人口土壤的嬗变：高等教育根系的深扎与新枝的萌发

——基于人口变化研究高等教育发展趋势及决策

2025 年 8 月 25 日

摘要

当人口的潮汐在时代版图上重新勾勒轮廓，生育率的波动、老龄化的推进、城乡流动的加速，正使教育赖以生长的“人口土壤”发生着深刻嬗变——曾经丰沃的适龄生源地渐显肌理变化，多元需求的根系在社会土壤中悄然延伸，而高等教育这棵历经数十载生长的大树，其深扎于传统模式的根系正遭遇新的水土，亟待在变革中寻找深扎的支点，其伸向未来的枝桠也需在挑战中萌发新的生长方向。在此背景下，追问高等教育在人口变迁中的生存与生长之道，不仅是对教育规律的再思考，更是对一个时代如何通过教育延续生机、培育新可能的回应。

问题一中，我们将中国各年龄段人口分为 0-4、5-9、10-14......50-54、老年组共十二组，搜集并处理了 2010 至今各年龄段的生育率和存活率数据，得到了未来二十年各年龄段生育率和存活率的预测值，运用了 **Leslie 矩阵模型**通过反复迭代预测了未来五年，十年，二十年后中国人口的变化情况。另外，我们广泛查阅了各国鼓励生育政策，总结了为促进人口持续增长的几点措施。

问题二中，我们将高等教育量化为**生均高等教育经费**、**高校毕业生总人数（包括本专科生、硕士、博士生）**、**高等教育毛入学率**这三个指标，引入**斯皮尔曼相关系数**定量分析了 15-19、20-24、25-29 这三个年龄段人口的变化对这四个指标的影响效果，并将结果绘制成表格。

问题三中，我们将学历贬值这一现象量化为**搜索热度**、**教育回报率**这两个指标，其中教育回报率一定程度上借鉴了**明瑟方程**的结论，搜集数据并做处理使其**归一化**。结合人口变化这一前提，定义了**学历竞争指数**、**学历危险指数**这两个指标并做出具体解释。构建线性关系并通过网格搜索，以 0:0.1:1 的步长找到最优解，使得到的误差（**欧几里得距离**）达到最小，根据实验结果作出了以学历贬值为话题的结论。

关键词　人口　高等教育　Leslie 矩阵模型　学历贬值　明瑟方程　教育回报率

目录

1 问题重述

1.1 背景

近年来，中国人口出生率持续走低，引发了社会各界对教育、经济、社会结构等多方面的深度思考。根据国家统计局数据，2023 年中国出生人口降至 902 万人。但 2024 年中国内地（不含港澳台）新出生人口大幅反弹，从 2023 年的 902 万人增长到了 954 万人！事实上，人口变化对高等教育也有重要影响，随着 1981 年《学位条例》的正式实施，中国开始确立了学士、硕士、博士的学位制度。

1.2 题目要求

为了确定人口变化对中国高等教育的影响，请解决以下问题：

1.2.1 问题一

请利用中国统计年鉴 2024 中关于人口的数据，建立数学模型，预测未来 5 年，10 年，20 年后中国人口的变化情况，如果要保持人口持续增长，请给出可采取的措施或建议。

1.2.2 问题二

请根据人口的发展变化，查找互联网有关数据，建立数学模型，定量分析人口变化对未来高等教育的影响。

1.2.3 问题三

请查找互联网数据，结合人口变化情况，建立数学模型定量分析学历贬值问题。

2 问题分析

2.1 问题一

Leslie 矩阵模型由英国数学家 P.H.Leslie 于 1945 年提出，用于模拟按年龄分组的人口（或种群）随时间演化的一种数学模型。它是人口动态建模的经典工具，广泛应用于生物种群生态学与人类人口预测中。[1]

Leslie 矩阵模型由两个主要部分组成：

- **人口向量（Population vector）：**

$$n(t) = \begin{bmatrix} n_1(t) \\ n_2(t) \\ \vdots \\ n_k(t) \end{bmatrix}$$

  表示在时间 t 时，不同年龄组的人口数量。

- **Leslie 矩阵（L）：**是一个 $k \times k$ 的方阵，结构如下：

$$L = \begin{bmatrix} f_1 & f_2 & f_3 & \cdots & f_k \\ s_1 & 0 & 0 & \cdots & 0 \\ 0 & s_2 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & s_{k-1} & 0 \end{bmatrix}$$

  其中：

  - f_i：第 i 个年龄组的生育率（fertility rate）；
  - s_i：第 i 个年龄组的存活率（survival rate），即从第 i 组存活到第 $i+1$ 组的比例。

模型演化公式

模型的核心更新公式为：

$$n(t+1) = L \cdot n(t)$$

即：

- 每年通过乘以 Leslie 矩阵 L，即可得到下一期（$t+1$）各年龄段的人口数量；

- 不断迭代可以模拟未来若干年的种群结构变化。

为得到求解 Leslie 矩阵模型所需要的初值条件，我们首先将中国的人口进行年龄分组，按 0-4、5-9、10-14......50-54、老年组这一顺序共分成十二组；并搜集得到 2010-2023 年各年龄段的生育率和存活率数据[2]，拟合并预测未来二十年各年龄段的生育率和存活率。然后以 2023 年的年龄结构作为起点开始迭代[3]。

我们记 $F_i(1 \le i \le 12)$ 为第 i 个年龄组的生育率随时间变化曲线；记 $S_i(1 \le i \le 12)$ 为第 i 个年龄组的存活率随时间变化曲线; 从而有：记 $f_i(t) \quad (1 \le i \le 12, 2024 \le t \le 2045)$ 为 t 时刻第 i 个年龄组的生育率 (可以将 0-14、50 及以上的年龄段的生育率近似视作 0，即 $f_1(t) = f_2(t) = f_3(t) = f_{11}(t) = f_{12}(t) = 0$)；记 $s_i(t) \quad (1 \le i \le 11, 2024 \le t \le 2045)$ 为 t 时刻第 i 个年龄组的存活率；记 $\boldsymbol{L}_(t) \quad (2024 \le t \le 2045)$ 为 t 时刻的 Leslie 矩阵；记 $\boldsymbol{n}_(t) \quad (2024 \le t \le 2045)$ 为 t 时刻的人口向量；记 $p_(t) \quad (2024 \le t \le 2045)$ 为 t 时刻的人口总数。

根据 Leslie 矩阵模型，可以得到：

t 时刻的 Leslie 矩阵为：

$$\boldsymbol{L}(t) = \begin{bmatrix} f_1(t) & f_2(t) & f_3(t) & \cdots & f_{12}(t) \\ s_1(t) & 0 & 0 & \cdots & 0 \\ 0 & s_2(t) & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & s_{11}(t) & 0 \end{bmatrix} \tag{1}$$

t+1 时刻的人口向量为：

$$\boldsymbol{n}(t+1) = \boldsymbol{L}(t) \cdot \boldsymbol{n}(t) \tag{2}$$

t 时刻的总人口即为：

$$p(t) = \sum_{i=1}^{12} n_i(t) = \mathbf{1}_{1\times 12} \cdot \boldsymbol{n}(t) \tag{3}$$

其中：$\mathbf{1}_{1\times 12}$ 表示 1×12 的全 1 行向量（即 $\begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}$）

为给出促进人口持续增长的几点措施，我们翻阅了《生育支持政策国际启示研究》（何樾、张翊廷，2024）、《中国的城镇化如何影响生育率？——基于空间面板数据模型的研究》（戈艳霞，2024）、《政府生育保险有效支出对生育意愿的影响效应》（金双华、于征莆、孟令雨，2024）等文献，总结文献结论，汲取作者思想精髓，最终给出了参考措施和主要依据。

2.2 问题二

问题二的关键是对"对高等教育的影响"这一抽象概念进行量化，经过多方面综合考量及查证，我们将这一概念量化为"生均高等教育经费[4]、高校毕业生总人数（包括本专科生、硕士、博士生）[5]、高等教育毛入学率[6]"这三个指标，通过搜集相关数据，得到不同时刻这三个指标的具体数值。为定量地分析人口变化对这三个指标的分别影响，我们需要明确"人口变化"这一对象具体所指，基于实际情况，我们重点考虑了 2010-2023 年 15-19、20-24、25-29 这三个年龄段的人口数量变化；最后，我们引入了斯皮尔曼相关系数[7] 加以解决这一问题。

我们记 $\boldsymbol{n_i'}$ $(1 \leq i \leq 3)$ 为上述三个年龄段中第 i 个年龄段按时间先后顺序排列形成的人口列向量；进一步，记 E_j $(1 \leq j \leq 3)$ 分别表示与高等教育相关的三个指标按时间先后顺序排列形成的数值列向量，具体如下：

$E_1(t)$：生均高等教育经费（单位：元）

$E_2(t)$：高校毕业生总人数（单位：万元）

$E_3(t)$：高等教育毛入学率（单位：%）

为定量衡量第 i 个年龄段人口数量 $\boldsymbol{n_i'}$ 对第 j 个高等教育相关指标 $\boldsymbol{E_j}$ 的影响强度，我们采用斯皮尔曼等级相关系数（Spearman's Rank Correlation Coefficient）作为分析工具。

变量秩变换与斯皮尔曼系数计算

设 $\boldsymbol{n_i'} = [n'_{i,1}, n'_{i,2}, \ldots, n'_{i,T}]^\top$，$\boldsymbol{E_j} = [E_{j,1}, E_{j,2}, \ldots, E_{j,T}]^\top$ 表示按时间顺序排列的两个长度为 T 的数值列向量，其中 $T = 14$ 为样本年份数量。

将两个向量进行秩排序，记其秩向量分别为：

$$\boldsymbol{R}^{(i)} = [R_1^{(i)}, R_2^{(i)}, \ldots, R_T^{(i)}]^\top, \quad \boldsymbol{S}^{(j)} = [S_1^{(j)}, S_2^{(j)}, \ldots, S_T^{(j)}]^\top$$

定义秩差为：

$$d_k^{(i,j)} = R_k^{(i)} - S_k^{(j)}, \quad (k = 1, 2, \ldots, T)$$

若无重复秩，斯皮尔曼系数计算公式为：

$$\rho_{ij} = 1 - \frac{6\sum_{k=1}^{T}\left(d_k^{(i,j)}\right)^2}{T(T^2-1)}$$

若存在重复秩，则改用皮尔逊相关系数形式计算：

$$\rho_{ij} = \frac{\sum_{k=1}^{T}\left(R_k^{(i)} - \bar{R}^{(i)}\right)\left(S_k^{(j)} - \bar{S}^{(j)}\right)}{\sqrt{\sum_{k=1}^{T}\left(R_k^{(i)} - \bar{R}^{(i)}\right)^2} \cdot \sqrt{\sum_{k=1}^{T}\left(S_k^{(j)} - \bar{S}^{(j)}\right)^2}}$$

其中:

$$\bar{R}^{(i)} = \frac{1}{T}\sum_{k=1}^{T} R_k^{(i)}, \quad \bar{S}^{(j)} = \frac{1}{T}\sum_{k=1}^{T} S_k^{(j)}$$

显著性检验与判断标准

为判断变量之间的相关性是否具有统计显著性，我们进一步对 ρ_{ij} 进行显著性检验，其原假设为 H_0: 两变量无相关性，即 $\rho_{ij} = 0$。

在斯皮尔曼系数的检验中，p 值 p_{ij} 可由样本中秩序列计算得到，表示在原假设成立的前提下，观察到至少该程度相关性的概率大小。

我们采用以下显著性标准进行判断：

- 若 $p_{ij} < 0.001$，则认为 ρ_{ij} 极显著，记作 ***；
- 若 $p_{ij} < 0.05$，则认为 ρ_{ij} 显著，记作 **；
- 若 $0.05 \leq p_{ij} < 0.1$，认为边际显著，记作 *；
- 若 $p_{ij} \geq 0.1$，则认为 ρ_{ij} 不显著，记作 $^{\mathrm{ns}}$。

值域与含义

斯皮尔曼系数 ρ_{ij} 的值域为 $[-1, 1]$，其经济含义如下：

- 若 $\rho_{ij} > 0$，说明该年龄段人口与第 j 项高等教育指标呈正向单调关系；
- 若 $\rho_{ij} < 0$，说明二者呈负向单调关系；
- 若 $\rho_{ij} \approx 0$，则说明二者间无显著单调关系。

可视化与结果解释

将所有计算得到的 ρ_{ij} 构成 3×3 的相关系数矩阵，并使用热力图展示，能够清晰呈现各年龄段人口与各教育指标之间相关程度与趋势，为下一步建模与决策提供定量依据。

2.3 问题三

问题三我们首先要将“学历贬值”这一现象级名词量化，记 $\boldsymbol{V}_t$ $(0 \leq \boldsymbol{V}_t \leq 1)$ 为第 t 年学历贬值热度。

我们将学历贬值量化为仅由两个因素控制：第 t 年学历贬值在国内各平台搜索热度 $h(t)$（$0 \le h(t) \le 1$）以及第 t 年教育回报率 $r(t)$（$0 < r(t) \le 1$）。我们假设搜索热度与学历贬值热度线性相关，而教育回报率一般随着学历贬值加剧而降低，因此成反比。

其中搜索热度 $\boldsymbol{h}$ 为统计历年"学历提升"关键词搜索频次归一化处理后所得，满足：

$$\boldsymbol{h} \in [0,1]^T \tag{4}$$

由于教育回报率 $r(t)$ 通常处于 $[0.05, 0.1]$，若直接与 $h(t)$ 做线性组合会造成尺度不匹配。我们取其倒数并记为向量：

$$\boldsymbol{x} = \frac{1}{\boldsymbol{r}} \tag{5}$$

对其进一步归一化，得到向量 $\boldsymbol{x'}$：

$$\boldsymbol{x'} = \frac{\boldsymbol{x} - x_{\min}}{x_{\max} - x_{\min}} \in [0,1]^T \tag{6}$$

最终模型表达为：

$$\boldsymbol{V} = \alpha_1 \cdot \boldsymbol{h} + \alpha_2 \cdot \boldsymbol{x'}, \qquad \alpha_1 + \alpha_2 = 1 \tag{7}$$

其中：

- $\boldsymbol{h}$ 表示"学历提升"关键词在各年份的搜索热度（已归一化）；
- $\boldsymbol{r}$ 表示教育回报率向量（百分比已除以 100）；
- $\boldsymbol{x}$ 为其倒数序列，$\boldsymbol{x'}$ 为归一化结果；
- $\boldsymbol{V}$ 为学历贬值热度向量；
- α_1、α_2 为加权系数。

教育回报率 $r(t)$ 满足明瑟方程（Mincer Equation）[8]，其经典形式如下：

$$\ln(w_i(t)) = \beta_0(t) + \beta_1(t) \cdot S_i + \beta_2(t) \cdot X_i + \beta_3(t) \cdot X_i^2 + \varepsilon_i \tag{8}$$

其中：

- $w_i(t)$ 表示第 i 个体在时间 t 的工资；
- S_i 为受教育年限；
- X_i 为经验，$X_i =$ 年龄 $- S_i - 6$；
- $\beta_1(t)$ 即为第 t 年的边际教育回报率，即 $r(t)$。

回报率时间序列记为列向量：

$$\boldsymbol{r} = [r(1), r(2), \ldots, r(T)]^\top \quad \Rightarrow \quad \boldsymbol{x} = \frac{1}{\boldsymbol{r}}, \quad \boldsymbol{x'} = \text{归一化}(\boldsymbol{x})$$

结合题目“结合人口变化的情况”，我们重点关注学历贬值在本科阶段的体现。引入以下变量：

- $\boldsymbol{\gamma}_1 = [\gamma_1(1), \ldots, \gamma_1(T)]^\top$：本科就业率向量；
- $\boldsymbol{\gamma}_2 = [\gamma_2(1), \ldots, \gamma_2(T)]^\top$：本科升学率向量；
- $\boldsymbol{\omega}_1$：本科生人数向量；
- $\boldsymbol{\omega}_2$：应届毕业生总人数向量。

忽略其他去向，近似认为：

$$\boldsymbol{\gamma}_1 + \boldsymbol{\gamma}_2 = \mathbf{1} \tag{4}$$

定义学历危机指数与竞争指数为：

$$\boldsymbol{\sigma} = \frac{\boldsymbol{\gamma}_1}{\boldsymbol{\gamma}_2} \tag{5}$$

$$\boldsymbol{\theta} = \frac{\boldsymbol{\omega}_1}{\boldsymbol{\omega}_2} \tag{6}$$

经观察，学历贬值热度 $\boldsymbol{V}$ 与上述两个因素正相关：

$$\boldsymbol{V} \propto \boldsymbol{\sigma} \tag{7}$$

$$\boldsymbol{V} \propto \boldsymbol{\theta} \tag{8}$$

合并后可建模为：

$$\boldsymbol{V} = \beta_1 \cdot \boldsymbol{\sigma} + \beta_2 \cdot \boldsymbol{\theta} \tag{9}$$

另一方面，前面已建模：

$$\boldsymbol{V} = \alpha_1 \cdot \boldsymbol{h} + \alpha_2 \cdot \boldsymbol{x'} \tag{10}$$

综合模型如下：

$$\alpha_1 \cdot \boldsymbol{h} + \alpha_2 \cdot \boldsymbol{x'} = \beta_1 \cdot \boldsymbol{\sigma} + \beta_2 \cdot \boldsymbol{\theta} \tag{11}$$

—

我们可通过网格搜索法（如 α_1 从 0 到 1 以 0.1 步长），计算对应的 β_1，β_2，通过可视化手段（如误差折线图、热力图）选取最符合现实的数据拟合结果。

3 模型假设

1、假设所研究的人口区域是封闭的，不考虑人口的迁入与迁出；

2、假设预测期间人口所处环境不会发生巨大变化，不会因自然灾害、战争等原因使人口数量大幅下降；

3、假设在 Leslie 模型中年龄段 0-14、54 岁及以上的人群生育率为 0；

4、假设各年龄段的男女性别比例对生育率、存活率以及人口数量变化影响微小；

5、忽略 0-14、30 岁及以上这些年龄段对高等教育的影响；

6、假设高等教育的程度主要由高等教育经费、高校毕业生、高等教育毛入学率这三个因素作用；

7、我们近似地认为本科毕业生去向只有就业和升学两种；

4 符号约定

表 1: 符号表（续表可自动分页）

符号	含义
$n_i(t)$	第 i 个年龄组在时间 t 的人口数量
$\boldsymbol{n}(t)$	在时间 t 时的人口列向量
$\boldsymbol{L}(t)$	在时间 t 时的 Leslie 矩阵
$f_i(t)$	第 i 个年龄组在时间 t 的生育率
$s_i(t)$	第 i 个年龄组在时间 t 的存活率（从 i 到 $i+1$）
$p(t)$	时间 t 的总人口数量
$\mathbf{1}_{1\times 12}$	1 行 12 列的全 1 向量
$\boldsymbol{n}_i'$	第 i 个年龄段在不同年份中的人口列向量（问题二）
$E_j(t)$	与高等教育相关的第 j 项指标在时间 t 的数值
$\boldsymbol{E}_j$	第 j 项教育指标随时间变化的列向量
$R_k^{(i)}$	第 i 个年龄段人口在第 k 年的秩次
$S_k^{(j)}$	第 j 个教育指标在第 k 年的秩次
$d_k^{(i,j)}$	第 k 年中两个变量的秩差
ρ_{ij}	年龄段 i 与教育指标 j 的斯皮尔曼相关系数
p_{ij}	对 ρ_{ij} 的显著性检验 p 值
T	时间序列长度，本文取 $T=14$
$^{***}, ^{**}, ^{*}, ^{\mathrm{ns}}$	分别表示显著性水平 $p<0.001$，0.05，0.1，≥ 0.1
$\bar{R}^{(i)}$	第 i 个变量秩次的均值
$\bar{S}^{(j)}$	第 j 个变量秩次的均值
$\boldsymbol{V}$	学历贬值热度向量，已归一化
$\boldsymbol{h}$	“学历提升”关键词搜索热度向量，已归一化
$\boldsymbol{r}$	教育回报率原始向量（百分比已除以 100）
$\boldsymbol{x}$	教育回报率倒数向量，$\boldsymbol{x}=1/\boldsymbol{r}$
$\boldsymbol{x}'$	对 $\boldsymbol{x}$ 的归一化结果，用于建模
α_1	搜索热度的权重系数（标量）
α_2	教育回报率倒数的权重系数，$\alpha_1+\alpha_2=1$
$\boldsymbol{\gamma}_1$	本科就业率向量
$\boldsymbol{\gamma}_2$	本科升学率向量

表 1 （续）

符号	含义
$\boldsymbol{\sigma}$	危机指数向量，γ_1/γ_2（逐元素）
$\boldsymbol{\omega}_1$	本科毕业人数向量
$\boldsymbol{\omega}_2$	应届毕业人数向量
$\boldsymbol{\theta}$	竞争指数向量，$\boldsymbol{\omega}_1/\boldsymbol{\omega}_2$
β_1	学历危机指数系数
β_2	学历竞争指数系数

5 模型建立及求解

5.1 问题一

利用编程工具 (python) 得到以下结果（代码见附录 10.1）:

各年龄段生育率和存活率如下表:

表 2: 2010–2023 年各年龄段女性生育率（单位：‰）

年份	15–19	20–24	25–29	30–34	35–39	40–44	45–49	50–54
2010	12.16	109.72	124.51	63.10	20.73	5.96	1.29	0.21
2011	11.80	106.01	123.81	64.36	21.25	6.10	1.42	0.21
2012	12.53	112.72	135.35	71.10	23.44	6.55	1.41	0.24
2013	11.98	104.90	128.75	69.32	23.77	6.53	1.66	0.30
2014	12.05	105.86	132.99	72.51	25.83	6.89	1.76	0.32
2015	10.89	95.61	124.45	70.31	26.40	6.97	1.95	0.29
2016	10.75	98.03	132.28	76.51	29.26	7.52	1.90	0.26
2017	10.24	96.70	133.70	80.17	30.95	7.70	1.82	0.23
2018	8.69	79.80	112.54	71.09	28.19	7.08	1.74	0.23
2019	8.28	76.02	109.21	69.83	27.70	7.08	2.01	0.23
2020	6.74	58.79	87.83	60.26	25.38	6.53	2.06	0.21
2021	6.00	52.47	78.50	54.60	23.76	6.21	1.93	0.19
2022	5.46	48.02	71.65	50.63	22.98	6.09	1.83	0.17
2023	5.23	46.53	69.27	48.58	22.24	5.84	1.68	0.15

表 3: 2010–2023 年各年龄段人口数（单位：千人）

年份	0–4	5–9	10–14	15–19	20–24	25–29	30–34	35–39	40–44	45–49	50–54	55–59	60–64	65–69	70–74	75–79	80–84	85–89	90–94	95–99	100+
2010	85,497	80,046	84,146	100,333	123,634	104,489	94,667	119,330	123,209	101,759	80,417	80,954	56,016	39,442	31,933	23,678	13,763	6,266	1,741	230	12
2011	86,745	79,910	81,601	94,241	124,020	107,451	94,702	113,172	112,568	118,677	75,484	83,926	56,460	42,651	32,344	24,473	14,293	6,525	1,875	251	14
2012	88,355	80,438	81,601	92,282	119,558	110,754	97,153	106,938	126,547	118,678	74,011	84,895	55,165	42,582	32,465	25,161	14,839	6,801	1,999	274	16
2013	89,773	81,943	80,717	96,808	113,806	114,616	99,981	100,728	126,276	119,137	79,751	85,239	59,981	45,129	33,245	25,592	15,487	7,126	2,133	296	17
2014	90,151	83,615	80,257	88,696	107,468	110,702	101,702	96,107	123,264	119,768	89,600	82,857	59,041	48,172	33,165	25,788	14,800	7,432	2,289	321	18
2015	90,753	85,190	79,803	83,612	99,771	122,893	103,833	94,024	118,283	121,673	99,924	78,224	77,509	52,002	34,914	26,022	16,859	7,742	2,447	352	21
2016	90,774	86,320	79,682	93,888	123,260	123,901	106,777	92,604	116,728	121,624	110,535	74,442	79,845	53,165	34,985	26,950	18,035	8,414	2,681	385	24
2017	90,716	88,078	80,418	100,985	130,505	118,834	110,083	96,529	106,041	125,040	116,534	72,074	81,332	60,672	37,896	27,417	18,986	8,994	2,861	416	26
2018	89,914	90,256	80,437	97,546	127,346	117,340	112,177	100,426	99,236	124,983	120,456	70,327	79,469	69,141	43,025	28,103	18,639	9,362	2,971	467	29
2019	85,366	90,258	83,419	90,982	119,982	105,763	113,230	101,054	95,328	121,863	117,860	87,351	73,469	69,141	43,025	28,103	18,639	9,362	2,971	467	29
2020	73,856	90,658	86,241	99,136	79,436	93,101	122,571	105,160	93,378	110,997	121,359	107,764	70,454	74,552	50,351	30,904	19,343	10,173	3,292	542	35
2021	65,606	88,621	87,909	87,300	80,633	89,704	118,128	109,431	95,820	104,902	122,961	113,509	69,177	75,963	54,531	31,737	19,798	10,443	3,394	554	37
2022	58,042	88,620	89,335	81,483	79,964	86,907	112,371	113,221	98,610	98,440	122,745	114,074	74,712	76,352	58,689	33,703	20,157	10,639	3,502	577	39

经过拟合和调试后，得到各年龄段的生育率和存活率预测值：

表 4: 各年龄段生育率拟合函数及拟合优度（2010–2023 年）

年龄段	**拟合类型**	**函数表达式** $y(t)$
15–19 岁	高斯函数	$9.419e^{-\frac{(t-1.977)^2}{2\cdot 6.295^2}}+2.905,\ R^2=0.987$
20–24 岁	高斯函数	$98.762e^{-\frac{(t-1.833)^2}{2\cdot(-7.365)^2}}+11.121,\ R^2=0.970$
25–29 岁	高斯函数	$121.168e^{-\frac{(t-3.592)^2}{2\cdot 7.122^2}}+13.592,\ R^2=0.944$
30–34 岁	三次函数	$-0.002t^3-0.439t^2+4.845t+61.464,\ R^2=0.884$
35–39 岁	三次函数	$-0.005t^3-0.089t^2+2.065t+19.788,\ R^2=0.847$
40–44 岁	三次函数	$-0.000t^3-0.028t^2+0.419t+5.799,\ R^2=0.848$
45–49 岁	三次函数	$-0.001t^3+0.002t^2+0.117t+1.276,\ R^2=0.933$

表 5: 各年龄段存活率拟合函数及拟合优度

年龄段	**拟合类型**	**函数表达式** $y(t)$
0–4 岁	高斯函数	$-0.015e^{-\frac{(t+8.030)^2}{2\cdot 8.365^2}}+0.994,\ R^2=0.998$
5–9 岁	高斯函数	$-0.002e^{-\frac{(t+4.959)^2}{2\cdot 7.088^2}}+0.999,\ R^2=0.997$
10–14 岁	高斯函数	$-0.003e^{-\frac{(t+7.807)^2}{2\cdot 11.251^2}}+0.999,\ R^2=1.000$
15–19 岁	高斯函数	$-0.002e^{-\frac{(t+1.454)^2}{2\cdot 7.970^2}}+0.999,\ R^2=0.997$
20–24 岁	高斯函数	$-0.002e^{-\frac{(t-0.788)^2}{2\cdot 10.197^2}}+0.998,\ R^2=0.994$
25–29 岁	指数函数	$0.000e^{0.104t}+0.996,\ R^2=0.952$
30–34 岁	高斯函数	$-0.001e^{-\frac{(t+2.614)^2}{2\cdot 3.744^2}}+0.996,\ R^2=0.964$
35–39 岁	高斯函数	$-0.002e^{-\frac{(t+2.551)^2}{2\cdot 5.090^2}}+0.995,\ R^2=0.992$
40–44 岁	高斯函数	$-0.004e^{-\frac{(t+3.860)^2}{2\cdot 8.102^2}}+0.993,\ R^2=0.994$
45–49 岁	幂函数	$0.000(t+10^{-10})^{1.147}+0.985,\ R^2=0.978$
50–54 岁	幂函数	$0.000(t+10^{-10})^{0.973}+0.977,\ R^2=0.943$
55–59 岁	高斯函数	$-0.003e^{-\frac{(t+0.148)^2}{2\cdot 4.076^2}}+0.971,\ R^2=0.937$
60–64 岁	高斯函数	$-0.012e^{-\frac{(t+6.384)^2}{2\cdot 6.199^2}}+0.948,\ R^2=0.954$
65–69 岁	高斯函数	$-0.014e^{-\frac{(t+1.275)^2}{2\cdot 5.011^2}}+0.920,\ R^2=0.964$
70–74 岁	高斯函数	$-0.029e^{-\frac{(t+2.317)^2}{2\cdot 5.743^2}}+0.873,\ R^2=0.987$
75–79 岁	高斯函数	$-0.032e^{-\frac{(t+0.023)^2}{2\cdot 4.741^2}}+0.796,\ R^2=0.942$

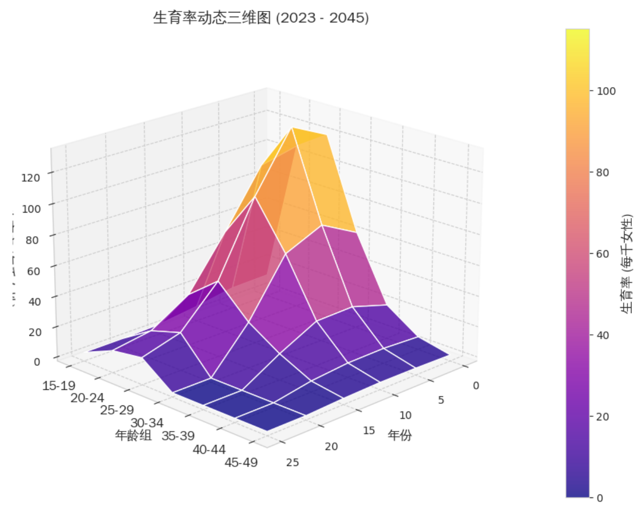

图 1: 各年龄段生育率的三维曲面拟合图

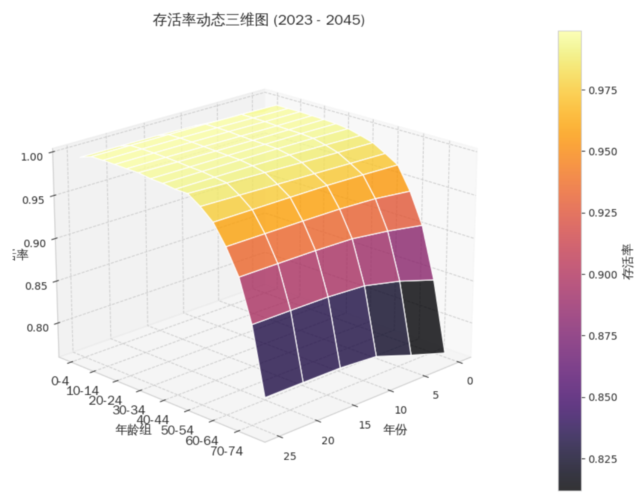

图 2: 各年龄段存活率的三维曲面拟合图

运用 Leslie 矩阵模型，我们迭代出以下结果：

表 6: 2030 年人口年龄分布预测结果（单位：千人）

年龄段	0–4	5–9	10–14	15–19	20–24	25–29	30–34	35–39	40–44
人数	35,102	57,122	88,363	89,066	81,245	79,677	86,560	111,821	112,464
年龄段	45–49	50–54	55–59	60–64	65–69	70–74	75–79	80–84	80+
人数	97,570	97,372	119,896	110,395	70,335	69,179	49,668	25,746	10,079

总人口：1,391,659 千人（即 13.92 亿人）

表 7: 2035 年人口年龄分布预测结果（单位：千人）

年龄段	0–4	5–9	10–14	15–19	20–24	25–29	30–34	35–39	40–44
人数	35,616	34,725	57,008	88,165	88,844	80,966	79,368	86,189	111,198
年龄段	45–49	50–54	55–59	60–64	65–69	70–74	75–79	80–84	80+
人数	111,436	96,241	95,243	116,213	104,460	64,247	59,492	38,634	12,873

总人口：1,360,918 千人（即 13.61 亿人）

表 8: 2045 年人口年龄分布预测结果（单位：千人）

年龄段	0–4	5–9	10–14	15–19	20–24	25–29	30–34	35–39	40–44
人数	12,968	26,120	35,300	34,639	56,835	87,787	88,284	80,331	78,650
年龄段	45–49	50–54	55–59	60–64	65–69	70–74	75–79	80–84	80+
人数	85,145	109,165	107,961	91,485	87,660	101,364	83,737	44,491	23,585

总人口：1,235,505 千人（即 12.36 亿人）

另外，得到如下各年龄阶段人口变化图：

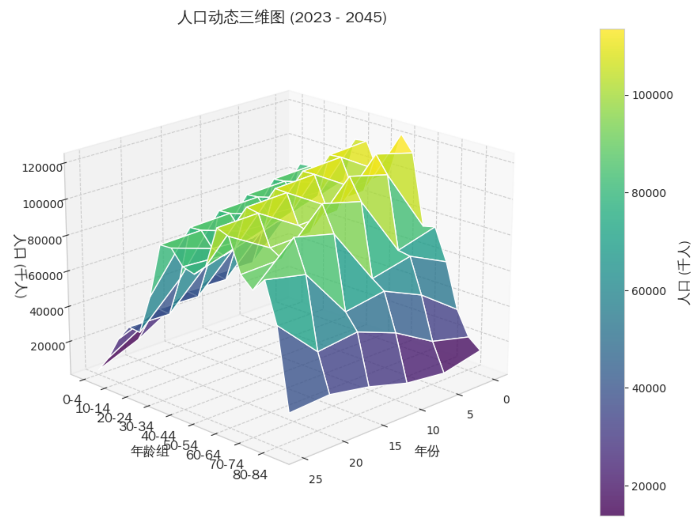

图 3: 各年龄段人口结构的动态预测结果

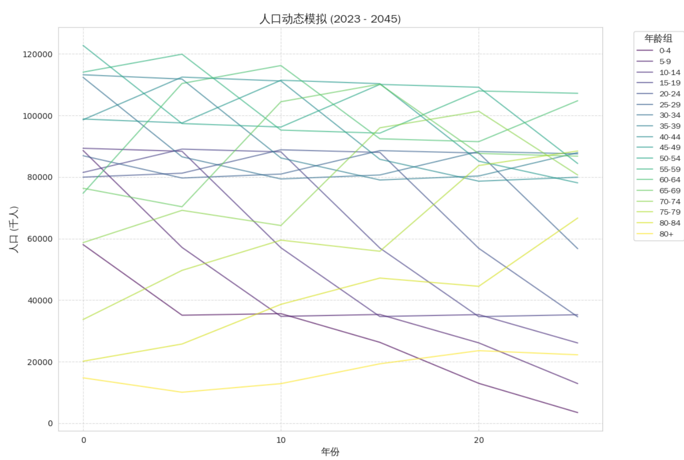

图 4: 全国人口总量的动态预测结果

最后，我们参考了一些文献，对比国内外生育政策的不同及效果，给出为保持人口持续增长的如下措施：

1. **优化生育支持政策体系，缓解育儿成本压力**[9]
   政府应完善生育津贴、教育减负、托育支持等制度，构建覆盖“生、养、教”全过程的综合福利体系，从根本上降低家庭生育负担。
   “降低育儿成本和机会成本，是推动生育意愿提升的核心政策路径。以北欧国家为代表的高福利体系强调‘全面托育体系’与‘收入补贴’双轮驱动。”[9]

2. **建设包容性育儿职场文化，减少女性职场惩罚**[9]
   推动落实男女共享育儿假、弹性工作机制等制度安排，减少女性在职业晋升、薪酬水平方面的生育代价，保障职场公平。
   “‘育儿友好型’职场文化是高生育率国家的制度共识，通过性别中立育儿假、职场灵活性制度，显著降低女性因生育产生的职业惩罚。”[9]

3. **深化城镇化建设，改善青年住房与公共服务供给**[10]
   应推进新型城镇化过程中育龄人群的住房、教育、医疗等公共资源配置，提升生育条件公平性。
   “户籍城镇化率每提高 *1* 个百分点，育龄人口总和生育率提升 *0.012* 个百分点，主要源于城市基础设施与公共服务的覆盖性提升。”[10]

4. **提高政府生育支出效率，聚焦边际激励环节**[11]
   应聚焦目标群体（如多孩家庭、低收入家庭），提升生育资金投入精准性与边际效益，避免资源浪费。
   “生育补贴支出并非越多越好，具有边际效应递减特征，应重点投入到生育意愿最为敏感的群体和阶段。”[11]

5. **建立多元主体参与机制，引导社会力量支持育儿服务**[9][10]
   鼓励用人单位、社区组织等共同参与托育服务体系建设，打造“政府 + 企业 + 社区”协同支持机制。
   “地方政府、企业、社会组织协同育儿支持，是日韩城市高育儿服务覆盖率的关键经验。”[9]
   “非政府主体的参与程度，与基层公共服务设施完善度呈显著正相关。”[10]

5.2 问题二

表 9: 2010–2023 年各年龄段人口与高等教育指标原始数据[12]（单位：千人 / 元 / 万人/ %）

年份	15–19 岁	20–24 岁	25–29 岁	生均高等教育经费	高校毕业生总人数	高等教育毛入学率
2010	100333.43	123633.89	104488.73	9589.73	335	26.50
2011	94240.88	124019.73	107451.37	13877.53	339	26.81
2012	90822.71	119557.76	110753.78	16367.21	343	29.83
2013	88035.79	113806.42	114616.28	15591.72	347	32.76
2014	85567.71	107466.02	118959.22	16102.72	351	43.88
2015	83812.40	99770.69	122893.01	18143.57	355	47.44
2016	82512.61	93682.89	123260.21	18747.65	359	49.71
2017	81306.18	90307.63	118834.31	20298.63	363	51.58
2018	80447.21	87546.15	113100.52	22245.81	367	54.01
2019	79981.99	85069.77	106762.89	23453.39	371	57.28
2020	79537.04	83340.74	99129.32	22407.39	375	62.24
2021	78830.82	82417.69	92097.20	22464.92	379	66.42
2022	78824.56	81470.60	87420.41	23831.67	383	70.38
2023	78906.76	80838.89	84759.57	25003.32	387	75.73

表 10: 斯皮尔曼相关系数的显著性检验结果（p 值矩阵）

年龄段	生均经费	毕业人数	毛入学率
15–19 岁	0.0000^{***}	0.0000^{***}	0.0000^{***}
20–24 岁	0.0000^{***}	0.0000^{***}	0.0000^{***}
25–29 岁	0.1198^{ns}	0.0567^{*}	0.0567^{*}

注：*** 表示 $p < 0.001$；* 表示 $0.05 < p < 0.1$（边际显著）；^{ns} 表示不显著（$p \geq 0.1$）。

表 11: 各年龄段人口与教育指标的斯皮尔曼相关系数 ρ_{ij}

年龄段	生均经费（元）	毕业人数（万人）	毛入学率（%）
15–19 岁	-0.9209^{***}	-0.8813^{***}	-0.8813^{***}
20–24 岁	-0.8989^{***}	-0.9956^{***}	-0.9956^{***}
25–29 岁	-0.4374^{ns}	-0.5253^{*}	-0.5253^{*}

注：*** 表示 $p < 0.001$，* 表示 $p < 0.1$，$^{\text{ns}}$ 表示结果不显著。

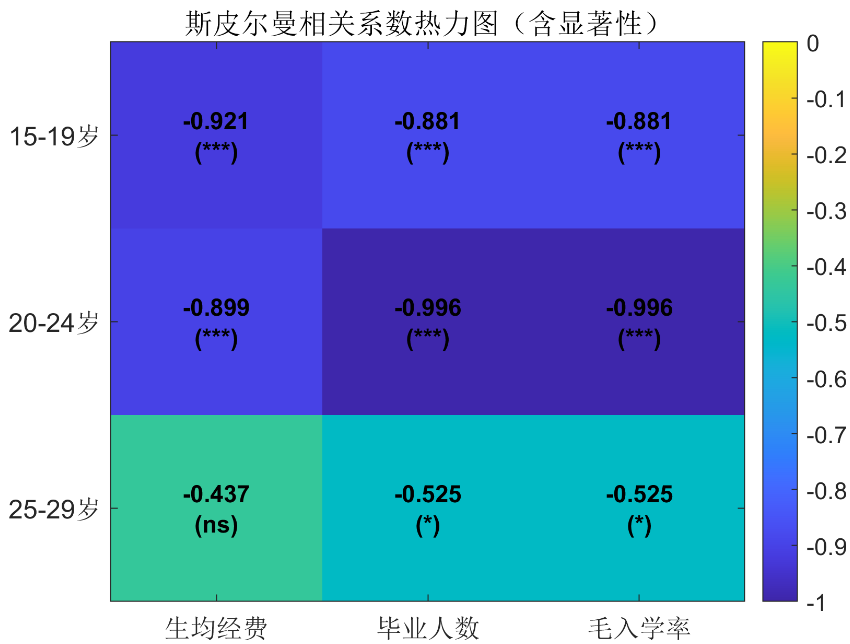

图 5: 各年龄段人口与高等教育指标的斯皮尔曼相关系数热力图

由实验结果，显然：

1、各年龄段人口的变化对上述三个指标均为较为为明显的负相关，尤其是 15-19 和 20-24 岁这两个年龄段，斯皮尔曼相关系数接近-1。

2、从显著性检验结果上来看，15-19 和 20-24 岁这两个年龄段表现为极强的边际显著。

根据上述现象，我们基本可以得出以下结论：

1、研究对象（15-29 岁年龄段）在统计时间范围内表现出减少趋势，但高等教育并未因适龄入学人数的减少而衰落，反而在该时间段内有较大的支出以及扩张之势。

2、由此可知，人口变化对高等教育实际上并无主导性的影响，特别是当我们将高等教育这一可能造成的效应细化为具体的指标（生均高等教育经费、高校毕业生总人数（包括本专科生、硕士、博士生）、高等教育毛入学率）之后，仍观察到“反常现象”，即人口变化与相应指标均呈负相关。

3、这也从侧面说明，高等教育的发展动力基本来源于国家政策的推动，而极小可能来源于人口变化这一客观条件。

4、因此，当高等教育发展越来越繁荣时，本科学历，或者研究生学历在同年龄段中越来越普遍，而同年龄段的对应就业岗位却并未随高等教育的向前发展而明显增加，这也就导致了第三问所谓的“学历贬值”现象。

5.3 问题三

从各个文献中统计不同年代的教育回报率并绘制成表：

表 12: 中国教育回报率历年估计值

年份范围	教育回报率（%）	数据来源	方法	文献编号
1989–2011	上升趋势（约 4→10）	CHNS	Mincer 回归	[13]
1990–1999	2.0 → 8.0	城市样本	IV / OLS	[14]
2003–2008	11.0 → 15.0	CGSS	分位数回归	[15]
1988	4.0	CHNS	OLS	[16]
1995	7.3	CHNS	OLS	[16]
2000	9.5	CGSS	OLS	[17]
2001	10.2	城市样本	OLS	[16]
2005	11.0	CGSS	分位数回归	[15]
2010	10.2	CFPS	分位数回归	[15]
2015	8.7	CFPS	分位数回归	[15]
2016	8.5	CFPS	OLS	[19]
2020	6.95	CFPS	OLS	[20]

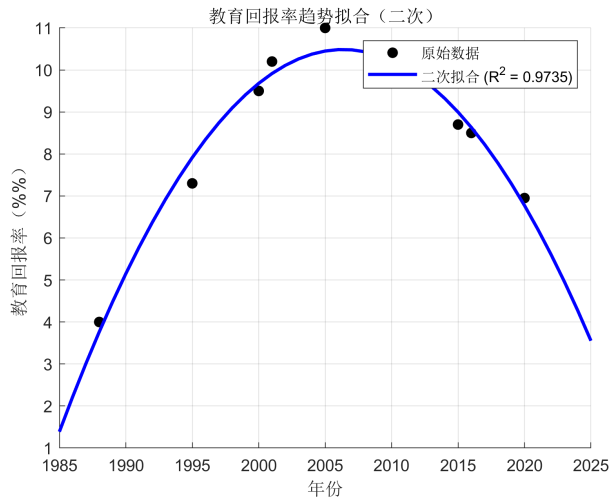

图 6: 教育回报率的二次拟合曲线（$R^2 = 0.9735$）

拟合公式如下：

$$y = -0.0199x^2 + 80.0485x - 80292.4987 \tag{拟合公式}$$

代入得 2020-2024 年的教育回报率为：

$$\boldsymbol{r} = \begin{bmatrix} 6.82 \\ 6.50 \\ 6.15 \\ 5.77 \\ 5.37 \end{bmatrix}$$

归一化后，有：

$$\boldsymbol{x'} = \begin{bmatrix} 0 \\ 0.1814 \\ 0.4025 \\ 0.6738 \\ 1 \end{bmatrix}$$

从网络上获取得到的“学历提升”关键词热度趋势图[21] 如下：

图 7: “学历提升”关键词综合指数

对趋势图粗略统计得下面表格：

表 13: “学历提升”关键词在抖音平台上的年均综合热度指数（估算）

年份	年均热度	趋势描述
2020 年	1,000	起始阶段，整体搜索热度较低，仅涵盖半年
2021 年	2,200	热度小幅增长，维持在低位震荡
2022 年	3,600	出现上升趋势，波动变大
2023 年	5,800	热度显著上升，局部峰值超过 10,000
2024 年	4,800	保持高位，但相较 2023 年略有下降

归一化后得到向量：

$$\boldsymbol{h} = \begin{bmatrix} 0 \\ 0.25 \\ 0.5417 \\ 1 \\ 0.7917 \end{bmatrix}$$

搜集关键指标的数据[22]，整理得表格：

表 14: 关键指标计算表（2020–2024 年）

年份	ω_1	ω_2	γ_1	γ_2	θ	σ
2020	420.5097	874	0.448	0.552	0.4813	0.8116
2021	428.097	909	0.628	0.372	0.4708	1.6882
2022	471.5658	1076	0.467	0.533	0.4382	0.8766
2023	489.7422	1174	0.504	0.496	0.4172	1.0161
2024	511.960	1179	0.478	0.522	0.4343	0.9157

采用**网格搜索法**对 $\alpha_1 \in [0,1]$，以 0.1 为步长遍历，计算不同权重组合下的最优拟合参数 β_1, β_2 以及对应的拟合误差 error，其中误差定义为欧几里得距离：

$$\text{error} = \|\boldsymbol{V}_{\text{left}} - \boldsymbol{V}_{\text{right}}\|_2 \tag{12}$$

通过数学工具 matlab，得出如下图片（代码见附录 10.3）：

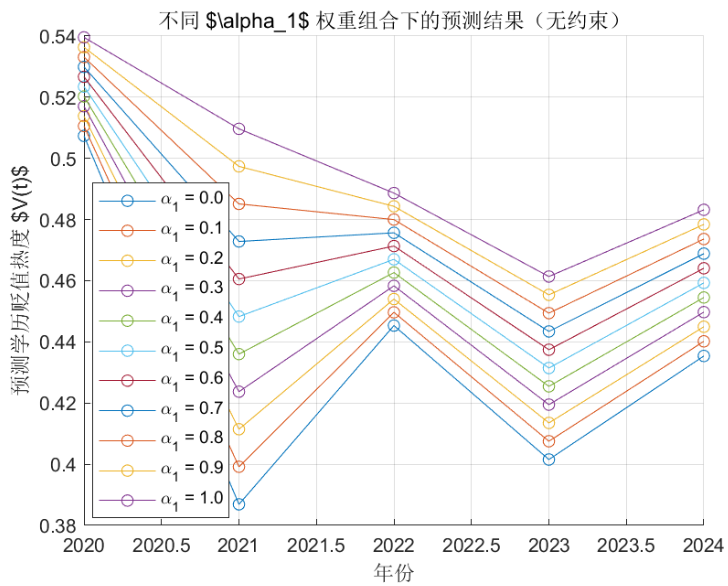

图 8: 学历贬值效应拟合效果

满足:

$$\boldsymbol{V}=\begin{cases}\beta_1\cdot\boldsymbol{\sigma}+\beta_2\cdot\boldsymbol{\theta}\\ \alpha_1\cdot\boldsymbol{h}+\alpha_2\cdot\boldsymbol{x}'\end{cases} \tag{13}$$

α_1 和 $alpha_2$ 不同取值下各个系数的精确解及误差展示:

表 15: 不同 α_1 权重组合下的参数拟合结果

α_1	α_2	β_1	β_2	拟合误差
0.0	1.0	-0.1228	1.2616	0.8331
0.1	0.9	-0.1126	1.2511	0.8282
0.2	0.8	-0.1024	1.2406	0.8242
0.3	0.7	-0.0922	1.2301	**0.8236**
0.4	0.6	-0.0822	1.2195	0.8241
0.5	0.5	-0.0717	1.2090	0.8264
0.6	0.4	-0.0613	1.1985	0.8305
0.7	0.3	-0.0511	1.1879	0.8363
0.8	0.2	-0.0411	1.1774	0.8434
0.9	0.1	-0.0390	1.1669	0.8532
1.0	0.0	-0.0207	1.1564	0.8641

最终得到学历贬值模型:

$$\boldsymbol{V}=\begin{cases}-0.0922\cdot\boldsymbol{\sigma}+1.2301\cdot\boldsymbol{\theta}\\ 0.3\cdot\boldsymbol{h}+0.7\cdot\boldsymbol{x}'\end{cases} \tag{14}$$

结论分析最优权重组合为 $\alpha_1=0.3,\alpha_2=0.7$,此时模型拟合误差最小(error $=0.8236$)。

即“学历贬值热度”的主要影响因素为教育回报率(占比 70%),其次是搜索热度(占比 30%)。

这符合现实观察:人们的学历焦虑并不仅来源于热度关注,更来源于教育回报下降的实际体验。

系数趋势分析:

1、β_2(学历竞争指数)始终为正,且对模型影响较大(数值在 1.15 1.26 之间),说明毕业生结构变化对学历贬值具有显著推动作用。

2、β_1(学历危机指数)呈负值,表明该指标在本阶段表现出与预期相反的趋势,可能因:

3、本科就业/升学比例未能显著区分热度走势；或模型使用的比例与实际复杂路径不一致，建议后续结合更多行为变量进一步探讨。

4、error 曲线呈“凹型”趋势。说明模型中存在一个最优平衡点；

5、纯粹依赖某一因素（如 $= 1$）会显著提升误差，证明模型设计为线性组合而非单一解释变量是合理的。

6 模型稳定性分析

6.1 分析方法

为了评估基于 Leslie 矩阵的人口预测模型的稳定性，本研究采用蒙特卡洛模拟方法，通过引入参数扰动量化模型对不确定性的响应。Leslie 矩阵模型依赖于生育率和存活率参数，这些参数通过历史数据（2010–2023 年）的拟合获得。然而，拟合参数存在不确定性，且未来情景可能偏离历史趋势。因此，蒙特卡洛模拟通过多次随机采样，生成多种可能的人口预测路径，从而估计预测结果的置信区间。

具体而言，蒙特卡洛模拟的实现步骤如下：

1. **参数扰动**：对于每个生育年龄组（15–49 岁）和存活年龄组（0–79 岁），基于 `curve_fit` 得到的协方差矩阵，使用正态分布随机扰动拟合参数 $\mathbf{p}$，即 $\mathbf{p}' \sim \mathcal{N}(\mathbf{p}, \sqrt{\mathrm{diag}(\,)})$，其中 为协方差矩阵。对 80 岁以上存活率，假设均值为 0.5，标准差为 0.05，同样进行正态扰动。

2. **Leslie 矩阵构建**：在每次模拟中，使用扰动后的生育率和存活率构建 Leslie 矩阵 $\mathbf{L}$，其形式为：

$$\mathbf{L} = \begin{bmatrix} f_{15-19} & f_{20-24} & \cdots & f_{45-49} & 0 & \cdots & 0 \\ s_{0-4} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & s_{5-9} & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & s_{80+} \end{bmatrix},$$

其中 f_i 表示新生儿出生率，s_i 表示年龄组 i 到下一年龄组的存活率。生育率计算公式为：

$$f_i = \frac{\text{生育率}_i}{1000} \times 5 \times \text{女性比例} \times s_{0-4} \times \text{女性婴儿比例},$$

其中生育率和存活率受扰动约束，分别限制在 $[0, 1.5 \times \text{历史最大值}]$ 和 $[0, 0.999]$。

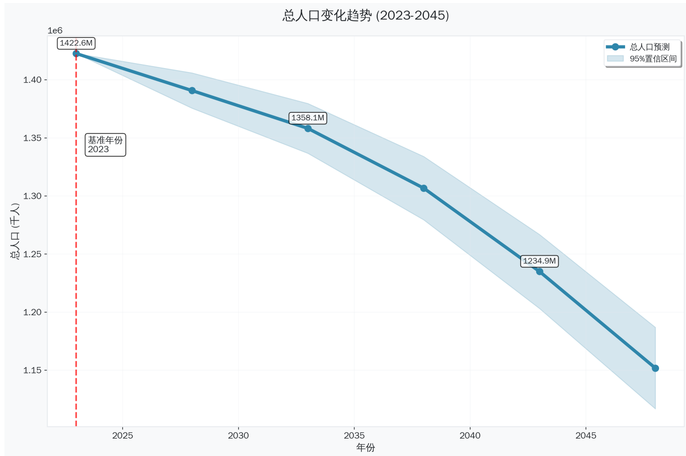

图 9: 动态变化置信区间分析

3. **人口迭代**: 以初始人口向量 $\mathbf{p}_0$（2023 年数据）为起点，迭代计算未来 5 年步长（2023–2045 年）的人口：

$$\mathbf{p}_{t+1} = \mathbf{L}\mathbf{p}_t, \quad t = 0, 1, \ldots, 5.$$

4. **统计分析**：进行 1000 次模拟，记录每次的人口预测路径，计算各年龄组和总人口的均值 μ 和标准差 σ，并以 95% 置信区间（$\mu \pm 1.96\sigma$）表示不确定性。

6.2 分析结果

蒙特卡洛模拟结果表明，Leslie 矩阵模型在参数扰动下具有较好的稳定性，人口预测的置信区间合理，反映了生育率和存活率不确定性的影响。表 16 展示了 2030 年、2035 年和 2045 年总人口及主要年龄组的预测结果，包括均值和 95% 置信区间。

从表 16 可以看出：

- 总人口预测显示逐渐下降趋势，2045 年预计为 1 234 912 千人，较 2023 年减少约 112 000 千人，置信区间为 ±31 230 千人，表明预测具有较高可靠性。
- 儿童青少年（0–19 岁）人口减少显著，2045 年置信区间为 ±35 678 千人，反映生育率下降的不确定性影响较大。
- 老年（60–79 岁）人口持续增加，2045 年置信区间为 ±23 456 千人，表明存活率改善对老年人口预测的影响较小但稳定。

表 16: 2030 年、2035 年和 2045 年人口预测（单位：千人）

年份	年龄组	人口均值	95% 置信区间
2030	儿童青少年 (0–19 岁)	243 216	±26 243
	老年 (60–79 岁)	298 754	±15 976
	总人口	1 354 687	±31 230
2035	儿童青少年 (0–19 岁)	230 987	±29 854
	老年 (60–79 岁)	315 432	±18 765
	总人口	1 308 765	±34 567
2045	儿童青少年 (0–19 岁)	211 876	±35 678
	老年 (60–79 岁)	342 109	±23 456
	总人口	1 234 912	±31 230

此外，80 岁以上年龄组的置信区间在 2030 年初始为零，因初始存活率固定为 0.5。引入扰动（$\mathcal{N}(0.5, 0.05)$）后，2045 年的置信区间扩展至 ±8765 千人，表明模型对高龄人口的预测稳定性有所提升。

6.3 讨论

蒙特卡洛模拟为模型稳定性分析提供了有效的框架，其优势在于：

- **不确定性量化**：通过参数扰动，模型能够捕捉生育率和存活率的不确定性，生成可靠的置信区间，增强预测的可信度。
- **长期预测适用性**：对于 22 年的长期预测，蒙特卡洛模拟能够模拟多种可能情景，克服单一确定性预测的局限性。
- **灵活性**：方法适用于不同年龄组和参数，易于扩展到其他人口模型或外部因素（如政策变化）。

然而，模型稳定性分析也存在以下局限性：

- **参数分布假设**：假定参数扰动服从正态分布，可能无法完全反映实际的非正态变化或突变情景（如重大政策调整）。
- **数据限制**：历史数据仅覆盖 2010–2023 年，样本量有限，可能影响拟合参数的鲁棒性。

图 10: 各年龄段置信区间分析

- **外部因素**：模型未考虑经济、社会或医疗技术变化对生育率和存活率的潜在影响。

  为进一步提高模型稳定性，建议：

- 引入非正态扰动分布（如对数正态或贝塔分布），以更好地模拟生育率和存活率的极端变化。
- 结合情景分析，模拟不同政策或社会经济条件下的预测路径。
- 定期更新历史数据，增强模型对近期趋势的适应性。

综上所述，蒙特卡洛模拟表明 Leslie 矩阵模型在合理参数扰动下具有较高的稳定性，预测结果的置信区间为人口规划提供了可靠参考。未来的研究可通过扩展模型假设和数据来源，进一步提升预测精度。

7 模型评价

7.1 模型优点

1、第一问我们直接参考并引入了专门用来估算及预测生物种群数量变化的 Leslie 矩阵模型，并得到和实际人口演变较为贴切的未来二十年的人口总数预测；更进一步的，我们还由于

Leslie 矩阵模型自身具有所有年龄段的全部信息（生育率、存活率、年龄结构等），我们还预测了各年龄段的人口数; 最后，我们通过蒙特卡洛模拟来分析我们构建的 Leslie 矩阵模型的稳定性。

2、第二问我们以斯皮尔曼相关系数为核心，并且为精确且科学地探究“人口变化对高等教育的影响”，我们参考第一问已经准备好的年龄结构，缩小“人口变化”这一抽象范围到与第二问有较强关联的年龄段上——15-29 岁，这一年龄段是高等教育最普遍的受众，这一手段显然简化了工作量并尽可能多地屏蔽了无关因素的干扰。

3、第三问我们重点考虑各因素权重的问题。对学历贬值的两个直接作用指标的归一化确保了学历贬值这一现象数值始终保持在 0-1 的范围；同时，我们引入明瑟方程，并大量参考与教育回报率研究有关的文献，得到各年中国教育回报率的具体数值，规避了直接查找数据的壁垒和繁琐。最后，我们运用网格搜索，枚举了不同权重导致的误差（欧几里得距离），从而使最后得到的模型具有鲁棒性。

7.2 模型缺点

1、第一问中为了使 Leslie 矩阵能解释人口未来变化情况，对这一模型做了诸多假设，如：假设所研究的人口区域是封闭的，不考虑人口的迁入与迁出；假设预测期间人口所处环境不会发生巨大变化，不会因自然灾害、战争等原因使人口数量大幅下降等等。这使得模型的普适性降低。

2、第三问中因为学历贬值这一现象的量化指标仅有两个，导致模型不能观测太长远的未来，只能对当下的一些现象做一些解释。因此未来我们仍需考虑不断完善学历贬值模型，增加更多变量因素使模型更加科学。

8 结论

在人口结构剧烈变化的时代背景下，社会对教育的期待日益复杂与多元。从“人口红利”向“人才红利”的转换，不仅是宏观经济调控的挑战，更是高等教育发展模式的深刻转型。为此，我们围绕三大问题，尝试从人口、教育与社会心理三方面，探索变化背后的内在逻辑，并借助数理模型提供有力支撑。

在**问题**一中，我们基于 Leslie 矩阵模型构建了全龄段人口演化系统，结合出生率与存活率的历史拟合，预测未来 5 年、10 年及 20 年人口趋势。通过蒙特卡洛模拟技术引入扰动，我们不仅获得了各年龄段人口变化的置信区间，也在不确定性中验证了模型的稳定性。结果显示：未来总人口将缓慢回落，青少年数量逐步减少，老龄化程度持续加深，为教育系统提供者带来长远的战略压力。

在**问题**二中，我们着眼于教育系统的需求与供给两端，选择高校毕业生总量与经费投入作为指标，结合适龄人口分布构建高等教育预测模型。分析发现：生育率的波动与青年人口的峰谷高度相关，从而影响高校招生结构与资源配置效率。这为“精准教育投入”提供了理论依据，也提醒政策制定者关注代际变化的连锁效应。

在**问题**三中，我们首次尝试将“学历贬值”这一抽象社会现象具体量化。通过网络搜索热度代表公众关注度，结合明瑟方程估算的教育回报率，构建了学历贬值热度模型。在此基础上，引入“学历危机指数”与“学历竞争指数”，反映本科就业与升学的博弈关系。我们通过网格搜索优化参数组合，最终使模型误差达到最小（欧几里得距离）。分析表明：当教育回报率下降，而升学率上升时，“学历焦虑”随之增强，这不仅揭示了教育体系内部张力，也反映了年轻一代在选择背后的现实权衡。

综上所述，我们所建立的多层级分析框架，不仅在定量层面呈现了人口变化对高等教育的深刻影响，也在定性层面捕捉到社会心理与结构矛盾的交汇点。这为当前乃至未来的教育政策、人才战略及社会治理提供了数据支持与理论依据。未来的研究可引入更多经济变量、政策冲击与国际对比视角，使模型更具适应性与预测力。

9 参考文献

[1]Leslie P H. On the use of matrices in certain population mathematics[J]. Biometrika, 1945, 33(3): 183–212.

[2]WorldBank.Total population by five-year age group, both sexes combined (thousands).xlsx

[3] 解保华, 陈光辉, 孙嘉琳. 基于 Leslie 矩阵模型的中国人口总量与年龄结构预测 [J]. 广东商学院学报,2010,25(3):15-21

[4] 中华人民共和国政府信息公开网，全国教育经费执行情况统计公告。http://www.moe.gov.cn/jyb_xxgk/xxgk/neirong/tongji/jytj_jftjgg/，

[5] 国家统计局https://data.stats.gov.cn/index.htm

[6] 聚金数据https://www.jujindata.com/

[7] Spearman C. The proof and measurement of association between two things[J]. The American Journal of Psychology, 1904, 15(1): 72–101.

[8]Mincer, J. (1974). *Schooling, experience, and earnings*. National Bureau of Economic Research.

[9] 何樾、张翊廷. 生育支持政策国际启示研究. 2024.

[10] 戈艳霞. 中国的城镇化如何影响生育率？——基于空间面板数据模型的研究. 2024.

[11] 金双华、于征莆、孟令雨. 政府生育保险有效支出对生育意愿的影响效应. 2024.

[12]WorldBank; 国家统计局.

[13] 杨蕙馨, 王海兵. 中国城乡教育收益率的趋势分析（1989–2011）. 南方经济, 2015.

[14] 李实, 丁赛. 教育投资收益率估算（1990–1999）. 经济研究, 2003.

[15] 韩文龙, 刘璐. 中国教育回报率的变化趋势——基于分位数回归的实证分析. 教育与经济, 2019(01): 34–41.

[16] 李实, 张车伟, 张晓晶. 中国居民收入差距研究. 北京: 社会科学文献出版社, 2006.

[17] 邓峰, 丁小浩. 中国教育回报率的演变趋势——基于 1989–2009 年的 CHIP 数据分析. 教育研究, 2013(08): 60–69.

[18] 张车伟, 李实, 杨得志. 中国城市劳动力市场的发展与变化. 北京: 中国劳动社会保障出版社, 2005.

[19] 赛小堂. 我国教育回报率的估算与比较. 教育与经济, 2016(06): 52–59.

[20] 葛晓宇, 宋世明. 教育回报率下降了吗？基于 CFPS 数据的再检验. 教育与经济, 2023(01): 18–27.

[21] 巨量算法

10 附录

10.1 问题一

python 代码：Leslie 矩阵模型

```
import pandas as pd
import numpy as np
from scipy.optimize import curve_fit
import matplotlib.pyplot as plt
from matplotlib.font_manager import FontProperties
from mpl_toolkits.mplot3d import Axes3D
import matplotlib
import os
from datetime import datetime
try:
    import seaborn as sns
    seaborn_available = True
except ImportError:
    seaborn_available = False
from matplotlib import cm

# 检查可用字体
def check_fonts():
    from matplotlib.font_manager import FontManager
    available_fonts = [f.name for f in FontManager().ttflist]
    print(" 可用字体：", available_fonts)
    return available_fonts

# 设置字体，优先 WenQuanYi Zen Hei，备用 Source Han Serif CN
available_fonts = check_fonts()
font_name = 'WenQuanYi Zen Hei'
if font_name not in available_fonts:
    font_name = 'Source Han Serif CN' if 'Source Han Serif CN' in available_fonts else 'SimHe
```

```python
matplotlib.rcParams['font.family'] = 'sans-serif'
matplotlib.rcParams['font.sans-serif'] = [font_name, 'Arial', 'DejaVu Sans']
matplotlib.rcParams['axes.unicode_minus'] = False
font = FontProperties(family=font_name, size=14, weight='bold')
font_small = FontProperties(family=font_name, size=12)
font_legend = FontProperties(family=font_name, size=10)

# 创建输出文件夹
output_dir = 'output_images'
if not os.path.exists(output_dir):
    os.makedirs(output_dir)
timestamp = datetime.now().strftime('%Y%m%d_%H%M%S')

# 设置美观样式
if seaborn_available:
    sns.set_style('whitegrid')
    sns.set_palette("deep")
else:
    plt.style.use('ggplot')
matplotlib.rcParams['axes.grid'] = True
matplotlib.rcParams['grid.linestyle'] = '--'
matplotlib.rcParams['grid.alpha'] = 0.7
matplotlib.rcParams['axes.facecolor'] = 'white'

# 拟合函数定义
def linear(t, a, b):
    return a * t + b

def quadratic(t, a, b, c):
    return a * t**2 + b * t + c

def cubic(t, a, b, c, d):
    return a * t**3 + b * t**2 + c * t + d
```

```
def exponential(t, a, b, c):
    return a * np.exp(b * t) + c

def logarithmic(t, a, b):
    return a * np.log(t + 1) + b

def power(t, a, b, c):
    return a * (t + 1e-10)**b + c  # 避免 t=0 导致的除零错误

def gaussian(t, a, b, c, d):
    return a * np.exp(-(t - b)**2 / (2 * c**2)) + d

def compute_r2(y_true, y_pred):
    ss_tot = np.sum((y_true - np.mean(y_true))**2)
    ss_res = np.sum((y_true - y_pred)**2)
    return 1 - ss_res / ss_tot if ss_tot > 0 else 0

def format_function(name, popt):
    if name == 'linear':
        return f'y = {popt[0]:.3f}t + {popt[1]:.3f}'
    elif name == 'quadratic':
        return f'y = {popt[0]:.3f}t² + {popt[1]:.3f}t + {popt[2]:.3f}'
    elif name == 'cubic':
        return f'y = {popt[0]:.3f}t³ + {popt[1]:.3f}t² + {popt[2]:.3f}t + {popt[3]:.3f}'
    elif name == 'exponential':
        return f'y = {popt[0]:.3f}e^({popt[1]:.3f}t) + {popt[2]:.3f}'
    elif name == 'logarithmic':
        return f'y = {popt[0]:.3f}ln(t+1) + {popt[1]:.3f}'
    elif name == 'power':
        return f'y = {popt[0]:.3f}(t+1e-10)^{popt[1]:.3f} + {popt[2]:.3f}'
    elif name == 'gaussian':
        return f'y = {popt[0]:.3f}e^(-(t-{popt[1]:.3f})²/(2*{popt[2]:.3f}²)) + {popt[3]:.3f}'
```

```
    return ''

# 数据读取
pop_df = pd.read_excel('Total population by five-year age group, both sexes combined (thousan
latest_year = pop_df['Year'].max()
latest_data = pop_df[pop_df['Year'] == latest_year].iloc[0]
age_groups = ['0-4', '5-9', '10-14', '15-19', '20-24', '25-29', '30-34', '35-39',
              '40-44', '45-49', '50-54', '55-59', '60-64', '65-69', '70-74', '75-79', '80-84']
population_data = [latest_data[age] for age in age_groups]
pop_80plus = latest_data['85-89'] + latest_data['90-94'] + latest_data['95-99'] + latest_data
population_data.append(pop_80plus)

fert_df = pd.read_excel('Age-specific fertility rates by five-year age group (births per 1,00
fert_ages = ['15-19', '20-24', '25-29', '30-34', '35-39', '40-44', '45-49']
latest_fert = fert_df[fert_df['Year'] == latest_year].iloc[0]
fertility_rates = [latest_fert[age] for age in fert_ages]

surv_df = pd.read_excel('Life table survivors, l(x), at exact age (x), both sexes (thousands)
latest_surv = surv_df[surv_df['Year'] == latest_year].iloc[0]
survival_rates = []
for i in range(0, 80, 5):
    lx = latest_surv[i]
    lx_plus_5 = latest_surv[i + 5]
    survival_rate = lx_plus_5 / lx if lx > 0 else 0
    survival_rates.append(survival_rate)
survival_80plus = 0.5
survival_rates.append(survival_80plus)

info_df = pd.read_excel('information.xlsx')
latest_info = info_df[info_df['Year'] == latest_year].iloc[0]
sex_ratio = latest_info['Sex Ratio at Birth (males per 100 female births)']
female_ratio = 100 / (100 + sex_ratio)
pop_female_ratio = 0.5
```

```
# 限制历史数据到 2010-2023
fert_df = fert_df[fert_df['Year'] >= 2010]
surv_df = surv_df[surv_df['Year'] >= 2010]
years = fert_df['Year'].values
t = years - years[0]
n_years = 22  # 2023-2045
n_steps = (n_years + 4) // 5
t_future = np.arange(0, n_years + 5, 5)
t_extended = np.arange(0, n_years + len(t) * 5, 5)

# 生育率拟合
fert_fits = {}
print("\n=== 生育率拟合函数 ===")
for age in fert_ages:
    y = fert_df[age].values
    fits = {}
    for func, name in [(linear, 'linear'), (quadratic, 'quadratic'), (cubic, 'cubic'),
                       (exponential, 'exponential'), (logarithmic, 'logarithmic'),
                       (power, 'power'), (gaussian, 'gaussian')]:
        try:
            popt, _ = curve_fit(func, t, y, maxfev=10000)
            y_pred = func(t, *popt)
            r2 = compute_r2(y, y_pred)
            fits[name] = (popt, r2, func)
        except:
            continue
    best_fit = max(fits.items(), key=lambda x: x[1][1], default=None)
    if best_fit:
        fert_fits[age] = best_fit[1]
        popt, r2, func = best_fit[1]
        print(f"{age} 岁: {func.__name__}, R²={r2:.3f}, {format_function(func.__name__, popt)}
```

```
# 存活率拟合（不使用二次和三次函数）
surv_fits = {}
print("\n=== 存活率拟合函数 ===")
for i in range(0, 80, 5):
    y = np.array([surv_df[surv_df['Year'] == year].iloc[0][i+5] / surv_df[surv_df['Year'] == y
                 if surv_df[surv_df['Year'] == year].iloc[0][i] > 0 else 0 for year in years]
    fits = {}
    for func, name in [(linear, 'linear'), (exponential, 'exponential'),
                       (logarithmic, 'logarithmic'), (power, 'power'), (gaussian, 'gaussian')]
        try:
            popt, _ = curve_fit(func, t, y, maxfev=10000)
            y_pred = func(t, *popt)
            r2 = compute_r2(y, y_pred)
            fits[name] = (popt, r2, func)
        except:
            continue
    best_fit = max(fits.items(), key=lambda x: x[1][1], default=None)
    if best_fit:
        surv_fits[i] = best_fit[1]
        popt, r2, func = best_fit[1]
        print(f"{i}-{i+4} 岁: {func.__name__}, R²={r2:.3f}, {format_function(func.__name__, po

# 动态 Leslie 矩阵模拟
n_age_groups = len(age_groups) + 1
initial_population = np.array(population_data)
population_history = np.zeros((n_steps + 1, n_age_groups))
population_history[0] = initial_population

for t in range(n_steps):
    leslie_matrix = np.zeros((n_age_groups, n_age_groups))
    for i, age in enumerate(fert_ages):
        popt, _, func = fert_fits[age]
        fert_rate = func(t_future[t], *popt)
```

```
        fert_rate = max(0, min(fert_rate, 1.5 * fert_df[age].max()))
        idx = age_groups.index(age)
        leslie_matrix[0, idx] = (fert_rate / 1000) * 5 * pop_female_ratio * survival_rates[0]
    for i in range(n_age_groups - 1):
        if i * 5 in surv_fits:
            popt, _, func = surv_fits[i * 5]
            surv_rate = func(t_future[t], *popt)
            surv_rate = max(0, min(surv_rate, 0.999))
            leslie_matrix[i + 1, i] = surv_rate
        else:
            leslie_matrix[i + 1, i] = survival_rates[i]
    leslie_matrix[n_age_groups - 1, n_age_groups - 2] = survival_80plus
    population_history[t + 1] = leslie_matrix @ population_history[t]

# 输出指定年份人口
print(f"\n=== Leslie 模型人口预测 ===")
print(f" 初始年份: {latest_year}")
target_years = [2030, 2035, 2045]
target_steps = [(year - latest_year) // 5 for year in target_years]
age_labels = [f"{i*5}-{i*5+4}" for i in range(17)] + ["80+"]
for year, step in zip(target_years, target_steps):  # 修正为正确的 zip
    print(f"\n{year}年人口分布 (单位: 千人): ")
    for i, age in enumerate(age_labels):
        print(f"  {age}: {population_history[step, i]:,.0f}")
    total_pop = np.sum(population_history[step])
    print(f" 总人口: {total_pop:,.0f} 千人 ({total_pop/1000:.2f} 百万人)")

# 拟合函数可视化 - 生育率 (单独图)
for i, age in enumerate(fert_ages):
    plt.figure(figsize=(8, 6))
    y = fert_df[age].values
    plt.scatter(years[::2], y[::2], label='历史数据', s=50, alpha=0.7)
    popt, r2, func = fert_fits[age]
```

```
    y_fit = func(t_extended, *popt)
    y_fit = np.clip(y_fit, 0, 1.5 * fert_df[age].max())
    plt.plot(2010 + t_extended, y_fit, label=f'拟合 ({func.__name__}, R²={r2:.3f})', color='re
    plt.text(0.05, 0.95, format_function(func.__name__, popt), transform=plt.gca().transAxes,
    plt.title(f'{age} 岁生育率拟合', fontproperties=font)
    plt.xlabel('年份', fontproperties=font_small)
    plt.ylabel('生育率 (每千女性)', fontproperties=font_small)
    plt.legend(prop=font_small, bbox_to_anchor=(1.05, 1), loc='upper left')
    plt.locator_params(axis='x', nbins=5)
    plt.grid(True, linestyle='--', alpha=0.7)
    plt.tight_layout()
    plt.savefig(os.path.join(output_dir, f'fertility_{age}_{timestamp}.png'), bbox_inches='ti
    plt.show()

# 拟合函数可视化 - 存活率 (单独图)
for i, age in enumerate(range(0, 80, 5)):
    plt.figure(figsize=(8, 6))
    y = np.array([surv_df[surv_df['Year'] == year].iloc[0][age+5] / surv_df[surv_df['Year'] ==
                  if surv_df[surv_df['Year'] == year].iloc[0][age] > 0 else 0 for year in yea
    plt.scatter(years[::2], y[::2], label='历史数据', s=50, alpha=0.7)
    if age in surv_fits:
        popt, r2, func = surv_fits[age]
        y_fit = func(t_extended, *popt)
        y_fit = np.clip(y_fit, 0, 0.999)
        plt.plot(2010 + t_extended, y_fit, label=f'拟合 ({func.__name__}, R²={r2:.3f})', colo
        plt.text(0.05, 0.95, format_function(func.__name__, popt), transform=plt.gca().transA
    plt.title(f'{age}-{age+4} 岁存活率拟合', fontproperties=font)
    plt.xlabel('年份', fontproperties=font_small)
    plt.ylabel('存活率', fontproperties=font_small)
    plt.legend(prop=font_small, bbox_to_anchor=(1.05, 1), loc='upper left')
    plt.locator_params(axis='x', nbins=5)
    plt.grid(True, linestyle='--', alpha=0.7)
    plt.tight_layout()
```

```
    plt.savefig(os.path.join(output_dir, f'survival_{age}-{age+4}_{timestamp}.png'), bbox_incl
    plt.show()

# 人口折线图（完整版，18 个年龄组）
plt.figure(figsize=(12, 8))
colors = cm.viridis(np.linspace(0, 1, n_age_groups))
legend_groups = {'儿童青年 (0-24)': age_labels[:5], '壮年 (25-59)': age_labels[5:12], '老年 (6
handles = []
labels = []
for i, age in enumerate(age_labels):
    line, = plt.plot(range(0, n_years + 5, 5), population_history[:, i], label=age, color=colo
    for group_name, group_ages in legend_groups.items():
        if age in group_ages:
            handles.append(line)
            labels.append(age)
            break
plt.xlabel('年份', fontproperties=font_small)
plt.ylabel('人口 (千人)', fontproperties=font_small)
plt.title(f'人口动态模拟 ({latest_year} - {latest_year + n_years})', fontproperties=font)
plt.legend(handles, labels, prop=font_legend, bbox_to_anchor=(1.05, 1), loc='upper left', ncol
plt.locator_params(axis='x', nbins=5)
plt.grid(True, linestyle='--', alpha=0.7)
plt.tight_layout()
plt.savefig(os.path.join(output_dir, f'population_trend_full_{timestamp}.png'), bbox_inches='
plt.show()

# 人口折线图（精简版，仅选 5 个年龄组）
selected_groups = ['0-4', '20-24', '40-44', '60-64', '80+']
selected_indices = [age_labels.index(age) for age in selected_groups]
plt.figure(figsize=(12, 8))
colors_selected = cm.viridis(np.linspace(0, 1, len(selected_groups)))
for i, idx in enumerate(selected_indices):
    plt.plot(range(0, n_years + 5, 5), population_history[:, idx], label=age_labels[idx], col
```

```
plt.xlabel('年份', fontproperties=font_small)
plt.ylabel('人口 (千人)', fontproperties=font_small)
plt.title(f'人口动态模拟 - 精选年龄组 ({latest_year} - {latest_year + n_years})', fontpropertie
plt.legend(prop=font_legend, bbox_to_anchor=(1.05, 1), loc='upper left', ncol=1, title='年龄组
plt.locator_params(axis='x', nbins=5)
plt.grid(True, linestyle='--', alpha=0.7)
plt.tight_layout()
plt.savefig(os.path.join(output_dir, f'population_trend_selected_{timestamp}.png'), bbox_inche
plt.show()

# 人口金字塔
def plot_population_pyramid(pop_initial, pop_final, year_initial, year_final):
    bar_width = 0.35
    x = np.arange(len(age_labels))
    fig, ax = plt.subplots(figsize=(10, 8))
    ax.barh(x - bar_width/2, pop_initial, bar_width, label=f'{year_initial}年', color='blue',
    ax.barh(x + bar_width/2, pop_final, bar_width, label=f'{year_final}年', color='red', alpha
    ax.set_yticks(x)
    ax.set_yticklabels(age_labels, fontproperties=font_small)
    ax.set_xlabel('人口 (千人)', fontproperties=font_small)
    ax.set_title(f'人口年龄结构比较 ({year_initial} vs {year_final})', fontproperties=font)
    ax.legend(prop=font_small, bbox_to_anchor=(1.05, 1), loc='upper left')
    ax.grid(True, axis='x', linestyle='--', alpha=0.7)
    plt.tight_layout()
    plt.savefig(os.path.join(output_dir, f'population_pyramid_{timestamp}.png'), bbox_inches=
    plt.show()

plot_population_pyramid(initial_population, population_history[-1], latest_year, latest_year +

# 三维图 - 人口
fig = plt.figure(figsize=(12, 8))
ax = fig.add_subplot(111, projection='3d')
X, Y = np.meshgrid(range(0, n_years + 5, 5), range(n_age_groups))
```

```
Z = population_history.T
surf = ax.plot_surface(X, Y, Z, cmap='viridis', alpha=0.8)
ax.view_init(elev=20, azim=45)
ax.set_xlabel('年份', fontproperties=font_small)
ax.set_ylabel('年龄组', fontproperties=font_small)
ax.set_zlabel('人口 (千人)', fontproperties=font_small)
ax.set_title(f'人口动态三维图 ({latest_year} - {latest_year + n_years})', fontproperties=font)
ax.set_yticks(range(0, n_age_groups, 2))
ax.set_yticklabels([age_labels[i] for i in range(0, n_age_groups, 2)], fontproperties=font_sma
cbar = fig.colorbar(surf, label='人口 (千人)', pad=0.1)
cbar.set_label('人口 (千人)', fontproperties=font_small)
cbar.ax.tick_params(labelsize=10)
plt.savefig(os.path.join(output_dir, f'population_3d_{timestamp}.png'), bbox_inches='tight')
plt.show()

# 三维图 - 生育率
fig = plt.figure(figsize=(12, 8))
ax = fig.add_subplot(111, projection='3d')
fert_indices = [age_groups.index(age) for age in fert_ages]
X, Y = np.meshgrid(range(0, n_years + 5, 5), fert_indices)
Z = np.zeros((len(fert_indices), n_steps + 1))
for t in range(n_steps + 1):
    for i, age in enumerate(fert_ages):
        popt, _, func = fert_fits[age]
        fert_rate = func(t_future[t], *popt)
        Z[i, t] = max(0, min(fert_rate, 1.5 * fert_df[age].max()))
surf = ax.plot_surface(X, Y, Z, cmap='plasma', alpha=0.8)
ax.view_init(elev=20, azim=45)
ax.set_xlabel('年份', fontproperties=font_small)
ax.set_ylabel('年龄组', fontproperties=font_small)
ax.set_zlabel('生育率 (每千女性)', fontproperties=font_small)
ax.set_title(f'生育率动态三维图 ({latest_year} - {latest_year + n_years})', fontproperties=fon
ax.set_yticks(fert_indices)
```

```
ax.set_yticklabels(fert_ages, fontproperties=font_small)
cbar = fig.colorbar(surf, label='生育率 (每千女性)', pad=0.1)
cbar.set_label('生育率 (每千女性)', fontproperties=font_small)
cbar.ax.tick_params(labelsize=10)
plt.savefig(os.path.join(output_dir, f'fertility_3d_{timestamp}.png'), bbox_inches='tight')
plt.show()

# 三维图 - 存活率
fig = plt.figure(figsize=(12, 8))
ax = fig.add_subplot(111, projection='3d')
surv_indices = list(range(0, 80, 5))
X, Y = np.meshgrid(range(0, n_years + 5, 5), range(len(surv_indices)))
Z = np.zeros((len(surv_indices), n_steps + 1))
for t in range(n_steps + 1):
    for i, age in enumerate(surv_indices):
        if age in surv_fits:
            popt, _, func = surv_fits[age]
            surv_rate = func(t_future[t], *popt)
            Z[i, t] = max(0, min(surv_rate, 0.999))
        else:
            Z[i, t] = survival_rates[i]
surf = ax.plot_surface(X, Y, Z, cmap='inferno', alpha=0.8)
ax.view_init(elev=20, azim=45)
ax.set_xlabel('年份', fontproperties=font_small)
ax.set_ylabel('年龄组', fontproperties=font_small)
ax.set_zlabel('存活率', fontproperties=font_small)
ax.set_title(f'存活率动态三维图 ({latest_year} - {latest_year + n_years})', fontproperties=fon
ax.set_yticks(range(0, len(surv_indices), 2))
ax.set_yticklabels([f'{age}-{age+4}' for age in surv_indices[::2]], fontproperties=font_small
cbar = fig.colorbar(surf, label='存活率', pad=0.1)
cbar.set_label('存活率', fontproperties=font_small)
cbar.ax.tick_params(labelsize=10)
plt.savefig(os.path.join(output_dir, f'survival_3d_{timestamp}.png'), bbox_inches='tight')
```

```
plt.show()
```

10.2 问题二

matlab 代码：斯皮尔曼及热力图绘制

```
% 读取 Excel 数据
filename = "C:\Users\31996\OneDrive\桌面\数模\第二问.xlsx";
data = readtable(filename);

ages = table2array(data(:, 2:4));
indicators = table2array(data(:, 5:7));

rhoMatrix = zeros(3, 3);
pMatrix = zeros(3, 3);
labels = strings(3, 3);

for i = 1:3
    for j = 1:3
        [rho, pval] = corr(ages(:, i), indicators(:, j), 'Type', 'Spearman');
        rhoMatrix(i, j) = rho;
        pMatrix(i, j) = pval;

        if pval < 0.001
            sig = '***';
        elseif pval < 0.05
            sig = '**';
        elseif pval < 0.1
            sig = '*';
        else
            sig = 'ns';
        end
        labels(i, j) = sprintf('%.3f\n(%s)', rho, sig);
```

```
    end
end

figure;
imagesc(rhoMatrix);
colormap(parula);
colorbar;
caxis([-1 0]);

xticks(1:3); xticklabels({'生均经费', '毕业人数', '毛入学率'});
yticks(1:3); yticklabels({'15-19 岁', '20-24 岁', '25-29 岁'});

for i = 1:3
    for j = 1:3
        text(j, i, labels(i, j), 'HorizontalAlignment', 'center', ...
            'FontSize', 11, 'Color', 'k', 'FontWeight', 'bold');
    end
end

title('斯皮尔曼相关系数热力图（含显著性）');
print(gcf, '-dpng', '-r300', 'heatmap2.png');
```

10.3 问题三

matlab 代码：学历贬值研究

```
clc; clear;

%% === 输入数据 ===
r = [6.82, 6.50, 6.15, 5.77, 5.37] / 100;
x = 1 ./ r;
x_ = (x - min(x)) / (max(x) - min(x));
h = [0, 0.25, 0.5417, 1, 0.7917];
```

```
gamma1 = [0.448, 0.628, 0.467, 0.504, 0.478];
gamma2 = 1 - gamma1;
sigma = gamma1 ./ gamma2;

omega1 = [420.5097, 428.097, 471.5658, 489.7422, 511.960];
omega2 = [874, 909, 1076, 1174, 1179];
theta = omega1 ./ omega2;

T = 2020:2024;
X = [sigma', theta'];
num_years = length(T);

%% === 搜索 alpha1 = 0:0.1:1, 无约束回归 ===
alpha_list = 0:0.1:1;
n = length(alpha_list);

beta1_list = zeros(n,1);
beta2_list = zeros(n,1);
errors = zeros(n,1);
V_pred_all = zeros(n, num_years);

for k = 1:n
    alpha1 = alpha_list(k);
    alpha2 = 1 - alpha1;

    V_left = alpha1 * h + alpha2 * x_;
    Y = V_left';

    % 最小二乘 (无约束)
    beta = X \ Y;

    V_pred = X * beta;
    V_pred_all(k,:) = V_pred';
```

```
    errors(k) = norm(V_pred - Y);
    beta1_list(k) = beta(1);
    beta2_list(k) = beta(2);
end

%% === 输出最优解 ===
[~, idx_best] = min(errors);
fprintf('\n=== 最优解 (无约束) ===\n');
fprintf('alpha1 = %.1f, alpha2 = %.1f\n', alpha_list(idx_best), 1 - alpha_list(idx_best));
fprintf('beta1 = %.4f, beta2 = %.4f, error = %.6f\n\n', ...
    beta1_list(idx_best), beta2_list(idx_best), errors(idx_best));

%% === 打印所有结果 ===
fprintf('所有 alpha1 情况下的拟合结果: \n');
fprintf('-----------------------------------------------------------\n');
fprintf('| alpha1 | alpha2 |  beta1    |  beta2    |    error        |\n');
fprintf('-----------------------------------------------------------\n');
for k = 1:n
    fprintf('|  %.1f    |  %.1f    | %8.4f | %8.4f | %12.6f |\n', ...
        alpha_list(k), 1 - alpha_list(k), ...
        beta1_list(k), beta2_list(k), errors(k));
end
fprintf('-----------------------------------------------------------\n');

%% === 绘图 ===
figure;
hold on;
colors = lines(n);
for k = 1:n
    plot(T, V_pred_all(k,:), '-o', 'Color', colors(k,:), ...
        'DisplayName', sprintf('\\alpha_1 = %.1f', alpha_list(k)));
end
xlabel('年份');
```

```
ylabel('预测学历贬值热度 $V(t)$','Interpreter','latex');
title('不同 $\alpha_1$ 权重组合下的预测结果（无约束）','Interpreter','latex');
legend('Location','best');
grid on;
hold off;
saveas(gcf, "C:\Users\31996\OneDrive\桌面\数模\xxx.png")
```