基于 CLR 的玻璃文物成分分析与分类模型

摘要

本文通过对玻璃文物的化学成分分析，一方面**实现了对已风化的铅钡、高钾类型玻璃风化前化学成分的预测，即“成分还原”**，另一方面构建了**稳健的**玻璃亚类分类模型，该模型可以实现，**在风化导致玻璃文物偏移其原始化学成分的情况下，对该玻璃的亚类有大致正确的判断，即对风化文物原始成分的预测偏差不会显著影响分类结果**。这对于经常面临被风化玻璃文物的考古工作者来说具有重要的现实意义。

首先，针对成分数据（compositional data）的分析应聚焦于各成分之间的相对含量，而非某成分的绝对含量。并且成分数据普遍具有高维、稀疏、右偏严重的特征。基于以上特征，我们**对成分数据进行了中心对数比变换 (Centered logratio transformation, CLR)，变换后的各指标反映了某成分相对其他成分的含量大小（相对重要性），并且一定程度上改善了成分数据集的一些不良性质**。

针对问题一，利用卡方检验、Fisher 精确检验等方法，发现纹饰、颜色与表面风化无关，而玻璃类型与风化有较强关联。随后，对不同类型的玻璃，我们给出了预测风化文物风化前成分相对含量的模型。由于已有数据中，我们无法精确将某一风化后的成分数据对应到另一风化前的成分数据，因此在建立此预测模型时，无法使用传统回归模型。因此，只能通过描述性统计对不同类型玻璃风化前后的成分变化趋势做分析，并用统计量刻画这种趋势，再结合相应统计量预测风化前的原始成分数据。

针对问题二，我们根据文物化学成分给出文物分类依据，**并且论证该分类模型的稳健性能确保对风化文物原始成分的预测偏差不会显著影响分类结果，敏感性分析结果良好**。高钾、铅钡玻璃表现为线性可分，我们**通过 SVM 得到了高钾、铅钡玻璃的显式分类表达式**。在亚类分类方面，我们先通过**层次分析法筛选出了区分亚类的化学成分**，将文物分为**铅钡-高纳、铅钡-低钠、高钾-低钙高镁、高钾-高钙低镁四个亚类，并通过 SVM 再次给出了显式分类表达式**。随后对未风化数据调用亚类分类模型，发现针对该分类器，**对风化文物原始成分的预测偏差不会显著影响分类结果，模型敏感性检验表现优异**。

针对问题三，我们将问题二中的分类器应用于未分类数据。结果显示，除 A5 样品以外，其他文物样品均能被分类器完美区分，尤其是已风化样品 A2、A6、A7，经过还原其原始相对成分含量数据以后（即考虑了风化对其相关化学成分的相对含量的影响趋势），依然能够被分类器分开。并且模型对 A5 的“跨类行为”给出了合理解释。**第三问中针对未分类样品的分类结果表明，该分类器可以有效地对风化后的玻璃文物做亚类区分，并且结果具有较好的合理性与可解释性**。

针对问题四，我们用奇异值分解法分别对两种玻璃的经过 CLR 变换后的协方差阵进行主成分分析，做出双标图，找出了多个具有强相关关系的成分组合，并解释其具体

的化学内涵，最后阐述了在各成分相对含量的变化程度和关联程度上各亚种玻璃之间存在的差异。

总之，本文对成分数据进行了合适的 CLR 变换预处理后，从“成分相对重要性”的角度提取了成分数据的信息。并且，本文良好解决了“如何在风化导致玻璃文物偏移其原始化学成分的情况下，对该玻璃的类别有大致正确的判断”这一核心问题。另外，由于数据特殊且样本量小，本文没有使用复杂的或可解释性差的模型，而是注重结果的可解释性和现实意义。**以上是本文的核心优点。**

关键字：CLR　SVM　层次聚类　风化文物

一、 问题重述

1.1 问题背景

丝绸之路是古代中西方文化交流的通道，其中玻璃是早期贸易往来的宝贵物证。玻璃的主要原料是石英砂，主要化学成分是二氧化硅（SiO2）。由于纯石英砂的熔点较高，为了降低熔化温度，在炼制时需要添加助熔剂。古代常用的助熔剂有草木灰、天然泡碱、硝石和铅矿石等，并添加石灰石作为稳定剂，石灰石煅烧以后转化为氧化钙（CaO）。添加的助熔剂不同，其主要化学成分也不同。

然而，古代玻璃极易受埋藏环境的影响而风化。在风化过程中，内部元素与环境元素进行大量交换，导致其成分比例发生变化，从而影响对其类别的正确判断。

1.2 问题要求

问题设置是层层递进，并且服务于同一主题的——如何在风化导致玻璃文物偏移其原始化学成分的情况下，对该玻璃的类别有大致正确的判断。

问题一首先要求以风化与否为结果变量，分析数据集中定类变量之间的关系。接下来要求分析不同类型玻璃文物风化前后化学成分的变化规律，并给出通过风化后成分数据预测风化前成分数据的方法。

问题二首先要求分析已经给出的高钾、铅钡两类玻璃化学成分，给出其分类规律；在此基础上，对每类玻璃进行亚类划分，并针对分类依据、结果做模型评估。

问题三要求利用问题二中建立的亚类分类模型，对未知类别的玻璃文物的化学成分进行分析，其中包括处于不同风化状态的玻璃文物。

问题四要求针对不同类别的玻璃文物样品，分析其化学成分之间的关联关系，并比较不同类别之间的化学成分关联关系的差异性。

二、 问题分析

2.1 总体分析

本题是一个关于古代玻璃文物的化学成分数据的数据挖掘类问题。

从分析目的来看，需要对玻璃文物进行分类，并在风化导致玻璃文物偏移其原始化学成分的情况下，对该玻璃的类别有大致正确的判断。因此本题主要需要达成三方面任务　其一，考察风化对不同类型玻璃文物化学成分含量的影响，并能通过风化后成分预测风化前成分；其二，根据不同文物的化学成分对文物样品进行合理分类，给出分类

标准，考察同类文物化学成分相似性、异类文物化学成分差异性；其三，对风化后文物进行分类，确保分类模型足够稳健，能包容一部分成分预测带来的误差。

从数据特性来看，成分数据具有高维、稀疏、严重右偏等特性，并且各成分累加和为定值 100%。因此，本题数据相对特殊，需要对数据做一定的预处理。

从模型选择来看，本体数据样本量较小，且分析目的均与实际情境息息相关，因此模型应当追求其可解释性，不宜用过于复杂的模型。

2.2 问题一分析

问题一的核心目标在于给出预测风化文物原始化学成分的模型。本模型的预测目标是，针对文物某一部位，根据其风化后的成分数据预测其风化前的成分数据。但是，在已给的数据集中，只有一组某一文物部位风化前后的成分数据（49 号文物），其他数据均是不同文物部位在不同风化状态下的数据。这导致**在已有数据中，我们无法精确将某一风化后的成分数据对应到另一风化前的成分数据，因此在建立此预测模型时，无法使用传统回归模型**。

因此，只能通过描述性统计对不同类型玻璃风化前后的成分变化趋势做分析，并用统计量刻画这种趋势，再结合相应统计量预测风化前的原始成分数据。

2.3 问题二分析

问题二的核心目标在于根据文物化学成分给出文物分类依据，并解释分类的现实合理性，并且**论证该分类模型的稳健性能确保对风化文物原始成分的预测偏差不会显著影响分类结果**。

在数据集上，高钾、铅钡玻璃表现为线性可分，并可得到显式分类表达式。在亚类分类方面，考虑到预测数据未必准确，因此需要通过已有的未风化数据建立亚类分类模型，再考察分类模型对预测偏差的包容度，也即模型的敏感性。

2.4 问题三分析

问题三的核心目标在于进一步通过未分类的数据检验分类模型的有效性。新数据中存在风化文物，这类文物完成原始成分预测后，需要考察预测误差在何种范围内时，分类结果保持稳定。

2.5 问题四分析

问题四的核心目标在于建立合适的模型来提取数据的有效特征，其一要控制表面风化的对成分相对含量的影响，其二要兼顾部分样本量偏少的亚类。最终形成的特征要具有可比较性。

三、 数据预处理

成分数据 (compositional data) 是一类特殊的数据。其恒为正值，且当成分数据以百分比形式呈现时，各成分比例累加和应为 1。另外，成分数据具有尺度不变性 (scale invariance)，因此**其信息全部蕴含在各成分相对大小之中，仅考察某成分比例的绝对大小具有误导性**。最后，一般来说成分数据具有高维、稀疏的特点，这也是本赛题所给数据集的特点。

综合以上特点，并结合参考文献，数据预处理主要从缺失值处理、对数据做中心对数比变换 (Centered logratio transformation, CLR) 两方面着手。

3.1 删除不符合要求的成分数据

赛题中提到“将成分比例累加和介于 85% 和 105% 之间的数据视为有效数据”。因此删除两条不符合此要求的数据。

3.2 缺失值的处理

赛题中提到，成分数据中缺失值为“表示未检测到该成分”。**对于本数据集，缺失值统一用 0.04 填充，其合理性体现在两方面**：

其一，查阅文献发现，成分数据的缺失（或零值）通常可能由检测仪器测量下限值导致。比如对于含量为 0.01% 的某化学成分，可能因为仪器无法检出而被视作不含有改成分。考察本数据集，发现所有成分数据中的最小值为 0.04%，可认为 0.04% 为仪器的测量下限值。

其二，在后续做数据中心对数比变换时，大量的缺失值、零值将导致转换后大量缺失值、Inf 的出现，因此对缺失值和零值做合理的填充很有必要。

3.3 数据中心对数比变换 (CLR)

对于某成分数据 $x=[x_1,x_2,...,x_D]\in S^D$, 定义：

$$\mathrm{clr}(\mathbf{x})=\left[\ln\frac{x_1}{\mathrm{g}_m(\mathbf{x})},\ln\frac{x_2}{\mathrm{g}_m(\mathbf{x})},\ldots,\ln\frac{x_D}{\mathrm{g}_m(\mathbf{x})}\right]=\xi \tag{3.1}$$

其中 $\mathrm{g}_m(\mathbf{x})$ 是几何平均，即：

$$g_m(\mathbf{x})=\sqrt[D]{\prod_{k=1}^{D}x_k}=\exp\left(\frac{1}{D}\sum_{k=1}^{D}\ln x_k\right)$$

对 $clr(x)$ 的分量进行简单代数运算后可发现：

$$\xi_1=\ln\frac{x_1}{g_m(\mathbf{x})}=\ln x_1-\ln g_m(\mathbf{x})=\ln x_1-\frac{1}{D}\sum_{i=k}^{D}\ln x_k \tag{3.2}$$

3.3.1 CLR 变换的直观理解

从（3.1）式可以看出，**CLR 变换后的指标不仅考虑了某成分占比的绝对大小，更考虑了该成分相比其他成分的相对大小，即相对重要性**。可试举一例：

分别对物品一：$[30,30,40]$ 与物品二：$[30,10,60]$ 两条成分数据做 CLR 变换，结果分别为 $[-0.10,-0.10,0.19]$ 与 $[0.13,-0.96,0.82]$。可以发现，两条数据的第一个分量虽然都是 30，但是由于其他成分组成不同，因此经过 CLR 变换后的分量值并不同。从两条数据也可以直观看出，第一种成分在物品二的构成的相对重要性要高于物品一。

从（3.2）式可以看出，**CLR 的实质是对数据取对数后进行中心化处理**（即 CLR 的命名由来），其不仅可以**解决成分数据严重右偏的问题**，而且经过 CLR 处理后的指标分量与 0 的大小关系体现了该成分的相对重要性是高于平均还是低于平均。

3.3.2 CLR 的理论支撑

CLR 的提出基于这样一种理论视角：将成分数据的样本空间（称为艾奇逊空间）看成一个不同于欧式空间的赋范线性空间，CLR 变换是 D 维成分数据样本空间到 D 维欧氏空间中 D-1 维超平面的等距变换。

关于艾奇逊空间的一些基本性质将呈现在附件 1 中。简单来说，艾奇逊空间通过定义一系列运算，为我们合理操作、解释成分数据提供保障，而 CLR 实现了艾氏空间到欧氏空间的相互转换，使得我们可以用熟悉的欧氏空间中的运算方法来处理 CLR 数据后，再映射回艾氏空间：用 $d(\cdot,\cdot),\|\cdot\|,\langle\cdot,\cdot\rangle$ 表示欧氏空间中的距离、模长、向量内积，用 $d_a(\cdot,\cdot),\|\cdot\|_a,\langle\cdot,\cdot\rangle_a$ 表示艾氏空间中的距离、模长、向量内积。**对于 $x_1,x_2\in S^D$ 以及实数 α,β，CLR 变换有如下性质：**

$(a)\ \mathrm{clr}(\alpha\odot\mathbf{x}_1\oplus\beta\odot\mathbf{x}_2)=\alpha\cdot\mathrm{clr}(\mathbf{x}_1)+\beta\cdot\mathrm{clr}(\mathbf{x}_2)$

$(b)\ \langle\mathbf{x}_1,\mathbf{x}_2\rangle_a=\langle\mathrm{clr}(\mathbf{x}_1),\mathrm{clr}(\mathbf{x}_2)\rangle;$

$(c)\ \|\mathbf{x}_1\|_a=\|\mathrm{clr}(\mathbf{x}_1)\|,\quad \mathrm{d}_a(\mathbf{x}_1,\mathbf{x}_2)=\mathrm{d}(\mathrm{clr}(\mathbf{x}_1),\mathrm{clr}(\mathbf{x}_2)).$

性质（a）体现出艾氏空间与欧氏空间同构，而性质（b）、（c）体现出 CLR 为等距变换。而将 CLR 变换后的数据转换为艾氏空间当中的成分数据（以百分比为例），只需要进行如下操作即可：

$$x_j=\frac{\exp(\xi_j)}{\sum_{k=1}^{D}exp(\xi_k)}\quad \text{for } j=1,\ldots,D$$

综上所述，我们对原始数据做 CLR 处理，并将处理后的数据看作欧氏空间当中的向量进行分析建模。针对 CLR 处理后的单个成分分量，**将其解释为该成分的相对含量或相对重要性**，单个分量蕴含了该成分相对其他成分的重要性。最后，在必要时刻，我们将分析、计算后的 CLR 数据重新转变为加和为一的成分数据，即通俗形式。

四、 问题一的模型建立与求解

4.1 玻璃表面风化与其类型、纹饰和颜色的关系

玻璃表面是否风化以及玻璃类型、纹饰和颜色均为定类变量，考虑以表面风化为结果变量分别和类型、纹饰、颜色构造列联表，并根据类别的期望频数大小分别做卡方独立性检验或 Fisher 精确性检验（小样本）。

卡方检验：对于 rxc 的列联表，若其总样本数大于 40 且每个类别的理论频数大于等于 5 可用卡方检验。

$$H_0 : p_{ij} = p_{i\cdot} \cdot p_{\cdot j}, \quad i = 1, ..., r, j = 1, ..., c.$$

检验统计量为

$$\chi^2 = \sum_{i=1}^{r}\sum_{j=1}^{c} \frac{(n_{ij} - n\hat{p}_{ij})^2}{n\hat{p}_{ij}},$$

在原假设 H_0 成立时上式近似服从自由度为 $(r-1)(c-1)$ 的 χ^2 分布。其中各 $\hat{p}_{ij}$ 是在 H_0 成立下得到的 $\hat{p}_{ij}$ 的最大似然估计，其表达式为

$$\hat{p}_{ij} = \hat{p}_{i\cdot} \cdot \hat{p}_{\cdot j} = \frac{n_{i\cdot}}{n} \cdot \frac{n_{\cdot j}}{n},$$

对给定的显著性水平 α，检验的拒绝域为 $W = \{\chi^2 \geqslant \chi^2_{1-\alpha}((r-1)(c-1))\}$

对于不符合卡方检验要求的小样本数据，利用 Fisher 精确检验，以 2x2 表格为例：此时独立性对应超几何分布：

$$P(t) = P(n_{11} = t) = \frac{\binom{n_{1+}}{t}\binom{n_{2+}}{n_{+1} - t}}{\binom{n}{n_{+1}}}$$

该公式表示 n_ij 的分布只取决于 $n_1 1$。给定边际总计，$n_1 1$ 决定了其他三个单元格计数。特别地，对于 2x2 表格，独立性等价于发生比之比 $\theta = 1$。检验 $H_0 : \theta = 1$，P 值就可以表示为多个超几何分布概率的求和。比如，考察 $H_a : \theta > 1$。对于给定的边际总计，$n_1 1$ 的值较大时对应的样本发生比之比也较大，检验结果也就更有可能支持 H_a。

从独立性检验的结果可知（见下页表 1），给定显著性水平 α=0.05，针对纹饰与风化、颜色与风化的假设检验，均没有充足理由拒绝原假设，即不能认为颜色、纹饰对玻璃是否风化有显著影响，符合直观认识。而玻璃类型与风化的假设检验中，p 值充分小，有充足理由拒绝原假设，认为玻璃类型与是否风化具有关联性。考虑到玻璃类型与其化学成分关系密切，**猜想化学成分是玻璃类型与是否风化的共同影响因素，即协变量。**

因此，接下来将结合玻璃的类型，分析文物样品表面有无风化的化学成分含量的统计规律，并根据风化点检测数据，预测其风化前的化学成分含量。

表 1　表面风化与纹饰、类型、颜色的独立性检验 (α=0.05)

变量	名称	表面风化		总计	假设检验方法	p 值
		无风化	风化			
纹饰	A	11	11	22	Fisher 精确检验	0.0836
	B	6	0	6		
	C	17	13	30		
合计		34	24	58		
类型	高钾	6	12	18	卡方检验	0.0195
	铅钡	28	12	40		
合计		34	24	58		
颜色	黑	2	0	2	Fisher 精确检验	0.616
	蓝绿	9	6	15		
	绿	0	1	1		
	浅蓝	12	8	20		
	浅绿	1	2	3		
	深蓝	0	2	2		
	深绿	4	3	7		
	紫	2	2	4		
合计		30	24	54		

4.2 各类玻璃风化前后化学成分的统计规律

4.2.1 描述性统计

从下图可以直观看出，对于高钾、铅钡两种不同类型的玻璃，它们风化前后化学成分含量的变化呈现出不同的统计规律。比如，高钾玻璃和铅钡玻璃风化前后的氧化钠(Na2O) 含量呈现相反的变化趋势　　风化后，高钾玻璃中钠的相对含量上升较多，铅钡玻璃的钠相对含量小幅下降。求出两类玻璃风化前后个化学成分相对含量均值（见下表），也可发现类似规律。

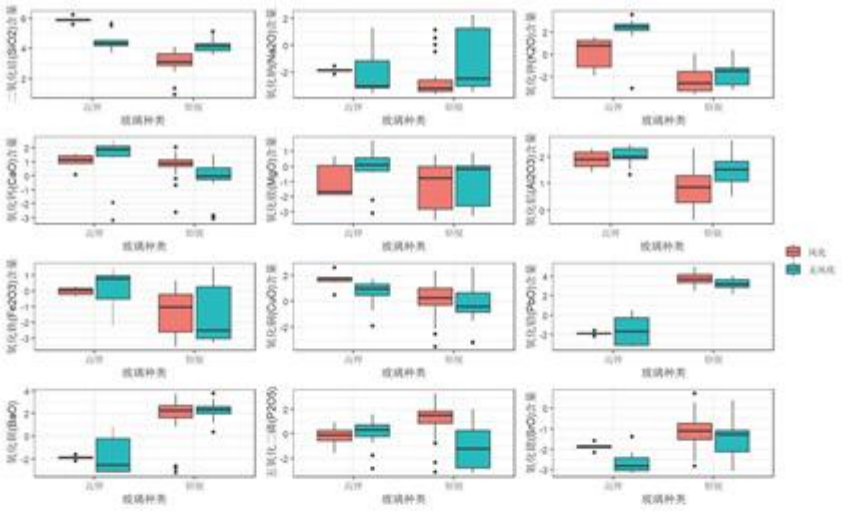


图 1　不同玻璃种类在风化前后各化学成分含量箱线图

4.2.2 预测模型：根据风化点数据，预测其风化前各成分含量

根据风化文物的化学成分含量数据预测其风化前的成分数据是一项现实且重要的工作。化学成分含量信息具有客观性，这将非常有助于考古学家推定该文物的具体年代、地域以及其制造工艺等信息，是考古学中文物鉴定的重要手段。

本模型的预测目标是，针对文物某一部位，根据其风化后的成分数据预测其风化前的成分数据。但是，在已给的数据集中，只有一组某一文物部位风化前后的成分数据(49 号文物)，其他数据均是不同文物部位在不同风化状态下的数据。**这导致在已有数据中，我们无法精确将某一风化后的成分数据对应到另一风化前的成分数据，因此在建立此预测模型时，无法使用传统回归模型。**

基于以上原因，我们将分别求出高钾、铅钡玻璃风化前后各成分相对含量均值差 $\Delta\xi_{mean}$（通过表 2 求出），并通过风化后成分加上均值差的方法预测该部位风化前的成分的相对含量：

$$\xi_{erode} + \Delta\xi_{mean} = \xi_{predict}$$

其中 ξ_{erode} 为风化后的相对成分数据，$\xi_{predict}$ 是预测后的数据。

这样做的合理性和好处有五：

(1) 考虑了风化对不同类型玻璃相对成分的不同影响

(2) 在响应变量和解释变量的样本不具有明确一一对应关系的情况下，防止盲目套用回归模型导致过大的预测偏差

(3) 充分利用了风化前数据的信息

(4) 预测方法相对直观，解释性强

(5) 不易受严重风化点的干扰

表 2　不同类型玻璃风化前后各化学成分相对含量均值

	铅钡风化	铅钡无风化	高钾风化	高钾无风化
二氧化硅 (SiO2)	3.13	4.168	5.886	4.43
氧化钠 (Na2O)	-2.643	-1.104	-1.876	-1.945
氧化钾 (K2O)	-2.449	-1.81	0.12	2.043
氧化钙 (CaO)	0.712	-0.118	1.034	1.172
氧化镁 (MgO)	-1.314	-0.976	-0.98	-0.171
氧化铝 (Al2O3)	0.841	1.483	1.893	2.04
氧化铁 (Fe2O3)	-1.416	-1.434	-0.016	0.185
氧化铜 (CuO)	0.146	-0.224	1.639	0.654
氧化铅 (PbO)	3.763	3.22	-1.876	-1.576
氧化钡 (BaO)	1.78	2.216	-1.876	-1.76
五氧化二磷 (P2O5)	0.957	-1.1	-0.197	0.019
氧化锶 (SrO)	-1.093	-1.489	-1.876	-2.668
二氧化硫 (SO2)	-2.412	-2.831	-1.876	-2.423

将预测后的 CLR 数据转变回成分数据后，预测结果如下：

五、 问题二的模型建立与求解

5.1 界定数据范围

在进行类别划分前，我们首先需要界定分析数据的范围，该规则同样适用于后续亚类的划分：

1. 只使用未风化点的数据[1]

2. 包含变量：氧化钠 (Na2O)、氧化钙 (CaO)、氧化镁 (MgO)、氧化铝 (Al2O3)、氧化铁 (Fe2O3)[2]

[1]若使用风化点数据，因为在部分成分的相对含量上风化前与风化后存在较大差距，例如二氧化硅，使用聚类算法时风化样本和未风化样本的组内距离小于组间距离，尽管能得到很好的轮廓系数，但区分出的簇与样本风化状态有关而与品类无关；若利用问题一中得到的映射将风化点的数据变换为未风化点，因为求平均的过程模糊了亚类特征，会使得大量数据中的信息含量减少，对聚类模型产生噪声

[2]不包含问题一中含量较高的成分，例如二氧化硅、氧化钡、氧化铅，避免其较高的对数比对距离计算的影响；不包含含量过低、非零样本量过少的成分，例如氧化锶、氧化锌，简化模型；不包含分布过于均匀的成分，例如五氧化二磷，根据问题一中描述型统计的分析，这部

文物采样点	二氧化硅	氧化钠	氧化钾	氧化钙	氧化镁	氧化铝	氧化铁	氧化铜	氧化铅	氧化钡	五氧化二磷	氧化锶	二氧化硫
2	69.02	0.13	1.34	0.69	1.11	7.33	1.23	0.12	18.58	0.04	0.31	0.09	0.02
8	42.12	0.14	0.06	0.48	0.04	1.88	0.03	5.32	12.35	35.79	0.34	0.18	1.26
08严重风化点	13.49	0.19	0.08	1.44	0.06	2.18	0.04	2.25	19.55	49.1	1	0.37	10.25
11	65.25	0.13	0.27	1.05	0.68	3.51	0.03	2.34	10.16	15.55	0.83	0.17	0.02
19	63.88	0.14	0.06	0.98	0.63	5.17	1	1.85	19	6.32	0.86	0.1	0.02
26	41.53	0.14	0.06	0.47	0.04	0.99	0.03	5.42	12.76	37.08	0.3	0.23	0.96
26严重风化点	10.35	0.18	0.75	1.29	0.06	2.21	0.04	2.45	17.14	54.03	0.76	0.41	10.34
34	67.61	0.12	0.32	0.23	0.04	2.06	0.31	0.7	18.11	10.35	0.03	0.1	0.02
36	66.7	6.18	0.16	0.1	0.03	1.81	0.19	0.28	14.44	10	0.01	0.09	0.02
38	62.12	4.3	0.05	0.2	0.04	3.26	0.19	0.34	19.15	10.12	0.04	0.18	0.02
39	59.89	0.15	0.06	0.39	0.05	0.77	0.03	0.49	28.67	9.03	0.12	0.33	0.02
40	46.59	0.18	0.07	0.81	0.06	0.84	0.18	0.03	40.31	10.22	0.22	0.45	0.03
41	47.65	0.17	0.76	1.98	3.5	5.78	1.61	0.12	23.45	13.8	0.87	0.29	0.02
43部位1	37.24	0.2	0.08	2.43	1.33	4.54	0.79	3.93	36.98	11.99	0.01	0.46	0.03
43部位2	56.95	0.17	0.07	2.59	1.24	6.02	1.27	0.97	24.18	4.69	1.53	0.29	0.02
48	73.08	1.81	0.29	0.6	1.05	12.58	0.49	0.01	4.43	5.49	0.07	0.08	0.01
49	62.5	0.14	0.06	1.54	1.58	7.85	2.07	0.37	15.28	7.26	1.09	0.24	0.02
50	47.64	0.17	0.07	1.31	0.62	3.33	0.3	0.73	24.01	20.62	0.76	0.42	0.02
51部位1	56.21	0.15	0.06	1.26	1.35	8.06	0.95	0.77	18.93	11.19	0.84	0.21	0.02
51部位2	59.34	0.18	0.07	2.2	2	4.69	0.41	0.51	29.38	0.06	1.1	0.03	0.03
52	58.1	4.55	0.06	0.79	0.62	1.76	0.18	0.39	22.04	10.68	0.58	0.24	0.02
54	52.57	0.16	0.51	1.16	1.5	6.58	0.03	0.48	26.94	9.1	0.45	0.49	0.02
54严重风化点	51.03	0.2	0.08	0.02	1.64	7.32	0.04	0.98	35.9	0.07	1.91	0.8	0.03
56	60.74	0.14	0.06	0.39	0.04	2.59	0.03	0.4	17.7	17.64	0.24	0.02	0.02
57	54.92	0.14	0.06	0.44	0.04	3.17	0.03	0.61	20.06	20.48	0	0.02	0.02
58	63.55	0.14	0.48	1.13	0.82	4.95	0.63	1.6	16.94	8.78	0.85	0.12	0.02

图 2　铅钡玻璃风化部位风化前成分预测

文物采样点	二氧化硅	氧化钠	氧化钾	氧化钙	氧化镁	氧化铝	氧化铁	氧化铜	氧化铅	氧化钡	五氧化二磷	氧化锶	二氧化硫
7	77.59	0.13	0.98	4.41	0.32	8.24	0.75	4.35	0.19	0.16	2.72	0.07	0.08
9	73.6	0.12	13.41	2.36	0.3	5.08	1.3	1.92	0.18	0.15	1.44	0.06	0.08
10	72.82	0.12	20.31	0.78	0.29	3.03	1.03	1.01	0.17	0.14	0.16	0.06	0.07
12	66.96	0.11	21.04	2.52	0.27	5.15	1.08	1.88	0.16	0.14	0.57	0.06	0.07
22	61.41	0.11	14.44	5.44	4.1	11.56	1.22	0.59	0.15	0.13	0.74	0.05	0.07
27	75.76	0.13	0.96	3.78	4.25	10.19	0.86	2.02	0.19	0.16	1.57	0.06	0.08

图 3　高钾玻璃风化部位风化前成分预测

考虑到本题可使用的数据量有限，基于决策树的机器学习效果较差，我们首先尝试了 Logistic 回归。Logistic 回归的模型呈现出明显的过拟合，这意味着样本线性可分。因此，在观察各成分分布的散点图后，我们选取氧化钡（BaO）和氧化铅（PbO）作为高钾玻璃、铅钡玻璃分类的主要依据。为了给出更为明确的分类依据，我们选取适用于小样本、高维模式识别的支持向量机（SVM）。

5.2 支持向量机 (SVM) 实现分类

5.2.1 支持向量机 (SVM) 介绍

支持向量机（SVM）建立于结构风险最小原理基础之上，以间隔最大化为学习策略，其算法本质是求解凸二次规划的最优化算法。由于支持向量机依靠接近平面的若干支持向量建立分界线或超平面，因此能够凭借有限的样本信息获取现有条件下的全局最优解，避免了神经网络方法可能面临的样本量过少和局部最优解问题，同时还有较好的泛化能力。经过层次聚类处理，现有古代玻璃样本未风化采样点数据 37 个，数据维

分均匀分布的变量很难作为聚类依据

度达到 14 维（14 种化学成分），当前分类目标是将高钾玻璃与铅钡玻璃区分为两大类，后续还涉及对高钾玻璃与铅钡玻璃进一步区分亚类，多分类或多个二分类支持向量机在该情形下具有很强的应用价值。

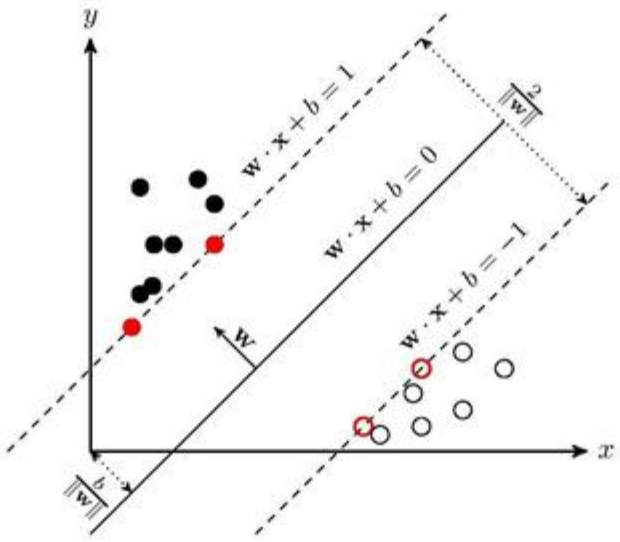


图 4　SVM 示意图

5.2.2 SVM 求解过程

我们假设数据集$(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$, 其中

$$y(x) = w^T \psi(x) + b, \psi(x) \tag{1}$$

是核函数。

Y 为样本类别，

$$\begin{aligned} y(x_i) > 0 <=> y_i = +1 \\ y(x_i) < 0 <=> y_i = -1 \end{aligned} \tag{2}$$

由此可以推出，

$$y_i \cdot y(x_i) > 0 \tag{3}$$

此时，各点距离分界线的距离可以表示为

$$\text{dist} = \frac{y_i \left(w^T \cdot \psi(x_i) + b\right)}{|w|} \tag{4}$$

基于结构风险最小原理和间隔最大化策略，该模型优化函数为：

$$\min_{w,b} \frac{1}{2} w^2 \text{ 且} y_i \left(w^T \cdot \psi \left(x_i \right) + b \right) \geq 1 \tag{5}$$

此后，支持向量机引入拉格朗日乘数法求解目标函数，得到最佳决策边界。

5.2.3 SVM 核函数选择

支持向量机中存在如下多种核函数。

线性核函数：

$$K \left(x_i, x_j \right) = x_i^T x_j \tag{6}$$

多项式核函数：

$$K \left(x_i, x_j \right) = \left(\gamma x_i^T x_j + b \right)^d \tag{7}$$

高斯核函数，又称 radial 核函数：

$$K \left(x_i, x_j \right) = \exp \left(-\gamma \left\| x_i - x_j \right\|^2 \right) \tag{8}$$

Sigmoid 核：

$$K \left(x_i, x_j \right) = \tanh \left(\gamma x_i^T x_j + b \right) \tag{9}$$

5.2.4 SVM 分类结果呈现

由于高钾玻璃和铅钡玻璃中氧化钡（BaO）、氧化铅（PbO）含量所构成的散点图线性可分，我们直接选择线性函数作为支持向量机的核函数，分析得到如下结果：

此时，分界线的函数表达式为：

$$\xi_{PbO} = -3.51 \xi_{BaO} + 5.97 \tag{10}$$

由此，高钾玻璃和铅钡玻璃的分类规律[3]可以概括为：

$$\begin{aligned} &\xi_{PbO} + 3.51 \xi_{BaO} - 5.97 > 0，\text{样本属于铅钡玻璃} \\ &\xi_{PbO} + 3.51 \xi_{BaO} - 5.97 < 0，\text{样本属于高钾玻璃} \end{aligned} \tag{11}$$

[3]此处的含量并非真实含量，而是经过 CLR 处理后的含量

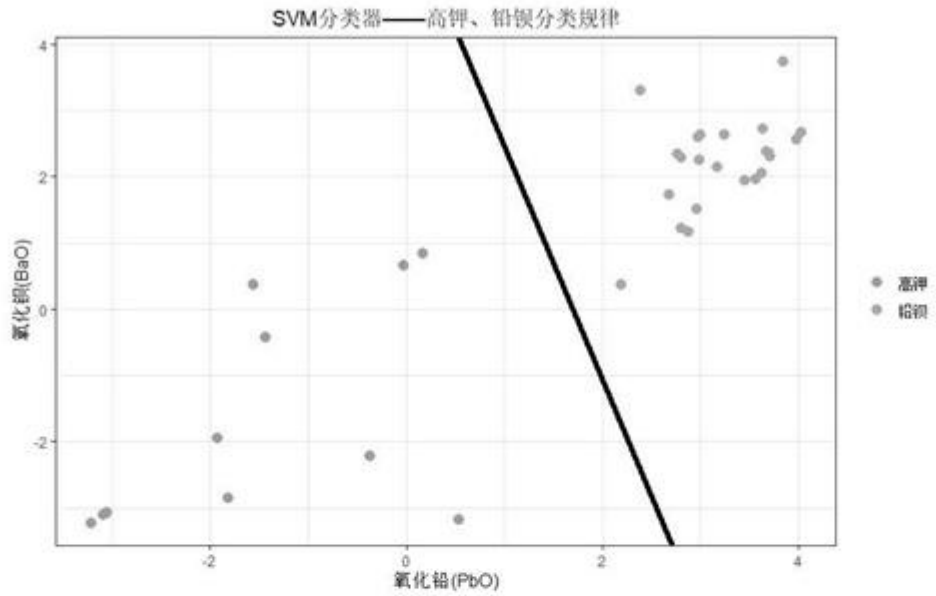


图 5　SVM 分类器　　高钾、铅钡玻璃分类规律

5.3 层次聚类模型探索性分析

5.3.1 层次聚类模型介绍

在界定高钾玻璃、铅钡玻璃这两个类别后，我们需要探寻两者内部的亚类划分标准。

考虑到数据特征　　维度多、样本少、分布稀疏，本文先采用层次聚类方法进行探索性分析。

层次聚类（Hierarchical clustering）是聚类算法的一种，通过计算不同类别数据点间的相似度来创建一棵有层次的嵌套聚类树。在聚类树中，不同类别的原始数据点是树的最低层，树的顶层是一个聚类的根节点。

5.3.2 层次聚类模型设置

- **欧几里得距离：** $d(x,y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$ 计算各变量间的相对距离[4]。
- **Ward 的最小方差方法 (Ward' s minimum variance method)：** 最小化整个群内方差。在每个步骤中，合并具有最小簇间距离的一对簇。
- **凝聚聚类：** 也被称为 AGNES（凝聚嵌套）。以自下而上的方式工作。每个对象最初被认为是单元素簇（叶子），在算法的每个步骤中，将最相似的两个群集组合成新的更大的群集（节点）。迭代此过程，直到所有点都只是一个单个大簇（root）的成员。结果可以绘制为树状图。

[4]这里的距离并非原始变量直接计算得出，而是由中心对数变换过的变量得出。

5.3.3 层次聚类的基本原理和计算方式

假设已对 n 个样本进行分类，并且分类个数为 k，$G_1, G_2, \cdots, G_k$ 表示为 k 个类，G_t 类中的样本个数用 n_t 表示，G_t 类的重心用 $\overline{X}^t$ 表示，G_t 类的第 i 个样品用 X_i^t 表示 $(t=1,2,\cdots,k)$，则 G_t 类中的样品离差平方和表示为：

$$W_t = \sum_{i=1}^{n_t} \left(X_i^t - \overline{X}^t\right)^T \left(X_i^t - \overline{X}^t\right) \tag{12}$$

其中 $X_i^t \square \overline{X}^t$ 为 m 维向量，k 类的总离差平方和为：

$$W = \sum_{t=1}^{k} W_t = \sum_{t=1}^{k} \sum_{i=1}^{n_t} \left(X_i^t - \overline{X}^t\right)^T \left(X_i^t - \overline{X}^t\right) \tag{13}$$

1. 将 n 个样本各自分为一类，此时 W=0；
2. 对其中某两个类进行合并，此时应满足使 W 的增加量最小；
3. 不断重复直至所有的样品归为一类。

将某两类合并之后增加的 $\triangle W$ 作为其之间的平方距离，即 $D_{pq}{}^2 = W_r - (W_p + W_q)$ 表示 G_p 和 G_q 之间的平方距离，则：

$$W_r = \sum_{i=1}^{n_r} \left(X_i^r - \overline{X}^r\right)^T \left(X_i^r - \overline{X}^r\right) = \sum_{i=1}^{n_p} \left(X_i^p - \overline{X}^p\right)^T \left(X_i^p - \overline{X}^p\right) + \sum_{i=1}^{n_q} \left(X_i^q - \overline{X}^q\right)^T \left(X_i^q - \overline{X}^q\right) \tag{14}$$

其中，$\overline{X}^r = \frac{1}{n_r}\left(n_p \overline{X}^p + n_q \overline{X}^q\right)$

$$D_{pq} = W_r - (W_p + W_q) = \frac{n_p n_q}{n_r} \left(\overline{X}^p - \overline{X}^q\right)^T \left(\overline{X}^p - \overline{X}^q\right) = \frac{n_p n_q}{n_r} d_{pq}^2 \tag{15}$$

当 G_p 和 G_q 合并为 G_r 后，G_r 与其它类 G_k 的距离为：

$$D_{rk}^2 = \frac{n_p n_q}{n_r} \left(X_r - \overline{X}^k\right)^T \left(X_r - \overline{X}^k\right) = \frac{n_p + n_k}{n_r + n_k} D_{pk}^2 + \frac{n_q + n_k}{n_r + n_k} D_{qk}^2 - \frac{n_k}{n_r + n_k} D_{pq}^2 \tag{16}$$

5.3.4 聚类过程和结果

高钾玻璃层次聚类：

观察聚类层次关系，在最大距离的垂直线处设置阈值（height=7），得到 3 个聚类

- 聚类一：3（部位 1）、13、14、16
- 聚类二：1、3（部位 2）、4、5、6（部位 2）、21
- 聚类三：6（部位 1）、18

为了直观反映分类标准，做出散点图如下：

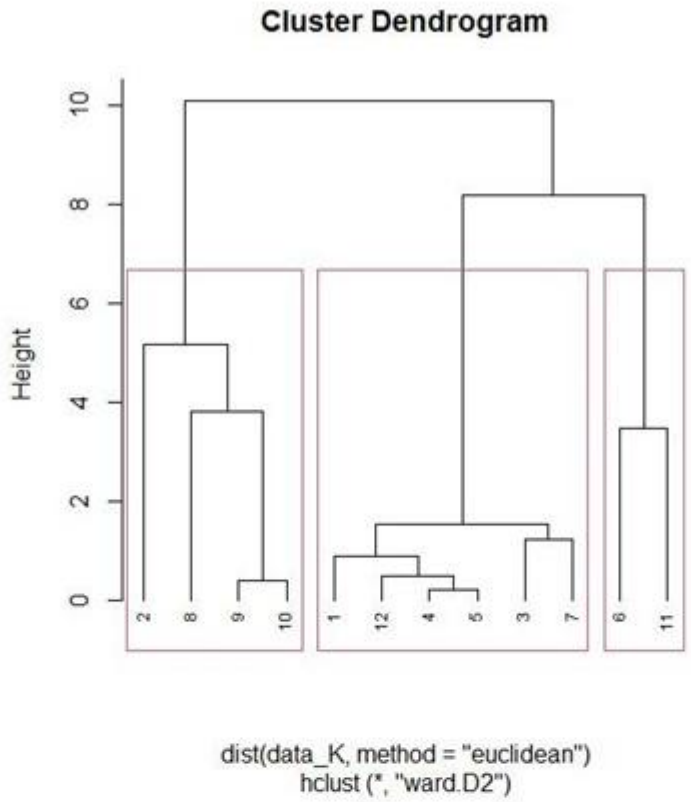


图 6　高钾分层聚类树及形成的亚类划分

可以发现不同聚类在氧化纳 (Na2O)、氧化钙 (CaO)、氧化镁 (MgO) 三个成分上有较好的区分度:

- 聚类一：高钠高钙
- 聚类二：低钠高钙
- 聚类三：低钙高镁

铅钡玻璃层次聚类:

观察聚类层次关系，在最大距离的垂直线处设置阈值（height=12），得到 2 个聚类

- 聚类一：42（部位 2）、33、23、29、32、47、30、25、44、45、28、53、35、31
- 聚类二：49、55、20、37、42（部位 1）、24、30

为了直观反映分类标准，做出散点图如下:

可以发现不同聚类在氧化纳 (Na2O) 一个成分上有较好的区分度:

- 聚类一：高钠
- 聚类二：低钠

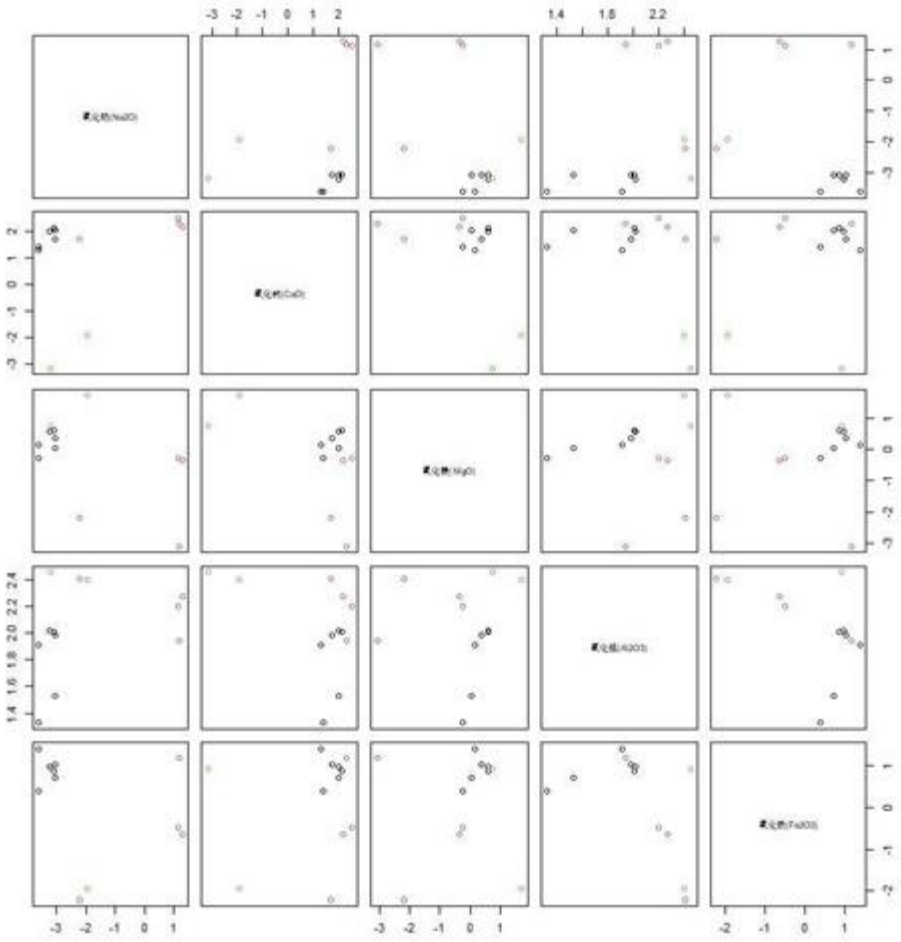


图 7　高钾亚类散点图

5.3.5 层次聚类总结

作为探索性分析的方法，层次聚类法为我们选择亚类和对应的指标提供了参考。在散点图中，可以发现部分成分也具有很好的区分度，呈现两极分布（例如铅钡玻璃中氧化钙），但氧化纳对于模型中的距离计算影响更大，所以这种成分的区分度并没有很好地为模型的亚类区分做参考；同时，在区分亚类时，还需要结合颜色、表面风化等分类变量和化学实践，这是该模型没有考虑到的。

综上，结合层次聚类法的结果，基于简单高效的目标，我们设定了亚类如下：

- 高钾玻璃：高纳/低钠　高钙低镁/低镁高钙
- 铅钡玻璃：高钠/低钠

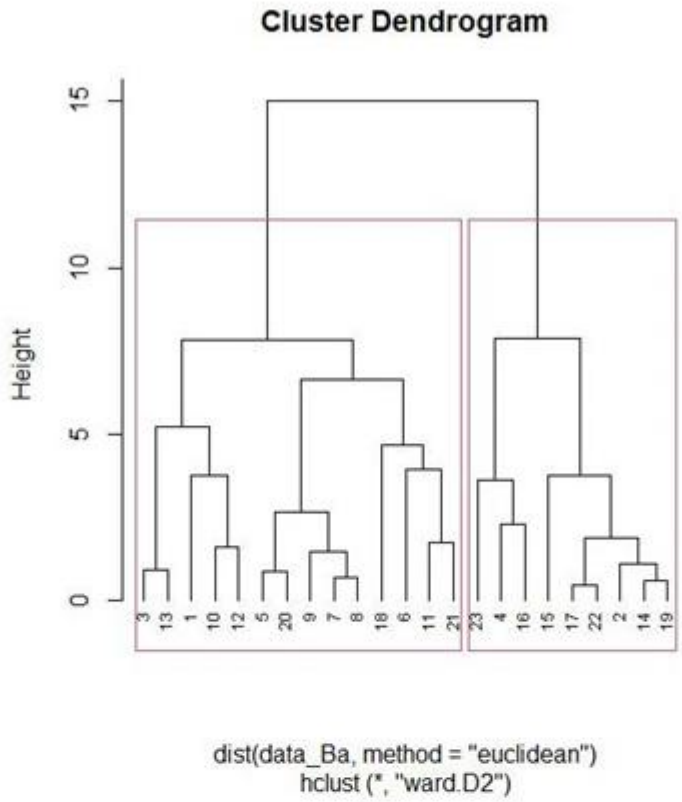


图 8　铅钡分层聚类树及形成的亚类划分

5.4 SVM 划分亚类

由于层次聚类属于无监督学习，其结果只能反映样本中化学成分的相似性。为了给出更为明确的分类依据，我们选取支持向量机寻找边界。

同样的，在观察层次聚类中各成分之间的散点图特征后，我们选择使用线性核函数的二分类支持向量机对亚类展开进一步分析，结果如下：

各个亚类包含的样本与其特征情况如下：

得到亚类划分标准如下：

$$\begin{aligned} &\xi_{MgO} + 2.10\xi_{CaO} - 1.51 > 0，\text{样本属于高钙低镁玻璃} \\ &\xi_{MgO} + 2.10\xi_{CaO} - 1.51 < 0，\text{样本属于低钙高镁玻璃} \end{aligned} \tag{17}$$

$$\begin{aligned} &\xi_{Na2O} + 0.10\xi_{CaO} - 0.75 > 0，\text{样本属于高钠玻璃} \\ &\xi_{Na2O} + 0.10\xi_{CaO} - 0.75 < 0，\text{样本属于低钠玻璃} \end{aligned} \tag{18}$$

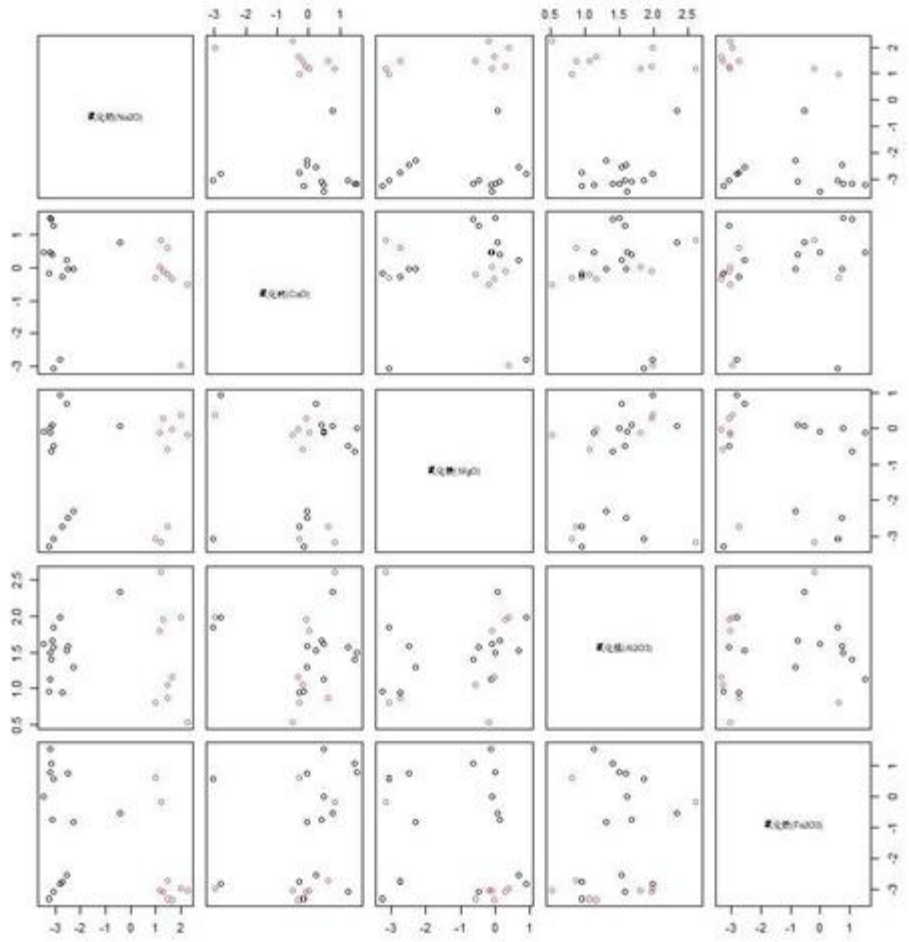


图 9　铅钡亚类散点图

5.5 SVM 总结与敏感性分析

在上图中，我们不仅展示了分类器对于训练模型的无风化数据的良好分类效果，而且对于通过风化数据“还原”出的未风化数据也有非常好的分类效果。[5]并且在后续进一步的敏感性分析当中可以看到，即使对分类直线的截距施加一定扰动，分类器依然可以做到有效分类。

因此，现有的文物亚类分类器具有良好的稳健性，其最关键地体现于——即使对已风化文物的“成分还原”会存在一定偏差，但这种偏差不会显著影响分类结果。因此，该分类模型可以有效地对风化后的玻璃文物做亚类区分。

[5]三角形数据点对应“还原”后的数据，它由某个已风化的文物部位的成分相对含量预测而来

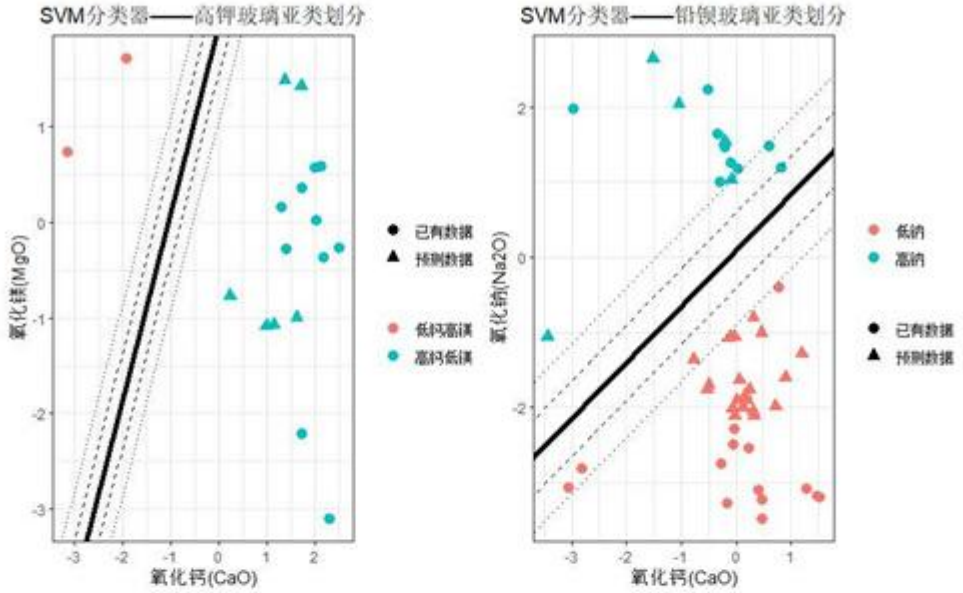


图 10　SVM 分类器　　亚类划分

亚类	包含样本	主要特点
高钠	23，25，36，38，42，44，45，47，48，52，53，54，55	多数呈浅蓝色
低钠	2，8，11，19，20，24，26，28，29，30，31，32，33，34，35，37，39，40，41，43，46，49，50，51，54，56，57，58	
低钙高镁	6，18	多数呈深蓝色，不易风化
高钙低镁	1，3，4，5，7，9，10，12，13，14，16，21，22，27	

图 11　Caption

六、 问题三的模型建立与求解

6.1 多重 SVM 分类器

基于问题二中引入的支持向量机，我们依次对高钾玻璃、铅钡玻璃以及其亚类的划分确立了标准。对于未知类别的玻璃文物，我们首先通过预测还原其未风化状态的化学

表 3　敏感性分析　不同程度扰动对于模型精准度的影响

	0	0.05	0.1	0.2	0.3	0.4	0.5
氧化镁 (MgO)	100.00%	100.00%	100.00%	100.00%	100.00%	94.00%	88.89%
氧化钙 (CaO)	100.00%	100.00%	100.00%	100.00%	94.44 %	83.33%	77.78%
氧化钠 (Na2O)	100.00%	100.00%	100.00%	93.88%	77.55%	53.06%	40.82%
氧化钙 (CaO)	100.00%	100.00%	100.00%	95.92%	91.84%	75.51%	61.22%

成分 CLR，而后根据先前确定的分类标准，经过二次分类确定其所属亚类。

6.2 类别划分

大类	亚类	包含样本
铅钡	高钠	A5
	低钠	A2，A3，A4，A8
高钾	低钙高镁	无
	高钙低镁	A1，A6，A7

图 12　问题三　文物亚类划分

在上图中，“Before” 经过预测的某已风化文物风化前的成分相对含量数据。可以看出，除了 A5 样品以外，其他文物样品均能被分类器完美区分，尤其是已风化样品 A2、A6、A7，经过还原其原始相对成分含量数据以后（即考虑了风化对其相关化学成分的相对含量的影响趋势），依然能够被分类器划分。该分类器可以有效地对风化后的玻璃文物做亚类区分。

针对样品 A5，其在第一次分类中被判定为铅钡玻璃，从铅钡玻璃的总体风化情况看，其钠的相对含量会有所上升而钙的相对含量会显著下降。因此，铅钡玻璃的风化倾

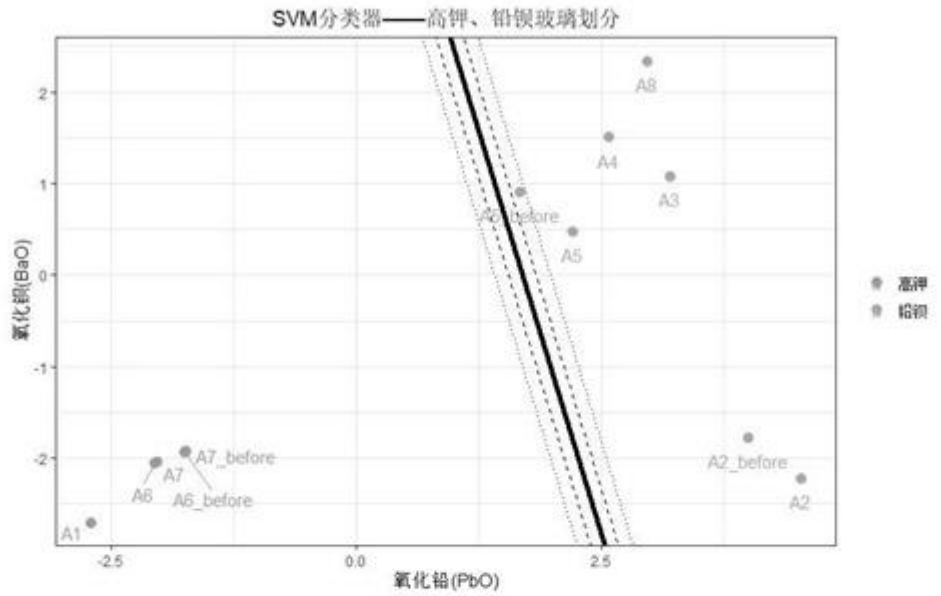


图 13　SVM 分类器　　划分高钾玻璃与铅钡玻璃

向于朝低钠亚类发展。而 A5 经过风化后依然保持了相对较高的钠含量，则分类器据此判定 A5 在风化前属于铅钡-高纳亚类。

综上所述，第三问中针对未分类样品的分类结果表明，该分类器可以有效地对风化后的玻璃文物做亚类区分，并且结果具有较好的合理性与可解释性。

七、 问题四的模型建立与求解

7.1 主成分分析法介绍

主成分分析法（PCA）是对数据矩阵奇异值分解（SVD）结果的解释过程，对于成分数据来说，中心对数变换过的数据集 X^* 的奇异值分解由三个矩阵决定：

$$clr(X^*) = U \cdot D \cdot V^t \tag{19}$$

- **矩阵 V** 的行表示变量；它的列是矩阵 D 分量的标准正交向量，称为右奇异向量、载荷或主成分；它们定义了单纯形的一组标准正交基。
- **矩阵 D** 是一个对角矩阵，其中 D 个奇异值按递减顺序排列；它们被解释为数据集中的坐标在新基上的标准差。
- **矩阵 U** 的行表示观测值，列表示 N 个成分的标准正交向量，称为左奇异向量或成分得分；U 的行被解释为数据集在新基上的标准化坐标。

现在选择前 r 个奇异值及其相关的前 r 个左右奇异向量，并计算 $X_r = U_r \cdot D_r \cdot V_r^t$

这是对 X^* 最好的 r 阶近似，整体的近似质量可以被以下统计量度量：

$$\pi_r = \sum_{i=1}^{r} d_{ii}^2 / \sum_{j=1}^{D} d_{jj}^2 \tag{20}$$

除了对多维数据进行降维，主成分分析法也能解读数据集 X^* 变量之间的相关关系，这正是本文选择主成分分析的主要原因。在进行计算前，需要对数据集进行中心对数变换，使左右奇异向量能反映成分数据的相对尺度。

7.2 根据协方差双标图进行探索性分析

主成分分析中常用的双标图（biplot）能够帮助我们找到一些更好地描述手头数据集特定结构的投影　　哪些投影是值得探索的。双标图是 r=2 情况下（或者 r=3 情况下的 3D 图）对变量和观察结果的图形表示，是数据云和变量轴在二维图（或 3D）上的同步投影

在构建双标图时，需要选择合适的形状参数（0 1 之间），通常用 α 表示。之后观测值通常被用散点表示，其位置由 $U_r \cdot D_r^{(1-\alpha)}$ 的行决定，同时，变量通常被绘制为箭头（从原点到以 $V_r \cdot D_r^{\alpha}$ 的行作为坐标的点）。$\alpha = 0$ 时的双标图被称为“形式双标图（form biplot）”；$\alpha = 1$ 时的双标图被称为“协方差双标图（covariance biplot）”

在协方差双标图中，箭头长度与每个变量的方差成正比，任意两个箭头之间的夹角余弦为它们的相关系数。对于成分数据，矩阵奇异值分解的对象必须是中心对数变换过的数据集，所以不能直接解释双标图中的射线：每条箭头代表一个变量的中心对数变换方差，与原始变量的关系相当复杂，但我们能根据箭头判断两个变量之间的对数比，这与我们数据集的变异矩阵密切相关。当用双标图研究相关性结构时，可以考虑以下规律：

1. 如果两个比例变量的对数比几乎为常数，它们之间的连线应该很短，即箭头靠得很近；
2. 相反，如果一个链接非常长，则说明两个变量之间的对数比是高度变化的。例如，如果有三个很长的箭头指向互成 120° 的方向，它们的三元图将会表现得很分散；
3. 两个箭头之间的角度应该近似于两个对数比之间的相关系数
   - 两个不相关的对数比的箭头正交，两个正交的箭头可能是不相关的；
   - 在同一条线上的箭头（夹角为 0° 或者 180°）对应的对数比是完全线性相关的，它们的子组合也共线；

7.2.1 高钾玻璃双标图

做出山崖落石图，观察解释比例，PC1-4 共可以解释这 13 个变量 92.11% 变异程度。

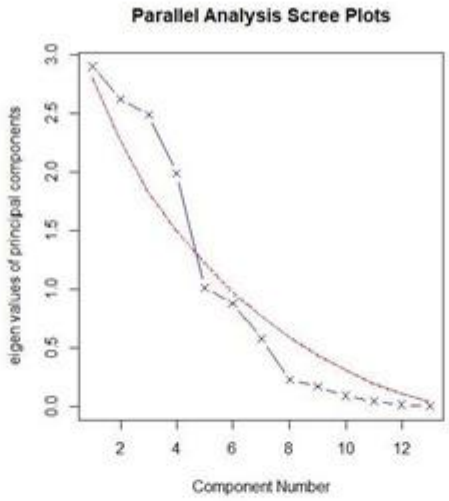


图 14　高钾玻璃成分的山崖落石图

做出主成分载荷矩阵图，发现：

- 第一主成分主要与氧化钙 (CaO)、氧化锶 (SrO) 斜交，而基本不能被其余变量表示；
- 第二主成分主要与氧化铁 (Fe2O3)、氧化钡 (BaO) 斜交，而基本不能被其余变量表示；
- 第三主成分主要与氧化铜 (CuO)、二氧化硫 (SO2) 斜交，而基本不能被其余变量表示；

双标图表现出的比例变量之间的关系：

1. 对数比变化不大的变量：氧化钙 (CaO)、氧化锶 (SrO)、氧化铁 (Fe2O3)、氧化钡 (BaO)
2. 高度线性相关的变量组合：
   - 二氧化硅 (SiO2)-氧化铝 (Al2O3)
   - 氧化纳 (Na2O)-氧化钾 (K2O)-二氧化硫 (SO2)
   - 氧化铁 (Fe2O3)-氧化铜 (CuO)
3. 对于以上组合，组间几乎正交，即相关性不强

部分理论解释：

- 对于二氧化硅 (SiO2) 和氧化铝 (Al2O3)：两者之间易以络合的方式形成 Si-O-Si 键、Si-O-Al 键，形成铝硅酸盐 ($xAl_2O_3 \cdot ySiO_2$)，即硅酸盐中的 SiO_4 四面体的一部分由 AlO_4 四面体取代组成，故两者之间含量可能存在一定相似性。
- 对于氧化纳（Na2O）和氧化钾（K2O）：钠和钾作为同族元素性质相似，在制造高钾玻璃的过程中会加入草木灰（碳酸钾和少量磷）、泡碱（水合碳酸钠）、硝石（硝酸钾）等物质，故钠和钾的含量具有一定相关性。

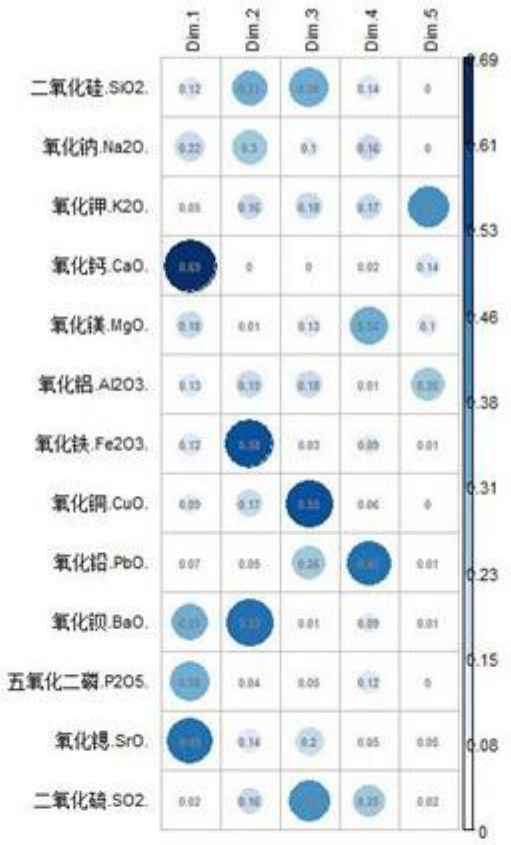


图 15　高钾玻璃主成分载荷矩阵图

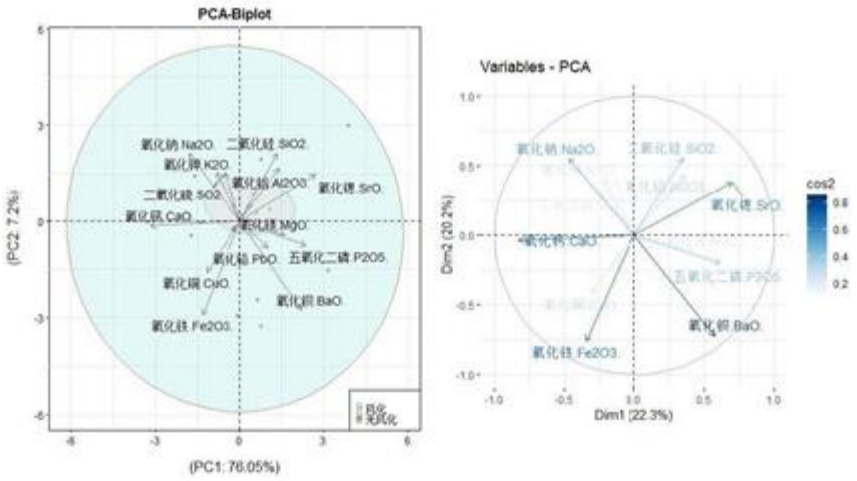
7.2.2 铅钡玻璃双标图

做出山崖落石图，观察解释比例，PC1-4 共可以解释这 13 个变量 90.92%变异程度。做出主成分载荷矩阵图，发现：

- 第一主成分主要与氧化铝 (Al2O3)、二氧化硅 (SiO2)、二氧化硫 (SO2) 斜交，而基本不能被其余变量表示；
- 第二主成分主要与氧化钠 (Na2O)、氧化钡 (CaO)、氧化铁 (Fe2O3)、氧化钡 (BaO)、五氧化二磷 (P2O5) 斜交，而基本不能被其余变量表示；

双标图表现出的比例变量之间的关系：

1. 对数比变化不大的变量：氧化钾 (K2O)、氧化锶 (SrO)
2. 高度线性相关的变量组合：
   - 氧化钡 (BaO)-氧化铜 (CuO)-氧化镁 (MgO)[负相关]-氧化铁 (Fe2O3)[负相关]
   - 五氧化二磷 (P2O5)-氧化钙 (CaO)-氧化钠 (Na2O)[负相关]
3. 对于以上组合，组间几乎正交，即相关性不强



(a) 风化前后对比　　(b) 各指标之间对比

图 16　双标图（横纵坐标分别为第一、第二主成分）

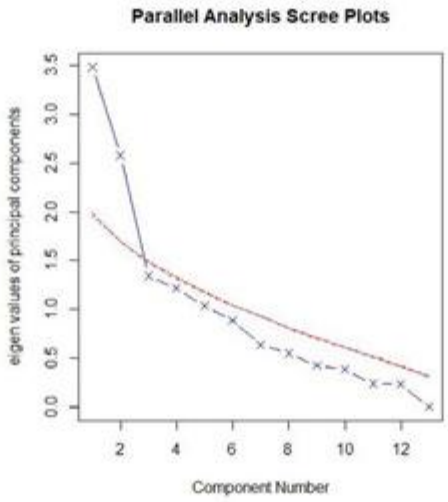


图 17　铅钡玻璃成分的山崖落石图

部分理论解释：

- 对于氧化钡 (BaO)、氧化铜 (CuO) 和氧化镁 (MgO)：镁钡作为同族元素，性质具有相似性，故具有强线性关系，而铜和镁都能提供氧化层保护以阻碍金属离子移动，达到防风化的目的，在功能上相互有替代性（混合碱效应和阻塞效应）

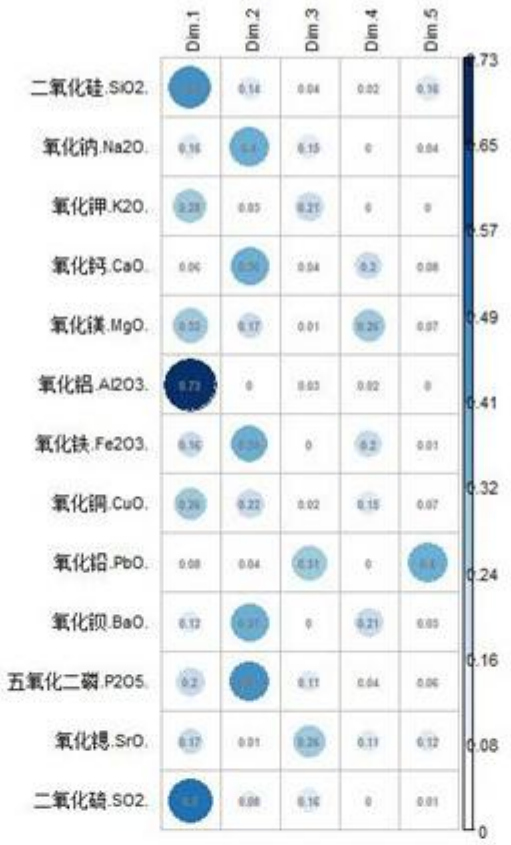
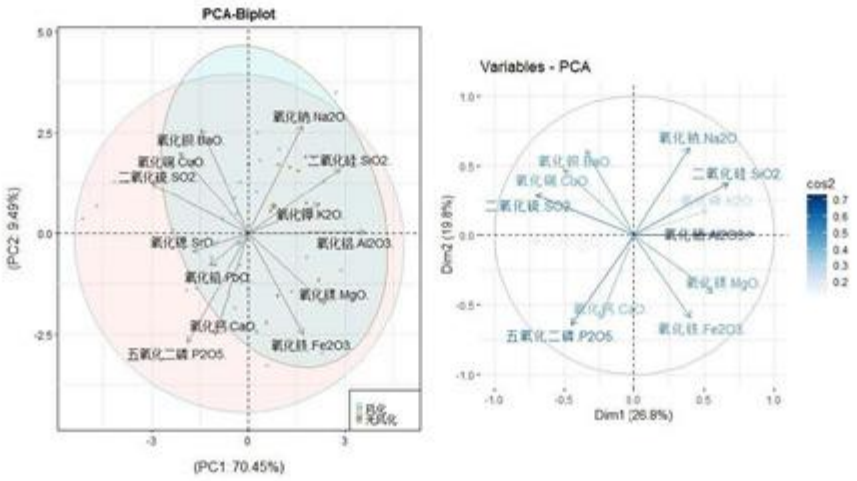


图 18　铅钡玻璃主成分载荷矩阵图

7.3 高钾玻璃和铅钡玻璃成分关系对比

- **变量对数比的大小方面**，铅钡玻璃整体上是高度变化的，但作为铅钡玻璃的主要成分之一的氧化铅相对含量却基本上没有变化，氯化钾和氧化锶也基本保持稳定；高钾玻璃变化大的变量相对较少，除去氧化钙、氧化锶、氧化铁、氧化钡，其它的相对含量都基本稳定。
- **变量对数比的相关性方面**，铅钡玻璃有强相关性的是氧化钡 (BaO)-氧化铜 (CuO)-氧化镁 (MgO)[负相关]-氧化铁 (Fe2O3)[负相关]、五氧化二磷 (P2O5)-氧化钙 (CaO)-氧化钠 (Na2O)[负相关]；高钾玻璃是二氧化硅 (SiO2)-氧化铝 (Al2O3)、氧化钠 (Na2O)-氧化钾 (K2O)-二氧化硫 (SO2)、氧化铁 (Fe2O3)-氧化铜 (CuO)。并且相对而言，高钾玻璃存在线性相关关系的变量对数比之间的相关系数更接近 1 或-1(余弦值绝对值更大)。



(a) 风化前后对比　　(b) 各指标之间对比

图 19　双标图（横纵坐标分别为第一、第二主成分）

附录 A　代码-R 语言

```
library(robCompositions)
library(ggplot2)
library(MASS)
library(mlr)
library(mvnormtest)
library(tidyverse)
library(plotly)
library(openxlsx)
library(tidyr)
library(ggpubr)

data = read.xlsx("C题数据.xlsx",sheet = 1)

for (i in 1:nrow(data)){
  data$累计[i] = sum(data[i,7:20],na.rm = T)
}
data = data[data$累计>=85&data$累计<=105,]
data = data[,-19]

for (i in 8:ncol(data)){
  data[,i][is.na(data[,i])] <- 0.04
  data[,i][data[,i]==0] = 0.04
}
data=tidyr::unite(data, "类型和风化程度", '类型', '部位风化', remove = FALSE)
data$类型 <- as.factor(data$类型)
data$表面风化 <- as.factor(data$表面风化)
```

```
data$部位风化 <- as.factor(data$部位风化)
data$类型和风化程度 = as.factor(data$类型和风化程度)


data_logcenter = data
data_logcenter[,8:20] = t(scale(t(log(data_logcenter[,8:20])),scale = F))

#############################################################################

#第一问属性数据分析#########################################################################
data1 = read.xlsx("C题数据.xlsx",sheet=3)

category = table(data1$类型,data1$表面风化)
addmargins(category)
chisq.test(category)

colour = table(data1$颜色,data1$表面风化)
addmargins(colour)
fisher.test(colour)

decoration = table(data1$纹饰,data1$表面风化)
addmargins(decoration)
fisher.test(decoration)


#############################################################################


#第一问预测#################################################################
unit = function(data){
  for(i in 1:nrow(data)){
    data[i,] = data[i,]/sum(data[i,])*100
  }
  return(data)
}

predict_Ba = read.xlsx("第一问预测2.0.xlsx",sheet = 1)
predict_Ba[,8:20] = unit(exp(predict_Ba[,8:20]))
write.xlsx(predict_Ba,"第一问铅钡预测.xlsx")

predict_K = read.xlsx("第一问预测2.0.xlsx",sheet = 2)
predict_K[,8:20] = unit(exp(predict_K[,8:20]))
write.xlsx(predict_K,"第一问高钾预测.xlsx")
#############################################################################

data_logcenter_K_no = data_logcenter[data_logcenter$类型 == "高钾" & data_logcenter$部位风化 ==
    "无风化",]
```

```
data_logcenter_K_with = data_logcenter[data_logcenter$类型 == "高钾" & data_logcenter$部位风化
    == "风化",]
data_logcenter_Ba_no = data_logcenter[data_logcenter$类型 == "铅钡" & data_logcenter$部位风化
    == "无风化",]
data_logcenter_Ba_with = data_logcenter[data_logcenter$类型 == "铅钡" & data_logcenter$部位风化
    == "风化",]

pairs(data_logcenter_K[,8:20],col = data_logcenter_K$部位风化)
pairs(data_logcenter_K[,8:20],col = data_logcenter_Ba$部位风化)

pairs(log(data[data$类型 == "高钾",8:20]),col = data$部位风化)


data_test1 = data_logcenter[,c(9,11:14)]
data_test2 = data[,c(9,11:15,18)]
data_test3 = log(data[data$部位风化=='无风化',c(9,11:15,18)])
pairs(data_test1)
pairs(data_test2)
pairs(data_test3)

#############################################################################
a = c(30,30,40)
b = c(30,10,60)
scale(log(a),scale = F)
scale(log(b),scale = F)
#############################################################################################

data = data_logcenter

g <-
    ggplot(data=data,aes(x=data$'类型',y=data$`二氧化硅(SiO2)`,fill=data$'部位风化',data$'类型'))
g1 <- g + geom_boxplot()+labs(x="玻璃种类",y =
    "二氧化硅(SiO2)含量")+theme_bw()+theme(legend.position = c(0.9,
    0.9),legend.title=element_blank())

g <- ggplot(data=data,aes(x=data$'类型',y=data$`氧化钠(Na2O)`,fill=data$'部位风化',data$'类型'))
g2 <- g + geom_boxplot()+labs(x="玻璃种类",y =
    "氧化钠(Na2O)含量")+theme_bw()+theme(legend.position = c(0.9,
    0.9),legend.title=element_blank())

g <- ggplot(data=data,aes(x=data$'类型',y=data$`氧化钾(K2O)`,fill=data$'部位风化',data$'类型'))
g3 <- g + geom_boxplot()+labs(x="玻璃种类",y =
    "氧化钾(K2O)含量")+theme_bw()+theme(legend.position = c(0.9,
    0.9),legend.title=element_blank())

g <- ggplot(data=data,aes(x=data$'类型',y=data$`氧化钙(CaO)`,fill=data$'部位风化',data$'类型'))
g4 <- g + geom_boxplot()+labs(x="玻璃种类",y =
```

```
    "氧化钙(CaO)含量")+theme_bw()+theme(legend.position = c(0.9,
    0.9),legend.title=element_blank())

g <- ggplot(data=data,aes(x=data$'类型',y=data$`氧化镁(MgO)`,fill=data$'部位风化',data$'类型'))
g5 <- g + geom_boxplot()+labs(x="玻璃种类",y =
    "氧化镁(MgO)含量")+theme_bw()+theme(legend.position = c(0.9,
    0.9),legend.title=element_blank())

g <- ggplot(data=data,aes(x=data$'类型',y=data$`氧化铝(Al2O3)`,fill=data$'部位风化',data$'类型'))
g6 <- g + geom_boxplot()+labs(x="玻璃种类",y =
    "氧化铝(Al2O3)含量")+theme_bw()+theme(legend.position = c(0.9,
    0.9),legend.title=element_blank())

g <- ggplot(data=data,aes(x=data$'类型',y=data$`氧化铁(Fe2O3)`,fill=data$'部位风化',data$'类型'))
g7 <- g + geom_boxplot()+labs(x="玻璃种类",y =
    "氧化铁(Fe2O3)含量")+theme_bw()+theme(legend.position = c(0.9,
    0.9),legend.title=element_blank())

g <- ggplot(data=data,aes(x=data$'类型',y=data$`氧化铜(CuO)`,fill=data$'部位风化',data$'类型'))
g8 <- g + geom_boxplot()+labs(x="玻璃种类",y =
    "氧化铜(CuO)含量")+theme_bw()+theme(legend.position = c(0.9,
    0.9),legend.title=element_blank())

g <- ggplot(data=data,aes(x=data$'类型',y=data$`氧化铅(PbO)`,fill=data$'部位风化',data$'类型'))
g9 <- g + geom_boxplot()+labs(x="玻璃种类",y =
    "氧化铅(PbO)含量")+theme_bw()+theme(legend.position = c(0.9,
    0.9),legend.title=element_blank())

g <- ggplot(data=data,aes(x=data$'类型',y=data$`氧化钡(BaO)`,fill=data$'部位风化',data$'类型'))
g10 <- g + geom_boxplot()+labs(x="玻璃种类",y = "氧化钡(BaO)")+theme_bw()+theme(legend.position
    = c(0.9, 0.9),legend.title=element_blank())

g <-
    ggplot(data=data,aes(x=data$'类型',y=data$`五氧化二磷(P2O5)`,fill=data$'部位风化',data$'类型'))
g11 <- g + geom_boxplot()+labs(x="玻璃种类",y =
    "五氧化二磷(P2O5)")+theme_bw()+theme(legend.position = c(0.9,
    0.9),legend.title=element_blank())

g <- ggplot(data=data,aes(x=data$'类型',y=data$`氧化锶(SrO)`,fill=data$'部位风化',data$'类型'))
g12 <- g + geom_boxplot()+labs(x="玻璃种类",y =
    "氧化锶(SrO)含量")+theme_bw()+theme(legend.position = c(0.9,
    0.9),legend.title=element_blank())

g <- ggplot(data=data,aes(x=data$'类型',y=data$`二氧化硫(SO2)`,fill=data$'部位风化',data$'类型'))
g14 <- g + geom_boxplot()+labs(x="玻璃种类",y =
    "二氧化硫(SO2)含量")+theme_bw()+theme(legend.position = c(0.9,
    0.9),legend.title=element_blank())
```

```
p <- ggarrange(g1,g2,g3,g4,g5,g6,g7,g8,g9,g10,g11,g12,ncol=3,nrow=4,common.legend =
    T,legend='right')


chemistry = names(data_logcenter)[8:20]
apply(data_logcenter_Ba_no[,8:20],MARGIN = 2,mean)
apply(data_logcenter_Ba_with[,8:20],MARGIN = 2,mean)
apply(data_logcenter_K_no[,8:20],MARGIN = 2,mean)
apply(data_logcenter_Ba_with[,8:20],MARGIN = 2,mean)

#第一问属性数据分析#############################################################################
data1 = read.xlsx("C题数据.xlsx",sheet=3)

category = table(data1$类型,data1$表面风化)
addmargins(category)
chisq.test(category)

colour = table(data1$颜色,data1$表面风化)
addmargins(colour)
fisher.test(colour)

decoration = table(data1$纹饰,data1$表面风化)
addmargins(decoration)
fisher.test(decoration)
#################################################################################

#第三问数据处理#################################################################################

data3 = read.xlsx("C题数据.xlsx",sheet=2)
data3 = data3[,-15]

for (i in 3:15){
  data3[,i][is.na(data3[,i])] = 0.04
  data3[,i][data3[,i]==0] = 0.04
}

data3[,3:15] = t(scale(t(log(data3[,3:15])),scale = F))

write.table(data3,"第三问.csv",sep=',',row.names = F)



#----------分层聚类------------#

library(ggplot2)
```

```
library(MASS)
library(mlr)
library(mvnormtest)
library(tidyverse)
library(plotly)
library(openxlsx)
library(tidyr)
library(ggpubr)
library(e1071)
library("pROC")


#筛选影响分层聚类的成分，保留“氧化钠”、“氧化钙”、“氧化镁”、“氧化铝”、“氧化铁”#
data_K = data[data$类型 == '高钾',]
data_K=data_K[,c(9,11:15,18)]
data_Ba = data[data$类型 == '铅钡',]
data_Ba=data_Ba[,c(9,11:15,18)]


#对高钾玻璃进行分层聚类#
hc_K=hclust(dist(data_K,method = "euclidean"),method = "ward.D2")
hc_K

plot(hc_K,hang = -0.01,cex=0.7)
h = locator(1)$y
rect.hclust(hc_K, h=h)
gr = cutree(hc_K, h=h)
plot(data_K,col=gr)


#对铅钡玻璃进行分层聚类#
hc_Ba=hclust(dist(data_Ba,method = "euclidean"),method = "ward.D2")
hc_Ba

plot(hc_Ba,hang = -0.01,cex=0.7)
h = locator(1)$y
rect.hclust(hc_Ba, h=h)
gr = cutree(hc_Ba, h=h)
plot(data_Ba,col=gr)



#----------PCA----------#

library(mvnormtest)
library(car)
library(ggplot2)
```

```
library(knitr)
library(kableExtra)
library(mvdalab)
library(psych)
library("readxl")
library("GPArotation")
library("ggpubr")
library(fmsb)
library(ggpubr)
library(tidyverse)
library(ggrepel)
library(factoextra)
library(RColorBrewer)
library(FactoMineR)
library(corrplot)
library(RColorBrewer)

data_n=read.csv('logcenter.csv')
data_ori=data_n[data_n$类型=='高钾',]
data_m=data_ori[,-c(1,2,3,4,5,6,7,21)]
data_m=as.data.frame(data_m)


fa.parallel(data_m,fa="pc",n.iter=100,show.legend=FALSE)

pca_m<-prcomp(data_m,center =FALSE)
summary(pca_m)

pca2_m = PCA(data_m,scale.unit = T,ncp=5,graph = T)
my_color = c('pink','seagreen','skyblue','plum')
colors = my_color[as.factor(data_ori$'表面风化')]
var_explained <- pca_m$sdev^2/sum(pca_m$sdev^2)
fviz_pca_biplot(pca2_m, axes = c(1, 2),geom.ind = c("point"),geom.var = c("arrow", "text"),
              pointshape = 16,pointsize=1,addEllipses = TRUE,
              label ="var",repel = TRUE,col.var = "grey50",
              labelsize=0.5,
              col.ind = data_ori$'表面风化')+
  scale_color_manual(values = colorRampPalette(brewer.pal(12,"Paired"))(4))+
  labs(x=paste0("(PC1: ",round(var_explained[1]*100,2),"%)"),
       y=paste0("(PC2: ",round(var_explained[2]*100,2),"%)"),
       title="PCA-Biplot")+
  theme(panel.background = element_rect(fill = 'white', colour = 'black'),
        axis.title.x = element_text(colour="black",size = 12,margin = margin(t=12)),
        axis.title.y = element_text(colour="black",size = 12,margin = margin(r=12)),
        axis.text=element_text(color="black"),
        plot.title = element_text(size=12,colour = "black",hjust=0.5,face = "bold"),
        legend.title = element_blank(),
```

```
      legend.key=element_blank(),
      legend.text = element_text(color="black",size=9),
      legend.spacing.x=unit(0.1,'cm'),
      legend.key.width=unit(0.2,'cm'),
      legend.key.height=unit(0.2,'cm'),
      legend.background=element_blank(),
      legend.box.background=element_rect(colour="black"),
      legend.position=c(1,0),legend.justification=c(1,0))
corrplot(get_pca_var(pca2_m)$cos2, is.corr=FALSE, col = brewer.pal(9, "Blues"),
    number.cex=0.5, addCoef.col="grey50", tl.col = "black", tl.cex = 0.7, cl.cex = 0.7)
fviz_pca_var(pca2_m,axes=c(1,2),
           col.var = "cos2",
           gradient.cols = brewer.pal(9, "Blues"),
           repel = TRUE)

pca_m$rotation



data_n=read.csv('logcenter.csv')
data_ori=data_n[data_n$类型=='铅锌',]
data_m=data_ori[,-c(1,2,3,4,5,6,7,21)]


fa.parallel(data_m,fa="pc",n.iter=100,show.legend=FALSE)


pca_m<-prcomp(data_m,center =FALSE)
summary(pca_m)


pca2_m = PCA(data_m,scale.unit = T,ncp=5,graph = T)
my_color = c('pink','seagreen','skyblue','plum')
colors = my_color[as.factor(data_ori$'表面风化')]
var_explained <- pca_m$sdev^2/sum(pca_m$sdev^2)
fviz_pca_biplot(pca2_m, axes = c(1, 2),geom.ind = c("point"),geom.var = c("arrow", "text"),
             pointshape = 16,pointsize=1,addEllipses = TRUE,
             label ="var",repel = TRUE,col.var = "grey50",
             labelsize=0.5,
             col.ind = data_ori$'表面风化')+
  scale_color_manual(values = colorRampPalette(brewer.pal(12,"Paired"))(4))+
  labs(x=paste0("(PC1: ",round(var_explained[1]*100,2),"%)"),
      y=paste0("(PC2: ",round(var_explained[2]*100,2),"%)"),
      title="PCA-Biplot")+
  theme(panel.background = element_rect(fill = 'white', colour = 'black'),
       axis.title.x = element_text(colour="black",size = 12,margin = margin(t=12)),
       axis.title.y = element_text(colour="black",size = 12,margin = margin(r=12)),
```

```
      axis.text=element_text(color="black"),
      plot.title = element_text(size=12,colour = "black",hjust=0.5,face = "bold"),
      legend.title = element_blank(),
      legend.key=element_blank(),
      legend.text = element_text(color="black",size=9),
      legend.spacing.x=unit(0.1,'cm'),
      legend.key.width=unit(0.2,'cm'),
      legend.key.height=unit(0.2,'cm'),
      legend.background=element_blank(),
      legend.box.background=element_rect(colour="black"),
      legend.position=c(1,0),legend.justification=c(1,0))
corrplot(get_pca_var(pca2_m)$cos2, is.corr=FALSE, col = brewer.pal(9, "Blues"),
    number.cex=0.5, addCoef.col="grey50", tl.col = "black", tl.cex = 0.7, cl.cex = 0.7)
fviz_pca_var(pca2_m,axes=c(1,2),
           col.var = "cos2",
           gradient.cols = brewer.pal(9, "Blues"),
           repel = TRUE)

pca_m$rotation



#######################################################
#箱线图&散点图#####################################
library(ggplot2)
library(MASS)
library(mlr)
library(mvnormtest)
library(tidyverse)
library(plotly)
library(openxlsx)
library(tidyr)
library(ggpubr)
library(e1071)
library("pROC")
setwd("C:\\Users\\Qiu\\Desktop\\R for MCM")
data = read.csv("logcenter.csv")

for (i in 8:ncol(data)){
  data[,i][is.na(data[,i])] <- 0
}
data_1=data[1:67,c(4,8:20)]

#SVM 数据导入
data_un = data[data$部位风化 == "无风化",]
data_un = data_un[1:35,c(4,8:20)]
data_un_K = read.csv("高钾_无风化.csv")
```

```
data_un_K = data_un_K[,c(4,8:20)]
data_un_P = read.csv("铅钡_无风化.csv")
data_un_P = data_un_P[,c(4,8:20)]
data_pre_K = read.csv("高钾_预测结果.csv")
data_pre_K = data_pre_K[,c(4,8:20,23)]
data_pre_P = read.csv("铅钡_预测结果.csv")
data_pre_P = data_pre_P[,c(4,8:20,23)]
data_2 = read.csv("第三问预测2.csv")
data_3 = data_2
data_3 = data_3[,c(2,4:16)]
data_3_K = data_3[c(1,7,8,11,12),]
data_3_P = data_3[-c(1,7,8,11,12),]


#SVM支持向量机(大类划分)
data_un$类型<-as.factor(data_un$类型)
svm_model_un=svm(类型~氧化铅.PbO.+氧化钡.BaO.,data=data_un,kernel = "linear")
summary(svm_model_un)
plot(svm_model_un,data_un, 氧化钡.BaO.~氧化铅.PbO.)
w=t(svm_model_un$coefs)%*%svm_model_un$SV
b = -svm_model_un$rho
-1/w[1,2]
-w[1,1]/w[1,2]
abline(a = -b/w[1,2],b=-w[1,1]/w[1,2])
p1 = ggplot(data_un) +
  aes(x = 氧化铅.PbO., y = 氧化钡.BaO., colour = 类型) +
  geom_point(shape = "circle", size = 3) +
  scale_color_hue(direction = 1) +
  theme_bw()+
  theme(legend.title=element_blank())+
  xlab("氧化铅(PbO)")+
  ylab("氧化钡(BaO)")+
  ggtitle("SVM分类器——高钾、铅钡分类规律") +
  theme(plot.title = element_text(hjust = 0.5))+
  annotate(geom="text", parse=T, label='y = -3.51x + 5.97')+
  geom_abline(intercept=5.965778,slope = -3.514822, lwd=1.5)
p1


data_un_K$氧化钠.Na2O.

#SVM支持向量机（高钾亚类划分）
data_un_K$类型="高钙低镁"
data_un_K$类型[6]= "低钙高镁"
data_un_K$类型[11]= "低钙高镁"
data_un_K$类型<-as.factor(data_un_K$类型)
svm_model_K=svm(data_un_K$类型~data_un_K$氧化钙.CaO.+data_un_K$氧化镁.MgO.,data=data_un_K,kernel
```

```
    = "linear")
summary(svm_model_K)
p2 = ggplot(data_un_K) +
  aes(x = 氧化钙.CaO., y = 氧化镁.MgO., colour = data_un_K$类型) +
  geom_point(shape = "circle", size = 3) +
  scale_color_hue(direction = 1) +
  theme_bw()+
  theme(legend.title=element_blank())+
  xlab("氧化钙(CaO)")+
  ylab("氧化镁(MgO)")+
  ggtitle("SVM分类器——高钾亚类分析结果") +
  theme(plot.title = element_text(hjust = 0.5))+
  geom_abline(intercept= 2.099382 ,slope = 1.95199, lwd=1.5 )+
  geom_abline(intercept= 1.599382 ,slope = 1.95199,lty=2)+
  geom_abline(intercept= 2.599382 ,slope = 1.95199,lty=2)+
  geom_abline(intercept= 1.099382 ,slope = 1.95199,lty=3)+
  geom_abline(intercept= 3.099382 ,slope = 1.95199,lty=3)
p2



#SVM支持向量机（铅钡亚类划分）
data_un_P$类型="低钠"
data_un_P$类型[2]= "高钠"
data_un_P$类型[4]= "高钠"
data_un_P$类型[14]= "高钠"
data_un_P$类型[15]= "高钠"
data_un_P$类型[16]= "高钠"
data_un_P$类型[17]= "高钠"
data_un_P$类型[19]= "高钠"
data_un_P$类型[22]= "高钠"
data_un_P$类型[23]= "高钠"

data_un_P$类型<-as.factor(data_un_P$类型)
x=data_un_P[,c(3,5)]
y=data_un_P$类型
svm_model_P=svm(x,y,data=data_un_P,kernel = "linear")
summary(svm_model_P)
w=t(svm_model$coefs)%*%svm_model$SV
b = -svm_model$rho
-1/w[1,2]
-w[1,1]/w[1,2]
p3 = ggplot(data_un_P) +
  aes(x = 氧化钙.CaO., y = 氧化钠.Na2O., colour = data_un_P$类型) +
  geom_point(shape = "circle", size = 3) +
  scale_color_hue(direction = 1) +
  theme_bw()+
```

```
  theme(legend.title=element_blank())+
  xlab("氧化钙(CaO)")+
  ylab("氧化钠(Na2O)")+
  ggtitle("SVM分类器——铅钡亚类分析结果") +
  theme(plot.title = element_text(hjust = 0.5))+
  geom_abline(intercept= 0.09629258 ,slope = 0.7481686, lwd=1.5 )+
  geom_abline(intercept= 0.59629258 ,slope = 0.7481686,lty=2)+
  geom_abline(intercept= -0.40370742 ,slope = 0.7481686,lty=2)+
  geom_abline(intercept= 1.09629258 ,slope = 0.7481686,lty=3)+
  geom_abline(intercept= -0.90370742 ,slope = 0.7481686,lty=3)
p3

#SVM支持向量机（无风化预测数据划分）
pred=predict(svm_model_K,newdata = data_pre_K[,5:6])
pred=predict(svm_model_P,newdata = data_pre_P)
pred

data_pre_P$类型 = as.factor(data_pre_P$类型)

data_pre_K$类型 = "高钙低镁"
data_pre_K[c(12,17),1] = "低钙高镁"
data_pre_K$类型 = as.factor(data_pre_K$类型)
data_pre_K$shape = as.factor(data_pre_K$shape)
p1 = ggplot(data_pre_K) +
  aes(x = 氧化钙.CaO., y = 氧化镁.MgO., colour = 类型, shape = shape) +
  geom_point(size = 3) +
  scale_color_hue(direction = 1) +
  theme_bw()+
  theme(legend.title=element_blank())+
  xlab("氧化钙(CaO)")+
  ylab("氧化镁(MgO)")+
  ggtitle("SVM分类器——高钾玻璃亚类划分") +
  theme(plot.title = element_text(hjust = 0.5))+
  geom_abline(intercept= 2.099382 ,slope = 1.95199,lwd = 1.5 )+
  geom_abline(intercept= 1.599382 ,slope = 1.95199,lty=2)+
  geom_abline(intercept= 2.599382 ,slope = 1.95199,lty=2)+
  geom_abline(intercept= 1.099382 ,slope = 1.95199,lty=3)+
  geom_abline(intercept= 3.099382 ,slope = 1.95199,lty=3)
p1


data_pre_P$类型="低钠"
data_pre_P$类型[which(data_pre_P$氧化钠.Na2O.>0)]= "高钠"
data_pre_P$类型[23]= "高钠"
data_pre_P$类型 = as.factor(data_pre_P$类型)
p2 = ggplot(data_pre_P) +
  aes(x = 氧化钙.CaO., y = 氧化钠.Na2O., colour = 类型,shape = shape) +
```

```
  geom_point(size = 3) +
  scale_color_hue(direction = 1) +
  theme_bw()+
  theme(legend.title=element_blank())+
  xlab("氧化钙(CaO)")+
  ylab("氧化钠(Na2O)")+
  ggtitle("SVM分类器——铅钡玻璃亚类划分") +
  theme(plot.title = element_text(hjust = 0.5))+
  geom_abline(intercept= 0.09629258 ,slope = 0.7481686,lwd = 1.5)+
  geom_abline(intercept= 0.59629258 ,slope = 0.7481686,lty=2)+
  geom_abline(intercept= -0.40370742 ,slope = 0.7481686,lty=2)+
  geom_abline(intercept= 1.09629258 ,slope = 0.7481686,lty=3)+
  geom_abline(intercept= -0.90370742 ,slope = 0.7481686,lty=3)
p2

p <- ggarrange(p1,p2)
p

#SVM支持向量机（第三问数据划分）
label = data_2$文物编号
p1 = ggplot(data_3) +
  aes(x = 氧化铅.PbO., y = 氧化钡.BaO., colour = 类型 ) +
  scale_fill_manual(values = colors)+
  geom_point(shape = "circle", size = 3) +
  theme_bw()+
  theme(legend.title=element_blank())+
  geom_text_repel(aes(label = label),nudge_y = -0.25)+
  xlab("氧化铅(PbO)")+
  ylab("氧化钡(BaO)")+
  ggtitle("SVM分类器——高钾、铅钡玻璃划分") +
  theme(plot.title = element_text(hjust = 0.5))+
  geom_abline(intercept=5.965778,slope = -3.514822,lwd = 1.5)+
  geom_abline(intercept= 5.465778 ,slope = -3.514822,lty=2)+
  geom_abline(intercept= 6.465778 ,slope = -3.514822,lty=2)+
  geom_abline(intercept= 6.965778 ,slope = -3.514822,lty=3)+
  geom_abline(intercept= 4.965778 ,slope = -3.514822,lty=3)
p1

data_3_K$类型="高钙低镁"
p2 = ggplot(data_3_K) +
  aes(x = 氧化钙.CaO., y = 氧化镁.MgO., colour = 类型) +
  geom_point(shape = "circle", size = 3) +
  scale_color_hue(direction = 1) +
  ylim(-4, 3)+
  xlim(-2,3)+
  theme_bw()+
  theme(legend.title=element_blank())+
```

```
  geom_text(aes(label = c("A1","A6","A7","A6_before","A7_before")),nudge_y = -0.25)+
  xlab("氧化钙(CaO)")+
  ylab("氧化镁(MgO)")+
  ggtitle("SVM分类器——高钾玻璃亚类划分") +
  theme(plot.title = element_text(hjust = 0.5))+
  geom_abline(intercept= 2.099382 ,slope = 1.95199,lwd = 1.5 )+
  geom_abline(intercept= 1.599382 ,slope = 1.95199,lty=2)+
  geom_abline(intercept= 2.599382 ,slope = 1.95199,lty=2)+
  geom_abline(intercept= 1.099382 ,slope = 1.95199,lty=3)+
  geom_abline(intercept= 3.099382 ,slope = 1.95199,lty=3)
p2

data_3_P$类型="低钠"
data_3_P$类型[7]= "高钠"
p3 = ggplot(data_3_P) +
  aes(x = 氧化钙.CaO., y = 氧化钠.Na2O., colour = 类型) +
  geom_point(shape = "circle", size = 3) +
  scale_color_hue(direction = 1) +
  theme_bw()+
  theme(legend.title=element_blank())+
  geom_text(aes(label = c("A3","A4","A8","A2","A5","A2_before","A5_before")),nudge_y =
      -0.15,nudge_x = 0.3)+
  xlab("氧化钙(CaO)")+
  ylab("氧化钠(Na2O)")+
  ggtitle("SVM分类器——铅钡玻璃亚类划分") +
  theme(plot.title = element_text(hjust = 0.5))+
  geom_abline(intercept= 0.09629258 ,slope = 0.7481686,lwd = 1.5)+
  geom_abline(intercept= 0.59629258 ,slope = 0.7481686,lty=2)+
  geom_abline(intercept= -0.40370742 ,slope = 0.7481686,lty=2)+
  geom_abline(intercept= 1.09629258 ,slope = 0.7481686,lty=3)+
  geom_abline(intercept= -0.90370742 ,slope = 0.7481686,lty=3)
p3

#data=tidyr::unite(data, "类型和风化程度", '类型', '表面风化', remove = FALSE)#



g <-
    ggplot(data=data,aes(x=data$'类型',y=data$`二氧化硅(SiO2)`,fill=data$'表面风化',data$'类型'))
g1 <- g + geom_boxplot()+labs(x="玻璃种类",y =
    "二氧化硅(SiO2)含量")+theme_bw()+theme(legend.position = c(0.9,
    0.9),legend.title=element_blank())

g <- ggplot(data=data,aes(x=data$'类型',y=data$`氧化钠(Na2O)`,fill=data$'表面风化',data$'类型'))
g2 <- g + geom_boxplot()+labs(x="玻璃种类",y =
    "氧化钠(Na2O)含量")+theme_bw()+theme(legend.position = c(0.9,
    0.9),legend.title=element_blank())
```

```
g <- ggplot(data=data,aes(x=data$'类型',y=data$`氧化钾(K2O)`,fill=data$'表面风化',data$'类型'))
g3 <- g + geom_boxplot()+labs(x="玻璃种类",y =
    "氧化钾(K2O)含量")+theme_bw()+theme(legend.position = c(0.9,
    0.9),legend.title=element_blank())

g <- ggplot(data=data,aes(x=data$'类型',y=data$`氧化钙(CaO)`,fill=data$'表面风化',data$'类型'))
g4 <- g + geom_boxplot()+labs(x="玻璃种类",y =
    "氧化钙(CaO)含量")+theme_bw()+theme(legend.position = c(0.9,
    0.9),legend.title=element_blank())

g <- ggplot(data=data,aes(x=data$'类型',y=data$`氧化镁(MgO)`,fill=data$'表面风化',data$'类型'))
g5 <- g + geom_boxplot()+labs(x="玻璃种类",y =
    "氧化镁(MgO)含量")+theme_bw()+theme(legend.position = c(0.9,
    0.9),legend.title=element_blank())

g <- ggplot(data=data,aes(x=data$'类型',y=data$`氧化铝(Al2O3)`,fill=data$'表面风化',data$'类型'))
g6 <- g + geom_boxplot()+labs(x="玻璃种类",y =
    "氧化铝(Al2O3)含量")+theme_bw()+theme(legend.position = c(0.9,
    0.9),legend.title=element_blank())

g <- ggplot(data=data,aes(x=data$'类型',y=data$`氧化铁(Fe2O3)`,fill=data$'表面风化',data$'类型'))
g7 <- g + geom_boxplot()+labs(x="玻璃种类",y =
    "氧化铁(Fe2O3)含量")+theme_bw()+theme(legend.position = c(0.9,
    0.9),legend.title=element_blank())

g <- ggplot(data=data,aes(x=data$'类型',y=data$`氧化铜(CuO)`,fill=data$'表面风化',data$'类型'))
g8 <- g + geom_boxplot()+labs(x="玻璃种类",y =
    "氧化铜(CuO)含量")+theme_bw()+theme(legend.position = c(0.9,
    0.9),legend.title=element_blank())

g <- ggplot(data=data,aes(x=data$'类型',y=data$`氧化铅(PbO)`,fill=data$'表面风化',data$'类型'))
g9 <- g + geom_boxplot()+labs(x="玻璃种类",y =
    "氧化铅(PbO)含量")+theme_bw()+theme(legend.position = c(0.9,
    0.9),legend.title=element_blank())

g <- ggplot(data=data,aes(x=data$'类型',y=data$`氧化钡(BaO)`,fill=data$'表面风化',data$'类型'))
g10 <- g + geom_boxplot()+labs(x="玻璃种类",y = "氧化钡(BaO)")+theme_bw()+theme(legend.position
    = c(0.9, 0.9),legend.title=element_blank())

g <-
    ggplot(data=data,aes(x=data$'类型',y=data$`五氧化二磷(P2O5)`,fill=data$'表面风化',data$'类型'))
g11 <- g + geom_boxplot()+labs(x="玻璃种类",y =
    "五氧化二磷(P2O5)")+theme_bw()+theme(legend.position = c(0.9,
    0.9),legend.title=element_blank())

g <- ggplot(data=data,aes(x=data$'类型',y=data$`氧化锶(SrO)`,fill=data$'表面风化',data$'类型'))
```

```
g12 <- g + geom_boxplot()+labs(x="玻璃种类",y =
    "氧化锶(SrO)含量")+theme_bw()+theme(legend.position = c(0.9,
    0.9),legend.title=element_blank())

g <- ggplot(data=data,aes(x=data$'类型',y=data$`氧化锡(SnO2)`,fill=data$'表面风化',data$'类型'))
g13 <- g + geom_boxplot()+labs(x="玻璃种类",y =
    "氧化锡(SnO2)")+theme_bw()+theme(legend.position = c(0.9, 0.9),legend.title=element_blank())

g <- ggplot(data=data,aes(x=data$'类型',y=data$`二氧化硫(SO2)`,fill=data$'表面风化',data$'类型'))
g14 <- g + geom_boxplot()+labs(x="玻璃种类",y =
    "二氧化硫(SO2)含量")+theme_bw()+theme(legend.position = c(0.9,
    0.9),legend.title=element_blank())


p <- ggarrange(g1,g2,g3,g4,g5,g6,g7,g8,g9,g10,g11,g12,g13,g14,ncol=3,nrow=5,common.legend =
    T,legend='right')
p


data$类型 <- as.factor(data$类型)
p1 = ggplot(data) +
  aes(x = `氧化钾(K2O)`, y = `二氧化硅(SiO2)`, colour = data$类型) +
  geom_point(shape = "circle", size = 1.5) +
  scale_color_hue(direction = 1) +
  theme_minimal()+
  theme(legend.title=element_blank())+
  ggtitle("Box Plus Minus vs. MVP Share") +
  theme(plot.title = element_text(hjust = 0.5))
p1

p2 = ggplot(data_1) +
  aes(x = 氧化铅.PbO., y = 氧化钡.BaO., colour = data_1$类型) +
  geom_point(shape = "circle", size = 1.5) +
  scale_color_hue(direction = 1) +
  theme_minimal()+
  theme(legend.title=element_blank())+
  ggtitle("Box Plus Minus vs. MVP Share") +
  theme(plot.title = element_text(hjust = 0.5))
p2


p2 = ggplot(data_un) +
  aes(x = 氧化铅.PbO., y = 氧化钡.BaO., colour = 类型) +
  geom_point(shape = "circle", size = 3) +
  scale_color_hue(direction = 1) +
  theme_bw()+
  theme(legend.title=element_blank())+
```

```
  xlab("氧化铅(PbO)")+
  ylab("氧化钡(BaO)")+
  ggtitle("SVM分类器——高钾与铅钡分类规律") +
  theme(plot.title = element_text(hjust = 0.5))+
  geom_abline(intercept=5.965778,slope = -3.514822)
p2

#交叉验证

split_train_test<-function(i,dataframe){
  test_data<<-dataframe[10:20,]
  train_data<<-dataframe[-c(10:20),]
}
count=0
for(i in 1:67){
  split_train_test(1,data1)
  svm_model=svm(x-.,data=train_data,knernel="radial")
  svm_pred=predict(svm_model,newdata=test_data)
  count = count+as.numeric(svm_pred==test_data[1,])
}
print(svm_model)
plot(svm_model, train_data)
obs_p_svm = data.frame(prob=svm_pred,obs=test_data$x)
table(test_data$x,svm_pred,dnn=c("真实值","预测值"))
###绘制ROC曲线
svm_roc <- roc(test_data$x,as.numeric(svm_pred))
plot(svm_roc, print.auc=TRUE, auc.polygon=TRUE, grid=c(0.1, 0.2),grid.col=c("green", "red"),
    max.auc.polygon=TRUE,auc.polygon.col="skyblue", print.thres=TRUE,main='SVM模型ROC曲线
    kernel = radial')



#SVM支持向量机
data_un$类型<-as.factor(data_un$类型)
svm_model=svm(类型~氧化铅.PbO.+氧化钡.BaO.,data=data_un,kernel = "linear")
summary(svm_model)
plot(svm_model,data_un, 氧化钡.BaO.~氧化铅.PbO.)
plot(svm_model,data_1, 氧化钡.BaO.~氧化钾.K2O.)
plot(svm_model,data_1, 氧化钾.K2O.~氧化铅.PbO.)

svm_model=svm(类型~氧化钙.CaO.+氧化镁.MgO.,data=data_un,kernel = "linear")
plot(svm_model,data_un, 氧化钙.CaO.~氧化镁.MgO.)

w=t(svm_model$coefs)%*%svm_model$SV
b = -svm_model$rho
abline(a = -b/w[1,2],b=-w[1,1]/w[1,2])
-b/w[1,2]
```

```
-w[1,1]/w[1,2]
svm_model
-1/w[1,2]
data_1$氧化铅.PbO.

svm_pred=predict(svm_model,newdata=data_un)
a=a+as.numeric(svm_pred=="高钾")

data_un$svm_pred=svm_pred

head(data2)
table(data_un$'类型',data_un$svm_pred)
###绘制ROC曲线
svm_roc <- roc(data_un$'类型',as.numeric(svm_pred))
plot(svm_roc, print.auc=TRUE, auc.polygon=TRUE, grid=c(0.1, 0.2),grid.col=c("green", "red"),
    max.auc.polygon=TRUE,auc.polygon.col="skyblue", print.thres=TRUE,main='SVM模型ROC曲线
    kernel = radial')

#logistic回归

data1 <- data[,c(3,7:20)]
names(data1)[names(data1) =="类型"] <-"x"
names(data1)[names(data1) =="氧化铅(PbO)"] <-"x1"
names(data1)[names(data1) =="氧化钡(BaO)"] <-"x2"
data1$x[which(data1$x=="无风化")]<-0
data1$x[which(data1$x=="风化")]<-1
data1$x<-as.numeric(data1$x)
data1$x<-as.factor(data1$x)

log_res<-glm(x~.,family=binomial(link='logit'),data = train_data,control=list(maxit=100))
summary(log_res)

anova(log_res,test = "Chisq")

predictest = predict(log_res,newdata = test_data,type = "response")
predictest <- ifelse(predictest > 0.5,1,0)
predictest
n = length(predictest)

correct=0
for(i in 2:n){
  if(predictest[i] <= predictest[1]){
    answer = TRUE
    answer
  }else{
    answer = FALSE
    answer
```

```
    break
  }
}

kmeans(data_un_K,2)

library(ggfortify)
autoplot(kmeans(data1,2), data = data1, label=TRUE, label.size = 3,frame = TRUE)

install.packages("ggforify")
```