互评作业1: 数据探索性分析与预处理

本次作业中，对于数据进行分析，一共分为下面三部分，也分为了3个jupyter notebook代码

1. recoginze，分析数据结构，以及简单预处理
2. single&vis，单变量分析以及可视化
3. profiling，用户画像，高价值用户分析

所有的代码都放到了DM2025/Peer1 at main · handle999/DM2025

1、数据预处理

1.1 读取数据

分析有哪些column，字段如下：

字段名	含义	类型	特征说明
id	用户唯一标识	int64	无需处理，可直接使用
last_login	最近登录时间	object (ISO时间)	时间格式需转换
user_name	用户名	object	唯一性可检查，可脱敏
fullname	用户全名	object	可拆分姓与名
email	电子邮件	object	可提取邮箱域名
age	年龄	int64	可检查异常值与分箱
income	收入	float64	可标准化或分箱处理
gender	性别	object	类别型，需标准化编码
country	国家	object	类别型，合并小众国家
address	地址	object	可选处理，如地理编码
purchase_history	购买记录（JSON）	object	需解析为结构化字段
is_active	是否活跃	bool	可直接使用
registration_date	注册时间	object (ISO时间)	时间格式需转换
phone_number	电话号码	object	可格式标准化，提取国家码
login_history	登录记录（JSON）	object	需展开提取如活跃天数、设备等特征

1.2 初步分析

这里复制了一份涉及到两个复杂嵌套结构的东西

1. purchase_history
   {"avg_price":9496,"categories":"零食","items":[{"id":7265}],"payment_method":"现金","payment_status":"已支付","purchase_date":"2023-07-30"}

- 不是，等会儿，谁家零食吃了1w块?
- 不过这不是很重要了，主要是结构，这个里面的嵌套结构有
  - avg_price, int
  - categories, str
  - items, list-dict
  - payment_method, str
  - payment_status, str
  - purchase_date, datetime(yyyy-mm-dd)

2. login_history
   {"avg_session_duration":105,"devices":["desktop","mobile"],"first_login":"2024-12-04","locations":["home","travel"],"login_count":73,"timestamps":["2024-12-04 21:29:00","2024-12-12 20:51:00","2024-12-20 19:00:00","2024-12-28 10:58:00","2025-01-05 06:58:00","2025-01-13 21:55:00","2025-01-21 18:03:00","2025-01-29 18:26:00","2025-02-06 19:31:00","2025-02-14 11:15:00","2025-02-22 06:41:00","2025-03-02 10:10:00","2025-03-10 20:17:00","2025-03-18 20:19:00"]}

- 同理，login数据似乎看起来更长一些
- 也是一个嵌套结构
  - avg_session_duration, int
  - devices, list-str
  - first_login, datetime(yyyy-mm-dd)
  - locations, list-str
  - login_count, int
  - timestamps, list-datetime(yyyy-mm-dd hh-mm-ss)

1.3 预处理

已经分析出各种数据的结构，那么就可以进行初筛，包括

- 缺失值分析
- 重复值分析
- 乱码
- 极端值
  - 年龄（<0 / >100）

至于其他的数据处理，比如异常值（性别为“武装直升机”），建议进行统计处理，比如实现一个class，统计一些类别的出现频次，然后可视化/分析。这些会在第二部分代码实现，因为已经涉及到对数据的分析处理了

理论上讲这些分析应该单独实现，但是问题是数据量实在是太大了，逐一分析会导致重复读取，浪费大量时间，因此选择实现一个汇总的代码，读取一次分析上述所有特殊情况

当然，这里不会直接进行处理，是先扫一遍，看看有没有出现上面的问题。如果有的话再决定后面怎么处理，直接删除/插值等方案都是备选

```
正在分析: part-00002.parquet
  - 总行数: 5625000
1.缺失值分析
很好，无缺失值！
这一部分用时：11.72s
2.重复值分析
重复行数: 0
这一部分用时：47.68s
3.乱码检测
这里出现了乱码:
    - purchase_history: 5625000 rows
    - login_history: 5625000 rows
这一部分用时：150.34s
4.极端值-年龄
极端年龄 (<0 or >100): 0
这一部分用时：0.03s
总计用时：209.77s
------------------------------------------------------------
```

10G的数据用了接近33min，分析过程中主要耗时在乱码检测。检测到的乱码出现在最后一行，这个事情是有点怪的，因为我看到拿到的前1000行的csv是没有问题的经过分析查证，应该和数据结构有关，最后一行会有文档结束标志，这个东西或许被识别为乱码了，而且恰好出现在两个嵌套结构中

所以以上涉及到的检测都没有问题，这份数据通过了预检测，目前看来不需要进行缺失值处理、异常值处理

```
正在分析: 30G_data_new/part-00012.parquet
  - 总行数: 8437500
1.缺失值分析
很好，无缺失值!
这一部分用时: 11.93s
2.重复值分析
重复行数: 0
这一部分用时: 67.22s
3.乱码检测
这里出现了乱码:
    - purchase_history: 8437500 rows
    - login_history: 8437500 rows
这一部分用时: 193.06s
4.极端值-年龄
极端年龄 (<0 or >100): 0
这一部分用时: 0.02s
总计用时: 272.23s
------------------------------------------------------------
```

30G的数据时间更夸张，每一个文件5min，24个文件大概用了2h上下，分析过程中主要耗时在乱码检测。检测到的乱码出现在最后一行，经过分析查证，应该和数据结构有关，最后一行会有文档结束标志，这个东西或许被识别为乱码了，而且恰好出现在两个嵌套结构中
这个事情是有点怪的，因为我看到拿到的前1000行的csv是没有问题的

所以以上涉及到的检测都没有问题，这份数据通过了预检测，目前看来不需要进行缺失值处理、异常值处理

2、可视化分析

2.1 可视化方式

2.1.1 数值字段：分箱统计

1. income（收入）
    - 分箱方式：按照 **1000为一个bin**，即 `[0-999]`，`[1000-1999]`，...直到最大值。
    - 统计内容：每个收过部分归为">N"。
    - 可视化：横轴为收入区间，纵轴为人数，使用**柱状图**或**直方图**展示分布。

---

2. purchase_history.avg_price（平均购买价格）
    - 分箱方式：按照 **100为一个bin**，即 `[0-99]`，`[100-199]`，...
    - 统计内容：每个价格区间的购买记录数量。
    - 可视化：使用**直方图**显示价格分布。

---

3. login_history.login_count（登录次数）
    - 分箱方式：按照 **1为一个bin**，即每个登录次数单独统计（如登录1次、2次...）
    - 统计内容：登录次数对应的用户数量。

- 可视化：横轴为登录次数，纵轴为用户数，使用**散柱状图**展示，适用于分析活跃度分层。

2.1.2 字符串/结构字段：提取与统计

4. phone_number（电话号码）
   - 处理方法：提取国家/地区区号（如 `+86`, `+1`, `+44` 等）。
   - 统计内容：每个区号的用户数量。
   - 可视化：**饼图**或展示常见区号。

---

5. email（邮箱）
   - 处理方法：提取域名（如 `gmail.com`, `163.com`, `qq.com`, `hotmail.com`）。
   - 统计内容：每个邮箱后缀的用量。
   - 可视化：使用**条形图**展示主流邮箱服务商使用比例。

2.1.3 时间字段：高分辨率统计

6. login_history.timestamps（登录时间）
   - 分箱方式：时间戳先转换为 datetime。再映射到：**一年 365 天 × 24 小时 = 8760 个小时**，即以小时为分箱单位。
   - 可视化：使用两张图**折线图**。

2.2 统计方法

根据上述分析，选择17个变量，统计为17 个 CSV 可视化表

2.1.1 17列数据说明

文件名（CSV）	字段含义简述	图类型	图文件格式	图形说明
`purchase_category_stats.csv`	购买的商品类别分布	横向柱状图	`.png`	类别多，适合柱状图展示
`login_first_date_stats.csv`	首次登录时间分布	折线图	`.png`	时间序列，适合折线图展示
`login_location_stats.csv`	登录地类别	饼图	`.png`	类别少
`email_domain_bins.csv`	用户邮箱的域名（如 qq.com）	饼图	`.png`	类别少
`timestamp_bins.csv`	活动或访问时间分布	折线图	`.png`	时间类字段

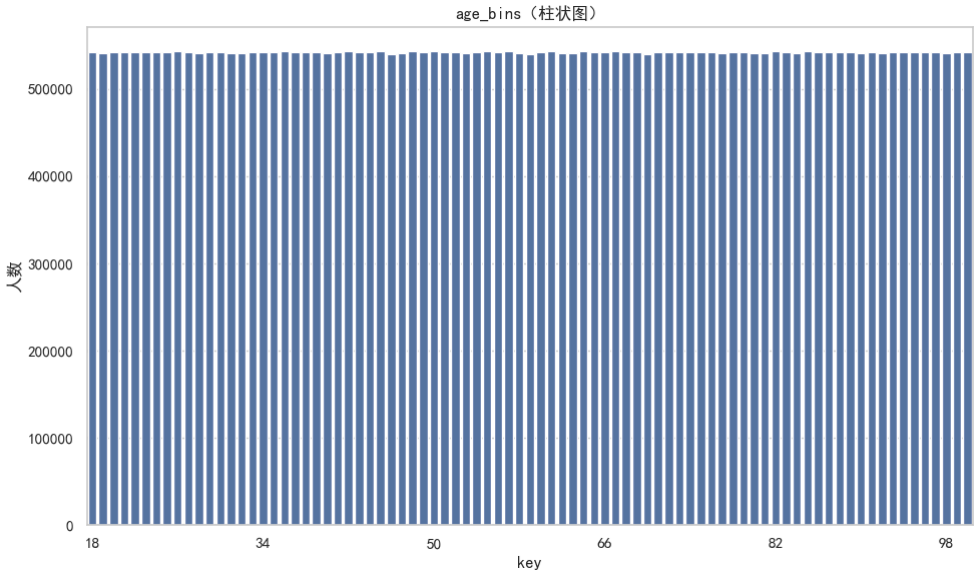
文件名（CSV）	字段含义简述	图类型	图文件格式	图形说明
`income_bins.csv`	收入分箱分布	横向柱状图	`.png`	连续值分箱，类别较多
`purchase_date_stats.csv`	购买时间分布	折线图	`.png`	时间序列
`login_device_stats.csv`	登录设备类型（手机/PC等）	饼图	`.png`	类别少，用饼图或柱状图
`country_bins.csv`	所在国家分布	地图（世界）	`.html`	适合用 Plotly 世界地图展示
`purchase_payment_method_stats.csv`	付款方式分布	饼图	`.png`	类别较少
`age_bins.csv`	年龄分箱	横向柱状图	`.png`	连续值分箱
`address_province_bins.csv`	所在省份分布（中国）	地图（省份）	`.html`	可用地图，暂用柱状图或等值地图
`purchase_status_stats.csv`	订单状态（已完成/退款等）	饼图或柱状图	`.png`	类别较少
`avg_price_bins.csv`	平均价格分箱	横向柱状图	`.png`	连续值分箱
`phone_code_bins.csv`	电话区号（如+86, +1）	横向柱状图	`.png`	类别中等偏多
`gender_bins.csv`	性别分布	饼图	`.png`	类别极少
`login_count_bins.csv`	登录次数分箱	横向柱状图	`.png`	连续值分箱

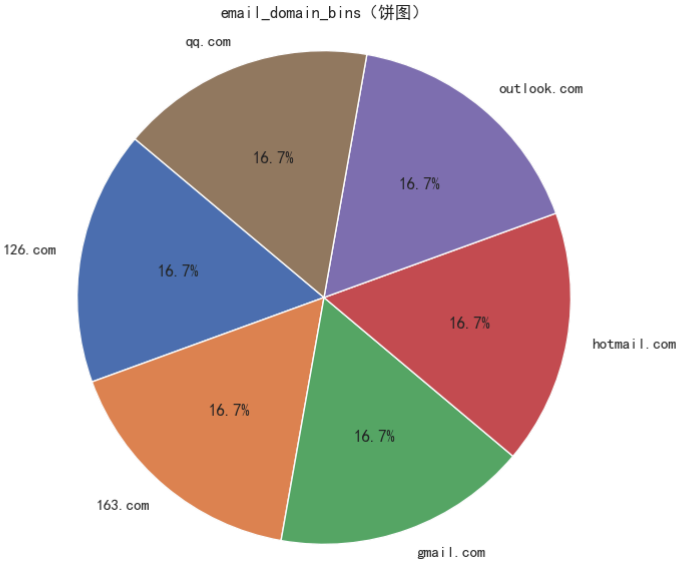
2.2.4 执行过程记录

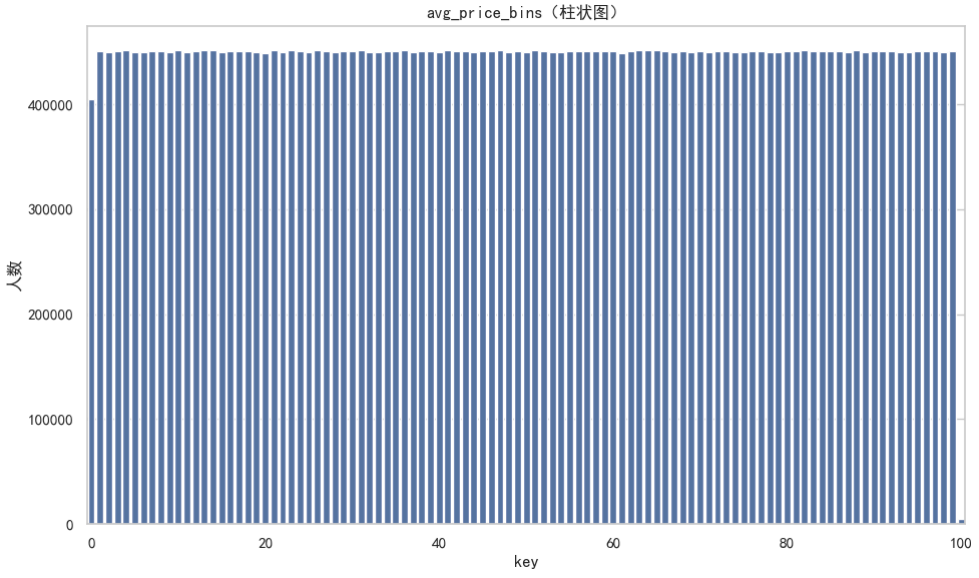
- 很明显可以看出，login_history是最费时间的，因为那里面的timestamp指不定有多长，然后我选择了逐一统计高频登录时间段hhh
- 这个里面呢在运行过程中看到预计剩余时间一直在不断乱跳，跳跃的上下波动大概在4min，说明确实不同的长度而且差的很多
- 运行时间：
    - 10G数据，一个parquet文件要1+25+2大概30min，那么8个文件大概推测需要4h
    - 30G数据，一个parquet文件要大概40min，16个文件大概需要11h
- 呃这个在jupyter notebook这种随时可能会down的情况下，感觉还是有点心惊胆战，感觉随时就爆了白跑了
- 为了防止读取太慢然后裂开，现在希望在传参的时候实现一个比如传文件夹然后整体分析，或者传文件列表只分析某个/某几个文件，然后对每个parquet文件保存所有信息到对应csv，最后再另外写一个代码汇总就好
- 合并了，大概扫了一眼，好像还是均匀分布，和之前的99%+重复率的老数据生成基本一致，但是提高了复杂度（比如login_history极其复杂，遍历一遍都爆炸）

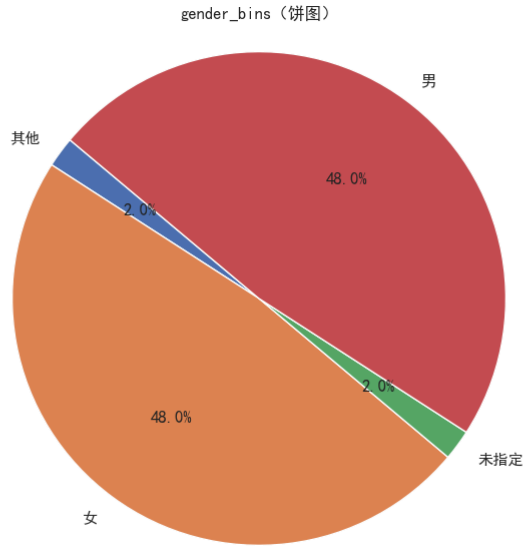
2.3 10G数据可视化结果

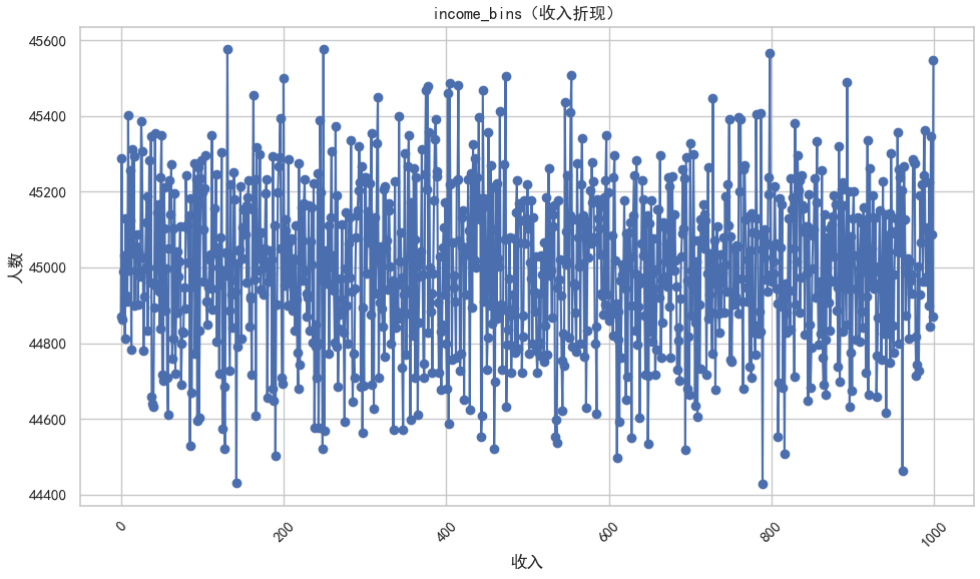
- 基本均匀分布
- 初次登录时间单调递增，是一个快速扩张并走向繁荣的网购平台
- 大概三月左右会有一个login高峰，以及每天的login高峰在18-24时（尤其是22-24），大家都是夜猫子

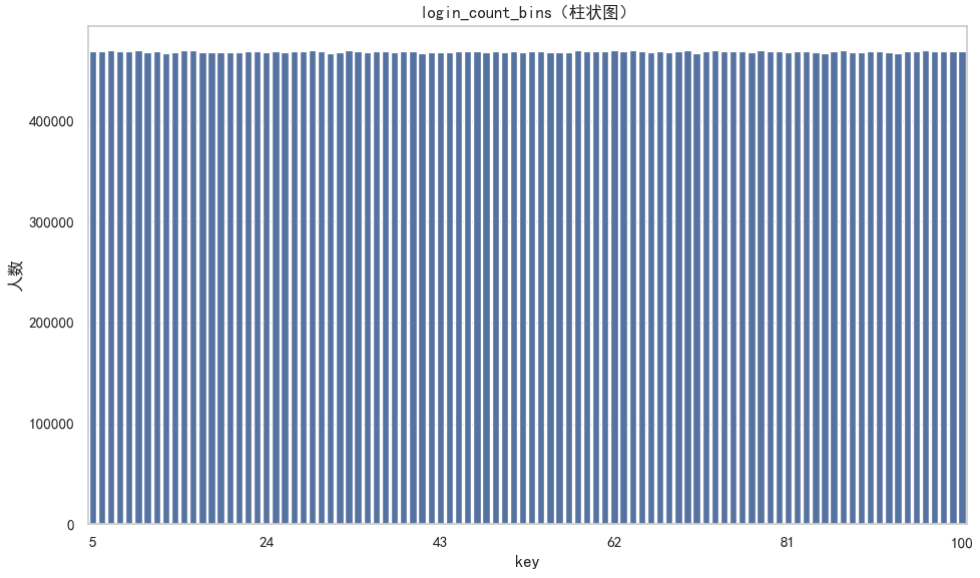

age_bins（柱状图）
500000
400000
300000
200000
100000
0
人数
18
34
50
66
82
98
key

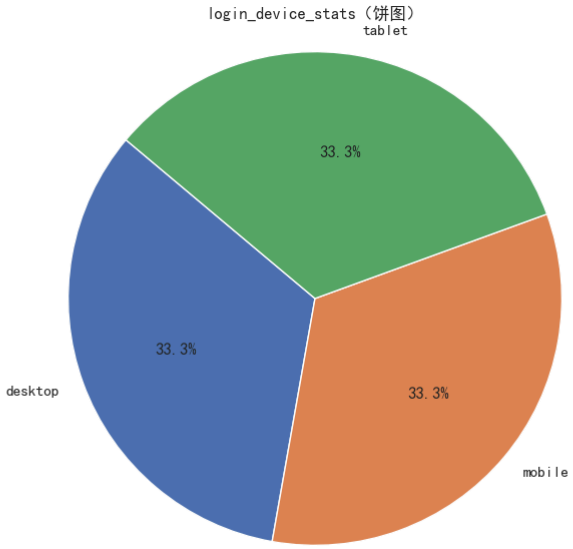

email_domain_bins（饼图）
qq.com
outlook.com
126.com
hotmail.com
163.com
gmail.com
16.7%
16.7%
16.7%
16.7%
16.7%
16.7%

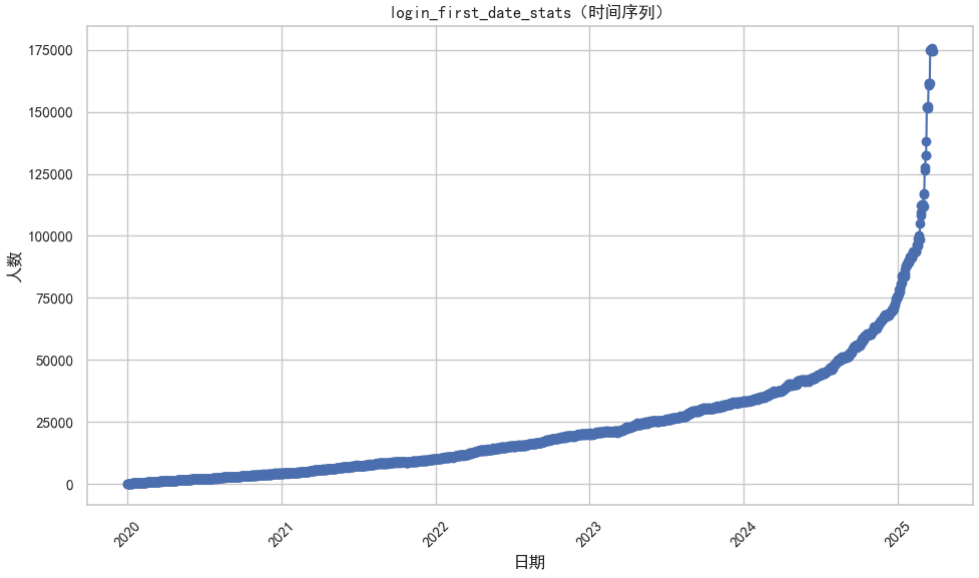

avg_price_bins（柱状图）
400000
300000
200000
100000
0
人数
0
20
40
60
80
100
key

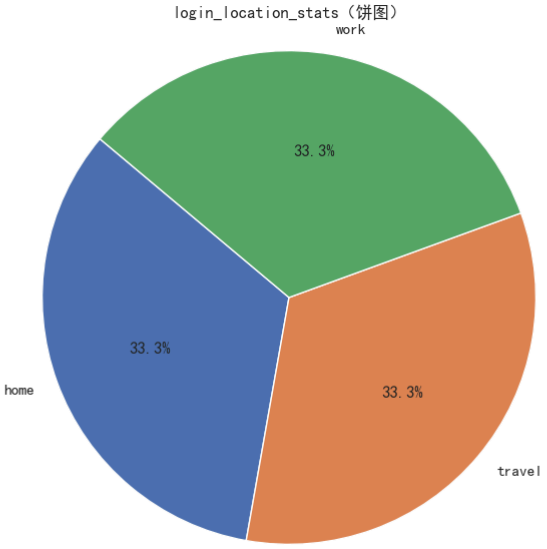

gender_bins（饼图）
男
其他
48.0%
2.0%
2.0%
48.0%
未指定
女

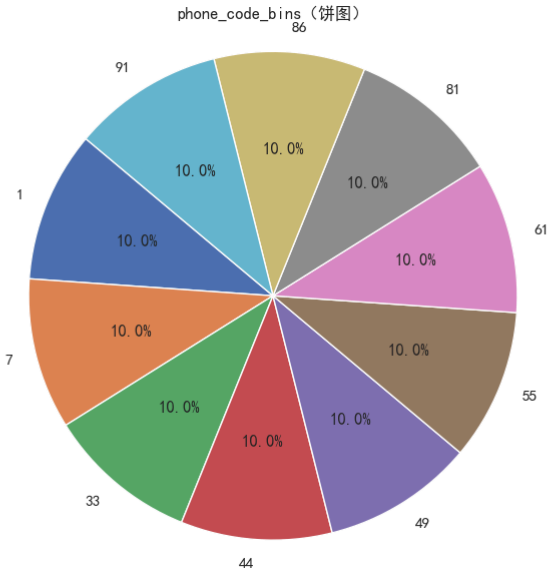

income_bins（收入折现）
45600
45400
45200
45000
44800
44600
44400
人数
0
200
400
600
800
1000
收入

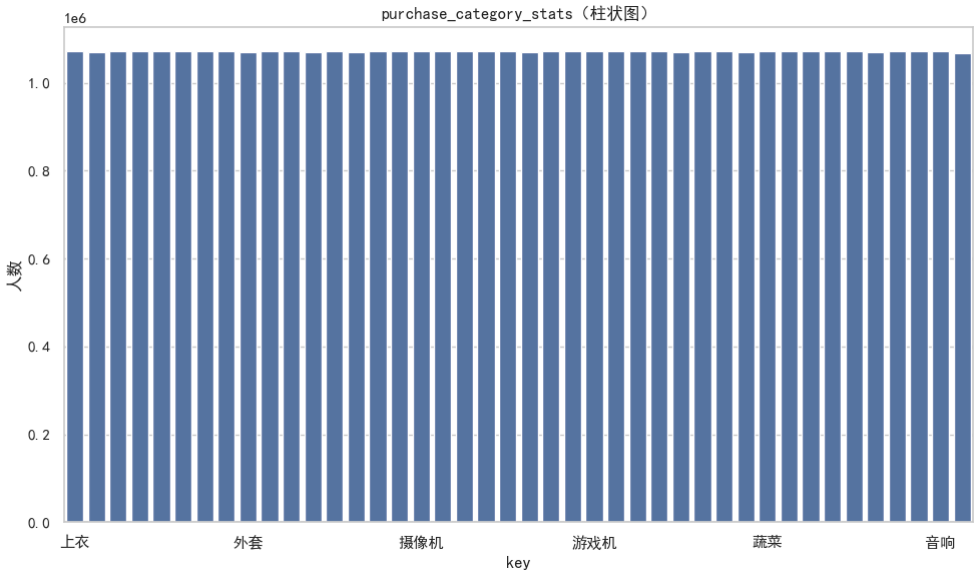

login_count_bins（柱状图）
400000
300000
200000
100000
0
人数
5
24
43
62
81
100
key

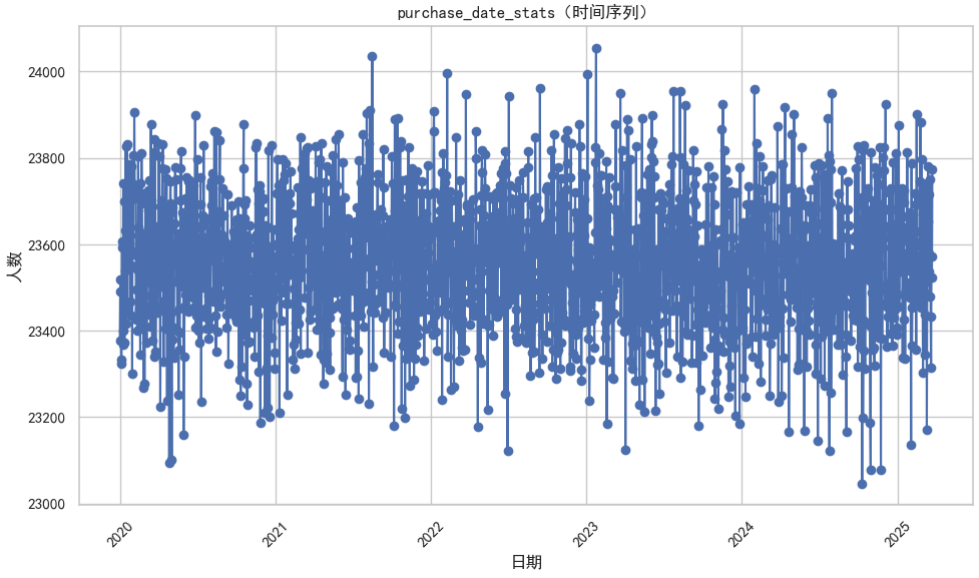

login_device_stats（饼图）
tablet
33.3%
desktop
33.3%
33.3%
mobile

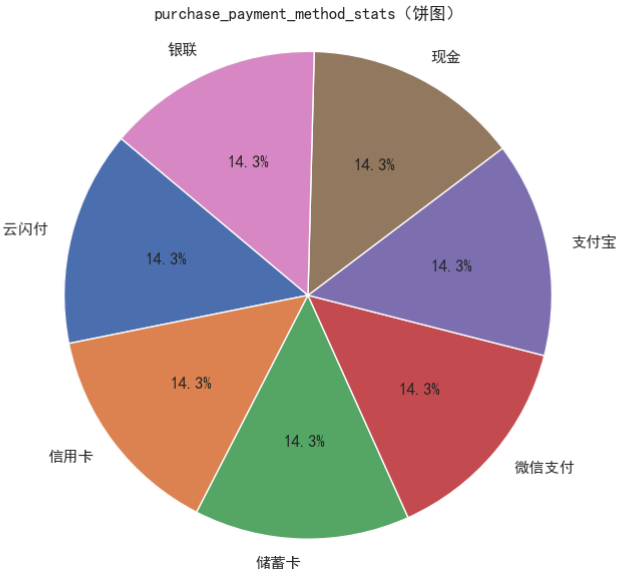

login_first_date_stats（时间序列）
175000
150000
125000
100000
75000
50000
25000
0
人数
2020
2021
2022
2023
2024
2025
日期

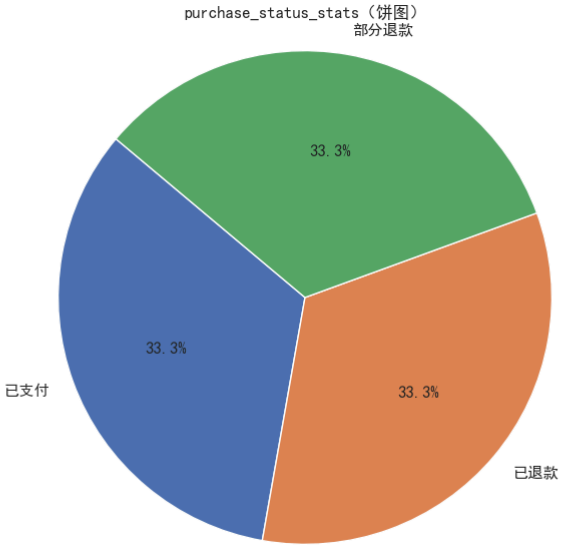

login_location_stats（饼图）
work
33.3%
home
33.3%
33.3%
travel

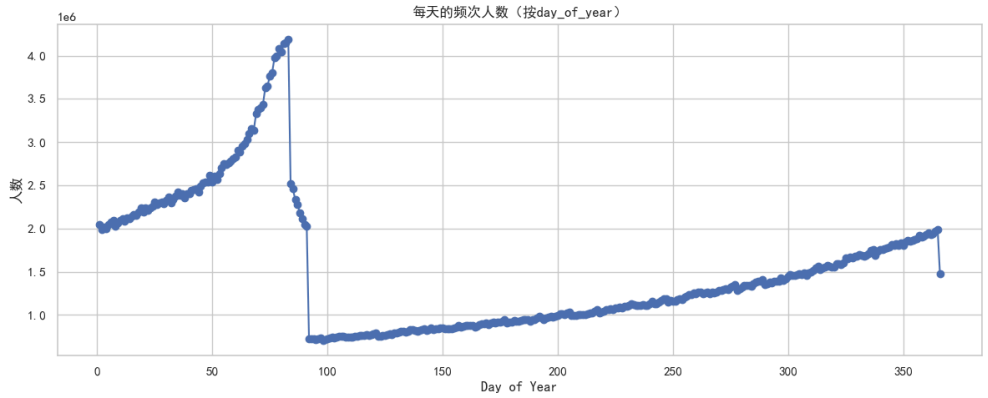

phone_code_bins（饼图）
86
81
61
55
49
44
33
7
1
91
10.0%

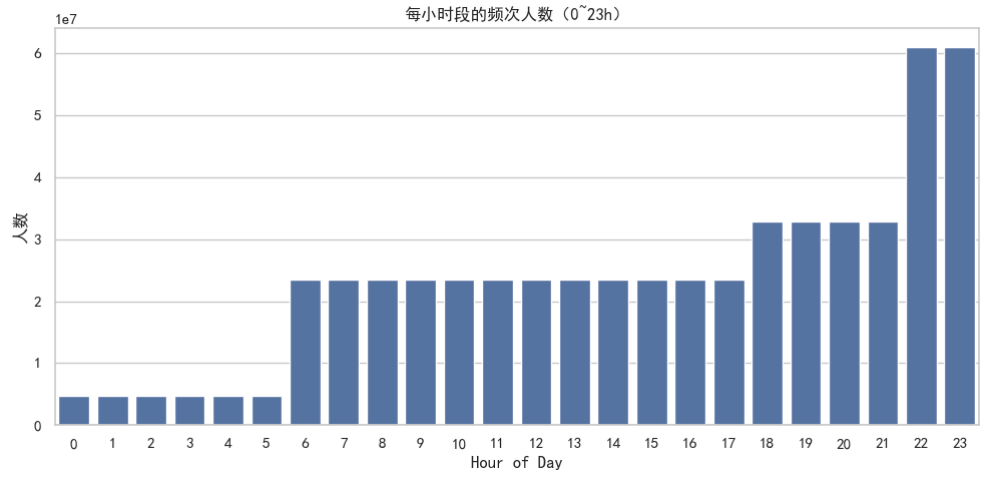

purchase_category_stats（柱状图）
1e6
1.0
0.8
0.6
0.4
0.2
0.0
人数
上衣
外套
摄像机
游戏机
蔬菜
音响
key


purchase_date_stats（时间序列）
24000
23800
23600
23400
23200
23000
人数
2020
2021
2022
2023
2024
2025
日期


purchase_payment_method_stats（饼图）
银联
现金
14. 3%
14. 3%
云闪付
14. 3%
支付宝
14. 3%
14. 3%
14. 3%
14. 3%
信用卡
微信支付
储蓄卡

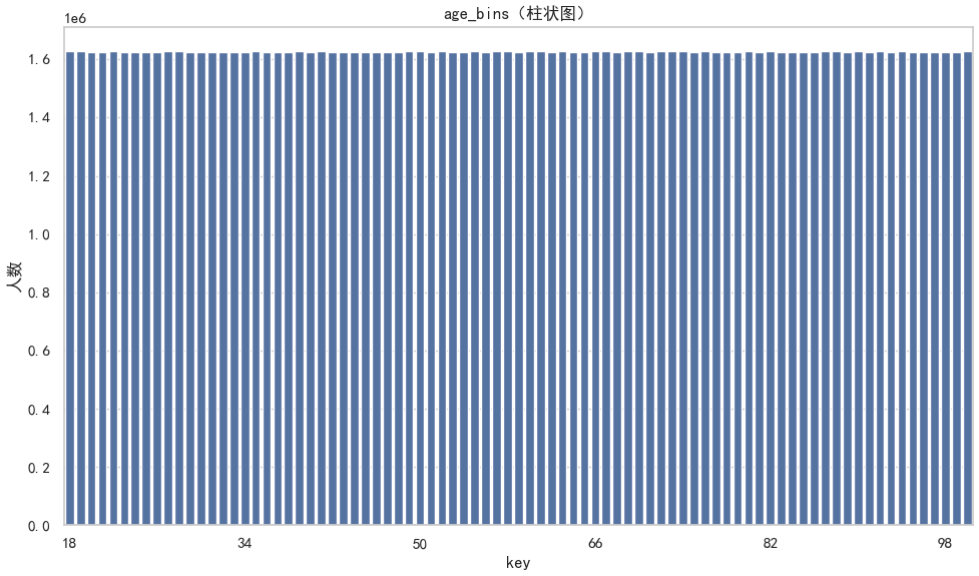

purchase_status_stats（饼图）
部分退款
33. 3%
33. 3%
已支付
33. 3%
已退款

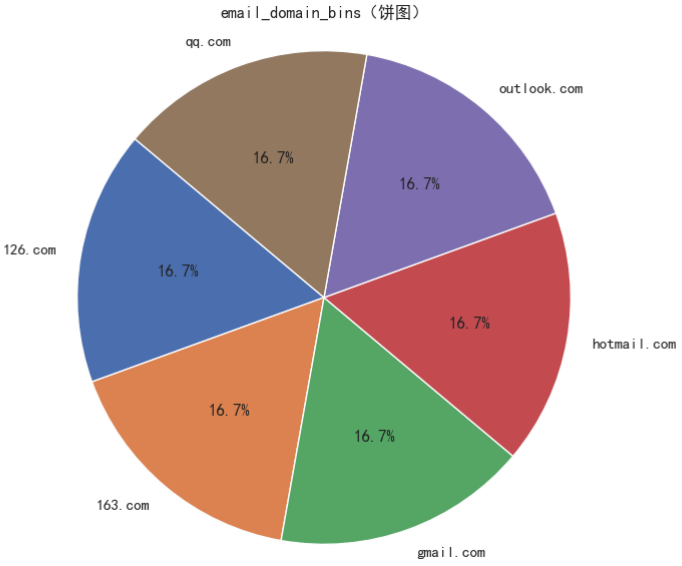


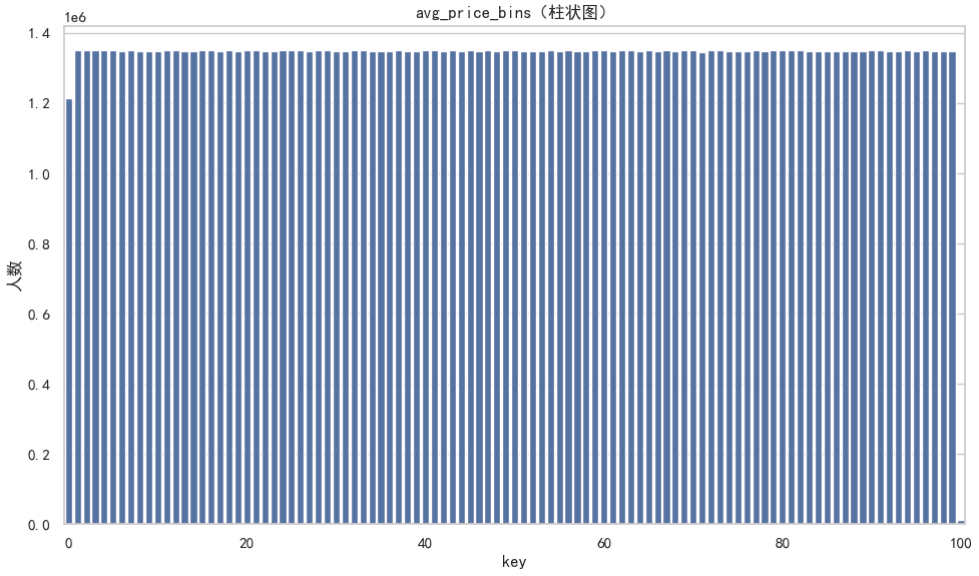


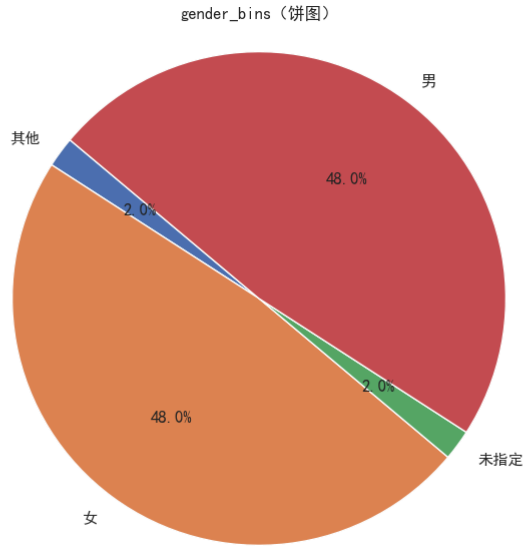
2.4 30G数据可视化结果

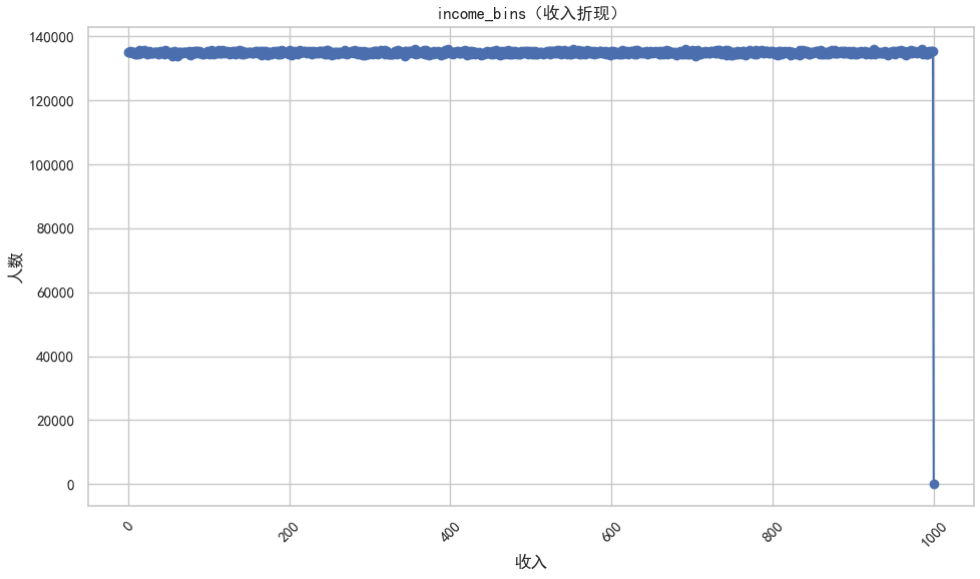
- 与10G数据分布基本一致，可以看出确实是同一份代码生成的。基本均匀分布
- 初次登录时间单调递增，是一个快速扩张并走向繁荣的网购平台
- 大概三月左右会有一个login高峰，以及每天的login高峰在18-24时（尤其是22-24），大家都是夜猫子

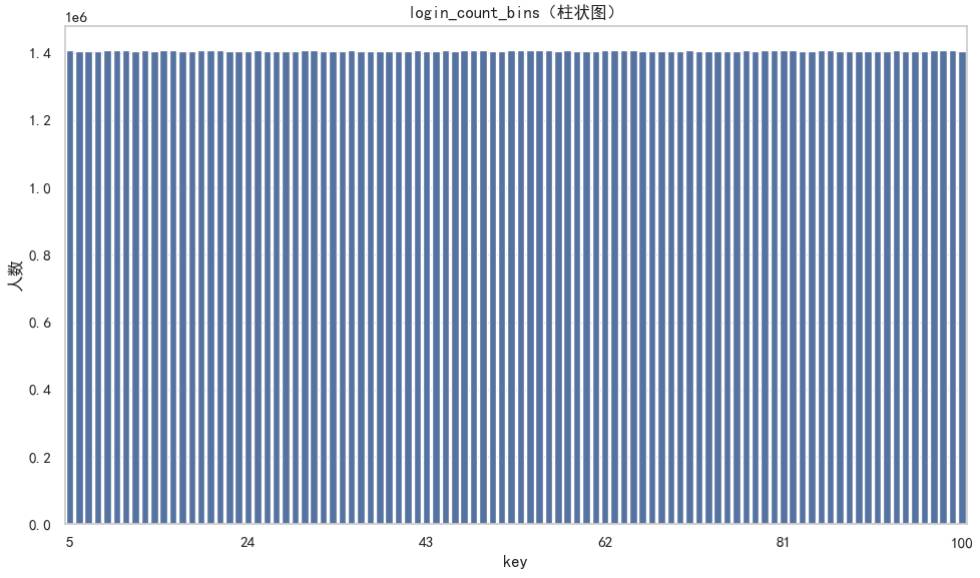

age_bins（柱状图）
1e6
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
人数
18
34
50
66
82
98
key

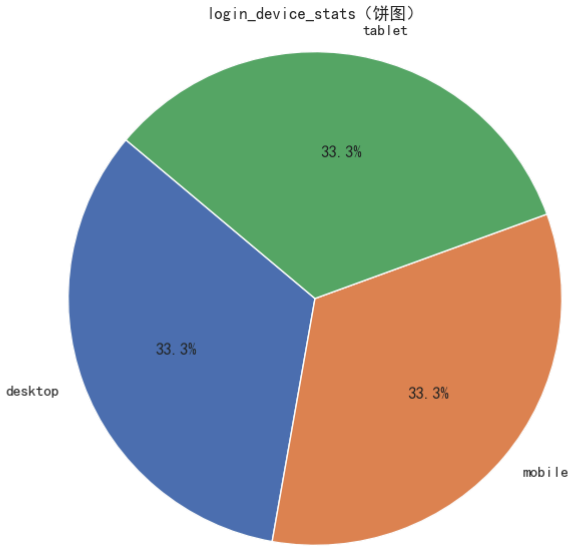

email_domain_bins（饼图）
qq.com
outlook.com
126.com
hotmail.com
163.com
gmail.com
16.7%
16.7%
16.7%
16.7%
16.7%
16.7%

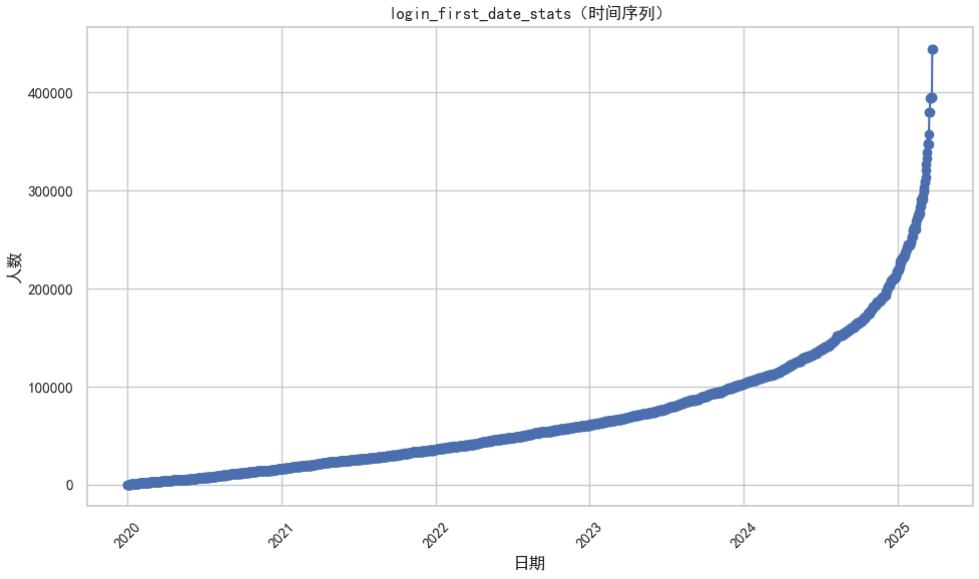

avg_price_bins（柱状图）
1e6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
人数
0
20
40
60
80
100
key

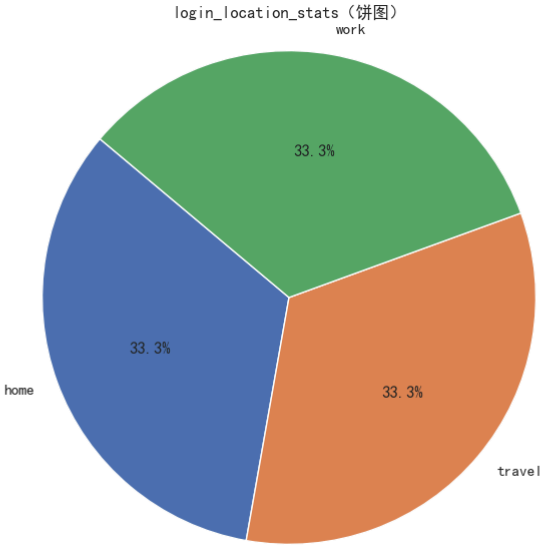

gender_bins（饼图）
男
其他
48.0%
2.0%
2.0%
48.0%
未指定
女

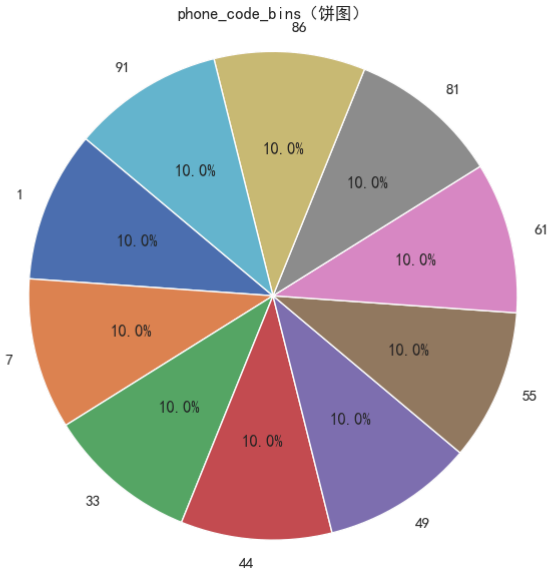

income_bins（收入折现）
140000
120000
100000
80000
60000
40000
20000
0
人数
0
200
400
600
800
1000
收入

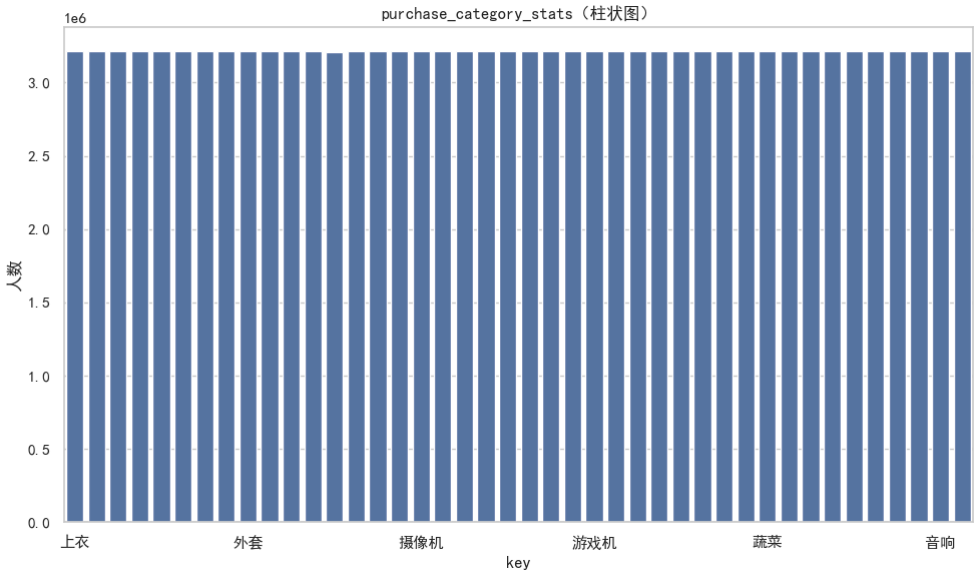

login_count_bins（柱状图）
1e6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
人数
5
24
43
62
81
100
key

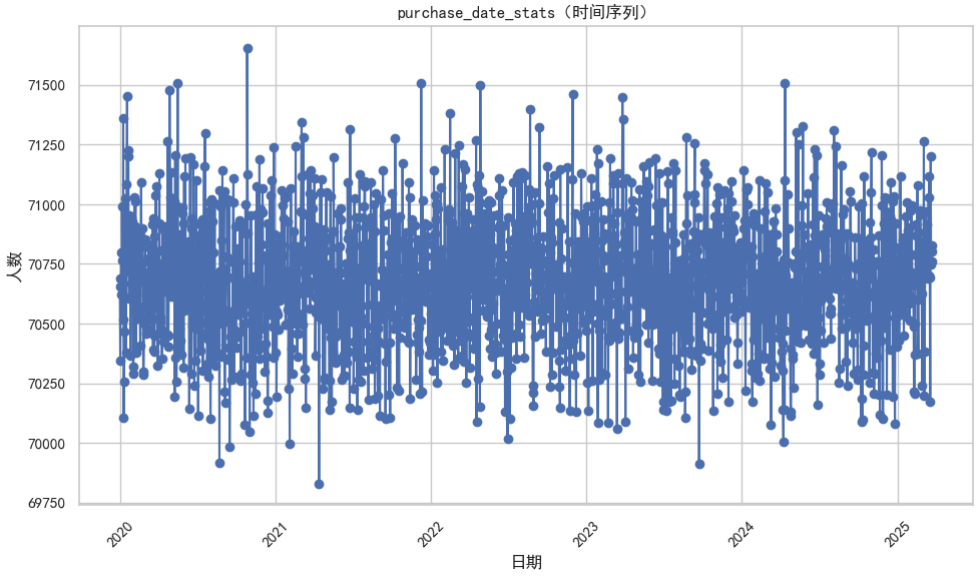

login_device_stats（饼图）
tablet
33.3%
desktop
33.3%
33.3%
mobile

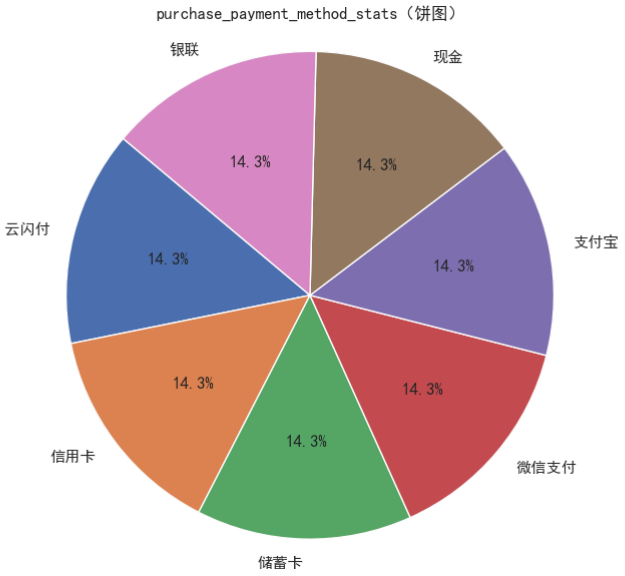

login_first_date_stats（时间序列）
400000
300000
200000
100000
0
人数
2020
2021
2022
2023
2024
2025
日期

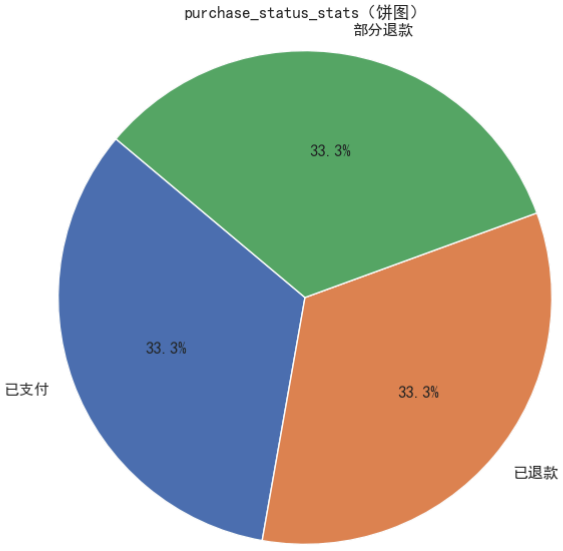

login_location_stats（饼图）
work
33.3%
home
33.3%
33.3%
travel

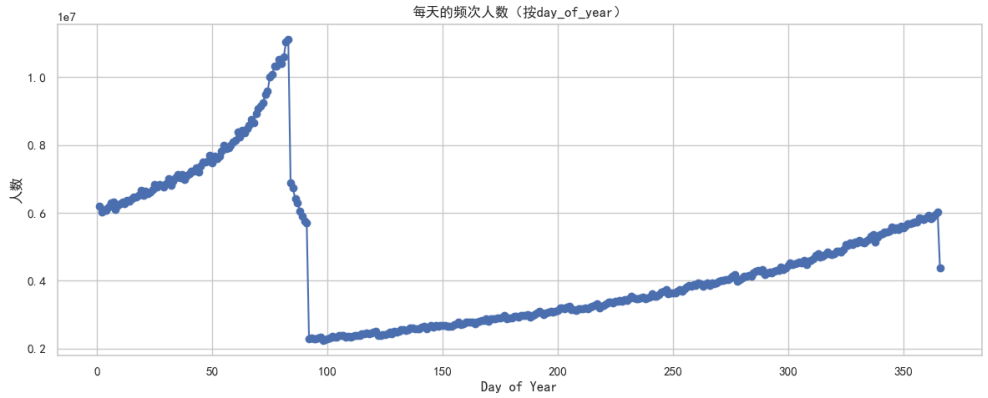

phone_code_bins（饼图）
86
81
61
55
49
44
33
7
1
91
10.0%

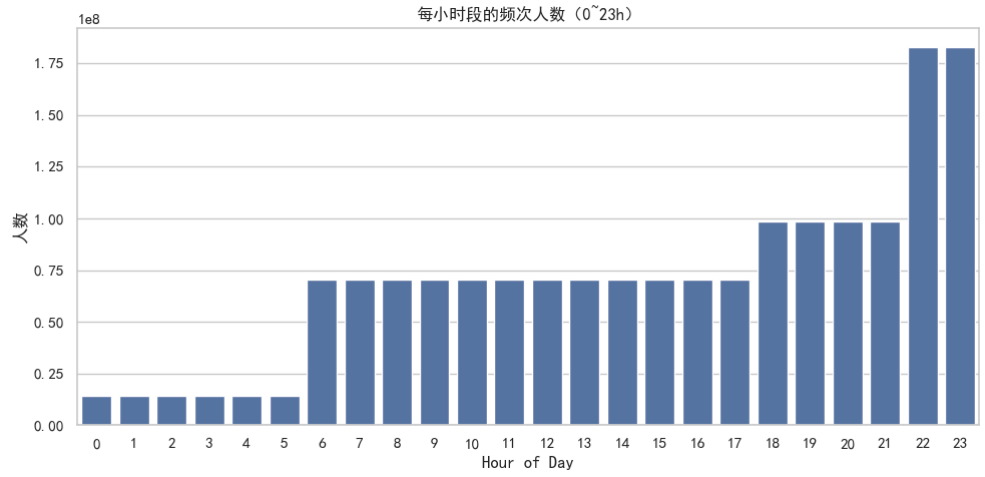

purchase_category_stats（柱状图）
1e6
3.0
2.5
2.0
1.5
1.0
0.5
0.0
人数
上衣
外套
摄像机
游戏机
蔬菜
音响
key


purchase_date_stats（时间序列）
71500
71250
71000
70750
70500
70250
70000
69750
人数
2020
2021
2022
2023
2024
2025
日期


purchase_payment_method_stats（饼图）
银联
现金
14. 3%
14. 3%
云闪付
14. 3%
支付宝
14. 3%
14. 3%
14. 3%
14. 3%
信用卡
微信支付
储蓄卡


purchase_status_stats（饼图）
部分退款
33. 3%
33. 3%
已支付
33. 3%
已退款





3、用户分析

在本节，建立用户群体画像，然后对用户群体画像进行分析，找出有用的信息列，制定打分标准，按照评分分析寻找高价值用户

3.1 用户群体画像

在上一节中，已经进行了可视化，以及17列重要变量的汇总csv。现在对着17个csv文件进行汇总，合成user_profile_summary.csv，用于汇总17列重要变量的基本统计量。

3.1.1 10G数据的基本统计量

文件名	总计	唯一键数	最多键	最多键计数	最少键	最少键计数	Top 3 占比
purchase_category_stats.csv	45000000	42	耳机	1073163	饮料	1068337	0.0715
login_first_date_stats.csv	45000000	1910	2025-03-22	175664	2020-01-01	84	0.0117
login_location_stats.csv	71663185	3	home	23889693	work	23884926	1.0
email_domain_bins.csv	45000000	6	outlook.com	7502985	163.com	7498113	0.5001
timestamp_bins.csv	562506950	8784	(83, 23)	455194	(95, 5)	5761	0.0024
income_bins.csv	45000000	1000	131	45576	789	44428	0.003
purchase_date_stats.csv	45000000	1910	2023-01-24	24054	2024-10-10	23046	0.0016
login_device_stats.csv	71660439	3	mobile	23888065	desktop	23886062	1.0
country_bins.csv	45000000	10	英国	4501669	中国	4498337	0.3001
purchase_payment_method_stats.csv	45000000	7	银联	6434053	云闪付	6425327	0.4287
age_bins.csv	45000000	83	82	543856	46	539866	0.0362

文件名	总计	唯一键数	最多键	最多键计数	最少键	最少键计数	Top 3 占比
address_province_bins.csv	45000000	35	其他	9003537	黑龙江	1001859	0.2558
purchase_status_stats.csv	45000000	3	已支付	15003797	部分退款	14996265	1.0
avg_price_bins.csv	45000000	101	51	451894	100	4537	0.0301
phone_code_bins.csv	45000000	10	44	4501669	86	4498337	0.3001
gender_bins.csv	45000000	4	男	21603397	其他	898865	0.98
login_count_bins.csv	45000000	96	33	470408	31	466853	0.0314

其中可以看到一些比较重要的信息

- login_history.login_device基本翻倍
- login_histoty.timestamps高了一个数量级
- 最多人在每年的第83天的23-24时登录，最少人在每年的第95天的5-6时登录
- 平均购买价格最低的为10000元上下，其人数比最多的5100元左右相差一个数量级

3.1.2 30G数据的基本统计量

文件名	总计	唯一键数	最多键	最多键计数	最少键	最少键计数	Top 3 占比
purchase_category_stats.csv	135000000	42	耳机	3218327	平板电脑	3210287	0.0715
login_first_date_stats.csv	135000000	1910	2025-03-23	443965	2020-01-01	63	0.0095
login_location_stats.csv	215003214	3	home	71671436	work	71663201	1.0
email_domain_bins.csv	135000000	6	qq.com	22503843	126.com	22492177	0.5001
timestamp_bins.csv	1687465900	8784	(83, 22)	1205212	(99, 3)	18406	0.0021
income_bins.csv	135000000	1001	552	135994	1000	2	0.003
purchase_date_stats.csv	135000000	1910	2020-10-26	71654	2021-04-13	69829	0.0016
login_device_stats.csv	214994008	3	tablet	71670950	mobile	71661029	1.0
country_bins.csv	135000000	10	澳大利亚	13502953	德国	13496833	0.3001
purchase_payment_method_stats.csv	135000000	7	储蓄卡	19288753	支付宝	19282785	0.4286
age_bins.csv	135000000	83	27	1629722	78	1623565	0.0362
address_province_bins.csv	135000000	35	其他	26996121	黑龙江	3010777	0.2557
purchase_status_stats.csv	135000000	3	已退款	45010827	部分退款	44987793	1.0
avg_price_bins.csv	135000000	101	62	1353810	100	13367	0.0301
phone_code_bins.csv	135000000	10	61	13502953	49	13496833	0.3001
gender_bins.csv	135000000	4	男	64810501	其他	2698372	0.98
login_count_bins.csv	135000000	96	79	1409003	36	1403542	0.0313

其中可以看到一些比较重要的信息

- login_history.login_device基本翻倍
- login_histoty.timestamps高了一个数量级
- 最多人在每年的第83天的22-23时登录，最少人在每年的第99天的3-4时登录
- 平均购买价格最低的为10000元上下，其人数比最多的6200元左右相差一个数量级

3.2 用户挖掘分析

接下来，需要根据群体画像

字段	描述	可分析行为
last_login, registration_date	时间戳	生命周期、活跃时间、冷启动用户
login_history（JSON）	登录次数、设备、时长、登录时间分布	登录频率、日活跃、设备偏好
income	收入	用户经济实力
purchase_history（JSON）	商品类别、支付、价格、时间	消费能力、购物频率、偏好模式
is_active	用户状态	用户活跃度

最终希望对于每个用户，统计这些比较关键的信息，用于刻画用户的行为

特征名	含义
login_days_count	登录天数
income	收入水平
purchase_frequency	购买频次
avg_price	平均消费金额
payment_status	用户购买支付情况

3.3 高价值用户挖掘

由于数据总量太多，所以正常用到的全量聚类分析不是很建议，现在有两种推荐的发现方法

1. 可以使用随机采样，获取一个符合数据分布的采样样本，对其聚类发掘高价值评价标准，然后遍历全集挖掘高价值用户
2. 直接使用打分机制

在这里，选择使用第2种方法，按照3.2中的分析，建立打分表

特征	条件	得分	说明
login_days	> 60	+2	活跃用户
login_days	30 ~ 60	+1	中活跃
avg_price	> 5000	+2	消费力高
avg_price	3000 ~ 5000	+1	中消费
income	>500000	+1	高收入
payment_status	已支付	+1	有效交易

特征	条件	得分	说明
payment_status	部分退款 / 已退款	-1	有争议交易

对两种数据，分别对每一个parquet执行分析，会把高价值用户的id汇总成csv。执行记录放在后面。

- 对于10G数据，对于每一个parquet，共计45,000,000数据，执行时间不到2min，大约能找到1,530,000高价值用户，大概在3.4%左右
- 对于30G数据，对于每一个parquet，共计135,000,000数据，执行时间不到3min，大约能找到2,300,000高价值用户，大概在2.2%左右

```
正在处理: 10G_data_new/part-00002.parquet
筛选高价值: 100%|████████████████████| 5625000/5625000 [01:21<00:00, 69340.00it/s]
总计1533736名高价值用户
已保存高价值用户到: 10G_data_new/part-00002.csv
用时102.91s

正在处理: 10G_data_new/part-00006.parquet
筛选高价值: 100%|████████████████████| 5625000/5625000 [01:25<00:00, 65717.01it/s]
总计1533133名高价值用户
已保存高价值用户到: 10G_data_new/part-00006.csv
用时115.45s

正在处理: 10G_data_new/part-00004.parquet
筛选高价值: 100%|████████████████████| 5625000/5625000 [01:26<00:00, 64854.09it/s]
总计1534805名高价值用户
已保存高价值用户到: 10G_data_new/part-00004.csv
用时114.71s

正在处理: 10G_data_new/part-00000.parquet
筛选高价值: 100%|████████████████████| 5625000/5625000 [01:26<00:00, 64926.40it/s]
总计1533512名高价值用户
已保存高价值用户到: 10G_data_new/part-00000.csv
用时113.04s

正在处理: 10G_data_new/part-00003.parquet
筛选高价值: 100%|████████████████████| 5625000/5625000 [01:22<00:00, 68430.22it/s]
总计1535339名高价值用户
已保存高价值用户到: 10G_data_new/part-00003.csv
用时105.19s

正在处理: 10G_data_new/part-00001.parquet
筛选高价值: 100%|████████████████████| 5625000/5625000 [01:23<00:00, 67548.73it/s]
总计1535606名高价值用户
已保存高价值用户到: 10G_data_new/part-00001.csv
用时105.84s

正在处理: 10G_data_new/part-00007.parquet
筛选高价值: 100%|████████████████████| 5625000/5625000 [01:22<00:00, 67967.18it/s]
总计1535312名高价值用户
已保存高价值用户到: 10G_data_new/part-00007.csv
用时105.97s

正在处理: 10G_data_new/part-00005.parquet
```

```
筛选高价值: 100%|████████████████████| 5625000/5625000 [01:21<00:00, 69120.32it/s]
总计1533328名高价值用户
已保存高价值用户到: 10G_data_new/part-00005.csv
用时105.59s
```

```
正在处理: 30G_data_new/part-00008.parquet
筛选高价值: 100%|████████████████████| 8437500/8437500 [02:01<00:00, 69386.50it/s]
总计2302888名高价值用户
已保存高价值用户到: 30G_data_new/part-00008.csv
用时152.95s

正在处理: 30G_data_new/part-00013.parquet
筛选高价值: 100%|████████████████████| 8437500/8437500 [02:07<00:00, 66307.64it/s]
总计2301143名高价值用户
已保存高价值用户到: 30G_data_new/part-00013.csv
用时163.45s

正在处理: 30G_data_new/part-00002.parquet
筛选高价值: 100%|████████████████████| 8437500/8437500 [02:04<00:00, 67713.19it/s]
总计2301888名高价值用户
已保存高价值用户到: 30G_data_new/part-00002.csv
用时160.19s

正在处理: 30G_data_new/part-00015.parquet
筛选高价值: 100%|████████████████████| 8437500/8437500 [02:01<00:00, 69648.76it/s]
总计2299702名高价值用户
已保存高价值用户到: 30G_data_new/part-00015.csv
用时162.84s

正在处理: 30G_data_new/part-00010.parquet
筛选高价值: 100%|████████████████████| 8437500/8437500 [02:07<00:00, 66030.67it/s]
总计2300315名高价值用户
已保存高价值用户到: 30G_data_new/part-00010.csv
用时165.62s

正在处理: 30G_data_new/part-00006.parquet
筛选高价值: 100%|████████████████████| 8437500/8437500 [02:05<00:00, 67326.72it/s]
总计2300878名高价值用户
已保存高价值用户到: 30G_data_new/part-00006.csv
用时165.55s

正在处理: 30G_data_new/part-00014.parquet
筛选高价值: 100%|████████████████████| 8437500/8437500 [02:08<00:00, 65718.83it/s]
总计2301688名高价值用户
已保存高价值用户到: 30G_data_new/part-00014.csv
用时163.65s

正在处理: 30G_data_new/part-00011.parquet
筛选高价值: 100%|████████████████████| 8437500/8437500 [02:08<00:00, 65850.35it/s]
总计2300344名高价值用户
已保存高价值用户到: 30G_data_new/part-00011.csv
用时161.78s

正在处理: 30G_data_new/part-00004.parquet
筛选高价值: 100%|████████████████████| 8437500/8437500 [02:06<00:00, 66480.77it/s]
```

```
总计2302634名高价值用户
已保存高价值用户到: 30G_data_new/part-00004.csv
用时162.53s

正在处理: 30G_data_new/part-00012.parquet
筛选高价值: 100%|████████████████████| 8437500/8437500 [02:07<00:00, 65963.24it/s]
总计2301805名高价值用户
已保存高价值用户到: 30G_data_new/part-00012.csv
用时162.06s

正在处理: 30G_data_new/part-00009.parquet
筛选高价值: 100%|████████████████████| 8437500/8437500 [02:06<00:00, 66902.58it/s]
总计2298601名高价值用户
已保存高价值用户到: 30G_data_new/part-00009.csv
用时159.63s

正在处理: 30G_data_new/part-00000.parquet
筛选高价值: 100%|████████████████████| 8437500/8437500 [02:08<00:00, 65613.79it/s]
总计2302179名高价值用户
已保存高价值用户到: 30G_data_new/part-00000.csv
用时163.83s

正在处理: 30G_data_new/part-00003.parquet
筛选高价值: 100%|████████████████████| 8437500/8437500 [02:02<00:00, 68839.34it/s]
总计2302014名高价值用户
已保存高价值用户到: 30G_data_new/part-00003.csv
用时160.15s

正在处理: 30G_data_new/part-00001.parquet
筛选高价值: 100%|████████████████████| 8437500/8437500 [02:03<00:00, 68477.68it/s]
总计2301469名高价值用户
已保存高价值用户到: 30G_data_new/part-00001.csv
用时159.85s

正在处理: 30G_data_new/part-00007.parquet
筛选高价值: 100%|████████████████████| 8437500/8437500 [01:59<00:00, 70444.06it/s]
总计2301664名高价值用户
已保存高价值用户到: 30G_data_new/part-00007.csv
用时152.58s

正在处理: 30G_data_new/part-00005.parquet
筛选高价值: 100%|████████████████████| 8437500/8437500 [02:07<00:00, 66253.43it/s]
总计2300040名高价值用户
已保存高价值用户到: 30G_data_new/part-00005.csv
用时163.31s
```