数据挖掘作业一

数据探索性分析与数据预处理

姓名：CHANTHAMATH DETHSOMSAY 学号：3820181029

一、 问题描述

本次作业中，将对 2 个数据集进行探索性分析与处理。分析和处理内容包括数据可视化和摘要、数据缺失的处理两部分。

- 在数据摘要任务中，对于数据集中的标称属性，给出每个可能取值的频数；对与数据集中的数值属性，给出最大、最小、均值、中位数、四分位数及缺失值的个数。

- 在数据可视化任务中，对于数据集中的数值属性，分别（1）绘制盒图图（2）绘制直方（3）绘制 qq 图以检验其分布是否为正态分布以对离群值进行识别。

- 在数据缺失处理任务中，观察数据集中的缺失数据，分析其缺失的原因,并分别使用四种策略对缺失值进行处理：
  （1）将缺失部分剔除
  （2）用最高频率值来填补缺失值
  （3）通过属性的相关关系来填补缺失值
  （4）通过数据对象之间的相似性来填补缺失值
  处理后，可视化地对比新旧数据集。

二、 数据说明

- 数据集 1：Trending YouTube Video Statistics 该数据集共包含 16 个属性，40881 条数据记录。

- 数据集 2: Wine Reviews 该数据集共包含 14 个属性，129971 条数据记录。

三、 数据分析过程

1. 数据可视化和摘要

数据摘要:

经过对数据集 1 的 16 个属性进行人工识别，其中包含标称属性 6 个，分别为：trending_date、publish_time、category_id、comments_disabled、rating_ disabled、video_error_or_removed。包含数值属性 4 个，分别为：views、likes、dislkes、comment_count。以及其他属性。

- 创建标称属性列表然后用 for 循环把属性的值添加到列表，每个值对应着自己的属性。

```
trending_date = []
for i in pr['trending_date']:
    if (i == i):
        trending_date.append(i) #
count = Counter(trending_date)  # 类型:  <class 'collections.Counter'>
count_dict = dict(count)
print("trending_date: ")
print(count_dict)
```

- 创建数值属性列表然后用 for 循环把属性的值添加到列表，每个值对应着自己的属性。

```
views = []
views1= []
for i in pr['views']:
    views.append(i)
    if(i==i):
        views1.append(i)
```

- 调用 numpy 模块获取描述数据分析的均值、中位数、最大、最小、缺失值、四分位数及众数的个数。

```
values=0
print("均值: ",np.mean(views1))
print("中位数: ",np.median(views1))
print("最大值: ",min(views1))
print("最小值: ",max(views1))
for i in views:
    if(i!=i):
        values=values+1
print("缺失值个数:",values)
print("四分位数:",np.percentile(views1, (25, 50, 75), interpolation='midpoint'))
counts = np.bincount(views1)
# 返回众数
print("众数: ",np.argmax(counts))
```

输出结果如下：

1. views

```
均值：  1147035.9107898534
中位数：  371204.0
最大值：  733
最小值：  137843120
缺失值个数：0
四分位数：[143902. 371204. 963302.]
众数：  4953
```

2. likes

```
均值：  39582.68824148137
中位数：  8780.0
最大值：  0
最小值：  5053338
缺失值个数：0
四分位数：[ 2191.  8780. 28717.]
众数：  0
```

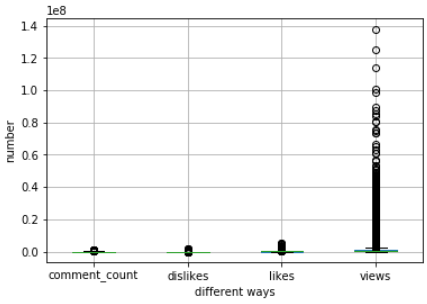
3. dislkes

```
均值：  2009.1954453168953
中位数：  303.0
最大值：  0
最小值：  1602383
缺失值个数：0
四分位数：[ 99. 303. 950.]
众数：  0
```

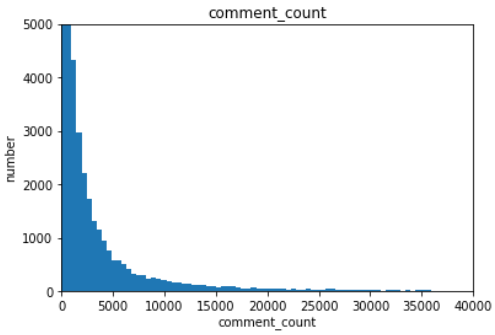
4. comment_count

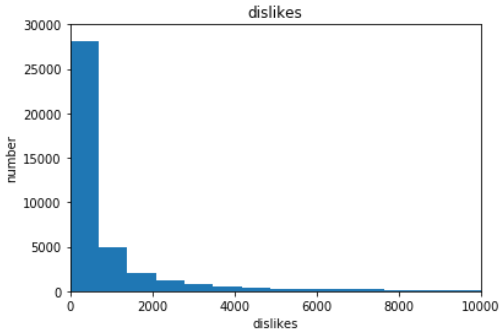
```
均值：  5042.974707076637
中位数：  1301.0
最大值：  0
最小值：  1114800
缺失值个数：0
四分位数：[ 417. 1301. 3713.]
众数：  0
```

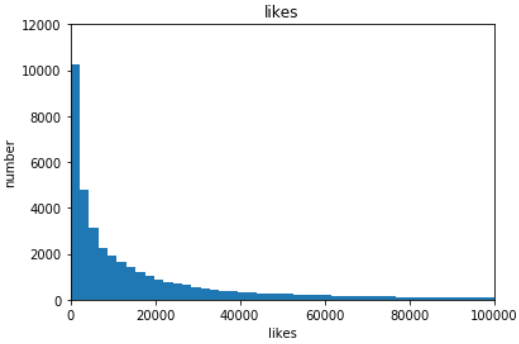
■ 画出盒图

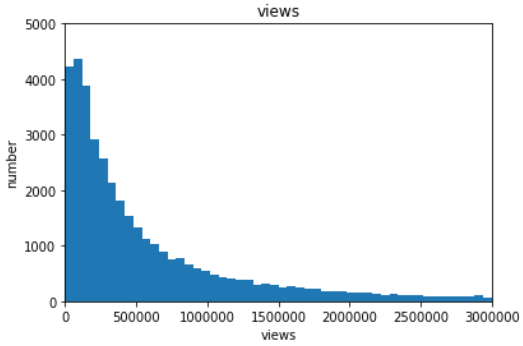


■ 画出直方图



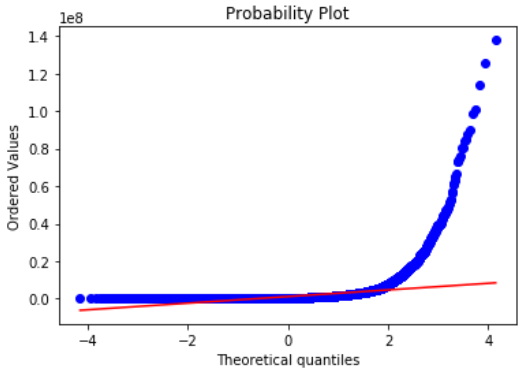




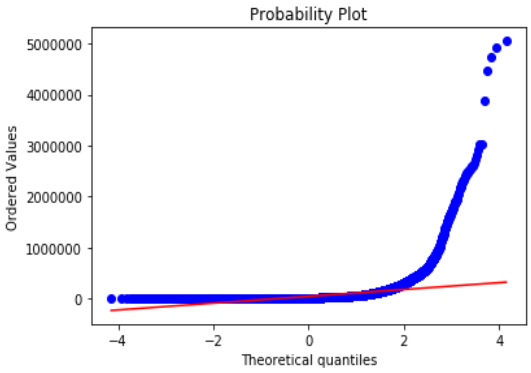


■ 画出 qq 图

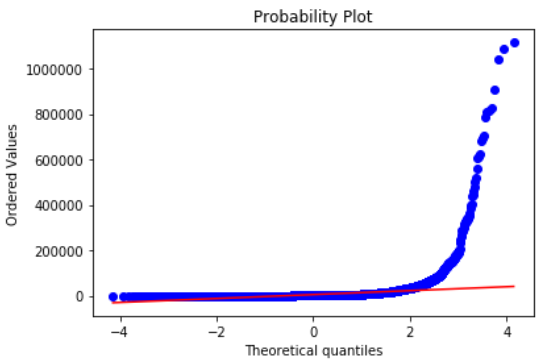
Views



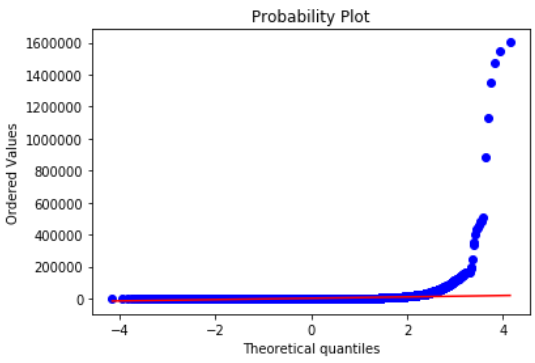
likes



comment_count



dislikes



■ 没有缺失值

经过对数据集 2 的 14 个属性进行人工识别，其中包含标称属性 7 个，分别为：country、point、province、region_1、region_2、variety、winery。包含数值属性 1 个为：price。

- 对数据集 2，也跟数据集 1 同样的操作。创建标称属性列表然后用 for 循环把属性的值添加到列表，每个值对应着自己的属性。

```
country = []
for i in pr['country']:
    if (i == i):
        country.append(i)
count = Counter(country)  # 类型:  <class 'collections.Counter'>
count_dict = dict(count)
print("country频数: ")
print(count_dict)
```

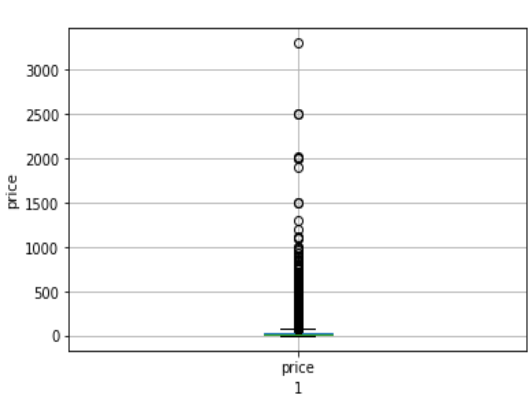
- 创建数值属性列表然后用 for 循环把属性的值添加到列表，每个值对应着自己的属性。调用 numpy 模块获取描述数据分析的均值、中位数、最大、最小、缺失值、四分位数及众数的个数。

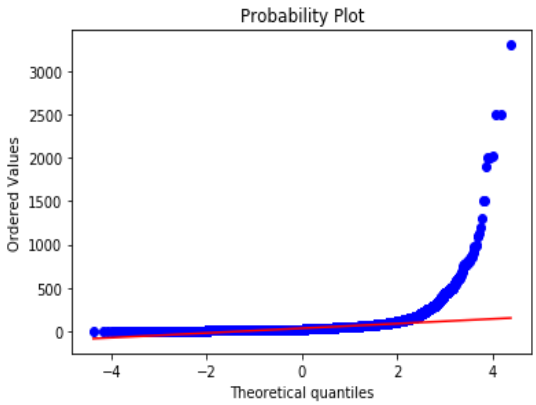
```
price = []
price1= []
for i in pr['price']:
    price.append(i)
    if(i==i):
        price1.append(i)
values=0
print("均值: ",np.mean(price1))
print("中位数: ",np.median(price1))
print("最大值: ",min(price1))
print("最小值: ",max(price1))
for i in price:
    if(i!=i):
        values=values+1
print("缺失值个数:",values)
print("四分位数:",np.percentile(price1, (25, 50, 75), interpolation='midpoint'))
counts = np.bincount(price1)
# 返回众数
print("众数: ",np.argmax(counts))
```

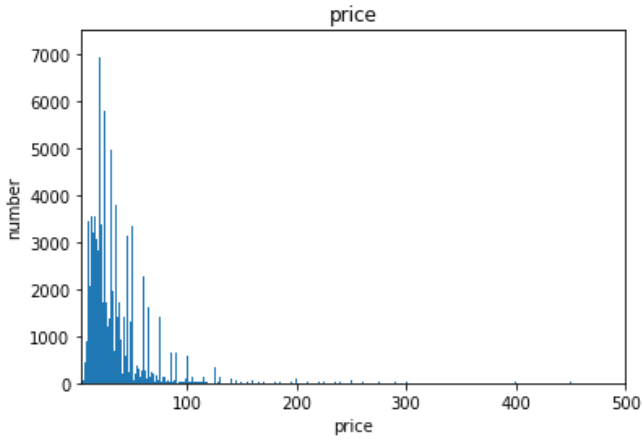
输出结果：

```
均值:  35.363389129985535
中位数:  25.0
最大值:  4.0
最小值:  3300.0
缺失值个数: 8996
四分位数: [17. 25. 42.]
众数:  20
```

■ 画出盒图、qq 图及直方图：





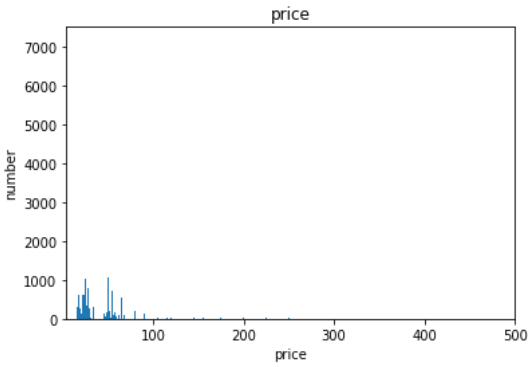


2. 数据缺失处理

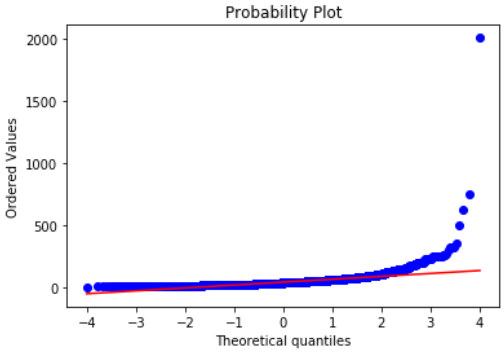
■ **处理缺失值将缺失部分剔除**

```
temp = pr.copy(deep = True);
temp1 = temp.dropna(axis=0, how='any');#有空属性则剔除元组
print('丢弃缺失值元祖');
```

用直方图显示已缺失部分剔除

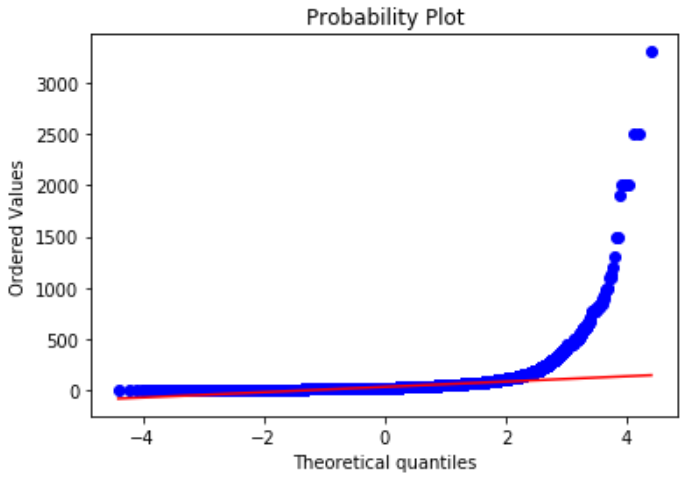


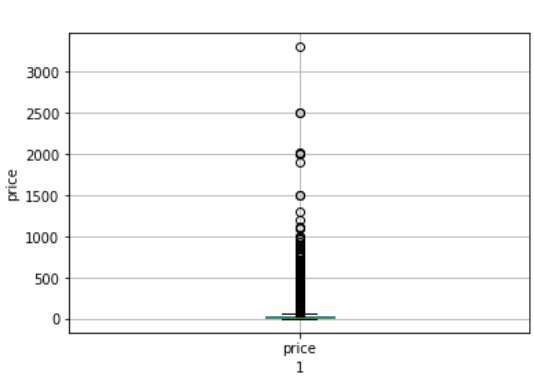
将缺失值剔除后，分布接近于正态分布

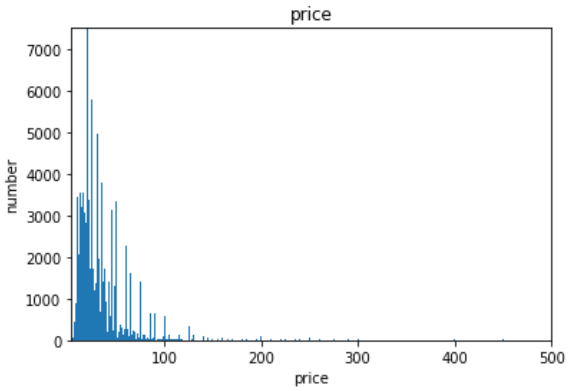


- 用最高频率值来填补缺失值

```
temp2 = pr.copy(deep = True);
for i in temp2.columns:
    DataColumn = temp[i];  # 获取该列
    MostFrequentElement = DataColumn.value_counts().idxmax();
    # print('属性', i, '的众数是', str(MostFrequentElement));
    DataColumn = DataColumn.fillna(value=MostFrequentElement);  # 众数填补缺失值
    temp2[i] = DataColumn;
else:
    print('缺失值用众数填补完成');
```

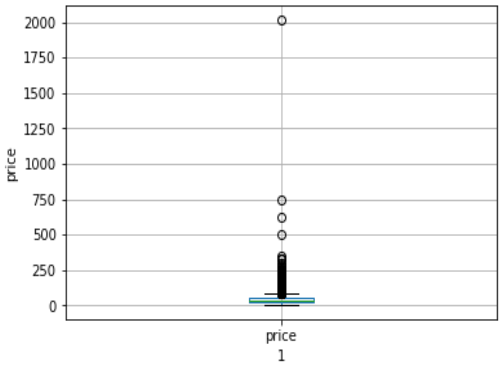






■ **可视化对比：**

将缺失部分剔除



用最高频率值来填补缺失值