数据下载：链接: https://pan.baidu.com/s/1Th1T_N3CYUG6SJwhdRN-XA (https://pan.baidu.com/s/1Th1T_N3CYUG6SJwhdRN-XA) 提取码: hkpd

基于淘宝电商的宠物消费生态大数据研究

一、项目背景

“撸猫”、“吸狗”近两年成为一种风潮，养宠群体加速扩大，宠物经济持续增长。 据《2018年中国宠物行业白皮书》显示，目前我国宠物市场规模已达1708亿元，预计2023年宠物市场规模将超过4000亿。

二、项目说明

本项目通过淘宝花鸟市场收集了宠物的居所、服饰、食品、保健等各类消费品信息，经过清洗过滤数据抽取特征后，借助丰富的可视化方法及数据挖掘模型，全面的解析宠物用品市场的消费特征，探索未来宠物经济的发展趋势，为商家决策提供数据支撑。

准备项目数据为：

- 宠物商品分类信息.xlsx
- 宠物商品信息.csv
- 宠物商品评论数据.csv

三、项目要求

1. 数据预处理

In [2]:

```
# 导入模块
import numpy as np
import pandas as pd
```

In [3]:

```
# 导入 天猫国际商品信息.csv 数据
pet_product=pd.read_csv("宠物商品信息.csv")
```

In [4]:

```
# 查看数据
pet_product
```

Out[4]:

	爬取时间(__time)	爬取链接(__url)	商品ID(product_id)	商品名称(name)	商品描述(description)	(curre
0	2019-11-12 15:10:23	https://item.taobao.com/item.htm?id=566960035676	566960035676	蛇仔鱼苦力泥鳅观赏鱼热带鱼除蛋白虫涡虫虾缸搭档清洁鱼易养鱼	NaN	
1	2019-11-12 15:10:22	https://item.taobao.com/item.htm?id=604255887198	604255887198	水母活物水族箱宠物水母活 淡水观赏水母缸活小型水族箱	NaN	20.
2	2019-11-12 15:10:21	https://item.taobao.com/item.htm?id=521281145450	521281145450	红绿灯鱼活体群游灯鱼热带观赏鱼小型灯科鱼孔雀鱼活体饲料鱼	NaN	
3	2019-11-12 15:10:18	https://item.taobao.com/item.htm?id=604750097046	604750097046	养水母用海盐海水盐珊瑚盐赤月小水母食物饵料饲料喂食吃的维护包	NaN	
4	2019-11-12 15:10:15	https://item.taobao.com/item.htm?id=43436641783	43436641783	水母活体套装迷你音乐玻璃鱼缸赤月情人节礼物海月水母生日礼物	NaN	
...	...	...	...	...	...	
23927	2019-11-15 10:43:23	https://item.taobao.com/item.htm?id=530942254401	530942254401	壹品红血鹦鹉增红鱼粮红鹦鹉增色鱼饲料地图招财鱼食一品红鱼食	NaN	
23928	2019-11-15 10:43:22	https://item.taobao.com/item.htm?id=37482165781	37482165781	海神 血鹦鹉增红专用粮500g/1000g血鹦鹉鱼食饲料鱼粮（升级版）	NaN	

	爬取时间(__time)	爬取链接(__url)	商品ID(product_id)	商品名称(name)	商品描述(description)	(curre
23929	2019-11-15 10:43:22	https://item.taobao.com/item.htm?id=597889386973	597889386973	益口红血鹦鹉饲料增红增色专用粮观赏鱼食招财鱼鱼粮包邮	NaN	
23930	2019-11-15 10:43:21	https://item.taobao.com/item.htm?id=534921855001	534921855001	帝溢红 血鹦鹉增红增色鱼食 高虾红素营养 上浮型5天增红鱼粮饲料	NaN	
23931	2019-11-15 10:43:21	https://item.taobao.com/item.htm?id=604810272138	604810272138	虹立方血鹦鹉增红鱼粮财神罗汉饲料热带观赏鱼鱼食 不浑水	NaN	

23932 rows × 21 columns

In [5]:

```
# 重命名列
pet_product.columns
```

Out[5]:

```
Index(['爬取时间(__time)', '爬取链接(__url)', '商品ID(product_id)', '商品名称(name)',
       '商品描述(description)', '商品现价(current_price)', '商品原价(original_price)',
       '月销量(sales_count)', '评论数(comments_count)', '发货地址(shipping_address)',
       '商品发布时间(sales_count)', '商品规格(sku)', '商品库存(stock)', '店铺名称(shop_name)',
       '店铺url(shop_url)', '商品参数(params)', '商品sku详情(product_sku_detail)',
       '商品链接(url)', '商品详情(detail)', '店铺评分(shop_score)', '宝贝收藏数(fav_count)'],
      dtype='object')
```

In [6]:

```
pet_product.columns=['爬取时间', '爬取链接', '商品ID', '商品名称','商品描述', '商品现价',
                     '商品原价','月销量', '评论数', '发货地址','商品发布时间', '商品规格',
                     '商品库存', '店铺名称','店铺url', '商品参数', '商品sku详情','商品链接'
                     '商品详情', '店铺评分', '宝贝收藏数']
pet_product.columns
```

Out[6]:

```
Index(['爬取时间', '爬取链接', '商品ID', '商品名称', '商品描述', '商品现价',
'商品原价', '月销量', '评论数',
       '发货地址', '商品发布时间', '商品规格', '商品库存', '店铺名称', '店铺ur
l', '商品参数', '商品sku详情',
       '商品链接', '商品详情', '店铺评分', '宝贝收藏数'],
      dtype='object')
```

In [7]:

```
# 导入 宠物商品分类信息.xlsx
pet_genre=pd.read_excel("宠物商品分类信息.xlsx")
```

In [8]:

```
# 查看数据
pet_genre
```

Out[8]:

	一级分类	二级分类	三级分类	商品ID
0	猫猫狗狗	猫猫狗狗	猫主粮	39456693691
1	猫猫狗狗	猫猫狗狗	猫零食	602386539981
2	猫猫狗狗	猫猫狗狗	猫零食	563737903849
3	猫猫狗狗	猫猫狗狗	猫零食	592530526353
4	猫猫狗狗	猫猫狗狗	猫零食	580322063342
...	...	...	...	...
24460	奇趣宠物	仓鼠类及其它小宠	香猪	589742817020
24461	奇趣宠物	仓鼠类及其它小宠	香猪	581946660178
24462	奇趣宠物	仓鼠类及其它小宠	香猪	562230862380
24463	奇趣宠物	仓鼠类及其它小宠	饲料/零食	606205536913
24464	奇趣宠物	仓鼠类及其它小宠	饲料/零食	603163402751

24465 rows × 4 columns

In [9]:

```
# 将商品信息和商品分类信息进行合并
df=pd.merge(pet_product,pet_genre,left_on="商品ID",right_on="商品ID")
```

In [10]:

```
# 按照列名称的排列顺序进行缺失值的处理和数据变换
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 23932 entries, 0 to 23931
Data columns (total 24 columns):
爬取时间        23932 non-null object
爬取链接        23932 non-null object
商品ID        23932 non-null int64
商品名称        23932 non-null object
商品描述        668 non-null object
商品现价        23932 non-null object
商品原价        23932 non-null object
月销量         22410 non-null object
评论数         22840 non-null float64
发货地址        23932 non-null object
商品发布时间      23932 non-null int64
商品规格        23932 non-null object
商品库存        23932 non-null int64
店铺名称        23929 non-null object
店铺url       23932 non-null object
商品参数        23932 non-null object
商品sku详情     23932 non-null object
商品链接        23932 non-null object
商品详情        23932 non-null object
店铺评分        23932 non-null object
宝贝收藏数       23932 non-null object
一级分类        23932 non-null object
二级分类        23932 non-null object
三级分类        23932 non-null object
dtypes: float64(1), int64(3), object(20)
memory usage: 4.6+ MB
```

In [11]:

```
# 处理商品描述数据，缺失值过多，删除该数据列
df=df.drop(columns=["商品描述"])
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 23932 entries, 0 to 23931
Data columns (total 23 columns):
爬取时间        23932 non-null object
爬取链接        23932 non-null object
商品ID        23932 non-null int64
商品名称        23932 non-null object
商品现价        23932 non-null object
商品原价        23932 non-null object
月销量         22410 non-null object
评论数         22840 non-null float64
发货地址        23932 non-null object
商品发布时间      23932 non-null int64
商品规格        23932 non-null object
商品库存        23932 non-null int64
店铺名称        23929 non-null object
店铺url       23932 non-null object
商品参数        23932 non-null object
商品sku详情     23932 non-null object
商品链接        23932 non-null object
商品详情        23932 non-null object
店铺评分        23932 non-null object
宝贝收藏数       23932 non-null object
一级分类        23932 non-null object
二级分类        23932 non-null object
三级分类        23932 non-null object
dtypes: float64(1), int64(3), object(19)
memory usage: 4.4+ MB
```

In [12]:

```
# 处理商品的现价和原价数据
df["商品原价"].apply(type)
```

Out[12]:

```
0        <class 'str'>
1        <class 'str'>
2        <class 'str'>
3        <class 'str'>
4        <class 'str'>
            ...
23927    <class 'str'>
23928    <class 'str'>
23929    <class 'str'>
23930    <class 'str'>
23931    <class 'str'>
Name: 商品原价, Length: 23932, dtype: object
```

In [13]:

```
df["商品现价"].apply(type)
```

Out[13]:

```
0        <class 'str'>
1        <class 'str'>
2        <class 'str'>
3        <class 'str'>
4        <class 'str'>
            ...
23927    <class 'str'>
23928    <class 'str'>
23929    <class 'str'>
23930    <class 'str'>
23931    <class 'str'>
Name: 商品现价, Length: 23932, dtype: object
```

In [14]:

```
# 将现价和原价区间价格转换为平均价格
def get_price(s):
    price=s.split("-")
    L=[float(i)for i in price]
    return np.mean(L)
```

In [15]:

```
df["商品原价"]=df["商品原价"].apply(get_price)
```

In [16]:

```
df["商品现价"]=df["商品现价"].apply(get_price)
```

In [17]:

```
df["商品原价"][:5]
```

Out[17]:

```
0     14.14
1    506.94
2     54.50
3    177.25
4    200.00
Name: 商品原价, dtype: float64
```

In [18]:

```
df["商品现价"][:5]
```

Out[18]:

```
0      9.90
1    243.33
2     54.50
3    177.25
4    173.00
Name: 商品现价, dtype: float64
```

In [19]:

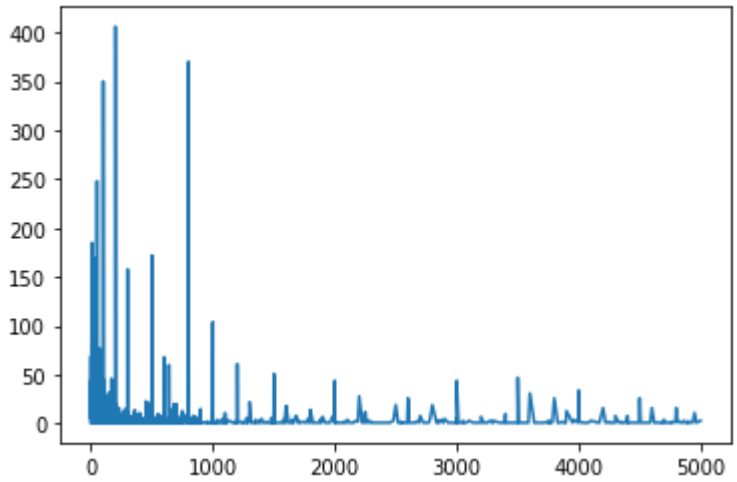
```
# 将商品现价做分箱处理
```

In [20]:

```
import matplotlib.pyplot as plt
```

In [21]:

```
# 查看商品现价小于5000 的数据折线图便于确定数据分箱大小
df_5000 = df[df['商品现价']<5000]
price = df_5000['商品现价'].value_counts().sort_index()
plt.plot(price.index,price.values)
plt.show()
```



In [22]:

```
# 确定数据分箱0-99,100-199,200-299,300-399,400-499,500+
def get_level(p):
    level=p//100
    if level==0:
        return"0-99"
    elif level==1:
        return"100-199"
    elif level==2:
        return"200-299"
    elif level==3:
        return"300-399"
    elif level==4:
        return"400-400"
    elif level>=5:
        return"500+"
    else:
        return"计算出错"
```

In [23]:

```
df["现价等级"]=df["商品现价"].apply(get_level)
```

In [24]:

```
df.loc[0]
```

Out[24]:

```
爬取时间                                                2019-11-12 15:10:23
爬取链接          https://item.taobao.com/item.htm?id=566960035676 (http
s://item.taobao.com/item.htm?id=566960035676)
商品ID                                                    566960035676
商品名称                                  蛇仔鱼苦力泥鳅观赏鱼热带鱼除蛋白虫涡虫虾缸搭
档清洁鱼易养鱼
商品现价                                                          9.9
商品原价                                                        14.14
月销量                                                           303
评论数                                                          1055
发货地址                                                        广东广州
商品发布时间                                                1558231120
商品规格          [{"label":"颜色分类","values":[{"desc":"蛇仔2条（4-5cm...
商品库存                                                        90500
店铺名称                                                      优鱼自然水族馆
店铺url                          https://shop64472738.taobao.com (http
s://shop64472738.taobao.com)
商品参数          [{"label":"品牌","value":"青青自然"},{"label":"品种","...
商品sku详情       [{"sku_id":"3770715903143","sku_name":"蛇仔2条（4-...
商品链接          https://item.taobao.com/item.htm?id=566960035676 (http
s://item.taobao.com/item.htm?id=566960035676)
商品详情          <div> <a name="hlg_promo_desc_35740188_start">...
店铺评分          {'描述相符': ['4.8', '持平0.30%'], '服务态度': ['4.8', '...
宝贝收藏数                                                        957
一级分类                                                        水族世界
二级分类                                                        活体生物
三级分类                                                         热带鱼
现价等级                                                        0-99
Name: 0, dtype: object
```

In [25]:

```
# 处理商品月销量的缺失值
```

In [26]:

```
df[df["月销量"].isnull()]
```

Out[26]:

	爬取时间	爬取链接	商品ID	商品名称	商品现价	商品原价	月销量	评论数	发货地址	商品发布
18465	2019-11-15 12:59:42	https://item.taobao.com/item.htm?id=542315757302	542315757302	德希恩硝化细菌水族消化菌胶囊鱼缸水质澄清剂干	98.50	98.50	NaN	179.0	上海	15239471

In [27]:

```
# 根据经验，月销量缺失都是对应的月销量为0，所有用0来填充缺失值。
df["月销量"]=df["月销量"].fillna("0")
```

In [28]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 23932 entries, 0 to 23931
Data columns (total 24 columns):
爬取时间        23932 non-null object
爬取链接        23932 non-null object
商品ID        23932 non-null int64
商品名称        23932 non-null object
商品现价        23932 non-null float64
商品原价        23932 non-null float64
月销量         23932 non-null object
评论数         22840 non-null float64
发货地址        23932 non-null object
商品发布时间      23932 non-null int64
商品规格        23932 non-null object
商品库存        23932 non-null int64
店铺名称        23929 non-null object
店铺url       23932 non-null object
商品参数        23932 non-null object
商品sku详情     23932 non-null object
商品链接        23932 non-null object
商品详情        23932 non-null object
店铺评分        23932 non-null object
宝贝收藏数       23932 non-null object
一级分类        23932 non-null object
二级分类        23932 non-null object
三级分类        23932 non-null object
现价等级        23932 non-null object
dtypes: float64(3), int64(3), object(18)
memory usage: 5.2+ MB
```

In [29]:

```
df["月销量"]=[i.replace("+","")for i in df["月销量"]]
```

In [30]:

```
#去掉"万"字，并将前面数字*10000
def replace_wan(list):
    for i in list:
        a="万"
        if a in i:
            i=i.replace("万","")
            return float(i)*10000
        else:
            return float(i)
df["月销量"]=df["月销量"].apply(replace_wan)
```

In [31]:

```
for i in df["月销量"]:
    print(i)
```

```
3.0
1.0
6.0
1.0
1.0
3.0
1.0
5.0
1.0
6.0
3.0
1.0
2.0
2.0
9.0
3.0
1.0
2.0
4.0
```

In [32]:

```
# 处理评论数缺失值,缺失原理与月销量相同，可以用0填充
df["评论数"]=df["评论数"].fillna(0)
```

In [33]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 23932 entries, 0 to 23931
Data columns (total 24 columns):
爬取时间        23932 non-null object
爬取链接        23932 non-null object
商品ID        23932 non-null int64
商品名称        23932 non-null object
商品现价        23932 non-null float64
商品原价        23932 non-null float64
月销量         23932 non-null float64
评论数         23932 non-null float64
发货地址        23932 non-null object
商品发布时间      23932 non-null int64
商品规格        23932 non-null object
商品库存        23932 non-null int64
店铺名称        23929 non-null object
店铺url       23932 non-null object
商品参数        23932 non-null object
商品sku详情     23932 non-null object
商品链接        23932 non-null object
商品详情        23932 non-null object
店铺评分        23932 non-null object
宝贝收藏数       23932 non-null object
一级分类        23932 non-null object
二级分类        23932 non-null object
三级分类        23932 non-null object
现价等级        23932 non-null object
dtypes: float64(4), int64(3), object(17)
memory usage: 5.2+ MB
```

In [34]:

```
# 处理发货地址,提取发货省份数据
df['发货地址'][:5]
```

Out[34]:

```
0    广东广州
1      上海
2      上海
3    山西运城
4      北京
Name: 发货地址, dtype: object
```

In [35]:

```
pro_list = ['北京',
 '上海',
 '天津',
 '重庆',
 '河北',
 '山西',
 '内蒙古',
 '黑龙江',
 '吉林',
 '辽宁',
 '陕西',
 '甘肃',
 '青海',
 '新疆',
 '宁夏',
 '山东',
 '河南',
 '江苏',
 '浙江',
 '安徽',
 '江西',
 '福建',
 '台湾',
 '湖北',
 '湖南',
 '广东',
 '广西',
 '海南',
 '四川',
 '云南',
 '贵州',
 '西藏',
 '香港',
 '澳门'
]
```

In [36]:

```
# 注意由于可能存在国外地区，所以如果在整个省份列表里没有找到对应的数据，
# 就使用原来的发货地址数据。
def get_place(s):
    for i in pro_list:
        if i in s:
            return i
    return s
```

In [37]:

```
# 使用get_place()函数修改原来的 发货地址
df["发货地址"]=df["发货地址"].apply(get_place)
```

In [38]:

```
# 查看发货地址数据
df['发货地址']
```

Out[38]:

```
0         广东
1         上海
2         上海
3         山西
4         北京
          ..
23927     山东
23928     河北
23929     上海
23930     江苏
23931     广东
Name: 发货地址, Length: 23932, dtype: object
```

In [39]:

```
# 处理商品发布时间,将时间戳转化成为字符串日期，注意该时间是指商品在店铺的上架时间
```

In [40]:

```
df["商品发布时间"]
```

Out[40]:

```
0         1558231120
1         1570842970
2         1516867541
3         1570198835
4         1505488373
             ...
23927     1461914206
23928     1567228364
23929     1562846725
23930     1504776600
23931     1572060713
Name: 商品发布时间, Length: 23932, dtype: int64
```

In [41]:

```
# 导入时间模块
import time
```

In [42]:

```
df["商品发布时间"].apply(lambda op:time.strftime("%y-%m-%d",time.localtime(op)))
```

Out[42]:

```
0         19-05-19
1         19-10-12
2         18-01-25
3         19-10-04
4         17-09-15
            ...
23927     16-04-29
23928     19-08-31
23929     19-07-11
23930     17-09-07
23931     19-10-26
Name: 商品发布时间, Length: 23932, dtype: object
```

In [43]:

```
df["商品发布时间"]=df["商品发布时间"].apply(lambda op:time.strftime("%y-%m-%d",time.loca
```

In [44]:

```
df["商品发布时间"]
```

Out[44]:

```
0         19-05-19
1         19-10-12
2         18-01-25
3         19-10-04
4         17-09-15
            ...
23927     16-04-29
23928     19-08-31
23929     19-07-11
23930     17-09-07
23931     19-10-26
Name: 商品发布时间, Length: 23932, dtype: object
```

In [45]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 23932 entries, 0 to 23931
Data columns (total 24 columns):
爬取时间        23932 non-null object
爬取链接        23932 non-null object
商品ID        23932 non-null int64
商品名称        23932 non-null object
商品现价        23932 non-null float64
商品原价        23932 non-null float64
月销量         23932 non-null float64
评论数         23932 non-null float64
发货地址        23932 non-null object
商品发布时间      23932 non-null object
商品规格        23932 non-null object
商品库存        23932 non-null int64
店铺名称        23929 non-null object
店铺url       23932 non-null object
商品参数        23932 non-null object
商品sku详情     23932 non-null object
商品链接        23932 non-null object
商品详情        23932 non-null object
店铺评分        23932 non-null object
宝贝收藏数       23932 non-null object
一级分类        23932 non-null object
二级分类        23932 non-null object
三级分类        23932 non-null object
现价等级        23932 non-null object
dtypes: float64(4), int64(2), object(18)
memory usage: 5.2+ MB
```

In [46]:

```
# 商品规格是指购买商品时需要用户进行选择的具体种类，由于宠物数据种类非常繁多，
# 该数据不具备统一的格式，所以不处理也不使用该数据
# 同样的道理，商品的参数也不做处理
```

In [ ]:

```

```

In [47]:

```
# 处理店铺评分数据
```

In [48]:

```
df["店铺评分"][:10]
```

Out[48]:

```
0    {'描述相符': ['4.8', '持平0.30%'], '服务态度': ['4.8', '...
1    {'描述相符': ['4.5', '低于5.52%'], '服务态度': ['4.7', '...
2    {'描述相符': ['4.8', '高于34.13%'], '服务态度': ['4.9', ...
3                                                 未获取到数据
4    {'描述相符': ['4.8', '高于30.77%'], '服务态度': ['4.9', ...
5    {'描述相符': ['4.8', '高于35.26%'], '服务态度': ['4.9', ...
6    {'描述相符': ['4.3', '低于9.67%'], '服务态度': ['4.4', '...
7    {'描述相符': ['4.8', '高于30.77%'], '服务态度': ['4.9', ...
8    {'描述相符': ['4.4', '低于7.63%'], '服务态度': ['4.5', '...
9    {'描述相符': ['4.4', '低于8.70%'], '服务态度': ['4.3', '...
Name: 店铺评分, dtype: object
```

In [49]:

```
# 首先确定有多少是'未获取到数据'
len(df[df["店铺评分"]=="未获取到数据"])
```

Out[49]:

```
275
```

In [50]:

```
# 由于只有275条数据，所以可以删除掉所有店铺评分是'未获取到数据'的数据行
df=df[df["店铺评分"]!="未获取到数据"]
df
```

Out[50]:

	爬取时间	爬取链接	商品ID	商品名称	商品现价	商品原价	月销量
0	2019-11-12 15:10:23	https://item.taobao.com/item.htm?id=566960035676	566960035676	蛇仔鱼苦力泥鳅观赏鱼热带鱼除蛋白虫涡虫虾缸搭档清洁鱼易养鱼	9.90	14.14	3.0
1	2019-11-12 15:10:22	https://item.taobao.com/item.htm?id=604255887198	604255887198	水母活物水族箱宠物水母活 淡水观赏水母缸活小型水族箱	243.33	506.94	1.0
2	2019-11-12 15:10:21	https://item.taobao.com/item.htm?id=521281145450	521281145450	红绿灯鱼活体群游灯鱼热带观赏鱼小型灯科鱼孔雀鱼活体饲料鱼	54.50	54.50	6.0
4	2019-11-12 15:10:15	https://item.taobao.com/item.htm?id=43436641783	43436641783	水母活体套装迷你音乐玻璃鱼缸赤月情人节礼物海月水母生日礼物	173.00	200.00	1.0
5	2019-11-12 15:10:15	https://item.taobao.com/item.htm?id=585814004251	585814004251	观赏活体海月水母赤月水母生日礼物七夕情人活体宠物倒立水母包邮	78.50	78.50	3.0
...	...	...	...	...	...	...	...
23927	2019-11-15 10:43:23	https://item.taobao.com/item.htm?id=530942254401	530942254401	壹品红血鹦鹉增红鱼粮红鹦鹉增色鱼饲料地图招财鱼食一品红鱼食	48.50	48.50	1.0

	爬取时间	爬取链接	商品ID	商品名称	商品现价	商品原价	月销量
23928	2019-11-15 10:43:22	https://item.taobao.com/item.htm?id=37482165781	37482165781	海神 血鹦鹉增红专用粮500g/1000g血鹦鹉鱼食饲料鱼粮（升级版）	105.00	105.00	1.0
23929	2019-11-15 10:43:22	https://item.taobao.com/item.htm?id=597889386973	597889386973	益口红血鹦鹉饲料增红增色专用粮观赏鱼食招财鱼鱼粮包邮	30.30	30.30	1.0
23930	2019-11-15 10:43:21	https://item.taobao.com/item.htm?id=534921855001	534921855001	帝溢红 血鹦鹉增红增色鱼食 高虾红素营养 上浮型5天增红鱼粮饲料	55.50	55.50	9.0
23931	2019-11-15 10:43:21	https://item.taobao.com/item.htm?id=604810272138	604810272138	虹立方血鹦鹉增红鱼粮财神罗汉饲料热带观赏鱼鱼食 不浑水	60.00	60.00	9.0

23657 rows × 24 columns

In [51]:

```
df.index=range(len(df))
```

In [52]:

```
# 提取描述相符、服务态度、物流服务三项内容的具体分数，
# 以及是处于持平、高于还是低于中的哪种状态
```

In [53]:

```
# 先通过eval()函数把每个字符串里面的字典取出来
[eval(i) for i in df["店铺评分"]]

# 出现报错，查看有多少是'该店铺尚未收到评价'
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call
 last)
<ipython-input-53-ce0489aeee70> in <module>
      1 # 先通过eval()函数把每个字符串里面的字典取出来
----> 2 [eval(i) for i in df["店铺评分"]]
      3
      4 # 出现报错，查看有多少是'该店铺尚未收到评价'

<ipython-input-53-ce0489aeee70> in <listcomp>(.0)
      1 # 先通过eval()函数把每个字符串里面的字典取出来
----> 2 [eval(i) for i in df["店铺评分"]]
      3
      4 # 出现报错，查看有多少是'该店铺尚未收到评价'

<string> in <module>

NameError: name '该店铺尚未收到评价' is not defined
```

In [54]:

```
len(df[df["店铺评分"]=="该店铺尚未收到评价"])
```

Out[54]:

```
318
```

In [55]:

```
# 删除掉所有'店铺评分'是'该店铺尚未收到评价'的数据
df=df[df["店铺评分"]!="该店铺尚未收到评价"]
df
```

Out[55]:

	爬取时间	爬取链接	商品ID	商品名称	商品现价	商品原价	月销量
0	2019-11-12 15:10:23	https://item.taobao.com/item.htm?id=566960035676	566960035676	蛇仔鱼苦力泥鳅观赏鱼热带鱼除蛋白虫涡虫虾缸搭档清洁鱼易养鱼	9.90	14.14	3.0
1	2019-11-12 15:10:22	https://item.taobao.com/item.htm?id=604255887198	604255887198	水母活物水族箱宠物水母活 淡水观赏水母缸活小型水族箱	243.33	506.94	1.0
2	2019-11-12 15:10:21	https://item.taobao.com/item.htm?id=521281145450	521281145450	红绿灯鱼活体群游灯鱼热带观赏鱼小型灯科鱼孔雀鱼活体饲料鱼	54.50	54.50	6.0
3	2019-11-12 15:10:15	https://item.taobao.com/item.htm?id=43436641783	43436641783	水母活体套装迷你音乐玻璃鱼缸赤月情人节礼物海月水母生日礼物	173.00	200.00	1.0
4	2019-11-12 15:10:15	https://item.taobao.com/item.htm?id=585814004251	585814004251	观赏活体海月水母赤月水母生日礼物七夕情人活体宠物倒立水母包邮	78.50	78.50	3.0
...	...	...	...	...	...	...	...
23652	2019-11-15 10:43:23	https://item.taobao.com/item.htm?id=530942254401	530942254401	壹品红血鹦鹉增红鱼粮红鹦鹉增色鱼饲料地图招财鱼食一品红鱼食	48.50	48.50	1.0

	爬取时间	爬取链接	商品ID	商品名称	商品现价	商品原价	月销量
23653	2019-11-15 10:43:22	https://item.taobao.com/item.htm?id=37482165781	37482165781	海神 血鹦鹉增红专用粮500g/1000g血鹦鹉鱼食饲料鱼粮（升级版）	105.00	105.00	1.0
23654	2019-11-15 10:43:22	https://item.taobao.com/item.htm?id=597889386973	597889386973	益口红血鹦鹉饲料增红增色专用粮观赏鱼食招财鱼鱼粮包邮	30.30	30.30	1.0
23655	2019-11-15 10:43:21	https://item.taobao.com/item.htm?id=534921855001	534921855001	帝溢红 血鹦鹉增红增色鱼食 高虾红素营养 上浮型5天增红鱼粮饲料	55.50	55.50	9.0
23656	2019-11-15 10:43:21	https://item.taobao.com/item.htm?id=604810272138	604810272138	虹立方血鹦鹉增红鱼粮财神罗汉饲料热带观赏鱼鱼食 不浑水	60.00	60.00	9.0

23339 rows × 24 columns

In [56]:

```
# 先通过eval()函数把每个字符串里面的字典取出来
list=[eval(i) for i in df["店铺评分"]]
list
```

Out[56]:

```
[{'描述相符': ['4.8', '持平0.30%'],
  '服务态度': ['4.8', '高于5.47%'],
  '物流服务': ['4.8', '高于1.50%']},
 {'描述相符': ['4.5', '低于5.52%'],
  '服务态度': ['4.7', '低于2.79%'],
  '物流服务': ['4.7', '低于1.53%']},
 {'描述相符': ['4.8', '高于34.13%'],
  '服务态度': ['4.9', '高于43.14%'],
  '物流服务': ['4.9', '高于32.40%']},
 {'描述相符': ['4.8', '高于30.77%'],
  '服务态度': ['4.9', '高于47.04%'],
  '物流服务': ['4.9', '高于43.80%']},
 {'描述相符': ['4.8', '高于35.26%'],
  '服务态度': ['4.9', '高于61.41%'],
  '物流服务': ['4.8', '持平0.48%']},
 {'描述相符': ['4.3', '低于9.67%'],
  '服务态度': ['4.4', '低于8.31%'],
  '物流服务': ['4.5', '低于6.18%']},
```

In [57]:

```
# 提取描述评分
df["描述相符打分"]=[float(i["描述相符"][0]) for i in list]
df["描述相符打分"]
```

```
/Users/wenying/opt/anaconda3/lib/python3.7/site-packages/ipykernel_lau
ncher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (h
ttp://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#re
turning-a-view-versus-a-copy)
```

Out[57]:

```
0        4.8
1        4.5
2        4.8
3        4.8
4        4.8
        ...
23652    4.8
23653    4.8
23654    4.7
23655    4.8
23656    4.9
Name: 描述相符打分, Length: 23339, dtype: float64
```

In [58]:

```
# 提取描述评分水平
df["描述评分水平"]=[i["描述相符"][1][:2] for i in list]
df["描述评分水平"]
```

```
/Users/wenying/opt/anaconda3/lib/python3.7/site-packages/ipykernel_lau
ncher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (h
ttp://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#re
turning-a-view-versus-a-copy)
```

Out[58]:

```
0        持平
1        低于
2        高于
3        高于
4        高于
         ..
23652    高于
23653    高于
23654    低于
23655    持平
23656    高于
Name: 描述评分水平, Length: 23339, dtype: object
```

In [59]:

```
# 相同的方式提取服务评分和服务评分水平
```

In [60]:

```
df["服务态度打分"]=[float(i["服务态度"][0]) for i in list]
df["服务态度打分"]
```

```
/Users/wenying/opt/anaconda3/lib/python3.7/site-packages/ipykernel_lau
ncher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (h
ttp://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#re
turning-a-view-versus-a-copy)
  """Entry point for launching an IPython kernel.
```

Out[60]:

```
0        4.8
1        4.7
2        4.9
3        4.9
4        4.9
        ...
23652    4.8
23653    4.9
23654    4.8
23655    4.8
23656    4.9
Name: 服务态度打分, Length: 23339, dtype: float64
```

In [61]:

```
df["服务态度水平"]=[i["服务态度"][1][:2] for i in list]
df["服务态度水平"]
```

```
/Users/wenying/opt/anaconda3/lib/python3.7/site-packages/ipykernel_lau
ncher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (h
ttp://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#re
turning-a-view-versus-a-copy)
  """Entry point for launching an IPython kernel.
```

Out[61]:

```
0        高于
1        低于
2        高于
3        高于
4        高于
        ..
23652    高于
23653    高于
23654    低于
23655    持平
23656    高于
Name: 服务态度水平, Length: 23339, dtype: object
```

In [62]:

```
# 相同的方式提取物流评分及物流评分水平
```

In [63]:

```
df["物流服务打分"]=[float(i["物流服务"][0]) for i in list]
df["物流服务打分"]
```

```
/Users/wenying/opt/anaconda3/lib/python3.7/site-packages/ipykernel_lau
ncher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (h
ttp://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#re
turning-a-view-versus-a-copy)
  """Entry point for launching an IPython kernel.
```

Out[63]:

```
0        4.8
1        4.7
2        4.9
3        4.9
4        4.8
        ...
23652    4.8
23653    4.8
23654    4.8
23655    4.8
23656    4.9
Name: 物流服务打分, Length: 23339, dtype: float64
```

In [64]:

```
df["物流服务水平"]=[i["物流服务"][1][:2] for i in list]
df["物流服务水平"]
```

```
/Users/wenying/opt/anaconda3/lib/python3.7/site-packages/ipykernel_lau
ncher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (h
ttp://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#re
turning-a-view-versus-a-copy)
  """Entry point for launching an IPython kernel.
```

Out[64]:

```
0        高于
1        低于
2        高于
3        高于
4        持平
        ..
23652    高于
23653    高于
23654    持平
23655    持平
23656    高于
Name: 物流服务水平, Length: 23339, dtype: object
```

In [65]:

```
# 创建平均评分 数据列，即将描述评分、服务评分和物流评分计算平均值
```

In [66]:

```
df["平均评分"]=(df["描述相符打分"]+df["服务态度打分"]+df["物流服务打分"])/3
df["平均评分"]
```

```
/Users/wenying/opt/anaconda3/lib/python3.7/site-packages/ipykernel_lau
ncher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (h
ttp://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#re
turning-a-view-versus-a-copy)
  """Entry point for launching an IPython kernel.
```

Out[66]:

```
0        4.800000
1        4.633333
2        4.866667
3        4.866667
4        4.833333
           ...
23652    4.800000
23653    4.833333
23654    4.766667
23655    4.800000
23656    4.900000
Name: 平均评分, Length: 23339, dtype: float64
```

In [67]:

```
# 处理宝贝收藏数，没有缺失值
# 查看数据
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 23339 entries, 0 to 23656
Data columns (total 31 columns):
爬取时间        23339 non-null object
爬取链接        23339 non-null object
商品ID        23339 non-null int64
商品名称        23339 non-null object
商品现价        23339 non-null float64
商品原价        23339 non-null float64
月销量         23339 non-null float64
评论数         23339 non-null float64
发货地址        23339 non-null object
商品发布时间      23339 non-null object
商品规格        23339 non-null object
商品库存        23339 non-null int64
店铺名称        23339 non-null object
店铺url       23339 non-null object
商品参数        23339 non-null object
商品sku详情     23339 non-null object
商品链接        23339 non-null object
商品详情        23339 non-null object
店铺评分        23339 non-null object
宝贝收藏数       23339 non-null object
一级分类        23339 non-null object
二级分类        23339 non-null object
三级分类        23339 non-null object
现价等级        23339 non-null object
描述相符打分      23339 non-null float64
描述评分水平      23339 non-null object
服务态度打分      23339 non-null float64
服务态度水平      23339 non-null object
物流服务打分      23339 non-null float64
物流服务水平      23339 non-null object
平均评分        23339 non-null float64
dtypes: float64(8), int64(2), object(21)
memory usage: 5.7+ MB
```

In [68]:

```
df["宝贝收藏数"]
```

Out[68]:

```
0          957
1            4
2        13617
3         8764
4          403
         ...
23652       28
23653       84
23654       12
23655       12
23656        9
Name: 宝贝收藏数, Length: 23339, dtype: object
```

In [69]:

```
# 强制将字符串转化为整数
df["宝贝收藏数"]=[int(i) for i in df["宝贝收藏数"]]
df["宝贝收藏数"]
```

```
/Users/wenying/opt/anaconda3/lib/python3.7/site-packages/ipykernel_lau
ncher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (h
ttp://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#re
turning-a-view-versus-a-copy)
```

Out[69]:

```
0          957
1            4
2        13617
3         8764
4          403
         ...
23652       28
23653       84
23654       12
23655       12
23656        9
Name: 宝贝收藏数, Length: 23339, dtype: int64
```

In [ ]:

In [70]:

```
# 梳理序号
df.index=range(len(df))
df.index
```

Out[70]:

```
RangeIndex(start=0, stop=23339, step=1)
```

In [71]:

```
# 创建销售额数据列,提示:商品现价数据列* 评论数据列
df["销售额数据"]=df["商品现价"]*df["评论数"]
df["销售额数据"]
```

```
/Users/wenying/opt/anaconda3/lib/python3.7/site-packages/ipykernel_lau
ncher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (h
ttp://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#re
turning-a-view-versus-a-copy)
```

Out[71]:

```
0         10444.5
1             0.0
2        233151.0
3          4498.0
4          4474.5
            ...
23334      2037.0
23335       840.0
23336       272.7
23337      1276.5
23338       300.0
Name: 销售额数据, Length: 23339, dtype: float64
```

In [72]:

```
# 创建商品折扣数据 即用商品现价/商品原价
df["商品折扣数据"]=df["商品现价"]/df["商品原价"]
df["商品折扣数据"]
```

```
/Users/wenying/opt/anaconda3/lib/python3.7/site-packages/ipykernel_lau
ncher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (h
ttp://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#re
turning-a-view-versus-a-copy)
```

Out[72]:

```
0        0.700141
1        0.479998
2        1.000000
3        0.865000
4        1.000000
           ...
23334    1.000000
23335    1.000000
23336    1.000000
23337    1.000000
23338    1.000000
Name: 商品折扣数据, Length: 23339, dtype: float64
```

In [73]:

```
# 将数据保存为csv格式
```

In [74]:

```
df.to_excel("淘宝宠物消费数据.xlsx")
```

```
/Users/wenying/opt/anaconda3/lib/python3.7/site-packages/xlsxwriter/wo
rksheet.py:939: UserWarning: Ignoring URL 'https://item.taobao.com/ite
m.htm?id=606278904927' since it exceeds Excel's limit of 65,530 URLS p
er worksheet.
  "65,530 URLS per worksheet." % force_unicode(url))
/Users/wenying/opt/anaconda3/lib/python3.7/site-packages/xlsxwriter/wo
rksheet.py:939: UserWarning: Ignoring URL 'https://item.taobao.com/ite
m.htm?id=585400418564' since it exceeds Excel's limit of 65,530 URLS p
er worksheet.
  "65,530 URLS per worksheet." % force_unicode(url))
/Users/wenying/opt/anaconda3/lib/python3.7/site-packages/xlsxwriter/wo
rksheet.py:939: UserWarning: Ignoring URL 'https://item.taobao.com/ite
m.htm?id=557539013472' since it exceeds Excel's limit of 65,530 URLS p
er worksheet.
  "65,530 URLS per worksheet." % force_unicode(url))
/Users/wenying/opt/anaconda3/lib/python3.7/site-packages/xlsxwriter/wo
rksheet.py:939: UserWarning: Ignoring URL 'https://item.taobao.com/ite
m.htm?id=547465859008' since it exceeds Excel's limit of 65,530 URLS p
er worksheet.
```

In [75]:

```
data=pd.read_excel("淘宝宠物消费数据.xlsx")
```

2. 可视化

2.1 Tableau 可视化

- 如果月销量等数值类型的数据出现在维度里面，将其拖放到度量中
- 如果描述评分水平、服务评分水平、物流评分水平的数据出现在度量中，将其拖放到维度，类型修改为字符串
- 将评论数改名为总销量

综合分析

- 评分的分布分析（将评分从度量值移动到维度，排除掉评分为0的数据列）
- 不同价格等级的商品记录数和销售额的构成饼图
- 其他自选

类别数据有非常多的维度可以分析研究，比如

- 研究一级分类的总销量、总销售额、总记录数、平均价格、库存、收藏量、折扣比、评分情况等
- 研究二级分类的总销量、总销售额、总记录数、平均价格、库存、收藏量、折扣比等
  注：二级分类中的活体生物是指水族生物中的，猫猫狗狗是各种猫狗食品、帐篷，喵世界是指活体猫及其用品，汪世界是指活体狗及其用品。
- 研究二级分类的总销量、总销售额、总记录数、平均价格、库存、收藏量、折扣比等

In [76]:

```
## 一级分类都有一些什么分类
```

In [77]:

```
data.columns
```

Out[77]:

```
Index(['Unnamed: 0', '爬取时间', '爬取链接', '商品ID', '商品名称', '商品现
价', '商品原价', '月销量',
       '评论数', '发货地址', '商品发布时间', '商品规格', '商品库存', '店铺名
称', '店铺url', '商品参数',
       '商品sku详情', '商品链接', '商品详情', '店铺评分', '宝贝收藏数', '一级
分类', '二级分类', '三级分类',
       '现价等级', '描述相符打分', '描述评分水平', '服务态度打分', '服务态度水
平', '物流服务打分', '物流服务水平',
       '平均评分', '销售额数据', '商品折扣数据'],
      dtype='object')
```

In [78]:

```
data["一级分类"].unique()
```

Out[78]:

```
array(['水族世界', '猫猫狗狗', '奇趣宠物'], dtype=object)
```

In [79]:

```
data["二级分类"].unique()
```

Out[79]:

```
array(['活体生物', '猫猫狗狗', '喵世界', '汪世界', '鱼粮鱼药', '水族器材', '水
草', '造景', '水质维护',
       '仓鼠类及其它小宠'], dtype=object)
```

In [80]:

```
data["三级分类"].unique()
```

Out[80]:

```
array(['热带鱼', '水母', '猫主粮', '猫抓板', '斗鱼', '布偶猫', '香波/浴液',
'狗主粮', '窝/帐篷',
       '加菲猫', '比格犬', '服装', '魔法鱼', '孔雀鱼', '灯科鱼', '逗猫棒',
'丝瓜络玩具', '猫爬架',
       '金吉拉', '波斯猫', '折耳猫', '英国短毛猫', '套装玩具', '漏食球', '飞
盘', '绳结', '发声玩具',
       '橡胶球', '吉娃娃', '比熊', '博美', '泰迪', '玩具', '猫零食', '增红鱼
粮', '加热棒', '睡莲类',
       '仿真水草', '锦鲤', '虾粮', '鱼饲料', '潜水泵', '风扇', '造流泵', '水
草灯', '水妖精', '过滤器',
       '氧气泵', '鱼缸', '假山', '木化石', '杜鹃根', '陶罐', '沉木', '造景石
料', '水草泥', '水生蕨类',
       '绿藻球', '水草套餐', '矮珍珠', '铁皇冠', '小水榕', '前景草', '金鱼',
'罗汉鱼', '龙鱼', '虾螺',
       '底栖鱼', '硝化细菌', '测试纸', '香猪', '饲料/零食', '除藻剂', '宠物
貂', '豚鼠', '飞鼠',
       '宠物狐狸', '龙猫', '仓鼠', '基底肥', '液肥', '除氯杀菌', '水质调理',
'水质检测', '颗粒鱼粮',
       '薄片鱼粮', '开口鱼粮', '饵干', '乌龟饲料'], dtype=object)
```

In [81]:

```
data.columns
```

Out[81]:

```
Index(['Unnamed: 0', '爬取时间', '爬取链接', '商品ID', '商品名称', '商品现
价', '商品原价', '月销量',
       '评论数', '发货地址', '商品发布时间', '商品规格', '商品库存', '店铺名
称', '店铺url', '商品参数',
       '商品sku详情', '商品链接', '商品详情', '店铺评分', '宝贝收藏数', '一级
分类', '二级分类', '三级分类',
       '现价等级', '描述相符打分', '描述评分水平', '服务态度打分', '服务态度水
平', '物流服务打分', '物流服务水平',
       '平均评分', '销售额数据', '商品折扣数据'],
      dtype='object')
```

发货地址分析

发货地址分析

- 发货地商品记录数、销售额、销量等的条形图或地图

店铺分析

- 销售额最好的10家店铺销售额条形图
- 销售额最好的10家店铺价格等级与一级分类和二级分类堆积柱形图
- 销售额最高的10家店铺的平均折扣力度比较
- 其他自选

单品分析

- 最热销（销量）的商品TOP10
- 销售额最高的商品TOP10
- 人气最高（收藏数）的商品TOP10
- 库存最高的10款商品
- 价格最高的TOP10商品
- 其他自选

2.2. 使用wordArt制作词云

- 商品名称词云
- 商品发货地址词云
- 其他自选

In [82]:

```
## 使用商品名称列数据制作词云图
```

In [83]:

```
data["商品名称"]
```

Out[83]:

```
0                    蛇仔鱼苦力泥鳅观赏鱼热带鱼除蛋白虫涡虫虾缸搭档清洁鱼易养鱼
1                      水母活物水族箱宠物水母活  淡水观赏水母缸活小型水族箱
2                     红绿灯鱼活体群游灯鱼热带观赏鱼小型灯科鱼孔雀鱼活体饲料鱼
3                    水母活体套装迷你音乐玻璃鱼缸赤月情人节礼物海月水母生日礼物
4                   观赏活体海月水母赤月水母生日礼物七夕情人活体宠物倒立水母包邮
                                  ...
23334              壹品红血鹦鹉增红鱼粮  红鹦鹉增色鱼饲料地图招财鱼食一品红鱼食
23335      海神  血鹦鹉增红专用粮500g/1000g血鹦鹉鱼食饲料鱼粮（升级版）
23336                  益口红血鹦鹉饲料增红增色专用粮观赏鱼食招财鱼鱼粮包邮
23337          帝溢红  血鹦鹉增红增色鱼食  高虾红素营养  上浮型5天增红鱼粮饲料
23338               虹立方血鹦鹉增红鱼粮  财神罗汉饲料热带观赏鱼鱼食  不浑水
Name: 商品名称, Length: 23339, dtype: object
```

In [84]:

```
[i for i in data["商品名称"][:5]]
```

Out[84]:

```
['蛇仔鱼苦力泥鳅观赏鱼热带鱼除蛋白虫涡虫虾缸搭档清洁鱼易养鱼',
 '水母活物水族箱宠物水母活 淡水观赏水母缸活小型水族箱',
 '红绿灯鱼活体群游灯鱼热带观赏鱼小型灯科鱼孔雀鱼活体饲料鱼',
 '水母活体套装迷你音乐玻璃鱼缸赤月情人节礼物海月水母生日礼物',
 '观赏活体海月水母赤月水母生日礼物七夕情人活体宠物倒立水母包邮']
```

In [85]:

```
##将全部商品名称放入一个列表
names=[i for i in data["商品名称"]]
names
```

Out[85]:

```
['蛇仔鱼苦力泥鳅观赏鱼热带鱼除蛋白虫涡虫虾缸搭档清洁鱼易养鱼',
 '水母活物水族箱宠物水母活 淡水观赏水母缸活小型水族箱',
 '红绿灯鱼活体群游灯鱼热带观赏鱼小型灯科鱼孔雀鱼活体饲料鱼',
 '水母活体套装迷你音乐玻璃鱼缸赤月情人节礼物海月水母生日礼物',
 '观赏活体海月水母赤月水母生日礼物七夕情人活体宠物倒立水母包邮',
 '水母活非淡水活物宠物大小型夜光便宜无毒发光观赏迷你荧光小号',
 '澳洲斑点水母活体圆点水母长裙摆尾尾部稀有海洋馆水母缸水母品种',
 '水母粮食水母饲料喂食水母食料水母液体饲料水母吃的食物冰冻丰年',
 '水母，活的水母，巴布亚硝水母',
 '赤月水母活体宠物水母观赏海蜇红火水母生日礼物包邮水母缸水母展',
 '水母，天草水母，活的水母',
 '水母，水母活体，宠物，大西洋海刺水母',
 '水母丰年虾饵料/水母专用饵料/液体饲料/海月赤月水母专用饲料',
 '水母养殖盐，海水盐，人工海水专用盐，不可食用',
 '特价包邮活体赤月水母宠物水母人工繁殖情人节生日节日礼物',
 '海月水母活体观赏宠物赤月海月水母养殖宠物透明运输包活七夕礼物',
 '悦海水族 宠物活体赤月水母和海月水母专业海水500ml/瓶',
 '水母，水母仙子，斑点水母，珍珠水母，澳洲斑点水母，活的水母，',
```

In [86]:

```
for i in names:
    print(i)
```

蛇仔鱼苦力泥鳅观赏鱼热带鱼除蛋白虫涡虫虾缸搭档清洁鱼易养鱼
水母活物水族箱宠物水母活 淡水观赏水母缸活小型水族箱
红绿灯鱼活体群游灯鱼热带观赏鱼小型灯科鱼孔雀鱼活体饲料鱼
水母活体套装迷你音乐玻璃鱼缸赤月情人节礼物海月水母生日礼物
观赏活体海月水母赤月水母生日礼物七夕情人活体宠物倒立水母包邮
水母活非淡水活物宠物大小型夜光便宜无毒发光观赏迷你荧光小号
澳洲斑点水母活体圆点水母长裙摆尾尾部稀有海洋馆水母缸水母品种
水母粮食水母饲料喂食水母食料水母液体饲料水母吃的食物冰冻丰年
水母，活的水母，巴布亚硝水母
赤月水母活体宠物水母观赏海蜇红火水母生日礼物包邮水母缸水母展
水母，天草水母，活的水母
水母，水母活体，宠物，大西洋海刺水母
水母丰年虾饵料/水母专用饵料/液体饲料/海月赤月水母专用饲料
水母养殖盐，海水盐，人工海水专用盐，不可食用
特价包邮活体赤月水母宠物水母人工繁殖情人节生日节日礼物
海月水母活体观赏宠物赤月海月水母养殖宠物透明运输包活七夕礼物
悦海水族 宠物活体赤月水母和海月水母专业海水500ml/瓶
水母，水母仙子，斑点水母，珍珠水母，澳洲斑点水母，活的水母，
仿真水母鱼缸造景水母装饰水母水仿真透明荧光水母漂浮式软体水母

In [87]:

```
##将全部商品发货地址放入一个列表
address=[i for i in data["发货地址"]]
address
```

Out[87]:

```
['广东',
 '上海',
 '上海',
 '北京',
 '山东',
 '山西',
 '北京',
 '河北',
 '广东',
 '湖北',
 '广东',
 '广东',
 '山东',
 '广东',
 '湖北',
 '山东',
 '上海',
 '广东',
```

In [88]:

```
for i in address:
    print(i)
```

```
广东
上海
上海
北京
山东
山西
北京
河北
广东
湖北
广东
广东
山东
广东
湖北
山东
上海
广东
广东
```

In [89]:

```
#输出词云图
import jieba
import jieba.analyse
from wordcloud import WordCloud
from matplotlib.pyplot import plot,savefig
font="/Users/wenying/Desktop/NotoSansCJKsc-Black.otf"
wc= WordCloud(font_path=font,width = 3000, height = 1500,background_color = 'white')
plt.imshow(wc)
plt.axis("off")
plt.savefig("/Users/wenying/Desktop/3.jpg")
plt.show()
```

In [ ]:

3. 挖掘建模

In [90]:

```
#加载处理后的数据
```

In [91]:

```
data=pd.read_excel("淘宝宠物消费数据.xlsx")
```

In [92]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23339 entries, 0 to 23338
Data columns (total 34 columns):
Unnamed: 0    23339 non-null int64
爬取时间          23339 non-null object
爬取链接          23339 non-null object
商品ID          23339 non-null int64
商品名称          23339 non-null object
商品现价          23339 non-null float64
商品原价          23339 non-null float64
月销量           23339 non-null int64
评论数           23339 non-null int64
发货地址          23339 non-null object
商品发布时间        23339 non-null object
商品规格          23339 non-null object
商品库存          23339 non-null int64
店铺名称          23339 non-null object
店铺url         23339 non-null object
商品参数          23339 non-null object
商品sku详情       23339 non-null object
商品链接          18852 non-null object
商品详情          23339 non-null object
店铺评分          23339 non-null object
宝贝收藏数         23339 non-null int64
一级分类          23339 non-null object
二级分类          23339 non-null object
三级分类          23339 non-null object
现价等级          23339 non-null object
描述相符打分        23339 non-null float64
描述评分水平        23339 non-null object
服务态度打分        23339 non-null float64
服务态度水平        23339 non-null object
物流服务打分        23339 non-null float64
物流服务水平        23339 non-null object
平均评分          23339 non-null float64
销售额数据         23339 non-null float64
商品折扣数据        23339 non-null float64
dtypes: float64(8), int64(6), object(20)
memory usage: 6.1+ MB
```

In [ ]:

```

```

In [93]:

```
# 将评论数 名称修改为 总销量
data=data.rename(columns={"评论数":"总销量"})
```

In [94]:

```
data.describe()
```

Out[94]:

	Unnamed: 0	商品ID	商品现价	商品原价	月销量	总销量
count	23339.000000	2.333900e+04	23339.000000	23339.000000	23339.000000	23339.000000
mean	11669.000000	5.162326e+11	527.257630	643.392630	2.862890	1058.608938
std	6737.533302	1.744212e+11	2673.142516	2762.486887	2.603165	6308.695474
min	0.000000	6.879048e+07	0.080000	0.100000	0.000000	0.000000
25%	5834.500000	5.463030e+11	19.700000	22.700000	1.000000	11.000000
50%	11669.000000	5.776392e+11	45.000000	54.450000	2.000000	87.000000
75%	17503.500000	5.991617e+11	135.000000	191.402500	5.000000	508.000000
max	23338.000000	6.076637e+11	175000.000000	175000.000000	9.000000	333712.000000

In [95]:

```
data.describe().columns
```

Out[95]:

```
Index(['Unnamed: 0', '商品ID', '商品现价', '商品原价', '月销量', '总销量',
'商品库存', '宝贝收藏数',
       '描述相符打分', '服务态度打分', '物流服务打分', '平均评分', '销售额数
据', '商品折扣数据'],
      dtype='object')
```

In [96]:

```
# 取出所有的数值型变量，创建热力图，查看各变量之间的相关性,制作热力图
```

In [97]:

```
data.columns
```

Out[97]:

```
Index(['Unnamed: 0', '爬取时间', '爬取链接', '商品ID', '商品名称', '商品现
价', '商品原价', '月销量',
       '总销量', '发货地址', '商品发布时间', '商品规格', '商品库存', '店铺名
称', '店铺url', '商品参数',
       '商品sku详情', '商品链接', '商品详情', '店铺评分', '宝贝收藏数', '一级
分类', '二级分类', '三级分类',
       '现价等级', '描述相符打分', '描述评分水平', '服务态度打分', '服务态度水
平', '物流服务打分', '物流服务水平',
       '平均评分', '销售额数据', '商品折扣数据'],
      dtype='object')
```

In [98]:

```
Xvar=data[['商品现价','商品原价','月销量','商品库存','宝贝收藏数','描述相符打分','服务态度打分
```

In [99]:

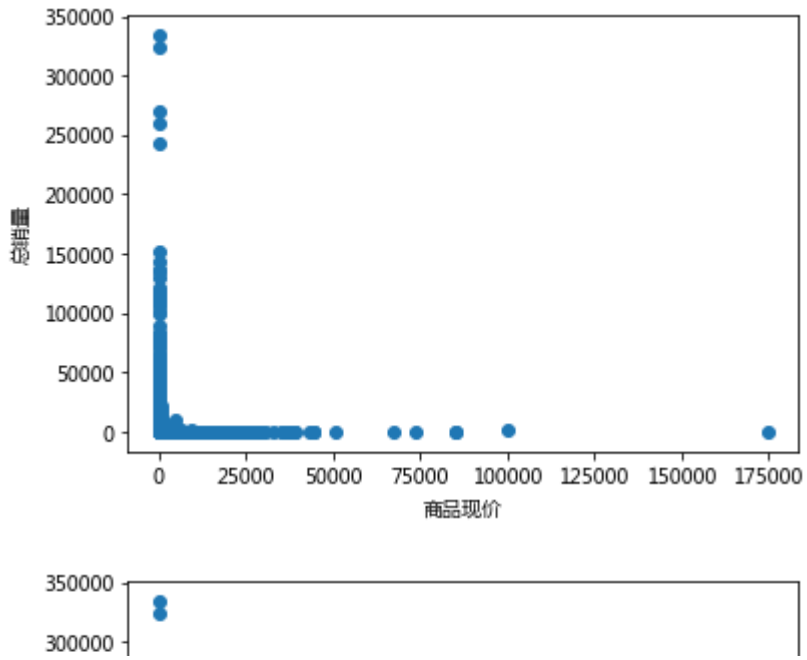
```
Yvar=data["总销量"]
```

In [100]:

```
import matplotlib.pyplot as plt
plt.rcParams["font.sans-serif"]=['Microsoft YaHei'] #正常显示中文标签
plt.rcParams["axes.unicode_minus"]=False #正常显示负号
```

In [101]:

```
#画散点图，检查有无异常值
for i in Xvar.columns:
    plt.xlabel(i)
    plt.ylabel("总销量")
    plt.scatter(Xvar[i],Yvar)
    plt.show()
```



In [102]:

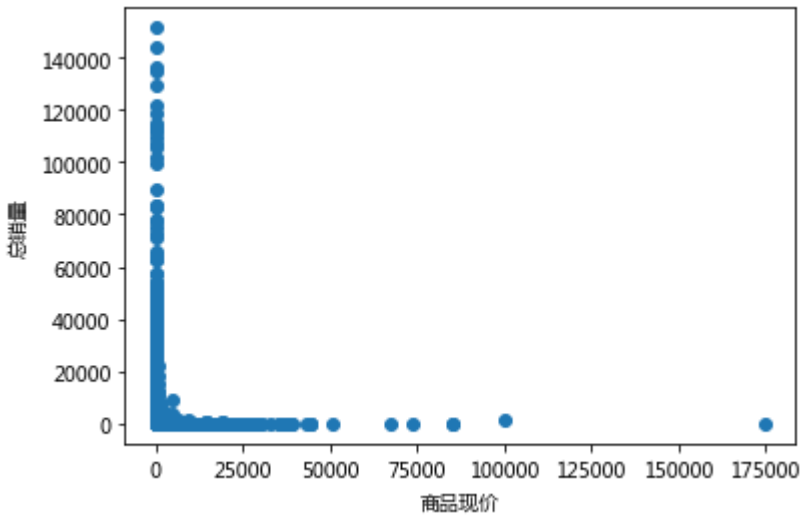
```
##剔除异常值：把总销量在20万以上的值排除
data=data[data["总销量"]<200000]
```

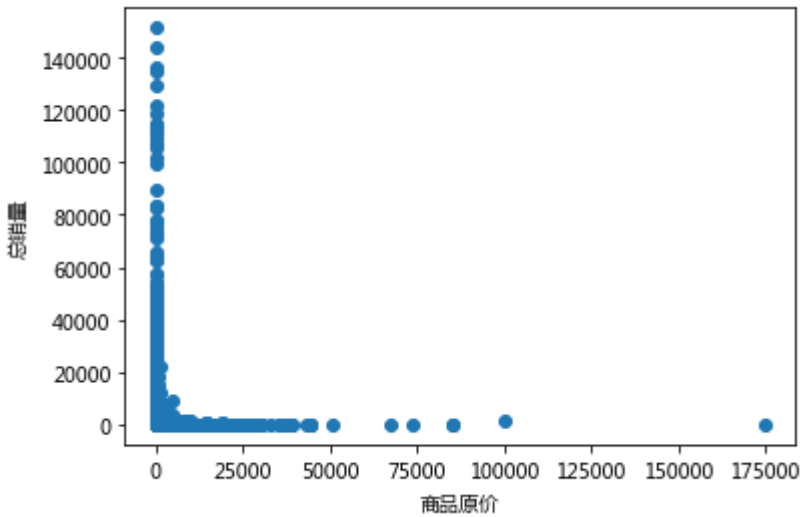
In [103]:

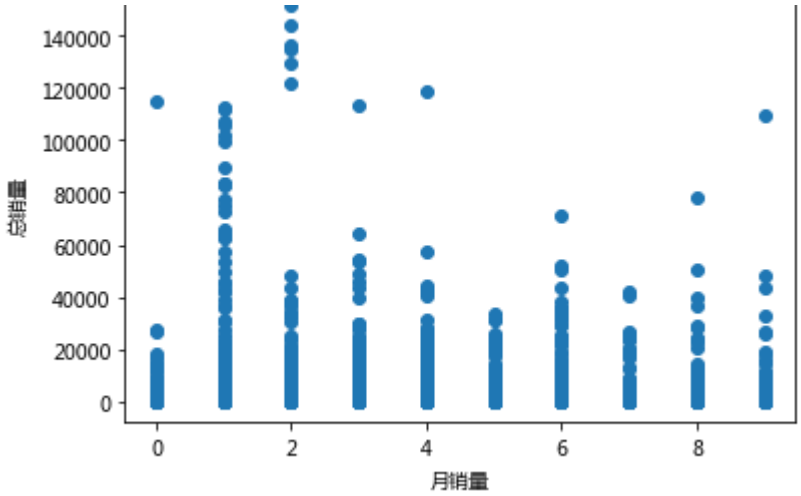
```
##重新梳理索引号
data.index=range(len(data))
```

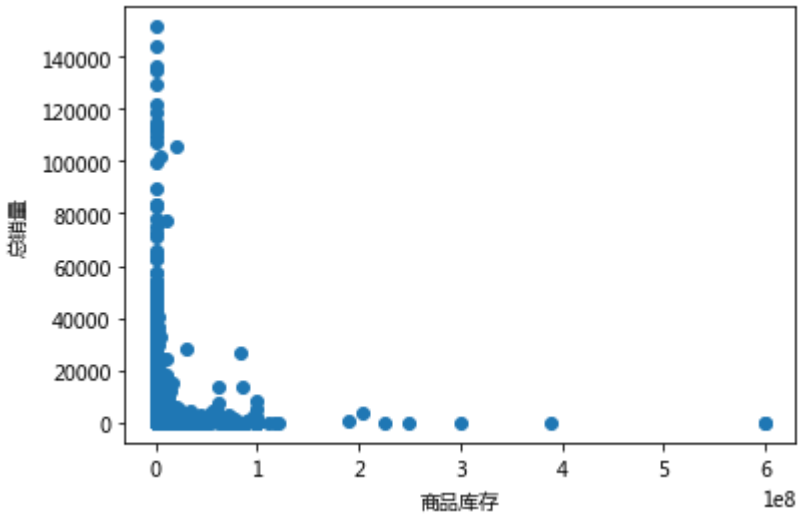
In [104]:

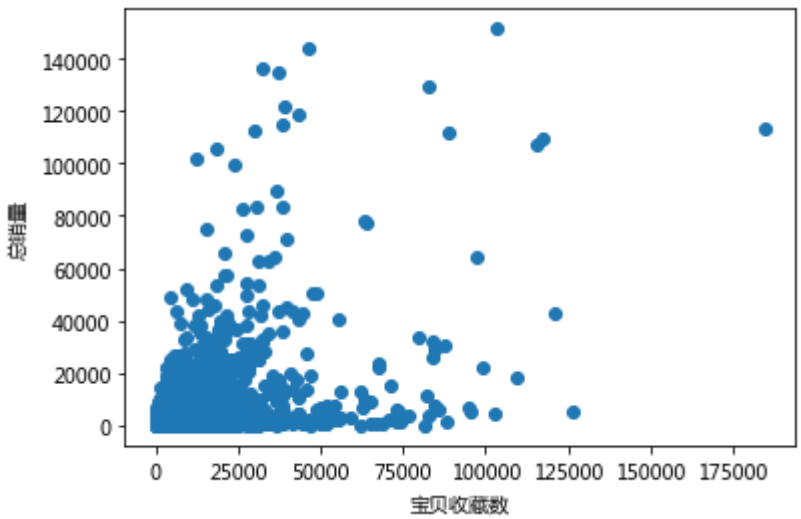
```
##重新运行7个散点图
Xvar=data[['商品现价','商品原价','月销量','商品库存','宝贝收藏数','描述相符打分','服务态度打分
Yvar=data["总销量"]

for i in Xvar.columns:
    plt.xlabel(i)
    plt.ylabel("总销量")
    plt.scatter(Xvar[i],Yvar)
    plt.show()
```

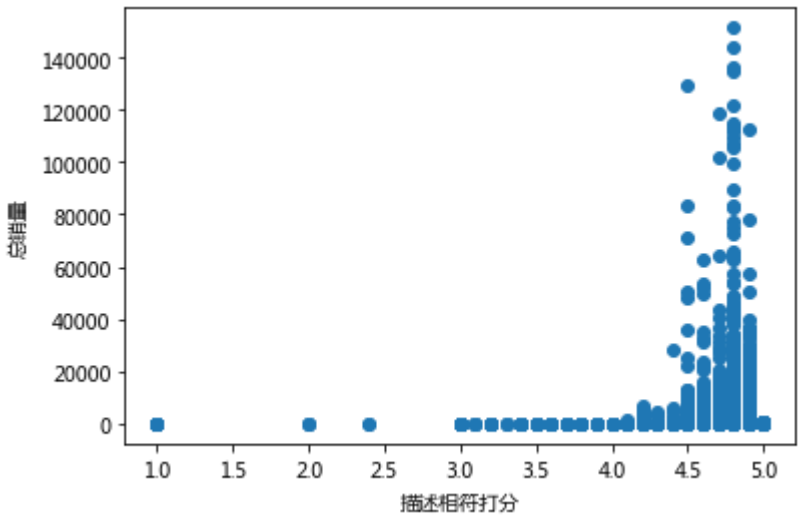


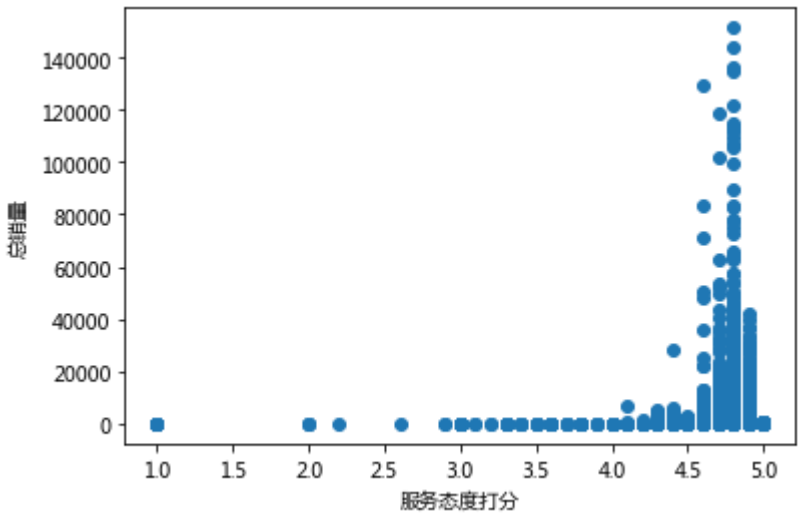


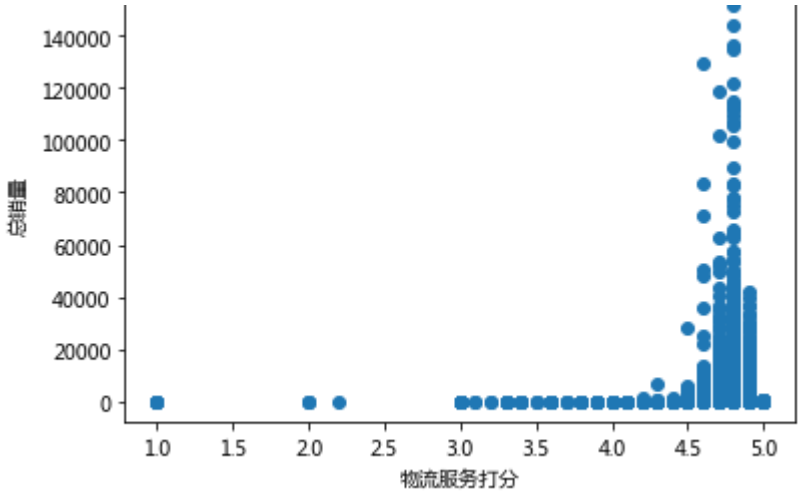

总销量
月销量

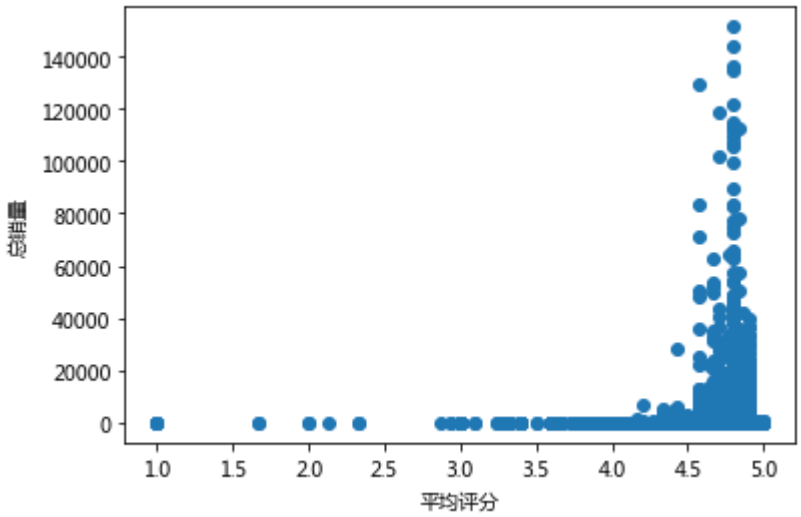

总销量
商品库存
1e8

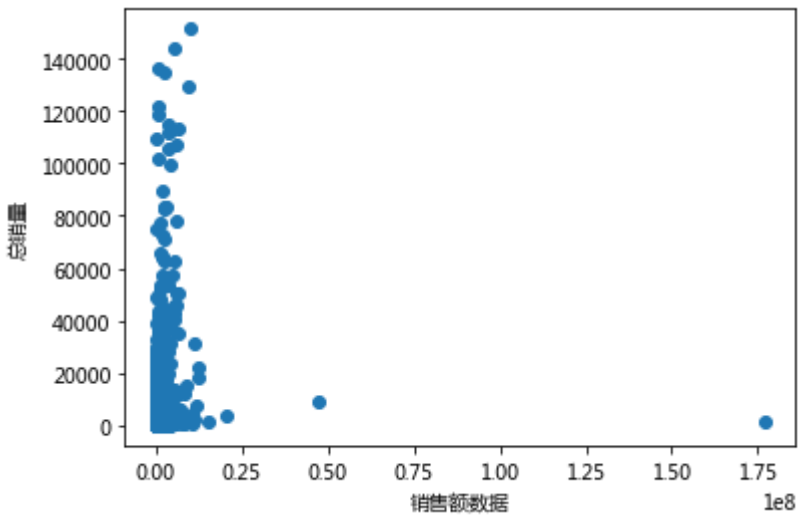

总销量
宝贝收藏数

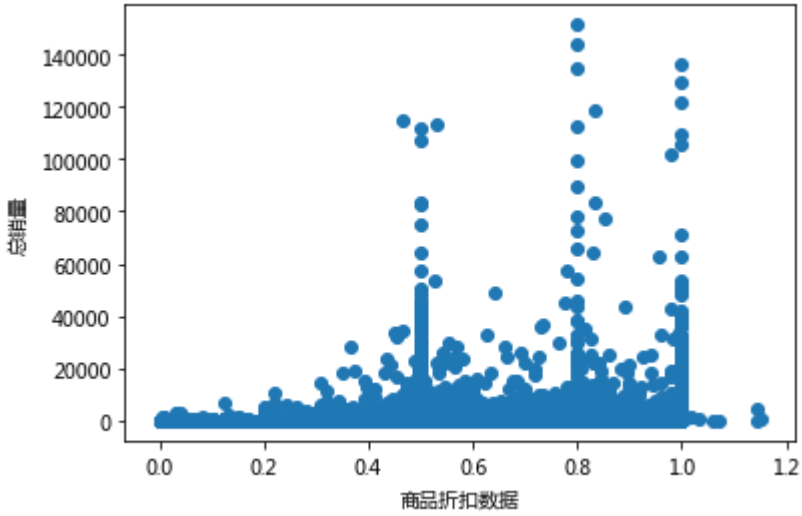

140000
120000
100000
80000
60000
40000
20000
0
总销量
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
描述相符打分


140000
120000
100000
80000
60000
40000
20000
0
总销量
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
服务态度打分


总销量
物流服务打分


总销量
平均评分


总销量
销售额数据
1e8



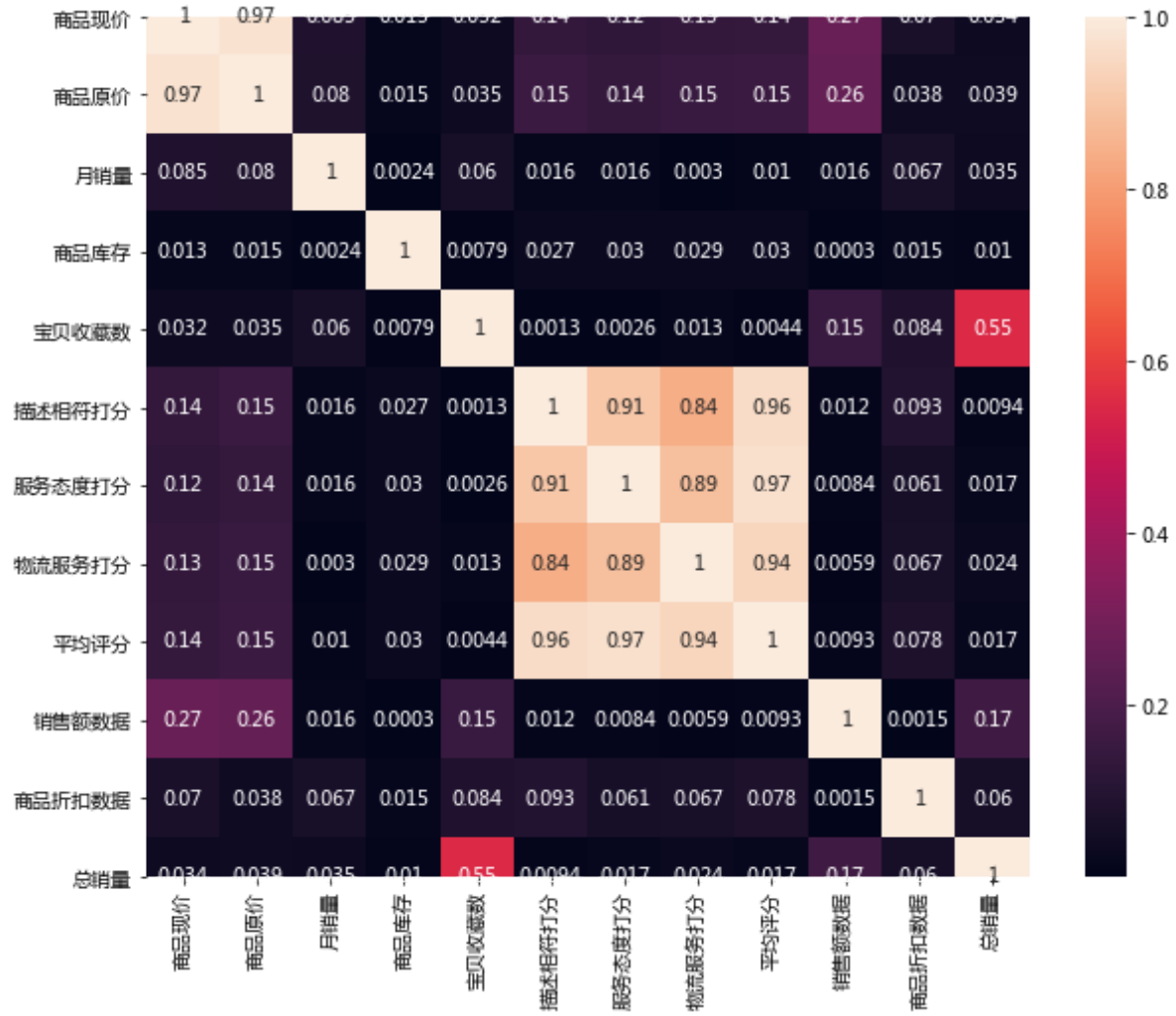
1.2 画热力图来展示自变量与因变量以及自变量之间的相关系数

In [105]:

```
t=data[['商品现价','商品原价','月销量','商品库存','宝贝收藏数','描述相符打分','服务态度打分','
```

In [106]:

```
import seaborn as sns
plt.figure(figsize=(10,8))
sns.heatmap(np.abs(t.corr()),annot=True)
plt.savefig("/Users/wenying/Desktop/4.jpg")
plt.show()
```



1.3 对自变量进行归一化处理

In [107]:

```
Xvar.describe()
```

Out[107]:

	商品现价	商品原价	月销量	商品库存	宝贝收藏数	描述相符打分
count	23334.000000	23334.000000	23334.000000	2.333400e+04	23334.000000	23334.000000
mean	527.363242	643.520411	2.862604	6.674909e+05	1585.580355	4.795033
std	2673.419170	2762.769049	2.603331	8.621999e+06	5683.395829	0.193785
min	0.080000	0.100000	0.000000	0.000000e+00	0.000000	1.000000
25%	19.700000	22.700000	1.000000	2.860000e+02	18.000000	4.700000
50%	45.000000	54.450000	2.000000	2.981500e+03	144.000000	4.800000
75%	135.337500	191.701250	5.000000	2.323225e+04	858.000000	4.900000
max	175000.000000	175000.000000	9.000000	5.999998e+08	184816.000000	5.000000

In [108]:

```
from sklearn.preprocessing import StandardScaler
std=StandardScaler()
```

In [109]:

```
Xstd=std.fit_transform(Xvar)
Xstd
```

```
/Users/wenying/.local/lib/python3.7/site-packages/sklearn/preprocessin
g/data.py:645: DataConversionWarning: Data with input dtype int64, flo
at64 were all converted to float64 by StandardScaler.
  return self.partial_fit(X, y)
/Users/wenying/.local/lib/python3.7/site-packages/sklearn/base.py:464:
DataConversionWarning: Data with input dtype int64, float64 were all c
onverted to float64 by StandardScaler.
  return self.fit(X, **fit_params).transform(X)
```

Out[109]:

```
array([[-0.19356274, -0.22781271,  0.05277816, ..., -0.06729402,
        -0.0703544 , -0.64912426],
       [-0.10624571, -0.04943712, -0.7154848 , ..., -1.05065994,
        -0.07845681, -1.54900039],
       [-0.17687962, -0.21320386,  1.20517259, ...,  0.32605234,
         0.10241201,  0.57659977],
       ...,
       [-0.1859319 , -0.22196338, -0.7154848 , ..., -0.2639672 ,
        -0.07824526,  0.57659977],
       [-0.17650556, -0.2128419 ,  2.35756703, ..., -0.06729402,
        -0.07746655,  0.57659977],
       [-0.17482229, -0.21121306,  2.35756703, ...,  0.52272553,
        -0.07822408,  0.57659977]])
```

1.4 对归一化结果进行建模

In [110]:

```
#看线性回归结果
import statsmodels.api as sm
Y=Yvar.values
X=Xstd
X=sm.add_constant(X)
lm=sm.OLS(Y,X).fit()
print("========多元线性回归结果==========")
print(lm.summary())
```

```
========多元线性回归结果==========
                            OLS Regression Results
==============================================================================
========
Dep. Variable:                      y   R-squared:
0.311
Model:                            OLS   Adj. R-squared:
0.311
Method:                 Least Squares   F-statistic:
1054.
Date:                Wed, 01 Jan 2020   Prob (F-statistic):
0.00
Time:                        10:08:16   Log-Likelihood:            -2.
2606e+05
No. Observations:               23334   AIC:
4.521e+05
Df Residuals:                   23323   BIC:
4.522e+05
Df Model:                          10
```

In [111]:

```
Xvar.columns
```

Out[111]:

```
Index(['商品现价', '商品原价', '月销量', '商品库存', '宝贝收藏数', '描述相符打
分', '服务态度打分', '物流服务打分',
       '平均评分', '销售额数据', '商品折扣数据'],
      dtype='object')
```

分析： 从多元线性回归结果来看，R值为0.311，说明因变量和自变量间的相关性不太好，F的值为0，小于0.05，符合要求； 从x1-x11的p值来看，x1、x2、x5、x6、x10、x11和常数项的p值均小于0.05，满足要求； 相关系数coef来看，x5（宝贝收藏数）对总销量的影响最大，其次是x2(商品原价)、x10（销售额数据）,x1（商品现价）,x6（描述相符打分）,x11（商品折扣数据）。宝贝收藏数越大，销售额数据越高、描述相符打分越高的店铺总销量越高；商品价格越高、商品折扣越大对店铺销量有不利影响。

1.5 降维：主成分分析

In [112]:

```
from sklearn.decomposition import PCA
```

In [113]:

```
#设置主成分的数量，就是说要降到几个维度
pca_model=PCA(n_components=3)
#给出XY
Y=Yvar.values
X=Xstd
#执行PCA方法
pca_model.fit(X)
#取得降维后的x
X_pca=pca_model.transform(X)
#回归的固定代码
X_pca=sm.add_constant(X_pca)
lm=sm.OLS(Y,X_pca).fit()
print("因变量：总销量")
print("===========主成分回归分析结果(降维)=============")
print(lm.summary())
```

```
因变量：总销量
===========主成分回归分析结果(降维)=============
                            OLS Regression Results
==============================================================================
========
Dep. Variable:                      y   R-squared:
0.192
Model:                            OLS   Adj. R-squared:
0.192
Method:                 Least Squares   F-statistic:
1853.
Date:                Wed, 01 Jan 2020   Prob (F-statistic):
0.00
Time:                        10:08:17   Log-Likelihood:            -2.
2791e+05
No. Observations:               23334   AIC:
4.558e+05
Df Residuals:                   23330   BIC:
4.559e+05
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
========
                 coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
--------
const        997.4889     27.653     36.072      0.000     943.288
1051.690
x1            46.1425     14.090      3.275      0.001      18.525
73.760
x2            13.0148     19.402      0.671      0.502     -25.014
51.044
x3          1884.8355     25.304     74.488      0.000    1835.238
1934.433
==============================================================================
========
Omnibus:                    43303.142   Durbin-Watson:
1.348
Prob(Omnibus):                  0.000   Jarque-Bera (JB):        11349
9836.821
Skew:                          13.822   Prob(JB):
0.00
```

```
Kurtosis:                       343.551   Cond. No.
1.96
==============================================================================
========

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
```

In [114]:

```
Xvar.columns
```

Out[114]:

```
Index(['商品现价', '商品原价', '月销量', '商品库存', '宝贝收藏数', '描述相符打
分', '服务态度打分', '物流服务打分',
       '平均评分', '销售额数据', '商品折扣数据'],
      dtype='object')
```

In [115]:

```
np.round(pca_model.components_,2)
```

Out[115]:

```
array([[-0.15, -0.16,  0.  ,  0.02,  0.  , -0.48, -0.49, -0.48, -0.5 ,
        -0.03,  0.05],
       [ 0.65,  0.64, -0.11, -0.  , -0.  , -0.1 , -0.11, -0.1 , -0.11,
         0.33,  0.06],
       [-0.07, -0.03,  0.43, -0.  ,  0.66,  0.01, -0.01, -0.02, -0.01,
         0.41, -0.44]])
```

x1=-0.15*商品现价*-0.16商品原价+0.02*商品库存*-0.48描述相符打分-0.49*服务态度打分*-0.48物流服务打分-0.5平均评分-0.03*销售额数据*+0.05商品折扣数据 x3=-0.07*商品现价*-0.03商品原价+0.43*月销量*+0.66宝贝收藏数+0.01*描述相符打分*-0.01服务态度打分-0.02*物流服务打分*-0.01平均评分+0.41*销售额数据*-0.44商品折扣数据
y=46.1425*x1+1884.8355 *x3+997.4889

构建情感分析和LDA主题模型

可以从以下方向中选择一个或多个进行分析，情感分析可以随机抽取出500或者1000条商品评论进行分析，LDA主题模型的评论数量建议不超过5000条（如果超过该数量可以随机抽取）

- 单品销售额TOP10的商品情感得分和LDA主题分析
- 销售额TOP10店铺情感得分和LDA主题分析
- 所有评分水平都是“低于”的店铺的情感得分和LDA主题分析
- 还可以从其他角度进行文本分析，比如了解布偶猫、猫零食、魔法鱼等类别商品

销售额Top10店铺情感得分和LDA主题分析

In [116]:

```
#在jupyter中取出销售额TOP10店铺的参考代码
```

In [117]:

```
df = data.groupby('店铺名称').sum().sort_values(by = ['销售额数据'],ascending =False)
df[:10]
```

Out[117]:

店铺名称	Unnamed: 0	商品ID	商品现价	商品原价	月销量	总销量	商品库存	宝贝收藏数	
马来西亚新轩龍鱼繁殖场	51369	1159853567894	148121.500	148268.500	10	1865	5954	10418	
宜兴渔场	60695	1705254138769	8262.500	8262.500	2	11041	3658088	11851	
意品宠物用品专营店	536944	16200093963507	2066.045	2555.035	129	1263655	366825	608722	2
sunsun森森旗舰店	587312	20343616670694	4302.700	8503.900	170	498137	26635416	489280	2
yee宠物用品旗舰店	457473	13384450553223	1758.850	3491.600	80	629022	383313	331252	1
上海大渔夫水族	124546	7731893365247	3653.500	3653.500	43	81042	785891	65870	
爱信水族	739456	19996107657160	2142.685	2142.685	218	273546	187516	242786	2
抱紧我的小鱼干	25577	1766128633617	5381.800	5381.800	7	6493	6177	753	
宠冠宠物用品专营店	212292	2192612297182	596.330	970.535	63	341948	1106697	257547	
本森水族用品	286296	12030906241374	2718.150	5403.900	85	88969	1799713	225777	1

In [118]:

```
shops = df[:10].index.tolist()
shops
```

Out[118]:

```
['马来西亚新轩龍鱼繁殖场',
 '宜兴渔场',
 '意品宠物用品专营店',
 'sunsun森森旗舰店',
 'yee宠物用品旗舰店',
 '上海大渔夫水族',
 '爱信水族',
 '抱紧我的小鱼干',
 '宠冠宠物用品专营店',
 '本森水族用品']
```

In [119]:

```
# 接下来和课上讲的不同品牌的手机的情感分析和LDA主题模型分析一样，只需要把店铺当做手机品牌来处理就可
```

In [120]:

```
## ?为什么变少了
df.columns
```

Out[120]:

```
Index(['Unnamed: 0', '商品ID', '商品现价', '商品原价', '月销量', '总销量',
'商品库存', '宝贝收藏数',
       '描述相符打分', '服务态度打分', '物流服务打分', '平均评分', '销售额数
据', '商品折扣数据'],
      dtype='object')
```

In [121]:

```
data.columns
```

Out[121]:

```
Index(['Unnamed: 0', '爬取时间', '爬取链接', '商品ID', '商品名称', '商品现
价', '商品原价', '月销量',
       '总销量', '发货地址', '商品发布时间', '商品规格', '商品库存', '店铺名
称', '店铺url', '商品参数',
       '商品sku详情', '商品链接', '商品详情', '店铺评分', '宝贝收藏数', '一级
分类', '二级分类', '三级分类',
       '现价等级', '描述相符打分', '描述评分水平', '服务态度打分', '服务态度水
平', '物流服务打分', '物流服务水平',
       '平均评分', '销售额数据', '商品折扣数据'],
      dtype='object')
```

In [122]:

```
comment_form=pd.read_excel("宠物商品评论信息.xlsx")
```

In [123]:

```
comment_form.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 356610 entries, 0 to 356609
Data columns (total 4 columns):
Unnamed: 0    356610 non-null int64
时间            356610 non-null object
url           356610 non-null int64
评论            356610 non-null object
dtypes: int64(2), object(2)
memory usage: 10.9+ MB
```

In [124]:

```
comment_form
```

Out[124]:

	Unnamed: 0	时间	url	评论
0	0	2019-12-15 13:45:48	18650288040	鱼收到了，很漂亮很活跃，养了很多天了，开始进食了。
1	1	2019-12-11 19:24:50	18650288040	特意养了一段时间才评价，鱼儿不错，很活波，好养活
2	2	2019-12-07 15:03:09	18650288040	越来越活泼了
3	3	2019-12-10 21:55:26	18650288040	物流快，包装仔细，🐟 活泼漂亮，很喜欢的
4	4	2019-12-04 17:19:47	18650288040	鱼很活跃，特意过了几天再来评论，物流快速，包装精致！
...	...	...	...	...
356605	354511	2019-07-10 13:29:26	596988278783	好胖的一只 刚刚到家就乱跑
356606	354512	2019-09-28 14:23:26	596988278783	自己是开宠物店的，第一次尝试在网上购买然后回来自己养，没想到出乎意料的好，很健康也很容易养大。
356607	354513	2019-09-07 13:07:45	596988278783	超级可爱，一直跑来跑去的 不知道在找什么东西
356608	354514	2019-07-31 10:24:56	596988278783	非常好，物流也非常快，尤其店家态度特别好，很关键挺满意的
356609	354515	2019-10-03 13:06:02	596988278783	非常喜欢

356610 rows × 4 columns

In [125]:

```
data
```

Out[125]:

	Unnamed: 0	爬取时间	爬取链接	商品ID	商品名称	商品现价
0	0	2019-11-12 15:10:23	https://item.taobao.com/item.htm?id=566960035676	566960035676	蛇仔鱼苦力泥鳅观赏鱼热带鱼除蛋白虫涡虫虾缸搭档清洁鱼易养鱼	9.90
1	1	2019-11-12 15:10:22	https://item.taobao.com/item.htm?id=604255887198	604255887198	水母活物水族箱宠物水母活 淡水观赏水母缸活小型水族箱	243.33
2	2	2019-11-12 15:10:21	https://item.taobao.com/item.htm?id=521281145450	521281145450	红绿灯鱼活体群游灯鱼热带观赏鱼小型灯科鱼孔雀鱼活体饲料鱼	54.50
3	3	2019-11-12 15:10:15	https://item.taobao.com/item.htm?id=43436641783	43436641783	水母活体套装迷你音乐玻璃鱼缸赤月情人节礼物海月水母生日礼物	173.00
4	4	2019-11-12 15:10:15	https://item.taobao.com/item.htm?id=585814004251	585814004251	观赏活体海月水母赤月水母生日礼物七夕情人活体宠物倒立水母包邮	78.50
...	...	...	...	...	...	...
23329	23334	2019-11-15 10:43:23	https://item.taobao.com/item.htm?id=530942254401	530942254401	壹品红血鹦鹉增红鱼粮红鹦鹉增色鱼饲料地图招财鱼食一品红鱼食	48.50
23330	23335	2019-11-15 10:43:22	https://item.taobao.com/item.htm?id=37482165781	37482165781	海神 血鹦鹉增红专用粮500g/1000g血鹦鹉鱼食饲料鱼粮（升级版）	105.00

	Unnamed: 0	爬取时间	爬取链接	商品ID	商品名称	商品现价
23331	23336	2019-11-15 10:43:22	https://item.taobao.com/item.htm?id=597889386973	597889386973	益口红血鹦鹉饲料增红增色专用粮观赏鱼食招财鱼鱼粮包邮	30.30
23332	23337	2019-11-15 10:43:21	https://item.taobao.com/item.htm?id=534921855001	534921855001	帝溢红 血鹦鹉增红增色鱼食 高虾红素营养 上浮型5天增红鱼粮饲料	55.50
23333	23338	2019-11-15 10:43:21	https://item.taobao.com/item.htm?id=604810272138	604810272138	虹立方血鹦鹉增红鱼粮财神罗汉饲料热带观赏鱼鱼食 不浑水	60.00

23334 rows × 34 columns

In [126]:

```
comment_form
```

Out[126]:

	Unnamed: 0	时间	url	评论
0	0	2019-12-15 13:45:48	18650288040	鱼收到了，很漂亮很活跃，养了很多天了，开始进食了。
1	1	2019-12-11 19:24:50	18650288040	特意养了一段时间才评价，鱼儿不错，很活波，好养活
2	2	2019-12-07 15:03:09	18650288040	越来越活泼了
3	3	2019-12-10 21:55:26	18650288040	物流快，包装仔细，🐟 活泼漂亮，很喜欢的
4	4	2019-12-04 17:19:47	18650288040	鱼很活跃，特意过了几天再来评论，物流快速，包装精致！
...	...	...	...	...
356605	354511	2019-07-10 13:29:26	596988278783	好胖的一只 刚刚到家就乱跑
356606	354512	2019-09-28 14:23:26	596988278783	自己是开宠物店的，第一次尝试在网上购买然后回来自己养，没想到出乎意料的好，很健康也很容易养大。
356607	354513	2019-09-07 13:07:45	596988278783	超级可爱，一直跑来跑去的 不知道在找什么东西
356608	354514	2019-07-31 10:24:56	596988278783	非常好，物流也非常快，尤其店家态度特别好，很关键挺满意的
356609	354515	2019-10-03 13:06:02	596988278783	非常喜欢

356610 rows × 4 columns

In [127]:

```
#导入库文件
from snownlp import SnowNLP
```

In [128]:

```
def emotion(s):
    positive=0
    negative=0
    smooth=0
    for i in s:
        if i>0.6:
            positive+=1
        elif i<0.4:
            negative+=1
        else:
            smooth+=1
    counts=positive+negative+smooth
    print('积极情绪：',str(round(positive/counts*100,0))+'%')
    print('消极情绪：',str(round(negative/counts*100,0))+'%')
    print('平和情绪：',str(round(smooth/counts*100,0))+'%')
```

In [129]:

```
#uid=comment_form[comment_form['url']=='i']['商品ID']
```

In [130]:

```
def func(ID):
    for i in uid:
        if ID==i:
            return True
    return False
```

In [131]:

```
#比较不同宠物店铺的情感分析结果：
for i in shops:
    uid=data[data['店铺名称']==i]['商品ID']#把data表里涉及到某一店铺的索引号找出来
    whether_true=comment_form['url'].apply(func)
    comments=comment_form[whether_true]#在comment_form表里比对上面的索引号，找到该店铺在评
    comments.index=np.arange(len(comments))
    a=[np.round(SnowNLP(sen).sentiments,2) for sen in comments['评论']]
    print(i,'宠物店铺情感分析结果：')
    emotion(a)
```

```
马来西亚新轩龍鱼繁殖场 宠物店铺情感分析结果：
积极情绪： 90.0%
消极情绪： 6.0%
平和情绪： 4.0%
宜兴渔场 宠物店铺情感分析结果：
积极情绪： 51.0%
消极情绪： 37.0%
平和情绪： 12.0%
意品宠物用品专营店 宠物店铺情感分析结果：
积极情绪： 59.0%
消极情绪： 30.0%
平和情绪： 11.0%
sunsun森森旗舰店 宠物店铺情感分析结果：
积极情绪： 54.0%
消极情绪： 33.0%
平和情绪： 13.0%
yee宠物用品旗舰店 宠物店铺情感分析结果：
积极情绪： 59.0%
消极情绪： 31.0%
平和情绪： 11.0%
上海大渔夫水族 宠物店铺情感分析结果：
积极情绪： 63.0%
消极情绪： 26.0%
平和情绪： 11.0%
爱信水族 宠物店铺情感分析结果：
积极情绪： 45.0%
消极情绪： 46.0%
平和情绪： 9.0%
抱紧我的小鱼干 宠物店铺情感分析结果：
积极情绪： 72.0%
消极情绪： 22.0%
平和情绪： 6.0%
宠冠宠物用品专营店 宠物店铺情感分析结果：
积极情绪： 60.0%
消极情绪： 28.0%
平和情绪： 12.0%
本森水族用品 宠物店铺情感分析结果：
积极情绪： 68.0%
消极情绪： 23.0%
平和情绪： 9.0%
```

In [132]:

```
# 可以使用循环也可以一个个查看，评论量太大，运行起来会花比较长的时间
```

2.2 LDA主题模型分析

In [133]:

```
import jieba
import lda
from collections import Counter #统计列表中每个元素出现的频次
```

In [134]:

```
string=open("stopwords.txt").read()
string
```

Out[134]:

```
'\ufeff\n1\n2\n3\n4\n5\n6\n7\n8\n9\nnan\ncom\n@\n? \n \n、\n。\n“\n”\n
《\n》\n!\n! \n, \n,\n: \n:\n; \n? \n-\n(\n)\n （\n） \n·\n--\n……\n/\n.\n|
\n——\n‘\n’\n□\n【\n】\nA\nB\nC\nD\n啊\n阿\n哎\n哎呀\n哎哟\n唉\n俺\n俺们\n
按\n按照\n吧\n吧哒\n把\n罢了\n被\n本\n本着\n比\n比方\n比如\n鄙人\n彼\n彼此\n边
\n别\n别的\n别说\n并\n并且\n不比\n不成\n不单\n不但\n不独\n不管\n不光\n不过\n不
仅\n不拘\n不论\n不怕\n不然\n不如\n不特\n不惟\n不问\n不只\n朝\n朝着\n趁\n趁着\n
乘\n冲\n除\n除此之外\n除非\n除了\n此\n此间\n此外\n从\n从而\n出\n打\n待\n但\n但
是\n当\n当着\n到\n得\n的\n的话\n等\n等等\n地\n第\n叮咚\n对\n对于\n多\n多少\n而
\n而况\n而且\n而是\n而外\n而言\n而已\n尔后\n反过来\n反过来说\n反之\n非但\n非徒
\n否则\n嘎\n嘎登\n刚\n刚刚\n该\n赶\n个\n各\n各个\n各位\n各种\n各自\n给\n根据\n
跟\n故\n故此\n固然\n关于\n管\n归\n果然\n果真\n过\n哈\n哈哈\n呵\n和\n何\n何处\n
何况\n何时\n嘿\n哼\n哼唷\n呼哧\n乎\n哗\n还是\n还有\n换句话说\n换言之\n或\n或是
\n或者\n极了\n及\n及其\n及至\n即\n即便\n即或\n即令\n即若\n即使\n几\n几时\n己\n
既\n既然\n既是\n继而\n加之\n假如\n假若\n假使\n鉴于\n将\n较\n较之\n叫\n接着\n结
果\n借\n紧接着\n进而\n尽\n尽管\n经\n经过\n就\n就是\n就是说\n据\n具体地说\n具体
说来\n开始\n开外\n靠\n咳\n可\n可见\n可是\n可以\n况且\n啦\n来\n来着\n离\n例如\n
哩\n连\n连同\n两者\n了\n临\n另\n另外\n另一方面\n论\n嘛\n吗\n慢说\n漫说\n冒\n么
\n每\n每当\n们\n莫若\n某\n某个\n某些\n拿\n哪\n哪边\n哪儿\n哪个\n哪里\n哪年\n哪
```

In [135]:

```
#上面有很多\n,先把它拆分
filterwords=string.split("\n")
filterwords
```

Out[135]:

```
['\ufeff',
 '1',
 '2',
 '3',
 '4',
 '5',
 '6',
 '7',
 '8',
 '9',
 'nan',
 'com',
 '@',
 '? ',
 ' ',
 '、',
 '。',
 '“',
```

In [136]:

```
#关闭警告和loging
import logging
import warnings
logging.disable(logging.WARNING)
warnings.filterwarnings('ignore')
```

In [137]:

```
for i in shops:
    uid=data[data['店铺名称']==i]['商品ID']#把data表里涉及到某一店铺的索引号找出来
    whether_true=comment_form['url'].apply(func)
    comments=comment_form[whether_true]#在comment_form表里比对上面的索引号，找到该店铺在评
    comments.index=np.arange(len(comments))
    #将三种情绪的评价内容分别保存在三个不同的评价框里
    neg=[]
    pos=[]
    mid=[]
    for sen in comments["评论"]:
        s=SnowNLP(sen).sentiments
        if s<0.4:
            neg.append(sen)
        elif s>0.6:
            pos.append(sen)
        else:
            mid.append(sen)
    #函数1，对某种情绪的数据列进行分词及过滤
    def word_cut(coms):
        b=[]
        for i in jieba.cut(coms):
            if i not in filterwords:
                b.append(i)
        return b
    #函数2，将高频词进行数值化：对每个高频词如果出现在某行分词评论列，则该行值记录为1，否则为0。得
    def get_vector(sentence,vocab):
        temp=[]
        for word in vocab:
            if word in sentence:
                temp.append(1)
            else:
                temp.append(0)
        return temp
    #对不同情绪做LDA主题模型分析的函数
    def get_lda(params):
        corpora_words=[]
        for i in params:
            ss=word_cut(i) #调用函数1
            corpora_words.append(ss)
        words=[]
        for i in corpora_words:
            words+=i
        word_count=Counter(words)
        vocab=[]
        for word in word_count.keys():
            if word_count[word]>1:
                vocab.append(word)
        X=[]
        for se in corpora_words:
            X.append(get_vector(se,vocab)) #调用函数2
        X=np.array(X)
        lda_model=lda.LDA(n_topics=10,n_iter=100,random_state=1)
        lda_model.fit(X)
        topic_word=lda_model.topic_word_
        for i in range(10):
            index=np.argsort(topic_word[i])[::-1]
            print('主题',i,':',end='')
            for j in np.array(vocab)[index][0:10]:
                print(j,end=' ')
```

```
            print()
    print(i+":")
    print("积极情绪:")
    print(get_lda(pos))
    print("消极情绪:")
    print(get_lda(neg))
    print("平和情绪:")
    print(get_lda(mid))
    print("==========================================")
```

```
马来西亚新轩龍鱼繁殖场:
积极情绪:
主题 0 :不错 好 耐心 客服 卖家 快 发货 小鱼 状态 精神状态
主题 1 :好 包装 活跃 漂亮 鱼儿 收到 回来 状态 发货 店家
主题 2 :好 价格 满意 好评 实惠 老板 服务 卖家 服务态度 购买
主题 3 :活泼 喜欢 鱼儿 漂亮 还会 下次 龙鱼 可爱 一如既往 孩子
主题 4 :鱼 收到 挺 很漂亮 满意 吃 活泼 不错 活跃 物流
主题 5 :很快 鱼 满意 物流 活蹦乱跳 活跃 描述 正品 活泼 游来游去
主题 6 :喜欢 好看 鱼 特别 鱼儿 收到 颜色 金龙鱼 担心 精神
主题 7 :好 养 鱼儿 活泼 几天 健康 龙鱼 一段时间 吃食 评价
主题 8 :颜色 好看 满意 收到 活泼 挺 宝贝 鱼儿 观察 长大
主题 9 :喜欢 很漂亮 精神 买 购买 值得 颜色 漂亮 真的 好评
None
消极情绪:
主题 0 :发货 收到 鱼 没 用户 担心 鱼儿 一切正常 评价 方未
主题 1 :一条 活蹦乱跳 收到 没 用户 担心 鱼儿 一切正常 评价 方未
主题 2 :东西 没想到 网上 第一次 没 用户 担心 鱼儿 一切正常 评价
主题 3 :鱼 好评 没 填写 担心 鱼儿 一切正常 评价 方未 做出
主题 4 :鱼缸 没 填写 担心 鱼儿 一切正常 评价 方未 做出 系统
```

In [ ]:

所有评分水平都是“低于”且平均评分在4.6以下的店铺情感得分和LDA主题分析

In [142]:

```
df1= data[(data["描述评分水平"]=="低于")&(data["服务态度水平"]=="低于")&(data["物流服务水平
df1
```

Out[142]:

	Unnamed: 0	爬取时间	爬取链接	商品ID	商品名称	商品现价	商品原价	月销量	总销量	发货地址
5	5	2019-11-12 15:10:14	https://item.taobao.com/item.htm?id=604505560036	604505560036	水母活非淡水活物宠物大小型夜光便宜无毒发	51.500	51.500	1	0	山西

In [143]:

```
len(df1)
```

Out[143]:

```
1120
```

In [144]:

```
uid=df1['商品ID']#把df1表里商品ID后存在uid里
whether_true=comment_form['url'].apply(func)
comments_lower=comment_form[whether_true]#在comment_form表里比对上面的索引号，找到所有“低
comments_lower.index=np.arange(len(comments_lower))
a=[np.round(SnowNLP(sen).sentiments,2) for sen in comments_lower['评论']]
print('所有评价为”低于“且平均评分小于4.6分的宠物店铺情感分析结果：')
emotion(a)
```

```
所有评价为”低于“且平均评分小于4.6分的宠物店铺情感分析结果：
积极情绪： 54.0%
消极情绪： 37.0%
平和情绪： 9.0%
```

In [145]:

```
#将三种情绪的评价内容分别保存在三个不同的评价框里
neg=[]
pos=[]
mid=[]
for sen in comments_lower["评论"]:
    s=SnowNLP(sen).sentiments
    if s<0.4:
        neg.append(sen)
    elif s>0.6:
        pos.append(sen)
    else:
        mid.append(sen)
#函数1，对某种情绪的数据列进行分词及过滤
def word_cut(coms):
    b=[]
    for i in jieba.cut(coms):
        if i not in filterwords:
            b.append(i)
    return b
#函数2，将高频词进行数值化：对每个高频词如果出现在某行分词评论列，则该行值记录为1，否则为0。得到n个
def get_vector(sentence,vocab):
    temp=[]
    for word in vocab:
        if word in sentence:
            temp.append(1)
        else:
            temp.append(0)
    return temp
#对不同情绪做LDA主题模型分析的函数
def get_lda(params):
    corpora_words=[]
    for i in params:
        ss=word_cut(i) #调用函数1
        corpora_words.append(ss)
    words=[]
    for i in corpora_words:
        words+=i
    word_count=Counter(words)
    vocab=[]
    for word in word_count.keys():
        if word_count[word]>1:
            vocab.append(word)
    X=[]
    for se in corpora_words:
        X.append(get_vector(se,vocab)) #调用函数2
    X=np.array(X)
    lda_model=lda.LDA(n_topics=10,n_iter=100,random_state=1)
    lda_model.fit(X)
    topic_word=lda_model.topic_word_
    for i in range(10):
        index=np.argsort(topic_word[i])[::-1]
        print('主题',i,':',end='')
        for j in np.array(vocab)[index][0:10]:
            print(j,end=' ')
        print()
print("积极情绪:")
print(get_lda(pos))
print("消极情绪:")
print(get_lda(neg))
```

```
print("平和情绪:")
print(get_lda(mid))
```

```
积极情绪:
主题 0 :可爱 活泼 仓鼠 好 买 一只 两只 布丁 养 挺
主题 1 :小鱼 好 鱼 死 一条 不错 买 挺 很漂亮 漂亮
主题 2 :好 快 物流 发货 很快 ; & 好评 包装 高
主题 3 :不错 活泼 好 鱼 小鱼 收到 很漂亮 喜欢 满意 鱼儿
主题 4 :好 包装 卖家 客服 鱼 不错 满意 很快 店家 好评
主题 5 :好 挺 收到 活泼 小鱼 可爱 活 活蹦乱跳 养 没
主题 6 :可爱 喜欢 仓鼠 活泼 好 超级 特别 健康 挺 孩子
主题 7 :好 不错 挺 活着 活 螺 状态 死 鱼 活泼
主题 8 :好 购买 不错 值得 卖家 第二次 服务 买 下次 还会
主题 9 :买 好 送 螺 不错 一个 满意 活 店家 鱼
None
消极情绪:
主题 0 :好评 评价 螺 虾 默认 做出 系统 方未 活着 吃
主题 1 :评论 用户 填写 买家 盒子 15 仓鼠 臭 打开 破
主题 2 :买 鱼 死 虾 太小 好 没 鱼苗 条 很小
主题 3 :好 客服 卖家 态度 商家 退款 店家 收到 快递 好评
主题 4 :鱼 死 一条 收到 没 好 买 两条 剩 两天
主题 5 :死 一只 好 好评 买 客服 两只 可爱 没 挺
主题 6 :死 买 一个 差评 螺 东西 两个 没 一只 活
主题 7 :死 一只 ; & hellip 虾 不到 找 放 卖家
主题 8 :鱼 好 没 死 虾 吃 挺 活 一个 买
主题 9 :死 鱼 好 一条 买 收到 退款 老板 商家 卖家
None
平和情绪:
主题 0 :好 鱼 卖家 收到 满意 不错 一条 挂 鱼儿 店家
主题 1 :买 好评 全活 好 两只 一个 这次 点 晚上 第二次
主题 2 :可爱 收到 好 一只 挺 健康 活泼 鼠鼠 特别 咬
主题 3 :螺 吃 太小 缸 活 还好 真的 养 苹果 😂
主题 4 :死 好 鱼 一条 挺 不错 买 很快 老板 下次
主题 5 :好评 好 活着 颜色 他家 死 图片 状态 到家 几个
主题 6 :好 客服 卖家 仓鼠 喜欢 态度 活 服务态度 活蹦乱跳 可爱
主题 7 :购买 活 好 养 没 一个 死 几天 快 活蹦乱跳
主题 8 :死 不错 好 包装 买 送 没 鱼 虾 一只
主题 9 :活着 鱼 收到 好看 好评 活跃 谢谢 还会 小虾 感觉
None
```

In [ ]:

```
```

布偶猫情感得分和LDA主题分析

In [146]:

```
data.columns
```

Out[146]:

```
Index(['Unnamed: 0', '爬取时间', '爬取链接', '商品ID', '商品名称', '商品现
价', '商品原价', '月销量',
       '总销量', '发货地址', '商品发布时间', '商品规格', '商品库存', '店铺名
称', '店铺url', '商品参数',
       '商品sku详情', '商品链接', '商品详情', '店铺评分', '宝贝收藏数', '一级
分类', '二级分类', '三级分类',
       '现价等级', '描述相符打分', '描述评分水平', '服务态度打分', '服务态度水
平', '物流服务打分', '物流服务水平',
       '平均评分', '销售额数据', '商品折扣数据'],
      dtype='object')
```

In [148]:

```
for i in set(data["三级分类"]):
    print(i)
```

```
孔雀鱼
潜水泵
泰迪
魔法鱼
斗鱼
比熊
水质调理
饲料/零食
猫主粮
热带鱼
杜鹃根
风扇
绳结
猫零食
金鱼
虾螺
陶罐
发声玩具
锦鲤
沉木
水草泥
猫抓板
龙猫
过滤器
乌龟饲料
假山
罗汉鱼
橡胶球
测试纸
加热棒
水草套餐
飞盘
水妖精
木化石
金吉拉
狗主粮
液肥
水生蕨类
豚鼠
宠物貂
造景石料
薄片鱼粮
波斯猫
前景草
套装玩具
铁皇冠
宠物狐狸
氧气泵
仓鼠
猫爬架
漏食球
虾粮
鱼缸
龙鱼
香波/浴液
博美
```

绿藻球
增红鱼粮
水质检测
仿真水草
窝/帐篷
颗粒鱼粮
除藻剂
饵干
除氯杀菌
水草灯
底栖鱼
小水榕
逗猫棒
灯科鱼
水母
造流泵
睡莲类
布偶猫
吉娃娃
硝化细菌
比格犬
香猪
飞鼠
服装
矮珍珠
英国短毛猫
折耳猫
鱼饲料
基底肥
加菲猫
丝瓜络玩具
开口鱼粮
玩具

In [151]:

```
df2=data[data["三级分类"]=="布偶猫"]
df2
```

Out[151]:

	Unnamed: 0	爬取时间	爬取链接	商品ID	商品名称	商品现价	商品原价	月销量	总销量	发货地址	.

In [152]:

```
len(df2)
```

Out[152]:

```
295
```

In [154]:

```
uid=df2['商品ID']#把df1表里商品ID后存在uid里
whether_true=comment_form['url'].apply(func)
comments=comment_form[whether_true]#在comment_form表里比对上面的索引号，找到所有"布偶猫"的
comments.index=np.arange(len(comments))
a=[np.round(SnowNLP(sen).sentiments,2) for sen in comments['评论']]
print('布偶猫铺情感分析结果：')
emotion(a)
```

```
布偶猫铺情感分析结果：
积极情绪：  96.0%
消极情绪：  2.0%
平和情绪：  2.0%
```

In [155]:

```
#将三种情绪的评价内容分别保存在三个不同的评价框里
neg=[]
pos=[]
mid=[]
for sen in comments["评论"]:
    s=SnowNLP(sen).sentiments
    if s<0.4:
        neg.append(sen)
    elif s>0.6:
        pos.append(sen)
    else:
        mid.append(sen)
#函数1，对某种情绪的数据列进行分词及过滤
def word_cut(coms):
    b=[]
    for i in jieba.cut(coms):
        if i not in filterwords:
            b.append(i)
    return b
#函数2，将高频词进行数值化：对每个高频词如果出现在某行分词评论列，则该行值记录为1，否则为0。得到n个
def get_vector(sentence,vocab):
    temp=[]
    for word in vocab:
        if word in sentence:
            temp.append(1)
        else:
            temp.append(0)
    return temp
#对不同情绪做LDA主题模型分析的函数
def get_lda(params):
    corpora_words=[]
    for i in params:
        ss=word_cut(i) #调用函数1
        corpora_words.append(ss)
    words=[]
    for i in corpora_words:
        words+=i
    word_count=Counter(words)
    vocab=[]
    for word in word_count.keys():
        if word_count[word]>1:
            vocab.append(word)
    X=[]
    for se in corpora_words:
        X.append(get_vector(se,vocab)) #调用函数2
    X=np.array(X)
    lda_model=lda.LDA(n_topics=10,n_iter=100,random_state=1)
    lda_model.fit(X)
    topic_word=lda_model.topic_word_
    for i in range(10):
        index=np.argsort(topic_word[i])[::-1]
        print('主题',i,':',end='')
        for j in np.array(vocab)[index][0:10]:
            print(j,end=' ')
        print()
print("积极情绪:")
print(get_lda(pos))
print("消极情绪:")
print(get_lda(neg))
```

```
print("平和情绪:")
print(get_lda(mid))
```

```
积极情绪:
主题 0 :可爱 喜欢 小猫咪 好 特别 好评 活泼 健康 猫 毛茸茸
主题 1 :物流 好 很快 收到 满意 快 服务态度 卖家 服务 挺
主题 2 :可爱 喜欢 好 小猫 特别 好看 猫猫 听话 摸 舒服
主题 3 :可爱 满意 喜欢 收到 猫猫 宝贝 实惠 价格 不错 便宜
主题 4 :可爱 喜欢 好 买 猫咪 真的 网上 猫 几天 儿子
主题 5 :喜欢 可爱 购买 收到 值得 满意 好 宝贝 推荐 价格
主题 6 :可爱 喜欢 猫咪 买 宠物 收到 第一次 超级 满意 网上
主题 7 :可爱 喜欢 猫咪 萌萌 小猫咪 收到 好 哒 特别 超
主题 8 :喜欢 好 下次 可爱 不错 买 收到 好评 猫咪 老板
主题 9 :可爱 喜欢 猫咪 好 健康 小猫咪 收到 毛绒绒 女儿 眼睛
None
消极情绪:
主题 0 :好 吃 商家 喜欢 收到 特别 哒 卖家 好评 用户
主题 1 :喜欢 吃 好 商家 收到 特别 哒 卖家 好评 用户
主题 2 :一模一样 哒 卖家 接到 吃 好评 商家 喜欢 收到 特别
主题 3 :简直 收到 特别 喜欢 吃 好 商家 哒 卖家 好评
主题 4 :好评 喜欢 吃 好 商家 收到 特别 哒 卖家 用户
主题 5 :担心 小猫 吃 好评 商家 喜欢 收到 特别 哒 卖家
主题 6 :评论 家 吃 好评 商家 喜欢 收到 特别 哒 卖家
主题 7 :吃 好 商家 喜欢 收到 特别 哒 卖家 好评 用户
主题 8 :商家 可爱 收到 好 喜欢 特别 哒 卖家 好评 吃
主题 9 :用户 填写 评论 好评 商家 喜欢 收到 特别 哒 卖家
None
平和情绪:
主题 0 :收到 超级 质量 满意 客服 可爱 好 好评 喜欢 货
主题 1 :好 质量 超级 满意 客服 可爱 好评 喜欢 收到 货
主题 2 :超级 可爱 质量 满意 客服 好 好评 喜欢 收到 货
主题 3 :喜欢 好评 超级 质量 满意 客服 可爱 好 收到 货
主题 4 :超级 质量 满意 客服 可爱 好 好评 喜欢 收到 货
主题 5 :好 超级 质量 满意 客服 可爱 好评 喜欢 收到 货
主题 6 :好评 收到 超级 质量 满意 客服 可爱 好 喜欢 货
主题 7 :客服 超级 质量 满意 可爱 好 好评 喜欢 收到 货
主题 8 :可爱 超级 质量 满意 客服 好 好评 喜欢 收到 货
主题 9 :货 满意 收到 好评 超级 质量 客服 可爱 好 喜欢
None
```