Time Series Forecasting

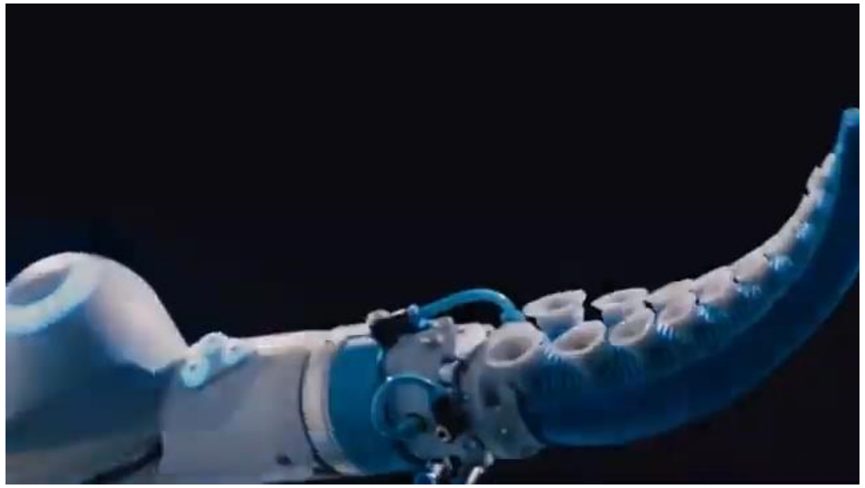
EDHEC BUSINESS SCHOOL

About me

- **Postdoctoral Researcher,** 2024-
    - Vista Group, Ecole Polytechnique France
    - Main topic: Control Algorithms for Soft Robotics

- **PhD,** 2019-2023:
    - Ecole Polytechnique, IRT SystemX
    - Mathematics and Informatics
    - Topic: How to make a robot perform a task that it was not trained for beforehand?

My work

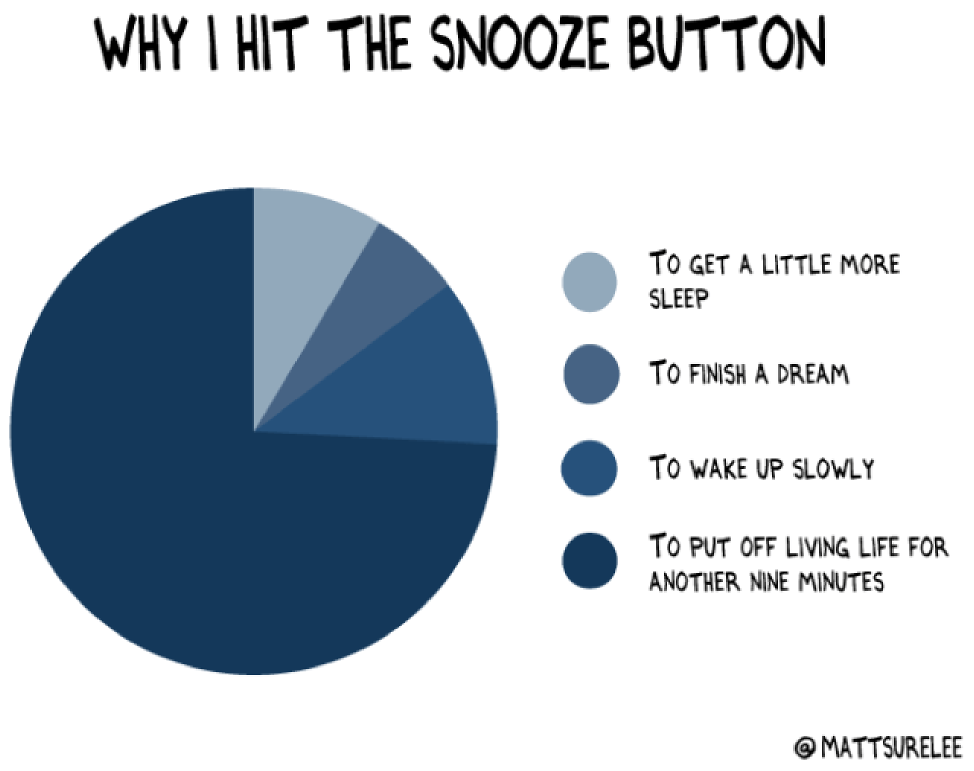


Grading

- 60% Final Exam (modalities to be disclosed very soon)
- 30% Assignments. 2 Labs will be graded, 15% each
- 10% participation and answers

<u>Introduction</u>

The future is influenced by the past

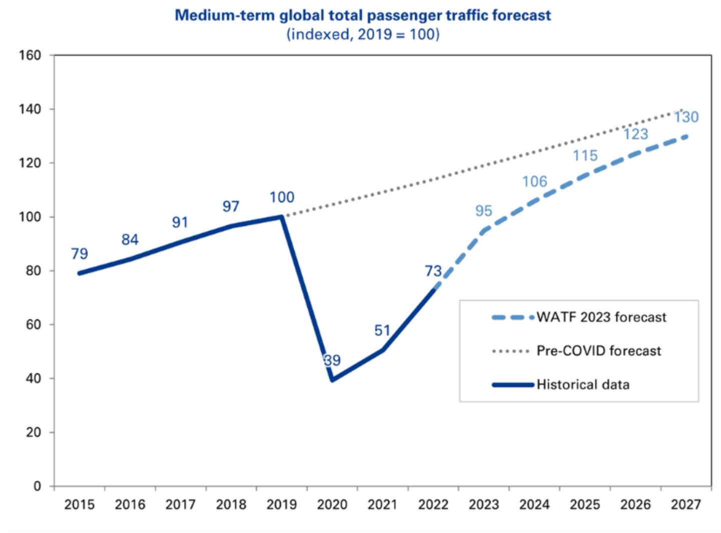


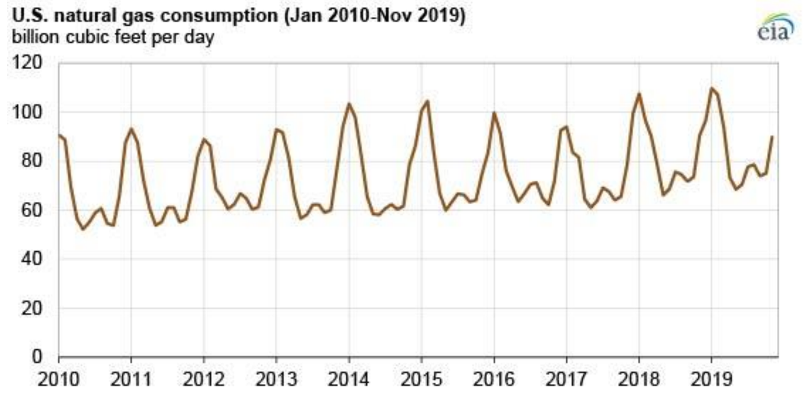
Day	Snooze Count	Snooze Times
Monday	3	7:05 AM, 7:12 AM, 7:20 AM
Tuesday	5	7:10 AM, 7:15 AM, 7:25 AM, 7:30 AM, 7:35 AM
Wednesday	2	7:08 AM, 7:18 AM
...	...	...

The power of patterns

More classical examples:

- Predicting next week hourly temperature
- the development of a complex linear stochastic model for predicting the movement of short-term interest rates.
- forecast the demand for airline capacity,
- seasonal energy demand
- future online sales
- …

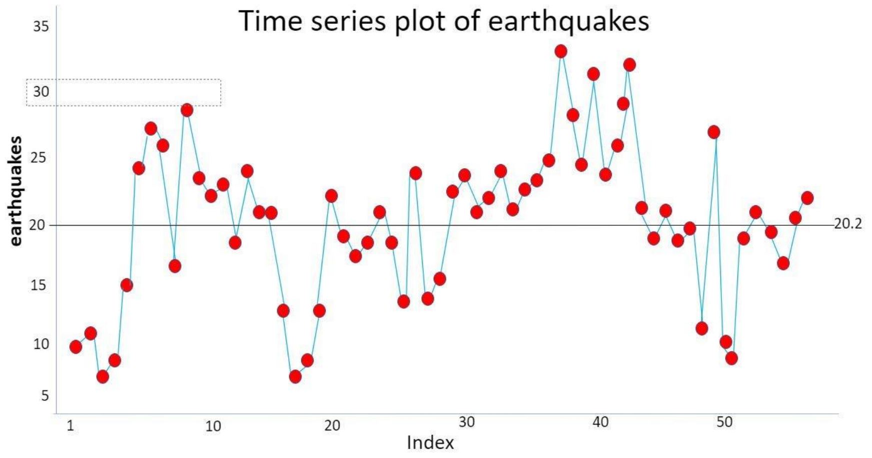




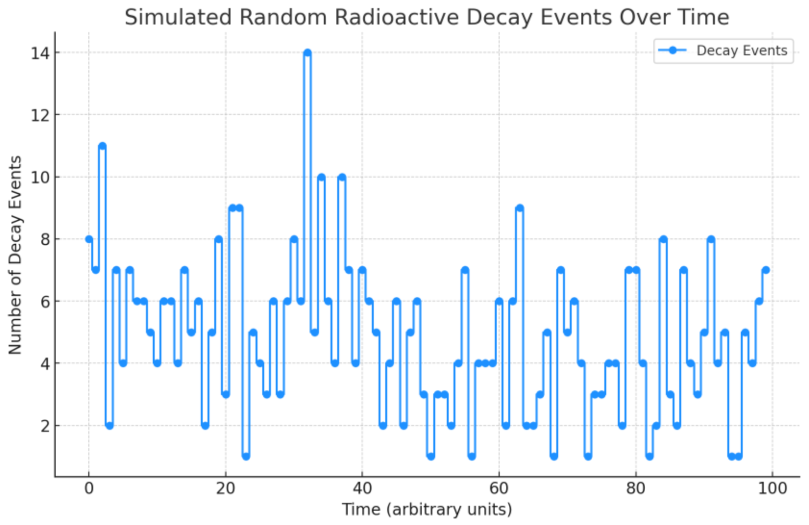
Predicting next values seems easy. WHY?

Trends and Seasonality

Patterns are not so easy to observe



→ Hidden Latent variables that are difficult to observe



→ No latent variable here, the phenomenon is entirely random.

Question: How to be sure to not try to use time forecasting models to predict processes that are actually random or when we have non sufficient or expressive data ?

Time Series Analysis vs. Forecasting

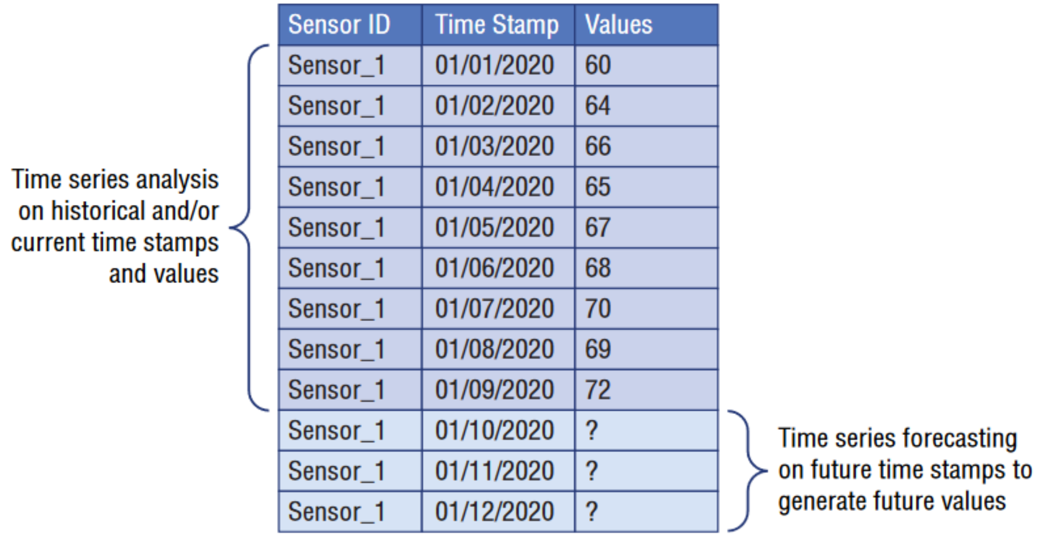
Step 1

Analysis: time series analysis is about identifying the intrinsic structure and extrapolating the hidden traits of your time series data in order to get helpful information from it

- Acquire clear insights of the underlying structures of historical time series data..
- Increase the quality of the interpretation of time series features to better inform the problem domain.
- Preprocess and perform high-quality feature engineering to get a richer and deeper historical data set.

Step 2

Forecasting: taking machine learning models, training them on historical time series data, and consuming them to forecast future predictions



Sensor ID	Time Stamp	Values
Sensor_1	01/01/2020	60
Sensor_1	01/02/2020	64
Sensor_1	01/03/2020	66
Sensor_1	01/04/2020	65
Sensor_1	01/05/2020	67
Sensor_1	01/06/2020	68
Sensor_1	01/07/2020	70
Sensor_1	01/08/2020	69
Sensor_1	01/09/2020	72
Sensor_1	01/10/2020	?
Sensor_1	01/11/2020	?
Sensor_1	01/12/2020	?

Time series analysis on historical and/or current time stamps and values
Time series forecasting on future time stamps to generate future values

Components of a Time Series

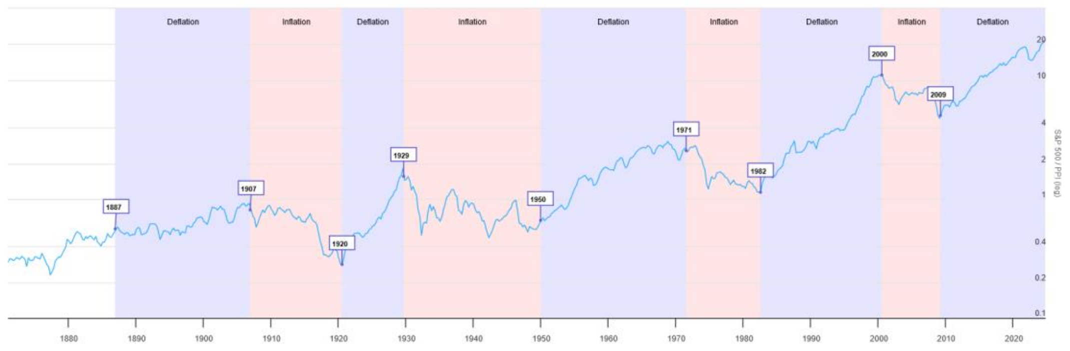
Long-Term Trend: A general upward or downward movement.

Seasonal Patterns: Regular fluctuations within a specific period.

Cyclic Movements: Less predictable short-term cycles.

Random Fluctuations: Unpredictable noise or variations.

Stocks to commodities ratio



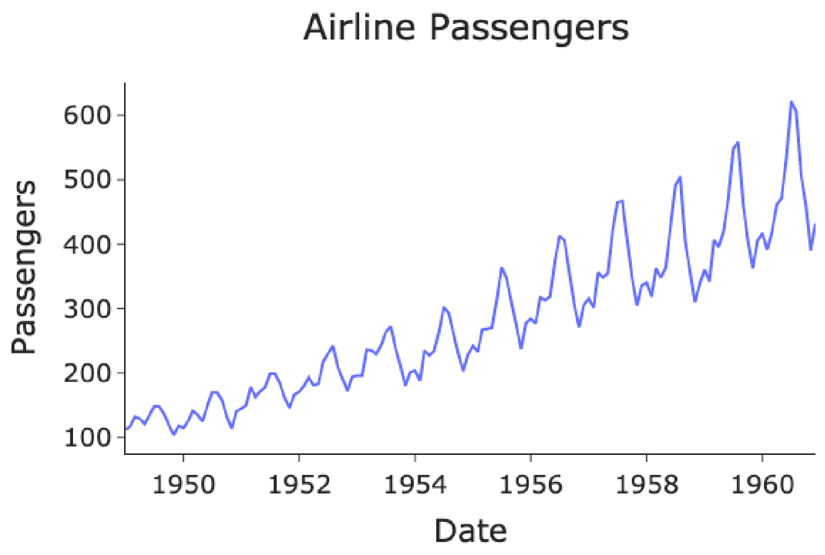
Components of a Time Series

Long-Term Trend: A general upward or downward movement.

Seasonal Patterns: Regular fluctuations within a specific period.

Cyclic Movements: Less predictable short-term cycles.

Random Fluctuations: Unpredictable noise or variations.



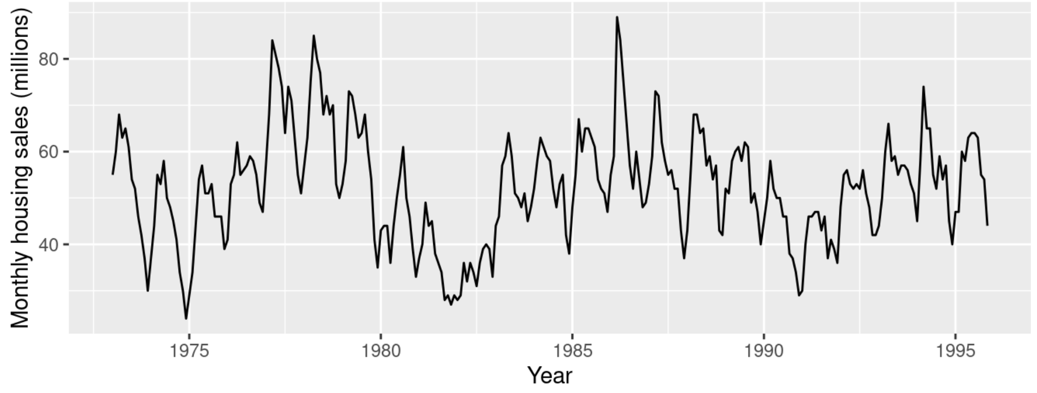
Components of a Time Series

Long-Term Trend: A general upward or downward movement.

Seasonal Patterns: Regular fluctuations within a specific period.

Cyclic Movements: Less predictable short-term cycles.

Random Fluctuations: Unpredictable noise or variations.



Difference Seansonal/cyclical: if the length of a cycle is fixed then it is seasonal. The length of each cycle is variable.

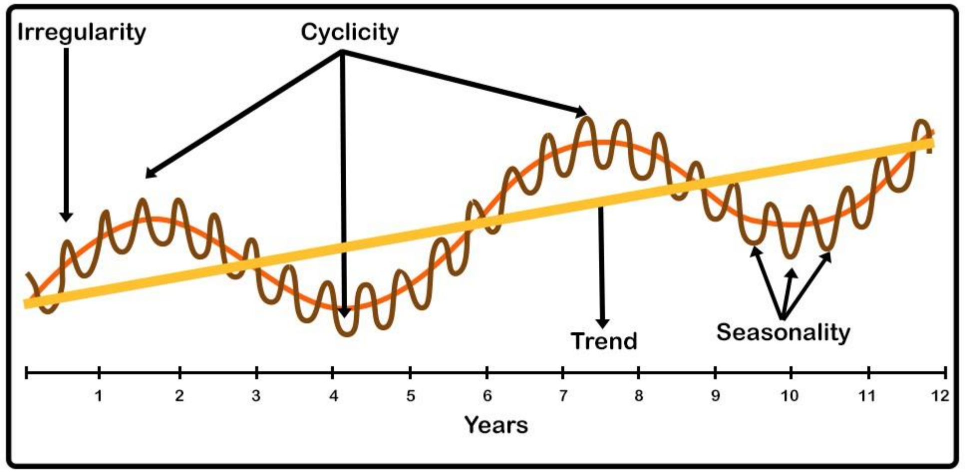
Components of a Time Series

Long-Term Trend: A general upward or downward movement.

Seasonal Patterns: Regular fluctuations within a specific period.

Cyclic Movements: Less predictable short-term cycles.

Random Fluctuations: Unpredictable noise or variations.



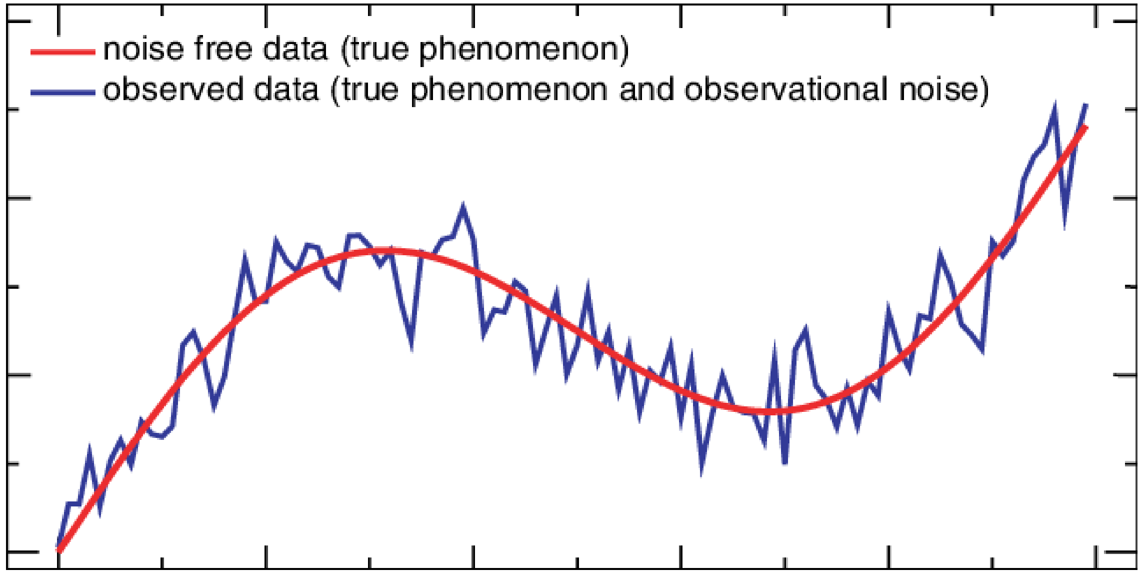
Signal vs noise

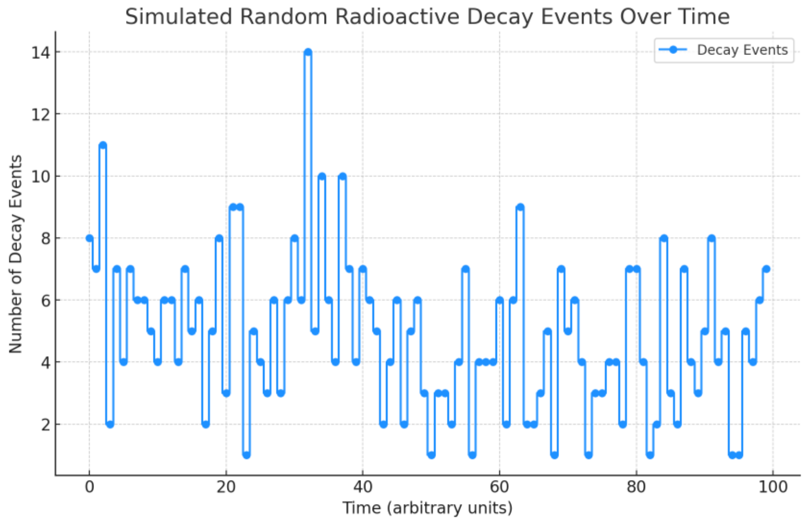
Signals: Deterministic components that can be extracted from the data: Trend, seasonality, cyclic movements.

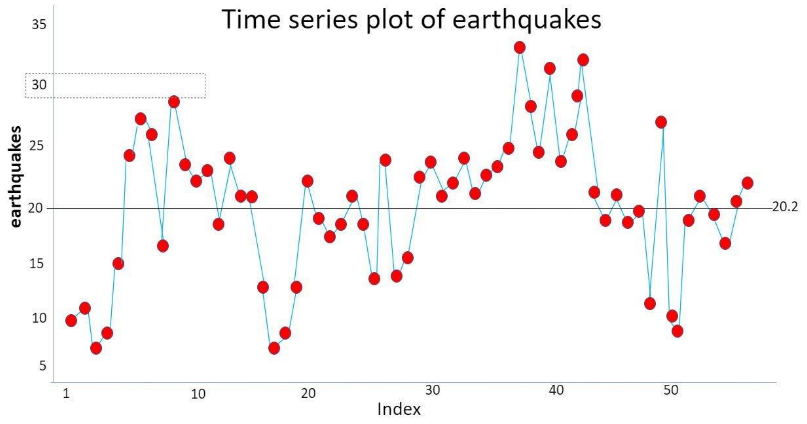
Noise: Random fluctuations that are difficult to predict. Independent of signals.

Predictability: Signals are more predictable than noise.

How to differentiate the earthquake case (signals+noise) from the decay events example (purely random process)?







Decomposing Time Series Data for Accurate Forecasting

- Component Identification: Crucial for building accurate forecasting models.
- Decomposition: Removing component effects from the data.
- Measurement: Quantifying the significance of each component.
- Feature Engineering: Using components as additional features.
- Recomposition: Adding components back to forecasted results.

Why removing trend and seasonality?

- **Isolating Underlying Trends**: Seasonality often masks the underlying trend and other important patterns in the data.

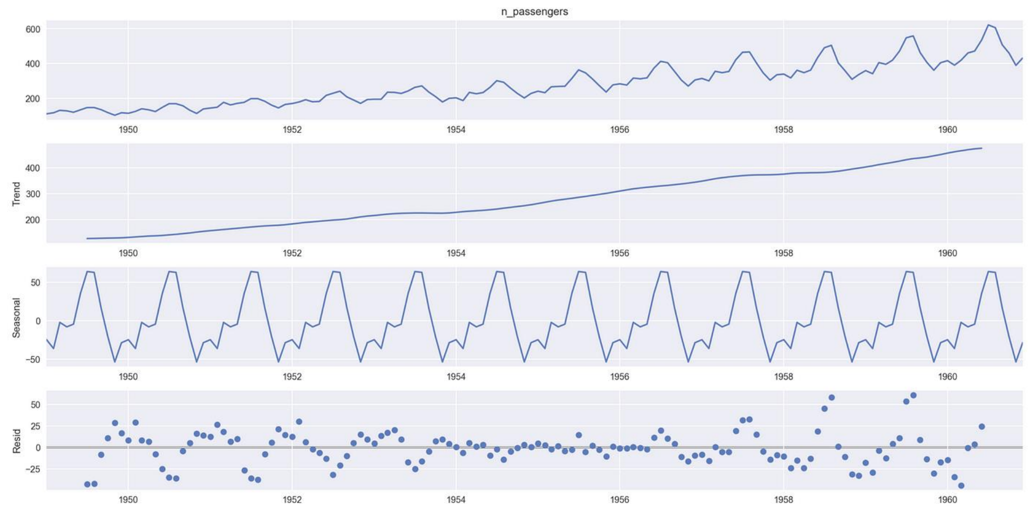
- **Focus on core patterns**: By removing these predictable patterns, you can focus on understanding and modeling the core dynamics of the data, such as unexpected changes or non-seasonal trends.

- **Enhancing Generalization**: Avoid overfitting to specific seasonal patterns.

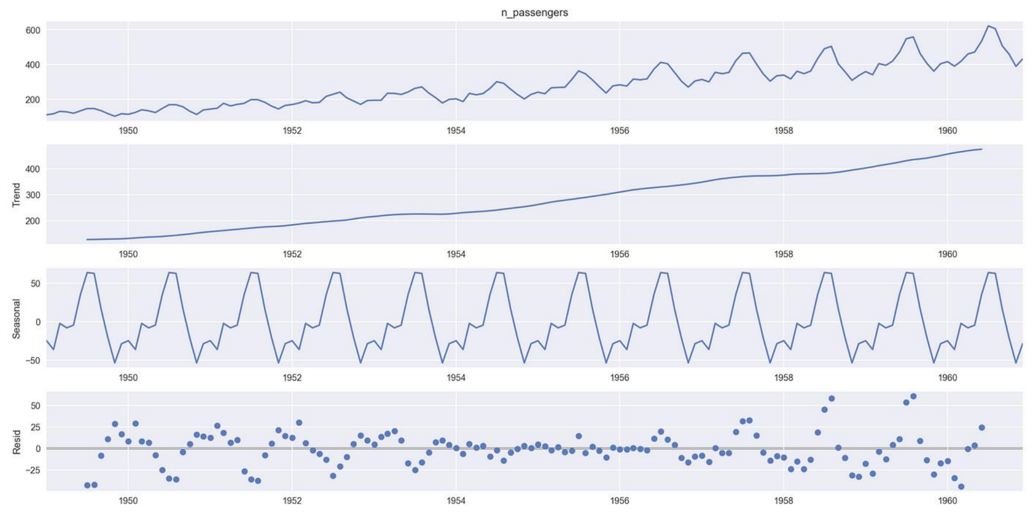
- **Simplifying Model Complexity**: Use simpler, more interpretable models.

- **Detecting Anomalies**: Easier to identify deviations from the norm.

- **Enabling Stationarity**: Satisfying assumptions for certain forecasting models.



Why removing trend and seasonality?



- **Isolating Underlying Trends**: Seasonality often masks the underlying trend and other important patterns in the data.

- **Focus on core patterns**: By removing these predictable patterns, you can focus on understanding and modeling the core dynamics of the data, such as unexpected changes or non-seasonal trends.

- **Enhancing Generalization**: Avoid overfitting to specific seasonal patterns.

- **Simplifying Model Complexity**: Use simpler, more interpretable models.

- **Detecting Anomalies**: Easier to identify deviations from the norm.

- **Enabling Stationarity**: Satisfying assumptions for certain forecasting models.

In some cases, keeping seasonality is important:

- **Forecasting with Seasonality**: Essential when seasonality is a key factor.

- **Using Seasonal Models**: Models like SARIMA and seasonal decomposition.

Stationarity

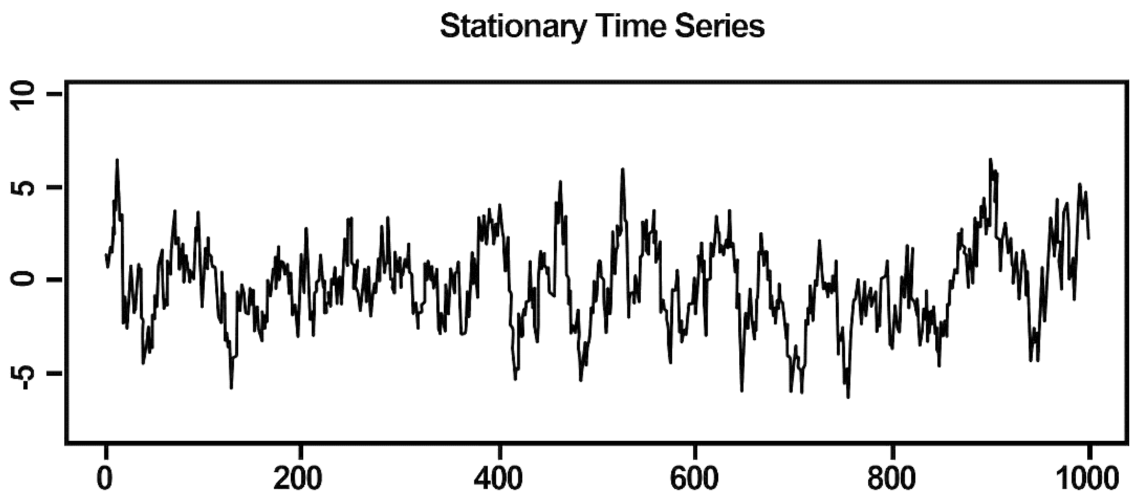
Definition: Statistical properties remain constant over time.
→ Examples of statistics: mean, variance…

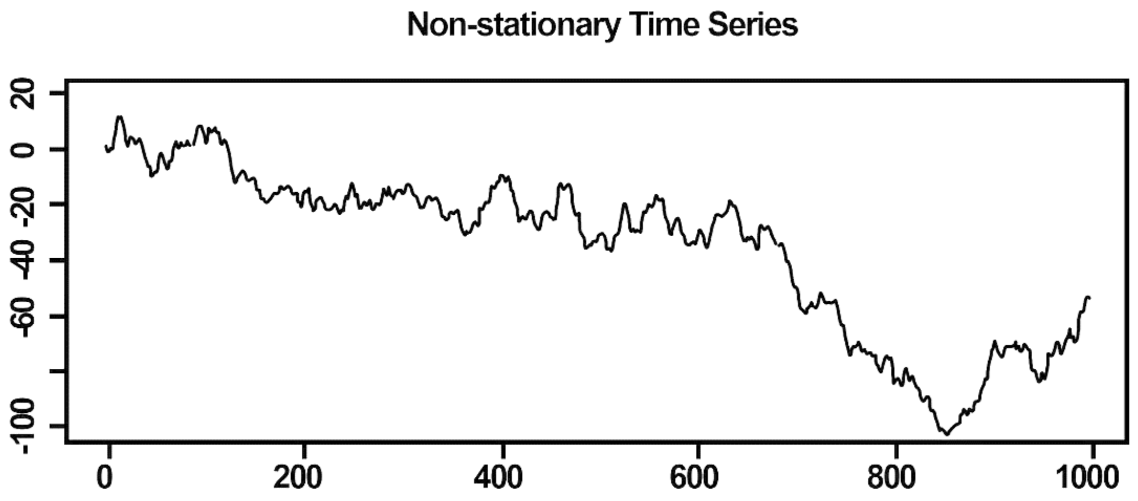
Benefits of Stationarity:

- **Easier analysis and modeling**: the basic assumption is that their properties are not dependent on time and will be the same in the future as they have been in the previous historical period of time.

Classically you should make your TS stationary

- **Meaningful sample statistics:** Many time series features assume stationarity and are useless unless the underlying data is stationary
→ the mean of a time series as a feature is practical only where the time series is stationary so that the idea of a mean makes sense.





Practical Considerations

Before building forecasting solutions, it is highly recommended to define the following forecasting aspects:

- **Inputs and Outputs**: Clearly define the data used for forecasting and the desired prediction results.

- **Granularity Level**: Determine the level of detail required in the data. The frequency at which time series values are collected. Always ask if the frequency used is adequate for the problem at hand.

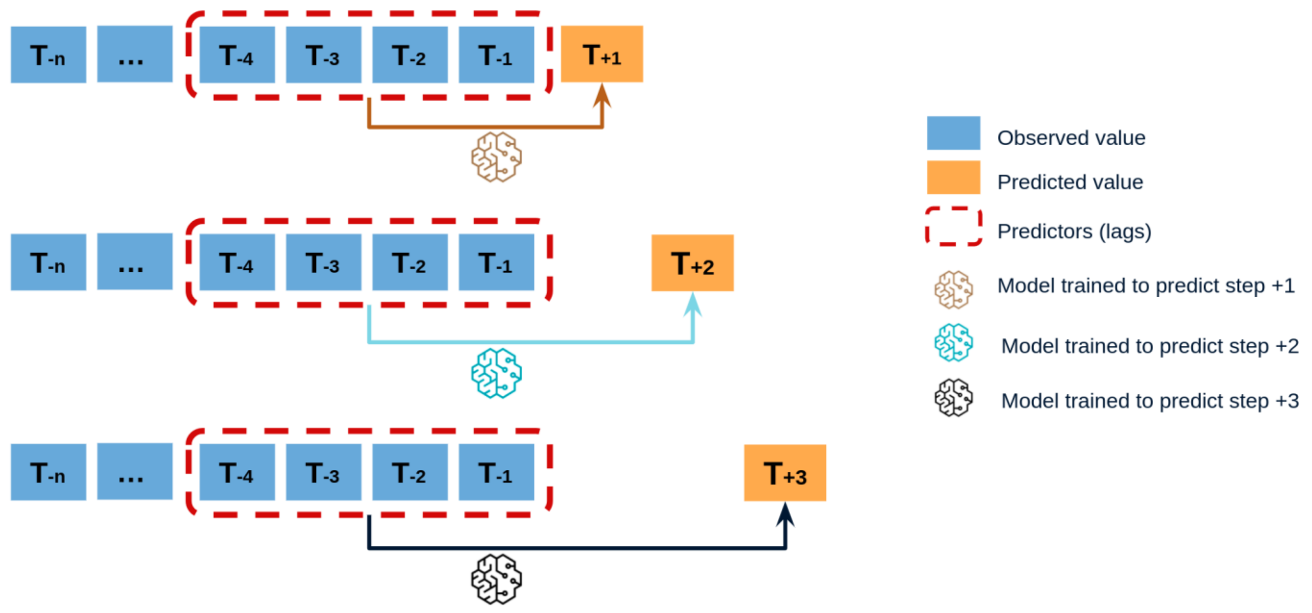
- **Horizon**: Specify the length of time into the future to be forecasted.
  - → Short term: less than three months, short-term objectives such as material requirement planning, scheduling, and budgeting
  - → Long term: objectives such as product diversification, sales, and advertising.

- **Endogenous and Exogenous Features**: Identify internal and external factors affecting the forecast.
  - → **Endogeneous feature**: Variables that are part of the system being modeled and have a direct relationship with the target variable. They are derived from the same time series data and reflect its internal structure. For forecasting stock prices, yesterday's price could be an endogenous feature
  - → **Exogeneous feature**: input variables that are not influenced by other variables in the system and on which the output variable depends, i.e the weather. They present common characteristics:
    - ▪ fixed when they enter the model
    - ▪ taken as a given in the model
    - ▪ They influence endogenous variables in the model without being influenced by them
    - ▪ They are not explained by the model

Practical Considerations

Single-Step or Multi-Step: Choose the appropriate forecasting structure

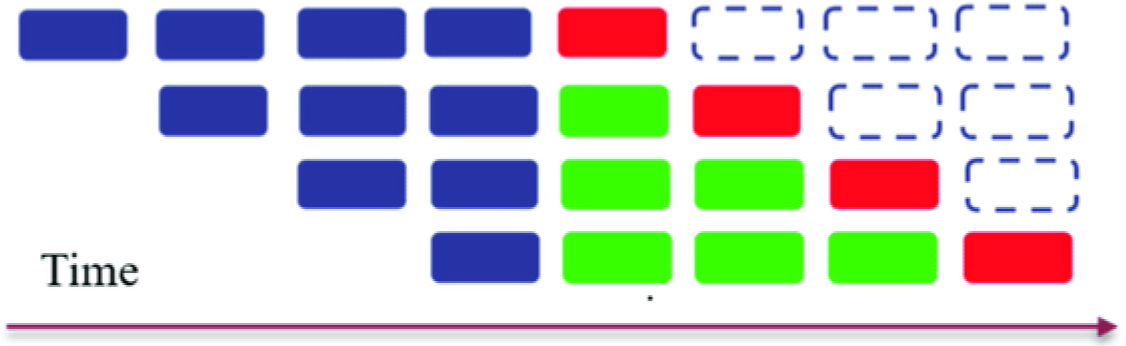
- **Direct multi-step forecast**: The direct method requires creating a separate model for each forecast time stamp.


T-n
...
T-4
T-3
T-2
T-1
T+1
T+2
T+3
Observed value
Predicted value
Predictors (lags)
Model trained to predict step +1
Model trained to predict step +2
Model trained to predict step +3

Practical Considerations

Single-Step or Multi-Step: Choose the appropriate forecasting structure

- **Direct multi-step forecast**: The direct method requires creating a separate model for each forecast time stamp.
- **Recursive multi-step forecast**: a single time series model is created to forecast next time stamp, and the following forecasts are then computed using previous forecasts.

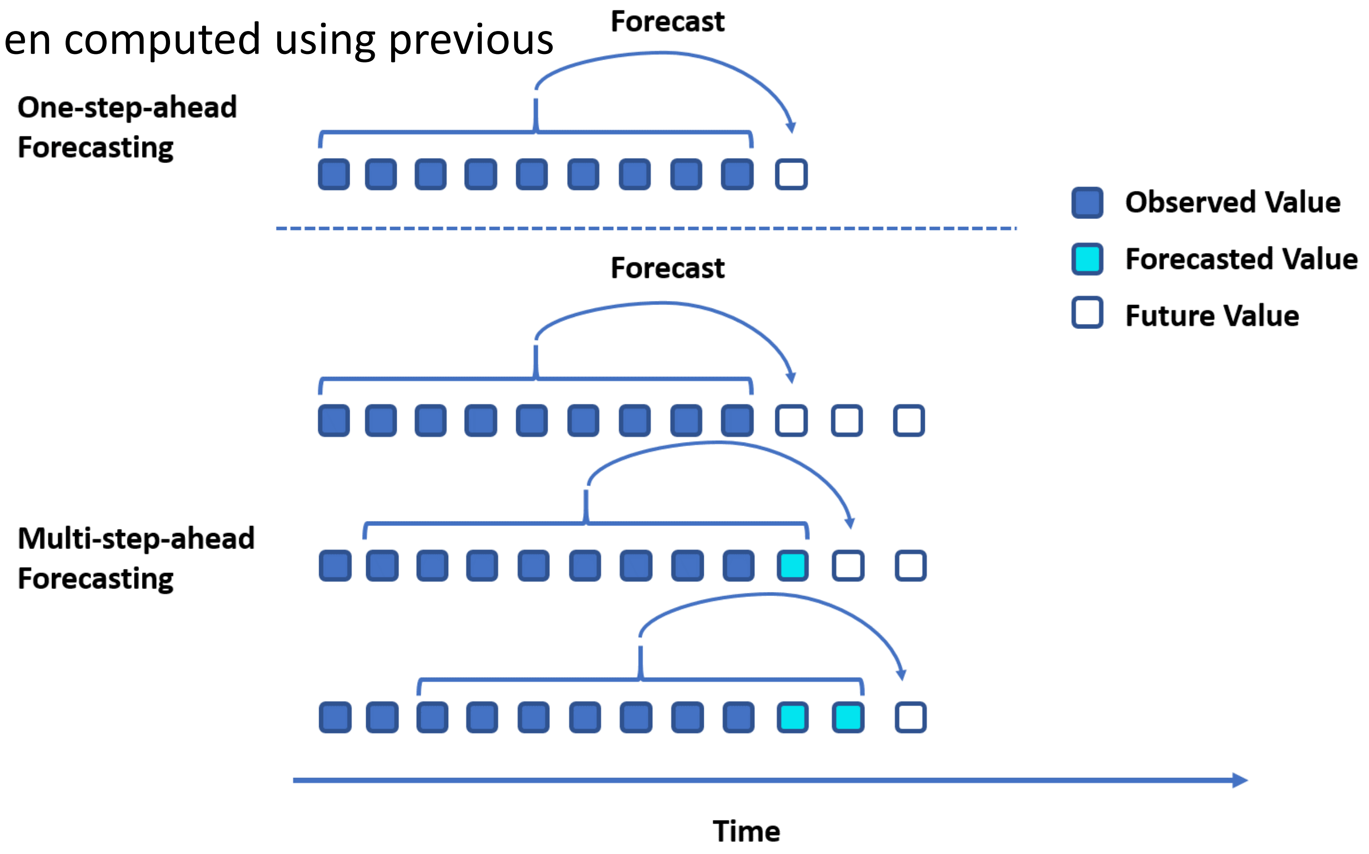


b. Multi-Step Ahead Forecasts

Practical Considerations

Single-Step or Multi-Step: Choose the appropriate forecasting structure

- **Direct multi-step forecast**: The direct method requires creating a separate model for each forecast time stamp.
- **Recursive multi-step forecast**: a single time series model is created to forecast next time stamp, and the following forecasts are then computed using previous forecasts.
- **Direct-recursive hybrid multi-step forecast**: The direct and recursive strategies can be combined to offer the benefits of both methods. For example, a distinct model can be built for each future time stamp, however each model may leverage the forecasts made by models at prior time steps as input values.

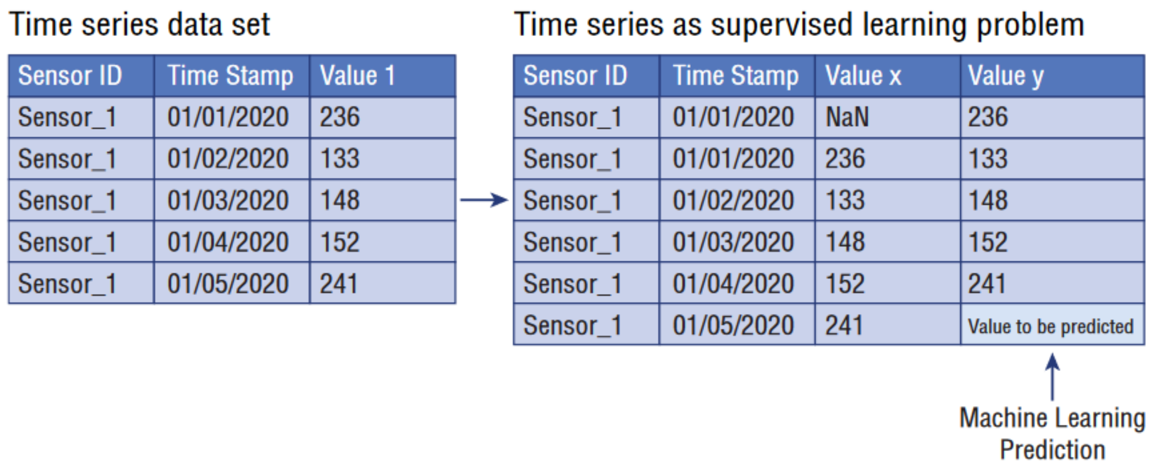


Forecasting Process

1. **Data Gathering**: Collect relevant data in the correct format.
2. **Data Processing and Cleaning**: Prepare and engineer features.
3. **Stationarity and Trend**: Analyze and handle these components.
4. **Supervised Learning Problem**: Convert the problem into a supervised learning format.
5. **Dataset Splitting**: Divide the data into training and testing sets.
6. **Model Application and Training**: Choose and train a suitable model.
7. **Evaluation**: Assess model performance using relevant metrics.
8. **Fine-Tuning**: Iterate and improve the model.

How to shape a TS into a supervised learning problem

Sliding window: exploiting previous time steps and using them as input and then leveraging the next time step as output of the model



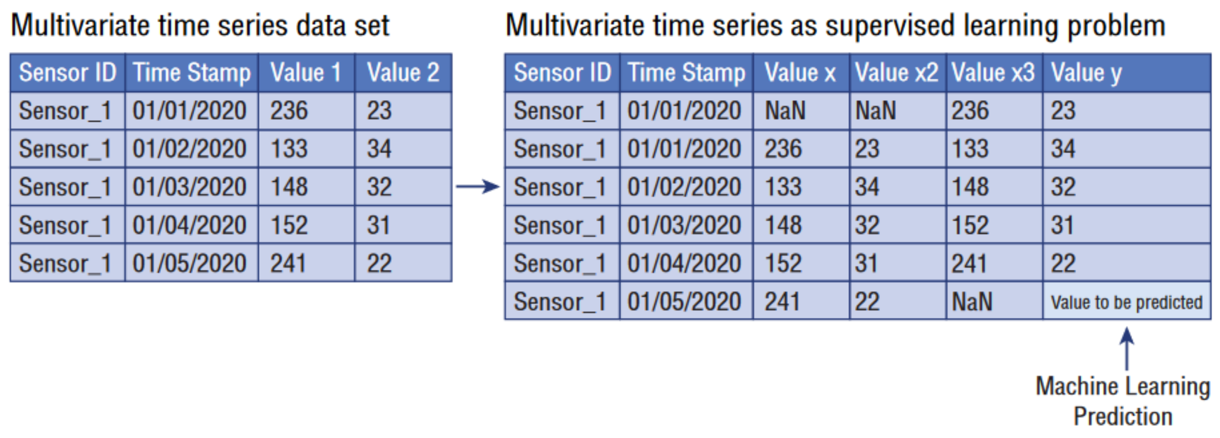
Time series data set

Sensor ID	Time Stamp	Value 1
Sensor_1	01/01/2020	236
Sensor_1	01/02/2020	133
Sensor_1	01/03/2020	148
Sensor_1	01/04/2020	152
Sensor_1	01/05/2020	241

Time series as supervised learning problem

Sensor ID	Time Stamp	Value x	Value y
Sensor_1	01/01/2020	NaN	236
Sensor_1	01/01/2020	236	133
Sensor_1	01/02/2020	133	148
Sensor_1	01/03/2020	148	152
Sensor_1	01/04/2020	152	241
Sensor_1	01/05/2020	241	Value to be predicted

Machine Learning Prediction



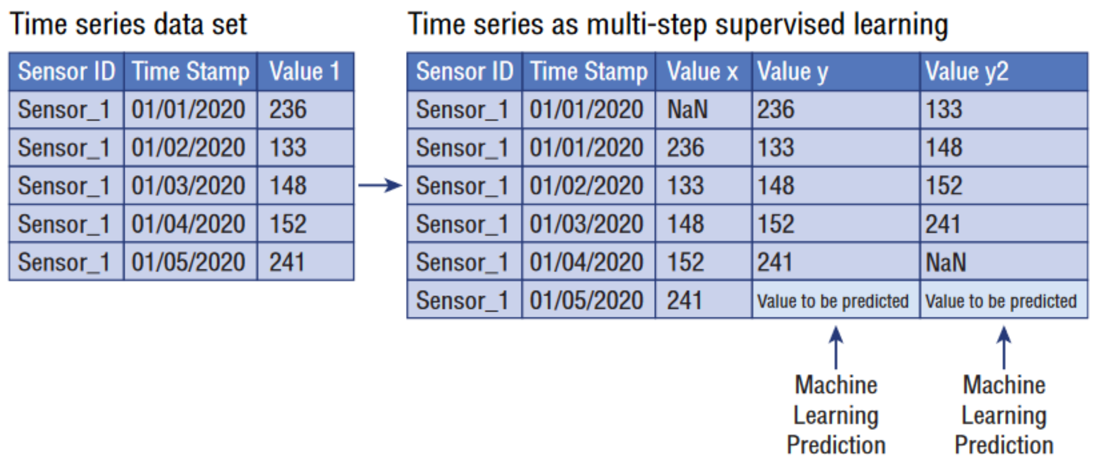
Multivariate time series data set

Sensor ID	Time Stamp	Value 1	Value 2
Sensor_1	01/01/2020	236	23
Sensor_1	01/02/2020	133	34
Sensor_1	01/03/2020	148	32
Sensor_1	01/04/2020	152	31
Sensor_1	01/05/2020	241	22

Multivariate time series as supervised learning problem

Sensor ID	Time Stamp	Value x	Value x2	Value x3	Value y
Sensor_1	01/01/2020	NaN	NaN	236	23
Sensor_1	01/01/2020	236	23	133	34
Sensor_1	01/02/2020	133	34	148	32
Sensor_1	01/03/2020	148	32	152	31
Sensor_1	01/04/2020	152	31	241	22
Sensor_1	01/05/2020	241	22	NaN	Value to be predicted

Machine Learning Prediction



Time series data set

Sensor ID	Time Stamp	Value 1
Sensor_1	01/01/2020	236
Sensor_1	01/02/2020	133
Sensor_1	01/03/2020	148
Sensor_1	01/04/2020	152
Sensor_1	01/05/2020	241

Time series as multi-step supervised learning

Sensor ID	Time Stamp	Value x	Value y	Value y2
Sensor_1	01/01/2020	NaN	236	133
Sensor_1	01/01/2020	236	133	148
Sensor_1	01/02/2020	133	148	152
Sensor_1	01/03/2020	148	152	241
Sensor_1	01/04/2020	152	241	NaN
Sensor_1	01/05/2020	241	Value to be predicted	Value to be predicted

Model Evaluation

Mean Absolute Error (MAE)

$$\text{MAE} = \frac{1}{N}\sum_{t=1}^{n} |y_t - \hat{y}_t|$$

- **Scale-Dependent**: MAE is expressed in the same units as the original data, which makes it intuitive and easy to interpret. For example, if you are forecasting temperature in degrees Celsius, the MAE will also be in degrees Celsius.

- **Non-Sensitive to Error Magnitude and robust to outliers**: MAE treats all errors equally, without giving more weight to larger errors. This makes it less sensitive to outliers

Limitations:

• **Does Not Penalize Larger Errors More Heavily**: Unlike metrics like MSE or RMSE, MAE does not square the errors, so it does not give extra weight to larger errors. In situations where large errors are more problematic (e.g., in critical applications like medicine or finance), MAE might not be the best choice.

• **Not Differentiable Everywhere**: MAE is not differentiable at zero, which can be a limitation when using gradient-based optimization techniques in machine learning models (such as neural networks).

• **Scale-Dependent**: Since MAE is in the same unit as the data, it cannot be used to compare errors across different datasets with different units or scales. In such cases, scale-independent metrics like Mean Absolute Percentage Error (MAPE) or Mean Squared Logarithmic Error (MSLE) may be more appropriate.

Mean Squared Error (MSE)

$$\mathrm{MSE} = \frac{1}{N}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2$$

- **Scale-Dependent**: SE is expressed in squared units of the original data. For example, if you are predicting temperatures in degrees Celsius, the MSE will be in degrees Celsius squared.
- **Sensitive to Outliers**
- **Smooth and Differentiable**: MSE is differentiable and has a continuous gradient, which makes it suitable for optimization algorithms, particularly in machine learning models that use gradient descent

RMSE

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}$$

- **Error is easily interpretable:** RMSE is expressed in the same units as the original data
- **Less sensitive to outliers**
- **Scale-Dependent**: Since RMSE is in the same unit as the data, it cannot be used to compare errors across different datasets with different units or scales. In such cases, scale-independent metrics like Mean Absolute Percentage Error (MAPE) or Mean Squared Logarithmic Error (MSLE) may be more appropriate.

Mean Absolute Percentage Error (MAPE)

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

- **Scale-Independent**: MAPE is expressed as a percentage, making it scale-independent and useful for comparing forecast accuracy across different datasets, models, and units of measurement.
- **Interpretable**: Since MAPE is presented as a percentage, it is easy to understand and communicate. For example, a MAPE of 5% means that, on average, the model's predictions are off by 5% from the actual values.
- **Equal Weight to All Errors:** MAPE treats all errors equally in terms of their percentage deviation, regardless of the magnitude of the actual values. Each error is normalized by the actual value, which provides a consistent measure across different scales.

Limitations:

- **Sensitive to Small Denominators**
- **Bias Towards Underestimation**: MAPE can introduce bias when the actual values are small because it will penalize overestimates more than underestimates. For example, if the actual value is 1 and the prediction is 2, the error is 100%, but if the actual value is 2 and the prediction is 1, the error is only 50%.

Symetric MAPE

$$\text{sMAPE} = \frac{1}{n}\sum_{t=1}^{n}\frac{|y_t - \hat{y}_t|}{(|y_t| + |\hat{y}_t|)/2} \times 100$$

Unlike MAPE, which can be biased when the actual values are very small or when there is a significant difference between the actual and forecasted values, sMAPE adjusts for these biases by using a symmetric formula.

Limitations:

Range Between 0% and 200%: Unlike MAPE, which ranges from 0% to infinity, sMAPE is constrained between 0% and 200%. This range limitation can make sMAPE less intuitive when dealing with exceptionally large deviations.

Mean Absolute Scaled Error (MASE)

$$\text{MASE} = \frac{\frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t|}{\frac{1}{n-1}\sum_{t=2}^{n}|y_t - y_{t-1}|}$$

MASE addresses some of the limitations of traditional error metrics by scaling the absolute errors relative to a naive forecasting method, typically the one-step-ahead in-sample forecast using the mean or the previous observation

- **Benchmark Comparison**: MASE explicitly compares the forecasting model's performance to a naïve benchmark, providing insight into whether the model offers a meaningful improvement over simple forecasting techniques.
- **Interpretable**: MASE values are straightforward to interpret: a MASE of 1 indicates that the model performs as well as the naïve benchmark, values less than 1 indicate better performance, and values greater than 1 indicate worse performance.
- **Handles Intermittent and Seasonal Data**: MASE can handle time series with intermittent or seasonal data better than other metrics like MAPE because it does not involve division by actual values, which can be problematic when actual values are close to zero.

Limitations:

- **Dependence on Benchmark**: MASE is dependent on the choice of the naïve benchmark. If the benchmark is poorly chosen o not representative, the MASE value might be misleading.
- **Interpretation with Non-Stationary Data**: For non-stationary data, where the mean or variance changes over time, the naïve benchmark might not be appropriate, leading to a distorted MASE value

Data Cleaning and Smoothing

Upsampling and downsampling

Downsampling: Subsetting data to a lower frequency.

- Example: you may be measuring something too often. Suppose you have a data set where someone had measured the outside air temperature every second → Not informative

Upsampling: Representing data at a higher frequency. **It is not the inverse of downsampling.**

- Trying to get data that was not measured → No free lunch. Adding more time labels but not more information
- Example: Irregular time series, knowledge of dynamics.

How to handle missing values

Causes in healthcare:

- Patient non-compliance
- health status
- medical errors
- device malfunctions
- data entry errors.

Methods to address the issue:

- **Imputation**: filling missing data based on observations about the entire data set (i.e replacing with the mean value)
- **Interpolation**: using neighboring data points to estimate the missing value. Interpolation can also be a form of imputation.
- **Deletion:** not using time periods that have missing data at all

We will focus on preserving data, whereas a method such as deleting time periods with missing data will result in less data for your model.

A focus on imputation methods

Forward Fill: Carrying forward the last known value prior to the missing one. (avoid lookaheads)

Backward Fill: Propagating values backward.

Moving Average: Using a rolling mean or median. Similar to a forward fill, however, you are using input from multiple recent times in the past.

Interpolation: Estimating missing values using neighboring data points.

A note on lookahead:

- **Definition:** Using future information to influence model behavior → You are not supposed to have such future information
- **Consequences**: Inaccurate predictions and biased models.
- **Prevention**: Use only past data for training and evaluation.

Time for Exercise 1

A focus on Moving Average

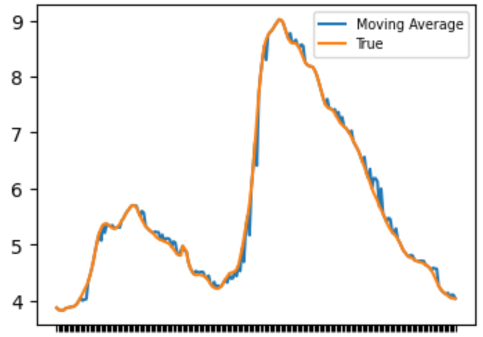
Moving average does not have to be arithmetic:

- **Exponential**: exponentially weighted moving averages gives more weight to recent data than to past data
- **Geometric**: helpful for time series that exhibit strong serial correlation and in cases where values compound over time.

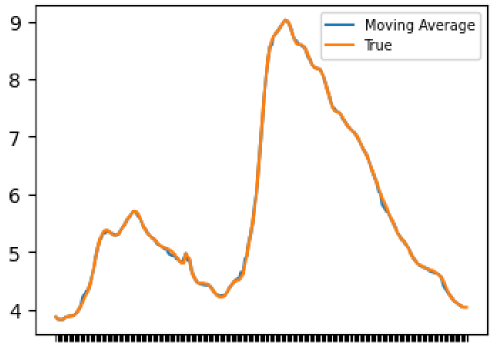
Always be cautious about lookahead!

If not concerned, the best estimate is always the one including lookahead.

NEVER use the global mean, that's lookahead !



Forward MA



Centered MA

Rolling mean data imputation reduces variance in the data set.

→accuracy, R^2 statistics, or other error metrics will overestimate your model's performance

Smoothing Data

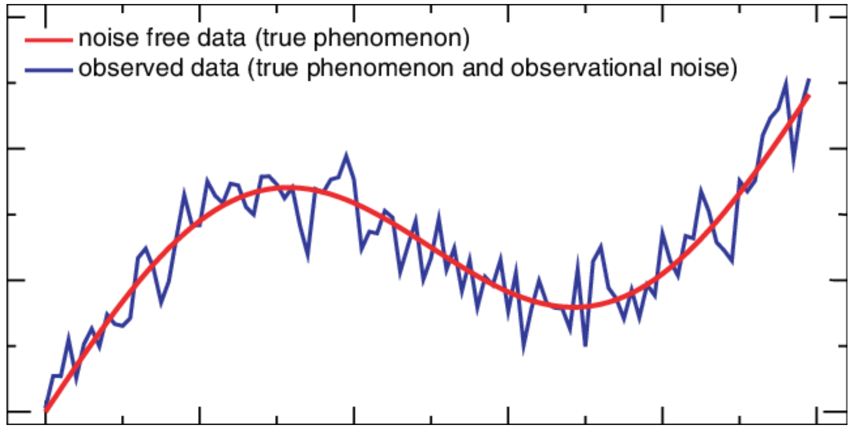
Purpose: Reduces noise to reveal underlying patterns & trends in data.

Common Smoothing Method:

- Moving Averages (MA): can be simple or weighted
- Exponential MA (EMA): assigns exponentially decreasing weights to older observations, giving more importance to recent observations.
  → useful when you want to smooth data but still respond quickly to recent changes
  → used for forecasting and analyzing time series with a smooth trend and seasonality.
- Kernel Smoothing: weights are determined by a kernel function (usually Gaussian)
- LOESS (Local Regression): fits local polynomial models to capture nonlinear trends.
  → Ideal for handling data with complex relationships and for exploratory analysis where you expect nonlinear trends or seasonality

Challenges:

- May lose important detail
- Requires careful parameter choice



A focus on exponential smoothing

Purpose: assigns exponentially decreasing weights to older observations, giving more importance to recent observations

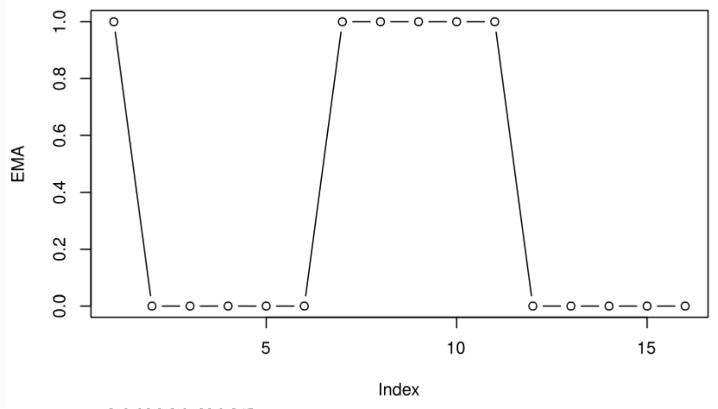
For a given time period t, the smoothed value of a series are computed using:

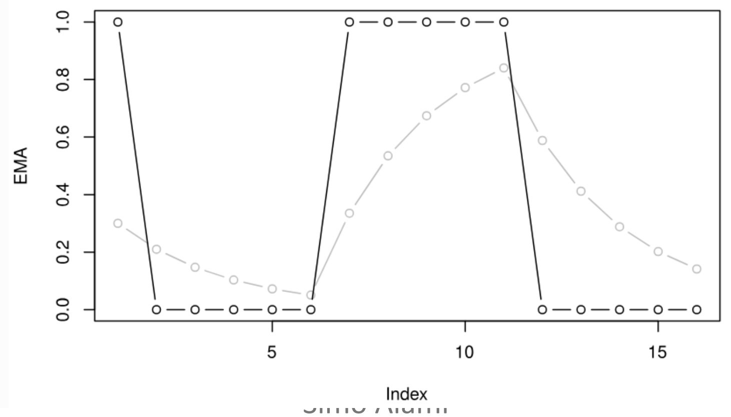
$$\boxed{\begin{gathered} S_t = (d^3 x_{t-3} + d^2 x_{t-2} + d x_{t-1}) / \sum d^i \\ S_t = d \times S_{t-1} + (1-d) x_t \end{gathered}}$$

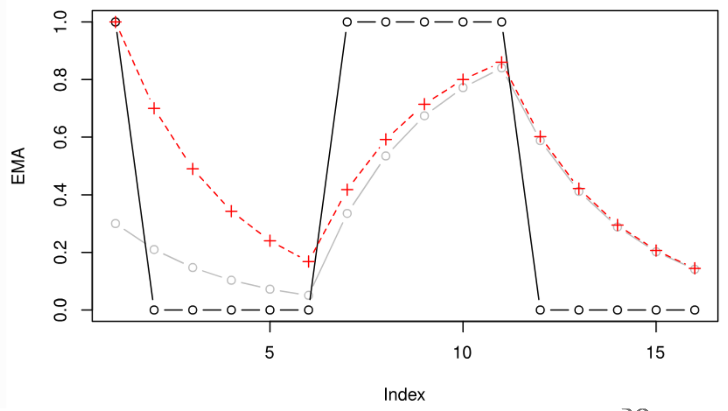
Given a series in the form of 3 then 6 with d=0.7, you would compute the second point in your series as:

$$3 \times 0.7 + 6 \times (1 - 0.7) = 3.9$$

If 3 is the first value of your time series, this is **Wrong.**
Why ? multiplying your discounting factorof 0.7 by 3 implicitly assumes that 3 is the sum of your knowledge going back in time to infinity, rather than a mere single data point you just measured.
→ It unduly weighs 3 relative to 6 as though 3 has a significantly more longstanding set of knowledge than it actually does.







Computing EMA correctly (1/3)

The correct approach is to actively account for how much data has gone into the EMA, versus how much of the EMA's value is from phantom data before our samples arrived

Let's expand the EMA formula over multiple time steps to see how the current EMA is influenced by past data points.

$$EMA_t = (1-r) \cdot x_t + r \cdot EMA_{t-1}$$

$$EMA_t = (1-r) \cdot x_t + r \cdot [(1-r) \cdot x_{t-1} + r \cdot EMA_{t-2}]$$

$$\mathrm{EMA}_t = (1-r) \cdot x_t + r \cdot (1-r) \cdot x_{t-1} + r^2 \cdot (1-r) \cdot x_{t-2} + \dots$$

$$EMA_t = (1-r) \sum_{i=0}^{t-1} r^i \cdot x_{t-i}$$

early in the process (when t is small), the sum of the weights $\sum_{i=0}^{t-1} r^i$ does not yet total 1, which means that the EMA is biased toward zero.

Computing EMA correctly (2/3)

why should the sum of weights total 1?

In any weighted average, the sum of the weights should equal 1 to ensure that the average properly reflects the contribution of each data point.
Example: in a simple arithmetic average of three numbers, each number is weighted equally, and the total weight is 1/3+1/3+1/3=1

$$\mathrm{EMA}_t = (1-r)\cdot x_t + r\cdot(1-r)\cdot x_{t-1} + r^2\cdot(1-r)\cdot x_{t-2} + \ldots$$

→ Each data point x has a weight attached to it

The total weight applied to the data points in the EMA is given by the sum of this infinite series:

$$(1-r) + r\cdot(1-r) + r^2\cdot(1-r) + \ldots \quad = \quad \sum_{i=0}^{\infty} r^i\cdot(1-r) = (1-r)\cdot\sum_{i=0}^{\infty} r^i = (1-r)\cdot\frac{1}{1-r} = 1$$

Computing EMA correctly (3/3)

When you start calculating the EMA, especially at the very beginning (e.g., at time t=1), there haven't been enough iterations for the weights to fully sum to 1.

→ the initial EMA tends to be biased toward zero.

After the first point, the EMA is simply: $EMA_1 = (1-r)\cdot x_1$ → the sum of the weights is only 1-r, which is less than 1

As more data points are added, the sum of the weights approaches 1, but during the initial stages, the sum is smaller, causing the EMA to be lower than it should be.

Correction: EMA is divided by the cumulative sum of weights up to time t

The sum of the weights is: $(1-r)\cdot\sum_{i=0}^{t-1} r^i = (1-r)\cdot\frac{1-r^t}{1-r} = 1-r^t$

$$\boxed{\text{Corrected EMA}_t = \frac{\text{EMA}_t}{1-r^t}}$$

Time for Exercise 2

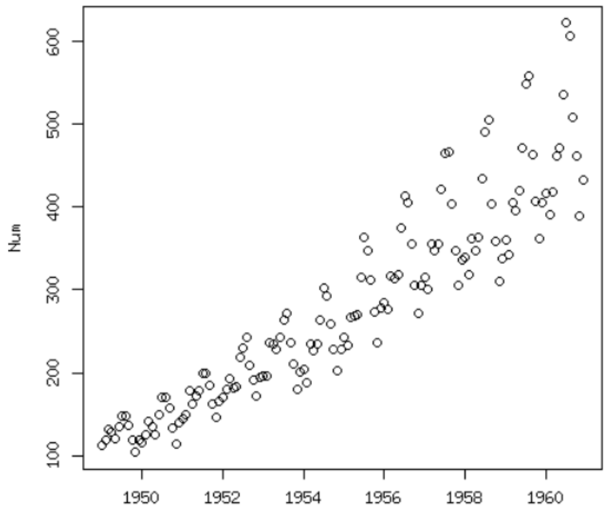
Visualization and Transformations

Seasonality

Definition: any kind of recurring behavior in which the frequency of the behavior is stable. It can occur at many different frequencies at the same time.

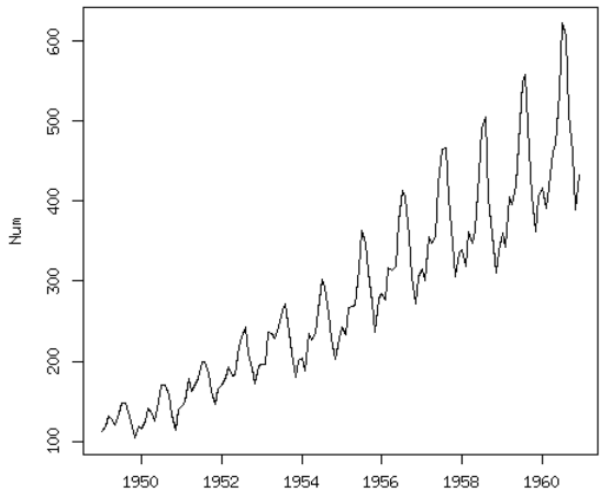
→ Identifying and addressing seasonality is crucial for accurate forecasting.

→ Seasonality # cycle: Seasonality displays fixed frequency while cycles exhibit variable periods (i.e business cycles, volcan eruptions



A scatter plot of yearly airline passenger count.
→ No visible seasonality

Turn into a linear plot



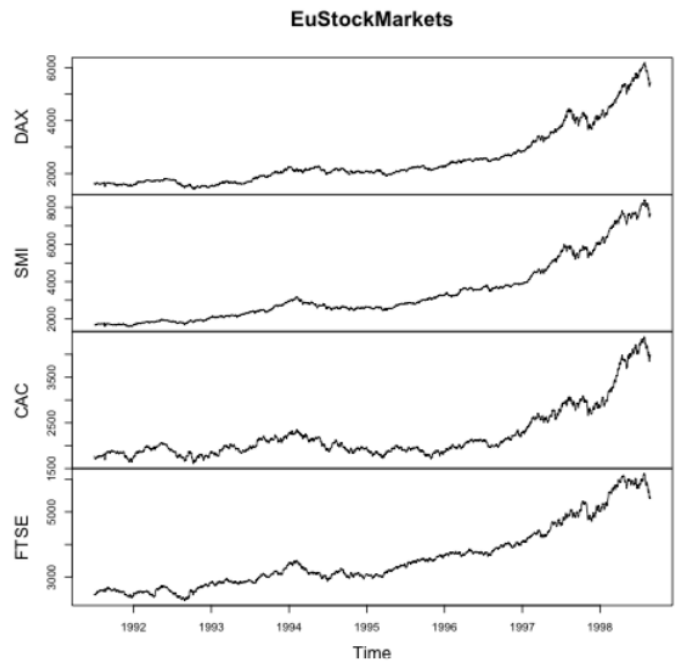
Seasonality is obvious

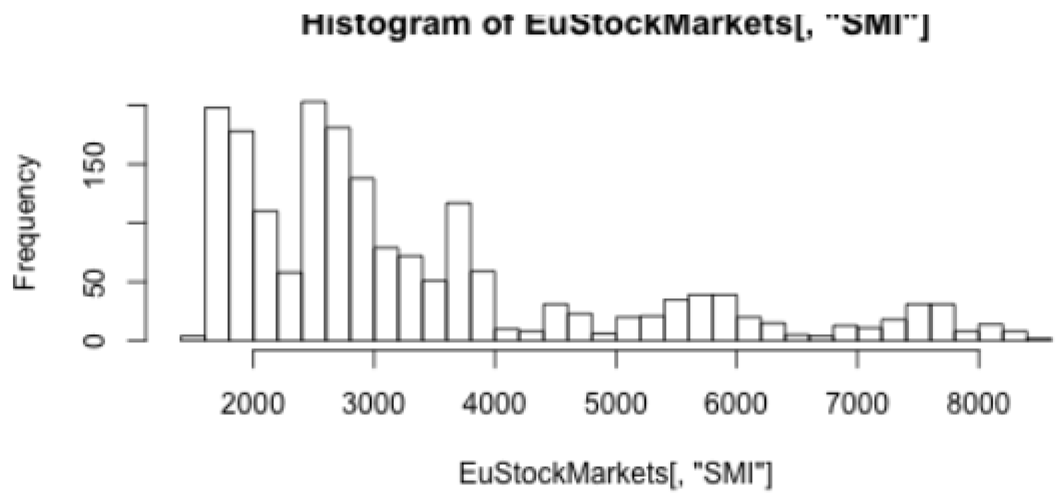
Making the right plot is part of time series expertise

With human behavior data there is almost always some form of seasonality, even with several cycles (an hourly pattern, a weeklypattern, a summer-winter pattern, etc.)

Scatter plots vs Histograms

Plot from EuStockMarkets: What can you observe?



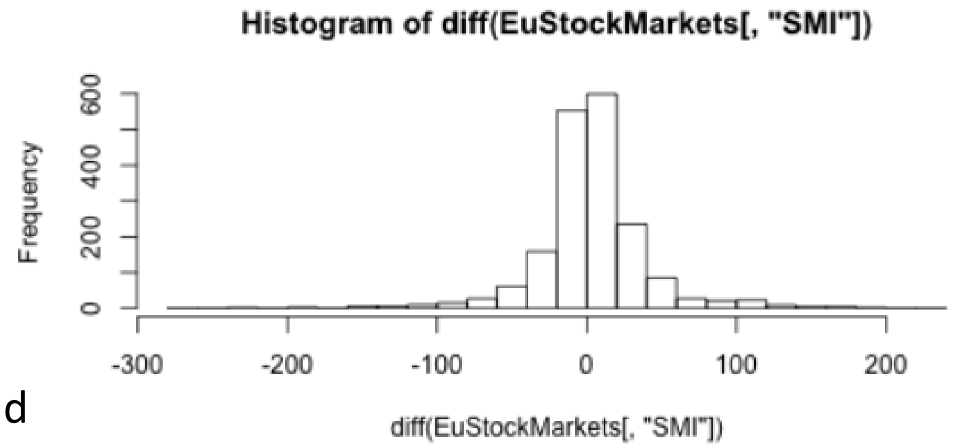


Data differencing

Variation is more interesting than actual values

Definition: Computing the differrence between a value and the next one in the time series

→ Removes trend
→ Data are actually bell shaped, i.e. Normal Distribution



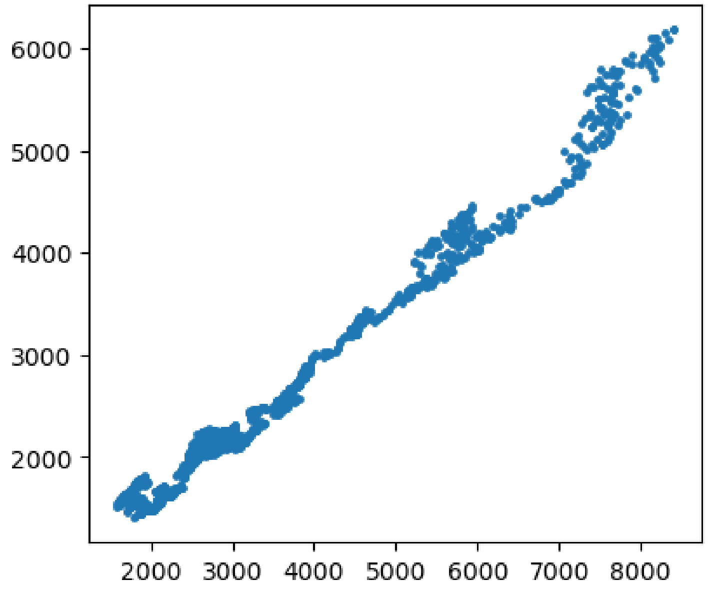
The values of the time series have gone both up (positive difference values) and down (negative difference values) about the same amount over time.

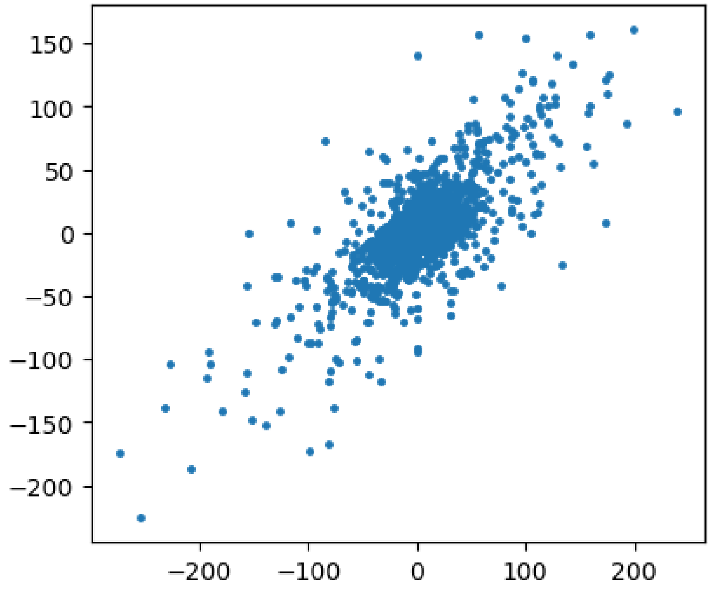
→ slight bias in favor of positive over negative differences is what accounts for that trend.

Scatter plots are still useful

Use scatter plots to determine both how two stocks are linked at a specific time and how their price shifts are related over time.

Original Data





Differenced Data

What can you observe?

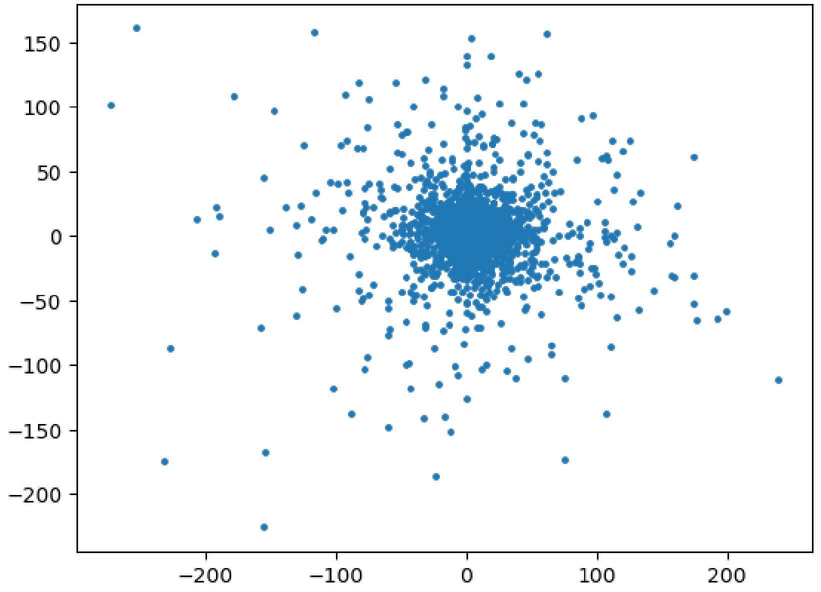
Correlation !

Is it useful?

No

Correct Correlation using scatter plot

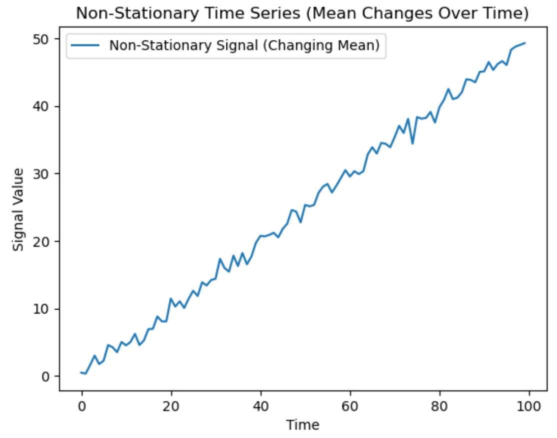
find out whether the change in one stock earlier in time can predict the change in another stock later in time.



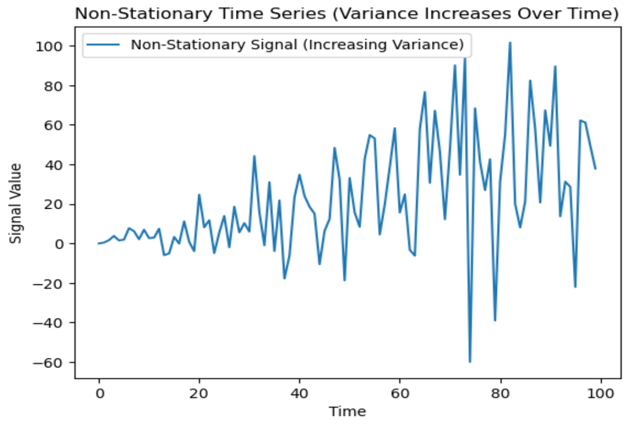
No correlation. We will see later how to detect true correlations

Stationarity

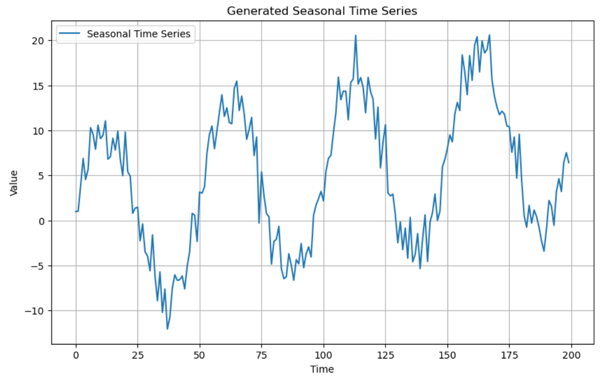
These time series are not stationary because of



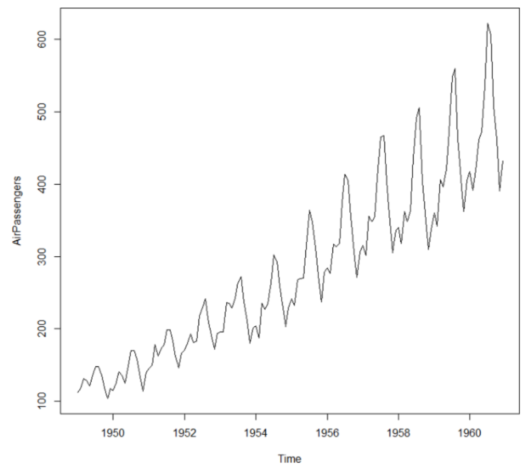
Trend, i.e changing mean



Variance

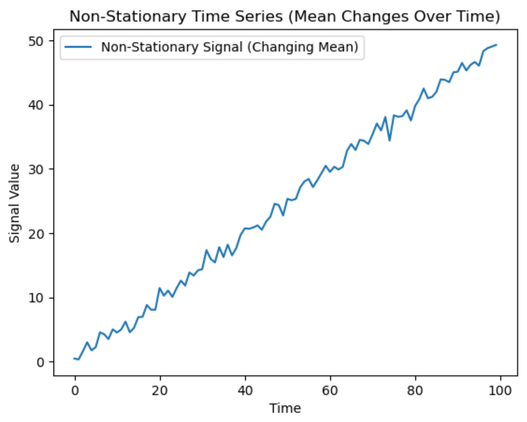


Seasonality

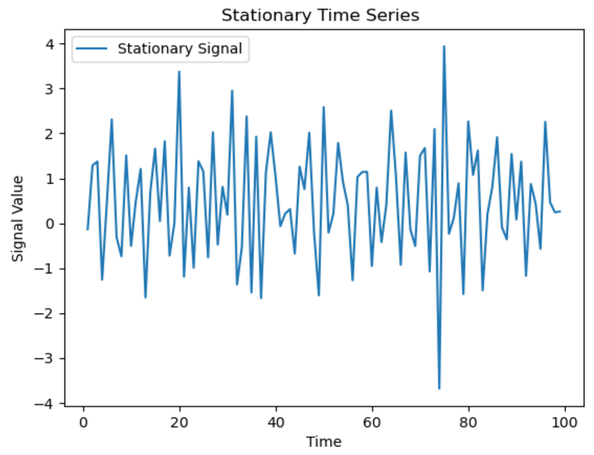


All of them

Handling Trends



Differentiate

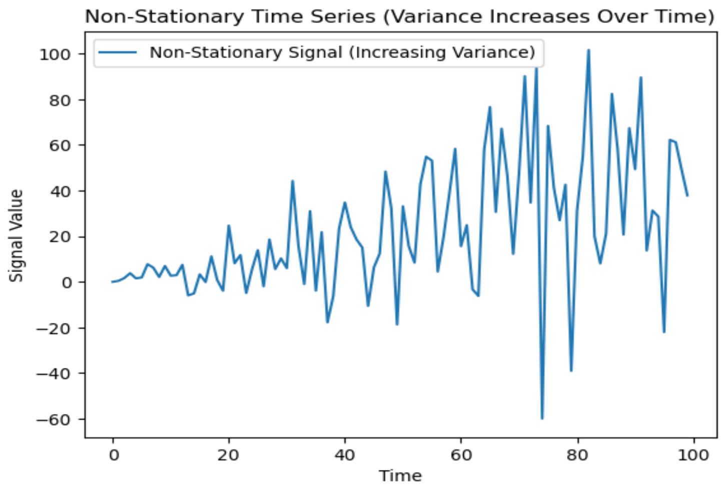


Suppose $y_t = \beta_0 + \beta_1 t + \epsilon_t$, with $\epsilon \sim \mathcal{N}(0, \sigma)$

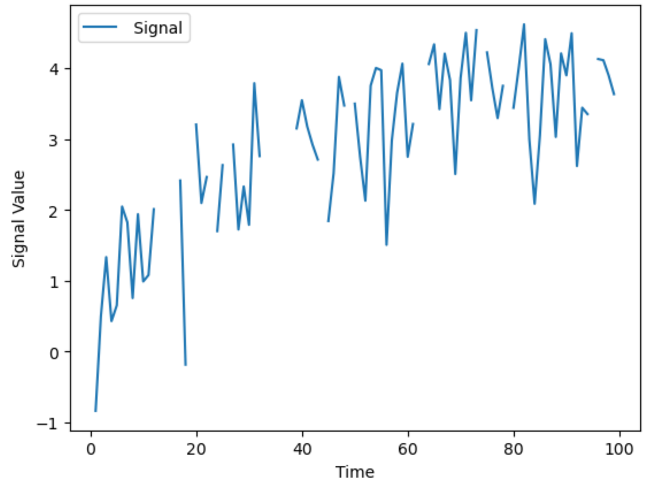
$$\begin{aligned} Z_t = y_t - y_{t-1} &= \beta_0 + \beta_1 t + \epsilon_t - \beta_0 - \beta_1(t-1) - \epsilon_{t-1} \\ &= \beta_1(t - t + 1) + (\epsilon_t - \epsilon_{t-1}) \\ &\rightarrow \mathbb{E}(Z_t) = \beta_1, \quad V(Z_t) = 2\sigma^2 \end{aligned}$$

⚠ Sometimes you may have to differentiate more than once

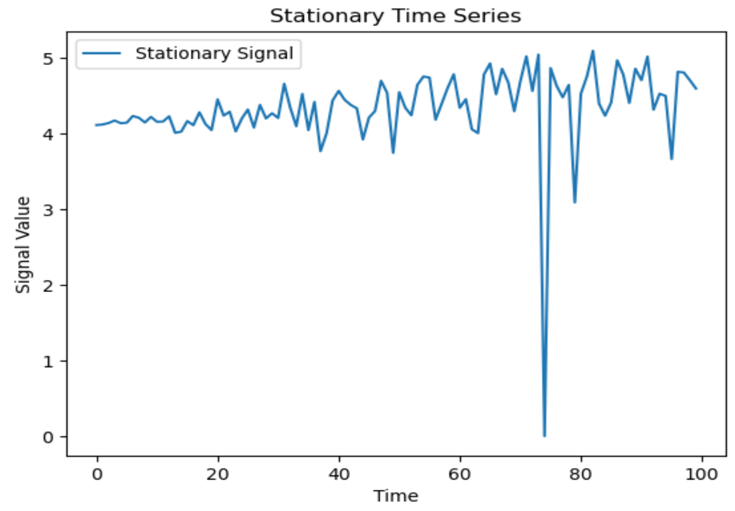
Handling Heteroscedasticity



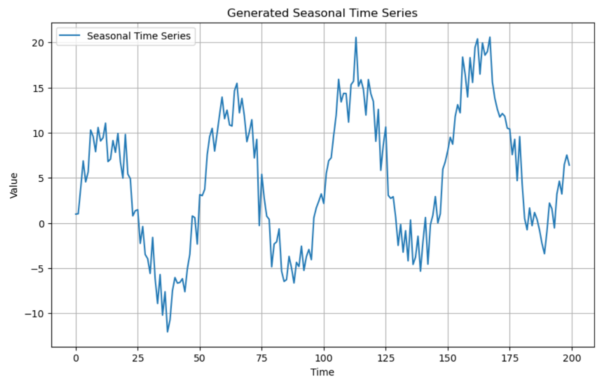
Use log or Sqrt



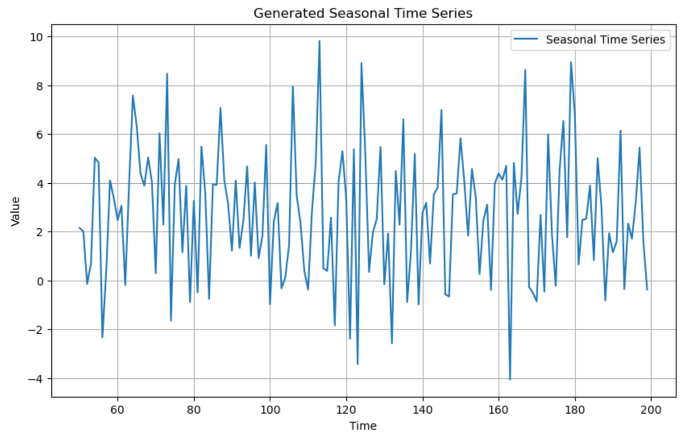
Shift time series by min+1 then apply log



Handling Seasonality



Differentiate by period length



Other transformations

Time for Exercise 3

Stationarity is not the only assumption forecasting models make. Another common but distinct assumption is the normality of the distribution of the input variables or predicted variable.

Box Cox transformation: makes non-normally distributed data (skewed data) more normal.

$$y(\lambda) = \begin{cases} \frac{y^\lambda - 1}{\lambda}, & \text{if } \lambda \neq 0 \\ \log(y), & \text{if } \lambda = 0 \end{cases}$$

Just because you can transform your data doesn't mean you should
→Make sure that transformations preserve the most important information.

Even log and sqrt make underlying assumptions.

- data should be positive,
- larger values become less different from one another, effectively compressing the space between larger values but not between smaller values, deemphasizing the differences between outliers

⚠ Once your model is learned, and predicctions done, **NEVER** forget to invert all the transformations you've done in pre-processing → Always use invertible transformations !