问题1

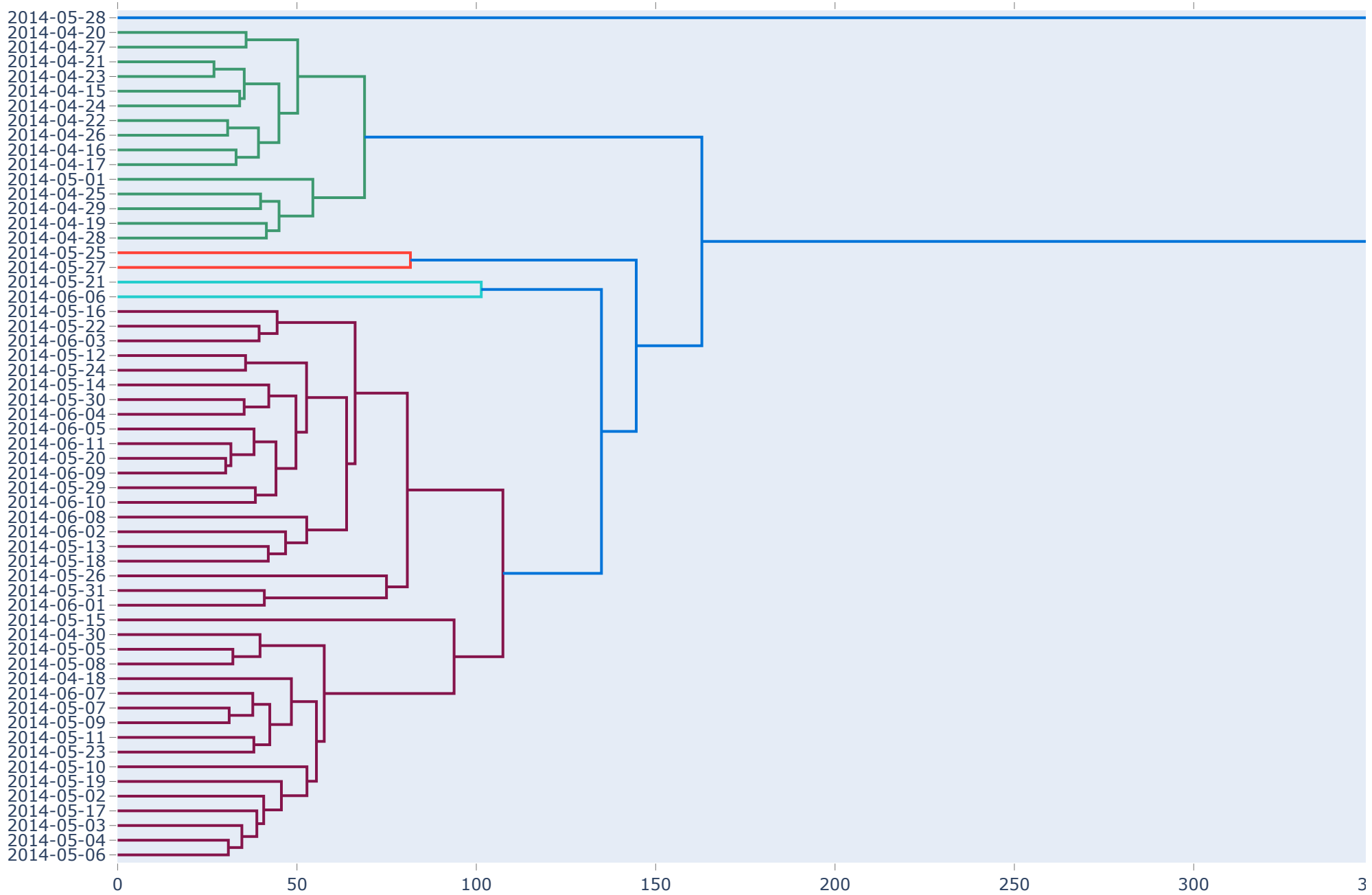
In [1]:
```
import numpy as np
import pandas as pd
import cufflinks as cf

import scipy
import scipy.cluster.hierarchy as sch
from sklearn.metrics import *

import plotly
import plotly.express as px
import plotly.graph_objects as go
import plotly.figure_factory as ff

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
```

层次聚类 (尝试)

保存 5-28 号

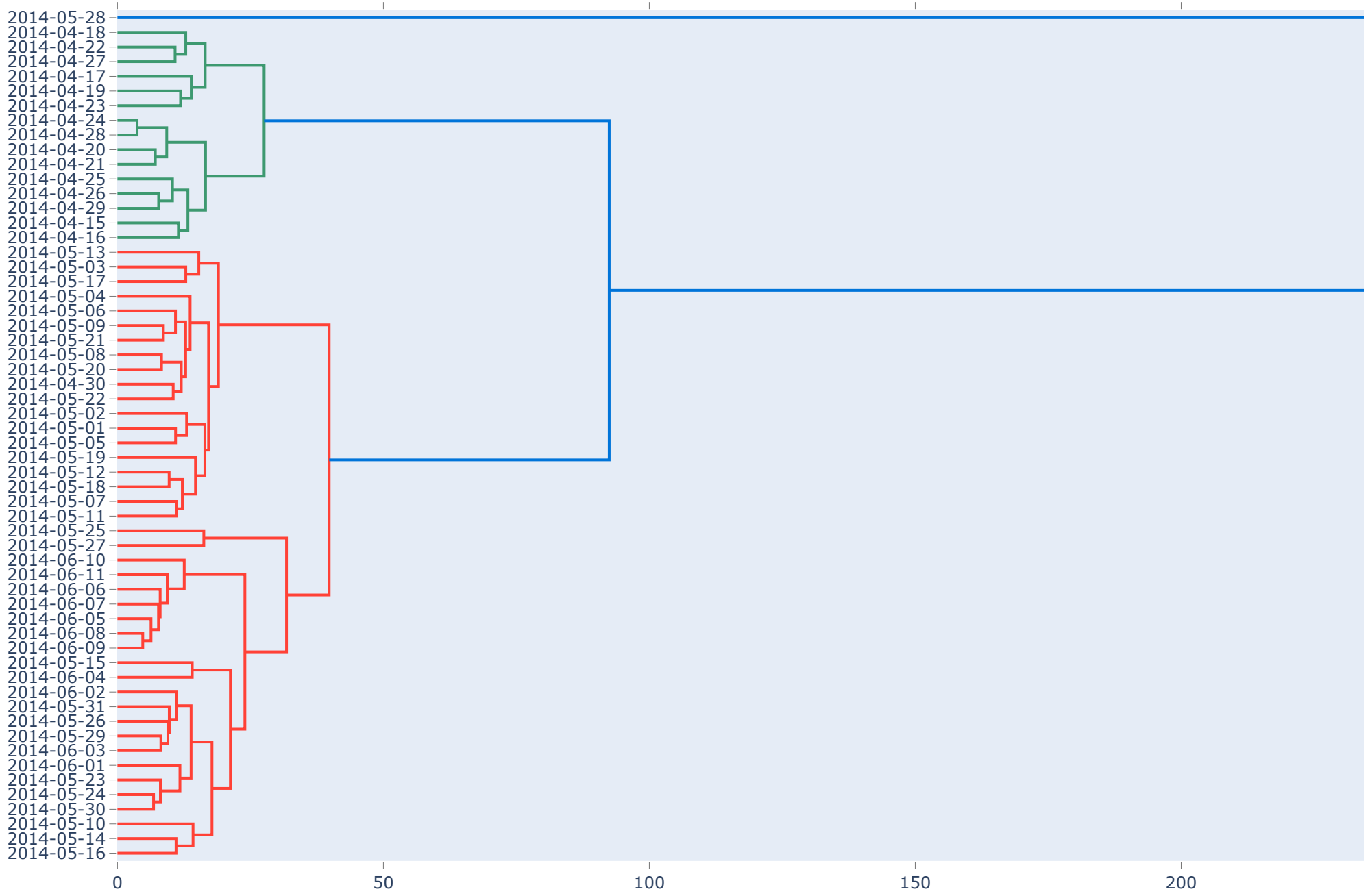
In [2]:
```
# TODO 尝试 DMA1 用水量
X = pd.read_excel("按照日期处理后的数据.xlsx", sheet_name='DMA1的用户用水量', index_col=0)
index = list(X.index.strftime("%Y-%m-%d"))
columns = list(X.columns)
fig = ff.create_dendrogram(X, orientation='left', labels=index, color_threshold=120)
fig.update_layout(
    width=1200,
    height=800,
```

```
    yaxis=dict(range=[-580, 0]),
)
fig.write_image('./img/svg/DMA1-保留5-28号的聚类树状图.svg')
fig.show()
```


2014-05-28
2014-04-20
2014-04-27
2014-04-21
2014-04-23
2014-04-15
2014-04-24
2014-04-22
2014-04-26
2014-04-16
2014-04-17
2014-05-01
2014-04-25
2014-04-29
2014-04-19
2014-04-28
2014-05-25
2014-05-27
2014-05-21
2014-06-06
2014-05-16
2014-05-22
2014-06-03
2014-05-12
2014-05-24
2014-05-14
2014-05-30
2014-06-04
2014-06-05
2014-06-11
2014-05-20
2014-06-09
2014-05-29
2014-06-10
2014-06-08
2014-06-02
2014-05-13
2014-05-18
2014-05-26
2014-05-31
2014-06-01
2014-05-15
2014-04-30
2014-05-05
2014-05-08
2014-04-18
2014-06-07
2014-05-07
2014-05-09
2014-05-11
2014-05-23
2014-05-10
2014-05-19
2014-05-02
2014-05-17
2014-05-03
2014-05-04
2014-05-06
0
50
100
150
200
250
300

In [ ]:

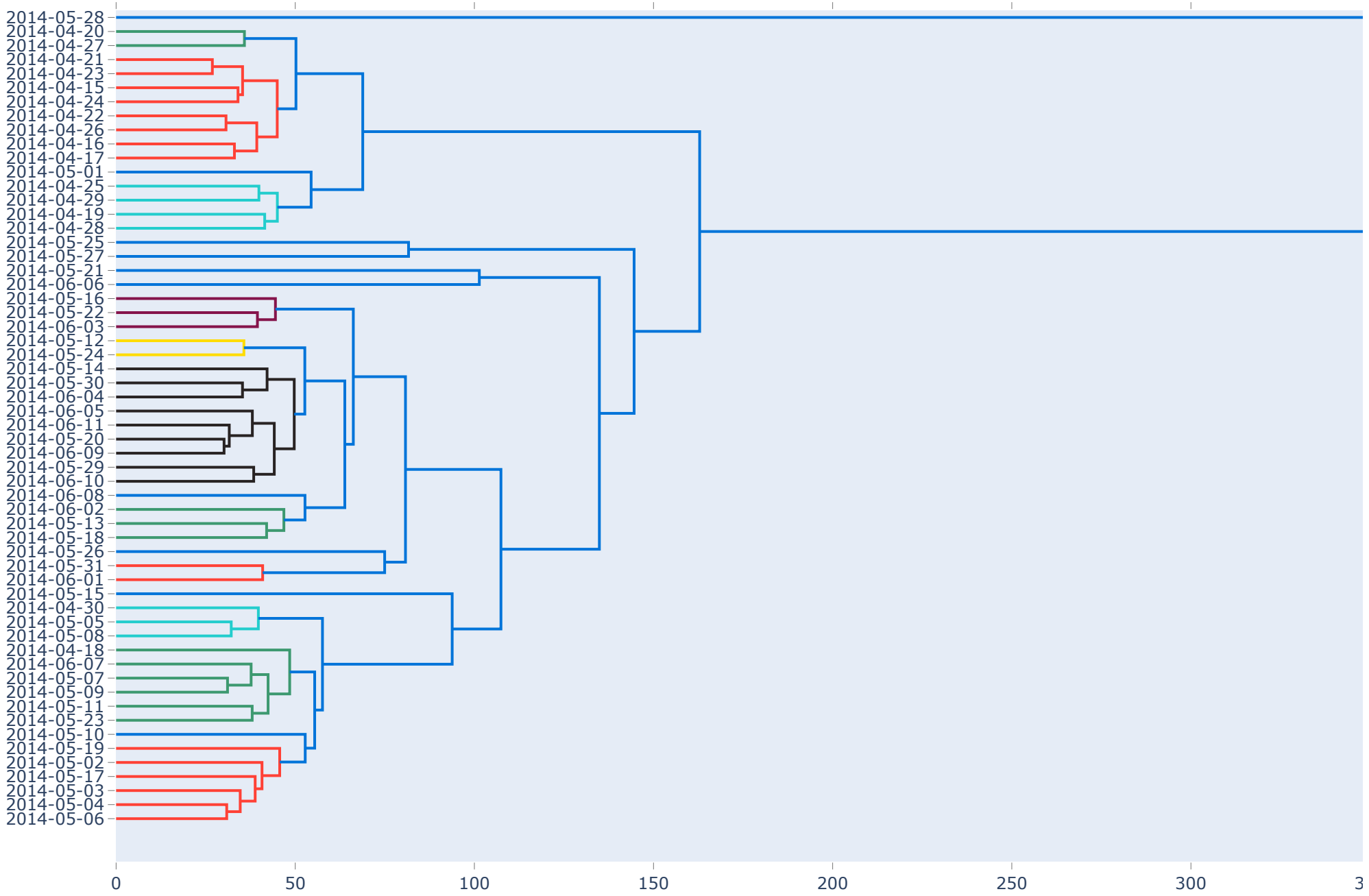
In [3]:
```
# TODO 尝试 DMA2 用水量
X = pd.read_excel("按照日期处理后的数据.xlsx", sheet_name='DMA2的用户用水量', index_col=0)
index = list(X.index.strftime("%Y-%m-%d"))
columns = list(X.columns)
fig = ff.create_dendrogram(X, orientation='left', labels=index, color_threshold=50)
fig.update_layout(
    width=1200,
    height=800,
    yaxis=dict(range=[-580, 0]),
)
fig.write_image('./img/svg/DMA2-保留5-28号的聚类树状图.svg')
fig.show()
```


2014-05-28
2014-04-18
2014-04-22
2014-04-27
2014-04-17
2014-04-19
2014-04-23
2014-04-24
2014-04-28
2014-04-20
2014-04-21
2014-04-25
2014-04-26
2014-04-29
2014-04-15
2014-04-16
2014-05-13
2014-05-03
2014-05-17
2014-05-04
2014-05-06
2014-05-09
2014-05-21
2014-05-08
2014-05-20
2014-04-30
2014-05-22
2014-05-02
2014-05-01
2014-05-05
2014-05-19
2014-05-12
2014-05-18
2014-05-07
2014-05-11
2014-05-25
2014-05-27
2014-06-10
2014-06-11
2014-06-06
2014-06-07
2014-06-05
2014-06-08
2014-06-09
2014-05-15
2014-06-04
2014-06-02
2014-05-31
2014-05-26
2014-05-29
2014-06-03
2014-06-01
2014-05-23
2014-05-24
2014-05-30
2014-05-10
2014-05-14
2014-05-16
0
50
100
150
200

划分时间段的层次聚类（尝试）

只做了DMA1，效果一般，猜测DMA2可能差不多，最终没有使用该方法

In [4]:
```
# TODO 尝试 DMA1 用水量
X = 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)
X.head()
index = list(X.index.strftime("%Y-%m-%d"))
columns = list(X.columns)
fig = ff.create_dendrogram(X, orientation='left', labels=index, color_threshold=50)
fig.update_layout(width=1200, height=800)
fig.show()
```


2014-05-28
2014-04-20
2014-04-27
2014-04-21
2014-04-23
2014-04-15
2014-04-24
2014-04-22
2014-04-26
2014-04-16
2014-04-17
2014-05-01
2014-04-25
2014-04-29
2014-04-19
2014-04-28
2014-05-25
2014-05-27
2014-05-21
2014-06-06
2014-05-16
2014-05-22
2014-06-03
2014-05-12
2014-05-24
2014-05-14
2014-05-30
2014-06-04
2014-06-05
2014-06-11
2014-05-20
2014-06-09
2014-05-29
2014-06-10
2014-06-08
2014-06-02
2014-05-13
2014-05-18
2014-05-26
2014-05-31
2014-06-01
2014-05-15
2014-04-30
2014-05-05
2014-05-08
2014-04-18
2014-06-07
2014-05-07
2014-05-09
2014-05-11
2014-05-23
2014-05-10
2014-05-19
2014-05-02
2014-05-17
2014-05-03
2014-05-04
2014-05-06
0
50
100
150
200
250
300

In [ ]:

In [5]:
```
# TODO 按照时间段分离数据

quarters = np.array([5, 6, 2, 8, 3]) * 4
quarters = quarters.cumsum()
quarters
columns = time_period = ['0:00-5:00', '5:00-11:00', '11:00-13:00', '13:00-21:00', '21:00-24:00']

data_time_split_sum = pd.DataFrame(columns=columns)
data_time_split_sum.loc[:, columns[0]] = X.iloc[:, :quarters[0]].sum(1)
for i in range(1, len(quarters)):
    data_time_split_sum.loc[:, columns[i]] = X.iloc[:, quarters[i-1]:quarters[i]].sum(1)
print("sum")
data_time_split_sum

data_time_split_mean = pd.DataFrame(columns=columns)
data_time_split_mean.loc[:, columns[0]] = X.iloc[:, :quarters[0]].mean(1)
for i in range(1, len(quarters)):
    data_time_split_mean.loc[:, columns[i]] = X.iloc[:, quarters[i-1]:quarters[i]].mean(1)
print("mean")
data_time_split_mean.head(10)
```

```
sum
mean
```

Out[5]:

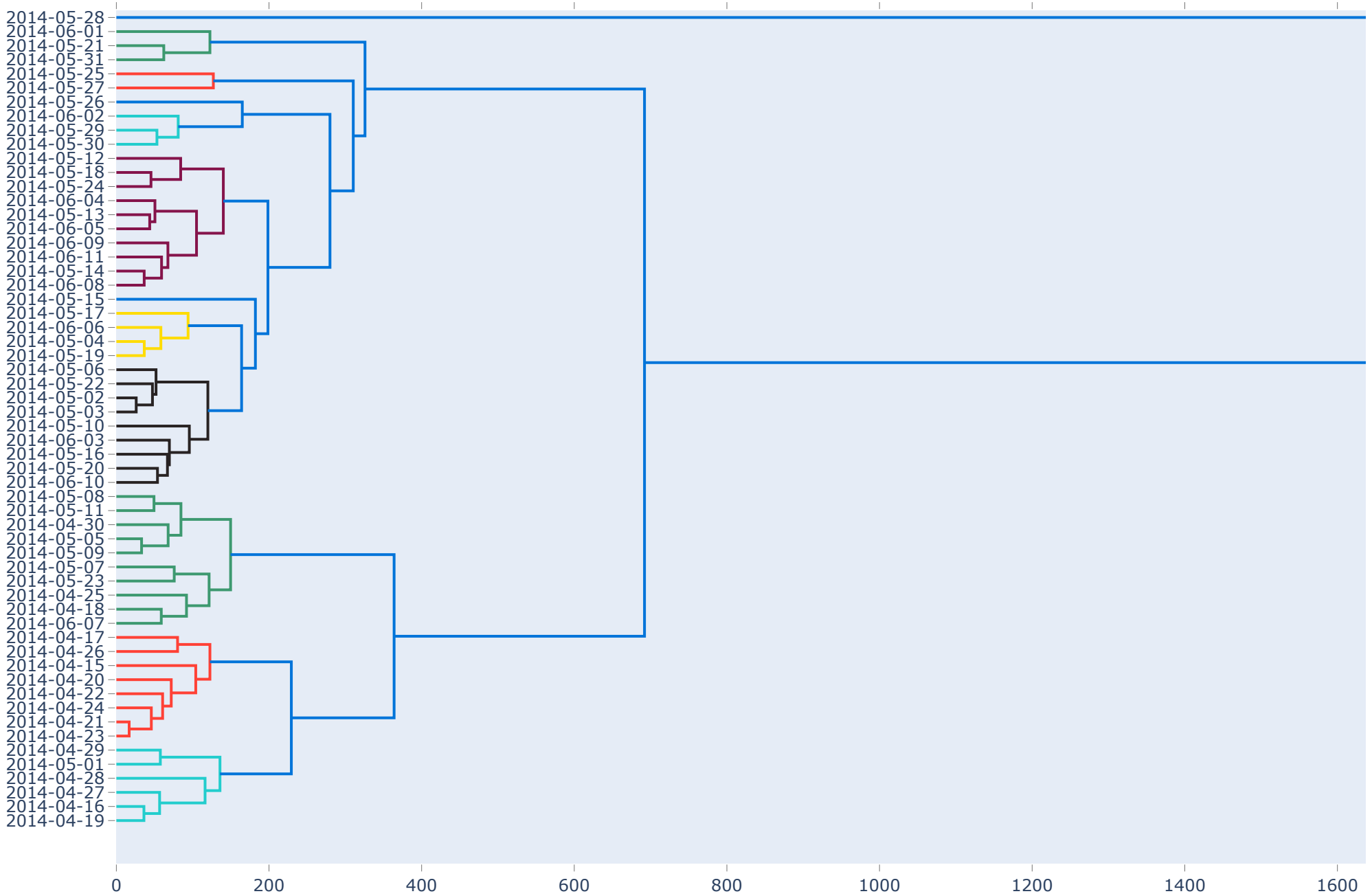
	0:00-5:00	5:00-11:00	11:00-13:00	13:00-21:00	21:00-24:00
2014-04-15	2.2780	15.232083	27.36500	21.808438	30.465000
2014-04-16	6.1105	18.471667	33.60875	23.924688	31.945000
2014-04-17	5.5010	19.030000	28.89125	21.079688	33.336667
2014-04-18	4.9995	19.812917	38.05625	30.207812	37.870000
2014-04-19	5.0550	19.122083	36.11250	23.995000	33.149167
2014-04-20	4.9480	16.948750	34.03250	23.336250	27.780833
2014-04-21	3.6105	15.926250	29.58375	22.433125	29.075833
2014-04-22	5.5010	17.733750	28.47375	21.911875	28.890833
2014-04-23	4.0535	15.924167	28.61000	22.569688	29.999167
2014-04-24	5.0540	16.247500	29.58125	22.290937	31.944167

In [ ]:

```
```

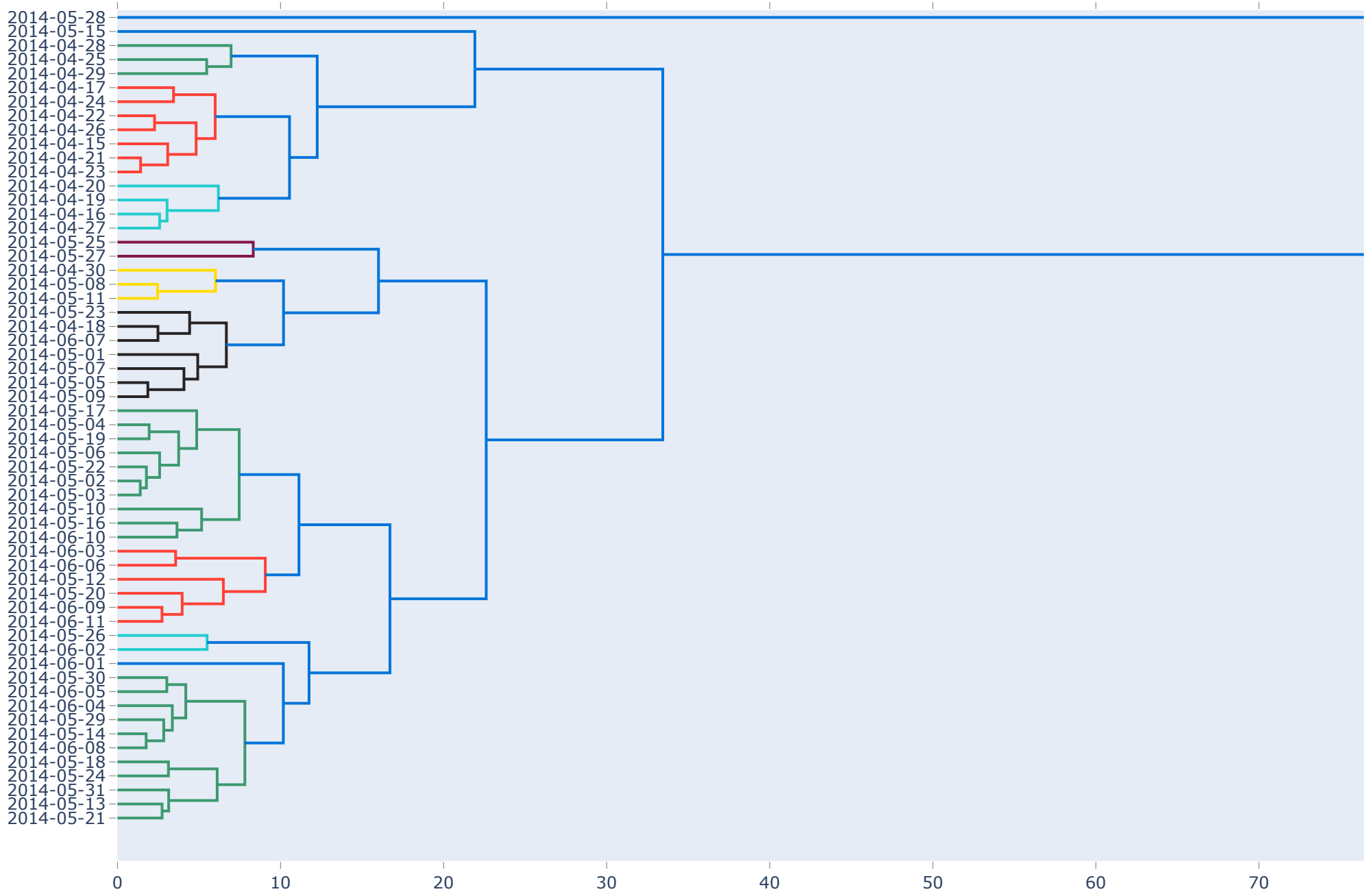
In [6]:

```
# TODO 对按照时间段分离的数据进行层次聚类 SUM

index = list(data_time_split_sum.index.strftime("%Y-%m-%d"))
columns = list(data_time_split_sum.columns)
fig = ff.create_dendrogram(
    data_time_split_sum,
    orientation='left',
    labels=index,
    color_threshold=150,
)
fig.update_layout(width=1200, height=800)
fig.show()
```


2014-05-28
2014-06-01
2014-05-21
2014-05-31
2014-05-25
2014-05-27
2014-05-26
2014-06-02
2014-05-29
2014-05-30
2014-05-12
2014-05-18
2014-05-24
2014-06-04
2014-05-13
2014-06-05
2014-06-09
2014-06-11
2014-05-14
2014-06-08
2014-05-15
2014-05-17
2014-06-06
2014-05-04
2014-05-19
2014-05-06
2014-05-22
2014-05-02
2014-05-03
2014-05-10
2014-06-03
2014-05-16
2014-05-20
2014-06-10
2014-05-08
2014-05-11
2014-04-30
2014-05-05
2014-05-09
2014-05-07
2014-05-23
2014-04-25
2014-04-18
2014-06-07
2014-04-17
2014-04-26
2014-04-15
2014-04-20
2014-04-22
2014-04-24
2014-04-21
2014-04-23
2014-04-29
2014-05-01
2014-04-28
2014-04-27
2014-04-16
2014-04-19
0
200
400
600
800
1000
1200
1400
1600

In [ ]:

In [7]:
```
# TODO 对按照时间段分离的数据进行层次聚类 MEAN

index = list(data_time_split_mean.index.strftime("%Y-%m-%d"))
columns = list(data_time_split_mean.columns)
fig = ff.create_dendrogram(
    data_time_split_mean,
    orientation='left',
    labels=index,
    color_threshold=10,
)
fig.update_layout(width=1200, height=800)
fig.show()
```


2014-05-28
2014-05-15
2014-04-28
2014-04-25
2014-04-29
2014-04-17
2014-04-24
2014-04-22
2014-04-26
2014-04-15
2014-04-21
2014-04-23
2014-04-20
2014-04-19
2014-04-16
2014-04-27
2014-05-25
2014-05-27
2014-04-30
2014-05-08
2014-05-11
2014-05-23
2014-04-18
2014-06-07
2014-05-01
2014-05-07
2014-05-05
2014-05-09
2014-05-17
2014-05-04
2014-05-19
2014-05-06
2014-05-22
2014-05-02
2014-05-03
2014-05-10
2014-05-16
2014-06-10
2014-06-03
2014-06-06
2014-05-12
2014-05-20
2014-06-09
2014-06-11
2014-05-26
2014-06-02
2014-06-01
2014-05-30
2014-06-05
2014-06-04
2014-05-29
2014-05-14
2014-06-08
2014-05-18
2014-05-24
2014-05-31
2014-05-13
2014-05-21
0
10
20
30
40
50
60
70

层次聚类 (正式)

剔除 5-28 号

DMA 1 日期 用水量聚类

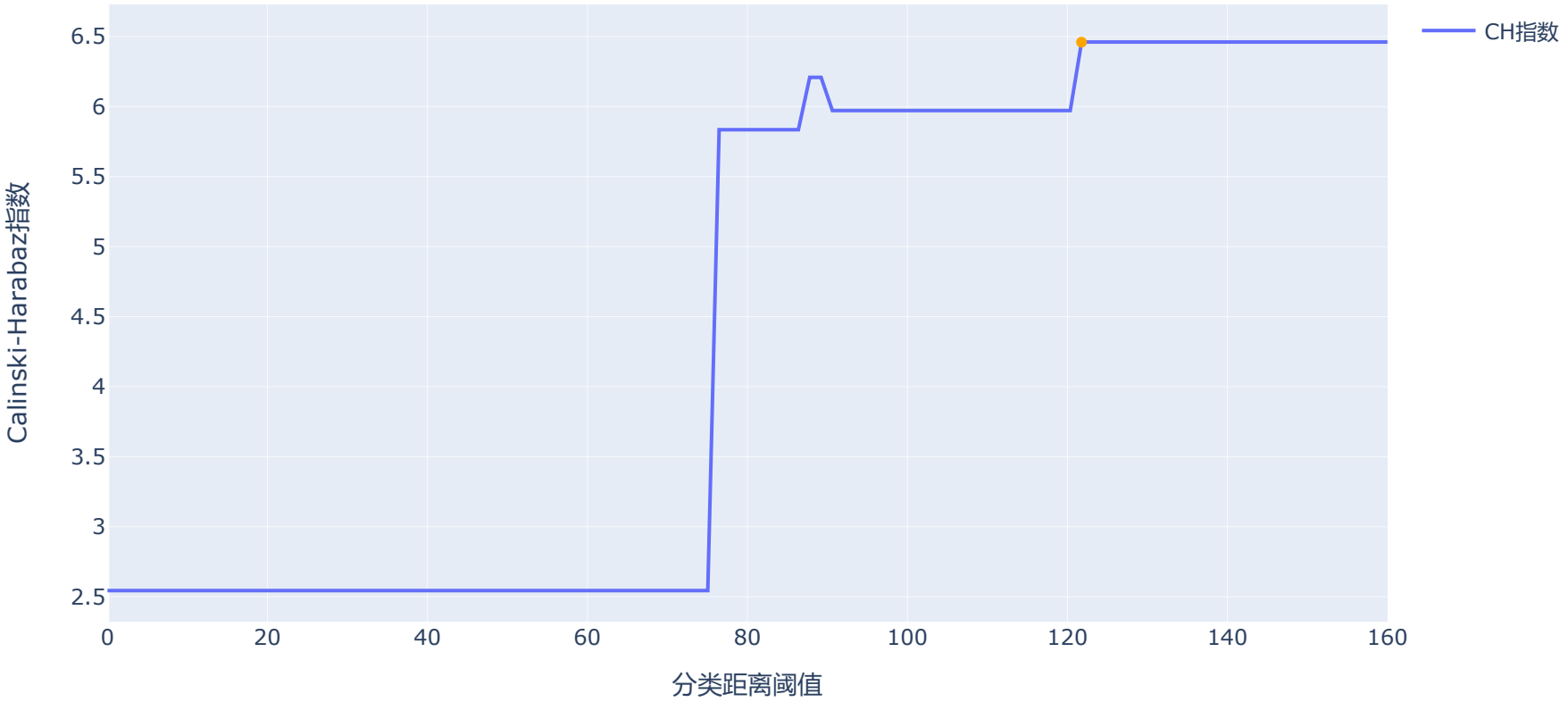
In [8]:
```
# DMA1 data
user_DMA1 = pd.read_excel("按照日期处理后的数据.xlsx", sheet_name='DMA1的用户用水量', index_col=0)
user_DMA1 = pd.concat([user_DMA1.iloc[:43, :], user_DMA1.iloc[44:, :]])  # 剔除 5-28
index = list(user_DMA1.index.strftime("%Y-%m-%d"))
columns = list(user_DMA1.columns)
```

In [ ]:
```
```

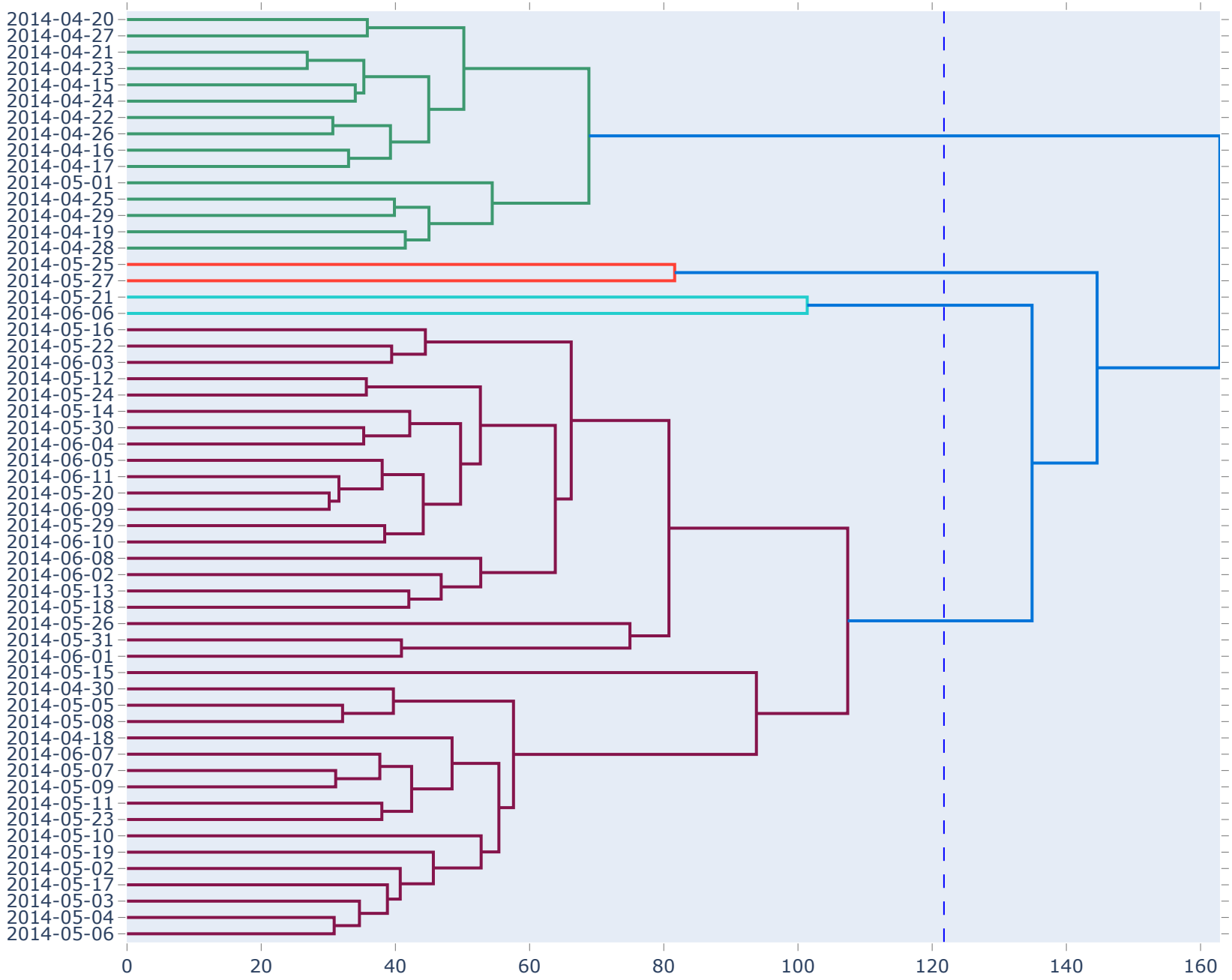
In [9]:
```
# InteractiveShell.ast_node_interactivity = 'all'
InteractiveShell.ast_node_interactivity = 'last'

dis_arr = np.array(user_DMA1)
disMat = sch.distance.pdist(dis_arr, 'euclidean')
Z = sch.linkage(disMat)
ch_score = []
b = 1.14
t = np.linspace(0, b, int(100*(b)+1))
tt = np.linspace(0, 160, int(100*(b)+1))
for d in t:
    cluster = sch.fcluster(Z, d, 'inconsistent')
    s = calinski_harabasz_score(user_DMA1, cluster)
    ch_score.insert(0, s)
    ch_score.insert(0, ch_score[0])
    ch_score.pop()
# len(set(sch.fcluster(Z, 0.88, 'inconsistent')))
trace = go.Scatter(x=tt, y=ch_score, mode='lines', name='CH指数')
fig = go.Figure(data=trace)
fig.update_layout(
    width=910,
    xaxis=dict(title='分类距离阈值'),
    yaxis=dict(title='Calinski-Harabaz指数'),
    title_text="DMA1用水量-Calinski-Harabaz指数随分类距离阈值的变化情况",
)
fig.add_trace(go.Scatter(
```

```
    x=[121.76], y=[6.46],
    line=dict(color='orange', width=5),
    showlegend=False,
))
fig.write_image('./img/svg/DMA1用水量-Calinski-Harabaz指数随分类距离阈值的变化情况.svg')
fig.show()

fig = ff.create_dendrogram(user_DMA1, orientation='left', labels=index, )
fig.update_layout(
    width=900,
    height=800,
    yaxis=dict(range=[-570, 0]),
    title_text='DMA1用水量-对日期的层次聚类树状图',
)
fig.add_trace(go.Scatter(
    x=[121.76] * len(ch_score),
    y=np.linspace(-570, 0, len(ch_score)),
    mode='lines',
    line=dict(color='blue', width=1, dash='dash'),
))
fig.write_image('./img/svg/DMA1用水量-对日期进行层次聚类结果.svg')
fig.show()
```

DMA1用水量-Calinski-Harabaz指数随分类距离阈值的变化情况



DMA1用水量-对日期的层次聚类树状图



In [ ]:

DMA 2 日期 用水量聚类

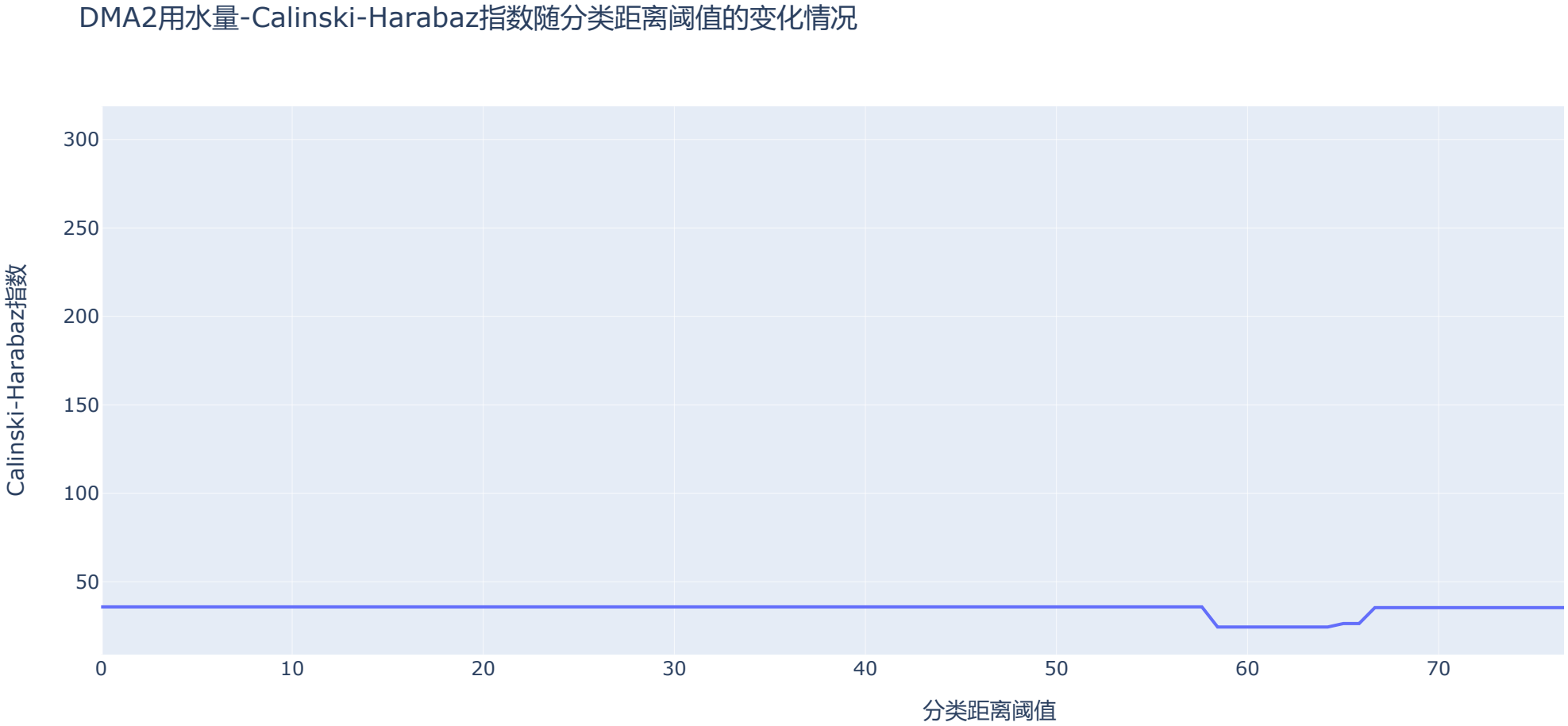
In [ ]:

In [10]:
```
# 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:, :]])  # 剔除 5-28
index = list(user_DMA2.index.strftime("%Y-%m-%d"))
columns = list(user_DMA2.columns)
```

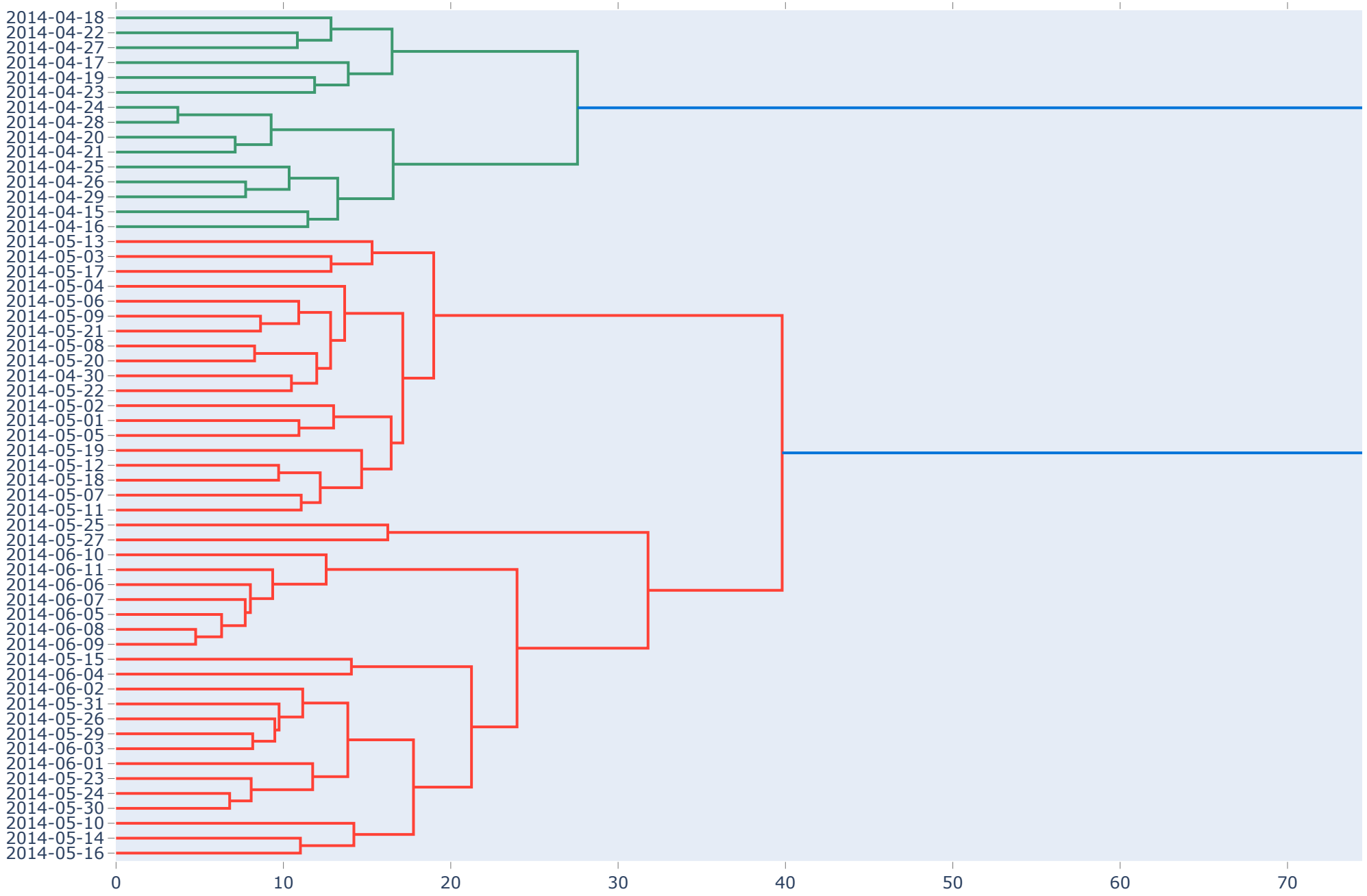
In [ ]:

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

dis_arr = np.array(user_DMA2)
disMat = sch.distance.pdist(dis_arr, 'euclidean')
Z = sch.linkage(disMat)
# P = sch.dendrogram(Z)
# plt.show()

ch_score = []
b = 1.14
t = np.linspace(0, b, int(100*(b)+1))
tt = np.linspace(0, 93, int(100*(b)+1))
for d in t:
    cluster = sch.fcluster(Z, d, 'inconsistent')  # 聚类结果
    s = calinski_harabasz_score(user_DMA2, cluster)
    ch_score.append(s)
# len(set(sch.fcluster(Z, 0.97, 'inconsistent')))
trace = go.Scatter(x=tt, y=ch_score, mode='lines', name='CH指数')
fig = go.Figure(data=trace)
fig.update_layout(
    width=1320,
    xaxis=dict(title='分类距离阈值'),
    yaxis=dict(title='Calinski-Harabaz指数'),
    title_text="DMA2用水量-Calinski-Harabaz指数随分类距离阈值的变化情况",
)
fig.add_trace(go.Scatter(
    x=[79.83], y=[299.8],
```

```
    line=dict(color='orange', width=5),
    showlegend=False,
))
fig.write_image('./img/svg/DMA2用水量-Calinski-Harabaz指数随分类距离阈值的变化情况.svg')
fig.show()


fig = ff.create_dendrogram(user_DMA2, orientation='left', labels=index)
fig.update_layout(
    width=1300,
    height=800,
    yaxis=dict(range=[-570, 0]),
    title_text='DMA2用水量-对日期的层次聚类树状图',
)
fig.add_trace(go.Scatter(
    x=[79.83] * len(ch_score),
    y=np.linspace(-570, 0, len(ch_score)),
    mode='lines',
    line=dict(color='blue', width=1, dash='dash'),
))
fig.write_image('./img/svg/DMA2用水量-对日期进行层次聚类结果.svg')
fig.show()
```


DMA2用水量-Calinski-Harabaz指数随分类距离阈值的变化情况
Calinski-Harabaz指数
300
250
200
150
100
50
0
10
20
30
40
50
60
70
分类距离阈值

DMA2用水量-对日期的层次聚类树状图



In [ ]: