CLASSICAL TIMESERIES FORECASTING: AMERICAN ENERGY POWER CONSUMPTION

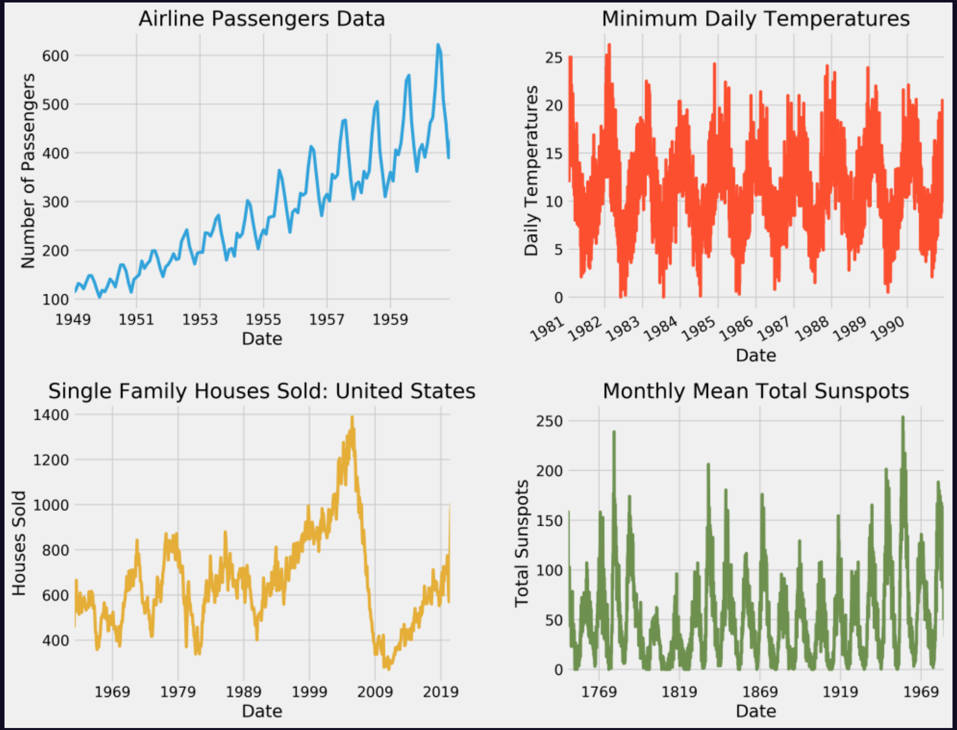
Deni Ramdani
https://www.linkedin.com/in/den-rmdani/

TABLE OF CONTENTS

INTRODUCTION 01

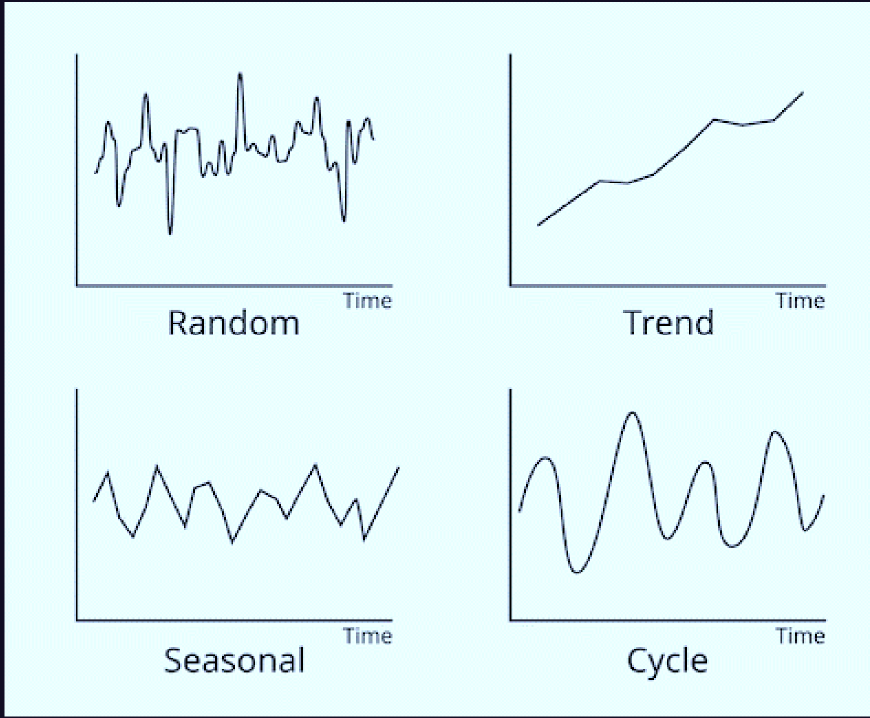
What is the time series?

Time series data is a sequence of data points collected or recorded at specific time intervals (e.g., daily stock prices, monthly sales figures, yearly temperature readings).



Timeseries Components

1. **Trend**: The long-term movement or direction in the data (upward, downward, or stable).
2. **Seasonality**: Regular, repeating patterns or cycles in the data (e.g., increased sales during holidays).
3. **Cyclical Patterns**: Long-term fluctuations due to economic or business cycles, which are not of fixed frequency.
4. **Irregular/Noise**: Random variations or anomalies that do not follow any pattern.



What can be forecast?

Forecasting is a critical tool for making informed decisions. It enables us to anticipate future trends and prepare for potential challenges, whether it's forecasting power plant demand or staffing needs in call centers.

The predictability of an event or a quantity depends on several factors including:

1. how well we understand the factors that contribute to it;
2. how much data are available;
3. whether the forecasts can a|ect the thing we are trying to forecast

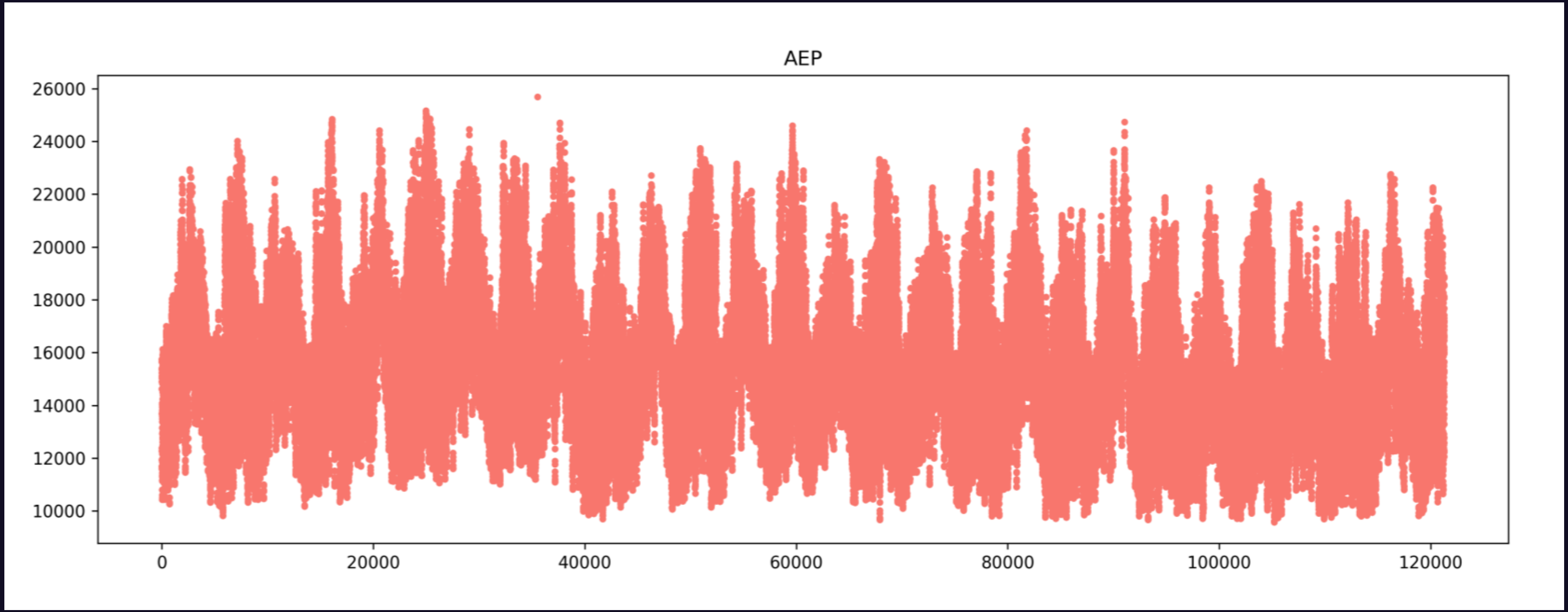
Forecasting, Planning, and Goals

Forecasting is about predicting the future as accurately as possible, given all of the information available, including historical data and knowledge of any future events that might impact the forecasts.

Planning is a response to forecasts and goals. Planning involves determining the appropriate actions that are required to make your forecasts match your goals.

Goals are what you would like to have happen. Goals should be linked to forecasts and plans, but this does not always occur. Too often, goals are set without any plan for how to achieve them, and no forecasts for whether they are realistic

DATA EXPLORATION 02



It is the American Energy Power (AEP) hourly energy consumption timeseries data.

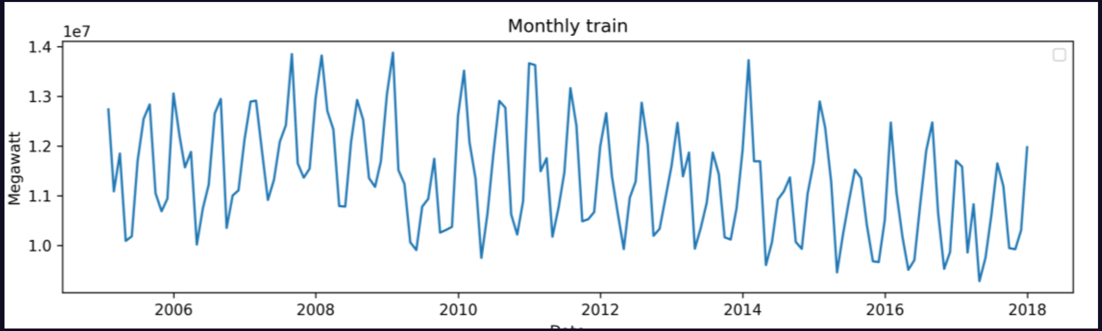
	Name	dtypes	Missing	Missing_%	Uniques	First Row	Last Row
0	Datetime	datetime64[ns]	0	0.0	121269	2004-12-31 01:00:00	2018-01-02 00:00:00
1	AEP_MW	float64	0	0.0	12643	13478.0	19993.0

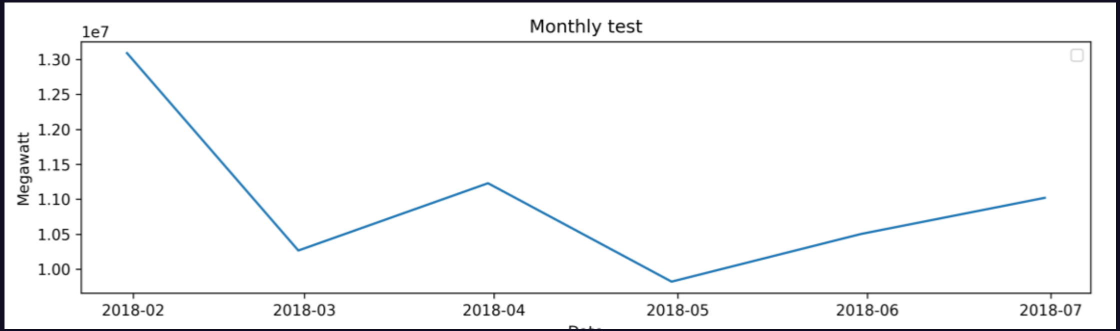
```
print(df['Datetime'].min())
print(df['Datetime'].max())
```

```
2004-10-01 01:00:00
2018-08-03 00:00:00
```

The data set comprises two columns, each containing a time stamp in the format "YYYY-MM-DD" and the amount of energy consumed in megawatts per hour. The data set spans the period from **October 2004 to August 2018**.

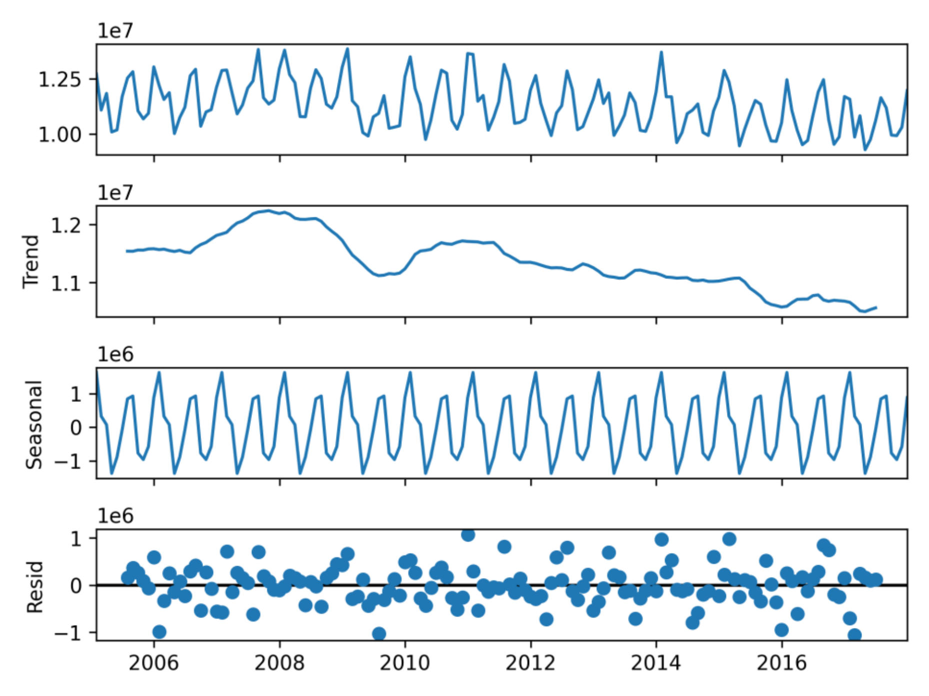
The data collected on an hourly basis will be aggregated to provide a total for each month. Subsequently, the data will be separated into two distinct categories: testing data and training data. This will result in the generation of the following graph.





Decomposition

The data illustrated on the accompanying diagram displays a **downward trend**, which is **seasonal** in nature, and exhibits **random residuals**.



ADF and KPSS Testing

```python
result = adfuller(monthly_train)
print('p-value: %f' % result[1])

result = adfuller(monthly_train.diff(3).dropna())
print('p-value: %f' % result[1])
```

```
p-value: 0.522386
p-value: 0.000000
```

```python
result = kpss(monthly_train['AEP_MW'])
print(f"KPSS Statistic: {result[0]}")
print(f"p-value: {result[1]}")

# Assuming a significance level of 0.05
if result[1] > 0.05:
    print("The data is likely trend stationary.")
else:
    print("The data appears to be non-stationary.")
```

```
KPSS Statistic: 1.0628473729471115
p-value: 0.01
The data appears to be non-stationary.
```

The results indicate that the ADF p-value is greater than 0.05 and the KPSS p-value is less than 0.05, which suggests that the **timeseries data are non-stationary**.

CLASSICAL TIMESERIES FORECASTING

03

Simple Moving Average

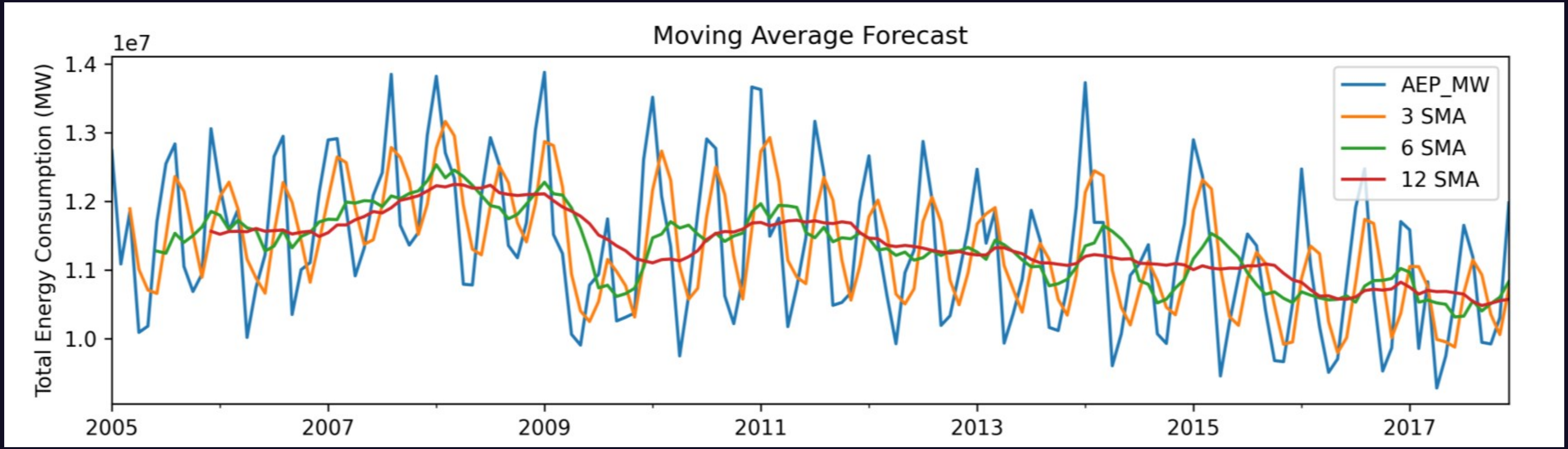
The **Simple Moving Average** (SMA) works by calculating the average of a specific number of past data points. Imagine a window that slides along the time series, constantly including the most recent data points and excluding the older ones.

$$M_T = \frac{1}{N} \sum_{t=T-N+1}^{T} y_t \,,$$

M_T : moving average at time periode T

N : number of recent observations

y_t : timeseries data point at time t



The image above illustrates the simple moving averages (SMAs), namely the 3-SMA, 6-SMA, and 12-SMA.

SMA:

- easy to understand, good for trends
- smooth noise, but limited forecasting.

Simple Exponential Smoothing

Simple Exponential Smoothing (SES) offers a more sophisticated alternative to the Moving Average (SMA) by assigning greater weight to recent data points. In contrast to the equal weighting applied in SMA, SES gradually reduces the weight given to older observations. This approach allows for a more responsive smoothing effect by prioritizing newer information.

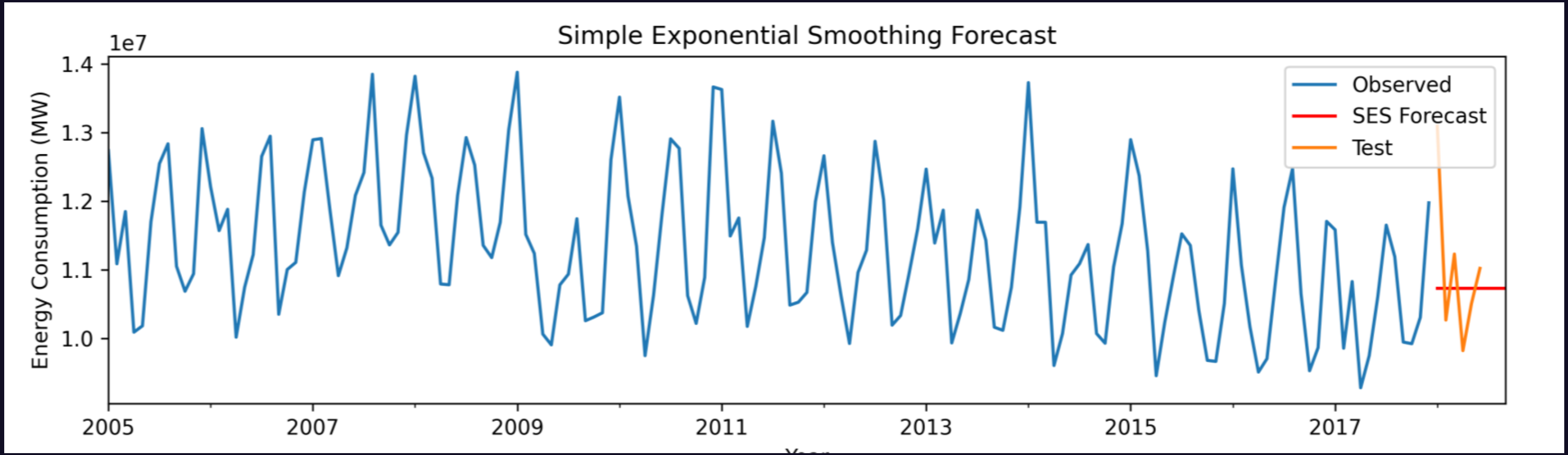
$$\mathcal{L}_t = \alpha y_t + (1-\alpha)\mathcal{L}_{t-1},$$
$$\tilde{y}_{\boldsymbol{t+h}} = \mathcal{L}_{\boldsymbol{t}}$$

$\mathcal{L}_t$: simple exponential smoothing at time periode t

α : smoothing factor $0 < \alpha < 1$

y_t : current observations

$\tilde{y}_{t+h}$: forecast



SES forecasting is most effective for steady data but less reliable for identifying trends. This method assumes that the data set is free of upward or downward trends, which can result in flat lines over time. This makes SES unsuitable for long-term predictions, especially when the data set contains trends, whether short-term or long-term.

The data set is non-stationary, which makes simple exponential smoothing an unsuitable forecasting method.

Holt's Linear Exponential Smoothing

Holt's LES builds on SES by adding "smoothing constants" for both **the level (α)** and **the trend (β)**. It estimates the level ($\mathcal{L}_t$) and trend ($\mathcal{T}_t$) at each point (t) using the previous estimates ($\mathcal{L}_{t-1}$, $\mathcal{T}_{t-1}$) and the observed value (y_t) at time t. These estimates are calculated through separate exponential smoothing equations, allowing the model to capture both the level and the underlying trend in the data.

$$
\begin{aligned}
\mathcal{L}_t &= \alpha y_t + (1-\alpha)(\mathcal{L}_{t-1} + \mathcal{T}_{t-1}),\\
\mathcal{T}_t &= \beta(\mathcal{L}_t - \mathcal{L}_{t-1}) + (1-\beta)\mathcal{T}_{t-1},\\
\tilde{y}_{t+h} &= \mathcal{L}_t + h\mathcal{T}_t.
\end{aligned}
$$

$\mathcal{L}_t$: estimated of level

α : smoothing factor $0 < \alpha < 1$

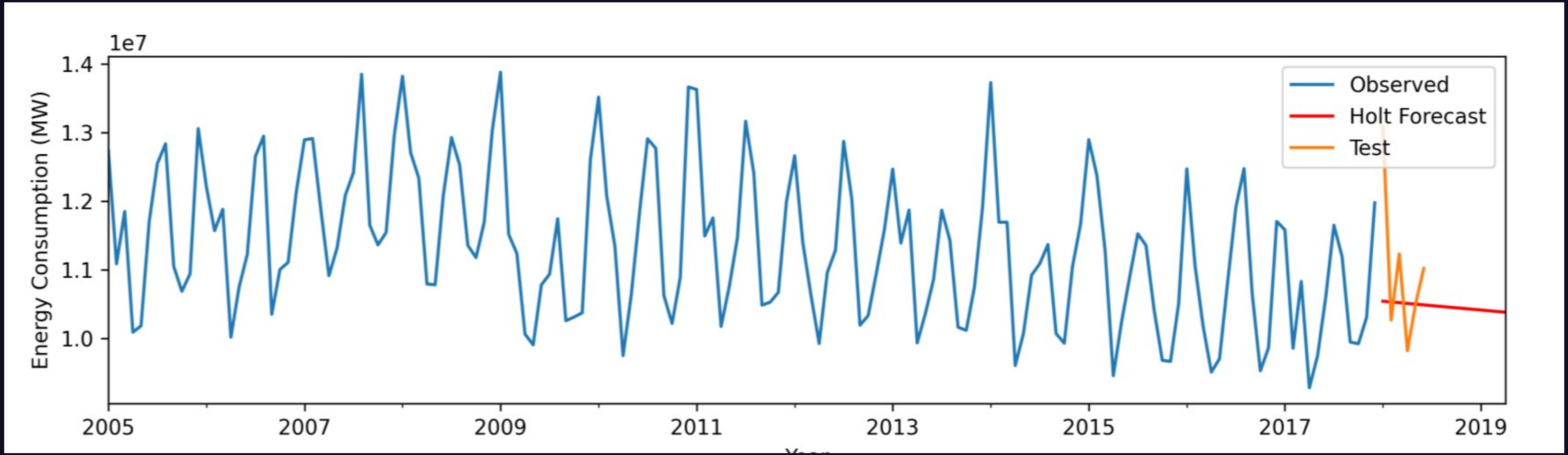
y_T : current observations

$\mathcal{T}_t$: estimated of trend

β : smoothing factor $0 < \beta < 1$

$\tilde{y}_T$: forecast

h : timestep

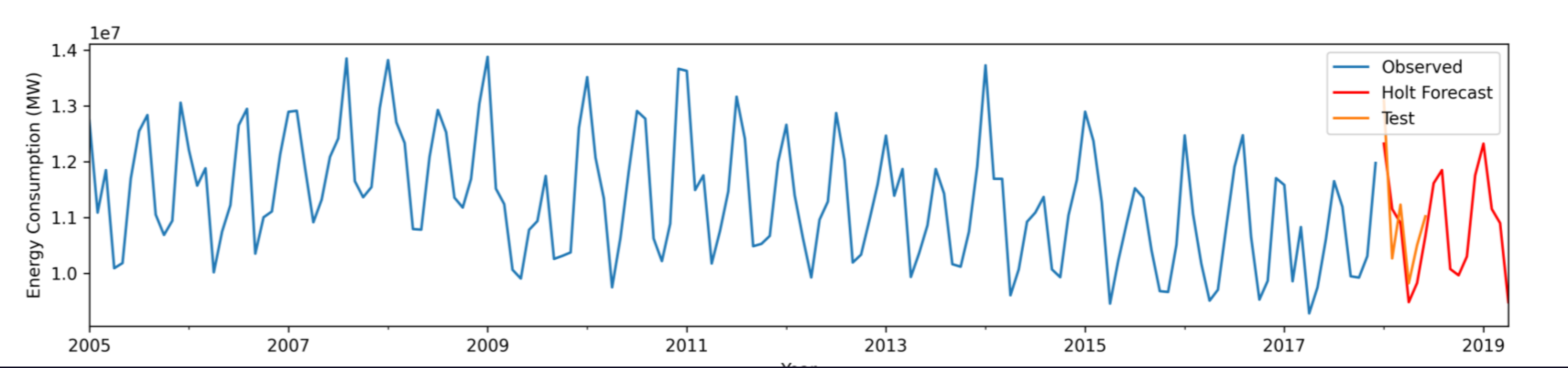


While this method is applicable when a data trend exists, it is important to exercise caution when extrapolating those trends over long periods, as they may not be representative of the full picture.

Holt's linear trend model is an appropriate choice for data sets exhibiting a linear trend without seasonal fluctuations.

Holt-Winters Seasonal Model

Holt-Winters extends the seasonal exponential smoothing (SES) method, which handles data with both trends and seasonal patterns. The Holt-Winters method incorporates a "seasonal component" alongside the level and trend, allowing it to capture those repeating seasonal fluctuations in the data.



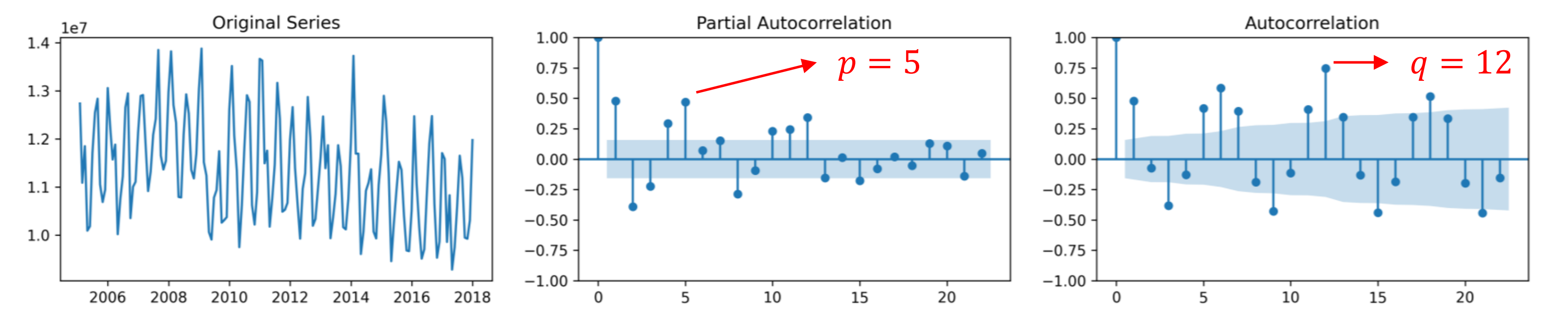
Holt's winter seasonal model is designed to capture the seasonal nuances of a timeseries. The results demonstrate that this method is an appropriate fit for the data in this case.

ARIMA

Univariate **ARIMA**, also known as Box-Jenkins after its creators, is a method of forecasting future values of a single series by analyzing its past behavior, which is characterized by inertia. The method is particularly effective for short-term predictions with at least 40 data points and consistent patterns; however, it is less adept at handling outliers or highly volatile data. While ARIMA often outperforms exponential smoothing for longer, stable data sets, smoothing methods may be more suitable for shorter or more volatile situations. In the event that the data set is limited (less than 40 points), it may be prudent to consider alternative forecasting techniques.

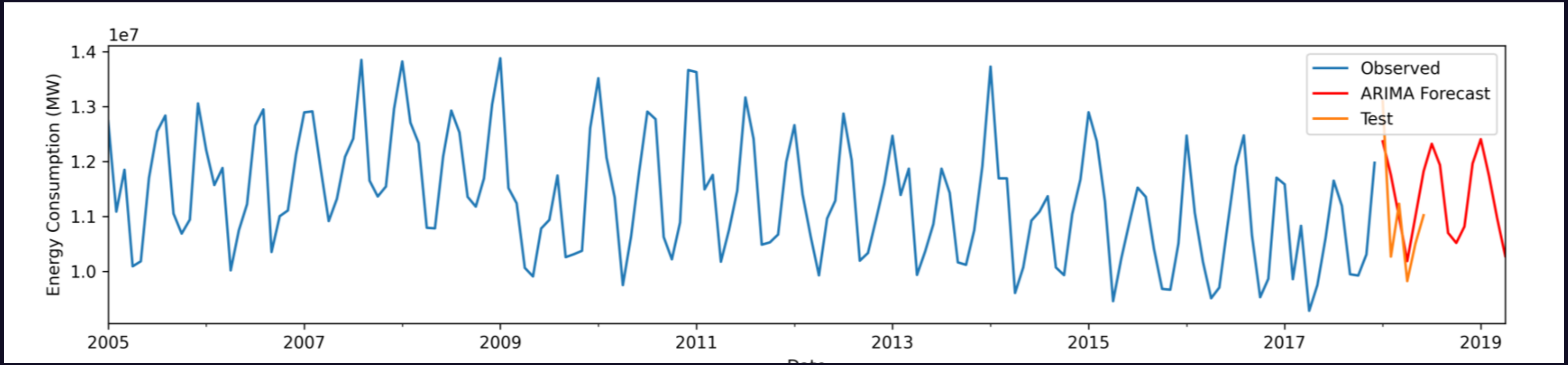
How to find *p*, *d*, and *q*?

It is established that the data in question is a non-stationary time series; consequently, differencing with a value of $d = 1$ is required.



The selection of these variables, p, d, and q, is based on the subjective assessment

ARIMA(5, 1, 12)

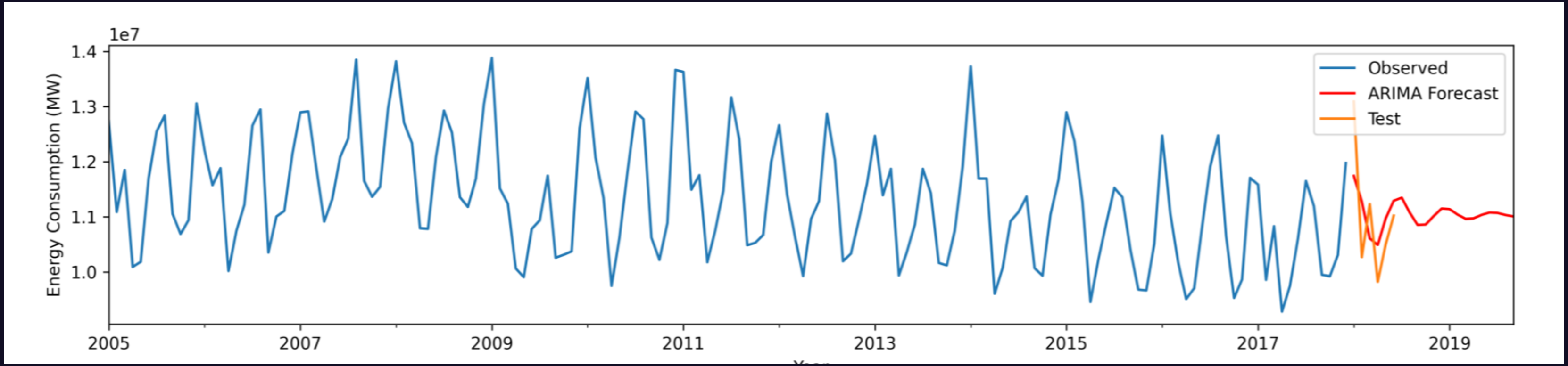


ARIMA provides an optimal balance between parsimony and flexibility. In contrast to exponential smoothing and linear regression, ARIMA is capable of accommodating more intricate data patterns. Indeed, some more basic exponential smoothing models are, in fact, specialised versions of ARIMA (such as the basic exponential smoothing model being equivalent to ARIMA(0,1,1)). This flexibility enables ARIMA to discern intricate patterns in the data set while maintaining a straightforward and robust methodology, thus avoiding the pitfalls of overfitting.

Using autoarima

```
Best model:  ARIMA(4,1,0)(0,0,0)[0]
Total fit time: 3.622 seconds
```

ARIMA(4, 1, 0)

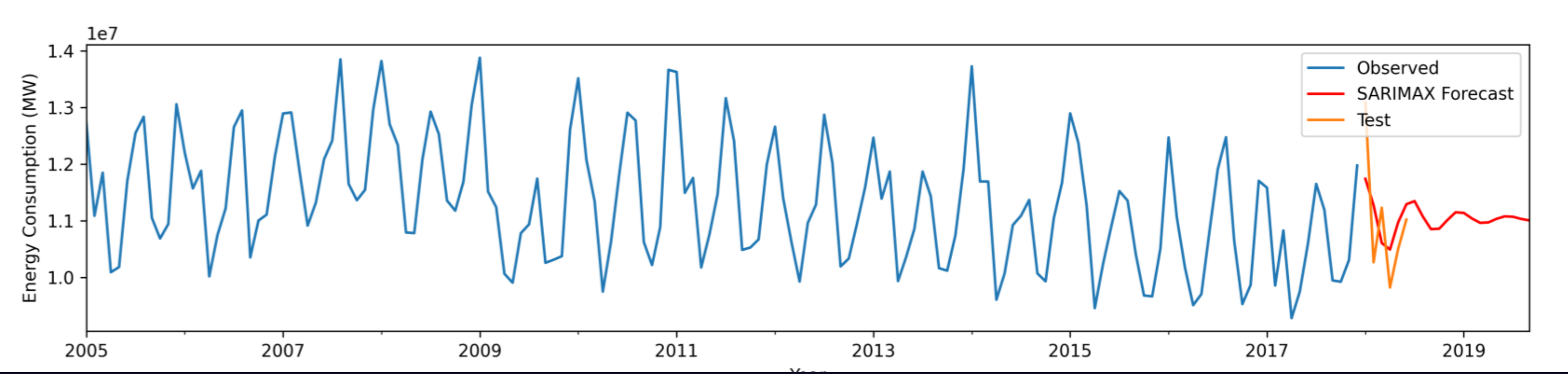


SARIMA

SARIMA extends ARIMA to handle seasonal data.

Indeed, this is the order for SARIMA, which is referred to as the seasonal order.

```
Best model:  ARIMA(4,1,0)(0,0,0)[0]
Total fit time: 3.622 seconds
```



CONCLUSIONS

Classical forecasting models, while popular (especially ARIMA), have shortcomings:

1. **They can't handle missing data**: This can be a major drawback if your data has gaps.
2. **They assume linear relationships**: Real-world data often has more complex relationships, which these models might miss. Data transformations can help somewhat.
3. **They focus on single variables**: Many forecasting problems involve multiple influential factors, which classical models can't account for directly.

These limitations make deep learning techniques a strong alternative, as they can handle more complex data structures and relationships. Even ARIMA, a common choice for financial forecasting, has issues:

- Limited ability to capture non-linear relationships: This can lead to inaccurate predictions when data patterns are not straightforward.
- Assumption of constant error variance: This might not hold true in real-world scenarios, affecting model accuracy. However, integrating ARIMA with GARCH models can address this.

REFERENCES

Books:

1. Montgomery, Douglas C., Cheryl L. Jennings, and Murat Kulahci. Introduction to Time Series Analysis and Forecasting. New Jersey: John Wiley & Sons, Inc., 2015.
2. Hyndman, Rob J., and George Athanasopoulos. Forecasting: Principles and Practice. Australia: Otexts, 2018.

Articles:

1. Dash, Shailey. "An Overview of Time Series Forecasting Models Part 1: Classical Time Series Forecasting Models." Medium. https://medium.com/@shaileydash/an-overview-of-time-series-forecasting-models-part-1-classical-time-series-forecasting-models-2d877de76e0f. Accessed July 17, 2024.
2. Ichramsyah, Mochamad Kautzar. "Classical Time-Series Forecasting Methods in Python and R for Beginners." CodeX. Medium, May 2024. https://medium.com/codex/classical-time-series-forecasting-methods-in-python-and-r-for-beginners-df0731b3199f. Accessed July 17, 2024.
3. Tiwari, Ajay. "Let's Forecast Your Time Series using Classical Approaches." Towards Data Science. Medium. https://medium.com/towards-data-science/lets-forecast-your-time-series-using-classical-approaches-f84eb982212c. Accessed July 17, 2024.

THANK YOU

den.rmdani@gmail.com

https://www.linkedin.com/in/den-rmdani/