基于多元统计分析对古代玻璃成分分析与鉴别

摘　要

古玻璃化学成分复杂，种类多样，风化过程中与外界环境的交换使其化学成分发生改变。本文综合运用**多元统计方法**及**机器学习**方法，结合风化状态对古玻璃的化学成分进行研究，并对玻璃种类进行鉴别和亚类划分。

针对问题一，本文首先运用**逻辑回归**分析古玻璃风化与否与颜色、纹饰及类型间的相关性，得出铅钡玻璃与纹饰为 B 的玻璃更容易风化。其次，为研究未风化、风化、严重风化与化学成分含量的关系，本文使用 **Lasso 回归**筛选出与风化程度相关的成分作为多元逻辑回归模型的输入，根据回归系数进一步分析成分含量的相关性强弱。然后结合**描述性统计**与**可视化图表**，探索得到风化前后化学成分差异规律，最后，本文对还原了风化文物风化前的成分比例，求解出风化前各个化学成分比例的上下界。

针对问题二，首先，本文建立**改进的 CART 决策树**模型，得到基于氧化铅含量的玻璃的分类规则，分类效果极佳；其次，我们利用 SPSS 进行**组间距离法**的**系统聚类**对两类玻璃进行亚类划分，并根据亚类间化学成分差异及文献资料，分别命名为：铅钡组：(I) PbO(~ 25*wt*%)-BaO-SiO_2 (II) PbO(~ 40*wt*%)-BaO-SiO_2 (III) PbO-BaO-SiO_2；高钾组：(I) K_2O-CaO(~7wt%)-SiO_2 (II) K_2O-SiO_2。经比对，本文的分类与文献中的亚类划分重合度高，模型合理准确；最后，我们运用**主成分分析法**对 14 个氧化物含量特征降至二维**绘制亚类分类结果图**，各亚类样本点边界清晰，中心距离较大，亚类划分结果**敏感性高**。

针对问题三，本文分别使用逻辑回归分类模型及 CART 决策树模型使用相同的训练集进行训练，对未知的 8 个文物鉴别得到了相同的分类结果，且在测试集上的准确率均达到 100%，说明分类准确度很高，预测得到文物 **A1、A6、A7 为高钾玻璃**，其余为铅钡玻璃。通过绘制 **ROC 曲线**研究决策树与逻辑回归的敏感性，且针对逻辑回归模型提出了一种基于**偏导数**研究化学成分的微小变化对玻璃文物分类结果影响的**灵敏度分析**模型。

针对问题四，本文首先运用 **spearman 相关系数**对化学成分两两关系进行分析，得出铅钡玻璃中氧化铝和氧化镁的相关性系数最大，为 0.674，高钾玻璃氧化铁和氧化铝关联度最大，达到 0.74。再运用 ***VIF* 方差膨胀因子**和**因子分析法**对化学成分的相关性及两类玻璃关联性的差异性进行探究，得出高钾玻璃中化学成分关联关系最紧密的是 K_2O、CaO、Al_2O_3、Fe_2O_3、P_2O_5、SrO，无关成分为 SnO_2，铅钡玻璃最相关的成分为 PbO、BaO、CaO、CuO、SO_2、SrO，无关成分为 Na_2O、K_2O、MgO、Fe_2O_3、SnO_2。

本文模型紧扣风化状态 0-1 分布特征，逻辑回归二分类思想在四个问题中相互贯通，回归结果相互印证，模型整体性强；问题一的 Lasso 特征选择模型突破传统多元回归的样本需求大、多重共线性等限制，更客观准确，富有创新性；本文将得到的多元统计规律与文献中古玻璃成分分析相验证，模型准确合理。本文一题多法，同时采用逻辑回归、决策树两个方法进行类别鉴定，采用 spearman 相关系数分析与 VIF 分析、因子分析三种方法探究化学成分相互关系，有利于模型最优。

关键词：逻辑回归　特征选择　系统聚类　改进的决策树

一、 问题重述

1.1 问题背景

丝绸之路在古代中外文化和技术交流中起到重要作用，其中玻璃是中外贸易的主要商品之一。我国吸收了西亚地区制造玻璃技术，产出了许多与外来制品外观相似的玻璃。因为制作过程中添加的助熔剂不同，玻璃制品中的化学成分成为了对其类别的重要判断依据。同时，玻璃制品受埋藏环境的变化发生风化腐蚀的过程中与外界产生物质交换，制品的化学成分发生改变，对类别判断产生较大影响。而正确判断玻璃制品的类别有助于分析文物的年代和制作手法，对研究我国古代玻璃物品和制造技术的发展具有重要意义。

1.2 问题重述

表单 1 给出了 58 种文物相应的纹饰、颜色、表面有无风化极其对应类型。

表单 2 给出了 58 种文物的化学成分比例。

表单 3 给出了未分类的 8 种文物化学成分比例。

- 问题一有三个任务：
  1. 根据表单一，研究玻璃风化与否与玻璃类型、纹饰及颜色三个特征的关系。
  2. 根据表单二，针对两类不同的玻璃，分析风化程度与玻璃的化学组成成分比例之间的数据规律。
  3. 运用上述规律对表单二中风化点进行逆推得出风化前的化学成分数据。
- 问题二有两个任务：
  1. 根据所有特征数据，给出两类玻璃的分类规律。
  2. 对于两大类别，依据化学成分这一指标再细分为亚类，给出具体分类标准及结果，并对分类结果进行合理性解释及灵敏度分析。
- 问题三：

  对附件三的未知类别文物进行分类，并分析结果敏感度。
- 问题四：

  分别研究两类玻璃化学成分间的相互关系，并分析其差异性。

二、 问题分析

2.1 问题一的分析

为了研究玻璃文物表面风化与否与类型、纹饰及颜色的相关性，我们首先将这些指标量化为 0-1 变量。通过查阅文献[1]可知玻璃的颜色主要由着色剂引起，如蓝色的Co^{2+}、绿色基体的CuO、红色和黑色着色剂Fe_2O_3，红色的Mn^{3+}与Fe^{2+}和Fe^{3+}组合成紫色，因此我们考虑将颜色划分为红绿蓝，并针对颜色深浅赋值不同权重。考虑到处理后的数据为离散型数据，无法采用皮尔逊相关系数，因此考虑选用斯皮尔曼相关性系数进行相关性分析。在相关系数的基础上进行回归分析对相关性进行进一步研究。考虑到风化与否是 0-1 变量，且传统的多元线性回归需要大样本、满足正态分布等前提，条件太强，我们选择采用在小数据集上表现优秀的逻辑回归模型，并用其回归系数对各个特征的相关性大小进行分析和比较。

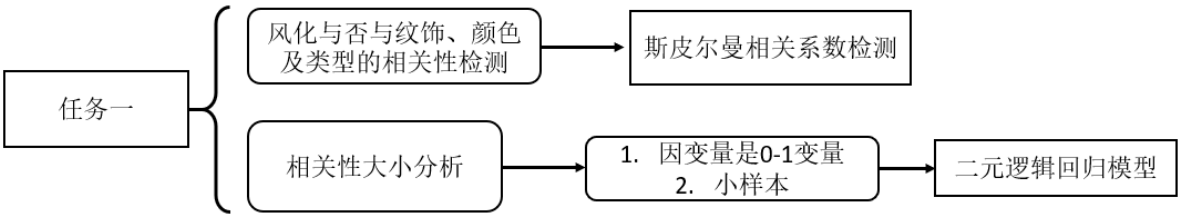


图 1 问题一任务一思路流程图

为分析风化与未风化文物表面化学成分含量的统计规律，我们从两个角度考虑此问题，其一是研究风化与每种化学成分含量的相关性程度，其二是分别对风化与未风化的文物样本进行描述性统计并可视化，分析其前后变化。对于角度一，考虑到组成成分自变量太多且显著性未知，若直接使用多元回归模型分析相关规律，容易产生回归系数不显著等问题。因此我们先采用可以将不重要变量的回归系数压缩至零的 lasso 回归模型，对 14 个化学组成成分进行特征选择，得出与风化相关性较大的化学成分。然后，我们利用三分类问题的特点，沿用任务一的逻辑回归思想，对 lasso 选择出的相关性较大的化学成分自变量，选择多元逻辑回归方法，建立风化程度与主要相关化学成分间的相关性分析模型。最后我们对所有 14 个自变量直接进行多元逻辑回归并，通过两个模型的对比，验证 lasso 特征选择对相关性分析模型的促进作用。

其次，联系任务三由风化后的数据预测风化前的化学成分的要求，我们要分类进行风化前后的针对性分析，挖掘玻璃在风化过程中隐性的化学成分变化信息。由于已知数据的离散性，我们可以分别对每一个成分对比分析风化前后两个状态的比例。为避免数据绝对大小对变化的错误估计，我们以风化后为标准，将化学成分的绝对比例转换成相对变化比例进行分析；最后，我们可以将得到的变化百分比结果与前一问中的多元逻辑回归系数进行对比，与各化学成分因子的影响程度相互印证。

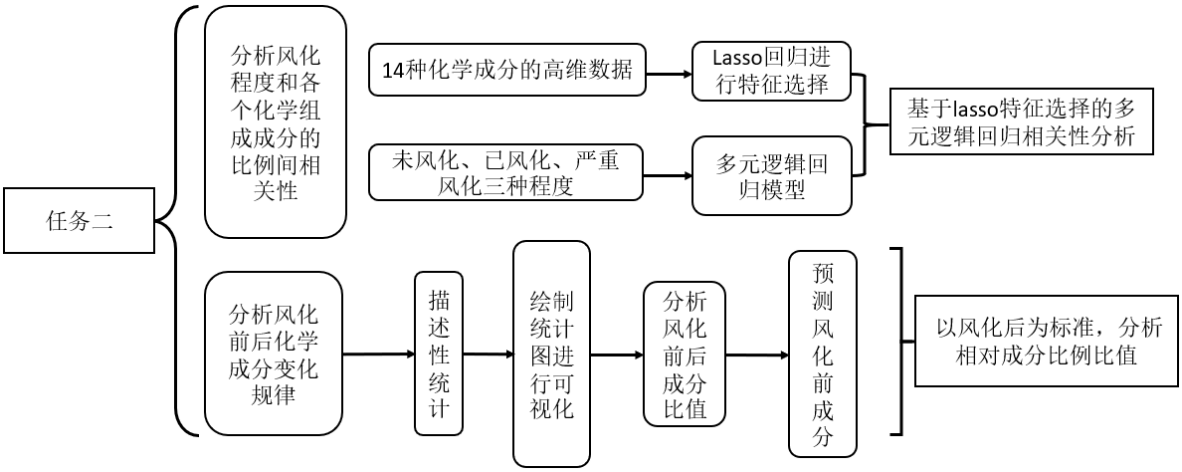


图 2 问题一任务二思路流程图

对于任务三的预测，我们利用任务二找出的风化过程中玻璃化学成分的相对变化规律，对已知的风化点数据进行逆推。同时，由于风化过程与周围环境及采样部位有着极大联系，数据点的不确定性较强，我们赋予相对百分比一定范围的误差，实现风化前各化学成分的上下界还原。

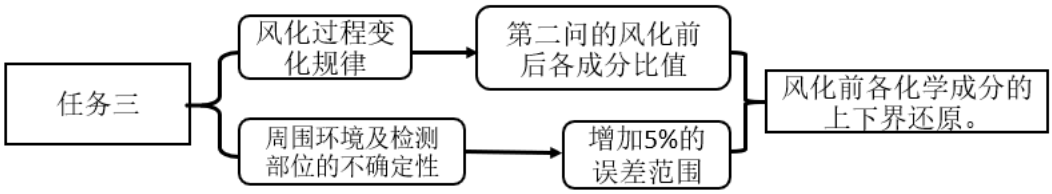


图 3 问题一任务三思路流程图

2.2 问题二的分析

任务一要求利用表单一及表单二中已知的全部数据，给出两类玻璃的分类规律。这是一个多特征的有监督分类任务，需要研究风化、纹饰、颜色、化学成分等多个特征对玻璃分类的影响从而找出分类规律。通过对机器学习中多种分类器如随机森林、支持向量机、决策树等进行比较，我们最终选择了适合处理多特征高维度小样本数据，泛化能力强，可解释性强的决策树模型，并对传统的决策树模型进行改进，选用 Gini 指数作为衡量不纯度的标准建立 CART 决策树，并适当采取了剪枝防止决策树过拟合。

任务二要求选择一定的化学成分，对 49 个铅钡玻璃样本点及 18 个高钾玻璃样本点进行亚分类，并给出分类方法及结果，再对分类结果进行合理性及敏感性分析。首先，我们运用 SPSS（Statistical Package for Social Science）软件进行系统聚类，得到两种玻璃的亚类划分结果。然后，我们针对每个类别间的化学成分差异，分析亚类划分标准，再结合论文资料中我国古玻璃依据化学成分分类的不同分类名称，对亚类进行划分标准确立及名称制订，并据此对本题样本点的分类结果进行合理性分析。最后，为了对分类结果敏感性进行检验并可视化分类效果，我们选择主成分分析法对十四种化学成分数据进行降维，得到二维平面上的成分载荷图，对亚类划分结果进行直观解释。

2.3 问题三的分析

问题三要求对给出化学成分与表面风化情况的未知类别文物进行分类，并进行灵敏度分析。为了更好地提高分类结果的准确度，我们在问题二 CART 决策树分类模型的基础上，又建立了逻辑回归模型求解分类结果，比较两种模型的分类能力与分类结果，选取更优者进行预测，倘若两个模型的预测效果均相同则说明预测结果非常准确。为了研究分类结果的敏感性，我们绘制出了两个模型的 ROC 曲线，并且针对逻辑回归模型采用基于偏导数的敏感度分析研究化学成分的微小变化对分类概率的影响。在此基础上，我们使用 PCA 主成分分析降维的方法对不同类型的样本进行可视化，通过绘制二维平面图分析不同类别样本之间的距离，如果距离远则说明敏感性较强。

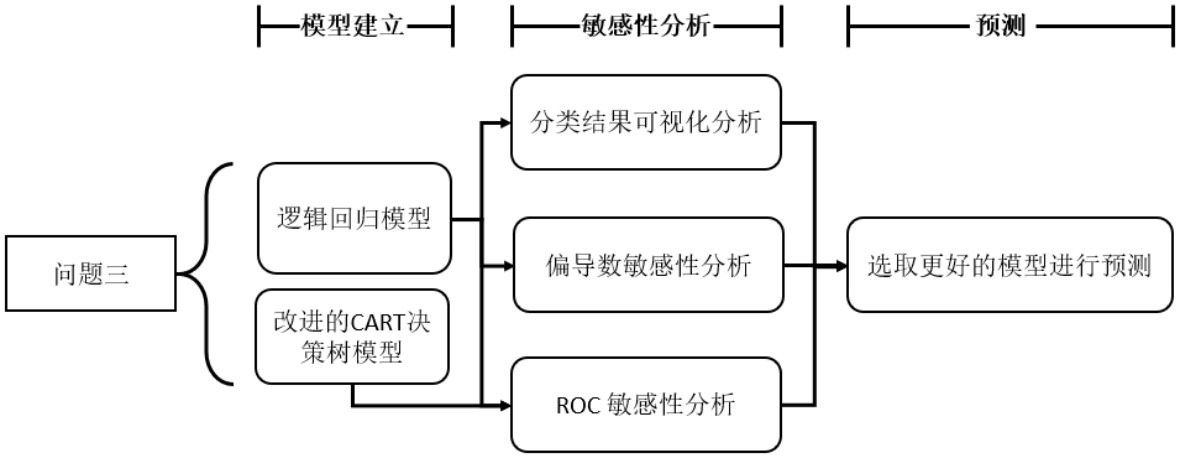


图 4 问题三思路流程图

2.4 问题四的分析

问题四的任务是分别对两类玻璃，研究其化学成分间的相互关系，再比较这两类玻璃的化学成分相互相关性的差异。我们从两个角度考虑化学成分的关联关系，其一是化学成分含量两两之间的关系，其二是一组变量之间的相关性分析。对于前者，我们运用 spearman 相关系数进行探索，并通过相关性热力图进行可视化，通过对比冷暖色块的数量及颜色深浅，直观定性地分析两种玻璃之间化学成分相互关系的差异；对于后者，由于方差膨胀因子（VIF）可以用于多重共线性，我们考虑对其中所有化学成分以 2-7 个为一组进行分组，遍历所有组别计算 VIF，筛选出有最大相互关系的一组化学成分，并结合文献资料进行结果解释。考虑到因子分析法可用于探索变量间的相互联系，我们利用因子分析筛选出剩余较弱相关性的变量组，用正交因子旋转得到因子载荷矩阵，对剩下的化学元素进行相互关系探究得到完全无关的化学成分。并通过两种玻璃相关性分析的结果进行对比，并进行相互关系差异的解释。

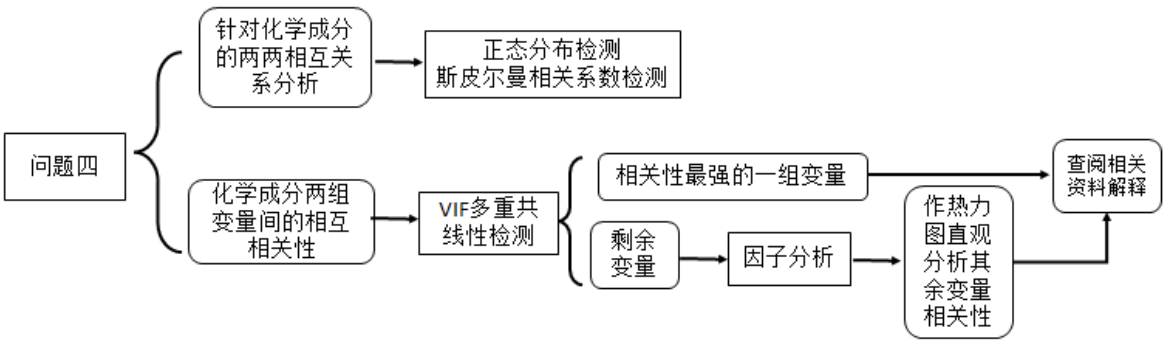


图 5 问题四思路流程图

三、 模型假设

1. 假设不考虑有效数据中因检测仪器带来的化学成分测量误差
2. 假设考古工作者对样本点的类型鉴定及风化程度鉴定准确无误
3. 假设风化前后化学成分按比例进行扩大缩小
4. 不考虑严重地质变化、自然灾害等对玻璃风化中化学成分的异常改变

四、 符号说明

表 1 符号说明

符号	说明
ω	回归系数向量
b	回归方程截距
z	回归函数
J	损失函数
λ	正则化系数
r_s	斯皮尔曼相关系数
D	样本集合
Sensitivity	敏感度
FPR	真阳性率
TPR	假阳性率
VIF	方差膨胀因子
f	公共因子
F	公因子向量
A	载荷矩阵
ε	残差向量

五、 模型的建立与求解

5.1 问题一模型的建立与求解

5.1.1 数据预处理

处理表单 1：对于 58 种文物的纹饰、类型、颜色和表面风化等文本型数据，使用编码的方式转化为数值型数据。首先针对 A、B、C 三种纹饰、“高钾”和“铅钡”两种类型、表面风化与否创建哑变量进行独热编码。针对颜色的编码，我们查阅相关资料得知，铅钡玻璃和高钾玻璃的颜色多为绿、蓝、紫、黑，玻璃之所以呈现五彩缤纷的颜色是因为制作过程中加入了元素周期表中的副族元素作为着色剂，文物颜色的种类与深浅往往和金属元素的种类和含量有关，如Co^{2+}为较强的蓝色着色离子，Mn^{3+}着色为红色，可与如Fe^{2+}或Fe^{3+}组合成紫色，蓝（绿）色着色剂则以CuO和Fe_2O_3为主，黑色和红色着色剂以Fe_2O_3为主。据此，我们将颜色划分为红绿蓝三种变量，并针对颜色的深浅创建有序变量，如文物颜色为“浅蓝”则变量蓝色为 1，深蓝则为 2，“蓝绿”则变量蓝色和绿色均为 1。

处理表单 2：由于只有化学成分比例总和在 85%-105%的数据是有效的，我们将无效数据文物 15 和 17 剔除。另外，鉴于比例总和非常接近 100%，且考虑到不可避免受到杂质和测量手段的影响，此处无需对组成比例按百分比进行重新赋值。对表单 2 进行进一步分析，观察到风化程度不仅含表单 1 中的风化与否，而且还对部分文物的严重风化点进行采样，考虑到严重风化点对研究风化过程中化学成分含量的变化具有较高参考价值，我们对其附以更高的权重，即用 0，1，2 分别表示风化的程度，即无风化、已风化与严重风化。

5.1.2 模型的建立

➢ 逻辑回归模型

逻辑回归（Logistic Regression）是一种利用概率思想解决分类问题的分类器，擅长处理标签为满足 0-1 分布的离散型变量。逻辑回归的输入是线性函数，它通过引入联系函数（link function）——*Sigmoid* 函数，对线性回归函数 z 变换为 $g(z)$，使其值分布在 $(0,1)$ 区间，$g(z)$ 反映分类的概率，接近 1 时样本标签为 1，接近 0 时则标签为 0，逻辑回归的表达式为：

$$g(z)=\frac{1}{1+e^{-z}} \tag{1}$$

将线性回归模型中的 $z=\omega x+b$ 带入，得到逻辑回归的一般概率公式：

$$\begin{aligned} P(Y=1\mid x)&=\frac{e^{\omega x+b}}{1+e^{\omega x+b}}\\ P(X=0\mid x)&=\frac{1}{1+e^{\omega x+b}} \end{aligned} \tag{2}$$

其中，$x\in R^n$ 为输入，$y\in\{0,1\}$ 为输出，$w\in R^n$，$b\in R$ ，w 为权重向量，b 称为偏置。将 $z=wx+b$ 写作矩阵形式 $z=\theta' x$，逻辑回归的损失函数由最大似然法得出：

$$J(\theta)=-\sum_{i=1}^{m}(y_i\log(y_\theta(x_i))+(1-y_i)\log(1-y_\theta(x_i))) \tag{3}$$

逻辑回归模型运用已知数据及其对应标签，找到使得损失函数最小的参数 ω 和 b，根据计算出的系数矩阵 ω 可以探究特征 X 与标签 Y 之间的关系来判断哪些特征与分类结果相关及相关程度的大小，并且也可以对新的数据依据概率预测分类结果。

在本题中，我们利用逻辑回归针对输入的颜色、纹饰、类别等构成的维度为 1×8 的向量 x，以及对应的分类结果风化与否标签 y，找到构成决定风化与否的决策边界 $\omega x+b$，使得其损失函数最小。我们再通过逻辑回归模型返回的颜色、纹饰等各个特征的回归参数 ω 对因变量风化与否和自变量颜色、纹饰、类别的相关关系进行量化分析。

➢ 基于 Lasso 回归自变量选择的多元逻辑回归模型

Lasso 回归是 1996 年由 Tibshiran 提出的多元线性回归的优化模型。它对传统的 OLS 的回归系数加上了 L2 惩罚项，能够识别出回归模型中的不重要变量并且将其系数压缩至零。因此，lasso 回归常常用于对高维数据进行特征选择，可以代替过程繁杂的逐步回归，对模型进行简化；又由于其惩罚项随系数调整而灵活变化，又使模型的可解释性增强。Lasso 回归系数确定如下：

$$\hat{\beta}=\arg\min\left[\sum_{i=1}^{n}(y_i-x_i^T\hat{\beta})^2+\lambda\sum_{i=1}^{k}\left|\hat{\beta}_i\right|\right] \tag{4}$$

由于分类结果涉及到无风化、已风化、严重风化，为三分类问题，定义三组二分类逻辑回归的权重 ω^0，ω^1，ω^2，使用联系函数 *softmax*，得到多元逻辑回归的一般概率公式：

$$P(y\mid x)=\frac{e^{W_y\cdot x}}{\sum_{c=0}^{N-1}e^{W_c\cdot x}} \tag{5}$$

定义每个类别 a 的概率为：

$$P(Y=a)=\frac{e^{\sum_i \omega_i^a x_i^a}}{\sum_{j=0}^{2} e^{w}} \tag{6}$$

对每个类别求出其分类概率，选择概率最大的类别作为分类结果。

5.1.3 模型的求解

Part 1：对于问题一任务一，首先使用 *spearman* 等级相关系数对不同特征的相关性进行分析， *spearman* 等级相关系数的定义如下：

$$r_s=1-\frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \tag{7}$$

设置置信水平为 99%，即采用 0.01 的 p 值验证了风化与否与玻璃类型、纹饰及颜色三个特征均有显著的相关性。然后，我们以是否风化为标签，以颜色、纹饰及类型经数据预处理后得到的 8 个特征为自变量，建立二元逻辑回归模型，得到各个特征回归系数如下。

表 2 二元逻辑回归系数

	纹饰 B	高钾	铅钡	纹饰 A	纹饰 C	蓝	绿	红
系数	**1.52**	**-1.02**	**1.02**	**-0.9**	**-0.6**	**0.6**	**0.36**	**0.02**

从上表中可以看出其中与风化最大关联的是纹饰 B，回归系数为 1.5184，说明带有纹饰 B 的玻璃制品极易风化。直接从表单 1 也可以观察到同样的结果，58 件文物中共有 6 件文物是纹饰 B 且全部风化。查阅相关资料可知，不同纹饰可能源自不同的制作工艺和制作材料，且与外界环境的接触面积也不尽相同，从而导致风化状态的差异。第二、第三位的特征分别是高钾、铅钡两种玻璃类型，且铅钡玻璃为正而高钾玻璃为负，说明玻璃的类型对是否风化具有较大影响，而且铅钡玻璃比高钾玻璃更易于风化。风化状态与玻璃类型紧密相关，是因为不同类型的玻璃炼制时加入了不同的助熔剂导致化学成分含量各异，因此风化的难易程度也不同。剩下的特征回归系数均小于 1，并不是造成风化的主要原因，例如三个颜色特征均较为不显著，结合古玻璃的着色制作过程，着色剂（如氧化铜，氧化铁等）并不是玻璃的主要成分，仅为美观而在制作过程中添加，因此含量往往很低，因此与风化的相关性较弱。

Part 2：对于问题一任务二，我们首先通过 Lasso 回归选取主要化学成分进行逻辑回归，通过回归系数研究各种化学成分含量对文物表面有无风化的相关性强弱的统计规律；其次对风化前后两类玻璃的各种成分进行了描述性统计，通过计算平均值分析了风化前后成分变化的比例，并对上述统计规律进行了可视化分析。

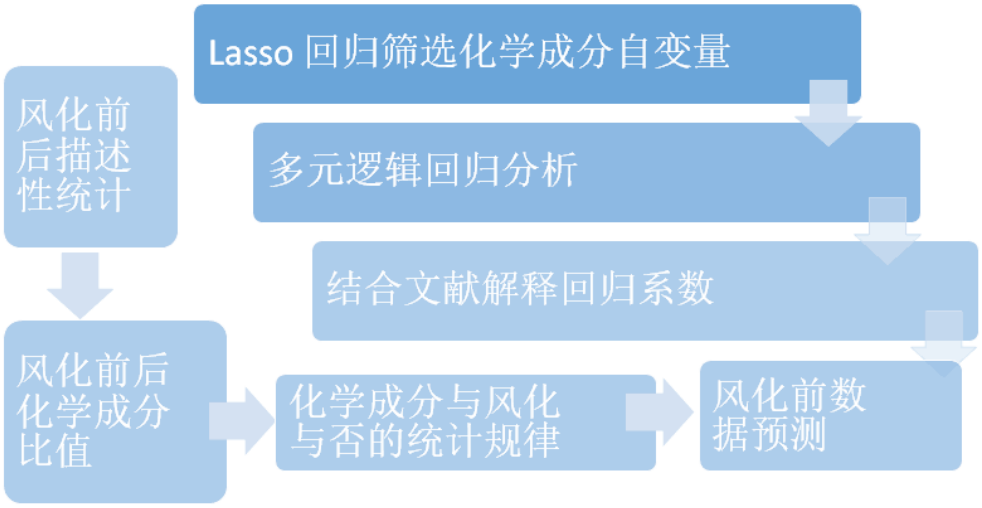


图 6 问题一任务二流程图

step 1：研究各种化学成分含量对文物表面有无风化的相关性强弱

考虑到文物的化学成分共有 14 种，过高维度的数据会影响逻辑回归的效果，因此考虑先使用 Lasso 回归对氧化物变量进行降维，将不重要的变量的系数通过 L2 惩罚项压缩到 0，进而达到变量选择的目的。得到铅钡玻璃与高钾玻璃的 Lasso 回归系数如下：

表 3 铅钡玻璃的 Lasso 回归系数

成分	SiO2	Na2O	K2O	CaO	MgO	Al2O3	Fe2O3	CuO	PbO	BaO	P2O5	SrO	SnO2	SO2
系数	**-1.3**	**0.0**	**0.0**	**-4.0**	**0.0**	**2.63**	**0.0**	**0.0**	**1.55**	**0.0**	**3.955**	**0.0**	**0.0**	**6.11**

表 4 高钾玻璃的 Lasso 回归系数

成分	SiO2	Na2O	K2O	CaO	MgO	Al2O3	Fe2O3	CuO	PbO	BaO	P2O5	SrO	SnO2	SO2
系数	**2.91**	**0.631**	**-1.0**	**1.37**	**0.0**	**0.0**	**0.0**	**1.4**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**

上表中系数为 0 的化学成分与风化与否的相关性较弱，将其剔除后得到如下结果：

铅钡玻璃与风化相关程度较高的变量有：SiO_2、CaO、Al_2O_3、PbO、P_2O_5 和 SO_2。

高钾玻璃与风化相关程度较高的变量有 SiO_2、Na_2O、K_2O、CaO 和 CuO。

通过 lasso 回归后将影响铅钡玻璃和高钾玻璃风化的自变量分别减少为 6 个和 5 个，防止了不相关的变量对统计规律研究的干扰。

为了进一步研究上述变量的相关性强弱，我们使用逻辑回归模型进行分析。由于风化程度有未风化、已风化、严重风化三种取值，因此考虑使用多元逻辑回归处理三分类问题，根据回归系数的大小对相关性强弱进行排序。

利用 Lasso 回归选择出的化学成分作为自变量，以风化程度作为因变量，分别对两种类型的玻璃进行多元逻辑回归分析。以铅钡玻璃为例得到的回归系数如下：

表 5 铅钡玻璃风化主要影响成分的多元逻辑回归系数

成分	SiO2	CaO	Al2O3	PbO	P2O5	SO2
未风化	**0.36**	**0.57**	**0.33**	**-0.25**	**-0.65**	**-0.0**

风化	**-0.02**	**-0.29**	**-0.35**	**-0.06**	**0.032**	**-0.19**
严重风化	**-0.33**	**-0.27**	**0.02**	**0.188**	**0.62**	**0.20**

为了检测模型效果，我们通过蒙特卡洛模拟的方式对测试集和训练集进行 100 次随机划分，统计出每次预测结果的平均值得出模型在测试集上的平均精度为 84.2%，说明我们的回归模型较为准确。

结合相关文献，对上述化学成分含量与风化与否的相关性分析结果进行解释如下：

①对于严重风化，由于样本点较少，具有较大偶然性误差，只选取显著负相关的五氧化二磷、二氧化硅进行分析：由于严重风化的样本部位产生磷酸钙沉积表面，所以五氧化二磷显著上升；同时严重风化的样本点与外界环境的元素交换更多，主成分二氧化硅会明显减少。

②对于风化一类，二氧化硫、氧化钙及氧化铝与风化标签具有较强负相关性，这是因为风化后助熔剂氧化钙及氧化铝流失，而二氧化硫转化成难溶性硫元素在外壳沉积，故比例有一定减少。

③对于未风化一类，二氧化硅、氧化钙、氧化铝具有正相关性，因为风化前助熔剂还未流失；氧化铅、五氧化二磷具有强负相关性，这是因为风化过程中铅的成分会逐渐降低，磷也会转移至表面沉积。

- 回归模型效果对比

为对比 Lasso 回归的自变量选择效果，我们尝试以下方法对比我们基于 Lasso 特征选择的多元逻辑回归模型的效果。①对全部变量的直接进行多元线性回归。②对选出的变量进行多元线性回归。③对全部变量进行多元逻辑回归。结果如下：

表 6 模型效果对比

模型	评估指标	指标值
基于 lasso 特征选择的多元逻辑回归模型	精确度	84.2%
全部变量的多元逻辑回归模型	精确度	82.2%
全部变量的最小二乘回归模型	R^2	0.874
基于 lasso 特征选择的最小二乘回归模型	R^2	0.836
多元线性回归模型	R^2	0.445%

由于数据量较小，针对全部变量的回归模型均容易产生过拟合现象，导致精确度偏高但相关性与实际不符。在这种情况下，选择后的逻辑回归模型返回的 84.2%与未选择返回的 82.2%对比，仍然精确度更高，可以得出我们的特征选择模型具有较大优势。

对于高钾玻璃进行类似分析，由于只有风化与否两类，我们对 Lasso 回归选择出的 5 个成分进行二元逻辑回归，得到的逻辑回归结果如下：

表 7 高钾玻璃逻辑回归系数

成分	SiO2	Na2O	K2O	CaO	CuO
系数	**0.620**	**0**	**-0.467**	**-0.08**	**0.065**

Step 2：对风化前后两类玻璃的各种成分进行了描述性统计并分析化学成分变化

分别对铅钡和高钾两类玻璃已风化与未风化的样本进行描述性统计，计算出均值、标准差、最小值、四分位数、中位数、最值、偏度、峰度。由于篇幅有限此处仅展示

铅钡玻璃的描述性统计结果：

表 8 铅钡玻璃已风化数据描述性统计

成分	SiO2	Na2O	K2O	CaO	MgO	Al2O3	Fe2O3	CuO	PbO	BaO	P2O5	SrO	SnO2	SO2
mean	24.92	0.216	0.133	2.69	0.65	2.97	0.58	2.27	43.31	11.80	5.27	0.41	0.068	1.36
std	10.61	0.556	0.239	1.65	0.71	2.63	0.73	2.82	12.23	9.978	4.19	0.26	0.26	4.20
min	3.72	0	0	0	0	0.45	0	0	15.71	0	0	0	0	0
25%	18.79	0	0	1.34	0	1.40	0	0.70	35.47	6.77	1.31	0.22	0	0
50%	25.02	0	0	2.87	0.57	2.38	0.305	1.145	44.06	8.79	4.9	0.42	0	0
75%	30.20	0	0.24	3.50	1.16	3.5	0.98	3.13	48.84	14.50	8.5	0.59	0	0
max	53.33	2.22	1.05	6.4	2.73	13.65	2.74	10.57	70.21	35.45	14.13	1.12	1.31	15.95
skew	0.31	2.66	2.51	0.383	1.04	2.82	1.40	2.10	-0.03	1.27	0.40	0.54	4.36	3.26
kurt	3.78	8.29	9.15	2.40	3.76	11.47	4.02	6.17	2.97	3.46	2.10	3.41	18.95	10.65

为了增强结果的可读性，对其做出可视化图像如下：

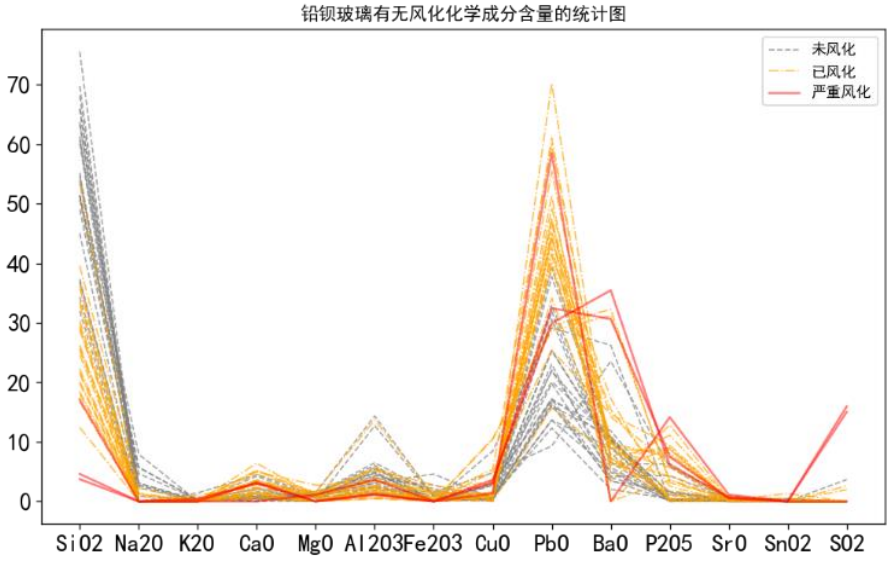


图 7 铅钡玻璃有无风化化学成分含量的统计图

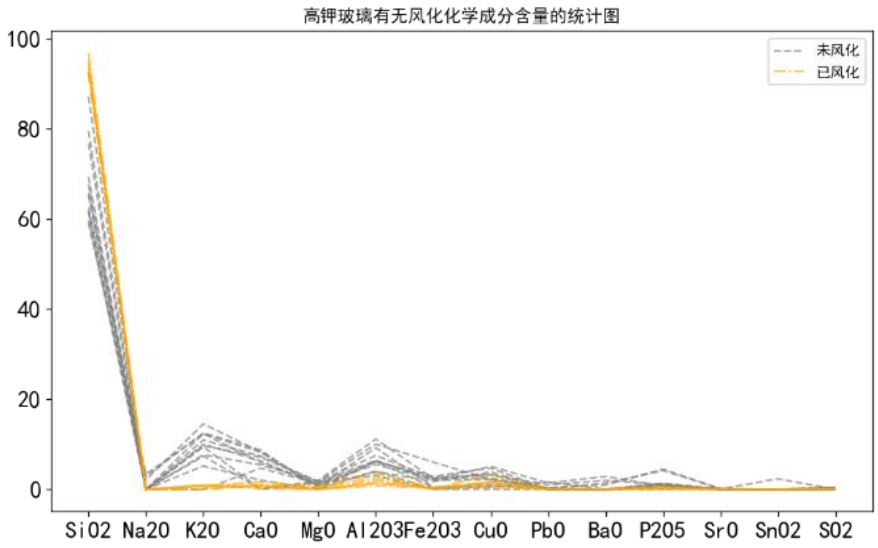
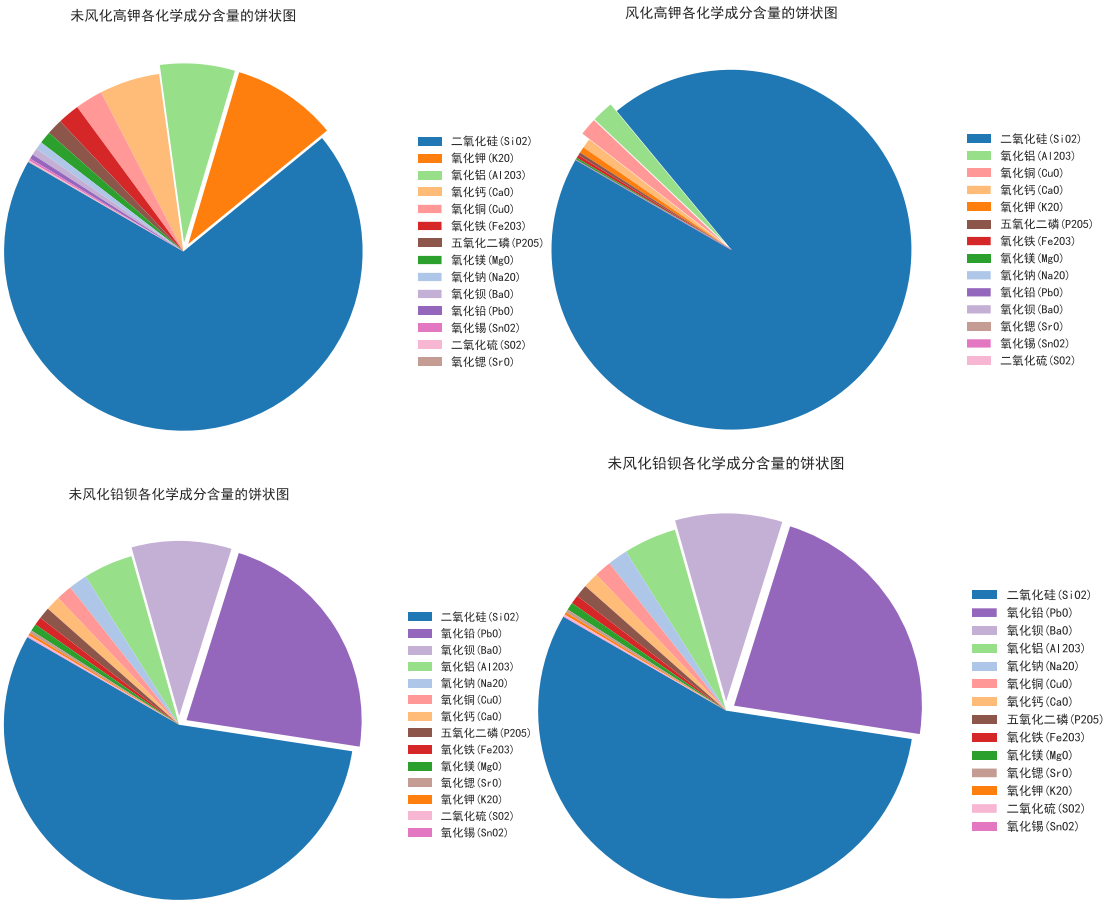


图 8 高钾玻璃有无风化化学成分含量的统计图



然后，我们以风化后的均值作为标准，求出风化前的各个化学成分相对比值如下：其中由于高钾风化组只有两个数据且有值为 0 的情况，固用 inf 表示无法进行比值操作的化学成分。

表 9 风化前后的各个化学成分相对比值

成分	SiO2	Na2O	K2O	CaO	MgO	Al2O3	Fe2O3	CuO	PbO	BaO	P2O5	SrO	SnO2	SO2
高钾	**2.19**	**7.78**	**1.64**	**0.49**	**0.99**	**1.50**	**1.26**	**0.63**	**0.51**	**0.76**	**0.20**	**0.64**	**0.68**	**0.12**
铅钡	**0.72**	**inf**	**17.17**	**6.13**	**5.49**	**3.43**	**7.29**	**1.57**	**inf**	**inf**	**5.01**	**inf**	**inf**	**inf**

从折线统计图和化学成分相对比值表可以观察到如下现象：

对于铅钡玻璃，风化后氧化纳流失将近 8 倍，二氧化硅，氧化钾、氧化铁及氧化铝也有一定程度的下降，而氧化镁含量相差不大。相反，五氧化二磷，二氧化硫在风化后采样点数据分别上升约 5 倍和 10 倍，氧化钙及氧化铅也上升一半左右；而对于高钾玻璃，氧化钾流失约 17 倍，氧化铁流失约 7 倍，氧化钙、氧化镁及五氧化二磷流失超过 5 倍，氧化铝流失约三倍。

查阅相关资料，我们可以印证模型求解出的风化前后化学成分变化规律：①古玻璃在风化过程中，助熔剂氧化纳、氧化钾、氧化钙等有着不同程度的流失，因此针对高钾玻璃，钾元素的流失量极为显著，针对铅钡玻璃，钙及钠元素流失显著。②对于铅钡玻璃，难溶性钡盐，硫元素和磷酸钙在风化过程中容易在玻璃表面沉积，导致样品检测点数据飞跃，这就是铅钡玻璃磷、硫元素在风格化后急剧上升的原因。同时，这个结果与任务二中的多元逻辑回归结果分析较为一致，显示出风化前后统计规律的合理性。

对于任务三，考虑到周围环境及检测部位的不确定性，我们假设各化学成分在 5%

的偏差范围内等比例扩大、缩小，于是我们对任务二的比值乘以100±5%作为比值的上下界，得到一个比值范围。最后，我们筛选出题目要求预测的风化点，用风化点数据乘以比值，逆推得到未风化时各成分的区间，如下表。（由于篇幅原因，我们只截取了高钾组的部分化学成分。其余在附录展示）

表 10 文物 7,9 风化前各成分范围预测

文物采样点	二氧化硅(Si02)	氧化钾(K20)	氧化钙(Ca0)	氧化铁(Fe203)	氧化铜(Cu0)
07	(63.38，70.05)	(0.0，0.0)	(5.86，6.47)	(1.6，1.77)	(3.65，4.03)
09	(65.01，71.86)	(17.16，18.97)	(3.39，3.75)	(3.02，3.33)	(1.75，1.93)

我们不采用单值预测而是区间预测，给出了风化样本点未风化时的成分范围，兼顾了数据的随机性与精确性，预测模型合理，结果符合实际。

5.2 问题二模型的建立与求解

5.2.1 模型建立

- CART 决策树

决策树(Decision Tree)是一种有监督的学习模型，能够从一系列有特征和标签的数据中总结出决策的规则，并且能以树状图地形式直观地呈现每次的决策规则，相比于其他算法有很强的可解释性，对小样本集有较大优势，对分类和回归问题都具有很好的效果。

CART 决策树算法是对 ID3 决策树算法和 C4.5 决策树算法的改进，使用了衡量数据集纯度的基尼系数取代了对不纯度更加敏感的信息熵进行二分递归，Gini 系数越小样本类别的不确定性越小，这种方式一定程度上减少了过拟合发生的可能性，有效提高了预测的精准度，其主要计算步骤如下：

图 9 CART 决策树算法流程图

1) 计算初始 Gini 系数值。

对本题中五十余种文物组成的样本集合 D，包含 N 个类别(即铅钡玻璃和高钾玻璃)，计算 D 所对应的基尼系数为

$$Gini(D)=1-\sum_{i=1}^{N}p_i^2 \tag{8}$$

2) 对于该样本集合 D 包含的 M 个样本，根据特征 A 的第 i 个值，将 D 分割成 D_1 和 D_2， $Gini$ 系数为

$$Gini_{A,i}(D)=\frac{D_1}{D}Gini(D_1)+\frac{D_2}{D}Gini(D_2) \tag{9}$$

3) 在最优二分方案下选择基尼系数的最小值作为文物组成的样本集合的分割方案。

$$\min_{A\in Attribute}\{\min_{i\in A}[Gini_{A,i}(D)]\} \tag{10}$$

4) 重复上述步骤，直到 CART 决策树叶子结点中的样本无杂质或达到剪枝的阈值。

5.2.2 模型求解

可能影响玻璃种类划分的变量有这些文物的 14 种化学成分和文物的颜色、纹饰，风化与否。我们将颜色和纹饰按第一问相同的处理方式创建哑变量，将其和化学成分含量一并作为特征，将已分类的两种类型的玻璃作为标签，建立 CART 决策树模型。同时，由于决策树易过拟合的性质，我们采用了两种方式防止过拟合，其一是限制 CART 决策树的最大深度，将超过设定深度的树枝全部剪掉；其二是限制每个节点分枝后子节点最少包含的训练样本个数，考虑到样本数量低，且若该超参数设置太大会阻止模型学习数据，设置太小会引起过拟合，综合考虑后选择叶节点最小样本点为 15 时效果较好。

从全部样本中选择 20%作为测试集，基于 80%训练集。将基于训练集得到的 CART 决策树对测试集样本进行分类，计算分类的准确度为 100%，可见效果极佳。

得到的 CART 决策树的判别规则如下：

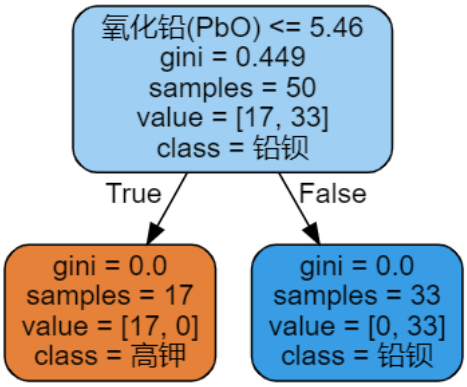


图 10 决策树判别树状图

再根据每一步计算出的 Gini 指数归一化后返回每个特征的重要程度。

5.2.3 模型效果分析

由上述建立的 CART 决策树分类规则可知，对五十余种玻璃文物组成的样本空间进行高钾玻璃和铅钡玻璃的类别分类，仅需判断样本中氧化铅的含量即可，如果大于 5.46%则说明属于铅钡玻璃,否则属于高钾玻璃，分类得到的不纯度为 0,说明效果十分优秀。上述分类标准正与铅钡玻璃在烧制过程中加入铅矿石作为助熔剂导致其氧化铅(PbO)的含量较高对应。

5.2.4 任务二的模型建立与求解

- 系统聚类模型

系统聚类是一种以最小距离为标准，两两组合两类数据点，重复合并过程直到所有数据点被归为一类的聚类模型。具体算法流程图如下：

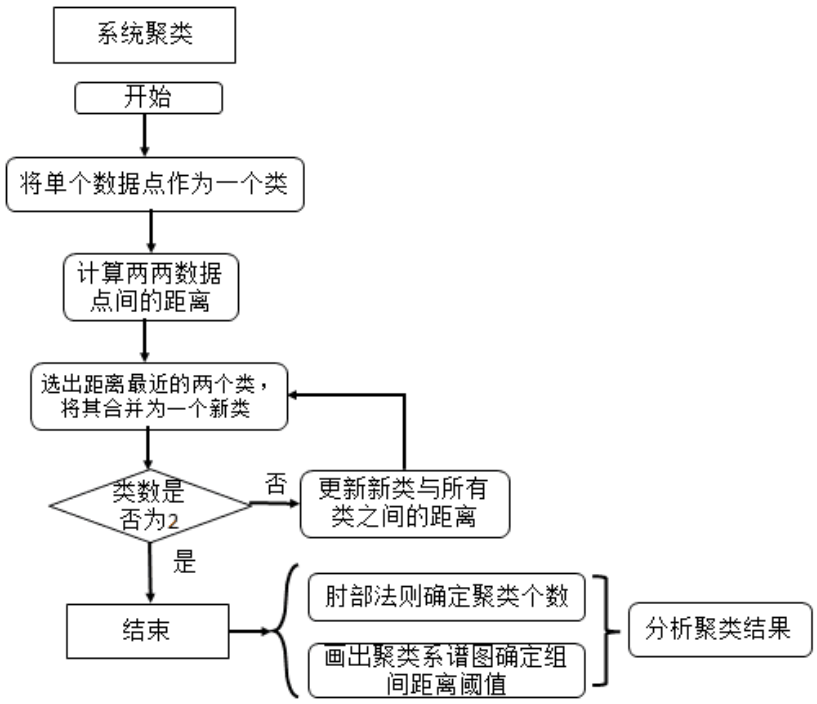


图 11 系统聚类算法流程图

我们以未风化样本点的各个化学成分为变量，以每个样本点代表一个事件进行聚类。运用 SPSS 软件，采用欧式平方和定义距离，运用不同的方法（离差平方和法、重心法、中位数法、组间平均距离法、组内平均距离法）进行聚类，经过对比选取最终方法。

Step 1 用 SPSS 分别对两种玻璃进行亚类划分

我们对 49 个铅钡玻璃样本点及 18 个高钾玻璃样本点进行亚类划分，对比各类方法后我们选取组间连接法（Between-groups linkage），树形图如下：

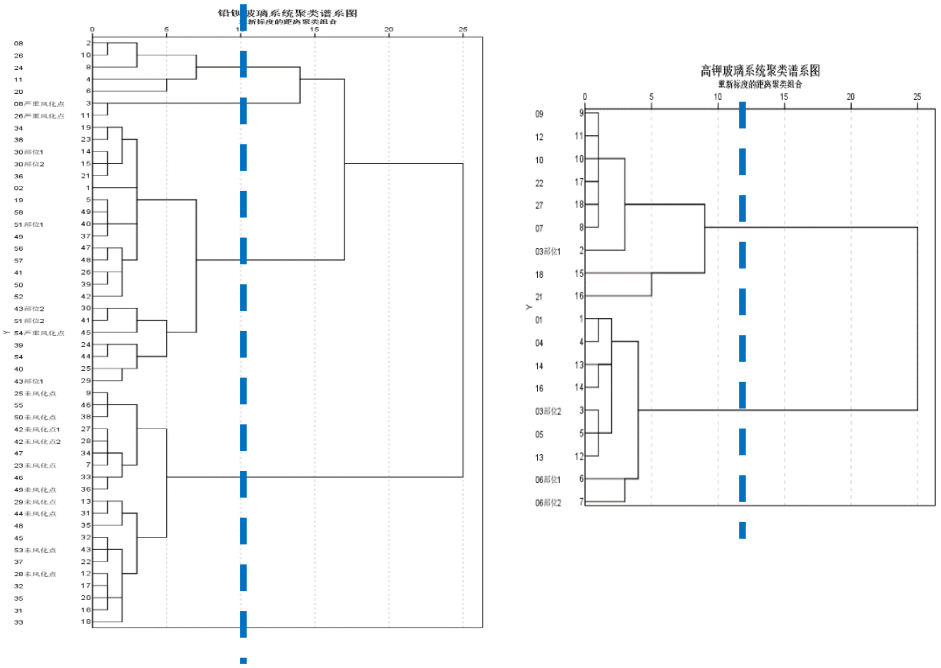


图 12 SPSS 系统聚类谱系图

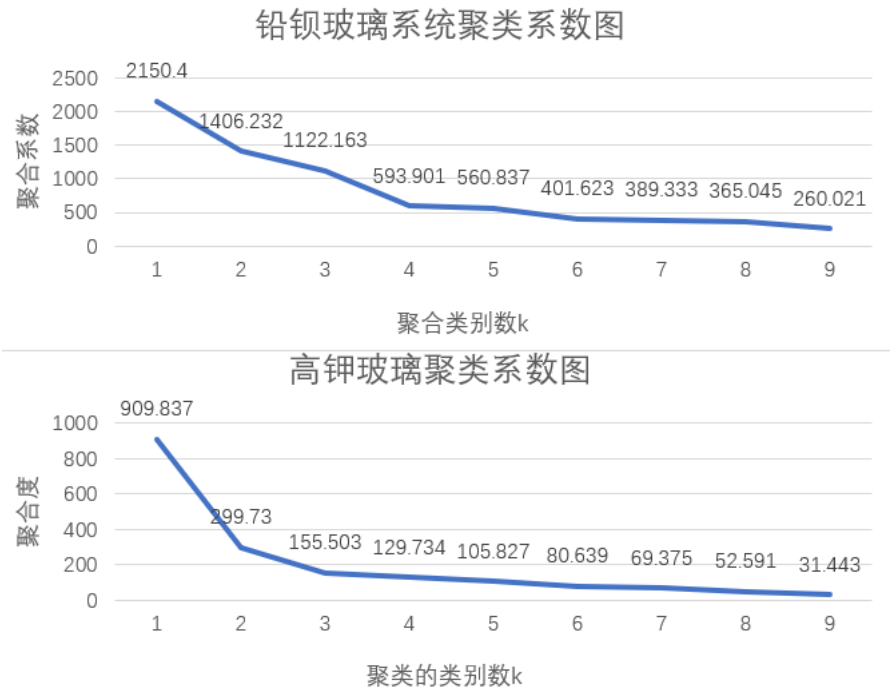


图 13 SPSS 系统聚类系数图

然后，我们使用肘部法则，画出聚类个数 k 与聚合系数间的折线图，根据折现趋势突变点确定聚类个数。由图可知，对于铅钡玻璃，聚合系数在聚类个数为 4 前显著下降，在 4 后基本稳定；高钾玻璃在聚类系数为 2 前下降迅速，在 2 后趋于平缓。由此得出聚类个数分别为 4 和 2。

聚类结果显示，铅钡玻璃 49 个数据点可以分成四个组；高钾玻璃 18 个数据点可以分成两个组。

Step 2 箱型图分析成分差异并确定划分标准

我们分别画出两种玻璃亚类划分后样本点的各化学成分箱型图，重点分析组别间差异较大的化学成分，并据此确定亚类划分标准及亚类命名。（此处仅展示组别间差异较大的元素的箱型图）

1. 铅钡玻璃亚类划分

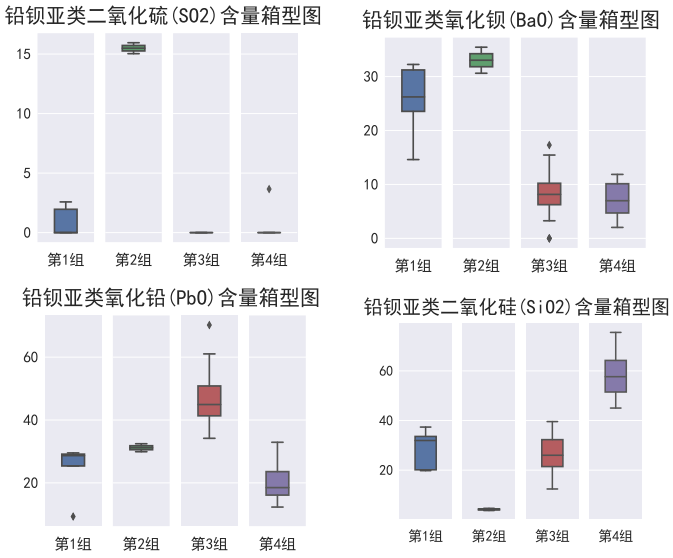


图 14 铅钡亚类含量对比箱型图

由图分析得，显著性差异存在于二氧化硅、氧化铅，氧化钡及二氧化硫四种元素：

① 二氧化硅在第一、三组中含量中等，第二组几乎为 0，第四组占比居显著高位。具体地，每一组含量分别为 20%-40%；<5%；10%-40%；45%-75%

② 氧化铅在第三组中占比较高，第四组中略低，其余中等。比例分别为 25%-30%，35%-70%，10%-30%。

③ 氧化钡在前两组中比例显著高于后两组。占比分别为 15%-30%；30%-35%；5%-20%；5%-10%。

④ 二氧化硫在第二组中高达 15%，其余组别中均极低。

值得注意的是，第二组只有两个样本点：08 样本的严重风化点和 26 样本的严重风化点，两个样本点的二氧化硅指标与其余样本点有显著差异，在百分之五以下，且二氧化硫的比例显著偏高，这说明风化极大地影响了其化学成分组成，所以我们将这个组别当成异常数据剔除，不对其进行亚类归类。

根据以上的化学成分差异分析，我们尝试将第一、三、四组划分为三个亚类，划分标准如下：

(I) PbO(~ 25*wt*%)-BaO-SiO_2 低硅低铅高钡型玻璃（第一组）：

二氧化硅含量约为 30%（较低），氧化铅约 25%（较低），氧化钡占比约 30%（较高）。

(II) PbO(~ 40*wt*%)-BaO-SiO_2 低硅高铅低钡型玻璃（第三组）：

二氧化硅占比 25%，氧化铅占比高达 50%，氧化钡占比约为 10%。

(III) PbO-BaO-SiO_2 高硅低铅低钡型玻璃（第四组）：

二氧化硅高达 60%%，氧化铅占比约 20%，氧化钡占比约 10%。

最终题目样本点的分类结果见附录 18，附录 19。

2. 高钾玻璃的亚类划分

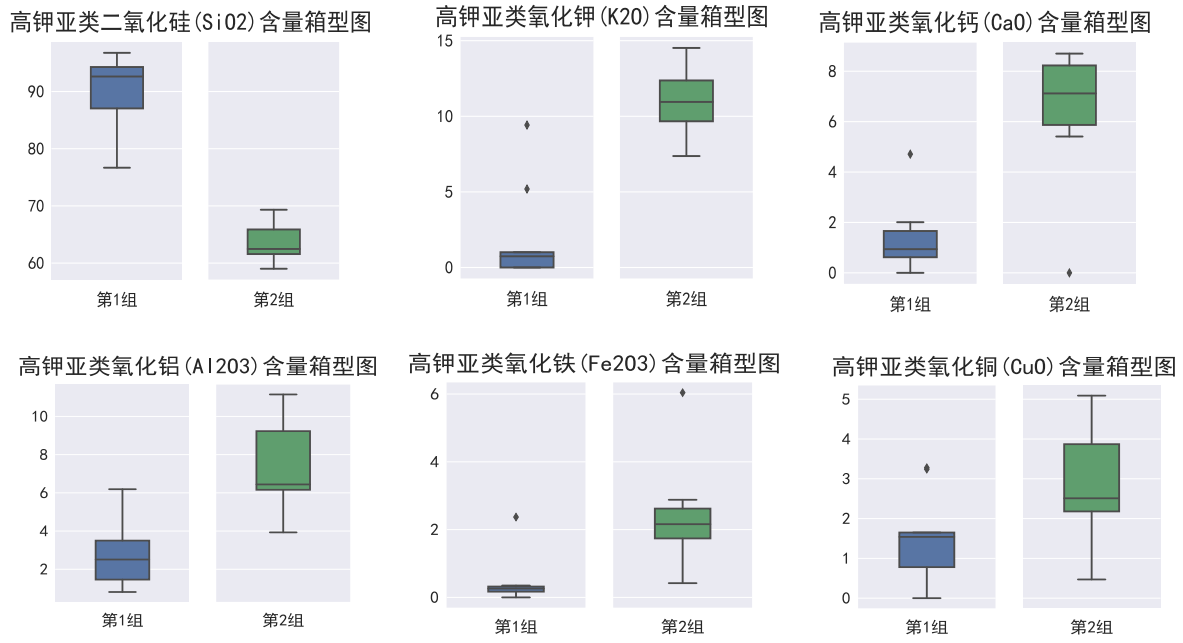


图 15 高钾亚类含量对比箱型图

由图分析得，显著性差异存在于二氧化硅、氧化钾、氧化钙、氧化铝、氧化铁及氧化铜六种元素：

① 二氧化硅在第一组中的含量为 75%-100%，明显高于第二组的 55%-70%。

② 氧化钾在第二组中占比较高，为 7%-15%；第一组中剔除离群点数据后氧化钾含量接近 0；

③ 氧化钙在第一组中占比小于 5%，在第二组中为 5%-10%

④ 氧化铝在第二组中占比 5%-12%，显著高于第一组的约 5%

⑤ 第二组中氧化铁、氧化铜含量较第一组高。

根据以上的化学成分差异分析，我们尝试将第一、二组划分为两个亚类，划分标准如下：

(I) K_2O-CaO(~7wt%)-SiO_2 中硅高钾高钙高铝型玻璃（第一组）：

二氧化硅含量约为 65%（中等），氧化钾约 10%（较高），氧化铝、氧化钙占比分别约 7%（较高）。

(II) K_2O-SiO_2 高硅低钾低钙低铝型玻璃（第二组）：

二氧化硅占比 90%，氧化钾约 2%（较低），氧化铝、氧化钙占比分别约 3%及 2%（较低）。

最终题目样本点的分类结果见附件 x。

3. 结合文献检验亚类划分的合理性

查阅相关资料，我国本土产古玻璃中对铅钡玻璃的含铅、含钡量做了细致的划分；高钾玻璃对钾、钙含量有细致划分，具体有以下五类：

C_1：K2O(~15wt%)-CaO(~10wt%)-SiO2

C_2：K2O-SiO2

C_3：PbO-BaO-SiO2

C_4：PbO(~25wt%)-BaO-SiO2

C_5：PbO(~40wt%)-BaO-SiO2

五种不同成分体系的玻璃由不同玻璃配方制造，其中第一第二类和本题中的高钾玻璃对应，C_1 类中采用较高的 K2O（~15wt%）作助熔剂，并含有较高的 CaO(~10wt%)作为调节剂；C_2 类中 K2O+SiO2 比重增加，达到 87.6w%.C_3、C_4、C_5 类属于铅钡硅酸盐玻璃，与我们的铅钡玻璃亚类划分结果中的第 III 类、第 I 类，第 II 类相符。这说明我们的亚类划分模型结果与实际考古研究结果一致，具有良好的解释性，分类结果合理可信。

Step 3 划分结果的敏感性分析及划分效果检测

我们分别对两种玻璃，使用主成分分析法对 14 个成分数据进行降维，在图中显示亚类划分效果，实现对划分结果的敏感性进行分析及检测。

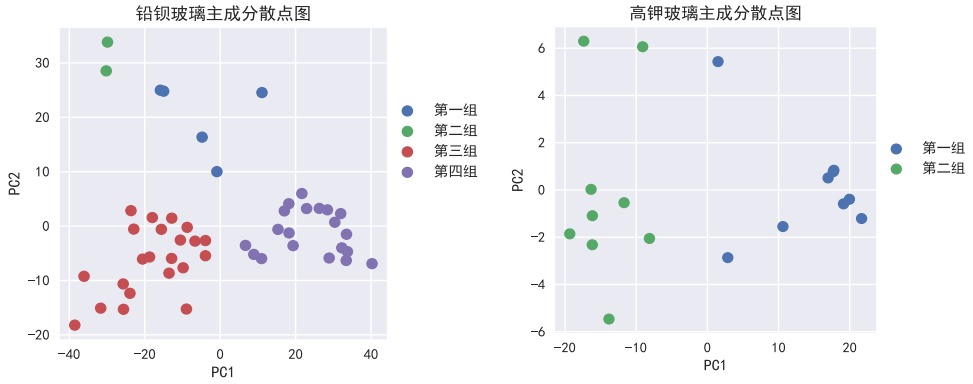


图 16 不同玻璃类别内主成分降维散点图

由图可见，铅钡玻璃第二组的严重风化点数据与其余数据点有较远距离，反映其显著的性质差异；其余红、紫、蓝三个亚类数据在二维平面上有较好的区分性，边界清晰且类别中心相距较远，这说明我们的亚类划分模型灵敏度较高，分类效果较好。同样，对于高钾玻璃，蓝、绿两类的数据点基本可以被 PC1=0 的纵轴划分，说明亚类划分较为精确，敏感性高。

5.3 问题三模型的建立与求解

5.3.1 利用逻辑回归与 CART 决策树模型对未知文物进行分类

为了更好地对 A1-A8 进行预测,此问通过对 CART 决策树模型和逻辑回归模型进行对比分析，检验其分类结果是否相同,如果相同则更加充分表明预测结果的准确性。CART 决策树与逻辑回归模型已在前文中给出。

根据表单 3，我们已知 A1-A8 文物的表面风化情况和文物的各种化学成分，因此在训练 CART 决策树和逻辑回归模型时我们将风化程度与各种化学成分均作为决策树和逻辑回归模型的特征,考虑到表单 2 中文物采样点含风化层中严重风化点的采样，而严重风化点更能反映风化过程中的化学成分变化,因此将该样本风化程度设置为 2，一般风化表面的采样设置为 1，未风化区域的采样的风化程度置为 0。选取训练集的占比为 80%分别对决策树和逻辑回归模型进行训练。

CART 决策树模型分类规则与问题二相同，逻辑回归得到的回归系数如下

表 11 逻辑回归的回归系数

成分	SiO2	Na2O	K2O	CaO	MgO	Al2O3	Fe2O3	CuO	PbO	BaO	P2O5	SrO	SnO2	SO2	风化
系数	-0.15	0.03	-0.23	-0.10	-0.02	-0.0	-0.02	-0.06	0.47	0.14	-0.03	0.00	0.00	0.00	-0.00

通过计算测试集的分类准确度，我们求得决策树与逻辑回归的准确度均 100%，说明两种模型都具有极佳的分类效果，使用两种模型分别对 A1-A8 文物进行预测，两者得到的预测结果完全相同，分类结果如下：

表 12 文物 A1-A8 分类结果

A1	高钾
A2	铅钡
A3	铅钡
A4	铅钡
A5	铅钡

A6	高钾
A7	高钾
A8	铅钡

无论是采用决策树模型还是逻辑回归模型都得到相同的结果，说明预测效果十分优秀。

5.3.2 敏感度分析

鉴于决策树模型仅根据氧化铅含量这一个特征进行分类，在氧化铅含量为 5.46 时预测结果会发生突变，敏感性较差，因此我们主要选择逻辑回归模型进行敏感性分析。

对于逻辑回归模型采用两个维度对敏感度进行分析。

- 基于偏导数的敏感度分析

敏感度分析可以定义为研究每个特征的微小变化对输出的影响，基于上文建立的逻辑回归模型，提出对回归函数的每一个特征变量分别求偏导的方式，通过偏导数值的大小判断敏感性，研究每种化学成分的变化对输出的影响程度，进而从多个特征中逐一找出对输出影响较大的敏感性特征作为分类着重考虑的因素

敏感性的计算公式推导：

由逻辑回归的判别函数 $g(z)=\frac{1}{1+e^{\omega x+b}}$，令

$$e^{\omega_2\cdot x_2+\omega_3\cdot x_3+\cdots+\omega_n\cdot x_n+b}=m \tag{11}$$

则

$$m=e^{\omega x+b-\omega_1\cdot x_1}=\frac{e^{\omega x+b}}{e^{\omega_1\cdot x_1}} \tag{12}$$

即

$$m\cdot e^{\omega_1\cdot x_1}=e^{\omega x+b} \tag{13}$$

于是对 x_1 求偏导得：

$$\frac{\partial y}{\partial x_1}=\frac{\partial(\frac{1}{1+e^{\omega x+b}})}{\partial x_1}=\frac{\partial(\frac{1}{1+me^{\omega_1 x_1}})}{\partial x_1}=(1+me^{\omega_1 x_1})^{-2}\cdot m\cdot\omega_1\cdot e^{\omega_1 x_1} \tag{14}$$

将 m 带回公式(14)

$$\frac{\partial y}{\partial x_1}=\omega_1\cdot e^{\omega x+b}\cdot(1+e^{\omega x+b})^{-2} \tag{15}$$

据此得到每个特征的敏感性计算公式为

$$Sensitivity=\frac{1}{n}\sum_{j=1}^{n}\left|\frac{\partial y}{\partial x_i^j}\right| \tag{16}$$

经上述公式计算得到各指标的敏感度如下：

表 13 逻辑回归模型各化学成分的敏感度

种类	敏感度
氧化铅(PbO)	0.0133

氧化钾(K2O)	0.0069
二氧化硅(SiO2)	0.0043
氧化钡(BaO)	0.0029
氧化钙(CaO)	0.0025
氧化铝(Al2O3)	0.0023
氧化铜(CuO)	0.0017
五氧化二磷(P2O5)	0.0016
氧化铁(Fe2O3)	0.0013
氧化钠(Na2O)	0.0004
氧化镁(MgO)	0.0003
氧化锶(SrO)	0.0002
二氧化硫(SO2)	0.0001
风化与否	0

由上表可知，所有变量的 *Sensitivity* 值都很小，说明逻辑回归模型的分类结果随着化学成分的微小变化对分类结果的影响较小，效果较好。样本中氧化铅的含量对分类结果的影响性最大，其次是氧化钾，鉴于铅钡玻璃即为是烧制过程中加入了氧化铅含量高的铅矿石制成，而高钾玻璃是以含钾量高的草木灰、硝石等制成，因此分类结果对氧化铅含量与氧化钾含量敏感是十分合理的。

- 基于 ROC 曲线的敏感度分析

ROC 曲线即受试者工作特征曲线(Receiver Operating Characteristic Curve)，是以预测结果的每一个值作为可能的判断阈值，由此计算得到相应的灵敏度的特异度，以假阳性率为横坐标，以真阳性率即灵敏度为纵坐标绘制而成，其曲线下面积 AUC(Area Under Curve)的大小作为模型预测准确度的衡量指标，其取值范围为[0,1]，值越大表示模型的判断力越强，理想情况是建立的模型预测文物种类与文物的实际种类完全吻合，此时 AUC 的值为 1。

计算得到逻辑回归和 CART 决策树的 ROC 曲线均如下：

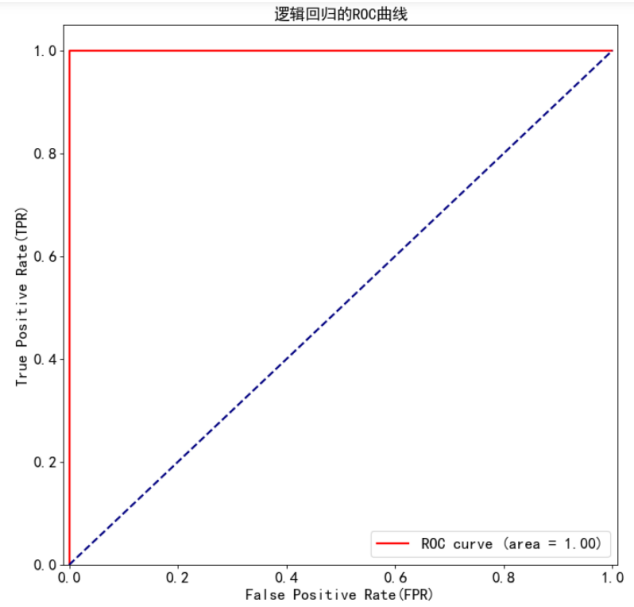


图 17 逻辑回归的 ROC 曲线

上图 ROC 曲线中，敏感性 TPR 恒为 1，中间部分不随 FPR 的变化而变化，且 AUC 的值为 1，说明模型敏感性好，分类效果极佳。

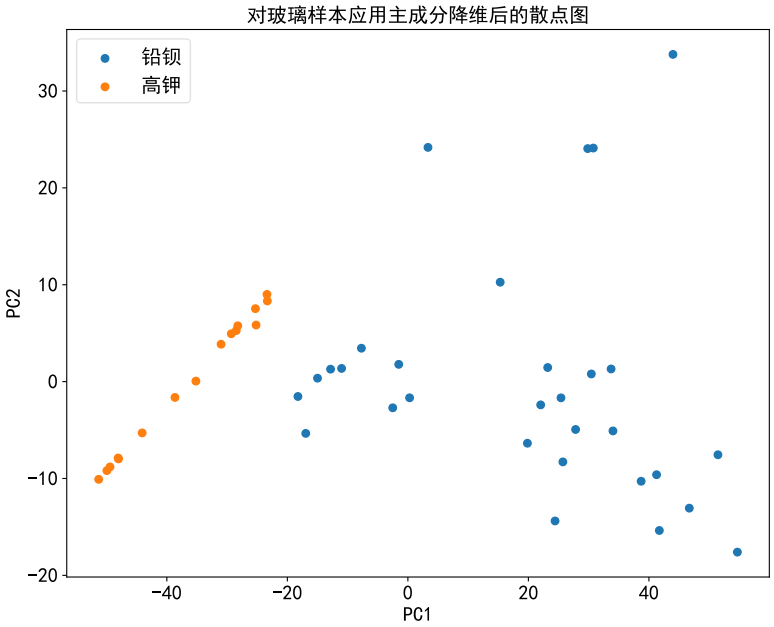


图 18 对玻璃样本应用主成分降维后的散点图

图说两种类别的样本组间距离较大，说明敏感性较强。

5.4 问题四的模型建立与求解

Part 1：为了分析化学成分之间的关联关系，我们从两个角度研究了此问题：其一是化学成分两两之间的关联关系，用 *spearman* 相关系数绘制热力图直观显示；其二是多个化学成分之间的关系，选定若干化学成分用 *VIF* 方差膨胀因子平均值刻画出其共线性程度，将 *VIF* 平均值最大的化学成分集合作为关联关系最强的多个化学成分，再用因子分析研究其余成分之间的相关性强弱。

5.4.1 使用相关系数研究化学成分两两之间关系

首先对数据进行正态分布检测，考虑到高钾组样本数较少，使用 *Shapiro Wilk* 检验，铅钡组样本更多,使用 *Jarque Bera* 检验,根据 p 值大小得出铅钡玻璃和高钾玻璃大部分化学成分数据均不满足正态分布假设，不能使用 *Pearson* 相关系数分析，因此此处使用 *spearman* 相关系数研究变量相关性。

Step 2 对 14 种化学成分计算 *spearman* 相关系数及 p 值，设置置信度为 99%，筛选掉相关性不显著的变量组合后做出热力图如下：

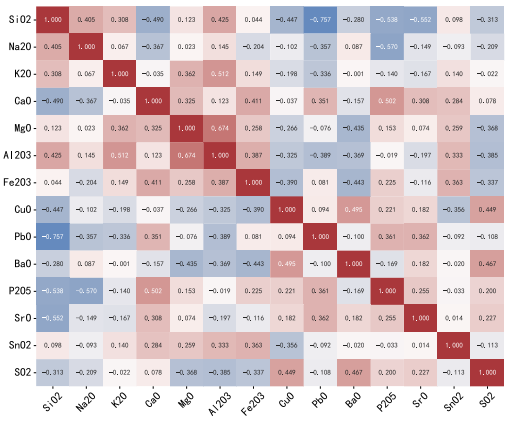


图 19 铅钡玻璃相关系数热力图

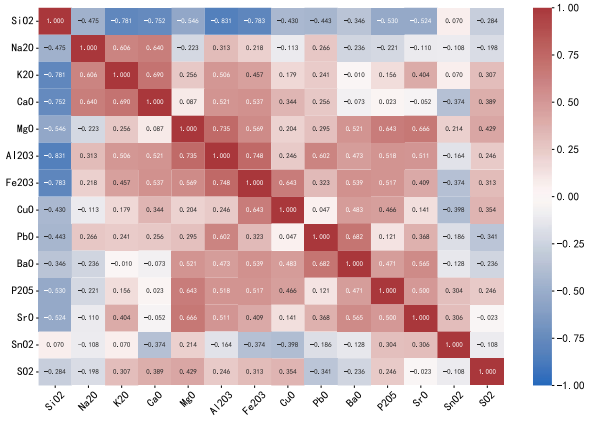


图 20 高钾玻璃相关系数热力图

◆ 差异性分析

对于铅钡玻璃：观察到冷暖色块较为均衡且颜色较浅，说明化学成分两两间正负相关个数较均衡且相关性不明显，而高钾玻璃热力图中暖色块居多，证明化学成分间正相关个数占大多数，其中正负相关性较强的部分集中在与铅钡玻璃同样的元素上。

铅钡玻璃相关性最高的氧化物为 Al_2O_3 和 MgO，相关系数为 0.674，其次是 K_2O 和 Al_2O_3，相关系数为 0.512，其余氧化物含量两两之间的相关系数不足 0.5。

高钾玻璃相关性最高的氧化物为 Fe_2O_3 与 Al_2O_3，相关系数为 0.748，其余大部分元素的相关性都比铅钡玻璃高。

5.4.2 使用 *VIF* 方差膨胀因子分析相关性最强的多个因素：

VIF 是衡量变量间相关程度的指标，常用来检验多重共线性，对于自变量 $X_1, X_2, \cdots, X_n$，分别以 X_i 为因变量，以除 X_i 以外的各自变量为自变量，建立线性回归模型，回归模型的决定系数记为 R_i^2，则方差膨胀因子

$$VIF = \frac{1}{1-R_i^2} \tag{17}$$

一般的，当 *VIF* 的最大值大于 10 或平均值大于 1 的时候可认为存在共线性，*VIF* 越大表示共线性越严重，也就是这几个自变量具有很强的相关关系。本题中我们将 14 中化学成分作为自变量集合，每次从自变量集合中抽取 2-7 个自变量计算 *VIF* 的平均值，遍历每种情况，将 *VIF* 最大的化学成分集合视为关联关系最紧密的化学成分。

通过计算得出：高钾玻璃中化学成分关联关系最紧密的是 K_2O、CaO、Al_2O_3、Fe_2O_3、P_2O_5、SrO，其 *VIF* 的平均值高达 39.19，可见其关联关系非常紧密。铅钡玻璃中化学成分关联最紧密的是 PbO、BaO、CaO、CuO、SO_2、SrO，其 *VIF* 的平均值为 4.43。

对高钾玻璃与铅钡玻璃强相关的多个变量进行差异性分析，可以看出高钾玻璃的钾、钙、铝、铁、磷、锶元素之间的相关性较强，查阅相关资料可知，Al^{3+} 和 Fe^{2+} 可以替代 $[SiO_4]$ 四面体中 Si^{4+}，因此相关性较强；对于铅钡玻璃，由于制作过程中加入了铅矿石作为助熔剂，因此其铅与钡元素具有强相关性。由于玻璃炼制时都需要加入石灰石作为稳定剂，而石灰石煅烧后转化为氧化钙，因此无论是铅钡玻璃还是高钾玻璃其主要相关成分都有钙元素。

5.4.3 使用因子分析探究其余元素相关性强弱：

➢ 因子分析概述

因子分析法是运用降维的思想从众多相关变量中抽取若干共同元素，使复杂的数

据得以简化，以最少的信息丢失把一些具有错综复杂关系的变量归结为少数几个综合因子，通过探究众多变量之间的内部依赖关系找出那些真正相关的变量。

原始变量：$x_1, x_2, \cdots, x_m$，公共因子：$f_1, f_2, \cdots, f_n$，各因子与原始变量之间的关系可以表示成：

$$\begin{cases} x_1 = a_{11} f_1 + a_{12} f_2 + \cdots + a_{1n} f_n + \varepsilon_1 \\ x_2 = a_{21} f_1 + a_{22} f_2 + \cdots + a_{2n} f_n + \varepsilon_2 \\ \cdots \\ x_m = a_{m1} f_1 + a_{m2} f_2 + \cdots + a_{mn} f_n + \varepsilon_n \end{cases} \tag{18}$$

写成矩阵形式

$$X = AF + \varepsilon \tag{19}$$

其中，$F = (f_1, f_2, \cdots, f_m)^T$ 为公因子向量，A 为公因子载荷稀疏矩阵，ε 为残差向量。通过最小残差法、最大似然法或主成分法可以求解出因子载荷矩阵 A。为了更好地解释每一个公共因子的实际意义，并且分析构成因子的变量之间的相互关系，可以通过正交旋转的方法得到新的公共因子和因子载荷矩阵。

通过对因子载荷矩阵的分析，得到表单 2 中各种氧化物变量与某一因子的联系系数，其绝对值越大则说明该因子与变量关系越近，即这些氧化物变量之间的相关性较强。

➢ 因子分析法求解：

下文以铅钡玻璃文物样品为例采用因子分析探究其化学成分之间的关联关系。

首先将上一步计算 *VIF* 均值得到的主要氧化物变量去除，对新得到的数据进行 *Bartlett's* 球状检验，原假设 H_0 为：相关系数矩阵为单位矩阵。计算卡方统计量得到 p 值为 6.08×10^{-30}，远小于 0.05，因此在 95%的置信水平上拒绝原假设，认为数据适合做因子分析。

为了确定公共因子的个数，通过碎石检验观察特征值的变化选择最佳的因子数量，得到的碎石图如下：

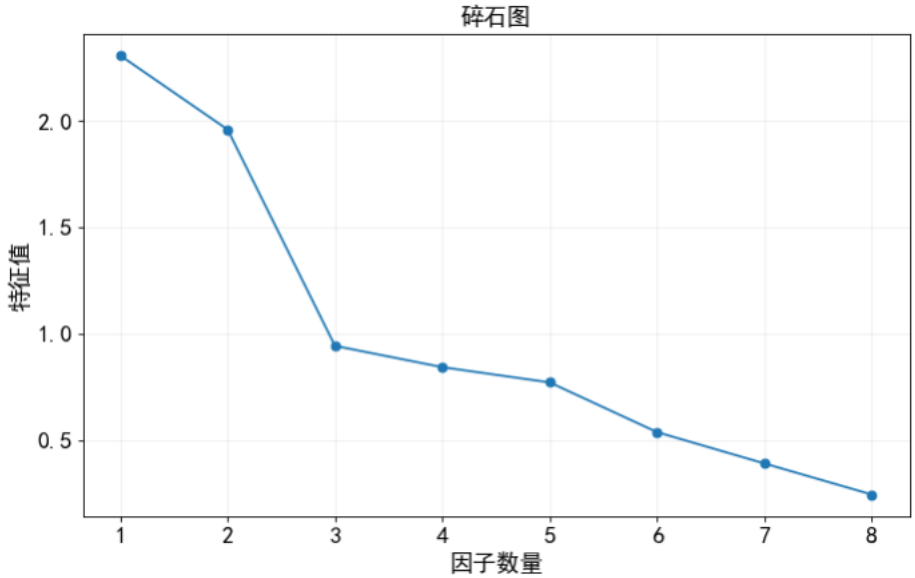


图 21 铅钡玻璃因子分析碎石图

前四个特征值分别为 2.31、1.96、0.94、0.84，结合图像可见从因子数量为 3 开始特征值的变化放缓，因此选择因子数量为 3 时对载荷矩阵进行分析。

选用最小残差法与正交变换得到因子载荷矩阵，将其绘制成热力图如下：

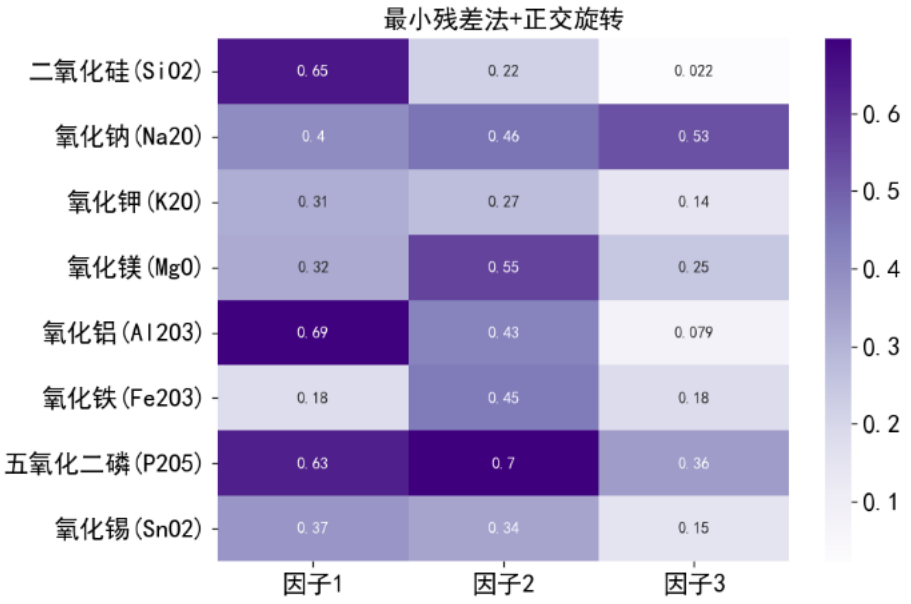


图 22 铅钡玻璃因子载荷热力图

由上图可知，因子 f_1 体现了 SiO_2、Al_2O_3、P_2O_5 的成分比重，说明这些氧化物具有较强的相关性；因子 f_2 主要体现了 P_2O_5 与 MgO 之间具有一定的关联关系；由于因子 f_3 仅有 Na_2O 载荷系数较大，因此可以看出其与其他化学成分相关性较弱。

我们对高钾玻璃进行同样的步骤得到如下结果：第一个因子中 SiO_2、MgO、CuO、PbO、BaO 相关性较强，第二个因子中 Na_2O、SO_2 具有一定的相关性，其余因子的对应的载荷系数较为均匀，无法明显反应氧化物之间的相关关系。

差异性分析：

由于已经用 *VIF* 方差膨胀因子筛选出了最相关的元素组合，而因子分析筛选出了较弱相关的元素组合，因此剩下的氧化物与其他化学成分可以视作没有相关关系。对比铅钡玻璃和高钾玻璃可以发现，铅钡玻璃剩余的不相关成分为 Na_2O、K_2O、MgO、Fe_2O_3、SnO_2，而高钾玻璃的不相关成分为 SnO_2。对上述结果的合理性进行解释，由于铅钡玻璃在烧制过程中并没有加入草木灰作为助熔剂，而草木灰中含有的大量成分即为 Na 与 K，因此这两种成分不相关是显而易见的，其次 Fe_2O_3 中的 Fe^{3+} 显红色，常用于为玻璃上红色，属于着色剂成分，是否添加取决于对成品颜色的需求，因此与铅钡玻璃的其余化学成分无关。通过对表单 2 的观察，文物中 SnO_2 的含量绝大部分为 0%，仅有一个样本超过 0.5%，说明 SnO_2 极有可能是杂质成分，因此与铅钡玻璃和高钾玻璃的其余化学成分均无关。

六、 模型的评价、改进与推广

6.1 模型的优点

★ 优点 1. 紧扣数据分布特征，模型思想相互联系：

对问题一，在第一个任务中我们紧抓风化与否这个满足 0-1 分布的离散型变量分布特征，建立二元逻辑回归模型；在任务二中我们对其进行优化，建立多元逻辑回归模型；在问题三中我们又抓住问题的二分类特征，沿用逻辑回归思想。整个问题求解过程紧扣数据特征，求解方法互相联系，思想浑然一体。

★ 优点 2：特征选择突破传统：

问题一的相关性分析中，Lasso 特征选择在 14 个化学成分与风化状态的相关性分析中起到了筛选重要变量，简化模型的作用。它避免传统多元回归分析在小样本上的多重共线性和异方差等局限性，从而我们能更客观准确地分析多个化学成分与风化状态的关系。同时，对比起逐步回归具有步骤少、速度快的特点。先进性特征选择，再利用多元回归分析方法，不仅避免了多重共线性等问题，而且对相关性提取更准确。

★ 优点 3. 查阅文献对统计规律进行验证，模型合理可靠：

针对问题一研究风化与否的化学成分规律，我们结合古玻璃研究文献对结果的相关性正负及相对大小进行验证，说明了模型结果的合理性；针对问题二的亚类划分，我们查阅古玻璃化学成分分类的相关论文，将分类结果与文献中的结果进行比对，得到较高的重合度，说明我们亚类划分模型的可靠性。

★ 优点 4 一题多法，利于结果最优：

问题一中我们分别对有无进行特征选择、使用线性回归或逻辑回归模型分别进行尝试，对比我们的基于 Lasso 回归特征选择的多元逻辑回归模型效果，有利于结果最优；问题三中我们利用分别建立 CART 决策树模型和逻辑回归分类模型，结果相互印证；问题四中先使用 VIF 检测两组变量间的相关性，再对筛选出的结果进行因子分析，有利于模型最优，具有独创性。

6.2 模型的缺点

- ✧ 缺点 1：玻璃的颜色与所使用的染色化学成分有关，本模型只在第一问考虑了这点，但在后几问的分析中没有考虑颜色和染色剂的关联关系，因此在进行分析时没有针对性地剔除掉不同颜色的文物对应的染色化学成分，从而对分析结果造成了一定程度的干扰。
- ✧ 缺点 2：亚类划分时未能较好地消除风化对样本化学成分的影响。

6.3 模型的改进

- ➢ 改进 1：考虑文物的颜色和染色化学成分的对应关系。由于染色剂在玻璃文物的化学成分中属于次要变量，因此，可以根据文物的颜色确定染色化学成分的组成和含量，在分析与染色剂不相关的问题时，文物中对应的化学成分可以减去其用于染色的含量，从而将染色剂对模型的干扰降至最低，使模型更加精准。
- ➢ 改进 2：定量考虑风化对样本点化学成分的影响，将问题一中风化数据点的预测数据应用在亚类划分模型上，得出更准确的亚类划分结果。
- ➢ 改进 3：可以对样品中主要元素及次要元素分别聚类，使得不同化学主要成分的玻璃得到更全面的比较。

6.4 模型的推广

- ✦ 推广 1：问题一中基于 lasso 回归特征选择的多元逻辑回归模型避免了异方差和多重共线性的影响，可以作为逐步回归的替代方法，作为研究多变量与单变量间相关性的首步骤。例如各个健康指标与人体血管健康的相关性；各个环境指标与居住舒适度的相关性等。
- ✦ 推广 2：聚类分析为古玻璃中颜色纹饰不同、地域不同、化学组成成分不同的玻璃分类提供了科学的方法。可以将其运用在对古陶瓷、古文物的类别划分上。
- ✦ 推广 3：因子分析法可以客服变量选择的人为性，找寻一组变量间的相互关

系，故可以运用在研究组内变量相关系数上。例如研究中草药成分间的相互关系。

七、 参考文献

[1] 董俊卿,李青会,干福熹,胡永庆,程永建,蒋宏杰.一批河南出土东周至宋代玻璃器的无损分析[J].中国材料进展,2012,31(11):9-15.

[2] 伏修锋,干福熹.基于多元统计分析方法对一批中国南方和西南地区的古玻璃成分的研究[J].文物保护与考古科学,2006(04):6-13.DOI:10.16334/j.cnki.cn31-1652/k.2006.04.002.

[3] 王凌妍,张鑫雨,许胜楠,王禹力,甄志龙.逻辑回归的敏感性分析及在特征选择中的应用[J].信息记录材料,2022,23(07):30-33.DOI:10.16009/j.cnki.cn13-1295/tq.2022.07.051.

[4] 李清临,徐承泰,凌雪,姚政权.一批金元时期古玻璃的 EDXRF 探针无损分析[J].光谱学与光谱分析,2011,31(07):1960-1963.

[5] 史美光,王礼云.世界古玻璃的化学成份[J].玻璃,1987(04):5-12+16.

[6] 吴宗道,周福征,史美光.几个古玻璃的显微形貌、成分及其风化的初步研究[J].电子显微学报,1986(04):65-71.

[7] 祁俊生,徐辉碧,周井炎,陆晓华,杨祥良,管竞环.植物类中药中微量元素的因子分析和聚类分析[J].分析化学,1998(11):1309-1314.

[8] 世界古代玻璃化学成分综论 http://www.gxmuseum.cn/a/science/31/2011/1073.html

附录

附录 1

介绍：利用 python 根据表单 1 数据生成哑变量并进行逻辑回归

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression as LR
df = pd.read_excel('附件.xlsx', sheet_name='表单 1')
df.head()
df.shape
df = df.dropna()
df.insert(df.shape[1], '红', 0)
df.insert(df.shape[1], '蓝', 0)
df.insert(df.shape[1], '绿', 0)
df.loc[df['颜色'] == '浅蓝', '蓝'] = 1
df.loc[df['颜色'] == '深蓝', '蓝'] = 2

df.loc[df['颜色'] == '浅绿', '绿'] = 1
df.loc[df['颜色'] == '深绿', '绿'] = 2

df.loc[df['颜色'] == '蓝绿', '蓝'] = 1
df.loc[df['颜色'] == '蓝绿', '绿'] = 1

df.loc[df['颜色'] == '紫', '蓝'] = 1
df.loc[df['颜色'] == '紫', '红'] = 1


df.loc[df['颜色'] == '黑', '蓝'] = 1
df.loc[df['颜色'] == '黑', '红'] = 1
df.loc[df['颜色'] == '黑', '绿'] = 1
df.head()
df1 = df.drop('颜色', axis=1)
df2 = pd.get_dummies(df1)
df2 = df2.drop(['表面风化_无风化'], axis=1)
df2.head()
df2.to_excel('表单 1 虚拟变量.xlsx', index=False)
X = df2.iloc[:, [*range(1, 9)]]
y = df2.iloc[:, -1]


model = LR(random_state=0)
model.fit(X, y)
```

```
    model.coef_
    # 逻辑回归系数如下
    pd.DataFrame([df2.columns[1:-1], *model.coef_])
```

附录 2

介绍：利用 python 针对问题一进行相关系数分析

```
    import seaborn as sns
import numpy as np
import pandas as pd
from sklearn.linear_model import Ridge, LinearRegression, Lasso
from sklearn.model_selection import train_test_split as TTS
import matplotlib.pyplot as plt
from sklearn.linear_model import LassoCV
import statsmodels.api as sm
import statsmodels.iolib.summary2
from matplotlib import pyplot as plt
from scipy import stats
from statsmodels.stats.diagnostic import het_white  # 怀特检验
from statsmodels.stats.outliers_influence import variance_inflation_factor
# 计算VIF方差
# 正常显示汉字和负号
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False
df = pd.read_excel('表单1虚拟变量.xlsx')
data = df.iloc[:, 1:]
### 正态分布检验
#### Jarque-Bera检验(大样本 $ n > 30 $)
def JB(X):
    result = stats.jarque_bera(X)
    return pd.Series(result, index=['JB', 'pvalue']).round(4)
JB_test = data.apply(JB)
JB_test
# 显著性检验，验证该随机变量是否符合正态分布
# 将p值与0.05比较，小于则可拒绝原假设，否则不能拒绝原假设
JB_test.iloc[1,:] > 0.05
# ## 斯皮尔曼相关系数
# ### 1.适用条件
# * 分布严重非正态
# * 变量非连续
# * 异常值影响大
# ### 2.斯皮尔曼相关系数矩阵
data.corr('spearman')
```

```
#### 3.P 值矩阵
def PvalueGetSpearman(X, Y):
    return stats.spearmanr(X, Y)[1].round(4)
data.corr(method=PvalueGetSpearman)
# #### 假设检验
# 1. 小样本(n $\leqslant$ 30): 查临界值表(看相关系数矩阵)
# 2. 大样本: 直接看 P 值矩阵
data.corr(method=PvalueGetSpearman).iloc[:,-1]
data.corr(method=PvalueGetSpearman).iloc[-1,:] > 0.01
pd.Series(data.corr(method=PvalueGetSpearman).iloc[0,:] > 0.01).count()
corr = data.corr(method=PvalueGetSpearman) > 0.01
mask = np.zeros_like(corr)
mask[np.triu_indices_from(mask)] = True
sns.heatmap(corr, mask=mask, annot=True, square=True, cmap='YlGnBu',
cbar=False)

df = pd.read_excel('1.2 铅钡.xlsx')
df.columns
## 原始的
# 高钾
origin1=df.iloc[:,3:-1]

X = origin1.iloc[:,1:]
y = origin1.iloc[:,0]
x = sm.add_constant(np.array(X))  # 添加常数项
reg = sm.OLS(y, x)
results = reg.fit()
print(results.summary())
# 铅钡
df = pd.read_excel('1.2 铅钡.xlsx')
origin1=df.iloc[:,3:-1]

X = origin1.iloc[:,1:]
y = origin1.iloc[:,0]
x = sm.add_constant(np.array(X))  # 添加常数项
reg = sm.OLS(y, x)
results = reg.fit()
print(results.summary())
plt.rcParams["font.sans-serif"] = ["SimHei"]
plt.rcParams["axes.unicode_minus"] = False
%matplotlib inline

res = results.resid  # 从 OLS 回归模型中获取残差
fitted = results.fittedvalues  # 从 OLS 回归模型中获取拟合值
```

```
plt.scatter(fitted, res)
plt.xlabel("Fitted values")
plt.ylabel("Residuals")
plt.title("残差与拟合值的散点图")
plt.show()
## 铅钡模型：挑出样本点做回归
data=df.iloc[:,3:-1].drop([ '氧化镁(MgO)','氧化钾(K2O)', '氧化钙(CaO)','氧化锶(SrO)', '氧化锡(SnO2)'],axis=1)
data.head()
X = data.iloc[:,1:]
y = data.iloc[:,0]
x = sm.add_constant(np.array(X))  # 添加常数项
reg = sm.OLS(y, x)
results = reg.fit()
print(results.summary())
## 残差图
plt.rcParams["font.sans-serif"] = ["SimHei"]
plt.rcParams["axes.unicode_minus"] = False
%matplotlib inline

res = results.resid  # 从OLS回归模型中获取残差
fitted = results.fittedvalues  # 从OLS回归模型中获取拟合值
plt.scatter(fitted, res)
plt.xlabel("Fitted values")
plt.ylabel("Residuals")
plt.title("残差与拟合值的散点图")
plt.show()
lm_statistic, lm_pval = het_white(results.resid, reg.exog)[:2]
print("原假设 H0：不存在异方差")
print("卡方检验值：", lm_statistic)
print("p 值：", lm_pval)
# 如果p值小于0.05，在95%的置信水平上拒绝H0，
# 此处扰动项不存在异方差
## 高钾玻璃
df = pd.read_excel('1.2 高钾.xlsx')
df.head()
data=df[[ '风化与否','二氧化硅(SiO2)','氧化钾(K2O)', '氧化铝(Al2O3)','氧化铜(CuO)']]
data.head()
X = data.iloc[:,1:]
y = data.iloc[:,0]
x = sm.add_constant(np.array(X))  # 添加常数项
reg = sm.OLS(y, x)
results = reg.fit()
```

```
print(results.summary())
plt.rcParams["font.sans-serif"] = ["SimHei"]
plt.rcParams["axes.unicode_minus"] = False
%matplotlib inline

res = results.resid  # 从 OLS 回归模型中获取残差
fitted = results.fittedvalues  # 从 OLS 回归模型中获取拟合值
plt.scatter(fitted, res)
plt.xlabel("Fitted values")
plt.ylabel("Residuals")
plt.title("残差与拟合值的散点图")
plt.show()
lm_statistic, lm_pval = het_white(results.resid, reg.exog)[:2]
print("原假设 H0：不存在异方差")
print("卡方检验值：", lm_statistic)
    print("p 值：", lm_pval)
```

附录 3

介绍：利用 python 画饼状图

```
    from statsmodels.stats.descriptivestats import describe
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
large = 22
med = 16
small = 12
params = {
    "font.sans-serif":"Simhei",
    "axes.titlesize": 22,
    "legend.fontsize": 16,
    "figure.figsize": (10, 8),
    "figure.dpi":80,
    "axes.labelsize": 16,
    "axes.titlesize": 16,
    "xtick.labelsize": 16,
    "ytick.labelsize": 16,
    "figure.titlesize": 22,
```

```
    "axes.unicode_minus": False
}
plt.rcParams.update(params)
pd.set_option('display.precision', 2)
# pd.reset_option('display.precision') # 还原精度设置
plt.style.use('seaborn')
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False
df1 = pd.read_excel('1.2 铅钡.xlsx')
data1 = df1.iloc[:, 4:]

X = data1.iloc[:,1:]
y = df1.iloc[:,3]

data1.describe()

# 风化铅钡
X1 = data1.iloc[(y>0).values]
X1.describe()

# 未风化铅钡
X2 = data1.iloc[(y==0).values]
X2.describe()
def plot_pie(data, explode, title, clr=plt.cm.tab20.colors):
    # 绘制图像
    fig, ax = plt.subplots(figsize=(7, 7), dpi=80)
    wedges, texts, autotexts = ax.pie(
        x=data,
        autopct="",
        colors=clr,  # 图形的颜色
        startangle=150,  # 第一瓣扇叶从什么角度开始，与水平线的夹角
        explode=explode,  # 扇叶与扇叶之间的距离
    )
    # 装饰图像
    categories = data.index
    ax.legend(
        categories,  # 输入数据
        loc="center left",
        bbox_to_anchor=(1, 0, 0.5, 1)
    )
    ax.set_title(title)
    plt.setp(
        autotexts, size=12, weight=700, color="w"
    )  # 设置某个对象(Artist)的属性(Property)， weight 表示加粗
```

```
    plt.savefig('./1.2 饼状图/' + title +'.svg', bbox_inches='tight')
    plt.show()

dt = X1.mean()[:-1]
dt.sort_values(ascending=False, inplace=True)
explode = [0, 0.05, 0.05, 0, 0, 0, 0, 0, 0,
           0, 0, 0, 0, 0]  # 用于准备突出去强调的图，表示扇叶与扇叶之间的距离
title = "风化铅钡各化学成分含量的饼状图"


plot_pie(dt, explode, title, )
clr_df = plt.cm.tab20.colors
clr = [clr_df[i % len(clr_df)] for i in range(dt.shape[0])]

dt2 = pd.DataFrame([X2.columns[:-1], clr, X2.mean()[:-1]]).T

dt2.sort_values(by=2, ascending=False, inplace=True)
dt2.index = dt2[0]

plot_pie(dt2[2], explode, '未风化铅钡各化学成分含量的饼状图', clr=dt2[1])
df2 = pd.read_excel('1.2 高钾.xlsx')
data2 = df2.iloc[:, 4:]

X = data2.iloc[:,1:]
y = df2.iloc[:,3]
# 风化高钾
X3 = data2.iloc[(y>0).values]
X3.describe()

# 未风化高钾
X4 = data2.iloc[(y==0).values]
X4.describe()
explode = [0, 0.05, 0.05, 0, 0, 0, 0, 0, 0,
           0, 0, 0, 0, 0]  # 用于准备突出去强调的图，表示扇叶与扇叶之间的距离
title = "风化高钾各化学成分含量的饼状图"

clr_df = plt.cm.tab20.colors
clr = [clr_df[i % len(clr_df)] for i in range(dt.shape[0])]

dt2 = pd.DataFrame([X3.columns[:-1], clr, X3.mean()[:-1]]).T

dt2.sort_values(by=2, ascending=False, inplace=True)
dt2.index = dt2[0]
```

```
plot_pie(dt2[2], explode, title, clr=dt2[1])
explode = [0, 0.05, 0.05, 0, 0, 0, 0, 0, 0,
           0, 0, 0, 0, 0]  # 用于准备突出去强调的图，表示扇叶与扇叶之间的距离
title = "未风化高钾各化学成分含量的饼状图"

clr_df = plt.cm.tab20.colors
clr = [clr_df[i % len(clr_df)] for i in range(dt.shape[0])]

dt2 = pd.DataFrame([X4.columns[:-1], clr, X4.mean()[:-1]]).T

dt2.sort_values(by=2, ascending=False, inplace=True)
dt2.index = dt2[0]

    plot_pie(dt2[2], explode, title, clr=dt2[1])
```

附录 4

介绍：利用 python 对数据进行描述性统计

```
    from statsmodels.stats.descriptivestats import describe
    import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    pd.set_option('display.precision', 2)
    # pd.reset_option('display.precision') # 还原精度设置
    plt.style.use('seaborn')
    plt.rcParams['font.sans-serif'] = ['SimHei']
    plt.rcParams['axes.unicode_minus'] = False
    df1 = pd.read_excel('1.2 铅钡.xlsx')
    data1 = df1.iloc[:, 4:]

    X = data1.iloc[:,1:]
    y = df1.iloc[:,3]
    # 铅钡
    data1.describe()
    data1.skew()
    from scipy import stats
    stats.kurtosis(data1, fisher=False)
    # 风化铅钡
    X1 = data1.iloc[(y>0).values]
    X1.describe()
```

```
    X1.skew()
    stats.kurtosis(X1, fisher=False)
    # 未风化铅钡
    X2 = data1.iloc[(y==0).values]
    X2.describe()
    X2.skew()
    stats.kurtosis(X2, fisher=False)
    # 铅钡组未风化均值 比 风化后均值
    rate1 = X2.mean()/X1.mean()
    rate1
    df1
    dc = df1.iloc[(y>0).values,[1,*range(4,df1.shape[1])]].copy()

    lb1 = X1*rate1*0.95
    ub1 = X1*rate1*1.05

    lb1 = round(lb1, 2)
    ub1 = round(ub1, 2)

    ls = []
    for i in range(lb1.shape[0]):
        l = []
        for j in range(lb1.shape[1]):
            l.append((lb1.iloc[i,j], ub1.iloc[i,j]))
        ls.append(l)

    pd.DataFrame(ls, index=dc['文物采样点'],
columns=X1.columns).to_excel('1.1 铅钡风化前预测.xlsx')
    df2 = pd.read_excel('1.2 高钾.xlsx')
    data2 = df2.iloc[:, 4:]

    X = data2.iloc[:,1:]
    y = df2.iloc[:,3]
    # 高钾
    data2.describe()
    data2.skew()
    stats.kurtosis(data2, fisher=False)
    # 风化高钾
    X3 = data2.iloc[(y>0).values]
    X3.describe()
    X3.skew()
    stats.kurtosis(X3, fisher=False)
    # 未风化高钾
    X4 = data2.iloc[(y==0).values]
```

```
    X4.describe()
    X4.skew()
    stats.kurtosis(X4, fisher=False)
    # 高钾组未风化均值 比 风化后均值
    rate2 = X4.mean()/X3.mean()
    rate2
    dc = df2.iloc[(y>0).values,[1,*range(4,df2.shape[1])]].copy()
    lb2 = X3*rate2*0.95
    ub2 = X3*rate2*1.05

    lb2.fillna(0, inplace=True)
    ub2.fillna(0, inplace=True)

    lb2 = round(lb2, 2)
    ub2 = round(ub2, 2)

    ls = []
    for i in range(lb2.shape[0]):
        l = []
        for j in range(lb2.shape[1]):
            l.append((lb2.iloc[i,j], ub2.iloc[i,j]))
        ls.append(l)

    pd.DataFrame(ls, index=dc['文物采样点'],
columns=X3.columns).to_excel('1.1 高钾风化前预测.xlsx')
    df = pd.DataFrame(rate1,columns=['铅钡组均值风化前后比值'])
    df['高钾组均值风化前后比值'] = rate2.fillna(0)
    df_new = pd.DataFrame(df.values.T, index=df.columns, columns=df.index)
    df_new
    df_new.to_excel('./1.2 预测/比值.xlsx')
    # 描述性统计
    def descriptive_statistics(df):
        skew_list = []
        kurt_list = []
        description = df.describe()
        for i in df.iteritems():
            skew_list.append(i[1].skew())
            # pandas 自带的kurtosis 计算偏度是是无偏的（除了n-1）
            # stats 里面是有偏的（除了n)
            # 课件里MATLAB 的值有偏
            kurt_list.append(stats.kurtosis(i[1], fisher=False))
        description.loc["skew"] = skew_list
        description.loc["kurt"] = kurt_list
        return description
```

```
    descriptive_statistics(X2).to_csv('./描述性统计/铅钡未风化描述性统
计.csv',encoding='utf-8-sig')  # 不加sig中文会乱码
    descriptive_statistics(X1).to_csv('./描述性统计/铅钡风化描述性统
计.csv',encoding='utf-8-sig')
    descriptive_statistics(X3).to_csv('./描述性统计/高钾风化描述性统
计.csv',encoding='utf-8-sig')  # 不加sig中文会乱码
    descriptive_statistics(X4).to_csv('./描述性统计/高钾未风化描述性统
计.csv',encoding='utf-8-sig')
```

附录 5

介绍：利用 python 画统计折线图

```
    # 统计规律可视化2
    import pandas as pd
    import matplotlib.pyplot as plt
    params = {
        "font.sans-serif":"Simhei",
        "figure.figsize": (10, 8),
        "figure.dpi":80,
        "axes.unicode_minus": False,
        "xtick.labelsize": 16,
        "ytick.labelsize": 16,
    }
    plt.rcParams.update(params)
    # 铅钡
    df=pd.read_excel("./1.2 铅钡.xlsx")
    data=df.iloc[:,4:-1]
    data.columns=["SiO2","Na2O","K2O","CaO","MgO","Al2O3","Fe2O3","CuO","Pb
O","BaO","P2O5","SrO","SnO2","SO2"]
    data.head()
    data1=data.iloc[(df.iloc[:,3]==0).values] # 未风化
    data1.index=range(len(data1))
    data2=data.iloc[(df.iloc[:,3]==1).values] # 已风化
    data2.index=range(len(data2))
    data3=data.iloc[(df.iloc[:,3]==2).values] # 严重风化
    data3.index=range(len(data3))

    fig, ax = plt.subplots(figsize=(10, 6),dpi=120,)
```

```
    for i in range(len(data1)-1):
        ax.plot(data1.iloc[i,:],"grey",linestyle="--
",linewidth=1,alpha=0.7)
    ax.plot(data1.iloc[len(data1)-1,:],"grey",label="未风化
",alpha=0.7,linestyle="--",linewidth=1)
    for i in range(len(data2)-1):

ax.plot(data2.iloc[i,:],"orange",linestyle="-.",linewidth=1,alpha=0.7)
    ax.plot(data2.iloc[len(data2)-1,:],"orange",label="已风化
",linestyle="-.",linewidth=1,alpha=0.7)
    for i in range(len(data3)-1):
        ax.plot(data3.iloc[i,:],"red",linestyle="-
",linewidth=1.5,alpha=0.5)
    ax.plot(data3.iloc[len(data3)-1,:],"red",label="严重风化",linestyle="-
",linewidth=1.5,alpha=0.5)

    plt.title("铅钡玻璃有无风化化学成分含量的统计图")
    plt.legend()
    plt.savefig('./可视化/铅钡玻璃有无风化化学成分含量的统计图',encoding='utf-
8-sig')
    plt.show()
    # 高钾
    df=pd.read_excel("./1.2 高钾.xlsx")
    data=df.iloc[:,4:-1]
    data.columns=["SiO2","Na2O","K2O","CaO","MgO","Al2O3","Fe2O3","CuO","Pb
O","BaO","P2O5","SrO","SnO2","SO2"]
    data
    data1=data.iloc[(df.iloc[:,3]==0).values] # 未风化
    data1.index=range(len(data1))
    data2=data.iloc[(df.iloc[:,3]==1).values] # 已风化
    data2.index=range(len(data2))
    fig, ax = plt.subplots(figsize=(10, 6),dpi=120,)
    for i in range(len(data1)-1):
        ax.plot(data1.iloc[i,:],"grey",alpha=0.7,linestyle="--
",linewidth=1.3)
    ax.plot(data1.iloc[len(data1)-1,:],"grey",label="未风化
",alpha=0.7,linestyle="--",linewidth=1.3)
    for i in range(len(data2)-1):

ax.plot(data2.iloc[i,:],"orange",alpha=0.7,linestyle="-.",linewidth=1.3)
    ax.plot(data2.iloc[len(data2)-1,:],"orange",label="已风化
",alpha=0.7,linestyle="-.",linewidth=1.3)
    plt.title("高钾玻璃有无风化化学成分含量的统计图")
    plt.legend()
```

```
    plt.savefig('./可视化/高钾玻璃有无风化化学成分含量的统计图',encoding='utf-8-sig')
    plt.show()
```

附录 6

介绍：利用 python 进行 Lasso 回归完成特征提取

```
    import numpy as np
    import pandas as pd
    from sklearn.linear_model import Lasso
    from sklearn.model_selection import train_test_split as TTS
    import matplotlib.pyplot as plt
    from sklearn.linear_model import LassoCV
    df = pd.read_excel('1.2 铅钡.xlsx')
    data = df.iloc[:, 3:]

    X = data.iloc[:,1:-1]
    y = data.iloc[:,0]

    lasso_ = Lasso(alpha=0.05).fit(X,y)
    pd.DataFrame([data.columns[1:-1],(lasso_.coef_*100).tolist()])


    pd.DataFrame([data.columns[1:-1],(lasso_.coef_*100).tolist()])
    df = pd.read_excel('1.2 高钾.xlsx')
    data = df.iloc[:, 3:]

    X = data.iloc[:,1:-1]
    y = data.iloc[:,0]

    lasso_ = Lasso(alpha=0.04).fit(X, y)


    pd.DataFrame([data.columns[1:-1],(lasso_.coef_*100).tolist()])
```

附录 7

介绍：利用 python 决策树进行玻璃的分类

```
    import numpy as np
    import pandas as pd
    from matplotlib import pyplot as plt
    import seaborn as sns
    from sklearn.model_selection import train_test_split
    from sklearn import tree
    df=pd.read_excel("./表单 2-处理 1.xlsx")
    df=df.fillna(0)
    df=pd.get_dummies(df,columns=["类型"])
    df=df.rename(columns={'类型_铅钡':'铅钡'})
    df
    data=df.iloc[:,2:22]
    target=df["铅钡"]
    random=2
    Xtrain, Xtest, Ytrain, Ytest = train_test_split(data, target,
test_size=0.2,random_state=random)
    clf = tree.DecisionTreeClassifier(
        criterion="gini",
        random_state=random,#随机数种子
        splitter="best",
        max_depth=3,  # 树的最大深度，防止过拟合
        min_samples_leaf=15,  #叶节点所需要的最小样本数
    )
    clf = clf.fit(Xtrain, Ytrain)
    score = clf.score(Xtest, Ytest)
    score
    import graphviz

    feature_name = data.columns

    dot_data = tree.export_graphviz(
        clf,
        feature_names=feature_name,
        class_names=["高钾","铅钡"],
        filled=True,  # 是否填充颜色
        rounded=True,  # 每一块的边框为弧形
    )
    graph = graphviz.Source(dot_data)
    graph
    pd.DataFrame(clf.feature_importances_,index=data.columns)
```

附录 8

介绍：利用 SPSS 进行高钾玻璃的系统聚类

```
GET DATA
  /TYPE=XLSX
  /FILE='C:\Users\19828\Documents\Code\国赛 C 题整理\2.2 高钾.xlsx'
  /SHEET=name 'Sheet1'
  /CELLRANGE=FULL
  /READNAMES=ON
  /DATATYPEMIN PERCENTAGE=95.0
  /HIDDEN IGNORE=YES.
EXECUTE.
DATASET NAME 数据集 1 WINDOW=FRONT.
CLUSTER    二氧化硅 SiO2 氧化钠 Na2O 氧化钾 K2O 氧化钙 CaO 氧化镁 MgO 氧
化铝 Al2O3 氧化铁 Fe2O3 氧化铜 CuO 氧化铅 PbO 氧化钡 BaO 五氧化二磷 P2O5
    氧化锶 SrO 氧化锡 SnO2 二氧化硫 SO2
  /METHOD BAVERAGE
  /MEASURE=SEUCLID
  /ID=文物采样点
  /PRINT SCHEDULE
  /PLOT DENDROGRAM VICICLE.
```

附录 9

介绍：利用 SPSS 进行铅钡玻璃的系统聚类

```
GET DATA
  /TYPE=XLSX
  /FILE='C:\Users\19828\Documents\Code\国赛 C 题整理\2.2 铅钡.xlsx'
  /SHEET=name 'Sheet1'
  /CELLRANGE=FULL
  /READNAMES=ON
  /DATATYPEMIN PERCENTAGE=95.0
  /HIDDEN IGNORE=YES.
EXECUTE.
DATASET NAME 数据集 1 WINDOW=FRONT.
CLUSTER    二氧化硅 SiO2 氧化钠 Na2O 氧化钾 K2O 氧化钙 CaO 氧化镁 MgO 氧
化铝 Al2O3 氧化铁 Fe2O3 氧化铜 CuO 氧化铅 PbO 氧化钡 BaO 五氧化二磷 P2O5
    氧化锶 SrO 氧化锡 SnO2 二氧化硫 SO2
```

```
  /METHOD BAVERAGE
  /MEASURE=SEUCLID
  /ID=文物采样点
  /PRINT SCHEDULE
  /PLOT DENDROGRAM VICICLE.
```

附录 10

介绍：利用 python 分析高钾聚类结果

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

plt.style.use('seaborn')
large = 22
med = 16
small = 12
params = {
    "font.sans-serif": "Simhei",
    "axes.titlesize": 22,
    "legend.fontsize": 16,
    "figure.figsize": (10, 8),
    "figure.dpi": 80,
    "axes.labelsize": 16,
    "axes.titlesize": 16,
    "xtick.labelsize": 16,
    "ytick.labelsize": 16,
    "figure.titlesize": 22,
    "axes.unicode_minus": False
}
plt.rcParams.update(params)

# 高钾
# spss 聚类结果
spss_cluster_label = [[9,11,10,17,18,8,2,15,16],
                      [1,4,13,14,3,5,12,6,7]]

label = []
```

```
    for i in spss_cluster_label:
        ls = []
        for j in i:
            ls.append(j-1)
        label.append(ls)
    df = pd.read_excel('2.2 高钾.xlsx')
    chemi = df.iloc[:,3:]

    chemi.insert(chemi.shape[1], '组类', 0)

    for i in range(len(label)):
        chemi.iloc[label[i],-1] = i

    chemi.head()
    # 每一组类中的文物编号为
    print('不同组类的文物编号如下:\n', [set(df[chemi['组类']==i].文物编
号.sort_values().values) for i in range(len(label))])
    print('不同组类的文物采样点如下:\n', [set(df[chemi['组类']==i].文物采样
点.sort_values().values) for i in range(len(label))])
    df['组类'] = chemi.组类+1
    df.to_excel('2.2 高钾聚类结果.xlsx', index=False)
    for col_id in range(chemi.shape[1]-1):
        fig, axes = plt.subplots(1,2,figsize=(5,4),sharey='row')
        title = '高钾亚类'+chemi.columns[col_id]+'含量箱型图'
        fig.suptitle(title)
        for i in range(len(label)):
            ax = axes[i]
            ax.set_xlabel('第%i 组'%(i+1))
            sns.boxplot(y=chemi[chemi['组类'] ==
i].iloc[:,col_id].sort_values().values,
color=sns.color_palette(n_colors=4)[i], ax=ax, width=0.4)
        plt.savefig('./系统聚类图/'+title+'.svg', bbox_inches='tight')
        plt.show()
    j = 5
    for i in range(len(label)):
        print(chemi[chemi['组类'] == i].iloc[:,j])
    print(chemi.columns[j])
    for i in range(len(label)):
        print(chemi[chemi['组类'] == i].iloc[:,j].min(), chemi[chemi['组类']
== i].iloc[:,j].mean(),
        chemi[chemi['组类'] == i].iloc[:,j].max())
```

附录 11

介绍：利用 python 分析铅钡玻璃的聚类结果

```
    import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns

    plt.style.use('seaborn')
    large = 22
    med = 16
    small = 12
    params = {
        "font.sans-serif": "Simhei",
        "axes.titlesize": 22,
        "legend.fontsize": 16,
        "figure.figsize": (10, 8),
        "figure.dpi": 80,
        "axes.labelsize": 16,
        "axes.titlesize": 16,
        "xtick.labelsize": 16,
        "ytick.labelsize": 16,
        "figure.titlesize": 22,
        "axes.unicode_minus": False
    }
    plt.rcParams.update(params)
    # 铅钡
    # spss 聚类结果
    spss_cluster_label = [[2,10,8,4,6],
                          [3,11],

[19,23,14,15,21,1,5,49,40,37,47,48,26,39,42,30,41,45,24,44,25,29],

[9,46,38,27,28,34,7,33,36,13,31,35,32,43,22,12,17,20,16,18]]

    label = []
    for i in spss_cluster_label:
        ls = []
        for j in i:
            ls.append(j-1)
        label.append(ls)
    df = pd.read_excel('2.2 铅钡.xlsx')
```

```
    chemi = df.iloc[:,3:]

    chemi.insert(chemi.shape[1], '组类', 0)

    for i in range(len(label)):
        chemi.iloc[label[i],-1] = i

    chemi.head()
    # 每一组类中的文物编号为
    print('不同组类的文物编号如下:\n', [set(df[chemi['组类']==i].文物编
号.sort_values().values) for i in range(len(label))])
    print('不同组类的文物采样点如下:\n', [set(df[chemi['组类']==i].文物采样
点.sort_values().values) for i in range(len(label))])
    df['组类'] = chemi.组类+1
    df.to_excel('2.2 铅钡聚类结果.xlsx', index=False)
    for col_id in range(chemi.shape[1]-1):
        fig, axes = plt.subplots(1,4,figsize=(5,4),sharey='row')
        title = '铅钡亚类'+chemi.columns[col_id]+'含量箱型图'
        fig.suptitle(title)
        for i in range(len(label)):
            ax = axes[i]
            ax.set_xlabel('第%i 组'%(i+1))
            sns.boxplot(y=chemi[chemi['组类'] ==
i].iloc[:,col_id].sort_values().values,
color=sns.color_palette(n_colors=4)[i], ax=ax, width=0.4)
        plt.savefig('./系统聚类图/'+title+'.svg', bbox_inches='tight')
        plt.show()
    j = 3
    for i in range(len(label)):
        print(chemi[chemi['组类'] == i].iloc[:,j])
    print(chemi.columns[j])
    for i in range(len(label)):
        print(chemi[chemi['组类'] == i].iloc[:,j].min(), chemi[chemi['组类']
== i].iloc[:,j].mean(),
        chemi[chemi['组类'] == i].iloc[:,j].max())
    j = 1
    for i in range(len(label)):
        print(chemi[chemi['组类'] == i].iloc[:,j])
```

附录 12

介绍：利用 python 分析聚类结果的敏感度

```
    import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
    from sklearn.decomposition import PCA

    plt.style.use('seaborn')
    plt.rcParams['font.sans-serif'] = ['SimHei']
    plt.rcParams['axes.unicode_minus'] = False
    # large = 22
    # med = 16
    # small = 12
    # params = {
    #     "font.sans-serif": "Simhei",
    #     "axes.titlesize": 22,
    #     "legend.fontsize": 16,
    #     "figure.figsize": (10, 8),
    #     "figure.dpi": 80,
    #     "axes.labelsize": 16,
    #     "axes.titlesize": 16,
    #     "xtick.labelsize": 16,
    #     "ytick.labelsize": 16,
    #     "figure.titlesize": 22,
    #     "axes.unicode_minus": False
    # }
    # plt.rcParams.update(params)
    # 铅钡
    # spss 聚类结果
    spss_cluster_label = [[2, 10, 8, 4, 6],
                          [3, 11],
                          [19, 23, 14, 15, 21, 1, 5, 49, 40, 37, 47, 48,
                              26, 39, 42, 30, 41, 45, 24, 44, 25, 29],
                          [9, 46, 38, 27, 28, 34, 7, 33, 36, 13, 31, 35, 32,
43, 22, 12, 17, 20, 16, 18]]

    label = []
    for i in spss_cluster_label:
        ls = []
        for j in i:
            ls.append(j-1)
        label.append(ls)
```

```
    df = pd.read_excel('2.2 铅钡.xlsx')
    chemi = df.iloc[:,3:]

    chemi.insert(chemi.shape[1], '组类', 0)

    for i in range(len(label)):
        chemi.iloc[label[i],-1] = i

    model = PCA(2)

    mat = model.fit_transform(chemi.loc[:,])

    plt.figure(figsize=(4,4))
    ttl = '铅钡玻璃主成分散点图'
    plt.title(ttl)
    plt.xlabel('PC1')
    plt.ylabel('PC2')

    plt.scatter(x=mat[chemi['组类']==0,0], y=mat[chemi['组类']==0,1],
label='第一组')
    plt.scatter(x=mat[chemi['组类']==1,0], y=mat[chemi['组类']==1,1],
label='第二组')
    plt.scatter(x=mat[chemi['组类']==2,0], y=mat[chemi['组类']==2,1],
label='第三组')
    plt.scatter(x=mat[chemi['组类']==3,0], y=mat[chemi['组类']==3,1],
label='第四组')
    plt.legend(bbox_to_anchor=(1, 0.5), loc=3, borderaxespad=0)

    plt.savefig(ttl + '.svg', bbox_inches='tight')
    plt.show()
    model.explained_variance_ratio_
    # 高钾
    # spss 聚类结果
    spss_cluster_label = [[9,11,10,17,18,8,2,15,16],
                          [1,4,13,14,3,5,12,6,7]]

    label = []
    for i in spss_cluster_label:
        ls = []
        for j in i:
            ls.append(j-1)
        label.append(ls)
```

```
    df = pd.read_excel('2.2 高钾.xlsx')
    chemi = df.iloc[:,3:]

    chemi.insert(chemi.shape[1], '组类', 0)

    for i in range(len(label)):
        chemi.iloc[label[i],-1] = i

    model = PCA(2)

    mat = model.fit_transform(chemi)


    plt.figure(figsize=(4,4))

    ttl = '高钾玻璃主成分散点图'
    plt.title(ttl)
    plt.xlabel('PC1')
    plt.ylabel('PC2')
    plt.scatter(x=mat[chemi['组类']==0,0], y=mat[chemi['组类']==0,1],
label='第一组')
    plt.scatter(x=mat[chemi['组类']==1,0], y=mat[chemi['组类']==1,1],
label='第二组')
    plt.legend(bbox_to_anchor=(1, 0.5), loc=3, borderaxespad=0)

    plt.savefig(ttl + '.svg', bbox_inches='tight')
    plt.show()
    model.explained_variance_ratio_
```

附录 13

介绍：利用 python 鉴别未知文物所属类型

```
    import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn import tree
params = {
    "font.sans-serif":"Simhei",
    "axes.titlesize": 22,
```

```
    "legend.fontsize": 16,
    "figure.figsize": (10, 8),
    "figure.dpi":80,
    "axes.labelsize": 16,
    "axes.titlesize": 16,
    "xtick.labelsize": 16,
    "ytick.labelsize": 16,
    "figure.titlesize": 22,
    "axes.unicode_minus": False
}
plt.rcParams.update(params)
df=pd.read_excel("./表单 2-处理 2.xlsx")
df=df.fillna(0)
df=pd.get_dummies(df,columns=["类型"])
df=df.rename(columns={'类型_铅钡':'铅钡'})
df
data=df.iloc[:,2:17]
target=df["铅钡"]
random=2
Xtrain, Xtest, Ytrain, Ytest = train_test_split(data, target,
test_size=0.2,random_state=random)
clf = tree.DecisionTreeClassifier(
    criterion="gini",
    random_state=random,#随机数种子
    splitter="best",
    max_depth=3,  # 树的最大深度，防止过拟合
    #     min_samples_leaf=10,  #叶节点所需要的最小样本数
    #     min_samples_split=25,# 拆分内部节点所需要的最小样本数，防止过拟合
)
clf = clf.fit(Xtrain, Ytrain)
score = clf.score(Xtest, Ytest)
score
df3=pd.read_excel("./表单 3-处理.xlsx")
df3=df3.fillna(0)
X_pre=df3.iloc[:,2:]
clf.predict(X_pre)
clf.predict_proba(Xtest).T[1]
from sklearn.metrics import roc_auc_score,roc_curve,auc
y_score = clf.predict_proba(Xtest)[:,1]

# Compute ROC curve and ROC area for each class
fpr,tpr,threshold = roc_curve(Ytest, y_score) ###计算真正率和假正率
roc_auc = auc(fpr,tpr) ###计算 auc 的值
```

```
plt.figure()
lw = 2
plt.figure(figsize=(10,10))
plt.plot(fpr, tpr, color='red',
         lw=lw, label='ROC curve (area = %0.2f)' % roc_auc) ###假正率为横坐
标，真正率为纵坐标做曲线
plt.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate(FPR)')
plt.ylabel('True Positive Rate(TPR)')
plt.title('决策树的 ROC 曲线')
plt.legend(loc="lower right")
plt.show()
import graphviz

feature_name = data.columns

dot_data = tree.export_graphviz(
    clf,
    feature_names=feature_name,
    class_names=["高钾","铅钡"],
    filled=True,  # 是否填充颜色
    rounded=True,  # 每一块的边框为弧形
)
graph = graphviz.Source(dot_data)
graph
pd.DataFrame(clf.feature_importances_,index=data.columns)
clf.predict_proba(Xtest).T[1]
# 决策树的分类结果
class_ = ["高钾" if i==0 else "铅钡" for i in clf.predict(X_pre)]
pd.DataFrame(class_,index=df3["文物编号"],columns=["玻璃种类"])
# 逻辑回归分类
from sklearn.linear_model import LogisticRegression as LR
Xtrain, Xtest, Ytrain, Ytest = train_test_split(data, target,
test_size=0.3,random_state=2)
model = LR(max_iter=500)
model.fit(Xtrain, Ytrain)
pd.DataFrame(model.coef_[0],index=data.columns,columns=["逻辑回归系数"])
model.score(Xtest,Ytest)
## 绘制分类结果的 ROC 曲线
from sklearn.metrics import roc_auc_score,roc_curve
import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve, auc  ###计算 roc 和 auc
```

```
y_score = model.fit(Xtrain, Ytrain).decision_function(Xtest)

# Compute ROC curve and ROC area for each class
fpr,tpr,threshold = roc_curve(Ytest, y_score) ###计算真正率和假正率
roc_auc = auc(fpr,tpr) ###计算 auc 的值

plt.figure()
lw = 2
plt.figure(figsize=(10,10))
plt.plot(fpr, tpr, color='red',
         lw=lw, label='ROC curve (area = %0.2f)' % roc_auc) ###假正率为横坐
标，真正率为纵坐标做曲线
plt.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')
plt.xlim([-0.01, 1.01])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate(FPR)')
plt.ylabel('True Positive Rate(TPR)')
plt.title('逻辑回归的 ROC 曲线')
plt.legend(loc="lower right")
plt.show()
## 预测
Xtrain, Xtest, Ytrain, Ytest = train_test_split(data, target,
test_size=0.3,random_state=5)
model = LR(max_iter=500)
model.fit(Xtrain, Ytrain)
class_ = ["高钾" if i==0 else "铅钡" for i in model.predict(X_pre)]
pd.DataFrame(class_,index=df3["文物编号"],columns=["玻璃种类"]).to_excel('3
未知文物类型预测.xlsx')

# 敏感度分析
# 逻辑回归敏感性

w = model.coef_[0]
w0 = model.intercept_

PD = []
for i in range(Xtrain.shape[0]):
    ls = []
    x = Xtrain.iloc[i]
    e = np.exp(-w@x.values + w0)
    for j in range(len(w)):
        ls.append(w[j] * e * ((1+e)**(-2)))
    PD.append(ls)
PD = pd.DataFrame(PD)
```

```
S = []
for i in range(Xtrain.shape[1]):
    s = 0
    for j in range(Xtrain.shape[0]):
        s += abs(PD.iloc[j,i])
    S.append(s / Xtrain.shape[0])

# 敏感度
mgd = pd.DataFrame(S, index=Xtrain.columns,columns=["敏感度
"]).sort_values(by="敏感度", ascending=False)
mgd
mgd.to_excel('3.1 逻辑回归敏感度.xlsx')
import seaborn as sns

from sklearn.decomposition import PCA
model = PCA(2)
mat = model.fit_transform(Xtrain)

plt.title('对玻璃样本应用主成分降维后的散点图')
plt.xlabel('PC1')
plt.ylabel('PC2')
plt.scatter(mat[Ytrain==1,0], mat[Ytrain==1,1], label='铅钡')
plt.scatter(mat[Ytrain==0,0], mat[Ytrain==0,1], label='高钾')
plt.legend()
    plt.savefig('对玻璃样本应用主成分降维后的散点图.svg')
```

附录 14

介绍：利用 python 分析不同类别玻璃化学成分的共线性

```
    import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
    from statsmodels.stats.outliers_influence import
variance_inflation_factor  # 计算 VIF 方差膨胀因子
    from itertools import combinations
    df = pd.read_excel('4.1 铅钡.xlsx')
    data = df.iloc[:,2:]

    ls = []
```

```
    for i in range(2,7):
        for cmb in combinations(data.columns, i):
            cmb = [*cmb]
            dt = data[cmb]
            vif = [variance_inflation_factor(dt.values,
dt.columns.get_loc(i)) for i in dt.columns]
            mn = np.mean(vif)
            ls.append([i, cmb, mn])

    pd.DataFrame(ls,columns=['特征个数','特征',
'VIF']).sort_values(by='VIF', ascending=False).to_excel('4.1 铅钡共线
性.xlsx', index=False)
    df = pd.read_excel('4.1 高钾.xlsx')
    data = df.iloc[:,2:]

    ls = []
    for i in range(2,7):
        for cmb in combinations(data.columns, i):
            cmb = [*cmb]
            dt = data[cmb]
            vif = [variance_inflation_factor(dt.values,
dt.columns.get_loc(i)) for i in dt.columns]
            # vif.sort(reverse=True)
            # mn =
pd.DataFrame(vif,index=dt.columns,columns=["VIF"]).mean()
            mn = np.mean(vif)
            ls.append([i, cmb, mn])

    pd.DataFrame(ls,columns=['特征个数','特征',
'VIF']).sort_values(by='VIF', ascending=False).to_excel('4.1 高钾共线
性.xlsx', index=False)
```

附录 15

介绍：利用 python 进行不同类别的玻璃进行因子分析

```
    # 铅钡因子分析
    import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
```

```
    from factor_analyzer import FactorAnalyzer
    from factor_analyzer.factor_analyzer import
calculate_bartlett_sphericity
    from factor_analyzer.factor_analyzer import calculate_kmo
    params = {
        "font.sans-serif":"Simhei",
        "axes.titlesize": 22,
        "legend.fontsize": 16,
        "figure.figsize": (10, 8),
        "figure.dpi":80,
        "axes.labelsize": 16,
        "axes.titlesize": 16,
        "xtick.labelsize": 16,
        "ytick.labelsize": 16,
        "figure.titlesize": 22,
        "axes.unicode_minus": False
    }
    plt.rcParams.update(params)
    df=pd.read_excel("4.1 铅钡.xlsx",index_col=0).iloc[:,1:]

    # 根据VIF 去除变量
    df = df.drop(['氧化钙(CaO)', '氧化铜(CuO)', '氧化铅(PbO)', '氧化钡(BaO)',
'氧化锶(SrO)', '二氧化硫(SO2)'], axis=1)

    data=df # factor_analyzer 库会自动标准化的
    # 检验数据是否适合做因子分析
    ## Bartlett's 球状检验
    # $H_0$原假设：相关系数矩阵是一个单位矩阵（不适合做因子分析）
    # $H_1$备择假设：相关系数矩阵不是单位矩阵（适合做因子分析）

    pd.DataFrame(calculate_bartlett_sphericity(data),index=["卡方统计量","p
值"],columns=["Bartlett's 球状检验"])
    # p 值<0.05，说明在95%的置信水平上拒绝原假设，我们认为数据适合做因子分析
    # # 碎石检验选择因子个数
    # 如果不知道因子个数则需要选择，如果有既定的则不需要选
    # 这里 FactorAnalyzer 参数暂时可以随便选
    fa = FactorAnalyzer(n_factors=3,
                        method='minres',# minres:最小残差法 ml:Maximum
Likelihood 最大似然法 principle:主成分法
                        rotation=None)
    fa.fit(data)
    e,v=fa.get_eigenvalues() # 排序好的特征值和特征向量

    plt.figure(figsize=(10,6),dpi=80)
```

```
    plt.plot(range(1,len(data.columns)+1),e,marker="o")
    plt.title("碎石图")
    plt.xlabel("因子数量")
    plt.ylabel("特征值")

    plt.grid(alpha=0.2)  # 显示网格
    plt.show()  # 显示图形
    # 进行因子分析
    plt.figure(figsize=(8,6),dpi=120)
    n_factors=3
    # methods = ['minres','ml','principal']
    methods = ['minres']
    # methods=["principal"]
    # rotations =
['varimax','promax','oblimin','oblimax','quartimin','quartimax','equamax',
'geomin_obl','geomin_ort']
    rotations = ['oblimax']
    for method in methods:
        for rotation in rotations:
            fa = FactorAnalyzer(n_factors=n_factors,
                                method=method,# minres:最小残差法 ml:Maximum
Likelihood 最大似然法 principal:主成分法
                                rotation=rotation)
            # 旋转方法：（一个一个试直到试到可解释性最强的系数）
            # varimax（正交旋转）、promax（倾斜旋转）、oblimin（倾斜旋转）、
oblimax（正交旋转）、quartimin（斜向旋转）
            # quartimax（正交旋转）、equamax（正交旋转）、geomin_obl（倾斜旋
转）、geomin_ort（正交旋转）
            fa.fit(data)
            factors=pd.DataFrame(fa.loadings_,index=df.columns,columns=[f"
因子{i}" for i in range(1,n_factors+1)]) # 因子载荷矩阵（对应课件里P25 成分矩
阵）
            plt.title("最小残差法"+'+'+"正交旋转")
            sns.heatmap(factors.abs(),annot=True, cmap="Purples") # 绘制热力
图便于直观看出
            plt.show()
    # 高钾因子分析
    df=pd.read_excel("4.1 高钾.xlsx",index_col=0).iloc[:,1:]

    df.drop(['氧化钾(K2O)', '氧化钙(CaO)', '氧化铝(Al2O3)', '氧化铁(Fe2O3)',
'五氧化二磷(P2O5)', '氧化锶(SrO)'], axis=1,inplace=True)

    data=df # factor_analyzer 库会自动标准化的
```

```
    pd.DataFrame(calculate_bartlett_sphericity(data),index=["卡方统计量","p
值"],columns=["Bartlett's 球状检验"])
    # 这里 FactorAnalyzer 参数暂时可以随便选
    fa = FactorAnalyzer(n_factors=3,
                        method='minres',# minres:最小残差法 ml:Maximum
Likelihood 最大似然法 principle:主成分法
                        rotation=None)
    fa.fit(data)
    e,v=fa.get_eigenvalues() # 排序好的特征值和特征向量

    plt.figure(figsize=(10,6),dpi=80)
    plt.plot(range(1,len(data.columns)+1),e,marker="o")
    plt.title("碎石图")
    plt.xlabel("因子数量")
    plt.ylabel("特征值")

    plt.grid(alpha=0.2)  # 显示网格
    plt.show()  # 显示图形
    n_factors = 3
    methods = ['principal']
    # rotations =
['varimax','promax','oblimin','oblimax','quartimin','quartimax','equamax',
'geomin_obl','geomin_ort']
    rotations = ['oblimax']
    for method in methods:
        for rotation in rotations:
            fa = FactorAnalyzer(n_factors=n_factors,
                                method=method,# minres:最小残差法 ml:Maximum
Likelihood 最大似然法 principal:主成分法
                                rotation=rotation)
            # 旋转方法：（一个一个试直到试到可解释性最强的系数）
            # varimax（正交旋转）、promax（倾斜旋转）、oblimin（倾斜旋转）、
oblimax（正交旋转）、quartimin（斜向旋转）
            # quartimax（正交旋转）、equamax（正交旋转）、geomin_obl（倾斜旋
转）、geomin_ort（正交旋转）
            fa.fit(data)
            factors=pd.DataFrame(fa.loadings_,index=df.columns,columns=[f"
因子{i}" for i in range(1,n_factors+1)]) # 因子载荷矩阵（对应课件里 P25 成分矩
阵）
            plt.title(method+'+'+rotation)
            sns.heatmap(factors.abs(),annot=True, cmap="Purples") # 绘制热力
图便于直观看出
            plt.show()
```

附录 16

介绍：利用 python 对表单 2 的缺失值进行填充

```
import pandas as pd
df = pd.read_excel('表单 2.xlsx')
df.fillna(0, inplace=True)
df.to_excel('表单 2.xlsx')
```

附录 17 描述性统计表

介绍：高钾组、铅钡组的风化类及未风化类描述性统计表

铅钡风化组描述性统计表

成分	SiO2	Na2O	K2O	CaO	MgO	Al2O3	Fe2O3	CuO	PbO	BaO	P2O5	SrO	SnO2	SO2
mean	24.91	0.22	0.13	2.7	0.65	2.97	0.58	2.28	43.31	11.81	5.28	0.42	0.07	1.37
std	10.61	0.56	0.24	1.66	0.71	2.63	0.74	2.82	12.23	9.98	4.2	0.26	0.27	4.21
min	3.72	0	0	0	0	0.45	0	0	15.71	0	0	0	0	0
25%	18.79	0	0	1.34	0	1.4	0	0.71	35.47	6.78	1.31	0.22	0	0
50%	25.02	0	0	2.88	0.57	2.38	0.3	1.14	44.06	8.79	4.97	0.42	0	0
75%	30.2	0	0.24	3.5	1.16	3.56	0.99	3.14	48.84	14.51	8.59	0.59	0	0
max	53.33	2.22	1.05	6.4	2.73	13.65	2.74	10.57	70.21	35.45	14.13	1.12	1.31	15.95
skew	0.31	2.67	2.51	0.38	1.04	2.83	1.4	2.1	-0.03	1.28	0.4	0.54	4.37	3.26
kurt	3.78	8.29	9.15	2.4	3.77	11.48	4.03	6.17	2.97	3.46	2.1	3.41	18.96	10.66

高钾风化组描述性统计表

成分	SiO2	Na2O	K2O	CaO	MgO	Al2O3	Fe2O3	CuO	PbO	BaO	P2O5	SrO	SnO2	SO2
mean	93.96	0	0.54	0.87	0.2	1.93	0.26	1.56	0	0	0.28	0	0	0
std	1.73	0	0.45	0.49	0.31	0.96	0.07	0.93	0	0	0.21	0	0	0
min	92.35	0	0	0.21	0	0.81	0.17	0.55	0	0	0	0	0	0
25%	92.65	0	0.15	0.64	0	1.36	0.22	1.01	0	0	0.16	0	0	0
50%	93.5	0	0.66	0.83	0	1.72	0.28	1.54	0	0	0.28	0	0	0
75%	94.84	0	0.88	1.04	0.4	2.38	0.31	1.62	0	0	0.36	0	0	0
max	96.77	0	1.01	1.66	0.64	3.5	0.35	3.24	0	0	0.61	0	0	0
skew	0.85	0	-0.54	0.5	1.01	0.78	-0.3	1.22	0	0	0.4	0	0	0
kurt	2.01	0	1.49	2.48	1.6	2.2	1.66	2.91	0	0	2.27	0	0	0

高钾未风化组描述性统计表

成分	SiO2	Na2O	K2O	CaO	MgO	Al2O3	Fe2O3	CuO	PbO	BaO	P2O5	SrO	SnO2	SO2
mean	67.98	0.7	9.33	5.33	1.08	6.62	1.93	2.45	0.41	0.6	1.4	0.04	0.2	0.1
std	8.76	1.29	3.92	3.09	0.68	2.49	1.67	1.66	0.59	0.98	1.43	0.05	0.68	0.19
min	59.01	0	0	0	0	3.05	0	0	0	0	0	0	0	0
25%	61.68	0	7.6	4.04	0.62	5.14	0.48	1	0	0	0.69	0	0	0
50%	65.53	0	9.83	6.1	1.16	6.18	2.11	2.34	0.16	0	1.02	0.02	0	0
75%	71.17	0.52	12.3	7.57	1.6	7.93	2.45	3.43	0.51	1.07	1.29	0.08	0	0.09
max	87.05	3.38	14.52	8.7	1.98	11.15	6.04	5.09	1.62	2.86	4.5	0.12	2.36	0.47
skew	1.16	1.5	-1.2	-0.88	-0.43	0.48	1.18	0.1	1.37	1.49	1.68	0.57	3.46	1.4
kurt	2.88	2.89	3.73	2.21	1.9	2.23	4.15	1.87	2.8	3.32	3.72	1.62	10.09	2.57

附录 18 铅钡亚类划分结果

文物编号	文物采样点	亚类类型
2	02	铅钡 II
8	08	铅钡 I
8	08 严重风化点	铅钡受风化影响严重
11	11	铅钡 I
19	19	铅钡 II
20	20	铅钡 I
23	23 未风化点	铅钡 III
24	24	铅钡 I
25	25 未风化点	铅钡 III
26	26	铅钡 I
26	26 严重风化点	铅钡受风化影响严重
28	28 未风化点	铅钡 III
29	29 未风化点	铅钡 III
30	30 部位 1	铅钡 II
30	30 部位 2	铅钡 II
31	31	铅钡 III
32	32	铅钡 III
33	33	铅钡 III
34	34	铅钡 II
35	35	铅钡 III
36	36	铅钡 II
37	37	铅钡 III
38	38	铅钡 II
39	39	铅钡 II
40	40	铅钡 II

41	41	铅钡 II
42	42 未风化点 1	铅钡 III
42	42 未风化点 2	铅钡 III
43	43 部位 1	铅钡 II
43	43 部位 2	铅钡 II
44	44 未风化点	铅钡 III
45	45	铅钡 III
46	46	铅钡 III
47	47	铅钡 III
48	48	铅钡 III
49	49	铅钡 II
49	49 未风化点	铅钡 III
50	50	铅钡 II
50	50 未风化点	铅钡 III
51	51 部位 1	铅钡 II
51	51 部位 2	铅钡 II
52	52	铅钡 II
53	53 未风化点	铅钡 III
54	54	铅钡 II
54	54 严重风化点	铅钡 II
55	55	铅钡 III
56	56	铅钡 II
57	57	铅钡 II
58	58	铅钡 II

附录 19 高钾亚类划分结果

文物编号	文物采样点	亚类类型
1	01	高钾 II
3	03 部位 1	高钾 I
3	03 部位 2	高钾 II
4	04	高钾 II
5	05	高钾 II
6	06 部位 1	高钾 II
6	06 部位 2	高钾 II
7	07	高钾 I
9	09	高钾 I
10	10	高钾 I
12	12	高钾 I
13	13	高钾 II
14	14	高钾 II
16	16	高钾 II
18	18	高钾 I
21	21	高钾 I
22	22	高钾 I