基于 fmepython 和 arcpy 的算法设计和应用开发

引言

信息化和大数据时代给我们提供了海量的数据源，而 FME 平台和自然开发语言 python 让数据挖掘增添了无限可能，使数据真正地为我所用。本文基于 fmepython 和 arcpy 设计了基本图生产算法，实现了数据从裁剪到服务发布和切片的自动化。同时针对 COVID-19 聚合数据，设计了从数据爬取到数据可视化以及轻量级 web 应用发布和集成展示一整套算法及流程。

1 “三驾马车”如何并驾齐驱取长补短

在 FME Desktop 中的关键平台 FME Workbench 中，结合 fmepython 主要能够实现以下几个方面的功能：①利用 python 脚本后台调用 fme 完成模板运行；②结合 PythonCreator 和 PythonCaller，实现要素的创建以及现有要素的批量流程化操作；③在启动和关闭模板时触发相应 python 脚本完成启动和关闭的辅助工作，例如启动时自动建立源数据备份。

ArcGIS API for Python 即 Arcpy ，ArcPy 是一个 Python 站点包，可提供以实用高效的方式通过 Python 执行地理数据分析、数据转换、数据管理和地图自动化。作为测绘地理信息的从业人员，ArcGIS 桌面端软件基本上是装机必备。由于 ArcMap 仍然是 32 位内核驱动，所以安装 ArcMap 时自带的 python2.7 为 32 位。64 位 python2.7 需要补充安装 ArcGIS Desktop BackgroundGP 工具。

Anaconda 是一个用于科学计算的开源 Python 发行版本，提供跨平台的包管理与环境管理的功能，可以很方便地解决多版本 python 并存、切换以及各种第三方包安装问题。Anaconda 作为一只“咬着自己尾巴的蟒蛇”，为 python 初学者提供了众多 python 科学包的集成闭合环式管理。Anaconda 安装时提供了诸如 JupyterLab、NoteBook 和 Spyder 等常用的 IDE 方便用户使用，其他 IDE 例如 PyCharm 也支持对 Anaconda 的配置使用。Anaconda 中基于 conda 设计的“万物皆是 package”的产品理念，与后面要介绍的“一切皆可配置”的 pyecharts 不谋而合，代表也引领者当前最流行的软件设计基本思路。

“三驾马车”在各自最擅长的领域里大放异彩。在这“三驾马车”的整合方面，在 FME Workbench 中可以实现 fmepython，arcpy 和 anaconda 的集成，只是在模板调试时配置 python 编译器分别指向 anaconda 和 arcpy 即可。由于 fmepython 尚未封装为独立包，脱离 FME Workbench 开发平台和环境之后，无法通过调用 fmepython 和 FME Workbench 中转换器。目前通过 PyCharm 等 IDE 支持 arcpy 和 anaconda 集成，不能调用 fmepython。

图 1 “三驾马车”关系图

2 基本图生产之自动化切片和服务发布

在工程项目生产过程中，拿到了一批尚未分幅而且缺少图幅结合表的基本图 dwg 数据，要求发布成栅格形式的切片地图服务。整个生产流程包括基本图分幅，镶嵌数据集构建以及服务发布和切片制作。

AutoCAD 平台对 dwg 数据的渲染效果，在目前的测绘地理信息生产模式下，尚没有其他平台可以完全替代。构建镶嵌数据集的原始数据是分幅后的 dwg 数据打印形成的 png 图像和 pgw 配置文件，此步骤无法摆脱 AutoCAD 平台。因此本次生产的技术路线是用 FME 完成基本图分幅工作，用 AutoCAD 实现 dwg 文件的批量打印生产 png 和 pgw，然后结合 arcpy 实现镶嵌数据集构建、服务发布和切片制作的自动化。

2.1 基本图分幅

基本图分幅过程中分别使用 FeatureReader 读取待裁剪的 dwg 数据和图幅结合表数据。CoordinateExtractor 提取结合表的 x 和 y 坐标，为写入到裁剪后的 dwg 数据文件名中做准备。Clipper 裁剪过程中融合写入数据流的属性信息，在写出 dwg 数据时，设置 fanout 属性，按照文件名严格控制输出的每个图幅内容，同时记录 x 坐标和 y 坐标到 dwg 文件名中。在 CAD 平台中开发了对 dwg 分幅内容批量打印输出的工具，打印时结合文件名中的 x 和 y 坐标构建正方形窗口确保打印内容的范围准确性。此部分内容不是本文重点，此处不再赘述。

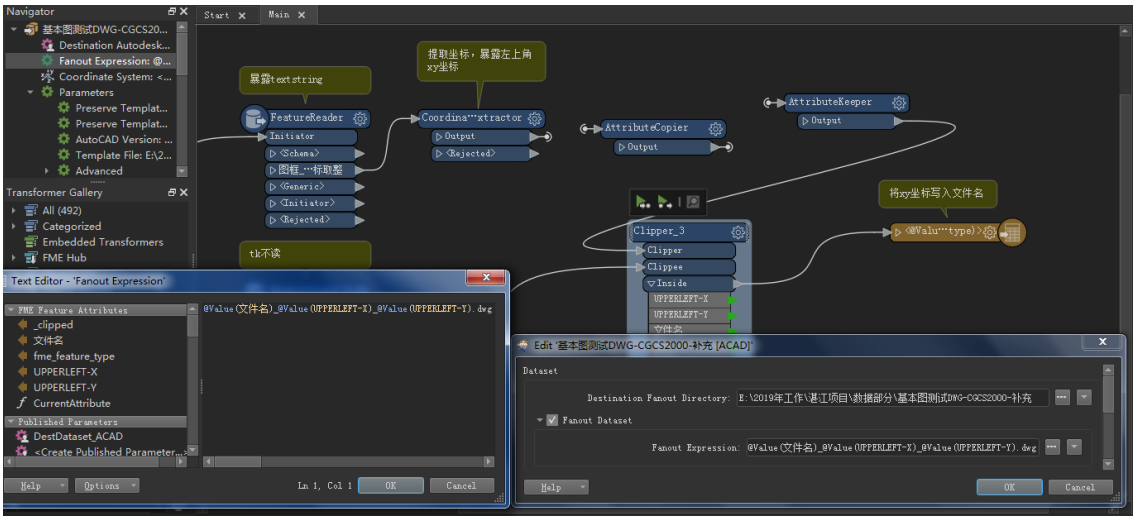


图 2　基本图分幅模板

2.2 镶嵌数据集构建

由于镶嵌数据集构建设计到用户的交互，此处结合 arcpy 在 ArcMap 中搭建了一个可视化的脚本工具。用户输入 gdb 路径和镶嵌数据集名称，选择坐标系和 png 以及 pgw 数据所在原始文件夹，arcpy 中接入用户输入的参数后，调用创建镶嵌数据集工具实现镶嵌数据集创建，同时调用工具实现在镶嵌数据集中添加包含坐标和分辨率信息的 png 栅格数据。

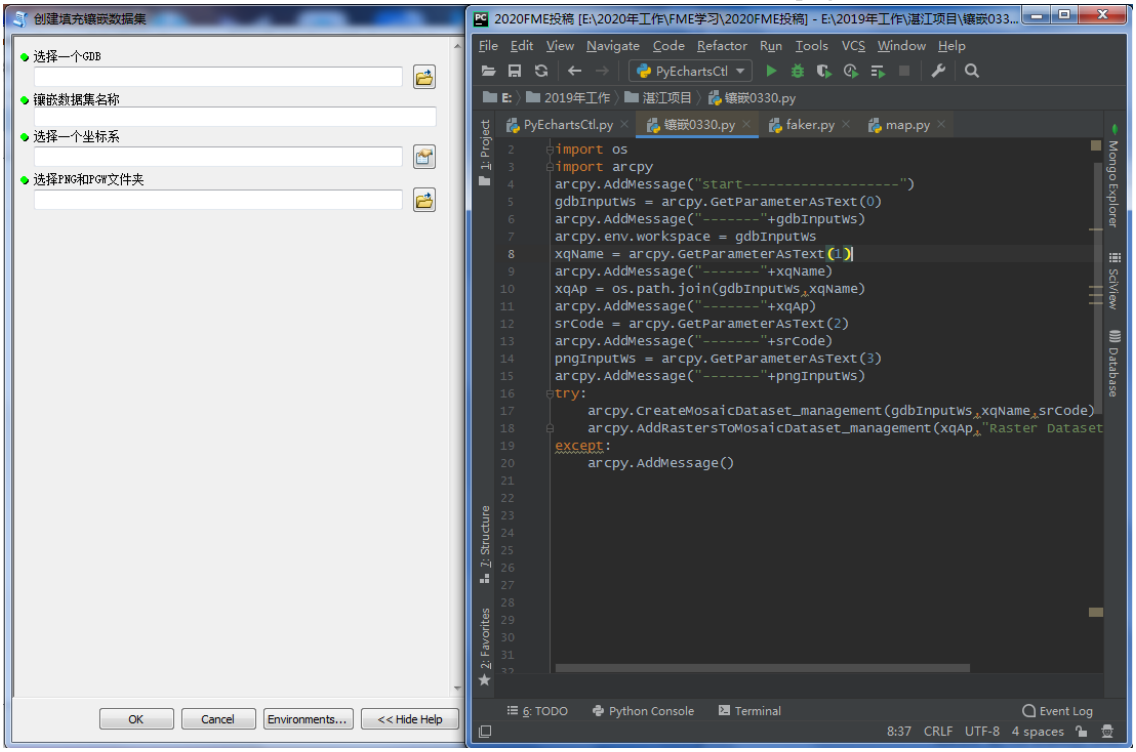


图 3　基于 arcpy 构建镶嵌数据集

2.3 服务发布和切片制作

镶嵌数据集构建完成后，在 FME 的 PythonCreator 中实现了基本图服务发布和切片制作。整个过程主要包括：1.制作地图文档；2.发布地图服务；3.制作地图服务器缓存；4.生成瓦片。在 FME 中接入 arcpy 时需要优先设置 python 编译器为 Esri ArcGIS Python 2.7。PythonCreator 和 PythonCaller 提供了一个 fmepython 环境下的 IDE，虽然这个 IDE 在调试时不是那么得心应手，但是这的确是最能整合“三驾马车”各自优势的 IDE 和解决方案了。具体设计实现的过程中，使用 arcpy 首先将镶嵌数据集添加到 mxd 中，然后指定服务器创建一个栅格数据服务的 draft，设置缓存格式和切片分辨率等参数后，制作地图服务器缓存，最后调用地图服务缓存管理工具生成瓦片。至此，整个自动化的生产过程搭建完成。

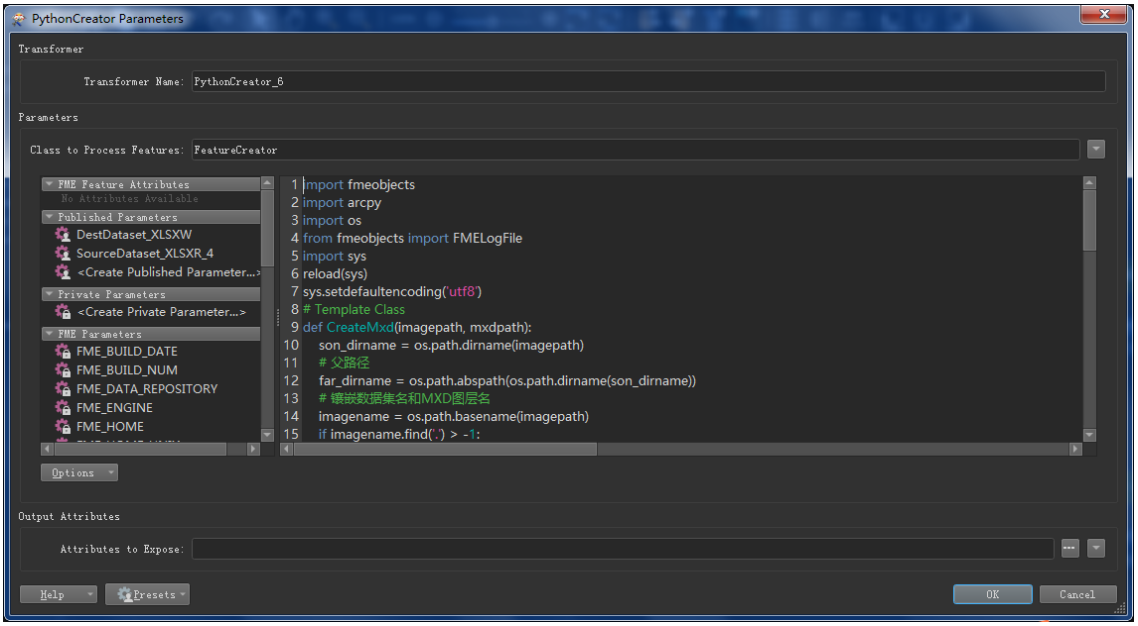


图 4　服务发布和切片制作

3 网络爬虫之疫情聚合数据抓取

结合 fmepython 进行测绘地理信息数据生产只是 FME 强大功能的“冰山一角”。COVID-19 作为一个席卷全球的大流行病，注定成为这个时代无法忽视和逾越的话题。基于 fmepython 和 anaconda 这个“包工头”，本文设计了疫情聚合数据抓取的两种技术路线：1.直接在 PyCharm 中调用 anaconda 抓取数据，保存为 xls 格式或其他格式存到本地；2.在 FME Workbench 调用 anaconda 抓取数据，结合 fmepython 和其他转换器进行数据预处理和输出。以技术路线 2 为主线，横向对比与技术路线 1 的相同和不同之处。

技术路线 2 爬取疫情数据，已有众多 FME 爱好者进行了研究和尝试，分别从丁香园和腾讯网爬取疫情专题数据。除此之外，也可从一些 python 大牛发布出来的网页上直接请求疫情数据。数据源不同，解析时方法略有不同，但是核心步骤基本如下：1.网页端获取数据；2.json

数据标准化；3.解析 json 数据；4.以 xls 或其他格式输出到本地。网页端获取数据使用 HTTPCaller 和 HTTPExtractor 转换器，类似于 PyCharm 中调用 requests 包获取网页信息。JSONFragmenter 转换器解析 json 数据的思路与 PyCharm 中调用 json 包一致。数据输出方面，FME 直接写出写 xls 模块，PyCharm 中则调用 pandas 进行表格数据输出。

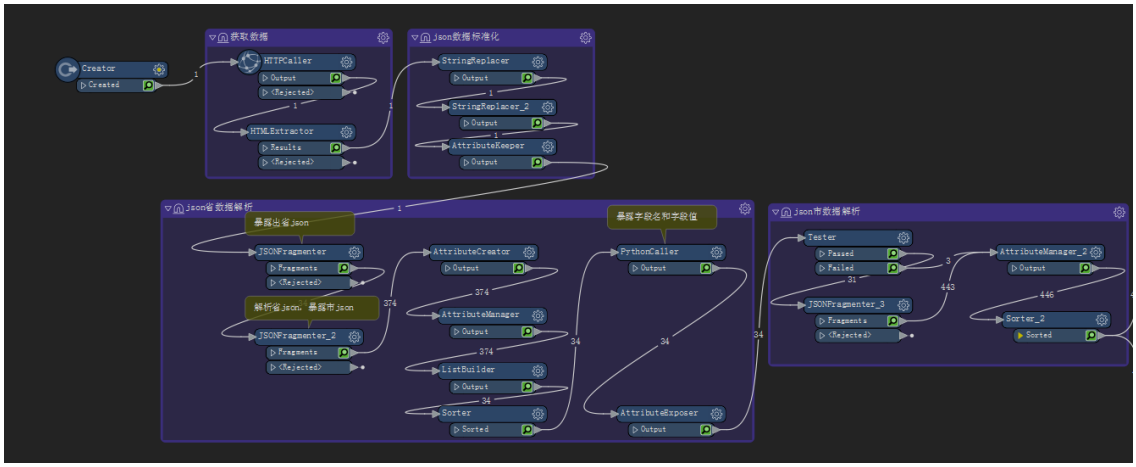
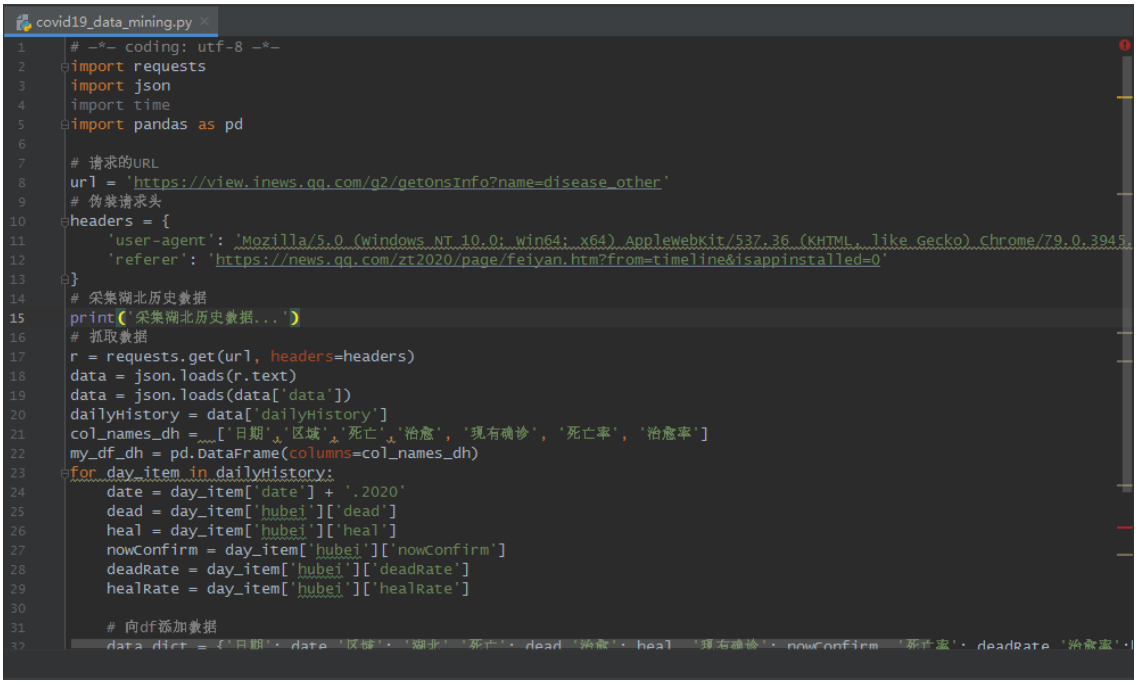


图 5　FME Workbench 爬取疫情数据



```
covid19_data_mining.py
# -*- coding: utf-8 -*-
import requests
import json
import time
import pandas as pd

# 请求的URL
url = 'https://view.inews.qq.com/g2/getOnsInfo?name=disease_other'
# 伪装请求头
headers = {
    'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) Applewebkit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.
    'referer': 'https://news.qq.com/zt2020/page/feiyan.htm?from=timeline&isappinstalled=0'
}
# 采集湖北历史数据
print('采集湖北历史数据...')
# 抓取数据
r = requests.get(url, headers=headers)
data = json.loads(r.text)
data = json.loads(data['data'])
dailyHistory = data['dailyHistory']
col_names_dh = ['日期','区域','死亡','治愈', '现有确诊', '死亡率', '治愈率']
my_df_dh = pd.DataFrame(columns=col_names_dh)
for day_item in dailyHistory:
    date = day_item['date'] + '.2020'
    dead = day_item['hubei']['dead']
    heal = day_item['hubei']['heal']
    nowConfirm = day_item['hubei']['nowConfirm']
    deadRate = day_item['hubei']['deadRate']
    healRate = day_item['hubei']['healRate']

    # 向df添加数据
    data_dict = {'日期': date, '区域': '湖北', '死亡': dead, '治愈': heal, '现有确诊': nowConfirm, '死亡率': deadRate, '治愈率':
```

图 6　PyCharm 爬取疫情数据

4 数据可视化之疫情数据可视化

2020 年 1 月 22 日，在大武汉“封城”前一天，约翰霍普金斯大学（JHU）两名博士生和导师一起，基于 ArcGIS Dashboard 开发了一个可视化、可交互的全球疫情地图。该网站发布在 ArcGIS Online 上，自上线以来网站日访问量从 2 亿次上升到三月初的 12 亿次，最高达到

了 20 亿次访问。在国外的码云 GitHub 上收藏量巨大，也登上了 GitHub 热搜榜第一。作为一个在行业耕耘多年，且与武汉 CDC 一度保持深度联系的 GIS 从业者，我在兴奋的同时也感到十分遗憾。兴奋是因为看到基于 ArcGIS 搭建的产品获得如此巨大的成功，而遗憾在于自己没有把握住这样用技术反哺武汉甚至全世界的机会。

疫情数据可视化的手段是多种多样的，使用 FME 平台结合 python 开发语言和 pyecharts 也可以搭建出轻量级的酷炫展示效果。Echarts 是一个由百度开源的数据可视化，凭借着良好的交互性，精巧的图表设计，得到了众多开发者的认可。而 Python 是一门富有表达力的语言，很适合用于数据处理。将数据分析与数据可视化结合，pyecharts 的用武之地完全凸显出来了。pyecharts 中“一切皆可配置”，用户通过设置全局配置项和系统配置项等，可以轻松便捷地实现 pyecharts 的初始化和集成展示。

4.1 Workbench 疫情可视化

FME Workbench 本身具备数据可视化的能力，实际上从 FME2019 版开始 Workbench 中集成 Data Inspector，也体现了 FME 产品未来的发展趋势。FME Workbench 中有 HTMLReportGenerator 和 ChartGenerator 两个转换器可以用于数据可视化展示。HTMLReportGenerator 转换器支持生成 html 前端页面，页面中可以嵌入折线图 Line，饼状图 Pie，柱状图 Bar 和地图 Map，可以结合具体需要进行多种表现形式的展示。ChartGenerator 转换器直接生成一个统计图的栅格数据另存为 jpg 或者 pdf 格式。

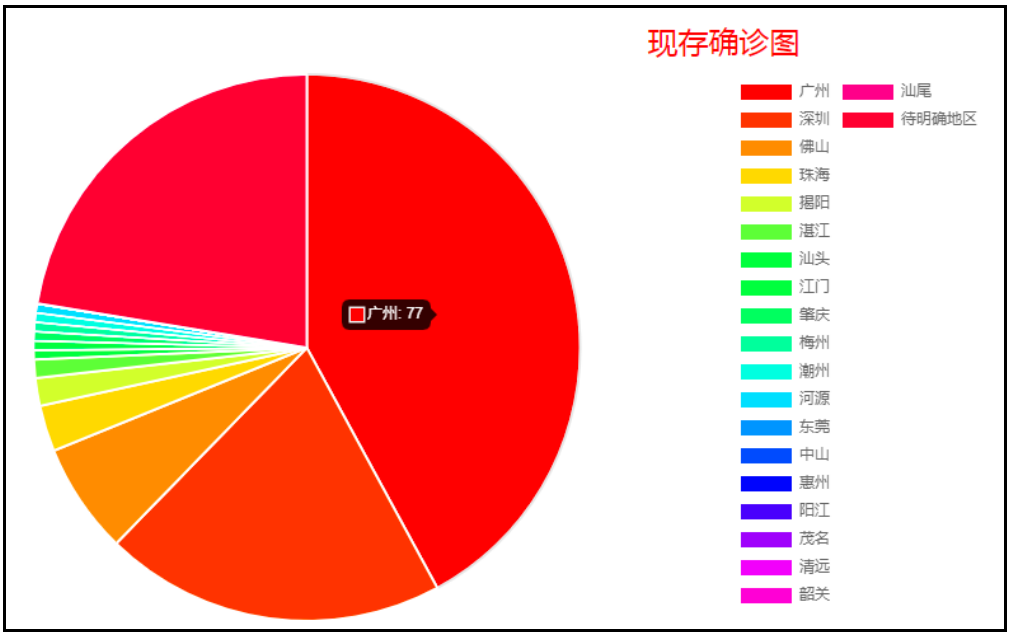


图 7　Workbench 实现疫情数据可视化（2020 年 4 月 12 日）

4.2 pyecharts 疫情可视化

工欲善其事，必先利其器。结合 JHU 提供的疫情数据，使用 FME Workbench 进行预处理，再使用 pyecharts 进行可视化。本文使用 2020 年 1 月 22 日至 2020 年 2 月 29 日和 2020 年 3 月 1 日至 2020 年 4 月 1 日的疫情数据，分别制作了全国 COVID-19 疫情时序变化地图和湖北省 COVID-19 疫情时序变化地图。地图由中间的地图组件 Map 和底部的时间轴组件 Timeline 组成。地图左下角显示地图图例。

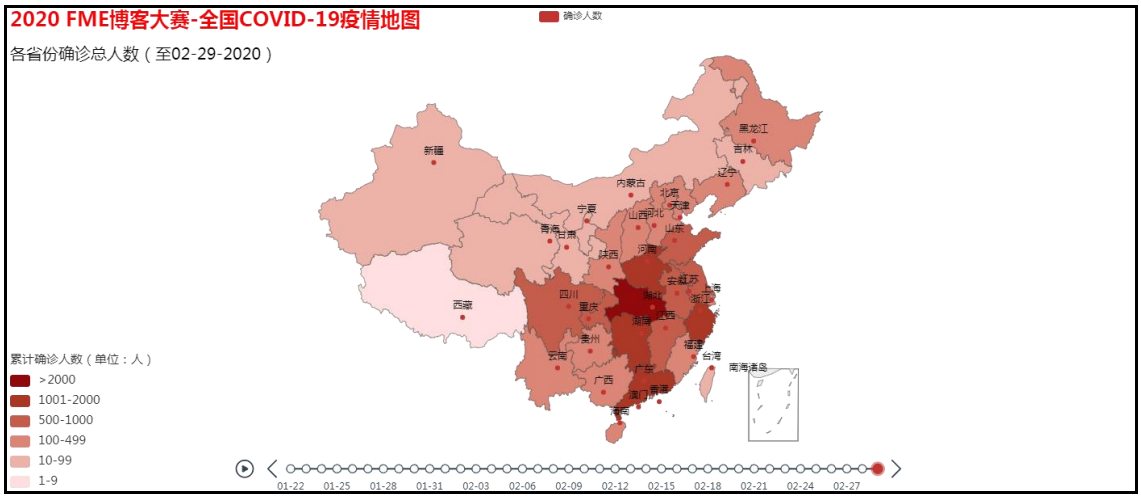


图 8　全国 COVID-19 疫情时序变化地图（至 2020 年 2 月 29 日）

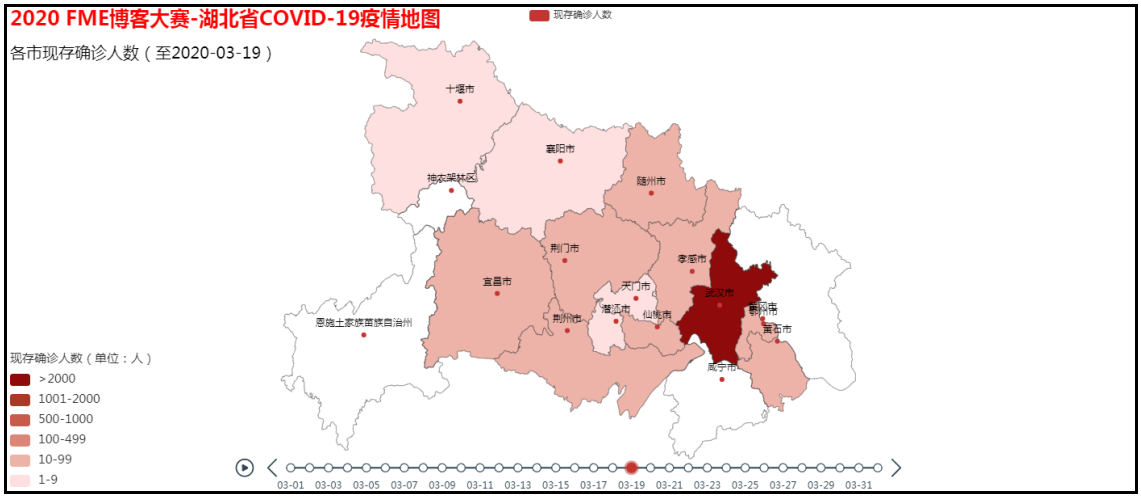


图 9　湖北省 COVID-19 疫情时序变化地图（至 2020 年 4 月 1 日）

4.3 streamlit 轻量级 Web 应用

FME 的用户群相当一大部分都侧重于后端的开发和应用，对于前段编程与设计经验相对欠缺，因此如何快速地搭建一个交互式的 Web 应用展示数据分析和可视化成果往往变得比较棘手。Streamlit 是第一个专门针对机器学习和数据科学团队的应用开发框架，它是开发自定

义机器学习工具的最快的方法。使用 Streamlit，寥寥数语就搭建好了一个轻量级的 WebAPP，快速高效地集成展示了前面生成的 COVID-19 疫情时序变化地图。

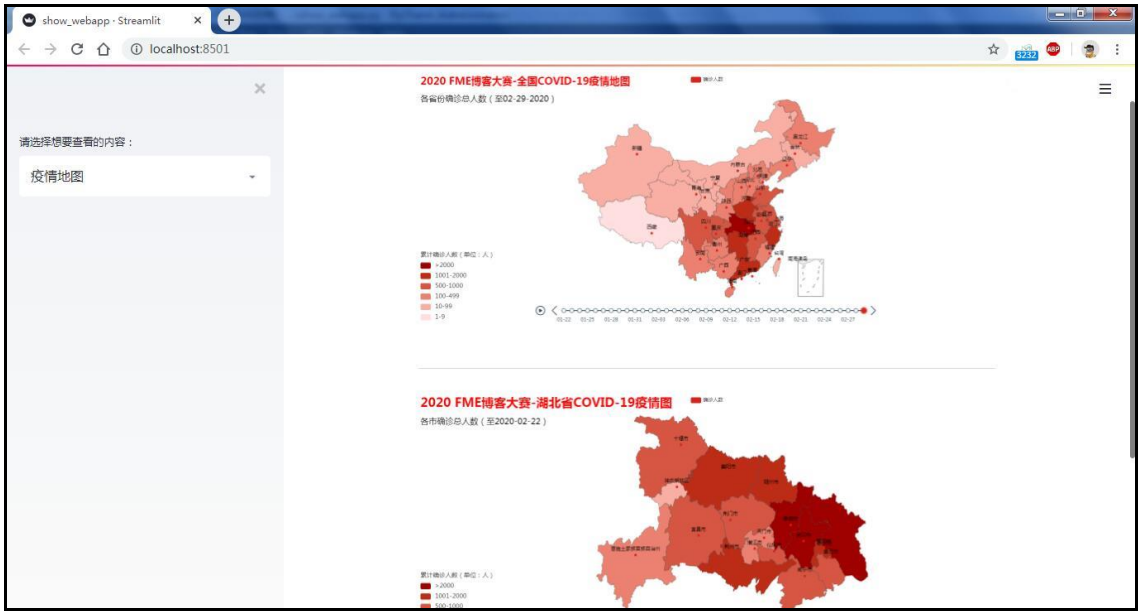


图 10　轻量级疫情可视化 web 应用

5 结语

人生苦短，初学 python。与 python 接触很早，但是一直没有真正地沉下心来学习和应用。最近几个月的亲密接触有一种相见恨晚的感觉。然而数据爬取、生产和可视化终究只是相对初级的应用方式，下一步的目标是结合机器学习和科学计算包，实现 GIS 大数据分析和应用。