Bachelor's Project
Power Grid Load Forecasting using Machine Learning Approaches

Omar Abdesslem
University of Geneva

April 2025

Contents

1 Abstract

This project focuses on developing machine learning models for load forecasting in the Swiss energy grid, using historical data on energy consumption, production, and cross-border exchanges. The datasets include detailed information on total energy consumed and produced in the Swiss control block, grid feed-ins, net outflows, and energy trades with neighboring countries (Germany, France, Austria, and Italy). By leveraging this data along with weather and seasonal factors, the project aims to improve the accuracy of short-term and long-term load forecasts using advanced machine learning techniques such as LSTM, Transformers, and Gradient Boosting, while comparing their performance with traditional models.

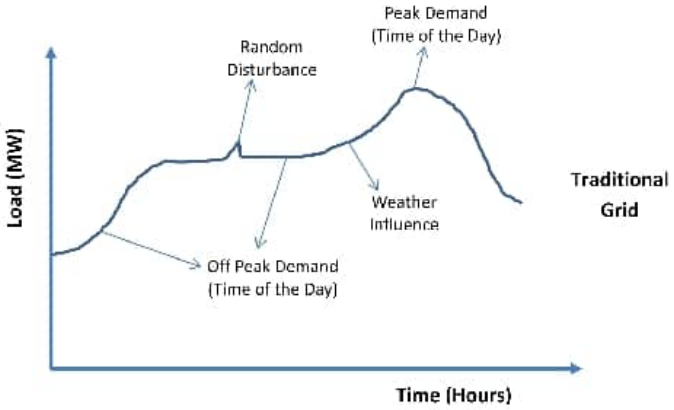


Figure 1: Load and influence

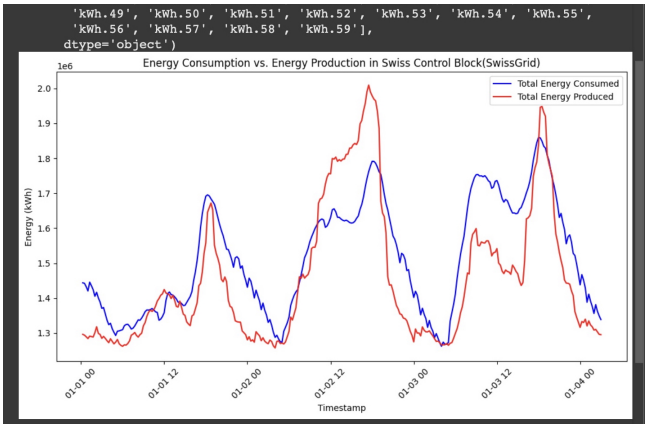


Figure 2: Plotted Energy Production visuals (Python3)

The goal of which is to combine Machine Learning, Data Structures, and Physics to predict real-life energy trends.

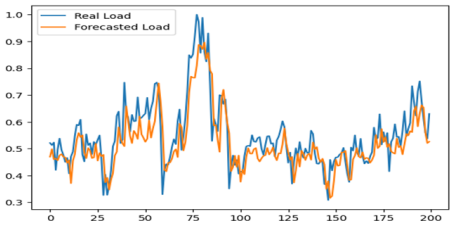


Figure 3: Endgoal

2 Objectives

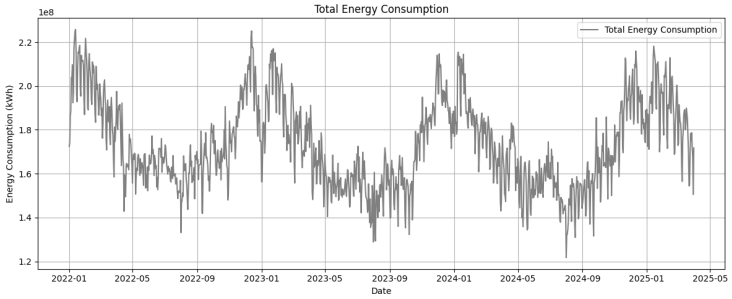
The primary objectives are:

1. Visualize the Data
2. Develop a Medium Term Forecasting Model (Predicting Total Amount of Energy Consumed per day/week)
3. Setting a Baseline Model and model evaluation Metric
4. Evaluate the results
5. Discussing challenges and conclusions

3 Visualisation

3.1 Variable to Predict

The target variable I'm trying to predict is the weekly energy load, specifically **Total Energy Production**. The original data is recorded at 15-minute intervals, and I aggregate it to weekly values using a 7-day window (Monday to Sunday), following EU forecasting standards. According to Swissgrid, this variable represents the total energy produced in the control block Switzerland, based on aggregated feed-in sequences reported by distribution network operators. It includes only production plants equipped with load profile meters.



l

Figure 4: Total Swiss Energy Consumption(2024)

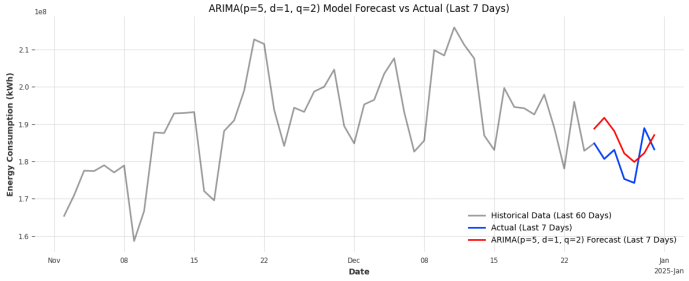


Figure 5: 1-week Forecasting example(ARIMA model)

3.2 Related Variables in the dataset

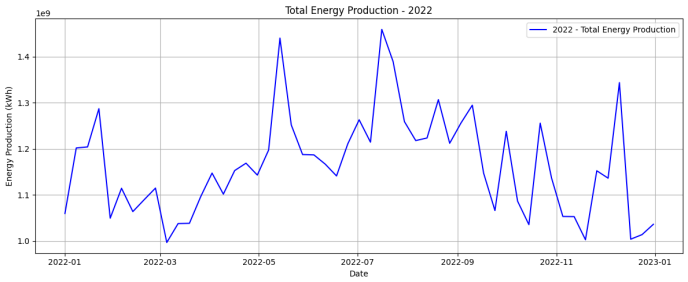


Figure 6: Total energy production over time

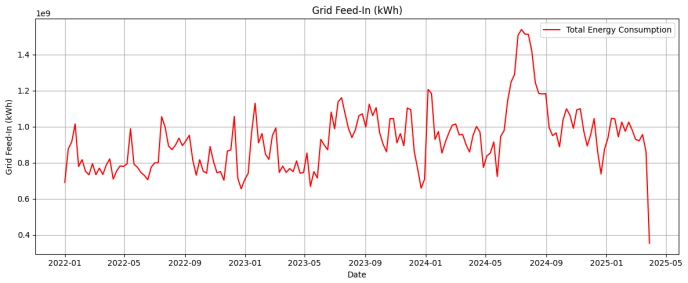


Figure 7: Electricity fed into the grid

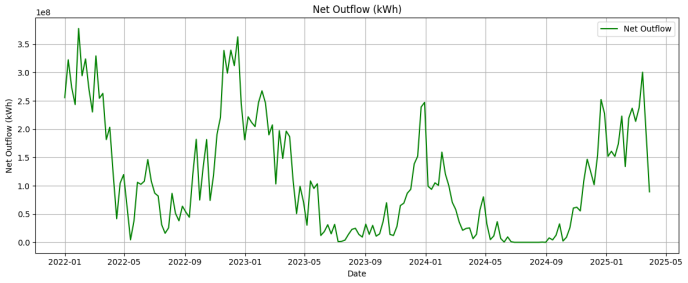


Figure 8: Net outflow of energy from the grid

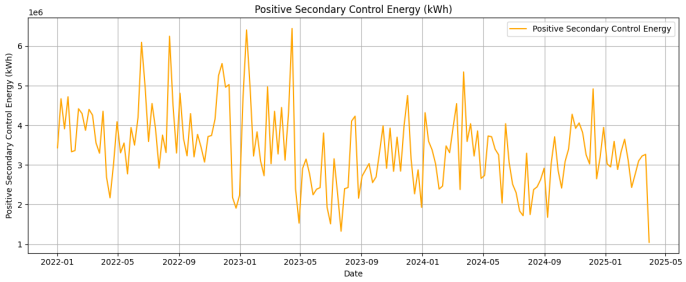


Figure 9: Activation of positive tertiary control reserves

4 Time Series Analysis

4.1 Data Cleaning

will be added in: for now: converted everything to numeric, replaced errors with NaN, removed NaN rows

4.2 Mathematical basics

4.2.1 Stationarity

Stationarity refers to the behavioral consistency of the time series. Mathematically, this means that the mean and covariance stay invariant regardless of the time shift.

Strict stationarity means that the mean, variance, and covariance are constant. Weak stationarity means that the mean, variance is constant, and the covariance function $\gamma(s,t)$ depends only on t - s, as in any two values depends only on the time difference between them, not on the actual time at which they occur. [4]

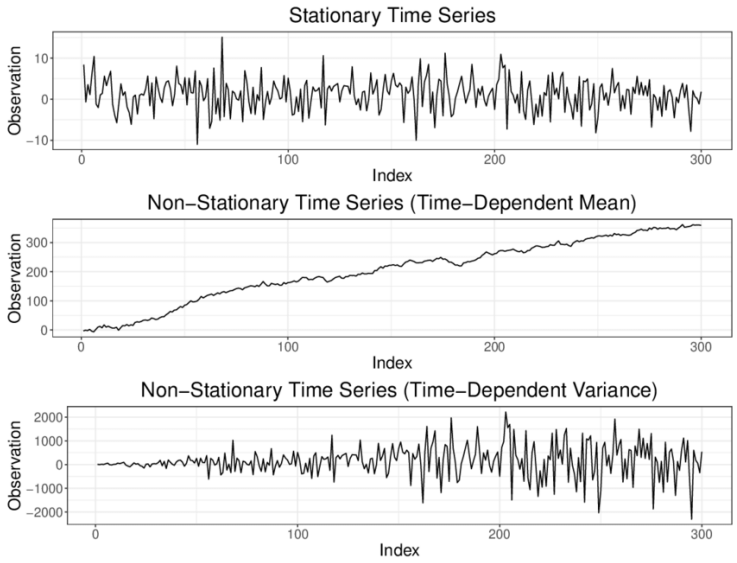


Figure 10: Stationary and non-stationary time series, Bauer 2021 [1]

4.2.2 White Noise

A stochastic process $\{Y_t\}$ is called *white noise* if all its elements are not correlated, with mean $\mathbb{E}(Y_t) = 0$ and constant variance $\text{Var}(Y_t) = \sigma^2$ [4].

The standard deviation, which reflects the dispersion of values around the mean, is further explained in [18].

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}$$

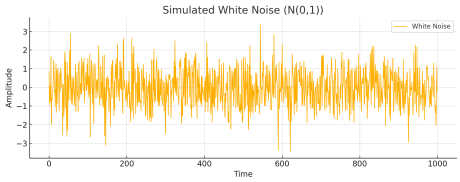


Figure 11: White noise using an np.random fct

As explained in Wikipedia [18], white noise is a stochastic process and does not have a deterministic function.

4.2.3 Random Walk

A time series $\{Y_t\}$ is called a *random walk* if it satisfies the relation

$$Y_t = Y_{t-1} + \varepsilon_t,$$

where ε_t is white noise. [4]

4.3 Initial data analysis

4.3.1 Mean

The mean, or expected value, of a time series quantifies its average level over time. Its formula is given by:

$$\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Where x_i are the observed values and n is the number of observations [2].

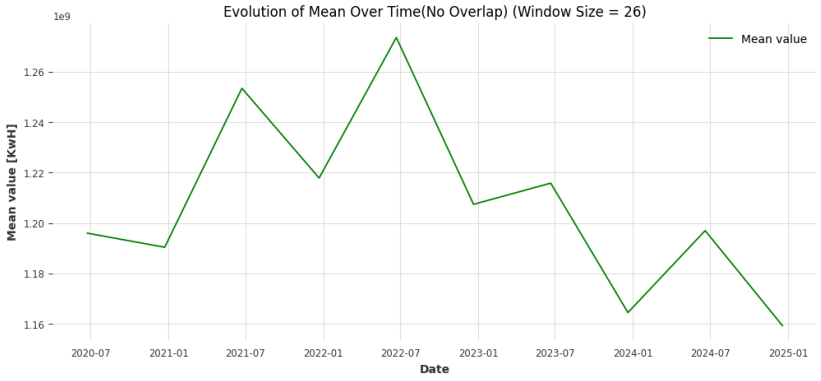


Figure 12: Evolution of Mean Over Time — 2024

In this case, the mean energy consumption shows a decreasing trend from February to September 2024, followed by a gradual increase into October. This might be due to seasonal changes (for example, warmer months might require less energy consumption than colder ones).

4.3.2 Variance

The variance measures the dispersion or spread of the data around the mean. Its formula is given by:

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2$$

Where μ is the mean and σ^2 is the variance [3].

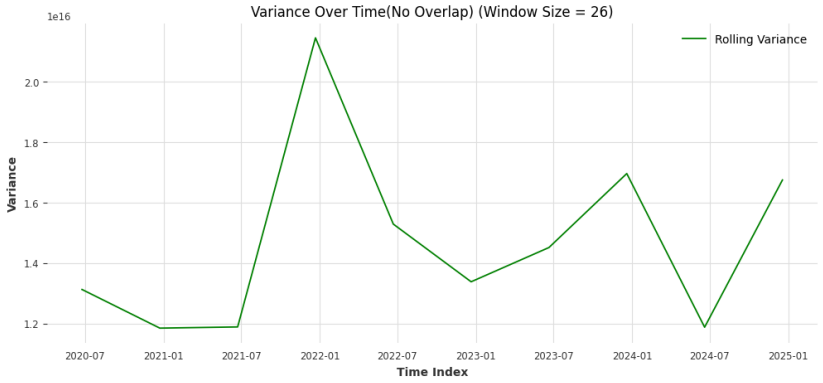


Figure 13: Evolution of Variance Over Time — Total Energy Consumption 2024

Observation: The variance fluctuates significantly over time.

4.3.3 Transformation

4.3.4 Moving Sum (Window)

The moving sumcalculate the sum of a fixed number of consecutive values (a "window") in a dataset. The Moving Sum formula is calculated by:

$$S_t = \sum_{i=0}^{n-1} x_{t-i}$$

In this case, To reduce noise and reveal trends, we will be using a 1 week window.

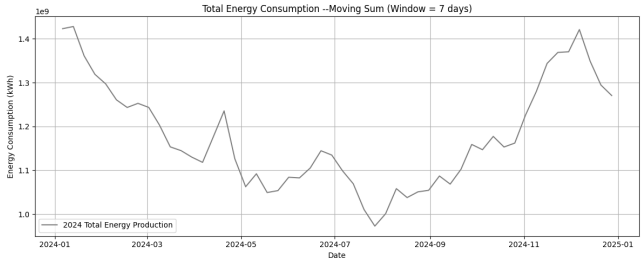


Figure 14: Weekly Sum

4.3.5 Moving Average

Averaging reduces variance, and introduces correlation in Yt [4]. A Simple Moving Average is calculated by the formula:

$$\mathrm{MA}_t = \frac{1}{n} \sum_{i=0}^{n-1} x_{t-i}$$

4.3.6 Outliers & Data Cleaning

An outlier is an observation that causes surprise relative to the rest of the data. It may be isolated or successive [4].

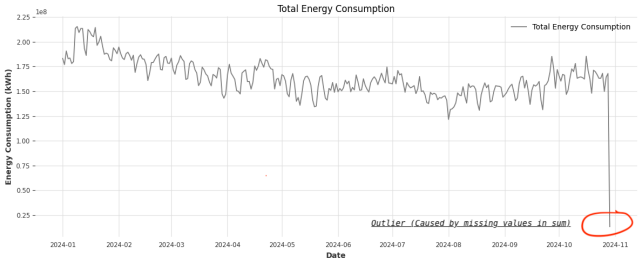


Figure 15: Outlier example in 2024 Consumption Graph

In the case for outliers, I will replace them with the value of the average in the Moving Average Window.

4.3.7 Trends

Trend is a pattern in data that shows the movement of a series to relatively higher or lower values over a long period of time [9]. Trends can be linear, quadratic, periodic, or more complex [4].

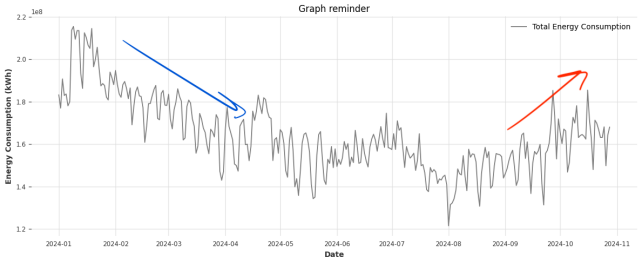


Figure 16: Trend Evolution in 2024 Consumption Graph

4.3.8 Seasonality

A repeating pattern that occurs at fixed and regular intervals (e.g. daily, weekly, yearly) [4]. Seasonality is a predictable cyclical pattern, whereas trends are a long-term change in data (increase/decrease).

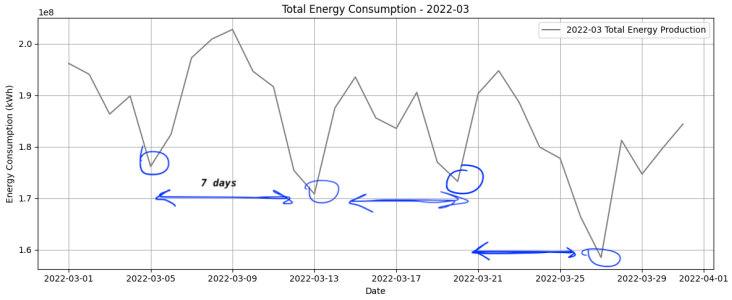


Figure 17: Weekly Seasonality, Month of March 2022

4.4 Differencing

4.4.1 Motivation

Differencing is a simple approach to removing trends. No need to estimate parameters. [4].

4.4.2 Differencing types

Differecing can be of *first-order* or *higher-order* [4]

4.4.3 First-order difference

The first order difference is defined as [4]:

$$\Delta Y_t = Y_t - Y_{t-1}$$

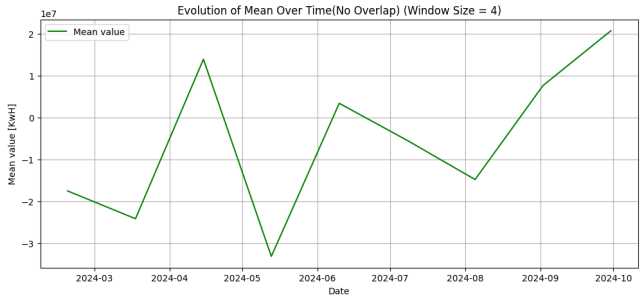


Figure 19: Mean Over Time after Differencing

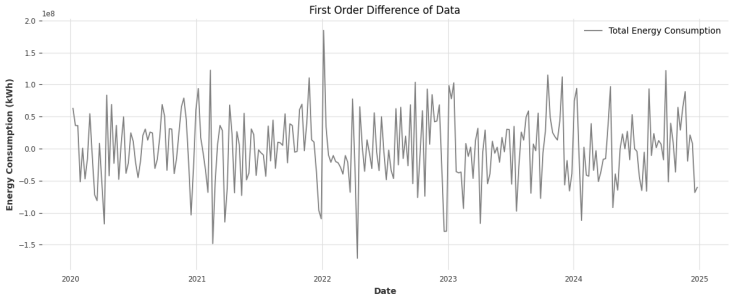


Figure 18: First Order Difference Weekly, 2024

Observation: The resulting time series does not look much different.

4.4.4 Higher-order difference

If one round of differencing is not sufficient to achieve stationarity, a *higher-order difference* can be applied, the second-order difference is [4]:

$$\begin{aligned}\Delta^2 Y_t &= \Delta(\Delta Y_t) = \Delta(Y_t - Y_{t-1}) \\ &= \Delta Y_t - \Delta Y_{t-1} = Y_t - 2Y_{t-1} + Y_{t-2}\end{aligned}$$

First-order differencing reduces a random walk to stationarity. In practice, we difference until plots of the differenced data appear stationary; often k= 1,2 suffices [4].

Figure 20: Second Order Difference Weekly, 2024

4.4.5 Seasonal differencing

It's simply (Yt - Yt - s), with s being the seasonality.

4.5 ACF and PACF

4.5.1 Motivation

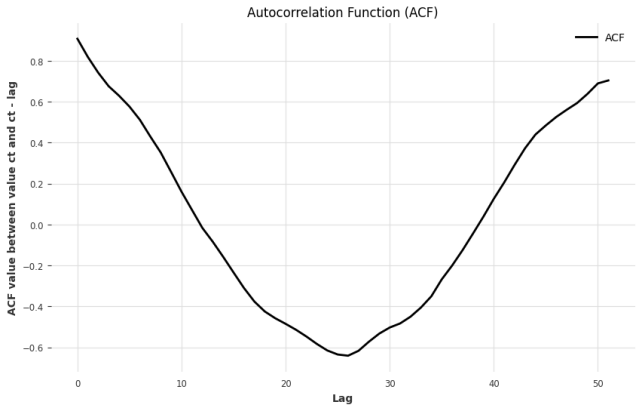
Autocorrelation and partial autocorrelation functions are used to understand the dependence structure of a time series. They help identify appropriate models [4].

4.5.2 Correlogram

The covariance function for equally spaced data $y_1, \ldots, y_n$ is defined as:

$$c_h = \frac{1}{n-h-1} \sum_{i=1}^{n-h} (y_i - \bar{y})(y_{i+h} - \bar{y}), \quad h = 0, 1, \ldots, n-2,$$

where $\bar{y}$ is the sample mean. The correlogram (ACF) is a graph of $\hat{\rho}_h = \frac{c_h}{c_0}$ against lag h [4].



1

Figure 21: AutoCorrelation Function 2024 - Total Energy Consumption

4.5.3 Partial correlogram

The PACF is interpreted in a similar way to the ACF, but it reveals the **direct** relationship between an observation and its lagged values, controlling for the values in between. Let $Y_0, \ldots, Y_h$ be successive observations. The partial autocorrelation function (PACF) at lag h measures the correlation between Y_t and Y_{t-h} after removing the linear influence of intermediate lags $Y_{t-1}, \ldots, Y_{t-h+1}$.

$$\tilde{\rho}_1 = \text{corr}(Y_1, Y_0)$$

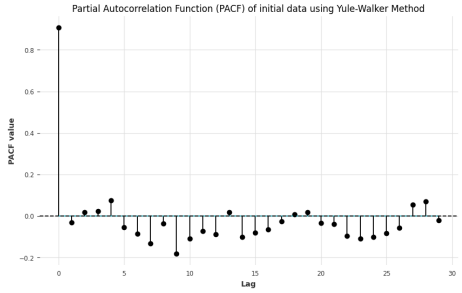


Figure 22: PACF computed using the Yule-Walker method

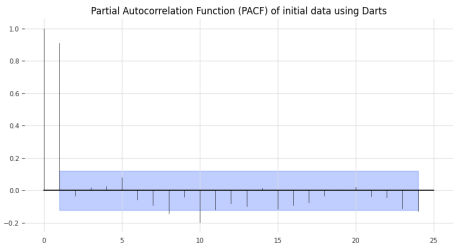


Figure 23: PACF generated using Darts

4.5.4 Testing for stationarity

To build reliable model, we need to check whether the data is stationary. One way to test this is to decompose the time series Y_t into three components:

$$Y_t = \xi_t + \eta_t + \varepsilon_t$$

[4]

Where:

- ξ_t is the deterministic trend, as in a fixed, predictable pattern over time
- η_t is a supposed random walk,
- ε_t is noise

Types of stationarity:

- **Level stationarity**: If $\sigma_u^2 = 0$ and $\xi_t = 0$, then Y_t is stationary around a constant mean.
- **Trend stationarity**: If $\sigma_u^2 = 0$ and $\xi_t = \beta t$, then Y_t becomes stationary after removing the trend.

KPSS Test: The KPSS test is used to test the null hypothesis that a time series is stationary. It does this by estimating the test statistic:

$$C(l) = \frac{1}{\sigma^2(l)} \sum_{t=1}^{n} S_t^2, \quad \text{where } S_t = \sum_{j=1}^{t} e_j$$

Here, $e_1, \ldots, e_n$ are the residuals from regressing Y_t on a constant or a linear trend (depending on whether testing for level or trend stationarity), and $\sigma^2(l)$ is a long-run variance estimate using a truncation lag l. [4] The test is interpreted as follows:

- If the test statistic is small (below the critical value), we **do not reject** the null hypothesis: the series is stationary.

- If the test statistic is large (above the critical value), we **reject** the null hypothesis: the series likely contains a unit root and is non-stationary.

```
    is_stationary = stationarity_test_kpss(initial_series)
    stat, p_value, lags, crit_vals = stationarity_test_kpss(initial_series)
    print(f"KPSS statistic: {stat}"), print(f"p-value: {p_value}"), print(f
✓ 0.0s

KPSS statistic: 0.09144255304652452
p-value: 0.1
Is stationary: True
```

Figure 24: KPSS result for differenced data

In this case, the KPSS test returns True for it is stationnary. This means that our time series is indeed stationnary.

However, running the ADF test returns false for stationnarity. This conflicting evidence will lead me to not rule out ARIMAs (for the time being).

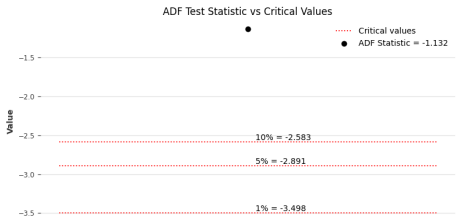


Figure 25: ADF test

4.5.5 Testing for white noise

There are many methods to test for white noise. One of which, documented in the Time Series Analysis book [4], is the Ljung–Box.

For a time series $y_1, \ldots, y_n$, the Ljung–Box test statistic is:

$$Q_m = n(n+2)\sum_{h=1}^{m}\frac{\hat{\rho}_h^2}{n-h}$$

Where: - $\hat{\rho}_h$ is the autocorrelation of the sample at lag h - n is the length of the series - m is the maximum lag to include in the test [4]

We shall use the calculated autocovariances to create the Ljung-Box function using the formula above. The ACFs were calculated using biased covariances.

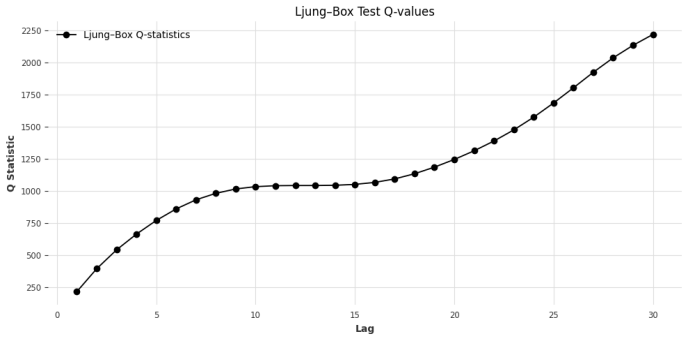


Figure 26: Ljung-Box Q values

Chi-squared distribution. Under the null hypothesis that the series is white noise, the Ljung–Box statistic Q_m follows a Chi-squared distribution with m degrees of freedom:

$$Q_m \sim \chi^2_m$$

The Chi-squared distribution is a continuous probability distribution. Its formula is the following for the squares of k independent standard normal variables:

$$\chi^2_k = Z_1^2 + Z_2^2 + \cdots + Z_k^2, \quad \text{where } Z_i \sim \mathcal{N}(0,1)$$

Its probability density function is given by:

$$f(x;k) = \frac{1}{2^{k/2}\Gamma(k/2)} x^{(k/2)-1} e^{-x/2}, \quad x > 0$$

where Γ is the gamma function, as in the non-integer and integer factorial calculator. The distribution is positively skewed, and its shape depends on the degrees of freedom k.

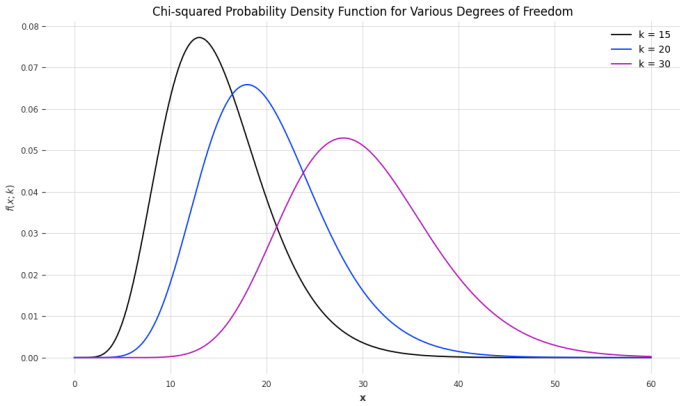


Figure 27: Enter Caption

In the Ljung–Box test, if the observed Q_m is greater than the critical value from the χ^2_m distribution at a given significance level (e.g., 5%), we reject the null hypothesis and conclude that the time series is not white noise.

Then, the p-value is:

$$\text{p-value} = P(\chi^2_m \geq Q_m) = 1 - F(Q_m)$$

Where:

$F(Q_m)$ is the distribution function of the Chi-squared distribution.

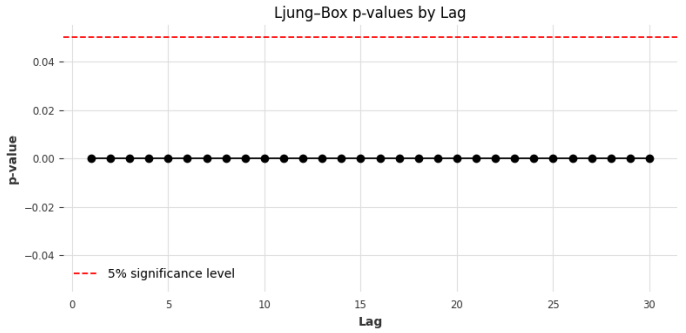


Figure 28: Ljung-Box p values by lag

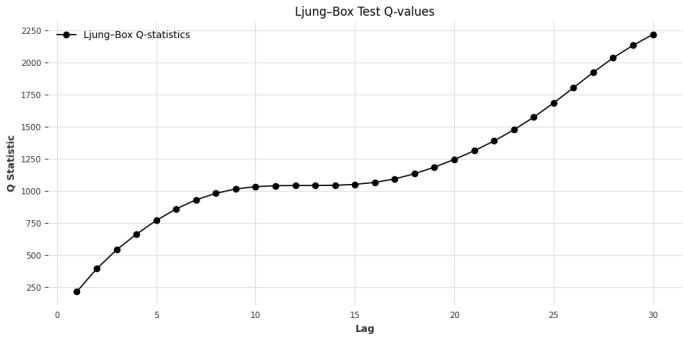


Figure 29: Ljung-Box Q values

4.5.6 Checking normality using QQ plots

We often need to compare data $y_1, \ldots, y_n$ with a given distribution F, usually the normal distribution (for example, to check if the standardized residuals are $\mathcal{N}(0,1)$).

A quantile-quantile (Q–Q) plot is a graph of the ordered values of the y_j:

$$y_{(1)} \leq y_{(2)} \leq \cdots \leq y_{(n)}$$

against theoretical quantiles of F, given by $x_i = F^{-1}\left(\frac{i}{n+1}\right)$: we plot pairs

$$(x_1, y_{(1)}), (x_2, y_{(2)}), \ldots, (x_n, y_{(n)})$$

It is best if the plot is square and if it includes confidence levels (often 95%). Properties:

- perfect linearity shows perfect fit of F to the data, while strong curvature suggests poor fit;
- outliers show as extreme values lying well off the line of the other data;
- for standard normal Q–Q plots we use $x_i = \Phi^{-1}\left(\frac{i}{n+1}\right)$, where Φ is the $\mathcal{N}(0,1)$ distribution function.

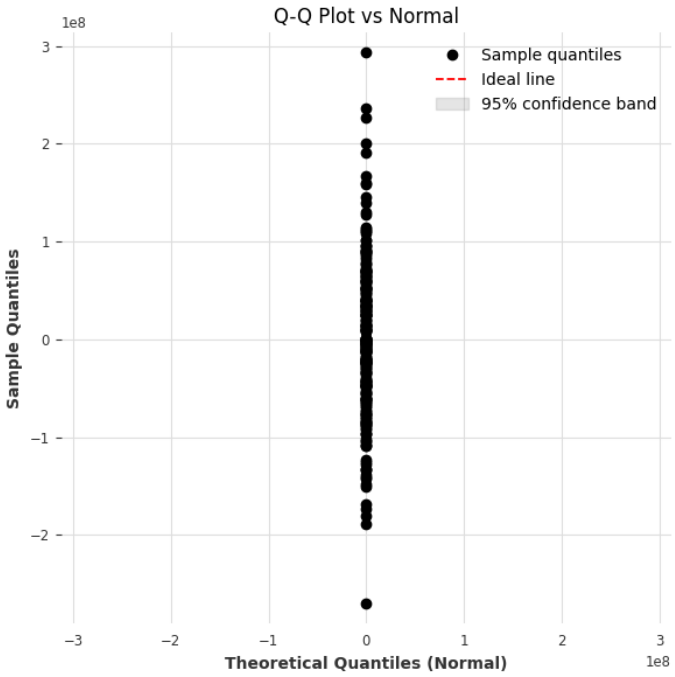


Figure 30: Q-Q Plot initial data

Standardization The graph is too close together, to have a clearere view , we must standarize the data before generating the Q–Q plot. we do so using:

$$z_i = \frac{x_i - \bar{x}}{s}$$

where $\bar{x}$ and s are the sample mean and standard deviation, respectively.

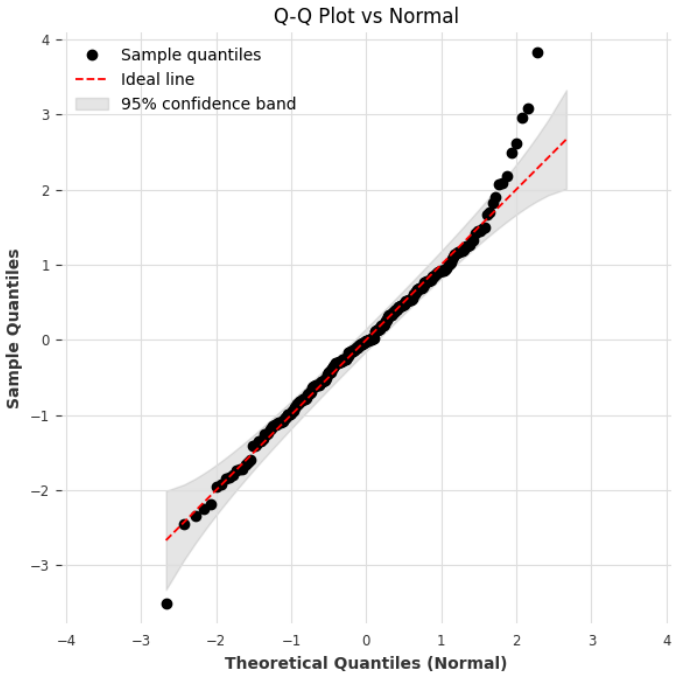


Figure 31: Q-Q Plot vs Normal

4.6 Periodogram

4.6.1 Motivation

Many series have cyclic structure (e.g. sunspots, CO2 data,...), but we may not know what the cycles are in advance of looking at the data. The periodogram is a summary description based on representing the observed series as a superposition of sine and cosine waves of various frequencies. [4]

Check si il y a plusieurs fréquences, lequels, c'est quoi leurs amplitudes, check residual stat

4.6.2 Discrete Fourier transform

We can avoid the previous regression and use the discrete Fourier transform (DFT) for frequency analysis of time series.

The discrete Fourier transform of a time series $y_1, \ldots, y_n$ is the complex-valued series

$$d(\omega_j) = \frac{1}{\sqrt{n}} \sum_{t=1}^{n} y_t e^{-2\pi i \omega_j t}$$

$$d(\omega_j) = \frac{1}{\sqrt{n}} \left(\sum_{t=1}^{n} y_t \cos(2\pi\omega_j t) - i \sum_{t=1}^{n} y_t \sin(2\pi\omega_j t) \right)$$

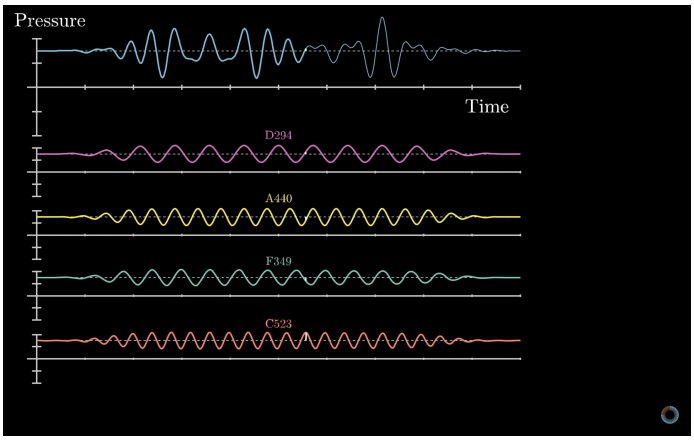


Figure 32: Signal decomposition using Fourier Transform (3Blue1Brown) [13]

We define the periodogram $I(\omega_j) = |d(\omega_j)|^2$.
The periodogram is related to the scaled periodogram: $I(\omega_j) = \frac{n}{4} P(\omega_j)$.

4.6.3 Periodogram

(a) If $y_1, \ldots, y_n$ is an equally-spaced time series, its periodogram ordinate for ω is defined as

$$I(\omega) = |d(\omega_j)|^2$$

this means that:

$$I(\omega) = \frac{1}{n} \left[\left(\sum_{t=1}^{n} y_t \cos(2\pi\omega t) \right)^2 + \left(\sum_{t=1}^{n} y_t \sin(2\pi\omega t) \right)^2 \right], \quad 0 < \omega \leq \frac{1}{2}$$

Our plot for the linear periodogram:

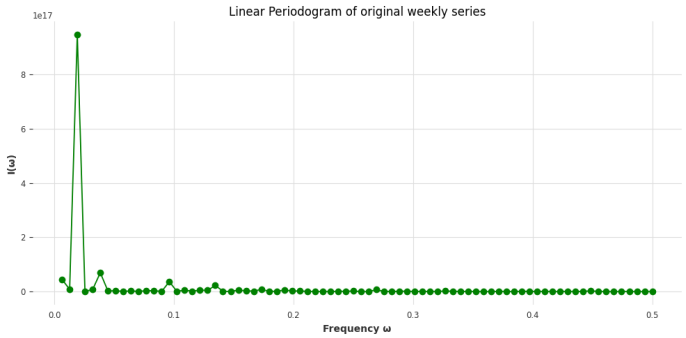


Figure 33: Linear Periodogram of original weekly series

4.6.4 Spectral Analysis— Power Spectrum

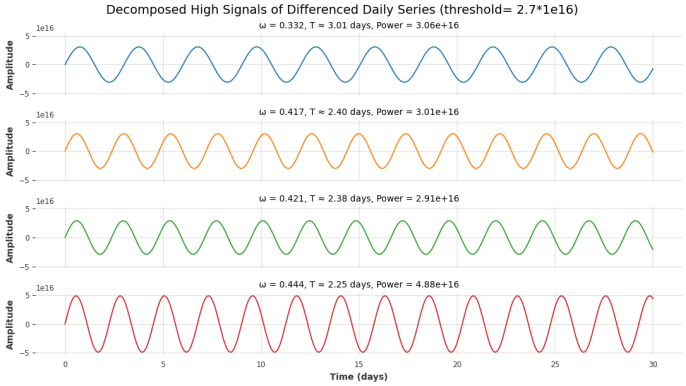


Figure 34: Decomposed Periodogram showing High Signals of Differenced Daily Series

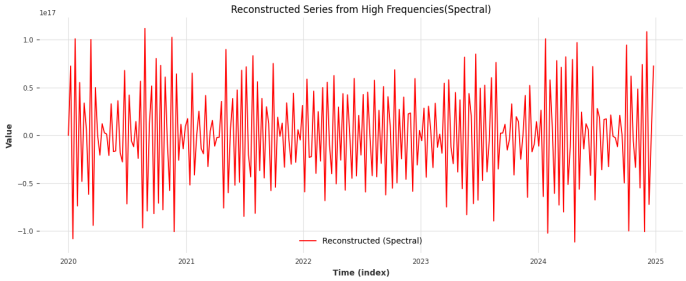


Figure 35: Spectral Reconstructed Series from High Frequencies(Spectral)

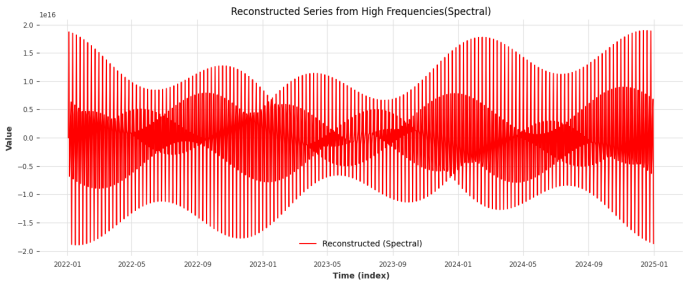


Figure 36: Reconstructed Frequencies Series for Daily values data

This visual was created from a previous project version. Added for visual appreciation.

4.6.5 Cumulative periodogram

(c) The cumulative periodogram

$$C_r = \frac{\sum_{j=1}^{r} I(\omega_j)}{\sum_{l=1}^{m} I(\omega_l)}, \quad r = 1, \ldots, m$$

is a plot of $C_1, \ldots, C_m$ against the frequencies ω_j for $j = 1, \ldots, m$. [4]

According to Davidson, Gaussian and non-Gaussian white noise has a flat spectrum [4]

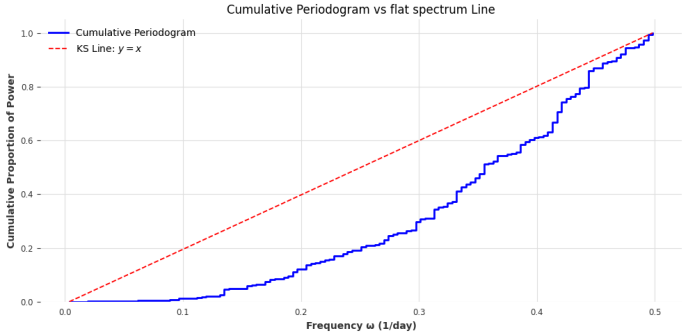


Figure 37: Cumulative Periodogram vs flat spectrum Line

4.6.6 Interpretation/Is this brownian noise

In the figure, the cumulative periodogram does not follow the red line (which represents white noise). This means, the frequencies in the time series are not evenly spread out. My weekly time series is then not white noise.

Spectral analysis of variance

Might be added if time allows

4.7 Smoothing

Smoothing data set is to create an approximating function that attempts to capture important patterns in the data, while leaving out noise or other fine-scale structures. [17]

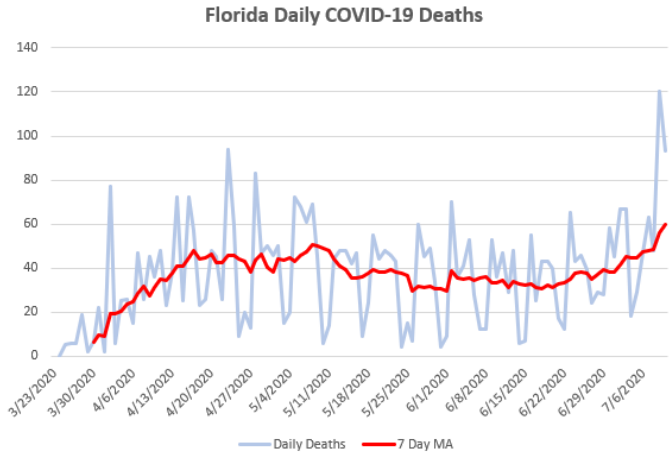


Figure 38: Smoothing Example - COVID Deaths 2019 (statisticsbyjim.com) [8]

4.7.1 Motivation

According to Boris's book, the underlying model is

$$Y_t = \mu(t) + Z_t,$$

[4] where $\mu(t)$ is smooth function of t and $\{Z_t\}$ is stationary. Among other things, smoothing can identify trends and seasonality. Differencing can remove trend to give stationary series. But differencing does not allow us to visualise the trend. [4]

We can implement smoothing to examine/estimate the trend, for example using:

- moving average (simple, related to differencing);
- polynomial (simple, doesn't work very well);
- local polynomial (simple, easy to robustify);
- STL decomposition (robust fitting of local polynomial, with seasonal effects). [4]

We will see later that using differencing results in large uncertainties in predictions. Intuitively, this is because differencing can remove very random trends which must be taken into account for later predictions of Y_{t+h}. If we can estimate the trend $\mu(t)$ accurately and predict it with low uncertainty, we can obtain better forecasts than when using differencing. [4]

4.7.2 Moving averages

Simple Moving Average

Moving Averages is one of the simplest smoothing methods to implement, it essentially computes the local average function of a window size = n.

Simple Moving Average formula is given by the formula:

$$\text{SMA}_t = \frac{1}{N} \sum_{i=0}^{N-1} y_{t-i}$$

[16]

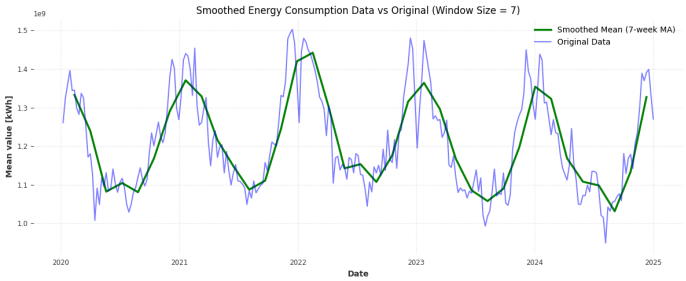


Figure 39: Smoothed 7 MA Energy Consumption Data vs Original

Weighted Moving average

Weighted Moving average is given by the formula :

$$s_t = \sum_{j=-p}^{p} w_j y_{t+j}, \quad t = p+1, \ldots, n-p, \quad p \in \mathbb{N},$$

[4]

Classical approaches to smoothing aim to reduce short-term fluctuations in time series data. Given a sequence of observations $y_1, \ldots, y_n$, one common method is to replace each value y_t with the average of its immediate neighbors:

$$y'_t = \frac{1}{3}(y_{t-1} + y_t + y_{t+1})$$

More generally, this method constructs a moving average of order $2p+1$, using weights w_j that satisfy

$$\sum_{j=-p}^{p} w_j = 1, \quad \text{with usually } w_j > 0 \text{ and } w_j = w_{-j}.$$

This is an example of a linear filter, where each smoothed value is a weighted sum of surrounding observations.

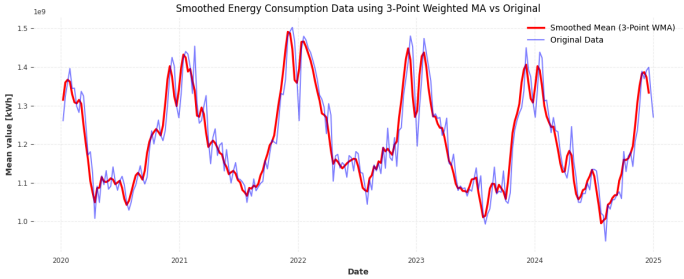


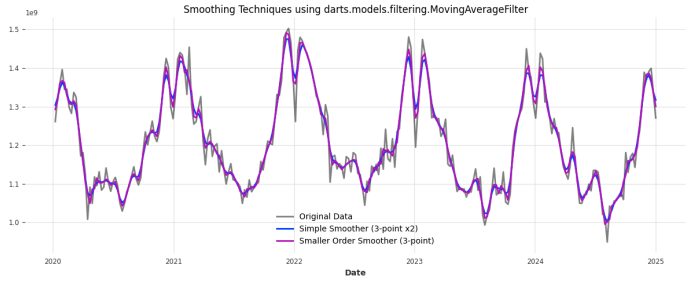
Figure 40: Smoothed Energy Consumption Data using 3-Point Weighted MA vs Original

[4]

Fixing Weights

Fixes are possible near the ends, but usually $p \ll n$, so the details of the fixes are unimportant. Choose weights by:

- iterating simple (equally-weighted) smoothers;
- choosing higher order to remove (or at least decrease) seasonality, for example taking $p = 6$, $w_6 = w_{-6} = 1/24$ and all other $w_j = 1/12$;
- taking smaller order to highlight seasonality. [4]



xn

Figure 41: Higher Order Smoother, weights 1/24 for w1, w2p+1, 1/12 for middle weights

4.7.3 Local polynomial regression

A gloabl fit polynomial of degree k to the data is given by the formula

$$Y_t = p(t) + \varepsilon_t = \beta_0 + \beta_1 t + \cdots + \beta_k t^k + Z_t,$$

where $\{Z_t\}$ is stationary series. Choose parameters. [4] $\beta_0, \ldots, \beta_k$ to minimise the sum of squares

$$\sum_{t=1}^{n} \{y_t - p(t)\}^2 = \sum_{t=1}^{n} \left[y_t - \left(\beta_0 + \beta_1 t + \cdots + \beta_k t^k\right)\right]^2,$$

[4]

Instead of fitting one global curve , we fit small polynomial curves locally, near each time point. We give more weight to nearby points and less weight to far-away ones using a "Kernel function" that assigns these weights, where we pick a time point, and assign weights to nearby observations using the kernel function, and repeat for the next time point.

Automatic choice of h (or equivalent degrees of freedom $\equiv$ degree of polynomial) for kernel tends to be too small, owing to autocorrelation of time series. [4]

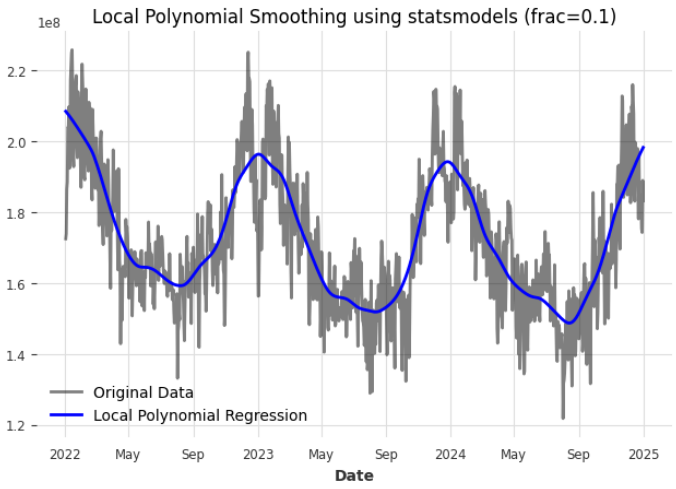


Figure 42: Local polynomial regression using statsmodels (p=0.25)

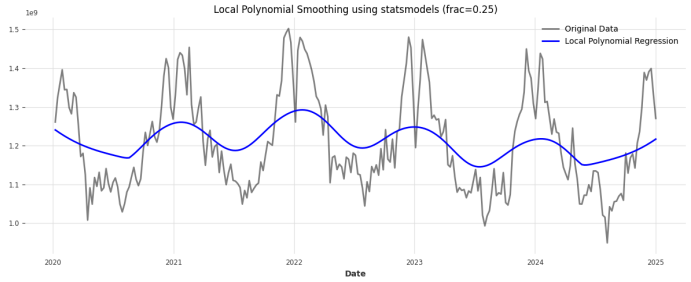


Figure 43: Local polynomial regression using statsmodels (p=0.25)

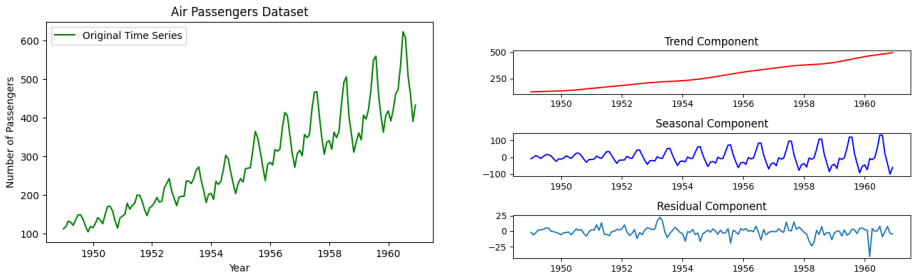
4.7.4 STL decomposition

An approach to removing overall trend and seasonal components, robust and (in principle) capable of handling missing data. [4]

The underlying model is:

$$Y_t = U(t) + S(t) + Z_t, \quad \{Z_t\} \text{ stationary},$$

where $U(t)$ is the trend component and $S(t)$ is the seasonal variation. [4]



(a) Original data [10]

(b) STL decomposition [10]

Figure 44: Example STL decomposition: (geeksforgeeks) [10]

Method: STL decomposition is based on the local polynomial regression. The seasonal component is found by smoothing the seasonal sub-series. This seasonal component is then removed from the initial data, and the remainder is smoothed to estimate the trend component. [4]

STL can fit either a single seasonal component or a slowly-varying one. There are several parameters to be chosen when using STL. The default values in the `stl` function are not always appropriate. [4]

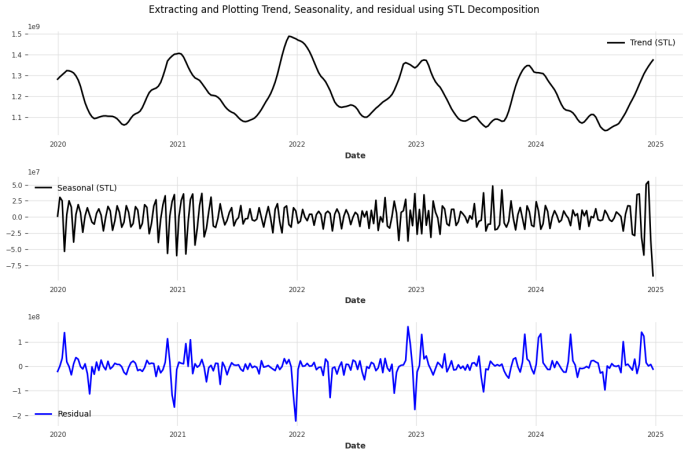


Figure 45: STL extracting trend seasonality

4.7.5 Additive vs Multiplicative Extraction Models

According to Mulius' article [12], in an additive model, the observed value is the sum of trend, seasonality, and residual components:

$$Y(t) = \text{Trend}(t) + \text{Seasonality}(t) + \text{Residual}(t).$$

This works best when the magnitude of seasonal fluctuations or residuals remains constant over time, regardless of the trend's growth or decline.

A multiplicative model represents the observed value as the product of its components:

$$Y(t) = \text{Trend}(t) \times \text{Seasonality}(t) \times \text{Residual}(t).$$

This is suitable when seasonal or residual effects scale with the trend [12].

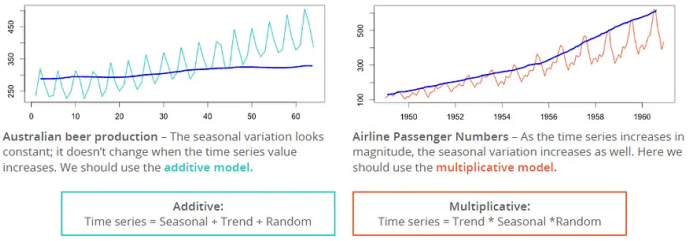


Figure 46: decomposition additive vs multiplicative (Nachi Keta) [11]

5 Time Series Interpretation

The goal of all this work is to analyze the weekly energy consumption data and understand its behavior.

Stationarity

To check if the data was stable over time, I used two tests: KPSS and ADF. KPSS said the data is already stationary, so I didn't apply differencing and kept $d = 0$ for ARIMA models. I did try differencing but it didn't make a difference.

ACF and PACF, Periodogram

The ACF and PACF plots helped identify the structure of the series. PACF dropped after lag 1, and ACF slowly faded, which suggested that an AR(1) model could work well (but maybe not for long term forecasting). I tested AR models from order 1 to 5. AIC and BIC scores were lowest for AR(1), meaning it was the best among those. This was also the case for MAPE scores.

The cumulative periodogram showed that my time series is not white noise.

Seasonality

STL decomposition didn't show clear seasonality. But the periodogram showed a strong signal every 52 weeks. So seasonal models like SARIMA with $s = 52$ make sense.

From my current undertanding, the series is stationary and has a repeating pattern every 52 weeks. AR(1) is a solid baseline, while an ARMA(5,2) might be better for longer term forecasting, and a SARIMAX should be the best model when including external data like temperature. I'll be testing them soon.

6 Models

6.1 Baseline Model

The baseline model employed is the *Naïve Drift model*. Unlike the standard naïve model (Yt+1 = Yt), which assumes no change from the last observed value, the drift model linearly extrapolates future values using the trend observed in the training window. Specifically, it forecasts the value at time $T + h$ as:

$$\hat{y}_{T+h} = y_T + h \cdot \frac{y_T - y_1}{T - 1}$$

where y_T is the last observed value in the training set, y_1 is the first, T is the length of the training window, and h is the forecast horizon. This model serves as a simple benchmark for evaluating the performance of more sophisticated forecasting methods.

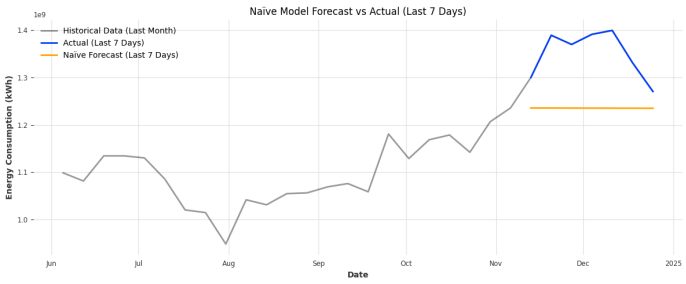


Figure 47: Baseline Naive model, 1 week daily forecast

6.2 AR

6.2.1 Definition

The first-order autoregressive, AR(1), process is a stationary process $\{Y_t\}$ satisfying

$$Y_t = \alpha Y_{t-1} + \varepsilon_t, \quad t \in \mathbb{Z}, \tag{1}$$

where α is the autoregressive parameter and $\{\varepsilon_t\}$ is white noise. The AR(1) process with mean μ is defined by

$$Y_t - \mu = \alpha(Y_{t-1} - \mu) + \varepsilon_t, \quad t \in \mathbb{Z}. \tag{2}$$

In theoretical discussion, we use (1), but in practice we must usually use (2) and estimate the mean μ.[4]

6.2.2 Plot

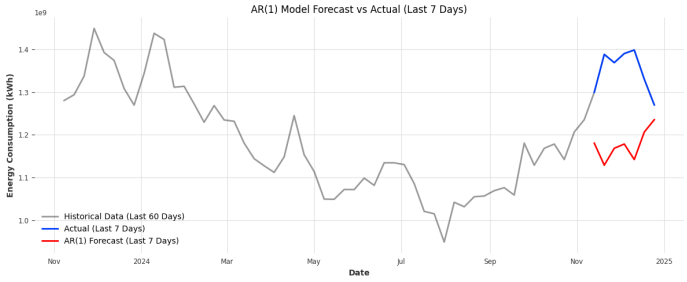


Figure 48: AR(1,0,0) model, 1 week daily forecast

6.2.3 Likelihood ratio test

6.2.4 Model comparison

According to Davison's book [4], a model $f_A(y)$ is nested within a model $f_B(y)$ if B may be reduced to A by restricting certain of the parameters.

- for example, a model $Y_1, \ldots, Y_n \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2)$ is nested within the model that the observations are from a Gaussian AR(1) process, because the first is obtained from the second by setting $\alpha = 0$.

Obviously the maximised log likelihoods satisfy $\ell_B \geq \ell_A$, because the more comprehensive model B contains the simpler model A.

The likelihood ratio statistic for comparing A with B is

$$W = 2(\ell_B - \ell_A).$$

If the model is regular, the simpler model is true, and B has q more parameters than A, then

$$W \dot{\sim} \chi^2_q.$$

```
Likelihood Ratio Test (LRT) Results (comparing AR(p) vs AR(p-1)):
AR(2) vs AR(1):  LR stat = 51.173, p-value = 0.0000
AR(3) vs AR(2):  LR stat = 37.705, p-value = 0.0000
AR(4) vs AR(3):  LR stat = 37.527, p-value = 0.0000
```

Figure 49: Likelihood-Ratio Test results for AR

6.2.5 Residuals

Standardized residuals are defined as

$$\tilde{e}_t = \frac{(y_t - \mu_t) - \alpha(y_{t-1} - \mu_{t-1})}{\sigma}, \quad t = 2, \ldots, n,$$

where $\mu_t = \mu + \delta I(t > 38)$.

The residuals should be approximately (Gaussian) white noise.
In the next slide we plot some diagnostics based on $\tilde{e}_2, \ldots, \tilde{e}_n$:

- original data with fitted mean,

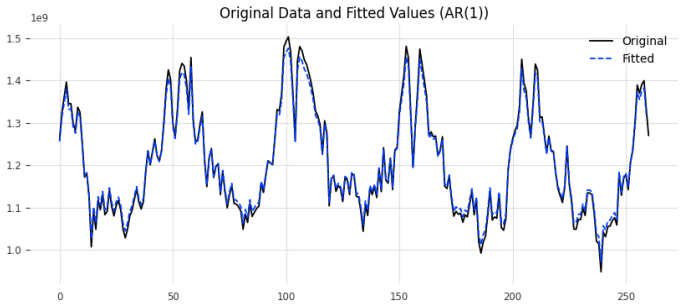


Figure 50: original data with fitted mean, AR model

- time series of residuals,

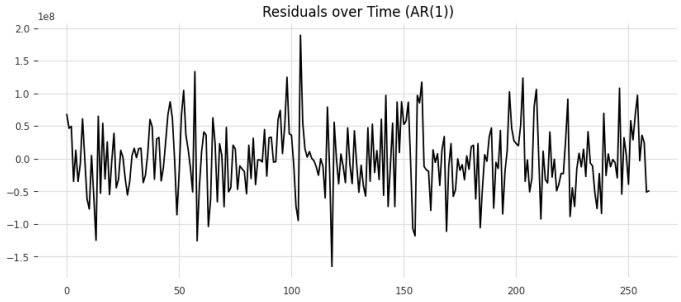


Figure 51: Residuals over Time(AR1)

- ACF,

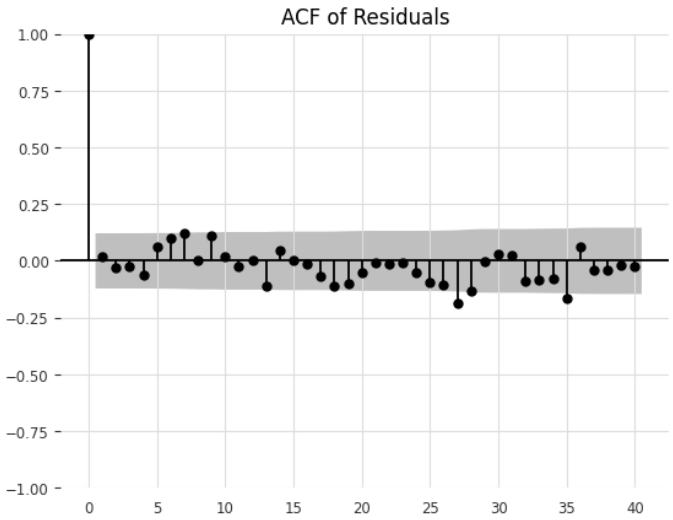


Figure 52: ACF of Residuals(AR(1))

- Ljung–Box test,

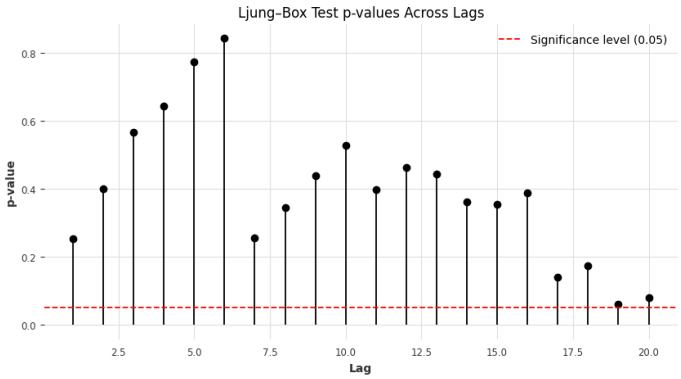


Figure 53: Ljung–Box test

- cumulative periodogram,

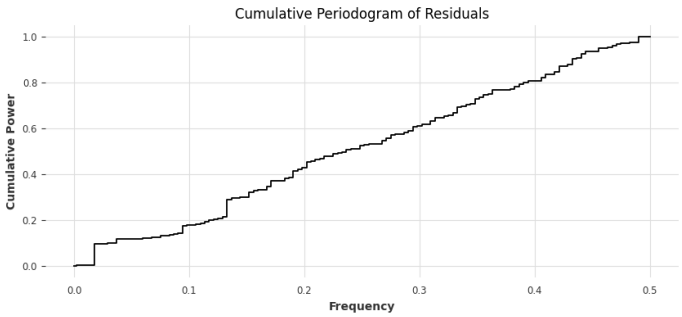


Figure 54: cumulative periodogram

- normal Q–Q plot.

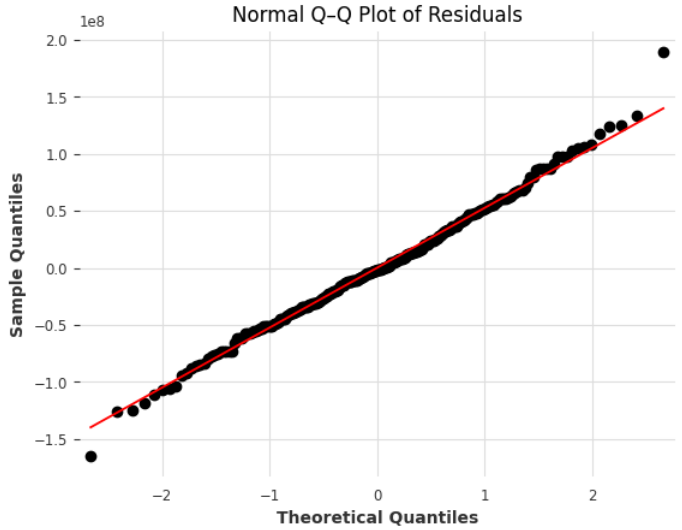


Figure 55: normal Q–Q plot

6.3 ARMA

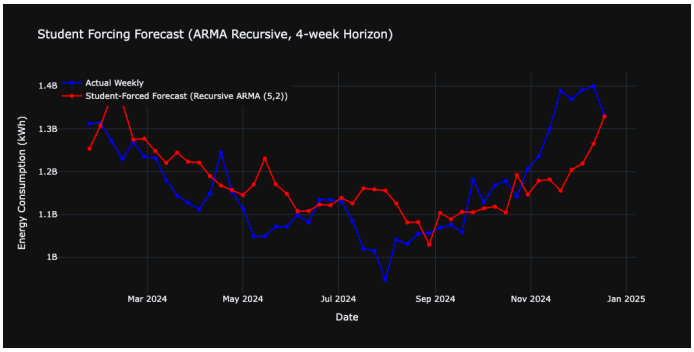


Figure 56: Recursive ARMA over 4-week horizon

6.3.1 ACF & PACF

According to the Time Series Analyis book, To summarise: for causal and invertible ARMA models the ACF and PACF have the following properties:

	AR(p)	**MA(q)**	**ARMA(p,q)**
ACF	Tails off	Cuts off after lag q	Tails off
PACF	Cuts off after lag p	Tails off	Tails off

This gives an approach to identifying AR and MA models based on the ACF and PACF, and suggests how to choose p or q. [4]

6.3.2 Model comparison

6.3.3 Residuals

6.3.4 ARIMA

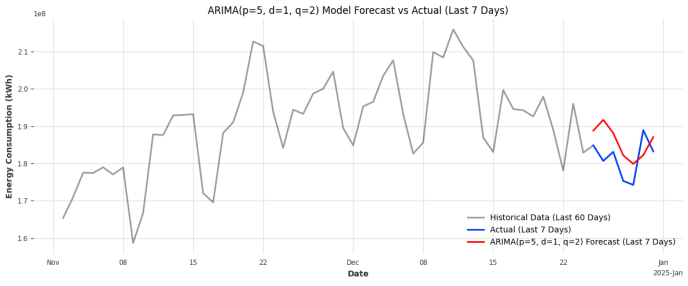


Figure 57: ARIMA 5 first attempt

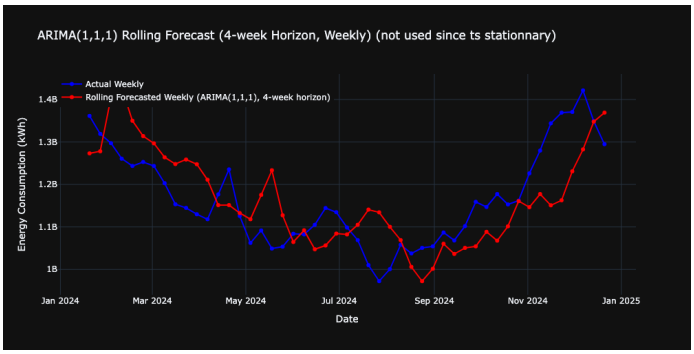


Figure 58: ARIMA(1,1,1) Rolling Forecast

6.3.5 Box-Jenkings Method

6.3.6 Model identification

Choice of d:

- Examine a plot of the data for non-stationarity.
- If the data seem non-stationary, we difference successively until they appear stationary.
- Usually $d = 1, 2$ is enough!

Choice of p and q:

- Examine the ACF and PACF of the differenced data.
- Sharp cut-off in the ACF after q lags suggests using an MA(q) model.
- Sharp cut-off in the PACF after p lags suggests using an AR(p) model.
- No sharp cut-off suggests ARMA model, hopefully with $p, q \leq 2$.
- No (or very slow) decline of ACF/PACF to zero suggests need to difference further, or to think again.
- If in doubt, opt for a parsimonious model, with fewer parameters. [4]

In this case, I have chosen ARMA(1,1) for now with d = 0.

6.4 SARIMA

6.4.1 Definition

Many geophysical time series have seasonal components. For example,

- hourly temperatures have 24-hour and annual cycles,

- monthly temperatures have a 12-month cycle.

For seasonal components the period is fixed and known (unlike cyclic behaviour). It may be useful to use an s-fold difference operator $I - B^s$, for example with $s = 12$ to remove the seasonal component from monthly temperatures.

The multiplicative seasonal autoregressive moving average model SARIMA$(p, d, q) \times (P, D, Q)_s$ is

$$\Phi_P(B^s)\phi(B)(I - B)^d(I - B^s)^D Y_t = \alpha + \Theta_Q(B^s)\theta(B)\varepsilon_t,$$

where $\{\varepsilon_t\}$ is Gaussian white noise. The ordinary autoregressive and moving average components are represented by the operators $\phi(B)$ and $\theta(B)$, respectively; the seasonal autoregressive and moving average components by $\Phi_P(B^s)$ and $\Theta_Q(B^s)$, of orders P and Q; and the ordinary and seasonal difference components by $(I - B)^d$ and $(I - B^s)^D$ of orders d and D.

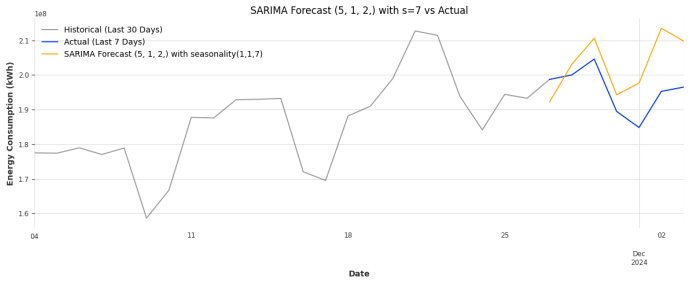


Figure 59: SARIMA 5 first attempt

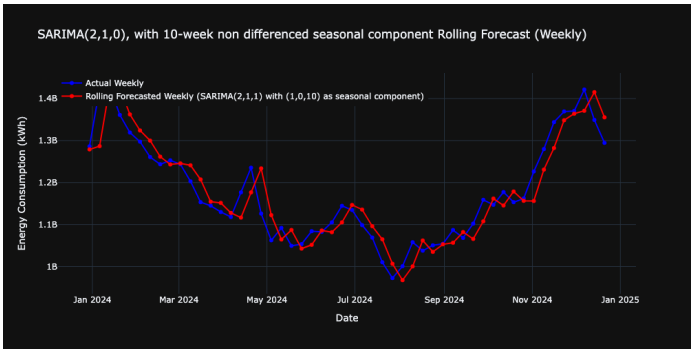


Figure 60: SARIMA(2,1,0), with 10-week non differenced seasonal component Rolling Forecast

6.4.2 Modeling procedure

6.4.3 Residuals

6.5 SARIMAX

6.5.1 Exogeneous Variables

6.5.2 Weather Data

7 Evaluation

7.1 MAPE

7.1.1 Definition

The mean absolute percentage error (MAPE) is a metric used to evaluate the accuracy of a forecasting model. It calculates the average of the absolute percentage errors between the predicted values and the actual values. Lower MAPE values indicate more accurate forecasts.

The formula is:

$$\text{MAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \left| \frac{y_t - \hat{y}_t}{y_t} \right|$$

where y_t is the true value and $\hat{y}_t$ is the predicted value at time t. [15]

7.1.2 MAPE comparison between models

Model Comparison – MAPE Scores

Model	MAPE (%)
AR(1)	5.9
AR(3)	6.2
ARMA(1,1)	6.02
ARIMA(1,1,1)	6.66

Figure 61: MAPE comparison between models

Other figures will be added.

7.2 AIC/BIC

7.2.1 Definition

The Akaike information criterion (AIC) and the Bayesian information criterion (BIC) are both used to compare the quality of different statistical models fitted to the same data. They take into account how well a model fits the data and how complex it is, penalizing models with more parameters.

The formulas are:

$$\text{AIC} = 2k - 2\ln(\hat{L}) \qquad \text{and} \qquad \text{BIC} = \ln(n)k - 2\ln(\hat{L})$$

where k number of estimated parameters, L maximum likelihood of the model, and n number of observations

BIC penalizes model complexity more than AIC because the penalty term $\ln(n)$ grows with the sample size, whereas AIC uses a constant penalty of 2 per parameter. This means BIC tends to prefer simpler models, especially with larger datasets. [14]

7.3 AIC/BIC Comparison between models

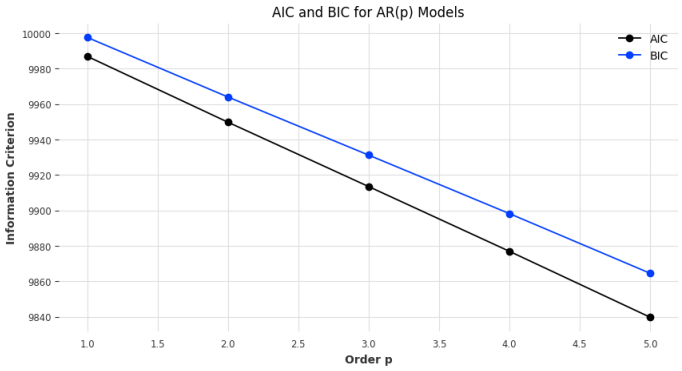


Figure 62: AIC and BIC values for AR models of increasing order

Other comparisons will be added.

7.4 Testing on other data

References

[1] André Bauer. *Automated Hybrid Time Series Forecasting: Design, Benchmarking, and Use Cases*. Doctoral thesis, Illinois Institute of Technology, 2021. License: CC BY-SA 4.0.

[2] Wikipedia contributors. Mean – Wikipedia, the free encyclopedia, 2024. Accessed: 2024-05-21.

[3] Wikipedia contributors. Variance – Wikipedia, the free encyclopedia, 2024. Accessed: 2024-05-21.

[4] Anthony Davison and Emeric Thibaud. *Time Series*. EPFL, 2019. MATH-342 Course, Anthony Davison © 2019.

[5] Figure 1. Electrical engineering portal. `https://engineering.electrical-equipment.org`.

[6] Figure 2. Coded using swissgrid data. Custom dataset, not published.

[7] Figure 3. Grid 2040 – nepal electricity authority. `https://grid2040.ku.edu.np`.

[8] Jim Frost. Using moving averages to smooth time series data, 2024. Accessed: 2025-05-30.

[9] GeeksforGeeks. What is a trend in time series? `https://www.geeksforgeeks.org/what-is-a-trend-in-time-series/`. Accessed: 2025-05-21.

[10] GeeksforGeeks Contributors. Seasonal decomposition of time series by loess (stl), 2021. Accessed: 2025-05-30.

[11] Nachi Keta. Do you know in what context the terms additive and multiplicative are used in the context of time series? `https://nachi-keta.medium.com/do-you-know-in-what-context-the-terms-additive-and-multiplicative-are-used-in-the-cont` 2021. Accessed: 2025-05-30.

[12] Milvus AI Contributors. What is the difference between additive and multiplicative time series models? `https://milvus.io/ai-quick-reference/what-is-the-difference-between-additive-and-multiplicative-time-series-models`, 2024. Accessed: 2025-05-30.

[13] Grant Sanderson. But what is the fourier transform? a visual introduction. `https://www.youtube.com/watch?v=spUNpyF58BY`, 2018. 3Blue1Brown, YouTube.

[14] Wikipedia contributors. Akaike information criterion — wikipedia, the free encyclopedia, 2024. Accessed: 2025-06-02.

[15] Wikipedia contributors. Mean absolute percentage error — wikipedia, the free encyclopedia, 2024. Accessed: 2025-06-02.

[16] Wikipedia contributors. Moving average – wikipedia, the free encyclopedia, 2024. Accessed: 2025-05-30.

[17] Wikipedia contributors. Smoothing – wikipedia, the free encyclopedia, 2024. Accessed: 2025-05-30.

[18] Wikipedia contributors. White noise — wikipedia, the free encyclopedia. `https://en.wikipedia.org/wiki/White_noise`, 2024. Accessed: 2025-05-21.