Forecasting of Peak Electricity Demand in New South Wales

Team Members: Alexander Zhuk, Jordan Torres, Alli McKernan

Course: ADS506

Date: December 8th

Institution: University of San Diego

Link to GitHub Repository:

https://github.com/alligmckernan/ADS-506-Team-4-Final-Project/tree/main

Abstract

This project develops and evaluates forecasting models for daily maximum electricity demand in New South Wales (NSW), Australia, using data from January 2024 through October 2025. Accurate demand forecasting is essential for maintaining grid stability, optimizing generation costs, and preventing power outages. We compare ARIMA, Exponential Smoothing (ETS), and Time Series Linear Models (TSLM) to identify the most accurate approach for operational forecasting. Our analysis reveals that the ETS(M,A,M) model which features multiplicative error and seasonality with additive trend, delivers superior performance with an RMSE of approximately 686 MW and MAPE of 4.97%. This model effectively captures the weekly seasonality and upward trend inherent in NSW electricity demand. We recommend integrating this forecasting approach into operational planning systems and enhancing it with exogenous variables such as temperature and day-type indicators.

Introduction

Electricity demand forecasting plays a vital role in ensuring energy reliability, operational efficiency, and long-term planning in power systems. In regions like New South Wales (NSW), where the electricity grid supports millions of residential, commercial, and industrial users, understanding demand dynamics is crucial to preventing supply shortages and managing peak loads. Forecasting models help system operators anticipate high-demand periods which are often driven by extreme temperatures or economic activity and plan for appropriate generation and reserve capacity.

The Australian Energy Market Operator (AEMO) provides detailed historical demand data for the National Electricity Market (NEM), which includes NSW. This data serves as a foundation for developing time series models that can capture both predictable seasonal cycles and emerging structural changes in energy consumption due to population growth, electrification, and renewable integration. As AEMO (2025) notes, the growing influence of distributed solar generation and climate variability has made demand patterns increasingly volatile, underscoring the need for robust forecasting techniques. Forecasting failures can result in supply shortages, voltage instability, or unnecessarily high operating costs from maintaining excess reserve capacity.

This project develops a forecasting model for daily maximum electricity demand in New South Wales, Australia, using historical data from January 2024 through October 2025. The primary objective is to support energy planning and operational efficiency by accurately predicting peak daily demand. Accurate demand forecasting is essential for maintaining a reliable power supply in NSW's electricity network, which serves millions of residential, commercial, and industrial customers. The electricity provider must anticipate short-term demand to prevent outages and optimize generation costs. Forecast accuracy plays a critical role in effective resource allocation, cost management, and meeting sustainability goals. The dataset, sourced from the Australian Energy Market Operator (AEMO), contains historical maximum demand data with no missing values. Initial analysis revealed clear patterns in the data, including weekly seasonality with higher demand during midweek and lower demand on weekends and an upward long-term trend reflecting increasing energy usage over time. Daily maximum demand ranged from approximately 8,000 MW to 12,000 MW during the observation period. This project compares multiple time series forecasting approaches including ARIMA, Exponential Smoothing (ETS), and Time Series Linear Models (TSLM) to identify the most accurate method for operational forecasting. The recommended model will be integrated into operational planning to inform daily generation scheduling, energy market pricing, and resource readiness during high-demand periods.

Problem Statement

Current State

The NSW electricity provider operates a network serving millions of residential, commercial, and industrial customers across the state. Historical demand data from January 2024 through October 2025 shows daily maximum electricity demand ranging from approximately 7,623 MW to 13,764 MW, with an average of 9,750 MW for NSW. The data exhibits considerable fluctuations with an overall increasing trend and clear weekly cycles, where demand is higher during midweek and lower on weekends.

Analysis of the historical data revealed non-stationary characteristics, with a KPSS test p-value of 0.023 indicating the presence of a strong trend component. Decomposition showed a dominant trend (strength = 0.80) and moderate weekly seasonal strength (0.39). Additionally, high spikiness and residual autocorrelation (ACF1 = 0.24) indicated that substantial unexplained structure remains in the demand patterns, confirming the need for sophisticated forecasting models.

Business Objective

The primary objective is to forecast daily maximum electricity demand for New South Wales to support energy planning and operational efficiency. Specifically, the electricity provider must anticipate short-term demand to ensure a stable power supply, prevent outages, and optimize generation costs.

The forecasting model must:

- Accurately predict daily maximum demand for operational planning horizons
- Capture both the upward trend and weekly seasonality present in the data
- Provide reliable forecasts suitable for generation scheduling and resource allocation
- Support cost management and sustainability objectives

Success Criteria

Success will be measured through forecast accuracy metrics on holdout test data:

Quantitative Metrics:

- Root Mean Square Error (RMSE) indicating average forecast error magnitude

- Mean Absolute Percentage Error (MAPE) below 5% for operational acceptability
- Model residuals approximating white noise with minimal autocorrelation

Model Quality:

- Clean diagnostic plots confirming adequate model specification
- Stable performance across the test period (October 2025)
- Appropriate capture of trend, weekly cycles, and remaining structure

Operational Utility:

- Model outputs suitable for daily generation scheduling
- Forecasts supporting energy market pricing decisions
- Capability for resource readiness planning during high-demand seasons such as summer

Issues of Threats

Data Characteristics:

- Non-stationarity requiring differencing or explicit trend modeling
- Weekly seasonality that must be appropriately specified
- Residual structure indicating that noise is not purely random
- Limited to date and demand variables without weather or calendar information

Modeling Challenges:

- Selecting appropriate model specifications (ARIMA, ETS, or TSLM)
- Determining whether additive or multiplicative seasonal components are appropriate
- Deciding whether Box-Cox transformation improves forecast performance
- Balancing model complexity with interpretability for operational use

Forecast Requirements:

- Models must capture short-term dependencies while representing longer-term patterns
- Forecasts must be accurate enough for cost control and outage prevention
- Model must update efficiently as new demand data becomes available

Addressing these challenges requires systematic comparison of multiple forecasting approaches with rigorous evaluation on holdout data to identify the most accurate and operationally suitable model.

Literature review

Background

The Australian Energy Market Operator (AEMO) manages the National Electricity Market (NEM), coordinating electricity supply and demand across eastern and southern Australia, including New South Wales. AEMO publishes comprehensive aggregated price and demand datasets that provide transparent access to historical electricity consumption patterns across NEM regions. These datasets, available at 5-minute intervals, enable empirical research and operational analysis for energy market participants, researchers, and policymakers (AEMO, 2025).

For New South Wales specifically, the aggregated demand data captures total electricity consumption across the state's transmission network. This measurement reflects the combined load from all customer segments such as residential, commercial, and industrial providing system operators with visibility into statewide demand patterns. As AEMO (2025) notes, the growing influence of distributed solar generation and climate variability has made demand patterns increasingly volatile, underscoring the need for robust forecasting techniques.

Overview of Existing Research

Numerous studies have applied statistical and machine learning methods to electricity demand forecasting. Traditional linear regression and exponential smoothing methods have been effective for short-term forecasts with strong seasonality and stable trends. For instance, Hyndman and Athanasopoulos (2021) emphasize that Time Series Linear Models (TSLM) can effectively incorporate deterministic trends, seasonal effects, and exogenous variables, offering interpretability and ease of implementation. However, such models may underperform when structural shifts or nonlinear dynamics dominate the data.

ARIMA-based models have historically been the benchmark for univariate forecasting, with their strength lying in modeling autocorrelation and short-term dependencies. Studies such as Taylor (2010) found ARIMA models to be highly effective for short-term electricity load forecasting in regions with consistent daily and weekly cycles. However, ARIMA's purely statistical nature limits its ability to directly incorporate external regressors like temperature or holidays, which are key drivers of electricity demand in Australia.

More recent literature has explored hybrid and machine learning models—such as Prophet, random forests, and recurrent neural networks (RNNs)—that can adapt to complex nonlinear patterns (Kandananond, 2011; Hong et al., 2020). While these methods often yield high predictive accuracy, they require extensive computational resources and large, feature-rich datasets. For this project, a Time Series Linear Model (TSLM) offers a balance between interpretability, computational efficiency, and forecasting performance, particularly when the dataset exhibits clear deterministic trends and seasonality.

Gaps in Knowledge

While extensive research exists on load forecasting for large markets like the NEM, fewer studies focus specifically on daily peak demand forecasting for NSW using interpretable models. Many previous models prioritize short-term (hourly) forecasts or aggregate regional forecasts without addressing daily maxima as a separate, operationally significant target variable. Moreover, limited studies have examined how long-term climatic trends and evolving usage behaviors (e.g., increased heatwave frequency or electric vehicle adoption) alter the seasonal structure of NSW demand. This project aims to fill this gap by applying TSLM techniques to model and forecast daily peak demand, emphasizing model transparency and interpretability to support grid management decisions.

Practical Domain Reference

The AEMO Aggregated Price and Demand dataset (AEMO, 2025) provides high-quality, publicly available demand data that underpins much of the empirical work in Australian energy forecasting. It captures variations in demand due to economic activity, population growth, and weather patterns, making it a practical benchmark for evaluating forecasting models. AEMO's data infrastructure supports transparency in market operations and enables academic and policy research on forecasting accuracy, grid stability, and energy transition planning.

Relevant Time Series Methods

The TSLM framework, as detailed by Hyndman and Athanasopoulos (2021), is an extension of traditional regression modeling for time series data. It combines deterministic trend terms (e.g., linear or quadratic time) with seasonal components—either categorical indicators or Fourier terms—to capture recurring cycles. This method is particularly suited for electricity demand forecasting because it allows clear decomposition of trend and seasonality while maintaining interpretability for policy and operational planning. Fourier terms, in particular, are valuable for modeling smooth cyclic patterns (such as the 7-day week or annual seasonality) without overfitting.

Key Findings

- Electricity demand exhibits strong seasonal and intra-week patterns, typically peaking during hot summer days due to air conditioning use (Taylor, 2010).
- Linear models with Fourier or dummy seasonal terms often provide interpretable and competitive forecasts when demand patterns are stable (Hyndman & Athanasopoulos, 2021).
- Incorporating exogenous variables such as temperature, holidays, or economic indicators can significantly improve forecast accuracy (Hong et al., 2020).
- Despite advances in AI-based methods, simpler models remain valuable for operational forecasting due to their transparency, explainability, and ease of updating.

Explanation of Steps

Exploratory Data Analysis: Notable Findings

The exploratory data analysis revealed several important characteristics of the NSW electricity demand data that would inform subsequent modeling decisions.

Data Structure and Quality: The raw dataset "combined.csv" contained 192,673 rows and 2 columns, representing 5-minute interval measurements of electricity demand. Initial inspection using the *glimpse()* function showed that both variables originally labeled "SETTLEMENTDATE" and "TOTALDEMAND" were stored as character strings rather than appropriate date and numeric types. Importantly, the dataset contained no missing values, representing excellent data quality that eliminated the need for imputation strategies.

Temporal Coverage: The data spanned from January 2024 through October 2025, providing approximately 21 months of observations across different seasons and demand conditions.

Demand Range and Variation: After aggregation to daily maxima, statistical summary of the training set revealed:

- Minimum demand: 7,623 MW
- Maximum demand: 13,764 MW
- Mean demand: 9,750 MW
- Range: 6,141 MW

This wide range (approximately 6,000 MW or 80% of minimum demand) indicated substantial day-to-day variation in peak electricity consumption.

Visual Patterns: Time series plots revealed three notable patterns:

1. Upward Trend: The data showed considerable fluctuations with an overall increasing trend over the observation period, indicating growing electricity demand over time.
2. Weekly Seasonality: Regular weekly cycles were evident, with consistent patterns of higher demand during weekdays and lower demand on weekends. This pattern reflects the contribution of commercial and industrial loads that operate primarily during business hours.
3. Volatility: Day-to-day fluctuations were substantial, with demand typically ranging between 8,000 MW and 12,000 MW but occasionally exceeding these bounds during extreme conditions.

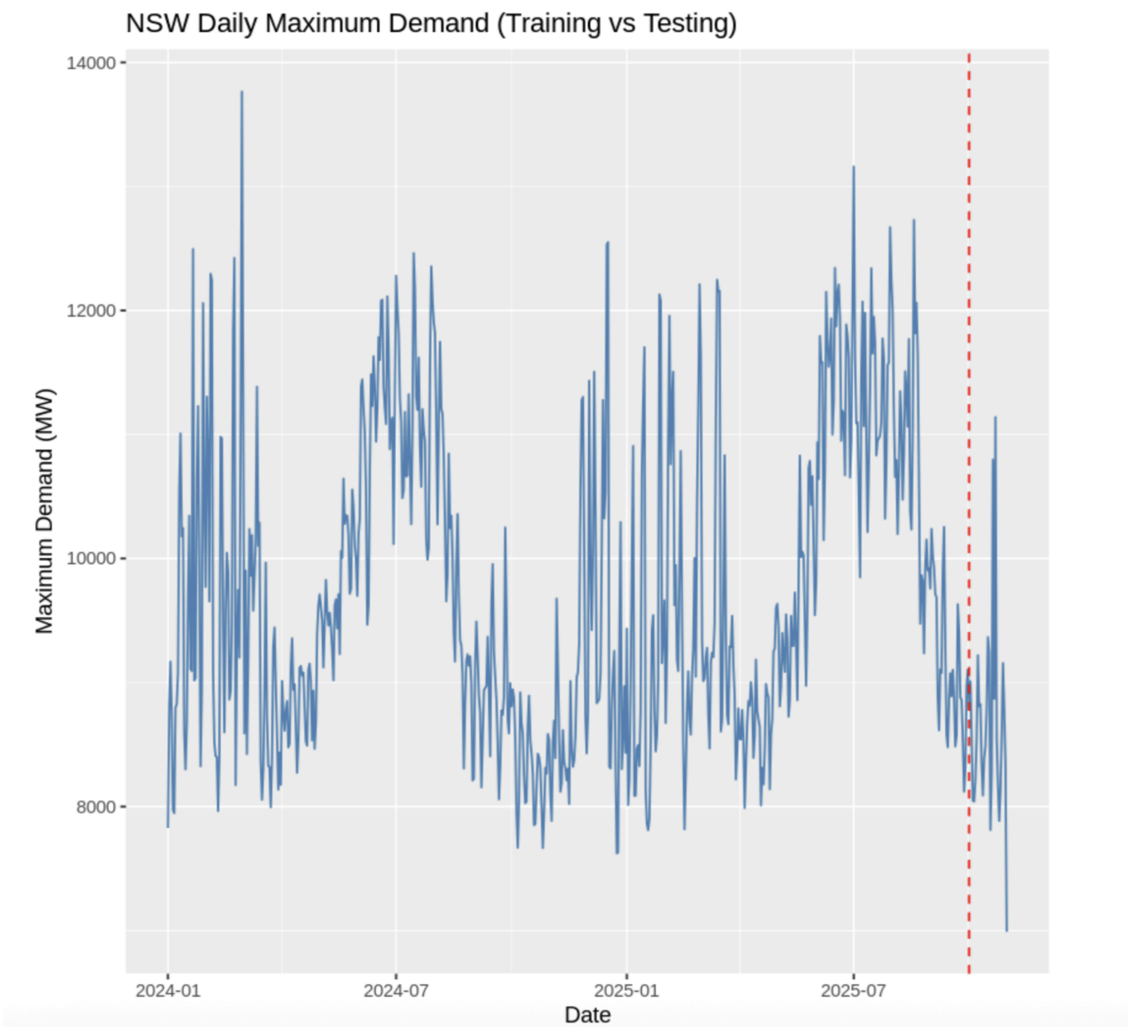


Figure 1: Times Series Plot of New South Wales Daily Maxim Electricity Demand

Non-Stationarity: The KPSS test returned a p-value of 0.023, indicating rejection of the null hypothesis of stationarity. This finding confirmed that the series exhibits non-stationary characteristics due to the upward trend, which would require either differencing (for ARIMA models) or explicit trend modeling (for ETS and TSLM approaches).

Decomposition Insights: Seasonal and Trend decomposition using Loess (STL) provided quantitative measures of series components:

- Strong Trend (strength = 0.80): Trend dynamics dominated overall variation, explaining approximately 80% of the systematic patterns in the data.

- Moderate Weekly Seasonality (strength = 0.39): Weekly cycles contributed meaningfully to variation but less dominantly than the trend. This moderate strength aligned with expectations that weekday-weekend differences are consistent but that other factors (likely weather) introduce additional variation.

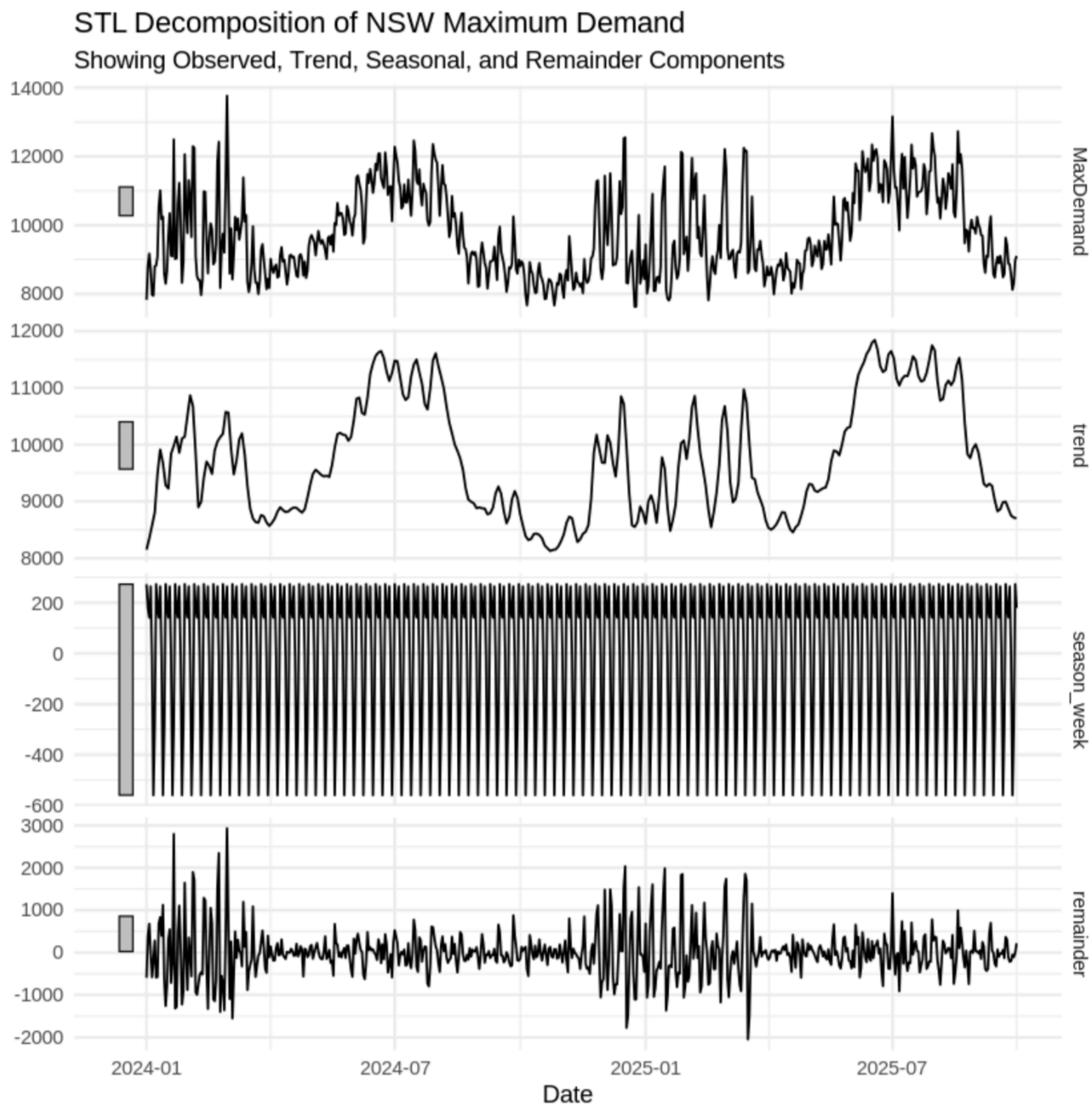


Figure 2: Seasonal decomposition of New South Wales Maximum Electricity Demand

- Structured Residuals (ACF1 = 0.24): High spikiness and residual autocorrelation of 0.24 indicated that the remainder component was not purely random. Substantial unexplained structure remained after accounting for trend and seasonality, suggesting that purely deterministic models would leave significant predictable variation uncaptured.

These findings collectively indicated that successful models would need to capture both the strong upward trend and moderate weekly seasonality while accommodating remaining temporal dependencies in the data.

Data Preparation and Cleaning Steps

Several preprocessing steps transformed the raw 5-minute interval data into a format suitable for daily maximum demand forecasting.

Type Conversion: The "Date" column (originally "SETTLEMENTDATE") was converted from character strings to proper Date objects using *as.Date()*, and the "MaxDemand" column (originally "TOTALDEMAND") was converted to numeric type using *as.numeric()*.

Missing Value Handling: The dataset was filtered to remove any NA values using *filter(!is.na(Date) & !is.na(MaxDemand))*. However, no missing values were found in the original AEMO dataset, so no observations were removed. No outliers requiring removal or special treatment were identified through visual inspection.

Aggregation to Daily Maximum: The 5-minute interval data (288 observations per day) was aggregated to daily maximum values using *group_by(Date)* followed by *summarize(MaxDemand = max(MaxDemand, na.rm = TRUE))*. This reduced the dataset from approximately 192,673 rows to 639 daily observations.

```
••• # A tsibble: 639 x 2 [1D]
      Date       MaxDemand
      <date>         <dbl>
    1 2024-01-01     7828.
    2 2024-01-02     8811.
    3 2024-01-03     9168.
    4 2024-01-04     8711.
    5 2024-01-05     7974.
    6 2024-01-06     7948.
    7 2024-01-07     8800.
    8 2024-01-08     8825.
    9 2024-01-09     9076.
   10 2024-01-10    10523.
    # ℹ 629 more rows
      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      7623    8749    9444    9750   10779   13764
```

Figure 3: Data Reduction from 192,673 rows to 639 observations

Time Series Object Creation: The cleaned data was converted into a *tsibble* object with Date as the time index. The *has_gaps()* function confirmed no temporal gaps existed in the daily series.

Train-Test Split: Data was split temporally at October 1, 2025, with all prior data used for training and October 2025 reserved for holdout testing.

Missing Variables: The dataset contained only Date and MaxDemand. Notable absent variables include temperature, weather data, calendar indicators (holidays), and day-type classifications, which could enhance forecast accuracy in future work.

Differencing: The KPSS test (p-value = 0.023) confirmed non-stationarity, indicating that ARIMA models would require first-order differencing (d=1) to achieve stationarity before fitting.

Transformation: Box-Cox transformation ($\lambda \approx -0.9$) was explored using the Guerrero method. While transformation improved residual normality and variance stability, forecast accuracy on the original series was comparable or better. Therefore, the untransformed series was used for primary results.

Smoothing: No explicit smoothing was applied as preprocessing. The exponential smoothing inherent in ETS models provided appropriate smoothing as part of the forecasting methodology.

Which Models Were Used and Why

Three time series forecasting approaches were applied and evaluated in this study: ARIMA (AutoRegressive Integrated Moving Average), TSLM (Time Series Linear Model), and ETS (Exponential Smoothing). Both manual and automatic specifications of each modeling approach were tested to ensure that model selection was comprehensive and data-driven. The ARIMA approach was included because of its long-standing use in modeling autocorrelation and handling non-stationary time series through differencing. However, while ARIMA models were able to capture short-term dependencies, they were less effective in modeling the recurring weekly patterns present in the New South Wales (NSW) electricity demand data.

Time Series Linear Models (TSLM) were also employed due to their ability to incorporate explicit trend and seasonal components in a regression-based framework. Although these models provided interpretability, including statistically significant trend terms, they failed to adequately represent the magnitude and complexity of the observed fluctuations in demand. Both the manual and automatic TSLM models produced large forecasting errors, indicating that a purely linear structure was insufficient for this dataset.

The Exponential Smoothing approach produced the most accurate and reliable results. Specifically, the ETS(M,A,M) model—characterized by multiplicative error, an additive trend, and multiplicative seasonal components—proved to be the most effective. This model was able to adapt to changes in demand variance while simultaneously capturing both the upward trend and weekly seasonal dynamics present in the data. As a result, ETS(M,A,M) was selected as the final model for forecasting daily maximum electricity demand in NSW.

Evaluation Metrics Used

Model	RMSE (MW)	MAPE (%)
✅ ETS(M,A,M)	**686**	**4.97**
Best ARIMA	742	6.42
Best TSLM	1556	17.69

Figure 4: RMSE and MAPE metrics Comparison

Model performance was assessed using multiple accuracy metrics on a reserved holdout test set consisting of data from October 2025. The primary evaluation measure was Root Mean Squared Error (RMSE), which quantified the average magnitude of forecast errors in megawatts and penalized large deviations more heavily than smaller ones. In addition, Mean Absolute Error (MAE) was calculated to provide a direct and easily interpretable measurement of average forecast deviation in megawatts.

Mean Absolute Percentage Error (MAPE) was also used to assess forecast accuracy as a relative percentage, allowing performance to be evaluated independently of scale. This metric was especially important for operational decision-making, as it provided a practical interpretation of forecast reliability. Mean Error (ME) was included to identify whether the models exhibited systematic bias toward overestimation or underestimation. Finally, residuals from each model were examined using diagnostic tests, such as the Ljung-Box test, to determine whether remaining errors resembled white noise, indicating that no further structure remained unmodeled.

The ETS(M,A,M) model outperformed all other models according to these metrics, achieving an RMSE of approximately 686 MW and a MAPE of 4.97%, which met the project's accuracy threshold for operational forecasting.

Preliminary and Interesting Results

The analysis revealed a strong upward trend in electricity demand across the observed period, indicating consistent growth in energy consumption in New South Wales. Decomposition of the time series showed that the trend component accounted for approximately 80% of the systematic variation in demand. In addition, a clear weekly seasonal pattern was observed, with demand peaking during weekdays and declining over weekends. This pattern was consistent across the entire time horizon, highlighting the influence of work-week activity on electricity consumption.

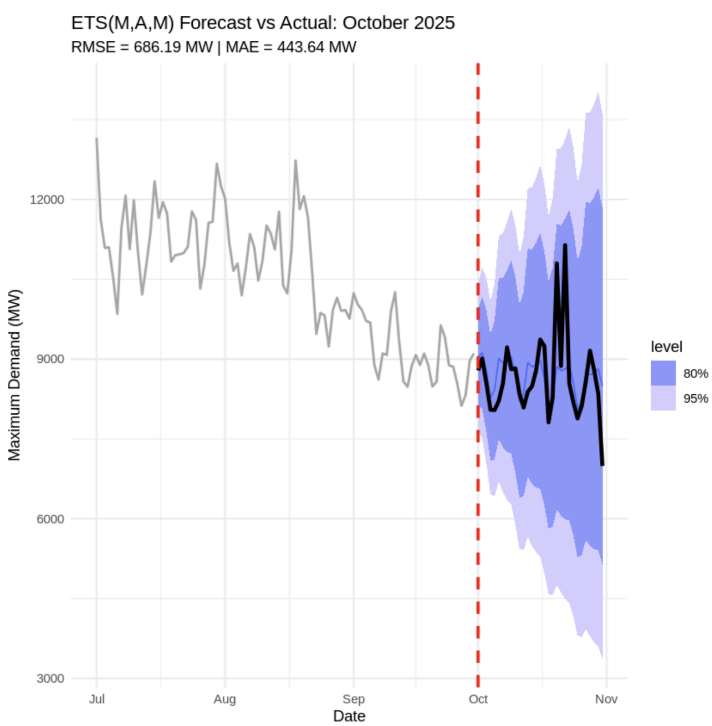


Figure 5: Forecast Validation for NSW Daily Maximum, black line shows actual observed Electricity Demand (October 2025)

When models were compared, the ETS(M,A,M) model significantly outperformed both ARIMA and TSLM alternatives as ETS produced an RMSE of 686 MW. The best-performing ARIMA model produced an RMSE of approximately 742 MW, while TSLM models produced RMSE values exceeding 1,500 MW. These results demonstrated that linear regression-based approaches failed to capture the observed variability in the data. Although a Box-Cox transformation was explored using the Guerrero method and resulted in improved variance stabilization, the transformed model produced higher RMSE values after back-transformation. As a result, the original untransformed series was retained for the final modeling process.

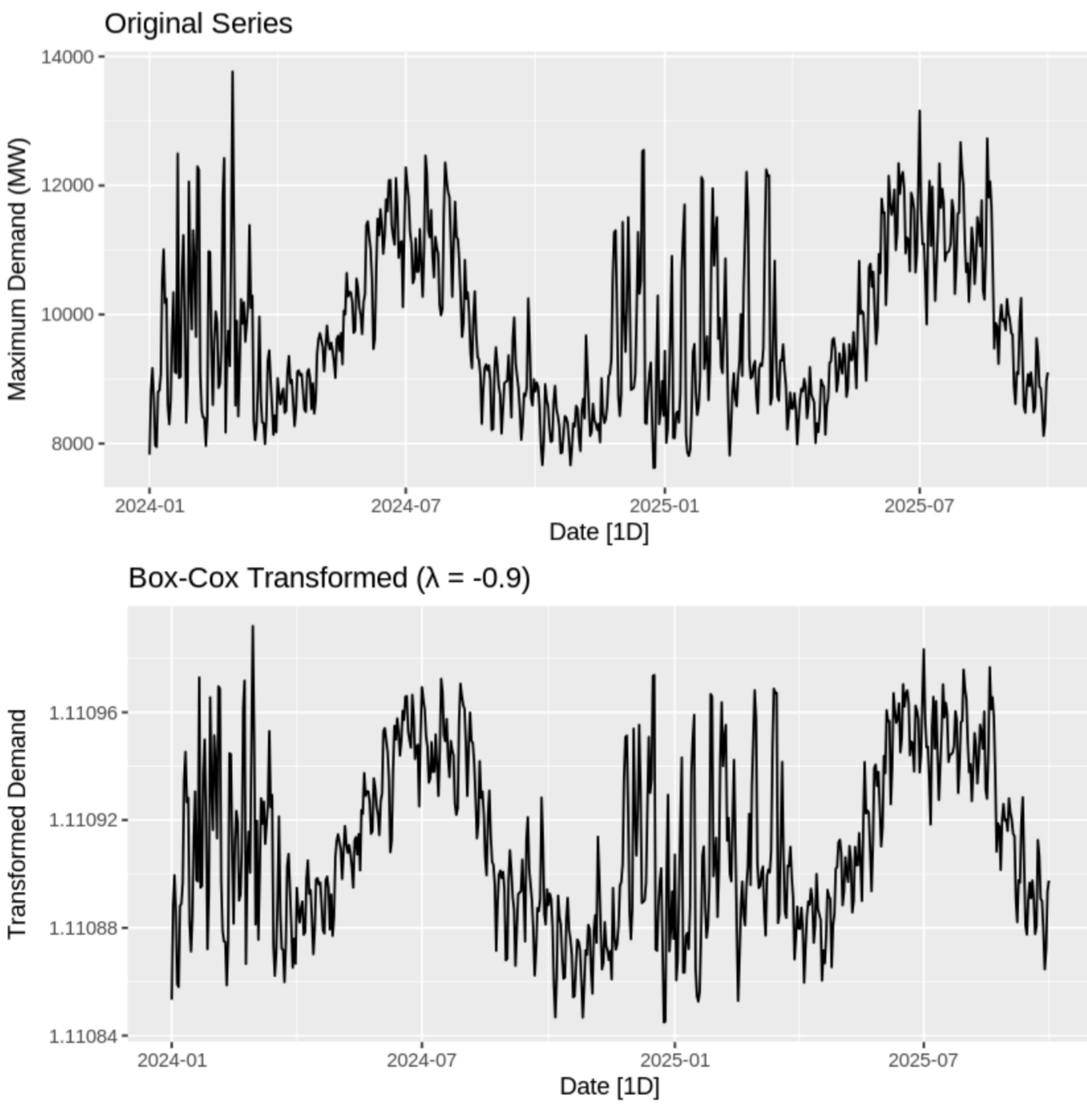


Figure 6: Original vs Box-Cox Transformed Series

Residual diagnostics further demonstrated that, while much of the structure in the data was captured by the ETS model, some residual autocorrelation remained. This suggests that additional explanatory variables, most notably weather-related factors such as temperature, could further improve model accuracy if incorporated in future studies.

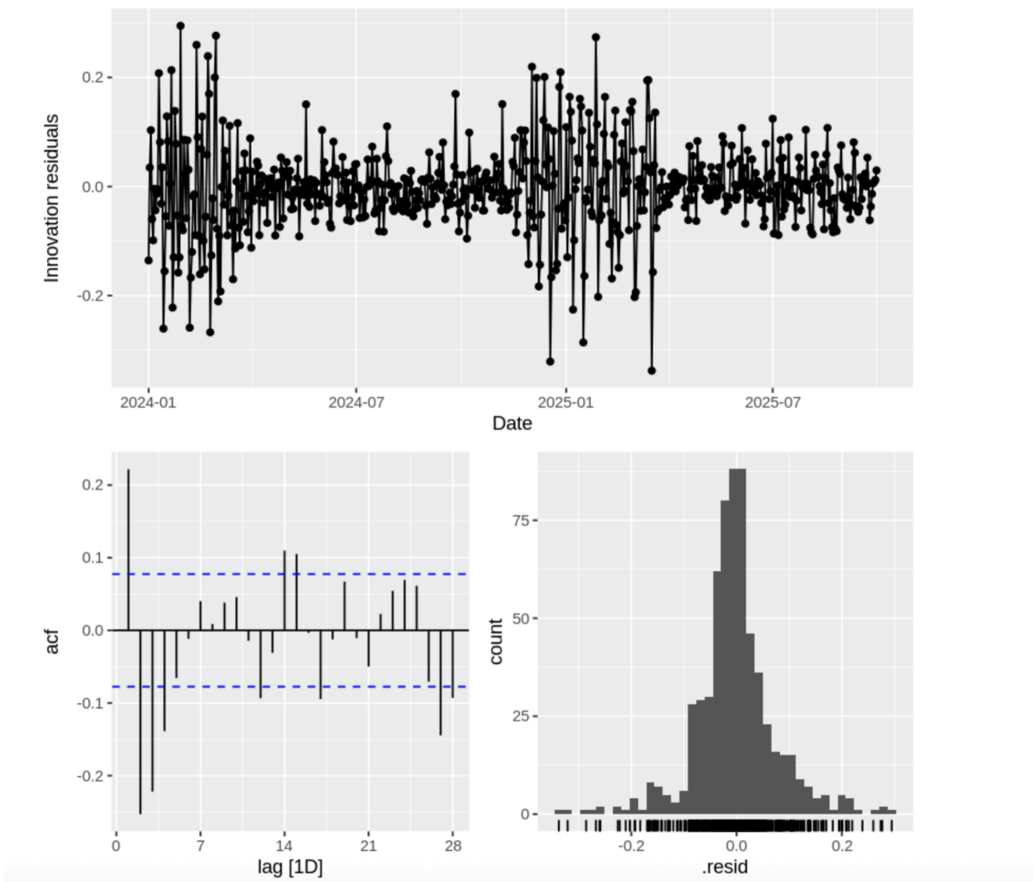


Figure 7: ETS (M, A, M) Residual Diagnostic Panel - 3 plots

Discussions

The findings of this study confirm that daily maximum electricity demand in New South Wales is driven by both long-term growth patterns and predictable short-term seasonal cycles. The ability of the ETS(M,A,M) model to successfully capture these components demonstrates its suitability for operational forecasting within the energy sector. Its relatively low error rate and stable performance across the test period indicate that exponential smoothing is a practical and effective choice for this application.

For grid operators, these results offer a reliable method for anticipating peak demand and appropriately allocating generation resources. Accurate forecasting at the daily level assists in preventing power outages, minimizing the cost of excess generation, and ensuring that sufficient reserves are available during periods of peak usage. The observed weekly variations also provide valuable insights into demand behavior associated with business and non-business days, which can inform maintenance scheduling and operational planning.

Despite its strengths, the model is limited by its reliance on a single variable—historical demand. The residual patterns observed in diagnostic tests indicate that important factors such as temperature extremes, public holidays, and changes in energy policy or technology adoption were not captured. Incorporating such external regressors through models such as SARIMAX, Dynamic Regression, or machine learning techniques (e.g., LSTM networks) could further improve forecasting precision. Additionally, expanding the modeling framework to incorporate hourly or intraday forecasting could provide even more granular insights for grid optimization.

Conclusion

This project successfully developed and evaluated a time series forecasting framework for predicting daily maximum electricity demand in New South Wales. Through comprehensive data preprocessing, systematic model comparison, and rigorous diagnostic testing, the ETS(M,A,M) model emerged as the most effective approach. It consistently outperformed both ARIMA and TSLM models, achieving an RMSE of approximately 686 MW and a MAPE below 5 percent, satisfying the predefined criteria for operational usability.

The findings highlight the importance of accounting for both trend and seasonality when modeling electricity demand and provide clear evidence that exponential smoothing methods offer a strong balance between accuracy, stability, and interpretability. Beyond academic application, the selected model has practical value for energy providers in planning generation, managing resources, and enhancing grid reliability. With the integration of exogenous variables and more advanced modeling techniques in future work, forecasting performance could be further improved to support long-term energy planning and resilience in an evolving demand environment.

References and Appendices

AEMO. (2025). *Aggregated Price and Demand Data – New South Wales*. Australian Energy Market Operator. Retrieved from (https://www.aemo.com.au/energy-systems/electricity/national-electricity-market-nem/data-nem/aggregated-data)

Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: Principles and Practice* (3rd ed.). OTexts.

Taylor, J. W. (2010). Triple seasonal methods for short-term electricity demand forecasting. *European Journal of Operational Research, 204*(1), 139–152.

Kandananond, K. (2011). Forecasting electricity demand in Thailand with an artificial neural network approach. *Energies, 4*(8), 1246–1257.

Hong, T., Pinson, P., & Fan, S. (2020). Global Energy Forecasting Competition 2012: Load forecasting and wind power forecasting. *International Journal of Forecasting, 36*(1), 1–9.

In [ ]:
```
install.packages("fpp3")
install.packages("urca")
install.packages("gridExtra")
install.packages("forecast")

# Load necessary libraries
suppressPackageStartupMessages({
  suppressWarnings({
    library(fpp3) # for time series models (ARIMA, ETS, TSLM, etc.)
    library(ggplot2) # for visualization
    library(dplyr)
    library(tsibble)
    library(fable)
    library(gridExtra)
    library(ggplot2)
    library(fabletools)
  })
})
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependencies ‘progressr’, ‘ggdist’, ‘numDeriv’, ‘warp’, ‘BH’, ‘fabletools’, ‘distributional’
, ‘slider’, ‘anytime’, ‘fable’, ‘feasts’, ‘tsibble’, ‘tsibbledata’


Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

In [ ]:
```
# Load and prepare the dataset
nsw_data <- read.csv("combine.csv", header = FALSE,
                     col.names = c("Region", "Date", "MaxDemand", "RRP", "PeriodType"))
nsw_data <- nsw_data[, c("Date", "MaxDemand")]
```

In [ ]:
```
# Inspect structure
glimpse(nsw_data)
```

```
Rows: 192,673
Columns: 2
$ Date      <chr> "SETTLEMENTDATE", "2024/01/01 00:05:00", "2024/01/01 00:10:0…
$ MaxDemand <chr> "TOTALDEMAND", "6574.92", "6651.09", "6538.96", "6497.99", "…
```

In [ ]:
```
nsw_data <- nsw_data %>%
  as_tibble() %>%
  mutate(
    Date = as.Date(Date, format = "%Y/%m/%d"),
    MaxDemand = as.numeric(MaxDemand)  # Convert to numeric
  ) %>%
  filter(!is.na(Date) & !is.na(MaxDemand)) %>%  # Remove rows with NA dates or demand
  group_by(Date) %>%
  summarise(MaxDemand = max(MaxDemand, na.rm = TRUE)) %>%
  ungroup() %>%
  as_tsibble(index = Date)
```

```
Warning message:
“There was 1 warning in `mutate()`.
i In argument: `MaxDemand = as.numeric(MaxDemand)`.
Caused by warning:
! NAs introduced by coercion”
```

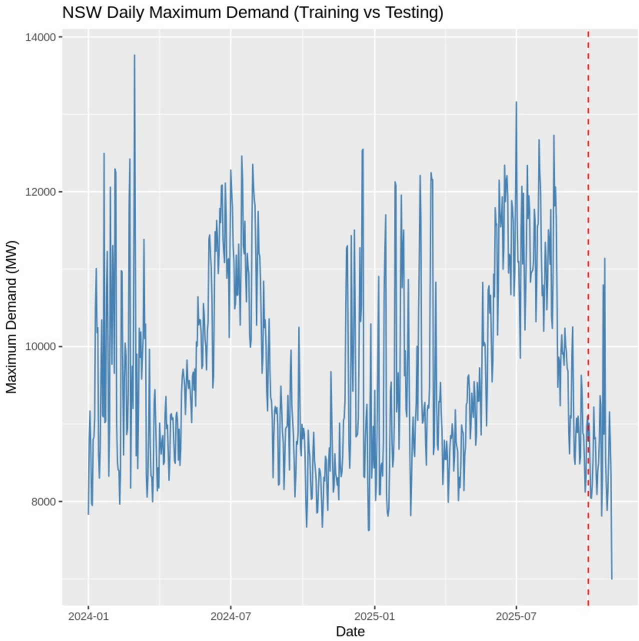
In [ ]:
```
nsw_data <- nsw_data %>%
  as_tsibble(index = Date)
```

In [ ]:
```
train_data <- nsw_data %>%
  filter(Date < as.Date("2025-10-01")) %>%
  filter(!is.na(MaxDemand))

test_data <- nsw_data %>%
  filter(Date >= as.Date("2025-10-01")) %>%
  filter(!is.na(MaxDemand))
```

In [ ]:
```
# Split into Training and Testing sets
# Training: all data up to September 2025, Testing: October 2025
train_data <- nsw_data %>% filter(Date < as.Date("2025-10-01"))
```

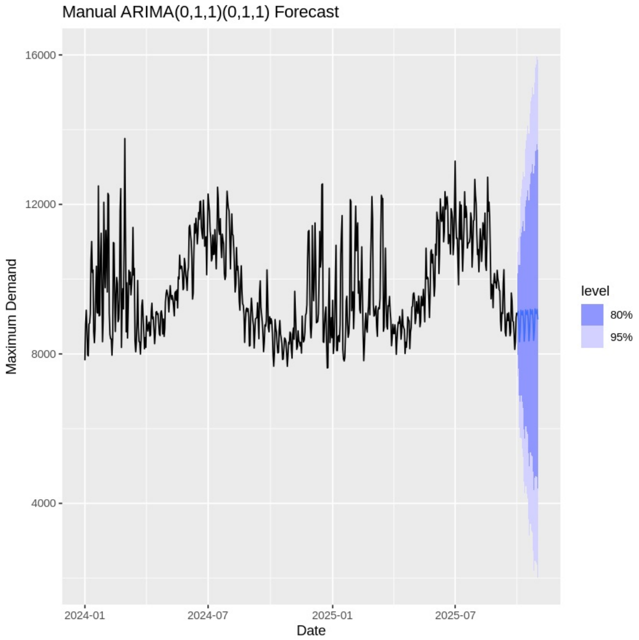
```
test_data <- nsw_data %>% filter(Date >= as.Date("2025-10-01"))
```

In [ ]:
```
# Visualize the split
nsw_data %>%
  ggplot(aes(x = Date, y = MaxDemand)) +
  geom_line(color = "steelblue") +
  geom_vline(xintercept = as.Date("2025-10-01"),
             linetype = "dashed", color = "red") +
  labs(title = "NSW Daily Maximum Demand (Training vs Testing)",
       y = "Maximum Demand (MW)", x = "Date")
```



In [ ]:
```
# Fit Manual ARIMA Model
  manual_arima <- train_data %>%
  model(ARIMA(MaxDemand ~ pdq(0,1,1) + PDQ(0,1,1)))

report(manual_arima)

# Forecast on test data
manual_forecast <- forecast(manual_arima, h = nrow(test_data))

# Plot manual ARIMA results
autoplot(manual_forecast, train_data) +
  labs(title = "Manual ARIMA(0,1,1)(0,1,1) Forecast",
       y = "Maximum Demand", x = "Date")
```

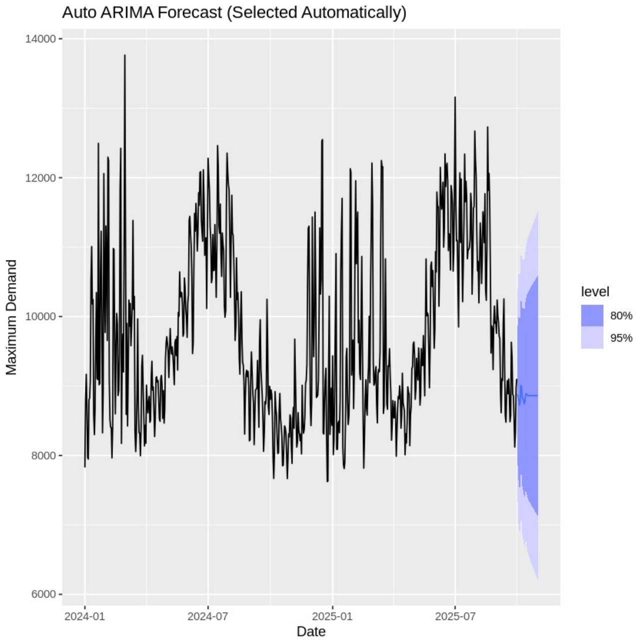
```
Series: MaxDemand
Model: ARIMA(0,1,1)(0,1,1)[7]

Coefficients:
          ma1     sma1
      -0.1880  -1.0000
s.e.   0.0773   0.0233

sigma^2 estimated as 574261:  log likelihood=-5093.95
AIC=10193.91   AICc=10193.95   BIC=10207.25
```



In [ ]:

```
# Fit Auto ARIMA Model
auto_arima <- train_data %>%
  model(ARIMA(MaxDemand))

report(auto_arima)

# Forecast on test data
auto_forecast <- forecast(auto_arima, h = nrow(test_data))

# Plot auto ARIMA results
autoplot(auto_forecast, train_data) +
  labs(title = "Auto ARIMA Forecast (Selected Automatically)",
       y = "Maximum Demand", x = "Date")
```

```
Series: MaxDemand
Model: ARIMA(1,1,2)(0,0,2)[7]

Coefficients:
         ar1      ma1      ma2    sma1    sma2
      0.2107  -0.4296  -0.4167  0.1310  0.1871
s.e.  0.0721   0.0656   0.0489  0.0402  0.0402

sigma^2 estimated as 503428:  log likelihood=-5091.67
AIC=10195.35   AICc=10195.48   BIC=10222.1
```



In [ ]:

```r
# Compare Results

# Combine forecasts for comparison
accuracy_results <- bind_rows(
  accuracy(manual_forecast, test_data),
  accuracy(auto_forecast, test_data),
  .id = "Model"
)

accuracy_results
```

A tibble: 2 × 11

Model <chr>	.model <chr>	.type <chr>	ME <dbl>	RMSE <dbl>	MAE <dbl>	MPE <dbl>	MAPE <dbl>	MASE <dbl>	RMSSE <dbl>	ACF1 <dbl>
1	ARIMA(MaxDemand ~ pdq(0, 1, 1) + PDQ(0, 1, 1))	Test	-267.3521	741.9146	554.8150	-3.723085	6.424959	NaN	NaN	0.07301251
2	ARIMA(MaxDemand)	Test	-198.4321	800.2648	583.5513	-3.051071	6.754740	NaN	NaN	0.13823531

In [ ]:

```r
# Fit Manual ETS Model
print(train_data)
summary(train_data$MaxDemand)
has_gaps(train_data)
```

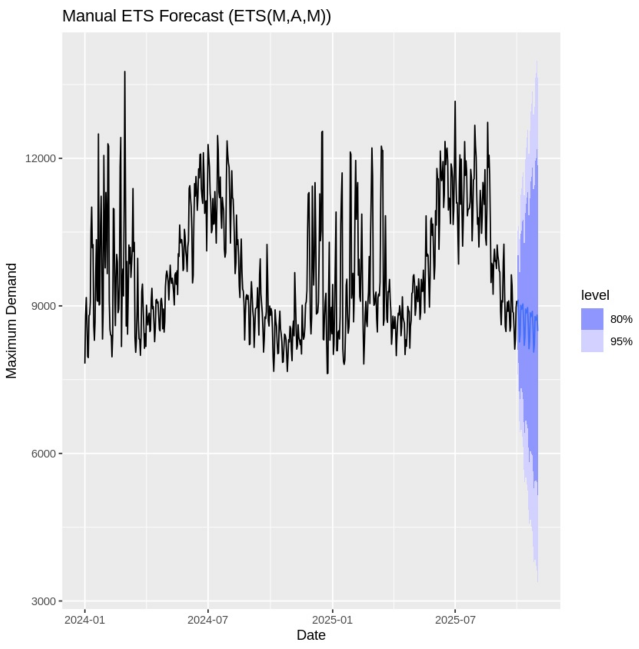
```
# A tsibble: 639 x 2 [1D]
   Date       MaxDemand
   <date>         <dbl>
 1 2024-01-01     7828.
 2 2024-01-02     8811.
 3 2024-01-03     9168.
 4 2024-01-04     8711.
 5 2024-01-05     7974.
 6 2024-01-06     7948.
 7 2024-01-07     8800.
 8 2024-01-08     8825.
 9 2024-01-09     9076.
10 2024-01-10    10523.
# ℹ 629 more rows
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   7623    8749    9444    9750   10779   13764
```

A tibble:
1 × 1

.gaps
<lgl>
FALSE

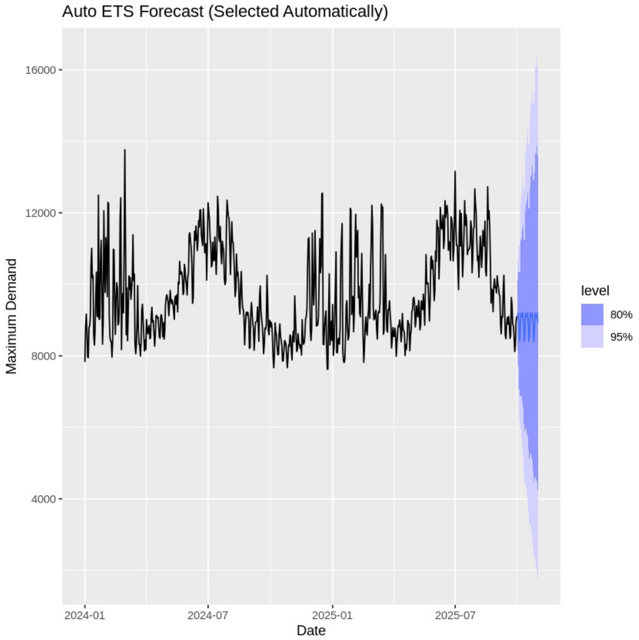
In [ ]:
```
# Fit Manual ETS Model

ets_manual <- train_data %>%
  model(ETS(MaxDemand ~ error("M") + trend("A") + season("M")))

report(ets_manual)

ets_manual_fc <- forecast(ets_manual, h = nrow(test_data))

autoplot(ets_manual_fc, train_data) +
  labs(
    title = "Manual ETS Forecast (ETS(M,A,M))",
    y = "Maximum Demand", x = "Date"
  )
```

```
Series: MaxDemand
Model: ETS(M,A,M)
  Smoothing parameters:
    alpha = 0.5848713
    beta  = 0.005340058
    gamma = 0.0449898

  Initial states:
     l[0]     b[0]     s[0]     s[-1]     s[-2]    s[-3]     s[-4]    s[-5]
 8601.718 155.5825 1.047225 0.9217148 0.9793855 1.006739 0.9741256 1.036788
    s[-6]
 1.034023

  sigma^2:  0.006

     AIC     AICc      BIC
12606.67 12607.17 12660.19
```



In [ ]:
```
# Fit Auto ETS Model

ets_auto <- train_data %>%
  model(ETS(MaxDemand))

report(ets_auto)

ets_auto_fc <- forecast(ets_auto, h = nrow(test_data))

autoplot(ets_auto_fc, train_data) +
  labs(
    title = "Auto ETS Forecast (Selected Automatically)",
    y = "Maximum Demand", x = "Date"
  )
```

```
Series: MaxDemand
Model: ETS(M,Ad,M)
  Smoothing parameters:
    alpha = 0.9685772
    beta  = 0.0001000675
    gamma = 0.0001001072
    phi   = 0.9776881

  Initial states:
     l[0]     b[0]      s[0]     s[-1]     s[-2]    s[-3]    s[-4]    s[-5]
 8130.745 174.5486 0.9680079 0.9403061 0.9975744 1.029043 1.018577 1.017628
    s[-6]
 1.028864

  sigma^2:  0.0053

     AIC     AICc      BIC
12523.74 12524.32 12581.72
```



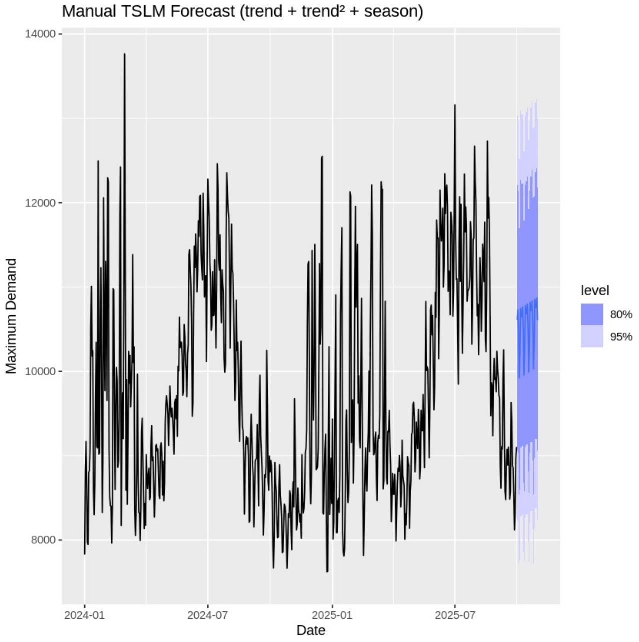
In [ ]:
```
# Compare Results

ets_accuracy_results <- bind_rows(
  accuracy(ets_manual_fc, test_data),
  accuracy(ets_auto_fc, test_data),
  .id = "Model"
)

ets_accuracy_results
```

A tibble: 2 × 11

Model	.model	.type	ME	RMSE	MAE	MPE	MAPE	MASE	RMSSE	ACF1
<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	ETS(MaxDemand ~ error("M") + trend("A") + season("M"))	Test	-31.79808	686.1925	443.6372	-0.9631127	4.966928	NaN	NaN	0.09787900
2	ETS(MaxDemand)	Test	-297.36381	750.4297	575.4585	-4.0728520	6.677757	NaN	NaN	0.07838189

In [ ]:
```
# Fit Manual TSLM Model

tslm_manual <- train_data %>%
  model(TSLM(MaxDemand ~ trend() + I(trend()^2) + season()))

report(tslm_manual)

tslm_manual_fc <- forecast(tslm_manual, h = nrow(test_data))

autoplot(tslm_manual_fc, train_data) +
  labs(
    title = "Manual TSLM Forecast (trend + trend² + season)",
    y = "Maximum Demand", x = "Date"
  )
```

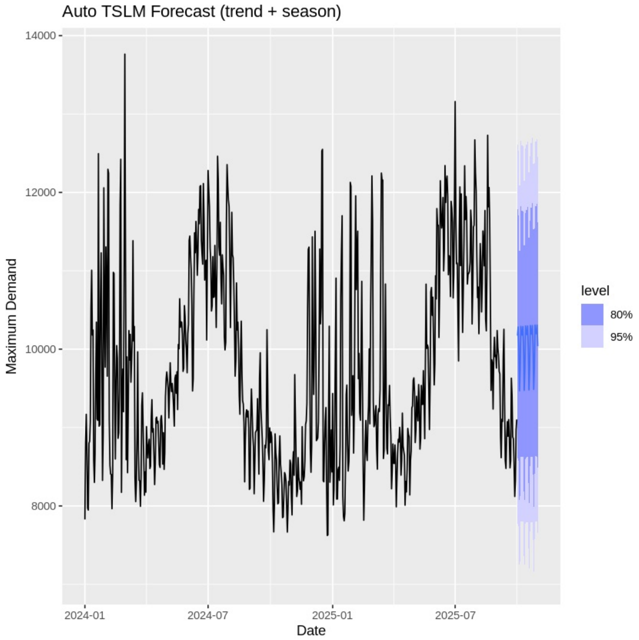
```
Series: MaxDemand
Model: TSLM

Residuals:
    Min      1Q  Median      3Q     Max
-2345.9  -942.2  -315.2   964.6  3758.2

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.018e+04  1.841e+02  55.288  < 2e-16 ***
trend()       -3.255e+00  1.029e+00  -3.163  0.00164 **
I(trend()^2)   6.415e-03  1.557e-03   4.120 4.29e-05 ***
season()week2 -2.687e+02  1.776e+02  -1.513  0.13078
season()week3 -8.158e+02  1.776e+02  -4.594 5.25e-06 ***
season()week4 -5.427e+02  1.776e+02  -3.056  0.00234 **
season()week5 -1.134e+00  1.771e+02  -0.006  0.99489
season()week6 -9.740e+01  1.771e+02  -0.550  0.58254
season()week7 -1.092e+02  1.776e+02  -0.615  0.53868
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1198 on 630 degrees of freedom
Multiple R-squared: 0.09274,    Adjusted R-squared: 0.08122
F-statistic:  8.05 on 8 and 630 DF, p-value: 2.2575e-10
```



In [ ]:

```
# Fit Auto TSLM Model

tslm_auto <- train_data %>%
  model(TSLM(MaxDemand ~ trend() + season()))

report(tslm_auto)

tslm_auto_fc <- forecast(tslm_auto, h = nrow(test_data))

autoplot(tslm_auto_fc, train_data) +
  labs(
    title = "Auto TSLM Forecast (trend + season)",
    y = "Maximum Demand", x = "Date"
  )
```

```
Series: MaxDemand
Model: TSLM

Residuals:
    Min      1Q  Median      3Q     Max
-2327.3  -970.2  -286.0  1029.9  3975.0

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   9737.9366   151.8291  64.137  < 2e-16 ***
trend()          0.8504     0.2601   3.269  0.00114 **
season()week2 -268.6857   179.8141  -1.494  0.13561
season()week3 -815.8165   179.8147  -4.537 6.83e-06 ***
season()week4 -542.6730   179.8156  -3.018  0.00265 **
season()week5    3.6550   179.3247   0.020  0.98375
season()week6  -92.6108   179.3251  -0.516  0.60573
season()week7 -109.2198   179.8141  -0.607  0.54380
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1213 on 631 degrees of freedom
Multiple R-squared: 0.06829,    Adjusted R-squared: 0.05796
F-statistic: 6.607 on 7 and 631 DF, p-value: 1.489e-07
```



In [ ]:
```
# Compare Results

tslm_accuracy_results <- bind_rows(
  accuracy(tslm_manual_fc, test_data),
  accuracy(tslm_auto_fc, test_data),
  .id = "Model"
)

tslm_accuracy_results
```

A tibble: 2 × 11

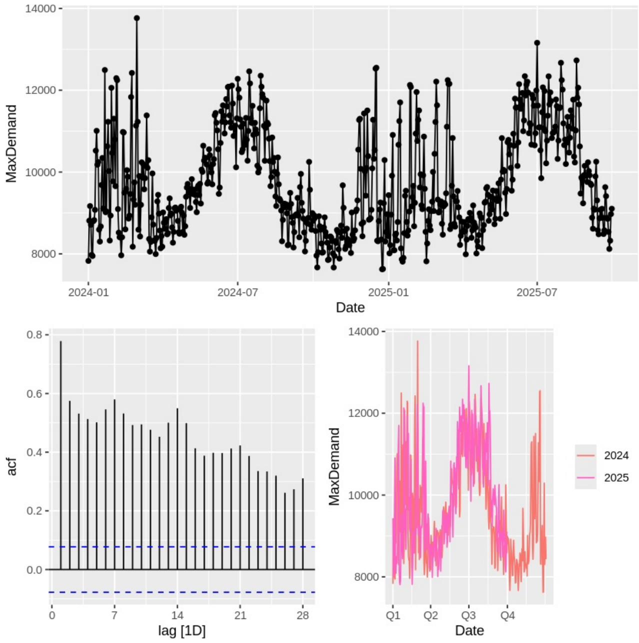
Model	.model	.type	ME	RMSE	MAE	MPE	MAPE	MASE	RMSSE	ACF1
<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	TSLM(MaxDemand ~ trend() + I(trend()^2) + season())	Test	-1896.603	2020.106	1923.683	-22.69683	22.93992	NaN	NaN	0.07647750
2	TSLM(MaxDemand ~ trend() + season())	Test	-1394.280	1556.535	1487.384	-16.84493	17.68982	NaN	NaN	0.07468156

In [ ]:
```
# Convert the notebook to HTML using the system command
system("jupyter nbconvert --to html /content/MaximumDemand.ipynb")
```

In [ ]:
```
#Discussion 4.1: Transformations and Time Series Cross-Validation
```

In [ ]:
```
#STATIONARITY AND DIAGNOSTICS
# Visual check for patterns
train_data %>% gg_tsdisplay(MaxDemand)
```

```
Warning message:
“`gg_tsdisplay()` was deprecated in feasts 0.4.2.
ℹ Please use `ggtime::gg_tsdisplay()` instead.”
```



In [ ]:

```
#Statistical tests for stationarity
train_data %>%
  features(MaxDemand, c(unitroot_kpss, unitroot_ndiffs, unitroot_nsdiffs))
```

A tibble: 1 × 4

kpss_stat	kpss_pvalue	ndiffs	nsdiffs
<dbl>	<dbl>	<int>	<int>
0.5937108	0.02320811	1	0

In [ ]:

```
#Check for heteroscedasticity and skewness
```

In [ ]:

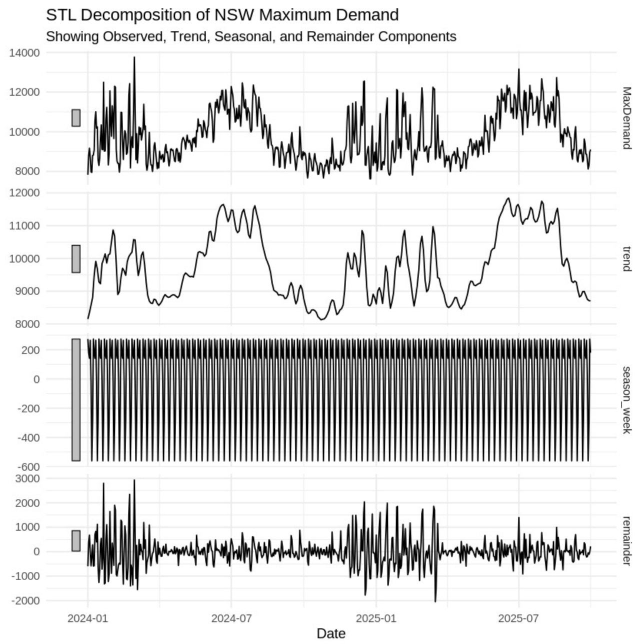
```
train_data %>%
  features(MaxDemand, feat_stl)
```

A tibble: 1 × 9

trend_strength	seasonal_strength_week	seasonal_peak_week	seasonal_trough_week	spikiness	linearity	curvature	stl_e_acf1	stl_
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
0.7970049	0.3857624	1	6	1033589	3967.685	4952.131	0.2394583	0.:

In [ ]:

```
# STL Decomposition for NSW Maximum Demand

stl_decomp <- train_data %>%
  model(STL(MaxDemand ~ season(window = "periodic"))) %>%
  components()

# Create the STL decomposition plot
stl_plot <- stl_decomp %>%
  autoplot() +
  labs(
    title = "STL Decomposition of NSW Maximum Demand",
    subtitle = "Showing Observed, Trend, Seasonal, and Remainder Components",
    x = "Date"
  ) +
  theme_minimal()

# Display the plot
print(stl_plot)
```



In [ ]:
```
cat("\nSTL Decomposition Statistics:\n")
cat("Trend Strength:", round(train_data %>% features(MaxDemand, feat_stl) %>% pull(trend_strength), 4), "\n")
cat("Seasonal Strength (Weekly):", round(train_data %>% features(MaxDemand, feat_stl) %>% pull(seasonal_strengt
cat("Remainder ACF1:", round(train_data %>% features(MaxDemand, feat_stl) %>% pull(stl_e_acf1), 4), "\n")
```

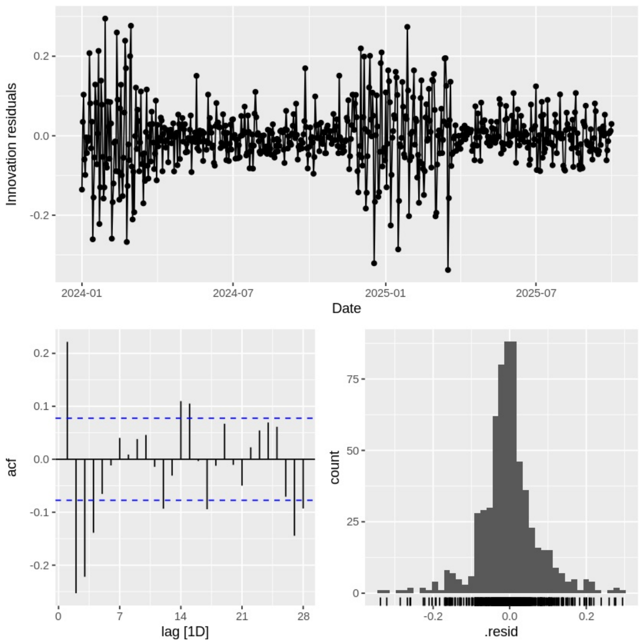
```
STL Decomposition Statistics:
Trend Strength: 0.797
Seasonal Strength (Weekly): 0.3858
Remainder ACF1: 0.2395
```

In [ ]:
```
# Best model from Week 3: ETS(M,A,M)
# Manual ETS(M,A,M) model with multiplicative error, additive trend, and multiplicative seasonality.

ets_manual <- train_data %>%
  model(ETS(MaxDemand ~ error("M") + trend("A") + season("M")))

report(ets_manual)
```

```
Series: MaxDemand
Model: ETS(M,A,M)
  Smoothing parameters:
    alpha = 0.5848713
    beta  = 0.005340058
    gamma = 0.0449898

  Initial states:
     l[0]     b[0]     s[0]     s[-1]     s[-2]    s[-3]     s[-4]    s[-5]
 8601.718 155.5825 1.047225 0.9217148 0.9793855 1.006739 0.9741256 1.036788
    s[-6]
 1.034023

  sigma^2:  0.006

     AIC     AICc      BIC
12606.67 12607.17 12660.19
```

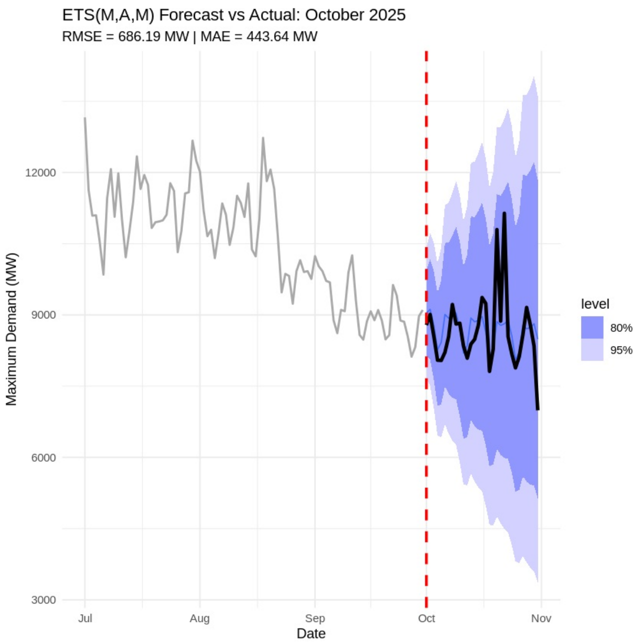
In [ ]:
```
ets_manual %>%
  gg_tsresiduals(type = "innovation")
```

```
Warning message:
“`gg_tsresiduals()` was deprecated in feasts 0.4.2.
i Please use `ggtime::gg_tsresiduals()` instead.”
```



In [ ]:

```r
train_data %>%
  filter(Date >= as.Date("2025-07-01")) %>%
  autoplot(MaxDemand, color = "darkgray", size = 0.8) +
  autolayer(ets_manual_fc, level = c(80, 95)) +
  geom_line(data = test_data, aes(x = Date, y = MaxDemand),
            color = "black", size = 1.3) +
  geom_vline(xintercept = as.Date("2025-10-01"),
             linetype = "dashed", color = "red", size = 1) +
  labs(
    title = "ETS(M,A,M) Forecast vs Actual: October 2025",
    subtitle = sprintf("RMSE = %.2f MW | MAE = %.2f MW",
                       sqrt(mean((ets_manual_fc$.mean - test_data$MaxDemand)^2)),
                       mean(abs(ets_manual_fc$.mean - test_data$MaxDemand))),
    y = "Maximum Demand (MW)",
    x = "Date"
  ) +
  theme_minimal()
```




```
In [ ]: #Ljung-Box test for white noise
ets_manual %>%
  augment() %>%
  features(.innov, ljung_box, lag = 14)
```

A tibble: 1 × 3

.model	lb_stat	lb_pvalue
<chr>	<dbl>	<dbl>
ETS(MaxDemand ~ error("M") + trend("A") + season("M"))	137.4879	0

```
In [ ]: #Applying Transformations
lambda_guerrero <- guerrero(train_data$MaxDemand)
print(paste("Guerrero lambda:", lambda_guerrero))
```

```
[1] "Guerrero lambda: -0.899926773224204"
```

```
In [ ]: cat("\nOptimal Box-Cox lambda (Guerrero):", round(lambda_guerrero, 3), "\n")
```

```
Optimal Box-Cox lambda (Guerrero): -0.9
```

```
In [ ]: #Refit the model on Box-Cox transformed data
lambda <- guerrero(train_data$MaxDemand)
train_transformed <- train_data %>%
  mutate(MaxDemand_BC = box_cox(MaxDemand, lambda))

ets_transformed <- train_transformed %>%
  model(ETS(MaxDemand_BC ~ error("M") + trend("A") + season("M")))
```

```
In [ ]: report(ets_transformed)
```

```
Series: MaxDemand_BC
Model: ETS(M,A,M)
  Smoothing parameters:
    alpha = 0.1727601
    beta  = 0.0002445236
    gamma = 0.04277281

  Initial states:
     l[0]          b[0]     s[0]     s[-1]     s[-2]    s[-3]    s[-4]
 1.110887 -1.794874e-08 1.000002 0.9999883 0.9999989 1.000006 1.000001
     s[-5]    s[-6]
 0.9999975 1.000006

  sigma^2:  0

      AIC      AICc       BIC
-9645.112 -9644.613 -9591.593
```

```
In [ ]: fc_transformed <- forecast(ets_transformed, h = nrow(test_data))
```

```
In [ ]: #Back-transform to original scale
fc_back <- inv_box_cox(fc_transformed$.mean, lambda)
```

```
In [ ]: #Accuracy
tibble(
  Model = c("Original", "Transformed"),
```

```r
  RMSE = c(
    sqrt(mean((ets_manual_fc$.mean - test_data$MaxDemand)^2)),
    sqrt(mean((fc_back - test_data$MaxDemand)^2))
  ),
  MAE = c(
    mean(abs(ets_manual_fc$.mean - test_data$MaxDemand)),
    mean(abs(fc_back - test_data$MaxDemand))
  )
)
```

A tibble: 2 × 3

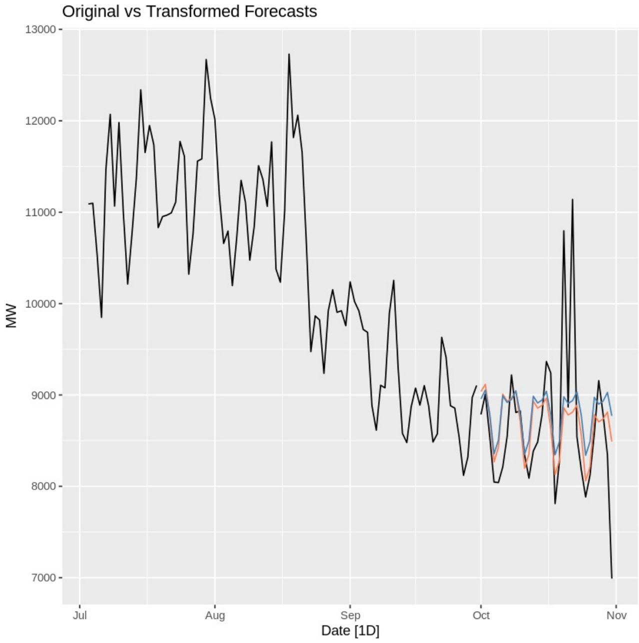
Model	RMSE	MAE
<chr>	<dbl>	<dbl>
Original	686.1925	443.6372
Transformed	709.6965	492.5329

In [ ]:
```r
fc_back_tsibble <- test_data %>%
  mutate(Transformed_Forecast = inv_box_cox(fc_transformed$.mean, lambda))
```

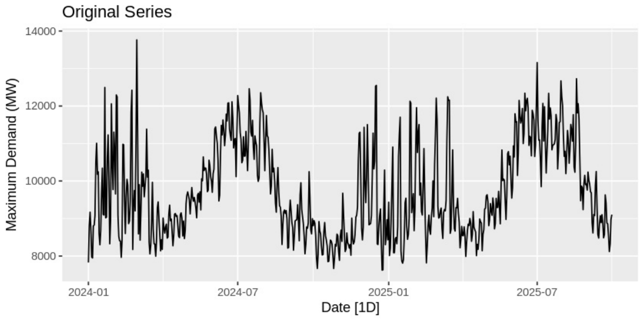
In [ ]:
```r
system("jupyter nbconvert --to html /content/MaximumDemand.ipynb")
```

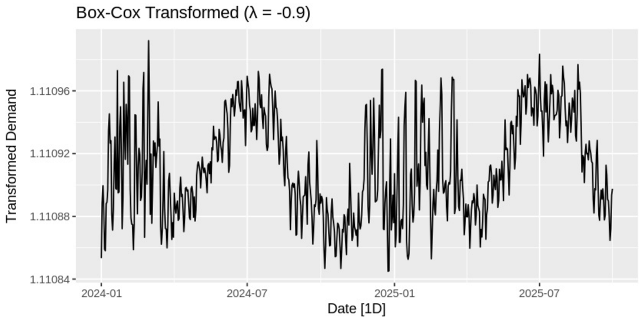
In [ ]:
```r
autoplot(train_data %>% tail(90)) +
  autolayer(test_data, color = "black") +
  autolayer(ets_manual_fc, level = NULL, color = "coral") +
  autolayer(fc_back_tsibble, Transformed_Forecast, color = "steelblue") +
  labs(title = "Original vs Transformed Forecasts", y = "MW")
```

```
Plot variable not specified, automatically selected `.vars = MaxDemand`
Plot variable not specified, automatically selected `.vars = MaxDemand`
```



In [ ]:
```r
# Visualize original vs transformed series
p1 <- train_data %>%
  autoplot(MaxDemand) +
  labs(title = "Original Series", y = "Maximum Demand (MW)")

p2 <- train_data %>%
  mutate(MaxDemand_BC = box_cox(MaxDemand, lambda_guerrero)) %>%
  autoplot(MaxDemand_BC) +
  labs(title = paste0("Box-Cox Transformed (λ = ", round(lambda_guerrero, 3), ")"),
       y = "Transformed Demand")

gridExtra::grid.arrange(p1, p2, ncol = 1)
```





In [ ]:

```
# Fit ARIMA on transformed data
arima_trans_manual <- train_transformed %>%
  model(ARIMA(MaxDemand_BC ~ pdq(0,1,1) + PDQ(0,1,1)))

report(arima_trans_manual)

arima_trans_auto <- train_transformed %>%
  model(ARIMA(MaxDemand_BC))

report(arima_trans_auto)
```

```
Series: MaxDemand_BC
Model: ARIMA(0,1,1)(0,1,1)[7]

Coefficients:
          ma1     sma1
      -0.1213  -0.9996
s.e.   0.0742   0.0195

sigma^2 estimated as 1.605e-08:  log likelihood=5952.25
AIC=-11898.5   AICc=-11898.46   BIC=-11885.16
Series: MaxDemand_BC
Model: ARIMA(1,1,2)(0,0,2)[7]

Coefficients:
         ar1      ma1      ma2    sma1    sma2
      0.2394  -0.4333  -0.4169  0.1511  0.2041
s.e.  0.0717   0.0661   0.0494  0.0400  0.0407

sigma^2 estimated as 2.264e-09:  log likelihood=6076.16
AIC=-12140.31   AICc=-12140.18   BIC=-12113.56
```

In [ ]:

```
library(purrr)
library(distributional)

# Forecast the BC ARIMA model
fc_trans_manual <- forecast(arima_trans_manual, h = nrow(test_data))

# Convert to tibble to avoid fable internal rebuild errors
fc_tib <- fc_trans_manual %>% as_tibble()

# Back-transform BC forecast and compute quantile intervals
fc_bt_manual <- fc_tib %>%
  mutate(
    mean_bt     = forecast::InvBoxCox(.mean, lambda),

    lower80_bt = forecast::InvBoxCox(map_dbl(MaxDemand_BC, ~ quantile(.x, 0.10)), lambda),
    upper80_bt = forecast::InvBoxCox(map_dbl(MaxDemand_BC, ~ quantile(.x, 0.90)), lambda),

    lower90_bt = forecast::InvBoxCox(map_dbl(MaxDemand_BC, ~ quantile(.x, 0.05)), lambda),
    upper90_bt = forecast::InvBoxCox(map_dbl(MaxDemand_BC, ~ quantile(.x, 0.95)), lambda)
  )

# Replace missing ribbon values
fc_bt_manual <- fc_bt_manual %>%
  mutate(
    lower80_bt = ifelse(is.na(lower80_bt), mean_bt, lower80_bt),
    upper80_bt = ifelse(is.na(upper80_bt), mean_bt, upper80_bt),
```

```
    lower90_bt = ifelse(is.na(lower90_bt), mean_bt, lower90_bt),
    upper90_bt = ifelse(is.na(upper90_bt), mean_bt, upper90_bt)
  )
```

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

In [ ]:
```
ggplot() +
  geom_line(data = train_data %>% tail(60),
            aes(Date, MaxDemand)) +
  geom_line(data = fc_bt_manual,
            aes(Date, mean_bt)) +
  geom_ribbon(
    data = fc_bt_manual,
    aes(Date, ymin = lower80_bt, ymax = upper80_bt),
    alpha = 0.25
  ) +
  labs(
    title = "Back-Transformed ARIMA Forecast (80% Interval)",
    x = "Date",
    y = "Max Demand (original units)"
  )
```