# 引入配置  
# 基础  
import numpy as np  
import pandas as pd  
from pandas import DataFrame, Series  
from datetime import datetime  
import json  
import warnings  
warnings.filterwarnings('ignore')# 忽略python运行过程中的警告  
  
# 可视化  
import matplotlib.pyplot as plt  
import seaborn as sns  
from wordcloud import WordCloud,STOPWORDS,ImageColorGenerator # 导入词云包  
%matplotlib inline

# 题目1：根据新冠肺炎数据，绘图

# 读取数据集  
url ='/Users/bennyrhys/Desktop/数据分析可视化/homework/2020冠状肺炎统计.xlsx'  
df = pd.read\_excel(url, index\_col='日期')  
df.head()

.dataframe tbody tr th:only-of-type {
vertical-align: middle;
}</body>

.dataframe tbody tr th {  
 vertical-align: top;  
}  
  
.dataframe thead th {  
 text-align: right;  
}

累计确诊
疑似
新增疑似
死亡
治愈
现有确诊
境外输入
现有境外输入确诊
日期
2020-01-11
41.0
0.0
NaN
1.0
2.0
38.0
NaN
NaN
2020-01-12
41.0
0.0
NaN
1.0
7.0
33.0
NaN
NaN
2020-01-13
41.0
0.0
NaN
1.0
7.0
33.0
NaN
NaN
2020-01-14
41.0
0.0
NaN
1.0
7.0
33.0
NaN
NaN
2020-01-15
41.0
0.0
NaN
2.0
12.0
27.0
NaN
NaN

df['新增疑似'].isnull()

日期  
2020-01-11 True  
2020-01-12 True  
2020-01-13 True  
2020-01-14 True  
2020-01-15 True  
 ...   
2020-04-27 False  
2020-04-28 False  
2020-04-29 False  
2020-04-30 True  
2020-05-01 True  
Name: 新增疑似, Length: 112, dtype: bool

df.fillna(0)

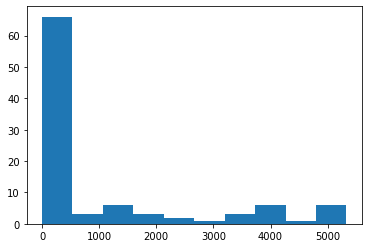
.dataframe tbody tr th:only-of-type {
vertical-align: middle;
}</body>

.dataframe tbody tr th {  
 vertical-align: top;  
}  
  
.dataframe thead th {  
 text-align: right;  
}

累计确诊
疑似
新增疑似
死亡
治愈
现有确诊
境外输入
现有境外输入确诊
日期
2020-01-11
41.0
0.0
0.0
1.0
2.0
38.0
0.0
0.0
2020-01-12
41.0
0.0
0.0
1.0
7.0
33.0
0.0
0.0
2020-01-13
41.0
0.0
0.0
1.0
7.0
33.0
0.0
0.0
2020-01-14
41.0
0.0
0.0
1.0
7.0
33.0
0.0
0.0
2020-01-15
41.0
0.0
0.0
2.0
12.0
27.0
0.0
0.0
...
...
...
...
...
...
...
...
...
2020-04-27
82836.0
9.0
1.0
4633.0
77555.0
648.0
1639.0
552.0
2020-04-28
82858.0
10.0
2.0
4633.0
77578.0
647.0
1660.0
553.0
2020-04-29
82862.0
10.0
3.0
4633.0
77610.0
619.0
1664.0
525.0
2020-04-30
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
2020-05-01
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0

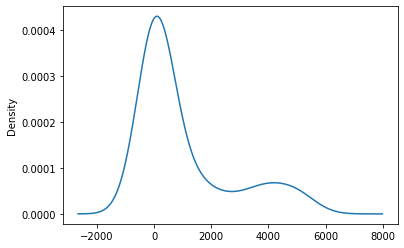
# 直方图  
plt.hist(df['新增疑似'])

(array([66., 3., 6., 3., 2., 1., 3., 6., 1., 6.]),  
 array([ 0. , 532.8, 1065.6, 1598.4, 2131.2, 2664. , 3196.8, 3729.6,  
 4262.4, 4795.2, 5328. ]),  
 <a list of 10 Patch objects>)



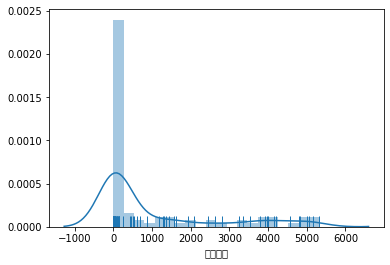
# 密度图 Series 的方法直接画.plot  
df['新增疑似'].plot(kind='kde')

<matplotlib.axes.\_subplots.AxesSubplot at 0x7fc5b6411490>



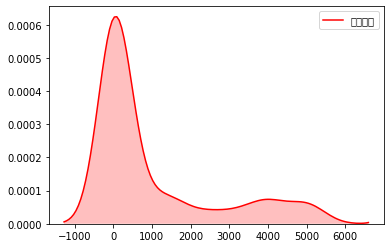
# 直方图，密度图.distplot()  
# 参数 数据，分块，是否直方图，是否密度图,rug分布情况  
sns.distplot(df['新增疑似'], bins=20, hist=True, kde=True, rug=True)

<matplotlib.axes.\_subplots.AxesSubplot at 0x7fc5b651b490>



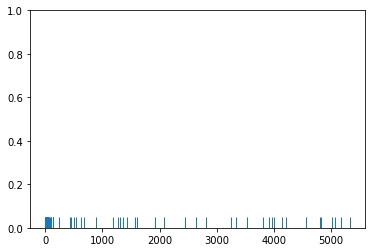
# 密度图  
# 参数 数据，颜色填充, 颜色  
sns.kdeplot(df['新增疑似'], shade=True, color='r')

<matplotlib.axes.\_subplots.AxesSubplot at 0x7fc5b64183d0>



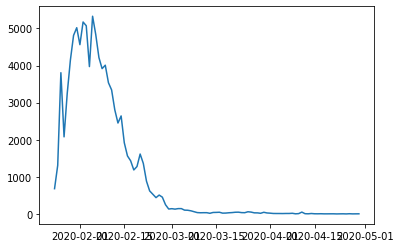
sns.rugplot(df['新增疑似'])

<matplotlib.axes.\_subplots.AxesSubplot at 0x7fc5b71c7ad0>



plt.plot(df['新增疑似'])

[<matplotlib.lines.Line2D at 0x7fc5b722d150>]



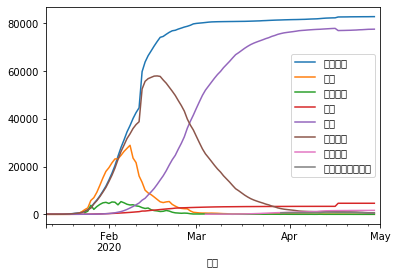
# 热力图(新增疑似)  
sns.heatmap(df)

<matplotlib.axes.\_subplots.AxesSubplot at 0x7fc5b74c74d0>



# df普通线性  
df.plot()

<matplotlib.axes.\_subplots.AxesSubplot at 0x7fc5b75b0510>



data = pd.read\_excel(url)  
data.head()

.dataframe tbody tr th:only-of-type {
vertical-align: middle;
}</body>

.dataframe tbody tr th {  
 vertical-align: top;  
}  
  
.dataframe thead th {  
 text-align: right;  
}

日期
累计确诊
疑似
新增疑似
死亡
治愈
现有确诊
境外输入
现有境外输入确诊
0
2020-01-11
41.0
0.0
NaN
1.0
2.0
38.0
NaN
NaN
1
2020-01-12
41.0
0.0
NaN
1.0
7.0
33.0
NaN
NaN
2
2020-01-13
41.0
0.0
NaN
1.0
7.0
33.0
NaN
NaN
3
2020-01-14
41.0
0.0
NaN
1.0
7.0
33.0
NaN
NaN
4
2020-01-15
41.0
0.0
NaN
2.0
12.0
27.0
NaN
NaN

fig = plt.figure(figsize=(10,8))  
ax = fig.add\_subplot(1,1,1)  
x = data['日期']  
plt.plot(x,df['境外输入'],lw=2,label="境外输入",color = "r")  
ax.set\_xticklabels(x,rotation = 30,fontsize='medium')  
plt.legend()

<matplotlib.legend.Legend at 0x7fc5ba5c7a10>

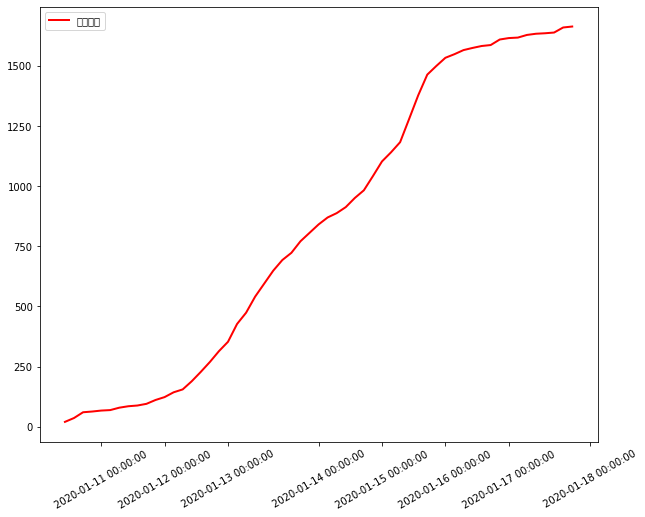
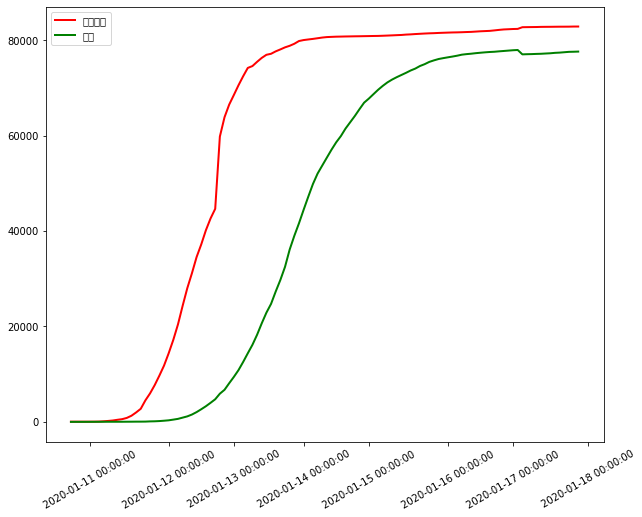


fig = plt.figure(figsize=(10,8))  
ax = fig.add\_subplot(1,1,1)  
x = data['日期']  
plt.plot(x,data['累计确诊'],lw=2,label="累计确诊",color = "r")  
plt.plot(x,data['治愈'],lw=2,label="治愈",color = "g")  
ax.set\_xticklabels(x,rotation = 30,fontsize='medium')  
plt.legend()

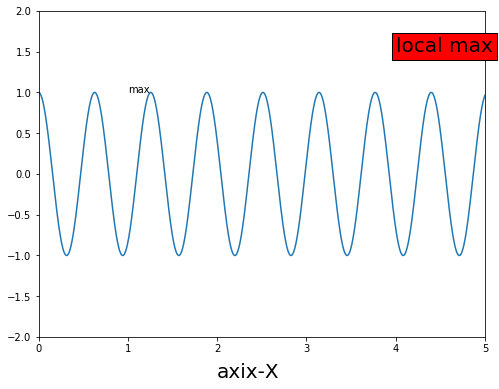
<matplotlib.legend.Legend at 0x7fc5ba8bd4d0>



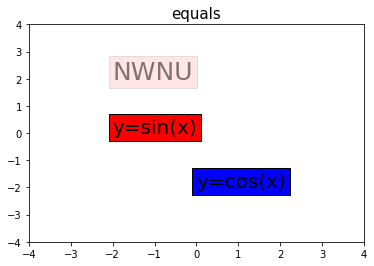
# 题目2：绘图(写出绘制下图的代码)

fig = plt.figure(figsize=(8,6))  
x = np.linspace(0,20,50000)  
y = np.cos(x\*10)  
plt.subplot(1,1,1)  
plt.xlim([0,5])  
plt.ylim([-2,2])  
plt.text(1, 1,r'max')  
plt.text(4,1.5,r'local max',size=20,bbox=dict(facecolor='red'))  
plt.text(2,-2.5,r'axix-X',size=20)  
  
plt.plot(x,y)

[<matplotlib.lines.Line2D at 0x7fc5ba9abe50>]



# 公式  
plt.title('equals',fontsize=15)  
plt.xlim([-4,4])  
plt.ylim([-4,4])  
plt.text(-2,2,r'NWNU',alpha=0.5,size=25,bbox=dict(facecolor='red', alpha=0.1))  
plt.text(-2,0,r'y=sin(x)',size=20,bbox=dict(facecolor='red'))  
plt.text(0,-2,r'y=cos(x)',size=20,bbox=dict(facecolor='BLUE'))  
  
plt.show()



# 题目3：导入survey数据集，设计分析的内容并加以实现。

#读数据  
survey = pd.read\_csv("/Users/bennyrhys/Desktop/数据分析可视化/homework/survey.csv")  
survey.head()

.dataframe tbody tr th:only-of-type {
vertical-align: middle;
}</body>

.dataframe tbody tr th {  
 vertical-align: top;  
}  
  
.dataframe thead th {  
 text-align: right;  
}

Timestamp
Age
Gender
Country
state
self\_employed
family\_history
treatment
work\_interfere
no\_employees
...
leave
mental\_health\_consequence
phys\_health\_consequence
coworkers
supervisor
mental\_health\_interview
phys\_health\_interview
mental\_vs\_physical
obs\_consequence
comments
0
2014-08-27 11:29:31
37
Female
United States
IL
NaN
No
Yes
Often
6-25
...
Somewhat easy
No
No
Some of them
Yes
No
Maybe
Yes
No
NaN
1
2014-08-27 11:29:37
44
M
United States
IN
NaN
No
No
Rarely
More than 1000
...
Don't know
Maybe
No
No
No
No
No
Don't know
No
NaN
2
2014-08-27 11:29:44
32
Male
Canada
NaN
NaN
No
No
Rarely
6-25
...
Somewhat difficult
No
No
Yes
Yes
Yes
Yes
No
No
NaN
3
2014-08-27 11:29:46
31
Male
United Kingdom
NaN
NaN
Yes
Yes
Often
26-100
...
Somewhat difficult
Yes
Yes
Some of them
No
Maybe
Maybe
No
Yes
NaN
4
2014-08-27 11:30:22
31
Male
United States
TX
NaN
No
No
Never
100-500
...
Don't know
No
No
Some of them
Yes
Yes
Yes
Don't know
No
NaN

survey.info()

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 1259 entries, 0 to 1258  
Data columns (total 27 columns):  
Timestamp 1259 non-null object  
Age 1259 non-null int64  
Gender 1259 non-null object  
Country 1259 non-null object  
state 744 non-null object  
self\_employed 1241 non-null object  
family\_history 1259 non-null object  
treatment 1259 non-null object  
work\_interfere 995 non-null object  
no\_employees 1259 non-null object  
remote\_work 1259 non-null object  
tech\_company 1259 non-null object  
benefits 1259 non-null object  
care\_options 1259 non-null object  
wellness\_program 1259 non-null object  
seek\_help 1259 non-null object  
anonymity 1259 non-null object  
leave 1259 non-null object  
mental\_health\_consequence 1259 non-null object  
phys\_health\_consequence 1259 non-null object  
coworkers 1259 non-null object  
supervisor 1259 non-null object  
mental\_health\_interview 1259 non-null object  
phys\_health\_interview 1259 non-null object  
mental\_vs\_physical 1259 non-null object  
obs\_consequence 1259 non-null object  
comments 164 non-null object  
dtypes: int64(1), object(26)  
memory usage: 265.7+ KB

survey.isnull().sum()

Timestamp 0  
Age 0  
Gender 0  
Country 0  
state 515  
self\_employed 18  
family\_history 0  
treatment 0  
work\_interfere 264  
no\_employees 0  
remote\_work 0  
tech\_company 0  
benefits 0  
care\_options 0  
wellness\_program 0  
seek\_help 0  
anonymity 0  
leave 0  
mental\_health\_consequence 0  
phys\_health\_consequence 0  
coworkers 0  
supervisor 0  
mental\_health\_interview 0  
phys\_health\_interview 0  
mental\_vs\_physical 0  
obs\_consequence 0  
comments 1095  
dtype: int64

survey['self\_employed'].value\_counts()

No 1095  
Yes 146  
Name: self\_employed, dtype: int64

survey['self\_employed'] = survey['self\_employed'].fillna(value='No',inplace=True)

df3 = survey[['Age','mental\_health\_consequence']].groupby(['Age']).count()  
df3

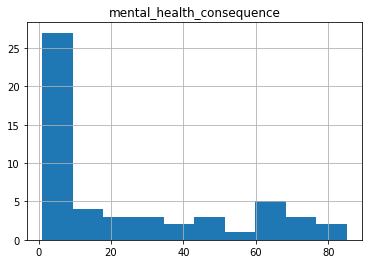
.dataframe tbody tr th:only-of-type {
vertical-align: middle;
}</body>

.dataframe tbody tr th {  
 vertical-align: top;  
}  
  
.dataframe thead th {  
 text-align: right;  
}

mental\_health\_consequence
Age
-1726
1
-29
1
-1
1
5
1
8
1
11
1
18
7
19
9
20
6
21
16
22
21
23
51
24
46
25
61
26
75
27
71
28
68
29
85
30
63
31
67
32
82
33
70
34
65
35
55
36
37
37
43
38
39
39
33
40
33
41
21
42
20
43
28
44
11
45
12
46
12
47
2
48
6
49
4
50
6
51
5
53
1
54
3
55
3
56
4
57
3
58
1
60
2
61
1
62
1
65
1
72
1
329
1
99999999999
1

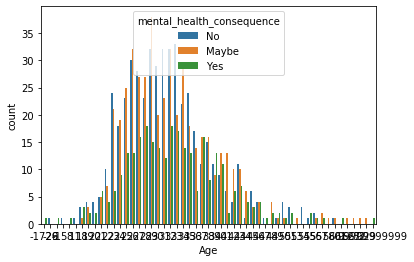
df3.hist()

array([[<matplotlib.axes.\_subplots.AxesSubplot object at 0x7fc5ba58bb50>]],  
 dtype=object)



sns.countplot(x='Age',hue='mental\_health\_consequence',data=survey)

<matplotlib.axes.\_subplots.AxesSubplot at 0x7fc5ba56cb50>

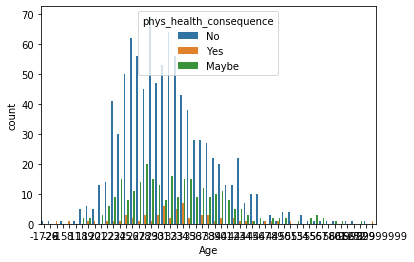


df4 = survey['Country'].groupby(survey['Country']).count()  
df4

Country  
Australia 21  
Austria 3  
Bahamas, The 1  
Belgium 6  
Bosnia and Herzegovina 1  
Brazil 6  
Bulgaria 4  
Canada 72  
China 1  
Colombia 2  
Costa Rica 1  
Croatia 2  
Czech Republic 1  
Denmark 2  
Finland 3  
France 13  
Georgia 1  
Germany 45  
Greece 2  
Hungary 1  
India 10  
Ireland 27  
Israel 5  
Italy 7  
Japan 1  
Latvia 1  
Mexico 3  
Moldova 1  
Netherlands 27  
New Zealand 8  
Nigeria 1  
Norway 1  
Philippines 1  
Poland 7  
Portugal 2  
Romania 1  
Russia 3  
Singapore 4  
Slovenia 1  
South Africa 6  
Spain 1  
Sweden 7  
Switzerland 7  
Thailand 1  
United Kingdom 185  
United States 751  
Uruguay 1  
Zimbabwe 1  
Name: Country, dtype: int64

sns.countplot(x='Age',hue='phys\_health\_consequence',data=survey)

<matplotlib.axes.\_subplots.AxesSubplot at 0x7fc5ba43da90>



df5 = survey[['Age','phys\_health\_consequence']].groupby(['Age']).count()  
df5

.dataframe tbody tr th:only-of-type {
vertical-align: middle;
}</body>

.dataframe tbody tr th {  
 vertical-align: top;  
}  
  
.dataframe thead th {  
 text-align: right;  
}

phys\_health\_consequence
Age
-1726
1
-29
1
-1
1
5
1
8
1
11
1
18
7
19
9
20
6
21
16
22
21
23
51
24
46
25
61
26
75
27
71
28
68
29
85
30
63
31
67
32
82
33
70
34
65
35
55
36
37
37
43
38
39
39
33
40
33
41
21
42
20
43
28
44
11
45
12
46
12
47
2
48
6
49
4
50
6
51
5
53
1
54
3
55
3
56
4
57
3
58
1
60
2
61
1
62
1
65
1
72
1
329
1
99999999999
1

# 热力图  
sns.heatmap(df5)

<matplotlib.axes.\_subplots.AxesSubplot at 0x7fc5be0fd710>

