基于聚类分析的玻璃文物分类模型

摘要

本文针对玻璃文物成分分析以及类别划分的多元分析问题，基于主成分分析、K-means 聚类、层次聚类，建立了数学模型，为考古工作者在发掘作业后判断玻璃文物的类型提供指导。

针对问题一，解决了三个问题。第一利用**卡方检测**根据**显著性 P 值**是否小于 **0.05** 判断出玻璃文物表面风化与玻璃类型相关性较强，而玻璃纹饰其次，并且与玻璃的颜色无关。第二分别选取玻璃类型、纹饰类型作为固定变量，对文物表面有无风化化学成分含量统计规律作出可视化分析。第三选取玻璃类型作为固定变量，分别构建不同玻璃类型文物风化前后化学成分含量的**平均比例关系模型**，对风化前化学成分含量做出了预测。

针对问题二，选择使用 **K-means 聚类分析**，分析当聚类数为 2 时，两种聚类的化学成分差异。可以得到高钾玻璃和铅钡玻璃在氧化铅，二氧化硅，氧化钡，氧化钾，氧化钙，氧化铝的含量差异较大，说明两类玻璃是以这些化合物为依据划分的。接着使用**层次聚类分析**，对无风化的高钾类和铅钡类玻璃进行划分，分别得到**最优聚类数**为 **5** 和 **4**，最后进行**灵敏度分析**，分别改变两种划分中的其中一种亚类的某种主要化学成分的含量，进行**扰动处理**，得出模型的敏感性表现较好。

针对问题三，首先使用**主成分分析**对化学成分进行**降维处理**，再使用**支持向量机模型**对已知数据进行训练，再使用训练后的模型对表单三中的数据进行预测并得出预测结果。接着对数据进行扰动处理，分别将变量值随机缩放 **5%、10%、20%、30%**，将得到的结果与扰动前的结果进行对比，得出结果变化率为 0%，说明模型具有较好的稳定性。

针对问题四，使用 **R 型 K-means 聚类算法**进行求解，首先以**总体样本平方和**为依据，得出每一个类别玻璃最佳的聚类个数，即 K 的值。再根据聚类结果分析不同类别的化学成分的**关联度**。接着对聚类结果以 CH，DBI，SC 为标准比较不同类别化学成分关联度之间的差异性，发现无论是高钾类玻璃还是铅钡类玻璃分化后变量之间的**关联度均增强**，并且通过比较 CH 值发现，高钾类玻璃内化学成分的关联度比铅钡类玻璃内化学成分的关联度**更强**。

关键词:卡方检验，层次聚类，支持向量机，K-means 聚类，主成分分析。

一、问题重述

1.1 问题背景

玻璃主要由石英砂制成，其主要成分是 SiO_2。纯石英砂炼制时需要加入助熔剂以降低其熔化温度。并加入石灰石作为稳定剂。石灰石煅烧后转化为 CaO。使用的助熔剂不同，玻璃的主要成分也不同。例如，铅钡玻璃以铅矿石作为助熔剂，其 PbO、BaO 含量较高。钾玻璃以草木灰等含钾量高的材料作为助熔剂，K 含量较高。风化程度太高的玻璃会影响对其分类的准确判断。

已给出一批古代玻璃制品的相关数据，且已将其分为高钾玻璃和铅钡玻璃两类。附件表单 2 各主要成分所占比例累加和介于 85%~105%的数据视为有效。

1.2 问题要求

问题一 对玻璃文物表面风化与其玻璃种类、纹饰、颜色进行相关性分析；分析文物样品表面有无风化的化学成分含量的统计规律，根据风化点的测试数据，预测风化前玻璃的的化学成分含量。

问题二 根据附件 2 数据分析高钾玻璃和铅钡玻璃的分类规则；对于每一类，选择合适的化学成分进行亚分类，给出具体的分类方法和结果，并分析分类结果的合理性和敏感性。

问题三 分析附件表 3 中未知玻璃文物的化学成分，鉴定其类型，分析分类结果的敏感性。

问题四 根据不同类型的玻璃文物样品，分析其化学成分之间的相关性，比较不同类型之间化学成分相关性的差异。

二、问题分析

2.1 对问题一的分析

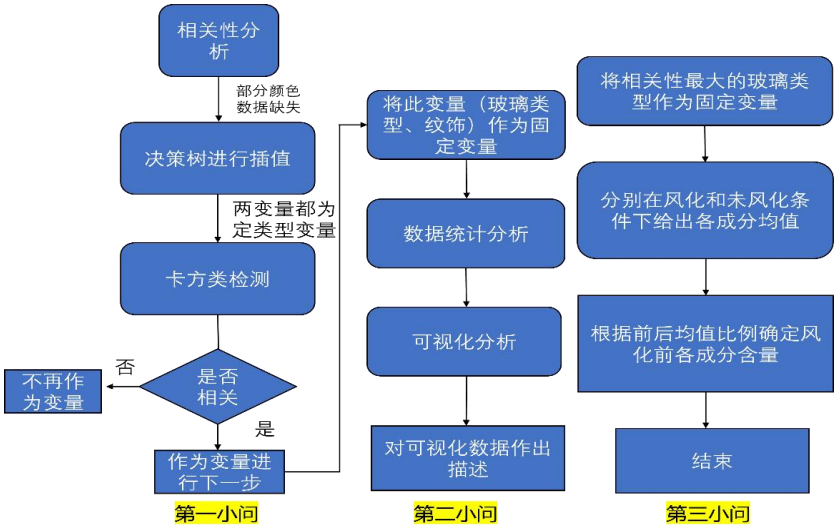
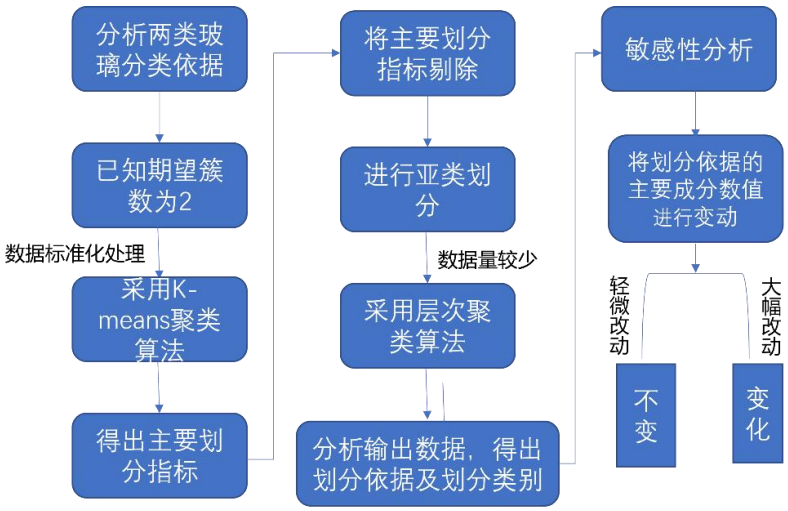


图 2.1-1 问题一分析流程图

第一小问要求对“文物表面风化”与其“玻璃类型”、“纹饰”和“颜色”的关系进行分析，是一道相关性分析问题。由于“文物表面分化与否”，“玻璃类型”、“纹饰”和“颜色”均为定类型变量，于是对其分析采用卡方检验。观察数据发现某些数据的颜色空缺，且数量较多不宜删除，因此采用插值法利用决策树模型补齐颜色空缺。

第二小问要求分析文物表面有无风化化学含量的统计规律。是一道数据统计分析类问题。采用可视化分析，首先将已风化与未风化文物样品表面的化学成分含量以表格以及图表的形式呈现，再对呈现出的可视化统计数据进行分析。

第三小问要求根据已知风化点检测数据，预测其风化前的化学成分含量。首先固定玻璃类型这一变量，由于风化与否与纹饰类型以及颜色相关性较低，因此不再做考虑。分别给出在高钾类和铅钡类中，有风化和无风化条件下各成分含量平均值。根据各成分风化前后含量均值的比例来确定风化前各成分含量。

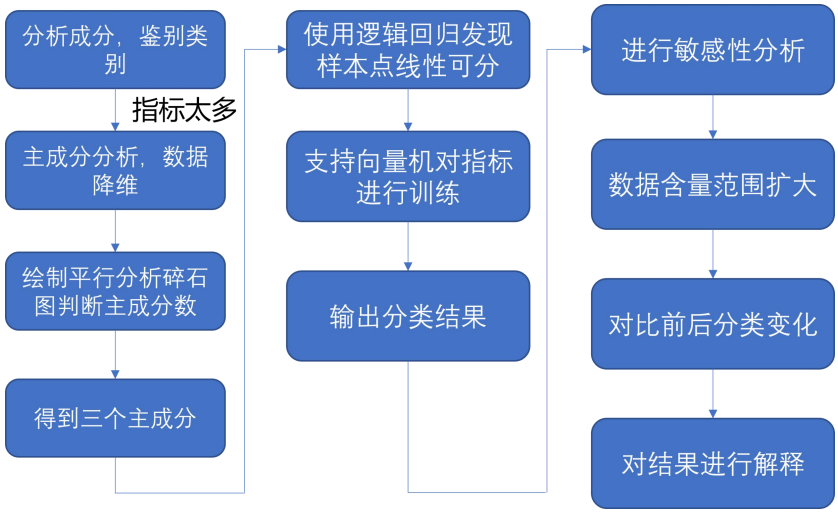


2.2 对问题二的分析

图 2.2-1 问题二分析流程图

第一小问要求分析高钾玻璃、铅钡玻璃的分类规律。问题一可视化分析数据其实也可以对两类玻璃的分类规律作出解释，但结果精确度不高。已知分成两类，则采用 k-means 聚类算法。令 K=2，得出划分类别的主要指标数据。

第二小问要求对选择合适的化学成分对每个类别进行亚类划分。即分别在高钾类以及铅钡类中选取其它化学成分进行类别的再次划分。因此先把第一小问中划分高钾类以及铅钡类的主要指标剔除，再根据剩下的指标化学成分含量的多少或是化学成分的有无进行亚类划分。因此我们采用聚类分析的方法，由于本题所给数据量较小，可以每个样本为中心进行聚类。因此采用层次聚类分析模型。



2.3 对问题三的分析

图 2.3-1 问题三分析流程图

题中要求对未知类别文物的化学成分进行分析并鉴别所属类别。由于化学成分较多，携带的信息有重复，于是采用主成分分析法进行数据降维。在使用逻辑回归时发现样本点之间是线性可分的，因此使用线性可分向量机来训练，利用机器学习对未知类别文物进行分类。

对于敏感性分析，对数据进行扰动处理，将数据值分别扩大一定百分比，观察文物类别前后变化，以判断模型敏感性。

2.4 对问题四的分析

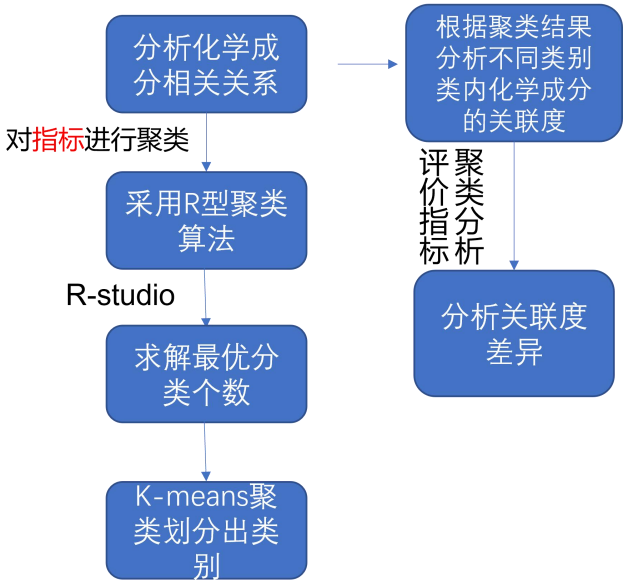


图 2.4-1 问题四分析

题目要求对不同类别玻璃文物样品给出化学成分之间的关联关系，即将指标进行聚类。因此采用 R 型 K-means 聚类分析法。在聚类前先使用 R 语言软件根

据整体样本平方和进行最优分类个数求解。通过分析结果分析相关关系。

根据聚类分析的评价指标去对比分析不同类别化学成分的关联度差异。

三、模型假设

假设一 所检测到的数值仅反映文物当前表面化学成分含量，与环境化学成分无关。

假设二 表面风化文物的未风化区域可视为无风化数据处理。

假设三 未检测到的成分即视为不含该成分，即数据为 0。

四、符号说明

符号	说明
f_{s_i}	指标实际观测频次
f_{q_i}	指标期望观测频次
a_j	某成分风化后的含量
a_i	某成分风化前含量
α_i	文物样本
x_{im}	文物各成分含量
$d(i,j)$	不同簇之间的距离

五、模型的建立与求解

5.1 数据预处理

根据题意，玻璃文物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很大改变，不再具有参考价值。因此将风化程度太高的数据删除。其次成分比例累加之和位于 85%~105%为有效数据，因此将此范围以外的数据删除。

把表面风化的样本检测部位的无风化点当做无风化来处理。

将未检测到的数据均看成不存在此成分，即将数值视为 0.

5.1.1 问题一数据预处理

观察表格数据发现某些数据的颜色空缺，且数量较多不宜删除，因此利用插值法补齐颜色空缺。

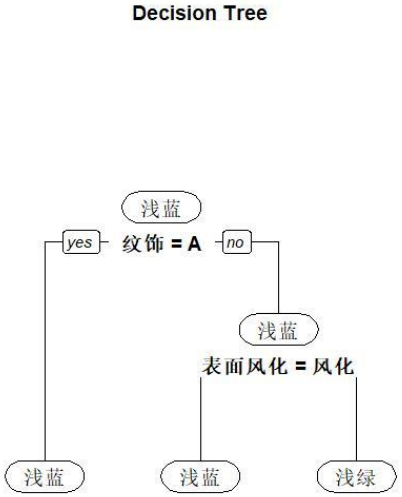


图 5.1-1 决策树填补颜色空缺原理

5.1.2 问题二数据预处理

对数据进行标准化处理，使其均值为 0，方差为 1，且无量纲。方法如下

$$y_i = \frac{x_i - \bar{x}}{s} \tag{1}$$

$$\bar{x} = \frac{1}{n}\sum^{n} x_i, s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{2}$$

5.2 问题一模型的建立与求解

5.2.1 卡方检验模型的建立

首先玻璃文物风化与否以及文物的玻璃类型、纹饰、颜色都是定类型变量。定类与定类型变量进行相关性分析，卡方检测的误差值最小。因此我们构造模型进行卡方检测。

Step1.假设文物风化与否与文物的玻璃类型、纹饰、颜色都无关。

Step2.计算出假设成立下的各变量频次。

表格 5.2-1 各变量期望频次

	类别	风化与否		合计
		风化	无风化	
玻璃类型	铅钡	23.4	16.6	40
	高钾	10.6	7.4	18
合计		34	24	58

纹饰	A	12.9	9.1	22
	B	3.5	2.5	6
	C	17.6	12.4	30
合计		34	24	58
颜色	黑	1.2	0.8	2
	蓝绿	6.2	6.2	15
	绿	0.6	0.4	1
	浅蓝	14.1	9.9	24
	浅绿	1.8	1.2	3
	深蓝	1.2	0.8	2
	深绿	4.1	2.9	7
	紫	2.3	1.7	4
合计		34	24	58

表格 5.2-2 各变量实际频次

	类别	**风化与否**		**合计**
		风化	无风化	
玻璃类型	铅钡	28	12	40
	高钾	6	12	18
合计		34	24	58
纹饰	A	11	11	22
	B	6	0	6
	C	17	13	30
合计		34	24	58
颜色	黑	2	0	2
	蓝绿	9	6	15
	绿	0	1	1
	浅蓝	16	8	24
	浅绿	1	2	3
	深蓝	0	2	2
	深绿	4	3	7
	紫	2	2	4
合计		34	24	58

Step3.设f_{s_i}为指标实际观测频次，f_{q_i}为期望观测频次。(i=1、2、3……)

5.2.2 卡方检测模型的求解

将上述指标带入卡方计算公式中：

$$\chi^2 = \sum \frac{(\mathrm{f}_{s_i} - \mathrm{f}_{q_i})^2}{\mathrm{f}_{q_i}} \tag{1}$$

利用 R-studio 软件计算出风化与否与玻璃类型、纹饰、颜色之间的χ^2以及$p-value$。结果如下表。

表格 5.2-3 卡方检验输出数值

	类别	校正χ^2	$p-value$
玻璃类型	铅钡	5.4518	0.01955
	高钾		
纹饰	A	4.9567	0.08389
	B		
	C		
颜色	黑	7.2338	0.405
	蓝绿		
	绿		
	浅蓝		
	浅绿		
	深蓝		
	深绿		
	紫		

P 值或者说观测的显著水平，即在假设为真时的前提下，检验统计量大于或等于实际观测值的概率。

如果 P<0.05，说明是较强的判定结果，拒绝假定的参数取值。

如果 P>0.05，说明较弱的判定结果，拒绝假定的参数取值。

由以上数据可知文物风化有无与玻璃类型相关性较强，与纹饰相关性较弱，与颜色几乎无关。[1]

5.2.3 对化学成分含量的统计规律可视化分析

经过对玻璃文物的表面风化与玻璃类型、纹饰和颜色的关系进行分析，我们可以大致了解到，玻璃的类型和纹饰会在一定程度上与玻璃文物风化有关，而玻璃文物的颜色几乎与玻璃文物的风化没有关系。我们将继续探讨不同类型玻璃以及不同纹饰的文物在风化前和风化后的化学成分会有怎么的变化趋势。

首先是对高钾类玻璃和铅钡类玻璃的分析，首先对数据进行简单处理，发现玻璃中二氧化硅的含量偏大且在风化前后的相对变化率也远大于其他化学成分，所以我们将二氧化硅与其他化合物分开讨论，并将除二氧化硅外其他化合物的相对含量通过折线图表示展示出来，并进行风化前后的趋势比较，具体结果如下：

图 5.2-1 高钾玻璃风化前后成分对比图

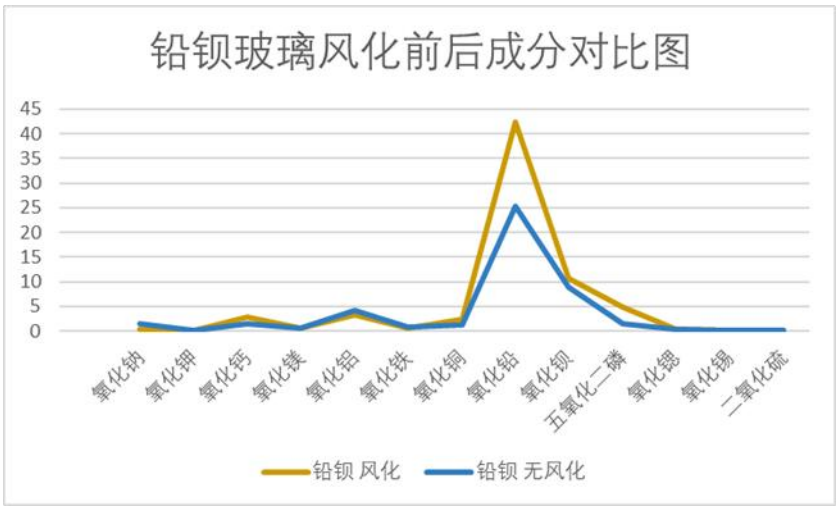
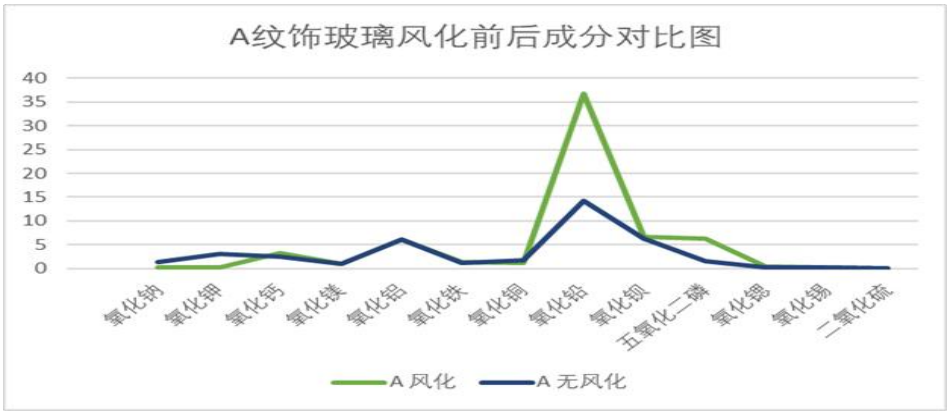


图 5.2-2 铅钡玻璃风化前后成分对比图

然后是根据玻璃纹饰的不同进行玻璃风化前后化学成分变化的分析，具体方法与前者类似，但需要注意的是，经过对数据的简单分类可知，数据中纹饰 B 的玻璃类型均为风化后的玻璃，因此在这里没有讨论的必要性，且二氧化硅对分析的影响性仍然存在因此依然去除二氧化硅。故对 A、C 纹饰的玻璃展示结果如下：



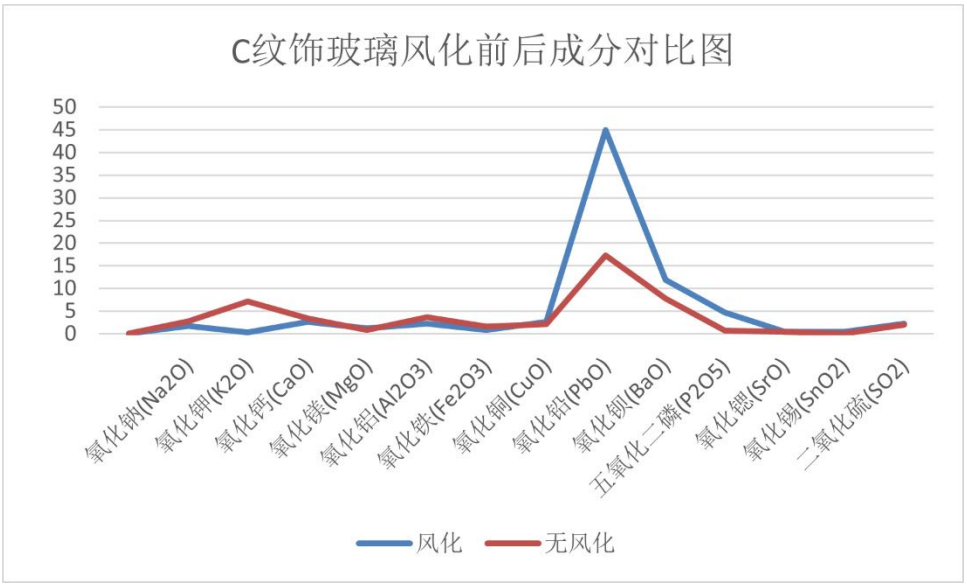


图 5.2-3 A 纹饰玻璃风化前后成分对比图

图 5.2-4 C 纹饰玻璃风化前后成分对比图

由图分析可直观看出在不同条件下风化前后化学成分含量变化的统计规律，我们选取变化较大的前三项对数据本身进行更细致分析，可得出如下表 4，表 5 的统计规律。

下图给出分别在高钾类和铅钡类中，有风化和无风化条件下各成分含量均值。

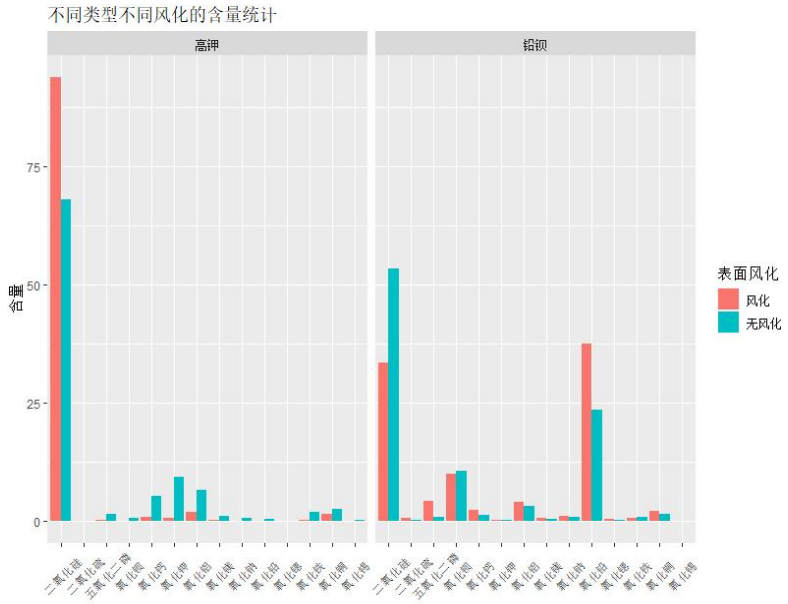


图 5.2-5 不同类型不同风化含量均值统计

由上述图表得到以下统计规律：

当固定变量玻璃类型时，文物表面有无风化主要差异成分以及成分大致含量为以下表格 4 所示。

高钾类			铅钡类		
主要成分	无风化	风化	主要成分	无风化	风化
二氧化硅	67.98417	93.96333	二氧化硅	54.65957	27.05609
氧化钙	5.3325	0.87	五氧化二磷	1.04913	4.76
氧化钾	9.3308333	0.5433333	氧化钙	1.320435	2.777391
氧化铝	6.62	1.93	氧化铅	22.0847826	43.71

表格 5.2-4 主要差异成分

当固定变量纹饰类型时，文物表面有无风化主要差异成分及成分大致含量为以下表格 5 所示。

A 纹饰			C 纹饰		
主要成分	无风化	风化	主要成分	无风化	风化
氧化铅	14.19181	25.24785	氧化铅	17.28333	44.89947
五氧化二磷	1.8075	4.313	氧化钾	7.13	0.283333
氧化钾	5.451666	0.337777	五氧化二磷	0.689	4.631176

5.2.4 预测风化前化学成分含量模型的建立

Step1 将某成分已知风化前含量记为a_i，风化后的含量记为a_j，给定的 1 风

化后一点的此成分检测数据记为x_j, 假设风化前该点此成分含量为x_i(其中 i,j=1、2、……、14，a_i, a_j，x_j为已知量，x_i为需预测的未知量)。

Step2 利用此比例公式，预测风化前成分的含量：

$$\frac{a_i}{a_j}=\frac{x_i}{x_j} \tag{1}$$

模型建立完毕。

5.2.5 预测风化前化学成分含量模型的求解

根据上述模型预测出不同玻璃类型文物表面未风化前各成分含量。由于数据量较多正文中只给出一部分数据，完整数据见支撑材料。

文物编号	类型	表面风化	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	氧化锡	二氧化硫
7	高钾	预测未风化前	67.01948	0.695	0	6.558362	0	6.791503	1.239182	5.088218	0.411667	0.598333	3.055446	0	0	0
9	高钾	预测未风化前	68.74868	0.695	10.13225	3.800172	0	4.527668	2.332579	2.434178	0.411667	0.598333	1.753125	0	0	0
10	高钾	预测未风化前	70.01484	0.695	15.79945	1.287155	0	2.778342	1.89522	1.319168	0.411667	0.598333	0	0	0	0
12	高钾	预测未风化前	68.22052	0.695	17.34505	4.413103	0	5.007876	2.113899	2.591222	0.411667	0.598333	0.751339	0	0	0
22	高钾	预测未风化前	66.81689	0.695	12.70825	10.17466	3.511864	12.00518	2.551258	0.863741	0.411667	0.598333	1.051875	0	0	0
27	高钾	预测未风化前	67.08459	0.695	0	5.761552	2.963136	8.60943	1.457862	2.418474	0.411667	0.598333	1.803214	0	0	0

图 5.2-6 高钾类文物未风化前成分含量预测结果

文物编号	类型	表面风化	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	氧化锡	二氧化硫
2	铅钡	预测未风化前	63.05508	0	1.778565	1.572505	1.103041	7.081874	2.903964	0.13382	29.57048	0	1.132365	0.155513	0	0
8	铅钡	预测未风化前	35.00357	0	0	0.994576	0	1.656145	0	5.357965	17.8807	26.76936	1.138709	0.30284	0	2.268995
11	铅钡	预测未风化前	58.37983	0	0.355713	2.358757	0.663694	3.324649	0	2.537442	15.82953	12.52323	2.975233	0.30284	0	0
19	铅钡	预测未风化前	51.51468	0	0	1.968991	0.55152	4.412267	2.07649	1.806576	26.69635	4.58585	2.800779	0.155513	0	0
26	铅钡	预测未风化前	34.39526	0	0	0.967695	0	0.86515	0	5.440316	18.41063	27.64367	0.992802	0.368319	0	1.723732
34	铅钡	预测未风化前	62.18608	0	0.423468	0.524168	0	2.002205	0.733797	0.777188	29.02184	8.571682	0.107844	0.180067	0	0
36	铅钡	预测未风化前	68.77314	3.445122	0.237142	0.248644	0	1.977487	0.499607	0.349992	25.94197	9.283132	0.022203	0.180067	0	0
38	铅钡	预测未风化前	57.23274	2.141562	0	0.456967	0	3.176338	0.452769	0.375727	30.74258	8.391677	0.152251	0.33558	0	0
39	铅钡	预测未风化前	45.62282	0	0	0.745932	0	0.617965	0	0.452931	38.04947	6.188755	0.367939	0.499277	0	0

图 5.2-7 铅钡类文物未风化前成分含量预测结果

5.3 问题二模型建立与求解

5.3.1 K-means 算法对应模型的建立

Step1.K 值的选取

由于文题已给出需要划分成两类：高钾类与铅钡类，因此取 K=2

Step2.距离度量

选取欧氏距离作为度量。以有效数据中的各文物作为向量α_i，文物中的各成分含量作为向量的分量x_{im}, $\alpha_i=(x_{i1},x_{i2},\ldots,x_{im},\ldots,x_{in})$

则欧式距离为：

$$\begin{aligned} d(\alpha_i,\alpha_j) &:= \sqrt{(x_{i1}-x_{j1})^2+(x_{i2}-x_{j2})^2+\cdots+(x_{in}-x_{in})^2} \\ &= \sqrt{\sum^{n}(x_{im}-x_{im})^2} \end{aligned} \tag{1}$$

Step3.经过第二步两个簇中已有新样本，于是重新计算质心，每个簇坐标的均值为新质心。

Step4.重复执行 step2,step3 直到质心不再改变。

5.3.2 K-means 算法对应模型求解

根据 K-means 算法基本过程, 运用 R-studio 软件编程得出将文物聚类为铅钡类和

高钾类的主要指标。输出结果如下表所示

表格 5.3-1 划分高钾类铅钡类主要指标

种类 / 成分	铅钡类	高钾类	相差的绝对值
氧化铅	22.178	1.468	20.710
二氧化硅	54.875	67.364	12.489
氧化钡	9.474	0.600	8.874
氧化钾	0.191	8.724	8.533
氧化钙	1.044	5.484	4.440
氧化铝	3.568	7.127	3.559
氧化铁	0.647	1.700	1.053
氧化钠	1.736	0.842	0.894
氧化铜	1.588	2.263	0.675
氧化镁	0.625	1.094	0.469
氧化锶	0.262	0.053	0.209
氧化锡	0.033	0.215	0.182
二氧化硫	0.183	0.111	0.072
五氧化二磷	1.151	1.111	0.040

由上表可知高钾类、铅钡类玻璃主要依靠氧化铅、二氧化硅、氧化钡、氧化钾、氧化钙、氧化铝含量的差异进行分类。其中氧化铅对分类影响的权重最高，二氧化硅其次，以此类推。

5.3.3 层次聚类模型建立

在层次聚类算法中，单个数据对象为初始簇，将最相近的簇聚合，直到得到所希望得到的聚类数为止。[2]

经过数据预处理后，有 n 个对象要被聚类，距离矩阵大小为 n*n，使用最小距离的方法进行聚类的过程如下。

Step1 将每个样本数据视为一簇，计算出每簇之间的距离$d(i,j)$,得到距离矩阵 D_0

Step2 将距离最近的两簇合并成一个新的簇。

Step3 再重新计算新簇与其它簇之间的距离，计算出新的距离矩阵 D_1

Step4 重复第 step2 step3，直到得到期望得到的簇的个数。

5.3.4 层次聚类模型求解

对于高钾类玻璃，因为都具有较多氧化钾，氧化钙，二氧化硅，因此应在分类时将这些变量去除，避免影响分类，同时观察严重风化的二氧化硫含量极高，因此其应不属于玻璃的主要成分，也要去除对于氧化锡和氧化钠来说，缺失值太多，不具有代表性，也应去除。

对于铅钡类玻璃来说，因为都具有较多氧化钡，氧化铅，二氧化硅，氧化钙故应该去除氧化铅，氧化钡，二氧化硅，氧化钙,二氧化硫的影响来对其进行分析

根据层次聚类算法基本过程利用 R-studio 软件进行编程，分别得到推荐的亚类个数、高钾亚类分类、铅钡亚类分类

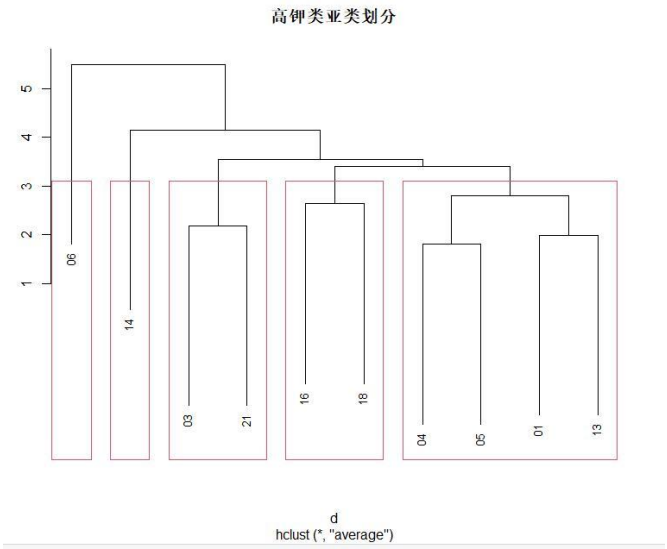
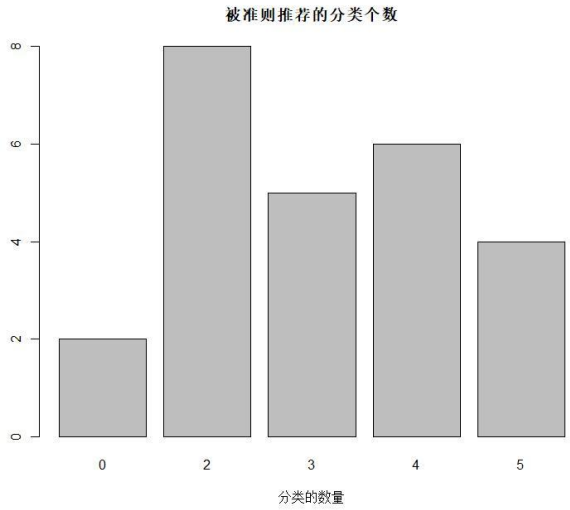


图 5.3-1 高钾亚类划分

类型 / 成分	1	2	3	4	5
氧化镁	1.0500	0.8875	1.8550	0.6600	1.0250
氧化铝	6.0075	5.4850	10.6000	9.2300	4.6150
氧化铁	2.325	1.725	4.215	0.500	0.210
氧化铜	3.5125	3.1075	2.3450	0.4700	0.5350
氧化铅	0.000	0.915	0.275	1.620	0.055
氧化钡	0.000	1.700	1.175	0.000	0.000
五氧化二磷	1.0425	0.8900	4.3400	0.1600	0.6800
氧化锶	0.015	0.025	0.115	0.000	0.055



表格 5.3-2 高钾类亚类

图 5.3-2 被推荐分类个数

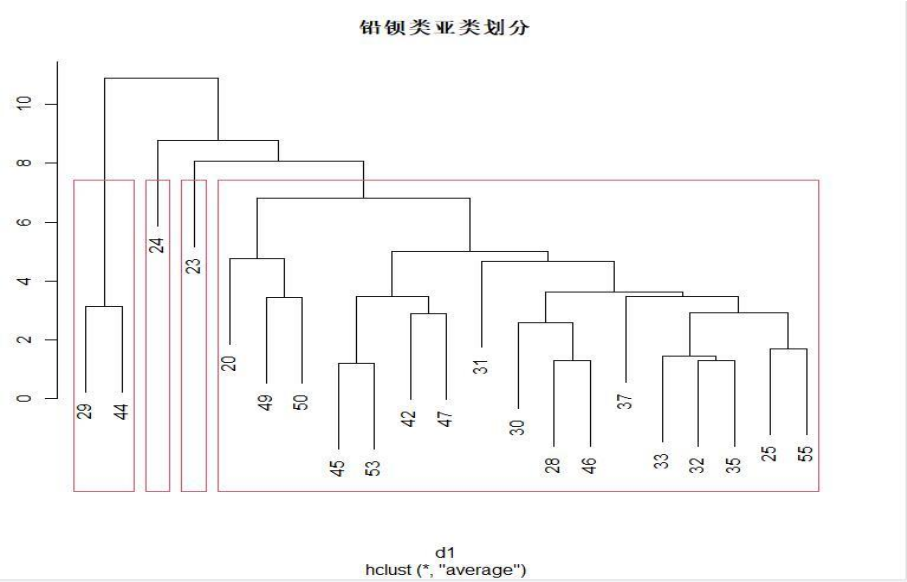


图 5.3-3 铅钡类亚类划分

类型 / 成分	1	2	3	4
氧化钠	1.240588	7.92	0	1.99
氧化钾	0.2102941	0	0	0.25
氧化镁	0.6270588	0.71	0	0.745
氧化铝	3.750	1.420	1.59	13.515
氧化铁	0.7605882	0	0	0.79
氧化铜	1.036176	2.99	8.46	0.585
五氧化二磷	1.343235	0	0.14	0.205
氧化锶	0.2252941	0	0.91	0.255
氧化锡	0.03823529	0	0	0

表格 5.3-3 铅钡类亚类

高钾类选取氧化镁、氧化铝、氧化铁、氧化铜、氧化铅、氧化钡、五氧化二磷、氧化锶对其进行亚类划分，共划分为 5 个亚类

铅钡类选取氧化钠、氧化钾、氧化镁、氧化铝、氧化铁、氧化铜、五氧化二磷、氧化锶、氧化锡对其进行亚类划分，共划分为 4 个亚类。

5.3.5 亚类划分合理性及敏感性分析

高钾类亚类划分合理性分析:

首先第 3 类中氧化铁和五氧化二磷含量与其它类别相差较大，因此将其作为一个亚类。再将剩下四个组别做对比。

第 4 类氧化铅含量与其他组别差异性较大, 因此将其作为一个亚类再将剩下三个组别做对比。

剩下 1、2、5 三组，仅有第 2 类中含有氧化铅及氧化钡，因此将第二类又划分为一个亚类。

最后对比 1、5 两类，可见两类中氧化铁、氧化铜含量差异很大，有近十倍的相差值，因此又将第 1 类，第 5 类分别划分为成两个亚类。

铅钡类亚类划分合理性分析：

首先第 2 类氧化钠含量与其它类别有很大差异, 显然应将第 2 类划分为一个亚类。再将剩下 3 类做对比。

可见第 3 类氧化铜含量与其它类别相差很大, 因此将第 3 类划分为一个亚类。

最后对比 1、4 两类，此两类氧化铝含量差异较大，因此可将此两类划分成两个亚类。

高钾类划分敏感性分析：

我们期望的是当改变某个亚类中较为重要的变量, 数据在小范围波动时分类不会发生变化，而含量大幅度变化时分类会发生变化。以高钾类亚类中第 4 类为代表进行敏感性分析。先小范围内改动氧化铅含量。如下图所示，氧化铅含量轻微改变没有对分类造成影响。

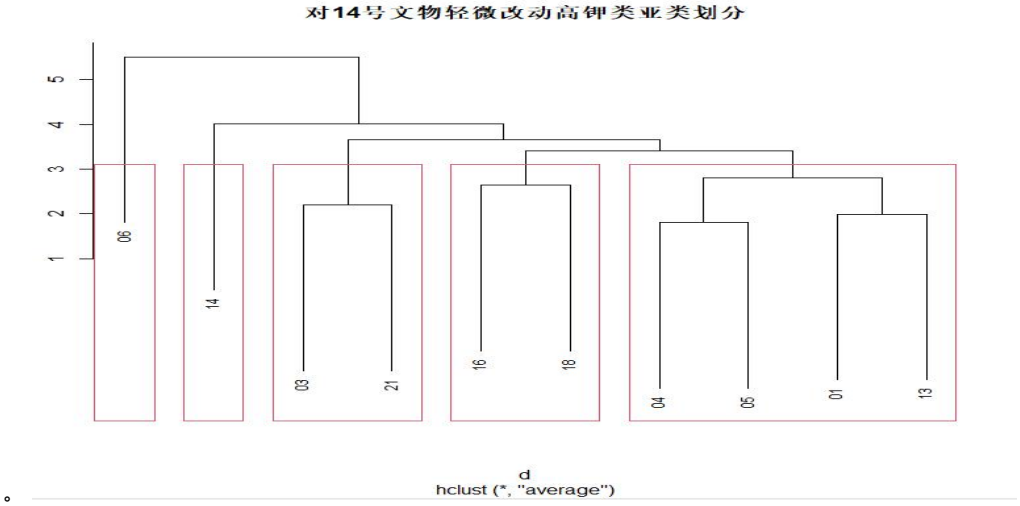


图 5.3-4 轻微改动后高钾类亚类划分

接着对氧化铅含量做出较大改动，亚类划分产生较大变化

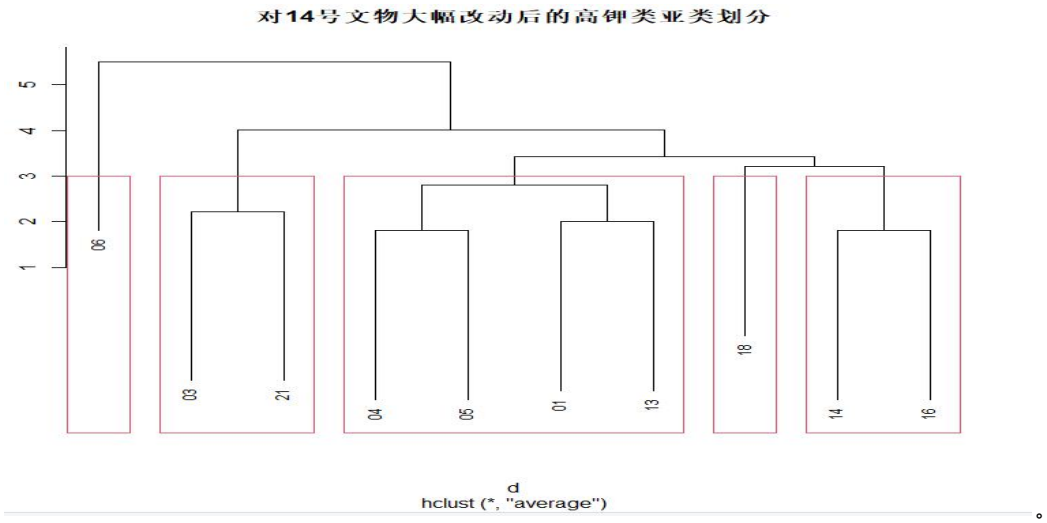


图 5.3-5 大幅改动后高钾亚类划分

铅钡类亚类划分敏感性分析：

选取第 3 类为代表，先对氧化铜含量进行小幅度改动，亚类分类没有改变。

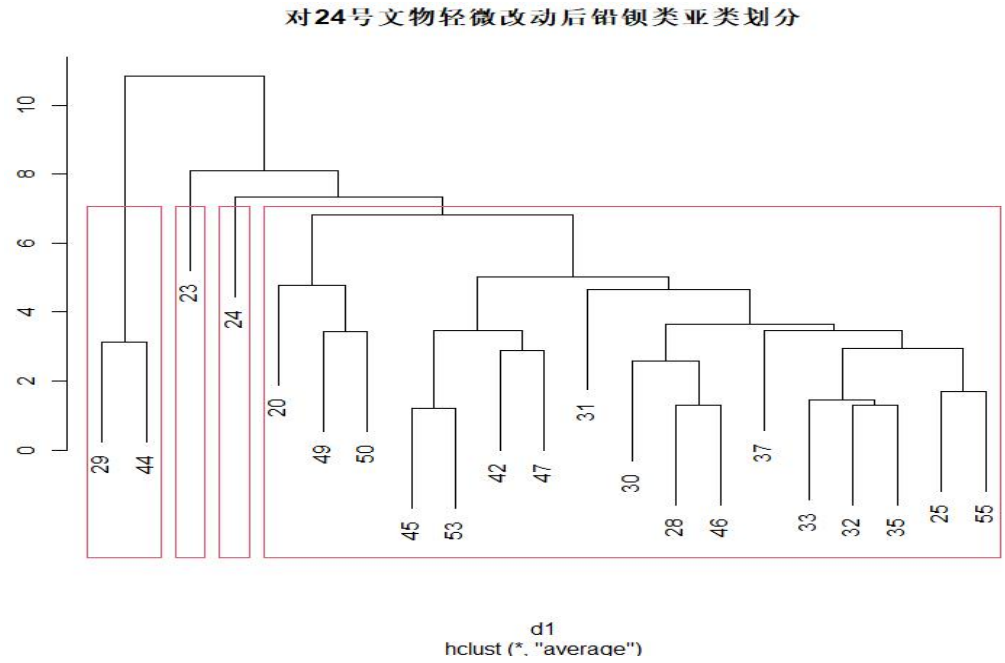


图 5.3-6 轻微改动后铅钡亚类划分

对氧化铜进行较大幅度改动分类发生较大变化。如图。

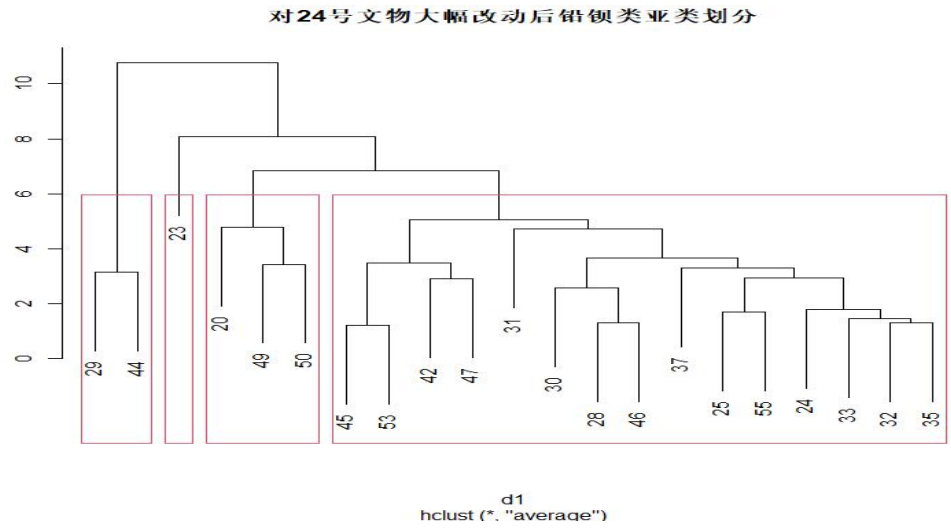


图 5.3-7 大幅改动后铅钡亚类划分

由敏感性分析可看出，模型给出的分类符合自然亚类分类规律，具有可应用性。

5.4 问题三模型建立与求解

5.4.1 主成分分析模型建立

由于表单三未知类别玻璃文物化学成分较多，先对各化学成分进行相关性分析。如下图。

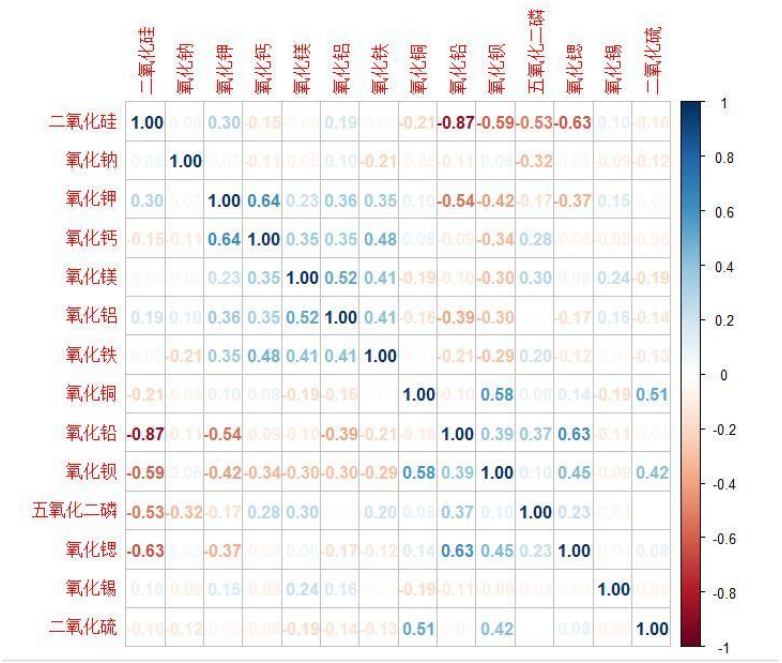


图 5.4-1 各成分之间相关系数

可见各成分间相关性较强，因此可采用主成分分析法对 14 种成分含量x_i进

行降维。

具体过程如下。

Step1 对初始数据进行标准化处理。

Step2 计算相关系数矩阵 A。

Step3 计算特征值与特征向量。

$$\begin{aligned} y_1 &= u_{11}x_1 + u_{21}x_2 + \ldots + u_{14,1}x_{14} \\ y_2 &= u_{12}x_1 + u_{22}x_2 + \ldots + u_{14,2}x_{14} \\ &\vdots \\ y_{14} &= u_{1,14}x_1 + u_{2,14}x_2 + \ldots + u_{14,14}x_{14} \end{aligned} \tag{1}$$

Step4 计算特征值λ_j的信息贡献率和累计贡献率。

$$a_j = \frac{\lambda_j}{\sum_{k=1}^{m} \lambda_k}, j = 1,2,\ldots,m \tag{2}$$

Step5 选择 q 个主成分，求综合得分。

$$I_t = \sum_{j=1}^{q} a_j y_j \tag{3}$$

Step6 对综合得分归一化处理。

$$I = \frac{I_t - min\,(I_t)}{max\,(I_t) - min\,(I_t)} \tag{4}$$

5.4.2 判断主成分个数

通过检查变量之间 n*n 的相关系数来判断保留的主成分数。

最常见的是基于特征值的方法。每个主成分都与相关系数矩阵的特征值相关联，第一主成分与最大的特征值相关联，第二主成分与第二大的特征值相关联，依此类推。Kaiser-Harris 准则建议保留特征值大于 1 的主成分

少。Cattell 碎石检验则绘制了特征值与主成分数的图形。这类图形可以清晰地展示图形弯曲状况，在图形变化最大处之上的主成分都可保留。最后，你还可以进行模拟，依据与初始矩阵相同大小的随机数据矩阵来判断要提取的特征值。若基于真实数据的某个特征值大于一组随机数据矩阵相应的平均特征值,那么该主成分可以保留。该方法称作平行分析。[3]

代码生成的图形展示了基于观测特征值的碎石检验。

红线上方为推荐选取的主成分，因此我们选取 3 个主成分。

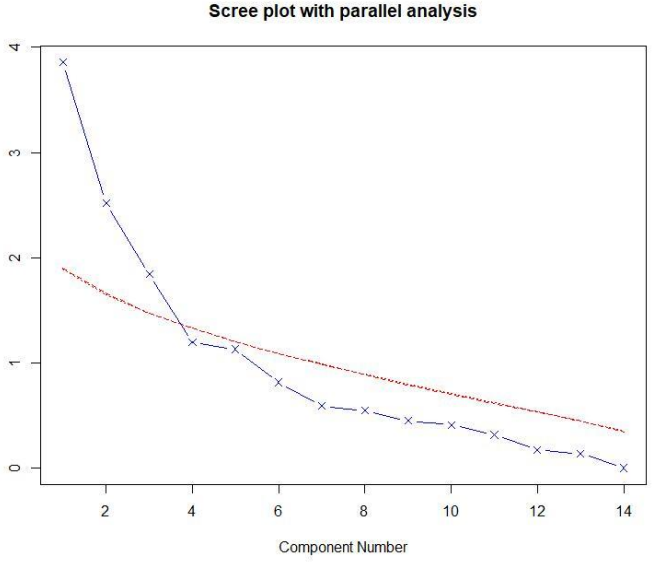


图 5.4-2 平行分析碎石图

5.4.3 主成分分析模型求解

表格 8 所示是主成分未旋转时的变量与主成分之间的相关系数。

表格 5.4-1 未旋转时变量与主成分之间的相关系数

	PC1	PC2	PC3	H1	U2	Com
二氧化硅	0.72	−0.60	0.01	0.884	0.12	1.9
氧化钠	0.03	−0.31	−0.11	0.107	0.89	1.3
氧化钾	0.70	0.16	0.41	0.685	0.32	1.7
氧化钙	0.42	0.65	0.31	0.688	0.31	2.2
氧化镁	0.40	0.62	−0.16	0.571	0.43	1.9
氧化铝	0.60	0.33	0.02	0.470	0.53	1.5
氧化铁	0.46	0.56	0.17	0.561	0.44	2.1
氧化铜	−0.32	0.01	0.84	0.816	0.18	1.3
氧化铅	−0.77	0.38	−0.36	0.872	0.13	1.9
氧化钡	−0.79	−0.04	0.36	0.748	0.25	1.4
五氧化二磷	−0.25	0.71	−0.02	0.573	0.43	1.3
氧化锶	−0.64	0.37	−0.12	0.568	0.43	1.7
氧化锡	0.21	0.03	−0.20	0.086	0.91	2.1
二氧化硫	−0.31	−0.12	0.69	0.588	0.41	1.4

但第一主成分似乎与很多的化学成分都有较高的相关性，为了使结果更具有解释性，下面对主成分进行旋转，在不改变累加方差解释度的前提下，使得结果更具有解释性。见表格 9。

表格 5.4-2 旋转后变量和主成分之间的相关系数

	RC1	RC2	RC3	H2	U2	Com
二氧化硅	−0.91	−0.10	−0.23	0.884	0.12	1.2
氧化钠	−0.18	−0.26	−0.08	0.107	0.89	2.0
氧化钾	−0.54	0.62	0.07	0.685	0.32	2.0
氧化钙	0.00	0.83	0.04	0.688	0.31	1.0
氧化镁	0.12	0.64	−0.38	0.571	0.43	1.7
氧化铝	−0.26	0.58	−0.26	0.470	0.53	1.8
氧化铁	−0.05	0.74	−0.09	0.561	0.44	1.0
氧化铜	0.01	0.11	0.90	0.816	0.18	1.0
氧化铅	0.91	−0.22	−0.05	0.872	0.13	1.1
氧化钡	0.46	−0.32	0.66	0.748	0.25	2.3
五氧化二磷	0.63	0.42	0.02	0.573	0.43	1.8
氧化锶	0.74	−0.08	0.12	0.568	0.43	1.1
氧化锡	−0.08	0.07	−0.27	0.086	0.91	1.3
二氧化硫	−0.04	−0.03	0.77	0.588	0.41	1.0

	PC1	PC2	PC3
SS loadings	3.86	2.52	1.84
Proportion Var	0.28	0.18	0.13
Cumulative Var	0.28	0.46	0.59
Proportion Explained	0.47	0.31	0.22
Cumulative Proportion	0.47	0.78	1.00

从得到的结果可以看到

第一个主成分主要由二氧化硅，氧化铅，五氧化二磷，氧化锶来解释。

第二个主成分主要由氧化钾，氧化钙，氧化镁，氧化铝，氧化铁来解释。

第三主成分主要由氧化铜，氧化钡，二氧化硫来解释。

三个主成分对应的特征值分别为 3.21, 2.84, 2.18, 累加方差解释度有 100%。

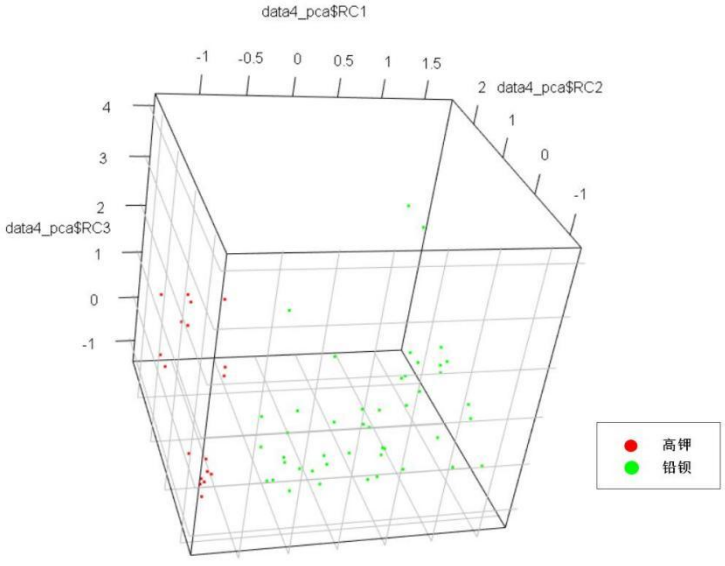
以下是各个化学成分的主成分得分

表格 5.4-3 各化学成分主成分得分

	RC1	*RC2*	*RC3*
二氧化硅	-0.29	-0.09	-0.05
氧化钠	-0.06	-0.10	-0.03
氧化钾	-0.16	0.22	0.12
氧化钙	0.03	0.31	0.08
氧化镁	0.10	0.23	-0.14
氧化铝	-0.04	0.20	-0.05
氧化铁	0.02	0.27	0.01
氧化铜	-0.06	0.10	0.45
氧化铅	0.29	-0.05	-0.11
氧化钡	0.09	-0.05	0.27
五氧化二磷	0.23	0.18	-0.01
氧化锶	0.23	0.01	0.00
氧化锡	0.00	0.00	-0.13
二氧化硫	-0.07	0.04	0.38

5.4.4 支持向量机模型的建立

在使用逻辑回归时发现样本点之间极有可能是线性可分的。通过三维图像可以看到，样本点之间基本是线性可分的，因此使用线性可分的支持向量机对已知



数据进行训练。

图 5.4-3 高钾类、铅钡类文物特征分布

由于训练样本集是线性的，因此假定函数为

$$y = f(x) = (\alpha \cdot x) + b \tag{1}$$

其中$(\alpha \cdot x)$为向量$\alpha \in R^n$，$x \in R^n$的内积，x为所选用样本集数据，其分量为 14 种成分含量。此时问题转化为优化问题

$$\min_{\omega,b} \frac{1}{2}\|\alpha\|^2 \tag{2}$$

$$\text{s.t. } |y_i - ((\alpha \cdot x) + b)| \leq \varepsilon \tag{3}$$

由于 VC 维满足$h \leq \|\omega\|^2 r^2 + 1$，优化$\frac{1}{2}\|\omega\|^2$则转变为求最小 VC 值。

若得到不止一条函数，则以高钾类、铅钡类文物中距离最近的样本之为约束条件，可更好穿过两样本之间的函数则为目标函数。

5.4.5 支持向量机模型的求解

将已知数据按 7：3 的比例划分为训练集和测试集，使用 Rstudio 软件对训练集进行训练，并在预测集上进行预测，并通过分析混淆矩阵，可以看到模型的准确率为 97%。

将训练得到的模型用来预测表单三中未知类别的文物的类别，先对表单三中的化学成分按照主成分得分进行降维，并将得到的三个主成分以及表面有无风化作为输入预测文物类别，得到结果如下表:

表格 5.4-4 未知文物类别

	A1	A2	A3	A4	A5	A6	A7	A8
类型	铅钡	铅钡	铅钡	铅钡	铅钡	高钾	高钾	铅钡

5.4.6 分类结果敏感性分析

为了验证分类模型的敏感性，我们对数据进行扰动处理，将各个化学成分的含量范围分别扩大 5%、10%、20%、30%，对扰乱后的结果与未进行处理的结果进行对比。可视化如下。

表格 5.4-5 扰乱后结果

编号 / 范围	A1	A2	A3	A4	A5	A6	A7	A8

5%	无变化	无变化	无变化	无变化	无变化	无变化	无变化	无变化
10%	无变化	无变化	无变化	无变化	无变化	无变化	无变化	无变化
20%	无变化	无变化	无变化	无变化	无变化	无变化	无变化	无变化
30%	无变化	无变化	无变化	无变化	无变化	无变化	无变化	无变化

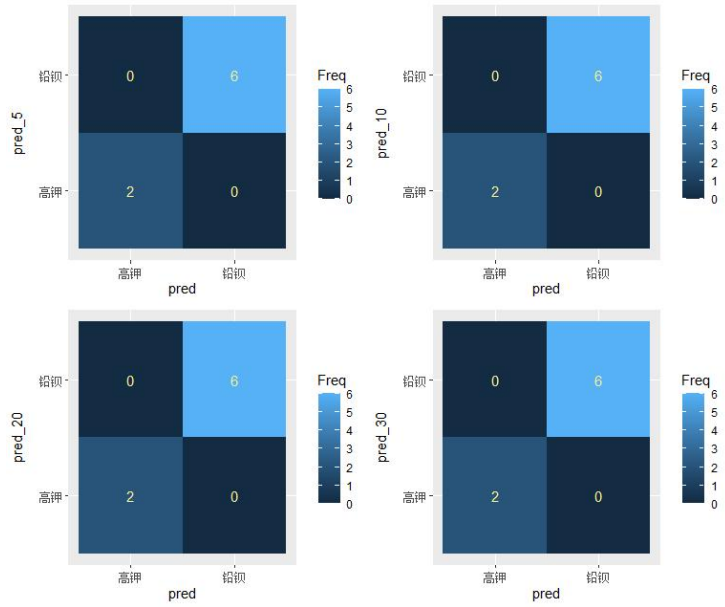


图 5.4-4 数据扰乱后结果

由上图及上表可知该模型稳定性较强。

5.5 问题四模型的建立与求解

5.5.1 R 型 K-means 模型的建立

由于问题四是分析各成分指标之间的关联关系，因此采用 R 型 K-means 聚类分析。以成分作为样本进行聚类。以相关系数举证进行相似性度量。

记各成分x_j的取值为$[x_{1j}, x_2 j, \cdots, x_{mj}]^t \in \mathbf{R}^*(j = 1,2,\cdots,p)$。求解变量$x_j$和$x_k$的相关系数。

$$r_{jk} = \frac{\sum_{i=1}^{m}(x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)}{[\sum_{i=1}^{m}(x_{ij} - \bar{x}_j)^2 \sum_{i=1}^{m}(x_{ik} - \bar{x}_k)^2]^{\frac{1}{2}}}, j,k = 1,2,\cdots,p, \quad (1)$$

其中$\bar{x}_j = \frac{1}{m}\sum_{i=1}^{m} x_{ij}$。[5]

$|r_{jk}|$越接近 1，则两变量相似性越高。从而输出以下结果。

5.5.2 R 型 K-means 模型的求解

对于 K-means 算法中的一个簇来说，所有样本点到质心的距离之和越小，就

认为这个簇中的样本越相似，簇内差异就越小。而距离的衡量方法有多种，令 x 表示簇中的一个样本点，μ 表示该簇中的质心，n 表示每个样本点中的特征数目，i 表示组成点 x 的每个特征，则该样本点到质心的距离可以由以下距离来度量：

$$\text{欧几里得距离：}\ d(x,\mu)=\sqrt{\sum_{i=1}^{n}(x_i-\mu_i)^2} \tag{1}$$

$$\text{曼哈顿距离：}\ d(x,\mu)=\sum_{i=1}^{n}(|x_i-\mu|) \tag{2}$$

如采用欧几里得距离，则一个簇中所有样本点到质心的距离的平方和为：

$$\text{Cluster Sum of Square}\ (CSS)=\sum_{j=0}\sum_{i=1}(x_i-\mu_i)^2 \tag{3}$$

$$\text{Cluster Sum of Square}\ (CSS)=\sum_{j=0}\sum_{i=1}(x_i-\mu_i)^2 \tag{4}$$

对于四类玻璃类型，从 1 开始对 k 的值进行累加赋值，并得到相应总平方和，将平方和与 k 值关系绘制为折线图如下：

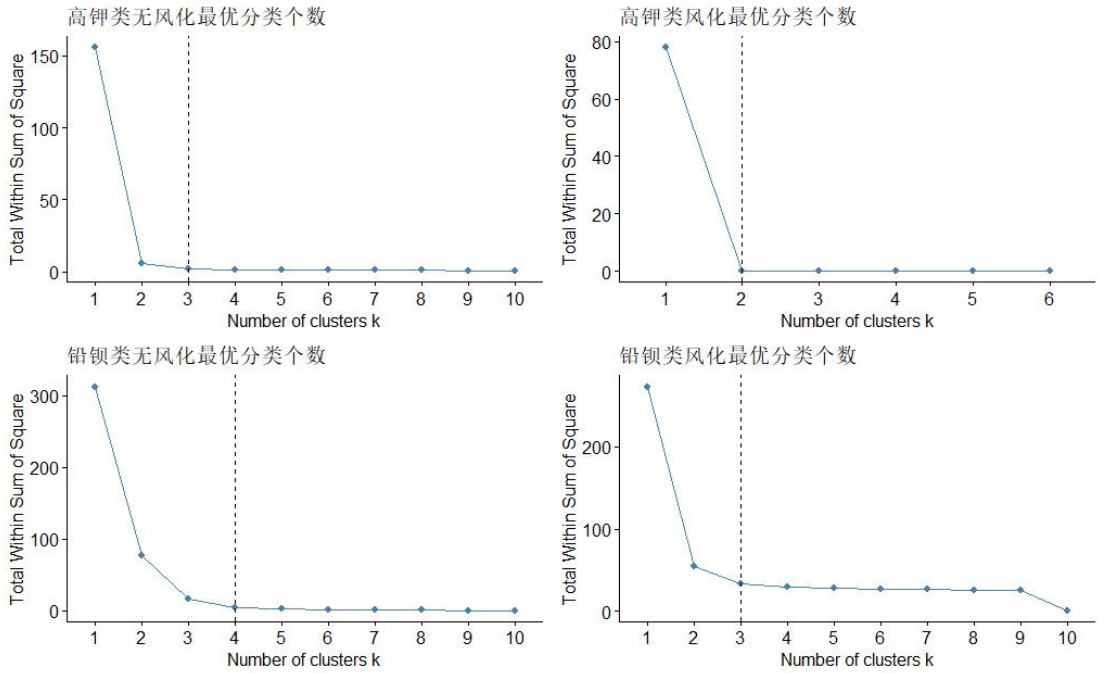


图 5.5-1 最优分类个数

由此可以初步确定不同玻璃类型较优分类的个数，并进行 K-means 聚类分析。

经过对变量之间关系的聚类分析，结合 R 语言的可视化图表，可以直观看到聚类分析中簇内和簇间的关联程度。颜色相同表示属于同一亚类的化合物。以高钾类无风化图为例，二氧化硅单独被分为一类，而氧化钾、氧化钙、氧化铝三个氧化物被分为一类（以蓝色表示），且簇内的关联程度低于“氧化纳”绿色组（包含氧化钠，五氧化二磷等化合物）。同时也可以看到“氧化钾”蓝色一组和

“氧化钠”绿色一组的关联程度远大于蓝色组和绿色组与“二氧化硅”的关联程度。

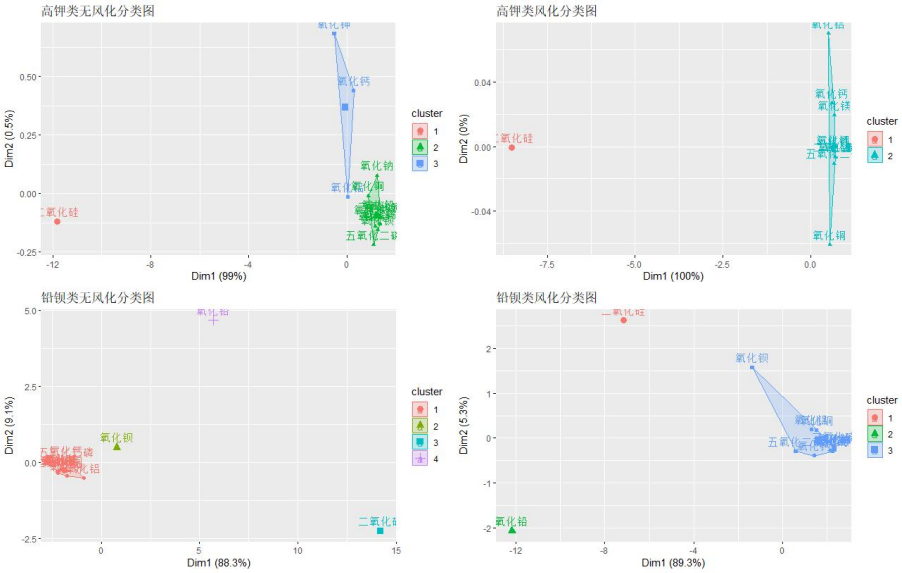


图 5.5-2 各成分关系图

使用 R-studio 编程求得风化前后的聚类指标变化，如下表所示：

表格 5.5-1 聚类指标变化

玻璃类别	推荐分类数目	CH	DBI	轮廓系数
高钾风化	2	14138.7	0.006293936	0.9205831
高钾无风化	3	312.6441	0.04438701	0.8696975
铅钡风化	3	39.54682	0.3859939	0.6380826
铅钡无风化	4	96.40703	0.08771918	0.7221896

其中：CH 指标通过计算类中各点与类中心的距离平方和，来度量类内的紧密度，通过计算各类中心点与整个数据集中心点距离平方和来度量数据集的分离度，CH 指标由分离度与紧密度的比值得到。从而，CH 越大代表着类自身越紧密，类与类之间越分散。

DBI 表示任意两类别的类内样本到类中心平均距离之和除以两类中心点之间的距离，取最大值。DBI 越小意味着类内距离越小，同时类间距离越大。

轮廓系数则结合内聚度和分离度两种因素，在相同原始数据的基础上用来评价不同算法、或者算法不同运行方式对聚类结果所产生的影响。其取值范围为[-1,1]，值越大，聚类效果越好。

通过数据对比可知，高钾类风化后 CH 指数骤增，说明高钾类风化后化学成分关联度减小。铅钡类风化后的 CH 值减小，说明铅钡类风化后化学成分关联度增强。

高价类风化前的化学成分的关联度要小于铅钡类风化前化学成分的关联度。

六、模型的评价与改进

6.1 模型优点

1、数据处理时考虑到空缺颜色、空缺数据、严重风化现实意义、微量元素等对模型结果的影响，考虑较全面。

2、使用支持向量机机器学习算法，使对文物类型划分精确度更高。

3、针对数据具体情况采取精度更高的模型，例如问题 2 所给数据量较小，相较 K-means 聚类算法，层次聚类算法输出结果精度更高。

6.2 模型缺点

1、有些化学成分含量较少，不满足正态分布，但依旧使用统计规律进行预测，会导致结果不准确。

2、在考虑亚类划分时仅仅考虑了高钾与铅钡这两个大类，没有将有无风化，纹饰，以及颜色纳入到划分中。

3、数据较少，使用支持向量机等机器学习模型时会出现过拟合现象，模型的泛化能力不足。使用 K-means 聚类分析可能不够合理。

6.3 模型改进方向

1、可作基于最小生成树改进型层次聚类，将要聚类的数据先计算其两两之间的距离再对距离进行排序。按顺序将不同簇融合，则可降低复杂度。基于最小生成树改进层次聚类实现上述想法。[4] ···

2、可查阅更多资料对数据进行扩充，将更多数据放入训练集以提高模型精度。

七、参考文献

[1] 赵桂梅. 统计假设检验 p-值修正方法研究[D]. 北京理工大学, 2014.
[2] 段明秀. 层次聚类算法的研究及应用[D].中南大学,2009.
[3] [美] Robert I.Kabacoff 著，王小宁刘撷芯黄俊文等译.R 语言实战[M].人民邮电出版社.p299
[4] 段明秀. 层次聚类算法的研究及应用[D].中南大学,2009.
[5] 司守奎.孙玺菁著.数学建模算法与应用（第三版）[M].国防工业出版社.p269.

八、附录

8.1 支撑材料列表

铅钡类风化点未风化前成分含量预测
高钾类风化点未风化前成分含量预测
国赛问题一代码
国赛问题二代码
国赛问题三代码
国赛问题四代码

8.2 问题一代码：

```
library(readxl)
library(tidyverse)
library(rpart)
library(DataExplorer)
library(ggplot2)
library(rpart.plot)

data1<-read_excel(file.choose(),sheet = 1)
data2<-read_excel(file.choose(),sheet = 2)
data3<-read_excel(file.choose(),sheet = 3)

data1<-as.data.frame(data1,na.rm = TRUE)
data2<-as.data.frame(data2,na.rm = TRUE)
data3<-as.data.frame(data3,na.rm = TRUE)
```

```
#观察缺失值
missing_value<- 1-nrow(na.omit(data1))/nrow(data1)
plot_missing(data1)

#变量类型的转换
for (i in c(2,3,4,5)){
  data1[,i]<-as.factor(data1[,i])
}

#使用分类回归树来对缺失数据进行插补
data1_temp<-data1%>%filter(类型 == "铅钡")
class_mode<-rpart(颜色~纹饰+表面风化+类型,data = data1_temp,method =
"class")
prp(class_mode,
      type = 2,
      tweak = 1.2,
      fallen.leaves = T,
```

```
    main = "Decision Tree")

color_pred<-predict(class_mode,data = data1,type = "class")
data1[19,4]<-factor("浅蓝")
data1[40,4]<-factor("浅蓝")
data1[48,4]<-factor("浅蓝")
data1[58,4]<-factor("浅蓝")
```

#生成三对定性变量的频数统计

```
stat1<-table(data1$表面风化,data1$类型)#表面风化与类型
stat2<-table(data1$表面风化,data1$纹饰)#表面风化与纹饰
stat3<-table(data1$表面风化,data1$颜色)#表面风化与颜色

#对以上三组定性变量进行卡方检验
print(chisq.test(stat1))
print(chisq.test(stat1),correct = FALSE)#非矫正
print(chisq.test(stat2))
print(chisq.test(stat2),correct = FALSE)#非矫正
print(chisq.test(stat3))
print(chisq.test(stat3),correct = FALSE)#非矫正

#通过上述结果可知，第一组 p 值为 0.0195，原假设不成立，通过假设性检验，
说明两者相关
#第二组 p 值为 0.08 大于 0.05，可以说明两者无关
#第三组 p 值为 0.432，不通过假设检验，故两者无关
```

#对表单二数据进行预处理

```
#将所有的 NA 值都转化为 0
for(i in 1:69){
   for(j in 2:15){
      if(is.na(data2[i,j]))
         data2[i,j]<-0
   }
}

#对无效数据进行删除
data2<-data2%>%mutate(rowsum = rowSums(.[2:15]))
for (i in 1:69){
   if(data2$rowsum[i] < 85)
      data2 <-data2[-i,]
}
```

```
#根据题目可知，表面风化严重的地方已于环境进行了大量的物质交换，会影响分类的准确性，不具有参考价值故删去
data2<-data2[-11,]
data2<-data2[-27,]
data2<-data2[-61,]

#先对表单二的首列进行拆分成文物编号与采样点，方便后续连接
data2_update<-data2%>%separate(文物采样点,c("文物编号","文物采样点"),sep=2)

#其次使用 semi_join 函数对表单二中删除的文物同样在表单一中删除，方便连接
data1%>%semi_join(data2_update,by = "文物编号")

#最后以文物编号为外键对两个表单进行连接
data4<-data1%>%semi_join(data2_update,by = "文物编号")%>%left_join(data2_update,by = "文物编号")
```

#首先对数据的分布进行一个大致的了解

```
#高钾类无风化
p1<-ggplot(data = data4_temp_G,mapping = aes(二氧化硅,..density..))+geom_histogram(color = "white",fill = "gray60",binwidth = 3)+geom_line(stat = "density",size = 1.5)
p2<-ggplot(data = data4_temp_G,mapping = aes(氧化钾,..density..))+geom_histogram(color = "white",fill = "gray60",binwidth = 3)+geom_line(stat = "density",size = 1.5)
p3<-ggplot(data = data4_temp_G,mapping = aes(氧化钙,..density..))+geom_histogram(color = "white",fill = "gray60",binwidth = 3)+geom_line(stat = "density",size = 1.5)
p4<-ggplot(data = data4_temp_G,mapping = aes(氧化镁,..density..))+geom_histogram(color = "white",fill = "gray60",binwidth = 3)+geom_line(stat = "density",size = 1.5)
p5<-ggplot(data = data4_temp_G,mapping = aes(氧化铝,..density..))+geom_histogram(color = "white",fill = "gray60",binwidth = 3)+geom_line(stat = "density",size = 1.5)
p6<-ggplot(data = data4_temp_G,mapping = aes(氧化铁,..density..))+geom_histogram(color = "white",fill = "gray60",binwidth = 3)+geom_line(stat = "density",size = 1.5)
p7<-ggplot(data = data4_temp_G,mapping = aes(氧化铜,..density..))+geom_histogram(color = "white",fill = "gray60",binwidth = 3)+geom_line(stat = "density",size = 1.5)
p8<-ggplot(data = data4_temp_G,mapping = aes(五氧化二磷,..density..))+geom_histogram(color = "white",fill = "gray60",binwidth =
```

```
3)+geom_line(stat = "density",size = 1.5)
p1+p2+p3+p4+p5+p6+p7+p8
```

```
#铅钡类无风化
p9<-ggplot(data = data_temp_Q,mapping = aes( 二 氧 化 硅 ,..density..))+geom_histogram(color = "white",fill = "gray60",binwidth = 3)+geom_line(stat = "density",size = 1.5)
p11<-ggplot(data = data_temp_Q,mapping = aes( 氧 化 钙 ,..density..))+geom_histogram(color = "white",fill = "gray60",binwidth = 3)+geom_line(stat = "density",size = 1.5)
p13<-ggplot(data = data_temp_Q,mapping = aes( 氧 化 铝 ,..density..))+geom_histogram(color = "white",fill = "gray60",binwidth = 3)+geom_line(stat = "density",size = 1.5)
p14<-ggplot(data = data_temp_Q,mapping = aes( 氧 化 铁 ,..density..))+geom_histogram(color = "white",fill = "gray60",binwidth = 3)+geom_line(stat = "density",size = 1.5)
p15<-ggplot(data = data_temp_Q,mapping = aes( 氧 化 铜 ,..density..))+geom_histogram(color = "white",fill = "gray60",binwidth = 3)+geom_line(stat = "density",size = 1.5)
p16<-ggplot(data = data_temp_Q,mapping = aes( 五 氧 化 二 磷 ,..density..))+geom_histogram(color = "white",fill = "gray60",binwidth = 3)+geom_line(stat = "density",size = 1.5)
p17<-ggplot(data = data_temp_Q,mapping = aes( 氧 化 钠 ,..density..))+geom_histogram(color = "white",fill = "gray60",binwidth = 3)+geom_line(stat = "density",size = 1.5)
p18<-ggplot(data = data_temp_Q,mapping = aes( 氧 化 钡 ,..density..))+geom_histogram(color = "white",fill = "gray60",binwidth = 3)+geom_line(stat = "density",size = 1.5)
p19<-ggplot(data = data_temp_Q,mapping = aes( 氧 化 铅 ,..density..))+geom_histogram(color = "white",fill = "gray60",binwidth = 3)+geom_line(stat = "density",size = 1.5)
p19+p18+p17+p9+p11+p13+p14+p15+p16
```

```
#在对表单进行统计前，需要观察各类别化学成分的分布，去除确实值
#同时在统计前将已风化的未风化点的测量数据当作未风化类型来处理
data4_temp<-data4
level<-levels(data4[,5])
for(i in 1:60){
  if(data4[i,6] == "未风化点")
    data4_temp[i,5]<-level[2]
}
data4_temp[c(43,44),5]<-level[2]
```

```
#先以类型和表面风化为分组，统计各组化学成分含量的均值
#求出每组的均值
data4_table<-aggregate(data4_temp[,7:20],by = list(data4_temp$ 类型,data4_temp$表面风化),mean)
```

#对表单二进行分组统计分析(可视化)

```
data4_groupMean<-gather(data4_table,key = "化学成分",value = "含量",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,氧化锡,二氧化硫)
data4_groupMean<-data4_groupMean%>%rename(类型 = Group.1,表面风化 = Group.2)
data4_groupMean%>%ggplot(mapping = aes(x = 化学成分,
                                       y = 含量 ))+geom_col(aes(fill = 表面风化),
                                                            position = "dodge")+labs(title = "不同类型不同风化的含量统计")+facet_wrap(~类型)+theme(axis.text.x = element_text(angle = 45,

hjust = 0.5,

vjust = 0.5))
```

#建立模型根据风化点监测数据预测风化前的化学成分含量

```
data4_table<-data4_table%>%rename(类型 = Group.1,表面风化 = Group.2)
vector_GN<-data4_table%>%filter(类型 == "高钾",表面风化 == "无风化")#无风化时各个化学成分的统计量
vector_GN<-vector_GN[,-c(1:2)]
vector_GY<-data4_table%>%filter(类型 == "高钾",表面风化 == "风化")#风化后各个化学成分的统计量
vector_GY<-vector_GY[,-c(1,2)]
#计算高钾类无风化与风化化学成分的比例
ratio_G<-vector_GN/vector_GY
#利用该公式即可预测风化前的化学成分含量

vector_QN<-data4_table%>%filter(类型 == "铅钡",表面风化 == "无风化")#无风化时各个化学成分的统计量
vector_QN<-vector_QN[,-c(1:2)]
vector_QY<-data4_table%>%filter(类型 == "铅钡",表面风化 == "风化")#风化后各个化学成分的统计量
vector_QY<-vector_QY[,-c(1,2)]
#计算高钾类无风化与风化化学成分的比例
ratio_Q<-vector_QN/vector_QY
```

```
data4_before_G<-data4_temp%>%filter(类型 == "高钾",表面风化 == "风化")%>%select(1,3,5,7:20)
data4_before_Q<-data4_temp%>%filter(类型 == "铅钡",表面风化 == "风化")%>%select(1,3,5,7:20)
```

```
#对高钾类风化前进行预测
ratio_temp<-rbind(ratio_G,ratio_G)
for(i in 1:4){
  ratio_temp<-rbind(ratio_temp,ratio_G)
}
predict<-data4_before_G[,4:17]*ratio_temp
data4_predict_G<-data4_before_G
data4_predict_G[,4:17]<-predict[,1:14]
```

```
#对铅钡类风化前进行预测
ratio_temp<-rbind(ratio_Q,ratio_Q)
for(i in 1:22){
  ratio_temp<-rbind(ratio_temp,ratio_Q)
}
predict<-data4_before_Q[,4:17]*ratio_temp
data4_predict_Q<-data4_before_Q
data4_predict_Q[,4:17]<-predict[,1:14]

write.table(data4_predict_G,"高钾类风化点未风化前数据.csv",row.names = FALSE,sep = ",")
write.table(data4_predict_Q,"铅钡类风化点未风化前数据.csv",row.names = FALSE,sep = ",")
```

8.3 问题二代码：

```
library(NbClust)

#除了可视化外，在已知分类种类的情况下，还可以使用划分聚类分析来对数据进行划分，探究划分规律
data_weifenghua<-data4_temp%>%filter(表面风化 == "无风化")
data_weifenghua[2,7:20]<-(data_weifenghua[2,7:20]+data_weifenghua[3,7:20])/2
data_weifenghua<-data_weifenghua[-3,]
data_weifenghua[5,7:20]<-(data_weifenghua[5,7:20]+data_weifenghua[6,7:20])/2
data_weifenghua<-data_weifenghua[-6,]
data_weifenghua[17,7:20]<-(data_weifenghua[17,7:20]+data_weifenghua[18,7:20])/2
data_weifenghua<-data_weifenghua[-18,]
data_weifenghua[25,7:20]<-(data_weifenghua[26,7:20]+data_weifenghua[26,7:20])/2
data_weifenghua<-data_weifenghua[-26,]
```

#对数据进行标准化处理

```
df1_weifenghua<-data_weifenghua
df1_weifenghua[,7:20]<-scale(data_weifenghua[,7:20])
df1_weifenghua_scaled<-df1_weifenghua


set.seed(1234)#为了是结果具有可复制
fit.km<-kmeans(df1_weifenghua_scaled[,7:20],2,nstart = 25)
fit.km$centers
aggregate(data_weifenghua[7:20],by = list(cluster = fit.km$cluster),mean)
```

#因为高钾类都具有氧化钾，氧化钙，二氧化硅，因此应在分类时将这些变量去除，避免影响分类

```
#同时观察严重风化的二氧化硫含量极高，因此其应不属于玻璃的主要成分，也要去除
#对于氧化锡来说，缺失值太多，不具有代表性，也应去除
data_G<-data4_temp[,-c(7,8,9,10,19,20)]
data_G<-data_G%>%filter(类型=="高钾",表面风化  == "无风化")
data_G[2,7:14]<-(data_G[2,7:14]+data_G[3,7:14])/2
data_G<-data_G[-3,]
data_G[5,7:14]<-(data_G[5,7:14]+data_G[6,7:14])/2
data_G<-data_G[-6,]

df1_G<-data_G
row.names(df1_G)<-df1_G$文物编号
#df1_G[,7:17]<-apply(data_G[,7:17],2,function(x){(x)/max(x)})
df1_G[,7:14]<-scale(data_G[,7:14])
df1_G_scaled<-df1_G

#使用欧几里得距离来计算观测值间距
d<-dist(df1_G_scaled[,7:14])


#使用平均联动来计算类之间的距离
fit.average<-hclust(d,method = "average")
plot(fit.average,hang = .3,cex = 1,main = "Average Linkage Clustering")

#探究比较优良的聚类数目
nc<-NbClust(df1_G_scaled[,c(7:14)],distance = "euclidean",
                  min.nc = 2,max.nc = 4,method = "average")


clusters<-cutree(fit.average,k = 5)
```

```
table(clusters)
aggregate(data_G[7:14],by = list(cluster = clusters),mean)
plot(fit.average,hang = 1,cex = .8,main = "高钾类亚类划分")
rect.hclust(fit.average,k = 5)
```

#对于铅钡类玻璃来说，应该去除氧化铅，氧化钡，二氧化硅，氧化钙,二氧化硫的影响来对其进行分析

```
data_Q<-data4_temp[,-c(7,10,15,16,20)]
data_Q<-data_Q%>%filter(类型  == "铅钡",表面风化  == "无风化")
data_Q[7,7:15]<-(data_Q[7,7:15]+data_Q[8,7:15])/2
data_Q<-data_Q[-8,]
data_Q[13,7:15]<-(data_Q[13,7:15]+data_Q[14,7:15])/2
data_Q<-data_Q[-14,]
row.names(data_Q)<-data_Q$文物编号


#进行数据的归一化处理，方便计算
df1_Q<-data_Q
#df1_Q[,7:15]<-scale(data_Q[,7:15])
df1_Q_scaled<-df1_Q

#使用欧几里得距离来计算观测值间距
d1<-dist(df1_Q_scaled[,7:15])
as.matrix(d1)[1:4,1:8]

#使用平均联动来计算类之间的距离
fit.average1<-hclust(d1,method = "average")
plot(fit.average1,hang = .3,cex = 1,main = "Average Linkage Clustering")

nc<-NbClust(data_Q[,c(7:15)],distance = "euclidean",
                  min.nc = 2,max.nc = 5,method = "average")

par(mfrow = c(1,1))
barplot(table(nc$Best.nc[1,]),
            xlab = "分类的数量",ylab = "评价准则的数量",
            main = "被准则推荐的分类个数")
#从图中可以看出最推荐的分类个数是 2 或 5 显然 5 分类更好解释一些

#同时为了可解释度，选取 k 为 5,进行解释
clusters<-cutree(fit.average1,k = 4)
table(clusters)
aggregate(data_Q[7:15],by = list(cluster = clusters),mean)
plot(fit.average1,hang = .3,cex = 1,main = "铅钡类亚类划分")
rect.hclust(fit.average1,k = 4)
```

```
#分类结果的敏感性分析
#根据分出的亚类可知，当改变某个亚类中较为重要的变量值，则分类方式会发生变化
#根据高钾类中的第四类为代表进行敏感性分析分析，先在较小的范围内改变氧化铅含量
data_G[7,11]<-data_G[7,11]*0.8
df1_G[,7:14]<-scale(data_G[,7:14])
df1_G_scaled<-df1_G
d<-dist(df1_G_scaled[,7:14])
fit.average<-hclust(d,method = "average")
clusters<-cutree(fit.average,k = 5)
table(clusters)
aggregate(data_G[7:14],by = list(cluster = clusters),mean)
plot(fit.average,hang = 1,cex = .8,main = "对 14 号文物轻微改动高钾类亚类划分")
rect.hclust(fit.average,k = 5)
```

```
#再对氧化铅含量进行较大的改动
data_G[7,11]<-data_G[7,11]-1
df1_G[,7:14]<-scale(data_G[,7:14])
df1_G_scaled<-df1_G
d<-dist(df1_G_scaled[,7:14])
fit.average<-hclust(d,method = "average")
clusters<-cutree(fit.average,k = 5)
table(clusters)
aggregate(data_G[7:14],by = list(cluster = clusters),mean)
plot(fit.average,hang = 1,cex = .8,main = "对 14 号文物大幅改动后的高钾类亚类划分")
rect.hclust(fit.average,k = 5)
```

```
#对铅钡类划分进行敏感性分析
#以 24 号文物为例，其氧化铜含量比较突出，现在先对其氧化铜含量进行小范围改动，期望使不会发生划分的变化
data_Q[3,12]<-data_Q[3,12]*0.8
df1_Q<-data_Q
df1_Q_scaled<-df1_Q
d1<-dist(df1_Q_scaled[,7:15])
fit.average1<-hclust(d1,method = "average")
clusters<-cutree(fit.average1,k = 4)
table(clusters)
aggregate(data_Q[7:15],by = list(cluster = clusters),mean)
```

```
plot(fit.average1,hang = .3,cex = 1,main = "对 24 号文物轻微改动后铅钡类亚类划分")
rect.hclust(fit.average1,k = 4)

#对 24 号文物的氧化铜含量大幅改动，期望划分会发生变化
data_Q[3,12]<-data_Q[3,12]*0.2
df1_Q<-data_Q
df1_Q_scaled<-df1_Q
d1<-dist(df1_Q_scaled[,7:15])
fit.average1<-hclust(d1,method = "average")
clusters<-cutree(fit.average1,k = 4)
table(clusters)
aggregate(data_Q[7:15],by = list(cluster = clusters),mean)
plot(fit.average1,hang = .3,cex = 1,main = "对 24 号文物大幅改动后铅钡类亚类划分")
rect.hclust(fit.average1,k = 4)

#可以看到文物划分发生了较大的改变
#通过上面的敏感性分析，两类的亚类划分结果的敏感性也较好，即小范围变动不会影响划分，而大范围变动则会对划分产生影响
```

8.4 问题三代码：

```
#因为化学成分较多，且变量之间或许会有相关关系，因此使用主成分分析进行降维处理，变成较少的不相干变量

#首先对不同化学成分之间的相关性进行比较
library(polycor)
library(corrplot)
corMatrix<-cor(data4_temp[,7:20])
#画出相关系数矩阵
corrplot::corrplot(corMatrix,method = "number")

#检验要选择的主成分个数
fa.parallel(data4_temp[,7:20],fm = "ml",fa = "pc",n.iter = 100,
            show.legend = FALSE,main = "Scree plot with parallel analysis")
#从图中可以看出，推荐我们保留三个主成分

#进行主成分分析
library(psych)
pc<-principal(data4_temp[,7:20],nfactors = 3,rotate = "none")
pc
#当提取了多个成分时，对他们进行旋转可以使结果更具有解释性
#旋转只会使各个主成分的方差解释性发生变化，但累加方差解释性使没有变
```

```
化
rc<-principal(data4_temp[,7:20],nfactors = 3,rotate = "varimax",score = TRUE)
rc
#从得到的结果可以看到，第一个主成分主要由二氧化硅，氧化铅，五氧化二磷，氧化锶来解释
#第二个主成分主要由氧化钾，氧化钙，氧化镁，氧化铝，氧化铁来解释
#第三个主成分主要由氧化铜，氧化钡，二氧化硫来解释

#获得主成分得分系数,方便对新数据进行主成分降维
round(unclass(rc$weights),2)

#为了能够使主成分代替原有变量，需要获取每个观测在成分上的得分
head(rc$scores)

#将主成分得分添加到数据当中
pca_scores<-rc$scores
data4_temp[,21]<-pca_scores[,1]
data4_temp[,22]<-pca_scores[,2]
data4_temp[,23]<-pca_scores[,3]
colnames(data4_temp)[21]<-"RC1"
colnames(data4_temp)[22]<-"RC2"
colnames(data4_temp)[23]<-"RC3"

#只提取出要用来分类的变量
data4_pca<-data4_temp%>%select(3,5,21,22,23)

#划分训练集与测试集
set.seed(1234)
train<-sample(nrow(data4_pca),0.7*nrow(data4_pca))
df.train<-data4_pca[train,]
df.test<-data4_pca[-train,]

#使用逻辑回归
fit.logit<-glm(类型~.,data = df.train,family = binomial(),maxit = 200)
summary(fit.logit)
prob<-predict(fit.logit,df.test,type = "response")
logit.perd<-factor(prob>.5,levels = c(FALSE,TRUE),
                           labels = c("高钾","铅钡"))
logit.perf<-table(df.test$类型,logit.perd,
                          dnn = c("Actual","Predicted"))
logit.perf
#根据报错信息可以看出逻辑回归出现了过拟合问题，这种情况就是因为数据样本本身就属于线性可分的
#同时根据混淆矩阵发现分类完全正确
```

```
#因此使用逻辑回归就会出现过拟合现象，因此不再使用这种方法
```

#经典决策树

```
library(rpart)
dtree<-rpart(类型~.,data = df.train,method = "class",
                    parms = list(split = "information"))
dtree$cptable
plotcp(dtree)
dtree_prune<-prune(dtree,cp = 0.096)
prp(dtree_prune,
    type = 2,
    tweak = 1.2,
    extra = 104,
    fallen.leaves = T,
    main = "Decision Tree")
dtree.pred<-predict(dtree_prune,df.test,type = "class")
dtree.perf<-table(df.test$类型,dtree.pred,
                          dnn = c("Actual","Predicted"))
dtree.perf
#library(partykit)
#plot(as.party(dtree_prune))
```

#支持向量机 SVM

```
library(rgl)
a<-data4_pca$类型
levels(a)<-c("red","green")
open3d()
mfrow3d(1,2)
plot3d(data4_pca$RC1,data4_pca$RC2,data4_pca$RC3,col = a)
grid3d(side = "x")
grid3d(side = "y")
grid3d(side = "z")

#从图中可以看出高钾类和铅钡类是线性可分的，因此我们可以使用线性可分向量机来进行训练
library(e1071)
set.seed(1234)
#先使用线性核函数来进行训练
fit.svm<-svm(类型~.,data = df.train,kernel = "linear")
tuned_linear<-tune.svm(类型~.,data = df.train,
                          cost = 10^(-10:10))
```

```
svm.pred<-predict(fit.svm,df.test)
svm.perf<-table(df.test$类型,svm.pred,
                dnn = c("Actual","Predicted"))


tuned_radial<-tune.svm(类型~.,data = df.train,
                       gamma = 10^(-6:1),
                       cost = 10^(-10:10))
tuned_radial
fit.svm_radial<-svm(类型~.,data = df.train,gamma = 0.01,cost = 10)
svm.pred_radial<-predict(fit.svm_radial,df.test)
svm.perf_radial<-table(df.test$类型,svm.pred_radial,
                dnn = c("Actual","Predicted"))
```

```
#对于表单三中的数据进行处理
for(i in 1:8){
   for(j in 3:16){
      if(is.na(data3[i,j]))
         data3[i,j]<-0
   }
}

#对表单三的数据进行降维
#在使用公式计算前，先对数据进行标准化
data3[,3:16]<-scale(data3[,3:16])
data3[,17]<-(-0.29)*data3[,3]+(-0.06)*data3[,4]+(-0.16)*data3[,5]+(0.03)*data3[,6]+(
0.1)*data3[,7]+(-0.04)*data3[,8]+(0.02)*data3[,9]+(-0.06)*data3[,10]+(0.29)*data3[,
11]+(0.09)*data3[,12]+(0.23)*data3[,13]+(0.23)*data3[,14]+(0.00)*data3[,15]+(-0.07
)*data3[,16]
data3[,18]<-(-0.09)*data3[,3]+(-0.10)*data3[,4]+(0.22)*data3[,5]+(0.31)*data3[,6]+(
0.23)*data3[,7]+(0.2)*data3[,8]+(0.27)*data3[,9]+(0.1)*data3[,10]+(-0.05)*data3[,11
]+(-0.05)*data3[,12]+(0.18)*data3[,13]+(0.01)*data3[,14]+(0.00)*data3[,15]+(0.04)*
data3[,16]
data3[,19]<-(-0.05)*data3[,3]+(-0.03)*data3[,4]+(0.12)*data3[,5]+(0.08)*data3[,6]+(-
0.14)*data3[,7]+(-0.05)*data3[,8]+(0.01)*data3[,9]+(0.45)*data3[,10]+(-0.11)*data3[
,11]+(0.27)*data3[,12]+(-0.01)*data3[,13]+(0.00)*data3[,14]+(-0.13)*data3[,15]+(-0.
38)*data3[,16]

colnames(data3)[c(17,18,19)]<-c("RC1","RC2","RC3")
rownames(data3)<-data3$文物编号
data3<-data3%>%select(表面风化,RC1,RC2,RC3)
pred<-predict(fit.svm,newdata = data3)
pred
```

```
data3_5<-data3

#对模型进行敏感性分析
#首先将变量范围扩大 5%
for (i in 1:8){
   for(j in 3:16){
      data3_5[i,j]<-runif(1,min = data3_5[i,j]*0.95,max = data3_5[i,j]*1.05)
   }
}
data3_5[,3:16]<-scale(data3_5[,3:16])
data3_5[,17]<-(-0.29)*data3_5[,3]+(-0.06)*data3_5[,4]+(-0.16)*data3_5[,5]+(0.03)*d
ata3_5[,6]+(0.1)*data3_5[,7]+(-0.04)*data3_5[,8]+(0.02)*data3_5[,9]+(-0.06)*data3_
5[,10]+(0.29)*data3_5[,11]+(0.09)*data3_5[,12]+(0.23)*data3_5[,13]+(0.23)*data3_
5[,14]+(0.00)*data3_5[,15]+(-0.07)*data3_5[,16]
data3_5[,18]<-(-0.09)*data3_5[,3]+(-0.10)*data3_5[,4]+(0.22)*data3_5[,5]+(0.31)*d
ata3_5[,6]+(0.23)*data3_5[,7]+(0.2)*data3_5[,8]+(0.27)*data3_5[,9]+(0.1)*data3_5[,
10]+(-0.05)*data3_5[,11]+(-0.05)*data3_5[,12]+(0.18)*data3_5[,13]+(0.01)*data3_5
[,14]+(0.00)*data3_5[,15]+(0.04)*data3_5[,16]
data3_5[,19]<-(-0.05)*data3_5[,3]+(-0.03)*data3_5[,4]+(0.12)*data3_5[,5]+(0.08)*d
ata3_5[,6]+(-0.14)*data3_5[,7]+(-0.05)*data3_5[,8]+(0.01)*data3_5[,9]+(0.45)*data3
_5[,10]+(-0.11)*data3_5[,11]+(0.27)*data3_5[,12]+(-0.01)*data3_5[,13]+(0.00)*data
3_5[,14]+(-0.13)*data3_5[,15]+(-0.38)*data3_5[,16]
colnames(data3_5)[c(17,18,19)]<-c("RC1","RC2","RC3")
#此时在使用模型进行预测
pred_5<-predict(fit.svm,newdata = data3_5)


#对变量范围扩大 10%
data3_10<-data3
for (i in 1:8){
   for(j in 3:16){
      data3_10[i,j]<-runif(1,min = data3_10[i,j]*0.9,max = data3_10[i,j]*1.1)
   }
}
data3_10[,3:16]<-scale(data3_10[,3:16])
data3_10[,17]<-(-0.29)*data3_10[,3]+(-0.06)*data3_10[,4]+(-0.16)*data3_10[,5]+(0.
03)*data3_10[,6]+(0.1)*data3_10[,7]+(-0.04)*data3_10[,8]+(0.02)*data3_10[,9]+(-0.
06)*data3_10[,10]+(0.29)*data3_10[,11]+(0.09)*data3_10[,12]+(0.23)*data3_10[,13]
+(0.23)*data3_10[,14]+(0.00)*data3_10[,15]+(-0.07)*data3_10[,16]
data3_10[,18]<-(-0.09)*data3_10[,3]+(-0.10)*data3_10[,4]+(0.22)*data3_10[,5]+(0.3
1)*data3_10[,6]+(0.23)*data3_10[,7]+(0.2)*data3_10[,8]+(0.27)*data3_10[,9]+(0.1)*
data3_10[,10]+(-0.05)*data3_10[,11]+(-0.05)*data3_10[,12]+(0.18)*data3_10[,13]+(
0.01)*data3_10[,14]+(0.00)*data3_10[,15]+(0.04)*data3_10[,16]
```

```
data3_10[,19]<-(-0.05)*data3_10[,3]+(-0.03)*data3_10[,4]+(0.12)*data3_10[,5]+(0.0
8)*data3_10[,6]+(-0.14)*data3_10[,7]+(-0.05)*data3_10[,8]+(0.01)*data3_10[,9]+(0.
45)*data3_10[,10]+(-0.11)*data3_10[,11]+(0.27)*data3_10[,12]+(-0.01)*data3_10[,1
3]+(0.00)*data3_10[,14]+(-0.13)*data3_10[,15]+(-0.38)*data3_10[,16]
colnames(data3_10)[c(17,18,19)]<-c("RC1","RC2","RC3")
pred_10<-predict(fit.svm,newdata = data3_10)


#对变量范围扩大 20%
data3_20<-data3
for (i in 1:8){
   for(j in 3:16){
      data3_20[i,j]<-runif(1,min = data3_20[i,j]*0.8,max = data3_20[i,j]*1.2)
   }
}
data3_20[,3:16]<-scale(data3_20[,3:16])
data3_20[,17]<-(-0.29)*data3_20[,3]+(-0.06)*data3_20[,4]+(-0.16)*data3_20[,5]+(0.
03)*data3_20[,6]+(0.1)*data3_20[,7]+(-0.04)*data3_20[,8]+(0.02)*data3_20[,9]+(-0.
06)*data3_20[,10]+(0.29)*data3_20[,11]+(0.09)*data3_20[,12]+(0.23)*data3_20[,13]
+(0.23)*data3_20[,14]+(0.00)*data3_20[,15]+(-0.07)*data3_20[,16]
data3_20[,18]<-(-0.09)*data3_20[,3]+(-0.10)*data3_20[,4]+(0.22)*data3_20[,5]+(0.3
1)*data3_20[,6]+(0.23)*data3_20[,7]+(0.2)*data3_20[,8]+(0.27)*data3_20[,9]+(0.1)*
data3_20[,10]+(-0.05)*data3_20[,11]+(-0.05)*data3_20[,12]+(0.18)*data3_20[,13]+(
0.01)*data3_20[,14]+(0.00)*data3_20[,15]+(0.04)*data3_20[,16]
data3_20[,19]<-(-0.05)*data3_20[,3]+(-0.03)*data3_20[,4]+(0.12)*data3_20[,5]+(0.0
8)*data3_20[,6]+(-0.14)*data3_20[,7]+(-0.05)*data3_20[,8]+(0.01)*data3_20[,9]+(0.
45)*data3_20[,10]+(-0.11)*data3_20[,11]+(0.27)*data3_20[,12]+(-0.01)*data3_20[,1
3]+(0.00)*data3_20[,14]+(-0.13)*data3_20[,15]+(-0.38)*data3_20[,16]
colnames(data3_20)[c(17,18,19)]<-c("RC1","RC2","RC3")
pred_20<-predict(fit.svm,newdata = data3_20)

#对变量范围扩大 30%
data3_30<-data3
for (i in 1:8){
   for(j in 3:16){
      data3_30[i,j]<-runif(1,min = data3_30[i,j]*0.7,max = data3_30[i,j]*1.3)
   }
}
data3_30[,3:16]<-scale(data3_30[,3:16])
data3_30[,17]<-(-0.29)*data3_30[,3]+(-0.06)*data3_30[,4]+(-0.16)*data3_30[,5]+(0.
03)*data3_30[,6]+(0.1)*data3_30[,7]+(-0.04)*data3_30[,8]+(0.02)*data3_30[,9]+(-0.
06)*data3_30[,10]+(0.29)*data3_30[,11]+(0.09)*data3_30[,12]+(0.23)*data3_30[,13]
+(0.23)*data3_30[,14]+(0.00)*data3_30[,15]+(-0.07)*data3_30[,16]
data3_30[,18]<-(-0.09)*data3_30[,3]+(-0.10)*data3_30[,4]+(0.22)*data3_30[,5]+(0.3
```

```
1)*data3_30[,6]+(0.23)*data3_30[,7]+(0.2)*data3_30[,8]+(0.27)*data3_30[,9]+(0.1)*
data3_30[,10]+(-0.05)*data3_30[,11]+(-0.05)*data3_30[,12]+(0.18)*data3_30[,13]+(
0.01)*data3_30[,14]+(0.00)*data3_30[,15]+(0.04)*data3_30[,16]
data3_30[,19]<-(-0.05)*data3_30[,3]+(-0.03)*data3_30[,4]+(0.12)*data3_30[,5]+(0.0
8)*data3_30[,6]+(-0.14)*data3_30[,7]+(-0.05)*data3_30[,8]+(0.01)*data3_30[,9]+(0.
45)*data3_30[,10]+(-0.11)*data3_30[,11]+(0.27)*data3_30[,12]+(-0.01)*data3_30[,1
3]+(0.00)*data3_30[,14]+(-0.13)*data3_30[,15]+(-0.38)*data3_30[,16]
colnames(data3_30)[c(17,18,19)]<-c("RC1","RC2","RC3")
pred_30<-predict(fit.svm,newdata = data3_30)

#分析改变变量范围以后划分的变化情况与未扰动时进行对比
#发现所有的划分均不发生变动，故该模型较稳定
```

```
#可视化
library(patchwork)
table_5<-table(pred,pred_5)
p1<-ggplot(data = as.data.frame(table_5),
           mapping = aes(pred,pred_5))+geom_tile(aes(fill =
Freq))+geom_text(aes(label = Freq),col = "khaki1")

table_10<-table(pred,pred_10)
p2<-ggplot(data = as.data.frame(table_10),
           mapping = aes(pred,pred_10))+geom_tile(aes(fill =
Freq))+geom_text(aes(label = Freq),col = "khaki1")

table_20<-table(pred,pred_20)
p3<-ggplot(data = as.data.frame(table_20),
           mapping = aes(pred,pred_20))+geom_tile(aes(fill =
Freq))+geom_text(aes(label = Freq),col = "khaki1")

table_30<-table(pred,pred_30)
p4<-ggplot(data = as.data.frame(table_30),
           mapping = aes(pred,pred_30))+geom_tile(aes(fill =
Freq))+geom_text(aes(label = Freq),col = "khaki1")

p1+p2+p3+p4
```

8.5 问题四代码：

```
#先绘制变量之间的散点图关系初步探究变量之间的关系

#对不同类别的变量进行 R 型聚类分析，观察变量之间的关联关系
```

```
library(factoextra)
library(cluster)
data_GY<-data4_temp%>%filter(类型 == "高钾",表面风化 == "无风化")
data_GY<-data_GY%>%select(c(7:20))
data_GY_t<-t(data_GY)
#对数据值进行标准化
data_GY_t_scaled<-scale(data_GY_t)
d1<-dist(data_GY_t_scaled)
#确定最优聚类数量
#寻找最佳聚类个数，使用的是总体平方和
fviz_nbclust(data_GY_t_scaled,kmeans,method = "wss")
set.seed(1234)
km1<-kmeans(data_GY_t_scaled,centers = 3,nstart = 25)
fviz_cluster(km1,data = data_GY_t_scaled)
```

```
#对高钾类风化数据进行聚类分析
#对不同类别的变量进行 R 型聚类分析，观察变量之间的关联关系
data_GN<-data4_temp%>%filter(类型 == "高钾",表面风化 == "风化")
data_GN<-data_GN%>%select(c(7:20))
data_GN_t<-t(data_GN)
#对数据值进行标准化
data_GN_t_scaled<-scale(data_GN_t)
d2<-dist(data_GN_t_scaled)
#确定最优聚类数量
#寻找最佳聚类个数，使用的是总体平方和
fviz_nbclust(data_GN_t_scaled,kmeans,method = "wss",k.max = 6)
set.seed(1234)
km2<-kmeans(data_GN_t_scaled,centers = 2,nstart = 25)
fviz_cluster(km2,data = data_GN_t_scaled)
```

```
#对铅钡类无风化数据进行聚类分析
#对不同类别的变量进行 R 型聚类分析，观察变量之间的关联关系
data_QN<-data4_temp%>%filter(类型 == "铅钡",表面风化 == "无风化")
data_QN<-data_QN%>%select(c(7:20))
data_QN_t<-t(data_QN)
#对数据值进行标准化
data_QN_t_scaled<-scale(data_QN_t)
d3<-dist(data_QN_t_scaled)
#确定最优聚类数量
#寻找最佳聚类个数，使用的是总体平方和
```

```
fviz_nbclust(data_QN_t_scaled,kmeans,method = "wss")
set.seed(1234)
km3<-kmeans(data_QN_t_scaled,centers = 4,nstart = 25)
fviz_cluster(km3,data = data_QN_t_scaled)
```

```
#对铅钡类风化数据进行聚类分析
#对不同类别的变量进行 R 型聚类分析，观察变量之间的关联关系
data_QY<-data4_temp%>%filter(类型 == "铅钡",表面风化 == "风化")
data_QY<-data_QY%>%select(c(7:20))
data_QY_t<-t(data_QY)
#对数据值进行标准化
data_QY_t_scaled<-scale(data_QY_t)
d4<-dist(data_QY_t_scaled)
#确定最优聚类数量
#寻找最佳聚类个数，使用的是总体平方和
fviz_nbclust(data_QY_t_scaled,kmeans,method = "wss")
set.seed(1234)
km4<-kmeans(data_QY_t_scaled,centers = 3,nstart = 25)
fviz_cluster(km4,data = data_QY_t_scaled)
```

```
#建立评价指标函数
#DBI 指数，越小代表聚类效果越好
calDBI <- function(x=data,labels=labesls)

  ##data 必须行为样本，列为特征
{
  clusters_n <- length(unique(labels))
  cluster_k <- list()
  for (i in c(1:clusters_n)) {
    cluster_k[[i]] <- x[which(labels==i),]
  }

  centroids <- list()
  for (i in c(1:clusters_n)) {
    centroids[[i]] <- apply(cluster_k[[i]],2,mean)
  }

  s <- list()
  for (i in c(1:clusters_n)) {
    a <- c()
    for (j in c(1:nrow(cluster_k[[i]]))) {
      b <- dist(rbind(cluster_k[[i]][j,],centroids[[i]]),method = "euclidean")
      a <- c(a,b)
```

```
    }
    s[[i]] <- mean(a)
  }

  Ri <- list()
  for (i in c(1:clusters_n)){
    r <- c()
    for (j in c(1:clusters_n)){
      if (j!=i){
        h <- (s[[i]]+s[[j]])/dist(rbind(centroids[[i]],centroids[[j]]),method =
"euclidean")
        r <- c(r,h)
      }
    }
    Ri[[i]] <- max(r)
  }
  dbi <- mean(unlist(Ri))
  return(dbi)
}

#CH 指数，越大说明聚类效果越好
calCH <- function(X,labels){
  ##X 必须行为样本，列为特征
  labels_n <- length(unique(labels))
  samples_n <- nrow(X)
  X_mean <- apply(X,2,mean)
  ex_disp <- c()
  in_disp <- c()
  for (i in c(1:labels_n)) {
    cluster_k <- X[which(labels==i),]
    mean_k <- apply(cluster_k,2,mean)
    a1 <- nrow(cluster_k)*sum((mean_k-X_mean)^2)
    ex_disp <- c(ex_disp,a1)
    a2 <- sum((t(t(cluster_k)-mean_k))^2)
    in_disp <- c(in_disp,a2)
  }
  k1<- sum(ex_disp)
  k2<- sum(in_disp)
  if(k2==0)
  {
    return(1)
  }
  else
  {
```

```
    return((k1*(samples_n-labels_n))/(k2*(labels_n-1)))
  }
}


p1<-fviz_nbclust(data_GY_t_scaled,kmeans,method = "wss")+geom_vline(xintercept = 3,linetype = 2)+labs(title = "高钾类无风化最优分类个数")
p2<-fviz_nbclust(data_GN_t_scaled,kmeans,method = "wss",k.max = 6)+geom_vline(xintercept = 2,linetype = 2)+labs(title = "高钾类风化最优分类个数")
p3<-fviz_nbclust(data_QN_t_scaled,kmeans,method = "wss")+geom_vline(xintercept = 4,linetype = 2)+labs(title = "铅钡类无风化最优分类个数")
p4<-fviz_nbclust(data_QY_t_scaled,kmeans,method = "wss")+geom_vline(xintercept = 3,linetype = 2)+labs(title = "铅钡类风化最优分类个数")
p1+p2+p3+p4

p5<-fviz_cluster(km1,data = data_GY_t_scaled)+labs(title = "高钾类无风化分类图")
p6<-fviz_cluster(km2,data = data_GN_t_scaled)+labs(title = "高钾类风化分类图")
p7<-fviz_cluster(km3,data = data_QN_t_scaled)+labs(title = "铅钡类无风化分类图")
p8<-fviz_cluster(km4,data = data_QY_t_scaled)+labs(title = "铅钡类风化分类图")
p5+p6+p7+p8
```

#对不同的聚类得出不同的聚类指标

```
#sc 指数越接近一代表内聚度和分离度较好
#对高钾类无风化聚类结果分析
sc1<-silhouette(km1$cluster,d1)
sc1<-mean(sc1[,3])
dbi1<-calDBI(as.data.frame(data_GY_t_scaled),km1$cluster)
ch1<-calCH(as.data.frame(data_GY_t_scaled),km1$cluster)

#对高钾类风化聚类结果分析
sc2<-silhouette(km2$cluster,d2)
sc2<-mean(sc2[,3])
dbi2<-calDBI(as.data.frame(data_GN_t_scaled),km2$cluster)
ch2<-calCH(as.data.frame(data_GN_t_scaled),km2$cluster)
```

```
#对铅钡类无风化聚类结果分析
sc3<-silhouette(km3$cluster,d3)
sc3<-mean(sc3[,3])
dbi3<-calDBI(as.data.frame(data_QN_t_scaled),km3$cluster)
ch3<-calCH(as.data.frame(data_QN_t_scaled),km3$cluster)
```

```
#对铅钡类风化聚类结果分析
sc4<-silhouette(km4$cluster,d4)
sc4<-mean(sc4[,3])
dbi4<-calDBI(as.data.frame(data_QY_t_scaled),km4$cluster)
ch4<-calCH(as.data.frame(data_QY_t_scaled),km4$cluster)
```

#分析比较不同聚类结果可以得出不同类别间化学成分关联度的差异性