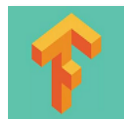


# [Python疫情大数据分析] 二.PyEcharts绘制全国各地区、某省各城市疫情地图及可视化分析

原创 Eastmount 最后发布于2020-04-11 23:28:00 阅读数 3676 ☆ 收藏

编辑 展开



## Python+TensorFlow人工智能

该专栏为人工智能入门专栏，采用Python3和TensorFlow实现人工智能相关算法。前期介绍安装流程、基础语法...

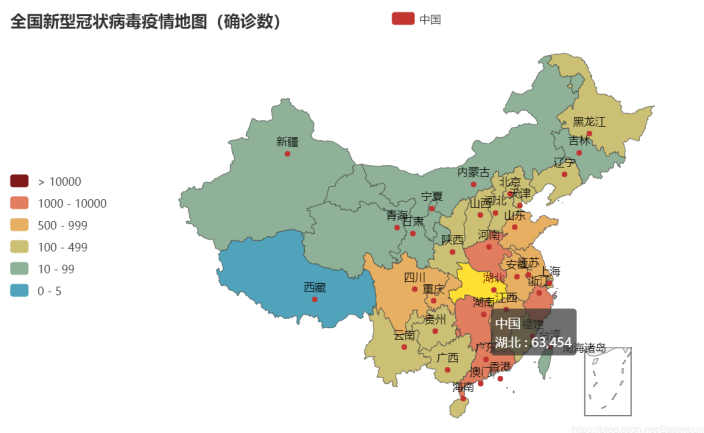
Eastmount

¥9.90

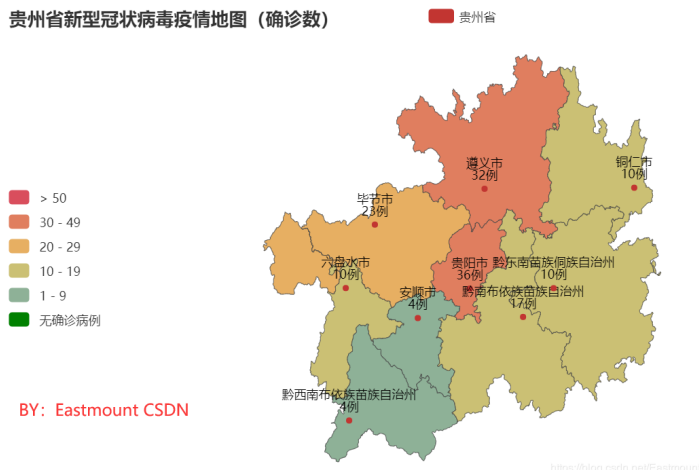
去订阅

思来想去，虽然很忙，但还是挤时间针对这次肺炎疫情写个Python大数据分析系列博客，包括网络爬虫、可视化分析、GIS地图显示、情感分析、舆情分析、主题挖掘、威胁情报溯源、知识图谱、预测预警及AI和NLP应用等。希望该系列线上远程教学对您有所帮助，也希望早点战胜病毒，武汉加油、湖北加油、全国加油。待到疫情结束樱花盛开，这座英雄的城市等你们来。

全国新型冠状病毒疫情地图（确诊数）



贵州省新型冠状病毒疫情地图（确诊数）



BY: Eastmount CSDN

前文分享了腾讯疫情实时数据抓取，获取全国各地和贵州省各地区的实时数据，并将数据存储至本地，最后调用Matplotlib和Seaborn绘制。本文将结合PyEcharts绘制地图、折线图、柱状图，希望这篇可视化分析文章对您有所帮助，也非常感谢参考文献中老师的分享，一起加油，战胜疫情！如果您有想学习的知识或建议，可以给作者留言~

代码下载地址：<https://github.com/eastmountyxz/Wuhan-data-analysis>

CSDN下载地址：<https://download.csdn.net/download/Eastmount/12239638>

## 文章目录

### 一.数据爬取及PyEcharts绘制折线图

二.PyEcharts绘制全国各地区

三.PyEcharts绘制贵州省地区

四.PyEcharts绘制其他图形

五.总结

同时推荐前面作者另外五个Python系列文章。从2014年开始，作者主要写了三个Python系列文章，分别是基础知识、网络爬虫和数据分析。2018年陆续增加了Python图像识别和Python人工智能专栏。

- Python基础知识系列: [Python基础知识学习与提升](#)
- Python网络爬虫系列: [Python爬虫之Selenium+BeautifulSoup+Requests](#)
- Python数据分析系列: 知识图谱、web数据挖掘及NLP
- Python图像识别系列: [Python图像处理及图像识别](#)
- Python人工智能系列: [Python人工智能及知识图谱实战](#)



前文阅读:

[Python疫情大数据分析] 一.腾讯实时数据爬取、Matplotlib和Seaborn可视化分析全国各地区、某省各城市、新增趋势

## 一.数据爬取及PyEcharts绘制折线图

前一篇文章作者详细讲解了腾讯新闻实时数据抓取过程，为了更好地进行可视化分析或数据分析，建议读者将数据存储至本地或数据库中。这里作者直接给出网络爬虫代码，将每日数据增长情况存储至本地。

爬虫目标网站:

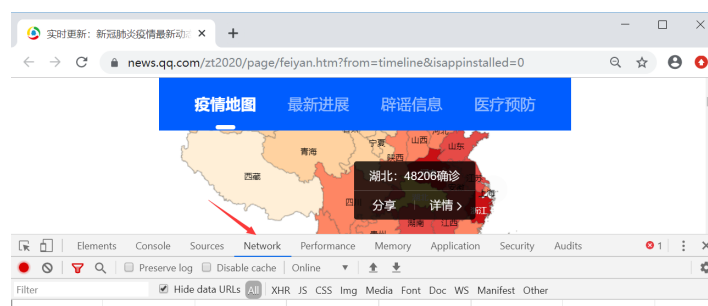
- <https://news.qq.com/zt2020/page/feiyan.htm>

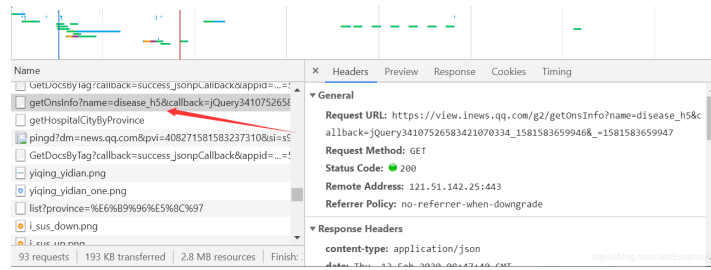
推荐参考文章:

- [Python疫情大数据分析] 一.腾讯实时数据爬取、Matplotlib和Seaborn可视化分析全国各地区、某省各城市、新增趋势
- [Python实战: 抓肺炎疫情实时数据, 画2019-nCoV疫情地图](#) - 许老师
- [用Python抓新型冠状病毒肺炎疫情数据, 绘制全国疫情分布图](#) - shineych老师
- [2020Python开发者日: 爬虫框架的技术实现与模块应用的经验分享](#) - 许老师

### 第一步 分析网站

通过浏览器“审查元素”查看源代码及“网络”反馈的消息, 如下图所示:





## 第二步 网络爬虫代码

通过分析url地址、请求方法、参数及响应格式，可以获取Json数据，注意url需要增加一个时间戳，同时根据不同日期将数据存储至本地CSV文件。

```
# -*- coding: utf-8 -*-
import time, json, requests
from datetime import datetime

#-----
# 第一步 抓取腾讯疫情实时json数据
# 参考文章: 许老师博客 https://blog.csdn.net/xufive/article/details/104093197
#-----

def catch_daily():
    url = 'https://view.inews.qq.com/g2/get0nsInfo?name=wuwei_ww_cn_day_counts&callback=&_=%d'%int(time.time()*1000)
    data = json.loads(requests.get(url=url).json()['data'])
    data.sort(key=lambda x:x['date'])

    date_list = list()      # 日期
    confirm_list = list()  # 确诊
    suspect_list = list()  # 疑似
    dead_list = list()     # 死亡
    heal_list = list()     # 治愈
    for item in data:
        month, day = item['date'].split('/')
        date_list.append(datetime.strptime('2020-%s-%s'%(month, day), '%Y-%m-%d'))
        confirm_list.append(int(item['confirm']))
        suspect_list.append(int(item['suspect']))
        dead_list.append(int(item['dead']))
        heal_list.append(int(item['heal']))
    return date_list, confirm_list, suspect_list, dead_list, heal_list

#-----
# 第二步 存储数据至CSV文件
#-----

def load_csv():
    # 获取数据
    date_list, confirm_list, suspect_list, dead_list, heal_list = catch_daily()
    print(date_list)      # 日期
    print(confirm_list)   # 确诊数据
    print(suspect_list)   # 疑似数据
    print(dead_list)      # 死亡数据
    print(heal_list)      # 治愈数据

    # 获取当前日期命名(2020-02-13-daily.csv)
    n = time.strftime("%Y-%m-%d") + "-daily.csv"
    fw = open(n, 'w', encoding='utf-8')
    fw.write('date,confirm,suspect,dead,heal\n')
    i = 0
```

```

while i<len(date_list):
    date = str(date_list[i].strftime("%Y-%m-%d"))

    fw.write(date+', '+str(confirm_list[i])+', '+str(suspect_list[i])+', '+str(dead_list[i])+', '+str(heal_list[i])+'\\n')
    i = i + 1
else:
    print("Over write file!")
    fw.close()

# 主函数
if __name__ == '__main__':
    load_csv()

```

输出结果如下图所示，可以看到将1月13日至2月21日的疫情数据抓取至本地。

```

[datetime.datetime(2020, 1, 13, 0, 0), datetime.datetime(2020, 1, 14, 0, 0), datetime.datetime(2020, 1, 15, 0, 0), datetime.datetime(2020, 1, 16, 0, 0), datetime.datetime(2020, 1, 17, 0, 0), datetime.datetime(2020, 1, 18, 0, 0), datetime.datetime(2020, 1, 19, 0, 0), datetime.datetime(2020, 1, 20, 0, 0), datetime.datetime(2020, 1, 21, 0, 0), datetime.datetime(2020, 1, 22, 0, 0), datetime.datetime(2020, 1, 23, 0, 0), datetime.datetime(2020, 1, 24, 0, 0), datetime.datetime(2020, 1, 25, 0, 0), datetime.datetime(2020, 1, 26, 0, 0), datetime.datetime(2020, 1, 27, 0, 0), datetime.datetime(2020, 1, 28, 0, 0), datetime.datetime(2020, 1, 29, 0, 0), datetime.datetime(2020, 1, 30, 0, 0), datetime.datetime(2020, 1, 31, 0, 0), datetime.datetime(2020, 2, 1, 0, 0), datetime.datetime(2020, 2, 2, 0, 0), datetime.datetime(2020, 2, 3, 0, 0), datetime.datetime(2020, 2, 4, 0, 0), datetime.datetime(2020, 2, 5, 0, 0), datetime.datetime(2020, 2, 6, 0, 0), datetime.datetime(2020, 2, 7, 0, 0), datetime.datetime(2020, 2, 8, 0, 0), datetime.datetime(2020, 2, 9, 0, 0), datetime.datetime(2020, 2, 10, 0, 0), datetime.datetime(2020, 2, 11, 0, 0), datetime.datetime(2020, 2, 12, 0, 0), datetime.datetime(2020, 2, 13, 0, 0), datetime.datetime(2020, 2, 14, 0, 0), datetime.datetime(2020, 2, 15, 0, 0), datetime.datetime(2020, 2, 16, 0, 0), datetime.datetime(2020, 2, 17, 0, 0), datetime.datetime(2020, 2, 18, 0, 0), datetime.datetime(2020, 2, 19, 0, 0), datetime.datetime(2020, 2, 20, 0, 0), datetime.datetime(2020, 2, 21, 0, 0)]
[41, 41, 41, 45, 62, 198, 275, 291, 440, 574, 835, 1297, 1985, 2761, 4535, 5997, 7736, 9720, 11821, 14411, 17238, 20471, 2436
3, 28060, 31211, 34598, 37251, 40235, 42708, 44730, 59882, 63932, 66576, 68584, 70635, 72528, 74279, 75002, 75993, 76392]
[0, 0, 0, 0, 0, 0, 54, 37, 393, 1072, 1965, 2684, 5794, 6973, 9239, 12167, 15238, 17988, 19544, 21558, 23214, 23260, 24702
, 26359, 27657, 28942, 23589, 21675, 16067, 13435, 10109, 8969, 8228, 7264, 6242, 5248, 4922, 5206, 5365]
[1, 1, 2, 2, 2, 3, 4, 6, 9, 17, 25, 41, 56, 80, 106, 132, 170, 213, 259, 304, 361, 425, 491, 564, 637, 723, 812, 909, 1017, 1
114, 1368, 1381, 1524, 1666, 1772, 1870, 2006, 2121, 2239, 2348]
[0, 0, 5, 8, 12, 17, 18, 25, 25, 34, 38, 49, 51, 60, 103, 124, 171, 243, 328, 475, 632, 892, 1153, 1542, 2052, 2651, 3283
, 3998, 4742, 5915, 6728, 8101, 9425, 10853, 12561, 14387, 16157, 18266, 20673]
Over write file!
>>>]

```

<https://blog.csdn.net/qq404040404>

### 第三步 安装PyEcharts扩展包

前端或网站开发的博主可能都使用过强大的Echarts技术。ECharts是一个纯Javascript的图表库，可以流畅的运行在PC和移动设备上，兼容当前绝大部分浏览器，底层依赖轻量级的Canvas类库ZRender，提供直观、生动、可交互、可高度个性化定制的数据可视化图表。ECharts提供了常规的折线图、柱状图、散点图、饼图、K线图，用于统计的盒形图，用于地理数据可视化的地图、热力图、线图，用于关系数据可视化的关系图、treemap，多维数据可视化的平行坐标，还有用于BI的漏斗图、仪表盘，并且支持图与图之间的混搭。

下图是Echarts简单绘制折线图的示例，左边是脚本代码，右边是显示图形，非常美观。

<http://echarts.baidu.com/echarts2/doc/example/line1.html#helianthus>

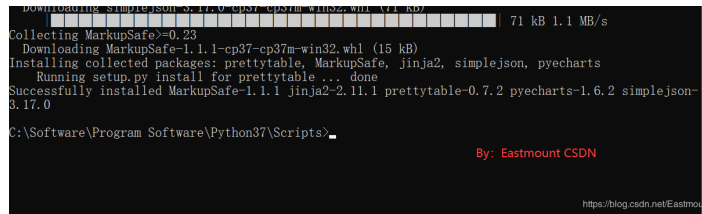


既然Echarts这么好用，Python也有相应的第三方扩展包支持，它就是这篇文章讲解的PyEcharts库。PyEcharts是一个用于生成Echarts图表的类库，即Echarts与Python的对接，并推荐大家结合Django或Flask使用。接下来我们通过“pip install pyecharts”安装扩展包，再调用PyEcharts绘制柱状图。

```

C:\WINDOWS\system32\cmd.exe
C:\Software\Program Software\Python37\Scripts>pip install pyecharts
Collecting pyecharts
  Downloading pyecharts-1.6.2-py3-none-any.whl (168 kB)
Collecting prettytable
  Downloading prettytable-0.7.2.tar.bz2 (21 kB)
Collecting jinja2
  Downloading Jinja2-2.11.1-py2.py3-none-any.whl (126 kB)
Collecting simplejson

```



推荐作者前文：

[\[Echarts可视化\] 一.入门篇之简单绘制中国地图和贵州地区](#)

[\[Python可视化\] pyecharts安装入门及绘制中国贵州地图](#)

#### 第四步 调用PyEcharts绘制柱状图

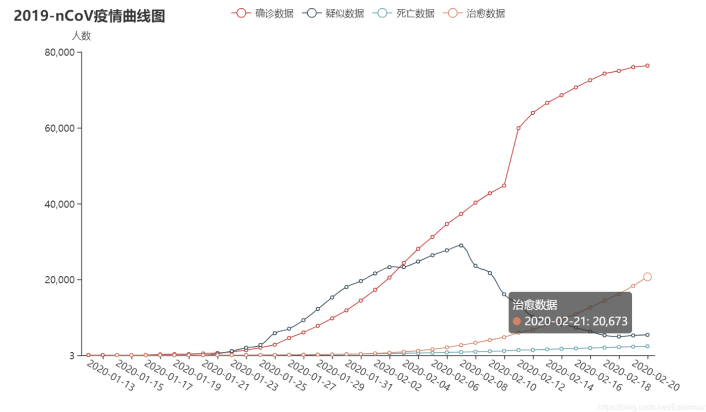
```
# -*- coding: utf-8 -*-
# By: Eastmount CSDN xiuzhang
import time, json, requests
from datetime import datetime
import pandas as pd
import pyecharts.options as opts
from pyecharts.charts import Line
from pyecharts.commons.utils import JsCode

#-----
# 第一步：读取数据
#-----
n = time.strftime("%Y-%m-%d") + "-daily.csv"
data = pd.read_csv(n)
date_list = list(data['date'])
confirm_list = list(data['confirm'])
suspect_list = list(data['suspect'])
dead_list = list(data['dead'])
heal_list = list(data['heal'])
print(date_list)           # 日期
print(confirm_list)        # 确诊数据
print(suspect_list)        # 疑似数据
print(dead_list)           # 死亡数据
print(heal_list)           # 治愈数据

#-----
# 第二步：绘制折线图
#-----
line = (
    Line()
    .add_xaxis(date_list)
    .add_yaxis('确诊数据', confirm_list)
    .add_yaxis('疑似数据', suspect_list, is_smooth=True) #平滑
    .add_yaxis('死亡数据', dead_list)
    .add_yaxis('治愈数据', heal_list)
    # 隐藏数字
    .set_series_opts(label_opts=opts.LabelOpts(is_show=False))
    # 设置x轴标签旋转角度
    .set_global_opts(xaxis_opts=opts.AxisOpts(axislabel_opts=opts.LabelOpts(rotate=-30)),
                     yaxis_opts=opts.AxisOpts(name='人数', min_=3),
                     title_opts=opts.TitleOpts(title='2019-nCoV疫情曲线图'))
)

line.render('2019-nCoV疫情曲线图.html')
```

输出结果如下图所示，隐藏了所有点对应的数字，否则整体效果比较乱。同时，当点击某个点能看到具体的数值，比如2月21日全国总治愈人数为20673。



## 第五步 绘制折线面积图及平均曲线

```
# -*- coding: utf-8 -*-
# By: Eastmount CSDN xiuzhang
import time, json, requests
from datetime import datetime
import pandas as pd
import pyecharts.options as opts
from pyecharts.charts import Line
from pyecharts.commons.utils import JsCode

# -----
# 第一步: 读取数据
# -----
n = time.strftime("%Y-%m-%d") + "-daily.csv"
data = pd.read_csv(n)
date_list = list(data['date'])
confirm_list = list(data['confirm'])
suspect_list = list(data['suspect'])
dead_list = list(data['dead'])
heal_list = list(data['heal'])
print(date_list)          # 日期
print(confirm_list)       # 确诊数据
print(suspect_list)       # 疑似数据
print(dead_list)          # 死亡数据
print(heal_list)          # 治愈数据

# -----
# 第二步: 绘制折线面积图
# -----
line = (
    Line()
    .add_xaxis(date_list)
    .add_yaxis('确诊数据', confirm_list, is_smooth=True,
               markline_opts=opts.MarkLineOpts(data=[opts.MarkLineItem(type_="average")]))
    .add_yaxis('疑似数据', suspect_list, is_smooth=True,
               markline_opts=opts.MarkLineOpts(data=[opts.MarkLineItem(type_="average")]))
    .add_yaxis('死亡数据', dead_list, is_smooth=True,
               markline_opts=opts.MarkLineOpts(data=[opts.MarkLineItem(type_="average")]))
    .add_yaxis('治愈数据', heal_list, is_smooth=True,
               markline_opts=opts.MarkLineOpts(data=[opts.MarkLineItem(type_="average")]))
    # 隐藏数字 设置面积
```

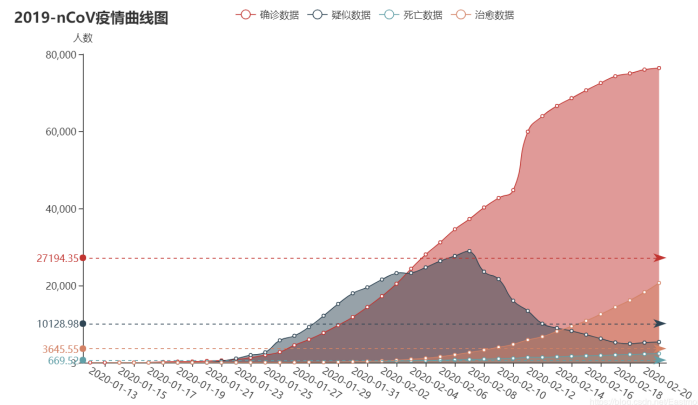
```

.set_series_opts(
    areastyle_opts=opts.AreaStyleOpts(opacity=0.5),
    label_opts=opts.LabelOpts(is_show=False))
# 设置x轴标签旋转角度
.set_global_opts(xaxis_opts=opts.AxisOpts(axislabel_opts=opts.LabelOpts(rotate=-30)),
    yaxis_opts=opts.AxisOpts(name='人数', min_=3),
    title_opts=opts.TitleOpts(title='2019-nCoV疫情曲线图'))
)

line.render('2019-nCoV疫情曲线图2.html')

```

输出结果如下图所示，可以看到确诊数据、疑似数据、死亡数据和治愈数据的面积及平均曲线。



## 第六步 增加最大值和最小值

```

# -*- coding: utf-8 -*-
# By: Eastmount CSDN xiuzhang
import time, json, requests
from datetime import datetime
import pandas as pd
import pyecharts.options as opts
from pyecharts.charts import Line
from pyecharts.commons.utils import JsCode

#-----
# 第一步: 读取数据
#-----

n = time.strftime("%Y-%m-%d") + "-daily.csv"
data = pd.read_csv(n)
date_list = list(data['date'])
confirm_list = list(data['confirm'])
suspect_list = list(data['suspect'])
dead_list = list(data['dead'])
heal_list = list(data['heal'])
print(date_list)           # 日期
print(confirm_list)        # 确诊数据
print(suspect_list)        # 疑似数据
print(dead_list)           # 死亡数据
print(heal_list)           # 治愈数据

#-----
# 第二步: 绘制折线面积图
#-----

line = (
    Line()

```

```

.add_xaxis(date_list)
# 平均线 最大值 最小值
.add_yaxis('确诊数据', confirm_list, is_smooth=True,
           markline_opts=opts.MarkLineOpts(data=[opts.MarkLineItem(type_="average")]),
           markpoint_opts=opts.MarkPointOpts(data=[opts.MarkPointItem(type_="max"),
                                                    opts.MarkPointItem(type_="min")])))

.add_yaxis('疑似数据', suspect_list, is_smooth=True,
           markline_opts=opts.MarkLineOpts(data=[opts.MarkLineItem(type_="average")]),
           markpoint_opts=opts.MarkPointOpts(data=[opts.MarkPointItem(type_="max"),
                                                    opts.MarkPointItem(type_="min")])))

.add_yaxis('死亡数据', dead_list, is_smooth=True,
           markline_opts=opts.MarkLineOpts(data=[opts.MarkLineItem(type_="average")]),
           markpoint_opts=opts.MarkPointOpts(data=[opts.MarkPointItem(type_="max"),
                                                    opts.MarkPointItem(type_="min")])))

.add_yaxis('治愈数据', heal_list, is_smooth=True,
           markline_opts=opts.MarkLineOpts(data=[opts.MarkLineItem(type_="average")]),
           markpoint_opts=opts.MarkPointOpts(data=[opts.MarkPointItem(type_="max"),
                                                    opts.MarkPointItem(type_="min")])))

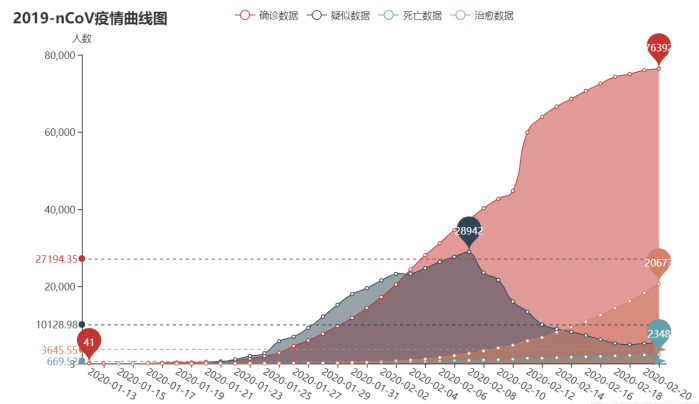
# 隐藏数字 设置面积
.set_series_opts(
    areastyle_opts=opts.AreaStyleOpts(opacity=0.5),
    label_opts=opts.LabelOpts(is_show=False))
# 设置x轴标签旋转角度
.set_global_opts(xaxis_opts=opts.AxisOpts(axislabel_opts=opts.LabelOpts(rotate=-30)),
                 yaxis_opts=opts.AxisOpts(name='人数', min_=3),
                 title_opts=opts.TitleOpts(title='2019-nCoV疫情曲线图'))

)

line.render('2019-nCoV疫情曲线图3.html')

```

输出结果如下图所示：



## 二.PyEcharts绘制全国各地区

### 第一步 下载数据

如何获取全国各地数据，推荐大家阅读上一篇文章。

```

# -*- coding: utf-8 -*-
# -----
# 第一步：抓取数据
# -----
import time, json, requests

```



```

# 抓取腾讯疫情实时json数据
url = 'https://view.inews.qq.com/g2/get0nsInfo?name=disease_h5&callback=&_=%d'%int(time.time()*1000)
data = json.loads(requests.get(url=url).json()['data'])
print(data)
print(data.keys())

# 统计省份信息(34个省份 湖北 广东 河南 浙江 湖南 安徽...)
num = data['areaTree'][0]['children']
print(len(num))
for item in num:
    print(item['name'],end=" ") # 不换行
else:
    print("\n") # 换行

# 显示湖北省数据
hubei = num[0]['children']
for item in hubei:
    print(item)
else:
    print("\n")

# 解析确诊数据
total_data = {}
for item in num:
    if item['name'] not in total_data:
        total_data.update({item['name']:0})
    for city_data in item['children']:
        total_data[item['name']] +=int(city_data['total']['confirm'])
print(total_data)
# {'湖北': 48206, '广东': 1241, '河南': 1169, '浙江': 1145, '湖南': 968, ..., '澳门': 10, '西藏': 1}

# 解析疑似数据
total_suspect_data = {}
for item in num:
    if item['name'] not in total_suspect_data:
        total_suspect_data.update({item['name']:0})
    for city_data in item['children']:
        total_suspect_data[item['name']] +=int(city_data['total']['suspect'])
print(total_suspect_data)

# 解析死亡数据
total_dead_data = {}
for item in num:
    if item['name'] not in total_dead_data:
        total_dead_data.update({item['name']:0})
    for city_data in item['children']:
        total_dead_data[item['name']] +=int(city_data['total']['dead'])
print(total_dead_data)

# 解析治愈数据
total_heal_data = {}
for item in num:
    if item['name'] not in total_heal_data:
        total_heal_data.update({item['name']:0})
    for city_data in item['children']:
        total_heal_data[item['name']] +=int(city_data['total']['heal'])
print(total_heal_data)

# 解析新增确诊数据
total_new_data = {}
for item in num:

```

下载数据如下图所示:

## 第二步 绘制地图

第10页 共18页

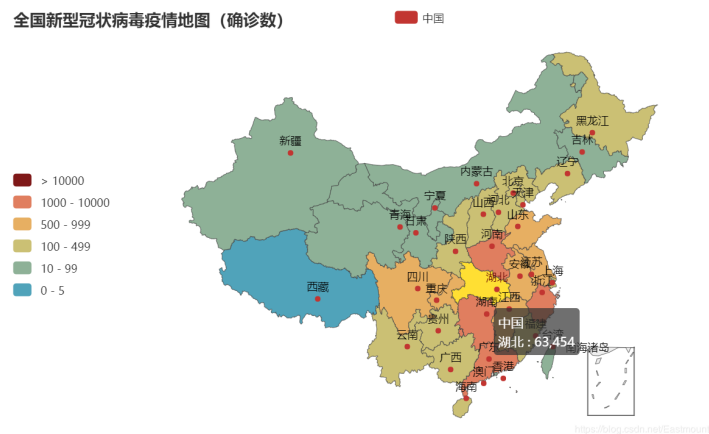
```

list_data = zip(list(data['province']), list(data['confirm']))
print(list_data)
# [('湖北', 48206), ('广东', 1241), ('河南', 1169), ('浙江', 1145), ..., ('澳门', 10), ('西藏', 1)]

#-----
# 第二步: 绘制全国疫情地图
# 参考文章: https://blog.csdn.net/shineych/article/details/104231072 [shineych大神]
#-----
def map_cn_disease_dis() -> Map:
    c = (
        Map()
        .add('中国', list_data, 'china')
        .set_global_opts(
            title_opts=opts.TitleOpts(title='全国新型冠状病毒疫情地图 (确诊数)'),
            visualmap_opts=opts.VisualMapOpts(is_show=True,
                                                split_number=6,
                                                is_piecewise=True, # 是否为分段型
                                                pos_top='center',
                                                pieces=[
                                                    {'min': 10000, 'color': '#7f1818'}, # 不指定 max
                                                    {'min': 1000, 'max': 10000},
                                                    {'min': 500, 'max': 999},
                                                    {'min': 100, 'max': 499},
                                                    {'min': 10, 'max': 99},
                                                    {'min': 0, 'max': 5} ],
                                                ),
        )
    )
    return c
map_cn_disease_dis().render('全国疫情地图.html')

```

输出结果如下图所示:



### 三.PyEcharts绘制贵州省地区

#### 第一步 下载数据

```

# -*- coding: utf-8 -*-
# By: Easmount CSDN xiuzhang

#-----
# 第一步: 抓取数据
#-----
import time, json, requests

```

```

# 抓取腾讯疫情实时json数据
url = 'https://view.inews.qq.com/g2/get0nsInfo?name=disease_h5&callback=&_=%d'%int(time.time()*1000)
data = json.loads(requests.get(url=url).json()['data'])
print(data)
print(data.keys())

# 统计省份信息(34个省份 湖北 广东 河南 浙江 湖南 安徽....)
num = data['areaTree'][0]['children']
print(len(num))

# 获取贵州下标
k = 0
for item in num:
    print(item['name'],end=" ") # 不换行
    if item['name'] in "贵州":
        print("")
        print(item['name'], k)
        break
    k = k + 1
print("") # 换行

# 显示贵州省数据
gz = num[k]['children']
for item in gz:
    print(item)
else:
    print("\n")

#-----
# 第二步: 解析数据
#-----
# 解析确诊数据
total_data = {}
for item in gz:
    if item['name'] not in total_data:
        total_data.update({item['name']:0})
    total_data[item['name']] = item['total']['confirm']
print('确诊人数')
print(total_data)
# {'贵阳': 33, '遵义': 25, '毕节': 22, '黔南州': 17, '六盘水': 10, '铜仁': 10, '黔东南州': 10, '黔西南州': 4, '安顺': 4}

# 解析疑似数据
total_suspect_data = {}
for item in gz:
    if item['name'] not in total_suspect_data:
        total_suspect_data.update({item['name']:0})
    total_suspect_data[item['name']] = item['total']['suspect']
print('疑似人数')
print(total_suspect_data)

# 解析死亡数据
total_dead_data = {}
for item in gz:
    if item['name'] not in total_dead_data:
        total_dead_data.update({item['name']:0})
    total_dead_data[item['name']] = item['total']['dead']
print('死亡人数')
print(total_dead_data)

```



## 第二步 绘制地图

```
# -*- coding: utf-8 -*-
import time, json, requests
import pandas as pd
from pyecharts.charts import Map
import pyecharts.options as opts

#-----
# 第一步: 读取数据
#-----

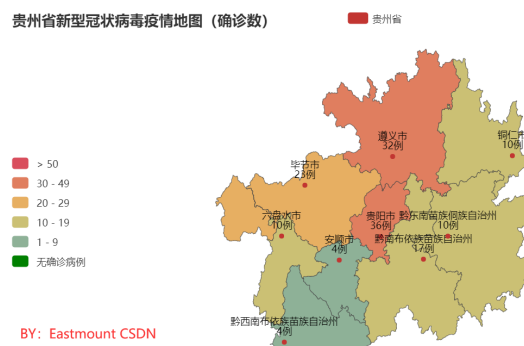
n = time.strftime("%Y-%m-%d") + "-gz.csv"
data = pd.read_csv(n)
list_data_gui Zhou = zip(list(data['province']), list(data['confirm']))
gz_data = list(list_data_gui Zhou)
print(gz_data)

for a,b in gz_data:
    print(a, b, type(b))

#-----
# 第二步: 绘制贵州疫情地图
#-----

def map_gz_disease_dis() -> Map:
    c = (
        Map()
        .add('贵州省', gz_data, '贵州')
        .set_series_opts(label_opts=opts.LabelOpts(is_show=True, formatter='{b}\n{c}例'))
        .set_global_opts(
            title_opts=opts.TitleOpts(title='贵州省新型冠状病毒疫情地图(确诊数)'),
            visualmap_opts=opts.VisualMapOpts(is_show=True,
                                                split_number=6,
                                                is_piecewise=True, # 是否为分段型
                                                pos_top='center',
                                                pieces=[
                                                    {'min': 50},
                                                    {'min': 30, 'max': 49},
                                                    {'min': 20, 'max': 29},
                                                    {'min': 10, 'max': 19},
                                                    {'min': 1, 'max': 9},
                                                    {'value': 0, "label": '无确诊病例', "color":
'green'} ],
                                                )
        )
    )
    return c
map_gz_disease_dis().render('贵州省疫情地图.html')
```

输出结果如下图所示:





<https://blog.csdn.net/Eastmount>

注意，读者可能会绘制图形无数据，如下图所示显示NaN。这是因为我们需要补充城市的完整称呼，比如“贵阳市”、“黔东南苗族侗族自治州”等。

province	confirm	dead	heal	new_confirm
贵阳市	36	1	18	0
遵义市	32	0	18	0
毕节市	23	0	13	0
黔南布依族苗族自治州	17	0	12	0
黔东南苗族侗族自治州	10	0	8	0
铜仁市	10	0	10	0
六盘水市	10	1	6	0
安顺市	4	0	1	0
黔西南布依族苗族自治州	4	0	4	0



<https://blog.csdn.net/Eastmount>

### 第三步 绘制柱状图

```

# -*- coding: utf-8 -*-
import time, json, requests
import pandas as pd
from pyecharts.charts import Bar
import pyecharts.options as opts

#-----
# 第一步：读取数据
#-----

n = time.strftime("%Y-%m-%d") + "-gz.csv"
data = pd.read_csv(n)
province_list = list(data['province'])
confirm_list = list(data['confirm'])
dead_list = list(data['dead'])
heal_list = list(data['heal'])
new_confirm_list = list(data['new_confirm'])
print(province_list)          # 地区
print(confirm_list)           # 确诊数据
print(dead_list)              # 死亡数据
print(heal_list)              # 治愈数据
print(new_confirm_list)       # 新增确诊

#-----
# 第二步：绘制贵州柱状图
#-----

bar=(
    Bar()
    .add_xaxis(province_list)

```

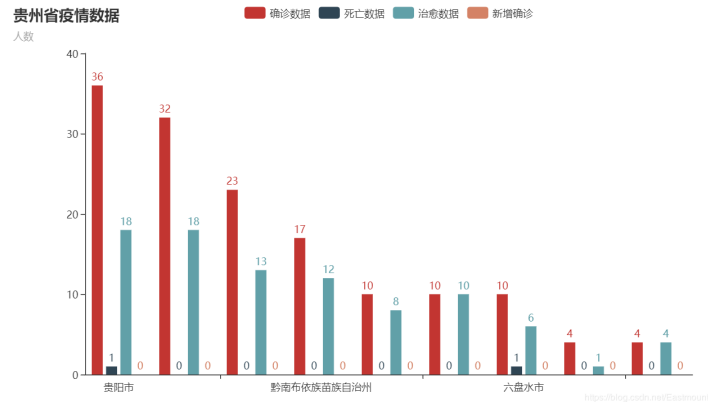
```

        .add_yaxis("确诊数据", confirm_list)
        .add_yaxis("死亡数据", dead_list)
        .add_yaxis("治愈数据", heal_list)
        .add_yaxis("新增确诊", new_confirm_list)
        .set_global_opts(title_opts=opts.TitleOpts(title="贵州省疫情数据", subtitle="人数"))
    )

    bar.render("贵州省疫情2.html")

```

输出结果如下图所示：



## 四.PyEcharts绘制其他图形

更多图形推荐读者结合自己的项目或论文进行实践，这里不再详细补充。后续作者可能也会分享相关知识。比如我们可以对数据进行简单预处理，再进行可视化分析。

```

# -*- coding: utf-8 -*-
import time, json, requests
import pandas as pd
from pyecharts import options as opts
from pyecharts.charts import Bar

# -----
# 第一步：读取数据
# -----

n = time.strftime("%Y-%m-%d") + "-china-bj.csv"
data = pd.read_csv(n)
province_list = list(data['province'])
confirm_list = list(data['confirm'])
dead_list = list(data['dead'])
heal_list = list(data['heal'])
new_confirm_list = list(data['new_confirm'])
print(province_list)          # 地区
print(confirm_list)           # 确诊数据
print(dead_list)              # 死亡数据
print(heal_list)              # 治愈数据
print(new_confirm_list)       # 新增确诊

# -----
# 第二步：绘制全国箱图
# -----

bar=(
    Bar()

```

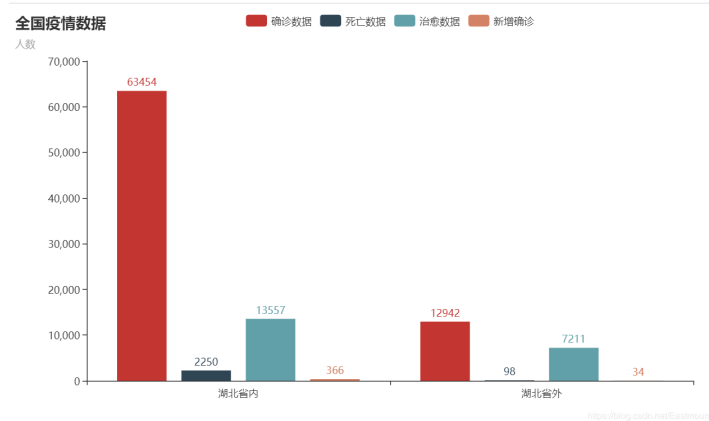


```

        .add_xaxis(province_list)
        .add_yaxis("确诊数据", confirm_list)
        .add_yaxis("死亡数据", dead_list)
        .add_yaxis("治愈数据", heal_list)
        .add_yaxis("新增确诊", new_confirm_list)
        .set_global_opts(title_opts=opts.TitleOpts(title="全国疫情数据", subtitle="人数"))
    )

bar.render("全国疫情柱状图.html")

```



## 五.总结

写到这里，第二篇疫情分析的文章就讲解完毕，希望对您有所帮助。主要包括两部分内容：

- 实时数据爬取
- PyEcharts可视化分析
- 中国地图绘制、贵州省地图绘制
- 柱状图及折线图

后续还会分享GIS地图显示、情感分析、舆情分析、主题挖掘、威胁情报溯源、知识图谱、预测预警及AI和NLP应用等。如果文章对您有所帮助，将是我写作的最大动力。作者将源代码上传至github，大家可以直接下载。

此电脑 > 本地磁盘 (D:) > [重点] 博客开源代码 > Wuhan+DA > blog02-pyecharts				
名称	修改日期	类型	大小	
2019-nCoV疫情曲线图2.html	2020/2/22 21:03	360 Chrome HTML ...	26 KB	
2019-nCoV疫情曲线图3.html	2020/2/22 21:09	360 Chrome HTML ...	28 KB	
2020-02-22-china.csv	2020/2/22 21:14	XLS 工作表	1 KB	
2020-02-22-china-bj.csv	2020/2/22 21:47	XLS 工作表	1 KB	
2020-02-22-daily.csv	2020/2/22 19:45	XLS 工作表	2 KB	
2020-02-22-gz.csv	2020/2/22 21:27	XLS 工作表	1 KB	
test01-spider.py	2020/2/22 19:42	Python File	3 KB	
test02-pyecharts-zhu.py	2020/2/22 20:53	Python File	2 KB	
test03-pyecharts-zhexian.py	2020/2/22 21:03	Python File	3 KB	
test04-pyecharts-zhexian.py	2020/2/22 21:09	Python File	4 KB	
test05-spider-china.py	2020/2/22 21:14	Python File	4 KB	
test06-pyecharts-mapchina.py	2020/2/22 21:17	Python File	3 KB	
test07-spider-guizhou.py	2020/2/22 21:20	Python File	4 KB	
test08-pyecharts-guizhou.py	2020/2/22 21:29	Python File	3 KB	
test09-pyecharts-guizhou.py	2020/2/22 21:35	Python File	2 KB	
test10-pyecharts-china.py	2020/2/22 21:47	Python File	2 KB	
贵州省疫情2.html	2020/2/22 21:35	360 Chrome HTML ...	7 KB	
贵州省疫情地图.html	2020/2/22 21:27	360 Chrome HTML ...	6 KB	
全国疫情地图.html	2020/2/22 21:17	360 Chrome HTML ...	8 KB	
全国疫情柱状图.html	2020/2/22 21:47	360 Chrome HTML ...	6 KB	

同时，向钟院士致敬，向一线工作者致敬。侠之大者，为国为民。咱们中国人一生的最高追求，为天地立心，为生民立命，为往圣继绝学，为万世开太平。以一人之力系万民康乐，以一身犯险保大业安全。他们真是做到了，武汉加油，中国加油！





(By:Eastmount 2020-02-22 晚上10点夜于贵阳 <http://blog.csdn.net/eastmount/>)

#### 参考文献：

- [1] [Python可视化] pyecharts安装入门及绘制中国贵州地图 - 杨秀璋
- [2] [Echarts可视化] 一.入门篇之简单绘制中国地图和贵州地区 - 杨秀璋
- [3] <https://news.qq.com/zt2020/page/feiyun.htm>
- [4] [Python疫情大数据分析] 一.腾讯实时数据爬取、Matplotlib和Seaborn可视化分析全国各地区、某省各城市、新增趋势
- [5] Python实战：抓肺炎疫情实时数据，画2019-nCoV疫情地图 - 许老师
- [6] 用Python抓新型冠状病毒肺炎疫情数据，绘制全国疫情分布图 - shineych老师
- [7] 2020Python开发者日：爬虫框架的技术实现与模块应用的经验分享 - 许老师
- [8] 用Python pyecharts v1.x 绘制图形（二）：折线图、折线面积图、散点图、雷达图、箱线图、词云图 - shineych老师
- [9] pyecharts v1版本 学习笔记 折线图面积图 - baili-luoyun

👍 点赞 38    ☆ 收藏    🔄 分享



Eastmount    博客专家

发布了452 篇原创文章 · 获赞 6414 · 访问量 505万+