# Efficient Algorithms for Time Series Retrieval

## Master Thesis Defense

Philipp Beer - 25.05.2022

# Motivation

## Focus and Target

We present an algorithm suitable as foundation for a time-series search engine that identifies **similar time series** from a pool of time series data.

We focus on:

- Time Complexity must adequate for a search engine
- Algorithm should be flexible to time series properties
- Results don’t need to be closest match (but rather results that are good enough)

# Agenda

1. Key Concepts
2. Methodology
3. Challenges
4. Demo
5. Results Overview
6. Discussion and Future Work

# Key Concepts

# Key Concepts

## Time Series Retrieval

- Analysis of time series data by means of making it searchable
- Utilize time series data properties for comparison

# Key Concepts

## Euclidean Distance

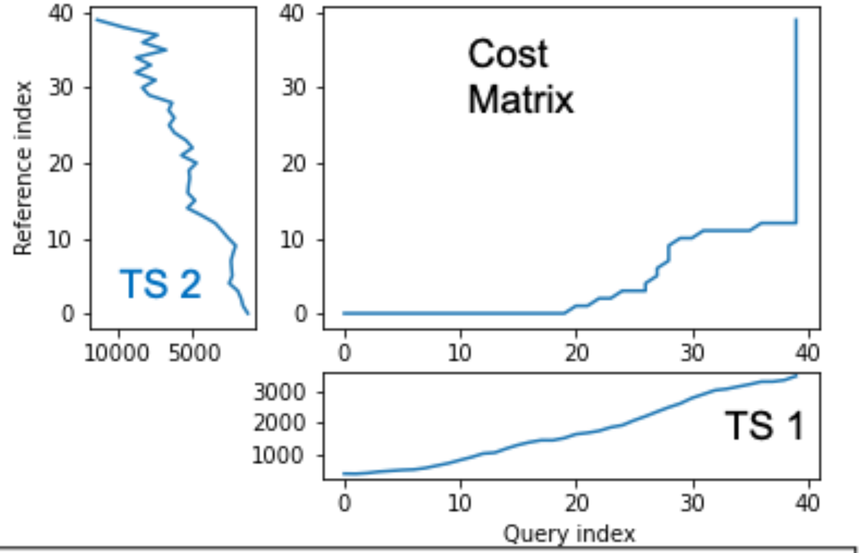
- Most widely-used similarity metric
- Easy to compute
- Intuitive to understand
- Cannot different length time series S and Q
- Struggles with outliers and noise
- Limited application in real-world scenarios

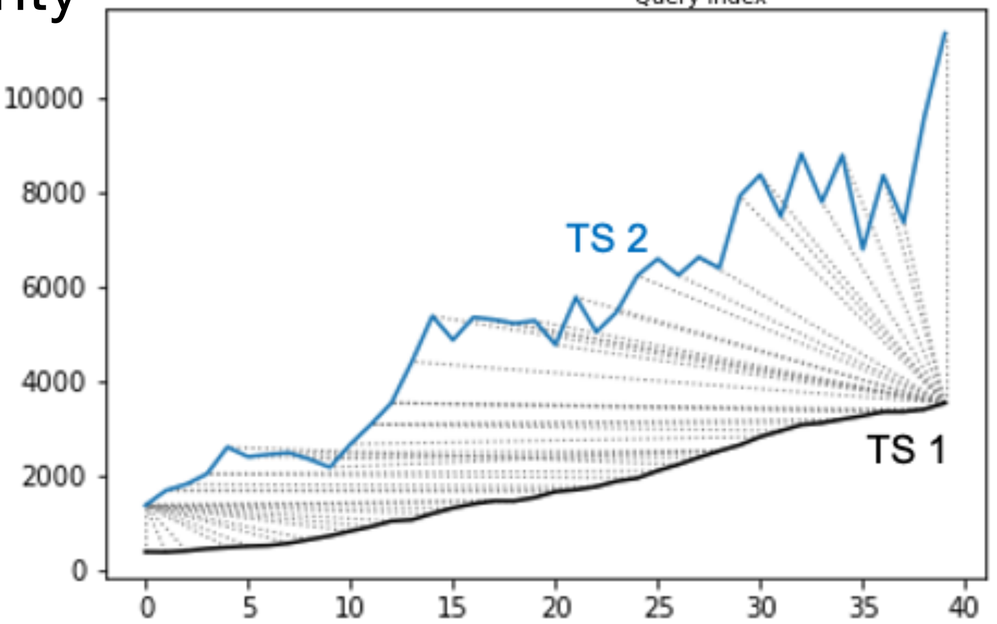
$$D(S, Q) = \sqrt{\Sigma(S_i - Q_i)^2}$$

# Key Concepts

## Dynamic Time Warping

- Minimal path through 2 time series via warping
- Baseline for our comparison
- Most widely used metric for time series similarity
- Exhaustive search of solution space
- Outliers can lead to clustering
- Not scalable
- Time Complexity $\mathcal{O}(n^2)$



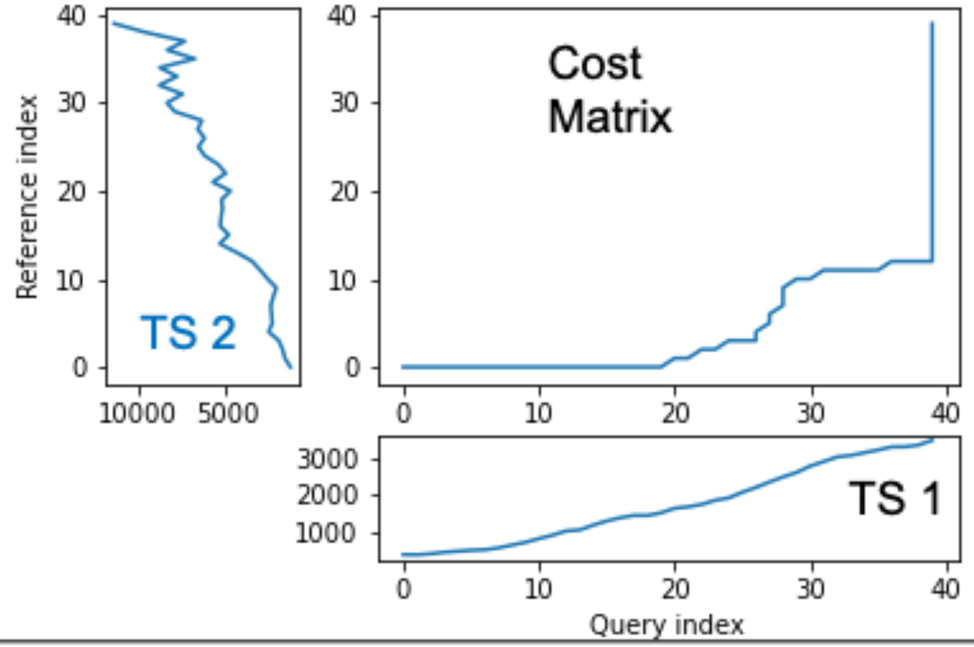



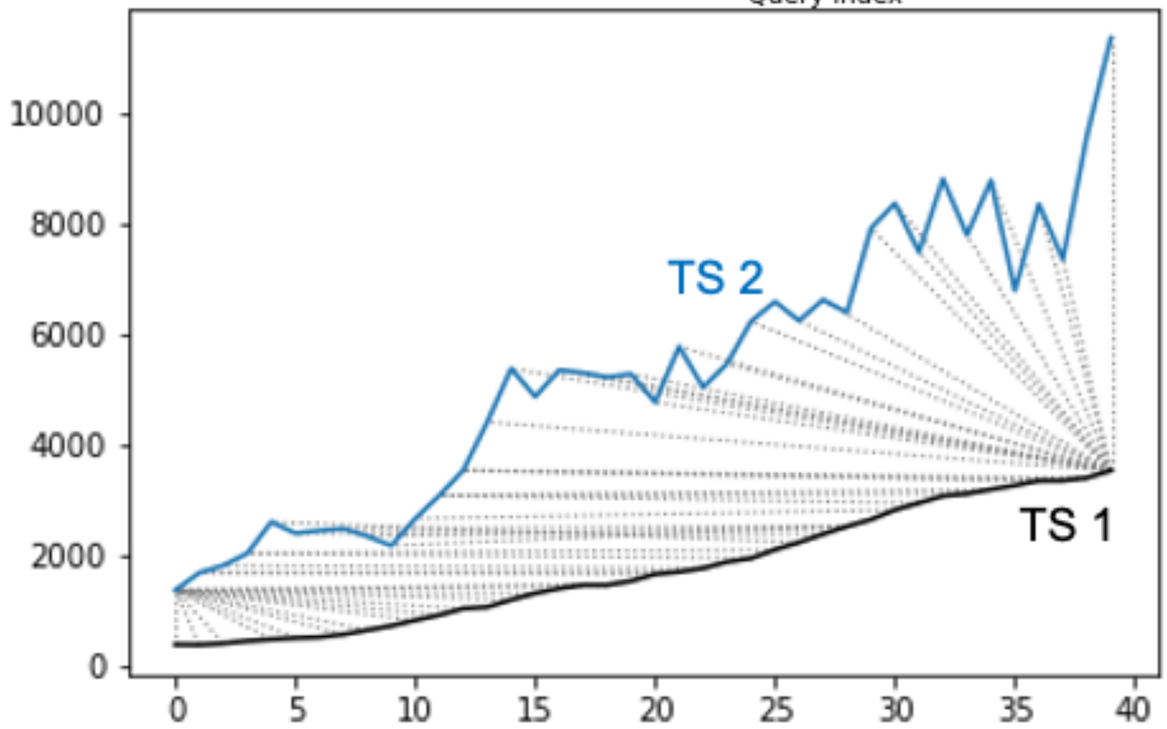
# Key Concepts

## Dynamic Time Warping

$$\gamma(i,j) = D(s_i, q_j) + min\{\gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1)\}$$

| | | |
|---|---|---|
| $S_{i-2,j}$ | $S_{i-1,j}$ | $S_{i,j}$ |
| $S_{i-2,j-1}$ | $S_{i-1,j-1}$ | $S_{i,j-1}$ |
| $S_{i-2,j-2}$ | $S_{i-1,j-2}$ | $S_{i,j-2}$ |



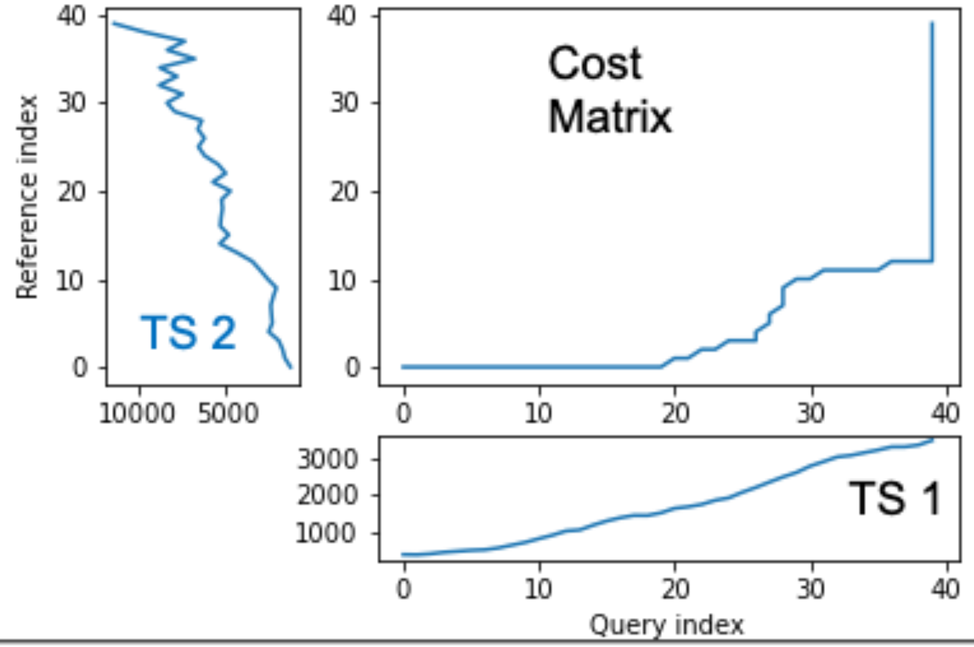



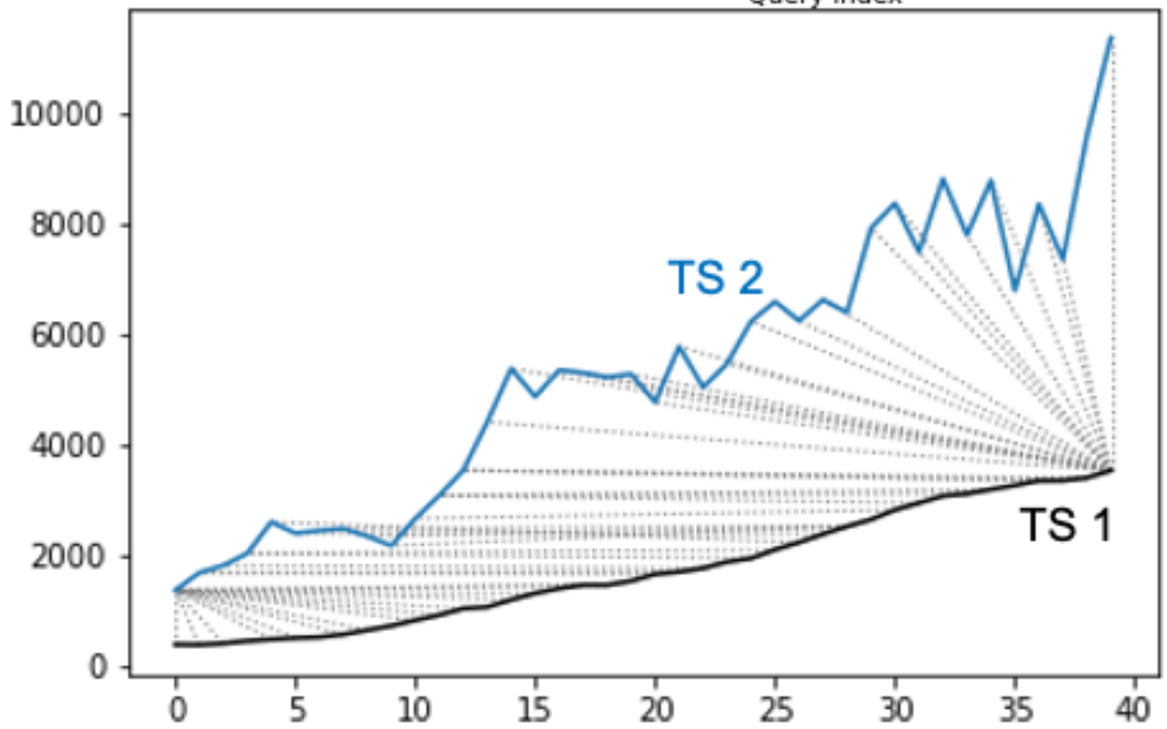
# Key Concepts

## Dynamic Time Warping

$$\gamma(i,j) = D(s_i, q_j) + min\{\gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1)\}$$

| | | |
|---|---|---|
| $S_{i-2,j}$ | $S_{i-1,j}$ | $S_{i,j}$ |
| $S_{i-2,j-1}$ | $S_{i-1,j-1}$ | $S_{i,j-1}$ |
| $S_{i-2,j-2}$ | $S_{i-1,j-2}$ | $S_{i,j-2}$ |



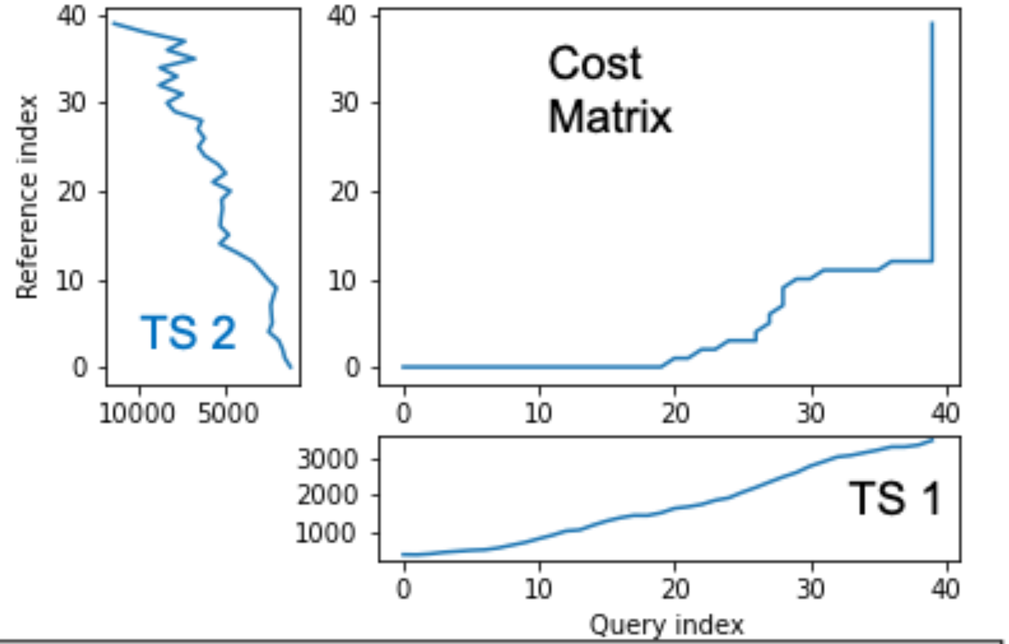



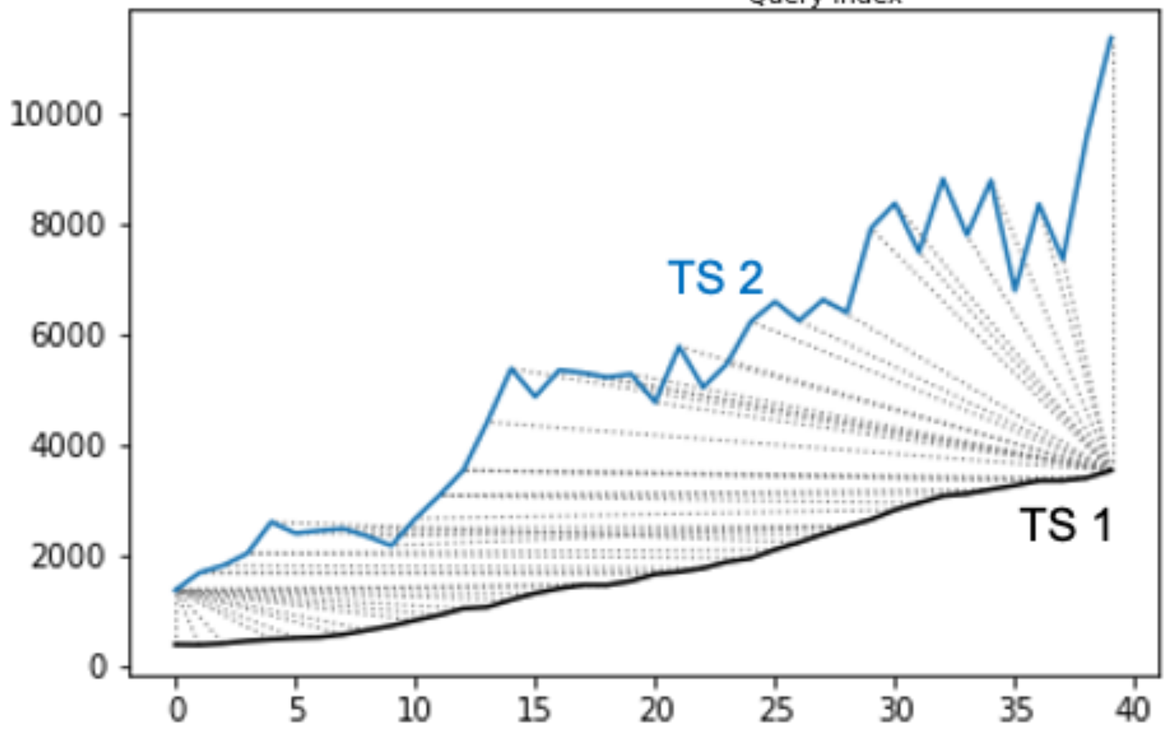
# Key Concepts

## Dynamic Time Warping

$$\gamma(i,j) = D(s_i, q_j) + min\{\gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1)\}$$

| | | |
|---|---|---|
| $S_{i-2,j}$ | $S_{i-1,j}$ | $S_{i,j}$ |
| $S_{i-2,j-1}$ | $S_{i-1,j-1}$ | $S_{i,j-1}$ |
| $S_{i-2,j-2}$ | $S_{i-1,j-2}$ | $S_{i,j-2}$ |



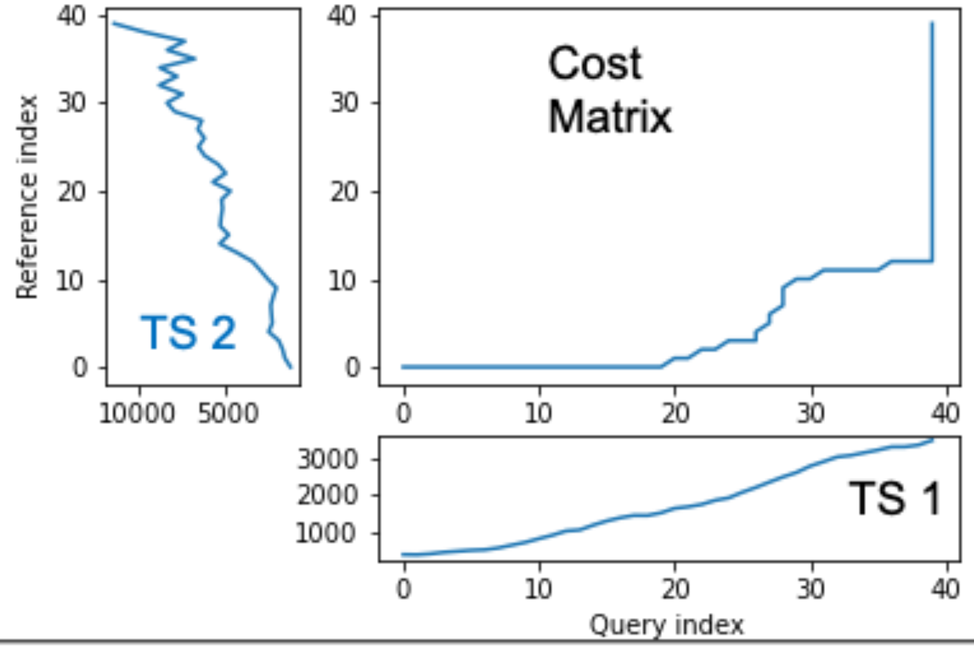



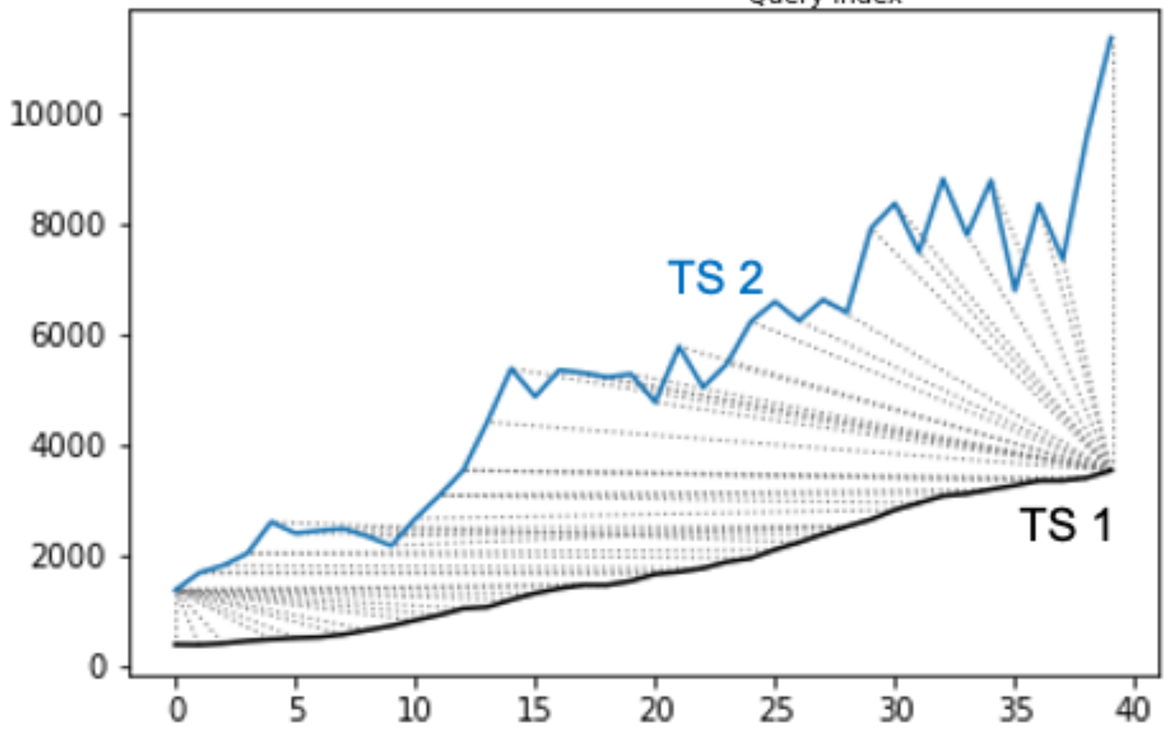
# Key Concepts

## Dynamic Time Warping

$$\gamma(i,j) = D(s_i, q_j) + min\{\gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1)\}$$

| | | |
|---|---|---|
| $S_{i-2,j}$ | $S_{i-1,j}$ | $S_{i,j}$ |
| $S_{i-2,j-1}$ | $S_{i-1,j-1}$ | $S_{i,j-1}$ |
| $S_{i-2,j-2}$ | $S_{i-1,j-2}$ | $S_{i,j-2}$ |



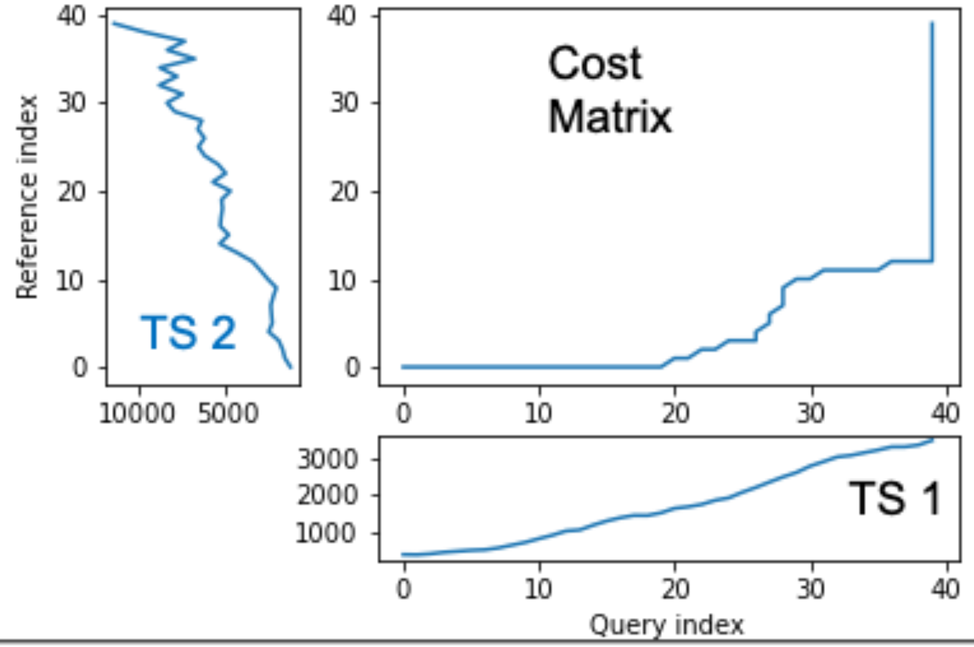



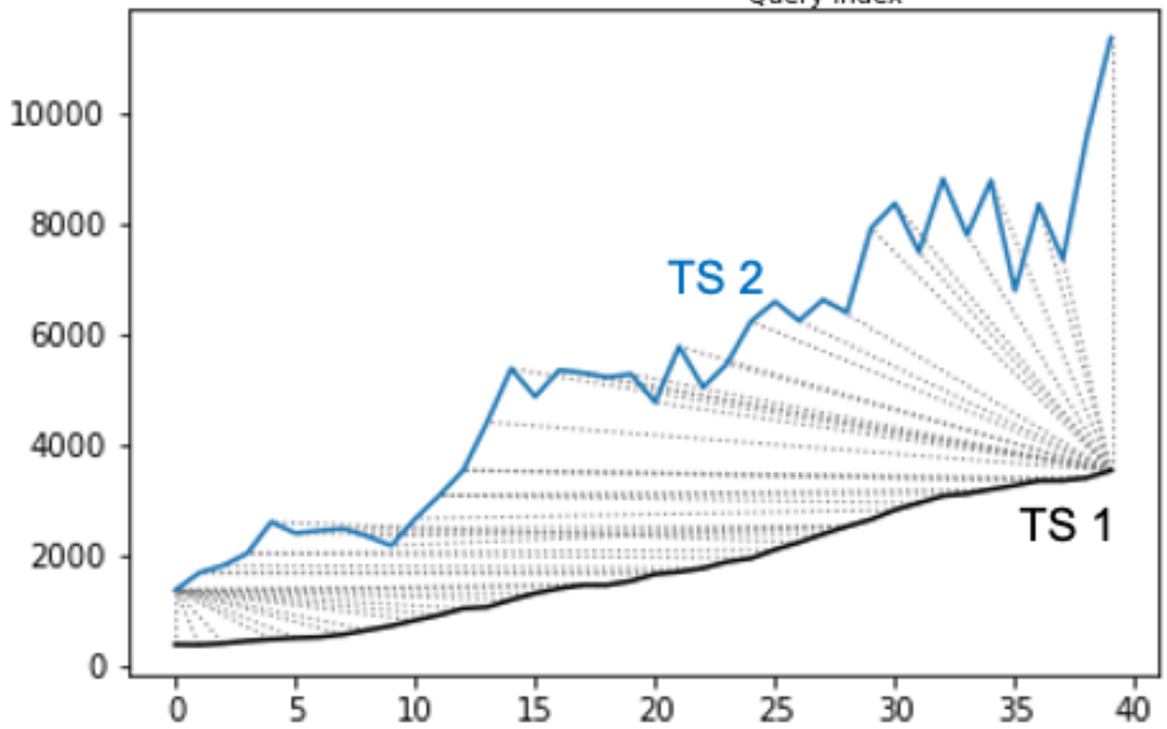
# Key Concepts

## Dynamic Time Warping

$$\gamma(i,j) = D(s_i, q_j) + min\{\gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1)\}$$

| | | |
|---|---|---|
| $S_{i-2,j}$ | $S_{i-1,j}$ | $S_{i,j}$ |
| $S_{i-2,j-1}$ | $S_{i-1,j-1}$ | $S_{i,j-1}$ |
| $S_{i-2,j-2}$ | $S_{i-1,j-2}$ | $S_{i,j-2}$ |



# Key Concepts

## Fourier Transformation

- Principal Idea: Project Functions and data vectors into a coordinate system of sine and cosine functions with increasing frequencies
- Describe data vector as Fourier coefficients
- Parseval's Theorem (preservation of L2-norm)
- Power Spectral Density - magnitude of frequencies described
- Key for our work: **Use largest few PSD-ranked frequencies to describe similarity between two time-series**

# Methodology

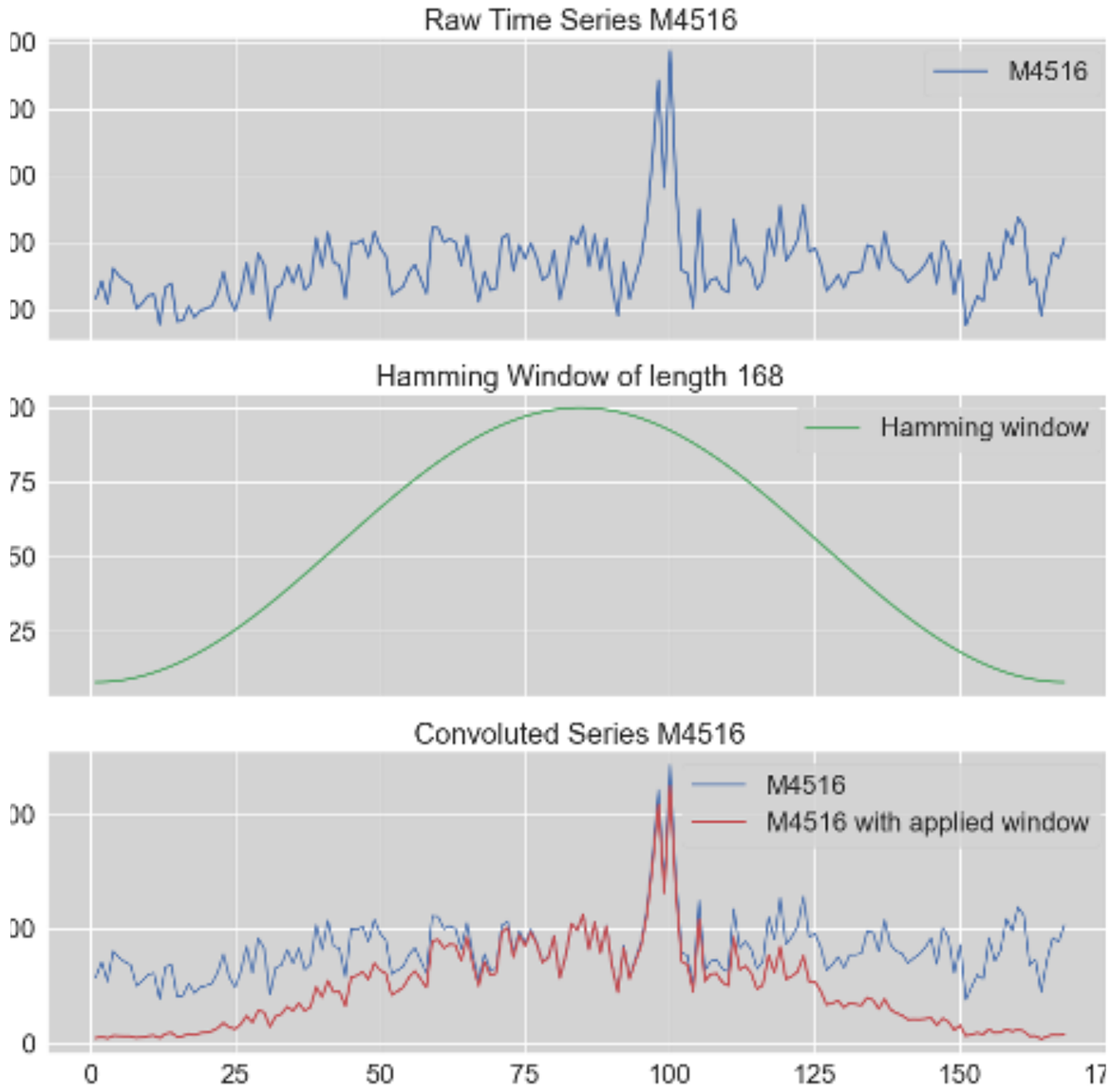
Raw Time Series

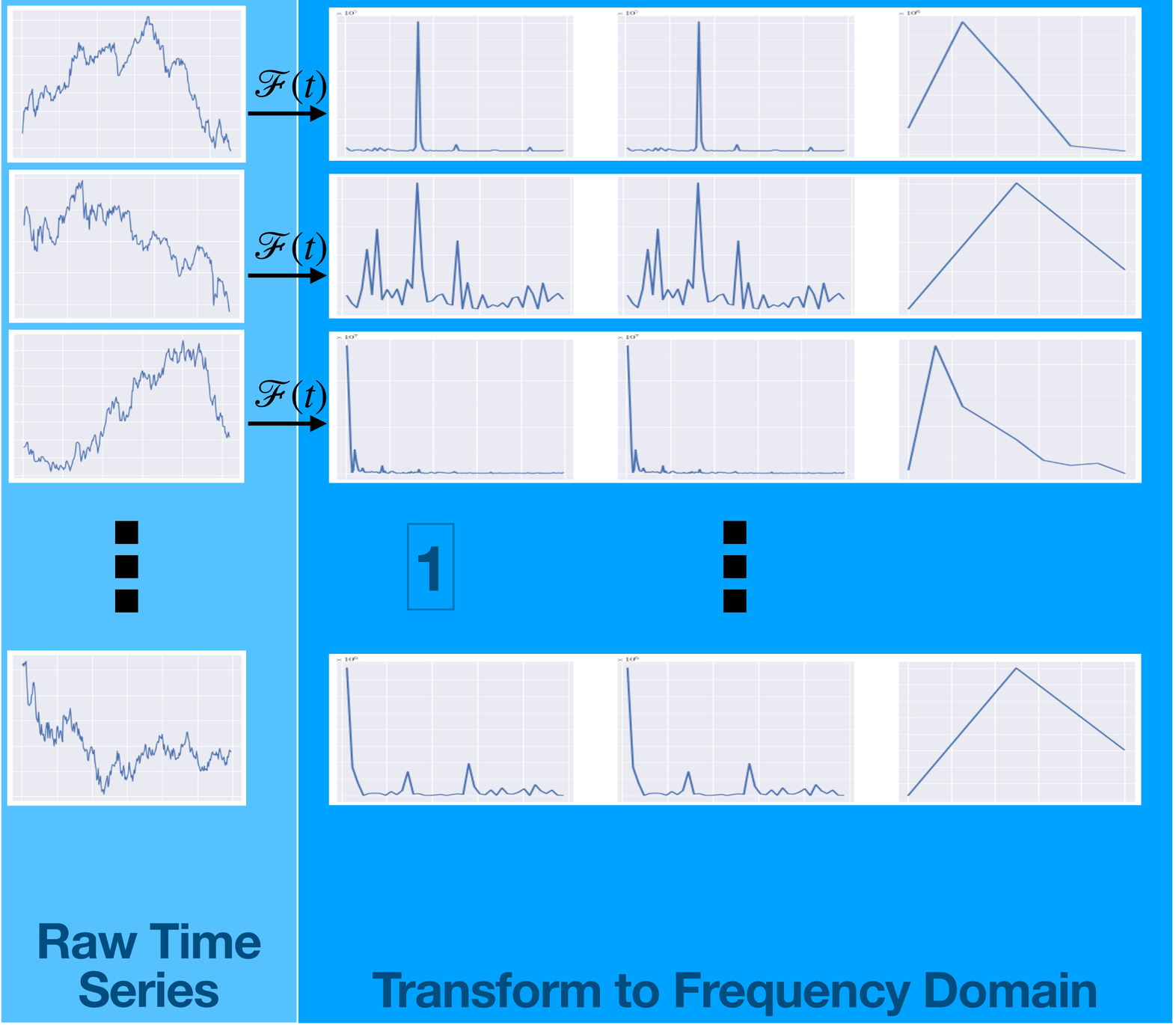
# Transformation & Statistics Computation of Time Series Pool

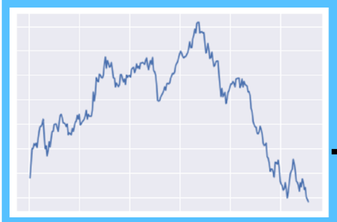
# Window Functions

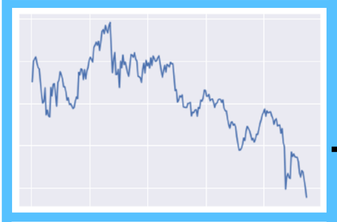
## Adjusted Time Series

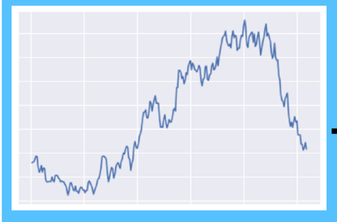
- Time Series + Window = Convoluted Series
- Address spectral leakage in time series




$\mathcal{F}(t)$
$\mathcal{F}(t)$
$\mathcal{F}(t)$
1
Raw Time Series
Transform to Frequency Domain



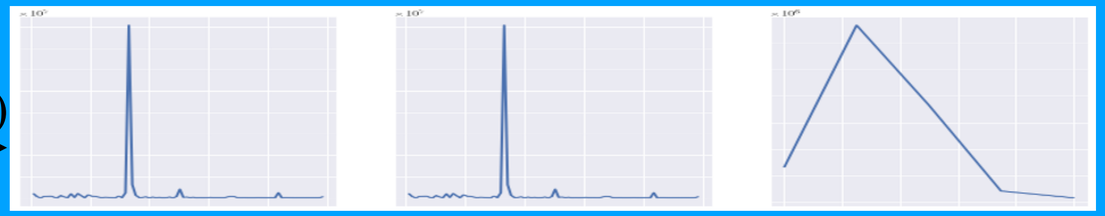


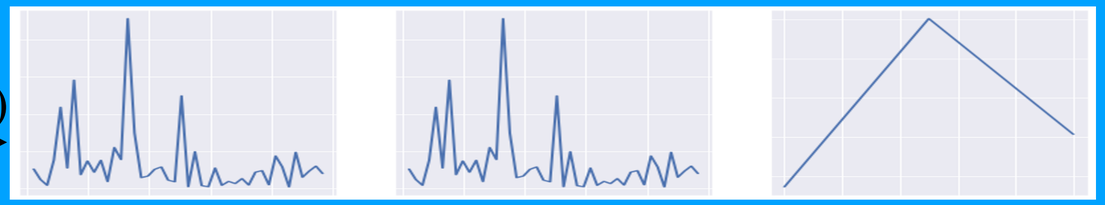


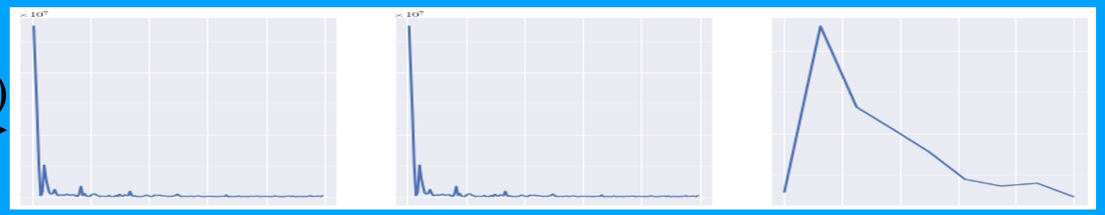
$\mathcal{F}(t)$

$\mathcal{F}(t)$

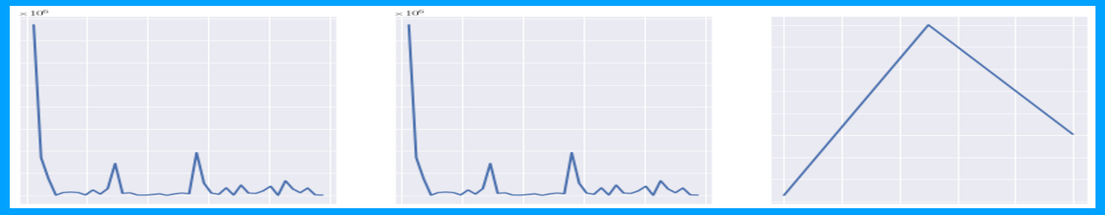
$\mathcal{F}(t)$

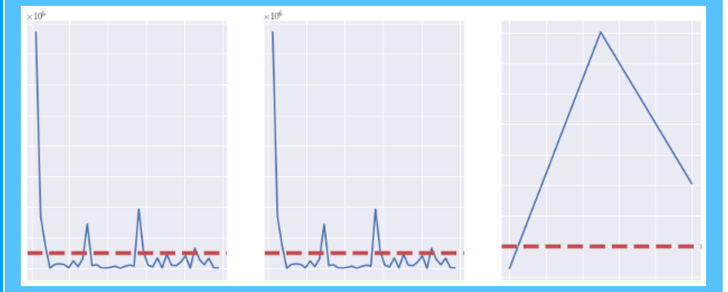






1



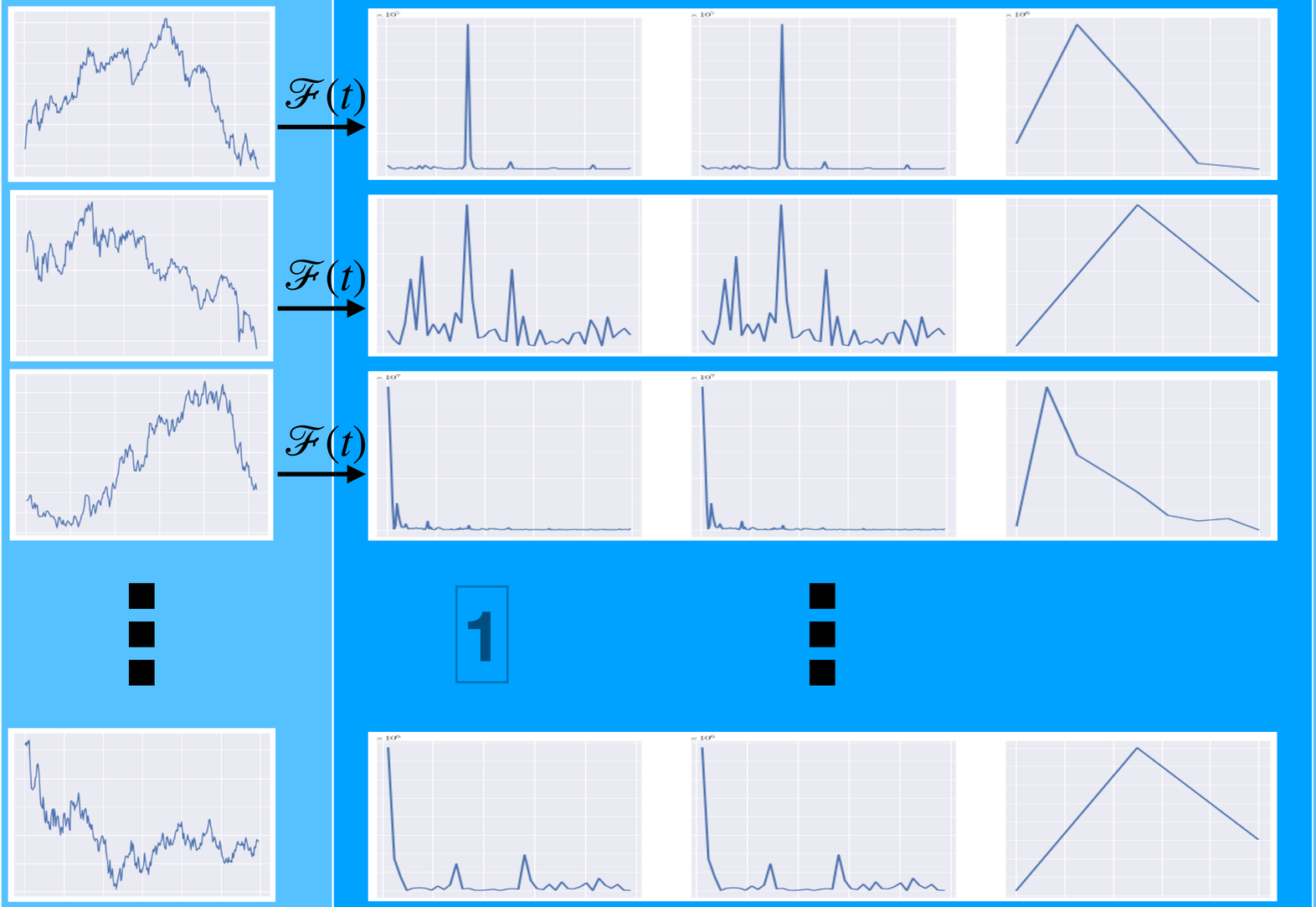


2

**Raw Time Series**

**Transform to Frequency Domain**

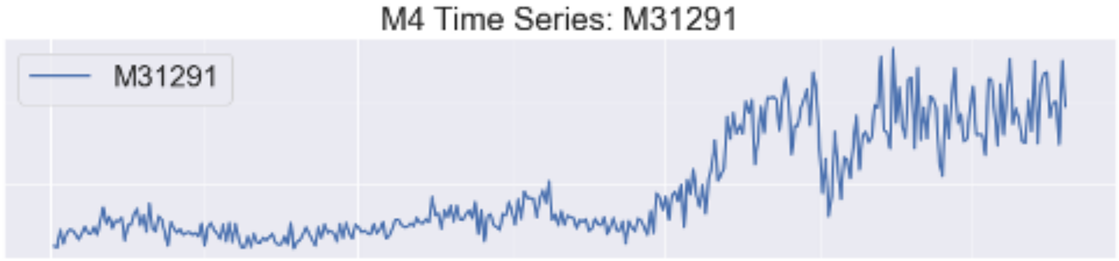
**Identify Top 5 Frequencies**

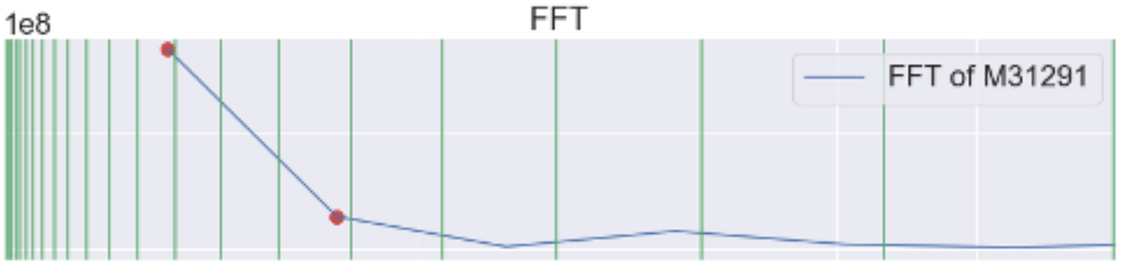

$\mathcal{F}(t)$
$\mathcal{F}(t)$
$\mathcal{F}(t)$
1
2
3
$10^{-2.2}$
$10^{-2.1}$
$10^{-1.5}$
$10^{-1.4}$
Raw Time Series
Transform to Frequency Domain
Identify Top 5 Frequencies
Compute Frequency Intervals

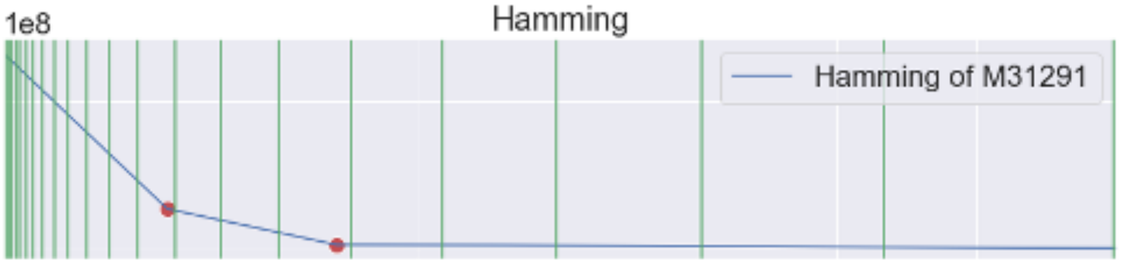
# Frequency Ranges

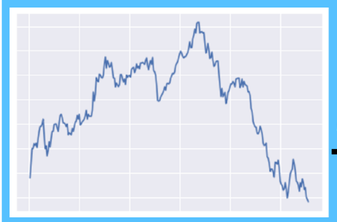
## Fitting Results into ranges

- Ranges hold values close to each other at the same index
- Allows for nominal classification of found frequencies
- Comparison via boolean logic as opposed to distance calculation

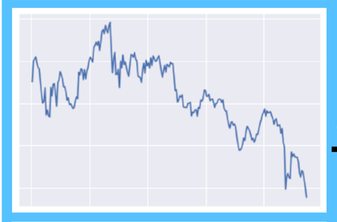


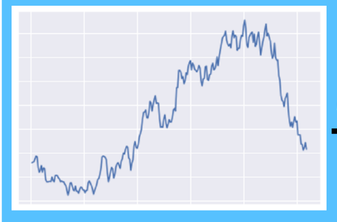



*Stats*



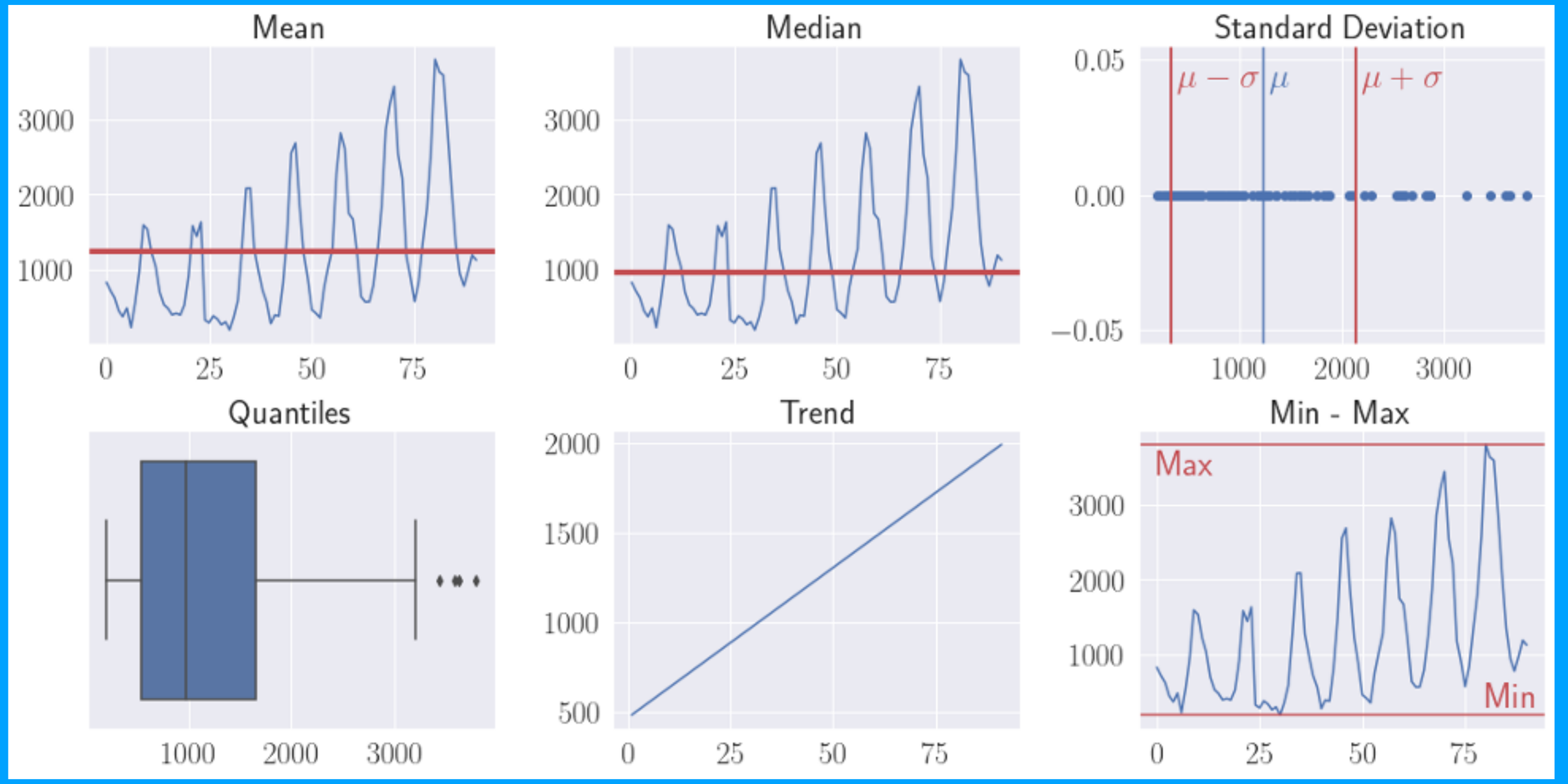
*Stats*



*Stats*

*Stats*

**Raw Time Series**




**Compute 9 Summary Statistics for each Time Series**

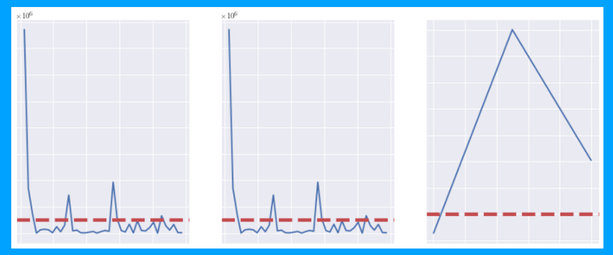
# Searching for a best matches of single time series

Template
Time Series

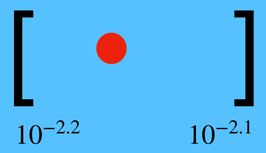
# Template Time Series

Template Time Series

$\mathcal{F}(t)$



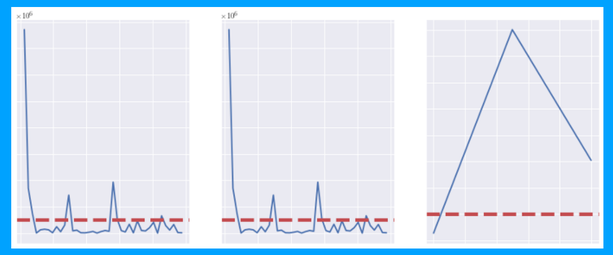
Identify Top K Frequencies



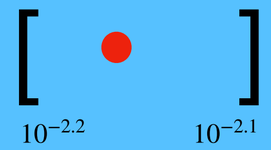

Transform to Frequency Domain

$\mathcal{F}(t)$

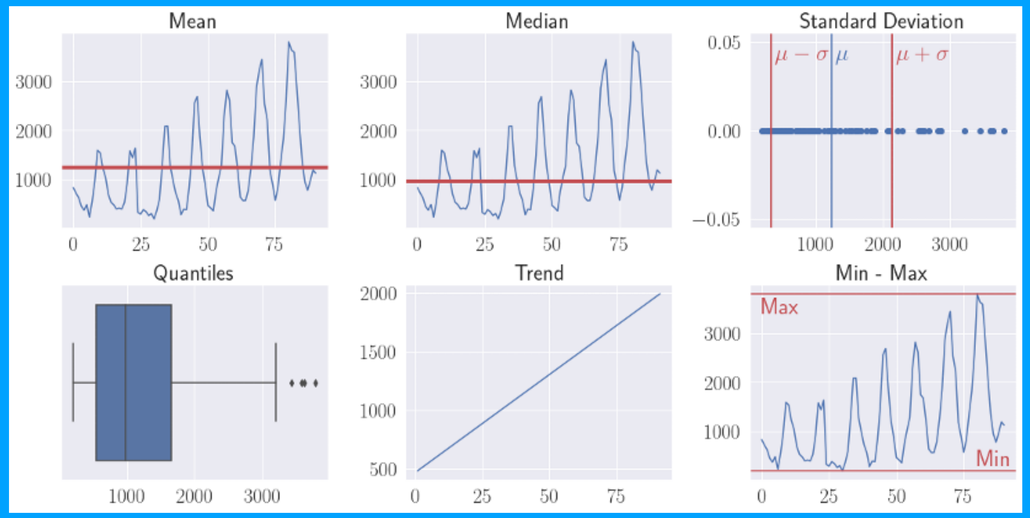
*Stats*



## Identify Top K Frequencies




## Transform to Frequency Domain



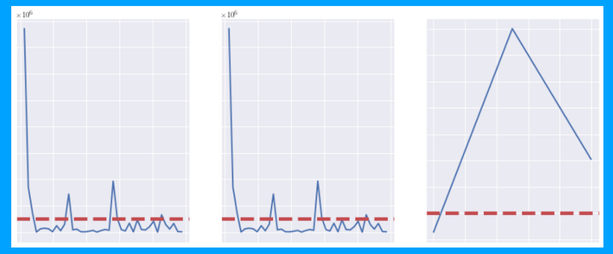

## Compute Summary Statistics

## Template Time Series

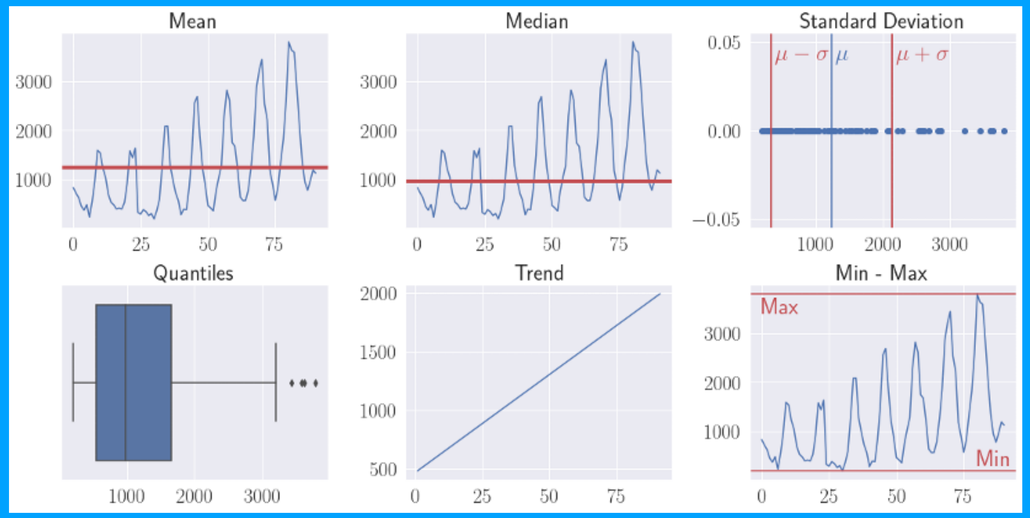
Template Time Series

$\mathcal{F}(t)$

Stats



Identify Top K Frequencies




Compute Summary Statistics

Transform to Frequency Domain

$$\begin{bmatrix} \bullet \end{bmatrix}$$

$10^{-2.2}$ $10^{-2.1}$

$$Templ\ [5,\ 9,\ 4,\ 7,\ 10]$$

$$Pool\ TS_1\ [5,\ 9,\ 4,\ 7,\ 10]$$

$$0*10^0 + 0*10^1 + 1*10^2 + 0*10^3 + 1*10^4$$

$$= 10100$$

$TS_1$ [3,4,6,8,10] *yields* 10100

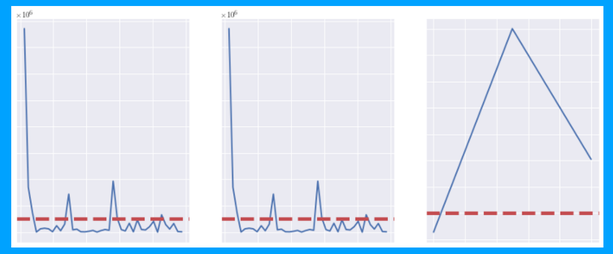
$TS_2$ [...] *yields* ...

$TS_n$ [...] *yields* ...

Compute Match Score

$\mathcal{F}(t)$

*Stats*



## Identify Top K Frequencies

## Transform to Frequency Domain

$$\begin{bmatrix} & \bullet & \\ 10^{-2.2} & & 10^{-2.1} \end{bmatrix}$$

$$Templ\ [5,\ 9,\ 4,\ 7,\ 10]$$

$$Pool\ TS_1\ [5,\ 9,\ 4,\ 7,\ 10]$$

$$0*10^0 + 0*10^1 + 1*10^2 + 0*10^3 + 1*10^4$$

$$= 10100$$

$TS_1$ [3,4,6,8,10] *yields* 10100

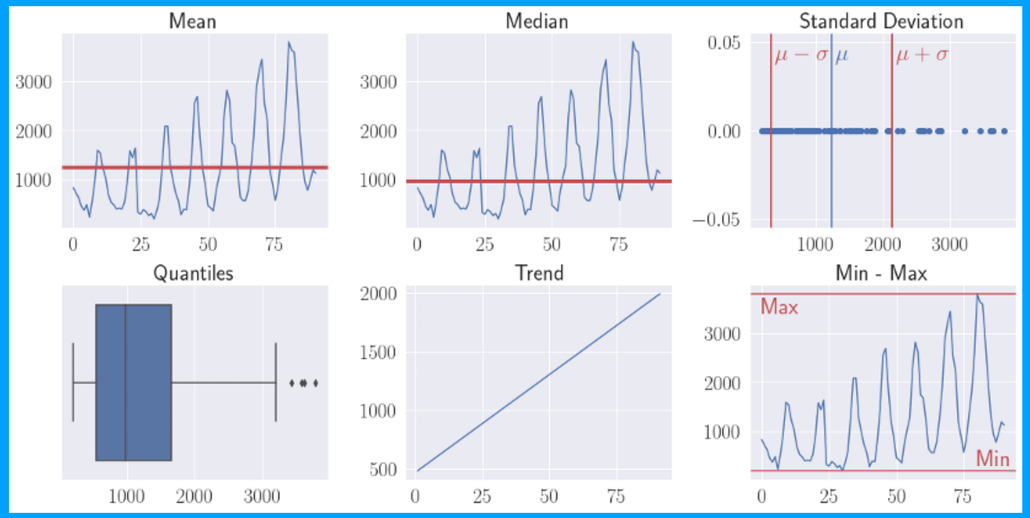
$TS_2$ [...] *yields* ...

$TS_n$ [...] *yields* ...

## Compute Match Score

| | FFT | Hamming | Welch |
|---|---|---|---|
| Best Match | 11010 | 10100 | 11011 |

**Highest Matches per Type**

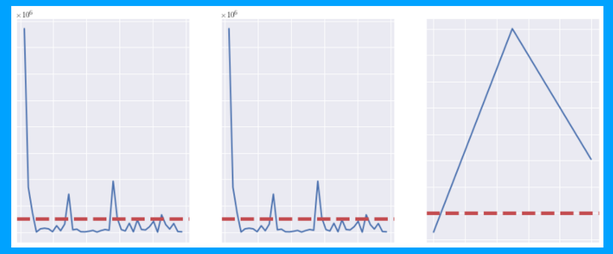



## Compute Summary Statistics

## Template Time Series

$\mathcal{F}(t)$

*Stats*



## Identify Top K Frequencies

## Transform to Frequency Domain

$$\begin{bmatrix} \bullet \end{bmatrix}$$

$10^{-2.2}$ $10^{-2.1}$

$$Templ\ [5,\ 9,\ 4,\ 7,\ 10]$$

$$Pool\ TS_1\ [5,\ 9,\ 4,\ 7,\ 10]$$

$$0*10^0 + 0*10^1 + 1*10^2 + 0*10^3 + 1*10^4$$

$$= 10100$$

$TS_1\ [3,4,6,8,10]$ *yields* 10100

$TS_2\ [\dots]$ *yields* …

$TS_n\ [\dots]$ *yields* …

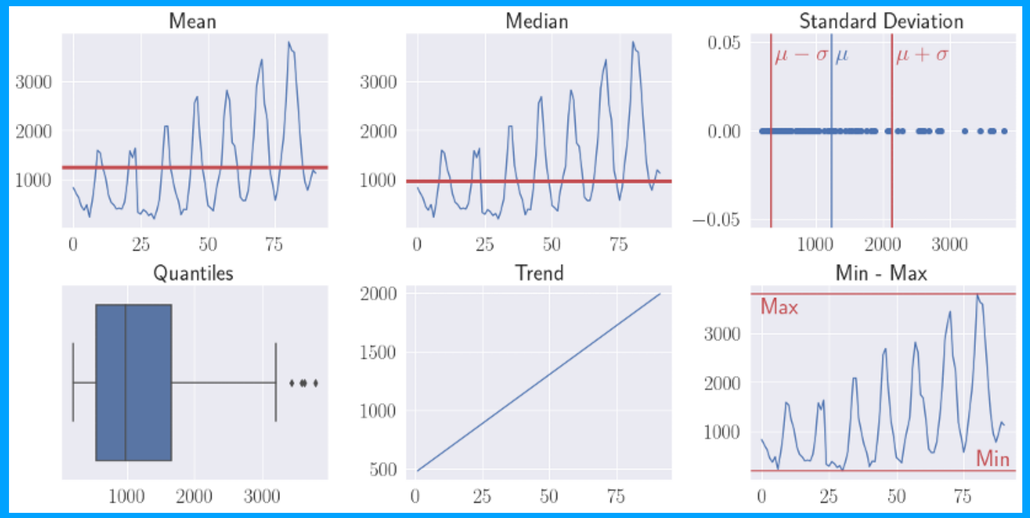
## Compute Match Score

| | FFT | Hamming | Welch |
|---|---|---|---|
| Best Match | 11010 | 10100 | 11011 |

**Highest Matches per Type**

$$A_{trend} = \{S_i \in S_n \mid 1\left(-\frac{m_{St}}{|m_{St}|} = -\frac{m_{S_i}}{|m_{S_i}|}\right)\}$$

**Match Slope Direction**



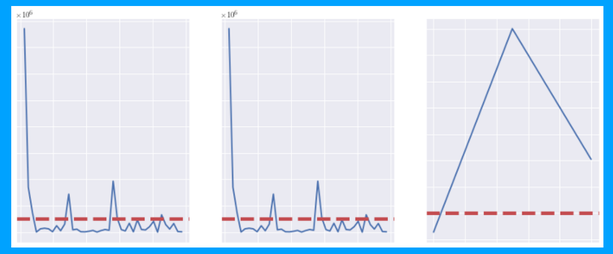

## Compute Summary Statistics

## Template Time Series

$\mathcal{F}(t)$

$Stats$

# Template Time Series

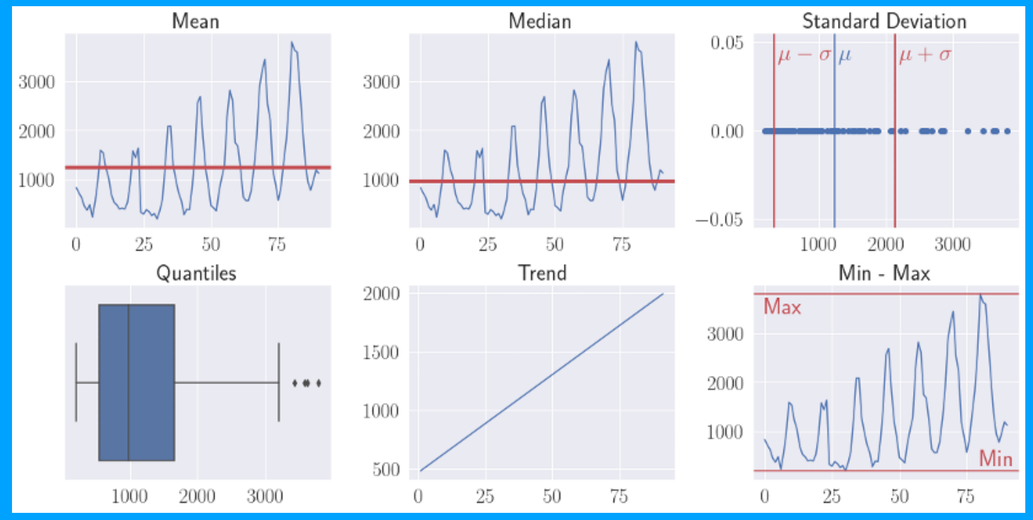


# Identify Top K Frequencies

$$\begin{bmatrix} \bullet \end{bmatrix}$$

$10^{-2.2}$ $10^{-2.1}$

# Transform to Frequency Domain




# Compute Summary Statistics

$$Templ\ [5,\ 9,\ 4,\ 7,\ 10]$$

$$Pool\ TS_1\ [5,\ 9,\ 4,\ 7,\ 10]$$

$$0*10^0 + 0*10^1 + 1*10^2 + 0*10^3 + 1*10^4$$

$$= 10100$$

$TS_1\ [3,4,6,8,10]$ yields 10100

$TS_2\ [\dots]$ yields $\dots$

$TS_n\ [\dots]$ yields $\dots$

# Compute Match Score

| | FFT | Hamming | Welch |
|---|---|---|---|
| Best Match | 11010 | 10100 | 11011 |

## Highest Matches per Type

$$A_{trend} = \{S_i \in S_n \mid 1\left(-\frac{m_{St}}{|m_{St}|} = -\frac{m_{S_i}}{|m_{S_i}|}\right)\}$$

## Match Slope Direction

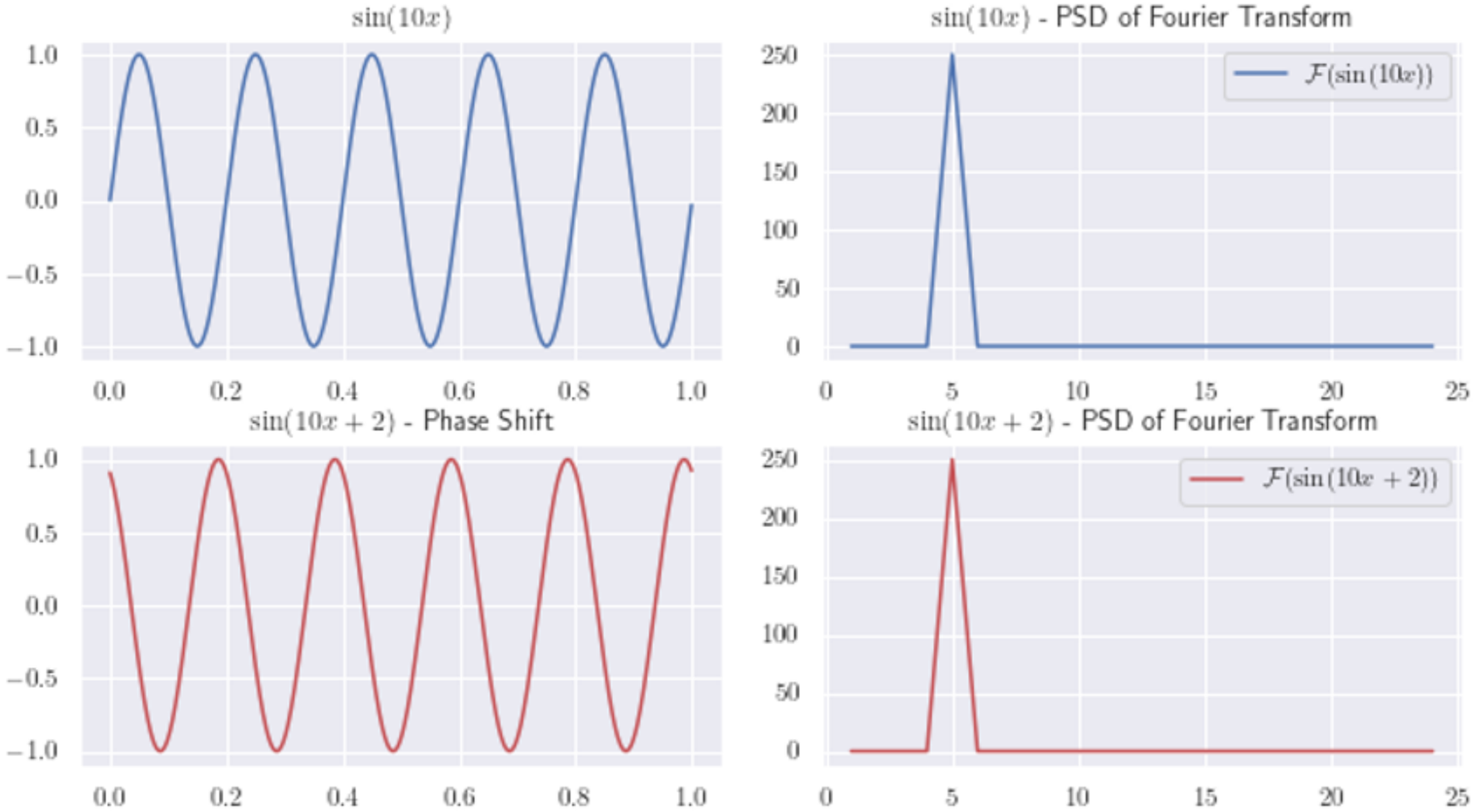
$$arg\ min\ f(S_i) := |\phi_{S_t} - \phi_{S_i}|$$

## Minimize Statistical Metric

# Challenges

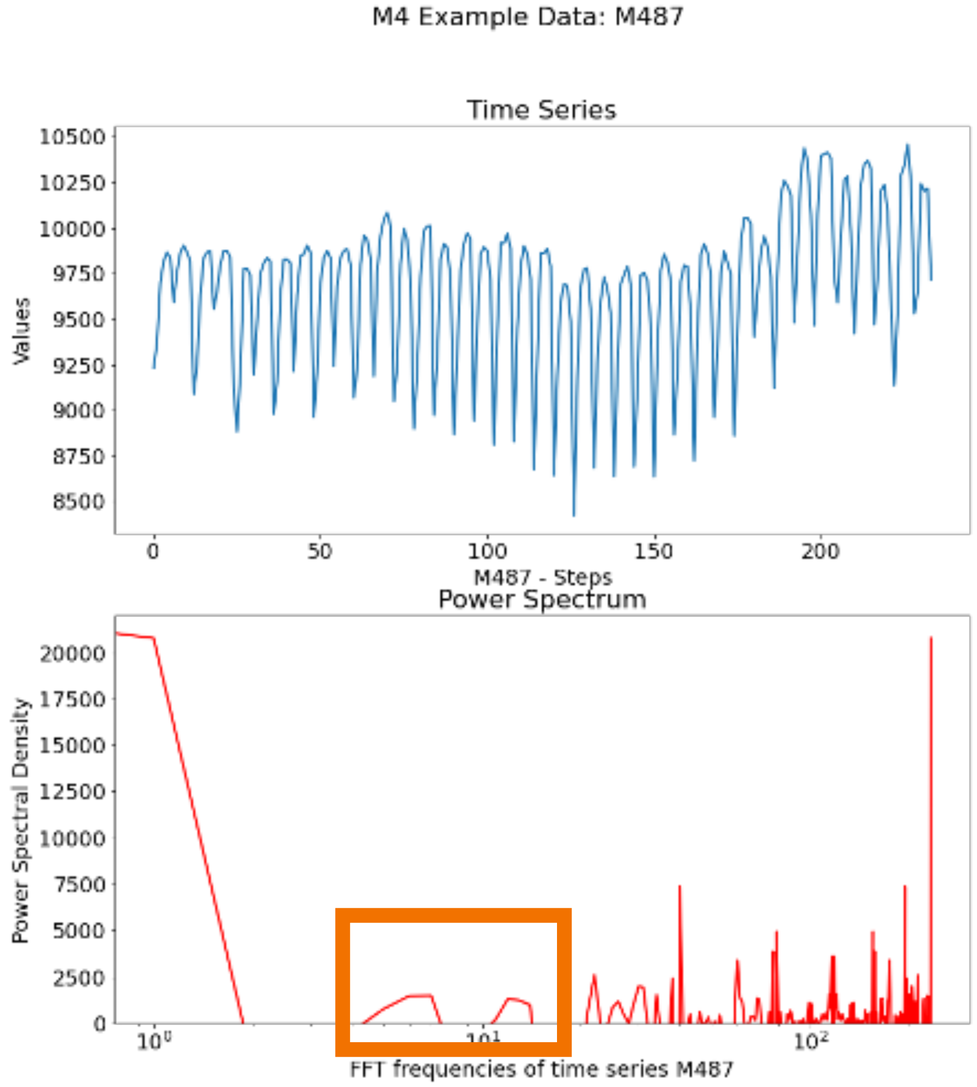
# Challenges

## Phase Shift



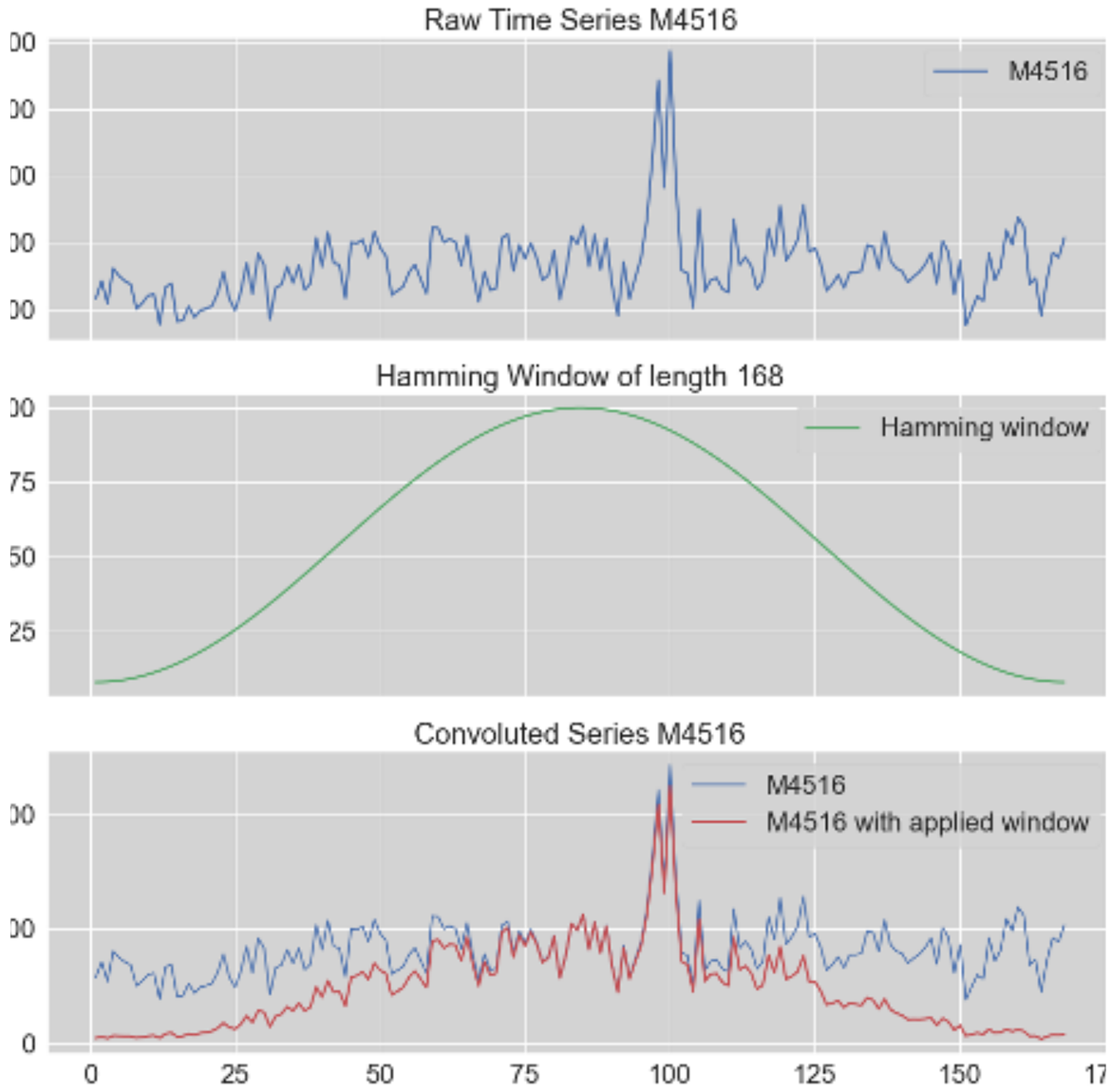
# Challenges

## Spectral Leakage



# Challenges

## Hamming Window as Solution

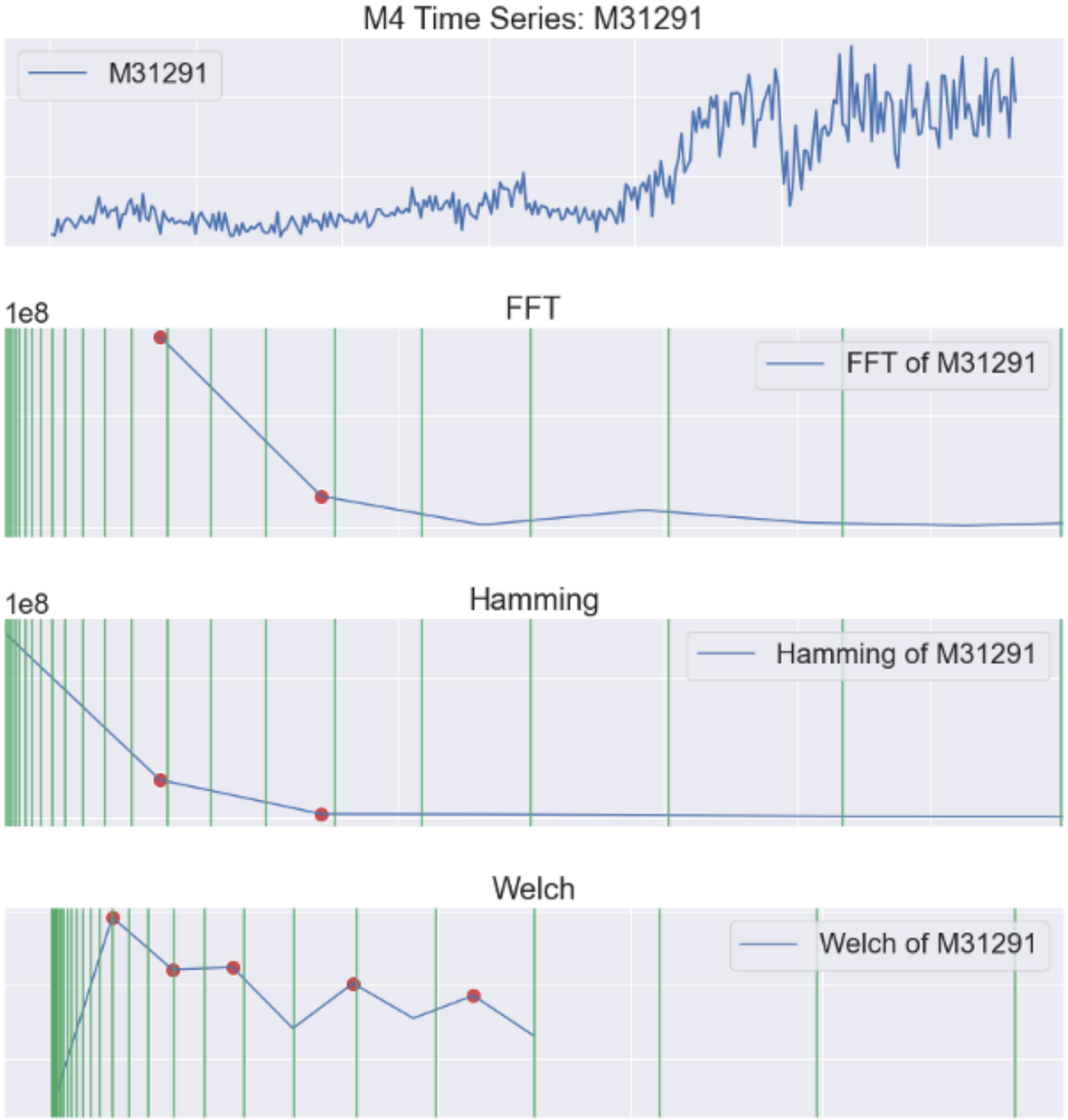


# Window Functions

## Welch’s Window as Solution

# Challenges

## Intervals as Solution



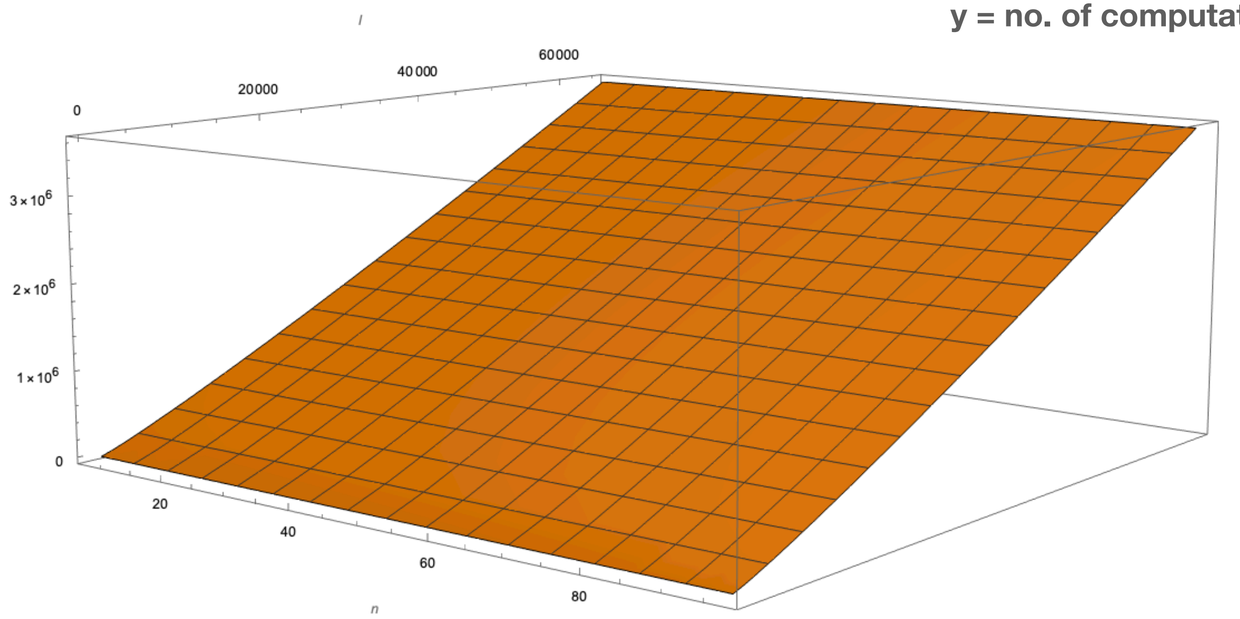
# Demo

# Time Complexity

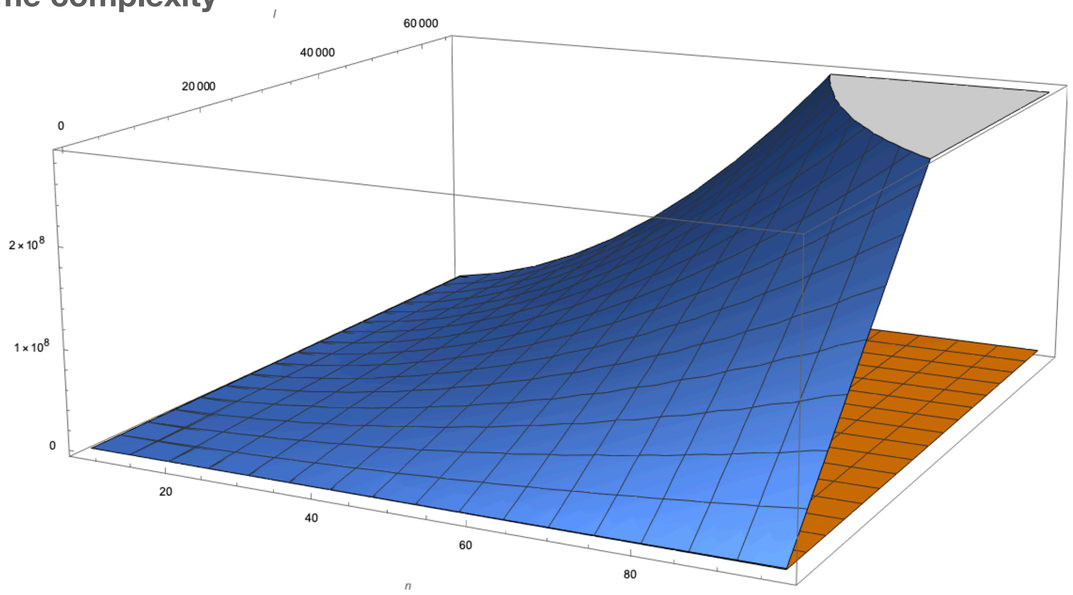
# Time Complexity

## Time Series Search Algorithm

l = no. of series in pool
n = no. of time steps
y = no. of computations / time complexity




$\mathcal{O}(n \log n)$

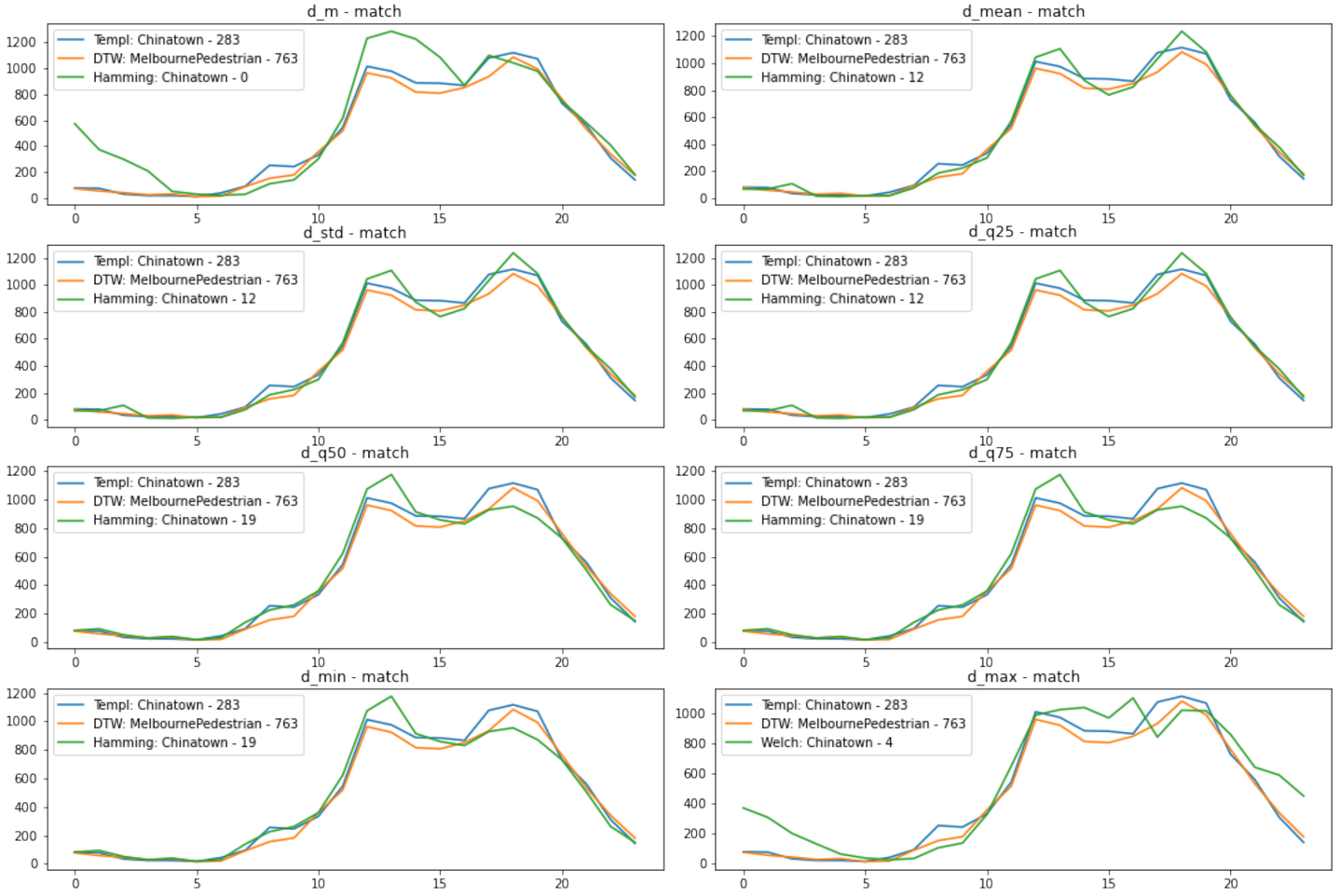



$\mathcal{O}(ln^2)$

# Results Overview

# Results Overview

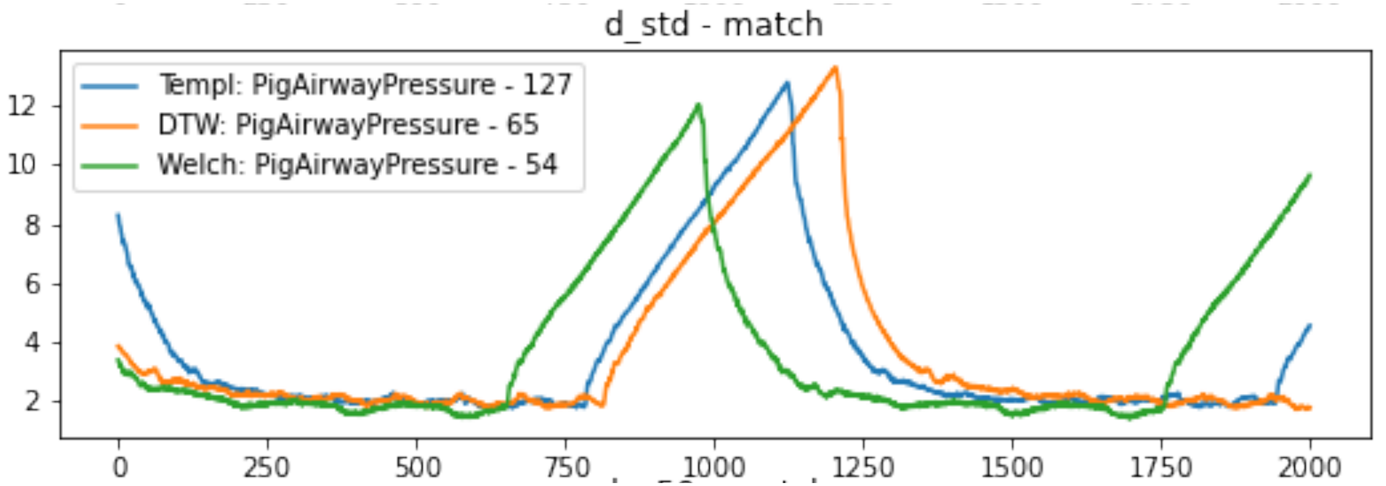
## General Overview

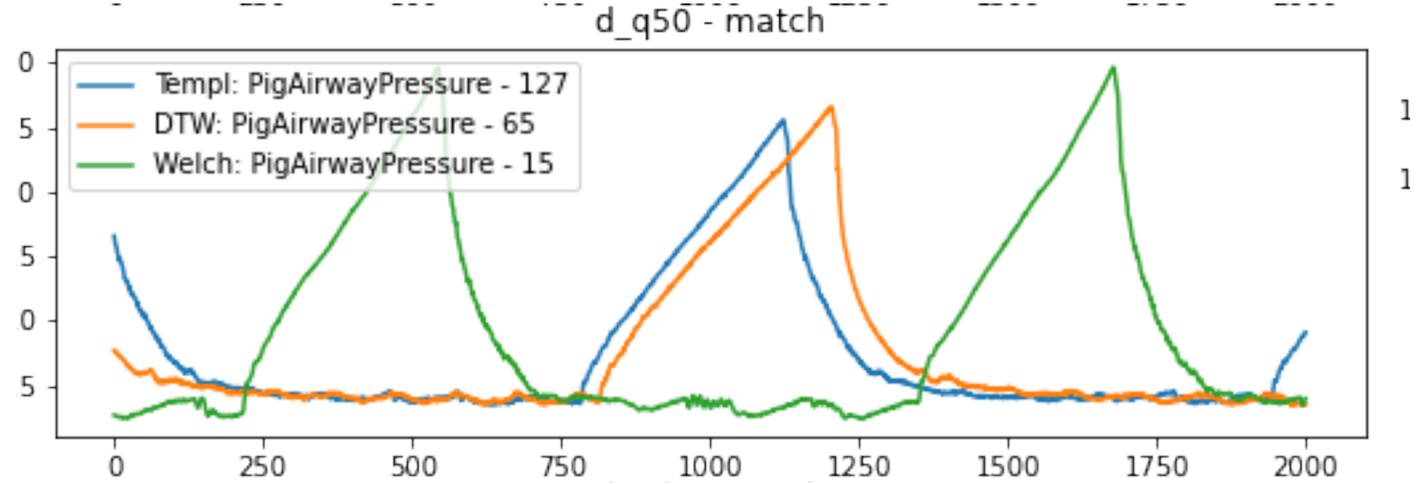


# Results Overview

## Comparison to DTW

- Results can be comparable to DTW
- Depends on customizing
- Phase shift and wrong window function statistic can lead on undesired results
-

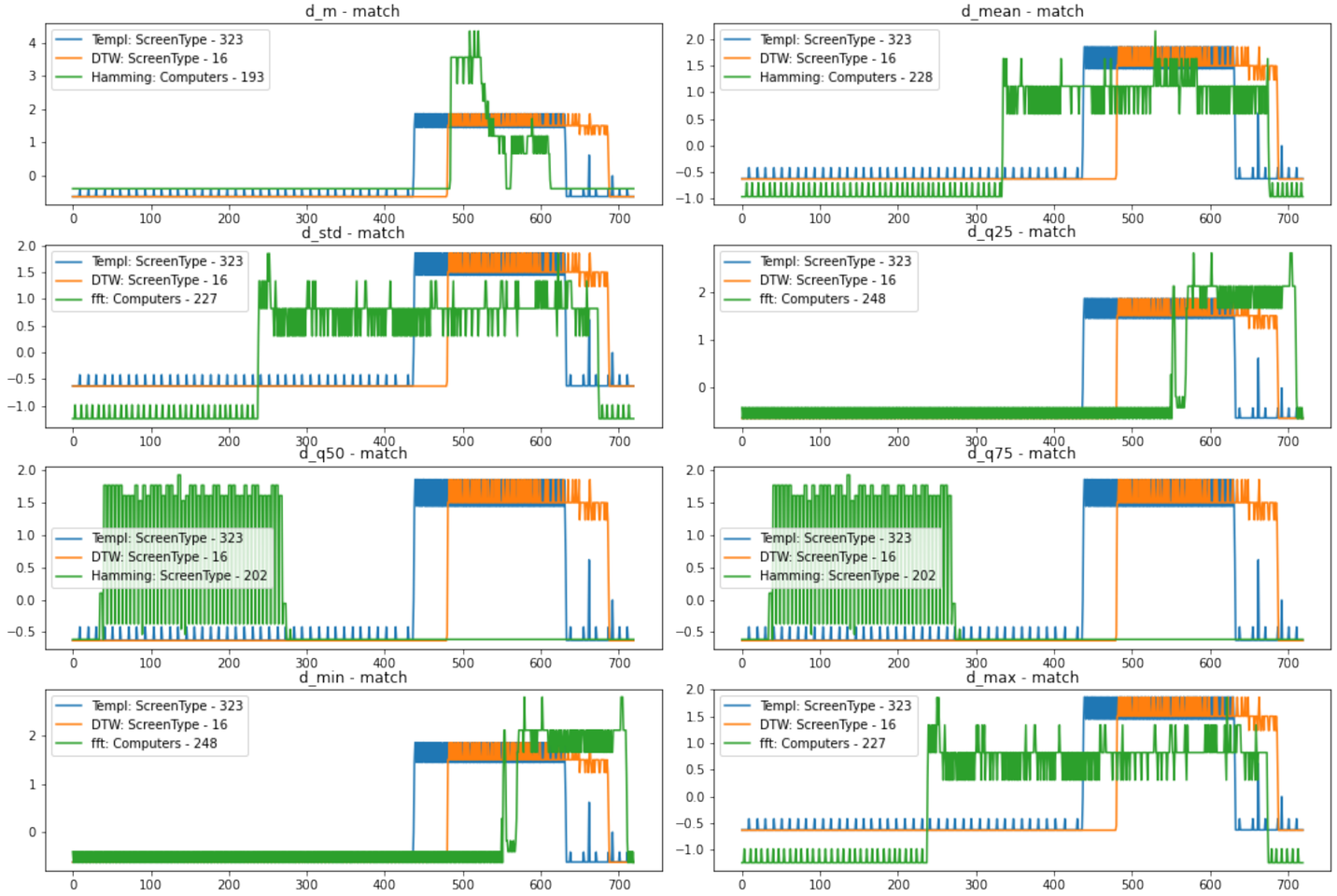


# Results Overview

## Matching of Frequencies

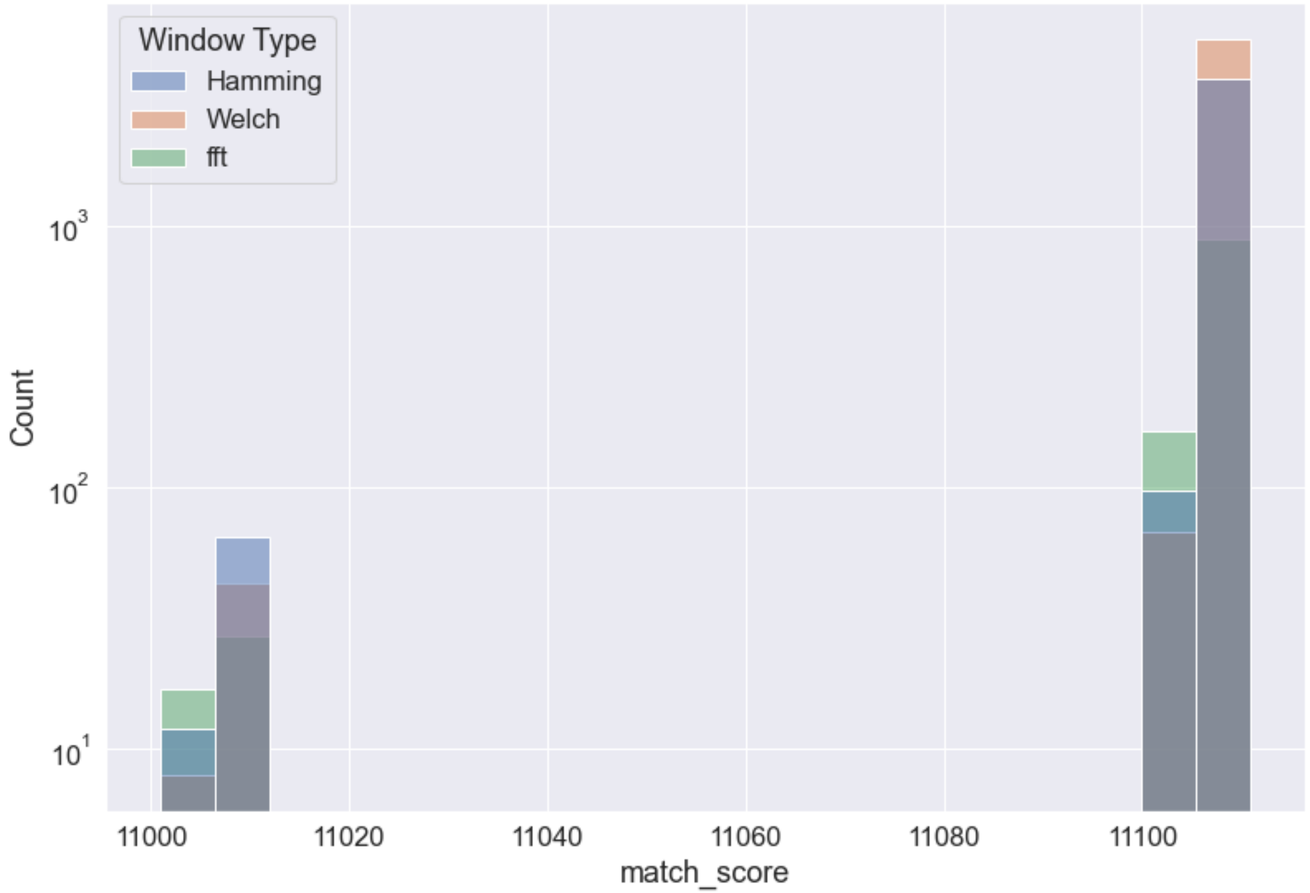
- Current approach to frequencies can be improved upon
- Good match on lower frequencies



# Results Overview

## Window Types

- Match scores indicate that windows generally lead to more precise matchings (higher scores)
- Welch's method beneficial to combat spectral leakage (especially on non-sinusoidal data patterns)

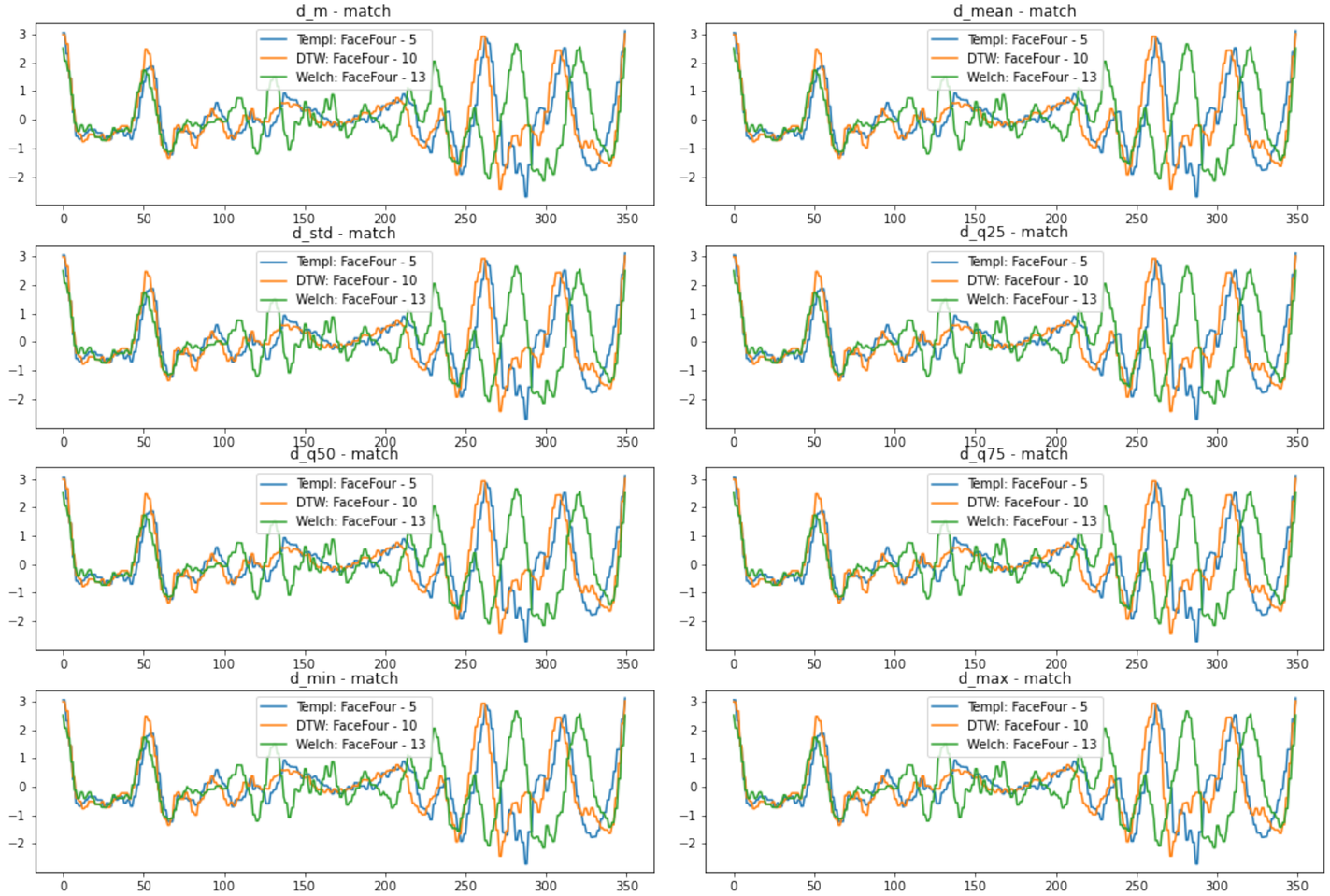


# Discussion and Future Work

# Discussion and Future Work

## Outlook

- FFT together with window functions and summary statistic is **powerful tool**
- Scale-invariant
- Struggles with **loss of temporal information in frequency domain**
- Summary statistics do not deliver **silver bullet**



# Discussion and Future Work

## Outlook - Frequency Domain

# Thank You!