题目

摘要

第一段：针对自己选择的题目，说明自己用了什么方法来解决的（这类题属于哪种典型的问题），其中利用了哪些关键的算法，再说出自己的所建模型的创新点。没有创新点，也可以说自己所建的模型相比较于其它的是一个很好的方案。

第二段：针对问题一中的具体问题，进行分析和求解，几句话介绍自己是怎么解决的，有数字结果的也可以直接贴结果。

。

。

第三段：问题二中，类比于第二段。

。

。

。

第四段：问题三中，类比于第三段。

。

。

。

第五段：问题四中，类比于第四段。

。

。

。

第六段：如果有问题五，类比于第五段，没有就结束，也可以写一下团队的想法。

。

。

。

关键词：

1 问题重述

1.1 问题背景

NIPT是一项革命性的产前筛查技术，旨在通过分析孕妇血液中的胎儿游离DNA，以早期、安全、准确地判断胎儿是否存在常见的染色体异常。NIPT仅需抽取孕妇的静脉血，对胎儿和母亲几乎无风险。其原理是：孕妇血液中存在少量来自胎儿的游离DNA片段。通过高通量测序技术对这些DNA片段进行测序和生物信息学分析，可以计算出每条染色体所占的比例。

1.2 问题提出

NIPT结果的准确性高度依赖于血液中胎儿游离的DNA浓度，一般来说，孕期时间越长，胎儿游离的DNA浓度越高；孕妇BMI越高，胎儿游离的DNA浓度越低。而发现异常的时机至关重要：早期（<12周）：发现异常，风险低，处理选择多，对孕妇身心影响小；中期（13-27周）：风险高，处理难度和伤害增大；晚期（>28周）：风险极高，可能面临引产等艰难决定。因此，找到一个“最佳检测时点”至关重要：这个时点要尽可能早（以降低后期风险），但又必须足够晚，以确保胎儿DNA浓度达标，使检测结果准确可靠。这个“最佳时点”因孕妇的BMI而异。在实际检测中，测序工作会因检测时点过早和不确定因素影响等情况失败。同时为了增加检测结果的可靠性，对某些孕妇有多次采血多次检测或一次采血多次检测的情况。

问题一：

基于提供的NIPT数据集，定量分析胎儿Y染色体浓度（ffY）与孕妇孕周（Gestational Age）及身体质量指数（BMI）等关键指标之间的统计关联性。需构建一个反映其内在关系的理论模型，并采用适当的统计检验方法对模型的显著性、拟合优度以及变量的贡献度进行严格评估，以验证模型的有效性和解释力。

问题二：

临床研究表明，孕妇BMI是影响男胎Y染色体浓度达标时间（即ffY $\geq$ 4%的最早孕周）的首要因素。需要依据BMI对孕妇群体进行聚类分析或最优区间划分，形成同质性较高的分组；同时为每个BMI分组确定一个最优NIPT检测时点（孕周），该时点的选择需以最小化孕妇因延迟发现胎儿异常所面临的潜在临床风险为优化目标。风险函数应与发现异常的孕周相关联；最后进行不确定性分析或灵敏度分析，量化检测误差（如Y染色体浓度的测量误差）对最佳时点决策稳健性的影响。

问题三：

在问题二的基础上，进一步考虑体重、年龄等多维协变量对Y染色体浓度达标时间及达标比例的综合影响。根据男胎孕妇的 BMI，给出合理分组以及每组的最佳 NIPT 时点，使得孕妇潜在风险最小，并分析检测误差对结果的影响。

问题四：

针对女胎无Y染色体的特点，需开发一种不依赖于Y染色体浓度的异常判定方法。要求以临床确诊的21、18、13号染色体非整倍体结果（AB列）为金标准，构建一个多特征融合的分类模型，综合考虑 X 染色体及上述染色体的 Z 值、GC 含量、读段数及相关比例、BMI 等因素，给出女胎异常的判定方法。

2 问题分析

2.1 问题一的分析

问题一需要建立一个数学模型，用来描述胎儿Y染色体浓度与孕妇各项指标，尤其是BMI与孕周的关系。可以由以下4步完成：

第1步：数据清洗与预处理

1. **处理缺失值与异常值**：

   - 检查列是否存在缺失值，可采用删除或合理的插补方法。
   - 检查异常值：例如，孕周是否在合理范围内（如10-25周），Y染色体浓度是否出现极端值（如远超100%或为负值），GC浓度是否处于40%-60%。可将异常值分为检测型异常与生物学型异常，剔除检测型异常，保留但标记生物学型异常。可通过箱线图（Boxplot）或散点图进行可视化识别，并对异常值进行处理（如剔除或修正）。

     检测型异常：由于检测手段导致的异常，本质上是实验操作、仪器或试剂等方面出现的问题，而非样本真实的生物学状态的反应。

     生物型异常：在实验技术操作完全正确、无误的情况下，由于样本本身固有的、非常规的生物学特性或状态，所导致的偏离预期或常规模式的结果。它反映的可能是个体差异或特殊的病理状态。

2. **数据转换**：

   - 检查数据分布。如果Y染色体浓度呈偏态分布，可以考虑进行对数变换或平方根变换，使其更接近正态分布，以满足后续线性回归的假设。
   - 新增怀孕天数一列，将孕周数据转化为整形类型数据。

第2步：探索性数据分析

在建立复杂模型前，先用可视化工具直观感受变量间的关系。通过绘制Y染色体浓度分别与孕周、BMI等属性的散点图，计算Pearson相关系数（对于线性关系）或Spearman秩相关系数（对于单调非线性关系）来量化变量之间的关联强度，初步判断哪些变量与因变量（Y浓度）的相关性更强。

第3步：模型建立

根据第2步中得到的信息，若显示出线性关系，可以使用多元线性回归模型；若显示出明显的曲线关系则需引入多项式项或使用广义可加模型。使用统计软件对上述模型进行拟合，得到回归系数的估计值。

第4步：模型检验与显著性分析

采用F检验对模型显著性进行评估，t检验对变量显著性进行评估。以R方决定系数评估模型的拟合优度。

2.2 问题二的分析

在问题一的基础上，问题二的目标是：利用Y染色体浓度与孕周、BMI的关系，为不同BMI的孕妇分组，并找到每组的最佳检测时间，以最小化潜在风险。从问题一得到的模型$Y = f(\text{BMI}, \text{孕周})$，可令$f(\text{BMI}, \text{孕周}) \geqslant 4\%$，对于某一特定的BMI值，可反解出对应的最小孕周$T_{\min}(\text{BMI})$，定义为该BMI下孕妇的理论最早达标时间。题目描述早期发现风险低，中期风险高，晚期风险极高，说明需要一个数学函数$\text{Risk}(t)$来分段量化在孕周t进行检测并发现异常时所带来的风险。

第1步：按BMI进行合理分组

可以基于“达标时间”聚类：直接对所有孕妇的$T_{\min}(\text{BMI})$进行聚类，聚类后同一组内的孕妇具有相似的达标时间。然后反过来查看每个聚类的BMI范围。

也可以计算$T_{\min}(\text{BMI})$关于BMI的导数，了解达标时间随BMI变化的敏感度。在变化剧烈的BMI区间，分组应更精细；在变化平缓的区间，分组可更粗粒度。以组内方差最小化为目标，寻找最佳的区间划分点。

第2步：为每个BMI分组确定最佳NIPT时点

对于第 i 个BMI分组，找到一个共同的检测孕周T_i^*，使得该组内所有孕妇的期望风险最小。可以用检测失败与发现晚两方面风险加权得到目标函数（期望风险）。

第3步：分析检测误差的影响（灵敏度分析）

检测误差主要指Y染色体浓度的测量值V存在误差。假设误差ε为高斯白噪，即$\varepsilon \sim N(0, \sigma^2)$。测量值可表示为$V_{\text{测量}} = V_{\text{真实}} + \varepsilon$。该误差会传递到问题一的回归模型中，导致预测的$Y$和反解出的$T_{\min}(\text{BMI})$存在不确定性。误差也会传递到优化过程中，导致计算出的最佳时点T_i^*产生波动。可以通过模特卡洛模拟，以不同的噪声得到不同的分组方案与最佳时点。查看这些时点的分布与关键统计量（如方差），方差越大，说明模型对误差越敏感，结果越不稳定。查看分组BMI区间的变化频率，评估分组的鲁棒性。

2.3 问题三的分析

在问题二的基础上，问题三要求考虑更多的因素（年龄、身高、体重等）。因此，需要拓展问题一的模型，将年龄、身高、体重等显著变量作为新的自变量加入问题一中。新模型形式将变为$Y_{\text{new}} = f(\text{孕周}, \text{BMI}, \text{年龄}, \text{体重}, \text{身高}, ...)$，随后进行显著性检验与残差分析，确保新模型有效。可能需要使用逐步回归等方法筛选出最相关的变量，防止过拟合。同时注意到身高、体重与BMI高度相关，可以只选其一或使用主成分回归将其合并为综合指标。

相较于问题二，问题三不再寻求一个确定的“最早达标时间”，而是计算在某个孕周t，对于具有特定特征的孕妇，其Y染色体浓度达标（$\geq 4\%$）的概率。我们的回归模型$Y_{\text{new}} = f(t, \text{其他特征}) + \varepsilon$，其中$\varepsilon \sim N(0, \sigma^2)$是误差项，服从正态分布。因此，对于固定孕周 t 和特征值，Y_{new}的预测值也是一个正态分布：$N(\mu(t, \text{其他特征}), \sigma^2)$。达标概率即为这个正态分布随机变量大于等于4%的概率：

$$P(\text{达标} | t, \text{其它特征}) = 1 - \Phi\left((4\% - \mu(t, \text{其他特征})) / \sigma \right)$$

其中 Φ 是标准正态分布的累积分布函数。

不再仅按BMI分组，而是按所有重要特征（如BMI、年龄、体重）进行综合分组。选择进入聚类模型的特征，对它们进行标准化，使每个特征具有相同的尺度。采用K-Means聚类或高斯混合模型对标准化后的特征向量进行聚类。使用肘部法则或轮廓系数来确定最优的聚类数量K。目标是将特征相似的孕妇分到同一组。聚类完成后，分析每个簇的中心点，来描述每个组的典型特征。

问题三需要精细化风险函数。在孕周 t 为一位特征为 X 的孕妇进行检测，其期望风险为：

$$E_{\text{risk}}(t, X) = [1 - P(\text{达标} | t, X)] \, R_{\text{fail}} + P(\text{达标} | t, X) \, \text{Risk}(t)$$

第一项：检测失败的风险。$1 - P(\text{达标} \,|\, t, X)$是失败概率，$R_{\text{fail}}$是失败带来的代价（一个较大的常数）。

第二项：检测成功但发现晚的风险。$P(\text{达标} \,|\, t, X)$ 是成功概率，$\text{Risk}(t)$是时间风险函数（与问题二相同）。

对于第 i 个聚类分组，组内包含多个孕妇（或其特征向量）。我们的目标是找到一个共同的检测时点T_i^*，使得该组的平均期望风险最小。同样在孕周范围 $[10, 25]$ 内进行网格搜索，计算每个候选孕周 t 对应的组内平均期望风险，选择风险最小的 t 作为最佳时点。

问题三的误差来源可分为以下3点：

- **测量误差**：Y染色体浓度的测量误差（如问题二所述）。
- **模型不确定性**：回归模型$f(t, \text{其它特征})$本身的参数（系数）估计是有不确定性的。
- **分组不确定性**：聚类分组的结果可能对初始值敏感。

同样可以使用蒙特卡洛模拟同时考虑多种不确定性。从回归模型系数的分布中抽样（假设系数服从多元正态分布），为Y浓度测量值添加随机误差。每次抽样后，重新计算达标概率，重新进行聚类分组（或扰动分组），重新优化最佳时点。重复上述过程多次，得到最佳时点T_i^*的一个分布。报告每个分组的最佳时点T_i^*的均值和置信区间（如95%区间）。区间越宽，说明模型对误差越敏感，结果的稳健性越差。同时可以绘制每个分组的最佳时点分布直方图，直观感受。

2.4 问题四的分析

问题四不再是寻求最佳的检测时间点，而是希望得到判定女胎异常的方案，其目标是构建一个基于多源数据的分类模型，用于判定女胎染色体（21, 18, 13号）是否存在异常。

2.5 总思路图（可选）

图 1. 总思路图（随便找的网图）

3 模型假设

1. 。。。

   。

2. 。。。

   。

3. 。。。

   。

4. 。。。

   。

4 符号说明

符号	说明	单位
d	两点间的距离	m
t	时间变量	s
v	速度	m/s
l	物体长度或路径长度	m
(x_i, y_i)	第i个点的平面坐标	-
θ_i	第i个角度变量	rad
A_i	第i个区域的面积	m^2
B_i	第i个模型的系数矩阵	-
C_i	第i类对象的成本或代价	元
α, β	模型参数（如权重系数）	-
ρ	密度	$\mathrm{kg/m}^3$
λ	到达率或衰减系数	1/s
$T_{\max}$	最大时间阈值	s
N	样本总数或迭代次数	-
R^2	拟合优度或决定系数	-
ε	误差项或极小量	-
∇f	函数f的梯度	-
$\sum_{i=1}^{n}$	从1到n的求和运算	-

5 模型的建立与求解

ps：这部分因人而异

5.1 问题一的求解

5.1.1 建模思路/解题步骤

-
  -
-
-

5.1.2 运算方程/运算方法

-
  -
-

（示例）

等距螺线的极坐标方程为

$$r(\theta)=a+b\theta \tag{1}$$

其中，r为极径；θ为极角；a和b均为实数，由题意可知$a=0$。

螺距p的大小可表示为

$$p=r(\theta+2\pi)-r(\theta)=b\cdot 2\pi$$

结合以上分析，得到

$$r(\theta)=\frac{p}{2\pi}\theta \tag{2}$$

将极坐标转换为直角坐标

$$\begin{cases} x=r(\theta)\cdot\cos\theta \\ y=r(\theta)\cdot\sin\theta \end{cases} \tag{3}$$

这样，就可以计算出舞龙队在平面上任一角度θ下的具体位置。

5.1.3 继续求解步骤（数据表、图）

-
  -
-
-

表格 1. ***变化情况

	0s	60s	120s	180s	240s	300s
龙头 (m/s)	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
第一节龙身 (m/s)						
第二节龙身 (m/s)						
第三节龙身 (m/s)						
第四节龙身 (m/s)						
龙尾 (m/s)						

5.1.4 继续求解步骤（数据表、图）

-
-
-

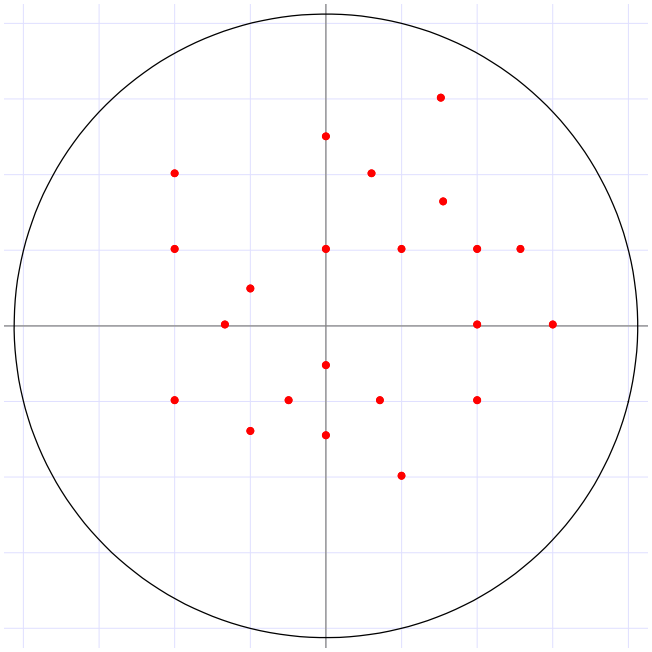

图 2. ***位置变化图

5.2 问题二的求解

5.2.1 建模思路/解题步骤

-
-
-

5.2.2 运算方程/运算方法

-
-
-

5.2.3 继续求解步骤（数据表、图）

-
-
-

5.2.4 继续求解步骤（数据表、图）

-
-
-

5.3 问题三的求解

5.3.1 建模思路/解题步骤

-
-
-

5.3.2 运算方程/运算方法

-
-
-

5.3.3 继续求解步骤（数据表、图）

-
-
-

5.3.4 继续求解步骤（数据表、图）

-
-
-

5.4 问题四的求解

5.4.1 建模思路/解题步骤

-
-
-

5.4.2 运算方程/运算方法

-
-
-

5.4.3 继续求解步骤（数据表、图）

-
-
-

5.4.4 继续求解步骤（数据表、图）

-
-
-

6 模型的评价

6.1 模型的优点

1.

2.

3.

6.2 模型的缺点

1.

2.

3.

6.3 模型的改进

1.

2.

3.

7 参考文献

[1] Alex Krizhevsky, Ilya Sutskever, 与 Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, 与 K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, 卷 25, 页面 0. Curran Associates, Inc., 2012.

附录A　支撑材料文件列表

文件名	说明
***-1.py	问题一到问题三的***
***-2.py	…
***-3.py	…
***-4.py	…
***1-1.py	…
***1-2.py	…
***1-3.py	…
***1-4.py	…
***2-1.py	…
***2-2py	…
***2-3.py	…

附录B　支撑材料的所有Python代码

```
"""
文件名: data_processing-1.py
用途: 2023全国大学生数学建模竞赛C题-蔬菜运输优化
      数据预处理模块（数据清洗+特征计算）
"""

import numpy as np
import pandas as pd
from scipy.optimize import linprog
import matplotlib.pyplot as plt

# ==================== 数据预处理函数 ====================
def load_and_clean_data(file_path):
    """
    数据加载与清洗
    参数:
        file_path : str - CSV文件路径
    返回:
        df : DataFrame - 处理后的干净数据
    """
    try:
        df = pd.read_csv(file_path, encoding='gbk')  # 处理中文编码
        df.dropna(inplace=True)  # 删除缺失值
        df = df[df['产量'] > 0]  # 过滤无效产量记录
        return df
    except Exception as e:
        print(f"数据加载失败:␣{str(e)}")
        return None

# ==================== 优化模型求解 ====================
def transport_optimization(cost_matrix, supply, demand):
    """
```

```
    运输问题线性规划求解
    参数:
        cost_matrix : ndarray - 运输成本矩阵 (m×n)
        supply : ndarray - 供应量数组 (m,)
        demand : ndarray - 需求量数组 (n,)
    返回:
        result : dict - 包含优化结果的字典
    """
    # 线性规划求解 (使用单纯形法)
    res = linprog(cost_matrix.flatten(),
                  A_eq=_build_constraints(supply, demand),
                  b_eq=_build_boundary(supply, demand),
                  method='highs')

    return {
        'status': res.status,
        'total_cost': res.fun,
        'schedule': res.x.reshape(cost_matrix.shape)
    }

# ==================== 可视化模块 ====================
def plot_solution(routes, nodes):
    """绘制运输路线图"""
    plt.figure(figsize=(10, 8))
    for (i,j), val in np.ndenumerate(routes):
        if val > 0:
            plt.plot([nodes[i][0], nodes[j][0]],
                     [nodes[i][1], nodes[j][1]],
                     'b-', alpha=0.5, linewidth=val*2)
    plt.scatter(nodes[:,0], nodes[:,1], c='r', s=50)
    plt.title("Optimal Transport Routes")
    plt.xlabel("X Coordinate")
    plt.ylabel("Y Coordinate")
    plt.grid(True)
    plt.savefig('routes.png', dpi=300)

if __name__ == '__main__':
    # 示例数据
    demo_cost = np.random.rand(5,3) * 100
    demo_supply = np.array([20, 30, 15, 25, 10])
    demo_demand = np.array([40, 30, 20])

    # 执行优化
    solution = transport_optimization(demo_cost, demo_supply, demo_demand)
    print(f"最优总成本: {solution['total_cost']:.2f} 元")
```