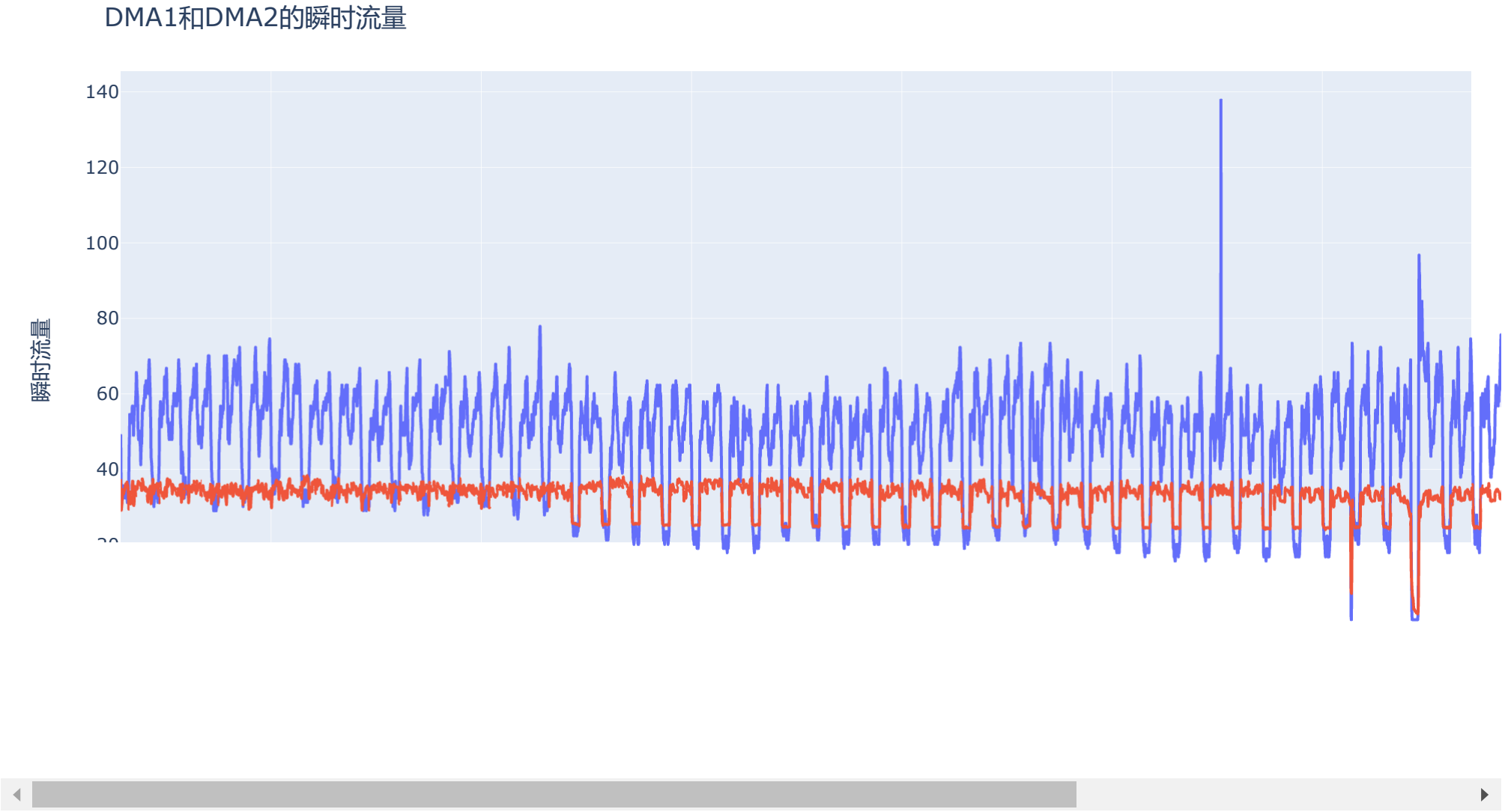
处理数据

In [1]:
```
import numpy as np
import pandas as pd
import cufflinks as cf
import plotly.express as px

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 warnings
warnings.filterwarnings('ignore')

import pylatex
import latexify
```

In [2]:
```
%reload_ext autoreload
%autoreload 2

# todo 设置配置信息（类似 plotly）
cf.set_config_file(
    offline=True,
    world_readable=True,  #
    theme='pearl',        # 设置绘图风格
    # offline=False,        # 离线
)

# todo 查看配置信息
cf.get_config_file()
```

Out[2]:
```
{'sharing': 'public',
 'theme': 'pearl',
 'colorscale': 'dflt',
 'offline': True,
 'offline_connected': True,
 'offline_url': '',
 'offline_show_link': True,
 'offline_link_text': 'Export to plot.ly',
 'datagen_mode': 'stocks',
 'dimensions': None,
 'margin': None,
 'offline_config': None}
```

In [3]:
```
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.show()
```



处理数据-问题1

DMA1、2 的总流量

In [4]:
```
import pandas as pd
writer = pd.ExcelWriter('按照日期处理后的数据.xlsx')
```

In [5]:
```
# 区域1总流量
InteractiveShell.ast_node_interactivity = 'all'
```

```
tmp = pd.DataFrame()
tmp_index = None
# for i in range(24):
for i in range(24):
    for j in range(0, 60, 15):
        tmp_data =  data_time.between_time(f"{i}:{j}:00", f"{i}:{j}:00").iloc[:, 0]
        if i == 0 and j == 0:
            tmp_index = list(tmp_data.index)
        try:
            tmp_data.index = list(tmp_index)
        except ValueError as e:
            tmp_data.index = list(tmp_index)[:-1]
        tmp[f"{i}:{j}:00"] = tmp_data
#         print()
#         fig = px.line(data_time.between_time(f"{i}:{j}:00", f"{i}:{j}:00"))
#         fig.show()
tmp[:-1].to_excel(writer, 'DMA1的瞬时流量')
tmp[:-1].head()
```

Out[5]:

	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
2014-04-15	48.89	37.78	34.44	33.33	33.33	32.22	32.22	33.33	32.22	34.44	...	61.11	61.11	60	64.44	67.78	68.89	66.67
2014-04-16	52.22	43.33	42.22	41.11	38.89	37.78	35.56	35.56	31.11	31.11	...	61.11	62.22	65.56	68.89	66.67	66.67	62.22
2014-04-17	51.11	43.33	38.89	38.89	38.89	44.44	43.33	38.89	38.89	33.33	...	67.78	67.78	70	68.89	70	66.67	66.67
2014-04-18	48.89	46.67	40	38.89	36.67	36.67	36.67	31.11	31.11	28.89	...	67.78	68.89	70	68.89	66.67	67.78	72.22
2014-04-19	53.33	50	46.67	40	40	37.78	32.22	30	32.22	31.11	...	57.78	60	62.22	67.78	72.22	73.33	74.44

5 rows × 96 columns

In [ ]:

In [6]:
```
# 区域2总流量
InteractiveShell.ast_node_interactivity = 'all'

tmp = pd.DataFrame()
tmp_index = None
# for i in range(24):
```

```
for i in range(24):
    for j in range(0, 60, 15):
        tmp_data =  data_time.between_time(f"{i}:{j}:00", f"{i}:{j}:00").iloc[:, 1]
        if i == 0 and j == 0:
            tmp_index = list(tmp_data.index)
        try:
            tmp_data.index = list(tmp_index)
        except ValueError as e:
            tmp_data.index = list(tmp_index)[:-1]
        tmp[f"{i}:{j}:00"] = tmp_data
#         print()
#         fig = px.line(data_time.between_time(f"{i}:{j}:00", f"{i}:{j}:00"))
#         fig.show()
tmp[:-1].to_excel(writer, 'DMA2的瞬时流量')
tmp[:-1].head()
```

Out[6]:

	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
2014-04-15	36.98	29.88	29.04	29.96	30.96	32.04	32.98	33.82	34.51	34.27	...	31.76	31.19	33.52	33.53	33.18	34.01	34.93
2014-04-16	35.06	30.78	31.44	32.99	34.11	35.46	36.19	37.22	33.81	33.38	...	34.42	34.77	34.42	33.89	34.08	34.4	32.98
2014-04-17	35.84	34.35	31.17	33.44	34.44	35.32	36.01	34.94	32.14	32.94	...	31.8	33.8	33.32	32.43	32.66	33.46	33.71
2014-04-18	35.67	34.63	30.79	32.36	32.82	34.08	35.5	32.21	31.77	32.38	...	34.46	34.48	33.93	33.02	33.29	33.59	34.07
2014-04-19	33.26	32.2	32.93	31.6	33.24	34.28	31.46	30.5	31.37	32	...	33.59	33.84	32.91	31.98	32	32.11	33

5 rows × 96 columns

DMA1、2 的漏水量

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

data_time_2_5 = data_time.between_time("2:00", "5:00")
data_time_2_5.T
data_time_2_5.index = data_time_2_5.index.strftime("%Y-%m-%d")
data_time_2_5.T

data_time_2_5_date = data_time_2_5.copy()
```

```
data_time_2_5_date["date"] = data_time_2_5_date.index
data_time_2_5_date.T
```

Out[7]:

当地时间(北京时间)	2014-04-15 02:00:00	2014-04-15 02:15:00	2014-04-15 02:30:00	2014-04-15 02:45:00	2014-04-15 03:00:00	2014-04-15 03:15:00	2014-04-15 03:30:00	2014-04-15 03:45:00	2014-04-15 04:00:00	2014-04-15 04:15:00	...	2014-06-12 02:45:00	2014-06-12 03:00:00	2014-06-12 03:15:00	2014-06-12 03:30:00	2014-06-12 03:45:00	2014-06-12 04:0
DMA1	32.22	34.44	35.56	34.44	34.44	33.33	35.56	33.33	32.22	33.33	...	20	20	21.11	21.11	18.89	
DMA2	34.51	34.27	33.47	34.25	34.31	34.61	34.96	35.74	36.03	35.77	...	24.55	24.47	24.44	24.48	24.37	2

2 rows × 767 columns

Out[7]:

当地时间(北京时间)	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	...	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12
DMA1	32.22	34.44	35.56	34.44	34.44	33.33	35.56	33.33	32.22	33.33	...	20	20	21.11	21.11	18.89	20	24.44	24.44	22.22	21.11
DMA2	34.51	34.27	33.47	34.25	34.31	34.61	34.96	35.74	36.03	35.77	...	24.55	24.47	24.44	24.48	24.37	24.27	24.57	24.67	24.63	24.41

2 rows × 767 columns

Out[7]:

当地时间(北京时间)	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	...	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12
DMA1	32.22	34.44	35.56	34.44	34.44	33.33	35.56	33.33	32.22	33.33	...	20	20	21.11	21.11	18.89	20	24.44	24.44	22.22	21.11
DMA2	34.51	34.27	33.47	34.25	34.31	34.61	34.96	35.74	36.03	35.77	...	24.55	24.47	24.44	24.48	24.37	24.27	24.57	24.67	24.63	24.41
date	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	...	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12

3 rows × 767 columns

In [ ]:

In [8]:
```
# 最小流量即漏水量
InteractiveShell.ast_node_interactivity = 'all'

min_flow1 = data_leaking1 = []
for date, data_date in data_time_2_5_date.groupby(by='date'):
    min_flow1.append(dict(data_date.min(0)))
```

```
min_flow1 = pd.DataFrame(min_flow1).iloc[:, 0::2]
min_flow_time1 = min_flow1.set_index("date")
min_flow_time1.index.name = '当地时间(北京时间)'
min_flow1.head()
min_flow_time1.head()

min_flow2 = data_leaking2 = []
for date, data_date in data_time_2_5_date.groupby(by='date'):
    min_flow2.append(dict(data_date.min(0)))
min_flow2 = pd.DataFrame(min_flow2).iloc[:, 1:]
min_flow_time2 = min_flow2.set_index("date")
min_flow_time2.index.name = '当地时间(北京时间)'
min_flow2.head()
min_flow_time2.head()

min_flow_time = pd.concat([min_flow_time1, min_flow_time2], axis=1)
min_flow_time.head()
min_flow_time.to_excel(writer, 'DMA1和DMA2的漏水量')
```

Out[8]:

	DMA1	date
0	32.22	2014-04-15
1	30.00	2014-04-16
2	32.22	2014-04-17
3	28.89	2014-04-18
4	31.11	2014-04-19

Out[8]:

	DMA1
当地时间(北京时间)	
2014-04-15	32.22
2014-04-16	30.00
2014-04-17	32.22
2014-04-18	28.89
2014-04-19	31.11

Out[8]:

	DMA2	date
0	33.47	2014-04-15
1	33.38	2014-04-16
2	32.14	2014-04-17
3	31.77	2014-04-18
4	31.37	2014-04-19

Out[8]:

	DMA2
当地时间(北京时间)	
2014-04-15	33.47
2014-04-16	33.38
2014-04-17	32.14
2014-04-18	31.77
2014-04-19	31.37

Out[8]:

	DMA1	DMA2
当地时间(北京时间)		
2014-04-15	32.22	33.47
2014-04-16	30.00	33.38
2014-04-17	32.22	32.14
2014-04-18	28.89	31.77
2014-04-19	31.11	31.37

DMA1、2 的用户用水量

In [9]:

```
data_time1 = pd.DataFrame(data_time.iloc[:, 0])
data_time2 = pd.DataFrame(data_time.iloc[:, 1])
# data_time1
# data_time2
pd.concat([data_time1, data_time2], axis=1).T
```

Out[9]:

当地时间(北京时间)	2014-04-15 00:00:00	2014-04-15 00:15:00	2014-04-15 00:30:00	2014-04-15 00:45:00	2014-04-15 01:00:00	2014-04-15 01:15:00	2014-04-15 01:30:00	2014-04-15 01:45:00	2014-04-15 02:00:00	2014-04-15 02:15:00	...	2014-06-12 19:45:00	2014-06-12 20:00:00	2014-06-12 20:15:00	2014-06-12 20:30:00	2014-06-12 20:45:00	2014-06-12 21:0
DMA1	48.89	37.78	34.44	33.33	33.33	32.22	32.22	33.33	32.22	34.44	...	56.67	60	65.56	63.33	62.22	6
DMA2	36.98	29.88	29.04	29.96	30.96	32.04	32.98	33.82	34.51	34.27	...	31.56	32.27	34.31	33.46	32.85	3

2 rows × 5657 columns

In [ ]:

In [10]:
```
data_time_YHD = data_time.copy()
data_time_YHD.index = data_time_YHD.index.strftime("%Y-%m-%d")
data_time_YHD.T
```

Out[10]:

当地时间(北京时间)	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	...	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12
DMA1	48.89	37.78	34.44	33.33	33.33	32.22	32.22	33.33	32.22	34.44	...	56.67	60	65.56	63.33	62.22	61.11	65.56	62.22	60	65.56
DMA2	36.98	29.88	29.04	29.96	30.96	32.04	32.98	33.82	34.51	34.27	...	31.56	32.27	34.31	33.46	32.85	32.19	31.89	32	31.88	31.27

2 rows × 5657 columns

In [ ]:

In [11]:
```
data_time_YHD1 = pd.DataFrame(data_time_YHD.iloc[:, 0])
data_time_YHD2 = pd.DataFrame(data_time_YHD.iloc[:, 1])
# data_time_YHD1
# data_time_YHD2
pd.concat([data_time_YHD1, data_time_YHD2], axis=1).T
```

Out[11]:

当地时间(北京时间)	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	...	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12
DMA1	48.89	37.78	34.44	33.33	33.33	32.22	32.22	33.33	32.22	34.44	...	56.67	60	65.56	63.33	62.22	61.11	65.56	62.22	60	65.56
DMA2	36.98	29.88	29.04	29.96	30.96	32.04	32.98	33.82	34.51	34.27	...	31.56	32.27	34.31	33.46	32.85	32.19	31.89	32	31.88	31.27

2 rows × 5657 columns

In [ ]:

In [12]:
```
data_use1 = data_time_YHD1 - min_flow_time1
data_use2 = data_time_YHD2 - min_flow_time2
# data_use1
# data_use2
pd.concat([data_use1, data_use2], axis=1).T
```

Out[12]:

当地时间(北京时间)	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	2014-04-15	...	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12	2014-06-12
DMA1	16.67	5.56	2.22	1.11	1.11	0	0	1.11	0	2.22	...	37.78	41.11	46.67	44.44	43.33	42.22	46.67	43.33	41.11	46.67
DMA2	3.51	-3.59	-4.43	-3.51	-2.51	-1.43	-0.49	0.35	1.04	0.8	...	7.29	8	10.04	9.19	8.58	7.92	7.62	7.73	7.61	7

2 rows × 5657 columns

In [ ]:

In [13]:
```
# 处理 < 0数据
data_use1.index = data_time1.index
data_use2.index = data_time2.index
data_use1[data_use1 < 0] = 0
data_use2[data_use2 < 0] = 0
# data_use1
# data_use2
pd.concat([data_use1, data_use2], axis=1).T
```

Out[13]:

当地时间(北京时间)	2014-04-15 00:00:00	2014-04-15 00:15:00	2014-04-15 00:30:00	2014-04-15 00:45:00	2014-04-15 01:00:00	2014-04-15 01:15:00	2014-04-15 01:30:00	2014-04-15 01:45:00	2014-04-15 02:00:00	2014-04-15 02:15:00	...	2014-06-12 19:45:00	2014-06-12 20:00:00	2014-06-12 20:15:00	2014-06-12 20:30:00	2014-06-12 20:45:00	2(0(21:0
DMA1	16.67	5.56	2.22	1.11	1.11	0	0	1.11	0	2.22	...	37.78	41.11	46.67	44.44	43.33	4
DMA2	3.51	0	0	0	0	0	0	0.35	1.04	0.8	...	7.29	8	10.04	9.19	8.58	

2 rows × 5657 columns

In [ ]:

In [14]:
```
data_use = pd.concat([data_use1, data_use2], axis=1)
data_use.to_excel('用户用水量数据.xlsx')
```

In [ ]:

In [15]:
```
# 区域1用户用水量
tmp = pd.DataFrame()
tmp_index = None
# for i in range(24):
for i in range(24):
    for j in range(0, 60, 15):
        tmp_data =  data_use1.between_time(f"{i}:{j}:00", f"{i}:{j}:00").iloc[:, 0]
        if i == 0 and j == 0:
            tmp_index = list(tmp_data.index)
        try:
            tmp_data.index = list(tmp_index)
        except ValueError as e:
            tmp_data.index = list(tmp_index)[:-1]
        tmp[f"{i}:{j}:00"] = tmp_data
#         print()
#         fig = px.line(data_time.between_time(f"{i}:{j}:00", f"{i}:{j}:00"))
#         fig.show()
tmp[:-1].to_excel(writer, 'DMA1的用户用水量')
tmp[:-1].head()
```

Out[15]:

	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
2014-04-15	16.67	5.56	2.22	1.11	1.11	0	0	1.11	0	2.22	...	28.89	28.89	27.78	32.22	35.56	36.67	34.45
2014-04-16	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
2014-04-17	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
2014-04-18	20	17.78	11.11	10	7.78	7.78	7.78	2.22	2.22	0	...	38.89	40	41.11	40	37.78	38.89	43.33
2014-04-19	22.22	18.89	15.56	8.89	8.89	6.67	1.11	0	1.11	0	...	26.67	28.89	31.11	36.67	41.11	42.22	43.33

5 rows × 96 columns

In [ ]:

In [16]:
```
# 区域2用户用水量
tmp = pd.DataFrame()
tmp_index = None
# for i in range(24):
for i in range(24):
    for j in range(0, 60, 15):
        tmp_data =  data_use2.between_time(f"{i}:{j}:00", f"{i}:{j}:00").iloc[:, 0]
        if i == 0 and j == 0:
            tmp_index = list(tmp_data.index)
        try:
            tmp_data.index = list(tmp_index)
        except ValueError as e:
            tmp_data.index = list(tmp_index)[:-1]
        tmp[f"{i}:{j}:00"] = tmp_data
#         print()
#         fig = px.line(data_time.between_time(f"{i}:{j}:00", f"{i}:{j}:00"))
#         fig.show()
tmp[:-1].to_excel(writer, 'DMA2的用户用水量')
tmp[:-1].head()
```

Out[16]:

	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
2014-04-15	3.51	0	0	0	0	0	0	0.35	1.04	0.8	...	0	0	0.05	0.06	0	0.54	1.46
2014-04-16	1.68	0	0	0	0.73	2.08	2.81	3.84	0.43	0	...	1.04	1.39	1.04	0.51	0.7	1.02	0
2014-04-17	3.7	2.21	0	1.3	2.3	3.18	3.87	2.8	0	0.8	...	0	1.66	1.18	0.29	0.52	1.32	1.57
2014-04-18	3.9	2.86	0	0.59	1.05	2.31	3.73	0.44	0	0.61	...	2.69	2.71	2.16	1.25	1.52	1.82	2.3
2014-04-19	1.89	0.83	1.56	0.23	1.87	2.91	0.09	0	0	0.63	...	2.22	2.47	1.54	0.61	0.63	0.74	1.63

5 rows × 96 columns

In [ ]:

In [17]:
```
writer.save()
writer.close()
```

DMA1、2瞬时流量 -- 星期划分

In [18]:
```
writer_week = pd.ExcelWriter('按照星期处理后的数据.xlsx')
```

In [19]:
```
df_time1 = pd.read_excel('按照日期处理后的数据.xlsx', sheet_name='DMA1的瞬时流量', index_col=0)
df_time1['week'] = df_time1.index.dayofweek + 1
df_time1.head().iloc[:, -2:]

data_user_week1 = []
for week, data in df_time1.groupby(by='week'):
    data_user_week1.append(data.mean(0))
data_user_week1 = pd.DataFrame(data_user_week1)
data_user_week1.index.name = '星期'
data_user_week1.index = ["星期一", "星期二", "星期三", "星期四", "星期五", "星期六", "星期日"]
data_user_week1 = data_user_week1.iloc[:, :-1]
data_user_week1.to_excel(writer_week, 'DMA1的瞬时流量')
data_user_week1
```

Out[19]:

	23:45:00	week
2014-04-15	55.56	2
2014-04-16	52.22	3
2014-04-17	57.78	4
2014-04-18	61.11	5
2014-04-19	57.78	6

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	2
星期一	48.195000	39.722500	34.307500	30.971250	27.917500	26.110000	24.861250	24.166250	24.166250	23.195000	...	59.305000	60.138750	61.390000	64.860000	67
星期二	49.505556	40.618889	35.554444	31.727778	28.642222	27.281111	25.925556	25.432222	24.073333	24.197778	...	58.271111	58.024444	59.506667	63.580000	65
星期三	43.210000	35.803333	30.617778	26.790000	24.814444	23.580000	22.468889	22.222222	20.864444	20.864444	...	59.874444	60.245556	61.234444	63.703333	65
星期四	48.471250	40.416250	35.138750	31.667500	29.585000	28.887500	27.222500	26.387500	25.693750	25.416250	...	58.750000	59.723750	61.665000	64.583750	67
星期五	49.860000	40.833750	34.442500	30.556250	28.197500	26.806250	25.277500	23.471250	22.500000	22.777500	...	57.638750	59.027500	59.027500	62.777500	64
星期六	49.443750	41.528750	37.223750	32.222500	30.000000	27.638750	25.555000	25.138750	24.166250	23.610000	...	56.805000	57.501250	58.333750	61.248750	64
星期日	48.888750	39.860000	34.723750	32.085000	29.582500	27.638750	26.250000	25.555000	24.443750	24.027500	...	60.138750	60.972500	61.945000	64.305000	64

7 rows × 96 columns

In [ ]:

In [20]:
```
df_time2 = pd.read_excel('按照日期处理后的数据.xlsx', sheet_name='DMA2的瞬时流量', index_col=0)
df_time2['week'] = df_time2.index.dayofweek + 1
```

```
df_time2.head().iloc[:, -2:]

data_user_week2 = []
for week, data in df_time2.groupby(by='week'):
    data_user_week2.append(data.mean(0))
data_user_week2 = pd.DataFrame(data_user_week2)
data_user_week2.index.name = '星期'
data_user_week2.index = ["星期一", "星期二", "星期三", "星期四", "星期五", "星期六", "星期日"]
data_user_week2 = data_user_week2.iloc[:, :-1]
data_user_week2.to_excel(writer_week, 'DMA2的瞬时流量')
data_user_week2
```

Out[20]:

	23:45:00	week
2014-04-15	35.16	2
2014-04-16	34.87	3
2014-04-17	35.93	4
2014-04-18	35.41	5
2014-04-19	34.79	6

Out[20]:

	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	2
星期一	34.516250	30.003750	28.086250	27.623750	27.523750	27.557500	27.397500	27.236250	27.597500	27.646250	...	32.321250	32.850000	32.657500	32.357500	32
星期二	35.108889	30.434444	28.284444	27.706667	27.412222	27.607778	27.943333	27.876667	27.838889	27.610000	...	32.364444	32.600000	32.545556	32.215556	32
星期三	31.266667	27.231111	25.275556	24.492222	24.190000	24.244444	24.301111	24.506667	24.170000	24.022222	...	33.207778	33.465556	33.244444	32.750000	32
星期四	34.470000	30.670000	28.278750	27.468750	27.446250	27.367500	27.236250	27.196250	26.993750	27.087500	...	33.181250	33.622500	33.088750	32.750000	32
星期五	35.295000	30.996250	28.333750	27.068750	27.140000	27.328750	27.545000	26.922500	26.552500	26.645000	...	32.953750	32.913750	32.622500	32.313750	32
星期六	33.991250	29.816250	28.391250	27.135000	27.188750	27.295000	26.947500	26.726250	26.818750	26.607500	...	33.723750	33.807500	33.318750	32.646250	32
星期日	34.773750	30.236250	26.997500	26.778750	27.023750	27.125000	27.195000	26.868750	26.922500	26.808750	...	33.450000	33.587500	33.646250	33.467500	33

7 rows × 96 columns

DMA1、2用户用水量 -- 星期划分

In [21]:

```
df_time1 = pd.read_excel('按照日期处理后的数据.xlsx', sheet_name='DMA1的用户用水量', index_col=0)
df_time1['week'] = df_time1.index.dayofweek + 1
df_time1.head().iloc[:, -2:]

data_user_week1 = []
for week, data in df_time1.groupby(by='week'):
    data_user_week1.append(data.mean(0))
data_user_week1 = pd.DataFrame(data_user_week1)
data_user_week1.index.name = '星期'
data_user_week1.index = ["星期一", "星期二", "星期三", "星期四", "星期五", "星期六", "星期日"]
data_user_week1 = data_user_week1.iloc[:, :-1]
```

```
data_user_week1.to_excel(writer_week, 'DMA1的用户用水量')
data_user_week1
```

Out[21]:

	23:45:00	week
2014-04-15	23.34	2
2014-04-16	22.22	3
2014-04-17	25.56	4
2014-04-18	32.22	5
2014-04-19	26.67	6

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
星期一	26.527500	18.055000	12.640000	9.303750	6.250000	4.442500	3.193750	2.637500	2.498750	1.527500	...	37.637500	38.471250	39.722500	43.192500	45.695000
星期二	27.282222	18.395556	13.331111	9.504444	6.418889	5.057778	3.702222	3.208889	1.850000	1.974444	...	36.047778	35.801111	37.283333	41.356667	43.578889
星期三	23.826667	16.420000	11.234444	7.406667	5.431111	4.196667	3.085556	2.838889	1.481111	1.481111	...	40.491111	40.862222	41.851111	44.320000	46.296667
星期四	26.248750	18.193750	12.916250	9.445000	7.362500	6.665000	5.000000	4.165000	3.471250	3.193750	...	36.527500	37.501250	39.442500	42.361250	44.860000
星期五	28.608750	19.582500	13.191250	9.305000	6.946250	5.555000	4.026250	2.220000	1.248750	1.526250	...	36.387500	37.776250	37.776250	41.526250	43.055000
星期六	27.360000	19.445000	15.140000	10.138750	7.916250	5.555000	3.471250	3.193750	2.082500	1.526250	...	34.721250	35.417500	36.250000	39.165000	41.943750
星期日	27.082500	18.053750	12.917500	10.278750	7.776250	5.832500	4.443750	3.748750	2.637500	2.221250	...	38.332500	39.166250	40.138750	42.498750	42.916250

7 rows × 96 columns

In [ ]:

In [22]:
```
df_time2 = pd.read_excel('按照日期处理后的数据.xlsx', sheet_name='DMA2的用户用水量', index_col=0)
df_time2['week'] = df_time2.index.dayofweek + 1
df_time2.head().iloc[:, -2:]

data_user_week2 = []
for week, data in df_time2.groupby(by='week'):
    data_user_week2.append(data.mean(0))
data_user_week2 = pd.DataFrame(data_user_week2)
data_user_week2.index.name = '星期'
data_user_week2.index = ["星期一", "星期二", "星期三", "星期四", "星期五", "星期六", "星期日"]
data_user_week2 = data_user_week2.iloc[:, :-1]
data_user_week2.to_excel(writer_week, 'DMA2的用户用水量')
data_user_week2
```

Out[22]:

	23:45:00	week
2014-04-15	1.69	2
2014-04-16	1.49	3
2014-04-17	3.79	4
2014-04-18	3.64	5
2014-04-19	3.42	6

Out[22]:

	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
星期一	7.348750	3.432500	1.631250	0.741250	0.416250	0.396250	0.308750	0.181250	0.430000	0.478750	...	5.930000	6.321250	6.250000	6.062500	6.220000	6.877500
星期二	7.822222	3.627778	1.733333	0.881111	0.420000	0.480000	0.711111	0.590000	0.552222	0.323333	...	5.361111	5.566667	5.258889	5.034444	5.280000	5.414444
星期三	7.451111	3.773333	1.790000	1.078889	0.596667	0.523333	0.575556	0.697778	0.354444	0.206667	...	9.392222	9.650000	9.428889	8.934444	8.968889	9.463333
星期四	7.718750	4.405000	1.991250	0.788750	0.685000	0.631250	0.652500	0.537500	0.232500	0.326250	...	6.882500	7.236250	6.712500	6.288750	6.268750	6.942500
星期五	8.891250	4.592500	2.052500	0.682500	0.736250	0.925000	1.142500	0.518750	0.148750	0.241250	...	6.725000	6.660000	6.513750	6.266250	6.501250	6.693750
星期六	7.566250	3.646250	2.332500	0.860000	0.763750	0.870000	0.522500	0.410000	0.393750	0.182500	...	7.362500	7.430000	7.043750	6.480000	6.301250	6.342500
星期日	8.238750	4.432500	1.323750	0.698750	0.613750	0.606250	0.603750	0.405000	0.330000	0.216250	...	7.080000	7.160000	7.218750	7.061250	7.390000	6.413750

7 rows × 96 columns

In [23]:
```
writer_week.save()
writer_week.close()
```

DMA1、2的距离矩阵

In [24]:
```
writer_dismat = pd.ExcelWriter('距离矩阵.xlsx')
```

In [25]:
```
# DMA1 data
user_DMA1 = pd.read_excel("按照日期处理后的数据.xlsx", sheet_name='DMA1的用户用水量', index_col=0)
user_DMA1 = pd.concat([user_DMA1.iloc[:43, :], user_DMA1.iloc[44:, :]])
index = list(user_DMA1.index.strftime("%Y-%m-%d"))
columns = list(user_DMA1.columns)
```

```
# DMA1 distance matrix
n, m = user_DMA1.shape
dismat = []
for i in range(n):
    dis = []
    for j in range(n):
        d = ((user_DMA1.iloc[i, :] - user_DMA1.iloc[j, :])**2).sum()**0.5
        dis.append(d)
    dismat.append(dis)
pd.DataFrame(dismat, index=index, columns=index).to_excel(writer_dismat, 'DMA1的距离矩阵')
```

In [ ]:

In [26]:
```
# DMA2 data
user_DMA2 = pd.read_excel("按照日期处理后的数据.xlsx", sheet_name='DMA2的用户用水量', index_col=0)
user_DMA2 = pd.concat([user_DMA2.iloc[:43, :], user_DMA2.iloc[44:, :]])
index = list(user_DMA2.index.strftime("%Y-%m-%d"))
columns = list(user_DMA2.columns)

# DMA2 distance matrix
n, m = user_DMA2.shape
dismat = []
for i in range(n):
    dis = []
    for j in range(n):
        d = ((user_DMA2.iloc[i, :] - user_DMA2.iloc[j, :])**2).sum()**0.5
        dis.append(d)
    dismat.append(dis)
pd.DataFrame(dismat, index=index, columns=index).to_excel(writer_dismat, 'DMA2的距离矩阵')
```

In [27]:
```
writer_dismat.save()
writer_dismat.close()
```

处理数据-问题2

DMA1、2漏水量占比

In [28]:
```
InteractiveShell.ast_node_interactivity = 'all'
# InteractiveShell.ast_node_interactivity = 'last'

path = './按照日期处理后的数据.xlsx'
sheet = 'DMA1的瞬时流量'
DMA1_flow = pd.read_excel(path, sheet_name=sheet, index_col=0)
```

```
DMA1_flow.head()

path = './按照日期处理后的数据.xlsx'
sheet = 'DMA2的瞬时流量'
DMA2_flow = pd.read_excel(path, sheet_name=sheet, index_col=0)
DMA2_flow.head()

path = './按照日期处理后的数据.xlsx'
sheet = 'DMA1和DMA2的漏水量'
DMA12_flow = pd.read_excel(path, sheet_name=sheet, index_col=0)
DMA12_flow.T
```

Out[28]:

	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
2014-04-15	48.89	37.78	34.44	33.33	33.33	32.22	32.22	33.33	32.22	34.44	...	61.11	61.11	60.00	64.44	67.78	68.89	66.67
2014-04-16	52.22	43.33	42.22	41.11	38.89	37.78	35.56	35.56	31.11	31.11	...	61.11	62.22	65.56	68.89	66.67	66.67	62.22
2014-04-17	51.11	43.33	38.89	38.89	38.89	44.44	43.33	38.89	38.89	33.33	...	67.78	67.78	70.00	68.89	70.00	66.67	66.67
2014-04-18	48.89	46.67	40.00	38.89	36.67	36.67	36.67	31.11	31.11	28.89	...	67.78	68.89	70.00	68.89	66.67	67.78	72.22
2014-04-19	53.33	50.00	46.67	40.00	40.00	37.78	32.22	30.00	32.22	31.11	...	57.78	60.00	62.22	67.78	72.22	73.33	74.44

5 rows × 96 columns

Out[28]:

	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
2014-04-15	36.98	29.88	29.04	29.96	30.96	32.04	32.98	33.82	34.51	34.27	...	31.76	31.19	33.52	33.53	33.18	34.01	34.93
2014-04-16	35.06	30.78	31.44	32.99	34.11	35.46	36.19	37.22	33.81	33.38	...	34.42	34.77	34.42	33.89	34.08	34.40	32.98
2014-04-17	35.84	34.35	31.17	33.44	34.44	35.32	36.01	34.94	32.14	32.94	...	31.80	33.80	33.32	32.43	32.66	33.46	33.71
2014-04-18	35.67	34.63	30.79	32.36	32.82	34.08	35.50	32.21	31.77	32.38	...	34.46	34.48	33.93	33.02	33.29	33.59	34.07
2014-04-19	33.26	32.20	32.93	31.60	33.24	34.28	31.46	30.50	31.37	32.00	...	33.59	33.84	32.91	31.98	32.00	32.11	33.00

5 rows × 96 columns

Out[28]:

当地时间(北京时间)	2014-04-15	2014-04-16	2014-04-17	2014-04-18	2014-04-19	2014-04-20	2014-04-21	2014-04-22	2014-04-23	2014-04-24	...	2014-06-03	2014-06-04	2014-06-05	2014-06-06	2014-06-07	2014-06-08	2014-06-09	2014-06-10	2014-06-11	2014-06-12
DMA1	32.22	30.00	32.22	28.89	31.11	33.33	31.11	32.22	28.89	28.89	...	17.78	17.78	16.67	20.00	20.00	17.78	17.78	18.89	17.78	18.89
DMA2	33.47	33.38	32.14	31.77	31.37	34.51	35.59	32.49	32.30	35.09	...	24.28	24.27	24.14	24.21	24.46	24.14	24.11	24.17	24.23	24.27

2 rows × 59 columns

In [ ]:

In [29]:
```
writer_leaking_precent = pd.ExcelWriter('漏水量占比.xlsx')
```

In [30]:
```
# InteractiveShell.ast_node_interactivity = 'all'
InteractiveShell.ast_node_interactivity = 'last'

# DMA1 漏水量占比
index = list(DMA1_flow.index)
columns = list(DMA1_flow.columns)
delta1 = DMA12_flow.iloc[:-1, 0].values.reshape(-1, 1) / DMA1_flow.values
delta1 = pd.DataFrame(delta1, index=index, columns=columns).fillna(0)
delta1.replace(np.inf, 0, inplace=True)
delta1.to_excel(writer_leaking_precent, 'DMA1漏水量占比')

# DMA2 漏水量占比
index = list(DMA2_flow.index)
columns = list(DMA2_flow.columns)
delta2 = DMA12_flow.iloc[:-1, 0].values.reshape(-1, 1) / DMA2_flow.values
delta2 = pd.DataFrame(delta2, index=index, columns=columns).fillna(0)
delta2.replace(np.inf, 0, inplace=True)
delta2.to_excel(writer_leaking_precent, 'DMA2漏水量占比')
# 这里会有一个除 0 的警告，暂时不管，后面处理数据把 Nan 或者 inf 换掉即可（因为告警中有个人信息，上面设置了忽略警告，所以这里没有打印告警）
```

In [31]:
```
# InteractiveShell.ast_node_interactivity = 'all'
InteractiveShell.ast_node_interactivity = 'last'

# DMA1 漏水量占比的均值
delta1_avg = pd.DataFrame(delta1.mean(1)).T
delta1_avg

# DMA2 漏水量占比的均值
delta2_avg = pd.DataFrame(delta2.mean(1)).T
```

```
delta2_avg

delta_avg = pd.concat([delta1_avg, delta2_avg], ignore_index=True)
delta_avg.rename({0: "DMA1", 1: "DMA2"}, inplace=True)
delta_avg = delta_avg.T
delta_avg.to_excel(writer_leaking_precent, 'DMA1和DMA2漏水量占比的均值')
```

In [32]:
```
writer_leaking_precent.save()
writer_leaking_precent.close()
```

In [ ]:
```

```

In [33]:
```
delta_avg.T
```

Out[33]:

	2014-04-15	2014-04-16	2014-04-17	2014-04-18	2014-04-19	2014-04-20	2014-04-21	2014-04-22	2014-04-23	2014-04-24	...	2014-06-02	2014-06-03	2014-06-04	2014-06-05	2014-06-06	20 06
DMA1	0.685302	0.625730	0.655978	0.596069	0.637407	0.663768	0.665214	0.661711	0.647750	0.638458	...	0.502560	0.469319	0.465482	0.437515	0.501165	0.510
DMA2	0.940355	0.867177	0.937448	0.849442	0.919896	0.970531	0.891716	0.939145	0.845115	0.837366	...	0.729062	0.582821	0.594049	0.561922	0.676238	0.675

2 rows × 58 columns

DMA1、DMA2的漏水量和漏水量占比

In [34]:
```
path_leaking = './按照日期处理后的数据.xlsx'
path_leaking_precent = './漏水量占比.xlsx'
sheet_leaking = 'DMA1和DMA2的漏水量'
sheet_leaking_precent = 'DMA1和DMA2漏水量占比的均值'

data_leaking = pd.read_excel(path_leaking, sheet_name=sheet_leaking, index_col=0).iloc[:-1, :]
data_leaking_precent = pd.read_excel(path_leaking_precent, sheet_name=sheet_leaking_precent, index_col=0)
data_leaking_precent.index = list(data_leaking.index)

data_leaking.shape, data_leaking_precent.shape
```

Out[34]:
```
((58, 2), (58, 2))
```

In [35]:
```
writer_question2 = pd.ExcelWriter('问题2数据.xlsx')
```

In [36]:
```
InteractiveShell.ast_node_interactivity = 'all'
# InteractiveShell.ast_node_interactivity = 'last'
```

```
# question2 DMA1
DMA1_q2 = pd.concat([data_leaking.iloc[:, 0], data_leaking_precent.iloc[:, 0]], axis=1, ignore_index=True)
DMA1_q2.columns = ["漏水量", "漏水量占比"]
DMA1_q2.to_excel(writer_question2, 'DMA1')

# question2 DMA2
DMA2_q2 = pd.concat([data_leaking.iloc[:, 1], data_leaking_precent.iloc[:, 1]], axis=1, ignore_index=True)
DMA2_q2.columns = ["漏水量", "漏水量占比"]
DMA2_q2.to_excel(writer_question2, 'DMA2')
```

In [37]:
```
writer_question2.save()
writer_question2.close()
```

In [ ]:
```
```