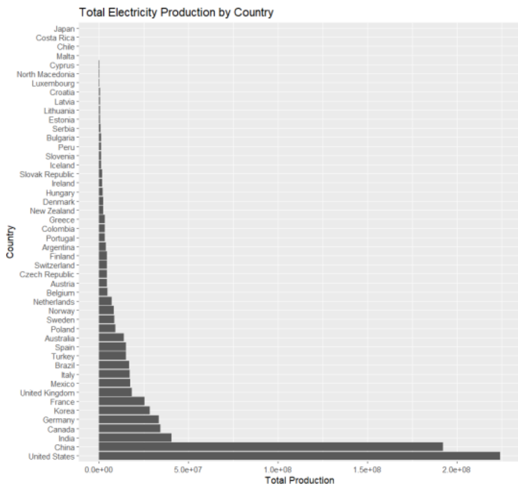
Towards a Low-Carbon Economy: Investing in Sustainable Energy Solutions for the U.S. and China

Over the past century, electricity has been providing humans with a necessary means of power. Countries such as the United States and China have been producing large amounts of electricity to support their successful economies. However, there is room for improvement in both countries to provide better sustainable and renewable energy sources. The concerns of climate change, environmental degradation, and the need to decrease greenhouse gas emissions have become pressing issues. This is particularly relevant as the energy sector, a major contributor to carbon emissions, has the potential to adapt sustainable renewable energy solutions. Therefore, the question arises: do policymakers in the United States and China prioritize investing in sustainable renewable energy sources to address the overconsumption of electricity production in both countries?

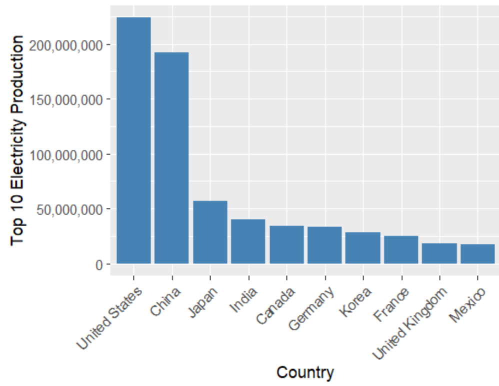
Currently, both the United States and China have a large mixture of greenhouse gas emissions, consisting of coal, oil, fossil fuels, natural gas, renewable energy, and nuclear power (Ritchie & Rosado, 2020). Energy production primarily involves burning fossil fuels, which is also the largest cause of climate change. Additionally, the use of fossil fuels and biomass significantly affects human health, potentially causing deadly effects due to pollution. This has led to approximately five million deaths due to air pollution every year (Ritchie & Rosado, 2020). Understanding that renewable energy sources should be an immediate response and solution to properly sustain not only electricity but also the health of many people is crucial. Although, with so much investment into electricity, it is also important analyze its production overtime for better process supply and demand as well.



Analyzing and comparing the electricity production between the U.S. and China is important because of their economic influence, energy consumption, and potential impact on climate change. The data shows that both countries play a major role in the global economy, affecting economic growth and industrial production. Countries with high electricity consumption often indicate infrastructure development. Climate change and environmental impact are also significant considerations.

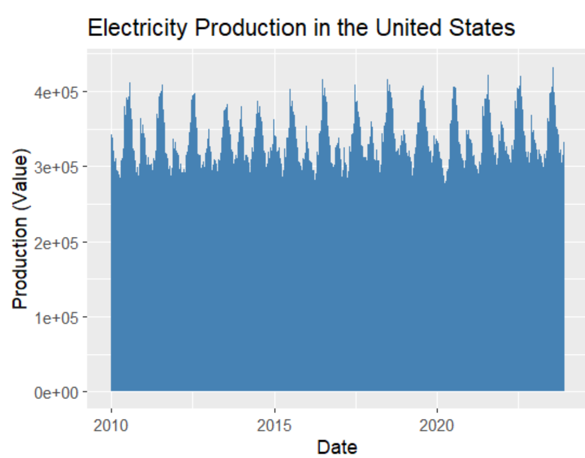
The objective of this report is to compare the trends, seasonal patterns, and volatility in electricity production between the two countries. It evaluates the performance of different forecasting models, such as ARMA, ARCH, and GARCH, while presenting a strong argument for prioritizing sustainable energy solutions in both countries.

The dataset reveals that China and the United States are the countries consuming the most electricity. Therefore, these two countries were chosen for a comparative analysis and forecast of their extreme electricity consumption. During the preprocessing and cleaning process, the dates needed to be converted to the "Date" format. Since the dataset contained information about many countries and their electricity consumption, focusing on the top two consumers helped minimize and narrow the analysis of their production forecast.



For this analysis, the ARMA, ARCH, and GARCH models were used for forecasting.

United States



Beginning with the United States' electricity production, the graph shows a clear seasonality from 2010 to 2020, with peaks at regular intervals. This pattern is likely due to high and low demand for electricity, influenced by factors such as weather conditions, including high demand for cooling in the summer and heating in the winter. This fluctuation pattern demonstrates the importance for utility companies to understand how to properly manage energy supply and demand. It also highlights the need to effectively plan during periods of high demand and implement strategies to meet those expected energy requirements. These strategies could include bringing additional power plants online or encouraging energy conservation measures during high-demand periods.

During the forecast period using the ARMA model, the Mean Error (ME) for forecasting electricity production in the U.S. is 609.9034. This represents the average distance between the forecasted values and the actual values. Although the ME is relatively small compared to the scale of the data, indicating that the bias is not too strong, the ARMA model tends to overpredict. Energy providers could use the ARMA model as an example for planning purposes, but they should be aware of the slight overestimation and adjust their plans accordingly.

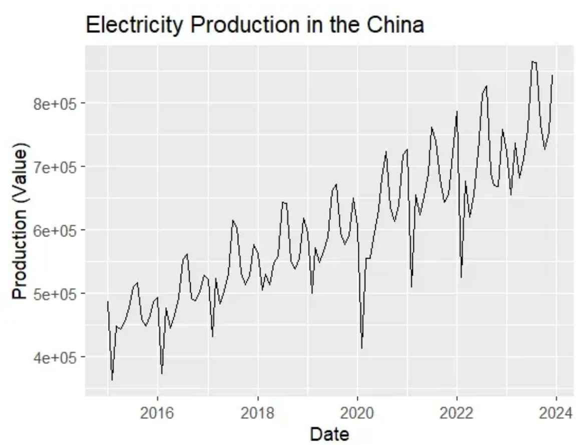
The Root Mean Squared Error (RMSE) is 109,510.1, which measures the average magnitude of the errors. A lower RMSE is always better, but this was not the case here. This suggests that the ARMA model may not capture all the nuances and volatility in the electricity production data. Relying solely on this forecast for decision-making related to electricity production could be problematic. Overall, the ARMA model performs moderately when it comes to forecasting electricity production in the U.S.

With the GARCH model, the Mean Error is -58,423.85, indicating that the model is underpredicting electricity production values, revealing a definite bias in the forecast. The RMSE of 127,658.6 also shows a repetitive average magnitude of errors, giving more weight to larger errors. The GARCH model forecast may have substantial errors, potentially leading to inaccurate energy supply planning or insufficient resource allocation, which could increase operational costs for electricity providers.

To enhance forecasting accuracy and reliability, potential recommendations and business insights for the U.S. could include:

- Exploring alternative forecasting models to capture the complexities of electricity production better.
- Incorporating additional relevant features, such as weather data, economic indicators, and population growth, to improve the models' predictive power.
- Conducting a thorough review of data quality to ensure the accuracy and consistency of the input data used for forecasting.

<u>China</u>



During the evaluation process, it was challenging to achieve a successful ARMA model forecast for China. The Mean Error of -54011.68 indicates that the forecast does not trust the overproduction of electricity in China. This could imply a potential supply shortage or capacity constraint if the forecast was used for planning energy consumption and resource allocation without accounting for this bias.

The RMSE of 191,688 is relatively high, although this might be due to the enormous amount of electricity consumption in China. A lower RMSE value would have been preferable, as it would show that the forecasted values are closer to the actual values. The high RMSE suggests that there could be significant deviations from the actual values.

The performance of the forecasting models varies between the two countries. For the U.S., the ARMA model appeared to perform relatively better than the ARCH and GARCH models. In contrast, for China, the ARMA model exhibited poor performance, while the ARCH model showed some potential in relation to volatility, and the GARCH model had certain convergence issues. These differing results highlight the importance of a country-specific forecasting approach and the potential benefits of using ensemble or hybrid forecasting techniques that combine multiple models.

Potential recommendations and business insights for China could include:

- Exploring alternative forecasting models that better suit the unique characteristics of China's electricity production data.
- Incorporating relevant forecasting features, such as economic growth, industrial production, and energy efficiency measures, to improve the models' predictive capabilities.
- Developing contingency plans to address potential supply shortages or capacity constraints identified by the forecasting models.

Overall, this was an interesting learning process of analyzing electricity consumption for both countries. Final realization of this is that analyzing electricity consumption is extremely complex because of the many challenging factor in brings especially because United Sates and China are two very developed and expanded countries. This analysis also revealed the importance of accurate forecasting to receive an effective forecast for effective planning and management. While these three models: ARMA, ARCH and GARCH provided certain insights, their performance depends between these two different countries.

Additionally, investing in sustainable renewable sources such as solar, wind, hydro renewable energy should be an essential solution and if not, then definitely alternative. The American and Chinese policymakers should maybe address some concerns of the overconsumption of electricity and reducing the impact it creates on the environment. To finish, by also approaching a practical approach of energy planning, but also of better forecasting advances method and making those renewable solutions as a priority.

References

Ritchie, H., & Rosado, P. (2020, July). Energy Mix. Our World in Data; Our World in Data. https://ourworldindata.org/energy-mix

Electricity Production Dataset. (n.d.). Www.kaggle.com. Retrieved June 3, 2024, from https://www.kaggle.com/datasets/sazidthe1/global-electricity-production

Appendix

R code

```
#Data
library(readr)
library(readr)   # For reading data
library(dplyr)   # For data manipulation
library(ggplot2) # For data visualization
library(scales)  # For formatting y-axis labels
library(lubridate)
library(forecast)
library(tseries)
library(zoo)
library(caret)

prod <- read_csv("Individual
Assignment/archive/global_electricity_production_data.csv")

head(prod)

                    ## EDA Analysis ##
glimpse(prod)  # To see the structure and data types
summary(prod)  # To get summary statistics

## Class and Mode are showing as character, therefore need to convert them.

##
```

```
prod$date <- as.Date(prod$date, format = "%m/%d/%Y")

unique(prod$date)

prod$country_name <- as.factor(prod$country_name)
prod$parameter <- as.factor(prod$parameter)
prod$product <- as.factor(prod$product)
prod$unit <- as.factor(prod$unit)

# Check for missing values
sum(is.na(prod))  # Count of missing values

# U.S Line plot of electricity production over time for US
ggplot(data = prod %>% filter(country_name == "United States", product == "Electricity"),
       aes(x = date, y = value, group = 1)) +
  geom_area(fill = "steelblue") +
  labs(title = "Electricity Production in the United States",
       x = "Date", y = "Production (Value)")

# Bar chart of electricity production by country
ggplot(data = prod %>% group_by(country_name) %>% summarise(total_prod = sum(value)),
```

```r
      aes(x = reorder(country_name, -total_prod), y = total_prod)) +
  geom_col() +
  labs(title = "Total Electricity Production by Country",
      x = "Country", y = "Total Production") +
  coord_flip()



# Assuming your data is in a data frame called 'electricity_prod'
top_10_countries <- prod %>%
  group_by(country_name) %>%
  summarise(total_prod = sum(value, na.rm = TRUE)) %>%
  arrange(desc(total_prod)) %>%
  slice(1:10) %>%
  pull(country_name)


filtered_data <- prod %>%
  filter(country_name %in% top_10_countries)


## GGPlot for top 10 ##
ggplot(filtered_data %>% group_by(country_name) %>% summarise(total_prod = sum(value, na.rm = TRUE)),
      aes(x = reorder(country_name, -total_prod), y = total_prod)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(x = "Country", y = "Top 10 Electricity Production") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_y_continuous(labels = comma)  # Add comma separators to y-axis labels
## Knowing now that United Stated has the most electricity production.
```

```
###  Forecast for United States ###

#Subset the data for the United States
us_data <- subset(prod, country_name == "United States")


#Prepare the data for time series analysis
us_data$date <- as.Date(us_data$date, format = "%m/%d/%Y")  # Replace with the appropriate format

#Create a time series object from the data
us_ts <- ts(us_data$value, start = c(date(min(us_data$date)), 1), frequency = 12)


## Time Series of 2023 ##
frequency(us_ts)

forecast_start <- c(14880, 1)  # January 2023
forecast_end <- c(14891, 12)   # December 2023


# Determine the split point
split_point <- c(14850, 12)  # December 2021 (assuming monthly data)

# Split the data into training and testing sets
train_data <- window(us_ts, end = split_point)
test_data <- window(us_ts, start = split_point + c(0, 1))
```

Train the models

arma_model <- auto.arima(train_data)

arch_model <- garch(train_data, order = c(1, 1))

garch_model <- garch(train_data, order = c(1, 1), type = "garch")

```
> arch_model <- garch(train_data, order = c(1, 1))

 ***** ESTIMATION WITH ANALYTICAL GRADIENT *****


     I     INITIAL X(I)        D(I)

     1     1.050622e+10     1.000e+00
     2     5.000000e-02     1.000e+00
     3     5.000000e-02     1.000e+00

    IT   NF      F         RELDF    PRELDF    RELDX   STPPAR   D*STEP   NPRELDF
     0    1  3.535e+04
     1    2  3.531e+04  1.14e-03  7.81e-02  4.7e-11  2.8e+03  1.0e+00  1.08e+02
     2    4  3.527e+04  1.10e-03  1.21e-03  2.3e-12  1.6e+01  5.0e-02  7.56e+00
     3    5  3.524e+04  9.43e-04  1.23e-03  4.5e-12  2.5e+00  1.0e-01  1.05e-01
     4    6  3.523e+04  3.42e-04  4.32e-04  4.3e-12  2.0e+00  1.0e-01  1.49e-01
     5    8  3.519e+04  9.79e-04  9.59e-04  9.5e-12  1.9e+00  2.0e-01  1.42e-01
     6    9  3.516e+04  8.48e-04  1.19e-03  1.8e-11  2.0e+00  4.0e-01  1.11e-01
     7   11  3.515e+04  2.41e-04  3.11e-04  1.9e-12  6.1e+00  4.0e-02  7.44e-03
     8   12  3.515e+04  4.97e-05  7.34e-05  1.5e-12  1.9e+00  4.0e-02  6.98e-04
     9   14  3.515e+04  1.74e-06  2.02e-06  1.9e-13  2.5e+00  4.0e-03  3.76e-06
    10   15  3.515e+04  7.47e-08  3.20e-07  1.8e-13  1.7e+00  4.0e-03  1.78e-06
    11   16  3.515e+04 -3.81e-08  8.15e-08  1.2e-13  0.0e+00  2.7e-03  8.15e-08

 ***** X-CONVERGENCE *****

 FUNCTION     3.515116e+04   RELDX        1.218e-13
 FUNC. EVALS      16         GRAD. EVALS      11
 PRELDF       8.155e-08      NPRELDF      8.155e-08

     I      FINAL X(I)        D(I)          G(I)

     1    1.050622e+10    1.000e+00     1.876e-08
     2    3.397416e-01    1.000e+00    -2.172e+00
     3    9.858274e-02    1.000e+00    -2.226e-01
```

```
> garch_model <- garch(train_data, order = c(1, 1), type = "garch")

 ***** ESTIMATION WITH ANALYTICAL GRADIENT *****


     I     INITIAL X(I)        D(I)

     1     1.050622e+10     1.000e+00
     2     5.000000e-02     1.000e+00
     3     5.000000e-02     1.000e+00

    IT   NF      F         RELDF    PRELDF    RELDX   STPPAR   D*STEP   NPRELDF
     0    1  3.535e+04
     1    2  3.531e+04  1.14e-03  7.81e-02  4.7e-11  2.8e+03  1.0e+00  1.08e+02
     2    4  3.527e+04  1.10e-03  1.21e-03  2.3e-12  1.6e+01  5.0e-02  7.56e+00
     3    5  3.524e+04  9.43e-04  1.23e-03  4.5e-12  2.5e+00  1.0e-01  1.05e-01
     4    6  3.523e+04  3.42e-04  4.32e-04  4.3e-12  2.0e+00  1.0e-01  1.49e-01
     5    8  3.519e+04  9.79e-04  9.59e-04  9.5e-12  1.9e+00  2.0e-01  1.42e-01
     6    9  3.516e+04  8.48e-04  1.19e-03  1.8e-11  2.0e+00  4.0e-01  1.11e-01
     7   11  3.515e+04  2.41e-04  3.11e-04  1.9e-12  6.1e+00  4.0e-02  7.44e-03
     8   12  3.515e+04  4.97e-05  7.34e-05  1.5e-12  1.9e+00  4.0e-02  6.98e-04
     9   14  3.515e+04  1.74e-06  2.02e-06  1.9e-13  2.5e+00  4.0e-03  3.76e-06
    10   15  3.515e+04  7.47e-08  3.20e-07  1.8e-13  1.7e+00  4.0e-03  1.78e-06
    11   16  3.515e+04 -3.81e-08  8.15e-08  1.2e-13  0.0e+00  2.7e-03  8.15e-08

 ***** X-CONVERGENCE *****

 FUNCTION     3.515116e+04   RELDX        1.218e-13
 FUNC. EVALS      16         GRAD. EVALS      11
 PRELDF       8.155e-08      NPRELDF      8.155e-08

     I      FINAL X(I)        D(I)          G(I)

     1    1.050622e+10    1.000e+00     1.876e-08
     2    3.397416e-01    1.000e+00    -2.172e+00
     3    9.858274e-02    1.000e+00    -2.226e-01
```

```
> # Make predictions on the testing set
> arma_forecast <- forecast(arma_model, h = length(test_data))
> arch_forecast <- predict(arch_model, n.ahead = length(test_data), trace = FALSE)
> garch_forecast <- predict(garch_model, n.ahead = length(test_data), trace = FALSE)
> arma_forecast <- arma_forecast$mean
> accuracy(arma_forecast, test_data)
                ME     RMSE      MAE       MPE     MAPE      ACF1 Theil's U
Test set 609.9034 109510.1 82900.36 -4586.443 4617.653 0.4442461  4.440175
>
```

```
> # Convert test_data to a time series object
> test_data_ts <- ts(test_data, frequency = 12, start = start(garch_forecast))
> # Align test_data with garch_forecast
> garch_time_index <- merge(yearmon(start(garch_forecast)), yearmon(end(garch_
> # Evaluate the model
> accuracy(garch_forecast, test_data_aligned)
                 ME     RMSE      MAE       MPE     MAPE      ACF1 Theil's U
Test set -58423.85 127658.6 115134.4 -8281.262 8300.674 0.4471945  8.847211
>
```

Make predictions on the testing set

```
arma_forecast <- forecast(arma_model, h = length(test_data))

arch_forecast <- predict(arch_model, n.ahead = length(test_data), trace = FALSE)

garch_forecast <- predict(garch_model, n.ahead = length(test_data), trace = FALSE)

# Extract the forecasted values
arma_forecast <- arma_forecast$mean
# arch_forecast and garch_forecast are already vectors of forecasted values


# Evaluate the models
accuracy(arma_forecast, test_data)

# Align arch_forecast and test_data
arch_forecast_aligned <- window(arch_forecast, start = c(14610, 1), end = c(14850, 12))
print(start(arch_forecast_aligned))
print(end(arch_forecast_aligned))

# Convert test_data to a time series object
test_data_ts <- ts(test_data, start = c(14610, 1), end = c(14850, 12), frequency = 12)

# Align test_data with arch_forecast
test_data_aligned <- window(test_data_ts, start = start(arch_forecast_aligned), end = end(arch_forecast_aligned))



## Garch Forecast
garch_forecast <- predict(garch_model, n.ahead = length(test_data), trace = FALSE)
```

```
# Print start and end indices of garch_forecast
print(start(garch_forecast))
print(end(garch_forecast))


# Convert test_data to a time series object
test_data_ts <- ts(test_data, frequency = 12, start = start(garch_forecast))

# Align test_data with garch_forecast
garch_time_index <- merge(yearmon(start(garch_forecast)),
yearmon(end(garch_forecast)))
# Evaluate the model
accuracy(garch_forecast, test_data_aligned)
```

Forecast for China

```
ggplot(data = prod %>% filter(country_name == "China", product == "Electricity"),
    aes(x = date, y = value, group = 1)) +
  geom_line() +
  labs(title = "Electricity Production in the China",
    x = "Date", y = "Production (Value)")

# Subset the data for China
china_data <- subset(prod, country_name == "China")

# Prepare the data for time series analysis
china_data$date <- as.Date(china_data$date, format = "%m/%d/%Y")  # Replace with
the appropriate format
```

```
# Create a time series object from the data
china_ts <- ts(china_data$value, start = c(year(min(china_data$date)), 1), frequency = 12)

# Determine the split point (assuming monthly data)
split_point <- c(year(max(china_data$date)), 12)  # December of the last year in the data

# Split the data into training and testing sets
train_data <- window(china_ts, end = split_point)
test_data <- window(china_ts, start = split_point + c(0, 1))

# Train the models
arma_model <- auto.arima(train_data)
arch_model <- garch(train_data, order = c(2, 0))
garch_model <- garch(train_data, order = c(2, 1), type = "garch")
```

```
> arch_model <- garch(train_data, order = c(2, 0))

 ***** ESTIMATION WITH ANALYTICAL GRADIENT *****


     I     INITIAL X(I)        D(I)

     1     5.396393e+10     1.000e+00
     2     5.000000e-02     1.000e+00
     3     5.000000e-02     1.000e+00

    IT   NF      F         RELDF    PRELDF    RELDX   STPPAR   D*STEP   NPRELDF
     0    1  1.389e+03
     1    3  1.386e+03  1.82e-03  2.43e-03  6.7e-13  3.4e+02  1.0e-01  4.10e-01
     2    4  1.386e+03  4.06e-04  5.46e-04  8.5e-13  1.8e+00  1.0e-01  5.57e-04
     3    5  1.386e+03  3.69e-05  4.32e-05  4.6e-13  0.0e+00  5.2e-02  4.32e-05

 ***** X-CONVERGENCE *****

 FUNCTION     1.385789e+03   RELDX        4.593e-13
 FUNC. EVALS       5         GRAD. EVALS       4
 PRELDF       4.315e-05      NPRELDF      4.315e-05

     I      FINAL X(I)        D(I)          G(I)

     1    5.396393e+10     1.000e+00    -1.288e-11
     2    1.618968e-01     1.000e+00    -4.852e-01
     3    1.784980e-01     1.000e+00    -5.184e-01
```

```
> garch_model <- garch(train_data, order = c(2, 1), type = "garch")

 ***** ESTIMATION WITH ANALYTICAL GRADIENT *****


     I     INITIAL X(I)        D(I)

     1     5.096593e+10     1.000e+00
     2     5.000000e-02     1.000e+00
     3     5.000000e-02     1.000e+00
     4     5.000000e-02     1.000e+00

    IT   NF      F         RELDF    PRELDF    RELDX   STPPAR   D*STEP   NPRELDF
     0    1  1.388e+03
     1    3  1.385e+03  2.35e-03  3.71e-03  7.7e-13  4.7e+02  1.1e-01  8.63e-01
     2    5  1.383e+03  9.53e-04  7.25e-04  8.7e-13  1.9e+00  1.1e-01  3.52e-02
     3    7  1.383e+03  2.36e-04  2.06e-04  1.7e-13  5.0e+00  2.1e-02  3.22e-01
     4    9  1.382e+03  4.96e-04  4.78e-04  3.2e-13  3.2e+02  4.2e-02  7.05e+00
     5   11  1.382e+03  1.05e-04  9.82e-05  5.8e-14  9.8e+00  8.4e-03  7.02e+02
     6   13  1.382e+03  2.12e-05  2.12e-05  1.2e-14  4.0e+02  1.7e-03  3.56e+03
     7   15  1.382e+03  4.27e-05  4.26e-05  2.3e-14  4.5e+02  3.4e-03  1.40e+05
     8   17  1.382e+03  8.58e-06  8.58e-06  4.7e-15  2.4e+04  6.7e-04  1.23e+07
     9   19  1.382e+03  1.72e-05  1.72e-05  9.3e-15  8.6e+03  1.3e-03  1.17e+09
    10   20  1.382e+03 -7.24e+06  3.44e-05  1.9e-14  1.7e+03  2.7e-03  1.16e+11

 ***** FALSE CONVERGENCE *****

 FUNCTION     1.382126e+03   RELDX        1.864e-14
 FUNC. EVALS      20         GRAD. EVALS      10
 PRELDF       3.444e-05      NPRELDF      1.159e+11

     I      FINAL X(I)        D(I)          G(I)

     1    5.096593e+10    1.000e+00     7.086e-11
     2    2.662597e-01    1.000e+00    -1.246e+01
     3    1.444044e-01    1.000e+00    -1.976e+00
     4    4.637375e-04    1.000e+00     1.238e+01

> # Evaluate the models
> accuracy(arma_forecast, test_data)
                ME   RMSE      MAE  MPE MAPE      ACF1 Theil's U
Test set -54011.68 191688 170713.2 -Inf  Inf 0.1643245         0
> 
```

Forecast for 2023

forecast_start <- c(year(max(china_data$date)) + 1, 1) # January of the following year

forecast_end <- c(year(max(china_data$date)) + 1, 12)   # December of the following year

Make predictions for 2023

arma_forecast <- forecast(arma_model, h = 12)

arch_forecast <- predict(arch_model, n.ahead = 12, trace = FALSE)

garch_forecast <- predict(garch_model, n.ahead = 12, trace = FALSE)

Extract the forecasted values

arma_forecast <- arma_forecast$mean

Evaluate the models (if you have actual values for 2023)

accuracy(arma_forecast, actual_values_2023)

accuracy(arch_forecast, actual_values_2023)

accuracy(garch_forecast, actual_values_2023)

```
# If you don't have actual values for 2023, you can plot the forecasts
plot(arma_forecast, main = "ARMA Forecast for China 2023")
plot(arch_forecast, main = "ARCH Forecast for China 2023")
plot(garch_forecast, main = "GARCH Forecast for China 2023")

# Make predictions on the testing set
arma_forecast <- forecast(arma_model, h = length(test_data))
arch_forecast <- predict(arch_model, n.ahead = length(test_data), trace = FALSE)
garch_forecast <- predict(garch_model, n.ahead = length(test_data), trace = FALSE)

# Extract the forecasted values
arma_forecast <- arma_forecast$mean

# Evaluate the models
accuracy(arma_forecast, test_data)
```