豆瓣读书数据分析

book.xlsx 文件保存的是爬取某图书网站得到的图书数据，共 60671 条。

导入数据

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

plt.rc('font', **{'family':'SimHei'})

# 导入数据
df = pd.read_excel('books.xlsx')

# 删除第9列
df = df.drop('Unnamed: 9', axis=1)
```

数据清洗

```
# 对数据做清洗（缺失值与异常值）

df.describe()
df.info()
df.dtypes

"""
书名        object
作者        object
出版社      object
出版时间     object
页数        object
价格        object
ISBN       object
评分        float64
评论数量     object
dtype: object
"""
```

目前只要评分是数值型数据，我们还要将页数、价格、评论数量转换成数值型数据。

处理页数数据

前期分析

```
df['页数'].describe()

df['页数'].isnull().sum()  # 这样看不出来

len(df[df['页数']=='None'])  # 看看有多少 None 值页数信息
```

转换

```
# 定义 convert_to_int 方法处理页数数据，如果为 None 则填充 0

import re
def convert2int(x):
    if re.match('^\d+$',str(x)):
        return x
    else:
        return 0

df['页数'] = df['页数'].apply(convert2int)

# 或者用 lambda 表达式
df['页数'] = df['页数'].apply(lambda x: x if re.match('^\d+$', str(x)) else 0)
df['页数'] = df['页数'].astype(int)
```

处理价格数据

```
# 处理价格数据
df['价格'] = df['价格'].apply(lambda x: x if re.match('^[\d\.]+$', str(x)) else 0)
df['价格'] = df['价格'].astype(float)

# 价格为 0 的图书数量
len(df[df['价格'] == 0])
```

处理评论数量数据

```
# 处理评论数量数据

df['评论数量'] = df['评论数量'].apply(lambda x: x if re.match('^\d+$', str(x)) else 0)
df['评论数量'] = df['评论数量'].astype(int)
```

处理完之后，此时 `df.dtypes` 如下：

```
书名        object
作者        object
出版社       object
出版时间     object
页数        int64
价格        float64
ISBN       object
评分        float64
评论数量     int64
```

随机抽取一些数据看看，`df.sample(10)`：

	书名	作者	出版社	出版时间	页数	价格	ISBN	评分	评论数量
55049	登天的感觉	岳晓东	上海人民出版社	2004/11	239	16.0	9787208048942	7.8	1456
58492	我们不是天使	亦舒	海天出版社	None	198	9.8	9787806154304	7.9	3739
40160	仿生人会梦见电子羊吗?	(美)菲利普·迪克	译林出版社	2013/9	200	28.0	9787544738767	8.7	1570
15671	金融的逻辑	陈志武	西北大学出版社	2015/2	384	58.0	9787560435305	8.1	178
39485	血之魔法	特萨·格拉顿	黄山书社	2011/10	291	29.0	9787546122564	7.0	13
56408	召唤神话的44型	莱昂纳德·科恩	联邦走马	2016/11/25	85	14.9	9785425853110	0.0	0
14655	西线1944.6-1945.4	彭志文	吉林文史出版社	2016/1/1	276	59.8	9787547222799	0.0	0
58858	药堂杂文	周作人	河北教育出版社	2002/01	166	8.1	9787543443907	8.6	64
18147	福布斯说资本主义真相	史蒂夫·福布斯	中华工商联合出版社有限责任公司	2011/4/1	346	49.8	9787802495791	7.4	24
2758	未来都市NO.6 #07	[作者] 浅野敦子	皇冠文化出版有限公司	2010/7/6	208	199.0	9789573326830	9.2	290

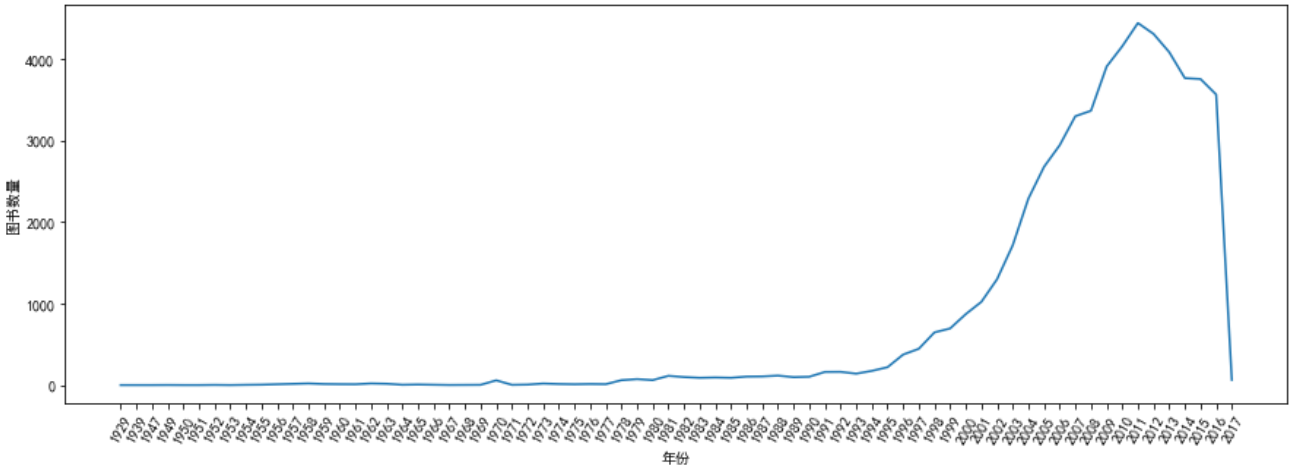
数据分析

后面需要用到年份信息，这里先对年份信息进行加工。

```
# 处理出版时间，只要年份

def year(s):
    y = re.findall('\d{4}',str(s))
    if len(y)>0:
        return y[0]
    return ''

df['出版年份'] = df['出版时间'].apply(year)

# 看看还有多少没有年份信息的
len(df[df['出版年份'] == ''])
```

分析图书数量与年份的关系

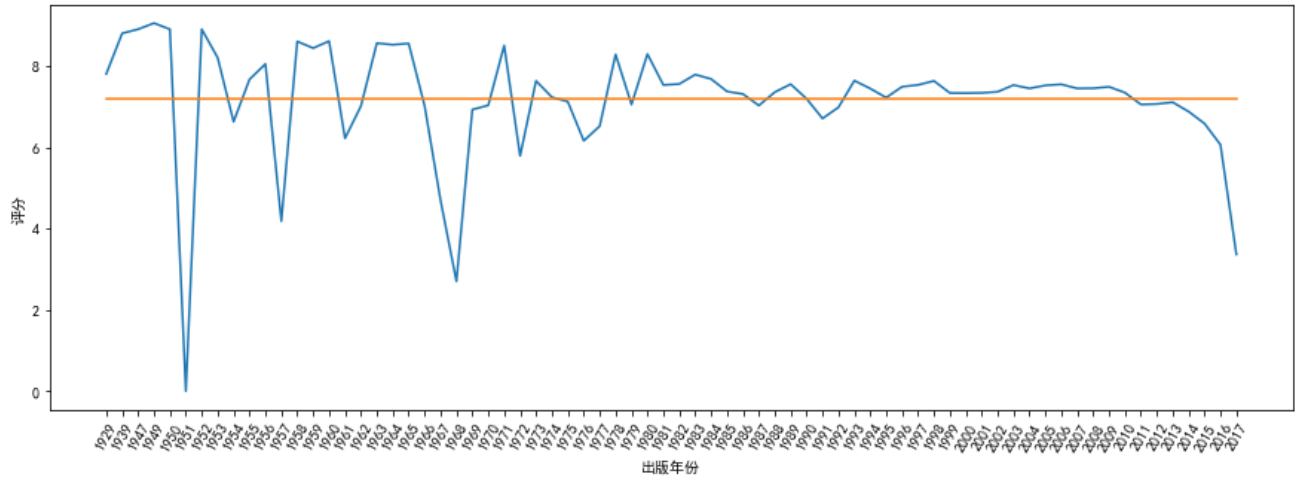
```
# 按出版年份进行分组
grouped = df.groupby('出版年份')

data = grouped['ISBN'].count()


# 有两条数据比较奇怪，处理一下
df[df['出版年份'] == '１９７９']
df.loc[df.index[60632], ['书名', '出版时间', '出版年份']]
"""
书名       鲁迅作品中的绍兴方言注释
出版时间         １９７９/初版印
出版年份             １９７９
Name: 60632, dtype: object
"""
df.loc[df.index[60632], ['出版年份']] = '1979'

df[df['出版年份'] == '２００２']
df.loc[df.index[4544], ['书名', '出版时间', '出版年份']]
"""
书名       俄罗斯插画作品集
出版时间       ２００２/２
出版年份         ２００２
Name: 4544, dtype: object
"""
df.loc[df.index[4544], ['出版年份']] = '2002'


# 然后按”出版年份“进行分组
grouped = df.groupby('出版年份')
data = grouped['ISBN'].count()
data
# 判断前7条数据和后4条数据属于异常数据，所以删除前7后4的数据
data2 = data[7:-4]

# 准备画图，设置宽一点
plt.figure(figsize=(15, 5))
# 设置 x 周标签的倾斜角度
plt.xticks(rotation=60)

plt.xlabel('年份')
plt.ylabel('图书数量')

plt.plot(data2.index, data2.values)
plt.show()
```



分析图书评分与年份的关系

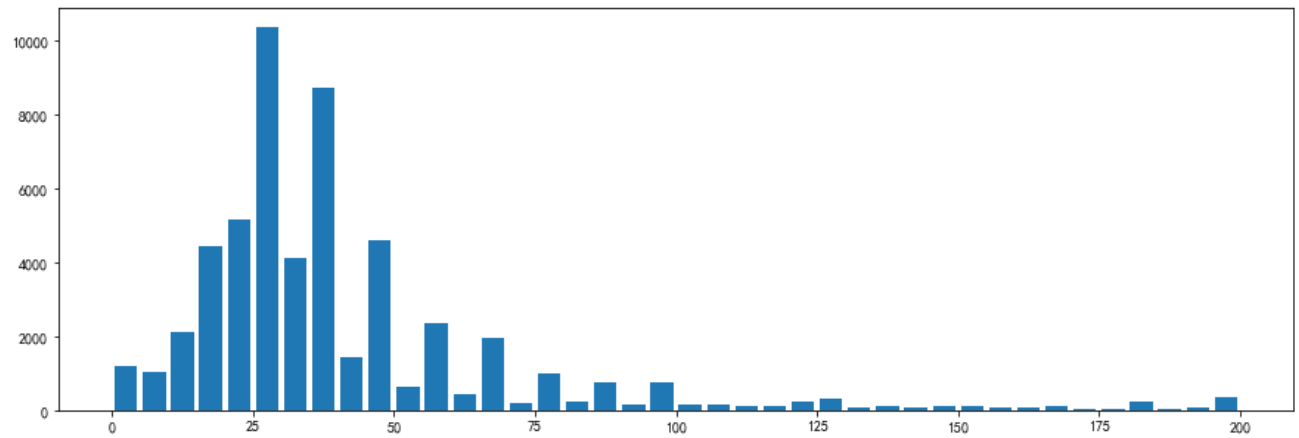
分析书籍的评分与年代之间是否有某种关系?

```
data3 = grouped['评分'].mean()
data3 = data3[7:-4]

# 折线图反映年份和评分之间的关系

# 设置宽一点
plt.figure(figsize=(15, 5))
# 设置 x 周标签的倾斜角度
plt.xticks(rotation=60)
plt.xlabel('出版年份')
plt.ylabel('评分')

plt.plot(data3.index, data3.values)

# 还要画均值线
m = data3.values.mean()
plt.plot(data3.index, [m]*len(data3.index))

plt.show()
```



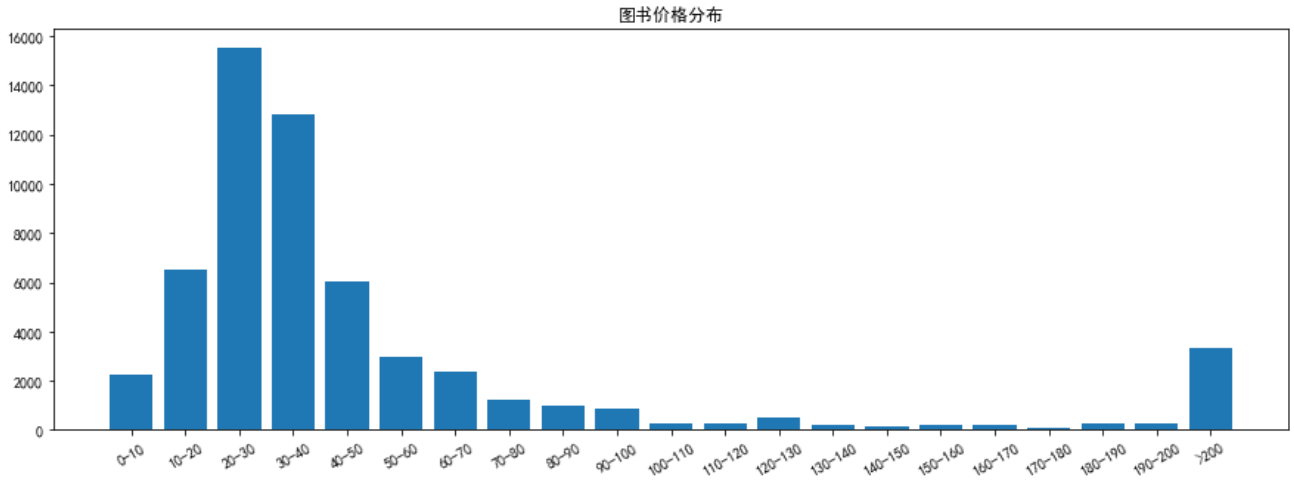
分析图书价格分布情况

```
df2 = df[df['价格'] > 0]

# 看看有多少价格大于0的
len(df2)

df2['价格'].describe()

# 直方图显示图书价格分布情况
plt.figure(figsize=(15, 5))
plt.hist(df2['价格'], bins=40, range=(0, 200), rwidth=0.8)
plt.show()
```



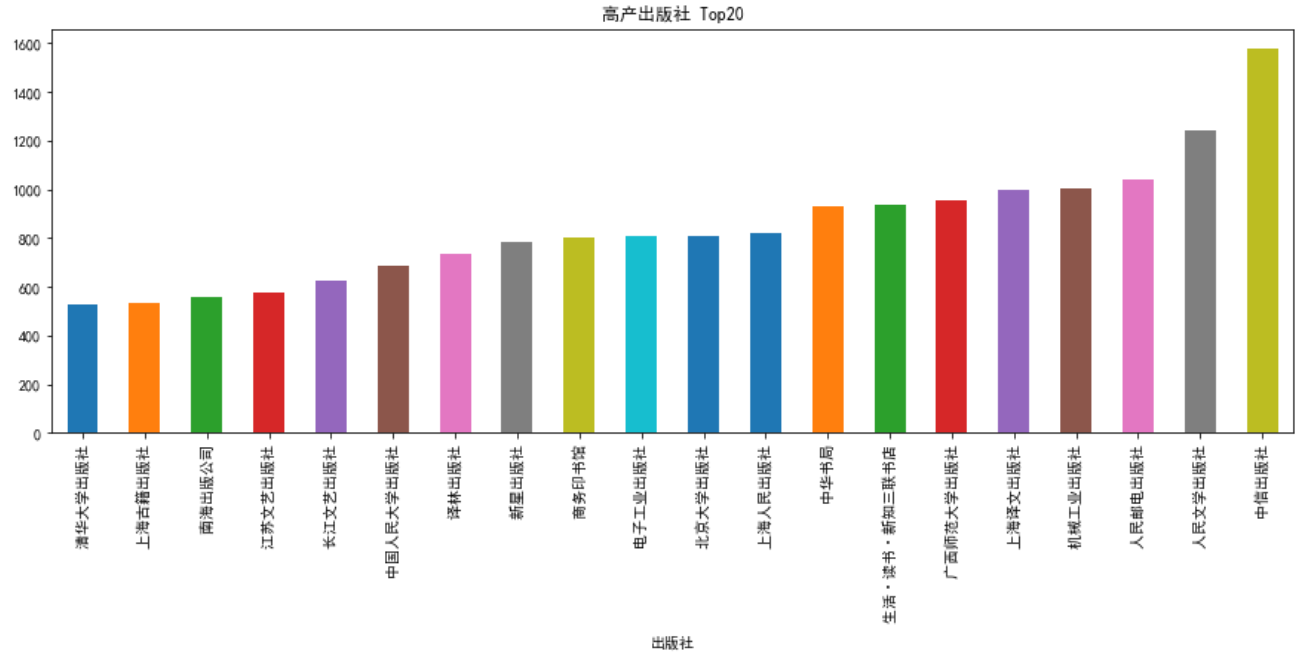
```
step = 10   # 步长
count = 20  # 柱形数量
x = []
y = []
for i in range(count):
    y.append(df2[(df2['价格']>=i*step) & (df2['价格']<i*step+step)].shape[0])
y.append(df2[df2['价格']>=count*step].shape[0])
```

```
for i in range(count):
    x.append(str(i*step)+'-'+str(i*step+step))
x.append('>'+str(count*step))

# 柱形图显示图书价格分布情况
plt.figure(figsize=(15, 5))
plt.xticks(rotation=30)
plt.title('图书价格分布')
plt.bar(x, y)
plt.show()
```



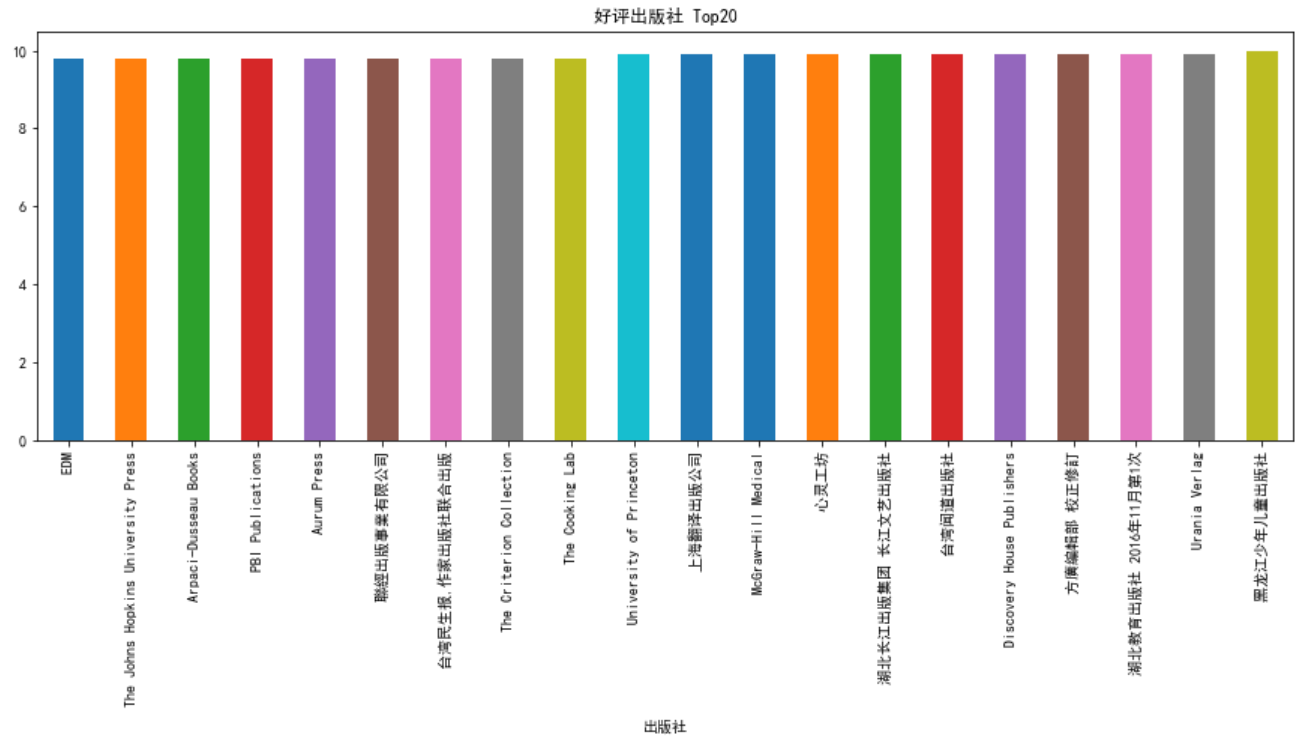
出版图书最多的前20个出版社

```
# 出版书籍最多的20个出版社

data4 = df.groupby('出版社')['ISBN'].count()

plt.figure(figsize=(15, 5))
plt.title('高产出版社 Top20')

# 最多的是 None，要去掉，所以选择 -21:-1
data4.sort_values()[-21:-1].plot(kind='bar')
plt.show()
```



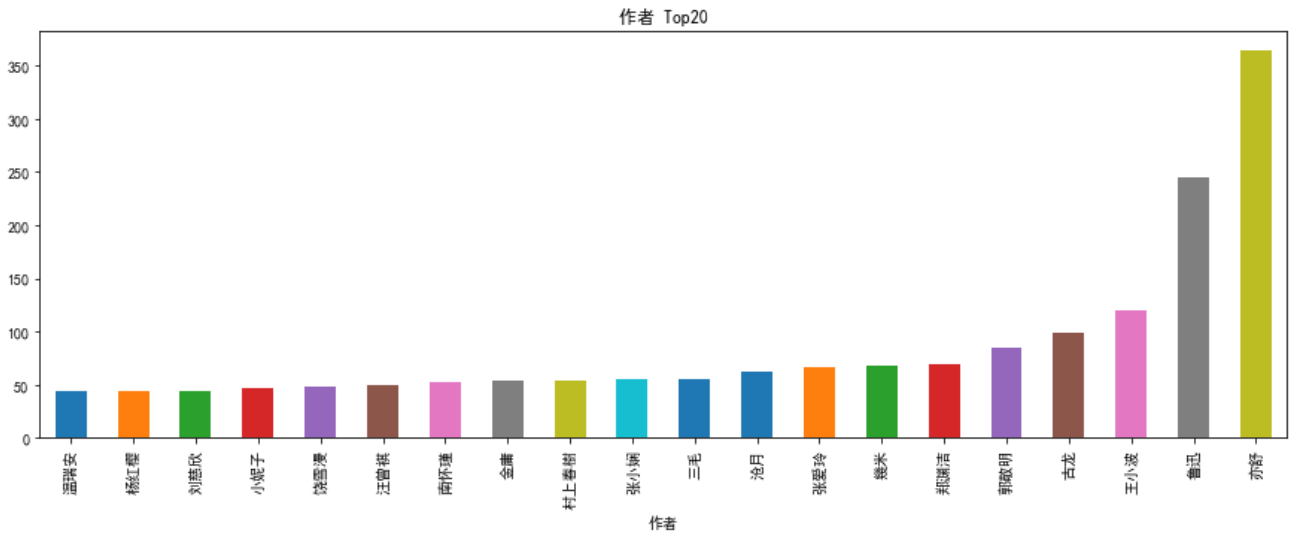
图书评分较高的出版社

```
# 评分较高的出版社

plt.figure(figsize=(15, 5))
plt.title('好评出版社 Top20')

data5 = df.groupby('出版社')['评分'].mean()
data5.sort_values()[-20:].plot(kind='bar')
plt.show()
```



出书较多的作者

```
# 出书较多的作者

plt.figure(figsize=(15, 5))
plt.title('作者 Top20')

data6 = df.groupby('作者')['ISBN'].count()
data6.sort_values()[-21:-1].plot(kind='bar')
plt.show()
```



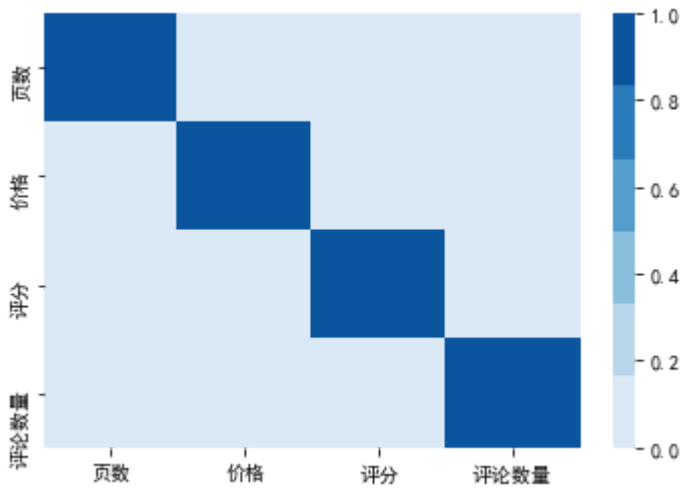
分析评分与评论数量的关系

评分高低与评论数量之间是否存在某种关系?

```
df.corr()
```

	页数	价格	评分	评论数量
页数	1.000000	-0.000030	0.003157	-0.000658
价格	-0.000030	1.000000	0.001443	-0.001673
评分	0.003157	0.001443	1.000000	0.063536
评论数量	-0.000658	-0.001673	0.063536	1.000000

```
# 评分高低与评论数量之间是否存在某种关系

import seaborn as sns

# 计算相关性矩阵
corr = df.corr()
sns.heatmap(corr, cmap=sns.color_palette('Blues'))
plt.show()
```



所以，评分高低与评论数量之间没有明显关系。