吉林大学数学建模竞赛

参赛编号：____049_____

题目：_____B_____

2025

Title

2025 年 9 月 4 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
G_a	孕周
V	胎儿 Y 染色体浓度

三、问题分析

- 针对问题一：本问题的核心是定量探究胎儿 Y 染色体浓度 V、孕妇孕周数 G_a 和 BMI 值之间的统计关系。依题意，得有以下思路：

  首先进行**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（如 GC 含量、过滤比例）剔除异常样本。

  接着进行**探索性分析 (EDA)**，绘制 Y 染色体浓度随孕周变化的散点图和平滑曲线并按 BMI 分层展示其变化趋势。计算 Spearman 相关系数，以初步评估 Y 染色体浓度与孕周、BMI 之间的相关性。

  然后开始**构建模型**：考虑到数据可能存在非线性和个体重复测量的情况（同一孕妇多次检测），我们决定在广义可加混合模型 (GAMM) 和线性混合效应模型 (LMM) 做出选择。我们的模型将包含孕周 G_a 的非线性平滑项、BMI 的线性

（或平滑）项、以及可能的孕周与 BMI 的交互项。孕妇 ID 将被设置为随机截距，以控制个体差异。

之后检验与解释**显著性**，通常使用**似然比检验 (LRT)**，或者基于 REML 的 **F 检验**来评估孕周平滑项、BMI 主效应及交互项的统计显著性。绘制不同 BMI 水平下 Y 染色体浓度随孕周变化的预测曲线来更直观解释变量间的关系。

最后分析**稳健性**：检查残差、尝试不同的变量变换方式、在数据子集上进行交叉验证，以检验最终模型的稳健性和可靠性。

- 针对问题二:
- 针对问题三:

四、问题一模型的建立与求解

4.1 数据分析与处理

4.2 建模求解

五、问题二模型的建立和求解

5.1 数据分析与处理

5.2 建模求解

六、问题三模型的建立与求解

6.1 数据分析与处理

6.2 建模求解

6.3 模型效果检验

6.4 策略分析

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
相关特性与模型显著性检验.py	python 代码文件	检验相关性和显著性的代码

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 代码

```python
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

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

# 1. 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())

# 2. 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('Gestational Age (weeks)')
plt.ylabel('Y Chromosome Concentration')
plt.title('Y Concentration Trajectories for Selected Pregnant Women')
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('Gestational Age (weeks)')
plt.ylabel('Y Chromosome Concentration')
plt.title('Y Concentration by BMI Group')
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 Correlation Matrix')
plt.show()

# 3. 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("Model M1 (GA only):")
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("Model M2 (GA + 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("Model M3 (GA + BMI + Interaction):")
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("Model M4 (Full Model):")
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 Model Summary:")
print(gam.summary())

# 4. 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('Gestational Age (weeks)')
plt.ylabel('Predicted Logit(Y Concentration)')
plt.title('LMM Predicted Trajectories by BMI Group')
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='Predicted Logit(Y Concentration)')
plt.xlabel('Gestational Age (weeks)')
plt.ylabel('BMI')
plt.title('GAM Predicted Surface')
plt.show()

# 5. 显著性检验
# 比较 M2 和 M3 (交互项显著性)
from scipy.stats import chi2
llf_m2 = result_m2.llf
```

```
llf_m3 = result_m3.llf
df_diff = result_m3.df_modelwc - result_m2.df_modelwc
lrt_stat = -2 * (llf_m2 - llf_m3)
p_value = chi2.sf(lrt_stat, df_diff)
print(f"LRT for Interaction (M2 vs M3): Stat = {lrt_stat:.2f},
↪  p-value = {p_value:.4f}")

# 6. 健壮性与敏感性分析
# 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("Model M3 (Robust, Y > 0.01):")
print(result_m3_robust.summary())

# 7. 统计结论
print("\n统计结论：")
print("- 孕周 (GA) 与 Y 染色体浓度呈显著正相关，非线性趋势明显。")
print("- BMI 对 Y 染色体浓度有负调节作用，高 BMI 孕妇的 Y 浓度上升较慢。
↪  ")
print("- GA × BMI 交互效应显著，BMI 调节了孕周对 Y 浓度的影响。")
print("- 测序质量指标（如 GC 含量）需校正以减少混杂效应。")
```