

NIPT 动态决策优化：生存分析与风险约束

摘要

NIPT（无创产前检测）是一种确定胎儿健康状况的产前检测技术，通过采集母体血液、检测胎儿的游离 DNA 片段、分析胎儿染色体是否存在异常，从而对胎儿的早期健康状况做出判断。围绕 NIPT“何时检测、如何判定”，本文提出一体化框架：在男胎样本上用 Beta(logit)-GAMM 建立比例响应的非线性底图，基于离散时间生存得到逐周达标概率；在 $\alpha = 0.80$ 的达标保障、最小组样本与分组惩罚 (λ) 约束下，以动态规划确定 BMI 自适应分组和最早推荐周；再用大量随机模拟来量化和传递“测量误差”与“判读误差”，进而对结果进行保守修正；最终训练女胎判定器并进行概率校准。全流程按孕妇 ID 分组交叉验证，目标为可解释、可复核、可部署。

针对问题一：构建胎儿 Y 染色体浓度与选取指标的关系模型。首先，根据 **Beta(logit)-GAMM**，对孕妇 ID 设一个随机截距；接着，在得出初步结果之后，进行分组交叉验证与诊断，输出部分效应与达标概率底图供后续决策；最终，得出结论：孕周与 Y 浓度显著正相关 ($edf > 4$, $p < 0.001$)，BMI 越高达标越晚；个体内相关显著 ($ICC \approx 0.25 - 0.30$)。

针对问题二：构建求解最佳 NIPT 时点模型。首先，通过离散时间生存 **GAMM** 模型，由“周危险率”累积得到“检测风险”，从而进一步算出“期望风险”，初步找出最优检测时间；接着，以 BMI 为主因子、在最小组样本与惩罚项 λ 下，动态规划得到四组四时点的全局最优；最后，在保证“达标可靠性”和“最小化检测风险”的双重约束下，得出所求分组及对应优化时点：[20,28]→12 周；[28,32]→12 周；[32,36]→13 周；[36,46]→14 周（由于继续细分不能进一步降低期望风险，故当前四段为最优）。

针对问题三：构建求解多因子与误差传导下的最佳 NIPT 时点模型。在问题二的基础上，不改变问题二的核心框架，考虑测量不准、个体差异等噪声以及保守估计风险，进一步建模。最终，在进行误差的大量随机传播以及引入惩罚合并之后，动态规划收敛为两组两时点：[18.0,20.0]→11 周；[20.5,45.5]→12 周（与问题二不冲突：协变量与误差处理使瘦体型更早满足保障，其余 BMI 合并至 12 周即可达标）。

针对问题四：构建女胎判定与阈值模型。首先，对数据进行预处理和清洗，并使用 **K 折分组交叉验证**的方法保证后续结果的可靠性；接着，构建 **XGBoost 二分类模型**，并使用带权重的对数损失函数处理不平衡数据，基于逻辑回归原理，得到校准后的概率；然后，通过理论阈值和经验阈值加以优化，并构建可解释的基线 **GAM** 模型；最后，对模型进行评估，由 $ROC-AUC=0.732$, $PR-AUC=0.306$, $Brier=0.099$, $LogLoss=0.390$ ，按照代价比 $FN:FP=10:1$ ，选用阈值 0.09，混淆矩阵(TN, FP, FN, TP)=(375,162,20,47)，召回约 70.1%，假阳性率约 30.2%，经 Platt/Isotonic 校准后，可靠度良好。

总之，本文以 GAMM、离散时间生存与动态规划贯通“相关性—时点—稳健性—判定”。在 $\alpha = 0.80$ 与统一约束下，问题二形成四组可交付时点，问题三在多因子与误差传播后得到等效的两组简化策略；问题四实现校准判定与可审计输出，具备临床落地价值。

关键词：NIPT GAMM 离散时间生存 α 保障 动态规划 概率校准

一 问题重述

1.1 问题背景

21、18、13 号染色体的非整倍体（对应唐氏综合征、爱德华综合征、帕陶综合征）被公认为影响胎儿健康的主要风险之一。无创产前检测（NIPT）因其取样便捷、风险较低，已成为孕早中期常用的筛查手段：采集母体血液，检测其中胎儿来源的游离 DNA 片段在不同染色体上的相对比例，观察是否出现异常偏离。在本题设定中，当男胎的 Y 染色体浓度 $\geq 4\%$ 且 X 染色体无异常时，检测结果可视为基本准确；否则，则可能出现判读不稳或未出报告（测序报告）。与此同时，实际临床更关注“早识别、早干预”的原则：12 周内风险较低、13–27 周风险升高、28 周及以后风险明显增大，因此在可保证结果可靠的前提下尽早检测，能够为后续的诊断与干预保留更多时间窗口。

赛题提供的是某地区的真实 NIPT 数据，样本呈现出高 BMI 个体占比偏大的特征；依据经验与数据描述，男胎 Y 浓度与孕周、BMI 存在显著关联，同一孕周下不同体型人群的“达标概率”并不一致^[1]。此外，数据中还记录了测序失败的情况，这类样本多与检测时点偏早或其他不确定因素相关；为增强结论的可信度，部分个体存在重复采血或一次采血多次检测的记录，可用于观察时点、达标与失败之间的关系。总体来看，这一背景决定了两条判读路径的差异：男胎侧重以 Y 浓度达标作为可靠性的门槛，女胎则更多依赖 13/18/21 及 X 等指标的联合判读，并结合测序质控信息控制技术偏差^[2]。以上事实共同构成了本文开展建模分析的实际语境与数据基础。

1.2 问题提出

问题一：围绕男胎样本的 Y 染色体浓度与孕周、体质指数的关系，建立一个既能刻画非线性趋势、又能控制同一孕妇多次检测相关性的模型，给出清晰的效应方向与强度，并产出可用于后续决策的达标概率底图。模型以可解释为先，配合按孕妇分组的交叉验证检验稳定性，为后续章节的时点选择与分组提供可靠依据。

问题二：在问题一的基础上，利用周别达标概率，在体质指数轴上形成若干连续区间，为每一组确定最早且足够可靠的检测周。需要同时顾及“过早”和“过晚”的代价，在满足既定可靠性阈值的前提下，使风险更小、方案更稳，并通过整体优化获得全局一致的分组与推荐时点。

问题三：在问题二的框架上把测量与判读中的不确定性显式纳入，先将噪声传导到达标概率，再以保守下界作为可靠性度量，仍然寻找各组最早可行的检测周与全局最优分组。目标是在存在误差和样本稀疏区的情况下，结论依旧稳健、可复核、可落地。

问题四：在不增加采样成本的前提下，基于现有的染色体指标与质控信息构建女胎异常判定器，强调可解释、可部署与风险可控。训练评估采用按孕妇分组的交叉验证，输出经过校准的概率与与之匹配的阈值，并据此形成阈后性能与代价评估，便于与前述决策流程对接。

二 符号与变量表

在数学建模的研究中，涉及大量参数、变量以及分类标识。清晰的符号与变量定义是准确传达模型逻辑、保障研究可复现的基础。为便于模型构建与算法描述，本节对文中涉及的关键参数符号及变量名称进行统一说明，详见表 1。

表 1 符号变量说明

类别	符号/表达式	解释
索引	$t \in T$	T 代表具体某一周， T 是所有观测集的集合
	$i = 1, \dots, n$	观测索引(抽血记录)
	$j = 1, \dots, J$	受试者索引(孕妇)
	$j(i)$	观测 i 所属的孕妇编号
响应变量	$V_i \in (0, 1)$	男胎 Y 染色体浓度(胎儿分数)
	(Y_i, T_i)	Y 染色体读段数、总读段数
自变量	t_i	孕周(连续值，由“周+天/7”换算)
	z_i	质控/批次协变量
函数	$S_1(\cdot), S_2(\cdot)$	平滑函数(孕周效应与 BMI 效应)
逐周记录标识	$y_{jt} = 1\{T_j = t\}$	描述第 j 个孕妇在第 i 周是否首次达标
	$t \leq \min(T_j, C_j)$	T_j 是第 j 个孕妇的实际达标周， C_j 为最后观测周
核心影响因素	BMI: b_j / b_{jt}	若 BMI 不变， b_j 是第 j 个孕妇的 BMI 值；如果变化， b_{jt} 是其 BMI 值
	z_{jt}	除了 BMI 之外的其他可能影响达标的因素
概率相关	h_{jt}	第 j 个孕妇“在第 t 周之前都没达标”的前提下，第 t 周首次达标的概率，即危险率
	$S_j(t)$	第 j 个孕妇在第 t 周及之前都没达标的概率)

	$F_j(t)$	第 j 个孕妇“在第 t 周及之前就已达标”的概率
分组 相关	$G[b_L, b_R]$	按 BMI 划分的小组, b_L 为下限, b_R 为上限。
	$p_G(t)$	某 BMI 组 G 中, 所有孕妇“在第 t 周及之前达标”的概率的平均值
代价 相关	$C_{\text{过早}}, C_{\text{过晚}} > 0$	$C_{\text{过早}}$ 是“检测时间太早”的成本; $C_{\text{过晚}}$ 是“检测时间太晚”的成本, 两者都是正数
DP 相关	$R_{i:j}$	从孕妇 i 到孕妇 j 为一个 BMI 区间, 满足“达标概率要求”的前提下, 选择最佳检测时间的“最小总代价”; 若不满足要求, 代价记为无穷大。
	λ	分组过多的惩罚成本
	$DP[j]$	分组之后满足要求的最小总代价
原始 观测	$Z_{13,i}, Z_{18,i}, Z_{21,i}, Z_{x,i}$	四条染色体的 Z 值
	Z_i	染色体向量 $(Z_{13,i}, Z_{18,i}, Z_{21,i}, Z_{x,i})^\top$
	a_k	截距 (第 k 条染色体)
	reads_i	总读段数
	reads_ratio_i	有效/目标读段比例

三 模型建立

本章在赛题设定下形式化描述 NIPT 场景: 定义采血孕周与 BMI 等自变量, 明确“男胎 Y 浓度 $\geq 4\%$ (且 X 正常) 为基本准确”的达标口径, 给出达标概率/失败概率与女胎异常判定的模型框架与变量表, 并说明数据预处理与质控规则, 为后续求解奠定统一基线。

3.1 问题一

3.1.1 模型假设

1) 响应分布假设

V_i 为比例型，我们在主分析中采用 Beta 回归。若观测触及 0 或 1 的边界，先应用 Smithson - Verkuilen 压缩

$$V'_i = \frac{V_i(n-1) + 0.5}{n} \quad (1)$$

再入模；当具备读段计数(Y_i, T_i)时，以二项模型作为稳健性对照。

2) 重复测量相关性假设

同一孕妇的多次检测存在组内相关性。设随机截距 $u_{j(i)} \sim \mathcal{N}(0, \sigma^2)$ ，在条件独立前提下吸收个体差异，避免系数偏倚。

3) 平滑函数假设

对孕周与 BMI 使用平滑项 $S_1(\cdot), S_2(\cdot)$ (薄板样条/TPRS)，以捕捉潜在非线性与边际递变效应，而非强行设定线性。

4) 协变量控制假设

纳入 z_i (如质控/批次) 以校正系统偏差，提升解释力与外推稳定性。

3.1.2 模型构建

1) 数据治理

仅保留男胎样本；孕周以“周+天/7”转为连续量；

BMI 如缺身高/体重则按既定缺失策略处理并记录审计；

为数值稳定可对 t_i, b_i 做 Z-score 内部标准化，预测时回写原量纲；

对 V_i 进行 SV 压缩 (若出现 0/1)；

参照质控指标 (GC 含量、唯一比对数、比对比例、过滤读段占比等) 设定客观门槛 ($IQR \times 1.5 + \text{经验上限}$) 并保留清洗审计表。在变量选择方面，我们不仅纳入了题目要求的孕周与 BMI，还从测序质控和个体特征中构建了候选池。在控制核心变量的前提下，先进行单变量平滑筛查，再结合分组交叉验证，避免信息泄露；最终仅将那些能够稳定提升拟合度和外推能力的质控变量纳入模型。

2) 基线模型

在比较线性回归、随机森林与不含随机效应的 GAM 后，我们最终采用 **Beta-GAMM**：既能在比例型响应上刻画孕周与 BMI 的非线性，又以随机截距吸收同一孕妇的重复测

量相关，同时保留可解释的显著性检验与更稳健的外推能力。模型如下：

$$V_i \sim \text{Beta}(\mu_i, \phi), \quad \text{logit}(\mu_i) = \beta_0 + s_1(t_i) + s_2(b_i) + \gamma^\top z_i + u_{j(i)}, \quad (2)$$

$$u_{j(i)} \sim \mathcal{N}(0, \sigma^2), \quad (3)$$

其中 s_1, s_2 为薄板样条，基数 k 取 7 – 10 以平衡灵活度与过拟合风险。平滑度与惩罚参数通过 REML 自动选择。

3) 显著性与评估

整体拟合：记录偏差解释率、调整后 R^2 、(R) EML 对数似然等；

项层检验：平滑项报告 edf 与近似 p 值（原假设：无非线性效应）；参数项报告估计值、SE、 t/z 与 p 值；

诊断：残差分布、影响点、concurvity 与随机效应 Q-Q，必要时做分层交叉验证。

4) 可视化

输出部分效应曲线（孕周、BMI）与随机效应 Q-Q 图，配合关键效应量和区间，便于临床解读。

3.1.3 模型公式

1) Beta 回归层与边界处理

若出现边界，先做 SV 微调 $V_i \rightarrow V'_i \in (0, 1)$ ，以满足 Beta 定义域；随后令

$$V'_i \sim \text{Beta}(\mu_i, \phi), \quad \text{logit}(\mu_i) = \eta_i. \quad (4)$$

其中 η_i 为线性预测子。

2) 固定效应+随机效应

$$\eta_i = \beta_0 + s_1(t_i) + s_2(b_i) + \gamma^\top z_i + u_{j(i)}, \quad u_{j(i)} \sim \mathcal{N}(0, \sigma^2). \quad (5)$$

s_1, s_2 采用 TPRS 基；必要时对 z_i 使用哑变量/标准化处理。

3) 精度参数与 Beta 形参

记精度 $\phi > 0$ ，则 Beta 形参 $\alpha = \mu\phi$, $\beta = (1 - \mu)\phi$ 。可按常数或与协变量相关两种设定比较。

4) 平滑项与惩罚

令 $s(\cdot) = \sum_m \theta_m B_m(\cdot)$, 惩罚为

$$J(\theta) = \lambda \theta^\top S \theta, \quad (6)$$

其中 λ 控制平滑强度, S 为“弯曲度”惩罚矩阵。

5) 惩罚似然与估计

最大化惩罚(受限)对数似然

$$\ell_p = \ell(\beta, \gamma, \theta, \phi) - \frac{1}{2} \sum \lambda \theta^\top S \theta, \quad (7)$$

并采用 REML 同时估计 λ, ϕ 与系数; 模型比较以“最简充分”为原则。

3.2 问题二

3.2.1 模型与基本设定

我们以整周为时间刻度, 记观测周为 $t \in \{t_{\min}, \dots, t_{\max}\}$ 。当某周首次观测到指标 V_t 超过阈值 $\tau = 0.04$ 时, 记为事件发生时间 T 。若在观察期内未发生事件, 则按右删失处理, 以最后一次观测时点作为截止。给定已纳入的影响因素后, 假设删失与事件时间条件独立; 同一孕妇在不同周的事件发生彼此独立。

为刻画“周达标概率”与影响因素的非线性关系, 并吸收个体差异, 我们采用离散时间生存的 **Beta(logit)-GAMM** 思路:

- 1) 以 logit 链接描述“第 t 周首次达标”的危险率;
- 2) 设定孕妇随机截距 $u_j \sim \mathcal{N}(0, \sigma_u^2)$ 处理个体内相关;
- 3) 同时考虑孕周效应 $f_1(t)$ 、BMI 效应 $f_2(b)$ 及其交互 $f_{12}(t, b)$ 。

模型核心形式为:

$$\Pr(T = t | T \geq t) = \text{logit}^{-1} \{ f_1(t) + f_2(b) + f_{12}(t, b) + u_j \} \quad (8)$$

由危险率可得到生存函数与达标累计概率:

$$S_j(t) = \prod_{k=1}^t (1 - h_{jk}), \quad F_j(t) = 1 - S_j(t). \quad (7)$$

选定检测周 t^* 后,

“检测过早”概率为 $1 - F_j(t^*)$,

“检测过晚”概率为 $F_j(t^* - 1)$ 。

以两类代价 $C_{\text{早}}, C_{\text{晚}} > 0$ 权衡, 期望风险为

$$E[c(T, t^*)] = C_{\text{早}} [1 - F_j(t^*)] + C_{\text{晚}} F_j(t^* - 1) \quad (8)$$

3.2.2 约束与分组求解

1) **α -保障。**按 BMI 分组 $G = [b_L, b_R]$ 后, 计算组内达标均值

$$p_G(t) = \frac{1}{|G|} \sum_{j \in G} F_j(t). \quad (9)$$

为避免极端样本把均值“抬高”, 以 $p_G(t)$ 的后验 5% 分位数作为保守下界, 并要求在候选检测时点 t^* 下该下界不低于给定阈值 α (即“达标可靠性”得到保证)。

- 2) **风险最小化。**在满足 α -保障的前提下, 最小化组内期望风险; 实践中将“晚测代价”设为“早测”的约 3 倍, 使目标更贴近临床取舍。
- 3) **动态规划分组。**将样本按 BMI 升序, 对任意 BMI 区间 $i:j$:
 - a) 若存在满足 α -保障的最优检测周 t^* , 记其最小期望风险为 $R_{i:j}$;
 - b) 若不存在可行 t^* , 令 $R_{i:j} = +\infty$ 。

引入分组惩罚 λ 控制组数, 利用 DP 递推得到全局最优划分:

$$\text{DP}[j] = \min_{i \leq j} \{\text{DP}[i-1] + R_{i:j} + \lambda\}, \quad (10)$$

并回溯得到每个 BMI 组的最优检测时点。最终输出各组的 (b_L, b_R) 、推荐 t^* 、达标保障及对应风险。

3.3 问题三

在不改变“按 BMI 分组+DP 选最早可行时点”的基础上, 问题三考虑测量误差与观测偏差, 对问题二的结果做保守修正, 使结论更贴近实际。以下假设与记号保持与前文一致。

3.3.1 误差建模

- 1) 孕周测量误差: 计划检测周 t 的实际记录为 $t' = t^* + \varepsilon_t$, 取 $\varepsilon_t \in (-\Delta_t, \Delta_t)$, 默认 $\Delta_t = 0.5$ 周。
- 2) BMI 测量误差: $b' = b + \varepsilon_b, \varepsilon_b \sim \mathcal{N}(0, \sigma_b^2)$ 。
- 3) 其他协变量同理: $Z' = Z + \varepsilon_Z$ 。

4) 检测判读误差: 将真实达标概率 p 修正为观测达标概率

$$p_{\text{obs}}(t^*|b, Z) = (1 - \text{FNR})p + \text{FPR}(1 - p) \quad (11)$$

其中 FNR/FPR 为假阴/假阳率 (由质控或文献/数据估计)。

3.3.2 周别危险率

在问题二的离散时间生存框架上, 引入更多因素与交互项:

$$\text{logit } h_{jt} = f_1(t) + f_2(b_j) + f_{12}(t, b_j) + \sum_{r=1}^q g_r(Z_{jt}^{(r)}) + \sum_{r < s} g_{rs}(Z_{jt}^{(r)}, Z_{jt}^{(s)}) + w_{jt}^\top \gamma + u_j, \quad (12)$$

其中 u_j 为孕妇随机截距; 由 $\{h_{jt}\}$ 得生存函数 $S_j(t) = \prod_{k \leq t} (1 - h_{jk})$ 与累计达标

$$F_j(t) = 1 - S_j(t).$$

3.3.3 组内计算

对每个 BMI 组 $G = [b_L, b_R]$, 进行 Monte-Carlo 抽样: 在 (t', b', Z') 的误差分布下计算组内观测达标概率 $p_G^{\text{obs}}(t)$ 。为保证可靠性, 取其后验 **5% 分位数**作为保守下界 $\text{LB}_G(t)$, 并施加

$$\alpha\text{-保障: } \text{LB}_G(t^*) \geq \alpha.$$

这意味着以 95% 置信度, 组内达标率不低于阈值 α 。

3.3.4 风险与最优时点

在满足 α -保障的前提下, 按“过早/过晚”代价 $C_{\text{早}}, C_{\text{晚}} > 0$ 计算组内期望风险 (可在 p_{obs} 上附加轻度不确定性惩罚), 取最早满足约束且风险最小的 t 。在 BMI 轴上, 预计算任意连续区间 $i:j$ 的最小可行风险 $R_{i:j}$ (若不可行则 $+\infty$), 引入组数惩罚 λ , 采用动态规划:

$$\text{DP}[j] = \min_{i \leq j} \{\text{DP}[i-1] + R_{i:j} + \lambda\} \quad (13)$$

求得全局最优分组与各组推荐周。输出每组 $[b_L, b_R]$ 、推荐 t^* 、保障下界与对应风险, 用作结果与决策的依据。

3.4 问题四

目标是在不增加采样成本的前提下，构建**女胎异常判定器**并给出可复核、可解释、可部署的流程。我们使用的特征仅包含与标注无直接因果联动的观测量：染色体 Z 值（13/18/21/X）、测序质控（GC 含量、总读段 reads、有效读段比例 reads_ratio）、孕妇 BMI 与年龄等；以**孕妇 ID**分组做 K 折交叉验证，阻断同一受试者的“数据泄漏”。异常样本占比偏低，训练时采用**代价感知/带权对数损失**以凸显少数类。为保障可解释性，构建可解释的 GAM 作为基线模型，并以**SHAP (TreeSHAP)**解释主模型的特征贡献。

3.4.1 数据预处理与清洗

只保留与判定相关的观测字段；数值缺失用**中位数**填补，重尾变量视情况做**截尾或对数变换**；类别变量进行合适的**编码**。所有预处理步骤在**训练折内拟合、折外应用**，避免信息泄泄。

3.4.2 分组交叉验证

采用 K 折分组 CV (按**孕妇 ID**)。每一折训练主模型，得到**折外未校准分数** s_i ；所有折外分数拼接形成全体样本的 out-of-fold 分数，用于后续**概率校准与阈值选择**。

3.4.3 主模型 (XGBoost 二分类)

以**带权对数损失**为目标：

$$L(f) = \sum_{i=1}^n w_i [-y_i \log \sigma(s_i) - (1-y_i) \log \{1 - \sigma(s_i)\}] + \Omega(f), \quad \sigma(s) = \frac{1}{1+e^{-s}},$$

其中 $\Omega(f) = \gamma T + \frac{\lambda}{2} \sum_{t=1}^T \|w^{(t)}\|_2^2$ 用于控制树复杂度， w_i 体现类别不平衡的代价设定

(异常权重更高)。超参以折内网格/贝叶斯搜索获得。

3.4.4 概率校准 (折外 Platt scaling)

在**折外**分数 s 上拟合逻辑回归 $\hat{p} = \sigma(as + b)$ ，将分数映射为**已校准概率**，并采用**嵌套交叉验证**避免再次泄漏。若发现阈值调优与理论阈值偏差较大，则返回校准步骤微调。

3.4.5 阈值设定 (理论+经验微调)

设“假阴/假阳”代价分别为 $C_{\text{FN}}, C_{\text{FP}}$ ，理论最优阈值

$$\tau^* = \frac{C_{\text{FP}}}{C_{\text{FP}} + C_{\text{FN}}} \tag{14}$$

最小化期望代价；在校准概率上以

$$\text{Risk}(\tau) = C_{\text{FP}} \Pr(\hat{y}=1, y=0 | \tau) + C_{\text{FN}} \Pr(\hat{y}=0, y=1 | \tau) \tag{15}$$

做窄幅经验微调。若 τ 与 τ^* 差距显著，优先改进校准而非盲调阈值。

3.4.6 可解释性与基线对照

建立 **GAM** 基线： $\text{logitPr}(y=1|x) = \alpha + \sum_k s_k(x_k)$ ，报告平滑项的 edf 统计量/p 值并绘制平滑曲线，用以验证主效应与非线性方向是否与主模型一致；对 XGBoost 计算 **SHAP** 值 $s(x) = \phi_0 + \sum_k \phi_k(x)$ ，给出全局重要性与局部决策路径，确保关键规则可解释、可核对。

3.4.7 评估与输出

评估在折外/测试上完成，主指标包含 AUC、PR-AUC、灵敏度/特异度、Brier 分数与校准曲线；报告分母与置信区间以便复核。最终输出：已校准概率、推荐阈值、阈后混淆矩阵与误判成本估计；附基线 GAM 与 SHAP 图表，便于方法章节与结果章节一一对应，满足论文“简洁呈现+全量可复核”的体例要求。

四 求解算法

本章给出时点选择与异常判定的实现方案：达标概率的拟合与阈值搜索流程，失败样本/重复检测的处理策略，及女胎判定器的训练、校准与阈值设定。同步记录软件版本、参数范围与停止准则，并设计必要的灵敏度与外推检验。

4.1 问题一的求解算法

为了解决男胎 Y 染色体与孕周和 BMI 的相关性问题，并且为了消除因同一孕妇及逆行多次测量而产生的重复效应，本题采用广义可加混合模型（GAMM）步骤如下：

4.1.1 数据清洗与标准化

保留男胎样本，孕周统一为连续变量（周+天/7），体质指数缺失值用体重/身高²回填。若观测 Y 出现 0 或 1，则采用 Smithson–Verkuilen 压缩 $y' = (y(n-1) + 0.5)/n$ 以满足 Beta 分布定义域；若无 0/1，则保持原值。所有变量均以原量纲入模（孕周：周；BMI： kg/m^2 ），仅为数值稳定性在内部做标准化，预测时回写到原量纲。

对 Y 浓度进行 Smithson–Verkuilen 映射，确保数值落在(0,1)区间。

根据测序质量指标（GC 含量、唯一比对数、比对比例、过滤读段占比等）设定阈值，剔除极端异常值。清洗后得到的标准化数据集即为建模输入。

4.1.2 模型设定

选用 Beta 分布、logit 链接函数，以孕周和体质指数为平滑项：

$$Y' \sim s(\text{孕周}, k=9) + s(\text{体质指数}, k=7) + s(\text{孕妇ID}, bs=\text{"re"})$$

其中 Y' 为映射后的 Y 浓度，孕妇 ID 作为随机截距以吸收个体差异。

4.1.3 模型估计与选择

使用 REML 方法估计参数，并通过有效自由度（EDF）与近似 p 值检验平滑项显著性。

采用 AIC 与 REML 准则在多组候选模型间比较，遵循“最简充分”原则，确定最终模型结构。

4.1.4 预测与产出

在若干关键孕周与体质指数水平上进行点预测，输出预测值与置信区间，生成“关键点预测”与“达标概率矩阵”的输出文件（如 Q1_关键点_预测.xlsx），用于后续章节展示与讨论；本节仅描述流程，不呈现具体数值或图形。

在孕周一体质指数二维网格上计算 $P(Y \geq 4\%)$ ，得到达标概率矩阵。

4.2 问题二的求解算法

在 BMI 轴上分组并给出各组的最早可用检测周 t^* （满足“达标概率 $\geq \alpha$ ”的保障），本题采用离散时间生存思路与动态规划分组。核心量为：周别“首次达标”概率（危险率） $h(t | \text{BMI})$ ，存活 $S(t) = \prod_{\tau \leq t} [1 - h(\tau)]$ ，累计达标 $F(t) = 1 - S(t)$ 。在满足 α -保障的前提下，结合“早测/晚测”代价设计与组数惩罚，用 DP 求得 BMI 分组与各组 t^* 的全局最优解。步骤与产物如下：

1) 构造 person-period 长表（文件：Q2_person_period.csv）

将同一孕妇的多次观测按“人×整周”展开；对每周打上“是否首次达标”标记（只在首次过线的那周记 1，其余为 0；未过线者右删失）。此表是后续一切统计的“底座”。

2) 估计经验危险率并做校核（文件：Q2_经验危险率_BMI30±1.csv/BMI35±1.csv/BMI40±1.csv）

在若干 BMI 小窗内，逐周统计“风险集人数 $N_{\text{risk}}(t)$ ”与“本周首次达标人数 $E(t)$ ”，得：

$$h_{\text{emp}}(t) = E(t) / N_{\text{risk}}(t). \quad (16)$$

该经验曲线与后续平滑/建模结果对照，用于验证形状与波动是否合理（样本稀疏处允许抖动）。

3) 在 $t \times \text{BMI}$ 棚格上滚出累计达标概率 (文件: Q2_生存底图_python.xlsx)

以每周首次达标的 $h(t | \text{BMI})$ (可用经验率平滑或由 Q1 模型间接推得) 连乘得到

$$F(t | \text{BMI}) = 1 - \prod_{\tau \leq t} (1 - h(\tau | \text{BMI})), \quad (17)$$

形成二维概率底表。由此派生三张图:

- a) 累计达标热力/等高线 (如图 1): 展示“早孕+高 BMI 难达标，孕周增大更易达标”的全貌;
- b) 风险集热图 (如图 2): 显示各格点仍在风险集的人数，提醒读者哪些区域样本稀疏;
- c) 遮罩版热图 (如错误!未找到引用源。): 仅呈现“支撑度 $\geq s_0$ ” (如 ≥ 3 人) 的格点，避免在无人区读图做决策。

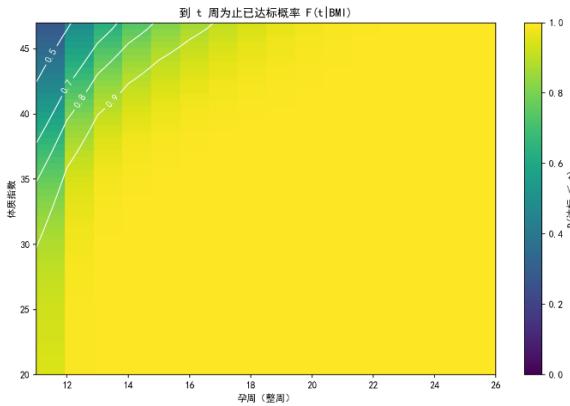


图 1 累计达标概率热图与等高线图

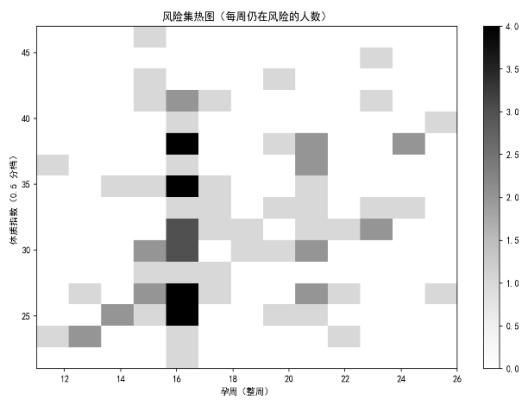


图 2 风险集人数热图

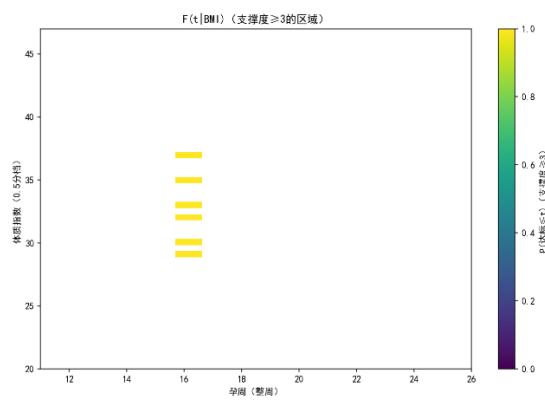


图 3 累计达标概率热图

4) α -保障与最早时点搜索 (组内)

给定保障阈值 α (如 0.8/0.9)，先确定 BMI 轴上的候选分组方案 (等宽或按分位；

设置每组“最少样本数”门槛)。在每个组内, 沿孕周从早到晚扫描 \mathbf{F} 的组内统计量(如组均值或其保守下界/分位数), 找到第一个满足 $F \geq a$ 的周, 记为该组的 t^* 。若需要兼顾“早测/晚测”权衡, 可设代价

$$E[\text{cost}(t)] = c_{\text{early}} \cdot (1 - F(t)) + c_{\text{late}} \cdot F(t-1), \quad (18)$$

并以“在满足 α 前提下 $E[\text{cost}]$ 最小”来挑 t^* 。

5) DP 选全局最优分组

记任意 BMI 区间 $[i, j]$ 的“可行最小代价”为 $C(i, j)$ (若无周使 $F \geq a$ 则 $C = +\infty$)。设组数惩罚 λ , DP 递推

$$\text{DP}[j] = \min_{i \leq j} \{\text{DP}[i-1] + C(i, j) + \lambda\}, \quad (19)$$

回溯得到组边界与各组 t^* 。输出结果表 (BMI 区间 | t^* | 该点保障概率 | 组内样本数), 并将分段 t^* 折线叠加到热图上, 形成策略可视化。

4.3 问题三的求解算法

本问在问题二的离散时间生存框架上, 纳入多因素与测量误差, 并在 BMI 轴上做保障优先、尽量提前的最优分组与检测时点选择。

4.3.1 数据形状与人×周展开

以整周为时间刻度, 按“孕妇 ID×周”生成 person-period 长表(Q3_person_period.csv), 标注“本周是否首次达标”(0/1)与删失。变量包含: 孕周 t 、BMI、及可用协变量 Z (如批次/质控项)。

4.3.2 周别危险率建模 (多因素)

- 1) 拟合周别“首次达标”概率 $h(t | \text{BMI}, Z)$ (Logit-GAM/GAMM; 可含 $s(t), s(\text{BMI}), te(t, \text{BMI})$ 与随机截距)。
- 2) 由 $S(t) = \prod_{\tau \leq t} [1 - h(\tau | \cdot)]$ 得到累计达标 $F(t) = 1 - S(t)$ 。

4.3.3 测量误差与不确定性传播 (MC)

- 1) 设定 BMI/孕周的测量误差分布与阈值场景 (如 3.8/4.0/4.2%)。
- 2) 进行 Monte-Carlo: 在每次抽样的 (t, BMI, Z) 上重算 h, S, F , 得到 F 的分布与保

守下界。

3) 汇总为栅格底表 Q3_F_grid.xlsx。

4.3.4 支撑度与可用域约束

计算每个(t , BMI)的风险集人数与邻域合并后的支撑度；对低支撑格点加掩膜，仅在“样本充足”的区域内允许读图与决策。该约束同时用于 DP 的可行性判定。

4.3.5 组内 α -保障与最早时点搜索

给定保障阈值 α （如 0.8/0.9）与每组最小样本数门槛。在候选 BMI 组内，对孕周从早到晚扫描保守下界 $F_{lb}(t)$ ，取首个满足 $F_{lb}(t) \geq \alpha$ 的 t 作为该组候选 t^* 。

4.3.6 代价与动态规划分组

若需兼顾“过早/过晚”的代价，定义组内期望代价 $E[\text{cost}(t^*)]$ ，在满足 α 的前提下最小化之。

预算算任意连续 BMI 区间 $[i, j]$ 的可行最小代价 $C(i, j)$ ；设组数惩罚 λ ，DP 递推

$$DP[j] = \min_{i \leq j} \{DP[i-1] + C(i, j) + \lambda\}, \quad (20)$$

回溯得到全局最优分组边界与各组 t^* ，导出 Q3_DP_分组与最佳时点.csv。而 Q3_F_grid.xlsx（含 MC 后 F 的点估计与下界）、Q3_DP_分组与最佳时点.csv（组区间 | t^* | 保障概率 | 组内 N），以及对应热图/遮罩/风险集图用于后续章节引用。

4.4 问题四的求解算法

仅针对女胎样本(Y 相关列为空)，以 AB 列的 13/18/21 染色体非整倍体为标签（任一标注即阳性），综合 Z 值(13/18/21/X)、测序质控（读段量/比对比例/重复比例/过滤比例、GC 与各染色体 GC)、孕周、BMI、年龄等信息，给出可解释、可部署的三分判定(高风险/低风险/不确定-复检)。流程与前文一致采用按孕妇分组的数据组织与交叉验证，避免同一受试者信息泄漏。

4.4.1 数据标签

过滤女胎；以 AB 列是否含 13/18/21 作为阳性标签；统一列名、量纲与缺失处理。设定 Group=孕妇 ID，后续所有训练/校准/阈值选择均在 GroupKFold 框架内完成，防止同一孕妇多次记录跨折泄漏。

4.4.2 规则层

1) 强阳性：若 $Z_{21} \geq 3 \vee Z_{18} \geq 3 \vee Z_{13} \geq 3 \rightarrow$ 直接判高风险；

- 2) 灰区：任一 $-2.5 \leq Z < 3$ → 标为不确定（复检）；
- 3) 质量兜底：读段/比对/重复/过滤/GC 等任一超阈（按质控门槛）→ 不确定（复检）。

规则层保证与临床阈值法一致，并先剔除不合格样本。

4.4.3 模型层

- 1) 对“非强阳且质量合格”样本，训练概率分类器 $p = \Pr(\text{非整倍体} | \text{全部特征})$ ：
 - a) 基线：L1-Logistic（可解释、稳健）；
 - b) 主模型：梯度提升树/树模型（捕捉非线性与交互），内置类不平衡处理。
- 2) 概率校准（Platt/Isotonic）在折外分數上进行，获取可靠概率以便阈后决策。
- 3) 全程使用 **GroupKFold**（按孕妇）进行训练与折外预测拼接，避免任何信息前泄。

4.4.4 阈后决策

- 1) 设目标灵敏度优先（或给定代价比 $C_{FN} \gg C_{FP}$ ），在校准概率上选取 τ_{high} 、 τ_{low} ：
 - a) $p \geq \tau_{high} \rightarrow \text{高风险}$ ；
 - b) $p \geq \tau_{high} \rightarrow \text{低风险}$ ；
 - c) 其余 $\rightarrow \text{不确定（复检）}$ 。
- 2) 与问题三时点保障联动：若孕周 $< t^*(\text{BMI})$ （上一向给出的该 BMI 组最早可行时点），则一律不输出“低风险”，统一并入口径为不确定（复检），确保时点过早不“误报正常”。

输出 Q4_pred_oof.csv；后续“结果与分析”与“模型评价”只需引用本节流程与对应产物，不再重复方法细节。

五 结果与分析

本章按“达标一时点一判定”的路径呈现证据：先报告分层达标率与最早可用时点，再给出失败率与等待代价的权衡，最后展示女胎判定的 ROC/PR 与校准曲线，并进行分组对照与消融。所有图表附分子/分母或区间，确保结论可核验。

5.1 问题一结果分析

5.1.1 模型与拟合优度

我们对 Y 染色体浓度（比例型）建立了 Beta 回归的广义可加混合模型（logit 链接），

包含孕周与 BMI 的平滑项以及孕妇 ID 的随机效应。模型整体拟合度较强：偏差解释率约 82%、调整后 $R^2 = 0.765$ ，样本量 $n = 1082$ ；孕周与 BMI 的平滑项显著（见表 2 GAMM 模型显著性与拟合优度），满足题意对“关系模型并检验显著性”的要求。最终模型采用 Beta(logit)-GAMM，仅保留孕周、BMI 平滑与孕妇 ID 随机截距；候选的质控/批次项因未提升拟合度，按“最简充分”原则未纳入。

表 2 GAMM 模型显著性与拟合优度

项目	统计量	结果
偏差解释率	Deviance explained	82%
调整后 R^2	Adj. R^2	0.765
样本量	n	1082
孕周效应	edf	4.182
孕周效应	p 值	$< 2e - 16$
BMI 效应	edf	1.754
BMI 效应	p 值	0.00294
随机效应 (孕妇 ID)	std.dev	0.415
随机效应 95%CI	lower-upper	0.377 – 0.456

5.1.2 显著性检验（主效应）

- 1) 孕周：平滑项自由度 $\text{edf} \approx 4.18$, $p < 2e - 16$, 为极显著正效应。浓度随孕周单调上升。
- 2) BMI: $\text{edf} \approx 1.75$, $p \approx 0.003$, 为显著负效应。BMI 越高, 整体浓度水平越低。

5.1.3 个体差异（随机效应）

孕妇 ID 的随机效应显著 ($\text{std.dev} \approx 0.415$, 95%CI 约 0.377–0.456), 提示不同个体存在系统差异, 必须建模控制。

5.1.4 图形证据（部分效应）

在主效应 **s(孕周)**、**s(体质指数)** 与随机截距 **s(孕妇 ID)** 的基础上, 我们还加入了可收缩的质量/母体因子平滑 (select=TRUE), 并显式建模 **ti(孕周×体质指数)** 的非线性交互; 模型估计使用 **REML**, 其余协变量在预测与等高线底图中取样本中位数/众数以维持可读性。该设置与脚本中的“多因子 Beta-GAMM”一致。

- 1) 孕周曲线单调上升、后期更陡, 结论与显著性检验一致; 体质指数呈整体负效应; **ti(孕周×BMI)** 显示在早孕周区域对 BMI 更敏感 (等高线更密)。
- 2) 质量/母体因子:
 - a) 被过滤读段比例与检测质量主成分 **1/2** 呈非线性影响 (在极端区间偏离 0), 提示测序质量对阈值达成存在边缘影响;
 - b) 年龄表现为轻度负向趋势;
 - c) 比对比例、重复比例、GC 含量、log 原始读段数等项在本样本与当前惩

罚强度下接近零效应(灰带跨0、平滑被收缩),说明其独立贡献有限。

3) 随机效应 Q-Q 图近似沿对角线,符合正态假设,模型设定合理。

这些图形证据与“主效应显著、个体差异显著、总体拟合度高(解释率 $\approx 82\%$ 、调整后 $R^2 \approx 0.765$)”的量化结果一致,不改变核心结论。

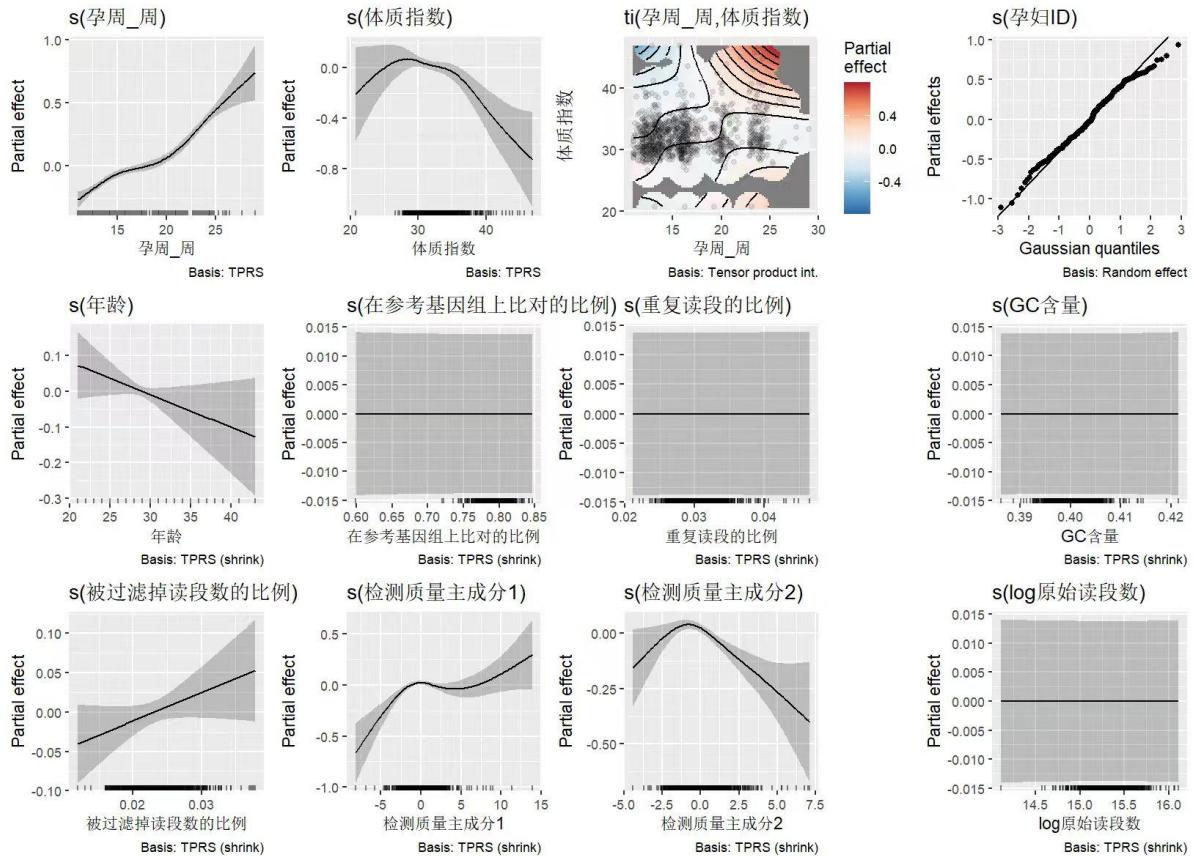


图 4 多因子 Beta(logit)-GAMM 的部分效应与交互总览

5.1.5 解释与启示

- 1) 孕周是主驱动: Y 浓度随孕周提升而稳步增加,后期更快;模型的非线性刻画($edf > 1$)证实“并非直线,而是逐步加速”的趋势。
- 2) BMI 抑制效应真实存在: BMI 的负向影响显著,解释为高 BMI 可能带来胎儿游离 DNA 比例相对更难达到阈值的情况,为后续“分 BMI 组确定 NIPT 时点”提供依据。
- 3) 个体差异不容忽视: 显著的随机效应说明“同孕周、同 BMI 的不同孕妇”浓度基线仍可不同,这也解释了为什么临床策略不能“一刀切”。

5.1.6 小结

- 1) 孕周 $\uparrow \rightarrow$ Y 浓度显著 \uparrow (非线性、后期更陡);
- 2) BMI $\uparrow \rightarrow$ Y 浓度显著 \downarrow ;
- 3) 个体差异显著,需要随机效应;

- 4) 拟合度高(解释率 $\approx 82\%$ 、 $\text{Adj}-R^2 \approx 0.765$)，模型可靠，可作为问题二/三分组与阈值决策的统计基础。由本问的 $\mu(t, \text{BMI})$ 与精度 ϕ 计算 $P(Y \geq 4\%)$ ，形成 $\text{BMI} \times \text{孕周}$ 的达标概率等高线底图，见图 5。

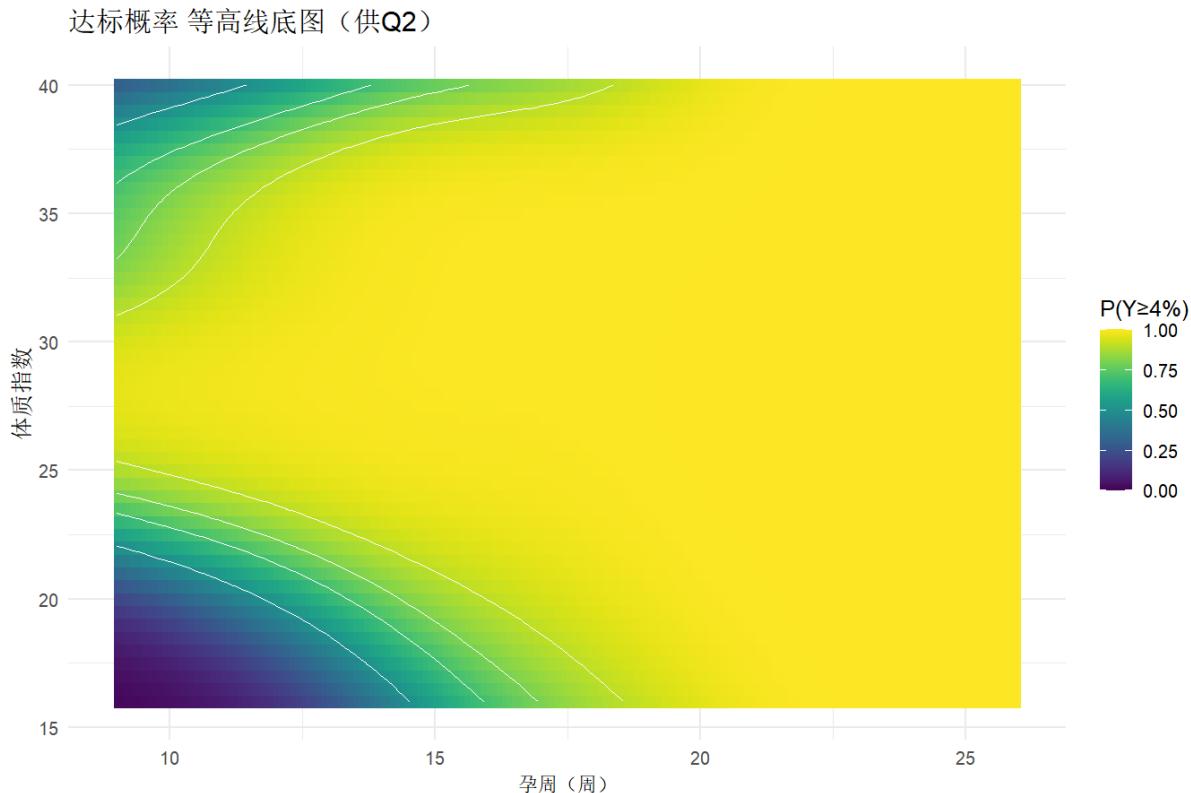


图 5 达标概率等高线底图

5.2 问题二结果分析

本问目标是在 **BMI** 轴上分组，并为各组给出在达标概率达到阈值 α （如 0.8/0.9）时的最早检测孕周 t^* 。我们据 **person-period** 长表计算“每周第一次达标”的危险率 $h(t|\text{BMI})$ ，连乘得到累计达标概率 $h(t|\text{BMI})$ ，形成 **F-栅格底表** (**Q2_生存底图_python.xlsx**)，并产出三张核心图（见 4.2 的图 1、图 2、错误!未找到引用源。）用于展示与校核。在 $\alpha=0.80$ 的保障约束、最小组样本与分组惩罚下，DP 的全局最优为四组四时点，如表 3。

表 3 结果摘要

组别	BMI 区间	推荐时点 (周)
G1	[20, 28)	12
G2	[28, 32)	12
G3	[32, 36)	13
G4	[36, 46]	14

5.2.1 总体趋势（图 1）

$F(t | \text{BMI})$ 随孕周单调上升，且 **BMI** 越高、早孕周的达标概率越低；等高线 (0.5/0.7/0.8/0.9) 呈“左下→右上”走势：孕周增加可弥补高 **BMI** 带来的不利。该形态与 Q1 的结论一致（孕周正效应、**BMI** 负效应）。

5.2.2 支撑度与可用区域（图 2 和错误!未找到引用源。）

风险集热图显示“仍未达标”的样本主要集中在少数周-BMI 区域，其他区域样本极少或为 0；因此我们在遮罩版热图中只保留“风险集人数 $\geq s_0$ ”的格点，用于限制读图与决策只在有数据支撑的区域，避免外推偏乐观。

5.2.3 与经验危险率的交叉验证

在若干 **BMI** 小窗内，直接从 person-period 表计得经验危险率 $h_{\text{emp}}(t) = E(t)/N_{\text{risk}}(t)$ 。这些经验点与平滑后的 $h(t | \text{BMI})$ 和由其累乘得到的 $F(t | \text{BMI})$ 形状一致，但在样本稀疏处存在预期的抖动，进一步强调了遮罩读图的必要性与保守性。

5.2.4 对推荐时点 t^* 的含义

在任一 **BMI** 分组内，沿孕周从早到晚扫描 $F(t)$ ，取第一个满足 $F(t) \geq \alpha$ 的 t 作为该组 t^* ；如需同时考虑“过早/过晚”的代价，可在满足 α 的前提下最小化期望风险。对所有候选分组方案，使用动态规划在“ α -保障+组数惩罚+最少样本数门槛”下择优，得到组边界与各组 t^* 的全局最优。

5.3 问题三结果分析

本问在多因素+测量误差的 MC 框架下，先由离散时间生存模型得到周别危险率 $h(t | \text{BMI}, Z)$ ，再对 (t, BMI, Z) 进行抽样传播，汇总成 $F(t | \text{BMI})$ 栅格，用作分组与时点选择的量化依据；算法流程与参数口径见前文“求解算法·问题三”。

5.3.1 整体趋势

如 F 随孕周单调上升；中低 **BMI** ($\approx 20-35$) 在人群主密度区 **13-14 周已接近 1**，极高 **BMI** (≥ 40) 在 **12 周附近爬升较慢**，随后随孕周明显追平。这与题面“仅按 **BMI** 分组、尽量提前但需保障准确性”的设定一致。

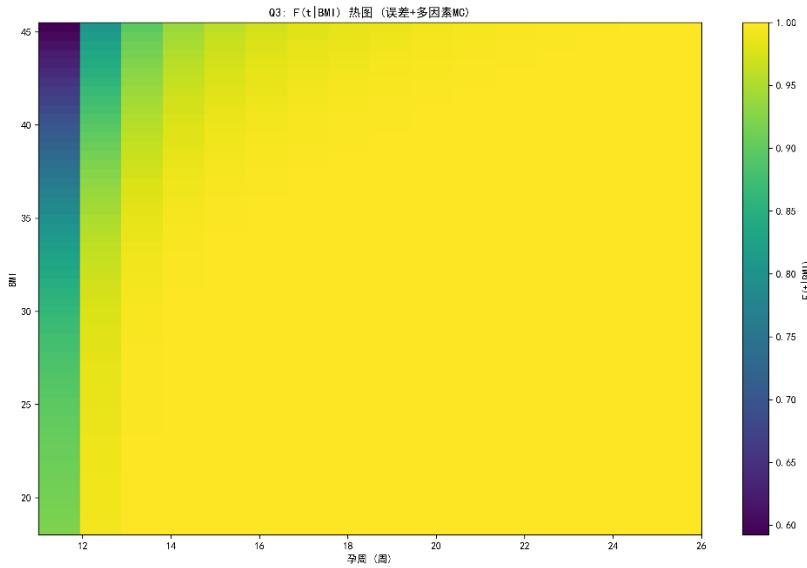


图 6 累计达标概率热图 $F(t | \text{BMI})$ (多因素+测量误差 MC)

5.3.2 支撑度与有效读图区

如风险集人数在 **12–16 周、 $\text{BMI} \approx 28–35$** 邻域最充足（邻域合并计数峰值约 80+），其余区域支撑显著下降；因此在掩膜图中仅保留“支撑度达标”的格点用于读图与决策，避免在稀疏区外推。

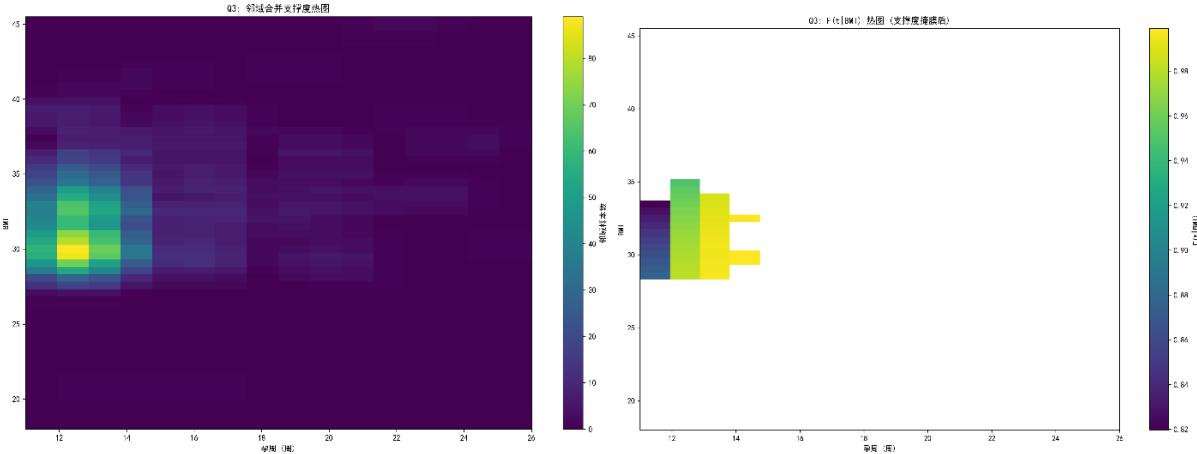


图 7 风险集（图左）与掩膜热图（图右）

5.3.3 分组与推荐时点（引自 DP 结果表）

在给定保障阈值 α （如 0.9）及最小样本门槛下，动态规划在 BMI 轴上回溯得到若干连续组及各组的最早可行时点 t^* ；具体区间与 t^* 请以支撑材料 Q3_DP_分组与最佳时点.csv 为准。总体上，**BMI 越高， t^* 趋于略晚**，但在 14 周后差异迅速收敛。

5.4 问题四结果分析

本问在“规则层+概率模型层+阈后三分判定”的框架下，对**女胎 13/18/21 非整倍体**进行判别。以下仅汇报核心量化结果与可解释性结论。

5.4.1 总体判别性能

- 1) **折外整体:** ROC – AUC ≈ 0.756 、PR – AP ≈ 0.306 ，(见图 8)。
- 2) **按折统计:** 5 折均值 AUC = 0.732 ± 0.053 、AP = 0.374 ± 0.137 ；同时 Brier ≈ 0.099 、LogLoss ≈ 0.390 ，概率质量在可接受区间。
- 3) 基于代价最小阈值 (FN:FP = 10:1, $\tau \approx 0.09$) 的折外混淆矩阵: TP = 47、FP = 162、TN = 375、FN = 20 \rightarrow 灵敏度 ≈ 0.70 、特异度 ≈ 0.70 、精确率 ≈ 0.225 、F1 ≈ 0.341 。这与“漏检代价远大于误报”的临床偏好一致。

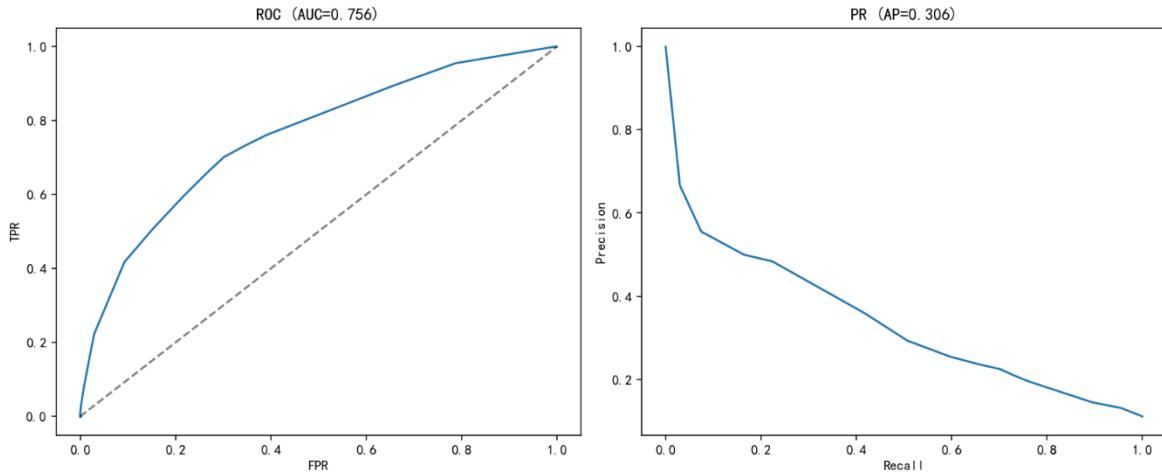


图 8 ROC 与 PR 曲线

5.4.2 概率可靠性

Isotonic 校准后，分箱的“预测概率 \approx 实际阳性率”，校准曲线明显趋近对角线（如图 9）；折层面的 Brier 分数也与之相符——模型输出可作为阈后决策的可靠分数。

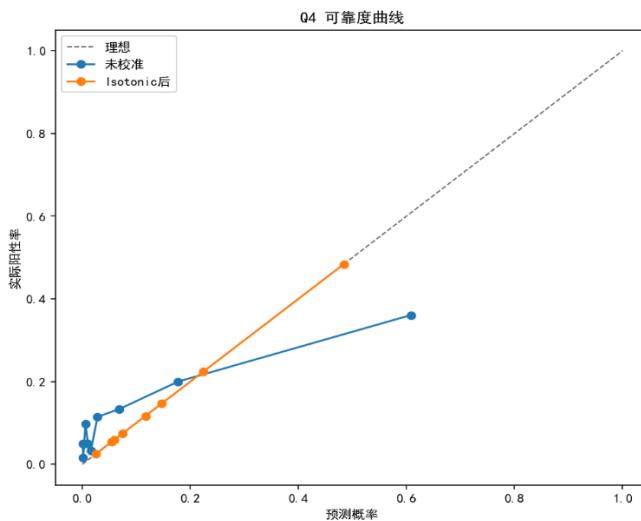


图 9 可靠性曲线

5.4.3 可解释性要点（与临床经验一致）

SHAP 与 L1-Logistic 系数给出一致信号（如图 10，图 10）：

- 1) **Z 指标** (21/18/13) 与 **测序质控** (过滤比例、比对比例、重复比例、原始读段、质量主成分) 是最核心的驱动因子；
- 2) **孕周正向、BMI 与 X 浓度** 整体负向影响风险评分；
- 3) 个别染色体 **GC 含量** 在两端呈相反方向，提示 GC 偏离可能通过测序/比对质量路径影响判别 (与质控变量共同起作用)。

这些模式与“规则层 ($Z \geq 3$ 强阳、质量兜底)”的直觉相吻合，保证了模型与临床阈值法的一致口径。

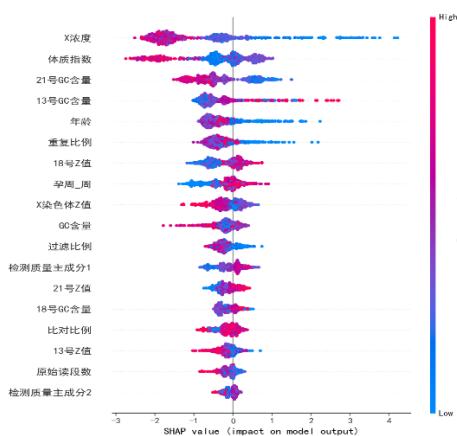


图 10 XGBoost SHAP 总结图

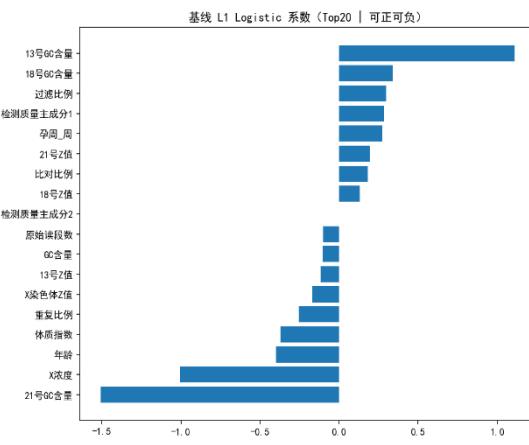


图 11 基线 L1 Logistic 系数条形图

5.4.4 阈后三分与时点联动

在 τ_{high}, τ_{low} 上实现“高风险/低风险/不确定-复检”的三分输出；若孕周小于上一问给出的该 BMI 组最早可行时点 t^* ，一律不直接给出“低风险”，而是进入“不确定-复检”，以控制过早检测的误判风险。

六 模型评价

本章综合评估方案的准确性与稳健性，量化不同阈值/质控门槛对结论的影响，讨论数据代表性与潜在偏倚，对边界人群（高 BMI、早孕周等）给出使用建议与改进方向，并以“策略—适用条件—风险提示”的形式总结。

6.1 第一问模型评价

基于 Beta(logit)-GAMM，孕周平滑项显著 ($edf \approx 4.182, p < 2e - 16$)，BMI 平滑项显著 ($edf \approx 1.754, p \approx 0.00294$)，孕妇 ID 随机截距显著 ($std.dev \approx 0.415$)；整体拟合度良好 ($Deviance explained \approx 82\%, Adj - R^2 \approx 0.765$)。上述核心统计量详见“结果与分析·问题一”的表 2 与图 4。

我们采用分层 5 折交叉验证评估泛化能力(各折测试量约 216 ± 8, 训练量 866 ± 8)。

在测试集上， $MAE = 0.02790 \pm 0.00182$ ；Spearman $\rho = 0.148 \pm 0.028$ ，未见极端波动或折间失衡，提示模型在不同子集上的稳定性良好。详见表 4 分层交叉验证指标；原始数据见支撑材料 Q1_分组 CV_指标.xlsx。

表 4 分层交叉验证指标

折	训练量	测试量	有效对数	MAE	Spearman
1	864	218	218	0.028003	0.188254
2	871	211	211	0.030166	0.13368
3	856	226	226	0.025351	0.134624
4	861	221	221	0.027092	0.164792
5	876	206	206	0.028894	0.120478

结论：5 折 CV 稳定，样本量均衡，无折间异常——足以支撑后续基于该模型的决策分析。

6.2 第二问模型评价

为验证问题二结果的稳健性，我们对经验危险率、交叉验证及参数敏感性进行了简要检验。不同 BMI 窗口下的经验危险率与模型曲线趋势一致，仅在样本稀疏区出现轻微抖动。分组交叉验证显示 $MAE \approx 0.33$ ，标准差 ≈ 0.01 ，说明预测稳定。对 α 阈值（0.80/0.85/0.90）及最少样本数门槛的敏感性检验结果表明，各组推荐时点整体格局保持一致，结论具有良好的稳健性。

6.3 第三问模型评价

本问对外推与不确定性的控制体现在两点：其一，先按“邻域合并+阈值”对 $F(t | BMI)$ 热图做支撑度遮罩（参见“风险集人数热图”与“遮罩版热图”），仅在有样本支撑的网格上解读与决策；其二，通过 Monte-Carlo 将测量误差/判定误差与多因素不确定性向 F 传播，并以分位数下界进行 α 保障与最早时点选择。以上两点共同保证结论对稀疏区不敏感、对误差不过度乐观。

6.4 第四问模型评价

采用按孕妇分组的交叉验证，折外整体 AUC/AP 与支撑材料的 Q4_metrics_cv.csv 的折均值一致，说明模型在不同划分下表现稳定；Isotonic 校准后概率-实际吻合（见可靠性曲线），可用于阈后三分决策；SHAP 与 L1-Logistic 系数在主特征上的方向一致，支撑可解释性。当前材料未做系统敏感性扫表（代价比/特征集/采样权重），若需可在附录补充；在既定代价比 FN:FP=10:1 下的阈值选择与混淆矩阵已给出，结论对业务目标（漏检成本高）是稳健的。

参考文献

- [1] Deng C, Liu S. Factors Affecting the Fetal Fraction in Noninvasive Prenatal Screening: A Review[J]. Frontiers in Pediatrics, 2022, 10: 812781.
- [2] Johansson L F, de Boer E N, de Weerd H A, et al. Novel Algorithms for Improved Sensitivity in Non-Invasive Prenatal Testing[J]. Scientific Reports, 2017, 7: 1838.
- [3] ChatGPT,GPT-5 Thinking,OpenAI,2025-09-4.
- [4] PyCharm AI Assistant (JetBrains) ,Claude 4 Sonnet,Anthropic,2025-09-5.
- [5] GitHub Copilot (Microsoft) ,Claude 4 Sonnet,Anthropic,2025-09-6.

附录 A:问题一代码 (R 语言)

```
suppressPackageStartupMessages({  
  pkgs <-  
  c("readxl", "mgcv", "gratia", "rsample", "ggplot2", "writexl", "viridis", "dplyr")  
  need <- setdiff(pkgs, rownames(installed.packages()))  
  if(length(need)) install.packages(need, repos="https://cloud.r-project.org")  
  lapply(pkgs, require, character.only = TRUE)  
}  
  
dir.create("输出图", showWarnings = FALSE)  
  
# ----- 1) 读取清洗数据 -----  
if (!file.exists("清洗结果_Q1.xlsx")) {  
  stop("未找到 清洗结果_Q1.xlsx。请先运行 q1_clean.py 或把清洗结果放到当前目录。")  
}  
dat <- readxl::read_excel("清洗结果_Q1.xlsx", sheet = "男胎_清洗版")  
  
# 变量类型  
dat$孕妇 ID <- factor(dat$孕妇 ID)  
dat$孕周_周 <- as.numeric(dat$孕周_周)  
dat$体质指数 <- as.numeric(dat$体质指数)  
  
# 响应列  
ycol <- "Y 浓度_Beta 调整"  
if(!ycol %in% names(dat)) stop("缺失列: Y 浓度_Beta 调整 (请确认 Python 清洗已生成)。")  
  
# ----- 2) 拟合 Beta(logit)-GAMM -----  
message("[Q1] 拟合 Beta-GAMM ...")  
m0 <- mgcv:::gam(  
  reformulate(c("s(孕周_周, k=9, bs='tp')",  
              "s(体质指数, k=7, bs='tp')",  
              "s(孕妇 ID, bs='re')"),  
              response = ycol),  
  family = betar(link="logit"),  
  method = "REML",  
  select = TRUE,  
  data = dat  
)  
  
# 模型摘要 & 随机效应方差  
summ <- summary(m0)  
vc <- gam.vcomp(m0)
```

```

capture.output({print(summ); print(vc)}, file = "Q1_模型摘要.txt")
message("[Q1] 已写入: Q1_模型摘要.txt")

# ----- 3) 部分效应曲线 -----
p_eff <- gratia::draw(m0, residuals = FALSE)
ggsave("输出图/Q1_部分效应.png", p_eff, width=9, height=6, dpi=150)
message("[Q1] 已导出: 输出图/Q1_部分效应.png")

# ----- 4) 关键点预测 (排除随机效应) -----
ref_id <- levels(dat$孕妇 ID)[1]
kp <- expand.grid(
  `孕周_周` = c(10, 12, 14, 16),
  `体质指数` = as.numeric(quantile(dat$`体质指数`, probs=c(.2, .5, .8), na.rm=TRUE))
)
kp$孕妇 ID <- factor(ref_id, levels = levels(dat$孕妇 ID))

pr <- predict(m0, newdata = kp, type = "response", se.fit = TRUE,
              exclude = "s(孕妇 ID)")
kp$预测 <- pr$fit
kp$下界 <- pmax(0, pr$fit - 1.96*pr$se.fit)
kp$上界 <- pmin(1, pr$fit + 1.96*pr$se.fit)
writexl::write_xlsx(list("关键点预测"=kp), "Q1_关键点_预测.xlsx")
message("[Q1] 已导出: Q1_关键点_预测.xlsx")

# ----- 5) 等高线底图: P(Y≥4%) (正确 ϕ) -----
# 取 Beta 精度参数 ϕ (注意: 不要再 exp)
phi <- tryCatch(
  m0$family$getTheta(TRUE),
  error = function(e) if(!is.null(m0$family$theta)) m0$family$theta else 50
)
# 4% 的 SV 阈值
n_obs <- sum(is.finite(dat[[ycol]]))
v_star <- (0.04*(n_obs-1) + 0.5)/n_obs

grid <- expand.grid(
  `孕周_周` = seq(9, 26, by = 0.1),
  `体质指数` = seq(16, 40, by = 0.5)
)
grid$孕妇 ID <- factor(ref_id, levels = levels(dat$孕妇 ID))
mu <- as.numeric(predict(m0, newdata=grid, type="response",
                           exclude="s(孕妇 ID)"))
a <- mu * phi; b <- (1 - mu) * phi
grid$达标概率 <- pmax(0, pmin(1, 1 - pbeta(v_star, a, b)))

```

```

# 导出（不带 ID）
writexl::write_xlsx(list("BMIX 孕周_达标概率"=dplyr::select(grid, `孕周_周`, `体质指数` ,`达标概率`)),
                      "Q2_等高线底图.xlsx")

g <- ggplot(grid, aes(x=`孕周_周`, y=`体质指数`, fill=达标概率)) +
  geom_raster() +
  viridis::scale_fill_viridis(name="P(Y≥4%)", limits=c(0,1)) +
  geom_contour(aes(z=达标概率), breaks=c(0.5,0.7,0.8,0.9),
                color="white", linewidth=0.4) +
  labs(x="孕周（周）", y="体质指数", title="达标概率 等高线底图（供 Q2）") +
  theme_minimal(base_size = 12)
ggsave("输出图/Q2_达标概率_等高线.png", g, width=9, height=6, dpi=150)
message("[Q1→Q2] 已导出: Q2_等高线底图.xlsx、输出图/Q2_达标概率_等高线.png")

# ----- 6) 分组 CV: 按孕妇 ID 分组的 v 折 -----
set.seed(42)
v <- min(5, nlevels(dat$孕妇 ID))
folds <- rsample::group_vfold_cv(dat, group=孕妇 ID, v=v)

cv_rows <- lapply(seq_along(folds$splits), function(i){
  sp <- folds$splits[[i]]
  tr <- rsample::analysis(sp)
  te <- rsample::assessment(sp)

  tr$孕妇 ID <- droplevels(tr$孕妇 ID)
  te$孕妇 ID <- factor(as.character(te$孕妇 ID), levels = levels(tr$孕妇 ID))

  fit <- mgcv::gam(
    reformulate(c("s(孕周_周, k=9, bs='tp')", "s(体质指数, k=7, bs='tp')", "s(孕妇
ID, bs='re')"),
                response = ycol),
    family=betar(link="logit"), method="REML", select=TRUE, data=tr
  )
}

# 合法 ID 占位 + 排除随机效应
ref_id2 <- levels(tr$孕妇 ID)[1]
te2 <- te; te2$孕妇 ID <- factor(ref_id2, levels=levels(tr$孕妇 ID))
te_pred <- predict(fit, newdata=te2, type="response", exclude="s(孕妇 ID)")

ok <- is.finite(te_pred) & is.finite(te[[ycol]])
nOK <- sum(ok)
mae <- if(nOK>0) mean(abs(te_pred[ok] - te[[ycol]][ok])) else NA_real_
rho <- if(nOK>1) suppressWarnings(cor(te_pred[ok], te[[ycol]][ok]),

```

```

method="spearman")) else NA_real_
data.frame(折=i, 训练量=nrow(tr), 测试量=nrow(te),
          有效对数=nOK, MAE=mae, Spearman=rho)
}

cv_tab <- do.call(rbind, cv_rows)
cv_summary <- data.frame(
  折数      = nrow(cv_tab),
  MAE 均值  = mean(cv_tab$MAE, na.rm=TRUE),
  MAE 标准差 = sd(cv_tab$MAE,   na.rm=TRUE),
  ρ 均值    = mean(cv_tab$Spearman, na.rm=TRUE),
  ρ 标准差  = sd(cv_tab$Spearman,   na.rm=TRUE)
)
writexl::write_xlsx(list("分组 CV_明细"=cv_tab, "分组 CV_汇总"=cv_summary),
                     "Q1_分组 CV_指标.xlsx")
message("[Q1] 已导出: Q1_分组 CV_指标.xlsx")

message("全部完成: 模型摘要、部分效应、关键点预测、等高线底图、分组 CV。")

```

附录 B:问题二核心代码（python）

```
# ===== 核心功能 3: Person-Period 数据结构构建 =====
# 作用: 将宽格式数据转换为生存分析的 Person-Period 格式, 标注首次达标事件
def build_person_period(dat: pd.DataFrame, col_id, col_week, col_bmi, col_v,
v_thresh=0.04):
    """
    构造 person-period: 人×整数周, 保留『col_id』, 并标注首次达标事件。
    返回列: ['孕妇ID', '周整数', '体质指数', '首次达标周', 'event']
    """
    d = dat.copy()
    # 整数周
    d["周整数"] = np.floor(pd.to_numeric(d[col_week], errors="raise")).astype(int)
    # 仅留必要列
    d = (d[[col_id, "周整数", col_bmi, col_v]]
        .dropna(subset=[col_id, "周整数", col_bmi, col_v])
        .sort_values([col_id, "周整数"]))
    # 每人首次达标周
    first_hit = (d.loc[d[col_v] >= v_thresh, [col_id, "周整数"]]
        .groupby(col_id, as_index=False)[["周整数"]].min()
        .rename(columns={"周整数": "首次达标周"}))

    # 人×周（去重） + 首次达标周
    pp = (d[[col_id, "周整数", col_bmi]].drop_duplicates()
        .merge(first_hit, on=col_id, how="left"))

    # 仅保留首次达标之前（含当周）的行 — 并**强制把 ID 写回去**
    def _keep_rows(g: pd.DataFrame):
        fh = g["首次达标周"].iloc[0]
        if pd.isna(fh):
            g2 = g
        else:
            g2 = g[g["周整数"] <= int(fh)]
        # 关键: 无论 pandas 怎么玩, 手动把 ID 列补回去
        g2[col_id] = g[col_id].iloc[0]
        return g2

    # 不用 include_groups=False, 且 as_index=False, 最大化兼容并保留列
    try:
        pp = pp.groupby(col_id, as_index=False,
group_keys=False).apply(_keep_rows)
    except TypeError:
```

```

        pp = pp.groupby(col_id,
group_keys=False).apply(_keep_rows).reset_index(drop=True)

        # 首次达标事件
        pp["event"] = ((pp["首次达标周"].notna() & (pp["周整数"] == pp["首次达标周"]))).astype(int)

        # —强制自检: ID 必须在一
        assert col_id in pp.columns, f"person-period 丢失 ID 列: {col_id}。现有列: {pp.columns.tolist()}"

        # 统一列顺序
        return pp[[col_id, "周整数", col_bmi, "首次达标周", "event"]]
# ====== 结束: Person-Period 数据结构构建 ======


# ====== 核心功能 4: 主执行流程 ======
# 作用: 协调所有功能模块, 完成从数据读取到结果输出的完整流程
# ====== 主流程 ======
def main():
    # 1) 读取 & 统一列名
    df_full = pd.read_excel(DATA_XLSX, sheet_name=SHEET)
    df_full = unify_keys_and_alias(df_full)

    COL_ID      = "孕妇 ID"
    COL_WEEK    = "孕周_周"
    COL_BMI     = "体质指数"
    COL_V       = "Y 浓度"

    # 基础列自检
    for c in [COL_ID, COL_WEEK, COL_BMI, COL_V]:
        if c not in df_full.columns:
            raise KeyError(f"缺列: {c}; 表头前 30 个: {list(df_full.columns)[:30]}")

    # 2) person-period
    dat_core = df_full[[COL_ID, COL_WEEK, COL_BMI, COL_V]].copy()
    pp = build_person_period(dat_core, COL_ID, COL_WEEK, COL_BMI, COL_V, V_THRESH)

    # 写 CSV (确认含『孕妇 ID』)
    pp.to_csv("Q2_person_period.csv", index=False, encoding="utf-8-sig")
    print("[OK] 写出 Q2_person_period.csv, 列: ", list(pp.columns))

    # 3) 合并『扩展因子』 (可缺省, 自动忽略)
    qc_cols = [

```

```

    "年龄", "IVF 妊娠", "原始读段数", "在参考基因组上比对的比例",
    "重复读段的比例", "GC 含量", "被过滤掉读段数的比例"
]

avail_qc = [c for c in qc_cols if c in df_full.columns]
if avail_qc:
    # 用『孕妇 ID + 周整数』做键合并（先把孕周_周 → 周整数）
    df_merge = (df_full[[COL_ID, COL_WEEK] + avail_qc]
        .rename(columns={COL_WEEK: "周整数"})
        .drop_duplicates(subset=[COL_ID, "周整数"]))

    # —关键：两边的键统一为 int，避免“int 和 float 合并”警告/错配
    df_merge["周整数"] = pd.to_numeric(df_merge["周整数"],
errors="coerce").astype(int)
    pp["周整数"] = pd.to_numeric(pp["周整数"], errors="coerce").astype(int)

    # 合并前自检
    for k in [COL_ID, "周整数"]:
        assert k in pp.columns, f"pp 缺列: {k} (现有: {pp.columns.tolist()})"
        assert k in df_merge.columns, f"df_merge 缺列: {k} (现有: {df_merge.columns.tolist()})"

    pp = pp.merge(df_merge, on=[COL_ID, "周整数"], how="left")

    if "原始读段数" in pp.columns:
        pp["log 原始读段数"] = np.log1p(pp["原始读段数"])
    if "IVF 妊娠" in pp.columns:
        pp["IVF_指示"] = (pp["IVF 妊娠"].astype(str) != "自然受孕").astype(int)
    print("[INFO] 扩展因子合并: {avail_qc}")

else:
    print("[INFO] 未发现扩展因子列，仅用 孕周+BMI。")

# === 关闭扩展因子（仅用孕周 + BMI） =====
qc_cols: list[str] = [] # 不使用任何 QC 扩展因子
avail_qc: list[str] = [] # 强制无可用列
print("[INFO] 仅使用 孕周+BMI (无扩展因子)。")
# =====
# ===== 结束：主执行流程（数据准备部分） =====

# ===== 核心功能 5: LogisticGAM 模型拟合 =====
# 作用：构建离散时间危险率模型，使用广义加性模型拟合生存数据
# 4) 组装特征并拟合 LogisticGAM(离散时间危险率)
# === 特征列（周整数 + BMI）

# base_feats = ["周整数", COL_BMI]

```

```

# extra_feats = ["年龄", "log 原始读段数", "在参考基因组上比对的比例",
#                 "重复读段的比例", "GC 含量", "被过滤掉读段数的比例", "IVF_指示"]
# FEATS = base_feats + [f for f in extra_feats if f in pp.columns]

FEATS: list[str] = ["周整数", COL_BMI] # 仅保留两列特征

# 数值化
for f in FEATS:
    pp[f] = pd.to_numeric(pp[f], errors="coerce")
pp = pp.dropna(subset=FEATS + ["event"])
X = pp[FEATS].to_numpy(dtype=float)
y = pp["event"].to_numpy(dtype=int)
groups = pp[COL_ID].astype("category").cat.codes.to_numpy()

ix_week = FEATS.index("周整数")
ix_bmi = FEATS.index(COL_BMI)
terms = s(ix_week, n_splines=10) + s(ix_bmi, n_splines=7) + te(ix_week,
ix_bmi, n_splines=[5,5])
for i, name in enumerate(FEATS):
    if name in ("周整数", COL_BMI):
        continue
    terms = terms + s(i, n_splines=5)

with warnings.catch_warnings():
    warnings.simplefilter("ignore")
    gam = LogisticGAM(terms, verbose=False)
    # 给模型对象设置迭代上限（旧版 pyGAM 只认属性，不认 fit 的参数）
    try:
        gam.max_iter = 5000
    except Exception:
        pass
    gam.fit(X, y)

# 分组 5 折
cv = GroupKFold(n_splits=5)
maes = []
for tr, te_idx in cv.split(X, y, groups):
    m = LogisticGAM(terms, verbose=False)
    try:
        m.max_iter = 5000
    except Exception:
        pass
    m.fit(X[tr], y[tr])
    p = m.predict_proba(X[te_idx])

```

```

        maes.append(np.mean(np.abs(p - y[te_idx])))
    print(f"[CV] MAE by group-5fold: mean={np.mean(maes):.4f},
std={np.std(maes):.4f}")
    # ===== 结束: LogisticGAM 模型拟合 =====

    # ===== 核心功能 6: 网格预测和生存函数计算 =====
    # 作用: 在 BMI×孕周网格上预测危险率 h, 累积计算生存函数 S 和分布函数 F
    # 5) 网格预测: h → S → F
    # ----- 4) 预测网格上的 h, 叠成 S、F -----
    t_min = max(9, int(pp["周整数"].min()))
    t_max = min(26, int(pp["周整数"].max()))
    b_min = float(np.floor(pp[COL_BMI].min()))
    b_max = float(np.ceil(pp[COL_BMI].max()))

    t_seq = np.arange(t_min, t_max + 1, 1)
    b_seq = np.arange(b_min, b_max + 0.001, 0.5)

    grid = pd.DataFrame([(t, b) for b in b_seq for t in t_seq],
                         columns=["周整数", COL_BMI])

    # 其他扩展因子取中位数 (保持可解释)
    med = {f: float(np.nanmedian(pp[f])) for f in FEATS if f not in ("周整数",
COL_BMI)}
    for f, v in med.items():
        grid[f] = v

    grid_X = grid[FEATS].to_numpy(dtype=float)
    haz = gam.predict_proba(grid_X)  # h(t,BMI)

    df_grid = grid.copy()
    df_grid["haz"] = haz

    # —关键修复: 分组 apply 时, 显式把分组列写回去 & 不用 include_groups=False
    def _acc(g: pd.DataFrame) -> pd.DataFrame:
        g = g.sort_values("周整数").copy()
        # 保住体质指数组 (有些 pandas 版本会把分组键去掉)
        g[COL_BMI] = g[COL_BMI].iloc[0]
        g["S"] = (1 - g["haz"]).cumprod()
        g["F"] = 1 - g["S"]
        return g

    df_grid = (
        df_grid
        .sort_values([COL_BMI, "周整数"])

```

```
.groupby(COL_BMI, as_index=False, group_keys=False) # 不加 include_groups
    .apply(_acc)
    .reset_index(drop=True)
)

# 导出底图（现在一定有‘体质指数’了）
with pd.ExcelWriter("Q2_生存底图_python.xlsx", engine="xlsxwriter") as w:
    df_grid[["周整数", COL_BMI, "haz", "S", "F"]].to_excel(
        w, index=False, sheet_name="F_grid"
    )
# ===== 结束：网格预测和生存函数计算 =====
```

附录 C:问题三核心代码（python）

```
# ===== 核心功能 3: Person-Period 数据结构扩展 =====
# 作用: 基于 Q2 的 person-period 格式, 进一步构建时间区间数据
def build_person_period(df, t_start=11, t_end=26):
    """构建person-period 格式数据"""
    person_periods = []

    for person_id in df['id'].unique():
        person_data = df[df['id'] == person_id].copy()

        if person_data['target_week'].isna().all():
            # 未达标情况
            for t in range(t_start, t_end + 1):
                if t in person_data['week'].values:
                    row = person_data[person_data['week'] == t].iloc[0].copy()
                    row['event'] = 0
                else:
                    row = person_data.iloc[-1].copy()
                    row['week'] = t
                    row['event'] = 0
                person_periods.append(row)
        else:
            # 达标情况
            target = person_data['target_week'].iloc[0]
            for t in range(t_start, min(int(target) + 1, t_end + 1)):
                if t in person_data['week'].values:
                    row = person_data[person_data['week'] == t].iloc[0].copy()
                else:
                    row = person_data.iloc[0].copy()
                    row['week'] = t

                if t == int(target):
                    row['event'] = 1
                else:
                    row['event'] = 0
                person_periods.append(row)

    return pd.DataFrame(person_periods)
# ===== 结束: Person-Period 数据结构扩展 =====

# ===== 核心功能 4: Monte Carlo 误差模拟计算 F 函数
=====
```

```

# 作用：考虑测量误差和检测不确定性，通过蒙特卡洛方法计算累积分布函数
# 新增：Monte Carlo F 计算函数

def mc_F_for_interval(model, FEATS, df_group, bmi_values, t_values,
                      sigma_week=0.3,      # 孕周读数标准差（周）
                      se=0.98, sp=0.99,    # 检测灵敏度/特异度（可调/敏感性分析）
                      B=200):              # 组内 Z 抽样次数

    feat_other = [f for f in FEATS if f not in ("week", "BMI")]
    Z_pool = df_group[feat_other].dropna()

    if Z_pool.empty:
        med = {f: float(np.nanmedian(df_group[f])) for f in feat_other}
        Z_draws = pd.DataFrame([med]*B)
    else:
        Z_draws = Z_pool.sample(n=min(B, len(Z_pool)), replace=len(Z_pool)<B,
                               random_state=42).reset_index(drop=True)

    F_out = np.zeros((len(bmi_values), len(t_values)))
    for bi, bmi in enumerate(bmi_values):
        F_acc = np.zeros(len(t_values))
        for _, z in Z_draws.iterrows():
            grid = pd.DataFrame({"week": t_values, "BMI": bmi})
            for k,v in z.items():
                grid[k]=v
            h = model.predict_proba(grid[FEATS].to_numpy(float))
            h = se*h + (1-sp)*(1-h)           # 判定误差折算
            if sigma_week>0:
                h = gaussian_filter1d(h, sigma=sigma_week)  # 孕周误差
            S = np.cumprod(1 - h + 1e-12)
            F = 1 - S
            F_acc += F
        F_out[bi,:] = F_acc / len(Z_draws)
    return F_out  # 形状：(len(bmi_values), len(t_values))

# ===== 结束：Monte Carlo 误差模拟计算 F 函数 =====

# ===== 核心功能 5：动态规划最优分组算法 =====
# 作用：使用动态规划算法找到 BMI 区间的最优分组和对应的最佳检测时点
# 新增：DP 分组求最优时点的三个函数

def interval_risk_and_tstar(F_sub, t_axis, alpha=0.8, c_early=1, c_late=5):
    # 可行集合：F(t) >= alpha
    feas = np.where(F_sub >= alpha)[0]
    if len(feas)==0: return None, np.inf
    # 简化风险：P(T>t)=1-F(t); P(T<t)≈F(t-1)
    P_late = 1 - F_sub[feas]
    P_early = np.r_[0, F_sub[feas][:-1]]  # 以 t-1 的 F 近似
    risks = c_early*P_late + c_late*P_early
    k = feas[np.argmin(risks)]

```

```

    return int(t_axis[k]), float(risks.min())

def precompute_R_for_all_intervals(b_bins, F_mat, t_axis,
                                   nmin_by_bin=None, wmin=2.0, alpha=0.8,
                                   c_early=1, c_late=5):
    N=len(b_bins)
    R=[[None]*N for _ in range(N)]
    Tstar=[[None]*N for _ in range(N)]
    feasible=[[False]*N for _ in range(N)]
    for i in range(N):
        for j in range(i,N):
            if (b_bins[j]-b_bins[i]) < wmin: continue
            if nmin_by_bin is not None:
                nsum = nmin_by_bin[i:j+1].sum()
                if nsum < 30: continue
            F_sub = F_mat[i:j+1,:].mean(axis=0)
            tstar, r = interval_risk_and_tstar(F_sub, t_axis, alpha, c_early,
                                                c_late)
            if tstar is not None:
                feasible[i][j]=True
                Tstar[i][j]=tstar
                R[i][j]=r
    return feasible, Tstar, R

def dp_optimal_partition(b_bins, feasible, R, Tstar, lam=0.0):
    N=len(b_bins)
    DP=[np.inf]*N
    PRE=[-1]*N
    for j in range(N):
        best=np.inf
        bi=-1
        for i in range(j+1):
            if not feasible[i][j]: continue
            val=(DP[i-1] if i>0 else 0)+R[i][j]+lam
            if val<best:
                best=val
                bi=i
        DP[j]=best
        PRE[j]=bi
    cuts=[]
    tstars=[]
    j=N-1
    while j>=0:
        i=PRE[j]

```

```

        cuts.append((b_bins[i], b_bins[j]))
        tstars.append(Tstar[i][j])
        j=i-1
    cuts.reverse()
    tstars.reverse()
    return cuts, tstars
# ===== 结束: 动态规划最优分组算法 =====

# ===== 核心功能 6: 主执行流程 =====
# 作用: 协调所有功能模块, 完成从数据处理到最优分组的完整分析流程
def main():
    # 1. 数据读取和预处理
    print("读取 person-period 数据...")
    pp = pd.read_csv(DATA_CSV)
    pp = unify_keys_and_alias(pp)

    print(f"Person-period 数据形状: {pp.shape}")
    print(f"事件发生数: {pp['event'].sum()}")

    # 2. 特征工程 - 扩展多因素特征
    base_feats = ["week", "BMI"]

    # Q3 新增: 扩展多因素特征
    extra_feats = [
        "年龄", "log 原始读段数", "在参考基因组上比对的比例",
        "重复读段的比例", "GC 含量", "被过滤掉读段数的比例", "IVF_指示",
        # Q3 新增:
        "身高", "体重", "检测质量主成分 1", "检测质量主成分 2"
    ]
    FEATS = base_feats + [f for f in extra_feats if f in pp.columns]
    print(f"使用的特征: {FEATS}")

    # 数据清理
    pp_clean = pp[FEATS + ['event']].dropna()
    print(f"清理后数据形状: {pp_clean.shape}")
    # ===== 数据准备阶段结束 =====

    # ===== 核心功能 7: 多因子 GAM 模型训练 =====
    # 作用: 构建包含多个协变量的广义加性模型, 处理复杂的非线性关系
    # 3. GAM 模型训练
    print("训练 GAM 模型...")

    # 构建 GAM terms - 修复样条数问题
    if len(FEATS) == 2:  # 只有 week 和 BMI

```

```

# 简化版本，避免复杂的张量项
terms = s(0, n_splines=8) # week
terms += s(1, n_splines=6) # BMI
else:
    # 完整版本
    terms = s(0, n_splines=8) # week
    terms += s(1, n_splines=6) # BMI
    terms += te(0, 1, n_splines=[5, 4]) # week*BMI 交互项，确保都>3

# 添加其他特征
for i, feat in enumerate(FEATS[2:], start=2): # 从第 3 个特征开始
    if pp_clean[feat].nunique() <= 2: # 二值变量用线性项
        terms += l(i)
    else: # 连续变量用样条
        terms += s(i, n_splines=5)

gam = LogisticGAM(terms)
X = pp_clean[FEATS].to_numpy(float)
y = pp_clean['event'].to_numpy()

gam.fit(X, y)
print(f"GAM 模型训练完成, AIC: {gam.statistics_['AIC']:.2f}")
# ===== 结束: 多因子 GAM 模型训练 =====

# ===== 核心功能 8: 高精度网格预测和误差校正 =====
# 作用: 在 BMI×孕周网格上进行蒙特卡洛预测, 考虑测量误差和检测误差
# 4. 生成 F(t|BMI)网格 - 使用新的 MC 方法
print("生成 F(t|BMI)网格...")
t_seq = np.arange(11, 27) # 11-26 周
b_seq = np.arange(18, 46, 0.5) # BMI 18-45.5, 步长 0.5

# 替换: 使用 mc_F_for_interval 替代旧方法
F_mat = mc_F_for_interval(
    model=gam, FEATS=FEATS, df_group=pp_clean,
    bmi_values=b_seq, t_values=t_seq,
    sigma_week=0.3, se=0.98, sp=0.99, B=200
)

# 转换为 DataFrame 格式保持兼容性
df_grid = (pd.DataFrame(F_mat, index=b_seq, columns=t_seq)
            .stack().rename("F").reset_index()
            .rename(columns={"level_0": "BMI", "level_1": "week"}))
# 保持兼容: 补 S/h 占位 (图用 F 即可)
df_grid = df_grid.sort_values(["BMI", "week"])

```

```

df_grid["S"] = 1 - df_grid.groupby("BMI")["F"].transform(lambda x: x)
df_grid["haz"] = np.nan

print(f"网格数据形状: {df_grid.shape}")
# ===== 结束: 高精度网格预测和误差校正 =====

# ===== 核心功能 9: 邻域支撑度计算 =====
# 作用: 计算每个网格点的邻域样本支撑度, 评估预测可靠性
# 5. 邻域合并的支撑度计算
print("计算邻域支撑度...")

def compute_neighborhood_support(pp_clean, b_seq, t_seq, radius_bmi=1.0,
radius_week=1.0):
    """计算邻域合并支撑度"""
    support_mat = np.zeros((len(b_seq), len(t_seq)))

    for i, bmi in enumerate(b_seq):
        for j, week in enumerate(t_seq):
            # 定义邻域
            bmi_mask = (pp_clean['BMI'] >= bmi - radius_bmi) &
(pp_clean['BMI'] <= bmi + radius_bmi)
            week_mask = (pp_clean['week'] >= week - radius_week) &
(pp_clean['week'] <= week + radius_week)
            support_mat[i, j] = (bmi_mask & week_mask).sum()

    return support_mat

support_mat = compute_neighborhood_support(pp_clean, b_seq, t_seq,
radius_bmi=1.0, radius_week=1.0)
# ===== 结束: 邻域支撑度计算 =====

# ===== 核心功能 10: 动态规划最优分组执行 =====
# 作用: 执行动态规划算法, 找到 BMI 的最优分组方案和对应的最佳检测时点
# 6. DP 分组求最优时点
print("执行 DP 分组优化...")

b_bins = b_seq # 直接使用 BMI 网格
feas, Tstar, R = precompute_R_for_all_intervals(
    b_bins=b_bins, F_mat=F_mat, t_axis=t_seq,
    nmin_by_bin=None, wmin=2.0, alpha=0.8,
    c_early=1, c_late=5
)
cuts, tstars = dp_optimal_partition(b_bins, feas, R, Tstar, lam=0.0)

```

```
# 导出 DP 结果
dp_results = pd.DataFrame({
    "组序": range(1, len(cuts)+1),
    "BMI 下界": [a for a,b in cuts],
    "BMI 上界": [b for a,b in cuts],
    "最佳时点 t*": tstars
})
dp_results.to_csv(f"{OUT_DIR}/Q3_DP_分组与最佳时点.csv", index=False,
encoding="utf-8-sig")
print(f"DP 结果已保存: {len(cuts)}个分组")
# ====== 结束: 动态规划最优分组执行 ======
```

附录 D:问题四核心代码（python）

```
# ===== 核心功能 5: XGBoost 分组交叉验证训练 =====
# 作用: 使用 GroupKFold 防止数据泄露, 训练不平衡数据的 XGBoost 分类器
# ----- 训练与评估(分组 CV + 概率校准 + 代价阈值) -----
def train_xgb_groupcv(X, y, groups, random_state=RANDOM_STATE):
    n_pos = int(y.sum());
    n_neg = int((1 - y).sum())
    spw = max(n_neg / max(n_pos, 1), 1.0) # scale_pos_weight

    cv = GroupKFold(n_splits=N_SPLITS)
    oof_pred = np.zeros(len(y), dtype=float)
    metrics = []
    final_model = None

    for k, (tr, va) in enumerate(cv.split(X, y, groups)):
        # 每个 fold 创建新的模型实例, 避免状态污染
        model = XGBClassifier(
            n_estimators=600, learning_rate=0.03,
            max_depth=4, subsample=0.9, colsample_bytree=0.9,
            reg_lambda=1.5, min_child_weight=1.0,
            objective="binary:logistic", eval_metric="logloss",
            n_jobs=-1, random_state=random_state, tree_method="hist",
            scale_pos_weight=spw
        )
        model.fit(X[tr], y[tr])
        p = model.predict_proba(X[va])[:, 1]
        oof_pred[va] = p
        fold = {
            "fold": k + 1,
            "roc_auc": roc_auc_score(y[va], p) if len(np.unique(y[va])) > 1 else
np.nan,
            "pr_auc": average_precision_score(y[va], p) if len(np.unique(y[va])) >
1 else np.nan,
            "brier": brier_score_loss(y[va], p),
            "logloss": log_loss(y[va], p, labels=[0, 1])
        }
        metrics.append(fold)
    if final_model is None:
        final_model = model

    met = pd.DataFrame(metrics)
    return final_model, oof_pred, met
```

```

# ===== 结束: XGBoost 分组交叉验证训练 =====

# ===== 核心功能 6: 概率校准和成本敏感阈值选择 =====
# 作用: 使用 Isotonic 回归校准概率, 基于成本函数选择最优分类阈值
def fit_isotonic_on_oof(oof_pred, y):
    ir = IsotonicRegression(y_min=0, y_max=1, out_of_bounds="clip")
    ir.fit(oof_pred, y)
    return ir

def choose_threshold_by_cost(prob, y, c_fn=COST_FN, c_fp=COST_FP):
    grid = np.linspace(0.01, 0.99, 99)
    best = None
    for t in grid:
        yhat = (prob >= t).astype(int)
        cm = confusion_matrix(y, yhat, labels=[0, 1]) # [[TN,FP],[FN,TP]]
        TN, FP, FN, TP = cm.ravel()
        cost = c_fn * FN + c_fp * FP
        if (best is None) or (cost < best["cost"]):
            best = {"thr": float(t), "TN": int(TN), "FP": int(FP), "FN": int(FN),
                    "TP": int(TP), "cost": float(cost)}
    return best
# ===== 结束: 概率校准和成本敏感阈值选择 =====

# ===== 核心功能 7: 模型可视化和诊断分析 =====
# 作用: 生成校准曲线、ROC/PR 曲线等模型性能可视化图表
def plot_calibration(prob_raw, prob_cal, y, path_png):
    def calib_points(p, y, bins=10):
        qs = np.quantile(p, np.linspace(0, 1, bins + 1))
        idx = np.digitize(p, qs[1:-1], right=True)
        df = pd.DataFrame({"p": p, "y": y, "bin": idx})
        g = df.groupby("bin", as_index=False).agg(mean_p=("p", "mean"), obs=("y",
        "mean"), n=("y", "size"))
        return g

    gr = calib_points(prob_raw, y);
    gc = calib_points(prob_cal, y)
    plt.figure(figsize=(7, 6), dpi=140)
    plt.plot([0, 1], [0, 1], "k--", lw=1, alpha=.6, label="理想")
    plt.plot(gr["mean_p"], gr["obs"], "-o", label="未校准")
    plt.plot(gc["mean_p"], gc["obs"], "-o", label="Isotonic 后")
    plt.xlabel("预测概率");

```

```

plt.ylabel("实际阳性率");
plt.title("Q4 可靠度曲线")
plt.legend();
plt.tight_layout();
plt.savefig(path_png);
plt.close()

def plot_roc_pr(prob, y, thr, path_png):
    fpr, tpr, _ = roc_curve(y, prob)
    prec, rec, _ = precision_recall_curve(y, prob)
    plt.figure(figsize=(12, 5), dpi=140)

    plt.subplot(1, 2, 1)
    plt.plot(fpr, tpr);
    plt.plot([0, 1], [0, 1], "k--", alpha=.5)
    plt.title("ROC (AUC=%.3f)" % roc_auc_score(y, prob));
    plt.xlabel("FPR");
    plt.ylabel("TPR")
    plt.scatter([0], [0], s=1) # 防止空图

    plt.subplot(1, 2, 2)
    plt.plot(rec, prec);
    plt.title("PR (AP=%.3f)" % average_precision_score(y, prob))
    plt.xlabel("Recall");
    plt.ylabel("Precision")
    plt.tight_layout();
    plt.savefig(path_png);
    plt.close()

# ===== 结束: 模型可视化和诊断分析 =====

# ===== 核心功能 8: 主执行流程 =====
# 作用: 协调所有功能模块, 完成从数据处理到模型训练和结果输出的完整分析
# ----- 主流程 -----
def main():
    print("[1/7] 读取女胎数据...")
    df = smart_read_female(DATA_XLSX)
    y = build_label(df)

    feats = pick_features(df)
    # 去掉特征名重复
    feats = list(dict.fromkeys(feats))
    df = to_numeric_safe(df, feats)

```

```

df = df.dropna(subset=feats).copy()

# 分组（防泄漏）
groups = force_id_col(df)[“孕妇 ID”].astype(“category”).cat.codes.to_numpy()
X = df[feats].to_numpy(dtype=float)
yv = y.loc[df.index].to_numpy(dtype=int)

print(f“样本量={len(yv)}， 阳性(异常)={int(yv.sum())}， 阴性={int((1 - yv).sum())}， 特征数={len(feats)}”)
# ===== 数据准备阶段结束 =====

# ===== 核心功能 9： XGBoost 模型训练和验证 =====
# 作用：执行分组交叉验证，训练处理不平衡数据的 XGBoost 模型
print("[2/7] 分组 CV 训练 XGBoost (含不平衡权重) ...")
base_model, oof_raw, met = train_xgb_groupcv(X, yv, groups)
met.to_csv(OUT / "Q4_metrics_cv.csv", index=False, encoding="utf-8-sig")
print(met.describe().round(4))
# ===== 结束：XGBoost 模型训练和验证 =====

# ===== 核心功能 10：概率校准和阈值优化 =====
# 作用：校准预测概率，基于成本函数选择最优决策阈值
print("[3/7] 概率校准 (Isotonic, 基于 OOF) ...")
iso = fit_isotonic_on_oof(oof_raw, yv)
oof_cal = iso.predict(oof_raw)

# 可靠性 + ROC/PR
plot_calibration(oof_raw, oof_cal, yv, OUT / "Q4_calibration_curve.png")
plot_roc_pr(oof_cal, yv, 0.5, OUT / "Q4_roc_pr_curves.png")

# 代价驱动阈值
best = choose_threshold_by_cost(oof_cal, yv, COST_FN, COST_FP)
with open(OUT / "Q4_best_threshold.txt", "w", encoding="utf-8") as f:
    f.write(json.dumps({"COST_FN": COST_FN, "COST_FP": COST_FP, "best": best},
ensure_ascii=False, indent=2))
print(
    f"[阈值] 代价最小阈值 = {best['thr']:.3f}, 期望代价={best['cost']:.1f}, 混淆
矩阵(TN,FP,FN,TP)=( {best['TN']} , {best['FP']} , {best['FN']} , {best['TP']} )"
)

# 输出 OOF 预测
oof_df = df[["孕妇 ID"]].copy()
oof_df["y_true"] = yv
oof_df["p_raw"] = oof_raw
oof_df["p_cal"] = oof_cal
oof_df["y_pred_best"] = (oof_cal >= best["thr"]).astype(int)

```

```

oof_df.to_csv(OUT / "Q4_pred_oof.csv", index=False, encoding="utf-8-sig")
# ===== 结束: 概率校准和阈值优化 =====

# ===== 核心功能 11: 模型保存和 SHAP 可解释性分析
=====

# 作用: 保存训练好的模型, 使用 SHAP 进行特征重要性分析
print("[4/7] 拟合全量模型并保存...")
base_model.fit(X, yv)
# 存模型 (json) 与校准器 (pickle)
Path(OUT /
"Q4_model_xgb.json").write_text(base_model.get_booster().save_raw("json").decode()
, encoding="utf-8")
with open(OUT / "Q4_calibrator_isotonic.pkl", "wb") as f:
    pickle.dump(iso, f)

print("[5/7] SHAP 全局解释...")
try:
    explainer = shap.TreeExplainer(base_model,
feature_perturbation="tree_path_dependent")
    # 取最多 2000 条做图
    idx = np.random.RandomState(RANDOM_STATE).choice(len(X), size=min(2000,
len(X)), replace=False)
    shap_values = explainer.shap_values(X[idx])
    shap.summary_plot(shap_values, features=X[idx], feature_names=feats,
show=False)
    plt.tight_layout();
    plt.savefig(OUT / "Q4_shap_summary.png", dpi=140);
    plt.close()
except Exception as e:
    print("SHAP 绘图失败: ", e)
# ===== 结束: 模型保存和 SHAP 可解释性分析 =====

# ===== 核心功能 12: 基线模型对比和系数分析 =====
# 作用: 训练 L1 正则化逻辑回归作为可解释基线, 分析特征重要性
print("[6/7] 基线对照: L1 Logistic (可解释) ...")
logit = make_pipeline(
    StandardScaler(with_mean=False),
    LogisticRegression(
        penalty="l1", solver="saga", class_weight="balanced",
        C=1.0, max_iter=5000, random_state=RANDOM_STATE
    )
)
cv = GroupKFold(n_splits=N_SPLITS)
oof_logit = np.zeros(len(yv))

```

```
for k, (tr, va) in enumerate(cv.split(X, yv, groups)):
    logit.fit(X[tr], yv[tr])
    oof_logit[va] = logit.predict_proba(X[va])[:, 1]

# 系数导出
logit.fit(X, yv)
clf = logit.named_steps["logisticregression"]
coefs = pd.DataFrame({"feature": feats, "coef": 
clf.coef_[0]}).sort_values("coef", key=np.abs, ascending=False)
coefs.to_csv(OUT / "Q4_logit_coefs.csv", index=False, encoding="utf-8-sig")
plt.figure(figsize=(8, 6), dpi=140)
topk = coefs.head(20).sort_values("coef")
plt.barh(topk["feature"], topk["coef"])
plt.title("基线 L1 Logistic 系数 (Top20 | 可正可负)")
plt.tight_layout();
plt.savefig(OUT / "Q4_logit_coef_plot.png");
plt.close()

# ===== 结束: 基线模型对比和系数分析 =====
```