import pandas as pd

df = pd.read\_csv('/mnt/vgsales.csv') # 加载数据集

print('数据基本信息：')

df.info()

rows, columns = df.shape # 查看数据集行数和列数

if rows < 100 and columns < 20:

# 短表数据（行数少于100且列数少于20）查看全量数据信息

print('数据全部内容信息：')

print(df.to\_markdown(numalign='left', stralign='left'))

else:

print('数据前几行内容信息：')

print(df.head().to\_markdown(numalign='left', stralign='left'))

数据基本信息：

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 16598 entries, 0 to 16597

Data columns (total 11 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Rank 16598 non-null int64

1 Name 16598 non-null object

2 Platform 16598 non-null object

3 Year 16327 non-null float64

4 Genre 16598 non-null object

5 Publisher 16540 non-null object

6 NA\_Sales 16598 non-null float64

7 EU\_Sales 16598 non-null float64

8 JP\_Sales 16598 non-null float64

9 Other\_Sales 16598 non-null float64

10 Global\_Sales 16598 non-null float64

dtypes: float64(6), int64(1), object(4)

memory usage: 1.4+ MB

数据前几行内容信息：

| | Rank | Name | Platform | Year | Genre | Publisher | NA\_Sales | EU\_Sales | JP\_Sales | Other\_Sales | Global\_Sales |

|:---|:-------|:-------------------------|:-----------|:-------|:-------------|:------------|:-----------|:-----------|:-----------|:--------------|:---------------|

| 0 | 1 | Wii Sports | Wii | 2006 | Sports | Nintendo | 41.49 | 29.02 | 3.77 | 8.46 | 82.74 |

| 1 | 2 | Super Mario Bros. | NES | 1985 | Platform | Nintendo | 29.08 | 3.58 | 6.81 | 0.77 | 40.24 |

| 2 | 3 | Mario Kart Wii | Wii | 2008 | Racing | Nintendo | 15.85 | 12.88 | 3.79 | 3.31 | 35.82 |

| 3 | 4 | Wii Sports Resort | Wii | 2009 | Sports | Nintendo | 15.75 | 11.01 | 3.28 | 2.96 | 33 |

| 4 | 5 | Pokemon Red/Pokemon Blue | GB | 1996 | Role-Playing | Nintendo | 11.27 | 8.89 | 10.22 | 1 | 31.37 |

根据执行结果，数据集包含 16598 行 11 列，其中Year列有部分缺失值，推测参与分析的列为Year、Genre和Platform。

下面，我将删除Year列的缺失值，以确保字段在日期类型转换时的完整性。

# 删除年份列的缺失值

df = df.dropna(subset=['Year'])

# 将年份列转为日期类型

df['Year'] = pd.to\_datetime(df['Year'], format='%Y')

# 保存处理后的 DataFrame

csv\_path = '/mnt/vgsales\_converted.csv'

df.to\_csv(csv\_path)

**图一**

import matplotlib.pyplot as plt

# 从数据中提取出年份和全球销售额列

total\_sale\_df = df[['Year', 'Global\_Sales']]

yearly\_sale\_df = total\_sale\_df.groupby('Year')['Global\_Sales'].sum().reset\_index()

# 设置图片清晰度

plt.rcParams['figure.dpi'] = 300

# 设置 matplotlib 支持中文

plt.rcParams['font.sans-serif'] = ['WenQuanYi Zen Hei']

# 设置画布大小

plt.figure(figsize=(16, 9))

# 绘制折线图

plt.plot(yearly\_sale\_df['Year'], yearly\_sale\_df['Global\_Sales'])

# 设置标题和坐标轴标签

plt.title('游戏全球销售收入')

plt.xlabel('年份')

plt.ylabel('收入总额')

# 设置时间刻度旋转 45 度

plt.xticks(rotation=45)

# 自动调整子图布局

plt.tight\_layout()

# 显示图表

plt.show()

**图二**

# 数据提取

country\_distribution\_df = df[['NA\_Sales', 'EU\_Sales', 'JP\_Sales', 'Other\_Sales']]

# 设置 figure size

plt.figure(figsize=(6, 9))

# 计算各地区总销售额

country\_distribution\_df['NA\_Sales\_sum'] = country\_distribution\_df['NA\_Sales'].sum()

country\_distribution\_df['EU\_Sales\_sum'] = country\_distribution\_df['EU\_Sales'].sum()

country\_distribution\_df['JP\_Sales\_sum'] = country\_distribution\_df['JP\_Sales'].sum()

country\_distribution\_df['Other\_Sales\_sum'] = country\_distribution\_df['Other\_Sales'].sum()

#比例总和

country\_distribution\_sum = country\_distribution\_df.iloc[0, 4:]

total\_sum = country\_distribution\_sum.sum()

# 设置保留两位小数

country\_distribution\_sum\_2 = [round(x / total\_sum, 2) for x in country\_distribution\_sum]

# 设置饼图标签标签和颜色

labels = ['北美', '欧洲', '日本', '其他']

colors = ['red', 'yellow', 'green', 'pink']

# 设置字体大小

plt.rcParams.update({'font.size': 8})

# 绘制饼图

plt.pie(country\_distribution\_sum\_2, labels=labels, colors=colors, autopct='%1.2f%%', textprops={'fontsize': 6})

# 设置饼图标题

plt.title('各地区销售比例')

# 显示图形

plt.show()

**图三**

# 选择要分组的列

group\_cols = ['Genre']

# 设置画布大小

plt.figure(figsize=(12, 8))

# 分组统计并绘图

genre\_total\_sales = df.groupby(group\_cols)['Global\_Sales'].sum().plot(kind='bar', title='不同游戏类型的全球销量分布', ylabel='全球销量')

# 输出结果到终端

print('不同游戏类型的全球销量分布：')

for genre, total\_sales in df.groupby(group\_cols)['Global\_Sales'].sum().items():

print(f'{genre}: {total\_sales}')

# 添加数据标签

for i, v in enumerate(df.groupby(group\_cols)['Global\_Sales'].sum()):

genre\_total\_sales.text(i, v, str(v), ha='center')

# 显示图形

plt.show()

不同游戏类型的全球销量分布：

Action: 1722.88

Adventure: 234.8

Fighting: 444.05

Misc: 797.62

Platform: 829.15

Puzzle: 242.22

Racing: 726.77

Role-Playing: 923.84

Shooter: 1026.2

Simulation: 390.16

Sports: 1309.24

Strategy: 173.43

**图四**

# 发行商全球销量前五柱状图

publisher\_total\_sales = df.groupby('Publisher')['Global\_Sales'].sum().nlargest(10, keep='all').sort\_values(ascending=False)

# 设置画布大小

plt.figure(figsize=(12, 9))

# 绘制柱状图

publisher\_total\_sales.plot(kind='bar', title='按发行商划分全球销量柱状图', xlabel='发行商',ylabel='全球总销量')

# 添加数据标签

for i, v in enumerate(publisher\_total\_sales):

plt.text(i, v, f'{v:.2f}', ha='center', va='bottom')

# 显示图形

plt.show()

**图五**

# 引入 matplotlib.pyplot 和 pandas 库

import matplotlib.pyplot as plt

import pandas as pd

# 以 gbk 编码读取数据集

df = pd.read\_csv('/mnt/vgsales\_converted.csv', encoding='gbk')

# 根据不同游戏平台进行分组，并求每组的全球销量

group\_df = df.groupby('Platform')['Global\_Sales'].sum().sort\_values(ascending=False)

# 设置图片清晰度

plt.rcParams['figure.dpi'] = 300

# 设置 matplotlib 支持中文

plt.rcParams['font.sans-serif'] = ['WenQuanYi Zen Hei']

# 创建画布

plt.figure(figsize=(10, 8))

# 绘制柱状图

plt.bar(group\_df.index, group\_df)

# 设置图形标题

plt.title('各游戏平台全球销量')

# 设置 X 轴名称

plt.xlabel('游戏平台')

# 设置 Y 轴名称

plt.ylabel('全球销量')

# 设置 X 轴刻度旋转 90 度

plt.xticks(rotation = 90)

# 添加数据标签

for i in range(len(group\_df.index)):

plt.text(i, group\_df[i], group\_df[i], ha = 'center', size = 10)

# 显示图形

plt.show()

# 查看图形结果

print('各游戏平台全球销量：\n', group\_df)

各游戏平台全球销量：

Platform

PS2 1233.46

X360 969.61

PS3 949.35

Wii 909.81

DS 818.96

PS 727.39

GBA 313.56

PSP 291.71

PS4 278.10

PC 255.05

GB 254.42

XB 252.09

NES 251.07

3DS 246.28

N64 218.21

SNES 200.05

GC 197.14

XOne 141.06

2600 86.57

WiiU 81.86

PSV 61.63

SAT 33.59

GEN 28.36

DC 15.97

SCD 1.87

NG 1.44

WS 1.42

TG16 0.16

3DO 0.10

GG 0.04

PCFX 0.03

Name: Global\_Sales, dtype: float64

**图六**

# 读取数据集

df = pd.read\_csv('/mnt/vgsales\_converted.csv', encoding='gbk')

# 提取相关游戏类型的数据

genres\_df = df[df['Genre'].isin(['Action', 'Sports', 'Role-Playing', 'Shooter'])]

# 按年份和游戏类型分组，计算每年各种游戏类型的全球总销量

yearly\_sale\_df = genres\_df.groupby(['Year', 'Genre'])['Global\_Sales'].sum().reset\_index()

# 设置图片清晰度

plt.rcParams['figure.dpi'] = 300

# 设置 matplotlib 支持中文

plt.rcParams['font.sans-serif'] = ['WenQuanYi Zen Hei']

# 创建画布

plt.figure(figsize=(16, 9))

# 绘制不同游戏类型的折线图

for genre in yearly\_sale\_df['Genre'].unique():

genre\_data = yearly\_sale\_df[yearly\_sale\_df['Genre'] == genre]

plt.plot(genre\_data['Year'].astype(str), genre\_data['Global\_Sales'], label=genre)

# 设置标题和坐标轴标签

plt.title('各大主要游戏类型 yearly 销量变化趋势')

plt.xlabel('年份')

plt.ylabel('全球销量')

# 设置图例

plt.legend()

plt.legend()

# 设置 x 轴标签旋转 45 度

plt.xticks(rotation=45)

plt.xticks(rotation=45)

# 获取图形边界范围

ylim = plt.ylim()

# 添加横向网格线

for y in range(int(ylim[0]), int(ylim[1]) + 1):

plt.axhline(y=y, color='gray', linewidth=0.5)

# 显示图形

plt.show()

**图七**

# 引入第三方库

import matplotlib.pyplot as plt

# 设置图片分辨率

plt.rcParams['figure.dpi'] = 300

# 设置 matplotlib 支持中文

plt.rcParams['font.sans-serif'] = ['WenQuanYi Zen Hei']

# 绘制柱状图

plt.figure(figsize=(12, 8))

pivot\_percentage\_df.plot(kind='bar', ax=plt.gca())

# 设置标题和坐标轴标签

plt.title('Action、Sports、Role - Playing、Shooter类游戏在PS2、X360、PS3、Wii上的销量占比柱状图')

plt.xlabel('游戏类型')

plt.ylabel('销量占比')

# 添加数据标签

for i, v in enumerate(pivot\_percentage\_df.values):

for j, val in enumerate(v):

plt.text(j, val, f'{val:.2f}', ha='center', va='bottom')

# 设置图例

plt.legend(bbox\_to\_anchor=(1.05, 1), loc='upper left', borderaxespad=0.)

# 使 x 轴标签刻度旋转 45 度

plt.xticks(rotation = 45)

# 显示图形

plt.show()

**图八**

# 清洗数据，只保留日期部分

df['Year'] = df['Year'].astype(str).str.split('/').str[0]

# 将 Year 字段转换为日期类型

df['Year'] = pd.to\_datetime(df['Year'])

# 根据发布年份和主要游戏平台分组，计算每年各平台的全球总销量

yearly\_platform\_sale = df.groupby([df['Year'].dt.year, 'Platform'])['Global\_Sales'].sum().unstack()

# 过去几十年各游戏平台的全球销量变化折线图

plt.figure(figsize=(16, 9))

for platform in yearly\_platform\_sale.columns:

if platform in ['PS2', 'X360', 'PS3', 'Wii']:

plt.plot(yearly\_platform\_sale.index.astype(str), yearly\_platform\_sale[platform], label=platform)

plt.title('过去几十年主要游戏平台的全球销量变化')

plt.xlabel('发行年份')

plt.ylabel('全球销量')

plt.legend()

# 设置 x 轴标签旋转 45 度

plt.xticks(rotation=45)

# 添加横向网格线

plt.grid(axis='y', linestyle='--')

# 获取图形边界范围

ylim = plt.ylim()

# 添加横向网格线

for y in range(int(ylim[0]), int(ylim[1]) + 1):

plt.axhline(y=y, color='gray', linewidth=0.5)

# 显示图像

plt.show()