DAT320

Summary.

Time series and exploratory analysis

Time series data is **numeric** data in **successive order** along an axis (e.g. time). Therefore, permutations (changes in the order) affects the information contained.

While random sample $\{x_1, \cdots, x_n\}$ fullfills the i.i.d. properti (i.e. statistical independence and identical distribution), time series data is by default **not** independent nor identically distributed. Temperature measurements on two consecutive days are correlated across the year (not independent). Daily average changes between seasons (distinct distributions).

Time steps in a time series should be regular (equal) between all measurements. Irregular (unequal) time steps would need to be aggregated to become regular, for further analysis.

Trend and seasonality

> A time series is a sum (or product) of the three components
>
> - **Trend** A smooth, non-periodic function over time (change in mean value).
> - **Seasonality** A periodic, recurring function over time.
> - **Error** A time-independent random noise term.

Summary statistics

> **Mean & variance**
>
> $$\bar{x} = \frac{\sum_{t=t_{\min}}^{t_{\max}} x_t}{t_{\max} - t_{\min} + 1} \qquad \& \qquad \sigma^2 = \frac{\sum_{t=t_{\min}}^{t_{\max}} (x_t - \bar{x})^2}{t_{\max} - t_{\min}}$$
>
> **Median & inter-quartile range**
>
> $$q_{0.5}(\{x_t : t \in T\}) \qquad \& \qquad q_{0.75}(\{x_t : t \in T\}) - q_{0.25}(\{x_t : t \in T\})$$
>
> **Minimum & maximum**
>
> $$\min_{t \in [t_{\min}, t_{\max}]}(x_t) \qquad \& \qquad \max_{t \in [t_{\min}, t_{\max}]}(x_t)$$

Backshift operator (lag)

$$B(x_t) \to x_{t-1}$$

$$\begin{aligned} B^k(x_t) &= x_{t-k} \\ B^{-1}(x_t) &= x_{t+1} \\ B^{-k}(x_t) &= x_{t+k} \end{aligned}$$

Correlation

- **Autocorrelation** The correlation between a time series and **its own** lagged version.
- **Cross-correlation** The correlation between a time series and the lagged version of **another** series.

Stationarity

A time series is stationary if it fulfills the following conditions:

i. Constant mean μ over time.

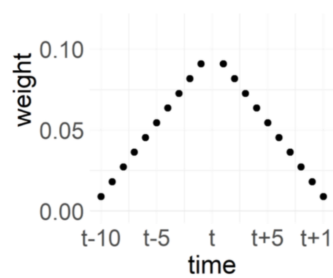
ii. Constant variance σ over time.

iii. Constant auto-correlation across all parts of the time series.

Missing value imputation

Linearly weighted moving average

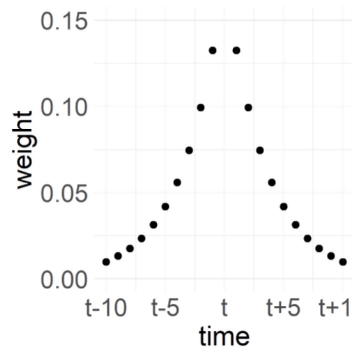
$$w_i = \begin{cases} \frac{i}{\ell(\ell+1)} & \text{if } i \geq \ell \\ \frac{2\ell-i-1}{\ell(\ell+1)} & \text{if } i < \ell \end{cases} \qquad \text{for } i = 1, \cdots, 2\ell$$



Exponentially weighted moving average

$$w_i = \begin{cases} C \times (1-\alpha)^{\ell - i} & \text{if } i \geq \ell \\ C \times (1-\alpha)^{i - \ell - i} & \text{if } i < \ell \end{cases} \qquad \text{for } i = 1, \cdots, 2\ell$$

$$C = \frac{\alpha}{2 - 2(1-\alpha)^{\ell}}$$



Linearly interpolation

$$x_t = \left(\frac{s_2 - t}{s_2 - s_1} x_{s_1} + \frac{t - s_1}{s_2 - s_1} x_{s_2} \right)$$

Transformations

- Same scale → standardize
- Remove skewness → power transform
- Remove trends → difference
- Remove noise → smoothing
- Missing values → imputation

Standardization & normalization

(b) **Standardization** is to transform the data to 0 mean and 1 standard deviation:

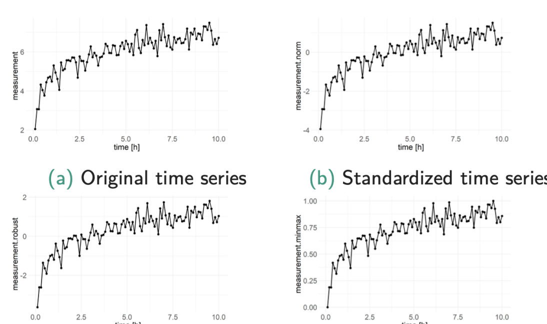
$$x'_t = \frac{x_t - \bar{x}}{\sigma}$$

(c) **Robust standardization** is to scale using median and iqr instead:

$$x'_t = \frac{x_t - \text{median}(x)}{\text{iqr}(x)}$$

(d) **Min-max normalization** is to scale all values to be in the range $[0, 1]$:

$$x'_t = \frac{x_t - \min(x)}{\max(x) - \min(x)}$$



(a) Original time series
(b) Standardized time series
(c) Normalized time series (robust)
(d) Normalized time series (min-max)

Power transform

Logatrithm

> Transforms skewed data in order to obtain a Gaussian-like distribution. $x'_t = \log(x_t)$.
>
> This works well for data that is approximately log-normal distributed.

Box-Cox transformation

> A generalization of the log-transform.
>
> $$x'_t = \begin{cases} \frac{x_t^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \\ \log(x_t) \end{cases}$$

STL decomposition (Seasonal Trend Remainder)

> Seasonal and trend decomposition using LOESS (Local RegrESSion).
>
> Decomposition of data into three components; seasonal, trend and remainder: $x_t = s_t + \tau_t + r_t$
>
> s_t Seasonality
>
> τ_t Trend
>
> r_t Remainder (residual, noise)
>
> **PARAMETERS**
>
> - **s.window** Seasonality window size.
> - **t.window** Trend window size.

■

Forecasting

Baseline models

> Four baseline models should be evaluated as minimum benchmarks for any more complex forecasting models.
>
> - **Average method**
>
>   Estimate future points as the average of the history.
>
> $$\hat{x}_{t+h} = \frac{1}{t}\sum_{s=1}^{t} x_s$$
>
> - **Drift method**
>
>   Estimate future points as last observed value plus drift (trend).
>
> $$\hat{x}_{t+h} = x_t + h\left(\frac{x_t - x_1}{t-1}\right)$$
>
> - **Naïve method**
>
>   Estimate future points as last observed value.
>
> $$\hat{x}_{t+h} = x_t$$

- **Seasonal naïve method**

  Estimate future points as same value one period ago.

$$\hat{x}_{t+h} = x_{t+h-p(k+1)}$$

Exponential smoothing

- Naïve method: only the most recent observation x_t is relevant.
- Average method: all historical observations $x_1, \cdots, x_t$ are equally relevant.

Methods

- **Simple exponential smoothing (SES)**
  Parameter $\alpha \in [0, 1]$ determines the strength of smoothing, i.e., the rate of weight decay.

$$\begin{aligned} \hat{x}_{t+h} &= \ell_t && \text{forecast} \\ \ell_t &= \alpha x_t + (1-\alpha)\ell_{t-1} && \text{smoothing} \end{aligned}$$

- **Exponential smoothing with trend (Holt's method)**
  Extension of SES by adding a trend component.

$$\begin{aligned} \hat{x}_{t+h} &= \ell_t + hb_t && \text{forecast} \\ \ell_t &= \alpha x_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) && \text{smoothing} \\ b_t &= \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1} && \text{trend} \end{aligned}$$

- **Exponential smoothing with damped trend**
  Introduces a damping parameter ϕ.

$$\begin{aligned} \phi &= 1 && \text{Holt's method (no damping)} \\ \phi &= 0 && \text{SES (no trend)} \end{aligned}$$

- **ETS Exponential smoothing with seasonality**
  Adds an additional term for the seasonality, with a new parameter γ. Error, trend and seasonality.

In R, this is an **ets** model, with different *model*-parameters. "MMM" for multiplicative error, trend and seasonality, and so on.

```
additive <- ets(data, model = "AAA")

multiplicativeHoltWinters <- ets(data, model = "MMM")
```

ARIMA (auto-regressive integrated moving-average)

Stationarity

> Stationarity can be obtained by differencing the data.
>
> $$\begin{aligned} \text{differencing} \quad & D(x_t) = x_t - B(x_t) = (1 - B)(x_t) \\ \text{seasonal differencing} \quad & D_S(x_t) = x_t - B^p(x_t) = (1 - B^p)(x_t) \end{aligned}$$
>
> To check for trends, a KPSS (Kwiatkowski-Phillips-Schmidt-Shin) test can be performed. If the p-value is below 0.05 there **is** trend, and if the p-value is above 0.05 there **is no** trend.
>
> ```
>             kpss.test(data1)  # < 0.05, i.e. trend
>             kpss.test(data2)  # > 0.05, i.e. no trend
> ```

AR(k) Auto-regressive model

> Uses the current time point as a target, and previous time points as predictors.
>
> $$\begin{aligned} x_t &= \varphi_0 + \varphi_1 B(x_t) + \varphi_2 B^2(x_t) + \cdots + \varphi_k B^k(x_t) + \varepsilon_t \\ &= \varphi_0 + \varphi_1 x_{t-1} + \varphi_2 x_{t-2} + \cdots + \varphi_k x_{t-k} + \varepsilon_t \end{aligned}$$
>
> where φ_0 is the global mean value, σ the variance of ε_t defines the scale and $\varphi_1, \cdots, \varphi_k$ the temporal pattern.
> ALL $\varphi_1, \cdots, \varphi_k$ must be between $[0, 1]$, and their sum $\sum_{i=1}^{k} \varphi_i z^i < 1$, where $|z| > 1$.

MA(q) Moving-average model

> Uses the current time point as a target, and previous **errors** as predictors.
>
> $$x_t = \theta_0 + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

ARMA(k,q) Auto-regressive moving-average model

> Combining an AR(k) and an MA(q) model, we get an ARMA(k,q) model.
>
> $$\begin{aligned} x_t = \varphi_0 &+ \varphi_1 x_{t-1} + \varphi_2 x_{t-2} + \cdots + \varphi_k x_{t-k} \\ &+ \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta \varepsilon_{t-q} + \varepsilon_t \end{aligned}$$
>
> - Any AR(k) model can be represented by an MA(∞) model.
> - Any MA(q) model can be represented by an AR(∞) model.

ARIMA(k,d,q) Auto-regressive integrated moving-average model

> Includes a differencing term.
>
> $$\begin{aligned} D^d(x_t) = \varphi_0 &+ \varphi_1 D^d(x_{t-1}) + \cdots + \varphi_k D^d(x_{t-k}) \\ &+ \theta_1 \varepsilon_{t-1} + \cdots + \theta \varepsilon_{t-q} + \varepsilon_t \end{aligned}$$

such that the final model is:

$$D^d(x_t) = \phi_0 + \phi_1 D^d(x_{t-1}) + \cdots + \phi_k D^d(x_{t-k}) \\ + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

$$(1-B)^d(x_t) = \phi_0 + \phi_1 (1-B)^d(B(x_t)) + \cdots + \phi_k (1-B)^d(B^k(x_t)) \\ + \theta_1 B(\varepsilon_t) + \cdots + \theta_q B^q(\varepsilon_t) + \varepsilon_t$$

$$(1 - \phi_1 B - \cdots - \phi_k B^k)(1-B)^d x_t = \phi_0 + (1 + \theta_1 B + \cdots + \theta_q B^q)\varepsilon_t$$

Where the different parts are differencing, auto-regressive and moving-average.

Parameter estimation

d	Can be estimated using KPSS test and differencing until it's p-value is greater than 0.05.
k, q	Can be estimated by (partial) autocorrelation and/or minimising AIC.

if ACF is exponentially decaying or sinusoidal:

- use AR(k) and determine k as maximum significant spike in PACF.

if PACF is exponentially decaying or sinusoidal:

- use MA(k) and determine k as maximum significant spike in ACF.

ETS vs. ARIMA

ETS decomposes the original data into trend, seasonality and error. These three components are then modelled to predict future points. Whereas an ARMA-model's MA uses previous prediction errors.

- Additive exponential smoothing is a special case of SARIMA.
- Non-additive exponential smoothing models are NOT covered by ARIMA.
- Not all ARIMA variants are covered by exponential smoothing.

SARIMA (seasonal ARIMA)

Allows both seasonal and non-seasonal components.

$$SARIMA(k,d,q)(K,D,Q)_p$$

$SAR(K)$	seasonal AR
$SMA(Q)$	seasonal MA

where k, d, q are the ARIMA parameters, K, D, Q the corresponding seasonal terms and p the seasonal period.

Terms

$$SARIMA(1,1,1)(1,1,1)_p \quad \Rightarrow \quad (1-\varphi_1 B)(1-\Phi_1 B^p)(1-B)(1-B^p)x_t = (1+\theta_1 B)(1+\Theta_1 B^p)\varepsilon_t$$

AR(k)	$(1-\varphi_1 B-\cdots-\varphi_k B^k)$
SAR(K)	$(1-\Phi_1 B^p-\cdots-\Phi_K B^{Kp})$
Differencing	$(1-B)^d$
Seasonal differencing	$(1-B^p)^D$
MA(q)	$(1+\theta_1 B+\cdots+\theta_q B^q)$
SMA(Q)	$(1+\Theta_1 B^p+\cdots+\Theta_Q B^{Qp})$

Parameter estimation

d Can be estimated using KPSS test and differencing until p-value > 0.05.

D Can be estimated using a HEGY test and seasonal differencing until p-value < 0.05.

$k,\ q$ Estimated through ACF and/or PACF. Or AIC.

if ACF is exponentially decaying or sinusoidal:

- use AR(k) and determine k as maximum significant spike in PACF.

if PACF is exponentially decaying or sinusoidal:

- use MA(k) and determine k as maximum significant spike in ACF.

$K,\ Q$ Estimate SAR or SMA parameters from seasonal spikes in ACF and PACF.

Statistical tests

- **KPSS**
  Trend test. p-value > 0.05 means the data is stationary.
- **ADF**
  Trend test. p-value < 0.05 means the data is stationary.
- **HEGY**
  Seasonality test. p-value < 0.05 means the data has no seasonality.

ARIMAX

Limited interpretability of β (compared to linear regression model).

When solving for the parameters of an ARIMAX-model, one can solve it in different ways – therefore leading to limited interpretability of coefficients.

Stochastic processes

A stochastic process is a series of steps or events where there's some element of randomness or chance involved, making it unpredictable and different each time. (Like rolling a dice.)

Stationarity

> A stochastic process is **strictly stationary** if all its distributions are equal over time.
>
> A stochastic process is **weakly stationary** if its expected value is constant over time and/or its autocovariance is constant over all time.
>
> Strict stationarity $\Rightarrow$ weak stationarity.

Markov property

> The Markov property states that the future is independent of the past given the present.
>
> $$P(X_t|X_{t-1}, X_{t-2}, \cdots, X_1) = P(X_t|X_{t-1})$$
>
> **Example violations**
>
> > $AR(2) = X_t = \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + \varepsilon_t$
> >
> > Current value (X_t) is dependent on:
> >
> > - X_{t-1}
> > - X_{t-2} $\Leftarrow$ Violation of Markov property.
> > - ε_t
> >
> > Likewise, $SAR(1)_p$ is a violation of the Markov property (whereas $AR(1)$ is not).

HMM — Hidden Markov models

Markov chains

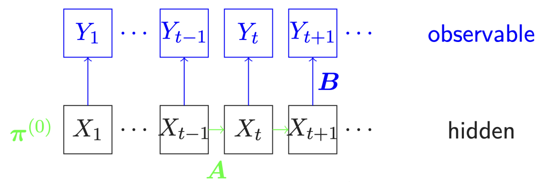
> $$\begin{aligned} P(X_{t+1}|X_t = x_t) &\quad \text{1-step transition probabilities} \\ P(X_0 = x_0) = \pi_0 &\quad \text{initial state distribution} \end{aligned}$$
>
> A Markov chain is **time-homogeneous** if its one-step transition probabilities are independent of time.

Markov chains for categorical variables

> For three categorical variables, e.g. $\{s, r, c\}$ the initial states and transitional probabilities will be:
>
> $$\pi^{(0)} = \begin{bmatrix} \pi_s^{(0)} \\ \pi_r^{(0)} \\ \pi_c^{(0)} \end{bmatrix} \qquad A = \begin{bmatrix} A_{s,s} & A_{s,r} & A_{s,c} \\ A_{r,s} & A_{r,r} & A_{c,c} \\ A_{c,s} & A_{c,r} & A_{c,c} \end{bmatrix}$$

- π_0 must sum to 1.
- Each row of A must sum to 1.

Hidden Markov models

> If the **hidden** (latent) states are $\{s, r, c\}$ and the **observable** states are $\{u, n\}$, the *emission* matrix B is the probabilistic relation between the latent and observable states
>
> $$B = \begin{bmatrix} B_{s,u} & B_{s,n} \\ B_{r,u} & B_{r,n} \\ B_{c,u} & B_{c,n} \end{bmatrix}$$
>
> 
>

Forward algorithm

> **INPUT** $\quad A, B, \pi^{(0)}$
>
> **OUTPUT** $\quad \alpha_i^{(t)} = P(Y_1 = y_1, \cdots Y_t = y_t, X_t = i | A, B, \pi^{(0)})$
>
> $\alpha_i^{(1)} = \pi_i^{(0)} B_{i,y_1}$
>
> **FOR** $t = 1, \cdots, t_{\max} - 1$:
>
> $\quad \alpha_i^{(t+1)} = B_{i,y_{t+1}} \sum_j \alpha_j^{(t)} A_{j,i}$
>
> **RETURN** $\alpha_i^{(t)}$

Backward algorithm

> **INPUT** $\quad A, B, \pi^{(0)}$
>
> **OUTPUT** $\quad \beta_i^{(t)} = P(Y_{t+1} = y_{t+1}, \cdots Y_{t_{\max}} = y_{t_{\max}}, X_t = i | A, B, \pi^{(0)})$
>
> $\beta_i^{t_{\max}} = 1$
>
> **FOR** $t = t_{\max} - 1, \cdots, 1$:
>
> $\quad \beta_i^{(t)} = \sum_j B_{i,y_{t+1}} \beta_j^{(t+1)} A_{i,j}$
>
> **RETURN** $\beta_i^{(t)}$

Training and predictions

> For training, the Baum-Welch algorithm is used to estimate the parameters $A, B, \pi^{(0)}$ based on observable states, and the Viterbi is used to predict underlying states.

Classification and clustering

Distance-based

- $d(x, x) = 0$
- $d(x, y) > 0$ for all $y \neq x$
- $d(x, y) = d(y, x)$
- $d(x, y) + d(y, z) \leq d(x, z)$

Minkowski distance

$$d_p(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$$

Euclidean for $p = 2$. Manhattan for $p = 1$.

These metrics are sensitive to standard transformations and outliers, and do not take temporal order (neighbours) into account.

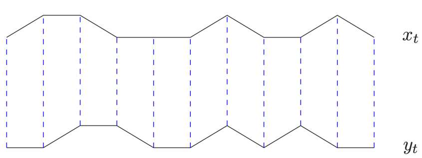
DTW

Dynamic time warping

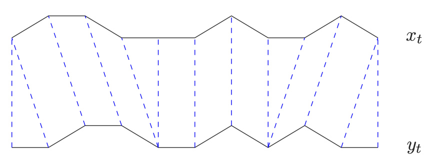
Takes neighbours into account when computing the distance between two series.

Invariant to shifting and scaling of the time-axis.

Minkowski distance



DTW distance



Correlation-based

Robust to scaling, but sensitive to shifts/scaling along the time-axis.

$$d_{\text{cor}}(x_T, y_T) = \text{cor}(x_T, y_T)$$

ACF-based

Robust to scaling **and** shifts along the time-axis. Cannot evaluate patterns.

Model-based

ARIMA

Train an ARIMA model with optimized parameters (AIC) for each time series. Compare the model parameters.

Feature-based

- Global statistics.
  - Distribution
  - Minimum, maximum
  - Number of local minima, maxima
  - Number of crossing the median
- Statistical properties.
  - Heterogenity
  - Nonlinearity
  - KPSS test
- Autocorrelation.
  - (P)ACF coefficient of original
  - (P)ACF coefficients of differenced time series
  - First minimum/zero-crossing of (P)ACF
- Model (meta)parameters.
- Frequencies.
  - Fourier-transform
  - Wavelet-transform
- Patterns.

Metrics

Accuracy	$\frac{TP+TN}{TP+FP+FN+TN}$
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
F1	$\frac{2TP}{2TP+FP+FN} = 2\frac{\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}$
MCC	$\frac{TP\times TN-FP\times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

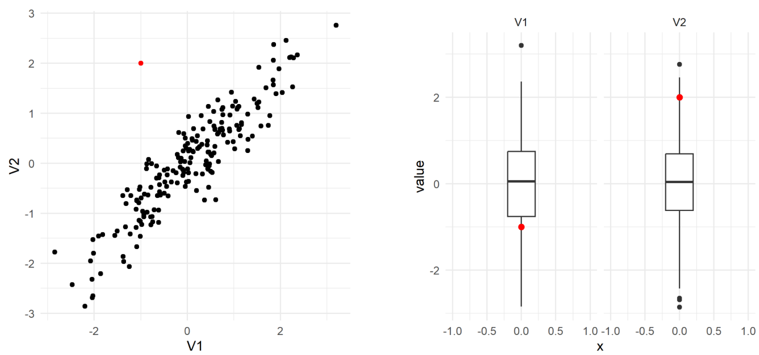
Outlier detection

OUTLIERS

- Point outliers (singular)
  Single extreme points.
- Collective outliers (subsequences)
  Collection of extreme points.

- Contextual outliers
  Cannot be seen when looking at the distribution alone, but by comparing to "ordinary" behaviour.

Univariate and multivariate outliers

> Univariate outliers are measurements outside the ordinary.
>
> Multivariate outliers only make sense when looking across variables. These outliers do not stand out when looking at only $V1$ or $V2$, but when both $V1$ and $V2$ is taken into account.
>
> 
>
>
> Here, B would be a multivariate outlier.
>
> #### z-score
>
> > If (univariate) data follows a Gaussian distribution, it can be scaled to a standard-Gaussian distribution.
> >
> > $$Z = \frac{X - \hat{\mu}}{\hat{\sigma}}$$
> >
> > "z-scores" $=$ standardized X
> >
> > How likely is a value under the distribution?
> >
> > $$P(Z \leq z) = \Phi(z)$$
> >
> > If $|z| \leq \textit{threshold}$ with $\textit{threshold} \in [0, 1]$, z is likely under that given distribution.

Mahalanobis distance

> For multivariate data, $|z|$ can be interpreted as a distance from the distribution mean.
>
> $$d_M(x, y) = \sqrt{(x-y)^T \sum{}^{-1}(x-y)}$$
>
> which is the "weighted" Euclidean distance by inverse covariances.
>
> Outliers: $d_M(x, \mu) > \textit{threshold}$

Temporal window

> For time-series data, apply a temporal window (e.g. 5 time steps) and check for "local" outliers. Repeat this for whole dataset.

Model-based

> **ESTIMATION-BASED**
>
> - Model is trained on all values. Outliers produce high residuals.
>
> **PREDICTION-BASED**
>
> - Model is trained only on history. Outliers produce inaccurate predictions.

Discord-detection

> Aims at determining "most unusual subsequence" (discord). Time-series subsequences are determined by sliding window, and compared to each other. Delivers **only one** "most unusual subsequence".
>
> ### Comparison with reference sequence
>
> > In the same way, each window is compared to a **reference** (and not each other).