淘宝用户价值分析
——基于RFM模型识别客户价值指标

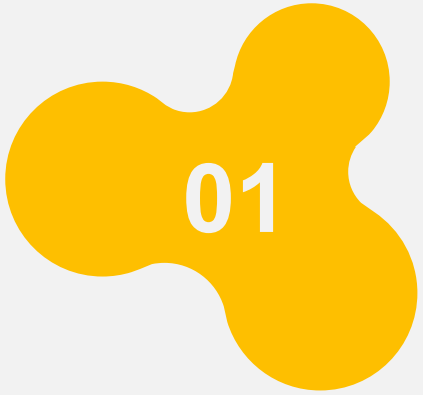
成员分工

陈泽泰：数据预处理与数据清洗

周楚凡：数据可视化与RFM分层

靖相宜：数据分析与提供战略意见

目录



项目介绍

客户群体分析

基于RFM模型分析

战略意见

01

项目介绍

项目概述

该项目是对淘宝用户信息进行解析，根据客户数据和用户类别特征，对客户进行分类，使用RFM方法来科学预测老客户今后的购买金额，分析出今后的客户价值，从而针对不同用户制定相应的营销策略。

项目背景

RFM模型是衡量客户价值和客户创利能力的重要工具和手段，它通过一个客户最近购买日Recency, 各期购买频率Frequency, 各期平均单次购买金额Monetary三项指标来描述该客户的价值状况。

对于电子商务卖家而言，运用RFM模型可以从所有历史客户群中迅速定位那些可能“最有价值“的客户，并通过随后及时的联络沟通，将其潜在购买转化为实际购买行为，从而进一步增强客户忠诚度，封杀竞争对手的市场空间。

02

客户群体分析

数据处理

讲无法进行分析的“信用得分”的字符串转换为有次序的数值型变量

```
# -*- coding: utf-8 -*-
"""
Created on Sun Oct  7 09:47:09 2018

@author: Administrator
"""

#导入三方库
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels. api as sm

#导入原始数据
df=pd.read_csv(r'D:\Application\BA\alibaba\customers.csv')
df2=pd.read_csv(r'D:\Application\BA\alibaba\dealing.csv')
#数据转换
df['buy_cred'].replace("信用得分=0",1,inplace=True)
df['buy_cred'].replace("信用得分<=3",2,inplace=True)
df['buy_cred'].replace("1星级",3,inplace=True)
df['buy_cred'].replace("2星级",4,inplace=True)
df['buy_cred'].replace("3星级",5,inplace=True)
df['buy_cred'].replace("4星级",6,inplace=True)
df['buy_cred'].replace("5星级",7,inplace=True)
df['buy_cred'].replace("1钻",8,inplace=True)
df['buy_cred'].replace("2钻",9,inplace=True)
df['buy_cred'].replace("3钻",10,inplace=True)
df['buy_cred'].replace("4钻",11,inplace=True)
df['buy_cred'].replace("5钻",12,inplace=True)
df['buy_cred'].replace("1皇冠",13,inplace=True)
```

数据处理

进行数据检查和处理：
剔除缺少“信用得分”、购买省份、性别、年龄的客户数据

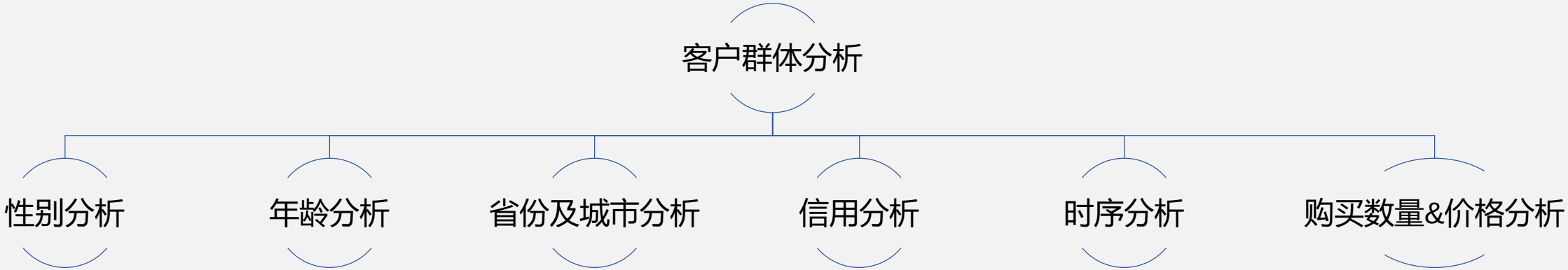
```
#%%
df.info()
df.describe()
df.isnull()
df.head()
df['buy_cred'].isnull().value_counts
#剔除缺少信用得分值的行
df["buy_cred"]=df["buy_cred"].apply(lambda x: np.NaN if str(x).isspace() else x)
df=df_not_null=df[df["buy_cred"].notnull()]
#剔除缺少购买省份的行
df=df_not_null1=df[~df['buy_city'].isin(['未知'])]
#剔除缺少性别、年龄的行（至此 所有 缺少值的行 已剔除）
df["buy_gender"]=df["buy_gender"].apply(lambda x: np.NaN if str(x).isspace() else x)
df=df_not_null2=df[df["buy_gender"].notnull()]
```

数据处理

导出处理后的客户数据，
将客户数据和客户订单合并，使客户ID对应上订单信息。
最后得出买家的购买次数。

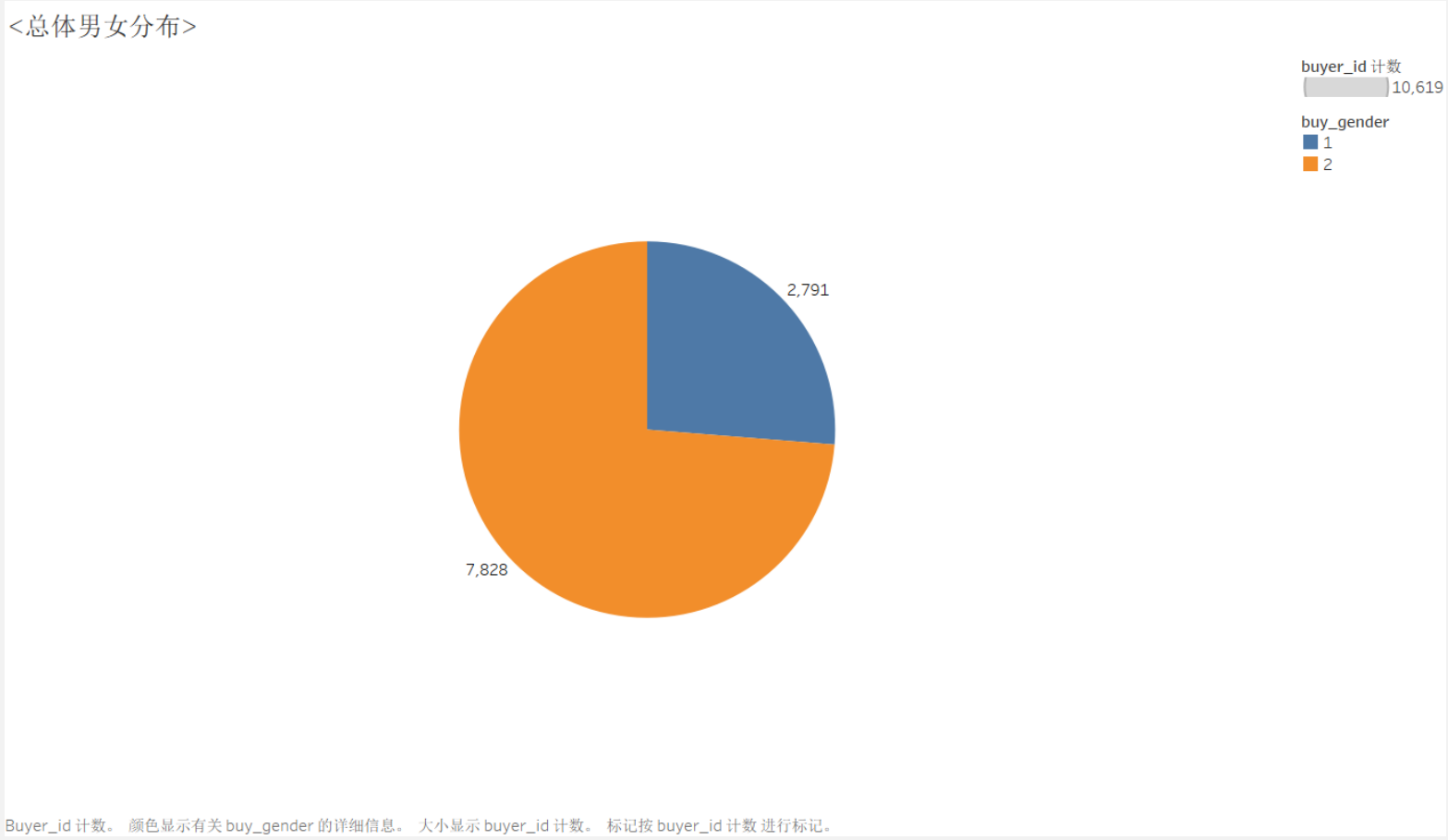
```
#%%
df.info()
df2.info()
df.to_csv('Clean Data Customer.csv')
#%%
#合并两表
df3=pd.merge(df,df2,how='left',on=['buyer_id'])
df3['shipcost']=df3['shipcost'].astype('int')
df3.sort_values(by=['buyer_id'],ascending=True,inplace=True)
df3.to_csv('Merge.csv')
#查看各买家ID购买的次数 筛选出有复购行为的买家ID
buyerid_purchasetimes=df3.groupby(['buyer_id'])['price'].count()
buyerid_purchasetimes.to_csv('purchasetimes.csv')
```

分析概况



性别分析

由图片可以看出，女性为购物主要群体，占比近75%

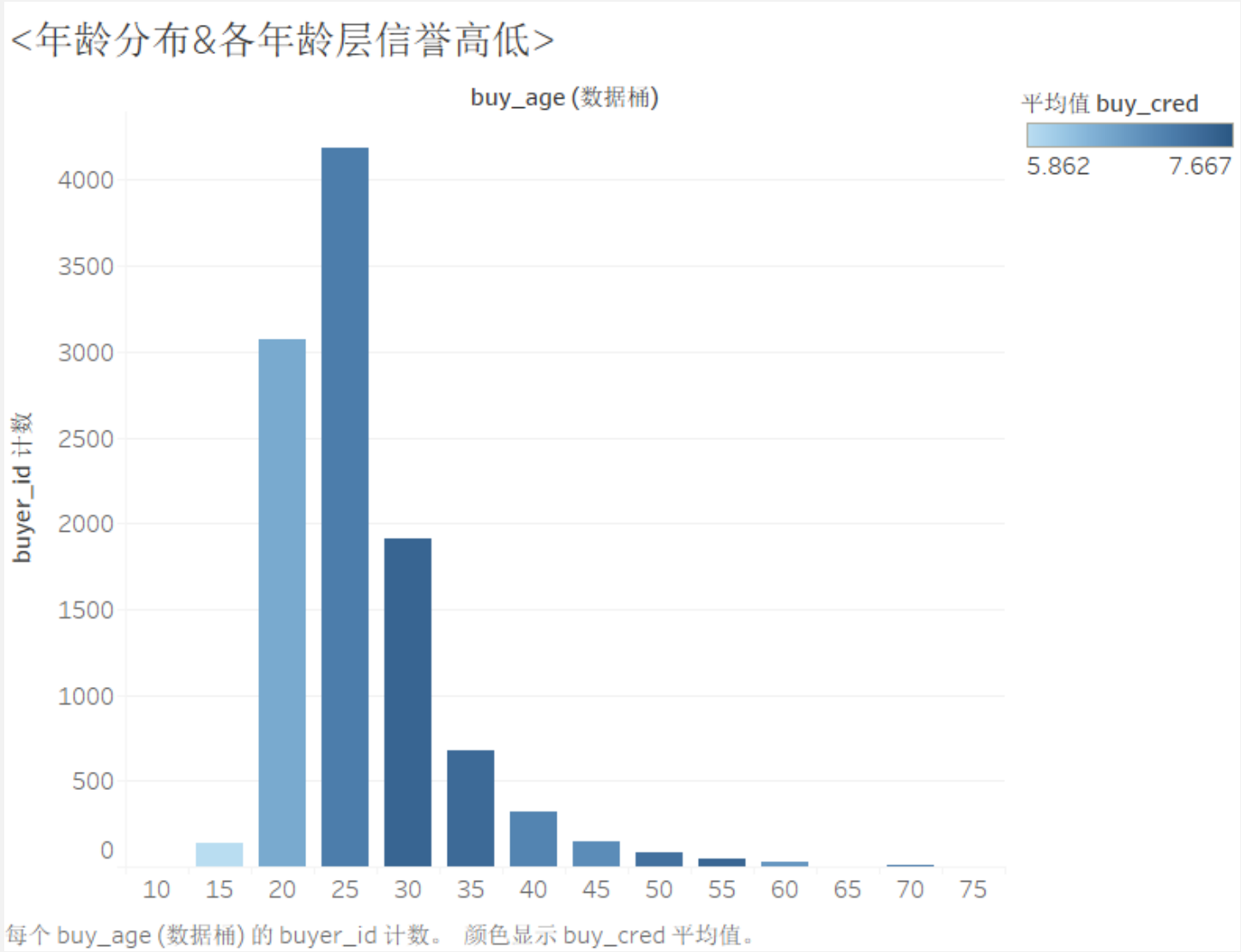


性别分析

由图片可以看出，上海、杭州等地男性购买者多于北京、天津等地。

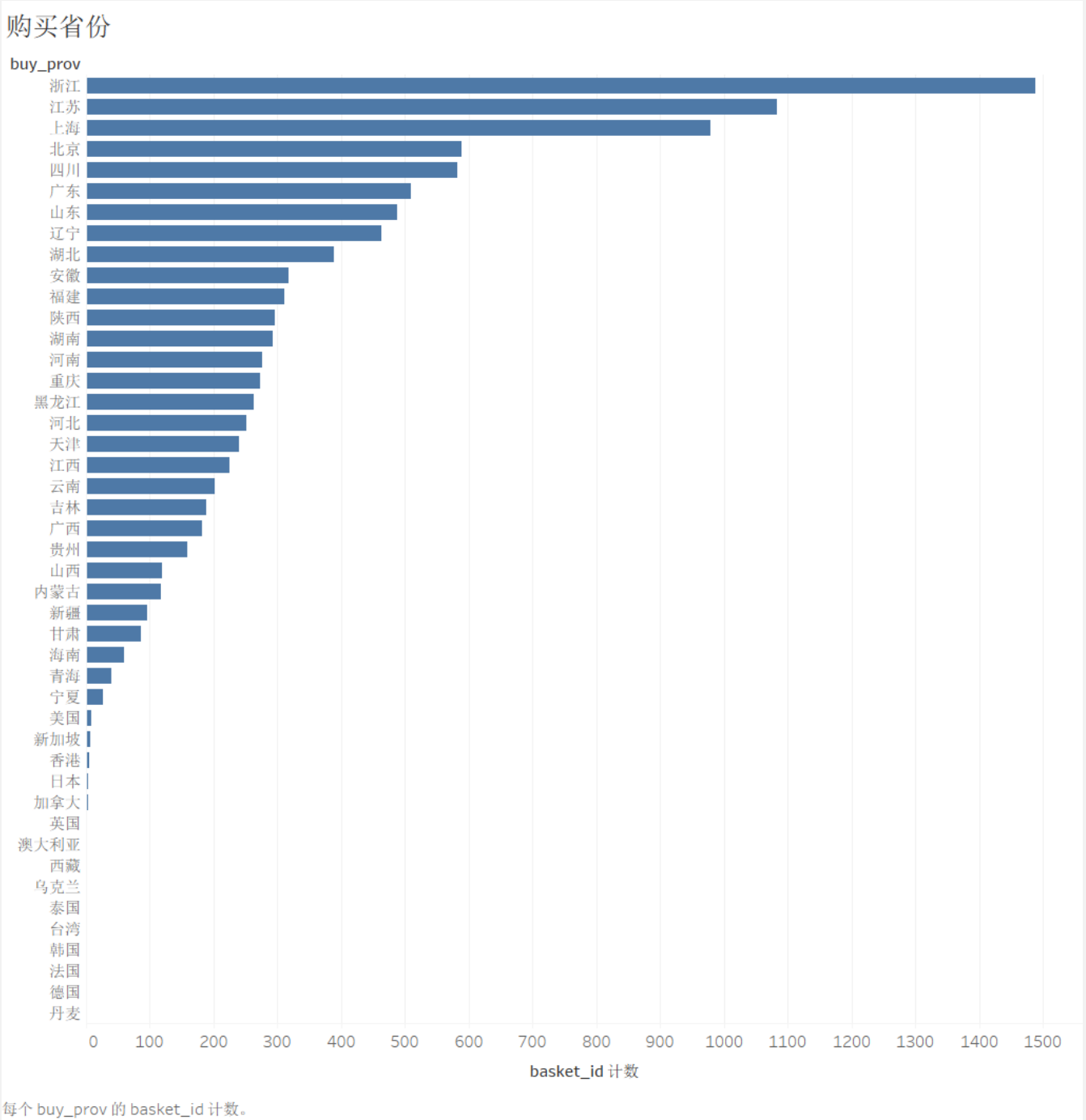
年龄分析

由图片可以看出，主要购买者的年龄为20-25岁，其次为25-30岁，50岁以上及10岁以下消费人群较少。
就信誉高低而言，30-35岁信誉等级较高，这类群体有固定收入，因此信用等级较高。
综合来看，25岁人群购买力最高，信誉等级也良好，可以为推销的主体。



省份及城市分析

从图片可知，江浙沪地区购买者比较多，上海位居第一，其次为杭州，北京为第三。



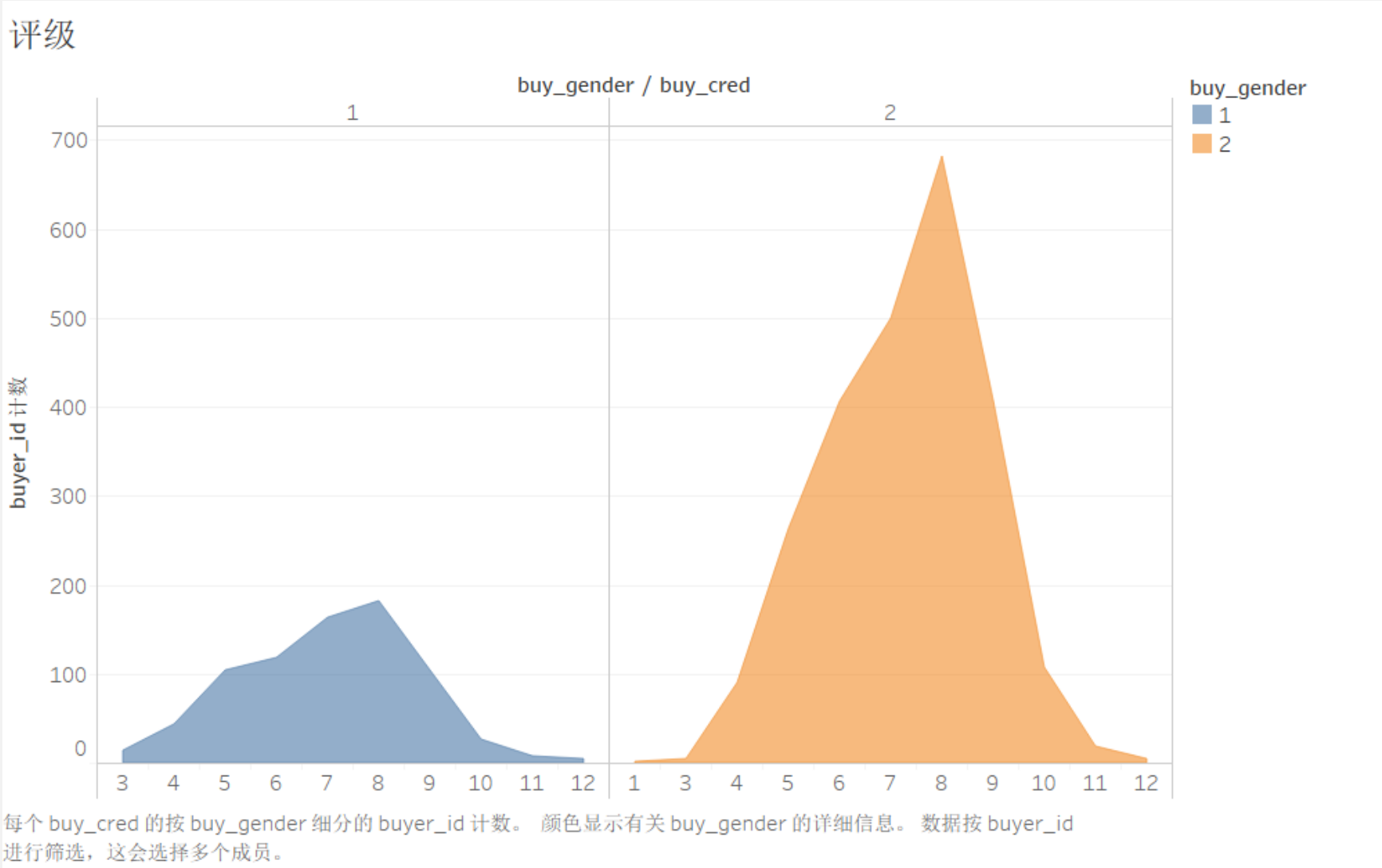
省份及城市分析

<各省份购买数量>

基于 经度(生成) 和 纬度(生成) 的地图。 颜色显示 buy_cred 平均值。 大小显示 goods_id 计数。 标记按 buy_city 进行标记。 数据按 buyer_id 进行筛选，这会选择多个成员。

信用分析

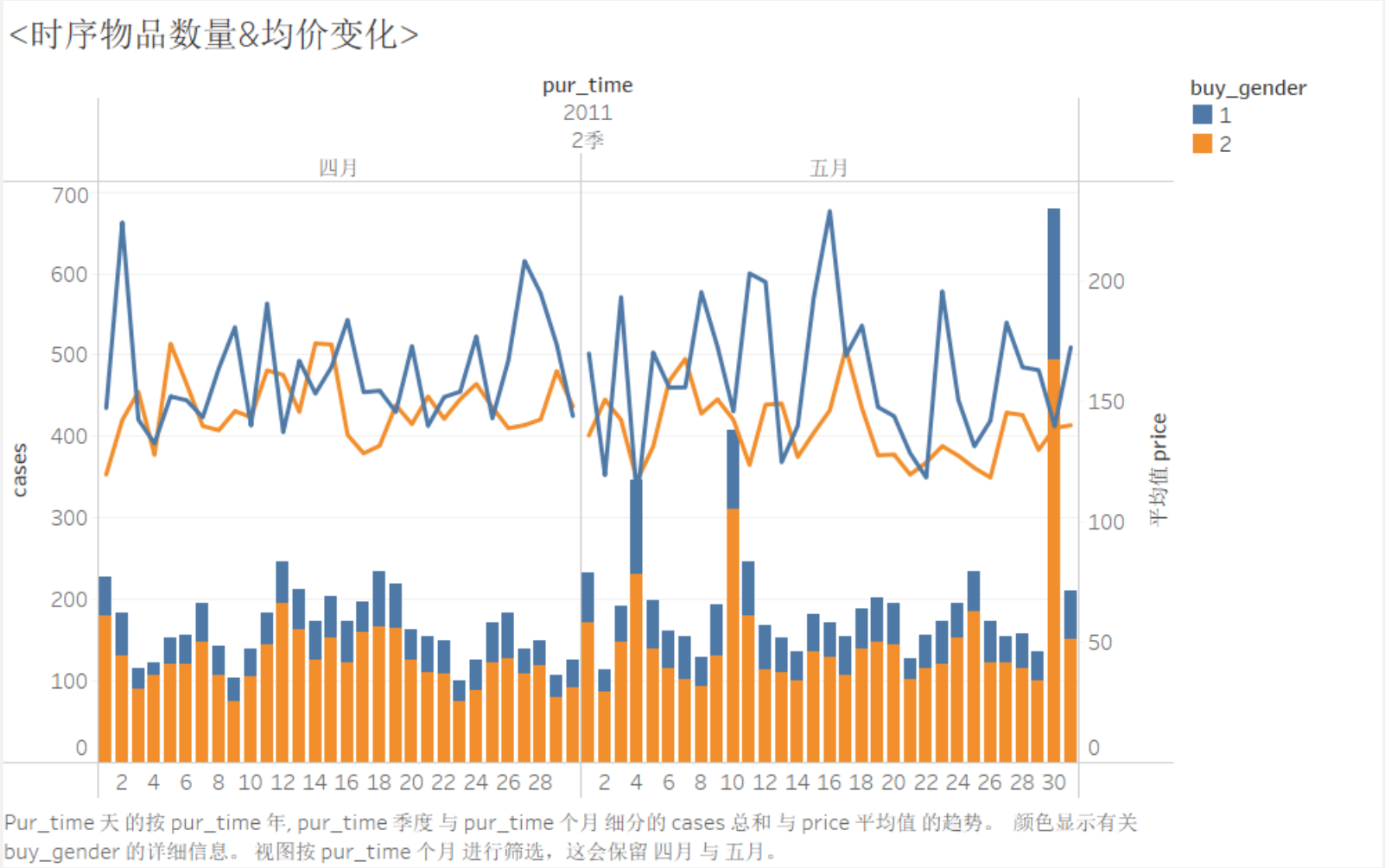
从图可知，女性信用得分总体高于男性买家。
且信用等级位于7-9消费次数较多。



购买数量&价格分析

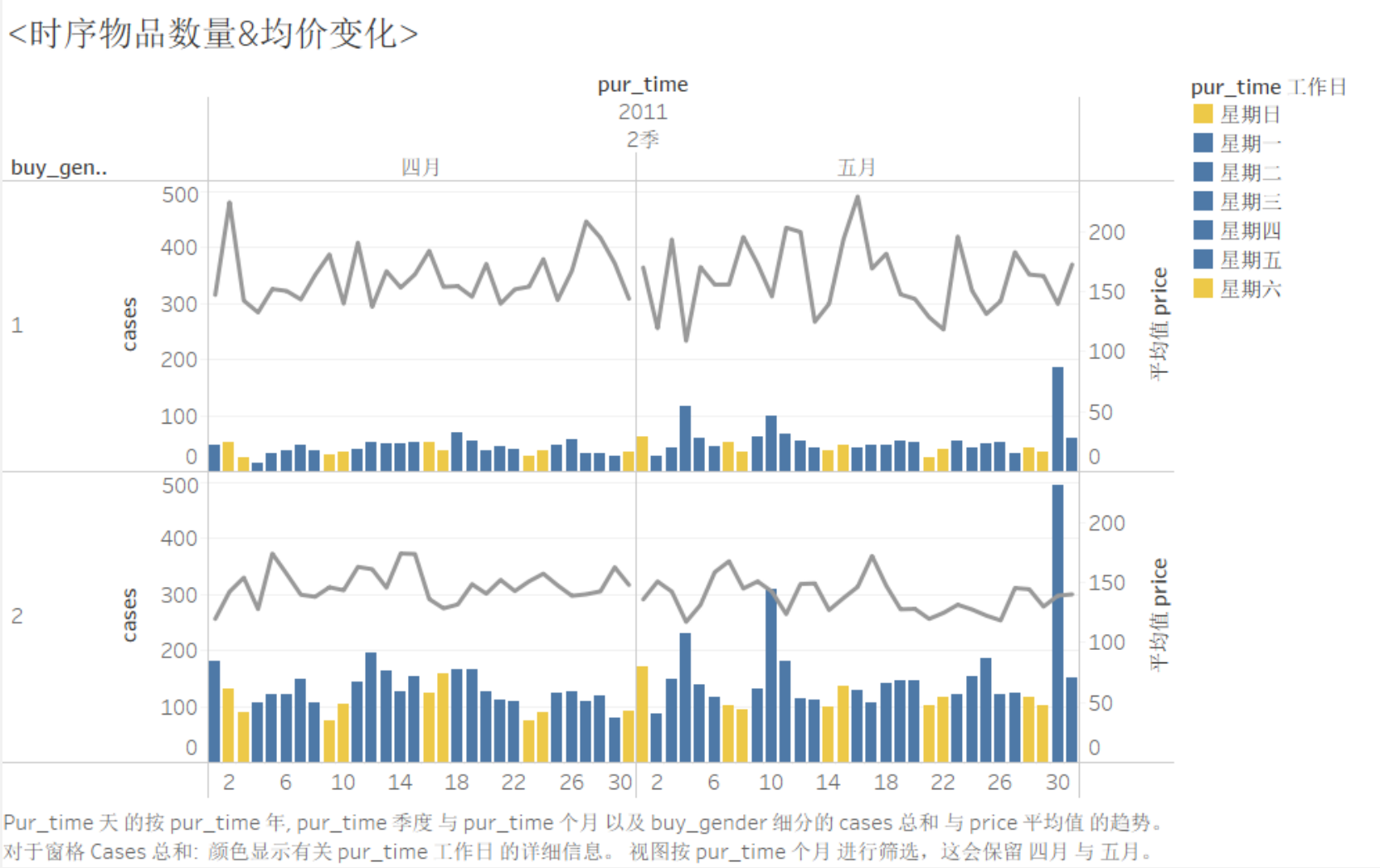
女性买家购买数量明显多于男性买家购买数量。

四月份购买数量变化总体趋于稳定而五月底购买数量波动较为明显，在月底达到峰值。



购买数量&价格分析(细分)

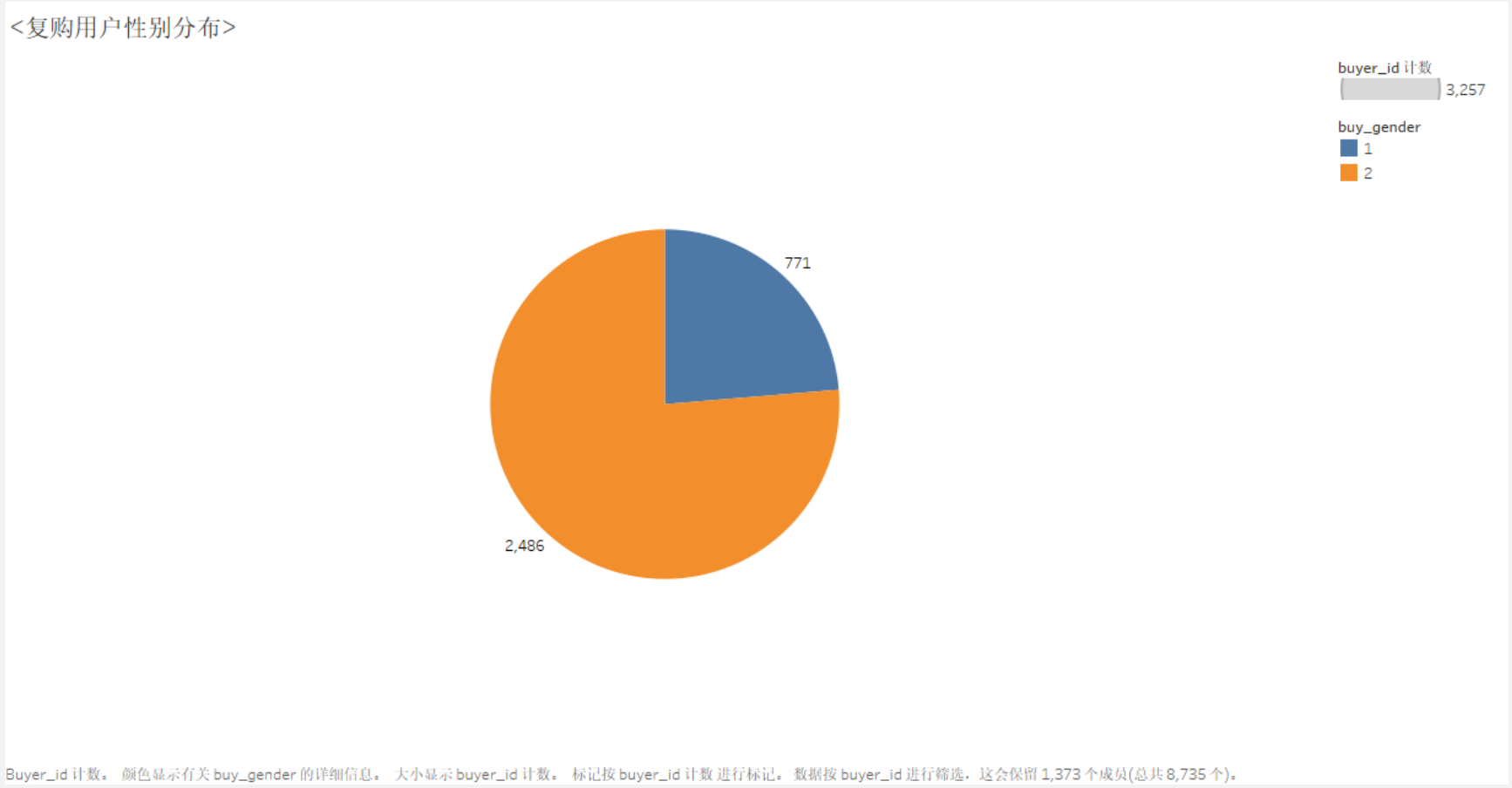
从图内可以看出，5月的波动较4月大，5月中正值母亲节与“521”等节日，因此价格较高。



复购用户分析

复购用户 性别分布

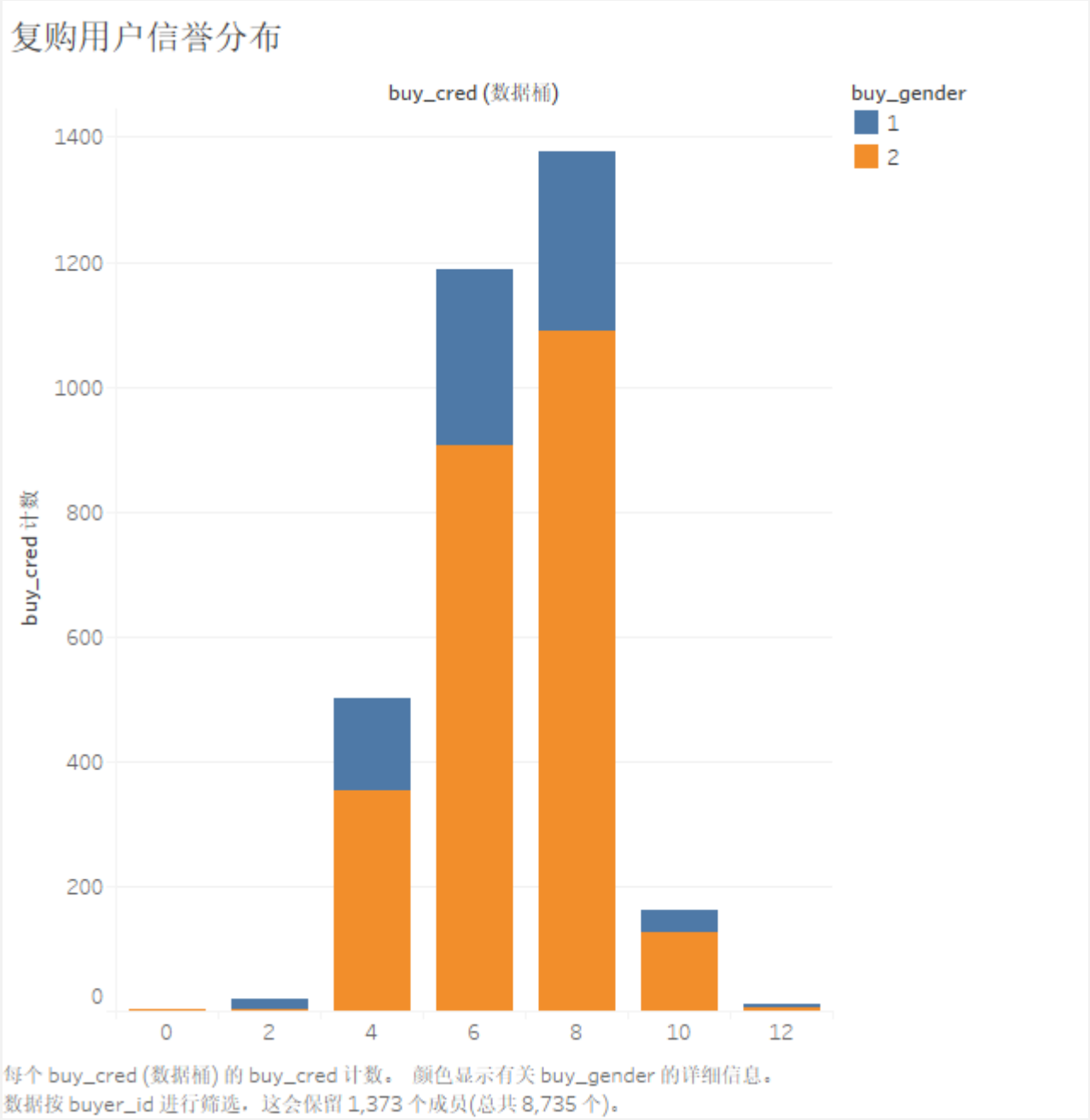
在复购用户中，女性占比高达76%。
而男性占比仅为25%不到。



复购用户分析

复购用户信誉分布

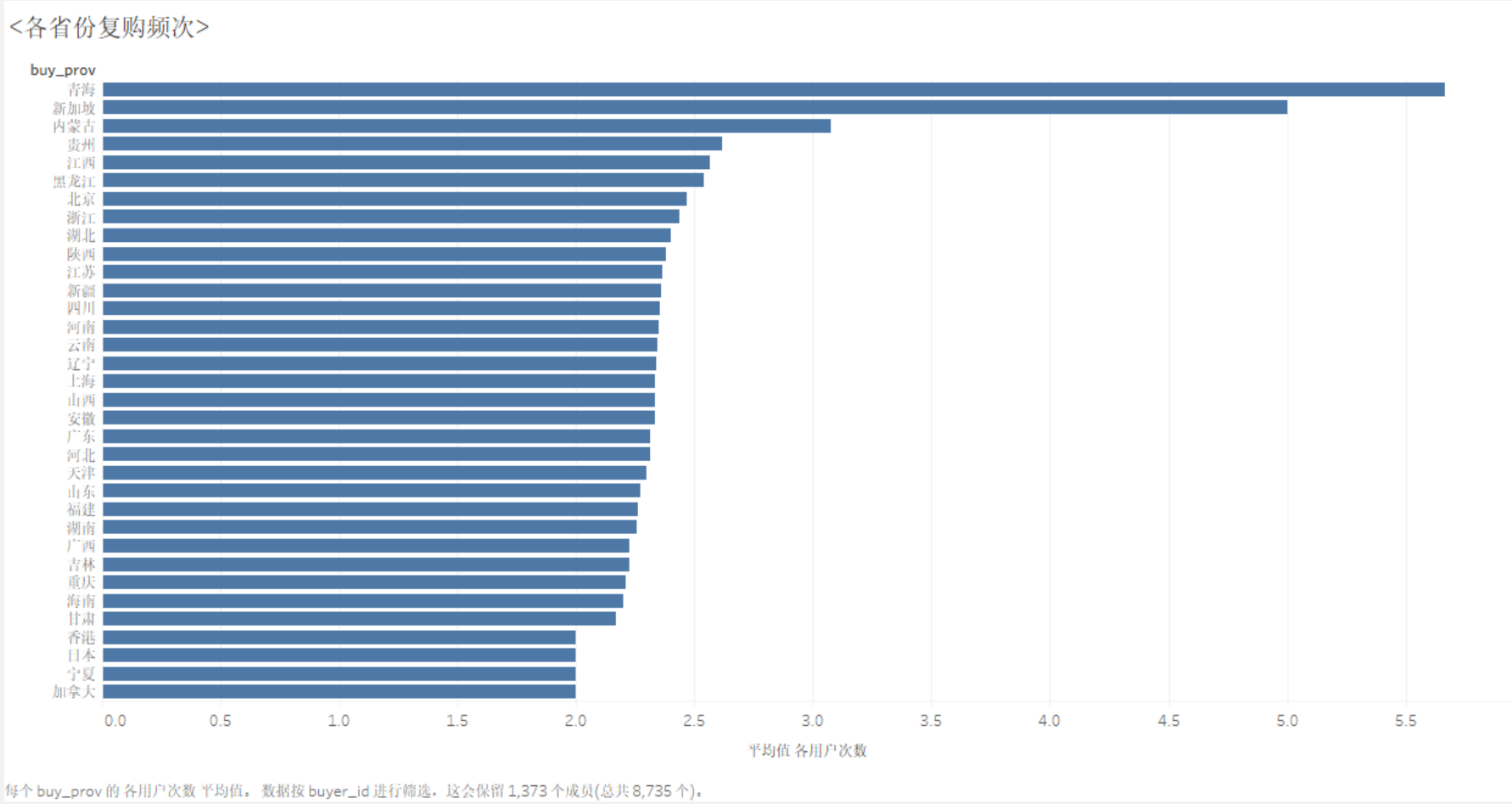
购买次数介于6次到8次之间的客户中，信用得分明显高于其他购买次数的用户。



复购用户分析

复购用户省份分布

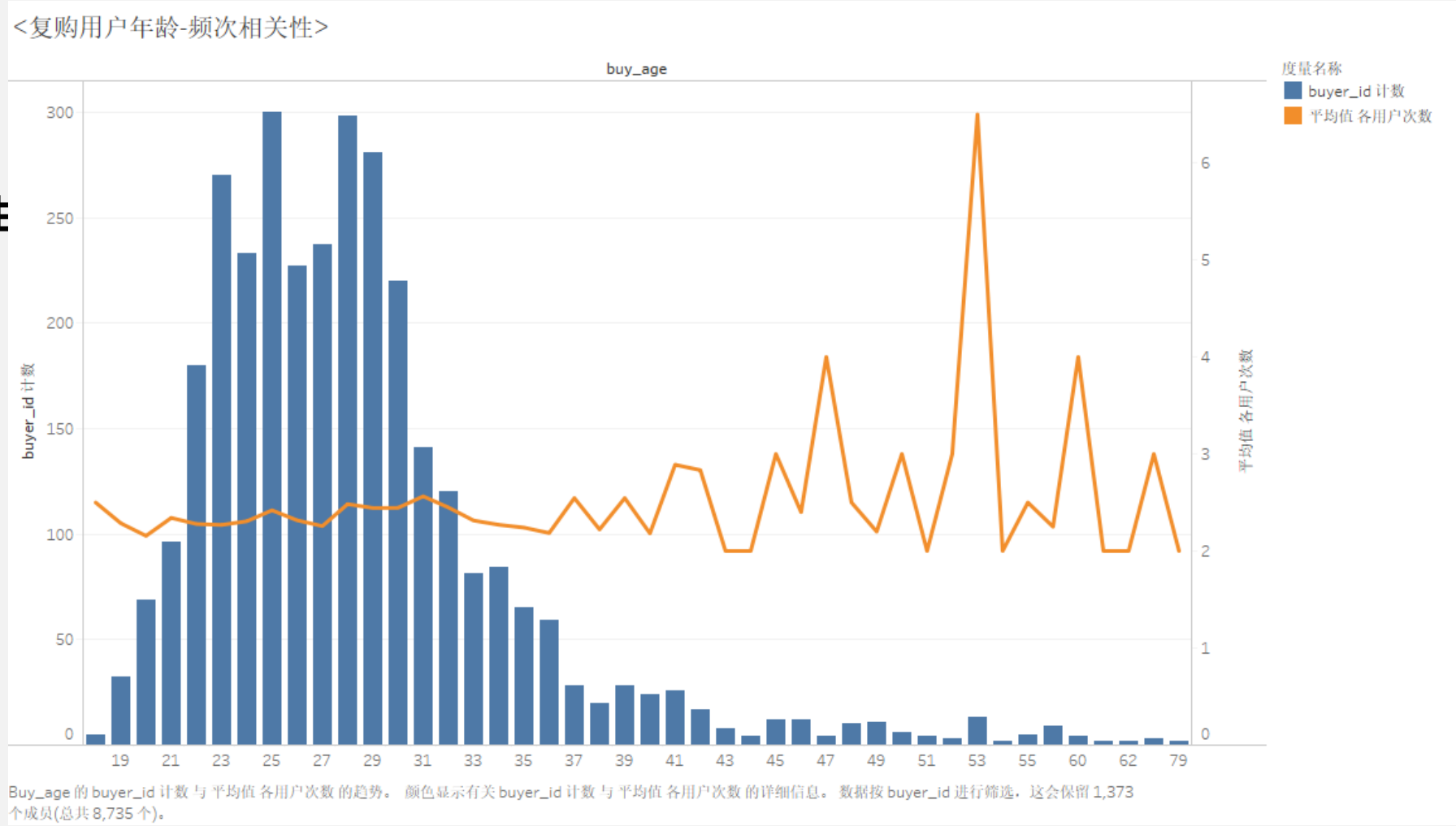
来自青海的用户平均购买次数最多，其次是新加坡，再次是内蒙区，而其他地区的用户购买次数较为平均 介于2.0-2.5次。



复购用户分析

复购用户各年龄-购买频次相关性

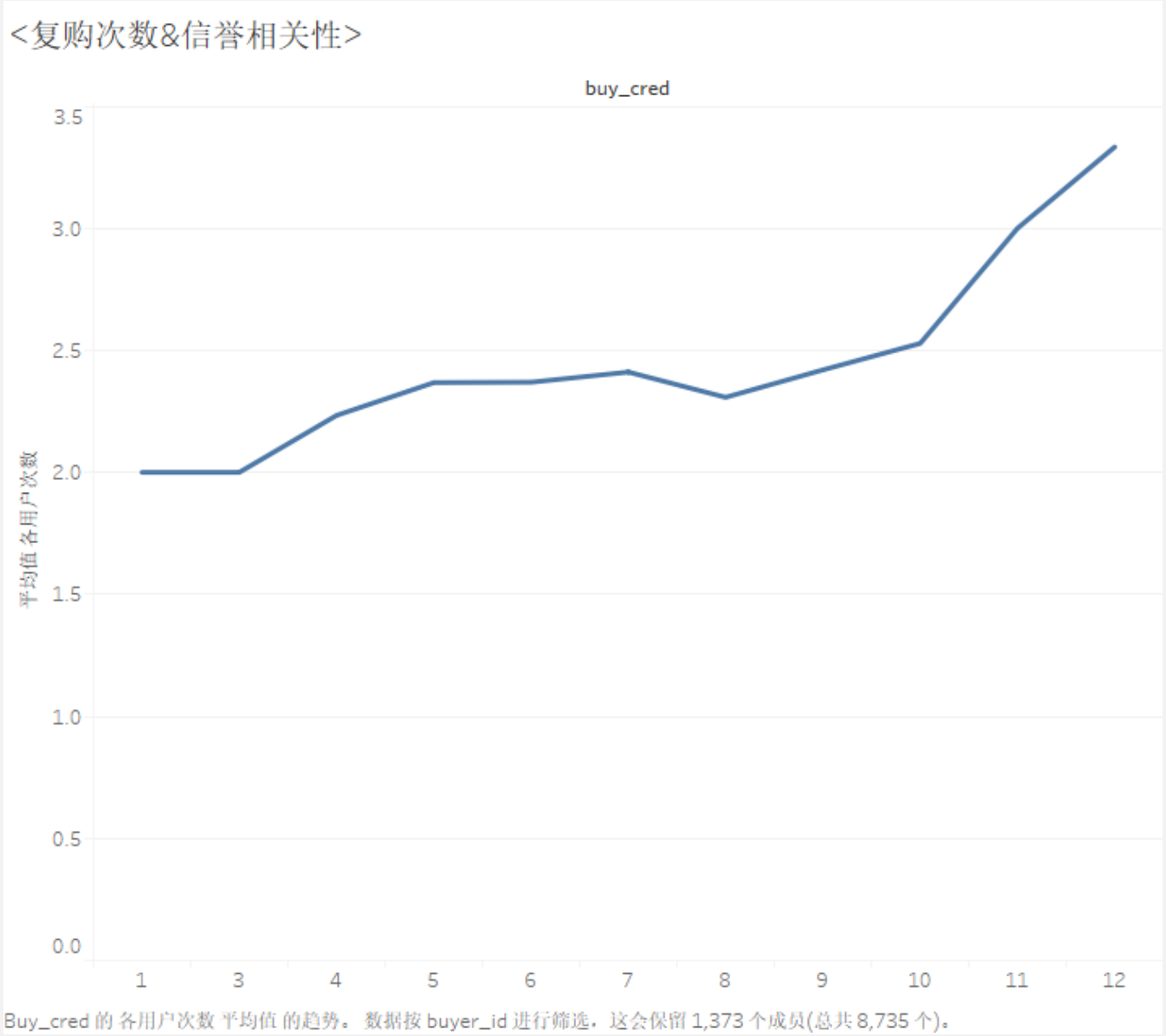
年龄介于22-32岁之间的用户平均购买次数接近2次，而其他年龄段波动较大。



复购用户分析

复购次数&信誉特征相关性

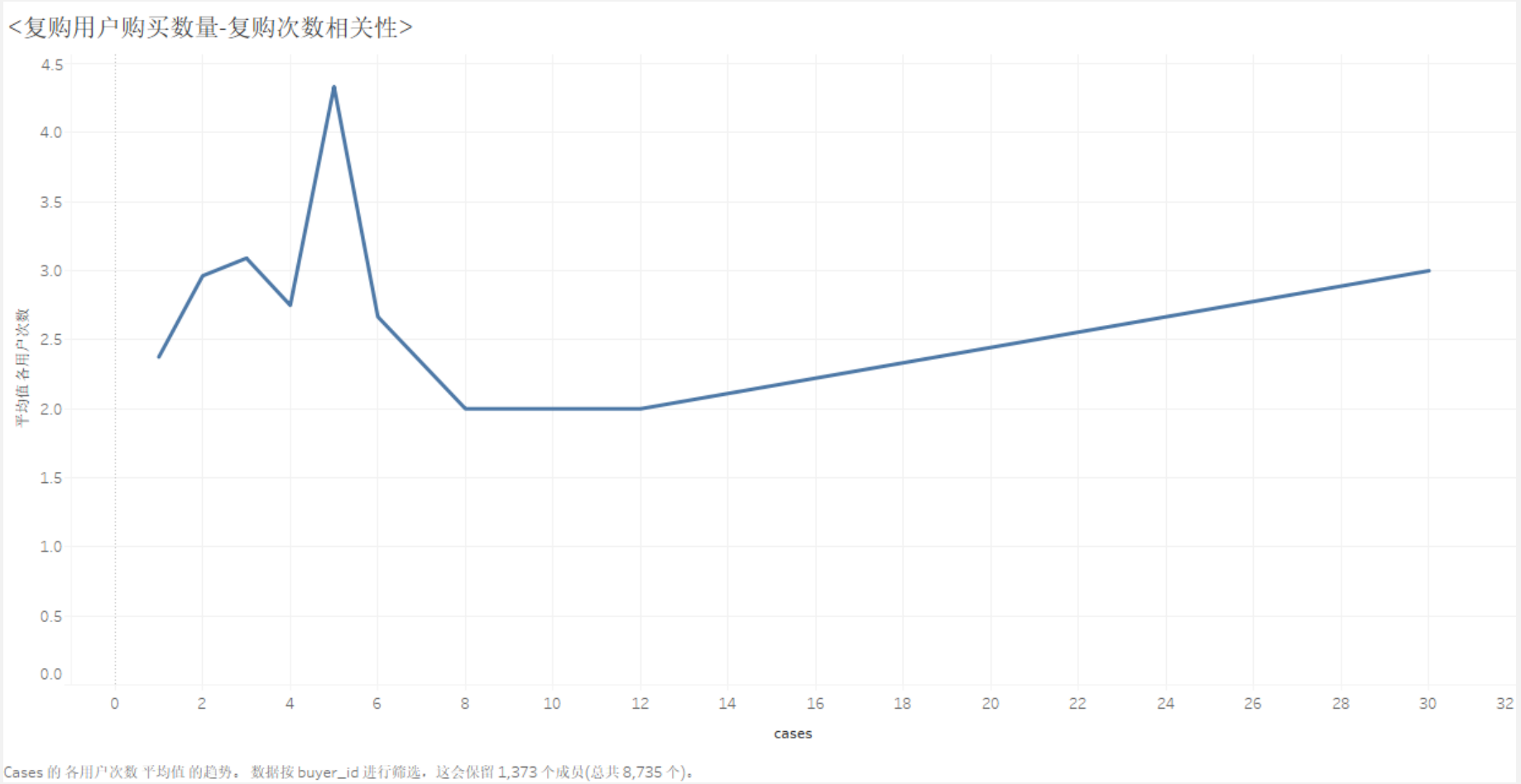
购买次数在2.0-2.5次之间的用户信用得分呈稳定增长，而购买次数在2.5以上的用户购买得分明显快速地增长。



复购用户分析

复购用户购买数量&复购次数相关性

从图而知，就整体趋势而言，购买数量越多的客户购买次数越多，但购买量为3-4及5-6出现了相反的趋势，并不符合整体规律。所以可以向购买数量多的客户多进行推销，增加其购买次数。



03

基于RFM模型分析

RFM分析过程

RFM分析

Recency

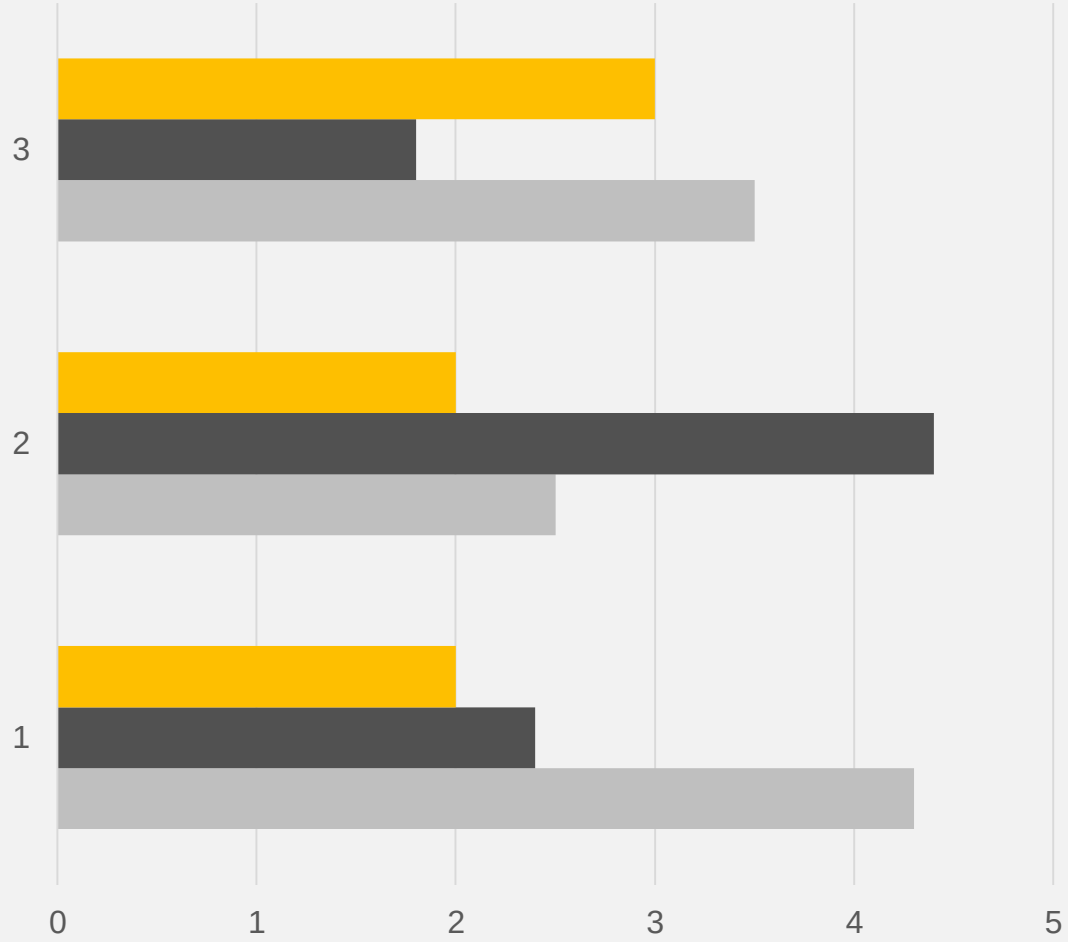
用户最近一次购买商品距离现在（或某一个阶段）的时间差—数值越大，得分越低（反比）

Frequency

用户在一段时间内购买商品的频次

Monetary

用户在某一段时间内所购买商品的总计金额



数据处理

ADD YOUR TEXE HERE ADD YOUR TEXE HERE

In [24]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import datetime
import time
```

In [25]:
```
df=pd.read_csv('C:\\Users\\Administrator\\Desktop\\Merge.csv',encoding='gb2312')
#df_frequency=pd.read_csv('C:\\Users\\admin\\Desktop\\purchasetimes.csv',encoding='gb2312')
#df=pd.merge(df,df_frequency,how='left',left_on=buyer_id,right_on=buyer_id,sort=True,suffixes=('_x', '_y'), copy=True)
```

In [26]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10619 entries, 0 to 10618
Data columns (total 13 columns):
Unnamed: 0    10619 non-null int64
buyer_id      10619 non-null int64
buy_gender    10619 non-null int64
buy_age       10619 non-null int64
buy_prov      10619 non-null object
buy_city      10619 non-null object
buy_cred      10619 non-null int64
basket_id     10619 non-null int64
goods_id      10619 non-null int64
pur_time      10619 non-null object
price         10619 non-null float64
shipcost      10619 non-null int64
cases         10619 non-null int64
dtypes: float64(1), int64(9), object(3)
memory usage: 1.1+ MB
```

In [27]:
```
df.buyer_id.nunique()
```

Out[27]:
```
8735
```

In [28]:
```
df['fmt_date']=pd.to_datetime(df['pur_time'],format='%m/%d/%Y')
df['DateDiff']=pd.to_datetime('01/01/2012',format='%m/%d/%Y')-pd.to_datetime(df['pur_time'],format='%m/%d/%Y')
df['day']=df['DateDiff'].dt.days
df['ttl_cost']=(df['price']*df['cases'])+df['shipcost']
df['buyer_id']=df['buyer_id'].astype('str')
```

过程

导入必要的包

总览使用数据

构建RFM

ADD YOUR TEXE HERE ADD YOUR TEXE HERE

In [84]:
```
R_target=df.groupby(by=['buyer_id'])['day'].agg({'Recency':np.min})
F_target=df.groupby(by=['buyer_id'])['buyer_id'].count()/12
M_target=df.groupby(by=['buyer_id'])['ttl_cost'].agg({'Monetary':np.sum})
```

```
C:\Users\Administrator\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: FutureWarning: using a dict on a Series for aggregation
is deprecated and will be removed in a future version
  """Entry point for launching an IPython kernel.
C:\Users\Administrator\Anaconda3\lib\site-packages\ipykernel_launcher.py:3: FutureWarning: using a dict on a Series for aggregation
is deprecated and will be removed in a future version
  This is separate from the ipykernel package so we can avoid doing imports until
```

In [85]:
```
total = R_target.join(F_target).join(M_target)
total.rename(columns={'buyer_id':'cnt'},inplace=True)
```

In [86]:
```
bins_r=total.Recency.quantile(q=[0,0.2,0.4,0.6,0.8,1],interpolation='nearest')
bins_r[0]=0
label=[5,4,3,2,1]
R_s=pd.cut(total.Recency,bins_r,labels=label)
```

In [88]:
```
bins_f=total.cnt.quantile(q=[0,0.5,1],interpolation='nearest')
bins_f[0]=0
label=[1,2]
F_s=pd.cut(total.cnt,bins_f,labels=label)
```

In [89]:
```
bins_m=total.Monetary.quantile(q=[0,0.2,0.4,0.6,0.8,1],interpolation='nearest')
bins_m[0]=0
label=[1,2,3,4,5]
M_s=pd.cut(total.Monetary,bins_m,labels=label)
```

In [90]:
```
total['R_s']=R_s
total['F_s']=F_s
total['M_s']=M_s
total['RFM'] = R_s.astype(int) + F_s.astype(int) + M_s.astype(int)
```

In [91]:
```
bins=total.RFM.quantile(q=[0,0.25,0.5,0.75,1],interpolation='nearest')
bins[0]=0
label1=[1,2,3,4]
total['lvl']=pd.cut(total.RFM,bins,labels=label1)
total=total.reset_index()
total=pd.DataFrame(data=total,columns=total.columns)
#total.sort(['lvl','RFM'],ascending=[1,1])
```

过程

创建所需的数据字段维度

1.R-Recency

2.F-cnt

3.M-Monetary

连接3个指标至同一数据表中

数据类型转换

三大指标按排序百分比分层

聚类处理

ADD YOUR TEXE HERE ADD YOUR TEXE HERE

In [40]:
```
for i in ['R_s', 'F_s', 'M_s']:
    total[i]=total[i].astype('int')
    total[i] = (total[i]- total[i].mean(axis=0)) / total[i].std(axis=0)
total.columns = ['Z' + i for i in total.columns]
```

In [41]:
```
from sklearn.cluster import KMeans

k = 8 # 共分为5类

kmodel = KMeans(n_clusters = k, n_jobs = 4,algorithm='auto')
kmodel.fit(total)

kmodel.cluster_centers_
kmodel.labels_
```

Out[41]:
```
array([2, 7, 7, ..., 4, 4, 7])
```

In [43]:
```
kmeansCenters = pd.DataFrame(kmodel.cluster_centers_, columns = total.columns)
labelsCounts = pd.DataFrame(kmodel.labels_)[0].value_counts()
kmeansLabels = pd.DataFrame(labelsCounts, index = None)
kmeansLabels.columns = ['Num']
kmeansResult = pd.concat([kmeansCenters, kmeansLabels], axis=1)
kmeansResult['Class'] = [1,2,3,4,5,6,7,8]
kmeansResult = kmeansResult[['Class','Num', 'Zbuyer_id', 'ZRecency', 'Zcnt', 'ZMonetary', 'ZR_s', 'ZF_s', 'ZM_s', 'ZRFM',
       'Zlvl']]
```

In [44]:
```
kmeansResult
```

Out[44]:

	Class	Num	Zbuyer_id	ZRecency	Zcnt	ZMonetary	ZR_s	ZF_s	ZM_s	ZRFM	Zlvl
0	1	560	3.572640e+08	242.162791	0.099583	180.002683	-0.043408	-0.023913	-0.040017	6.973166	2.275492
1	2	573	7.842786e+08	243.040140	0.102240	193.514991	0.008680	0.004477	-0.085702	6.993019	2.265271
2	3	2747	2.511881e+07	243.116491	0.101990	251.328173	0.012391	0.022219	0.030443	7.168912	2.337459
3	4	556	6.402324e+08	241.814414	0.099700	188.635622	-0.057869	-0.045724	-0.032733	6.954955	2.248649
4	5	587	9.271899e+08	243.350340	0.100624	175.983180	0.011604	-0.020670	-0.109718	6.954082	2.255102
5	6	540	2.154909e+08	242.112963	0.098302	194.563352	-0.057693	-0.085874	-0.032627	6.940741	2.246296
6	7	553	4.976005e+08	242.064982	0.100632	313.261516	-0.050412	0.004563	0.056588	7.110108	2.332130
7	8	2619	7.866694e+07	243.318442	0.102011	202.332440	0.026586	0.011891	0.021687	7.172967	2.352043

过程

将RFM三指标按z进行标准化
使用Kmeans聚类（默认分为8类）
按聚类后8类RFM值大小进行用户类型匹配

聚类处理

ADD YOUR TEXE HERE ADD YOUR TEXE HERE

In [40]:
```
for i in ['R_s', 'F_s', 'M_s']:
    total[i]=total[i].astype('int')
    total[i] = (total[i]- total[i].mean(axis=0)) / total[i].std(axis=0)
total.columns = ['Z' + i for i in total.columns]
```

In [41]:
```
from sklearn.cluster import KMeans

k = 8 # 共分为5类

kmodel = KMeans(n_clusters = k, n_jobs = 4,algorithm='auto')
kmodel.fit(total)

kmodel.cluster_centers_
kmodel.labels_
```

Out[41]:
```
array([2, 7, 7, ..., 4, 4, 7])
```

In [43]:
```
kmeansCenters = pd.DataFrame(kmodel.cluster_centers_, columns = total.columns)
labelsCounts = pd.DataFrame(kmodel.labels_)[0].value_counts()
kmeansLabels = pd.DataFrame(labelsCounts, index = None)
kmeansLabels.columns = ['Num']
kmeansResult = pd.concat([kmeansCenters, kmeansLabels], axis=1)
kmeansResult['Class'] = [1,2,3,4,5,6,7,8]
kmeansResult = kmeansResult[['Class','Num', 'Zbuyer_id', 'ZRecency', 'Zcnt', 'ZMonetary', 'ZR_s', 'ZF_s', 'ZM_s', 'ZRFM',
       'Zlvl']]
```

In [44]:
```
kmeansResult
```

Out[44]:

	Class	Num	Zbuyer_id	ZRecency	Zcnt	ZMonetary	ZR_s	ZF_s	ZM_s	ZRFM	Zlvl
0	1	560	3.572640e+08	242.162791	0.099583	180.002683	-0.043408	-0.023913	-0.040017	6.973166	2.275492
1	2	573	7.842786e+08	243.040140	0.102240	193.514991	0.008680	0.004477	-0.085702	6.993019	2.265271
2	3	2747	2.511881e+07	243.116491	0.101990	251.328173	0.012391	0.022219	0.030443	7.168912	2.337459
3	4	556	6.402324e+08	241.814414	0.099700	188.635622	-0.057869	-0.045724	-0.032733	6.954955	2.248649
4	5	587	9.271899e+08	243.350340	0.100624	175.983180	0.011604	-0.020670	-0.109718	6.954082	2.255102
5	6	540	2.154909e+08	242.112963	0.098302	194.563352	-0.057693	-0.085874	-0.032627	6.940741	2.246296
6	7	553	4.976005e+08	242.064982	0.100632	313.261516	-0.050412	0.004563	0.056588	7.110108	2.332130
7	8	2619	7.866694e+07	243.318442	0.102011	202.332440	0.026586	0.011891	0.021687	7.172967	2.352043

过程

将RFM三指标按z进行标准化
使用Kmeans聚类（默认分为8类）
按聚类后8类RFM值大小进行用户类型匹配

结论

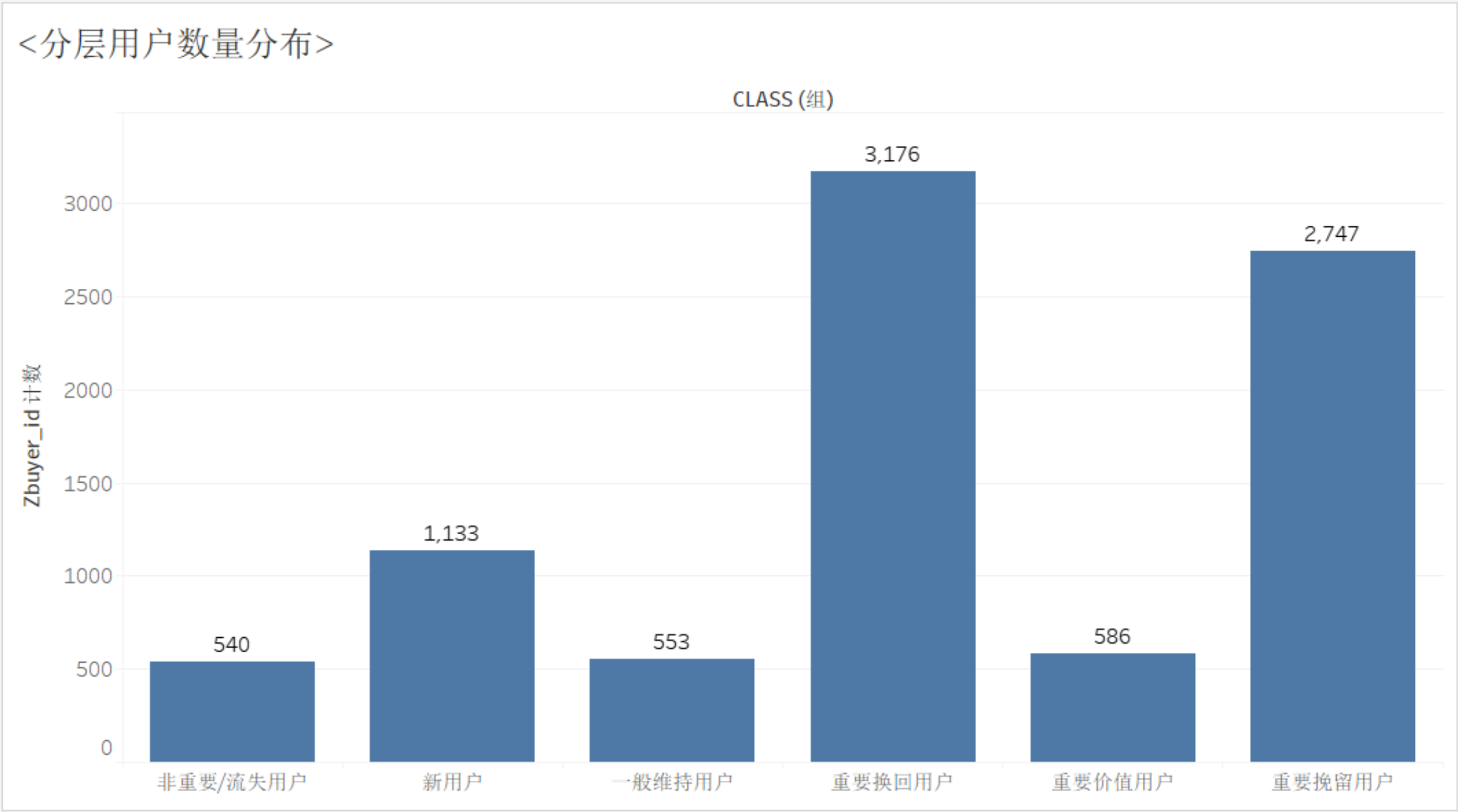
ADD YOUR TEXE HERE ADD YOUR TEXE HERE

Class	Num	Zbuyer_id	ZRecency	Zcnt	ZMonetary	ZR_s	RANK1	ZF_s	RANK2	ZM_s	RANK2	ZRFM	Zlvl
1	555	4.98E+08	242.07387	0.100601	312.55474	-0.050268	0	0.003777	1	0.05523	1	7.108108	2.331532
2	2619	7.87E+07	243.31844	0.102011	202.33244	0.026586	1	0.011891	1	0.021687	1	7.172967	2.352043
3	574	7.86E+08	243.1324	0.101336	193.59723	0.012391	1	-0.010642	1	-0.080604	0	7	2.266551
4	559	3.57E+08	242.14107	0.099554	180.55446	-0.044534	0	-0.024642	0	-0.0374	0	6.975	2.276786
5	561	6.42E+08	241.76471	0.100416	189.48672	-0.060693	0	-0.035162	0	-0.036053	0	6.950089	2.245989
6	541	2.15E+08	242.11296	0.098302	194.56335	-0.057693	0	-0.085874	0	-0.032627	1	6.940741	2.246296
7	579	9.28E+08	243.34024	0.100892	174.81539	0.012539	1	-0.014279	0	-0.114032	0	6.951641	2.255613
8	2747	2.51E+07	243.11649	0.10199	251.32817	0.012391	1	0.022219	1	0.030443	1	7.168912	2.337459

用户最终定义

Class	R	F	M	定义	策略
1	1	1	1	重要价值用户	优质客户，需要保持
2	0	1	1	重要换回客户	交易金额大，但最近无交易
3	0	1	0	一般维持客户	交易频繁，但贡献不大
4	1	0	0	新客户	近期接触且有交易，值得推广
5	1	0	0	新客户	近期接触且有交易，值得推广
6	1	0	1	重要挽留客户	潜在有价值（高频高额消费可能）
7	0	0	0	非重要/流失客户	意义不大无需太多运营动作
8	0	1	1	重要换回客户	交易金额大，但最近无交易

<分层用户数量分布>

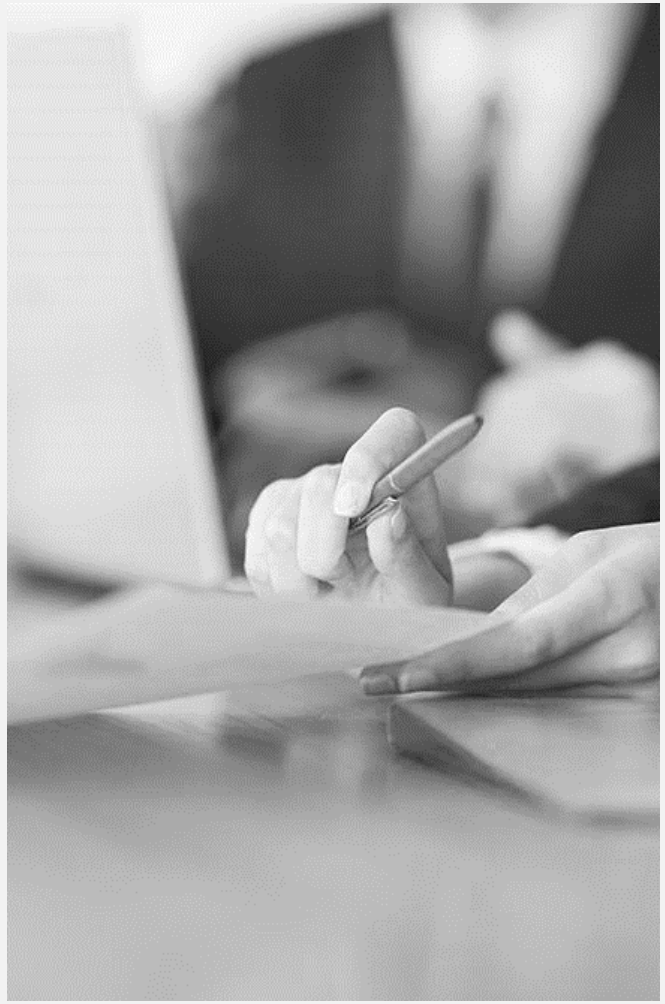

CLASS (组)
Zbuyer_id 计数
3000
2500
2000
1500
1000
500
0
540
1,133
553
3,176
586
2,747
非重要/流失用户
新用户
一般维持用户
重要换回用户
重要价值用户
重要挽留用户

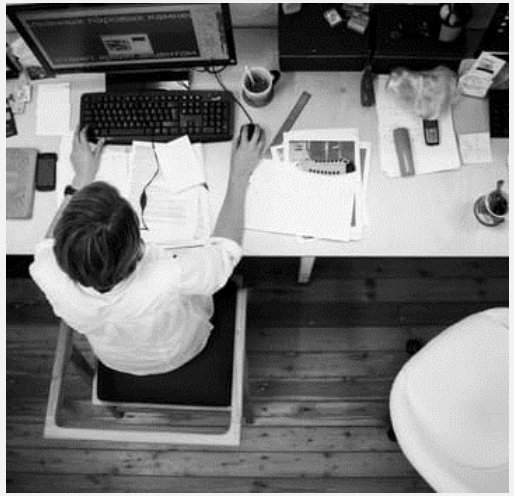
04

战略意见

整体策略方向

ADD YOUR TEXE HERE ADD YOUR TEXE HERE





重要价值

1.定向发送同以往购买记录物品类型的差异品类/搭配物品

2.第一时间发送相关领域/品类的促销信息，提高对该相关品类活动的频次与形式变化， 保证活动的多样性与吸引力

重要挽留/发展

Con1.发展：满送活动，套餐搭配等提高粘性F；

Con2.保持：适当给予特殊的VIP资格，建立忠诚度

重要换回

给予特定的限时大额高吸引优惠活动

设计客户接触频率规则

对于具有价值的客户，主动关心消费者是否有使用方面的问题，一个月后对满意度进行查询，三个月后则提供交叉销售的建议，并开始注意客户的流失可能性，不断地创造主动接触客户的机会。这样一来，客户再购买的机会也会大幅提高。

THANK YOU