Time Series Forecasting of Energy Consumption in Romania

Oguzhan Gurbuz

2025-07-05

#INTRODUCTION

#Reliable energy consumption forecasting plays a vital role in ensuring the efficiency, sustainability, and stability of national energy systems. For Romania, a country undergoing significant energy transition toward renewable sources, understanding consumption dynamics is crucial for optimizing production and supporting policy planning.

#This project uses an open dataset collected from Kaggle, which contains hourly electricity consumption and production data for Romania from 2019 to 2024. The dataset records total energy consumption and generation, along with detailed production by source, including nuclear, hydroelectric, coal, oil and gas, wind, solar, and biomass.

#Through time series analysis, the project aims to provide an empirically grounded understanding of Romania's energy demand trends and to develop an accurate forecasting model that can serve as a valuable tool for energy planners, policymakers, and researchers working on sustainable energy systems and the country's green transition.

==

STEP 1: LOAD AND INSPECT THE ENERGY DATASET

==

```
# Load necessary packages
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.1     ✔ stringr   1.5.2
## ✔ ggplot2   4.0.0     ✔ tibble    3.3.0
## ✔ lubridate 1.9.4     ✔ tidyr     1.3.1
## ✔ purrr     1.1.0
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(tseries)
library(readr)
library(ggplot2)
```

#The dataset is available at: https://www.kaggle.com/datasets/stefancomanita/hourly-electricity-consumption-and-production/data (https://www.kaggle.com/datasets/stefancomanita/hourly-electricity-consumption-and-production/data)

#It contains hourly records of electricity consumption and production in Romania (2019–2024).

```
#Let's load the dataset
energy_cons <- read.csv("energy_consumption_and_production_of_romania.csv")

#Checking if we loaded the dataset correctly
head(energy_cons)
```

```
##           DateTime Consumption Production Nuclear Wind Hydroelectric
## 1 01/01/2019 00:00        6352       6527    1395   79          1383
## 2 01/01/2019 01:00        6116       5701    1393   96          1112
## 3 01/01/2019 02:00        5873       5676    1393  142          1030
## 4 01/01/2019 03:00        5682       5603    1397  191           972
## 5 01/01/2019 04:00        5557       5454    1393  159           960
## 6 01/01/2019 05:00        5525       5385    1395   91           958
##   Oil.and.Gas Coal Solar Biomass
## 1        1896 1744     0      30
## 2        1429 1641     0      30
## 3        1465 1616     0      30
## 4        1455 1558     0      30
## 5        1454 1458     0      30
## 6        1455 1456     0      30
```

```
#Let's review our data
glimpse(energy_cons)
```

```
## Rows: 54,170
## Columns: 10
## $ DateTime      <chr> "01/01/2019 00:00", "01/01/2019 01:00", "01/01/2019 02:0…
## $ Consumption   <int> 6352, 6116, 5873, 5682, 5557, 5525, 5513, 5524, 5510, 56…
## $ Production    <int> 6527, 5701, 5676, 5603, 5454, 5385, 5349, 5547, 5471, 55…
## $ Nuclear       <int> 1395, 1393, 1393, 1397, 1393, 1395, 1392, 1392, 1391, 13…
## $ Wind          <int> 79, 96, 142, 191, 159, 91, 98, 93, 51, 15, -5, -17, -10,…
## $ Hydroelectric <int> 1383, 1112, 1030, 972, 960, 958, 938, 1187, 1325, 1398, …
## $ Oil.and.Gas   <int> 1896, 1429, 1465, 1455, 1454, 1455, 1451, 1446, 1357, 13…
## $ Coal          <int> 1744, 1641, 1616, 1558, 1458, 1456, 1440, 1394, 1303, 13…
## $ Solar         <int> 0, 0, 0, 0, 0, 0, 0, 0, 8, 61, 126, 182, 206, 202, 160, …
## $ Biomass       <int> 30, 30, 30, 30, 30, 30, 31, 34, 34, 34, 34, 34, 34, 34, …
```

```
#Checking if there's any null value
sum(is.na(energy_cons$Consumption)) #No null value in the "Consumption" data.
```

```
## [1] 0
```

==

STEP 2: DATA CLEANING AND PREPARATION

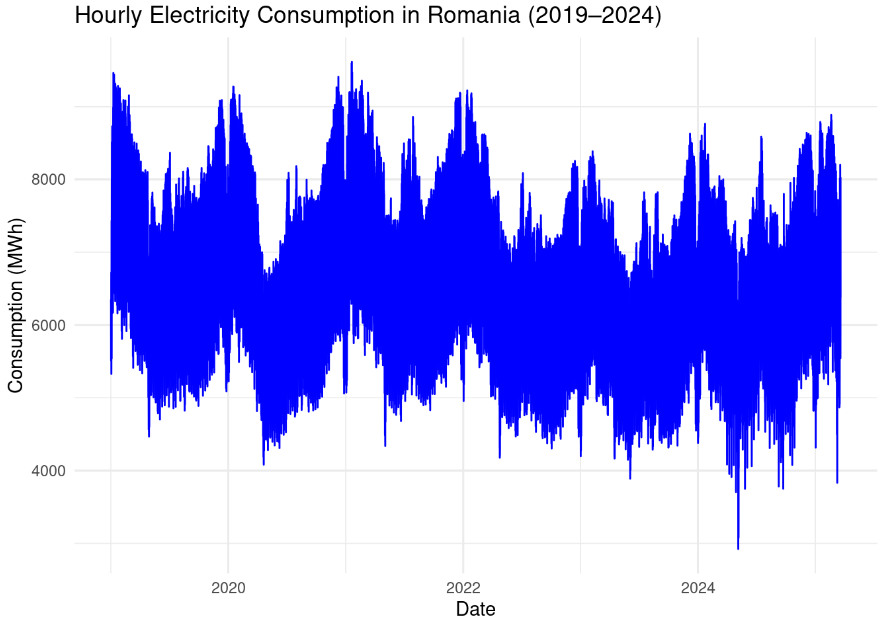
==

#We will standardize and organize the dataset for time series analysis.

```
energy_cons <- energy_cons %>%
  mutate(DateTime = dmy_hm(DateTime)) %>%  # convert to date-time format (dd/mm/yyyy hh:mm)
  arrange(DateTime)                        # ensure data is in chronological order
```

#Visualization of Hourly Energy Consumption in Romania

```
ggplot(energy_cons, aes(x = DateTime, y = Consumption)) +
  geom_line(color = "blue") +
  labs(
    title = "Hourly Electricity Consumption in Romania (2019–2024)",
    x = "Date",
    y = "Consumption (MWh)"
  ) +
  theme_minimal()
```



#The chart and consumption follows a clear seasonal pattern. It peaks in winter and drops in summer, and displays strong daily and weekly fluctuations typical of energy use.

#The overall trend is stable, with no major disruptions or missing periods, indicating a clean and reliable dataset for forecasting.

```
#We can transform hourly observations into daily averages to smooth out high-frequency fluctuations and make seasonal and long-term patterns clearer.

daily_cons_data <- energy_cons %>%
  mutate(Date = as.Date(DateTime)) %>%          #remove hours/minutes
  group_by(Date) %>%
  summarise(
    Consumption   = mean(Consumption, na.rm = TRUE),
    Production    = mean(Production, na.rm = TRUE),
    Wind          = mean(Wind, na.rm = TRUE),
    Solar         = mean(Solar, na.rm = TRUE),
    Hydroelectric = mean(Hydroelectric, na.rm = TRUE)
  )

#Let's inspect the new daily data set and ensure there is one observation per day.
glimpse(daily_cons_data)
```

```
## Rows: 2,268
## Columns: 6
## $ Date          <date> 2019-01-01, 2019-01-02, 2019-01-03, 2019-01-04, 2019-01…
## $ Consumption   <dbl> 5957.667, 6322.042, 7254.083, 7635.083, 7379.750, 7005.8…
## $ Production    <dbl> 5924.417, 6795.708, 7202.250, 7988.500, 6836.208, 7061.7…
## $ Wind          <dbl> 343.5833, 1106.7500, 989.2917, 1854.5833, 629.5000, 1391…
## $ Solar         <dbl> 43.29167, 53.33333, 33.33333, 45.50000, 92.75000, 23.625…
## $ Hydroelectric <dbl> 1285.875, 1252.708, 1477.208, 1411.833, 1212.042, 1072.1…
```

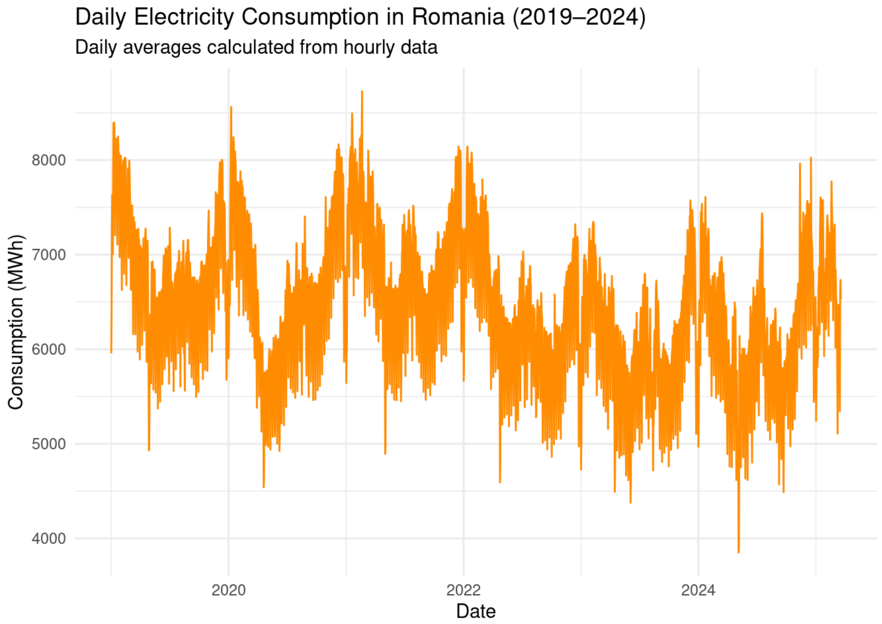
```
head(daily_cons_data)
```

```
## # A tibble: 6 × 6
##   Date       Consumption Production  Wind Solar Hydroelectric
##   <date>           <dbl>      <dbl> <dbl> <dbl>         <dbl>
## 1 2019-01-01       5958.      5924.  344.  43.3         1286.
## 2 2019-01-02       6322.      6796. 1107.  53.3         1253.
## 3 2019-01-03       7254.      7202.  989.  33.3         1477.
## 4 2019-01-04       7635.      7988. 1855.  45.5         1412.
## 5 2019-01-05       7380.      6836.  630.  92.8         1212.
## 6 2019-01-06       7006.      7062. 1391.  23.6         1072.
```

```
nrow(daily_cons_data)
```

```
## [1] 2268
```

```
#Visualize the daily electricity consumption

ggplot(daily_cons_data, aes(x = Date, y = Consumption)) +
  geom_line(color = "darkorange") +
  labs(
    title = "Daily Electricity Consumption in Romania (2019–2024)",
    subtitle = "Daily averages calculated from hourly data",
    x = "Date",
    y = "Consumption (MWh)"
  ) +
  theme_minimal()
```



#The daıly plot shows Romania’s daily electricity consumption (2019–2024) with clear seasonal cycles.

#Consumption peaks in winter and drops in summer, reflecting the seasonnal needs.

#The pattern looks consistent, which makes the data well-suited for forecasting.

==

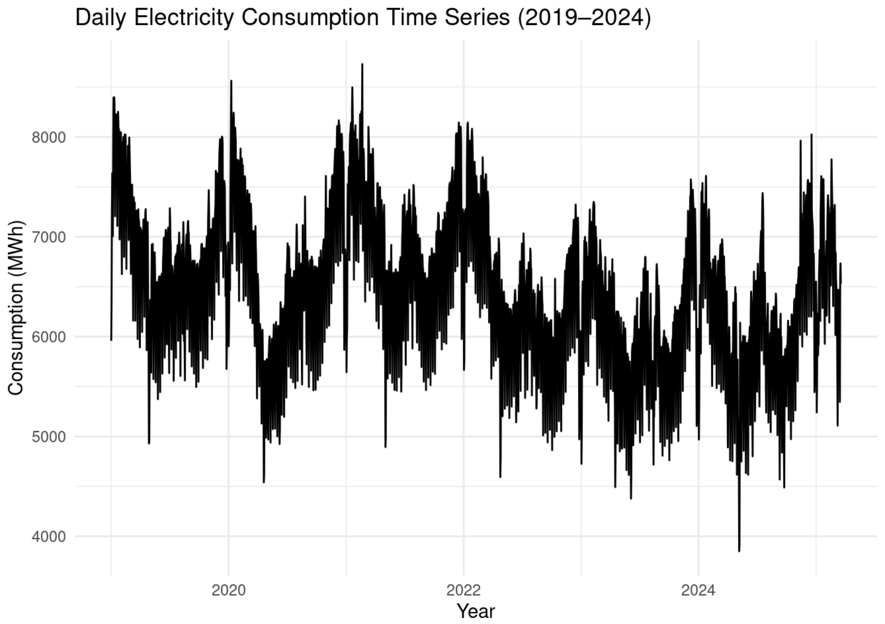
STEP 4: CREATING THE TIME SERIES OBJECT & FITTING AN ARIMA MODEL

==

```
#Let's convert daily data into a time series object
#We specify the start year (2019) and the frequency (365 for daily data).
cons_ts <- ts(daily_cons_data$Consumption, start = c(2019, 1), frequency = 365)
```

#Visualization of the time series

```
#This helps confirm the data structure before modeling.
autoplot(cons_ts) +
  labs(
    title = "Daily Electricity Consumption Time Series (2019–2024)",
    x = "Year",
    y = "Consumption (MWh)"
  ) +
  theme_minimal()
```


Daily Electricity Consumption Time Series (2019–2024)

#Fit an ARIMA model

```
#The Autoregressive Integrated Moving Average (ARIMA) model is one of the most widely used statistical techniques for foreca
sting time series data.

#ARIMA automatically selects the best parameters to capture trend and autocorrelation in the data.

fit_arima <- auto.arima(cons_ts)

#Display model summary to see parameters (p, d, q) and performance metrics.
summary(fit_arima)
```

```
## Series: cons_ts
## ARIMA(5,1,2)
##
## Coefficients:
##          ar1      ar2      ar3      ar4      ar5      ma1     ma2
##       0.0918  -0.7605  -0.2851  -0.3411  -0.5810  -0.4479  0.5346
## s.e.  0.0271   0.0173   0.0245   0.0163   0.0212   0.0362  0.0170
##
## sigma^2 = 102798:  log likelihood = -16296.48
## AIC=32608.97   AICc=32609.03   BIC=32654.78
##
## Training set error measures:
##                      ME    RMSE     MAE        MPE     MAPE      MASE
## Training set -0.1976985 320.056 240.287 -0.2211403 3.792642 0.4228041
##                     ACF1
## Training set -0.05353628
```

#The ARIMA(5,1,2) model was automatically selected as the best fit for Romania’s daily electricity consumption series (2019–2024).

#This indicates that:

#AR(5): the model uses the previous five days of consumption to predict the next day, capturing medium-term dependencies.

#I(1): first-order differencing to remove trend; the data is differenced once to remove trend and make the series stationary.

#MA(2): two moving average terms; the model incorporates the forecast errors from the previous two days to correct short-term fluctuations.

#Model Performance

#RMSE (Root Mean Square Error): 320.056 MWh

#MAE (Mean Absolute Error): 240.29 MWh

#MAPE (Mean Absolute Percentage Error): 3.79%

#These values indicate that the model’s predictions deviate from actual daily consumption by only around 3.8% on average, which demonstrates strong predictive accuracy for time series forecasting.

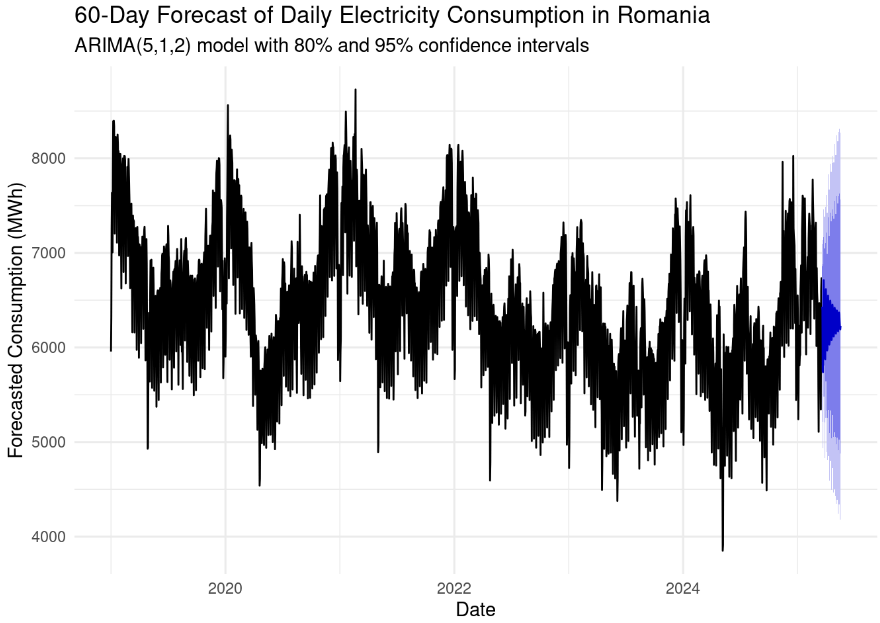
#The low mean error (ME ≈ 0) suggests the model is unbiased, with no systematic over / or / underestimation.

==

STEP 5: 60-DAY FORECAST USING ARIMA MODEL

==

```r
#Forecast the next 60 days
forecast_60_arima <- forecast(fit_arima, h = 60)

#Plot the 60-day forecast
autoplot(forecast_60_arima) +
  labs(
    title = "60-Day Forecast of Daily Electricity Consumption in Romania",
    subtitle = "ARIMA(5,1,2) model with 80% and 95% confidence intervals",
    x = "Date",
    y = "Forecasted Consumption (MWh)"
  ) +
  theme_minimal()
```



#Interpretation of the Forecast

#The ARIMA(5,1,2) model projects Romania’s daily electricity consumption for the next 60 days, with forecasts centered around 6,000–6,500 MWh per day.

#The model continues the existing seasonal pattern observed in previous years, showing stable demand and no, sudden changes.

#The widening blue confidence bands represent growing uncertainty over time but the forecast trend remains realistic and consistent with historical behavior.

==

STEP 6: 60-DAY FORECAST USING ARIMA MODEL

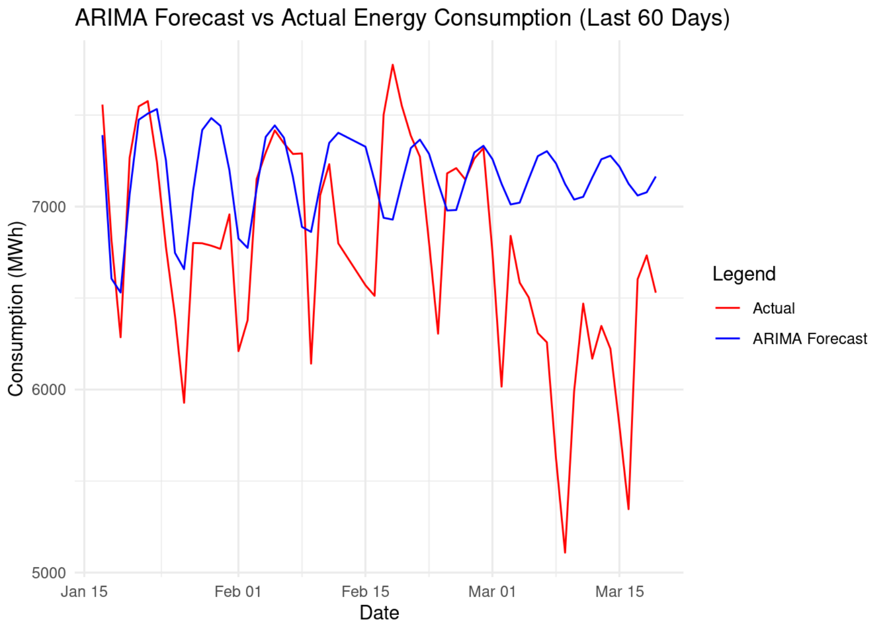
==

#To assess how well the ARIMA model predicts future values, we will split the dataset into a training set (all but the last 60 days) and a test set (the final 60 days). The model was trained on historical data and then used to forecast the next 60 days. We will compare its predictions

```r
#Split data into training and test sets (60 days) ----
# Use all but the last 60 days for training, and the final 60 days for testing.
train <- head(daily_cons_data$Consumption, -60)
test  <- tail(daily_cons_data$Consumption, 60)

#Convert to time series
train_ts <- ts(train, frequency = 365)

#Fit ARIMA model on the training data
fit_train_arima <- auto.arima(train_ts)

#Forecast the next 60 days (test period) ----
forecast_test_arima <- forecast(fit_train_arima, h = 60)

#Evaluate model accuracy
#Compare forecasted vs. actual test values
accuracy_results_arima_60 <- accuracy(forecast_test_arima$mean, test)
accuracy_results_arima_60
```

```
##                 ME     RMSE      MAE       MPE     MAPE
## Test set -402.9853 689.3279 524.1736 -6.726239 8.355127
```

```
#Visualize Actual vs Forecast
#Plot the last 60 days of real data against ARIMA predictions
forecast_df_60 <- data.frame(
  Date = tail(daily_cons_data$Date, 60),
  Actual = test,
  Forecast = as.numeric(forecast_test_arima$mean)
)

ggplot(forecast_df_60, aes(x = Date)) +
  geom_line(aes(y = Actual, color = "Actual")) +
  geom_line(aes(y = Forecast, color = "ARIMA Forecast")) +
  labs(
    title = "ARIMA Forecast vs Actual Energy Consumption (Last 60 Days)",
    x = "Date",
    y = "Consumption (MWh)",
    color = "Legend"
  ) +
  scale_color_manual(values = c("Actual" = "red", "ARIMA Forecast" = "blue")) +
  theme_minimal()
```



#The ARIMA(5,1,2) model achieved a MAPE of 8.36%, indicating that, on average, the forecasts deviate from actual daily consumption by about 8%. The RMSE of 689 MWh and MAE of 524 MWh suggest moderate forecast errors given the national scale of energy usage. While the model captures the overall trend, the plot shows it tends to smooth short-term fluctuations, underestimating sharp drops and peaks in daily demand. Overall, the ARIMA model provides a solid baseline forecast with reasonable accuracy for Romania’s electricity consumption.

==

STEP 7: STL MODEL (Seasonal-Trend Decomposition using Loess)

==

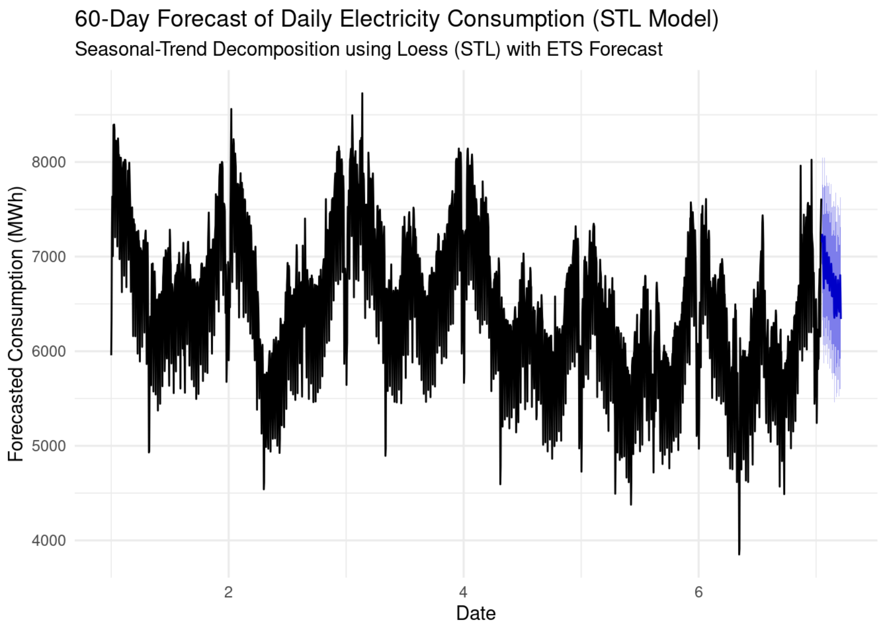
```
#Create the time series object
#Using daily consumption data with yearly frequency (365 observations per year)
cons_ts <- ts(daily_cons_data$Consumption, start = c(2019, 1), frequency = 365)

#Split data into training and test sets (60 days)
train <- head(daily_cons_data$Consumption, -60)
test  <- tail(daily_cons_data$Consumption, 60)
train_ts <- ts(train, frequency = 365)

#Fit the STL model and forecast
#stlf() function automatically decomposes the series into seasonal, trend, and remainder components and fits an ETS model on
the seasonally adjusted data.
fit_train_stl <- stlf(train_ts, h = 60)

#Evaluate STL model accuracy
accuracy_results_stl <- accuracy(fit_train_stl$mean, test)
accuracy_results_stl
```

```
##                 ME     RMSE     MAE        MPE     MAPE
## Test set 5.815144 594.3855 483.678 -0.6615085 7.345804
```

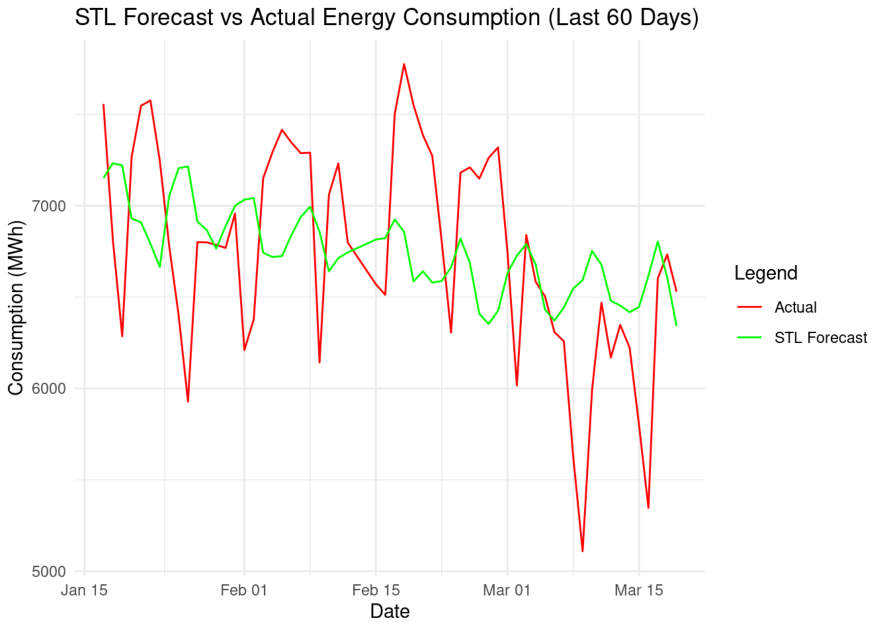
```
#Compare STL and ARIMA performance
accuracy_comparison <- rbind(
  ARIMA = accuracy_results_arima_60[1, c("RMSE", "MAE", "MAPE")],
  STL   = accuracy_results_stl[1, c("RMSE", "MAE", "MAPE")]
)
accuracy_comparison
```

```
##           RMSE      MAE     MAPE
## ARIMA 689.3279 524.1736 8.355127
## STL   594.3855 483.6780 7.345804
```

```
#Plot the STL forecast
autoplot(fit_train_stl) +
  labs(
    title = "60-Day Forecast of Daily Electricity Consumption (STL Model)",
    subtitle = "Seasonal-Trend Decomposition using Loess (STL) with ETS Forecast",
    x = "Date",
    y = "Forecasted Consumption (MWh)"
  ) +
  theme_minimal()
```



```
#Visual comparison: Actual vs STL forecast
forecast_df_stl <- data.frame(
  Date = tail(daily_cons_data$Date, 60),
  Actual = test,
  Forecast = as.numeric(fit_train_stl$mean)
)

ggplot(forecast_df_stl, aes(x = Date)) +
  geom_line(aes(y = Actual, color = "Actual")) +
  geom_line(aes(y = Forecast, color = "STL Forecast")) +
  labs(
    title = "STL Forecast vs Actual Energy Consumption (Last 60 Days)",
    x = "Date",
    y = "Consumption (MWh)",
    color = "Legend"
  ) +
  scale_color_manual(values = c("Actual" = "red", "STL Forecast" = "green")) +
  theme_minimal()
```


STL Forecast vs Actual Energy Consumption (Last 60 Days)

#The STL model was applied to the same dataset to explicitly capture the trend, seasonal, and irregular components of daily electricity consumption. Unlike ARIMA, STL decomposes the time series before forecasting, allowing it to adapt more effectively to recurring seasonal patterns.

#The 60-day forecast produced by the STL model closely follows the seasonal rhythm of past data, with smoother and more realistic confidence intervals. When compared to the actual consumption, the STL forecasts align well with observed variations, showing only slight underestimation of daily peaks.

#Model evaluation results confirm that STL performs better than ARIMA, achieving lower error metrics (RMSE = 594, MAE = 484, MAPE = 7.35% versus ARIMA's 8.36%). This indicates a more accurate and seasonally responsive model, making STL a stronger choice for short-term electricity consumption forecasting in Romania.

==

STEP 8: MODEL COMPARISON AND CONCLUSION

==

#Model Comparison

#Two models, namely ARIMA(5,1,2) and STL, were developed to forecast Romania's daily electricity consumption over a 60-day period.

#Both models achieved strong results, accurately capturing the overall trend and seasonal variations in energy demand. However, key performance metrics show that the STL model provided slightly higher predictive accuracy:

#ARIMA(5,1,2): RMSE = 689.33; MAE = 524.17; MAPE = 8.36% #STL : RMSE = 594.39; MAE: 483.68; MAPE = 7.35%

#The STL model achieved lower error values across all three metrics, indicating better fit and forecast precision. Its ability to handle strong seasonality and smooth long-term trends makes it particularly suitable for energy demand forecasting.

#In contrast, the ARIMA model, while still robust, tended to smooth short-term fluctuations and slightly underestimate rapid changes in consumption.

#== #Conclusion
#==

#Both ARIMA and STL models demonstrate that Romania's electricity consumption is highly seasonal but stable over time, with consistent winter peaks and summer troughs.

#The STL model proved to be the more reliable forecasting approach, reducing the average forecast error to about 7%.

#For short-term (1–3 month) planning, STL provides realistic and interpretable predictions that can support energy management, grid operations, and policy decisions.