

# DATA MINING 2

## Time Series - Similarities & Distances

---

Riccardo Guidotti

a.a. 2023/2024

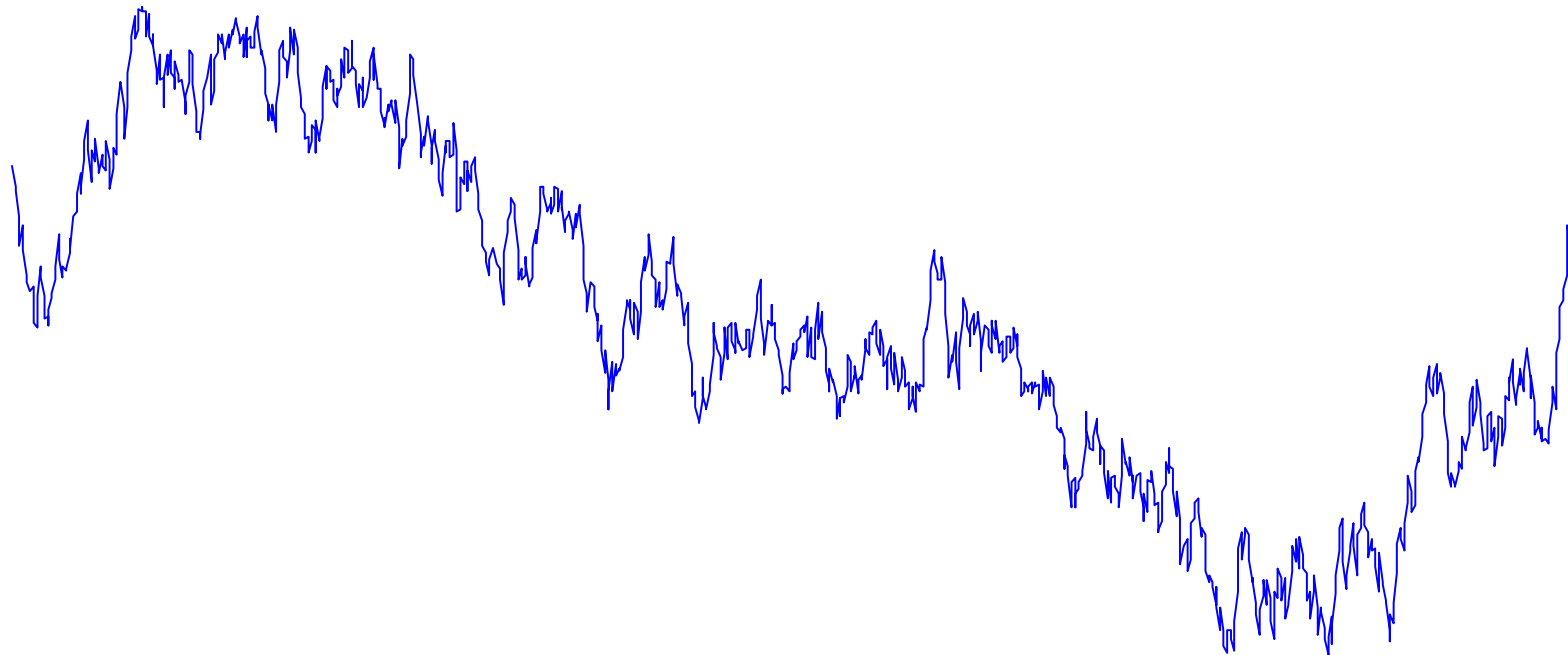
Slides edited from Keogh Eamonn's tutorial



UNIVERSITÀ DI PISA

# What is a Time Series?

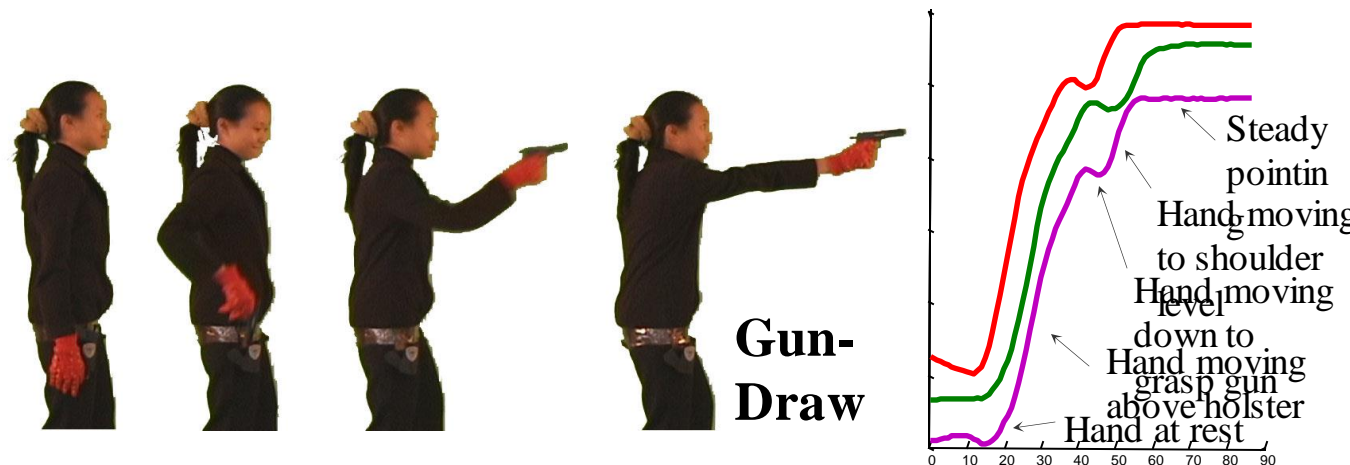
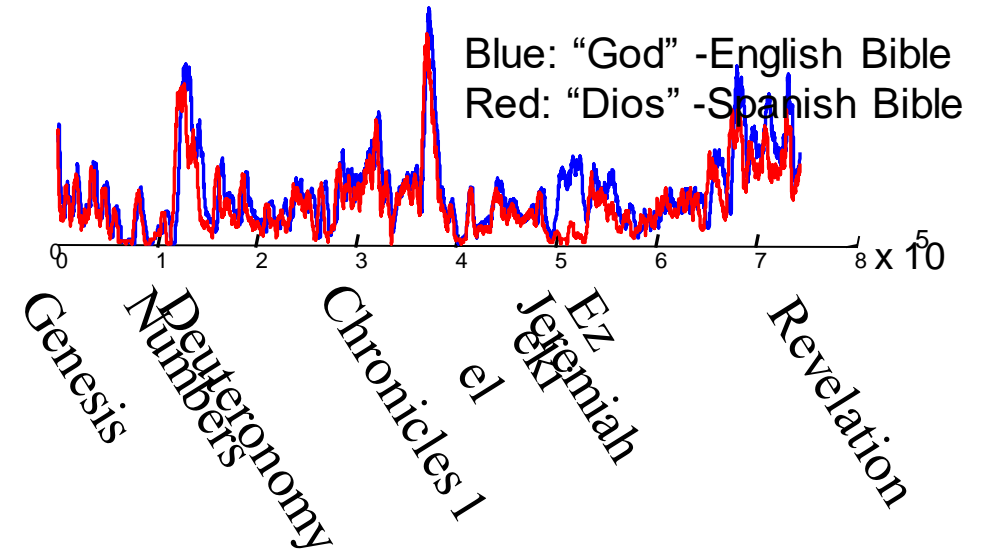
- A time series is a collection of observations made sequentially in time, generally at constant time intervals.



25.1750  
25.2250  
25.2500  
25.2500  
25.2750  
25.3250  
25.3500  
25.3500  
25.4000  
25.4000  
25.3250  
25.2250  
25.2000  
25.1750  
...  
24.6250  
24.6750  
24.6750  
24.6250  
24.6250  
24.6250  
24.6750  
24.7500

# Time Series are Ubiquitous

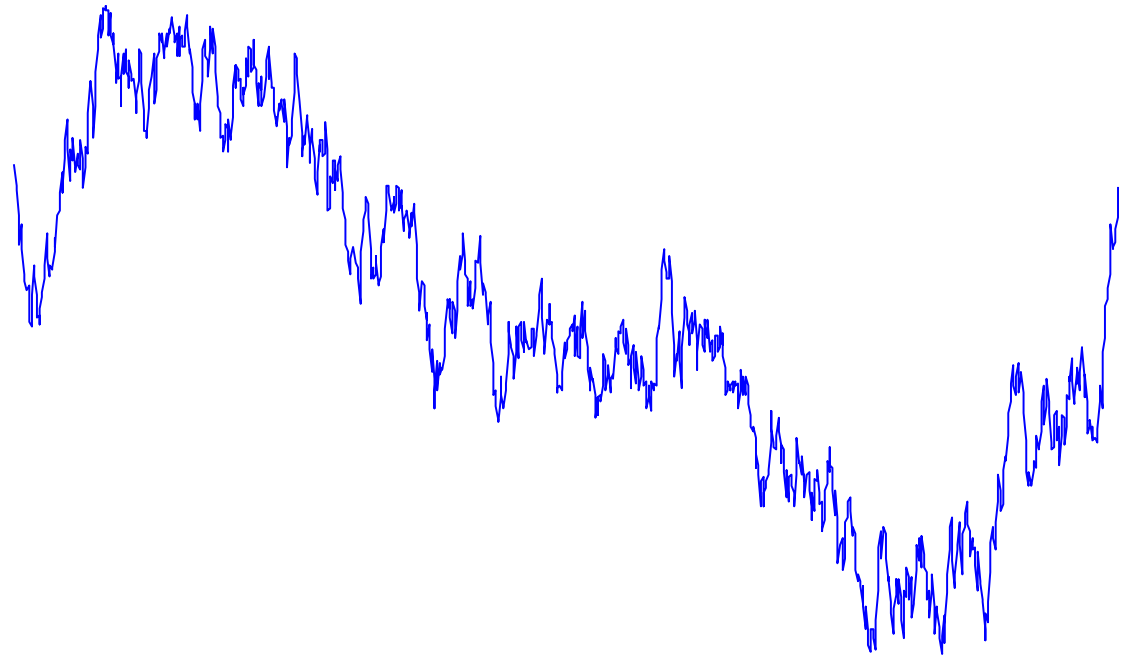
- You can measure many things ... and things change over time.
  - Blood pressure
  - Donald Trump's popularity rating
  - The annual rainfall in Pisa
  - The value of your stocks
- In addition other data type can thought of as time series
  - Text data: words count
  - Images: edges displacement
  - Videos: object positioning



# Problems in Working with Time Series

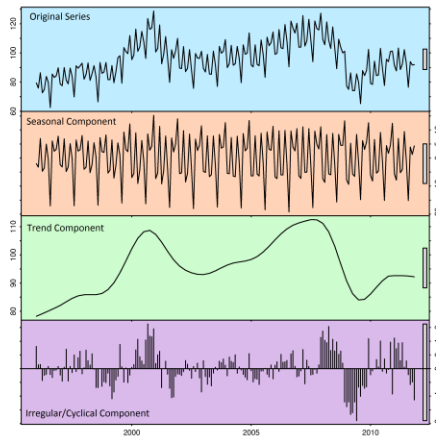
---

- Large amount of data.
- Similarity is not easy to estimate.
- Different data formats.
- Different sampling rates.
- Noise, missing values, etc.

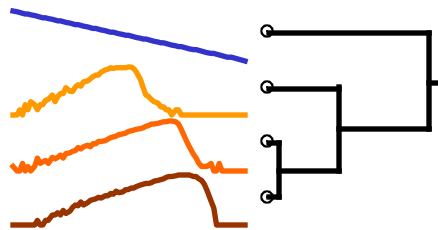


# What We Can Do With Time Series?

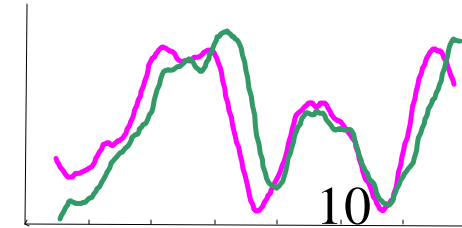
- Trends, Seasonality



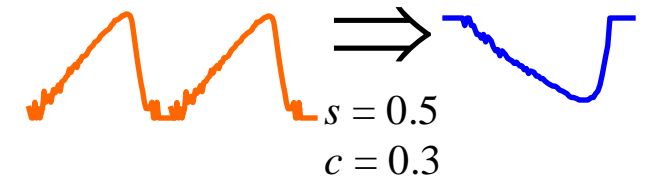
- Clustering



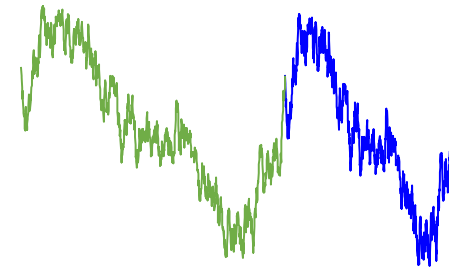
- Motif Discovery



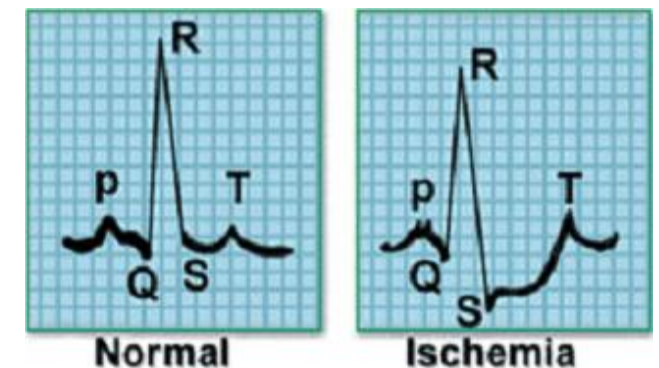
- Rule Discovery



- Forecasting

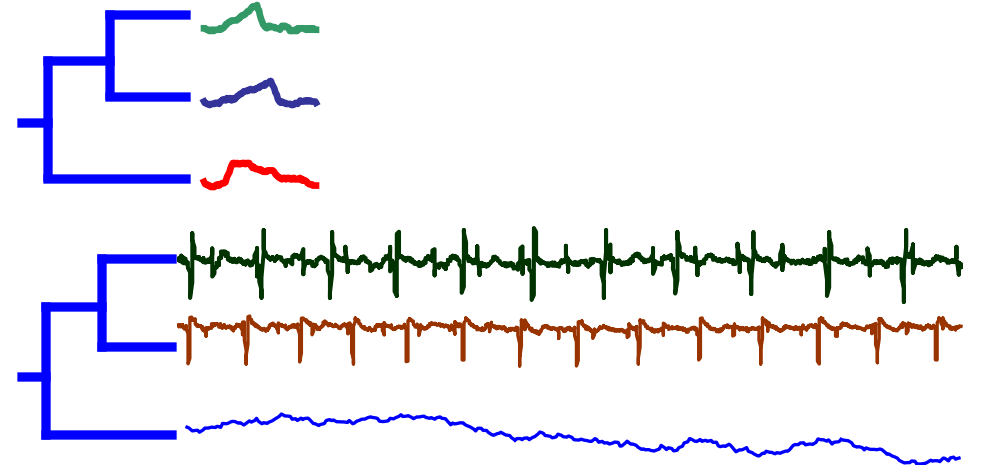


- Classification



# Similarity

- All these problems require similarity matching.
- What is Similarity?
  - It is the quality or state of being similar, likeness, resemblance, as a similarity of features.
- In time series analysis we recognize two kinds of similarity:
  - Similarity at the level of *shape*
  - Similarity at the *structural level*

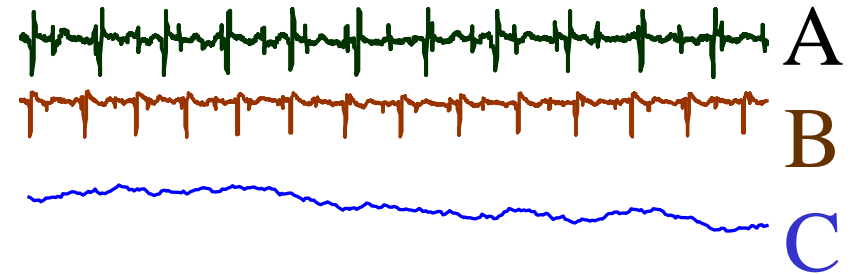


# Structural-based Similarities

---

# Structure or Model Based Similarity

- For long time series, shape-based similarity give very poor results.
- We need to measure similarity based on high level structure.
- The basic idea is to:
  1. extract *global* features from the time series,
  2. create a feature vector, and
  3. use it to measure similarity and/or classify
- Example of features:
  - mean, variance, skewness, kurtosis,
  - 1<sup>st</sup> derivative mean, 1<sup>st</sup> derivative variance, ...
  - parameters of regression, forecasting, Markov model



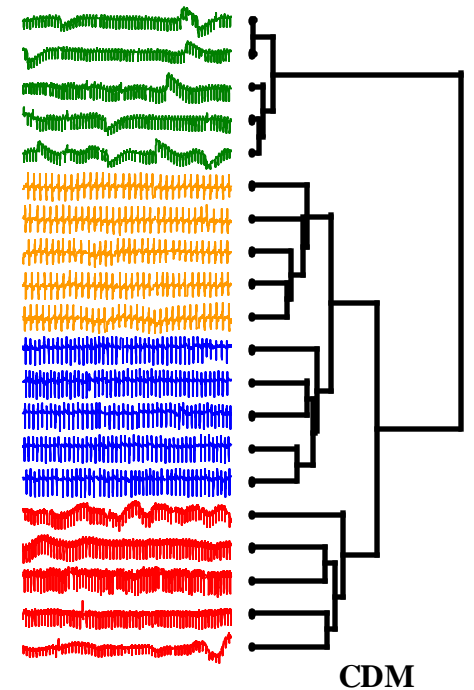
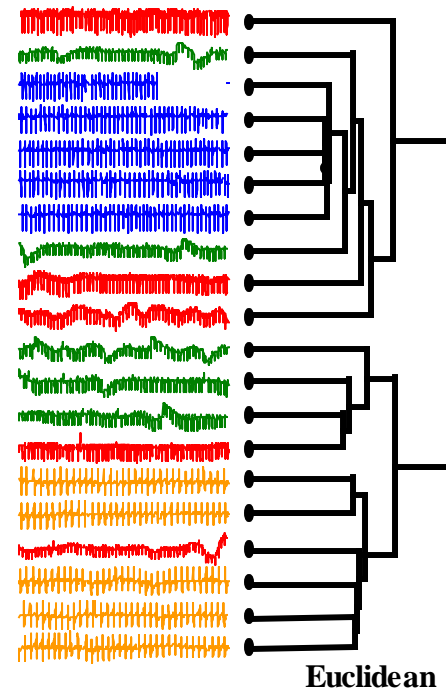
Feature\Time Series	A	B	C
Max Value	11	12	19
Mean	5.3	6.4	4.8
Min Value	3	2	5
Autocorrelation	0.2	0.3	0.5
...	...	...	...



# Compression Based Dissimilarity

- Use as features whatever structure a given compression algorithm finds.

- $d(x, y) = CDM(x, y) = \frac{C(x, y)}{C(x) + C(y)}$

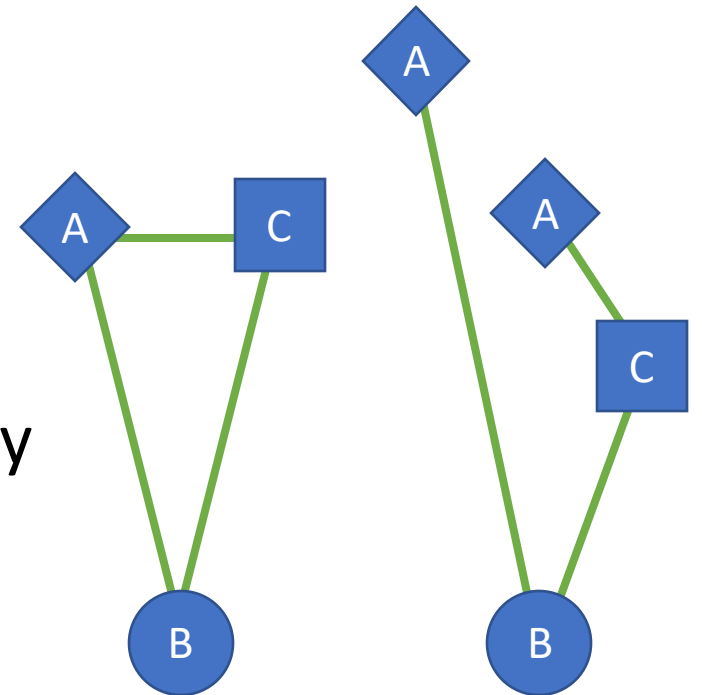


# Shape-based Similarities

---

# Defining Distance Measures

- Let A and B be two objects from the universe of possible objects. The distance (dissimilarity) is denoted by  $D(A,B)$ .
- Properties in a distance measure.
  - $D(A,B) = D(B,A)$  Symmetry
  - $D(A,A) = 0$  Constancy
  - $D(A,B) = 0$  iff  $A = B$  Positivity
  - $D(A,B) \leq D(A,C) + D(B,C)$  Triangular Inequality



# Euclidean Distance

- Given two time series:

- $Q = q_1 \dots q_n$

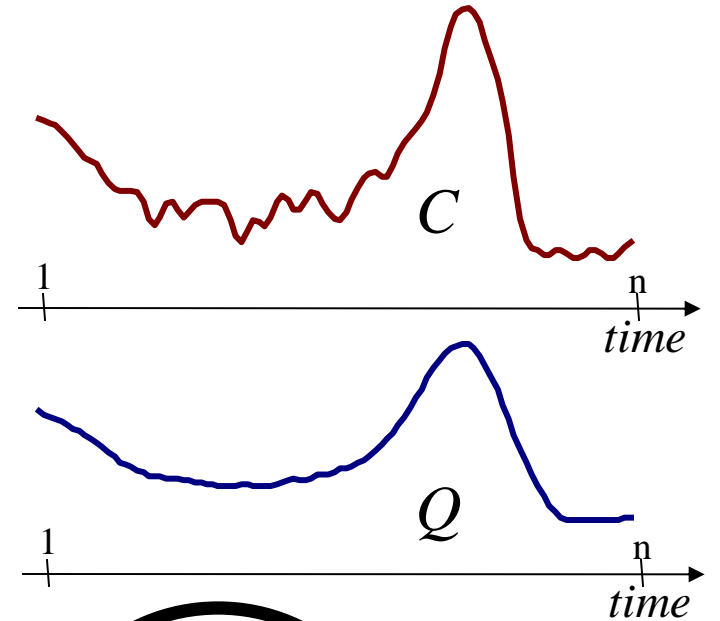
- $C = c_1 \dots c_n$

$$D(Q, C) \equiv \sqrt{\sum_{i=1}^n (q_i - c_i)^2}$$

- $T1 = \langle 56, 176, 110, 95 \rangle$

- $T2 = \langle 36, 126, 180, 80 \rangle$

$$D(T1, T2) = \text{sqrt} [ (56-36)^2 + (176-126)^2 + (110-180)^2 + (95-80)^2 ]$$

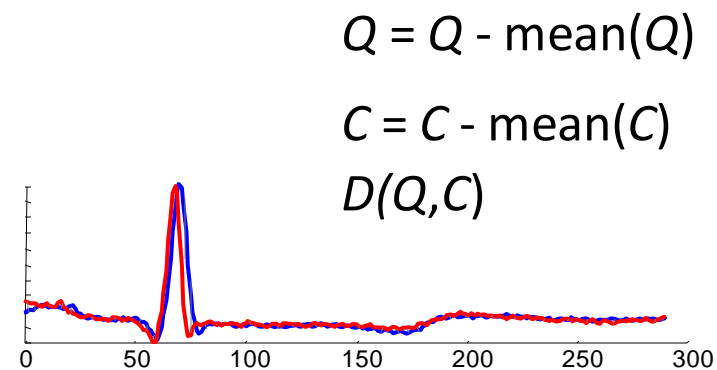
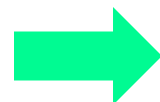
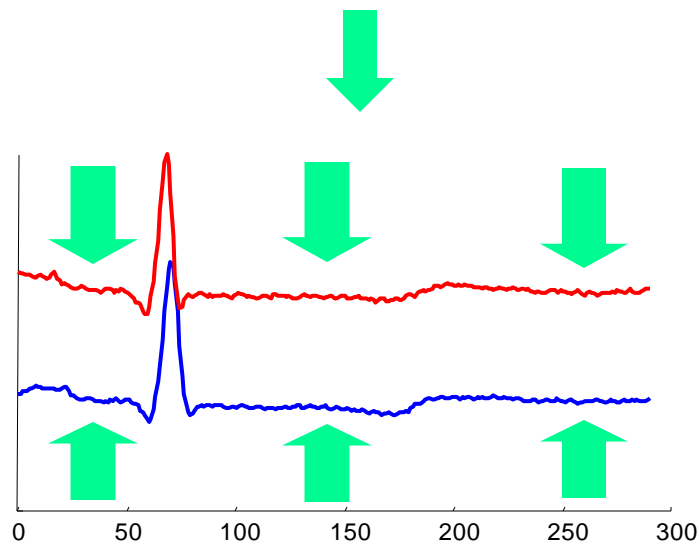
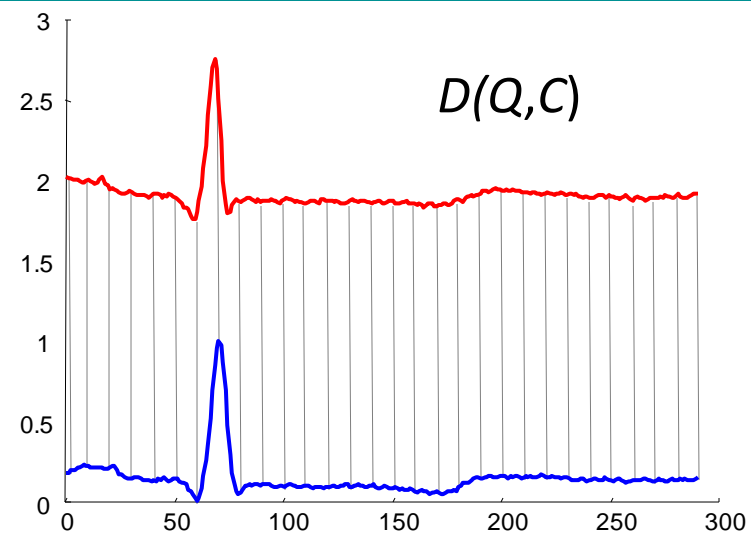
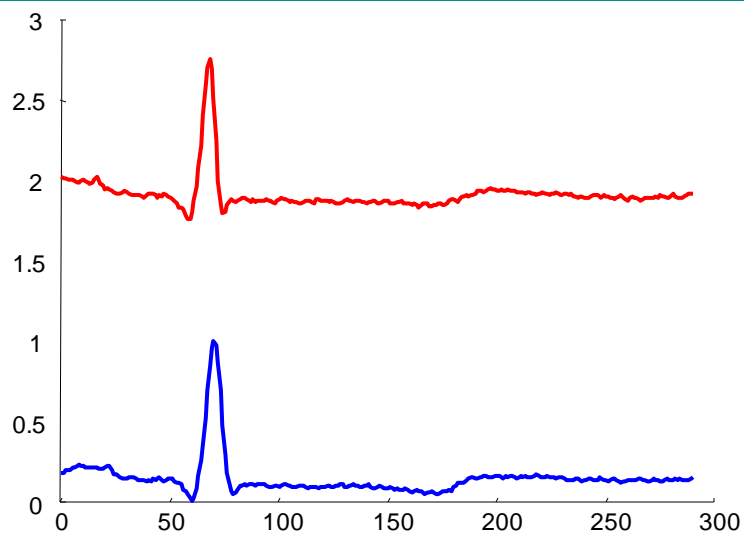


# Problems with Euclidean Distance

---

- Euclidean distance is very sensitive to “distortions” in the data.
- These distortions are dangerous and should be removed.
- Most common distortions:
  - Offset Translation
  - Amplitude Scaling
  - Linear Trend
  - Noise
- They can be removed by using the appropriate transformations.

# Transformation I: Offset Translation

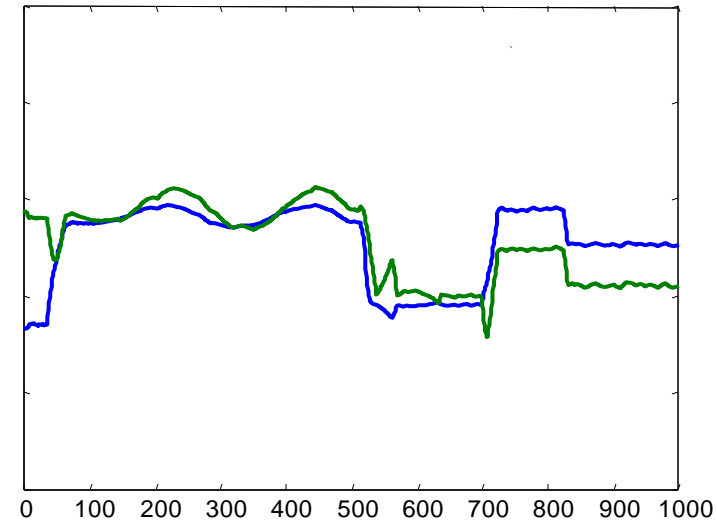
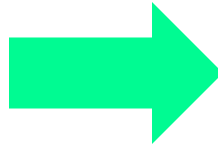
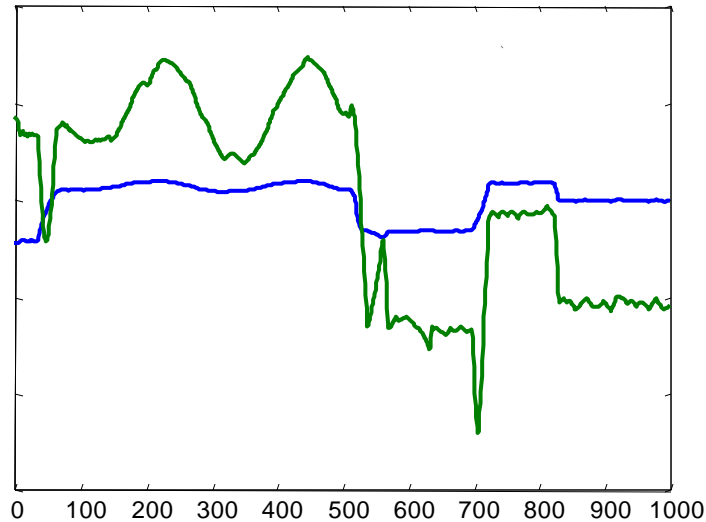


$$Q = Q - \text{mean}(Q)$$

$$C = C - \text{mean}(C)$$

$$D(Q,C)$$

# Transformation II: Amplitude Scaling



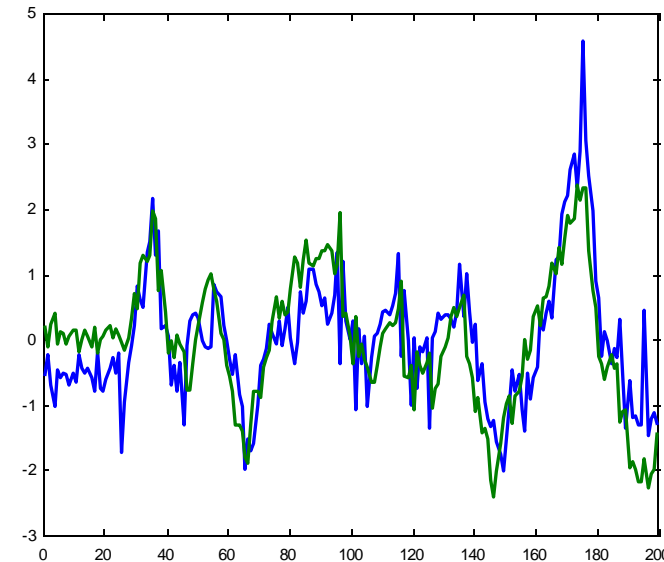
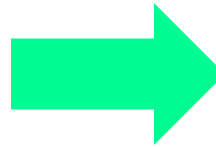
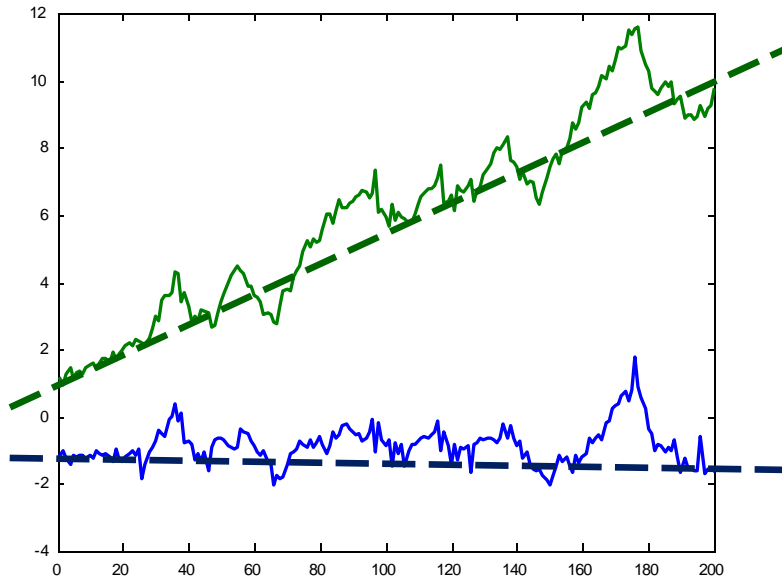
$$Q = (Q - \text{mean}(Q)) / \text{std}(Q)$$

$$C = (C - \text{mean}(C)) / \text{std}(C)$$

$$D(Q, C)$$

# Transformation III: Linear Trend

- Removing linear trend: fit the best fitting straight line to the time series, then subtract that line from the time series.

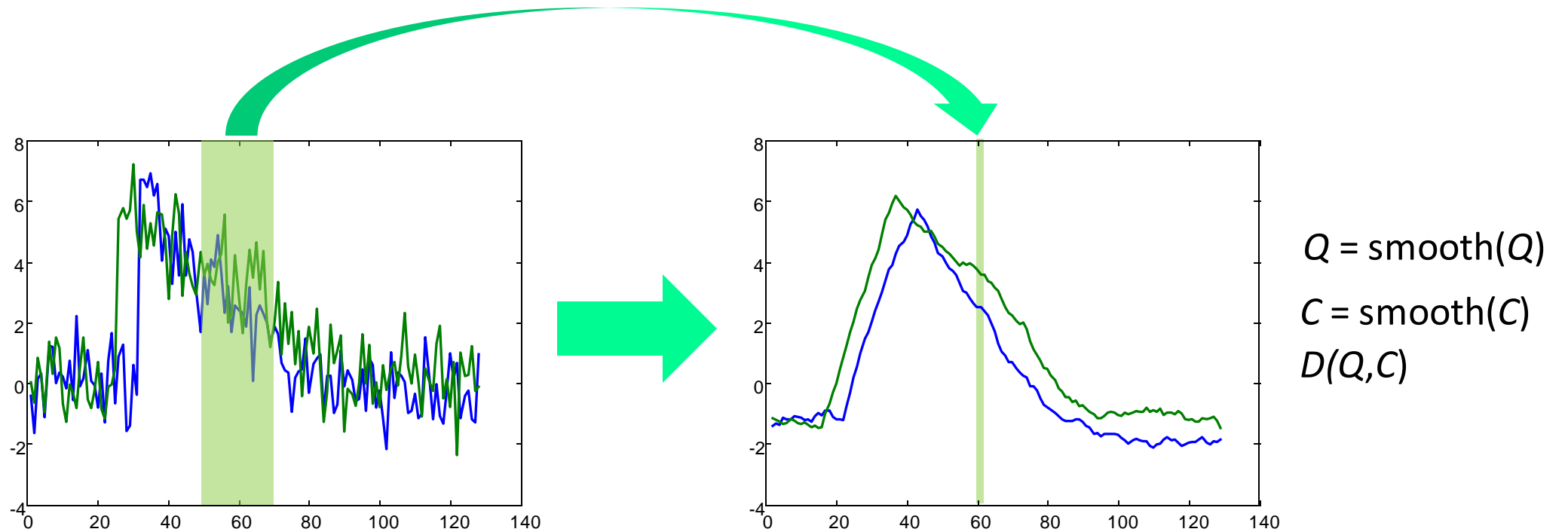


Removed linear trend,  
offset translation,  
amplitude scaling



# Transformation IV: Noise

- The intuition behind removing noise is to average each datapoints value with its neighbors.



# Moving Average

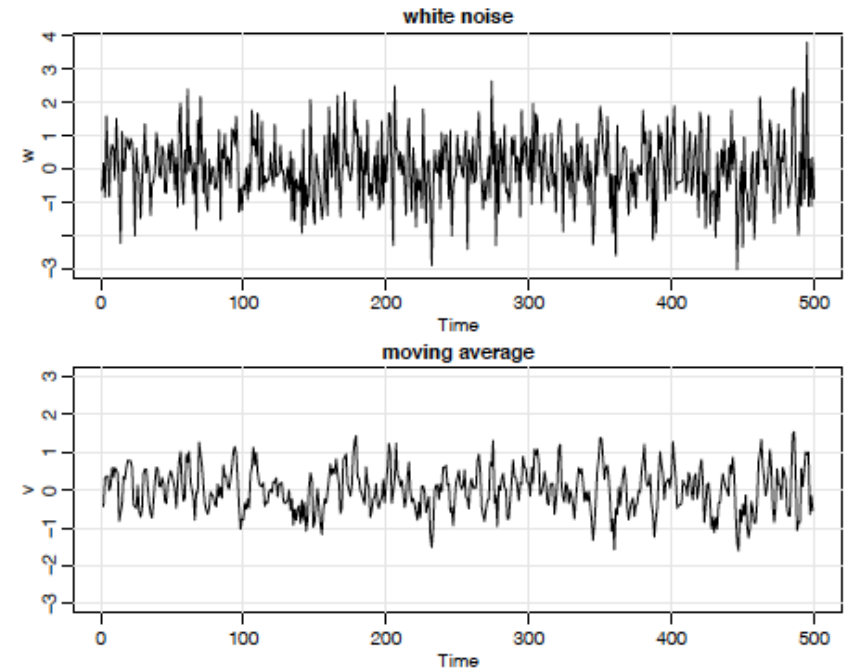
- Noise can be removed by a **moving average** (MA) that smooths the TS.
- Given a window of length  $w$  and a TS  $t$ , the MA is applied as follows

- $t_i = \frac{1}{w} \sum_{j=i-w/2}^{w/2} t_j$  for  $i = 1, \dots, n$

- For example, if  $w=3$  we have

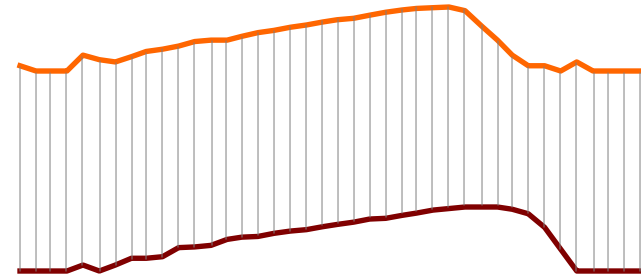
- $t_i = \frac{1}{3} (t_{i-1} + t_i + t_{i+1})$

		w=3
time	value	ma
t1	20	-
t2	24	22.0
t3	22	24.0
t4	26	24.3
t5	25	-

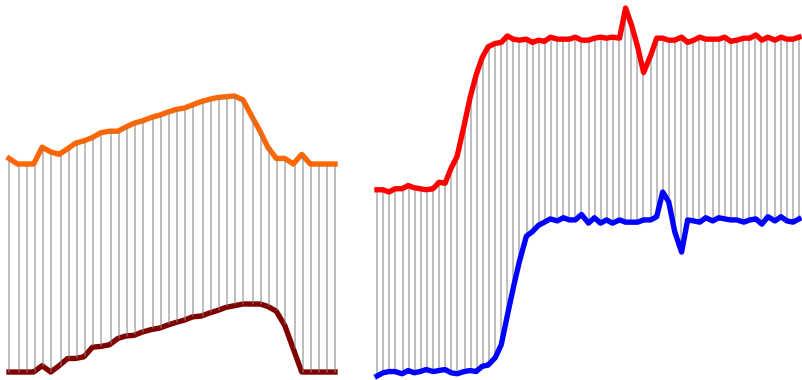


# Dynamic Time Warping

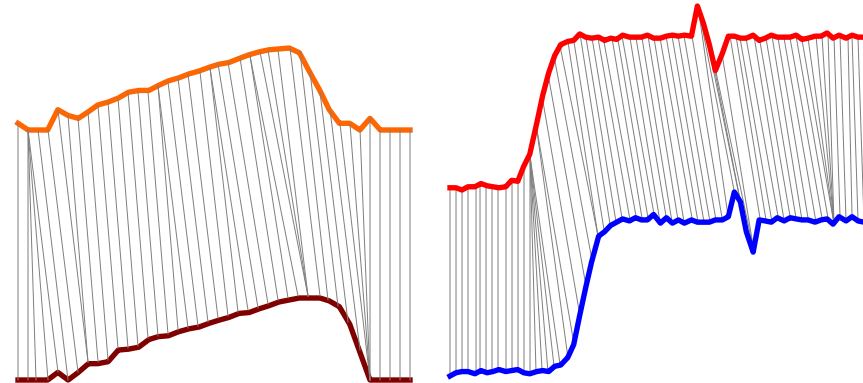
- Sometimes two time series that are conceptually equivalent evolve at different speeds, at least in some moments.



E.g. correspondence of peaks in two similar time series

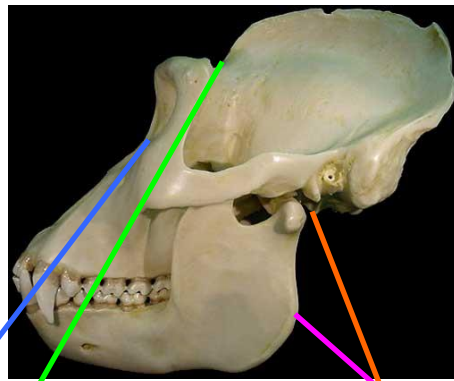


**Fixed Time Axis.** Sequences are aligned “one to one”. Greatly suffers from the misalignment in data. Euclidean.

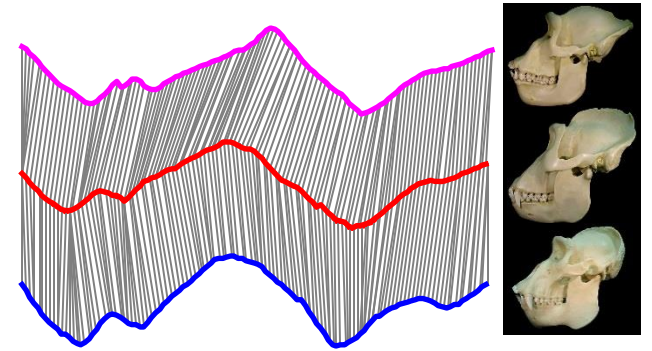
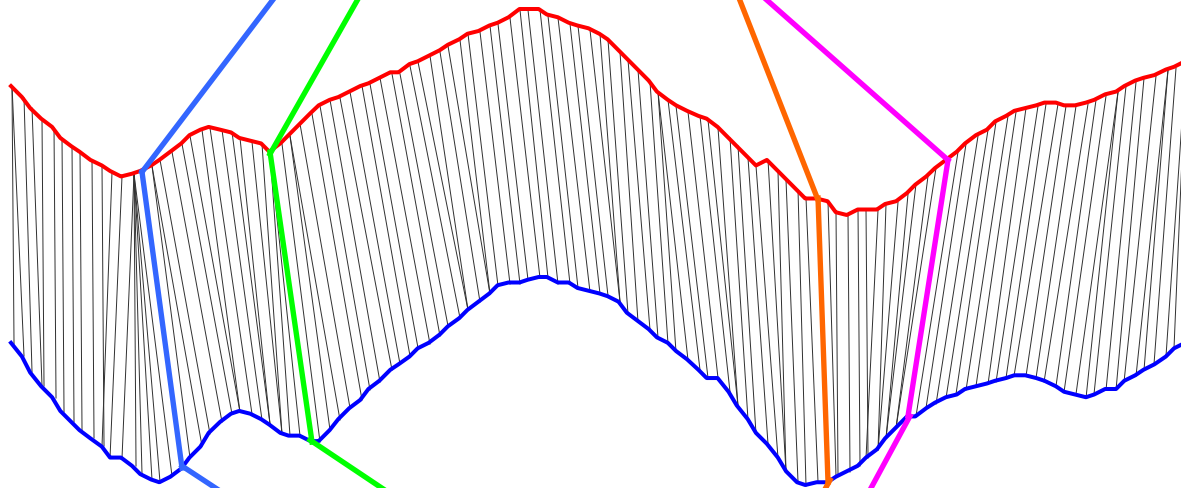
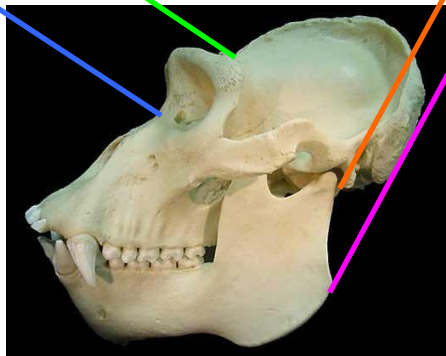


**Warped Time Axis.** Nonlinear alignments are possible. Can correct misalignments in data. Dynamic Time Warping.

Lowland Gorilla

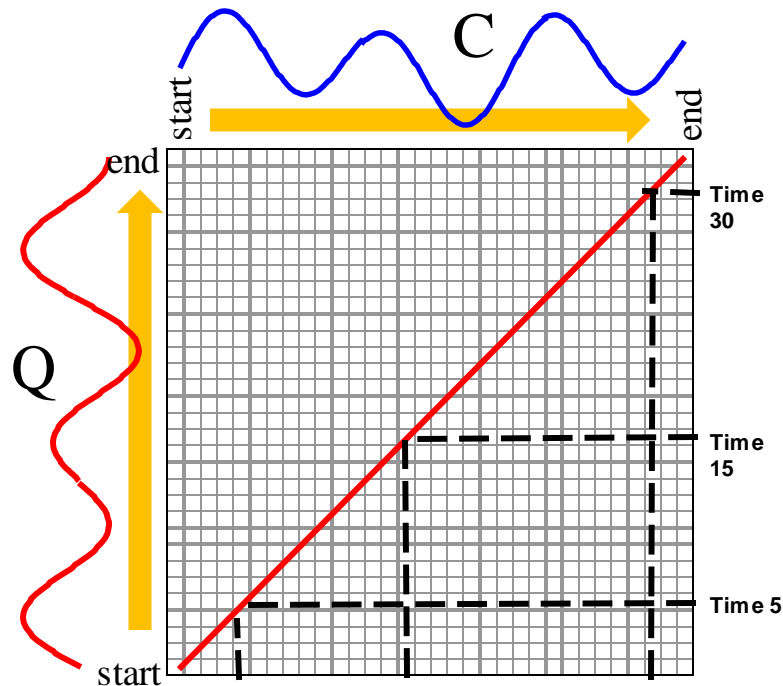
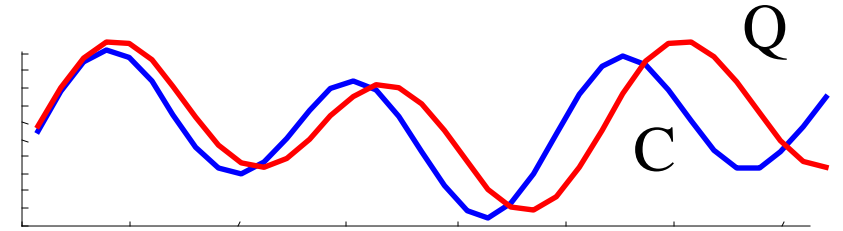


Mountain Gorilla



# How is DTW Calculated?

- We create a matrix the size of  $|Q|$  by  $|C|$ , then fill it in with the distance between every pair of point in our two time series.

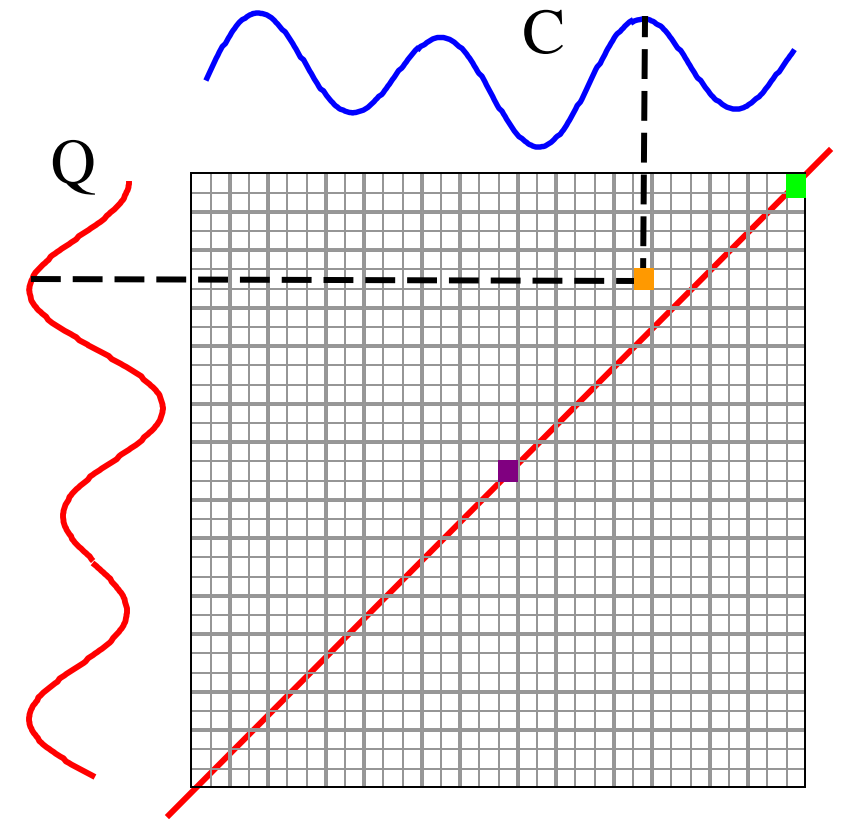
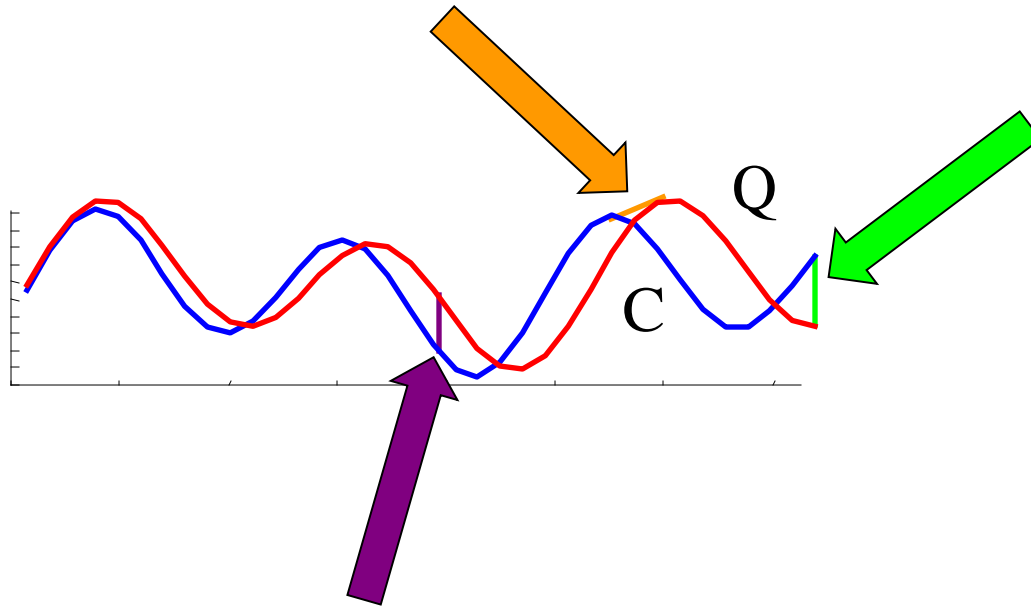


The Euclidean distance works only on the diagonal of the matrix. The sequence of comparisons performed:

- Start from pair of points  $(0,0)$
- After point  $(i,i)$  move to  $(i+1,i+1)$
- End the process on  $(n,n)$

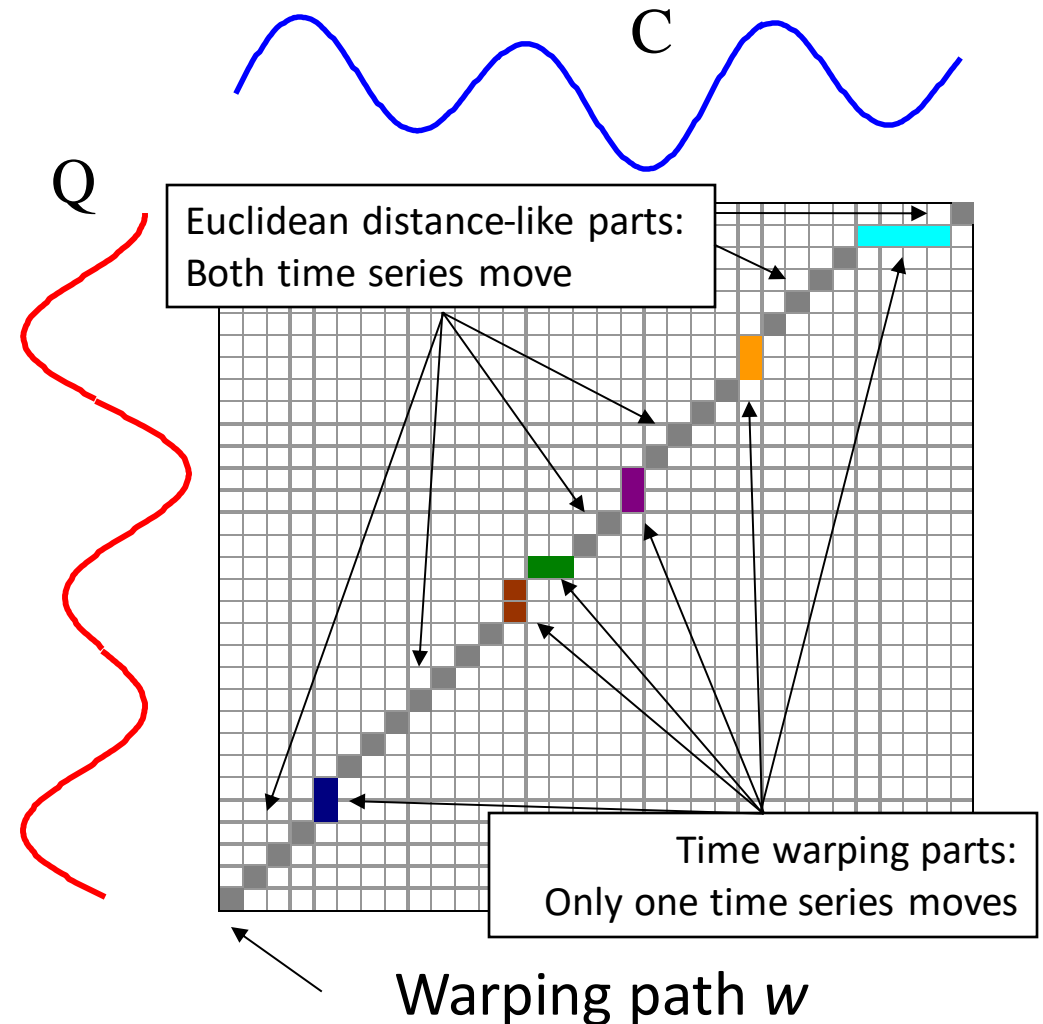
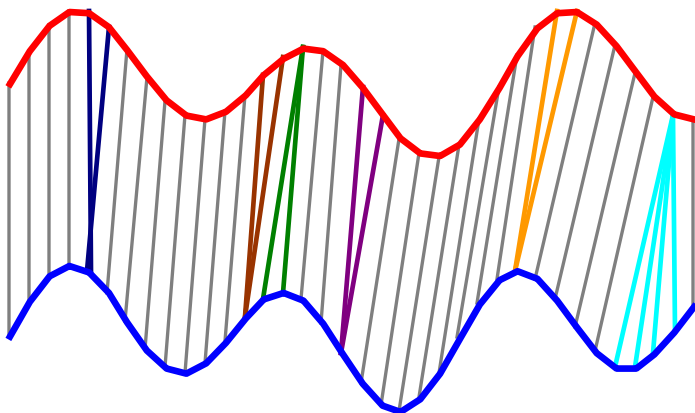
# How is DTW Calculated?

- The DTW distance can “freely” move outside the diagonal of the matrix
- Such cells correspond to temporally shifted points in the two time series



# How is DTW Calculated?

- Every possible warping between two time series, is a path through the matrix.
- The constrained sequence of comparisons performed:
  - Start from pair of points  $(0,0)$
  - After point  $(i,j)$ , either  $i$  or  $j$  increase by one, or both of them
  - End the process on  $(n,n)$



# How is DTW Calculated?

- Every possible warping between two time series, is a path through the matrix.
- We find the best one using a recursive definition of the DTW:

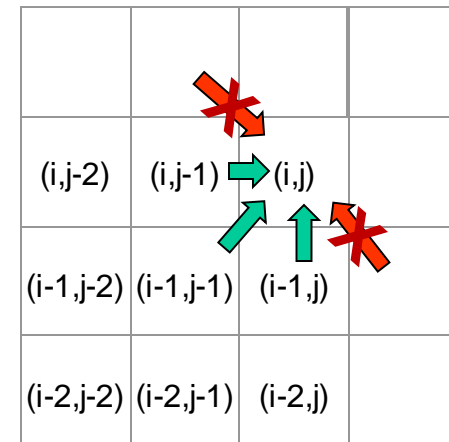
$$\begin{aligned}\gamma(i,j) &= \text{cost of best path reaching cell } (i,j) \\ &= d(q_i, c_j) + \min\{ \gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1) \}\end{aligned}$$

- Idea: best path must pass through  $(i-1, j)$ ,  $(i-1, j-1)$  or  $(i, j-1)$

$$DTW(Q, C) = \min \left\{ \sqrt{\sum_{k=1}^K w_k} / K \right\}$$

$w_k$  = cost of the k-th points comparison

- $w_k = |Q_i - C_j|$
- $w_k = (Q_i - C_j)^2$

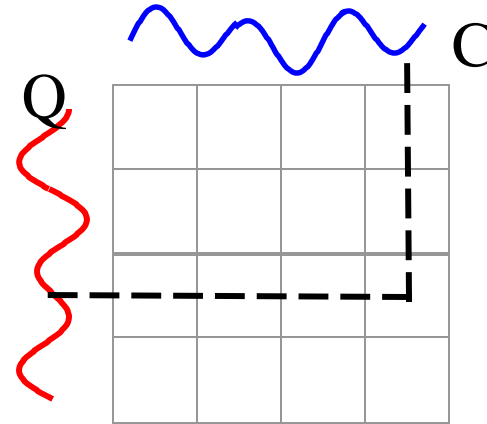




# Dynamic Programming Approach

Step 1: compute the matrix of all  $d(q_i, c_j)$

- Point-to-point distances  $D(i,j) = |Q_i - C_j|$



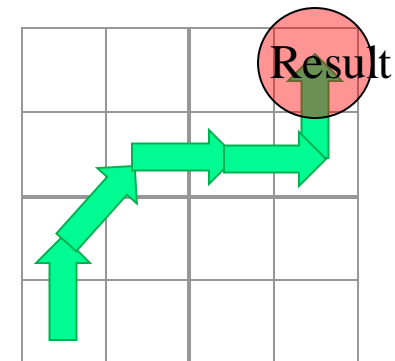
$$\gamma(i,j) = d(q_i, c_j) + \min\{ \gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1) \}$$

Step 2: compute the matrix of all path costs  $\gamma(i,j)$

- Start from cell  $(1,1)$
- Compute  $(2,1), (3,1), \dots, (n,1)$
- Repeat for columns  $2, 3, \dots, n$
- Final result in last cell computed



Step 3: find the path with the lowest value (best alignment)



# Dynamic Programming Approach

$$\gamma(i,j) = d(q_i, c_j) + \min\{ \gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1) \}$$

Step 2: compute the matrix of all path costs  $\gamma(i,j)$

- Start from cell  $(1,1)$

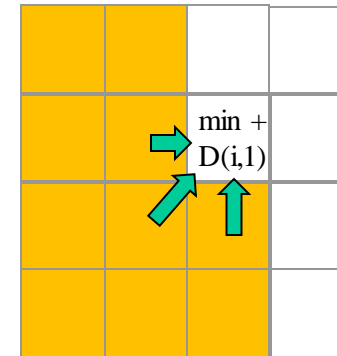
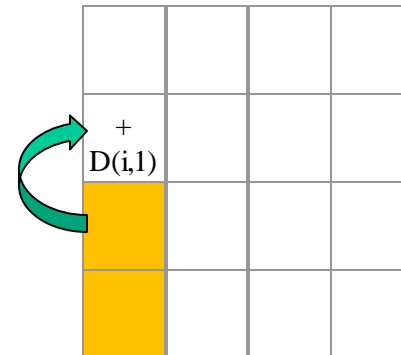
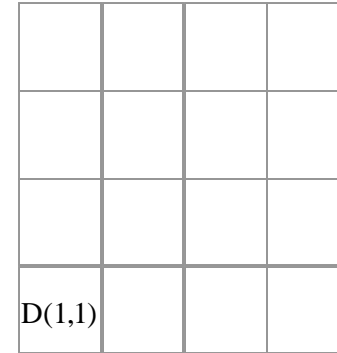
$$\begin{aligned}\gamma(1,1) &= d(q_1, c_1) + \min\{\gamma(0,0), \gamma(0,1), \gamma(1,0)\} \\ &= d(q_1, c_1) \\ &= D(1,1)\end{aligned}$$

- Compute  $(2,1), (3,1), \dots, (n,1)$

$$\begin{aligned} \gamma(i,1) &= d(q_i, c_1) + \min\{ \gamma(i-1,0), \gamma(i-1,1), \gamma(i,0) \} \\ &= d(q_i, c_1) + \gamma(i-1,1) \\ &= D(i,1) + \gamma(i-1,1) \end{aligned}$$

- Repeat for columns  $2, 3, \dots, n$

- The general formula applies

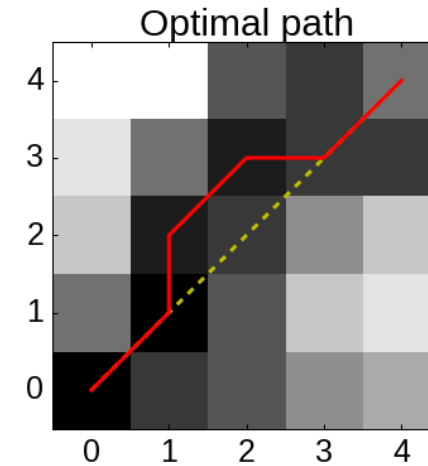
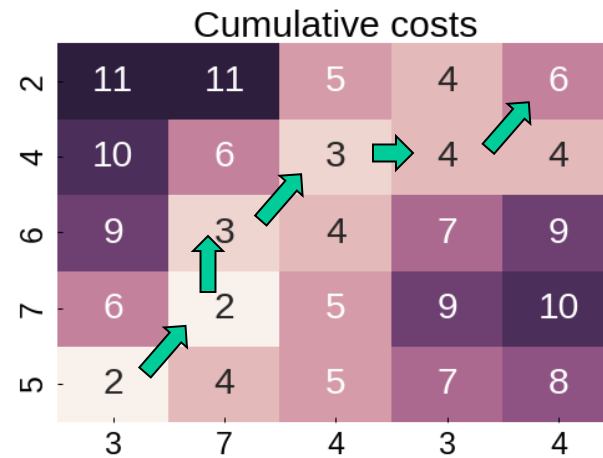
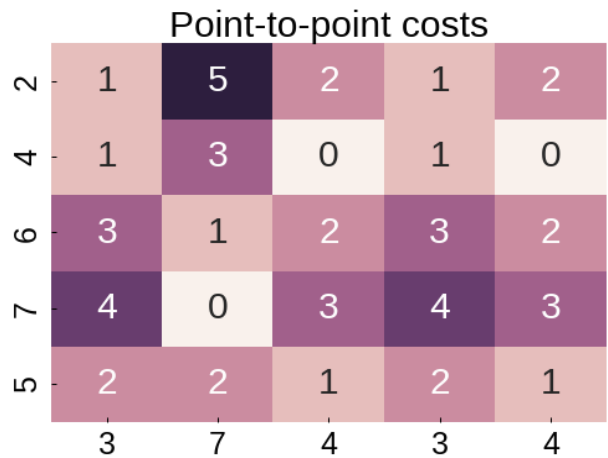


# Dynamic Programming Approach

## Example

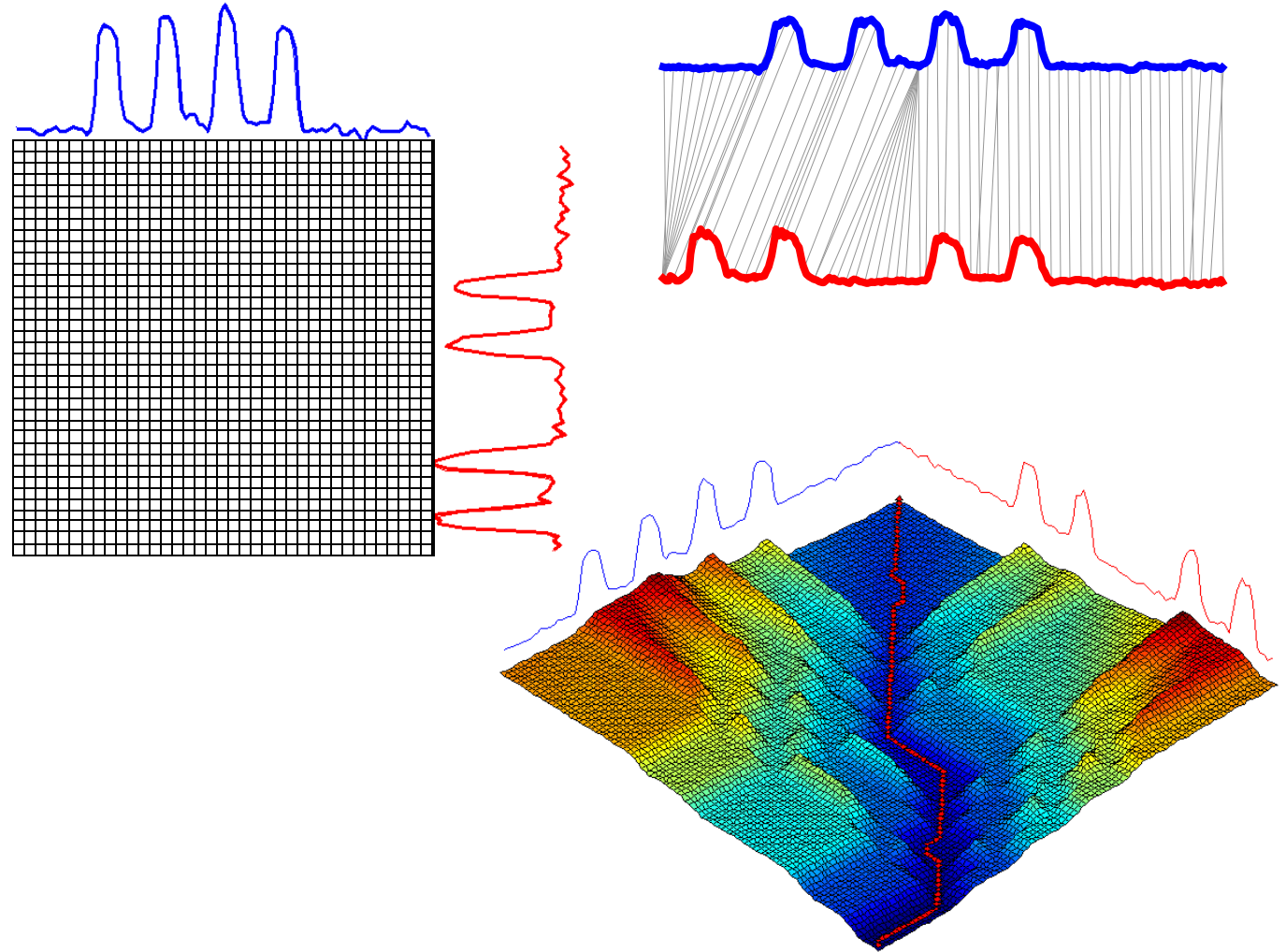
- $c = \langle 3, 7, 4, 3, 4 \rangle$
- $q = \langle 5, 7, 6, 4, 2 \rangle$

$$\gamma(i,j) = d(q_i, c_j) + \min\{ \gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1) \}$$



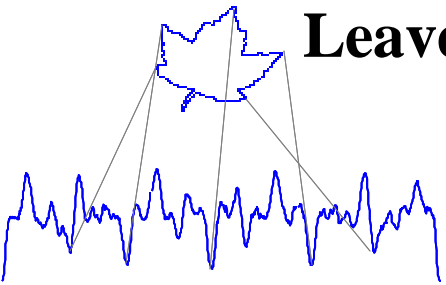
# Dynamic Time Warping – A Real Example

- A Real Example
- This example shows 2 one-week periods from the power demand time series.
- Note that although they both describe 4-day work weeks, the blue sequence had Monday as a holiday, and the red sequence had Wednesday as a holiday.

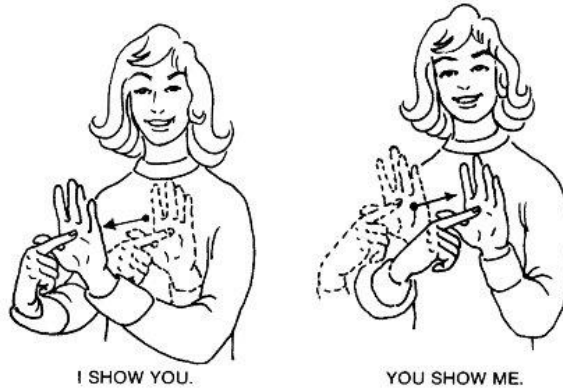


# Comparison of Euclidean Distance and DTW

**Leaves**



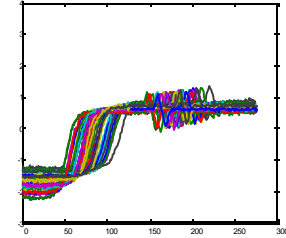
**Faces**



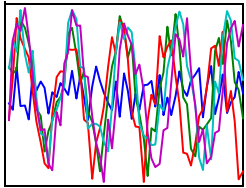
I SHOW YOU.

YOU SHOW ME.

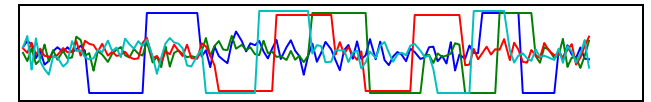
**Trace**



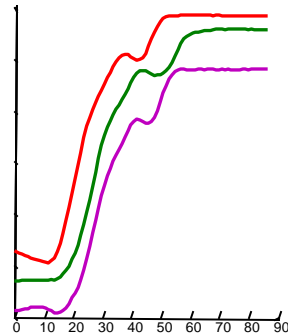
**Control**



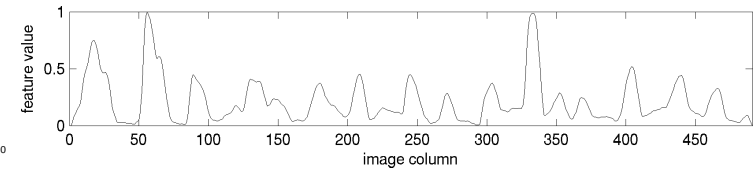
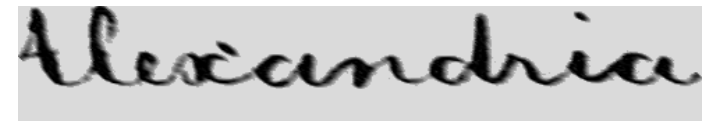
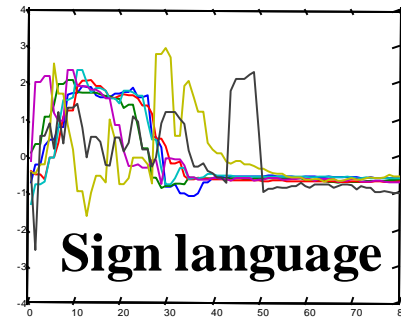
**2-Patterns**



**Gun**



**Sign language**



**Word Spotting**

# Comparison of Euclidean Distance and DTW

- Classification using 1-NN
- $\text{Class}(x)$  = class of most similar training object
- Leaving-one-out evaluation
- For each object: use it as test set, return overall average

Dataset	Error Rate	
	Euclidean	DTW
Word Spotting	4.78	1.10
Sign language	28.70	25.93
GUN	5.50	1.00
Nuclear Trace	11.00	0.00
Leaves <sup>#</sup>	33.26	4.07
(4) Faces	6.25	2.68
Control Chart <sup>*</sup>	7.5	0.33
2-Patterns	1.04	0.00

# Comparison of Euclidean Distance and DTW

- Classification using 1-NN
- $\text{Class}(x)$  = class of most similar training object
- Leaving-one-out evaluation
- For each object: use it as test set, return overall average
- DTW is two to three orders of magnitude slower than Euclidean distance.

Milliseconds		
Dataset	Euclidean	DTW
Word Spotting	40	8,600
Sign language	10	1,110
GUN	60	11,820
Nuclear Trace	210	144,470
Leaves	150	51,830
(4) Faces	50	45,080
Control Chart	110	21,900
2-Patterns	16,890	545,123

# What we have seen so far...

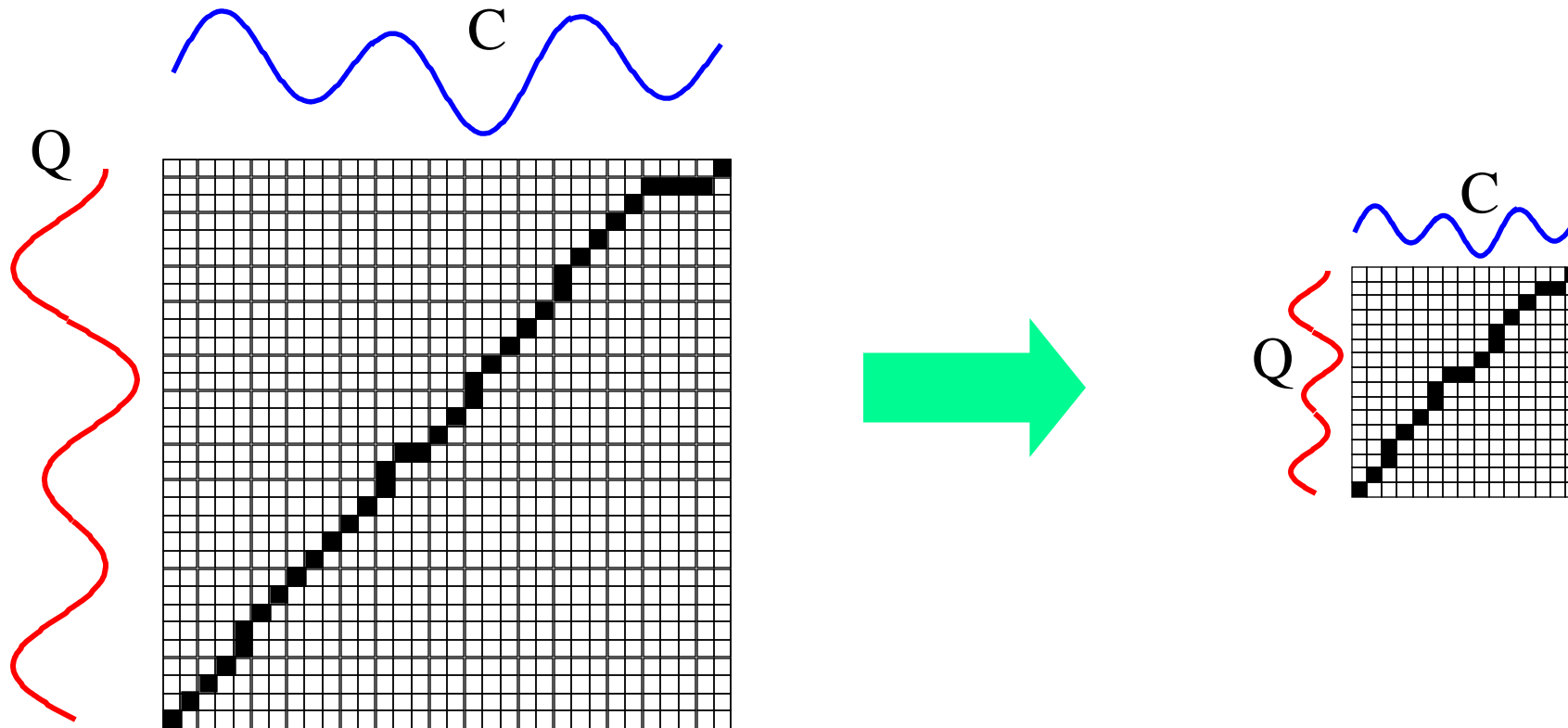
---

- Dynamic Time Warping gives much better results than Euclidean distance on many problems.
- Dynamic Time Warping is very very slow to calculate!
- Is there anything we can do to speed up similarity search under DTW?



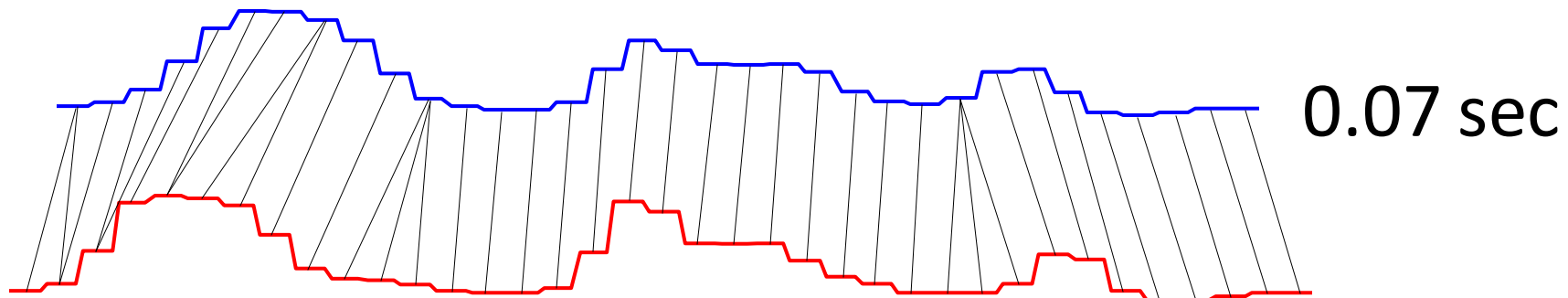
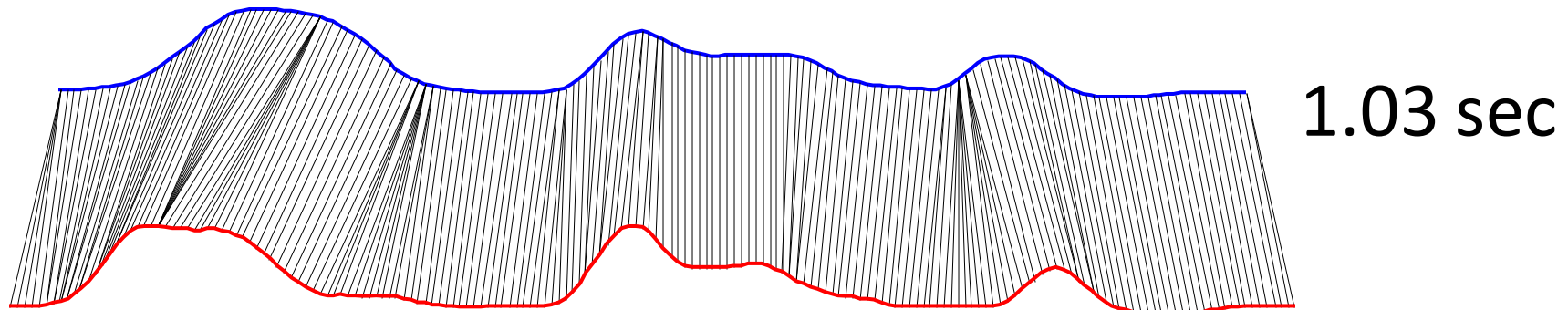
# Fast Approximations to DTW

- Approximate the time series with some compressed or downsampled representation, and do DTW on the new representation.



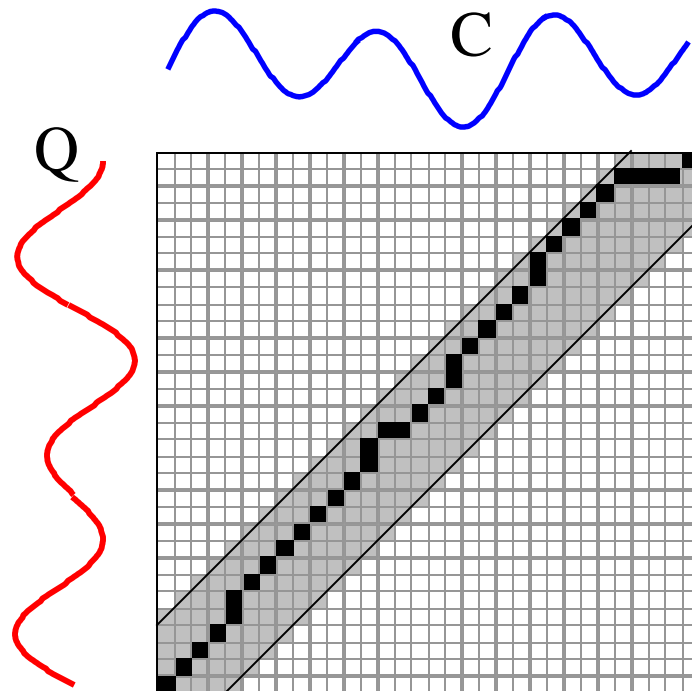
# Fast Approximations to DTW

- There is strong visual evidence to suggest it works well
- In the literature there is good experimental evidence for the utility of the approach on clustering, classification, etc.

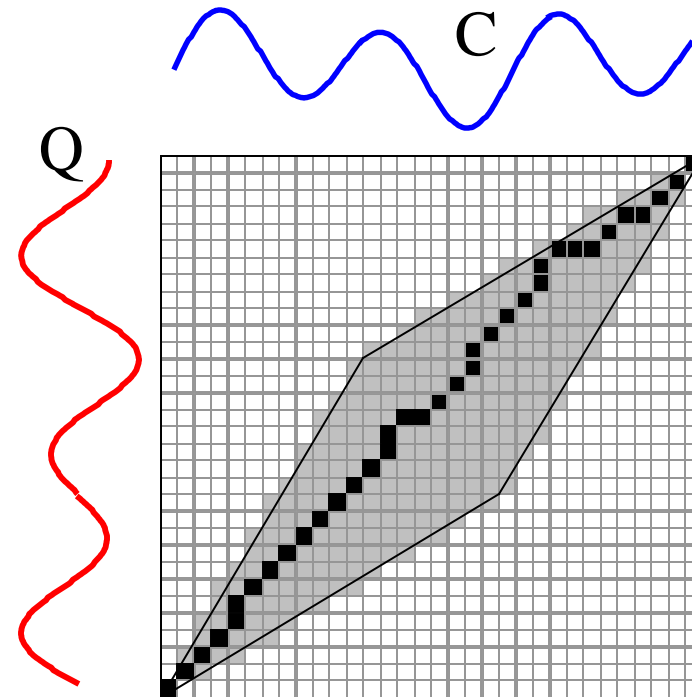


# Global Constraints

- Slightly speed up the calculations
- Prevent pathological warpings



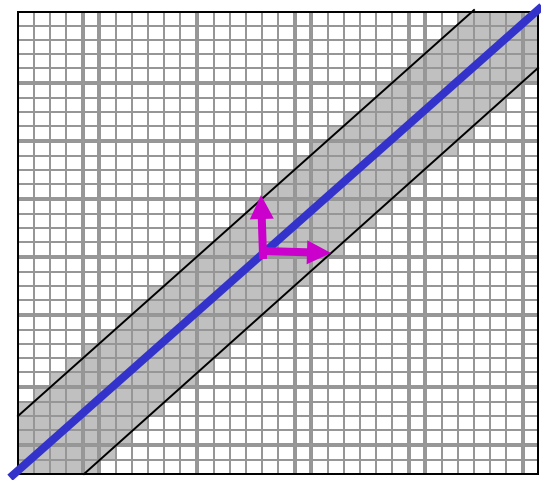
Sakoe-Chiba Band



Itakura Parallelogram

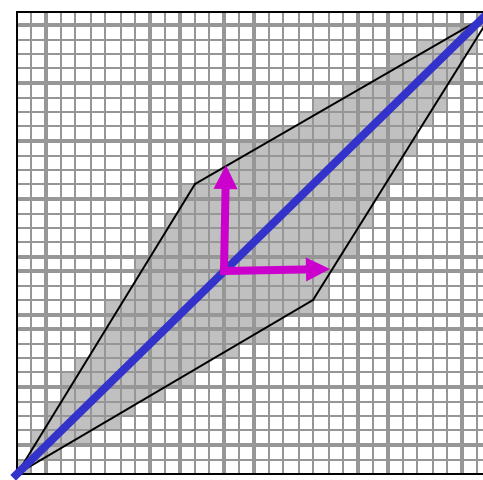
# Global Constraints

- A global constraint constrains the indices of the warping path  $w_k = (i, j)_k$  such that  $j-r \leq i \leq j+r$ , where  $r$  is a term defining allowed range of warping for a given point in a sequence.
- $r$  can be considered as a *window* that reduces the number of calculus.



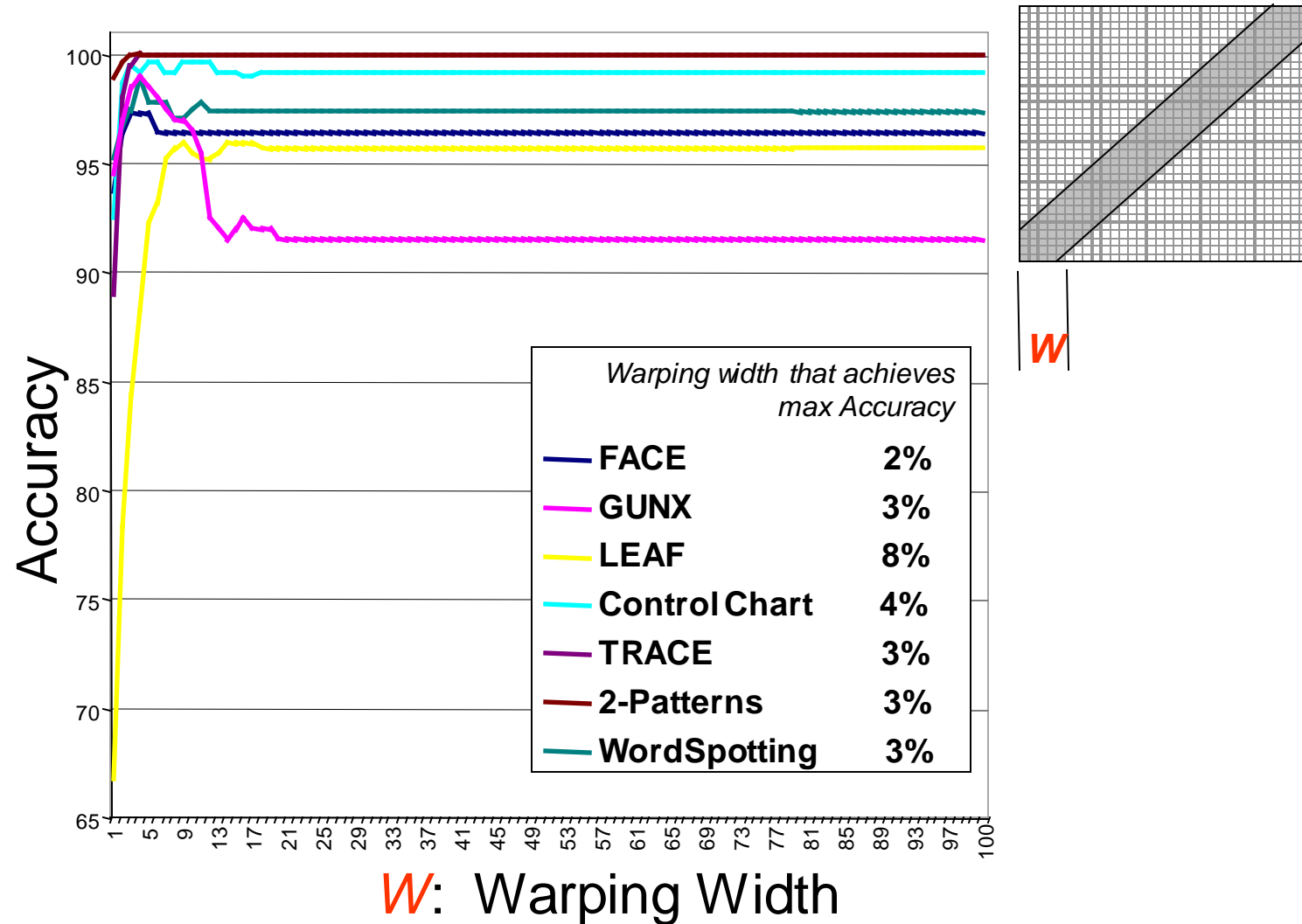
Sakoe-Chiba Band

$r_i$



Itakura Parallelogram

# Accuracy vs. Width of Warping Window



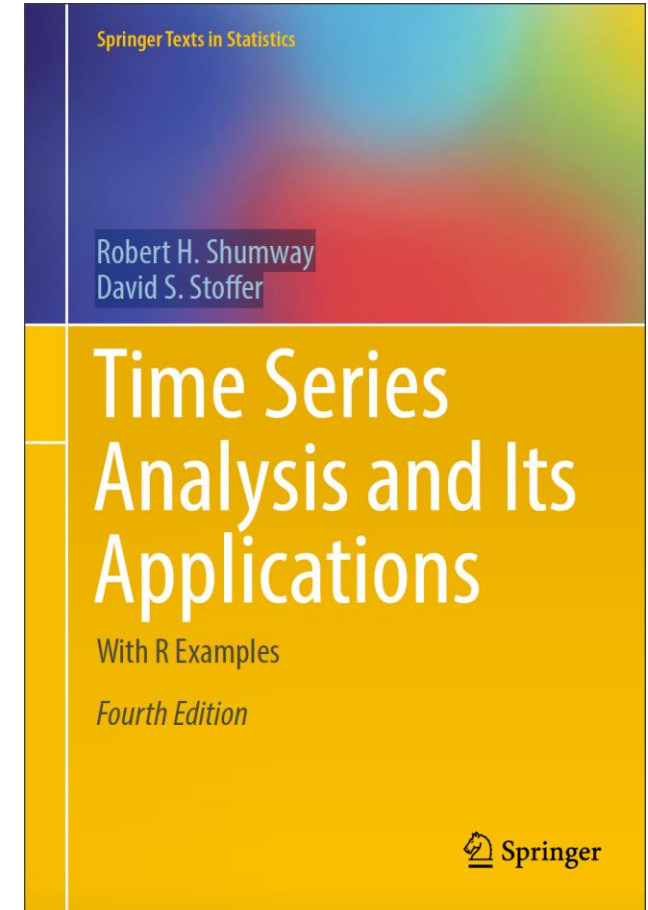
# Summary of Time Series Similarity

---

- If you have short time series
  - use DTW after searching over the warping window size
- If you have long time series
  - if you do know something about your data => extract features
  - and you know nothing about your data => try compression/approximation-based dissimilarity

# References

- Forecasting: Principles and Practice. Rob J Hyndman and George Athanasaopoulos. (<https://otexts.com/fpp2/>)
- Time Series Analysis and Its Applications. Robert H. Shumway and David S. Stoffer. 4<sup>th</sup> edition. (<https://www.stat.pitt.edu/stoffer/tsa4/tsa4.pdf>)
- Mining Time Series Data. Chotirat Ann Ratanamahatana et al. 2010. ([https://www.researchgate.net/publication/227001229\\_Mining\\_Time\\_Series\\_Data](https://www.researchgate.net/publication/227001229_Mining_Time_Series_Data))
- Dynamic Programming Algorithm Optimization for Spoken Word Recognition. Hiroaki Sakode et al. 1978.
- Experiencing SAX: a Novel Symbolic Representation of Time Series. Jessica Line et al. 2009
- Compression-based data mining of sequential data. Eamonn Keogh et al. 2007.



# Exercises Dynamic Time Warping

---



# DTW – Exercise 1

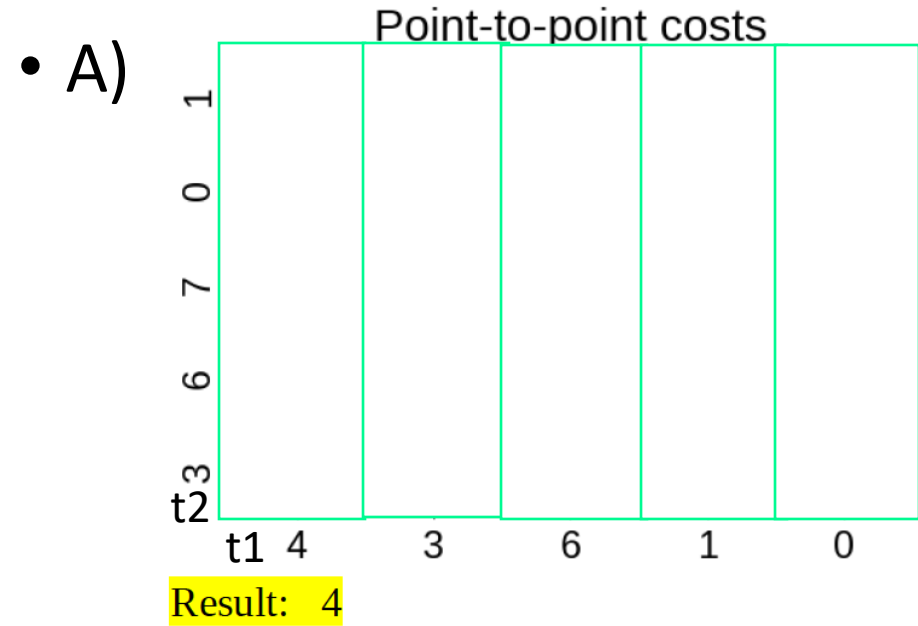
- Given the following input time series:

<b>t1</b>	< 4, 3, 6, 1, 0 >
<b>t2</b>	< 3, 6, 7, 0, 1 >

- A) Compute the distance between “t1” and “t2”, using the DTW with distance between points computed as  $d(x,y) = |x - y|$ .
- B) If we repeat the computation of point (A) above, this time with a Sakoe-Chiba band of size  $r=1$ , does the result change? Why?
- C) If we compute  $DTW(T1,T2)$ , where  $T1$  is equal to  $t1$  in reverse order (namely  $T1=<0,1,6,3,4>$ ) and similarly for  $T2$  (namely  $T2=<1,0,7,6,3>$ ), is it true that  $DTW(T1,T2) = DTW(t1,t2)$ ? Discuss the problem without providing any computation.

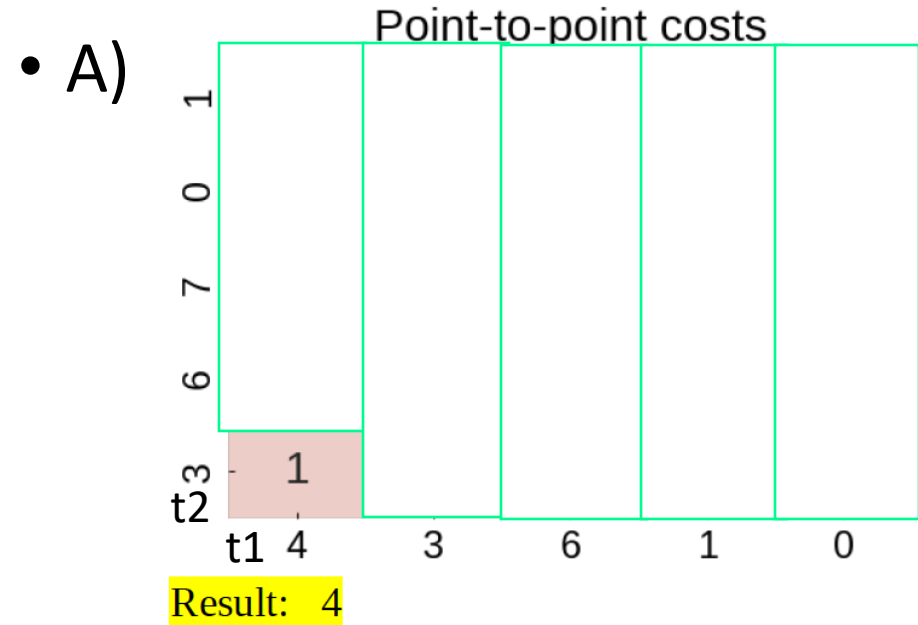
# DTW – Exercise 1 - Solution

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >



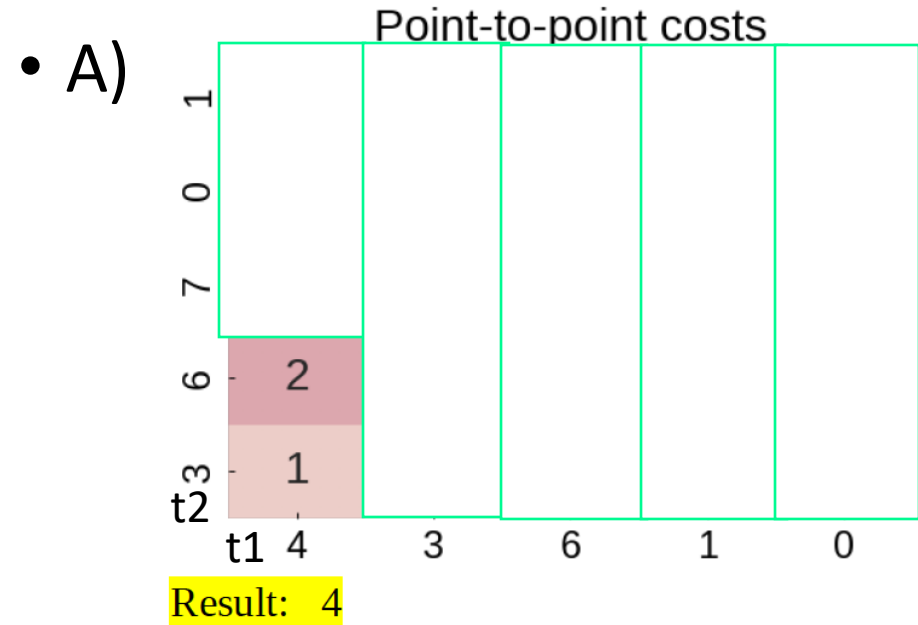
# DTW – Exercise 1 - Solution

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >



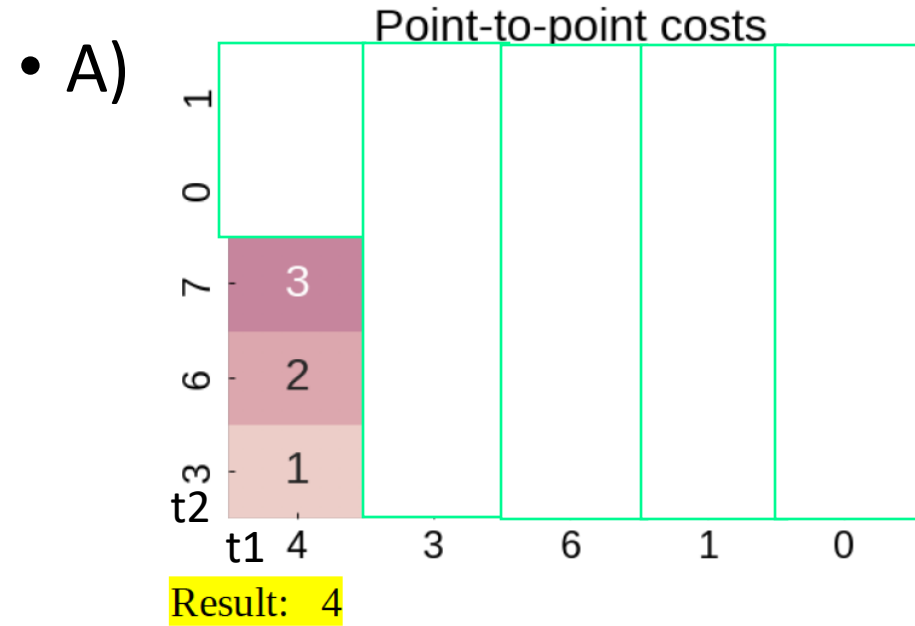
# DTW – Exercise 1 - Solution

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >



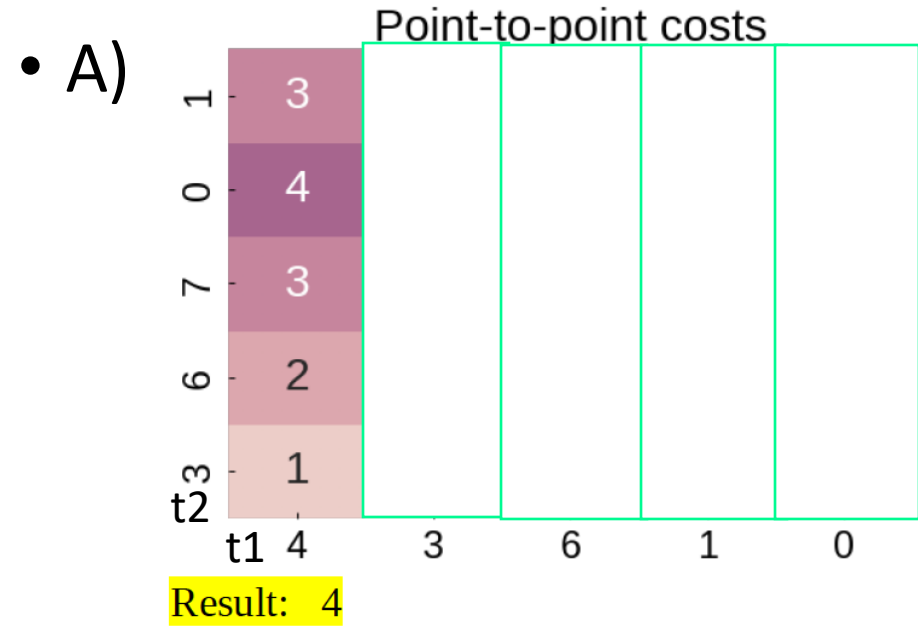
# DTW – Exercise 1 - Solution

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >



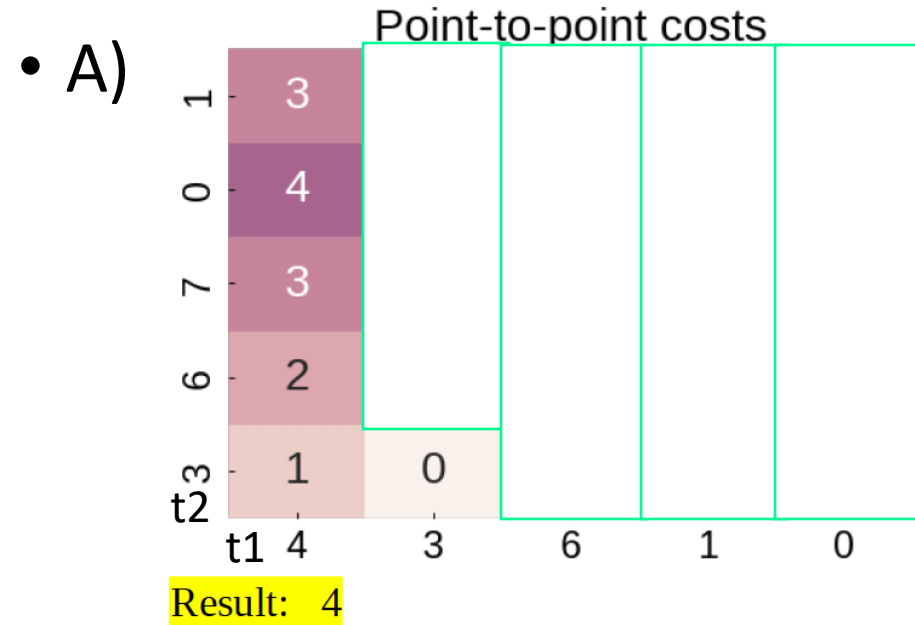
# DTW – Exercise 1 - Solution

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >



# DTW – Exercise 1 - Solution

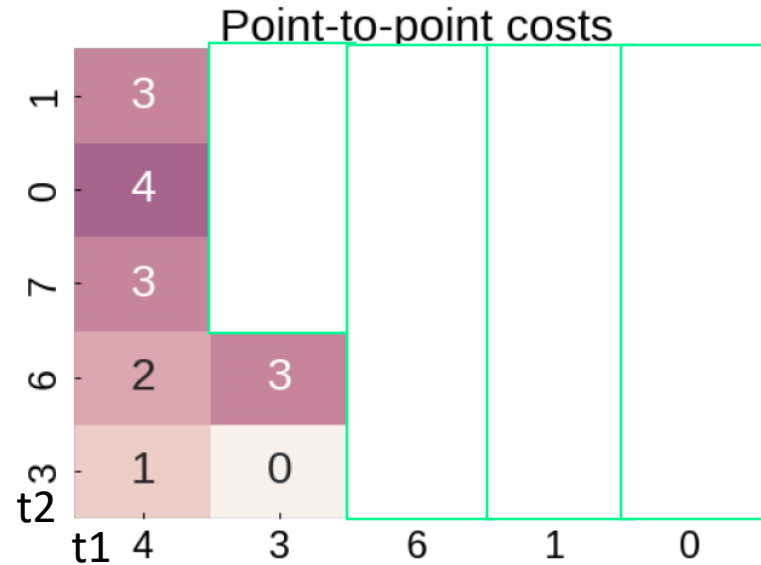
t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >



# DTW – Exercise 1 - Solution

<b>t1</b>	< 4, 3, 6, 1, 0 >
<b>t2</b>	< 3, 6, 7, 0, 1 >

- A)

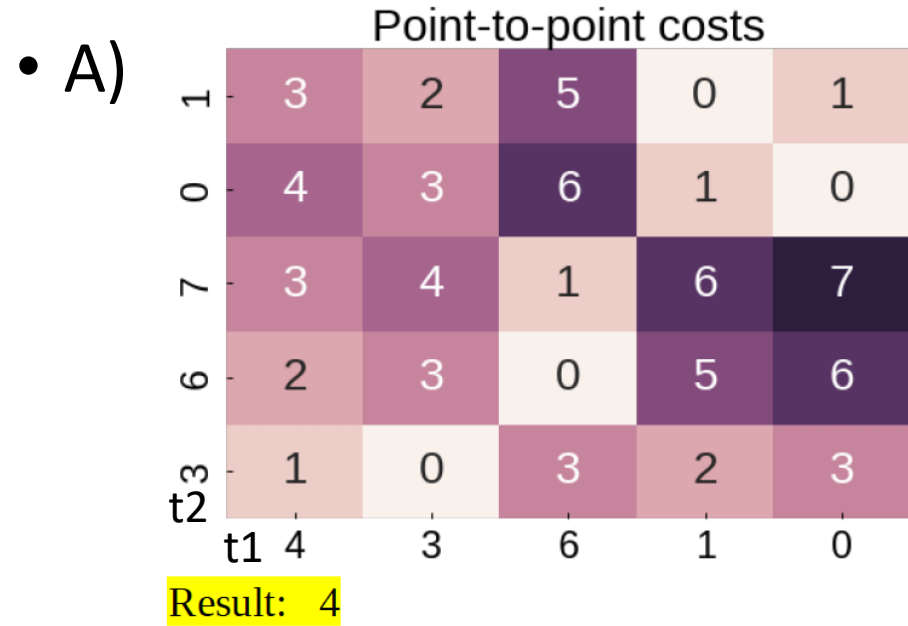


Result: 4



# DTW – Exercise 1 - Solution

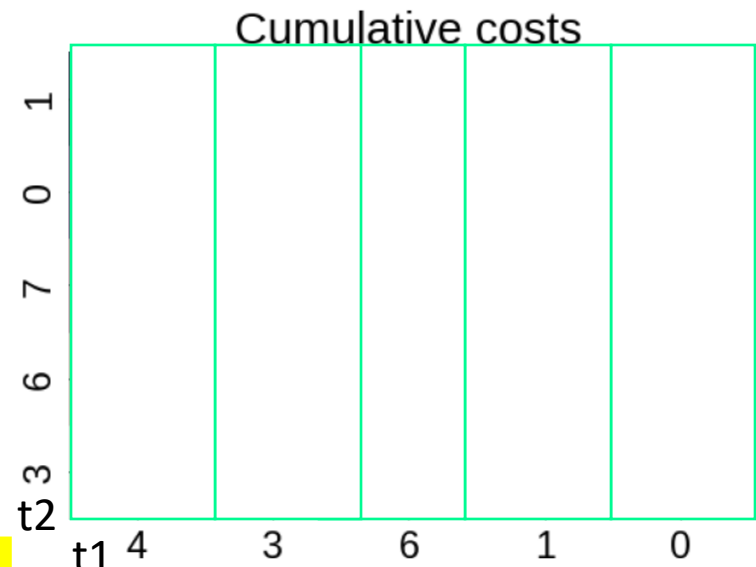
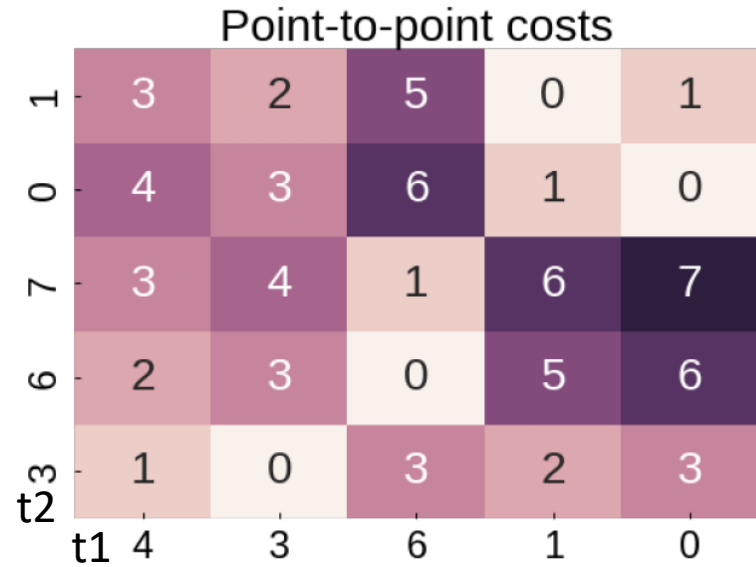
t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >



# DTW – Exercise 1 - Solution

<b>t1</b>	< 4, 3, 6, 1, 0 >
<b>t2</b>	< 3, 6, 7, 0, 1 >

• A)

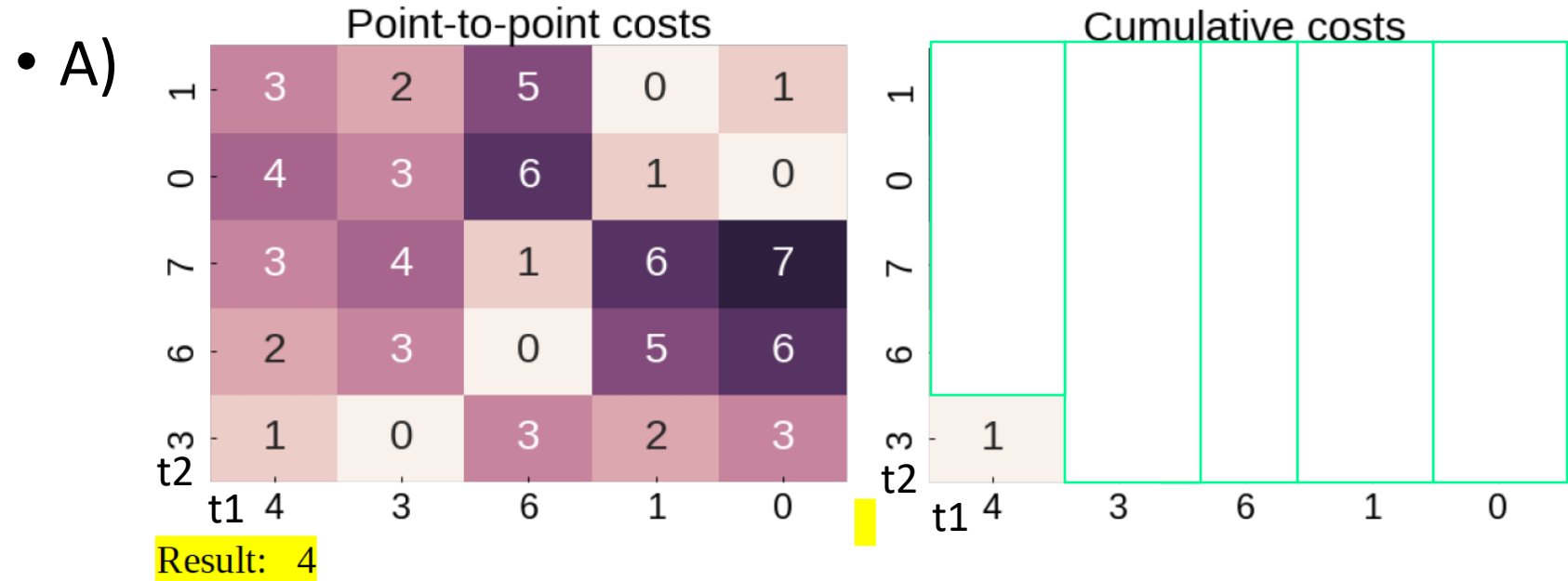


Result: 4

$$\gamma(i,j) = d(q_i, c_j) + \min \{ \gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1) \}$$

# DTW – Exercise 1 - Solution

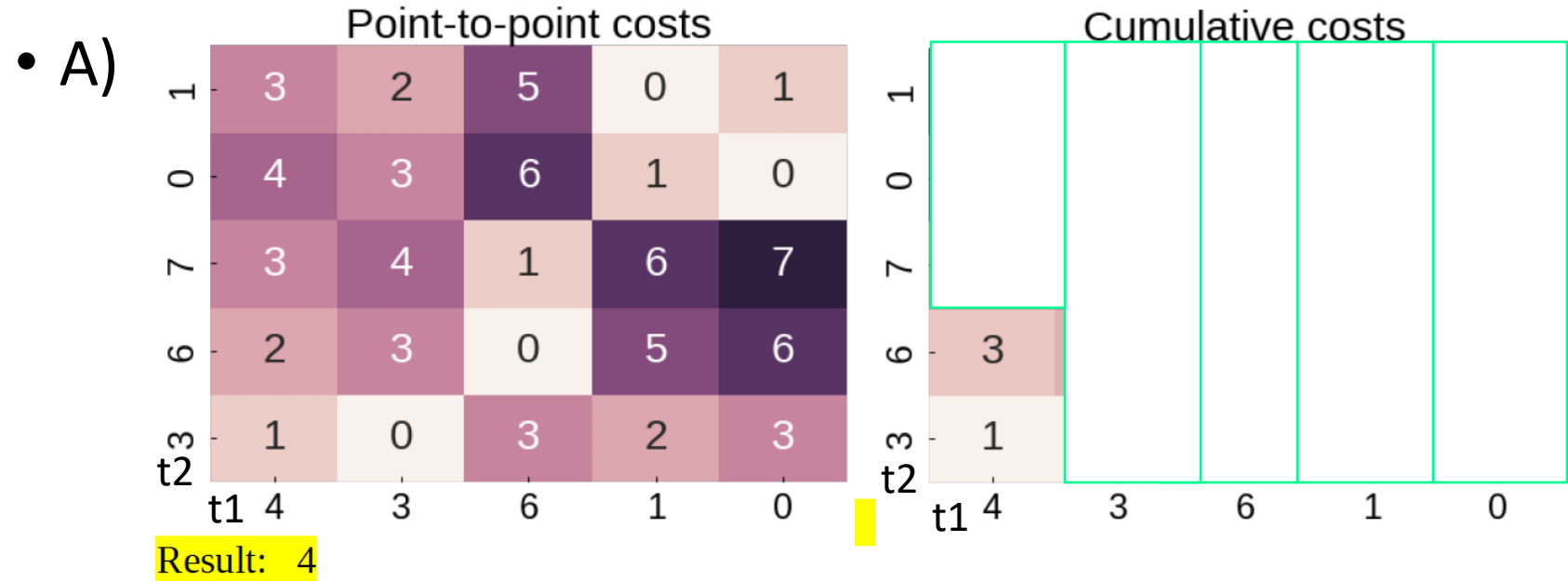
<b>t1</b>	< 4, 3, 6, 1, 0 >
<b>t2</b>	< 3, 6, 7, 0, 1 >



$$\gamma(i,j) = d(q_i, c_j) + \min\{ \gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1) \}$$

# DTW – Exercise 1 - Solution

<b>t1</b>	< 4, 3, 6, 1, 0 >
<b>t2</b>	< 3, 6, 7, 0, 1 >



$$\gamma(i,j) = d(q_i, c_j) + \min\{ \gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1) \}$$

# DTW – Exercise 1 - Solution

<b>t1</b>	< 4, 3, 6, 1, 0 >
<b>t2</b>	< 3, 6, 7, 0, 1 >

• A)



Result: 4

$$\gamma(i,j) = d(q_i, c_j) + \min\{ \gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1) \}$$

# DTW – Exercise 1 - Solution

<b>t1</b>	< 4, 3, 6, 1, 0 >
<b>t2</b>	< 3, 6, 7, 0, 1 >

• A)



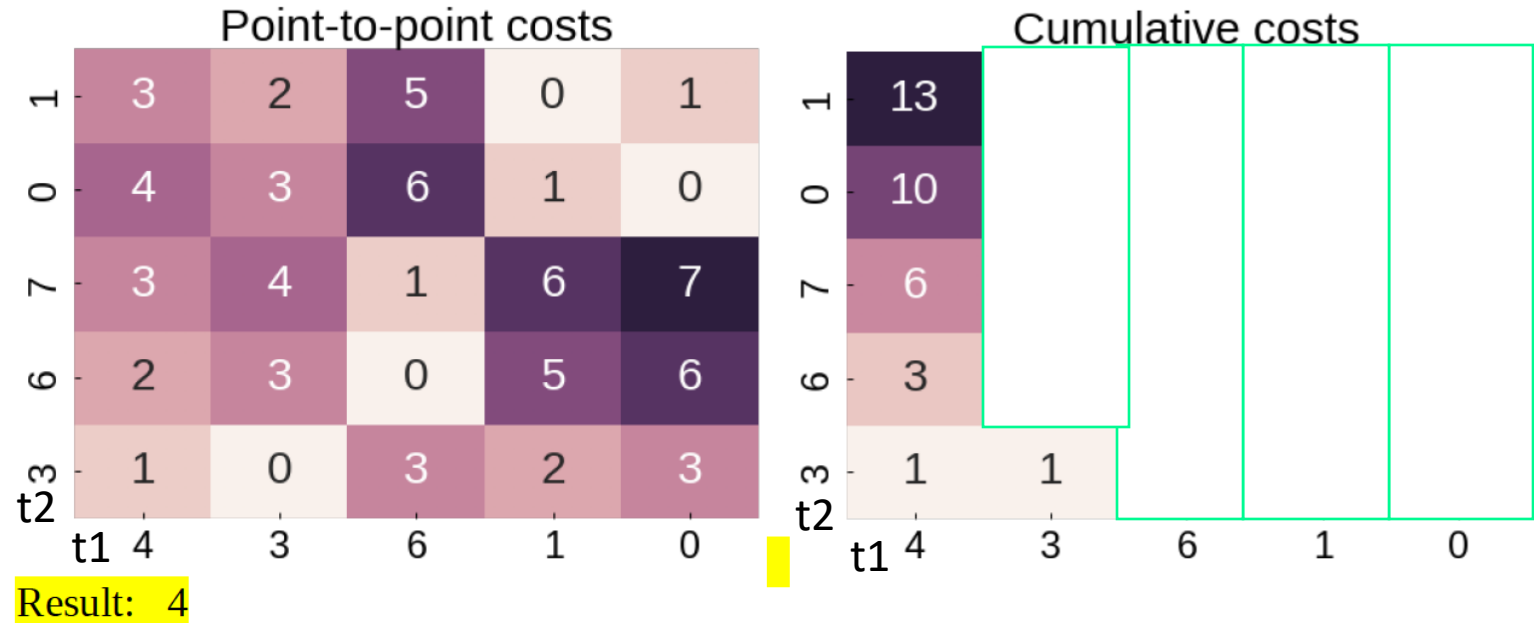
Result: 4

$$\gamma(i,j) = d(q_i, c_j) + \min\{ \gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1) \}$$

# DTW – Exercise 1 - Solution

<b>t1</b>	< 4, 3, 6, 1, 0 >
<b>t2</b>	< 3, 6, 7, 0, 1 >

• A)

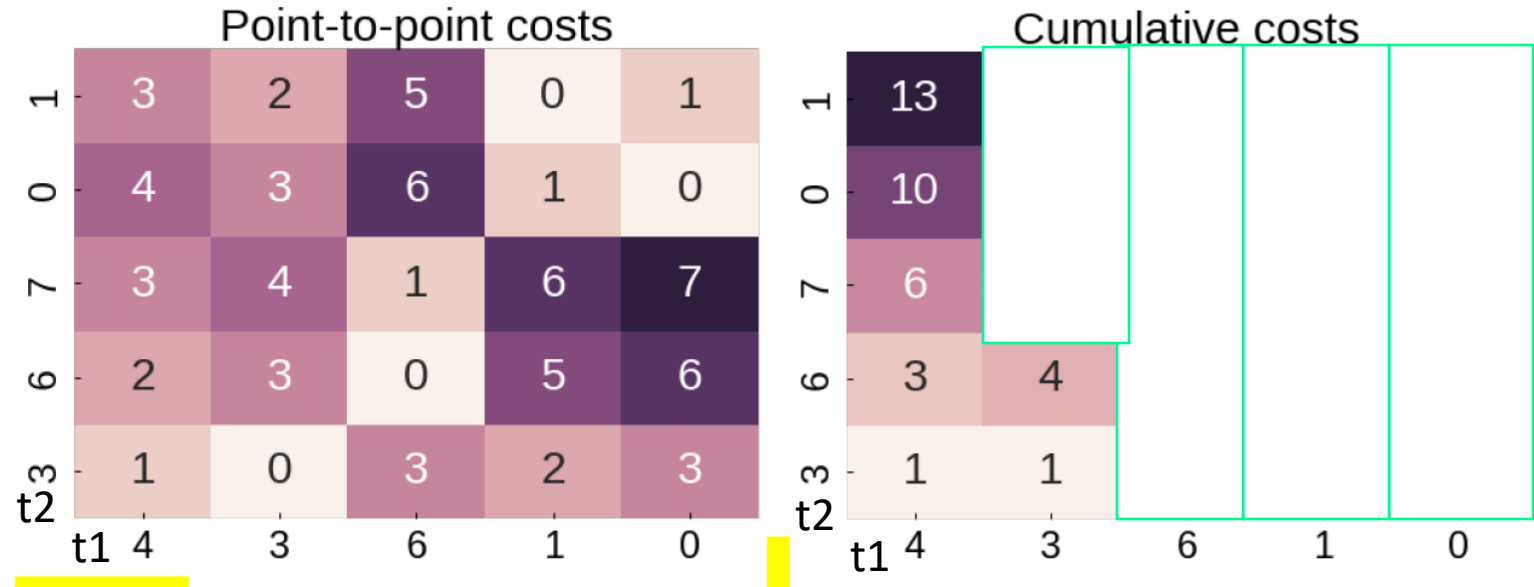


$$\gamma(i,j) = d(q_i, c_j) + \min\{ \gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1) \}$$

# DTW – Exercise 1 - Solution

<b>t1</b>	< 4, 3, 6, 1, 0 >
<b>t2</b>	< 3, 6, 7, 0, 1 >

• A)



Result: 4

$$\gamma(i,j) = d(q_i, c_j) + \min\{ \gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1) \}$$



# DTW – Exercise 1 - Solution

<b>t1</b>	< 4, 3, 6, 1, 0 >
<b>t2</b>	< 3, 6, 7, 0, 1 >

• A)



Result: 4

$$\gamma(i,j) = d(q_i, c_j) + \min\{ \gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1) \}$$

# DTW – Exercise 1 - Solution

<b>t1</b>	< 4, 3, 6, 1, 0 >
<b>t2</b>	< 3, 6, 7, 0, 1 >

• A)



Result: 4

$$\gamma(i,j) = d(q_i, c_j) + \min\{ \gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1) \}$$

# DTW – Exercise 1 - Solution

<b>t1</b>	< 4, 3, 6, 1, 0 >
<b>t2</b>	< 3, 6, 7, 0, 1 >

• A)



Result: 4

$$\gamma(i,j) = d(q_i, c_j) + \min\{ \gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1) \}$$

# DTW – Exercise 1 - Solution

<b>t1</b>	< 4, 3, 6, 1, 0 >
<b>t2</b>	< 3, 6, 7, 0, 1 >

• A)

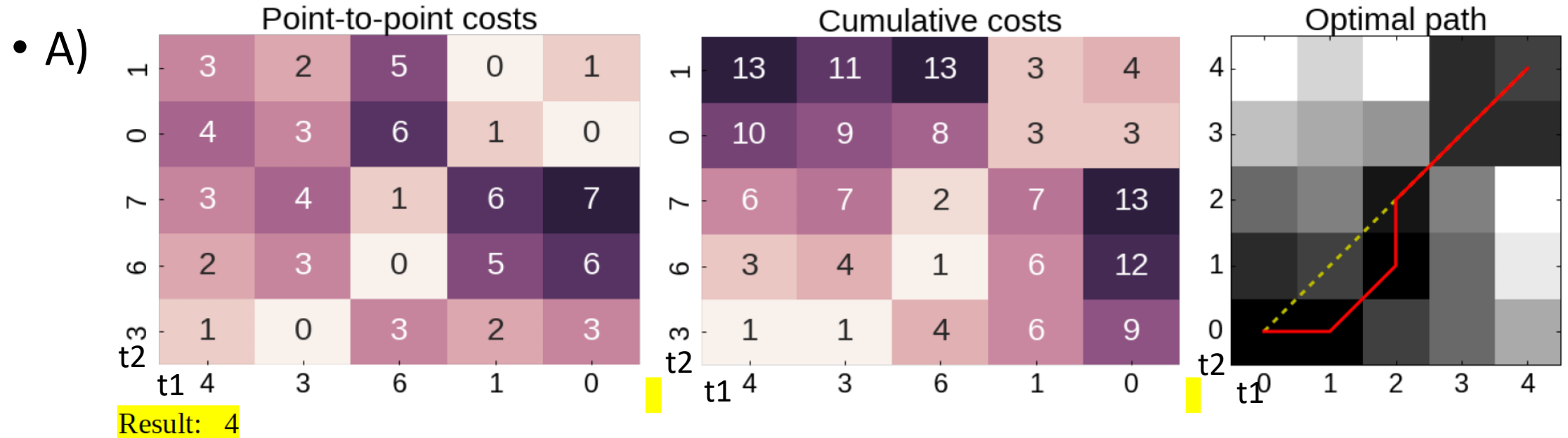


Result: 4

$$\gamma(i,j) = d(q_i, c_j) + \min \{ \gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1) \}$$

# DTW – Exercise 1 - Solution

t1	< 4, 3, 6, 1, 0 >
t2	< 3, 6, 7, 0, 1 >



- B) No. Because the DTW optimal path remains inside the band of size  $r=1$
- C) Yes. The optimal path in one direction is the same in the opposite direction. Though, the cumulative costs matrix might look different.

# DTW – Exercise 2

---

- Given the following time series:

t	=	< 2, 6, 9, 1, 6, 2 >
q	=	< 5, 1, 5, 5, 8, 4 >

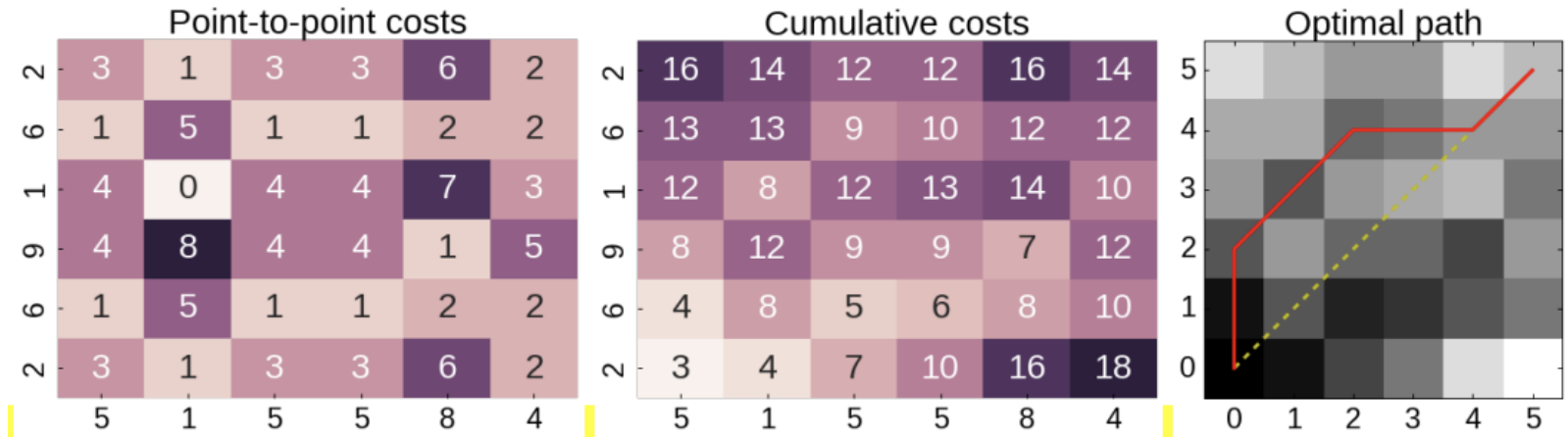
compute

- (i) their Manhattan and Euclidean distance,
- (ii) their DTW, and (iii) their DTW with Sakoe-Chiba band of size  $r=1$  (i.e. all cells at distance  $\leq 1$  from the diagonal are allowed).
- For points (ii) and (iii) show the cost matrix and the optimal path found.

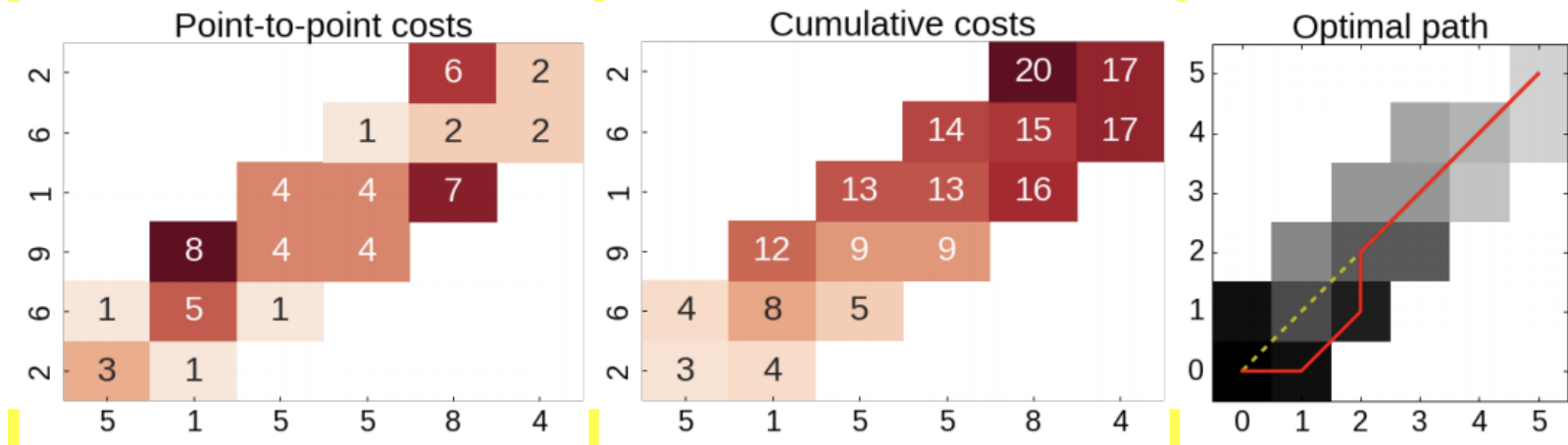
# DTW – Exercise 2 - Solution

- Euclidean =  $\sqrt{74} = 8.6$ , Manhattan = 20

- DTW = 14



- DTW  $r=1$  = 17



# DTW – Exercise 3

- Given the following time series:

ID	Time series
W	< 6, 11, 13, 15 >
X	< 10, 7, 7, 12, 14, 17 >
Y	< 9, 11, 14, 13, 20 >

- Compute the distances among all pairs of time series adopting a Dynamic Time Warping distance, and computing the distances between single points as  $d(x,y) = |x - y|$ . For each pair of time series compared also show the matrix used to compute the final result.



# DTW – Exercise 3 - Solution

ID	Time series
W	< 6, 11, 13, 15 >
X	< 10, 7, 7, 12, 14, 17 >
Y	< 9, 11, 14, 13, 20 >

W – X

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
[1,]	(4) 4	(1) 5	(1) 6	(6) 12	(8) 20	(11) 31
[2,]	(1) 5	(4) 8	(4) 9	(1) 7	(3) 10	(6) 16
[3,]	(3) 8	(5) 11	(5) 14	(1) 8	(1) 8	(4) 12
[4,]	(5) 13	(8) 16	(8) 19	(3) 11	(4) 9	(2) 10

W – Y

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	(3) 3	(5) 8	(8) 16	(7) 23	(14) 37
[2,]	(2) 5	(0) 3	(3) 6	(2) 8	(9) 17
[3,]	(5) 9	(2) 5	(1) 4	(0) 4	(7) 11
[4,]	(6) 15	(4) 9	(1) 5	(2) 6	(5) 9

X – Y

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	(1) 1	(1) 2	(4) 6	(3) 9	(10) 19
[2,]	(2) 3	(4) 5	(7) 9	(6) 12	(13) 22
[3,]	(2) 5	(4) 7	(7) 12	(6) 15	(13) 25
[4,]	(3) 8	(1) 6	(2) 8	(1) 9	(8) 17
[5,]	(5) 13	(3) 9	(0) 6	(1) 7	(6) 13
[6,]	(8) 21	(6) 15	(3) 9	(4) 10	(3) 10