DATA 3421

GLOBAL BIRTH RATE PREDICTION: ANALYZING WORLD DATA

MARYAM AJIBOLA-ALIAS
ZEHRA ERDEN
THAO WELLS

CONTENTS

DATASET INTRODUCTION

> Global Country Information Dataset 2023

> source: Kaggle

> type: csv

> 195 rows

> 35 features

	Country	Density\n(P/Km2)	Abbreviation	Agricultural Land(%)	Land Area(Km2)	Armed Forces size	Birth Rate	Calling Code	Capital/Major City	Co2-Emissions
0	Afghanistan	60	AF	58.10%	652,230	323,000	32.49	93.0	Kabul	8,672
1	Albania	105	AL	43.10%	28,748	9,000	11.78	355.0	Tirana	4,536
2	Algeria	18	DZ	17.40%	2,381,741	317,000	24.28	213.0	Algiers	150,006
3	Andorra	164	AD	40.00%	468	NaN	7.20	376.0	Andorra la Vella	469
4	Angola	26	AO	47.50%	1,246,700	117,000	40.73	244.0	Luanda	34,693
...	...	...	...	...	...	...	...	...	...	...
190	Venezuela	32	VE	24.50%	912,050	343,000	17.88	58.0	Caracas	164,175
191	Vietnam	314	VN	39.30%	331,210	522,000	16.75	84.0	Hanoi	192,668
192	Yemen	56	YE	44.60%	527,968	40,000	30.45	967.0	Sanaa	10,609
193	Zambia	25	ZM	32.10%	752,618	16,000	36.19	260.0	Lusaka	5,141
194	Zimbabwe	38	ZW	41.90%	390,757	51,000	30.68	263.0	Harare	10,983

DATASET INTRODUCTION

Country: Name of the country.

Density (P/Km2): Population density measured in persons per square kilometer.

Abbreviation: Abbreviation or code representing the country.

Agricultural Land (%): Percentage of land area used for agricultural purposes.

Land Area (Km2): Total land area of the country in square kilometers.

Armed Forces Size: Size of the armed forces in the country.

Birth Rate: Number of births per 1,000 population per year.

Calling Code: International calling code for the country.

Capital/Major City: Name of the capital or major city.

CO2 Emissions: Carbon dioxide emissions in tons.

DATASET INTRODUCTION

CPI: Consumer Price Index, a measure of inflation and purchasing power.

CPI Change (%): Percentage change in the Consumer Price Index compared to the previous year.

Currency_Code: Currency code used in the country.

Fertility Rate: Average number of children born to a woman during her lifetime.

Forested Area (%): Percentage of land area covered by forests.

Gasoline_Price: Price of gasoline per liter in USD.

GDP: Gross Domestic Product, the total value of goods and services produced in the country.

Gross Primary Education Enrollment (%): Gross enrollment ratio for primary education.

Gross Tertiary Education Enrollment (%): Gross enrollment ratio for tertiary education.

DATASET INTRODUCTION

Infant Mortality: Number of deaths per 1,000 live births before reaching one year of age.

Largest City: Name of the country's largest city.

Life Expectancy: Average number of years a newborn is expected to live.

Maternal Mortality Ratio: Number of maternal deaths per 100,000 live births.

Minimum Wage: Minimum wage level in USD.

Official Language: Official language spoken in the country.

Out of Pocket Health Expenditure (%): Percentage of total health expenditure paid out-of-pocket by individuals.

Physicians per Thousand: Number of physicians per thousand people.

Population: Total population of the country.

Labor Force Participation (%): Percentage of the population that is part of the labor force.

DATASET INTRODUCTION

Tax Revenue (%): Tax revenue as a percentage of GDP.

Total Tax Rate: Overall tax burden as a percentage of commercial profits.

Unemployment Rate: Percentage of the labor force that is unemployed.

Urban Population: Percentage of the population living in urban areas.

Latitude: Latitude coordinate of the country's location.

Longitude: Longitude coordinate of the country's location.

DATA PREPROCESSING

```
Country                                     object
Density\n(P/Km2)                            object
Abbreviation                                object
Agricultural Land( %)                       object
Land Area(Km2)                              object
Armed Forces size                           object
Birth Rate                                 float64
Calling Code                               float64
Capital/Major City                          object
Co2-Emissions                               object
CPI                                         object
CPI Change (%)                              object
Currency-Code                               object
Fertility Rate                             float64
Forested Area (%)                           object
Gasoline Price                              object
GDP                                         object
Gross primary education enrollment (%)      object
Gross tertiary education enrollment (%)     object
Infant mortality                           float64
Largest city                                object
Life expectancy                            float64
Maternal mortality ratio                   float64
Minimum wage                                object
Official language                           object
Out of pocket health expenditure            object
Physicians per thousand                    float64
Population                                  object
Population: Labor force participation (%)   object
Tax revenue (%)                             object
Total tax rate                              object
Unemployment rate                           object
Urban_population                            object
Latitude                                   float64
Longitude                                  float64
```

> object to float

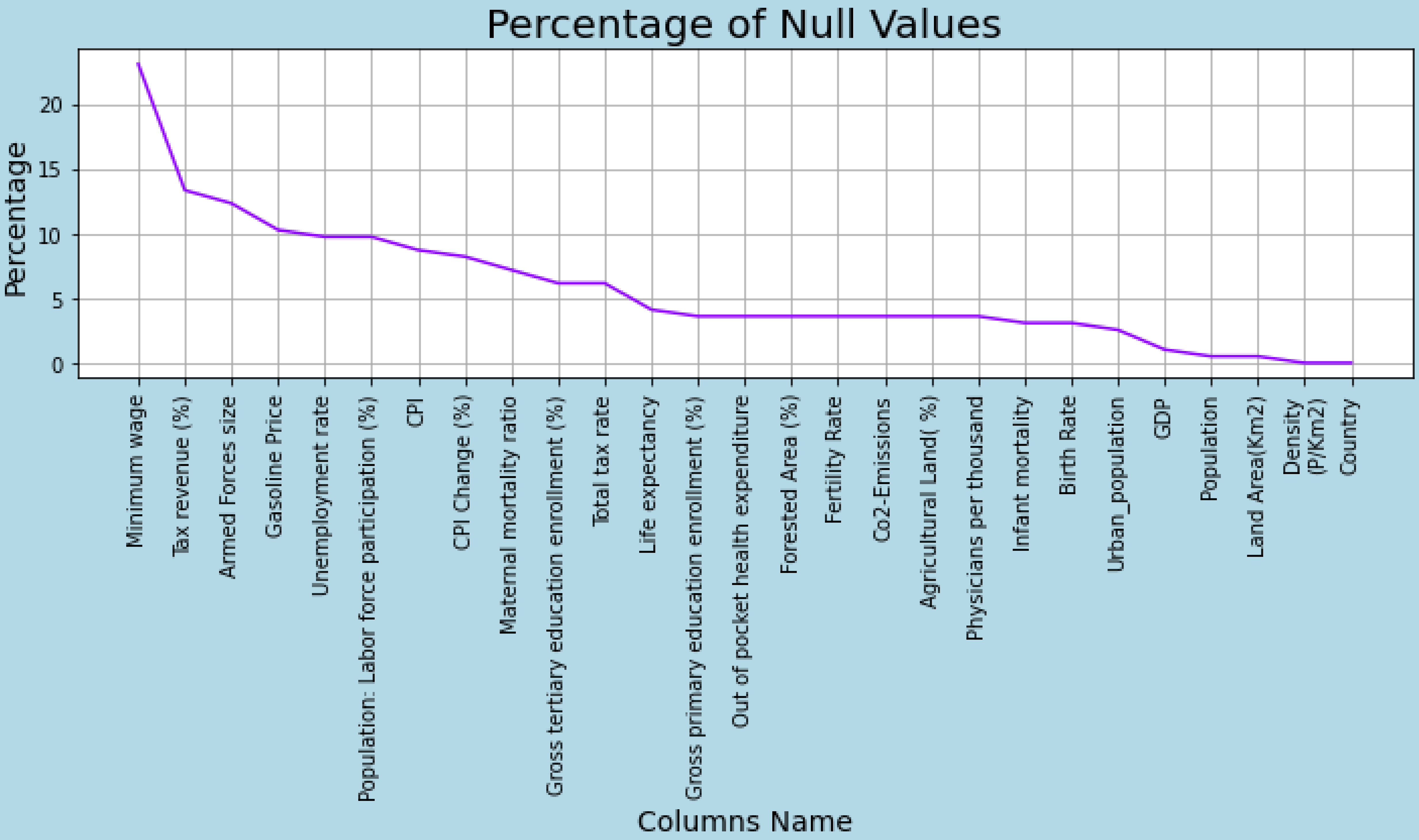
> removing '%', '$', ','.

>removed columns that are not necessary

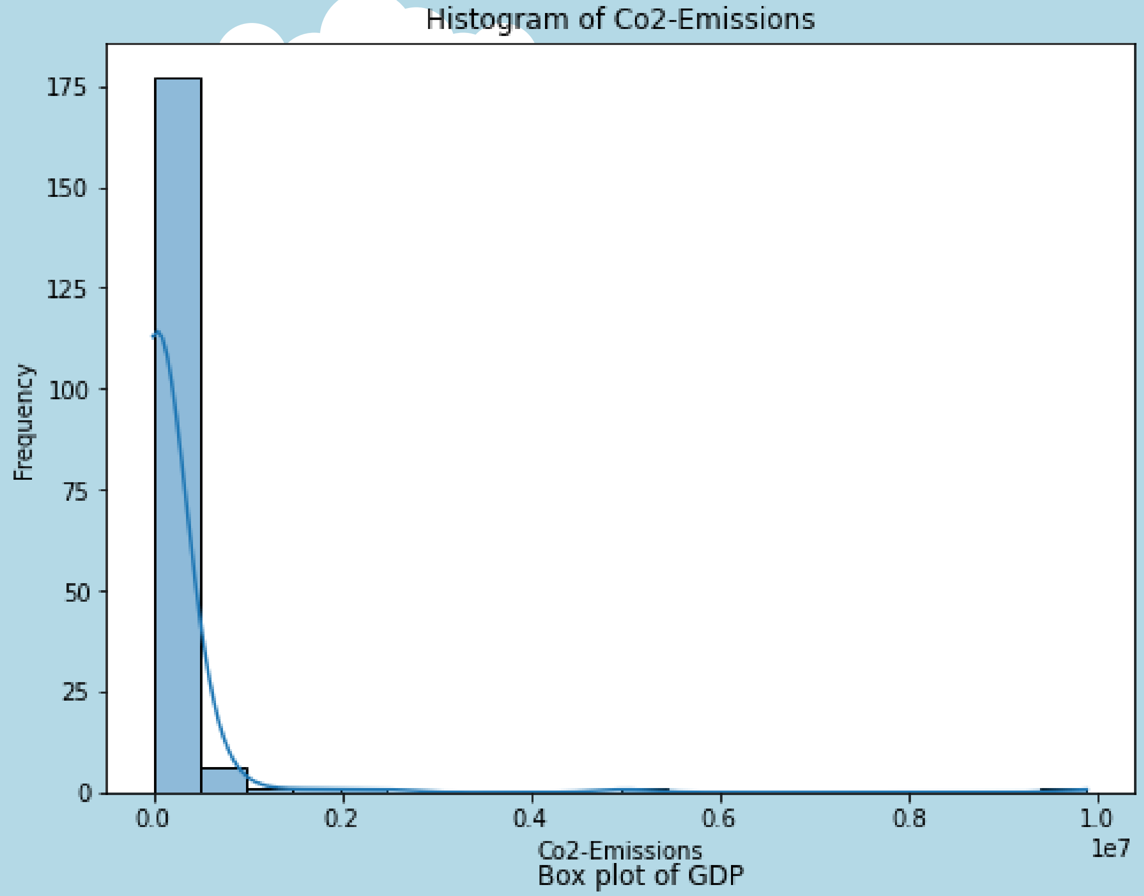
'Abbreviation', 'Calling Code', 'Capital/Major City', 'Currency-Code',

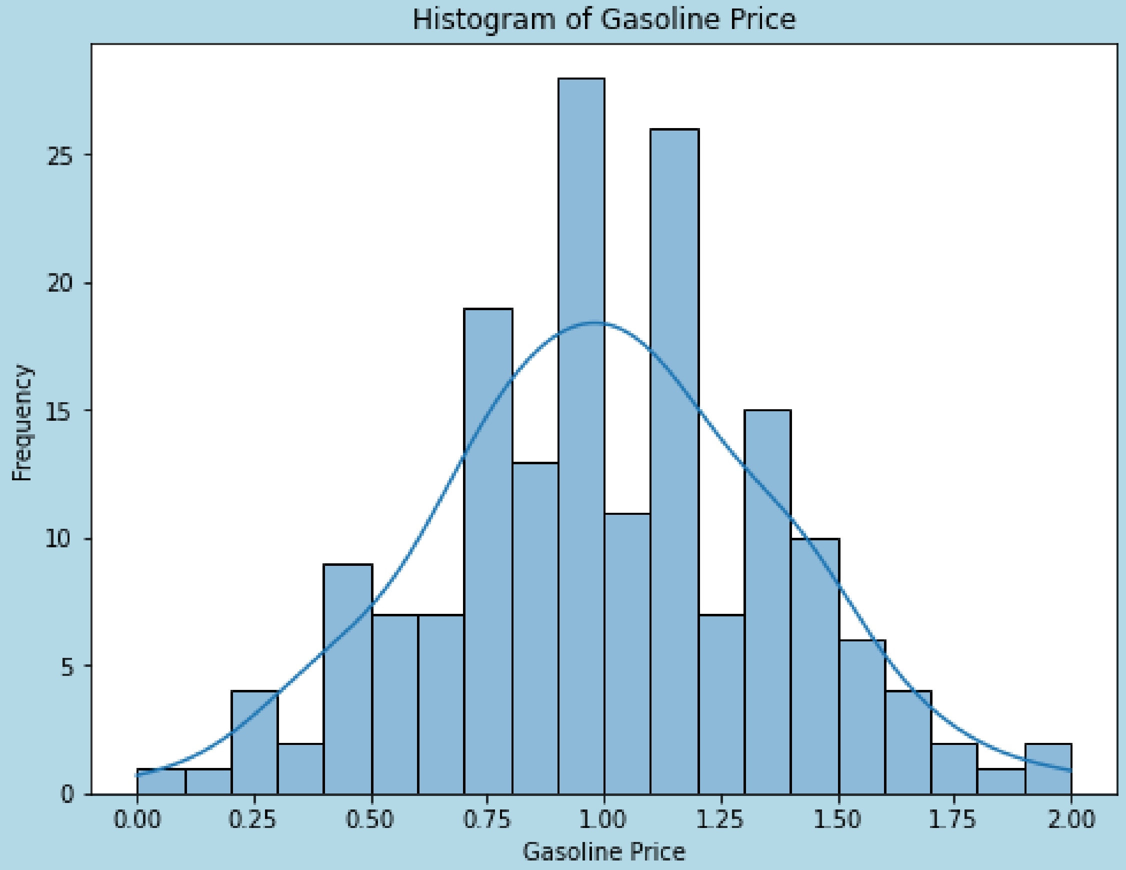
'Largest city', 'Official language', 'Latitude', 'Longitude'

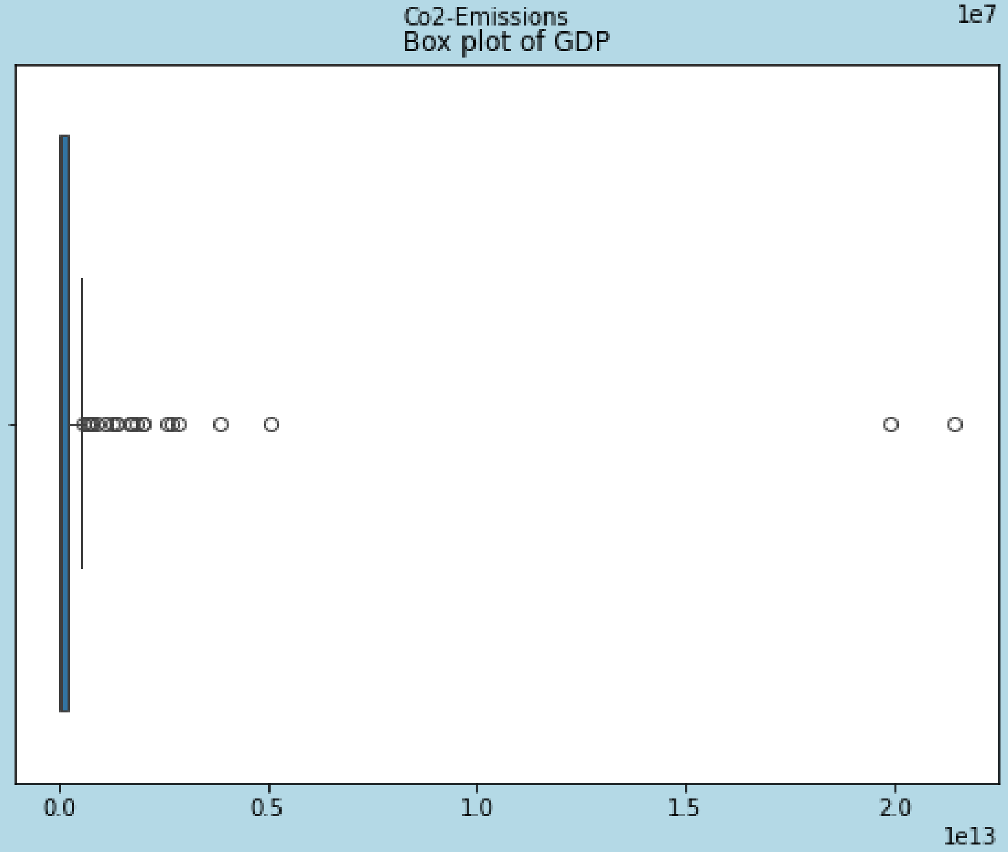
DATA PREPROCESSING



DATA PREPROCESSING







> mean imputation for normally distributed features:

- **Agricultural Land(%)**
- **Population: Labor force participation (%)**
- **Tax revenue (%)**
- **Gasoline Price**

>median imputation for the rest

DATA PREPROCESSING

- we have many features with outliers present.
- these outliers being the countries with high performance mostly and their data is important.
- we decided to keep outliers but deal with them in the transformation section.

DATA TRANSFORMATION

> normalization (min-max scaling)

- goal is to bring all features to a similar scale
- used to scale features to a specific range, typically between 0 and 1

$$x' = \frac{x - \mu}{\max(x) - \min(x)}$$

- less affected by outliers compared to standardization

> standardization (z-score normalization)

- assumes that the data follows a normal distribution
- sensitive to outliers because it uses the mean and standard deviation

$$Z = \frac{x - \mu}{\sigma}$$

> robust normalization

- subtracts the median of the feature and then divides by the interquartile range (IQR)
- suitable when the data contains outliers or features with non-normal distributions

Robust Standardised Value

Original Value

Sample Median

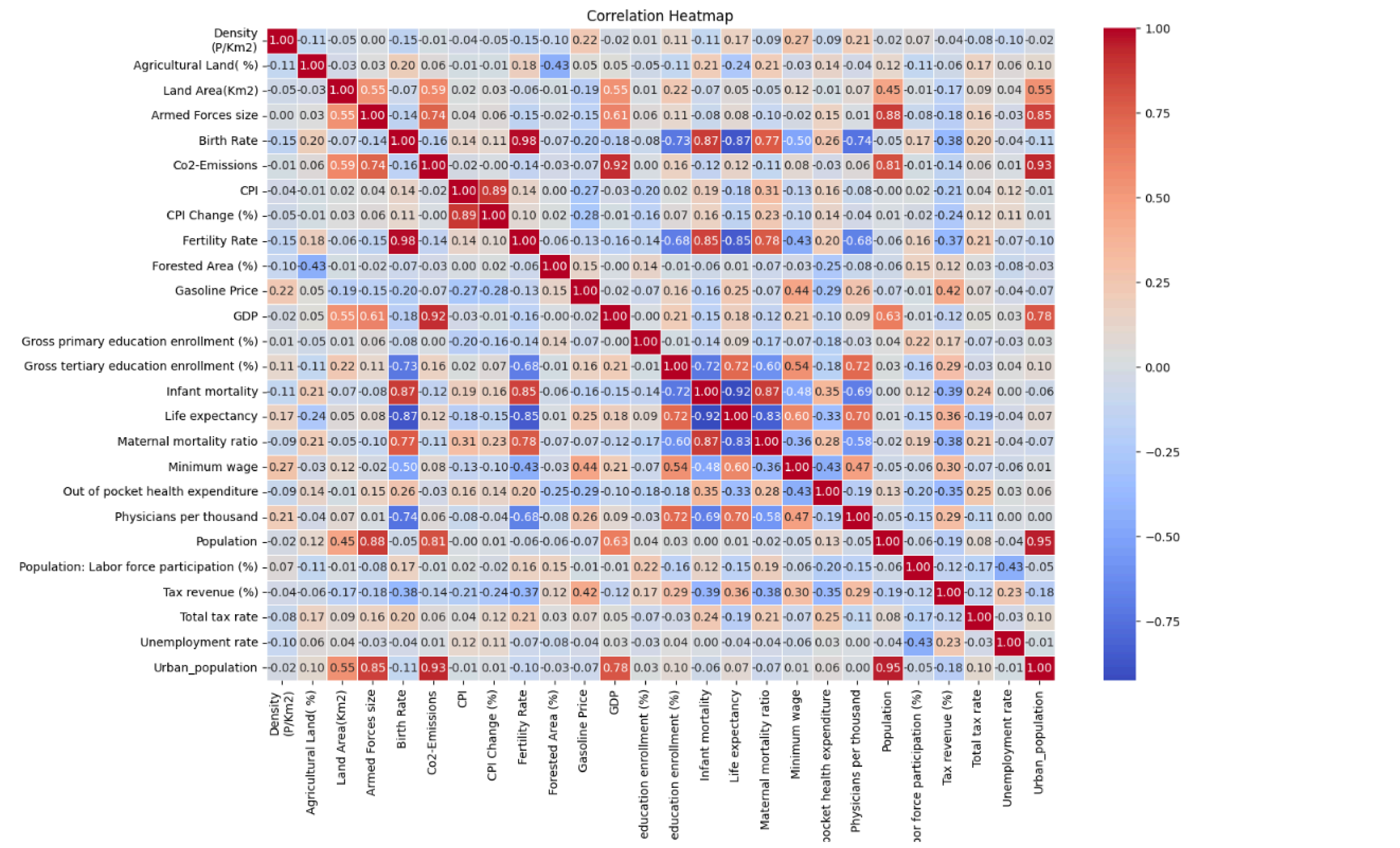
$$x' = \frac{x - \text{median}(x)}{(Q3-Q1)}$$

Interquartile Range = Q3 – Q1

DATA UNDERSTANDING

DATA UNDERSTANDING

Correlation Matrix



DATA UNDERSTANDING

Since our focus is birth rate, the features that have high correlations and one that seem important to birth rates are Fertility Rate, Infant mortality, Maternal mortality rate, Life expectancy, Gross tertiary education enrollment(%), Physician per thousands. and the Minimum wage.

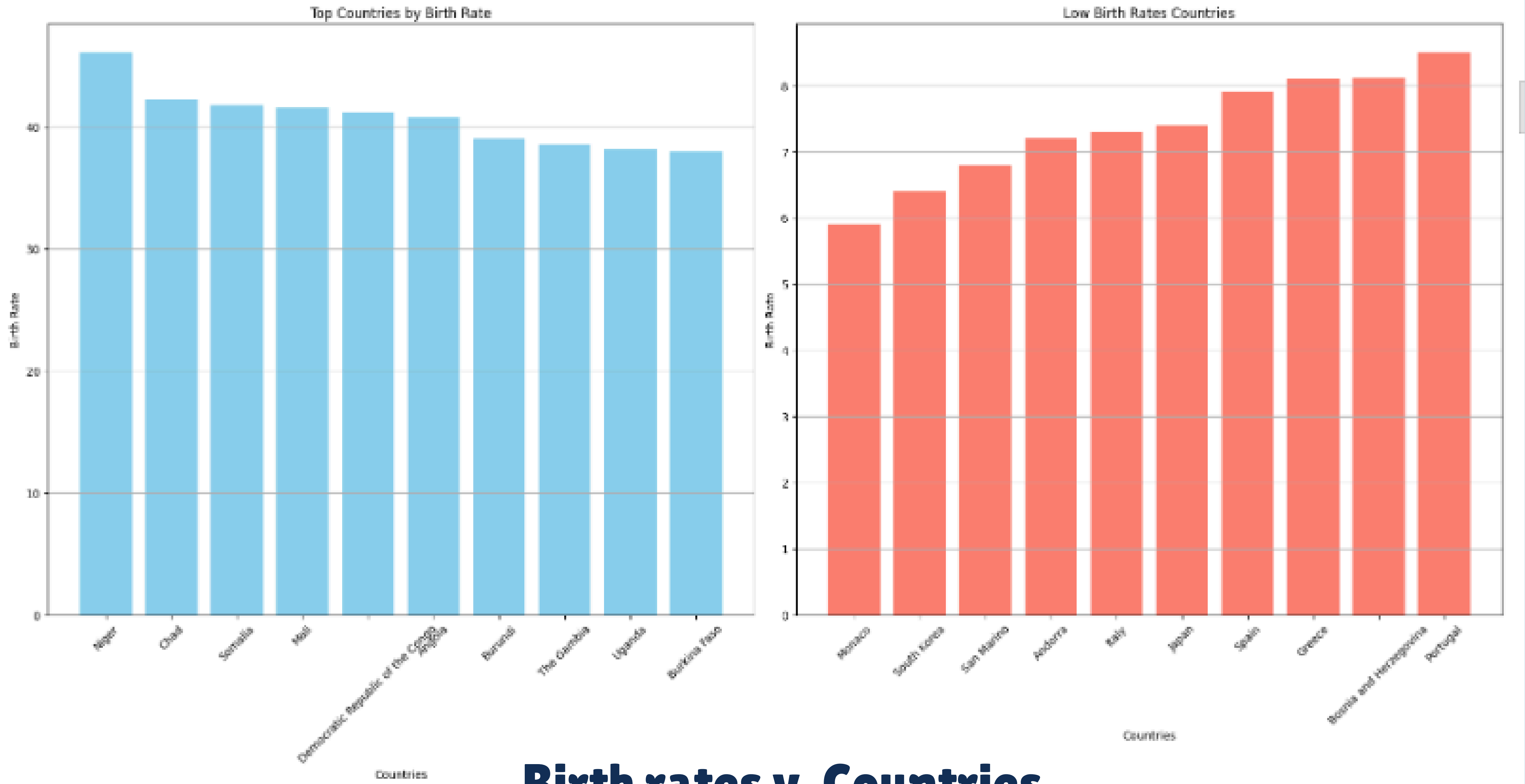
DATA UNDERSTANDING

```
#birth rate insights

birth_rate_high = df.sort_values('Birth Rate',ascending=False)
birth_rate_high_countries = birth_rate_high.head(10)[['Country','Birth Rate']]

birth_rate_low = df.sort_values('Birth Rate',ascending=True)
birth_rate_low_countries = birth_rate_low.head(10)[['Country','Birth Rate']]

fig, axis = plt.subplots(nrows=1, ncols=2, figsize=(19, 10))

axis[0].bar(birth_rate_high_countries['Country'], birth_rate_high_countries['Birth Rate'], color='skyblue')
axis[0].set_title('Top Countries by Birth Rate')
axis[0].set_xlabel('Countries')
axis[0].set_ylabel('Birth Rate')
axis[0].tick_params(axis='x', rotation=45)
axis[0].grid(axis='y')

axis[1].bar(birth_rate_low_countries['Country'], birth_rate_low_countries['Birth Rate'], color='sal
axis[1].set_title('Low Birth Rates Countries')
axis[1].set_xlabel('Countries')
axis[1].set_ylabel('Birth Rate')
axis[1].tick_params(axis='x', rotation=45)
axis[1].grid(axis='y')

plt.tight_layout()
plt.show()
```

The goal is just to better understand the relationship "Birth Rates" and other features that have shown to have influence with the birth rates. And showcasing the relationship by comparing it to the top 10 and 10 last countries. This is showcasing top 10 countries with highest and lowest birth rates.

Top 10 Countries w/ high birth rates:
Niger, Chad, Somalia, Mali, DR Congo, Angola, Burundi, The Gambia, Uganda, & Burkina Faso.

Top 10 Countries w/ Low birth rates:
Monaco, South Korea, San Marino, Andorra, Italy, Japan, Spain, Greece, Bosnia and Herzegovina, and Portugal.

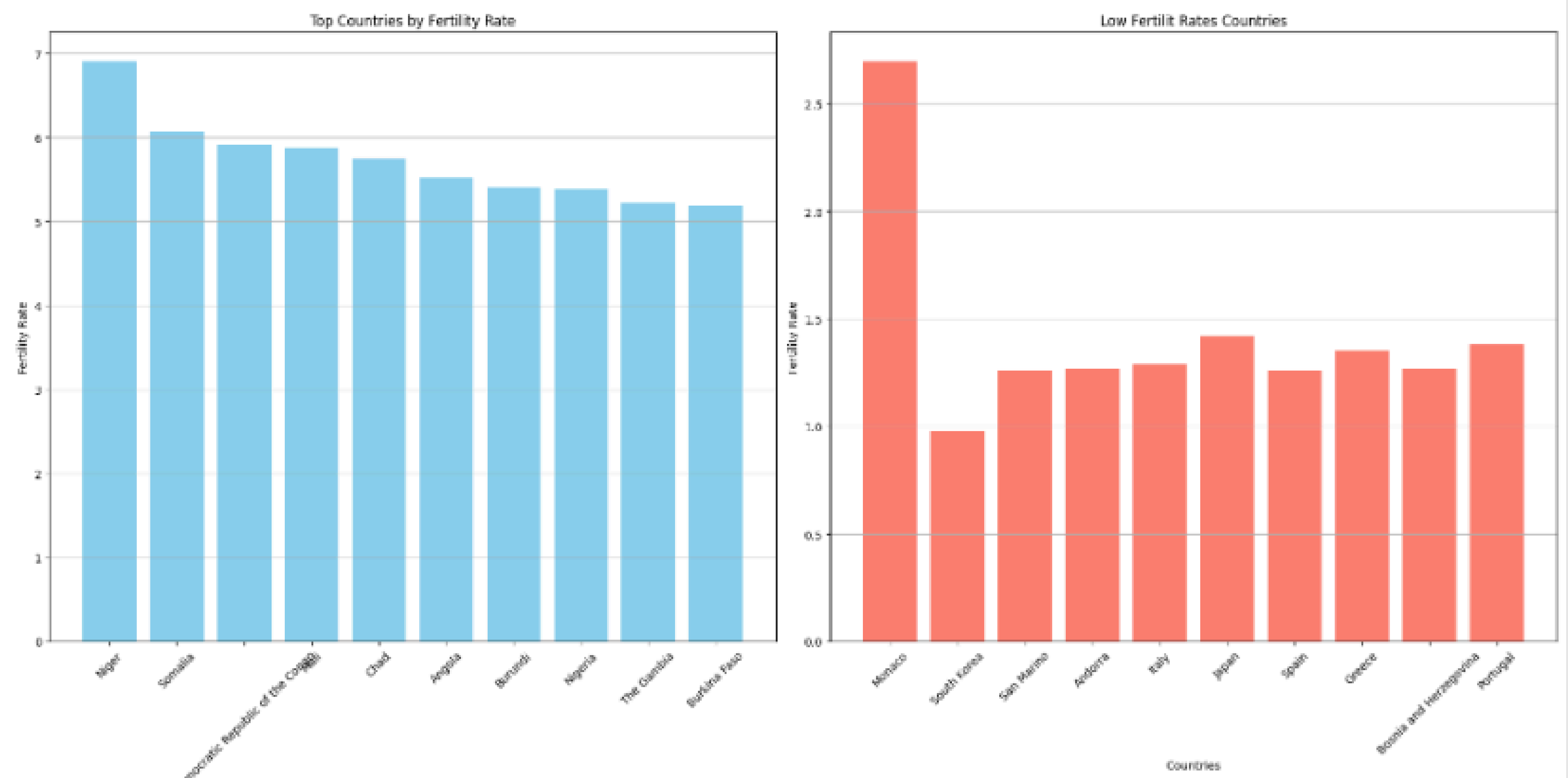


Birth rates v. Countries

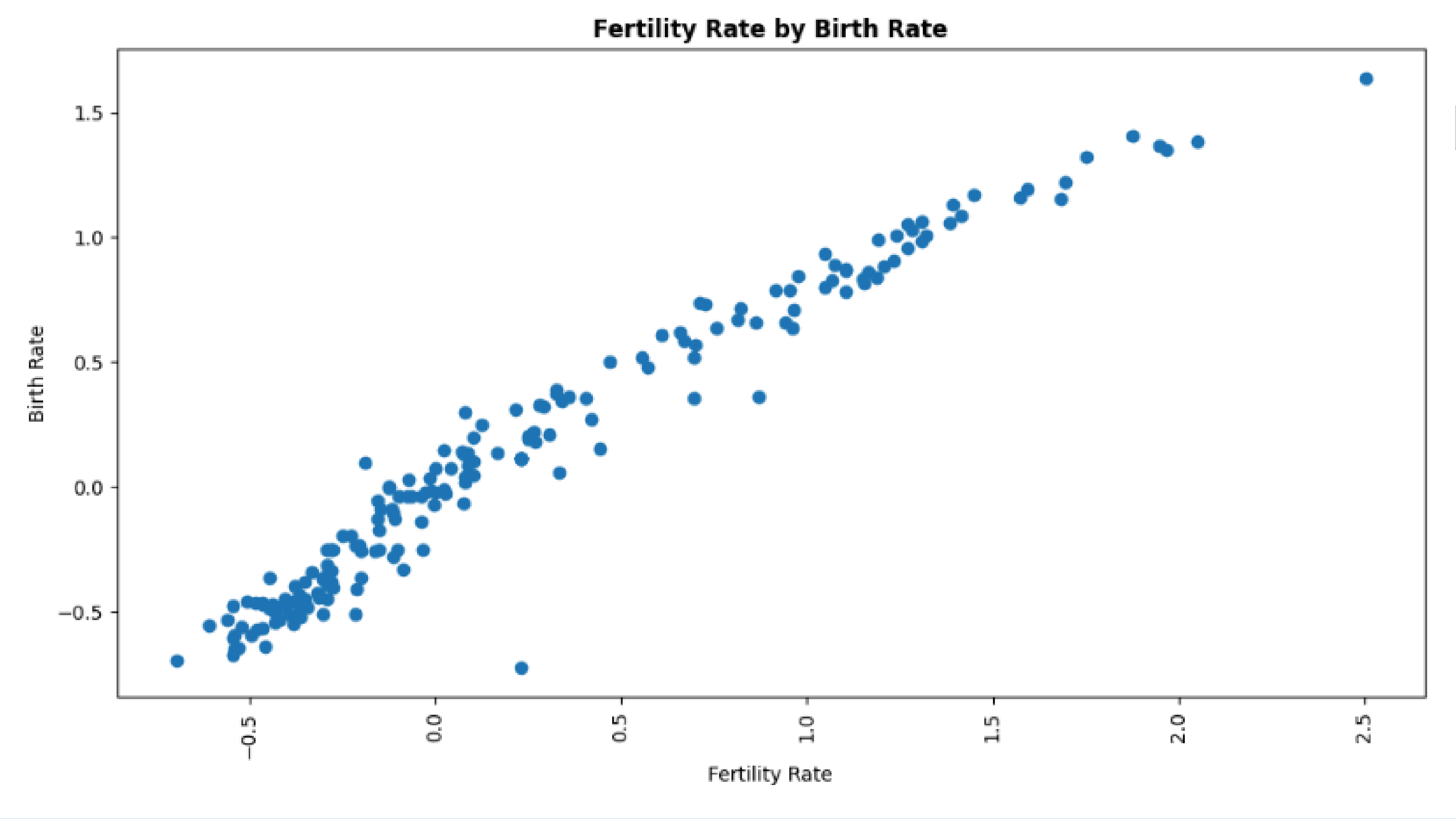
DATA UNDERSTANDING

```
fertility_rate_high = df.sort_values('Fertility Rate',ascending=False)
fertility_rate_high_countries = fertility_rate_high.head(10)[['Country','Fertility Rate']]

fertility_rate_low = df.sort_values('Birth Rate',ascending=True)
fertility_rate_low_countries = fertility_rate_low.head(10)[['Country','Fertility Rate']]

fig, axis = plt.subplots(nrows=1, ncols=2, figsize=(19, 10))

axis[0].bar(fertility_rate_high_countries['Country'], fertility_rate_high_countries['Fertility Rate'], color='skyblue')
axis[0].set_title('Top Countries by Fertility Rate')
axis[0].set_xlabel('Countries')
axis[0].set_ylabel('Fertility Rate')
axis[0].tick_params(axis='x', rotation=45)
axis[0].grid(axis='y')

axis[1].bar(fertility_rate_low_countries['Country'], fertility_rate_low_countries['Fertility Rate'], color='salmon')
axis[1].set_title('Low Fertilit Rates Countries')
axis[1].set_xlabel('Countries')
axis[1].set_ylabel('Fertility Rate')
axis[1].tick_params(axis='x', rotation=45)
axis[1].grid(axis='y')

plt.tight_layout()
plt.show()
```

Fertility Rates v. Countries



Knowing Fertility rates has a positive relationship with birth rates, we also check whether they would be similarities in the countries w/ high & low birth rates/ fertility rates, and the results were almost the same, with Nigeria been new on the graph and on countries with high fertility rate. while for the low births and fertility rates countries were the same.

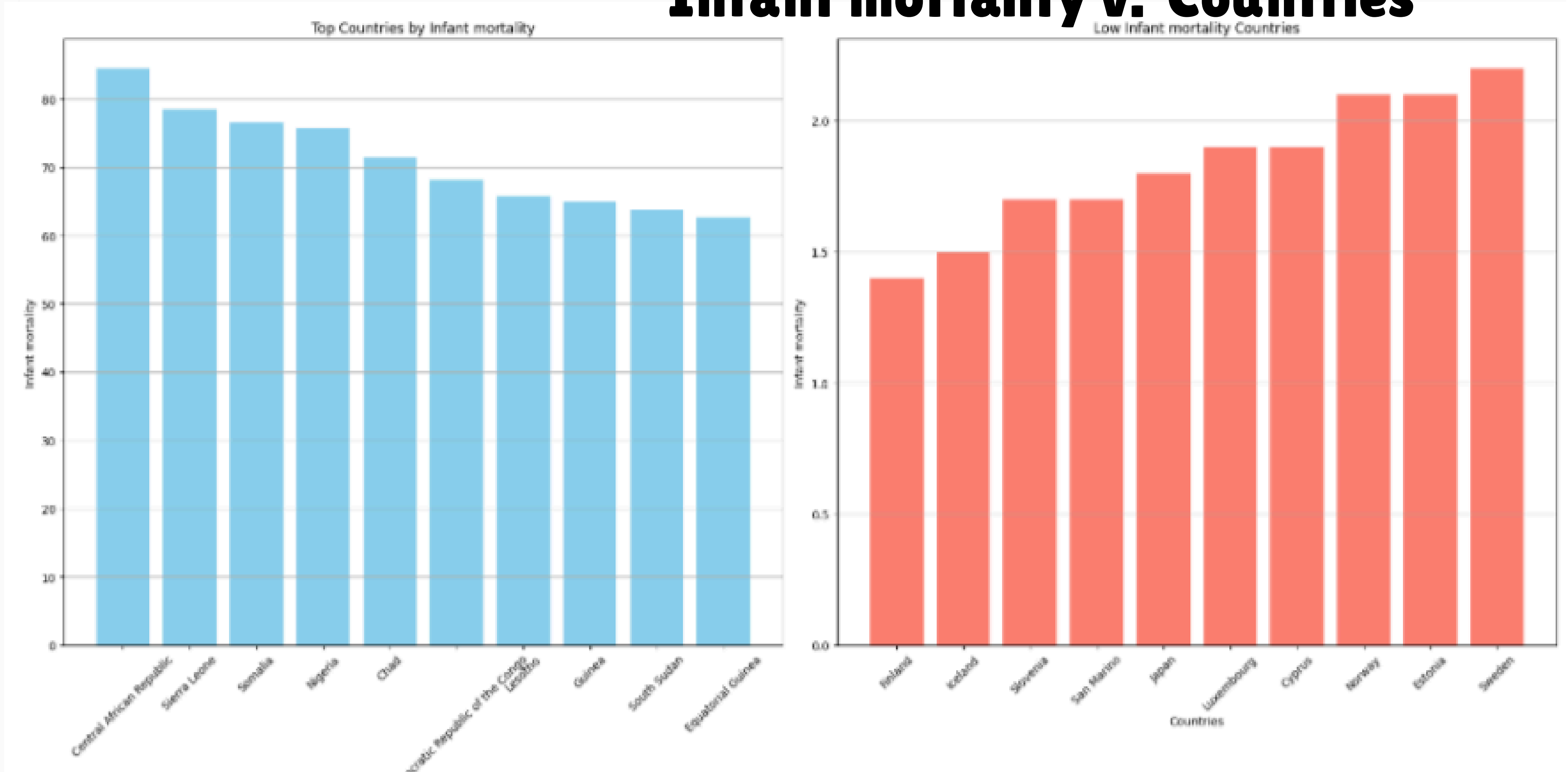


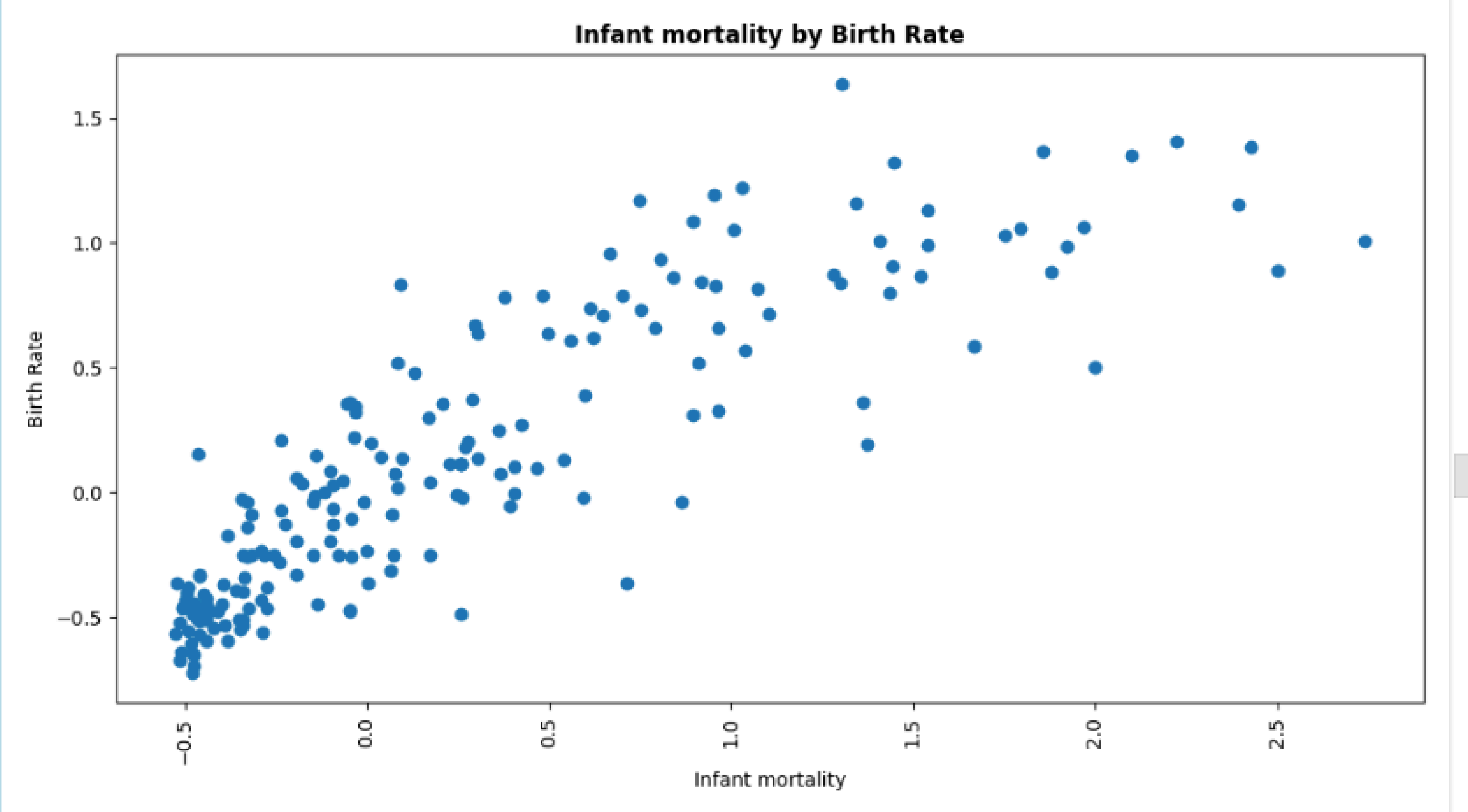
Fertility Rates v. Birth Rates

DATA UNDERSTANDING

```
Infantm_high = df.sort_values('Infant mortality',ascending=False)
Infantm_high_countries = Infantm_high.head(10)[['Country','Infant mortality']]

Infantm_low = df.sort_values('Infant mortality',ascending=True)
Infantm_low_countries = Infantm_low.head(10)[['Country','Infant mortality']]

fig, axis = plt.subplots(nrows=1, ncols=2, figsize=(19, 10))

axis[0].bar(Infantm_high_countries['Country'], Infantm_high_countries['Infant mortality'], color='skyblue')
axis[0].set_title('Top Countries by Infant mortality')
axis[0].set_xlabel('Countries')
axis[0].set_ylabel('Infant mortality')
axis[0].tick_params(axis='x', rotation=45)
axis[0].grid(axis='y')

axis[1].bar(Infantm_low_countries['Country'],Infantm_low_countries['Infant mortality'], color='salmon')
axis[1].set_title('Low Infant mortality Countries')
axis[1].set_xlabel('Countries')
axis[1].set_ylabel('Infant mortality')
axis[1].tick_params(axis='x', rotation=45)
axis[1].grid(axis='y')

plt.tight_layout()
plt.show()
```

Infant mortality v. Countries





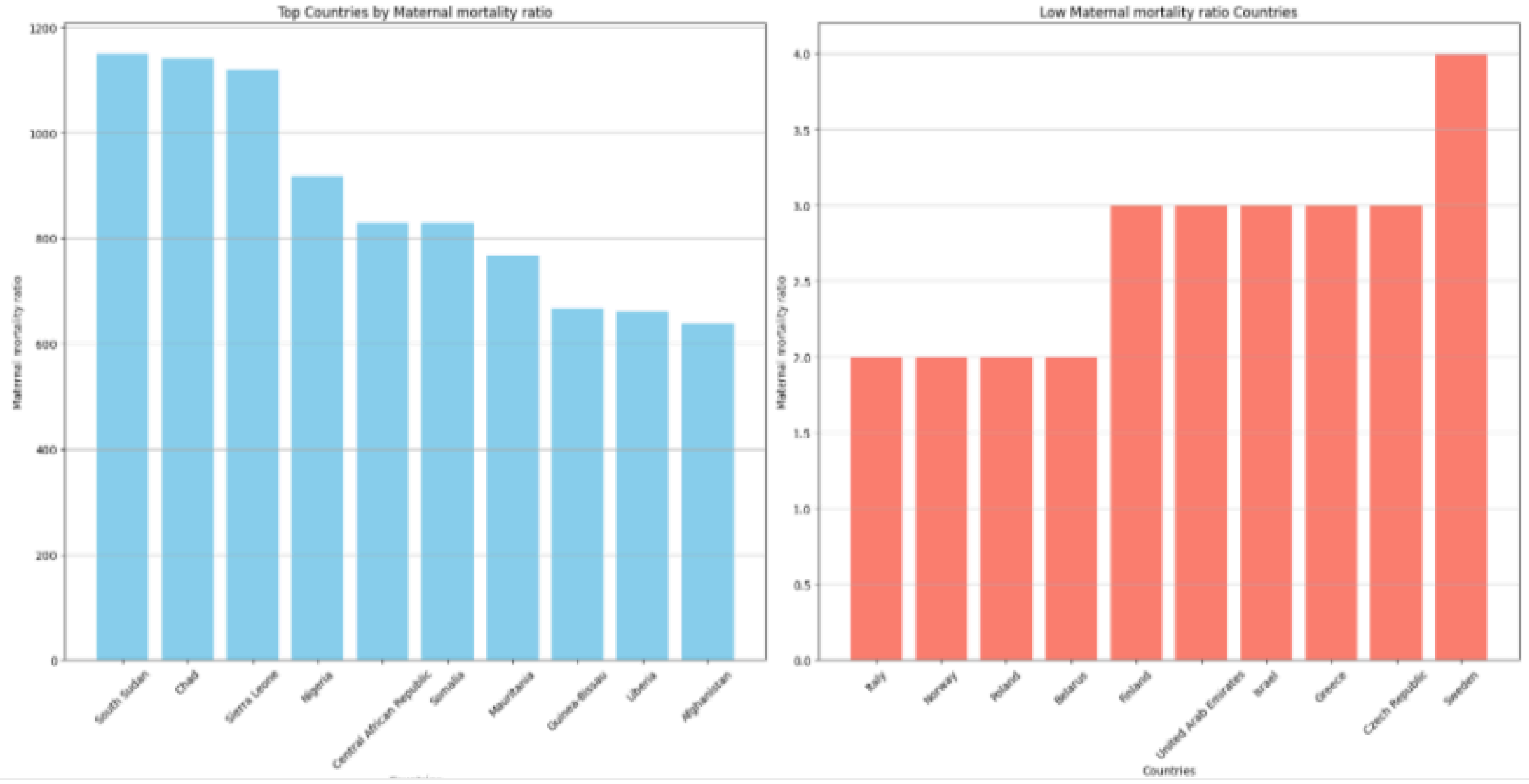
Infant mortality v. Birth rates

Infant mortality has a positive relationship with birth rates, we also check whether they would be similarities in the countries with high & low birth rate, and the countries with higher birth rates showed up on ones with high infant mortality, i.e Chad, Congo, & Somalia, for the countries with low birth rates showed up on low infant mortality countries, i.e Japan.

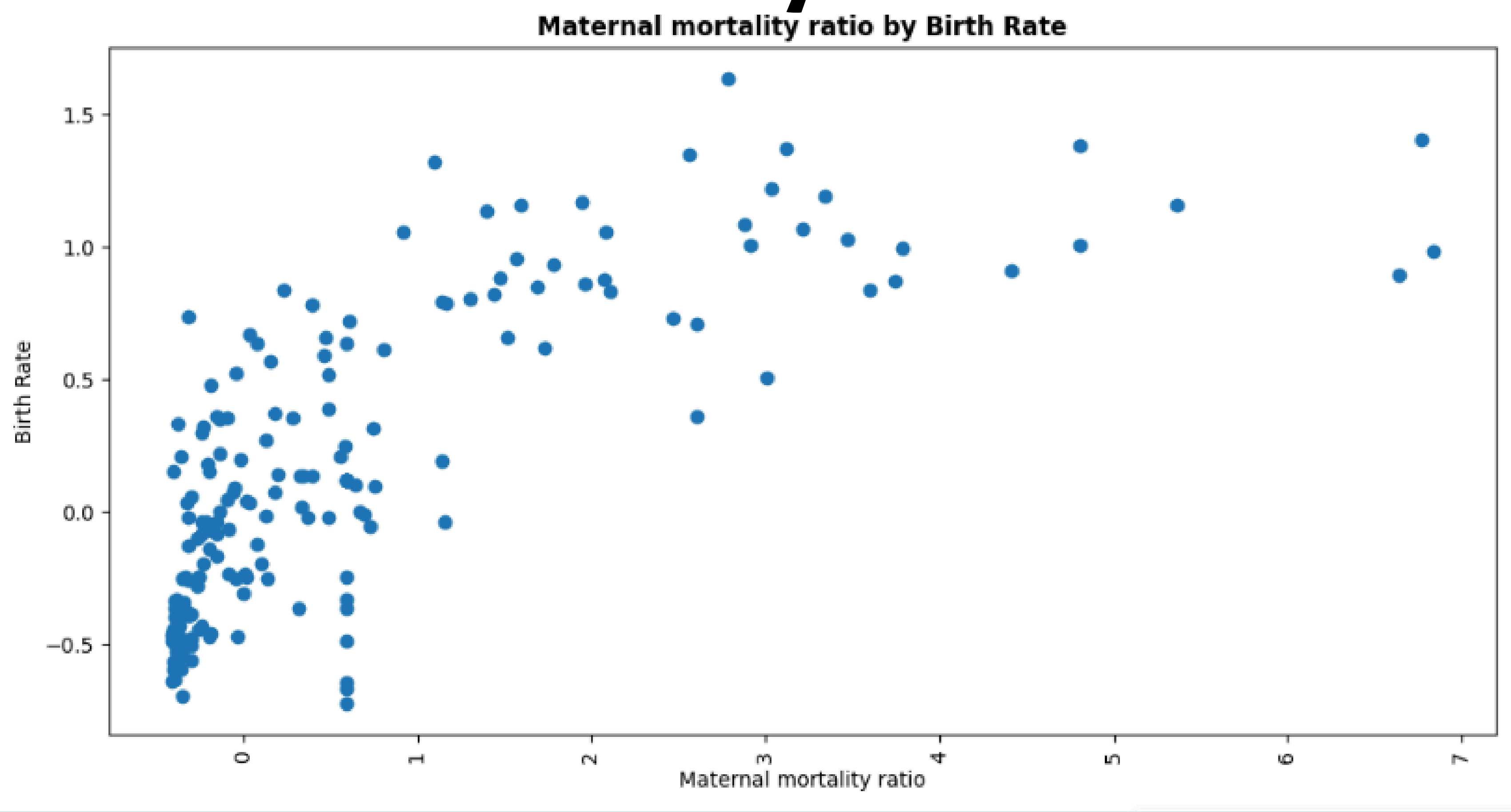
DATA UNDERSTANDING

```
mmr_high = df.sort_values('Maternal mortality ratio',ascending=False)
mmr_high_countries = mmr_high.head(10)[['Country','Maternal mortality ratio']]

mmr_low = df.sort_values('Maternal mortality ratio',ascending=True)
mmr_low_countries = mmr_low.head(10)[['Country','Maternal mortality ratio']]

fig, axis = plt.subplots(nrows=1, ncols=2, figsize=(19, 10))

axis[0].bar(mmr_high_countries['Country'], mmr_high_countries['Maternal mortality ratio'], color='skyblue')
axis[0].set_title('Top Countries by Maternal mortality ratio')
axis[0].set_xlabel('Countries')
axis[0].set_ylabel('Maternal mortality ratio')
axis[0].tick_params(axis='x', rotation=45)
axis[0].grid(axis='y')

axis[1].bar(mmr_low_countries['Country'],mmr_low_countries['Maternal mortality ratio'], color='salmon')
axis[1].set_title('Low Maternal mortality ratio Countries')
axis[1].set_xlabel('Countries')
axis[1].set_ylabel('Maternal mortality ratio')
axis[1].tick_params(axis='x', rotation=45)
axis[1].grid(axis='y')

plt.tight_layout()
plt.show()
```

Maternal mortality ratio v. Countries



Maternal mortality ratio has a positive relationship with birth rates, the only countries that was in the top high birth rates were that were also in high maternal mortality ratio were Chad & Somalia, while for low maternal mortality rate countries with low birth rates that showed up was just Italy.

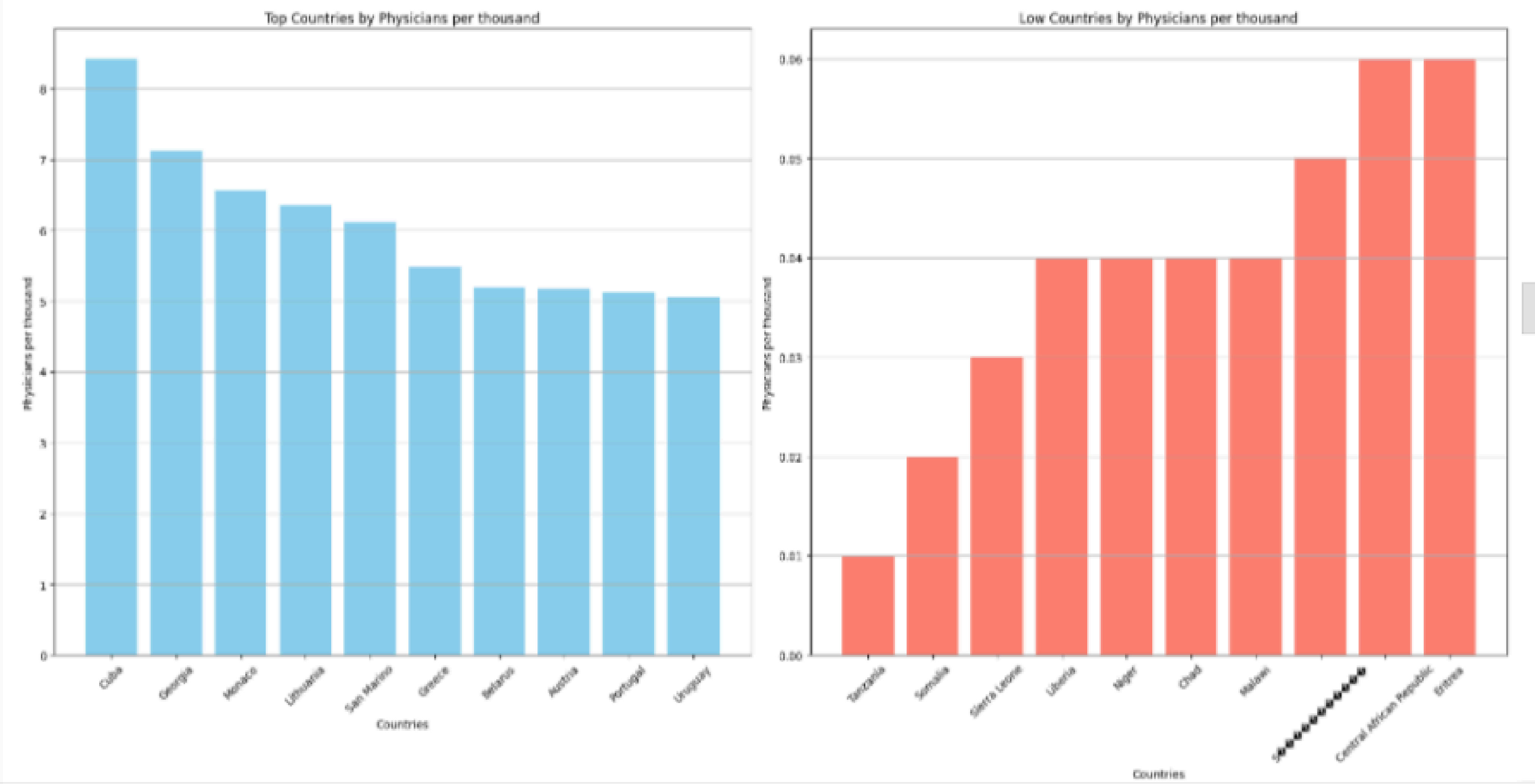


Maternal mortality ratio v. birth rates

DATA UNDERSTANDING

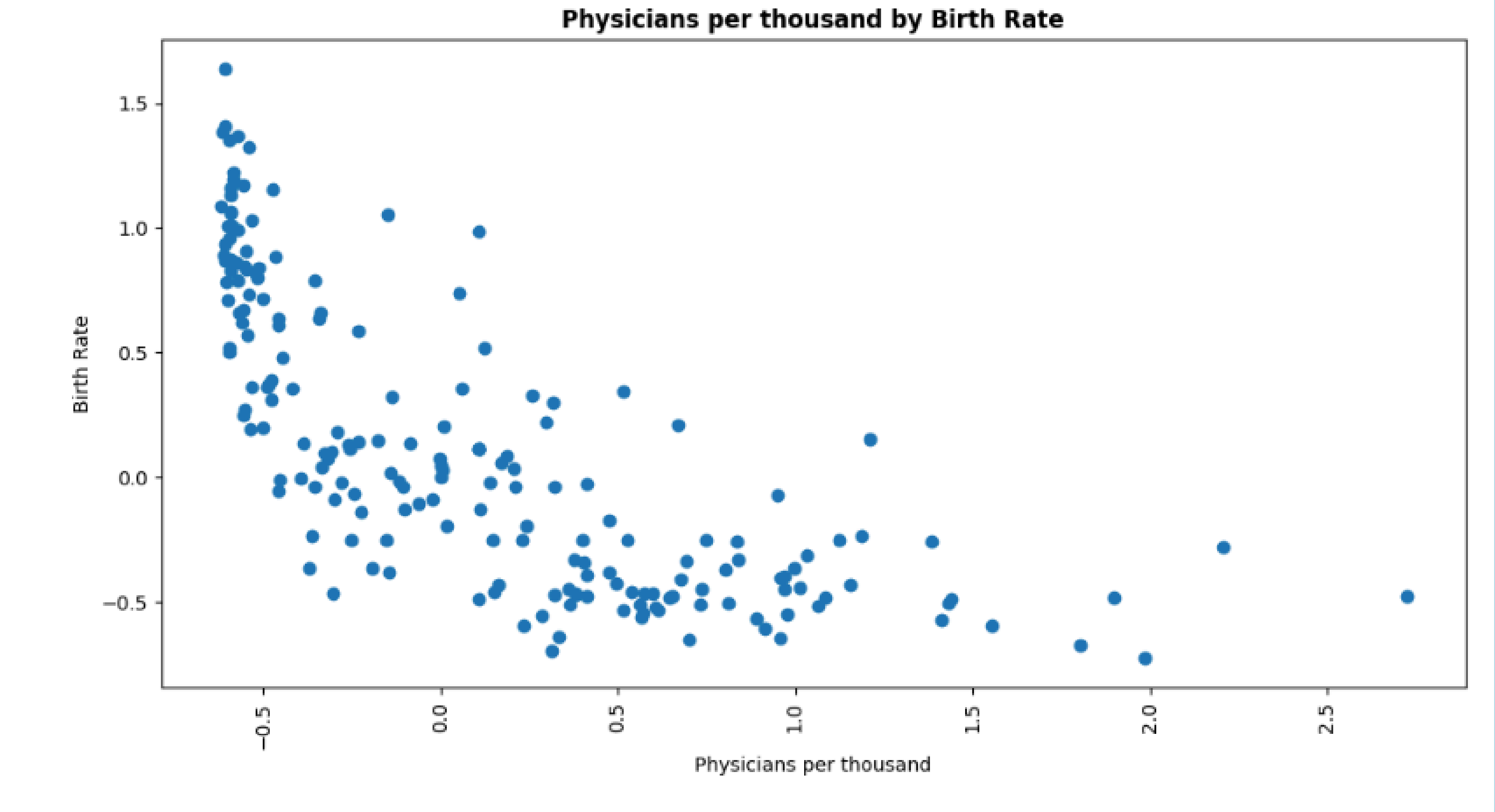
```python
physician_high = df.sort_values('Physicians per thousand',ascending=False)
physician_high_countries = physician_high.head(10)[['Country','Physicians per thousand']]

physician_low = df.sort_values('Physicians per thousand',ascending=True)
physician_low_countries = physician_low.head(10)[['Country','Physicians per thousand']]

fig, axis = plt.subplots(nrows=1, ncols=2, figsize=(19, 10))

axis[0].bar(physician_high_countries['Country'], physician_high_countries['Physicians per thousand'], color='skyblue')
axis[0].set_title('Top Countries by Physicians per thousand')
axis[0].set_xlabel('Countries')
axis[0].set_ylabel('Physicians per thousand')
axis[0].tick_params(axis='x', rotation=45)
axis[0].grid(axis='y')

axis[1].bar(physician_low_countries['Country'],physician_low_countries['Physicians per thousand'], color='salmon')
axis[1].set_title('Low Countries by Physicians per thousand')
axis[1].set_xlabel('Countries')
axis[1].set_ylabel('Physicians per thousand')
axis[1].tick_params(axis='x', rotation=45)
axis[1].grid(axis='y')

plt.tight_layout()
plt.show()
```

Physicians per thousand v. Countries



Physicians per thousand has a negativie correlation with birth rates, so in this case none of the countries with higher birth rate showed up instead, we found countries w/ low birth rates like Greece , San Marino, & Portugal, and its the opposite in countries with low physicians per capital, countries with high birth rates were found like Chad, Niger, & Somalia.

```
Text(0.5, 1.0, 'Physicians per thousand by Birth Rate')
```

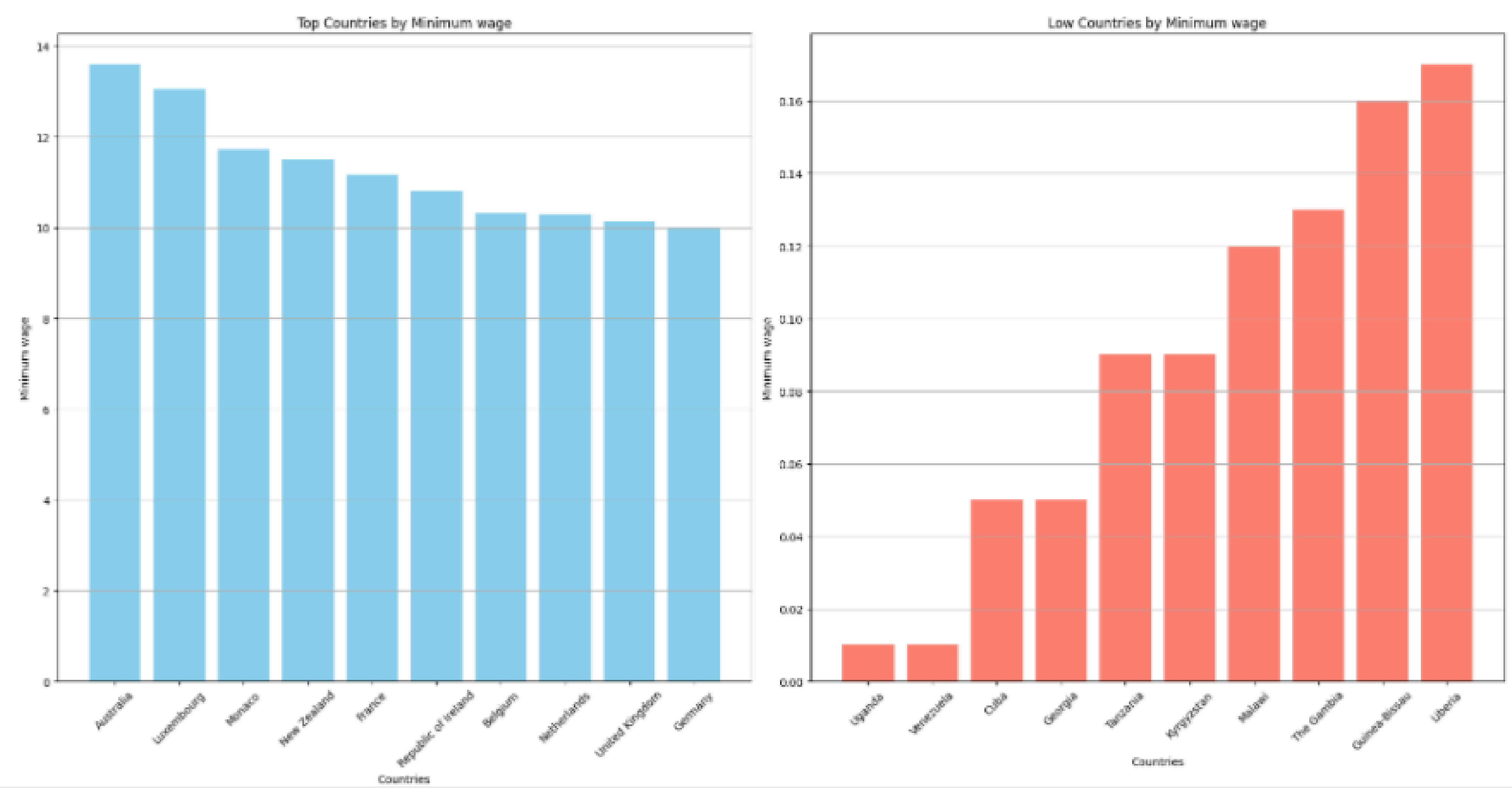


Physicians per thousand v. Birth rates

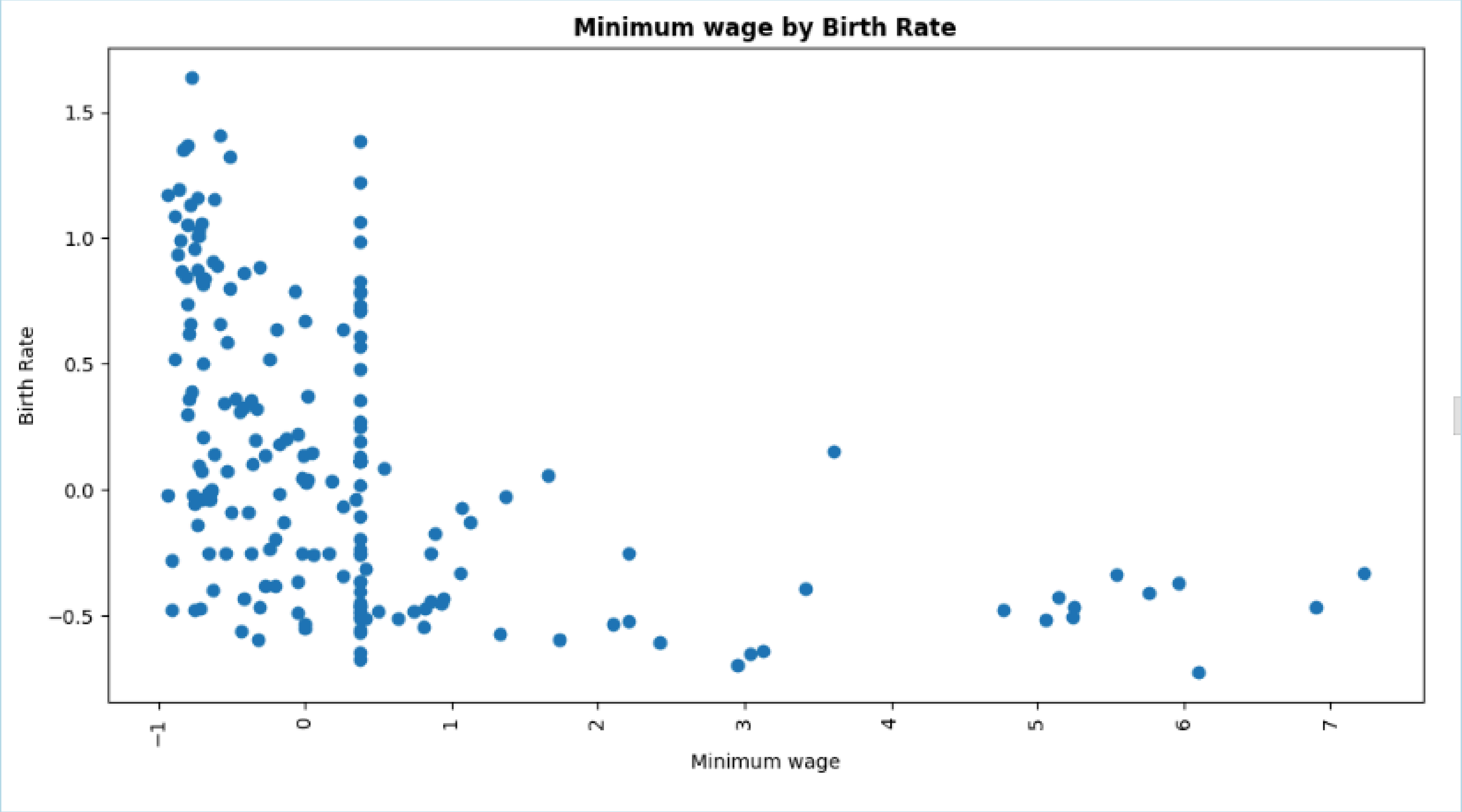
DATA UNDERSTANDING

```
wage_high = df.sort_values('Minimum wage',ascending=False)
wage_high_countries = wage_high.head(10)[['Country','Minimum wage']]

wage_low = df.sort_values('Minimum wage',ascending=True)
wage_low_countries = wage_low.head(10)[['Country','Minimum wage']]

fig, axis = plt.subplots(nrows=1, ncols=2, figsize=(19, 10))

axis[0].bar(wage_high_countries['Country'], wage_high_countries['Minimum wage'], color='skyblue')
axis[0].set_title('Top Countries by Minimum wage')
axis[0].set_xlabel('Countries')
axis[0].set_ylabel('Minimum wage')
axis[0].tick_params(axis='x', rotation=45)
axis[0].grid(axis='y')

axis[1].bar(wage_low_countries['Country'],wage_low_countries['Minimum wage'], color='salmon')
axis[1].set_title('Low Countries by Minimum wage')
axis[1].set_xlabel('Countries')
axis[1].set_ylabel('Minimum wage')
axis[1].tick_params(axis='x', rotation=45)
axis[1].grid(axis='y')

plt.tight_layout()
plt.show()
```

Minimum wage v. Countries



Minimum wage has close to zero correlation to birth rates, so almost no countries with high birth rates were found in the one with higher minimum wage , but instead found one with low birth rates, i.e Monaco and while in countries with low minimum wage, we found just two countries with higher birth rate, i.e Uganda and Gambia.



Minimum wage v. Birth rates

MODEL BUILDING

- Random Forest
- XGBoost
- Regression
- KNN

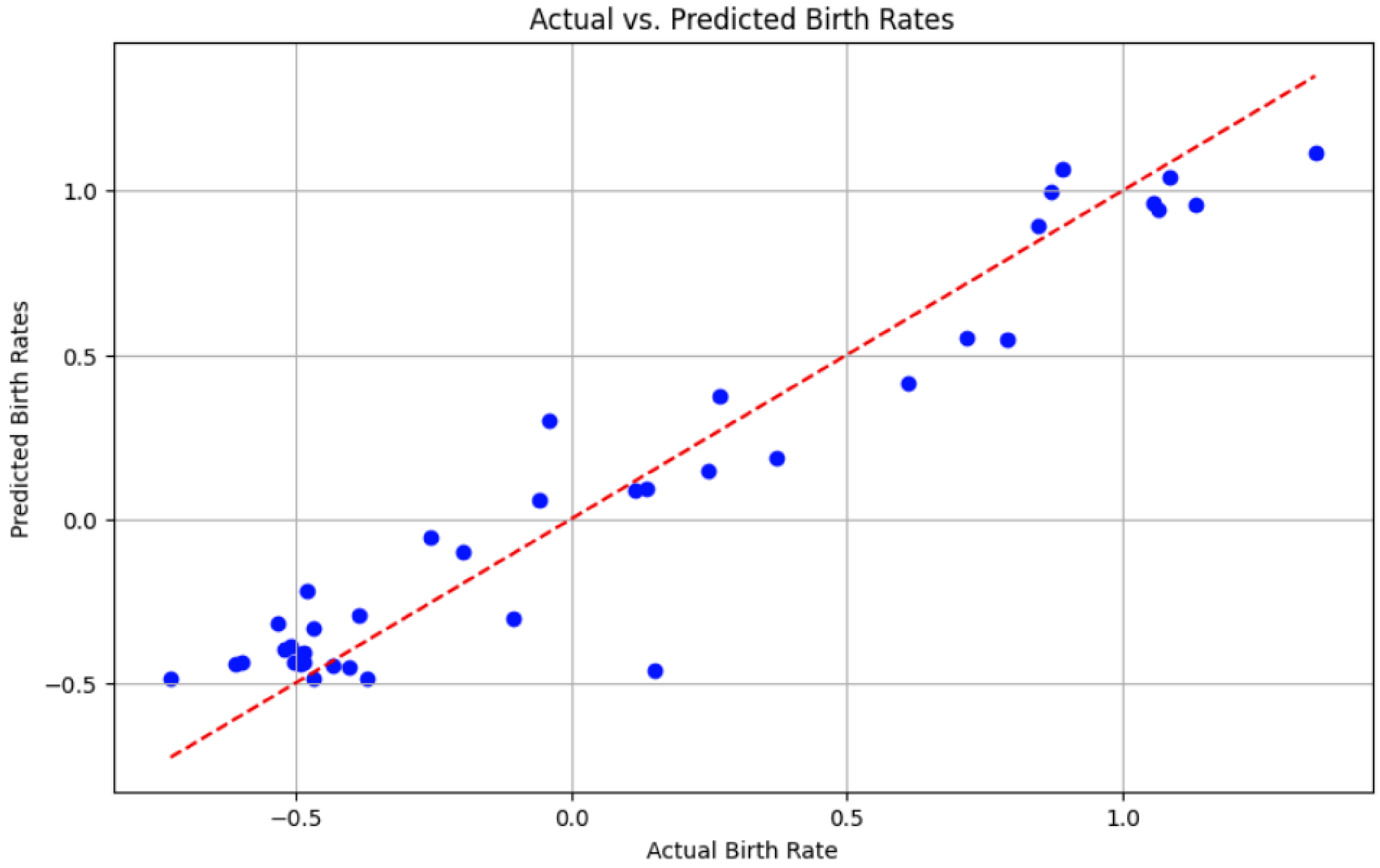
RANDOM FOREST

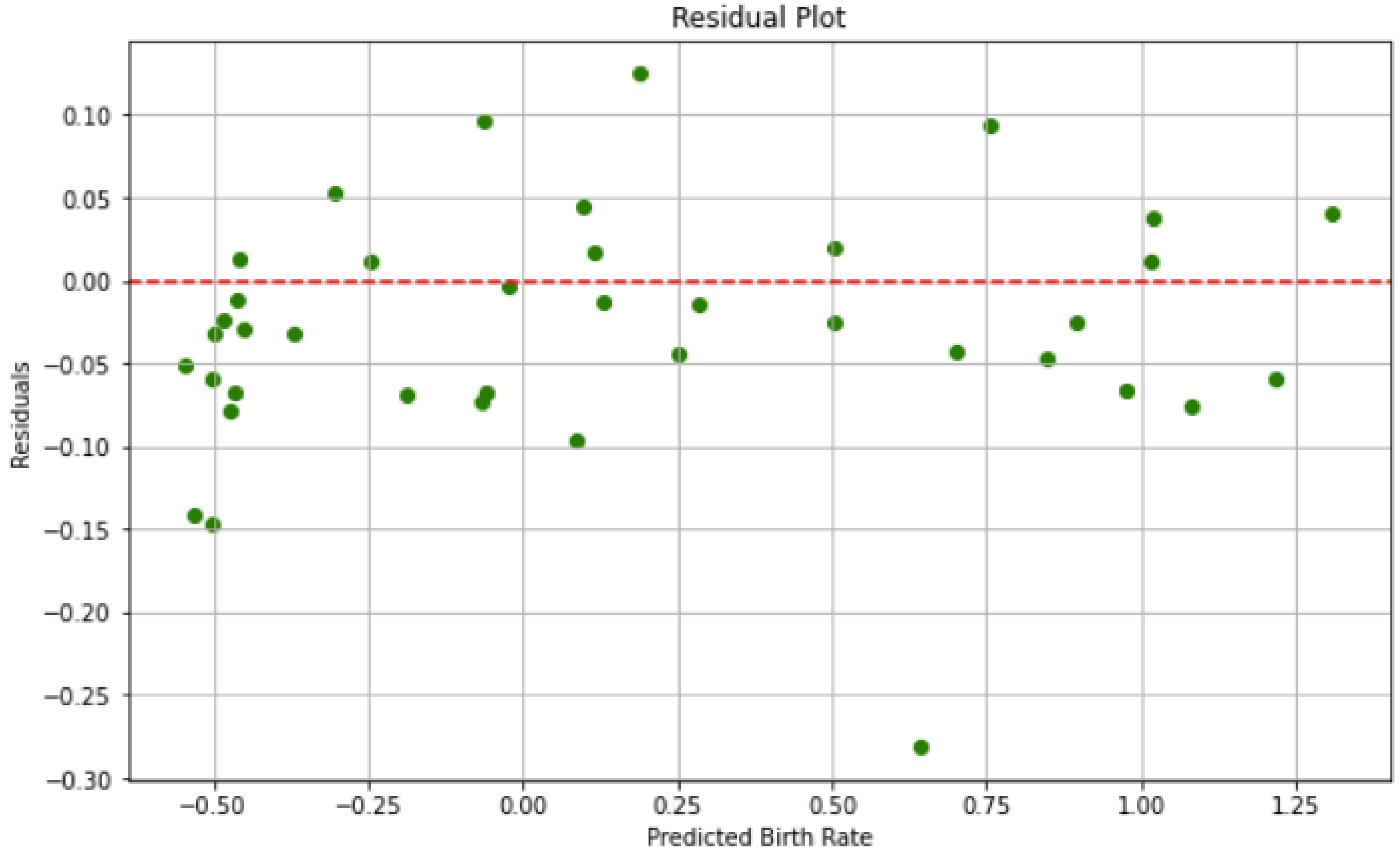
```
# splitting the data into features (X) and target (y)
X = df.drop('Birth Rate', axis=1)
y = df['Birth Rate']

# spliting the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

```
#fitting random forest
rf = RandomForestRegressor()
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)
```

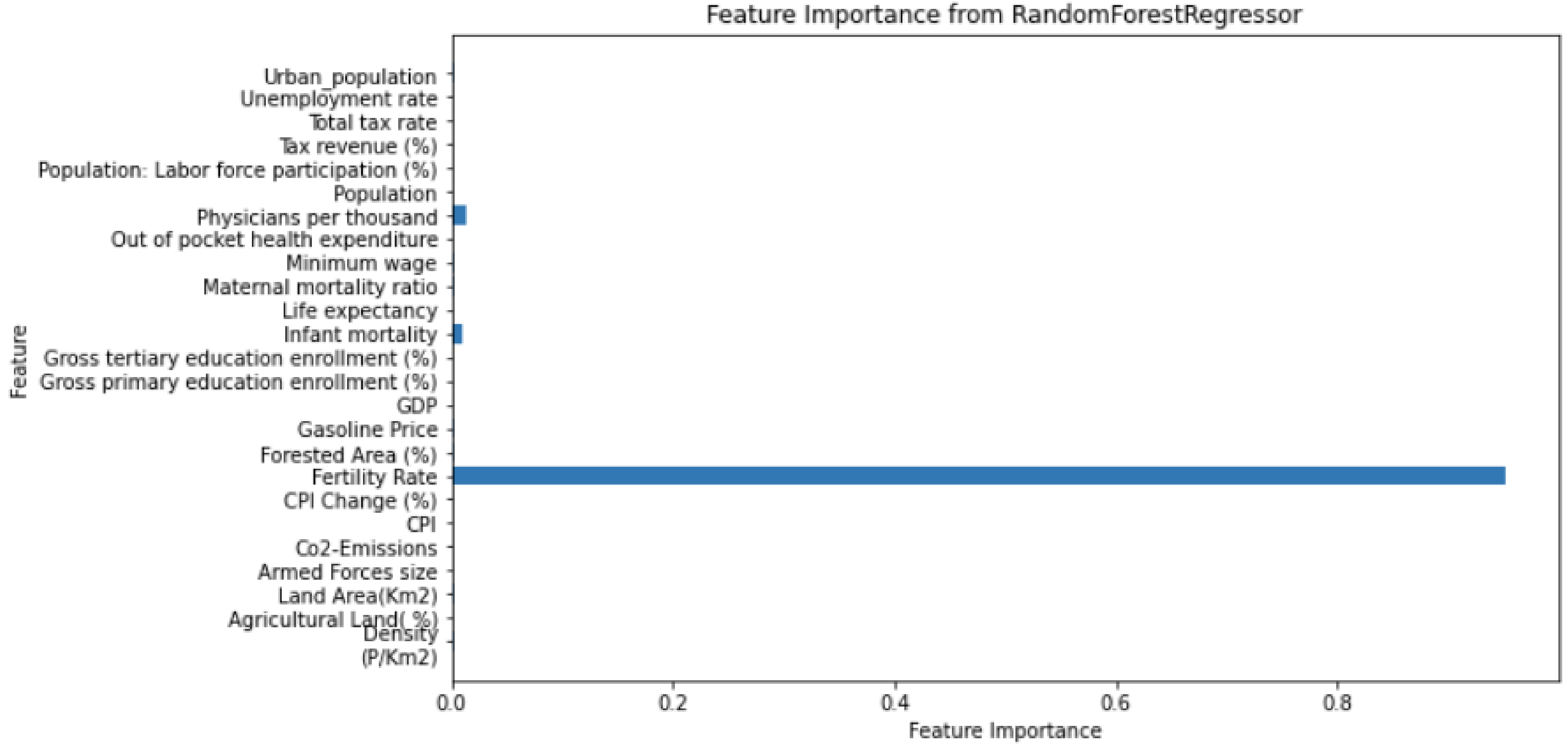
Random forest is a tree-based model. RF provides a measure of feature importance, allowing us to identify the most relevant features for prediction. This helps in feature selection and understanding the underlying data relationships. It also has high accuracy and is robust to overfitting.

```
Mean Absolute Error: 0.057474303234821084
Mean Squared Error: 0.0058589395118180225
Root Mean Squared Error: 0.07654370981222443
R-squared: 0.9833738156080455
```






```
Selected Features: Index(['Density\n(P/Km2)', 'Land Area(Km2)', 'Fertility Rate',
       'Infant mortality', 'Life expectancy', 'Minimum wage',
       'Physicians per thousand', 'Urban_population'],
      dtype='object')
```



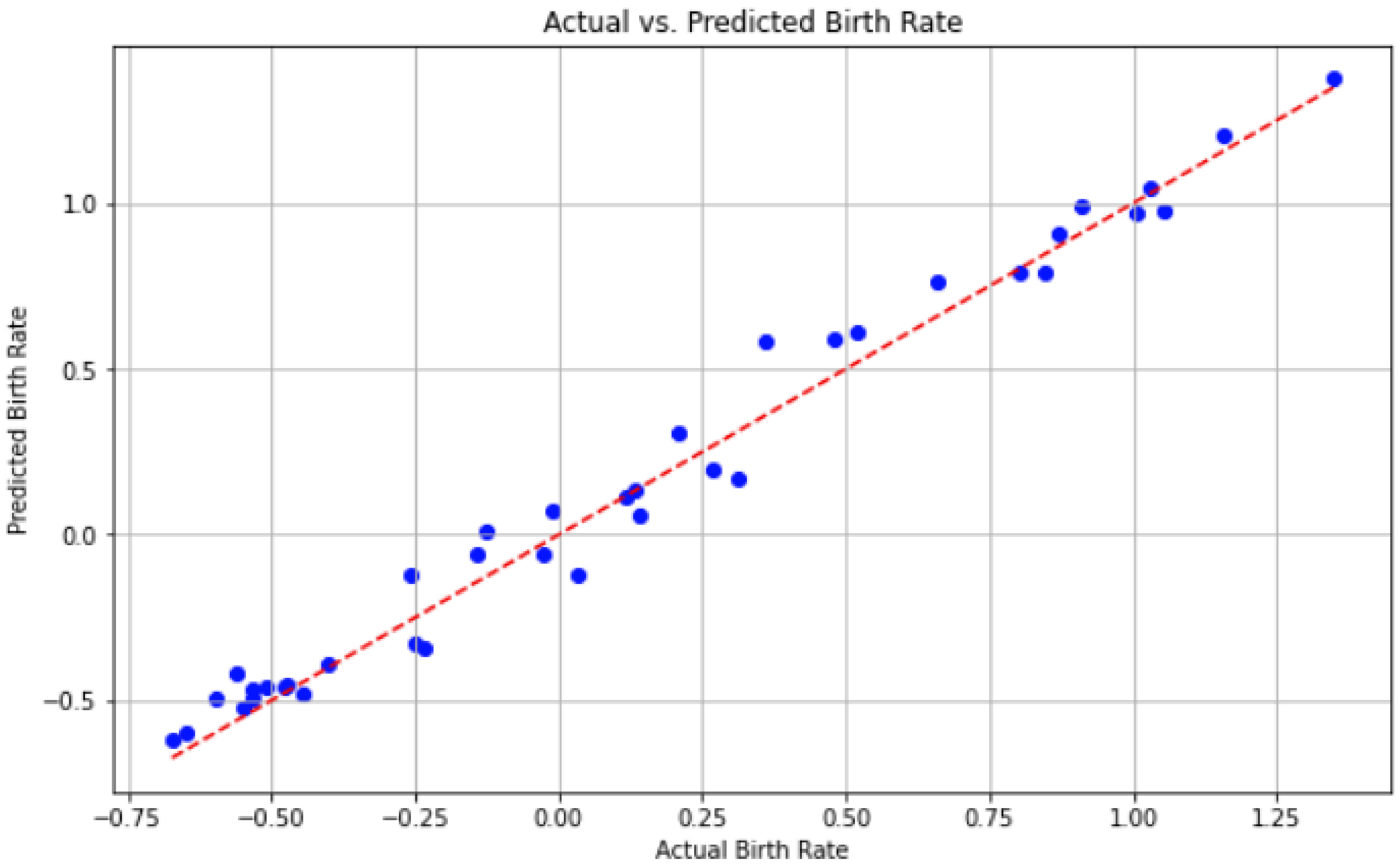
XGBOOST

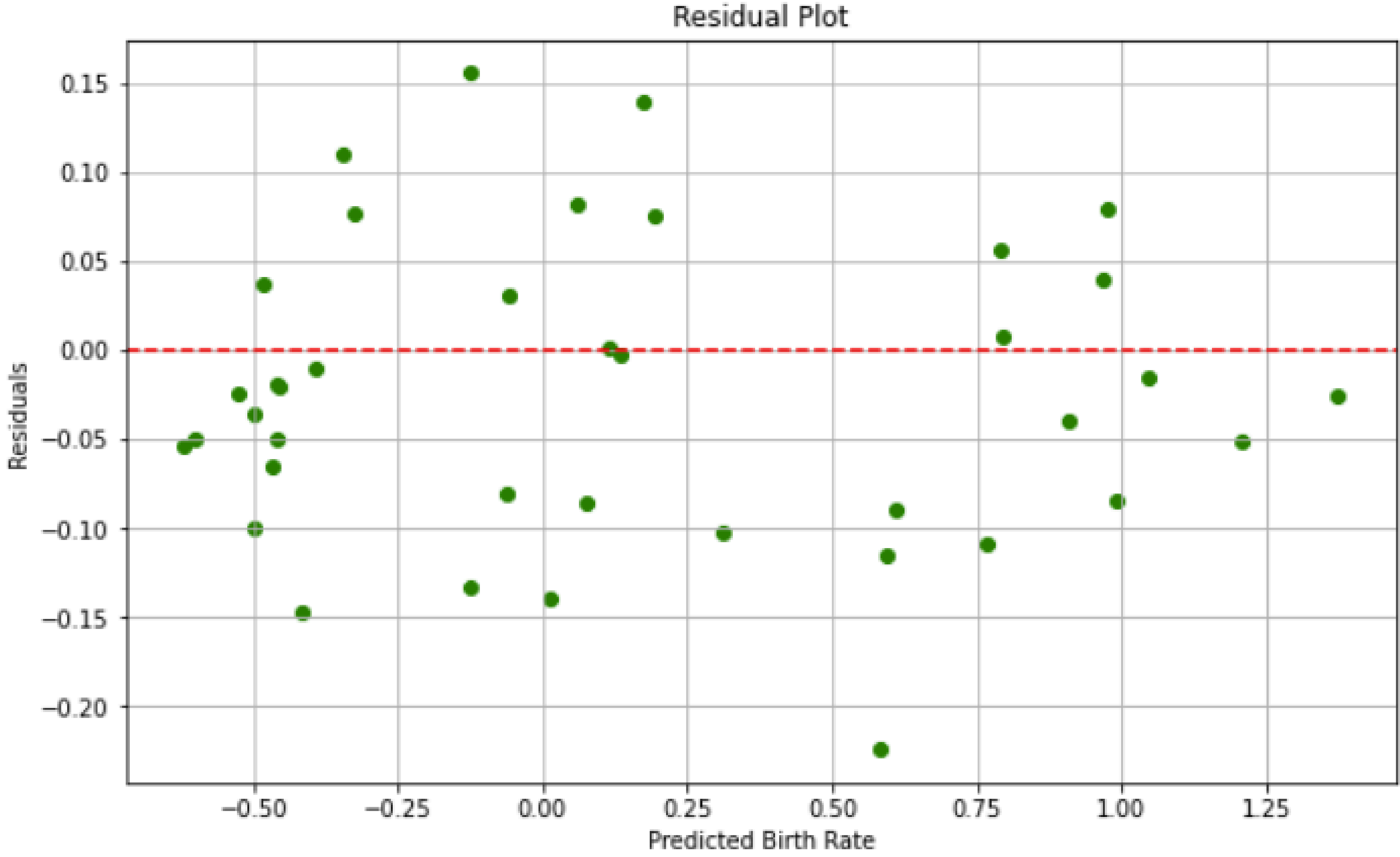
```
xg_reg = xgb.XGBRegressor()
xg_reg.fit(X_train, y_train)
```

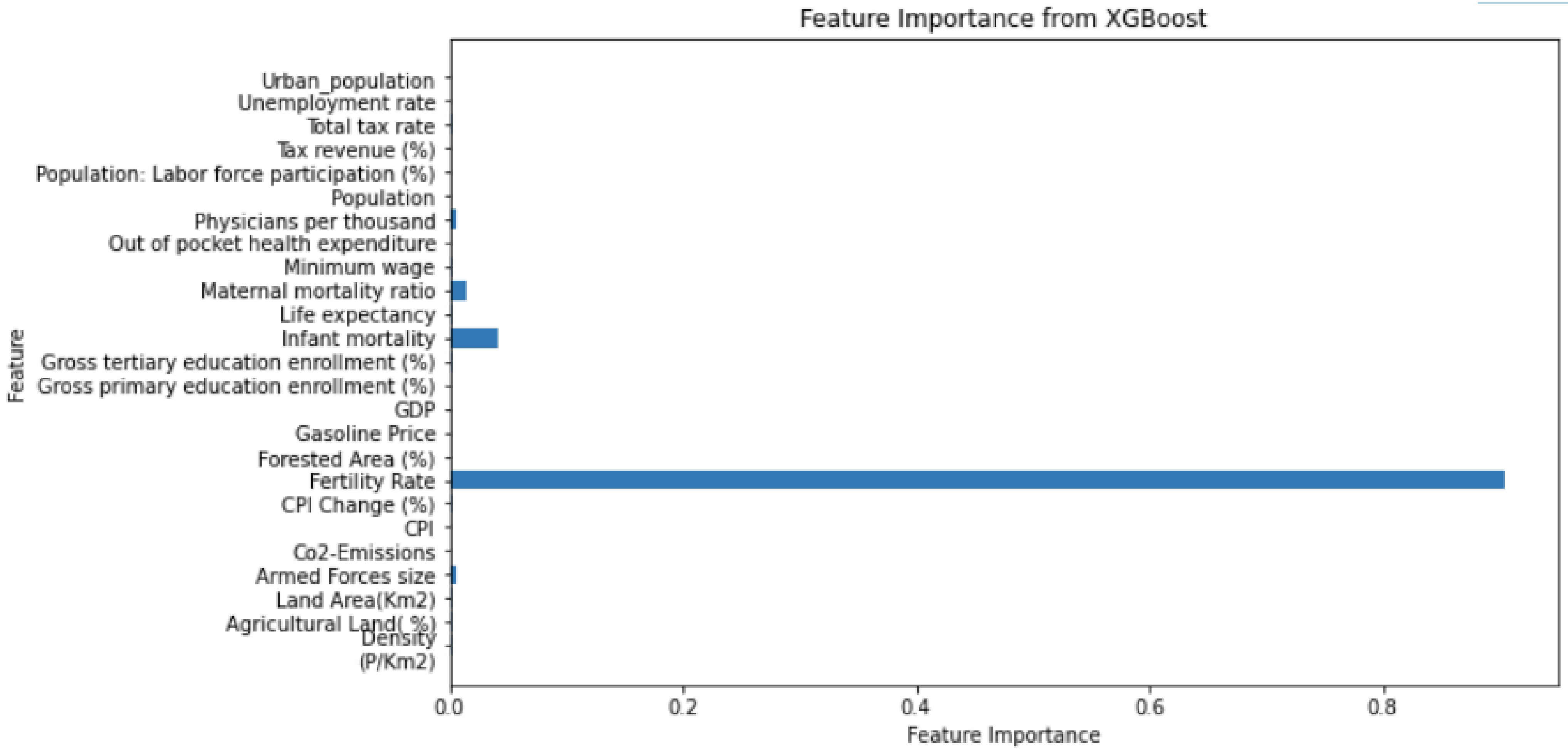
```
y_pred_xg = xg_reg.predict(X_test)
mae = mean_absolute_error(y_test, y_pred_xg)
mse = mean_squared_error(y_test, y_pred_xg)
rmse = mean_squared_error(y_test, y_pred_xg, squared=False)
r2 = r2_score(y_test, y_pred_xg)
```

```
Mean Absolute Error (MAE): 0.07084819173080982
Mean Squared Error (MSE): 0.007448505443319899
Root Mean Squared Error (RMSE): 0.08630472433951632
R-squared (R2): 0.9788630306397059
```

Powerful gradient boosting algorithm. Great for feature understanding, has high accuracy, is very resistant to overfitting, and has good scalability for large datasets.








```
Top Features: Index(['Fertility Rate', 'Infant mortality', 'Maternal mortality ratio',
       'Physicians per thousand', 'Armed Forces size', 'Density\n(P/Km2)',
       'Land Area(Km2)', 'Life expectancy', 'Minimum wage',
       'Gross tertiary education enrollment (%)'],
      dtype='object')
```

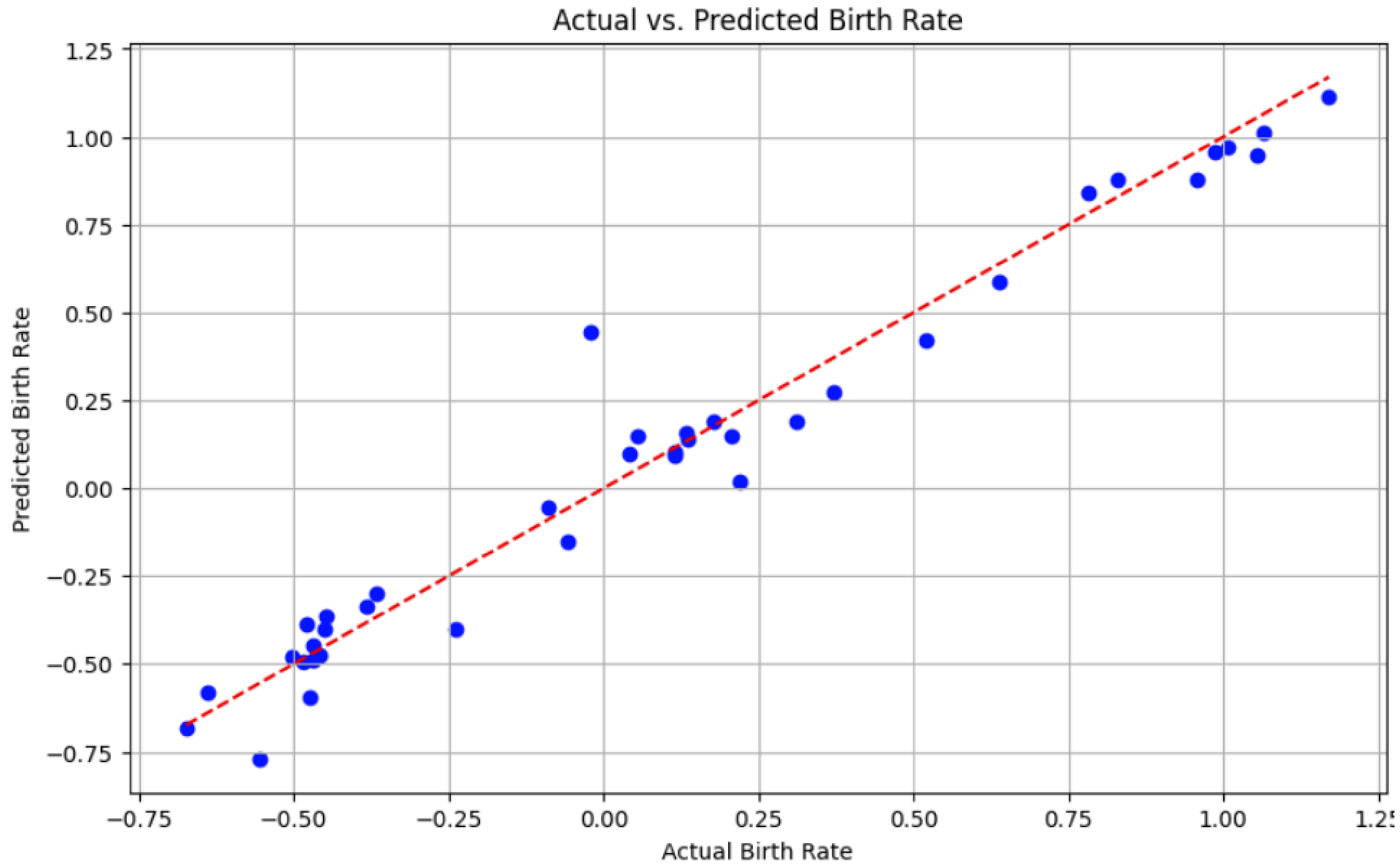
REGRESSION

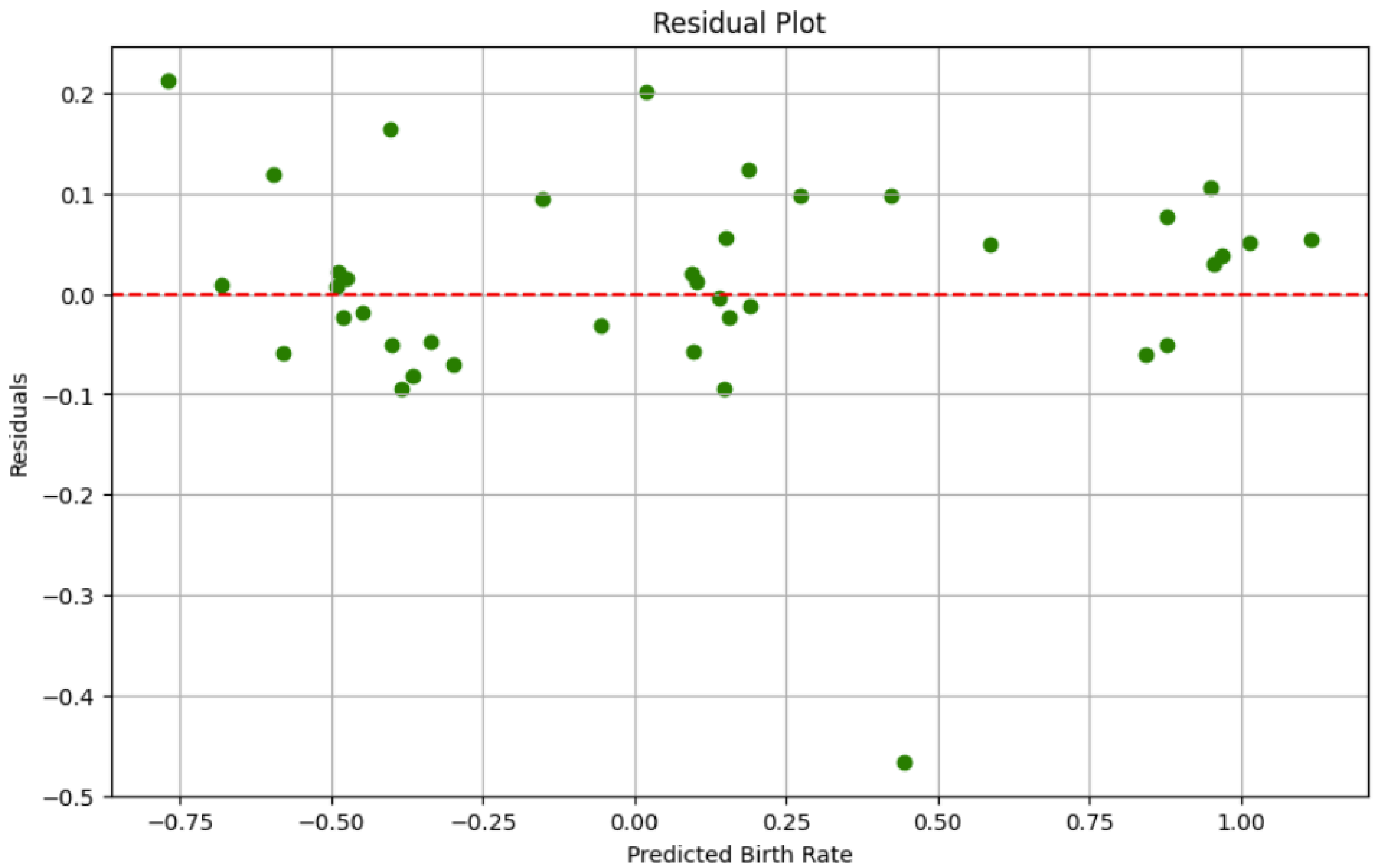
```
linear_reg = LinearRegression()

linear_reg.fit(X_train, y_train)
y_pred_linear = linear_reg.predict(X_test)
```

```
mae_linear = mean_absolute_error(y_test, y_pred_linear)
mse_linear = mean_squared_error(y_test, y_pred_linear)
rmse_linear = mean_squared_error(y_test, y_pred_linear, squared=False)
r2_linear = r2_score(y_test, y_pred_linear)
```

```
Linear Regression Model Evaluation:
Mean Absolute Error (MAE): 0.074485718695179
Mean Squared Error (MSE): 0.012038504236126752
Root Mean Squared Error (RMSE): 0.10972011773656987
R-squared (R2): 0.961213532821302б
```

Assumes a linear relationship between features and the target variable, and thus is also susceptible to outliers. Provided coefficients to better understand our chosen feature importance.





coefficients

Density (P/Km2): -0.003760283387886344
Agricultural Land(%): 0.0301780943028231
Land Area(Km2): 0.004685712620059385
Armed Forces size: -0.009158874768493764
Co2-Emissions: 0.0009864345109885345
CPI: -0.006703894922076731
CPI Change (%): 0.014208429233953927
Fertility Rate: 0.7216974390118495
Forested Area (%): -0.009471174880162225
Gasoline Price: -0.020512936070434026
GDP: -0.0012757942059010313
Gross primary education enrollment (%): 0.015488216553666933
Gross tertiary education enrollment (%): -0.027141456373032297
Infant mortality: 0.04007056933184624
Life expectancy: -0.0075499378774995796
Maternal mortality ratio: -0.026301104512538833
Minimum wage: -0.0019793065605077363
Out of pocket health expenditure: 0.031515522592488555
Physicians per thousand: -0.09392897892845468
Population: 0.004910768806841059
Population: Labor force participation (%): 0.02408799421779655
Tax revenue (%): 0.0076471450885876625
Total tax rate: 0.001767081134413488
Unemployment rate: 0.015759673740195632
Urban_population: -0.0059013798642849865

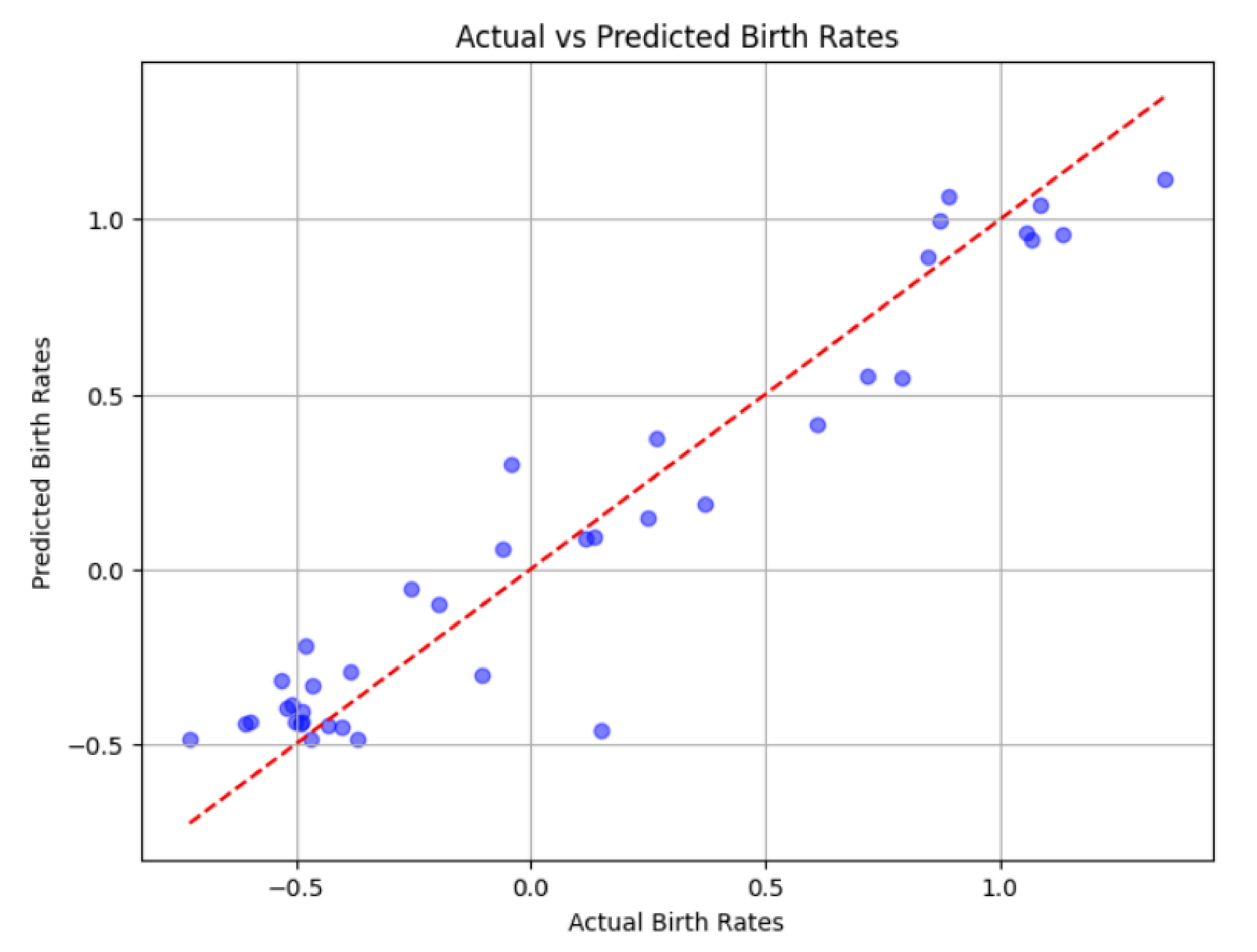
KNN

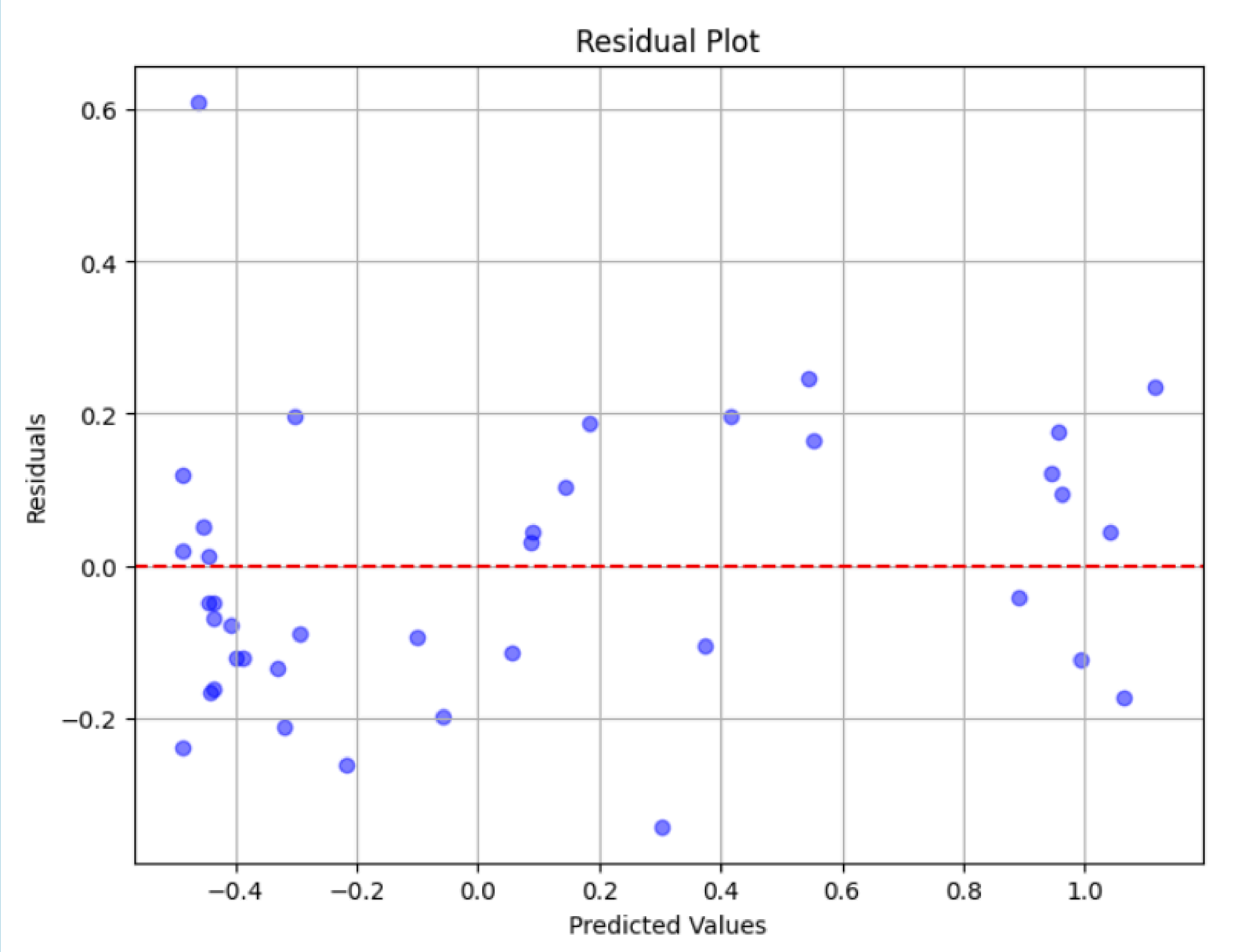
```
selected_features = ['Fertility Rate', 'Infant mortality', 'Maternal mortality ratio',
                     'Life expectancy', 'Gross tertiary education enrollment (%)',
                     'Physicians per thousand', 'Minimum wage']

X = df[selected_features]
y = df['Birth Rate']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
#scaled
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
num_instances = X_train_scaled.shape[0]
sqrt_num_instances = np.sqrt(num_instances)
k = int(np.round(sqrt_num_instances))

knn = KNeighborsRegressor(n_neighbors=k)
knn.fit(X_train_scaled, y_train)

y_pred = knn.predict(X_test_scaled)

mse = mean_squared_error(y_test, y_pred)
rmse = mse ** 0.5
print("Root Mean Squared Error:", rmse)
```

```
Root Mean Squared Error: 0.17905361368249223
```

```
Mean Absolute Error: 0.14373193229719655
Mean Squared Error: 0.03206019657275917
Root Mean Squared Error: 0.17905361368249223
R-squared: 0.9162737322732969
```

makes predictions based on the similarity of data points, not a linear model. Finding K value is most important, which we accomplished by finding the square root of the number of instances in the dataset.





KNN doesn't inherently give feature importance like XGBoost and other models.

MODEL EVALUATION

```
Mean Absolute Error: 0.057474303234821084
Mean Squared Error: 0.0058589395118180225
Root Mean Squared Error: 0.07654370981222443
R-squared: 0.9833738156080455
```

```
Mean Absolute Error (MAE): 0.07084819173080982
Mean Squared Error (MSE): 0.007448505443319899
Root Mean Squared Error (RMSE): 0.08630472433951632
R-squared (R2): 0.9788630306397059
```

```
Linear Regression Model Evaluation:
Mean Absolute Error (MAE): 0.074485718695179
Mean Squared Error (MSE): 0.012038504236126752
Root Mean Squared Error (RMSE): 0.10972011773656987
R-squared (R2): 0.9612135328213026
```

```
Mean Absolute Error: 0.14373193229719655
Mean Squared Error: 0.03206019657275917
Root Mean Squared Error: 0.17905361368249223
R-squared: 0.9162737322732969
```

RF: 98.3%

XGB: 97.9%

LR: 96.1%

KNN: 91.6%