



中国研究生创新实践系列大赛
“华为杯”第二十届中国研究生
数学建模竞赛

学 校 温州大学

参赛队号 23103510013

1.吴玉龙

队员姓名 2.范永希

3.吴欣怡

中国研究生创新实践系列大赛 “华为杯”第二十届中国研究生 数学建模竞赛

题 目： 大规模创新类竞赛评审方案研究

摘 要：

随着科技和社会的发展，越来越多的创新类竞赛应运而生。其中较大规模的竞赛通常采用两阶段或三阶段的评估模式。然而，由于创新竞赛的特点是开放式问题，没有明确的标准答案，所以如何正确地、合理地对作品进行评价和打分，成为了当下需要解决的问题。

针对问题一：首先定义作品-专家 0-1 矩阵，并根据题目已知条件，建立约束条件。为了衡量不同评审专家之间的可比性，定义**相似性度量因子**。考虑专家相互之间具有高相似性并且差距较小，从而定义目标函数，建立**多目标 0-1 规划模型**^[1]，并基于时空复杂度的考量，提出在有限可行域之下的**搜索优化算法**，改进原始模型。

针对问题二：从“个体”与“总体”角度出发，使用多种统计检验方式，深度挖掘专家、作品信息，从而发现数据异常排名，进行合理修正。提出 **spearman 系数**^[2]与 **kendall 系数**^[3]，用于衡量排名变动。为了消除样本正态性假定，我们提出 **Max-min scale** 和 **Mean-scale** 替代标准分计算，并提出**震荡因子理论**修正计算，修正之后的两个方案分别在 spearman 系数上提升 **247%**和 **262%**，在 kendall 系数上，提升 **365%**和 **343%**。进一步提出**偏差因子理论**，拟合函数关系，最优方案的 spearman 系数为 **0.83689**，kendall 系数为 **0.65079**，效果击败二阶段标准分模型。

针对问题三：首先对数据 2.1，数据 2.2 进行排名修正，异常值处理。分析极差与成绩变动特点，深刻考虑极差阈值、极差区间的变动情况，建立**极差修正模型**。综合问题二最优方案（方案六），做出进一步改进，提出**方案八**，相比于方案六效果分别提升了 **15.43%**和 **29.49%**。对于数据 2.1，spearman 系数为 **0.9660**，kendall 系数为 **0.8427**；对于数据 2.2，spearman 系数为 **0.8788**，kendall 系数为 **0.7047**，并针对一阶段（中等分段）数据，提出**校正模型**，对分数进行“复议”。

针对问题四：提出完整的评审模型。使用基于搜索优化算法的 0-1 规划模型得出分配方案，在一阶段中使用校正模型，二阶段中使用方案八模型，最终得出成绩。方案改进方面，新颖性的提出利用**量表问卷**，获取专家打分倾向，进而修正模型。

关键词：0-1 规划；秩相关系数；震荡因子；偏差因子

目录

大规模创新类竞赛评审方案研究	1
一、 问题重述	3
1.1 问题背景	3
1.2 问题重述	3
二、 模型假设与符号说明	3
2.1 模型假设	3
2.2 符号说明	3
三、 问题 1 的模型建立与求解	4
3.1 问题分析	4
3.2 模型建立与求解	5
3.2.1 指标定义	5
3.2.2 相似性度量	5
3.2.3 多目标 0-1 规划求解	5
3.2.4 搜索优化算法	6
四、 问题 2 的模型建立与求解	7
4.1 问题分析	7
4.2 模型建立与求解	8
4.2.1 描述性统计分析	8
4.2.2 原始评审方案分析	14
4.2.3 模型提出评估	18
4.2.4 改进模型	20
4.2.5 最终评估结果展示	25
五、 问题 3 的模型建立与求解	25
5.1 问题分析	25
5.2 模型建立与求解	26
5.2.1 数据预处理	26
5.2.2 成绩与极差变动分析	27
5.2.3 极差调整分析	29
5.2.4 建立极差模型	31
5.2.5 方案六模型改进	33
5.2.6 模型评估	34
5.2.7 中等分段作品处理	34
六、 问题 4 的模型建立与求解	36
6.1 问题分析	36
6.2 模型建立与求解	36
6.2.1 评审方案流程设计	36
6.2.2 方案改进	37
七、 模型评价	39
7.1 模型的优点	39
7.2 模型的缺点	39
八、 参考文献	40

一、问题重述

1.1 问题背景

随着科技和社会的发展，越来越多的创新类竞赛应运而生。其中较大规模的竞赛通常采用两阶段或三阶段的评估模式。然而，由于创新竞赛的特点是开放式问题，没有明确的标准答案，所以如何正确地、合理地对作品进行评价和打分，成为了当下需要解决的问题。

一般的评审方案往往采取标准化评分、去除最高最低分以及出现分歧重新评分的三种方式进行打分。尽管这些方案都有其合理性，但也存在局限性，特别是在大规模创新竞赛评审方面，目前的方法相对简单且研究有限。此外，由于不可抗力因素，专家数量通常受限，所以我们需要在现有的打分机制以及限制条件下进行改良，寻求更加合理公平的评审方案。

1.2 问题重述

基于 1.1 的背景之下，现我们需要解决以下四个问题

问题一：现有 3000 支队伍，125 位评委，在确保每份作品由 5 位专家陪审的情况下，制定一个最优的“交叉分配”方案，该方案要求每位专家的作品集合的交集尽量大，同时也要保证任意两位评委的作品交集大小相似。

问题二：题中指出，若以标准分为基准进行排名，其前提假设与客观事实不符，因此要求选择两种或者两种以上现有的评审方案并结合附件数据，分析不同标准之下的成绩分布特点，据此制定一套合理的评分标准，最后以数据 1 中的二阶段排名作为现实依据，完善之前提出的模型

问题三：在大规模的创新比赛中，通常涉及开放性题目，也就是没有标准答案，这也导致评委的打分往往不一致，因此同一作品的成绩可能会有很大的差异。所以需要对极差高的作品进行调整，而低分段的作品不在获奖的范畴内，所以调整低分段的意义不是很大，也就需要将重心放在高分段作品的调整中。

题目要求，根据提供的已知数据 2.1 和 2.2，探讨两阶段成绩和两阶段极差的整体变化，分析两阶段评审方案和不分阶段评审方案的优劣，并建立极差模型，在已知数据的情况下，给出一种程序化处理方式，以应对中等分段作品出现“大极差”情况。

问题四：题目要求我们给出一个完整的方案流程，并且回答在已有数据的情况之下，如何进行求解，并基于现有方案，给出改进建议

二、模型假设与符号说明

2.1 模型假设

- (1) 每位专家的评阅份数差距不大
- (2) 专家的打分水平不会因为疲惫等因素而变化
- (3) 入选二阶段的人数不会发生变化（问题 2）
- (4) 原始打分真实可靠
- (5) 二阶段排名为权威排名（问题 3）
- (6) 排名顺序由分数决定（异常值）

2.2 符号说明

符号	含义
$v_{i,j}$	第 <i>i</i> 个作品是否被第 <i>j</i> 个专家评审
$s_{i,j}$	相似性度量因子
Z_{mean}	度量矩阵均值
Z_{var}	度量矩阵方差
$o(P)$	容忍度
γ	震荡因子
δ	偏差因子
\hat{y}	最终成绩
\bar{y}	拟合成绩

三、问题 1 的模型建立与求解

3.1 问题分析

根据 1.2 的问题重述，我们知道，本问题的核心在于使用交集的大小来度量“可比性”，所以我们需要对交集指标进行量化，并且根据题目已知条件，将其余变量转化为定量指标，从而求解该规划问题。

所以第一问的解答需要分以下步骤

（1）将题目中已知的条件（例如：参赛队数目，专家数目）进行量化，将数目大小转化为索引值。

（2）根据“可比性”含义，建立相似性度量因子，提取不同专家集合之间的相似性信息。

（3）根据相似性度量因子，建立相似性矩阵，并且根据范式，建立评估因子，将该问题转化为含有约束的规划问题，并求解最终的专家评卷分配矩阵。

（4）为了节省内存和时间，使用搜索优化算法，改进模型。

具体流程如图 1 所示，详细的步骤见 3.2 节。

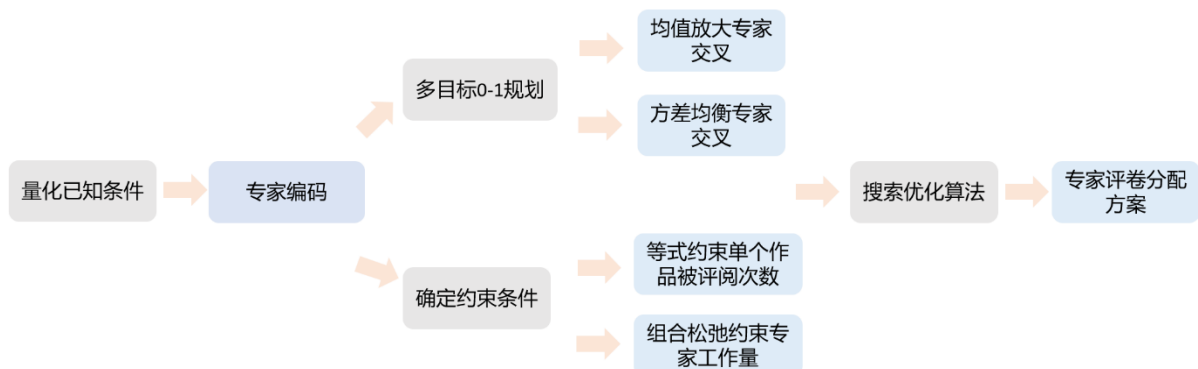


图 1 问题一流程图

3.2 模型建立与求解

3.2.1 指标定义

根据题目条件，我们知道，一共有 3000 支队伍和 125 位评审人员，那么我们将其转化为矩阵形式，方便求解，即：

$$V = \begin{bmatrix} v_{1,1} & \cdots & v_{1,n} \\ \vdots & \ddots & \vdots \\ v_{m,1} & \cdots & v_{m,n} \end{bmatrix} \quad (3.1)$$

其中 $i = \{1, 2, \dots, m\}, j = \{1, 2, \dots, n\}, m = 3000, n = 125$ 分别表示队伍和评委的数目， $v_{i,j} = \{0, 1\}$ 表示第 i 个作品是否被第 j 个专家评审。

由于每份作品由 5 位专家评审，所以一共需要评审 15000 次，为简化模型，我们假设每位专家的评审作品的数量是相同的，也就是 $15000/125=120$ 份。所以公式 3.1 需要做一定的约束，即：

$$w_i = \sum_{j=1}^n v_{i,j} = 5 \quad p_j = \sum_{i=1}^m v_{i,j} = 120 \quad (3.2)$$

其中 w_i 表示第 i 件作品的评委数量， p_j 表示第 j 位评委的评审作品数量。

3.2.2 相似性度量

由于我们需要度量不同评委专家的之间的交集“大小”，所以我们定义了相似性度量因子，即：

$$s_{i,j} = \frac{\mathbf{V}_i^T \odot \mathbf{V}_j}{p_j} \in \mathbb{R}, \quad \mathbf{V}_j = [v_{1,j}, \dots, v_{m,j}]^T \in \mathbb{R}^m \quad (3.3)$$

其中 $s_{i,j}$ 表示评委 i 和评委 j 之间的相似性 ($0 \leq s_{i,j} \leq 1$)， \odot 表示向量乘积操作， \mathbf{V}_j 表示第 j 位评委的评审分布向量

根据相似性度量因子，我们最终可以得出相似性度量矩阵，即：

$$S = \begin{bmatrix} s_{1,1} & \cdots & s_{1,n} \\ \vdots & \ddots & \vdots \\ s_{n,1} & \cdots & s_{n,n} \end{bmatrix} \quad (3.4)$$

由于矩阵 S 是对称阵，所以我们只取下三角的结果即可，并且我们不需要对角元的取值（自身与自身的相似性都是 1），所以我们真正需要的信息是

$$S_{\text{tril}} = \begin{bmatrix} s_{2,1} & & \\ \vdots & s_{3,2} & \\ s_{n,1} & \cdots & s_{n,n-1} \end{bmatrix} \quad (3.5)$$

3.2.3 多目标 0-1 规划求解

由式(3.5)得， S_{tril} 中的值表示任意两个评委的相似性。为了得到最优的“交叉分发”方案，我们需要考虑两个方面，一个方面是要保证所有的相似性度量因子都不能太小，也不能太大，需要在一个折中的范围内；另一个方面是，我们需要保证所有的相似性度量因子差距不能太大，这样才能保证分配方案的合理性。

所以基于上述直觉，我们建立了两个指标来量化上面所叙述的两个问题。

$$Z_{\text{mean}} = \frac{1}{n(n-1)/2} \sum_{s_{i,j} \in S_{\text{tril}}} s_{i,j} \quad (3.6)$$

$$Z_{\text{var}} = \frac{1}{n(n-1)/2} \sum_{s_{i,j} \in S_{\text{tril}}} (s_{i,j} - Z_{\text{mean}})^2 \quad (3.7)$$

自此，我们构造出了两个目标变量用于实现“交叉分发”的合理性和有效性。最后，结合(3.4), (3.6), (3.7)式，求解以下规划问题

$$\begin{aligned} & \text{Max } Z_{\text{mean}} - \lambda * Z_{\text{var}} \\ & s. t. \begin{cases} \mathbf{w}_i = 5 \quad i = 1, 2, \dots, m \\ \mathbf{p}_j = 120 \quad j = 1, 2, \dots, n \\ 0 \leq s_{i,j} \leq 1 \\ s_{i,j} = \frac{\mathbf{v}_i^T \odot \mathbf{V}_j}{p_j} \\ v_{i,j} \in \{0,1\} \end{cases} \end{aligned} \quad (3.8)$$

其中， $v_{i,j}$ 是 0-1 决策变量，为了防止 Z_{sum} , Z_{var} 量纲不统一，我们引入一个权衡因子 λ 来对目标变量 Z_{var} 进行缩放。

3.2.4 搜索优化算法

虽然决策变量仅为 0-1 变量，变量的个数高达 3000*125 个，对于这种高维问题直接求解并不现实，因此我们考虑采用分治法，对高维问题进行分组后分别进行优化。首先将 3000 份作品随机分为 n 个集合，同时将 125 名评委也分为 n 个集合，然后将 n 个作品集合和 n 个评委集合进行一一配对，配对完成后再对每一个集合对进行单独优化。

尝试使用启发式算法求解最优解时，由于我们将 0-1 矩阵中的每一个值都当作决策变量，指数爆炸问题使得无论如何改进启发式算法都需要超过 10T 的内存去运行。

所以为了解决该问题，我们提出在有限的可行域之下寻找最优解，简单来说，就是生成符合约束条件的矩阵 V 再通过有限次的迭代寻找最优解。该算法求解速度较快，并且随着迭代次数的上升，算法的精度也在逐渐提高，相当于寻找局部最优解。

由于是局部搜索，我们应当适当的放宽约束条件，即每个专家的评审作品数量在一定范围内，并且允许少数的专家评审作品数量超过范围。从而扩大可行域范围，提高局部最优解的精度。

所以我们需要对公式(3.2)进行修正

$$p_j = \sum_{i=1}^m v_{i,j} \quad 110 \leq p_j \leq 130 \quad (3.9)$$

我们考虑每位专家的评阅作品数量在 110~135 之间，进而公式 (3.3) 被修正为：

$$s_{i,j} = \frac{\mathbf{v}_i^T \odot \mathbf{V}_j}{\max(p_i, p_j)} \quad (3.10)$$

我们提出一个新的约束条件

$$o(P) = n - \sum_{j=1}^n I_{[110:135]}(p_j) \quad (3.11)$$

其中 I 表示示性函数, $P = [p_1, \dots, p_n]$

公式(3.9)和公式(3.10)为我们提出的新的约束, 原始的 p_j 要求等于 120, 我们认为该等式约束过于严格, 可行域较小, 所以进行修正, 并且 $o(P)$ 表示不在限制区间范围内的容忍度, 这里我们将其设置为 $[0.2n] = 25$, 所以, 我们将公式(3.8)修改为

$$\begin{aligned} & \text{Max } Z_{\text{mean}} - \lambda * Z_{\text{var}} \\ & s. t. \begin{cases} w_i = 5 \quad i = 1, 2, \dots, m \\ 110 \leq p_j \leq 135 \quad j = 1, 2, \dots, n \\ 0 \leq s_{i,j} \leq 1 \\ 0 \leq o(P) \leq [0.1n] \\ s_{i,j} = \frac{V_i^T \odot V_j}{\max(p_i, p_j)} \\ v_{i,j} \in \{0, 1\} \end{cases} \end{aligned} \quad (3.12)$$

求解模型, 我们最终选出最优方案的 $Z_{\text{mean}} = 0.03225, Z_{\text{var}} = 0.000254$, 部分求解结果如表 1 所示

表 1 参赛队-评委分配表

参赛队	专家一	专家二	专家三	专家四	专家五
0	4	62	93	107	111
1	14	60	62	73	76
2	6	84	100	112	122
3	45	82	109	114	124
4	8	55	56	85	111
5	1	2	17	37	42
6	7	36	48	89	91
7	39	40	54	68	107
8	43	47	61	71	80
9	17	34	60	65	73

四、问题 2 的模型建立与求解

4.1 问题分析

该问题要求我们分析每位专家、每份作品原始成绩、调整之后的分布特点, 挖掘数据信息, 然后选择不同的方案, 从而得出不同的成绩排序, 并据此分析这些方案的优劣, 之后, 根据方案的优劣性, 提出新的评审方案, 设计新的标准分计算模型, 并根据数据 1 中的现有权威排名, 优化模型。

所以第二问的解答需要分为以下步骤进行:

- (1) 描述性统计分析, 并利用问题 1 提出的指标, 探究专家评分的变化规律
- (2) 提出排名测度指标: Spearman 系数和 kendall 系数, 并根据已有多种评审方案, 得出排名, 使用指标量化, 比较优劣。
- (3) 在已知方案的基础上, 提出新方案, 建立排名模型, 并进行指标度量。根据已知数据, 提出震荡因子理论, 改进模型。
- (4) 根据已知数据, 针对 27 位一等奖的作品, 将新方案的得分与已知权威得分进行比较, 提取残差信息, 并根据已知统计量, 拟合函数关系, 进一步改进模型, 最终得出方案。

问题二的整体建模流程如图 2 所示

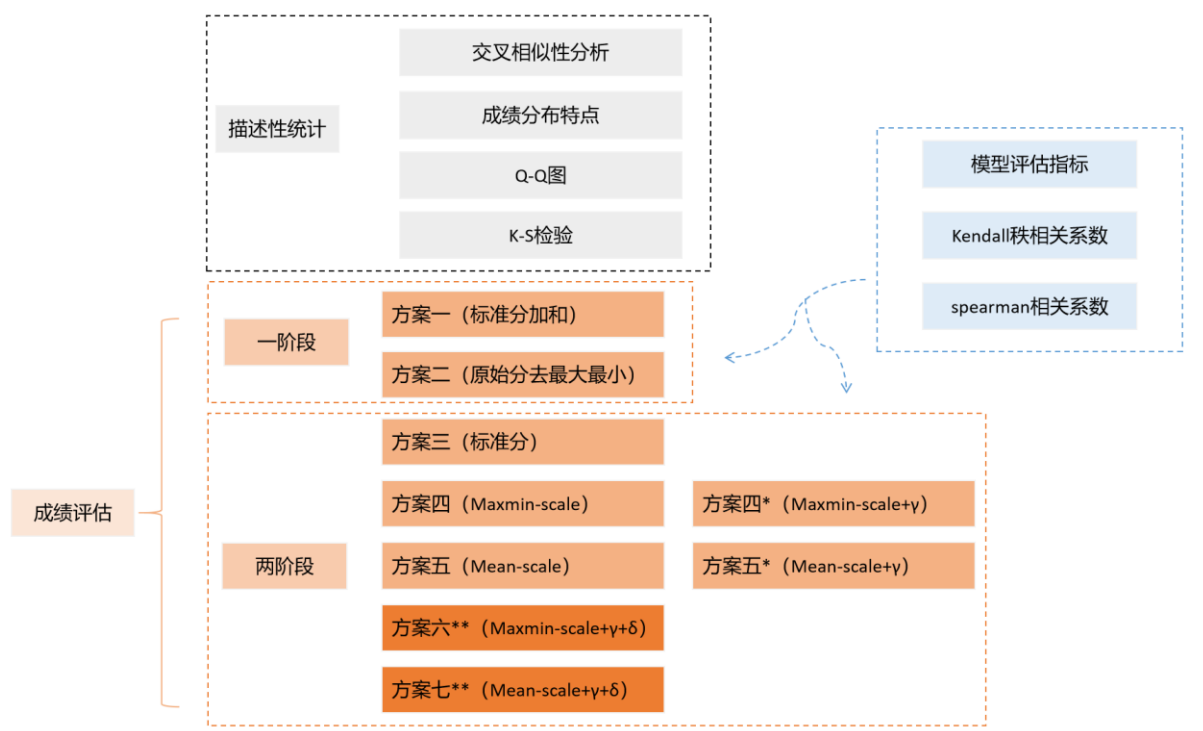


图 2 问题二流程图

4.2 模型建立与求解

4.2.1 描述性统计分析

(1) 专家评分分布分析

首先，我们需要分析各个专家的打分分布特点，根据数据 1，我们知道对于第一次评审的时候，每个作品由五个专家进行打分，所以我们将所有作品的所有专家编码提取出来，并经过计算得知，对于数据 1，总共有 97 位专家进行打分，并计算每个专家的评阅作品数量，并绘制直方图，如图 3 所示。

我们发现，专家的评阅数量大致在 102~104 之间，只有少数几位专家的数量超过该区间，但也在 105~109 之间。所以，我们认为，每位专家的评阅作品数量大致相同，这也印证了我们在问题 1 中作出的“每位专家评阅数量相同”的假设是合理的。

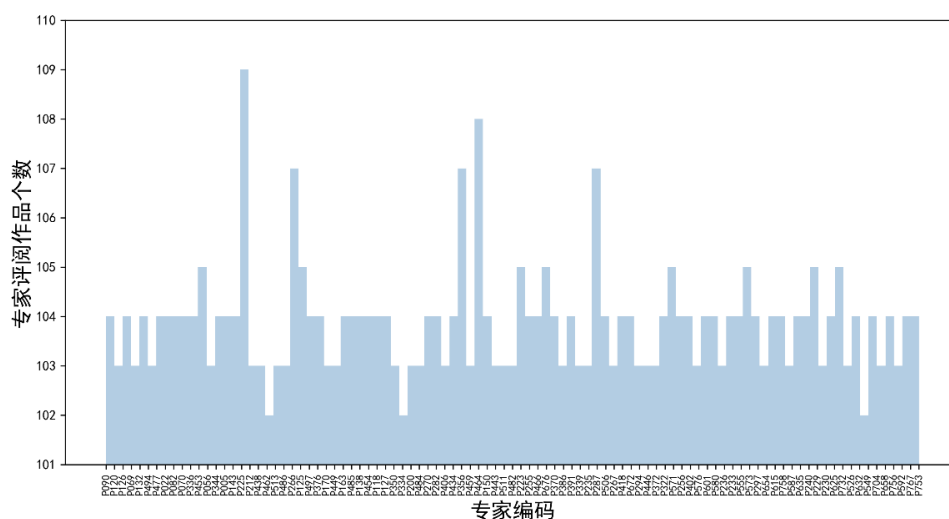


图 3 专家评阅数量直方图

由于每位专家评审的数量大都相同，所以我们可以绘制每位专家的打分分布，寻找分布特点。但是专家的分布不同可能是由于分配到的作品水平不同，或者是专家本身的打分倾向不同，所以，我们需要计算专家之间的相似性度量因子（式（3.3）），来判断分布的不同究竟是由哪种原因导致，或者是两个原因综合作用的结果。

所以我们需要计算相似性度量矩阵 S ，根据题目数据我们发现，一共有 2015 组参赛队，97 名评委，由于参赛队并无任何信息，所以我们将参赛队的排名作为其唯一标识，然后通过计算得出相似性度量矩阵 S ，由于该问题中的专家评阅数量不同，不符合我们之前问题 1 中的假设，所以，我们需要对公式(3.3)进行修正。

$$s_{i,j} = \frac{V_i \odot V_j}{S_{scale}} \in \mathbb{R} \quad S_{scale} = \text{Max}(\text{sum}(V_i), \text{sum}(V_j)) \quad (4.1)$$

我们绘制出了所有专家的相似性度量矩阵，如图 4 所示。

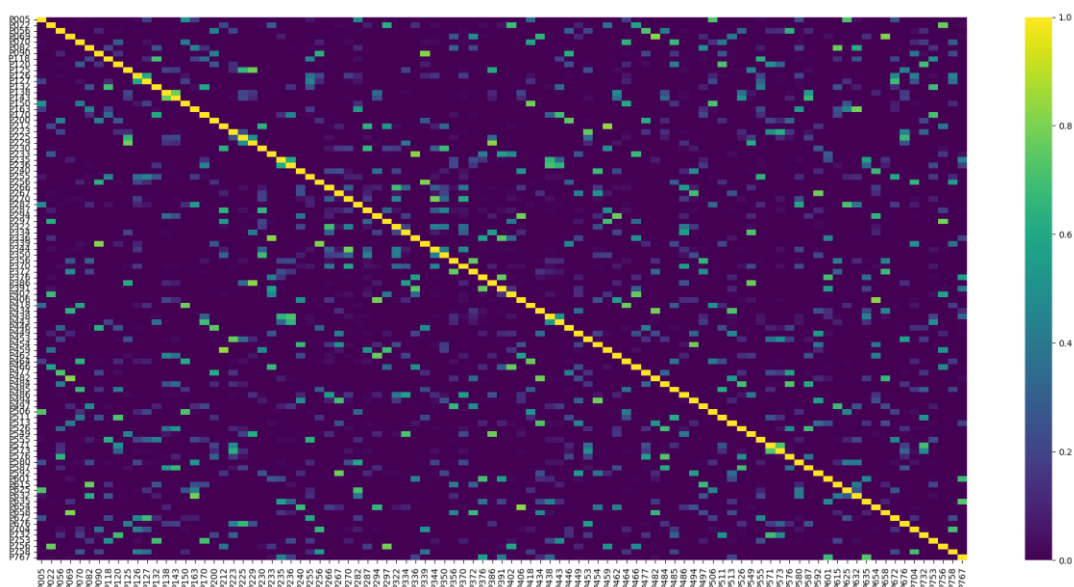


图 4 97 位专家相似性矩阵

我们进一步绘制了度量因子的频数分布，如图 5 所示

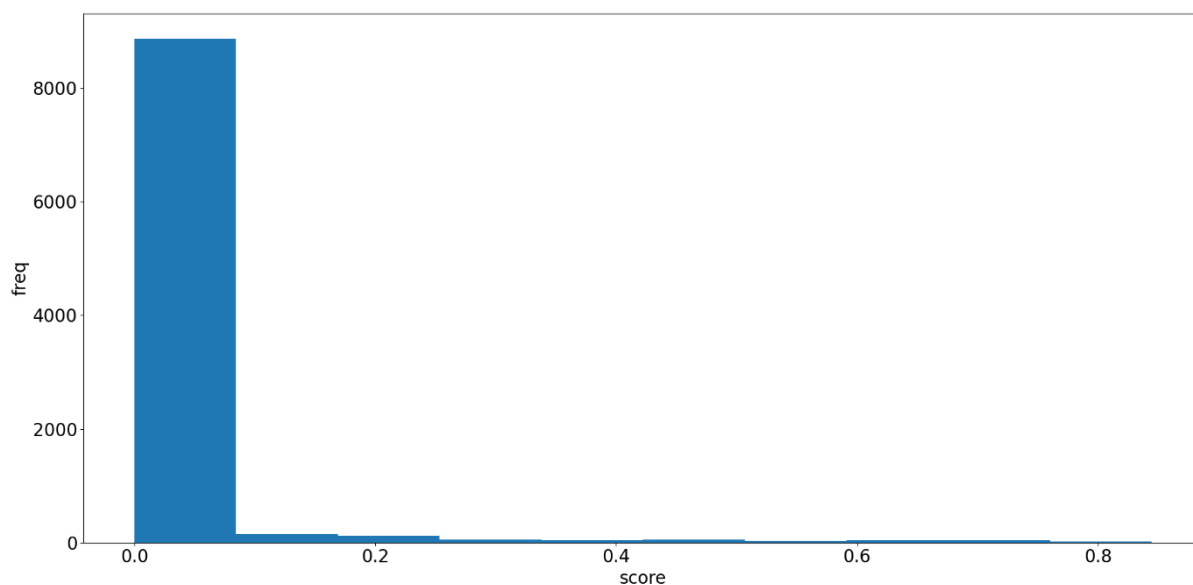


图 5 度量因子频数图

我们发现，该度量矩阵实际上是稀疏的，大多数的度量因子都集中在 0~0.1 之间，并且出现了题目 1 中提到的问题，即有的专家相似性高了，一定会有有的专家相似性变低。为了方便分析，对于专家分布部分我们仅对 97 位专家进行抽样，选择其中的 8 位进行分析，选取的依据是在相似性度量矩阵中选择相似性最大的 4 组专家（共 8 人），然后选择这 8 位专家进行分析。

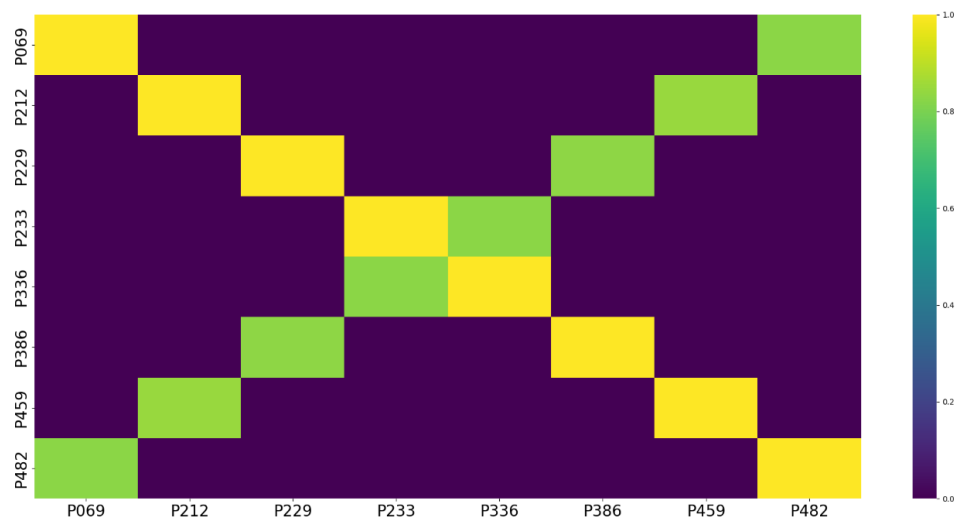


图 6 Top-8 专家相似性矩阵

如我们可以发现，相似性最高的这 4 组数据中，每组数据仅仅与自身组的专家高度相关，但是于其余组的相关性为 0，这也印证了高度相关的组，往往与其余的专家不相关。具体的组别和相似性度量因子在表 2 中给出。

表 2 Top-8 度量组别

相似性组类	相似性度量因子大小（0~1）
P459 P212	0.84466
P386 P229	0.82857
P336 P233	0.82692
P482 P069	0.82524

以上结果均保留了 5 位小数

为了探究影响影响成绩分布的因素，我们选择表 2 中的所有专家，分别绘制其评阅作品的成绩分布。

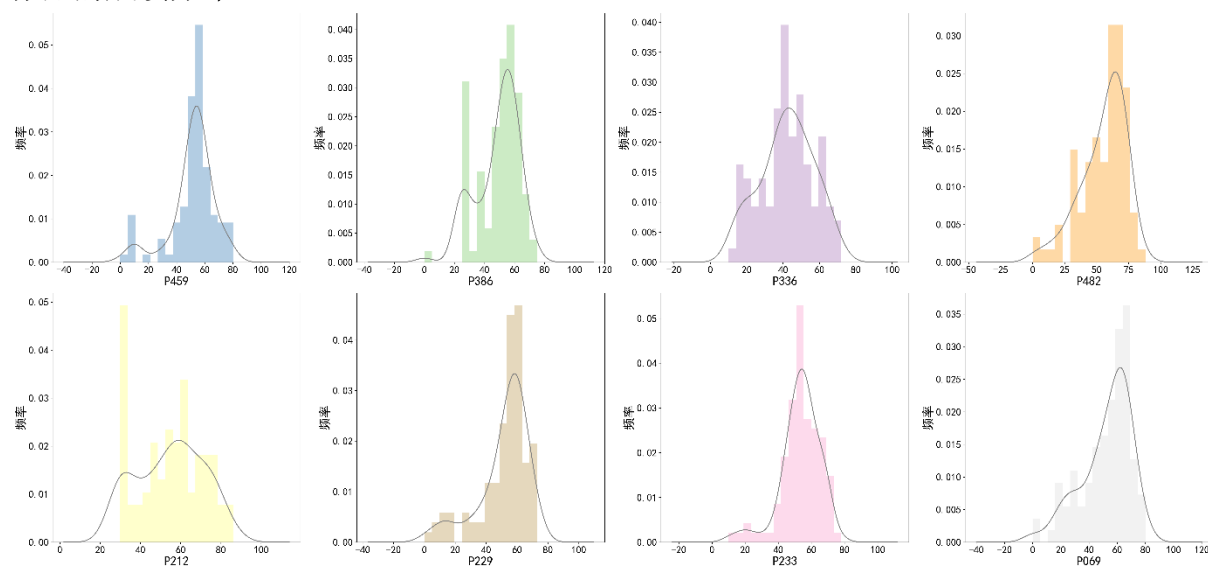


图 7 Top-8 专家原始分布图

如图 7 所示，每一列表示表示同一组类（相似）的专家，一共有 4 组，并且根据图 6，我们已知，任意组间专家的度量因子为 0。可以发现，从宏观来看，这几位专家的分布都大不相同，并且如果以 50 为均值的话，大多数专家的打分都是左偏的，即倾向于打较高的得分，当然这有可能是因为作品水平分布不同所致。

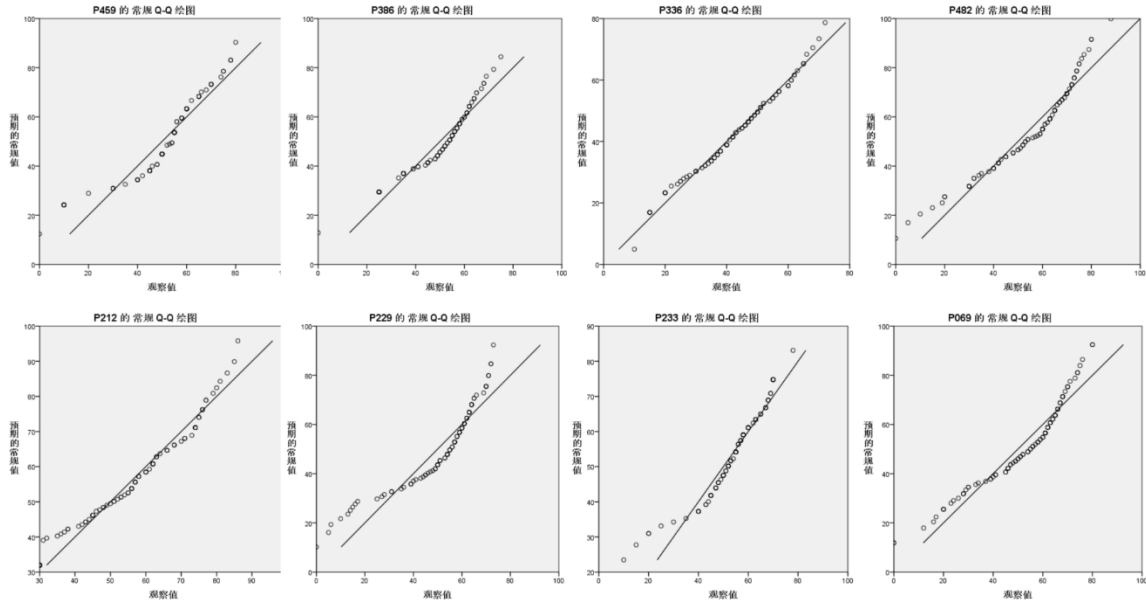


图 8 Top-8 专家原始分分布 QQ 图

QQ 图是检验样本是否服从正态分布的常规手段，简单来说，若图中的散点在直线上的话，则表示样本是服从正态分布的，所以我们可以发现，对于上述的 8 位专家我们发现，只有编号为“P336”的专家的评阅作品分布是近似服从正态分布的，其余的专家均不服从正态分布。

从组内（作品分布大致相同）来看，仅有最后一组的打分分布相似，其余几组的打分分布均不相同，为了合理印证我们的想法，确保严谨性，我们使用 KS-检验（Kolmogorov-Smirnov test）来检验同一组的专家的打分分布是否一致。

KS 检验^{[5][6]}:

在统计学中，Kolmogorov - Smirnov 检验（亦称 K-S 检验或 KS 检验）是一种连续的一维概率分布均等性的非参数检验，可用于比较一个样本与一个参考概率分布（例如：正态分布），或比较两个样本（两个样本的 K-S 检验）。对于该问题，我们使用的是两样本的 KS 检验。

两样本 KS 检验的统计量表示为

$$D_{n,m} = \sup_x |F_{1,n}(x) - F_{2,m}(x)| \quad (4.2)$$

其中 $F_{1,n}(x)$ 和 $F_{2,m}(x)$ 分别是第一样本和第二样本的经验分布函数，而 \sup 是上确界函数。 $F_n(x)$ 的表达式被定义为

$$F_n(x) = \frac{1}{n} \sum_{i=1}^n I_{[-\infty, x]}(X_i) \quad (4.3)$$

其中 $I_A(X_i)$ 为示性函数，表示当 $X_i \in A$ 时等于 1，否则等于 0

KS 检验的原假设为两个样本的分布相同，备择假设为两个样本的分布不同，对于大样本（30 以上）如果满足以下条件，则原假设在显著性水平 α 之下被拒绝。

$$D_{n,m} = c(\alpha) \sqrt{\frac{n+m}{n * m}} \quad (4.4)$$

其中 n, m 为第一个样本和第二个样本的大小，在本问题，显著性水平 α 取 0.05，经过查表可得 $c(\alpha) = 1.358$ 。

表 3 KS 检验结果

两样本 Kolmogorov-Smirnov 检验			
专家一	专家二	Kolmogorov-Smirnov Z	渐近显著性(双尾)
P459	P212	1.610	0.011
P386	P229	1.190	0.118
P336	P233	3.011	0.000
P482	P069	0.910	0.379

检验结果如表 3 所示，可以发现，如果在 α 值为 0.05 的假定下，第一组和第三组的 P 值分别为 0.011, 0.000 小于 0.05，拒绝原假设，认为二者并不服从同一分布，同理第二组和第四组服从同一分布，这也就意味着不同的老师的打分倾向虽有差异，但也有相同，所以我们可以根据老师的打分信息，将其划分为不同倾向的（例如：低分、平均、高分）的分组内。

同样的，我们也绘制了标准分的专家分布图，如图 9 所示

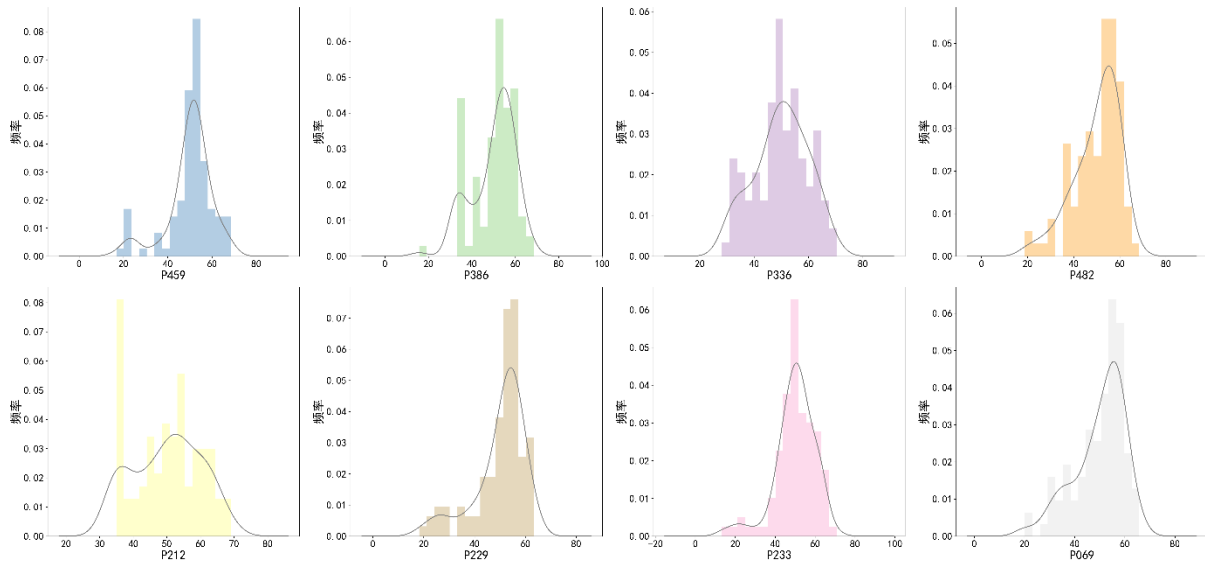


图 9 Top-8 专家标准分分布图

我们发现，除了频率尺度发生变化之外，原始分和标准分的分布几乎一致，我们知道标准分的计算公式为

$$x_k = 50 + 10 \times \frac{a_k - \bar{a}}{s} \quad (4.5)$$

其中 $\bar{a} = \frac{1}{n} \sum_{k=1}^n a_k$ ，表示专家给出成绩的均值， $s = \sqrt{\frac{1}{n-1} \sum_{k=1}^n (a_k - \bar{a})^2}$ 表示样本标准差

直观上来看，引入标准分的意义在于将评委的打分区间规范到 40-60 之间，但是这是基于样本 a_k 服从正态分布的前提下（上文已经说明样本并不服从正态分布），并且该方案根本没有考虑到专家的不同打分倾向。从结果来看，图 7 和图 9 的分布差别不大，同样是有偏的，并且得分得区间也没有变动，这说明标准分的方案效果是不佳的。

由于我们现已知专家的打分倾向是有所不同的，现我们需要进一步探究所有专家的倾向分布，结果如图 10 所示。

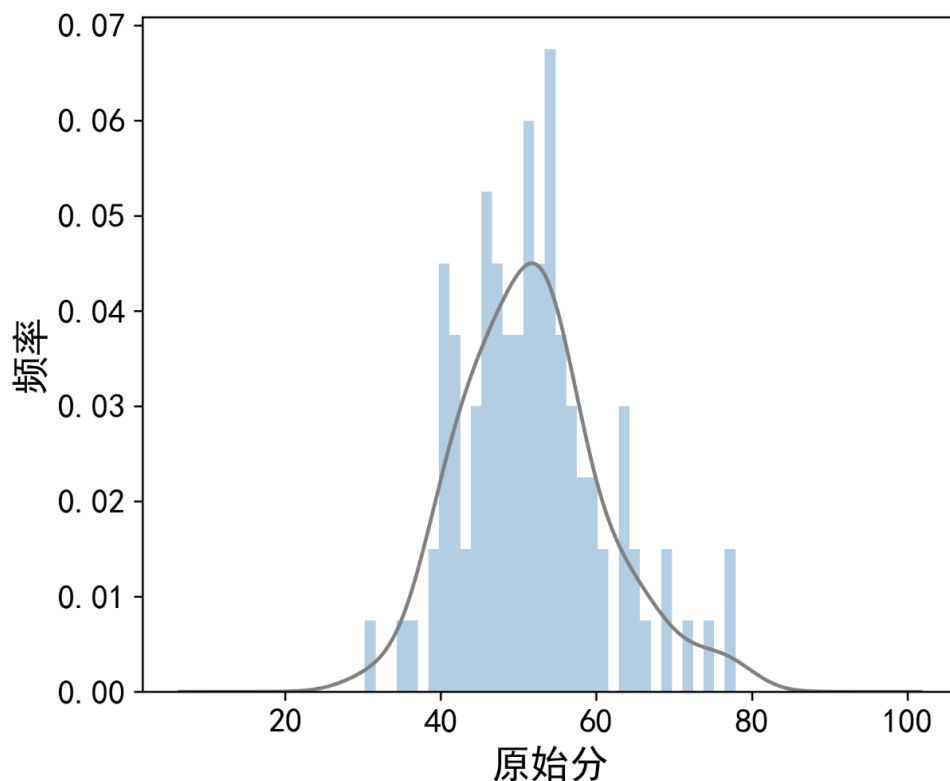


图 10 专家打分均值直方图

根据上图，我们发现，尽管专家的打分倾向不一致，但是对于所有的专家而言，打分大多数都在 40-60 之间，少数专家打分倾向分布在两侧（两侧）。

综上所述，我们得出结论，不同专家的评阅作品得分分布有所不同，尽管在作品大多数都一致的情况下。这也就意味着专家的打分倾向具有异质性，但是大多数的专家倾向都在平均分（50）之间，少数的在两侧。

（2）作品成绩分布分析

我们需要分析所有作品的原始成绩以及调整之后的成绩分布，这里我们先分析题目中已知的两种方案，即方案一：取标准分之后求和；方案二：去除最低最高分后求和。

我们将总计 2015 个参赛队的成绩分布绘制成直方图，结果如图 11 所示

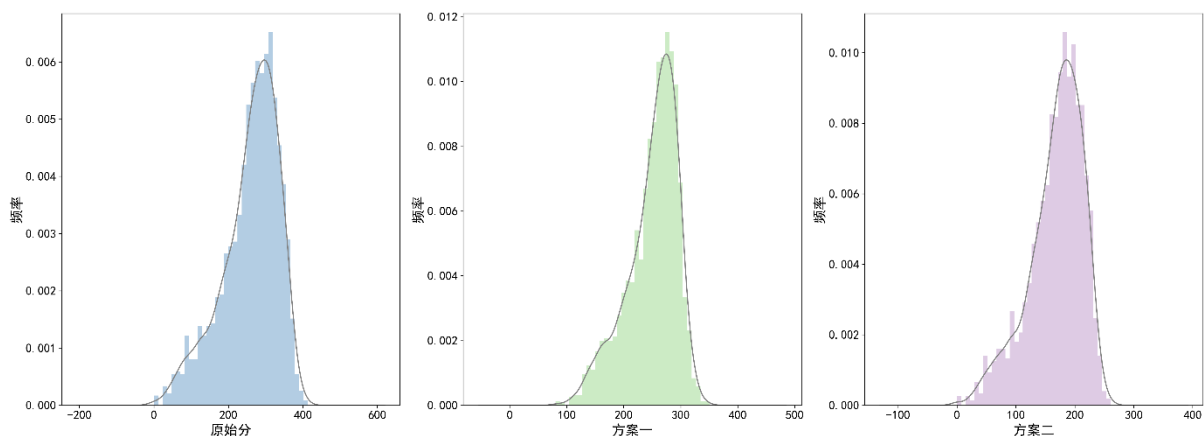


图 11 不同方案下的作品成绩分布图

我们发现三者的分布大致相同，也就是单峰的左偏分布，即两个方案在总的评分分布上并无显著差异，并且对于上述三个分布，我们发现，虽然参赛队的水平不同，但是得分还是与正态分布相似，也就是说参赛队的水平分布可以近似认为是存在一定的左偏的正态分布，综合之前的分析，我们认为大部分评委的打分是合理的，剩余评委的打分倾向中有较多人倾向于打高分，使得作品的综合成绩被拉高，从数据中也可以体现，我们发现图 11 中原始分之和大概在 300 分左右，由于上述的结果都是基于一阶段中得出，所以作品的打分人数为 5 人，即平均分为 60 分，显然是存在一定数量的高打分倾向的评委，将平均分拉高。

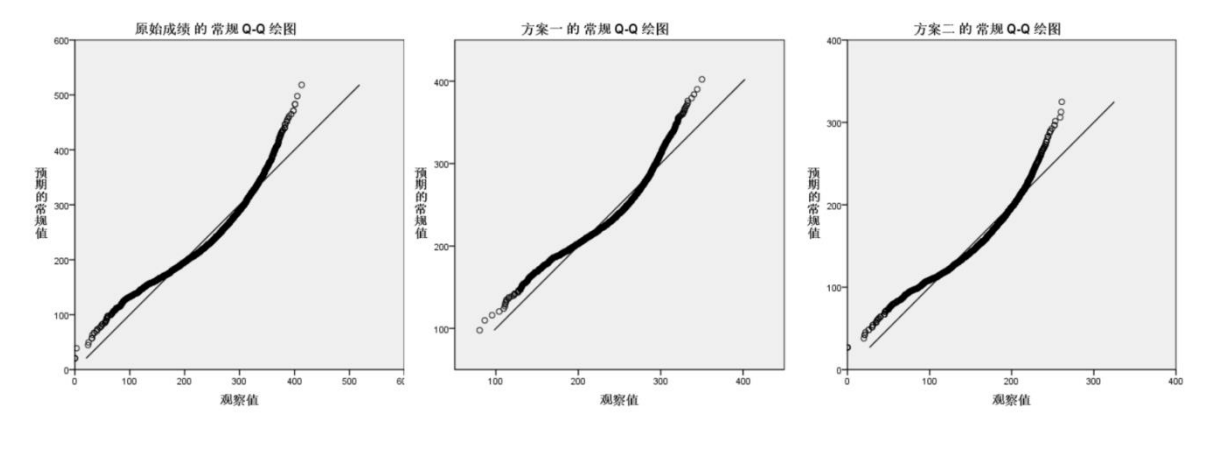


图 12 不同方案下的作品成绩分布 QQ 图

经过检验，我们得知，原始分、方案一、方案二的所有作品的得分分布均不服从正态分布，尾部数据的偏离实际上也就表示了高分作品的数量偏多。

4.2.2 原始评审方案分析

由于题目要求我们按照不同的评审方案进行排序，现在我们仅考虑上一小节中提出的方案一与方案二，现我们需要对排名的好坏进行评估，查阅相关资料，我们考虑将肯德尔相关性检验的方法作为理论基础，用于衡量评审方案的优劣。

由题目的已知条件，我们知道进入二阶段的人员是由一阶段的排名决定的。为了保证排名能够比较，我们需要保证在基于现有指标或者新指标的情况下，进入二阶段的人员不发生变动，所以，为了简化模型，我们需要做出以下假设，即二阶段的人员不发生变动。

所以，我们分为两个部分进行比较，一是比较进入二阶段的所有作品，二是比较所有只进行一阶段的所有作品。我们根据题中给出的计算公式，得出以下三个现有的方案。

方案一：不同评委有对应的不同的打分集合，对集合中的每一个原始分依据偏差值法转换为标准分，然后将标准分相加得每个作品总分，然后依总分排序。

方案二：是去掉同一份作品得到的五个原始分中的最高分、最低分，再将剩余评分相加依总分排序。

方案三：不同评委有对应的不同的打分集合，对集合中的每一个原始分依据偏差值法转换为标准分，只是这一次结合两阶段的评分即用第一阶段五个标准分的均值加上第二阶段的三个标准分，得到各个作品的总分后进行排序。这个方案剔除了专家复议的影响，为后续的标准分计算模型对比提供参照。

事实上，方案三和方案一的区别仅仅在于是否进入两阶段，计算公式都是使用标

准分计算公式，方案二仅仅适用于一阶段。由于我们需要比较方案的排序的好坏，所以我们可以使用秩检验方式作为评估指标。所以，得到不同计算模型的排序结果后，利用 Spearman 相关系数和 kendall 秩相关系数比较不同方案的优劣性。

spearman 相关系数:

spearman 相关系数是统计学中常用的用于衡量两个样本相关性的一个指标，它是依据两列成对等级的各对等级数之差来进行计算的，所以又称为“等级差数法”，而对于本问题中，这里的等级也就是排名，所以我们可以使用该系数来检验我们现有方案的排名优劣。斯皮尔曼相关系数被定义成等级变量之间的皮尔逊相关系数。对于样本容量为 n 的样本， n 个原始数据被转换成等级数据，相关系数 ρ 被定义为

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}} \quad (4.6)$$

但是在本问题中我们使用另外一个计算公式:

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2 - 1)} \quad (4.7)$$

其中， d_i 表示第 i 对样本的等级之差， n 表示样本数量。

Kendall 秩相关系数:

在统计学中，kendall 秩相关系数评估了给定相同对象集的两组秩之间的相似程度，是用于测量两组数据之间序列关联的统计量。该系数依赖于将一个秩次顺序转换为另一个秩次顺序所需的逆序对的数量。kendall 系数有两个计算公式，一个是 Tau-a，另一个是 Tau-b。两者的区别是 Tau-b 可以处理有相同值的情况，即并列。Tau-a 的计算公式如下所示

$$\text{Tau-a} = \frac{c - d}{\frac{1}{2}n(n - 1)} \quad (4.8)$$

其中， c 表示一致对数， d 表示分歧对数。 $\frac{1}{2}n(n - 1)$ 表示所有样本两两组合的数量，当没有重复值时，组合数量等于 $c + d$ 。Tau-a 的计算中假定原始数据中不存在并列排位。当原始数据中存在并列排位时，则应该使用 Tau-b，其计算公式为:

$$\text{Tau-b} = \frac{c - d}{\sqrt{(c + d + t_x)(c + d + t_y)}} \quad (4.9)$$

其中 c 和 d 则分别代表一致对和分歧对的个数， t_x 和 t_y 则分别表示数据 X 中的并列排位个数，和数据 Y 中的并列排位个数。如果是同时发生在 X 和 Y 中并列排位，则既不计入 t_x ，也不计入 t_y 。由于存在并列排位，故本文中使用 Tau-b 作为评价指标。

以经过两阶段评分专家协商后的排序结果为标准排序结果，得到不同排序与其他的 kendall 秩相关系数和斯皮尔曼系数。具体数值如表 4 所示:

表 4 不同方案的系数表

阶段	方案	spearman	P 值	Tau_b	P 值
一阶段	方案一	0.42857	0.02572	0.32194	0.01850
	方案二	0.21119	0.29032	0.13814	0.31619
二阶段	方案三	0.71429	0.00003	0.58974	0.00000

上述结果均保留 5 位小数

需要注意的是，由于题目已知条件，我们知道，一等奖作品排序是评委协商之后认定的，所以我们把这前 27 个排名作为基准，来对所有的方案进行评估。

方案二直接将原始分去掉极值后取平均，虽然这种方法可以提高数据集的抗噪能力避免异常值对最终结果的不利影响，但它对消弱不同评委间的评分差异性没有帮助，从表中数据可以看到其 spearman 相关系数和 kendall 系数都很小，仅有 0.21119 和 0.13814，并且 P 值都大于 0.05，不拒绝原假设，认为这两个系数的结果都不是很显著。

方案一是基于偏差值法对作品进行排序，这种方法可以比较和分析不同数据点相对于某个基准值的差异，在每组样本量足够大时可以有比较好的统计学比较。可以看到相比较方案二有了较大提升，但仍没有方案三优秀，这也是因为最终的排名是在方案三的基础上再经过调整得到的。

方案三不同于其他方案，考虑了两个阶段的原始分，基于题中所给标准分计算方式即以各个评委的打分集合为依据通过偏差值法得到标准分，只是剔除了二阶段的专家复议分对排序结果的影响，因此它可以作为一个与题中所给标准分计算方式进行对比的参照。

我们可以进一步分析上述方案在其余作品集合上的排名情况。

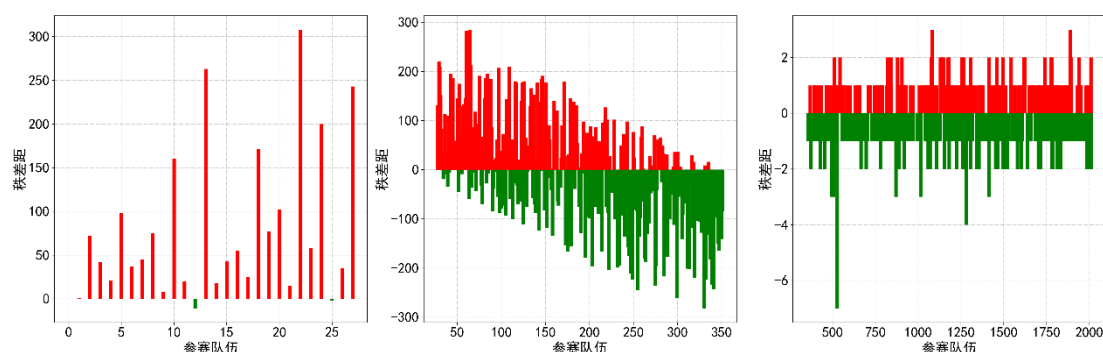


图 13 方案一秩差图

上图为分别前 27 支参赛队（一等奖），前 352 支进入二阶段的参赛队，以及未进入二阶段的队伍的秩差图，由于未进入二阶段参赛队的排名是在方案一的基础上对参赛队成绩求均值得出的，也就意味着没有秩差，但是实际上的秩差图都有不小的偏差。于是，我们观察数据 1，选取了其中 4 支队伍的成绩作为修正示例。

表 5 数据 1 排名修正

成绩	排名	修正后排名
271.4	756	757
271.42	757	756
271.39	758	759
271.4	759	757

可以发现，与文中提出的排名方式不符，文中是对标准分进行求和后按成绩进行排名，而对于上述数据中，我们发现，数据中不仅存在相同数据（这也是我们选择使用Tau-b的原因），还存在着排名错乱的问题，所以我们统一将排名修改为秩计算方式，使用“Min”秩计算方式，也就是在遇到相同的数据时，排名同取较低的秩（较高的排名）正如上述的修正后排名所示。

对排名进行修正后，重新计算结果，如图 14 所示

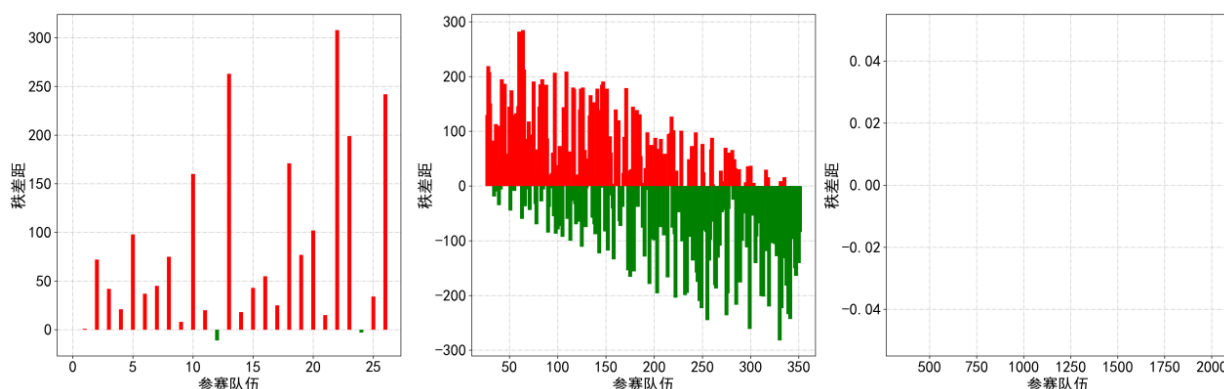


图 14 修正后方案一秩差图

我们发现，对于权威排名，方案一的秩差为正（红线），这也就意味着方案一低估了排名，也就是说单纯使用 5 个专家的评分加总，存在一定的误差，并且往往会低估高质量的作品，这可能是由于使用成绩求和的方式，很大程度上取决于专家的评分倾向，方差较大，所以有许多“高分”队伍出现。

其次，我们观察二阶段秩差图，发现排名波动很大，无明显趋势，方案一的效果不佳，并且由于数据修正，第三张秩差图并无出现秩差。

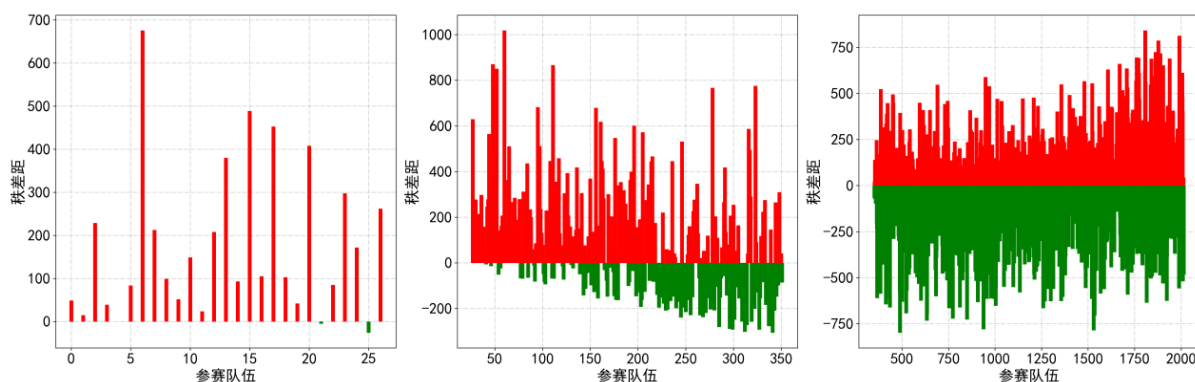


图 15 方案二秩差图

我们发现，第一张图与方案一相似，即低估参赛队水平，并且观察第二张图我们发现，与方案一不同的地方在于，方案二似乎普遍低估参赛队水平，并且两张图的秩差最高达到 600 以及 1000，所以认为，单纯使用方案二，即去除最高最低分的方法是很差的，不如方案一。

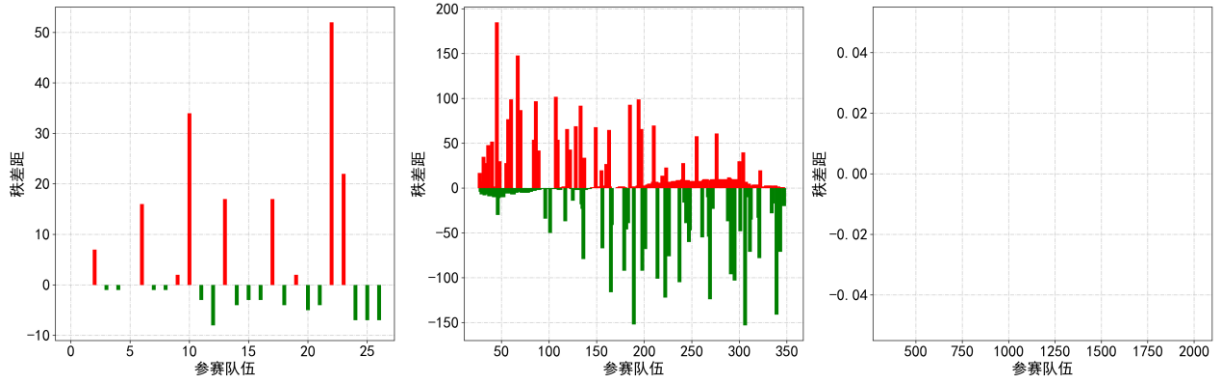


图 16 方案三秩差图

我们发现，经过两阶段评比的方案三效果有了显著的提升，对于部分参赛队的排名预测无误，并且秩差值相比于上两个方案都小很多。

不难看出以上提出的方案都无法有效体现出作品的真实水平，有研究表明在大多数的主观评分中，将近有 50% 的误差来自评分员之间的不一致性，结果造成评分员间的信度很低。这样就使评分员的评分和被试的真分数之间产生了差距，能力相同的被试也许得到了不同的分数，相同的分数也许并不代表相同的能力。

综上所述，我们认为使用去除最高和最低分的方案二效果是最差的，使用文中提出的标准分的方案较好，并且使用二阶段优于单纯使用一阶段的方式，所以基于上述分析，我们提出一种基于原始分的“标准化”方式，并且沿用题中的二阶段进行方案设计。

4.2.3 模型提出评估

原题目中的提出的标准分计算公式如（4.10）式所示，这种计算方式实际上就是基于机器学习中的 Z-score 标准化提出的，但是使用这个公式需要基于样本的正态假定之下，这样才能使得缩放后的样本服从 $N(50, 10^2)$ ，为了打破正态假定的限制，我们提出以下标准分计算公式。

$$\text{标准分: } x_{k,i} = 50 + 10 \times \sigma(a_{k,i}) \quad (4.10)$$

$x_{k,i}$ 表示第 k 个作品的第 i 个评委的打分，其中 $\sigma(a_{k,i})$ 为缩放方式， $x_{k,i}$ 原题目中为 $\frac{a_{k,i} - \bar{a}}{s}$ 。我们提出以下两个新的缩放方式。

均值标准化：

$$\text{Mean-scale}(a_{k,i}) = \frac{a_{k,i} - \bar{a}_i}{\max(a_i) - \min(a_i)} \quad (4.11)$$

其中， $\bar{a} = \frac{1}{n} \sum_{k=1}^n a_{k,i}$ ， a_k 表示某位专家给出的第 k 个成绩，该方法可以将 a_k 缩放到 $[-1, 1]$ 之间，并且不用考虑作品分布的问题。

最大最小标准化：

$$\text{Maxmin-scale}(a_{k,i}) = \frac{a_{k,i} - \min(a_i)}{\max(a_i) - \min(a_i)} \quad (4.12)$$

该方法可以将 $a_{k,i}$ 缩放到 $[0, 1]$ 之间，并且同样不用考虑作品分布的问题。

我们将 $\sigma(a_k)$ 替换为Mean-scale($a_{k,i}$)和Maxmin-scale(a_k)输入模型，计算结果如表 6 所示。

表 6 新方案系数表

阶段	方案	spearman	P 值	Tau_b	P 值
两阶段	方案四 (Maxmin-scale)	0.20024	0.31661	0.11681	0.40798
	方案五 (Mean-scale)	0.18664	0.34161	0.11640	0.39907

我们发现单纯将缩放方式进行替换，效果极差，4.2.1 节中，指出不同评委的打分倾向不同，一份作品在观点不同的评委手上，打的分数自然不同，所以我们需要对公式(4.10)进行修正，需要将评委的打分倾向考虑在内。

震荡因子：

我们的标准分计算公式中的 50 为所有老师提供了一个相同的打分基准，但单纯地加入基准分无法平衡打分的震荡区间。从各个老师的打分分布图中可以看到，所有老师都倾向于打高分，因此高分区间会密集地存在很多次优等作品。

但不同老师打分的震荡区间不一样，例如有的老师认为最优等和次优等作品间应该相隔 4 分，而有的老师认为应该相隔 8 分。为了消除震荡区间对最终排名的影响，我们在标准分计算公式中引入了震荡因子，当某位老师的平均分越高，说明最优等成绩和次优等成绩之间的间隔就越小，此时震荡因子变小就可以在一定程度上降低次优等成绩，这种变化就等价于增大打分的震荡区间。

因此通过加入震荡因子调整各个老师的打分震荡区间可以显著地提高标准分计算模型的可信性。震荡因子的计算公式为

$$\gamma_i = \frac{A}{\frac{1}{n_i} \sum_k a_{k,i}} \quad (4.13)$$

其中 $A = \frac{1}{ki} \sum_{k,i} a_{k,i}$ ，表示所有参赛队得分的均值， n_i 为第*i*位评委评阅的作品数量。基于震荡因子，我们将公式(4.10)进行修正。

$$\text{标准分: } x_{k,i} = 50 + 10 \times \sigma(a_{k,i}) \cdot \gamma_i \quad (4.14)$$

我们将公式(4.12)输入模型进行计算，结果如表 7 所示

表 7 方案改进系数表

阶段	方案	spearman	P 值	Tau_b	P 值
两阶段	方案四* (Maxmin-scale+ γ)	0.69536	0.00006	0.54416	0.00001
	方案五* (Mean-scale+ γ)	0.67705	0.00011	0.51567	0.00009

我们发现，经过震荡因子修正之后，我们模型的效果得到了显著的提升，在 spearman 系数上，方案四和方案五的提升高达 247%和 262%，在 kendall 系数上，提升高达 365%以及 343%。

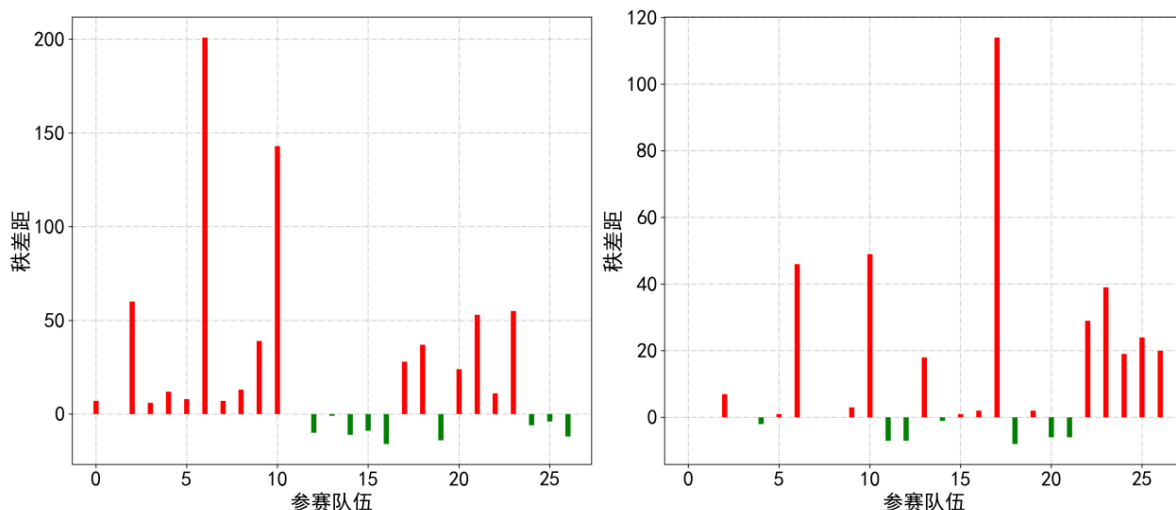


图 17 方案四改进前后秩差图对比 (left:改进前, right:改进后)

如上图所示,我们可以发现,秩差得到的显著的提升,最大秩差的样本(序号为 6 的样本)秩差从 200 提升到了 50 以内,序号为 10 的样本从 150 左右提升到了 50 以内,这表明,震荡因子的引入对于处理极端值是有效的。

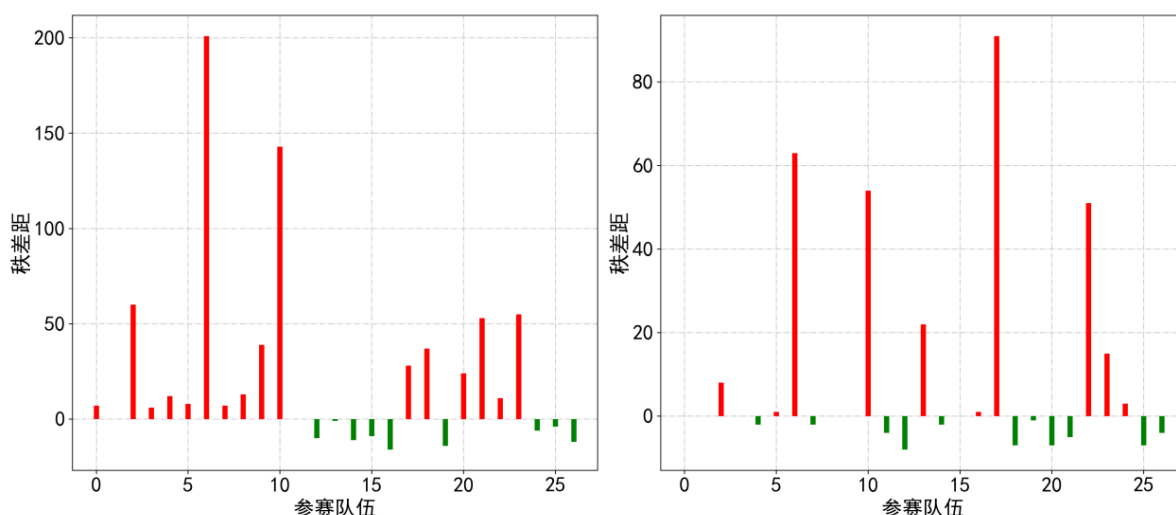


图 18 方案五改进前后秩差图对比 (left:改进前, right:改进后)

我们发现,对于方案五同样是序号为 6 和 10 的极端样本得到了显著的改善,但是我们发现无论是方案四还是方案五改进之后,序号为 17 的样本秩差都显著变大了,为了探究原因,我们找到了该样本,发现该样本是所有 27 个一等奖作品中第二次评审分标准差极差最大的作品,高达 36.21,所以我们需要根据题目所给数据在考虑极差的情况下进一步改进模型。

4.2.4 改进模型

虽然加入了震荡因子,使得方案四和方案五的效果得到了显著的提升,但是并没有超过原有的标准分方案,我们认为,这实际上是合理的,这是因为我们作为权威排名比较的 27 支一等奖队伍的成绩,本身就是基于标准分计算公式综合复议成绩,然后按最终成绩排名的。所以我们需要从数据入手来优化模型,由于我们并不知道复议的成绩如何计算,此时我们仅仅知道原始分、标准分、极差等信息。由于是否复议取决

于二阶段的标准分极差，所以我们需要从该方面入手，探求函数关系。

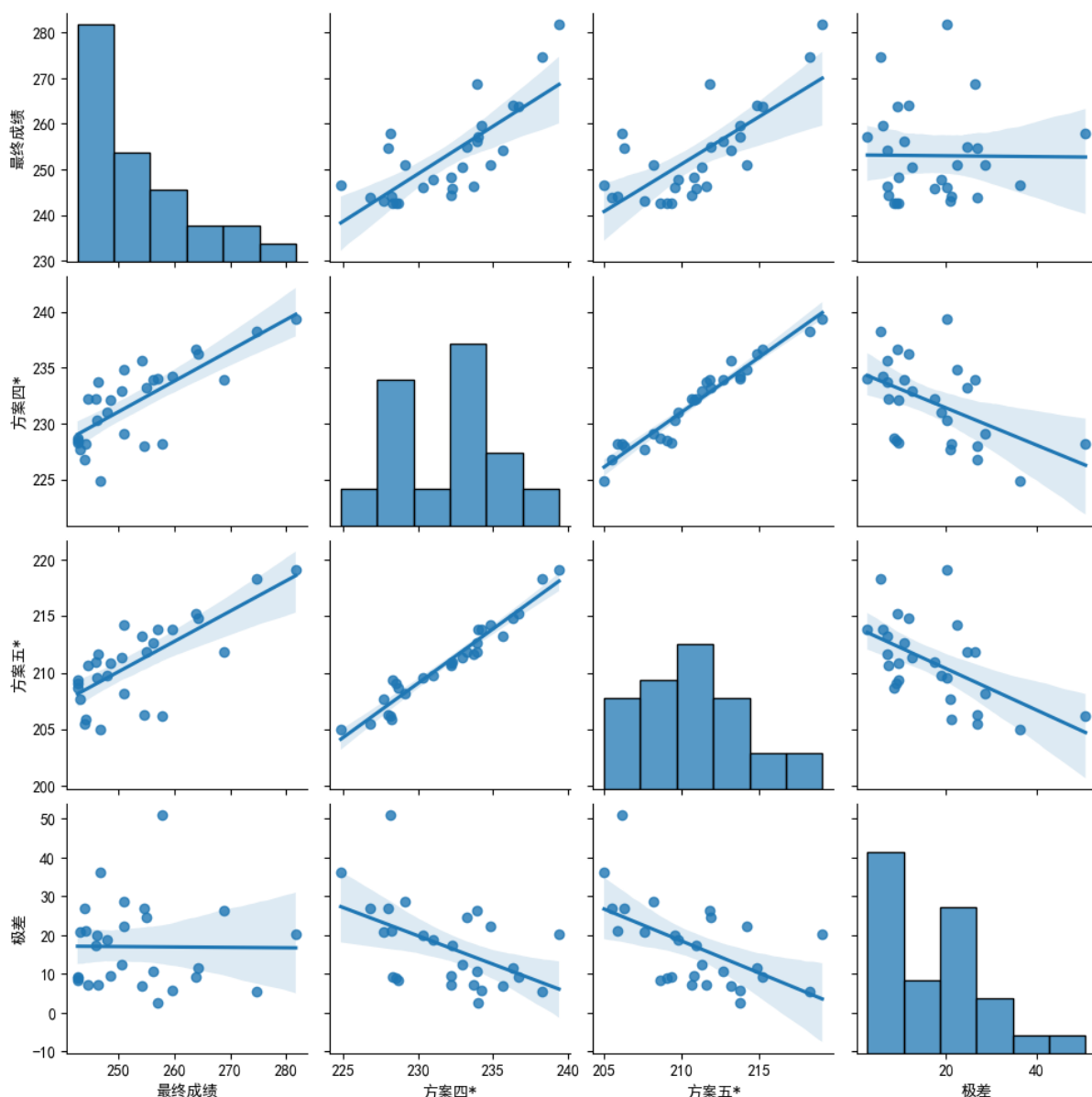


图 19 散点图矩阵

图 19 为最终成绩与方案四成绩、方案五成绩，以及极差之间的散点关系矩阵，我们发现变量之间存在一定的线性关系，但是由于方案四和方案五的共线性较强，所以我们考虑对两个方案分别建立模型。

我们认为，排名成绩是按照得分进行排序，所以我们需要找到新体系之下的得分和权威得分之间的差距，所以我们引入偏差因子 δ ：

$$y_k = f(\bar{y}_k, \delta_k) + \varepsilon_i \quad (4.15)$$

其中 δ_k 表示第 k 支参赛队的偏差因子， $k = \{1, 2, \dots, n\}$, $n = 27$ 表示具有权威排名的 27 支参赛队， $\varepsilon_i \sim N(0, 1)$, \bar{y}_k 表示通过我们的方案计算出的最终得分， y_k 表示数据中给出的最终得分。

我们认为该偏差因子与评委的组内异质性导致的评分不均衡有关，即

$$\delta_k = g(x_{k,6}, x_{k,7}, x_{k,8}) = g(\text{Max}\{x_{k,6}, x_{k,7}, x_{k,8}\} - \min\{x_{k,6}, x_{k,7}, x_{k,8}\}) \quad (4.16)$$

其中 $x_{k,6}$ 表示第 k 个作品第6位评委打的标准分（也就是二阶段的第一位评委）
我们使用多元线性回归拟合模型，结果如下所示

表 8 拟合优度检验

R^2	调整后的 R^2	Durbin-Watson(U)
0.733	0.71	1.303

由表 8 可知， R^2 为 0.733，表示拟合结果较好。

表 9 方案四方差分析表

	平方和	自由度	均方	F	显著性
回归	1981.321	2	990.661	32.893	0.000***
残差	722.83	24	30.118		
总计	2704.151	26			

* $p < 0.1$; ** $p < 0.05$; *** $p < 0.01$

拟合优度为 0.733，方差分析的 P 值为 0.000，在显著性水平为 0.05 的情况下，拒绝原假设，认为自变量和因变量之间存在显著关系，并且拟合优度为 0.733，拟合效果较好。

表 10 方案四系数显著性检验表

	非标准化系数		标准系数	t	显著性	95.0% 置信区间	
	B	标准误	Beta			下限值	上限值
(常量)	-354.998	75.101		-4.727	0.000***	-509.999	-199.997
\bar{y}_k	2.588	0.32	0.941	8.09	0.000***	1.928	3.248
δ_k	1.872	0.476	0.457	3.931	0.001***	0.889	2.855

* $p < 0.1$; ** $p < 0.05$; *** $p < 0.01$

我们发现，系数的显著性检验 P 值均小于显著性水平 0.05，均拒绝原假设，认为系数都是显著不为零的。所以我们方案四的模型最终改进如下所示

$$\hat{y}_k = 2.588 * \bar{y}_k + 1.872 * \delta_k - 354.998 \quad (4.17)$$

其中 \bar{y}_k 表示使用方案四计算出的第 k 个作品最终得分，我们使用 \hat{y}_k 取代 \bar{y}_k ，并将该方案命名为方案六。

表 11 拟合优度检验

R^2	调整后的 R^2	Durbin-Watson(U)
0.745	0.724	1.586

表 12 方案五方差分析表

	平方和	自由度	均方	F	显著性
回归	2089.731	2	1044.865	40.814	.000***
残差	614.421	24	25.601		
总计	2704.151	26			

* $p < 0.1$; ** $p < 0.05$; *** $p < 0.01$

表 13 方案五系数显著性检验表

	非标准化系数		标准系数	t	显著性	95.0% 置信区间	
	B	标准误	Beta			下限值	上限值
(常量)	-341.551	71.159		-4.8	0.000***	-488.416	-194.686
\bar{y}_k	2.776	0.333	0.971	8.349	0.000***	2.09	3.462
δ_k	2.093	0.476	0.511	4.394	0.000***	1.11	3.076

* p<0.1 ;** p<0.05;*** p<0.01

同理，由表 12 和表 13 可知，模型的显著性检验和参数的显著性检验均通过，并且 R^2 为 0.745，拟合效果较好，最终模型如下所示

$$\hat{y}_k = 2.776 * \bar{y}_k + 2.093 * \delta_k - 341.551 \quad (4.18)$$

其中 \bar{y}_k 表示使用方案五计算出的第 k 个作品最终得分，我们使用 \hat{y}_k 取代 \bar{y}_k ，并将该方案命名为方案七。

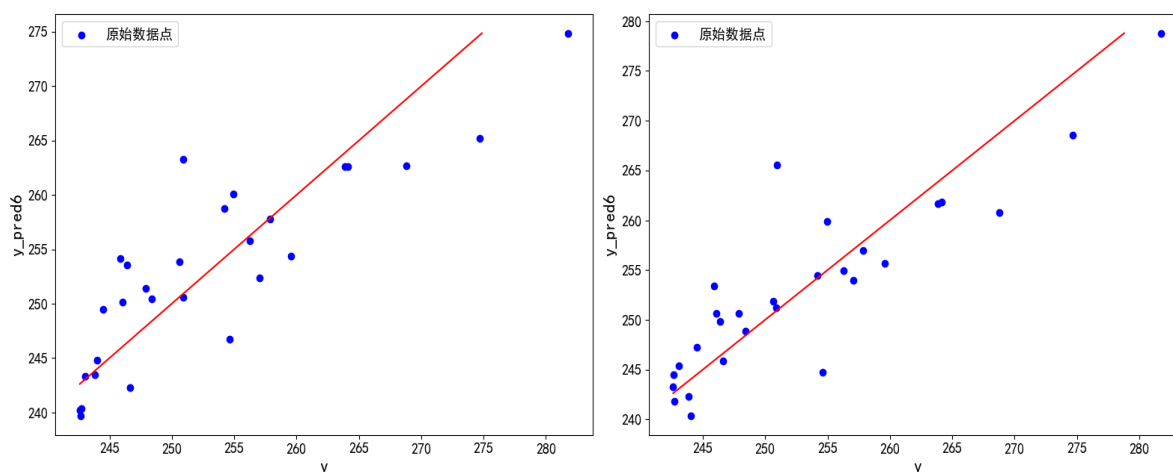


图 20 方案六（左）与方案七（右）拟合效果图

上图为 y_k 与 \hat{y}_k 的散点图，红线为对称轴，越接近对称轴表示拟合效果越好，我们发现，整体的拟合效果还不错。

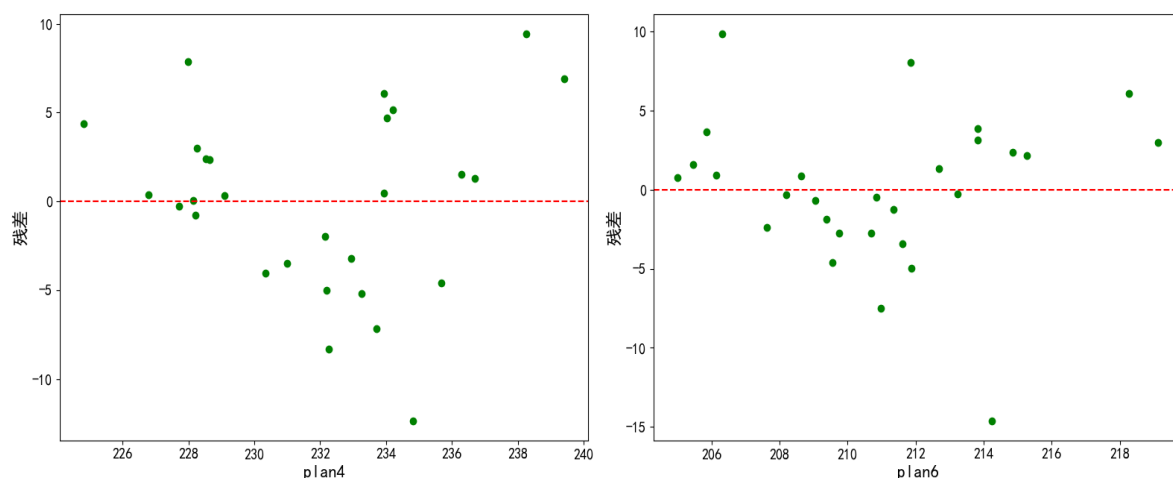


图 21 方案六（左）与方案七（右）残差图

根据残差图，我们发现，残差都大致在 10 以内波动，为了检验模型的真实效果，我们同样计算 spearman 系数和 kendall 系数，结果如所示。

表 14 改进后方案结果

阶段	方案	spearman	P 值	Tau_b	P 值
两阶段	方案三 (baseline)	0.71429	0.00003	0.58974	0.00000
	方案六* (Maxmin-scale+ γ + δ)	0.83689	0.00000	0.65079	0.00000
	方案七 (Mean-scale+ γ + δ)	0.76683	0.00000	0.59259	0.00000

*表示最优模型

尽管方案七的拟合优度高于方案六，但是从我们对于排名的评价指标来看，spearman 系数方案六和方案七分别有 17.16%和 7.35%的提升，kendall 系数有着 10.35%和 0.48%的提升，综合来看，方案六的结果优于方案七，所以我们选择方案六作为我们的最终模型。

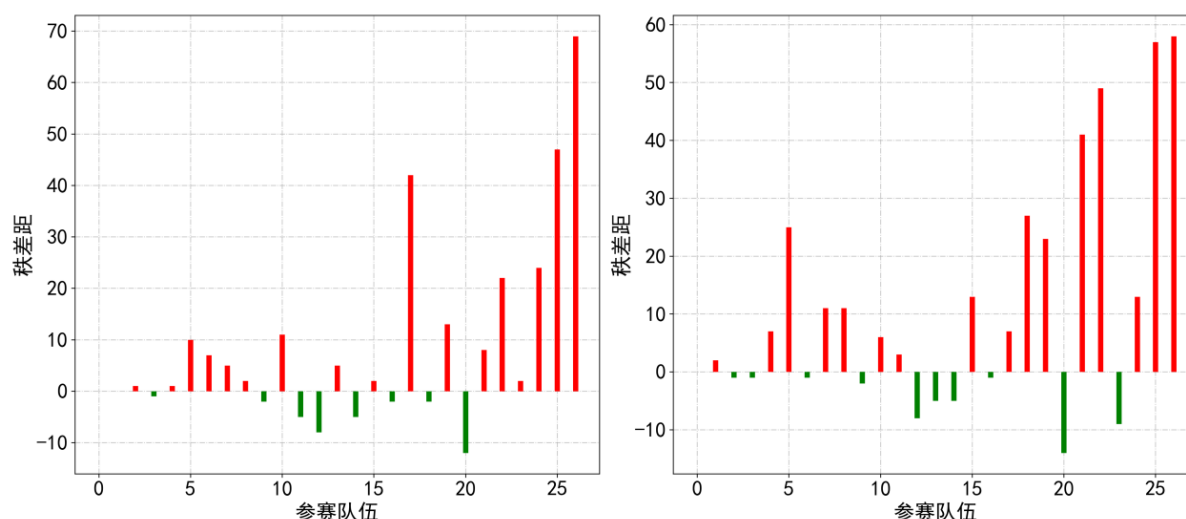


图 22 方案六（左）和方案七（右）的秩差图

我们发现相比于之前的方案四和方案五，上图的秩差大部分均控制在了 25 以内，并且极端值的秩差也都在 70 以内，说明模型的效果具有较大的提升。

4.2.5 最终评估结果展示

表 15 全部方案指标评估结果

阶段	方案	spearman	P 值	Tau_b	P 值
一阶段	方案一（标准分）	0.42857	0.02572	0.32194	0.01850
	方案二（去最大最小值）	0.21119	0.29032	0.13814	0.31619
二阶段	方案三 （标准分，baseline）	0.71429	0.00003	0.58974	0.00000
	方案四 （Maxmin-scale+ γ ）	0.69536	0.00006	0.54416	0.00001
	方案五 （Mean-scale+ γ ）	0.67705	0.00011	0.51567	0.00009
	方案六* （Maxmin-scale+ $\gamma+\delta$ ）	0.83689	0.00000	0.65079	0.00000
	方案七 （Mean-scale+ $\gamma+\delta$ ）	0.76683	0.00000	0.59259	0.00000

*表示最优模型

五、 问题 3 的模型建立与求解

5.1 问题分析

由于对成绩进行调整仅在二阶段的三位评委中进行，所以我们首先需要考虑在什么情况下，需要对作品的成绩进行调整（与极差相关），然后，根据题目中所给出的数据，分析两阶段成绩和极差的变化情况，分析分阶段评审和不分阶段评审的利弊，寻求创新性与极差之间的关系，建立极差模型，并给出在中等分段中处理大极差的方法。

所以第三问的解答需要分为以下步骤进行

- （1）对数据 2.1 与数据 2.2 进行预处理，对排名进行重排，对极端值进行处理（评阅作品数量过少的评委）。
- （2）分别从个体和整体的角度分析使用二阶段方案与不区分方案的成绩与极差的变化。
- （3）挖掘数据信息，探究是否复议与极差之间的关系。进一步寻找极差调整前后的变化关系。
- （4）建立极差关系模型，分极差区间进行调整，拟合函数关系，据此进一步优化方案，并根据题目已知的一阶段极差大小和优化方案模型，对中等分段（未进入二阶段）的参赛队成绩进行调整。

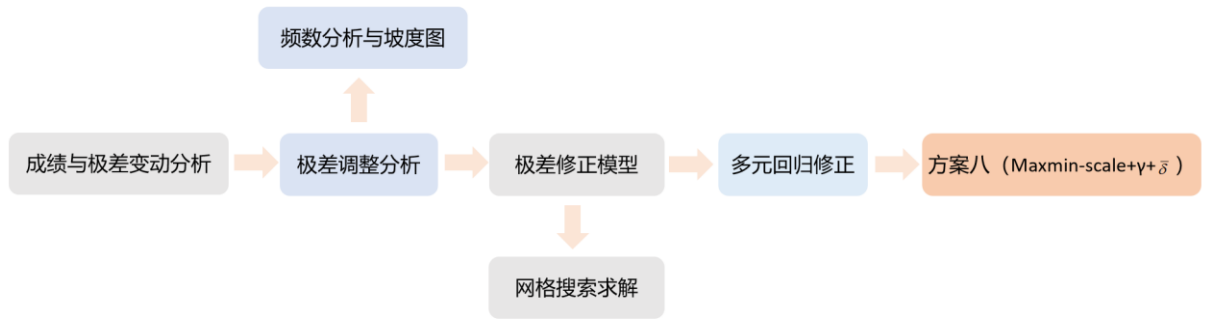


图 23 问题三流程图

5.2 模型建立与求解

5.2.1 数据预处理

(1) 数据 2.1 处理

我们发现，数据 2.1 中共有 884 支参赛队，42 位评委，其中一阶段参与评委 42 位，二阶段参与评委 10 位。相比于数据 1，可以认为该数据集是较小数据集，并且我们发现数据中存在相同分数的现象。

表 16 数据 2.1 部分样本

出现位置	最终得分	排名	奖项	最高分	最低分
275	269.47	301	未获奖	58.21	51.42
295	269.47	302	未获奖	64.74	48.91
625	269.47	300	未获奖	59.30	47.70

对于相同得分数据，数据中采取的是顺序排序（名次各不相同）的做法，但是排名的先后顺序与出现位置和最高分均无关系，为了确保后续评估的合理性，我们统一使用第二问中提出的“Min”秩排方式，对数据进行重排。

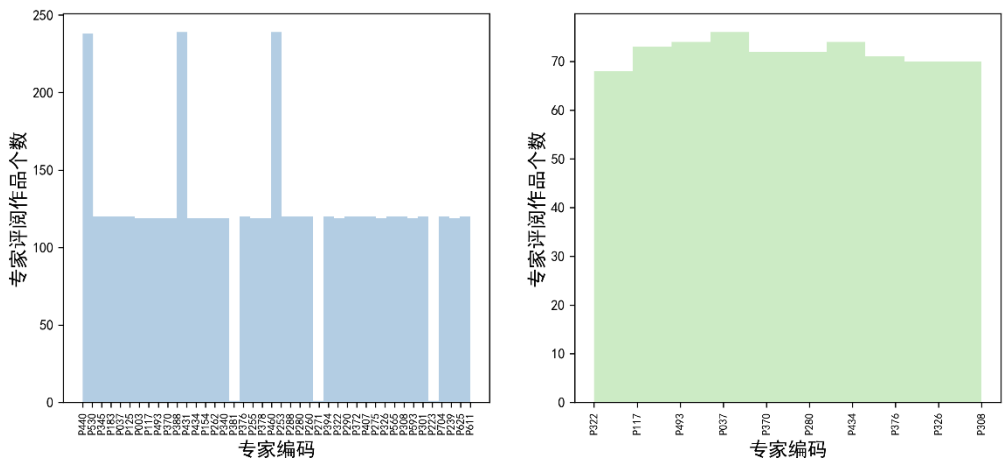


图 24 专家评阅作品数量（数据 2.1）

上图为数据 2.1 中一阶段和二阶段所有专家的评阅作品数量，我们可以发现存在个别专家评阅数量仅有一份，分别是 **P381,P271,P275,P223,P611**。对于评阅作品数量极少的专家，标准分的计算公式失去意义，所以在后续计算标准分时，我们使用原始分进行填充。

(1) 数据 2.2 处理

数据 2.2 中共有 9317 支参赛队，380 位评委参与评分，一阶段参与评委 380 位，二阶段参与评委 78 位。该数据集相比与数据 1 和数据 2.1，样本量较大，数据三者中的大数据集，与（1）相同，我们首先处理数据的排名问题。

表 17 数据 2.2 部分样本

最终得分	排名	奖项	调整后排名
215.48	623	二等奖	621
215.48	622	二等奖	621
215.48	625	二等奖	621
215.48	621	二等奖	621
215.46	624	二等奖	625

以表 17 为例，我们发现，数据 2.2 中出现了与数据 1 中相似的错误，排名错乱问题，分数高的作品反而排名低，为了保证计算排名的一致性，我们给出了新的排名结果。

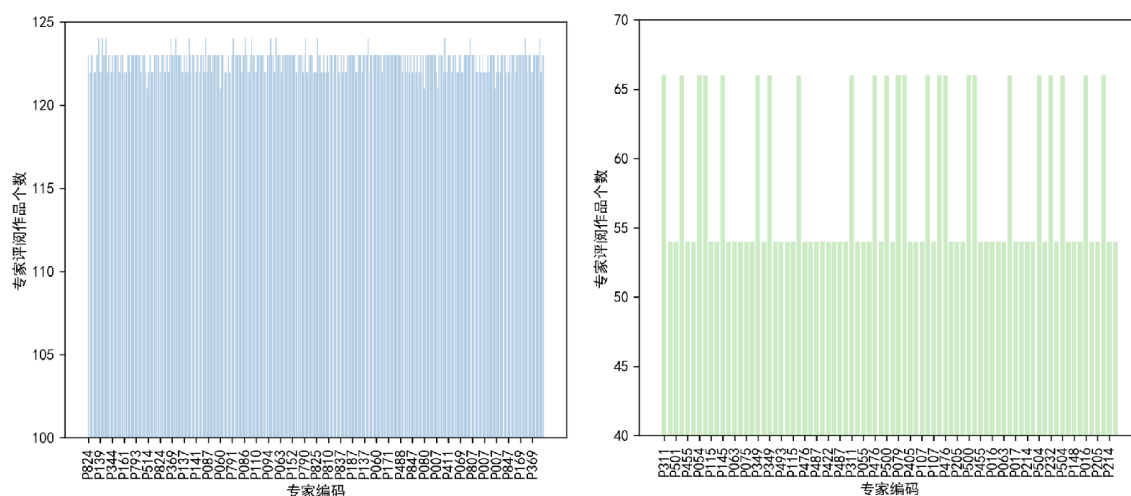


图 25 专家评阅作品数量（数据 2.2）

我们发现，对于数据 2.2 而言，并没有出现专家评阅作品数量极少的情况出现，一阶段中的评阅数量都在 120~125 之间，二阶段都在 53~70 之间。所以对于该部分，我们不做异常值处理。

5.2.2 成绩与极差变动分析

（1）成绩变动分析

经历二阶段的成绩难免会有所变动，不分阶段评审也就意味着直接使用一阶段的五位评委成绩进行加权，得出排名，最终成绩为五位评委打分之和。二阶段为前五位评委得分的均值加上后三位评委的打分作为最终得分，即不分阶段的最终成绩是 5 份成绩之和，分阶段的最终成绩是 4 份成绩之和。所以我们认为直接对成绩进行比较是没有意义的，最终结果肯定是总体呈下降趋势，所以，我们考虑使用排名的变动来分析，是否经历二阶段对参赛队的影响。

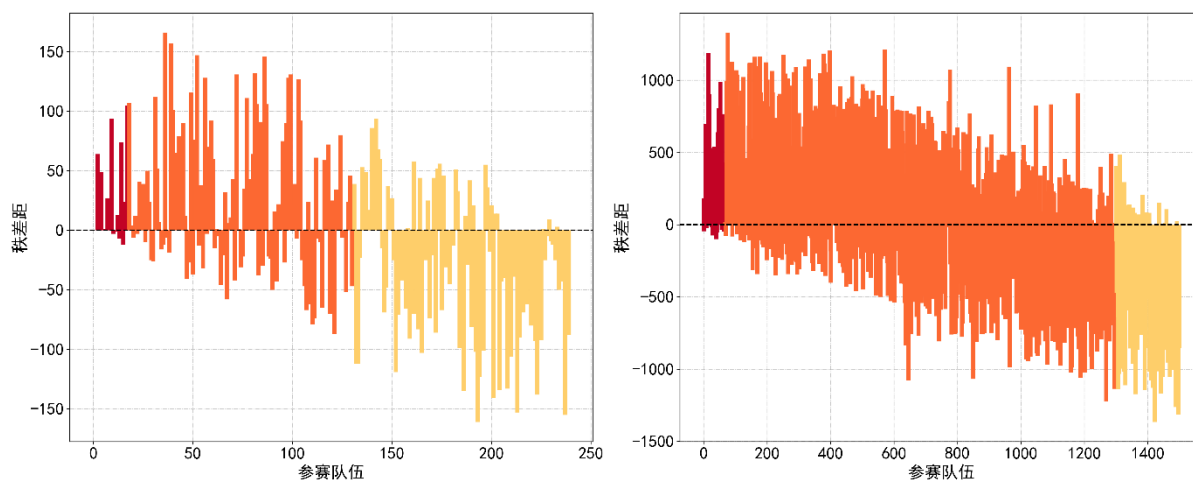


图 26 不同方案排名秩差（left:数据 2.1，right:数据 2.2）

三种不同颜色的线分别表示一等奖、二等奖、三等奖的参赛队（以数据中给出的二阶段排名作为基准），秩差为正表示排名上升，我们发现，经历过二阶段的之后的一等奖参赛队与二等奖参赛队的排名大体上呈上升趋势，这说明两阶段的方案更有利于筛选出真正优秀的作品。与之相反的是，三等奖的作品排名都下降了，这也证明了，两阶段对成绩的考量更多，优中择优，能够给出更为合理的最终成绩。

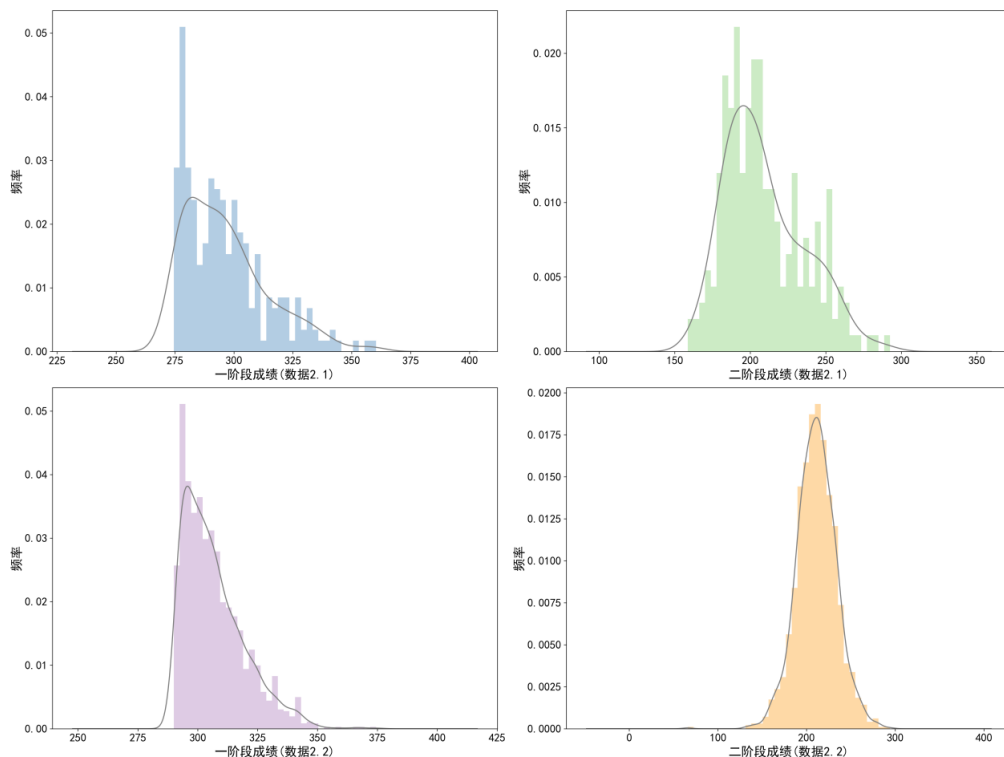


图 27 成绩分布图

由图 27 可知，我们发现，对于一阶段而言，大部分的成绩都在低分区域，也就是成绩的分布是右偏的，数据 2.1 中的大部分成绩集中在 275~300 之间，2.2 的数据大部分集中在 300~325 之间。

对于二阶段的方案，我们发现，数据 2.1 分布做了一定的修正，成绩的峰度变大，数据分布向中心靠拢，数据 2.2 对于成绩修正的效果更加明显，几乎都从右偏分布修正到了对称分布，所以我们认为二阶段对于成绩的优势在于对数据进行修正，使得数据是中心对称的。

(2) 极差变动分析

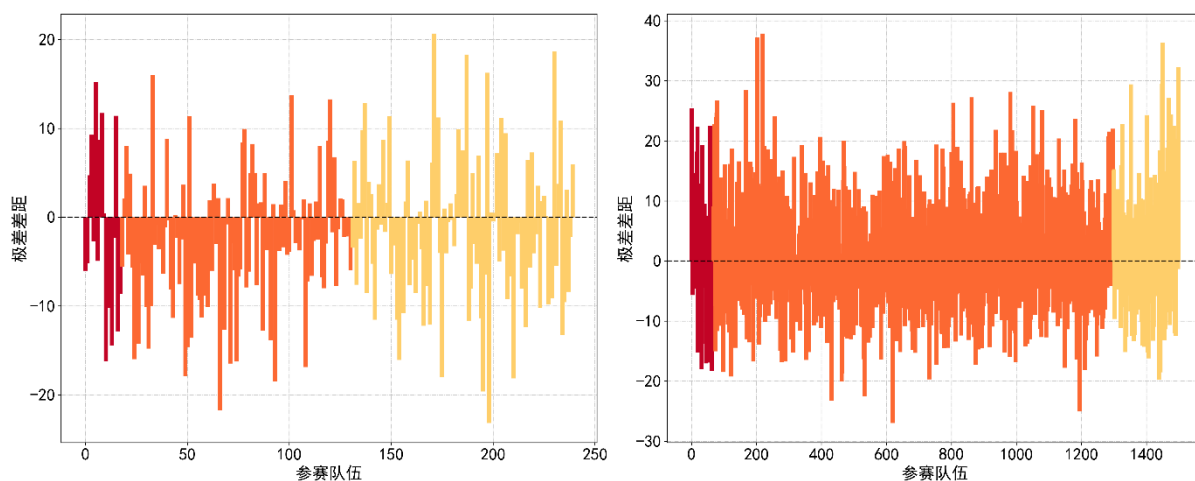


图 28 不同方案极差变动图（left:数据 2.1，right:数据 2.2）

上图为二阶段极差减去不区分阶段极差的差距图，对于左图，我们可以发现，二等奖部分的队伍，极差大部分都减小了，说明采取两阶段的方案可以有效的减少极差，消除异常因素影响，对于右图，整体的极差波动较大，但是极差的变换并无明显趋势，增减交替。

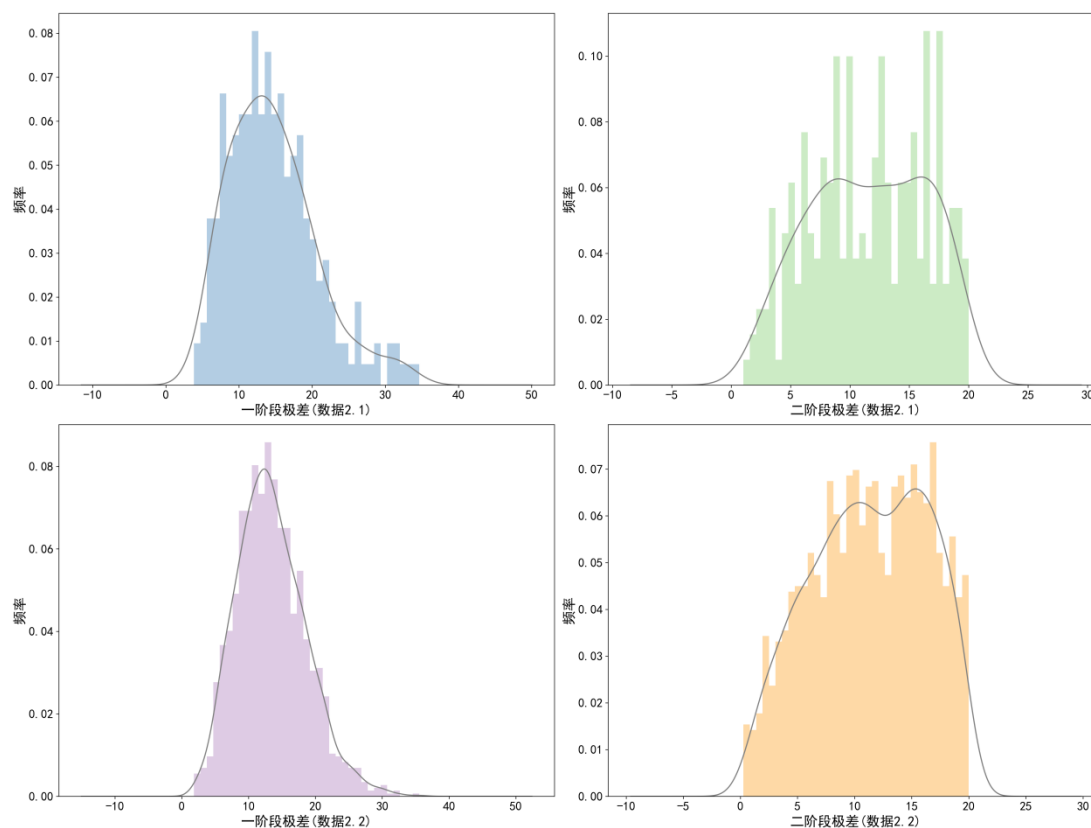


图 29 极差分布图

我们发现，无论是数据 2.1 还是数据 2.2 的极差，都是呈现尖峰分布，极差普遍都在 10~20 之内，二对于二阶段的极差，我们发现，数据呈现一个扁平分布的态势，并且我们发现，极差均在 20 以内。

所以，我们得出结论，两阶段评审方案的效果是优于一阶段评审的，不仅对最终成绩进行了“正态”修正，还使得极差变小，分布均匀。一阶段的优点仅仅在于耗费的资源较少，可以减少一定的人力物力消耗。

5.2.3 极差调整分析

简单观察数据，我们发现，仅有二阶段的三位评委的评审成绩涉及到是否复议，一阶段的五位评委的评阅成绩并不会复议，并且是否需要调整与二阶段标准分的极差相关，所以我们需要探求极差与是否调整（复议）之间的关系。

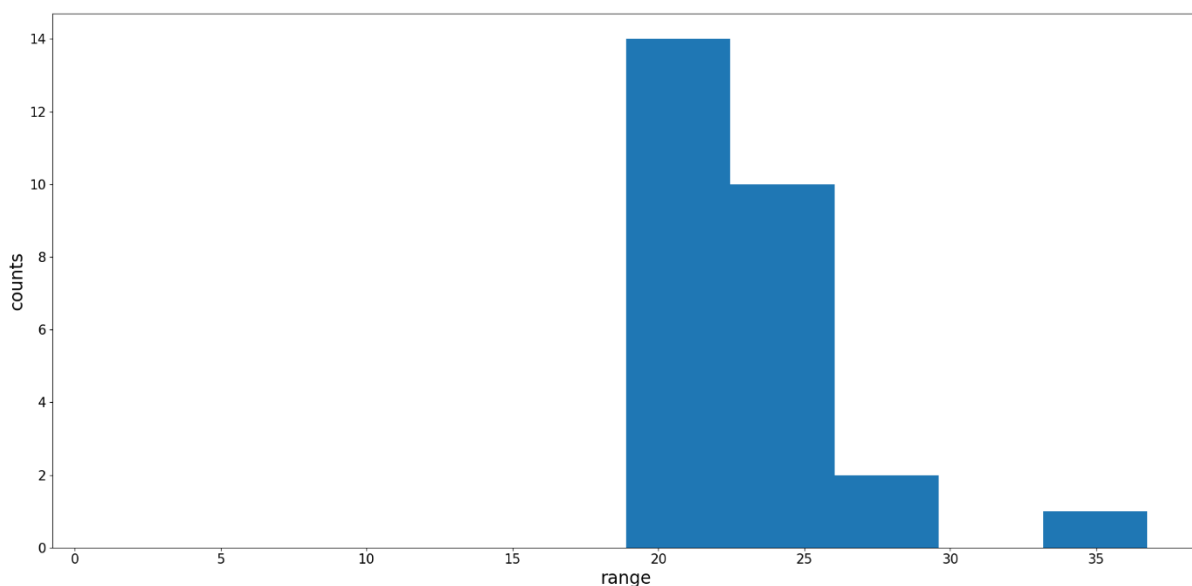


图 30 数据 2.1 极差-复议频数图

如图 30 所示，我们发现，对于数据 2.1，第二次评审的标准分极差在 1.02~36.76 之间，我们将区别划分为 10 等分，对在区间内是否复议的作品作频数统计，我们发现，只有极差大于 20 左右时，才会对作品进行重新评审。

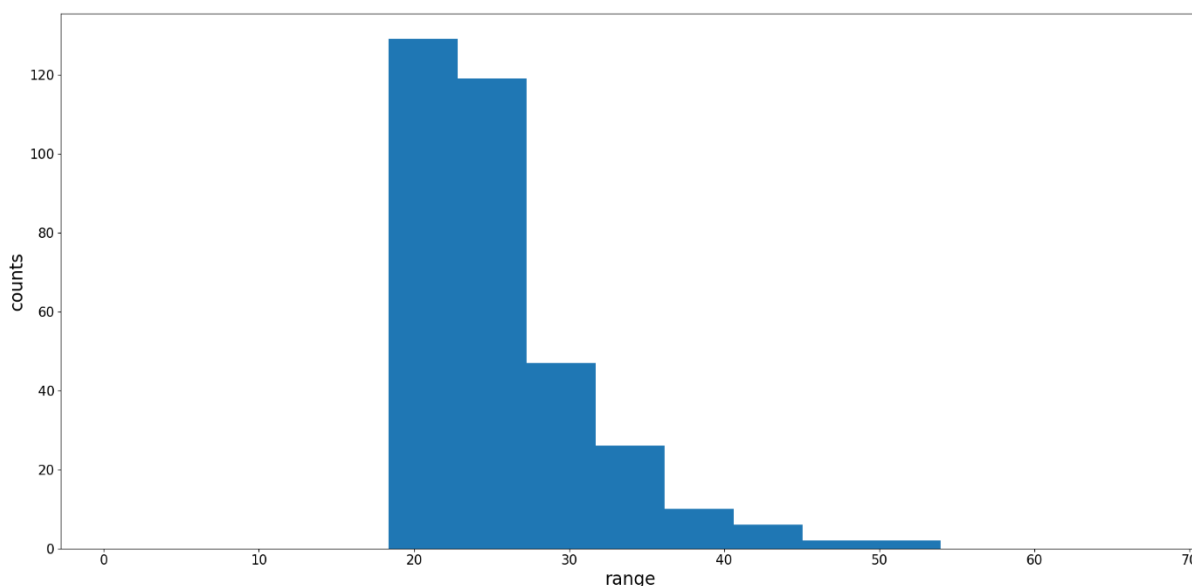


图 31 数据 2.2 极差-复议频数图

同理，我们发现，对于数据 2.2 而言，也是在极差大于 20 的部分才出现了复议情况，所以我们可以得出结论，当二阶段的标准分极差大于 20 时，三位评委需要重新评审。

尽管根据上文，我们已经知道了当标准分的极差大于 20 时，二阶段的三位评委需要复议，然后给出复议之后的分数，但是我们发现，复议的三位评委中，存在一个现象，即有 1~2 位评委的复议前后成绩是不会变动的。

我们认为，复议前后的成绩是否变动是与该为评委在该组中的打分相对位置相关，也就是说，是与均值差相关。均值差大的分值会经过调整，逼近均值，从而使得调整之后的极差变小。

并且我们在问题二中引入的偏差因子的方案在效果上达到了最优，当时是通过偏差因子来拟合函数关系，并且偏差因子是由极差构成的，所以我们有理由认为，方案六（未经历调整阶段）与真实得分之间的差距可以由极差拟合，在问题2中我们已经考虑了复议前极差。现在，我们需要进一步将调整后极差作为影响因素考虑在内。

5.2.4 建立极差模型

现在，我们合并数据 2.1 和 2.2，并且只考虑重新评审的样本，我们需要建立以下函数关系

$$\bar{\delta}_k = f(\delta_k) \quad (5.1)$$

其中， $\bar{\delta}_k$ 是复议后极差， δ_k 为问题二中的偏差因子，为二阶段标准分的极差。

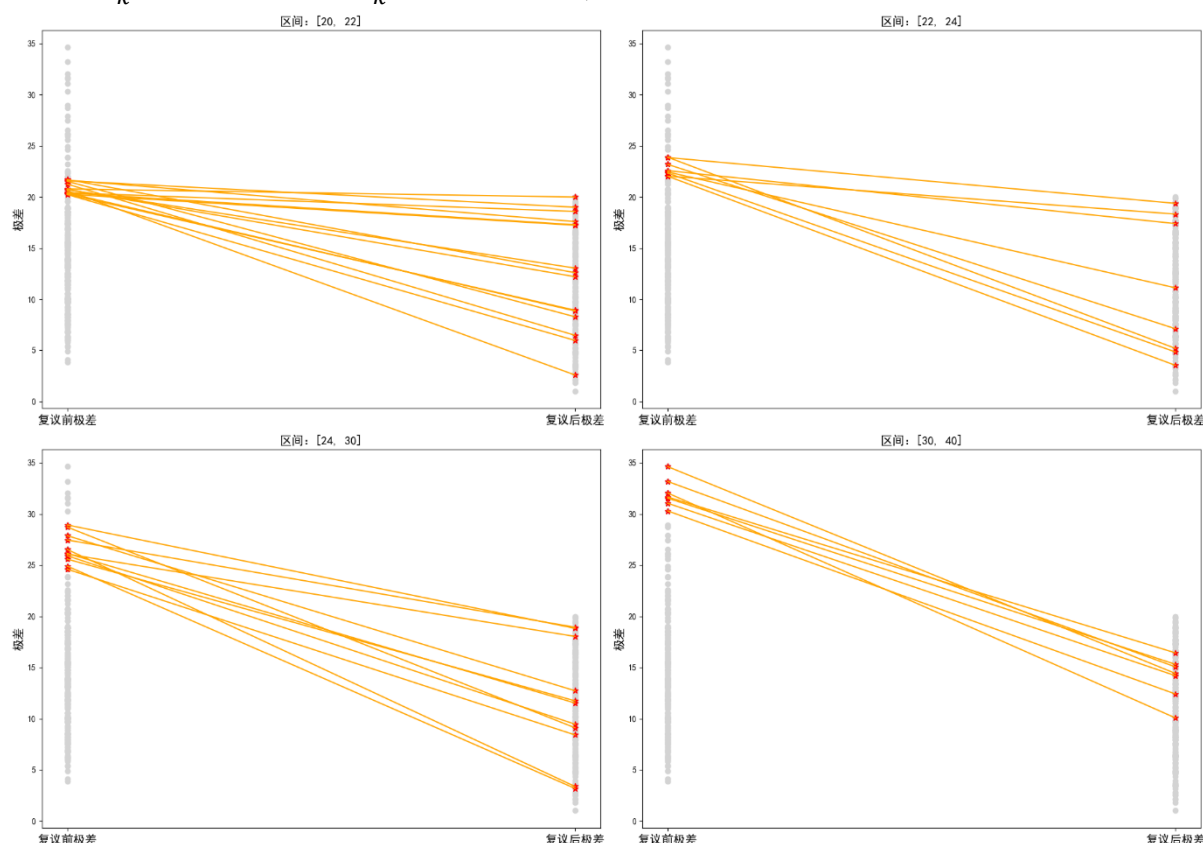


图 32 复议前后坡度图

坡度图，如图 32 所示，常常用于分析变量的变化情况。我们发现，复议前后的极差呈现一定的线性关系，由于上图的区间是等分的，所以我们考虑对极差区间进行合适的划分，使得区间内映射线段斜率相同。

我们认为，如果在一个区间内复议的样本量较多，那么应该对区间作更加细致的划分，从而更好的拟合函数关系式。基于上述目的，我们提出以下模型。

极差修正模型

第一步，我们在残差区间内（大于 20）作 N 等分，然后我们统计各个区间内的复议作品个数。

$$N_r = [n_1, n_2, \dots, n_N] \quad (5.2)$$

其中， N_r 表示 N 等分区间的频数统计。

第二步，我们使用频数的倒数作为新的比例，并在缩放之后重新对区间进行划分。

$$\tilde{N}_r = \left[\frac{1}{n_1}, \frac{1}{n_2}, \dots, \frac{1}{n_N} \right] \quad (5.3)$$

$$P_r = \frac{\tilde{N}_r}{\text{Sum}(\tilde{N}_r)} \quad (5.4)$$

第三步，使用 P_r 对区间重新进行划分 $R = [r_1, r_2, \dots, r_{2N}]$ ， $r_1 \sim r_{2N}$ 为 N 个区间的左右端点，分区间对函数进行拟合，即

$$\bar{\delta}_k = f(\delta_k) = \begin{cases} f_1(\delta_k) & = w_1 \delta_k & \text{if } \delta_k \in [r_1, r_2] \\ f_2(\delta_k) & = w_2 \delta_k & \text{if } \delta_k \in [r_3, r_4] \\ \dots & \\ f_N(\delta_k) & = w_N \delta_k & \text{if } \delta_k \in [r_{2N-1}, r_{2N}] \end{cases} \quad (5.5)$$

并且考虑两两区间重合, $r_3 - r_2 = r_5 - r_4 = \dots = r_{2N-1} - r_{2N-2} = \varepsilon$, $r_1 = 20 - \varepsilon$, 当 δ_k 落入第一个重合区间 $[r_3: r_2]$ 时, $w = \frac{w_1 + w_2}{2}$, $f(\delta_k) = w \delta_k$, 依次类推。

模型求解

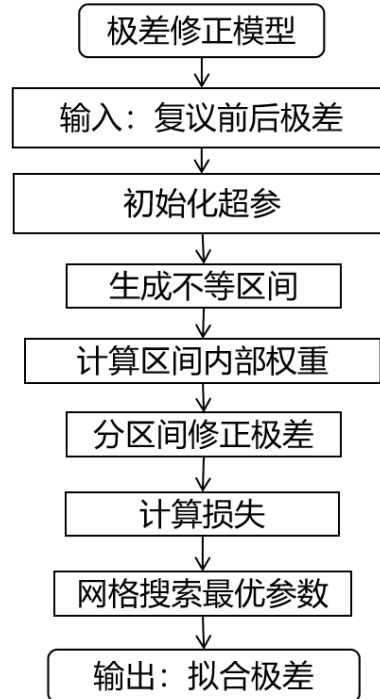


图 33 算法流程图

我们在参数 $N = \{2, 3, 4, 5, 6, 7\}$, $\varepsilon = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$ 中进行网格搜索调优，并且对 ε 进行多次迭代折半搜索，最终结果如下所示

表 18 求解结果

parameter	value
N	7
ε	0.65
R	[19.35, 22.87, 21.57, 25.09, 23.79, 27.31, 26.01, 29.53, 28.23, 31.75, 30.45, 33.97, 32.67, 36.19]
$W = [w_1, w_2, \dots, w_N]$	[0.526, 0.432, 0.42, 0.401, 0.426, 0.473, 0.086]

于是模型的最终表达式为

$$\bar{\delta}_k = f(\delta_k) = \begin{cases} f_1(\delta_k) = 0.526\delta_k & \text{if } \delta_k \in [19.35, 22.87] \\ f_2(\delta_k) = 0.432\delta_k & \text{if } \delta_k \in [21.57, 25.09] \\ \dots & \\ f_7(\delta_k) = 0.086\delta_k & \text{if } \delta_k \in [32.67, 36.19] \end{cases} \quad (5.6)$$

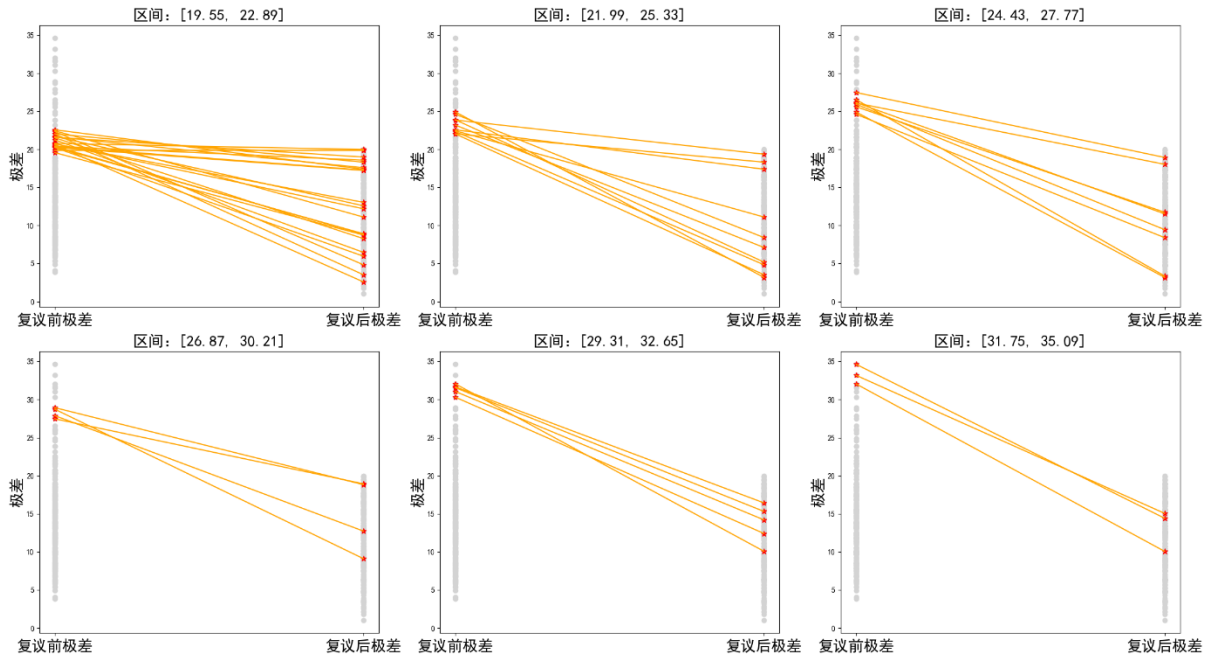


图 34 拟合结果图

最终拟合出来的结果如图 34 所示，可以发现，除了在第一个区间效果不是很好，在其余区间效果较好，大部分数据斜率相近。

5.2.5 方案六模型改进

我们在问题二中提出的所有模型，均不考虑复议的影响，所以使用一个偏差因子进行修正，此时，我们考虑将复议后极差作为新的影响因子，替换原始因子，所以方案六被修正为

$$y_k = f(\bar{y}_k, \bar{\delta}_k) + \varepsilon_i \quad (5.7)$$

$\bar{\delta}_k$ 见公式(5.6)

我们针对数据 2.1 与数据 2.2 重新拟合模型，结果如下所示：

表 19 拟合优度检验

R	R ²	调整后的 R ²
.872	0.761	0.76

表 20 方案六方差分析表

	平方和	自由度	均方	F	显著性
回归	575534.096	2	287767.048	2380.119	.000***
残差	180994.001	1497	120.904		
总计	756528.097	1499			

* p<0.1 ;** p<0.05;*** p<0.01

表 21 方案六系数显著性检验表

	非标准化系数		标准系数	t	显著性
	B	标准误	Beta		
(常量)	-589.659	11.613		-50.775	0.000***
\bar{y}_k	3.604	0.052	0.872	68.99	0.000***
$\bar{\delta}_k$	-0.114	0.053	-0.027	-2.15	0.000***

* p<0.1 ;** p<0.05;*** p<0.01

$$\hat{y}_k = 3.604 * \bar{y}_k - 0.114 * I_{\Omega}(\delta_k) * \bar{\delta}_k - 589.659 \quad (5.8)$$

其中 $\Omega = \{x|x \geq 20\}$, δ_k 表示调整前极差, $\bar{\delta}_k$ 表示调整后极差。

我们在方案六的基础上进一步考虑了复议极差, 将其命名为方案八, 下文与此相同。

5.2.6 模型评估

我们对方案八进行模型评估, 评估结果如表 22 所示

表 22 方案六与方案八模型评估结果

阶段	方案	数据	spearman	P 值	Tau_b	P 值
两阶段	方案六					
	(Maxmin-scale+ $\gamma+\delta$)	1	0.83689	0.00000	0.65079	0.00000
	方案八	2.1	0.9660	0.0000	0.8427	0.0000
	(Maxmin-scale+ $\gamma+\bar{\delta}$)	2.2	0.8788	0.0000	0.7047	0.0000

我们发现, 使用新的偏差因子之后, 模型的效果极好, 在 2.1 的小数据中 spearman 系数甚至达到了 0.9660, 这意味着排名近乎正确, 并且我们的基准数据是进入二阶段的所有作品。虽然在大数据集 2.2 中效果有所下降, 但是相对而言, 仍然比我们之前提出的方案六 (最佳模型) 效果要好。

5.2.7 中等分段作品处理

我们认为, 中等分段的作品为排名在进入二阶段作品之后的, 区间长度为二阶段作品的数量作品集合。我们使用模型八对中等作品得分进行“复议”。

需要注意的是, 由于中等分段极差阈值和我们之前的可能有所不同, 我们假设“大极差”为我们之前得到的极差阈值 20。所以我们对于计算公式, 需要进行跳跃

（skip）操作，使得极差阈值降低到 10.并且由于模型八是针对二阶段的模型，若是需要对一阶段的作品使用，需要进行 transform 操作。我们将该模型称为“校正模型”，具体公式如下所示

$$\delta_k = \text{skip}(\delta_k) \quad (5.9)$$

$$\hat{y}_k = 3.604 * \text{transform}(\bar{y}_k) - 0.114 * I_{\Omega}(\delta_k) * \bar{\delta}_k - 589.659 \quad (5.10)$$

$$\hat{y}_k = \text{Inverse_transform}(\hat{y}_k) \quad (5.11)$$

其中， $\text{transform}(x) = \frac{4}{5}x$, $\text{skip}(x) = x + 10$, δ_k 表示调整前极差， $\bar{\delta}_k$ 表示调整后极差。

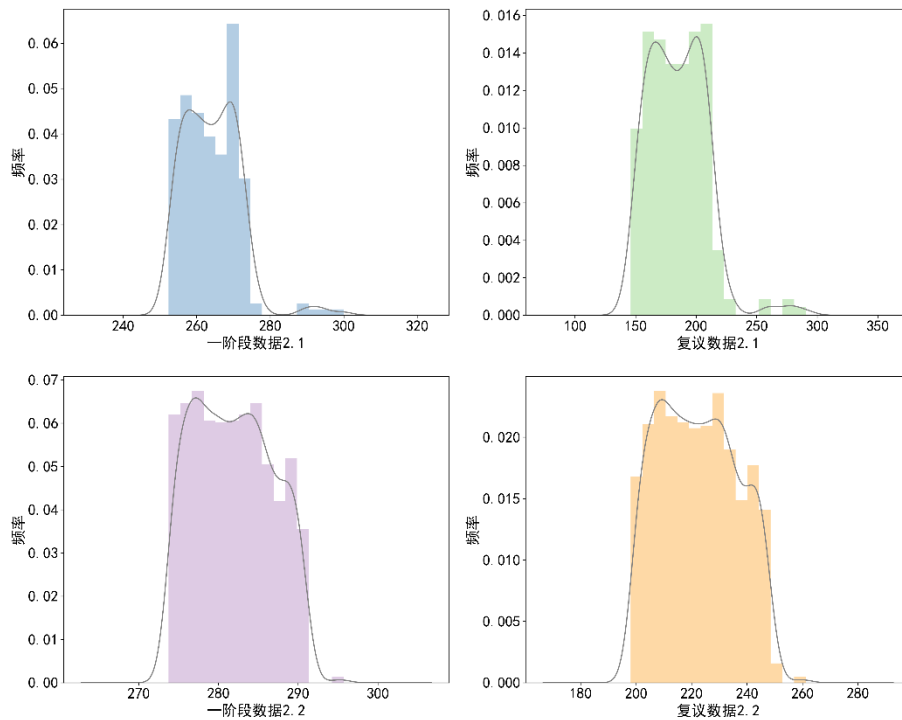


图 35 标准分“复议”前后成绩分布图

由于方案八本身是对二阶段提出的，所以我们计算出的 \bar{y}_k 需要乘以缩放因子 4/5，然后最终得出的调整成绩，实际上是二阶段的成绩预测，所以若是需要得到基于一阶段尺度的成绩，需要在最终结果 \hat{y}_k 的基础上乘以 5/4.上图中并未进行“逆变换”操作。

我们直接使用标准分之和对公式（5.11）中的 \bar{y}_k 进行替代，探究复议对于一阶段样本的影响，我们发现，无论是数据 2.1 还是数据 2.2，“复议”均使得样本的分布变得扁平，数据分布更为分散。

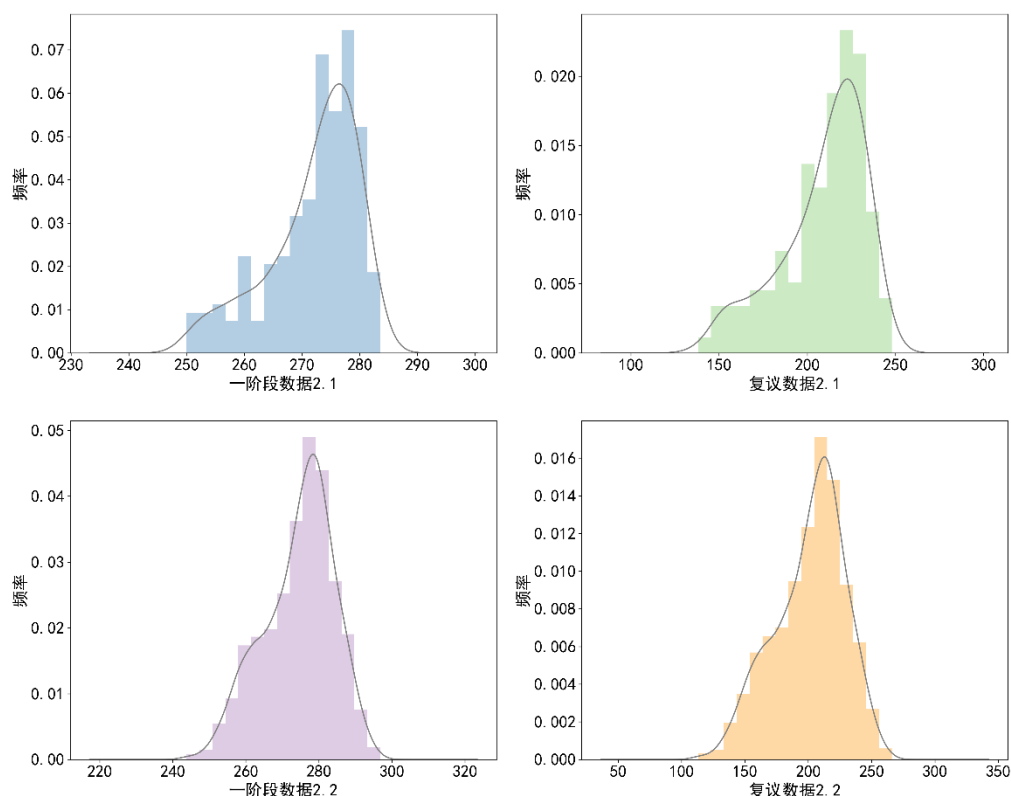


图 36 方案八“复议”前后成绩分布图

可以发现，如果使用我们的标准分计算公式，在复议前后分数均变得“矮胖”，轮廓变化不大，并且，我们可以发现，2.2 的一阶段数据近似对称分布，这说明我们提出的标准分计算公式是有效的。

六、问题 4 的模型建立与求解

6.1 问题分析

问题四中要求我们给出完整的评审模型，其中应当包括作品分配、一阶段筛选、二阶段评审（重审）。并且说明如何根据已知的原始分数数据进行完整流程计算，得出最终排名。

对于模型不足的地方，提出改进建议，完善模型。

6.2 模型建立与求解

6.2.1 评审方案流程设计

（1）作品分配：

对于作品分配，我们使用在问题一中提出的搜索优化算法

$$\begin{aligned} & \text{Max } Z_{\text{mean}} - \lambda * Z_{\text{var}} \\ & \text{s. t. } \begin{cases} \mathbf{w}_i = 5 \quad i = 1, 2, \dots, m \\ 110 \leq \mathbf{p}_j \leq 135 \quad j = 1, 2, \dots, n \\ 0 \leq s_{i,j} \leq 1 \\ 0 \leq o(P) \leq [0.2n] \\ s_{i,j} = \frac{\mathbf{V}_i^T \odot \mathbf{V}_j}{\max(p_i, p_j)} \\ v_{i,j} \in \{0, 1\} \end{cases} \end{aligned} \quad (6.1)$$

根据公式(6.1)求解出最佳的作品-评委矩阵。

(2) 一阶段评审

对于一阶段，我们使用在 5.2.7 小节中提出的校正模型，

$$\delta_k = \text{skip}(\delta_k) \quad (6.2)$$

$$\hat{y}_k = 3.604 * \text{transform}(\bar{y}_k) - 0.114 * I_{\Omega}(\delta_k) * \bar{\delta}_k - 589.659 \quad (6.3)$$

$$\hat{y}_k = \text{Inverse_transform}(\hat{y}_k) \quad (6.4)$$

其中 $\text{transform}(x) = \frac{4}{5}x$, $\text{skip}(x) = x + 10$

求解出 \hat{y}_k 之后，根据 \hat{y}_k 对作品进行排名，按事先约定的比例取排名在前的作品，进入第二阶段评审。

(3) 二阶段评审

对于二阶段评审，选择使用我们的最优方案（方案八）进行打分。

$$\hat{y}_k = 3.604 * \bar{y}_k - 0.114 * I_{\Omega}(\delta_k) * \bar{\delta}_k - 589.659 \quad (6.5)$$

$$\bar{\delta}_k = f(\delta_k) = \begin{cases} f_1(\delta_k) & = 0.526\delta_k & \text{if } \delta_k \in [19.35, 22.87] \\ f_2(\delta_k) & = 0.432\delta_k & \text{if } \delta_k \in [21.57, 25.09] \\ \dots & \\ f_7(\delta_k) & = 0.086\delta_k & \text{if } \delta_k \in [32.67, 36.19] \end{cases} \quad (6.6)$$

最后，得出最终成绩，对参赛队进行排名。

6.2.2 方案改进

我们在问题二中讨论过不同的老师有不同的打分倾向，其中专业领域的倾向（例如：歌唱比赛中，有的老师是古典音乐领域，有的是流行音乐领域）已经通过我们之前提出的震荡因子 γ 和基于极差修正模型的方案八消除。

现如今，我们应当考虑的问题在于，老师的打分倾向（低分、中分、高分）我们无法得知，根本原因在于，我们无法知道分配到该老师的作品质量如何（作品学术水平分布非正态）。所以我们通过问卷的方式提取不同老师的打分偏好，具体流程图如下所示。

其中，黄色的模块是我们提出的改进部分，老师的偏好因子通过问卷获取，问卷

信息如表 23 所示

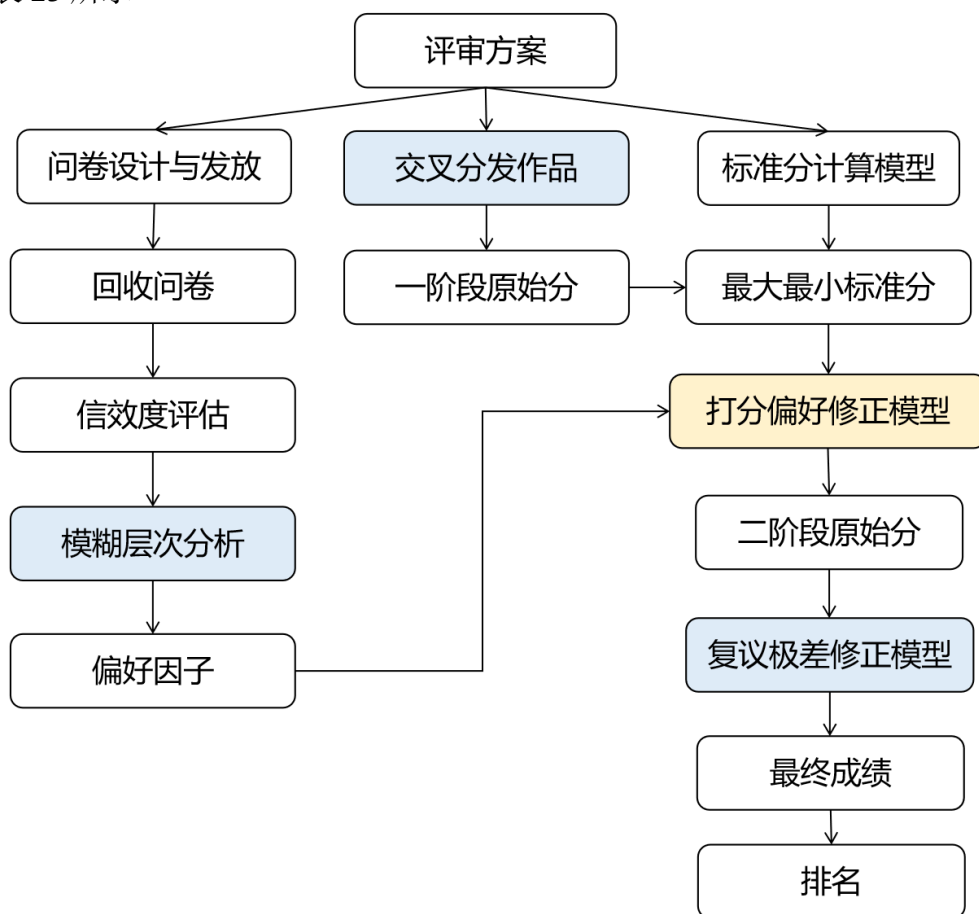


图 37 改进方案流程图

表 23 问卷设计表

问题编号	问卷问题
1	您认同您的打分标准与同组其他专家一致
2	在评分时，您会给作品打鼓励分
3	在评分时，您会避免打出超高分
4	作品的创新性可以弥补规范性上的欠缺
5	作品应该严格遵守规范和标准
6	本次参赛作品质量整体偏高
7	您对涉及本专业的作品会更严格
8	您没有理解作品的内容也不会打低分
9	您预估今天自己的打分普遍偏高
10	一等和二等作品之间的分差应该在 15 分以上

在设置问卷时，我们给各问题 1-5 分的赋分机制，由评委根据自身实际情况进行赋分。在收集到问卷后进行信度效度检验，在本问卷中采用 KMO 检验和 Bartlett 球形检验来衡量问卷效度^[6]。KMO 检验是 Kaiser 等人提出的抽样适合性检验，Bartlett 球形检验是用于检验相关系数矩阵是否为单位阵。信度分析的方法主要有以下四种：重测信度法、复本信度法、折半信度法、 α 信度系数法^[7]。我们使用用于反映内部一致性的指标： α 系数来衡量问卷的信度，计算公式如下所示：

$$\alpha = \frac{K}{K-1} \left(1 - \frac{\sum_{i=1}^K S_i^2}{S_K^2} \right) \quad (6.7)$$

其中 K 表示量表的题目个数， S_i^2 表示题内方差， S_K^2 表示所有题目的方差。问卷经过信度效度检验后可以基于问卷信息计算出优化因子。

模糊数学^[8]是在模糊集合，模糊逻辑的基础上发展起来的数学领域的统称，是研究和处理模糊性现象的一种数学理论和方法。在实际分析中，各个指标的重要程度是有差异的，对于影响力大的指标应赋予较大的权重，反之应该赋予较小的权重，本方案使用模糊数学确定权重来计算优化因子的大小。

七、模型评价

7.1 模型的优点

本模型的优点在于结构清晰，过程层层递进。首先，提出搜索优化算法，确保模型的可行性，在评价指标上，我们选择了具有权威性的非参数检验指标：**spearman** 系数和 **kendall** 系数。在新标准分方案中，充分考虑了专家、小组之间的异质性，并且效果上超越原始的标准分方案。最后，新颖的提出使用问卷提取信息，改善模型。

7.2 模型的缺点

出于打破正态性假设，模型并未考虑专家评分倾向。基于搜索优化算法模型的结果依赖于迭代次数，较少的迭代次数模型效果不佳。

八、参考文献

- [1]斯燕方. 基于 0-1 规划问题的动态 DNA 折纸计算模型[D].安徽理工大学,2021.DOI:10.26918/d.cnki.ghngc.2020.000066.
- [2]Fieller, E.C.; Hartley, H.O.; Pearson, E.S. (1957) Tests for rank correlation coefficients. I. *Biometrika* 44, pp. 470–481.
- [3]李世宽. 基于 Kendall 秩相关系数的沙漠地震噪声性质研究及应用[D].吉林大学,2020.DOI:10.27162/d.cnki.gjlin.2020.006289.
- [4]马瑄. 基于改进 DBSCAN 算法和 K-S 检验的收视数据集异常检测方法[D].北京邮电大学,2022.DOI:10.26969/d.cnki.gbydu.2021.002690.
- [5]陈希镇.现代统计分析方法的理论和应用=THEORY AND APPLICATION OF MODERN STATISTICAL ANALYSIS METHOD,国防工业出版社,2016.05,第 222 页.
- [6]"KMO and Bartlett's Test". IBM.
- [7]Goforth, Chelsea (November 16, 2015). "Using and Interpreting Cronbach's Alpha University of Virginia Library Research Data Services + Sciences". University of Virginia Library.
- [8]王雪. 基于 AHP-熵权法和模糊数学的城市生态系统健康评价研究[D].华东师范大学,2016.

附录:

```
import numpy as np
import pandas as pd
# 定义矩阵的行数和列数
num_rows = 3000
num_cols = 125
# 循环生矩阵
n_circle = 3000
z_multi = []
z_var = []
matrix_all = []
for m in range(n_circle):
    # 单个生成矩阵
    matrix = np.zeros((num_rows, num_cols))
    # 单个生成 p 矩阵
    p_matrix = np.zeros((num_cols, num_cols))
    for i in range(num_rows):
        row_sum = 0
        available_cols = list(range(num_cols))
        while row_sum < 5 and available_cols:
            col_idx = np.random.choice(available_cols)
            # 赋值为 1
            value = 1
            matrix[i, col_idx] = value
            row_sum = np.sum(matrix[i, :])
            available_cols.remove(col_idx)
    # 对矩阵进行筛选
    col_sums = np.sum(matrix, axis=0)
    # 检查列加和是否满足条件, 筛选矩阵
    #if all(100 <= col_sum <= 140 for col_sum in col_sums) or
    (len(col_sums) - sum(1 for col_sum in col_sums if 100 <= col_sum <=
    140) >= 25):
        if len(col_sums)-sum(1 for col_sum in col_sums if 100 <= col_sum <=
    140)<=25:
            # 计算矩阵的相关性和方差
            for i in range(num_cols):
                for j in range(num_cols):
                    a = sum(matrix[:,i])
                    b = sum(matrix[:,j])
                    p_matrix[i, j] = np.dot(matrix[:, i], matrix[:,
j])/max(a,b)
            # 计算相关性 z_multi
            z_multi_now = 0
            z_var_now = 0
            p_sanjiao = []
            for i in range(num_cols):
                for j in range(num_cols):
                    if i < j:
                        p_sanjiao.append(p_matrix[i, j])
```

```

        z_multi_now = np.sum(p_sanjiao)
        z_var_now = np.var(p_sanjiao)
        z_multi.append(z_multi_now)
        z_var.append(z_var_now)
        matrix_all.append(matrix)
    else:
        continue
# 最大 z_muti
min_var = min(z_var)
# 找到最大值的索引
min_index = z_var.index(min_var)
print("最小值:", min_var)
print("最小值的索引:", min_index)
best_matrix = matrix_all[min_index]
p_best_matrix = np.zeros((num_cols, num_cols))
for i in range(num_cols):
    for j in range(num_cols):
        a = sum(best_matrix[:,i])
        b = sum(best_matrix[:,j])
        p_best_matrix[i, j] = np.dot(best_matrix[:, i], best_matrix[:, j]) / max(a,b)
# 找出前 5 行中数字为 1 的列索引
num_rows_to_check = 10
num_cols_to_check = 10
result = []
for row in best_matrix[:num_rows_to_check]:
    indices = np.where(row == 1)[0] # 找到值为 1 的列索引
    result.append(indices[:5]) # 取前 5 个索引
result = pd.DataFrame(result)
import pandas as pd
col_sums = np.sum(best_matrix,axis = 0)

# 计算相似性度量
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
path = r'C:\Users\wyl\Desktop\2023 年中国研究生数学建模竞赛赛题\C 题'
data_name = '数据 1.xlsx'
data = pd.read_excel(path+'/' + data_name, sheet_name='Sheet2')

master_enc = data.iloc[:,1:].values
num_master = len(np.unique(master_enc)) # 97

rank = data.iloc[:,0].values
num_term = len(np.unique(rank)) # 2015
master = np.unique(master_enc)

V = pd.DataFrame(0,index=rank,columns=master)
master_enc = pd.DataFrame(master_enc,index=rank)

```

```

for i in rank:
    col = master_enc.loc[i,:].values
    V.loc[i,col] = 1

S = pd.DataFrame(0,index=master,columns=master)
for i in master:
    for j in master:
        scale = np.max((V.loc[:,i].sum(),V.loc[:,j].sum()))
        S.loc[i,j] = np.dot(V.loc[:,i],V.loc[:,j])/scale

plt.figure(figsize=(12,8))
plt.matshow(S.values)
plt.xticks(range(len(master)),master,rotation=90)
plt.yticks(range(len(master)),master)
plt.colorbar()
plt.tight_layout()

import seaborn as sns
plt.figure(figsize=(10,12))
sns.heatmap(S.values,xticklabels=master,yticklabels=master,cmap='viridis')
plt.tight_layout()

s_u = np.tril(S.values,k=-1)
import torch
s_u = torch.from_numpy(s_u)
_,idx = torch.topk(s_u.reshape(-1),k=4)
idx = [[i//97,i%97] for i in idx]

top_idx = []
top_score = []
for i,j in idx:
    r,c = master[i.item()],master[j.item()]
    s = S.loc[r,c]
    top_idx.extend([r,c])
    top_score.append(s)
    print(f'专家的编码为 {r} {c}')
    print(f'相似度因子为 {s}')

top_S = S.loc[sorted(top_idx),sorted(top_idx)]

sns.heatmap(top_S.values,
            xticklabels=sorted(top_idx),
            yticklabels=sorted(top_idx),
            cmap='viridis')
plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
top_idx
np.round(top_score,5)
top_score
plt.hist(s_u.numpy().reshape(-1))

```

```
plt.xlabel('score',fontsize=20)
plt.ylabel('freq',fontsize=20)
plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.tight_layout()
```

问题二

图 5 专家原始分分布图

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
from scipy.stats import kendalltau
from scipy.stats import spearmanr
from scipy.stats import kurtosis,skew

%matplotlib inline
plt.rcParams['font.sans-serif']=['SimHei']
plt.rcParams['axes.unicode_minus']=False
warnings.filterwarnings('ignore')

prof_scores=pd.read_excel('专家评分分布.xlsx',sheet_name='五位专家')
print(prof_scores)
code=list(set(prof_scores.code.values))[:9]
print(code)
raw_scores=[]
std_scores=[]
for i in range(len(code)):
    raw_scores.append(prof_scores[prof_scores.code==code[i]].raw)
    std_scores.append(prof_scores[prof_scores.code==code[i]].stand)
print(raw_scores)
print(std_scores)
data=raw_scores
labels=code
# print(labels)
fig=plt.figure(figsize=(22,18))
for i in range(1,10):
    fig.add_subplot(3,3,i)
    plt.hist((data[i-1].values),bins=15,color=sns.color_palette("Pastel1",10)[i],density=True)
    data[i-1].plot(kind='kde',color='gray')
    plt.xlabel(labels[i-1],fontsize=18)
    plt.ylabel("频率",fontsize=18)
    plt.xticks(fontsize=16)
    plt.yticks(fontsize=16)
plt.show()
fig.savefig('专家原始打分.png',dpi=300)
```

图 8 专家打分均值直方图

```
data=[scores.avg,scores.std1,scores.end]
labels=['原始分','标准分','复议分']
print(labels)
fig=plt.figure(figsize=(6,5))
for i in [1]:
    fig.add_subplot(1,1,i)
    plt.hist((data[i-1].values),bins=35,color=sns.color_palette("Pastel1",10)[i],density=True)
    data[i-1].plot(kind='kde',color='gray')
    plt.xlabel(labels[i-1],fontsize=16)
    plt.ylabel("频率",fontsize=16)
    plt.xticks(fontsize=14)
    plt.yticks(fontsize=14)
plt.show()
fig.savefig('专家均值分布对比.png',dpi=300)
```

图 11 方案一秩差图

```
rank_data=pd.read_excel('专家评分分布.xlsx',sheet_name='排名')
print(rank_data)
rank_data['end_rank']=np.array(rank_data.end_score.rank(method='min',ascending=False),dtype=np.int)
print(rank_data)

rank_data['plan1_rank']=np.array(rank_data.plan1.rank(method='min',ascending=False),dtype=np.int)
rank_data['plan1_drunk']=rank_data.plan1_rank-rank_data.end_rank

plan1_color=['red' if rank_data.plan1_drunk[i]>0 else 'green' for i in range(len(rank_data.plan1_drunk))]
rank_data

num1=27 # 获奖人数
num2=352
x=list(range(len(rank_data.plan1_drunk)))
fig=plt.figure(figsize=(22,7))
fig.add_subplot(1,3,1)
plt.vlines(x=x[:num1],ymin=0,ymax=rank_data.plan1_drunk[:num1],color=rank_data.plan1_color[:num1],linewidth=5)
plt.ylabel('秩差距',fontsize=20) # 添加斜体字符
plt.xlabel('参赛队伍',fontsize=20)
plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.grid(linestyle='-.',alpha=0.7)
```

```

fig.add_subplot(1,3,2)
x=list(range(len(rank_data.plan1_drunk)))
plt.vlines(x=x[num1:num2],ymin=0,ymax=rank_data.plan1_drunk[num1:num2],color=rank_data.plan1_color[num1:num2],linewidth=5)
plt.ylabel('秩差距',fontsize=20) # 添加斜体字符
plt.xlabel('参赛队伍',fontsize=20)
plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.grid(linestyle='-.',alpha=0.7)

fig.add_subplot(1,3,3)
x=list(range(len(rank_data.plan1_drunk)))
plt.vlines(x=x[num2:],ymin=0,ymax=rank_data.plan1_drunk[num2:],color=rank_data.plan1_color[num2:],linewidth=5)
plt.ylabel('秩差距',fontsize=20) # 添加斜体字符
plt.xlabel('参赛队伍',fontsize=20)
plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.grid(linestyle='-.',alpha=0.7)
plt.tight_layout()
plt.show()

fig.savefig('方案1 秩差图 .png',dpi=300)

```

图 17 散点图矩阵

```

import seaborn as sns
import statsmodels.api as sm
lm_data=pd.read_excel('专家评分分布.xlsx',sheet_name='回归特征')
# print(lm_data)
data=lm_data.iloc[:28,1:5]
print(data)
# 使用 seaborn 绘制散点图矩阵
fig=plt.figure(figsize=(20,16))
g=sns.pairplot(data,kind='reg')
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.show()
g.savefig('散点图系数矩阵.png')

```

自定义标准分计算

```

data=[rank_data.plan1,rank_data.plan2,rank_data.plan3,rank_data.plan4,rank_data.plan6]
for d in data:
    kurt = kurtosis(d)

```

```

    skewness = skew(d)
    print("峰度: ", kurt)
    print("偏度: ", skewness)

raw_data=pd.read_excel('专家评分分布.xlsx',sheet_name='五位专家')
print(raw_data)
pr_dict={}
code_list=list(set(raw_data.code))
# code_list

for c in code_list:
    temp=raw_data.raw.values[raw_data.code==c].tolist()[ :102]
    pr_dict[c]=temp

df_raw=pd.DataFrame(pr_dict)
df_raw.to_csv('df_raw.csv')
raw_data['scores']=list(range(raw_data.shape[0]))
print(raw_data)
param_data=pd.read_excel('专家评分分布.xlsx',sheet_name='参数')
param_data.index=code_list
print(param_data)
for i,c in enumerate(code_list):
    params=param_data.iloc[i,:].values
#     print(params)

raw1_data=pd.read_excel('专家评分分布.xlsx',sheet_name='三位专家')
print(raw1_data)
pr_dict1={}
code_list1=list(set(raw1_data.code1))
print(len(code_list1))

for c in code_list1:
    temp=raw1_data.raw1.values[raw1_data.code1==c].tolist()[ :66]
    pr_dict1[c]=temp
# print(pr_dict1)
df_raw1=pd.DataFrame(pr_dict1)
df_raw1.to_csv('df_raw1.csv')

```

数据预处理以及评估指标计算—肯德尔 Tau_b

```

# Tau_b
def Tau_b(dat1,dat2,data_name):
    c = 0
    d = 0
    t_x = 0
    t_y = 0
    for i in range(len(dat1)):
        for j in range(i+1,len(dat1)):

```



```

        if (dat1[i]-dat1[j])*(dat2[i]-dat2[j])>0:
            c = c + 1
        elif (dat1[i]-dat1[j])*(dat2[i]-dat2[j])<0:
            d = d + 1
        else:
            if (dat1[i]-dat1[j])==0 and (dat2[i]-dat2[j])!=0:
                t_x = t_x + 1
            elif (dat1[i]-dat1[j])!=0 and (dat2[i]-dat2[j])==0:
                t_y = t_y + 1

    tau_b = (c - d) / np.sqrt((c+d+t_x)*(c+d+t_y))

    print('tau_b = {0}'.format(tau_b))
print(f'{data_name} kendalltau(dat1,dat2) = {kendalltau(dat1,dat2)}')

num0=28
dat1 = rank_data.end_rank.values.tolist()[ :num0]
dat2 = rank_data.plan1_rank.values.tolist()[ :num0]
dat1 = imp_data.end_rank.values.tolist()[ :num0]
dat2 = imp_data.rank4.values.tolist()[ :num0]
Tau_b(dat1,dat2,'改进的最大最小')
dat1 = imp_data.end_rank.values.tolist()[ :num0]
dat2 = imp_data.rank6.values.tolist()[ :num0]
Tau_b(dat1,dat2,'改进的减均值')

```

数据预处理以及评估指标计算—斯皮尔曼相关系数

```

def Sperman_rank(dat1,dat2,data_name):
#    print('tau_b = {0}'.format(tau_b))
    print(f'{data_name}          spearmanr(dat1,dat2)          =
{spearmanr(dat1,dat2)}')
    dat1 = rank_data.end_rank.values.tolist()[ :num0]
    dat2 = rank_data.plan1_rank.values.tolist()[ :num0]
    Sperman_rank(dat1,dat2,'plan1')
    dat1 = imp_data.end_rank.values.tolist()[ :num0]
    dat2 = imp_data.rank4.values.tolist()[ :num0]
    Sperman_rank(dat1,dat2,'改进的最大最小')
    dat1 = imp_data.end_rank.values.tolist()[ :num0]
    dat2 = imp_data.rank6.values.tolist()[ :num0]
    Sperman_rank(dat1,dat2,'改进的减均值')

```

图 18 方案六与方案七拟合效果图

```

import statsmodels.api as sm
import matplotlib.pyplot as plt
import pandas as pd

```

```

import numpy as np

X4 = data[['plan4','range1']]
X6 = data[['plan6','range1']]
y = data['end_score']
# 添加常数列作为回归模型的截距
X4 = sm.add_constant(X4)
X6 = sm.add_constant(X6)

model4 = sm.OLS(y, X4).fit()
model6 = sm.OLS(y, X6).fit()

# 计算预测值和置信区间
y_pred4 = model4.predict(X4)
y_pred6 = model6.predict(X6)
confidence4 = model4.get_prediction(X4).conf_int()
confidence6 = model6.get_prediction(X6).conf_int()
# 绘制拟合图
fig=plt.figure(figsize=(16,6))
fig.add_subplot(1,2,1)
plt.scatter(y,y_pred4, color='blue', label='原始数据点') # 绘制原始数据点
plt.plot([y.min(), y_pred4.max()], [y.min(), y_pred4.max()],
color='red') # 绘制回归线
# plt.fill_between([y.min(), y.max()], confidence4[:, 0],
confidence4[:, 1], color='gray', alpha=0.3, label='置信区间') # 绘制置信区间
# 添加置信度标注
# plt.text(X4['plan4'].min(), confidence4[:, 0].min()-0.5, '95%
Confidence', ha='left', va='top', color='gray')
plt.legend(fontsize=13)
plt.xlabel('y',fontsize=16)
plt.ylabel('y_pred6',fontsize=16)
plt.xticks(fontsize=13)
plt.yticks(fontsize=13)

fig.add_subplot(1,2,2)
plt.scatter(y,y_pred6, color='blue', label='原始数据点') # 绘制原始数据点
plt.plot([y.min(), y_pred6.max()], [y.min(), y_pred6.max()],
color='red') # 绘制回归线
# plt.fill_between([y.min(), y.max()], confidence4[:, 0],
confidence4[:, 1], color='gray', alpha=0.3, label='置信区间') # 绘制置信区间
# 添加置信度标注
# plt.text(X4['plan4'].min(), confidence4[:, 0].min()-0.5, '95%
Confidence', ha='left', va='top', color='gray')
plt.legend(fontsize=13)

```

```

plt.xlabel('y',fontsize=16)
plt.ylabel('y_pred6',fontsize=16)
plt.xticks(fontsize=13)
plt.yticks(fontsize=13)

plt.tight_layout()
plt.show()
fig.savefig('多因素最大最小拟合图.png')

```

图 19 方案六与方案七残差图

```

X4 = data[['plan4','range1']]
X6 = data[['plan6','range1']]
y = data['end_score']
# 添加常数列作为回归模型的截距
X4 = sm.add_constant(X4)
X6 = sm.add_constant(X6)

model4 = sm.OLS(y, X4).fit()
model6 = sm.OLS(y, X6).fit()

# 计算预测值和置信区间
y_pred4 = model4.predict(X4)
y_pred6 = model6.predict(X6)
# 计算残差
residuals4 = model4.resid
residuals6 = model6.resid
# 绘制拟合残差图
fig=plt.figure(figsize=(16,6))
fig.add_subplot(1,2,1)
plt.scatter(X4['plan4'], residuals4, color='green')
plt.axhline(y=0, color='red', linestyle='--') # 添加水平参考线
plt.xlabel('plan4',fontsize=16)
plt.ylabel('残差',fontsize=16)
plt.xticks(fontsize=13)
plt.yticks(fontsize=13)

fig.add_subplot(1,2,2)
plt.scatter(X6['plan6'], residuals6, color='green')
plt.axhline(y=0, color='red', linestyle='--') # 添加水平参考线
plt.xlabel('plan6',fontsize=16)
plt.ylabel('残差',fontsize=16)
plt.xticks(fontsize=13)
plt.yticks(fontsize=13)

plt.tight_layout()
plt.show()

```

```
fig.savefig('双因素最大最小残差图.png')
```

问题三

图 22 专家评阅作品数量（数据 2.2）

```
# 数据预处理
raw_data=pd.read_excel('问题三.xlsx',sheet_name='一阶段 2.2')
print(raw_data)

pr_dict={}
code_list=list(set(raw_data.code))
print(len(code_list))

for c in code_list:
    temp=raw_data.raw.values[raw_data.code==c].tolist()[ :121]
    pr_dict[c]=temp

df_raw=pd.DataFrame(pr_dict)
df_raw.to_csv('./data/p3_prof122.csv')
raw_data['scores']=list(range(raw_data.shape[0]))
print(raw_data)
param_data=pd.read_excel('问题三.xlsx',sheet_name='一阶段参数 2.2')
param_data.index=code_list
print(param_data)
for i,c in enumerate(code_list):
    params=param_data.iloc[i,:].values
#     print(params)

raw_data.loc[raw_data.code==c,'scores']=50+10*(raw_data.loc[raw_data
.code==c,'raw']-params[2])/(params[3]-params[2])*params[1]
df_scores=pd.DataFrame(raw_data)
df_scores.to_csv('./data/p3_score122.csv')
# 数据预处理
raw1_data=pd.read_excel('问题三.xlsx',sheet_name='二阶段 2.2')
print(raw1_data)

pr_dict={}
code_list=list(set(raw1_data.code))
print(len(code_list))

for c in code_list:
    temp=raw1_data.raw.values[raw1_data.code==c].tolist()[ :54]
    pr_dict[c]=temp

df_raw=pd.DataFrame(pr_dict)
df_raw.to_csv('./data/p3_prof222.csv')
raw1_data['scores']=list(range(raw1_data.shape[0]))
```

```

# print(raw1_data)
param_data=pd.read_excel('问题三.xlsx',sheet_name='二阶段参数 2.2')
param_data.index=code_list
print(param_data)
for i,c in enumerate(code_list):
    params=param_data.iloc[i,:].values
#     print(params)

raw1_data.loc[raw1_data.code==c,'scores']=50+10*(raw1_data.loc[raw1_
data.code==c,'raw']-params[2])/(params[3]-params[2])*params[1]
df_scores=pd.DataFrame(raw1_data)
df_scores.to_csv('./data/p3_score222.csv')

from collections import Counter

frequency
[dict(Counter(raw_data.code)),dict(Counter(raw1_data.code))]
labels = [list(frequency[0].keys()),list(frequency[1].keys())]
print(labels[0])
values = [list(frequency[0].values()),list(frequency[1].values())]
num0=10
num1=2
# print(idx)
xts=[raw_data.code.values[::num0],raw1_data.code.values[::num1]]
print(len(xts[0]))
xlabel=['专家编码','专家编码']
# print(labels)
fig=plt.figure(figsize=(12,5))
for i in [0,1]:
    fig.add_subplot(1,2,i+1)
#         plt.hist((data[i-1].values),bins=num_prof[i-
1],color=sns.color_palette("Pastel1",10)[i],density=False)
    plt.bar(labels[i],
values[i],color=sns.color_palette("Pastel1",10)[i+1])
    plt.xlabel(xlabel[i],fontsize=16)
    plt.ylabel("专家评阅作品个数",fontsize=16)
    plt.yticks(fontsize=10)
    if i==0:
        plt.ylim(100,125)
        plt.xticks(labels[0][::num0],xts[0],rotation=90,fontsize=8)
    else:
        plt.ylim(40,70)
        plt.xticks(labels[1][::num1],xts[1],rotation=90,fontsize=8)
plt.show()
fig.savefig('./data/专家评阅个数 2.2 .png',dpi=300)

```

图 23 不同方案排名秩差 (left:数据 2.1, right:数据 2.2)

```
change_score1=pd.read_excel('问题三.xlsx',sheet_name='排名 2.1')
```

```

print(change_score1[:4])
change_score2=pd.read_excel('问题三.xlsx',sheet_name='排名 2.2')
# print(change_score2)
change_score1['drank']=-
np.array(change_score1.mm_score.rank(method='min',ascending=False),d
type=np.int)+np.array(change_score1.score1.rank(method='min',ascendi
ng=False),dtype=np.int)
# print(change_score1.drank)
change_score2['drank']=-
np.array(change_score2.mm_score.rank(method='min',ascending=False),d
type=np.int)+np.array(change_score2.score1.rank(method='min',ascendi
ng=False),dtype=np.int)
# print(change_score2.drank)
up_color=sns.color_palette('YlOrRd_r')[:6:2] # 成绩与获奖对应三个排名的
颜色
# print(up_color)
dn_color=sns.color_palette('Greens')[:6:2]
# print(dn_color)

change1_color=[]
for i in range(len(change_score1.drank)):
    if change_score1.drank[i]>0:
        if change_score1.awards[i]=='一等奖':
            change1_color.append(up_color[0])
        elif change_score1.awards[i]=='二等奖':
            change1_color.append(up_color[1])
        else:
            change1_color.append(up_color[2])
    else:
        if change_score1.awards[i]=='一等奖':
            change1_color.append(up_color[0])
        elif change_score1.awards[i]=='二等奖':
            change1_color.append(up_color[1])
        else:
            change1_color.append(up_color[2])
# print(change1_color)

change2_color=[]
for i in range(len(change_score2.drank)):
    if change_score2.drank[i]>0:
        if change_score2.awards[i]=='一等奖':
            change2_color.append(up_color[0])
        elif change_score2.awards[i]=='二等奖':
            change2_color.append(up_color[1])
        else:
            change2_color.append(up_color[2])
    else:
        if change_score2.awards[i]=='一等奖':

```

```

        change2_color.append(up_color[0])
    elif change_score2.awards[i]=='二等奖':
        change2_color.append(up_color[1])
    else:
        change2_color.append(up_color[2])
# print(change2_color)

colors=[change1_color,change2_color]
data=[change_score1.drunk,change_score2.drunk]
# 分数变化图
x=[list(range(len(change_score1.drunk))),list(range(len(change_score
2.drunk)))]
fig=plt.figure(figsize=(20,8))
fig.add_subplot(1,2,1)
plt.vlines(x=x[0],ymin=0,ymax=data[0],color=colors[0],linewidth=5)
plt.axhline(y=0, color='black', linestyle='--')
plt.ylabel('秩差距',fontsize=20) # 添加斜体字符
plt.xlabel('参赛队伍',fontsize=20)
plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.grid(linestyle='-.',alpha=0.7)

fig.add_subplot(1,2,2)
plt.vlines(x=x[1],ymin=0,ymax=data[1],color=colors[1],linewidth=5)
plt.axhline(y=0, color='black', linestyle='--')
plt.ylabel('秩差距',fontsize=20) # 添加斜体字符
plt.xlabel('参赛队伍',fontsize=20)
plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.grid(linestyle='-.',alpha=0.7)
plt.tight_layout()
plt.show()

fig.savefig('./data/分类秩差图.png',dpi=300)

```

极差模型与坡度图

```

range_data=pd.read_excel('问题三.xlsx',sheet_name='极差 2.1')
# print(range_data[:4])
r1_data=range_data[range_data.cls==0]
r2_data=range_data[range_data.cls==1]
# r1_data
r2_data

def loss(r1,r2): # 目标函数定义为欧氏距离
    return np.linalg.norm(r1 - r2)

point1 = np.array([1, 2, 3])

```

```
point2 = np.array([4, 5, 6])
loss(point1,point2)
```

```
def get_theta(start,end,delta,lap):
    # break_points: 断点个数
    num=(end-start)//delta+1
    param_theta=[]
    s=start
    while s<end:
        param_theta.append([s-lap,s+delta+lap])
        s=s+delta
    return param_theta
get_theta(20,40,2,0.5)
```

```
def get_theta(start,end,break_points,lap,r1):
    # break_points: 断点个数
    num=break_points+1
    delta=(end-start)/num
    param_theta=[]
    thetas=[[start+i*delta,start+(i+1)*delta] for i in range(num)]
    num_list=[]
    for th in thetas:
        num_list.append(len(r1[(r1>th[0]) & (r1<th[1])]))
    ratio=1/np.array(num_list) # 均匀区间的倒数比例（未来使得区间内部的
    极差个数匀称）
    ratio=ratio/(sum(ratio))
    length=ratio*(end-start)
    s=start
    for l in length:
        param_theta.append([round(s-lap,2),round(s+l+lap,2)])
        s=s+l
    return param_theta
get_theta(20,40,3,0.5,r1_data)
```

```
def range_model(start,end,break_points,lap,r1_data,r2_data):
    """
    start: 普遍调整偏差的起始点
    end: 偏差最大值
    delta: 调整区间大小
    lap: 区间重叠率（考虑边界交叉重叠影响）
    """
    r1_data['range_pred']=r1_data.range1
    param_theta=get_theta(start,end,break_points,lap,r1_data)
    weights=[]
    for theta in param_theta:
        # print(f'theta: {theta}')
```



```

    r1=r1_data.range1.values
    r2=r2_data.range1.values
    c1=r1_data.cls.values
    c2=r2_data.cls.values
    r11=r1[(r1>theta[0]) & (r1<theta[1])]
    r22=r2[(r1>theta[0]) & (r1<theta[1])]
    c11=c1[(r1>theta[0]) & (r1<theta[1])]
    c22=c2[(r1>theta[0]) & (r1<theta[1])]
    w=np.mean(np.array(r22)/np.array(r11))
    weights.append(round(w,3))
    r11_pred=np.array(r11)*w
    r1_data.loc[(r1>theta[0]) &
(r1<theta[1]),'range_pred']=r11_pred
    obj=loss(r1_data.range_pred.values,r2_data.range1.values)
return obj,r1_data,param_theta,weights

```

```

param_theta=[[20,22],[22,24],[24,30],[30,40]]
fig=plt.figure(figsize=(20,15))
for i,theta in enumerate(param_theta):
    r1=r1_data.range1.values
    r2=r2_data.range1.values
    c1=r1_data.cls.values
    c2=r2_data.cls.values
    r11=r1[(r1>theta[0]) & (r1<theta[1])]
    r22=r2[(r1>theta[0]) & (r1<theta[1])]
    c11=c1[(r1>theta[0]) & (r1<theta[1])]
    c22=c2[(r1>theta[0]) & (r1<theta[1])]
    c=list(c1)+list(c2)
    r=list(r1)+list(r2)
    c111=list(c11)+list(c22)
    r111=list(r11)+list(r22)

    fig.add_subplot(2,2,i+1)
    plt.scatter(c,r,color='lightgray')
    plt.scatter(c111,r111,marker="*",s=50,color='red')
    plt.xticks([0,1],['复议前极差','复议后极差'],fontsize=20)
    plt.ylabel('极差',fontsize=20)
    plt.title(f'区间: {theta}',fontsize=20)
    for i in range(len(r11)):
        plt.plot([c11[i],c22[i]],[r11[i],r22[i]],color='orange')
    plt.tight_layout()
plt.show()
fig.savefig(f'./data/坡度图 theta.png')

# 网格搜索寻找最优模型
start=20
end=max(max(r1_data.range1.values),max(r2_data.range1.values))
obj_list=[]

```

```

best_obj=np.inf
best_data=r1_data
best_break_points=1
best_lap_ratio=0
best_theta=[]
best_weights=[]
for break_points in [2,3,4,5,6]:
for lap in [0.1,0.2,0.3,0.4,0.5]:
#for break_points in [5,6,7]: 二次搜索
#    for lap in [0.5,0.6,0.7]:
#for break_points in [5,6,7,8]: 三次搜索
#    for lap in [0.45,0.5,0.55,0.6]:

obj,r1_data,param_theta,weights=range_model(start,end,break_points,l
ap,r1_data,r2_data)
    obj_list.append(obj)
    if obj<best_obj:
        best_obj=obj
        best_data=r1_data
        best_break_points=break_points
        best_lap=lap
        best_theta=param_theta
        best_weights=weights

print(f'最优断点个数: {best_break_points}')
print(f'最优损失: {best_obj}')
print(f'最优区间重叠: {best_lap}')
print(f'区间划分: {best_theta}')
print(f'区间划分: {best_weights}')

```

图 31 拟合结果图

```

# param_theta=[[20,21],[21,22],[22,24],[24,30],[30,40]]
fig=plt.figure(figsize=(22,12))
for i,theta in enumerate(best_theta):
    r1=r1_data.range1.values
    r2=r2_data.range1.values
    c1=r1_data.cls.values
    c2=r2_data.cls.values
    r11=r1[(r1>theta[0]) & (r1<theta[1])]
    r22=r2[(r1>theta[0]) & (r1<theta[1])]
    c11=c1[(r1>theta[0]) & (r1<theta[1])]
    c22=c2[(r1>theta[0]) & (r1<theta[1])]
    c=list(c1)+list(c2)
    r=list(r1)+list(r2)
    c111=list(c11)+list(c22)
    r111=list(r11)+list(r22)

```

```

fig.add_subplot(2,3,i+1)
plt.scatter(c,r,color='lightgray')
plt.scatter(c111,r111,marker="*",s=50,color='red')
plt.xticks([0,1],['复议前极差','复议后极差'],fontsize=20)
plt.ylabel('极差',fontsize=20)
plt.title(f'区间: {theta}',fontsize=20)
for i in range(len(r11)):
    plt.plot([c11[i],c22[i]],[r11[i],r22[i]],color='orange')
plt.tight_layout()
plt.show()
fig.savefig(f'./data/坡度图 data2.1.png')

```