Python 数据分析报告

目录

一 小组信息.....2
二 前言.....2
三 Python 数据可视化初步.....3
3.0 概述.....3
3.1 数据可视化的概念.....3
3.2 数据可视化与 csv 和 json 文件的结合.....5
3.3 数据可视化与 API 的结合.....12
3.4 数据可视化与降维方法 PCA 的结合.....13
3.5 数据可视化与机器学习算法 KNN 的结合.....15
四 葡萄酒品质—从实际案例总体来看数据可视化.....18
4.1 问题描述.....19
4.2 思路分析.....20
4.3 数据分析.....22
五 总结.....63
六 参考资料.....64

选题说明：在一开始的大作业要求中，如果不做抖音课题，Python 编程分析只要与 Python 和所学知识有关即可，报告课题是可以任选的。在与大作业检查负责人赵源助教确认并获得许可后，我们选择了 Python 可视化的课题。

一 小组信息

小组人数：3 人

人员信息：

	职位	学号	邮箱	完成数量	分工
李振羽	组长	181250079	181250079@smail.nju.edu.cn	199	论文第三部分
车一晗	组员	181250009	181250009@smail.nju.edu.cn	195	论文第四部分
纳思彧	组员	181250107	181250107@smail.nju.edu.cn	193	论文第四部分

二 前言

在数据科学越来越重要的当今社会，对庞大的数据进行分析，处理，找出数据中存在的客观规律，建立描绘群体行为的数学模型，已成为促进社会进步的重要手段。

而 Python 是目前市面上用于大数据分析的优先选择，Python 数据功能强大，对数据抽取，收集整理，分析挖掘都可以实现，避免了开发程序的切换。Python 的数据挖掘能力和产品构建能力兼而有之，是跨平台且开源的技术，成本较小。

数据时代，通过数据分析挖掘数据的价值，Python 就是很好的选择，它包含了 Numpy、Pandas、Matplotlib、Scipy、iPython 等主要数据分析库。

数据可视化是数据探索的主要途径。数据可视化的目标是通过所选方法的视觉展示，清晰有效地与用户交流信息。有效的可视化有助于分析和推理数据和证据。这使得复杂数据更容易接触，理解和使用。

本论文讨论了 Python 在数据分析中如何利用自身的语言优势进行数据可视化并与概率统计知识相结合，从而直观地分析数据集，数据集的客观规律以及特征。

三 Python 数据可视化初步

3.0 概述

在论文第二部分，主要初步介绍 Python 是如何进行数据可视化的。

3.1 介绍了数据可视化的概念，即将事件或数据集以图表的形式进行反映，让观看者能够看明白其含义，发现数据集中原本未意识到的规律和意义。

3.2 介绍了 Python 如何将常见的数据格式 CSV 和 JSON 进行可视化。

3.3 介绍了 Python 怎样与 API 结合来可视化。

3.4 介绍了 Python 怎样与常见降维方法 PCA 进行结合

3.5 介绍了 Python 怎样与机器学习 KNN 算法结合并评估 KNN 模型的预测能力

3.1 数据可视化的概念

我们通过运用 Python 描述日常生活中的现象来解释数据可视化的概念。

案例 1 利用 Matplotlib 库

本案例来进行绘制散点图模拟水分子无规则运动。如图 1，水分子从原点出发，进行 5000 次任意方向的运动，图上的 5000 个点代表每次运动后的位置，使水分子无规则运动的路径通过散点图进行呈现，按时间顺序，颜色越深的越晚发生。

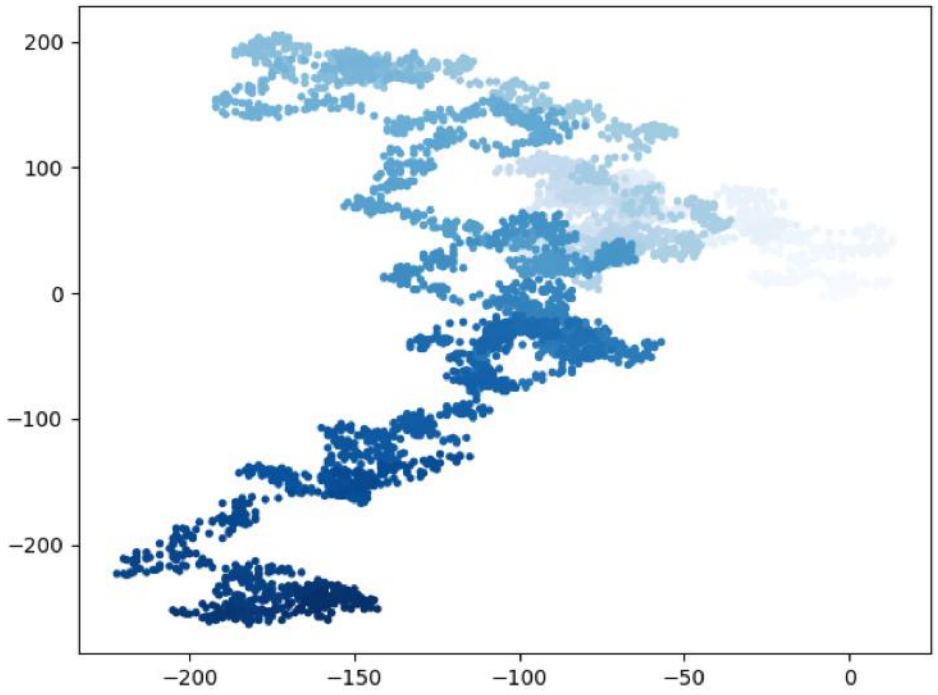


图 1 水分子进行 5000 次运动的位置图

从图 1 可以看出，比起大量数据，通过这些数据生成的散点图可以很直观地模拟出水分子的一个大致运动范围，而毫无规律可循的位置图可以帮助理解水分子运动的“随机”的概念。

案例 2 利用 Pygal 库

本案例中统计扔 1000 次骰子后各点数出现的次数，将其绘制成条形图，如图 2 所示。

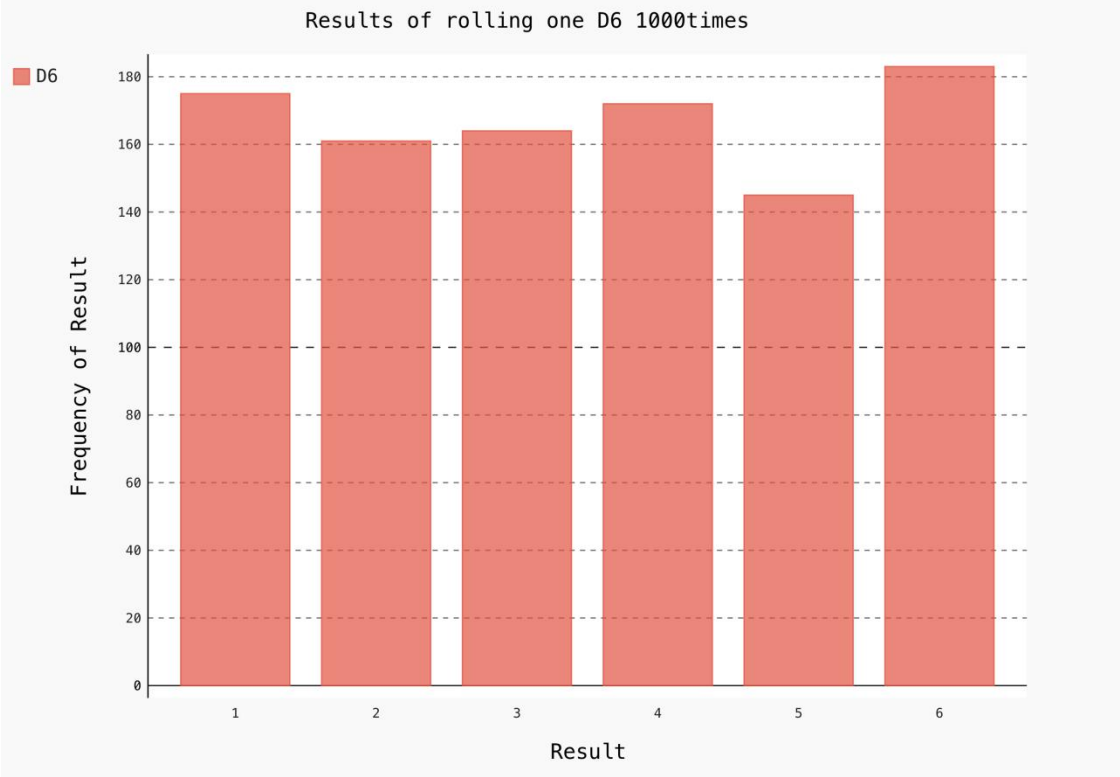


图 2 扔 1000 次骰子后各点数出现的次数

条形图可以清楚地对各个次数进行比较，以看出差别。这里取的样本还是偏小，不足以描绘出接近正确的结果。

3.2 数据可视化与 csv 和 json 文件的结合

csv 文件，即 Comma-Separated Values，逗号分隔值。csv 文件是以纯文本的形式来存储表格数据，有时候是数字，有时候是文本。如果只是单纯地人工去分析处理 csv 文件，是非常困难的，而 Python 有着非常优秀处理 csv 文件的能力，只需将文件导入，库中提供的方法会帮助完成分析数据的任务。

json 文件是一个序列化的对象或是数组，通常应用于前端的数据解析。Python 也可以用来解析它们，进行可视化的工作。

下面用两个案例来详细说明 Python 是如何将上述两种文件进行可视化的。

案例 3 处理 csv 文件

从天气网站上获得伦敦与多伦多两个地区天气的历史数据并把它们存储为 csv 格式文件，利用 Python 获取两个地区每日的最高气温和最低气温，并将气温数据转化为图表进行对比。csv 文件如图 3 所示。

PST	Max TemperatureF	Mean TemperatureF	Min TemperatureF	Max Dew PointF	MeanDew PointF	Min DewpointF
2014/1/1	63	42	24	14	9	5
2014/1/2	66	47	28	22	19	16
2014/1/3	64	46	28	24	21	15
2014/1/4	66	47	28	25	21	16
2014/1/5	61	44	26	18	7	-1
2014/1/6	57	40	23	7	3	-1
2014/1/7	57	44	30	10	7	5
2014/1/8	60	40	21	11	8	3
2014/1/9	57	42	27	19	14	8
2014/1/10	63	44	26	25	20	14
2014/1/11	64	47	30	29	25	17
2014/1/12	63	50	35	29	22	17
2014/1/13	64	46	28	16	14	11
2014/1/14	70	50	30	15	10	7
2014/1/15	70	50	30	12	8	4
2014/1/16	72	50	27	10	3	-1
2014/1/17	71	50	28	7	3	1
2014/1/18	69	48	28	9	4	-1
2014/1/19	69	48	27	8	5	1
2014/1/20	68	49	30	10	6	3
2014/1/21	62	45	28	11	8	4

图 3 伦敦天气数据（部分）

利用 Python 得到两个地区一年中每日的最高气温和最低气温，然后将两个地区一年中每日的最高气温与最低气温分别绘制成图表，如图 4 和图 5 所示。

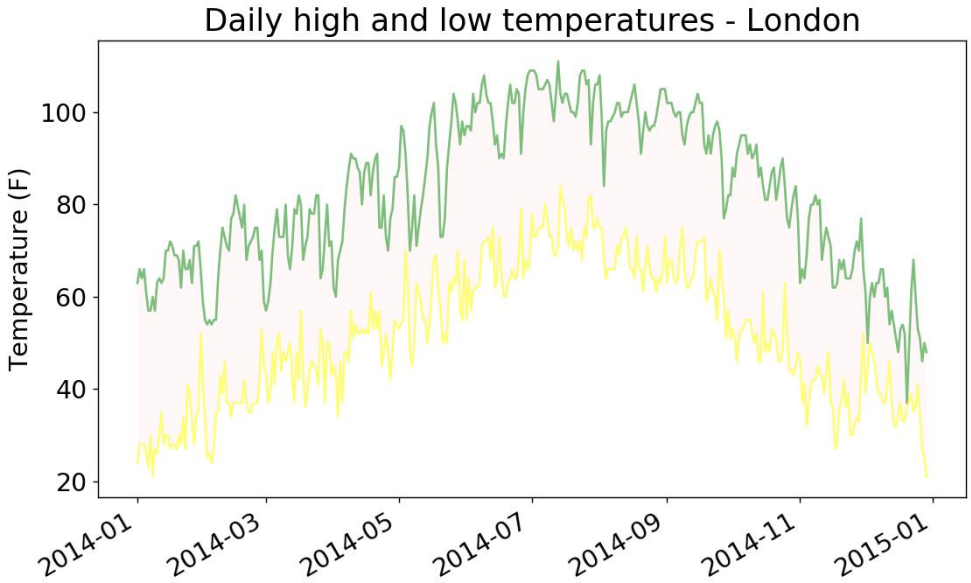


图 4 伦敦一年中每日最高与最低气温图表

同理，我们可以画出多伦多一年中的每日最高温与最低温图表。

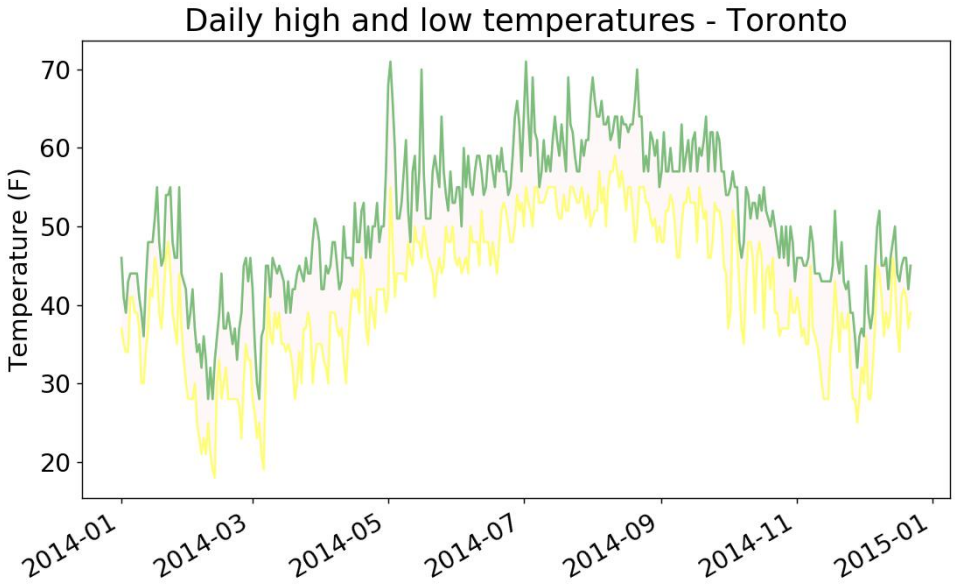


图 5 多伦多一年中每日最高与最低气温图表

案例 4 处理 json 文件

图 6 所示为我们从外部网站获取到 2017 年全年的股票每日收盘价，是以 json 文件的格式存储的。我们将利用 Python 来对这些数据进行分析，进而得出一些结论。


```
[{
    "date" : "2017-01-01",
    "month" : "01",
    "week" : "52",
    "weekday" : "Sunday" ,
    "close" : "6928.6492"
},
{
    "date" : "2017-01-02",
    "month" : "01",
    "week" : "1",
    "weekday" : "Monday",
    "close" : "7070.2554"
},
{
    "date" : "2017-01-03",
    "month" : "01",
    "week" : "1",
    "weekday" : "Tuesday" ,
    "close" : "7175.1082"
}]
```


图 6 2017 年全年股票收盘价（部分）

第一步，我们绘制出股票收盘价的折线图。利用 2.1 中提到的 Pygal 库。

图 7 2017 年全年股票收盘价折线图

从收盘价的折线图可以看出，2017 年的总体趋势是非线性的，而且增长幅度不断增大，似乎呈现出指数分布。我们同时发现，在每个季度末,股票收盘价似乎有一些相似的波动。为了验证波动的周期性，我们使用对数变换消除了非线性的趋势。

图 8 股票收盘价对数变换折线图

用对数变换剔除非线性趋势后，整体上涨的趋势更接近线性增长。从图 8 可以看出，收盘价在 3 月，6 月，9 月，即每个季度末都出现了剧烈的波动。为了进一步探求股票价格变化规律的周期性，我们绘制了收盘价的月日均值，周日均值，以及星期均值，分别为图 9，图 10 和图 11。

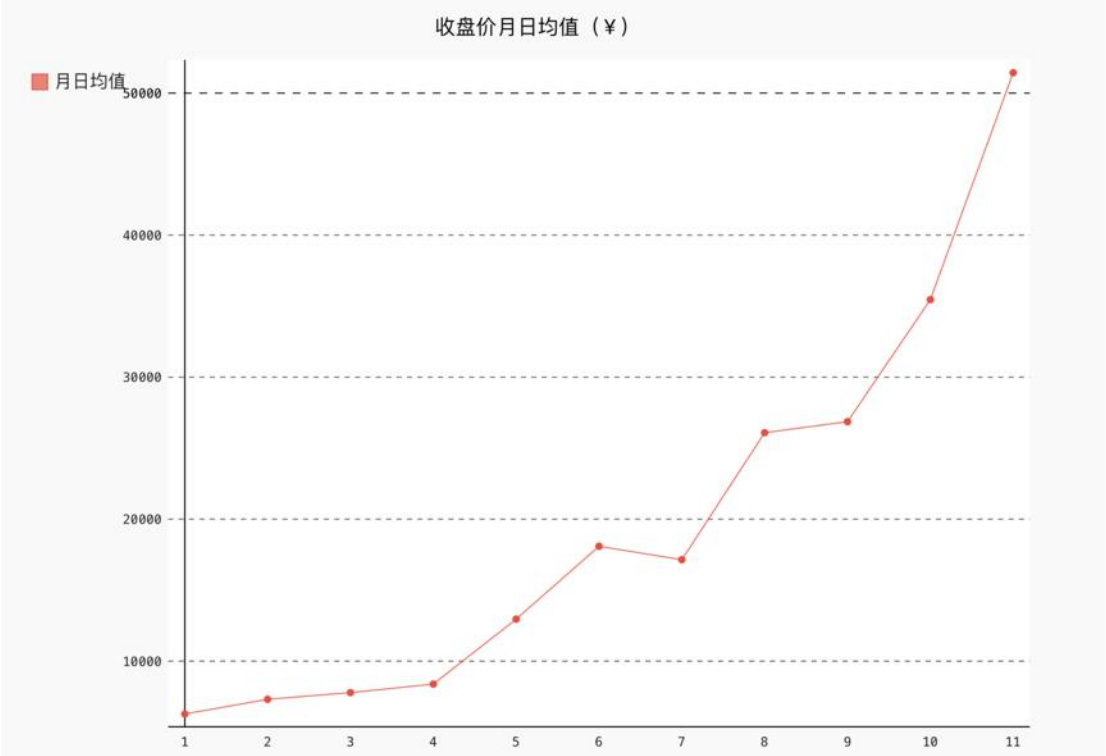


图 9 股票收盘价月日均值折线图

图 10 股票收盘价周日均值折线图

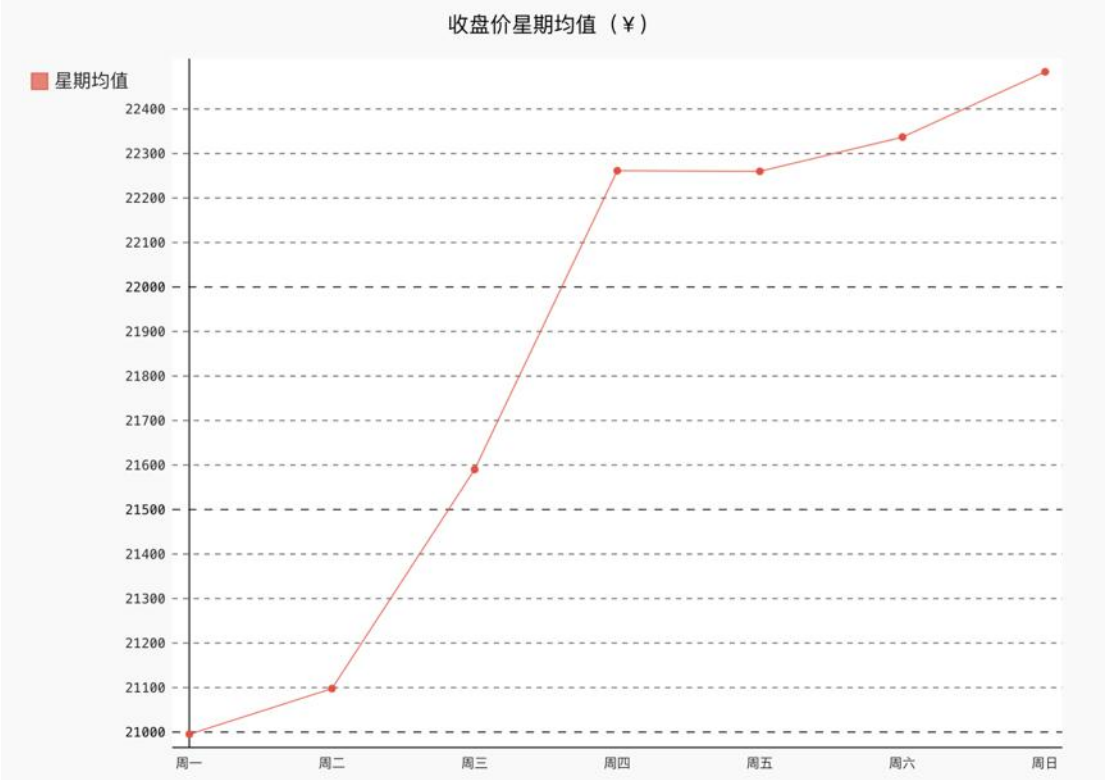


图 11 股票收盘价星期均值折线图

3.3 数据可视化与 API 的结合

Web API 是网站的一部分，用于与使用非常具体的 URL 请求特定信息的程序交互。这种请求称为 API 调用。我们通过 Python 的 reqeust 包执行 API 调用，来获取网站上的数据，进而开始可视化处理。

案例 5 得到 Github 上 star 数最多的 Python 项目

分析方法：我们将从 Github 提供的 API 中获取到的数据存储在字典中，分析字典中的信息，将所有的 Python 项目按星从高到低排序，创建一个交互式条形图。处理后所得到的如图 12 所示。

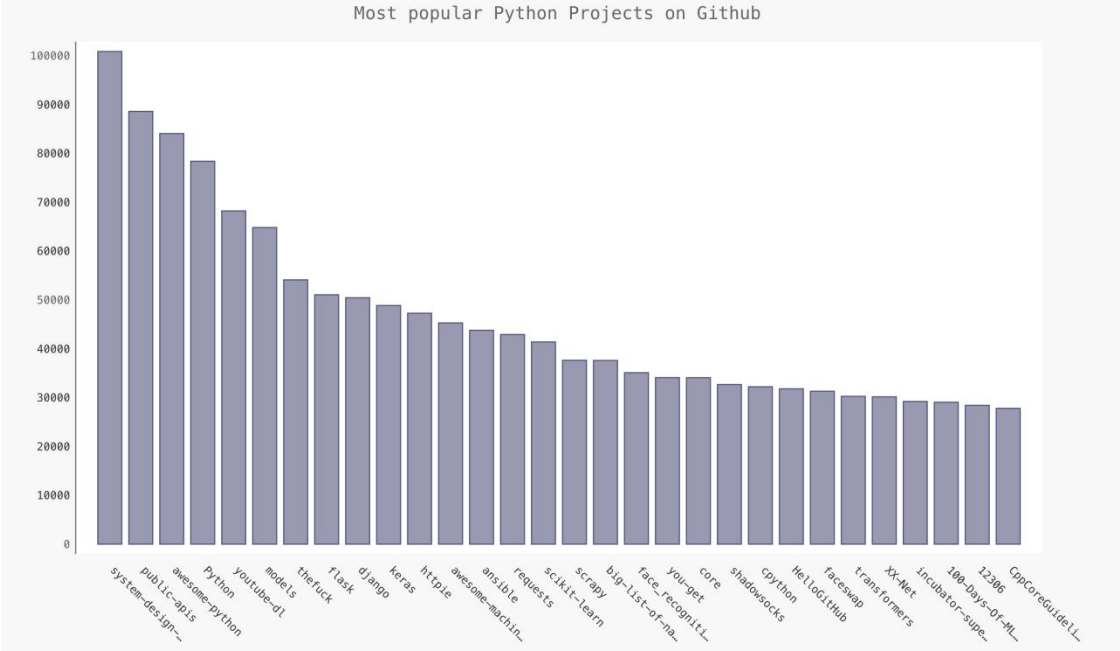


图 12 GitHub 上受欢迎程度最高的 Python 项目

当我们将鼠标移动到某一条上，就会自动显示出对应的总数，可以看出 star 数最多的项目是 system-design-primer，可以进一步查看 star 数达到了 100857 个。

3.4 数据可视化与降维方法 PCA 的结合

PCA 原理：主成分分析（Principal Component Analysis，PCA），是一种统计方法。通过正交变换将一组可能存在相关性的变量转换为一组线性不相关的变量，转换后的这组变量叫主成分。

利用的 Python 函数原型：sklearn.decomposition.PCA(n_components=None)

PCA 最常见的应用之一是将高维数据集可视化。

案例 1 处理 sklearn 里的高维 Iris 数据集

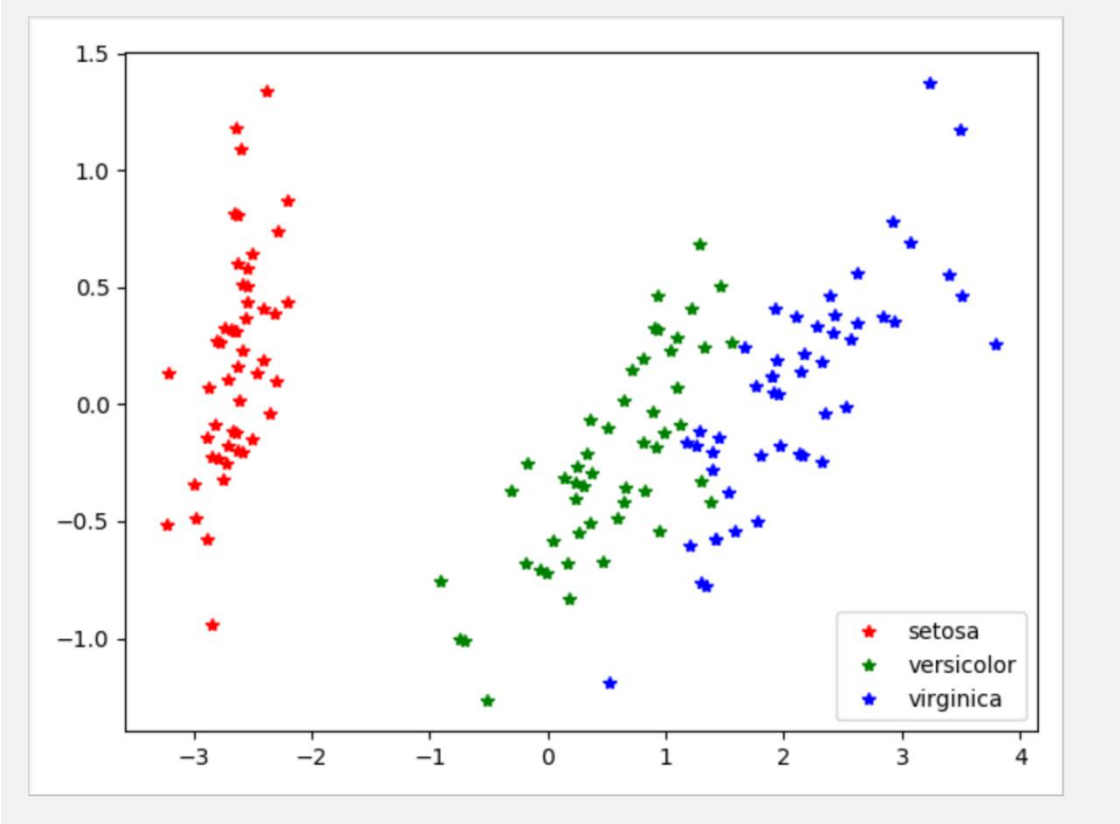
我们以 sklearn 中最常见的 Iris 数据集为例，Iris 数据集包含 150 个样本，分为 3 类，每个样本具有四个特征。

数据集来源：sklearn 里的 datasets 数据集。

分析方法：从 sklearn 里的 datasets 引入 Iris 数据集。

调用 Sklearn 里已经封装好的 PCA 方法，由于样本特征数大于 2 已属于高维数据，我们可以使用 PCA 进行降维，使样本数只包含两个新特征，设置 PCA 的主成分为 2。

使用 Matplotlib 里已经封装好的 plt 方法绘制散点图，直观地表现出两个主成分对 Iris 类别的影响。



案例 2 处理 64 维数字图像，将其降到 2 维

数据集来源：

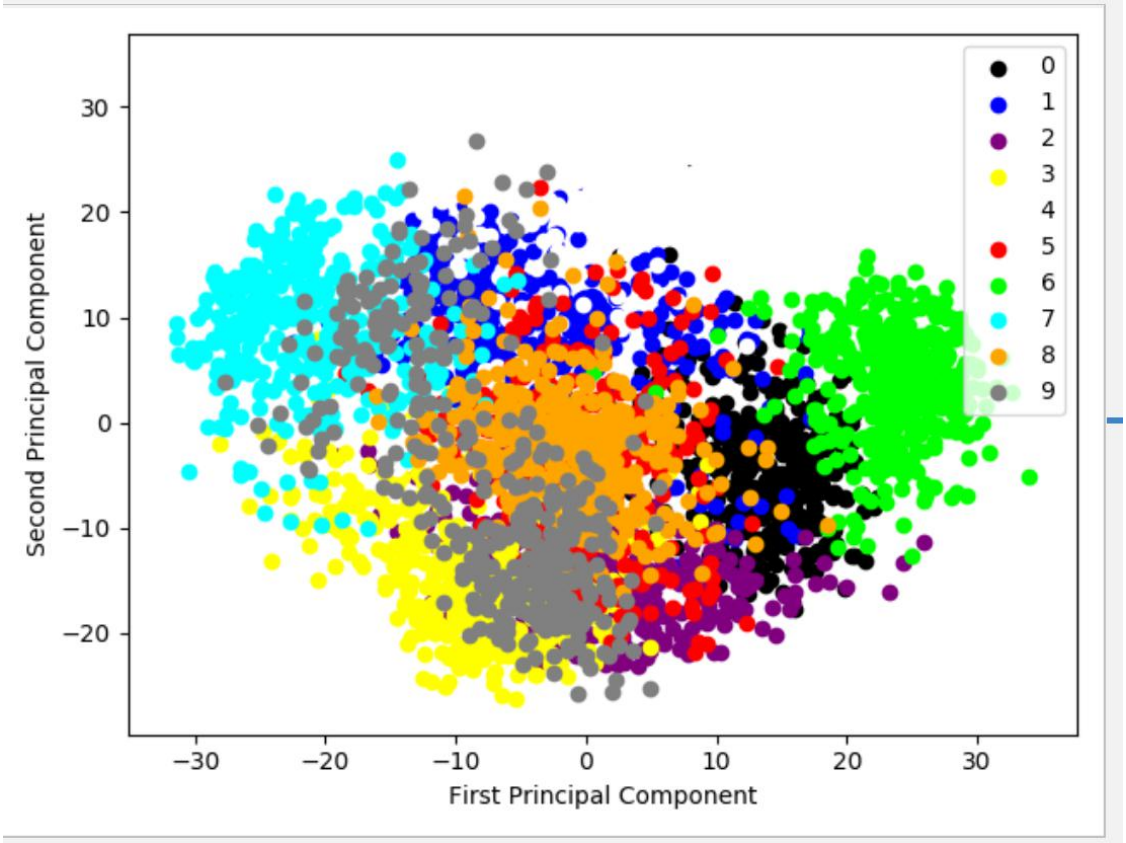
https://archive.ics.uci.edu/ml/machine-learning-databases/optdigits/optdigits.tra

该数据集包含了 3000 多个具有 64 个 Feature 的数字图像样本，可分为 10 类。

分析方法：引入数据集。

调用 Sklearn 里已经封装好的 PCA 方法，由于样本特征数为 64 已属于高维数据，我们可以使用 PCA 进行降维，使样本数只包含两个新特征，设置 PCA 的主成分为 2。

使用 Matplotlib 里已经封装好的 plt 方法绘制图像，直观地表现出两个主成分对 I 图像类别的影响。



图中 10 种不同的颜色代表样本划分的 10 个不同的类别，影响样本类别的 64 个变量因素经过 PCA 降维后,变成 firstcomponent 和 secondcomponent 两个主成分。

3.5 数据可视化与机器学习算法 KNN 的结合

KNN 算法：是机器学习里监督学习中的一种算法。当对新数据点做出预测时，算法会在训练数据集中找到最近的数据点，也就是它的"最邻近"。

例如，如果取 k=1，我们想要预测的数据点的结果就是离它最近的训练数据点的值。

案例与分析方法：在本例中，我们处理与汽车性能有关的数据集，数据集存储在名为 mtcars-clean 的 CSV 文件中，调用 Python 的 Head 方法，该数据集大概内容如下：

```
In [2]: # Load the cars dataset using pandas
        dfcars = pd.read_csv("data/mtcars-cleaned.csv")
        dfcars.head()
```

Out[2]:

	car name	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
1	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
3	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2

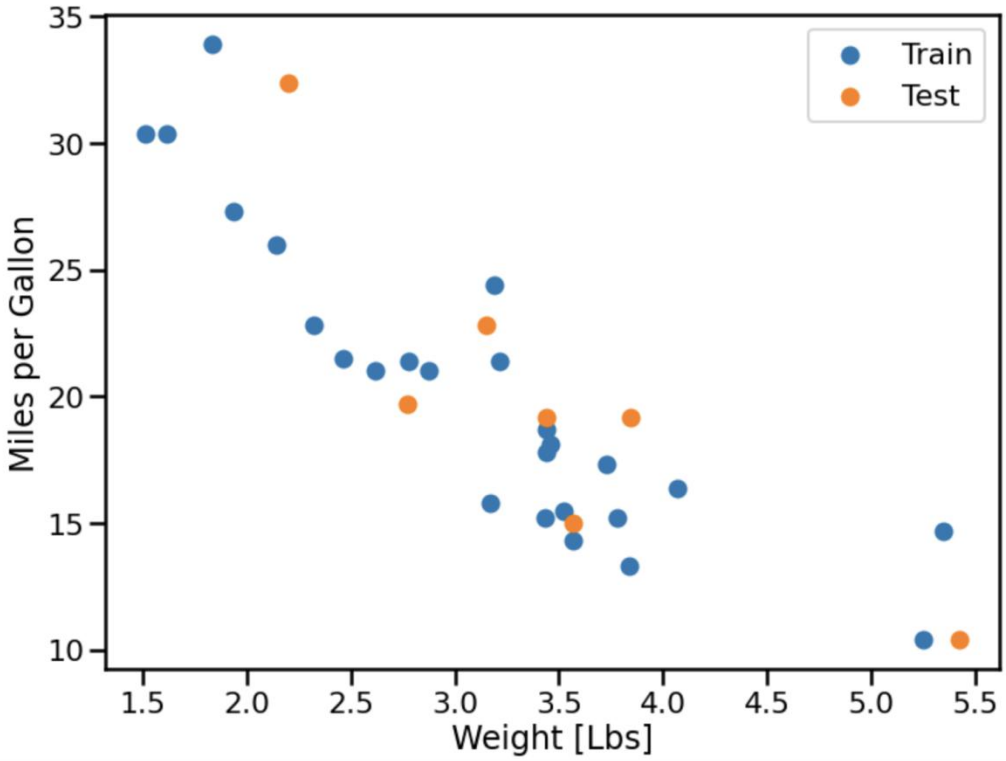
我们想以汽车的 weight（重量）作为自变量，用它来预测汽车的 mpg 值（miles per gallon),所以我们调用 sklearn 里的 test——split 方法，取数据集里 80%的数据为训练数据，20%的数据为测试数据。划分具体方法如下：

```
In [4]: xtrain = dfcars.wt.values[i_train]
        xtrain
Out[4]: array([1.935, 3.73 , 2.62 , 3.44 , 5.345, 3.46 , 3.78 , 4.07 , 3.84 ,
               2.875, 2.32 , 2.14 , 3.215, 3.52 , 1.513, 3.435, 1.615, 2.78 ,
               2.465, 3.19 , 3.44 , 5.25 , 3.17 , 1.835, 3.57 ])
```

click to scroll output; double click to hide

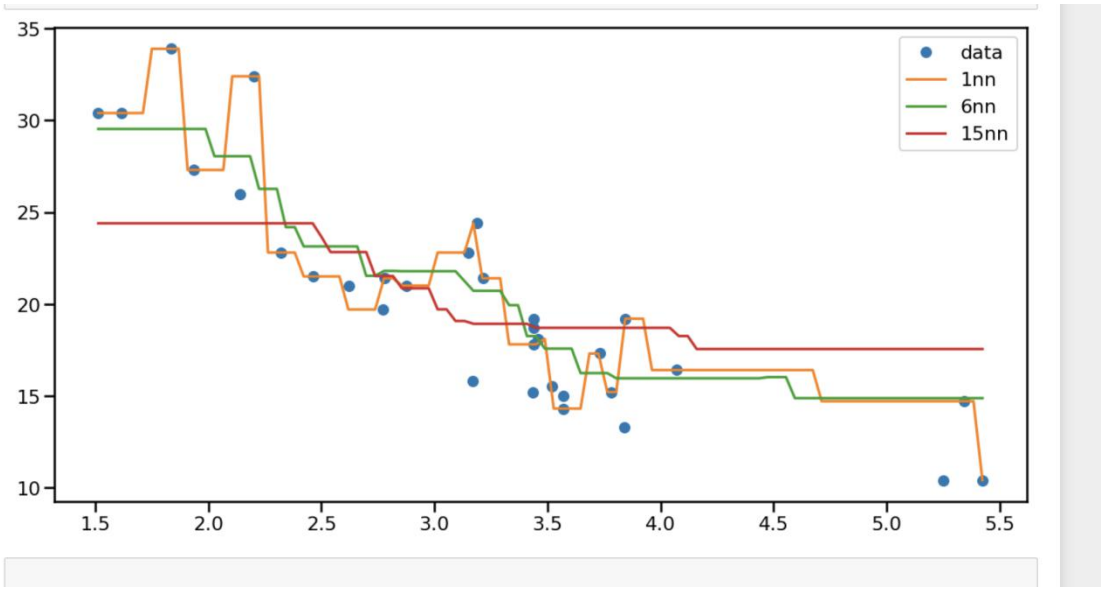
```
In [5]: xtest = dfcars.wt.values[i_test]
        xtest
Out[5]: array([2.77 , 5.424, 3.845, 2.2  , 3.15 , 3.44 , 3.57 ])

In [ ]: ytrain = dfcars.mpg.values[i_train]
        ytest = dfcars.mpg.values[i_test]
        ytrain, ytest
```

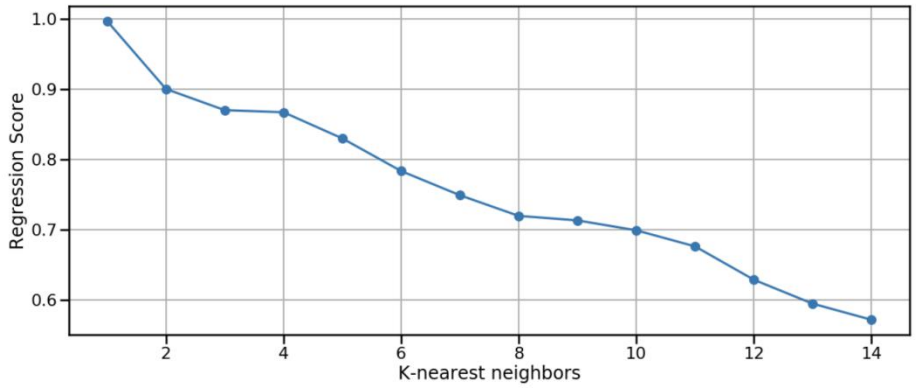
利用可视化工具 subplot 方法可将划分后的数据进行可视化，给人以直观的感受。



接下来我们调用 sklearn 里已经封装好的 KNeighborsRegressor(n_neighbors)方法，让 n_neighbors 遍历 1 到 14 的所有值，利用 knn.score 方法判断不同 K 值下 KNN 算法预测的准确率。

我们将 KNN 的预测正确率随 K 值变化的情况与不同 K 值下预测的数据结果全部可视化，以图表的形式给人以直观的感受。





四 葡萄酒品质—从实际案例总体来看数据可视化

前面我们介绍了 Python 关于数据可视化的基本概念，简单理解了其应用。接下来我们将通过完整地分析一个实际案例来展现出如何利用 Python 进行数据可视化，期间我们也会用到一些数据分析的方法。

4.1 问题描述

数据集存储在源代码文件夹中，包括两种葡萄酒，分别是红葡萄酒和白葡萄酒，还包括了一个数据集说明，这里简单概括一下：

1、数据量方面，红葡萄酒有 1599 条记录，白葡萄酒有 4898 条记录。

2、输入内容包括客观测试（一共 11 个，例如有 pH 值等），输出内容基于感官数据（葡萄酒专家至少进行 3 次评估的中位数）。每位专家都将葡萄酒的质量评定为 0（非常差）至 10（非常好）。

3、关于输入信息说明如下（基于理化性质）：

- fixed acidity：固定酸
- volatile acidity：挥发性酸
- citric acid：柠檬酸
- residual sugar：残留糖
- chlorides：氯化物
- free sulfur dioxide：游离二氧化硫
- total sulfur dioxide：二氧化硫总量
- density：密度
- pH：酸碱值
- sulphates：硫酸盐
- alcohol：酒精

关于输出信息说明如下（基于感官数据）：

- quality (介于 0 到 10 之间)：品质

4、此外，数据集说明中还提到了一些相关信息，主要可以概括为，葡萄酒的质量并不是均匀分布的，普通的葡萄酒要远多于劣质葡萄酒或优质葡萄酒，因此离群值检测算法可用于检测少数优质或劣质葡萄酒。

4.2 思路分析

由于本案例是对葡萄酒进行分析，所以我们要对葡萄酒的标准做一些事先了解，以下部分信息来自葡萄酒新国家标准 GB15037-2006

1、葡萄酒的基本特征主要有酸度、单宁、酒精和甜味。当这四种特征处于一个良好的平衡状态时，葡萄酒的品质才会最优质。

2、酸主要可分为固定酸和挥发酸，常说的总酸就是两者的总和。葡萄酒中含有许多种酸，主要是酒石酸、苹果酸、柠檬酸、琥珀酸、乳酸、醋酸，挥发酸是葡萄酒中以游离状态或以盐的形式存在的所有乙酸等脂肪酸的总和，但不包括乳酸、琥珀酸以及 CO_2 和 SO_2，其中醋酸是主要的挥发酸。挥发酸的含量是葡萄酒健康状态的“体温表”，因为它是发酵、贮藏管理不良留下的标记，通过挥发酸含量的测定可以了解葡萄酒是否生病、病害的严重性以及预测贮藏的困难程度。在国标中对挥发酸和柠檬酸做了明确规定。

挥发酸（以乙酸计）/(g/L)	≤	1.0
柠檬酸/(g/L) ≤	干、半干、半甜葡萄酒	1.0
	甜葡萄酒	2.0

图 13 新国标中对挥发酸和柠檬酸的规定

注：总酸不作要求，以实测值表示（以酒石酸记，g/L）

3、pH 值是衡量葡萄酒中酸度的程度，一般来说，白葡萄酒的酸度一般在 3.1 至 3.5 之间，高于红葡萄酒的 3.5 至 4 的区间值。相较而言酸度是衡量葡萄酒中酸含量的多少。

4、一般来说，酸度对葡萄酒口感的影响要大于 pH 值，但如果 pH 值位于一个极端的位置，就会产生较大的影响。总酸度是告诉我们这款酒的浓度，而 pH 值显示的是这款酒品尝起来口感的浓郁度。例如，在 pH 值相同的情况下，一款总酸度为 6g/L 的葡萄酒品尝起来会比总酸度为 4g/L 的葡萄酒更酸。

5、残留糖分（简称残糖）是衡量葡萄酒中甜度的标准。通常，残留糖分低于 4 克/升的葡萄酒为干型葡萄酒，许多干型葡萄酒几乎不含残糖。

总糖[a]（以葡萄糖计）/(g/L)	平静葡萄酒	干葡萄酒[b]	≤	4.0
		半干葡萄酒[c]		4.1～12.0
		半甜葡萄酒		12.1～45.0
		甜葡萄酒	≥	45.1
	高泡葡萄酒	天然型高泡葡萄酒	≤	12.0（允许差为3.0）
		绝干型高泡葡萄酒		12.1～17.0（允许差为3.0）
		干型高泡葡萄酒		17.1～32.0（允许差为3.0）
		半干型高泡葡萄酒		32.1～50.0
		甜型高泡葡萄酒	≥	50.1

图 14 新国标中对糖分的标准

6、酒精度指葡萄酒中所含酒精的百分比，大部分葡萄酒的酒精度都在 10-15%之间，但也有些特殊的葡萄酒，如阿斯蒂（Moscato d’Asti）（酒精度非常低），波特酒（Port）（酒精度非常高）。

酒精度[a]（20℃）/%（体积分数）	≥	葡萄酒	7.0

图 15 新国标中对酒精度的规定

7、氯化物和硫酸盐都属于葡萄酒中的矿物盐成分，一般来说含量分别是 0.1-0.4g/L 和 0.25-0.85g/L。值得一提的是，虽然这些矿物质成分存在葡萄酒中且可以增强葡萄酒的风味，但它们并不是某些葡萄酒带有矿物风味的主要原因。一般而言，红葡萄酒所含的矿物质多于白葡萄酒。

8、并不是所有葡萄酒中都会有二氧化硫，但二氧化硫能起到如杀菌、抗氧化、澄清酒液和提高色素和酚类物质含量等作用，因此一般葡萄酒中或多或少地带有一定的二氧化硫，只是整体而言其含量非常少，多为 80-200mg/L，个别葡萄酒中还含有 10-50mg/L 的游离态二氧化硫。不过，适当的摇杯或者醒酒等可以令其挥发掉，因此几乎可以忽略不计。

总二氧化硫/(mg/L)	≤	干葡萄酒	200
		其他类型葡萄酒	250

图 16 新国标中对二氧化硫的规定

总结：根据查到的信息，更新输入变量表格如下：

输入指标	说明	备注
fixed acidity	固定酸（g/L）	总酸组成之一
volatile acidity	挥发性酸（g/L）	总酸组成之一
citric acid	柠檬酸（g/L）	属于固定酸
residual sugar	残留糖（g/L）	基本指标之一
chlorides	氯化物（g/L）	矿物盐成分
free sulfur dioxide	游离二氧化硫（mg/L）	防腐保鲜剂
total sulfur dioxide	二氧化硫总量（mg/L）	防腐保鲜剂
density	密度（g/ml）	略
pH	酸碱值	酸度的另一种测量角度
sulphates	硫酸盐（g/L）	矿物盐成分
alcohol	酒精 （%vol）	基本指标之一

由此可知，11 种输入变量可以大致划分成三类，第一类是基本指标及其内含的个别具体指标，第二类是附加指标（矿物盐、二氧化硫），第三类是密度这个物理性质。

4.3 数据分析

4.3.1 数据整理

使用语言为 Python，编程工具为 PyCharm 和 Jupyter Notebook。首先我们导入需要用到的库：numpy、pandas、matplotlib 等，然后我们进行对数据集文件的读取。我们先打印出两个数据集的 head 部分。

In [2]:
```
dfr.head()
```

Out[2]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.998	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.997	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.997	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.998	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.998	3.51	0.56	9.4	5

In [3]:
```
dfw.head()
```

Out[3]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.001	3.00	0.45	8.8	6
1	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.994	3.30	0.49	9.5	6
2	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.995	3.26	0.44	10.1	6
3	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.996	3.19	0.40	9.9	6
4	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.996	3.19	0.40	9.9	6

图 17

因为总酸作为葡萄酒基本指标值之一，是固定酸和挥发酸的合，所以可以在表中增加一列"total acid"作为总酸，并放置在表格首列，这样数据集就变成了：

In [5]:
```
dfr.describe()
```

Out[5]:

	total acid	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	1599.000	1599.000	1599.000	1599.000	1599.000	1599.000	1599.000	1599.000	1599.000	1599.000	1599.000	1599.000	1599.000
mean	8.847	8.320	0.528	0.271	2.539	0.087	15.875	46.468	0.997	3.311	0.658	10.423	5.636
std	1.704	1.741	0.179	0.195	1.410	0.047	10.460	32.895	0.002	0.154	0.170	1.066	0.808
min	5.120	4.600	0.120	0.000	0.900	0.012	1.000	6.000	0.990	2.740	0.330	8.400	3.000
25%	7.680	7.100	0.390	0.090	1.900	0.070	7.000	22.000	0.996	3.210	0.550	9.500	5.000
50%	8.445	7.900	0.520	0.260	2.200	0.079	14.000	38.000	0.997	3.310	0.620	10.200	6.000
75%	9.740	9.200	0.640	0.420	2.600	0.090	21.000	62.000	0.998	3.400	0.730	11.100	6.000
max	16.285	15.900	1.580	1.000	15.500	0.611	72.000	289.000	1.004	4.010	2.000	14.900	8.000

In [6]:
```
dfw.describe()
```

Out[6]:

	total acid	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	4898.000	4898.000	4898.000	4898.000	4898.000	4898.000	4898.000	4898.000	4898.000	4898.000	4898.000	4898.000	4898.000
mean	7.133	6.855	0.278	0.334	6.391	0.046	35.308	138.361	0.994	3.188	0.490	10.514	5.878
std	0.848	0.844	0.101	0.121	5.072	0.022	17.007	42.498	0.003	0.151	0.114	1.231	0.886
min	4.110	3.800	0.080	0.000	0.600	0.009	2.000	9.000	0.987	2.720	0.220	8.000	3.000
25%	6.570	6.300	0.210	0.270	1.700	0.036	23.000	108.000	0.992	3.090	0.410	9.500	5.000
50%	7.070	6.800	0.260	0.320	5.200	0.043	34.000	134.000	0.994	3.180	0.470	10.400	6.000
75%	7.590	7.300	0.320	0.390	9.900	0.050	46.000	167.000	0.996	3.280	0.550	11.400	6.000
max	14.470	14.200	1.100	1.660	65.800	0.346	289.000	440.000	1.039	3.820	1.080	14.200	9.000

图 18

因为不存在值必须唯一的变量且需要分析分类数量，故此处不对数据集进行去重（经试验两个数据集都存在一定数量的重复值），而且经检查不存在异常值。

4.3.2 分类讨论

11 种输入变量可以大致划分成三类，第一类是基本指标及其内含的个别具体指标，第二类是附加指标（矿物盐、二氧化硫），第三类是密度这个物理性质。

4.3.2.1 描述统计

在统计指标之前，应当对各个变量有直观的分析。

数值描述:

参考图 18

由上面结果首先可以知道参与测试的红葡萄酒获得的评分分布在 3-8 分，白葡萄酒在 3-9 分，且各分位数数据一致，这说明两种葡萄酒的品质在大体上无明显区别，但具体是否存在细节上的差距还需要进一步分析。另外对于输入变量的各参数，数据表格展示的形式太繁杂不直观，一时看不出什么信息，需要进一步加工成图像便于分析比对。

箱线图:

我们来绘制两种葡萄酒每个变量的箱线图

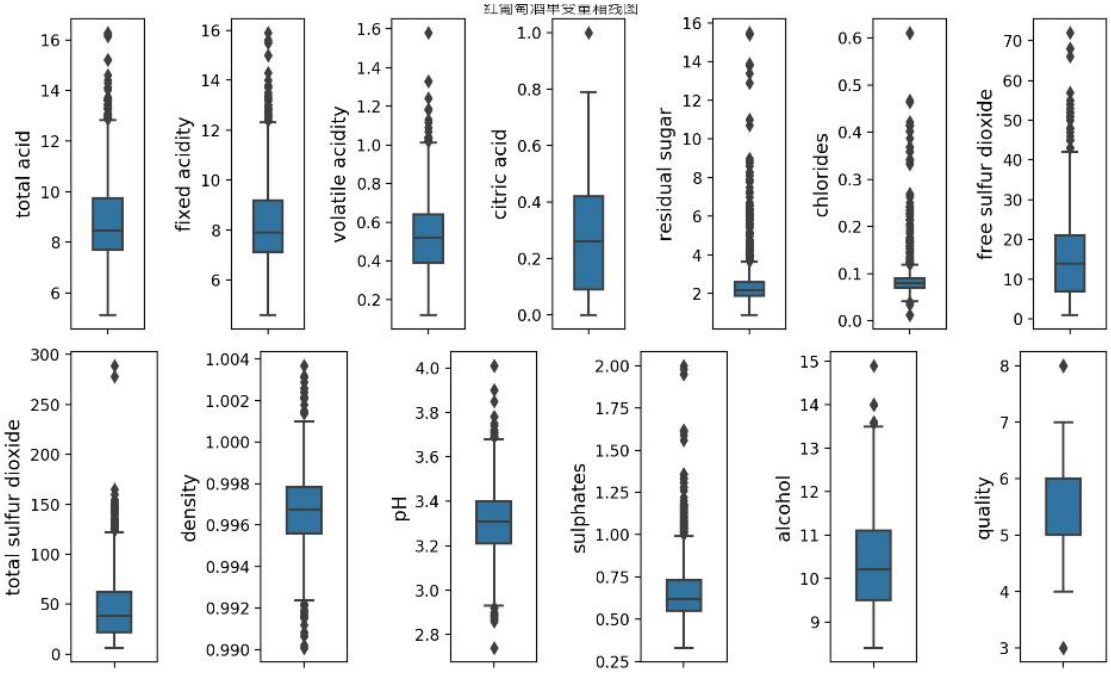


图 19 红葡萄酒单变量箱线图

结合箱线图和刚才的数据表格可以直观了解到各个变量的分布特征，大致归纳如下。

红葡萄酒变量	分布特征
total acidity	整体呈正偏，尾长较为对称，高浓度部分存在一定量离群点
fixed acidity	整体呈正偏，尾长较为对称，高浓度部分存在一定量离群点，整体分布与总酸相近
volatile acidity	整体呈正偏，尾长较为对称，高浓度部分存在一定量离群点，浓度范围低总酸一个数量级
citric acid	整体呈正偏，上尾长较长，最小值取到 0，有极个别离群点取到 1
residual sugar	整体呈高度正偏，尾长较为对称，存在大量高浓度离群点
chlorides	整体呈高度正偏，尾长较为对称，存在少量低浓度离群点和大量高浓度离群点
free sulfur dioxide	整体呈正偏，上尾长较长，存在一定量高浓度离群点
total sulfur dioxide	整体呈正偏，上尾长较长，存在一定量高浓度离群点且离群点间断大
density	整体几乎呈正态分布，上下各有一定量离群点
pH	整体呈轻度正偏，尾长较为对称，存在少量低值离群点和一定量高值离群点
sulphates	整体呈正偏，尾长较为对称，存在较多高浓度离群点
alcohol	整体呈正偏，上尾长较长，存在少量高浓度离群点
quality	整体几乎呈正态分布，上下各有极少量离群点

可以大致总结知道，对于红葡萄酒而言，除密度和评分这两项数据分布均匀呈正态之外，其他所有变量都呈现出不同程度的正偏分布，这说明大多数变量都存在可控下限却没有明确的上限，由于品质波动都可能出现较高取值的情况。同理，我们画出白葡萄酒单变量箱线图。

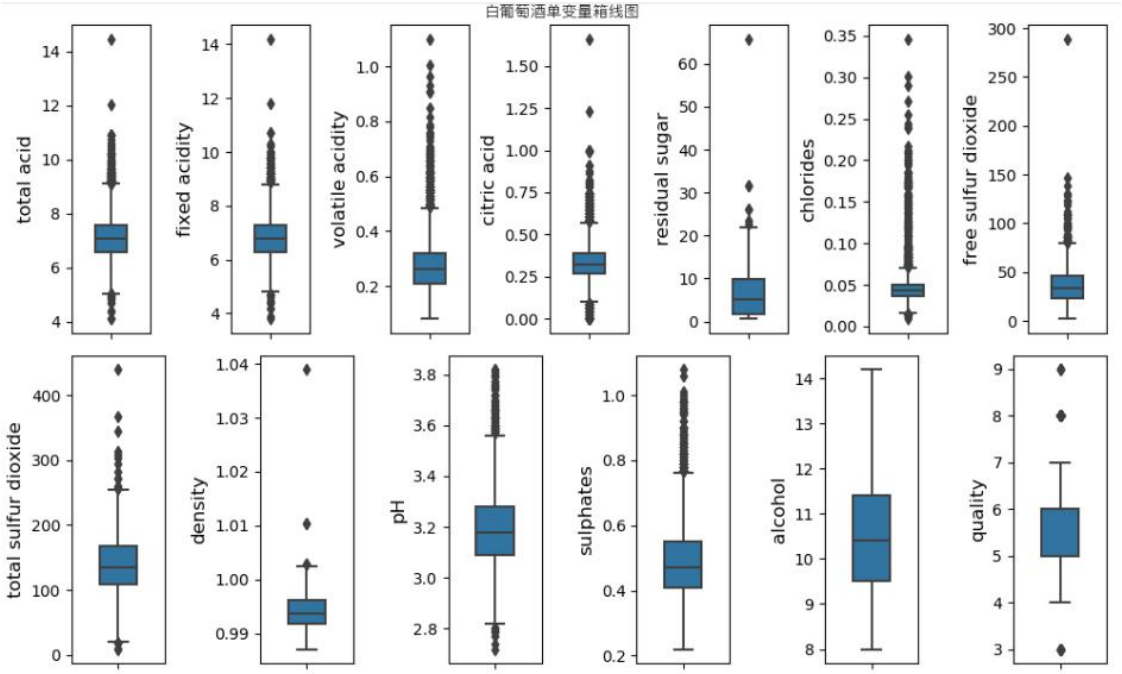


图 20 白葡萄酒单变量箱线图

我们也可以得到白葡萄酒各个变量的分布特征。

白葡萄酒变量	分布特征
total acidity	整体呈正偏，尾长较为对称，存在少量低离群点和一定量高离群点
fixed acidity	整体呈正偏，尾长较为对称，存在少量低离群点和一定量高离群点，整体分布与总酸相近
volatile acidity	整体呈正偏，尾长较为对称，存在大量高离群点，浓度范围低总酸一个数量级
citric acid	整体呈正偏，尾长较为对称，存在少量低离群点和一定量高离群点
residual sugar	整体呈正偏，上尾长较长，存在少量高浓度离群点且离群点间断大
chlorides	整体呈高度正偏，尾长较为对称，存在少量低浓度离群点和大量高浓度离群点
free sulfur dioxide	整体呈正偏，上尾长较长，存在一定量高浓度离群点且离群点间断大
total sulfur	整体呈正偏，尾长较为对称，存在少量低离群点一定量高浓度离群点

dioxide	
density	整体呈轻度正偏，存在极少量高离群点
pH	整体呈轻度正偏，尾长较为对称，存在少量低离群点和一定量高离群点
sulphates	整体呈正偏，尾长较为对称，存在较多高浓度离群点
alcohol	整体呈正偏，上尾长较长，无离群点
quality	整体几乎呈正态分布，上下各有极少量离群点

可以发现白葡萄酒也是在绝大部分变量上呈现正偏分布，且相比红葡萄酒有更多变量有低离群点，整体上红葡萄酒和白葡萄酒在一些变量上表现不太相同，这些指标可能是造成品类不同的主要因素之一。

我们接下来将两个箱线图放在一起进行观察，可以得出以下结论：

- 在酸度上白葡萄酒取值低于红葡萄酒且分布更紧凑；
- 在残留糖浓度上白葡萄酒分布更广泛，相比之下红葡萄酒的分布就很紧凑；
- 在氯化物浓度上白葡萄酒取值低于红葡萄酒，二者的分布都比较分散；
- 在二氧化硫浓度上白葡萄酒取值高于红葡萄酒；
- 在密度上二者的绝大部分取值均低于水的密度，白葡萄酒整体密度更低但分布范围更大；
- 在 pH 上二者整体分布相似，白葡萄酒取值整体低于红葡萄酒；
- 在硫酸盐酸浓度上白葡萄酒整体取值低于红葡萄酒；
- 在酒精浓度上二者分布相近且离群点很少；
- 在品质评分上二者十分相近，除白葡萄酒有 9 分取值外几乎无异；

直方图

除了箱线图，还可以通过直方图从另一种角度观察每种变量的分布情况。

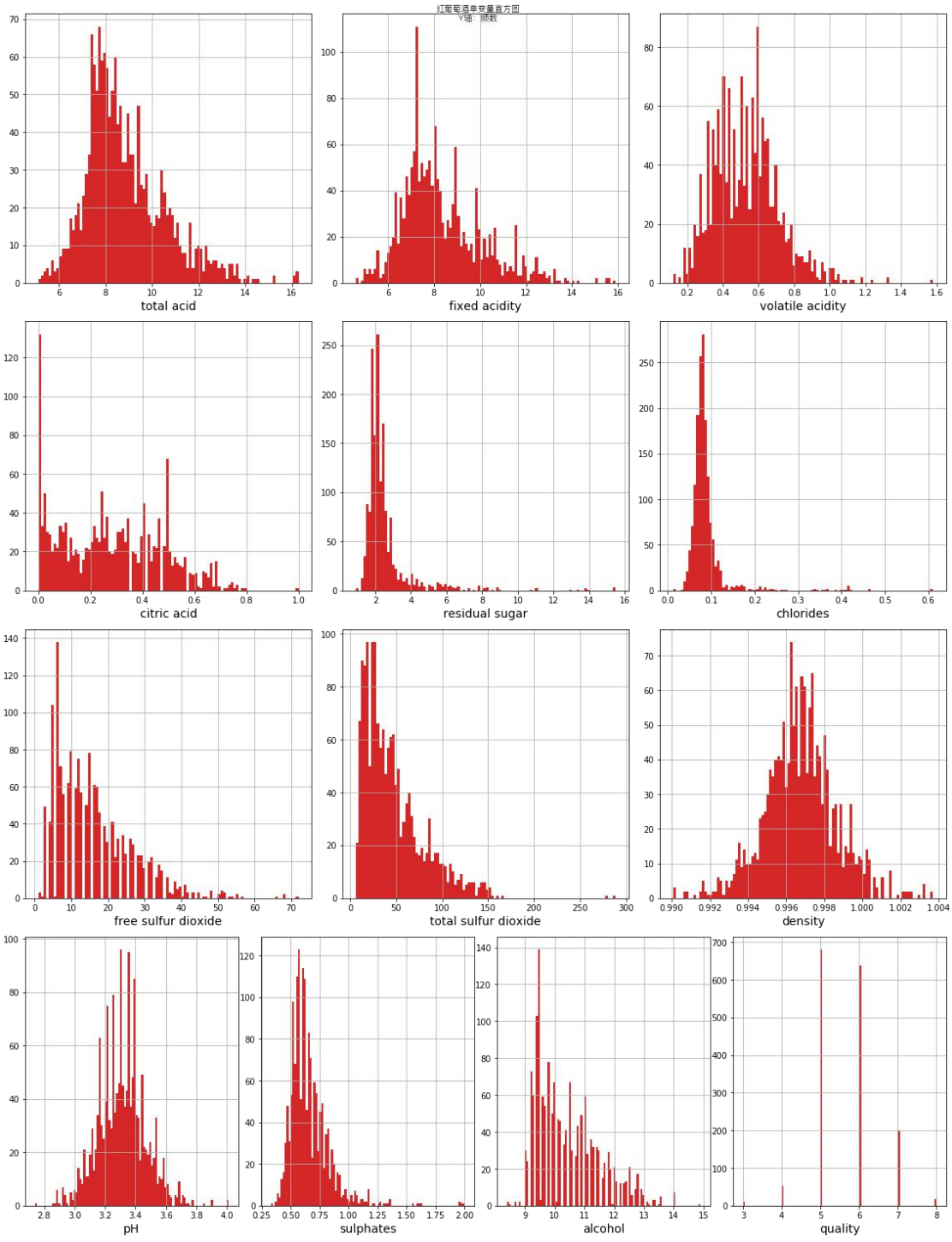


图 21 红葡萄酒单变量直方图

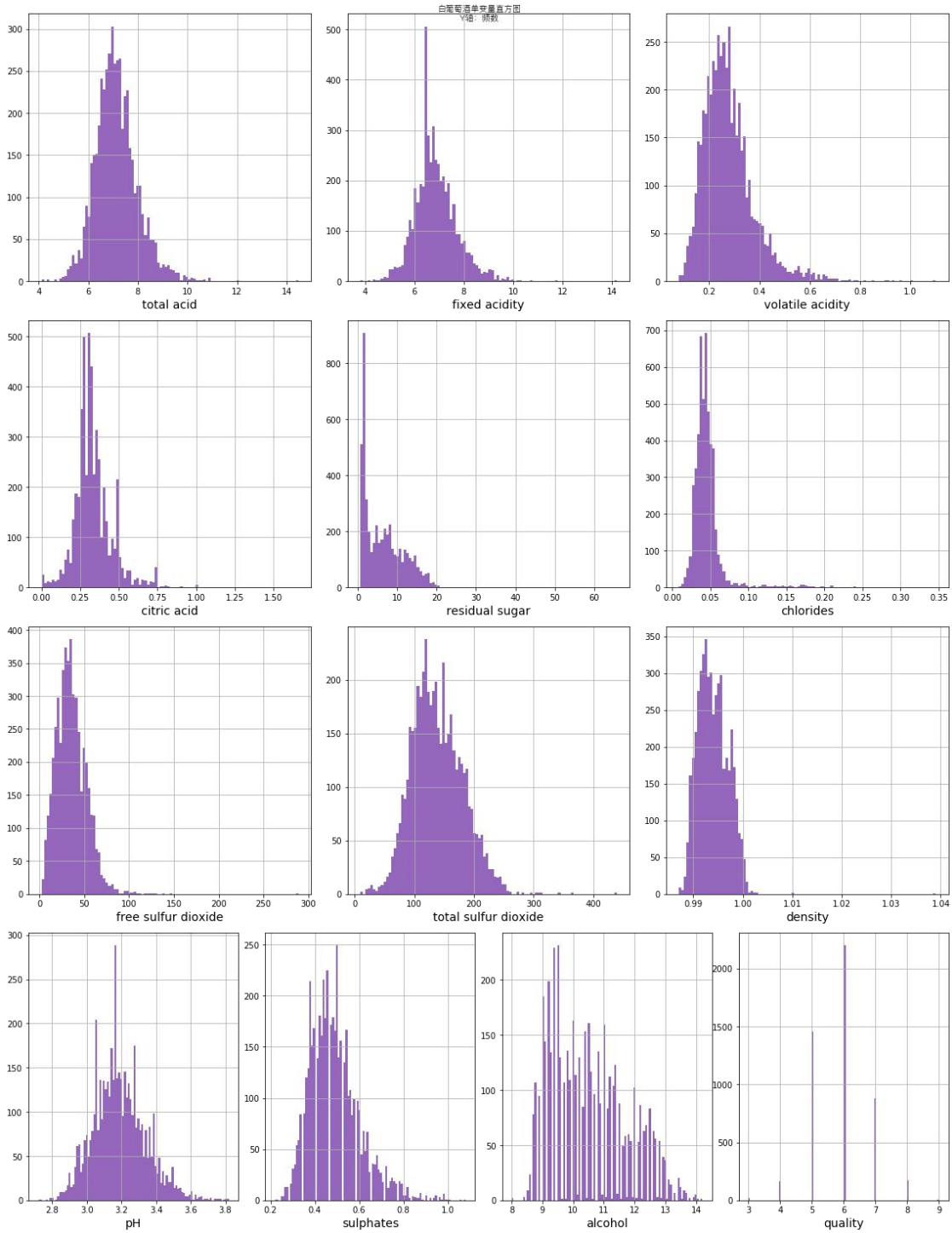


图 22 白葡萄酒单变量直方图

当然，也可以将两种葡萄酒的变量直方图放在一张图上进行更为直观的对比分析，结果如下：

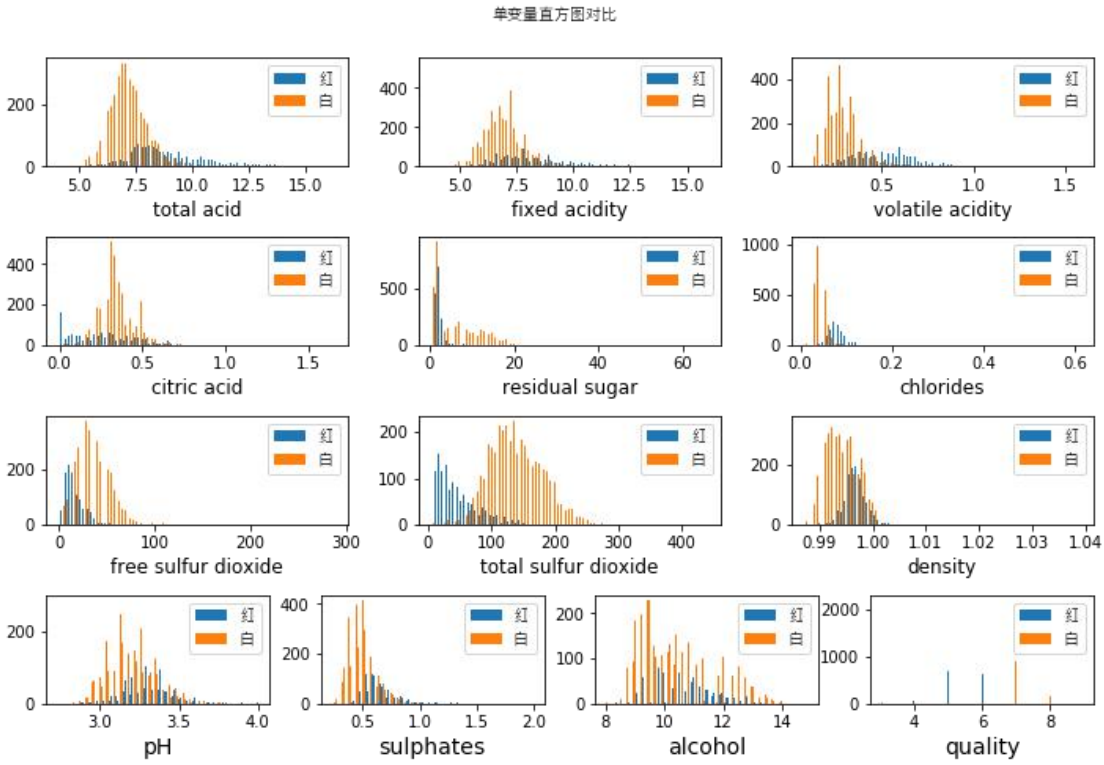


图 23 红白葡萄酒单变量直方图

4.3.2.2 基本指标分析

酸度分析

输入变量中酸度有关的变量占到了三个，需逐个对其进行分析。首先是固定酸在总酸中的占比，观察占比的分布情况。

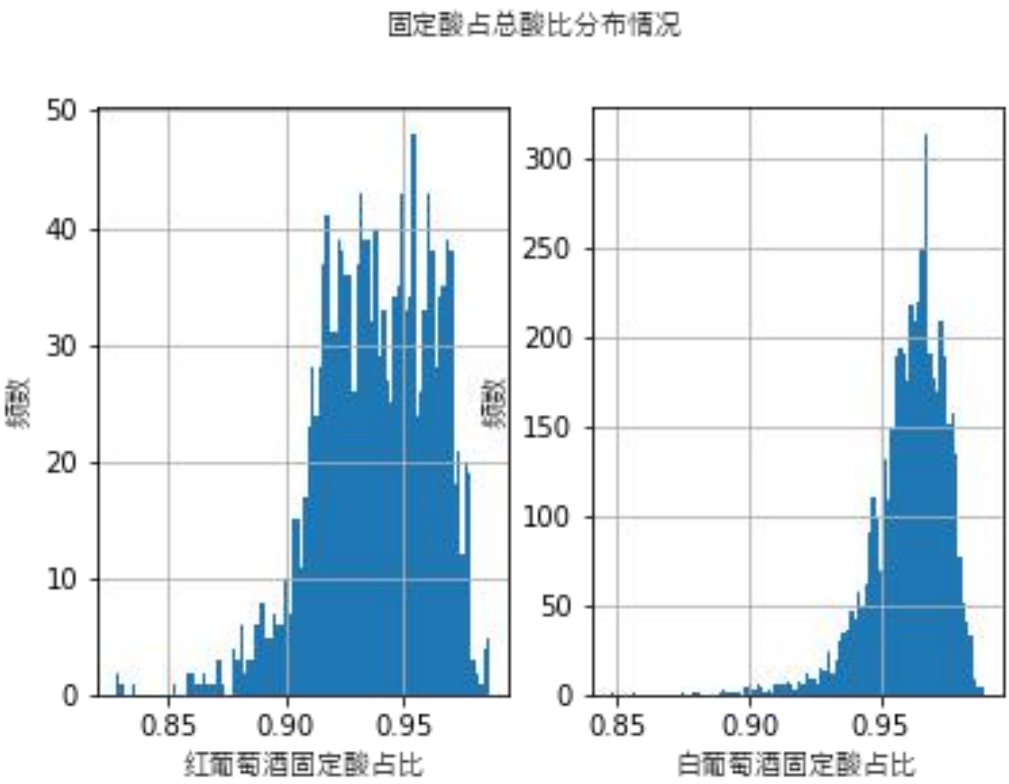


图 24 红白葡萄酒固定酸占总酸比分布图

可以发现红葡萄酒中固定酸的占比比较分散，而白葡萄酒中固定酸的占比则比较集中，形成一个单峰分布。两种葡萄酒的固定酸占比大多数情况下都达到了90%以上。

其次是关于固定酸占总酸比重对评分的影响。

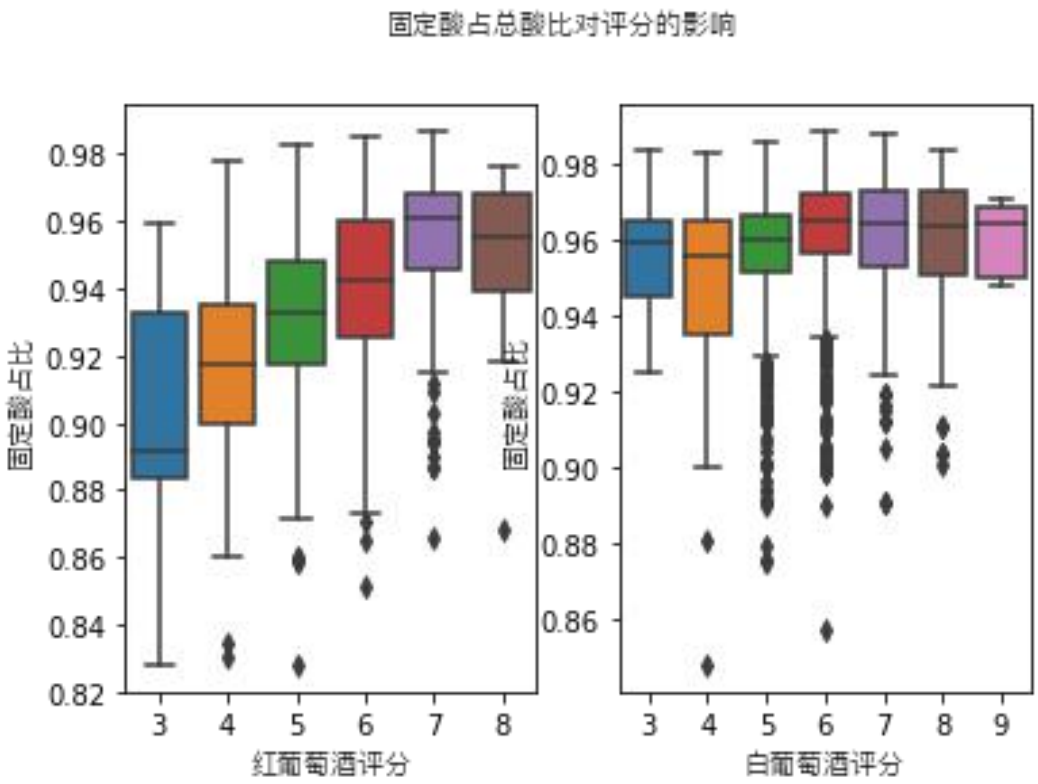


图 25 红白葡萄酒固定酸占总酸比影响图

可以发现随着占比的提高，红葡萄酒有更大可能取得较高的评分，而占比的提高对白葡萄酒影响不显著。

柠檬酸是固定酸的一种，若观察固定酸和柠檬酸的关系，首先可以观察柠檬酸在固定酸中占比的分布情况。

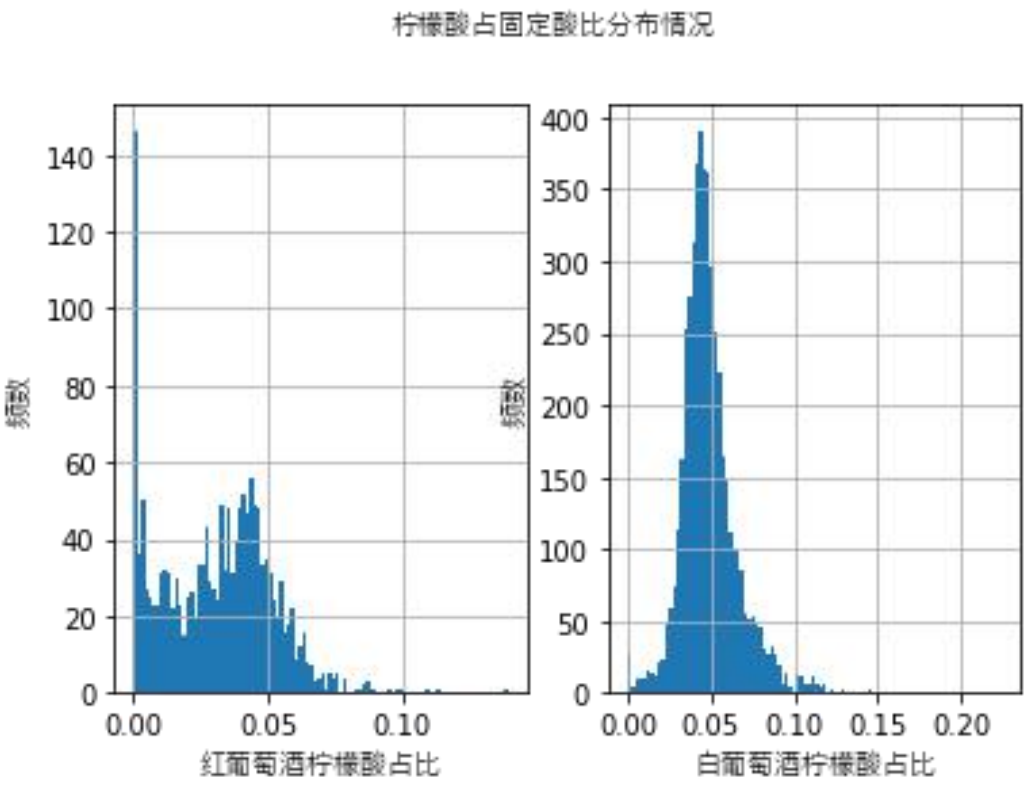


图 26 红白葡萄酒柠檬酸占固定酸比分布图

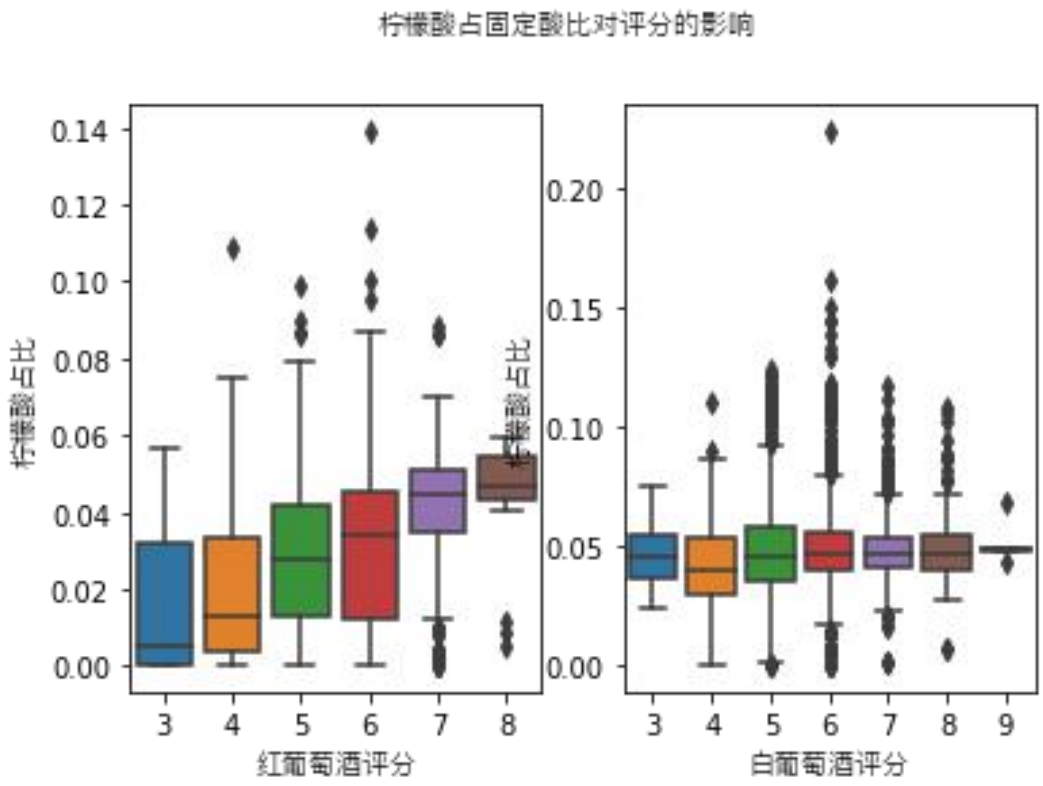


图 27 红白葡萄酒柠檬酸占固定酸比影响图

从两图看出，柠檬酸的占比在红葡萄酒中比较分散，大部分分布于 0-8%，且有大量 0 值。白葡萄酒中柠檬酸占比呈明显的单峰分布，大概集中于 4.5%附近。

随着柠檬酸占比的提高，红葡萄酒有更大可能取得较高的评分。而占比的提高对白葡萄酒影响不显著，但过高的占比会导致评分处于中间分段。

挥发酸在评价中属不良指标，所以挥发酸含量对评分的影响应该能形成规律，以此观察挥发酸在总酸中的占比的分布情况。

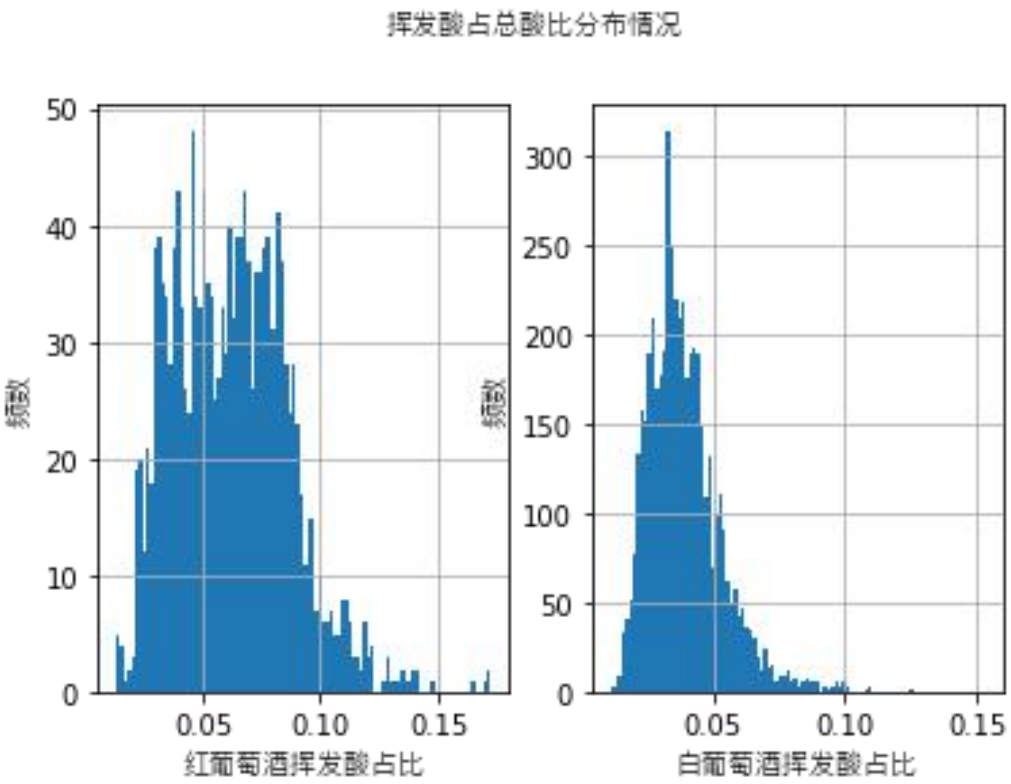


图 28 红白葡萄酒挥发酸占总酸比分布图

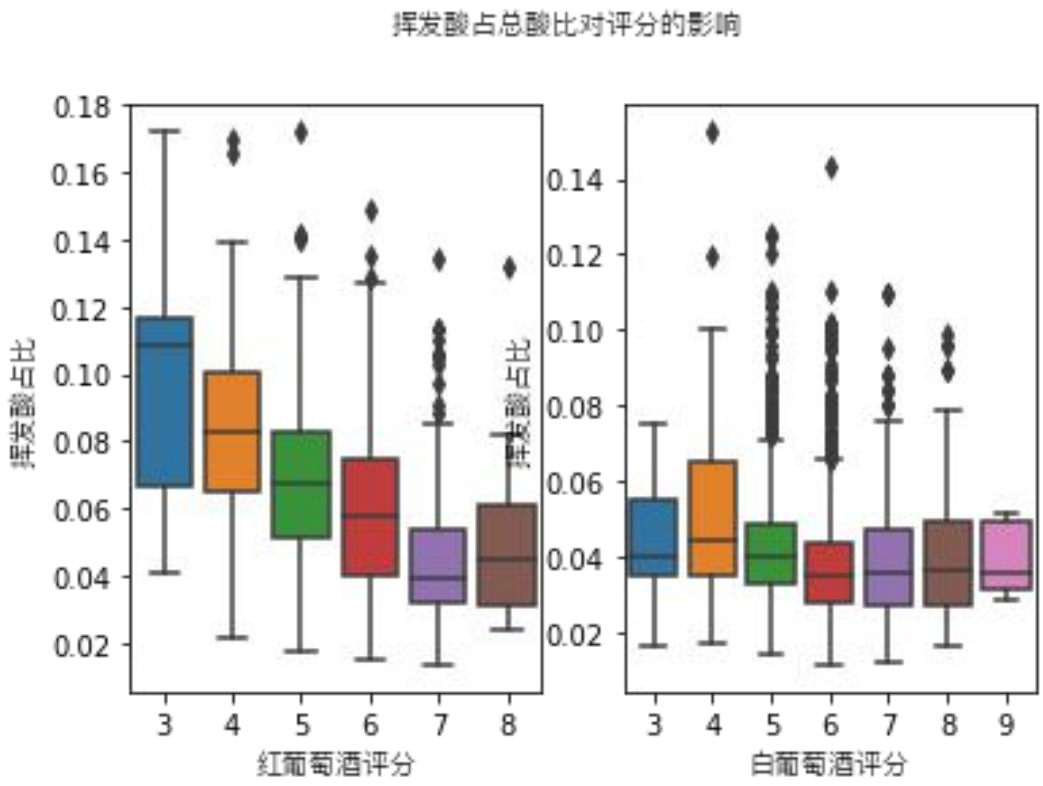


图 29 红白葡萄酒挥发酸占总酸比影响图

可以发现红葡萄酒中挥发酸的占比比较分散，大部分分布在 2.5%-10%之间。而白葡萄酒中固定酸的占比则比较集中，形成一个单峰分布，大概集中在 3%附近。

通过观察影响图不难看出，随着挥发酸占比降低，红葡萄酒有更大可能取得较高得分，而占比的降低对白葡萄酒的影响不显著。

对于酸度的考量不只是酸度，pH 也是对酸性的度量，只是角度不同，需观察二者对评分结果的影响力是否有差距。

首先是总酸对评分的影响：

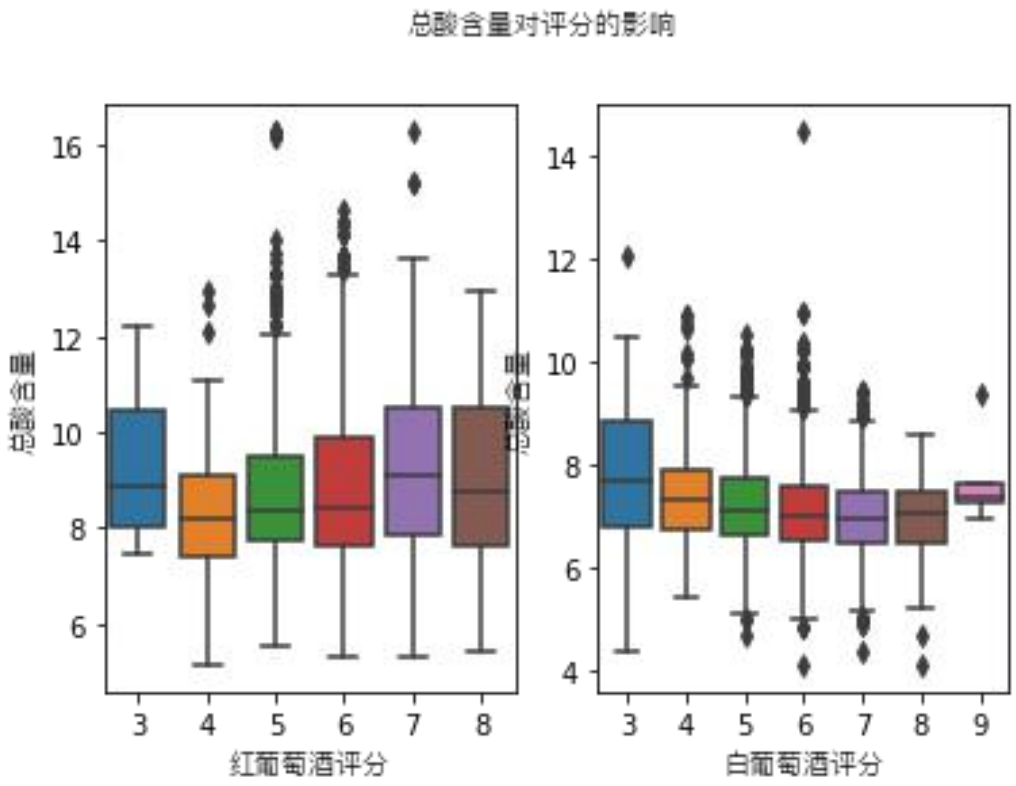


图 30 总酸含量对评分影响

其次是 pH 值对评分的影响:

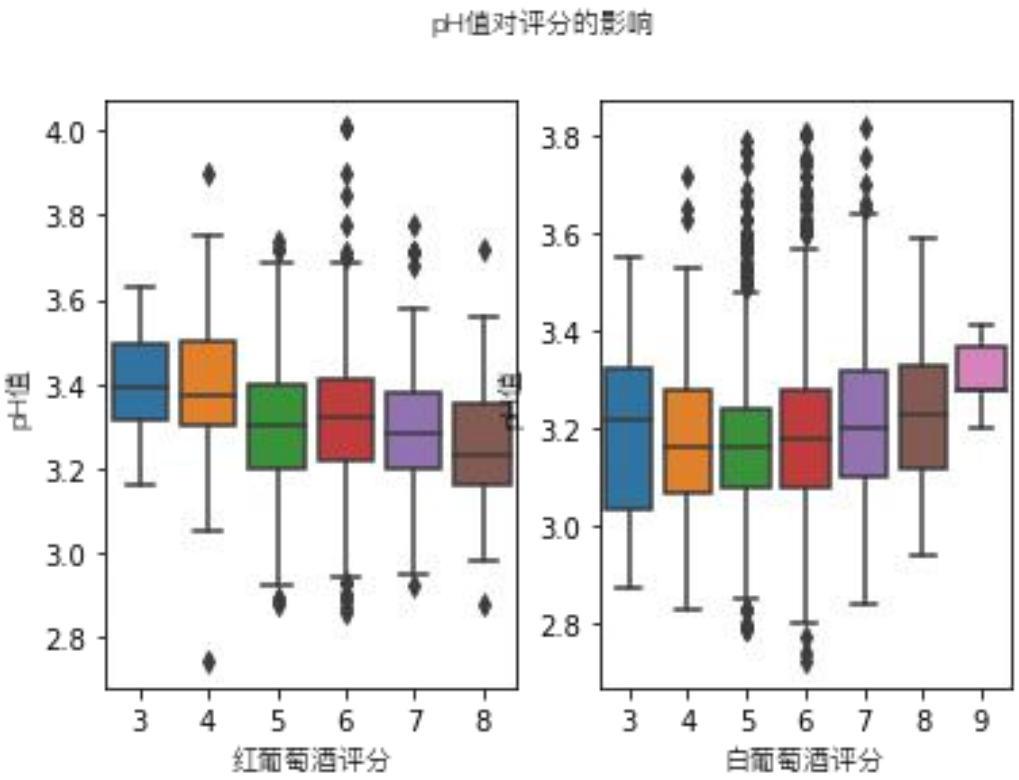


图 31 pH 值对评分影响

可以发现，总酸含量的变化对红白葡萄酒均没有显著的影响，而 pH 值的影响较为显著。

有意思的一点是，pH 值得降低会使红葡萄酒有更大可能获得高分评价，却使白葡萄酒有更大可能获得低分评价，它的影响是恰好相反的。

国标中有根据柠檬酸的浓度对葡萄酒分类的标准，此处按这个标椎看一下分类结果：

挥发酸（以乙酸计）/(g/L)	≤	1.0
柠檬酸/(g/L) ≤	干、半干、半甜葡萄酒	1.0
	甜葡萄酒	2.0

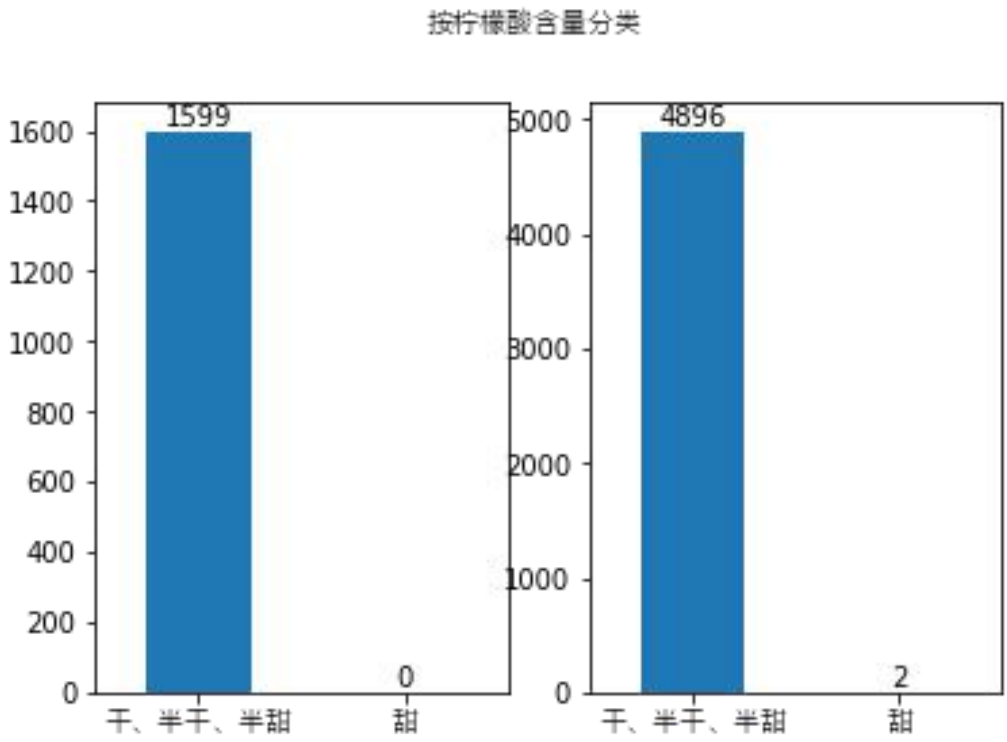


图 33 依据柠檬酸含量分类

按照国标，实验中所有的红葡萄酒都属于（干、半干、半甜）类，没有甜葡萄酒；实验中 99%的白葡萄酒都属于（干、半干、半甜）类，只有 2 例甜葡萄酒。

糖度分析

残留糖作为分析基本指标之一，先观察其对评分的影响。

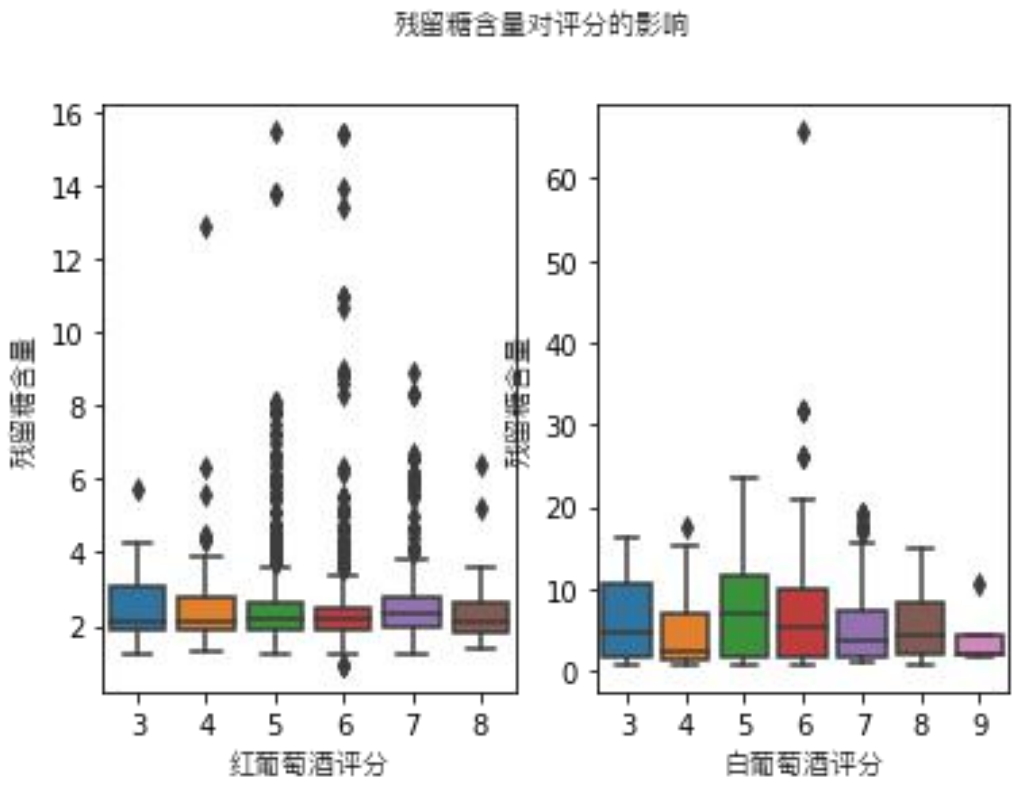


图 34 残留糖对评分影响图

除去离群点，各分段的红葡萄酒的残留糖含量都比较相近，可以认为残留糖含量对红葡萄酒的评分影响不大。可以观察到 8 分红葡和 3 分红葡的残留糖含量分布区间很相近，而高浓度离群点主要出现在中间分段，不见于高分段和低分段。

类似地，除去离群点，各分段的白葡萄酒的残留糖含量都比较相近，可以认为残留糖含量对白葡萄酒的评分影响不大。可以观察到 9 分白葡和 3 分白葡的残留糖含量分布区间很相近，而高浓度离群点主要出现在中间分段，不见于高分段和低分段。

国标中有根据糖分的浓度对葡萄酒分类的标准，此处虽不严格复合（应以葡萄糖计），但也可按这个标椎大致看一下分类结果。

总糖[a]（以葡萄糖计）/(g/L)	平静葡萄酒	干葡萄酒[b]	≤	4.0
		半干葡萄酒[c]		4.1～12.0
		半甜葡萄酒		12.1～45.0
		甜葡萄酒	≥	45.1
	高泡葡萄酒	天然型高泡葡萄酒	≤	12.0（允许差为3.0）
		绝干型高泡葡萄酒		12.1～17.0（允许差为3.0）
		干型高泡葡萄酒		17.1～32.0（允许差为3.0）
		半干型高泡葡萄酒		32.1～50.0
		甜型高泡葡萄酒	≥	50.1

适用于柠檬酸的策略，同样的，我们以残留糖进行分类。

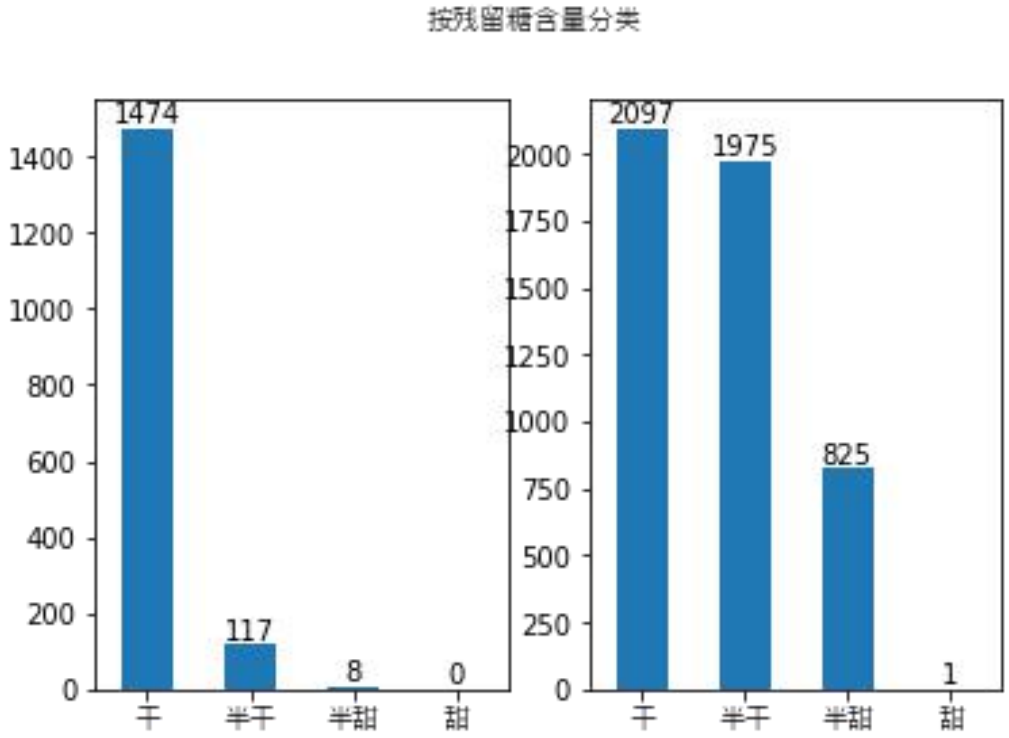


图 36 按残留糖分类图

按照国标，实验中干红葡萄酒有 1474 例，占 92.2%，半干红葡萄酒有 117 例，占 7.3%，半甜红葡萄酒有 8 例，占 0.5%，无甜红葡萄酒；实验中干白葡萄酒有 2097 例，占 42.82%，半干白葡萄酒有 1975 例，占 40.32%，半甜白葡萄酒有 825 例，占 16.84%，甜白葡萄酒 1 例，占 0.02%。

酒精分析

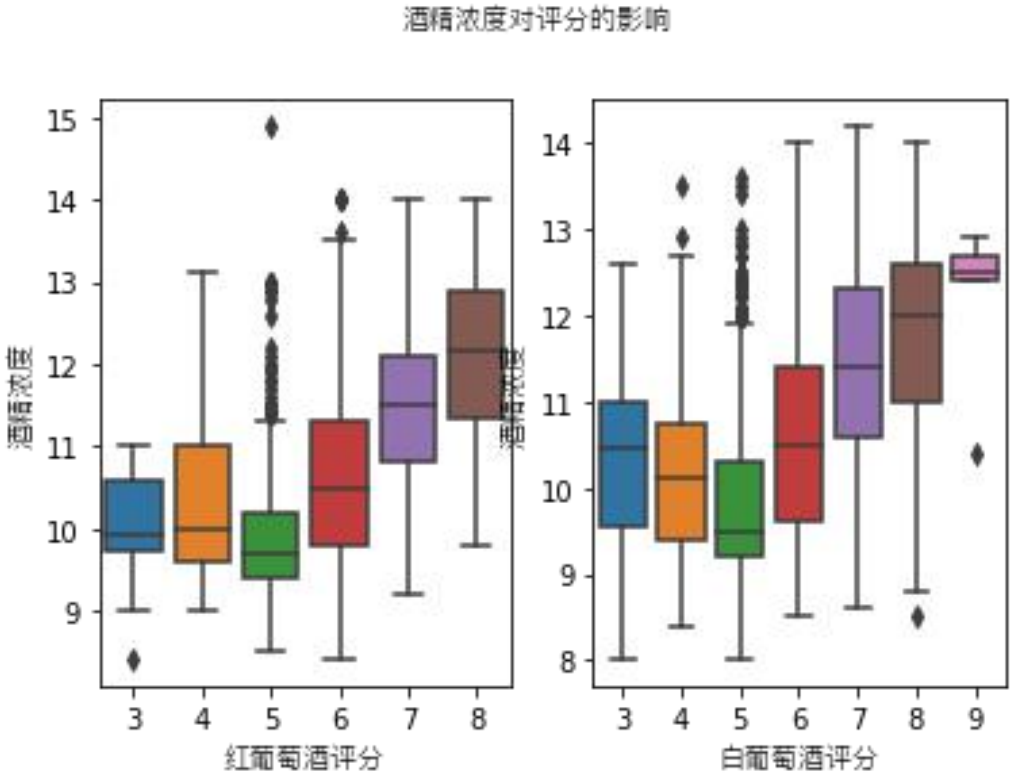


图 37 酒精浓度对评分影响图

可以发现，对于红白葡萄酒而言，酒精浓度的上升会带来一定的评分上涨趋势，较高的酒精浓度更有可能带来较高的评分。

按国标标准，葡萄酒酒精浓度应不低于 7.0

```
(dfr[dfr['alcohol'] < 7]).alcohol.count()
(dfw[dfw['alcohol'] < 7]).alcohol.count()
```

此处按这个标椎做一下检查，可见均没有不符标准的情况。

- 补图 37(结果截图)

4.3.2.3 附加指标分析

矿物盐

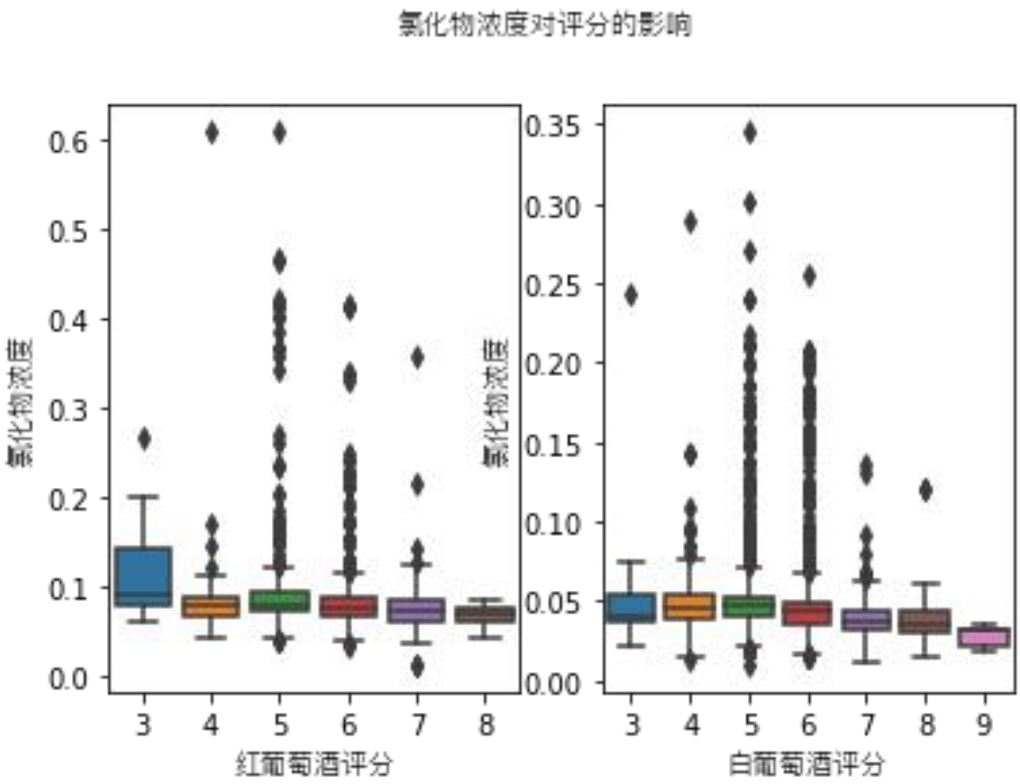


图 38 氯化物含量对评分影响图

可以发现有一个微弱的随浓度降低评分上升的趋势，总体来说还是不算显著。但若出现较高浓度，则很有可能该评分处于中间分段。

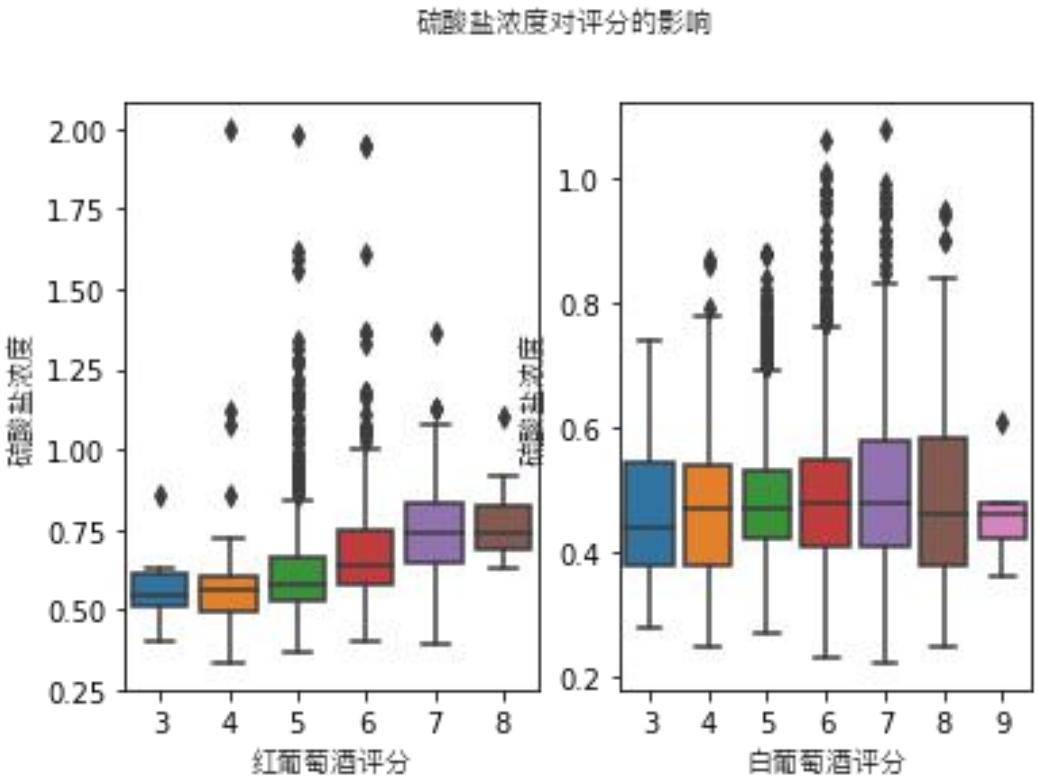


图 39 硫酸盐含量对评分影响图

可以发现硫酸盐浓度对红葡萄酒的评分有一个较明显影响趋势，随着浓度增加更有可能得到一个较高的评分。

但是不能从硫酸盐浓度有效的估计白葡萄酒的评分，但若出现较高浓度，则很有可能该评分处于中间分段。

二氧化硫

首先看一下游离二氧化硫的占比以及其的影响情况。

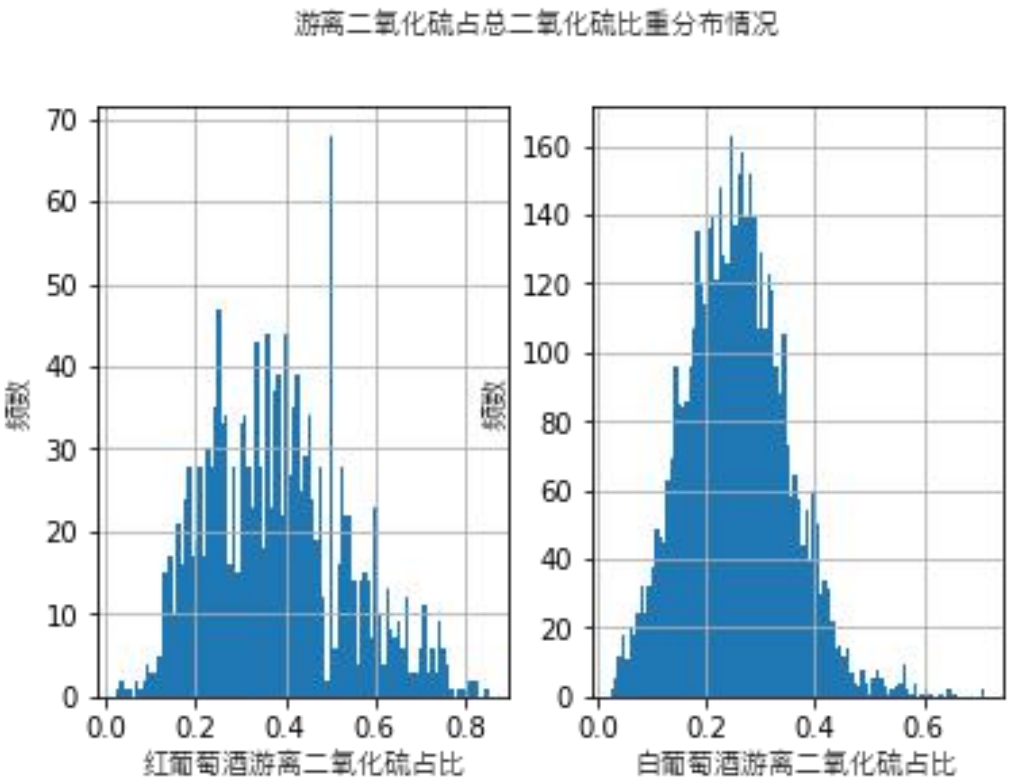


图 40 游离二氧化硫含量占比二氧化硫图

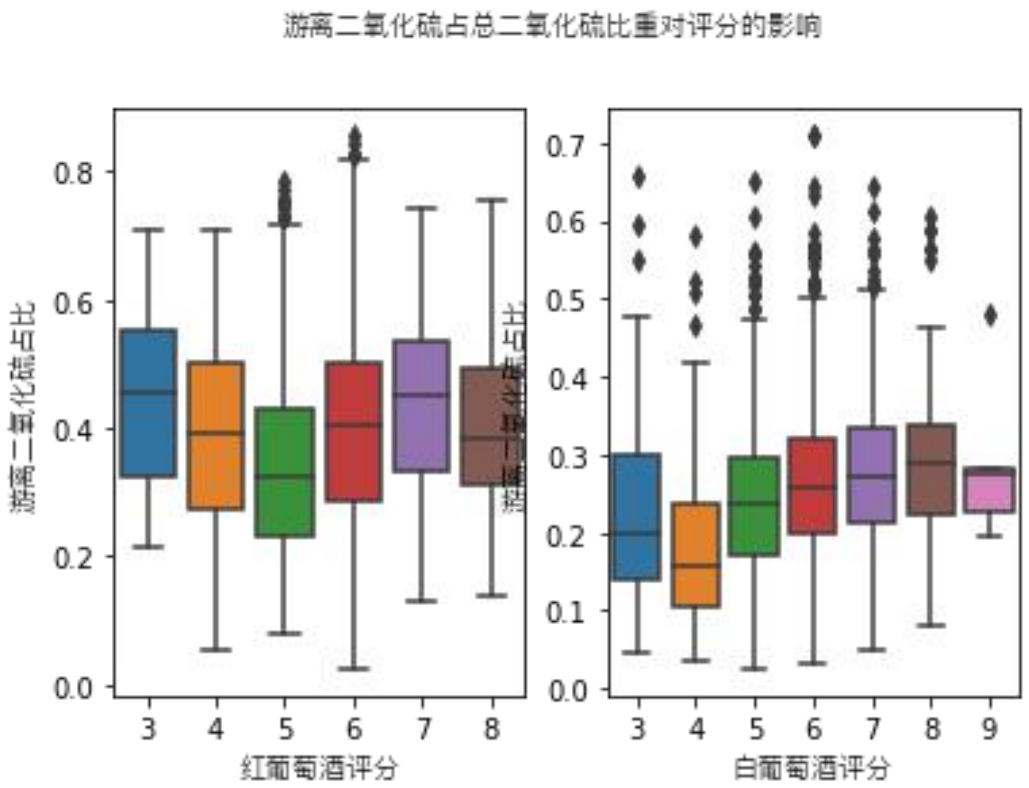


图 41 游离二氧化硫含量占比对评分影响图

可以发现，仅从游离二氧化硫占总二氧化硫的比重无法有效估计红、白葡萄酒的评分情况。

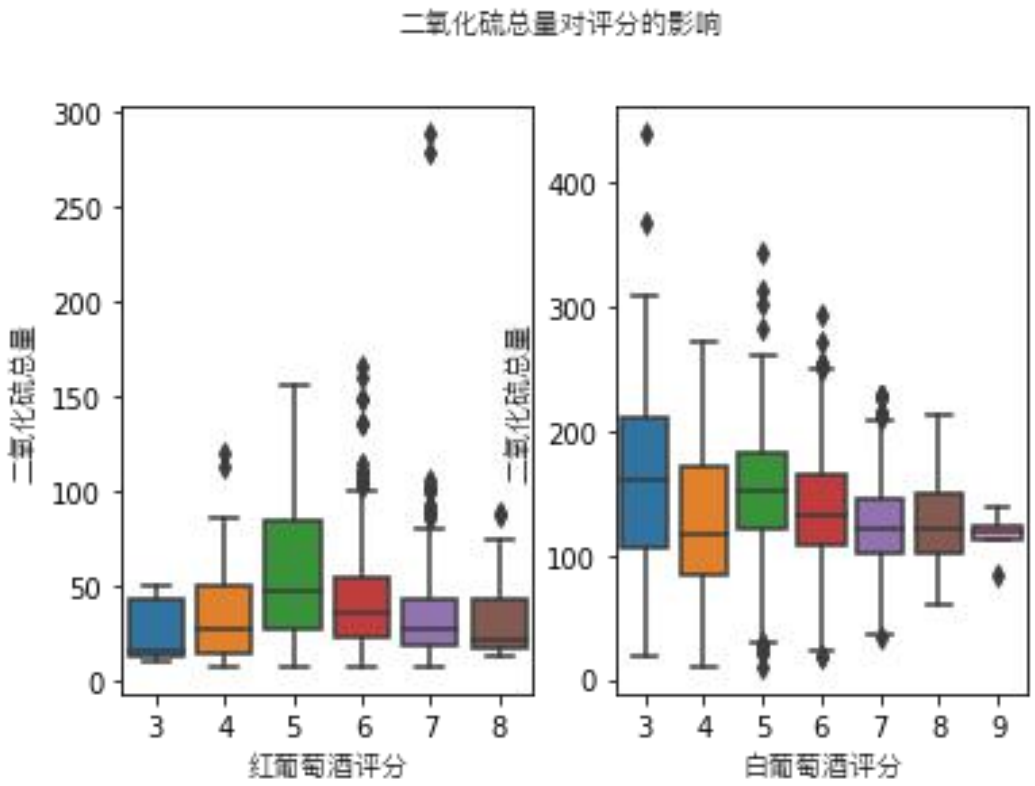


图 42 二氧化硫含量对评分影响图

可以发现，仅从二氧化硫的总浓度亦无法有效估计红、白葡萄酒的评分情况，不过若出现高浓度二氧化硫，则有更大可能获得中间段评分。

4.3.2.4 密度分析

除了之前在描述统计中看到的密度分布情况，这里可以进一步观察密度对评分的影响。

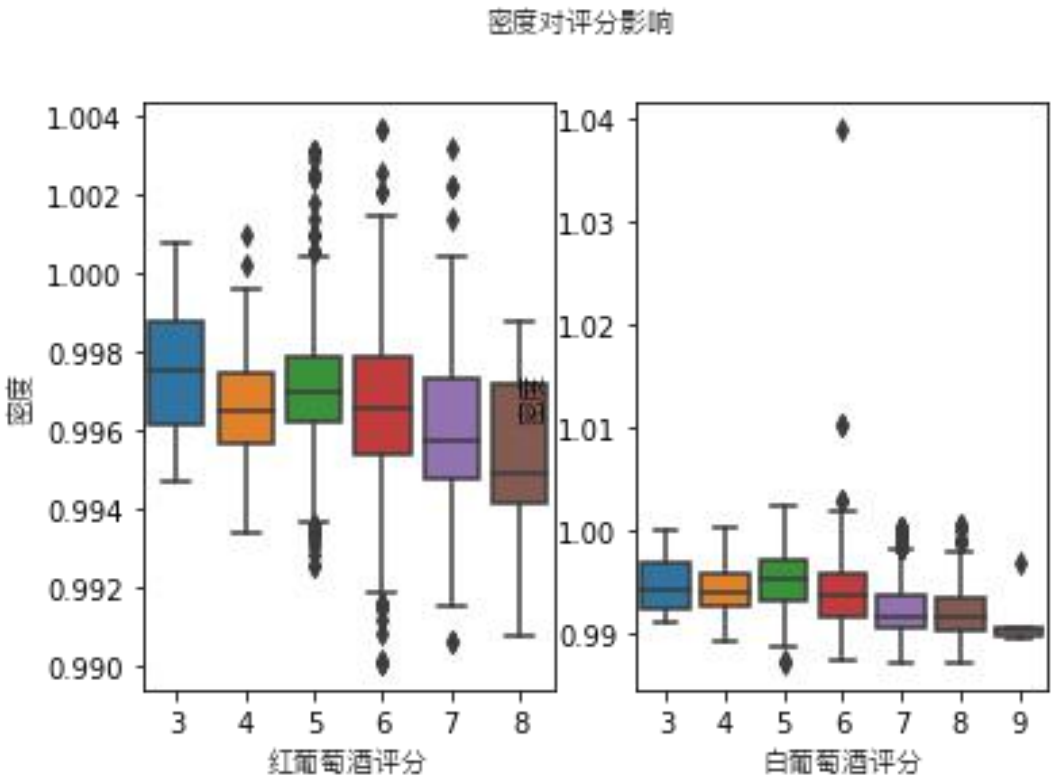


图 43 密度对评分影响图

可以发现，对于红葡萄酒而言，随着密度的降低会有一个轻微的取得高分的趋势，而对于白葡萄酒而言，密度的变化并不会影响其评分情况。

4.3.3 综合分析

4.3.3.1 各变量间的关系

在上一个章节中选择性的查看了大部分输入变量对评分的影响，此处可以进一步将所有输入变量对评分的影响放在一起，类似于总结，可以有一个更加直观的视角。

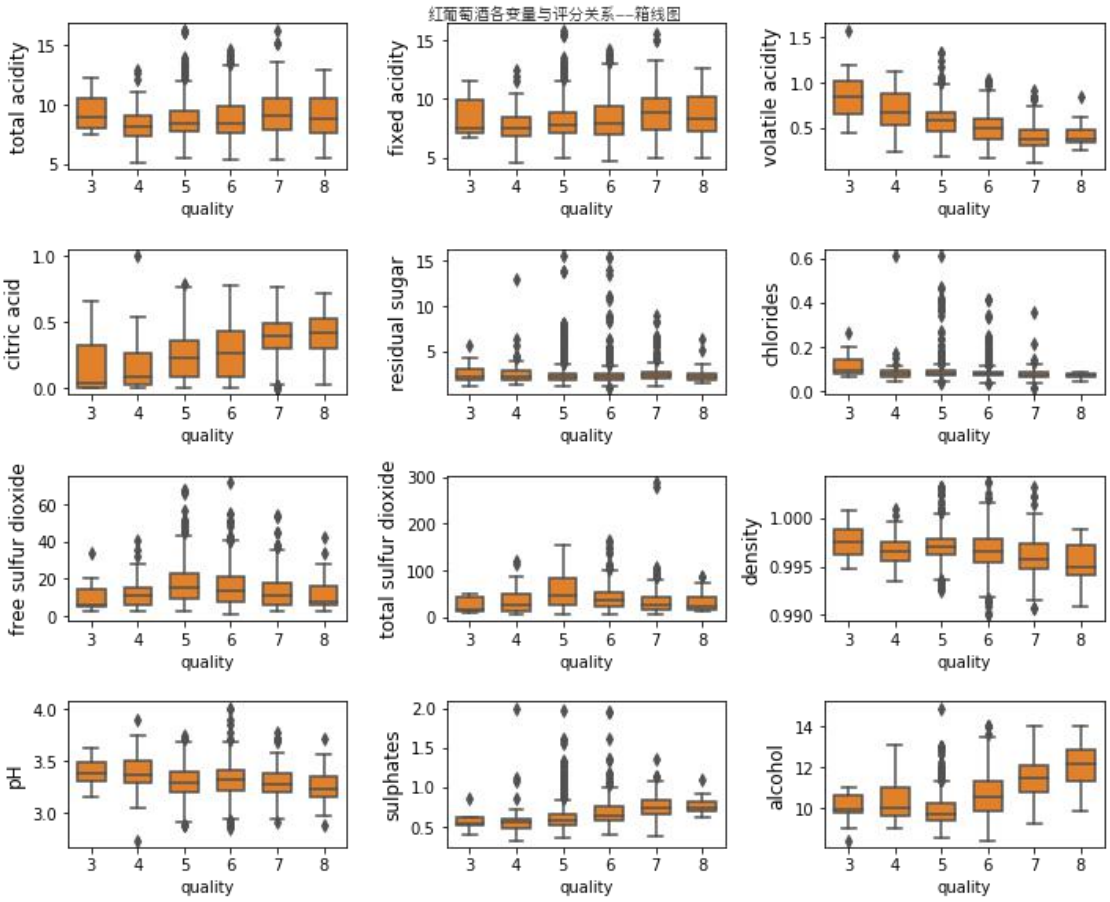


图 44 红葡各变量对评分影响图

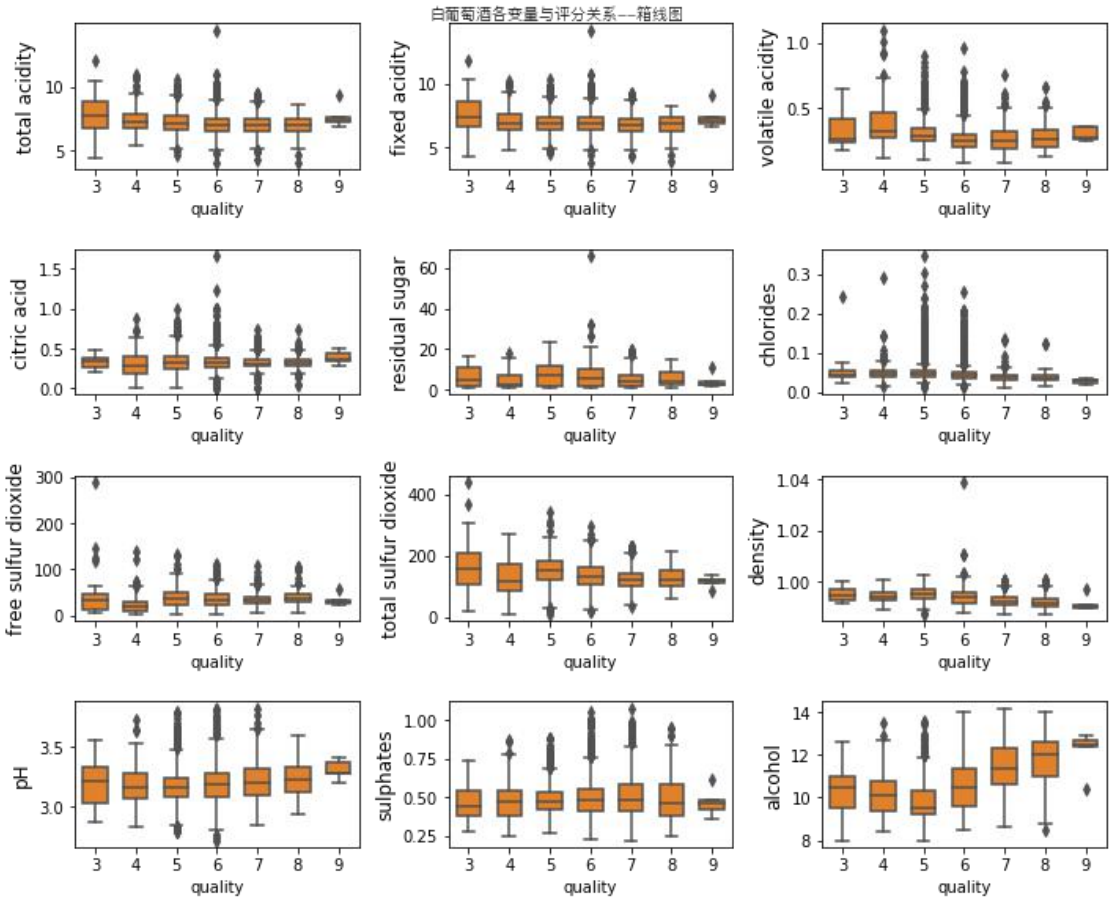


图 45 白葡各变量对评分影响图

综上可以较为清晰的看出各个变量对评分的影响。

接下来看一下红白葡萄酒的热力相关图。

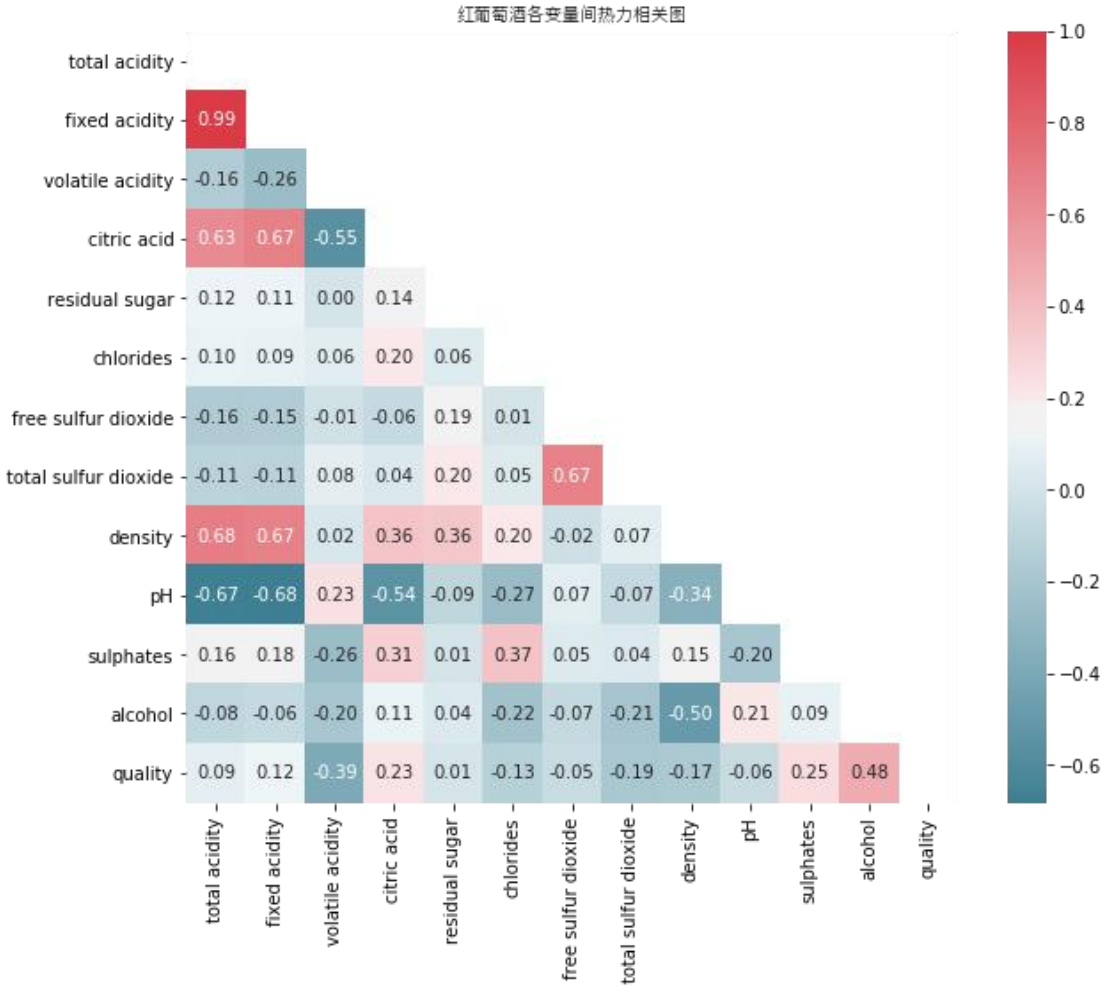


图 46 红葡热力相关图

可以发现，就红葡萄酒 quality 得分与各输入变量的相关关系而言，与之前的读图分析结果基本保持一致，只不过是以数值形式进行了量化，此处不复赘述。

而各输入变量之间，可以发现：

- 总酸与固定酸、柠檬酸、密度有较强正相关，与 pH 有较强负相关；
- 固定酸与柠檬酸、密度有较强正相关，与 pH 有较强负相关；
- 挥发酸与柠檬酸有较强负相关；
- 柠檬酸与 pH 有较强负相关；
- 游离二氧化硫与二氧化硫总量有较强正相关；
- 密度与酒精有较强负相关

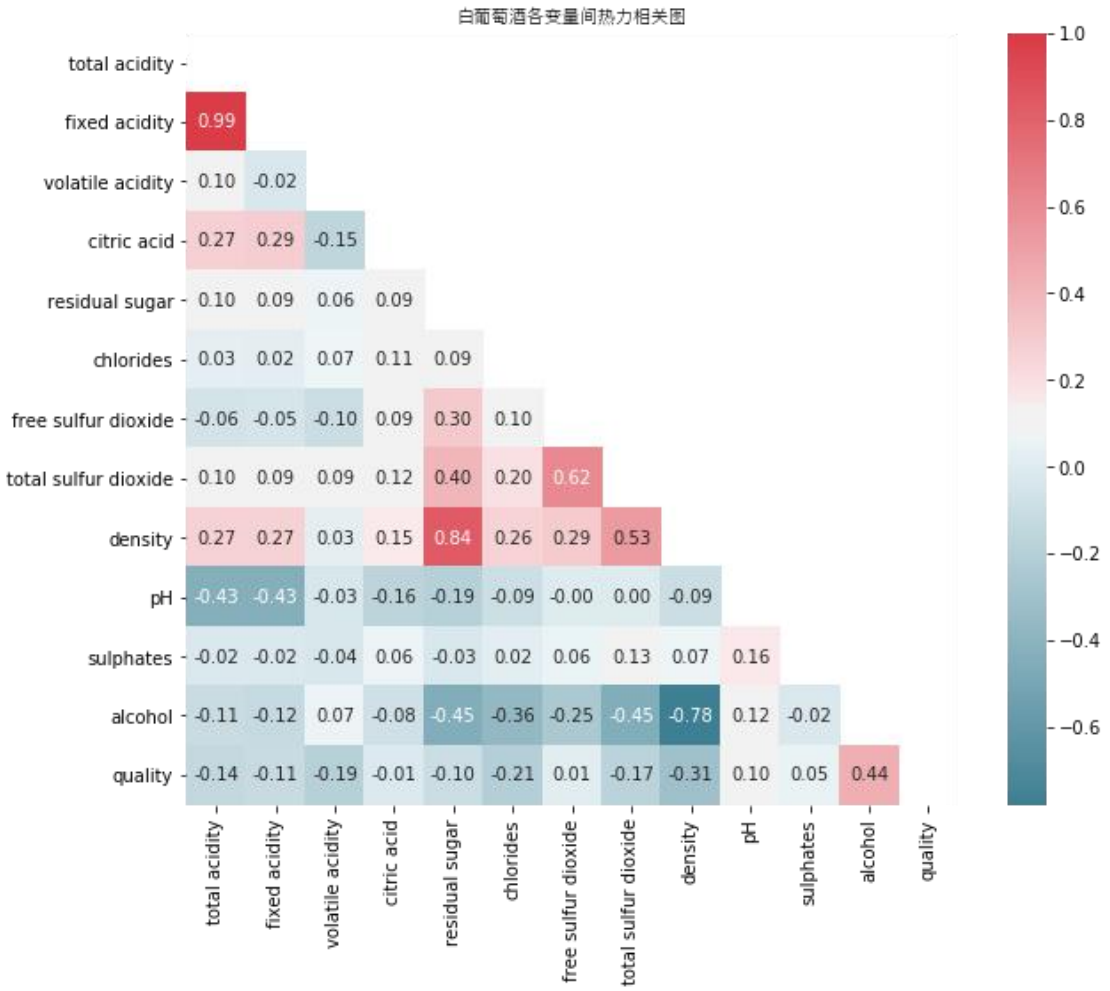


图 47 白葡热力相关图

对于白葡萄酒类似地，各输入变量之间可以发现：

- 总酸与固定酸有较强正相关，与 pH 有较弱负相关；
- 固定酸与 pH 有较弱负相关；
- 残留糖浓度与密度有较强正相关，与酒精浓度有较弱负相关；
- 游离二氧化硫与二氧化硫总量有较强正相关；
- 二氧化硫总量与密度有较强正相关，与酒精有较弱负相关；
- 密度与酒精有较强负相关

各种酸之间的相关、酸与 pH 的相关以及游离二氧化硫和总二氧化硫的相关关系都易于理解。而密度与酒精浓度的负相关可以进一步作图进行展示。

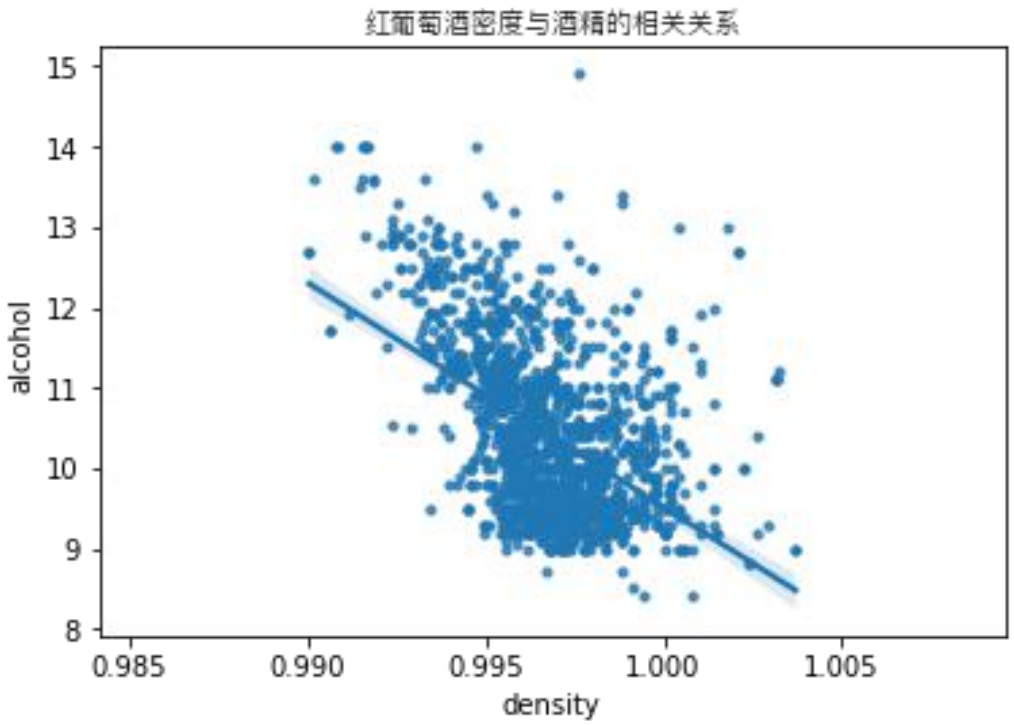


图 48 红葡密度与酒精相关图

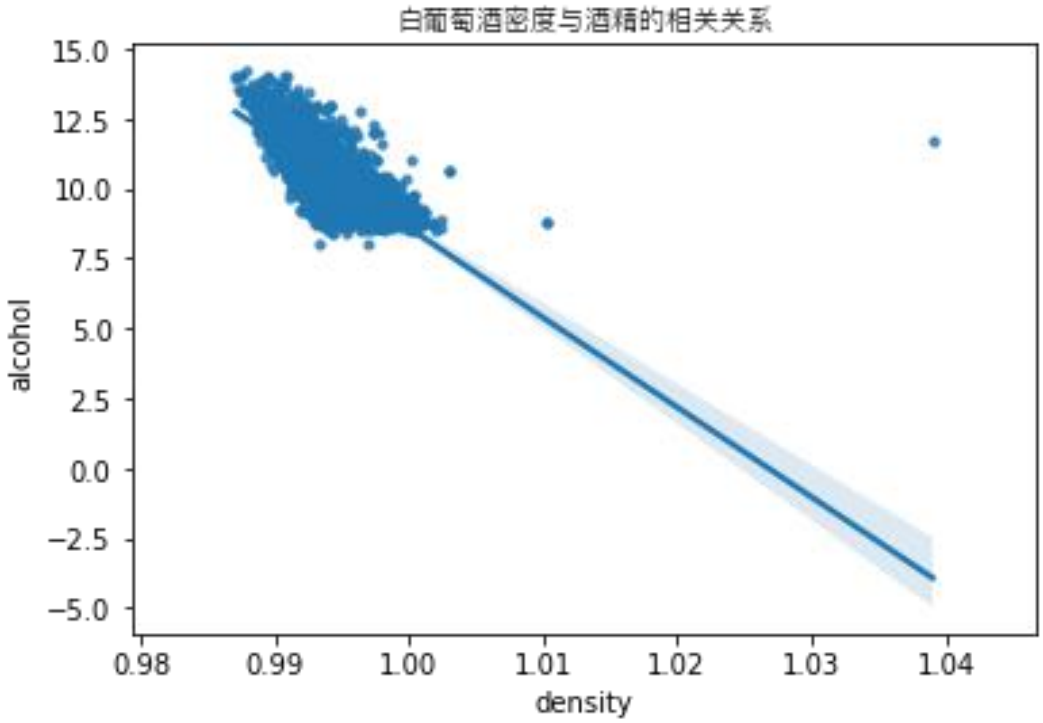


图 49 白葡密度与酒精相关图

同理可以制作残留糖与酒精的相关关系。

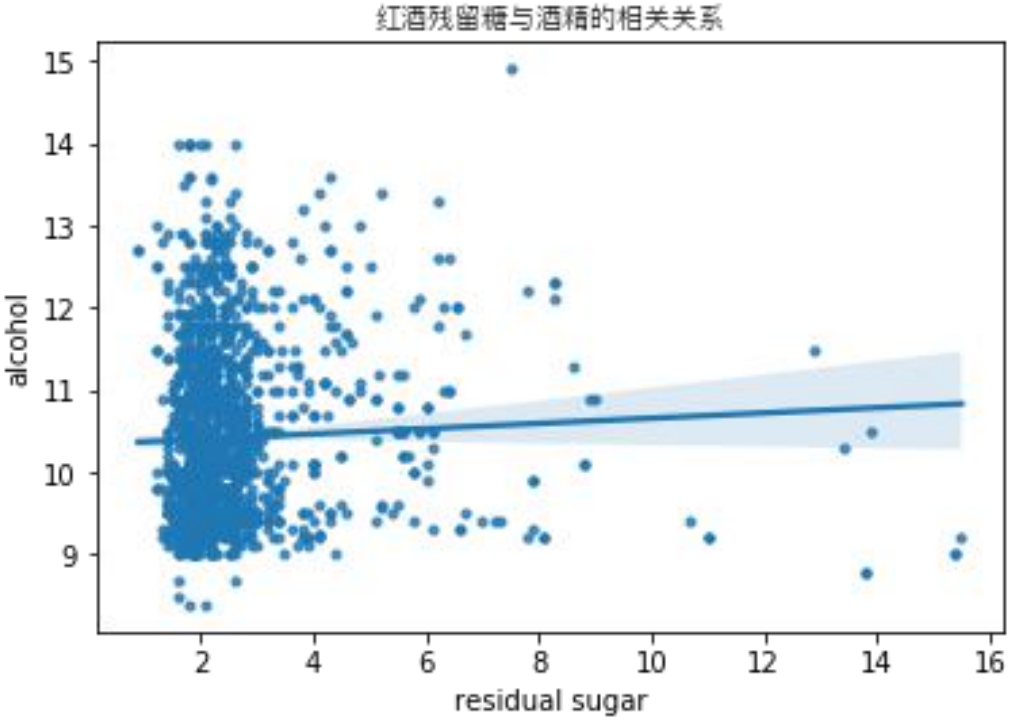


图 50 红葡残留糖与酒精相关图

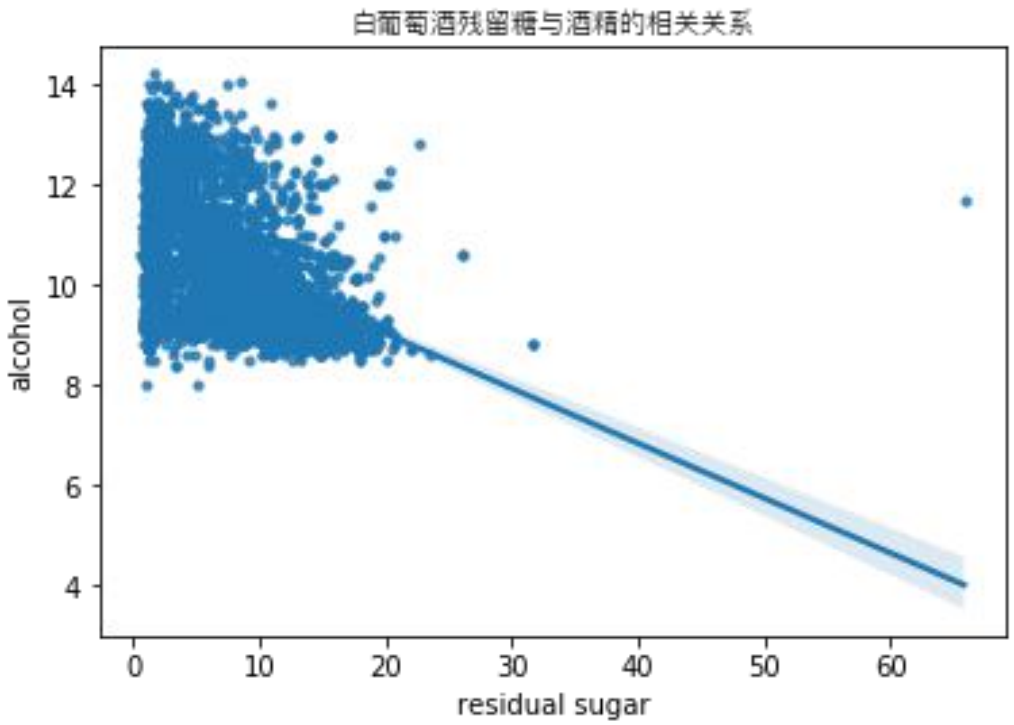


图 51 白葡残留糖与酒精相关图

4.3.3.2 主成分分析

前述分析中已经知道有些变量之间存在较强相关关系，这表示这些变量指代的含义可能出现重复，即存在浓缩变量的可能。

为验证想法，此处选择运用主成分分析（PCA）对输入变量进行降维，尝试找出具有代表性的少数几个新变量。因为各个变量之间度量单位不同且取值范围差异较大，故选择先将数据进行中心标准化处理，再对标准化的数据求解相关阵（也即协方差阵），进而求解特征值和特征向量。为避免高度重合，在计算之前要先将 total acidity 和 quality 这两列剔除出数据。

首先写出主题代码，封装成函数的形式。

```
#输入变量主成分分析
"""
参数:
```

```
    - XMat：传入的是一个 numpy 的矩阵格式，行表示样本数，列表示特征
    - k：表示取前 k 个特征值对应的特征向量
函数解释：
    - pca_mat()：获取参与运算的多维数组
    - pca_eig()：返回满足要求的前 k 个特征值和特征向量
    - pca_coe()：返回主成分系数
    - pca()：返回每个样本的主成分得分
    - pca_draw()：返回前两个主成分得分的散点图
"""
def pca_mat(x):
    temp = x.drop(['quality','total acidity'],axis=1) #获取去除评分项的数据表
    XMat = np.array(temp) #dataframe 格式转为多维数组
    average = np.mean(XMat,axis=0) #axis=0 表示按照列来求均值
    standard = np.std(XMat,axis=0) #求每列标准差
    data_adjust = (XMat - average)/standard #中心标准化
    return data_adjust


def pca_eig(data_adjust):
    covmat = np.cov(data_adjust, rowvar=0)  #计算协方差矩阵
    eigVals,eigVects = np.linalg.eig(covmat) #求解协方差矩阵的特征值和特征向量
    eigValInd = np.argsort(-eigVals) #按照 eigVals 进行从大到小排序（给出序号，不修改原特征值列表）
    """确定前 k 的主成分，使选取的主成分贡献 90%以上的方差"""
    val_sum = 0
    val_total = eigVals.sum()
    for k in eigValInd:
        val_sum += eigVals[k]
        if val_sum/val_total < 0.90:
            continue
        else:
```

```
            break
    """分割线"""
    x = int(np.argwhere(eigValInd==k)+1) #定位 k 所在位置，结果加 1
    eigValInd = eigValInd[:x:1] #截取前 k 个特征值的序号
    """取前 k 特征值"""
    list = []
    for i in eigValInd:
        list.append(eigVals[i])
    redEigVals = np.array(list)
    """对应前 k 的特征向量"""
    redEigVects = []
    for i in eigValInd:
        redEigVects.append(eigVects[i])
    redEigVects = np.array(redEigVects).T
    return redEigVals, redEigVects, eigVals, eigVects


def pca_coe(data_adjust):
    return pca_eig(data_adjust)[1]/(pca_eig(data_adjust)[0]**0.5)


def pca(data_adjust):
    lowDDataMat = np.matrix(data_adjust) * pca_eig(data_adjust)[1]
    return lowDDataMat


def pca_draw(data_adjust):
    df = pd.DataFrame(pca(data_adjust))
    plt.scatter(x=df[0], y=df[1])
    if data_adjust.sum() - pca_mat(dfr).sum() == 0:
        i = '红葡萄酒'
    else:
        i = '白葡萄酒'
```

```
    plt.title(f'{i}'+'主成分得分--散点图')
    plt.xlabel('第一主成分', fontsize=12)
    plt.ylabel('第二主成分', fontsize=12)
    plt.show()

"""红、白葡萄酒初始分析数据"""
pr = pca_mat(dfr)
pw = pca_mat(dfw)
```

在进行主成分分析的时候，有两点需要注意：

1、变量间不能高度线性相关（故已提前提出明确已知的 total acidity），特征值应不要出现十分接近 0 的情况，若存在则说明变量中存在严重的多重共线性，一定存在某些变量之间高度相关，此时的主成分分析效果将不是很理想。

2、如果各变量之间相关性不大，主成分分析也不会获得理想的效果。

对于第一点，经检查可以认为不存在特征值十分接近 0 的情况：

```
In [9]: # 主成分分析
        pca_eig(pr)[2]

Out[9]: array([ 3.88070825e+00,  1.93782510e+00,  1.59121221e+00,  1.26119209e+00,
                9.62744293e-01, -2.51040967e-16,  7.44436784e-02,  6.93452298e-01,
                6.36767526e-01,  1.86870832e-01,  4.31335890e-01,  3.50957225e-01])

In [10]: pca_eig(pw)[2]

Out[10]: array([3.36570674e+00, 2.22131017e+00, 1.64244188e-15, 2.09602992e-02,
                2.92254807e-01, 4.27978385e-01, 1.22380645e+00, 6.87381846e-01,
                7.54085539e-01, 1.05187249e+00, 9.40654582e-01, 1.01643918e+00])
```

对于第二点，则需要看看在主成分覆盖 90%以上方差的原则下，选取了多少个主成分，若降维效果理想则少数的 2-3 个主成分即可满足要求。然而如下图所示，在 90%标准下，红葡萄酒选出了 7 个特征值即对应 7 个主成分，同理白葡萄酒选出了 8 个主成分，可见两者的主成分降维效果都不是很理想。

```
In [11]: pca_eig(pr)[0]
Out[11]: array([3.88070825, 1.9378251 , 1.59121221, 1.26119209, 0.96274429,
                0.6934523 , 0.63676753])

In [12]: pca_eig(pw)[0]
Out[12]: array([3.36570674, 2.22131017, 1.22380645, 1.05187249, 1.01643918,
                0.94065458, 0.75408554, 0.68738185])
```

结合之前的热力相关图就可以理解，因为大部分变量之间的相关性都很低，具有较强相关的只有少数几个变量，结合上述主成分分析需注意的第二点就知道效果是肯定不会很理想的。

因为效果不理想，其实分析到这里就可以停止了。下面仅作展示将代码执行完成的结果：

- 主成分系数(解释为第 K 个主成分表示为 11 个输入变量的线性组合。可见很难清晰的描述除各主成分代表的含义)

```
In [13]: """主成分系数 解释为第K个主成分表示为11个输入变量的线性组合。可见很难清晰的描述除各主成分代表的含义。"""
         pd.DataFrame(pca_coe(pr))
```

Out[13]:

	0	1	2	3	4	5	6
0	0.237	0.341	-0.143	3.564e-01	1.138e-01	-1.529e-02	4.582e-01
1	0.033	0.063	-0.181	9.074e-02	-2.901e-01	-6.960e-01	-3.071e-01
2	-0.073	-0.064	-0.397	2.689e-01	8.429e-02	3.217e-01	-3.852e-01
3	-0.084	-0.121	0.055	-3.598e-02	-3.280e-01	-1.371e-01	-8.778e-02
4	0.024	0.048	-0.152	3.541e-02	-7.740e-01	2.805e-01	-2.304e-01
5	0.354	-0.512	-0.058	5.237e-17	-1.054e-15	-6.359e-16	1.934e-15
6	0.172	0.240	-0.021	-4.069e-02	2.222e-01	7.922e-02	-8.139e-01
7	-0.130	-0.133	-0.503	7.406e-02	1.672e-01	-1.892e-01	3.445e-01
8	0.080	0.101	0.103	-1.469e-01	-2.748e-01	-2.693e-02	4.199e-01
9	-0.069	-0.125	0.315	5.634e-01	5.834e-02	-4.785e-01	-1.987e-01
10	-0.050	-0.071	0.022	-2.845e-01	2.609e-01	8.009e-02	-2.398e-01
11	-0.048	-0.081	0.157	3.946e-01	-5.997e-02	6.915e-01	5.043e-02

```
In [14]: pd.DataFrame(pca_coe(pw))
```

Out[14]:

	0	1	2	3	4	5	6	7
0	0.146	0.180	2.273e-01	-1.818e-01	-3.996e-01	3.107e-02	5.609e-01	4.426e-01
1	-0.296	-0.365	2.583e-01	3.489e-01	-1.795e-01	1.070e-01	1.652e-01	2.996e-01
2	0.385	-0.472	2.636e-16	-1.253e-15	3.440e-15	-5.393e-16	8.629e-15	-6.964e-16
3	0.047	0.057	-2.866e-02	1.378e-01	-3.580e-01	4.351e-02	-8.817e-01	4.378e-02
4	0.032	0.022	5.109e-01	8.882e-02	-2.968e-01	6.393e-02	6.267e-02	-8.589e-01
5	-0.056	-0.082	-1.672e-01	-2.072e-01	-6.191e-01	-1.350e-02	-7.852e-02	2.765e-01
6	0.017	0.065	2.713e-01	1.738e-01	1.140e-01	4.838e-01	-1.517e-01	1.824e-01
7	-0.101	-0.120	8.084e-02	-7.669e-01	7.385e-02	4.830e-01	-1.210e-01	-1.260e-01
8	0.101	0.153	1.513e-01	3.232e-02	-2.141e-01	6.595e-03	-8.335e-02	1.376e-01
9	0.110	0.085	4.904e-02	2.219e-01	1.912e-01	5.831e-01	-3.854e-02	3.217e-01
10	-0.011	-0.034	4.586e-01	-2.304e-01	2.015e-01	-4.010e-01	-3.880e-01	3.840e-01
11	-0.023	-0.033	-3.052e-01	6.586e-02	-2.255e-01	2.764e-01	-9.165e-03	-1.845e-01

- 主成分得分(解释为每个样本点在主成分上投影的坐标)

In [15]:
```
"""主成分得分 解释为每个样本点在主成分上投影的坐标。"""
pd.DataFrame(pca(pr))
```

Out[15]:

	0	1	2	3	4	5	6
0	-0.339	-2.650e-02	3.575e-01	5.717e-01	6.405e-01	-0.466	1.077e-01
1	0.320	1.881e-01	-1.834e+00	-1.662e-01	9.628e-01	0.788	-7.978e-01
2	0.010	-2.486e-02	-9.822e-01	-1.177e-01	6.700e-01	0.366	-7.253e-02
3	0.844	1.244e+00	-4.479e-01	-5.204e-01	-7.743e-01	-1.675	1.050e+00
4	-0.339	-2.650e-02	3.575e-01	5.717e-01	6.405e-01	-0.466	1.077e-01
5	-0.318	2.267e-02	3.682e-01	4.984e-01	7.449e-01	-0.558	1.068e-01
6	-0.235	3.767e-01	-5.439e-01	1.183e-03	7.805e-01	-0.445	3.530e-01
7	-0.462	1.719e-01	4.455e-01	5.833e-01	1.263e+00	0.102	-4.111e-01
8	-0.116	2.679e-01	4.878e-01	-7.931e-02	4.265e-01	-0.255	2.568e-01
9	-0.870	-2.561e-01	-1.015e+00	-2.446e-01	-1.626e+00	0.501	-1.312e-01
10	-0.293	-5.202e-01	-4.911e-01	-6.451e-01	9.726e-01	-0.093	1.451e-01
11	-0.870	-2.561e-01	-1.015e+00	-2.446e-01	-1.626e+00	0.501	-1.312e-01

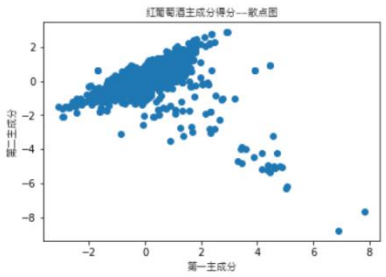
In [16]:
```
pd.DataFrame(pca(pw))
```

Out[16]:

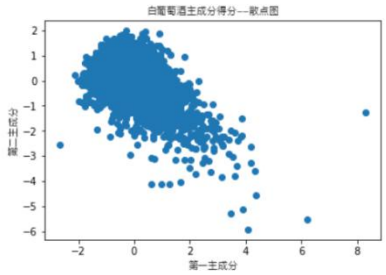
	0	1	2	3	4	5	6	7
0	2.037e-01	5.051e-01	2.538e+00	-3.760e-01	-1.373e+00	-1.046e-01	-5.554e-02	-1.864e+00
1	4.462e-01	2.904e-02	-9.807e-01	-2.222e-01	7.227e-01	-6.068e-01	-3.361e-01	4.633e-01
2	-6.250e-03	-6.937e-02	5.925e-01	1.209e+00	-1.313e+00	-2.769e-02	7.557e-01	1.034e+00
3	-5.599e-01	1.946e-01	5.045e-01	-6.035e-01	-6.420e-01	1.096e+00	3.261e-01	-6.619e-02
4	-5.599e-01	1.946e-01	5.045e-01	-6.035e-01	-6.420e-01	1.096e+00	3.261e-01	-6.619e-02
5	-6.250e-03	-6.937e-02	5.925e-01	1.209e+00	-1.313e+00	-2.769e-02	7.557e-01	1.034e+00
6	4.860e-01	-9.297e-02	-1.738e-01	-3.517e-01	9.576e-01	-4.883e-01	7.396e-01	-5.684e-01
7	2.037e-01	5.051e-01	2.538e+00	-3.760e-01	-1.373e+00	-1.046e-01	-5.554e-02	-1.864e+00
8	4.462e-01	2.904e-02	-9.807e-01	-2.222e-01	7.227e-01	-6.068e-01	-3.361e-01	4.633e-01
9	-7.682e-01	2.993e-02	-2.418e-01	5.594e-01	-9.473e-01	2.251e-01	4.335e-01	1.411e+00
10	-6.545e-01	-5.852e-01	-6.089e-01	1.122e+00	-8.908e-01	-1.996e+00	7.098e-01	9.914e-01
11	-7.604e-01	-1.142e-01	9.616e-01	6.676e-01	-1.004e+00	-1.045e+00	9.613e-01	1.526e+00

- 前两个主成分散点图

In [17]:
```
"""主成分得分散点图"""
pca_draw(pr)
```



In [18]:
```
pca_draw(pw)
```



4.3.4 数据总结

整体而言，11 种输入变量对红葡萄酒的品质评分产生较多的影响，而对白葡萄酒则是在大多情况下无显著趋势，故在此猜测对白葡萄酒评分产生重要影响的另有因素，未在此次实验中被测量。

1、从实验结果来看，红、白葡萄酒的对输入变量的反应可以总结为以下表格

变量指标	红葡萄酒	白葡萄酒
固定酸占总酸比分布情况	分布分散，多数>=88%	单峰分布，集中在 97%附近
固定酸占比对评分影响	高占比易得高分	无显著趋势
柠檬酸占固定酸比分布情况	分散于 0-8%，有很多 0 值	单峰分布，集中在 4%附近
柠檬酸占比对评分影响	高占比易得高分	无显著趋势
挥发酸占总酸比分布情况	大多数分散于 2.5%-12%	单峰分布，集中在 3%附近
挥发酸占比对评分影响	低占比易得高分	无显著趋势
总酸对评分影响	无显著趋势	无显著趋势
pH 对评分影响	低 pH 得高分，趋势微弱	高 pH 得高分，趋势微弱
按柠檬酸分类	属于（干、半干、半甜）类	99%属于（干、半干、半甜）类，只有 2 例甜葡萄酒
残留糖对评分影响	无显著趋势，高含量在中间分段	无显著趋势，高含量在中间分段
按残留糖分类	干、半干、半甜	干、半干、半甜、甜
酒精浓度对评分影响	高浓度易得高分	高浓度易得高分
氯化物对评分影响	低浓度得高分，趋势微弱	低浓度得高分，趋势微弱
硫酸盐对评分影响	高浓度易得高分	无显著趋势

游离二氧化硫占总量比分布情况	分布分散大多数在10%-60%	单峰分布，集中在 25%附近
游离二氧化硫占比对评分影响	无显著趋势	无显著趋势
二氧化硫总量对评分影响	无显著趋势	无显著趋势
密度对评分影响	低密度易得高分	低密度易得高分

2、关于上表的总结如下

- 红葡萄酒品质主要与固定酸含量（柠檬酸）、酒精浓度、硫酸盐浓度正相关，与挥发酸含量、pH 值、氯化物浓度、密度负相关；
- 白葡萄酒品质主要与 pH 值、酒精浓度正相关，与氯化物、密度负相关；
- 对两种葡萄酒而言，总酸含量、残留糖含量、二氧化硫都是没有什么影响力的变量

3、从变量样本分布情况来看，两种葡萄酒存在明显区别，故在成分含量上对红、白葡萄酒进行区分较为可行。

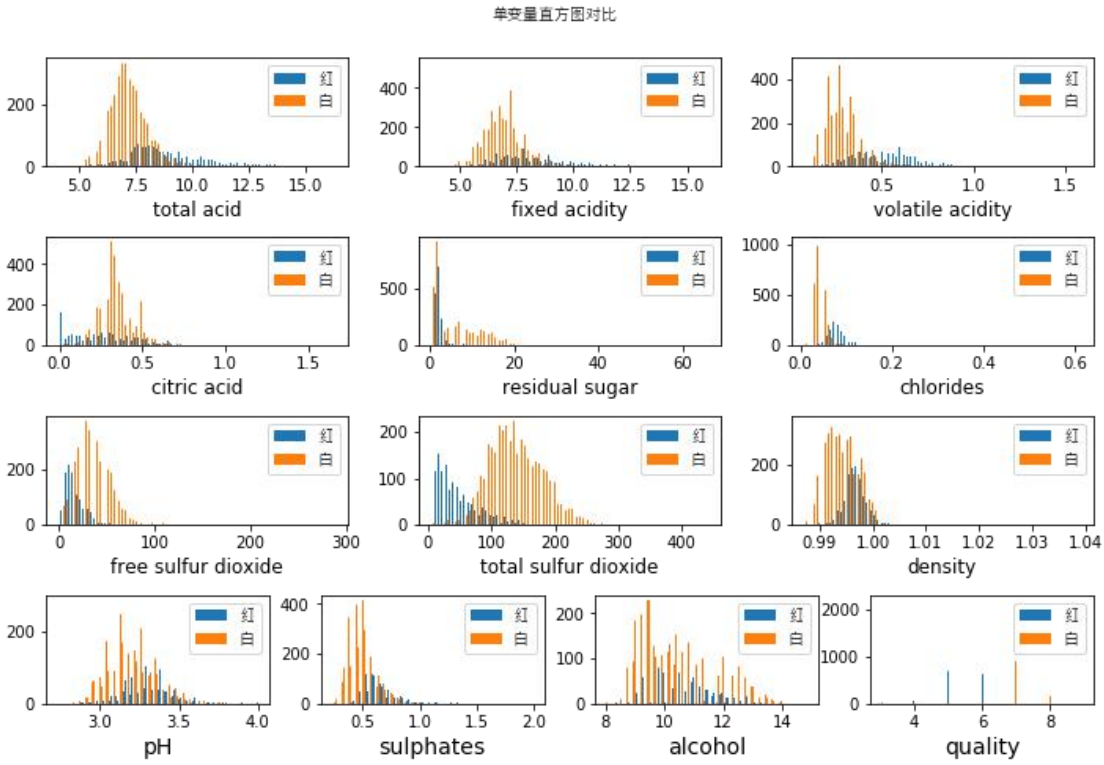


图 57 红白葡萄酒单变量直方图

4、11 种输入变量之间普遍相关性不大，仅少数具有较高相关性，因此不具备理想的降维条件，无法将 11 种输入变量整合为少数几个综合变量（对红、白葡萄酒均是如此）。

五 总结

以上，我们用了一些篇幅介绍了我们的工作，主要可以总结为以下几点：

- 发掘了 Python 的多种用途，多方面实现了数据可视化
- 将课堂所学运用其中，并进行了一定程度的拓展
- 借鉴博客，GitHub 仓库以及 Python 数据分析书籍上分析数据的方法，从一开始的模仿到后面能够结合自己的想法自己进行一些分析与扩展
- 分析一个整体的案例

通过这次的作业，我们第一次认识到了该怎样去分析数据，去得出结论，以及如何再对得出的结论进行验证。我们有许许多多的方法对数据进行分析，但随着数据量的加大，我们开始求助于计算机来解决。对 Python 的合理应用让处理超大数据集变成了可能，使我们不再受拘于数据整理，而是专注方法的思考和完善。因此想做一个数据科学工作者，既要求对编程技术熟练掌握，也需要数学知识的深厚功底。

我们也逐渐意识到，现实生活中的数据集，往往都不是那么“干净”的，处理数据很重要的一步，是把这些数据集转换成计算机能处理的数据，比如转换成 CSV 或者 Json 格式。而根据我们找到的资料来看，即使是一位优秀的数据分析行业人员，转换数据的工作对他而言也是非常有挑战性的。

这次写论文的过程，也是锻炼我们自学能力的一个过程。面对从网络上获取的大量数据，选用怎样的可视化方法，我们一开始也是毫无头绪的。我们购买了相关的数据分析书籍（均在参考资料里列出），又搜索了海量的数据分析的视频，博客，以及 GitHub 上的代码。我们从一开始模仿书本上的分析方法，到最后能对类似数据集做自己的分析，拓展。我们的编程能力与分析能力都在提升，同时也了解到了很多从未接触过的图，比如热力相关图等。并且自己亲手实践了课堂上讲过的 PCA 降维等方法，加深了自己的理解。

六 参考资料

《Python 机器学习基础教程》 Andreas C. M ü ller，Sarah Guido 著，张亮译，人民邮电出版社

《Python 数据分析基础》 Clinton W. Brownley 著，陈光欣译，人民邮电出版社

《Python 编程从入门到实践》Eric Matthes 著 袁忠国译

介绍数据分析方法，Python Matplotlib 库的国内外博客、教学视频

所用数据集源自 Python 里 Sklearn 自带的数据集，ics 机器学习数据集，以及相关网站。

本项目所有代码和资料均在 github 开源，仓库地址：

https://github.com/JHSUYU/final_python.git