

摘要

目前,我国人口形式呈现显著变化,这种变动趋势深刻影响社会发展的各个领域。本文围绕人口变动及其对高等教育的影响,基于权威数据建立数学模型,给出了未来 20 年人口规模的预测值以及人口变化对高等教育水平影响的定量分析,并从人口变动角度量化了学历贬值问题。

针对问题一的人口预测问题,考虑到人口发展的多重影响因素,本文选取了**经济发展和高等教育发展**两个维度展开分析。首先,本文采用**主成分分析法**确定 **GDP** 为“经济发展指数”、**高等教育毛入学率**为“高等教育发展指数”,并采用 **ARIMA** 模型对未来 20 年的这两个指标进行预测。随后,本文对两个指标进行二次多项式特征拓展,并对历史人口数据进行**多元非线性回归分析**,得到了未来 20 年人口规模不断下降的预测趋势,以及不同时期人口规模预测值。结合预测结果,本文从**生育支持、女性权益和公共服务**三个方面提出了人口发展政策建议。

针对问题二,本文首先选取了四个核心指标作为**高等教育发展水平**的评价标准:高等教育生均教育经费、高等学校师生比、高等学校专任教师数占比和研究生毕业生数占比。考虑到教育指标与时间和人口规模的双重相关性,本文采用**灰度预测模型**对这四个指标进行未来 20 年的预测,并采用**归一法**将各指标数值的范围统一到 $[0,1]$ 。随后,采用**熵权法**计算了各个指标的权重,计算逐年各指标得分并合成得到**逐年教育水平总得分**。最后,本文通过四个核心指标和高等教育综合得分的动态变化趋势,得出结论:我国高等教育发展水平将在未来 20 年突破人口规模的限制,保持显著发展态势。

针对问题三,我们探究了**高学历毕业生的劳动力市场供给增速与需求增速的差值** ($S_t - D_t$),和**本科以上学历毕业生平均工资比** (Y_1) 及**失业率** (Y_2) 的关系。本文以归一化后的年份数据为自变量,以毕业生人数、第三产业占 GDP 比重、 Y_1 或 Y_2 为因变量作两张**趋势图**,得出结论:近年来 Y_1 和 Y_2 的不利变化趋势,与 S_t 显著快于 D_t 的时期吻合。基于上述观察,本文建立了两个经济模型,定量分析**供需失衡对学历贬值的影响**。模型一通过线性回归法,建立了 Y_1 和 $(S_t - D_t)$ 的关系,结果显示高学历毕业生劳动力供需增速差异的升高,导致其相对工资优势下降。模型二需要引入第二个自变量,建立 Y_2 和 $(S_t - D_t)$ 、GDP 增长率之间的关系。结果显示,高学历毕业生劳动力供需增速差异的升高对失业率有显著正向影响,同时,GDP 增长率的提高对失业率有抑制作用。

最后,本文对所建立的模型进行了灵敏度分析和优缺点评价。

关键字: ARIMA 模型 多元非线性回归 灰度预测模型 熵权法

一、问题重述

1.1 问题背景

当前，我国人口形势出现了显著变化，出生率持续走低，人口发展呈现老龄化、少子化等趋势性特征。2023 年统计数据显示，我国全年出生人口为 902 万。而 2024 年出现小幅回升，新出生人口增至 954 万。这种人口变动趋势深刻影响社会发展的各个领域，引发了社会各界对教育、经济和社会可持续发展的广泛关注。

自改革开放以来，我国高等教育事业实现了跨越式发展。1981 年《学位条例》^[2] 的颁布实施，标志着我国开始确立现代学位制度。经过四十余年的发展，我国已建成规模成熟的高等教育体系。然而，在人口结构转型的背景下，高等教育发展正面临新的机遇和挑战。

1.2 问题重述

问题一：基于权威数据，构建预测模型，分析未来不同时期的人口规模变化，并提出维持人口永续发展的建议；

问题二：建立量化分析模型，解析人口因素对高等教育发展的影响；

问题三：建立量化分析模型，解析人口变动的背景下学历价值稀释的问题。

二、问题分析

2.1 问题一

问题一要求我们构建科学的数学模型，预测我国未来不同时期的人口规模变化，并提出促进人口长期健康发展的政策建议。首先，在指标选取方面，基于人口发展的多维度影响因素，我们重点考察经济发展和高等教育发展两个重要维度。为了更准确地刻画这两个维度的影响，需要采用主成分分析法进行降维处理，分别得到一个“经济发展指数”和“高等教育发展指数”。

在建模方法上，需要采用分步预测的方式。首先基于统计年鉴中的历史数据，对经济发展指数和高等教育发展指数进行时间序列分析，获得未来 20 年的预测值。随后，运用多元非线性回归的方法，建立人口规模与这两个综合指数之间的关系模型。最后，将预测得到的经济和教育指数代入模型，便可得到未来 20 年的人口规模预测值。

2.2 问题二

基于上问预测的人口规模数据，问题二要求我们建立量化分析模型，解析人口因素对高等教育发展的影响。高等教育发展水平涉及多个维度，因此，首先需要确定评价高等教育的核心指标。结合现有权威数据，我们选取高等教育生均教育经费、高等学校师生比、高等学校专任教师数占比和研究生毕业生数占比四个关键指标，综合反映高等教育的发展水平。

得到各指标的历史数据后，需要预测未来 20 年各指标的变化趋势。本研究需要采用多变量灰度预测模型，综合考虑时间和人口因素对各指标的影响。同时，需要对各指标的预测数据进行归一化处理，确保后续计算的公平性与可比性。

各教育指标的权重需要通过熵权法确定，体现各指标对高等教育综合得分的贡献。计算权重后，需要进一步计算未来各年单个指标的教育得分，并合成得到各年的教育总得分。为便于进行比较分析，还需要将总得分归一化到 60-100 分区间。最后，可以通过综合得分的变化趋势，定量分析人口变化对高等教育的影响。

2.3 问题三

学历贬值的规律在于学历的客体与主体间供需不平衡^[7]。为了深入分析这一问题，我们从收入和就业两个维度对学历贬值进行量化。从收入角度来看，学历贬值表现为高学历层次的毕业生相对于其他低学历或社会平均工资的相对收入优势下降；从就业角度来看，则体现为特定学历层次的毕业生失业率上升。

在明确了学历贬值的定义后，我们需要探究人口变化的影响。从供给端来看，人口结构的变化直接决定了高学历劳动力市场的供给规模；从需求端来看，经济增长和产业结构升级决定了市场对高学历人才的需求。核心问题在于，高学历劳动力的供给增速 (S) 是否超过了需求增速 (D)。如果供给增长快于需求增长，就会导致学历贬值。

因此，从人口规模变化角度分析学历贬值的关键，在于探究高学历劳动力供给增速与需求增速的差异 ($S - D$)，和高等教育人才相对工资优势及失业率的关系。

为了定量分析这一问题，需要搜集多方面的数据，包括本科及以上学历毕业生的平均工资与社会平均工资的比值以及本科及以上学历人群的失业率。人口方面需要关注研究生毕业生人数，用于衡量新增高学历劳动力的供给。此外，经济数据如第三产业占比，能够反映劳动力市场的需求变化。

在分析方法上，可以通过趋势图直观展示学历贬值与人口变化的关系。同时，可以通过线性回归模型，直观分析供给增速与需求增速的差异对相对收入优势及失业率的影响。

三、模型假设

本文提出以下合理假设：

- 假设回归模型中扰动项服从独立的正态分布；
- 不考虑大规模人口迁徙、自然灾害等突发性因素对人口和教育指标的影响；
- 假设高等教育的评价指标间不存在强相关性
- 不考虑政策干预、技术进步等外界因素对市场机制的影响

四、符号说明

符号	意义
X_{ij}	第 i 年的第 j 项教育指标数据
Z_{ij}	归一化后第 i 年的第 j 项教育指标数据
$\max(X_j)$	第 j 项教育指标在 20 年预测期内的最大值
$\min(X_j)$	第 j 项教育指标在 20 年预测期内的最小值
e_j	第 j 项教育指标的信息熵
g_j	第 j 项教育指标的差异系数
w_j	第 j 项教育指标的权重
M_{ij}	第 i 年的第 j 项教育指标的得分
M_j	第 i 年的教育总得分

五、模型的建立与求解

5.1 问题一模型的建立与求解

为了预测我国未来 20 年的人口规模变化，我们考察经济发展和高等教育发展两个维度，并在统计年鉴中获取 1995-2023 年的相关统计数据。

5.1.1 基础指标的主成分分析与未来预测

在本题中，我们首先对经济发展和高等教育发展两个维度的指标进行主成分分析。结果显示，国内生产总值（GDP）在经济维度的载荷量远高于其他经济指标，毛入学率

在高等教育维度的载荷量显著其他高等教育指标，最终选择 GDP 作为“经济发展指数”，选择毛入学率作为“高等教育发展指数”。

整合统计年鉴中 GDP 和高等教育毛入学率的数据，我们分别对二者进行时间序列分析，采用 ARIMA 模型对未来 20 年的这两个指数进行预测。

表 1 GDP 和高等教育毛入学率的预测结果

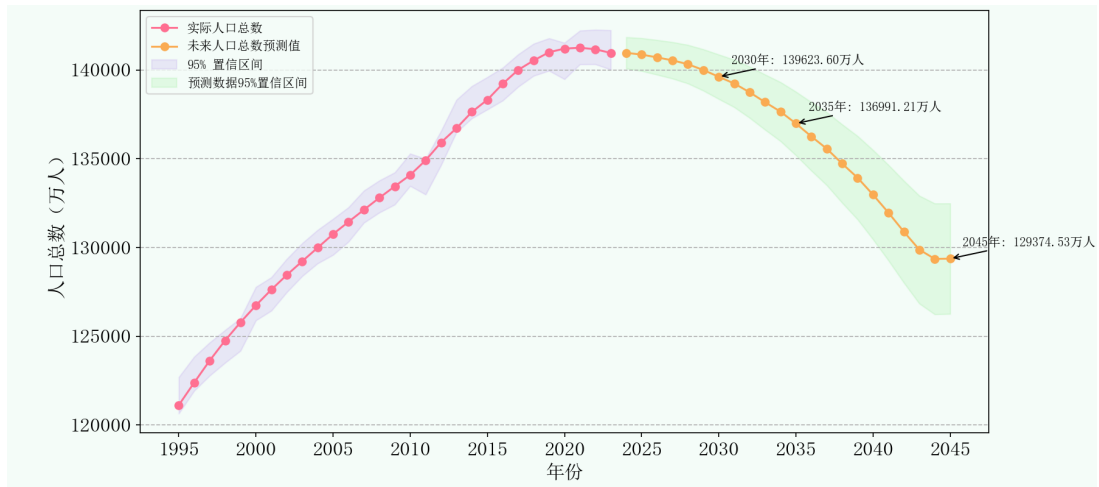
年份	GDP（亿元）	高等教育毛入学率（%）	年份	GDP（亿元）	高等教育毛入学率（%）
2024	1141306	58.8	2035	1628243	80.4
2025	1185573	60.8	2036	1672510	82.4
2026	1229840	62.8	2037	1716777	84.3
2027	1274107	64.7	2038	1761044	86.3
2028	1318374	66.6	2039	1805311	88.2
2029	1362641	68.6	2040	1849578	90.2
2030	1406908	70.6	2041	1893845	92.2
2031	1451175	72.5	2042	1938112	94.2
2032	1495442	74.5	2043	1982379	96.1
2033	1539709	76.5	2044	2015243	97.3
2034	1583976	78.4	2045	2076458	98.6

5.1.2 人口规模的未来预测

选取主要影响人口的两个变量，即 GDP（经济发展指数）和高等教育毛入学率（高等教育发展指数），并进行二次多项式特征扩展。接下来，用拓展后的特征对历史人口数据进行多元非线性回归建模，建立人口预测模型，实现对未来 5 年、10 年、20 年的人口规模预测。

如图??预测结果显示，未来 20 年我国人口规模将呈现逐渐下降的趋势，预计到 2030 年人口总数将降至 13.962 亿人左右，到 2035 年将降至 13.699 亿人，到 2045 年将降至 12.937 亿人。

图 1 人口规模预测与置信区间

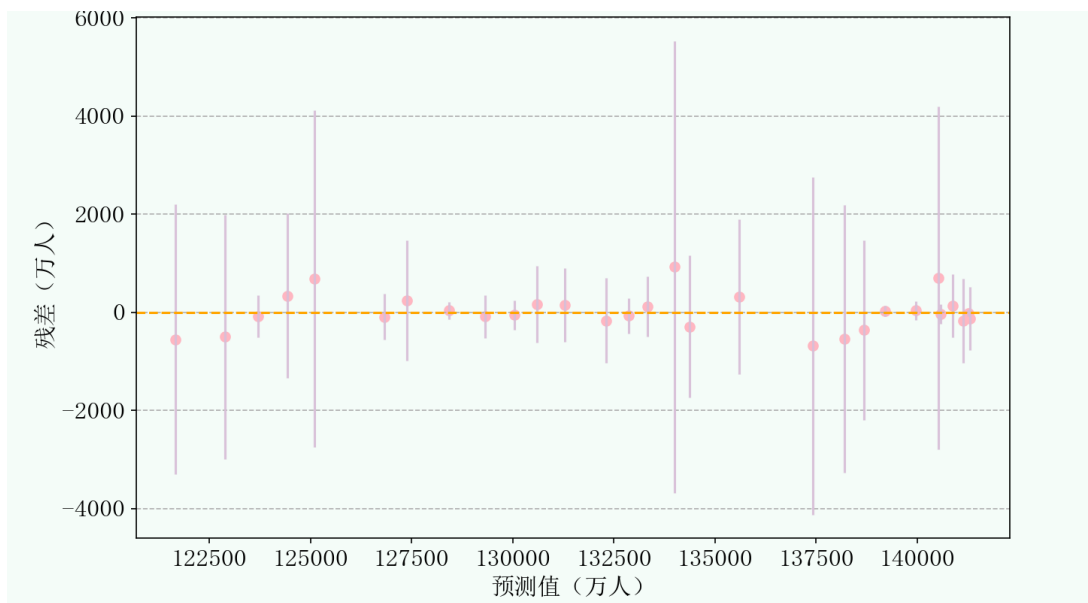


5.1.3 模型诊断与进阶分析

为量化预测不确定性，我们基于 t 分布的置信区间计算方法，得到了历史与未来人口数据的置信区间，并在时间序列预测曲线上叠加绘制。如图??所示，置信区间范围随时间扩大，符合外推预测误差累积的规律。

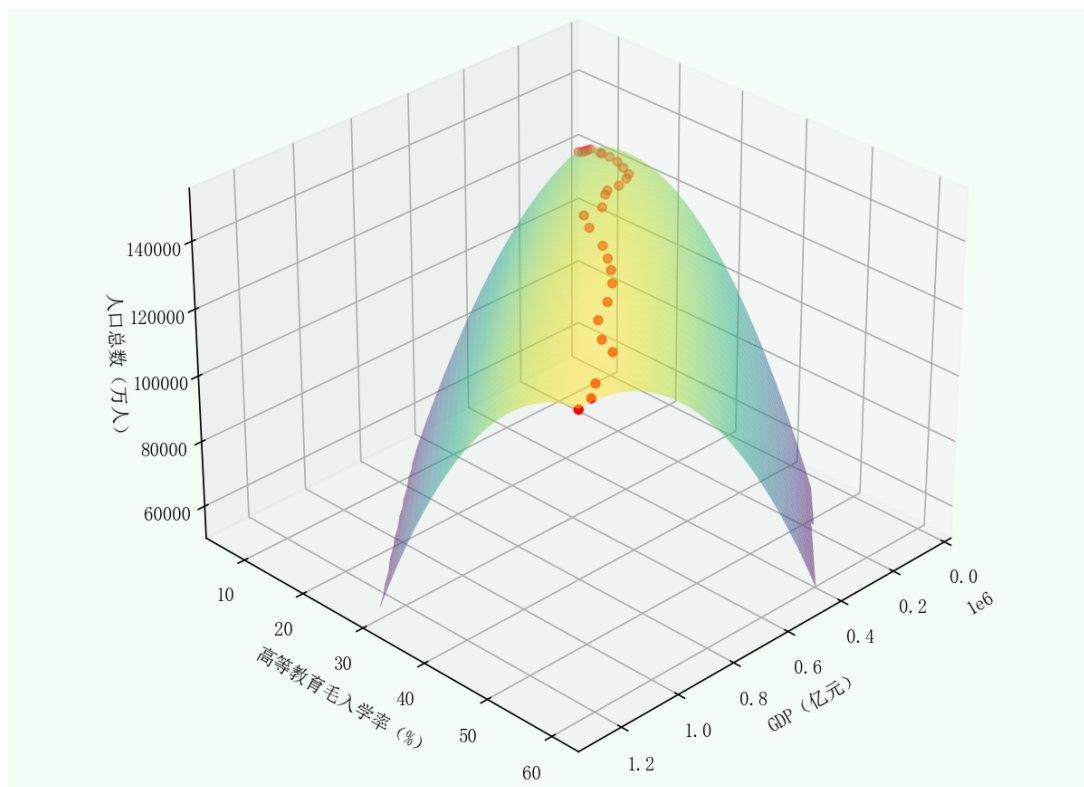
同时，我们计算了历史数据中人口总数实际值与预测值的偏差，并绘制了残差图，以诊断回归模型的拟合效果。如图??所示，绝大多数数据点的残差分布在零基准线附近，表明二次多项式回归模型对人口总数的拟合效果较好。这种随机分布的残差模式验证了模型假设的合理性，说明当前模型已较好地捕捉了 GDP 和高等教育毛入学率与人口总数之间的非线性关系。

图 2 人口规模预测残差图



为体现 GDP（亿元）、高等教育毛入学率（%）与人口总数（万人）之间的非线性关系，我们绘制了如图??所示的三维曲面图。曲面呈现平滑的非线性趋势，表明 GDP 和毛入学率对人口总数的联合影响具有复杂性。此外，图中红色散点为实际观测值，大部分散点紧邻曲面，进一步验证模型有效性。

图 3 多元非线性回归模型人口规模预测三维曲面图



5.1.4 政策建议

为应对人口规模下降的趋势，实现人口长期均衡发展，我们提出以下政策建议：

1、生育支持政策：一方面，可以扩大现有的生育津贴和育儿补贴等直接经济支持^[7]。此外，可考虑实施递进式补贴政策，即根据生育孩次逐步提高补贴标准，同时对低收入家庭给予额外补助。同时，可实行税收抵免政策，例如按照子女数量调整个人所得税起征点，切实降低家庭生育成本。

2、女性职业权益：首先，企业应优化生育假期制度，完善企业孕期生育津贴、交通津贴等经济补贴制度^[9]。同时，可推行“弹性返岗”制度，允许女职工分阶段恢复全职工作。针对企业，政府可推行“生育成本社会化分担”机制，均衡分摊产假成本。在职业发展方面，建议规定企业在员工生育期间保留其岗位职级，并将产假时间从晋升考评年限中扣除，保障女性在生育期间的职业发展权益。

3、公共服务供给：在实践中着力构建普惠性、专业性、共担性和系统性的托育服务。同时，要建立托育服务人员职业资格认证体系，提高从业人员待遇，从而促进托育

服务的发展。此外，应积极打造“孕-产-育”全程服务链，包括孕期保健、产后康复、早教指导等一体化服务，真正实现儿童优先、家庭为本的社会支持网络，促进生育友好型社会建设^[7]。

5.2 问题二模型的建立与求解

为了定量分析人口规模对高等教育发展的影响，本研究构建了一个系统的分析框架，通过灰度预测模型和熵权法相结合的方式，实现了从数据预测到综合评价的全过程建模。

5.2.1 高等教育发展指标的选取与预测

首先，我们选取了四个核心指标作为高等教育发展水平的评价标准：高等教育生均教育经费、高等学校师生比、高等学校专任教师数占比和研究生毕业生数占比，这些指标从资源投入、师资配置、人才培养等维度全面反映了高等教育的发展状况。

考虑到教育指标与时间和人口规模的双重相关性，我们采用灰度预测模型对这四个指标进行未来 20 年的预测。(补充描述)

随后，对预测结果进行标准化处理，采用最大-最小归一化方法，将各指标的数值范围统一到 [0,1] 区间，计算公式如下：

$$Z_{ij} = \frac{X_{ij} - \min(X_j)}{\max(X_j) - \min(X_j)} \quad (1)$$

其中 Z_{ij} 为归一化后第 i 年的第 j 项指标数据， X_{ij} 为原始指标数据，第 j 项指标在 20 年预测期内的最大值和最小值分别为 $\max(X_j)$ 和 $\min(X_j)$ 。指标预测与归一化结果见附录??。

5.2.2 高等教育水平的量化分析

为客观反映各指标在高等教育发展中的相对重要性，我们采用熵权法计算各指标的权重。首先，计算第 j 项指标的信息熵 e_j ，公式如下：

$$e_j = -\frac{1}{\ln n} \sum_{i=1}^n \left(\frac{Z_{ij}}{\sum_{i=1}^n Z_{ij}} \ln \frac{Z_{ij}}{\sum_{i=1}^n Z_{ij}} \right) \quad (2)$$

其中 n 为预测的指标数量 ($n=22$)。熵值越低，说明该指标的离散度越高，信息量越集中，对高等教育发展贡献越大。

接下来，计算第 j 项指标的权重 w_j ，利用公式：

$$w_j = \frac{g_j}{\sum_{j=1}^n g_j} \quad (3)$$

其中 $g_j = 1 - e_j$ 为第 j 项指标的差异系数。各指标信息熵、差异系数和权重计算结果如表??所示。

表 2 各教育指标信息熵、差异系数和权重计算结果

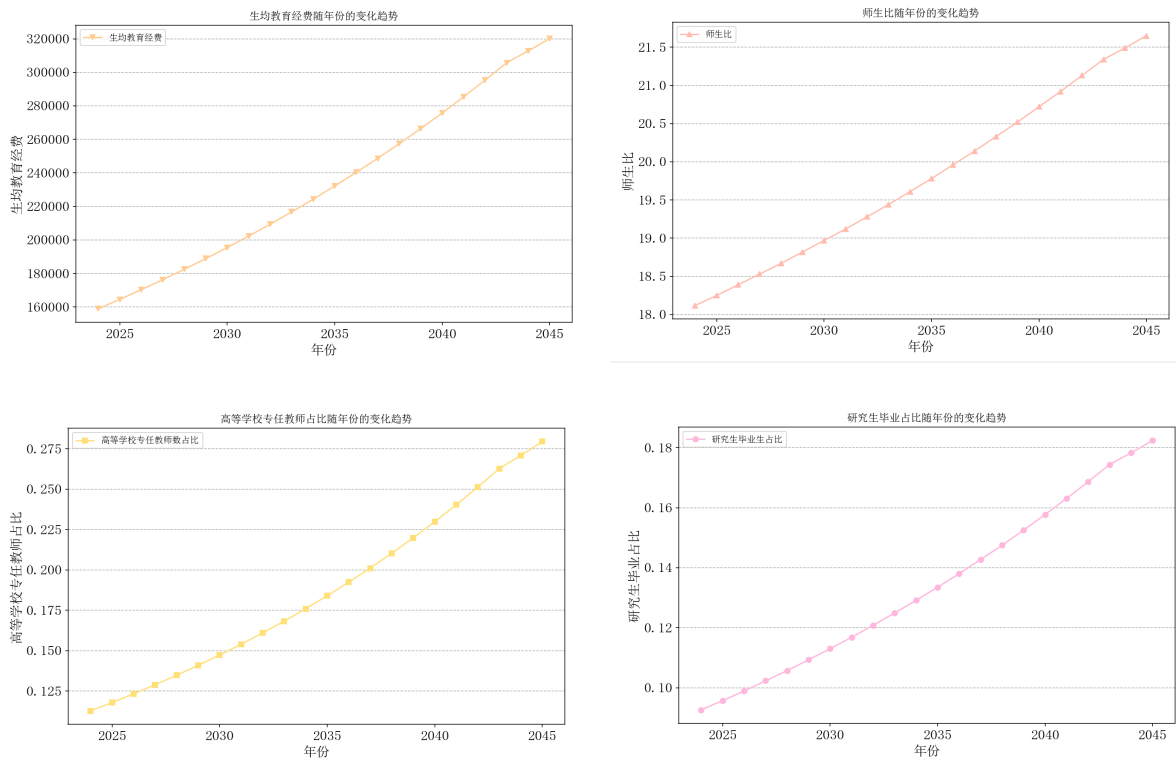
指标	信息熵	差异系数	权重
生均教育经费	0.9930	0.0070	0.2566
高等学校师生比	0.9877	0.0123	0.4523
高等学校专任教师数占比	0.9995	0.0005	0.0181
研究生毕业生数占比	0.9926	0.0074	0.2730

最后，计算逐年各指标得分 $M_{ij} = 100Z_{ij} \cdot w_j$ ，并合成得到 2025-2045 逐年总得分 $M_i = \sum_{j=1}^4 M_{ij}$ ，最终结果如图??所示。

5.2.3 人口因素对高等教育发展的影响分析

首先，通过四个核心指标的动态变化趋势，可以深入揭示人口变化对高等教育系统的多维影响。数据分析显示，生均教育经费、高等学校师生比、高等学校专任教师数占比和研究生毕业生数占比四项指标，均与人口变化存在显著负相关，如图??所示。人口每下降 1%，生均教育经费因资源集中效应将提升 0.61 个百分点，师生比将改善 0.53 个百分点。当人口总数从 2030 年的 13.96 亿下降至 2045 年的 12.94 亿时（降幅 7.3%）时，专任教师占比从 14.73% 大幅提升至 27.94%，研究生占比从 11.30% 稳步增长至 18.24%，这直接反映了我国高等教育人口规模将持续快速增长，有力推进以人口高质量发展支撑中国式现代化^[9]。

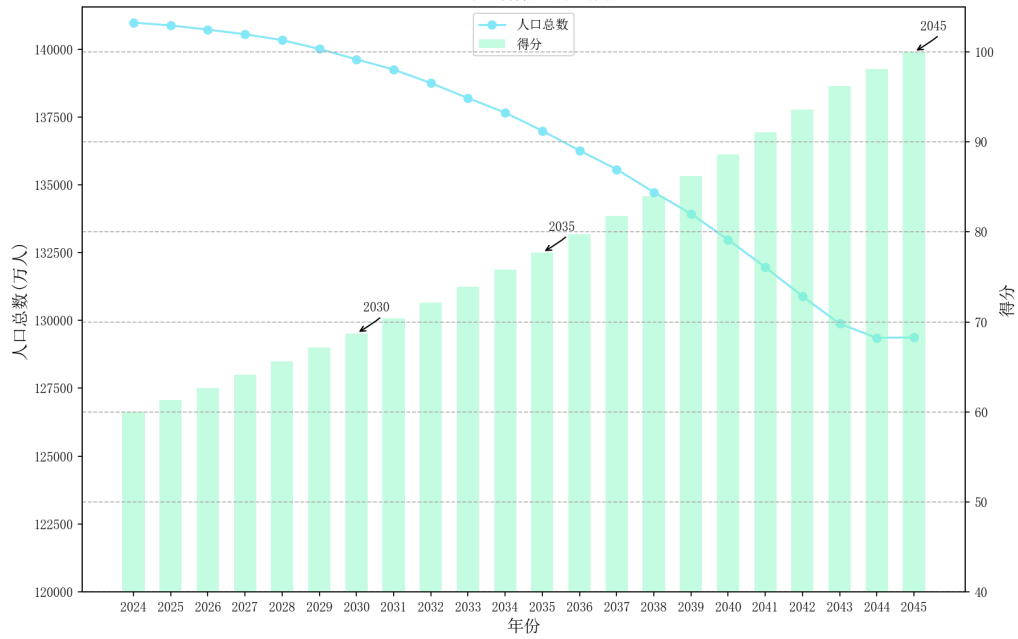
图 4 核心指标随年份的变化趋势



其次，总体分析可得，在人口总量持续减少（2024-2045 年）与社会经济持续发展的双重作用下，我国高等教育综合得分呈现明显的上升趋势，如图??所示。

两条曲线形成的“剪刀差”图形生动地体现了我国高等教育质量的显著提升，具体来看，得分曲线的上升过程可以分为三个阶段：2024-2030 年是平稳上升期，每年增长幅度较为均匀；2031-2040 年进入加速上升期，这段时间的增长斜率相对增大；2041 年后虽然增速有所放缓，但仍保持稳定上升态势。

图5 人口总数和高等教育得分随年份的变化



综上所述，我国高等教育发展水平将在未来 20 年突破人口规模的限制，保持显著发展态势。这充分展现了高等教育系统在人口转型背景下的强大发展韧性，推进我国在人口新常态下实现教育现代化。

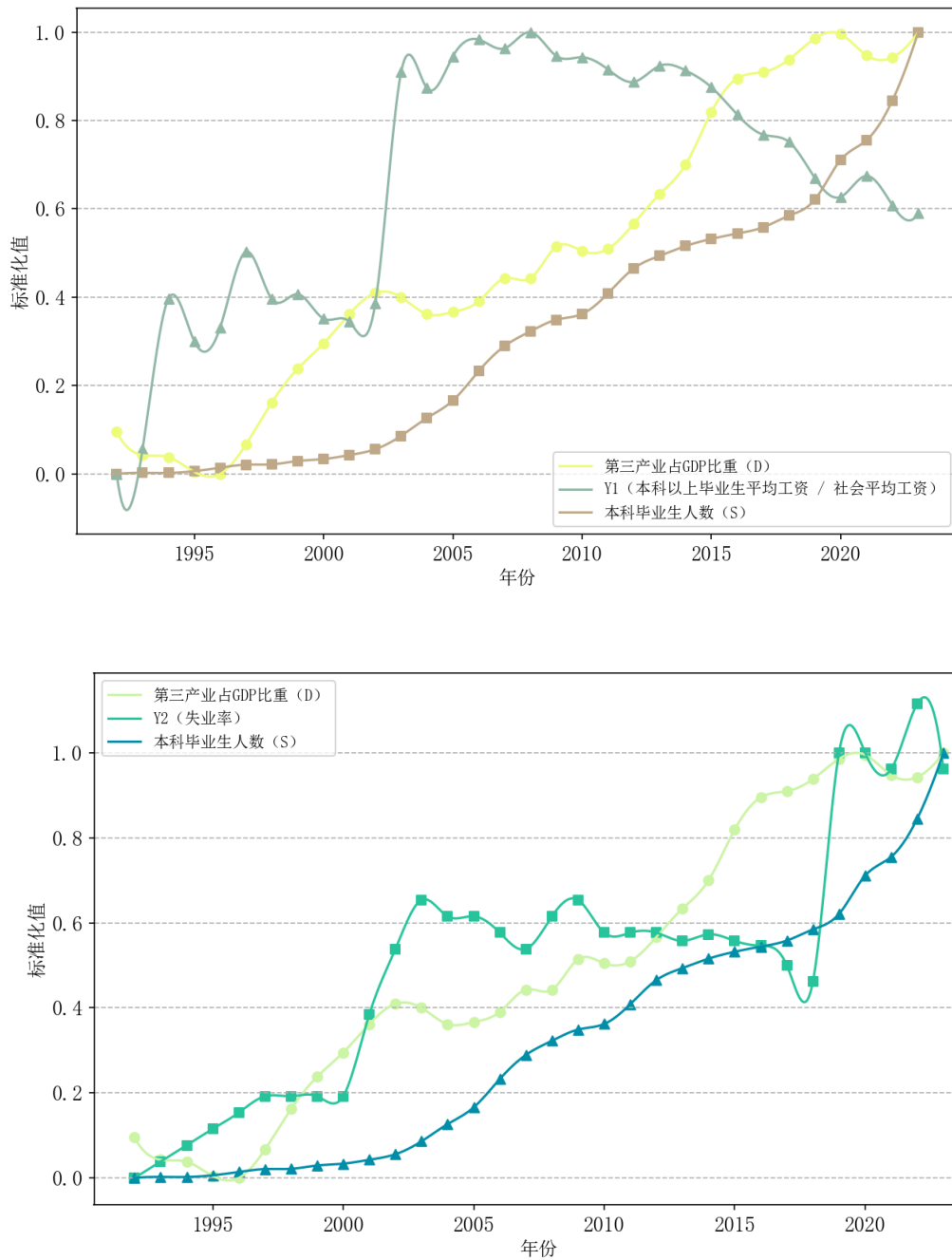
5.3 问题三模型的建立与求解

5.3.1 学历贬值问题中定量分析模型的建立

在数据处理上，我们搜集了本科及以上学历毕业生的平均工资与社会平均工资的比值、本科及以上学历人群的失业率、研究生毕业生人数和第三产业占 GDP 比重的权威数据，并进行标准化处理。

本研究首先通过趋势图直观展示学历贬值与人口变化的关系。在趋势图中，以因变量毕业生人数 (S) 代表高学历劳动力的市场供给，以因变量第三产业占 GDP 比重 (D) 代表市场对高学历人才的需求，二者能够体现人口变化的影响。左图中，以本科以上学历毕业生平均工资比 (Y_1) 作为因变量，其变化能够反映学历贬值的收入维度。右图中，将 Y_1 替换为 Y_2 ，即本科以上学历人群的失业率，从就业维度反映学历贬值。

图 6 1992-2023 年三个标准化值折线图



如图??中两张趋势图所示：2017-2023 年间， S 的增长显著快于 D 的增长，与此同时， Y_1 呈下降趋势， Y_2 呈上升趋势。2002-2005 年间，当 S 的增长显著快于 D 的增长时， Y_1 反而呈上升趋势，这是因为二十一世纪初各个行业均在飞速发展，学历贬值现象尚不明显。综合分析可得，近年来 Y_1 和 Y_2 的不利变化趋势，与 S 增长显著快于 D 增长的时期吻合。

基于上述观察，我们建立了两个经济模型，定量分析供需失衡对学历贬值的影响。

1、模型一：学历贬值在收入维度的表现

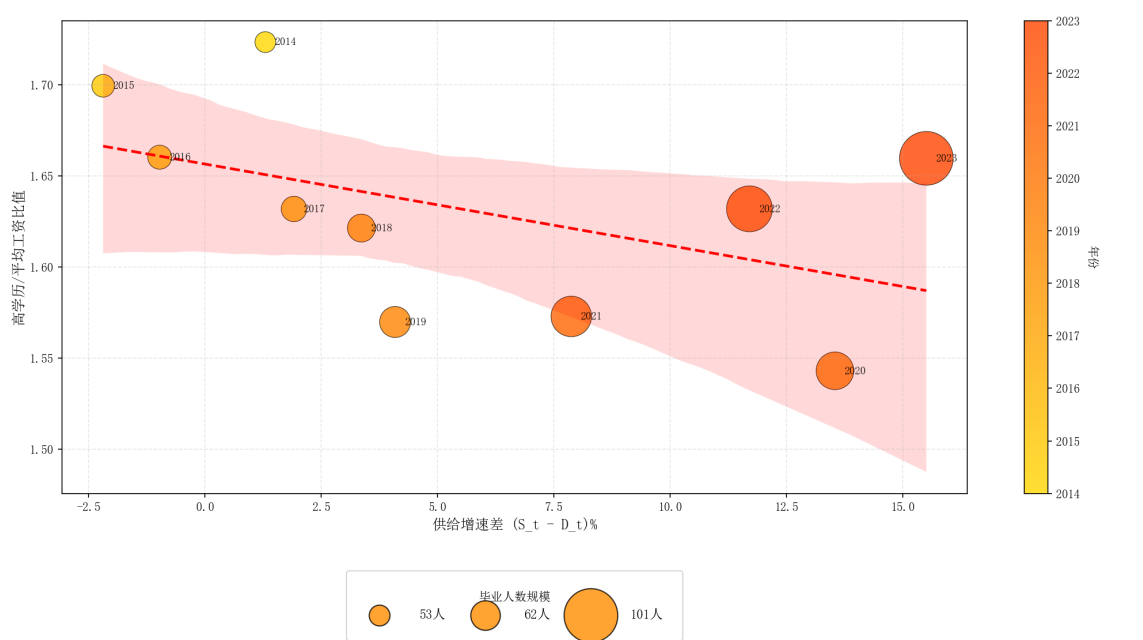
模型一通过线性回归法，利用 2014-2023 年的相关数据，建立高学历毕业生相对工资比和供给速度差之间的关系：

$$Y_{1_t} = \beta_0 + \beta_1 \cdot (S_t - D_t) + \varepsilon_t \quad (4)$$

其中，因变量 Y_{1_t} 表示第 t 年的工资比值。 S_t 表示第 t 年本科毕业生人数增长率 (%)，代表供给增长速度； D_t 表示第 t 年第三产业占 GDP 比重增长率 (%)，代表需求增长速度。核心自变量 $S_t - D_t$ 表示第 t 年高学历劳动力的供给速度超过需求速度的程度。 β_0 为常数项， ε_t 为随机误差项。

模型计算结果如图??所示。 $\beta_1 < 0$ 且统计显著，证明了高学历毕业生劳动力供需增速差异的升高，导致其相对工资优势下降。

图 7 高学历毕业生工资比值与供给增速差的关系



2、模型二：学历贬值在就业维度的表现

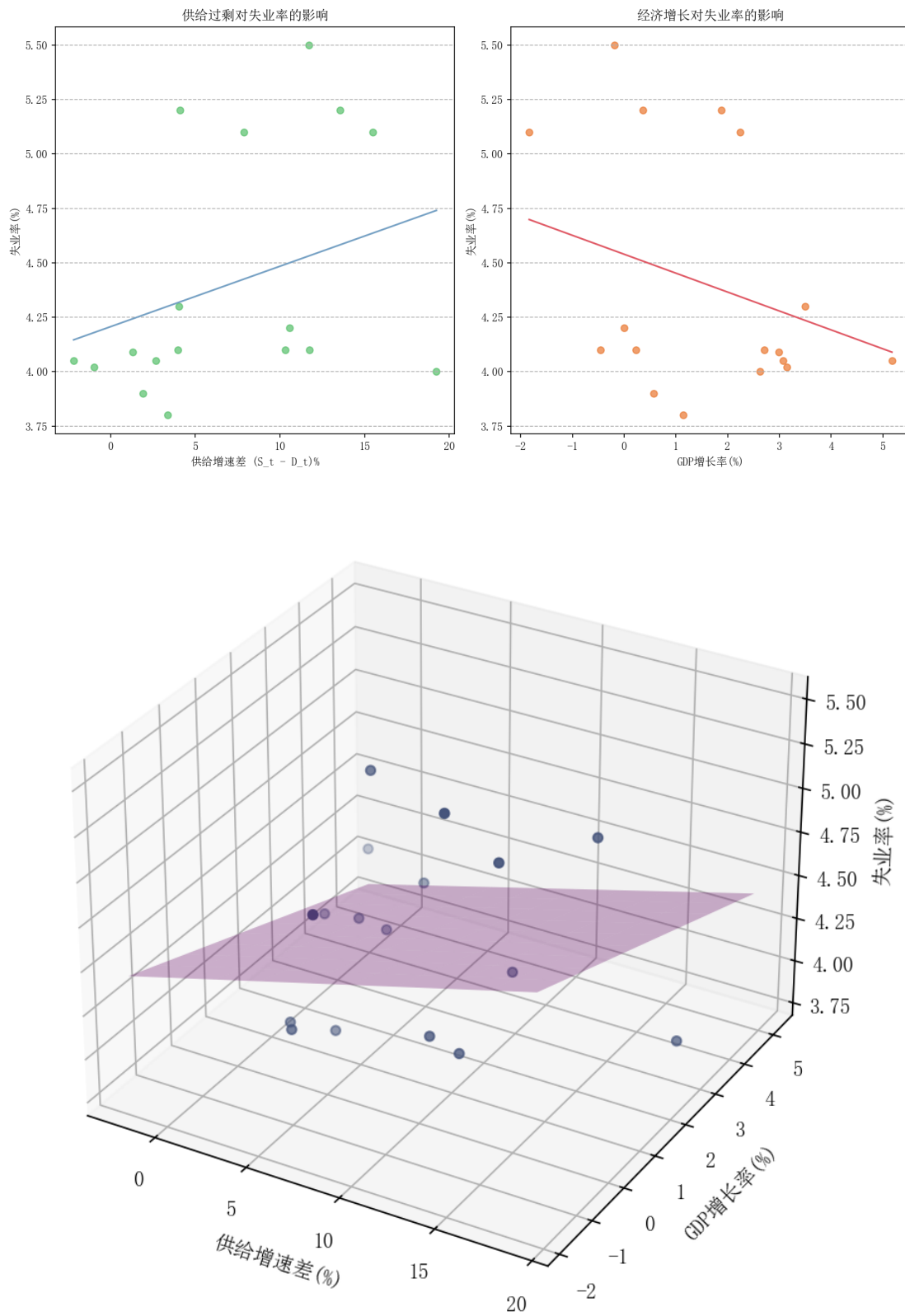
由于失业率本身会受到经济周期的影响，故而模型二需要引入第二个自变量，建立高学历毕业生失业率和供给速度差、GDP 增长率之间的关系：

$$Y_{2_t} = \beta_0 + \beta_1 \cdot (S_t - D_t) + \beta_2 \cdot G_t + \varepsilon_t \quad (5)$$

其中，因变量 Y_{2_t} 表示第 t 年的失业率，新增自变量 G_t 表示 GDP 增长率。

模型计算结果如图??所示。一方面，高学历毕业生劳动力供需增速差异的升高，对失业率有显著正向影响；另一方面，GDP 增长率的提高对失业率有抑制作用。这一结果不仅验证了供需失衡对就业市场的负面影响，同时表明，经济增长本身是降低失业率的关键因素，揭示了经济周期对失业率的调节作用。

图8 供给过剩和经济增长对失业率的影响



5.3.2 人口因素对学历贬值影响的总结分析

首先,本研究通过系统分析揭示了学历贬值现象的形成机制与表现特征。从供给端来看,80年代末至90年代初的人口出生高峰为后续高等教育扩张提供了基础人口规模。2003年启动的高校扩招政策使得这批人口在2010-2018年间集中转化为本科毕业生供给。这种人口结构驱动的供给激增具有典型的周期性特征,其影响在劳动力市场上持续释放,构成了学历贬值的供给端基础。

需求端的分析揭示了深层次的结构性矛盾。虽然我国第三产业占比在2017-2023总体呈现上升趋势,但其增速仍然显著落后于本科毕业生供给增速。这体现了劳动力市场供需的长期结构性失衡,进而导致学历在劳动力市场的贬值。

学历贬值具有双重表现:从收入维度上看,高学历毕业生劳动力供需增速差异的升高,导致了其相对工资优势下降;在就业维度,本科毕业生供需失衡是导致其失业率上升的重要因素。综合分析,人口结构驱动的供给冲击(S),叠加需求(D)未能同步跟上,导致了学历在劳动力市场的双重贬值——收入水平下降和就业难度加大。

六、模型的分析

七、模型的评价

参考文献

- [1] 中华人民共和国教育部. 中华人民共和国学位条例. 国家法律法规数据库, 1980. 1980年2月12日第五届全国人民代表大会常务委员会第十三次会议通过, 2004年8月28日第十届全国人民代表大会常务委员会第十一次会议修正.
- [2] 万春林, 朱启飏, 闫雪凌. 生育支持、延迟退休与老年福利. 系统工程理论与实践, pages 1–22, 2025. [J/OL]. 访问日期: 2025-07-25.
- [3] 马家兴, 姚云云. 生育友好型社会构建: 现实逻辑、挑战及路径探讨——基于三孩政策的思考. 黑龙江科学, 16(11):1–3, 11, 2025.
- [4] 胡玉萍, 陈德云, 古之叶. 生育友好型社会背景下托育服务的多重价值与实践路径. 新视野, (03):99–108, 2025.

附录 A 源代码

1.1 问题一源代码

1.1.1 时间序列分析与预测

```
1 import pandas as pd
2 from statsmodels.tsa.arima.model import ARIMA

3 excel_file = pd.ExcelFile('国赛校赛第一问数据 2.0.xlsx')
4 df = excel_file.parse('Sheet1')

5 df.set_index('年份', inplace=True)

6 def arima_forecast(data, order=(1, 1, 1), steps=22):
7     model = ARIMA(data, order=order)
8     model_fit = model.fit()
9     return model_fit.forecast(steps=steps)

10 # 预测高等教育毛入学率
11 higher_education_forecast = arima_forecast(df['高等教育毛入学率 (%)'])

12 # 预测 GDP
13 gdp_forecast = arima_forecast(df['GDP (亿元)'])
```

1.1.2 人口预测

```
1 import pandas as pd
2 from sklearn.preprocessing import PolynomialFeatures
3 from sklearn.linear_model import LinearRegression
4 from sklearn.metrics import mean_squared_error, r2_score
5 import matplotlib.pyplot as plt
6 from scipy import stats
7 import numpy as np
8 from matplotlib.font_manager import FontProperties
9 plt.rcParams['figure.facecolor']='#F4FCF8'
10 plt.rcParams['axes.facecolor']='#F4FCF8'
11 # 设置中文字体
12 font = FontProperties(fname=r"C:\Windows\Fonts\simsun.ttc", size=12)
13 plt.rcParams['font.family'] = font.get_name()
14 plt.rcParams['axes.unicode_minus'] = False
15 excel_file_1 = pd.ExcelFile('国赛校赛第一问数据 2.0.xlsx')
16 sheet_names_1 = excel_file_1.sheet_names
17 for sheet_name in sheet_names_1:
18     df = excel_file_1.parse(sheet_name)
19     #print(f'sheet 表名为 {sheet_name} 的基本信息: ')
```



```

20 df.info()
21 rows, columns = df.shape
22 # if rows < 100 and columns < 20:
23 #     #print(f'sheet 表名为 {sheet_name} 的全部内容信息: ')
24 #     #print(df.to_csv(sep='\t', na_rep='nan'))
25 # else:
26 #     # 长表数据查看数据前几行信息
27 #     #print(f'sheet 表名为 {sheet_name} 的前几行内容信息: ')
28 #     #print(df.head().to_csv(sep='\t', na_rep='nan'))

29 excel_file_2 = pd.ExcelFile('预测 GDP 入学.xlsx')
30 sheet_names_2 = excel_file_2.sheet_names
31 for sheet_name in sheet_names_2:
32     df = excel_file_2.parse(sheet_name)
33     #print(f'sheet 表名为 {sheet_name} 的基本信息: ')
34     df.info()

35     rows, columns = df.shape

36     # if rows < 100 and columns < 20:
37     #
38     #     print(f'sheet 表名为 {sheet_name} 的全部内容信息: ')
39     #     print(df.to_csv(sep='\t', na_rep='nan'))
40     # else:
41     #
42     #     print(f'sheet 表名为 {sheet_name} 的前几行内容信息: ')
43     #     print(df.head().to_csv(sep='\t', na_rep='nan'))

44 df1 = excel_file_1.parse('Sheet1')
45 df2 = excel_file_2.parse('Sheet1')
46 poly = PolynomialFeatures(degree=2)
47 X_poly = poly.fit_transform(df1[['GDP (亿元)', '高等教育毛入学率 (%)']])

48 model = LinearRegression()
49 model.fit(X_poly, df1['人口总数 (万人)'])

50 future_X_poly = poly.transform(df2[['GDP (亿元)', '高等教育毛入学率 (%)']])
51 future_population = model.predict(future_X_poly)

52 df2['人口总数 (万人) 预测值'] = future_population

53 y_pred = model.predict(X_poly)

54 residuals = df1['人口总数 (万人)'] - y_pred

55 # 绘制残差图

```

```

56 plt.figure(figsize=(10, 6))
57 for x, y in zip(y_pred, residuals):
58     half_length = abs(y) * 5
59     plt.vlines(x, y - half_length, y + half_length, color='#D8BFD8',
60               ↪ alpha=0.95)
61 plt.scatter(y_pred, residuals, marker='o', color= '#FFB6C1')
62 plt.axhline(y=0, color='#FFA500', linestyle='--')
63 plt.xlabel('预测值 (万人)', fontsize=15)
64 #plt.xticks(rotation=45)
65 plt.ylabel('残差 (万人)', fontsize=15)
66 plt.xticks(fontsize=16)
67 plt.yticks(fontsize=16)
68 plt.title('人口总数预测残差图')
69 plt.grid(axis='y', linestyle='--')
70 plt.show()

71 mse = mean_squared_error(df1['人口总数 (万人)'], y_pred)
72 r2 = r2_score(df1['人口总数 (万人)'], y_pred)

73 # 计算置信区间
74 n = len(df1)
75 p = X_poly.shape[1]
76 dof = n - p - 1
77 t = stats.t.ppf(0.975, dof)
78 residuals = df1['人口总数 (万人)'] - y_pred
79 std_error = np.sqrt(np.sum(residuals**2) / dof)
80 margin_of_error = t * std_error * np.sqrt(1 + np.diag(X_poly @
81               ↪ np.linalg.inv(X_poly.T @ X_poly) @ X_poly.T))
82 lower_bound = y_pred - margin_of_error
83 upper_bound = y_pred + margin_of_error

84 XtX_inv = np.linalg.inv(X_poly.T @ X_poly)
85 se_future = np.sqrt(mse * (1 + np.diag(future_X_poly @ XtX_inv @
86               ↪ future_X_poly.T)))
87 lower_bound_future = future_population - t * se_future
88 upper_bound_future = future_population + t * se_future

89 # 绘制预测结果和置信区间
90 plt.figure(figsize=(12, 6))
91 plt.plot(df1['年份'], df1['人口总数 (万人)'], color='#ff6f91', label='实际
92               ↪ 人口总数', marker='o')
93 plt.plot(df2['年份'], df2['人口总数 (万人) 预测值'], color='#FAAB52',
94               ↪ label='未来人口总数预测值', marker='o')
95 plt.fill_between(df1['年份'], lower_bound, upper_bound, color='#B694ED',
96               ↪ alpha=0.2, label='95% 置信区间')
97 plt.fill_between(df2['年份'], lower_bound_future, upper_bound_future,
98               ↪ color='lightgreen', alpha=0.2, label='预测数据 95%置信区间')
99 # 标注 2030 年, 2035 年和 2045 年的数据

```

```

93 for year in [2030, 2035, 2045]:
94     if year in df2['年份'].values:
95         pop = df2[df2['年份'] == year]['人口总数（万人）预测值'].values[0]
96         plt.annotate(f'{year}年: {pop:.2f}万人', xy=(year, pop),
97                     ↪ xytext=(10, 10), textcoords='offset points',
98                     ↪ arrowprops=dict(arrowstyle='->'))
99 plt.xticks(np.arange(min(df1['年份'].min(), df2['年份'].min()),
100               ↪ max(df1['年份'].max(), df2['年份'].max()) + 1, 5))
101 plt.xlabel('年份', fontsize=15)
102 plt.ylabel('人口总数（万人）', fontsize=15)
103 plt.title('人口总数预测及置信区间')
104 plt.legend()
105 plt.grid(axis='y', linestyle='--')
106 plt.xticks(fontsize=16)
107 plt.yticks(fontsize=16)
108 plt.show()
109
110 # 生成网格数据用于绘制三维曲面
111 gdp_min, gdp_max = df1['GDP（亿元）'].min(), df1['GDP（亿元）'].max()
112 enrollment_min, enrollment_max = df1['高等教育毛入学率（%）'].min(),
113     ↪ df1['高等教育毛入学率（%）'].max()
114 gdp_grid, enrollment_grid = np.meshgrid(
115     np.linspace(gdp_min, gdp_max, 100),
116     np.linspace(enrollment_min, enrollment_max, 100)
117 )
118
119 # 将网格数据转换为适合模型预测的格式
120 grid_data = np.column_stack([gdp_grid.ravel(), enrollment_grid.ravel()])
121 grid_poly = poly.transform(grid_data)
122
123 # 进行预测
124 predicted_population = model.predict(grid_poly)
125 predicted_population = predicted_population.reshape(gdp_grid.shape)
126
127 # 创建三维图
128 fig = plt.figure(figsize=(10, 8))
129 ax = fig.add_subplot(111, projection='3d')
130
131 # 绘制三维曲面
132 ax.plot_surface(gdp_grid, enrollment_grid, predicted_population,
133               ↪ cmap='viridis', alpha=0.5)
134 ax.scatter(df1['GDP（亿元）'], df1['高等教育毛入学率（%）'], df1['人口总数
135     ↪ （万人）'], c='r', marker='o')
136 # 设置坐标轴标签和标题
137 ax.set_xlabel('GDP（亿元）')
138 ax.set_ylabel('高等教育毛入学率（%）')
139 ax.set_zlabel('人口总数（万人）')
140 ax.set_title('多元非线性回归模型预测人口总数三维曲面')

```

```

130 plt.show()

131 print('模型均方误差: ', mse)
132 print('模型决定系数: ', r2)
133 print('未来 20 年人口总数预测结果: ')
134 print(df2[['年份', '人口总数 (万人) 预测值']])

```

1.2 问题二源代码

1.2.1 教育指标权重计算

```

1 import pandas as pd
2 import numpy as np

3 # 读取文件
4 excel_file = pd.ExcelFile('第二问指标.xlsx')

5 # 获取指定工作表中的数据
6 df = excel_file.parse('Sheet1')

7 # 查看数据的基本信息
8 print('数据基本信息: ')
9 df.info()

10 # 查看数据集行数和列数
11 rows, columns = df.shape

12 if rows < 100 and columns < 20:
13     # 短表数据 (行数少于 100 且列数少于 20) 查看全量数据信息
14     print('数据全部内容信息: ')
15     print(df.to_csv(sep='\t', na_rep='nan'))
16 else:
17     # 长表数据查看数据前几行信息
18     print('数据前几行内容信息: ')
19     print(df.head().to_csv(sep='\t', na_rep='nan'))
20 # 提取需要的指标数据
21 selected_columns = ['研究生毕业生占比', '高等学校专任教师数占比', '师生
    ↳ 比', '生均教育经费']
22 new_df = df[selected_columns]

23 # 对数据进行标准化
24 P = new_df.div(new_df.sum(axis=0), axis=1)

25 # 计算每个指标的熵值
26 E = -(P * np.log(P).replace([np.inf, -np.inf], 0)).sum(axis=0) /
    ↳ np.log(len(P))

```

```

27 # 计算每个指标的差异系数
28 G = 1 - E

29 # 计算每个指标的权重
30 W = G / G.sum()

31 # 输出结果
32 result = pd.DataFrame({
33     '指标': selected_columns,
34     '权重': W
35 })

36 print(result)

```

1.2.2 灰度预测

```

1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 from sklearn.metrics import mean_squared_error

5 def grey_model(data):
6     AGO = np.cumsum(data)

7     Z = (AGO[:-1] + AGO[1:]) / 2.0

8     # 构造矩阵 B 和 Y
9     B = np.vstack((-Z, np.ones(len(Z)))) .T
10    Y = data[1:].reshape(len(Z), 1)

11    # 计算参数 a 和 b
12    a, b = np.dot(np.linalg.pinv(B), Y)

13    # 预测函数
14    def predict(k):
15        return (data[0] - b / a) * np.exp(-a * k) + b / a

16    # 计算模拟值
17    sim_values = []
18    sim_values.append(data[0])
19    for i in range(1, len(data)):
20        sim_values.append(predict(i))

21    # 模型检验
22    errors = data - sim_values

```

```

23     S1 = np.std(data, ddof=1)
24     S2 = np.std(errors, ddof=1)
25     C = S2 / S1
26     P = np.sum(np.abs(errors - np.mean(errors)) < 0.6745 * S1) /
        ↪ len(data)

27     return predict, sim_values, (a[0], b[0]), (C[0], P[0])

28 # 1. 读取历史数据
29 history_df = pd.read_excel('生均教育经费.xlsx')
30 years = history_df['年份'].values
31 population = history_df['人口总数 (万人)'].values
32 expenditure = history_df['生均教育经费 (元)'].values

33 # 2. 灰度模型预测
34 predict_func, sim_values, (a, b), (C, P) = grey_model(expenditure)

35 print(f"灰色模型参数: a={a:.4f}, b={b:.2f}")
36 print(f"模型检验: 后验差比值 C={C:.4f}, 小误差概率 P={P:.4f}")

37 # 3. 读取未来人口数据
38 future_df = pd.read_excel('第一问预测结果.xlsx')
39 future_years = future_df['年份'].values
40 future_population = future_df['人口总数 (万人)'].values

41 # 4. 预测未来生均经费
42 future_k = np.arange(len(expenditure), len(expenditure) +
        ↪ len(future_years))
43 future_expenditure = [predict_func(k) for k in future_k]

44 # 5. 计算增长率
45 growth_rates = [(future_expenditure[i] - future_expenditure[i - 1]) /
        ↪ future_expenditure[i - 1] * 100
46                 for i in range(1, len(future_expenditure))]
47 growth_rates.insert(0, np.nan)

48 # 6. 创建结果 DataFrame
49 result_df = pd.DataFrame({
50     '年份': future_years,
51     '预测人口 (万人)': future_population,
52     '预测生均经费 (元)': np.round(future_expenditure, 2),
53     '增长率 (%)': np.round(growth_rates, 2)
54 })

```

1.2.3 归一化处理

```
1 import pandas as pd
2 # 1. 原始数据（年份 + 指标值）
3 data = [
4     (2024, 0.0926), (2025, 0.0957), (2026, 0.0989), (2027, 0.1023),
5     (2028, 0.1057), (2029, 0.1093), (2030, 0.113), (2031, 0.1168),
6     (2032, 0.1208), (2033, 0.1249), (2034, 0.1291), (2035, 0.1335),
7     (2036, 0.138), (2037, 0.1427), (2038, 0.1475), (2039, 0.1525),
8     (2040, 0.1577), (2041, 0.1631), (2042, 0.1686), (2043, 0.1744),
9     (2044, 0.1783), (2045, 0.1824)
10 ]
11 # 2. 转换为 DataFrame
12 df = pd.DataFrame(data, columns=["年份", "指标值"])
13 # 3. 归一化计算
14 min_val = df["指标值"].min()
15 max_val = df["指标值"].max()
16 df["标准化值"] = (df["指标值"] - min_val) / (max_val - min_val)
17 # 4. 写入 Excel
18 output_path = "教育指标标准化结果.xlsx"
19 df.to_excel(output_path, index=False, sheet_name="标准化数据")
20 print(f"已保存到: {output_path}")
21 print("最小值:", min_val)
22 print("最大值:", max_val)
23 print("标准化结果示例: ")
24 print(df.head()) # 打印前 5 行验证"
```

1.3 问题三源代码

1.3.1 图一源代码

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 from scipy.interpolate import interp1d
4 from matplotlib.font_manager import FontProperties
5 import numpy as np
6 font = FontProperties(fname=r"C:\Windows\Fonts\simsum.ttc", size=12)
7 plt.rcParams['font.family'] = font.get_name()
8 plt.rcParams['axes.unicode_minus'] = False
9 # 读取文件
10 excel_file = pd.ExcelFile('第三问整合 1.xlsx')
```

```

11 # 获取指定工作表中的数据
12 df = excel_file.parse('Sheet1')
13 # 筛选 2010 - 2023 年的数据
14 selected_df = df[(df['年份'] >= 1992) & (df['年份'] <= 2023)]

15 x_new = np.linspace(selected_df['年份'].min(), selected_df['年份'].max(),
    ↪ 500)

16 # 对标准化 1 进行插值
17 f1 = interp1d(selected_df['年份'], selected_df['标准化 1'], kind='cubic')
18 y1_smooth = f1(x_new)

19 # 对标准化 2 进行插值
20 f2 = interp1d(selected_df['年份'], selected_df['标准化 5'], kind='cubic')
21 y2_smooth = f2(x_new)

22 # 对标准化 3 进行插值, 处理可能存在的缺失值
23 f3 = interp1d(selected_df['年份'], selected_df['标准化 3'], kind='cubic')
24 y3_smooth = f3(x_new)

25 # 绘制平滑折线图
26 plt.figure(figsize=(10, 6))
27 plt.plot(x_new, y1_smooth, label='第三产业占 GDP 比重 (D)
    ↪ ',color='#edfc78')
28 plt.plot(x_new, y3_smooth, label='Y1 (本科以上毕业生平均工资 / 社会平均工
    ↪ 资)',color='#90b6a5')
29 plt.plot(x_new, y2_smooth, label='本科毕业生人数 (S)',color='#bea887')

30 plt.scatter(selected_df['年份'], selected_df['标准化 1'], marker='o',
    ↪ color='#edfc78')
31 plt.scatter(selected_df['年份'], selected_df['标准化 5'], marker='s',
    ↪ color='#bea887')
32 plt.scatter(selected_df['年份'], selected_df['标准化 3'], marker='^',
    ↪ color='#90b6a5')
33 # 添加标题和标签
34 plt.title('1992 - 2023 年 3 个标准化值折线图')
35 plt.xlabel('年份',fontsize='12')
36 plt.ylabel('标准化值',fontsize='12')
37 plt.xticks(fontsize=13)
38 plt.yticks(fontsize=13)
39 # 添加图例
40 plt.legend()
41 # 显示网格
42 plt.grid(axis='y',linestyle='--')

```



```
43 # 显示图形
44 plt.show()
```

1.3.2 图二源代码

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 from scipy.interpolate import interp1d
4 from matplotlib.font_manager import FontProperties
5 import numpy as np
6 font = FontProperties(fname=r"C:\Windows\Fonts\simsun.ttc", size=12)
7 plt.rcParams['font.family'] = font.get_name()
8 plt.rcParams['axes.unicode_minus'] = False
9 # 读取文件
10 excel_file = pd.ExcelFile('第三问整合 1.xlsx')
11
12 # 获取指定工作表中的数据
13 df = excel_file.parse('Sheet1')
14 # 筛选 2010 - 2023 年的数据
15 selected_df = df[(df['年份'] >= 1992) & (df['年份'] <= 2023)]
16
17 x_new = np.linspace(selected_df['年份'].min(), selected_df['年份'].max(),
18 ↪ 500)
19
20 # 对标准化 1 进行插值
21 f1 = interp1d(selected_df['年份'], selected_df['标准化 1'], kind='cubic')
22 y1_smooth = f1(x_new)
23
24 # 对标准化 2 进行插值
25 f2 = interp1d(selected_df['年份'], selected_df['标准化 4'], kind='cubic')
26 y2_smooth = f2(x_new)
27
28 # 对标准化 3 进行插值, 处理可能存在的缺失值
29 f3 = interp1d(selected_df['年份'], selected_df['标准化 5'], kind='cubic')
30 y3_smooth = f3(x_new)
31
32 # 绘制平滑折线图
33 plt.figure(figsize=(10, 6))
34 plt.plot(x_new, y1_smooth, label='第三产业占 GDP 比重 (D)
35 ↪ ', color='#cbf4a5')
36 plt.plot(x_new, y2_smooth, label='Y2 (失业率)', color='#27c49c')
37 plt.plot(x_new, y3_smooth, label='本科毕业生人数 (S)', color='#008da7')
38
39 plt.scatter(selected_df['年份'], selected_df['标准化 1'], marker='o',
40 ↪ color='#cbf4a5')
```

```

31 plt.scatter(selected_df['年份'], selected_df['标准化 4'], marker='s',
    ↪ color='#27c49c')
32 plt.scatter(selected_df['年份'], selected_df['标准化 5'], marker='^',
    ↪ color='#008da7')
33 # 添加标题和标签
34 plt.title('1992 - 2023 年 3 个标准化值折线图')
35 plt.xlabel('年份',fontsize='12')
36 plt.ylabel('标准化值',fontsize='12')
37 plt.xticks(fontsize=13)
38 plt.yticks(fontsize=13)
39 # 添加图例
40 plt.legend()
41 # 显示网格
42 plt.grid(axis='y',linestyle='--')

43 # 显示图形
44 plt.show()

```

1.3.3 模型一源代码

```

1 import pandas as pd
2 import seaborn as sns
3 import numpy as np
4 import statsmodels.api as sm
5 import matplotlib.pyplot as plt
6 from matplotlib.font_manager import FontProperties
7 font = FontProperties(fname=r"C:\Windows\Fonts\simsum.ttc", size=12)
8 plt.rcParams['font.family'] = font.get_name()
9 plt.rcParams['axes.unicode_minus'] = False
10 # 1. 数据读取与预处理
11 df = pd.read_excel('第三问整合 1.xlsx')
12 df = df[df['年份']>=2013]
13 # 提取需要的列
14 data = df[['年份', '高/平均', '研究生毕业人数', 'GDP 增长指数']].copy()

15 # 计算研究生毕业人数增长率 ( $S_t$ )
16 data['S_t'] = data['研究生毕业人数'].pct_change() * 100 # 转换为百分比

17 # 计算 GDP 增长率 ( $G_t$ ), 假设 GDP 增长指数基期为 100
18 data['G_t'] = data['GDP 增长指数'].pct_change() * 100

19 # 计算供给增速差 ( $S_t - G_t$ )
20 data['供给增速差'] = data['S_t'] - data['G_t']

21 # 删除缺失值 (第一年没有增长率)
22 data = data.dropna()

```

```

23 # 2. 模型构建
24 X = data['供给增速差']
25 y = data['高/平均']
26 X = sm.add_constant(X) # 添加常数项

27 model = sm.OLS(y, X)
28 results = model.fit()

29 print("="*50)
30 print("供需失衡与收入贬值模型回归结果")
31 print("="*50)
32 print(results.summary())
33 print("\n关键结论: ")
34 print(f"- 供给增速差系数: {results.params[1]:.4f} (p 值
    ↪ = {results.pvalues[1]:.4f})")
35 print(f"- 模型 R 平方: {results.rsquared:.4f}")

36 # 4. 可视化
37 plt.figure(figsize=(12, 6))

38 # 散点图与回归线
39 plt.subplot(1, 2, 1)
40 plt.scatter(data['供给增速差'], y, alpha=0.7,color='#949fff')
41 plt.plot(data['供给增速差'], results.fittedvalues, 'r-',color='#ffe8ff')
42 plt.xlabel('供给增速差 (S_t - G_t)%')
43 plt.ylabel('高学历/平均工资比值')
44 plt.title('供需关系对工资比值的影响')
45 plt.grid(axis='y',linestyle='--')
46 plt.legend()
47 # 残差诊断图
48 plt.subplot(1, 2, 2)
49 plt.scatter(results.fittedvalues, results.resid,
    ↪ alpha=0.7,color='#ffba8c')
50 plt.axhline(y=0, color='#b3a7b7', linestyle='--')
51 plt.xlabel('预测值')
52 plt.ylabel('残差')
53 plt.title('残差诊断图')
54 plt.grid(axis='y',linestyle='--')
55 plt.legend()
56 plt.tight_layout()
57 plt.show()
58 # 5. 输出关键指标表格
59 result_table = pd.DataFrame({
60     '年份': data['年份'],
61     '高学历/平均工资': data['高/平均'],
62     '研究生毕业人数增长率 (%)': data['S_t'].round(2),

```

```

63     'GDP 增长率 (%)': data['G_t'].round(2),
64     '供给增速差 (%)': data['供给增速差'].round(2),
65     '模型预测值': results.fittedvalues.round(4)
66 })

67 print("\n详细计算结果: ")
68 print(result_table.to_string(index=False))

```

1.3.4 模型二源代码

```

1 import pandas as pd
2 import numpy as np
3 import statsmodels.api as sm
4 import matplotlib.pyplot as plt
5 from mpl_toolkits.mplot3d import Axes3D
6 from matplotlib.font_manager import FontProperties
7 font = FontProperties(fname=r"C:\Windows\Fonts\simsum.ttc", size=12)
8 plt.rcParams['font.family'] = font.get_name()
9 plt.rcParams['axes.unicode_minus'] = False
10 # 1. 数据读取与预处理
11 df = pd.read_excel('第三问整合 1.xlsx')
12 df = df[df['年份']>=2018]
13 # 提取需要的列
14 data = df[['年份', '失业率', '研究生毕业人数', 'GDP 增长指数']].copy()

15 # 计算研究生毕业人数增长率 (S_t)
16 data['S_t'] = data['研究生毕业人数'].pct_change() * 100 # 转换为百分比

17 # 计算 GDP 增长率 (G_t), 假设 GDP 增长指数基期为 100
18 data['G_t'] = data['GDP 增长指数'].pct_change() * 100

19 # 计算供给增速差 (S_t - G_t)
20 data['供给增速差'] = data['S_t'] - data['G_t']

21 # 删除缺失值 (第一年没有增长率)
22 data = data.dropna()

23 # 2. 多元线性回归模型构建
24 X = data[['供给增速差', 'G_t']]
25 y = data['失业率']
26 X = sm.add_constant(X) # 添加常数项

27 model = sm.OLS(y, X)
28 results = model.fit()

29 # 3. 结果分析
30 print("="*60)

```

```

31 print("供需失衡与就业贬值模型回归结果")
32 print("="*60)
33 print(results.summary())
34 print("\n关键结论: ")
35 print(f"- 供给增速差系数: {results.params[1]:.4f} (p 值
    ↪   = {results.pvalues[1]:.4f})")
36 print(f"- GDP 增长率系数: {results.params[2]:.4f} (p 值
    ↪   = {results.pvalues[2]:.4f})")
37 print(f"- 模型 R 平方: {results.rsquared:.4f}")

38 # 4. 可视化分析
39 plt.figure(figsize=(18, 6))

40 # 子图 1: 供给增速差与失业率关系
41 plt.subplot(1, 3, 1)
42 plt.scatter(data['供给增速差'], y, alpha=0.7)
43 plt.xlabel('供给增速差 (S_t - G_t)')
44 plt.ylabel('失业率 (%)')
45 plt.title('供给过剩对失业率的影响')
46 plt.grid(True)

47 # 添加回归线 (控制其他变量)
48 x_pred = np.linspace(data['供给增速差'].min(), data['供给增速差'].max(),
    ↪   100)
49 g_mean = data['G_t'].mean() # 固定 GDP 增长率为均值
50 y_pred = results.params[0] + results.params[1]*x_pred +
    ↪   results.params[2]*g_mean
51 plt.plot(x_pred, y_pred, 'r-')

52 # 子图 2: GDP 增长率与失业率关系
53 plt.subplot(1, 3, 2)
54 plt.scatter(data['G_t'], y, alpha=0.7)
55 plt.xlabel('GDP 增长率 (%)')
56 plt.ylabel('失业率 (%)')
57 plt.title('经济增长对失业率的影响')
58 plt.grid(True)

59 # 添加回归线 (控制其他变量)
60 x_pred = np.linspace(data['G_t'].min(), data['G_t'].max(), 100)
61 s_mean = data['供给增速差'].mean() # 固定供给增速差为均值
62 y_pred = results.params[0] + results.params[1]*s_mean +
    ↪   results.params[2]*x_pred
63 plt.plot(x_pred, y_pred, 'r-')

64 # 子图 3: 3D 关系图
65 ax = plt.subplot(1, 3, 3, projection='3d')
66 ax.scatter(data['供给增速差'], data['G_t'], y, c='b', marker='o')
67 ax.set_xlabel('供给增速差 (%)')

```

```

68 ax.set_ylabel('GDP 增长率 (%)')
69 ax.set_zlabel('失业率 (%)')
70 ax.set_title('三维影响因素关系')

71 # 生成回归平面
72 x_surf = np.linspace(data['供给增速差'].min(), data['供给增速差'].max(),
    ↪ 10)
73 y_surf = np.linspace(data['G_t'].min(), data['G_t'].max(), 10)
74 x_surf, y_surf = np.meshgrid(x_surf, y_surf)
75 z_pred = results.params[0] + results.params[1]*x_surf +
    ↪ results.params[2]*y_surf
76 ax.plot_surface(x_surf, y_surf, z_pred, color='r', alpha=0.3)

77 plt.tight_layout()
78 plt.show()

79 # 5. 输出关键指标表格
80 result_table = pd.DataFrame({
81     '年份': data['年份'],
82     '失业率 (%)': data['失业率'],
83     '研究生毕业人数增长率 (%)': data['S_t'].round(2),
84     'GDP 增长率 (%)': data['G_t'].round(2),
85     '供给增速差 (%)': data['供给增速差'].round(2),
86     '模型预测失业率 (%)': results.fittedvalues.round(2)
87 })

88 print("\n详细计算结果: ")
89 print(result_table.to_string(index=False))

```

附录 B 问题二教育指标预测与标准化结果

年份	生均教育经费（元）	标准化 1	师生比	标准化 2	高等学校专任教师数占比	标准化 2	研究生毕业生占比	标准化 1
2024	158921	0.000	18.120	0.000	0.113	0.000	0.093	0.000
2025	164512	0.035	18.250	0.037	0.118	0.031	0.096	0.035
2026	170293	0.071	18.390	0.076	0.123	0.063	0.099	0.070
2027	176271	0.108	18.530	0.116	0.129	0.097	0.102	0.108
2028	182453	0.146	18.670	0.156	0.135	0.132	0.106	0.146
2029	188847	0.186	18.820	0.198	0.141	0.169	0.109	0.186
2030	195461	0.227	18.970	0.241	0.147	0.207	0.113	0.227
2031	202304	0.269	19.120	0.283	0.154	0.247	0.117	0.269
2032	209384	0.313	19.280	0.329	0.161	0.289	0.121	0.314
2033	216711	0.359	19.440	0.374	0.168	0.333	0.125	0.360
2034	224293	0.406	19.610	0.422	0.176	0.379	0.129	0.406
2035	232140	0.455	19.780	0.470	0.184	0.427	0.134	0.455
2036	240262	0.505	19.960	0.521	0.192	0.478	0.138	0.506
2037	248669	0.557	20.140	0.572	0.201	0.530	0.143	0.558
2038	257371	0.611	20.330	0.626	0.210	0.585	0.148	0.611
2039	266379	0.667	20.520	0.680	0.220	0.642	0.153	0.667
2040	275704	0.725	20.720	0.737	0.230	0.702	0.158	0.725
2041	285357	0.785	20.920	0.793	0.240	0.765	0.163	0.785
2042	295350	0.847	21.130	0.853	0.251	0.831	0.169	0.846
2043	305695	0.911	21.340	0.912	0.263	0.900	0.174	0.911
2044	312769	0.955	21.490	0.955	0.271	0.948	0.178	0.954
2045	320015	1.000	21.650	1.000	0.279	1.000	0.182	1.000