**数据可视化技术实验**

**实验一（2023.12.5）**

**大数据2101班 李嘉鹏 U202115652**

1. 利用折线图和散点图绘制excel文件中data\_history对应的数据（按日期的新冠疫情数据），要求分别在折线图和散点图上显示confirm，dead和heal数据，使用不同的视觉通道（样式、颜色等）。

注意：（1）中文标注的使用；

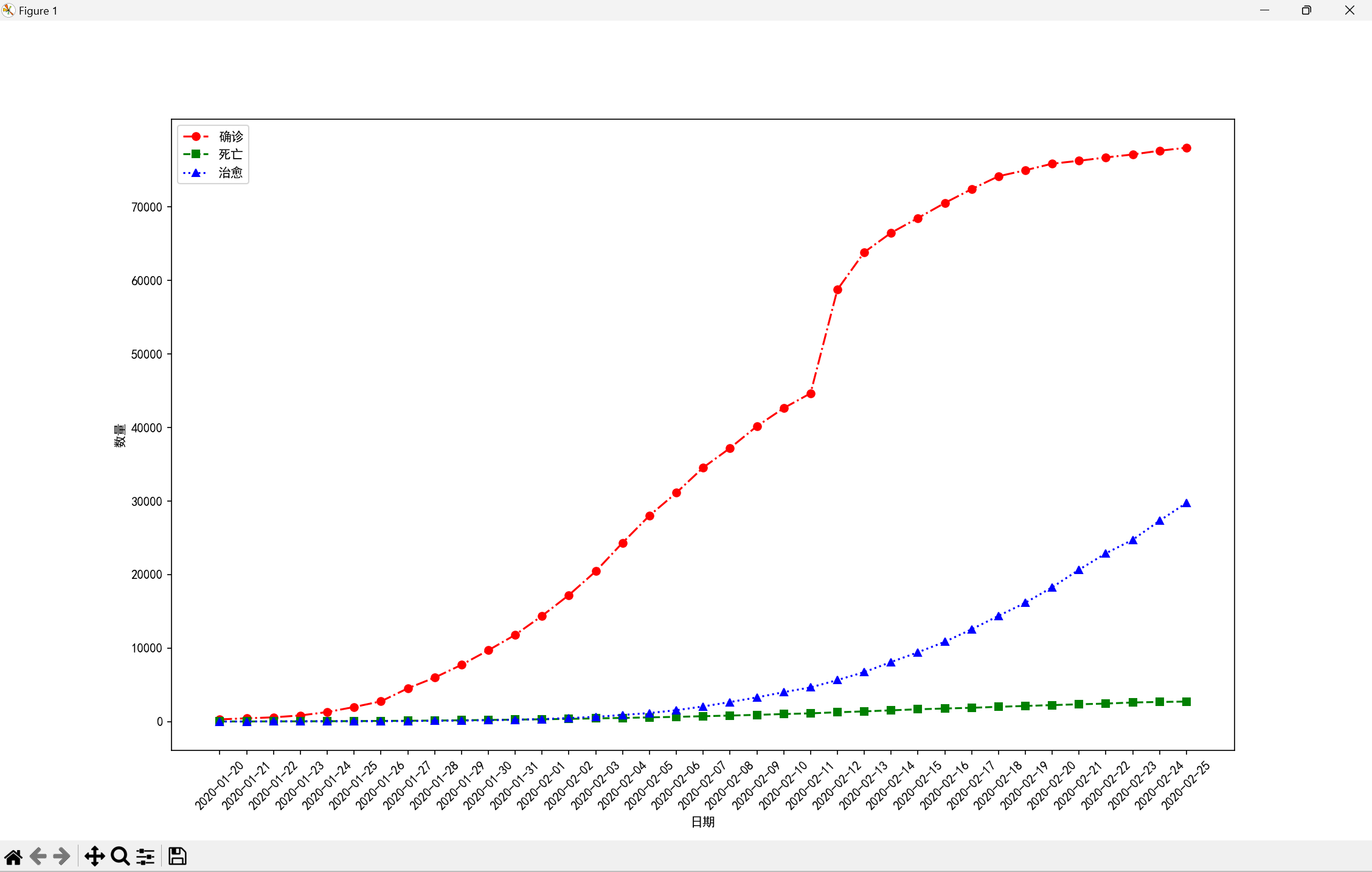
（2）xticks和yticks对坐标轴的处理。

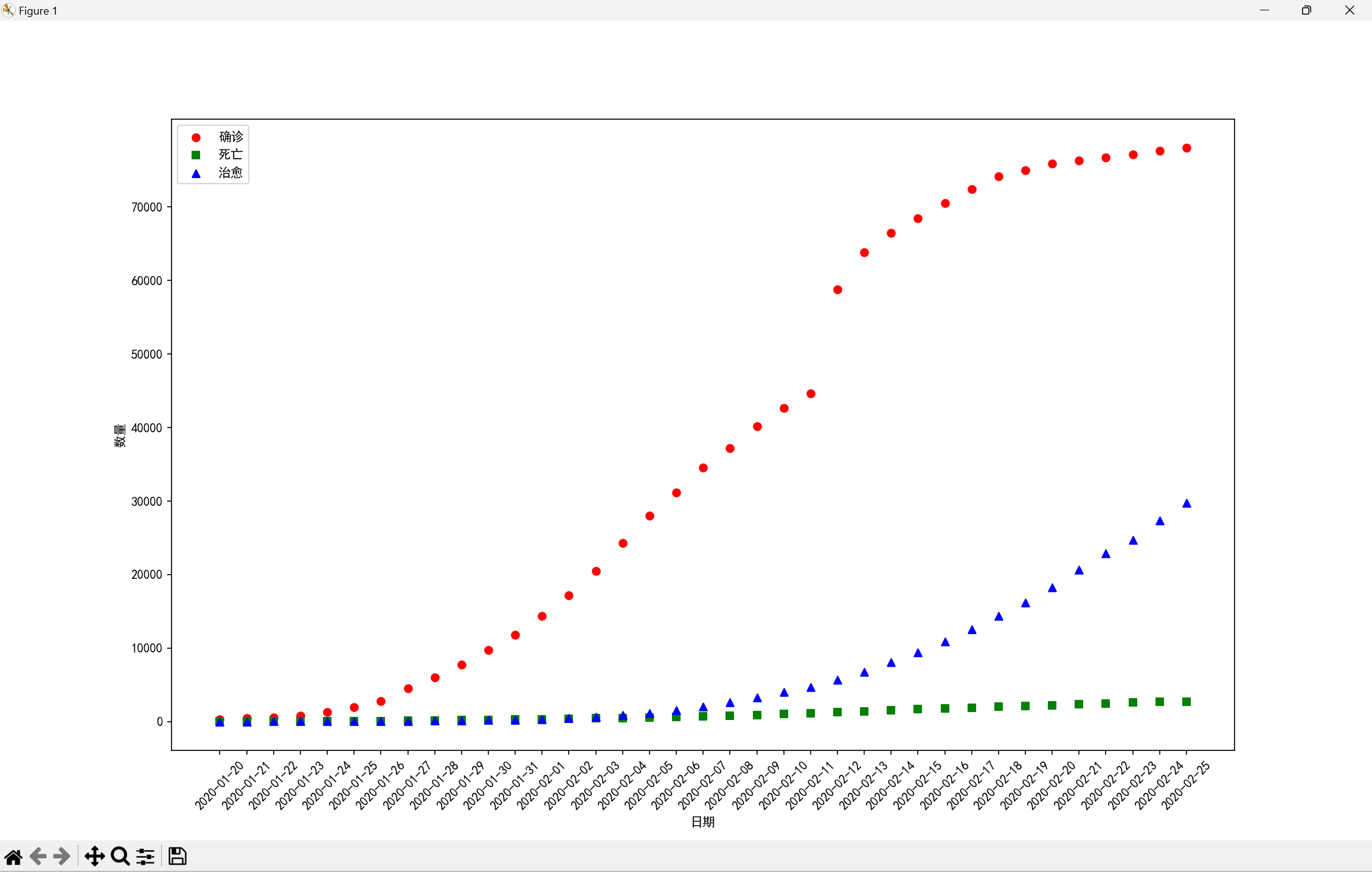
思考：哪一个图更为有效？**（折线图，因为可以反映数据的变化趋势）**

代码：

**import** pandas **as** pd  
**import** matplotlib.pyplot **as** plt  
  
*# 读取excel文件*file\_path = **'covid19\_data.xls'**data = pd.read\_excel(file\_path, sheet\_name=**'data\_history'**)  
  
*# 设置中文显示*plt.rcParams[**'font.sans-serif'**] = [**'SimHei'**]  
  
*# 绘制折线图*plt.figure(figsize=(10, 5))  
plt.plot(data[**'date'**], data[**'confirm'**], label=**'确诊'**, color=**'r'**, linestyle=**'-.'**, marker=**'o'**)  
plt.plot(data[**'date'**], data[**'dead'**], label=**'死亡'**, color=**'g'**, linestyle=**'--'**, marker=**'s'**)  
plt.plot(data[**'date'**], data[**'heal'**], label=**'治愈'**, color=**'b'**, linestyle=**':'**, marker=**'^'**)  
  
*# 设置坐标轴标签*plt.xlabel(**'日期'**)  
plt.ylabel(**'数量'**)  
  
*# 设置坐标轴刻度*plt.xticks(rotation=45)  
plt.yticks(range(0, max(data[**'confirm'**])+1000, 10000))  
  
*# 显示图例*plt.legend()  
  
*# 显示折线图*plt.show()  
  
*# 绘制散点图*plt.figure(figsize=(10, 5))  
plt.scatter(data[**'date'**], data[**'confirm'**], label=**'确诊'**, color=**'r'**, marker=**'o'**)  
plt.scatter(data[**'date'**], data[**'dead'**], label=**'死亡'**, color=**'g'**, marker=**'s'**)  
plt.scatter(data[**'date'**], data[**'heal'**], label=**'治愈'**, color=**'b'**, marker=**'^'**)  
  
*# 设置坐标轴标签*plt.xlabel(**'日期'**)  
plt.ylabel(**'数量'**)  
  
*# 设置坐标轴刻度*plt.xticks(rotation=45)  
plt.yticks(range(0, max(data[**'confirm'**])+1000, 10000))  
  
*# 显示图例*plt.legend()  
  
*# 显示散点图*plt.show()

结果：



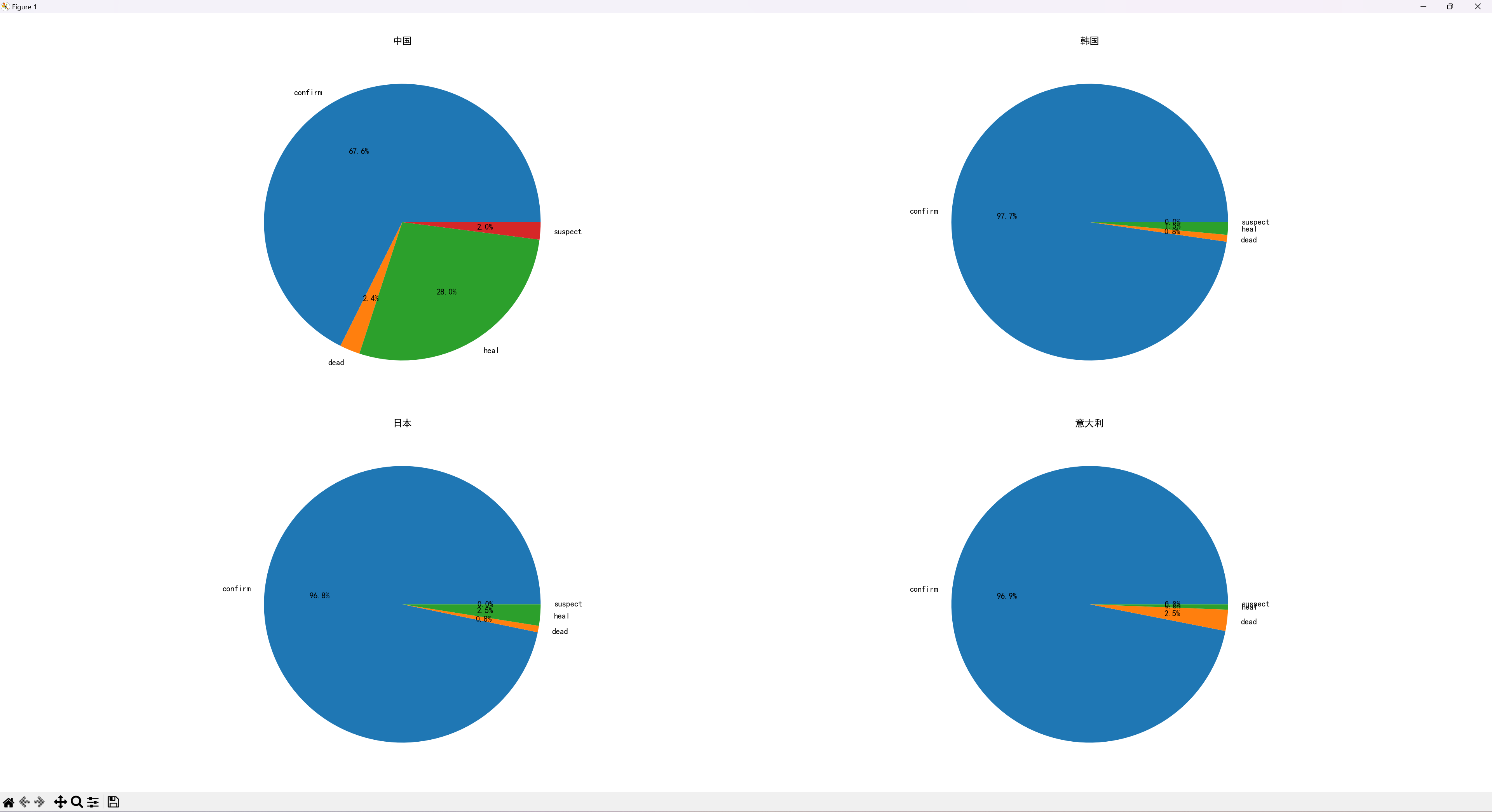


2. 利用饼图绘制excel文件中data\_world对应的数据（各国新冠疫情数据），要求显示确诊人数最多的前4个国家的confirm，dead、heal和suspect的分布饼图。

代码：

**import** pandas **as** pd  
**import** matplotlib.pyplot **as** plt  
  
*# 读取Excel文件*data = pd.read\_excel(**'covid19\_data.xls'**, sheet\_name=**'data\_world'**)  
  
*# 按确诊人数降序排序，取前4个国家*top\_countries = data.nlargest(4, **'confirm'**)  
  
*# 设置中文显示*plt.rcParams[**'font.sans-serif'**] = [**'SimHei'**]  
  
*# 创建一个包含4个子图的画布*fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(12, 8))  
  
*# 绘制每个国家的饼图***for** i, country **in** enumerate(top\_countries[**'country'**]):  
 *# 获取该国家的confirm、dead、heal和suspect数据* confirm = top\_countries.loc[top\_countries[**'country'**] == country, **'confirm'**].values[0]  
 dead = top\_countries.loc[top\_countries[**'country'**] == country, **'dead'**].values[0]  
 heal = top\_countries.loc[top\_countries[**'country'**] == country, **'heal'**].values[0]  
 suspect = top\_countries.loc[top\_countries[**'country'**] == country, **'suspect'**].values[0]  
  
 *# 绘制饼图* ax = axes[i // 2, i % 2]  
 ax.pie([confirm, dead, heal, suspect], labels=[**'confirm'**, **'dead'**, **'heal'**, **'suspect'**], autopct=**'%1.1f%%'**)  
 ax.set\_title(country)  
  
*# 调整子图之间的间距*plt.tight\_layout()  
  
*# 显示图形*plt.show()

结果：



3. 利用直方图和条形图绘制excel文件中current\_prov对应的数据（各省新冠疫情数据），要求使用多个子图，使用合适的视觉通道。

思考：哪一个图更为有效？**（条形图，因为可以比较两组或以上的数值，找出其联系。但是在此题里实际上直方图和条形图的效果都比较差，因为横坐标“省份”相互独立，彼此之间没有关系；并且由于“湖北”的相关数据比其它省份的数据大很多，因此后面一些省份的数据都看起来很少）**

代码：

**import** pandas **as** pd  
**import** matplotlib.pyplot **as** plt  
**import** numpy **as** np  
  
*# 读取excel文件*file\_path = **'covid19\_data.xls'**current\_prov = pd.read\_excel(file\_path, sheet\_name=**'current\_prov'**)  
  
*# 获取需要可视化的数据，例如各省的确诊人数、治愈人数和疑似人数*provinces = current\_prov[**'province'**]  
confirm\_data = current\_prov[**'confirm'**]  
heal\_data = current\_prov[**'heal'**]  
dead\_data = current\_prov[**'dead'**]  
  
plt.rcParams[**'font.sans-serif'**] = [**'SimHei'**]  
  
*# 创建画布和子图*fig, axs = plt.subplots(4, 1, figsize=(10, 10))  
  
*# 条形图*index = np.arange(len(provinces)) *# 省份的索引*bar\_width = 0.2 *# 条形宽度  
  
# 条形图：各省确诊人数*rects1 = axs[0].bar(index - bar\_width, confirm\_data, bar\_width, color=**'skyblue'**, label=**'确诊人数'**)  
  
*# 条形图：各省治愈人数*rects2 = axs[0].bar(index, heal\_data, bar\_width, color=**'lightcoral'**, label=**'治愈人数'**)  
  
*# 条形图：各省死亡人数*rects3 = axs[0].bar(index + bar\_width, dead\_data, bar\_width, color=**'green'**, label=**'死亡人数'**)  
  
axs[0].set\_xlabel(**'省份'**)  
axs[0].set\_ylabel(**'人数'**)  
axs[0].set\_title(**'各省确诊、治愈和死亡人数对比'**)  
axs[0].set\_xticks(index)  
axs[0].set\_xticklabels(provinces)  
axs[0].legend()  
  
*# 直方图：各省确诊人数*axs[1].bar(provinces, confirm\_data, color=**'skyblue'**)  
axs[1].set\_xlabel(**'省份'**)  
axs[1].set\_ylabel(**'确诊人数'**)  
axs[1].set\_title(**'各省确诊人数'**)  
  
axs[2].bar(provinces, heal\_data, color=**'lightcoral'**)  
axs[2].set\_xlabel(**'省份'**)  
axs[2].set\_ylabel(**'治愈人数'**)  
axs[2].set\_title(**'各省治愈人数'**)  
  
axs[3].bar(provinces, dead\_data, color=**'green'**)  
axs[3].set\_xlabel(**'省份'**)  
axs[3].set\_ylabel(**'死亡人数'**)  
axs[3].set\_title(**'各省死亡人数'**)  
  
*# 调整布局*plt.tight\_layout()  
  
*# 显示图形*plt.show()

结果：

