

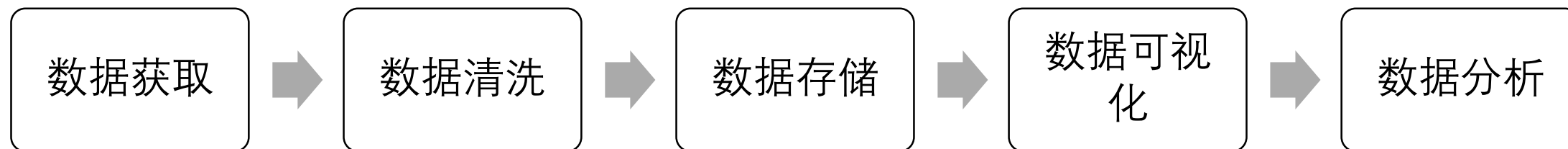
2019-nCoV Analysis

作者：张战罗

2020年2月21日

2019年12月，新型冠状病毒在武汉爆发，进而波及到全国甚至全世界，给人们的生活带来了极大的影响。在防疫过程中，需要及时地对疫情分布进行分析，以实现更加有效的防控措施。同时，将数据通过好的可视化方式展现给大众，防止大众恐慌，对于防疫过程也非常有帮助。

本项目结合数据工程的基本过程，对疫情分布进行了可视化。



数据获取是指通过技术手段获取数据，是数据工程的第一步。数据的质量对于数据工程而言至关重要。数据获取环节由以下三个步骤组成：

数据来源寻找

数据来源评估

数据下载

数据来源寻找：根据背景不同，数据工程的数据可能来源于多种不同的渠道，常见的有传感器系统、公开的数据集以及相关网站等类型。

数据来源评估：对于数据来源，要从数据可靠性、数据可获取性等方面进行评估。例如在本例中，数据可以从国家卫健委官方的通报中获取，可以从丁香园网站获取，也可以从已有的GitHub项目中获取。从数据的可靠性和可获取性来说，丁香园网站因其标准的格式，比较高的可靠性，适合作为数据下载的来源。

数据下载：对于不同的数据来源有不同的数据下载方式。对于传感器系统，需要建立采集系统对数据进行采集传输；对于一些公开的数据及或网站，可以直接去相关页面下载。若想自动实时获取最新数据，就需要借助一定的工具来自动实现这一过程，而这一工具也通常被人们称之为爬虫。

数据获取是指通过技术手段获取数据，是数据工程的第一步。数据的质量对于数据工程而言至关重要。数据获取环节由以下三个步骤组成：

数据来源寻找

数据来源评估

数据下载

在本例中，使用了爬虫从丁香园的网站（<https://ncov.dxy.cn/ncovh5/view/pneumonia>）中自动获取最新的数据。使用了requests包来完成这一过程。

代码示例：

```
# requests会自动访问给定的连接并获取网页的内容。  
r = requests.get('https://ncov.dxy.cn/ncovh5/view/pneumonia')
```

获取的数据往往不能直接使用，例如通过传感器系统获取的数据可能会存在缺失值、异常值的情况；从网页获取的数据以特定的格式混杂于整个网页内容中，需要使用一定的技术手段提取出需要的数据。

在本例中，使用了BeautifulSoup包对爬虫获取的html文件进行处理，提取有效地数据并根据需要进行了重新组织。

代码示例：

```
r.encoding = 'utf-8'
soup = BeautifulSoup(r.text, 'html.parser')
area_stat_raw = soup.find(id='getAreaStat').get_text()
area_stat_js = area_stat_raw[len('try { window.getAreaStat = '): -len('}catch(e){}')]
area_stat = json.loads(area_stat_js)

# 进行省级和市级统计
province_name = []
p_current_confirmed_count = []
for province in area_stat:
    province_name.append(province['provinceShortName'])
    p_current_confirmed_count.append(province['currentConfirmedCount'])
```

清洗好的数据需要存储以便于使用。数据可以存储于本地，利用以text、csv、excel等形式存储于电脑硬盘中，也可以存储于数据库中并存放于本地或云端。最常见得关系型数据管理系统是MYSQL。

本例所建的可视化项目并未实现实时更新的功能，因此数据直接进入下一环节，未对数据进行存储。

代码示例：

无

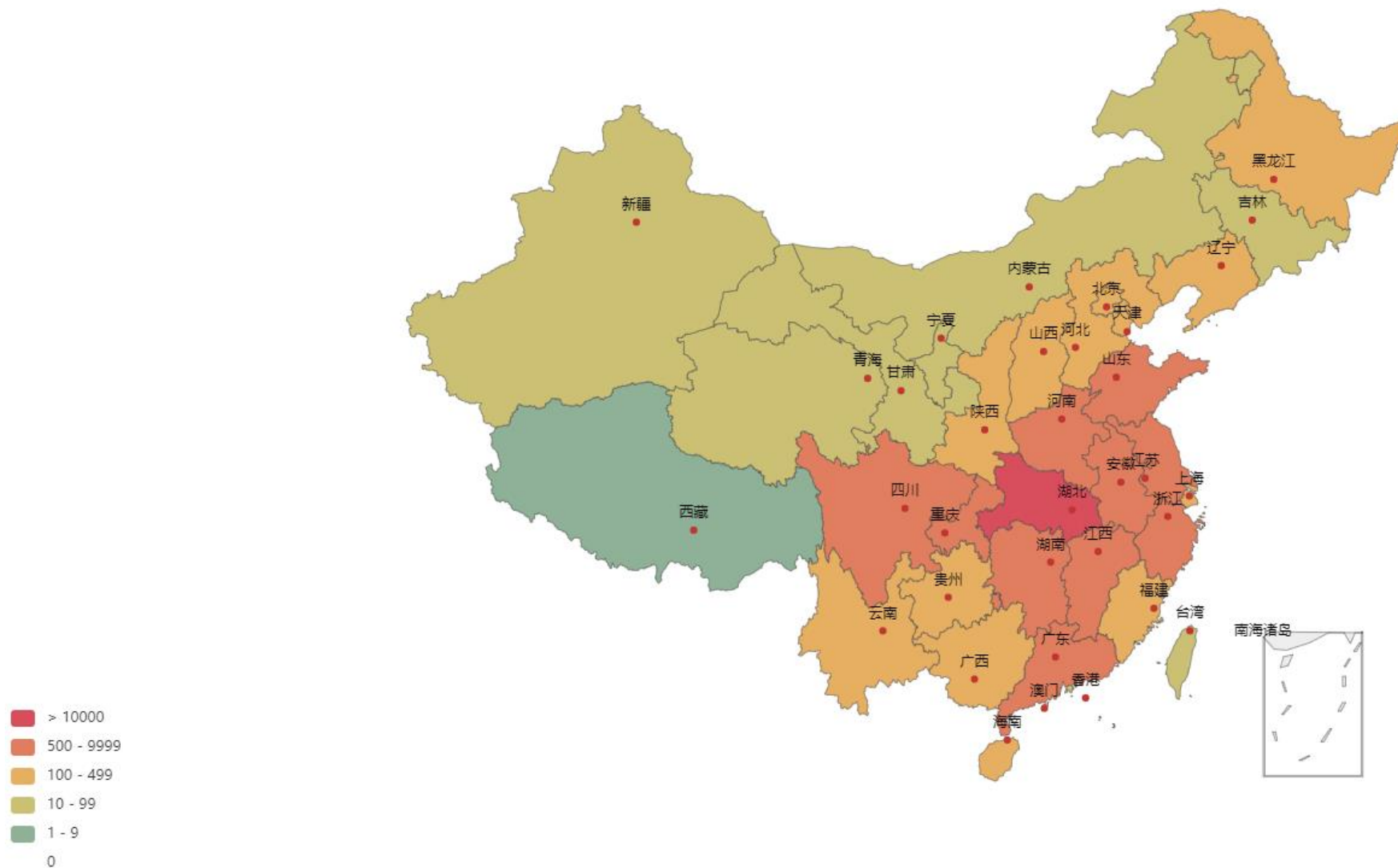
数据可视化有利于对于数据的展示和理解。数据可视化的形式多种多样，随着数据工程的发展，相关的工具也日益丰富和强大，例如百度开源的echarts项目（<https://www.echartsjs.com/zh/index.html>）具有丰富的可视化模板。基于echarts，陈键冬等人开发了适合python使用的工具pyecharts（<https://github.com/pyecharts/pyecharts>）。

本例使用pyrcharts实现了数据可视化。

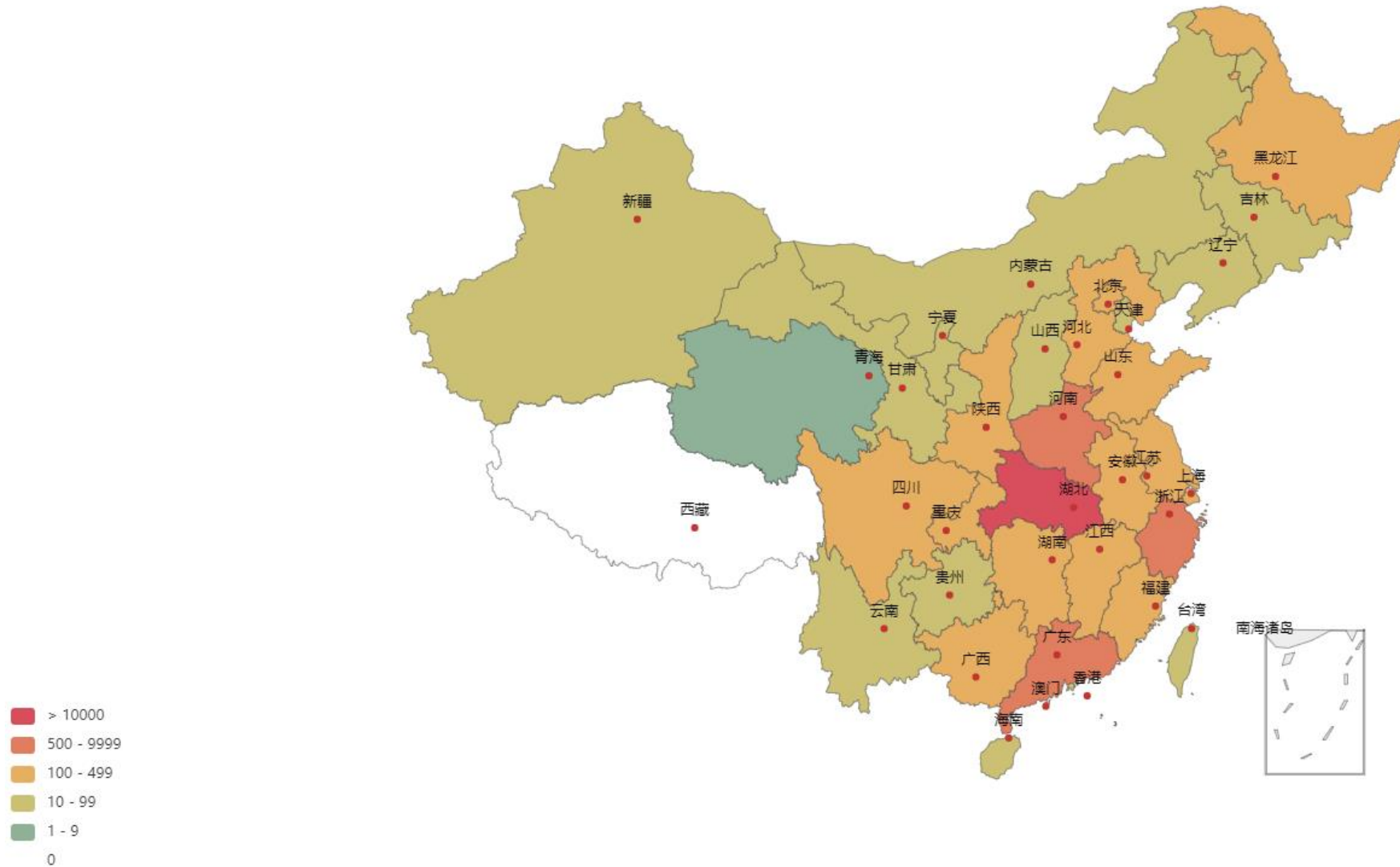
代码示例：

```
chart = Map(init_opts=opts.InitOpts(width='1500px', height='800px'))
chart.add('现存确诊', [list(z) for z in zip(province_name, p_current_confirmed_count)], 'china')
chart.set_global_opts(toolbox_opts=opts.ToolboxOpts(is_show=True, pos_top='20px'),
                      title_opts=opts.TitleOpts(title='2019-nCoV疫情地图： {} ({} )'.format('现存确诊', time.asctime()),
                                                  pos_left='center', pos_top='20px'),
                      legend_opts=opts.LegendOpts(is_show=False),
                      visualmap_opts=opts.VisualMapOpts(is_pieewise=True,
                                                         pieces=[{'min': 10000},
                                                                {'min': 500, 'max': 9999},
                                                                {'min': 100, 'max': 499},
                                                                {'min': 10, 'max': 99},
                                                                {'min': 1, 'max': 9},
                                                                {'max': 0}]))
```

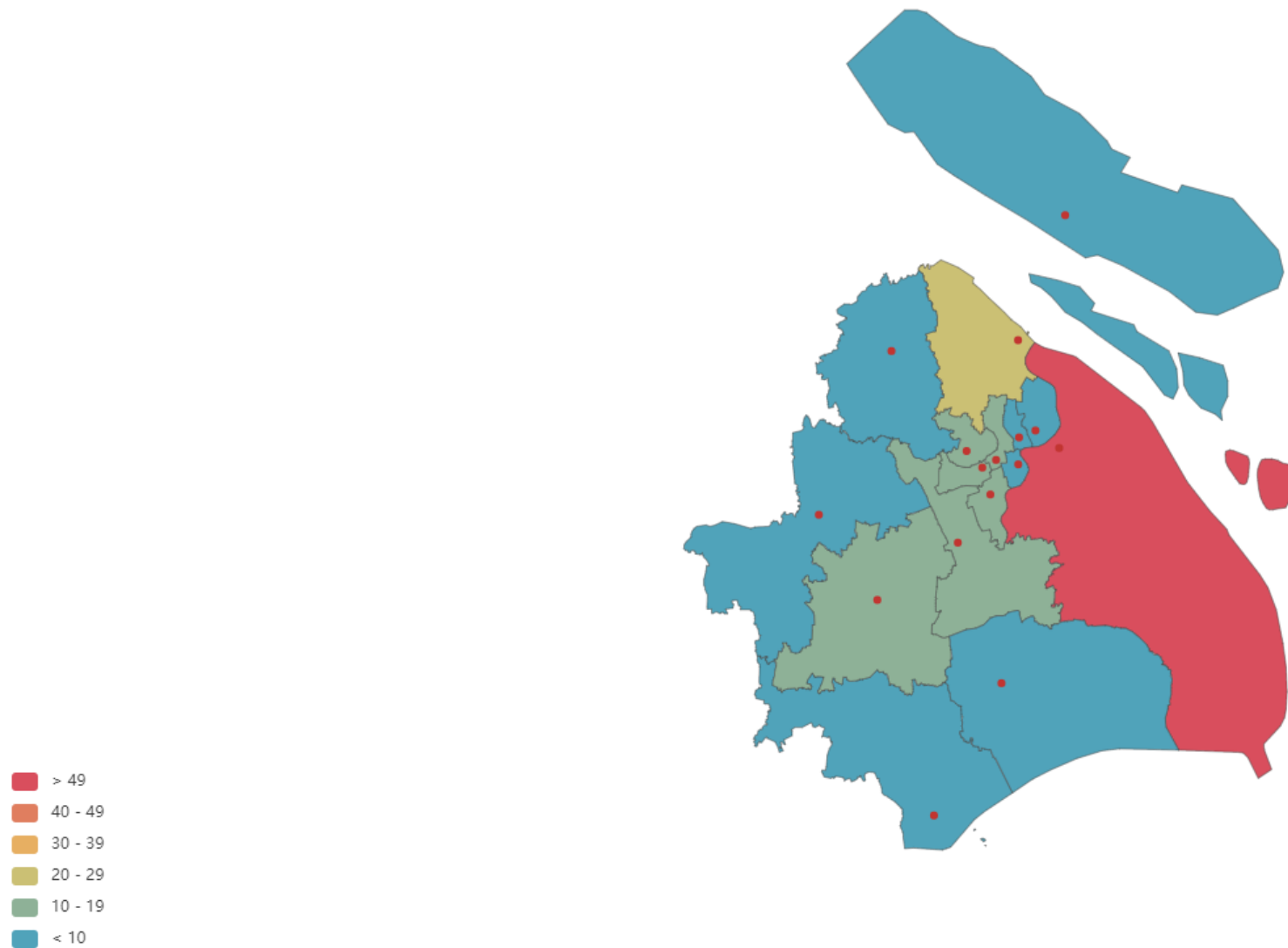

2019-nCoV疫情地图：累计确诊 (Fri Feb 21 11:47:39 2020)



2019-nCoV疫情地图：现存确诊 (Fri Feb 21 11:47:39 2020)



2019-nCoV疫情地图：累计确诊（上海市，Fri Feb 21 11:47:39 2020）



结合机器学习或者传统的统计学方法，可以对数据进行进一步分析以获取更多有价值的信息。例如对数据建模以分析疫情扩展的规律，并对疫情发展做出预测，对疫情防控措施进行评估。

本例未对数据进行进一步分析，可以参考以下两篇文章了解数据进一步分析的一般方法和结果。

- [最新！交大医学院团队分阶段估计武汉市新冠肺炎疫情趋势](#)
- [真·学霸！疫情何时结束？交大300多名学子做了这件事…](#)

代码示例：

无