

# Machine Learning for Time Series

Dr. Emmanuelle Salin

Machine Learning and Data Analytics (MaD) Lab  
Friedrich-Alexander-Universität Erlangen-Nürnberg  
28.11.2024

- 
- 1. Time series fundamentals and definitions (Part 1)
  - 2. Time series fundamentals and definitions (Part 2)
  - 3. Bayesian Inference and Gaussian Processes
  - 4. State space models (Kalman Filters)
  - 5. State space models (Particle Filters)
  - 6. Autoregressive models
  - 7. Data mining on time series
  - 8. Deep Learning (DL) for Time Series (Introduction to DL)
  - 9. DL – Convolutional models (CNNs)
  - 10. DL – Recurrent models (RNNs and LSTMs)
  - 11. DL – Attention-based models (Transformers)
  - 12. DL – From BERT to ChatGPT
  - 13. DL – New Trends in Time Series processing
  - 14. Time series in the real world

- 
- 1. Time series fundamentals and definitions (Part 1)
  - 2. Time series fundamentals and definitions (Part 2)
  - 3. Bayesian Inference and Gaussian Processes
  - 4. State space models (Kalman Filters)
  - 5. State space models (Particle Filters)
  - 6. Autoregressive models
  - 7. Data mining on time series**
  - 8. Deep Learning (DL) for Time Series (Introduction to DL)
  - 9. DL – Convolutional models (CNNs)
  - 10. DL – Recurrent models (RNNs and LSTMs)
  - 11. DL – Attention-based models (Transformers)
  - 12. DL – From BERT to ChatGPT
  - 13. DL – New Trends in Time Series processing
  - 14. Time series in the real world

## In This Lecture...

---

**1. Introduction to Data Mining**

**2. Frequency analysis**

**3. Dynamic time warping**

**4. Feature extraction techniques**

# Data Mining with Time Series

## Data Mining



## What is Data Mining?

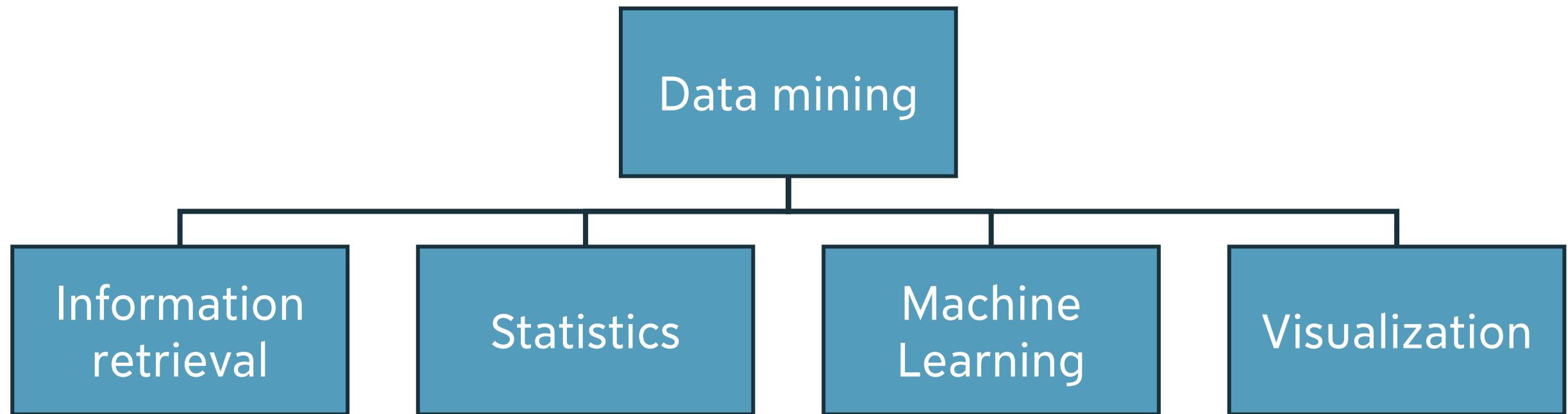
---

**Definition 1.** Extraction of non-simple, implicit, previously unknown, possibly useful data from database.

**Definition 2.** Automatic or semi-automatic search and analysis of a large amount of data with the goal of discovering significant patterns.

Data mining is useful when the information is hidden due to **large** amount of data, its **complexity, heterogeneity**, the speed at which it is collected and need for non-tradition queries.

# What is Data Mining?



## Data Mining

---

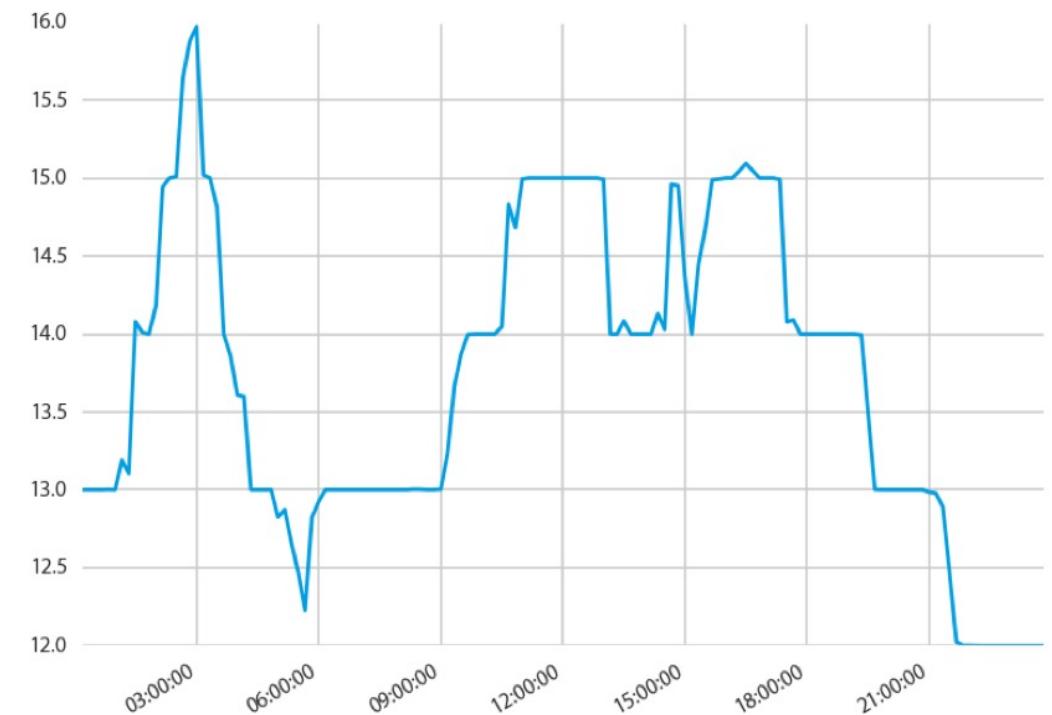
The most basic approach for data mining can include:

- **Computing data characteristics**
- **Data reduction**
- **Data transformation**

## Computing Data Characteristics

Suppose we have a time series of ambient temperature recorded during one day.

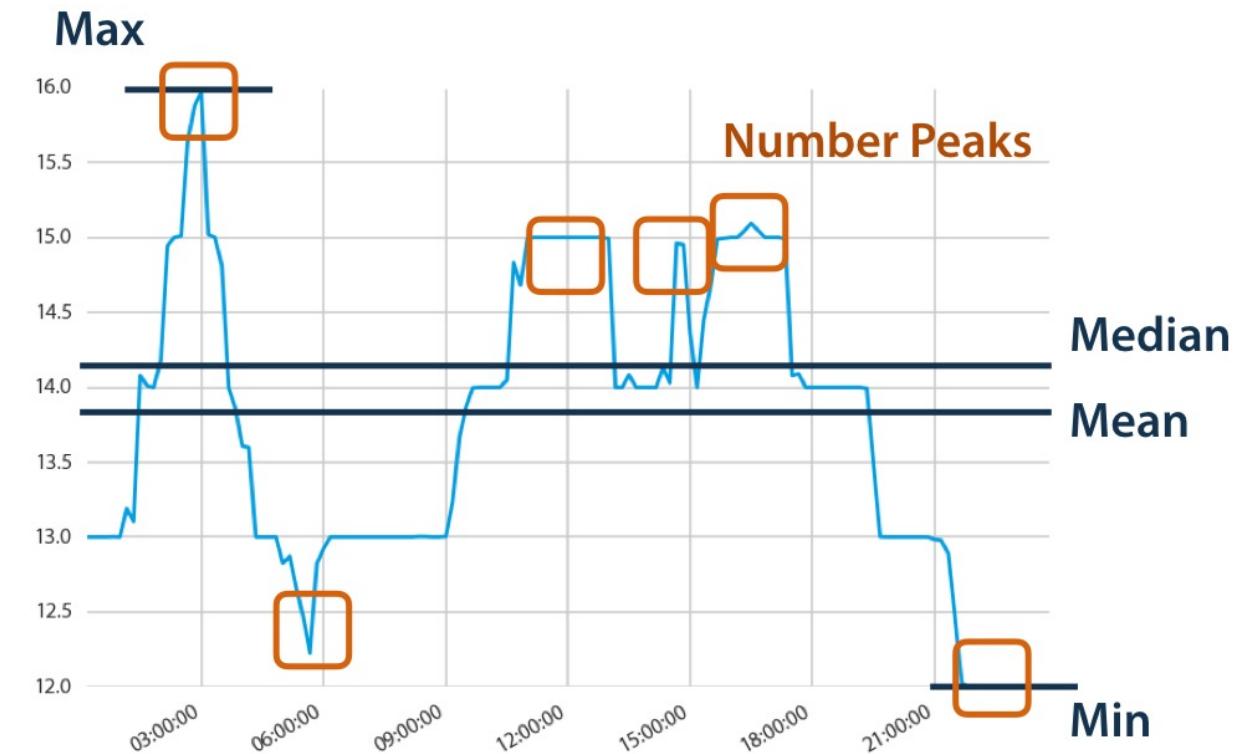
A characterization of this time series can be given by a set of **basic statistics**.



## Computing Data Characteristics

Suppose we have a time series of ambient temperature recorded during one day.

A characterization of this time series can be given by a set of **basic statistics**.



## Data Reduction

---

When dealing with a big dataset, the application of **data reduction techniques** is required in order to allow reasoning on **smaller dimensional** spaces.

- Efficiency
- Interpretability
- Simplicity

There are three main ways to reduce data size:

- **Sampling:** Reducing the number of observations.
- **Selection and projection:** Reducing the number of features.
- **Discretization and aggregation:** Reducing the number of possible values.

## Data Transformation

In most applications, it is necessary to **apply transformations to our data** to make it usable for further analysis.

- Most machine learning algorithms perform better if data has a **consistent** scale distribution.
- E.g., data is **heterogenous** because of different data sources (units of measure, sampling rates, data types).

Different data transformations:

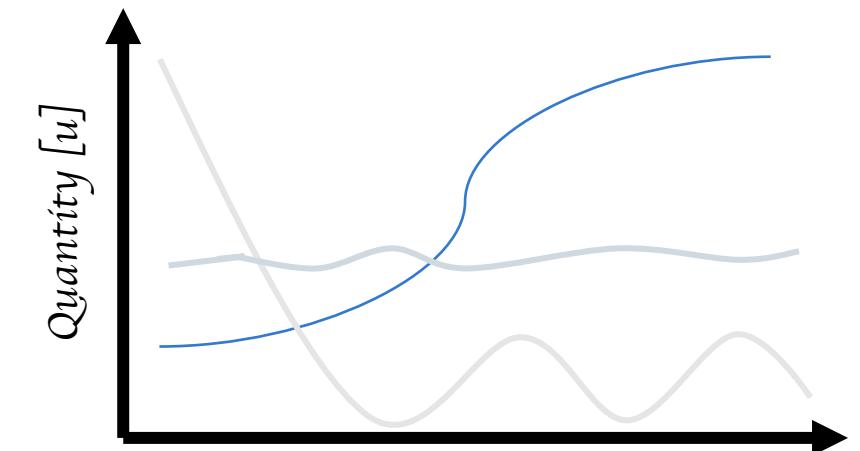
- Normalization
- Standardization

## Data Transformation: Normalization

Feature normalization is used to normalize the range of values of features.

Methods of data normalization:

- Mean normalization
- Min-Max normalization



## Data Transformation: Mean Normalization

Let  $S = (s_1, \dots, s_T)$  be a multivariate time series,  $s_t \in \mathbb{R}^d$  is a  $d$ -dimensional observation at time  $t$ .

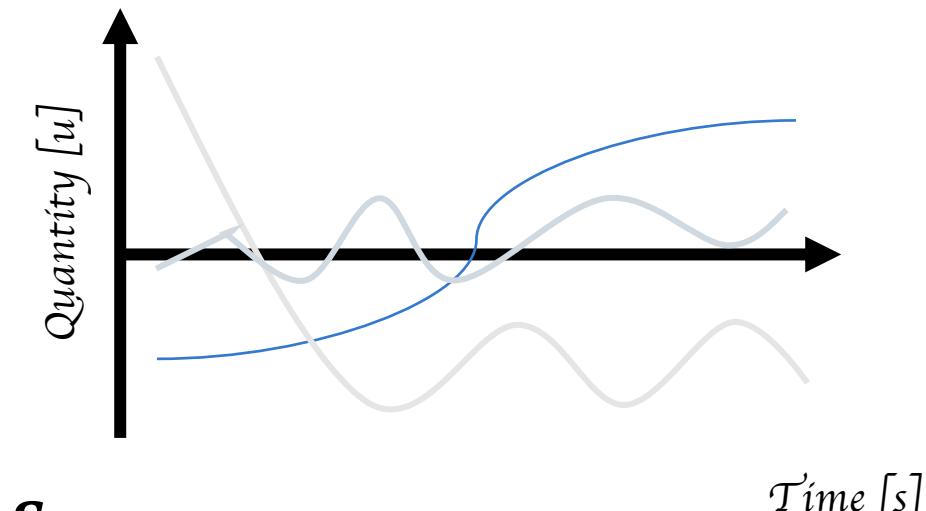
We denote with  $s_j = (s_{1j}, \dots, s_{Tj})$  the  $j$ -th feature of the time series  $S$ , where  $s_{tj} \in \mathbb{R}$  is an observation of the  $j$ -th feature at time  $t$ .

**Mean normalization** is defined by:

$$s'_{tj} = \frac{s_{tj} - \mu_j}{s_{\max,j} - s_{\min,j}}$$

where  $s_{\max,j}$  and  $s_{\min,j}$  are the max and min values of  $s_j$ ,

and  $\mu_j = \frac{1}{T} \sum_{t=1}^T s_{tj}$ .

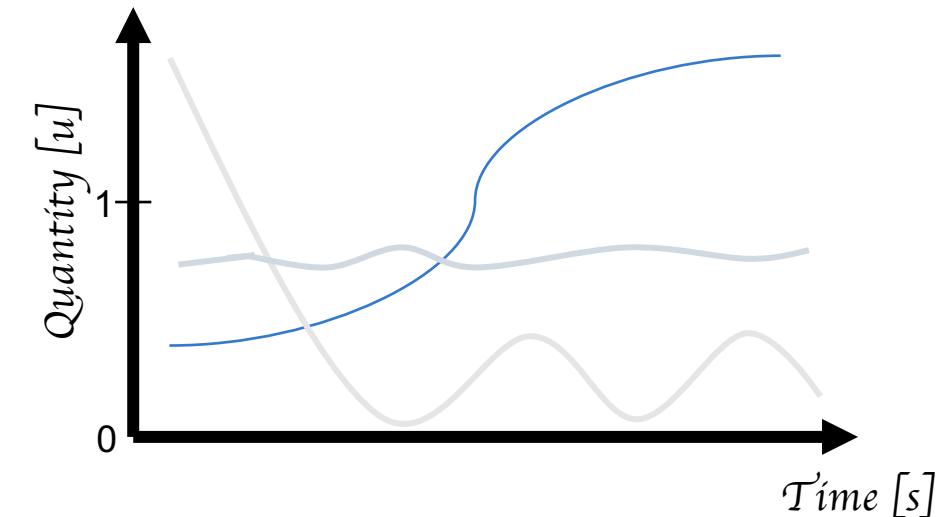


## Data transformation: Min-Max Normalization

**Min-Max normalization** is defined by:

$$s'_{tj} = \frac{s_{tj} - s_{\min,j}}{s_{\max,j} - s_{\min,j}}$$

where  $s_{\max,j}$  and  $s_{\min,j}$  are the max and min values of  $S_j$ .



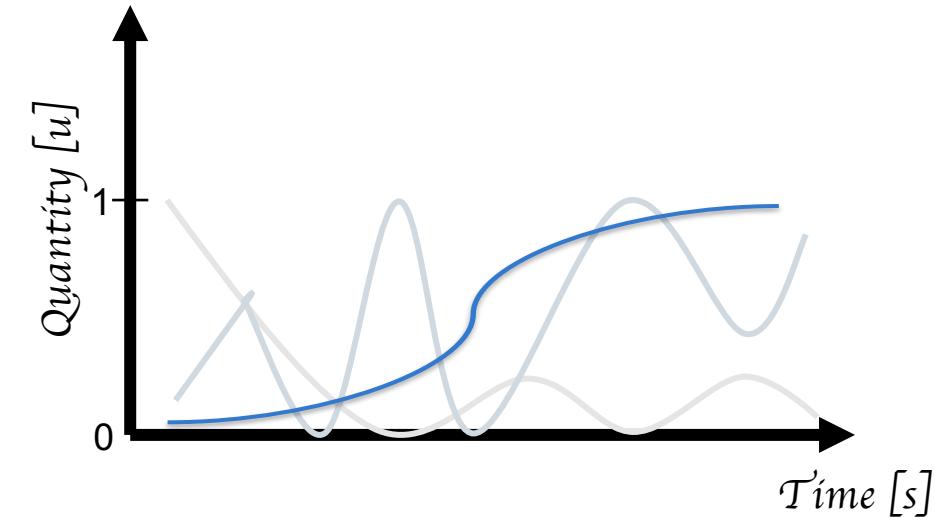
## Data transformation: Min-Max Normalization

**Min-Max normalization** is defined by:

$$s'_{ij} = \frac{s_{ij} - s_{\min,j}}{s_{\max,j} - s_{\min,j}}$$

where  $s_{\max,j}$  and  $s_{\min,j}$  are the max and min values of  $S_j$ .

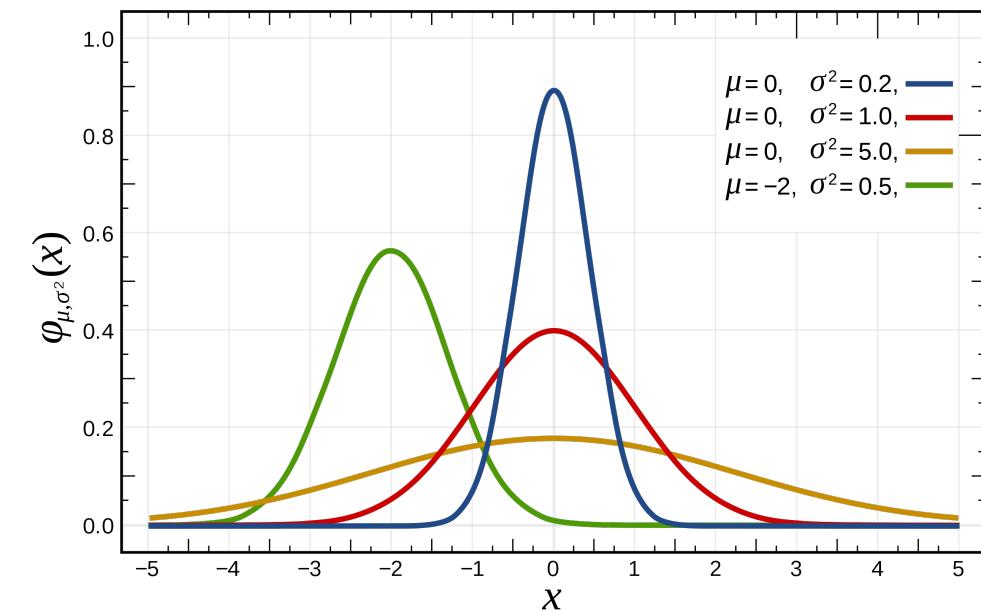
This normalization establishes two boundaries for the data between 0 and 1.



## Data Transformation: Standardization

**Standardization** is the process of converting a random variable with **mean  $\mu$**  and **standard deviation  $\sigma$**  to a “**standard distribution**” (i.e., zero mean and unit standard deviation).

The **standard score** is the number of standard deviations by which the value of a raw score (i.e., an observed value or data point) is above or below the mean value of what is being observed or measured.



[https://en.wikipedia.org/wiki/Normal\\_distribution](https://en.wikipedia.org/wiki/Normal_distribution)

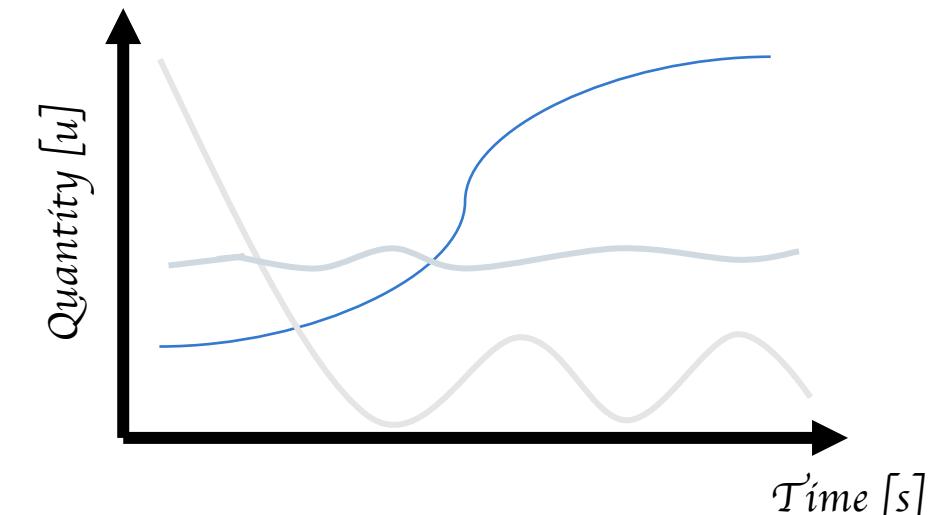
## Data Transformation: Z-score Standardization

**Z-score standardization** is defined by:

$$s'_{tj} = \frac{s_{tj} - \mu_j}{\sigma_j}$$

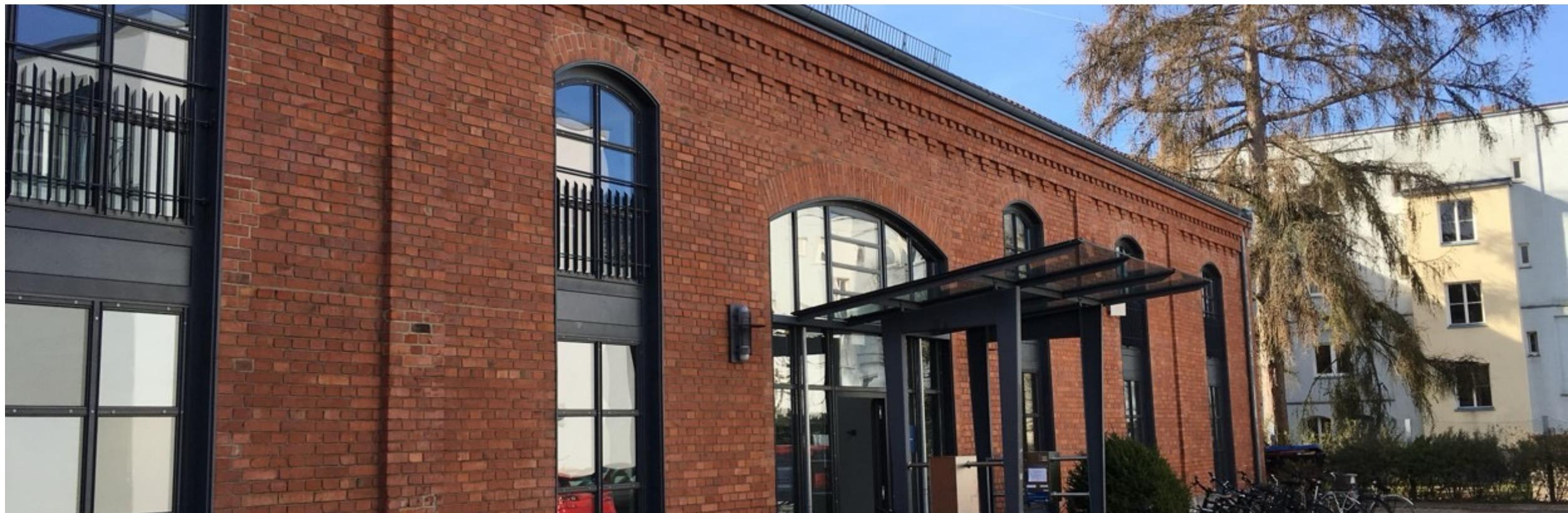
where  $s_{\max,j}$  and  $s_{\min,j}$  are the max and min values of  $S_j$ , and  $\mu_j$  and  $\sigma_j$  are the feature mean and standard deviation.

- The transformed data will have zero mean and unit standard deviation.
- It is empirically shown that, if the original distribution is Gaussian, z-score based transformation generates values that are in the range  $(-3, 3)$ .



# Data Mining with Time Series

## Frequency Analysis



## Spectral Analysis

Spectral analysis is a technique that allows us to discover underlying periodicities.

- Many time series show periodic behavior.
- This periodic behavior can be very complex.
- Additional tool to analyse time series (complementary to the time-domain analysis)

To perform spectral analysis, we first must **transform data from time domain to frequency domain**.

→ We use Fourier transform to convert the time series from the time-domain to the frequency domain

# Spectral Analysis: Sleep Stage Classification

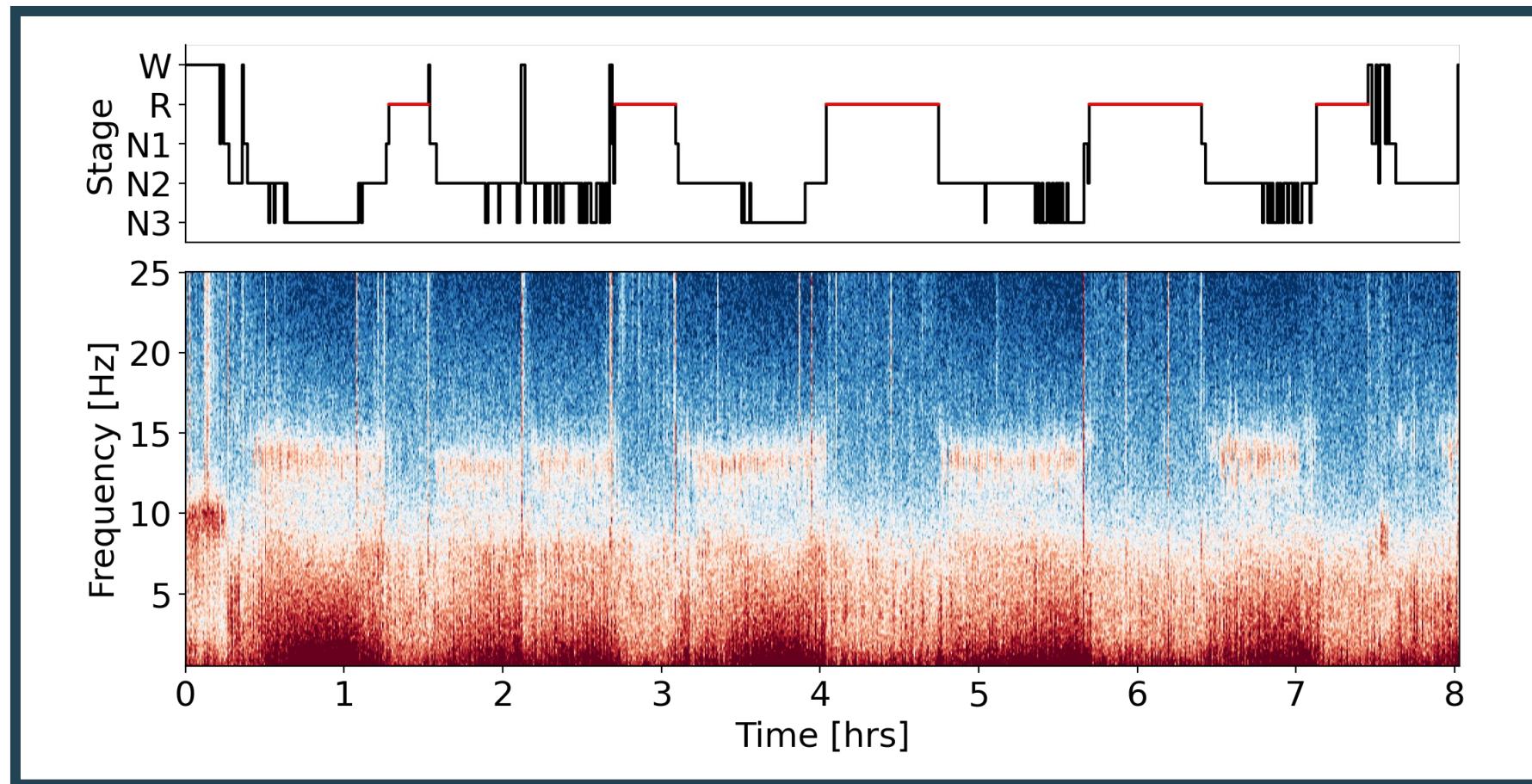


Image from <https://raphaelvallat.com/yasa/build/html/quickstart.html>

## Fourier Representation

Given a time series  $S = (x_1, \dots, x_N)$ , the goal of spectral analysis is that of determining how to construct it using sines and cosines, i.e.,

$$S = \sum_k a_k \sin\left(2\pi \frac{k}{N} t\right) + b_k \cos\left(2\pi \frac{k}{N} t\right)$$

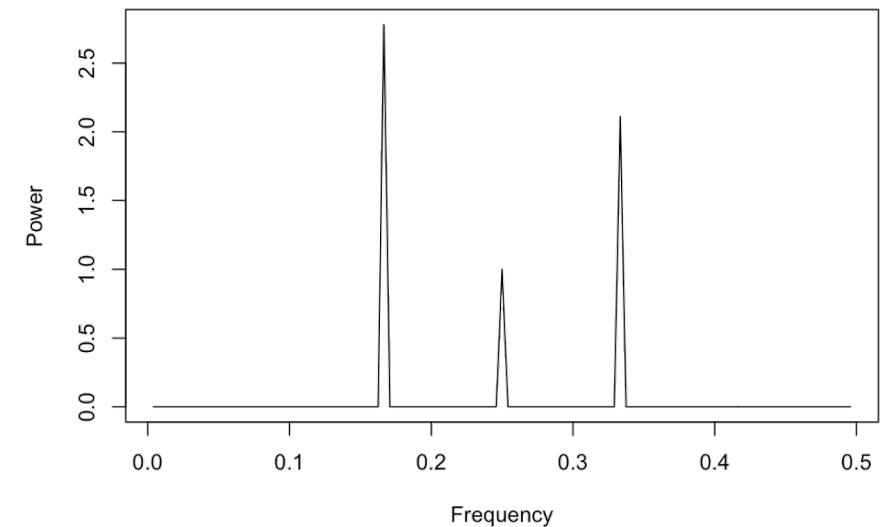
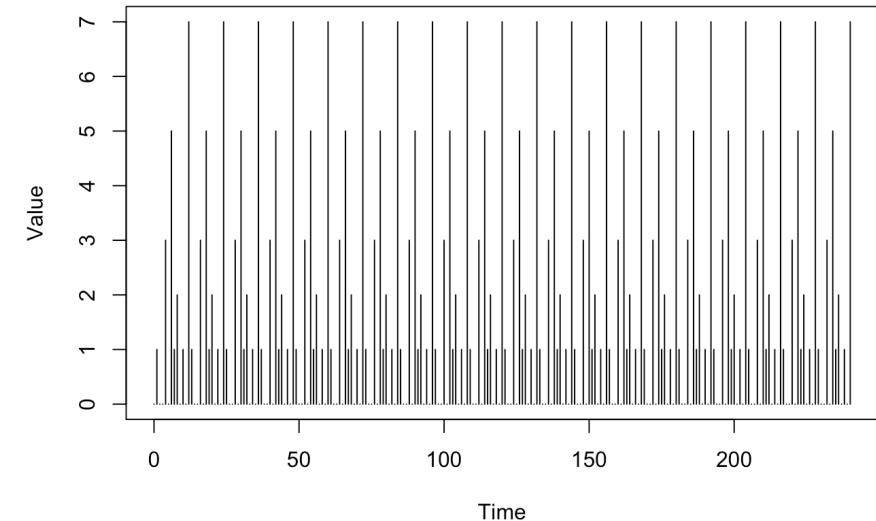
The above expression is called **Fourier representation** for a time series.

- It allows us to re-express time series in a standard way
- Different time series are characterized by different coefficients
  - $a_k$ 's and  $b_k$ 's can be determined in a closed form
  - We can compare time series by comparing their coefficients

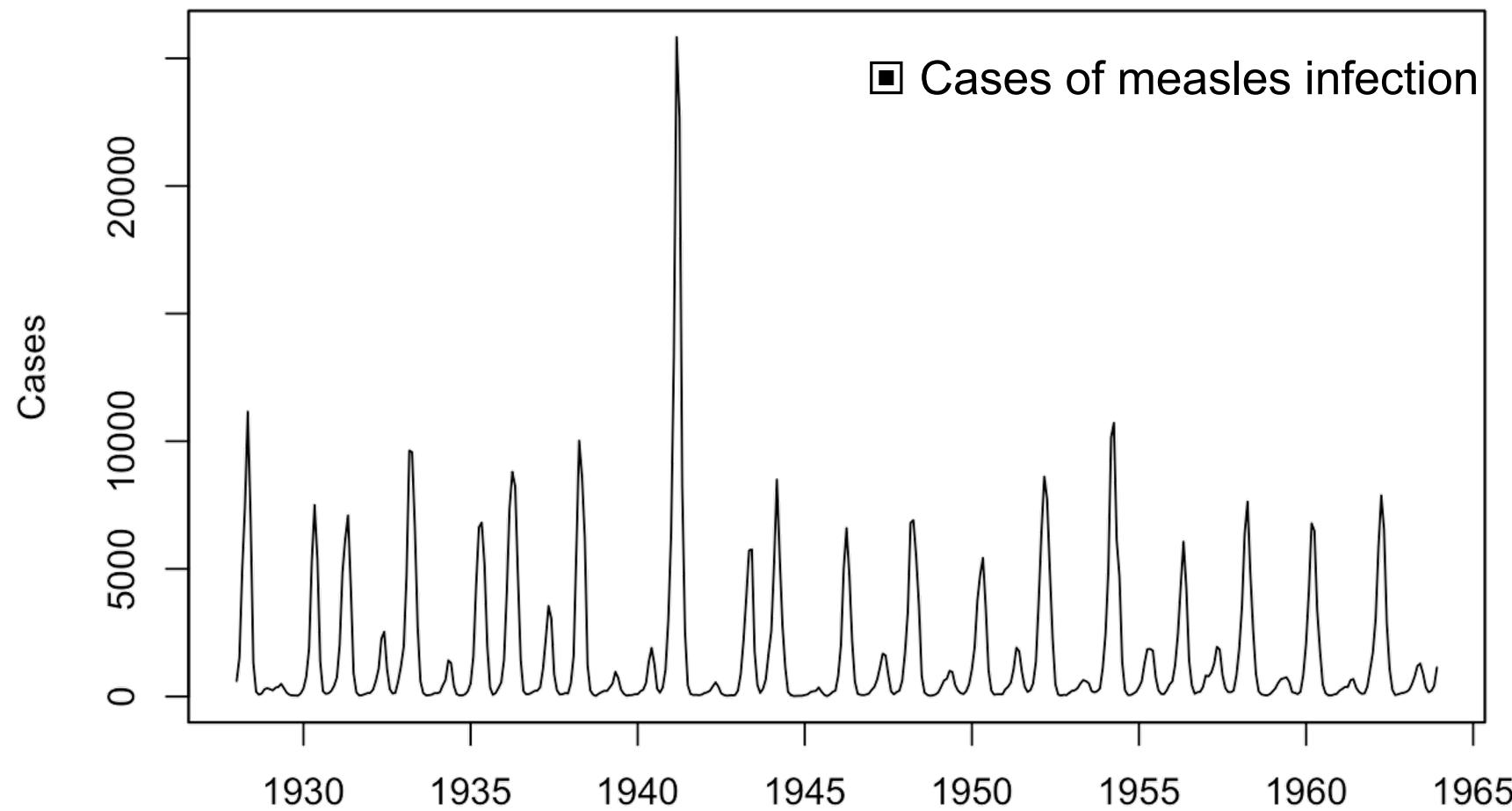
## Spectral Density

In brief, the covariance of the time series can be represented by a function known as the **spectral density**.

The spectral density can be estimated using an object known as a **periodogram**, which is the squared correlation between our time series and sine/cosine waves at the different frequencies spanned by the series.



## Example: Spectral Analysis

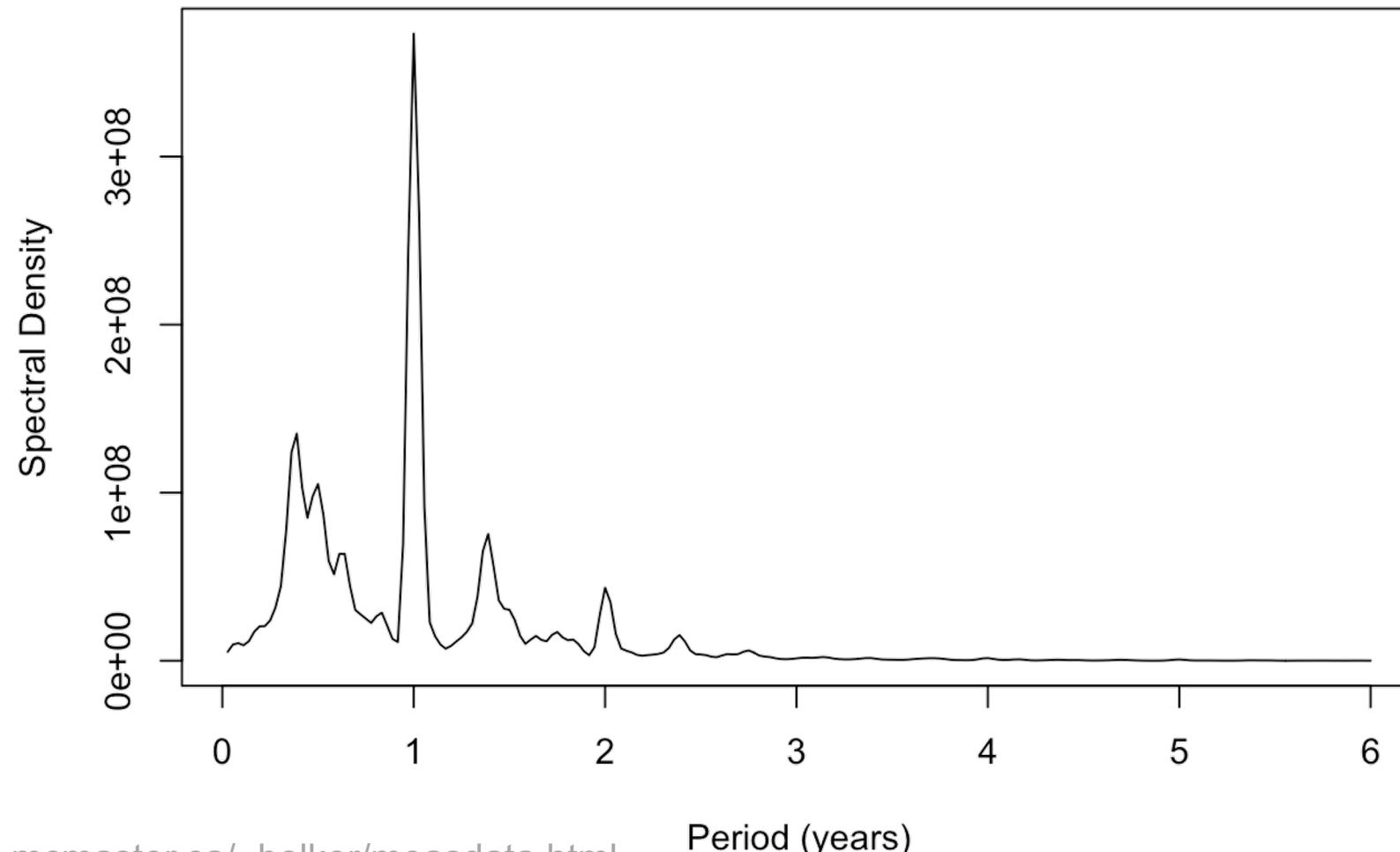


Data from: <https://ms.mcmaster.ca/~bolker/measdata.html>

Date

Images from: <http://web.stanford.edu/class/earthsys214/notes/series.html#spectral-analysis>

## Example: Spectral Analysis



Data from: <https://ms.mcmaster.ca/~bolker/measdata.html>

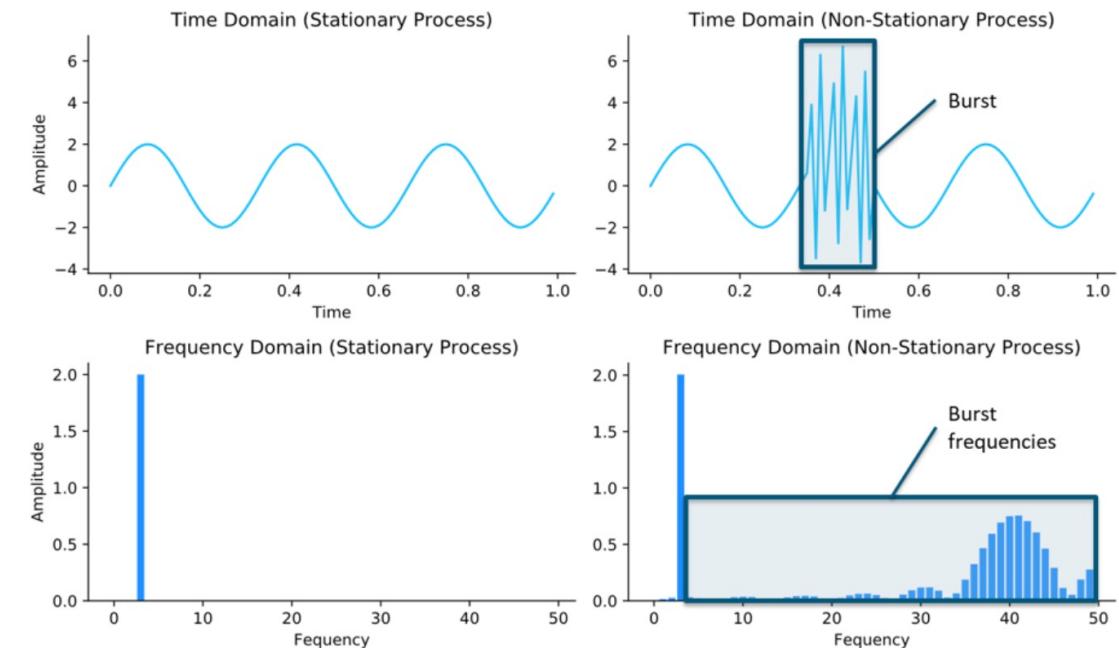
Images from: <http://web.stanford.edu/class/earthsys214/notes/series.html#spectral-analysis>

## Limitations of the Fourier Transform

Fourier transform works well for sin/cosin waves generated by **stationary** signals.

For a non-stationary signal (e.g., containing an anomaly), the Fourier transform outputs frequencies that construct the signal but we cannot identify which of them represents the burst/anomaly.

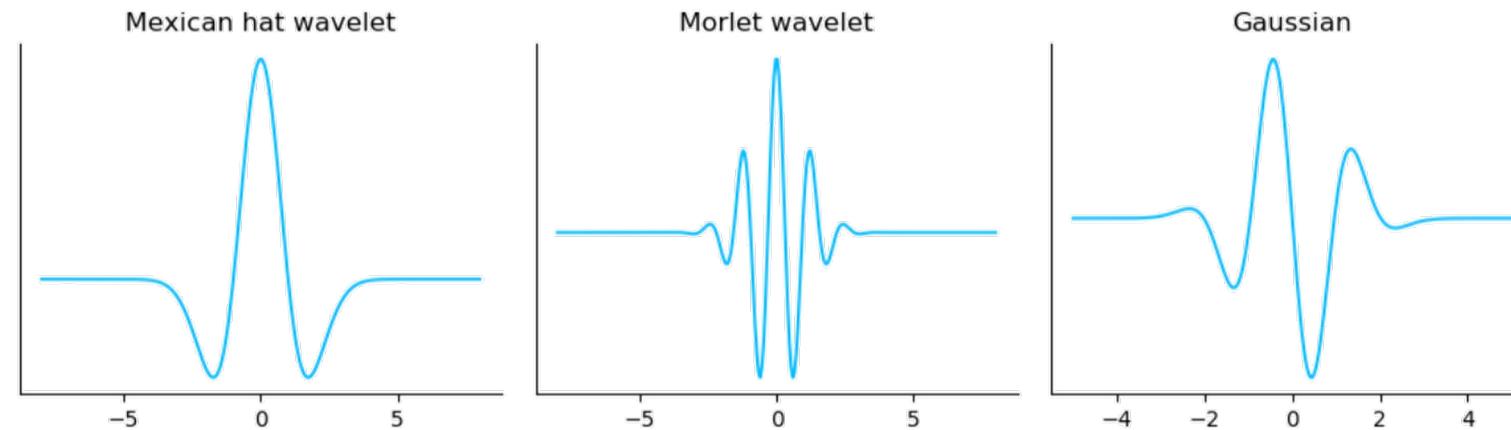
→ With non-stationary signals, we are confined to either time or frequency domain.



<https://towardsdatascience.com/multiple-time-series-classification-by-using-continuous-wavelet-transformation-d29df97c0442>

## Continuous Wavelet Transform

Continuous Wavelet Transform (CWT) is based on the concept of **wavelets** (or mini wavelets).



- In contrast to sines/cosines used in Fourier transform, wavelets are limited (in time)
- Have zero mean.

Both these conditions allow a localization in **time and frequency** at the same time.

## Continuous Wavelet Transform

Continuous Wavelet Transform (CWT) is defined as follows:

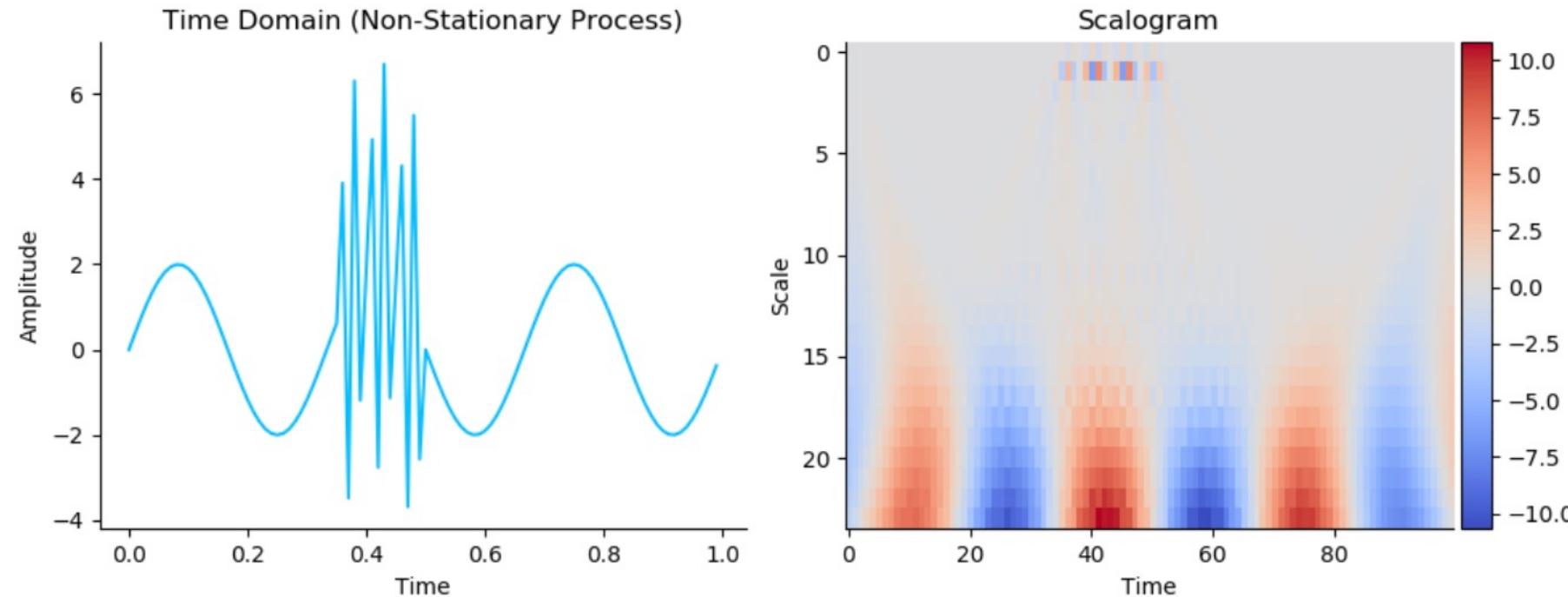
$$cwt(\tau, s) = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{+\infty} x(t)\psi\left(\frac{t - \tau}{s}\right) dt$$

where  $\tau$  is the translation,  $s$  is the scale,  $\psi$  is the mother wavelet.

- The CWT is based on the concepts of scaling and shifting.
- It transforms the original 1-D time series to a N-D "image".

## Example: Continuous Wavelet Transform

If we apply CWT to the previous non-stationary signal, we obtain:



## Continuous Wavelet Transform

---

Continuous Wavelet Transform (CWT) are:

- Efficient in determining the damping ratio of oscillating signals (e.g. identification of damping in dynamic systems)
- Robust to the noise in the signal
- 2D scalogram can be used to improve the distinction between varying types of a signal. E.g.,
  - differentiate between different production processes in a machine
  - identifying components or tools faults
- **Pros:** The wavelet transform considers both frequency and time components
- **Cons:** Reduced resolution

# Data Mining with Time Series

## Dynamic Time Warping

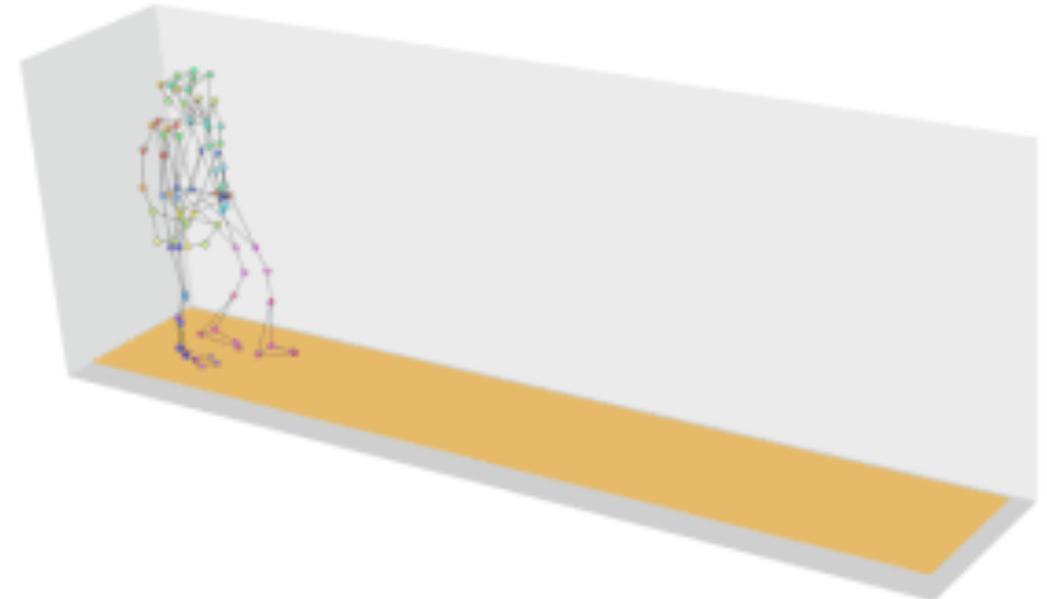


## Dynamic Time Warping

In many applications, there is the need to analyse multiple time series at the same time to find similarities between time series.

- E.g., speech recognition, signature recognition, similarity in walking, ...

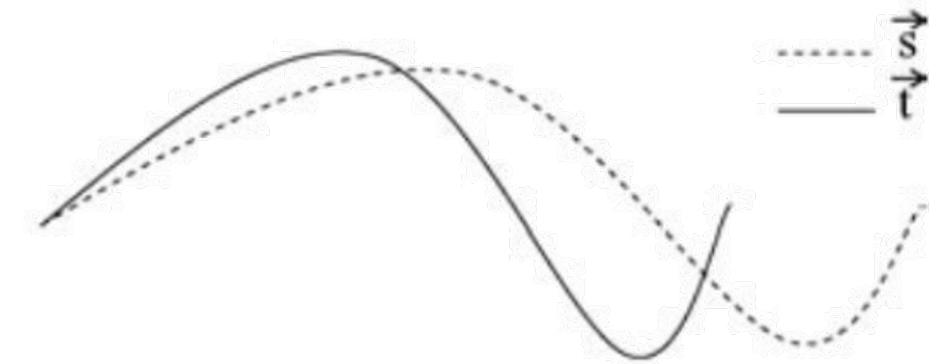
Euclidean distance does not work for time series that are not perfectly synchronized.



## Dynamic Time Warping

**Dynamic time warping (DTW)** is an algorithm to measure similarity between two time-series that may **vary in speed and time**.

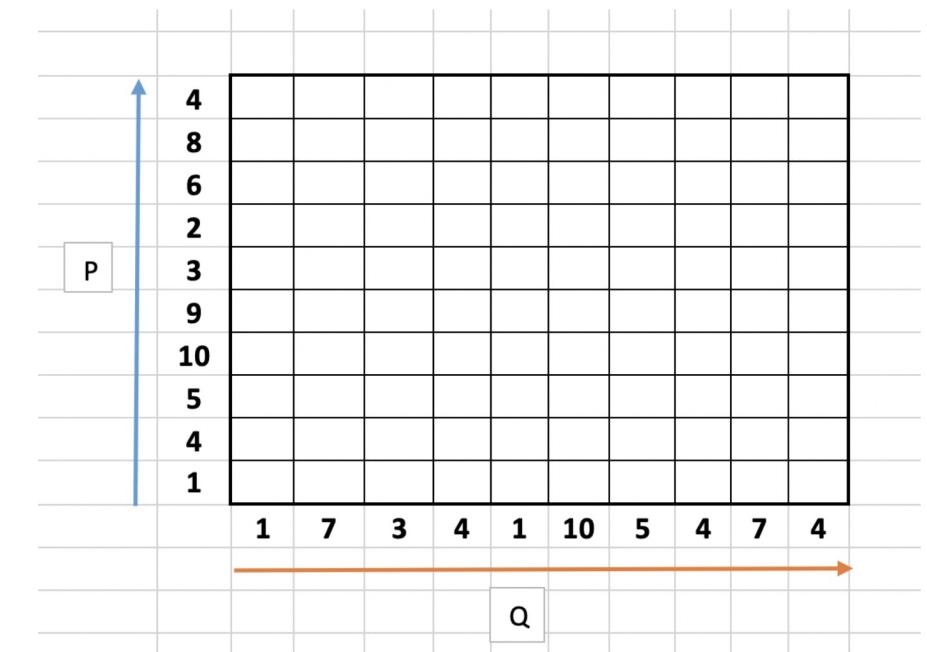
DTW determines the optimal global alignment between two time series.



## Dynamic Time Warping: Algorithm

Given two time series  $x_t$  and  $y_t$ , we can compute the DTW distance as follows:

1. Initialize the distance matrix  $M$



## Dynamic Time Warping: Algorithm

Given two time series  $x_t$  and  $y_t$ , we can compute the DTW distance as follows:

1. Initialize the distance matrix  $M$
2. Fill  $M$  from the bottom left corner, according to the formula:

$$M(i, j) = \text{dist}(x_i, y_j) + \min(M(i - 1, j - 1), M(i, j - 1), M(i - 1, j))$$

Please notice:

- $M(i, j)$  denotes the cell of the  $i$ -th row and  $j$ -th column in the  $M$  matrix, starting from the bottom left.
- Near to the axis,  $M(i - 1, j - 1)$ ,  $M(i, j - 1)$  and  $M(i - 1, j)$  fall “outside” the matrix  $M$  and are considered as “0” in the equation.

4	42	24	-	-	-	-	-	-	-	-
8	39	21	-	-	-	-	-	-	-	-
6	32	20	-	-	-	-	-	-	-	-
2	27	19	-	-	-	-	-	-	-	-
3	26	14	-	-	-	-	-	-	-	-
9	24	10	-	-	-	-	-	-	-	-
10	16	8	12	11	-	-	-	-	-	-
5	7	5	5	5	-	-	-	-	-	-
4	3	3	4	4	7	13	14	14	17	17
1	0	6	8	11	11	20	24	27	33	36

1    7    3    4    1    10    5    4    7    4

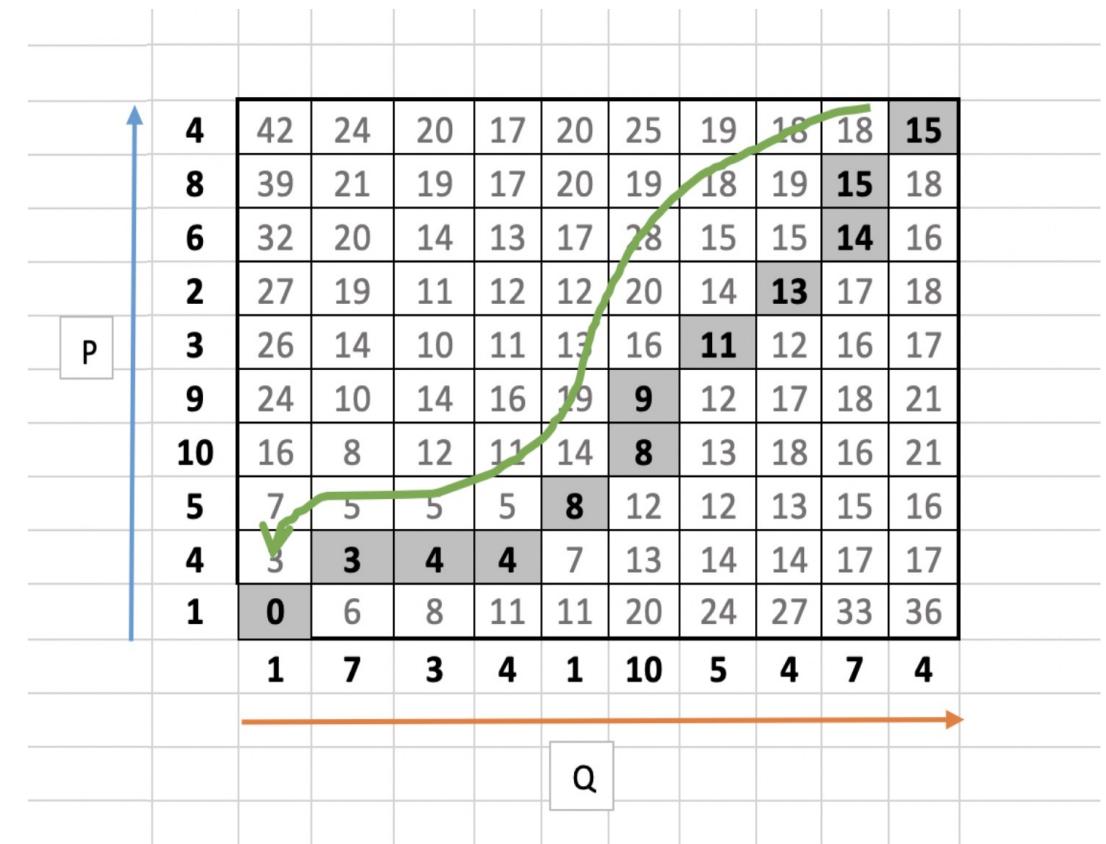
Q

# Dynamic Time Warping: Algorithm

Given two time series  $x_t$  and  $y_t$ , we can compute the DTW distance as follows:

1. Initialize the distance matrix  $M$
  2. Fill  $M$  from the bottom left corner, according to the formula:
$$M(i, j) = \text{dist}(x_i, y_j) + \min(M(i - 1, j - 1), M(i, j - 1), M(i - 1, j))$$
  3. Identify the **warping path**  $d$ , starting from the top right corner

$$M(i, j) = \text{dist}(x_i, y_j) + \min(M(i - 1, j - 1), M(i, j - 1), M(i - 1, j))$$



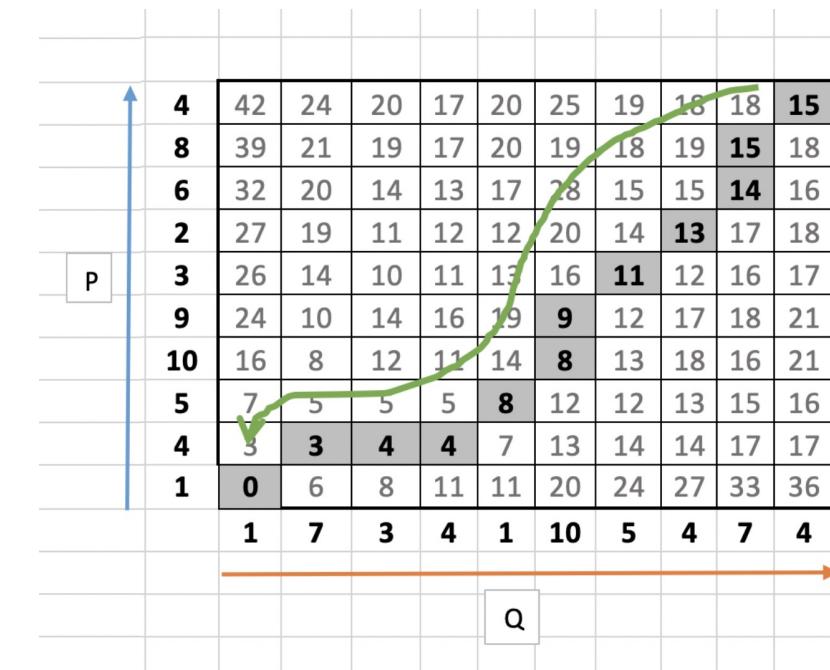
## Dynamic Time Warping: Algorithm

Given two time series  $x_t$  and  $y_t$ , we can compute the DTW distance as follows:

1. Initialize the distance matrix  $M$
2. Fill  $M$  from the bottom left corner, according to the formula:  

$$M(i, j) = dist(x_i, y_j) + \min(M(i - 1, j - 1), M(i, j - 1), M(i - 1, j))$$
3. Identify the **warping path**  $d$ , starting from the top right corner. ( $d$  is a list of indices pairs)
4. The overall path cost can be calculated as

$$D = \sum_{(i,j) \in d} dist(x_i, y_j)$$



## Dynamic Time Warping: Pros and Cons

---

Pros:

- Exploit a **non-linear** distortion (in time) to find non-trivial similarity

Cons:

- **High computational cost.** Alternatives for computing the alignment path more efficiently have been presented.
- It needs the preparation of reliable reference templates for the set of words to be recognized.

# Data Mining with Time Series

Other feature extraction methods for time series



## Similarity Join

---

One method to find anomalies and trends in time series is to perform a **similarity join**.

- Compare pair of snippets in a time series based on a similarity measure
- Retrieve all data pairs whose distances are smaller than a predefined threshold  $\epsilon$ .
- **Very easy to implement**
- **Computationally expensive for moderately large collection of data**

## Matrix Profiles

---

**Matrix profile (MP)** is a data structure for time series analysis.

Advantages:

- MP is domain agnostic
- Efficient
- Provides exact solutions
- Only requires a single parameter

# Matrix Profiles

The Matrix Profile has two primary components:

- **distance profile**: a vector of minimum Z-Normalized Euclidean distances
- **profile index**: it contains the index of its first nearest-neighbor

The algorithms that compute the Matrix Profile use a **sliding window** approach.

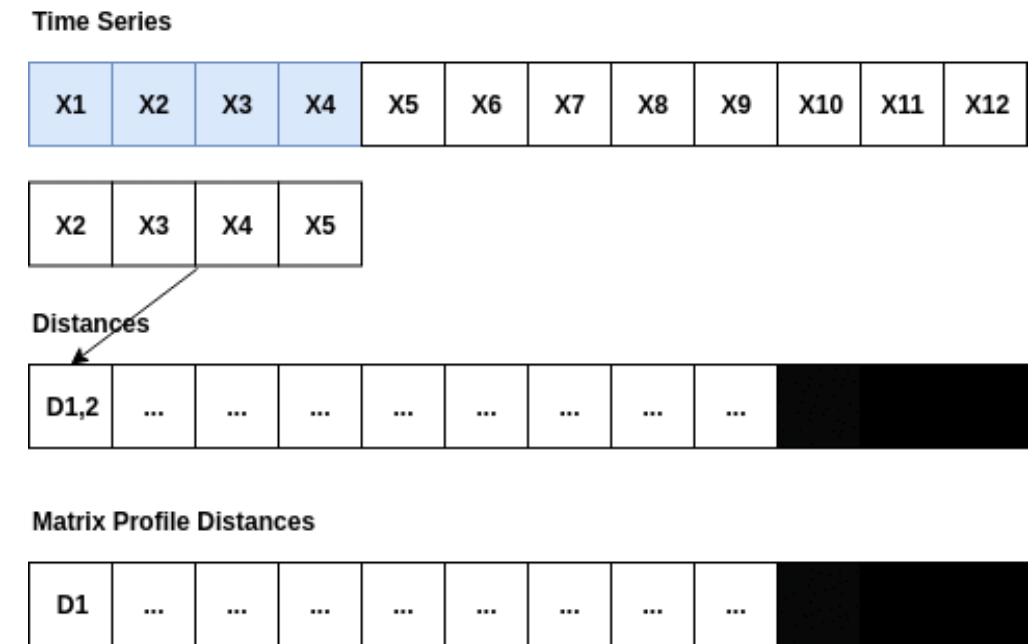
Let  $m$  be the window size, and  $n$  be the time series length.



# Matrix Profiles: the Algorithm

## The general algorithm:

1. Compute the distances for the windowed sub-sequence against the entire time series
2. Set an exclusion zone to ignore trivial matches
3. Updates the distance profile with the minimal values
4. Set the first nearest-neighbor index

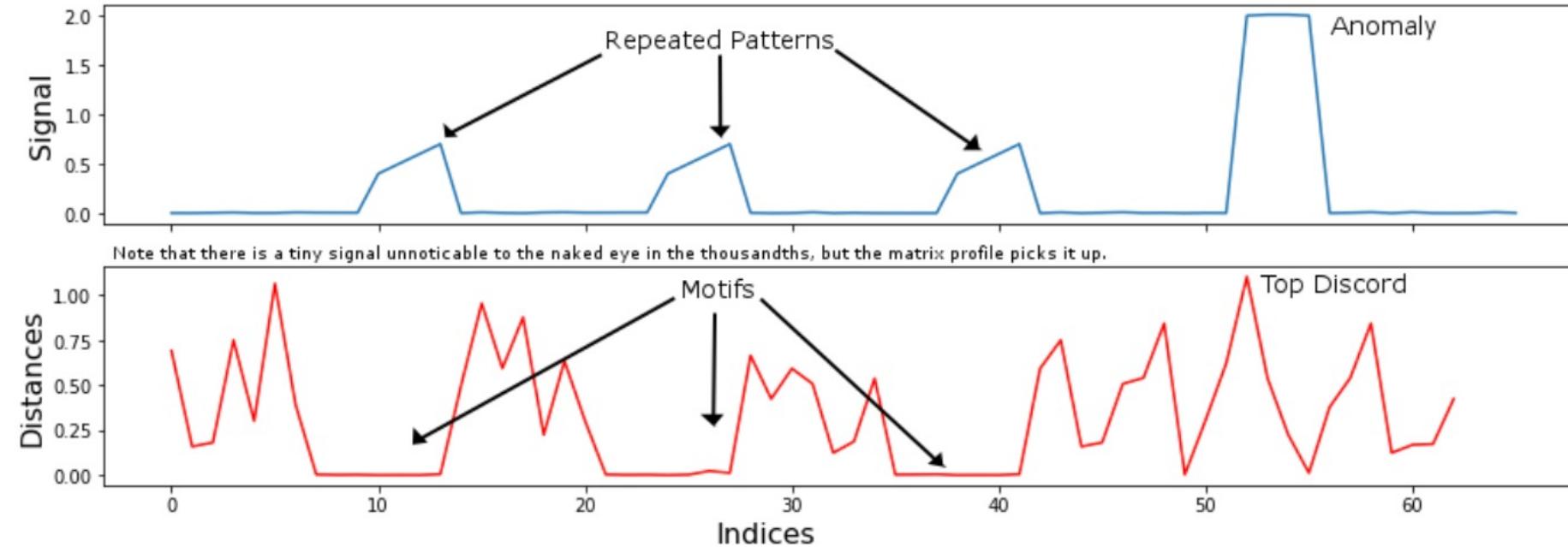


<https://towardsdatascience.com/introduction-to-matrix-profiles-5568f3375d90>

Original paper: All Pairs Similarity Joins for Time Series: A Unifying View that Includes Motifs, Discords and Shapelets. Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, Eamonn Keogh (2016). IEEE ICDM 2016.

## Matrix Profiles: Motifs and Discords

When the matrix profile is computed, it is possible to identify **motifs (repeated patterns)** and **discords (anomalies)** in a time series.



<https://towardsdatascience.com/introduction-to-matrix-profiles-5568f3375d90>

Original paper: All Pairs Similarity Joins for Time Series: A Unifying View that Includes Motifs, Discords and Shapelets. Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, Eamonn Keogh (2016). IEEE ICDM 2016.

## Signature Method

The **signature method** refers to a collection of feature extraction techniques for multivariate time series, derived from the theory of controlled differential equations.

→ Successfully applied in a wide range of ML tasks with **sequential data**, e.g.,  
chinese character recognition or feature extraction from financial data streams.

Given a multivariate time series  $X: [a, b] \rightarrow \mathbb{R}^d$  (or more generally called **path** in this context), we define the increment of the  $i$ -th coordinate of the path as:

$$S(X)_{a,t}^i = \int_{a < s < t} dX_s^i = X_t^i - X_0^i$$

## Signature Method

For every  $(i, j) \in \{1, \dots, d\}^2$  we define the **double iterated integral**:

$$S(X)_{a,t}^{i,j} = \int_{a < s < t} S(X)_{a,s}^i dX_s^j = \int_{a < r < s < t} dX_r^i dX_s^j$$

Similarly, we can define the **triple-iterated integral**:

$$S(X)_{a,t}^{i,j,k} = \int_{a < s < t} S(X)_{a,s}^{i,j} dX_s^k$$

And, recursively, we can construct the **k-fold iterated integral**:

$$S(X)_{a,t}^{i_1, \dots, i_k} = \int_{a < s < t} S(X)_{a,s}^{i_1, \dots, i_{k-1}} dX_s^{i_k}$$

## Signature Method

We define signature of a time series (or, path) the **infinite series of all the iterated integrals**, defined by:

$$S(X)_{a,b} = (1, S(X)_{a,b}^1, \dots, S(X)_{a,b}^d, S(X)_{a,b}^{1,1}, S(X)_{a,b}^{1,2}, \dots)$$

where the superscripts vary within the set of all multi-indexes

$$W = \{(i_1, \dots, i_k) | k \geq 1; i_1, \dots, i_k \in \{1, \dots, d\}\}$$

# Data Mining with Time Series

## Recap



## In This Lecture

---

- Introduction to Data Mining
- Frequency analysys
- Dynamic time warping
- Feature extraction techniques

## Python Resources

---

- **Tsfresh:** python library which is used to extract characteristics from time series.

<https://tsfresh.readthedocs.io/en/latest/>

## Python Resources

---

- **Tsfresh** is a Python library which is used to extract characteristics from time series.

<https://tsfresh.readthedocs.io/en/latest/>

- **TSFEL** is a Python package for feature extraction on time series data.

<https://tsfel.readthedocs.io/en/latest/>

# Python Resources

- **Tsfresh** is a Python library which is used to extract characteristics from time series.

<https://tsfresh.readthedocs.io/en/latest/>

- **TSFEL** is a Python package for feature extraction on time series data.

<https://tsfel.readthedocs.io/en/latest/>

- **A systematic review of Python packages for time series**, Siebert et al.

Paper: <https://arxiv.org/abs/2104.07406>

Website: <https://siebert-julien.github.io/time-series-analysis-python/>

Name	Tasks								Data Preparation			
	forecasting	classification	clustering	anomaly detection	segmentation	pattern recognition	change point detection	dimensionality reduction	missing values imputation	decomposition	preprocessing	simi me
	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
arch	true	false	false	false	false	false	false	false	false	false	false	false
atspy	true	false	false	false	false	false	false	false	true	true	true	false
banpei	false	false	false	true	false	false	true	false	false	false	false	false
cesium	false	false	false	false	false	false	false	false	false	false	true	false
darts	true	false	false	false	false	false	false	false	true	true	true	false
deeptime	true	false	true	false	false	false	false	true	false	true	true	false
delタپy	true	false	true	false	false	false	false	true	false	true	true	true
dtaidistance	false	false	true	false	false	false	false	false	false	false	false	true
EMD-signal	false	false	false	false	false	false	false	false	true	false	false	false
flood-forecast	true	false	false	false	false	false	false	false	true	false	true	false

