实验 3-4 定义红酒的价格区间

建议课时：20 分钟

一、 实验目的

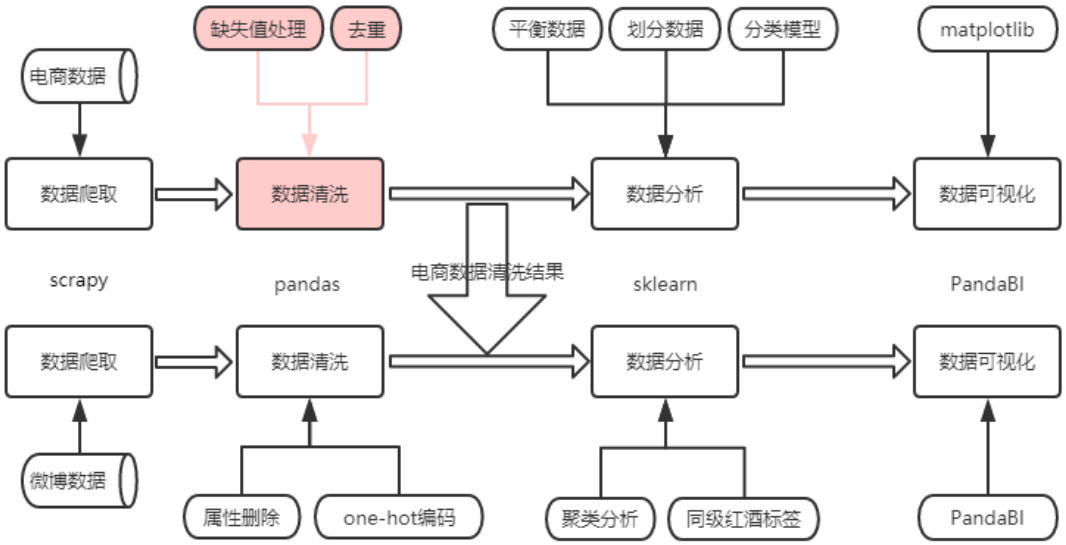
- 了解 pandas 的常用函数
- 了解数据清洗的基础方法

二、 实验环境

Python3 开发环境，第三方包有 pandas

三、 实验步骤

本节处理的内容有：



首先查看一下红酒的价格最小值和最大值

```
print(df["price"].min())
print(df["price"].max())
```

```
3.32
79560.0
```

查看数值分布数量的常见方式有三种：等宽分箱、等频分箱和基于聚类的分箱

代码如下：

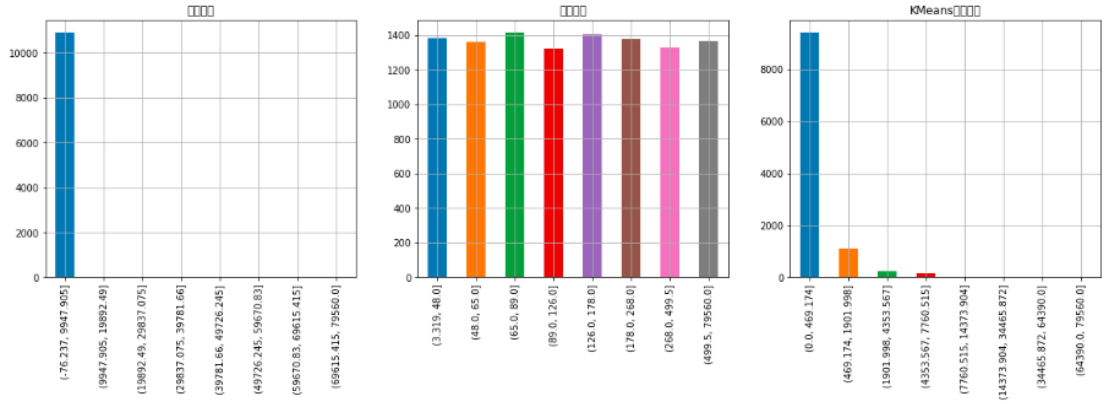
```
# 三种方式对价格的划分
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
%matplotlib inline
plt.rcParams['font.family'] = ['sans-serif']
plt.rcParams['font.sans-serif'] = ['SimHei']

k = 8
kmodel = KMeans(n_clusters = k,n_jobs=5)

fig,ax= plt.subplots(1,3,figsize=(20,5))
cat = pd.cut(df['price'],k)
cat2 = pd.qcut(df['price'],k)
kmodel.fit(df['price'].values.reshape(len(df),1))
c = pd.DataFrame(kmodel.cluster_centers_).sort_values(0)
w = c.rolling(2).mean().iloc[1:]
w = [0] + list(w[0]) + [df['price'].max()]
cat3 = pd.cut(df['price'], w)

cat.value_counts(sort = False).plot.bar(grid= True,ax=ax[0],title = '等宽分箱')
cat2.value_counts(sort = False).plot.bar(grid= True,ax=ax[1],title = '等频分箱')
cat3.value_counts(sort = False).plot.bar(grid= True,ax=ax[2],title = 'KMeans聚类分箱')
```

结果如下：



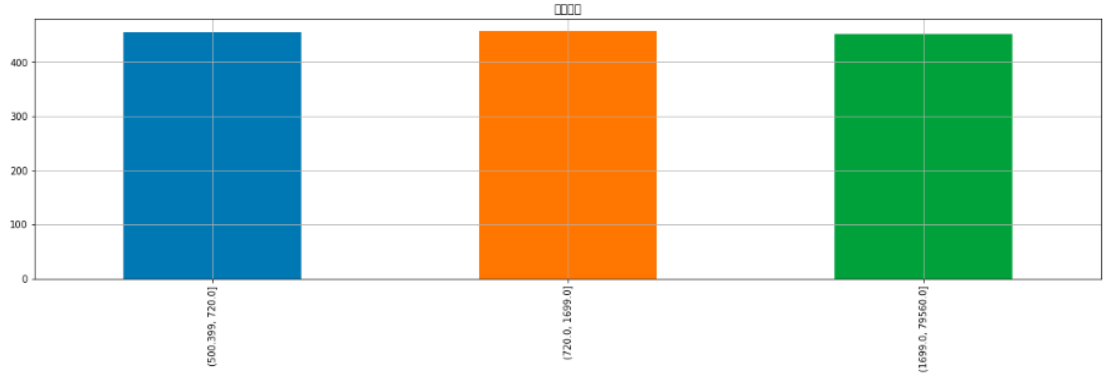
根据上述三种价格区间的划分，选择“等频分箱”方法，由于最后一组价格区间（499,79560）跨度过大，对价格在 500 以上和 500 以下的重新分箱。

本案例中自定义分成 8 个价格区间（观察下来很多属性有 6-8 种取值），因数据主要集中分布在 500 之前的数据，故设计前 500 包含 5 个区间，大于 500 的再设立 3 个区间。（注意：价格区间的个数比较自由化，可根据真实项目中的测试结果进行调整）

```
df3 = df[df["price"]>500]
fig,ax= plt.subplots(1,1,figsize=(20,5))
cat4 = pd.qcut(df3['price'],3)
cat4.value_counts(sort = False).plot.bar(grid= True,title = '等频分箱')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f18af717cc0>
```

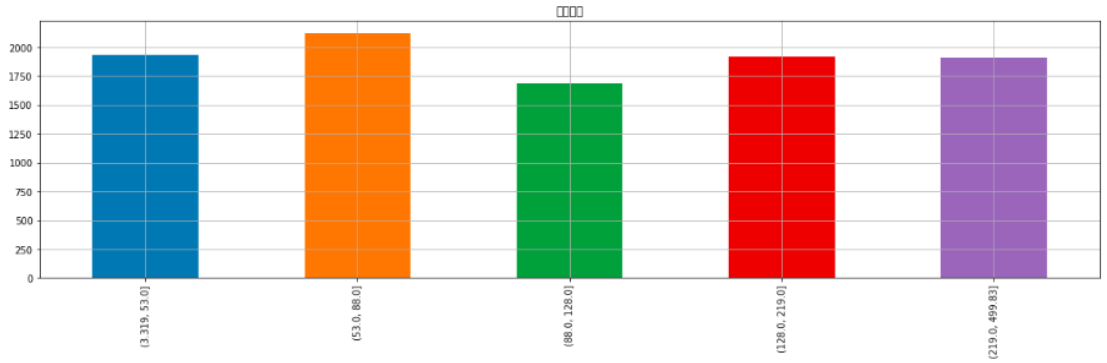
```
/usr/local/python3/lib/python3.6/site-packages/matplotlib/font_manager.py:1328: UserWarning: findfont: Font family ['sans-ser
if'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```




```
df4 = df[df["price"]<=500]
fig,ax= plt.subplots(1,1,figsize=(20,5))
cat4 = pd.qcut(df4['price'],5)
cat4.value_counts(sort = False).plot.bar(grid= True,title = '等频分箱')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f18af6dea90>
```

```
/usr/local/python3/lib/python3.6/site-packages/matplotlib/font_manager.py:1328: UserWarning: findfont: Font family ['sans-ser
if'] not found. Falling back to DejaVu Sans
  (prop.get_family(), self.defaultFamily[fontext]))
```



结合上述统计结果：将价格定位成以下 8 个区间：[0,50],[50-100],[100-150],[150-250],[250-500],[500-1000],[1000-2000],[2000-MAX]

处理代码如下：

```
# 处理价格区间
# 价格区间划分为：[0,50],[50,100],[100,150],[150,250],[250,500],[500,1000],[1000,2000],[2000,max]
import sys
def get_price_scope(price):        #获取价格区间
    scope = [[0,50],[50,100],[100,150],[150,250],[250,500],[500,1000],[1000,2000],[2000,sys.maxsize]]
    for j in range(len(scope)):
        if price >= scope[j][0] and price < scope[j][1]:
            result = '-'.join(str(x) for x in scope[j])
    return result
df['price'] = df['price'].map(get_price_scope)
```

本节的清洗数据综上已收尾，将处理结果保存为 wine_processed.csv 供后续模型训练用。