目　　录

基于 Python 的澳大利亚自杀率数据分析

【内容摘要】 当前世界面临着很多复杂的社会问题，如自然灾害、人口老龄以及家庭暴力等，在存在的众多问题中自杀俨然变成当下需要解决的全球公共卫生问题。针对此问题，我们选取了澳大利亚自杀的数据集，论文首先通过下载自杀率的数据集，整理和筛选有价值的属性，利用 Python 索引出澳大利亚的数据集；其次用 Pyecharts 库对性别、年龄、出生年代、人均 GDP 等属性绘制图形，对各个图形进行分析，得出男性的自杀率是女性的 3.5 倍、青年的自杀率最高、人均 GDP 对其家庭的幸福指数有影响，较低时会增加自杀率；最后查询出生年代的代称所属年份及年代发生事件，结合图形得出了出生年代的战争与当时的经济直接影响国家的自杀，战争使国家的自杀率增高，经济的影响使得人民迫于经济的压力而选择自杀的行为。而此次关于澳大利亚自杀率的数据分析可帮助人们更清晰了解自杀大多根源以及为上层决策意识提供参考。

【关键词】 Python；自杀；数据分析

1 背景

自杀是一种社会病,在不同的时期都有着特定的自杀倾向，它涉及一个国家的经济、社会、文化发展等公共卫生问题，是精神卫生研究领域的重要课题之一。2009 年彭现美对美国人口自杀状况进行分析研究中，通过美国不同性别、年龄、种族、区域等自杀人群的分析，了解美国自杀人群的特点，以期采取相应的措施，表明了尽管美国的自杀率在世界各国中不是最高的，但同其他发达国家一样所造成的社会问题最为突出。结合前人的研究，利用 Python 编程语言对自杀人群所出生的年代做进一步的分析，查询各个年代的代称所指为哪个年代及其年代所存在的社会问题。对澳大利亚不同性别、年龄段、出生年代等进行数据分析，了解自杀人群的特点。

2 相关技术

2.1 统计学

统计学是应用数学的一个分支，主要通过利用概率论建立数学模型，通过收集数据，进行量化分析、总结，做出推测，为相关决策提供依据和参考。由于统计学的定量研究具

有客观、准确和可检验的特点，所以统计方法就成为实证研究的最重要的方法，广泛适用于自然、社会、经济、科学技术各个领域的分析研究。

2.2 Python

Python 语言是一种简单、易学、免费开源的编程语言,具有丰富的数据结构、灵活的程序处理方式以及大量支持该语言的第三方函数库,同时在数据处理方面也具有非常明显的编程优势。

Python 需要第三方库增加数据分析能力,如:pandas、pyecharts、seaborn、Matplotlib 等。其中 Pandas 纳入大量库和一些标准的数据模型提供了高效地操作大型数据集所需的工具，提供了很多用于数据操作与分析的功能；pyecharts 是一个数据可视化的库，用于生成 echarts 图标的类库，方便与 Python 对接，可以直接生成图；seaborn 是基于 Matplotlib 的可视化包，它提供了高度的交互式界面，方便用户做出各种有吸引力的统计图表；Matplotlib 是 Python 的绘画库，可快速生成多种图像格式。

3 数据分析流程

数据分析流程一般包括收集数据，提出问题，数据清理，数据分析，总结结论。

先获取所需要的数据集，针对数据集提出相关的问题，对已有的数据集进行清洗，使得数据集干净，接着对所提出的问题对清洗后的数据集进行数据分析，最后对数据分析的结果做一个总结。

图 1 数据分析流程图

4 自杀率数据数据分析

4.1 数据集

在 DataFountain 官网中，从数据集中获得原数据名为“自杀率概述 1985 年至 2016 年”直接下载其数据。数据集中于几个大洲与不同国家的自杀统计，由于数据集的数据量较大，本次分析只选取了澳大利亚的自杀率进行数据分析。本数据集包含了一个文件

（master.csv），每一行都有年份 GDP、人口总数、自杀人数等信息。

表 1 字段的含义

country	城市
year	年份
sex	性别
age	年龄
suicides_no	自杀人数
population	人口
suicides/100k pop	自杀率
country-year	城市-年份
HDI for year	人类发展指数
gdp_for_year($)	年 GDP
generation	出生年代

4.2 定义问题

问题是数据分析流程中的一个核心，后续所有的操作也都是在围绕着这个问题展开，提出问题，后续才能有针对性的分析。本文基于上述数据提出下列问题。

（1）澳大利亚每年的自杀率

（2）不同年龄段的变化趋势

（3）不同年龄段与不同的年份的自杀率

（4）男女自杀比率的变化趋势

（5）不同出生年代的自杀率变化趋势

（6）自杀率与人均 GDP 的关系

5 数据分析具体操作

利用 Python 进行数据分析，主要是 Pandas（数据分析）和 pyecharts（数据可视化）。

5.1 收集数据

读取文件，首先要安装 pandas 库:pip install pandas，因为用 annaconda 不用安装此库，即导入此数据分析库即可。用 pandas.read_csv()参数读取数据。

代码如下：

```
import pandas as pd                    #导入数据分析库且命名为 pd
data = pd.read_csv('master.csv')  #读取 master 中的数据赋值给 data
```

	country	year	sex	age	suicides_no	population	suicides/100k pop	country-year	HDI for year	gdp_for_year ($)	gdp_per_capita ($)	generation
0	Albania	1987	male	15-24 years	21	312900	6.71	Albania1987	NaN	2,156,624,900	796	Generation X
1	Albania	1987	male	35-54 years	16	308000	5.19	Albania1987	NaN	2,156,624,900	796	Silent
2	Albania	1987	female	15-24 years	14	289700	4.83	Albania1987	NaN	2,156,624,900	796	Generation X
3	Albania	1987	male	75+ years	1	21800	4.59	Albania1987	NaN	2,156,624,900	796	G.I. Generation
4	Albania	1987	male	25-34 years	9	274300	3.28	Albania1987	NaN	2,156,624,900	796	Boomers
...	...	...	...	...	...	...	...	...	...	...	...	...
27815	Uzbekistan	2014	female	35-54 years	107	3620833	2.96	Uzbekistan2014	0.675	63,067,077,179	2309	Generation X
27816	Uzbekistan	2014	female	75+ years	9	348465	2.58	Uzbekistan2014	0.675	63,067,077,179	2309	Silent
27817	Uzbekistan	2014	male	5-14 years	60	2762158	2.17	Uzbekistan2014	0.675	63,067,077,179	2309	Generation Z
27818	Uzbekistan	2014	female	5-14 years	44	2631600	1.67	Uzbekistan2014	0.675	63,067,077,179	2309	Generation Z
27819	Uzbekistan	2014	female	55-74 years	21	1438935	1.46	Uzbekistan2014	0.675	63,067,077,179	2309	Boomers

27820 rows × 12 columns

图 2 输出数据集

从图 1 可以看出，csv 文件有包含了 27820 行*12 列的数据集，包含几大洲跟不同的国家。

5.2 数据清洗及获取澳大利亚数据

数据清洗主要是针对数据中的错误值、异常值、缺失值进行处理的过程，以及删除那些取值很多的类别型字段，或者取值一致性程度极高的字段。由于原数据中其城市-年份（country-year）为重复值、人类的发展指数（HDI for year）为空值，删除这两个无用值。因此把数据进行清洗，再进行探索分析。代码如下：

```
data.drop(['country-year','HDI for year'],axis=1,inplace=True)
```

由于数据量较大，且大洲与国家的分布复杂，在本次分析只选取了澳大利亚的自杀率进行数据分析。

用 pandas 布尔索引，通过布尔运算选取澳大利亚的行数据。行可以直接通过[]选择，必须是数字范围或字符串范围索引。代码如下：

```
data_test = data[data['country']=='Australia']#澳大利亚数据集为 data_test
```

	country	year	sex	age	suicides_no	population	suicides/100k pop	country-year	HDI for year	gdp_for_year ($)	gdp_per_capita ($)	generation
1426	Australia	1985	male	75+ years	67	219000	30.59	Australia1985	NaN	180,190,994,861	12374	G.I. Generation
1427	Australia	1985	male	25-34 years	357	1299100	27.48	Australia1985	NaN	180,190,994,861	12374	Boomers
1428	Australia	1985	male	55-74 years	282	1177400	23.95	Australia1985	NaN	180,190,994,861	12374	G.I. Generation
1429	Australia	1985	male	15-24 years	315	1355800	23.23	Australia1985	NaN	180,190,994,861	12374	Generation X
1430	Australia	1985	male	35-54 years	411	1906800	21.55	Australia1985	NaN	180,190,994,861	12374	Silent
...	...	...	...	...	...	...	...	...	...	...	...	...
1781	Australia	2015	female	25-34 years	119	1747715	6.81	Australia2015	NaN	1,349,034,029,453	60656	Millenials
1782	Australia	2015	female	55-74 years	152	2411343	6.30	Australia2015	NaN	1,349,034,029,453	60656	Boomers
1783	Australia	2015	female	75+ years	52	884347	5.88	Australia2015	NaN	1,349,034,029,453	60656	Silent
1784	Australia	2015	female	5-14 years	8	1428159	0.56	Australia2015	NaN	1,349,034,029,453	60656	Generation Z
1785	Australia	2015	male	5-14 years	6	1507502	0.40	Australia2015	NaN	1,349,034,029,453	60656	Generation Z

360 rows × 12 columns

图 3 澳大利亚的数据集

从图 3 可以看出，澳大利亚的数据集有 360 行。

5.3 自杀率描述性统计指标分析

每年的自杀率=该年的自杀总人数/该年的澳大利亚总人口数，即计算每一年的自杀总人数和该年的澳大利亚总人口数。

5.3.1 人口总数指标统计

用 groupby()函数，以年作为参数，对数据进行分组。用 cumsum 函数返回给定 axis 上的累计和。代码如下：

```
data_test['all_suicide_no'] = data_test['suicides_no'].groupby(data_test['year']).cumsum()#获取每年自杀人数总人数
data_test['all_population'] = data_test['population'].groupby(data_test['year']).cumsum()#获取每年统计人数
```

5.3.2 进行重复值和排序的处理

处理重复值，使用 drop_duplicates()方法，其具体用法是：drop_duplicates（subset=’列名’,keep=’firsrt’,inplace=’True’）函数是删除 DataFrame 的某列中重复项的函数。subset，输入列名，形式为 subset=’列名 1’，可输入多列，形式为 subset=[’列名 1’,’列名 2’]，keep 包括’first’，’last’，False，三个参数，注意 first 和 last 带引号，而 False 没有，’first’是保留重复项中第一个，last 是保留最后一个，False 是都不保留。

排序处理，函数 sort_values()可以将数据集依照某个字段中的数据进行排序，即可根据指定列数据或行数据排序。其具体用法是：dataFrame.sort_values(by=

‘##’,axis=0,ascending=True)by 指定列名（axis=0）；axis=0 则按照指定列中数据大小排序；ascending 是否按指定列的数组升序排列，默认为 True，即升序排列。代码如下：

```
data_test1 = data_test.drop_duplicates(subset='year',keep='last').sort_values(by="year" ,axis=0,ascending=True)#删除重复的年份 ， 保留所有数据的最后一个
```

5.3.3 把自杀人数和统计人数转换成列表

values 方法返回结果为数组，将数组转化为列表 tolist 方法。代码如下：

```
suicide_num = data_test1['all_suicide_no'].values.tolist() #将自杀人数转换成列表，suicide_num 为自杀人口
population_num = data_test1['all_population'].values.tolist()#将统计人数转换成列表，population_num 为人口数
year = data_test1['year'].values.tolist()#将年份转换成列表
```

5.3.4 每年的自杀率指标统计

list 列表，Python 最常用的数据类型。列表由[]来表示，列表能装对象的对象。range()可创建整数列表，一般用在 for 循环中，使用 for 和 range 将数值添加在列表里。range()函数直接返回一个列表可通过外加 list 函数将迭代对象返回一个列表即可。 代码如下：

```
new_list = []                                #定义一个名为 new_list 的 list
for i in range(len(suicide_num)):            #计算每年自杀率
    m = round((suicide_num[i]/population_num[i])*100000,1)
    new_list.append(m)
print(new_list)                              #生成列表
```

```
[12.8, 13.8, 14.4, 14.2, 13.3, 13.9, 14.3, 13.9, 12.4, 13.8, 12.9, 14.3, 15.4, 15.1, 14.1, 13.4, 13.6, 12.6, 11.6, 11.2, 11.1, 11.3, 11.6, 1
1.4, 11.6, 11.5, 12.1, 12.1, 13.2, 13.6]
```

图 4 各个年份的自杀率

从图 4 可以看出，澳大利亚每年的自杀率。

5.3.5 输出各个年份及对应的自杀率

用 print 函数直接输出，代码如下：

```
print(year)
print(new_list)
```

```
[1985, 1986, 1987, 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2006, 2007, 2008, 2
009, 2010, 2011, 2012, 2013, 2014, 2015]
[12.8, 13.8, 14.4, 14.2, 13.3, 13.9, 14.3, 13.9, 12.4, 13.8, 12.9, 14.3, 15.4, 15.1, 14.1, 13.4, 13.6, 12.6, 11.6, 11.2, 11.1, 11.3, 11.6, 1
1.4, 11.6, 11.5, 12.1, 12.1, 13.2, 13.6]
```

图 5 年份与对于的自杀率

从图 5 可以看出，在 1985 年至 2015 年中，缺少了 2005 年的年份，原本 1985 年至 2015 年的数据为 31 个，而现只有 30 个数据，与自杀率数据对应。

5.4 澳大利亚每年自杀率的变化趋势图

由于数量较大，因此使用 pyecharts 库，它是一个用户数据可视化的包，可以展示动态效果图，使用比较美观，并且显示数据方便，鼠标悬停在图上，即可显示数值、标签等。因 Anaconda 没有此数据可视化的包，所以需自行下载安装 pyecharts 库。利用折线图显示该自杀率的变化趋势，从 pyecharts 库中导入 Line 图类，即 from pyecharts import Line。

渲染折线图，用 add()方法，设置折线图的数据和配置各种的项。rander()默认将会在根目录下生成一个 html 的文件，文件用浏览器打开，即可直接下载该折线图。代码如下：

```
year = ['1985', '1986', '1987', '1988', '1989', '1990', '1991', '1992', '1993',
'1994', '1995', '1996', '1997', '1998', '1999', '2000', '2001', '2002', '2003',
'2004',  '2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013', '2014',
'2015']   #每个年份
line = Line("澳大利亚自杀率变化图",width=1500,height=500)#折线图的主题及长和宽
```

```
line.add("自杀率", year, new_list,        #折线图标注、横纵坐标的表示
mark_point=["max", "min"],                #最大最小点
mark_line=["average"],                    #平均线
is_smooth=True,                           #折线平滑曲线处理
mark_point_textcolor='#00FF00',           #设置标注点颜色
mark_point_symbolsize=50)                 #设置标注点大小
```

line.render('1985 年-2015 年澳大利亚自杀变化图.html')#生成本地的 html，命名为 1985 年-2015 年澳大利亚自杀变化图

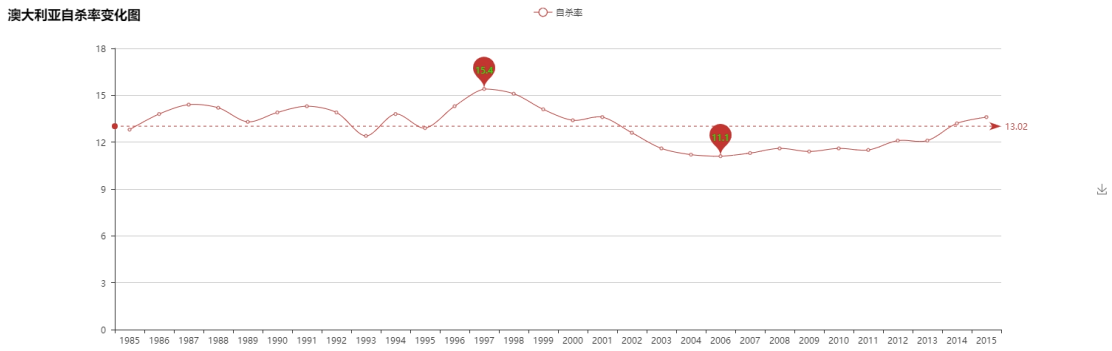

图 6 1985 年-2015 年澳大利亚自杀变化图

从图 6 可以看出，1985 年至 2015 年自杀率的平均值为 13.02,整体的自杀情况忽高忽低。在 1985 年至 1997 年间，呈上升的趋势，且在 1997 年是自杀率达到了峰值，数值达到了 15.4。

达到峰值之后，自杀率呈下降的趋势，澳大利亚自杀的原数据中，缺少了 2005 年的数据，缺少数据之后的一年中，即 2006 年出现了自杀率最低的一年，数值为 11.1，低于平均值的年份也在零几年的时间段出现。

5.5 不同年龄段的变化趋势

5.5.1 分类不同年龄段

用 pandas 布尔索引，通过布尔运算选取澳大利亚同一年龄的行数据。行可以直接通过[]选择，必须是数字范围或字符串范围索引。代码如下：

```
data1 = data_test[data_test['age'] == '5-14 years']#5-14 岁的年龄段
```

5.5.2 同一年龄段的人数指标统计

用 groupby()函数，以年龄作为参数，对数据进行分组。用 cumsum 函数返回给定 axis 上的累计和。代码如下：

```
new_data_test['the_suicides_no'] = new_data_test['suicides_no'].groupby(new_data_test['age']).cumsum()# 不同年龄段自杀人数总分数
new_data_test['the_population'] = new_data_test['population'].groupby(new_data_test['age']).cumsum()#不同年龄段统计总人数
```

5.5.3 进行重复值和排序的处理

使用 drop_duplicates()方法，做处理重复值；用函数 sort_values()可以将数据集依照某个字段中的数据进行排序。代码如下：

```
new_data_test1 = new_data_test.drop_duplicates(subset=['age'],keep='last').sort_values(by="age" , ascending=True)#删除重复的年龄段 ，保留所有数据的最后一个
```

5.5.4 把自杀人数和统计人数转换成列表

values 方法返回结果为数组，将数组转化为列表 tolist 方法。代码如下：

```
suicide_num1 = new_data_test1['the_suicides_no'].values.tolist()#将自杀人数转换成列表
population_num1 = new_data_test1['the_population'].values.tolist()#将统计人数转换成列表
print(suicide_num1)#输出同一年龄段的自杀人数
print(population_num1)#输出同一年龄段的统计总人数
```

```
[11086, 15034, 26150, 320, 12911, 4610]
[84591312, 88168450, 159205548, 79793947, 98814679, 31803850]
```

图 7 同一年龄段的自杀人数与总人数

从图 7 可知，不同年龄段的自杀人数和统计总人数。

5.5.5 每年的自杀率指标统计

利用 list 列表和 range（）可创建整数列表，计算不同年龄段的自杀率。

5.5.6 不同年龄段变化趋势

使用 pyecharts 库，利用雷达图显示该自杀率的变化趋势，从 pyecharts 库中导入 Radar 图类，即 from pyecharts import Radar。

雷达图用到的 matplotlibli 里面的 pyplot，并命名为 plt，代码： import matplotlib.pyplot as plt。

用 add()方法，设置雷达图的数据和配置各种的项，rander()默认生成一个 html 的文件。代码如下：

```
radar = Radar("不同年龄自杀率",width = 1200,height = 500)
radar.config(schema,radar_text_size=20,shape= "circle",)
radar.add("",
          the_list,
          area_opacity=0.2,                #填充区域透明度
          legend_top='bottom',             #图例位置，默认 top
          line_width=2                     #线条宽度
          )
radar.render("不同年龄段的自杀率.html")
```

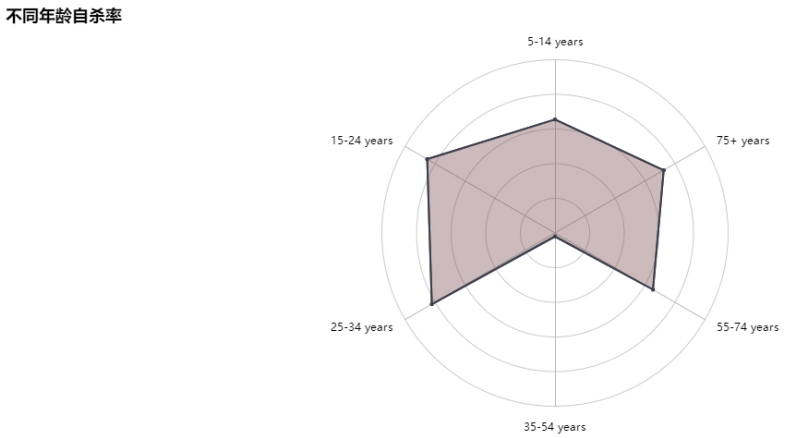

图 8 不同年龄段自杀率的雷达图

从图 8 可以看出，在这六个不同年龄段中，15-24 岁的自杀率最高；35-54 年的自杀率最低。由此可见，青少年心理承受的压力较弱，社会及家庭应该多引导青少年具备释放压力的能力；中老年人心理可以承受压力，中年人有着上有老下有小压力，不会轻易选择自杀，而老年人在享受着晚年及儿孙满堂的生活，更不会选择自杀。

5.6 不同年龄段与不同的年份的自杀率的分析

5.6.1 不同年龄段与不同的年份指标统计

用 groupby()函数，以年份作为参数，对数据进行分组。用 value_counts()统计数据

中年份列中有多少个不同值。

5.6.2 绘制不同年龄段与不同的年份的自杀率图

用 plt.rcParams[]，使用 rc 配置文件来自定义图片的各种默认属性，通过 rc 参数默认属性，用 plt.figure()设置图形的宽和高，dpi 指定绘画的分辨率。设置该折线图 x,y 轴的属性及各个年龄段的图示。

用 seaborn 库中的 linepolot(折线图)，设定不同年龄段不同年份的折线图。

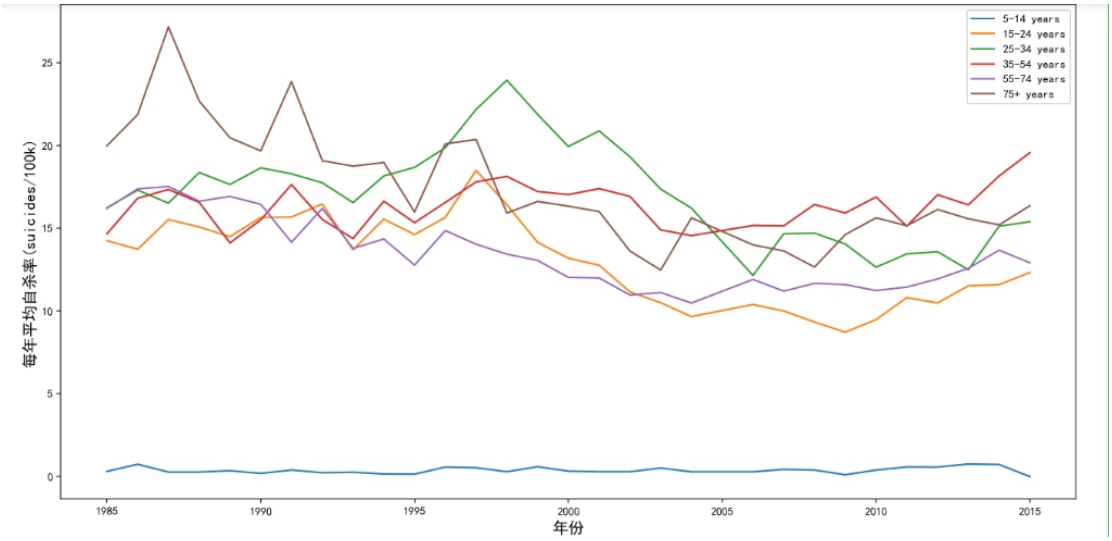

图 9 不同年龄段与不同年份的自杀率折线图

从图 9 可看出，除 5 至 14 岁之间的人群外，其他的年龄段的自杀率起伏不定且相差较大。75+的年龄段中，在前期的时间里，自杀率一直位居榜首，直至 1995 年微下降，而后仍旧存在摇摆不定的情况。25 至 43 岁之间的人群中，在 1995 年至 2005 年这十年之间存在很大的落差。在不同的年份中，各个年龄段的每年自杀率大多集中在 15suicides/100k。

5.7 男女自杀比率的变化趋势

5.7.1 男女性自杀率指标统计

通过布尔运算选取男性或女性的数据的数量。计算其全部的人数，最后利用除法运算，计算出男女性自杀的比率。代码如下：

```
data_1 = data_test[data_test['sex'] == 'female']#女性
data_2 = data_test[data_test['sex'] == 'male']#男性
total = data_1['suicides_no'].values.sum()+data_2['suicides_no'].values.sum()# 自 杀 总
```

人数

```
m = data_1['suicides_no'].values.sum()/total#女性自杀比率
n = data_2['suicides_no'].values.sum()/total#男性自杀比率
print(m)#输出女性自杀比率
print(n)#输出男性自杀比率
```

5.7.2 绘制男女自杀比率饼状图

从 pyecharts 库中导入 Pie 图类，即 from pyecharts import Pie。用 add()方法，设置饼图的数据和配置各种的项，生成一个 html 的文件。代码如下：

```
x = ['女性','男性']#饼图的两大属性
y = [m,n]#男女自杀比率数值
pie = Pie("男女自杀率比例",width = 1200,height = 600)#饼图主题与宽、高
pie.add("", x, y, is_label_show=True)
pie.render('男女自杀比率.html')#是否显示标签
```

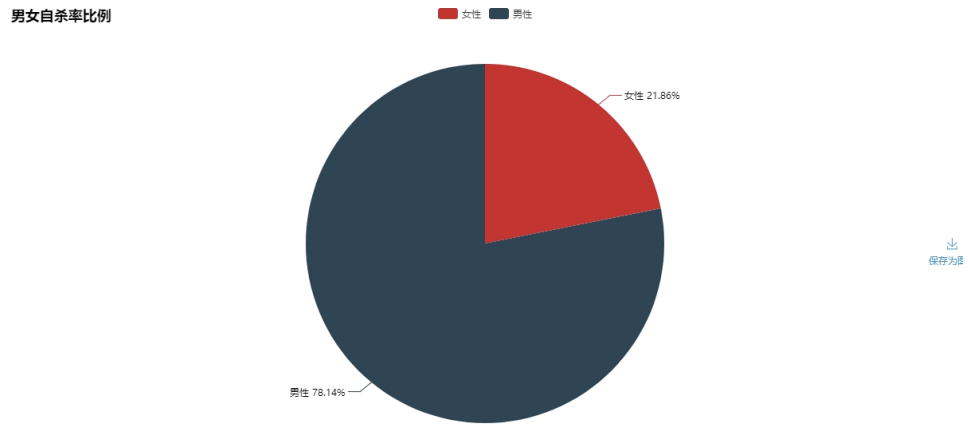

图 10　男女自杀比率饼图

从图 10 可以看出，男性的自杀率为 78.14%，女性的自杀率为 21.86%。男性的自杀比率是女性自杀率的 3.5 倍。由此可见，性别对自杀有着直接的关系，而男性的自杀率高于女性，男性所承受压力相对于女性来说会更加大。

5.8 不同出生年代的自杀率变化趋势

表 2 出生年代的知识普及

代称	年代	年份
G.I.Generation	大兵的一代(第二次世界大战期间出生)	1939-1945
Silent	沉默的一代（出生在大萧条时期）	1925-1940
Boomers	婴儿潮一代（历史上最大的人口一代 ）	1945-1960
Generation X	X 一代（在艾滋病流行期间长大）	1980-1990
Generation Z	Z 世代（第一个真正的数字原生代）	1995-2005
Millenials	千禧一代（现在人口最大的群体）	2000 年左右

5.8.1 不同年代的自杀人数指标统计

用 groupby()函数，以年代作为参数，对数据进行分组。用 cumsum 函数返回给定 axis 上的累计和。代码如下：

```
last_data_test['last_suicides_no'] = new_data_test['suicides_no'].groupby(new_data_test['generation']).cumsum()# 统计不同年代自杀人数
last_data_test1 = last_data_test.drop_duplicates(subset=['generation'],keep='last')#去重保留每个年代最后一列
suicide_num2 = last_data_test1['last_suicides_no'].values.tolist()
print(suicide_num2)#获取总自杀人数
```

5.8.2 绘制不同出生年代的自杀率漏斗图

从 pyecharts 库中导入 Funnel 图类，即 from pyecharts import Funnel。用 add()方法，设置漏斗图各种的项，生成一个 html 的文件。代码如下：

```
last_data_test1 = [4677, 20951, 6964, 23464, 13929, 126]#各个年代自杀数据
funnel =Funnel('不同年代自杀分布图',title_text_size = 20,title_pos = 'center',width = 1200,height = 600)#设置标题大小、宽高
attr = ['G.I.Generation','Millenials','Generation X','Silent','Generation Z','Boomers']#各个年代的代称
funnel.add( '',attr,last_data_test1,is_label_show=True,label_pos='inside', is_legend_show = False)#添加漏斗图的各种项
```

funnel.render('不同年代自杀分布图.html')

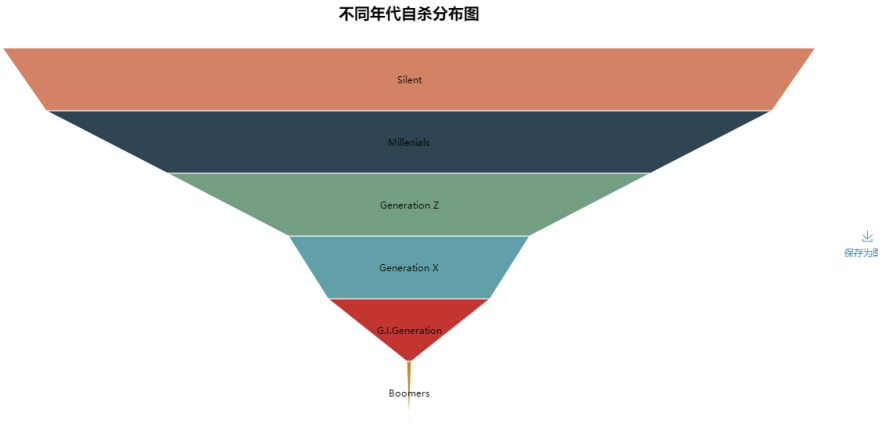

图 11 不同年代自杀漏斗图

从图 11 可以看出，出生在沉默的一代（slient）自杀率为高，出生在婴儿潮一代（boomers）自杀率最低。由此可见，年代对自杀率有一定的影响力。

在沉默的一代中，这一代人出生在 1946 年至 1964 年，由于二战造成的低生育率，使得当时人口数量锐减，加上当时经济大萧条，由于经济的压力，导致自杀率在这六大年代中，其自杀率为最高；在婴儿潮一代中，这一代人出生在 1945 年至 1960 年，该年代是二战后随之而来的生育大潮中出生的一代，使得国家的人口增加，当时也没有经济萧条，其自杀率是最低的。

总的来说，战争与经济大萧条的因素下，会直接影响澳大利亚的自杀率。

5.9 自杀率与人均 GDP 的关系

5.9.1 对自杀率与人均 GDP 做透视表

用数据透视表 prvot_table,以年份作为索引，索引出自杀率与 gdp 之间的关系。

5.9.2 绘制自杀率与人均 GDP 的关系图

plt.figure()在 plt 中绘制一张图，设置图形的宽和高及分辨率；plt.title()设置标题及图片之间的距离；seaborn 库中的线性回归图 regplot()。代码如下：

```
plt.figure(figsize=(16,6),dpi = 80)
plt.title('自杀率与人均 GDP 的关系',fontsize = 10)
sns.regplot(x="gdp_per_capita ($)", y="suicides/100k pop", data=data_new,marker="+",color = 'red')
```

```
plt.xlabel('人均 GDP',fontsize = 10)
plt.ylabel('自杀率(suicides/100k)',fontsize = 10)
plt.show()
```

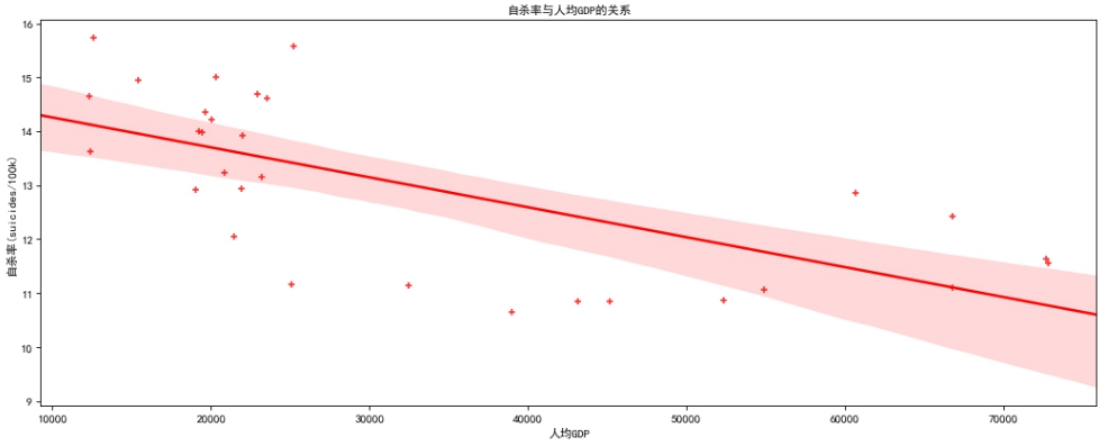

图 12 自杀率与人均 gdp 的关系

从图 12 可以看出，人均 gdp 对自杀率具有一定的影响，从点的分布情况可以看出随着人均 gdp 增加，自杀率明显有下降，人均 gdp 在 10000-20000 之间，其自杀率较密集且自杀率的数值相对较高。

6 总结

本文以 Python 语言为基础，对澳大利亚自杀率数据进行了分析处理。自杀率分析中得出青年的自杀率最高，中老年人的的自杀率最低；男性的自杀率是女性的 3.5 倍，男性所承担的压力比女性大很多；出生年代的战争与当时的经济都直接影响国家的自杀，战争使国家的自杀率增高，经济的影响使得人民迫于经济的压力而选择自杀的行为了断人生；人均 gdp 衡量了该家庭的经济状况，人均 gdp 较低时，会对其家庭的幸福指数有影响，造成经济的危机，迫使买房难、成家难、养家难等压力，从而影响自杀率。

使用了 Python 中其他的库以增加数据分析的能力，在绘制可视化的图中，在 Matplotlib 数据可视化化的基础上，使用了 pyecharts 库的雷达图、折线图、饼图、漏斗图，展示了其动态效果图，从视觉上比较美观；使用 seaborn 库提供了交互式界面，便于用户做出各种有吸引力的统计图表。

雷达图可直观比较年龄段之间差异；数据的数量较大，折线图数据显示更为方便且齐全，能直观年龄段与年份之间关系；饼图利用圆内扇形面积来表示数值大小，用于男女性组成部分所占比重及其差距；漏斗图展示两端数据，直观不同出生年代对自杀率影响大小。

【参考文献】

[1]洪居兴,王璐,王虹,祁俊,陈秀明,王世颖.基于Python的情感数据分析[J].青海大学学报,2019,37(05):97-104.

[2]许素,许新华,柏瑶,张盼,黄瑾.基于python的微信公众号关注者数据分析[J].电脑与信息技术,2019,27(05):61-63.

[3]赵雅欣,宁士勇.基于Python的超市020营销数据分析[J].哈尔滨商业大学学报(自然科学版),2019,35(04):431-435.

[4]徐勤亚,蔡继鹏,王星.基于Python的影片数据分析[J].信息技术与信息化,2019(08):113-115.

[5]杨众.基于Python语言的招聘信息可视化分析[J].计算机与网络,2020,46(02):61-64.

[6]刘航.基于Python的重庆二手房爬取及分析[J].电脑知识与技术,2019,15(36):6-7+17.

[7]殷丽凤,张浩然.基于Python网上招聘信息的爬取和分析[J].电子设计工程,2019,27(20):22-26.

[8]许素,许新华,柏瑶,张盼,黄瑾.基于python的微信公众号关注者数据分析[J].电脑与信息技术,2019,27(05):61-63.

[9]王芳.基于Python的招聘网站信息爬取与数据分析[J].信息技术与网络安全,2019,38(08):42-46+57.

[10]常逢佳,李宗花,文静,常逢锦.基于Python的招聘数据爬虫设计与实现[J].软件导刊,2019,18(12):130-133.

[11]曹洁,崔霄.面向新工科的Python数据分析课程内容浅析[J].河南教育(高教),2019(07):95-97.

[12]陆承佳.基于Python的工程图数据分析研究[J].电脑知识与技术,2019,15(18):266-268+273.

[13]苗玥,刘晓勇,金佳妮,李可心.基于Python的医学数据爬取及分析处理[J].信息技术与信息化,2019(04):56-58.

Data analysis of suicide rate in Australia based on Python

[Abstract] Suicide is an important social problem all over the world, which has become an urgent global public health problem. Firstly, the data sets of suicide rate in different countries are obtained, and the acquired data sets are sorted out to screen valuable data; secondly, a large number of data are screened from the data sets in Australia in this study, and the data sets are analyzed pertinently; finally, the impact of gender, age and birth year on suicide rate in Australia and its influencing factors are analyzed. In this analysis, it is concluded that the suicide rate of young people is the highest, the suicide rate of men is 3.5 times higher than that of women. The war in the year of birth and the economy at that time directly affect the suicide rate of the country. The war makes the suicide rate of the country increase, and the economic influence makes people choose to commit suicide under the pressure of the economy. When the per capita GDP is low, it will have an impact on the happiness index of their families Economic crisis, which affects the suicide rate.

[Key words] Python；Commit suicide；Data analysis