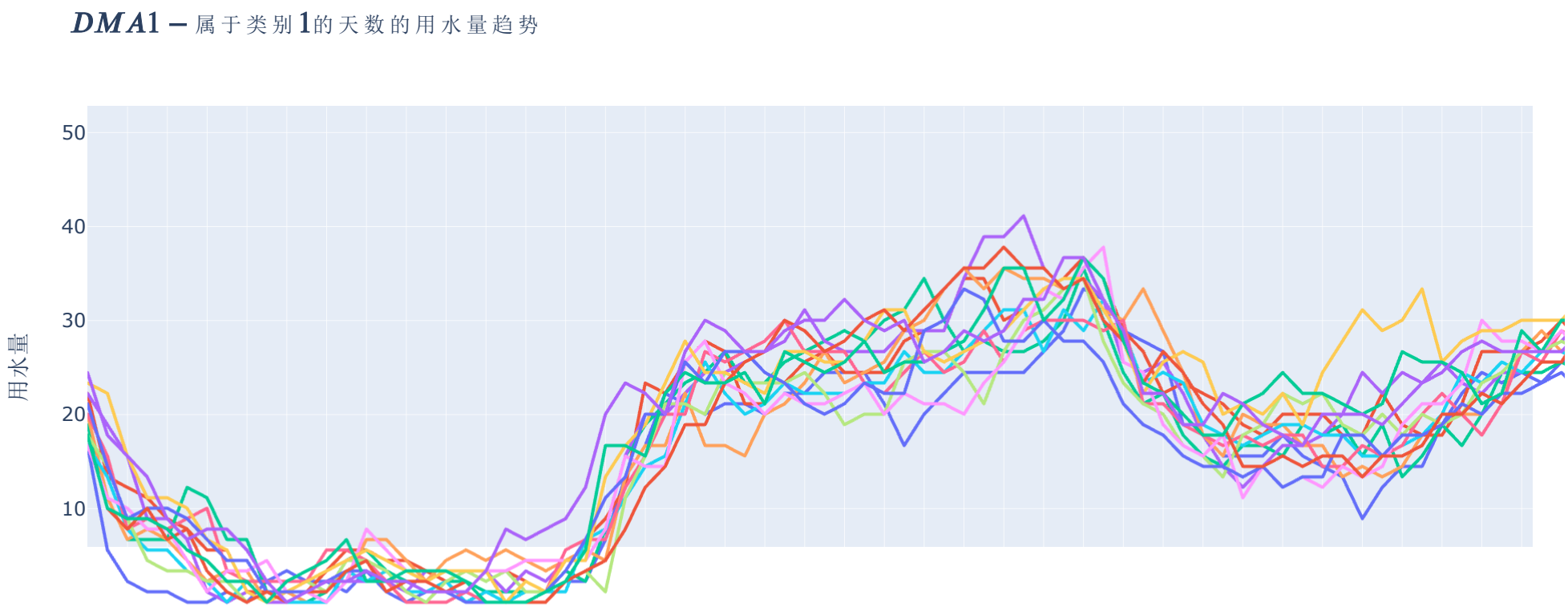
	0:0:00	0:15:00	0:30:00	0:45:00	1:0:00	1:15:00	1:30:00	1:45:00	2:0:00	2:15:00	...	21:30:00	21:45:00	22:0:00	22:15:00	22:30:00	22:45:00	23:0:00	23
04-15	16.67	5.56	2.22	1.11	1.11	0.00	0.00	1.11	0.00	2.22	...	28.89	28.89	27.78	32.22	35.56	36.67	34.45	
04-16	22.22	13.33	12.22	11.11	8.89	7.78	5.56	5.56	1.11	1.11	...	31.11	32.22	35.56	38.89	36.67	36.67	32.22	
04-17	18.89	11.11	6.67	6.67	6.67	12.22	11.11	6.67	6.67	1.11	...	35.56	35.56	37.78	36.67	37.78	34.45	34.45	
04-19	22.22	18.89	15.56	8.89	8.89	6.67	1.11	0.00	1.11	0.00	...	26.67	28.89	31.11	36.67	41.11	42.22	43.33	
04-20	20.00	11.11	6.67	7.78	6.67	4.45	2.23	3.34	3.34	0.00	...	27.78	28.89	32.23	34.45	32.23	31.11	28.89	
04-21	16.67	13.33	7.78	5.56	5.56	3.33	2.22	0.00	2.22	2.22	...	26.67	27.78	27.78	32.22	34.45	34.45	32.22	
04-22	20.00	15.56	7.78	8.89	7.78	8.89	10.00	3.34	2.22	2.22	...	28.89	28.89	27.78	31.11	31.11	33.34	34.45	
04-23	17.78	14.44	8.89	4.44	3.33	3.33	2.22	2.22	0.00	1.11	...	31.11	28.89	32.22	31.11	36.67	34.44	34.44	
04-24	16.67	11.11	10.00	7.78	7.78	4.44	1.11	3.33	3.33	4.44	...	30.00	30.00	31.11	36.67	37.78	38.89	40.00	
04-25	23.33	22.22	15.55	11.11	11.11	10.00	6.66	5.55	1.11	0.00	...	34.44	34.44	34.44	36.66	43.33	42.22	40.00	
04-26	21.11	14.45	8.89	10.00	10.00	8.89	6.67	4.45	4.45	1.11	...	27.78	26.67	33.33	33.33	34.45	32.22	31.11	
04-27	17.78	10.00	7.78	10.00	6.67	7.78	3.33	1.11	0.00	1.11	...	31.11	36.67	38.89	42.22	41.11	35.56	30.00	
04-28	17.78	10.00	8.89	8.89	7.78	5.55	4.44	2.22	2.22	0.00	...	35.55	34.44	36.66	41.11	42.22	48.89	50.00	
04-29	24.44	17.78	15.55	13.33	8.89	6.66	7.78	7.78	5.55	2.22	...	32.22	33.33	36.66	38.89	40.00	37.78	38.89	

14 rows × 96 columns

In [ ]:

In [20]:
```
# DMA1_data_class1.T.iplot(title='$DMA1-属于类别1的天数的用水量趋势$')
```

```
fig = px.line(
    DMA1_data_class1.T,
    title='$DMA1-属于类别1的天数的用水量趋势$'
)
fig.update_layout(
    xaxis={"title": "$时间$"},
    yaxis={"title": "$用水量$"},
    legend_title_text='$日期$',
)
fig.write_image('./img/svg/DMA1-属于类别1的天数的用水量趋势.svg')
fig.show()
```



In [ ]:

DMA1-类2

In [ ]:

In [21]:
```
DMA1_class2_May = [25, 27]
DMA1_class2 = [f'05-{i}' for i in DMA1_class2_May]
DMA1_class2 = ['05-25', '05-27']

DMA1_data_class2_mask = DMA1_data.apply(lambda x: x.name in DMA1_class2, axis=1)
DMA1_data_class2 = DMA1_data[DMA1_data_class2_mask]
DMA1_data_class2
```

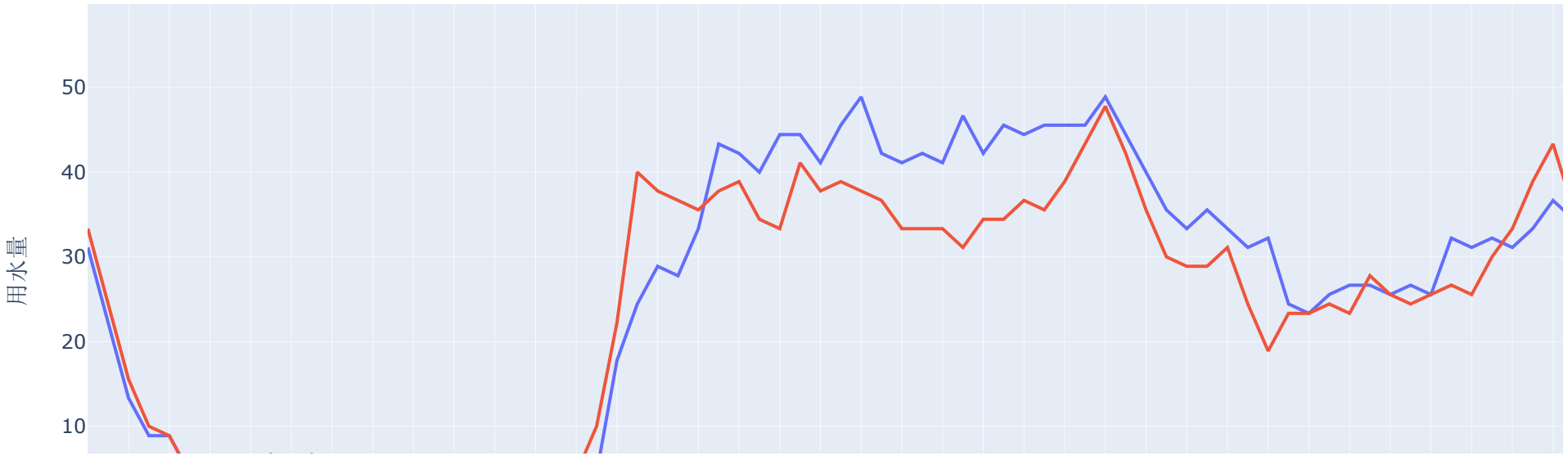
Out[21]:

	0:0:00	0:15:00	0:30:00	0:45:00	1:0:00	1:15:00	1:30:00	1:45:00	2:0:00	2:15:00	...	21:30:00	21:45:00	22:0:00	22:15:00	22:30:00	22:45:00	23:0:00	23
05-25	31.11	22.22	13.33	8.89	8.89	4.44	5.55	4.44	3.33	1.11	...	40.00	41.11	40.00	43.33	46.66	0.00	0.00	
05-27	33.33	24.44	15.55	10.00	8.89	4.44	2.22	3.33	2.22	6.67	...	36.67	36.67	37.78	45.55	50.00	24.44	1.11	

2 rows × 96 columns

In [ ]:

In [22]:
```
# DMA1_data_class2.T.iplot(title='$DMA1-属于类别2的天数的用水量趋势$')

fig = px.line(
    DMA1_data_class2.T,
    title='$DMA1-属于类别2的天数的用水量趋势$'
)
fig.update_layout(
    xaxis={"title": "$时间$"},
    yaxis={"title": "$用水量$"},
    legend_title_text='$日期$',
)
fig.write_image('./img/svg/DMA1-属于类别2的天数的用水量趋势.svg')
fig.show()
```



In [ ]:

DMA1-类3

In [ ]:

In [23]:
```
DMA1_class3_May = [21]
DMA1_class3_June = [6]
DMA1_class3 = [f'05-{i}' for i in DMA1_class3_May] + [f'06-{i}' for i in DMA1_class3_June]
DMA1_class3 = ['05-21', '06-06']

DMA1_data_class3_mask = DMA1_data.apply(lambda x: x.name in DMA1_class3, axis=1)
```

```
DMA1_data_class3 = DMA1_data[DMA1_data_class3_mask]
DMA1_data_class3
```

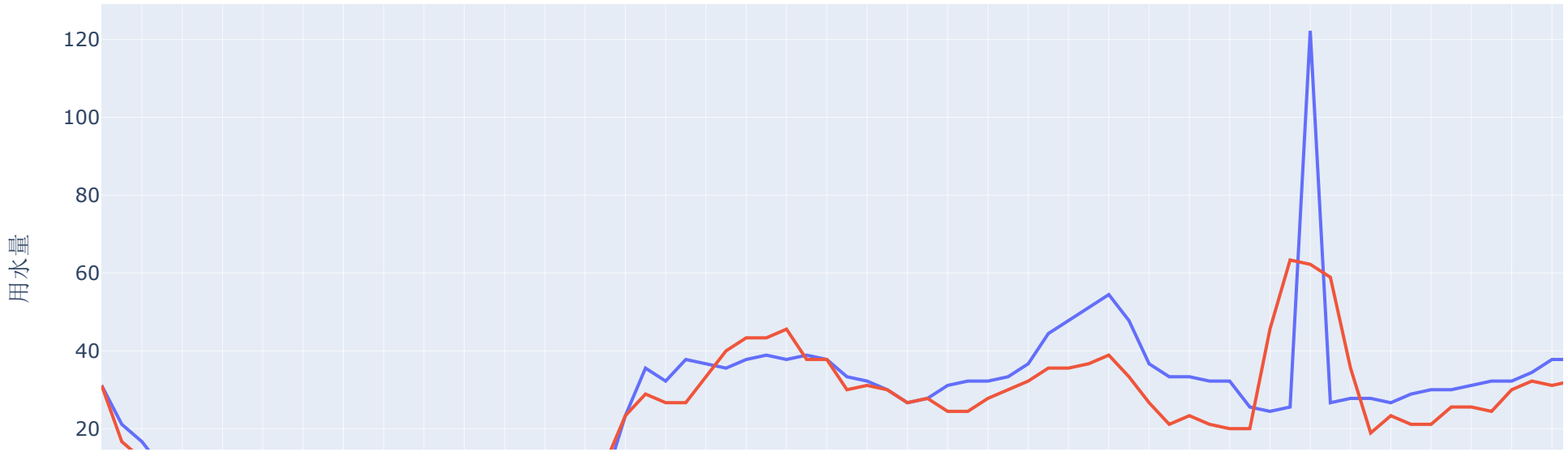
Out[23]:

	0:0:00	0:15:00	0:30:00	0:45:00	1:0:00	1:15:00	1:30:00	1:45:00	2:0:00	2:15:00	...	21:30:00	21:45:00	22:0:00	22:15:00	22:30:00	22:45:00	23:0:00	23
05-21	31.11	21.11	16.66	10.00	7.77	5.55	2.22	2.22	1.11	2.22	...	43.33	43.33	46.66	47.77	51.11	51.11	45.55	
06-06	31.11	16.67	12.22	11.11	6.67	6.67	3.33	2.22	3.33	3.33	...	34.44	36.67	36.67	41.11	46.67	48.89	45.56	

2 rows × 96 columns

In [ ]:

In [24]:
```
# DMA1_data_class3.T.iplot(title='$DMA1-属于类别3的天数的用水量趋势$')

fig = px.line(
    DMA1_data_class3.T,
    title='$DMA1-属于类别3的天数的用水量趋势$'
)
fig.update_layout(
    xaxis={"title": "$时间$"},
    yaxis={"title": "$用水量$"},
    legend_title_text='$日期$',
)
fig.write_image('./img/svg/DMA1-属于类别3的天数的用水量趋势.svg')
fig.show()
```



In [ ]:

DMA1-类4

In [ ]:

In [25]:
```
# DMA1_class4  # auto get
DMA1_class4_ = DMA1_class1 + DMA1_class2 + DMA1_class3 + ['05-28']

DMA1_data_class4_mask = DMA1_data.apply(lambda x: x.name not in DMA1_class4_ , axis=1)
DMA1_data_class4 = DMA1_data[DMA1_data_class4_mask]
DMA1_data_class4.head(10)
```

Out[25]:

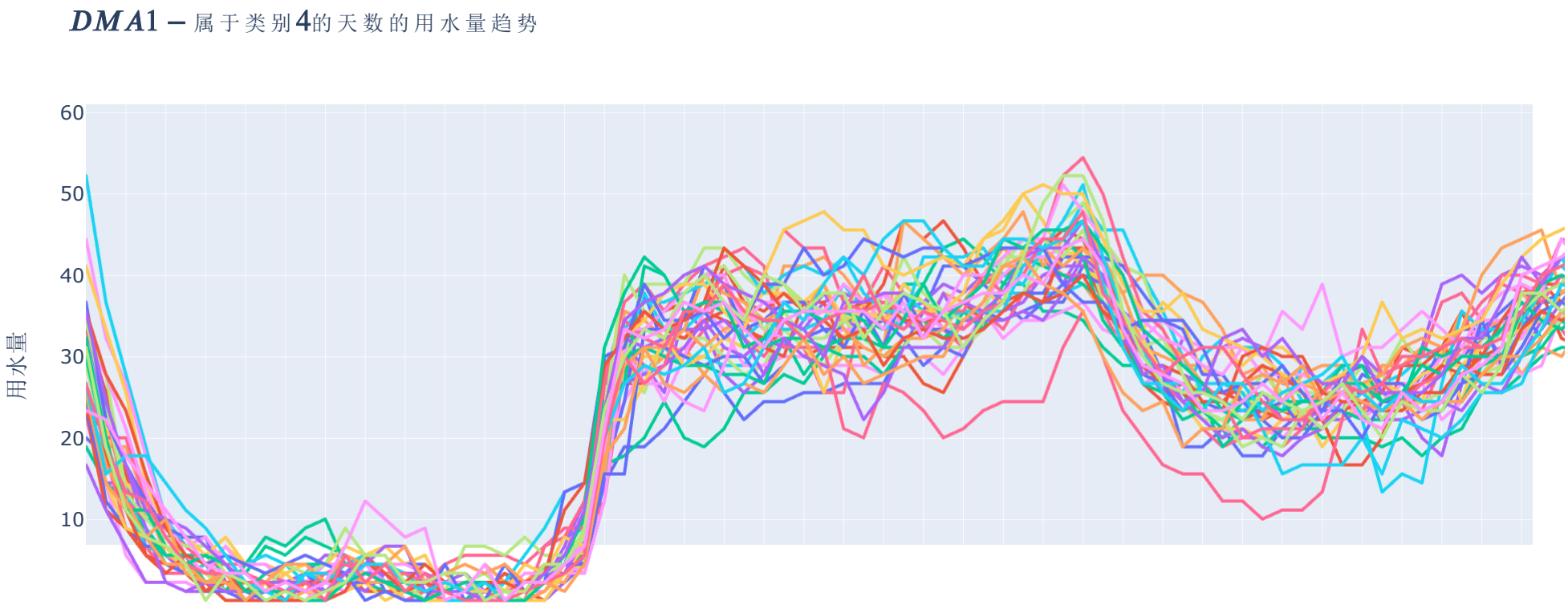
	0:0:00	0:15:00	0:30:00	0:45:00	1:0:00	1:15:00	1:30:00	1:45:00	2:0:00	2:15:00	...	21:30:00	21:45:00	22:0:00	22:15:00	22:30:00	22:45:00	23:0:00	23
04-18	20.00	17.78	11.11	10.00	7.78	7.78	7.78	2.22	2.22	0.00	...	38.89	40.00	41.11	40.00	37.78	38.89	43.33	
04-30	33.34	25.56	15.56	5.56	4.45	3.34	2.22	0.00	0.00	0.00	...	31.11	30.00	30.00	31.11	31.11	30.00	31.11	
05-01	18.89	14.45	12.22	6.67	5.56	5.56	5.56	3.33	3.33	6.67	...	36.67	34.45	33.33	33.33	37.78	37.78	36.67	
05-02	24.44	14.44	14.44	6.67	5.56	3.33	1.11	1.11	0.00	1.11	...	36.67	36.67	38.89	41.11	41.11	42.22	38.89	
05-03	25.56	20.00	16.67	10.00	7.78	5.56	3.33	1.11	2.22	1.11	...	41.11	40.00	38.89	42.22	41.11	42.22	36.67	
05-04	26.67	18.89	11.11	6.67	5.55	4.44	2.22	3.33	1.11	2.22	...	43.33	42.22	41.11	41.11	40.00	41.11	37.78	
05-05	22.22	17.78	8.89	6.66	3.33	3.33	1.11	3.33	3.33	2.22	...	33.33	33.33	31.11	34.44	37.78	35.55	35.55	
05-06	28.89	18.89	14.44	11.11	6.66	5.55	4.44	2.22	1.11	0.00	...	41.11	36.66	37.78	38.89	41.11	38.89	36.66	
05-07	22.22	14.45	5.56	2.22	2.22	2.22	1.11	2.22	2.22	1.11	...	33.33	36.67	35.56	35.56	38.89	37.78	36.67	
05-08	30.00	21.11	14.45	12.22	6.67	3.33	4.45	6.67	4.45	4.45	...	31.11	32.22	33.33	34.45	37.78	36.67	35.56	

10 rows × 96 columns

In [ ]:

In [26]:
```
# DMA1_data_class4.T.iplot(title='$DMA1-属于类别4的天数的用水量趋势$')

fig = px.line(
    DMA1_data_class4.T,
    title='$DMA1-属于类别4的天数的用水量趋势$'
)
fig.update_layout(
    xaxis={"title": "$时间$"},
    yaxis={"title": "$用水量$"},
    legend_title_text='$日期$',
)
```

```
fig.write_image('./img/svg/DMA1-属于类别4的天数的用水量趋势.svg')
fig.show()
```



In [ ]:

DMA2-类1

In [ ]:

In [27]:
```
DMA2_class1_April = [18, 22, 27, 17, 19, 23, 24, 28, 20, 21, 25, 26, 29, 15, 16]
DMA2_class1 = [f'04-{i}' for i in DMA2_class1_April]
```

```
DMA2_data_class1_mask = DMA2_data.apply(lambda x: x.name in DMA2_class1, axis=1)
DMA2_data_class1 = DMA2_data[DMA1_data_class1_mask]
DMA2_data_class1
```

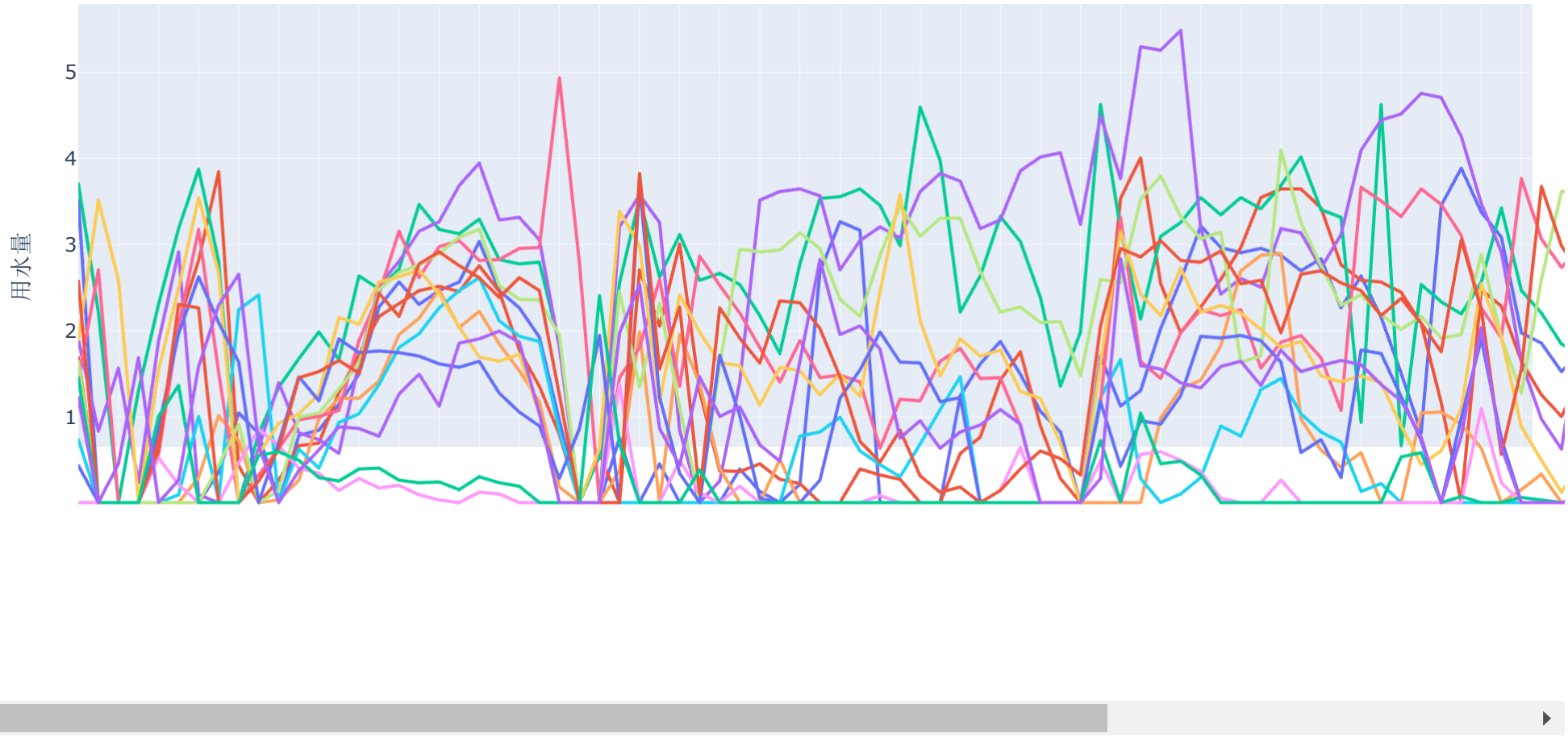
Out[27]:

	0:0:00	0:15:00	0:30:00	0:45:00	1:0:00	1:15:00	1:30:00	1:45:00	2:0:00	2:15:00	...	21:30:00	21:45:00	22:0:00	22:15:00	22:30:00	22:45:00	23:0:00	23
04-15	3.51	0.00	0.00	0.00	0.00	0.00	0.00	0.35	1.04	0.80	...	0.00	0.00	0.05	0.06	0.00	0.54	1.46	
04-16	1.68	0.00	0.00	0.00	0.73	2.08	2.81	3.84	0.43	0.00	...	1.04	1.39	1.04	0.51	0.70	1.02	0.00	
04-17	3.70	2.21	0.00	1.30	2.30	3.18	3.87	2.80	0.00	0.80	...	0.00	1.66	1.18	0.29	0.52	1.32	1.57	
04-19	1.89	0.83	1.56	0.23	1.87	2.91	0.09	0.00	0.00	0.63	...	2.22	2.47	1.54	0.61	0.63	0.74	1.63	
04-20	0.00	0.00	0.00	0.00	0.00	0.00	0.29	1.01	0.72	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
04-21	0.73	0.00	0.00	0.00	0.00	0.09	1.00	0.03	2.24	2.41	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
04-22	1.42	2.70	0.00	0.00	0.56	2.05	3.17	1.51	0.00	0.32	...	1.39	2.27	1.54	0.76	0.79	1.40	1.96	
04-23	1.64	0.00	0.00	0.00	0.00	0.00	0.00	0.43	0.91	0.00	...	0.93	1.51	1.78	1.25	1.25	2.65	3.64	
04-24	0.00	0.00	0.00	0.00	0.52	0.21	0.00	0.00	0.46	0.85	...	0.00	0.00	0.00	0.00	0.00	0.00	0.18	
04-25	1.89	3.51	2.58	0.00	1.54	2.49	3.54	2.66	0.00	0.56	...	0.00	0.00	0.00	0.00	0.00	0.44	1.25	
04-26	0.43	0.00	0.00	0.00	0.87	1.96	2.62	2.09	1.63	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
04-27	2.57	0.00	0.00	0.00	0.61	2.30	2.26	0.00	0.00	0.25	...	2.34	2.04	1.78	1.52	2.68	1.82	2.42	
04-28	1.45	0.00	0.00	0.00	1.00	1.36	0.00	0.00	0.00	0.55	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
04-29	1.22	0.00	0.45	1.68	0.00	0.26	1.58	2.29	2.65	0.66	...	0.00	0.77	0.23	0.00	0.00	0.00	0.89	

14 rows × 96 columns

In [ ]:

In [28]:
```
# DMA2_data_class1.T.iplot(title='$DMA2-属于类别1的天数的用水量趋势$')

fig = px.line(
    DMA2_data_class1.T,
    title='$DMA2-属于类别1的天数的用水量趋势$'
)
fig.update_layout(
    xaxis={"title": "$时间$"},
    yaxis={"title": "$用水量$"},
    legend_title_text='$日期$',
)
fig.write_image('./img/svg/DMA2-属于类别1的天数的用水量趋势.svg')
fig.show()
```



In [ ]:

DMA2-类2

In [ ]:

In [29]:
```
# DMA2_class2  # auto get
DMA2_class2_ = DMA2_class1 + ['05-28']

DMA2_data_class2_mask = DMA2_data.apply(lambda x: x.name not in DMA2_class2_ , axis=1)
```

```
DMA2_data_class2 = DMA2_data[DMA2_data_class2_mask]
DMA2_data_class2
```

Out[29]:

	0:0:00	0:15:00	0:30:00	0:45:00	1:0:00	1:15:00	1:30:00	1:45:00	2:0:00	2:15:00	...	21:30:00	21:45:00	22:0:00	22:15:00	22:30:00	22:45:00	23:0:00	23
04-30	10.87	7.20	2.40	0.59	0.35	0.33	0.21	0.22	0.30	0.25	...	10.59	10.63	10.82	10.11	10.50	10.80	11.37	
05-01	7.32	5.43	3.49	0.60	0.25	0.00	0.00	0.08	0.08	0.14	...	8.14	8.54	8.11	7.45	7.40	7.68	8.55	
05-02	10.57	4.34	4.48	0.55	0.28	0.14	0.00	0.01	0.04	0.00	...	7.88	7.74	9.31	8.04	8.84	9.38	9.97	
05-03	9.59	5.33	4.19	1.27	0.36	0.25	0.08	0.03	0.05	0.12	...	10.29	10.45	9.61	9.08	9.08	9.80	10.10	
05-04	11.62	7.11	1.94	0.46	0.45	0.13	0.00	0.00	0.01	0.00	...	10.41	10.38	10.87	11.16	11.38	12.35	12.07	
05-05	8.69	6.74	2.49	0.31	0.00	0.00	0.01	0.02	0.09	0.06	...	8.37	8.14	8.80	9.19	9.65	10.83	12.07	
05-06	12.55	4.74	1.06	0.52	0.19	0.14	0.11	0.23	0.05	0.16	...	9.57	9.93	9.50	8.98	9.17	9.96	10.09	
05-07	11.68	6.33	0.58	0.44	0.23	0.05	0.04	0.09	0.12	0.08	...	9.79	9.89	9.42	9.70	9.58	10.14	10.37	
05-08	11.15	5.75	1.33	0.84	0.47	0.11	0.06	0.19	0.19	0.21	...	10.84	10.90	9.77	9.44	10.04	11.13	11.33	
05-09	11.75	3.14	2.16	0.52	0.56	0.28	0.24	0.00	0.00	0.09	...	8.37	8.14	8.55	9.86	10.22	10.65	11.86	
05-10	12.03	3.93	1.10	0.54	0.37	0.20	0.20	0.17	0.38	0.00	...	9.87	9.43	9.18	8.58	8.13	8.03	8.33	
05-11	11.60	5.90	2.03	0.80	0.52	0.17	0.10	0.03	0.00	0.10	...	8.89	9.70	9.79	9.87	10.67	11.43	12.89	
05-12	10.64	4.46	0.92	0.34	0.06	0.11	0.00	0.01	0.02	0.04	...	7.33	10.04	9.93	9.00	9.46	10.20	11.07	
05-13	11.06	4.43	5.20	2.65	0.41	0.08	0.04	0.03	0.00	0.08	...	7.98	8.26	8.46	7.98	7.89	8.48	8.37	
05-14	10.75	3.52	3.40	1.60	0.53	0.39	0.41	0.46	0.45	0.46	...	8.43	8.45	7.90	7.03	7.14	7.81	8.91	
05-15	9.21	3.92	6.21	1.93	0.92	0.61	0.65	0.43	0.51	0.18	...	7.96	8.19	7.83	6.88	6.61	7.46	8.20	
05-16	10.92	8.75	2.03	1.30	0.85	0.47	0.23	0.23	0.35	0.33	...	8.55	8.23	6.84	5.89	5.68	5.73	7.95	

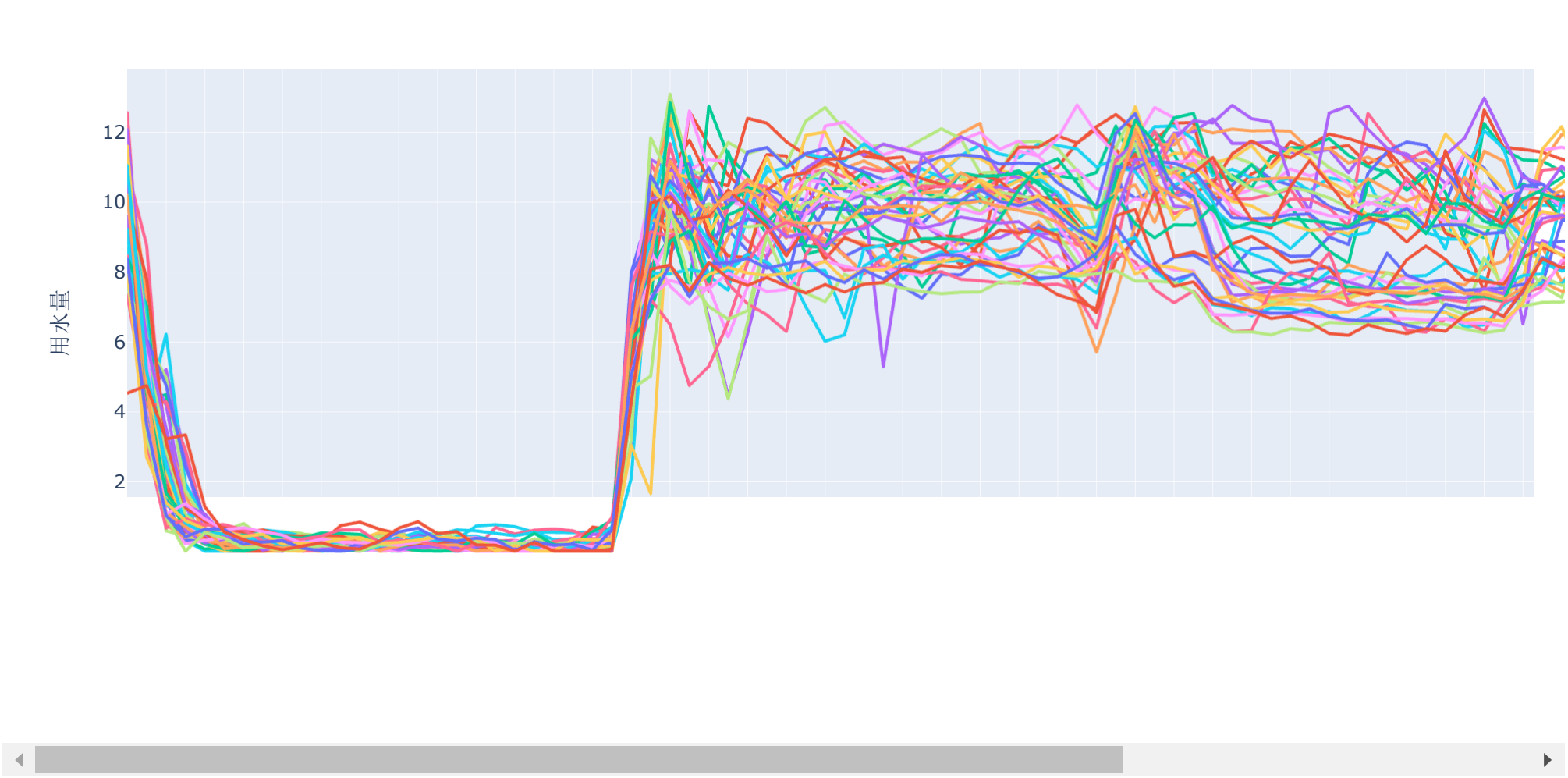
	0:0:00	0:15:00	0:30:00	0:45:00	1:0:00	1:15:00	1:30:00	1:45:00	2:0:00	2:15:00	...	21:30:00	21:45:00	22:0:00	22:15:00	22:30:00	22:45:00	23:0:00	23
05-17	7.60	5.97	4.97	1.72	0.87	0.57	0.79	0.39	0.56	0.51	...	10.59	11.11	11.09	10.24	10.53	9.32	9.48	
05-18	11.14	4.43	1.12	0.69	0.56	0.42	0.45	0.42	0.45	0.29	...	10.61	10.73	11.27	10.47	10.89	9.41	10.53	
05-19	11.03	4.95	2.31	0.57	0.20	0.00	0.04	0.05	0.15	0.18	...	8.68	8.90	8.34	7.66	8.08	8.52	9.39	
05-20	10.75	4.99	1.46	0.32	0.27	0.38	0.52	0.27	0.44	0.20	...	8.86	8.55	7.63	9.26	9.84	10.66	11.84	
05-21	11.07	3.06	1.79	0.57	0.37	0.10	0.00	0.00	0.04	0.00	...	9.68	9.89	9.60	8.87	8.81	9.58	10.70	
05-22	8.42	7.02	1.62	0.63	0.20	0.06	0.00	0.12	0.20	0.11	...	11.04	11.51	11.15	10.27	10.11	10.46	9.77	
05-23	12.06	5.92	2.89	0.60	0.24	0.34	0.46	0.46	0.48	0.00	...	9.63	10.11	9.93	9.04	9.18	9.68	10.32	
05-24	11.03	3.75	1.11	0.70	0.31	0.21	0.12	0.17	0.24	0.01	...	9.22	9.61	9.21	8.37	7.92	7.82	8.75	
05-25	10.86	6.59	1.33	1.03	0.99	0.55	0.51	0.61	0.55	0.44	...	7.47	7.22	7.05	6.61	7.04	0.00	0.00	
05-26	7.34	4.12	4.28	2.90	0.82	0.57	0.55	0.41	0.32	0.00	...	8.37	8.59	7.97	7.38	7.36	8.05	9.45	
05-27	10.93	6.17	2.89	1.04	0.65	0.38	0.30	0.34	0.34	0.49	...	4.11	3.77	3.14	3.55	3.80	0.00	0.00	
05-29	10.92	5.75	0.84	0.21	0.31	0.38	0.25	0.34	0.24	0.09	...	8.51	8.40	7.54	8.29	7.59	8.49	8.70	
05-30	11.59	5.10	1.62	0.97	0.63	0.61	0.33	0.06	0.01	0.00	...	9.02	8.62	7.87	8.36	7.80	8.10	8.73	
05-31	10.12	6.06	4.75	2.42	0.93	0.49	0.18	0.31	0.26	0.11	...	9.12	8.78	8.39	7.37	6.81	7.01	7.70	
06-01	10.02	7.78	3.11	1.25	0.86	0.67	0.55	0.61	0.45	0.43	...	9.83	9.79	9.63	9.75	9.34	9.18	9.00	
06-02	10.84	4.51	1.66	0.94	0.61	0.46	0.45	0.45	0.42	0.37	...	7.23	7.32	7.46	6.88	6.70	8.12	7.71	
06-03	10.59	6.00	3.52	1.31	1.07	0.41	0.39	0.03	0.15	0.11	...	8.68	8.78	8.65	7.72	7.96	9.14	10.34	

	0:0:00	0:15:00	0:30:00	0:45:00	1:0:00	1:15:00	1:30:00	1:45:00	2:0:00	2:15:00	...	21:30:00	21:45:00	22:0:00	22:15:00	22:30:00	22:45:00	23:0:00	23
06-04	10.20	4.70	0.75	0.38	0.67	0.07	0.12	0.13	0.10	0.22	...	6.17	6.14	5.70	4.89	4.81	5.25	6.88	
06-05	11.03	5.16	2.44	0.80	0.51	0.50	0.39	0.34	0.18	0.23	...	8.57	8.69	8.12	7.69	7.88	9.00	10.02	
06-06	8.45	3.12	0.66	0.93	0.74	0.76	0.61	0.29	0.31	0.34	...	7.66	7.73	7.45	7.69	8.77	7.75	8.30	
06-07	7.84	3.30	0.98	0.00	0.53	0.37	0.10	0.12	0.03	0.08	...	7.59	7.59	7.33	7.59	7.31	8.02	8.70	
06-08	8.10	3.65	1.06	1.36	0.92	0.61	0.67	0.56	0.46	0.22	...	7.09	7.42	7.36	7.11	7.12	7.12	8.36	
06-09	8.07	2.68	1.39	0.87	0.64	0.58	0.42	0.48	0.20	0.22	...	7.46	7.58	7.50	8.39	8.51	9.30	9.89	
06-10	8.37	3.62	1.02	0.41	0.63	0.62	0.29	0.26	0.30	0.09	...	7.66	7.77	8.13	7.00	8.07	8.55	3.77	
06-11	4.52	4.74	3.21	3.33	1.27	0.55	0.33	0.15	0.04	0.14	...	7.20	8.48	8.36	8.30	8.32	7.88	8.14	

42 rows × 96 columns

In [ ]:

In [30]:
```
# DMA2_data_class2.T.iplot(title='$DMA2-属于类别2的天数的用水量趋势$')

fig = px.line(
    DMA2_data_class2.T,
    title='$DMA2-属于类别2的天数的用水量趋势$'
)
fig.update_layout(
    xaxis={"title": "$时间$"},
    yaxis={"title": "$用水量$"},
    legend_title_text='$日期$',
)
fig.write_image('./img/svg/DMA2-属于类别2的天数的用水量趋势.svg')
fig.show()
```



In [ ]:

绘图 -- 问题3：类中的异常模式的趋势图

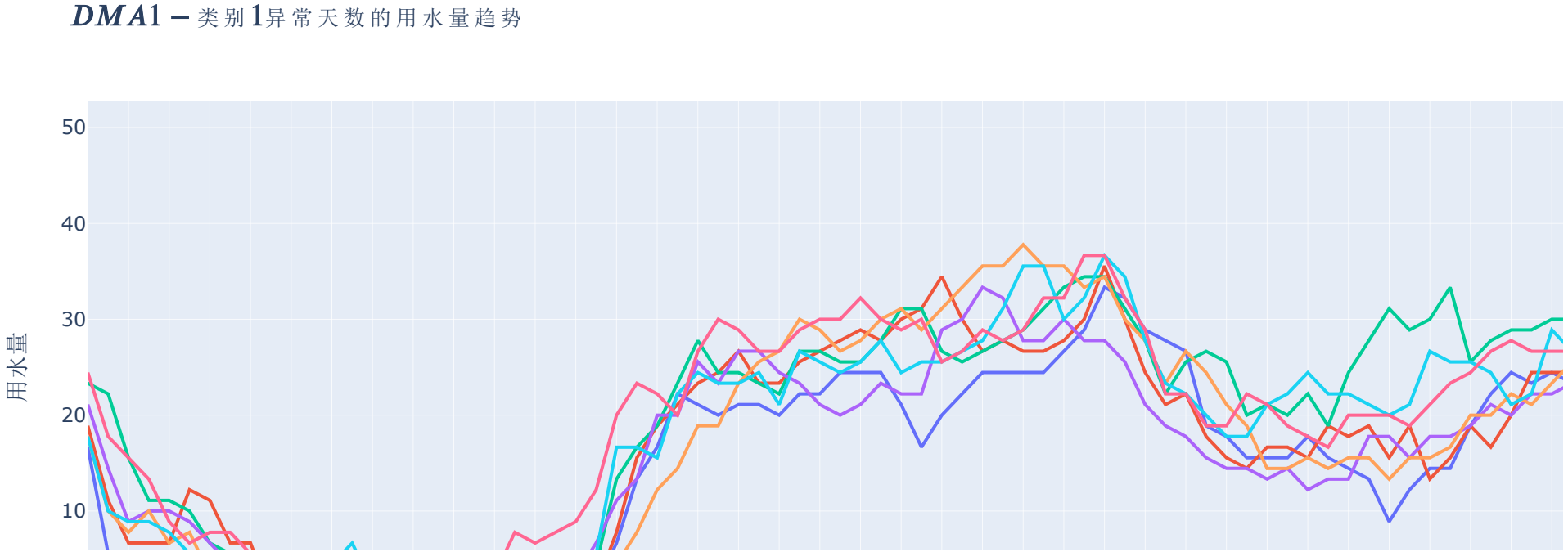
In [31]:
```
def plot_ab(abnormal, title):
    """
    :param abnormal: index 为日期, columns 为时间的数据
    :param title: 图的标题
    :return: fig go.Figure 对象
    """
    fig = px.line(
```

```
        abnormal.T,
        title='$' + title + '$',
    )
    fig.update_layout(
        xaxis={"title": "$时间$"},
        yaxis={"title": "$用水量$"},
        legend_title_text='$日期$',
    )
    fig.write_image('./img/svg/' + title + '.svg')
    return fig
```

DMA1-类1

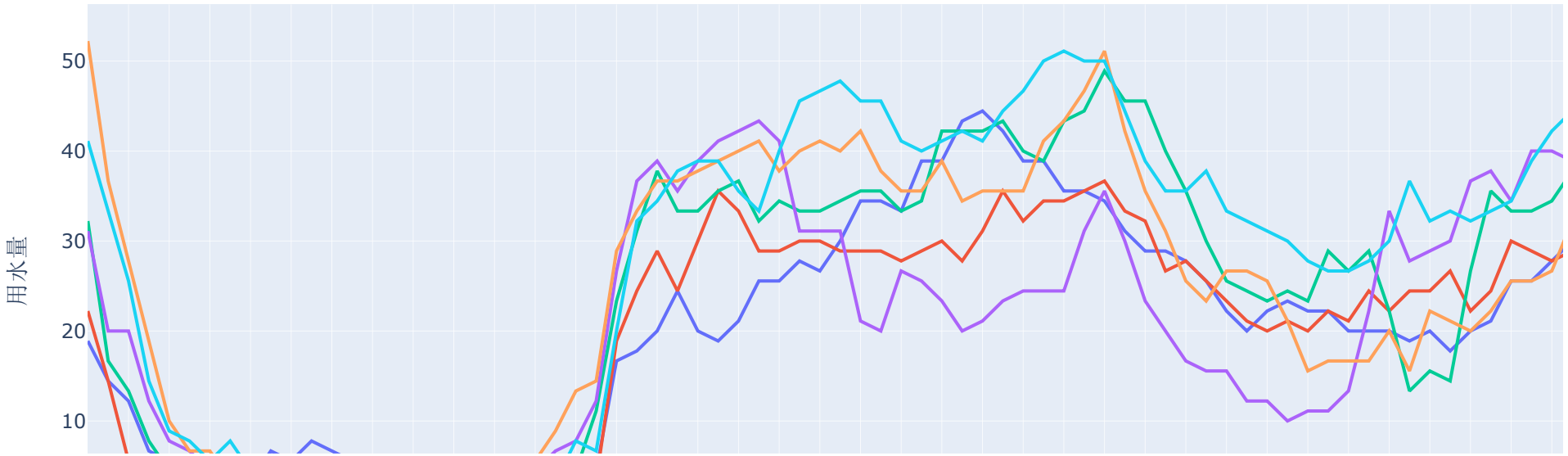
In [32]:

```
InteractiveShell.ast_node_interactivity = 'last'

DMA1_class1_normal = pd.read_excel('问题3数据.xlsx', sheet_name='DMA1-class1-normal', index_col=0)
DMA1_class1_abnormal = pd.read_excel('问题3数据.xlsx', sheet_name='DMA1-class1-abnormal', index_col=0)
DMA1_class1 = pd.concat([DMA1_class1_normal, DMA1_class1_abnormal])

fig = plot_ab(DMA1_class1_abnormal, 'DMA1-类别1异常天数的用水量趋势')
fig.show()
```



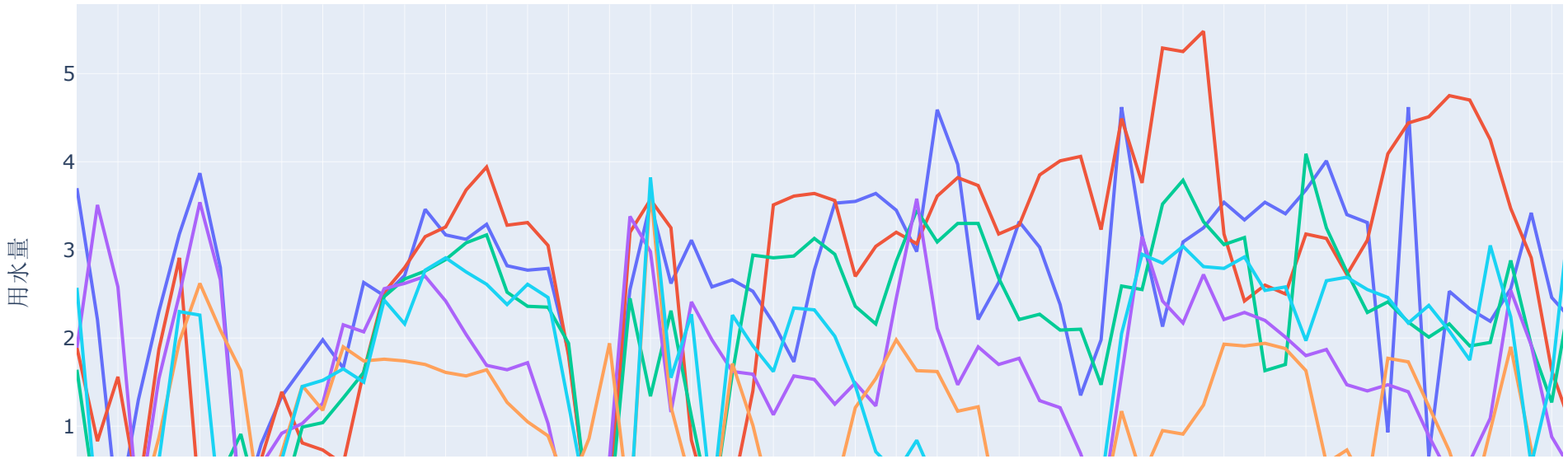
DMA1-类4

In [33]:
```
InteractiveShell.ast_node_interactivity = 'last'

DMA1_class4_normal = pd.read_excel('问题3数据.xlsx', sheet_name='DMA1-class4-normal', index_col=0)
DMA1_class4_abnormal = pd.read_excel('问题3数据.xlsx', sheet_name='DMA1-class4-abnormal', index_col=0)
DMA1_class4 = pd.concat([DMA1_class4_normal, DMA1_class4_abnormal])

fig = plot_ab(DMA1_class4_abnormal, 'DMA1-类别4异常天数的用水量趋势')
fig.show()
```



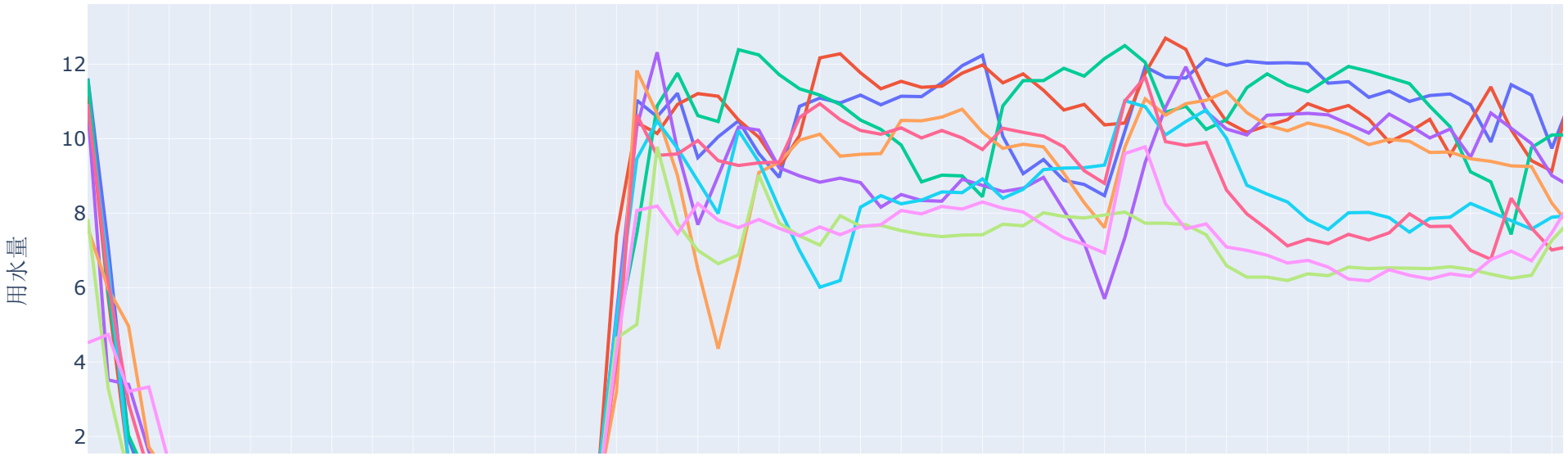
DMA2-类1

In [34]:
```
InteractiveShell.ast_node_interactivity = 'last'

DMA2_class1_normal = pd.read_excel('问题3数据.xlsx', sheet_name='DMA2-class1-normal', index_col=0)
DMA2_class1_abnormal = pd.read_excel('问题3数据.xlsx', sheet_name='DMA2-class1-abnormal', index_col=0)
DMA2_class1 = pd.concat([DMA2_class1_normal, DMA2_class1_abnormal])

fig = plot_ab(DMA2_class1_abnormal, 'DMA2-类别1异常天数的用水量趋势')
fig.show()
```



DMA2-类2

In [35]:
```
InteractiveShell.ast_node_interactivity = 'last'

DMA2_class2_normal = pd.read_excel('问题3数据.xlsx', sheet_name='DMA2-class2-normal', index_col=0)
DMA2_class2_abnormal = pd.read_excel('问题3数据.xlsx', sheet_name='DMA2-class2-abnormal', index_col=0)
DMA2_class2 = pd.concat([DMA2_class2_normal, DMA2_class2_abnormal])

fig = plot_ab(DMA2_class2_abnormal, 'DMA2-类别2异常天数的用水量趋势')
fig.show()
```



绘图 -- 问题3：对比正常、异常

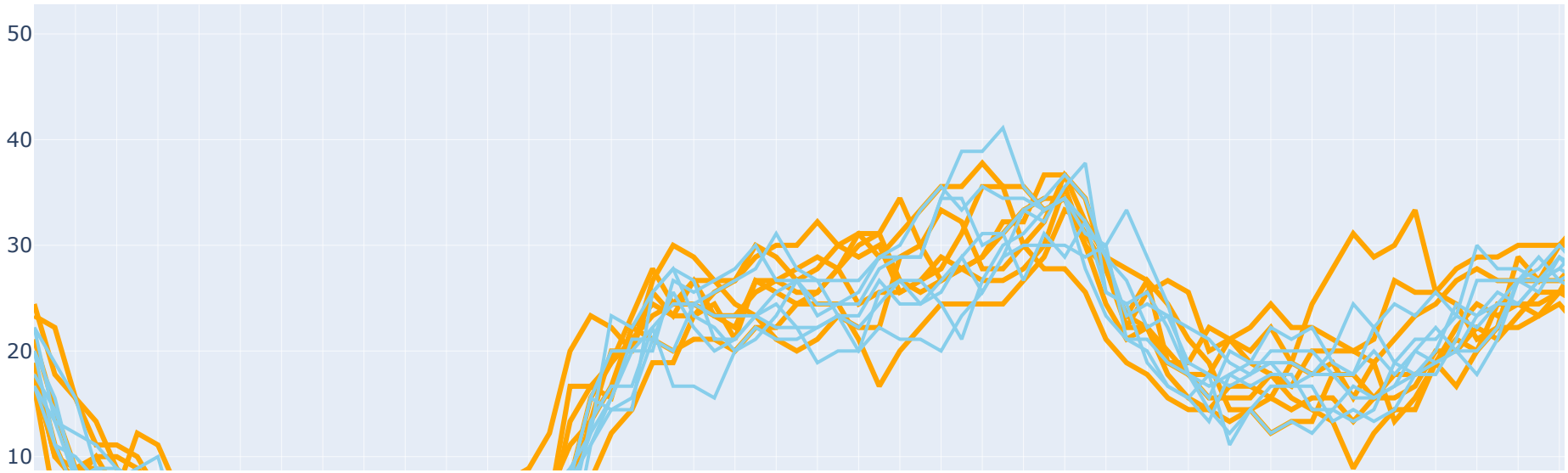
In [ ]:

In [36]:
```
def plot_ab_norm(normal, abnormal, title):
    """
    :param normal: 正常数据
    :param abnormal: 正常数据
    :param title: 标题
    :return: fig
    """
    traces = []
```

```
    # abnormal
    index = list(abnormal.index)
    columns = list(abnormal.columns)
    for i in range(abnormal.shape[0]):
        trace = go.Scatter(x=columns, y=abnormal.iloc[i, :], line=dict(color='orange', width=3), name=index[i] + " 异常", legendgroup="grou
        traces.append(trace)
    # normal
    index = list(normal.index)
    columns = list(normal.columns)
    for i in range(normal.shape[0]):
        trace = go.Scatter(x=columns, y=normal.iloc[i, :], line=dict(color='skyblue', width=2), name=index[i] + " 正常", legendgroup="group
        traces.append(trace)
    layout = go.Layout(legend=dict(traceorder='grouped+reversed'))
    fig = go.Figure(data=traces, layout=layout)
    fig.update_layout(
        title_text='$' + title + "$",
    )
    fig.write_image('./img/svg/' + title + '.svg')
    return fig
```

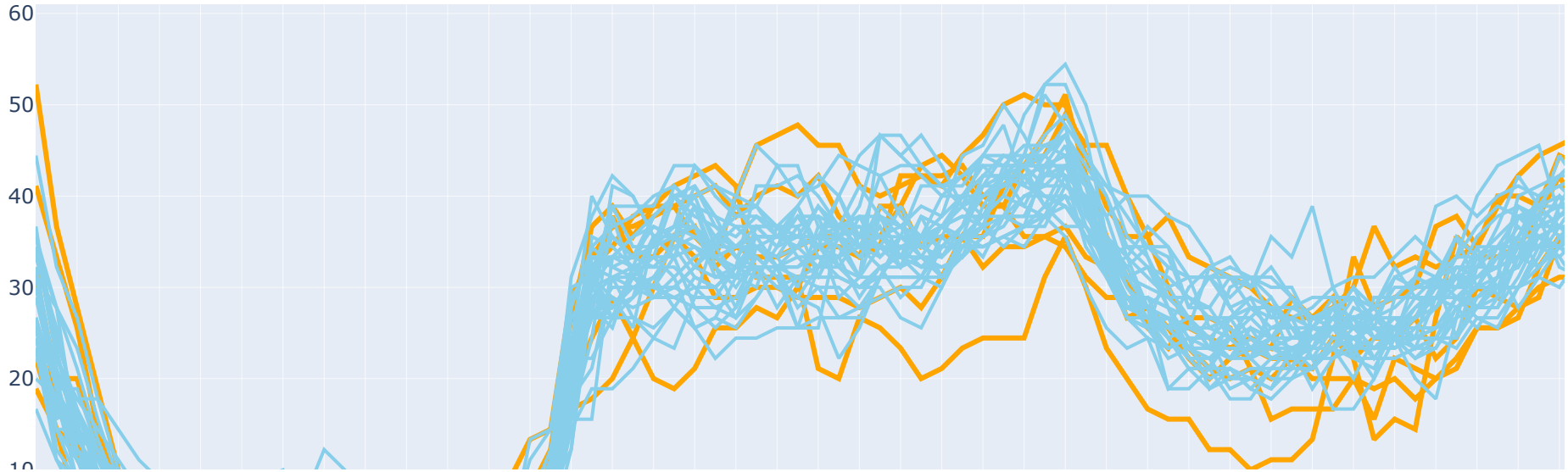
DMA1-类1

In [37]:
```
InteractiveShell.ast_node_interactivity = 'last'

DMA1_class1_normal = pd.read_excel('问题3数据.xlsx', sheet_name='DMA1-class1-normal', index_col=0)
DMA1_class1_abnormal = pd.read_excel('问题3数据.xlsx', sheet_name='DMA1-class1-abnormal', index_col=0)
DMA1_class1 = pd.concat([DMA1_class1_normal, DMA1_class1_abnormal])

fig = plot_ab_norm(DMA1_class1_normal, DMA1_class1_abnormal, 'DMA1-类别1正常和异常天数的用水量趋势的对比')
fig.show()
```



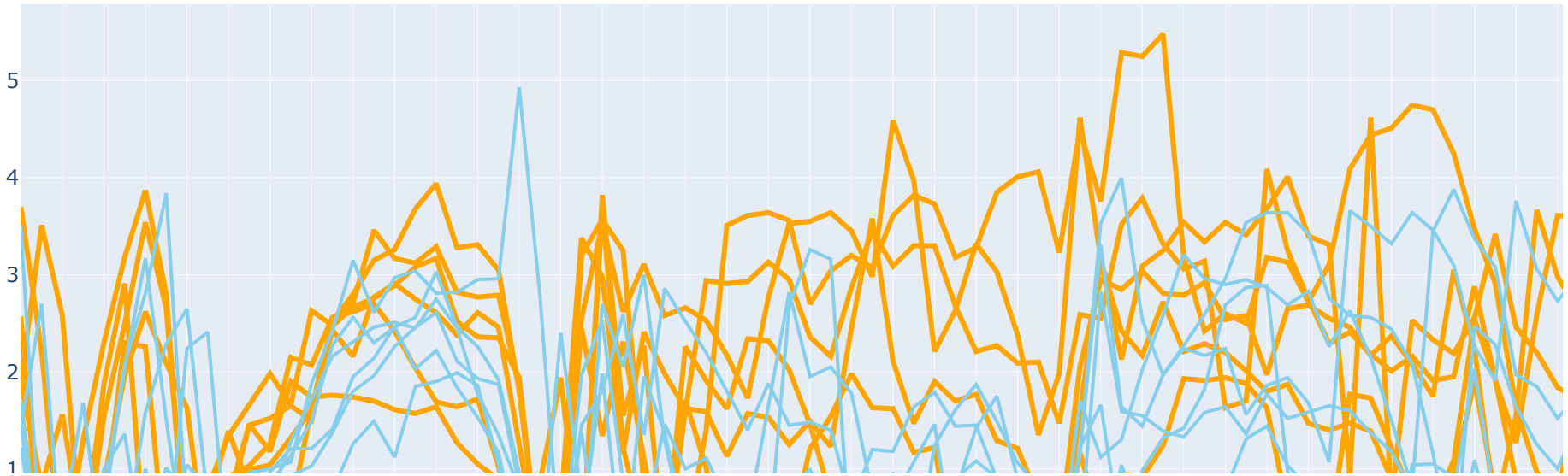
DMA1-类4

In [38]:
```
InteractiveShell.ast_node_interactivity = 'last'

DMA1_class4_normal = pd.read_excel('问题3数据.xlsx', sheet_name='DMA1-class4-normal', index_col=0)
DMA1_class4_abnormal = pd.read_excel('问题3数据.xlsx', sheet_name='DMA1-class4-abnormal', index_col=0)
DMA1_class4 = pd.concat([DMA1_class4_normal, DMA1_class4_abnormal])

fig = plot_ab_norm(DMA1_class4_normal, DMA1_class4_abnormal, 'DMA1-类别4正常和异常天数的用水量趋势的对比')
fig.show()
```



DMA2-类1

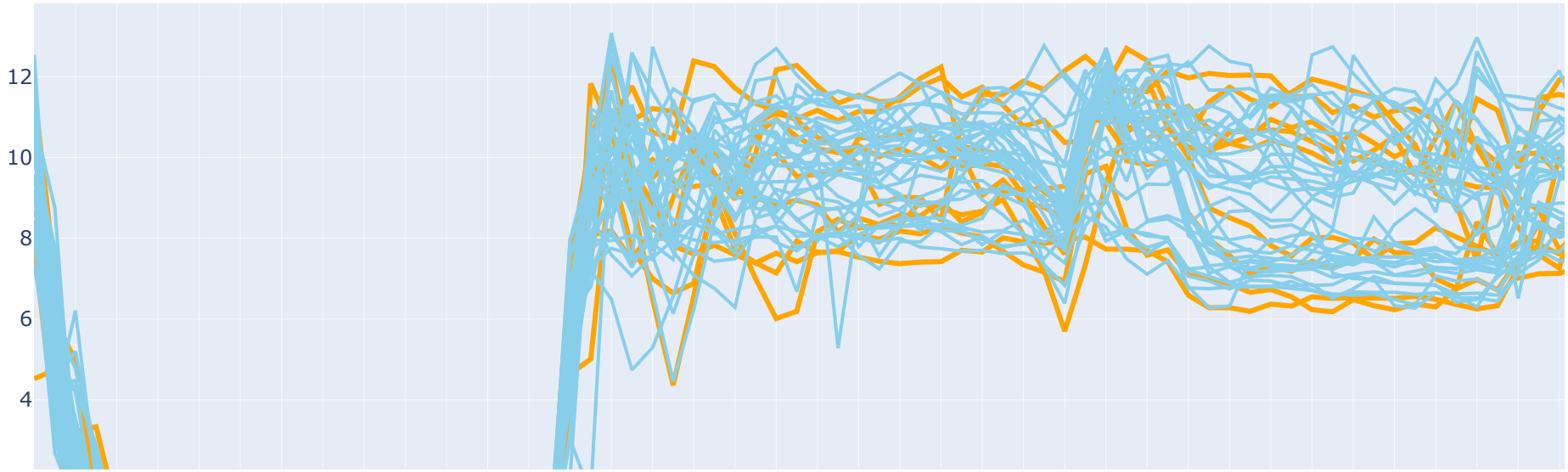
In [39]:
```
InteractiveShell.ast_node_interactivity = 'last'

DMA2_class1_normal = pd.read_excel('问题3数据.xlsx', sheet_name='DMA2-class1-normal', index_col=0)
DMA2_class1_abnormal = pd.read_excel('问题3数据.xlsx', sheet_name='DMA2-class1-abnormal', index_col=0)
DMA2_class1 = pd.concat([DMA2_class1_normal, DMA2_class1_abnormal])

fig = plot_ab_norm(DMA2_class1_normal, DMA2_class1_abnormal, 'DMA2-类别1正常和异常天数的用水量趋势的对比')
fig.show()
```



DMA2-类2

In [40]:
```
InteractiveShell.ast_node_interactivity = 'last'

DMA2_class2_normal = pd.read_excel('问题3数据.xlsx', sheet_name='DMA2-class2-normal', index_col=0)
DMA2_class2_abnormal = pd.read_excel('问题3数据.xlsx', sheet_name='DMA2-class2-abnormal', index_col=0)
DMA2_class2 = pd.concat([DMA2_class2_normal, DMA2_class2_abnormal])

fig = plot_ab_norm(DMA2_class2_normal, DMA2_class2_abnormal, 'DMA2-类别2正常和异常天数的用水量趋势的对比')
fig.show()
```

$DMA2-$类别2正常和异常天数的用水量趋势的对比



In [ ]:

绘图 -- 问题3：对比工作日、非工作日

In [ ]:

In [41]:
```
def plot_workday(data, title):
    """
    :param data: 数据
    :param title: 标题
    :return: fig
    """
```

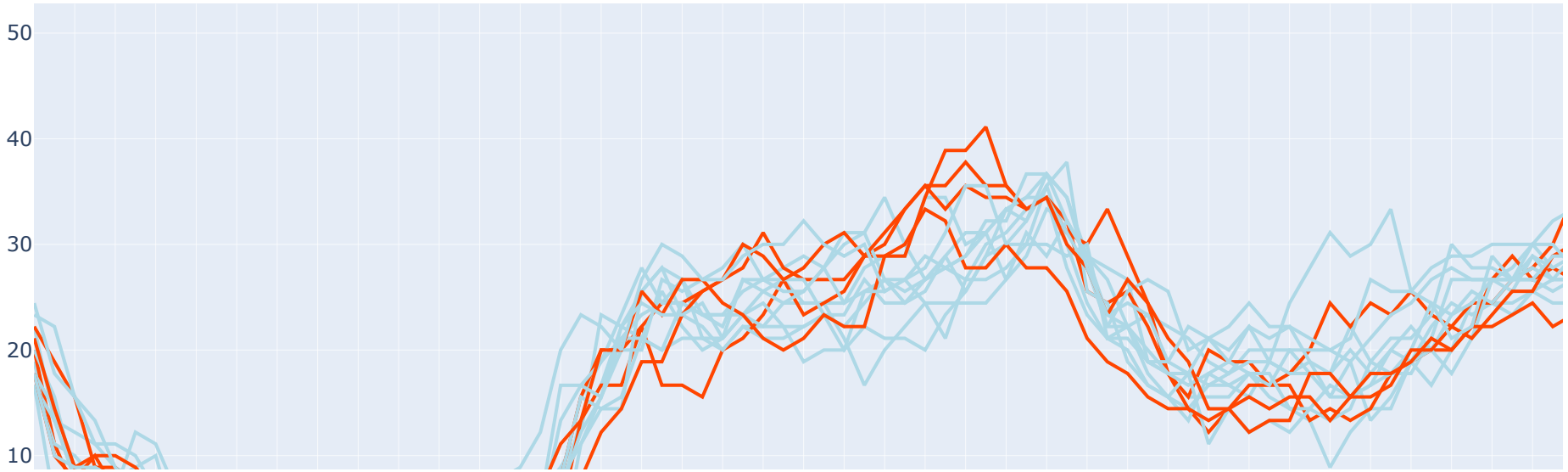
```
    week = pd.to_datetime('2014-' + data.index)
    week = week.map(lambda x:is_workday(x))

    traces = []
    index = list(data.index)
    columns = list(data.columns)
    for i, w in enumerate(week):
        if w:  # workday
            trace = go.Scatter(x=columns, y=data.iloc[i, :], line=dict(color='lightblue'), name=index[i] + "工作日")
        else:  # no-workday
            trace = go.Scatter(x=columns, y=data.iloc[i, :], line=dict(color='orangered'), name=index[i] + "非工作日")
        traces.append(trace)
#     layout = go.Layout(legend=dict(traceorder='grouped+reversed'))
    fig = go.Figure(data=traces)
    fig.update_layout(
        title_text='$' + title + "$",
    )
    fig.write_image('./img/svg/' + title + '.svg')
    return fig
```

DMA1-类1

In [ ]:

In [42]:
```
InteractiveShell.ast_node_interactivity = 'last'
DMA1_class1 = pd.read_excel('问题3数据.xlsx', sheet_name='DMA1-class1-all', index_col=0)
fig = plot_workday(DMA1_class1, "DMA1-类别1工作日和非工作日用水量趋势对比.svg")
fig.show()
```

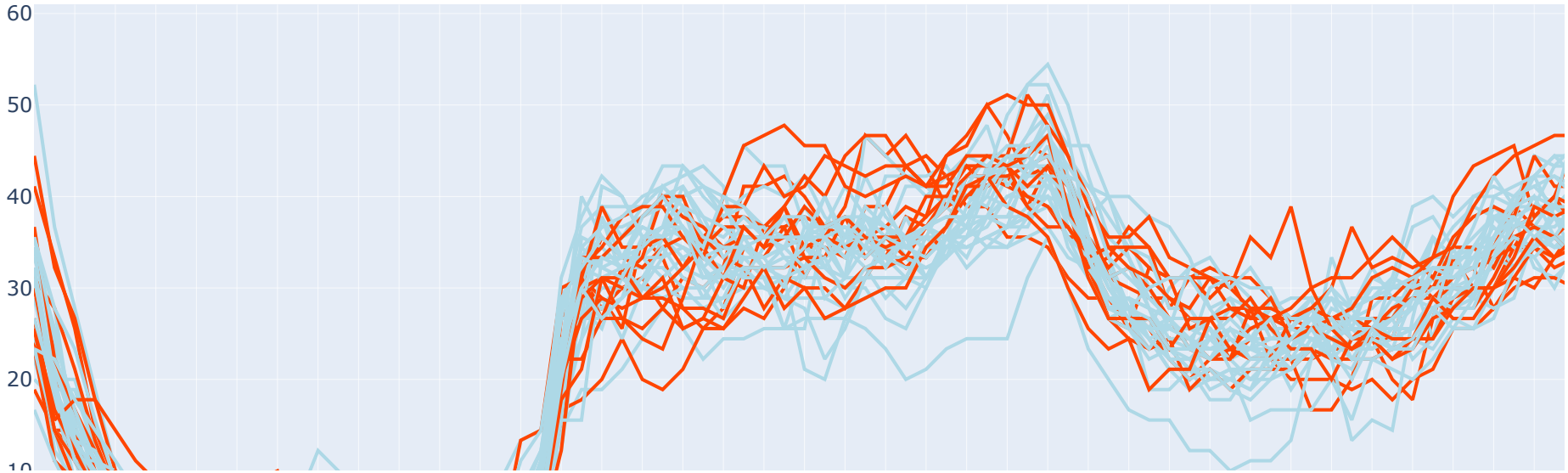
DMA1 – 类别1工作日和非工作日用水量趋势对比.svg



DMA1-类4

In [43]:
```
InteractiveShell.ast_node_interactivity = 'last'
DMA1_class4 = pd.read_excel('问题3数据.xlsx', sheet_name='DMA1-class4-all', index_col=0)
fig = plot_workday(DMA1_class4, "DMA1-类别4工作日和非工作日用水量趋势对比.svg")
fig.show()
```

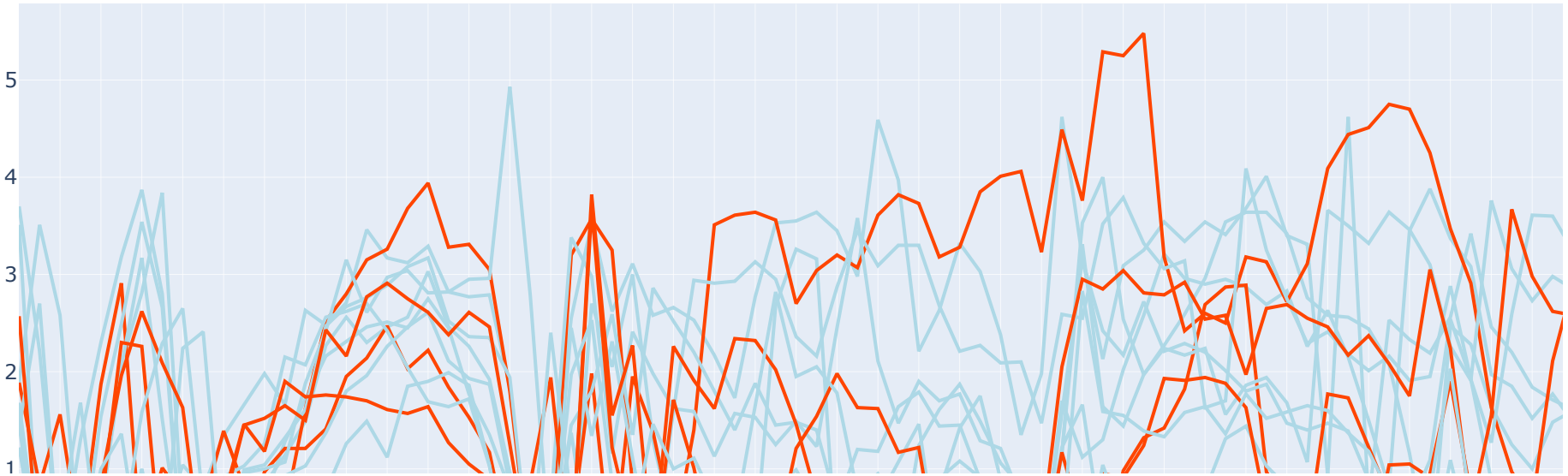
DMA1 – 类别4工作日和非工作日用水量趋势对比.svg



DMA2-类1

In [44]:
```
InteractiveShell.ast_node_interactivity = 'last'
DMA2_class1 = pd.read_excel('问题3数据.xlsx', sheet_name='DMA2-class1-all', index_col=0)
fig = plot_workday(DMA2_class1, "DMA2-类别1工作日和非工作日用水量趋势对比.svg")
fig.show()
```

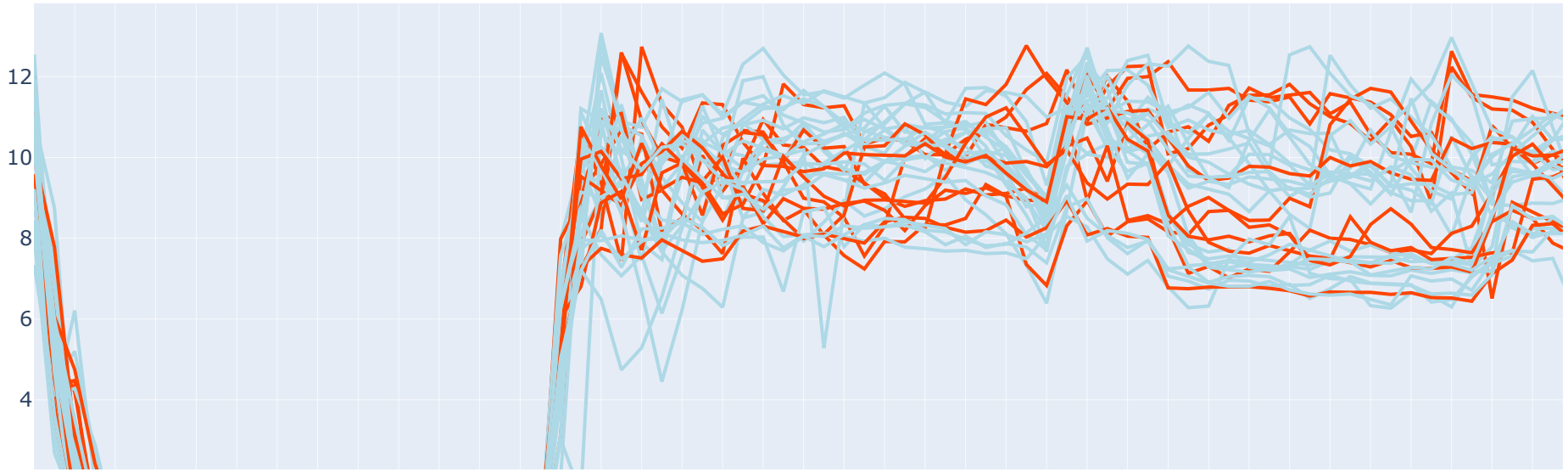
DMA2 – 类别1工作日和非工作日用水量趋势对比.svg



DMA2-类2

In [45]:
```
InteractiveShell.ast_node_interactivity = 'last'
DMA2_class2 = pd.read_excel('问题3数据.xlsx', sheet_name='DMA2-class2-normal', index_col=0)
fig = plot_workday(DMA2_class2, "DMA2-类别2工作日和非工作日用水量趋势对比.svg")
fig.show()
```

DMA2 – 类别2工作日和非工作日用水量趋势对比.svg



In [ ]:

绘图 -- 问题3-改进：对比正常、异常

In [ ]:

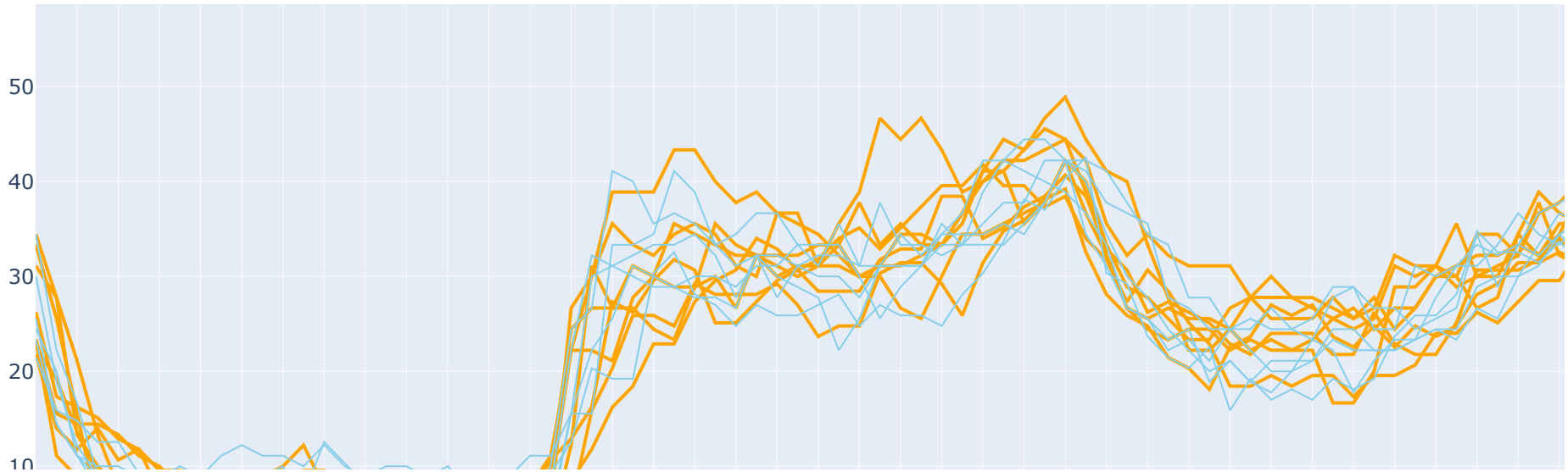
In [46]:
```
def plot_ab_norm(normal, abnormal, title):
    """
    :param normal: 正常数据
    :param abnormal: 正常数据
    :param title: 标题
    :return: fig
```

```
    """
    traces = []
    # abnormal
    index = list(abnormal.index)
    columns = list(abnormal.columns)
    for i in range(abnormal.shape[0]):
        trace = go.Scatter(x=columns, y=abnormal.iloc[i, :], line=dict(color='orange', width=2), name=index[i] + " 异常", legendgroup="grou
        traces.append(trace)
    # normal
    index = list(normal.index)
    columns = list(normal.columns)
    for i in range(normal.shape[0]):
        trace = go.Scatter(x=columns, y=normal.iloc[i, :], line=dict(color='skyblue', width=1), name=index[i] + " 正常", legendgroup="group
        traces.append(trace)
    layout = go.Layout(legend=dict(traceorder='grouped+reversed'))
    fig = go.Figure(data=traces, layout=layout)
    fig.update_layout(
        title_text='$' + title + "$",
    )
    fig.write_image('./img/svg/' + title + '.svg')
    return fig
```

DMA1-类1

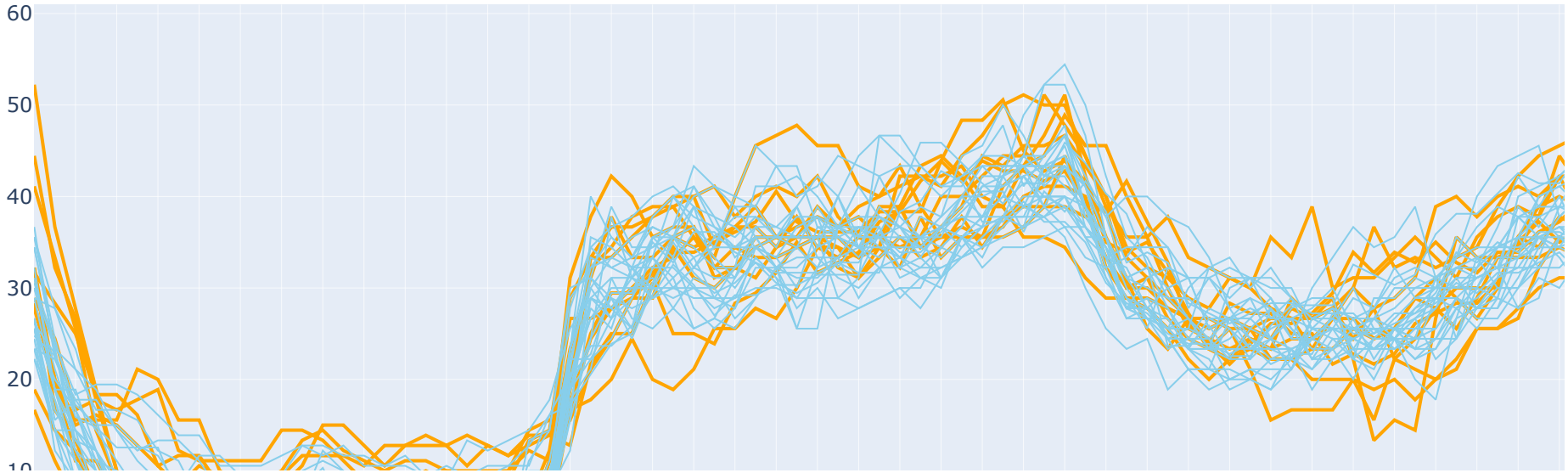
In [47]:

```
InteractiveShell.ast_node_interactivity = 'last'

DMA1_class1_normal = pd.read_excel('./问题3-模型改进-孤立森林数据.xlsx', sheet_name='DMA1-class1-normal', index_col=0)
DMA1_class1_abnormal = pd.read_excel('./问题3-模型改进-孤立森林数据.xlsx', sheet_name='DMA1-class1-abnormal', index_col=0)
DMA1_class1 = pd.concat([DMA1_class1_normal, DMA1_class1_abnormal])

fig = plot_ab_norm(DMA1_class1_normal, DMA1_class1_abnormal, 'DMA1-模型改进后类别1正常和异常天数的用水量趋势的对比')
fig.show()
```



DMA1-类4

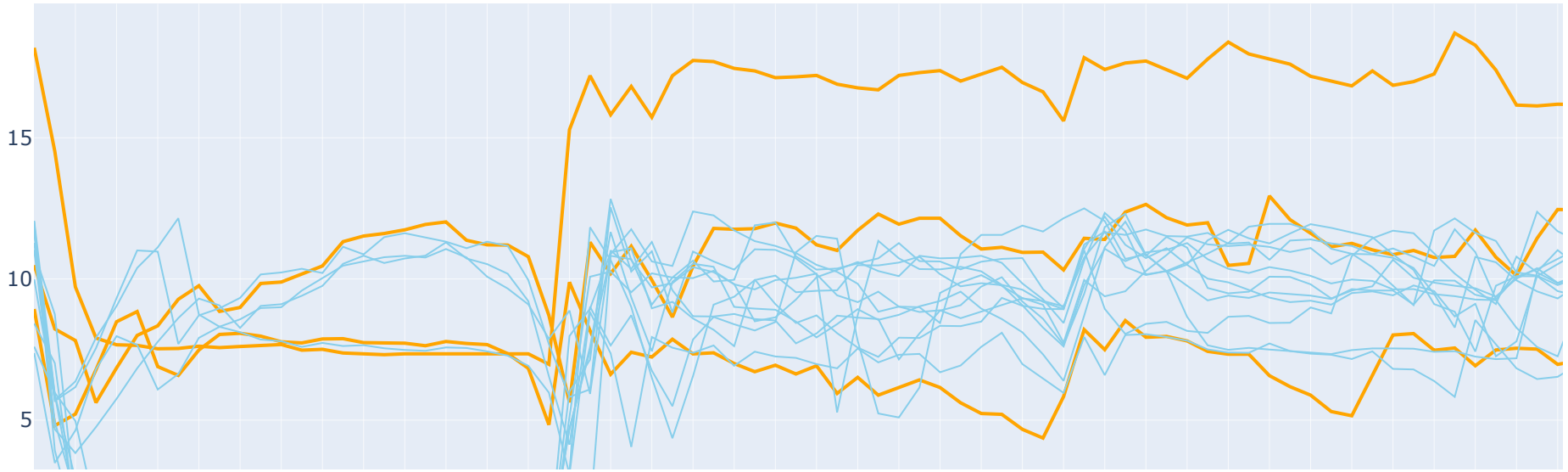
In [48]:
```
InteractiveShell.ast_node_interactivity = 'last'

DMA1_class2_normal = pd.read_excel('./问题3-模型改进-孤立森林数据.xlsx', sheet_name='DMA1-class2-normal', index_col=0)
DMA1_class2_abnormal = pd.read_excel('./问题3-模型改进-孤立森林数据.xlsx', sheet_name='DMA1-class2-abnormal', index_col=0)
DMA1_class2 = pd.concat([DMA1_class2_normal, DMA1_class2_abnormal])

fig = plot_ab_norm(DMA1_class2_normal, DMA1_class2_abnormal, 'DMA1-模型改进后类别2正常和异常天数的用水量趋势的对比')
fig.show()
```



DMA2-类1

In [49]:
```
InteractiveShell.ast_node_interactivity = 'last'

DMA2_class1_normal = pd.read_excel('./问题3-模型改进-孤立森林数据.xlsx', sheet_name='DMA2-class1-normal', index_col=0)
DMA2_class1_abnormal = pd.read_excel('./问题3-模型改进-孤立森林数据.xlsx', sheet_name='DMA2-class1-abnormal', index_col=0)
DMA2_class1 = pd.concat([DMA2_class1_normal, DMA2_class1_abnormal])

fig = plot_ab_norm(DMA2_class1_normal, DMA2_class1_abnormal, 'DMA2-模型改进后类别1正常和异常天数的用水量趋势的对比')
fig.show()
```

DMA2 — 模型改进后类别1正常和异常天数的用水量趋势的对比



In [ ]: