手游玩家付费预测EDA

目的： 利用制表、作图、特征分析等方法，对手游玩家的付费数据进行探索性数据分析，找出不同玩家之间的特点和规律，为接下来的预测模型提供思路和方法

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
plt.rcParams['font.sans-serif']=['simHei']
import warnings
warnings.filterwarnings('ignore')
```

```
# 分块读取再concat拼接
temp_df = []
for chunk in pd.read_table('E:/数据分析学习资料汇总/游戏玩家付费金额预测/tap_fun_train.csv',\
                 sep=',', chunksize=10000):
    temp_df.append(chunk)
data = pd.concat(temp_df, axis=0)
del temp_df
```

```
data.shape
```

```
(2288007, 109)
```

观察和理解特征

```
data.head()
```

	user_id	register_time	wood_add_value	wood_reduce_value	stone_add_value	stone_reduce_value	ivory_add_value
0	1	2018-02-02 19:47:15	20125.0	3700.0	0.0	0.0	0.0
1	1593	2018-01-26 00:01:05	0.0	0.0	0.0	0.0	0.0
2	1594	2018-01-26 00:01:58	0.0	0.0	0.0	0.0	0.0
3	1595	2018-01-26 00:02:13	0.0	0.0	0.0	0.0	0.0
4	1596	2018-01-26 00:02:46	0.0	0.0	0.0	0.0	0.0

```
data.describe()
```

	user_id	wood_add_value	wood_reduce_value	stone_add_value	stone_reduce_value	ivory_add_value	ivory_reduce_value
count	2.288007e+06	2.288007e+06	2.288007e+06	2.288007e+06	2.288007e+06	2.288007e+06	2.288007e+06
mean	1.529543e+06	4.543069e+05	3.698433e+05	1.897788e+05	1.376074e+05	8.075623e+04	3.613170e+04
std	9.399393e+05	4.958667e+06	3.737720e+06	4.670620e+06	3.370166e+06	2.220540e+06	1.782499e+06
min	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	7.499925e+05	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
50%	1.419095e+06	4.203800e+04	9.830000e+03	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
75%	2.299006e+06	1.531180e+05	9.855700e+04	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
max	3.190530e+06	1.239962e+09	7.995875e+08	1.214869e+09	7.962378e+08	5.744961e+08	4.481972e+08

数据类型的观察（int/float/str）与转换（时间类型）

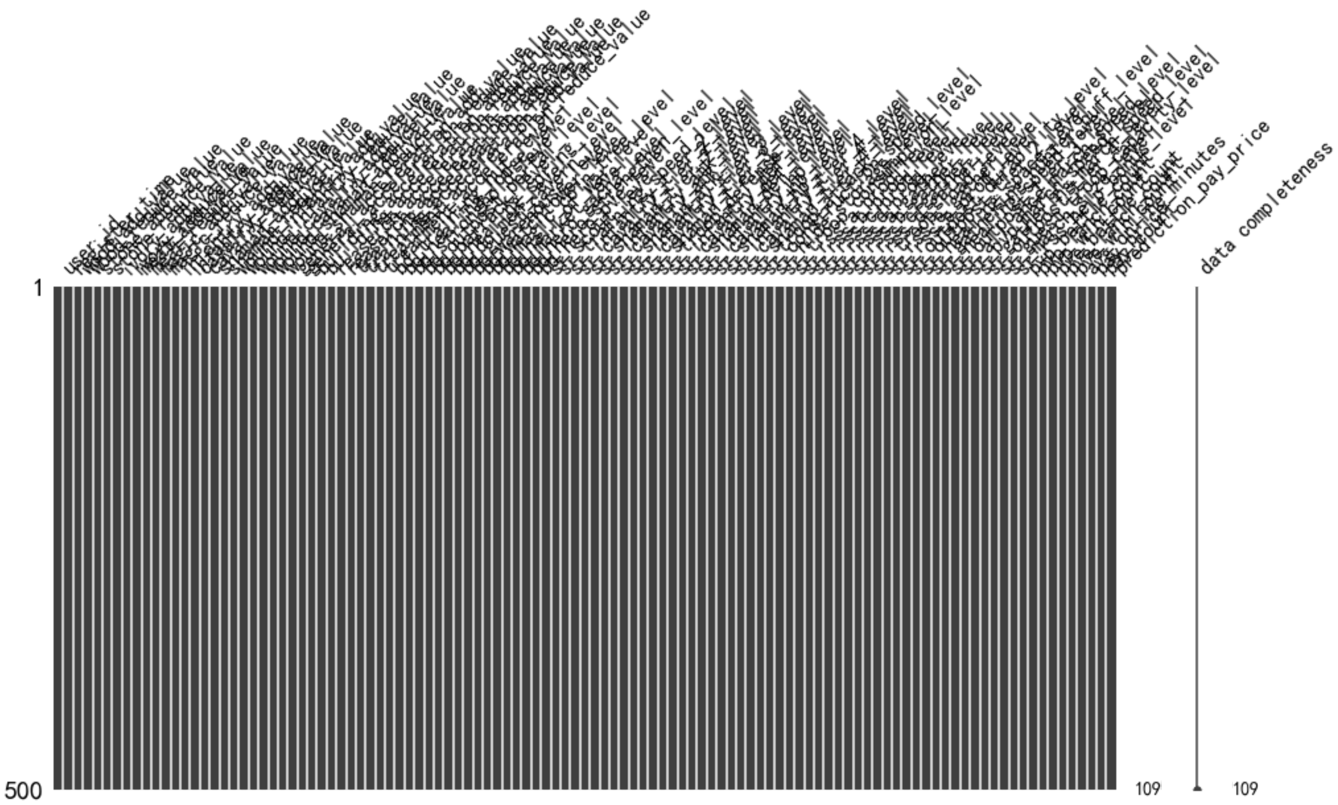
```
data.columns[data.dtypes=='object']
Index(['register_time'], dtype='object')
data.dtypes.value_counts()
data.select_dtypes(include='object').head()
```

	register_time
0	2018-02-02 19:47:15
1	2018-01-26 00:01:05
2	2018-01-26 00:01:58
3	2018-01-26 00:02:13
4	2018-01-26 00:02:46

```
# 去除时分秒
data.register_time = pd.to_datetime(data.register_time).dt.normalize()
```

缺失值检测与 填充/删除

```
import missingno as msno
msno.matrix(data.sample(500),
            figsize=(16,7),
            width_ratios=(15,1),
            labels=True)
```

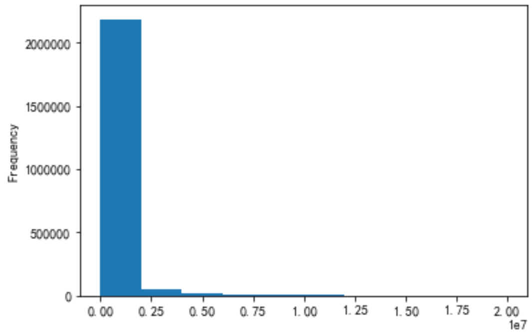



```
# 数据无缺失值
(data.isnull().sum()==0).value_counts()
```

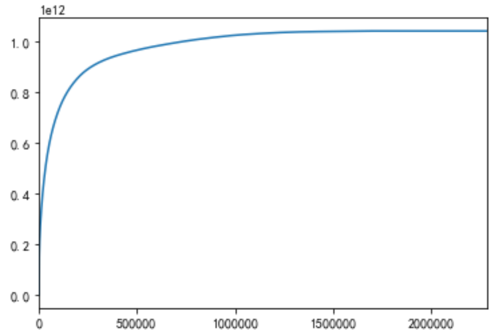
```
True     109
```

提取信息

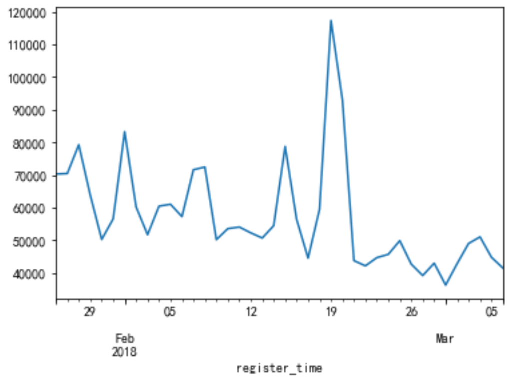
```
# 切比雪夫来消除极值
data[data.wood_add_value<20000000].wood_add_value.plot.hist()
```




```
# 累加图，可以看出大部分的贡献集中在少部分玩家中，二八法则
data.sort_values('wood_add_value',ascending = False).wood_add_value.cumsum().reset_index(drop=True).plot()
```




```
# 每日新增用户数,总体下降，2月19号前后有一波大高潮
data.groupby('register_time').user_id.count().plot()
```

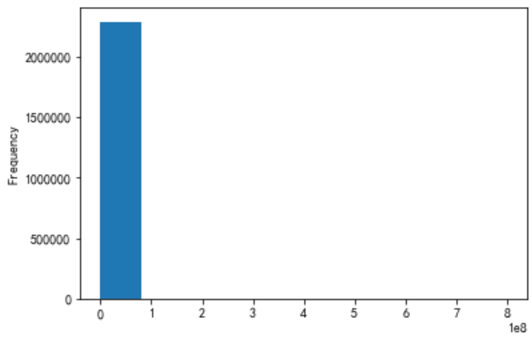



```
# 45日内无消费用户占比
(data.prediction_pay_price==0).value_counts()
```

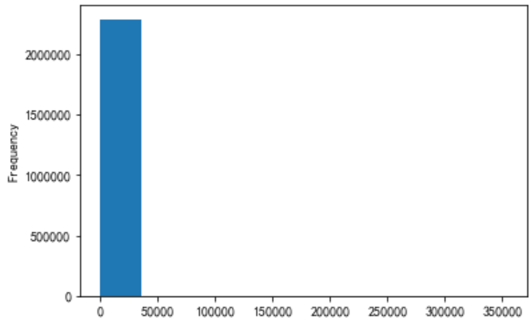
```
True     2242019
False      45988
```

抽样分析有代表性特征的分布情况

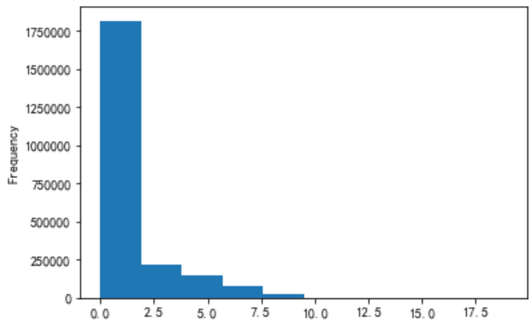
```
data.wood_reduce_value.plot.hist()
```



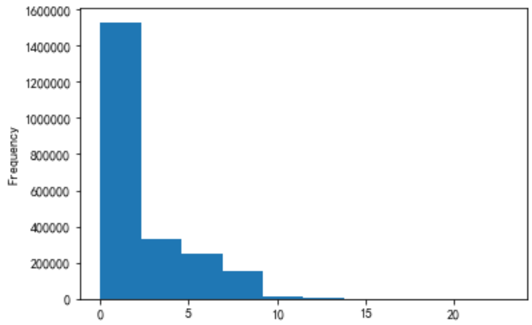

```
data.cavalry_add_value.plot.hist()
```



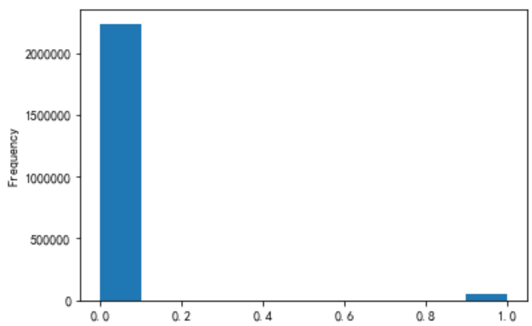

```
data.bd_healing_lodge_level.plot.hist()
```



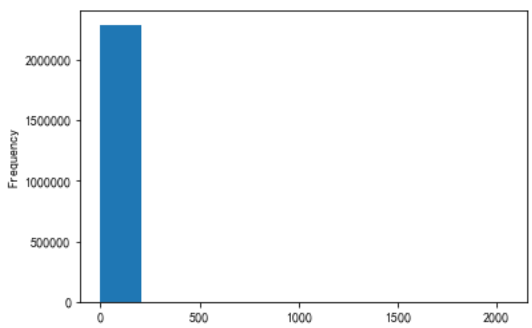

```
data.bd_stronghold_level.plot.hist()
```



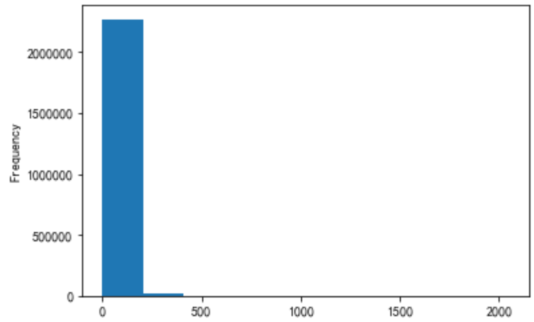

```
data.sr_cavalry_tier_2_level.plot.hist()
```



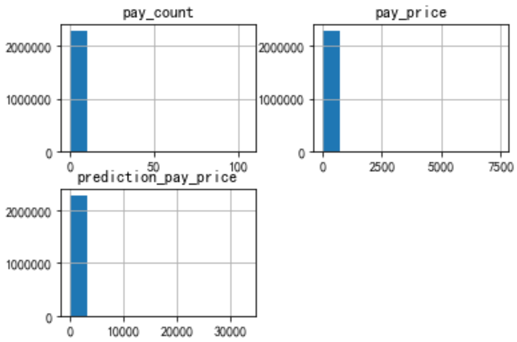

```
data.pvp_battle_count.plot.hist()
```



```
data.avg_online_minutes.plot.hist()
```




```
data[['pay_price','pay_count','prediction_pay_price']].hist()
```

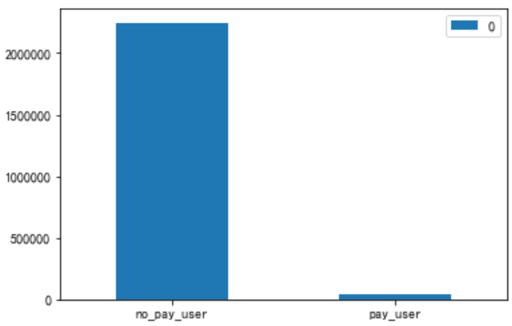


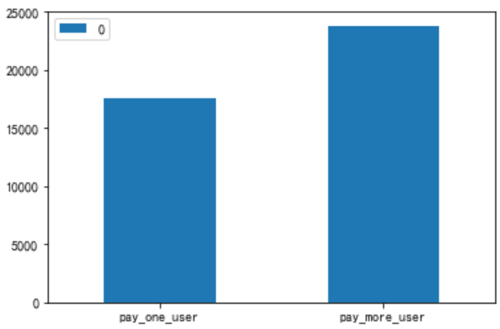
- 整体特征都严重偏斜，大部分玩家只玩了很短的时间

```
lt = []
lt.append(data[data.pay_count==0].pay_count.count())
lt.append(data[data.pay_count>0].pay_count.count())
lt.append(data[data.pay_count==1].pay_count.count())
lt.append(data[data.pay_count>1].pay_count.count())
```

```
df = pd.DataFrame(lt, index=['no_pay_user', 'pay_user', 'pay_one_user', 'pay_more_user'])
df.iloc[0:2].plot.bar()
plt.xticks(rotation=0)
df.iloc[2:].plot.bar()
plt.xticks(rotation=0)
# 付费率
print('付费率:', round(data[data.pay_count>0].user_id.count() / data.user_id.count(), 3))
```

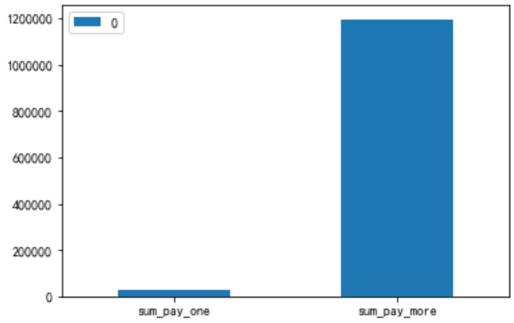
```
付费率: 0.018
```





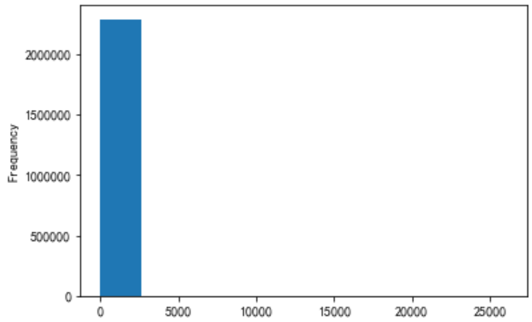
- 大部分是不付费用户
- 在付费用户中，多次付费的用户高于只付费一次的用户，因此要提高付费率

```
df1 = pd.DataFrame([data[data.pay_count==1].pay_price.sum(), data[data.pay_count>1].pay_price.sum()],\
                  index=['sum_pay_one', 'sum_pay_more'])
df1.plot.bar()
plt.xticks(rotation=0)
plt.show()
```



- 付费多次的用户付费总额也远高于付费一次的用户

```
data['price_diff'] = data.prediction_pay_price - data.pay_price
data.price_diff.plot.hist()
```



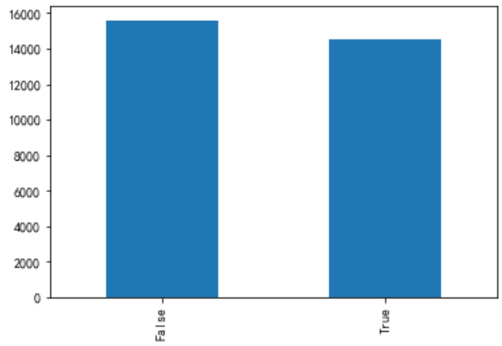

```
# 前7日付费后不再付费用户
unactive_user = data[(data.pay_price>0) & (data.price_diff==0)]
unactive_user.shape
```

```
(30130, 110)
```

```
# 前7日不付费，后45日有付费的用户
data[(data.pay_price==0) & (data.price_diff>0)].shape
```

```
(4549, 110)
```

```
(unactive_user.pay_count>1).value_counts().plot.bar()
```



- 前7日付费后不再付费用户中， 只付费1次与付费多次的占比相当

```
# ARPU ARPPU
print('ARPU =', data.pay_price.sum() / data.user_id.count())
print('ARPPU =', data.pay_price.sum() / data[data.pay_price>0].user_id.count())
```

```
ARPU  = 0.5346691072186407
ARPPU = 29.52114336735926
```

```
# 所有用户日平均在线时长，和周平均在线时长
print('所有用户日平均在线时长(min):', data.avg_online_minutes.sum()/data.user_id.count())
print('所有用户周平均在线时长(min):', data.avg_online_minutes.sum()*7/data.user_id.count())
```

```
所有用户日平均在线时长(min): 10.20
所有用户周平均在线时长(min): 71.45
```

```
# 付费用户日平均在线时长 与 不付费用户日平均在线时长
print('付费用户日平均在线时长(min):', data[data.pay_price>0].avg_online_minutes.sum()/data[data.pay_price>0].user_id.count())
print('不付费用户日平均在线时长(min):', data[data.pay_price==0].avg_online_minutes.sum()/data[data.pay_price==0].user_id.count())
```

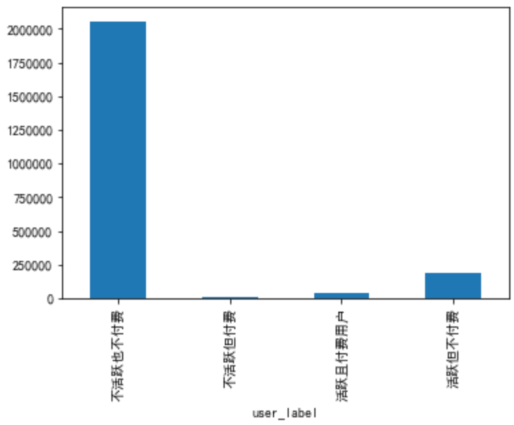
```
付费用户日平均在线时长(min):  140.19
不付费用户日平均在线时长(min): 7.81
```

用象限法 划分 活跃与付费

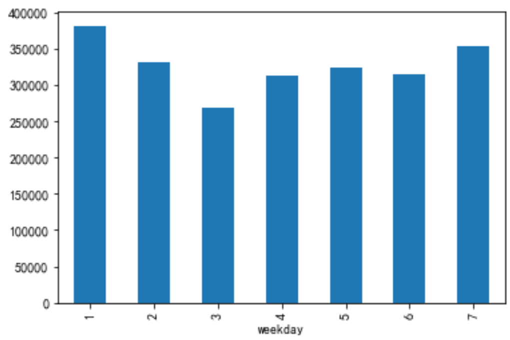
- 第一象限：活跃且付费，重点维护用户
- 第二象限：不活跃但付费，重点发展用户
- 第三象限：不活跃也不付费，一般发展用户
- 第四象限：活跃但不付费，一般维护用户

```
data['user_label'] = 'a'
data['user_label'][(data['avg_online_minutes']>15)& (data['pay_price']>0)] = '活跃且付费用户'
data['user_label'][(data['avg_online_minutes']<=15)& (data['pay_price']>0)] = '不活跃但付费'
data['user_label'][(data['avg_online_minutes']<=15)& (data['pay_price']==0)] = '不活跃也不付费'
data['user_label'][(data['avg_online_minutes']>15)& (data['pay_price']==0)] = '活跃但不付费'
```

```
# 各类型用户占比，重点维护付费且活跃用户，重点发展付费但不活跃用户，一般维护活跃但不付费用户
data.groupby('user_label').user_id.count().plot.bar()
```




```
data['weekday'] = data.register_time.dt.weekday+1
data.groupby('weekday').user_id.count().plot.bar()
```



- 不同星期新增用户数对比：周一与周日比较多，周三最少