

How have athletes of different genders, regions, and sports events performed over the years

Lekevia

Software engineering class 1

Abstract

The dataset of this article analyzes Olympic medal data since 1920. Through this analysis, we can gain a glimpse into the historical changes of the Olympic Games and also gain a deeper understanding of the performance of various countries and sports events.

Firstly, we focus on the cumulative number of medals awarded by each country. From this perspective, we find that over the past century, some countries have consistently maintained a world leading medal count. This indicates that these countries have a deep foundation and sustained investment in sports competition.

At the athlete level, we collected data on the number of gold medals generated and the athletes who received the most gold medals in each event, revealing which sports were the most important in the Olympics and which athletes were the strongest gold medal winners.

In addition, we also analyzed the trends in the number of participants and the proportion of women participating in the Olympics to understand how participation in the Olympics has evolved and the performance of gender equality in the Olympics.

Finally, we focused on the performance of major countries, showcasing their performance in terms of gold medals, medals, and gold/medal ratios, while also exploring their dominance in certain specific events.

Through these data and analysis, we can not only understand the history and current situation of the Olympic Games, but also see the development trends of sports competition, as well as the competitive landscape of various countries and events. This has important implications and reference significance for our understanding and appreciation of the Olympics, as well as for our future sports training and competitive strategies.

Introduction

With the continuous improvement of global sports competition level, the Olympic Games, as the highest level sports event, has attracted more and more attention. On this stage, athletes from all over the world demonstrated their talents and spirit of hard work. This article aims to analyze the performance of different genders and events in the Olympic Games in recent years, in order to reveal the reasons behind these changes and the performance trends of athletes in various events.

I will analyze from the following aspects:

1. Gender differences: By comparing the performance of athletes of different genders in the Olympics, we can understand the differences in athletic level between male and female athletes, as well as the causes of these differences.

2. Project distribution: Study the gender distribution of various events in the Olympic Games, analyze the advantages and disadvantages of athletes of different genders in different events, in order to provide reference for future project development and talent cultivation.

3. Athlete performance: By analyzing the gender, age, nationality, and other information of award-winning athletes in previous Olympic Games, we can better understand the success factors of these outstanding athletes and the development trends of the sports they represent.

4. International Competition Pattern: By analyzing the achievements of different countries in the Olympic Games, we can understand the competitiveness of each country in different sports events and the changes in the international sports pattern.

5. Social impact: By analyzing the impact of Olympic achievements on society, we can understand the status and role of sports in social development, as well as how to promote social progress by improving the level of sports competition.

This paper will use data visualization, statistical analysis and other methods to analyze the achievements of different gender and different events in the recent Olympic Games in order to provide beneficial enlightenment for the development of

sports.

Methodology

The Python language has rich libraries and tools in visualization, which can help developers easily create various charts and visualization effects. The following are the Python visualization methods used in the article:

1. Matplotlib: Matplotlib is a popular Python drawing library, which supports a variety of chart types, such as line chart, scatter chart, histogram, pie chart, etc. It provides rich setting options and parameters to meet various visualization needs.

2. Seaborn: Seaborn is an advanced visualization library based on Matplotlib, which provides a more concise API and richer themes. Seaborn is mainly used for processing statistical data and visualizing datasets, which can help developers quickly generate beautiful charts.

3. Plot: Plot is a web-based Python drawing library that supports real-time interaction and cross platform sharing. Plotly supports multiple chart types, such as scatter plots, line plots, bar charts, pie charts, etc. It also provides rich customization options and extension functions.

4. pyecharts: Pyecharts is a Python visualization library based on Echarts, an open source visualization framework developed by Baidu. Pyecharts provides a rich range of chart types and interactive features, while also supporting map visualization and dynamic charts.

5. ggplot2: ggplot2 is a visualization library based on the R language, but the Python community also has corresponding ported versions, such as geoplolib. Ggplot2 provides a grammar based on "aesthetic mapping" that makes it easy to create aesthetically pleasing charts.

6. Bokeh: Bokeh is a Python based interactive visualization library primarily used to create interactive charts in modern, customizable web browsers. Bokeh supports a variety of chart types, such as line chart, scatter chart, histogram, pie chart, etc., and also provides rich interactive functions.

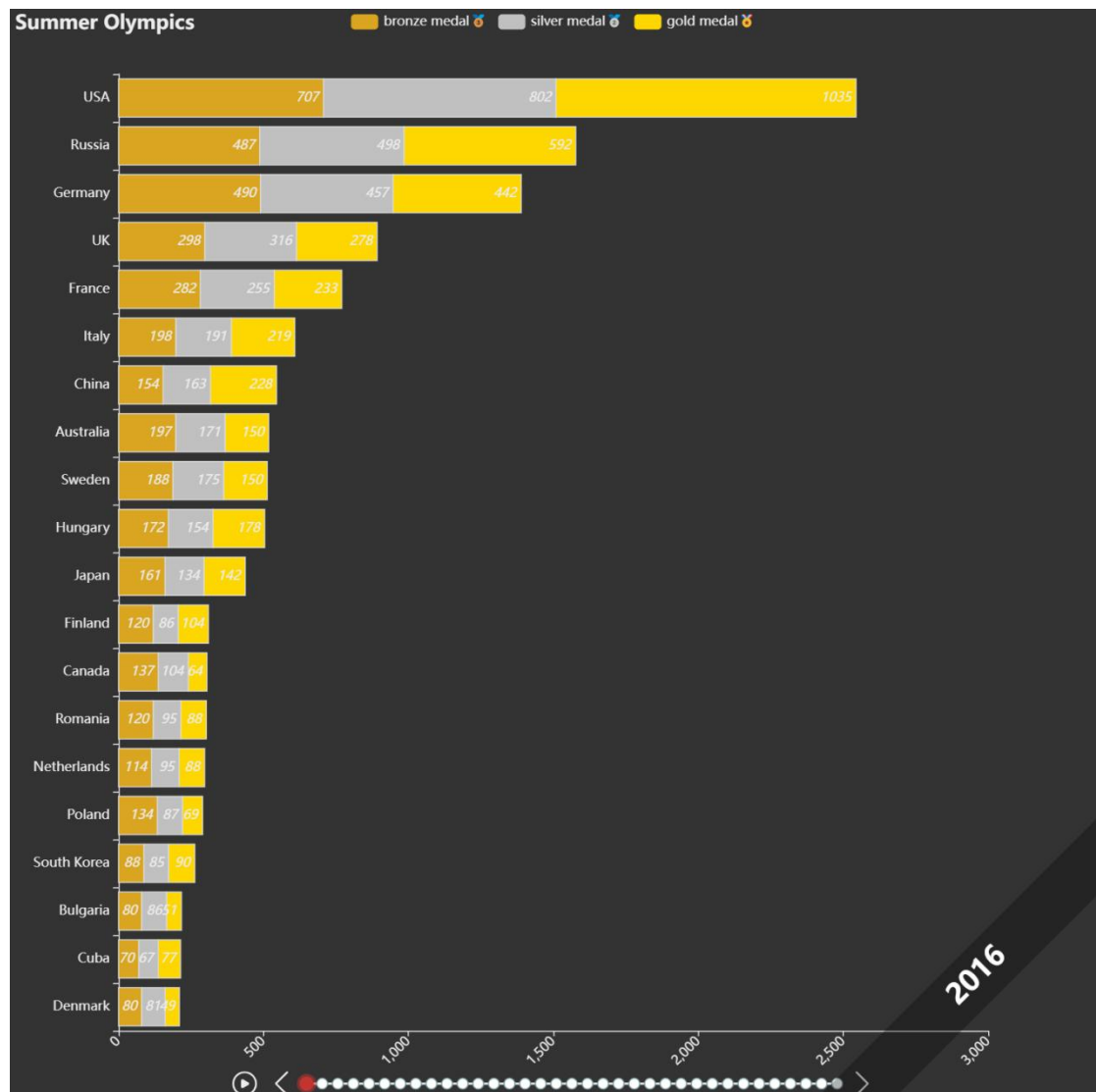
Results and discussion

1. Import Library&Data Å

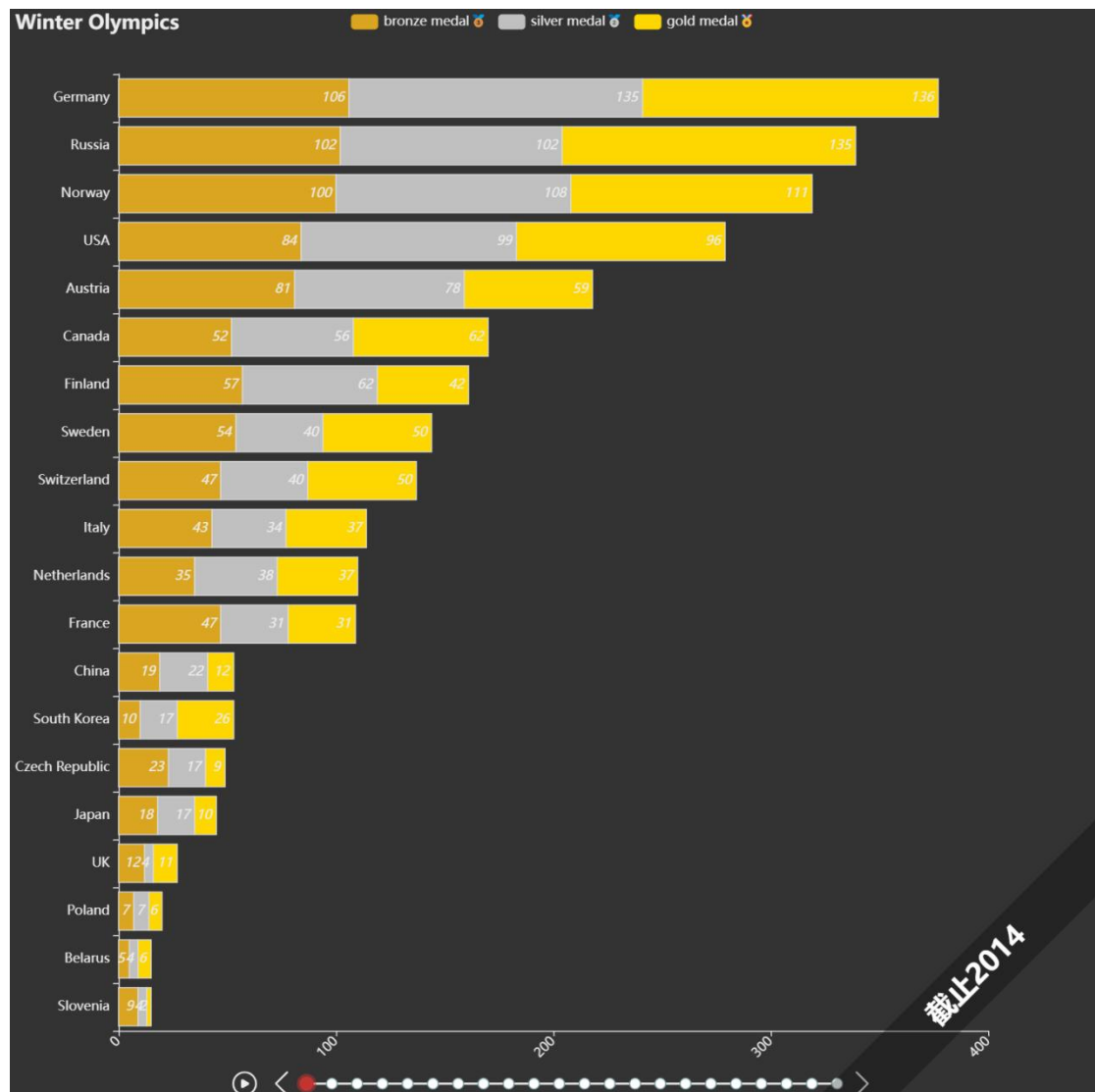
ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal	region	notes
0 1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN	China	NaN
1 2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN	China	NaN
2 3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN	Denmark	NaN
3 4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold	Denmark	NaN
4 5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 500 metres	NaN	Netherlands	NaN

2.Accumulated number of medals from various countries (Summer Olympics)

The timeline below can be freely switched. When switching to a specific time point, the dot representing that time point below will turn red

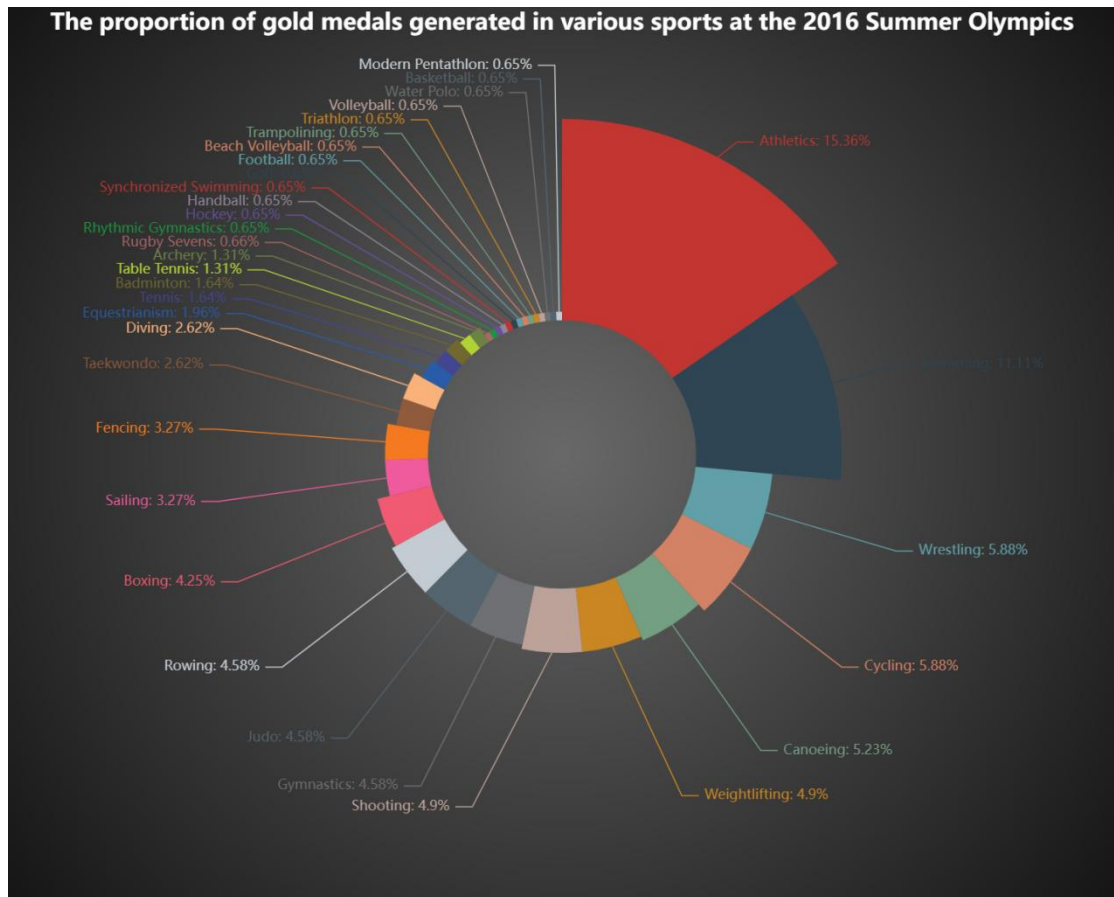


3. Winter Olympics



4. Number of gold medals generated in various sports

Based on the statistics of the 2016 Summer Olympics and the 2014 Winter Olympics;

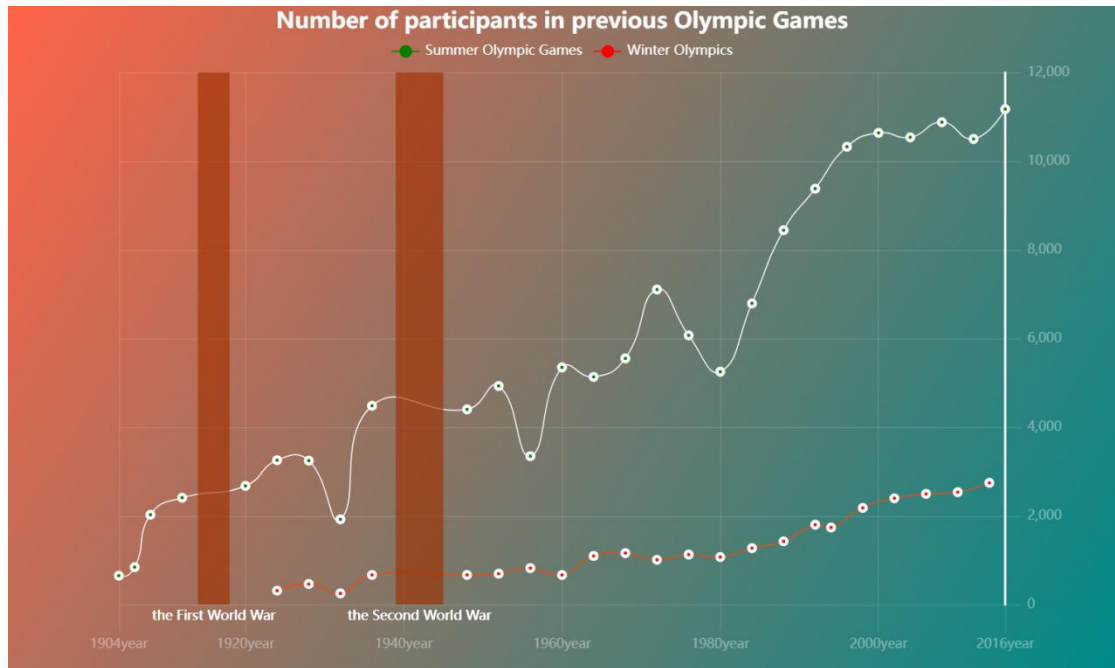


5. Athlete level

Trend of participants over the years

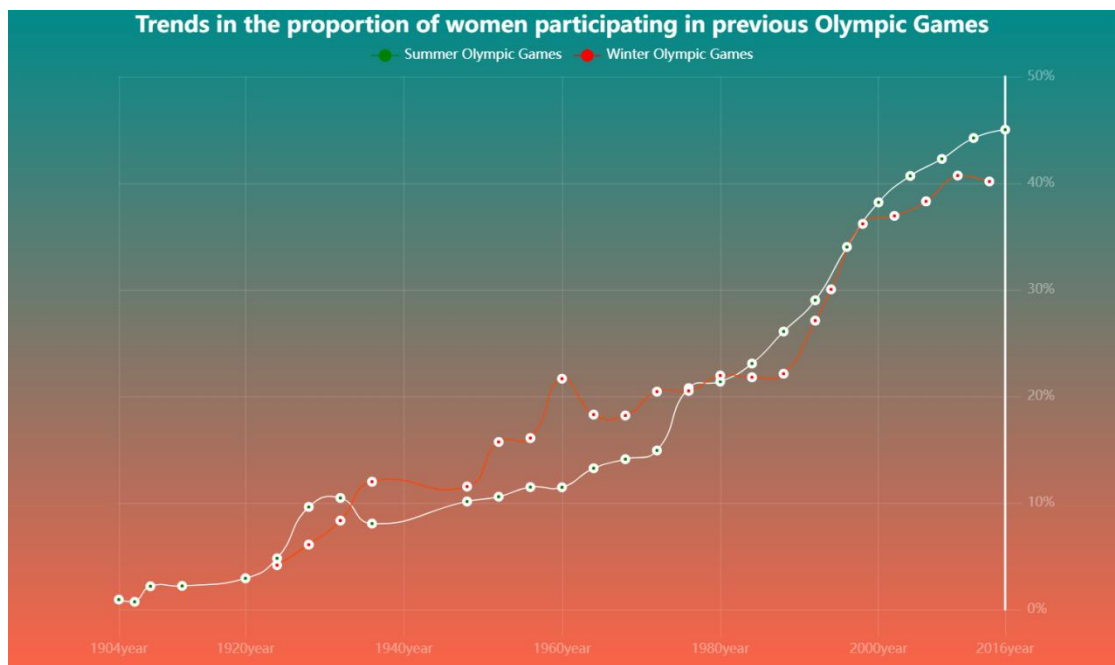
In terms of the number of participants, the number of participants in each Summer Olympics is 4-5 times that of the Winter Olympics;

The overall number of participants has shown an upward trend, but there have also been fluctuations due to historical reasons, such as the boycott of 65 countries during the 1980 Moscow Olympics;



6. Trend in the proportion of female athletes over the years

At the beginning, the Olympics were basically "men's sports", with female athletes only in single digits. In recent Olympics, the number of male and female athletes has tended to be equal;

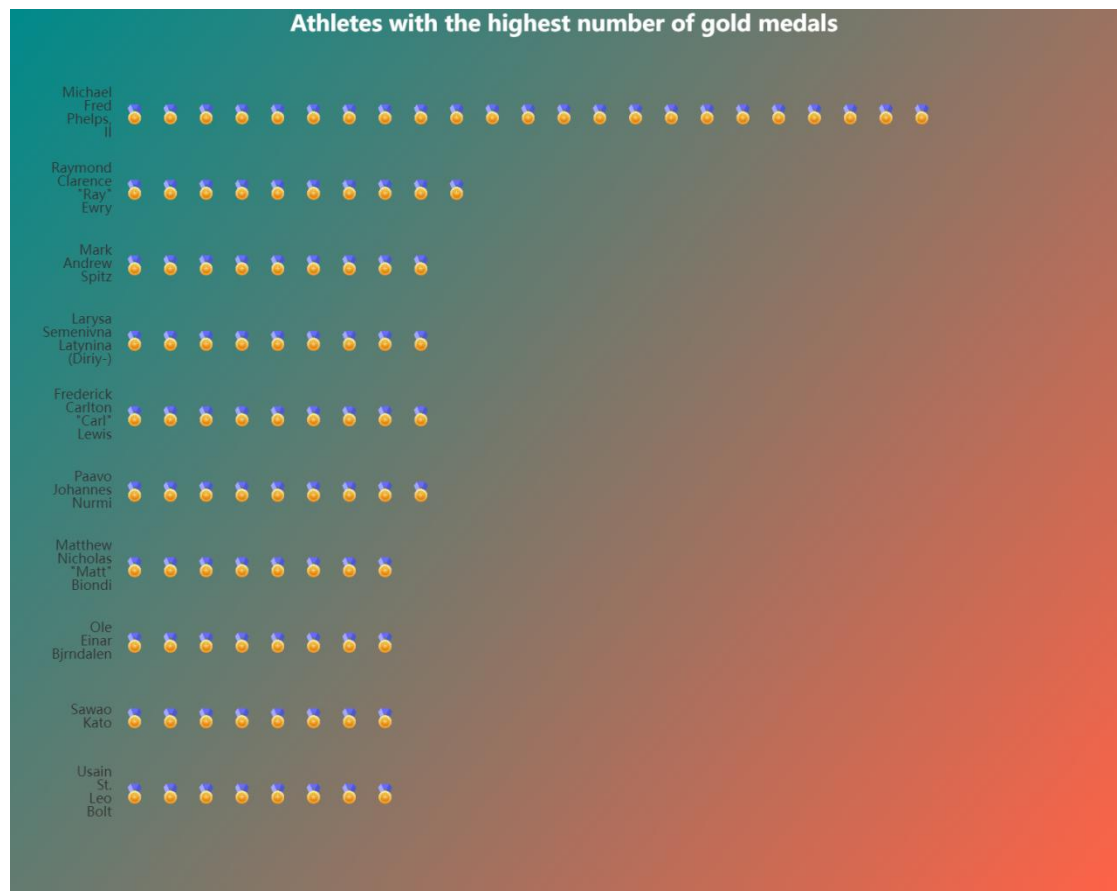


7. Athletes who have won the most gold medals

At the top of the list is American swimming star "Michael Phelps", who has won

23 gold medals by the end of the 2016 Olympic Games;

Bolt has accumulated 8 Olympic gold medals;



8. Ratio of Gold Medals/Medals Received

It seems that Michael Phelps is soft to win a gold medal, but in fact, it is very difficult to win a gold medal

In the history of the entire Olympic Games (including the Summer and Winter Olympics), the number of participants was 134732, with only 10413 athletes winning gold medals, accounting for 7.7%;

There are 28202 athletes who have won medals (including gold, silver, and copper), accounting for 20.93%;

Percentage of gold medals won



Percentage of medals won



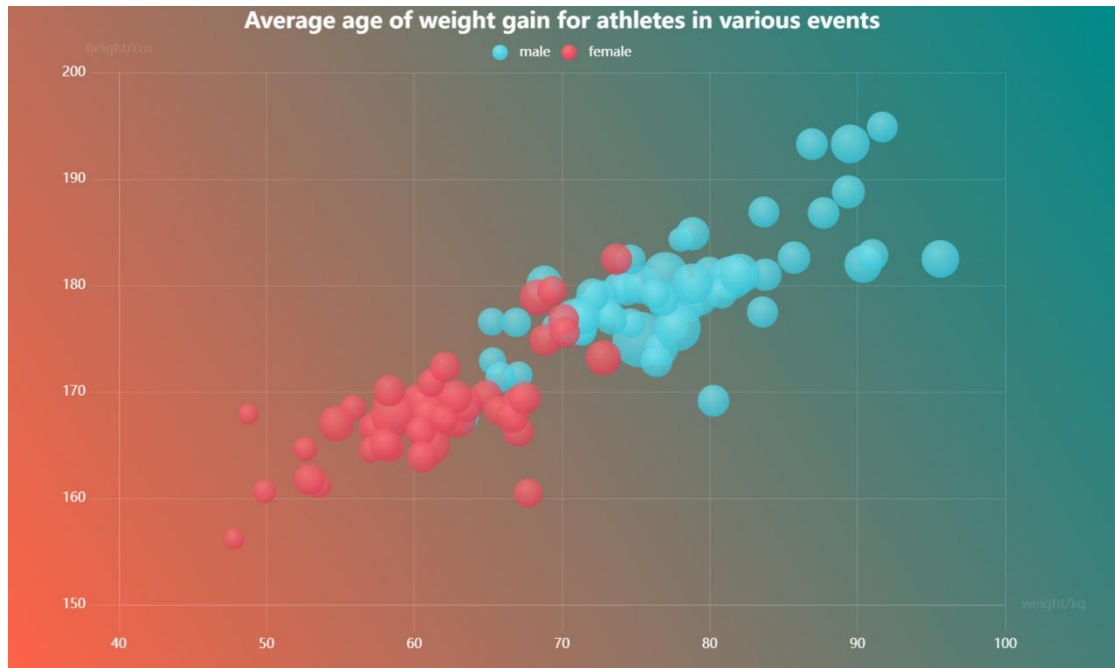
9. Average physical fitness data of athletes

Statistics based on different sports events

The highest average height event for athletes is basketball, with women reaching an average height of 182cm and men reaching an average height of 194cm;

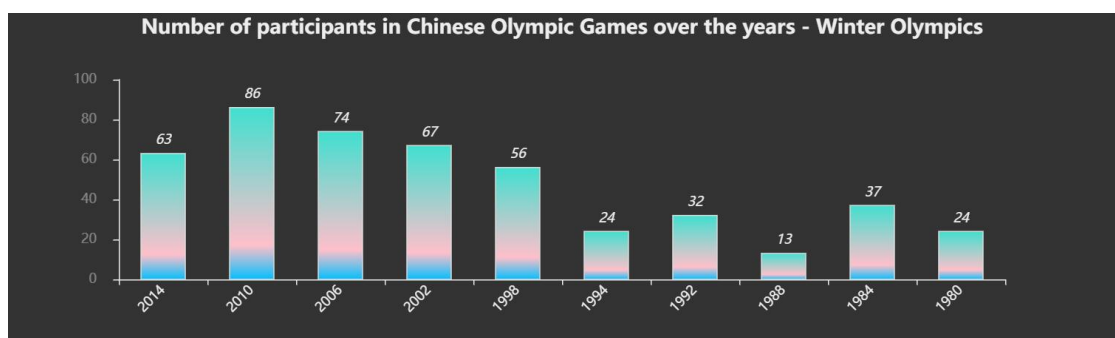
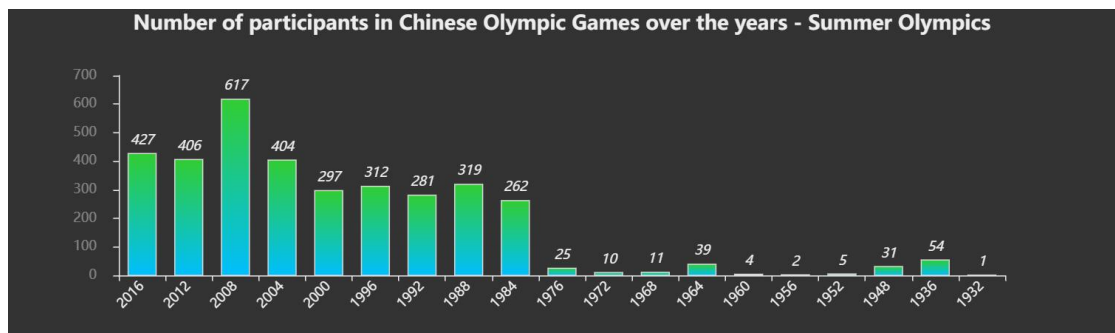
In men's events, the event with the highest average weight for athletes is tug of war, with an average weight of 96kg (tug of war has been cancelled since the 7th Olympic Games);

The event with the highest average age of athletes is Art competition (a strange event like Baidu), with an average age of 46 years. In addition, it is equestrian and shooting, with an average age of 34.4 years and 34.2 years for men and 34.22 years and 29.12 years for women, respectively;

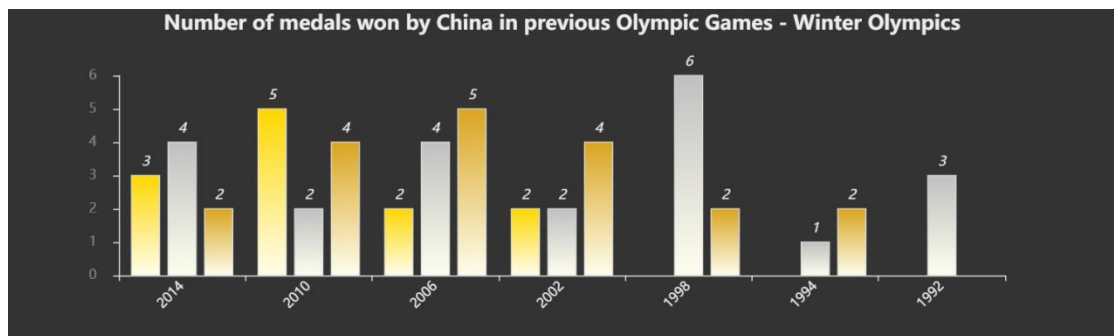
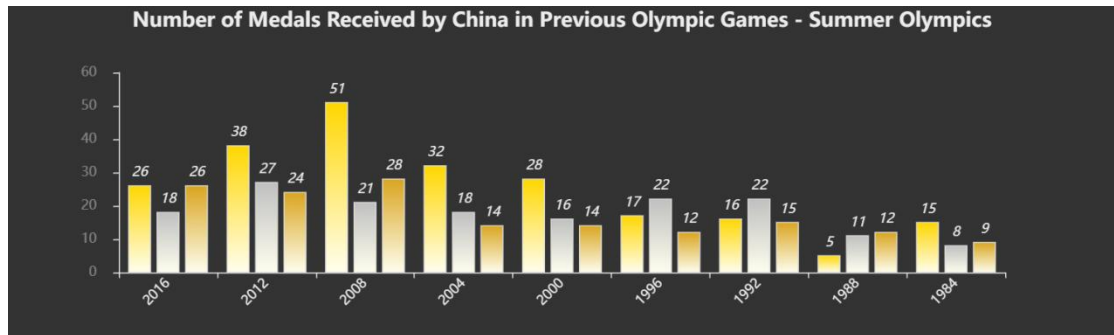


10.China's Olympic Performance

Number of participants in previous Olympic Games

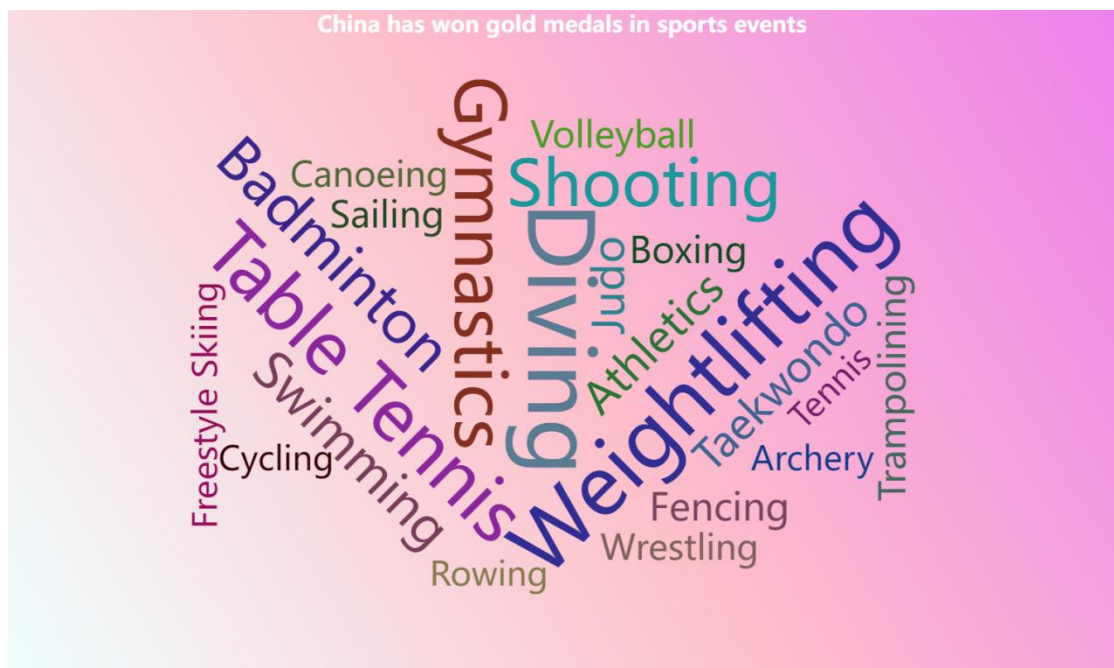


Number of medals from previous Olympic Games



Advantage projects

Diving, gymnastics, shooting, weightlifting, table tennis, badminton



11. Olympic events dominated by a single country

Many sports have long been ruled by a certain country, such as the well-known China Table Tennis, USA Basketball;

This screening selected events that have accumulated over 10 gold medals in the

past 5 Olympic Games (after the 2000 Sydney Olympics) and have a single country's "gold medal winning rate" exceeding 50%;

Russia Having won all 20 gold medals in synchronized swimming and artistic gymnastics since 2000;

China In table tennis, he won 9 out of 10 gold medals since 2000, and lost the gold medal in the men's singles event at the 2004 Athens Olympics;

United States In basketball, he also won 9 out of the past 10 gold medals, and lost one gold medal in 2004, losing to Argentina in the men's basketball semi-finals and ultimately winning a bronze medal;

In diving events, China Having won 31 out of the past 40 gold medals, the Dream Team deserves its reputation;

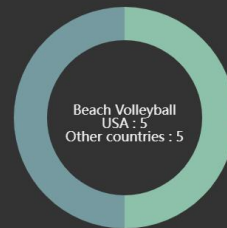
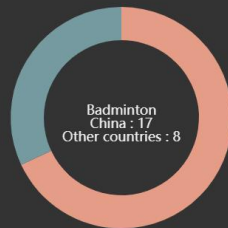
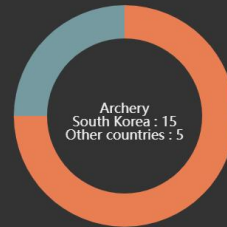
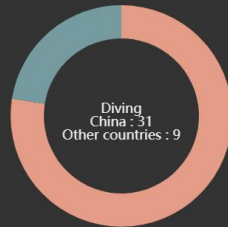
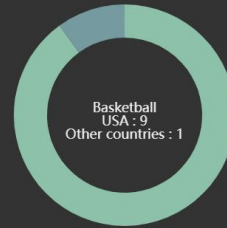
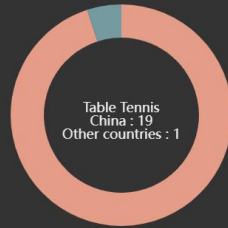
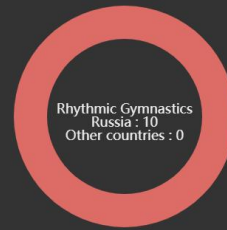
In archery, South Korea Obtained 15 out of the past 20 gold medals;

In badminton, China Won 17 out of the past 25 gold medals;

In the beach volleyball event, the United States Won 5 out of the past 10 gold medals;

Projects governed by a single country

Statistical cycle: Since the 2000 Sydney Olympics



References

- [1] Teng Yue, Yu An, Tang Yi Miao, Jiang Zi Yang, Guan Wen Jie, Li Zhi Shuai, Yu Hong Yan, Zou Lu Yi. Visualization and quantification of cadmium accumulation, chelation and antioxidation during the process of vacuolar compartmentalization in the hyperaccumulator plant *Solanum nigrum* L.[J]. *Plant Science*, 2021, 310 (prepublish).
- [2] Zhu Yong, Tao Shanlong, Chen Chen, Liu Jiahua, Chen Mingxia, Shangguan Wenfeng. The experimental and simulation investigation of the dynamic characteristic of submicron-scale aerosol in high-voltage electric field by a visualization method[J]. *Journal of Hazardous Materials*, 2021, 416.
- [3] Jin Xilang, Ma Xuehao, Zhou Hongwei, Chen Jiawei, Li Minzhi, Yang Jin, Bai Haiyan, She Mengyao. Construction of DCM-based NIR fluorescent probe for visualization detection of H₂S in solution and nanofibrous film[J]. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, 2021, 257.
- [4] Morrison Sam, Henderson Annette M.E., Sagar Mark, Kennedy-Costantini Siobhan, Adams Josie. Peek-a-who? Exploring the dynamics of early communication with an interactive partner swap paradigm and state space grid visualization[J]. *Infant Behavior and Development*, 2021, 64.
- [5] Zeng Lingyu, Su Yun, Stejskal Vaclav, Opit George, Aulicky Radek, Li Zhihong. Primers and visualization of LAMP: A rapid molecular identification method for *Liposcelis entomophila* (Enderlein) (Psocodea: Liposcelididae)[J]. *Journal of Stored Products Research*, 2021, 93.
- [6] Zeng Lingyu, Su Yun, Stejskal Vaclav, Opit George, Aulicky Radek, Li Zhihong. Primers and visualization of LAMP: A rapid molecular identification method for *Liposcelis entomophila* (Enderlein) (Psocodea: Liposcelididae)[J]. *Journal of Stored Products Research*, 2021, 93.

Appendix

1.Import Library&Data Å+

```
import pandas as pd
import numpy as np
import pyechartsfrom pyecharts.charts
import *from pyecharts
import options as optsfrom pyecharts.commons.utils
import JsCode
athlete_data = pd.read_csv('/home/kesci/input/olympic/athlete_events.csv')
noc_region = pd.read_csv('/home/kesci/input/olympic/noc_regions.csv')
data = pd.merge(athlete_data, noc_region, on='NOC', how='left')data.head()
```

2.Accumulated number of medals from various countries (summer Olympics)

```
year_list = sorted(list(set(medal_data['Year'].to_list())), reverse=True)

tl = Timeline(init_opts=opts.InitOpts(theme='dark', width='1000px', height='1000px'))
tl.add_schema(is_timeline_show=True,is_rewind_play=True, is_inverse=False,
              label_opts=opts.LabelOpts(is_show=False))

for year in year_list:
    t_data = medal_stat(year)[::-1]
    bar = (
        Bar(init_opts=opts.InitOpts())
        .add_xaxis([x[0] for x in t_data])
        .add_yaxis("bronze medal, [x[3] for x in t_data],
                    stack='stack1',
                    itemstyle_opts=opts.ItemStyleOpts(border_color='rgb(220,220,220)',color='r
gb(218,165,32)'))
        .add_yaxis("silver medal, [x[2] for x in t_data],
                    stack='stack1',
                    itemstyle_opts=opts.ItemStyleOpts(border_color='rgb(220,220,220)',color='r
gb(192,192,192)'))
        .add_yaxis("gold medal, [x[1] for x in t_data],
                    stack='stack1',
                    itemstyle_opts=opts.ItemStyleOpts(border_color='rgb(220,220,220)',color='r
gb(255,215,0)'))
        .set_series_opts(label_opts=opts.LabelOpts(is_show=True,
                                                    position='insideRight',
                                                    font_style='italic'),)
        .set_global_opts(
            title_opts=opts.TitleOpts(title="Summer Olympics"),
            xaxis_opts=opts.AxisOpts(axislabel_opts=opts.LabelOpts(rotate=45)),
            legend_opts=opts.LegendOpts(is_show=True),
```



```

        graphic_opts=[opts.GraphicGroup(graphic_item=opts.GraphicItem(
            rotation=JsCode("Math.PI / 4"),
            bounding="raw",
            right=110,
            bottom=110,
            z=100),
        children=[
            opts.GraphicRect(
                graphic_item=opts.GraphicItem(
                    left="center", top="center", z=100
                ),
                graphic_shape_opts=opts.GraphicShapeOpts(
                    width=400, height=50
                ),
                graphic_basicstyle_opts=opts.GraphicBasicStyleOpt

s(
                    fill="rgba(0,0,0,0.3)"
                ),
            ),
            opts.GraphicText(
                graphic_item=opts.GraphicItem(
                    left="center", top="center", z=100
                ),
                graphic_textstyle_opts=opts.GraphicTextStyleOpts(
                    text=year,
                    font="bold 26px Microsoft YaHei",
                    graphic_basicstyle_opts=opts.GraphicBasicStyle

Opts(
                    fill="#fff"
                ),
            ),
        ],
    ],
    ],)

    .reversal_axis()
    tl.add(bar, year)

    tl.render_notebook()

```

3..Accumulated number of medals from various countries (winter Olympics)

```

year_list = sorted(list(set(medal_data["Year"][medal_data.Season=="Winter"].to_list()), reverse=True
e)

tl = Timeline(init_opts=opts.InitOpts(theme='dark', width='1000px', height='1000px'))
tl.add_schema(is_timeline_show=True, is_rewind_play=True, is_inverse=False,
              label_opts=opts.LabelOpts(is_show=False))

for year in year_list:
    t_data = medal_stat(year, 'Winter')[::-1]
    bar = (
        Bar(init_opts=opts.InitOpts(theme='dark'))
        .add_xaxis([x[0] for x in t_data])
        .add_yaxis("bronze medal, [x[3] for x in t_data],
                    stack='stack1',
                    itemstyle_opts=opts.ItemStyleOpts(border_color='rgb(220,220,220)',color='r
gb(218,165,32)'))
        .add_yaxis("silver medal, [x[2] for x in t_data],
                    stack='stack1',
                    itemstyle_opts=opts.ItemStyleOpts(border_color='rgb(220,220,220)',color='r
gb(192,192,192)'))
        .add_yaxis("gold medal, [x[1] for x in t_data],
                    stack='stack1',
                    itemstyle_opts=opts.ItemStyleOpts(border_color='rgb(220,220,220)',color='r
gb(255,215,0)'))
        .set_series_opts(label_opts=opts.LabelOpts(is_show=True,
                                                    position='insideRight',
                                                    font_style='italic'),)
        .set_global_opts(
            title_opts=opts.TitleOpts(title="Winter Olympics"),
            xaxis_opts=opts.AxisOpts(axislabel_opts=opts.LabelOpts(rotate=45)),
            legend_opts=opts.LegendOpts(is_show=True),
            graphic_opts=[opts.GraphicGroup(graphic_item=opts.GraphicItem(
                rotation=JsCode("Math.PI / 4"),
                bounding="raw",
                right=110,
                bottom=110,
                z=100),
                children=[
                    opts.GraphicRect(
                        graphic_item=opts.GraphicItem(
                            left="center", top="center", z=100
                        ),
                        graphic_shape_opts=opts.GraphicShapeOpts(
                            width=400, height=50

```

```

        ),
        graphic_basicstyle_opts=opts.GraphicBasicStyleOpt
s(
        fill="rgba(0,0,0,0.3)"
    ),
),
opts.GraphicText(
    graphic_item=opts.GraphicItem(
        left="center", top="center", z=100
    ),
    graphic_textstyle_opts=opts.GraphicTextStyleOpts(
        text='截止{}'.format(year),
        font="bold 26px Microsoft YaHei",
        graphic_basicstyle_opts=opts.GraphicBasicStyle
Opts(
        fill="#fff"
    ),
),
),
],
)
],)

.reversal_axis())
tl.add(bar, year)

tl.render_notebook()

```

4.Number of gold medals generated in various sports

```

background_color_js = """new echarts.graphic.RadialGradient(0.5, 0.5, 1, [{
    offset: 0,
    color: '#696969'
}, {
    offset: 1,
    color: '#000000'
}])"""

tab = Tab()
temp = data[(data['Medal']=='Gold') & (data['Year']==2016) & (data['Season']=='Summer')]

event_medal = temp.groupby(['Sport'])['Event'].nunique().reset_index()
event_medal.columns = ['Sport', 'Nums']
event_medal = event_medal.sort_values(by="Nums" , ascending=False)

pie = (Pie(init_opts=opts.InitOpts(bg_color=JsCode(background_color_js), width='1000px', height='
800px'))

```

```

        .add('gold medal, [(row['Sport'], row['Nums']) for _, row in event_medal.iterrows()],
            radius=["30%", "75%"],
            rosetype="radius")

        .set_global_opts(title_opts=opts.TitleOpts(title="The proportion of gold medals generated in
various sports at the 2016 Summer Olympics",
                                                    pos_left="center",
                                                    title_textstyle_opts=opts.TextStyleOpts(color="white", font_size=20),
                                                    ),
                        legend_opts=opts.LegendOpts(is_show=False))

        .set_series_opts(label_opts=opts.LabelOpts(formatter="{b}: {d}%"),
                        tooltip_opts=opts.TooltipOpts(trigger="item", formatter="{a} <br/>{b}: {c}
({d}%"))
    )
    tab.add(pie, '2016 Summer Olympics')

temp = data[(data['Medal']=='Gold') & (data['Year']==2014) & (data['Season']=='Winter')]

event_medal = temp.groupby(['Sport'])['Event'].nunique().reset_index()
event_medal.columns = ['Sport', 'Nums']
event_medal = event_medal.sort_values(by="Nums" , ascending=False)

pie = (Pie(init_opts=opts.InitOpts(bg_color=JsCode(background_color_js), width='1000px', height='
800px'))

        .add('gold medal, [(row['Sport'], row['Nums']) for _, row in event_medal.iterrows()],
            radius=["30%", "75%"],
            rosetype="radius")

        .set_global_opts(title_opts=opts.TitleOpts(title="The proportion of gold medals generated in
various sports at the 2014 Winter Olympics",
                                                    pos_left="center",
                                                    title_textstyle_opts=opts.TextStyleOpts(color="white", font_size=20),
                                                    ),
                        legend_opts=opts.LegendOpts(is_show=False))

        .set_series_opts(label_opts=opts.LabelOpts(formatter="{b}: {d}%"),
                        tooltip_opts=opts.TooltipOpts(trigger="item", formatter="{a} <br/>{b}: {c}
({d}%"))
    ),)
    )
    tab.add(pie, '2014 Winter Olympics')
    tab.render_notebook()

```

5. Trend of participants over the years

```

athlete = data.groupby(['Year', 'Season'])['Name'].nunique().reset_index()
athlete.columns = ['Year', 'Season', 'Nums']
athlete = athlete.sort_values(by="Year" , ascending=True)

```

```

x_list, y1_list, y2_list = [], [], []

for _, row in athlete.iterrows():
    x_list.append(str(row['Year']))
    if row['Season'] == 'Summer':
        y1_list.append(row['Nums'])
        y2_list.append(None)
    else:
        y2_list.append(row['Nums'])
        y1_list.append(None)

background_color_js = (
    "new echarts.graphic.LinearGradient(1, 1, 0, 0, "
    "[{offset: 0, color: '#008B8B'}, {offset: 1, color: '#FF6347'}], false)"
)

line = (
    Line(init_opts=opts.InitOpts(bg_color=JsCode(background_color_js), width='1000px', height='600px'))
    .add_xaxis(x_list)
    .add_yaxis("Summer Olympic Games",
        y1_list,
        is_smooth=True,
        is_connect_nones=True,
        symbol="circle",
        symbol_size=6,
        linestyle_opts=opts.LineStyleOpts(color="#fff"),
        label_opts=opts.LabelOpts(is_show=False, position="top", color="white"),
        itemstyle_opts=opts.ItemStyleOpts(
            color="green", border_color="#fff", border_width=3),
        tooltip_opts=opts.TooltipOpts(is_show=True))
    .add_yaxis("Winter Olympics",
        y2_list,
        is_smooth=True,
        is_connect_nones=True,
        symbol="circle",
        symbol_size=6,
        linestyle_opts=opts.LineStyleOpts(color="#FF4500"),
        label_opts=opts.LabelOpts(is_show=False, position="top", color="white"),
        itemstyle_opts=opts.ItemStyleOpts(
            color="red", border_color="#fff", border_width=3),
        tooltip_opts=opts.TooltipOpts(is_show=True))
    .set_series_opts(

```

```

markarea_opts=opts.MarkAreaOpts(
    label_opts=opts.LabelOpts(is_show=True, position="bottom", color="white"),
    data=[
        opts.MarkAreaItem(name="the First World War", x=(1914, 1918)),
        opts.MarkAreaItem(name="the Second World War", x=(1939, 1945)),
    ]
)

.set_global_opts(title_opts=opts.TitleOpts(title="Number of participants in previous Olympic Games",
    pos_left="center",
    title_textstyle_opts=opts.TextStyleOpts(color="white", font_size=20)),
    legend_opts=opts.LegendOpts(is_show=True, pos_top='5%',
    textstyle_opts=opts.TextStyleOpts(color="white", font_size=12)),
    xaxis_opts=opts.AxisOpts(type_="value",
    min_=1904,
    max_=2016,
    boundary_gap=False,
    axislabel_opts=opts.LabelOpts(margin=30, color="#ffffff63",
    formatter=JsCode("""function
    {return value+'year';}""")),
    axisline_opts=opts.AxisLineOpts(is_show=False),
    axistick_opts=opts.AxisTickOpts(
    is_show=True,
    length=25,
    linestyle_opts=opts.LineStyleOpts(color="#ffffff1f"),
    ),
    splitline_opts=opts.SplitLineOpts(
    is_show=True, linestyle_opts=opts.LineStyleOpts(color="#ffffff1f")
    ),
    ),
    yaxis_opts=opts.AxisOpts(
    type_="value",
    position="right",
    axislabel_opts=opts.LabelOpts(margin=20, color="#ffffff63",
    axisline_opts=opts.AxisLineOpts(
    linestyle_opts=opts.LineStyleOpts(width=2, color="#fff")
    ),

```

```

        axistick_opts=opts.AxisTickOpts(
            is_show=True,
            length=15,
            linestyle_opts=opts.LineStyleOpts(color="#ffffff1f"),
        ),
        splitline_opts=opts.SplitLineOpts(
            is_show=True, linestyle_opts=opts.LineStyleOpts(color="
#ffffff1f")
        ),
    ),
),)

)

line.render_notebook()

```

6. Trend in the proportion of female athletes over the years

```

m_data = data[data.Sex=='M'].groupby(['Year', 'Season'])['Name'].nunique().reset_index()
m_data.columns = ['Year', 'Season', 'M-Nums']
m_data = m_data.sort_values(by="Year" , ascending=True)

f_data = data[data.Sex=='F'].groupby(['Year', 'Season'])['Name'].nunique().reset_index()
f_data.columns = ['Year', 'Season', 'F-Nums']
f_data = f_data.sort_values(by="Year" , ascending=True)

t_data = pd.merge(m_data, f_data, on=['Year', 'Season'])
t_data['F-rate'] = round(t_data['F-Nums'] / (t_data['F-Nums'] + t_data['M-Nums'] ), 4)

x_list, y1_list, y2_list = [], [], []

for _, row in t_data.iterrows():
    x_list.append(str(row['Year']))
    if row['Season'] == 'Summer':
        y1_list.append(row['F-rate'])
        y2_list.append(None)
    else:
        y2_list.append(row['F-rate'])
        y1_list.append(None)

background_color_js = (
    "new echarts.graphic.LinearGradient(0, 0, 0, 1, "
    "[{offset: 0, color: '#008B8B'}, {offset: 1, color: '#FF6347'}], false)"
)

line = (

```

```

Line(init_opts=opts.InitOpts(bg_color=JsCode(background_color_js), width='1000px', height='600px'))
.add_xaxis(x_list)
.add_yaxis("Summer Olympic Games",
    y1_list,
    is_smooth=True,
    is_connect_nones=True,
    symbol="circle",
    symbol_size=6,
    linestyle_opts=opts.LineStyleOpts(color="#fff"),
    label_opts=opts.LabelOpts(is_show=False, position="top", color="white"),
    itemstyle_opts=opts.ItemStyleOpts(color="green", border_color="#fff", border_width=3),
    tooltip_opts=opts.TooltipOpts(is_show=True),)
.add_yaxis("Winter Olympic Games",
    y2_list,
    is_smooth=True,
    is_connect_nones=True,
    symbol="circle",
    symbol_size=6,
    linestyle_opts=opts.LineStyleOpts(color="#FF4500"),
    label_opts=opts.LabelOpts(is_show=False, position="top", color="white"),
    itemstyle_opts=opts.ItemStyleOpts(color="red", border_color="#fff", border_width=3),
    tooltip_opts=opts.TooltipOpts(is_show=True),)
.set_series_opts(tooltip_opts=opts.TooltipOpts(trigger="item", formatter=JsCode("""function (params)
    {return params.data[0]+ 'year:
' + Number(params.data[1])*100 +'%;'}""")),)
.set_global_opts(title_opts=opts.TitleOpts(title="Trends in the proportion of women participating in previous Olympic Games",
    pos_left="center",
    title_textstyle_opts=opts.TextStyleOpts(color="white", font_size=20)),)
    legend_opts=opts.LegendOpts(is_show=True, pos_top='5%',
    textstyle_opts=opts.TextStyleOpts(color="white", font_size=12)),
    xaxis_opts=opts.AxisOpts(type_="value",
    min_=1904,
    max_=2016,
    boundary_gap=False,
    axislabel_opts=opts.LabelOpts(margin=30, color="#ffffff",
    formatter=JsCode("""function
    {return value+'year';}""")),)

```



```

axisline_opts=opts.AxisLineOpts(is_show=False),
axistick_opts=opts.AxisTickOpts(
    is_show=True,
    length=25,
    linestyle_opts=opts.LineStyleOpts(color="#ffffff1f"),
),
splitline_opts=opts.SplitLineOpts(
    is_show=True, linestyle_opts=opts.LineStyleOpts(color="#ffffff1f")
),
),
yaxis_opts=opts.AxisOpts(
    type_="value",
    position="right",
    axislabel_opts=opts.LabelOpts(margin=20, color="#ffffff63",
    formatter=JsCode("""function (value)
    {return Number(value *100)+'%';}""")),
axisline_opts=opts.AxisLineOpts(
    linestyle_opts=opts.LineStyleOpts(width=2, color="#fff")
),
axistick_opts=opts.AxisTickOpts(
    is_show=True,
    length=15,
    linestyle_opts=opts.LineStyleOpts(color="#ffffff1f"),
),
splitline_opts=opts.SplitLineOpts(
    is_show=True, linestyle_opts=opts.LineStyleOpts(color="#ffffff1f")
),
),
),
line.render_notebook()

```

7.Athletes who have won the most gold medals

```

temp = data[(data['Medal']=='Gold')]

athlete = temp.groupby(['Name'])['Medal'].count().reset_index()
athlete.columns = ['Name', 'Nums']
athlete = athlete.sort_values(by="Nums" , ascending=True)

background_color_js = (
    "new echarts.graphic.LinearGradient(0, 0, 1, 1, "

```

```

        "[{offset: 0, color: '#008B8B'}, {offset: 1, color: '#FF6347'}], false)"
    )

    pb = (
        PictorialBar(init_opts=opts.InitOpts(bg_color=JsCode(background_color_js), width='1000px', height='800px'))
        .add_xaxis([x.replace(' ', '\n') for x in athlete['Name'].tail(10).tolist()])
        .add_yaxis(
            "",
            athlete['Nums'].tail(10).tolist(),
            label_opts=opts.LabelOpts(is_show=False),
            symbol_size=25,
            symbol_repeat='fixed',
            symbol_offset=[0, 0],
            is_symbol_clip=True,
            symbol='image://https://cdn.kesci.com/upload/image/q8f8otrffc.png')
        .reversal_axis()
        .set_global_opts(
            title_opts=opts.TitleOpts(title="Athletes with the highest number of gold medals", pos_left='center',
                                       title_textstyle_opts=opts.TextStyleOpts(color="white", font_size=20)),
            xaxis_opts=opts.AxisOpts(is_show=False,),
            yaxis_opts=opts.AxisOpts(
                axistick_opts=opts.AxisTickOpts(is_show=False),
                axisline_opts=opts.AxisLineOpts(
                    linestyle_opts=opts.LineStyleOpts(opacity=0)
                ),
            ),
        ))

    pb.render_notebook()

```

8. Ratio of Gold Medals/Medals Received

```

total_athlete = len(set(data['Name']))
medal_athlete = len(set(data['Name'][data['Medal'].isin(['Gold', 'Silver', 'Bronze'])]))
gold_athlete = len(set(data['Name'][data['Medal']=='Gold']))

l1 = Liquid(init_opts=opts.InitOpts(theme='dark', width='1000px', height='800px'))
l1.add("Winning a medal", [medal_athlete/total_athlete],
       center=["70%", "50%"],
       label_opts=opts.LabelOpts(font_size=50,

```

```

        formatter=JsCode(
            """function (param) {
                return (Math.floor(param.value * 10000) / 100) + '%';
            }""",
            position="inside",
        ))
l1.set_global_opts(title_opts=opts.TitleOpts(title="Percentage of medals won", pos_left='62%', pos_top='8%'))
l1.set_series_opts(tooltip_opts=opts.TooltipOpts(is_show=False))

l2 = Liquid(init_opts=opts.InitOpts(theme='dark', width='1000px', height='800px'))
l2.add("Winning a gold medal",
        [gold_athlete/total_athlete],
        center=["25%", "50%"],
        label_opts=opts.LabelOpts(font_size=50,
            formatter=JsCode(
                """function (param) {
                    return (Math.floor(param.value * 10000) / 100) + '%';
                }""",
                position="inside",
            )),)
l2.set_global_opts(title_opts=opts.TitleOpts(title="Percentage of gold medals won", pos_left='17%', pos_top='8%'))
l2.set_series_opts(tooltip_opts=opts.TooltipOpts(is_show=False))

grid = Grid().add(l1, grid_opts=opts.GridOpts()).add(l2, grid_opts=opts.GridOpts())
grid.render_notebook()

```

9. Average physical fitness data of athletes

```

tool_js = """function (param) {return param.data[2] + '<br/>'
    + '平均体重: ' + Number(param.data[0]).toFixed(2) + ' kg<br/>'
    + '平均身高: ' + Number(param.data[1]).toFixed(2) + ' cm<br/>'
    + '平均年龄: ' + Number(param.data[3]).toFixed(2);}"""

background_color_js = (
    "new echarts.graphic.LinearGradient(1, 0, 0, 1, "
    "[{offset: 0, color: '#008B8B'}, {offset: 1, color: '#FF6347'}], false)"
)

temp_data = data[data['Sex']=='M'].groupby(['Sport'])['Age', 'Height', 'Weight'].mean().reset_index().dropna(how='any')

scatter = (Scatter(init_opts=opts.InitOpts(bg_color=JsCode(background_color_js), width='1000px', height='600px'))
    .add_xaxis(temp_data['Weight'].tolist())

```

```

.add_yaxis("male", [[row['Height'], row['Sport'], row['Age']] for _, row in temp_data.iterrows()]),

        # 渐变效果实现部分
        color=JsCode("""new echarts.graphic.RadialGradient(0.4, 0.3, 1, [{
            offset: 0,
            color: 'rgb(129, 227, 238)'
        }, {
            offset: 1,
            color: 'rgb(25, 183, 207)'
        }])"""))
.set_series_opts(label_opts=opts.LabelOpts(is_show=False))
.set_global_opts(
    title_opts=opts.TitleOpts(title="Average age of weight gain for athletes in various events", pos_left="center",
                                title_textstyle_opts=opts.TextStyleOpts(color="white", font_size=20)),
    legend_opts=opts.LegendOpts(is_show=True, pos_top='5%',
                                   textstyle_opts=opts.TextStyleOpts(color="white", font_size=12)),

    tooltip_opts = opts.TooltipOpts(formatter=JsCode(tool_js)),
    xaxis_opts=opts.AxisOpts(
        name='weight/kg',
        # 设置坐标轴为数值类型
        type_="value",
        is_scale=True,
        # 显示分割线
        axislabel_opts=opts.LabelOpts(margin=30, color="white"),
        axisline_opts=opts.AxisLineOpts(is_show=True, linestyle_opts=opts.LineStyleOpts(color="#ffffff1f")),
        axistick_opts=opts.AxisTickOpts(is_show=True, length=25,
                                         linestyle_opts=opts.LineStyleOpts(color="#ffffff1f")),
        splitline_opts=opts.SplitLineOpts(is_show=True, linestyle_opts=opts.LineStyleOpts(color="#ffffff1f")),
    ),

    yaxis_opts=opts.AxisOpts(
        name='height/cm',
        # 设置坐标轴为数值类型
        type_="value",
        # 默认为 False 表示起始为 0
        is_scale=True,
        axislabel_opts=opts.LabelOpts(margin=30, color="white"),
        axisline_opts=opts.AxisLineOpts(is_show=True, linestyle_opts=opts.LineStyleOpts(color="#ffffff1f")),
        axistick_opts=opts.AxisTickOpts(is_show=True, length=25,

```

```

                                linestyle_opts=opts.LineStyleOpts(color="#ffffff1f")),
                                splitline_opts=opts.SplitLineOpts(is_show=True, linestyle_opts=opts.LineStyleOpts(c
olor="#ffffff1f")
                                )),
                                visualmap_opts=opts.VisualMapOpts(is_show=False, type_='size', range_size=[5,50], m
in_=10, max_=40)
                                ))

temp_data = data[data['Sex']=='F'].groupby(['Sport'])['Age', 'Height', 'Weight'].mean().reset_index().
dropna(how='any')

scatter1 = (Scatter()
            .add_xaxis(temp_data['Weight'].tolist())
            .add_yaxis("female", [[row['Height'], row['Sport'], row['Age']] for _, row in temp_data.it
errows()],
            itemstyle_opts=opts.ItemStyleOpts(
                color=JsCode("""new echarts.graphic.RadialGradient(0.4, 0.3, 1, [{
                    offset: 0,
                    color: 'rgb(251, 118, 123)'
                }, {
                    offset: 1,
                    color: 'rgb(204, 46, 72)'
                }])"""))
            .set_series_opts(label_opts=opts.LabelOpts(is_show=False))
            )
scatter.overlap(scatter1)
scatter.render_notebook()

```

10.China's Olympic Performance

```

background_color_js = (
    "new echarts.graphic.LinearGradient(1, 0, 0, 1, "
    "[{offset: 0, color: '#008B8B'}, {offset: 1, color: '#FF6347'}], false)"
)

athlete = CN_data.groupby(['Year', 'Season'])['Name'].nunique().reset_index()
athlete.columns = ['Year', 'Season', 'Nums']
athlete = athlete.sort_values(by="Year" , ascending=False)

s_bar = (
    Bar(init_opts=opts.InitOpts(theme='dark', width='1000px', height='300px'))
    .add_xaxis([row['Year'] for _, row in athlete[athlete.Season=='Summer'].iterrows()])
    .add_yaxis("Number of participants", [row['Nums'] for _, row in athlete[athlete.Season=='
Summer'].iterrows()],

```

```

        category_gap='40%',
        itemstyle_opts=opts.ItemStyleOpts(
            border_color='rgb(220,220,220)',
            color=JsCode("""new echarts.graphic.LinearGradient(0, 0, 0, 1,
                [{
                    offset: 1,
                    color: '#00BFFF'
                }, {
                    offset: 0,
                    color: '#32CD32'
                }]"""))
        .set_series_opts(label_opts=opts.LabelOpts(is_show=True,
            position='top',
            font_style='italic'))

        .set_global_opts(
            title_opts=opts.TitleOpts(title="Number of participants in Chinese Olympic Games over
the years - Summer Olympics", pos_left='center'),
            xaxis_opts=opts.AxisOpts(axislabel_opts=opts.LabelOpts(rotate=45)),
            legend_opts=opts.LegendOpts(is_show=False),
            yaxis_opts=opts.AxisOpts(axislabel_opts=opts.LabelOpts(margin=20, color="#ffffff63")),
            graphic_opts=[
                opts.GraphicImage(
                    graphic_item=opts.GraphicItem(
                        id="logo", right=0, top=0, z=-10, bounding="raw", origin=[75, 75]
                    ),
                    graphic_imagestyle_opts=opts.GraphicImageStyleOpts(
                        image="https://timgsa.baidu.com/timg?image&quality=80&size=b9999_10000
&sec=1586619952245&di=981a36305048f93eec791980acc99cf7&imgtype=0&src=http%3A%2
F%2Fimg5.mtime.cn%2Fmg%2F2017%2F01%2F06%2F172210.42721559.jpg",
                        width=1000,
                        height=600,
                        opacity=0.6,),
                )
            ],)
        )

w_bar = (
    Bar(init_opts=opts.InitOpts(theme='dark',width='1000px', height='300px'))
        .add_xaxis([row['Year'] for _, row in athlete[athlete.Season=='Winter'].iterrows()])
        .add_yaxis("Number of participants", [row['Nums'] for _, row in athlete[athlete.Season=='
Winter'].iterrows()])
        category_gap='50%',
        itemstyle_opts=opts.ItemStyleOpts(

```

```

border_color='rgb(220,220,220)',
color=JsCode("""new echarts.graphic.LinearGradient(0, 0, 0, 1,
    [{
        offset: 1,
        color: '#00BFFF'
    }, {
        offset: 0.8,
        color: '#FFC0CB'
    }, {
        offset: 0,
        color: '#40E0D0'
    }]"""))
.set_series_opts(label_opts=opts.LabelOpts(is_show=True,
    position='top',
    font_style='italic'))

.set_global_opts(
    title_opts=opts.TitleOpts(title="Number of participants in Chinese Olympic Games over
the years - Winter Olympics", pos_left='center'),
    xaxis_opts=opts.AxisOpts(axislabel_opts=opts.LabelOpts(rotate=45)),
    legend_opts=opts.LegendOpts(is_show=False),
    yaxis_opts=opts.AxisOpts(axislabel_opts=opts.LabelOpts(margin=20, color="#ffffff63")),
    graphic_opts=[
        opts.GraphicImage(
            graphic_item=opts.GraphicItem(
                id_"logo", right=0, top=-300, z=-10, bounding="raw", origin=[75, 75]
            ),
            graphic_imagestyle_opts=opts.GraphicImageStyleOpts(
                image="https://timgsa.baidu.com/timg?image&quality=80&size=b9999_10000
&sec=1586619952245&di=981a36305048f93eec791980acc99cf7&imgtype=0&src=http%3A%2
F%2Fimg5.mtime.cn%2Fimg%2F2017%2F01%2F06%2F172210.42721559.jpg",
                width=1000,
                height=600,
                opacity=0.6,),
        )
    ],)
)

page = (
    Page()
    .add(s_bar,)
    .add(w_bar,)
)
page.render_notebook()

```

```

background_color_js = (
    "new echarts.graphic.LinearGradient(1, 0, 0, 1, "
    "[{offset: 0, color: '#008B8B'}, {offset: 1, color: '#FF6347'}], false)"
)

CN_medals = CN_data.groupby(['Year', 'Season', 'Medal'])['Event'].nunique().reset_index()
CN_medals.columns = ['Year', 'Season', 'Medal', 'Nums']
CN_medals = CN_medals.sort_values(by="Year" , ascending=False)

s_bar = (
    Bar(init_opts=opts.InitOpts(theme='dark', width='1000px', height='300px'))
    .add_xaxis(sorted(list(set([row['Year'] for _, row in CN_medals[CN_medals.Season=='Summer'].iterrows()]))), reverse=True))
    .add_yaxis("gold medal", [row['Nums'] for _, row in CN_medals[(CN_medals.Season=='Summer') & (CN_medals.Medal=='Gold')].iterrows()],
        category_gap='20%',
        itemstyle_opts=opts.ItemStyleOpts(
            border_color='rgb(220,220,220)',
            color=JsCode("""new echarts.graphic.LinearGradient(0, 0, 0, 1,
                [{
                    offset: 0,
                    color: '#FFD700'
                }, {
                    offset: 1,
                    color: '#FFFFFF0'
                }])"""))
    .add_yaxis("silver medal", [row['Nums'] for _, row in CN_medals[(CN_medals.Season=='Summer') & (CN_medals.Medal=='Silver')].iterrows()],
        category_gap='20%',
        itemstyle_opts=opts.ItemStyleOpts(
            border_color='rgb(220,220,220)',
            color=JsCode("""new echarts.graphic.LinearGradient(0, 0, 0, 1,
                [{
                    offset: 0,
                    color: '#COCOCO'
                }, {
                    offset: 1,
                    color: '#FFFFFF0'
                }])"""))
    .add_yaxis("bronze medal", [row['Nums'] for _, row in CN_medals[(CN_medals.Season=='Summer') & (CN_medals.Medal=='Bronze')].iterrows()],
        category_gap='20%',

```



```

        itemstyle_opts=opts.ItemStyleOpts(
            border_color='rgb(220,220,220)',
            color=JsCode("""new echarts.graphic.LinearGradient(0, 0, 0, 1,
                [{
                    offset: 0,
                    color: '#DAA520'
                }, {
                    offset: 1,
                    color: '#FFFFFF0'
                }])"""))
        .set_series_opts(label_opts=opts.LabelOpts(is_show=True,
            position='top',
            font_style='italic'))

        .set_global_opts(
            title_opts=opts.TitleOpts(title="Number of Medals Received by China in Previous Olympic Games - Summer Olympics", pos_left='center'),
            xaxis_opts=opts.AxisOpts(axislabel_opts=opts.LabelOpts(rotate=45)),
            legend_opts=opts.LegendOpts(is_show=False),
            yaxis_opts=opts.AxisOpts(axislabel_opts=opts.LabelOpts(margin=20, color="#ffffff63")),
            graphic_opts=[
                opts.GraphicImage(
                    graphic_item=opts.GraphicItem(
                        id="logo", right=0, top=0, z=-10, bounding="raw", origin=[75, 75]
                    ),
                    graphic_imagestyle_opts=opts.GraphicImageStyleOpts(
                        image="https://timgsa.baidu.com/timg?image&quality=80&size=b9999_10000&sec=1586619952245&di=981a36305048f93eec791980acc99cf7&imgtype=0&src=http%3A%2F%2Fimg5.mtime.cn%2Fimg%2F2017%2F01%2F06%2F172210.42721559.jpg",
                        width=1000,
                        height=600,
                        opacity=0.6,
                    )
                ],
            )

        w_bar = (
            Bar(init_opts=opts.InitOpts(theme='dark', width='1000px', height='300px'))
            .add_xaxis(sorted(list(set([row['Year'] for _, row in CN_medals[CN_medals.Season=='Winter'].iterrows()]))), reverse=True))
            .add_yaxis("gold medal", [row['Nums'] for _, row in CN_medals[(CN_medals.Season=='Winter') & (CN_medals.Medal=='Gold')].iterrows()],
                category_gap='20%',
                itemstyle_opts=opts.ItemStyleOpts(

```

```

border_color='rgb(220,220,220)',
color=JsCode("""new echarts.graphic.LinearGradient(0, 0, 0, 1,
    [{
        offset: 0,
        color: '#FFD700'
    }, {
        offset: 1,
        color: '#FFFFFF0'
    }])"""))
.add_yaxis("silver medal", [row['Nums'] for _, row in CN_medals[(CN_medals.Season=='Winter') & (CN_medals.Medal=='Silver')].iterrows()],
    category_gap='20%',
    itemstyle_opts=opts.ItemStyleOpts(
        border_color='rgb(220,220,220)',
        color=JsCode("""new echarts.graphic.LinearGradient(0, 0, 0, 1,
            [{
                offset: 0,
                color: '#COCOCO'
            }, {
                offset: 1,
                color: '#FFFFFF0'
            }])"""))
.add_yaxis("bronze medal", [row['Nums'] for _, row in CN_medals[(CN_medals.Season=='Winter') & (CN_medals.Medal=='Bronze')].iterrows()],
    category_gap='20%',
    itemstyle_opts=opts.ItemStyleOpts(
        border_color='rgb(220,220,220)',
        color=JsCode("""new echarts.graphic.LinearGradient(0, 0, 0, 1,
            [{
                offset: 0,
                color: '#DAA520'
            }, {
                offset: 1,
                color: '#FFFFFF0'
            }])"""))
.set_series_opts(label_opts=opts.LabelOpts(is_show=True,
    position='top',
    font_style='italic'))
.set_global_opts(
    title_opts=opts.TitleOpts(title="Number of medals won by China in previous Olympic Games - Winter Olympics", pos_left='center'),
    xaxis_opts=opts.AxisOpts(axislabel_opts=opts.LabelOpts(rotate=45)),
    legend_opts=opts.LegendOpts(is_show=False),
    yaxis_opts=opts.AxisOpts(axislabel_opts=opts.LabelOpts(margin=20, color="#ffffff63")),

```

```

        graphic_opts=[
            opts.GraphicImage(
                graphic_item=opts.GraphicItem(
                    id="logo", right=0, top=-300, z=-10, bounding="raw", origin=[75, 75]
                ),
                graphic_imagestyle_opts=opts.GraphicImageStyleOpts(
                    image="https://timgsa.baidu.com/timg?image&quality=80&size=b9999_10000
&sec=1586619952245&di=981a36305048f93eec791980acc99cf7&imgtype=0&src=http%3A%2
F%2Fimg5.mtime.cn%2Fimg%2F2017%2F01%2F06%2F172210.42721559.jpg",
                    width=1000,
                    height=600,
                    opacity=0.6,),
            )
        ],)
    )

    page = (
        Page()
        .add(s_bar,)
        .add(w_bar,)
    )
    page.render_notebook()
    background_color_js = (
        "new echarts.graphic.LinearGradient(1, 0, 0, 1, "
        "[{offset: 0.5, color: '#FFC0CB'}, {offset: 1, color: '#FOFFFF'}, {offset: 0, color: '#EE82EE'}],
false)"
    )

    CN_events = CN_data[CN_data.Medal!='Gold'].groupby(['Year', 'Sport'])['Event'].nunique().reset_inde
ex()
    CN_events = CN_events.groupby(['Sport'])['Event'].sum().reset_index()
    CN_events.columns = ['Sport', 'Nums']

    data_pair = [(row['Sport'], row['Nums']) for _, row in CN_events.iterrows()]

    wc = (WordCloud(init_opts=opts.InitOpts(bg_color=JsCode(background_color_js), width='1000px',
height='600px'))
        .add("", data_pair, word_size_range=[30, 80])
        .set_global_opts(title_opts=opts.TitleOpts(title="China has won gold medals in sports events", p
os_left="center",
                                title_textstyle_opts=opts.TextStyleOpts(color="white", font_size
=20)))
    )

```

```
wc.render_notebook()
```

11. Olympic events dominated by a single country

```
f1 = lambda x: max(x['Event']) / sum(x['Event'])
f2 = lambda x: x.sort_values('Event', ascending=False).head(1)

t_data = data[(data.Medal=='Gold') & (data.Year>=2000) & (data.Season=='Summer')].groupby(['Year', 'Sport', 'region'])['Event'].nunique().reset_index()
t_data = t_data.groupby(['Sport', 'region'])['Event'].sum().reset_index()
t1 = t_data.groupby(['Sport']).apply(f2).reset_index(drop=True)
t2 = t_data.groupby(['Sport'])['Event'].sum().reset_index()
t_data = pd.merge(t1, t2, on='Sport', how='inner')
t_data['gold_rate'] = t_data.Event_x / t_data.Event_y
t_data = t_data.sort_values('gold_rate', ascending=False).reset_index(drop=True)

t_data = t_data[(t_data.gold_rate>=0.5) & (t_data.Event_y>=10)]

background_color_js = (
    "new echarts.graphic.LinearGradient(1, 0, 0, 1, "
    "[{offset: 0, color: '#008B8B'}, {offset: 1, color: '#FF6347'}], false)"
)

fn = """
function(params) {
    if(params.name == 'Other countries')
        return '\n\n\n\n' + params.name + ' : ' + params.value ;
    return params.seriesName+ '\n' + params.name + ' : ' + params.value;
}
"""

def new_label_opts():
    return opts.LabelOpts(formatter=JsCode(fn), position="center")

pie = Pie(init_opts=opts.InitOpts(theme='dark', width='1000px', height='1000px'))
idx = 0

for _, row in t_data.iterrows():

    if idx % 2 == 0:
        x = 30
        y = int(idx/2) * 22 + 18
    else:
        x = 70
        y = int(idx/2) * 22 + 18
```

```

idx += 1
pos_x = str(x)+'%'
pos_y = str(y)+'%'
pie.add(
    row['Sport'],
    [[row['region'], row['Event_x']], ['Other countries', row['Event_y']-row['Event_x']]],
    center=[pos_x, pos_y],
    radius=[70, 100],
    label_opts=new_label_opts(),)

pie.set_global_opts(
    title_opts=opts.TitleOpts(title="Projects governed by a single country",
                               subtitle='Statistical cycle: Since the 2000 Sydney Olympics',
                               pos_left="center",
                               title_textstyle_opts=opts.TextStyleOpts(color="white", font_size=20)),
    legend_opts=opts.LegendOpts(is_show=False),
)

pie.render_notebook()

```