New section

```
import pandas as pd

df = pd.read_excel('education_economy_final.xls')
df.head()
```

	year	gov_exp_pct_gdp	school_enrol_secondary_pct	school_enrol_tertiary_pct	ec
	2010	5.642792	106.424942	54.881721	
	2010	5.095880	104.891357	NaN	
	2011	5.480115	106.610641	55.625729	
	2011	4.984250	105.346512	NaN	
	2012	5.488356	105.813583	57.909012	

Next steps: Generate code with df | New interactive sheet

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35 entries, 0 to 34
Data columns (total 12 columns):
 #   Column                                        Non-Null Count  Dtype
---  ------                                        --------------  -----
 0   country                                       35 non-null     object
 1   country_id                                    35 non-null     object
 2   year                                          35 non-null     int64
 3   gov_exp_pct_gdp                               25 non-null     float64
 4   school_enrol_secondary_pct                    33 non-null     float64
 5   school_enrol_tertiary_pct                     29 non-null     float64
 6   economic_data_2025.Inflation (CPI %)          32 non-null     float64
 7   economic_data_2025.GDP (Current USD)          35 non-null     int64
 8   economic_data_2025.GDP per Capita (Current USD)  35 non-null  float64
 9   economic_data_2025.Unemployment Rate (%)      35 non-null     float64
 10  economic_data_2025.Inflation (GDP Deflator, %)  35 non-null  float64
 11  economic_data_2025.GDP Growth (% Annual)      35 non-null     float64
dtypes: float64(8), int64(2), object(2)
memory usage: 3.4+ KB
```

```
df.isna().sum()
```

	0
country	0
country_id	0
year	0
gov_exp_pct_gdp	10
school_enrol_secondary_pct	2

```
numeric_cols = df.columns.drop(['country', 'year'])
df[numeric_cols] = df[numeric_cols].apply(pd.to_numeric, errors='coerce')
```

economic_data_2025.GDP (Current USD) 0

```
df.describe()
```

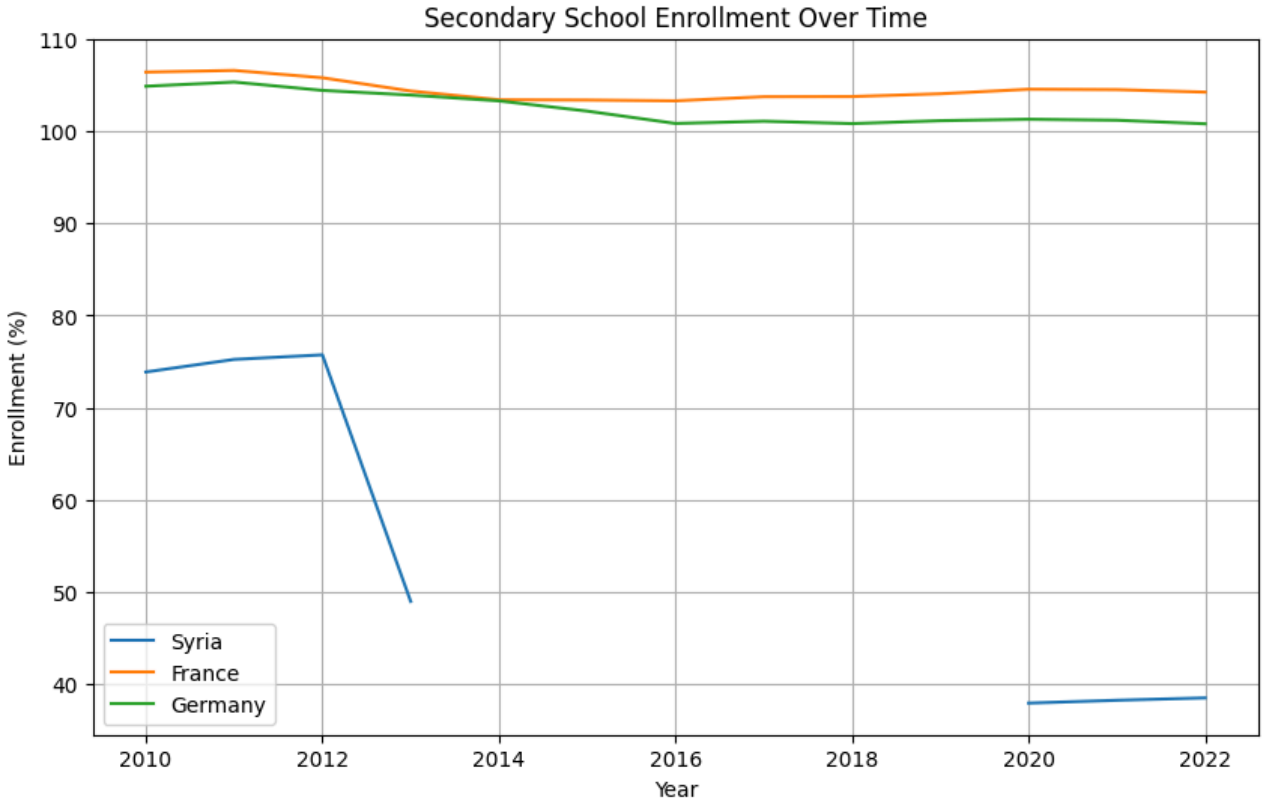
economic_data_2025.Unemployment Rate (%) 0

	country_id	year	gov_exp_pct_gdp	school_enrol_secondary_pct	school_enrol_tertiary_pct	e
count	0.0	35.000000	25.000000	33.000000	29.000000	
mean	NaN	2015.828571	5.234658	93.276423	59.997015	
std	NaN	3.951811	0.311923	21.534164	14.536478	
min	NaN	2010.000000	4.543890	37.921520	25.643700	
25%	NaN	2012.500000	4.935390	100.837303	55.625729	
50%	NaN	2015.000000	5.407170	103.308098	65.184479	
75%	NaN	2019.500000	5.480115	104.376602	68.870941	
max	NaN	2022.000000	5.659810	106.610641	77.365700	

economic_data_2025.Inflation (GDP Deflator, %) 0

dtype: int64

```
import matplotlib.pyplot as plt

countries = ['Syria', 'France', 'Germany']

plt.figure(figsize=(10,6))

for c in countries:
    temp = df[df['country'] == c]
    plt.plot(temp['year'], temp['school_enrol_secondary_pct'], label=c)

plt.legend()
plt.title("Secondary School Enrollment Over Time")
plt.xlabel("Year")
plt.ylabel("Enrollment (%)")
plt.grid(True)
plt.show()
```




```
syria = df[df['country'] == 'Syria']
syria.sort_values('year').head(10)
```

	country	country_id	year	gov_exp_pct_gdp	school_enrol_secondary_pct	school_enrol_tertiary_pct	e
26	Syria	NaN	2010	NaN	73.869118	25.688560	
27	Syria	NaN	2011	NaN	75.235291	25.643700	
28	Syria	NaN	2012	NaN	75.730698	30.272779	
29	Syria	NaN	2013	NaN	48.963531	33.139679	
30	Syria	NaN	2014	NaN	NaN	47.382881	
31	Syria	NaN	2015	NaN	NaN	51.165951	
32	Syria	NaN	2020	NaN	37.921520	NaN	
33	Syria	NaN	2021	NaN	38.228901	NaN	
34	Syria	NaN	2022	NaN	38.487030	NaN	

```
df.isna().mean() * 100
```

	0
country	0.000000
country_id	100.000000
year	0.000000
gov_exp_pct_gdp	28.571429
school_enrol_secondary_pct	5.714286
school_enrol_tertiary_pct	17.142857
economic_data_2025.Inflation (CPI %)	8.571429
economic_data_2025.GDP (Current USD)	0.000000
economic_data_2025.GDP per Capita (Current USD)	0.000000
economic_data_2025.Unemployment Rate (%)	0.000000
economic_data_2025.Inflation (GDP Deflator, %)	0.000000
economic_data_2025.GDP Growth (% Annual)	0.000000

dtype: float64

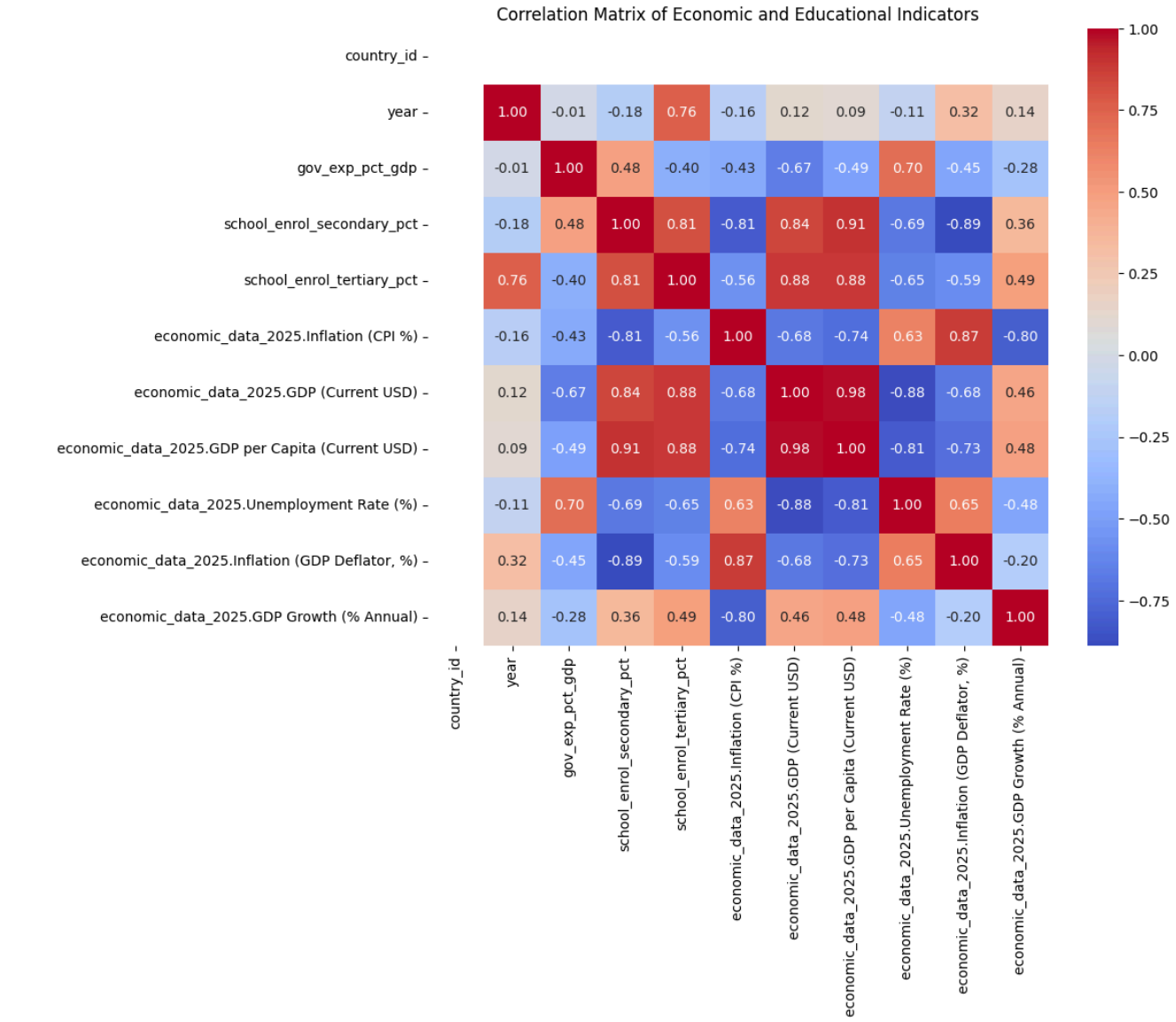
```
corr = df.corr(numeric_only=True)
corr
```

	country_id	year	gov_exp_pct_gdp	school_enrol_secondary_pct	sch
country_id	NaN	NaN	NaN	NaN	
year	NaN	1.000000	-0.008082	-0.184560	
gov_exp_pct_gdp	NaN	-0.008082	1.000000	0.478888	
school_enrol_secondary_pct	NaN	-0.184560	0.478888	1.000000	
school_enrol_tertiary_pct	NaN	0.763726	-0.404043	0.809172	
economic_data_2025.Inflation (CPI %)	NaN	-0.162215	-0.432105	-0.814616	
economic_data_2025.GDP (Current USD)	NaN	0.116521	-0.665393	0.843224	
economic_data_2025.GDP per Capita (Current USD)	NaN	0.090696	-0.488002	0.906919	
economic_data_2025.Unemployment Rate (%)	NaN	-0.113248	0.701429	-0.692103	
economic_data_2025.Inflation (GDP Deflator, %)	NaN	0.321141	-0.449158	-0.888832	
economic_data_2025.GDP Growth (% Annual)	NaN	0.142477	-0.279765	0.355232	

Next steps: Generate code with corr | New interactive sheet

```
import seaborn as sns
import matplotlib.pyplot as plt
```

```
plt.figure(figsize=(10,8))
sns.heatmap(corr, annot=True, fmt=".2f", cmap="coolwarm")
plt.title("Correlation Matrix of Economic and Educational Indicators")
plt.show()
```




```
df = df.rename(columns={
    'economic_data_2025.GDP per Capita (Current USD)': 'gdp_per_capita',
    'economic_data_2025.Unemployment Rate (%)': 'unemployment_rate'
})
```

```
df[['gdp_per_capita', 'unemployment_rate']].head()
```

	gdp_per_capita	unemployment_rate
0	40694.82117	9.279
1	42409.93570	7.043
2	43929.78409	9.228
3	47646.58204	5.967
4	40863.58144	9.841

```
analysis_cols = [
    'country',
    'year',
    'gdp_per_capita',
    'unemployment_rate',
    'school_enrol_secondary_pct',
    'school_enrol_tertiary_pct'
]

analysis_df = df[analysis_cols]
```

```
analysis_df.isna().sum()
```

	0
country	0
year	0
gdp_per_capita	0
unemployment_rate	0
school_enrol_secondary_pct	2
school_enrol_tertiary_pct	6

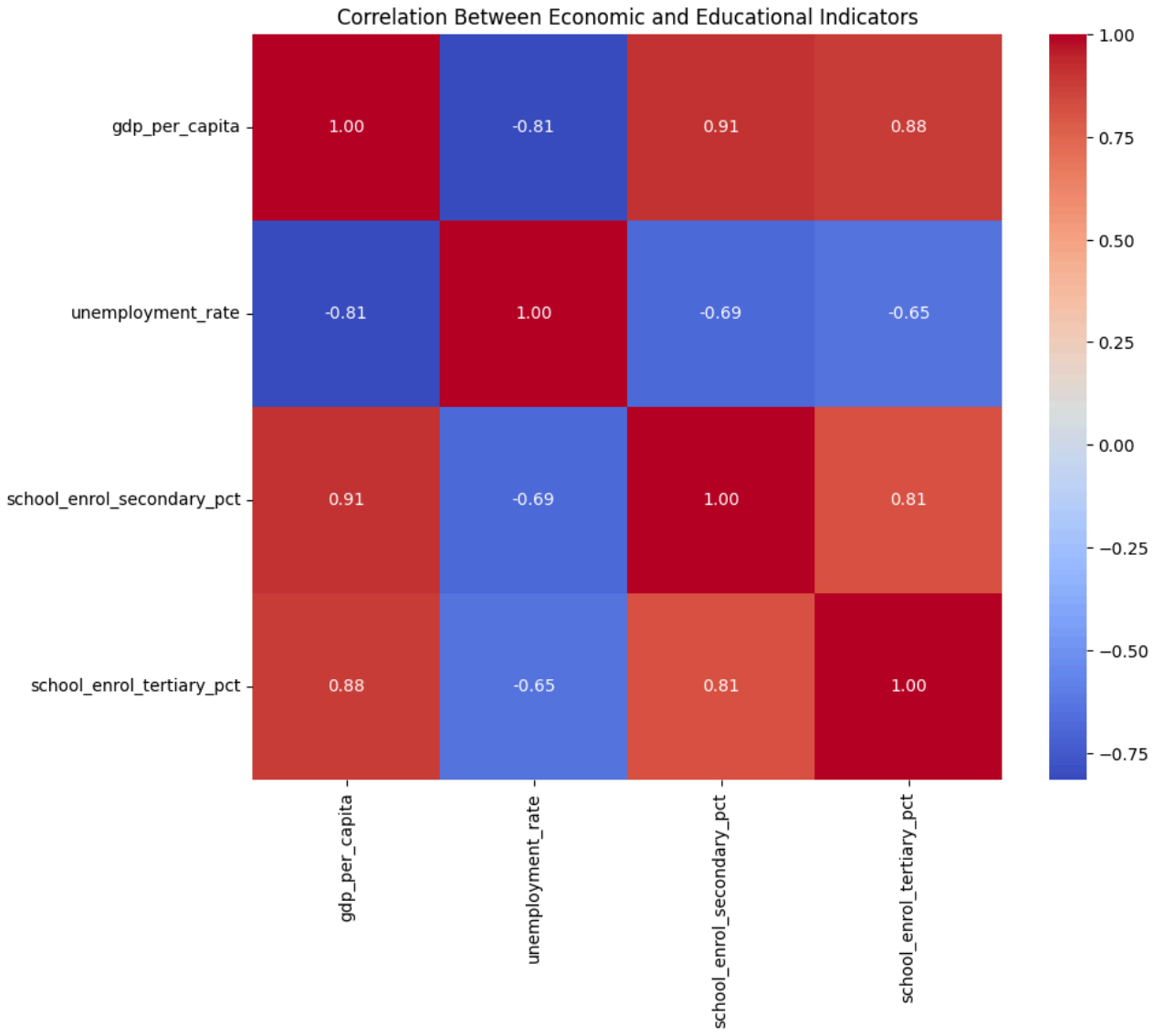
dtype: int64

```
analysis_df.describe()
```

	year	gdp_per_capita	unemployment_rate	school_enrol_secondary_pct	school_enrol_tertiary_
count	35.000000	35.000000	35.000000	33.000000	29.000
mean	2015.828571	32842.795368	8.434086	93.276423	59.997
std	3.951811	19083.061496	3.789539	21.534164	14.536
min	2010.000000	572.355290	3.120000	37.921520	25.643
25%	2012.500000	19827.287932	5.147500	100.837303	55.625
50%	2015.000000	41418.176650	8.610000	103.308098	65.184
75%	2019.500000	45131.094095	10.165000	104.376602	68.870
max	2022.000000	52265.654160	15.191000	106.610641	77.365

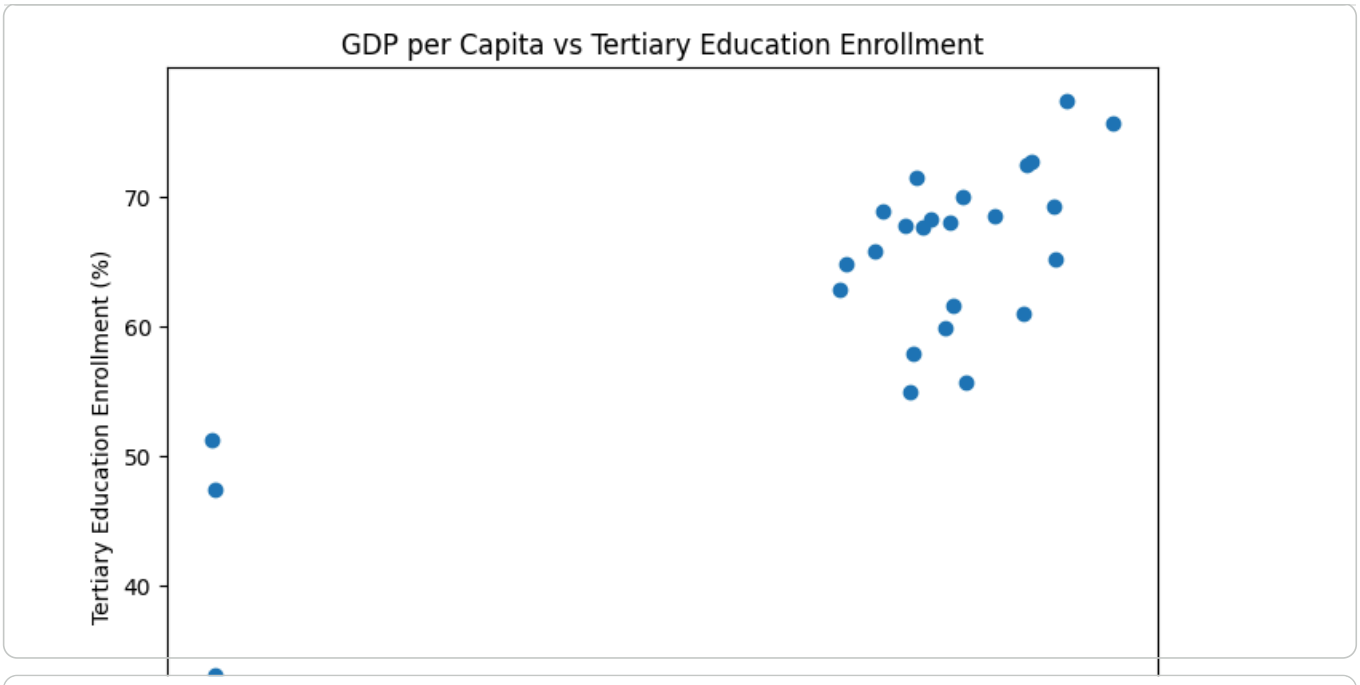
```
import seaborn as sns
import matplotlib.pyplot as plt

corr = analysis_df.drop(columns=['country', 'year']).corr()
```

```
plt.figure(figsize=(10,8))
sns.heatmap(corr, annot=True, fmt=".2f", cmap="coolwarm")
plt.title("Correlation Between Economic and Educational Indicators")
plt.show()
```




```
reg_df = analysis_df[['gdp_per_capita', 'school_enrol_tertiary_pct']].dropna()
```

```
plt.figure(figsize=(8,6))
plt.scatter(reg_df['gdp_per_capita'], reg_df['school_enrol_tertiary_pct'])
plt.xlabel('GDP per Capita (USD)')
plt.ylabel('Tertiary Education Enrollment (%)')
plt.title('GDP per Capita vs Tertiary Education Enrollment')
plt.show()
```




```
import statsmodels.api as sm

X = reg_df['gdp_per_capita']
y = reg_df['school_enrol_tertiary_pct']

X = sm.add_constant(X)

model = sm.OLS(y, X).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     school_enrol_tertiary_pct   R-squared:                       0.780
Model:                                   OLS   Adj. R-squared:                  0.772
Method:                        Least Squares   F-statistic:                     95.98
Date:                       Sun, 04 Jan 2026   Prob (F-statistic):           2.20e-10
Time:                               14:00:01   Log-Likelihood:                -96.279
No. Observations:                         29   AIC:                             196.6
Df Residuals:                             27   BIC:                             199.3
Df Model:                                  1
Covariance Type:                   nonrobust
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const             34.5890      2.896     11.945      0.000      28.648      40.530
gdp_per_capita     0.0007   7.43e-05      9.797      0.000       0.001       0.001
==============================================================================
Omnibus:                        0.057   Durbin-Watson:                   0.560
```