Statistical Study on Green Productivity and High-Quality Economic Development

Chen Chen, Pukun Zhao and Fanqing Zhou*

Guangdong University of Finance & Economics, Guangzhou, Guangdong, 510320, China

*Corresponding author's e-mail: nan89888@qq.com

Abstract. Green development is the foundation of high-quality development, and green productivity is its new form. This study examines the relationship between green productivity and six tertiary indicators under resource-efficient, environmentally friendly productivity. Missing and outlier values are handled using the K-Nearest Neighbors (KNN) method, and indicator weights are calculated with the entropy method, revealing that gas utilization has the highest weight. A model is then developed by replacing the gas indicator with the air pollution index to improve accuracy. A multiple regression model is optimized using a genetic algorithm, with optimal parameters indicating the need to improve energy intensity, energy structure, water use intensity, and waste utilization, while reducing gas and wastewater emissions. Empirical analysis confirms the model's reliability with a goodness of fit of 0.99, supporting its use in government green development policy-making.

Keywords: Green productivity, K-Nearest Neighbors, Multiple regression model, Genetic algorithm

1 Introduction

Green productivity is a new form of productivity that emphasizes green development, driven by innovation, high-tech, and sustainable growth. Its goal is to balance human development with nature, supporting carbon peaking and neutrality. President Xi Jinping [1] has highlighted that green development is key to high-quality growth, with green productivity embodying this new model. This paper employs tertiary indicators to quantify green productivity and proposes strategies for sustainable development [2].

Resource-Efficient Productivity: Energy Intensity (B1): Energy consumption per GDP (%); Energy Structure (B2): Fossil fuel consumption per GDP (%); Water Use Intensity (B3): Industrial water use per GDP (%). **Environmentally Friendly Productivity:** Waste Utilization (B4): Proportion of industrial waste utilization per production (%); Waste Gas Emissions (B5): Industrial waste gas emissions per GDP (%); Wastewater Emissions (B6): Industrial SO_2 emissions per GDP (%)

W. He et al. (eds.), *Proceedings of the 2024 10th International Conference on Advances in Energy Resources and Environment Engineering (ICAESEE 2024)*, Atlantis Highlights in Engineering 36,
https://doi.org/10.2991/978-94-6463-708-3_36

2 Data Preprocessing

2.1 Detection of Missing Values

Environmental monitoring data, such as air quality and wastewater utilization, may contain missing values. Python is used to detect and visualize these missing values in the datasets.

2.2 Outlier Detection and Handling

Outliers, often caused by equipment malfunctions, can distort the analysis. These outliers are addressed separately from missing values.

2.3 Handling Missing Data

For missing data, especially in air quality measurements, the **KNN** method is used [3]. This method calculates the Euclidean distance to identify the nearest valid data points, and missing values are estimated by averaging the nearest neighbors, weighted by their distance.

Results: KNN adjustments corrected clustering issues and improved accuracy. Missing values were deleted when imputation was not possible. The tertiary indicators of green productivity in Jiangsu are shown in Table 1.

Table 1. Tertiary Indicators of Green Productivity in Jiangsu

Province	Year	Energy / GDP (%)	Water Use / GDP (%)	Waste Utilization / Production (%)	Gas Emissions / GDP (%)	SO2 Emissions / GDP (%)
Jiangsu	2010	0.06	0.46	96.65	0.13	0.000025
Jiangsu	2011	0.06	0.39	95.43	0.12	0.000022
				...		
Jiangsu	2022	0.03	0.20	93.12	0.00	0.000001

3 Entropy Weight Measurement Method

3.1 Indicator Normalization

To apply the entropy weight method, indicator values are first normalized to the [0, 1] range using the range normalization method. Weights are then assigned based on the relative information each indicator provides. A close neighbor approach is used to optimize interactions. The formula for normalization is:

$$x_{ij}^{*} = \frac{x_{ij} - m_j}{M_j - m_j} \tag{1}$$

Where $M_j = \max\{x_{ij}\}, m_j = \min\{x_{ij}\}$.

3.2 Calculating Entropy Weight Divergence Coefficient

The entropy weight method assigns weights to each indicator based on the amount of information it provides, either more or less. According to the traditional entropy method, the divergence coefficient needs to be calculated first. The specific calculation steps are as follows:

(1) Calculate the characteristic ratio q_{ij} of indicator j for sample i:

$$q_{ij} = \frac{x_{ij}}{\sum_{i=1}^{m} x_{ij}} \tag{2}$$

where m is the number of samples.

(2) Calculate the entropy value w_j of indicator j:

$$w_j = \frac{1}{\ln m \sum_{i=1}^{m} q_{ij} \ln \frac{1}{q_{ij}}} \tag{3}$$

Where $a_{ij} = \ln \frac{1}{q_{ij}}$ represents the information amount, and $b_j = \ln m \sum_{i=1}^{m} q_{ij} \ln \frac{1}{q_{ij}}$ represents the total information amount. w_j represents the entropy value.

(3) Calculate the Divergence Coefficient g_j of Indicator j

$$g_j = 1 - w_j \tag{4}$$

3.3 Calculating Green Productivity

(1) Based on the normalized indicator data and their coefficients of variation, weights are assigned to each of the six tertiary indicators. The coefficients of variation for each indicator are as follows:

Resource-efficient productivity (Green Productivity):

Energy intensity (B1): 0.11; Energy structure (B2): 0.13; Water intensity (B3): 0.12

Environmentally friendly productivity (Green Productivity): Waste utilization (B4): 0.14; Wastewater discharge (B5): 0.13; Exhaust emission (B6): 0.37

(2) Green productivity calculations

Green productivity will be obtained from a weighted average of the six individual indices of the three tiers of indicators mentioned above. The calculation formula is:

$$GP = \sum_{i=1}^{m} x_i g_j \tag{5}$$

Where x_j is the i individual index of the three-level index and $g_j$is the coefficient of variation of the j index. According to the above method, the final results of the green productivity index are presented in Table 2.

Table 2. Green Productivity Component Results

year	Beijing	Guangdong	Hainan	Hebei	Hunan	Hubei	nationwide
2014	14.090	25.401	2.657	5.670	5.970	6.742	7.452
2015	14.695	27.329	3.017	5.935	6.374	7.252	7.941
			...				
2022	16.236	30.584	2.908	6.264	7.106	7.437	7.870

4 Big Data and the Green Productivity Model

In the above entropy weight method for calculating the coefficient of variability, green productivity consists of two secondary indicators, resource-saving productivity and environment-friendly productivity. Next, the big data multiple linear regression indicator model will be constructed from the two productivity respectively.

$$GP = \sum_{i,j=1}^{m} R_i g_j + \sum_{i,j=1}^{m} E_i g_j \tag{6}$$

$$R_i = 0.11EP_{ij} + 0.13ES_{ij} + 0.12WP_{ij} + 0.14RR_{ij} + \mu_{ij} + \epsilon \tag{7}$$

$$E_i = 0.13WD_{ij} + 0.37OD_{ij} + \mu_{ij} + \epsilon \tag{8}$$

Description of the six tertiary indicators of green productivity: energy intensity(EP_{ij}): Energy consumption/GDP (%); energy structure(ES_{ij}): Primary energy information was collected and multiplied by a coking coal coefficient to derive; water intensity(WP_{ij}): Industrial water consumption/GDP (%); utilization of waste materials(RR_{ij}): Comprehensive utilization/generation of industrial solid waste (%); Wastewater discharge(WD_{ij}): Industrial wastewater discharges/GDP; exhaust emission(OD_{ij}): AQI calculated; Unobservable individual factors(μ_{ij}); error term(ϵ).

Note: Due to limited data on energy structure, the primary energy share is calculated based on primary energy production, as detailed in the China Statistical Yearbook.

5 Calculation and Modeling of the Environmental Pollution Index Aqi

The coefficient of variation for exhaust emissions is 0.37, the highest among the six tertiary indicators. This leads to calculating the environmental pollution index (AQI) for each region based on exhaust emissions to better assess green productivity.

The AQI is calculated using the Air Quality Sub-Index (IAQI) for each pollutant, as outlined in the "Environmental Pollution Quality Index (AQI) Technical Regulations (Trial)" (HJ633-2012). The IAQI for each pollutant is calculated using the formula:

$$IAQI_P = \frac{IAQI_{Hi} - IAQI_{Lo}}{BP_{Hi} - BP_{Lo}} . (C_P - BP_{Lo}) + IAQI_{Lo} \tag{9}$$

The meaning of the symbols in the formula is as follows: $IAQI_P$ pollutant environmental pollution sub-index, C_P pollutant concentration value;B_{Hi}, BP_{Lo} with the high and low values of similar pollutant concentrations;$IAQI_{Hi}$, $IAQI_{Lo}$ Corresponding air pollution sub-index.

The Air Quality Index (AQI) is calculated based on the highest IAQI value from pollutants such as SO_2, NO_2, PM10, PM2.5, O_3, and CO. The AQI classification ranges from 'Excellent' (AQI ≤ 50) to 'Severe Pollution' (AQI > 300), with specific concentration limits for each pollutant.

So, the AQI formula can be expressed as:

$$AQI = max\{IAQI_{SO_2}, IAQI_{NO_2}, IAQI_{PM_{10}}, IAQI_{PM_{2.5}}, IAQI_{O_3}, IAQI_{CO}\}$$

Air quality is classified according to the AQI value, the specific range is shown in Table 3. When the AQI is ≤50, the air quality is rated as "excellent" and there is no primary pollutant. If the AQI exceeds 50, it is necessary to pay attention to the Individual Air Quality Index (IAQI) for each pollutant, with the pollutant with the highest value being the primary pollutant. If two or more IAQIs have the highest value, these pollutants will be recognized as the top pollutant at the same time. In addition, if the IAQI for a pollutant exceeds 100, that pollutant is considered an exceedance pollutant.

Table 3. Ambient Pollution Index Levels and AQI Ranges.

Environmental Pollution Index Scale	Ambient Pollution Index (AQI) range	Environmental Pollution Index Scale	Ambient Pollution Index (AQI) range
Excellent	[0,50]	moderate pollution	[151,200]
Good	[51,100]	heavy pollution	[201,300]
light pollution	[101,150]	severe contamination	[301,+∞)

Using the standardized AQI formula, the IAQI of each monitoring data is calculated and compared with the concentration limits, with the maximum value selected as the AQI. The AQI results for each city are shown in Table 4 below (see Appendix 1 for detailed results).

Table 4. Environmental Pollution Index AQI Values

dates	Tianjin	Wuhan	Beijing	Shanghai	Shenzhen	Guangzhou	Luoyang
February 13	68.83 (Good)	27.54 (Excellent)	68.17 (Good)	85.29 (Good)	56.38 (Good)	18.17 (Excellent)	77.08 (Good)
February 14	77.67 (Good)	20.67 (Excellent)	27.29 (Excellent)	76.58 (Good)	48.75 (Excellent)	18.88 (Excellent)	55.71 (Good)
...							

Based on the above ESI levels, and the calculated partial ESI values as tried in the table above. The total analysis is as follows:

1. Guangzhou and Wuhan had better air quality, showing smaller fluctuations and lower average AQI values.

2. Tianjin have poorer air quality, showing higher average AQI values and larger fluctuations.

3. The trend of AQI values varies from city to city, with relatively large fluctuations in Tianjin, Jiangsu and Luoyang, while other cities are more stable.

4. The AQIs of all cities are low and there is no highly polluted weather.

6 Optimization Model based on Green Productivity Model

6.1 Optimization Modeling

After modeling the big data and green productivity model, genetic algorithms are used to optimize the multiple regression model and find the optimal green productivity value for each city, aiding government policy planning.

Decision variables are 6 variables that affect green productivity, respectively, energy intensity EP_{ij},energy structure ES_{ij},water intensity WP_{ij},utilization of waste materials RR_{ij},Wastewater discharge WD_{ij},exhaust emission OD_{ij}.

Green raw productivity is

$$GP = \sum_{i,j=1}^{m} R_i g_j + \sum_{i,j=1}^{m} E_i g_j \tag{10}$$

The optimization function is

$$\text{Max } GP(EP_{ij}, ES_{ij}, WP_{ij}, RR_{ij}, WD_{ij}, OD_{ij})$$

$$\text{s.t. } 0 < EP_{ij}, ES_{ij}, WP_{ij}, RR_{ij}, WD_{ij} < 1$$

$$0 < OD_{ij} < 50$$

The optimization function is fitted with the objective of maximizing green productivity using multiple linear regression, and genetic algorithms are used for optimization to find the optimal ratios as a guideline for government decision-making.

Genetic algorithm results [4] are illustrated: After the above genetic algorithm calculations, the genetic algorithm results are shown in Table 5.

Table 5. Genetic Algorithm Results

variable name	energy intensity	energy structure	water intensity	Utilizing Waste	Wastewater discharge	exhaust emission	Green Productivity
Symbol	EP_{ij}	ES_{ij}	WP_{ij}	RR_{ij}	WD_{ij}	OD_{ij}	GP
Parameter value	0.99	0.98	0.98	0.91	0.58	14	5.8

Conclusion: The results show that higher values for most ratios are better, except for wastewater and exhaust emissions (AQI), where lower values are preferred. This paper aims to minimize AQI emissions, especially exhaust emissions. Fig. 1 and Fig. 2 show

the graph and diagram of 100 and 1000 iterations of the genetic algorithm, respectively.7

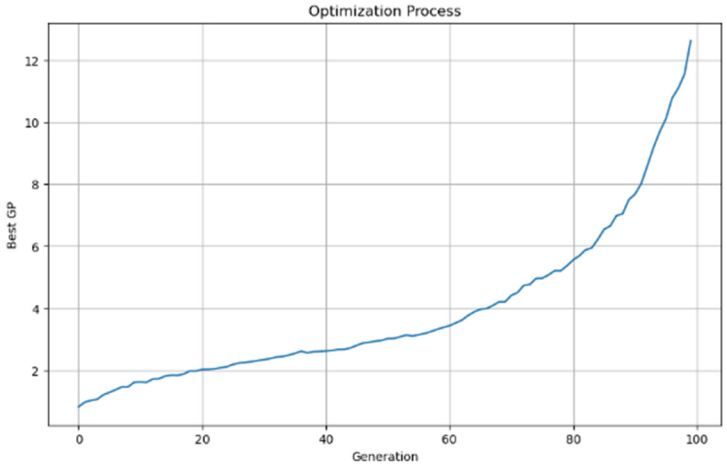

Fig. 1. Graph of 100 iterations of the genetic algorithm

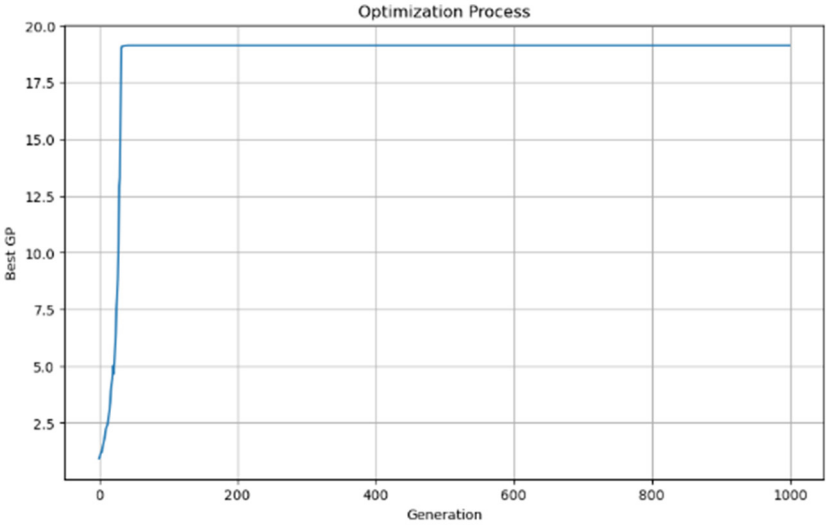

Fig. 2. Diagram of 1000 iterations of the genetic algorithm

The fit of each ensemble and the total ensemble was obtained by running Matlab on the training set ensemble, the establishment ensemble, the test ensemble and the total data ensemble. Figure 3 contains the graphs of the fitting situation of the four sets where RT=0.99, RVa=0.99, RTest=0.99, RAll=0.99. It is found that the fitting is very good and it also proves that the result of the above mentioned genetic algorithm for finding the optimal results is effective.

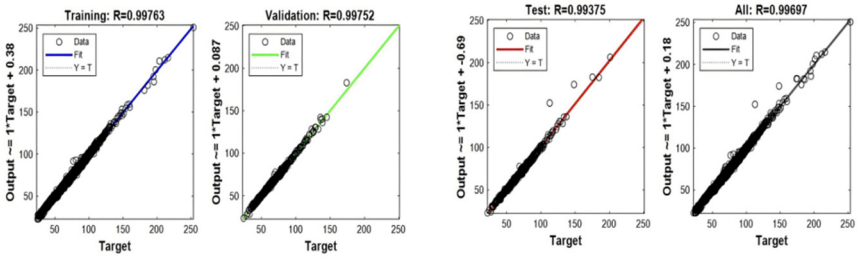

Fig. 3. Genetic Algorithm Optimization Ensemble Fitting Situation

7 Empirical Analysis of Green Productivity

7.1 Empirical Model Setting

To study the effects of green productivity and six key factors (such as air pollution index and water intensity), a panel fixed-effects model is used to avoid large bias errors. The model is defined as follows:

$$GP = \sum_{i,j=1}^{m} R_i g_j + \sum_{i,j=1}^{m} E_i g_j \tag{11}$$

Green Productivity (GP): Overall green productivity measure; Energy intensity (EP): Energy consumption per GDP (%); Energy structure (ES): Proportion of primary energy derived from fossil fuels; Water intensity (WP): Industrial water use per GDP (%); Waste utilization (RR): Proportion of industrial solid waste utilized; Wastewater discharge (WD): Industrial wastewater per GDP; Exhaust emission (OD): Calculated AQI for emissions

7.2 Testing and Analysis

Descriptive Statistics.

The key descriptive statistics for the core variables show the following trends:

Exhaust emission has the highest values, with a maximum of 87.68, while other variables such as energy intensity and wastewater discharge have relatively smaller variations.

The AQI calculations for most variables remain below 100, indicating good air quality, with waste emissions showing the largest fluctuations.

It can be seen that only the waste emissions AQI calculation is larger, the other are proportionally smaller data, and the waste emissions calculation AQI does not exceed the Environmental Pollution Index (EPI) of 100, which is a good weather. Fig. 4 presents the correlation matrix heatmap, highlighting the relationships between key variables in the dataset.

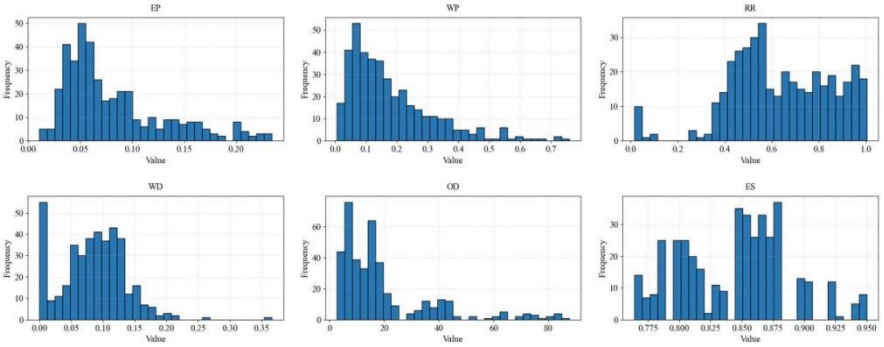

Fig. 4. Histogram of Descriptive Statistics

Correlation Analysis.

The correlation analysis was used to initially explore the correlation between the variables, and the correlation coefficients between the variables are shown in Figure 5 below [5].

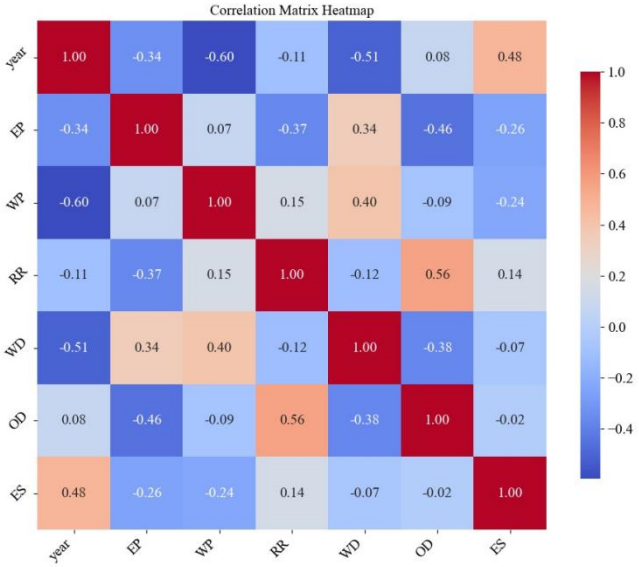

Fig. 5. Correlation coefficient graph

Multicollinearity Test.

The Variance Inflation Factor (VIF) test was conducted to assess multicollinearity. All VIF values were below 10, indicating acceptable levels of multicollinearity. Detailed results are provided in Table 6.

Table 6. Multicollinearity test results

variable name	energy intensity	energy structure	water intensity	utilization of waste materials	exhaust emission	Wastewater discharge
notation	EP_{ij}	ES_{ij}	WP_{ij}	RR_{ij}	OD_{ij}	WD_{ij}
VIF	1.21	0.18	0.02	0.88	1.37	0.01

Hausmann Test.

The Hausmann test confirmed that a fixed effects model is more suitable for this analysis. The detailed test results can be found in Table 7.

Table 7. Hausman test results

Hausman Test	Chi-sq value	degrees of freedom	P value
NQF	623.78	5	0.000

8 Conclusions

In this paper, we preprocess the raw data, fill in the missing values using the KNN method, and normalize them using the entropy weight method. A green productivity big data model was established, focusing on the air pollution index, which has the greatest impact, was modeled separately, and the environmental pollution index of each province was calculated. Subsequently, a genetic algorithm optimization model was constructed, with green productivity as the dependent variable and the six tertiary indicators as the independent variables, setting the lower the emissions of exhaust gases and wastewater the better, and the more the remaining indicators (energy structure, energy intensity, water intensity, and waste utilization) the better. The final empirical analysis passes a variety of statistical tests, proving that the model is effective and can provide support for government decision-making.

References

1. Wang S, Abbas J, Sial M S, et al. (2022). Achieving green innovation and sustainable development goals through green knowledge management: Moderating role of organizational green culture. Journal of Innovation & Knowledge, 7(4), 100272.
2. Stjepanović S, Tomić D, Škare M. (2017). A new approach to measuring green GDP: a cross-country analysis. Entrepreneurship and Sustainability Issues, 4(4), 574.
3. Zhang S. (2021). Challenges in KNN classification. IEEE Transactions on Knowledge and Data Engineering, 34(10): 4663-4675.
4. Katoch S, Chauhan S S and Kumar V. (2021). A review on genetic algorithm: past, present, and future. Multimedia tools and applications, 80: 8091-8126.
5. Gogtay N J, Thatte U M. (2017). Principles of correlation analysis. Journal of the Association of Physicians of India, 65(3), 78-81.

Appendix 1

code1

```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import make_pipeline
import matplotlib.pyplot as plt

plt.rcParams.update({
    'font.family': 'Times New Roman',
    'font.size': 9,
    'axes.titlesize': 9,
    'axes.labelsize': 9,
    'xtick.labelsize': 9,
    'ytick.labelsize': 9
})

data = [
    0.812166284, 0.817346091, 0.815029293, 0.802794936, 0.787287721,
    0.783601968, 0.83432885, 0.845629812, 0.860572829, 0.872936654,
    0.878915263, 0.865456068, 0.851145106, 0.865282227, 0.848225352,
    0.854419532, 0.876297495, 0.895028248, 0.904689902, 0.921569552,
    0.947784271
]

years = list(range(2000, 2021))

df = pd.DataFrame({'year': years, 'value': data})

X = np.array(years).reshape(-1, 1)

linear_reg = LinearRegression()
linear_reg.fit(X, df['value'])

df['linear_trend'] = linear_reg.predict(X)

poly = PolynomialFeatures(degree=2)
pipeline = make_pipeline(poly, LinearRegression())
pipeline.fit(X, df['value'])

df['poly_trend'] = pipeline.predict(X)
```

```
print("数据及其趋势：")
print(df)

plt.figure(figsize=(10, 6))
plt.plot(df['year'], df['value'], label='Original Data')
plt.plot(df['year'], df['linear_trend'], label='Linear Trend', linestyle='--')
plt.plot(df['year'], df['poly_trend'], label='Polynomial Trend (Degree 2)', linestyle='-.')
plt.xlabel('Year')
plt.ylabel('Value')
plt.title('Original Data and Trends')
plt.legend(prop={'family': 'Times New Roman', 'size': 9})
plt.tight_layout()
plt.show()
```

code2

```
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt

plt.rcParams.update({
    'font.family': 'Times New Roman',
    'font.size': 9,
    'axes.titlesize': 9,
    'axes.labelsize': 9,
    'xtick.labelsize': 9,
    'ytick.labelsize': 9
})

data = pd.read_excel('/Users/nongnong/Downloads/绿色生产力与经济高质量发展的统计研究/附录4-绿色生产力_with_GP.xlsx')

GP = data['GP']

independent_vars = ['EP', 'WP', 'RR', 'WD', 'OD', 'ES']
X = data[independent_vars]

X = sm.add_constant(X)

model = sm.OLS(GP, X)

results = model.fit()

print(results.summary())
```

code3

```
import numpy as np
from deap import base, creator, tools, algorithms
import pandas as pd
import matplotlib.pyplot as plt

plt.rcParams.update({
    'font.family': 'Times New Roman',
    'font.size': 9,
    'axes.titlesize': 9,
    'axes.labelsize': 9,
    'xtick.labelsize': 9,
    'ytick.labelsize': 9
})

data = pd.read_excel('/Users/nongnong/Downloads/绿色生产力与经济高质量发展的统计研究/附录3-绿色生产力.xlsx')

od_index = data.columns.get_loc('OD')

def fitness_function(individual):
    EP, ES, WP, RR, WD = individual[:5]
    OD = individual[od_index]
    GP = 0.11 * EP + 0.13 * ES + 0.12 * WP + 0.14 * RR + 0.13 * WD + 0.37 * OD
    return -GP,

creator.create("FitnessMin", base.Fitness, weights=(-1.0,))  # 最小化问题
creator.create("Individual", list, fitness=creator.FitnessMin)

toolbox = base.Toolbox()
toolbox.register("attr_float", np.random.uniform, 0, 1)
toolbox.register("individual", tools.initRepeat, creator.Individual, toolbox.attr_float,
len(data.columns) - 1)
toolbox.register("population", tools.initRepeat, list, toolbox.individual)
toolbox.register("evaluate", fitness_function)
toolbox.register("mate", tools.cxBlend, alpha=0.5)
toolbox.register("mutate", tools.mutGaussian, mu=0, sigma=0.1, indpb=0.2)
toolbox.register("select", tools.selTournament, tournsize=3)

population_size = 500
crossover_prob = 0.7
mutation_prob = 0.2
num_generations = 100
```

```
population = toolbox.population(n=population_size)

for generation in range(num_generations):
    fitnesses = list(map(toolbox.evaluate, population))
    for individual, fitness in zip(population, fitnesses):
        individual.fitness.values = fitness

    offspring = toolbox.select(population, len(population))
    offspring = list(map(toolbox.clone, offspring))

    for child1, child2 in zip(offspring[::2], offspring[1::2]):
        if np.random.random() < crossover_prob:
            toolbox.mate(child1, child2)
            for child in [child1, child2]:
                child[:5] = np.clip(child[:5], 0, 1)
                child[od_index] = np.clip(child[od_index], 0, 50)
            del child1.fitness.values
            del child2.fitness.values

    for mutant in offspring:
        if np.random.random() < mutation_prob:
            toolbox.mutate(mutant)
            mutant[:5] = np.clip(mutant[:5], 0, 1)
            mutant[od_index] = np.clip(mutant[od_index], 0, 50)
            del mutant.fitness.values

    population[:] = offspring

best_individual = tools.selBest(population, 1)[0]
best_GP = -best_individual.fitness.values[0]  # 由于适应度是负值，取负值为最佳
GP
best_parameters = best_individual

print("最优 GP 值:", best_GP)
print("对应的参数值:", best_parameters)
```

code4

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

```
plt.rcParams.update({
    'font.family': 'Times New Roman',
    'font.size': 14,
    'axes.titlesize': 14,
    'axes.labelsize': 14,
    'xtick.labelsize': 14,
    'ytick.labelsize': 14
})

data = pd.read_excel('/Users/nongnong/Downloads/绿色生产力与经济高质量发展的统计研究/附录3-绿色生产力.xlsx')

numeric_columns = data.select_dtypes(include=[np.number]).columns
categorical_columns = data.select_dtypes(exclude=[np.number]).columns

plot_columns = [col for col in numeric_columns if col not in ['year', 'Year']]

plt.figure(figsize=(20, 8))
for i, col in enumerate(plot_columns, 1):
    plt.subplot(2, 3, i)
    plt.hist(data[col], bins=30, edgecolor='black', color='#2878B5')
    plt.title(col, pad=10, fontsize=14, fontfamily='Times New Roman')
    plt.xlabel('Value', fontsize=14, fontfamily='Times New Roman')
    plt.ylabel('Frequency', fontsize=14, fontfamily='Times New Roman')
    plt.grid(True, linestyle='--', alpha=0.3)
    plt.tick_params(axis='both', which='major', labelsize=14)

plt.tight_layout(pad=2.0)
plt.show()

correlation_matrix = data[numeric_columns].corr()

plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix,
            annot=True,
            cmap='coolwarm',
            fmt='.2f',
            square=True,
            cbar_kws={"shrink": .8},
            annot_kws={"size": 14, "family": "Times New Roman"})

plt.title('Correlation Matrix Heatmap', fontsize=14, fontfamily='Times New Roman')
plt.xticks(rotation=45, ha='right', fontsize=14, fontfamily='Times New Roman')
plt.yticks(rotation=45, fontsize=14, fontfamily='Times New Roman')
plt.tight_layout()
```

```
plt.show()

province_mapping = {
    '安徽': 'Anhui',
    '北京': 'Beijing',
    '重庆': 'Chongqing',
    '福建': 'Fujian',
    '甘肃': 'Gansu',
    '广东': 'Guangdong',
    '广西': 'Guangxi',
    '贵州': 'Guizhou',
    '海南': 'Hainan',
    '河北': 'Hebei',
    '黑龙江': 'Heilongjiang',
    '河南': 'Henan',
    '湖北': 'Hubei',
    '湖南': 'Hunan',
    '内蒙古': 'Neimenggu',
    '江苏': 'Jiangsu',
    '江西': 'Jiangxi',
    '吉林': 'Jilin',
    '辽宁': 'Liaoning',
    '宁夏': 'Ningxia',
    '青海': 'Qinghai',
    '陕西': 'Shaanxi',
    '山东': 'Shandong',
    '上海': 'Shanghai',
    '山西': 'Shanxi',
    '四川': 'Sichuan',
    '天津': 'Tianjin',
    '西藏': 'Xizang',
    '新疆': 'Xinjiang',
    '云南': 'Yunnan',
    '浙江': 'Zhejiang'
}

if '省份' in data.columns:  # 假设省份列的名称为'省份'
    data['Province_PY'] = data['省份'].map(province_mapping)
```