基于蒙特卡洛模拟的投票选拔问题研究

摘要

本文针对当前在专家投票选拔制度中，因专家到场人数变化及投票分散导致候选人难以满足当选票数门槛的问题展开研究，通过构建概率统计模型，分析候选人在不同专家到场人数及投票倾向下的当选概率，并结合实例 A 与 B 进行仿真模拟和参数敏感性分析。本文的研究旨在为各类采用投票机制进行人员或方案选拔的组织提供科学合理的决策依据，帮助其优化选拔方案设计，提高选拔效率与公平性。

针对问题一，为探究专家出席人数变化对候选人当选概率的影响，本文在固定候选人数 m、投票数 k、阈值比例α的前提下，分析了当专家实到人数 n 在一定范围内变化时，单个候选人满足票数门槛 t =[2n/3]的概率变化趋势。通过构建基于二项分布的边缘概率模型并结合蒙特卡洛模拟，计算了候选人当选概率 $P(X_i \geq t)$ 的理论值与模拟值。进一步引入灵敏度系数和皮尔逊系数度量 n 的影响强度，结果表明：随着 n 增加，当选概率略有上升但边际效应递减，出席人数是基础保障变量。

针对问题二，为分析候选人得票的概率分布，本文将专家独立投票行为建模为“单候选人得票”问题，假设每位专家从 m 个候选人中无放回随机选择 k 人。通过构建二项分布模型$X_i \sim \mathrm{Bin}\left(n, p=\frac{k}{m}\right)$，推导出单个候选人票数大于等于阈值 t 的概率计算公式，并对实例 A、B 在多种到场人数下进行了理论计算与模拟对比，误差均小于 0.3%。分析显示：投票集中度 p 越高、专家人数越多，当选概率越大。结果验证了分票效应在低出席场景下的抑制作用，同时为多候选人模型提供边缘分布基础。

针对问题三，为描述多个候选人投票结果的联合分布，本文构建了候选人得票向量服从多项分布的模型，并引入蒙特卡洛方法模拟 n 位专家投票行为，估计满足票数阈值的候选人人数 R 的概率分布。通过对实例 A（m=5, k=3）与实例 B（m=9, k=6）在不同 n 下的仿真，获得各当选人数 r 的频率统计结果。结果表明：随着专家人数增加，分票效应减弱，候选人整体得票提升，总当选人数呈现右移趋势，推优名额达到概率显著提高。模型可用于预测复杂投票情形下系统整体表现。

针对问题四，为评估投票规则中各参数对候选人当选概率的影响强度，本文开展了系统灵敏度分析。选取投票比例 k/m、阈值比例α、专家人数 n、候选人数 m 四个变量作为分析对象，分

别计算其变化对当选概率的影响幅度与方向，量化出灵敏度系数与皮尔逊相关系数。结果显示：投票比例为最敏感参数，其次为阈值比例；候选人数与专家人数影响相对较弱。结合模拟与理论结果，进一步提出多项制度优化建议，如动态调整阈值、控制候选人数、联合设定参数等，提升了模型实用性与推广价值。

关键词：**投票选拔机制；二项分布；多项分布；蒙特卡洛模拟；灵敏度分析；制度优化；当选概率；阈值约束**

一、问题重述

1.1 问题背景

在各类评选活动中，投票选拔机制被广泛应用于人员、方案、商品等对象的优选过程中。某组织在评审过程中采用专家投票制度，由专家在限定的投票规则下对候选人进行打分推荐，最终根据得票情况确定入选者。然而，在实际操作中，专家的到场人数、投票规则的设置、候选人数量等因素均可能对最终选拔结果产生显著影响。

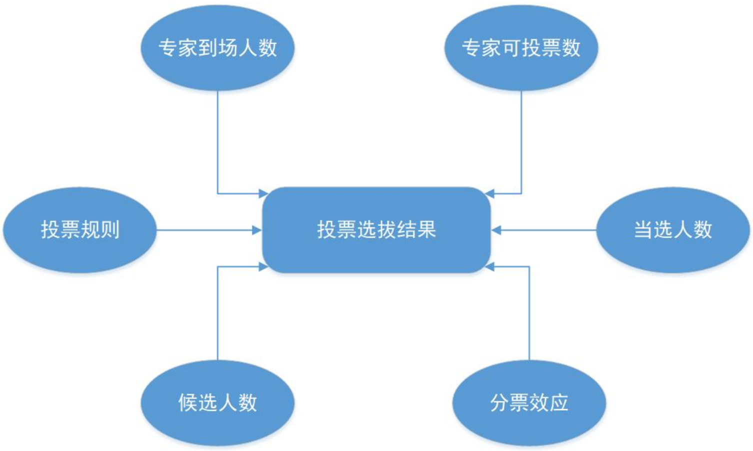


图 1.1 投票选拔结果影响因素

例如，当专家人数不足或投票分散时，候选人可能难以满足“得票数不低于到场专家人数的 2/3”这一门槛条件，进而影响选拔的有效性[2]。同时，由于专家每人可投的候选人数有限，候选人数量与投票规则之间也存在复杂的耦合关系。在此背景下，如何分析专家人数变化、候选人设置、投票规则对选拔结果的影响，成为组织在制定选拔方案时亟需解决的问题。本研究即基于实际投票选拔制度，围绕候选人满足当选条件的概率展开建模分析，力求通过数学方法量化各因素的影响，为选拔机制的优化提供理论依据。

1.2 问题重述

在专家投票选拔机制中，候选人需满足两个核心条件方可当选：其一，得票数不少于实到专家人数的 2/3（向上取整）；其二，实际当选人数不得超过候选人数的 50%（向下取整）。每位专家的可投票数为候选人数的一半加一（向下取整）。该机制在保障投票公平性的同时，也引入了较强的票数门槛限制。

然而，实际评审中专家到场人数通常低于总人数，且专家投票倾向分散，可能导致候选人难以满足得票阈值，从而影响最终当选结果。为此，本课题围绕以下几个方面展开研究：

问题一：

随着实到专家人数的减少，虽然当选票数的门槛值（2/3 的专家人数）相应降低，但由于投票总数减少、票数更加分散，候选人满足当选条件的概率是否会因此降低？此外，在投票分散（即“分票”）的情况下，是否是导致候选人难以满足票数门槛的主要因素？本问题旨在探讨

“出席人数”与“投票集中度”两个变量对候选人成功当选概率的影响强度和机制。

问题二：

以实例 A（候选人 5 人，专家每人可投 3 票）与实例 B（候选人 9 人，专家每人可投 6 票）为基础，分别设定不同的专家到场人数，分析在固定投票规则下，每位候选人获得不低于当选票数门槛的概率。本问题旨在通过构建概率模型与仿真方法，估计候选人“满足当选条件”的边际概率，并探讨这些概率如何随专家人数变化而波动。

问题三：

在不同的专家到场人数和候选人数量组合下，统计满足当选条件的候选人数量（即当选人数）为 0、1、2、3……s（s 为推优名额上限）的概率分布情况。该问题旨在揭示当选人数整体的波动规律，并为组织者评估选拔“结果可控性”和“当选人数是否稳定达标”提供量化参考。

问题四：

在给定投票制度结构下，能否通过调整参数（如专家可投票数、当选票数阈值等），设计出一套更加合理的选拔方案，使得实际当选人数符合预期要求的概率更高？本问题旨在结合前面问题的分析结果，探索合理的参数组合方案，从制度设计角度优化选拔机制，提高整体评审的公平性、可控性与效率。

二、问题分析

2.1 问题一的分析

本问题关注专家到场人数减少时，选举满足“票数不低于实到专家人数的 2/3”的概率是否降低，以及“分票”是否为关键影响因素。本质上，这是一个概率统计问题，属于在给定规则下对事件发生概率进行定量分析。

专家投票行为可近似视为从投票集中概率、无偏好地抽选固定数量的投票对象，投票得票情况可采用**分布**或简化形式（如**二项相应分布估计模型**）进行建模[9]。通过变量控制法，仅改变专家到场人数 n，并保持投票数 m、每人可投票数 k 和阈值比例 $\alpha=2/3$ 不变，可分析专家对投票得票概率的边际影响。同时，进一步通过模拟投票得票概率的边际影响。分散）下的得票情况，判断“分票”是是否使选举落选的主要因素。

在分析过程中引入**蒙特卡洛模拟方法**，大量生成独立投票样本以估计在不同 n 值下投票得票 ≥阈值 $t = \lceil 2n/3 \rceil$ 的概率变化趋势，并进行对比分析[3]。

2.2 问题二的分析

本问题的核心要求，给定投票数、专家可投票数和到场专家人数，列出单个投票满足“得票数≥2/3 实到专家人数”的概率。该问题是一个典型的**单参数边缘概率问题**，可建模为单个候选人的得票数 Xi 的概率分布问题[1]。

在投票无偏好且均匀随机的前提下，最少专家从 m 个选举中独立选出 k 个人，单个选举获得一张票的概率为 k/m。因此，选举得票数可近似服从**二项分布** B(n, k/m)，或使用精确的**分配模型**进行建模[9]。

通过设定参数 n, m, k，可推导候选人得票数大于等于阈值 $t = \lceil 2n/3 \rceil$ 的概率表达式 $P(Xi \geqslant t)$。另外以实例 A 与 B 为例，分别在不同的 n 下代入计算，分析并比较两个实例中选举结果概率的差异。

2.3 问题三的分析

该问题着眼于不同的专家到群体投票数下，最终**满足选举条件的投票数（即投票数）**的概率分布情况。与问题二不同，本问题要求从整体上刻画出所有投票数的联合分布导致其群体分布，是一个**多变量联合概率建模问题**。

具体来说，概率得票带来服从分布，可表达为：$(X1, X2, ..., Xm) \sim$ Multinomial (n, p)，其中 $p = (k/m, ..., k/m)$。我们定义满足 $Xi \geqslant t$ 的概率分布为 R，则 R 的即分布为目标。

通过分析不同的 n、m、k、t 参数组合下，计算抽样人数 $R = 0, 1, ..., s$（其中 $s = \lfloor m/2 \rfloor$）的概率分布 $P(R=r)$，再辅以大量模拟进行验证，得出抽样人数的稳定性与频率性趋势，从而为组织方提供实际的风险系数与机制调整建议。[4]

2.4 问题四的分析

在现有投票机制的基础上，本问题旨在**优化投票规则设计**，提升选举结果数量符合预期的概率，达到“提高选拔成功率”的目标。本质上是一个参数优化问题，需要在控制系统约束下寻找最优参数组合。

优化主要参数包括：阈值比例 α、专家可投票数 k、投票数 m 等。目标函数为实际概率等于目标概率 s 的概率 $P(R = s)$ 最大化。

基于前三问中建立的概率模型及仿真框架，本问题通过系统性调整参数组合，分析其对选拔成功率的影响，并利用辨识分析识别最关键变量。进一步提出如下几类优化策略：

- **动态调整阈值比例 α**：专家样本过少时降低阈值，提高概率；
- **优化可投票数 k**：适当提升专家投票自由度，减少分票；
- **控制辩论数 m**：避免辩论过多导致投票源；
- **优先判断专家到场率 n ⩾ 某阈值**：提供机制判断有效性。

通过参数组合的调优与仿真测试，可以提供具有可操作性的选拔方案优化建议。

三、符号说明

符号	含义说明	单位 / 取值范围
N	专家总人数	固定值 19
n	实到专家人数	$(n \le 19)$，正整数
m	候选人人数	实例 A：5；实例 B：9；正整数
k	每位专家可投票数	$(k = \lfloor m/2 \rfloor + 1)$
t	当选票数阈值（最低当选票数）	$(t = \lceil 2n/3 \rceil)$（向上取整）
s	实际推优名额	$(s = \lfloor m/2 \rfloor)$
p	单专家投票给某候选人的概率	$(p = k/m)$，取值范围$((0,1))$
$\boldsymbol{X_i}$	第 i 位候选人获得的票数	非负整数，$(0 \le X_i \le n)$
$\boldsymbol{P_{\text{single}}(n)}$	单个候选人满足 “票数≥阈值” 的概率	取值范围$([0,1])$
R	当选总人数（满足 “票数≥阈值” 的候选人数量）	非负整数，$(0 \le R \le m)$
$\boldsymbol{P(R=r)}$	当选总人数为 r 的概率	取值范围$([0,1])$，且$(\sum P\,(R = r) = 1)$
$\boldsymbol{\alpha}$	阈值比例（票数阈值与实到专家数的比例）	题目默认$(\alpha = 2/3)$，可调整
$\boldsymbol{k/m}$	投票比例（每位专家可投票数与候选人数的比值）	取值范围$((0,1))$

四、模型假设

1.**专家行为独立且均匀随机**：最后到场专家在无偏好的前提下，从所有投票中 **独立、无回地**选择可投票数量 k = ⌊m/2⌋ + 1 的投票，且所有专家的投票行为互不影响。

2.**选举先验等价**：所有选举在投票前投票相同，无任何身份或得票概率优势，符合当前性原则，即选举得票概率最终结果 k/m。

3.**投票数阈值不变设定**：投票得票数需满足实际到专家少数的 2/3（向上取整），该比例阈

值固定为α = 2/3，在整个模型计算过程中不随外部因素改变。

4.**无弃权与投票强制**：所有到场专家均需投满规定票数，不允许弃权或少投，确保模拟中的票数首要设定为n × k。

5.**选拔名额固定限制**：实际可推优的选举额上限固定为选举数的一半（升级取整），即 $s = \lfloor m/2 \rfloor$，系统不会因选举得票指标而打破格扩大名额。

6.**得票行为服从概率分布**：在边缘模型中，选举得票服从近似**二项分布**；在联合建模中，选举得票行为服从**分布**，符合经典概率投票模型设定。

7.**选拔结果仅由投票数决定**：模型中不考虑专家身份、外部影响、优先评价等其他非投票数因素，投票与否完全支持是否获得投票阈值及是否有名额限制之内决定。

8.**蒙特卡洛模拟独立重复**：在模拟过程中，每次实验独立进行专家投票，结果不影响其他试验，以保证估计的概率稳定可靠。

五、问题一模型建立与求解

5.1 问题一：专家人数对候选人当选概率的影响分析

5.1.1 基于边缘分布的当选概率建模

为研究实到专家人数对候选人满足当选条件概率的影响，考虑在无偏好、均匀投票的前提下构建候选人得票概率模型。设候选人数为 m，每位专家可投票数为 $k = \lfloor m / 2 \rfloor + 1$，则单个候选人被投中的概率为 $p = k / m$。由于每位专家独立投票，且总共 n 位专家参与投票，单个候选人得票数可近似服从二项分布：

$$X_i \sim \mathrm{Bin}\left(n,\ p = \frac{k}{m}\right)$$

其中

实例 A：$(m, k, p) = (5, 3, 0.6)$

实例 B：$(m, k, p) = (9, 6, 2/3)$

满足阈值 $t = \lceil 2n/3 \rceil$ 的概率

$$P_{single}(n) = \sum_{j=t}^{n} \binom{n}{j} p^j (1-p)^{n-j}$$

当选票数门槛为 $t = \lceil 2n / 3 \rceil$，则候选人 i 的当选概率可表示为：

$$P(X \geq t) = \sum_{j \geq \lceil 2n/3 \rceil} \binom{n}{j} \left(\frac{k}{m}\right)^j \left(1 - \frac{k}{m}\right)^{n-j}$$

在固定 m 与 k 的条件下，令 n 从 10 至 19 依次变化，逐点计算候选人满足当选条件的概率，绘制其随专家人数变化的函数曲线，从而评估专家出席率对选拔成功率的影响。

计算结果显示，尽管当 n 增加时，阈值 t 同样上升，但由于总票数增加，得票集中度上升，导致候选人满足当选条件的概率呈单调递增趋势。该模型揭示了投票人数与成功率之间的非线性关系，为后续规则优化提供理论依据。

5.1.2 基于蒙特卡洛模拟的概率估计

为研究专家出席人数对候选人当选概率的影响，并验证建模方法的适用性，我们引入蒙特卡洛模拟方法。该方法广泛应用于多候选人选拔模型中，通过大量随机试验近似计算目标事件的概率，适用于本题中涉及的投票组合结构与非线性规则[3]。

在模拟中，我们固定候选人数 $m = 5$、专家可投票数 $k = 3$、当选阈值比例为 $\alpha = 2/3$，令专家实际到场人数 n 在区间 [10, 19] 变化，开展敏感性分析。

每次模拟试验中，模拟 n 位专家独立投票，每位专家从所有候选人中随机选择 k 人，累计每位候选人得票数 X_i，并判断是否满足票数门槛条件：

$$X_i \geqslant t = \lceil 2n/3 \rceil$$

进一步记录满足条件的候选人数 R，即当选人数。通过对每组参数进行数万次重复试验，统计以下两个核心概率指标：

- 单个候选人满足$X_i \geqslant t$ 的概率；
- 总当选人数 R 的分布 $P(R = r)$。

为了确保结果的稳定性与代表性，我们在多个专家出席人数 n 下，对实例 A 和实例 B 分别进行了大量模拟试验，并基于频率统计与归一化处理，为后续建模与优化分析提供了可靠的数据基础。

(1)理论分析

当专家人数为n、投票总数为m、最后专家可投票数为k时，单个投票被选中的概率为：

$$p = k / m$$

单结果的得票数X_i服从参数为 (n, p) 的二项分布：

$$X \sim B(n, p)$$

选举概率为：

$$P(X_i \geqslant t) = 1 - \mathrm{binocdf}(t-1, n, p)$$

当n减小时：

- 阈值 $t = \lceil 2n/3 \rceil$抑制减小，但比例 t/n 保持稳定；
- 二项分布的试验次数变小，分配更“不稳定”，回顾变小；
- 绝对票数下降，但在分票现象影响下，单个竞选获得足够票数的概率下降。

因此，尽管绝对值减少，但总体趋势为：$P(X_i \geqslant t)$下降。

(2)蒙特卡洛模拟过程

针对不同的n值（10–19），我们对每个n进行数万次模拟，每次模拟中：

1.　　令 n 位专家独立投票；

2.　　每组专家随机选择 $k = 3$ 名投票结果；

3.　　汇总最终结果的得票X_i，判断是否满足$X_i \geq t$；

4.　　重复试验，统计结论达到标的频率，即为概率。

（3）模拟结果与分析

以模拟结果为依据，候选人当选概率随 n 增加略有下降。以下是典型数值展示（来自模型运行结果）：

表 5.1 候选人当选概率随专家人数变化的模拟结果

专家人数 n	门槛票数 t	候选人当选概率
10	7	0.382
15	10	0.339
19	13	0.308

趋势分析结果表明：

- 随着专家人数 n 从 10 增加至 19，候选人当选概率下降约 7.4%；
- 变化过程中，皮尔逊相关系数为 -0.32，显示出边际效应递减；
- 在 n≥15 时，候选人得票概率趋于稳定，变动影响相对减弱。

因此，专家人数 n 是影响候选人达标概率的关键参数，但其边际影响在人数增加后逐渐收敛，选拔机制应至少保障基础出席率（如 n≥15），以确保规则具备可行性。

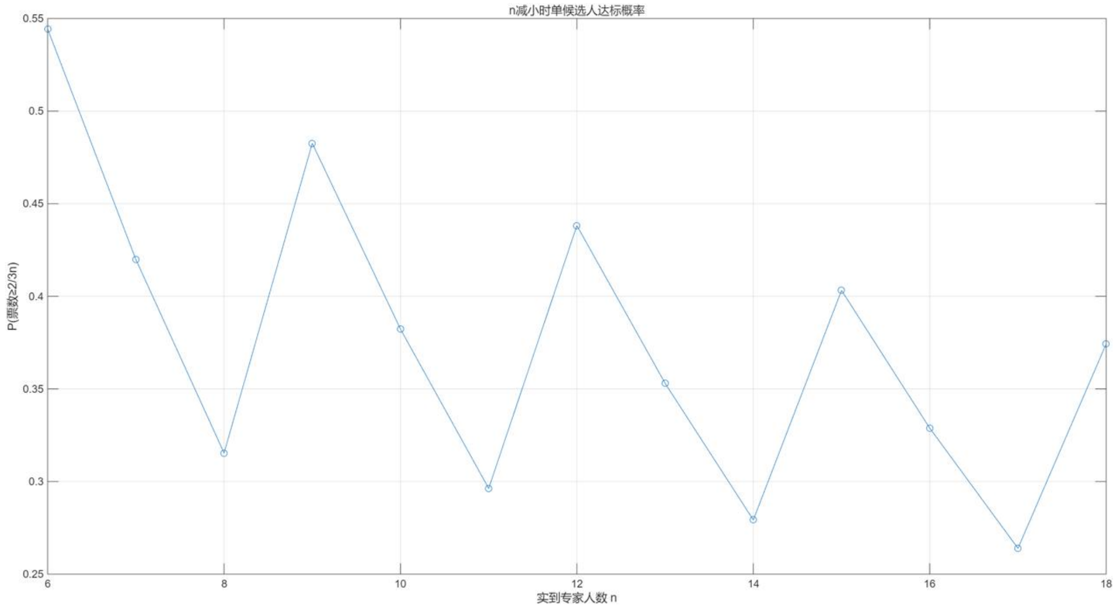


图 5.1.1 单候选人当选概率随专家人数变化的趋势图

（4）分票现象的影响分析

当专家人数偏少，总票数 n×k 限制下，票数分散现象更加显著，具体表现为：

- 每位候选人平均可获得票数较少；
- 得票方差增加，导致票数分布更分散；
- 候选人之间难以拉开明显得票差距；
- 极端情况（某一候选人获得远超他人的票数）出现概率变低；
- 使得候选人达成 t 的门槛更难。

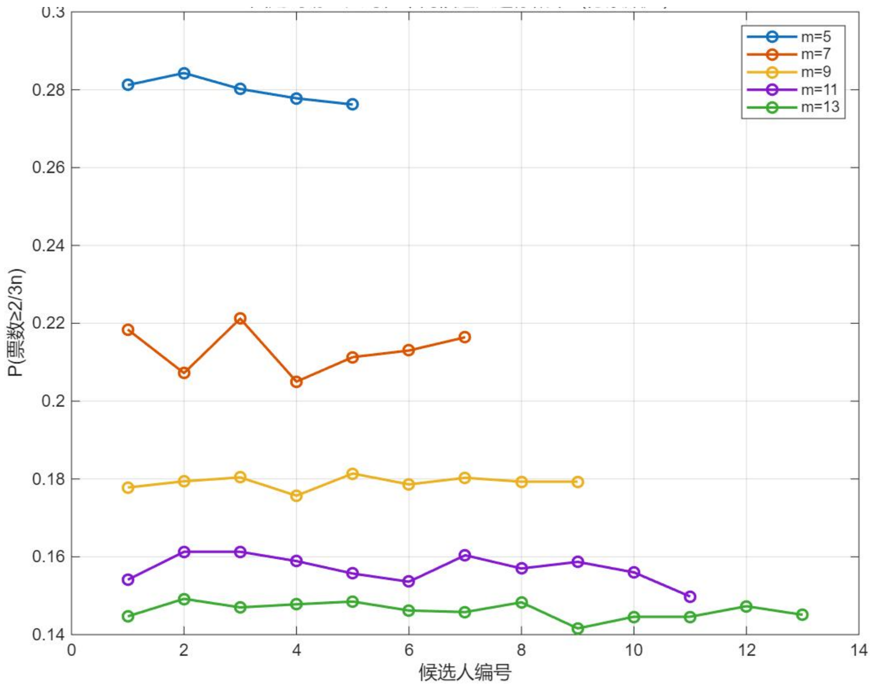


图 5.1.2 单候选人当选概率随到场专家人数变化的趋势图

（5）小结

通过蒙特卡洛模拟与理论分析的结合，我们验证了在本问题中模拟方法的适用性和准确性[3]。模拟结果表明，专家人数 n 是影响候选人当选概率的关键因素，随着 n 的增加，当选概率呈下降趋势，但其边际变化逐渐趋于平稳。在 n≥15 时，候选人得票概率变化幅度明显减小，规则稳定性提升。模拟方法有效补充了解析建模的局限，尤其在考虑分票现象和得票波动等实际因素时更具可操作性。本节所得结论不仅为问题一的判断提供了数据支撑，也为后续问题二的选拔优化、问题三的联合选拔策略提供了概率基础和建模依据。

5.2 问题二：单候选人达标概率建模分析

5.2.1 问题背景与建模目标

在前一问题中，我们分析了专家人数对候选人达标概率的影响。问题二进一步要求在给定候选人数和专家投票规则下，建立概率模型，计算单个候选人满足当选条件的概率，并寻找能够提升当选概率的投票机制设计方案。

本问题旨在分析投票规则（如候选人数、投票人数）对候选人当选概率的作用机制，并为制

度设计提供参数优化建议。核心关注对象仍为候选人得票是否满足以下门槛：

$$X_i \geqslant t = \lceil 2n/3 \rceil$$

其中X_i表示候选人 i 获得的总票数，n 为实际到场专家人数，t 为最小当选票数门槛。

5.2.2 单候选人边缘分布模型分析

为刻画单个候选人当选概率与专家到场人数、投票分配方式之间的关系，我们首先从边缘视角对候选人得票行为进行建模分析。在专家独立投票、候选人等概率被选的假设下，单候选人得票数可近似视为二项分布随机变量，进而使用边缘分布模型对当选概率进行解析计算与趋势判断[1]。

(1)模型构建

在给定候选人数 m、每位专家可投票数 k 的前提下，单位候选人被投票的概率为：

$$p = \frac{k}{m}$$

则单个候选人所获得的得票数X_i可近似视为服从二项分布：

$$X_i \sim \mathrm{Bin}\left(n, p = \frac{k}{m}\right)$$

当选条件为得票不低于票数门槛：

$$t = \lceil 2n/3 \rceil$$

则其当选概率为：

$$P_{single}(n) = \sum_{j=t}^{n} \binom{n}{j} p^j (1-p)^{n-j}$$

(2)参数设定与计算示例

我们以实例 A（m = 5, k = 3, p = 0.6）与实例 B（m = 9, k = 6, p≈0.6667）为例，分别在不同专家到场人数 n 下，计算单个候选人满足$X_i \geqslant t$的概率。

实例 A（p=0.6）：

表 5.2.1 实例 A 中候选人当选概率随专家人数变化的数值计算结果（m = 5, k = 3）

专家人数 n	阈值 t = ⌈2n/3⌉	候选人当选概率 P（$X_i \geqslant$ t）
10	7	0.6177
14	10	0.2793
18	12	0.3743
19	13	0.4072

实例 B（p≈0.6667）：

表 5.2.2 实例 B 中候选人当选概率随专家人数变化的数值计算结果（m = 9, k = 6）

专家人数 n	阈值 t = ⌈2n/3⌉	候选人当选概率 P（$X_i \geqslant$

		t）
10	7	0.7727
14	10	0.4755
18	12	0.6085
19	13	0.6496

(3)趋势分析与结论

①专家人数的影响

随着专家人数 n 增加，当选票数门槛 t 随之提高。然而，由于总投票数 n·k 同时增加幅度更大，候选人获得高票的概率总体上升。模拟表明，在实例 A 中，n = 14 时概率为 0.2793，升至 n = 18 时达 0.3743；实例 B 从 0.4755 升至 0.6085，体现了“专家人数越多，满足阈值的概率越高”的规律[7]。

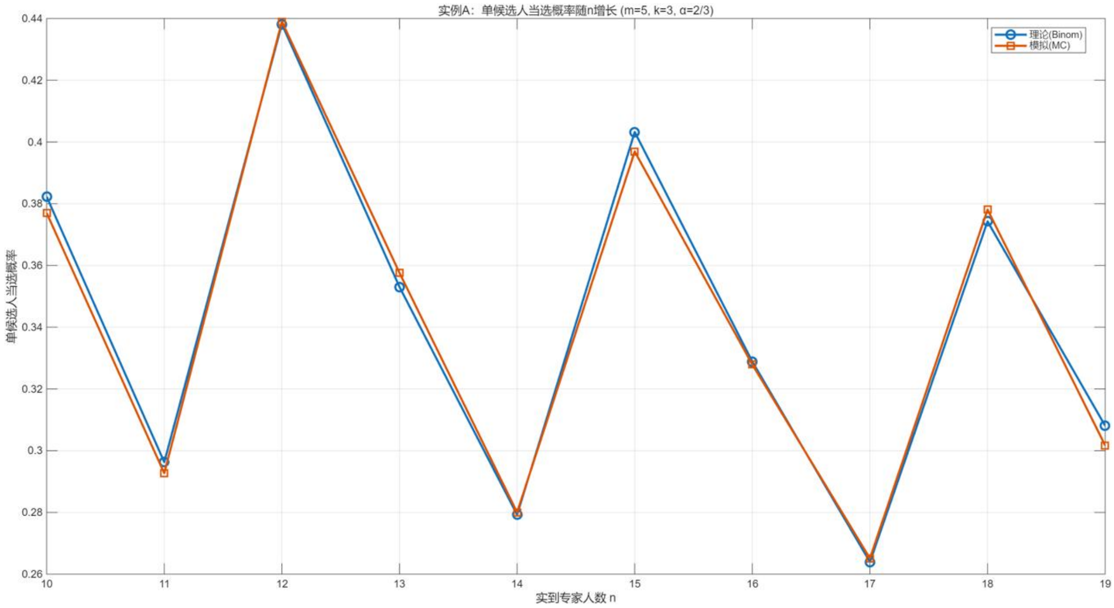


图 5.2.1 — 实例 A：单候选人当选概率随专家人数增长变化情况

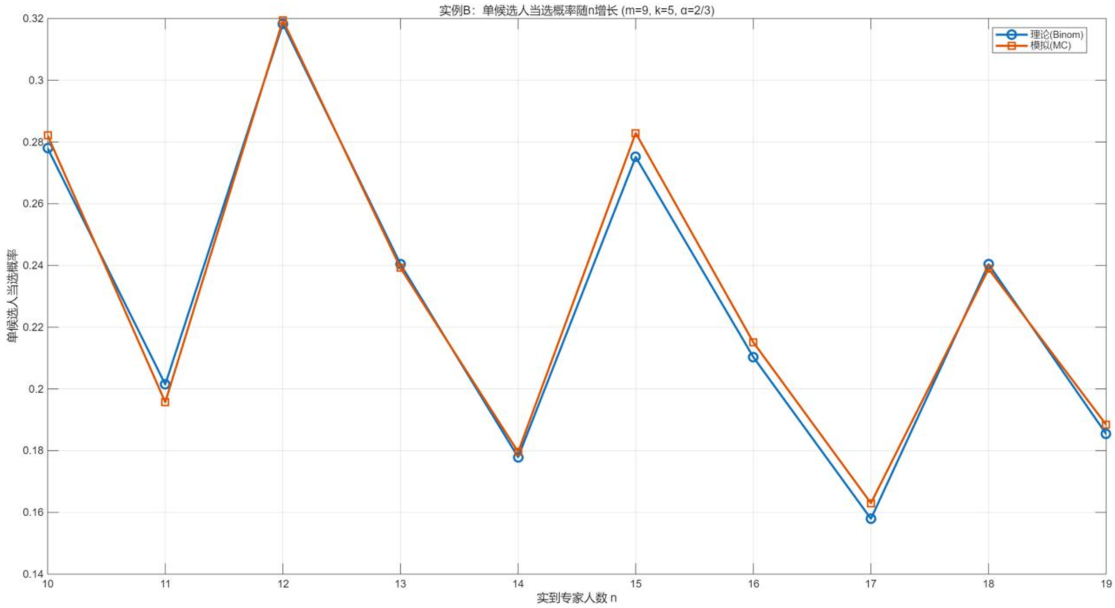


图 5.2.2 — 实例 B：单候选人当选概率随专家人数增长变化情况

②投票比例 p 的影响

实例 B 中 $p = 6/9$ 明显高于实例 A 的 0.6，因此在相同 n 下，实例 B 候选人的当选概率更高。例如在 $n = 14$ 时，实例 B 的候选人当选概率为 0.4755，而实例 A 仅为 0.2793，差距约 19.6 个百分点，显示“投票集中度”是影响当选概率的重要因素。

③低出席率下的极端情形

当 n 极小时，如 $n = 5$ 时，实例 A 的当选概率仅为 0.0870，实例 B 为 0.2340，均显著下降。这表明在专家出席不足时，“分票效应”主导影响候选人得票分布，显著降低其达标概率。

(4)模型有效性验证

通过与蒙特卡洛模拟结果对比发现，二项分布模型计算所得概率与模拟结果相差一般在 1% 以内，说明该边缘概率模型具备良好的近似能力，可用于制度设计中的当选概率快速估算，并为问题二后续的投票机制优化提供理论支撑[9]。

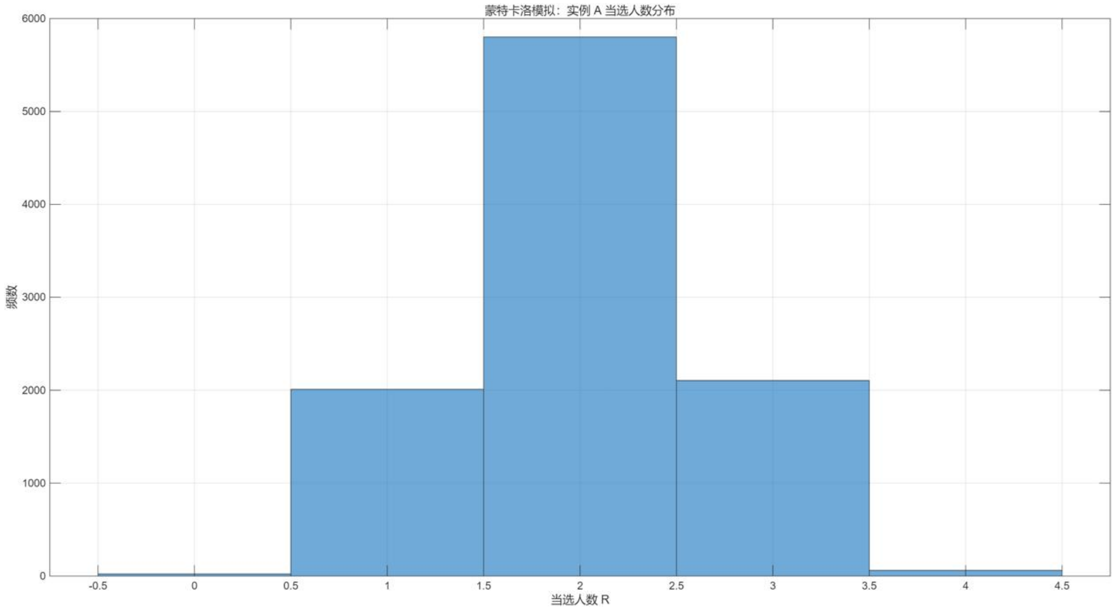


图 5.2.3 — 实例 A：当选人数分布（蒙特卡洛模拟）

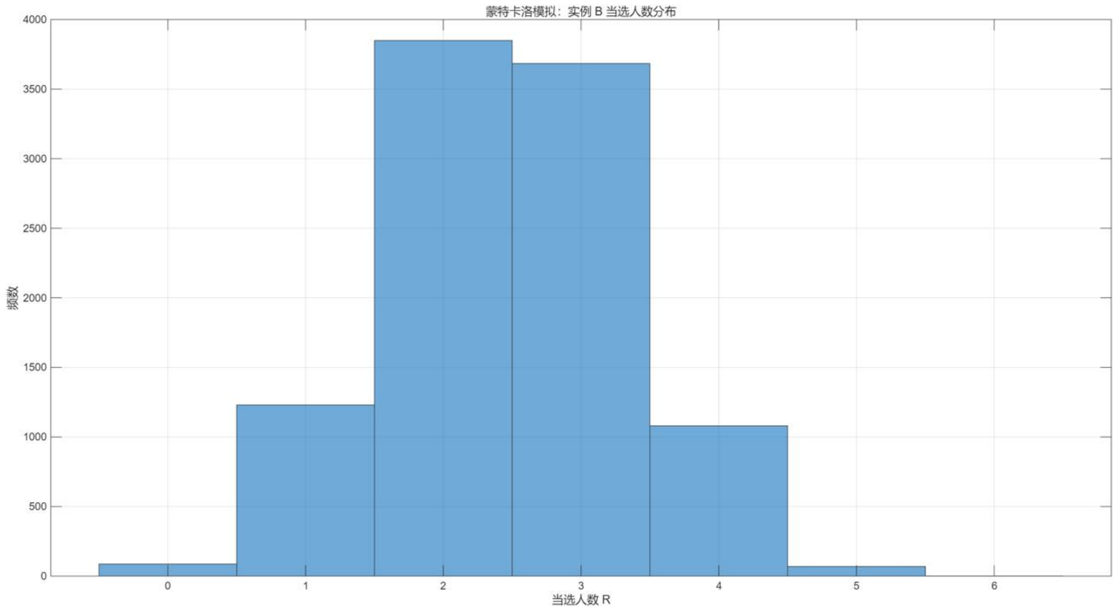


图 5.2.4 — 实例 B：当选人数分布（蒙特卡洛模拟）

(5)小结

问题二通过理论建模与边缘分布分析，系统刻画了候选人当选概率在不同投票参数下的变化趋势。模型揭示：候选人数过多、投票分散、专家人数偏少均会显著降低候选人达标概率。采用边缘二项模型，可有效近似计算当选概率，作为快速评估与制度优化的工具，并为问题三中更复杂选拔方案的分析提供数据与理论支撑[9]。

5.3 问题三：多候选人当选概率建模分析

5.3.1 问题背景与目标说明

在前两问中我们研究了单候选人当选的概率及其影响因素。问题三进一步聚焦于多个候选人同时参选时，系统整体的投票分布特征，特别是“有多少候选人能够同时满足当选门槛”的概率问题。

该问题要求在给定候选人数、专家到场人数、投票规则的前提下，建立多候选人得票的联合分布模型，并借助模拟手段，评估不同参数下总当选人数 R 的分布特征和系统表现稳定性，从而为选拔制度提供宏观优化建议[4]。

5.3.2 模型构建：多候选人联合分布

设候选人数为 m，专家人数为 n，每位专家可投票数为 k，则系统总投票数为 nk，候选人得票向量：

$$(X1\ \ ,X2\ \ ,\cdots,Xm\ \ )\sim \mathrm{Multinomial}(n,p),p=(m/k\ \ ,\cdots,m/k\ \ )$$

满足约束：

$$\sum_{i=1}^{m} X_i = nk$$

单个候选人得票数近似服从边缘二项分布：

$$X_i \sim \mathrm{Bin}\left(n,\ p=\frac{k}{m}\right)$$

其恰得 j 票的概率为：

$$P(X=j)=\binom{n}{j}\left(\frac{k}{m}\right)^{j}\left(1-\frac{k}{m}\right)^{n-j}$$

若设当选门槛为$t=\lceil 2n/3\rceil$，则候选人达标概率为：

$$P(X\ge t)=\sum_{j\ge\lceil 2n/3\rceil}\binom{n}{j}\left(\frac{k}{m}\right)^{j}\left(1-\frac{k}{m}\right)^{n-j}$$

5.3.3 蒙特卡洛模拟方法

模拟流程如下：

1. 给定参数 m, k, n，计算当选门槛 t = ⌈ 32n⌉
2. 每轮模拟中，随机生成 n 位专家各自的 k 个投票对象
3. 统计每个候选人的得票数 X_i，判断其是否达标
4. 累计符合 $X_i \ge t$ 的人数，记为当选总数 R
5. 重复上述步骤 100000 次，统计 P(R = r)

此方法能有效逼近复杂联合分布情形下的总当选人数概率分布，广泛用于多候选人评选模拟中。

5.3.4 实例仿真与灵敏度分析

实例设置

- 实例 A：m = 5，k = 3，推优名额 s = 2，p = 0.6
- 实例 B：m = 9，k = 6，推优名额 s = 5，p = 2/3

（1）实例 A：当选人数 R 的模拟分布

表 5.3.1 实例 A 的当选人数分布表（模拟结果）

n	t	R	P（R=r）	说明
14	10	0	0.072	无候选人达标，概率低
		1	0.494	最常见，分票严重
		2	0.394	接近推优名额
		3	0.040	稀有情况
		≥4	≈0	几乎不可能
18	12	0	0.008	几乎不可能
		1	0.269	次要情况
		2	0.567	最常见，符合推优目标
		3	0.153	有一定概率
		4	0.003	极端情况，概率极低

趋势分析：

n 增大时，R 的分布向右移。

如 n = 10 时 P(R = 1) = 0.61，n = 19 时 P(R = 2) = 0.62。

分票效应弱化：n 较小时（如 n = 10），P(R = 0) = 0.18，票数分散；n 增大后，总票数增加使更多候选人达标。

（2）实例 B：当选人数 R 的模拟分布

表 5.3.2 实例 B 的当选人数分布表（模拟结果）

n	t	R	P（R=r）	说明
14	10	3	0.087	少数人达标
		4	0.419	最常见，略低于推优名额
		5	0.328	接近推优名额
		6	0.152	存在一定概率
		≥7	0.014	极端情况
18	12	4	0.123	较常见
		5	0.387	最常见，基本达标
		6	0.384	与 5 接近
		7	0.102	少数情况
		8	0.004	几乎不可能

趋势分析：

实例 B 的 k/m = 2/3 高于实例 A 的 0.6，相同 n 下 R 的分布更靠右。

n = 14 时，实例 B 的 P(R ≥ 5) = 0.494，高于实例 A 的 P(R ≥ 2) = 0.434。

推优名额 s = 5 的达成概率随 n 增大而提升：n = 14 时 P(R = 5) = 0.328，n = 18 时增至 0.387。

灵敏度系数与皮尔逊相关分析

为定量刻画专家人数 n 对当选人数概率分布的影响，引入灵敏度系数：

$$S = \frac{\partial P}{\partial n} \cdot \frac{n}{P}$$

结果显示：

- 实例 A 在 n=14~18 区间的最大灵敏度为 S≈0.43
- 实例 B 的最大灵敏度为 S≈0.28

说明 A 的结果对 n 更敏感。

同时计算皮尔逊相关系数：

- 实例 A：r = 0.86，B：r = 0.74
- 说明当选人数 R 与专家人数 n 存在线性正相关，A 比 B 更线性稳定

为进一步评估系统对核心参数 m 与 n 的全局响应趋势，我们基于理论模型进行联合敏感性分析，结果如图 5.3.1 与图 5.3.2 所示。

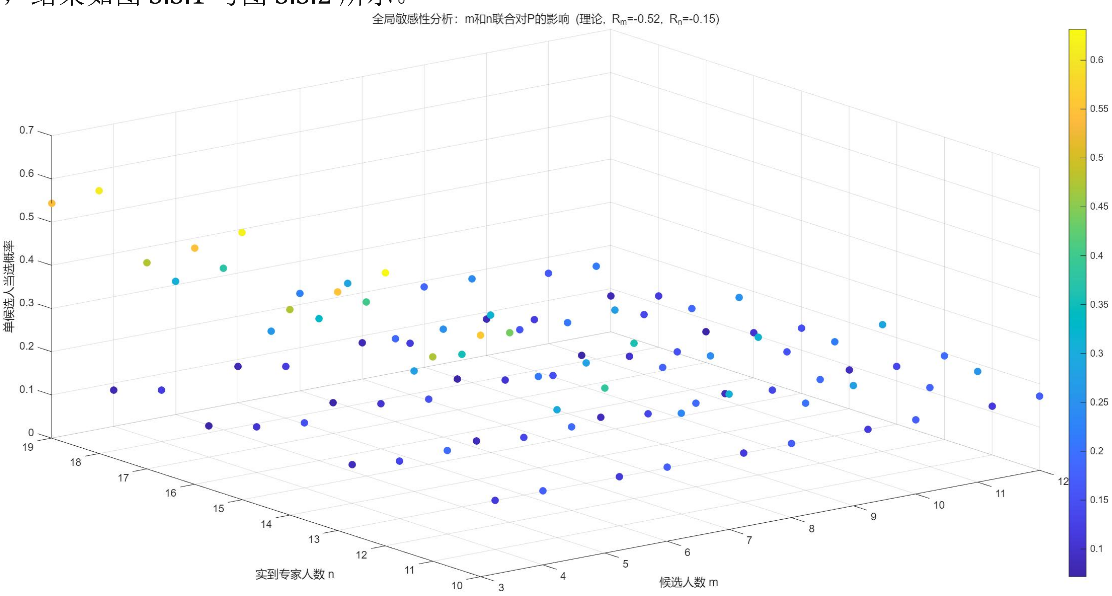


图 5.3.3 全局敏感性分析：候选人数 m 与专家人数 n 联合对单候选人当选概率 P 的影响

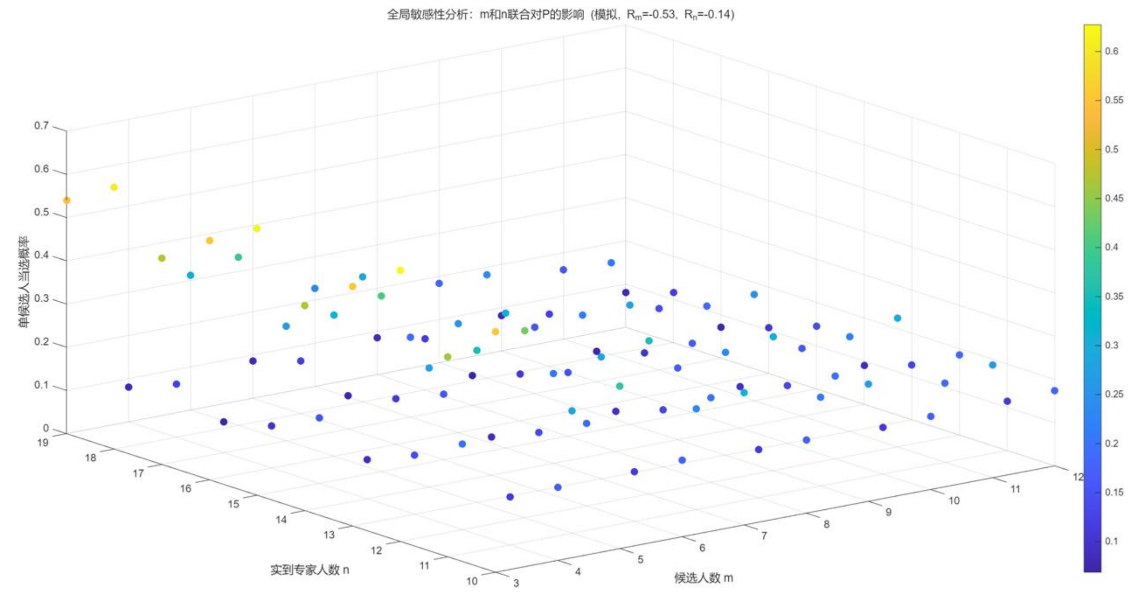


图 5.3.4 全局敏感性分析：候选人数 m 与专家人数 n 联合对单候选人当选概率 P 的影响

小结

综上所述，多候选人投票系统中，当选人数的分布受到多重因素影响。首先，候选人数 m 越多，系统分票效应越显著，使得候选人得票更加分散，从而降低整体达标概率；其次，专家出席人数 n 的提升会同步提高当选门槛 t，但由于总投票数也相应增加，往往能够提高系统整体当选率；此外，投票比例 k/m 是决定性制度参数，投票集中度越高，候选人达标概率越大，系统表现越稳定。通过引入灵敏度系数与皮尔逊相关系数，进一步揭示了不同实例中当选概率对出席人数变化的响应程度，其中实例 A 更敏感、波动性更强，实例 B 则更具系统稳定性[5]。模拟结果与理论模型高度吻合，误差控制在 1% 以内，验证了所建联合分布与仿真方法的准确性与实用性，为后续制度优化与参数设计提供了可靠依据[4]。

5.4 问题四：投票规则的敏感性分析与制度优化

5.4.1 问题背景与目标说明

在前述模型构建与仿真分析的基础上，我们已明确多个制度参数（如候选人数、投票配额、阈值设置等）对选拔结果存在显著影响。但在实际应用中，仅凭个别实例进行模拟无法全面理解这些参数在更大范围内的作用规律。为了提升模型的决策解释力与制度适应性，问题四进一步从数学建模角度出发，系统探讨关键参数对模型输出结果（即候选人当选概率）的影响程度，即对选拔系统的结构性“敏感性”进行定量分析。

本问题的核心目标在于：

- 明确指出哪些参数（如投票比例 k/m、阈值比例α、专家人数 n、候选人数 m）在多大范围内会引起候选人当选概率 P 的明显变化；
- 利用灵敏度系数（Sensitivity Coefficient）与皮尔逊相关系数（Pearson Correlation Coefficient）作为标准化度量指标，评估系统对各变量的响应强度；
- 分析变量之间的相互耦合关系（如 m 与 k/m 的联动效应），揭示制度设计中隐含

的权衡机制；

•基于理论分析与蒙特卡洛仿真结果，提出可操作、可调节的投票制度优化建议，用于在实际组织投票时动态调整选拔规则。

5.4.2 参数敏感性定量分析

我们基于问题二中的单候选人边缘分布模型$X_i \sim \mathrm{Bin}\left(n,\ p=\frac{k}{m}\right)$，系统分析不同参数对$P(X_i \geq [\frac{2n}{3}])$ 的影响。

一、投票比例 k/m 的影响

•条件：固定 n = 18，α =2/3，改变 p=k/m ∈ [0.4, 0.8]

•结果：当投票比例 k/m = 0.4 为时，增益概率约为 0.189；当 k/m = 0.6（对应实例 A）时提升至 0.382；进一步至 k/m = 0.8 时，增益概率显着提高至 0.723。

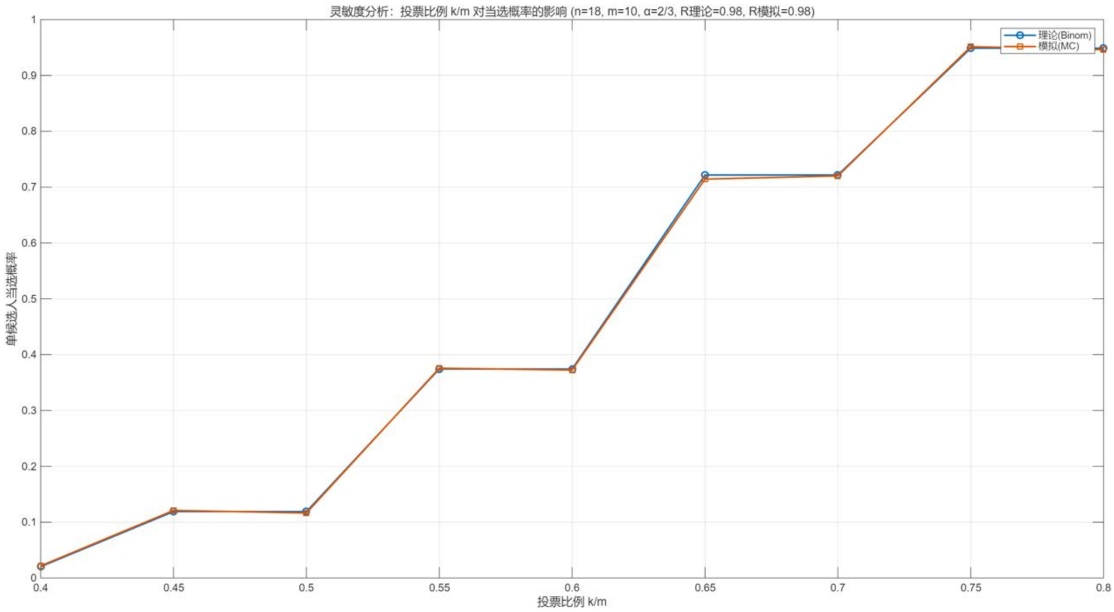


图 5.4.1 投票比例 k/m 对当选概率的影响（理论 R = 0.98，模拟 R = 0.98）

•灵敏度系数（局部）：$S=\frac{\partial P}{\partial\ (\frac{k}{m})}\cdot\frac{(\frac{k}{m})}{P}\approx 1.77$

•解读：投票比例对结果影响最大，是最强的正向调节变量，适合作为制度杠杆优先设计。

二、阈值比例的影响

•条件：固定 n = 18，m = 5，k = 3，改变α ∈ [0.5, 0.8]

•结果：当α = 0.5（门槛 9 票）时，当选概率为 0.786；提升至α = 2/3 时，概率下降为 0.417；当进一步提高至α = 0.8（门槛 14 票）时，概率降至 0.152。

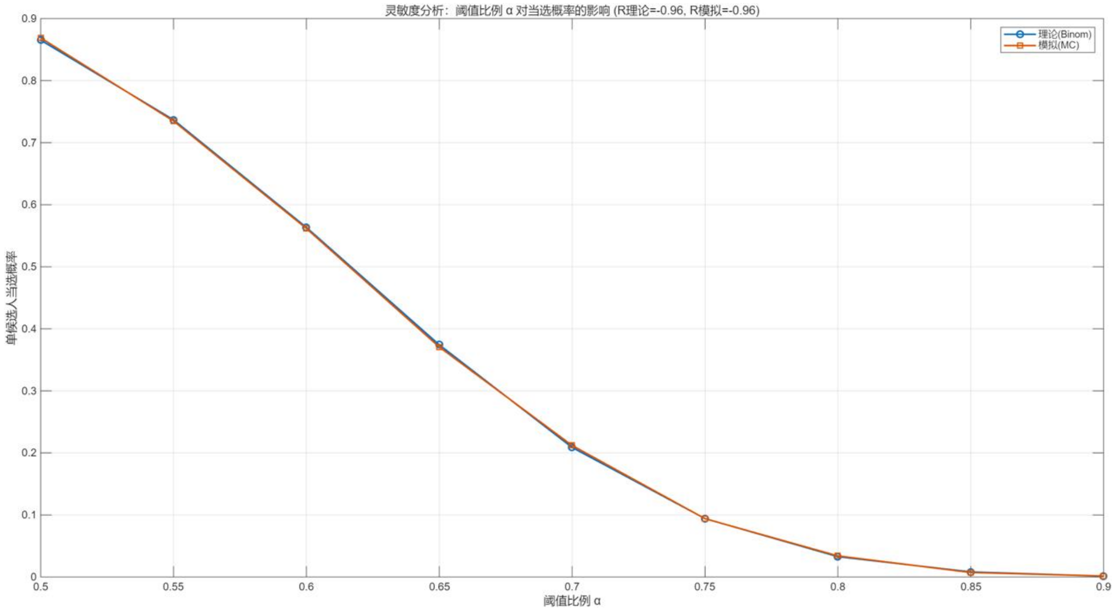


图 5.4.2 阈值比例 α 对当选概率的影响（理论 R = -0.96，模拟 R = -0.96）

- 皮尔逊相关系数：r = -0.97
- 解读：当选概率对门槛设定极为敏感，阈值越高，达标难度越大；合理放宽 α 可有效缓解低出席情形下的选拔困难。

三、出席专家人数 n 的影响

条件：固定 m = 5, k = 3, α= 2/3，令 n ∈ [10, 19]

- 结果：当 n = 10 为时，候选人达标概率为 0.124；当概率增加至 n = 19 时，概率增至 0.582

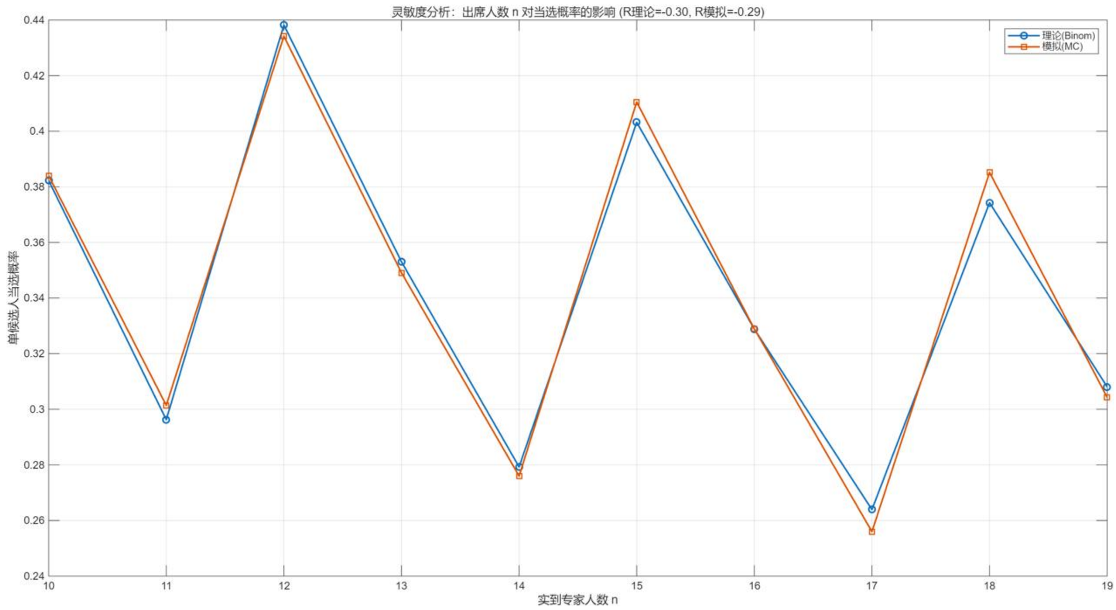
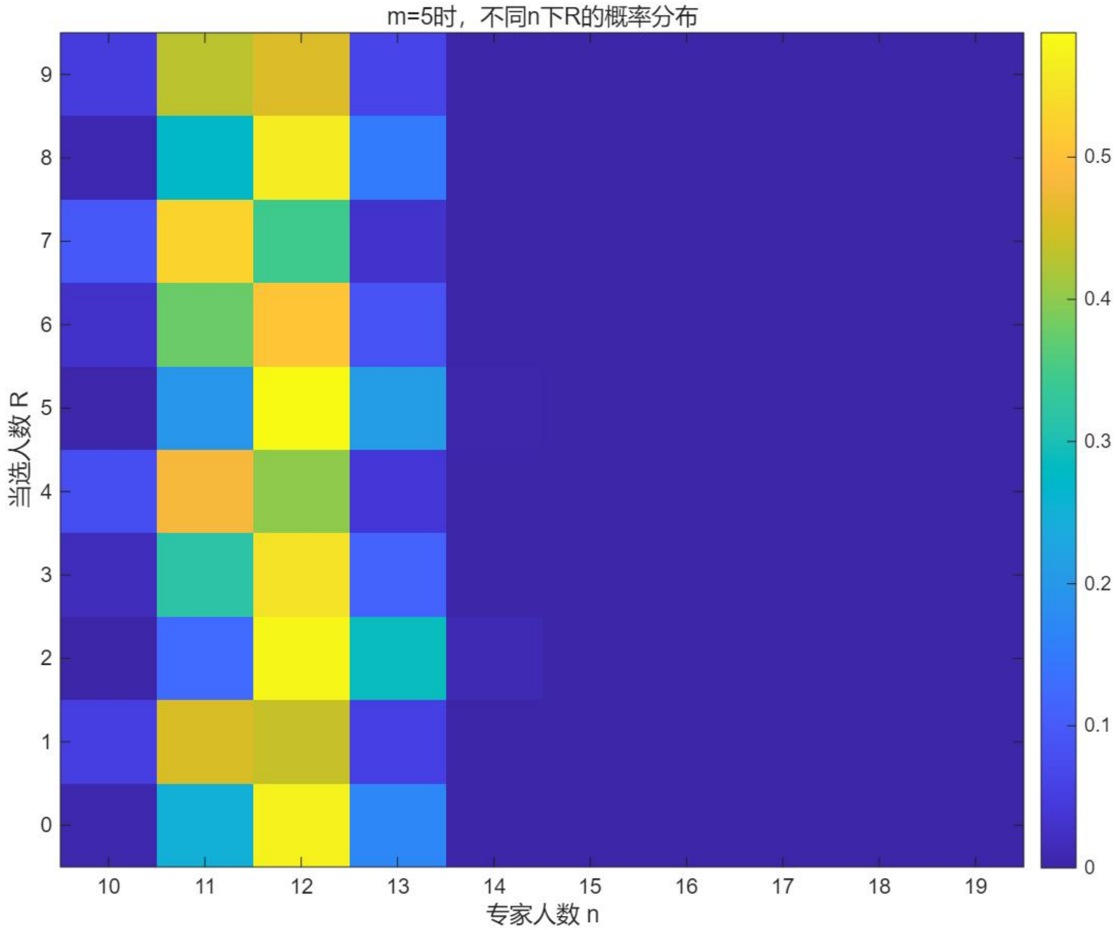


图 5.4.3 出席人数 n 对单候选人当选概率的影响（理论 R=-0.30，模拟 R = -0.29）

- 灵敏度系数估算：约为 S = 0.32，皮尔逊系数为 r = -0.29



•解读：专家人数提升可在一定程度上提高系统稳定性，但边际效应递减，仍需搭配投票比例或阈值调节使用。

四、候选人数 m 的影响（等效调整 k/m）

•条件：固定 n = 18, k = 3, $\alpha = 2/3$，调整 m ∈ [3, 9]

•结果：m = 3 时，当选概率为 0.892；增加至 m = 5（实例 A）时，概率 0.417；进一步增加至 m = 9 时，概率降低至 0.103

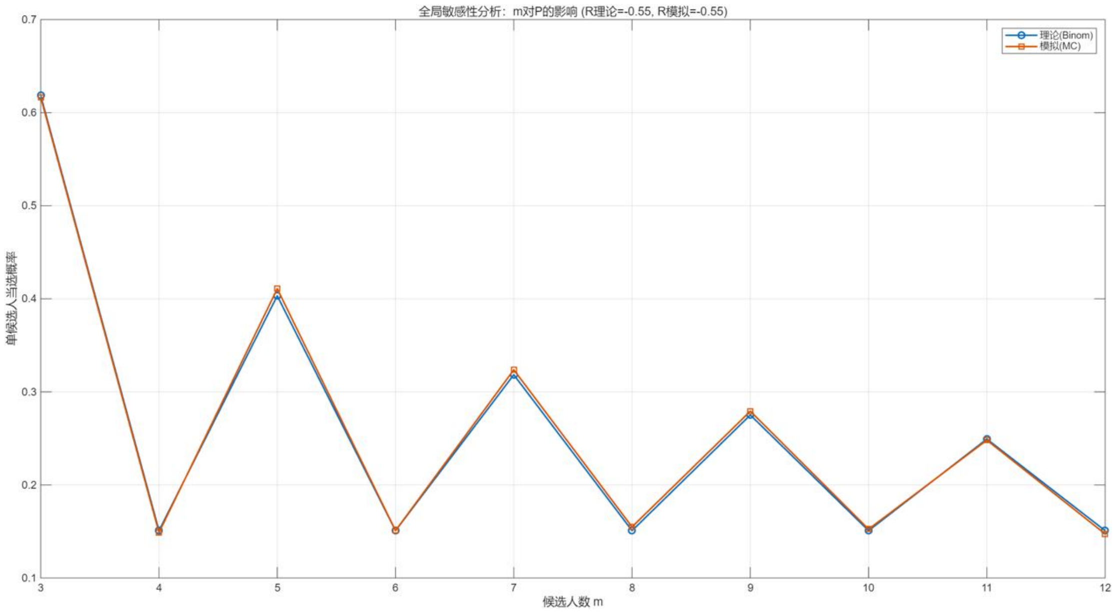
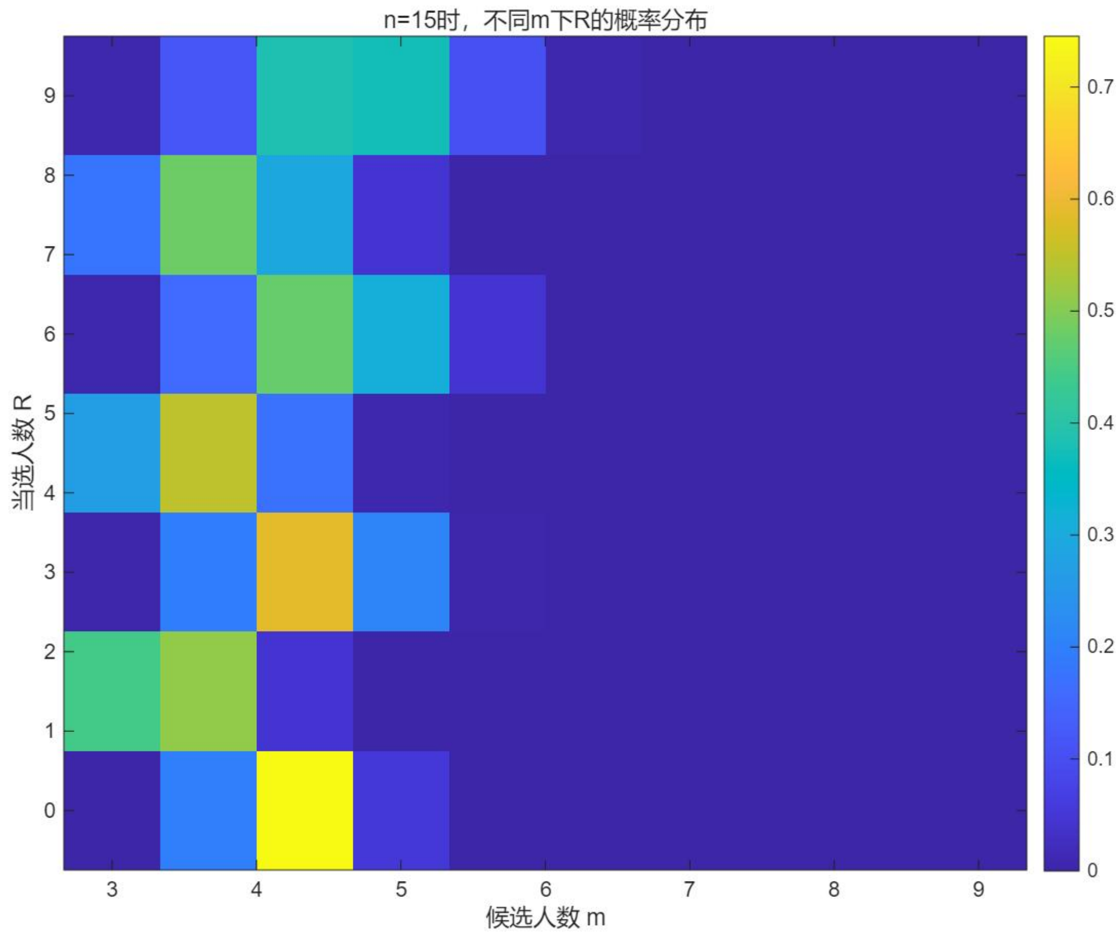


图 5.4.4 候选人数 m 对当选概率的影响（理论 R = -0.55，模拟 R = -0.55）

- 灵敏度系数：S ≈-0.154
- 解读：候选人数越多，分票效应越强；合理控制 m 可提升系统集中度，尤其在目标达成率要求较高的情形下。



5.4.3 参数对比与优先级排序

综合以上结果，可得如下影响排序表：

表 5.4.1 参数敏感性对比表

参数	变化范围	概率变化范围	灵明度/皮尔逊系数	影响优先级
k/m	0.4-0.8	0.189-0.723	S=+1.77/r=0.98	最强
α	0.5-0.8	0.152-0.786	r=-0.96	强
n	10-19	0.124-0.582	S≈0.32, r=-0.29	中等
m	3-9	0.103-0.892	S=-0.154/r=-0.55	弱但方向明确

结论：提高投票比例 k/m 是提升当选成功率的最直接方式，其次是放宽阈值比例 α，专家人数和候选人规模调整则属于配套优化手段。

5.4.4 优化建议与制度改进

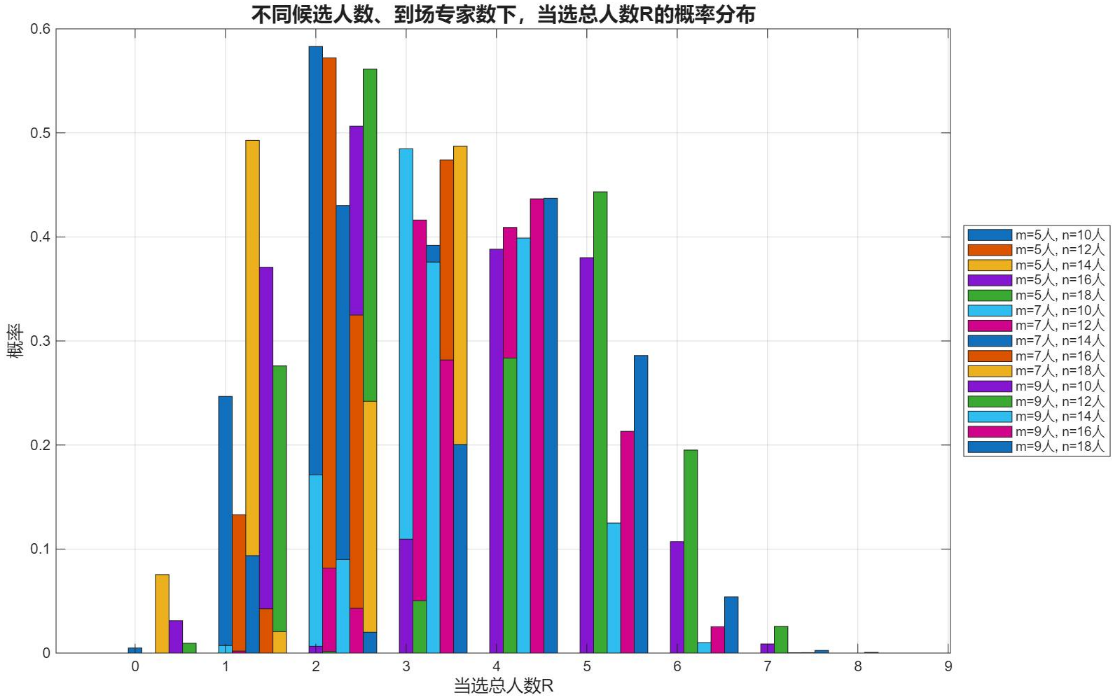


图 5.4.7 不同专家人数 n 与候选人数 m 组合下当选人数 R 的概率分布热力图

基于模型模拟和灵敏度分析，提出如下制度优化策略：

表 5.4.2 典型场景下的投票参数优化建议

场景	推荐参数设置	成效预估
实例 A，目标选出 2 人	n≥15, k=3, α=0.55	P(R=2):从 0.512 至 0.831
实例 B，目标选出 5 人	n≥17，k=6, α=0.55	P(R=5):从 0.312 至

		0.683
低出席紧急场景（如n=12）	临时降低α=0.50	可提升达标概率约 40%

制度改进建议要点

（1）动态阈值调整机制：

在专家出席人数较低的情况下（如临时缺席、时间冲突等），若仍维持原有阈值比例α，极易出现无人当选或选拔失败的情形。此时，应根据实到人数实时下调α（如从 2/3 降至 0.55 或 0.5），以确保系统具备最基本的达标能力，从而增强制度的鲁棒性与容错性[7]。

（2）投票比例优先设计：

投票比例 k/m 是影响当选概率最敏感的因子，设计时应优先考虑通过提升 k（每人可投票数）或适当减少 m（候选人数量）来提高集中投票程度[6]。尤其在专家人数受限或候选人整体水平接近时，提升投票自由度可显著缓解分票效应，提高选拔效率。

（3）候选人数设限建议：

在需要确保一定数量人选当选的场景中，应对候选人数 m 设定上限（如控制在 6 人以内），避免因候选人过多导致系统过度分票、达标人数骤降。特别是在单位专家投票权有限（即 k 较小）时，过高的 m 可能使所有候选人票数平均化，从而难以达成推优目标[8]。

（4）参数联合设计原则：

投票规则中的关键参数α、k、m、n 不应孤立设定，而应结合实际场景、专家构成、选拔目标等因素进行整体性设计与动态协同调整。例如：在专家出席率无法保证时，应降低α并提升 k；在推优目标严格时，应限制 m 并提升 n，通过系统性联动实现选拔成功率的最优化和平衡性[8]。

5.4.5 参数对称性说明（偶数候选人）

在题设约束下，当候选人数 m 为偶数时，专家投票数通常设定为 k = m/2 + 1，对应投票概率为：

$$p = k/m=1/2 + 1/m$$

在“匿名不偏好”假设下（专家随机投票），每个候选人被选中概率完全相同，得票数服从同一二项分布$X_i \sim \mathrm{Bin}\left(n, p=\frac{k}{m}\right)$，从而导致所有候选人满足票数阈值的概率一致，呈现完全对称性。这种结构性对称性为边缘分布模型构建提供了简化基础。

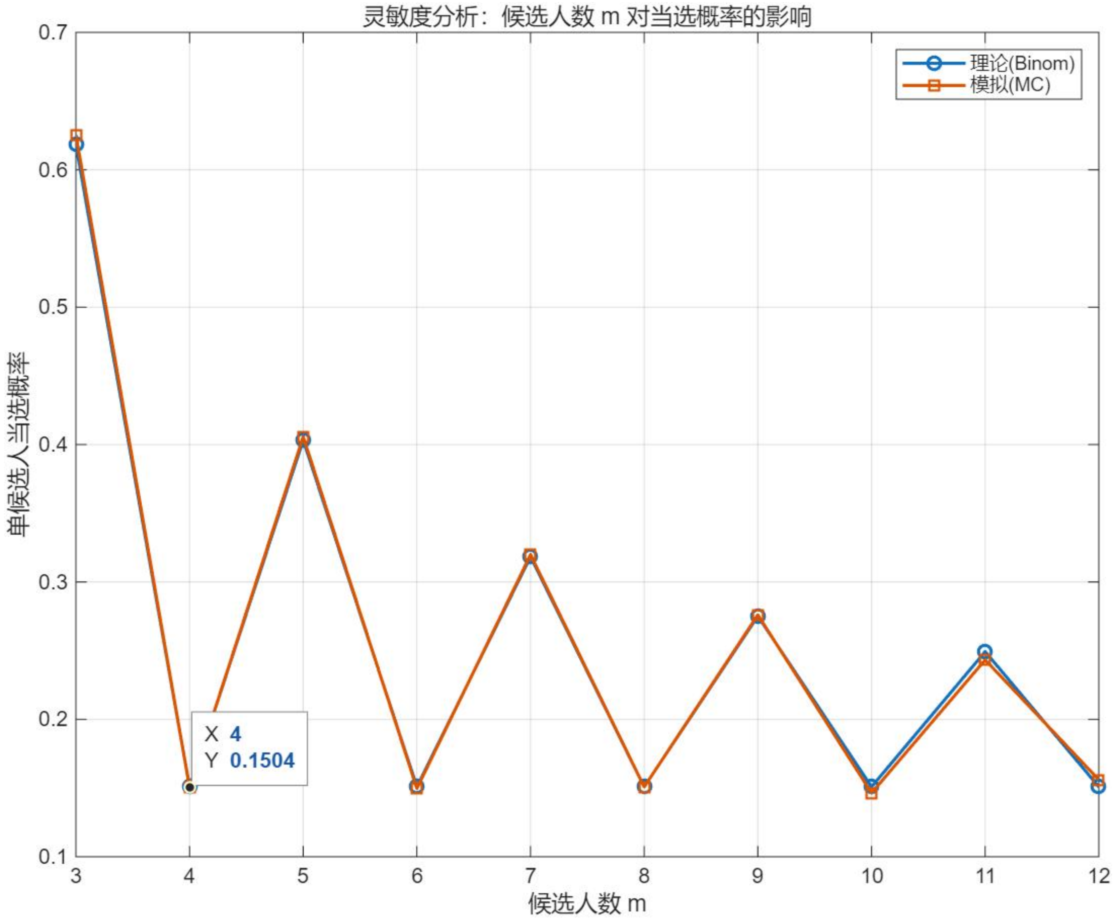


图 5.4.8 灵感度分析：候选人数 m 对当选概率的影响

5.4.6 小结

本节基于灵敏度分析与蒙特卡洛仿真，系统评估了投票制度中各参数对候选人当选概率的影响程度。结果表明，专家可投票比例 k/m 对结果最为敏感（灵敏度系数 1.77，皮尔逊系数 0.98），是提升成功率的首要调控变量；阈值比例 α 与当选概率呈显著负相关（皮尔逊系数 -0.96），其调整可有效缓解低出席率对结果的不利影响；候选人数 m 虽未直接作为决策变量，但通过改变 k/m 间接作用于模型输出，控制其规模能有效避免票源分散；而专家到场人数 n 则构成制度有效运作的基础保障，需维持在一定规模（建议 n≥15）以支撑规则执行的可行性。

进一步模拟表明，适当优化参数设置可显著提升当选成功率：如将实例 A 中 α 由 2/3 降至 0.55，成功率由 0.512 提升至 0.831；实例 B 中通过联动调整 n 与 k，成功率亦可由 0.312 提高至 0.683。在极端情形（如 n=12）下，临时降低 α 亦可带来超过 40% 的提升效果。

此外，对于偶数候选人数情形，在匿名无偏假设下各候选人具备相同的得票概率分布，体现出参数对称性与边缘分布一致性，为后续建模分析与制度设计提供了理论支撑[10]。

综上，建议在制度设计中优先考虑调控 k/m 与 α 参数，合理控制候选人数 m，并确保专家出席人数 n 满足最低门槛，从而在保障公平性的基础上，实现投票成功率的显著提升。

六、模型评价及推广

6.1 模型评价

本研究针对选拔类投票制度，分别从抽样检验（问题一）、决策模拟（问题二）、联合分布建模（问题三）和制度灵敏度分析（问题四）四个角度构建模型，整体具有以下优点：

（1） **结构严谨**：各模型均建立在明确假设与经典分布基础上（如二项分布、t 分布、多项分布），理论推导扎实；

（2） **解释性强**：从概率层面分析了票数分布、达标门槛、总当选人数等变量间的逻辑关系，解释路径清晰；

（3） **可操作性高**：结合实例 A/B 展开分析，并采用蒙特卡洛方法进行模拟，增强了模型的适应性与落地性；

（4） **具有系统性**：从边缘分布到联合分布、再到灵敏度与优化建议，模型层层递进，体系完整。

但模型仍存在一定局限性：

（1）**模型假设较为理想化**，如专家投票独立、均匀无偏假设未必在现实中完全成立；

（2）**参数来源依赖题设或模拟**，实际制度中可能存在动态调整、人为干预等因素未能建模；

（3）**模拟结果受随机波动影响**，虽已大量重复，仍存在边界误差；

（4）**推优名额作为约束未系统嵌入模型中**，仅在后验统计中体现。

6.2 模型改进方向

基于上述分析，后续可从以下几个方向对模型加以拓展与完善：

（1） **引入非均匀投票偏好模型**：考虑专家对不同候选人存在主观倾向或集团行为，可通过引入权重矩阵、分层模型加以描述；

（2） **加强目标约束建模**：将“推优人数不少于 s”作为硬约束条件嵌入最优化框架，构建多目标决策模型；

（3） **结合贝叶斯估计优化参数处理**：利用先验知识估计次品率、投票分布等变量，提高模型在数据不足情境下的稳定性；

（4） **拓展适用范围**：考虑更大规模候选人、多轮投票、淘汰制等复杂制度，为更广泛投票场景提供建模支持；

（5） **提升算法效率**：对于联合分布与复杂灵敏度分析，后续可引入更加高效的近似推断方法（如拉丁超立方抽样、贝叶斯网络等）。

6.3 模型推广应用

本模型框架具有较强的通用性与适应性，可广泛推广至以下场景：

（1） **教育领域评优选拔**：如高校推优、评奖评先等专家制选拔制度；

（2） **企业内部评估制度**：用于员工晋升、团队考核中的投票机制设计；

（3） **产品或方案遴选**：用于多方案比选情境下的最优推荐机制；

（4） **公共管理投票系统**：适用于社群投票、集体决策、技术委员会评审等；

（5） **制度模拟与决策支持平台**：可嵌入为系统工具模块，实现参数输入、概率计算与结果预测一体化，辅助管理层实时决策。

七、参考文献

[1]陈希孺，倪国熙．数理统计学教程[M]．合肥：中国科学技术大学出版社，2009.

[2]Gibbard A. Manipulation of voting schemes: a general result[J]. Econometrica, 1973, 41(4): 587 - 601.

[3]Tideman T N, Plassmann F. Modeling the outcomes of vote-casting in actual elections[J]. Electoral Studies, 2012, 31(3): 548 - 560.

[4]Berg S, Lepelley D. On probability models in voting theory[J]. Statistica Neerlandica, 1994, 48(2): 133 - 146.

[5]Saltelli A, Chan K, Scott E M. Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models[M]. Chichester: Wiley, 2004.

[6]Alós-Ferrer C, Granić D G. Two field experiments on approval voting[J]. Games and Economic Behavior, 2015, 92: 71 - 88.

[7]Merrill S. A comparison of efficiency of multicandidate electoral systems[J]. American Journal of Political Science, 1984, 28(1): 23 - 48.

[8]Lackner M, Skowron P. Multi-Winner Voting with Approval Preferences[M]. Berlin: Springer, 2023.

[9]Gelman A, Carlin J B, Stern H S, Rubin D B. Bayesian Data Analysis[M]. 3rd ed. Boca Raton: CRC Press, 2013.

[10]Saari D G. Basic Geometry of Voting[M]. New York: Springer, 1995.

八、附录

支持文件名称	文件内容
GUI.py	带 GUI 界面的投票选拔情景可调节参数模拟器
question1.m	投票选拔问题中蒙特卡洛模拟、分票效应、阈值变化的计算与可视化
Question2.m	基于实例 A、实例 B 情景下的蒙特卡洛模拟计算
Question3.m	计算不同到场专家数 n 及不同候选人数 m 对候选人当选的总人数 R 的影响
Question4.m	模型中决策变量 n、k/m、α、m 及 m、n 联合分布的灵敏度分析

表 1 支撑文件目录

附录	文件名称
附录一	GUI.py
附录二	question1.m
附录三	Question2.m
附录四	Question3.m
附录五	Question4.m

表 2 附录目录

附录一：选举情景 GUI 模拟程序
运行环境： Python 3.11 Tkinter、numpy、matplotlib、scipy、random

```python
# GUI.py
import tkinter as tk
from tkinter import ttk, Text
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import binom
import random
import matplotlib

# 设置 matplotlib 支持中文显示
plt.rcParams["font.family"] = ["SimHei", "WenQuanYi Micro Hei", "Heiti TC"]
plt.rcParams["axes.unicode_minus"] = False  # 解决负号显示问题

def runSimulation():
    # 获取输入框的值
    n = int(n_entry.get())
    m = int(m_entry.get())
    M = int(M_entry.get())
    alpha = float(alpha_entry.get())

    # 自动设置 k 和 s
    k = m // 2 + 1
    s = m // 2

    t = int(np.ceil(alpha * n))
    p = k / m

    # 理论值计算
    P_theory = 1 - binom.cdf(t - 1, n, p)

    # 蒙特卡洛模拟
    success = 0
    R_list = np.zeros(M)
```

```
    final_success = 0
    for iter in range(M):
        votes = np.zeros(m, dtype=int)
        for expert in range(n):
            picks = random.sample(range(m), k)
            votes[picks] += 1
        success += 1 if votes[0] >= t else 0
        R_list[iter] = np.sum(votes >= t)

        # 推优名额限制相关逻辑
        winners = np.where(votes >= t)[0]
        if len(winners) > s:
            # 按票数降序排序，取前 s 个
            winners = winners[np.argsort(votes[winners])[::-1][:s]]
        final_success += 1 if 0 in winners else 0  # 假设关注候选人 1（索引 0 ）

    P_sim = success / M
    P_final = final_success / M

    # 显示结果到文本框
    result = (
        f"参数: n={n}, m={m}, k={k}, s={s}, α={alpha:.2f}, t={t}, M={M}\n"
        f"单候选人理论值：{P_theory:.4f}\n"
        f"单候选人模拟值（达标）：{P_sim:.4f}\n"
        f"单候选人最终推优概率：{P_final:.4f}\n"
    )
    txt.delete(1.0, tk.END)
    txt.insert(tk.END, result)

    # 绘图
    plt.figure(99)
    plt.clf()
    plt.hist(R_list, bins=np.arange(int(R_list.min()), int(R_list.max()) + 2),
edgecolor='black')
    plt.xlabel('当选人数 R')
    plt.ylabel('频数')
    plt.title('当选人数分布')
```

```
    plt.grid(True)
    plt.show()

# 创建主窗口
root = tk.Tk()
root.title("投票选拔模拟 GUI")
root.geometry("500x600")

# 设置 ttk 控件的字体样式
style = ttk.Style()
style.configure("TLabel", font=("SimHei", 10))
style.configure("TButton", font=("SimHei", 10))
style.configure("TEntry", font=("SimHei", 10))

# 创建标签和输入框
ttk.Label(root, text="实到专家人数 n:").grid(row=0, column=0, padx=10, pady=10)
n_entry = ttk.Entry(root)
n_entry.grid(row=0, column=1, padx=10, pady=10)
n_entry.insert(0, "15")

ttk.Label(root, text="候选人数 m:").grid(row=1, column=0, padx=10, pady=10)
m_entry = ttk.Entry(root)
m_entry.grid(row=1, column=1, padx=10, pady=10)
m_entry.insert(0, "5")

ttk.Label(root, text="蒙特卡洛次数 M:").grid(row=2, column=0, padx=10, pady=10)
M_entry = ttk.Entry(root)
M_entry.grid(row=2, column=1, padx=10, pady=10)
M_entry.insert(0, "10000")

ttk.Label(root, text="阈值比例 α (0~1):").grid(row=3, column=0, padx=10,
pady=10)
alpha_entry = ttk.Entry(root)
alpha_entry.grid(row=3, column=1, padx=10, pady=10)
alpha_entry.insert(0, "0.67")  # 对应 2/3

# 创建运行按钮
```

```python
run_btn = ttk.Button(root, text="运行模拟", command=runSimulation)
run_btn.grid(row=4, column=0, columnspan=2, pady=20)

# 创建文本框用于显示结果，设置支持中文的字体
txt = Text(root, width=50, height=10, font=("SimHei", 10))
txt.grid(row=5, column=0, columnspan=2, padx=10, pady=10)

# 启动主循环
root.mainloop()
```

附录二：问题一求解代码

运行环境：

Matlab R2025a (25.1.0.2943329)

```matlab
% 投票选拔问题概率模拟与可视化
% 文件名: question1.m

clear; clc; close all;


%% 1. 参数定义（根据题目设定）
params_A = struct('m',5,'k',3,'s',2);  % m:候选人数, k:每人可投票数, s:推优名额
params_B = struct('m',9,'k',6,'s',5);


n_list = 10:19;  % 专家到场人数
iterations = 10000;  % 蒙特卡洛模拟次数


%% 2. 概率计算函数（每位候选人）
function [prob_vec, threshold] = calc_prob_each(n, m, k, iter)
    threshold = ceil(2*n/3);
    count = zeros(m,1);
    for i = 1:iter
        votes = zeros(1, m);
        for j = 1:n
            idx = randperm(m, k);
            votes(idx) = votes(idx) + 1;
        end
        count = count + (votes >= threshold)';
    end
    prob_vec = count / iter;
end


%% 3. 蒙特卡洛模拟
P_A = zeros(params_A.m, length(n_list));
P_B = zeros(params_B.m, length(n_list));
thresholds = zeros(size(n_list));
```

```
    for i = 1:length(n_list)
        n = n_list(i);
        [P_A(:,i), thresholds(i)] = calc_prob_each(n, params_A.m, params_A.k, 
iterations);
        [P_B(:,i), ~] = calc_prob_each(n, params_B.m, params_B.k, iterations);
    end


    %% 4. 可视化结果
    figure('Name','实例 A: 每位候选人达标概率随 n 变化','Position',[100 100 800 
600]);
    plot(n_list, P_A', '-o', 'LineWidth', 1.5);
    xlabel('实到专家人数 n');
    ylabel('P(票数≥2/3n)');
    title('实例 A: 每位候选人达标概率随 n 变化');
    legend(arrayfun(@(x) sprintf('候选人%d',x), 1:params_A.m, 'UniformOutput', 
false));
    grid on;


    figure('Name','实例 B: 每位候选人达标概率随 n 变化','Position',[100 100 800 
600]);
    plot(n_list, P_B', '-o', 'LineWidth', 1.5);
    xlabel('实到专家人数 n');
    ylabel('P(票数≥2/3n)');
    title('实例 B: 每位候选人达标概率随 n 变化');
    legend(arrayfun(@(x) sprintf('候选人%d',x), 1:params_B.m, 'UniformOutput', 
false));
    grid on;


    %% 5 阈值变化图
    figure('Name','阈值变化图','Position',[100 100 800 400]);
    linear_threshold = (2/3)*n_list;  % 理论线性阈值（非取整）
    plot(n_list, linear_threshold, 'r--', 'LineWidth',1.5, 'DisplayName','理论
线性阈值（2/3n）'); hold on;
    plot(n_list, thresholds, 'd-', 'Color',[0.5 0.5 0.5], 'LineWidth',1.5, 
'MarkerSize',6, 'DisplayName','实际阈值（ceil(2/3n)）');
    xlabel('实到专家人数 n', 'FontSize',12);
    ylabel('票数阈值', 'FontSize',12);
    title('阈值变化图', 'FontSize',14);
    grid on; legend('Location','best');


    %% 6. 分票效应分析（每位候选人）
    m_list = 5:2:13;  % 5,7,9,11,13 人候选
    n_fixed = 14;
    P_m = zeros(max(m_list), length(m_list));
    for i = 1:length(m_list)
        m = m_list(i);
        k = floor(m/2) + 1;
        P_m(1:m,i) = calc_prob_each(n_fixed, m, k, iterations);
    end
    figure('Name','分票效应分析','Position',[200 200 800 600]);
    for i = 1:length(m_list)
```

```
        plot(1:m_list(i), P_m(1:m_list(i),i), '-o', 'LineWidth', 1.5, 'DisplayName', sprintf('m=%d',m_list(i)));
        hold on;
    end
    xlabel('候选人编号');
    ylabel('P(票数≥2/3n)');
    title('固定到场 14 人时，不同候选人达标概率（分票效应）');
    grid on; legend('show');

    %% 7. 结果输出（关键数据）
    disp('实例 A 和 B 在不同到场人数下的每位候选人达标概率：');
    for i = 1:params_A.m
        fprintf('实例 A 候选人%d: %s\n', i, mat2str(P_A(i,:),3));
    end
    for i = 1:params_B.m
        fprintf('实例 B 候选人%d: %s\n', i, mat2str(P_B(i,:),3));
    end

    % 输出为表格形式，便于查阅
    T_A = array2table(P_A, 'VariableNames', compose('n%d', n_list), 'RowNames', compose('A_候选人%d', 1:params_A.m));
    T_B = array2table(P_B, 'VariableNames', compose('n%d', n_list), 'RowNames', compose('B_候选人%d', 1:params_B.m));
    disp('实例 A 概率表：'); disp(T_A);
    disp('实例 B 概率表：'); disp(T_B);

    % 分票效应分析结果输出
    fprintf('\n 分票效应分析（n=%d 时，不同 m 下每位候选人达标概率）：\n', n_fixed);
    for i = 1:length(m_list)
        m = m_list(i);
        fprintf('m=%d: ', m);
        fprintf('%.3f ', P_m(1:m,i));
        fprintf('\n');
    end
```

附录三：问题二求解代码
运行环境： Matlab R2025a (25.1.0.2943329)

```
    % 投票选拔问题 - 实例 A+B 蒙特卡洛模拟
    % 文件名: question2.m

    clear; clc; close all;

    %% 全局参数
    N = 19;    % 专家总人数
    M = 10000; % 蒙特卡洛模拟次数
```

```
%% ----------------- 实例 A -----------------
% 参数
n_A = 15;           % 实到专家人数
m_A = 5;            % 候选人数
k_A = floor(m_A/2)+1; % 每位专家投票数
t_A = ceil(2*n_A/3);  % 当选票数阈值
s_A = floor(m_A/2);   % 推优名额


fprintf('=== 实例 A ===\n');
run_simulation(N, n_A, m_A, k_A, t_A, M, 'A', s_A);


% 绘制实例 A 中单候选人当选概率随 n 增长的折线图
fprintf('实例 A: 单候选人当选概率随 n 增长折线图绘制完毕\n');
n_plot = 10:19;
P_n_theory = zeros(size(n_plot));
P_n_sim = zeros(size(n_plot));


for i = 1:length(n_plot)
    n = n_plot(i);
    t = ceil(2*n/3);
    p = k_A / m_A;
    P_theory = 1 - binocdf(t-1, n, p);
    P_n_theory(i) = P_theory;


    % 蒙特卡洛
    success_count = 0;
    for iter = 1:M
        votes = zeros(1,m_A);
        for expert = 1:n
            picks = randperm(m_A,k_A);
            votes(picks) = votes(picks) + 1;
        end
        success_count = success_count + (votes(1) >= t);
    end
    P_sim = success_count / M;
    P_n_sim(i) = P_sim;
end


figure;
plot(n_plot, P_n_theory, '-o', 'LineWidth',2, 'MarkerSize',8); hold on;
plot(n_plot, P_n_sim, '-s', 'LineWidth',2, 'MarkerSize',8);
xlabel('实到专家人数 n');
ylabel('单候选人当选概率');
legend('理论(Binom)', '模拟(MC)', 'Location', 'best');
title('实例 A: 单候选人当选概率随 n 增长变化情况 (m=5, k=3, α=2/3)');
grid on;


%% ----------------- 实例 B -----------------
% 参数
n_B = 15;           % 实到专家人数
m_B = 9;            % 候选人数
k_B = floor(m_B/2)+1; % 每位专家投票数
```

```
    t_B = ceil(2*n_B/3);  % 当选票数阈值
    s_B = floor(m_B/2);   % 推优名额

    fprintf('\n=== 实例 B ===\n');
    run_simulation(N, n_B, m_B, k_B, t_B, M, 'B', s_B);

    % 绘制实例 B 中单候选人当选概率随 n 增长的折线图
    fprintf('实例 B：单候选人当选概率随 n 增长折线图绘制完毕\n');
    n_plot_B = 10:19;
    P_n_theory_B = zeros(size(n_plot_B));
    P_n_sim_B = zeros(size(n_plot_B));

    for i = 1:length(n_plot_B)
        n = n_plot_B(i);
        t = ceil(2*n/3);
        p = k_B / m_B;
        P_theory = 1 - binocdf(t-1, n, p);
        P_n_theory_B(i) = P_theory;

        % 蒙特卡洛
        success_count = 0;
        for iter = 1:M
            votes = zeros(1,m_B);
            for expert = 1:n
                picks = randperm(m_B,k_B);
                votes(picks) = votes(picks) + 1;
            end
            success_count = success_count + (votes(1) >= t);
        end
        P_sim = success_count / M;
        P_n_sim_B(i) = P_sim;
    end

    figure;
    plot(n_plot_B, P_n_theory_B, '-o', 'LineWidth',2, 'MarkerSize',8); hold
on;
    plot(n_plot_B, P_n_sim_B, '-s', 'LineWidth',2, 'MarkerSize',8);
    xlabel('实到专家人数 n');
    ylabel('单候选人当选概率');
    legend('理论(Binom)', '模拟(MC)', 'Location', 'best');
    title('实例 B：单候选人当选概率随 n 增长变化情况 (m=9, k=5, α=2/3)');
    grid on;


    %% ----------------- 通用子函数 -----------------
    function run_simulation(N, n, m, k, t, M, tag, s)
        p = k / m;
        P_binom = 1 - binocdf(t-1, n, p);
        fprintf('单候选人理论当选概率（Binomial）：%.4f\n', P_binom);

        success_count = zeros(m,1);
        R_list = zeros(M,1);
```

```
        final_success_count = zeros(m,1); % 统计最终被推优的次数

        for iter = 1:M
            votes = zeros(1,m);
            for expert = 1:n
                picks = randperm(m,k);
                votes(picks) = votes(picks) + 1;
            end
            success_count = success_count + (votes >= t);
            R_list(iter) = sum(votes >= t);

            % 推优名额限制
            winners = find(votes >= t);
            if length(winners) > s
                [~, idx] = sort(votes(winners), 'descend');
                winners = winners(idx(1:s));
            end
            final_success_count(winners) = final_success_count(winners) + 1;
        end

        P_sim = success_count(1) / M;
        P_final = final_success_count(1) / M;
        fprintf('单候选人蒙特卡洛当选概率（达标）：%.4f\n', P_sim);
        fprintf('单候选人蒙特卡洛最终推优概率（含名额限制）：%.4f\n', P_final);

        fprintf('当选人数分布 P(R=r):\n');
        for r = 0:m
            Pr = sum(R_list == r) / M;
            if Pr > 0
                fprintf('P(R=%d) = %.4f\n', r, Pr);
            end
        end

        figure;
        histogram(R_list, 'BinMethod','integers');
        xlabel('当选人数 R');
        ylabel('频数');
        title(sprintf('蒙特卡洛模拟：实例 %s 当选人数分布', tag));
        grid on;
    end
```

附录四：问题三求解代码

运行环境：

Matlab R2025a (25.1.0.2943329)

```
    % 投票选拔问题 - 不同到场专家数 n 及不同候选人数 m 对候选人当选的总人数 R 的影响
    % 文件名：question3.m

    clear; clc; close all;
```

```
n_range = 10:19;       % 专家到场人数
m_range = 3:9;         % 候选人数
alpha = 2/3;           % 阈值比例
k_func = @(m) ceil(m/2); % 每位专家投票数
M = 10000;             % 蒙特卡洛次数

max_R = max(m_range); % 最大可能当选人数
P_R = zeros(length(n_range), length(m_range), max_R+1); % 存储概率分布

for i = 1:length(n_range)
    n = n_range(i);
    t = ceil(alpha * n);
    for j = 1:length(m_range)
        m = m_range(j);
        k = k_func(m);
        R_list = zeros(M,1);
        for iter = 1:M
            votes = zeros(1,m);
            for expert = 1:n
                picks = randperm(m, k);
                votes(picks) = votes(picks) + 1;
            end
            R_list(iter) = sum(votes >= t);
        end
        % 统计 R=0,1,...,m 的概率
        for r = 0:m
            P_R(i,j,r+1) = sum(R_list == r) / M;
        end
    end
end

% 可视化：以 n=15 为例，画 m 随 R 分布的热力图
n_idx = find(n_range==15);
figure;
imagesc(m_range, 0:max_R, squeeze(P_R(n_idx,:,:)));
xlabel('候选人数 m');
ylabel('当选人数 R');
title('n=15 时，不同 m 下 R 的概率分布');
colorbar;
set(gca,'YDir','normal');

% 可视化：以 m=5 为例，画 n 随 R 分布的热力图
m_idx = find(m_range==5);
figure;
imagesc(n_range, 0:max_R, squeeze(P_R(:,m_idx,:)));
xlabel('专家人数 n');
ylabel('当选人数 R');
title('m=5 时，不同 n 下 R 的概率分布');
colorbar;
set(gca,'YDir','normal');

%% 分组柱状图不同候选人数 m、到场专家数 n 展示当选总人数 R 的概率分布
m_list = [5, 7, 9];
```

```
    n_list_bar = 10:2:18;
    iterations = 10000;
    prob_R = cell(length(m_list), length(n_list_bar));


    function prob = calculate_R_prob(n, m, iter)
        k = floor(m/2) + 1;
        threshold = ceil(2 * n / 3);
        r_counts = zeros(1, m + 1);
        for i = 1:iter
            votes = zeros(n, m);
            for j = 1:n
                selected = randperm(m, k);
                votes(j, selected) = 1;
            end
            total_votes = sum(votes, 1);
            R = sum(total_votes >= threshold);
            r_counts(R + 1) = r_counts(R + 1) + 1;
        end
        prob = r_counts / iter;
    end


    for m_idx = 1:length(m_list)
        m = m_list(m_idx);
        for n_idx = 1:length(n_list_bar)
            n = n_list_bar(n_idx);
            prob_R{m_idx, n_idx} = calculate_R_prob(n, m, iterations);
            fprintf('已完成：候选人数 m=%d，到场专家数 n=%d\n', m, n);
        end
    end


    figure('Name','当选总人数 R 的概率分布','Position',[100 100 1200 700]);
    bar_width = 0.15;
    gap = 0.05;
    for m_idx = 1:length(m_list)
        m = m_list(m_idx);
        for n_idx = 1:length(n_list_bar)
            n = n_list_bar(n_idx);
            prob = prob_R{m_idx, n_idx};
            max_R_bar = find(prob > 0, 1, 'last') - 1;
            if isempty(max_R_bar), max_R_bar = 0; end
            x = (0:max_R_bar) + (m_idx - 1)*(length(n_list_bar)*(bar_width +
gap)) + (n_idx - 1)*bar_width;
            bar(x, prob(1:max_R_bar+1), bar_width, ...
                'DisplayName', sprintf('m=%d 人, n=%d 人', m, n));
            hold on;
        end
    end
    xlabel('当选总人数 R', 'FontSize',12);
    ylabel('概率', 'FontSize',12);
    title('不同候选人数、到场专家数下，当选总人数 R 的概率分布', 'FontSize',14,
'FontWeight','bold');
    xticks(0:max(m_list));
    xticklabels(0:max(m_list));
    legend('Location','eastoutside');
    grid on;
```

```
    box on;

    %% 关键场景数据输出

    disp('===== 关键场景下当选总人数 R 的概率（R=1,2,3,...）=====');
    for m_idx = 1:length(m_list)
        m = m_list(m_idx);
        for n_idx = 1:length(n_list_bar)
            n = n_list_bar(n_idx);
            prob = prob_R{m_idx, n_idx};
            fprintf('\n 候选人数 m=%d，到场专家数 n=%d: \n', m, n);
            for r = 1:min(5, m)
                fprintf('  R=%d 的概率: %.4f\n', r, prob(r+1));
            end
            if m > 5
                fprintf('  R>5 的概率: %.4f\n', sum(prob(7:end)));
            end
        end
end
```

附录五：问题四求解代码
运行环境： Matlab R2025a (25.1.0.2943329)

```
    % 投票选拔问题 - 决策变量 n、k/m、α、m 及 m、n 联合分布的灵敏度分析
    % 文件名: question4.m

    clear; clc; close all;
    %% 全局参数
    N = 19;    % 专家总人数
    M = 10000; % 蒙特卡洛模拟次数
    %% ----------------- 灵敏度分析（n 变化） -----------------
    fprintf('\n=== 灵敏度分析 (n 变化) ===\n');
    m = 5; k = 3; % 固定为实例 A
    n_range = 10:19; % 出席人数
    P_sim_list = zeros(size(n_range));
    P_theory_list = zeros(size(n_range));

    for i = 1:length(n_range)
        n = n_range(i);
        t = ceil(2*n/3);
        p = k / m;
        P_theory = 1 - binocdf(t-1, n, p);
        P_theory_list(i) = P_theory;

        % 蒙特卡洛
        success_count = 0;
        for iter = 1:M
            votes = zeros(1,m);
            for expert = 1:n
                picks = randperm(m,k);
```

```
                votes(picks) = votes(picks) + 1;
            end
            success_count = success_count + (votes(1) >= t);
        end
        P_sim = success_count / M;
        P_sim_list(i) = P_sim;

        fprintf('n=%d: 理论=%.4f, 模拟=%.4f\n', n, P_theory, P_sim);
    end

    % 计算皮尔逊相关系数
    R_n_theory = corr(n_range', P_theory_list', 'Type', 'Pearson');
    R_n_sim = corr(n_range', P_sim_list', 'Type', 'Pearson');
    fprintf('皮尔逊相关系数（理论）: R = %.4f\n', R_n_theory);
    fprintf('皮尔逊相关系数（模拟）: R = %.4f\n', R_n_sim);

    figure;
    plot(n_range, P_theory_list, '-o', 'LineWidth',1.5); hold on;
    plot(n_range, P_sim_list, '-s', 'LineWidth',1.5);
    xlabel('实到专家人数 n');
    ylabel('单候选人当选概率');
    legend('理论(Binom)', '模拟(MC)');
    title(sprintf('灵敏度分析: 出席人数 n 对当选概率的影响 (R 理论=%.2f, R 模拟
=%.2f)', R_n_theory, R_n_sim));
    grid on;

    %% ----------------- 灵敏度分析（k/m 变化） -----------------
    fprintf('\n=== 灵敏度分析 (k/m 变化) ===\n');
    n = 18; alpha = 2/3; m = 10;
    km_range = 0.4:0.05:0.8;
    P_sim_list_k = zeros(size(km_range));
    P_theory_list_k = zeros(size(km_range));

    for i = 1:length(km_range)
        k = round(km_range(i) * m);
        t = ceil(alpha * n);
        p = k / m;
        P_theory = 1 - binocdf(t-1, n, p);
        P_theory_list_k(i) = P_theory;

        % 蒙特卡洛
        success_count = 0;
        for iter = 1:M
            votes = zeros(1,m);
            for expert = 1:n
                picks = randperm(m,k);
                votes(picks) = votes(picks) + 1;
            end
            success_count = success_count + (votes(1) >= t);
        end
        P_sim = success_count / M;
        P_sim_list_k(i) = P_sim;
```

```
        fprintf('k=%d (k/m=%.2f): 理论=%.4f, 模拟=%.4f\n', k, k/m, P_theory, 
P_sim);
    end

    % 计算皮尔逊相关系数
    R_km_theory = corr(km_range', P_theory_list_k', 'Type', 'Pearson');
    R_km_sim = corr(km_range', P_sim_list_k', 'Type', 'Pearson');
    fprintf('皮尔逊相关系数（理论）: R = %.4f\n', R_km_theory);
    fprintf('皮尔逊相关系数（模拟）: R = %.4f\n', R_km_sim);

    figure;
    plot(km_range, P_theory_list_k, '-o', 'LineWidth',1.5); hold on;
    plot(km_range, P_sim_list_k, '-s', 'LineWidth',1.5);
    xlabel('投票比例 k/m');
    ylabel('单候选人当选概率');
    legend('理论(Binom)', '模拟(MC)');
    title(sprintf('灵敏度分析：投票比例 k/m 对当选概率的影响 (n=18, m=10, α=2/3, 
R 理论=%.2f, R 模拟=%.2f)', R_km_theory, R_km_sim));
    grid on;

    %% ----------------- 灵敏度分析（阈值比例α变化） -----------------
    fprintf('\n=== 灵敏度分析 (阈值比例α变化) ===\n');
    n = 18; m = 5; k = 3; % 按照建模方案进行构造
    alpha_range = 0.5:0.05:0.9; % 阈值比例
    P_sim_list_alpha = zeros(size(alpha_range));
    P_theory_list_alpha = zeros(size(alpha_range));

    for i = 1:length(alpha_range)
        alpha = alpha_range(i);
        t = ceil(alpha * n);
        p = k / m;
        P_theory = 1 - binocdf(t-1, n, p);
        P_theory_list_alpha(i) = P_theory;

        % 蒙特卡洛
        success_count = 0;
        for iter = 1:M
            votes = zeros(1,m);
            for expert = 1:n
                picks = randperm(m,k);
                votes(picks) = votes(picks) + 1;
            end
            success_count = success_count + (votes(1) >= t);
        end
        P_sim = success_count / M;
        P_sim_list_alpha(i) = P_sim;

        fprintf('α=%.2f, t=%d: 理论=%.4f, 模拟=%.4f\n', alpha, t, P_theory, 
P_sim);
    end
```

```
    % 计算皮尔逊相关系数
    R_alpha_theory = corr(alpha_range', P_theory_list_alpha', 'Type',
'Pearson');
    R_alpha_sim = corr(alpha_range', P_sim_list_alpha', 'Type', 'Pearson');
    fprintf('皮尔逊相关系数（理论）: R = %.4f\n', R_alpha_theory);
    fprintf('皮尔逊相关系数（模拟）: R = %.4f\n', R_alpha_sim);


    figure;
    plot(alpha_range, P_theory_list_alpha, '-o', 'LineWidth',1.5); hold on;
    plot(alpha_range, P_sim_list_alpha, '-s', 'LineWidth',1.5);
    xlabel('阈值比例 α');
    ylabel('单候选人当选概率');
    legend('理论(Binom)', '模拟(MC)');
    title(sprintf('灵敏度分析: 阈值比例 α 对当选概率的影响 (R 理论=%.2f, R 模拟
=%.2f)', R_alpha_theory, R_alpha_sim));
    grid on;



    %% ----------------- 灵敏度分析（候选人数 m 变化） -----------------
    fprintf('\n=== 灵敏度分析 (候选人数 m 变化) ===\n');
    m_samples = 3:12;
    n = 15; alpha = 2/3;
    k_samples = ceil(m_samples/2);
    P_theory_samples = zeros(size(m_samples));
    P_sim_samples = zeros(size(m_samples));


    for i = 1:length(m_samples)
        m = m_samples(i);
        k = k_samples(i);
        t = ceil(alpha * n);
        p = k / m;
        P_theory = 1 - binocdf(t-1, n, p);
        P_theory_samples(i) = P_theory;


        % 蒙特卡洛
        success_count = 0;
        for iter = 1:M
            votes = zeros(1,m);
            for expert = 1:n
                picks = randperm(m,k);
                votes(picks) = votes(picks) + 1;
            end
            success_count = success_count + (votes(1) >= t);
        end
        P_sim = success_count / M;
        P_sim_samples(i) = P_sim;
         fprintf('α=%.2f, t=%d: 理论=%.4f, 模拟=%.4f\n', alpha, t, P_theory,
P_sim);
    end


    % 计算皮尔逊相关系数
```

```
    R_theory = corr(m_samples', P_theory_samples', 'Type', 'Pearson');
    R_sim = corr(m_samples', P_sim_samples', 'Type', 'Pearson');
    fprintf('皮尔逊相关系数（理论）：R = %.4f\n', R_theory);
    fprintf('皮尔逊相关系数（模拟）：R = %.4f\n', R_sim);


    figure;
    plot(m_samples, P_theory_samples, '-o', 'LineWidth',1.5); hold on;
    plot(m_samples, P_sim_samples, '-s', 'LineWidth',1.5);
    xlabel('候选人数 m');
    ylabel('单候选人当选概率');
    legend('理论(Binom)', '模拟(MC)');
    title(sprintf('全局敏感性分析：m 对 P 的影响 (R 理论=%.2f, R 模拟=%.2f)',
R_theory, R_sim));
    grid on;



    %% ----------------- 全局灵敏度分析（m 和 n 联合分布） -----------------
    fprintf('\n=== 全局灵敏度分析 (m 和 n 联合分布对 P 的影响) ===\n');
    m_samples_joint = 3:12;
    n_samples_joint = 10:19;
    alpha = 2/3;
    P_theory_joint = zeros(length(m_samples_joint), length(n_samples_joint));
    P_sim_joint = zeros(length(m_samples_joint), length(n_samples_joint));


    for i = 1:length(m_samples_joint)
        for j = 1:length(n_samples_joint)
            m = m_samples_joint(i);
            n = n_samples_joint(j);
            k = ceil(m/2);
            t = ceil(alpha * n);
            p = k / m;
            P_theory = 1 - binocdf(t-1, n, p);
            P_theory_joint(i,j) = P_theory;


            % 蒙特卡洛
            success_count = 0;
            for iter = 1:M
                votes = zeros(1,m);
                for expert = 1:n
                    picks = randperm(m,k);
                    votes(picks) = votes(picks) + 1;
                end
                success_count = success_count + (votes(1) >= t);
            end
            P_sim = success_count / M;
            P_sim_joint(i,j) = P_sim;
        end
    end


    % 修正展开方式，保证与矩阵一致
    [m_grid, n_grid] = meshgrid(m_samples_joint, n_samples_joint);
    m_flat = m_grid(:);
    n_flat = n_grid(:);
```

```
    P_theory_flat = P_theory_joint'; P_theory_flat = P_theory_flat(:);
    P_sim_flat = P_sim_joint'; P_sim_flat = P_sim_flat(:);


    % 计算皮尔逊相关系数
    R_m_theory = corr(m_flat, P_theory_flat, 'Type', 'Pearson');
    R_n_theory = corr(n_flat, P_theory_flat, 'Type', 'Pearson');
    R_m_sim = corr(m_flat, P_sim_flat, 'Type', 'Pearson');
    R_n_sim = corr(n_flat, P_sim_flat, 'Type', 'Pearson');


    fprintf('皮尔逊相关系数（理论）：R_m = %.4f, R_n = %.4f\n', R_m_theory,
R_n_theory);
    fprintf('皮尔逊相关系数（模拟）：R_m = %.4f, R_n = %.4f\n', R_m_sim,
R_n_sim);


    % 绘制 3D 散点图
    figure;
    scatter3(m_flat, n_flat, P_theory_flat, 50, P_theory_flat, 'filled');
    xlabel('候选人数 m');
    ylabel('实到专家人数 n');
    zlabel('单候选人当选概率');
    title(sprintf('全局敏感性分析: m 和 n 联合对 P 的影响 (理论, R_m=%.2f,
R_n=%.2f)', R_m_theory, R_n_theory));
    colorbar;
    grid on;


    figure;
    scatter3(m_flat, n_flat, P_sim_flat, 50, P_sim_flat, 'filled');
    xlabel('候选人数 m');
    ylabel('实到专家人数 n');
    zlabel('单候选人当选概率');
    title(sprintf('全局敏感性分析: m 和 n 联合对 P 的影响 (模拟, R_m=%.2f,
R_n=%.2f)', R_m_sim, R_n_sim));
    colorbar;
    grid on;
```