

目录

当当网图书销售数据分析报告	2
第一章 绪论	2
1.1 引言	2
1.2 课题开发的目的是和意义	3
1.3 实验环境	4
第二章 数据预处理	4
2.1 数据爬取：	4
2.2 数据清洗：	8
第三章 数据分析及可视化	11
3.1 图书评论数分析	11
3.2 书籍价格分析	12
3.3 价格排名靠前的书籍分布	14
3.5 评论数词云分析	18
3.6 书籍出版时间分布	19
3.7 2024 年每月平均打折情况	20
3.8 2024 年畅销书目分析	22
3.9 书本出版时间与打折力度的关系	23
第四章 实验总结	24
第五章 实验思考	24
第六章 实验结论	25

当当网图书销售数据分析报告

第一章 绪论

1.1 引言

随着互联网技术的飞速发展，电子商务已成为人们日常生活中不可或缺的一部分。在线购物平台如当当网、亚马逊等，提供了丰富的商品信息和便捷的购物体验。在这些平台中，图书作为重要的商品类别之一，吸引了大量读者的关注。为了深入了解图书市场的动态和趋势，分析读者的阅读偏好和购买行为，有必要从在线购物平台获取图书的详细信息。

当当网作为中国知名的图书销售平台，拥有丰富的图书资源和庞大的用户群体。其网站上的图书榜单和畅销书排行，反映了当前读者的阅读热点和购买趋势。因此，通过爬取当当网的图书信息，我们可以获取宝贵的市场数据，为图书出版、销售和推广提供决策支持。

当当网是中国较大的在线书店之一，提供图书、音像制品、电子书等商品。进行热销书籍的分析不仅可以了解当前流行的书籍类型和内容，还可以分析读者的购买偏好和特点，这些书籍往往能够反映出当前读者群体的兴趣点和趋势。进行热销书籍分析的目的是为了深入了解这些畅销书籍的销售情况、读者群体、市场趋势等信息，以此来帮助书店或出版商更好地定位市场需求，以及制定相应的营销策略。

本实验借助 python 等工具，对当当网畅销榜数据进行一个研究

分析，希望通过数据分析和可视化等方法，帮助图书市场进行一个良性发展。对于读者而言，该榜单由多个分类别组成，涵盖文学、教育、社科、儿童等不同领域，由于当下互联网的普及使得在线购物比线下实体店购物更加普遍和便利，热销书籍榜单能够反映出广大读者的阅读需求，如果读者想寻找优秀的图书，可以通过查看榜单进行参考和借鉴，形成读者、作者、书店、出版社四方之间的良性循环。

1.2 课题开发的目的是和意义

自 1993 年初 Matthew Gray' s Wandered 在麻省理工学院开发出有史记载的第一个网络爬虫以来，爬虫技术历经 20 多年的发展，技术已日趋多样。但是同时随着爬虫的不断壮大发展，各大网站反爬虫的机制也在不断完善。对于当当网来说，网络当中很多绕过当当网爬虫检测的方法也大多已经失效。因此本设计将会更加详细的爬取当当网图书的销售数据。

其次，尽管目前爬虫的发展形势一片火热，数据分析也正处于时代的浪潮之上，针对当当网图书销售数据这一方面，对于这部分数据的爬取与分析也并不是十分的详尽，而能够将二者结合在一起的讨论的实际上并不是很多。因此将当当网手机的销售数据的爬取与分析放在一起讨论虽然无法做到尽善尽美，但实际在一定程度上做到了对目前当当图书信息的爬取与数据分析两个方面知识与实际操作地更新、补充与整合。

综上所述，当当网销售数据的爬取与可视化分析的意义包括以下

方面：

- (1) 帮助消费者清楚的了解市场中图书的销售情况。
- (2) 帮助消费者更好的根据预算以及当下当当网图书销售情况选购图书。

1.3 实验环境

操作系统：Windows 11

Python 版本：Python 3.12

库和依赖：requests、lxml、pandas

开发工具：Spyder、Jupyter Notebook

第二章 数据预处理

2.1 数据爬取：

使用 Python 的 requests 模块编写网络爬虫，爬取的目标选取当当网历史销售排行，爬取的时间范围是 2020 年至 2024 年。首选在当当网销售排行榜界面下将所有上榜的图书链接爬取到本地，再一次将所有链接遍历爬取，将可用的信息存入本地数据库，以便下一步分使用。

```

# 导入所需的库
import requests # 用于发送HTTP请求
from lxml import etree # 用于解析HTML和XML文档
import pandas as pd # 用于数据分析和处理

# 设置请求头，模拟浏览器访问
headers = {
    "User-Agent": "xxxsh; Intel Mac OS X 10_14_1 AppleWebKit/537.36 (KHTML, Like Gecko)"
}

# 定义函数，用于获取当当网的图书信息
def get_dangdang_info(url, year):
    # 发送HTTP GET请求获取网页内容
    html = requests.get(url, headers=headers)
    # 设置正确的编码，以避免乱码
    html.encoding = html.apparent_encoding
    # 使用lxml的etree解析HTML文档
    selector = etree.HTML(html.text)
    # 通过XPath查询获取包含图书信息的div元素列表
    datas = selector.xpath('//div[@class="bang_list_box"]')
    # 初始化一个空列表，用于存储图书信息
    book_info = []
    # 遍历每个包含图书信息的div元素
    for data in datas:
        # 提取排名信息
        Ranks = data.xpath('ul/li/div[1]/text()')
        # 提取书名链接中的文本（书名）
        names = data.xpath('ul/li/div[3]/a/text()')
        # 提取评论链接中的文本（评论数）
        pingluns = data.xpath('ul/li/div[4]/a/text()')
        # 提取作者链接中的文本（作者）
        authors = data.xpath('ul/li/div[5]/a/text()')

```

```

# 提取评论链接中的文本（评论数）
        pingluns = data.xpath('ul/li/div[4]/a/text()')
# 提取作者链接中的文本（作者）
        authors = data.xpath('ul/li/div[5]/a/text()')
# 提取出版社信息
        chubans = data.xpath('ul/li/div[6]/span/text()')
# 提取价格信息
        jiages = data.xpath('ul/li/div[7]/p[1]/span[1]/text()')
# 提取原价信息
        yuanjias = data.xpath('ul/li/div[7]/p[1]/span[2]/text()')
# 提取折扣信息
        discounts = data.xpath('ul/li/div[7]/p[1]/span[3]/text()')
# 提取书名链接的URL
        urls = data.xpath('ul/li/div[3]/a/@href')
# 使用zip函数同时遍历上述所有列表，并组装成包含所有图书信息的列表
        for Rank, url, name, pinglun, author, chuban, jiage, yuanjia, discount in zip(Ranks, urls, names, pingluns, authors, chubans, jiages, yuanjias, discounts):
            book_info.append([year, Rank, url, name, pinglun, author, chuban, jiage, yuanjia, discount])
# 返回包含所有图书信息的列表
    return book_info

# 主程序入口
if __name__ == '__main__':
    # 初始化一个空列表，用于存储近四年的图书数据
    book_data = []
    # 遍历2020年到2023年
    for i in range(2020, 2024):
        # 遍历每一页（1到25页，当当网的分页通常是每页显示一定数量的图书）
        for j in range(1, 26):
            url = f'http://bang.dangdang.com/books/bestsellers/01.00.00.00.00-year-{i}-0-1-{j}'
            print(url) # 打印URL，方便调试
            # 调用函数获取当前页面的图书信息，并添加到book_data列表中
            book_data += get_dangdang_info(url, i)
# 使用pandas的DataFrame创建一个新的DataFrame对象，用于存储图书数据
df = pd.DataFrame(book_data, columns=['年份', '排名', '链接', '书名', '评论数', '作者', '出版年份', '价格', '原价', '折扣'])
# 将DataFrame对象保存为Excel文件
df.to_excel('2020-2023data.xlsx', index=False)
# 打印提示信息，告知用户数据已保存
print("2020-2023年图书数据已保存")

```

以上代码进行了当当网 2020-2023 年四年的数据爬取，结果存取到 2020-2023data.xlsx 文件当中。

查看数据：

A1	年份									
	A	B	C	D	E	F	G	H	I	J
1	年份	排名	链接	书名	评论数	作者	出版年份	价格	原价	折扣
2	2020	1.	http://p.	你当像鸟	1777074条	塔拉	2019-11-	¥34.80	¥59.00	5.9折
3	2020	2.	http://p.	人间失格	2230022条	韦斯特弗	2015-08-	¥9.75	¥25.00	3.9折
4	2020	3.	http://p.	乌合之众	756895条	新经典	2018-04-	¥10.14	¥26.00	3.9折
5	2020	4.	http://p.	神奇校车	1800369条	太宰治	2018-05-	¥198.00	¥198.00	10.0折
6	2020	5.	http://p.	月亮与六	1657165条	杨伟	2017-01-	¥19.90	¥39.80	5.0折
7	2020	6.	http://p.	人生海海	1245656条	古斯塔夫	2019-04-	¥55.00	¥55.00	10.0折
8	2020	7.	http://p.	正面管教	2218125条	马晓佳	2016-07-	¥32.30	¥38.00	8.5折
9	2020	8.	http://p.	云边有个	1716507条	乔安娜柯	2018-07-	¥28.00	¥42.00	6.7折
10	2020	9.	http://p.	小熊和最	1560079条	布鲁斯·	2007-11-	¥17.50	¥35.00	5.0折
11	2020	10.	http://p.	啊2.0（大	794854条	毛姆	2020-08-	¥24.00	¥39.60	6.1折
12	2020	11.	http://p.	马尔克斯	2463868条	作家榜经	2017-08-	¥39.60	¥55.00	7.2折
13	2020	12.	http://p.	神奇校车	1587548条	徐淳刚	2014-04-	¥82.50	¥150.00	5.5折
14	2020	13.	http://p.	你就是孩	864562条	诗人	2020-03-	¥27.00	¥36.00	7.5折
15	2020	14.	http://p.	三体(3册)	2443129条	大星文化	2010-11-	¥88.40	¥93.00	9.5折
16	2020	15.	http://p.	断舍离（	1101915条	高更	2013-07-	¥23.00	¥32.00	7.2折
17	2020	16.	http://p.	少年读史	1429922条	麦家	2015-09-	¥78.00	¥100.00	7.8折
18	2020	17.	http://p.	流浪的地	1717768条	新经典	2018-05-	¥11.20	¥29.00	3.9折
19	2020	18.	http://p.	活着	2980683条	简·尼尔	2012-08-	¥22.10	¥28.00	7.9折
20	2020	19.	http://p.	窗边的小	1776971条	张嘉佳	2018-05-	¥27.20	¥39.50	6.9折

共 2000 条数据，属性值包含以下字段，年份、排名、链接、书名、评论数、作者、出版年份、价格、原价、折扣。

```
def get_dangdang_info(url, month):
    # 发送HTTP GET请求获取网页内容
    html = requests.get(url, headers=headers)
    # 设置正确的编码，以避免乱码
    html.encoding = html.apparent_encoding
    # 使用lxml的etree解析HTML文档
    selector = etree.HTML(html.text)
    # 通过XPath查询获取包含图书信息的div元素列表
    datas = selector.xpath('//div[@class="bang_list_box"]')
    # print(datas)
    # 初始化一个空列表，用于存储图书信息
    book_info = []
    # 遍历每个包含图书信息的div元素
    for data in datas:
        # 提取排名信息
        Ranks = data.xpath('ul/li/div[1]/text()')
        # 提取书名链接中的文本（书名）
        names = data.xpath('ul/li/div[3]/a/text()')
        # 提取评论链接中的文本（评论数）
        pingluns = data.xpath('ul/li/div[4]/a/text()')
        # 提取作者链接中的文本（作者）
        authors = data.xpath('ul/li/div[5]/a/text()')
        # 提取出版社信息
        chubans = data.xpath('ul/li/div[6]/span/text()')
        # 提取价格信息
        jiages = data.xpath('ul/li/div[7]/p[1]/span[1]/text()')
        # 提取原价信息
        yuanyias = data.xpath('ul/li/div[7]/p[1]/span[2]/text()')
        # 提取折扣信息
        discounts = data.xpath('ul/li/div[7]/p[1]/span[3]/text()')
        # 提取书名链接的URL
        urls = data.xpath('ul/li/div[3]/a/@href')
        # 使用zip函数同时遍历上述所有列表，并组装成包含所有图书信息的列表
        for Rank, url, name, pinglun, author, chuban, jiage, yuanyia, discount in zip(Ranks, urls, names, pingluns, authors, chubans, jiages, yuanyias, discounts):
            book_info.append([month, Rank, url, name, pinglun, author, chuban, jiage, yuanyia, discount])
    # 返回包含所有图书信息的列表
    return book_info
```

```
# 主程序入口
if __name__ == '__main__':
    # 初始化一个空列表，用于存储近四年的图书数据
    book_data = []
    # 遍历1-11月
    for i in range(1,12):
        # 遍历每一页（1到25页，当当网的分页通常是每页显示一定数量的图书）
        for j in range(1, 26):
            url = f'http://bang.dangdang.com/books/bestsellers/01.00.00.00.00-month-2024-{i}-1-{j}'
            print(url) # 打印URL
            # 调用函数获取当前页面的图书信息，并添加到book_data列表中
            book_data += get_dangdang_info(url, i)
    # 使用pandas的DataFrame创建一个新的DataFrame对象，用于存储图书数据
    df = pd.DataFrame(book_data, columns=['月份', '排名', '链接', '书名', '评论数', '作者', '出版年份', '价格', '原价', '折扣'])
    # 将DataFrame对象保存为Excel文件
    df.to_excel('month1-11.xlsx', index=False)
    # 打印提示信息，告知用户数据已保存
    print("2024年month1-11数据已保存")
```

以上代码进行了当当网 2024 年 1-11 月份的数据爬取，结果存取到 month1-11.xlsx 文件当中。

查看数据：

	A	B	C	D	E	F	G	H	I	J
1	月份	排名	链接	书名	评论数	作者	出版年份	价格	原价	折扣
2	1	1.	http://p1	额尔古纳	1111768条	迟子建	2019-06-	¥32.00	¥32.00	10.0折
3	1	2.	http://p1	我们生活	711534条	余华	2015-01-	¥35.00	¥35.00	10.0折
4	1	3.	http://p1	我与地坛	1583929条	史铁生	2011-06-	¥16.00	¥29.00	5.5折
5	1	4.	http://p1	读读童谣	559020条	曹文轩	2018-01-	¥27.10	¥39.80	6.8折
6	1	5.	http://p1	中国古代	231528条	陈先云	2019-01-	¥24.50	¥25.00	9.8折
7	1	6.	http://p1	克雷洛夫	204379条	曹文轩	2019-01-	¥24.50	¥25.00	9.8折
8	1	7.	http://p1	经典常谈	128064条	陈先云	2023-01-	¥20.10	¥26.80	7.5折
9	1	8.	http://p1	伊索寓言	187181条	曹文轩	2019-01-	¥24.50	¥25.00	9.8折
10	1	9.	http://p1	活着（余	1591886条	陈先云	2021-10-	¥31.00	¥45.00	6.9折
11	1	10.	http://p1	病隙碎笔	473076条	朱自清	2021-11-	¥45.60	¥48.00	9.5折
12	1	11.	http://p1	繁花批注	53567条	曹文轩	2023-06-	¥63.70	¥98.00	6.5折
13	1	12.	http://p1	俗世奇人	464966条	陈先云	2018-04-	¥28.00	¥28.00	10.0折
14	1	13.	http://p1	繁花（原	126125条	余华	2020-10-	¥68.00	¥68.00	10.0折
15	1	14.	http://p1	大头儿子	231606条	新经典	2018-12-	¥10.90	¥16.00	6.8折
16	1	15.	http://p1	【印签版	593794条	史铁生	2022-08-	¥68.00	¥68.00	10.0折
17	1	16.	http://p1	骆驼祥子	370181条	博集天卷	2018-04-	¥26.00	¥26.00	10.0折
18	1	17.	http://p1	骆驼祥子	258220条	金宇澄	2017-06-	¥21.60	¥28.80	7.5折
19	1	18.	http://p1	被讨厌的	1535847条	沈宏非	2020-03-	¥29.80	¥55.00	5.4折
20	1	19.	http://p1	小马过河	39530条	批注	2022-01-	¥15.00	¥48.00	3.1折
21	1	20.	http://p1	十万个为	323942条	姜庆共	2018-10-	¥19.40	¥22.80	8.5折
22	1	21.	http://p1	苏东坡传	1183166条	林语堂	2018-01-	¥49.40	¥52.00	9.5折
23	1	22.	http://p1	鲁滨逊漂	410469条	博集天卷	2012-06-	¥12.60	¥14.80	8.5折
24	1	23.	http://p1	汤姆索亚	540398条	丹尼尔	2021-06-	¥12.60	¥14.80	8.5折
25	1	24.	http://p1	课文作家	105332条	笛福	2019-09-	¥11.25	¥25.00	4.5折
26	1	25.	http://p1	快乐读书	95698条	马克	2018-11-	¥52.90	¥62.20	8.5折
27	1	26.	http://p1	十万个为	82714条	吐温	2020-01-	¥43.50	¥75.00	5.8折
28	1	27.	http://p1	生死疲劳	735021条	鲁冰	2022-01-	¥45.40	¥69.90	6.5折
29	1	28.	http://p1	钢铁是怎	256323条	王林	2018-06-	¥36.00	¥36.00	10.0折
30	1	29.	http://p1	长安的荔	550185条	马克	2022-10-	¥29.00	¥45.00	6.4折
31	1	30.	http://p1	小学生漫	114273条	吐温	2023-08-	¥18.80	¥100.00	1.9折
32	1	31.	http://p1	海底两万	214329条	笛福	2017-06-	¥29.10	¥38.80	7.5折

共 5499 条数据，属性值包含以下字段，月份、排名、链接、书名、评论数、作者、出版年份、价格、原价、折扣。

2.2 数据清洗：

根据爬取数据的类型，将各类数据的格式进行修正，删除重复数据，修补缺失数据，最终得到可以直接用于分析使用的数据，并保存到本地 excel 中。

先对 2020-2023 年的数据进行处理：

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1999 entries, 0 to 1998
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype  
---  --
 0   年份        1999 non-null  int64  
 1   排名        1999 non-null  float64
 2   链接        1999 non-null  object  
 3   书名        1999 non-null  object  
 4   评论数      1999 non-null  object  
 5   作者        1999 non-null  object  
 6   出版年份    1999 non-null  object  
 7   价格        1999 non-null  object  
 8   原价        1999 non-null  object  
 9   折扣        1999 non-null  object  
dtypes: float64(1), int64(1), object(8)
memory usage: 156.3+ KB
```

```
# 删除价格和原价列的¥
df['价格'] = df['价格'].str.replace('¥', '')
df['原价'] = df['原价'].str.replace('¥', '')
```

```
# 删除评论数列的“条评论”
df['评论数'] = df['评论数'].str.replace('条评论', '')
# 删除折扣列的“折”字
df['折扣'] = df['折扣'].str.replace('折', '')
```

```
# 将每一步的处理变成数值类型
df['价格'] = df['价格'].astype(float)
df['原价'] = df['原价'].astype(float)
df['折扣'] = df['折扣'].astype(float)
df['评论数'] = pd.to_numeric(df['评论数'], errors='coerce')
```

```
# 将出版年份转成时间格式
df['出版年份'] = pd.to_datetime(df['出版年份'])
```

```
print('df数据初步处理后的显示')
df.describe().T
df.to_excel('2020-2023_good_data.xlsx', index=False)
```

查看清洗后的数据：

	C	D	E	F	G	H	I	J
1	链接	书名	评论数	作者	出版年份	价格	原价	折扣
2	http://p	你当像鸟	1777074	塔拉	2019年11月1日	34.8	59	5.9
3	http://p	人间失格	2230022	韦斯特弗	2015年8月1日	9.75	25	3.9
4	http://p	乌合之众	756895	新经典	2018年4月6日	10.14	26	3.9
5	http://p	神奇校车	1800369	太宰治	2018年5月10日	198	198	10
6	http://p	月亮与六	1657165	杨伟	2017年1月10日	19.9	39.8	5
7	http://p	人生海海	1245656	古斯塔夫	2019年4月16日	55	55	10
8	http://p	正面管教	2218125	马晓佳	2016年7月1日	32.3	38	8.5
9	http://p	云边有个	1716507	乔安娜柯	2018年7月1日	28	42	6.7
10	http://p	小熊和最	1560079	布鲁斯·	2007年11月1日	17.5	35	5
11	http://p	啊2.0 (才	794854	毛姆	2020年8月1日	24	39.6	6.1
12	http://p	马尔克斯	2463868	作家榜经	2017年8月1日	39.6	55	7.2
13	http://p	神奇校车	1587548	徐淳刚	2014年4月1日	82.5	150	5.5
14	http://p	你就是孩	864562	诗人	2020年3月1日	27	36	7.5
15	http://p	三体(3册	2443129	大星文化	2010年11月1日	88.4	93	9.5
16	http://p	断舍离 (1101915	高更	2013年7月1日	23	32	7.2
17	http://p	少年读史	1429922	麦家	2015年9月1日	78	100	7.8
18	http://p	流浪的地	717768	新经典	2018年5月1日	11.2	29	3.9
19	http://p	活着	2980683	简·尼尔	2012年8月1日	22.1	28	7.9
20	http://p	窗边的小	1776971	张嘉佳	2018年5月1日	27.2	39.5	6.9
21	http://p	平凡的世	1449694	博集天卷	2017年6月1日	81	108	7.5
22	http://p	人间值得	751832	中村恒子	2019年9月1日	27.5	49.9	5.5

接下来对 2024 年的数据进行处理：

```
import pandas as pd
df = pd.read_excel('D:\\Eytalsixone\\wajuedata\\last\\month1-11.xlsx')
df.info()

df['价格'] = df['价格'].str.replace('¥', '')
df['原价'] = df['原价'].str.replace('¥', '')

df['评论数'] = df['评论数'].str.replace('条评论', '')
# 删除折扣列的“折”字
df['折扣'] = df['折扣'].str.replace('折', '')

# df['价格'] = df['价格'].astype(float)
# df['原价'] = df['原价'].astype(float)
df['价格'] = pd.to_numeric(df['价格'].str.replace(',', '', '. '), errors='coerce')
df['原价'] = pd.to_numeric(df['原价'].str.replace(',', '', '. '), errors='coerce')

df['折扣'] = df['折扣'].astype(float)
df['评论数'] = pd.to_numeric(df['评论数'], errors='coerce')

df['出版年份'] = pd.to_datetime(df['出版年份'])

# print('df数据初步处理后的显示')
df.describe().T
# df.to_excel('month_good_data.xlsx', index=False)
```

处理后的结果：

	A	B	C	D	E	F	G	H	I	J
1	月份	排名	链接	书名	评论数	作者	出版年份	价格	原价	折扣
2	1	1	http://product.€ 额尔古纳	1111768	迟子建		2019年6月1日	32	32	10
3	1	2	http://product.€ 我们生活	711534	余华		2015年1月1日	35	35	10
4	1	3	http://product.€ 我与地坛	1583929	史铁生		2011年6月1日	16	29	5.5
5	1	4	http://product.€ 读读童谣	559020	曹文轩		2018年1月1日	27.1	39.8	6.8
6	1	5	http://product.€ 中国古代	231528	陈先云		2019年1月1日	24.5	25	9.8
7	1	6	http://product.€ 克雷洛夫	204379	曹文轩		2019年1月1日	24.5	25	9.8
8	1	7	http://product.€ 经典常谈	128064	陈先云		2023年1月1日	20.1	26.8	7.5
9	1	8	http://product.€ 伊索寓言	187181	曹文轩		2019年1月1日	24.5	25	9.8
10	1	9	http://product.€ 活着（余	1591886	陈先云		2021年10月1日	31	45	6.9
11	1	10	http://product.€ 病隙碎笔	473076	朱自清		2021年11月1日	45.6	48	9.5
12	1	11	http://product.€ 繁花批注	53567	曹文轩		2023年6月1日	63.7	98	6.5
13	1	12	http://product.€ 俗世奇人	464966	陈先云		2018年4月1日	28	28	10
14	1	13	http://product.€ 繁花（原	126125	余华		2020年10月1日	68	68	10
15	1	14	http://product.€ 大头儿子	231606	新经典		2018年12月1日	10.9	16	6.8
16	1	15	http://product.€ 【印签版	593794	史铁生		2022年8月1日	68	68	10
17	1	16	http://product.€ 骆驼祥子	370181	博集天卷		2018年4月1日	26	26	10
18	1	17	http://product.€ 骆驼祥子	258220	金字澄		2017年6月1日	21.6	28.8	7.5
19	1	18	http://product.€ 被讨厌的	1535847	沈宏非		2020年3月5日	29.8	55	5.4
20	1	19	http://product.€ 小马过河	39530	批注		2022年1月1日	15	48	3.1
21	1	20	http://product.€ 十万个为	323942	姜庆共		2018年10月1日	19.4	22.8	8.5
22	1	21	http://product.€ 苏东坡传	1183166	林语堂		2018年1月1日	49.4	52	9.5
23	1	22	http://product.€ 鲁滨逊漂	410469	博集天卷		2012年6月1日	12.6	14.8	8.5
24	1	23	http://product.€ 汤姆索亚	540398	丹尼尔		2021年6月1日	12.6	14.8	8.5
25	1	24	http://product.€ 课文作家	105332	笛福		2019年9月1日	11.25	25	4.5
26	1	25	http://product.€ 快乐读书	95698	马克		2018年11月1日	52.9	62.2	8.5
27	1	26	http://product.€ 十万个为	82714	吐温		2020年1月1日	43.5	75	5.8
28	1	27	http://product.€ 生死疲劳	735021	鲁冰		2022年1月1日	45.4	69.9	6.5
29	1	28	http://product.€ 钢铁是怎	256323	王林		2018年6月1日	36	36	10
30	1	29	http://product.€ 长安的荔	550185	马克		2022年10月31日	29	45	6.4
31	1	30	http://product.€ 小学生漫	114273	吐温		2023年8月15日	18.8	100	1.9
32	1	31	http://product.€ 海底两万	214329	笛福		2017年6月1日	29.1	38.8	7.5

第三章 数据分析及可视化

3.1 图书评论数分析

先查看 2020-2023 年图书评论数最高的作者：

```
# In[8]:

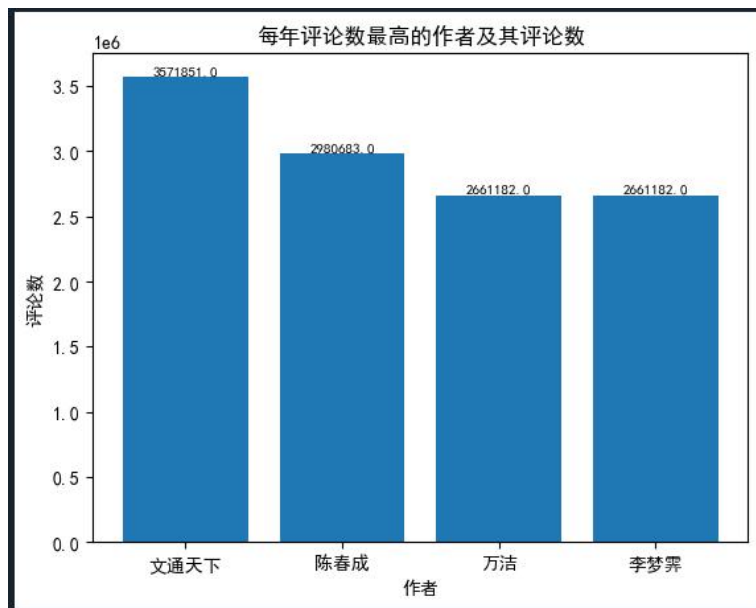
max_comments_author = df.groupby('年份')['评论数'].idxmax().apply(lambda x: df.loc[x]['作者'])

# 首先，我们按'年份'分组，并找到每个组中'评论数'的最大值的索引
# idxmax() 返回最大值的索引，这里是对每个年份的评论数进行操作的
max_comments_idx = df.groupby('年份')['评论数'].idxmax()
# 接下来，我们使用 apply 函数和 lambda 表达式来获取这些索引对应的作者的姓名
# lambda x: df.loc[x]['作者'] 对每个索引 x，使用 df.loc[x] 获取对应的行，然后取['作者']列的值
max_comments_author = max_comments_idx.apply(lambda x: df.loc[x]['作者'])
# 同时，我们也需要获取这些索引对应的评论数
max_comments_count = df.loc[max_comments_idx, '评论数']
# 打印结果
print("\n每个年份评论数最高的作者及其评论数:")
for year, (author, count) in zip(max_comments_idx.index, zip(max_comments_author, max_comments_count)):
    print(f"年份: {year}, 作者: {author}, 评论数: {count}")
```

输出结果：

```
每个年份评论数最高的作者及其评论数：
年份：2020，作者：文通天下，评论数：3571851.0
年份：2021，作者：陈春成，评论数：2980683.0
年份：2022，作者：万洁，评论数：2661182.0
年份：2023，作者：李梦霁，评论数：2661182.0
```

进行可视化分析：



每年评论数最高的作者通常意味着他们的作品受到了广泛的关注和讨论。这可以反映这些作者在读者中的影响力和受欢迎程度。高评论数也可能意味着作品具有较高的质量或吸引力，能够激发读者的讨论欲望。同时，这也可能反映了作品在市场上的接受度和受欢迎程度。在读者参与度方面，评论数的高低还可以反映读者对作品的参与度和兴趣程度。高评论数意味着更多的读者愿意分享他们的阅读体验和看法。

通过观察不同年份评论数最高的作者和作品，可以分析出市场上的流行趋势和读者偏好的变化。例如，某些类型的书籍可能在某个时间段内特别受欢迎，而另一些则可能逐渐失去市场。我们可以进一步查看不同作者所代表的流派，进而可以知道每年的图书流行趋势。

通过分析不同作者之间的评论数差异，可以了解市场上的竞争态势。评论数较高的作者可能拥有更强的市场地位和品牌影响力。

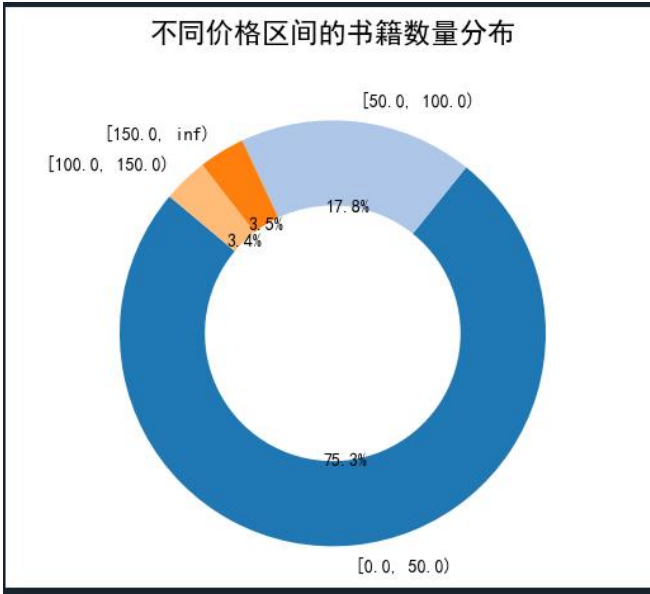
3.2 书籍价格分析

查看不同价格区间的书籍数量分布：

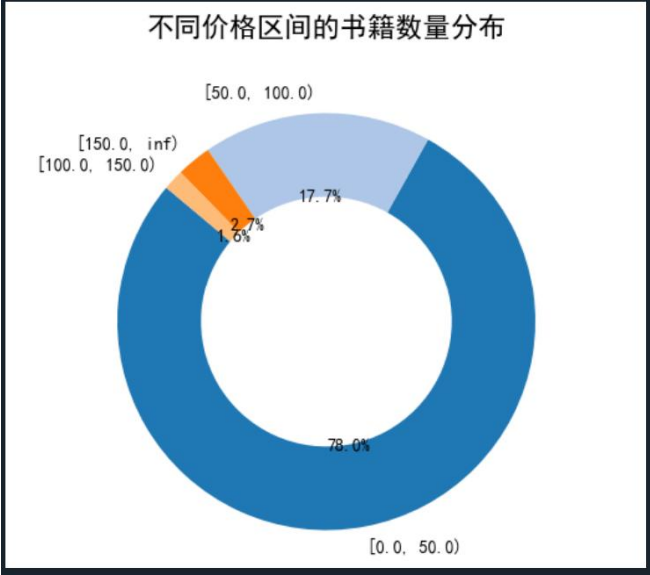
不同价格区间的书籍数量分布

```
import numpy as np
price_range = pd.cut(df['价格'], bins=[0, 50, 100, 150, np.inf], right=False)
price_counts = price_range.value_counts()
fig, ax = plt.subplots()
wedges, texts, autotexts = ax.pie(price_counts, labels=price_counts.index, autopct='%1.1f%%', startangle=140, colors=plt.cm.tab20.colors, wedgeprops=dict(
    bbox_props = dict(boxstyle="square,pad=0.3", fc="w", ec="k", lw=0.72)
    kw = dict(arrowprops=dict(arrowstyle="-"),
        bbox=bbox_props, zorder=0, va="center")

plt.axis('equal')
plt.title('不同价格区间的书籍数量分布', fontsize=16, y=1.1)
plt.show()
```



2020-2023 年不同价格区间的数量分布



2024 年 11 个月内不同价格区间的数量分布

通过分析 2020-2024 年的图书价格数据，可以估计价格区间与市场需求的关系，不同价格区间的图书数量分布可能反映了市场对不同价格段图书的需求情况。例如，0-50 元区间的图书数量占较大比重，意味着该价格段的图书更符合消费者的购买能力和购买意愿。出版商可以根据市场趋势和读者需求，制定不同的出版策略和市场定位，从而在不同价格区间推出相应数量的图书。通过分析可以得出，近五年的图书价格呈现较为稳定的分布。

3.3 价格排名靠前的书籍分布

先分析 2020-2023 年价格排名前十的书籍分布

价格排名前10的书籍

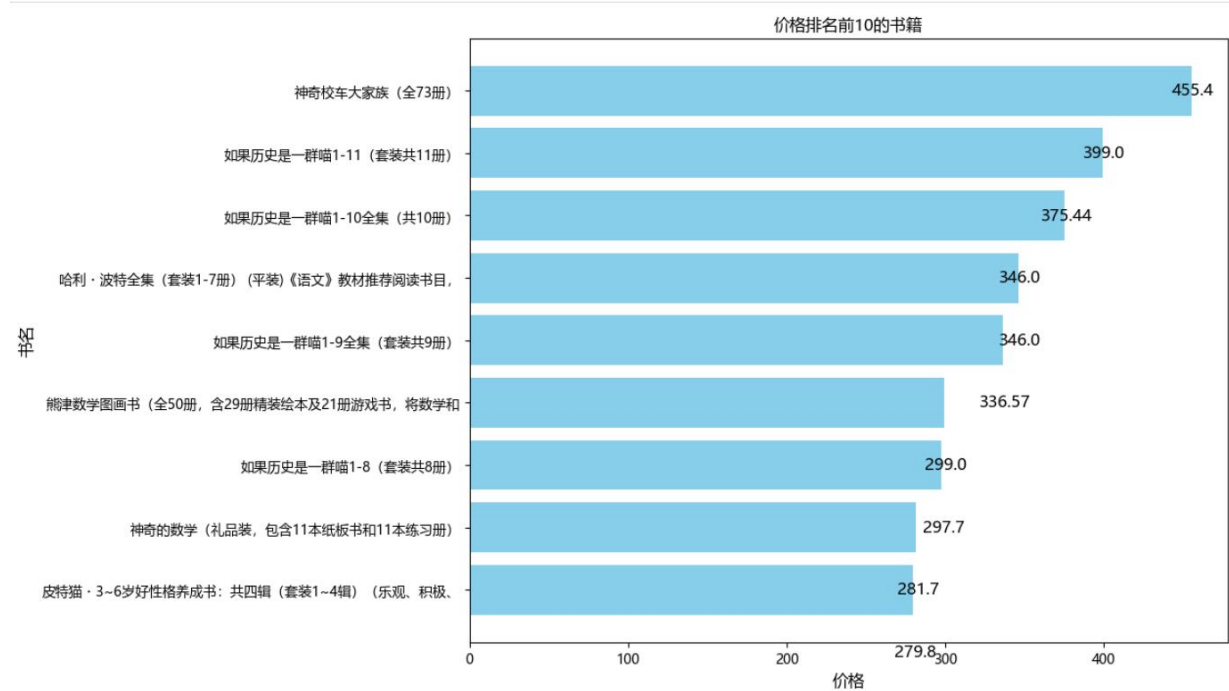
```
import matplotlib.pyplot as plt
import pandas as pd
from matplotlib.font_manager import FontProperties

import matplotlib
# 指定字体为 Microsoft YaHei, 适用于 Windows 系统
matplotlib.rcParams['font.sans-serif'] = ['Microsoft YaHei'] # 指定默认字体
matplotlib.rcParams['axes.unicode_minus'] = False # 用来正常显示负号

top_10_books = df.sort_values(by='价格', ascending=False).head(10)

# 创建 FontProperties 对象
font = FontProperties(fname=r"c:\windows\fonts\msyh.ttc", size=12) # 注意路径可能需要调整为您的系统字体路径
# 如果您的系统默认支持 Microsoft YaHei 并且已经设置了 rcParams, 这一步其实可以省略, 直接使用默认字体。

plt.figure(figsize=(10, 8))
plt.barh(top_10_books['书名'], top_10_books['价格'], color='skyblue')
plt.xlabel('价格', fontproperties=font)
plt.ylabel('书名', fontproperties=font)
plt.title('价格排名前10的书籍', fontproperties=font)
plt.gca().invert_yaxis()
for i, v in enumerate(top_10_books['价格']):
    plt.text(v + 1, i, str(v), fontproperties=font, ha='center', va='center') # 添加数据标签, 并设置字体
plt.show()
```

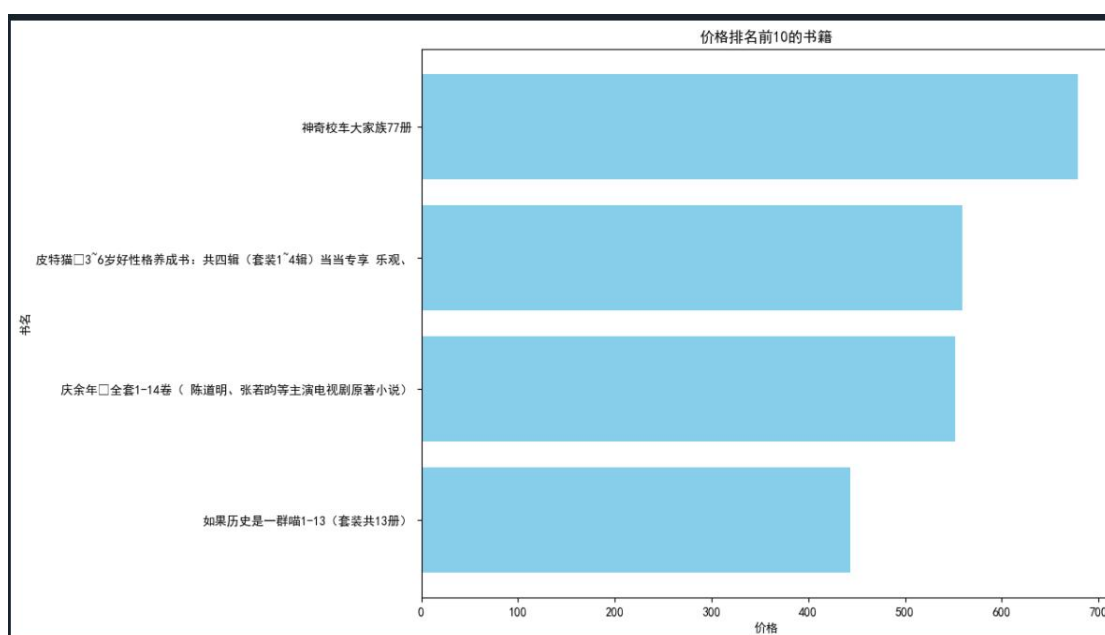


其次分析 2024 年价格排名前十位的书籍分布

```

In [27]: top_10_books['书名']
Out[27]:
488          |          神奇校车大家族77册
665    皮特猫·3~6岁好性格养成书：共四辑（套装1~4辑）当当专享 乐观、
2498    皮特猫·3~6岁好性格养成书：共四辑（套装1~4辑）当当专享 乐观、
2878    皮特猫·3~6岁好性格养成书：共四辑（套装1~4辑）当当专享 乐观、
1553    皮特猫·3~6岁好性格养成书：共四辑（套装1~4辑）当当专享 乐观、
2865    庆余年·全套1-14卷（陈道明、张若昀等主演电视剧原著小说）
2660          如果历史是一群喵1-13（套装共13册）
1085          如果历史是一群喵1-13（套装共13册）
2038          如果历史是一群喵1-13（套装共13册）
5437          如果历史是一群喵1-13（套装共13册）
Name: 书名, dtype: object

```



我们分析近五年价格较高的畅销图书，在价格与畅销度的关系方面，价格排名前十的书籍可能反映了市场对高价书籍的接受程度以及高价书籍在畅销榜上的表现。如果高价书籍频繁出现在畅销榜上，可能意味着部分读者愿意为高质量的书籍支付更高的价格。高价书籍可能面向对内容质量有更高要求的读者，而低价书籍则可能吸引更广泛的消费者群体。我们可以发现，今年的数据成套系列的图书可能售卖的更昂贵一些，这与市场环境有关，例如电视剧庆余年的播出，在社会上引起了广泛的反响，这也带动了庆余年小说图书的售卖。

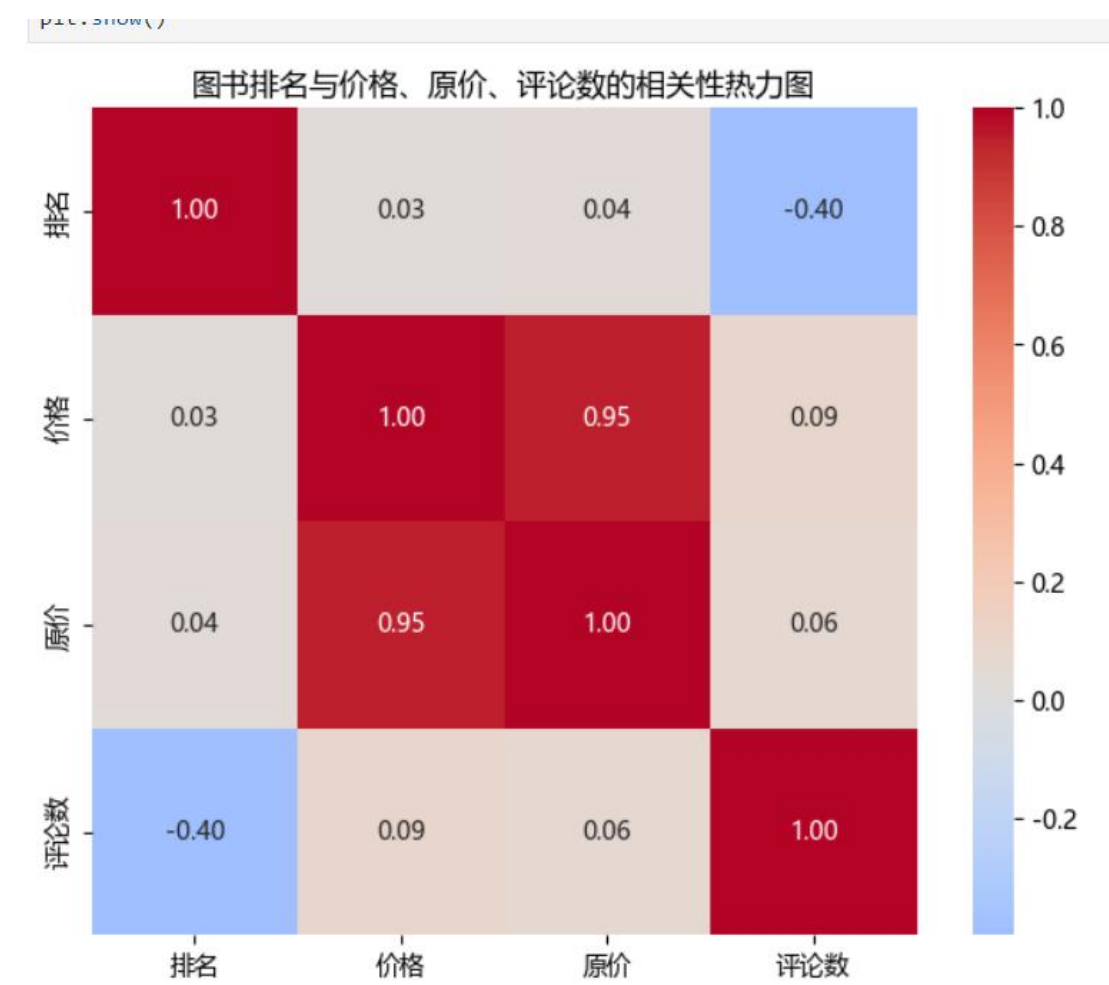
更重要的是，我们发现神奇校车大家族、皮特猫·3~6岁好性格养成书：共四辑（套装1~4辑）、如果历史是一群喵1-13（套装共13册）这三套图书是这五年内消费者购买的价格比较高的成套图书，这可能意味着这三套图书的质量更高。

3.4 相关性分析

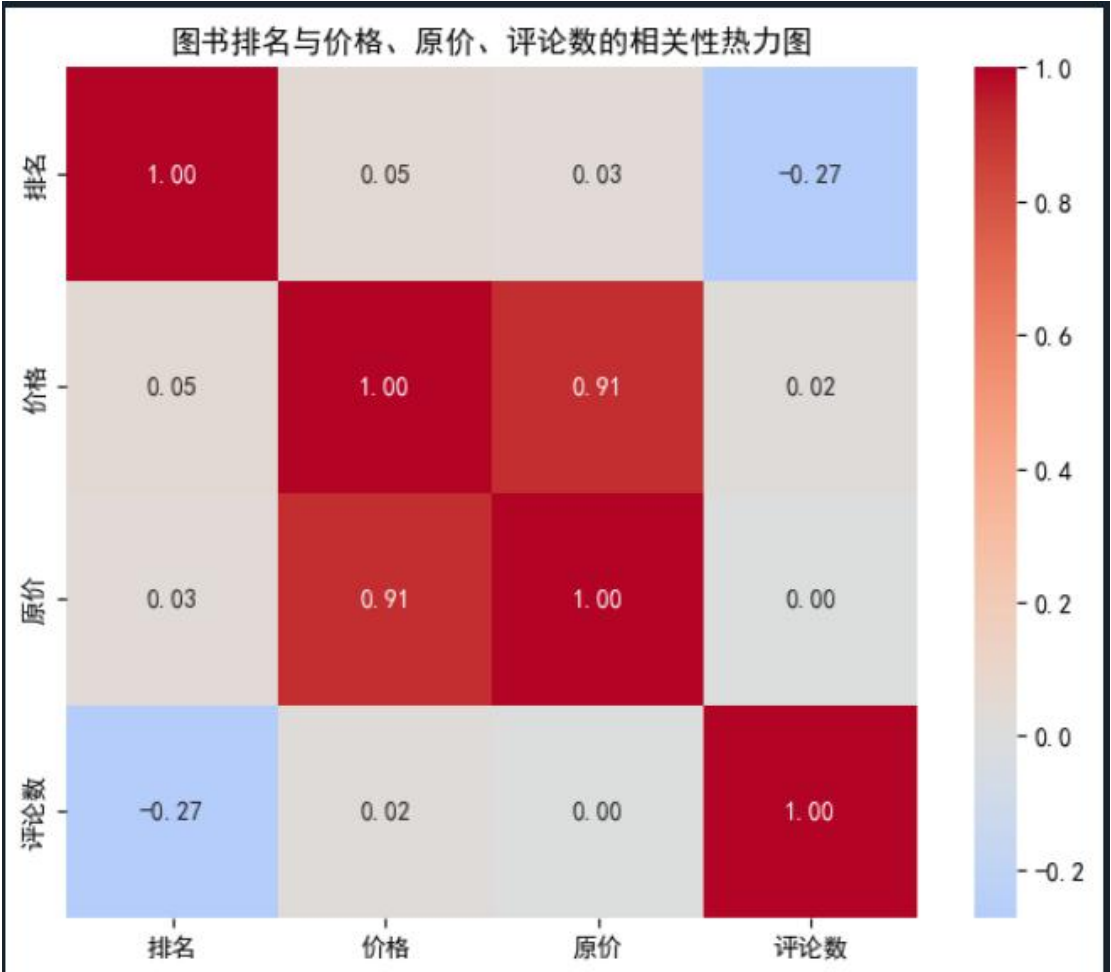
2020-2023 图书排名与价格、原价、评论数的相关性分析

相关性分析

```
book_data = df[['排名', '价格', '原价', '评论数']]
correlation_matrix = book_data.corr()
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', center=0, square=True, fmt='.2f')
plt.title('图书排名与价格、原价、评论数的相关性热力图')
plt.show()
```



2024 年图书排名与价格、原价、评论数的相关性分析



排名与价格：相关性数值为较小，表明排名与价格之间的相关性很弱，几乎可以忽略不计。这意味着图书的畅销程度（排名）与其售价之间没有直接的关联。

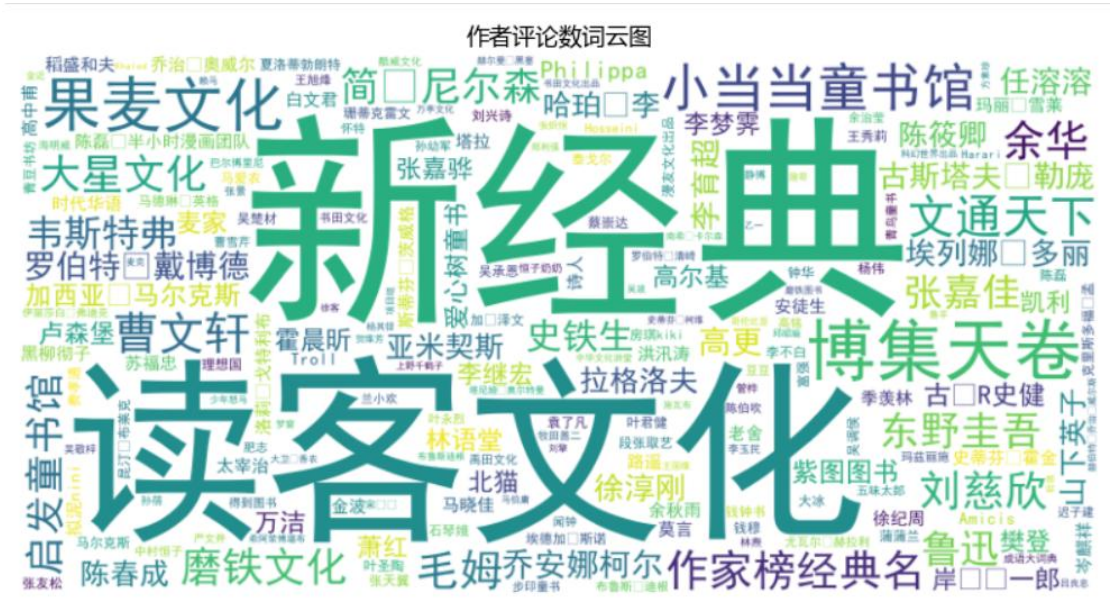
排名与评论数：相关性数值为-0.40，表明排名与评论数之间存在一定程度的负相关。即，评论数较多的图书在畅销榜上的排名可能相对较低。这可能是由于评论数多的图书往往是更老或更知名的作品，而新作品或不太知名的作品虽然评论数少，但可能因新颖性或独特性而更受欢迎，从而在畅销榜上排名更高。

3.5 评论数词云分析

查看 2020-2023 年所有作者评论数的词云图，可以看到，新经典作者和读客文化作者作品评论总数最多。

作者词云图

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt
word_frequency = df.groupby('作者')['评论数'].sum().reset_index()
word_frequency_dict = dict(zip(word_frequency['作者'], word_frequency['评论数']))
wordcloud = WordCloud(font_path='simhei.ttf', width=800, height=400, background_color='white').generate_from_frequencies(word_frequency_dict)
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('作者评论数词云图')
plt.show()
```



词云图中字体较大的作者名字通常代表了他们获得的评论数较多，这可以直观地反映出哪些作者在读者中更受欢迎。排名前两位的为新经典和读客文化，这些作者可能因其独特的写作风格、引人入胜的故事情节或深刻的主题思想而赢得了读者的青睐。词云图中出现了多个字体较大的作者名字，说明畅销榜上的作品具有多样性，不同作者的书籍都有机会获得读者的关注和喜爱。这反映了读者对图书内容的多元化需求。

词云分析可以量化地展示不同作者在读者中的影响力。通过对比

不同作者的评论数，我们可以清楚地看到哪些作者在读者中拥有更高的知名度和影响力。在市场需求方面，词云图中作者名字的大小和分布可以反映出读者对不同类型书籍的需求。如果某些特定类型的书籍获得了大量评论，那么这可能意味着这类书籍在当前市场上具有较高的需求。虽然评论数不能完全代表作品的质量，但它在一定程度上可以反映作品的受欢迎程度。因此，词云分析也可以间接地反映出哪些作品在读者中获得了较高的评价。

3.6 书籍出版时间分布

查看不同书籍的出版时间分布情况，可以看到，近四年畅销的图书大部分都是在 2017-2020 年出版的。

绘制不同年份出版图书的数据：

```
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import rcParams

# 读取数据
df = pd.read_excel('2020-2023data.xlsx')

# 数据处理（删除不需要的字符，转换数据类型）
df['价格'] = df['价格'].str.replace('¥', '').astype(float)
df['原价'] = df['原价'].str.replace('¥', '').astype(float)
df['评论数'] = df['评论数'].str.replace('条评论', '').apply(pd.to_numeric, errors='coerce')
df['折扣'] = df['折扣'].str.replace('折', '').astype(float)

if df['出版年份'].dtype == object: # 检查是否为对象类型（通常是字符串）
    df['出版年份'] = df['出版年份'].astype(str).str.extract('(\\d{4})').astype(int)
elif df['出版年份'].dtype not in [int, float]: # 如果不是整型也不是我们预设的字符串格式，则抛出警告
    raise ValueError("出版年份的格式不符合预期，请检查数据！")

# 设置字体为SimHei（确保该字体在您的系统上可用）
rcParams['font.sans-serif'] = ['SimHei'] # 指定默认字体
rcParams['axes.unicode_minus'] = False # 用来正常显示负号

# 数据统计：统计所有年份出版的书籍数量
publication_counts = df.groupby('出版年份').size()

# 设置图表风格（可选）
plt.style.use('seaborn-v0_8-darkgrid') # 使用Seaborn的darkgrid风格

# 绘制折线图
fig, ax = plt.subplots(figsize=(12, 8))
line = ax.plot(publication_counts.index, publication_counts.values, marker='o', linestyle='-', color='tab:blue', label='The number of books')

# 添加阴影效果（可选）
ax.fill_between(publication_counts.index, publication_counts.values, color='tab:blue', alpha=0.3)

# 设置图表标题和轴标签
ax.set_title('Distribution of Books by Year', fontsize=16, fontweight='bold')
ax.set_xlabel('Year of publication', fontsize=14)
ax.set_ylabel('The number of books', fontsize=14)

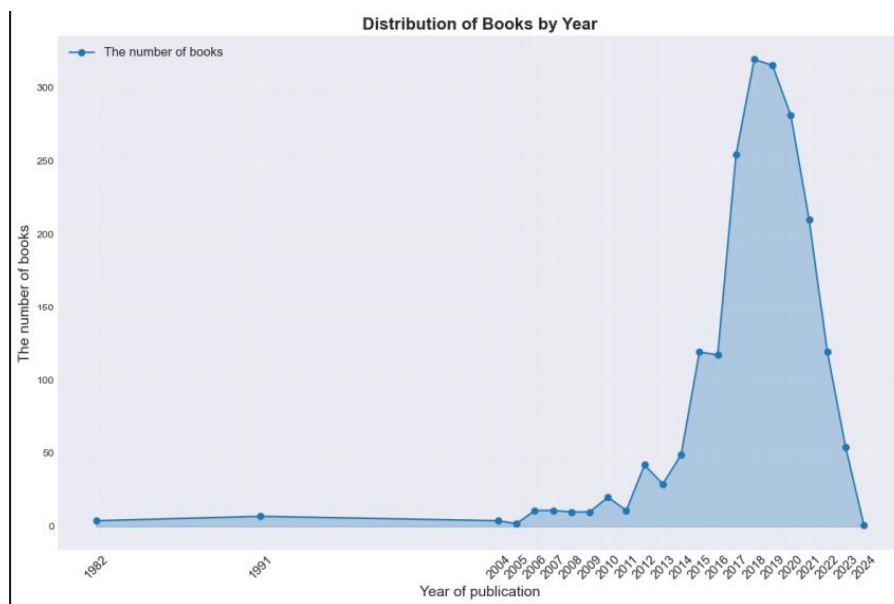
# 设置网格线（可选）
ax.grid(True, linestyle='--', linewidth=0.5, alpha=0.7)

# 设置x轴刻度标签的旋转角度和字体大小（如果年份很多，可能需要调整）
ax.set_xticks(publication_counts.index)
ax.set_xticklabels(publication_counts.index, rotation=45, fontsize=12)

# 添加图例
ax.legend(loc='upper left', fontsize=12)

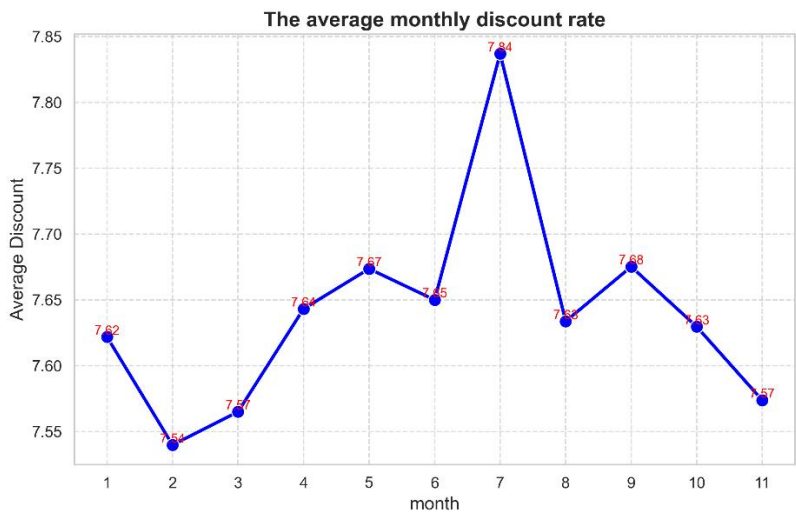
# 优化布局（可选）
plt.tight_layout()

# 显示图表
plt.show()
```



从 1948 年到 2017 年，书籍的出版数量呈现出一种增长趋势，尽管中间可能存在波动。特别是在近些年，书籍的出版数量有显著增加，这可能与出版业的繁荣、读者需求的增长以及数字化出版的兴起有关。我绘制了书籍出版年份的分布折线图。图表清晰地展示了近年来畅销书籍出版时间的趋势，包括哪些年份出版的书籍数量较多，哪些年份相对较少，有利于了解出版行业的动态，为出版社、作者及读者提供有价值的参考信息。

3.7 2024 年每月平均打折情况



本图反映了 2024 年 1 月至 11 月期间，每月图书的平均打折情况。通过折线图，我们可以观察到不同月份打折力度的变化趋势，为出版商、书店和消费者提供市场动态的参考。

1 月：平均折扣率为 7.62，为全年起始水平。

2 月：折扣率下降至全年最低点 7.55，可能与春节假期期间的销售策略有关。

3 月：折扣率略微上升至 7.57，显示出假期结束后市场的逐渐回暖。

4 月-6 月：折扣率稳步上升，6 月达到 7.65，可能与春季图书促销活动有关。

7 月：折扣率显著上升至全年最高点 7.81，这可能与暑期图书销售旺季有关，出版商和书店可能增加了折扣以吸引学生和家長。

8 月：折扣率下降至 7.63，尽管仍处于较高水平，但显示出暑期销售旺季的结束。

9 月：折扣率小幅上升至 7.68，可能与开学季的图书促销有关。

10 月-11 月：折扣率逐渐下降，11 月为 7.57，可能与年末销售策略调整有关。

2024 年图书折扣率的变化与季节性因素密切相关，尤其是暑期的销售旺季。出版商和书店可以根据这些趋势调整销售策略，以最大化销售和利润。

3.8 2024 年畅销书目分析

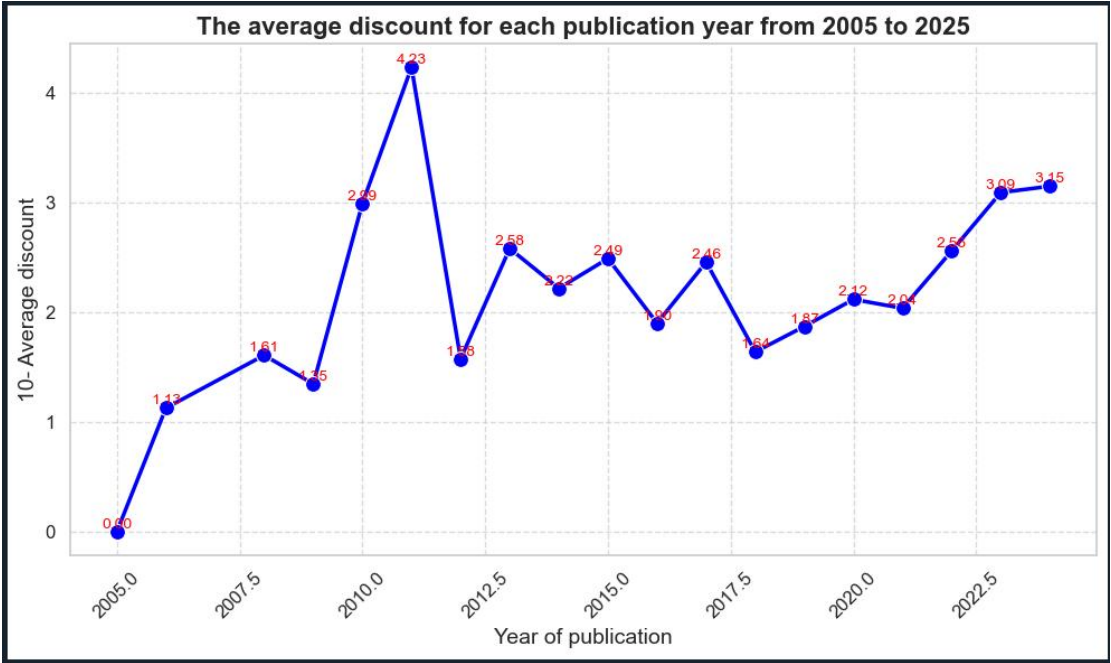


2024 年畅销书目的书名情况，通过词云图的形式，我们可以直观地看出哪些词汇在书名中出现的频率较高，从而推断出读者购买的书籍主要包含哪些主题或关键词。

从词云图中可以看出，教育类书籍，尤其是语文教材和阅读材料，在 2024 年的畅销书目中占据了显著的位置。这可能与教育政策、学校推荐读书目以及家长对孩子教育的重视有关。此外，特定出版社和版本的书籍的流行，可能与它们的教学质量、内容权威性以及市场推广策略有关。

2024 年畅销书目中教育类书籍占据了重要地位，特别是语文教材和阅读材料。这反映了教育领域对图书市场的影响，以及家长和学生的高质量教育资源的需求。

3.9 书本出版时间与打折力度的关系



本图反映了 2005 年至 2025 年间，不同出版年份的图书平均打折力度。打折力度通过“10 - 平均折扣”来表示，即折扣越高，10 - 平均折扣的值越小，反映出消费者购买时的实际优惠程度。

从整体趋势来看，打折力度在 2011 年达到最高点后，经历了一段时间的波动和下降，但在 2020 年后又开始逐渐回升。这可能与以下几个因素有关：

- 市场策略：出版商可能根据市场需求和竞争状况调整折扣策略。
- 经济环境：经济状况的变化可能影响消费者的购买力和出版商的定价策略。
- 消费者行为：消费者对折扣的敏感度可能随时间变化，影响出版商的折扣决策。

建议出版商应密切关注市场动态和消费者行为，灵活调整折扣策略以最大化销售和利润。对消费者而言，在购买图书时，可以关注出

版年份较新的图书，这些图书可能享有更高的折扣。

第四章 实验总结

通过本实验，我成功地从当当网爬取了 2020 年至 2024 年 1-11 月份的图书信息，并将其保存到 Excel 文件中，进行了部分数据可视化的分析。

第五章 实验思考

在实验过程中，我采用了以下算法和策略来爬取和处理数据：

- HTTP 请求算法：**使用 requests 库发送 HTTP GET 请求，获取网页内容。为了提高请求的成功率，我们可以设置请求头（如 User-Agent）以模拟浏览器访问；同时，我们还可以添加重试机制、超时设置等，以应对网络波动和请求失败的情况。
- HTML 解析算法：**使用 lxml 库的 etree 模块解析 HTML 文档。首先，我们需要根据网页的 DOM 结构编写 XPath 查询语句；然后，使用 etree 模块的 xpath 方法提取所需的信息。为了提高解析的准确性和效率，我们可以对 XPath 查询语句进行优化和调整。
- 数据清洗算法：**在提取的图书信息中，可能存在一些无效或重复的数据。为了得到准确的数据集，我们需要对数据进行清洗和去重。具体算法包括：去除空值、去除重复项、转换数据类型等。
- 数据分析算法：**根据爬取的图书信息，我们可以使用 pandas 库进行数据分析。具体算法包括：统计热门类别和畅销书排行、分析读者

的阅读偏好和购买行为、计算价格分布情况等。这些分析结果可以为图书出版、销售和推广提供决策支持。

第六章 实验结论

本实验通过对当当网 2020 年至 2024 年 1-11 月份的图书销售数据进行爬取、清洗和分析，揭示了图书市场的多个重要趋势和特征。通过数据可视化和统计分析，我们得出了以下主要结论：

1. 评论数与作者影响力：

2020 年至 2023 年间，评论数最高的作者通常具有较高的市场影响力和读者关注度。这些作者的作品往往能够引发广泛的讨论和分享，反映了他们在读者中的受欢迎程度。

词云分析显示，新经典和读客文化等出版社的作者作品在评论数上表现突出，这表明这些作者的作品在质量和内容上具有较高的吸引力。

2. 书籍价格分布：

2020 年至 2024 年的图书价格分布相对稳定，0-50 元区间的图书数量占较大比重，这表明该价格段的图书更符合消费者的购买能力和意愿。

价格排名前十的书籍中，成套系列图书占据了较高比例，特别是《神奇校车大家族》、《皮特猫·3~6 岁好性格养成书：共四辑（套装 1~4 辑）》和《如果历史是一群喵 1-13（套装共 13 册）》等，这可能反映了这些系列图书的质量和市場认可度较高。

3. 相关性分析：

图书排名与价格之间的相关性较弱，说明图书的畅销程度与其售价之间没有直接的关联。

排名与评论数之间存在负相关关系，即评论数较多的图书在畅销榜上的排名可能较低。这可能是由于评论数多的图书往往是较老或知名的作品，而新作品或不太知名的作品虽然评论数少，但可能因新颖性或独特性而更受欢迎。

4. 出版时间与打折力度：

2024 年 1 月至 11 月期间，每月图书的平均打折情况呈现出一定的季节性变化。暑期（7 月）是折扣率最高的时期，这可能与学生和家长的购书需求增加有关。

2005 年至 2025 年间，不同出版年份的图书平均打折力度有所波动。2011 年后，打折力度经历了一段时间的下降，但在 2020 年后又开始逐渐回升。这反映了出版商根据市场需求和经济环境调整折扣策略的趋势。

5. 畅销书目分析：

2024 年的畅销书目中，教育类书籍尤其是语文教材和阅读材料占据了显著位置。这反映了教育领域对图书市场的影响，以及家长和学生的高质量教育资源的需求。

词云图显示，教育类书籍在 2024 年的畅销书目中出现频率较高，这可能与教育政策、学校推荐阅读书目以及家长对孩子教育的重视有关。

通过本次实验，我们不仅掌握了网页爬虫技术、数据处理工具和编程能力，还对图书市场的动态和趋势有了更深入的了解。未来，我们可以进一步优化爬虫性能和准确性，并结合更多数据源进行综合分析，为图书出版、销售和推广提供更精准的决策支持。此外，我们还可以探索新的技术和方法，以更好地理解 and 预测市场变化，推动图书行业的健康发展。