**ARIMA: Code Description**

This Python code analyzes historical Bitcoin Greenhouse Gas (GHG) emission data and performs predictions for future emissions using the ARIMA (AutoRegressive Integrated Moving Average) model. It uses the following libraries:

* pandas: For data manipulation and analysis.
* numpy: For numerical computations.
* matplotlib.pyplot: For creating plots and visualizations.
* seaborn: For creating more sophisticated visualizations.
* statsmodels.api: For implementing the ARIMA model.
* sklearn.metrics.mean\_squared\_error: For evaluating the model performance using mean squared error.

**The code performs the following tasks:**

1. Data Loading and Preprocessing:
   * Loads the historical Bitcoin GHG emission data from a CSV file.
   * Prints the shape and column datatypes of the loaded data.
   * Renames the columns and modifies the date format.
   * Splits the data into training and test sets.
   * Converts the 'Date' column to datetime type and sets it as the index.
2. Model Training and Prediction:
   * Fits an ARIMA model to the training data with the specified order.
   * Makes predictions for the next one year using the fitted model.
   * Plots the historical data, test set, and predictions.
3. Model Evaluation:
   * Calculates the root mean squared error (RMSE) on the test set.

**Instructions**

To run this code, follow these steps:

1. Install the required libraries: pandas, numpy, matplotlib, seaborn, and statsmodels.
2. Prepare the data: Provide the path to the GHG Emission.csv file in the pd.read\_csv() function.
3. Run the code in a Python environment that supports the required libraries.
4. Examine the output plots and printed information to analyze the GHG emissions data.
5. Review the model performance by checking the RMSE score on the test set.

Feel free to modify the code and experiment with different ARIMA model orders or data sources to gain further insights into Bitcoin GHG emissions.