爬虫数据的预处理，数据分析及重点新闻评论筛选

作者：汪喆191250134

实验环境：python3.7，xlrd模块，scipy模块，pandas模块，numpy模块，matplotlib.pyplot模块，math模块，xlsxwriter模块，pycharm，WPS的excel

实验目标：从爬虫数据中筛选出与疫情有关的新闻，并对新闻评论数作数据分析与可视化，再以数据分析结果为依据筛选出疫情相关重点新闻评论。

实验步骤：

一、爬虫数据预处理

所获的爬虫数据是四个Excel表格，第一列为新闻摘要，第二列为新闻评论数，第三至若干列为新闻评论。



Step1:只关注excel表格的第一列即可。注意到新闻摘要普遍开头第一个自符为【，则将其作为后续程序的判断依据，与疫情相关的新闻其摘要以【开头，而与疫情无关新闻的摘要以其他自符开头。故删去疫情无关新闻摘要开头的【，为不是【开头的疫情相关新闻摘要加上【。

1. 疫情相关数据的读入

以第一阶段为例，另外三阶段同理。

Step1:遍历excel表的每一行，首先判断每行第一列单元格的首字符是否为【。若是，则进一步读取该行数据，否则跳过该行。

Step2:对于疫情相关新闻（行首单元格首字符为【），再遍历行，将该行的第二个单元格即评论数以及之后的评论不超过70条存入小列表thisnews中，若所得的thisnews列表合法，则将其添加到大列表relatednews（疫情相关新闻中。

Step3:将所有新闻的评论数存入列表numOfcomments，作为后续数据分析的样本。

所存储的数据结构为：列表relatednews中元素也为列表，即若干列表thisnews的数据。列表thisnews中第1个元素为对应新闻的评论数（int），第2～71个元素为对应新闻的70条合法评论（string）。

代码如下：

workbook**=**xlrd.open\_workbook**(r'/Users/sebas/第一阶段（12.8~1.22）before.xls')**sheet\_name **=** workbook.sheet\_names**()[**0**]**sheet **=** workbook.sheet\_by\_index**(**0**)**relatednews**=[]  
for** i **in** range**(**0,len**(**sheet.col\_values**(**0**))):** #i是遍历每行  
 **if** sheet.cell\_value**(**i,0**)[**0**]=='【':** thisnews**=[]** #每个疫情新闻存到一个列表里，第一项评论数，后面70条评论  
 commentnumOfthisnews**=**int**(**sheet.cell\_value**(**i,1**))** thisnews.append**(**commentnumOfthisnews**)** j**=**2  
 **while** j**<**len**(**sheet.row\_values**(**i**)) and** len**(**thisnews**)<**71**:** #加够1+70即可  
 **if(**sheet.cell\_value**(**i,j**)=="我是只大大龙"):** #若非法，则不添加  
 j**+=**1  
 **continue** thisnews.append**(**sheet.cell\_value**(**i,j**))** j**+=**1  
 **if(**len**(**thisnews**)==**71**):** #1+70，则加到relatednews里  
 relatednews.append**(**thisnews**)**

**#获得评论数数组**

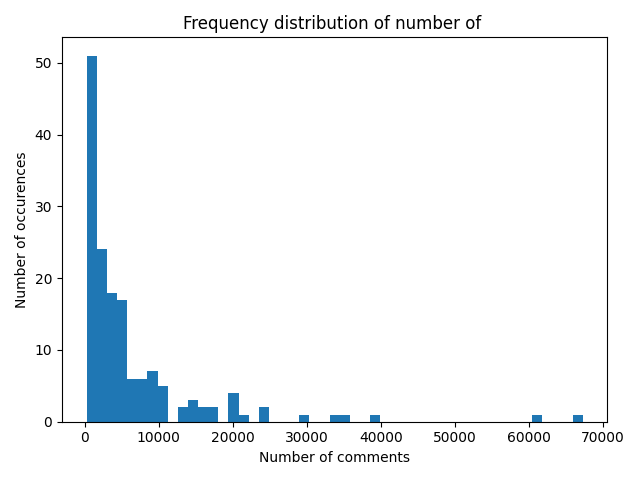
numOfcomments**=[]**   
**for** i **in** range**(**0,len**(**relatednews**)):** numOfcomments.append**(**relatednews**[**i**][**0**])**

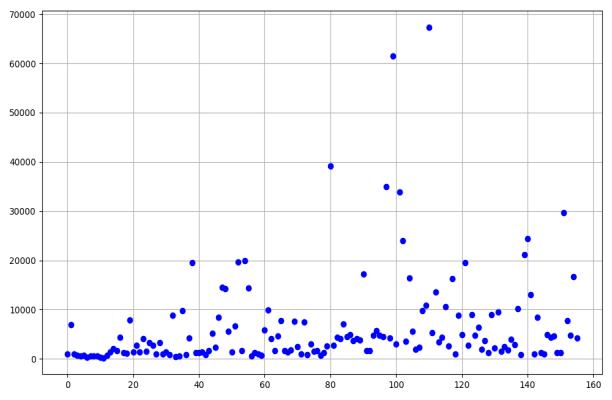
1. 数据处理以及可视化

由于四个阶段所获得的评论数数据存在差异，采用了一些不同的数据处理方法，遂分为以下四个阶段进行描述。

第一阶段（12.8～1.22）：

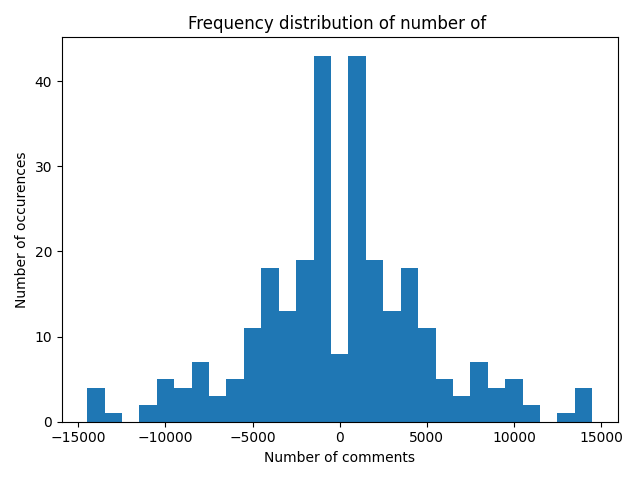
共爬取新闻938条，其中疫情相关新闻160条，占17%，去除格式异常的新闻后，剩余156条，即样本容量为156.

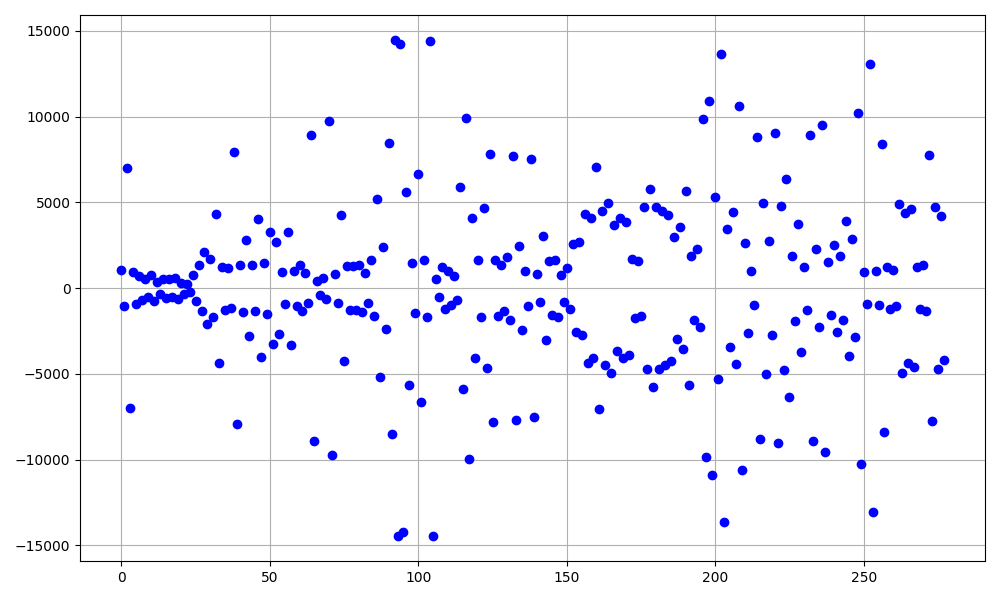
Step1:对总体评论数进行可视化



由图中可以看出，若去掉极端数据，其分布近似服从半正态分布，下面进行检验。

Step2:去掉极端数据，补全y轴左半边数据，并对近似正态数据进行可视化

若评论数大于15000，则去掉，最终所得符合条件的新闻数为139.



代码如下：

**for** i **in** range**(**0,len**(**numOfcomments**)):** # 去掉过于极端数据,156-=139求得p值=0.09>0.05有95%的可信度为正态分布  
 **if(**numOfcomments**[**i**]>**15000**):  
 continue** normdata.append**(**numOfcomments**[**i**])** normdata.append**(-**numOfcomments**[**i**])** numofndata**+=**1  
print**(**numofndata**)**

visual**(**normdata,30**)**kscheck**(**normdata**)**normfit**(**normdata**)**

Step3：进行正态分布ks检验以及频率分布直方图与正态拟合曲线可视化

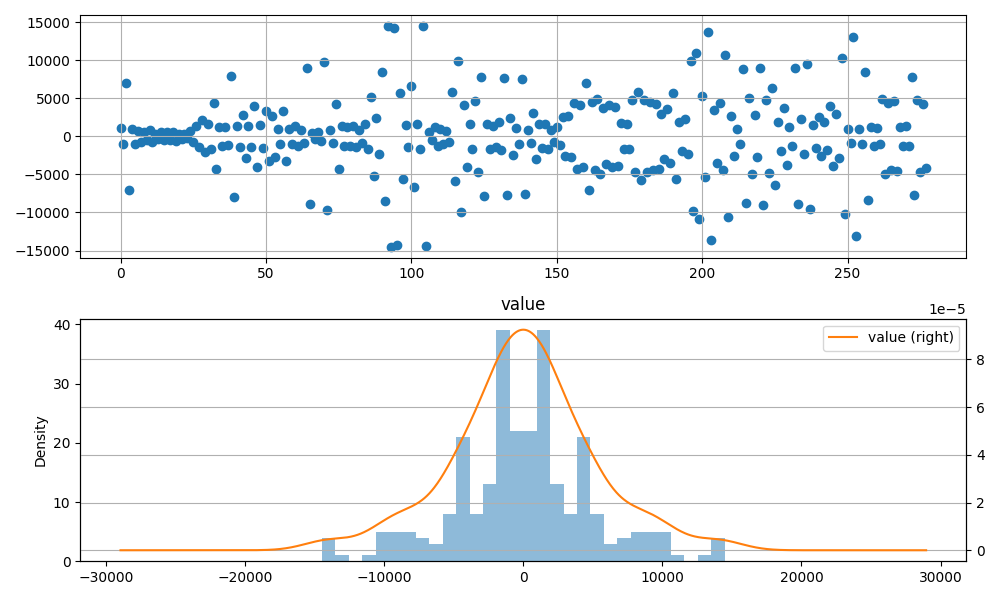
由于卡方拟合优度检验主要用于类别数据，且需要对数据进行分组，而ks检验能直接对原始数据进行检验，所以它对数据的利用比较完整，故采用ks检验，ks检验也是通过计算理论频数与实际频数的差值来判断某一组数据是否近似符合某一分布，属于非参数假设检验。若p值>0.05（显著性水平）,则可以认为样本服从正态分布。

结果分别为：近似正态数据的均值，标准差，ks检验结果

0.0 5073.570405962642

KstestResult(statistic=0.07410443750871726, pvalue=0.08972989524283492)

p值>0.05，故近似服从正态分布。拟合图像为：

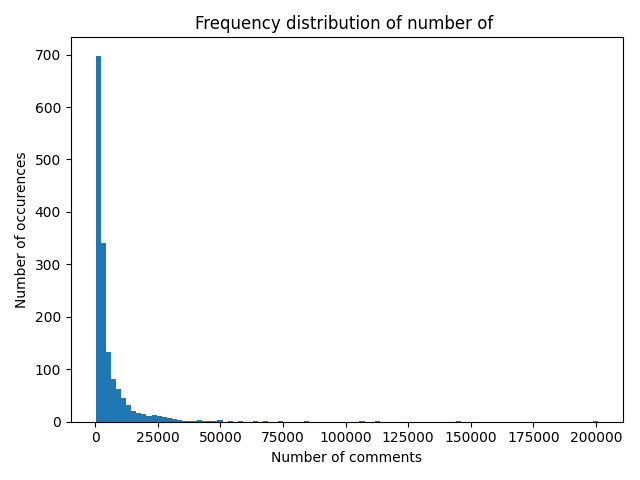


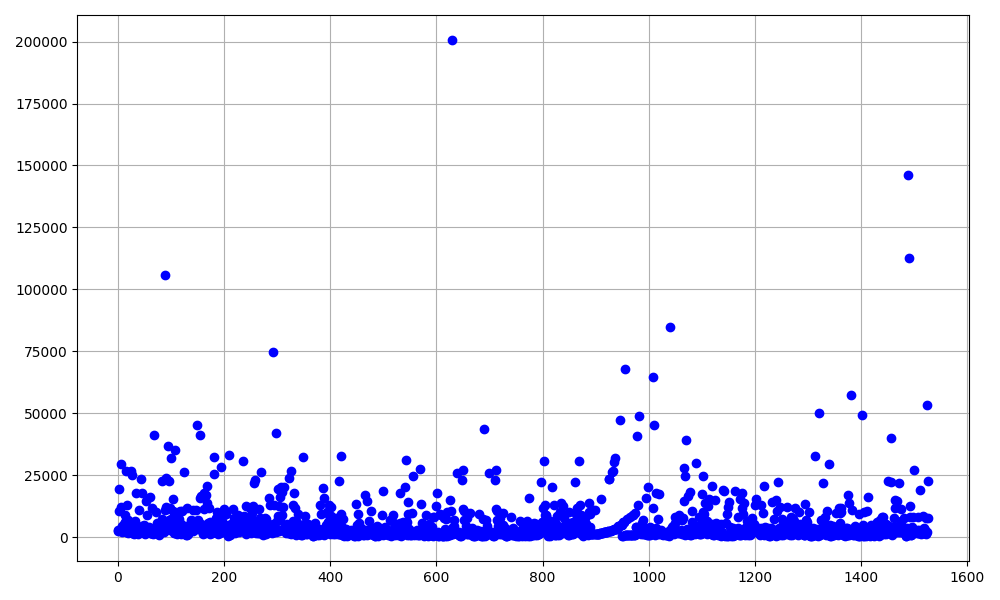
将此正态分布标准化，再查得标准正态分布下的**上alpha分位点**（alpha=0.40）=0.26，由Z=(X-u)/sigma得，相应的**评论数分位点**为1483，即评论数大于1483的为重点疫情新闻。

第二阶段：

共爬取新闻1836条，其中疫情相关新闻1556条，占85%，去除格式异常的新闻后，剩余1528条，即样本容量为1528.

Step1:对总体评论数进行可视化以及ks正态检验





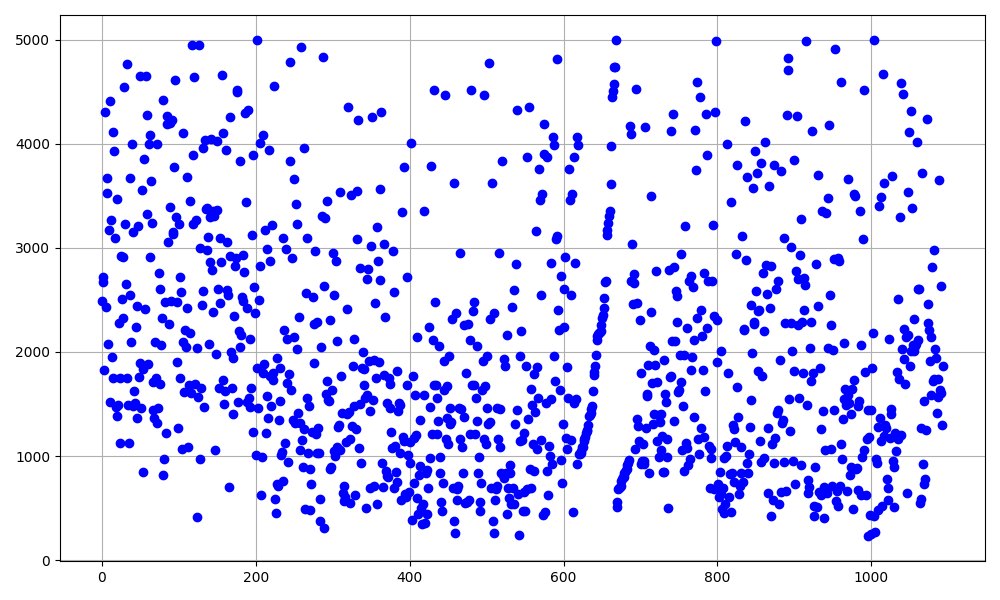
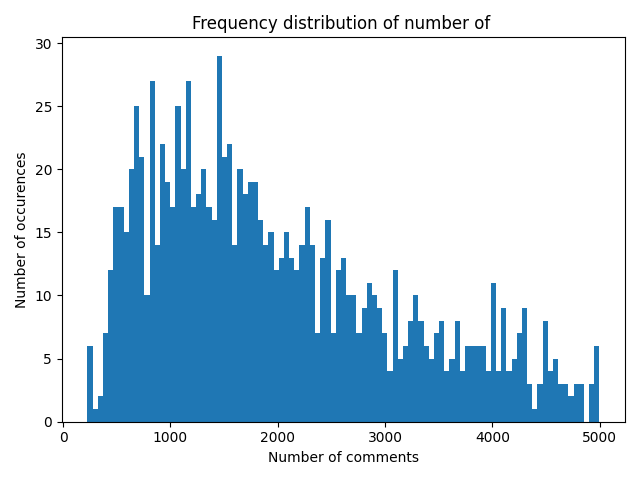
初始样本均值，标准差，ks检验d值，p值：

5723.558900523561 10733.086274517093

KstestResult(statistic=0.3043183735662171, pvalue=4.509114967667845e-126)

可以看出，p值几近趋于0了，这说明初始样本数据几乎不可能服从正态分布，下面对一些极端的数据进行删除。我们知道，正态分布应具有3sigma原则，出现过大或者过小的样本值都是几乎不可能的。因此，去掉那些评论数过大的数据，来保证能够用最小二乘法拟合出正态分布的未知参数。

Step2:去掉极端数据，再进行ks正态检验与可视化，保证一定的可信度。



剩余样本均值，标准差，ks检验d值，p值：

2014.573126142596 1157.9450863929048

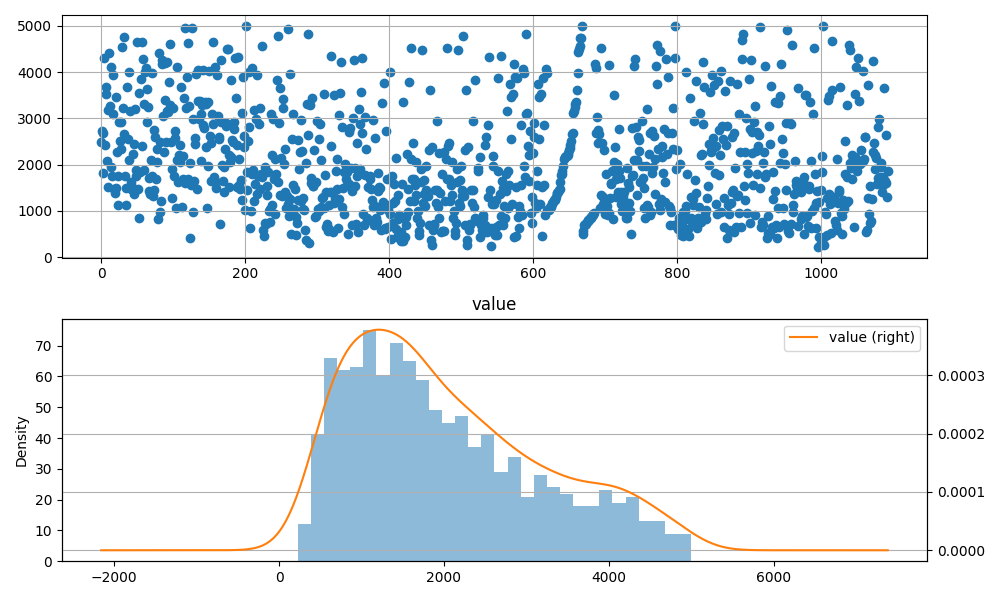
KstestResult(statistic=0.0937781144287082, pvalue=7.98741928777996e-09)

代码如下：

#去掉极端数据  
normdata **= []  
for** i **in** range**(**0, len**(**numOfcomments**)):** # 去掉过于极端数据,第二阶段：1556-=1094，  
 **if (**numOfcomments**[**i**] >** 5000**):  
 continue** normdata.append**(**numOfcomments**[**i**])**

visual**(**normdata, 100**)**kscheck**(**normdata**)**

Step3:用最小二乘法拟合高斯分布，并作可视化



拟合多项式系数,并算出u和sigma

u=1584.1365069686658

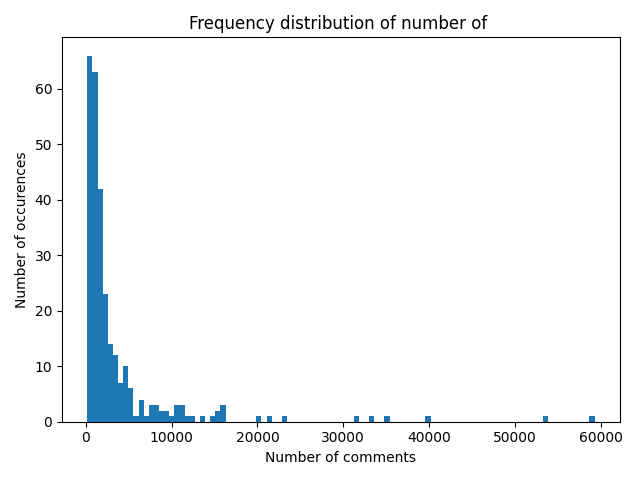
Sigma=1740.831865971861

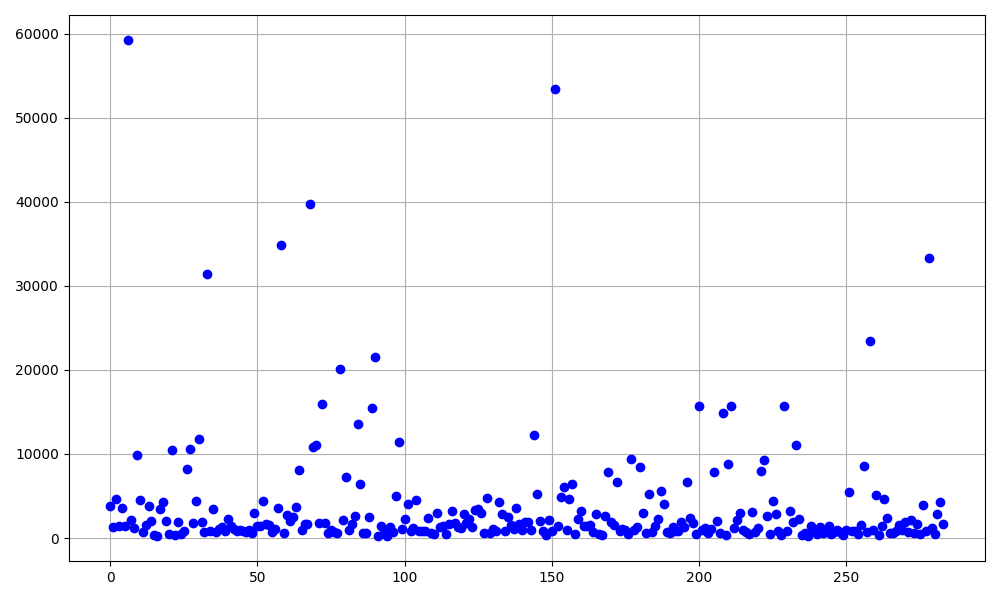
将此正态分布标准化，再查得标准正态分布下的**上alpha分位点**（alpha=0.01）=2.33，由Z=(X-u)/sigma得，相应的**评论数分位点**为5636，即评论数大于5636的为重点疫情新闻。

第三阶段：

共爬取新闻336条，其中疫情相关新闻284条，占85%，去除格式异常的新闻后，剩余284条，即样本容量为284。

Step1:对总体评论数进行可视化以及ks正态检验



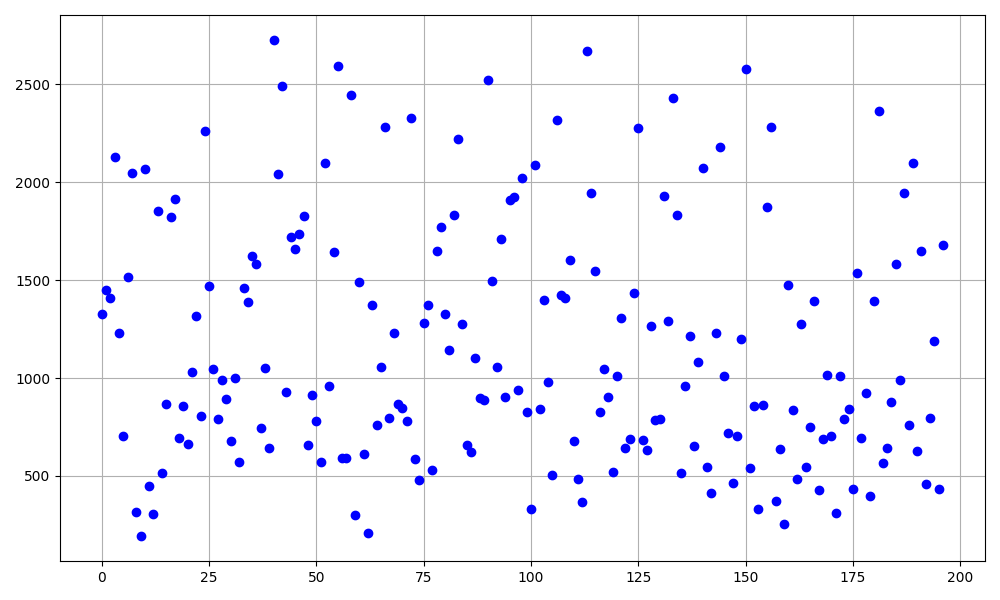
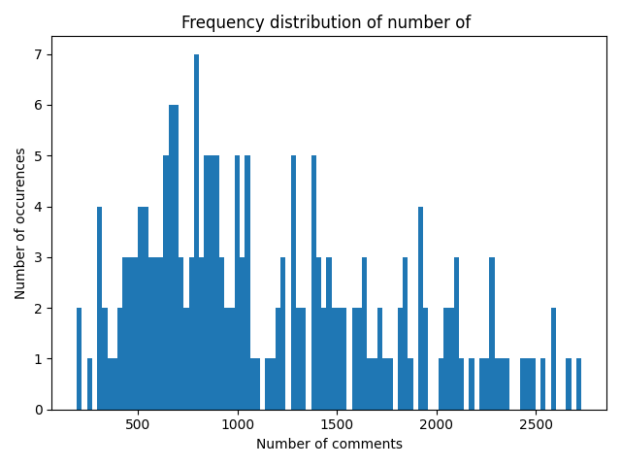


初始样本均值，标准差，ks检验d值，p值：

3722.6830985915494 6915.048457625112

KstestResult(statistic=0.30477207070414913, pvalue=6.578009214583049e-24)

Step2:去掉极端数据，再进行ks正态检验与可视化，保证一定的可信度。



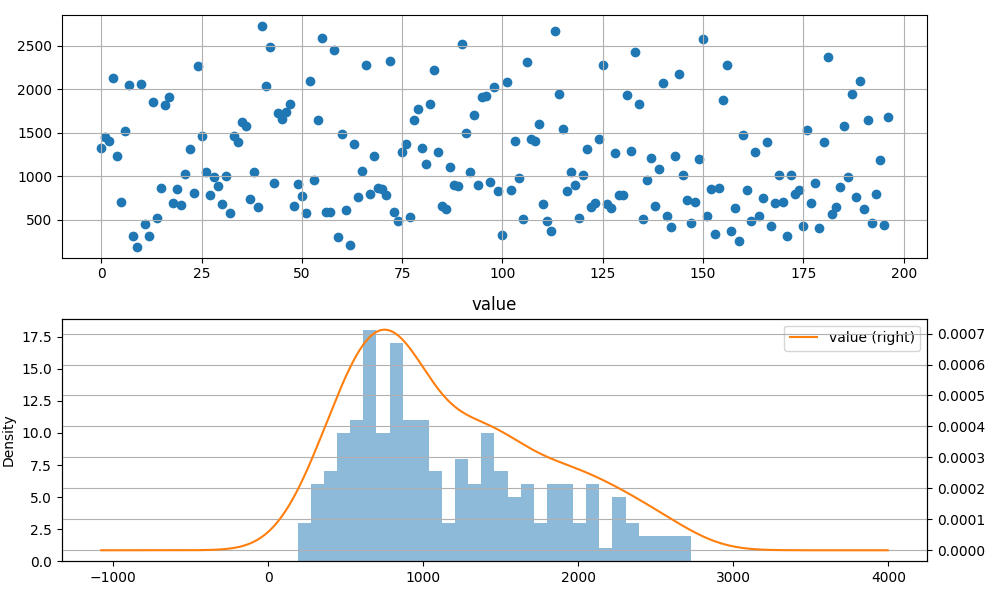
剩余样本均值，标准差，ks检验d值，p值：

1165.2893401015228 616.6037596928999

KstestResult(statistic=0.1230041321502004, pvalue=0.004682842866705808)

可见，p值大幅度增大。

Step3:用最小二乘法拟合高斯分布，并作可视化



拟合多项式系数,算出u和sigma为：

1044.8015291207537

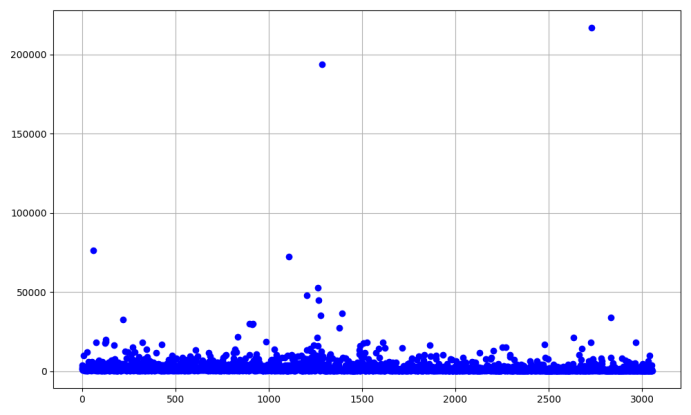
885.9194814672795

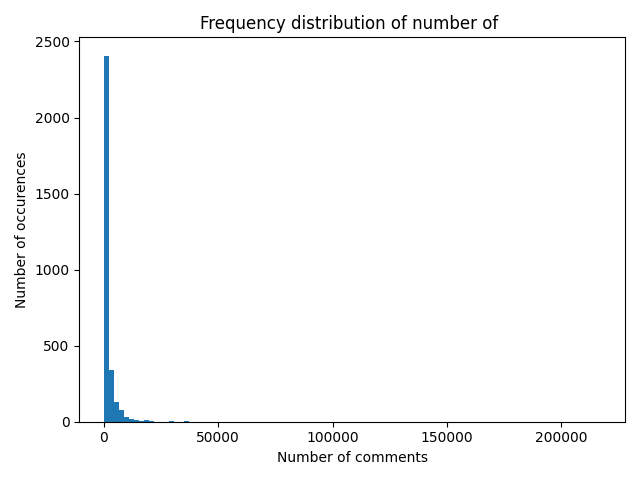
将此正态分布标准化，再查得标准正态分布下的**上alpha分位点**（alpha=0.4）=0.26，由Z=(X-u)/sigma得，相应的**评论数分位点**为1399，即评论数大于1399的为重点疫情新闻。

第四阶段：

共爬取新闻5904条，其中疫情相关新闻3055条，占52%，去除格式异常的新闻后，剩余3055条，即样本容量为3055。

Step1:对总体评论数进行可视化以及ks正态检验





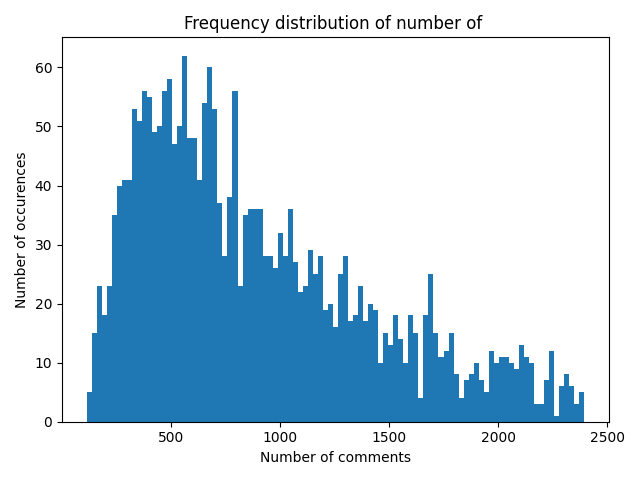
初始样本均值，标准差，ks检验d值，p值：

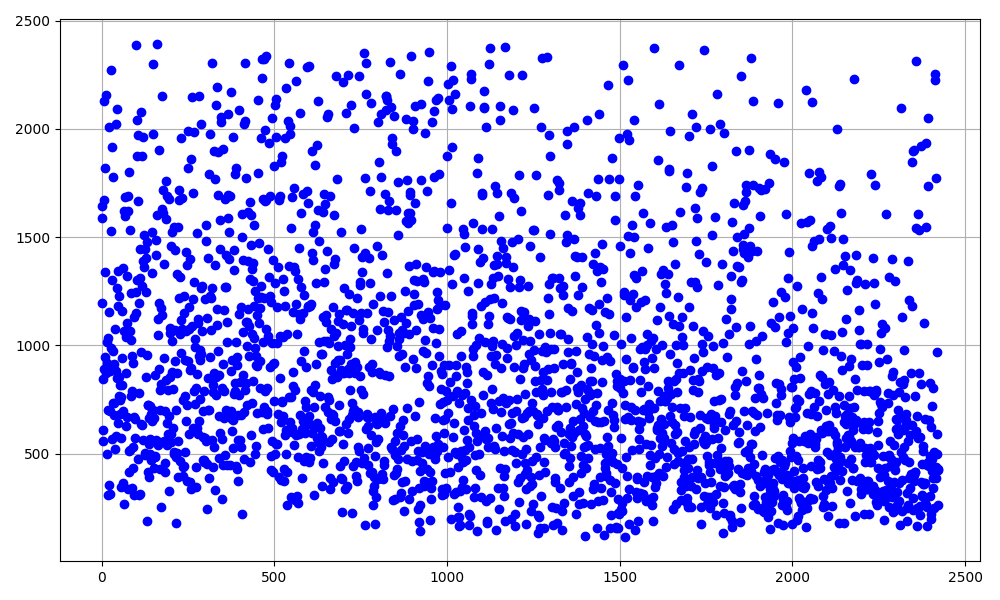
2157.411456628478 6443.74845473788

KstestResult(statistic=0.37569580107091854, pvalue=0.0)

可以看出，p值检验结果直接小到浮点数无法表示了，样本总体不服从正态分布，那么接下来思路与上几个阶段一样，去除一小部分极端数据，并用最小二乘法拟合成高斯分布。

Step2:去掉极端数据，再进行ks正态检验与可视化，保证一定的可信度。





剩余样本均值，标准差，ks检验d值，p值：

904.4913366336634 535.3347527388047

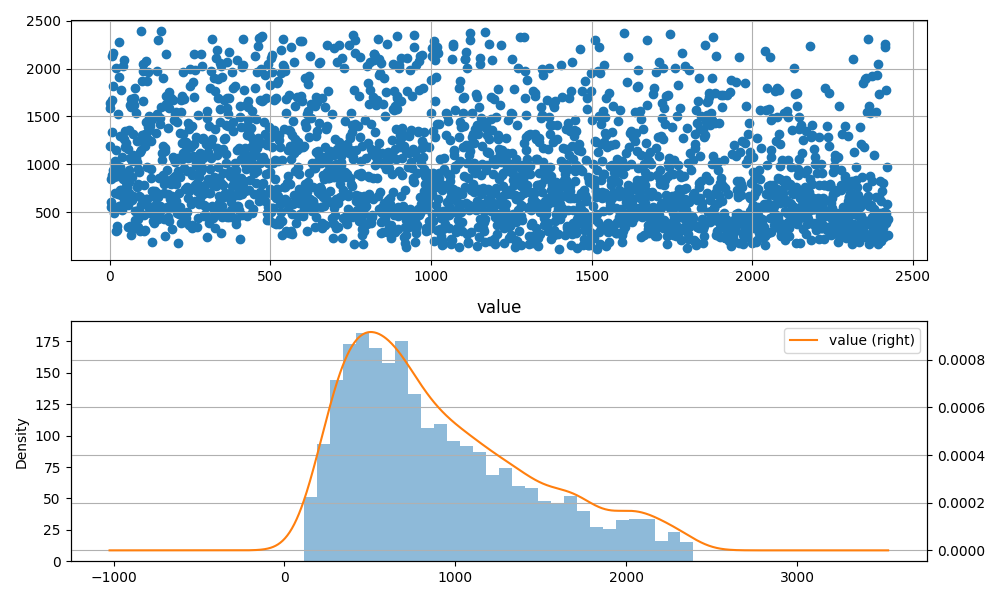
KstestResult(statistic=0.10680217468675945,pvalue=1.5653317522436728e-24)

Step3:用最小二乘法拟合高斯分布，并作可视化

拟合多项式系数,并算出u和sigma

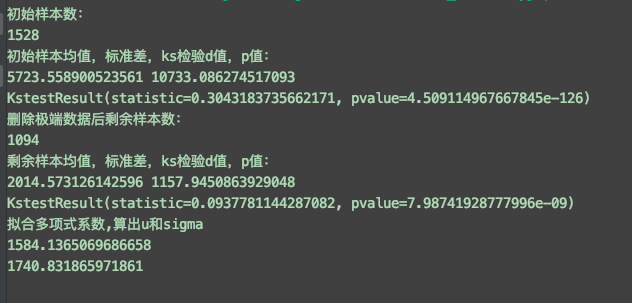
u=478.17889719991797

Sigma=945.8740982452367



将此正态分布标准化，再查得标准正态分布下的**上alpha分位点**（alpha=0.01）=2.33，由Z=(X-u)/sigma得，相应的**评论数分位点**为2682，即评论数大于2682的为重点疫情新闻。

以下为控制台输出：（以第二阶段为例）：



以下对数据处理以及可视化部分所调用的主要方法做个说明与阐释：

Method1:visual，用于数据可视化画图，利用python 的matplotlib.pyplot模块画出一组数据的散点图以及条形统计图（频数图），传入参数为一组数据data（list），和画条形图时，将数据按最大值与最小值之差分为多少组groupsnum。

其中最开头调用了画散点图的方法scatter\_plot，其参数也为一组列表数据。Scatter\_plot方法横坐标为dataframe的从1到n的数据的index，然后纵坐标为具体数据。

代码如下：

**def visual(**data,groupsnum**):** scatter\_plot**(**data**)** # 对数据进行切片，将数据按照从最小值到最大值分组，分成20组  
 bins **=** np.linspace**(**min**(**data**)**, max**(**data**)**, groupsnum**)** # 这个是调用画直方图的函数，意思是把数据按照从bins的分割来画  
 plt.hist**(**sorted**(**data**)**, bins**)** # 设置出横坐标  
 plt.xlabel**('Number of comments')** # 设置纵坐标的标题  
 plt.ylabel**('Number of occurences')** # 设置整个图片的标题  
 plt.title**('Frequency distribution of number of ')** # 展示出我们的图片  
 plt.show**()**

**def scatter\_plot(**data**):** data**=**np.array**(**data**)** s**=** pd.DataFrame**(**data, columns**=['array'])** plt.figure**("beautiful graph"**,figsize**=(**10, 6**))** plt.plot**(**s.index,s.values,**'bo')** plt.grid**()** plt.show**()**

Method2:kscheck，用于检验一组数据是否符合正态分布，传入参数为一组数据（list），先将数据转化为dataframe类型，然后计算并输出此组数据的均值和标准差，再输出ks检验的结果。

**def kscheck(**data**):** # scipy包是一个高级的科学计算库，它和Numpy联系很密切，Scipy一般都是操控Numpy数组来进行科学计算  
 df **=** pd.DataFrame**(**data, columns**=['value'])** u **=** df**['value']**.mean**()** # 计算均值  
 std **=** df**['value']**.std**()** # 计算标准差  
 print**(**u, std**)** print**(**stats.kstest**(**df**['value']**, **'norm'**, **(**u, std**)))**

Method3:normfit，用于最小二乘法拟合正态分布，传入参数为一组数据（list）。

首先对这组数据划分区间，统计频数，并计算频率。

其次，将每个区间中间值作为横坐标，频率/区间长度作为纵坐标，从而获得散点，并且去掉那些频数为0的点。

再次，做最小二乘法拟合。

在这里需要说明，当把原高斯函数y=(1/(math.sqrt(2\*math.pi)\*sigma))\*(np.exp(-(x-u)\*(x-u)/(2\*sigma\*sigma)))作为最小二乘法的拟合函数时，所得的pcov协方差矩阵会直接上溢（即其值为inf），多次调整数据也无法拟合。这是因为原数据正态性太差造成的。因此，采用下面这种取对数转化的方式进行拟合。

由于y=(1/(math.sqrt(2\*math.pi)\*sigma))\*(np.exp(-(x-u)\*(x-u)/(2\*sigma\*sigma)))正态函数具有非负性，故两边取对数得：

ln(y)=(-1/(2\*sigma^2))\*x^2+(u/(2\*sigma^2))\*x+ln(1/(sqrt(2\*pi)\*sigma))-u^2/(2\*sigma^2)

再令 a = -1/(2\*sigma^2) ；

b = u/(2\*sigma^2) ；

c = ln(1/(sqrt(2\*pi)\*sigma))-u^2/(2\*sigma^2)

则lny与x有二次函数的关系，于是转而拟合lny与x，即设lny=ax^2+bx+c

作为拟合函数，得到a，b后，进而解得u和sigma的值。

最后即可建立正态分布。

代码如下：

**def normal\_func(**x,u,sigma**):  
 return (**1**/(**math.sqrt**(**2**\***math.pi**)\***sigma**))\*(**np.exp**(-(**x**-**u**)\*(**x**-**u**)/(**2**\***sigma**\***sigma**)))  
def normal\_polfunc(**x, a, b, c**):  
 return** a **\*** x **\*** x **+** b **\*** x **+** c  
**def normfit(**data**):** y **= []** step **=** 100  
 i **=** min**(**data**)** j **=** i **+** step  
 sorteddata **=** sorted**(**data**)** index **=** 0  
 x **= []  
 while** i **<=** max**(**data**) and** index **<** len**(**sorteddata**):** f **=** 0 # 此区间频数  
 **while** index **<** len**(**sorteddata**) and** sorteddata**[**index**] >=** i **and** sorteddata**[**index**] <** j**:** f **+=** 1  
 index **+=** 1  
 i **=** j  
 j **=** i **+** step  
 **if** f **==** 0**:  
 continue** x.append**((**i **+** j**) /** 2**)** # 中间值  
 y.append**(**float**(**f **/ (**len**(**sorteddata**) \*** step**)))** x **=** np.array**(**x**)** y **=** np.array**(**y**)** lny**=**np.log**(**y**)** popt, pcov **=** curve\_fit**(**normal\_polfunc, x, lny**)** a **=** popt**[**0**]** b **=** popt**[**1**]** c **=** popt**[**2**]** u **= (-**b**) / (**2 **\*** a**)** sigma **=** math.sqrt**(-**1 **/ (**2 **\*** a**))** print**(**u**)** print**(**sigma**)**

当做完最小二乘法拟合高斯分布后，再画出近似拟合曲线与频率分布直方图，可以直观地看到数据的特征。

代码如下：

# 绘图  
s **=** pd.DataFrame**(**data, columns**=['value'])**# 画散点图和直方图  
fig **=** plt.figure**(**figsize**=(**10, 6**))**ax1 **=** fig.add\_subplot**(**2, 1, 1**)** # 创建子图1  
ax1.scatter**(**s.index, s.values**)**plt.grid**()**ax2 **=** fig.add\_subplot**(**2, 1, 2**)** # 创建子图2  
s.hist**(**bins**=**30, alpha**=**0.5, ax**=**ax2**)**s.plot**(**kind**='kde'**, secondary\_y**=True**, ax**=**ax2**)**plt.grid**()**plt.show**()**

1. 重点新闻评论写入excel

Step1:以第二阶段为例，先建立要写入的excel文件，再选取评论数在评论分位点以上的新闻，将其70条评论写入

writecol**=**0  
workbook **=** xlsxwriter.Workbook**('/Users/sebas/第二阶段（12.8~1.22）.xlsx')** # 创建一个excel文件  
worksheet **=** workbook.add\_worksheet**(u'sheet1')** # 在文件中创建一个名为TEST的sheet,不加名字默认为sheet1  
**for** i **in** range**(**0,len**(**relatednews**)):  
 if (**relatednews**[**i**][**0**] >** 5624**):  
 for** j **in** range**(**1,71**):** worksheet.write**(**writecol, 0,relatednews**[**i**][**j**])** # 使用行列的方式写上数字32,35,5  
 writecol**+=**1  
workbook.close**()**

实验结果：

对四个阶段的数据进行了可视化，建立了分布。

第一阶段新闻评论数近似服从正态分布。

第一，二，三，四阶段的数据用最小二乘法拟合成了高斯分布。

得到疫情相关重点新闻评论第一阶段7630条，第二阶段27160条，第三阶段12110条，第四阶段38710条。

讨论与结论：

从每个阶段疫情相关新闻的占比来看，在第一阶段12.8～1.22时，疫情相关新闻较少，只占到15%，其评论数也很少，体现出疫情刚刚开始蔓延并未引起极大关注，疫情的了解程度较低，知晓疫情相关情况的人也较少，这段时间疫情悄然大幅度蔓延。第二阶段武汉开始封城，1.23～2.7正是人们要回家过年，享受家人团聚的时候。然而这段时间也是我国与疫情作战的关键时段，湖北物资和医护人员短缺，各省各地区都派出援鄂医疗队，此时几乎每天的绝大多数新闻都是疫情相关的，占85%每天的确诊人数都在不断更新报道。人们开始极大重视，进行隔离。第三阶段只有三天，中央政府开始严格管控，全国人民极大关注，新闻占比达到85%以上。第四阶段时间跨度较长，包括了武汉解封，人们的生活渐渐回到正轨的过程。然而，疫情依然不容轻视，我们依然要做好防护，故这个时间段疫情新闻也占到了52%，这说明疫情并未过去，我们还有很长的路要走。

从每个阶段疫情相关新闻评论数的分布来看，各个阶段的正态均值由小到大再变小，也体现了媒体和大众对于疫情的重视程度的变化。由一开始每条新闻的评论基本在1000条以下，到第二阶段，第三阶段正态均值达到5723，3745，再到第四阶段2100条，人们一开始虽然不够重视，但这也取决于相关媒体的报道较少，然而，有了二三阶段和疫情的战斗，现在人们依旧不敢掉以轻心，第四阶段的评论量依然处在较高水平。

从极端数据来看，第一阶段只有两条新闻的评论量在6w以上，其余都在5w以下。这两条新闻摘要为：【#全国已确诊473例新型肺炎病例#】#新型肺炎疫情最新汇总#：截至1月22日19时20分，全国新型冠状病毒感染肺炎确诊473例。【#新型肺炎已致湖北17人死亡#】#湖北新型肺炎累计444例死亡17例#湖北省政府召开新闻发布会，截至今天晚上8点，湖北省累计报告新型冠状病毒感染病例444例，17人死亡。这说明，新冠的前期较高死亡率和迅速上升的确诊人数已经引起了人们的警觉。第二阶段只有一条评论20w以上的新闻，为：【24小时不间断直播！#直击火神山医院施工现场#】武汉火神山医院正昼夜不停加紧建设，人民直播联合@长江日报，带你走进施工现场↓#云监工火神山建设#人民日报的微博视频。体现了当时全国人民一起为火神山医院的建设加油，关注着抗疫前线。第三阶段有两条新闻评论量在5w以上，分别为：【#武汉排查出1499名重症患者全部入院#】武汉市委书记马国强：关于集中救治，进行了全市的排查。通过排查，到2月8日，有确诊重症患者1499人没有入院治疗，到10日中午1499名确诊重症患者全部入院。#11日武汉目标全部疑似患者检测清零#。L人民日报的微博视频。这条说明了武汉的医疗物资和人员终于能够将所有重症患者纳入并开始治疗了。这标志着经过数天各地区援鄂医疗队和武汉人民的努力，终于不至于让患者无法接受治疗了，武汉疫情终于得到了缓和。【#教育部回应高考是否推迟#】在12日召开的国务院联防联控机制发布会上，教育部高校学生司司长王辉表示，要把考生和涉考工作人员健康安全放第一位，距离高考还有4个月，将密切关注疫情发展变化，会同相关部门和各地评估疫情对组织考试可能引发的问题，审慎研究制定今年高考方案。L人民日报的微博视频。这条是关于高考的。由此看出疫情对于高考的影响程度以及人民对高考的重视程度。第四阶段只有一条是评论20w以上，为：【直播来了！#致敬白衣天使公益云演唱会#】这是一场独特的演唱会，每首歌都为白衣天使们点唱。这更是一次真心的感谢，感谢你们不畏生死、奋不顾身。今晚你们的愿望，我们一起来守护。超强嘉宾阵容，一起为爱放歌！（来自O@人民日报的微博直播）APP下载地址：O网页链接。这是社会各界都在以他们的方式致敬那些在疫情中作出贡献的逆行者们。

参考资料：

<https://blog.csdn.net/huanyingzhizai/article/details/90479836>

<https://blog.csdn.net/fenghao289/article/details/6340745>

<https://blog.csdn.net/laobai1015/article/details/107845838>

<https://blog.csdn.net/u014597198/article/details/83104653>