Dynamic Time Warping (DTW)

⇒ **Goal**: Apply k-means to Time Series Data

- Clustering is unsupervised learning
- Similarity between data points is measured with a distance metric

Clustering different time series into similar groups is a challenging clustering task because each data point is an ordered sequence.

Why Euclidean distance metric is unsuitable:

- it is invariant to time shifts

If two time series are highly correlated, but one is shifted by even one time step, Euclidean distance would erroneously measure them as further apart.

DTW - Dynamic Time Warping

DTW is a technique to measure similarity between two temporal sequences that do not align exactly in time, speed, or length.

$X = (x_0, \cdots, x_n)$

$Y = (y_0, \cdots, y_m)$

$$DTW(x, y) = \min_{\pi} \sqrt{\sum_{(i,j)\in\pi} d(x_i, y_j)^2}$$

where $\pi = [\pi_0, \dots, \pi_K]$ is a path that satisfies the following properties:

- it is a list of index pairs $\pi_k = (i_k, j_k)$ with $0 \le i_k < n$ and $0 \le j_k < m$
- $\pi_0 = (0,0)$ and $\pi_K = (n-1, m-1)$
- for all $k > 0$, $\pi_k = (i_k, j_k)$ is related to $\pi_{k-1} = (i_{k-1}, j_{k-1})$ as follows:
  - $i_{k-1} \le i_k \le i_{k-1} + 1$
  - $j_{k-1} \le j_k \le j_{k-1} + 1$

Points could be double (?)

Examples for π:

$\pi_n = [\pi_0, \pi_1] = [(0,0), (n-1, m-1)]$

$\pi_2 = [\pi_0, \pi_1, \pi_2] = [(0,0), (1,2), (n-1, m-1)]$

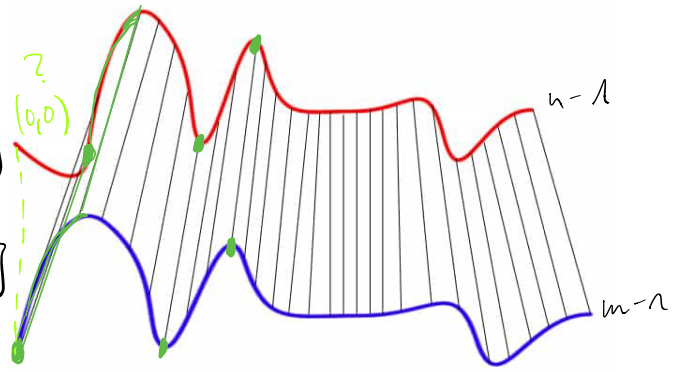
$0 \le 1 \le 1$

$0 \le 2 \le 1$ ✗

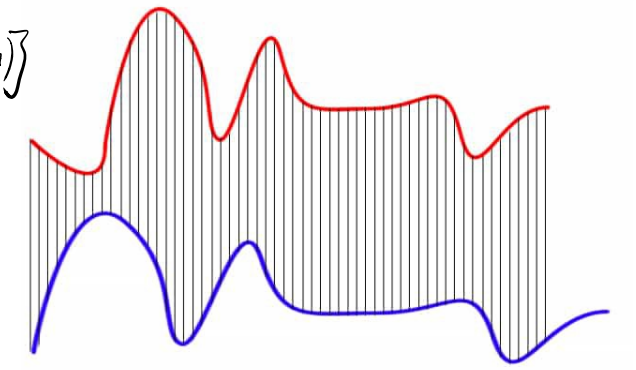
$\pi_3 = [\pi_0, \dots, \pi_3] = [(0,0), (0,1), (0,2), (n-1, m-1)]$

constructing a list is not trivial

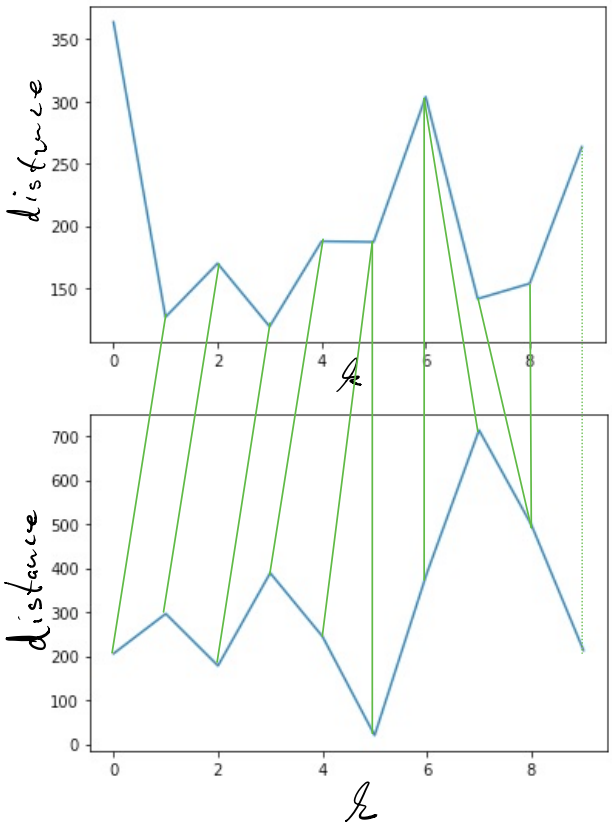
K is choosing how many data points I want to look at.



Dynamic Time Warping Matching



Euclidean Matching



	timestamp	x	y	label	distance
0	0	612	335	saccade	205.548048
1	43	729	504	saccade	296.195881
2	84	765	210	saccade	177.910090
3	151	791	386	saccade	388.820010
4	219	1156	252	saccade	244.697773
5	285	997	66	saccade	19.697716
6	351	979	58	saccade	388.253783
7	418	709	337	saccade	713.595123
8	483	862	-360	saccade	498.658200
9	550	1298	-118	saccade	213.103261

	timestamp	x	y	label	distance
0	0	521	441	saccade	363.743041
1	46	851	288	saccade	126.649122
2	92	969	334	saccade	169.752761
3	140	1109	238	saccade	119.067208
4	185	990	234	saccade	187.555325
5	232	1154	143	saccade	187.024063
6	300	967	140	saccade	303.644529
7	367	1057	430	saccade	141.315250
8	433	1110	299	saccade	153.629424
9	500	1171	158	saccade	263.427409

$$\pi = [(0,0), (1,0), (2,1), (3,2), (4,3), (5,4), (5,5), (6,6), (6,7), (7,8), (8,8), (9,9)]$$

$$K = 12 \qquad DTW_\pi(x,y) = \sqrt{\sum_\pi d(x_i, y_i)^2} \approx 739,6$$

```
pi = [(0,0), (1,0), (2,1), (3,2), (4,3), (5,4), (5,5), (6,6), (6,7), (7,8), (8,8), (9,9)]

def dtw_Pi(listOfTuple):
  sum = 0
  for el in l:
    sum += ( ts1["distance"].iloc[el[0]] - ts2["distance"].iloc[el[1]] )**2
  return math.sqrt(sum)


dtw_Pi(pi)
```

```
739.6096638313528
```

What does us tell this number?

Is π really optimal?