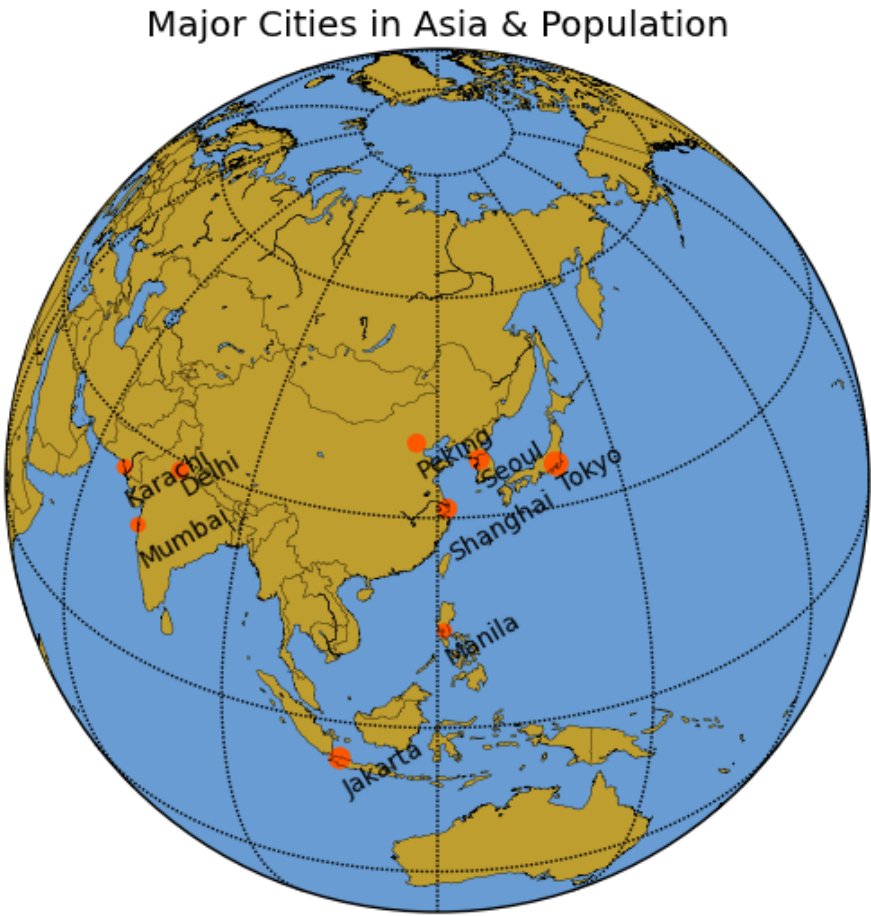


在数据可视化过程中, 我们常常需要将数据根据其采集的地理位置在地图上显示出来。比如说我们会想要在地图上画出城市, 飞机的航线, 乃至军事基地等等。通常来说, 一个地理信息系统都会带有这样的功能。今天我们讨论如何在Python上实现, 并且使用免费的工具包。


matplotlib是Python常用的数据绘制包。它基于numpy的数组运算功能。matplotlib绘图功能强大, 可以轻易的画出各种统计图形, 比如散点图, 条形图, 饼图等。matplotlib常与numpy和scipy相配合, 用于许多研究领域。他们是免费工具, 但其功能足可以与科研界的大佬Matlab竞争。

Basemap是Matplotlib的一个子包, 负责地图绘制。在数据可视化过程中, 我们常需要将数据在地图上画出来。比如说我们在地图上画出城市人口, 飞机航线, 军事基地, 矿藏分布等等。这样的地理绘图有助于读者理解空间相关的信息。

我们下面用Basemap画出亚洲主要城市的人口。如下图, 人口的数量用圆圈的大小表示:



数据如下 (我从Wikipedia上整理的, 你可以随意使用)。将数据保存在文件major_city:



Shanghai	23019148	31.23N	121.47E	China
Mumbai	12478447	18.96N	72.82E	India
Karachi	13050000	24.86N	67.01E	Pakistan
Delhi	16314838	28.67N	77.21E	India
Manila	11855975	14.62N	120.97E	Philippines

Seoul	23616000	37.56N	126.99E	Korea (South)
Jakarta	28019545	6.18S	106.83E	Indonesia
Tokyo	35682460	35.67N	139.77E	Japan
Peking	19612368	39.91N	116.39E	China



第一列是城市名，第二列是人口，第三第四列为纬度和经度，最后一列为所在国家。

下面是我的Python代码，用以绘制上面的地图：



```
# Written by Vamei, http://www.cnblogs.com/vamei/
# Feel free to use or modify this script.

from mpl_toolkits.basemap import Basemap
import matplotlib.pyplot as plt
import numpy as np

#=====# read data
names = []
pops = []
lats = []
lons = []
countries = []
for line in file("../data/major_city"):
    info = line.split()
    names.append(info[0])
    pops.append(float(info[1]))
    lat = float(info[2][: -1])
    if info[2][ -1] == 'S': lat = -lat
    lats.append(lat)
    lon = float(info[3][: -1])
    if info[3][ -1] == 'W': lon = -lon + 360.0
    lons.append(lon)
    country = info[4]
    countries.append(country)

#=====
# set up map projection with
# use low resolution coastlines.
map = Basemap(projection='ortho', lat_0=35, lon_0=120, resolution='l')
# draw coastlines, country boundaries, fill continents.
map.drawcoastlines(linewidth=0.25)
map.drawcountries(linewidth=0.25)
# draw the edge of the map projection region (the projection limb)
map.drawmapboundary(fill_color='#689CD2')
# draw lat/lon grid lines every 30 degrees.
map.drawmeridians(np.arange(0, 360, 30))
map.drawparallels(np.arange(-90, 90, 30))
# Fill continent wit a different color
map.fillcontinents(color='#BF9E30', lake_color='#689CD2', zorder=0)
# compute native map projection coordinates of lat/lon grid.
x, y = map(lons, lats)
max_pop = max(pops)
# Plot each city in a loop.
# Set some parameters
size_factor = 80.0
y_offset = 15.0
rotation = 30
for i, j, k, name in zip(x, y, pops, names):
```

```
size = size_factor*k/max_pop
cs = map.scatter(i,j,s=size,marker='o',color='#FF5600')
plt.text(i,j+y_offset,name,rotation=rotation,fontsize=10)

plt.title('Major Cities in Asia & Population')
plt.show()
```



程序分为两个部分，第一部分为从文件读取数据并处理。第二部分才是真正用basemap绘图。

地图的大小、投影方法等重要信息，是在Basemap()的调用中实现的：

```
map = Basemap(projection='ortho',lat_0=35,lon_0=120,resolution='l')
```

projection参数规定了投影方法。改变投影方法，绘图结果也将非常不同。

城市所在位置是经纬度。我们想要把经纬度对应图像的像素点，需要转换：

```
x, y = map(lons, lats)
```

这个语句转换为图像上的位置。

最后，调用绘制散点图的方法scatter()：

```
cs = map.scatter(i,j,s=size,marker='o',color='#FF5600')
```

在地图上画出数据。

总结

matplotlib中的Basemap是很好用的，具有专业标准的地图绘制工具。它可以与matplotlib的一般绘图功能结合，在地图上绘制数据。

文中需要的软件包：

```
numpy, matplotlib, mpl_toolkits
```

可以使用pip来下载安装各个包。

在Ubuntu的repository中，你可以找到python-matplotlib包。