

AI 工具使用详情

1. 所用 AI 工具名称和版本

在完成 C 题的过程中，我们使用了来自深度求索公司的 DeepSeek,其版本为 DeepSeek-R1-0528，以及来自北京春田知韵科技有限公司（字节跳动旗下）的豆包，其版本为 Doubao-Seed-1.6。

2. 具体使用目的和环节

在此题建模过程中，AI 工具仅用于非核心辅助。我们希望由 AI 来进行某些非思考工作量的完成，具体表现为：辅助减少重复运算、编写简单代码，在论文中进行文字公式润色；而模型核心逻辑与关键参数，则由我们独立完成。

2.1 辅助减少重复运算

在“数据清洗环节”，我们独立确定“Y 染色体浓度用 3σ 法则剔除异常值”“BMI 剔除 < 20 或 > 45 极端值”的核心规则，使用 AI 计算 Y 浓度均值 $\mu=0.052$ 、标准差 $\sigma=0.018$ 及 BMI 均值 $\mu=31.2$ 、标准差 $\sigma=4.5$ ，避免重复运算；同时针对“孕周‘周 + 天’转连续值”的规则，AI 辅助生成 pandas 批量转换代码，替代数百条样本的手动转换。

在第一问的“MARS 模型残差分析”中，AI 辅助计算残差标准差 s 、 $(X^T X)$ 逆矩阵对角元等复杂统计量，替代手动推导计算，减少计算量。第三问的“每组达标时间统计”，AI 辅助批量计算各组均值、标准差。

在误差分析环节，AI 辅助执行重复抽样计算：在第三问的“测量误差模拟”，AI 辅助代入正态分布参数，批量输出风险值与达标比例样本，替代手动逐条抽样。

2.2 编写简单代码

在可视化基础代码环节，AI 辅助生成简单绘图代码，但 AI 仅提供绘图函数调用框架：第一问中“MARS 模型残差散点图”的代码，第三问中“每组达标时间直方图”的代码。

在算法基础框架代码环节，AI 辅助生成简单代码块，但 AI 仅生成通用框架：如第二问中“LOESS 拟合局部加权函数”的基础循环代码，第三问中“模

拟退火 Metropolis 准则” 的 if-else 接受概率代码。

2.3 论文文字公式润色

在模型公式排版润色环节，我们独立设计了公式核心形式，AI 辅助优化 LaTeX 格式；

3. 关键交互记录（重要提示词与回复）

3.1 “孕周批量转换” 运算辅助

提示词：在数据预处理中，需将‘12+3’‘15+5’格式的孕周字符串转为连续值（如 $12+3/7 \approx 12.43$ 周），基于 pandas 实现。

回复：

```
1 # 函数：将孕周转换为连续数值，例如 '12w+3' -> 12 + 3/7 ≈ 12.43
2 def convert_gest_week(week_str):
3     if pd.isna(week_str):
4         return np.nan
5     # 预处理：去除空格并转为小写，增强兼容性
6     week_str = str(week_str).replace(' ', '').lower()
7     # 正则匹配：支持 '12+3' 和 '15w+5' 等格式
8     match = re.match(r'(\d+)w?(?:(\d+))?', week_str)
9     if match:
10         weeks = int(match.group(1)) # 提取周数部分
11         # 提取天数部分（如果存在则转换为整数，否则默认为0）
12         days = int(match.group(2)) if match.group(2) else 0
13         # 计算连续值：周数 + 天数/7，保留两位小数
14         return round(weeks + days / 7, 2)
15     # 若格式不匹配，返回缺失值
16     return np.nan
17
18 # 应用转换到检测孕周列
19 df_male['检测孕周'] = df_male['检测孕周'].apply(convert_gest_week)
20 # 女胎数据同样应用该转换
21 df_female['检测孕周'] = df_female['检测孕周'].apply(convert_gest_week)
22 |
```

AI 生成

3.2Y 染色体浓度用 3σ 法则剔除异常值，BMI 剔除 < 20 或 > 45 极端值

提示词：在数据预处理中，需将 Y 染色体浓度用 3σ 法则剔除异常值，BMI 剔除 < 20 或 > 45 极端值；

回复：

```
1 # 计算残差标准差s
2 residuals = y - y_pred_final
3 s = np.std(residuals, ddof=1) # 使用样本标准差（自由度为n-1）
4 print(f"\nMARS模型残差标准差 s = {s:.6f}")
5
```

AI 生成

```

1  # -----
2  # Y染色体浓度异常值处理 (3σ法则)
3  # -----
4  # 计算Y染色体浓度的均值(mu)和标准差(sigma)
5  mu = df_male['Y染色体浓度'].mean()
6  sigma = df_male['Y染色体浓度'].std()
7
8  # 应用3σ法则: 只保留在[mu-3σ, mu+3σ]范围内的数据
9  # 即剔除偏离均值超过3个标准差的极端值
10 df_male_zscore = df_male[(df_male['Y染色体浓度'] >= mu - 3 * sigma) &
11                           (df_male['Y染色体浓度'] <= mu + 3 * sigma)]
12
13 # -----
14 # BMI异常值处理 (固定阈值法)
15 # -----
16 # 剔除BMI < 20 或 > 45的极端值 (根据医学常识设定的合理范围)
17 # 注意: 原代码中使用的是IQR法, 这里修改为固定阈值法以符合需求
18 df_male_bmi_filtered = df_male[(df_male['孕妇BMI'] >= 20) &
19                                 (df_male['孕妇BMI'] <= 45)]
20
21 # -----
22 # 综合应用到最终清洗数据
23 # -----
24 # 结合缺失值删除法、Y染色体浓度3σ法则和BMI固定阈值过滤
25 df_male_clean = df_male_dropna.copy()
26 df_male_clean = df_male_clean[(df_male_clean['Y染色体浓度'] >= mu - 3 * sigma) &
27                                (df_male_clean['Y染色体浓度'] <= mu + 3 * sigma)]
28 df_male_clean = df_male_clean[(df_male_clean['孕妇BMI'] >= 20) &
29                                (df_male_clean['孕妇BMI'] <= 45)]
30
31 # -----
32 # 女胎BMI同样处理
33 # -----
34 # 女胎数据也应用相同的BMI阈值过滤
35 df_female = df_female[(df_female['孕妇BMI'] >= 20) &
36                        (df_female['孕妇BMI'] <= 45)]
37

```

3.3 “MARS 模型残差统计量” 计算辅助

提示词: MARS 模型显著性检验中, 需计算残差标准差 s 和 $(X^T X)$ 逆矩阵的对角元, 用于 t 检验。请基于 `statsmodels` 和 `numpy` 生成计算代码, 输入为模型拟合结果 (`linear_model`) 和特征矩阵 (`X`)。

回复:

```

1  # 计算类似(X^T X)逆矩阵对角元的特征影响度指标
2  # 对于MARS模型，使用特征置换后的预测方差作为替代指标
3  print("\n计算MARS模型特征影响度指标（类似(X^T X)^-1对角元）...")
4  n_repeats = 10
5  influence = {col: 0.0 for col in X.columns}
6
7  for col in X.columns:
8      # 存储原始特征值
9      original = X[col].copy()
10
11     # 多次置换计算预测方差
12     pred_vars = []
13     for _ in range(n_repeats):
14         # 置换特征
15         X_perm = X.copy()
16         X_perm[col] = np.random.permutation(X_perm[col])
17
18         # 计算置换后的预测值
19         y_pred_perm = final_model.predict(X_perm)
20         pred_vars.append(np.var(y_pred_perm))
21
22     # 取平均预测方差作为影响度指标
23     influence[col] = np.mean(pred_vars)
24
25     # 恢复原始特征
26     X[col] = original
27
28 # 输出结果
29 print("\nMARS模型特征影响度指标（类似(X^T X)^-1对角元）:")
30 for col, val in influence.items():
31     print(f"    {col}: {val:.6f}")
32

```

AI生成

3.4 AI 替代手动逐条抽样。

提示词：第三问的“测量误差模拟”，写出可以代入正态分布参数，批量输出风险值与达标比例样本，替代手动逐条抽样的 python 代码。

回答：

```

1  def monte_carlo_group_robustness(group_rows, pipeline, features, t_candidate, n_sim=1000):
2      # (省略部分代码：估计测量误差sigma)
3
4      # 基线预测值
5      base_preds = []
6      for _, row in group_rows.iterrows():
7          # (省略部分代码：构造特征向量)
8          base_preds.append(pipeline.predict(arr)[0])
9      base_preds = np.array(base_preds).flatten()
10
11     rng = np.random.default_rng(0)
12     risk_samps = []
13     reach_props = []
14     for _ in range(n_sim):
15         # 代入正态分布参数（均值0，标准差sigma）生成模拟扰动
16         sim = base_preds + rng.normal(0, sigma, size=base_preds.shape)
17         # 计算达标比例
18         reach = np.mean(sim >= THRESH)
19         # 计算风险值
20         risk = (1 - reach) + 0.05 * (t_candidate - T_MIN)
21         # 批量存储结果
22         risk_samps.append(risk)
23         reach_props.append(reach)
24     risk_samps = np.array(risk_samps)
25     # (省略返回部分)
26     return {'cv': cv, 'risk_samples': risk_samps, 'reach_samples': np.array(reach_props), 'sigma_est': sigma}
27

```

AI生成

3.5 生成 MARS 模型残差散点图

提示词：就一探讨模型，给出生成 MARS 模型残差散点图的代码；

回复：

```
1 # 可视化残差
2 residuals = y - y_pred_final
3 plt.figure(figsize=(10, 6))
4 plt.scatter(y_pred_final, residuals, alpha=0.5)
5 plt.axhline(0, color='r', linestyle='--')
6 plt.xlabel('预测值')
7 plt.ylabel('残差')
8 plt.title('MARS模型残差图')
9 plt.grid(True)
10 plt.show()
11
```

AI 生成

3.6 生成每组达标时间直方图

提示词：就探讨模型，给出生成每组达标时间直方图的代码；

回复：

```
1 # 绘图：组内达标时间直方图 + SA 时点
2 L, U = map(float, info['bmi_range'].strip('[]()').split(','))
3 rows = df[(df['BMI_original'] >= L) & (df['BMI_original'] < U)]
4 plt.figure(figsize=(6,4))
5 sns.histplot(rows['Target_Time'].dropna(), bins=15)
6 plt.axvline(info['t_star'], color='r', linestyle='--', label=f"t*={info['t_star']:.2f}")
7 plt.title(f"组 {gi} (BMI {info['bmi_range']}) 达标时间分布")
8 plt.xlabel('达标孕周')
9 plt.legend()
10 plt.tight_layout()
11 plt.savefig(os.path.join(OUT_DIR, f'group_{gi}_target_time_hist.png'))
12 plt.close()
13
```

AI 生成

3.7 生成“模拟退火 Metropolis 准则”的 if-else 接受概率代码

提示词：就探讨模型，给出生成“模拟退火 Metropolis 准则”的 if-else 接受概率代码；

回复：

```
1 cand = float(np.clip(t_curr + rng.normal(0,0.5), T_MIN, T_MAX))
2 r_c = risk(cand)
3 d = r_c - r_curr
4 if d < 0 or rng.random() < math.exp(-d / max(T,1e-12)):
5     t_curr, r_curr = cand, r_c
6     traj.append(r_curr)
7     improved = improved or (d < 0)
8     if r_curr < r_best:
9         t_best, r_best = t_curr, r_curr
10
```

AI 生成

3.8 生成“LOESS 拟合局部加权函数”的基础循环代码

提示词：就探讨模型，给出“LOESS 拟合局部加权函数”的基础循环代码；

回复:

```
1 for i in range(n):
2     dist = np.abs(weeks - weeks.iloc[i])
3     max_dist = bandwidth * np.max(dist) if np.max(dist) > 0 else 1e-10
4     weights = np.exp(-(dist / max_dist) ** 3)
5     fitted[i] = np.sum(weights * concentrations) / np.sum(weights) if np.sum(weights) > 0 else 0
6
```

AI 生成

4. 采纳和人工修改情况

对于 Y 染色体浓度用 3σ 法则剔除异常值, BMI 剔除 <20 或 >45 极端值和针对“孕周‘周 + 天’转连续值”的转化, 我们采纳了 AI 所给出来的代码和结果, 进行了相关的数据清洗;

对于 MARS 模型残差统计量计算, 我们采纳了采纳 AI 生成的“特征置换法计算影响度指标”逻辑, 包括“多次置换特征→计算预测方差→取均值作为影响度”, 替代手动推导 $(X^T X)$ 逆矩阵的复杂运算, 简化 MARS 模型特征重要性的量化过程, 但是我们将 AI 计算的“特征影响度指标”与 t 检验结合, 人工推导相关方程, 完成变量显著性验证。

对于 AI 替代手动逐条抽样, 我们采纳 AI 生成的 monte_carlo_group_robustness 函数核心框架, 包括“正态噪声生成 (rng.normal(0, sigma))”、“达标比例计算 (np.mean(sim >= THRESH))”及“风险值批量存储”, 替代手动抽样的重复操作, 提升误差分析效率。

对于生成 MARS 模型残差散点图的代码, 我们采纳 AI 生成的基础绘图逻辑, 包括“残差计算 (residuals = y - y_pred_final)”、“散点图绘制 (plt.scatter)”、“参考线添加 (plt.axhline(0, color='r', linestyle='--'))”及“坐标轴标签设置”, 确保残差可视化的基础格式规范, 同时通过调整散点透明度 (alpha=0.5→alpha=0.6)、添加网格线 (plt.grid(True, linestyle='--', alpha=0.7))、调整图尺寸 (figsize=(10,6)→figsize=(12,6)), 提升图表可读性。

对于生成每组达标时间直方图的代码, 我们采纳 AI 生成的“BMI 区间筛选” (rows = df[(df['BMI_original'] >= L) & (df['BMI_original'] < U)])、“直方图绘制 (sns.histplot)”及“最优时点标注 (plt.axvline)”, 确保每组达标时间分布的基础可视化效果, 但在运行中有一些报错, 进行了语法修正。

对于生成模拟退火 Metropolis 准则的代码, 我们采纳 AI 生成的“候选时点生成 (np.clip(t_curr + rng.normal(0, 0.5), T_MIN, T_MAX))”、“风险差计算 (d

`=r_c - r_curr)`”及“接受概率判断 (`d < 0 or rng.random() < math.exp(-d/max(T,1e-12))`)”，确保 Metropolis 准则的基础逻辑正确，但同时我们基于建模需要，人工设定关键参数初始值，AI 未提供该核心参数。

对于生成 LOESS 拟合局部加权函数的代码，我们采纳 AI 生成的“局部权重计算逻辑”（基于距离的指数权重函数）、“循环遍历每个时点拟合”及“加权均值计算 (`np.sum(weights * concentrations) / np.sum(weights)`)”，避免手动推导局部加权公式的重复运算，并未进行过多修改。

对于模型公式 LaTeX 格式优化，人工审阅了相关数学式，保证公式数学准确性。