google_store_app_analysis

2020 年 6 月 11 日

谷歌应用商店的 APP 分析

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

[2]:
```
# 加载文件
# 这次分析'App', 'Category', 'Rating', 'Reviews', 'Size', 'Installs', 'Type'
df = pd.read_csv('./googleplaystore.csv', usecols=(0, 1, 2, 3, 4, 5, 6))
```

[3]:
```
# 简单浏览下数据
df.head()
```

[3]:

	App	Category	Rating	Reviews	Size	Installs	Type
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free
2	U Launcher Lite – FREE Live Cool Themes, Hide …	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free

[4]:
```
df.info()  # 查看数据的行列数量以及数据类型
```

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

```
RangeIndex: 10841 entries, 0 to 10840
Data columns (total 7 columns):
App         10841 non-null object
Category    10841 non-null object
Rating      9367 non-null float64
Reviews     10841 non-null object
Size        10841 non-null object
Installs    10841 non-null object
Type        10840 non-null object
dtypes: float64(1), object(6)
memory usage: 593.0+ KB
```

[13]:
```
df.count()  # 查看各个列的非空数据量
```

[13]:
```
App         10841
Category    10841
Rating       9367
Reviews     10841
Size        10841
Installs    10841
Type        10840
dtype: int64
```

0.1 有很多缺失值，需要清洗

[15]:
```
# App 处理
# 查看有没有重复值
df['App'].value_counts(dropna=False)
# 也可以: pd.unique(df['App']).size 去重后的统计值
```

[15]:
```
ROBLOX                                              9
CBS Sports App - Scores, News, Stats & Watch Live   8
ESPN                                                7
Candy Crush Saga                                    7
Duolingo: Learn Languages Free                      7
                                                   ..
Resume PDF Maker / CV Builder                       1
```

```
Lakeside AG Moultrie                                1
2GIS: directory & navigator                         1
EO SA Benefits                                      1
BOO! - Next Generation Messenger                    1
Name: App, Length: 9660, dtype: int64
```

0.2 有重复值，先不着急删除重复值，为了不把其他列的异常值留下，先处理数值异常的列

[16]:
```
# Category 处理
df['Category'].value_counts(dropna=False)
```

[16]:
```
FAMILY                 1972
GAME                   1144
TOOLS                   843
MEDICAL                 463
BUSINESS                460
PRODUCTIVITY            424
PERSONALIZATION         392
COMMUNICATION           387
SPORTS                  384
LIFESTYLE               382
FINANCE                 366
HEALTH_AND_FITNESS      341
PHOTOGRAPHY             335
SOCIAL                  295
NEWS_AND_MAGAZINES      283
SHOPPING                260
TRAVEL_AND_LOCAL        258
DATING                  234
BOOKS_AND_REFERENCE     231
VIDEO_PLAYERS           175
EDUCATION               156
ENTERTAINMENT           149
MAPS_AND_NAVIGATION     137
FOOD_AND_DRINK          127
```

```
HOUSE_AND_HOME           88
AUTO_AND_VEHICLES        85
LIBRARIES_AND_DEMO       85
WEATHER                  82
ART_AND_DESIGN           65
EVENTS                   64
PARENTING                60
COMICS                   60
BEAUTY                   53
1.9                       1
Name: Category, dtype: int64
```

[17]:
```
# 有一条异常值
df[df['Category'] == '1.9']
```

[17]:
```
                                          App Category  Rating Reviews  \
10472  Life Made WI-Fi Touchscreen Photo Frame      1.9    19.0    3.0M

         Size Installs Type
10472  1,000+     Free    0
```

[18]:
```
# Rating 处理
df['Rating'].value_counts(dropna=False)
```

[18]:
```
NaN     1474
4.4     1109
4.3     1076
4.5     1038
4.2      952
4.6      823
4.1      708
4.0      568
4.7      499
3.9      386
3.8      303
5.0      274
3.7      239
```

```
4.8      234
3.6      174
3.5      163
3.4      128
3.3      102
4.9       87
3.0       83
3.1       69
3.2       64
2.9       45
2.8       42
2.6       25
2.7       25
2.5       21
2.3       20
2.4       19
1.0       16
2.2       14
1.9       13
2.0       12
1.8        8
1.7        8
2.1        8
1.6        4
1.5        3
1.4        3
1.2        1
19.0       1
Name: Rating, dtype: int64
```

[19]:
```
# 用平均值填充
df['Rating'].fillna(value=df['Rating'].mean(), inplace=True)
```

[22]:
```
# Rating 处理后的结果
df['Rating'].value_counts(dropna=False)
```

[22]:

4.193338	1474
4.400000	1109
4.300000	1076
4.500000	1038
4.200000	952
4.600000	823
4.100000	708
4.000000	568
4.700000	499
3.900000	386
3.800000	303
5.000000	274
3.700000	239
4.800000	234
3.600000	174
3.500000	163
3.400000	128
3.300000	102
4.900000	87
3.000000	83
3.100000	69
3.200000	64
2.900000	45
2.800000	42
2.700000	25
2.600000	25
2.500000	21
2.300000	20
2.400000	19
1.000000	16
2.200000	14
1.900000	13
2.000000	12
2.100000	8
1.800000	8
1.700000	8

```
1.600000        4
1.400000        3
1.500000        3
1.200000        1
19.000000       1
Name: Rating, dtype: int64
```

[24]:
```
df[df['Rating']==19]
```

[24]:
```
                                             App Category  Rating Reviews  \
10472  Life Made WI-Fi Touchscreen Photo Frame      1.9    19.0    3.0M

         Size Installs Type
10472  1,000+     Free    0
```

0.2.1 有一条值是 19 的异常记录，和 Category 的异常是同一条记录

[25]:
```
# Reviews 清洗
# 用 value_counts 看数据分布挺广，看起来都是数字
df['Reviews'].value_counts(dropna=False)
```

[25]:
```
0          596
1          272
2          214
3          175
4          137
          ...
1604146      1
8116142      1
3069         1
2894         1
1340         1
Name: Reviews, Length: 6002, dtype: int64
```

[26]:
```
df['Reviews'].str.isnumeric().sum()
```

[26]:
```
10840
```

[27]:
```
# 查看有问题的那一行数据
df[~df['Reviews'].str.isnumeric()]
```

[27]:
```
                                             App Category  Rating Reviews  \
10472  Life Made WI-Fi Touchscreen Photo Frame      1.9    19.0    3.0M

         Size Installs Type
10472  1,000+     Free    0
```

[28]:
```
# 异常值和其他的一样，删除这条记录
df.drop(index=10472, inplace=True)
```

[29]:
```
# 转换数据类型
df['Reviews'] = df['Reviews'].astype('i8')
```

[30]:
```
# Size 的清洗处理
df['Size'].value_counts()
```

[30]:
```
Varies with device    1695
11M                    198
12M                    196
14M                    194
13M                    191
                      ...
720k                     1
82k                      1
549k                     1
208k                     1
902k                     1
Name: Size, Length: 461, dtype: int64
```

[31]:
```
df['Size'] = df['Size'].str.replace('M', 'e+6')
```

[32]:
```
df['Size'] = df['Size'].str.replace('k', 'e+3')
```

[35]:
```
# df['Size'].astype('f8')  # 尝试转换，此时转换报错，还有字符串
```

[36]:
```
# 定义一个字符串判断是否可以转换
def is_convertable(v):
    try:
        float(v)
        return True
    except ValueError:
        return False
```

[37]:
```
# 查看不能转换的字符串分布
temp = df['Size'].apply(is_convertable)
```

[40]:
```
temp
```

[40]:
```
0         True
1         True
2         True
3         True
4         True
         …
10836     True
10837     True
10838     True
10839    False
10840     True
Name: Size, Length: 10840, dtype: bool
```

[44]:
```
df['Size'][~temp].value_counts()
```

[44]:
```
Varies with device    1695
Name: Size, dtype: int64
```

[45]:
```
# 转换剩下的字符串
df['Size'] = df['Size'].str.replace('Varies with device', '0')
```

[46]:
```
# 再看下是不是还有没转换的字符串
temp = df['Size'].apply(is_convertable)
df['Size'][~temp].value_counts()
```

[46]:
```
Series([], Name: Size, dtype: int64)
```

[47]:
```
# 转换类型
# e+5 这种格式使用 astype 直接转为 int 有问题，如果想转成 int，可以先转成 f8，再转 i8
# df['Size'] = df['Size'].astype('f8').astype('i8')
df['Size'] = df['Size'].astype('f8')
```

[48]:
```
# 将 Size 为 0 的填充为平均数
df['Size'].replace(0, df['Size'].mean(), inplace=True)
df.describe()
```

[48]:

	Rating	Reviews	Size
count	10840.000000	1.084000e+04	1.084000e+04
mean	4.191972	4.441529e+05	2.099045e+07
std	0.478907	2.927761e+06	2.078345e+07
min	1.000000	0.000000e+00	8.500000e+03
25%	4.100000	3.800000e+01	5.900000e+06
50%	4.200000	2.094000e+03	1.800000e+07
75%	4.500000	5.477550e+04	2.600000e+07
max	5.000000	7.815831e+07	1.000000e+08

[49]:
```
# Installs 数据清洗
# 先查看分布
df['Installs'].value_counts()
```

[49]:
```
1,000,000+        1579
10,000,000+       1252
100,000+          1169
10,000+           1054
1,000+             907
5,000,000+         752
100+               719
500,000+           539
50,000+            479
5,000+             477
100,000,000+       409
10+                386
```

```
500+                 330
50,000,000+          289
50+                  205
5+                    82
500,000,000+          72
1+                    67
1,000,000,000+        58
0+                    14
0                      1
Name: Installs, dtype: int64
```

[50]:
```
# 分布比较少，直接替换
df['Installs'] = df['Installs'].str.replace('+', '')
df['Installs'] = df['Installs'].str.replace(',', '')
```

[51]:
```
# 转换
df['Installs'] = df['Installs'].astype('i8')
df.describe()
```

[51]:

	Rating	Reviews	Size	Installs
count	10840.000000	1.084000e+04	1.084000e+04	1.084000e+04
mean	4.191972	4.441529e+05	2.099045e+07	1.546434e+07
std	0.478907	2.927761e+06	2.078345e+07	8.502936e+07
min	1.000000	0.000000e+00	8.500000e+03	0.000000e+00
25%	4.100000	3.800000e+01	5.900000e+06	1.000000e+03
50%	4.200000	2.094000e+03	1.800000e+07	1.000000e+05
75%	4.500000	5.477550e+04	2.600000e+07	5.000000e+06
max	5.000000	7.815831e+07	1.000000e+08	1.000000e+09

[52]:
```
# Type 处理
# df.info() 查看到有 na 值，这里需要 dropna 参数
df['Type'].value_counts(dropna=False)
```

[52]:
```
Free    10039
Paid      800
NaN         1
Name: Type, dtype: int64
```

[54]:
```
df[df['Type'].isnull()]
```

[54]:

	App	Category	Rating	Reviews	Size	Installs	Type
9148	Command & Conquer: Rivals	FAMILY	4.193338	0	1.815209e+07	0	NaN

[56]:
```
# 删除这条数据
df.drop(index=9148, inplace=True)
```

[58]:
```
# 删除 App 重复的行
df.drop_duplicates('App', inplace=True)
```

0.2.2 数据清洗完毕，开始分析

[61]:
```
# 查看基本统计值
df.describe()
```

[61]:

	Rating	Reviews	Size	Installs
count	9658.000000	9.658000e+03	9.658000e+03	9.658000e+03
mean	4.176285	2.166150e+05	2.011053e+07	7.778312e+06
std	0.494391	1.831413e+06	2.040865e+07	5.376100e+07
min	1.000000	0.000000e+00	8.500000e+03	0.000000e+00
25%	4.000000	2.500000e+01	5.300000e+06	1.000000e+03
50%	4.200000	9.670000e+02	1.600000e+07	1.000000e+05
75%	4.500000	2.940800e+04	2.500000e+07	1.000000e+06
max	5.000000	7.815831e+07	1.000000e+08	1.000000e+09

[62]:
```
# 分类的个数
# 也可以 df['Category'].unique().size
df.Category.unique().size
```

[62]: 33

0.2.3 根据每个分类的 App 数量，排序，可以得出哪些分类的 app 最受开发者欢迎，前三是 FAMILY,GAME,TOOLS

[63]:
```
df.groupby('Category').count().sort_values('App', ascending=False)
```

[63]:

Category	App	Rating	Reviews	Size	Installs	Type
FAMILY	1831	1831	1831	1831	1831	1831
GAME	959	959	959	959	959	959
TOOLS	827	827	827	827	827	827
BUSINESS	420	420	420	420	420	420
MEDICAL	395	395	395	395	395	395
PERSONALIZATION	376	376	376	376	376	376
PRODUCTIVITY	374	374	374	374	374	374
LIFESTYLE	369	369	369	369	369	369
FINANCE	345	345	345	345	345	345
SPORTS	325	325	325	325	325	325
COMMUNICATION	315	315	315	315	315	315
HEALTH_AND_FITNESS	288	288	288	288	288	288
PHOTOGRAPHY	281	281	281	281	281	281
NEWS_AND_MAGAZINES	254	254	254	254	254	254
SOCIAL	239	239	239	239	239	239
BOOKS_AND_REFERENCE	222	222	222	222	222	222
TRAVEL_AND_LOCAL	219	219	219	219	219	219
SHOPPING	202	202	202	202	202	202
DATING	171	171	171	171	171	171
VIDEO_PLAYERS	163	163	163	163	163	163
MAPS_AND_NAVIGATION	131	131	131	131	131	131
EDUCATION	119	119	119	119	119	119
FOOD_AND_DRINK	112	112	112	112	112	112
ENTERTAINMENT	102	102	102	102	102	102
AUTO_AND_VEHICLES	85	85	85	85	85	85
LIBRARIES_AND_DEMO	84	84	84	84	84	84
WEATHER	79	79	79	79	79	79
HOUSE_AND_HOME	74	74	74	74	74	74
EVENTS	64	64	64	64	64	64
ART_AND_DESIGN	64	64	64	64	64	64

PARENTING	60	60	60	60	60	60
COMICS	56	56	56	56	56	56
BEAUTY	53	53	53	53	53	53

0.2.4 分类的安装量排序：娱乐社交类最被用户所需要

[64]:
```
df.groupby('Category').mean().sort_values('Installs', ascending=False)
```

[64]:

Category	Rating	Reviews	Size	Installs
COMMUNICATION	4.134943	907337.676190	1.289365e+07	3.504215e+07
VIDEO_PLAYERS	4.058283	414015.754601	1.631384e+07	2.409143e+07
SOCIAL	4.239164	953672.807531	1.643765e+07	2.296179e+07
ENTERTAINMENT	4.135294	340810.294118	2.122137e+07	2.072216e+07
PHOTOGRAPHY	4.159716	374915.551601	1.618812e+07	1.654501e+07
PRODUCTIVITY	4.185331	148638.098930	1.363180e+07	1.548955e+07
GAME	4.244720	648903.763295	3.973997e+07	1.447229e+07
TRAVEL_AND_LOCAL	4.087611	122464.570776	2.293315e+07	1.321866e+07
TOOLS	4.059823	277335.644498	9.870441e+06	9.675661e+06
NEWS_AND_MAGAZINES	4.135697	91063.889764	1.365578e+07	9.327629e+06
BOOKS_AND_REFERENCE	4.308770	75321.234234	1.376752e+07	7.504367e+06
SHOPPING	4.226007	220553.118812	1.593927e+07	6.932420e+06
WEATHER	4.238650	155634.987342	1.427317e+07	4.570893e+06
PERSONALIZATION	4.303405	142401.808511	1.168523e+07	4.075784e+06
HEALTH_AND_FITNESS	4.235441	74171.371528	2.018017e+07	3.972300e+06
MAPS_AND_NAVIGATION	4.052011	135337.007634	1.669496e+07	3.841846e+06
SPORTS	4.211591	108765.578462	2.333144e+07	3.373768e+06
EDUCATION	4.362969	112303.764706	1.882895e+07	2.965983e+06
FAMILY	4.181330	78550.239214	2.666982e+07	2.418319e+06
FOOD_AND_DRINK	4.175715	56473.464286	1.999241e+07	1.891060e+06
ART_AND_DESIGN	4.349688	22175.046875	1.255163e+07	1.786533e+06
BUSINESS	4.133938	23548.202381	1.431609e+07	1.659916e+06
LIFESTYLE	4.111781	32066.859079	1.515860e+07	1.365375e+06
FINANCE	4.125257	36701.756522	1.747266e+07	1.319851e+06
HOUSE_AND_HOME	4.157028	26079.013514	1.632407e+07	1.313682e+06
DATING	4.018442	21190.315789	1.583592e+07	8.241293e+05

COMICS	4.181905	41822.696429	1.433960e+07	8.032348e+05
LIBRARIES_AND_DEMO	4.181747	10795.607143	1.087250e+07	6.309037e+05
AUTO_AND_VEHICLES	4.190824	13690.188235	1.981538e+07	6.250613e+05
PARENTING	4.282223	15972.183333	2.207688e+07	5.253518e+05
BEAUTY	4.260882	7476.226415	1.428892e+07	5.131519e+05
EVENTS	4.363647	2515.906250	1.442185e+07	2.495806e+05
MEDICAL	4.173672	2994.863291	1.911849e+07	9.669159e+04

0.2.5 分类的评论数据：社交游戏视频评论多

[88]:
```
df.groupby('Category').mean().sort_values('Reviews', ascending=False)
```

[88]:

Category	Rating	Reviews	Size	Installs
SOCIAL	4.239164	953672.807531	1.643765e+07	2.296179e+07
COMMUNICATION	4.134943	907337.676190	1.289365e+07	3.504215e+07
GAME	4.244720	648903.763295	3.973997e+07	1.447229e+07
VIDEO_PLAYERS	4.058283	414015.754601	1.631384e+07	2.409143e+07
PHOTOGRAPHY	4.159716	374915.551601	1.618812e+07	1.654501e+07
ENTERTAINMENT	4.135294	340810.294118	2.122137e+07	2.072216e+07
TOOLS	4.059823	277335.644498	9.870441e+06	9.675661e+06
SHOPPING	4.226007	220553.118812	1.593927e+07	6.932420e+06
WEATHER	4.238650	155634.987342	1.427317e+07	4.570893e+06
PRODUCTIVITY	4.185331	148638.098930	1.363180e+07	1.548955e+07
PERSONALIZATION	4.303405	142401.808511	1.168523e+07	4.075784e+06
MAPS_AND_NAVIGATION	4.052011	135337.007634	1.669496e+07	3.841846e+06
TRAVEL_AND_LOCAL	4.087611	122464.570776	2.293315e+07	1.321866e+07
EDUCATION	4.362969	112303.764706	1.882895e+07	2.965983e+06
SPORTS	4.211591	108765.578462	2.333144e+07	3.373768e+06
NEWS_AND_MAGAZINES	4.135697	91063.889764	1.365578e+07	9.327629e+06
FAMILY	4.181330	78550.239214	2.666982e+07	2.418319e+06
BOOKS_AND_REFERENCE	4.308770	75321.234234	1.376752e+07	7.504367e+06
HEALTH_AND_FITNESS	4.235441	74171.371528	2.018017e+07	3.972300e+06
FOOD_AND_DRINK	4.175715	56473.464286	1.999241e+07	1.891060e+06
COMICS	4.181905	41822.696429	1.433960e+07	8.032348e+05
FINANCE	4.125257	36701.756522	1.747266e+07	1.319851e+06

LIFESTYLE	4.111781	32066.859079	1.515860e+07	1.365375e+06
HOUSE_AND_HOME	4.157028	26079.013514	1.632407e+07	1.313682e+06
BUSINESS	4.133938	23548.202381	1.431609e+07	1.659916e+06
ART_AND_DESIGN	4.349688	22175.046875	1.255163e+07	1.786533e+06
DATING	4.018442	21190.315789	1.583592e+07	8.241293e+05
PARENTING	4.282223	15972.183333	2.207688e+07	5.253518e+05
AUTO_AND_VEHICLES	4.190824	13690.188235	1.981538e+07	6.250613e+05
LIBRARIES_AND_DEMO	4.181747	10795.607143	1.087250e+07	6.309037e+05
BEAUTY	4.260882	7476.226415	1.428892e+07	5.131519e+05
MEDICAL	4.173672	2994.863291	1.911849e+07	9.669159e+04
EVENTS	4.363647	2515.906250	1.442185e+07	2.495806e+05

0.2.6 分类的打分数据

[66]:
```
df.groupby('Category').mean().sort_values('Rating', ascending=False)
```

[66]:

Category	Rating	Reviews	Size	Installs
EVENTS	4.363647	2515.906250	1.442185e+07	2.495806e+05
EDUCATION	4.362969	112303.764706	1.882895e+07	2.965983e+06
ART_AND_DESIGN	4.349688	22175.046875	1.255163e+07	1.786533e+06
BOOKS_AND_REFERENCE	4.308770	75321.234234	1.376752e+07	7.504367e+06
PERSONALIZATION	4.303405	142401.808511	1.168523e+07	4.075784e+06
PARENTING	4.282223	15972.183333	2.207688e+07	5.253518e+05
BEAUTY	4.260882	7476.226415	1.428892e+07	5.131519e+05
GAME	4.244720	648903.763295	3.973997e+07	1.447229e+07
SOCIAL	4.239164	953672.807531	1.643765e+07	2.296179e+07
WEATHER	4.238650	155634.987342	1.427317e+07	4.570893e+06
HEALTH_AND_FITNESS	4.235441	74171.371528	2.018017e+07	3.972300e+06
SHOPPING	4.226007	220553.118812	1.593927e+07	6.932420e+06
SPORTS	4.211591	108765.578462	2.333144e+07	3.373768e+06
AUTO_AND_VEHICLES	4.190824	13690.188235	1.981538e+07	6.250613e+05
PRODUCTIVITY	4.185331	148638.098930	1.363180e+07	1.548955e+07
COMICS	4.181905	41822.696429	1.433960e+07	8.032348e+05
LIBRARIES_AND_DEMO	4.181747	10795.607143	1.087250e+07	6.309037e+05
FAMILY	4.181330	78550.239214	2.666982e+07	2.418319e+06

```
FOOD_AND_DRINK       4.175715   56473.464286  1.999241e+07  1.891060e+06
MEDICAL              4.173672    2994.863291  1.911849e+07  9.669159e+04
PHOTOGRAPHY          4.159716  374915.551601  1.618812e+07  1.654501e+07
HOUSE_AND_HOME       4.157028   26079.013514  1.632407e+07  1.313682e+06
NEWS_AND_MAGAZINES   4.135697   91063.889764  1.365578e+07  9.327629e+06
ENTERTAINMENT        4.135294  340810.294118  2.122137e+07  2.072216e+07
COMMUNICATION        4.134943  907337.676190  1.289365e+07  3.504215e+07
BUSINESS             4.133938   23548.202381  1.431609e+07  1.659916e+06
FINANCE              4.125257   36701.756522  1.747266e+07  1.319851e+06
LIFESTYLE            4.111781   32066.859079  1.515860e+07  1.365375e+06
TRAVEL_AND_LOCAL     4.087611  122464.570776  2.293315e+07  1.321866e+07
TOOLS                4.059823  277335.644498  9.870441e+06  9.675661e+06
VIDEO_PLAYERS        4.058283  414015.754601  1.631384e+07  2.409143e+07
MAPS_AND_NAVIGATION  4.052011  135337.007634  1.669496e+07  3.841846e+06
DATING               4.018442   21190.315789  1.583592e+07  8.241293e+05
```

0.2.7 免费占比大，付费占比小，免费仍然是主流

[67]:
```
# 分 Type 数据
df.groupby('Type').count()
```

[67]:
```
        App  Category  Rating  Reviews  Size  Installs
Type
Free   8902      8902    8902     8902  8902      8902
Paid    756       756     756      756   756       756
```

0.2.8 只有两个类型，且数据量差别很大，没必要继续对比了

[89]:
```
df.groupby('Type').sum().sort_values('Installs', ascending=False)
```

[89]:
```
            Rating     Reviews          Size     Installs
Type
Free  37124.373193  2085471559  1.799151e+11  75065572646
Paid   3210.187424     6596015  1.431240e+10     57364881
```

[69]:
```
# Category 和 Type 一起分析
df.groupby(['Type', 'Category']).mean().sort_values('Reviews', ascending=False)
```

[69]:

Type	Category	Rating	Reviews	Size	Installs
Free	COMMUNICATION	4.139376	992108.173611	1.350167e+07	3.832263e+07
	SOCIAL	4.243927	965794.741525	1.656355e+07	2.325365e+07
	GAME	4.234010	707783.190422	4.036479e+07	1.580151e+07
	VIDEO_PLAYERS	4.057233	424347.176101	1.636918e+07	2.469705e+07
	PHOTOGRAPHY	4.167583	401664.270992	1.667036e+07	1.773767e+07
…		…	…	…	…
Paid	NEWS_AND_MAGAZINES	4.800000	100.500000	1.490000e+07	2.750000e+03
	SOCIAL	3.864446	80.666667	6.533333e+06	2.000000e+03
	BOOKS_AND_REFERENCE	4.216670	64.142857	1.258550e+07	8.327143e+02
	LIBRARIES_AND_DEMO	4.193338	4.000000	4.700000e+06	1.000000e+02
	EVENTS	4.193338	0.000000	6.700000e+06	1.000000e+00

[63 rows x 4 columns]

0.2.9 收费的 app 评论比率更高

[70]:
```
# 评论安装比
g = df.groupby(['Type', 'Category']).mean()
(g['Reviews'] / g['Installs']).sort_values(ascending=False)
```

[70]:
```
Type  Category
Paid  VIDEO_PLAYERS          0.188268
      FAMILY                 0.175913
      WEATHER                0.168031
      PARENTING              0.166986
      DATING                 0.141674
                               …
Free  BOOKS_AND_REFERENCE    0.010036
      NEWS_AND_MAGAZINES     0.009763
      PRODUCTIVITY           0.009569
      TRAVEL_AND_LOCAL       0.009259
Paid  EVENTS                 0.000000
Length: 63, dtype: float64
```

0.2.10 评论数和安装数强相关，其他的连 0.1 都不到，可以认为是不相关的（0.5 以上可以认为是相关的，0.3 以上可以认为是弱相关）

[71]:
```
# 相关性分析
df.corr()
```

[71]:

	Rating	Reviews	Size	Installs
Rating	1.000000	0.054278	0.052600	0.039174
Reviews	0.054278	1.000000	0.080578	0.625164
Size	0.052600	0.080578	1.000000	0.050675
Installs	0.039174	0.625164	0.050675	1.000000

[ ]:
```

```