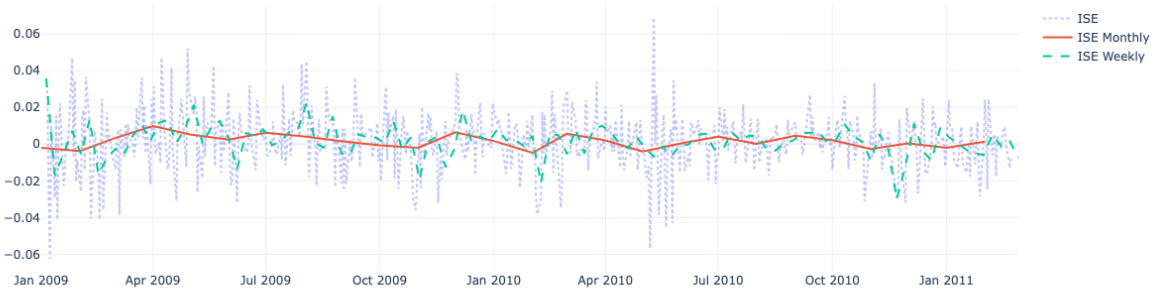
Exercise Week 1

Select 2-3 time series datasets: opendata.swiss, ec.europa.eu/Eurostat, Kaggle, etc. Reflect on the data generation process:

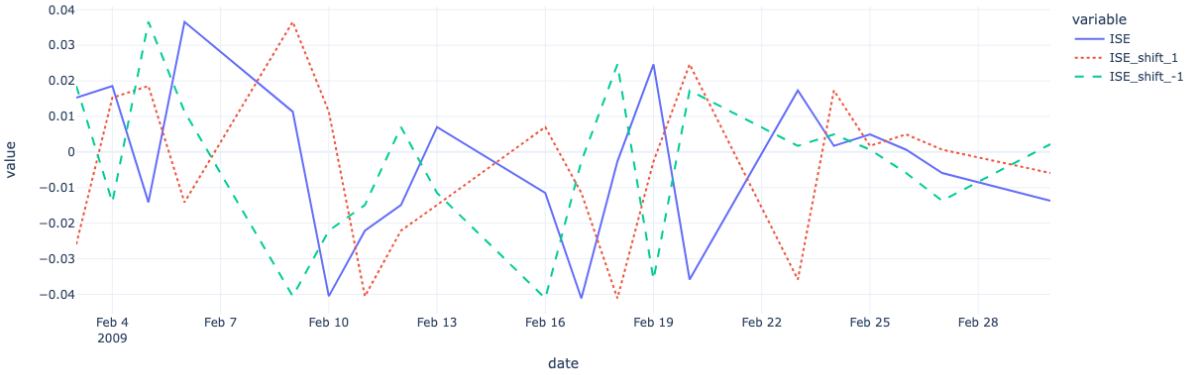
1. Airline Asset, Liability and Debt Ratio 2013-2023
    a. How was the data collected? Time interval, time range, etc.
    The data was collected by the United States Bureau of Transportation Statistics. It is quarterly.
    b. Which factors/laws influence the data?
    economic shocks (COVID-19), fuel prices, interest rates.
    c. Which patterns do you expect to find?
    Crisis-driven spikes (COVID-19), Recovery thereof, Seasonal fluctuations in prices
2. Avocado prices
    a. How was the data collected? Time interval, time range, etc.
    The dataset was compiled from the publicly available source Hass Avocado Board and retail scanner data. The date range is 2015-01-04 – 2018-03-25 weekly.
    b. Which factors/laws influence the data?
    Supply and demand fluctuations, Seasonality, Regulatory Policies (tariffs), weather/crop yields.
    c. Which patterns do you expect to find?
    Long-term upward trend in avocado purchases, Seasonal fluctuations in prices
3. Climate Change: Earth Surface Temperature Data
    a. How was the data collected? Time interval, time range, etc.
    The data comes from the Berkeley Earth project. They aggregate temperature records from many sources: weather stations, oceans, land-based sensors. It starts in 1743 until 2013, monthly
    b. Which factors/laws influence the data?
    Natural Climate Variability, Human-caused Climate Change
    c. Which patterns do you expect to find?
    Overall increase in global surface temperatures over the decades, especially in the 20th and 21st centuries. Because data is monthly, there will be recurring annual cycles. Regional differences

Review notebooks of Modern Time Series Forecasting with Python – Chapter 2 (GitHub repository)
Load and visualize datasets with Python.

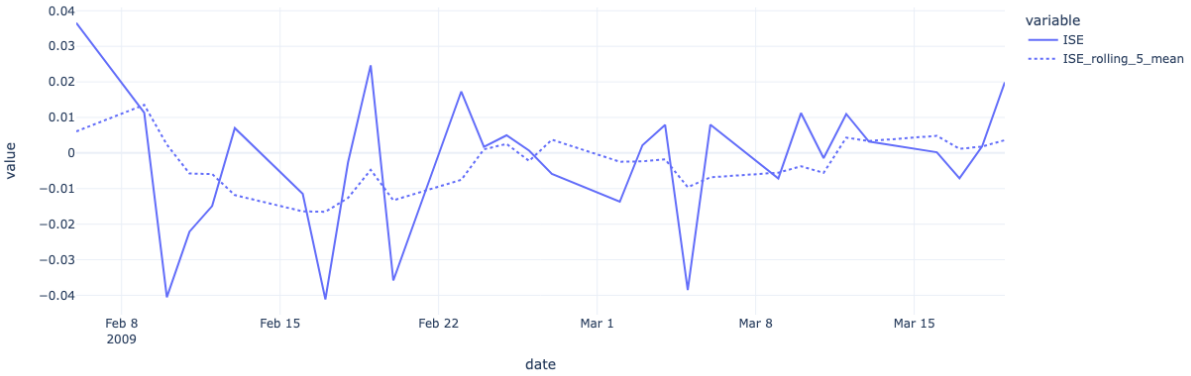
1 *Pandas Refresher & Missing Values Treatment.ipynb*



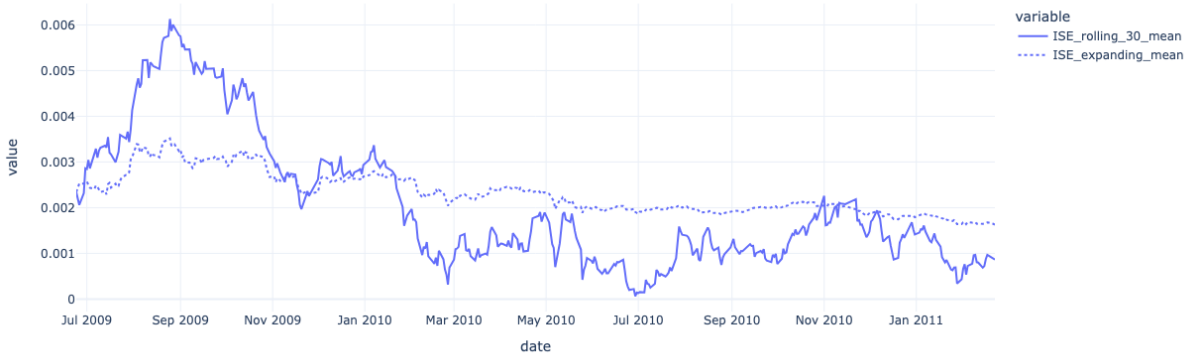
The initial plot effectively illustrates the volatility of daily stock returns. The blue line, representing daily returns, is highly volatile, which is characteristic of financial market data. The red and green lines, showing the weekly and monthly moving averages respectively, provide a smoothed view of the underlying trend, filtering out the daily noise.
Financial markets react to a constant stream of news, economic data, and investor sentiment, leading to the random, volatile behavior seen in the daily returns.
There is a period of increased volatility. This timing corresponds with the European Sovereign Debt Crisis, particularly of the Greek debt crisis reaching its first major peak.



The blue line fluctuates daily around zero, showing high noise and no visible long-term trend. The red and green dashed lines represent lagged values, which closely track the original but shifted in time. The fluctuations are sharp and frequent, typical of financial return data.
This behavior matches expectations for stock market returns: They are centered near zero with frequent swings up and down. There's no clear seasonality or smooth trend, consistent with efficient market behavior.
Around Feb 6–10, there are relatively large movements a change of 0.08. (Couldn't find anything linking to it)

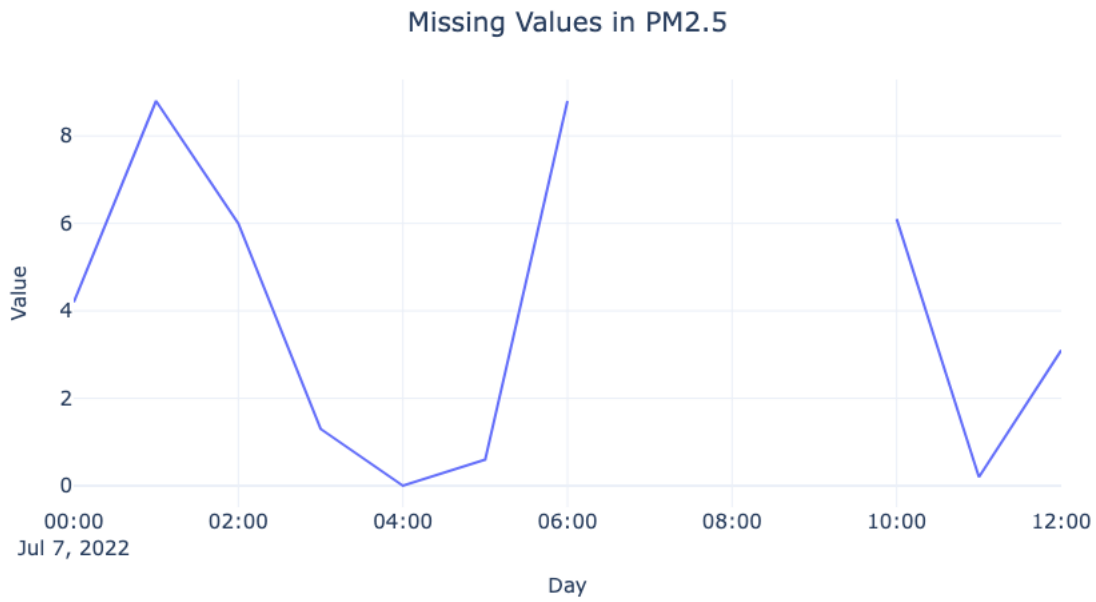


The daily ISE returns are a measure of the percentage change in the index's value from one day to the next. The rolling mean, on the other hand, is the average of the ISE returns over the previous five trading days. This helps to smooth out the daily fluctuations and highlight the underlying trend.
The data appears to be quite noisy and shows significant swings, suggesting a period of high volatility. The rolling average line, which stays close to zero, hints that financial data often show a tendency for returns to revert to their mean.
February 9, 2009: There's a significant drop in the ISE returns to nearly -0.04, followed by a sharp rise.
February 17, 2009: Another notable drop occurs, close to -0.04.
March 4, 2009: A similar sharp decline is visible, again approaching -0.04.
The Turkish stock market was likely a reflection of global economic uncertainty hence both positive and negative peaks.

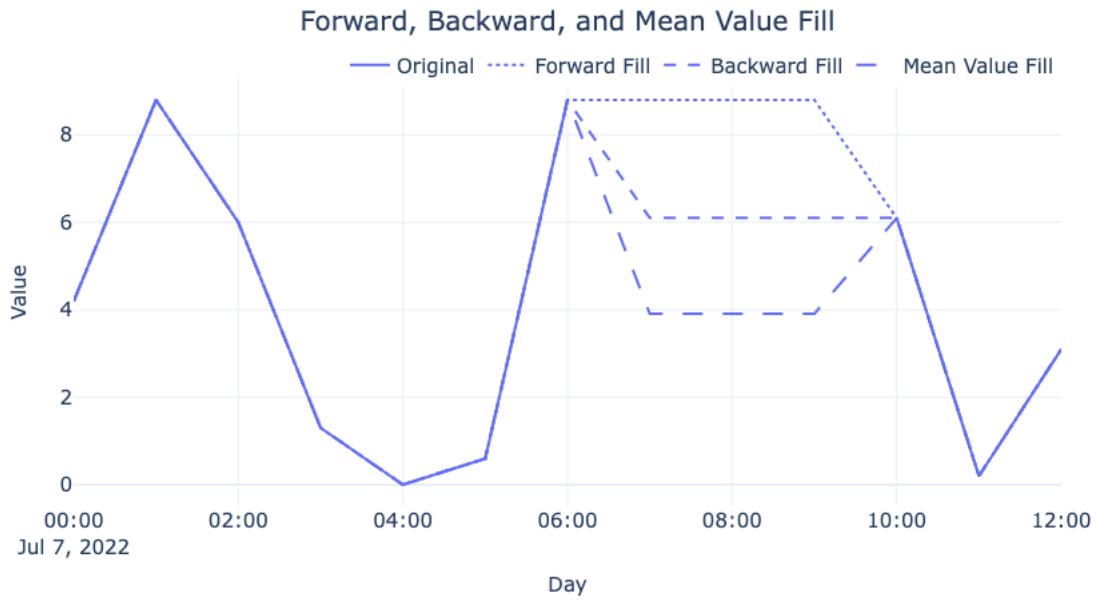


The solid line shows the 30-day rolling mean of ISE returns. This value at any point in time is the average of the daily returns over the preceding 30 days. It smooths out short-term fluctuations to reveal the medium-term trend. The dotted line shows the expanding mean. This is the average of all daily returns from the start of the data set (July 2009) up to each specific date.
Both the rolling and expanding means show a clear upward trend between July-September 2009. The rolling mean rises sharply, indicating a period of strong, positive returns for the ISE. This could reflect a recovery from the 2008 financial crisis. Following the peak in September 2009, the rolling mean shows a significant decline, indicating a period of lower average returns or even negative returns. The expanding mean also

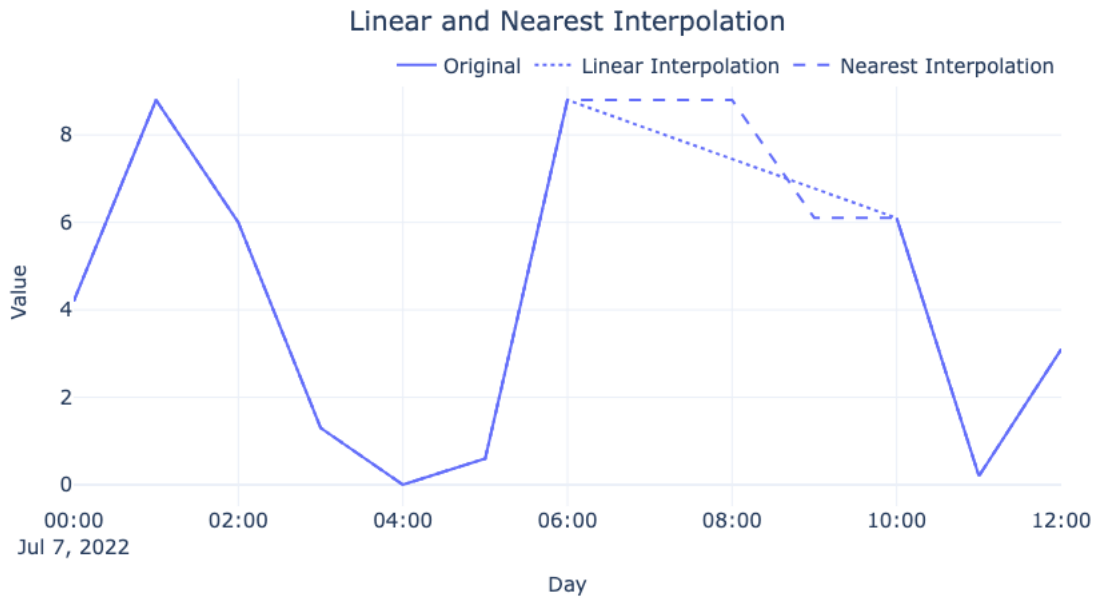
decreases, but much more gradually, which is expected since it's averaging over a much larger dataset.
The Greek debt crisis became a major concern in late 2009 and early 2010, which would be a prime candidate for explaining the downturn.



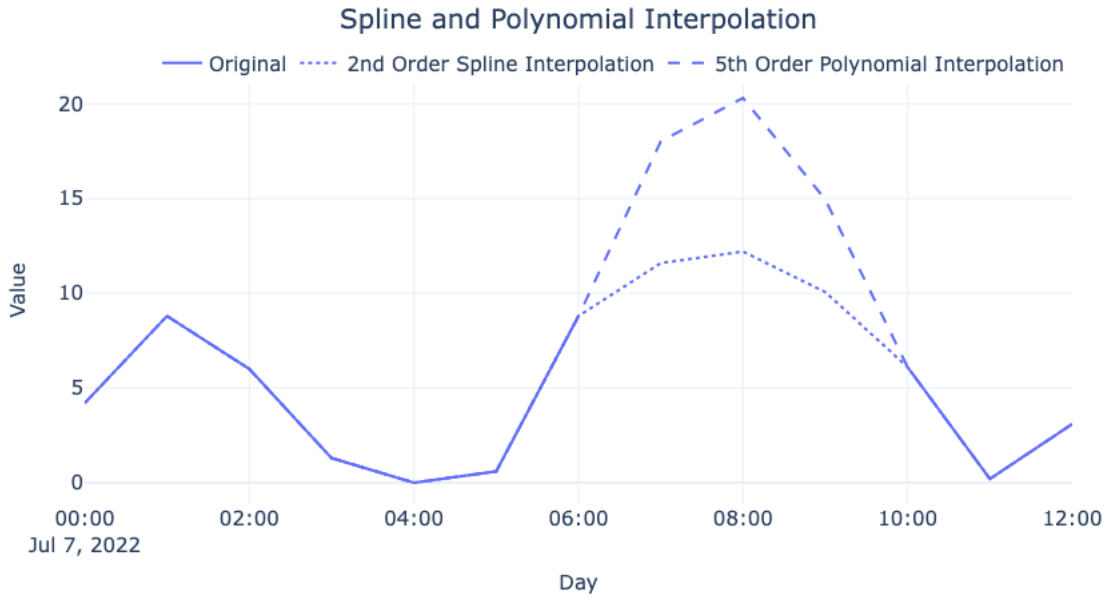
The plot shows a time series of PM2.5 measurements with significant data gaps.
The data points that exist are relatively smooth, showing a consistent pattern of rising and falling values. However, the large, unexplained gaps in the data (between 4:30 AM and 6:00 AM, and 6:00 AM and 10:00 AM) make it impossible to determine the true values or trends during those periods.
The available data suggests a daily pattern of increasing PM2.5 levels in the early morning. The large gaps, however, break this pattern and prevent a complete understanding of the underlying data-generating process for the entire day.
The most significant "outliers" are the missing data periods. They are likely caused by for example, sensor or instrument failure.



The plot displays the effects of different ways of filling in the missing PM2.5 values from the original plot. The Forward Fill method replaces missing values with the last known value. The Backward Fill method replaces missing values with the next available value. The Mean Value Fill method replaces missing values with the average of the values before and after the gap. The best method depends on the nature of the data.



This plot shows other common methods for handling the missing data from the previous plot. Linear Interpolation fills the gaps by drawing a straight line between the last known value and the next known value. It assumes a constant rate of change. Nearest Interpolation fills the gaps by using the value of the nearest data point.
They are both based on assumptions, not actual measurements.

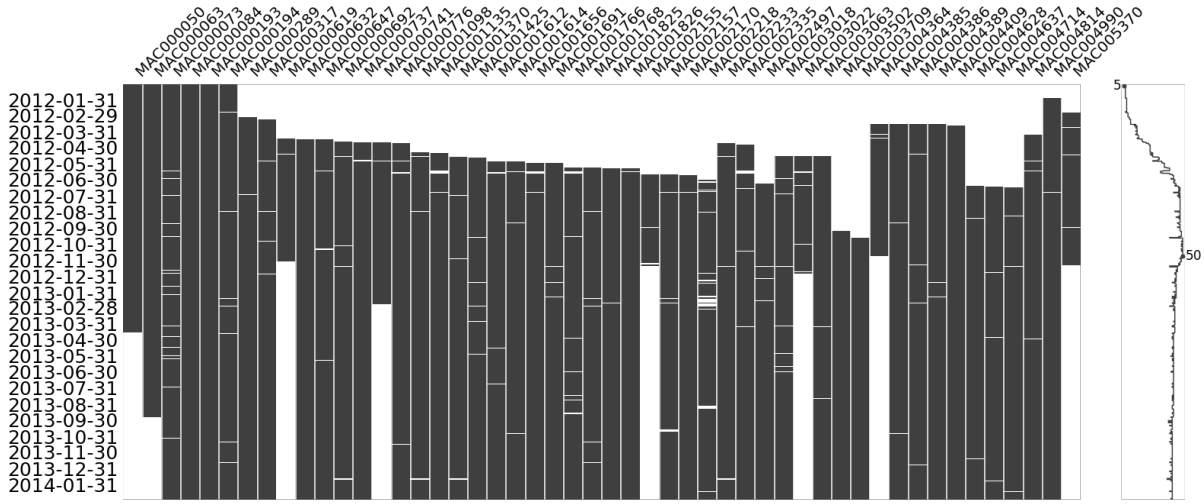


This plot shows another two ways of filling the missing data. Spline and Polynomial Interpolation. Both methods are more sophisticated than simple linear or nearest neighbor approaches because they attempt to capture the underlying curvature of the data.

Spline interpolation works by fitting a series of smooth curves between the known data points. Polynomial interpolation fits a single, complex curved line to all the known data points at once.

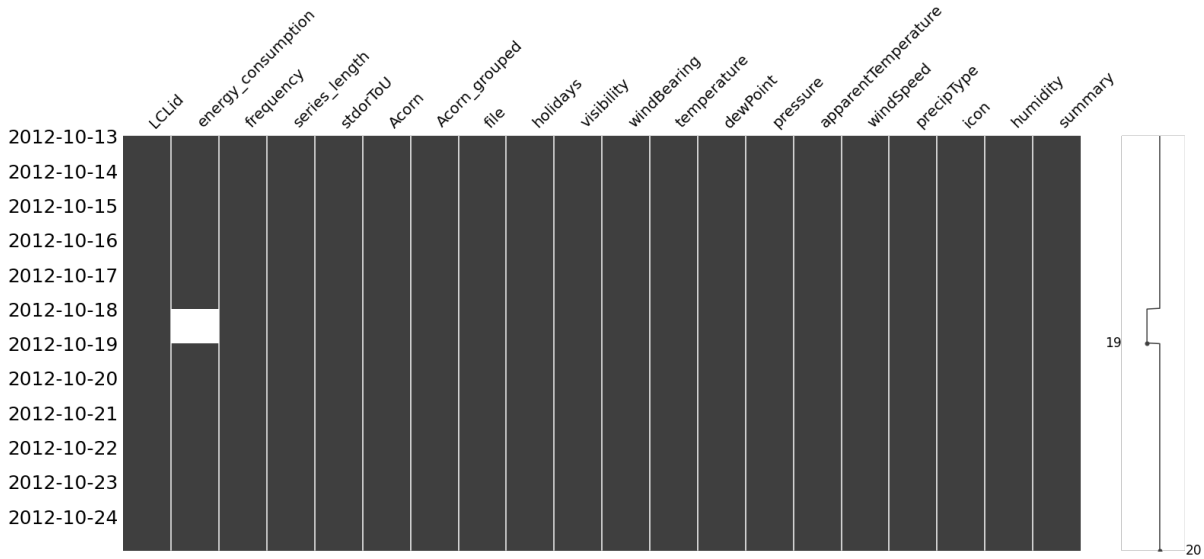
2 *Preprocessing London Smart Meter Dataset.ipynb*
No Plots?

3 *Handling Missing Data (Long Gaps).ipynb*

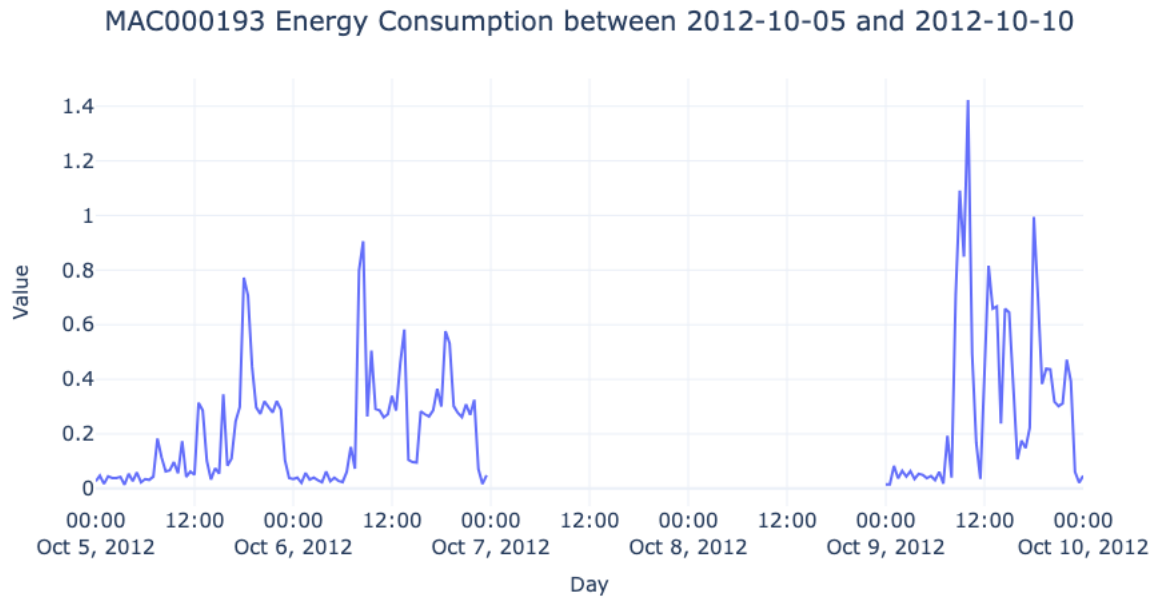


In this specific plot, each vertical bar represents a single time series, and each horizontal line corresponds to a specific point in time. The black segments within each bar represent present data, while the white lines or gaps indicate missing values for that particular time series at that specific time.

The pattern of missing data appears to be sporadic. There are visible vertical groupings of missing values.
The most plausible real-world event corresponding to these large-scale missing data periods is data collection or system failures.

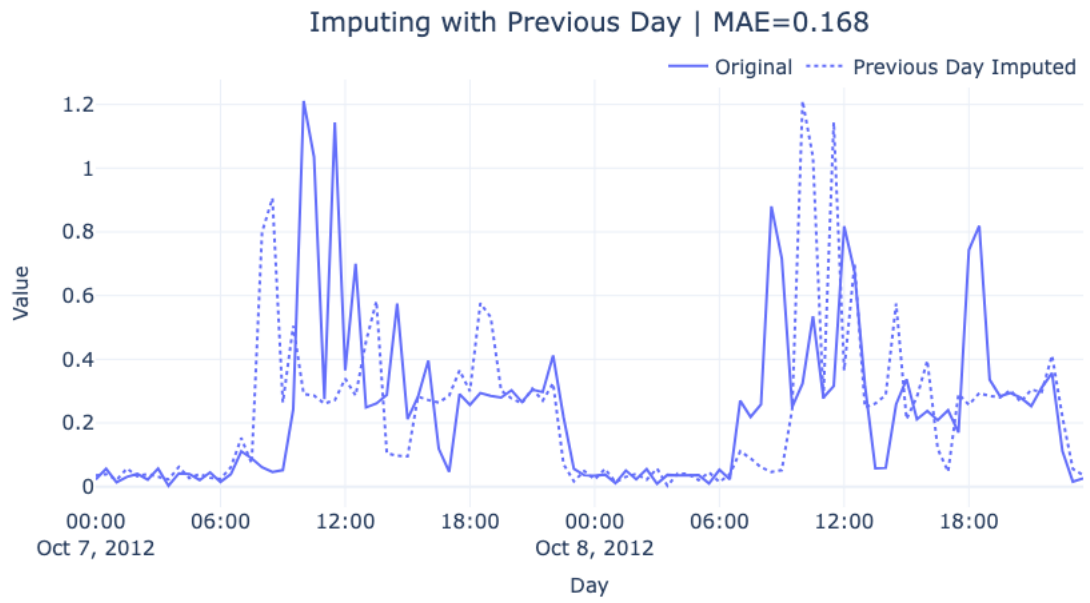


This plot is a missing data matrix focusing on a single smart meter with only one visible white space. The single white square is the only "outlier" in this plot. Possible real-world causes for this isolated missing data point could be a temporary power outage at the location of the meter.

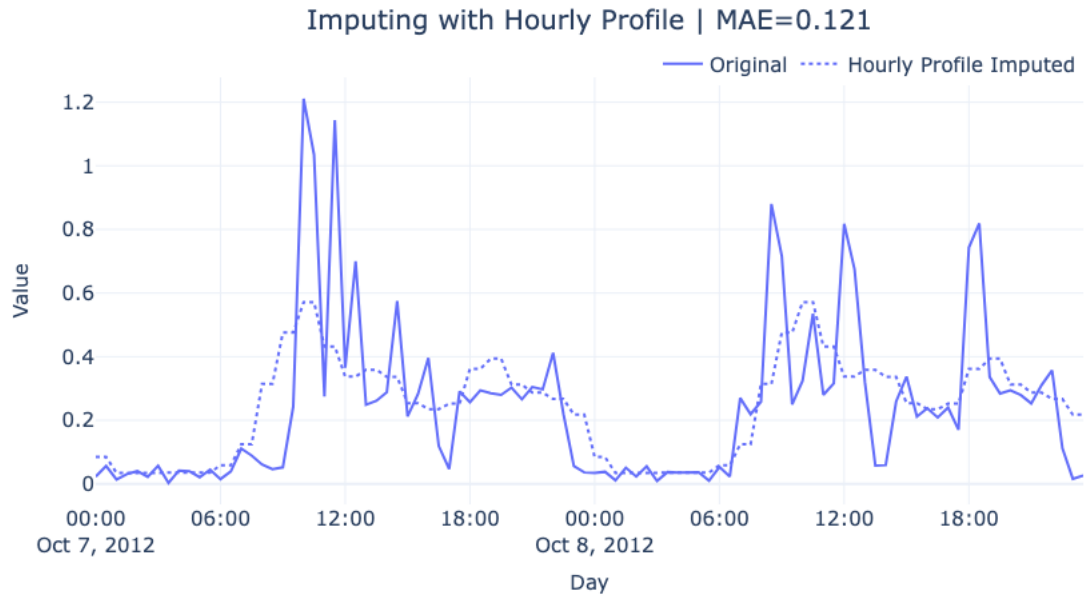


The plot reveals a clear daily consumption pattern. We see two distinct periods of high energy usage: one on October 6 and another on October 9.
The data shows a typical residential energy usage pattern, with spikes in the morning and evening as people wake up and return home from work, and a base load during the overnight hours. This pattern aligns with expectations for household energy use.

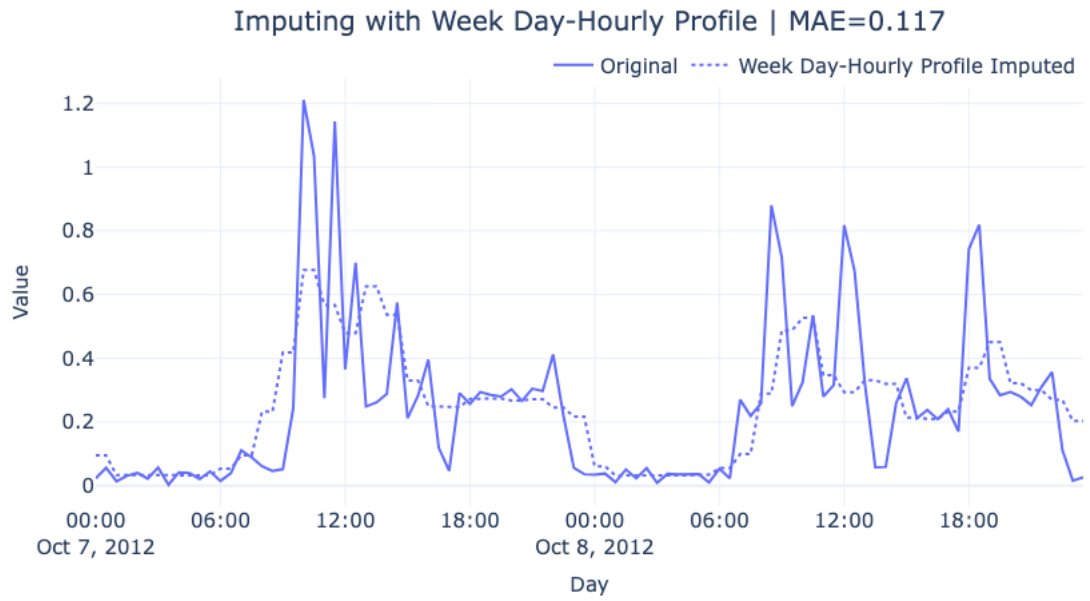
The most obvious feature in the plot is the large gap of missing data between the end of October 7 and the start of October 9.
Possible real-world events for the missing data gap include a local power outage or a meter malfunction specific to that location.



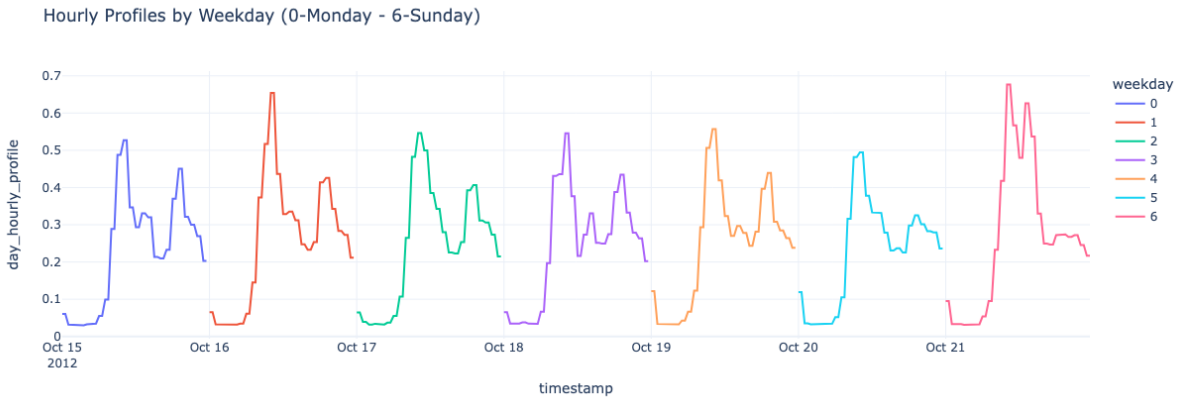
This plot is a time series chart that compares the original energy consumption data for a specific smart meter against data imputed using the "previous day" method.
The original data shows a clear and expected daily consumption pattern, with usage peaking in the daytime and dropping at night.
Variations in behavior, such as a change in routine, a day off, or different weather conditions, can cause day-to-day consumption patterns to vary.



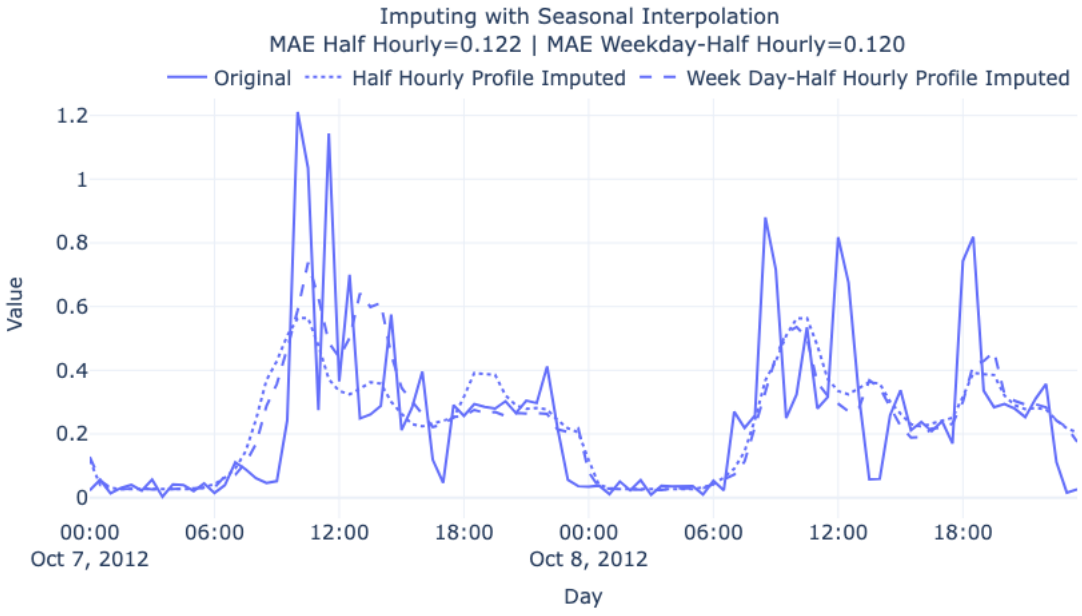
Compares the original energy consumption data for a smart meter to data imputed using an hourly profile method.
The original data shows again a typical daily consumption pattern with usage peaks during the day. The lower MAE of 0.121 compared to the previous "Previous Day Imputed" method (MAE=0.168) suggests that using an hourly profile is a more accurate imputation technique for this dataset.
It averages multiple days of data to create a representative profile, which is a more robust approach than simply copying a single previous day.



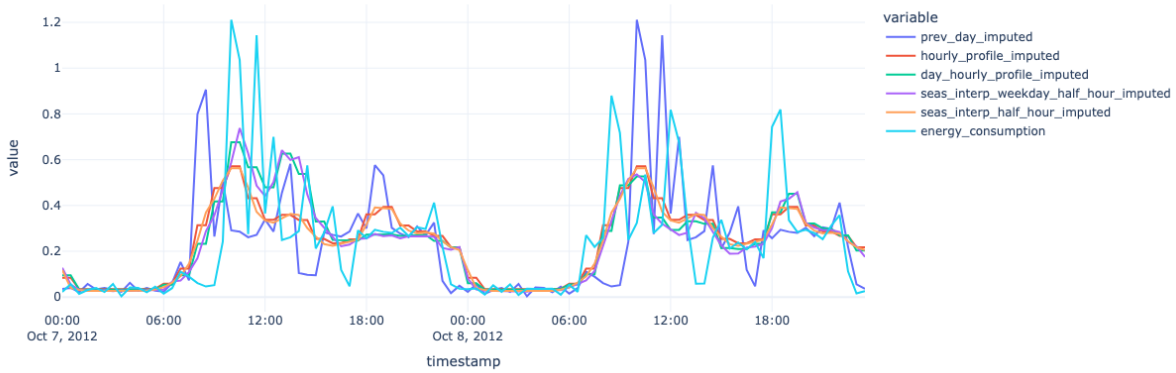
Compares the original energy consumption data for a smart meter against data imputed using a week day-hourly profile method.
The imputed data, based on a profile of what consumption typically looks like for that specific day of the week and hour, closely follows the general shape of the original data. The lower MAE of 0.117 compared to the previous methods (hourly profile: 0.121; previous day: 0.168) suggests that this method is the most accurate imputation technique for this dataset. Supports the notion that energy consumption is heavily dependent on daily and weekly routines.



This plot shows the average hourly energy consumption profiles for a smart meter, broken down by the day of the week.
The data is not noisy in the sense of random fluctuations; instead, it reveals strong, predictable patterns. Each day has a distinct profile, which is expected.
Weekdays show a similar pattern: a gradual increase in consumption in the morning, a plateau, and then a drop around midday, followed by an increase in the evening. This matches expectations for a typical household where people leave for work in the morning and return in the evening.
Weekends are markedly different. The energy consumption starts higher in the morning and remains elevated throughout the day, without the midday dip seen on weekdays. This is a very common pattern for residential households where people spend more time at home on weekends.



Compares the original energy consumption data to data imputed using two different methods: Half-Hourly Profile and Week Day-Half Hourly Profile. The plot's main purpose is to compare the performance of different imputation models. Both imputation methods closely follow the general shape of the original data. While the models capture the overall pattern, they cannot perfectly predict the day-to-day fluctuations in household behavior.



Comparison of several imputation methods against the original energy consumption data. The purple line ("prev_day_imputed") is the least accurate, as it simply copies the previous day's data, which doesn't align well with the original pattern. The other methods, particularly the orange line ("hourly_profile_imputed") and the green line ("day_hourly_profile_imputed"), provide a much better fit, as they capture the underlying hourly and daily seasonality.