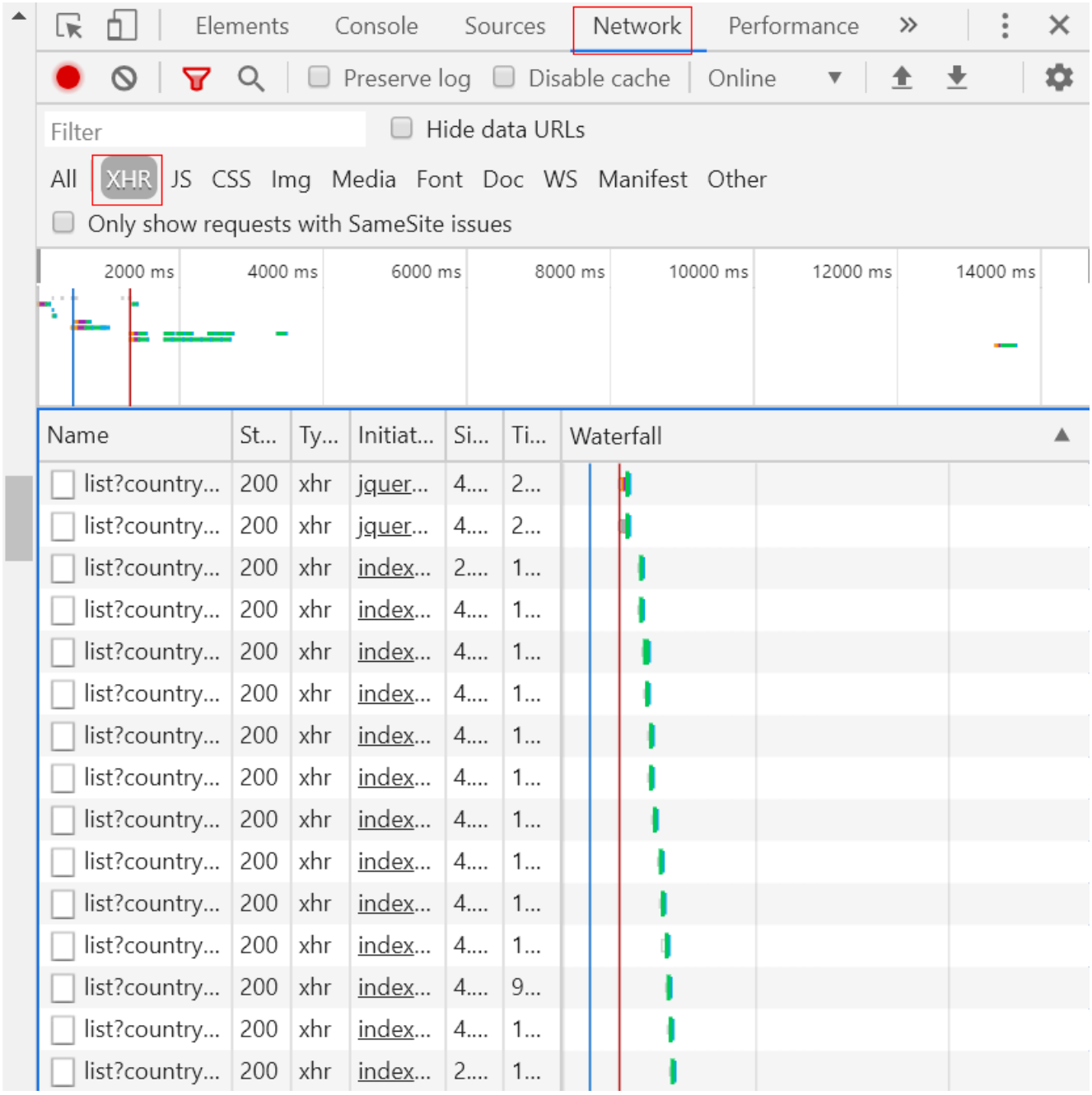
9 数据获取

使用Python获取新冠肺炎数据，并进行本地保存

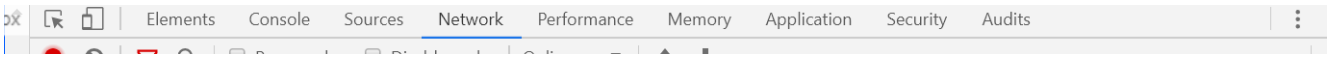
目标网站：https://news.qq.com/zt2020/page/feiyan.htm#/global 获取网站中意大利（其他国家同理）疫情感染人数

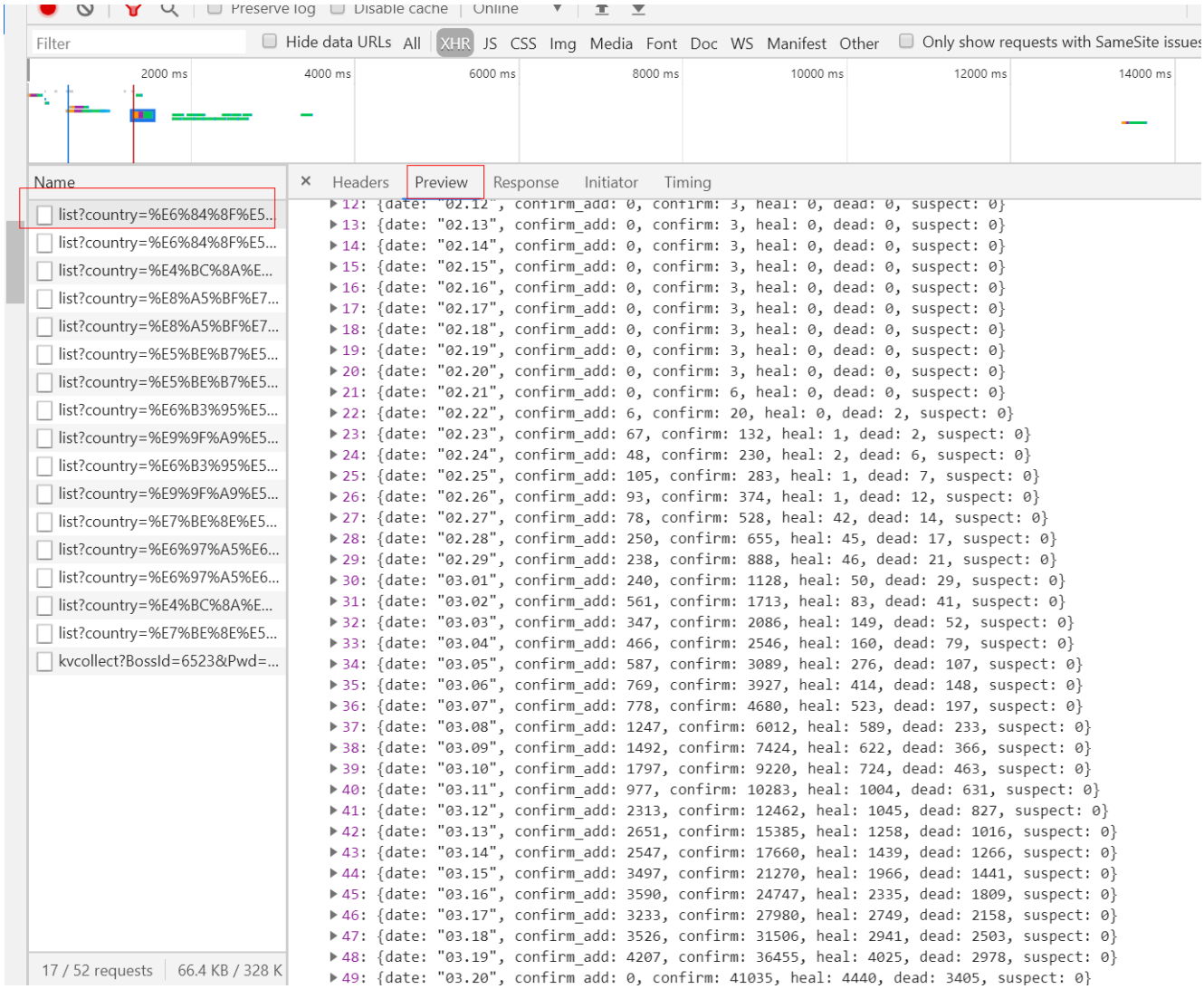
9.1 网站分析

网站中的数据加载通常有两种加载形式：动态与静态，打开浏览器的开发者平台（F12，推荐chrome浏览器），选择network中的XHR，查看网站中的数据加载项：

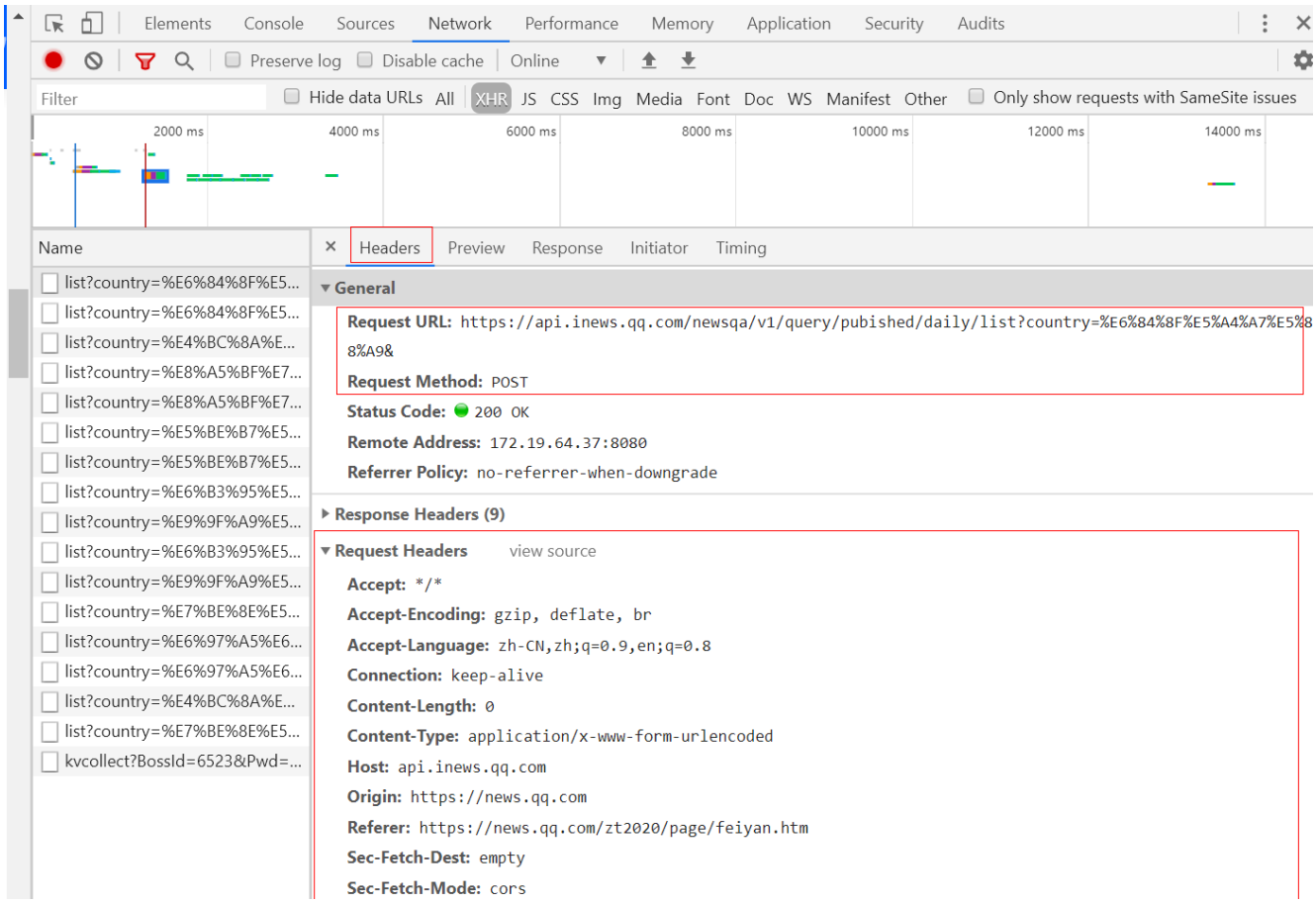


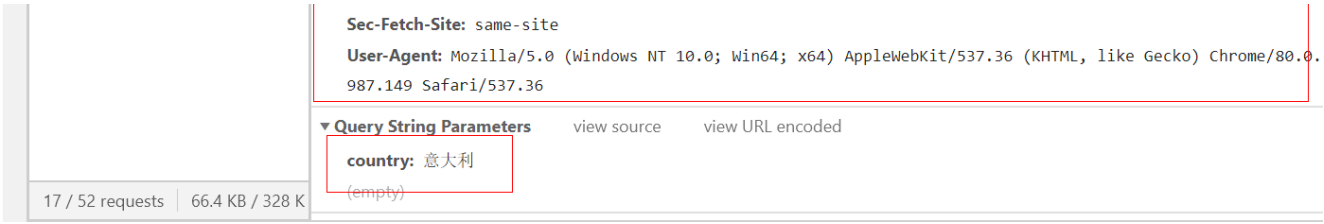
选择第一个加载项，打开可以发现文件中的信息为所需要获取的数据，通过与网页中的数据对比可以发现为意大利的数据。





选择headers查看数据项的具体信息。





http请求为POST请求，参数为country，对应的url为https://api.inews.qq.com/newsqa/v1/query/pubished/daily/list?country=%E6%84%8F%E5%A4%A7%E5%88%A9&

In [2]:

```
# 解析url
from urllib import parse #  urllib为Python中用于处理http请求，parse子模块用于处理url

# 使用unquote方法解析url
URL = "https://api.inews.qq.com/newsqa/v1/query/pubished/daily/list?country=%E6%84%8F%E5%A4%A7%E5%88%A9
&"

parse.unquote(URL)
```

Out[2]:

```
'https://api.inews.qq.com/newsqa/v1/query/pubished/daily/list?country=意大利&'
```

url最后类似于乱码形式的字符串为国家名转码以后的文本。如果需要获取其他国家的数据，只需要将url变为https://api.inews.qq.com/newsqa/v1/query/pubished/daily/list?country=国家名称&即可， 在拼接时无法使用中文，可以使用quote方法进行转码。

In [3]:

```
URL = "https://api.inews.qq.com/newsqa/v1/query/pubished/daily/list?country=%s&"
country = input("请输入查询国家：")
print(URL%(parse.quote(country)))
```

```
https://api.inews.qq.com/newsqa/v1/query/pubished/daily/list?country=%E6%84%8F%E5%A4%A7%E5%88%A9&
```

9.2 数据获取

使用requests模块模拟浏览器获取数据。

requests模块是Python中的http请求库，使用简单且功能强大。需要进行安装。

In [5]:

```
import requests


URL = "https://api.inews.qq.com/newsqa/v1/query/pubished/daily/list?country=%s&"

headers = {
    "Referer": "Referer: https://news.qq.com/zt2020/page/feiyan.html",
    "Host": "api.inews.qq.com",
    "Origin": "https://news.qq.com",
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chr
ome/80.0.3987.149 Safari/537.36"
} # 浏览器在发送请求时会携带一些信息，为了达到模拟浏览器的目的，爬虫也需要加入这些信息进行伪装，可以直接将浏览器中对应信息直接粘贴过来
```

requests.post(url,[data,] [headers,] [proxies,] *args)：使用post方式向目标发送请求。

-url：服务器对应的地址。

-data：请求时携带的数据。

-headers：HTTP请求中headers字段。

-proxies：设置代理IP地址

In [30]:

```
# 查看获取的内容
country = input("请输入查询国家：")
res = get_response(country)

print(res.text) # resposne.text 查看响应中的文本信息
```


{"ret":0,"info":"","data":[{"date":"01.31","confirm_add":2,"confirm":2,"heal":0,"dead":0,"suspect":0},{"date":"02.01","confirm_add":0,"confirm":2,"heal":0,"dead":0,"suspect":0},{"date":"02.02","confirm_add":0,"confirm":2,"heal":0,"dead":0,"suspect":0},{"date":"02.03","confirm_add":0,"confirm":2,"heal":0,"dead":0,"suspect":0},{"date":"02.04","confirm_add":0,"confirm":2,"heal":0,"dead":0,"suspect":0},{"date":"02.05","confirm_add":0,"confirm":2,"heal":0,"dead":0,"suspect":0},{"date":"02.06","confirm_add":0,"confirm":2,"heal":0,"dead":0,"suspect":0},{"date":"02.07","confirm_add":1,"confirm":3,"heal":0,"dead":0,"suspect":0},{"date":"02.08","confirm_add":0,"confirm":3,"heal":0,"dead":0,"suspect":0},{"date":"02.09","confirm_add":0,"confirm":3,"heal":0,"dead":0,"suspect":0},{"date":"02.10","confirm_add":0,"confirm":3,"heal":0,"dead":0,"suspect":0},{"date":"02.11","confirm_add":0,"confirm":3,"heal":0,"dead":0,"suspect":0},{"date":"02.12","confirm_add":0,"confirm":3,"heal":0,"dead":0,"suspect":0},{"date":"02.13","confirm_add":0,"confirm":3,"heal":0,"dead":0,"suspect":0},{"date":"02.14","confirm_add":0,"confirm":3,"heal":0,"dead":0,"suspect":0},{"date":"02.15","confirm_add":0,"confirm":3,"heal":0,"dead":0,"suspect":0},{"date":"02.16","confirm_add":0,"confirm":3,"heal":0,"dead":0,"suspect":0},{"date":"02.17","confirm_add":0,"confirm":3,"heal":0,"dead":0,"suspect":0},{"date":"02.18","confirm_add":0,"confirm":3,"heal":0,"dead":0,"suspect":0},{"date":"02.19","confirm_add":0,"confirm":3,"heal":0,"dead":0,"suspect":0},{"date":"02.20","confirm_add":0,"confirm":3,"heal":0,"dead":0,"suspect":0},{"date":"02.21","confirm_add":0,"confirm":6,"heal":0,"dead":0,"suspect":0},{"date":"02.22","confirm_add":6,"confirm":20,"heal":0,"dead":2,"suspect":0},{"date":"02.23","confirm_add":67,"confirm":132,"heal":1,"dead":2,"suspect":0},{"date":"02.24","confirm_add":48,"confirm":230,"heal":2,"dead":6,"suspect":0},{"date":"02.25","confirm_add":105,"confirm":283,"heal":1,"dead":7,"suspect":0},{"date":"02.26","confirm_add":93,"confirm":374,"heal":1,"dead":12,"suspect":0},{"date":"02.27","confirm_add":78,"confirm":528,"heal":42,"dead":14,"suspect":0},{"date":"02.28","confirm_add":250,"confirm":655,"heal":45,"dead":17,"suspect":0},{"date":"02.29","confirm_add":238,"confirm":888,"heal":46,"dead":21,"suspect":0},{"date":"03.01","confirm_add":240,"confirm":1128,"heal":50,"dead":29,"suspect":0},{"date":"03.02","confirm_add":561,"confirm":1713,"heal":83,"dead":41,"suspect":0},{"date":"03.03","confirm_add":347,"confirm":2086,"heal":149,"dead":52,"suspect":0},{"date":"03.04","confirm_add":466,"confirm":2546,"heal":160,"dead":79,"suspect":0},{"date":"03.05","confirm_add":587,"confirm":3089,"heal":276,"dead":107,"suspect":0},{"date":"03.06","confirm_add":769,"confirm":3927,"heal":414,"dead":148,"suspect":0},{"date":"03.07","confirm_add":778,"confirm":4680,"heal":523,"dead":197,"suspect":0},{"date":"03.08","confirm_add":1247,"confirm":6012,"heal":589,"dead":233,"suspect":0},{"date":"03.09","confirm_add":1492,"confirm":7424,"heal":622,"dead":366,"suspect":0},{"date":"03.10","confirm_add":1797,"confirm":9220,"heal":724,"dead":463,"suspect":0},{"date":"03.11","confirm_add":977,"confirm":10283,"heal":1004,"dead":631,"suspect":0},{"date":"03.12","confirm_add":2313,"confirm":12462,"heal":1045,"dead":827,"suspect":0},{"date":"03.13","confirm_add":2651,"confirm":15385,"heal":1258,"dead":1016,"suspect":0},{"date":"03.14","confirm_add":2547,"confirm":17660,"heal":1439,"dead":1266,"suspect":0},{"date":"03.15","confirm_add":3497,"confirm":21270,"heal":1966,"dead":1441,"suspect":0},{"date":"03.16","confirm_add":3590,"confirm":24747,"heal":2335,"dead":1809,"suspect":0},{"date":"03.17","confirm_add":3233,"confirm":27980,"heal":2749,"dead":2158,"suspect":0},{"date":"03.18","confirm_add":3526,"confirm":31506,"heal":2941,"dead":2503,"suspect":0},{"date":"03.19","confirm_add":4207,"confirm":36455,"heal":4025,"dead":2978,"suspect":0},{"date":"03.20","confirm_add":0,"confirm":41035,"heal":4440,"dead":3405,"suspect":0}]}

In [ ]:

```
def get_response(country):
    url = URL%(parse.quote(country))
    data = {
        "country": country,
    }
    response = reqeusts.post(url,data=data,headers=headers)
    return response
```

In [12]:

```
# 将数据写入本地
with open("yiqing.txt","a") as f:
    f.write(res.text)
```

虽然已经获取数据，但是数据本身并不直观，需要进行处理。通过前面的观察可以发现，数据时字符串形式的字典（json），我们将其转化为字典，然后使用pandas进行处理，最后保存至Excel文件中。

In [31]:

```
data = res.text
data = eval(data) # 转为字典
# 数据中只有data对应的键值对为数据
data = data["data"]
print(data)
```

```
[{'date': '01.31', 'confirm_add': 2, 'confirm': 2, 'heal': 0, 'dead': 0, 'suspect': 0}, {'date': '02.01
', 'confirm_add': 0, 'confirm': 2, 'heal': 0, 'dead': 0, 'suspect': 0}, {'date': '02.02', 'confirm_add'
: 0, 'confirm': 2, 'heal': 0, 'dead': 0, 'suspect': 0}, {'date': '02.03', 'confirm_add': 0, 'confirm':
2, 'heal': 0, 'dead': 0, 'suspect': 0}, {'date': '02.04', 'confirm_add': 0, 'confirm': 2, 'heal': 0, 'd
ead': 0, 'suspect': 0}, {'date': '02.05', 'confirm_add': 0, 'confirm': 2, 'heal': 0, 'dead': 0, 'suspec
t': 0}, {'date': '02.06', 'confirm_add': 0, 'confirm': 2, 'heal': 0, 'dead': 0, 'suspect': 0}, {'date':
'02.07', 'confirm_add': 1, 'confirm': 3, 'heal': 0, 'dead': 0, 'suspect': 0}, {'date': '02.08', 'confir
m_add': 0, 'confirm': 3, 'heal': 0, 'dead': 0, 'suspect': 0}, {'date': '02.09', 'confirm_add': 0, 'conf
irm': 3, 'heal': 0, 'dead': 0, 'suspect': 0}, {'date': '02.10', 'confirm_add': 0, 'confirm': 3, 'heal':
0, 'dead': 0, 'suspect': 0}, {'date': '02.11', 'confirm_add': 0, 'confirm': 3, 'heal': 0, 'dead': 0, 's
uspect': 0}, {'date': '02.12', 'confirm_add': 0, 'confirm': 3, 'heal': 0, 'dead': 0, 'suspect': 0}, {'d
ate': '02.13', 'confirm_add': 0, 'confirm': 3, 'heal': 0, 'dead': 0, 'suspect': 0}, {'date': '02.14', '
confirm_add': 0, 'confirm': 3, 'heal': 0, 'dead': 0, 'suspect': 0}, {'date': '02.15', 'confirm_add': 0,
'confirm': 3, 'heal': 0, 'dead': 0, 'suspect': 0}, {'date': '02.16', 'confirm_add': 0, 'confirm': 3, 'h
eal': 0, 'dead': 0, 'suspect': 0}, {'date': '02.17', 'confirm_add': 0, 'confirm': 3, 'heal': 0, 'dead':
0, 'suspect': 0}, {'date': '02.18', 'confirm_add': 0, 'confirm': 3, 'heal': 0, 'dead': 0, 'suspect': 0}
, {'date': '02.19', 'confirm_add': 0, 'confirm': 3, 'heal': 0, 'dead': 0, 'suspect': 0}, {'date': '02.2
0', 'confirm_add': 0, 'confirm': 3, 'heal': 0, 'dead': 0, 'suspect': 0}, {'date': '02.21', 'confirm_add
': 0, 'confirm': 6, 'heal': 0, 'dead': 0, 'suspect': 0}, {'date': '02.22', 'confirm_add': 6, 'confirm':
20, 'heal': 0, 'dead': 2, 'suspect': 0}, {'date': '02.23', 'confirm_add': 67, 'confirm': 132, 'heal': 1
, 'dead': 2, 'suspect': 0}, {'date': '02.24', 'confirm_add': 48, 'confirm': 230, 'heal': 2, 'dead': 6,
'suspect': 0}, {'date': '02.25', 'confirm_add': 105, 'confirm': 283, 'heal': 1, 'dead': 7, 'suspect': 0
}, {'date': '02.26', 'confirm_add': 93, 'confirm': 374, 'heal': 1, 'dead': 12, 'suspect': 0}, {'date':
'02.27', 'confirm_add': 78, 'confirm': 528, 'heal': 42, 'dead': 14, 'suspect': 0}, {'date': '02.28', 'c
onfirm_add': 250, 'confirm': 655, 'heal': 45, 'dead': 17, 'suspect': 0}, {'date': '02.29', 'confirm_add
': 238, 'confirm': 888, 'heal': 46, 'dead': 21, 'suspect': 0}, {'date': '03.01', 'confirm_add': 240, 'c
onfirm': 1128, 'heal': 50, 'dead': 29, 'suspect': 0}, {'date': '03.02', 'confirm_add': 561, 'confirm':
1713, 'heal': 83, 'dead': 41, 'suspect': 0}, {'date': '03.03', 'confirm_add': 347, 'confirm': 2086, 'he
al': 149, 'dead': 52, 'suspect': 0}, {'date': '03.04', 'confirm_add': 466, 'confirm': 2546, 'heal': 160
, 'dead': 79, 'suspect': 0}, {'date': '03.05', 'confirm_add': 587, 'confirm': 3089, 'heal': 276, 'dead'
: 107, 'suspect': 0}, {'date': '03.06', 'confirm_add': 769, 'confirm': 3927, 'heal': 414, 'dead': 148,
'suspect': 0}, {'date': '03.07', 'confirm_add': 778, 'confirm': 4680, 'heal': 523, 'dead': 197, 'suspec
t': 0}, {'date': '03.08', 'confirm_add': 1247, 'confirm': 6012, 'heal': 589, 'dead': 233, 'suspect': 0}
, {'date': '03.09', 'confirm_add': 1492, 'confirm': 7424, 'heal': 622, 'dead': 366, 'suspect': 0}, {'da
te': '03.10', 'confirm_add': 1797, 'confirm': 9220, 'heal': 724, 'dead': 463, 'suspect': 0}, {'date': '
03.11', 'confirm_add': 977, 'confirm': 10283, 'heal': 1004, 'dead': 631, 'suspect': 0}, {'date': '03.12
', 'confirm_add': 2313, 'confirm': 12462, 'heal': 1045, 'dead': 827, 'suspect': 0}, {'date': '03.13', '
confirm_add': 2651, 'confirm': 15385, 'heal': 1258, 'dead': 1016, 'suspect': 0}, {'date': '03.14', 'con
firm_add': 2547, 'confirm': 17660, 'heal': 1439, 'dead': 1266, 'suspect': 0}, {'date': '03.15', 'confir
m_add': 3497, 'confirm': 21270, 'heal': 1966, 'dead': 1441, 'suspect': 0}, {'date': '03.16', 'confirm_a
dd': 3590, 'confirm': 24747, 'heal': 2335, 'dead': 1809, 'suspect': 0}, {'date': '03.17', 'confirm_add'
: 3233, 'confirm': 27980, 'heal': 2749, 'dead': 2158, 'suspect': 0}, {'date': '03.18', 'confirm_add': 3
526, 'confirm': 31506, 'heal': 2941, 'dead': 2503, 'suspect': 0}, {'date': '03.19', 'confirm_add': 4207
, 'confirm': 36455, 'heal': 4025, 'dead': 2978, 'suspect': 0}, {'date': '03.20', 'confirm_add': 0, 'con
firm': 41035, 'heal': 4440, 'dead': 3405, 'suspect': 0}]
```

In [32]:

```
# 导入pandas ，需要进行安装
import pandas as pd # 使用as给包起别名

df = pd.DataFrame(data) # DataFrame为pandas中的一个数据结构，类似于二维表格
df
```

Out[32]:

	date	confirm_add	confirm	heal	dead	suspect
0	01.31	2	2	0	0	0

	date	confirm_add	confirm	heal	dead	suspect
1	02.01	0	2	0	0	0
2	02.02	0	2	0	0	0
3	02.03	0	2	0	0	0
4	02.04	0	2	0	0	0
5	02.05	0	2	0	0	0
6	02.06	0	2	0	0	0
7	02.07	1	3	0	0	0
8	02.08	0	3	0	0	0
9	02.09	0	3	0	0	0
10	02.10	0	3	0	0	0
11	02.11	0	3	0	0	0
12	02.12	0	3	0	0	0
13	02.13	0	3	0	0	0
14	02.14	0	3	0	0	0
15	02.15	0	3	0	0	0
16	02.16	0	3	0	0	0
17	02.17	0	3	0	0	0
18	02.18	0	3	0	0	0
19	02.19	0	3	0	0	0
20	02.20	0	3	0	0	0
21	02.21	0	6	0	0	0
22	02.22	6	20	0	2	0
23	02.23	67	132	1	2	0
24	02.24	48	230	2	6	0
25	02.25	105	283	1	7	0
26	02.26	93	374	1	12	0
27	02.27	78	528	42	14	0
28	02.28	250	655	45	17	0
29	02.29	238	888	46	21	0
30	03.01	240	1128	50	29	0
31	03.02	561	1713	83	41	0
32	03.03	347	2086	149	52	0
33	03.04	466	2546	160	79	0
34	03.05	587	3089	276	107	0
35	03.06	769	3927	414	148	0
36	03.07	778	4680	523	197	0
37	03.08	1247	6012	589	233	0
38	03.09	1492	7424	622	366	0
39	03.10	1797	9220	724	463	0
40	03.11	977	10283	1004	631	0
41	03.12	2313	12462	1045	827	0
42	03.13	2651	15385	1258	1016	0
43	03.14	2547	17660	1439	1266	0
44	03.15	3497	21270	1966	1441	0
45	03.16	3590	24747	2335	1809	0
46	03.17	3233	27980	2749	2158	0
47	03.18	3526	31506	2941	2503	0
48	03.19	4207	36455	4025	2978	0

	date	confirm_add	confirm	heal	dead	suspect
49	03.20	0	41035	4440	3405	0

此时的数据比较明了，但是数据中有两项存在问题：confirm_add和suspect，前者表示新增病例，后者表示疑似病例。

疑似病例中都为零，所以没有意义（实际中不可能都为0），需要删除。而新增病例中存在一个0（此处的0可能是因为当日没有结束，所以暂不统计），但是我们可以根据当前病例数减去昨日病例进行计算，所以此处也进行删除。

In [33]:

```
 # DataFrame中的drop方法可以通过指定标签名称和相应的轴，或直接指定索引或列名称来删除行或列。通过axis指定
df = df.drop(["confirm_add", "suspect"], axis=1)
```

In [34]:

```
df
```

Out[34]:

	date	confirm	heal	dead
0	01.31	2	0	0
1	02.01	2	0	0
2	02.02	2	0	0
3	02.03	2	0	0
4	02.04	2	0	0
5	02.05	2	0	0
6	02.06	2	0	0
7	02.07	3	0	0
8	02.08	3	0	0
9	02.09	3	0	0
10	02.10	3	0	0
11	02.11	3	0	0
12	02.12	3	0	0
13	02.13	3	0	0
14	02.14	3	0	0
15	02.15	3	0	0
16	02.16	3	0	0
17	02.17	3	0	0
18	02.18	3	0	0
19	02.19	3	0	0
20	02.20	3	0	0
21	02.21	6	0	0
22	02.22	20	0	2
23	02.23	132	1	2
24	02.24	230	2	6
25	02.25	283	1	7
26	02.26	374	1	12
27	02.27	528	42	14
28	02.28	655	45	17
29	02.29	888	46	21
30	03.01	1128	50	29
31	03.02	1713	83	41

	date	confirm	heal	dead
32	03.03	2086	149	52
33	03.04	2546	160	79
34	03.05	3089	276	107
35	03.06	3927	414	148
36	03.07	4680	523	197
37	03.08	6012	589	233
38	03.09	7424	622	366
39	03.10	9220	724	463
40	03.11	10283	1004	631
41	03.12	12462	1045	827
42	03.13	15385	1258	1016
43	03.14	17660	1439	1266
44	03.15	21270	1966	1441
45	03.16	24747	2335	1809
46	03.17	27980	2749	2158
47	03.18	31506	2941	2503
48	03.19	36455	4025	2978
49	03.20	41035	4440	3405

In [35]:

```
# 保存数据，写入本地Excel文件中
# python操作Excel文件需要安装环境依赖xlwt、xlrd工具包
df.to_excel("yiqing.xls", index=False)  # to_csv可以写入到csv文件中
```

In [36]:

```
# 读取文件，查看写入的数据
df = pd.read_excel("yiqing.xls", index_col=None)
df
```

Out[36]:

	date	confirm	heal	dead
0	1.31	2	0	0
1	2.01	2	0	0
2	2.02	2	0	0
3	2.03	2	0	0
4	2.04	2	0	0
5	2.05	2	0	0
6	2.06	2	0	0
7	2.07	3	0	0
8	2.08	3	0	0
9	2.09	3	0	0
10	2.10	3	0	0
11	2.11	3	0	0
12	2.12	3	0	0
13	2.13	3	0	0
14	2.14	3	0	0
15	2.15	3	0	0
16	2.16	3	0	0

	date	confirm	heal	dead
17	2.17	3	0	0
18	2.18	3	0	0
19	2.19	3	0	0
20	2.20	3	0	0
21	2.21	6	0	0
22	2.22	20	0	2
23	2.23	132	1	2
24	2.24	230	2	6
25	2.25	283	1	7
26	2.26	374	1	12
27	2.27	528	42	14
28	2.28	655	45	17
29	2.29	888	46	21
30	3.01	1128	50	29
31	3.02	1713	83	41
32	3.03	2086	149	52
33	3.04	2546	160	79
34	3.05	3089	276	107
35	3.06	3927	414	148
36	3.07	4680	523	197
37	3.08	6012	589	233
38	3.09	7424	622	366
39	3.10	9220	724	463
40	3.11	10283	1004	631
41	3.12	12462	1045	827
42	3.13	15385	1258	1016
43	3.14	17660	1439	1266
44	3.15	21270	1966	1441
45	3.16	24747	2335	1809
46	3.17	27980	2749	2158
47	3.18	31506	2941	2503
48	3.19	36455	4025	2978
49	3.20	41035	4440	3405

In [ ]: