Title

2025 年 9 月 7 日

摘　要

摘要

关键词： 关键词

一、问题重述

1.1 问题背景

问题背景 NIPT（Non-invasive Prenatal Test，无创产前检测）是一种通过采集孕妇外周血中胎儿的游离 DNA 片段，分析胎儿染色体是否异常的产前筛查技术。该技术主要用于早期发现如唐氏综合征（21 号染色体异常）、爱德华氏综合征（18 号染色体异常）和帕陶氏综合征（13 号染色体异常）等常见染色体疾病。NIPT 的准确性高度依赖于胎儿性染色体浓度：男胎 Y 染色体浓度需达到或超过 4%，女胎 X 染色体浓度需无异常。

孕妇的孕周、BMI（身体质量指数）等因素会影响胎儿染色体浓度的检测准确性。因此，合理分组孕妇并确定最佳检测时点，对提高检测成功率、降低因延迟发现胎儿异常所带来的风险（如治疗窗口期缩短）具有重要意义。

1.2 问题要求

问题一：分析胎儿 Y 染色体浓度与孕妇孕周数、BMI 等指标的相关性，建立关系模型并检验其显著性。

问题二：以男胎孕妇的 BMI 为主要因素，对其进行合理分组，确定每组的最佳 NIPT 时点，以最小化潜在风险，并分析检测误差对结果的影响。

问题三：综合考虑体重、年龄、检测误差及 Y 染色体浓度达标比例等因素，对男胎孕妇的 BMI 进行分组，确定每组的最佳 NIPT 时点，以最小化潜在风险，并分析检测误差的影响。

问题四：针对女胎孕妇，以 21、18、13 号染色体非整倍体为判定依据，结合 X 染色体 Z 值、GC 含量、读段数、BMI 等因素，建立女胎异常的判定方法。

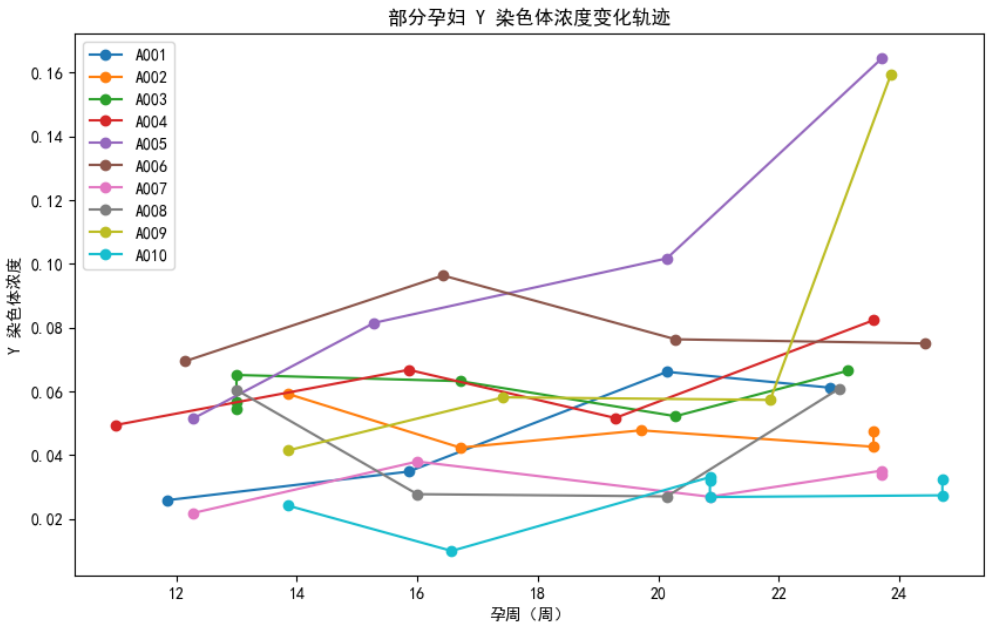
二、符号说明

符号	说明
GA	孕周
V	胎儿 Y 染色体浓度
β_0	胎儿 Y 染色体浓度基准水平
β_1	孕周与 Y 染色体浓度的回归关系系数
β_2	BMI 与 Y 染色体浓度的回归关系系数
β_3	孕周 ×BMI 与 Y 染色体浓度的回归关系系数
β_4	年龄与 Y 染色体浓度的回归关系系数
β_5	GC 含量与 Y 染色体浓度回归关系系数
β_6	IVF 妊娠方式与 Y 染色体浓度回归关系系数
β_7	影响 NIPT 最佳时点的参数的基准值
AG	孕妇年龄
GC	孕妇体内 GC 含量
u_i	测量误差 (残差)
T_r	是否为 NIPT 最佳时点的判断值
GA_{first}	首次达标孕周
$Risk()$	逻辑回归模型中的风险函数
ω_1	假阴性风险权重 (过早检测)
ω_2	过晚检测风险权重
$\mathbf{X}$	影响 NIPT 最佳时点的多元因素构成的向量
$\boldsymbol{\beta}^\top$	影响 NIPT 的变量向量对应的回归系数向量
W	孕妇体重
H	孕妇身高
$Excep()$	染色体非整倍异常判定函数
$\mathbf{T}$	影响染色体非整倍异常的变量构成的向量
$E()$	z 值、GC 含量、BMI 等因素的影响函数
λ_i	各影响函数所占权重

三、问题分析

- 针对问题一：本题的核心是定量探究胎儿 Y 染色体浓度 V、孕妇孕周数 GA 和 BMI 值之间的**相关性**。

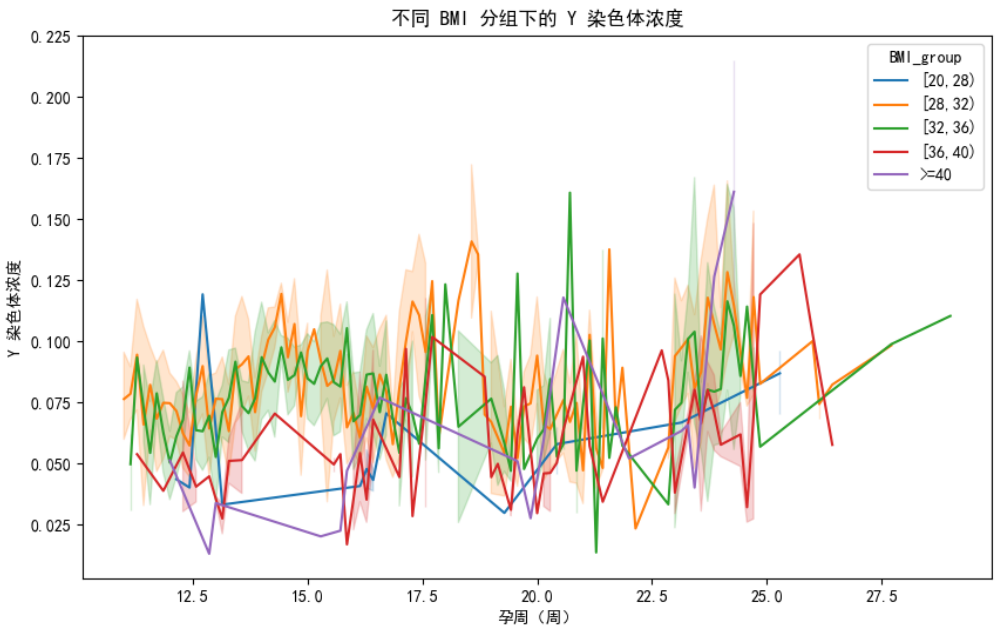
  注意到以上变量都是连续型的，我们最初考虑计算 Pearson 相关系数建立线性回归模型，但是，当我们绘制出部分孕妇 Y 染色体浓度变化轨迹如下图:



部分孕妇 Y 染色体浓度变化轨迹

可以清楚地发现 Y 染色体浓度和孕周之间单独线性关系很弱。

进一步绘制不同 BMI 分组下的 Y 染色体浓度如下图：



不同 BMI 分组下的 Y 染色体浓度

可以推测出胎儿 Y 染色体浓度随孕周和 BMI 的变化都非**简单线性**，那么就需要对变量做某种非线性变换，然而合适的变换方式并不能从数据中直观地看出，

于是我们想到了 Spearman 相关性分析，根据 Spearman 相关性矩阵的计算结果来考虑模型的构建。

- 针对问题二: 本题的核心是探索 BMI 与首次达标孕周的关系，求出合理 BMI 分组以及对应分组下的最佳 NIPT 时点, 并分析检测误差的影响。

  为了得到更可靠的结果，我们可以通过不同方案来求解。分位数统计法能快速得到假阴性率较低的结果，逻辑回归模型则是动态的、可预测风险的，这两个方案相得益彰。

  两个方案都需要先求出合理的 BMI 分组，聚类算法非常适合解决这个问题。我们采用 K-means 聚类算法，因为它足够高效。

- 针对问题三: 本题需在第二问基础上，进一步加入孕妇年龄、体重、检测误差及胎儿 Y 染色体浓度达标比例等多个变量，求出更合理的 BMI 分组以及对应分组下的最佳 NIPT 时点，并分析检测误差的影响。

  本情景下考虑的影响因素较多，分位数统计法很可能不再稳定，因此我们集中于优化和推广逻辑回归模型的一般形式，依赖风险函数决定最优 NIPT 时点。

- 针对问题四：本题最直接的方案是构建一个基于包括 X 染色体 Z 值、GC 含量、读段数、BMI 等影响因素在内的变量的异常判定模型，以判断胎儿是否存在 21 号、18 号或 13 号染色体非整倍体异常。

  这样的模型**可能**建立异常函数如：

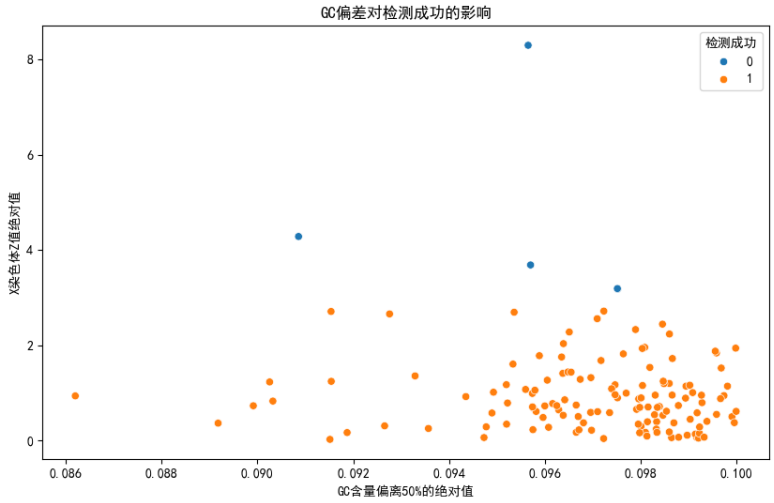
  $$Excep(\mathbf{T}) = \sum \lambda_i \cdot E(T_i)$$

  其中：

  - $Excep()$ 为异常函数。
  - $\mathbf{T}$ 为影响因素构成的向量。
  - $E(T_i)$ 为各影响因素变量的影响函数，其中，$E(T_0)$ 为 z 值影响函数。
  - λ_i 为各影响函数的权重。

  然后，根据 Excep() 函数的值来判定胎儿是否异常。但我们建模过程中发现，影响函数是难以拟合的。注意到 Excep() 函数的判定值是由样本数据决定的，我们想到了更简洁的解决方案，即设计合理的分组，计算不同分组下的 z 值阈值。

  在求解过程中，我们逐步发现 BMI 是影响 z 值的最主要因素，GC 含量影响 z 值阈值判定的准确度（如下图）————事实上 GC 含量本身就影响了测序质量准确度。



GC 偏差对检测成功的影响

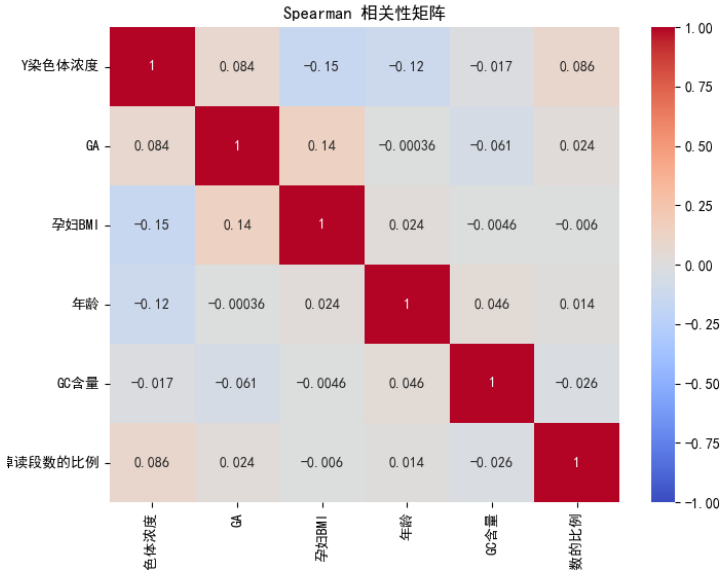
至于 z 值阈值的求解，这是一个典型的二分类问题，ROC 曲线（Receiver Operating Characteristic Curve）和 Youden 指数将能够很好地完成这个任务。

四、问题一模型的建立与求解

4.1 数据预处理

对孕周数据进行**标准化**处理，转换为连续数值（如 12w+3d 转换为 12.43 周）。针对 Y 染色体浓度 V，进行 Logit 变换或 Arcsin 平方根变换可以满足线性模型对因变量的要求。此外，还需处理缺失值，并基于测序质量指标剔除异常样本。

而更关键的另一个角色：Spearman 相关性矩阵，它的计算结果如下：



Spearman 相关性矩阵

观察图中数据，GA-Y 染色体浓度的数值为 0.084 说明孕周（GA）与 Y 染色体浓度相关性较弱，BMI-Y 染色体浓度的数值为-0.15, 年龄-Y 染色体浓度的数值为-0.12, 这说明 BMI 与年龄这两个变量与 Y 染色体浓度之间应该存在较强的负相关性。

4.2 模型的构建

我们这样逐步构建了线性混合模型（Linear Mixed Model，LLM）:
M1(孕周主效应)

$$y = \beta_0 + \beta_1 GA + u_i$$

其中：

- β_0 为基准水平, 表示自变量效应为 0 时，Y 染色体浓度水平。
- β_1 为孕周效应与 Y 染色体浓度回归关系系数，表示孕周对胎儿 Y 染色体浓度的影响。
- u_i 为残差，即实际值与计算值的差值。

M2(孕周 +BMI；独立效应)

$$y = \beta_0 + \beta_1 GA + \beta_2 BMI + u_i$$

其中：

- β_2 为 BMI 与 Y 染色体浓度回归关系的系数。

M3(孕周 ×BMI；交互效应)

$$y = \beta_0 + \beta_1 GA + \beta_2 BMI + \beta_3 GA \times BMI + u_i$$

其中：

- β_3 为孕周与 BMI 交互项回归系数。

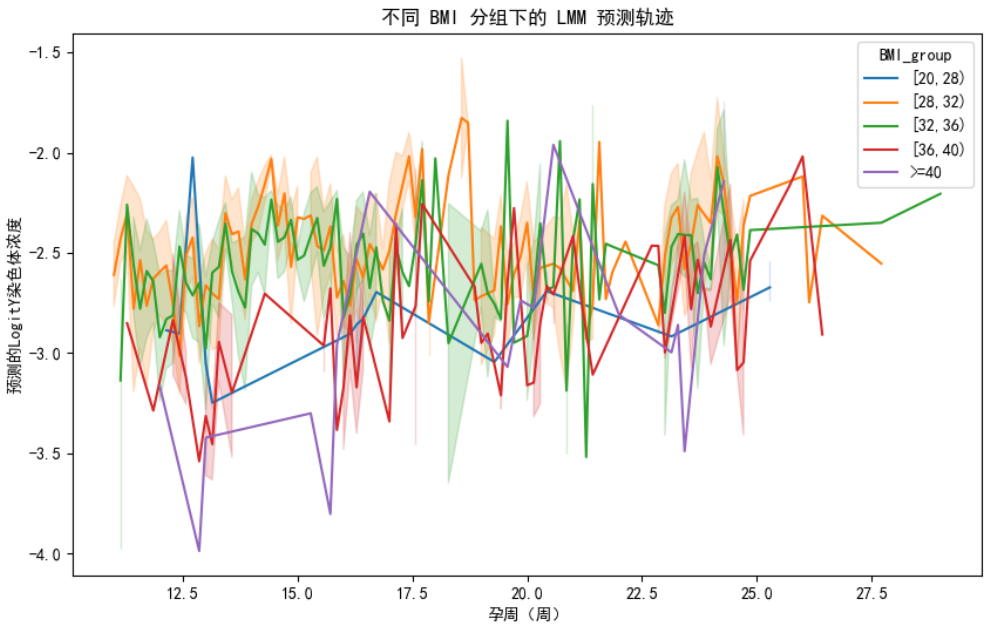
M4(协变量扩展)

$$y = \beta_0 + \beta_1 GA + \beta_2 BMI + \beta_3 (GA \times BMI) + \beta_4 AG + \beta_5 GC + \beta_6 I + u_i$$

其中：

- $\beta_i (i = 0, 1, ..., 6)$ 表示所乘变量的回归关系系数。
- AG 表示孕妇年龄效应值。
- GC 表示孕妇 GC 含量，即序列中碱基 G（鸟嘌呤）和 C（胞嘧啶）所占的比例。
- I 表示孕妇 IVF 妊娠方式效应值。

完整的 LLM 模型（M4）构建成功后，我们能够做出如下预测：

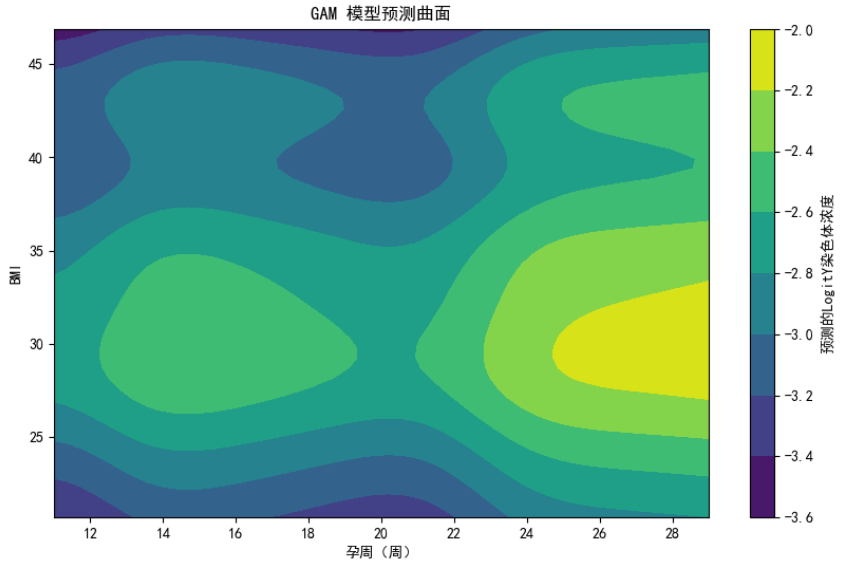


不同 BMI 分组下的 LMM 预测轨迹

为了获得更确切的结果，我们同时构建了广义可加模型（Generalized Additive Model,GAM）[1]，这个模型的一大优势是它可以出色地解释本题中的非线性关系，其具体表达式如下:

$$y = \beta_0 + f_1(GA) + f_2(BMI) + f_3(GA, BMI)$$

为获得更加直观的结果，我们将计算出的结果绘制成曲面，得到了 GAM 模型预测曲面如下：



GAM 模型预测曲面

接下来需要检验显著性，我们选择调用 MixedLM。在查阅参考资料 [2] 后，我

们考虑到似然比检验（Likelihood Ratio test，LRT）方法可能不稳定，遂参考模型 summary 中交互项系数的 z 值、p 值进行显著性判断。

以下为计算结果：

变量	p 值	说明
GA	0.688	单独孕周对 Y 染色体浓度没有显著影响
BMI	0.006	BMI 对 Y 染色体浓度有显著负向影响
$GA \times BMI$	0.104	BMI 对孕周影响的调节作用可能存在但不强
AG	0.103	年龄可能对 Y 染色体浓度有轻微负向影响
GC	0.012	测序质量影响结果
I	0.276	IVF 妊娠方式对 Y 染色体浓度影响不显著

4.3 结论

孕妇 BMI 对男胎 Y 染色体浓度具有显著负向影响，孕周的影响不显著（即使 BMI 可能调节孕周的效应，这种调节也是轻微的）。而孕周 ×BMI 的交互效应，处于边缘显著水平。

五、问题二模型的建立和求解

5.1 数据分析与处理

在这个模型的数据预处理中，我们同样需要将孕周 GA 转化为连续型变量。
同时，我们剔除了出现数值异常的样本。

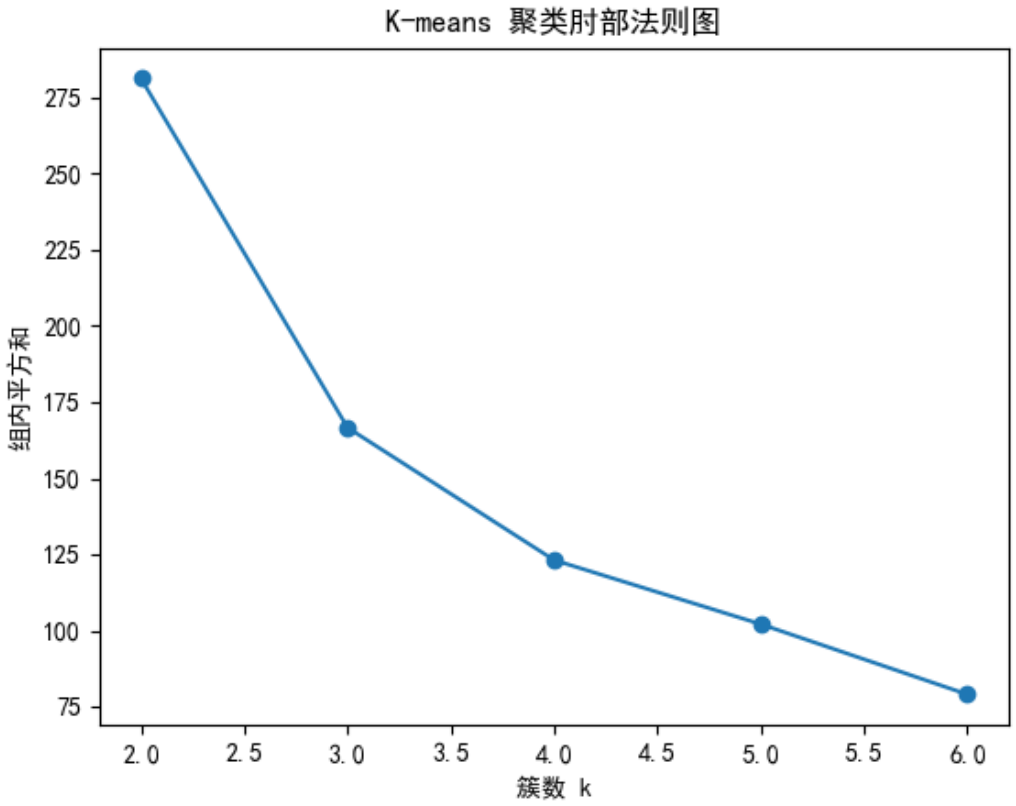
5.2 建模求解

5.2.1 BMI 分组与最佳 NIPT 时点

首先需要得到合理的 BMI 分组，我们通过 K-means 聚类算法求解。
确定聚类变量如下：

- BMI，表示孕妇体脂率。GA_{first}，表示首次达标孕周。

接下来需要确定 K-means 算法中的 k 值，我们选择通过肘部法来确定 k, 其法则如下图:



K-means 聚类肘部法则图

求出最佳 k 值为 4，我们综合考虑了聚类算法与等分分组的稳定性，得出合理的 BMI 分组如下表：

BMI 区间	BMI 最小值	BMI 最大值
(26.618, 29.885]	26.619343	29.878128
(29.885, 31.321]	29.891162	31.320929
(31.321, 33.419]	31.344816	33.409205
(33.419, 41.133]	33.428446	41.132812

接下来就可以求解最佳 NIPT 时点了。

对于每个 BMI 簇，计算 GA_{first} 的 80% 分位数作为最佳 NIPT 时点，因为取 80% 分位点能很好地在假阴性和过晚检测之间找到平衡。

计算结果如下表：

BMI Interval	$BMI_{Min}(\mathrm{kg}/m^2)$	$BMI_{Max}(\mathrm{kg}/m^2)$	Optimal Timing (week)
(26.618, 29.885)	26.619343	29.878128	16.000000
(29.885, 31.321)	29.891162	31.320929	18.600000
(31.321, 33.419)	31.344816	33.409205	15.485714
(33.419, 41.133)	33.428446	41.132812	20.028571

5.2.2 分析检测误差影响

建立逻辑回归模型：

$$T_r \sim GA + BMI + GA \times BMI$$

$$P(T_r = 1|GA, BMI) \in (0, 1)$$

其中：

- T_r 表示是否为 NIPT 最佳时点；若是，则值为 1，反之为 0。
- $P(T_r = 1|GA, BMI)$ 为达标概率曲线。

风险函数:

- 假阴性风险：$P(V < 0.04|GA, BMI)$。
- 过晚检测风险：$(GA - GA_{min})^2$，$GA_{min} = 10$。
- 总风险：

$$Risk(GA, BMI) = w_1 \cdot P(V < 0.04|GA, BMI) + w_2 \cdot (GA - 10)^2$$

其中：

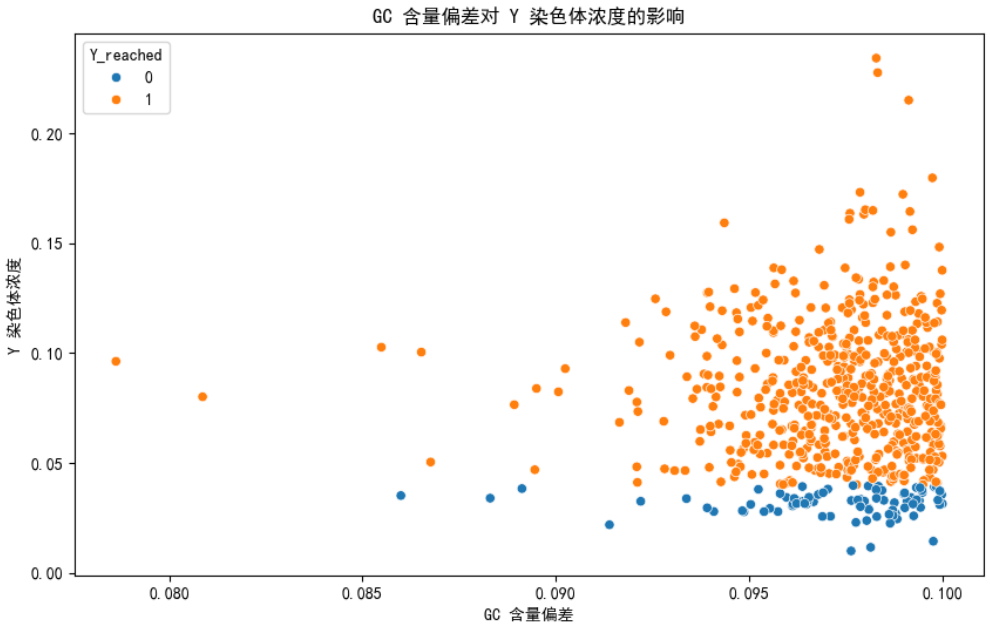
- ω_1 表示假阴性风险权重。
- ω_2 表示过晚检测风险权重。

误差的可能来源：

- GC 含量偏差：偏离 40%-60% 导致 V 测量不准确。
- 总读段数：过低增加 V 方差。
- 过滤比例：过高表示数据质量差。

通过计算风险函数、查询相关资料 [3] 发现,GC 偏离 50% 会导致测序偏倚直接影响 Y 染色体浓度的准确性, 因此可以认为 GC 含量偏差是误差的主要来源。

我们绘制出 GC 含量偏差对 Y 染色体浓度影响如下图：



GC 含量偏差对 Y 染色体浓度的影响

六、问题三模型的建立与求解

6.1 模型推广

本题我们基于第二题建立的模型进行推广，完善为更加一般的模型如下：

$$P(T_r = 1|\mathbf{X}) = \frac{1}{1 + \exp\big(-(\beta_7 + \boldsymbol{\beta}^\top \mathbf{X})\big)}$$

其中：

- $P(T_r = 1|\mathbf{X})$ 为达标概率曲线。
- $\mathbf{X}$ 表示孕妇 BMI、身高、体重、年龄等变量构成的向量。
- β_7 表示基准值。
- $\boldsymbol{\beta}^\top$ 为变量向量对应的回归系数向量。

在求解最佳 NIPT 时点分析风险函数之前，我们仍需要求出合理的 BMI 分组，K-means 算法再次发力：

- 确定聚类变量如下：
  - BMI, 表示孕妇体脂率。
  - GA_{first}，表示首次达标孕周。
  - AG ，表示孕妇年龄。
  - W，表示孕妇体重。
  - H, 表示孕妇身高。

- 求得最佳 k 值为 4，由于考虑的影响因素较多，我们认为等分分组不再具备足够稳定性，最终依据 k 值得到合理的 BMI 分组如下表：

BMI Interval	$BMI_{Min}(\mathrm{kg}/m^2)$	$BMI_{Max}(\mathrm{kg}/m^2)$
(26.619, 28.071)	26.619343	28.071089
(28.071, 31.888)	28.080249	31.887755
(31.888, 35.684)	31.896616	35.684444
(35.684, 41.133)	35.685352	41.132813

接下来讨论风险函数:

- 假阴性风险：$P(V < 0.04|\mathbf{X})$。
- 过晚检测风险：$(GA - GA_{min})^2$，$GA_{min} = 10$。
- 总风险：$Risk(\mathbf{X}) = \omega_1 \cdot P(V < 0.04|\mathbf{X}) + \omega_2 \cdot (GA - GA_{min})^2$，$GA_{min} = 10$

  其中：
  - ω_1 表示假阴性风险权重。
  - ω_2 表示过晚检测风险权重。

依据题意，我们取风险函数最小的孕周（Risk Optimal GA）作为最佳 NIPT 时点即可。

最终的 BMI 分组及最佳 NIPT 时点如下表：

BMI Interval	$BMI_{Min}(\mathrm{kg}/m^2)$	$BMI_{Max}(\mathrm{kg}/m^2)$	Risk Optimal GA(week)
(26.619, 28.071)	26.619343	28.071089	13.2
(28.071, 31.888)	28.080249	31.887755	13.3
(31.888, 35.684)	31.896616	35.684444	13.3
(35.684, 41.133)	35.685352	41.132813	14.1

七、问题四模型的建立与求解

7.1 数据分析与处理

分析中已经提到，GC 含量的偏离将同时影响测序质量和我们 z 值阈值判定的准确性，所以我们需要先筛选出 GC 含量在正常范围 (40% 60%) 内的孕妇。

我们取每位孕妇第一次检测数据，并标记染色体非整倍体异常的孕妇。

在运用 ROC 曲线和 Youden 指数之前，最关键的一步是给出 BMI 分组。在观察 K-means 算法给出的分组结果后，不失一般性地，我们给出这样的分组方案：

BMI Interval	Quantity
(25,30)	50
(30,35)	71
(35,40)	15
(40,+∞)	2

7.2 建模求解

对于上述分组，我们需要生成 ROC 曲线:

- $TPR = \frac{TP}{TP+FN}$ 表示真正例率。
- $FPR = \frac{FP}{FP+TN}$ 表示假正例率。其中：
  - TP 表示真实异常，预测异常。
  - TN 表示真实正常，预测正常。
  - FP 表示真实正常，预测异常。
  - FN 表示真实异常，预测正常。

接下来遍历每个阈值:

$$TPR(z) = \frac{TP(z)}{TP(z)+FN(z)} = \frac{TP(z)}{P}$$

$$FPR(z) = \frac{FP(z)}{FP(z)+TN(z)} = \frac{FP(z)}{N}$$

其中：

- P 是数据集中所有真实异常的样本总数。
- N 是数据集中所有真实正常的样本总数。
- $TP(z), FP(z), FN(z), TN(z)$ 的值都依赖于阈值 z 的选择。

同时，依靠 Youden 指数选择最优阈值：

$$J(z) = TPR(z) - FPR(z)$$

最后我们求得结果如下：

BMI Interval	z 值阈值
(25,30)	1.506379
(30,35)	1.000000
(35,40)	1.000000
(40,+∞)	3.0000000

故我们的结论是：对于 BMI 处于 (25,30) 内的孕妇，若 z 值高于 1.506379，则判定相应染色体存在非整倍体异常；对于 BMI 处于 (30,35) 内的孕妇，若 z 值高于 1，则判定相应染色体存在非整倍体异常；对于 BMI 处于 (35,40) 内的孕妇，若 z 值高于 1，则判定相应染色体存在非整倍体异常；对于 BMI 大于 40 的孕妇，若 z 值高于 3，则判定相应染色体存在非整倍体异常。

参考文献

[1] 统计机器学习方法（三十四）-广义加性模型 *GAM*. chinese. 2025. URL: https://zhuanlan.zhihu.com/p/718244682 (visited on 09/06/2025).

[2] 似然比检验 *LRT*. chinese. 2025. URL: https://blog.csdn.net/fjsd155/article/details/84866222 (visited on 09/06/2025).

[3] *GC bias GC* 偏好. chinese. 2025. URL: https://blog.csdn.net/qq_36654309/article/details/114539013 (visited on 09/06/2025).

附录 A 支撑材料文件列表

文件名	文件类型	简介
附件.xlsx	Excel 表格文件	孕妇数据集
1 数据预处理.py	python 代码文件	数据预处理代码
2 探索性分析.py	python 代码文件	探索性分析代码
3 构建关系模型与可视化.py	python 代码文件	模型关系代码
4 稳定性与敏感性分析.py	python 代码文件	稳定性与敏感性分析代码

附录 B 源数据

附录 C 所用软件

所用软件包括：Excel、PycharmCommunity、VisualStudioCode、TexStudio

附录 D 数据预处理代码

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.regression.mixed_linear_model import MixedLM
↪  #线性混合效应模型
from pygam import LinearGAM, s # 广义线性模型
import statsmodels.api as sm
from datetime import datetime
from scipy.stats import spearmanr # 相关性检验

plt.rcParams['font.sans-serif'] = ['SimHei']  # 黑体，支持中文
plt.rcParams['axes.unicode_minus'] = False    # 解决负号显示为方块

data = pd.read_excel(" 附件.xlsx",sheet_name=" 男胎检测数据")

# 数据预处理
# 转换孕周函数（小数表示）
def convert_gestational_age(ga_str):
    if isinstance(ga_str, str):  # 检查是否为字符串
        try:
            # 将大写 W 转换为小写 w，确保兼容大小写
            ga_str = ga_str.lower()
            # 分割字符串，提取周数和天数
            weeks_part = ga_str.split('w')[0].strip()
            weeks = int(weeks_part)  # 转换为整数
            # 检查是否有天数部分
            days = 0
            if '+' in ga_str:
```

```
                days_part = ga_str.split('+')[1].strip()
                days = int(days_part) if days_part else 0
            # 确保周数和天数合理
            if weeks >= 0 and 0 <= days < 7:
                return weeks + days / 7
            else:
                return np.nan
        except (ValueError, IndexError):
            return np.nan
    return np.nan

# 读取并清洗列名
data = pd.read_excel(" 附件.xlsx", sheet_name=" 男胎检测数据")
# 去掉首尾空白与换行
data.columns = data.columns.str.strip().str.replace("\n", "",
↪  regex=True)
print("Columns in sheet:", data.columns.tolist())
# 计算孕周数值
data['GA'] = data['检测孕周'].apply(convert_gestational_age)

# 对 Y 染色体浓度进行 logit 变换
epsilon = 1e-6  # 避免溢出
data['Y_concentration_logit'] = np.log(data['Y染色体浓度'] / (1 -
↪  data['Y染色体浓度'] + epsilon))

# 处理日期（末次月经时间和检测日期，后续计算时间差）
data['末次月经'] = pd.to_datetime(data['末次月经'])
data['检测日期'] = pd.to_datetime(data['检测日期'])

# 检查缺失值
print(" 缺失值检查: ")
print(data.isnull().sum())
```

附录 E 探索性分析代码

```
# 探索性分析 (EDA)
# 轨迹图: 每位孕妇的 Y 染色体浓度随孕周变化
plt.figure(figsize=(10, 6))
for id in data['孕妇代码'].unique()[:10]:  # 展示前 10 个孕妇,
↪  直观感受趋势。
    subset = data[data['孕妇代码'] == id]
    plt.plot(subset['GA'], subset['Y染色体浓度'], marker='o',
    ↪  label=id)
plt.xlabel('孕周（周）')
plt.ylabel('Y 染色体浓度')
plt.title('部分孕妇 Y 染色体浓度变化轨迹')
plt.legend()
```

```
plt.show()

# BMI 分层分析
bmi_bins = [20, 28, 32, 36, 40, np.inf]
bmi_labels = ['[20,28)', '[28,32)', '[32,36)', '[36,40)', '>=40']
data['BMI_group'] = pd.cut(data['孕妇BMI'], bins=bmi_bins,
↪  labels=bmi_labels, right=False)

plt.figure(figsize=(10, 6))
sns.lineplot(x='GA', y='Y 染色体浓度', hue='BMI_group', data=data)
plt.xlabel('孕周（周）')
plt.ylabel('Y 染色体浓度')
plt.title('不同 BMI 分组下的 Y 染色体浓度')
plt.show()

# Spearman 相关性分析
corr_vars = ['Y染色体浓度', 'GA', '孕妇BMI', '年龄', 'GC含量',
↪  '被过滤掉读段数的比例']
corr_matrix = data[corr_vars].corr(method='spearman')
plt.figure(figsize=(8, 6))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', vmin=-1,
↪  vmax=1)
plt.title('Spearman 相关性矩阵')
plt.show()
```

附录 F 模型关系代码

```
# 模型构建
# 线性混合效应模型 (LMM)
# 模型 M1: 仅包含孕周主效应
model_m1 = MixedLM.from_formula(
    "Y_concentration_logit ~ GA",
    data,
    groups=data['孕妇代码']
)
result_m1 = model_m1.fit()
print(" 模型 M1（仅孕周): ")
print(result_m1.summary())

# 模型 M2: 孕周 + BMI
model_m2 = MixedLM.from_formula(
    "Y_concentration_logit ~ GA + 孕妇 BMI",
    data,
    groups=data['孕妇代码']
)
result_m2 = model_m2.fit()
print(" 模型 M2（孕周 + BMI): ")
```

```
print(result_m2.summary())

# 模型 M3: 孕周 + BMI + 交互项
data['GA_BMI_interaction'] = data['GA'] * data['孕妇 BMI']
model_m3 = MixedLM.from_formula(
    "Y_concentration_logit ~ GA + 孕妇 BMI + GA_BMI_interaction",
    data,
    groups=data['孕妇代码']
)
result_m3 = model_m3.fit()
print(" 模型 M3（孕周 + BMI + 交互项): ")
print(result_m3.summary())

# 模型 M4: 加入协变量
model_m4 = MixedLM.from_formula(
    "Y_concentration_logit ~ GA + 孕妇BMI + GA_BMI_interaction + 年龄
    ↪  + GC含量 + IVF妊娠",
    data,
    groups=data['孕妇代码']
)
result_m4 = model_m4.fit()
print(" 模型 M4（完整模型，含协变量): ")
print(result_m4.summary())

# 非线性模型 (GAM) 使用 pygam
gam = LinearGAM(s(0, n_splines=10) + s(1, n_splines=10)).fit(
    data[['GA', '孕妇 BMI']], data['Y_concentration_logit']
)
print("GAM 模型结果: ")
print(gam.summary())

# 可视化模型结果
# LMM 预测轨迹
data['predicted_m3'] = result_m3.fittedvalues
plt.figure(figsize=(10, 6))
sns.lineplot(x='GA', y='predicted_m3', hue='BMI_group', data=data)
plt.xlabel('孕周（周）')
plt.ylabel('预测的 LogitY 染色体浓度')
plt.title('不同 BMI 分组下的 LMM 预测轨迹')
plt.show()

# GAM 预测曲面
XX, YY = np.meshgrid(np.linspace(data['GA'].min(), data['GA'].max(),
↪  50),
                     np.linspace(data['孕妇BMI'].min(),
                     ↪  data['孕妇BMI'].max(), 50))
Z = gam.predict(np.c_[XX.ravel(), YY.ravel()]).reshape(XX.shape)
```

```
plt.figure(figsize=(10, 6))
contour = plt.contourf(XX, YY, Z, cmap='viridis')
plt.colorbar(contour, label='预测的 LogitY 染色体浓度')
plt.xlabel('孕周（周）')
plt.ylabel('BMI')
plt.title('GAM 模型预测曲面')
plt.show()
```

附录 G 稳定性与敏感性分析代码

```
# 稳定性与敏感性分析
# 剔除异常值（例如 Y 染色体浓度 < 0.01）
data_robust = data[data['Y 染色体浓度'] > 0.01]
model_m3_robust = MixedLM.from_formula(
    "Y_concentration_logit ~ GA + 孕妇 BMI + GA_BMI_interaction",
    data_robust,
    groups=data_robust['孕妇代码']
)
result_m3_robust = model_m3_robust.fit()
print(" 模型 M3（健壮性分析，剔除 Y 浓度 < 0.01 的样本）: ")
print(result_m3_robust.summary())
```

附录 H

```
data['GA'] = data['检测孕周'].apply(convert_gestational_age)
data['Y_reached'] = (data['Y 染色体浓度'] >= 0.04).astype(int)

# 过滤异常值
data = data[(data['GC 含量'] >= 0.4) & (data['GC 含量'] <= 0.6)]


first_reached = data[data['Y_reached'] ==
↪  1].groupby('孕妇代码')['GA'].min().reset_index()
first_reached = first_reached.rename(columns={'GA': 'GA_first'})
data_unique = data.drop_duplicates('孕妇代码')[['孕妇代码',
↪  '孕妇BMI']].merge(first_reached, on='孕妇代码')


X_cluster = data_unique[['孕妇 BMI', 'GA_first']].dropna()
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_cluster)


inertias = []
for k in range(2, 7):
    kmeans = KMeans(n_clusters=k, random_state=42)
```

```
    kmeans.fit(X_scaled)
    inertias.append(kmeans.inertia_)
plt.plot(range(2, 7), inertias, marker='o')
plt.xlabel('簇数 k')
plt.ylabel('组内平方和')
plt.title('K-means 聚类肘部法则图')
plt.show()

kmeans = KMeans(n_clusters=4, random_state=42)
data_unique['Cluster'] = kmeans.fit_predict(X_scaled)

bmi_edges = pd.qcut(data_unique['孕妇 BMI'], q=4)
data_unique['BMI_group'] = bmi_edges
bmi_ranges =
↪  data_unique.groupby('BMI_group')['孕妇BMI'].agg(BMI_Min='min',
↪  BMI_Max='max')
```

附录 I

```
X = data[['GA', '孕妇 BMI']].dropna()
y = data['Y_reached'].dropna()
model_lr = LogisticRegression().fit(X, y)

def risk_function_updated(ga, bmi, model, w1=0.7, w2=0.3):
    X_pred = pd.DataFrame({'GA': [ga], '孕妇 BMI': [bmi]})
    prob_false_negative = 1 - model.predict_proba(X_pred)[:, 1]
    delay_penalty = ((ga - 12) / 15) ** 2
    return w1 * prob_false_negative + w2 * delay_penalty

ga_grid = np.arange(10, 25, 0.1)

quantiles_80 = data_unique.groupby('BMI_group')['GA_first'].quantile」
↪  (0.8).rename('NIPT_80_Quantile')
risk_results_updated = {}
for group in data_unique['BMI_group'].unique():
    bmi_mean = data_unique[data_unique['BMI_group'] ==
    ↪  group]['孕妇BMI'].mean()
    risks = [risk_function_updated(ga, bmi_mean, model_lr) for ga in
    ↪  ga_grid]
    optimal_ga = ga_grid[np.argmin(risks)]
    risk_results_updated[group] = optimal_ga
```

附录

```
features = ['孕妇 BMI', 'GA_first', '身高', '体重', '年龄']
X_cluster = data_unique[features].dropna()
```

```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_cluster)

kmeans = KMeans(n_clusters=3, random_state=42)
data_unique.loc[X_cluster.index, 'Cluster'] =
↪  kmeans.fit_predict(X_scaled)

cluster_order = data_unique.groupby('Cluster')['孕妇BMI'].mean().sor
↪  t_values().index
cluster_map = {old: new for new, old in enumerate(cluster_order)}
data_unique['Cluster_ordered'] =
↪  data_unique['Cluster'].map(cluster_map)

cluster_stats_ordered = data_unique.groupby('Cluster_ordered').agg({
    '孕妇 BMI': ['min', 'max', 'mean'],
    '身高': 'mean',
    '体重': 'mean',
    '年龄': 'mean',
}).reset_index()
cluster_stats_ordered.columns = ['Cluster', 'BMI_Min', 'BMI_Max',
↪  'BMI_Mean', 'Height_Mean', 'Weight_Mean', 'Age_Mean']
```

附录

```
 # 特征工程
data['GC 偏差'] = np.abs(data['GC 含量'] - 0.5)
data['读段质量'] = data['唯一比对的读段数'] / data['原始读段数']
data['Z值综合'] = data[['13号染色体的Z值', '18号染色体的Z值',
↪  '21号染色体的Z值']].abs().mean(axis=1)
data['检测成功'] = (data['X 染色体的 Z 值'].abs() < 3).astype(int)

# 过滤数据
data = data[(data['GC 含量'] >= 0.4) & (data['GC 含量'] <= 0.6) &
            (data['GA'] >= 10) & (data['GA'] <= 25)].copy()

# 取每位孕妇第一次检测数据
data_unique = data.sort_values(['孕妇代码',
↪  'GA']).groupby('孕妇代码').first().reset_index()

# 异常标签（基于 AB 列）
data_unique['异常'] = data_unique['染色体的非整倍体'].apply(lambda x:
↪  1 if pd.notna(x) and x != '' else 0)
```

附录

```
bins = [20, 25, 30, 35, 40, float('inf')]
labels = ['[20, 25)', '[25, 30)', '[30, 35)', '[35, 40)', '40+']
data_unique['BMI分组'] = pd.cut(data_unique['孕妇BMI'], bins=bins,
↪  labels=labels, right=False)

for label in labels:
    group_data = data_unique[data_unique['BMI分组'] ==
    ↪  label].dropna(subset=['Z值综合', '异常'])
    print(f"BMI 组 {label} 样本量: {len(group_data)}")

z_thresholds = {}
for label in labels:
    group_data = data_unique[data_unique['BMI分组'] ==
    ↪  label].dropna(subset=['Z值综合', '异常'])
    if len(group_data) > 5:  # 要求至少 5 个样本
        # 初始阈值: 均值 + 3 倍标准差
        initial_threshold = group_data['Z值综合'].mean() + 3 *
        ↪  group_data['Z值综合'].std()
        fpr, tpr, thresholds = roc_curve(group_data['异常'],
        ↪  group_data['Z值综合'])
        youden_idx = np.argmax(tpr - fpr)
        z_thresholds[label] = max(min(thresholds[youden_idx], 5.0),
        ↪  1.0)  # 限制范围1.0-5.0
    else:
        z_thresholds[label] = 3.0  # 默认值, 若样本量不足

optimal_ga = data_unique.groupby('BMI 分组')['GA'].median()
abnormal_rate =
↪  data_unique.groupby('BMI分组')['异常'].mean().fillna(0)  #
↪  填充NaN为0

results_df = pd.DataFrame({
    'BMI 范围': labels,
    'Z 值阈值': [z_thresholds[label] for label in labels],
    '最佳孕周': optimal_ga,
    '异常率': abnormal_rate
})
results_df.to_csv('nipt_fourth_results.csv', index=False)

print("\n结果表格: ")
print(results_df)
```