In [4]:

```
#-*- utf-8 -*-
# 离差标准化（最大最小规范化）：保留了原来数据中存在的关系，消除梁刚和数据取值范围影响最简单的方法
# 目标：消除数量级数据带来的影响，数据标准化到[0,1]
import pandas as pd

filename = 'business_circle.xls'
data = pd.read_excel(filename, index_col = u'基站编号')
data = (data - data.min()) / (data.max() - data.min()) # 离差标准化
data.to_excel('standardization.xls')
data
```

Out[4]:

	工作日上班时间人均停留时间	凌晨人均停留时间	周末人均停留时间	日均人流量
基站编号				
36902	0.103865	0.856364	0.850539	0.169153
36903	0.263285	1.000000	0.725732	0.118210
36904	0.144928	0.740000	0.644068	0.038909
36905	0.082126	0.992727	0.993837	0.020031
36906	0.374396	0.867273	0.987673	0.102217
...	...	...	...	...
35562	0.125604	0.081818	0.291217	0.608771
38624	0.152174	0.072727	0.354391	0.590718
36017	0.205314	0.003636	0.129430	0.973539
38827	0.154589	0.089091	0.118644	0.927129
37787	0.154589	0.001818	0.329738	0.802984

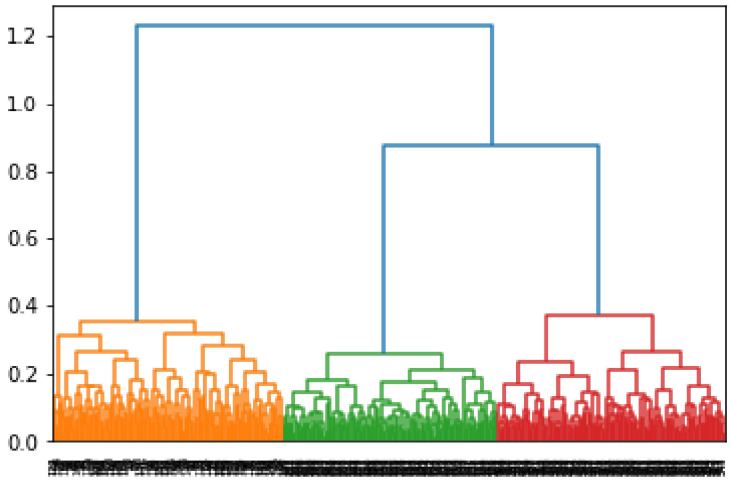
431 rows × 4 columns

In [5]:

```
#-*- coding:utf-8 -*-
# 画谱系聚类图

import pandas as pd

# 参数初始化
filename = 'standardization.xls'
data = pd.read_excel(filename, index_col = u'基站编号')

import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import linkage, dendrogram
# 这里使用scipy的层次聚类函数
```

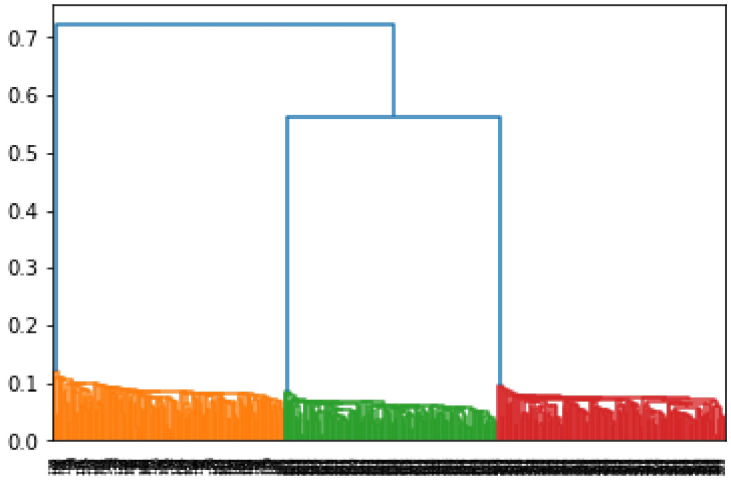
In [6]:

```
z = linkage(data, method = 'weighted', metric = 'euclidean') # method = 'weighted'时的谱系聚类图

p = dendrogram(z, 0) # 画谱系聚类图
plt.show()
```



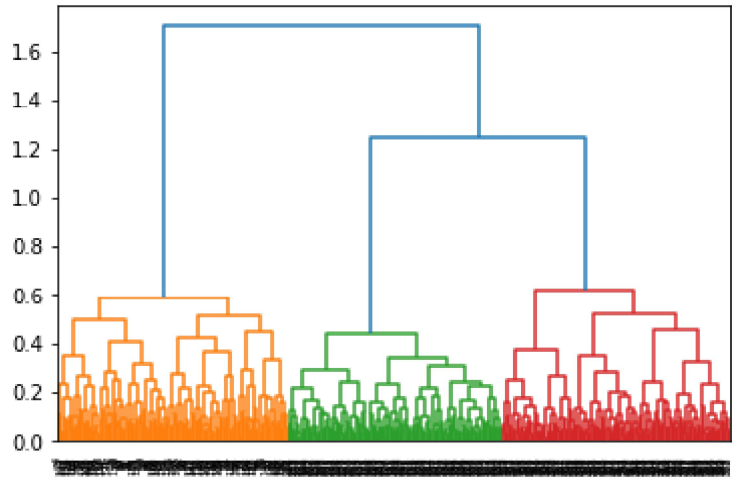
In [7]:

```
z = linkage(data, method = 'single', metric = 'euclidean') # method = 'single'谱系聚类图

p = dendrogram(z, 0) # 画谱系聚类图
plt.show()
```



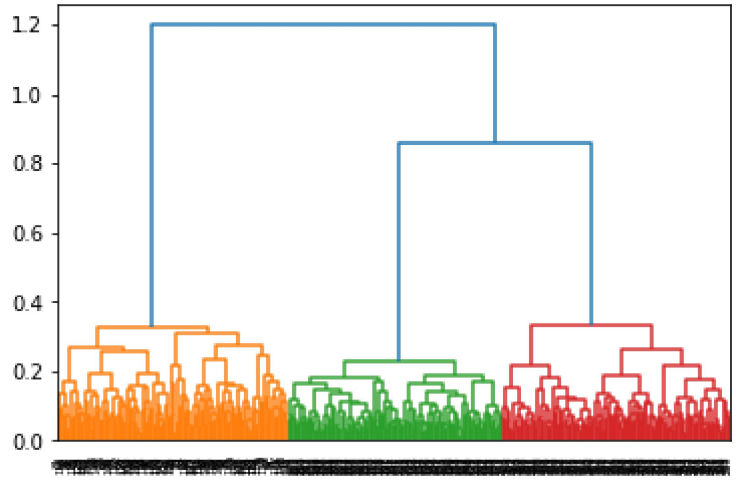
In [8]:

```
z = linkage(data, method = 'complete', metric = 'euclidean') # method = 'complete'谱系聚类图

p = dendrogram(z, 0) # 画谱系聚类图
plt.show()
```



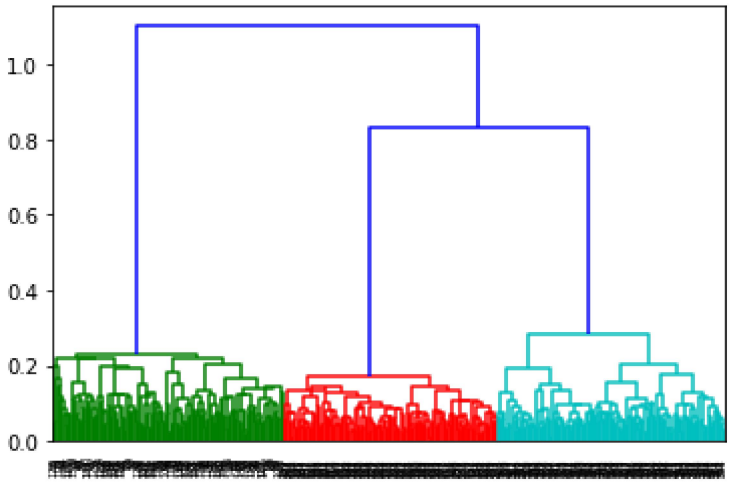
In [9]:

```
z = linkage(data, method = 'average', metric = 'euclidean') # method = 'average'谱系聚类图

p = dendrogram(z, 0) # 画谱系聚类图
plt.show()
```



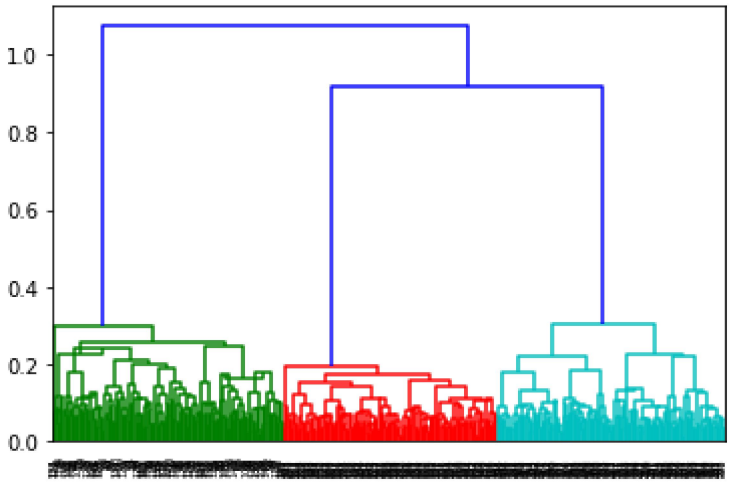
In [6]:

```
z = linkage(data, method = 'centroid', metric = 'euclidean') # method = 'centroid'谱系聚类图

p = dendrogram(z, 0) # 画谱系聚类图
plt.show()
```



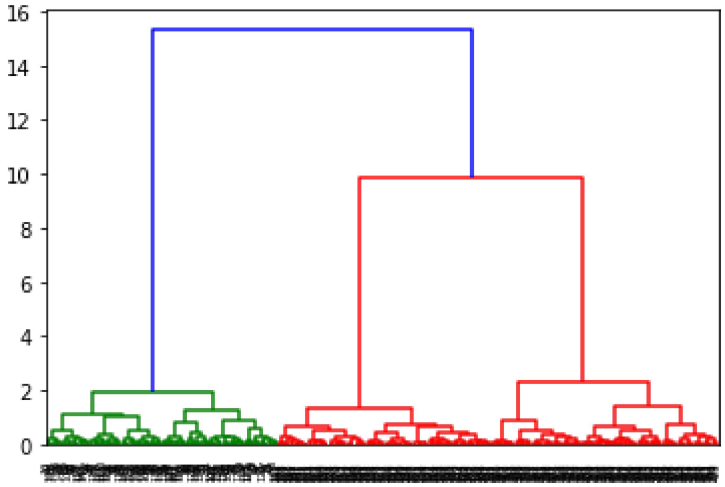
In [7]:

```
z = linkage(data, method = 'median', metric = 'euclidean') # method = 'median'谱系聚类图

p = dendrogram(z, 0) # 画谱系聚类图
plt.show()
```



In [8]:

```
z = linkage(data, method = 'ward', metric = 'euclidean') # method = 'ward'谱系聚类图 # Ward方差最

p = dendrogram(z, 0) # 画谱系聚类图
plt.savefig('puxijulei.jpg')
plt.show()
```



In [18]:

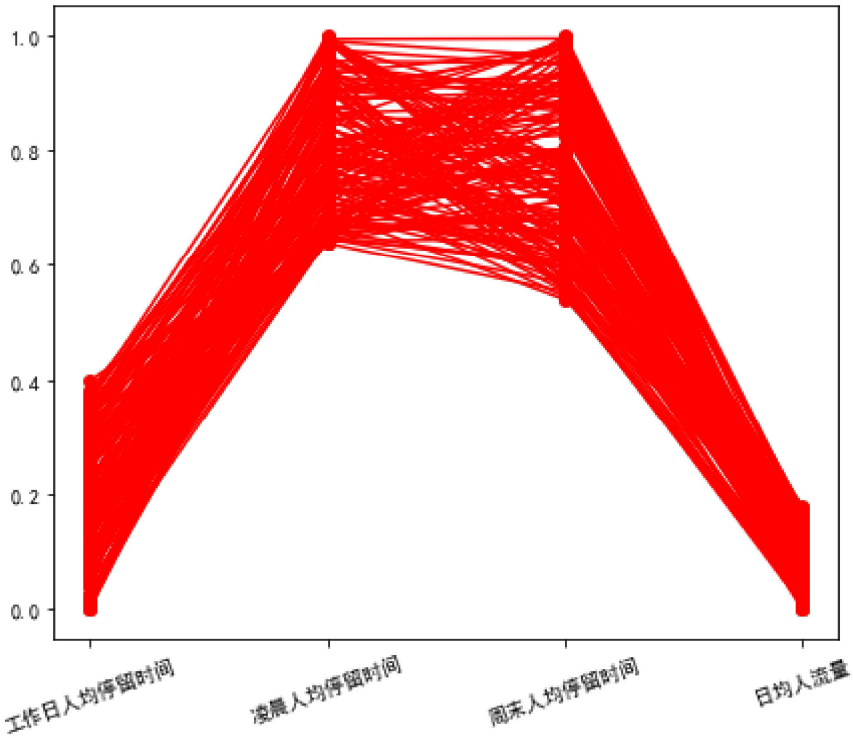
```
# 由谱系聚类图可知，聚类类别为3类
# 层次聚类算法

import pandas as pd

# 参数初始化
filename = 'standardization.xls'
data = pd.read_excel(filename, index_col = u'基站编号')
k = 3 # 聚类数

from sklearn.cluster import AgglomerativeClustering # 导入sklearn的层次聚类函数
model = AgglomerativeClustering(n_clusters = k, linkage = 'ward')

model.fit(data) # 训练模型
```

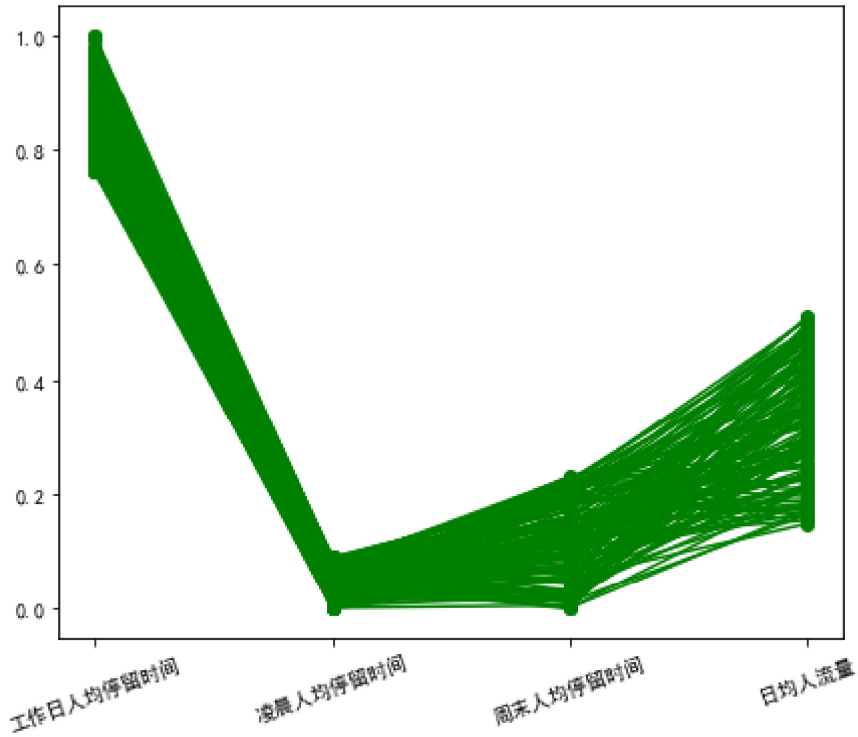
Out[18]:

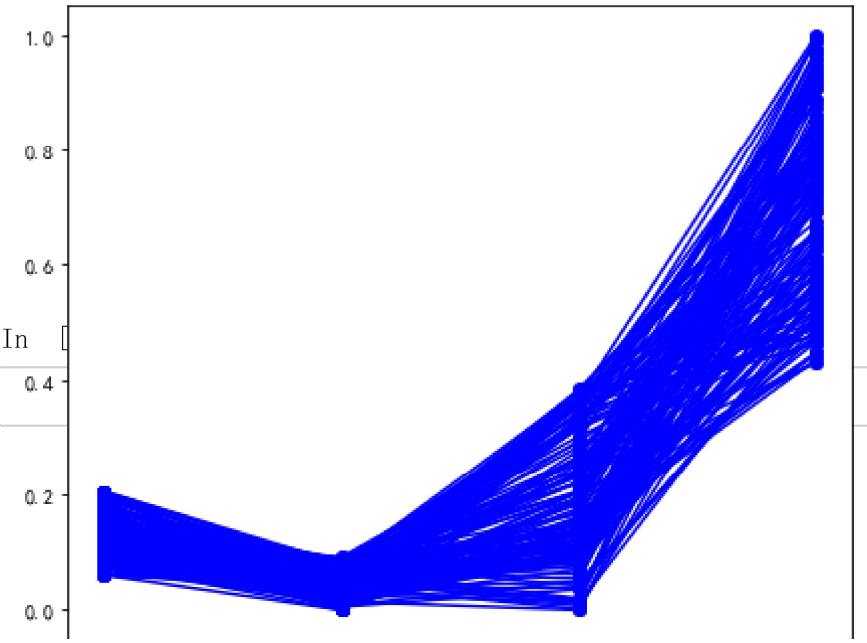
KMeans(n_clusters=3)

In [19]:

```
# 详细输出原始数据及其类别
r = pd.concat([data, pd.Series(model.labels_, index = data.index)], axis = 1) # 详细输出每个样本对
r.columns = list(data.columns) + [u'聚类类别'] # 重命名表名

import matplotlib.pyplot as plt
plt.rc('figure',figsize=(7,6))
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False # 用来正常显示负号

style = ['ro-', 'go-', 'bo-']
xlabels = [u'工作日人均停留时间', u'凌晨人均停留时间', u'周末人均停留时间', u'日均人流量']
pic_output = 'type_'

for i in range(k): # 逐一作图，作出不同样式
    plt.figure()
    tmp = r[r[u'聚类类别'] == i].iloc[:,:4] # 提取每一类
    for j in range(len(tmp)):
        plt.plot(range(1,5), tmp.iloc[j], style[i])

    plt.xticks(range(1,5), xlabels, rotation = 20) # 坐标标签 (***)
    plt.subplots_adjust(bottom=0.15) # 调整底部 (***)
    plt.savefig(u'%s%s.png' % (pic_output, i)) # 保存图片
```




1.0
0.8
0.6
0.4
0.2
0.0
工作日人均停留时间
凌晨人均停留时间
周末人均停留时间
日均人流量


1.0
0.8
0.6
0.4
0.2
0.0

In [ ]