UNIVERSITY OF TWENTE.

Data Science

TOPIC: FEATURE EXTRACTION FROM TIME SERIES DATA

FAIZAN AHMED

2021-22-2A

CONTENTS

UNIVERSITY OF TWENTE.

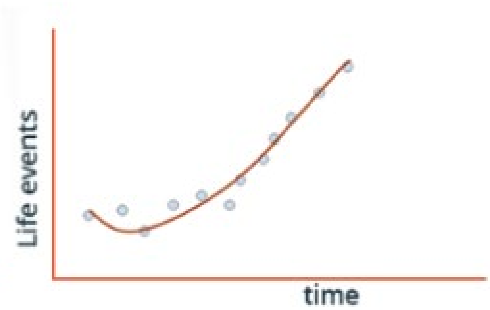
TIME SERIES FORECASTING

Reading Material

- Slides are based on the following online book: Forecasting: Principles and Practice
- Another interesting read (though more Python focused): Practical Time Series Analysis (available through university library)
- A mathematical treatment of the topic:
  - R. De Veaux, S. Fienberg, I. Olkin, Introduction to Time series and forecasting, Springer (available through university library)

TIME SERIES

- Number of observations collected over a successive period of time.
- Variable — *anything that changes over time*
- Time periods — *Can be daily, weekly, monthly, yearly*
- Variable Behaviour — *Quantifiable value*



https://medium.com/99xtechnology/time-series-forecasting-in-machine-learning-3972f7a7a467

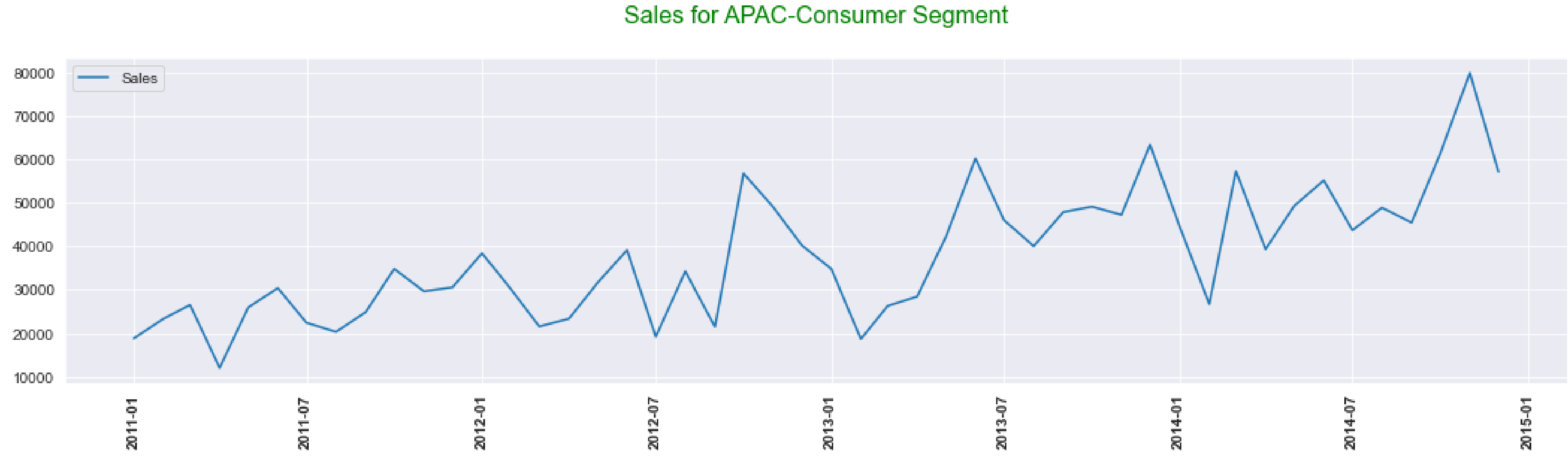
GOAL OF TIME SERIES ANALYSIS

Forecasting	What will future look like?
Filtering	Updating the estimate of the true state of nature
Time Scale Analysis	What time scale of variation dominates or account for the majority of temporal variation in the data
Regression Modelling	Given time series of two phenomenon what is the association between them
Smoothing	Given a complete (noisy) dataset, what can be inferred about the true state of nature in the past

Source: https://bookdown.org/rdpeng/timeseriesbook/goals-of-time-series-analysis.html

TIME SERIES VISUALIZATION

- Time Plot
  - observations vs time of observation



Source: https://medium.com/analytics-vidhya/time-series-forecasting-a-complete-guide-d963142da33f

CORRELATION

- Relative strength of linear relationship
- Unit-less
- Ranges between –1 and 1

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}}$$

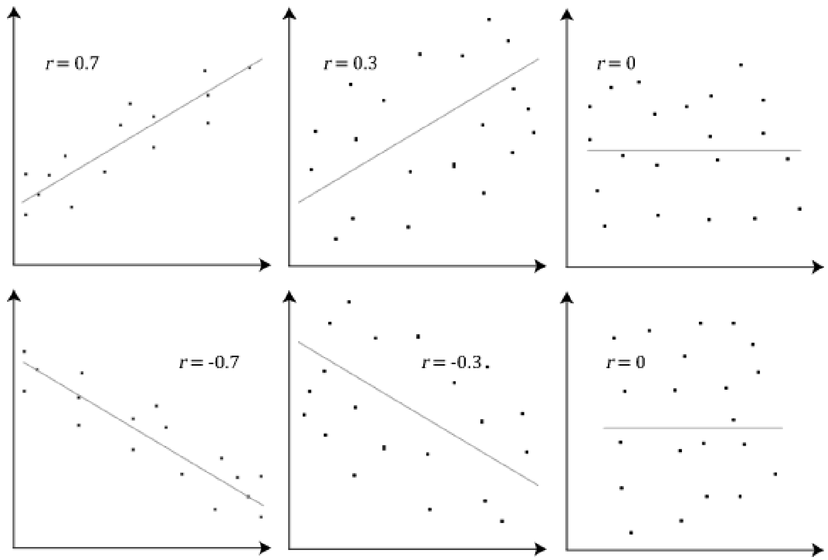
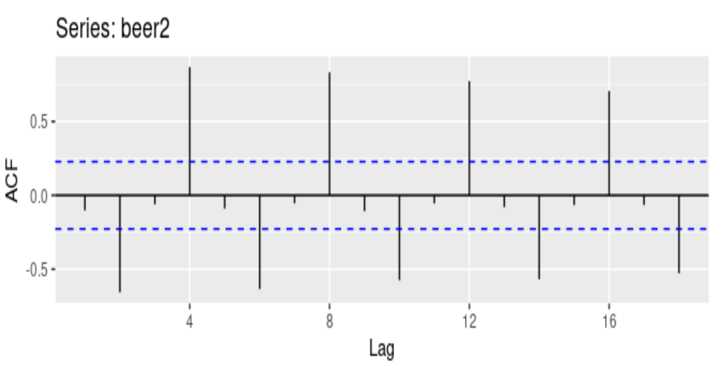


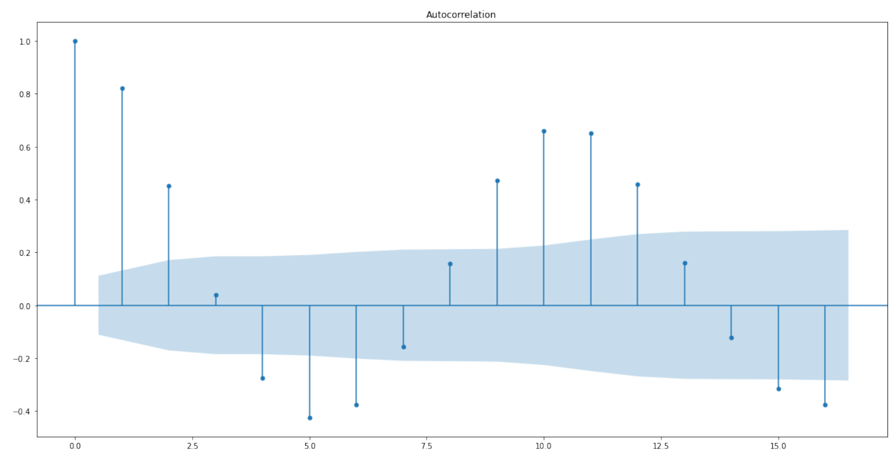
Image from: By Laerd Statistics - https://statistics.laerd.com/statistical-guides/pearson-correlation-coefficient-statistical-guide.php, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=84255641

AUTOCORRELATION

- Linear relationship between lagged values.

$$r_k = \frac{\sum_{t=k+1}^{T} (y - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^{T} (y_t - \bar{y})^2}$$

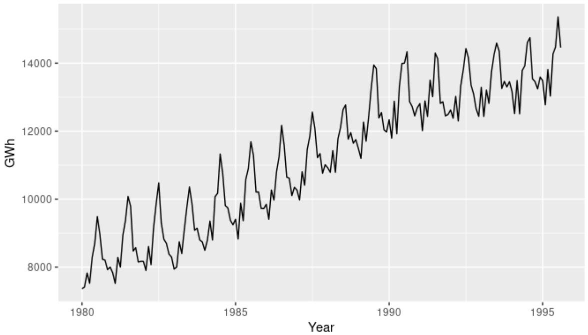


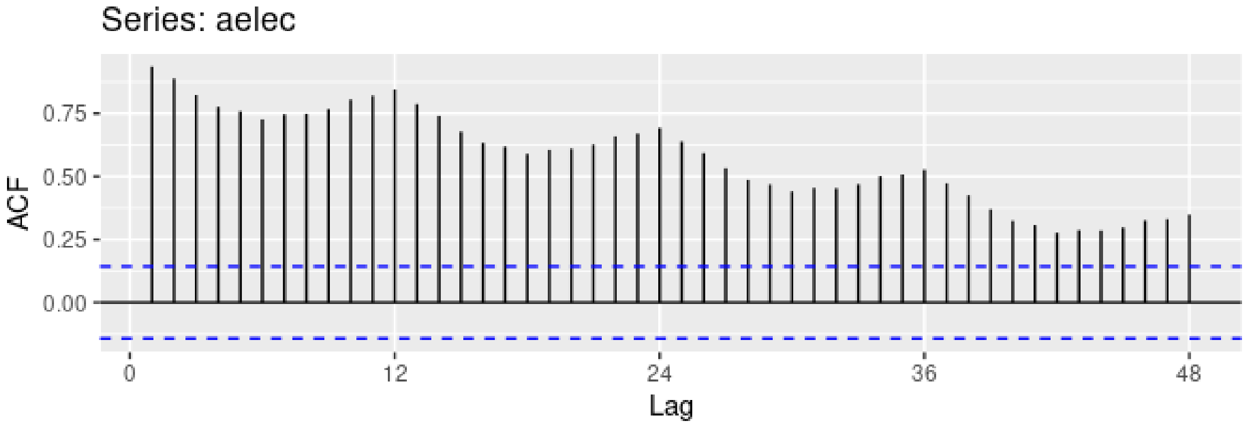


Source
https://otexts.com/fpp2/autocorrelation.html

AUTOCORRELATION

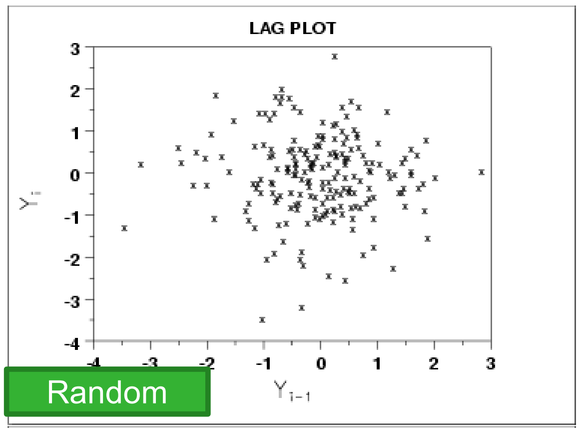
- Trend: the autocorrelations for small lags tend to be large and positive
- seasonal, the autocorrelations will be larger for the seasonal lags (at multiples of the seasonal frequency) than for other lags

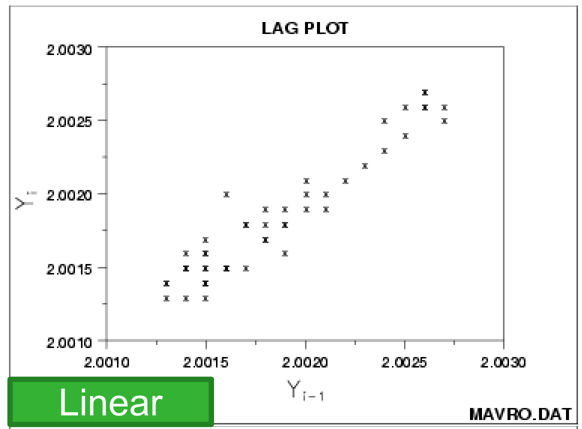


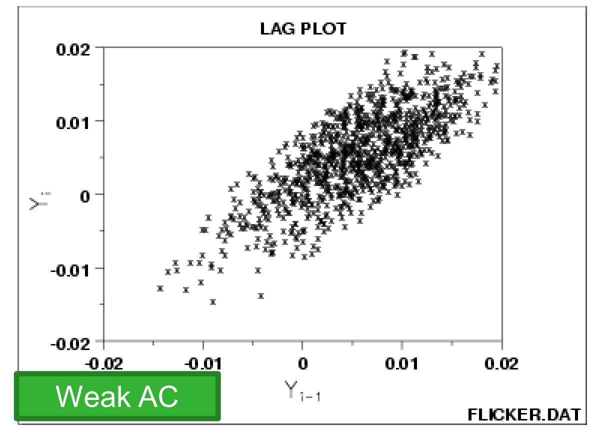


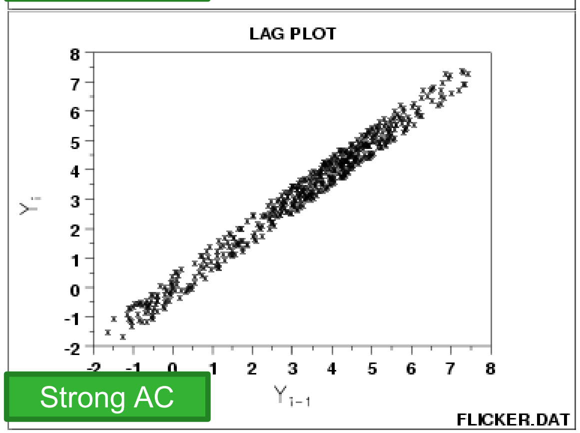
LAG PLOT

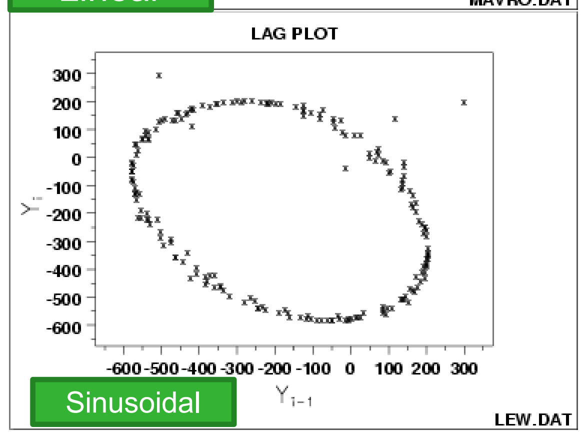
LAG PLOT CHECKS IF A TIME SERIES IS RANDOM OR NOT.







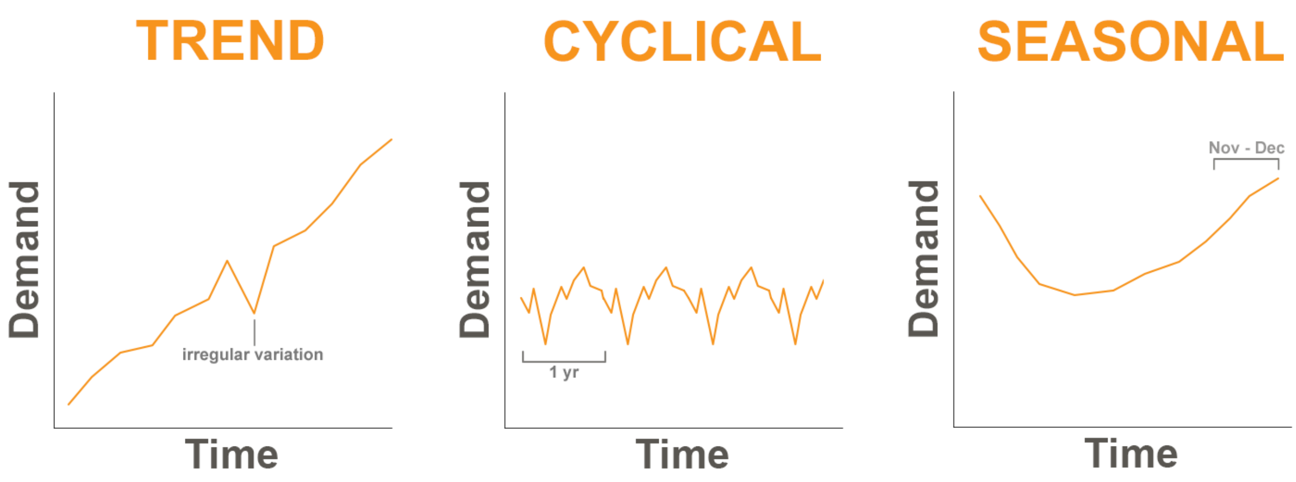




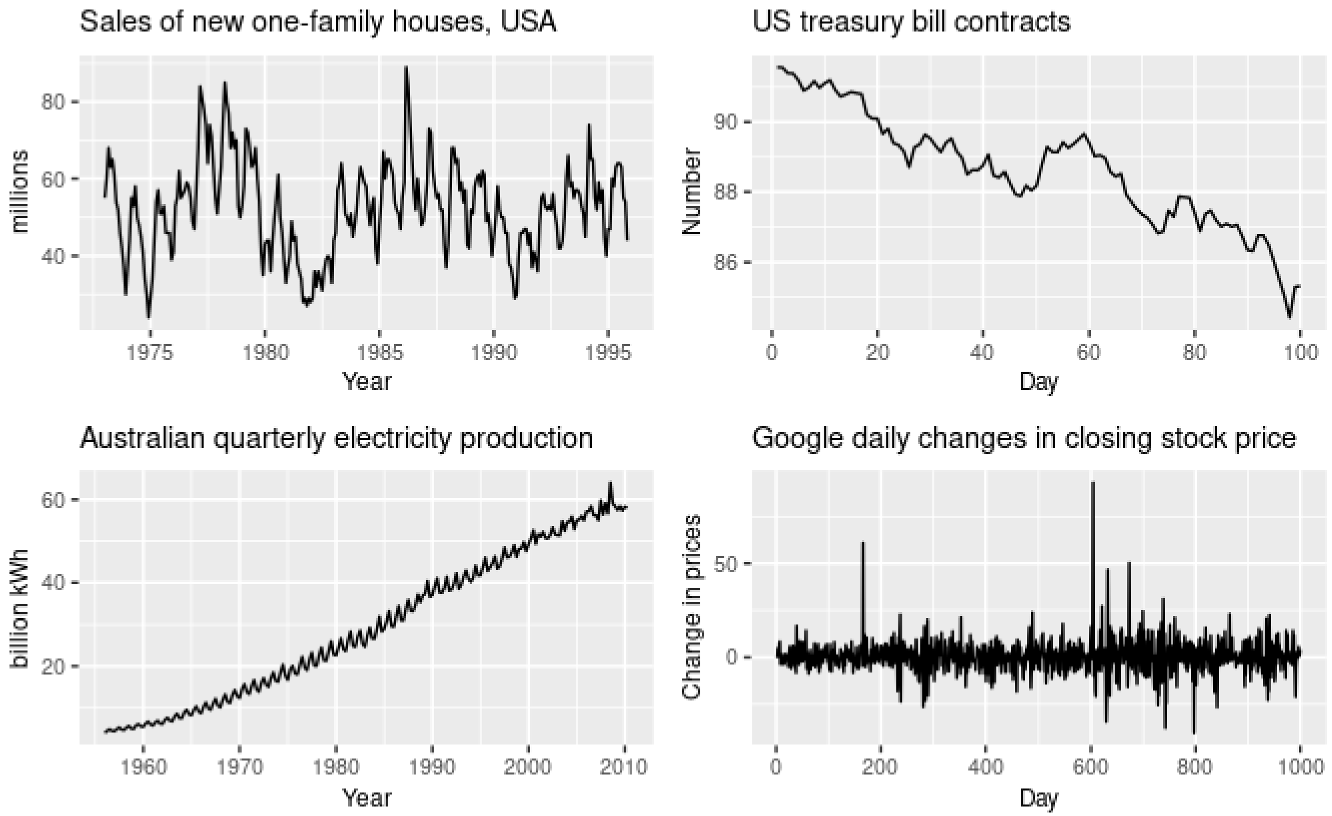
[Details….]

TIME SERIES PATTERNS

- ***Trend:*** *Long term increase or decrease*
- ***Seasonality:*** *Repetition of pattern over a period*
- ***Cyclicity:*** *Repetition of pattern aperiodically*



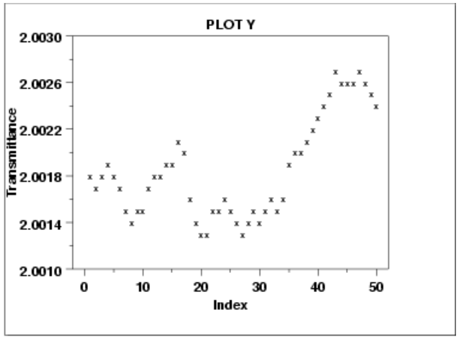
TIME SERIES PATTERNS

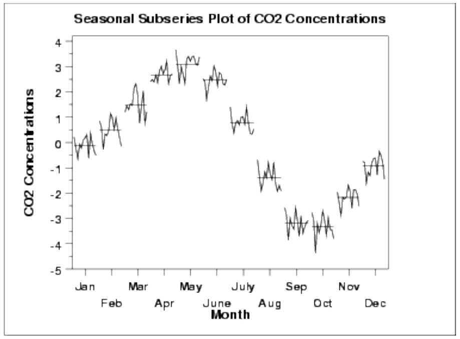


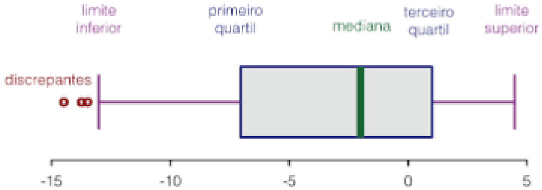
- Source: https://otexts.com/fpp2/tspatterns.html

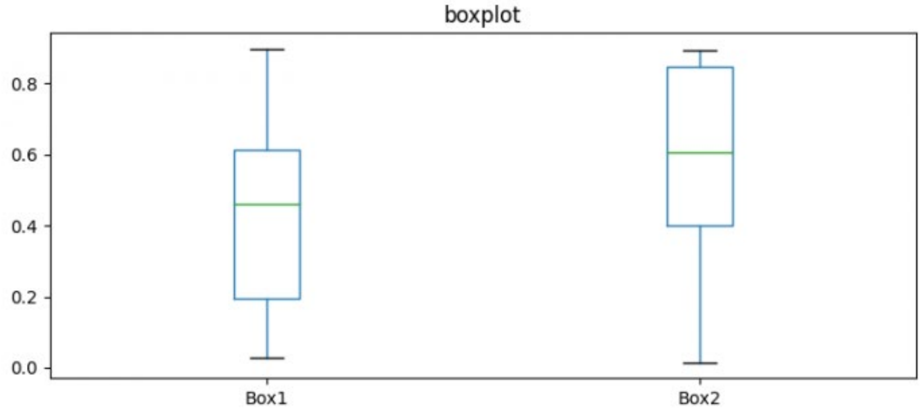
SEASONALITY THROUGH VISUALIZATION

- Run sequence plot
- Seasonal sub series plot
- Multiple box plots









TIME SERIES DECOMPOSITION

- Trend and cycle decomposed into a single trend-cycle component
- Additive time series:

$$Value = Base\ Level + Trend + Seasonality + Error$$

Assumption: These four components are independent of each other.

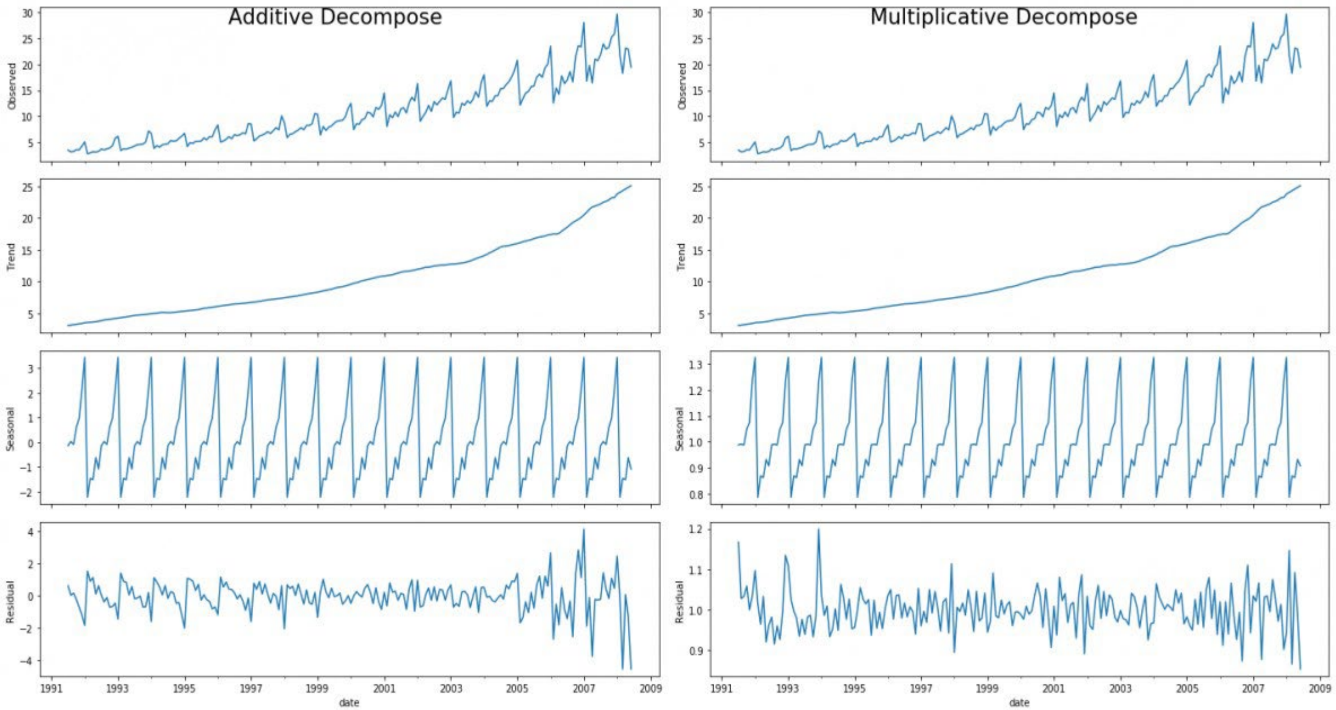
- Multiplicative time series:
  - *Assumption*: These four components of a time series are not necessarily independent, and they can affect one another.

$$Value = Base\ Level \times Trend \times Seasonality \times Error$$

- Decomposition
  - Improve understanding
  - Improve forecast accuracy

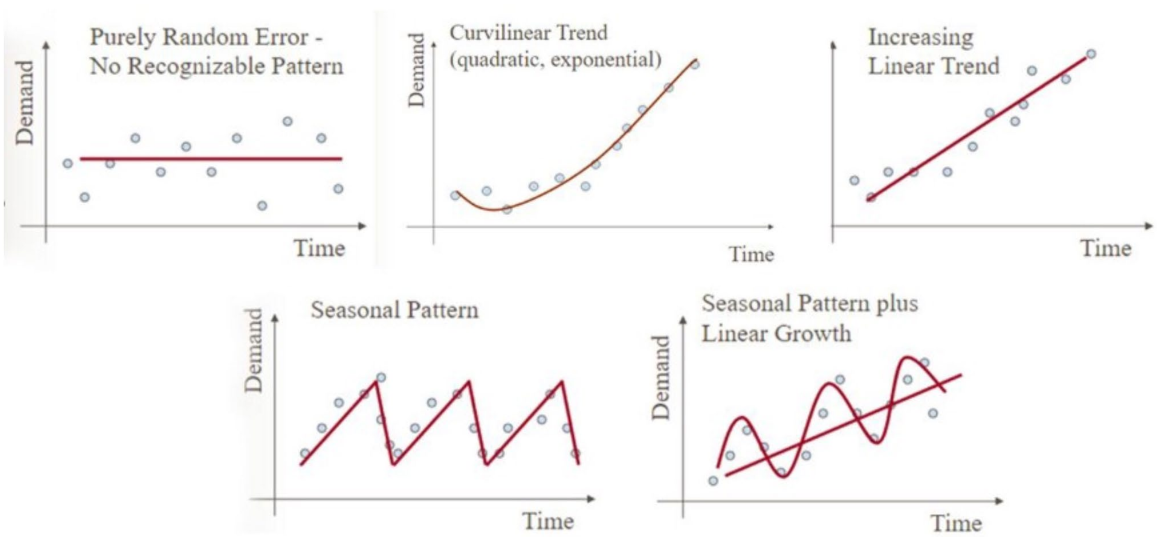
See also: https://otexts.com/fpp2/classical-decomposition.html sections 6.1-6.3

TIME SERIES DECOMPOSITION



TIME SERIES MODELLING

- Plot the series and examine main features
  - Trend
  - Seasonal component
  - Sharp changes in behaviour
  - Outlying observation



SIMPLE FORECASTING

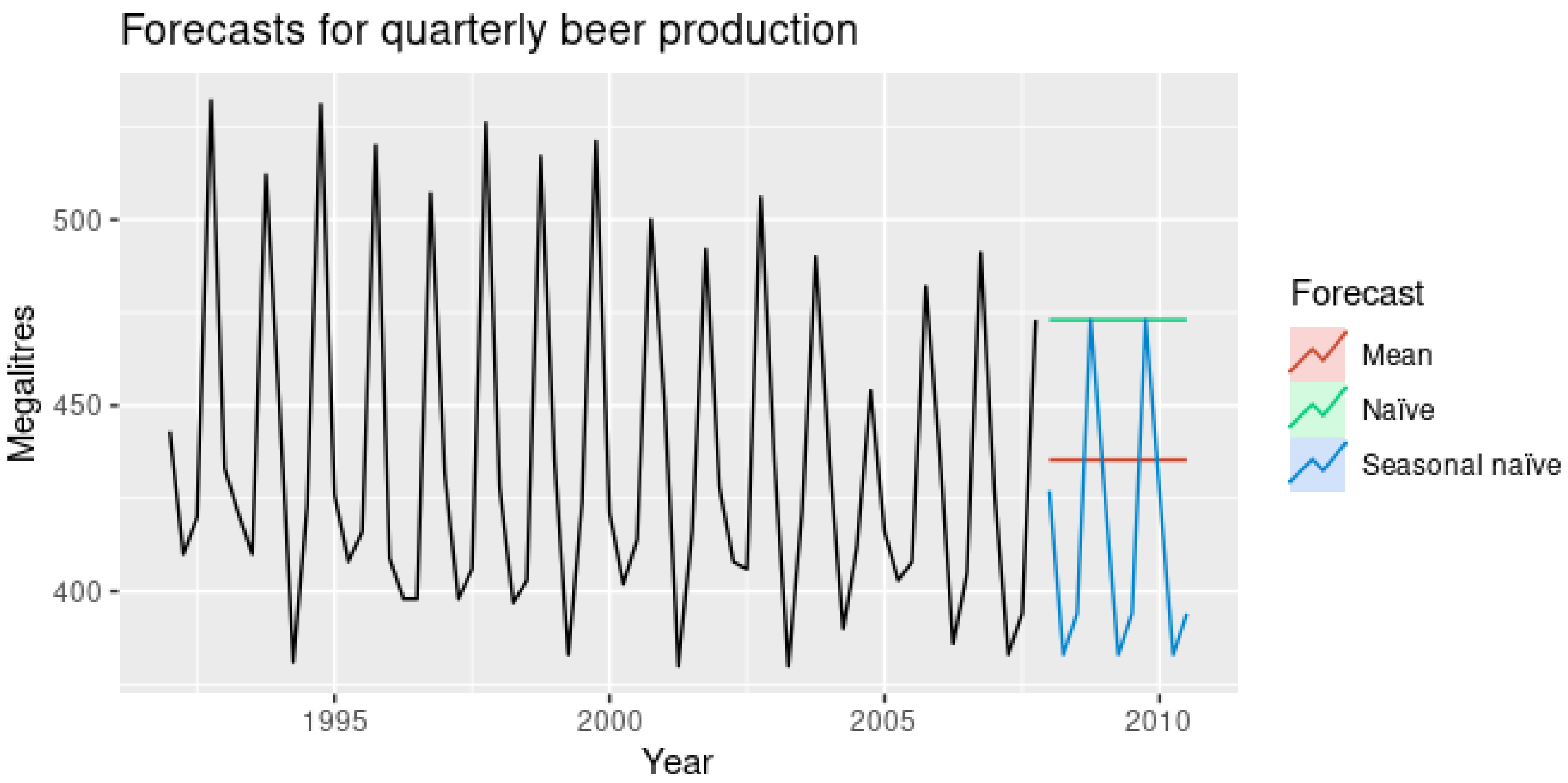
Average method: The forecast of all future values are equal to the average

Naïve method: Forecast is the last observed value

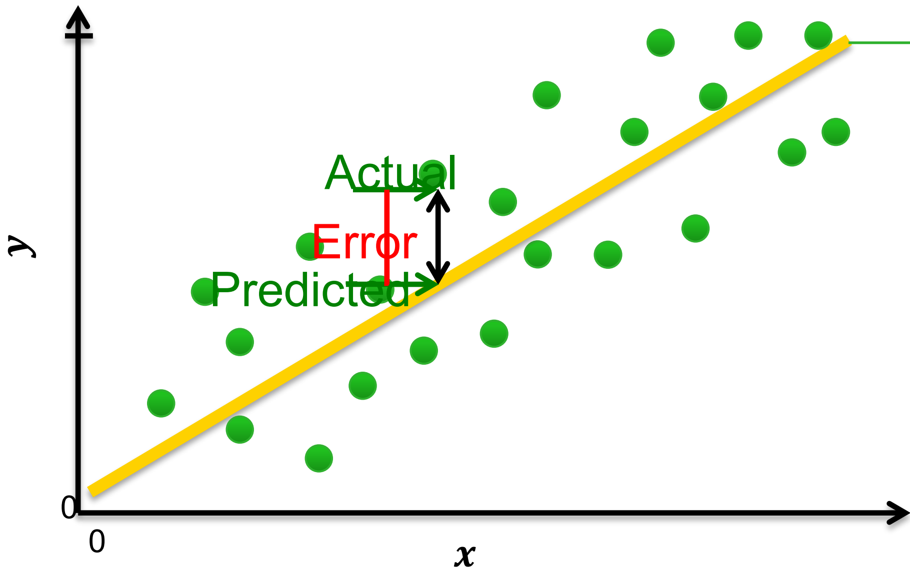
Seasonal naïve method: Last observed value from the same season

Drift method: Naïve+ average change in the data.

SIMPLE FORECASTING



LINEAR REGRESSION



$\overline{y} = \boldsymbol{ax} + b$

$$SSE(X) = \sum_{i=0}^{n} (y_i - \overline{y_i})^2$$

Find a and b such that SSE is minimum.

variable variable variable

$$y = a_1\, \boldsymbol{x^1} + a_2\boldsymbol{x^2} + \cdots + a_n\boldsymbol{x^n} + b$$

Multiple Regression

$\boldsymbol{x}$ or $\boldsymbol{x}^i$ - is vector

x_i is the i^{th} value of the vector $\boldsymbol{x}$

MEASURE OF PREDICTIVE ACCURACY

- Adjusted R^2
$$SSE = \sum_{t=1}^{T} e_t^2 \ , \overline{R}^2 = 1 - (1 - R^2)\frac{(T-1)}{T-k-1}$$
- Cross-validation (CV)
  - $while\ t = 1, \cdots, T$
    - Remove observation t
    - Fit the model
    - compute the error $e_t^* = y_t - \widehat{y}_t$
  - Compute MSE from $e_1^*, \cdots, e_T^*$

MEASURE OF PREDICTIVE ACCURACY-2

- Akaike's Information Criterion

$$AIC = Tlog\left(\frac{SSE}{T}\right) + 2(k+2)$$

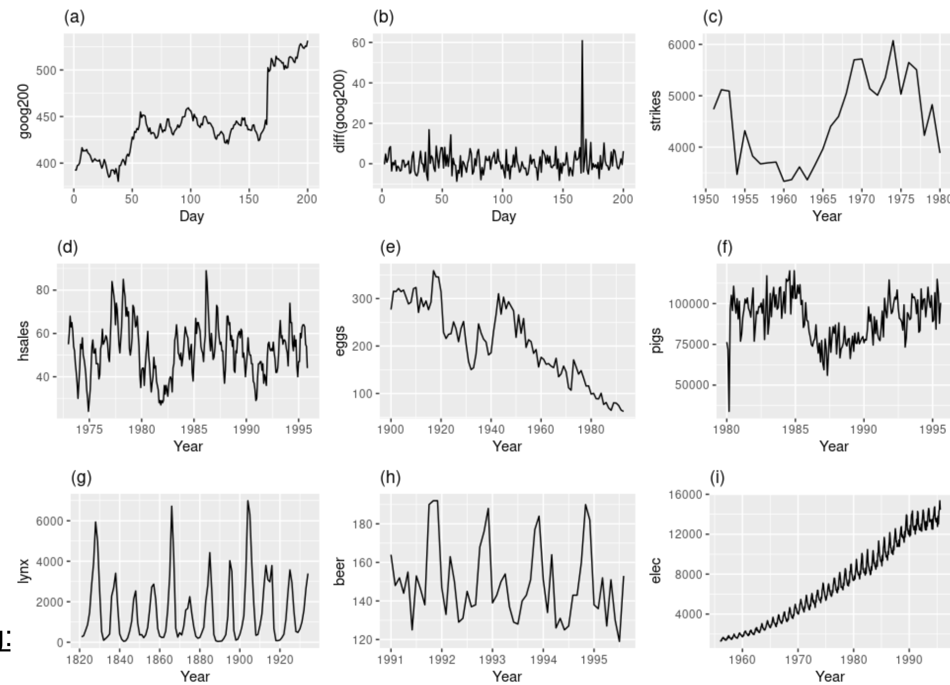
$$AIC_c = AIC + \frac{2(k+2)(k+3)}{T-k-3}$$

  - k =number of predictors, T =number of observations
  - Interesting read: https://towardsdatascience.com/introduction-to-aic-akaike-information-criterion-9c9ba1c96ced
- Schwarz's Bayesian Information Criterion

$$BIC = Tlog\left(\frac{SSE}{T}\right) + (k+2)\log(T)$$

STATIONARY AND NON-STATIONARY TIME SERIES

- A stationary time series is one whose properties do not depend on the time at which the series is observed.
  - Series with trend and seasonality is not stationary
  - white noise series is stationary
  - Time plots show constant variance



Further reading: 8.1 Stationarity and differencing | Forecasting: Principles and Practice (2nd ed) (otexts.com)

STATIONARY AND NON-STATIONARY TIME

- How to test for stationarity?
  - By looking at the plot of the series.[next slide]
  - Via Summary Statistics
    - split the series into 2 or more contiguous parts
    - compute the summary statistics like the mean, variance and the autocorrelation.
    - If the stats are quite different, then the series is not likely to be stationary.
  - using statistical tests called Unit Root Tests

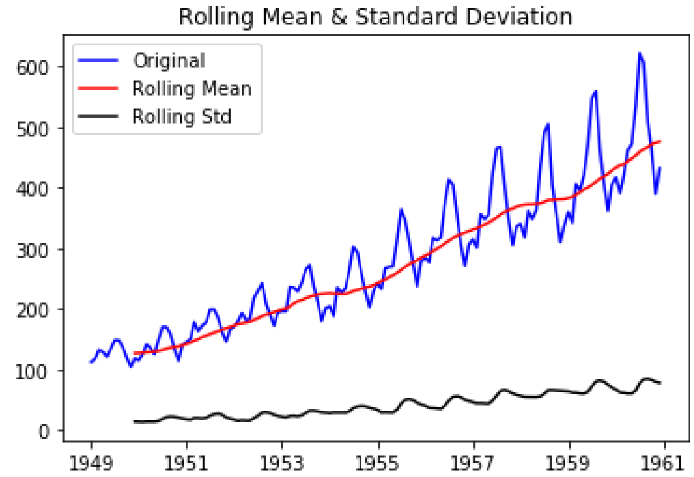


Image from: https://www.kaggle.com/code/freespirit08/time-series-for-beginners-with-arima

Further reading: https://otexts.com/fpp2/stationarity.html

STATIONARY AND NON-STATIONARY TIME SERIES

- How to make a time series stationary?
  - Differencing the series (once or more) -can help stabilise the mean of a time series
  - Take the log of the series-can help stabilise the variance
  - Take the *n*th root of the series -
  - Combination of the above

Further reading: 8.1 Stationarity and differencing | Forecasting: Principles and Practice (2nd ed) (otexts.com)

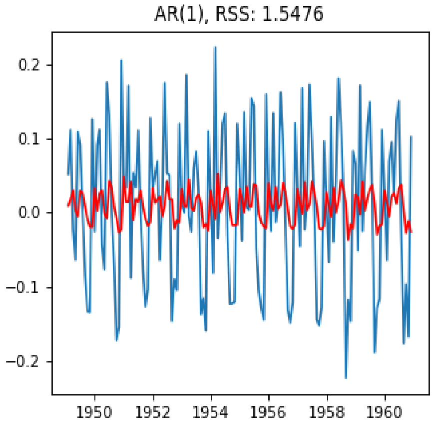
TIME SERIES MODELLING-AUTOREGRESSION

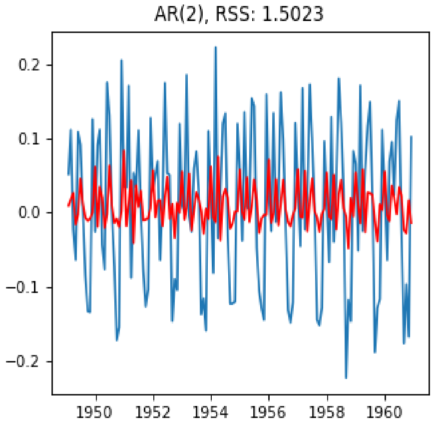
- **AutoRegression (AR(p))**- Regression with itself

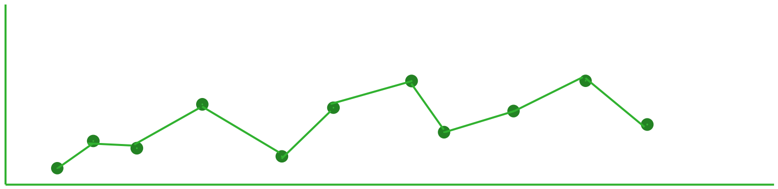
$$y_t = c + \phi_1 y_{t_{-1}} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \epsilon_t$$

$$= c + \sum_{i=1}^{p} \phi_i y_{t-i} + \epsilon_t$$

y_t –value at time t
e_t – random error
p – order of the model
c – constant
ϕ_i – model parameter







Further reading: 8.3 Autoregressive models | Forecasting: Principles and Practice (2nd ed) (otexts.com)
Python: https://www.statsmodels.org/devel/examples/notebooks/generated/autoregressions.html

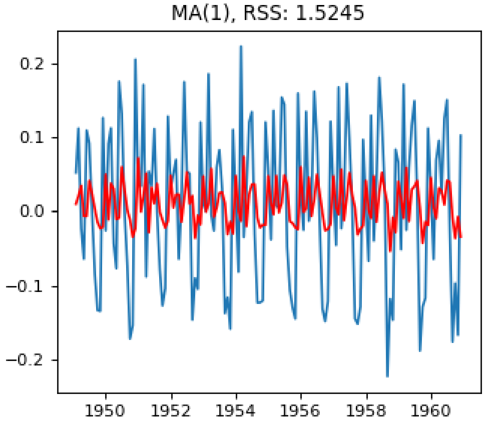
TIME SERIES MODELLING-MOVING AVERAGES

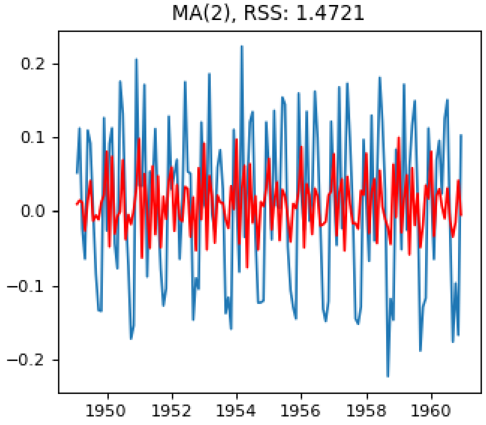
- **Moving Averages (MA(q))**- Make decision based on previous errors instead of previous values.

$$y_t = \mu + \theta_1\, \epsilon_{t_{-1}} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q y_{t-q} + \epsilon_t$$

$$= c + \sum_{i=1}^{q} \phi_q \epsilon_{t-i} + \epsilon_t$$

•For an MA(1) model: $-1 < \theta_1 < 1$
•For an MA(2) model:
$-1 < \theta_2 < 1\,, \theta_2 + \theta_1 < 1, \theta_2 - \theta_1 < 1$





How to do it in Python: https://www.statsmodels.org/stable/examples/notebooks/generated/tsa_arma_0.html

TIME SERIES MODELLING-ARMA

- ARMA(p , q)

$$y_t = c + \sum_{i=1}^{p} \phi_p y_{t-i} + \sum_{i=1}^{q} \phi_q \epsilon_{t-i} + \epsilon_t$$

- How to select p, q:
  - Use ACF/PACF
  - Apply grid search together with some model selection technique

How to do it in Python: https://www.statsmodels.org/stable/examples/notebooks/generated/tsa_arma_0.html

TIME SERIES REGRESSION MODELS

- Autoregression (AR) Moving Average (MA)
- Autoregressive Moving Average (ARMA) Autoregressive Integrated Moving Average (ARIMA)
- Seasonal Autoregressive Integrated Moving-Average (SARIMA)
- Seasonal Autoregressive Integrated Moving-Average with Exogenous Regressors (SARIMAX)
- Vector Autoregression (VAR)
- Vector Autoregression Moving-Average (VARMA)
- Vector Autoregression Moving-Average with Exogenous Regressors (VARMAX) Simple Exponential Smoothing (SES)
- Holt Winters Exponential Smoothing (HWES)

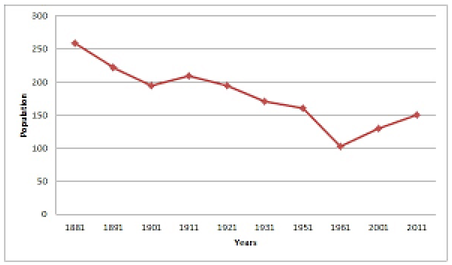
UNIVERSITY OF TWENTE.

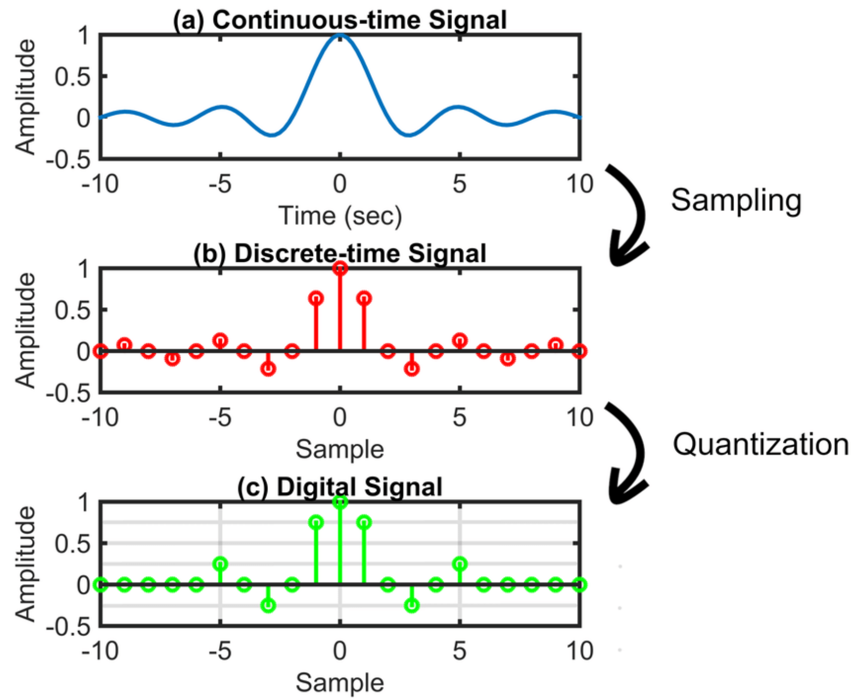
FEATURE SELECTION & FEATURE EXTRACTION

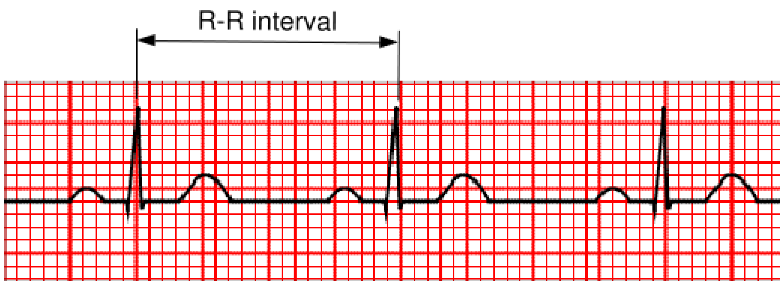
FEATURE EXTRACTION

TIME SERIES AND SIGNAL

SIGNAL: TIME VARYING QUANTITIES THAT REPRESENT PHYSICAL EVENTS.







FEATURE EXTRACTION

FEATURE EXTRACTION METHODS

Feature extraction is a process that identifies important features or attributes of the data.

Objective

- To find a set of
  - distinctive,
  - informative, and
  - reduced features.

Feature Extraction Methods

- **Parametric**
- Non-parametric
  - Power spectral density, spectrogram, etc
- Eigenvector methods
  - Eigen decomposition based, Multiple signal classification, Pisarenko, etc
- **Time-frequency**

FEATURE EXTRACTION: PARAMETRIC METHODS

Produces a signal model with known signal form

Estimates parameters in the produced model

Examples:

- Autoregressive
- Moving Average
- Autoregressive moving average
- Lyapunov exponents

FEATURE EXTRACTION : TIME FREQUENCY METHODS

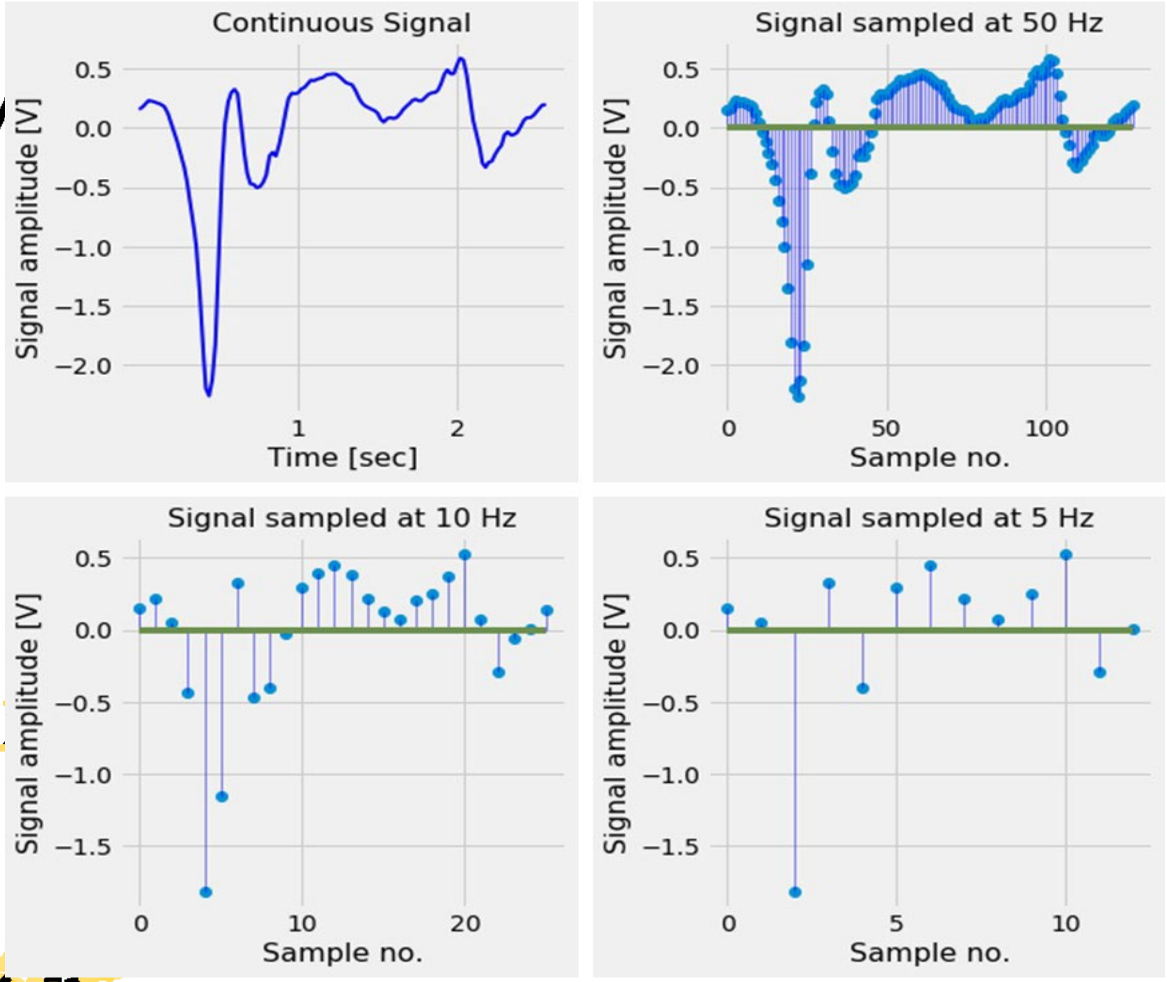
Extensively employed in medical domain

Examples Includes:

- **Fourier Transformation**
  - Short-time Fourier transform
- Wavelet transform
  - Discrete Wavelet transform
  - Wavelet packet decomposition
  - Stationary wavelet transform
  - …
- Empirical mode decomposition
  - Ensemble EMD

FAST FOURIER TRANSFORMATION

SIGNALS VS TIME-SERIES

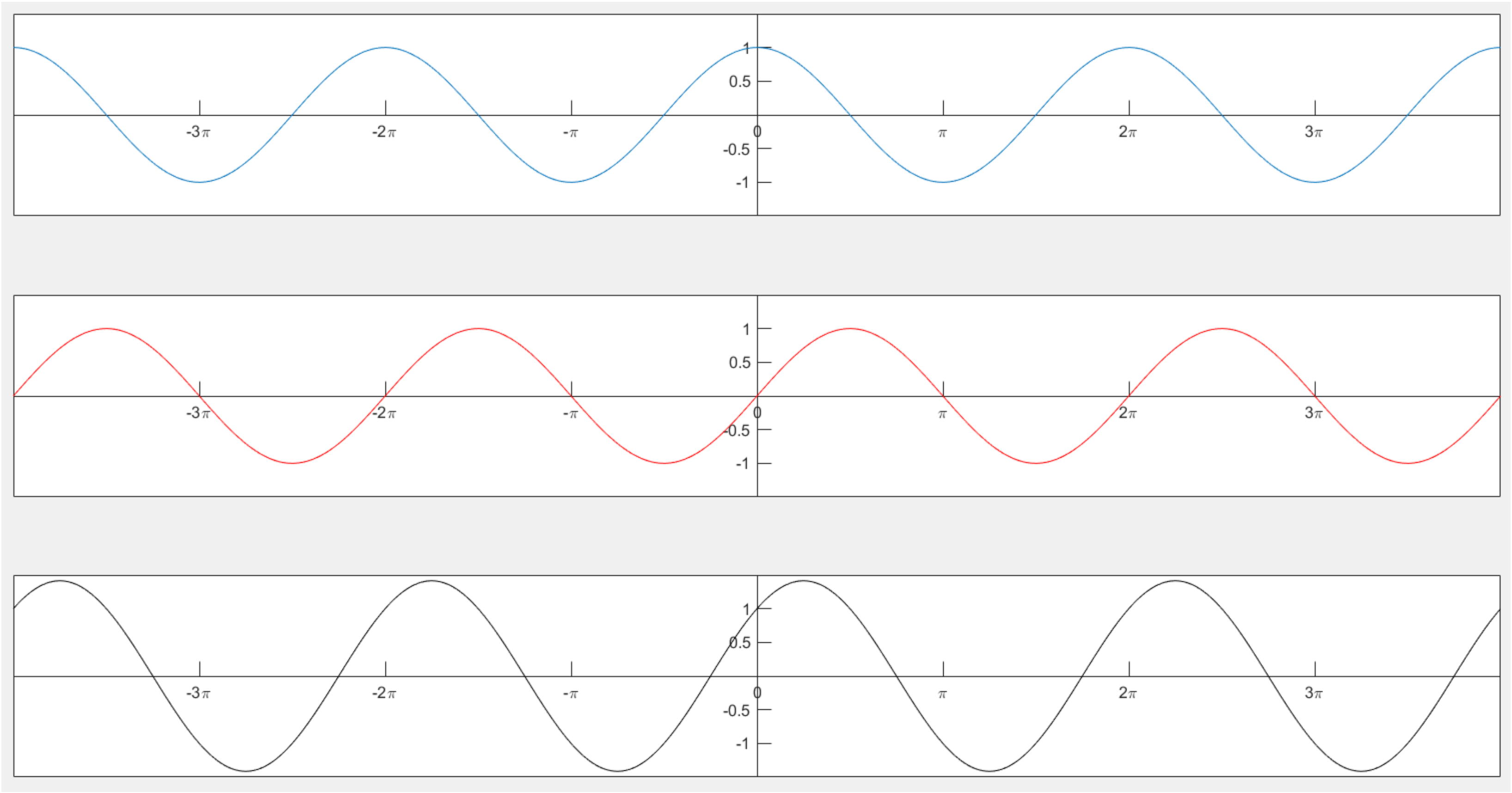


- our focus

Discrete Time Signals

- a signal is said to be under-sampled if the sampling rate is smaller than the Nyquist rate. The Nyquist rate is twice the highest frequency present in the signal.

FAST FOURIER TRANSFORMATION



FROM TIME-DOMAIN TO FREQUENCY DOMAIN

- How we can transform a periodical signal (time series) from the time-domain to the frequency-domain?
- Some possible solutions are
  - Fast Fourier Transform (FFT)
  - Power Spectral Density (PSD)
  - The auto-correlation function
  - The Wavelet Transform

FAST FOURIER TRANSFORM (FFT)

- Fourier series are periodic functions and signals may be expanded into a series of sine and cosine functions

$$X_t = \sum_{n=0}^{N-1} x_n \left[\cos\left(\frac{2\pi}{N} tn\right) - i\, sin\left(\frac{2\pi}{N} tn\right)\right]$$

- A Fourier transform takes one function (or signal) and turns it into another function (or signal)
- The Discrete Fourier Transform:
- Why FFT is so popular?
  - In 1969, the 2048 point analysis of a seismic trace took $13^{1/2}$ hours. Using the FFT, the same task on the same machine took 2.4 seconds!

FAST FOURIER TRANSFORM (FFT)

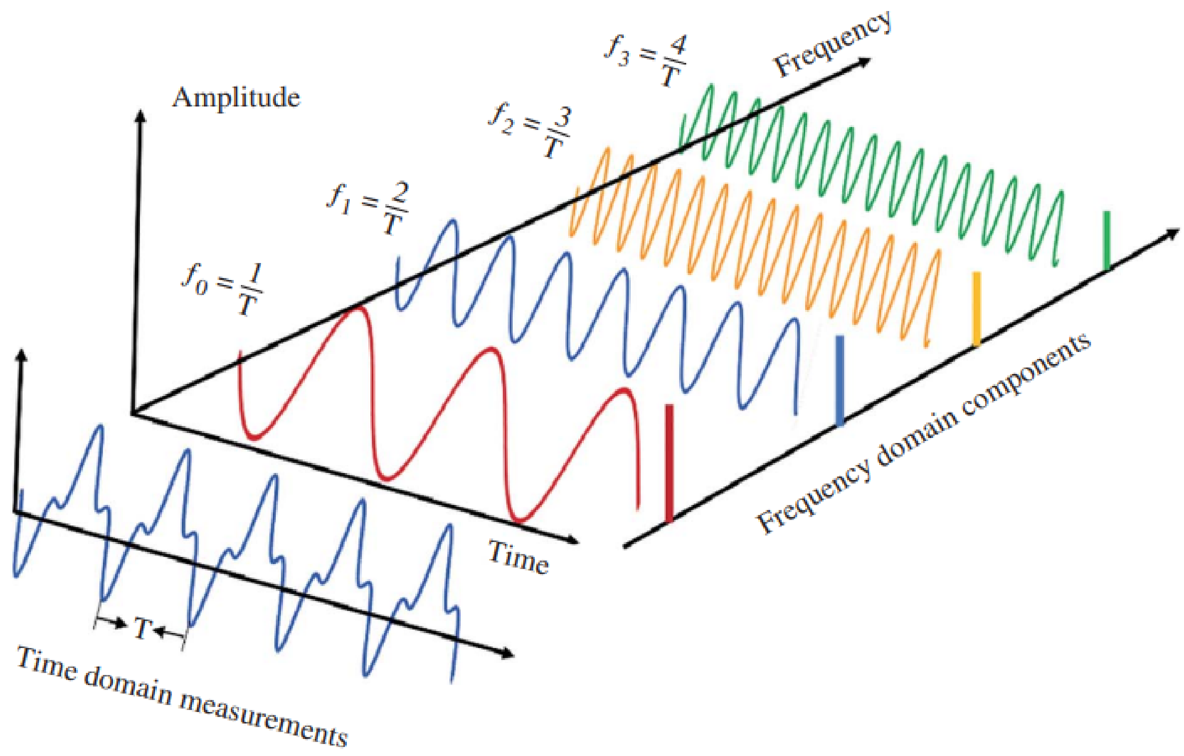
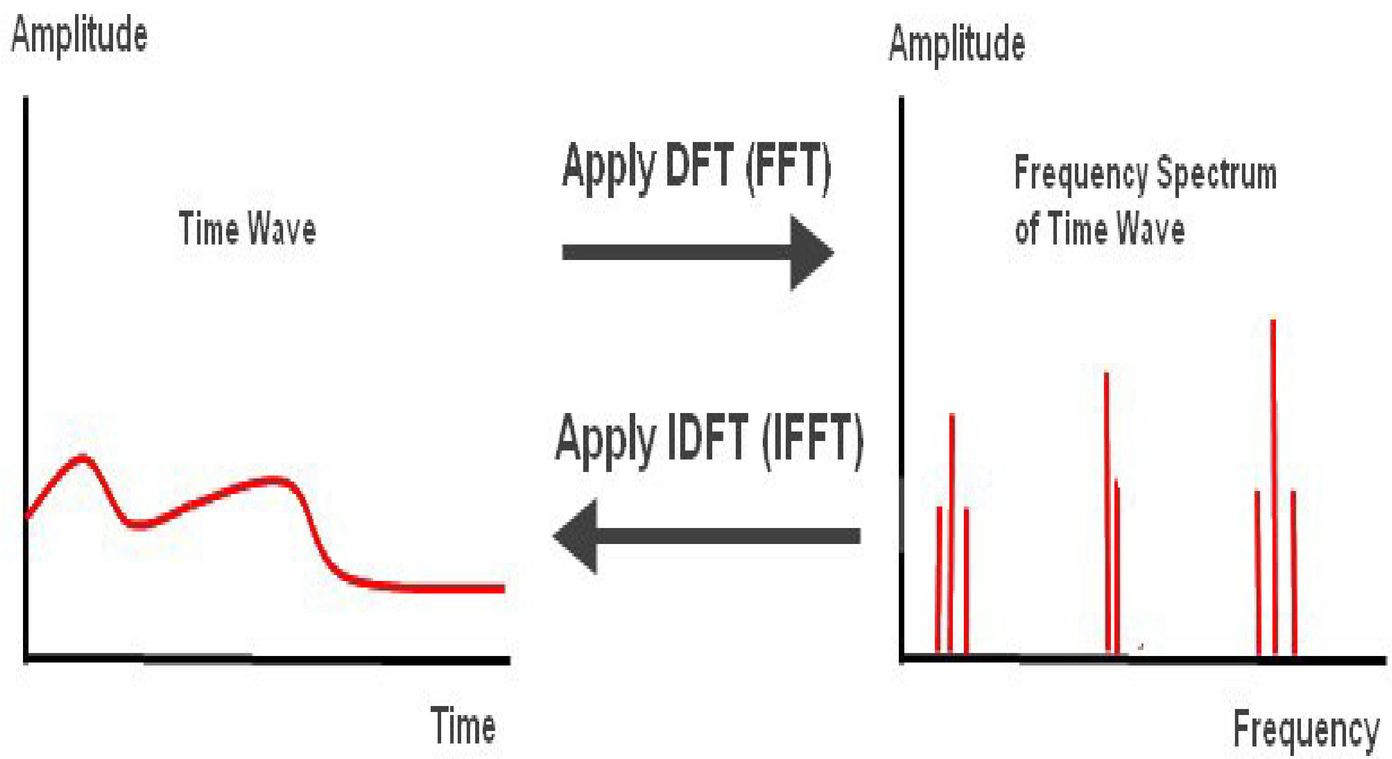


Figure 4.1 Time domain measurements vs. frequency domain measurements. Source: (Brandt 2011).

FAST FOURIER TRANSFORM (FFT)



In Python: Package scipy, https://docs.scipy.org/doc/scipy/reference/tutorial/fft.html
Further reading: 10.1002/9781119544678.ch4

TIME DOMAIN AND FREQUENCY DOMAIN FEATURES

Time domain	Frequency domain
Mean	Mean
Standard deviation	Standard deviation
Inter Quartile range	Maximum
Percentile	Minimum
Peak-to-peak-amplitude	Spectral centroid
Power	Dominant frequency
Skewness	Ratio Low Frequency, High Frequency
Kurtosis	Entropy
Root squared error	.
.	.
.	.

Further reading: 10.1002/9781119544678.ch4
https://ataspinar.com/2018/04/04/machine-learning-with-signal-processing-techniques/

FEATURE EXTRACTION: EPOCH/TIME WINDOW/ EVENT LENGTH

Signal divided to smaller interval, generally equal length

- Experiment design decision

Two methods

- Non-overlapping
- Overlapping
  - Generally, in terms of percentage (10%,25%, 50% etc)
  - Rolling window

EPOCH/TIME WINDOW/ EVENT LENGTH

Time Unit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Value	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

Window size 6.

		1	2	3	4	5	6
1 to 6	Window 1	1	2	3	4	5	6
7 to 12	Window 2	7	8	9	10	11	12
13 to 18	Window 3	13	14	15	16	17	18
19 to 24	Window 4	19	20	21	22	23	24
25-30	Window 5	25	26	27	28	29	30

Number of observations in time unit

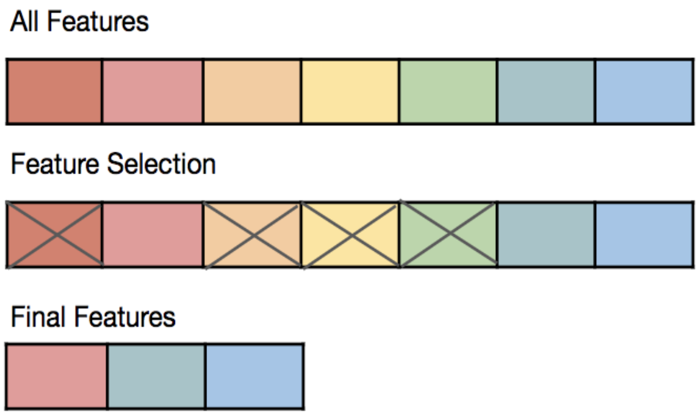
Window size 6 (50% overlap)

1 to 6	Window 1	1	2	3	4	5	6
4 to 9	Window 2	4	5	6	7	8	9
7 to 12	Window 3	7	8	9	10	11	12
10 to 15	Window 4	10	11	12	13	14	15
13 to 18	Window 5	13	14	15	16	17	18
16 to 21	Window 6	16	17	18	19	20	21
19 to 24	Window 7	19	20	21	22	23	24
22 to 27	Window 8	22	23	24	25	26	27
25 to 30	Window 9	25	26	27	28	29	30

1 to 6	Window 1	1	2	3	4	5	6	1
4 to 9	Window 2	4	5	6	7	8	9	0
7 to 12	Window 3	7	8	9	10	11	12	1
10 to 15	Window 4	10	11	12	13	14	15	0
13 to 18	Window 5	13	14	15	16	17	18	1
16 to 21	Window 6	16	17	18	19	20	21	0
19 to 24	Window 7	19	20	21	22	23	24	1
22 to 27	Window 8	22	23	24	25	26	27	0
25 to 30	Window 9	25	26	27	28	29	30	1

Time Unit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Value	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Label	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1

FEATURE SELECTION



- A process that chooses an optimal subset of features according to an objective function
- Why?
  - To reduce dimensionality and remove noise
  - To improve mining performance
    - Speed of learning
    - Predictive accuracy
    - Simplicity and comprehensibility of mined results

For tabular data it means reducing the number of columns.

FEATURE SELECTION

- Filter Methods - use a ranking criterion to score the features and then remove the features with scores below a threshold.
  - Ex. Laplacian score, Correlation, Mutual Information, Bayesian scoring function, t-test scoring, and Information theory based criteria
  - Example: Fast Fourier Transform based filters
- Wrapper Methods - evaluate different subsets of features to detect the best subset.
  - Unlike the filter methods, they consider the interaction between features that may lead to better accuracy. However, this makes them more computation intensive.
  - Example: Dynamic Time Warping
- Autoencoders, Denoising Autoencoders, ...
  - Example: Concrete Autoencoder (CAE) method selects features by learning a concrete distribution over input features.
- Embedded Methods - unify the learning process and the feature selection

Further Reading: https://ieeexplore-ieee-org.ezproxy2.utwente.nl/abstract/document/7160458

FACEBOOK PROPHET

- introduced by Facebook (S. J. Taylor & Letham, 2018)
- originally developed for forecasting daily data with weekly and yearly seasonality, plus holiday effects.
- works best with time series that have strong seasonality and several seasons of historical data
- Nonlinear regression model of the form

$$y_t = g(t) + s(t) + h(t) + \epsilon_t$$

$g(t)$=piecewise linear trend

$s(t)$=various seasonal pattern

$h(t)$=(captures)holiday effect

ϵ_t=white noise error term

WHICH MODEL TO CHOOSE?

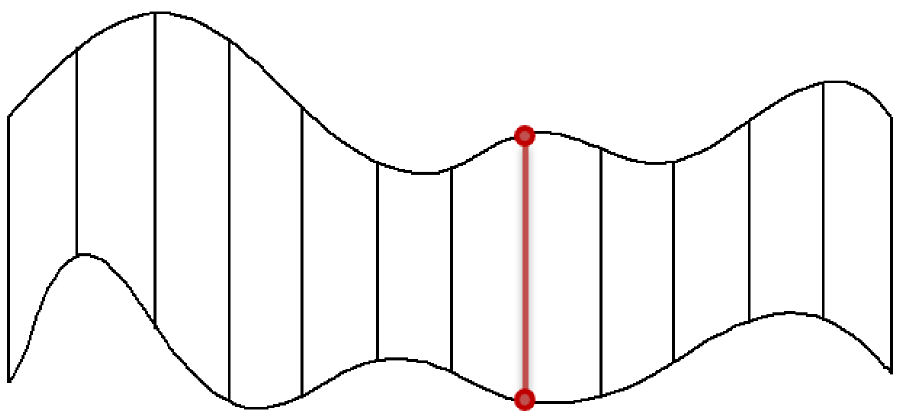
- The choice of **the most appropriate method of forecasting**
- is influenced by a number of factors, that are:
  - **Forecast horizon**, in relation to TSA objectives
  - Type/amount of **available data**
  - Expected **forecastability**
  - Required **readability** of the results
  - **Number of series** to forecast
  - **Deployment** frequency of the models
  - Development **complexity**
  - Development **costs**

DYNAMIC TIME WARPING

HOW TO ALIGN TWO SERIES?

DYNAMIC TIME WARPING

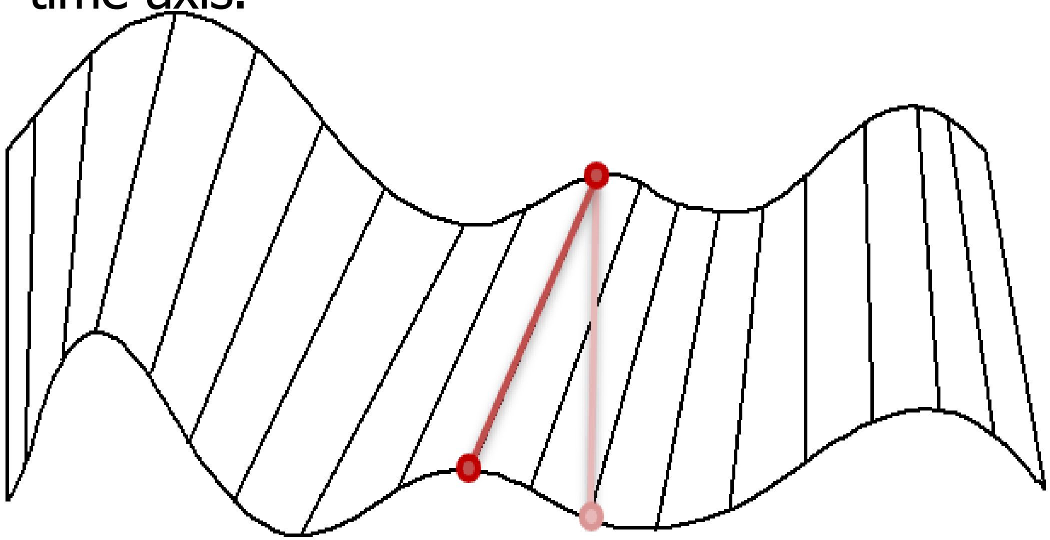
- Any distance (Euclidean, Manhattan,) which aligns the *i*-th point on one time series with the *i*-th point on the other will produce a poor similarity score.



- Euclidean distance
  - does not align values
  - both time series need to be of the same size

DYNAMIC TIME WARPING

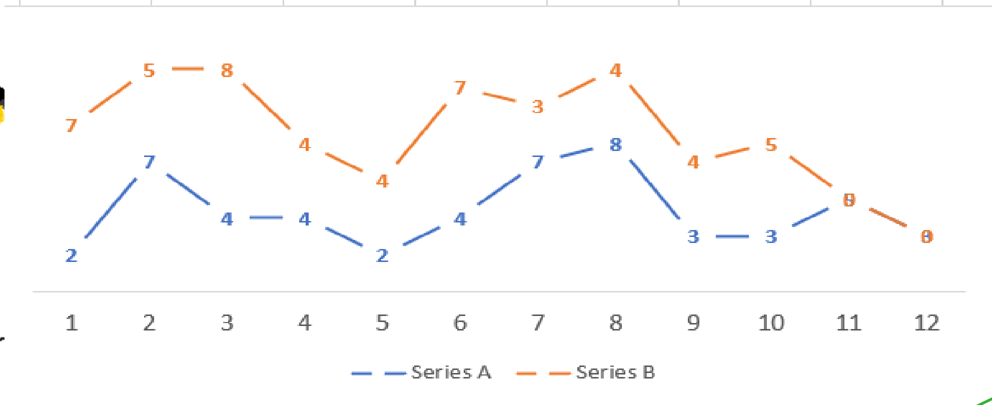
- A non-linear (elastic) alignment produces a more intuitive similarity measure, allowing similar shapes to match even if they are out of phase in the time axis.



- DTW
  - aligns two time series in the way some distance measure is minimized
  - time series sizes may vary

DYNAMIC TIME WARPING

EXAMPLE

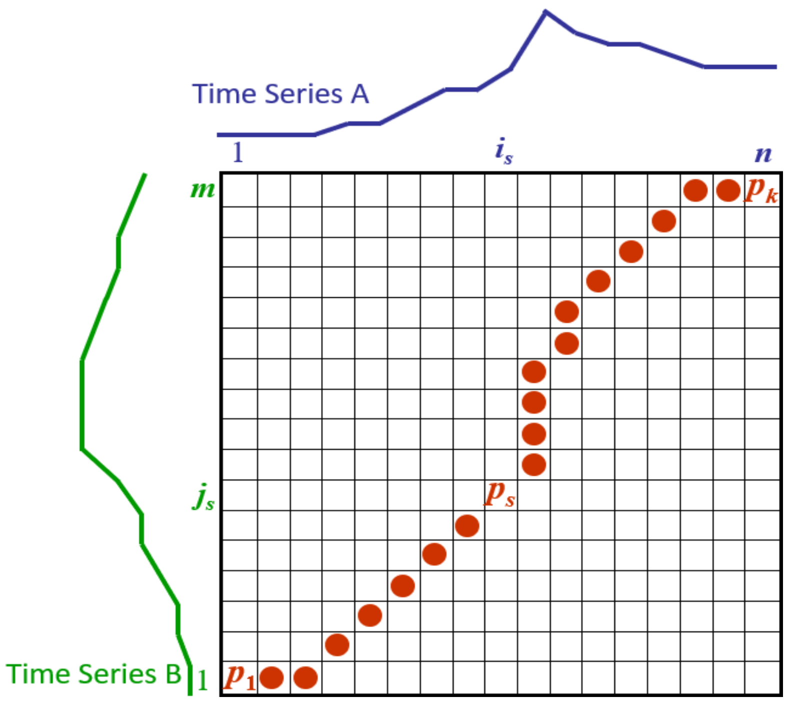


index

		1	2	3	4	5	6	7	8	9	10	11	12
		2	7	4	4	2	4	7	8	3	3	5	3
10	5						6	7	10	10	10	8	10
9	4					5	5	8	12	8	8	8	8
8	4				3	5	5	8	12	7	7	7	7
7	3			3	4	5	6	9	10	6	6	8	6
6	7		2	5	6	8	8	5	6	8	8	6	
5	4	2	5	3	3	5	5	8	10	4	4		
4	4	2	5	3	3	5	5	6	6	3			
3	8	5	3	5	5	8	8	3	2				
2	5	3	2	1	2	5	4	2					
1	7	5	0	3	3	5							

DYNAMIC TIME WARPING

- To find the best alignment between A and B one needs to find the path through the grid $P = p_1, , p_s, , p_k$, where $p_s = (i_s, j_s)$ which minimizes the total distance between them. P is called a warping function.



DYNAMIC TIME WARPING

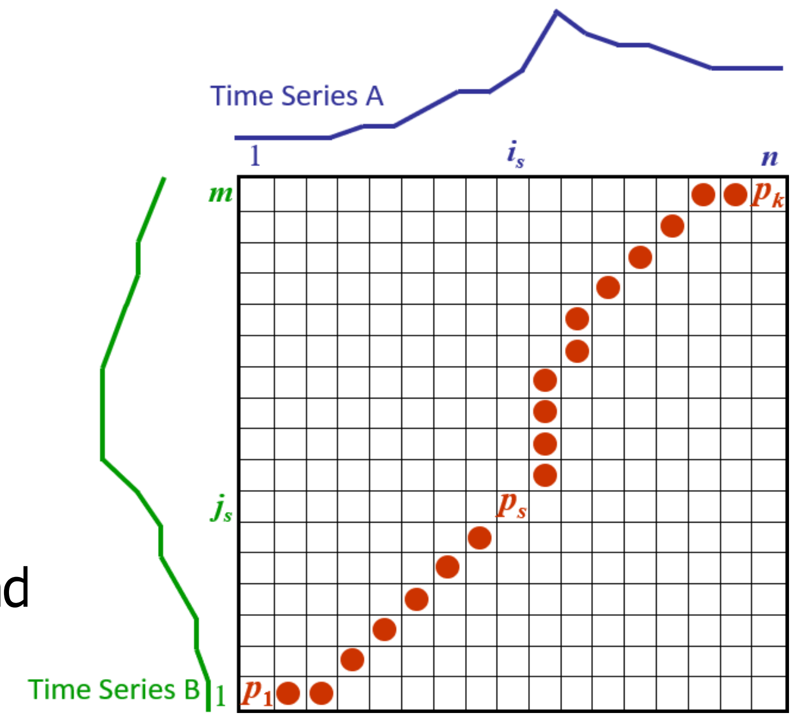
- Time-normalized distance measure between A and B:

$$D(A,B) = \left[\frac{\sum_{s=1}^{k} d(p_s).w_s}{\sum_{s=1}^{k} w_s}\right]$$

where $d\,(p_s)$ is the distance between i_s and j_s, and w_s are weighting coefficient.

- Best alignment path between A and B :

$$P_0 = \arg\min_{p} D(A,B)$$

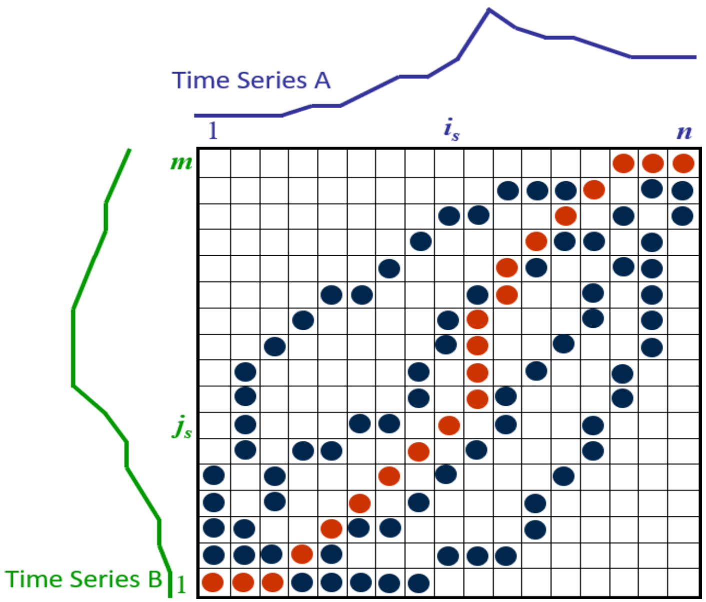


DYNAMIC TIME WARPING

- The number of possible warping paths through the grid is exponentially explosive!
- Reduction of the search space

→ Restrictions on the warping function:

  - monotonicity
  - continuity
  - boundary conditions
  - warping window
  - slope constraint.



DYNAMIC TIME WARPING

COMPUTING WEIGHT COEFFIECIENTS

- Time-normalized distance measure between A and B:

$$D(A,B) = \min_{p}\left[\frac{\sum_{s=1}^{k} d(p_s).w_s}{\sum_{s=1}^{k} w_s}\right]$$

Take: $C = \sum_{s=1}^{k} w_s$

- Symmetric: $w_s = (i_s - i_{s-1}) + (j_s - j_{s-1})$ than $C = n + m$
- Asymmetric: $w_s = (i_s - i_{s-1})$ than $C = n$ OR

  $w_s = (j_s - j_{s-1})$ than $C = m$

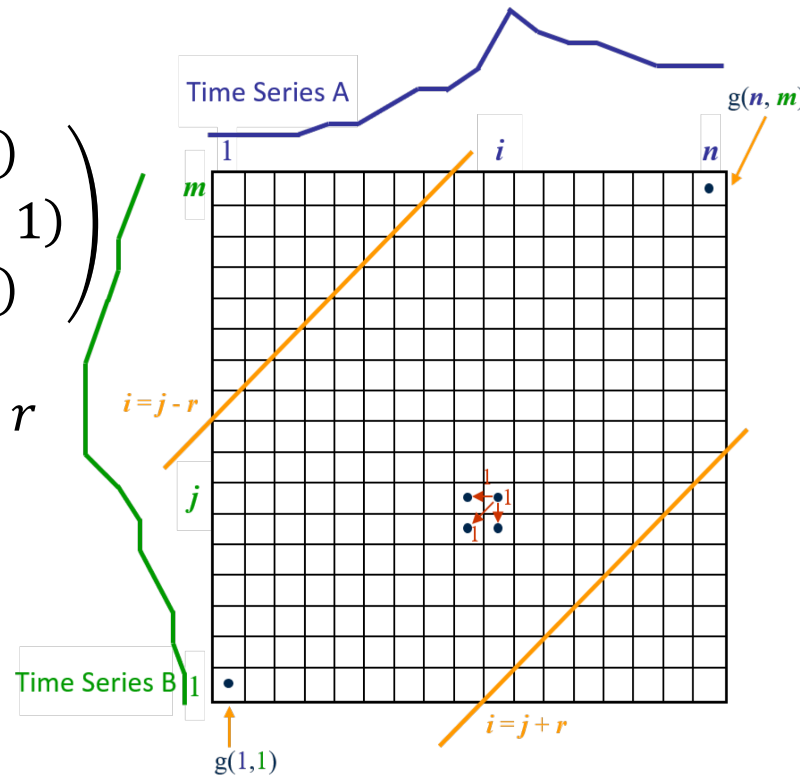
So we can use Dynamic Programming (DP) at following:

$$D(A,B) = \frac{1}{C}\min_{p}\left[\sum_{s=1}^{k} d(p_s).w_s\right]$$

DYNAMIC TIME WARPING

- Initial Condition: $g(1,1) = d(1,1)$
- DP-equation

$$g(i,j) = d(i,j) + \min\begin{pmatrix} g(i,j-1) \\ g(i-1,j-1) \\ g(i-1,j) \end{pmatrix}$$



- Warping window: $j - r < i < j + r$
- Time normalized distance:

$$D(A,B) = \frac{g(n,m)}{C}$$

DYNAMIC TIME WARPING

- Start with the calculation of

$$g\,(1,1)\;=\;d\,(1,1).$$

- Calculate the first row

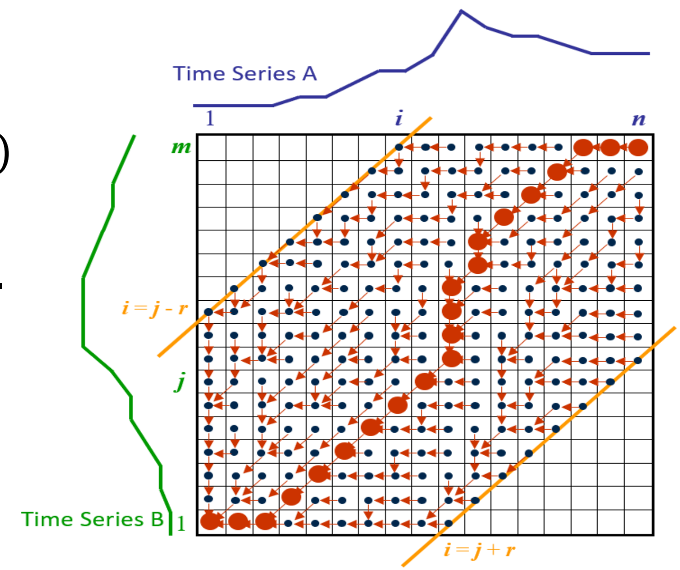
$$g\,(i\,,1)\;=\;g\,(i\,1,1)\;+\;d\,(i\,,1)$$

- Calculate the first column

$$g\,(1,j)\;=\;g\,(1,j)\;+\;d\,(1,j).$$

- Move to the second row
- $g(i,2) = \min\big(g(i,1), g(i1{,}1), g(i1{,}2)\big) + d(1{,}2)$

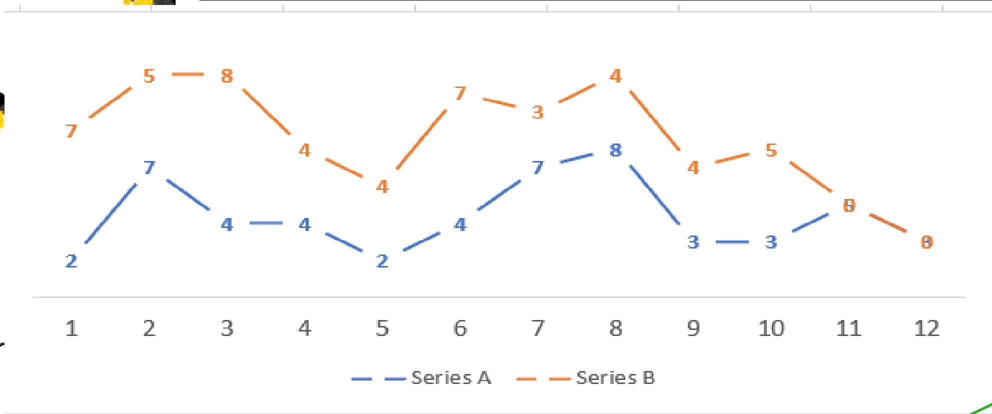
Book keep for each cell the index of this neighbouring cell, which contributes the minimum score (red arrows).



- Carry on from left to right and from bottom to top with the rest of the grid
- $g\,(i\,,j)\;=\;\min(g\,(i\,,j\,1), g\,(i\,1,j\,1), g\,(i\,1,j))\;+\;d\,(i\,,j).$
- Trace back the best path through the grid starting from $g\,(n,m)$ and moving towards $g\,(1,1)$ by following the red arrows.

DYNAMIC TIME WARPING

EXAMPLE



index

		1	2	3	4	5	6	7	8	9	10	11	12
		2	7	4	4	2	4	7	8	3	3	5	3
10	5						6	7	10	10	10	8	10
9	4					5	5	8	12	8	8	8	8
8	4				3	5	5	8	12	7	7	7	7
7	3			3	4	5	6	9	10	6	6	8	6
6	7		2	5	6	8	8	5	6	8	8	6	
5	4	2	5	3	3	5	5	8	10	4	4		
4	4	2	5	3	3	5	5	6	6	3			
3	8	5	3	5	5	8	8	3	2				
2	5	3	2	1	2	5	4	2					
1	7	5	0	3	3	5							

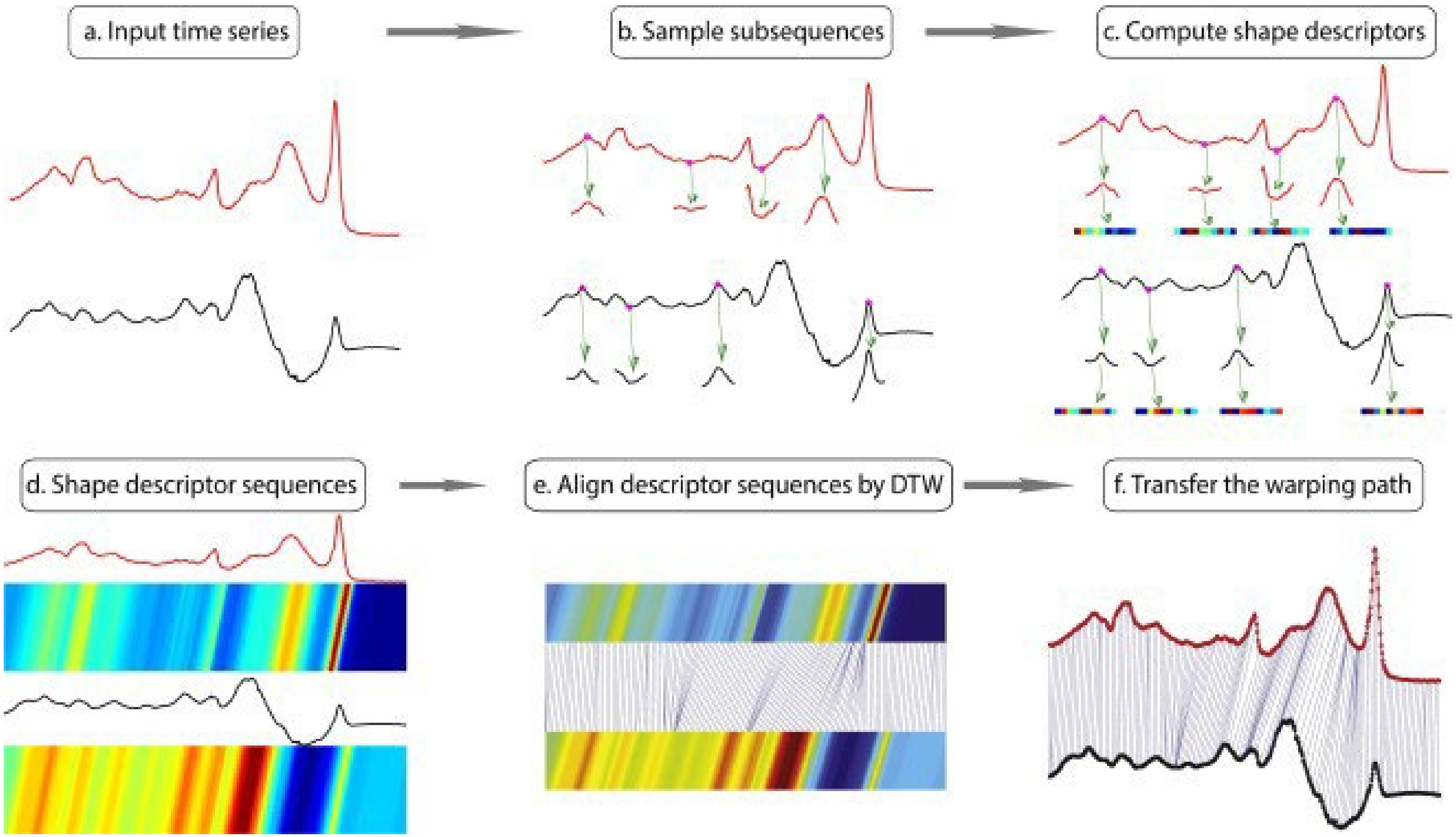
DYNAMIC TIME WARPING

EXAMPLE

		1	2	3	4	5	6	7	8	9	10	11	12
		2	7	4	4	2	4	7	8	3	3	5	3
10	5						6	7	10	10	10	8	**10**
9	4					5	5	8	12	8	8	**8**	8
8	4				3	5	5	8	12	7	**7**	7	7
7	3			3	4	5	6	9	10	**6**	6	8	6
6	7		2	5	6	8	8	**5**	**6**	8	8	6	
5	4	2	5	3	3	5	**5**	8	10	4	4		
4	4	2	5	**3**	**3**	**5**	5	6	6	3			
3	8	5	**3**	5	5	8	8	3	2				
2	5	3	**2**	1	2	5	4	2					
1	7	**5**	0	3	3	5							

Boundary

DTW VISUAL INSIGHTS



TIME SERIES AND FEATURE EXTRACTION