



大连理工大学
DALIAN UNIVERSITY OF TECHNOLOGY

大 作 业

课程名称: 可视化分析

学生姓名: 王之义、刘卉、程新皓、游传捷

题 目: 奥运会运动员数据可视化分析

修读方式: 正常

授课教师: 郑晓朋

目 录

引 言	3
1 所用技术	4
2 系统设计	5
2.1 算法设计	5
2.2 数据流分析	5
2.3 算法详细设计	6
2.3.1 读取数据，清洗数据	6
2.3.2 奥运会的参与人数与时间的关系	6
2.3.3 历届奥运会举办地点	6
2.3.4 各个项目的参与人数	7
2.3.5 男女田径项目获奖前十的国家	7
2.3.6 冰球项目参与国家及获奖情况	7
2.3.7 乒乓球项目参与国家及获奖情况	7
2.3.8 运动员身高与体重关系	7
2.3.9 各国历年女子 4×100 米游泳接力情况	8
2.3.10 参与人数前 10 的项目夺奖的身高体重年龄关系	8
2.3.11 平均奖牌数排名前五的国家的历年奖牌占比情况	8
3 系统算法实现	9
3.1 数据预处理	9
3.1.1 数据读取	9
3.1.2 数据清洗	9
3.2 可视化算法代码实现	10
3.2.1 奥运会的参与人数与时间的关系可视化	10
3.2.2 历届奥运会举办地点可视化	10
3.2.3 各个项目的参与人数可视化	10
3.2.4 男女田径项目获奖前十的国家数据可视化	11
3.2.5 冰球项目参与国家及获奖情况可视化	12
3.2.6 乒乓球项目参与国家及获奖情况可视化	13
3.2.7 运动员身高与体重关系可视化	14
3.2.8 各国历年女子 4×100 米游泳接力情况可视化	14

3.2.9	中国历年女子 4×100 米游泳接力情况可视化.....	15
3.2.10	参与人数前 10 的项目夺奖的身高体重年龄关系可视化.....	15
3.2.11	平均奖牌数排名前五的国家的历年奖牌占比情况可视化.....	16
4	可视化结果及分析.....	17
4.1	奥运会的发展.....	17
4.1.1	规模.....	17
4.1.2	参与人数.....	18
4.1.3	奥运会各项目的参与人数分析.....	20
4.2	各国的优势项目.....	21
4.2.1	田径.....	21
4.2.2	乒乓球.....	22
4.2.3	冰球.....	23
4.3	运动员的身材特点.....	23
4.3.1	从总体角度的分析.....	23
4.3.2	对于特定项目.....	25
4.4	各国奥运之路.....	26
4.4.1	各国历届奥运得奖情况.....	26
4.4.2	优势项目.....	27
4.5	参会的国家地理位置分布.....	28
5	结束语.....	29
5.1	成员感悟.....	29
	程新皓的感悟.....	29
	王之义的感悟.....	29
	刘 卉的感悟.....	29
	游传捷的感悟.....	29
5.2	结论.....	30
	参考文献.....	31
	附录 A. 部分关键代码.....	32

引 言

随着大数据时代的到来，出现了海量的数据，高效并且有效的利用这些数据有着重要的作用。人类利用人脑、计算机等工具，在对数据进行处理后的数值结果与现实世界之间建立映射，进而依据这些映射结果采取行动，从而达到利用这些数据的目的。这个时候有人可能会说机器学习是“机器”学得“知识”，但在事实上，几乎所有机器学习的过程都意味着一系列的判定过程^[1]，无论是监督式学习（回归），还是非监督式学习（分类），都是为了给现实生活中的人类提供行动某些参考。参考这些结果，人类可以决定一些行动策略，进而建立起抽象的海量数据与现实生活之间映射。

如何建立起这种映射呢？**可视化是一个强有力的工具。**

在人类的发展史乃至进化史上，图形的历史比文字和数字长的多。文字的历史可以分为图画文字-表意文字-表音文字几个阶段^[2]，由此可见，人类对于图形的认知往往可以跨越知识的桎梏，不同知识背景的人进行交流，一个内涵丰富的可视化案例往往胜过结构严密的文字甚至是公式推导。因此，可视化是分析的基础，它是数据分析者和广大受众对于一个数据案例的最基本、最直观的认识。将数据可视化之后，人们可以发挥人类特有的创造力和天马行空的想象力，摆脱文字信息的限制，得到意想不到的信息。将数据可视化之后，可视化的结果可以用于更加高层次的数据分析，**可视化是一切数据分析的基础，分析是可视化的灵魂。**

奥运会是人类体育领域的殿堂，它代表着人类体育竞技的最高水平。在大数据背景下，将奥运会数据可视化，对评估运动员的竞技状态、各国体育事业的发展情况乃至人类现阶段的身体机能有着重要的意义。本案例不求面面俱到（例如分析所有国家的所有项目），希望运用一些基本的可视化方法，得出一些有意思的结论。

1 所用技术

Python 是一种面向对象的高级语言，兼具函数化语言编程和高级语言编程的特点，语法的规则较为简洁、自然，其开源性赋予它广阔的应用和发展前景。并且随着 Python 的发展，许多非常实用的工具包逐渐产生，可以极大地方便我们编程。Python 在数据计算和分析中，可以被作为一个强有力的工具，Python 拥有动态类型系统和垃圾回收功能，能够自动管理内存使用，并且支持多种编程范式，包括面向对象、命令式、函数式和过程式编程。其本身拥有一个巨大而广泛的标准库^[3]。

Jupyter 是一个免费的，开源的，交互式的网络工具，被称为**计算笔记本**，研究人员可以使用它将软件代码，计算输出，解释性文本和多媒体资源组合在一个文档中。著名的计算机工作者 Péraze 说，在某种程度上，网络软件的增长和改进推动了 Gmail 和 Google Docs 等应用程序的发展^[4]。计算笔记本已经存在了几十年，但尤其是 Jupyter 在过去几年中已经大受欢迎。这一快速发展得到了热情的用户开发者社区和重新设计的架构的帮助，该架构允许笔记本编写几十种编程语言，为程序员随时随地收集自己的灵感创造了良好的条件。

Matplotlib 是 Python 进行可视化分析重要而有力的工具，它是 NumPy 的可视化工具之一，它提供了一个面向对象的 API，用于使用通用 GUI 工具包（如 Tkinter, wxPython, Qt 或 GTK +）将绘图嵌入到应用程序中。还有一个基于状态机（如 OpenGL）的过程“pylab”接口，设计与 MATLAB 非常相似，但是就专业性而言，Matplotlib 可能比不上 MATLAB 等软件，但是因其开源性，加之近些年来随着计算机硬件的不断发展，Matplotlib 取得了巨大的发展。Pyplot 是一个 Matplotlib 模块，它提供类似 MATLAB 的接口。^[5] Matplotlib 被设计为与 MATLAB 一样可用，具有和 Python 结合使用的能力，以及免费和开源的优势。

将 HTML 动态渲染技术用于数据可视化过程当中，有助于更加直观、准确的分析数据，获取信息，相比于传统的导出图片形式，HTML 可以适配屏幕分辨率，动态地显示一张图片。

2 系统设计

2.1 算法设计

本案例的算法设计如图 2.1.1 所示：

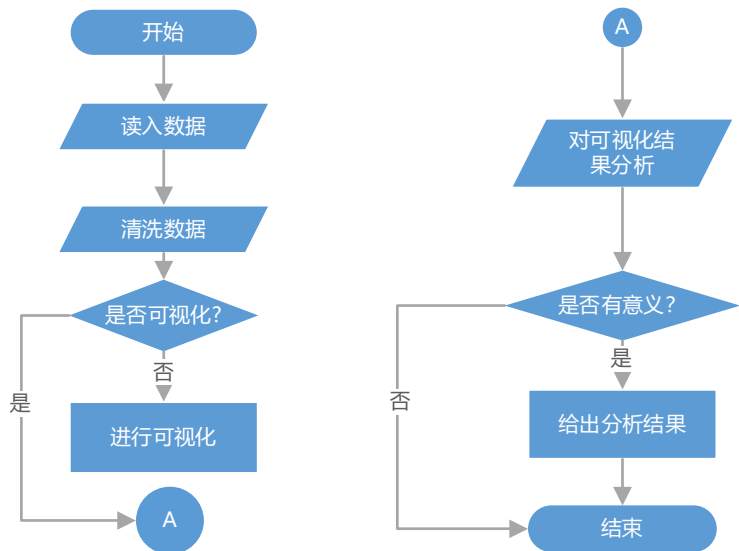


图 2.1.1 系统算法流程图

2.2 数据流分析

数据流动过程图 3.1.2 所示。

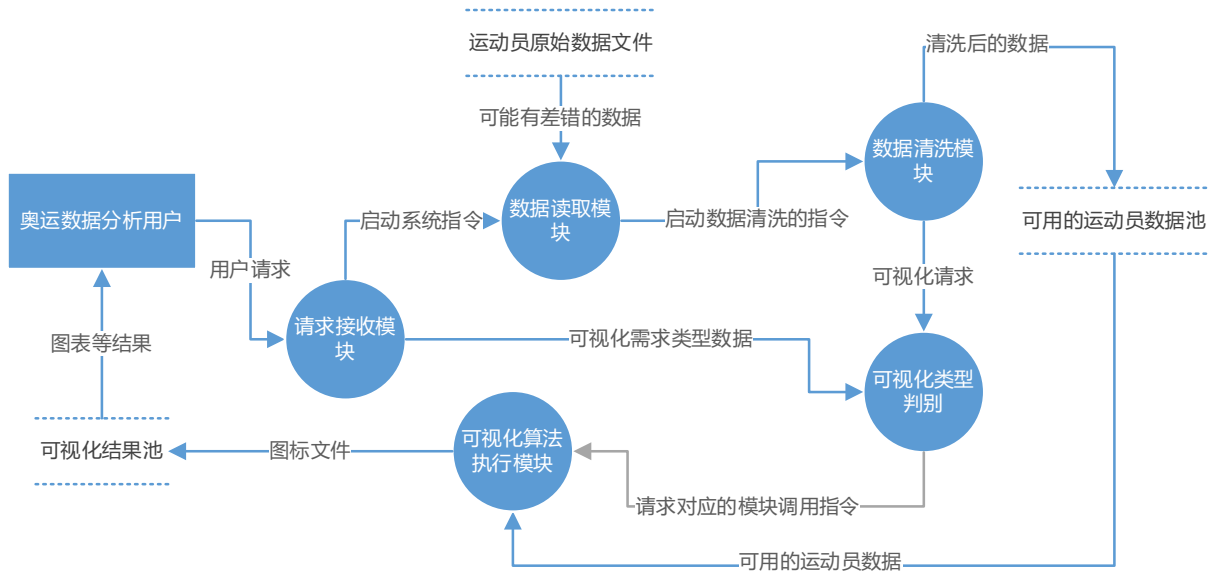


图 2.1.2 系统数据流图

2.3 算法详细设计

由于本案例是是一个可视化案例，借助了一些现有的工具库，囿于篇幅，此处只描述除了调用工具包以外的算法设计。

2.3.1 读取数据，清洗数据

Input: 数据集"Athlete.csv"
Output: 清洗后的数据集
1. **if** (data==NaN)
2. delete(data)
3. **i** 将 data 存入 memory 中，供进一步的分析。
4. **#**显示数据的基本信息。
5. 利用 data.info() 查看数据信息。

2.3.2 奥运会的参与人数与时间的关系

Input: 清洗后的数据集"Athlete.csv"
Output: 可视化结果—折线图
1. 将数据集"Athlete.csv"的"Summer"、"Winter"项目按照"Games"的个数聚集得到"GamesTime"项目集合。
2. 统计"GamesTime"的"Year"的数量。
3. 输出可视化结果，按照折线图输出。横轴为年份，纵轴为人数。

2.3.3 历届奥运会举办地点

Input: 清洗后的数据集"Athlete.csv"
Output: 可视化结果—散点图
1. 将数据集"Athlete.csv"的"Summer"、"Winter"项目按照"Games"的个数聚集得到"GamesTime"项目集合。
2. 统计"GamesTime"的"Year"的数量。
3. 输出可视化结果，按照折线图输出。横轴为年份，纵轴为人数。

2.3.4 各个项目的参与人数

Input: 清洗后的数据集"Athlete.csv"的"Sport"项目

Output: 可视化结果-柱状图

1. 将数据集"Athlete.csv"的"Sport"项目计数。
2. 输出可视化结果。横轴为人数，纵轴为项目。

2.3.5 男女田径项目获奖前十的国家

Input: 清洗后的数据集"Athlete.csv"的"Gymnastics"项目

Output: 可视化结果-饼图

1. 将数据集"Athlete.csv"的"NOC"项目为聚集。
2. 将数据集"Athlete.csv"的"Sport"项目聚集
3. 输出可视化结果。横轴为人数，纵轴为项目。

2.3.6 冰球项目参与国家及获奖情况

Input: 清洗后的数据集"Athlete.csv"的"Ice Hockey"项目

Output: 可视化结果--折线图

1. 将数据集"Athlete.csv"的'NOC','Medal'项目为聚集并计数。
2. 输出可视化结果。横轴为人数，纵轴为项目。

2.3.7 乒乓球项目参与国家及获奖情况

Input: 清洗后的数据集"Athlete.csv"的"Table Tennis"项目

Output: 可视化结果--折线图

1. 将数据集"Athlete.csv"的'NOC','Medal'项目为聚集并计数。
2. 输出可视化结果。横轴为人数，纵轴为项目。

2.3.8 运动员身高与体重关系

Input: 清洗后的数据集中的运动员身高体重

Output: 可视化结果--散点图，拟合曲线图

1. 将身高体重数据规范到(-1,1)上。
2. 用3次多项式拟合。
3. 输出可视化结果。

2.3.9 各国历年女子 4×100 米游泳接力情况

Input: 数据集中的"Swimming Women's 4 x 100 metres Medley Relay"项目
Output: 可视化结果表格
1. 将数据按照['Team','Medal','Year']
2. 输出可视化结果

中国历年女子 4×100 米游泳接力情况可视化的方式和此算法类似，不再赘述。

2.3.10 参与人数前 10 的项目夺奖的身高体重年龄关系

Input: 数据集中的"'Sport', 'Age', 'Height', 'Weight'"项目
Output: 可视化结果--气泡图
1. 找到参与人数前十的项目。
2. 气泡图半径最大值为 1，将数据按照排名算出对应半径。
3. 分别绘制气泡，以用不同颜色区分。

2.3.11 平均奖牌数排名前五的国家的历年奖牌占比情况

Input: 数据集中的"'NOC', 'Year'"项目
Output: 可视化结果--折线图以及横轴与年分的映射关系
1. 建立国家名，年份到数组的映射，便于计算和绘图。
2. 数据筛选，筛选前五名。
3. 给出映射关系和可视化结果。

3 系统算法实现

3.1 数据预处理

3.1.1 数据读取

```
1. import pandas as pd
2. import matplotlib.pyplot as plt
3. import numpy as np
4. from pylab import *
5. mpl.rcParams['font.sans-serif']=['SimHei']
6. mpl.rcParams['axes.unicode_minus']=False
7. #让中文字正确显示
8. data=pd.read_csv("Athlete.csv")
```

3.1.2 数据清洗

```
1. data.dropna(inplace=True)
```

3.2 可视化算法代码实现

3.2.1 奥运会的参与人数与时间的关系可视化

```

1. GamesTime=data[data.Season=="Summer"].groupby('Games').count()
2. x=x.index
3. y1=GamesTime.Year
4. GamesTime=data[data.Season=="Winter"].groupby('Games').count()
5. x2=data[data.Season=="Winter"].groupby('Year').count()
6. x2=x2.index
7. y2=GamesTime.Year
8. fig=plt.figure(figsize=[15,10])
9. plt.title("奥运会参与人数")
10. plt.plot(x2,y2,'-x',label='Winter Olympic Game')
11. plt.plot(x,y1,'-*',label='Summer Olympic Game',color='r')
12. plt.legend()
13. plt.show()

```

3.2.2 历届奥运会举办地点可视化

```

1. fig = plt.figure(figsize=[15,15])
1. plt.scatter(data.Year,data.City)
1. plt.tick_params(labelsize=20)

```

3.2.3 各个项目的参与人数可视化

```

1. Sports=data.Sport
2. Num=Sports.value_counts()
3. y=Num
4. x=Num.index
5. fig=plt.figure(figsize=[10,30])
6. plt.title("奥运会项目参与人数",size=14)
7. plt.barh(x,y,color=['#F8696B','#FB9674','#FCA577','#FDC47D','#FEC97E','#F
ECA7E','#FECC7E','#FEC7F','#FEC7F','#FED07F','#FED07F','#FED17F','#FED280
','#FED680','#FED981','#FFDB81','#FFDC81','#FFDC82','#FFDC82','#FFE183','#F
FE283','#FFE283','#FFE483','#FFE784','#FFE784','#FFE84','#FFE84','#FFEB84
','#FAE983','#ECE582','#D8DF81','#D4DE81','#C4DA80','#C4DA80','#C1D980','#C
1D980','#BCD780','#B7D67F','#B7D67F','#B1D47F','#ADD37F','#ADD37F','#AAD27F
','#9ACE7E','#93CB7D','#8AC97D','#87C87D','#72C27B','#72C27B','#72C27B','#6
FC17B','#66BE7B','#64BE7B','#63BE7B','#63BE7B'])
8. plt.show()

```

3.2.4 男女田径项目获奖前十的国家数据可视化

```

1. need = ['NOC', 'ID']
2. dataN = data.loc[:, np.isin(data.columns, need)]
3. dataSM = dataN.loc[np.array(data['Sport']=='Gymnastics')+
    np.array(data['Sex']=='M')]
4. dataSF = dataN.loc[np.array(data['Sport']=='Gymnastics')+
    np.array(data['Sex']=='F')]
5. #=====男性=====
6. tmpM = dataSM.groupby(by=['NOC']).count().sort_values(by=['ID'],
    ascending=False)
7. N = np.array(tmpM.axes[0].values)
8. V = np.array(tmpM['ID'])
9. V = V/V.sum()
10. for i in np.arange(12,V.shape[0],1):
11.     V[11] += V[i]
12. N[11] = '其他'
13. V = V[:12]
14. N = N[:12]
15. for i in range(12):
16.     N[i] += '\n' + '%.2f%%'%(V[i] * 100.0)
17. plt.title('男性田径项目国家获奖前十占比\n\n')
18. plt.pie(V, labels=N, startangle=50, explode=np.ones(len(V))/10.0)
19. plt.show()
20. #=====女性=====
21. tmpF = dataSF.groupby(by=['NOC']).count().sort_values(by=['ID'],
    ascending=False)
22. N = np.array(tmpF.axes[0].values)
23. V = np.array(tmpF['ID'])
24. V = V/V.sum()
25. for i in np.arange(12,V.shape[0],1):
26.     V[11] += V[i]
27. N[11] = '其他'
28. V = V[:12]
29. N = N[:12]
30. for i in range(12):
31.     N[i] += '\n' + '%.2f%%'%(V[i] * 100.0)
32. plt.title('女性田径项目国家获奖前十占比\n\n')
33. plt.pie(V, labels=N, startangle=50, explode=np.ones(len(V))/10.0)
34. plt.show()

```

3.2.5 冰球项目参与国家及获奖情况可视化

```
1. #冰球项目参与国家及获奖情况可视化
2. MadelRank=data[data.Sport=="Ice Hockey"]
3. MadelRank
4. MadelRank=MadelRank.groupby(['NOC','Medal']).count()
5. MadelRank.Team
6. MadelRank=MadelRank.Team
7. x=MadelRank.index.levels[0]
8. y1=[169,21,22,0,0,6,0,0,0,46,0,123,55]
9. y2=[33,24,0,129,18,0,1,21,19,84,57,10,23]
10. y3=[79,0,0,45,0,0,0,22,0,59,57,20,158]
11. fig=plt.figure(figsize=[16,6])
12. plt.title("冰球项目参与国家及获奖情况")
13. plt.plot(x,y1,0.5,label='Gold',color='g',)
14. plt.plot(x,y2,0.5,label='Bronze',color='r')
15. plt.plot(x,y3,0.5,label='Silver',color='b')
16. for a,b in zip(x,y1):
17.     plt.text(a, b,'%0f' % b ,ha='center', va= 'bottom',
fontsize=15,color='g')
18. for a,b in zip(x,y2):
19.     plt.text(a, b,'%0f' % b ,ha='center', va= 'bottom',
fontsize=15,color='r')
20. for a,b in zip(x,y3):
21.     plt.text(a, b,'%0f' % b ,ha='center', va= 1.
'bottom',fontsize=15,color='b')
22. plt.legend(loc='upper center')
23. plt.show()
```

3.2.6 乒乓球项目参与国家及获奖情况可视化

```
1. #乒乓球项目参与国家及获奖情况可视化
2. MadelRank=data[data.Sport=="Table Tennis"]
3. MadelRank
4. MadelRank=MadelRank.groupby(['NOC','Medal']).count()
5. MadelRank.Team
6. MadelRank=MadelRank.Team
7. x=MadelRank.index.levels[0]
8. y1=[169,21,22,0,0,6,0,0,0,46,0,123,55]
9. y2=[33,24,0,129,18,0,1,21,19,84,57,10,23]
10. y3=[79,0,0,45,0,0,0,22,0,59,57,20,158]
11. fig=plt.figure(figsize=[16,6])
12. plt.title("冰球项目参与国家及获奖情况")
13. plt.plot(x,y1,0.5,label='Gold',color='g',)
14. plt.plot(x,y2,0.5,label='Bronze',color='r')
15. plt.plot(x,y3,0.5,label='Silver',color='b')
16. for a,b in zip(x,y1):
17.     plt.text(a, b,'%0f' % b ,ha='center', va= 'bottom',
fontsize=15,color='g')
18. for a,b in zip(x,y2):
19.     plt.text(a, b,'%0f' % b ,ha='center', va= 'bottom',
fontsize=15,color='r')
20. for a,b in zip(x,y3):
21.     plt.text(a, b,'%0f' % b ,ha='center', va= 1.
'bottom',fontsize=15,color='b')
22. plt.legend(loc='upper center')
23. plt.show()
```

3.2.7 运动员身高与体重关系可视化

```

1. fig = plt.figure(figsize=[15,10])
2. #定义 x、y 散点坐标
3. xWeight=data.Weight
4. yHeight=data.Height
5. x=xWeight.values
6. #print(x.T)
7. y=yHeight.values
8. #用 3 次多项式拟合
9. f1 = np.polyfit(x, y, 3)
10. p1 = np.poly1d(f1)
11. #print(p1)
12. #也可使用 yvals=np.polyval(f1, x)
13. #yvals = p1(TMP) #拟合 y 值
14. #print(yvals)
15. x=xWeight.values.reshape(-1,1)
16. y=yHeight.values.reshape(-1,1)
17. #绘图
18. z=array(range(int(x.min()),int(x.max())))
19.
plt.scatter(x,y,color=['#F8696B','#FB9674','#FCA577','#FDC47D','#FEC97E','#FEC47E','#FECC7E','#FEC7F','#FEC7F','#FED07F','#FED07F','#FED17F','#FED280','#FED680','#FED981','#FFDB81','#FFDC81','#FFDC82','#FFDC82','#FFE183','#FFE283','#FFE283','#FFE483','#FFE784','#FFE784','#FFE884','#FFE884','#FEB84','#FAE983','#ECE582','#D8DF81','#D4DE81','#C4DA80','#C4DA80','#C1D980','#C1D980','#BCD780','#B7D67F','#B7D67F','#B1D47F','#ADD37F','#ADD37F','#AAD27F','#9ACE7E','#93CB7D','#8AC97D','#87C87D','#72C27B','#72C27B','#72C27B','#6FC17B','#66BE7B','#64BE7B','#63BE7B','#63BE7B'],s=4,alpha=0.6)
20. plot2 = plt.plot(z,p1(z), color='#8497B0',linewidth=4,label='polyfit values')
21. plt.xlabel('x')
22. plt.ylabel('y')
23. plt.legend(loc=4) #指定 legend 的位置右下角
24. plt.title('身高与体重的关系',size=24)
25. plt.show()

```

3.2.8 各国历年女子 4×100 米游泳接力情况可视化

```

1. SWMMR=data[data.Event=="Swimming Women's 4 x 100 metres Medley Relay"]
2. SWMMR=SWMMR.groupby(['Team','Medal','Year']).count()
3. SWMMR

```

3.2.9 中国历年女子 4×100 米游泳接力情况可视化

```
1. CHN=data[data.NOC=="CHN"]
2. CHN_Swimming=CHN[CHN.Sport=="Swimming"]
3. CHN_Swimming[CHN_Swimming.Event=="Swimming Women's 4 x 100 metres Medley Relay"]
```

3.2.10 参与人数前 10 的项目夺奖的身高体重年龄关系可视化

```
1. need = ['Sport', 'Age', 'Height', 'Weight']
2. dataN = data.loc[:, np.isin(data.columns, need)]
3. tmp = dataN.groupby(by=['Sport']).mean()
4. Ctmp = dataN.groupby(by=['Sport']).count().sort_values(by=['Age'], ascending=False)['Age'].iloc[:10].axes[0].values
5. X = np.array(tmp['Age'])
6. Y = np.array(tmp['Height'])
7. R = np.array(tmp['Weight'])
8. minR = R.min()-10
9. maxR = R.max()
10. R = (R-minR)/(maxR-minR)*400
11. N = np.array(list(tmp.axes[0].values))
12. X = X[np.isin(N, Ctmp)]
13. R = R[np.isin(N, Ctmp)]
14. Y = Y[np.isin(N, Ctmp)]
15. fig = plt.figure(figsize=[12,8])
16. for i in range(len(X)):
17.     plt.scatter(X[i], Y[i], R[i], alpha=0.9, edgecolors='k', label=N[i])
18. plt.title('前十热度项目夺奖的身高体重年龄关系',size=12)
19. plt.xlabel('年龄')
20. plt.ylabel('身高（半径为体重）')
21. plt.tick_params(top = 'off', right = 'off')
22. plt.legend()
23. plt.show()
```


3.2.11 平均奖牌数排名前五的国家的历年奖牌占比情况可视化

```
1. need = ['NOC', 'Year']
2. fig = plt.figure(figsize=[12,8])
3. dataN = data.loc[:, np.isin(data.columns, need)]
4. N = np.array(dataN.groupby(by=['NOC']).count().axes[0].values)
5. Y = np.array(dataN.groupby(by=['Year']).count().axes[0].values)
6. IN = dict()
7. IY = dict()
8. for n in range(len(N)):
9.     IN[N[n]] = n
10. for y in range(len(Y)):
11.     IY[str(Y[y])] = y
12. V = np.zeros([N.shape[0], Y.shape[0]])
13. for i in range(dataN.shape[0]):
14.     t = dataN.iloc[i]
15.     V[IN[t['NOC']], IY[str(t['Year'])]] += 1
16. S = V.mean(axis=1) > 32
17. TOT = V.sum(axis=0)
18. for i in range(TOT.shape[0]):
19.     for j in range(S.shape[0]):
20.         V[j, i] /= TOT[i]
21. for i in range(S.shape[0]):
22.     if(S[i]):
23.         plt.plot(range(35), np.reshape(V[i], [-1]), label=N[i])
24. print(IY)
25.
26. plt.title('平均奖牌前 5 国家的历年奖牌占比情况')
27. plt.xlabel('年份')
28. plt.ylabel('奖牌占比')
29. plt.legend()
30. plt.show()
```

4 可视化结果及分析

4.1 奥运会的发展

4.1.1 规模

奥运会举办年份及地点：

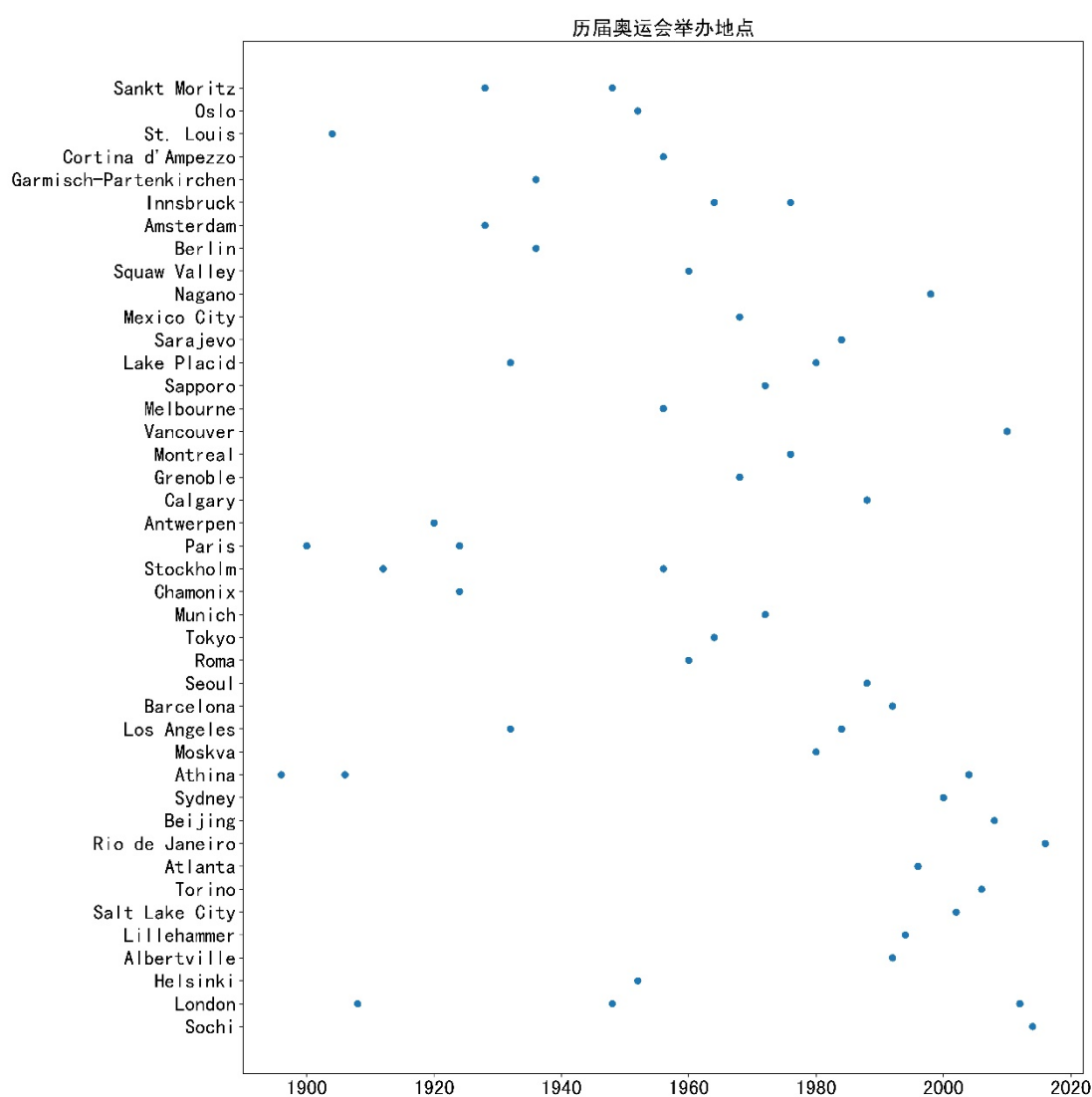


图 4.1.1 奥运会举办年份及地点

由图 4.1.1 可知，我们可以看出奥运会举办的地点还是较为均衡的，不会每年都在一个同一个国家的城市举办，同时也有之后才开始举办奥运会的地区。并且绝大多数国家都有举办冬奥会和夏季奥运会历史，这个图也方便了我们预测下一次奥运会的举办地

点。并且，奥运会举办的城市往往是经济水平和国际影响力比较高的国家的重要城市。没有一个城市连续两届举办奥运会，这也符合奥运会的组织惯例。

到目前为止，夏季奥运会已经举办了 31 届（不包括二战一战期间）冬季奥运会举办了 21 届。第 29 届夏季奥林匹克运动会（Games of the XXIX Olympiad），又称 2008 年北京奥运会，于 2008 年 8 月 8 日晚上 8 时整在中华人民共和国首都北京举办。

一个国家举办下一次奥运会时间相隔最短的是美国，即 1984 年洛杉矶到 1996 年亚特兰大，一是因为 1984 年奥运会申办城市不多，二是美国极为独特，没有第二个国家有如此完备的城市和举办奥运会的设施。其它国家伦敦从 1948 年到 2012 年历经 64 年，澳大利亚从 1956 年墨尔本到 2000 年 44 年。还有一些重要国家一直未能实现再申办成功，如巴黎 1924 年至今数度要求举办至今没有实现，日本 1964 年举办后大阪也正式提出申请未获成功，奥运会每个洲的举办权轮换一次基本要十几年，亚洲间隔时间更长（1964 年东京奥运会—1988 年汉城奥运会间隔 24 年，1988 年汉城奥运会—2008 年北京奥运会历时 20 年）。

4.1.2 参与人数

冬季/夏季奥运会参与人数及年份分析：

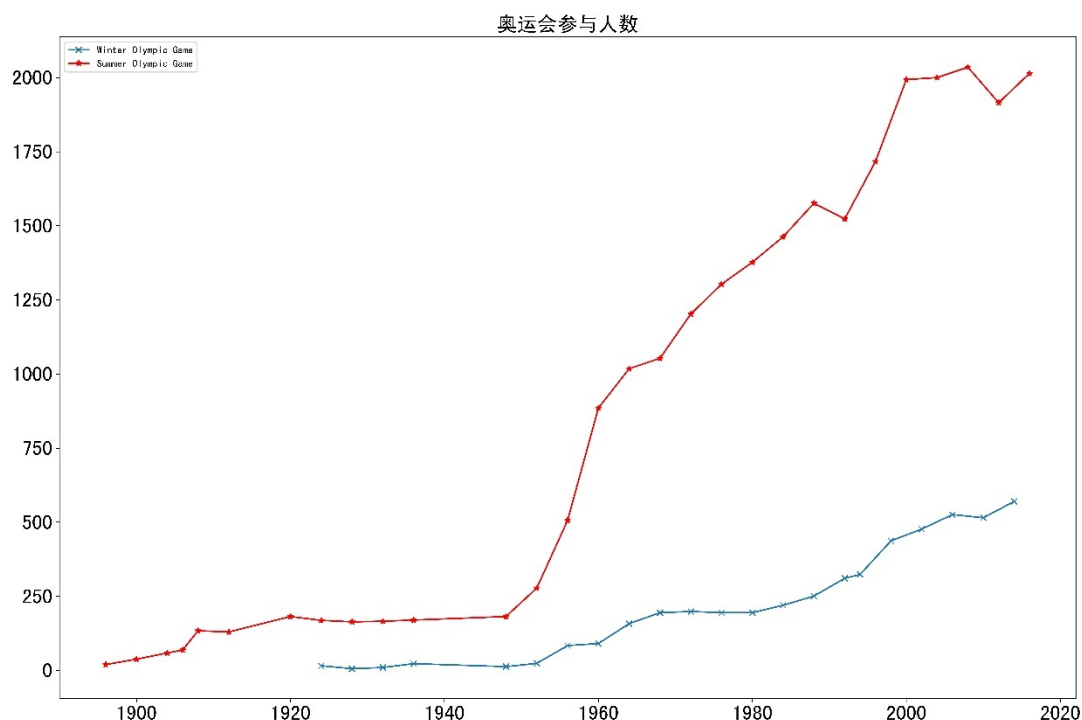


图 4.1.2 历届奥运会参与人数

从图 4.1.2 中我们可以看到，夏季奥运会的参加人数增长的比较迅速，冬奥会相比于夏季奥运会人数较少但也在逐年增加，但增加相对缓慢。说明随着世界发展，人们对于奥运的参与逐渐趋于理性。

冬季奥运会相比于夏季奥运会，参与的规模约为夏季奥运会的 1/3 左右，这也和具备冬奥会训练条件的国家数量较少有关。

值得注意的是，在 1900—1945 年之间，即第一次，第二次世界大战期间，奥运会的参与人数为 250 人左右，并且在 1940 和 1944 年因为战争，并没有如期举办奥运会，说明当时的社会条件下，人们对与奥运会的参与度并远不及今天的水平。

战后，1948-1960 年之间，奥运会的参与人数激增，说明战后各国百废待兴，积极建设，初有成效，可以借助奥运会提升自己的国际影响力。

以上体现了奥运会作为一个人类竞技体育的最高规格盛会，除了体现出体育发展水平外，它还诉说着各国乃至世界的历史发展进程。奥运会的兴衰史体现着世界各国在国际舞台的竞争力的变化史，是和平年代国家体现实力的重要舞台。

夏季奥运会从 1904 年起，夏季奥运会的每个项目都会颁发奖牌，其中第一名为金牌，第二名为银牌，第三名为铜牌。夏季奥运会取得的成功促使了冬季奥林匹克运动会的产生。而在 1906 年，**届间运动会**于 4 月 22 日至 5 月 2 日在希腊雅典举行。这次综合运动会是在第三届与第四届的奥林匹克运动会之间所举办的，故称为“届间运动会”

（Intercalated Games）；值得一提的是，**但该会成绩没有被国际奥林匹克委员会正式承认。**^[6]

4.1.3 奥运会各项目的参与人数分析

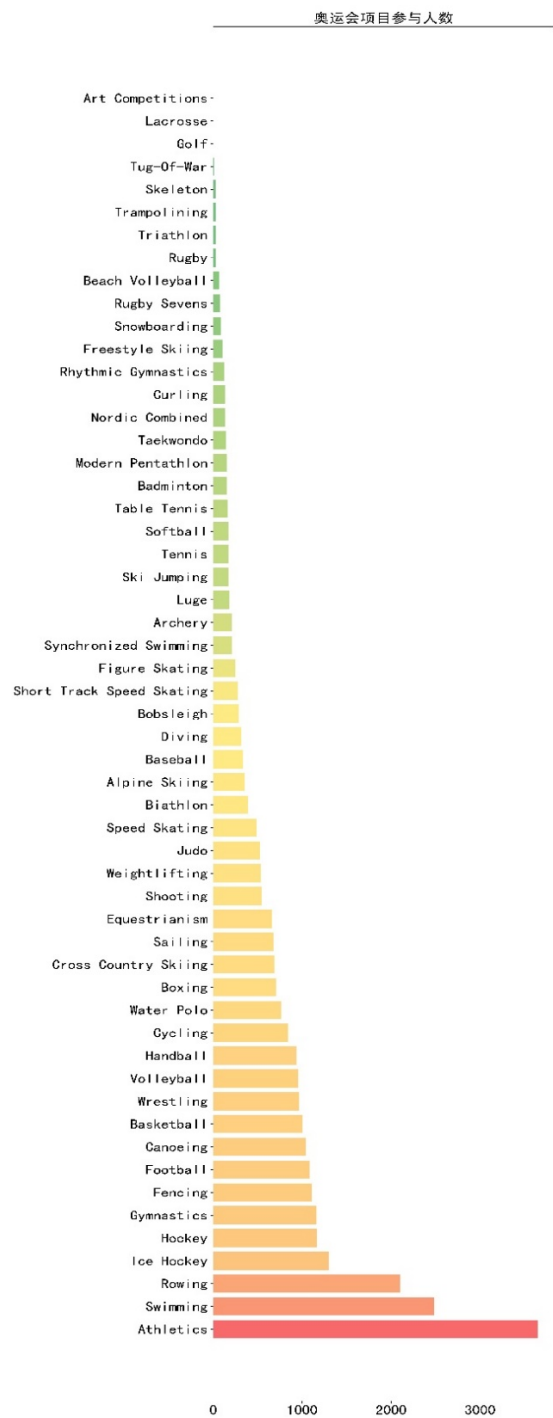


图 4.1.3 奥运会项目参与人数

由图 4.1.3 可知，我们可以看到参赛人数最多的项目是田径运动，游泳运动次之，划船运动排在第三位，而高尔夫、长曲棍球和美术比赛人数很少。

相比之下，奥运会的各大项目中网上搜索的热度排行是足球、乒乓球、网球。田径，游泳和划船参赛人数多首先是因为其没有人数的限制，其次每场比赛所需要的参赛选手很多。

近十年来,为了推动世界田径运动的发展,国际田联采取了一系列措施将田径运动推向市场。从 1985 年起,开始实施系列大奖赛并一直发展到今天的 A, B 两类的大奖赛和总决赛,以及连续七站奖金丰厚的黄金联赛, 并从 1997 年起对室内外室锦赛前三名给予奖励室外冠军 6 万美元, 亚军 3 万美元, 季军 2 万, 室内冠军, 亚军, 季军分别为 5, 2, 1 万美元^[7]。这大大激发了各国运动员的参赛热情, 田径比赛更具观赏性、竞争性, 吸引了更多观众。

4.2 各国的优势项目

4.2.1 田径

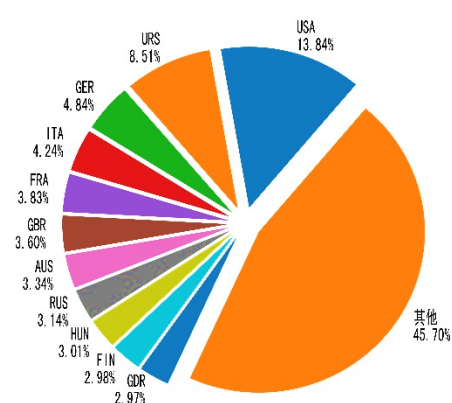


图 4.2.1 男性田径项目国家获奖前十占比

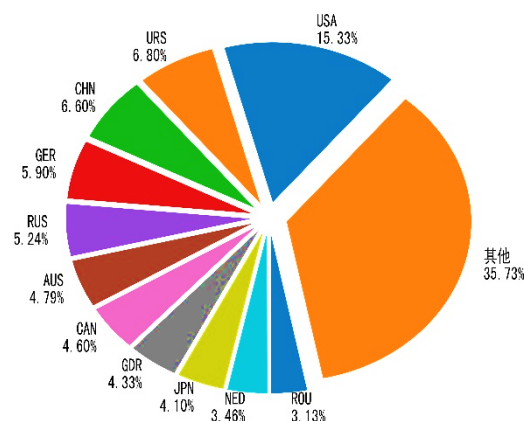


图 4.2.2 女性田径项目国家获奖前十占比

从图 4.2.1 可知，美国在田径项目上具有绝对优势，其余国家则势均力敌。

美国人当中有很大一部分是黑人，黑人身体素质很好已经是公认的了，也是科学的结论了。黑人的肌腱中白肌纤维丰富，比白人和黄色人种要好很多。而肌腱中的白肌纤维是直接影响到爆发力。因此黑人在肌肉力量，爆发力，耐力，弹跳等方面有着先天的优势。^[8]

但自从 2008 年北京奥运会开始，美国在田径这个项目上的优势逐渐减弱，田径比赛的格局也随之发生巨变。但是在短跑长跑这些传统的田径项目上，还是黑人在主导。博尔特，短跑强人。肯尼亚、埃塞俄比亚是中长跑强国。这些运动员都是黑人居多。

4.2.2 乒乓球

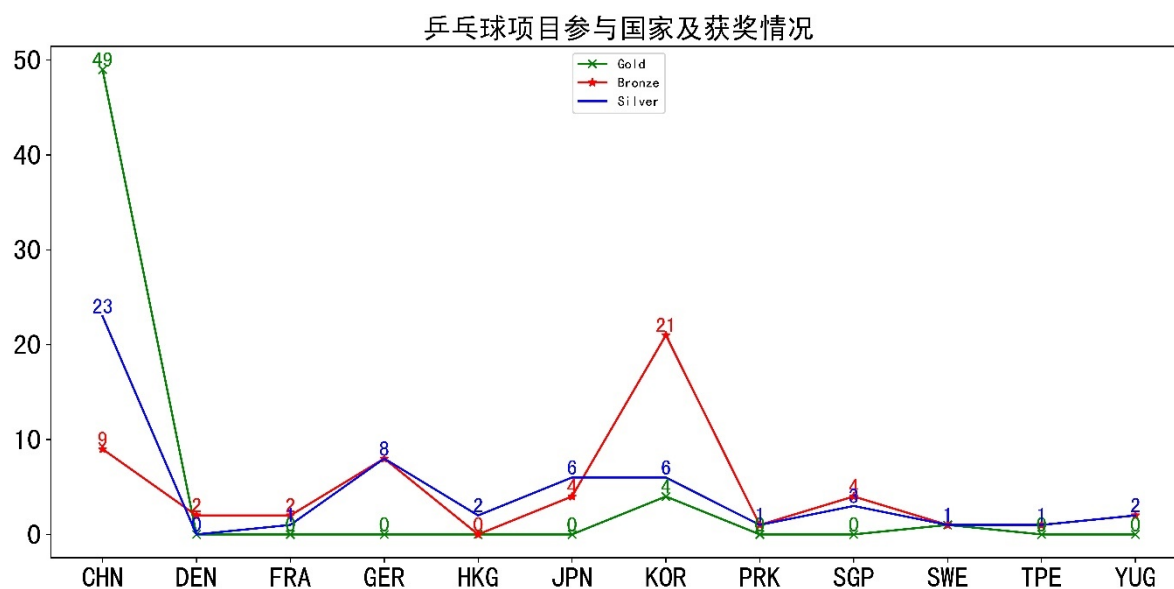


图 4.2.3 乒乓球项目参与国家及获奖情况

由图 4.2.3 可知，中国的乒乓球获奖情况远超其他国家，且几乎拿到了所有的金牌，具有绝对优势，其他的国家拿到的奖牌数量比较平均并且和中国的差距很大。

从上世纪 60 年代起，中国乒乓球对于乒乓球技术发展所做出的贡献显而易见。首使乒乓球成为“国球”，上世纪 90 年代起，中国乒乓球逐渐在世界乒坛建立起霸主地位，世界乒坛中国的主要对手瑞典，韩国，德国偶尔出现一两名天才选手，难以形成对中国队的有效威胁。^[9]

进入新世纪以来，这种情况更加明显，虽然国际乒联采取了诸多针对性极强的限制中国队的措施，但是，反而导致竞争对手们更佳竞争力丧失，每次中国队都能更快适应新规则，取得更佳统治力的霸主地位，以最受重视的奥运会为例，2008，2012，2016 连续三届奥运会，国乒实现了金牌全部包揽，在其余的国际大赛中，也几乎处于同样水平，乒乓球世界杯女子单人比赛单自诞生之日起，就被中国队垄断至今，中国乒乓球相继涌现了邓亚萍，张怡宁，王楠，李晓霞，丁宁，刘国梁，孔令辉，张继科，马龙等全满贯选手，几乎都是自上世纪 90 年代至今获得的。

4.2.3 冰球

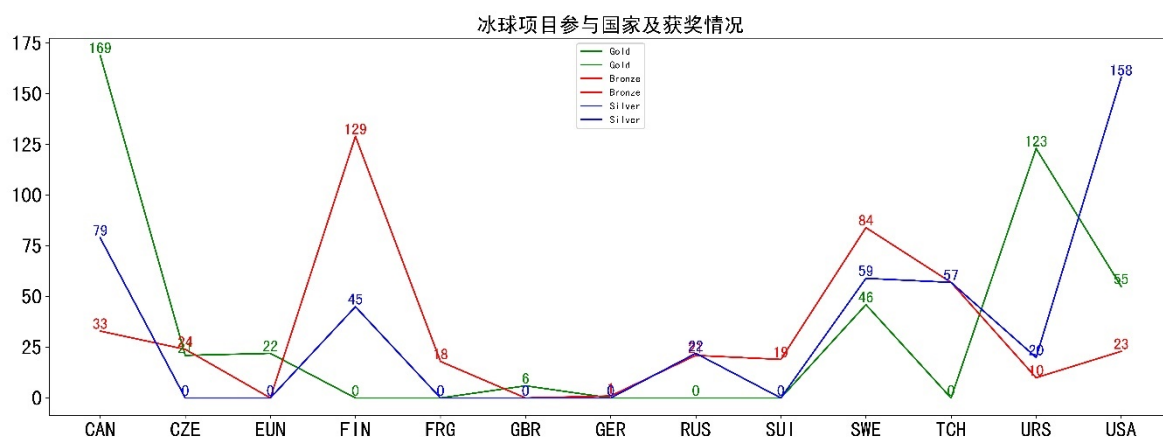


图 4.2.4 冰球项目参与国家及获奖情况

由图 4.2.4 可知，加拿大和于尔拿到的金牌数最多，芬兰拿到的银牌数最多，美国拿到的铜牌数最多，同时可以看出冰球项目的竞争很激烈。

这也体现出区位优势对一个国家优势项目的影响。在几十年前比，大多数国家得天独厚的场地条件（在现代化之前，冬天有天然冰球场的国家不多）所带来的运动练习优势和国家队实力长期全球领先的自豪感，以及以上各点带来的长期群众基础，良性循环了。^[10]

在 19 世纪初期，开始有记载，在加拿大的印第安斯，人们在进行的类似的游戏，他们使用的是棍棒和木质的圆饼。还有一种说法是冰球起源于一种古老的美洲运动——长曲棍球。然而，最广为流传，也是被广泛接受的说法是，冰球由地面上的曲棍球演变而来，曲棍球发源于北欧，已有 500 多年的历史，驻扎在加拿大的英国士兵把这项运动引入北美，发明冰球让加拿大人备感骄傲。

4.3 运动员的身材特点

4.3.1 从总体角度的分析

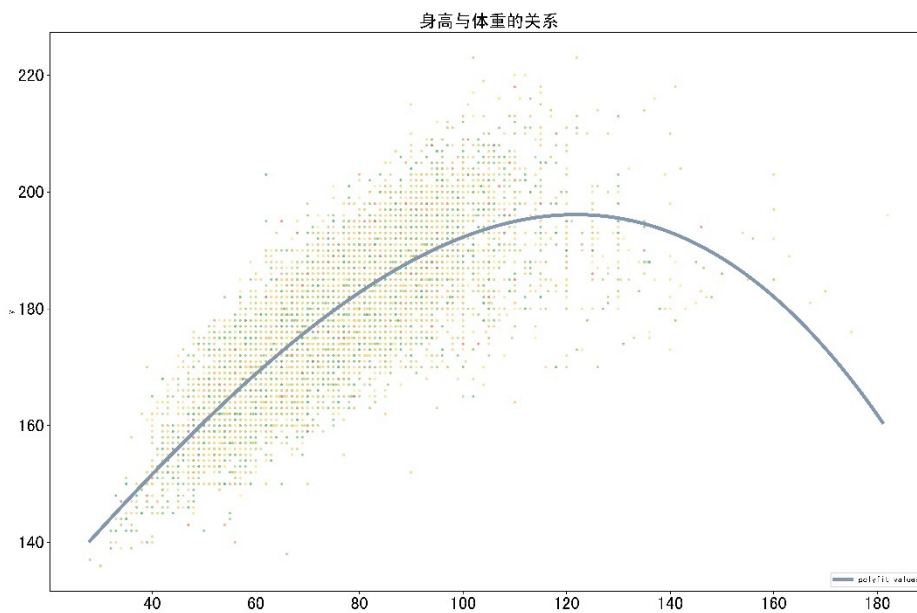


图 4.3.1 运动员身高体重关系

由图 4.3.1 可知，参赛选手的身高与体重大致呈三次函数的关系，大部分的参赛选手所呈现的是一次函数的关系，即随着身高增加，体重增加。同时最高的参赛选手在二米二以上。

世界卫生组织推荐的身高体重关系计算方法：^[11]

男性：(身高 cm－80)×70%=标准体重

女性：(身高 cm－70)×60%=标准体重

体重指数（BMI）的计算方法

BMI=体重（kg）除以身高（m）的平方。（最后得出的指数单位是 kg/m²）

BMI 体重指数算出来的结果：

正常体重： 体重指数 = 18 - 25 （中国体质标准为女性 18-22，男性 20-24）

由此可见，运动员大多都是标准的身材参加运动会的，这样才能获得良好的成绩。

4.3.2 对于特定项目

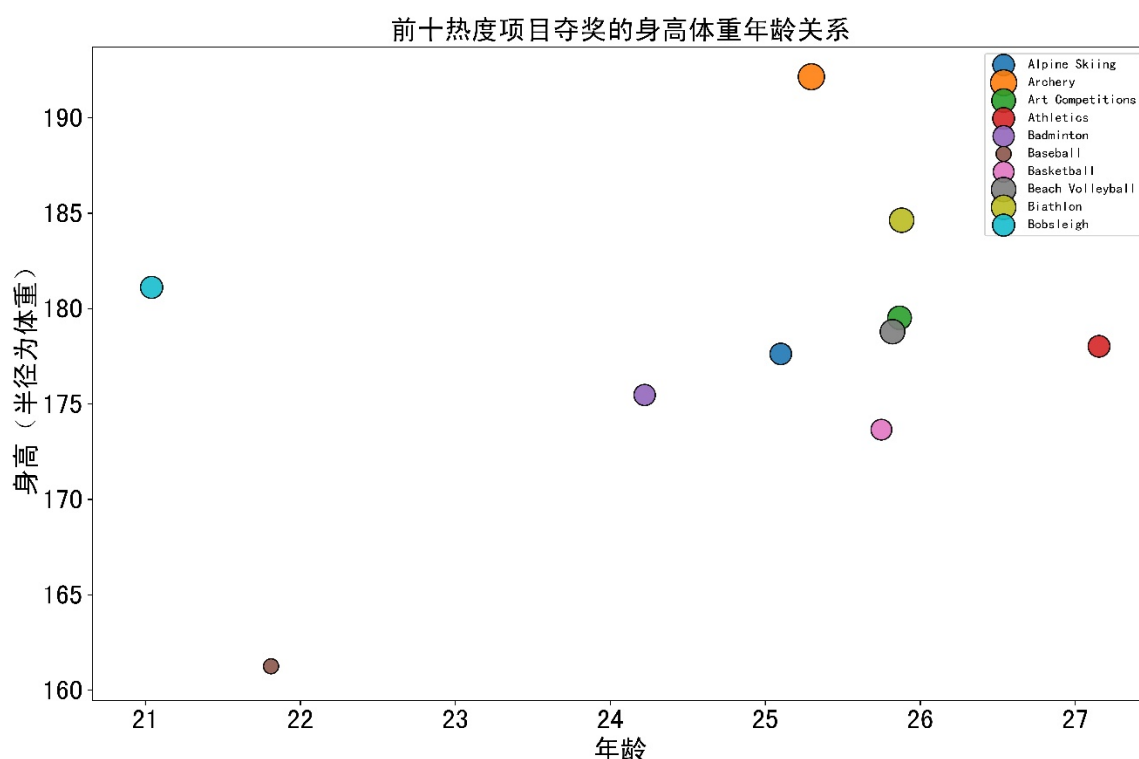


图 4.3.2 前十热度项目夺冠的身高体重年龄关系

由图 4.3.2 可知，我们可以看到大部分的夺冠者年龄在 25 到 26 岁之间，可以推测这一年龄的参赛者参赛经验相对比较丰富，身体达到巅峰，夺冠的可能性较大。

竞技体育的运动员大部份都是 24 岁左右达到巅峰期，而球类运动更是会到 28, 9 岁，甚至 30 岁才达到巅峰期。而真正制约的能力上涨的因素更多来自于身体体力原因。相对的，年纪较大的运动员，由于经验丰富练习时间更长，如除却体力影响肯定在技术上会更上一筹。

每个类别下的运动员黄金年龄是不同的。黄金年龄是在 25 岁以上的。通过对 1986—1994 年世界和中国优秀短跑运动员进行分析比较，运动员保持最佳成绩的运动年限不断延长。30 岁以上获得世界短跑冠军的运动员不断增多。短跑运动员的运动寿命，即保持最佳成绩的年限不断延长。

4.4 各国奥运之路

4.4.1 各国历届奥运得奖情况

使用如下方法，得到历届奥运各国的得奖情况。

```
1. ALL=data.groupby(['Team','Medal','Year']).count()
2. ALL
```

图 4.4.1 为可视化结果。

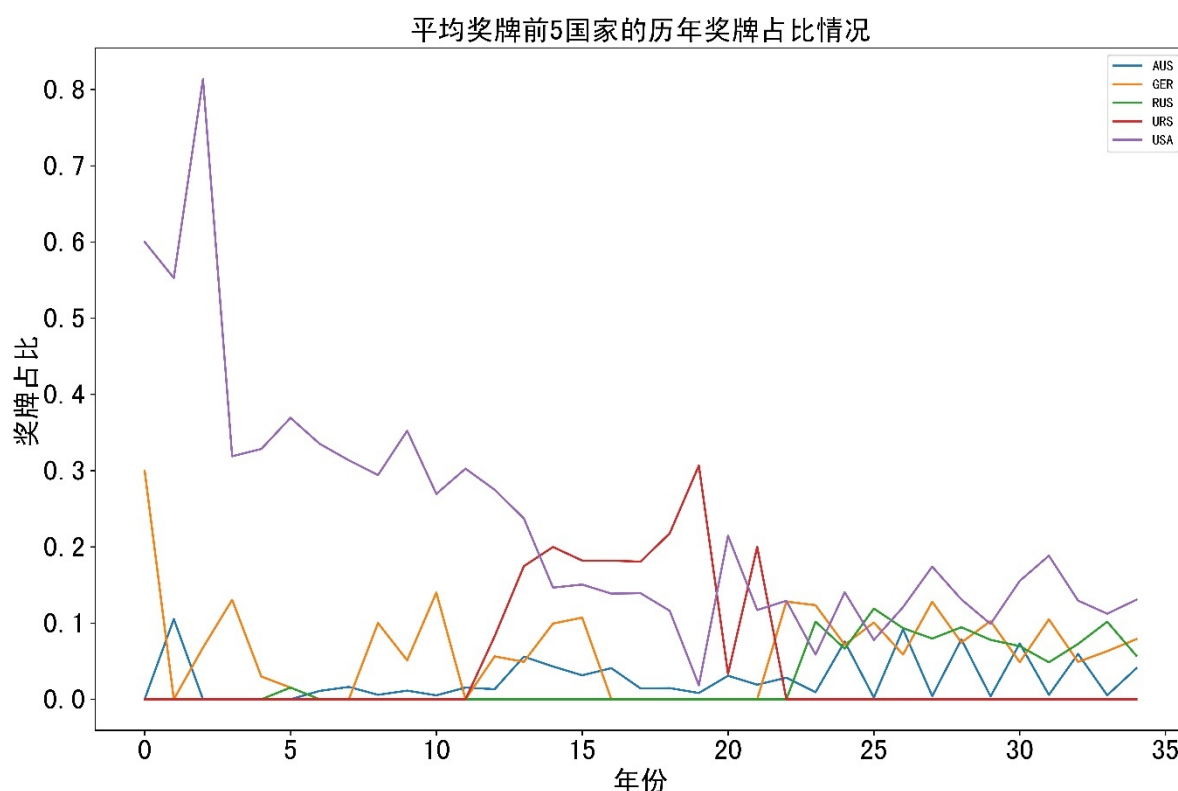


图 4.4.1 平均奖牌前 5 国家的历年奖牌占比情况

由图 4.4.1 可知，以往的美国几乎占据了所有的奖牌，之后各国的奖牌分布比例逐渐趋于平均，可以看出各国对奥运会的重视，以及各国运动水平的提升，换句话讲，竞争力不断加大。

注意，在查阅此图时请参考，实际年份与横轴的映射关系：{'1964': 15, '1904': 2, '2006': 29, '1976': 18, '2002': 27, '1908': 4, '1928': 8, '1906': 3, '2012': 32, '1984': 20, '1956': 13, '1896': 0, '1968': 16, '2016': 34, '2004': 28, '2000': 26, '1912': 5, '1952': 12, '2010': 31, '1924': 7, '1992': 22, '1932': 9, '1980': 19, '1994': 23, '1988': 21, '1936': 10, '1948': 11, '1972': 17, '2014': 33, '1998': 25, '2008': 30, '1996': 24, '1960': 14, '1900': 1, '1920': 6}

4.4.2 优势项目

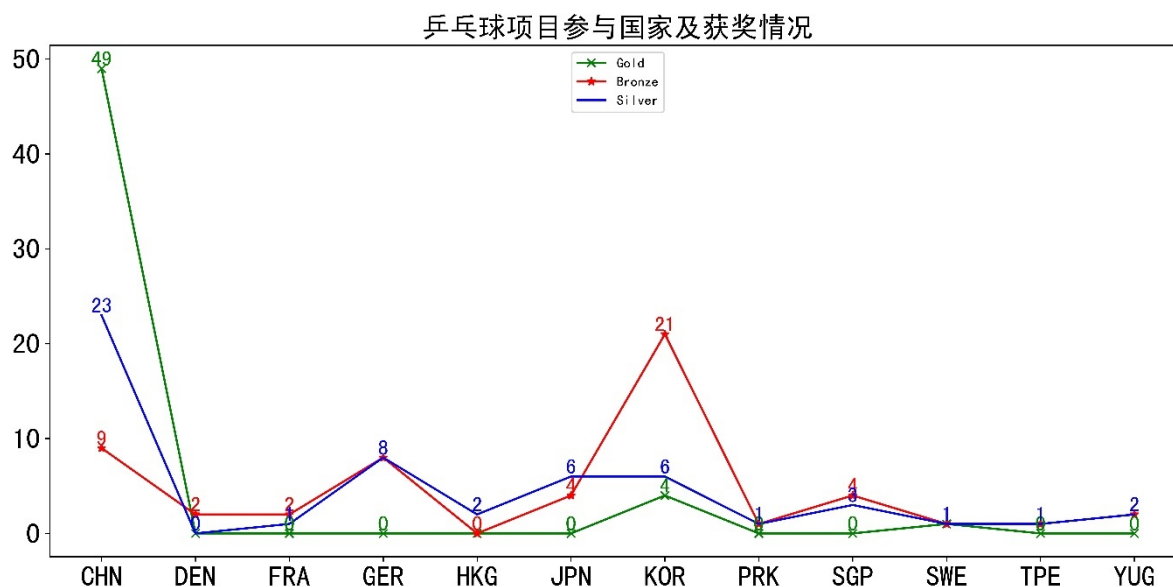


图 4.4.2 乒乓球项目参与国家及获奖情况

由图 4.4.2 可见，在乒乓球项目的优势比较中，奖牌总数最多的国家是中国。除了中国之外，乒乓球金牌的得主国家还有韩国和瑞典，韩国四枚，瑞典一枚。前三名金牌得主金牌数占比分别为 90.7%，7.4% 和 1.9%。银牌数量前三名的得主分别是中国，德国，日本和韩国。银牌占比分别为 42.6%，14.8%，11.1% 和 11.1%。铜牌前三名得主分别为韩国，韩国，中国和德国。奖牌占比分别为 38.9%，16.7% 和 14.9%。在乒乓球项目中，黄色人种占据了明显的优势。

不同运动项目和国家的历史和区位优势有关，例如地处寒带、亚热带的国家的运动员比较擅长冰雪类项目，例如，加拿大、挪威、芬兰等国家在冰雪项目上非常强势。来自亚热带、温带国家的运动员在田径方面取得了骄人的成绩。亚洲国家，例如中国、韩国、日本等国家，在游泳、跳水、乒乓球、羽毛球等比赛中则有着绝对的优势，这些国家的经济水平足以支撑其发展体育事业，并且这些国家的普通百姓往往乐于参加和欣赏这些项目，有着较好的历史和群众基础。

4.5 参会的国家地理位置分布

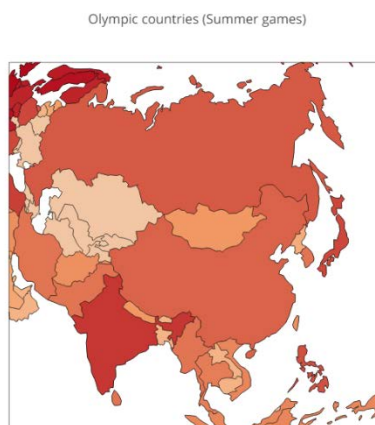


图 4.5.1 亚洲参与情况

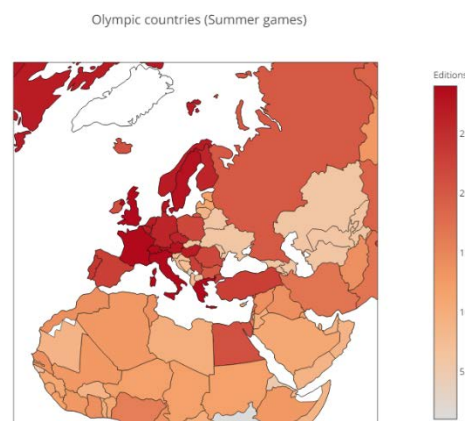


图 4.5.2 欧洲参与情况



图 4.5.3 美洲参与情况

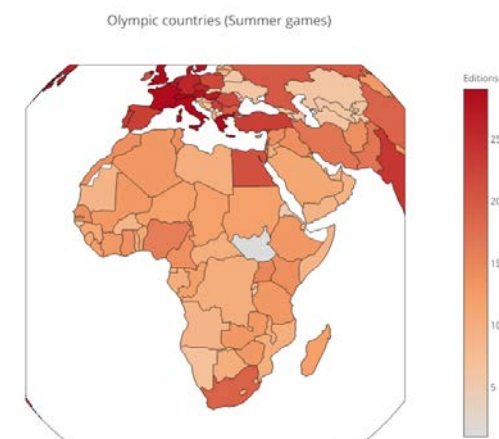


图 4.5.4 非洲参与情况

从以图 4.5.1~4.5.4 中，可以清楚地看到，越发达的国家，例如美国、英国等国，奥运会的参与度非常高，而不能忽略的是人口因素给奥运会参与带来的影响，例如，中国、印度等国，因为人口众多，因而参与奥运会的规模也比较大。但是例如非洲、南美洲的一些国家，经济欠发达，甚至有些国家处于动荡的状态，参与奥运会更是天方夜谭。

以上结果再次证明了奥运会是当今世界各国的实力的体现。采用 HTML 动态渲染的技术，具备一定的交互性，可以将这些结果映射到一个可以旋转的地球上，用户可以点击、缩放等功能而获取有效信息。对于奖牌的分析，方法类似，可视化结果放在附件当中，以供参考。

5 结束语

5.1 成员感悟

程新皓的感悟

本学期开设的可视化分析课程是一门很有用的学科。它不仅仅能够帮忙处理数据，更能提升数据的美观程度，让大家清晰明了的看到数据的结果。通过对本课程的学习，我学会了怎么使用 Python 中的 IDE 来实行数据的可视化，提高了我对可视化的认知，使我受益匪浅。更使我感受颇深的是数据的多样化处理，不同的数据需要不同的处理方式，而很可能有些新的处理方式需要我们去发现。希望我今后的学习中能更加努力。

王之义的感悟

数据可视化是一种广泛运用于不同领域的沟通工具。不同领域有不同的需求，可以说数据可视化已经成为了数据交流不可或缺的工具。有幸这学期我能学习到这么重要的一门课，通过这门课，我认识到了很多可视化的方法，他们帮助了我从一大堆复杂的数据中看到明显的结果。从这门课程中我学会了如何去实现可视化并有了自己初步的成果，希望今后的生活中能发挥用处。

刘 卉的感悟

众所周知，可视化能将数据变化为图形，方便与读者们观察，其重要性不言而喻。学习可视化的过程中，我更深入的理解了这门课程的意义。当我将千行数据变化为一个直观的图像时，我觉得可视化是一件不可或缺的东西。同时我更希望我能更加深入的去了解它使用它，让它便利我的生活。希望更多人能通过可视化的图形来知道它的好处，同时利用好它。

游传捷的感悟

数据可视化通俗说就是要用图形讲故事，大家关心的从来不是数据，数据不是重点，大家关心的是数据背后的意义。可视化也能将有意义的数据进行处理，从而得出有意义的图形方便观测学习。在这个大数据的时代，数据的可视化更为重要，当面对几千万上亿的数据时，不免会感到束手无策，而可视化能很好地让我们看清其中的关系，可视化就是这样一门重要的课程。通过此次大作业，我的收获颇丰，也希望今后能多多接触这一方面。

5.2 结论

通过本案例的分析，深刻地体现出了奥运会丰富的含义和独特的价值。奥运会之所以能从古希腊走向世界，除了它自身的高水准之外，更多的是它为众多国际政治、商业、经济等领域的团体提供了一个展示影响力的平台。你是否发现，为什么美国雄踞奥运会金牌榜首？为什么加拿大的冰雪项目如此强势？

就中国而言，从 1932 年刘长春代表中国参加第十届洛杉矶奥运会，到 2008 年中国成功举办夏季奥运会并且实现金牌榜位居世界第一成绩的漫漫长路上，中国经历的曲折历史是有目共睹的。这也体现着奥运会的历史价值。中国的发展史与奥运的关系可以折射出世界发展史与奥运会的关系，在了解中国历史的情况下，可以管中窥豹，通过这个案例思考奥运会历史变迁的深刻含义。

对于不同体育项目而言，对于场地要求较低的项目，例如田径、游泳等项目，参与人数较多，即使一个国家的经济水平不足以支持运动员训练高尔夫、击剑等项目，他们的运动员仍有机会参与到跑步等项目中去，这也是奥运会人文性的体现。

可视化是一个古老而又常新的话题，原始社会的图腾、象形文字是人们思想的可视化，是一个演绎的过程，而可视化分析则可以将图形变成思想，是一个归纳的过程。可视化分析分为两个部分，“可视化”和“分析”，随着计算机硬件和软件的飞速发展，人们将站在更高的角度去分析问题，可视化为人人可以参与到数据分析中来铺平了道路。

可视化是“朴素的”，并不是体现编程技艺的舞台，对于可视化的结果，要发挥创造力，善于联系生活，这样一定会发现数据背后的知识，获得物外之趣。

参考文献

- [1] 周志华. (2016). *机器学习*. 北京: 清华大学出版社.
- [2] Fischer, S. (2001). *A history of writing* (Globalities). London: Reaktion Books.
- [3] 伊德里斯, & 韩波. (2016). *Python 数据分析*. 北京: 人民邮电出版社.
- [4] Perkel, J. (2018). BY JUPYTER, IT ALL MAKES SENSE. *Nature*, 563(7729), 145-146.
- [5] <https://matplotlib.org/index.html>
- [6] 刘晓树. (2015). *体育的盛宴 奥运会* (全民阅读体育知识百科丛书 quan min yue du ti yu zhi shi bai ke cong shu). 南昌: 二十一世纪出版社.
- [7] 俞继英主编. (2001). *奥运会项目大全 奥林匹克田径*.
- [8] 刘林生, & LIU Lin-sheng. (2005). 世界田径赛场黑人运动员优势成因探析. *德州学院学报(自然科学版)*, 21(4), 100-102.
- [9] 袁玉峰. (2013). 世界乒乓球运动发展现状与趋势研究. *体育文化导刊*, (06), 63-66.
- [10] 郑重. (2015). 勇敢者的游戏——冰球. *青春期健康*, (06), 72-73..
- [11] <http://apps.who.int/gho/data/node.home>

附录 A. 部分关键代码

```
#!/usr/bin/env python
# coding: utf-8

# In[4]:

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from pylab import *
mpl.rcParams['font.sans-serif']=['SimHei']
mpl.rcParams['axes.unicode_minus']=False
#让中文汉字正确显示

# In[5]:

#导入数据
data=pd.read_csv("Athlete.csv")

# In[6]:

#显示数据
data

# In[7]:

#显示数据的基本信息
data.info()

# In[8]:

#清洗数据
data.dropna(inplace=True)
```

```
# In[9]:

#显示数据的基本信息
data.info()

# In[41]:

#初步可视化，显示奥运会的参与人数与时间的关系
GamesTime=data[data.Season=="Summer"].groupby('Games').count()
x=data[data.Season=="Summer"].groupby('Year').count()
x=x.index
y1=GamesTime.Year
y1
GamesTime=data[data.Season=="Winter"].groupby('Games').count()
x2=data[data.Season=="Winter"].groupby('Year').count()
x2=x2.index
y2=GamesTime.Year
fig=plt.figure(figsize=[15,10])
plt.title("奥运会参与人数",size=20)
plt.tick_params(labelsize=20)
plt.plot(x2,y2,'-x',label='Winter Olympic Game')
plt.plot(x,y1,'-*',label='Summer Olympic Game',color='r',)
plt.legend()

plt.tight_layout()
plt.savefig(r'D:\Pythonplace\可视化大作业\图片\奥运会参与人数.jpg',dpi=800)
plt.show()

# In[36]:

#奥运会各个项目的参与人数可视化
Sports=data.Sport
Num=Sports.value_counts()
y=Num
x=Num.index
fig=plt.figure(figsize=[10,30])
plt.tick_params(labelsize=20)
plt.title("奥运会项目参与人数",size=20)
plt.barh(x,y,color=['#F8696B','#FB9674','#FCA577','#FDC47D','#FEC97E','#FEC
A7E','#FECC7E','#FECD7F','#FECD7F','#FED07F','#FED07F','#FED17F','#FED280',
```

```

'#FED680', '#FED981', '#FFDB81', '#FFDC81', '#FFDC82', '#FFDC82', '#FFE183', '#FFE
283', '#FFE283', '#FFE483', '#FFE784', '#FFE784', '#FFEA84', '#FFEA84', '#FFEB84',
'#FAE983', '#ECE582', '#D8DF81', '#D4DE81', '#C4DA80', '#C4DA80', '#C1D980', '#C1D
980', '#BCD780', '#B7D67F', '#B7D67F', '#B1D47F', '#ADD37F', '#ADD37F', '#AAD27F',
'#9ACE7E', '#93CB7D', '#8AC97D', '#87C87D', '#72C27B', '#72C27B', '#72C27B', '#6FC
17B', '#66BE7B', '#64BE7B', '#63BE7B', '#63BE7B']])
plt.tight_layout()
plt.savefig(r'D:\Pythonplace\可视化大作业\图片\奥运会项目参与人
数.jpg', dpi=800)
plt.show()

# In[38]:

# In[50]:

need = ['NOC', 'ID']
dataN = data.loc[:, np.isin(data.columns, need)]
dataSM = dataN.loc[np.array(data['Sport']=='Gymnastics')+
np.array(data['Sex']=='M')]
dataSF = dataN.loc[np.array(data['Sport']=='Gymnastics')+
np.array(data['Sex']=='F')]

#=====男性=====
tmpM = dataSM.groupby(by=['NOC']).count().sort_values(by=['ID'],
ascending=False)
N = np.array(tmpM.axes[0].values)
V = np.array(tmpM['ID'])
V = V/V.sum()
for i in np.arange(12, V.shape[0], 1):
    V[11] += V[i]
N[11] = '其他'
V = V[:12]
N = N[:12]
for i in range(12):
    N[i] += '\n' + '%.2f%%'%(V[i] * 100.0)
plt.tick_params(labelsize=20)
plt.title('男性田径项目国家获奖前十占比\n\n')
plt.pie(V, labels=N, startangle=50, explode=np.ones(len(V))/10.0)

```

```

plt.tight_layout()
plt.savefig(r'D:\Pythonplace\可视化大作业\图片\男性田径项目国家获奖前十占
比.jpg',dpi=800)

plt.show()

#=====女性=====
tmpF = dataSF.groupby(by=['NOC']).count().sort_values(by=['ID'],
ascending=False)
N = np.array(tmpF.axes[0].values)
V = np.array(tmpF['ID'])
V = V/V.sum()
for i in np.arange(12,V.shape[0],1):
    V[11] += V[i]
N[11] = '其他'
V = V[:12]
N = N[:12]
for i in range(12):
    N[i] += '\n' + '%.2f%%'%(V[i] * 100.0)
plt.tick_params(labelsize=20)
plt.title('女性田径项目国家获奖前十占比\n\n')
plt.pie(V, labels=N, startangle=50, explode=np.ones(len(V))/10.0)

plt.tight_layout()
plt.savefig(r'D:\Pythonplace\可视化大作业\图片\女性田径项目国家获奖前十占
比.jpg',dpi=800)

plt.show()

# In[55]:

#冰球项目参与国家及获奖情况可视化
MadelRank=data[data.Sport=="Ice Hockey"]
MadelRank
MadelRank=MadelRank.groupby(['NOC','Medal']).count()
MadelRank.Team
MadelRank=MadelRank.Team
x=MadelRank.index.levels[0]
y1=[169,21,22,0,0,6,0,0,0,46,0,123,55]
y2=[33,24,0,129,18,0,1,21,19,84,57,10,23]
y3=[79,0,0,45,0,0,0,22,0,59,57,20,158]
fig=plt.figure(figsize=[16,6])

```

```

plt.tick_params(labelsize=20)
plt.title("乒乓球项目参与国家及获奖情况",size=20)

plt.plot(x,y1,'-x',label='Gold',color='g')
plt.plot(x,y2,'-*',label='Bronze',color='r')
plt.plot(x,y3,'-',label='Silver',color='b')
for a,b in zip(x,y1):
    plt.text(a, b,'%0f' % b ,ha='center', va=
'bottom',fontsize=15,color='g')
for a,b in zip(x,y2):
    plt.text(a, b,'%0f' % b ,ha='center', va=
'bottom',fontsize=15,color='r')
for a,b in zip(x,y3):
    plt.text(a, b,'%0f' % b ,ha='center', va=
'bottom',fontsize=15,color='b')
plt.legend(loc='upper center')

plt.tight_layout()
plt.savefig(r'D:\Pythonplace\可视化大作业\图片\乒乓球项目参与国家及获奖情
况.jpg',dpi=800)

plt.show()

# In[158]:

#运动员身高体重线性回归
from sklearn import linear_model
fig = plt.figure(figsize=[15,10])
regr = linear_model.LinearRegression()
xWeight=data.Weight
yHeight=data.Height

x=xWeight.values.reshape(-1,1)
y=yHeight.values.reshape(-1,1)

regr.fit(x,y)
plt.scatter(x,y,color=['#F8696B', '#FB9674', '#FCA577', '#FDC47D', '#FEC97E', '#
FECA7E', '#FECC7E', '#FEC7F', '#FEC7F', '#FED07F', '#FED07F', '#FED17F', '#FED28
0', '#FED680', '#FED981', '#FFDB81', '#FFDC81', '#FFDC82', '#FFDC82', '#FFE183', '#
FFE283', '#FFE283', '#FFE483', '#FFE784', '#FFE784', '#FFE884', '#FFE884', '#FFEB8
4', '#FAE983', '#ECE582', '#D8DF81', '#D4DE81', '#C4DA80', '#C4DA80', '#C1D980', '#
C1D980', '#BCD780', '#B7D67F', '#B7D67F', '#B1D47F', '#ADD37F', '#ADD37F', '#AAD27
F', '#9ACE7E', '#93CB7D', '#8AC97D', '#87C87D', '#72C27B', '#72C27B', '#72C27B', '#

```

```

plt.tick_params(labelsize=20)
plt.title("冰球项目参与国家及获奖情况",size=20)

'''plt.barh(y1,x,0.5,label='Gold',color='g',)
plt.barh(y2,x,0.5,label='Bronze',color='r')
plt.barh(y3,x,0.5,label='Silver',color='b')
'''

plt.plot(x,y1,0.5,label='Gold',color='g',)
plt.plot(x,y2,0.5,label='Bronze',color='r')
plt.plot(x,y3,0.5,label='Silver',color='b')

for a,b in zip(x,y1):
    plt.text(a, b,'%0f' % b ,ha='center', va=
'bottom',fontsize=15,color='g')
for a,b in zip(x,y2):
    plt.text(a, b,'%0f' % b ,ha='center', va=
'bottom',fontsize=15,color='r')
for a,b in zip(x,y3):
    plt.text(a, b,'%0f' % b ,ha='center', va=
'bottom',fontsize=15,color='b')
plt.legend(loc='upper center')

plt.tight_layout()
plt.savefig(r'D:\Pythonplace\可视化大作业\图片\冰球项目参与国家及获奖情
况.jpg',dpi=800)
plt.show()

# In[56]:

#乒乓球项目参与国家及获奖情况可视化
MadelRank=data[data.Sport=="Table Tennis"]
MadelRank
MadelRank=MadelRank.groupby(['NOC','Medal']).count()
MadelRank.Team
MadelRank=MadelRank.Team
x=MadelRank.index.levels[0]
y1=[49,0,0,0,0,0,4,0,0,1,0,0]
y2=[9,2,2,8,0,4,21,1,4,1,1,2]
y3=[23,0,1,8,2,6,6,1,3,1,1,2]
fig=plt.figure(figsize=[12,6])

```

```
6FC17B', '#66BE7B', '#64BE7B', '#63BE7B', '#63BE7B'], s=4, alpha=0.6)
#plt.scatter(x,y,color='b',s=4,alpha=0.6)
#plt.scatter(x,y,color='g',s=4,alpha=0.6)
print(regr.predict(x))

alpha = regr.intercept_[0]
beta = regr.coef_[0][0]
plt.plot(x, regr.predict(x), label="$y = 134.58780175890575x + 0.5837624621126675$", color='#8497B0', linewidth=1)
plt.legend()
plt.show()

# In[59]:

fig = plt.figure(figsize=[15,15])
plt.scatter(data.Year, data.City)
plt.title("历届奥运会举办地点", size=20)
plt.tick_params(labelsize=20)

plt.tight_layout()
plt.savefig(r'D:\Pythonplace\可视化大作业\图片\奥运会举办地点.jpg', dpi=800)

plt.show()
#plt.plot([0, 'Beijing'])
...

ArrayA=array(data.Year)
ArrayB=array(data.City)
ArrayC=list(zip(ArrayA, ArrayB))
x=zeros(30181)
y=ArrayB
Point=list(zip(x,y))
plt.plot(x,y)
print(len(ArrayC))
...

#print(ArrayC)

# In[63]:

import numpy as np

import matplotlib.pyplot as plt
fig = plt.figure(figsize=[15,10])
#定义 x、y 散点坐标
```

```
xWeight=data.Weight
yHeight=data.Height
x=xWeight.values
#print(x.T)
y=yHeight.values
#用 3 次多项式拟合
f1 = np.polyfit(x, y, 3)
p1 = np.poly1d(f1)
#print(p1)
#也可使用 yvals=np.polyval(f1, x)
#yvals = p1(TMP) #拟合 y 值
#print(yvals)
x=xWeight.values.reshape(-1,1)
y=yHeight.values.reshape(-1,1)

#绘图
z=array(range(int(x.min()),int(x.max())))
plt.scatter(x,y,color=['#F8696B','#FB9674','#FCA577','#FDC47D','#FEC97E','#F
FECA7E','#FECC7E','#FECD7F','#FECD7F','#FED07F','#FED07F','#FED17F','#FED28
0','#FED680','#FED981','#FFDB81','#FFDC81','#FFDC82','#FFDC82','#FFE183','#
FFE283','#FFE283','#FFE483','#FFE784','#FFE784','#FFEA84','#FFEA84','#FFEB8
4','#FAE983','#ECE582','#D8DF81','#D4DE81','#C4DA80','#C4DA80','#C1D980','#
C1D980','#BCD780','#B7D67F','#B7D67F','#B1D47F','#ADD37F','#ADD37F','#AAD27
F','#9ACE7E','#93CB7D','#8AC97D','#87C87D','#72C27B','#72C27B','#72C27B','#
6FC17B','#66BE7B','#64BE7B','#63BE7B','#63BE7B'],s=4,alpha=0.6)
plot2 = plt.plot(z,p1(z), color='#8497B0',linewidth=4,label='polyfit
values')
plt.xlabel('x')
plt.ylabel('y')
plt.legend(loc=4) #指定 legend 的位置右下角
plt.title('身高与体重的关系',size=20)
plt.tick_params(labelsize=20)

plt.tight_layout()
plt.savefig(r'D:\Pythonplace\可视化大作业\图片\身高与体重的关系.jpg',dpi=800)
plt.show()
```