Google Play Store Apps数据集分析

**一、设计内容**

利用Kaggle数据集（股票、新闻、金融、商业皆可），要求：自行找到公开数据集，进行基本的数据存储、数据加载、数据清洗、数据挖掘算法建模，报告中必须包含数据的详细信息和建模结果。

**二、问题分析**

我们小组选题来自于kaggle数据集：题目需要根据数据集分析谷歌商店APP的一些评论、正面、负面的相关信息，分析出APP价格、APP种类、用户情绪等因素对APP评分的影响。Play商店应用数据具有巨大的潜力，可以推动应用制作业务取得成功。可以得出可行的见解，以供开发人员从事和占领Android市场！

总共使用到了以下两个数据集：

①googleplaystore.csv （谷歌商店APP的信息）

②googleplaystore\_user\_reviews.csv （用户的评论）

**三、环境搭建**

工具:Visual Studio Code

环境:python 3.8.3

导入环境依赖：

1. %matplotlib inline # 配置
2. # 导包
3. **import** pandas as pd # 一种分析结构化数据集的工具
4. **import** numpy as np  # 数学运算函数库
5. **from** matplotlib **import** pyplot as plt # 画图库
6. plt.style.use('ggplot')   # 给画图库初始化
8. **import** seaborn as sns # 可视化库
9. color = sns.color\_palette() # 获取调色盘
10. sns.set(rc={'figure.figsize': (25, 15)}) #初始化 可视化库
12. **import** plotly # 可视化框架
13. plotly.offline.init\_notebook\_mode(connected=True) # 离线可视化,让图表可以嵌入到notebook中
14. **import** plotly.graph\_objs as go # 图形对象集合
16. **import** warnings # 压制警告
17. warnings.filterwarnings('ignore')

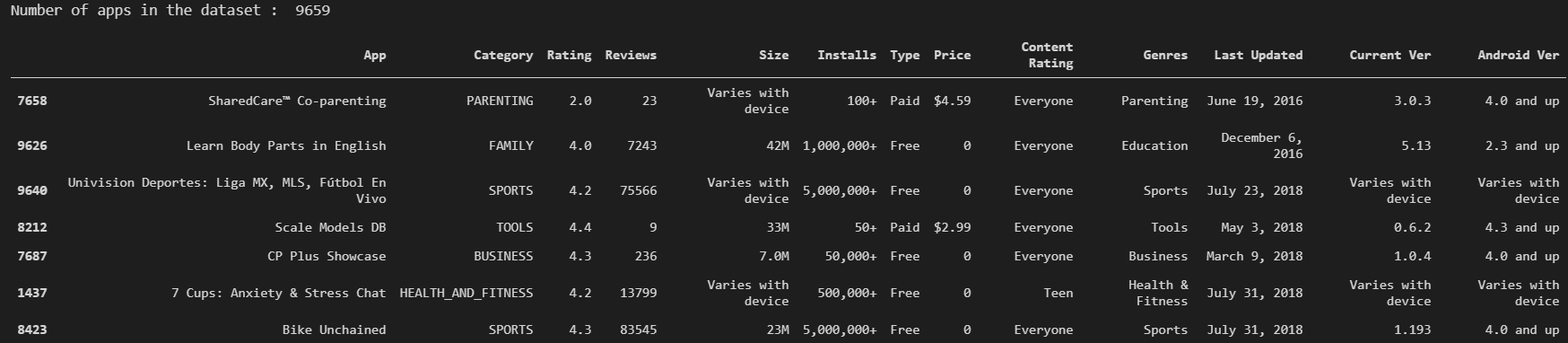
**四、数据观察与预处理**

1.首先读入题目所给数据

1. df = pd.read\_csv('googleplaystore.csv') #读取数据
2. # 处理特殊情况
3. df.drop\_duplicates(subset='App', inplace=True) # 去重
4. df = df[df['Android Ver'] != np.nan] #去空
5. df = df[df['Android Ver'] != 'NaN']
6. df = df[df['Installs'] != 'Free']
7. df = df[df['Installs'] != 'Paid']
9. **print**('Number of apps in the dataset : ' , len(df))
10. df.sample(7) # 简单输出

通过read\_csv函数将googleplaystore.csv这个数据集读入，然后对数据集进行了去重、去空等简单处理，使读取到的原始数据能达到更好的效果。

可以看到，训练集总共有9659条数据，部分数据如下：



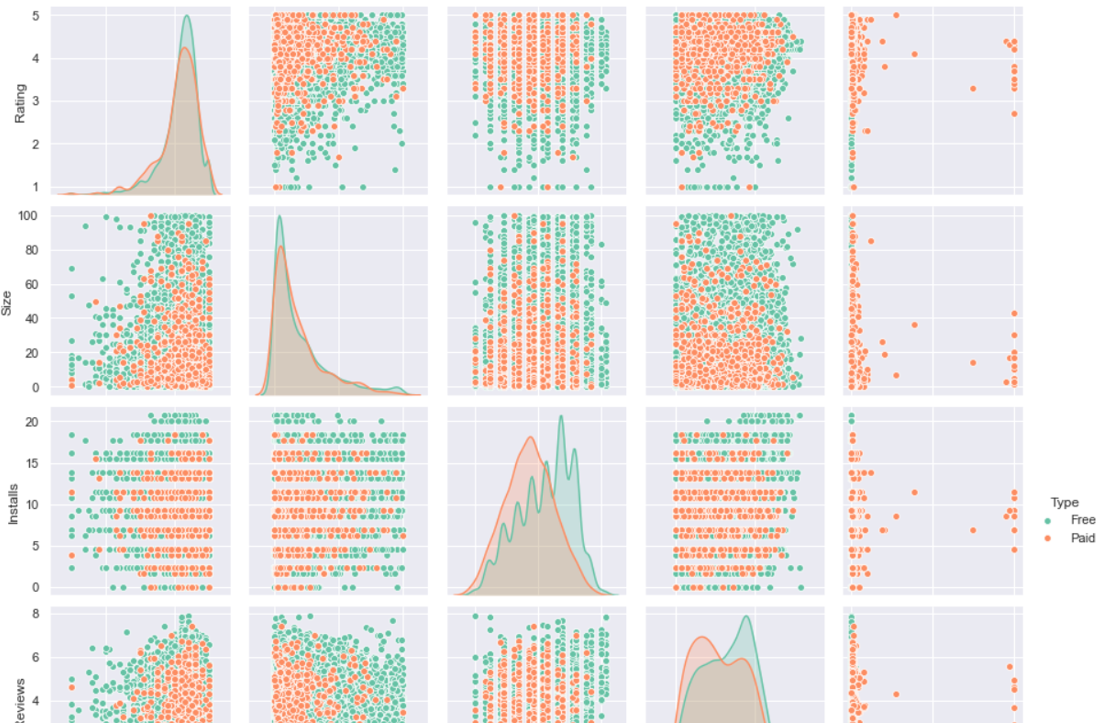
2.进行缺失值的填充以及符号处理：将数据集中的“+”“，”等特殊符号替换为空字符串，便于后续的分析与处理：

1. # - Installs : Remove + and ,
2. df['Installs'] = df['Installs'].apply(**lambda** x : x.replace('+', '') **if** '+' **in** x **else** x)
3. df['Installs'] = df['Installs'].apply(**lambda** x : x.replace(',', '') **if** ',' **in** x **else** x)
4. df['Installs'] = df['Installs'].apply(**lambda** x : int(x))
5. # df.head()
6. # - Size : Remove 'M', Replace 'k' and divide by 10^-3
7. #df['Size'] = df['Size'].fillna(0)
8. df['Size'] = df['Size'].apply(**lambda** x : x.replace('Varies with device', 'NaN') **if** 'Varies with device' **in** str(x) **else** x)
9. df['Size'] = df['Size'].apply(**lambda** x : x.replace('M', '') **if** 'M' **in** x **else** x)
10. df['Size'] = df['Size'].apply(**lambda** x : float(x.replace('k', '')) / 1000 **if** 'k' **in** x **else** x)
11. df['Size'] = df['Size'].apply(**lambda** x : float(x))
12. df['Installs'] = df['Installs'].apply(**lambda** x : float(x))
13. df['Price'] = df['Price'].apply(**lambda** x : x.replace('$', '') **if** '$' **in** x **else** x).astype(float)
14. df['Reviews'] = df['Reviews'].astype(int)
15. # df.head()

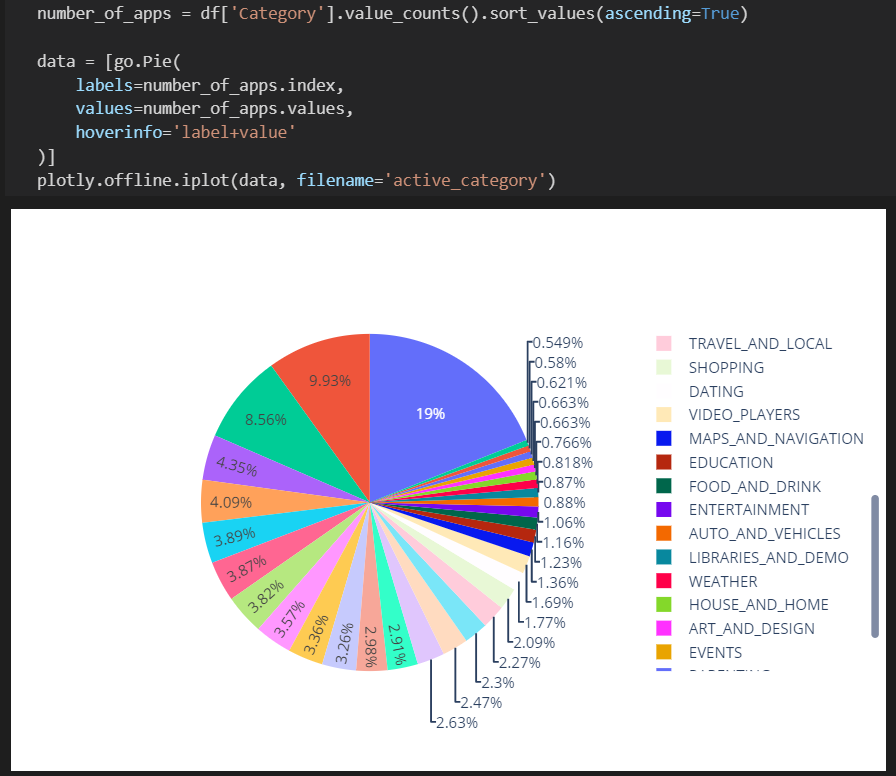
**五、数据可视化**

完成准备数据、预处理等工作之后，接下来就可以利用一些可视化技术来完成对数据的可视化分析。

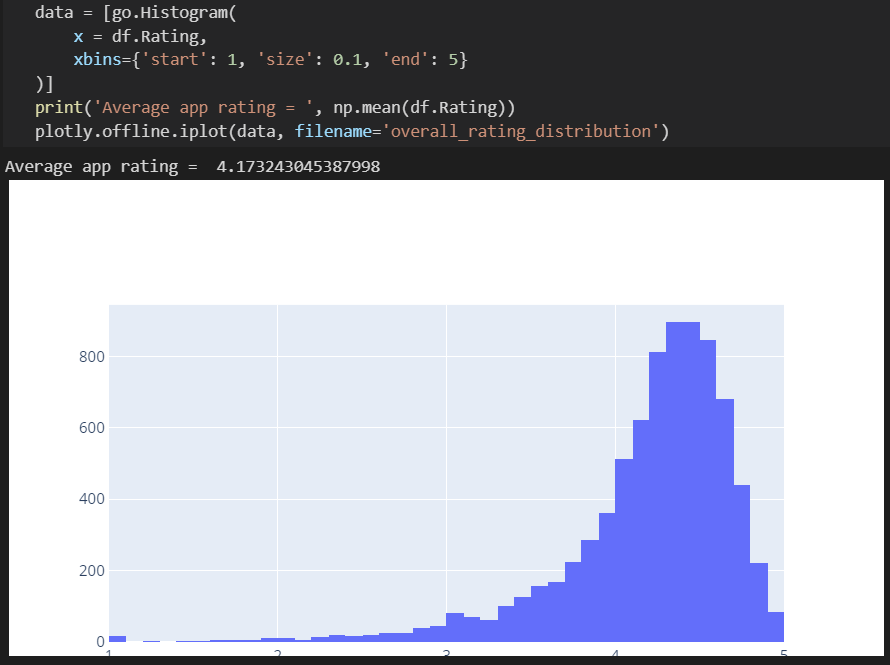
5.1 首先我们对’Rating', 'Size', 'Installs', 'Reviews', 'Type', 'Price’这6个属性进行了两两成对关系的分析，结果如下图所示：

通过上图我们可以清晰明了的看出两个属性之间的耦合性。在其中耦合性较高的有Rating- Reviews，Installs- Reviews。

5.2然后我们通过画Pie图对数据集中各种软件类型的占比进行了分析，结果如下图所示：

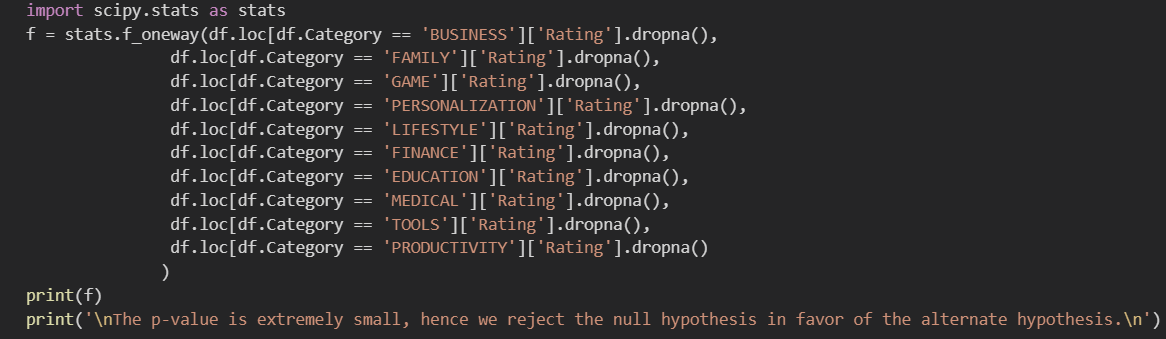


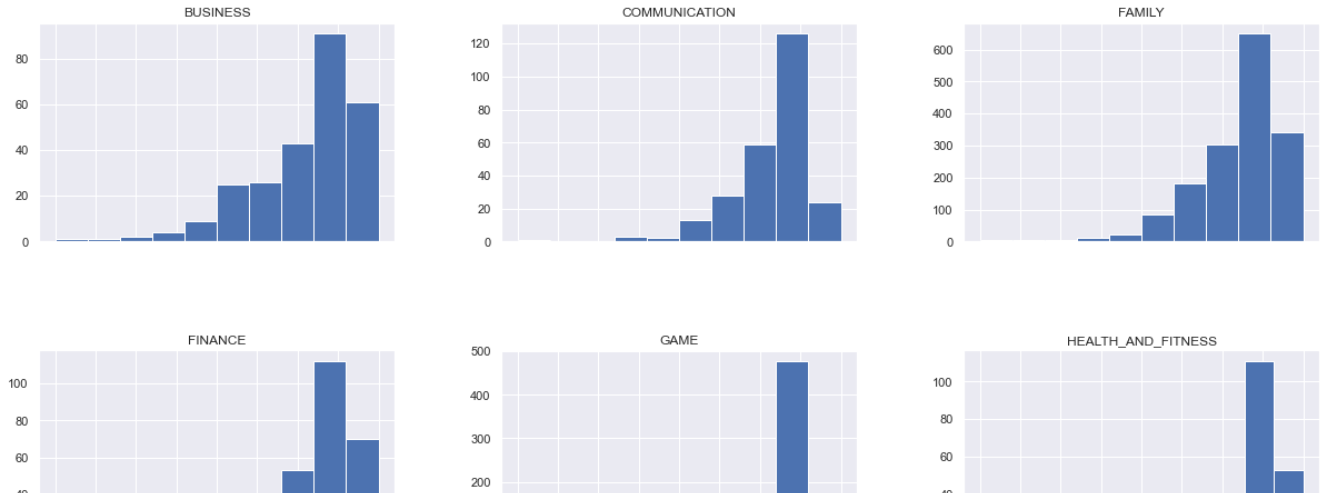
5.3接下来对数据集中的评分一栏的信息作出了可视化分析，结果如下图所示：



从这个直方图中可以清晰明显的看出，在对APP的评分中，处于4.3和4.4评分的用户数是最多的，处于2.0评分以下的用户数占比极小，可以大致说明用户对APP的满意度还是较高的。而关于对APP满意度的分析我们还将在后续继续进行。

5.4 对各方面的单因素方差进行了分析，结果如下图所示：

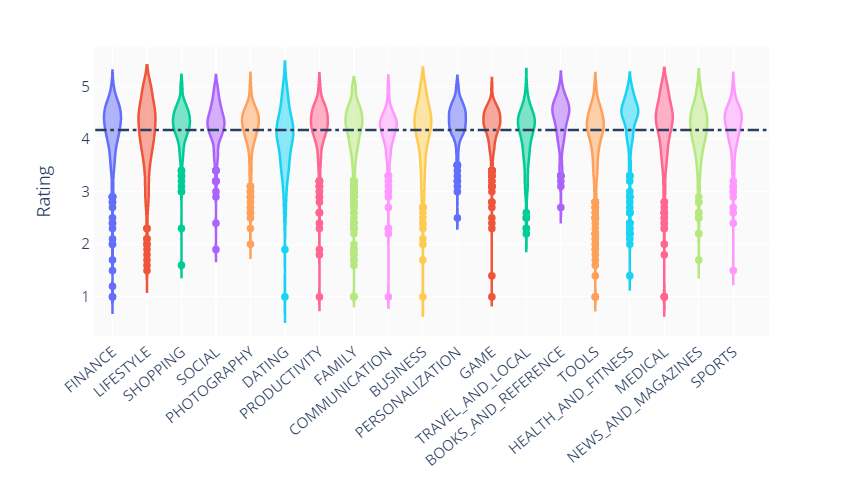




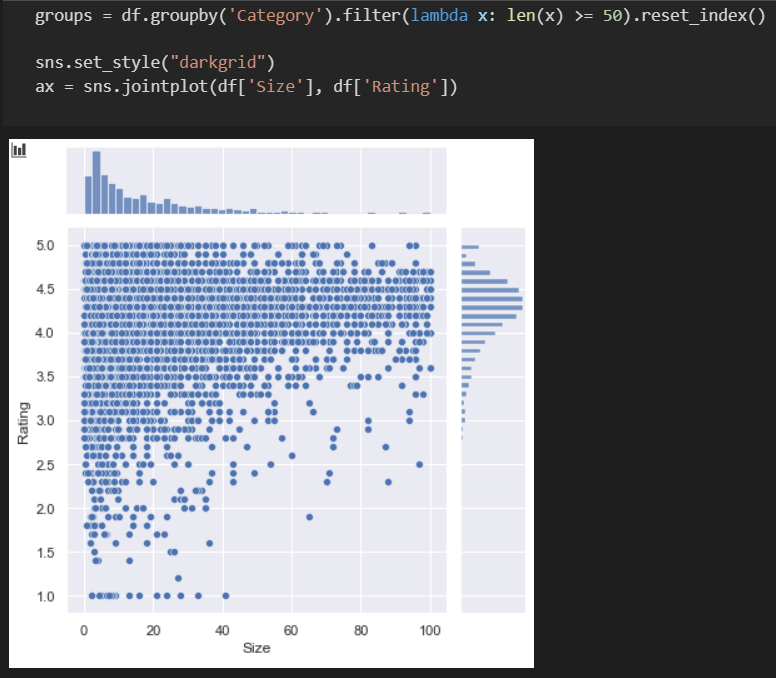
通过直方图对方差进行可视化分析，可以清楚的度量出属性的离散程度。

5.5 然后是对最佳表现类别做了分析：



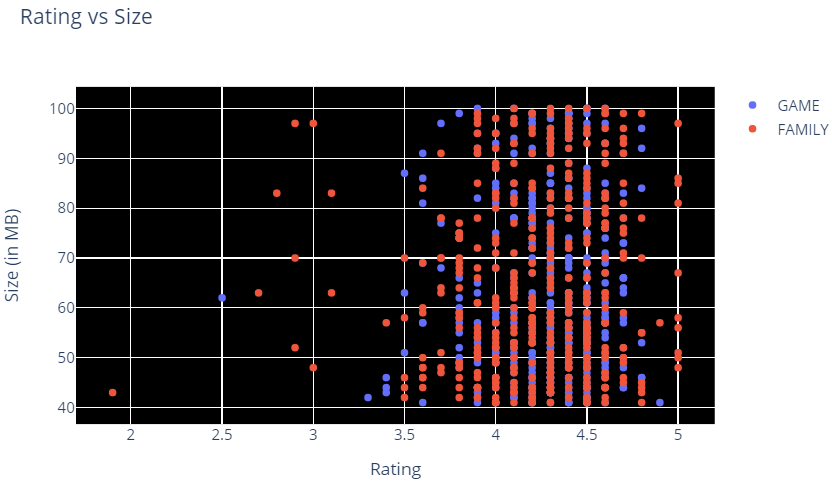


5.6 分析安装包大小与评分的关系



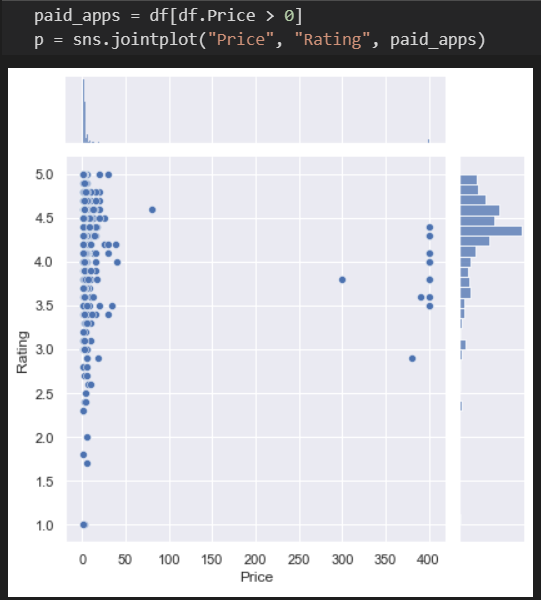
从上图中可以看出，整体的评分集中度是在3.8-4.8分之间的。当我们以50MB作为界限进行划分的时候，小于50MB的安装包的数量更多，且有低分现象出现。大于50MB的安装包中，评分较为均衡，只有少数中等偏下的分。

5.7 GAME & FAMILY 的大小，评分分布统计



**六、对商店的价格策略的分析**

6.1 APP价格统计

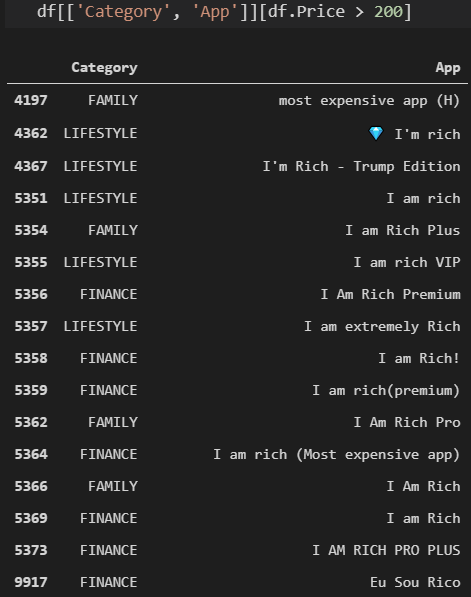


可以看出，APP价格95%以上位于[0-50]的价格区间内，只能少数价格偏高。

6.2 价格分类统计

6.3 将少量的高价应用去除后在进行价格的分类统计

首先找出高价应用：



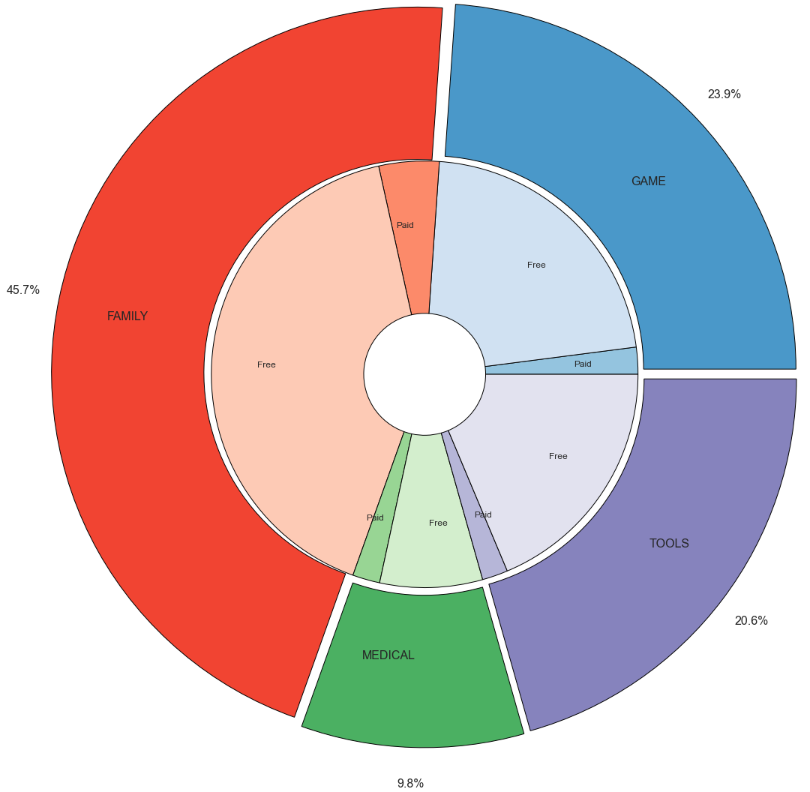
然后将上述应用去除掉后在进行分类统计：



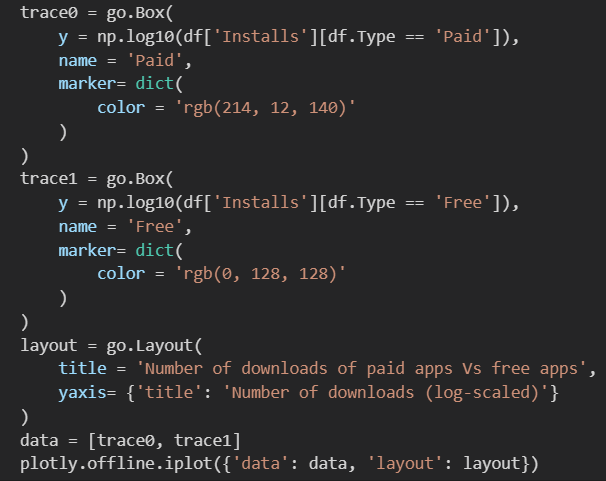
通过与先前的进行对比，可以看出：将极少量的高价应用去除掉后，在[0-50]价格区间内所呈现出来的散点图更细腻，更直观。因为绝大部分的APP价格是位于这个区间的，所以这种做法能够让我们分析起来更加容易，得出的结果的准确率也会更高。

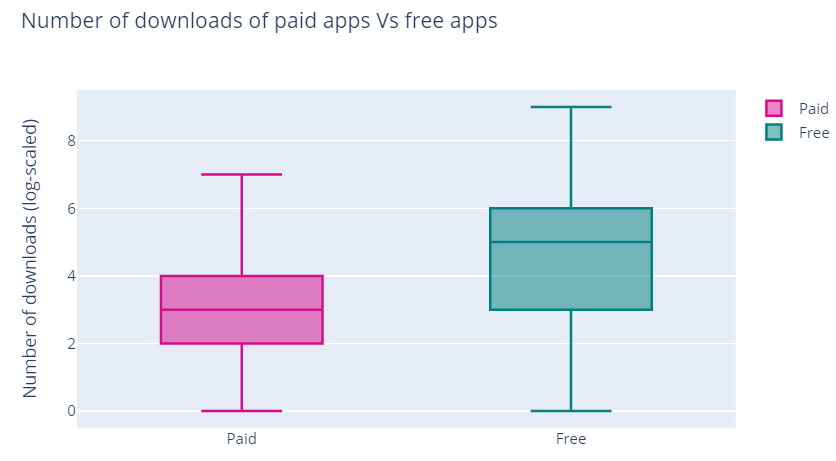
6.4 画Pie图，分类统计Free和Paid占比



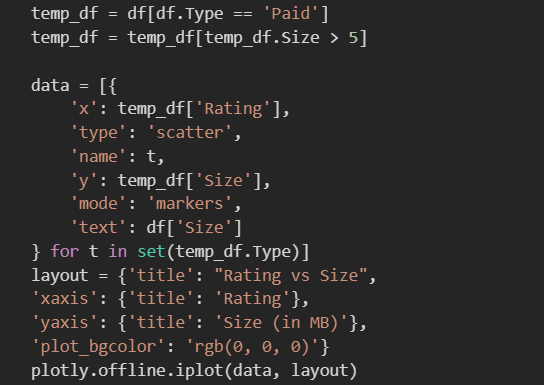


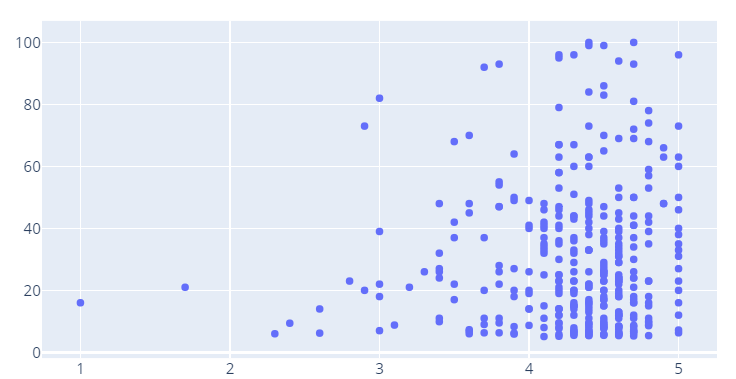
6.5 分析付费下载应用数量对比免费应用



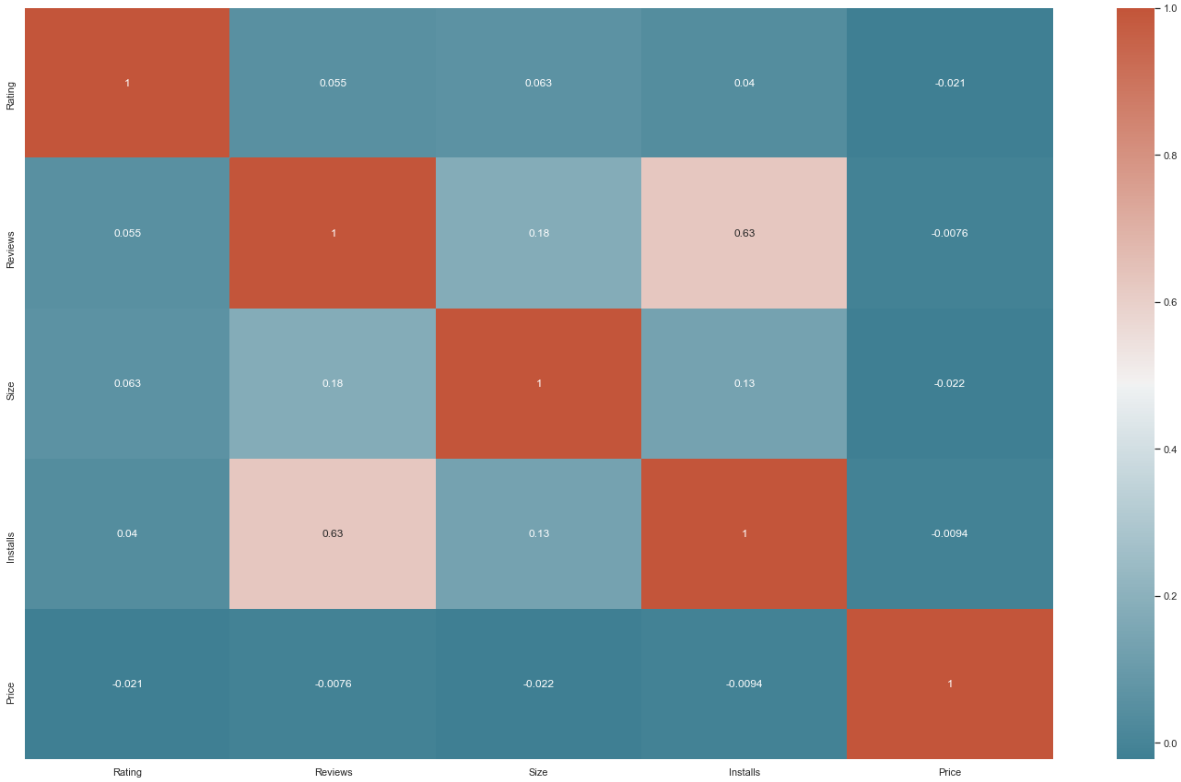


6.6 在价格策略中，对评分与安装包大小的统计

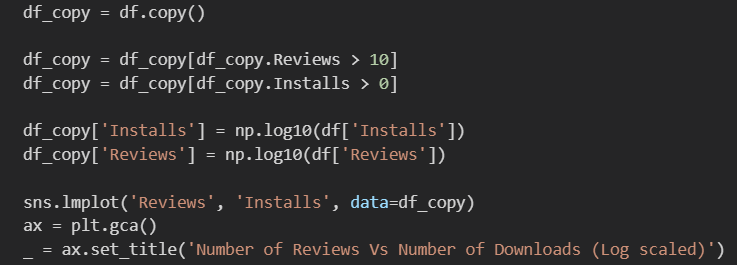


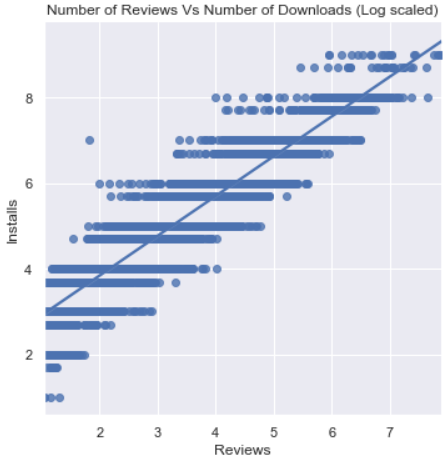


然后统计其相关性：



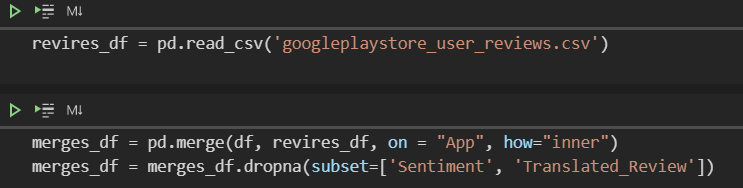
6.7 在价格策略中，统计下载量与评论量的关系

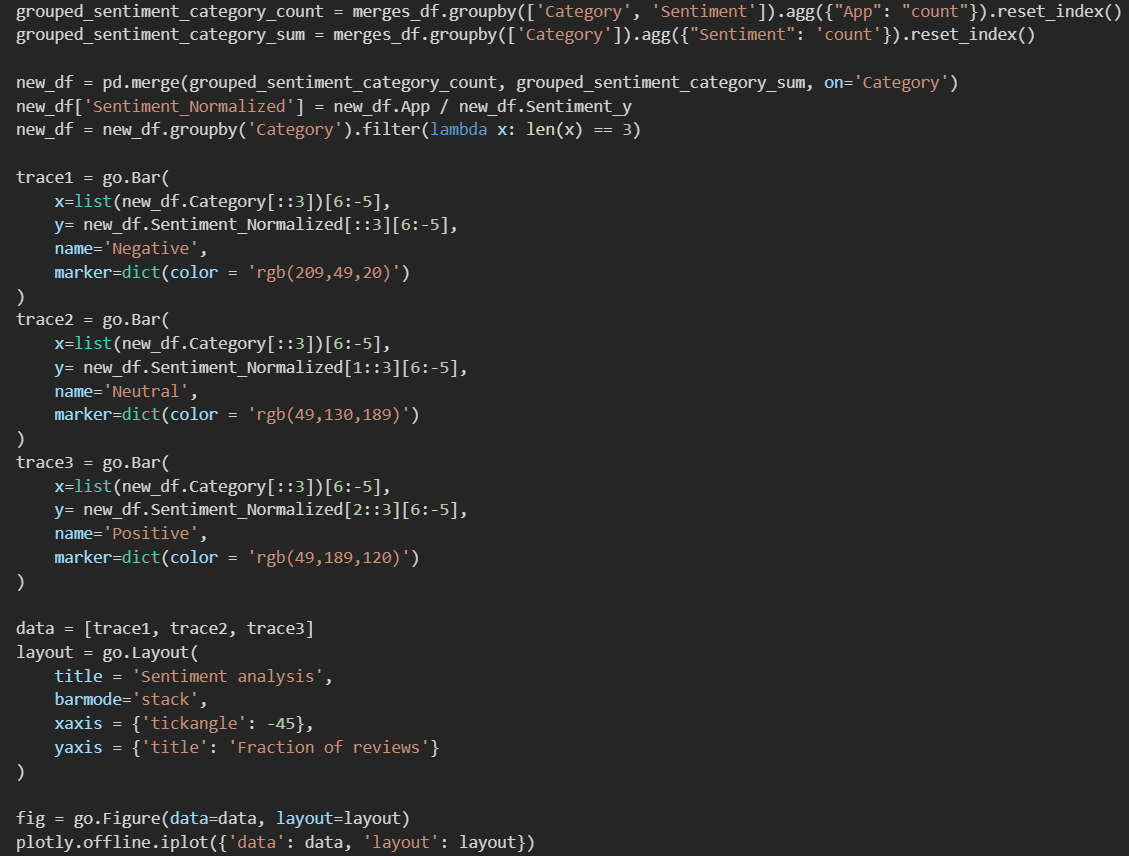


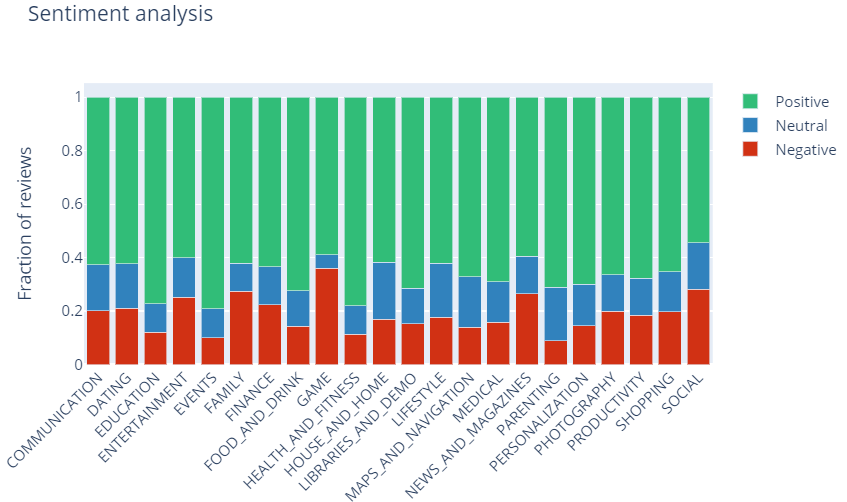


**七、用户评论的情绪分析**

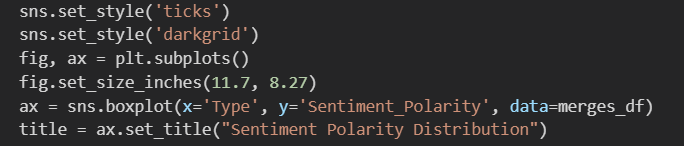
7.1对评论中积极的、中性的、消极的评论的分类统计

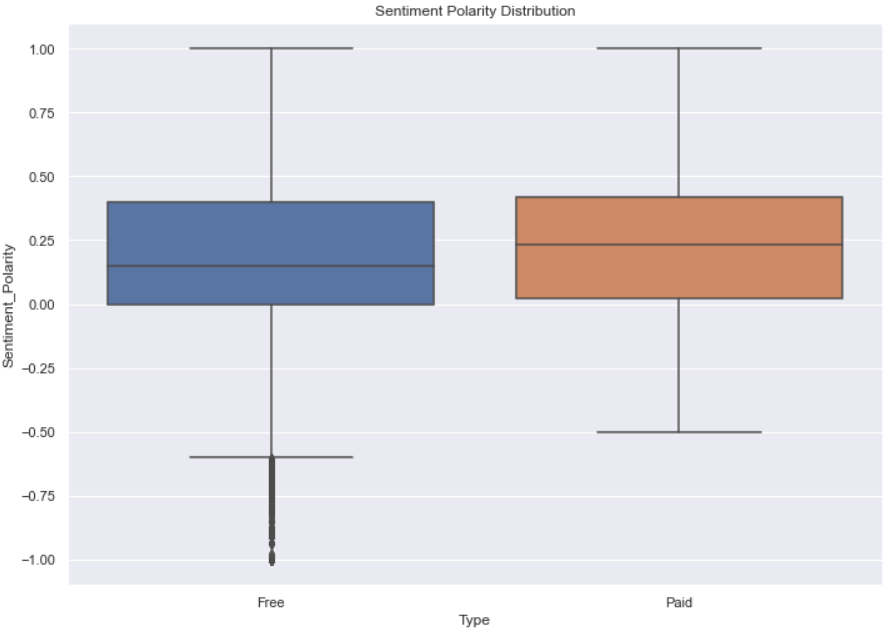






7.2 付费应用与免费应用的情绪分布

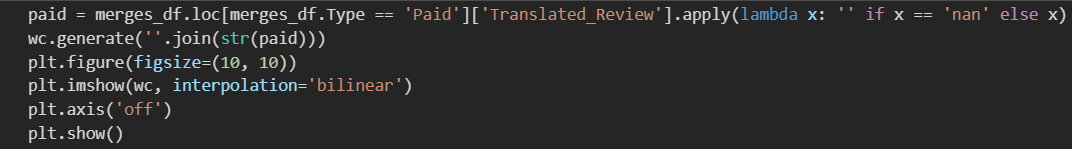




7.3 用户评论的高频词语统计









**八、附录**

8.1 代码以及数据集合已上传至Github