robfilter: An R-Package for Robust Time Series Filters

Karen Schettlinger, Roland Fried, Ursula Gather

technische universität dortmund

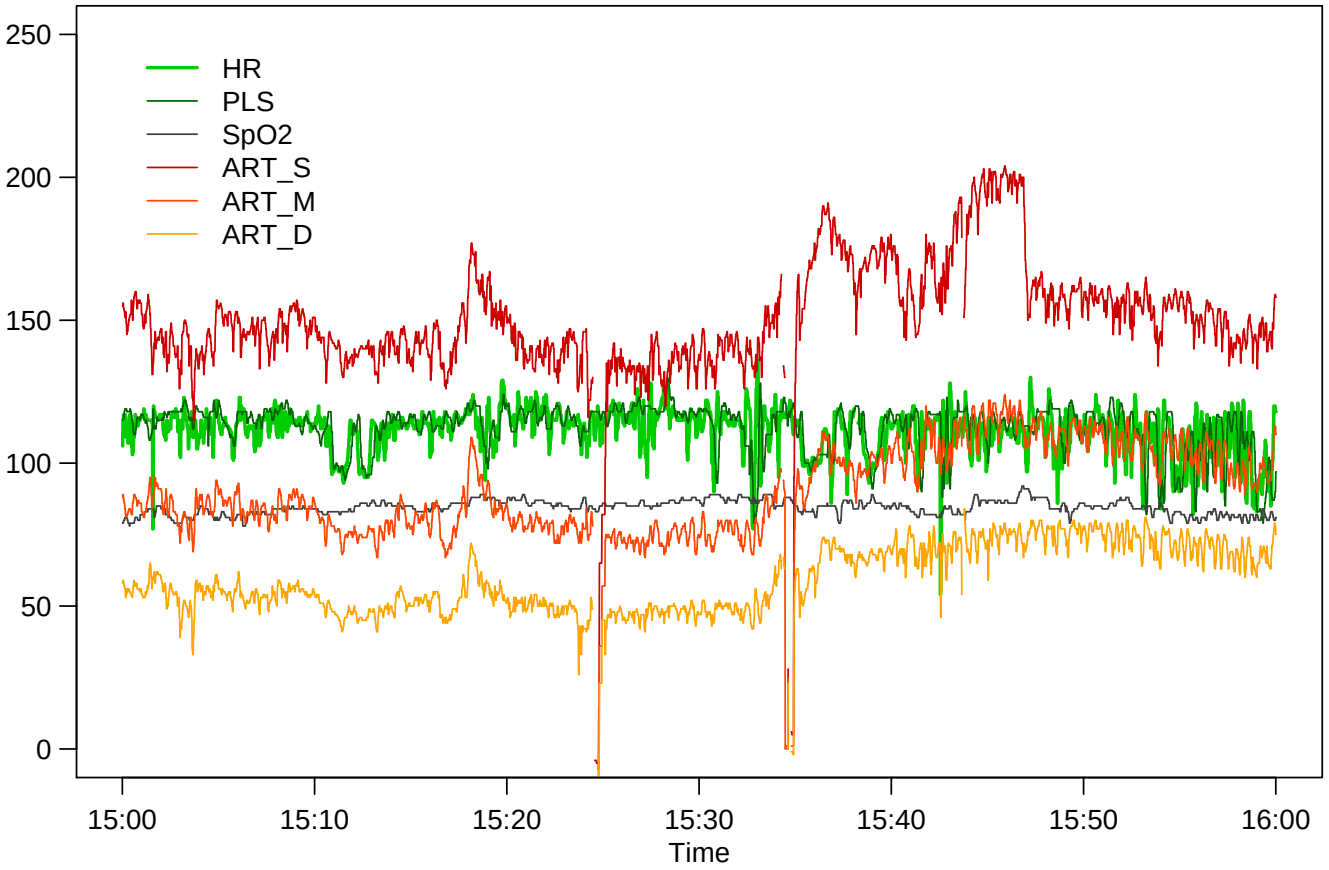
Fakultät Statistik

Technische Universität Dortmund

SFB 475: Reduction of Complexity in Multivariate Data Structures

Motivation

Multivariate physiological time series



➡ Filter signals to improve ICU monitoring systems

Contents of `robfilter` Version 2.0

7 time series filters

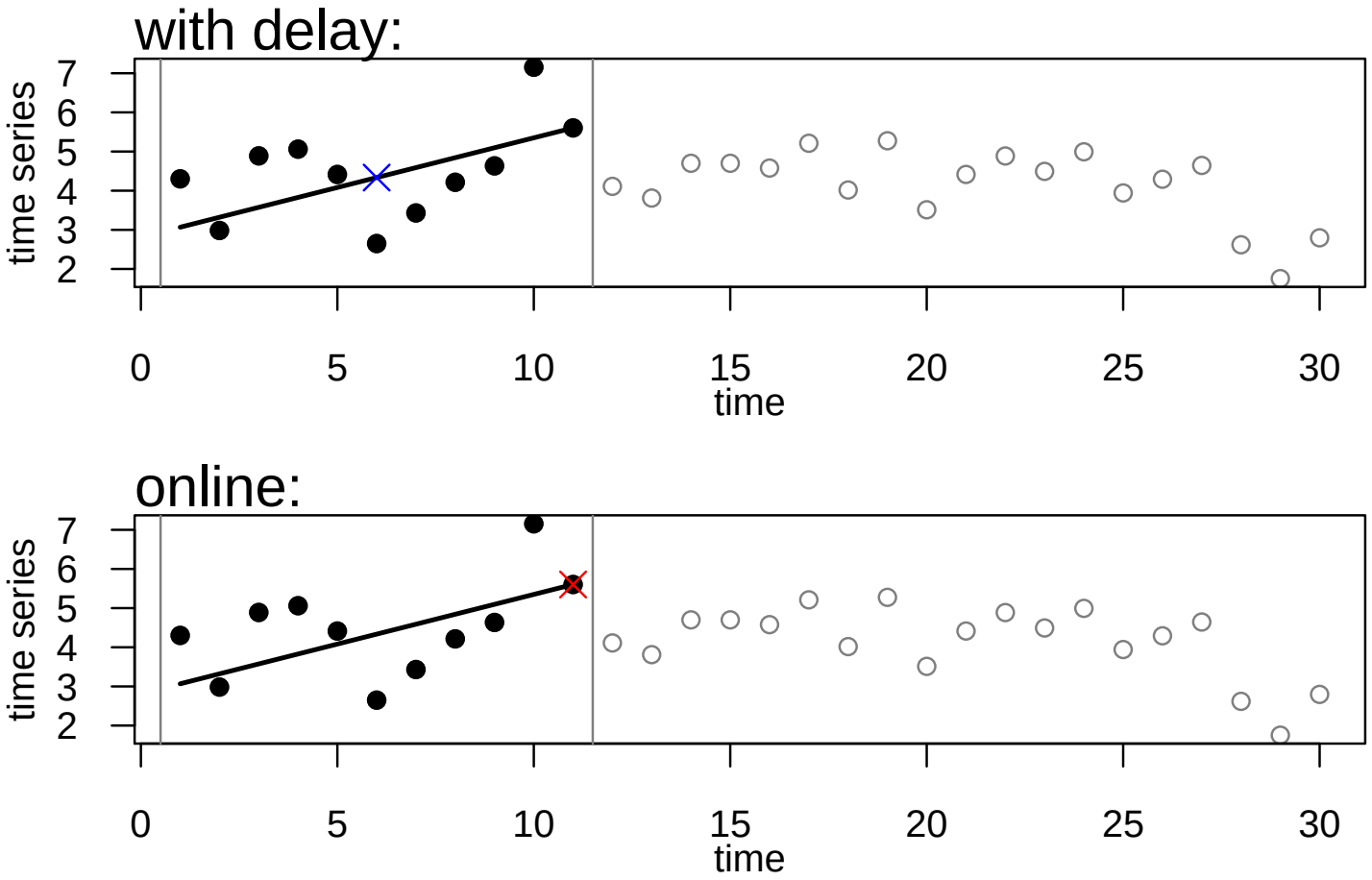
`robreg.filter`	Simple regression filters
`hybrid.filter`	Median and repeated median hybrid filters
`dw.filter`	Two-step location-/regression-based filters
`wrm.filter`	Weighted repeated median filters
`robust.filter`	Regression filters with additional rules (outlier & level shift detection)
`adore.filter`	Adaptive repeated median filters
`madore.filter`	Multivariate adaptive repeated median filters

1 univariate smoothing method

`wrm.smooth`	Weighted repeated median smoothing

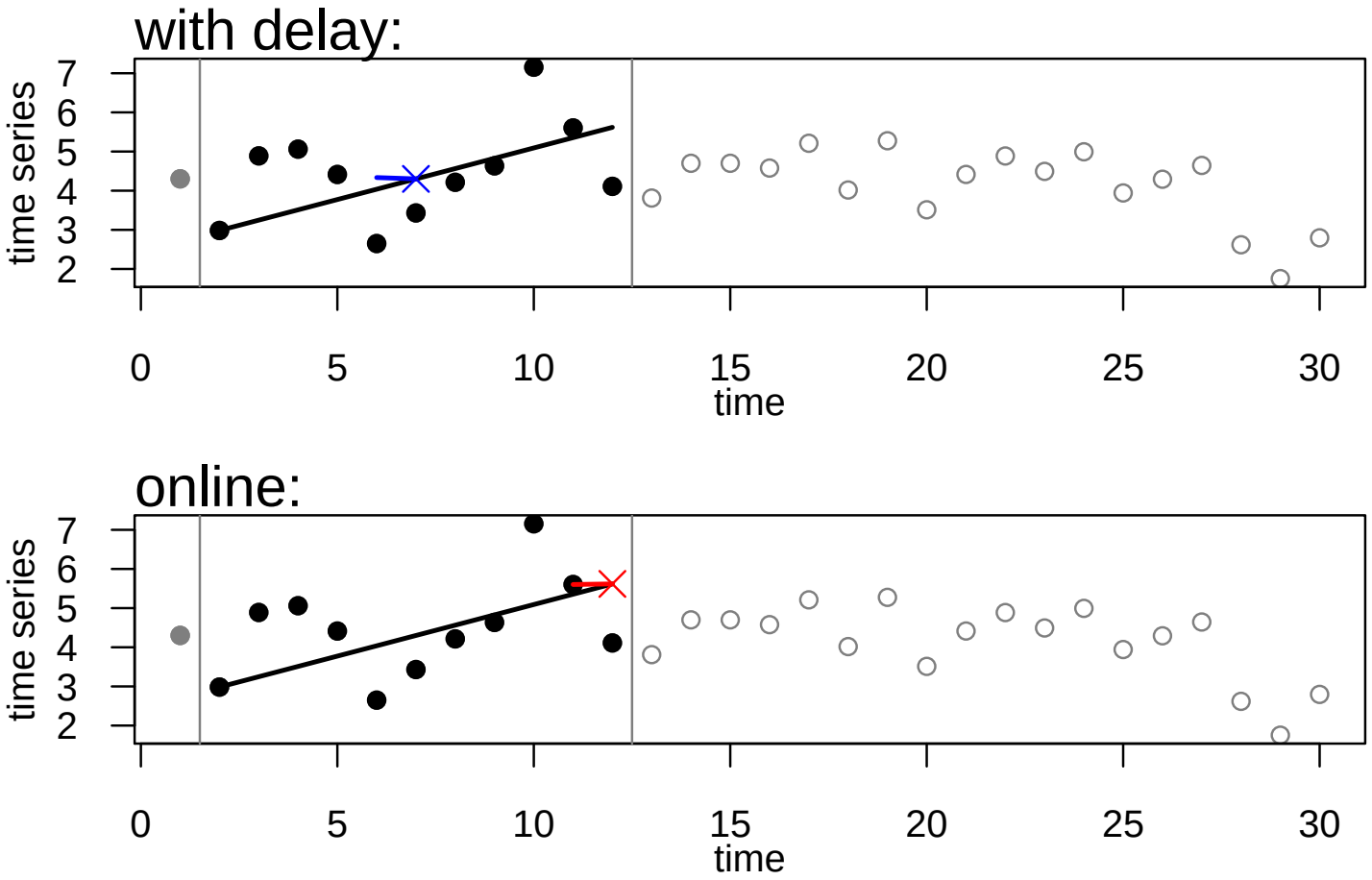
`robreg.filter` – Illustration

Robust **reg**ression in a moving time window:



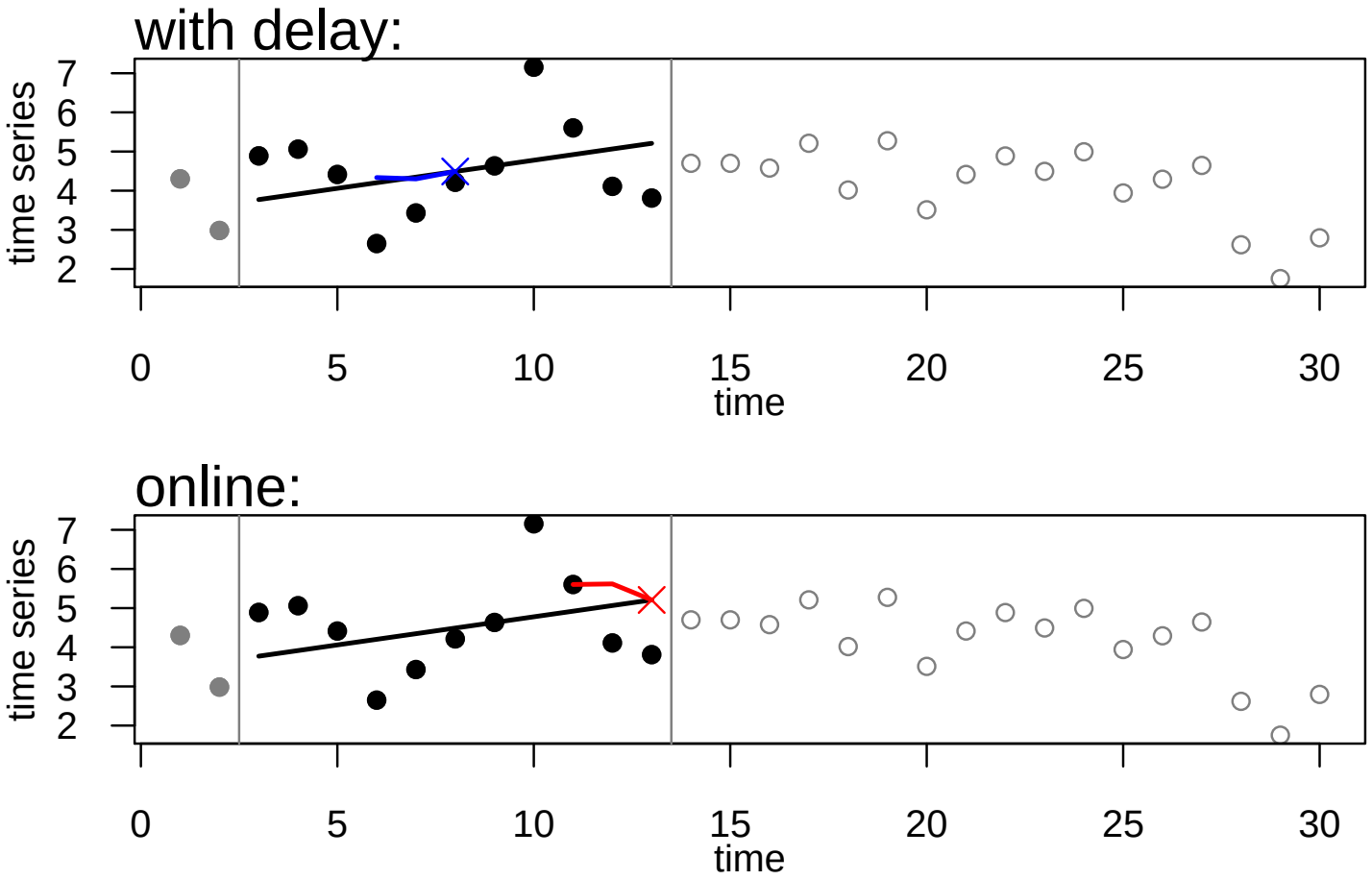
`robreg.filter` – Illustration

Robust **reg**ression in a moving time window:



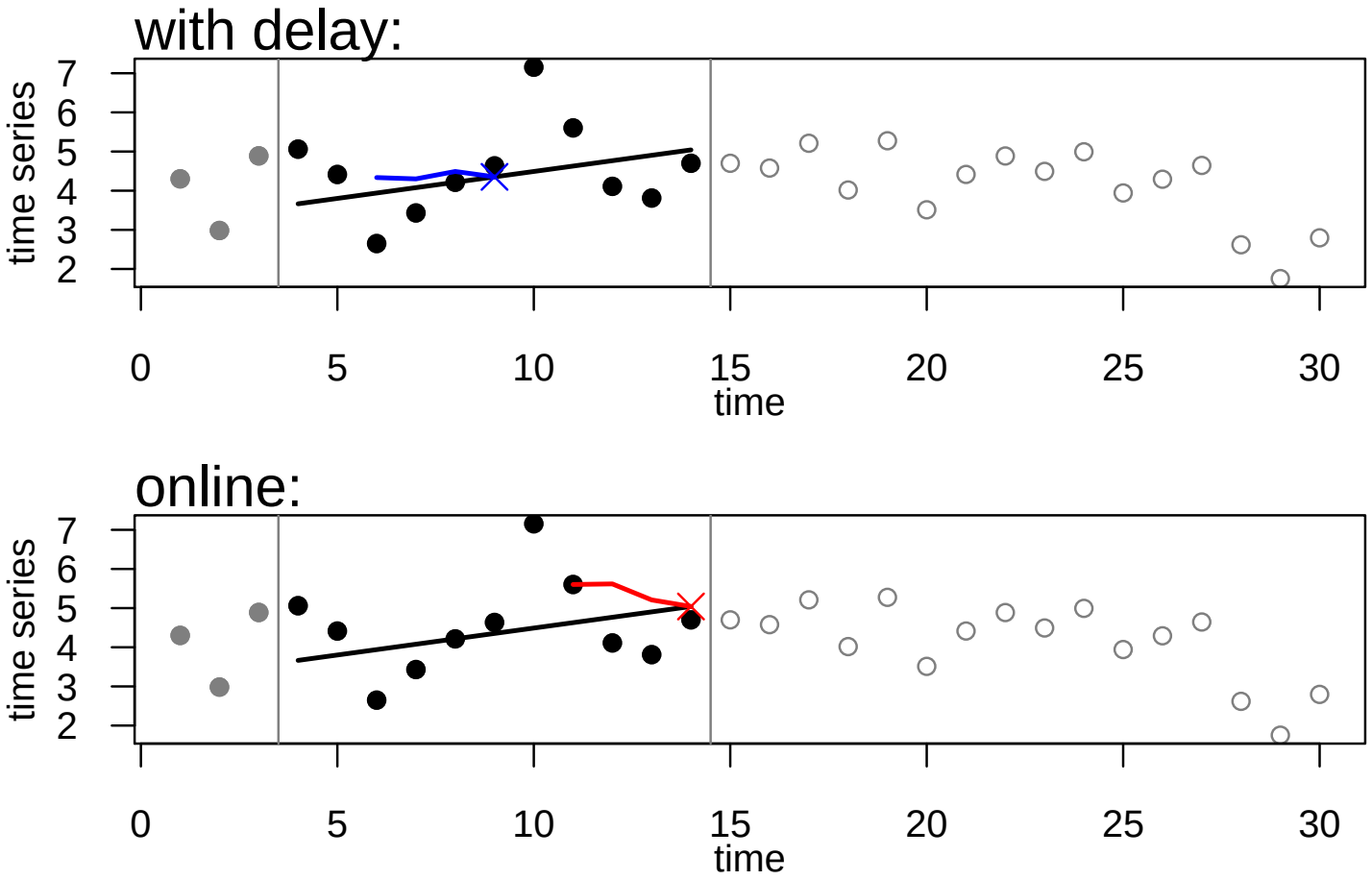
`robreg.filter` – Illustration

Robust **reg**ression in a moving time window:



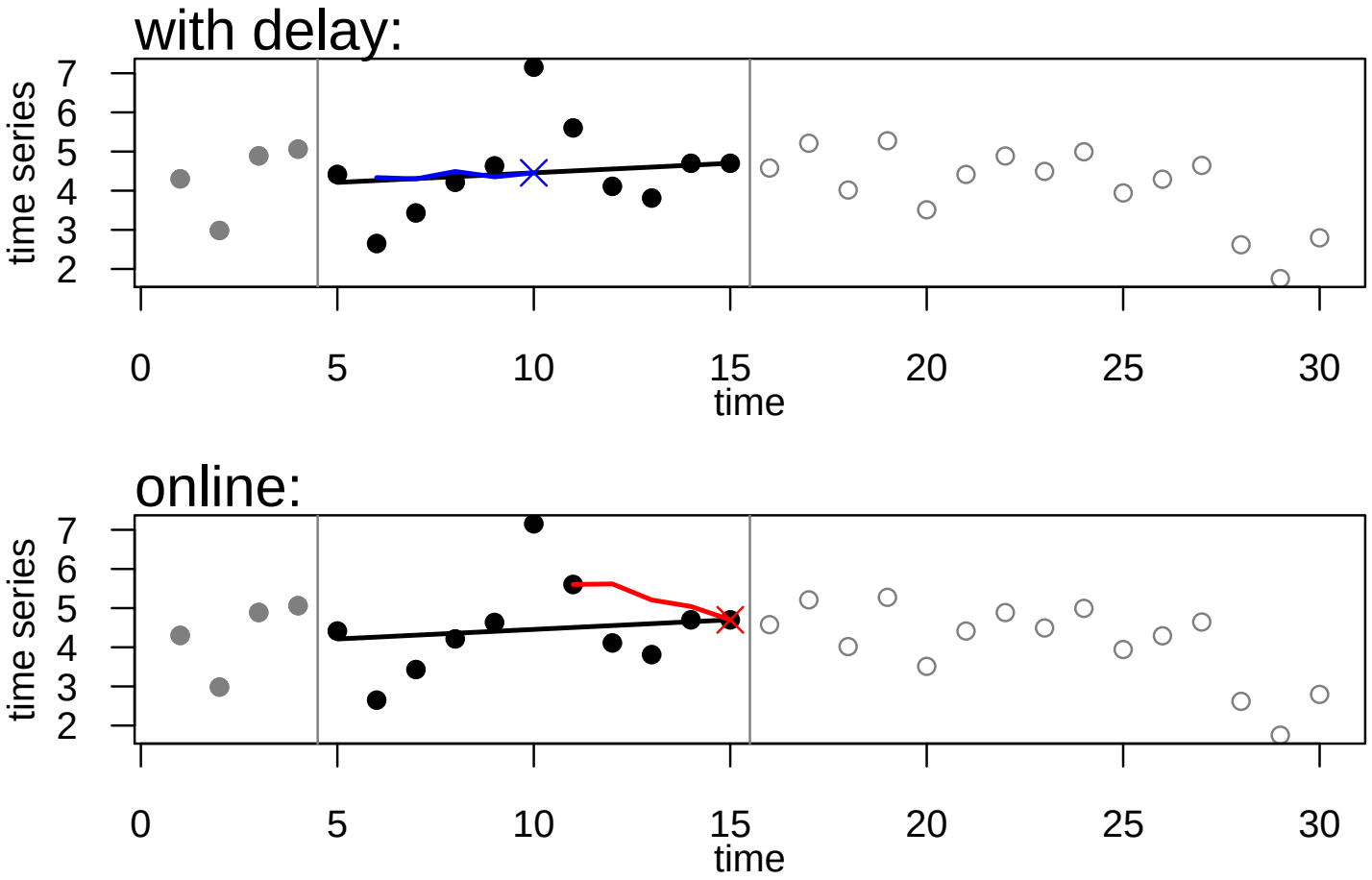
`robreg.filter` – Illustration

Robust **reg**ression in a moving time window:



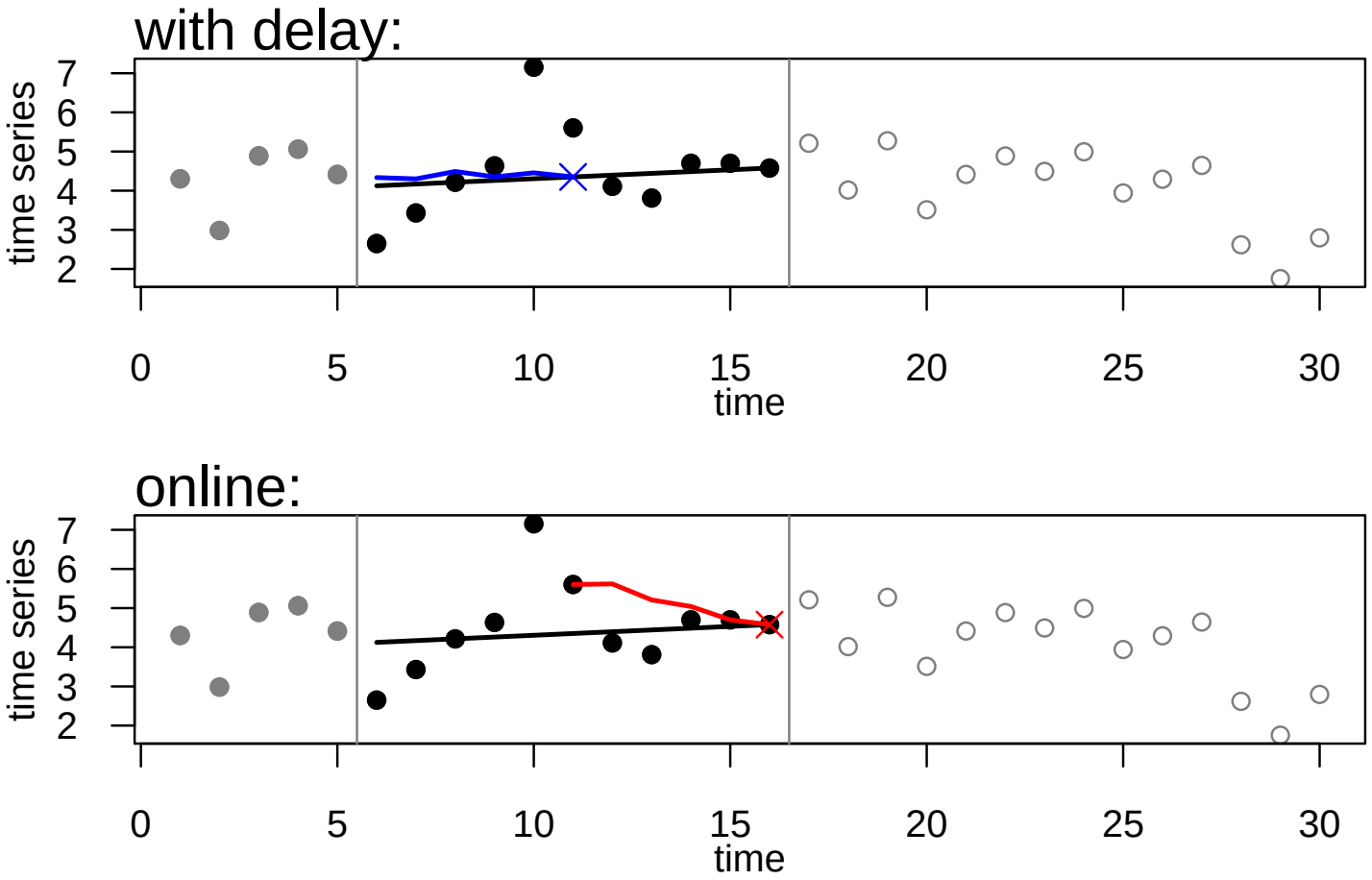
`robreg.filter` – Illustration

Robust **reg**ression in a moving time window:



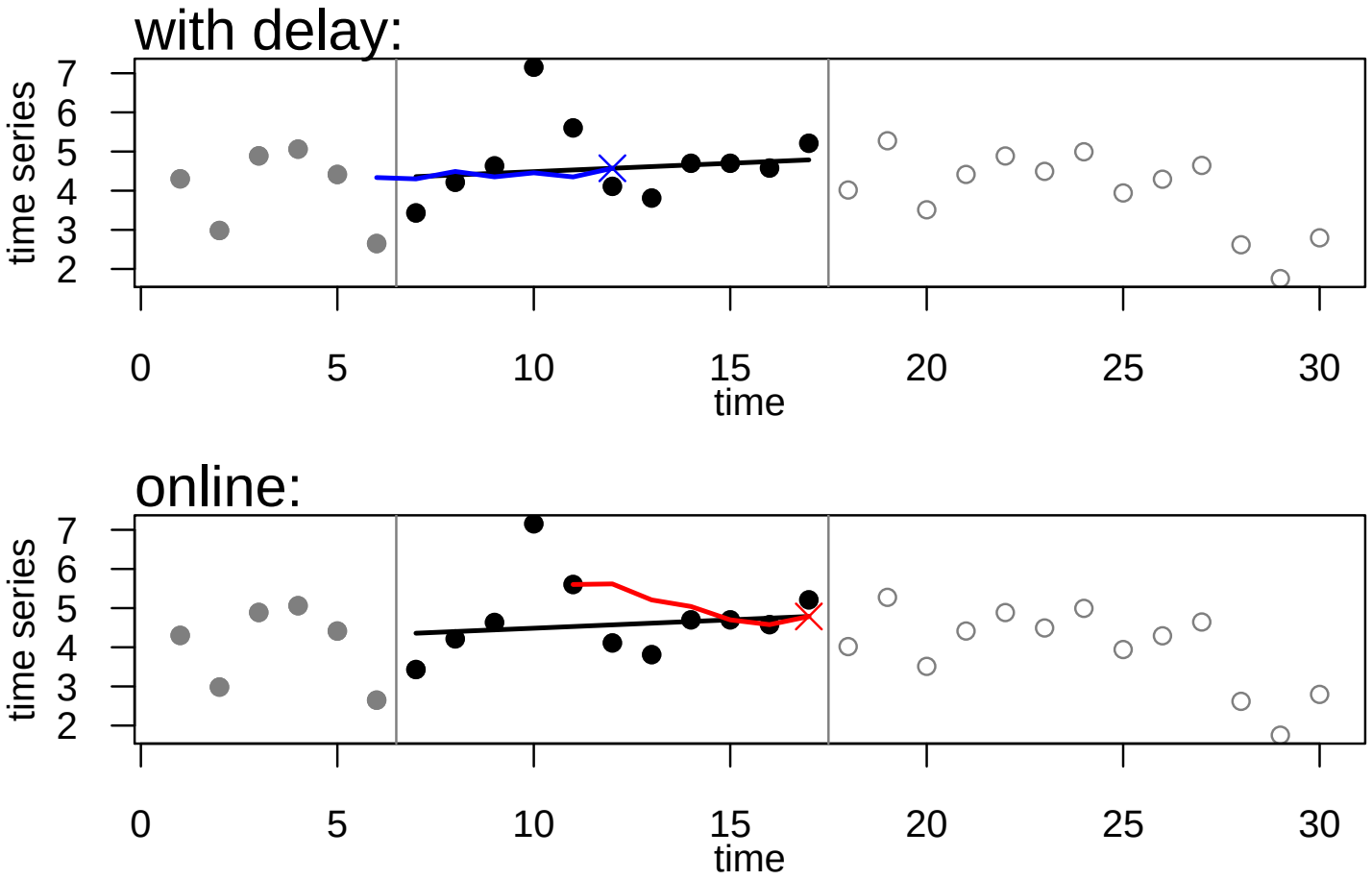
`robreg.filter` – Illustration

Robust **reg**ression in a moving time window:



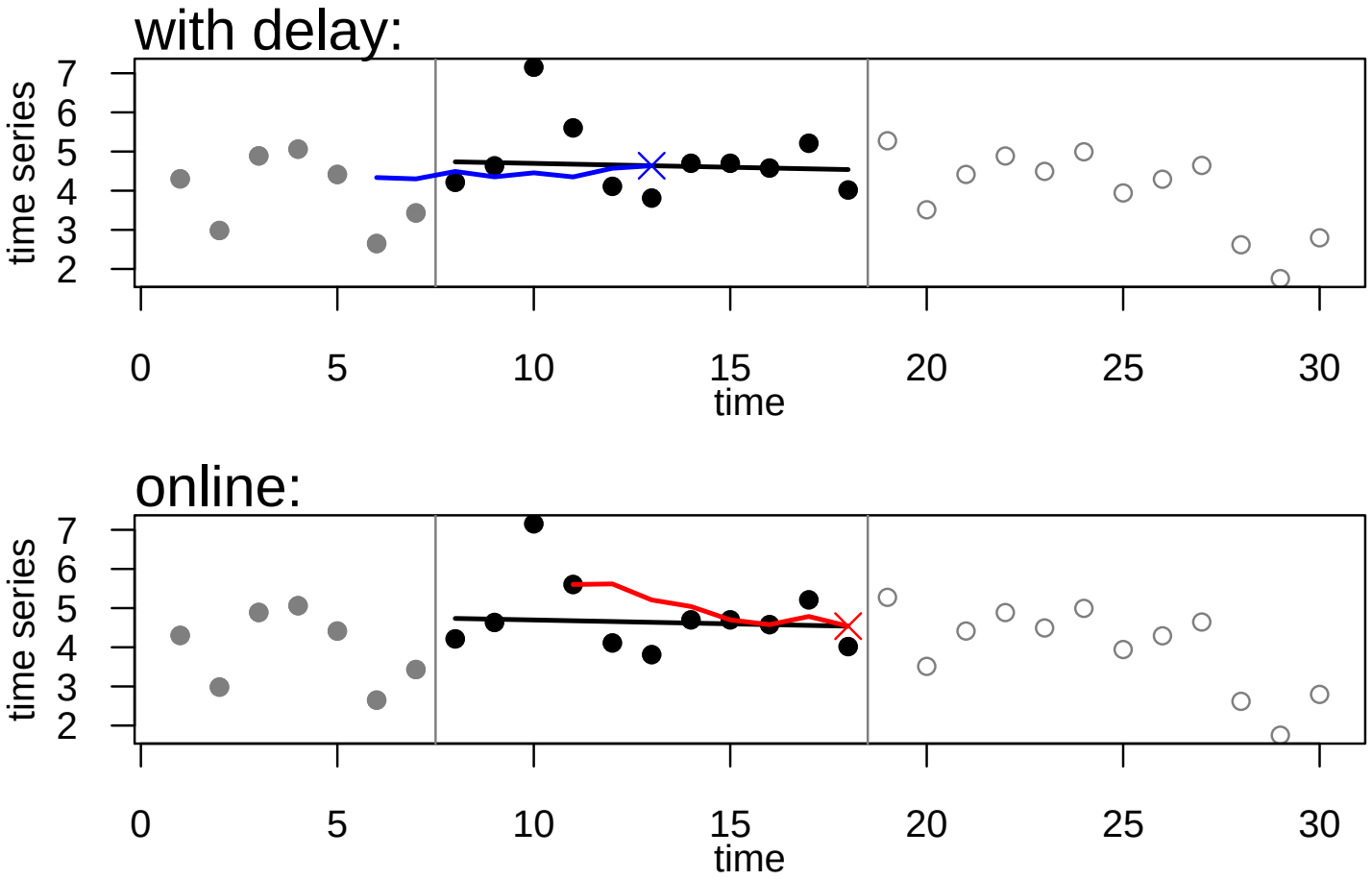
`robreg.filter` – Illustration

Robust **reg**ression in a moving time window:



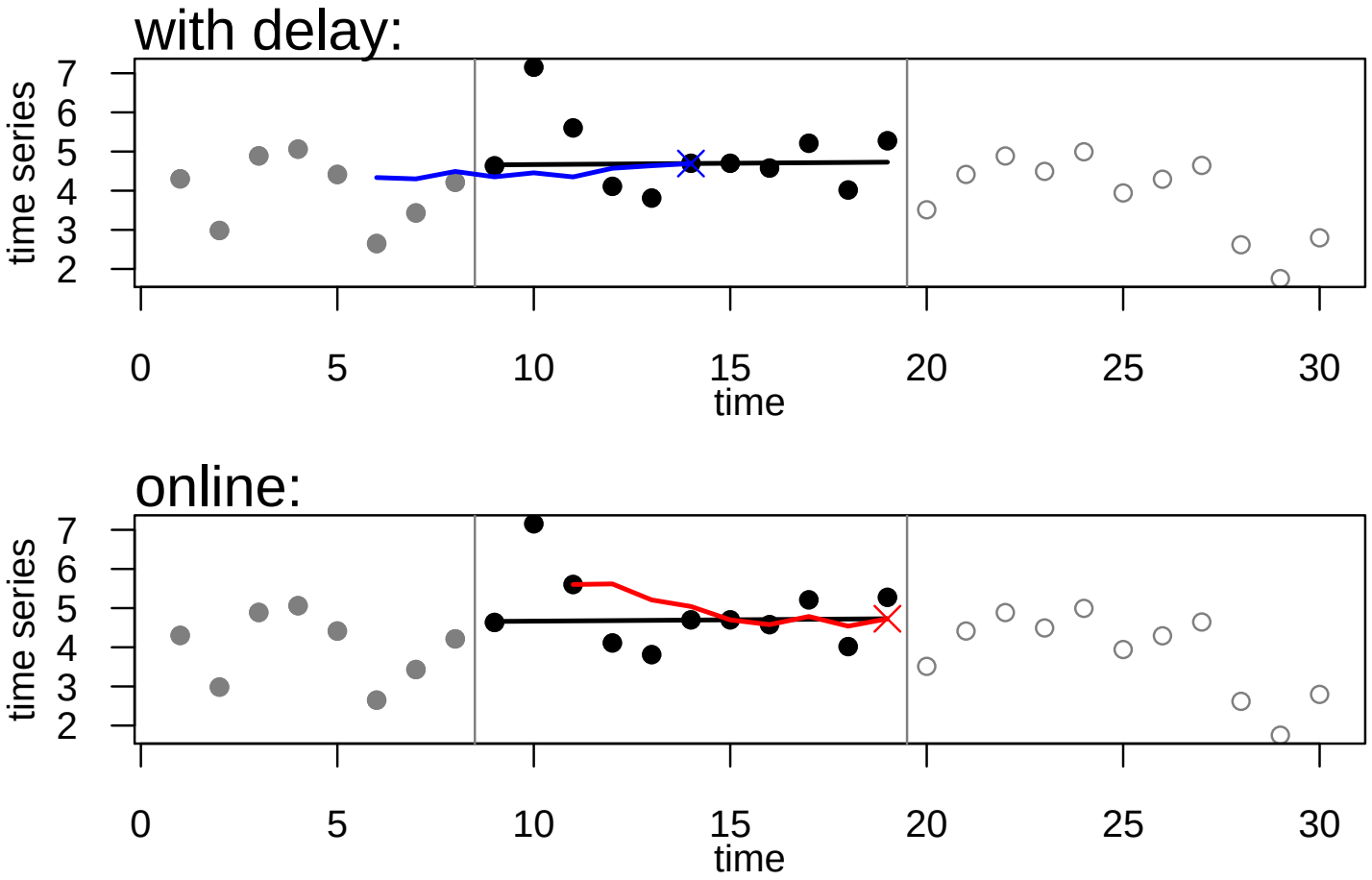
`robreg.filter` – Illustration

Robust **reg**ression in a moving time window:



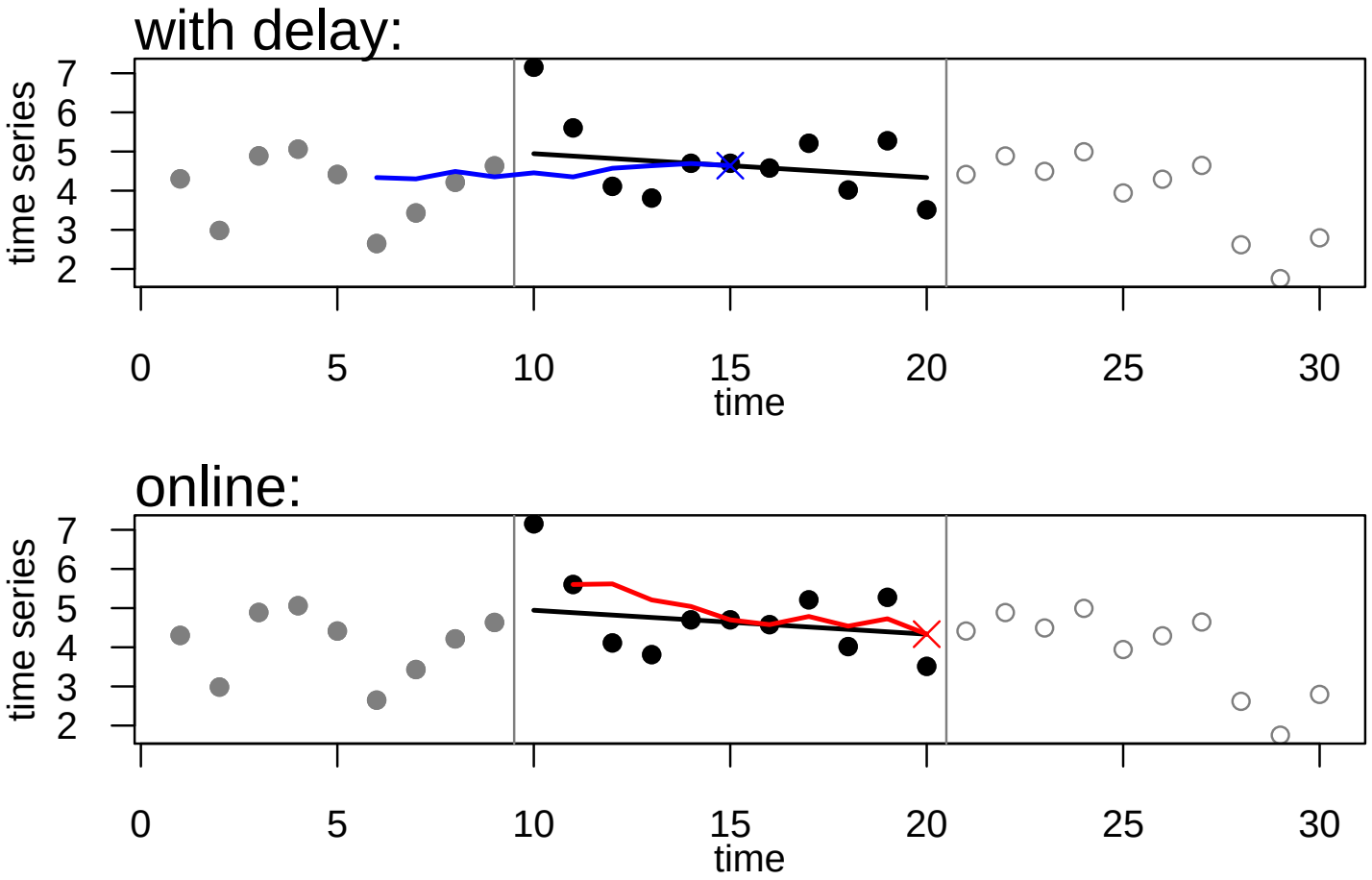
`robreg.filter` – Illustration

Robust **reg**ression in a moving time window:



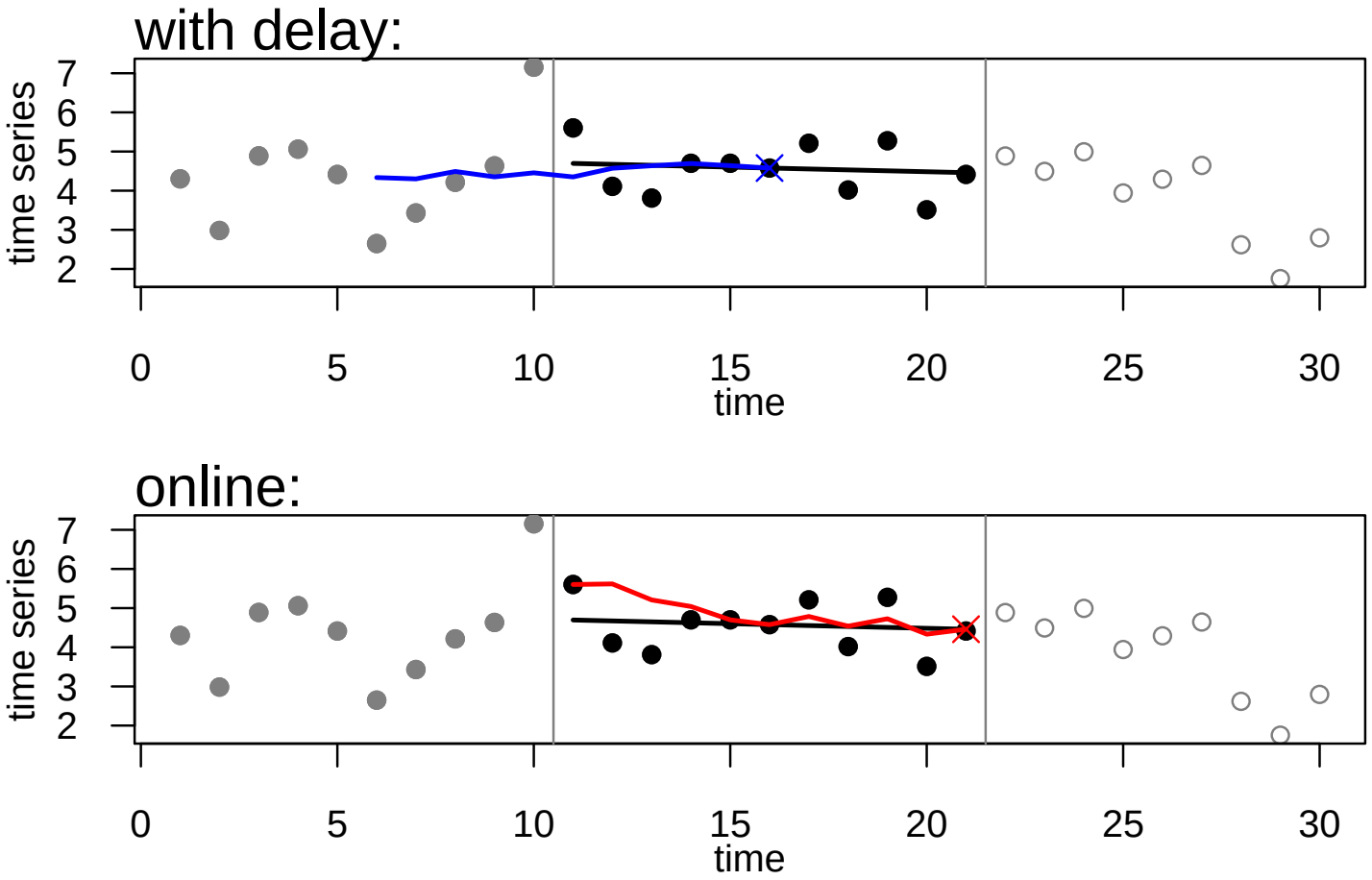
`robreg.filter` – Illustration

Robust **reg**ression in a moving time window:



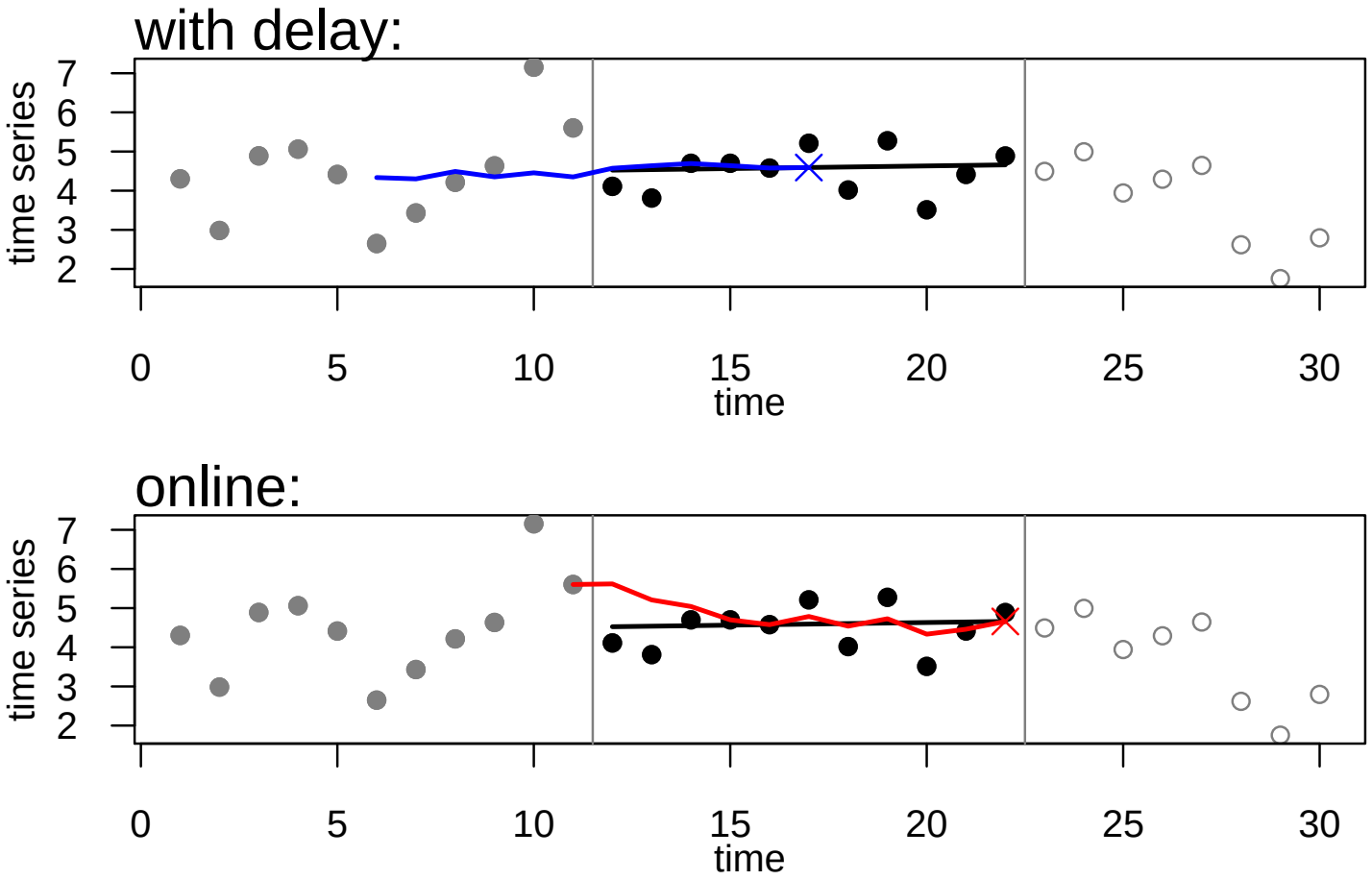
`robreg.filter` – Illustration

Robust **reg**ression in a moving time window:



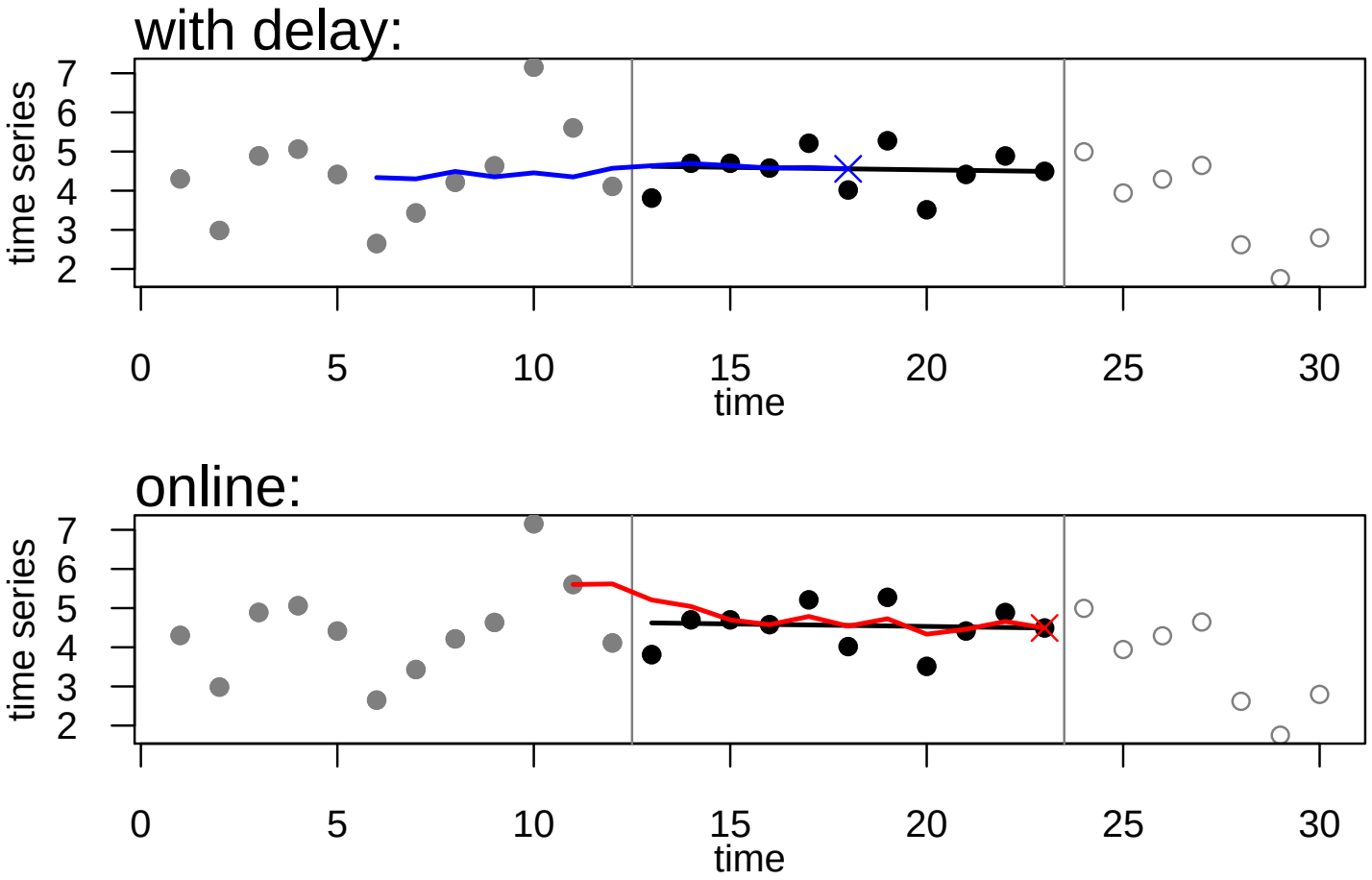
`robreg.filter` – **Illustration**

Robust **reg**ression in a moving time window:



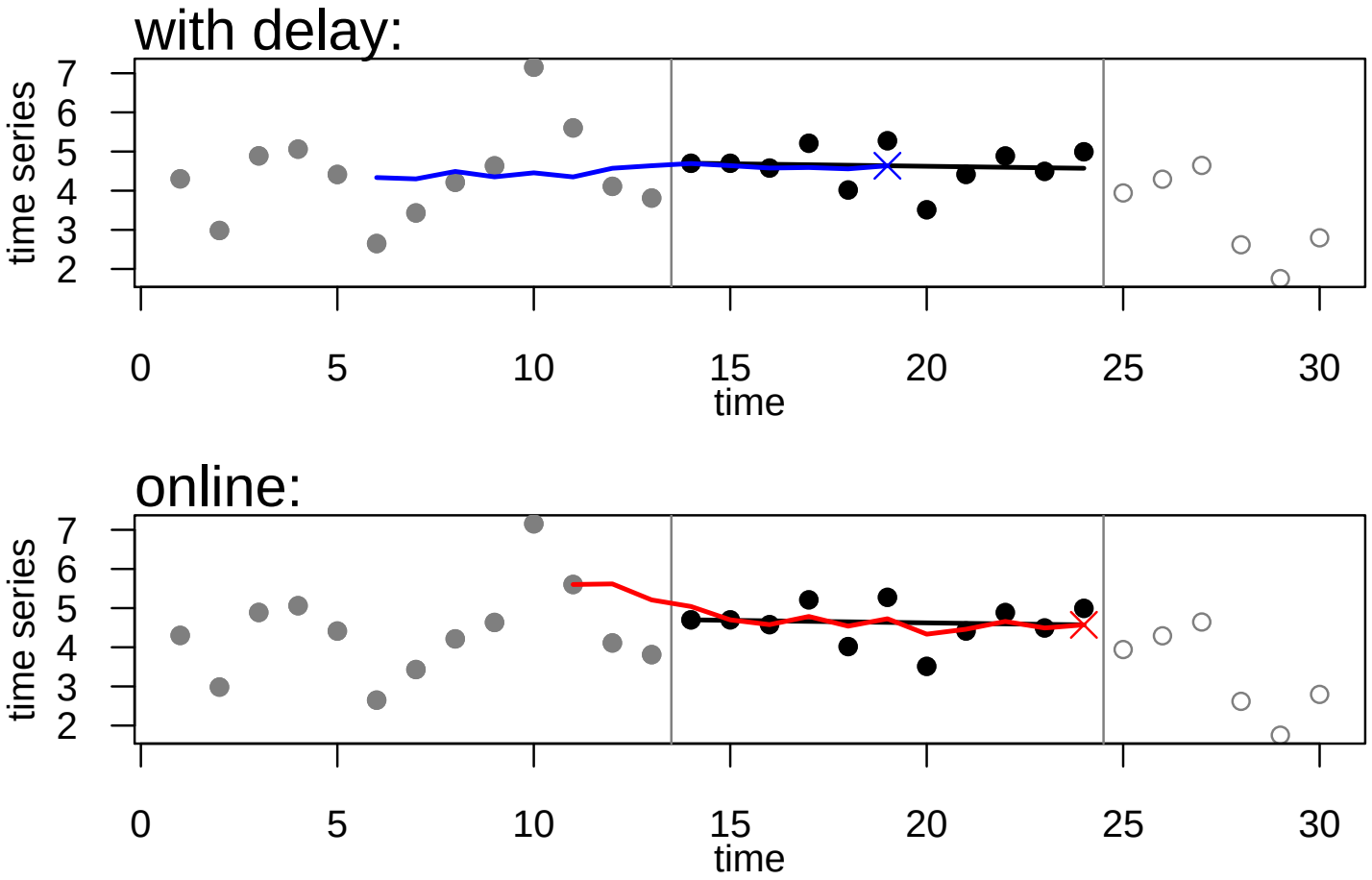
`robreg.filter` – Illustration

Robust **reg**ression in a moving time window:



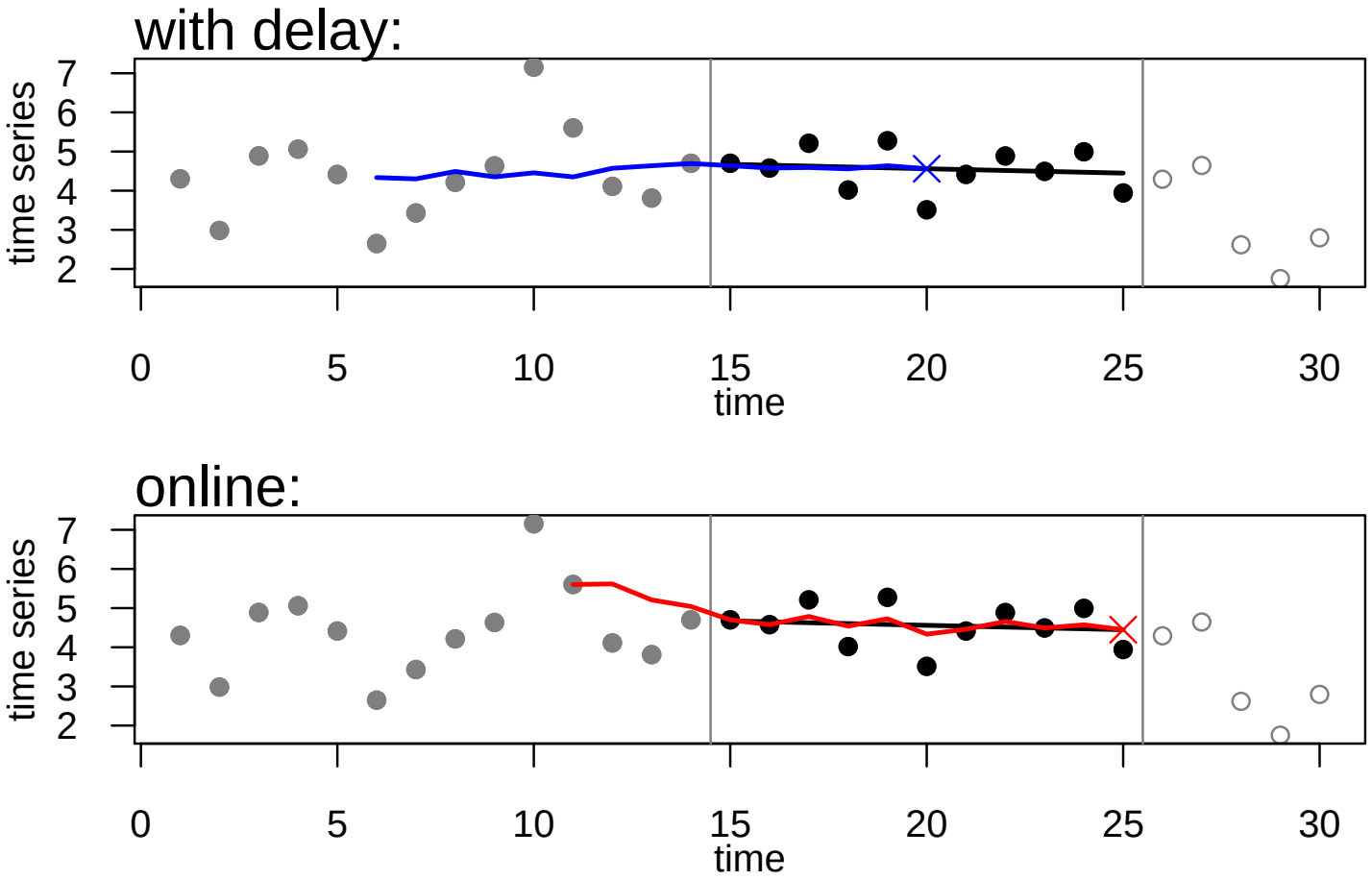
`robreg.filter` – Illustration

Robust **reg**ression in a moving time window:



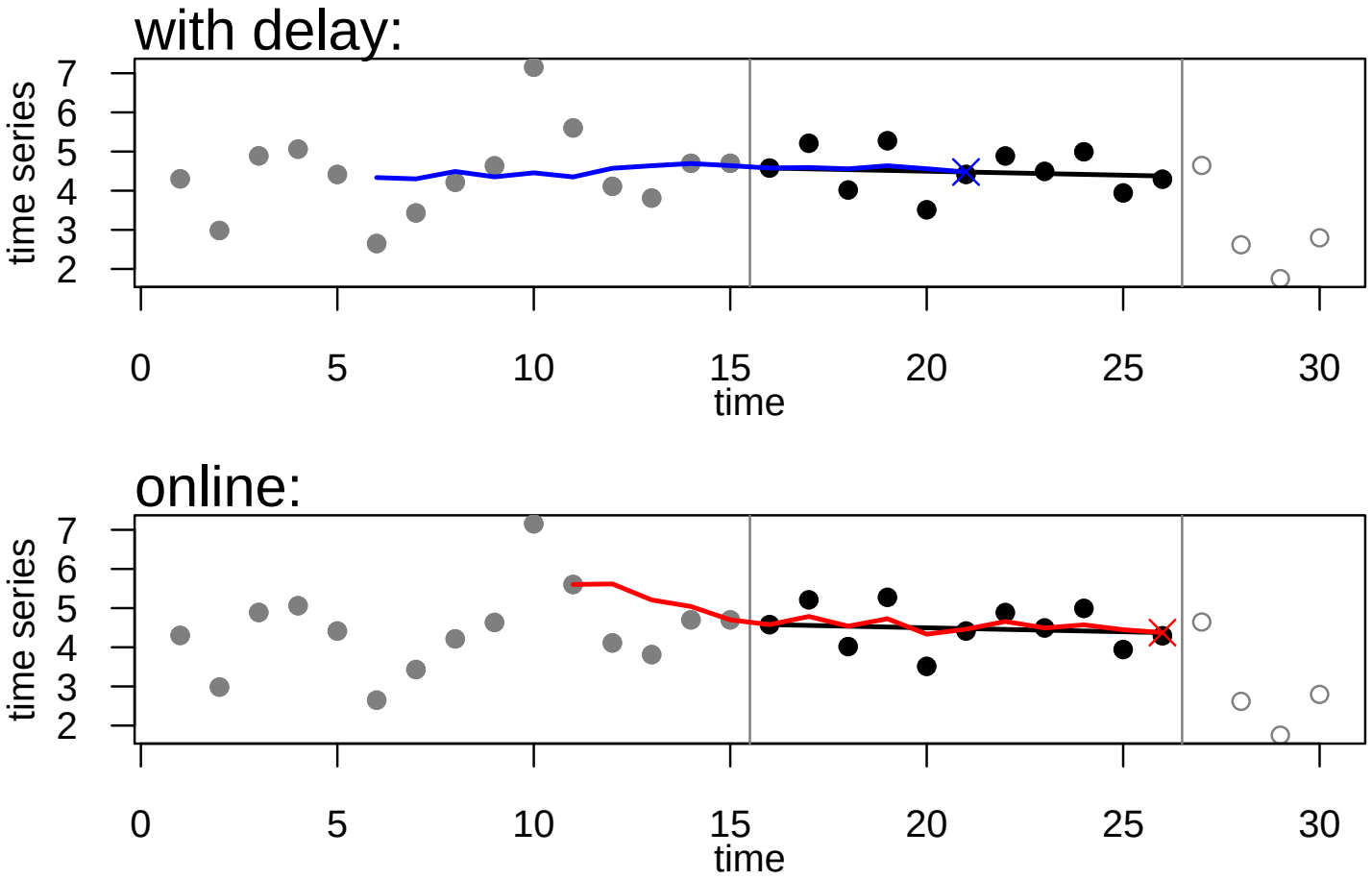
`robreg.filter` – Illustration

Robust **reg**ression in a moving time window:



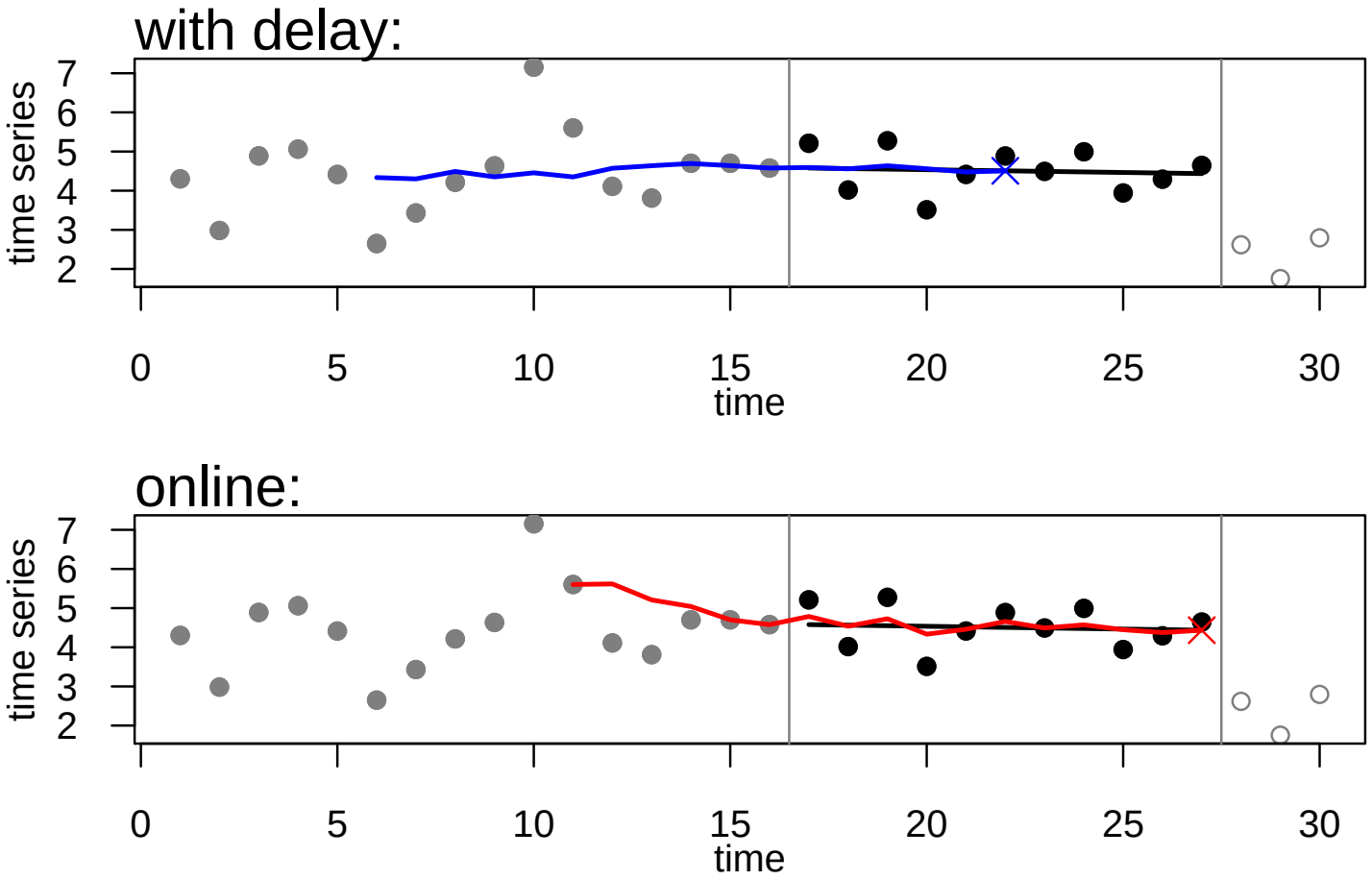
`robreg.filter` – Illustration

Robust **reg**ression in a moving time window:



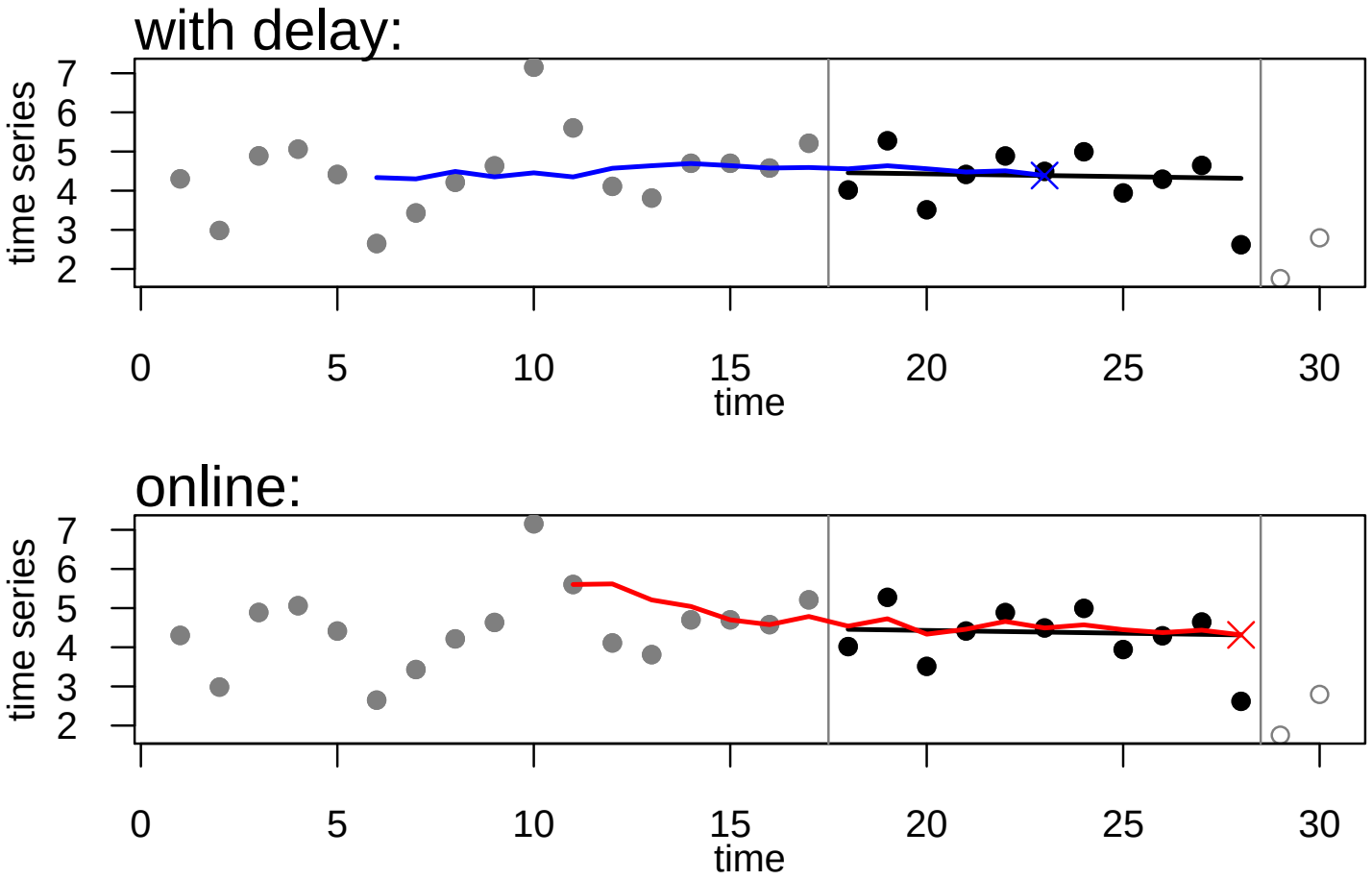
`robreg.filter` – **Illustration**

Robust **reg**ression in a moving time window:



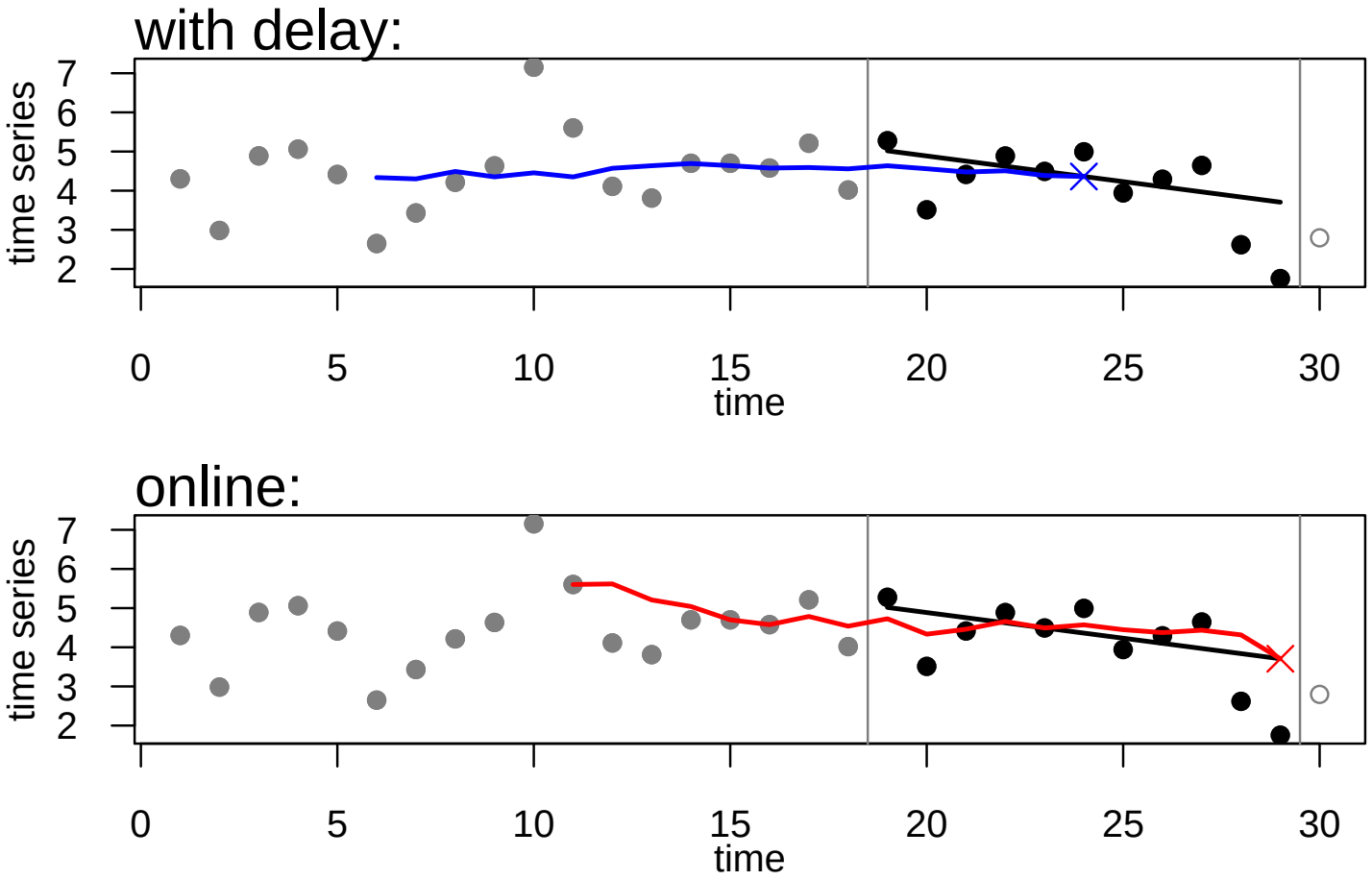
`robreg.filter` – Illustration

Robust **reg**ression in a moving time window:



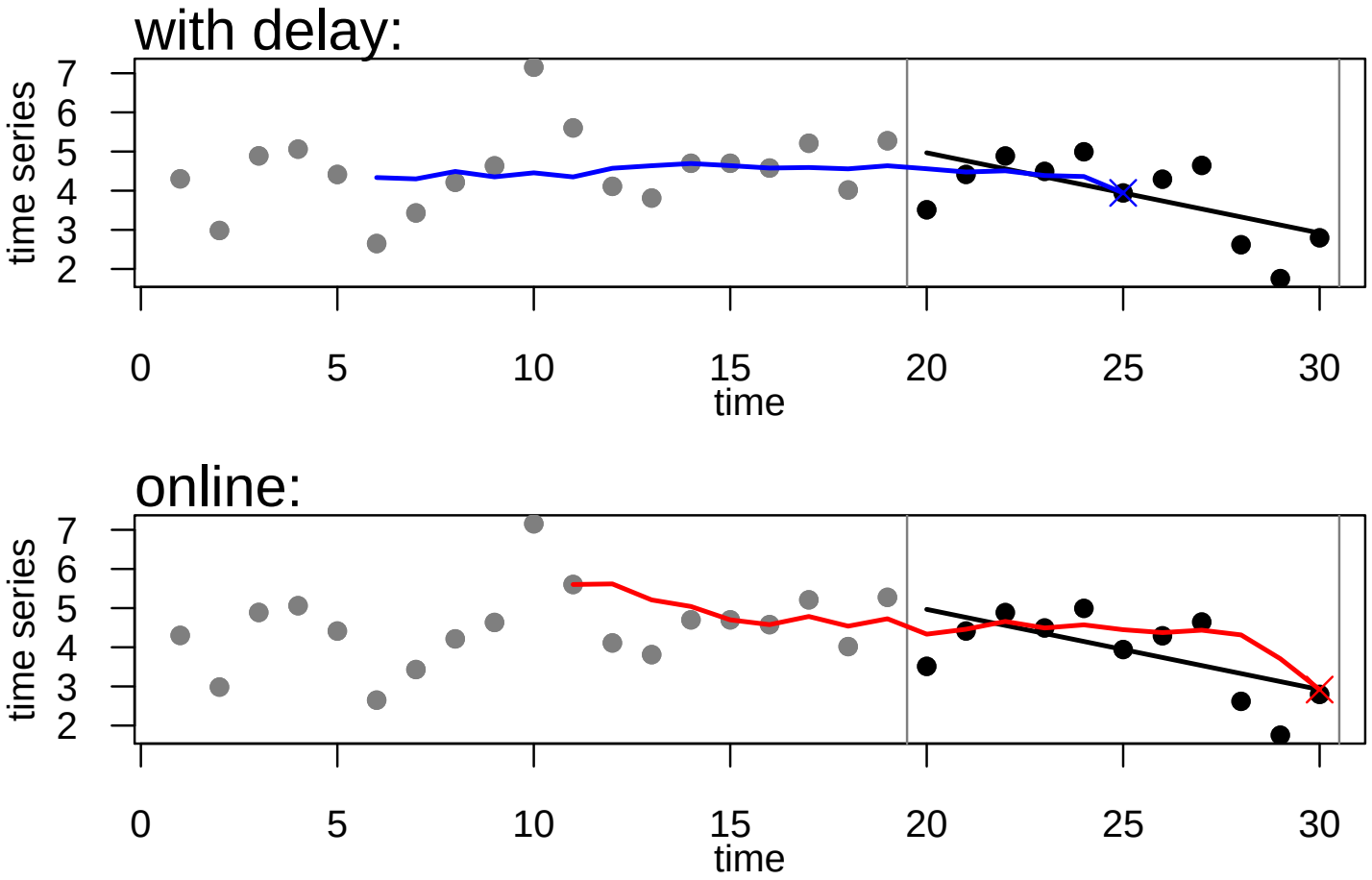
`robreg.filter` – Illustration

Robust **reg**ression in a moving time window:



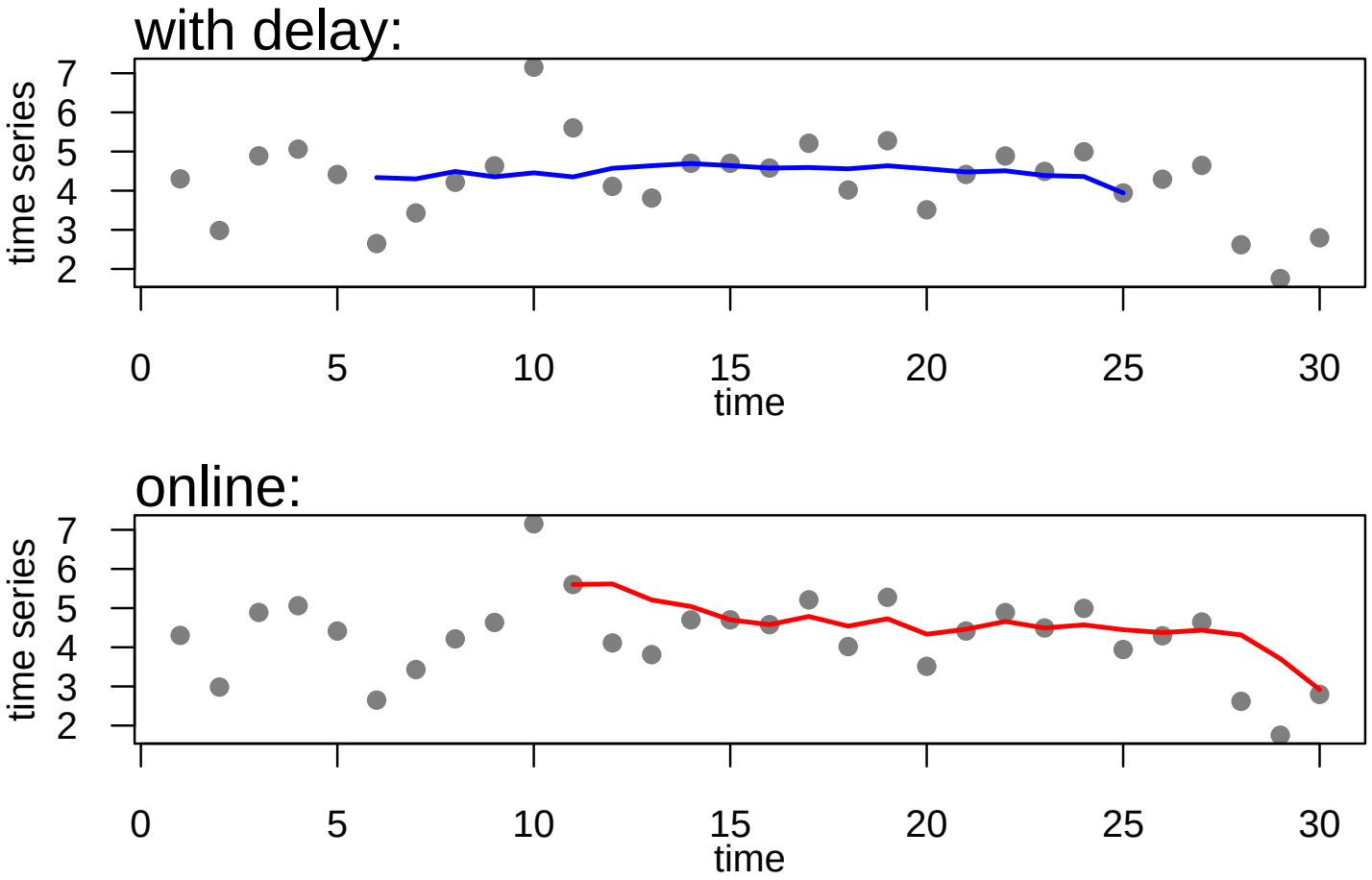
`robreg.filter` – Illustration

Robust **reg**ression in a moving time window:



`robreg.filter` – Illustration

Robust **reg**ression in a moving time window:



`robreg.filter` – Parameter Options

- Robust regression techniques

  `robreg.filter(...,method=" ")`

RM	Repeated Median	(Siegel, 1982)
LMS	Least Median of Squares	(Hampel, 1975; Rousseeuw, 1984)
LTS	Least Trimmed Squares	(Rousseeuw, 1983)
LQD	Least Quartile Difference	(Croux, Rousseeuw, Hössjer, 1994)
DR	Deepest Regression	(Rousseeuw and Hubert, 1999)

`robreg.filter` – Parameter Options

- Robust regression techniques

  `robreg.filter(...,method=" ")`

MED	Median	
RM	Repeated Median	(Siegel, 1982)
LMS	Least Median of Squares	(Hampel, 1975; Rousseeuw, 1984)
LTS	Least Trimmed Squares	(Rousseeuw, 1983)
LQD	Least Quartile Difference	(Croux, Rousseeuw, Hössjer, 1994)
DR	Deepest Regression	(Rousseeuw and Hubert, 1999)

`robreg.filter` – Parameter Options

- Robust regression techniques

  `robreg.filter(...,method=" ")`

MED	Median	
RM	Repeated Median	(Siegel, 1982)
LMS	Least Median of Squares	(Hampel, 1975; Rousseeuw, 1984)
LTS	Least Trimmed Squares	(Rousseeuw, 1983)
LQD	Least Quartile Difference	(Croux, Rousseeuw, Hössjer, 1994)
DR	Deepest Regression	(Rousseeuw and Hubert, 1999)

- Window width $\in \mathbb{N}$ (`width`)

`robreg.filter` – Parameter Options

- Robust regression techniques

  `robreg.filter(...,method=" ")`

MED	Median	
RM	Repeated Median	(Siegel, 1982)
LMS	Least Median of Squares	(Hampel, 1975; Rousseeuw, 1984)
LTS	Least Trimmed Squares	(Rousseeuw, 1983)
LQD	Least Quartile Difference	(Croux, Rousseeuw, Hössjer, 1994)
DR	Deepest Regression	(Rousseeuw and Hubert, 1999)

- Window width $\in \mathbb{N}$ (`width`)
- Minimum number of non-missing observations within one window (`minNonNAs`)

`robreg.filter` – Parameter Options

- Robust regression techniques

  `robreg.filter(...,method=" ")`

MED	Median	
RM	Repeated Median	(Siegel, 1982)
LMS	Least Median of Squares	(Hampel, 1975; Rousseeuw, 1984)
LTS	Least Trimmed Squares	(Rousseeuw, 1983)
LQD	Least Quartile Difference	(Croux, Rousseeuw, Hössjer, 1994)
DR	Deepest Regression	(Rousseeuw and Hubert, 1999)

- Window width $\in \mathbb{N}$ (`width`)
- Minimum number of non-missing observations within one window (`minNonNAs`)
- Online estimation (`online = TRUE / FALSE`)

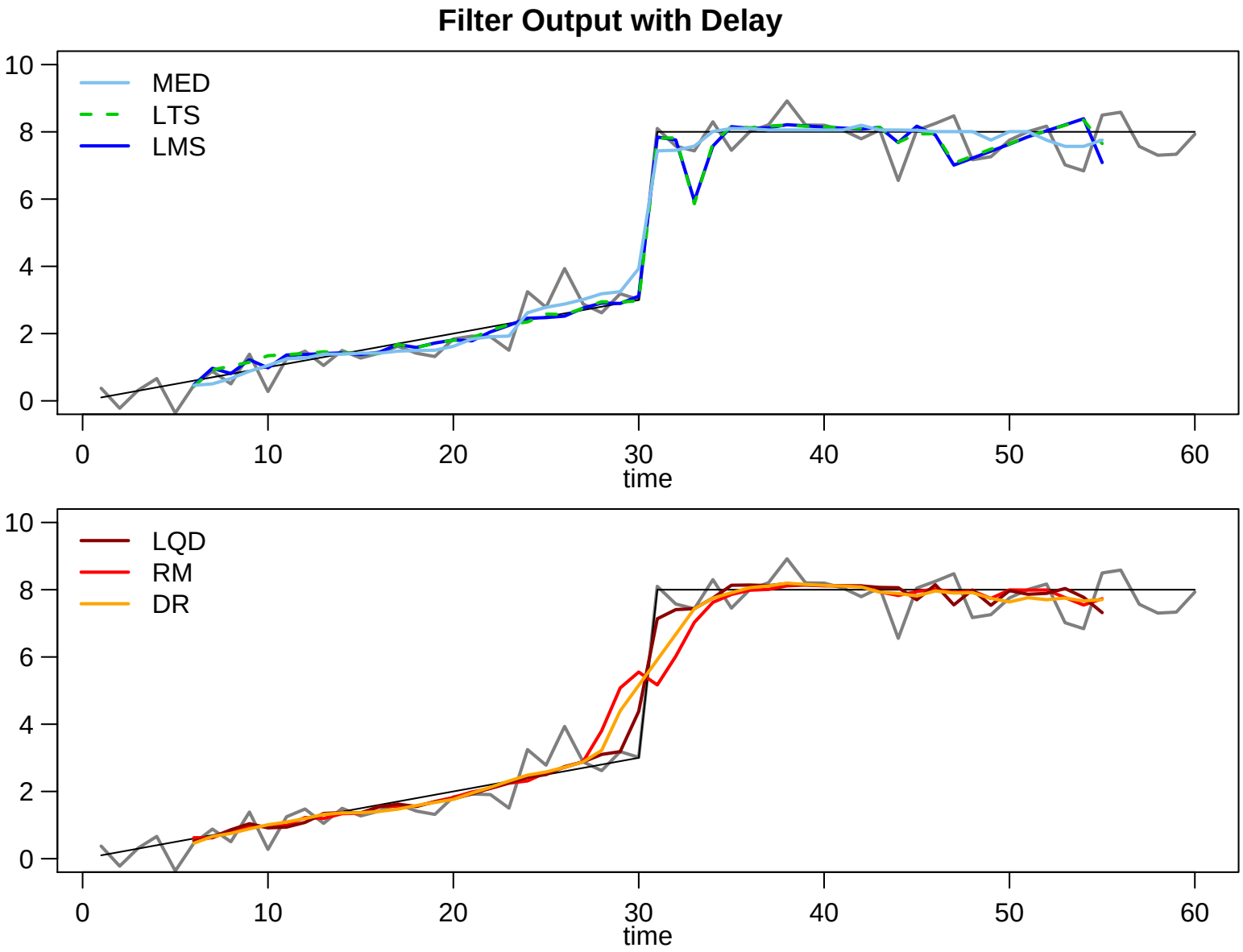
`robreg.filter` – Parameter Options

- Robust regression techniques

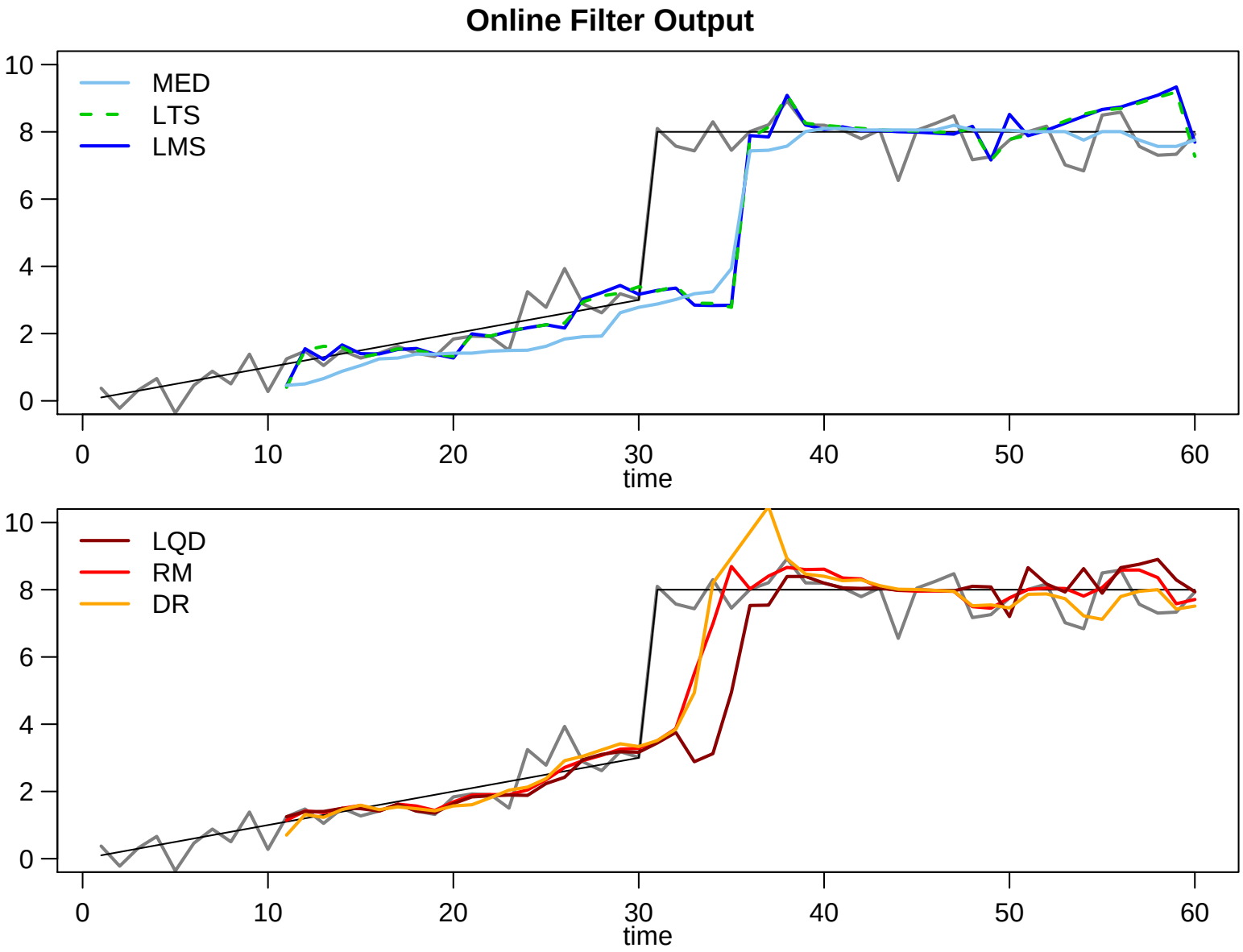
`robreg.filter(...,method=" ")`		**Aliases**
MED	Median	`med.filter`
RM	Repeated Median	`rm.filter`
LMS	Least Median of Squares	`lms.filter`
LTS	Least Trimmed Squares	`lts.filter`
LQD	Least Quartile Difference	`lqd.filter`
DR	Deepest Regression	`dr.filter`

- Window width $\in \mathbb{N}$ (`width`)
- Minimum number of non-missing observations within one window (`minNonNAs`)
- Online estimation (`online = TRUE / FALSE`)

robreg.filter – Example

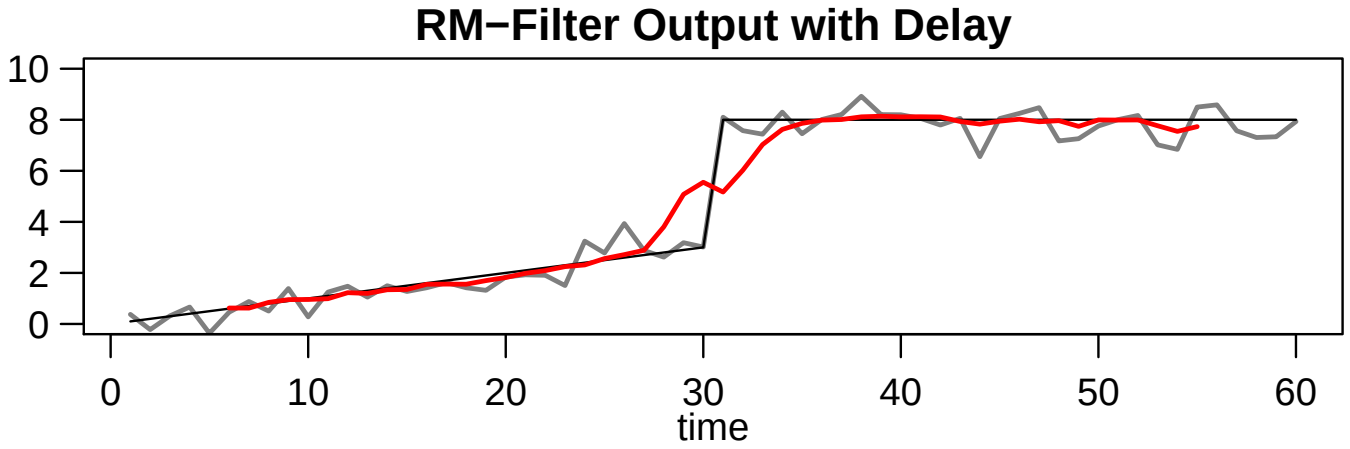


robreg.filter – Example

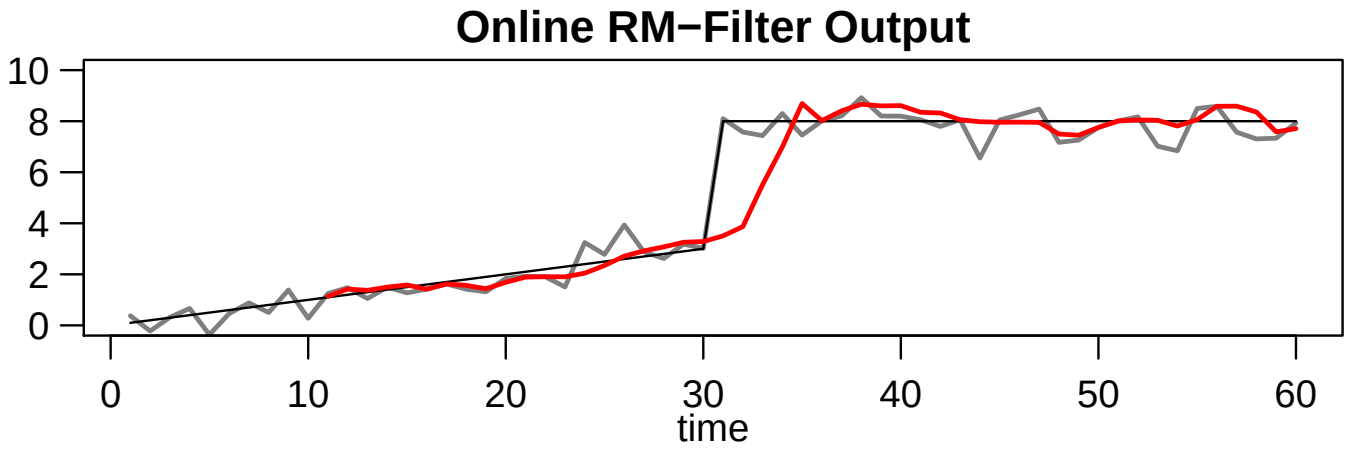


robreg.filter – **Option** extrapolate

```
rm.filter(ts, width=11,  extrapolate=FALSE, ... )
```



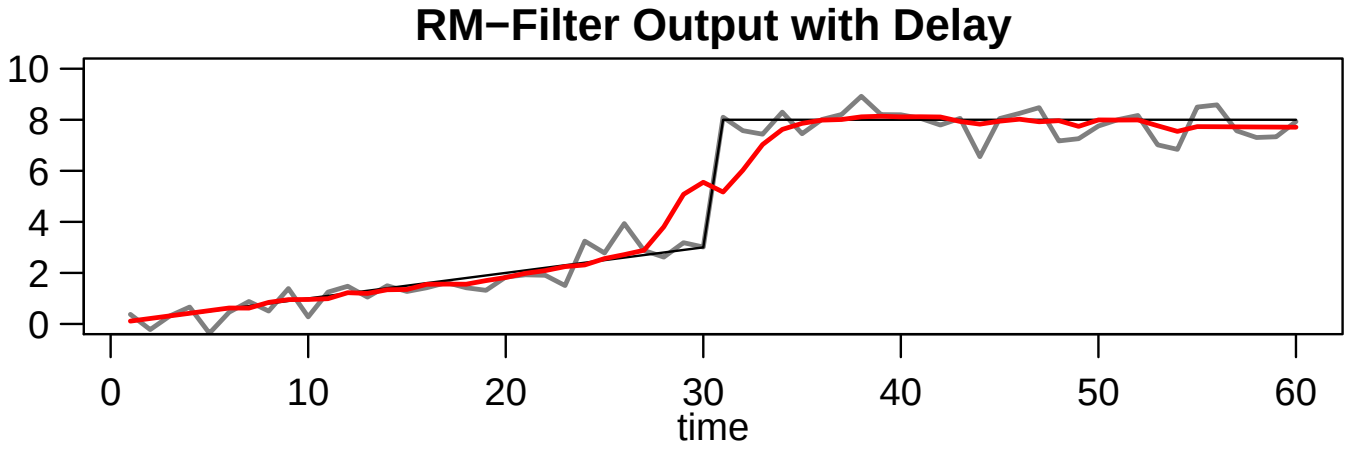
online=FALSE



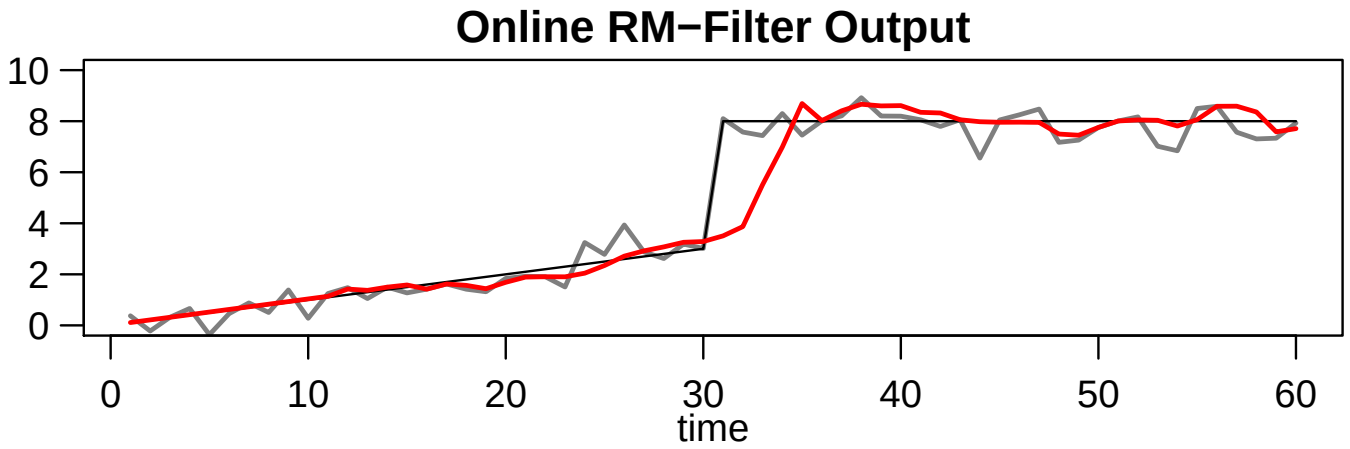
online=TRUE

robreg.filter – **Option** extrapolate

```
rm.filter(ts, width=11,  extrapolate=TRUE , ... )
```



online=FALSE



online=TRUE

Repeated Median Regression

(Siegel, 1982)

Sample in one time window $\mathbf{y}_t = (y_{t+i})'$

$$\widehat{\beta}_t^{\text{RM}} = \underset{i}{\text{med}} \left\{ \underset{j \neq i}{\text{med}} \left\{ \frac{y_{t+i} - y_{t+j}}{i - j} \right\} \right\}$$

$$\widehat{\mu}_t^{\text{RM}} = \underset{i}{\text{med}} \left\{ y_{t+i} + \widehat{\beta}_t^{\text{RM}} \cdot i \right\}$$

with $i = \begin{cases} -n+1, \ldots, -1, 0 & \text{(online)} \\ -m, \ldots, m \text{ and } n = 2m+1 & \text{(with delay)} \end{cases}$

`hybrid.filter`

Application of 'subfilters' to a moving time window

$$\mathbf{y}_t = (y_{t-m}, \ldots, y_{t-1},\ y_t,\ y_{t+1}, \ldots, y_{t+m})'$$

with $m = \frac{\texttt{width}-1}{2}$

`hybrid.filter`

Application of 'subfilters' to a moving time window

$$\mathbf{y}_t = (y_{t-m}, \ldots, y_{t-1},\ y_t,\ y_{t+1}, \ldots, y_{t+m})'$$

with $m = \frac{\texttt{width}-1}{2}$

$\Phi_j(\mathbf{y}_t),\ j = 1, \ldots, k,$ subfilters

Filter output $\hat{\mu}_t = \text{med}\{\Phi_1(\mathbf{y}_t), \Phi_2(\mathbf{y}_t), \ldots, \Phi_k(\mathbf{y}_t)\}$

hybrid.filter

Application of 'subfilters' to a moving time window

$$\mathbf{y}_t = (y_{t-m}, \ldots, y_{t-1},\ y_t,\ y_{t+1}, \ldots, y_{t+m})'$$

with $m = \frac{\texttt{width}-1}{2}$

$\Phi_j(\mathbf{y}_t),\ j = 1, \ldots, k,$ subfilters

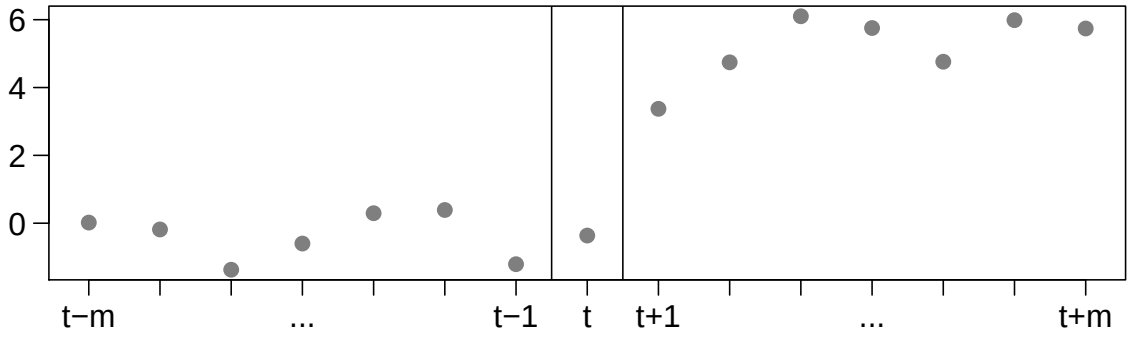
Filter output $\hat{\mu}_t = \text{med}\{\Phi_1(\mathbf{y}_t), \Phi_2(\mathbf{y}_t), \ldots, \Phi_k(\mathbf{y}_t)\}$

Parameters options:

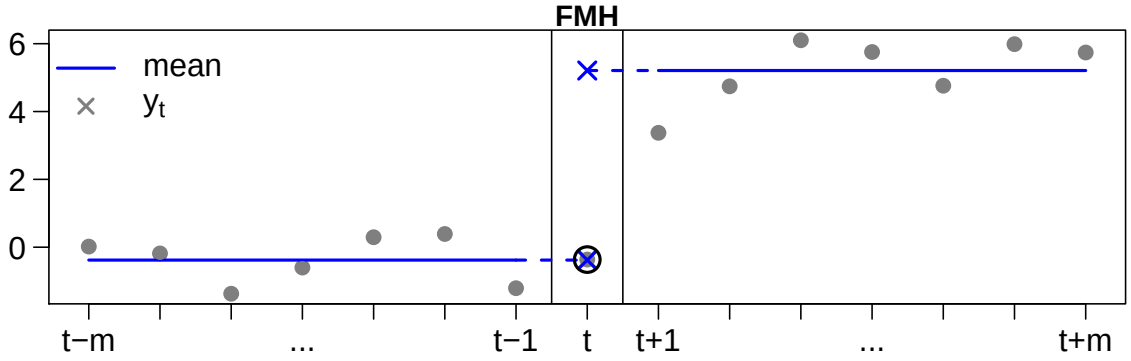
`method`, `width`, `minNonNAs`, and `extrapolate`

Online estimation not possible!!!

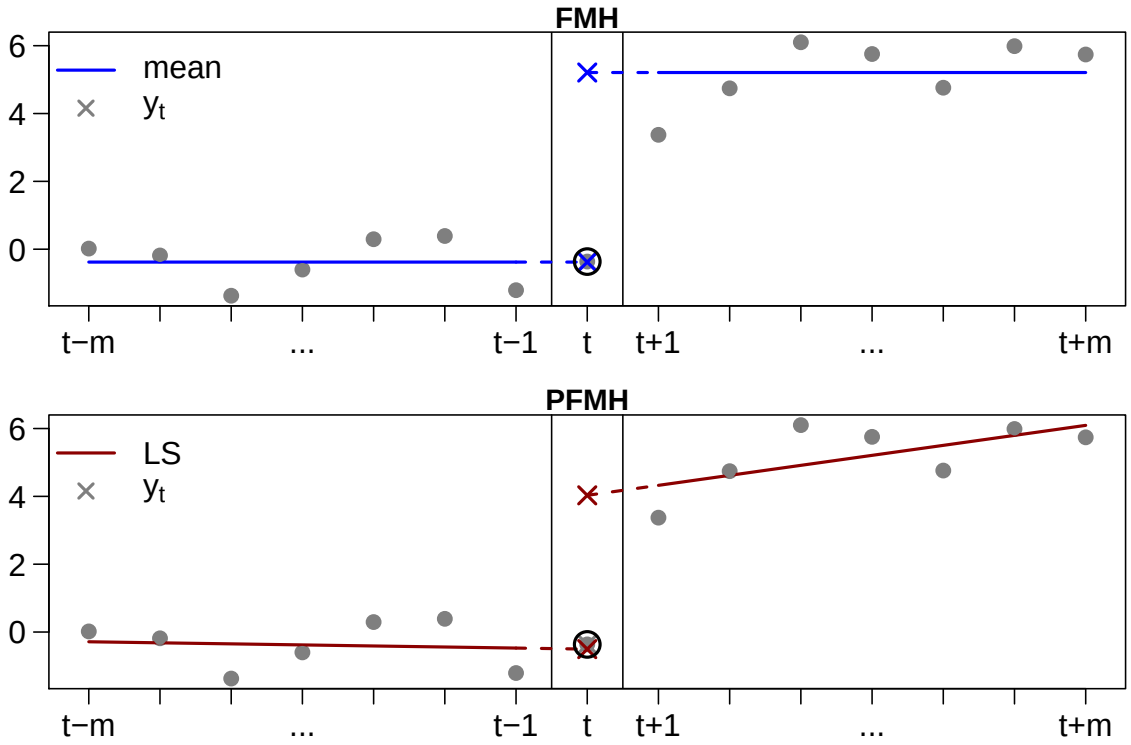
`hybrid.filter` – Estimation in One Window



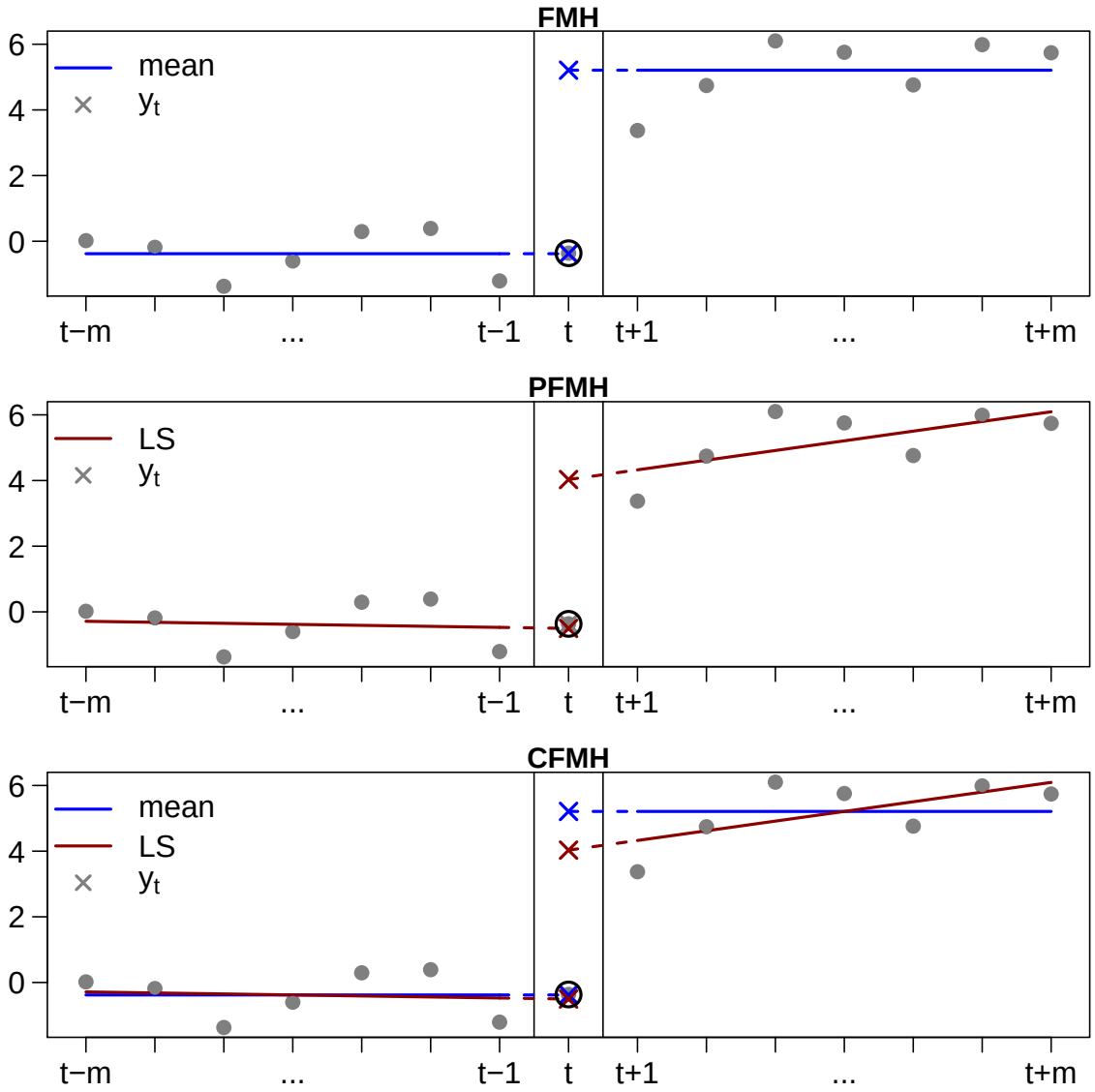
`hybrid.filter` – Estimation in One Window



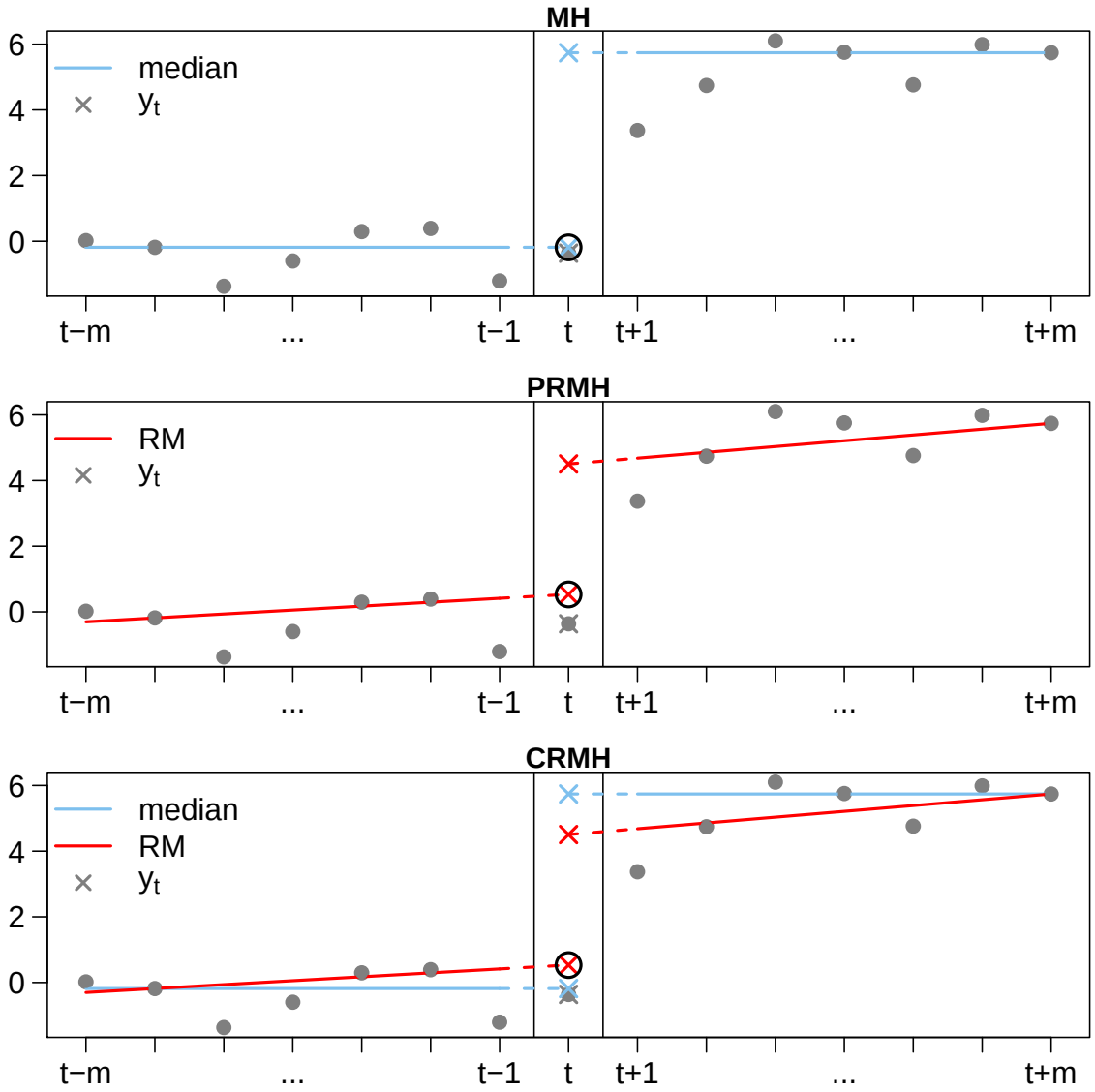
`hybrid.filter` – Estimation in One Window



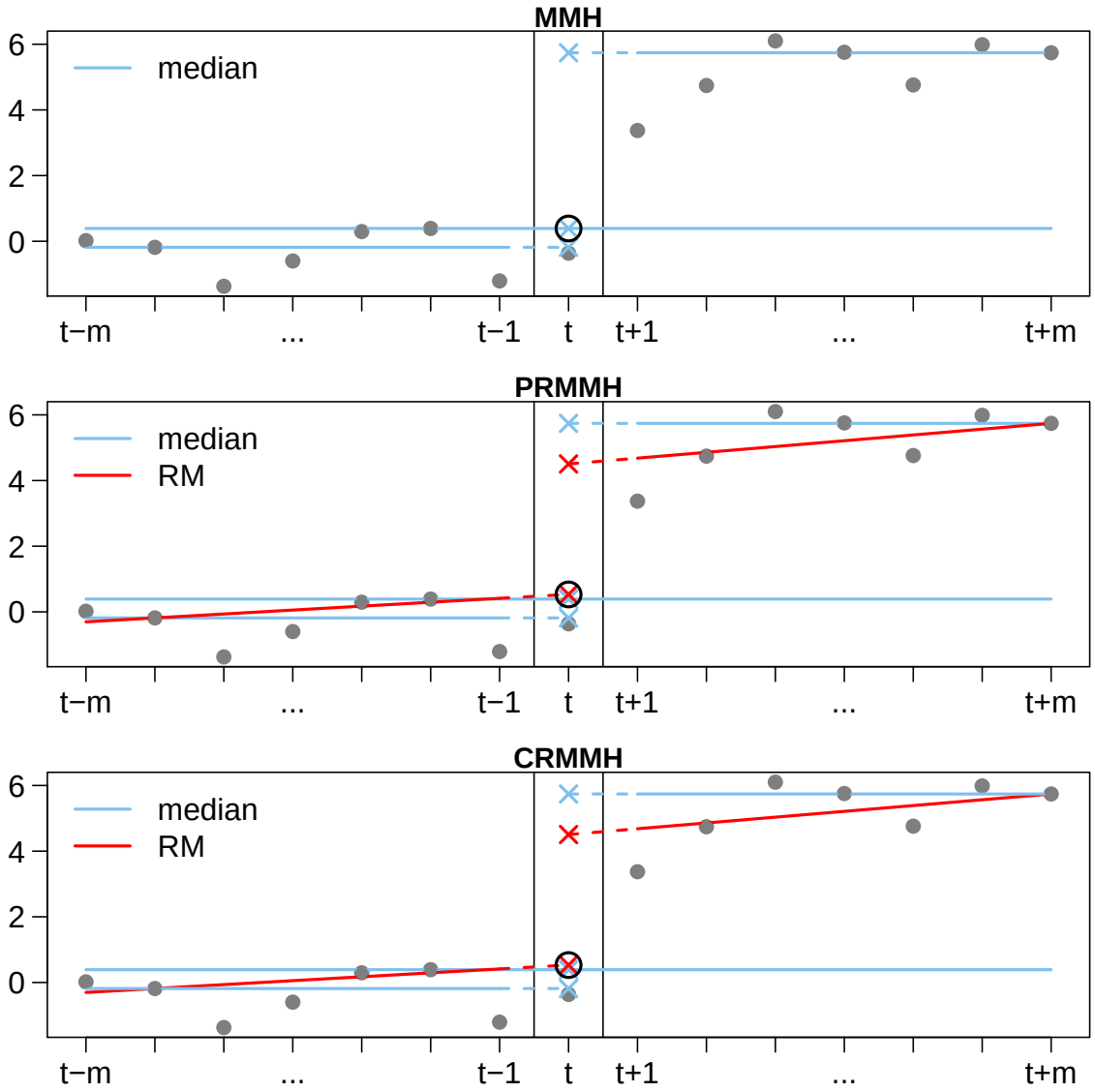
`hybrid.filter` – Estimation in One Window



`hybrid.filter` – Estimation in One Window



`hybrid.filter` – Estimation in One Window

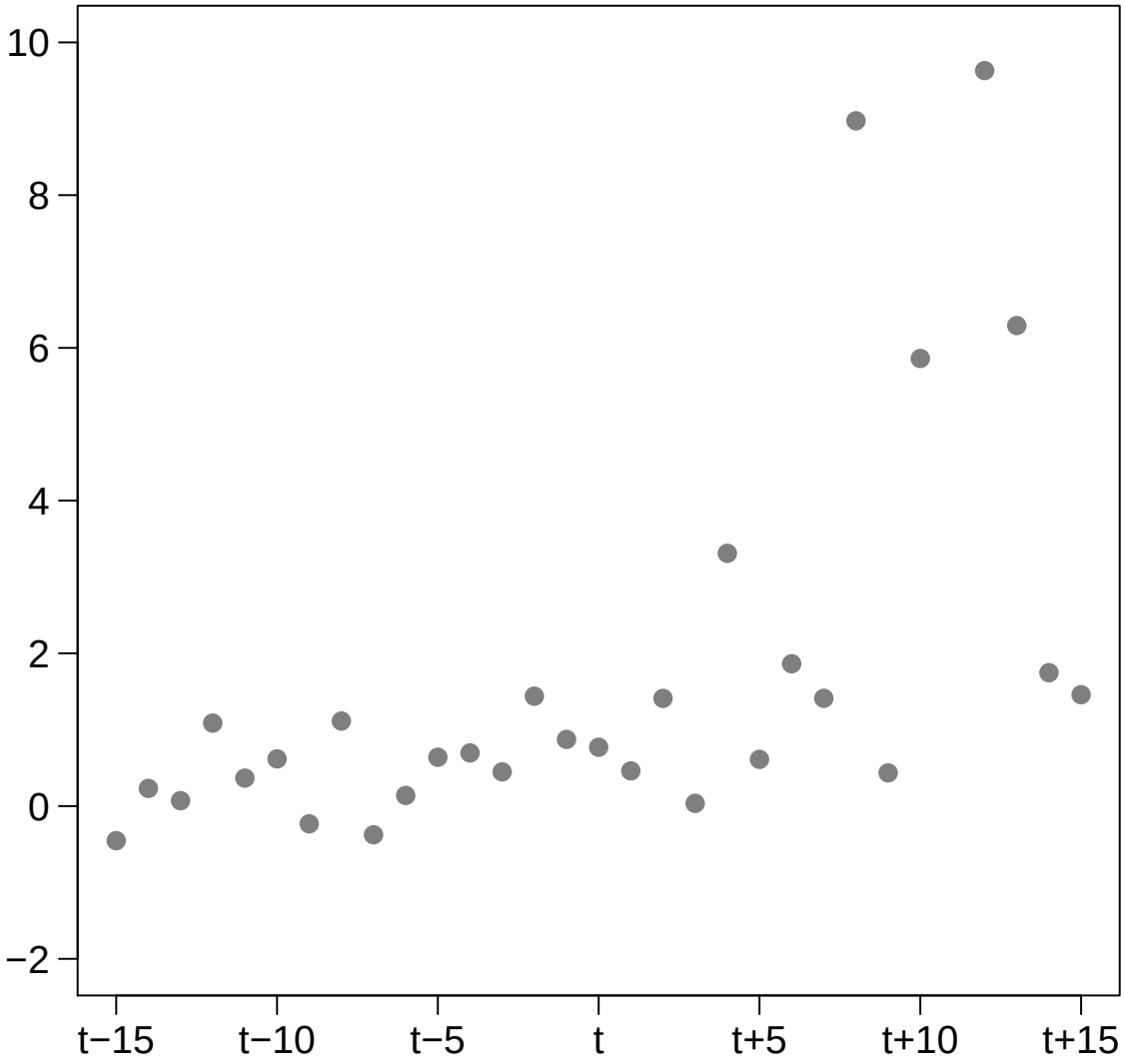


`dw.filter` – Parameter Options

Double **w**indow filtering techniques with

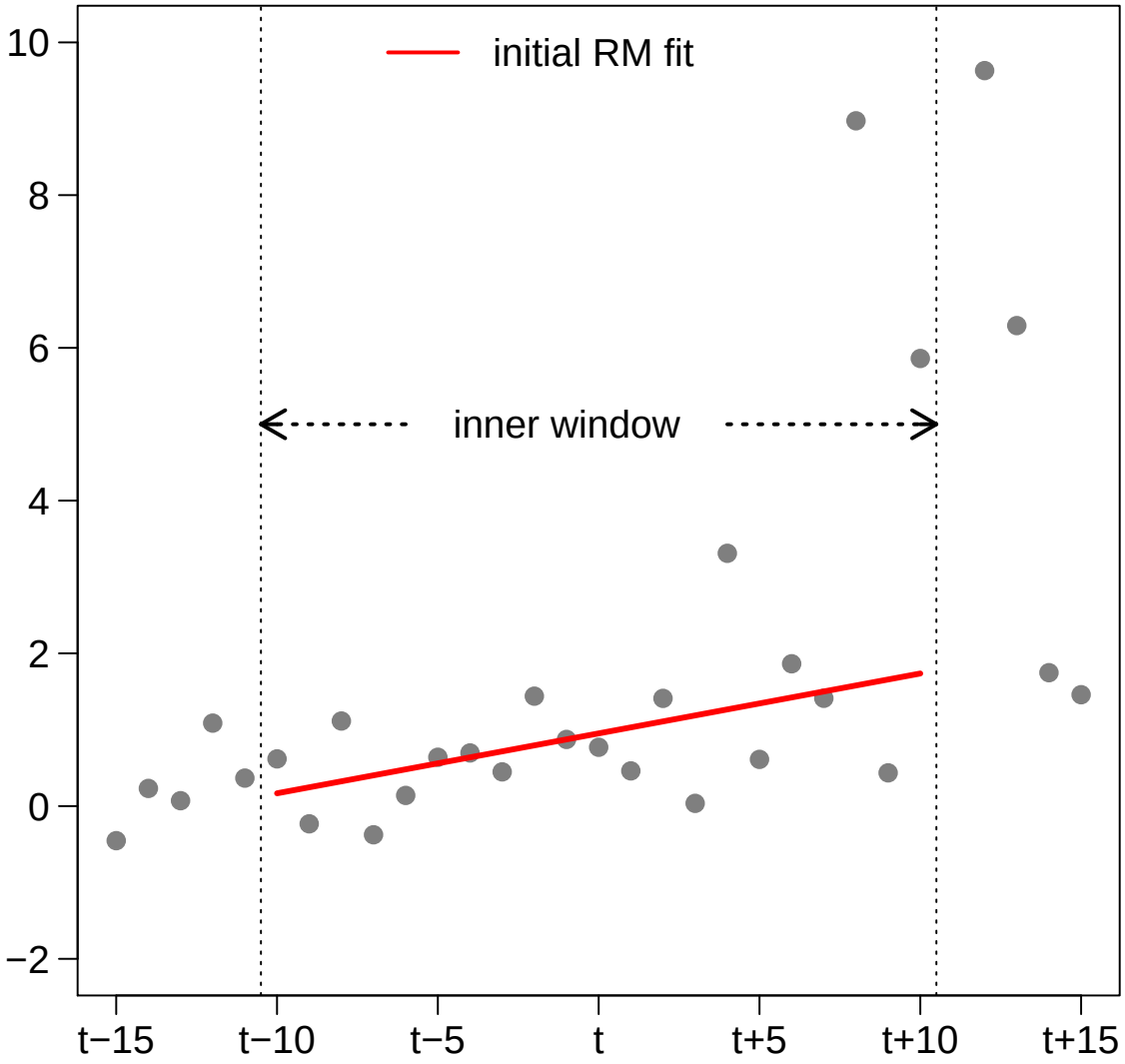
`outer.width`	Outer window width ($\in \mathbb{N}$)
`inner.width`	Inner window width ($\in \mathbb{N}$, $<$ `outer.width`)
`method`	Filter method(s)
`scale`	Scale estimation method
`d`	Trimming factor
`minNonNAs`	Minimum number of non-missing observations within one window
`online`	`TRUE` / `FALSE` for online / delayed estimation
`extrapolate`	`TRUE` / `FALSE` for extrapolation to the edges

dw.filter – Estimation in One Window



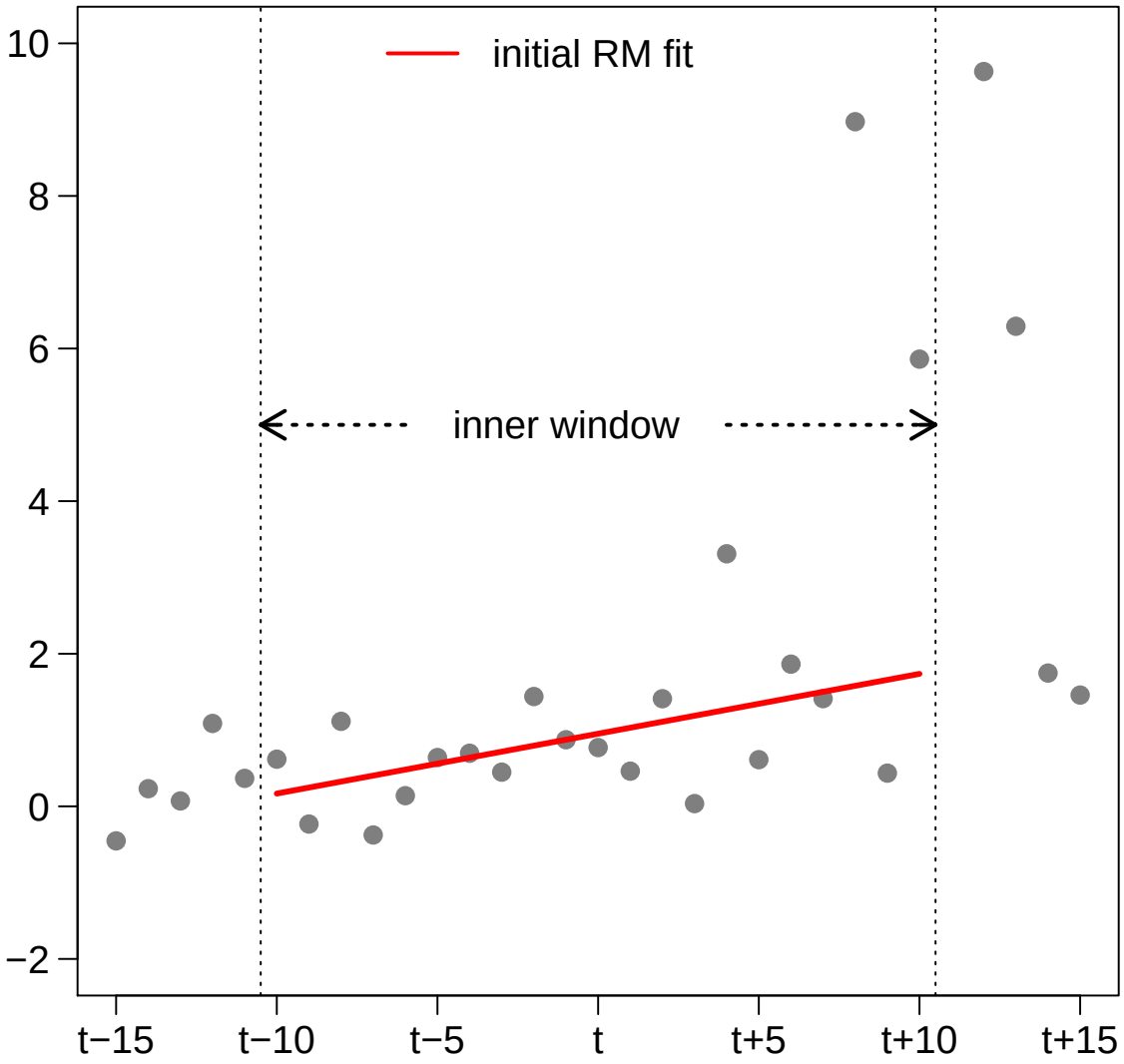
method
MTM
TRM
MRM
RM

dw.filter – Estimation in One Window



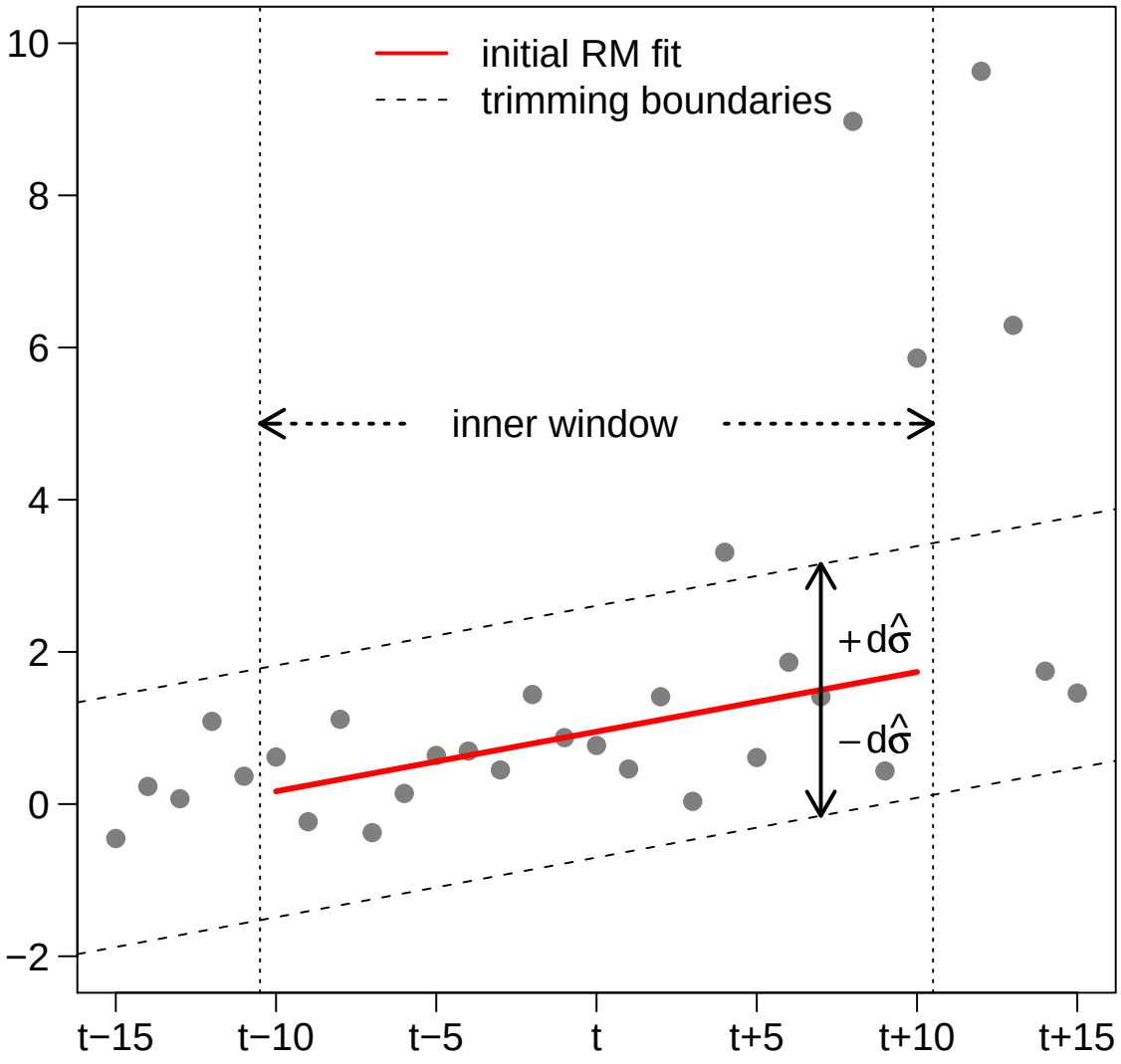
method	inner window
MTM	MED
TRM	RM
MRM	RM
RM	RM slope

dw.filter – Estimation in One Window



method	inner window
DWMTM	MED
DWTRM	RM
DWMRM	RM
DWRM	RM slope

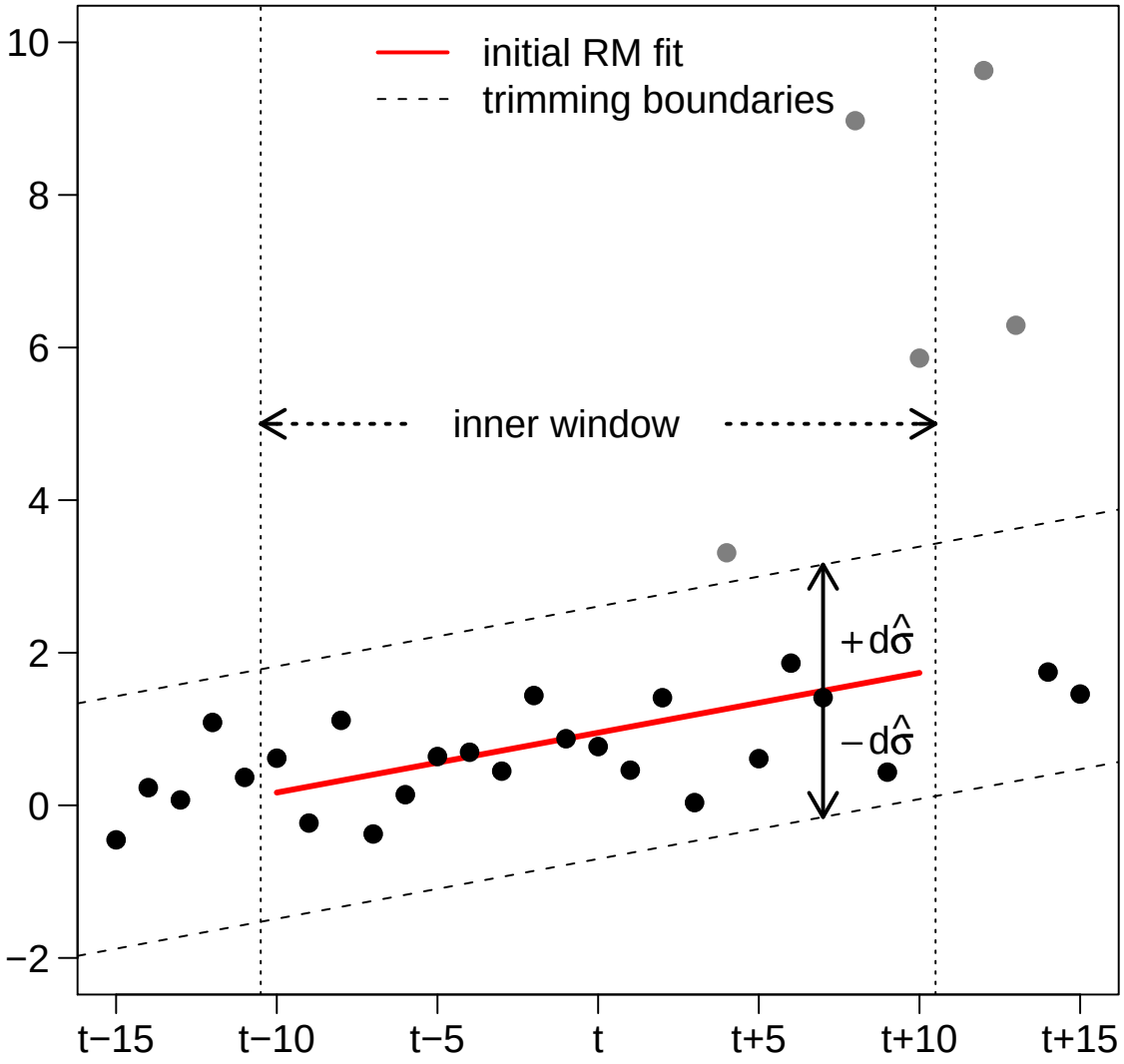
dw.filter – Estimation in One Window



method	inner window
DWMTM	MED
DWTRM	RM
DWMRM	RM
DWRM	RM slope

scale	$\hat{\sigma}$
MAD	median absolute deviation
QN	Q_n scale
SN	S_n scale (Rousseeuw, Croux, 1993)

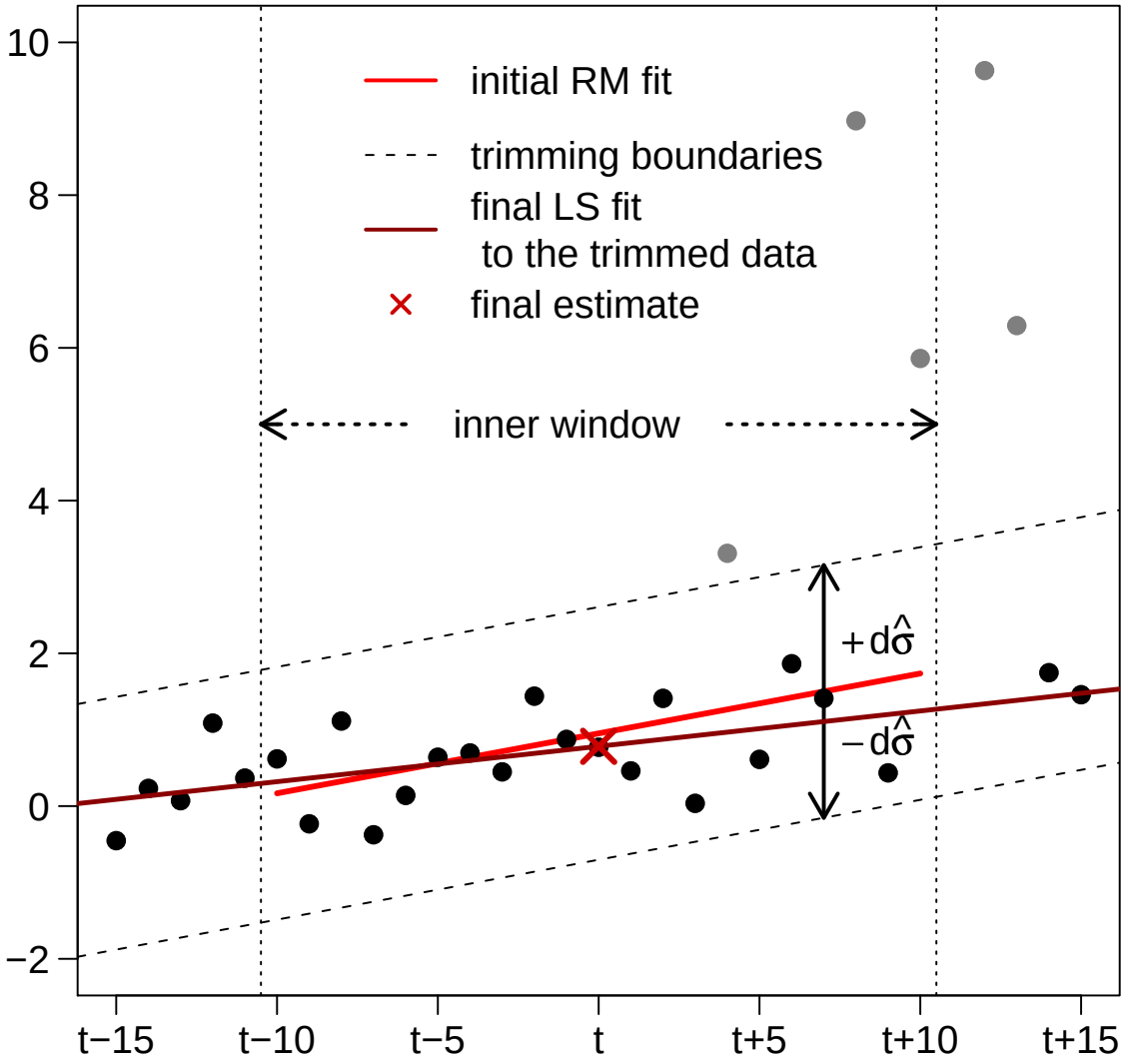
dw.filter – Estimation in One Window



method	inner window
DWMTM	MED
DWTRM	RM
DWMRM	RM
DWRM	RM slope

scale	$\hat{\sigma}$
MAD	median absolute deviation
QN	Q_n scale
SN	S_n scale (Rousseeuw, Croux, 1993)

dw.filter – Estimation in One Window



method	inner window	outer window
DWMTM	MED	MEAN
DWTRM	RM	LS
DWMRM	RM	RM
DWRM	RM slope	RM location

scale	$\hat{\sigma}$
MAD	Median Absolute Deviation
QN	Q_n scale
SN	S_n scale (Rousseeuw, Croux, 1993)

`robust.filter` – Parameter Options

`width`	Window width $\in \mathbb{N}$	
`trend`	`"MED"`	Median
	`"RM"`	Repeated Median
	`"LMS"`	Least Median of Squares
	`"LTS"`	Least Trimmed Squares

`robust.filter` – Parameter Options

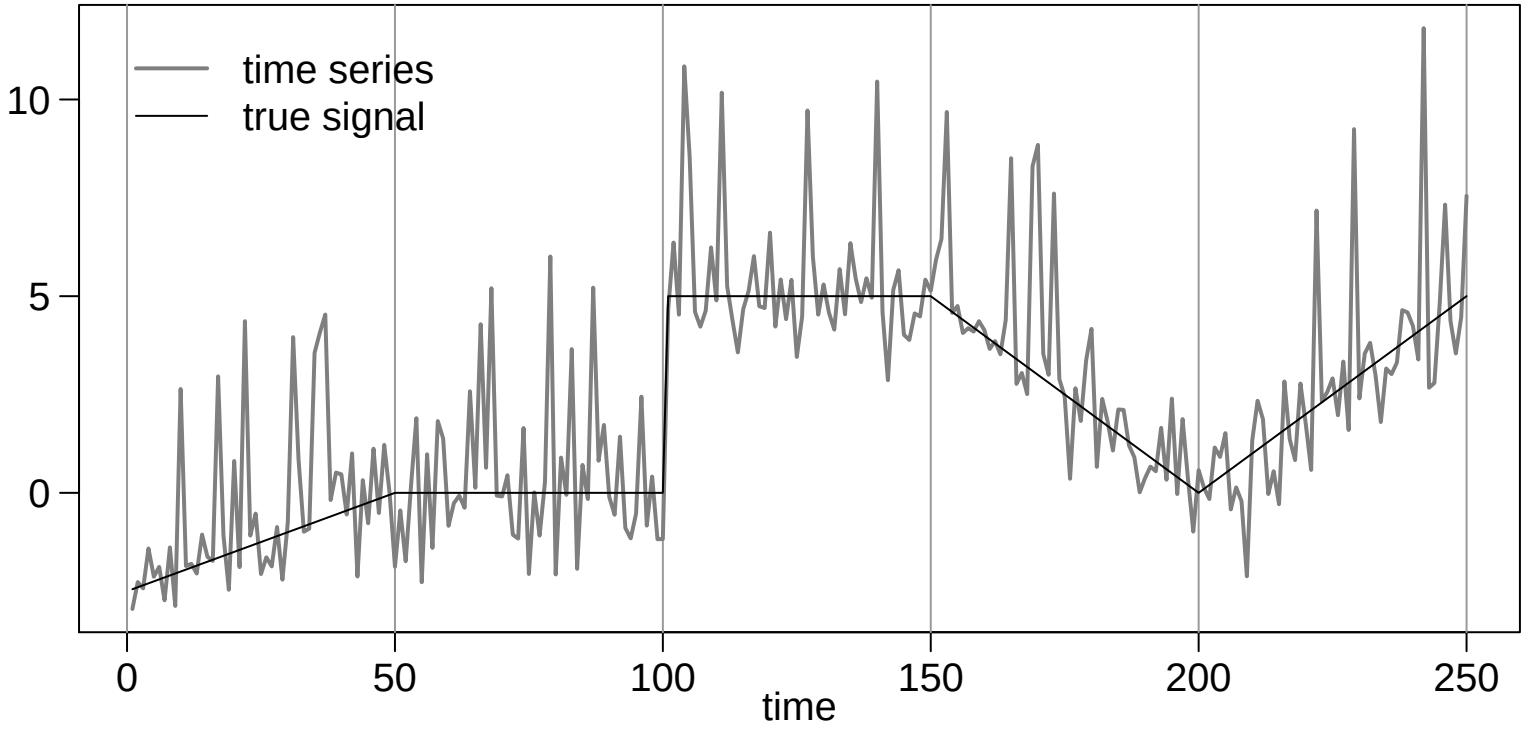
`width`	Window width $\in \mathbb{N}$	
`trend`	`"MED"`	Median
	`"RM"`	Repeated Median
	`"LMS"`	Least Median of Squares
	`"LTS"`	Least Trimmed Squares
`scale`	`"MAD"`	Median Absolute Deviation
	`"QN"`	Q_n scale
	`"SN"`	S_n scale
	`"LSH"`	Length of the Shortest Half

`robust.filter` – Parameter Options

`width`	Window width $\in \mathbb{N}$	
`trend`	`"MED"`	Median
	`"RM"`	Repeated Median
	`"LMS"`	Least Median of Squares
	`"LTS"`	Least Trimmed Squares
`scale`	`"MAD"`	Median Absolute Deviation
	`"QN"`	Q_n scale
	`"SN"`	S_n scale
	`"LSH"`	Length of the Shortest Half

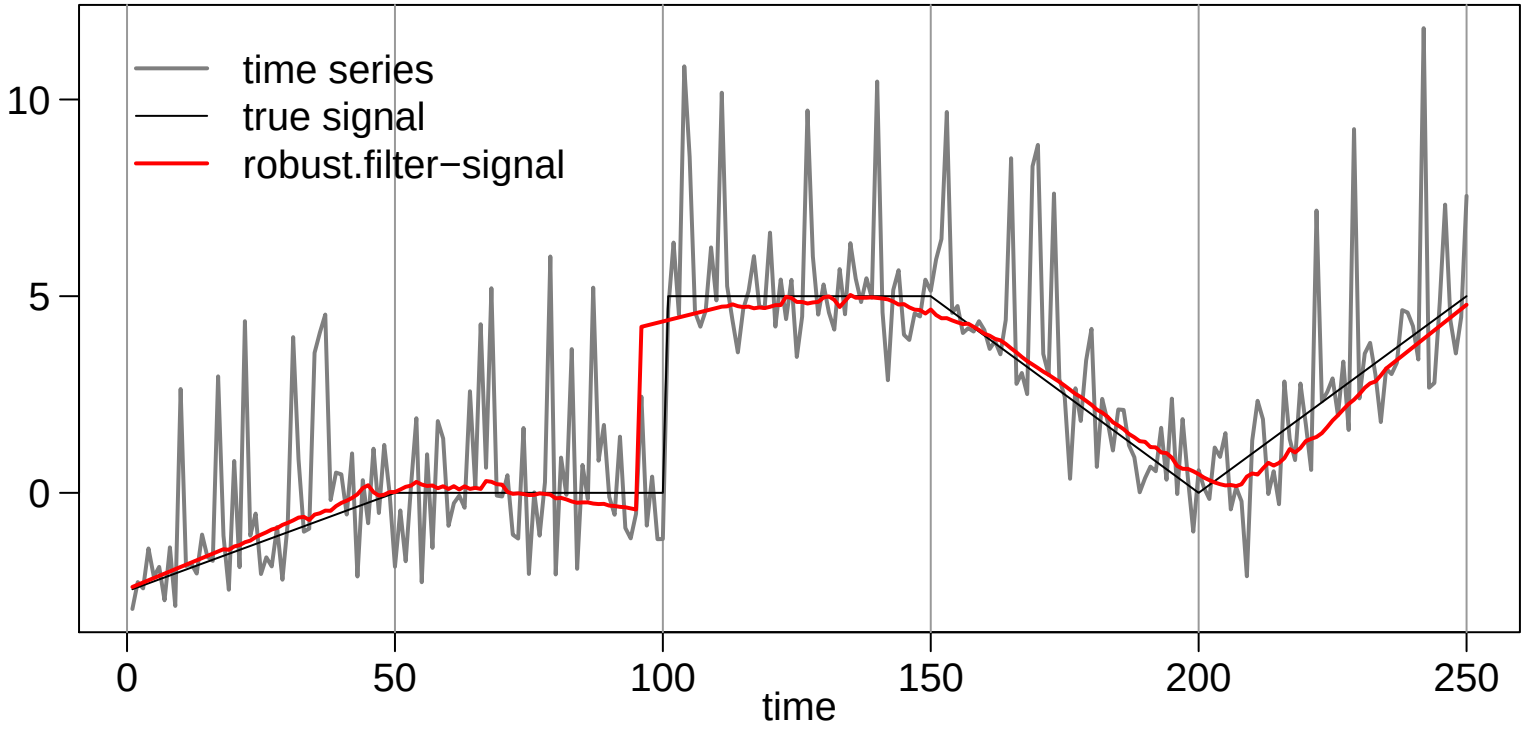
- Options `online` and `extrapolate`
- Further arguments for level shift & outlier detection and outlier treatment

`robust.filter` – Illustration



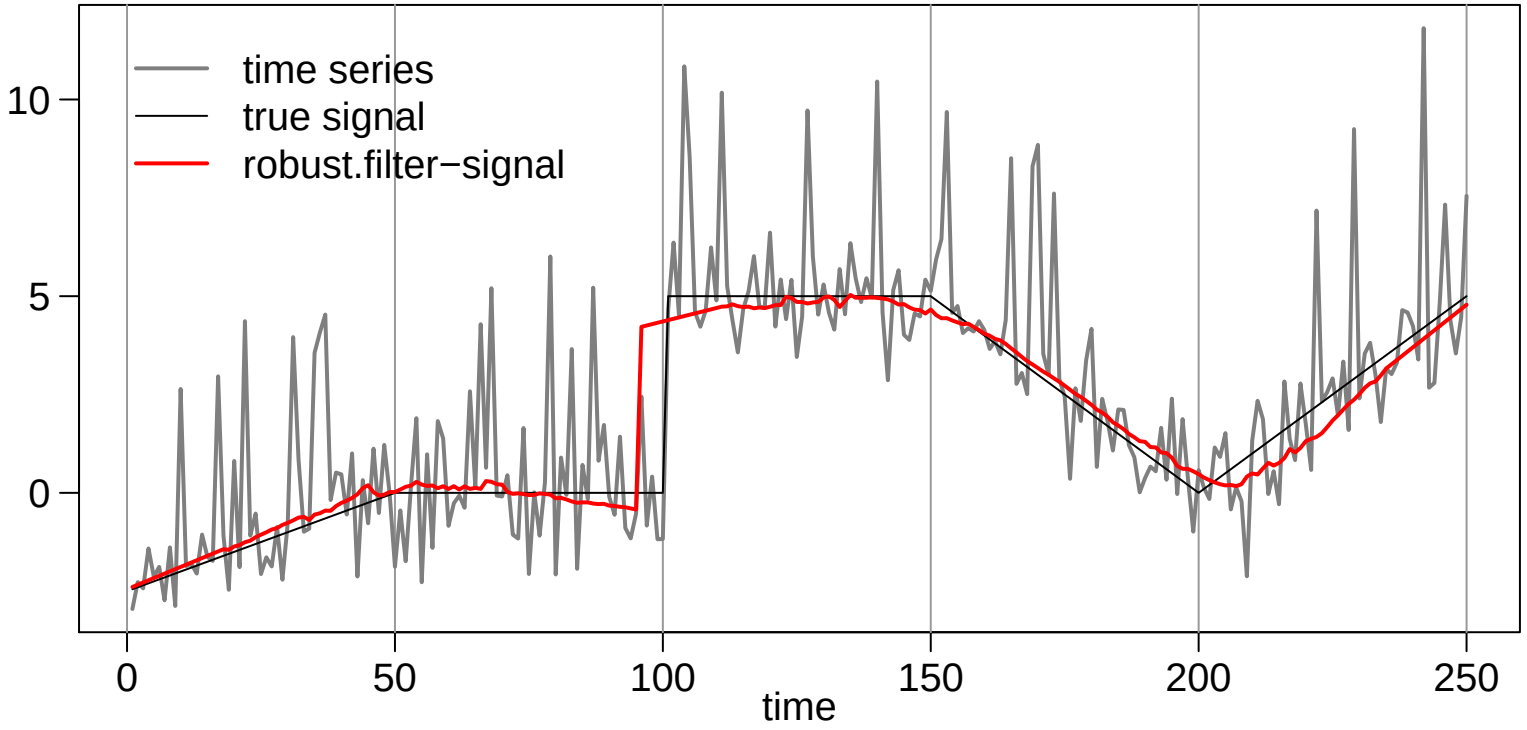

```
fit <- robust.filter(series, width=31)
```

`robust.filter` – Illustration



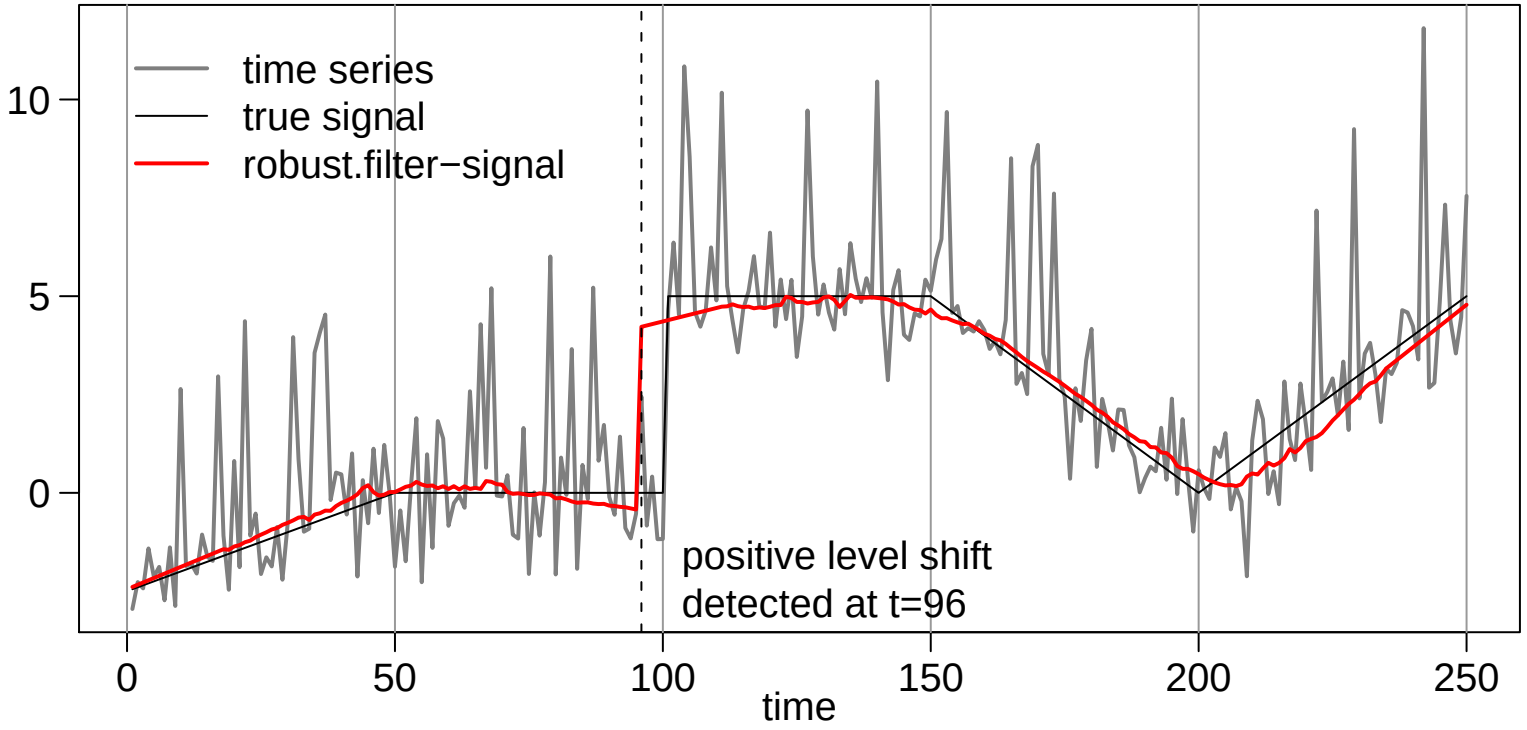

```
> fit  $level
       [1] -2.391516 -2.335128 -2.278741 -2.222354 -2.165966 ...
     $slope
       [1]  0.056387  0.056387  0.056387  0.056387  0.056387 ...
     $sigma
       [1]  0.798726 0.798726   0.798726  0.798726  0.798726 ...
     245  observations omitted
```

`robust.filter` – Illustration



Level shifts?

`robust.filter` – Illustration

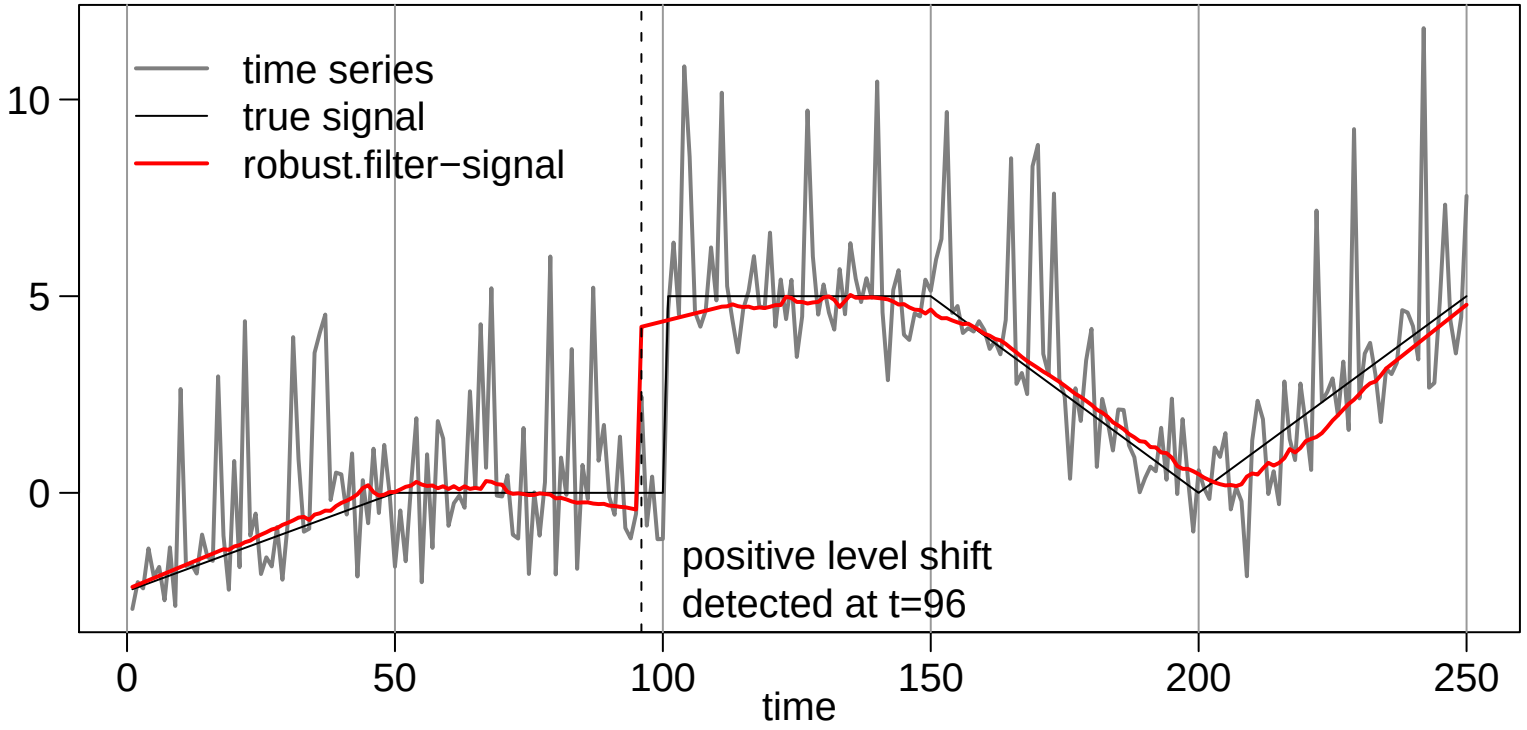


Level shifts?

```
> which(fit$level.shift!=0)
[1] 96
```

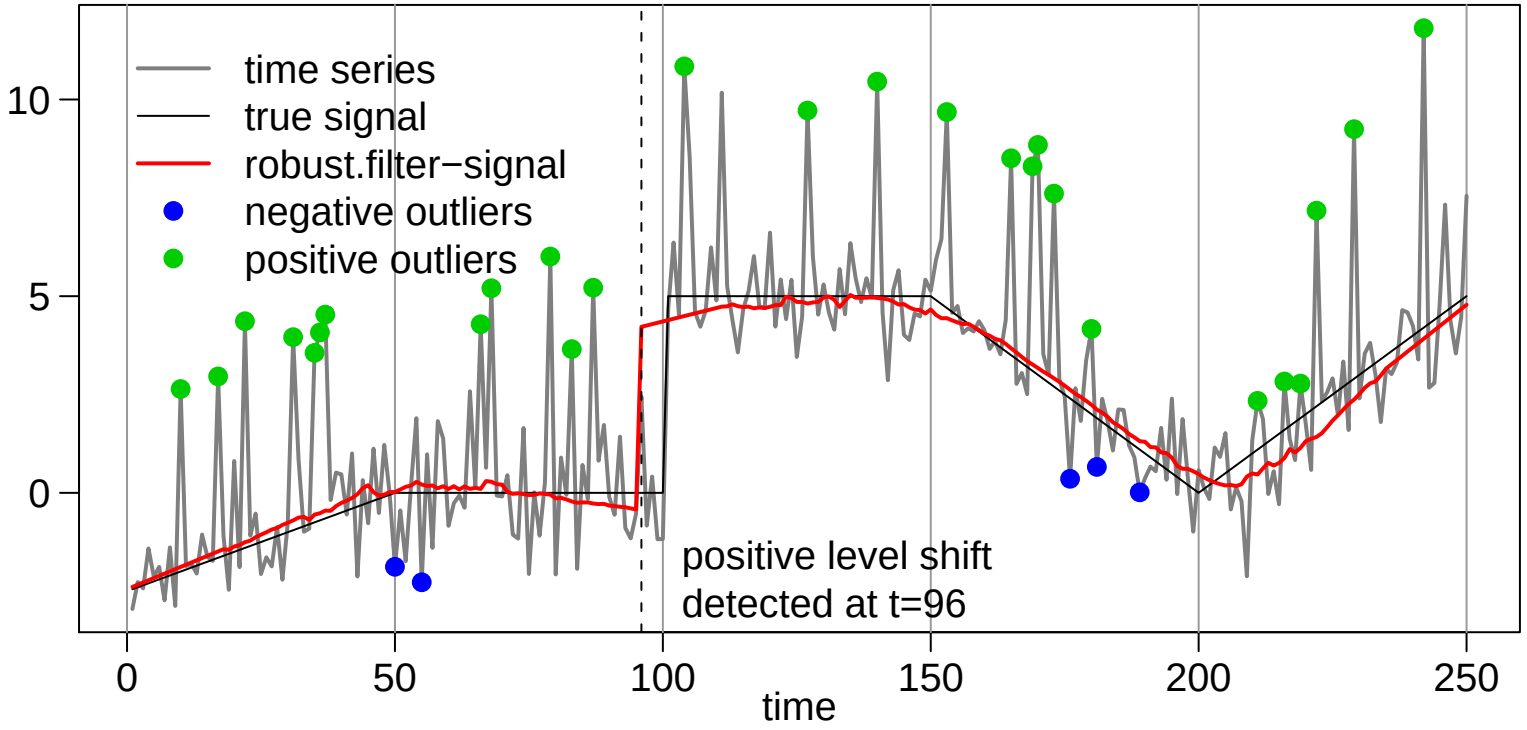
```
> fit$level.shift[which(fit$level.shift!=0)]
[1] 1
```

`robust.filter` – Illustration



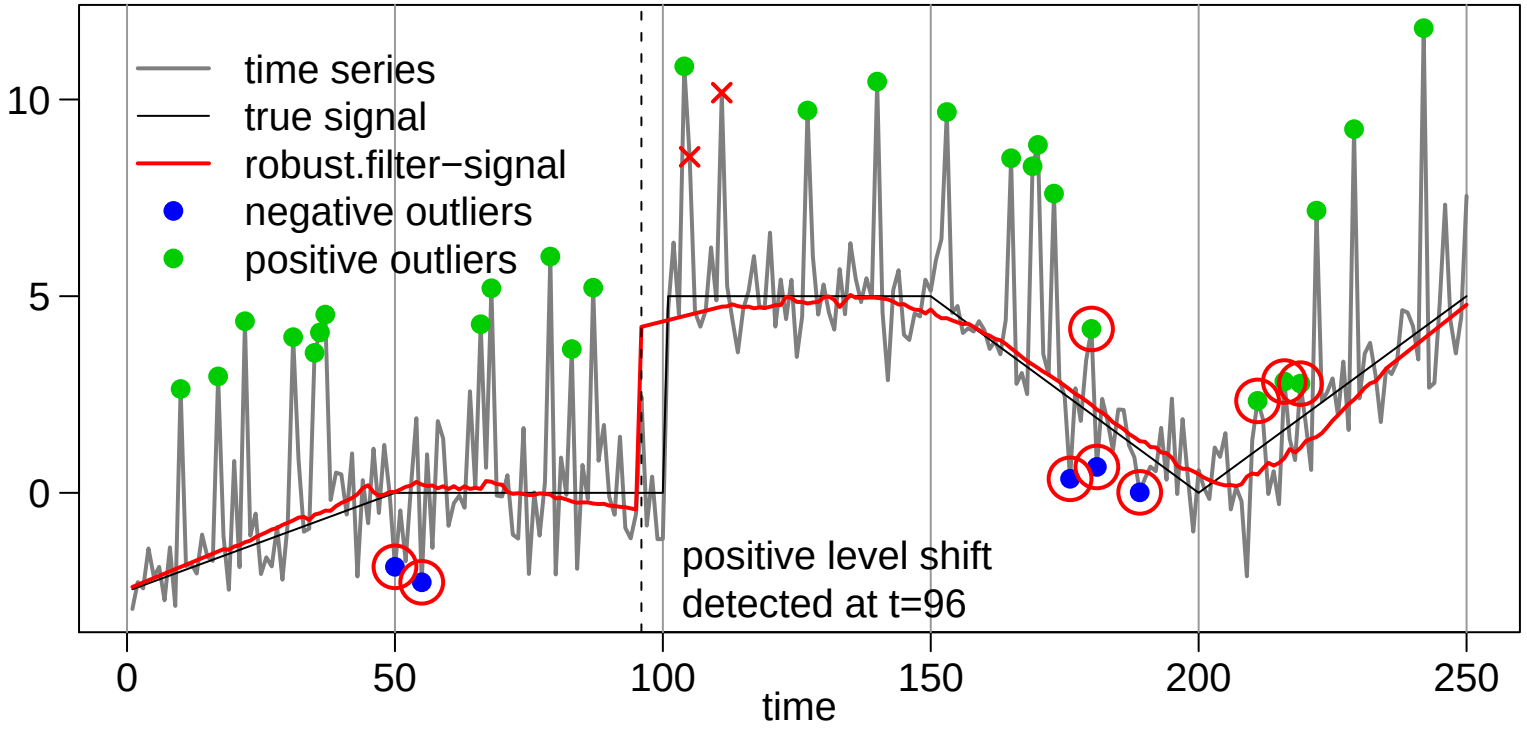
Outliers?

`robust.filter` – Illustration



Outliers?

`robust.filter` – Illustration



Outliers?

23 correctly detected

9 falsely detected

2 not detected

Weighted Repeated Median Regression

(Fried, Einbeck, Gather, 2007)

Sample in one time window $\mathbf{y}_t = (y_{t+i})'$

$$\widehat{\beta}_t^{\mathsf{RM}} = \underset{i}{\operatorname{med}} \left\{ \underset{j \neq i}{\operatorname{med}} \left\{ \frac{y_{t+i} - y_{t+j}}{i - j} \right\} \right\}$$

$$\widehat{\mu}_t^{\mathsf{RM}} = \underset{i}{\operatorname{med}} \left\{ (y_{t+i} + \widehat{\beta}_t^{\mathsf{RM}} \cdot i) \right\}$$

with $i = \begin{cases} -n+1, \ldots, -1, 0 & \text{(online)} \\ -m, \ldots, m \text{ and } n = 2m+1 & \text{(with delay)} \end{cases}$

Weighted Repeated Median Regression

(Fried, Einbeck, Gather, 2007)

Sample in one time window $\mathbf{y}_t = (y_{t+i})'$

$$\widehat{\beta}_t^{\text{RM}} = \underset{i}{\text{med}}\; w_i \diamond \left\{ \underset{j \neq i}{\text{med}}\; w_j \diamond \left\{ \frac{y_{t+i} - y_{t+j}}{i - j} \right\} \right\}$$

$$\widehat{\mu}_t^{\text{RM}} = \underset{i}{\text{med}} \left\{ w_i \diamond (y_{t+i} + \widehat{\beta}_t^{\text{RM}} \cdot i) \right\}$$

with $i = \begin{cases} -n+1, \ldots, -1, 0 & \text{(online)} \\ -m, \ldots, m \;\text{ and }\; n = 2m+1 & \text{(with delay)} \end{cases}$

$w \diamond a$ denotes the set of w replications of a: $3 \diamond a = \{a, a, a\}$.

Weighted Repeated Median Regression

(Fried, Einbeck, Gather, 2007)

Generally: Sample $(\mathbf{x}, \mathbf{y}) = (x_i, y_i)', \quad i = 1, \ldots, n$

$$\widehat{\beta}^{\mathsf{RM}} = \underset{i}{\operatorname{med}}\, w_i \diamond \left\{ \underset{j \neq i}{\operatorname{med}}\, w_j \diamond \left\{ \frac{y_i - y_j}{x_i - x_j} \right\} \right\}$$

$$\widehat{\mu}^{\mathsf{RM}} = \underset{i}{\operatorname{med}} \left\{ w_i \diamond (y_i + \widehat{\beta}^{\mathsf{RM}} \cdot x_i) \right\}$$

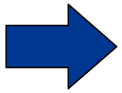
Weighted Repeated Median Regression

(Fried, Einbeck, Gather, 2007)

Generally: Sample $(\mathbf{x}, \mathbf{y}) = (x_i, y_i)', \quad i = 1, \ldots, n$

$$\widehat{\beta}^{\mathsf{RM}} = \underset{i}{\operatorname{med}}\, w_i \diamond \left\{ \underset{j \neq i}{\operatorname{med}}\, w_j \diamond \left\{ \frac{y_i - y_j}{x_i - x_j} \right\} \right\}$$

$$\widehat{\mu}^{\mathsf{RM}} = \underset{i}{\operatorname{med}} \left\{ w_i \diamond (y_i + \widehat{\beta}^{\mathsf{RM}} \cdot x_i) \right\}$$



`wrm.smooth`:

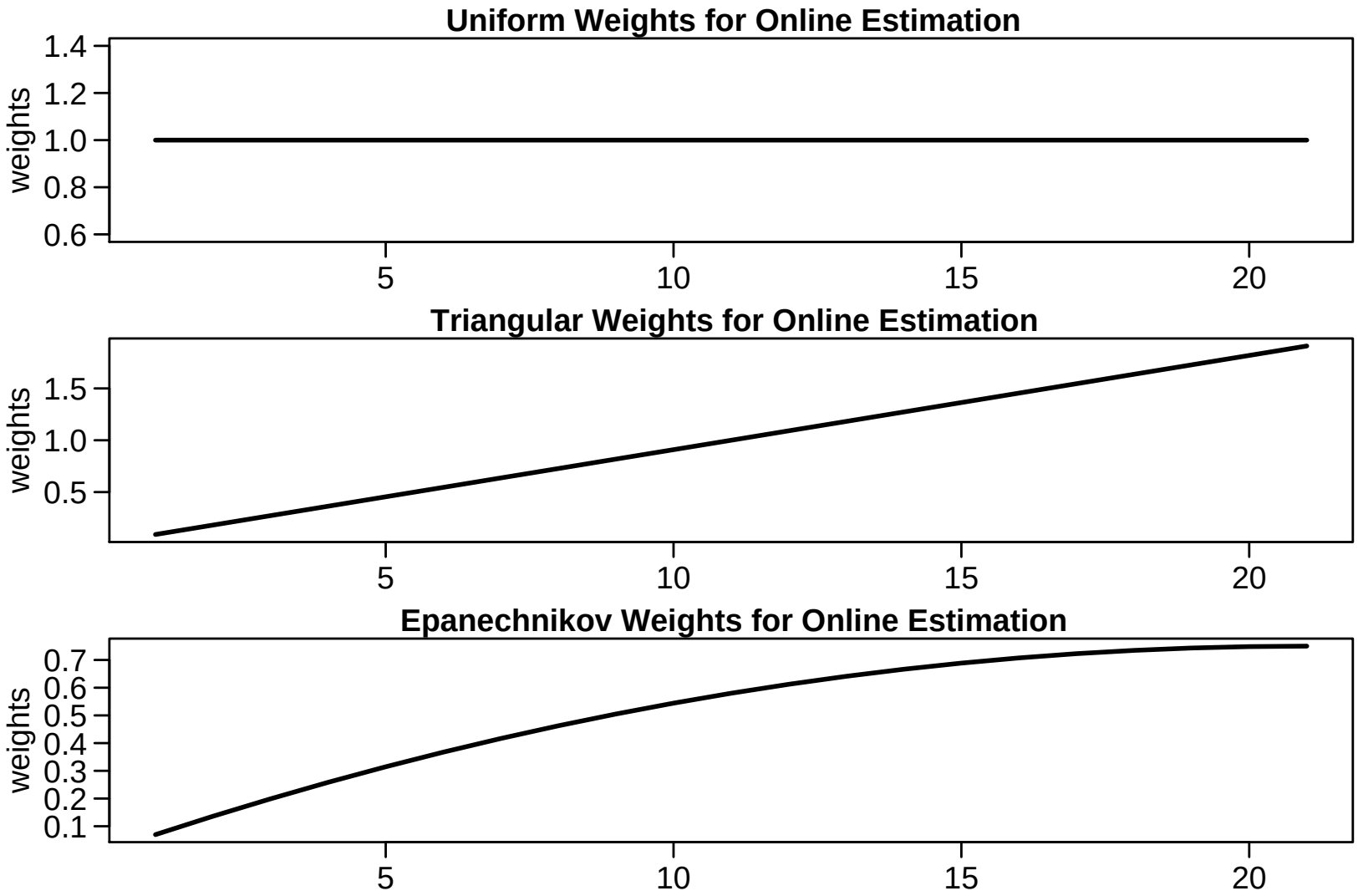
- Non-equidistant regressor variables
- Smoothing with kernel weight functions

`wrm.filter` – Parameter Options

`y`	input time series (`ts`-object or vector)
`width`	window width ($\in \mathbb{N}$)
`del`	delay of the extracted signal: `del=0` means 'online', default is `del=floor(width/2)` (delayed)
`weight`	weight function: 0: equal weighting 1: triangular weights 2: Epanechnikov weights

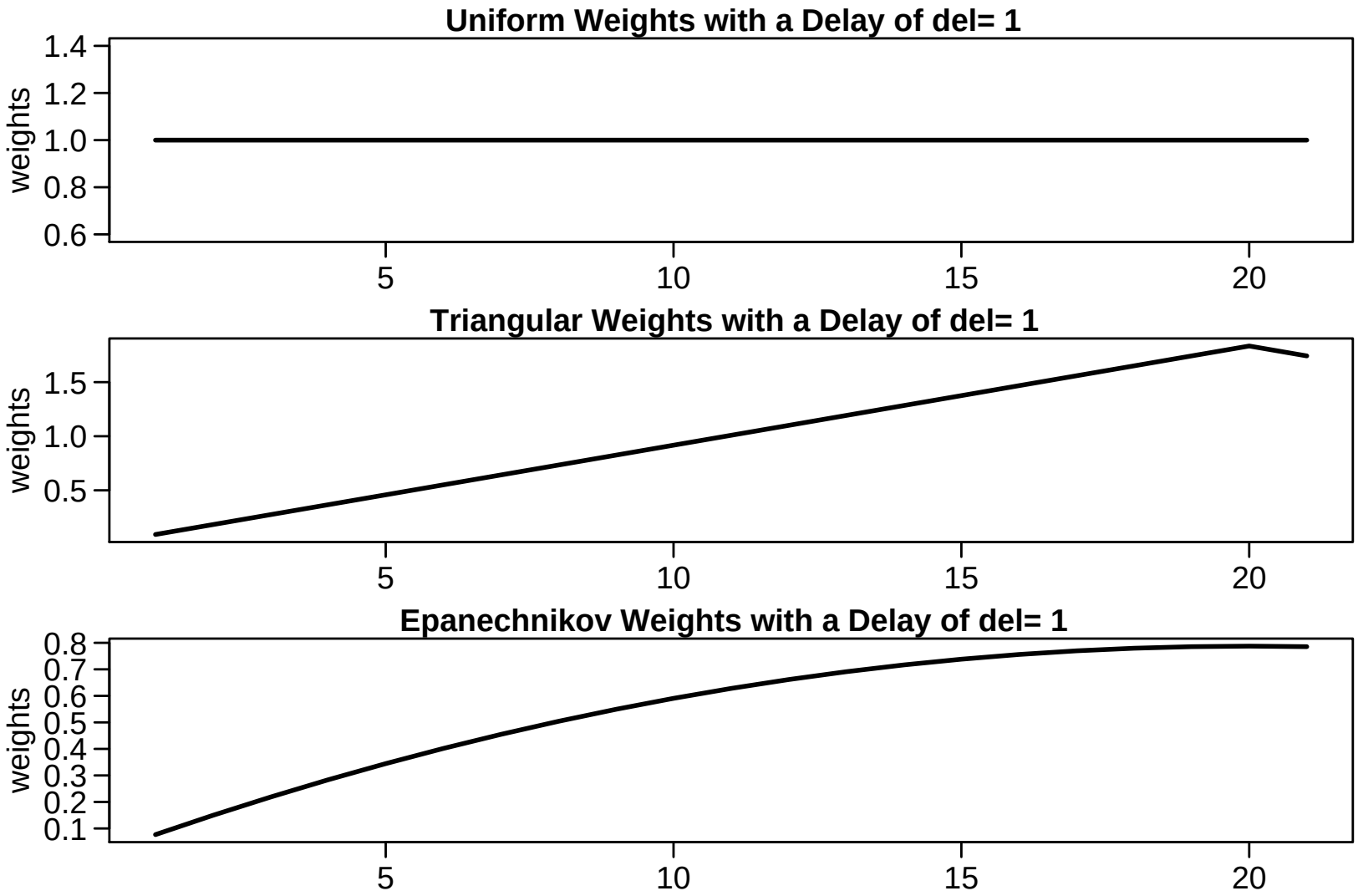
`wrm.filter` – Weight Functions

Window `width = 21`



`wrm.filter` – Weight Functions

Window `width = 21`



`wrm.filter` – Weight Functions

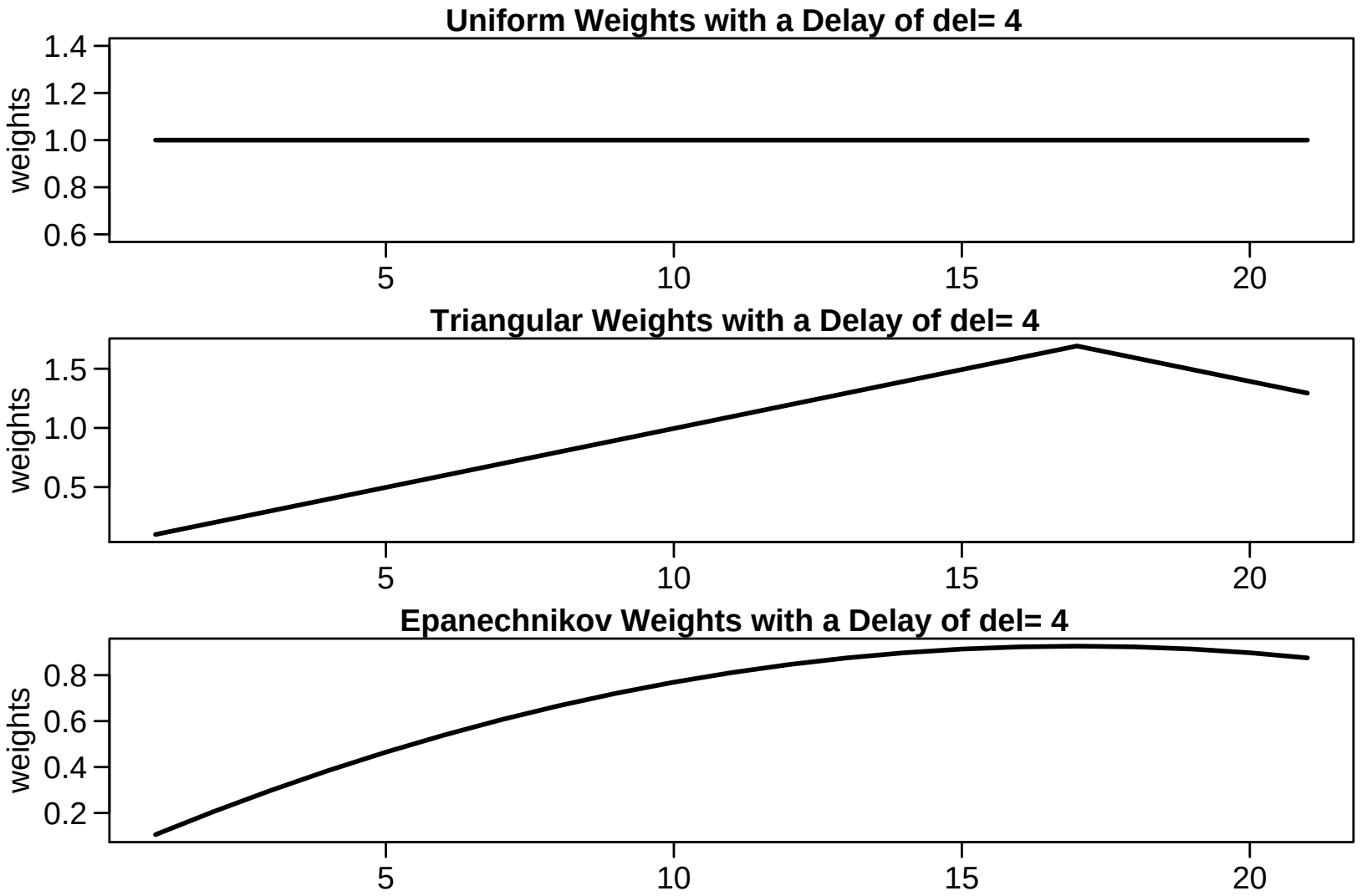
Window `width = 21`

`wrm.filter` – Weight Functions

Window `width = 21`

`wrm.filter` – Weight Functions

Window `width = 21`

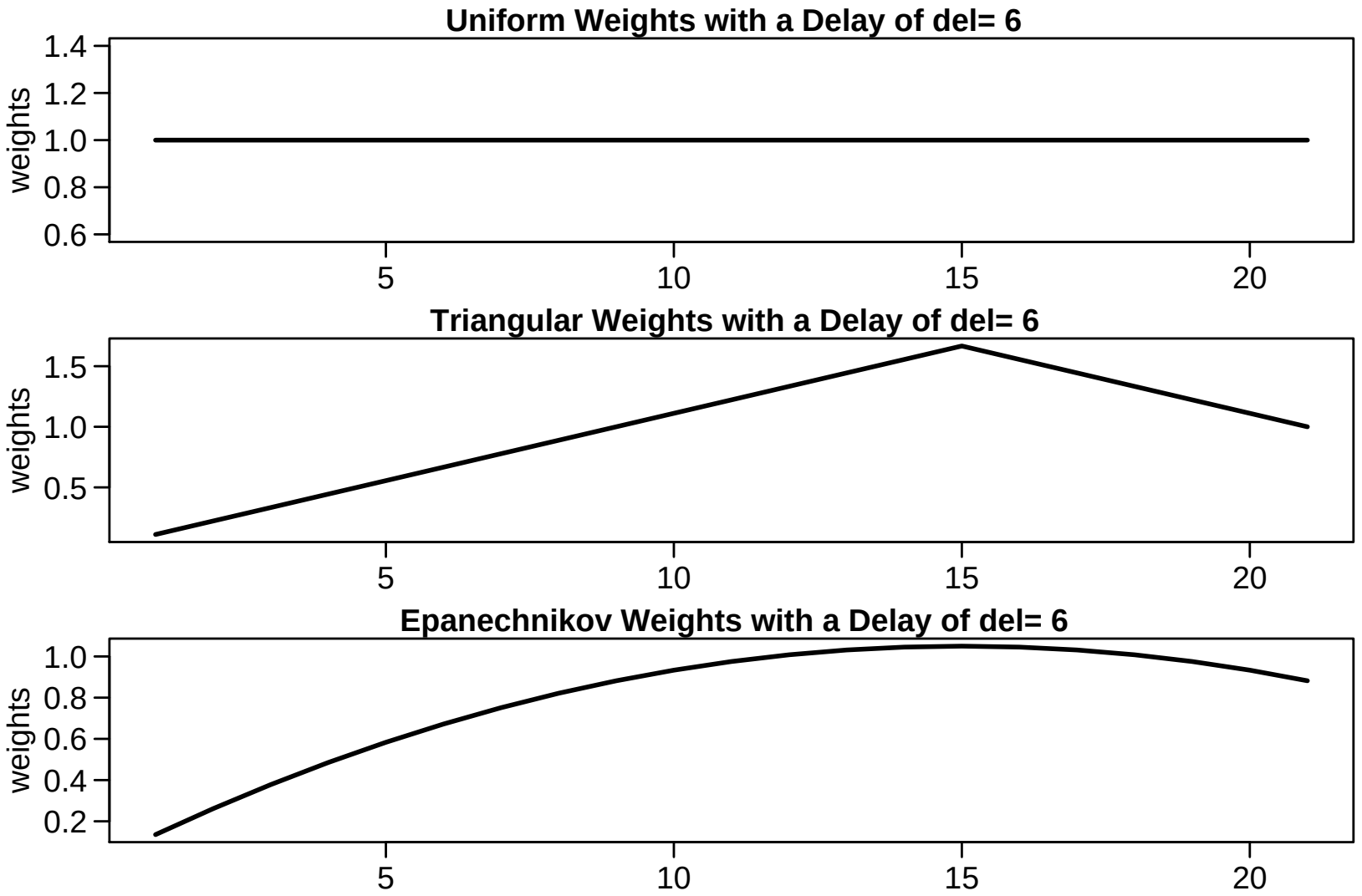


`wrm.filter` – Weight Functions

Window `width = 21`

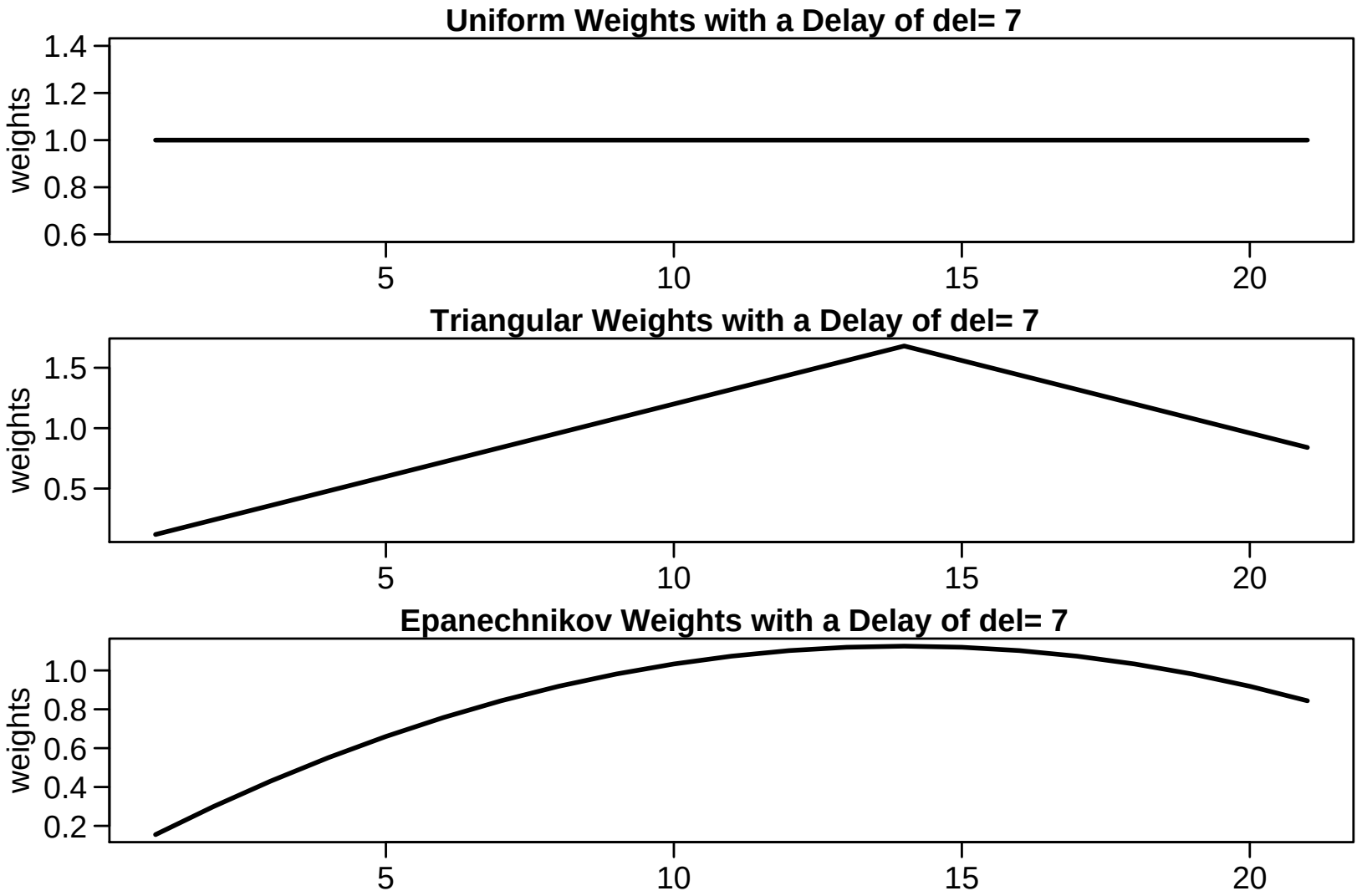
`wrm.filter` – Weight Functions

Window `width = 21`



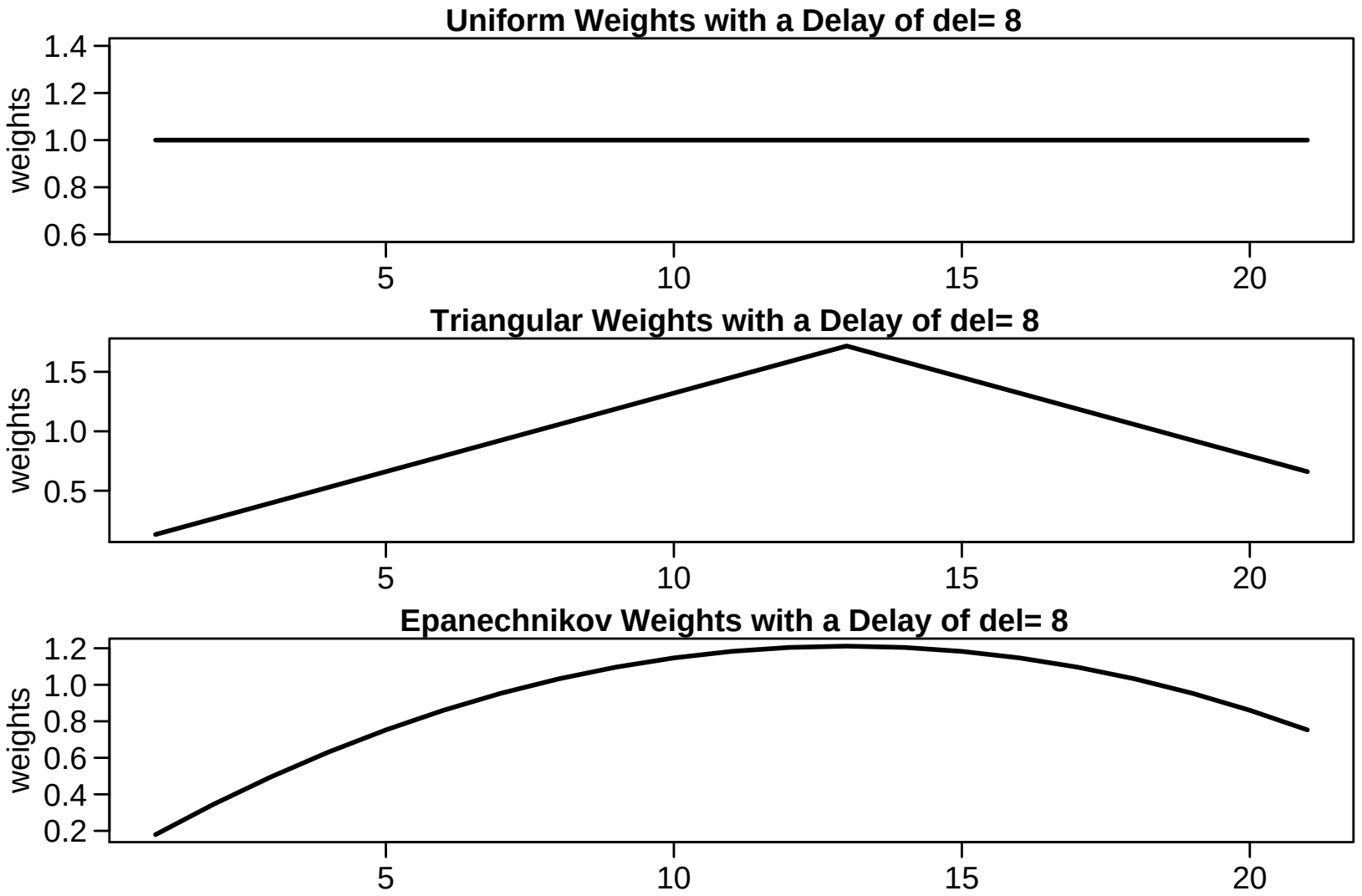
`wrm.filter` – Weight Functions

Window `width = 21`



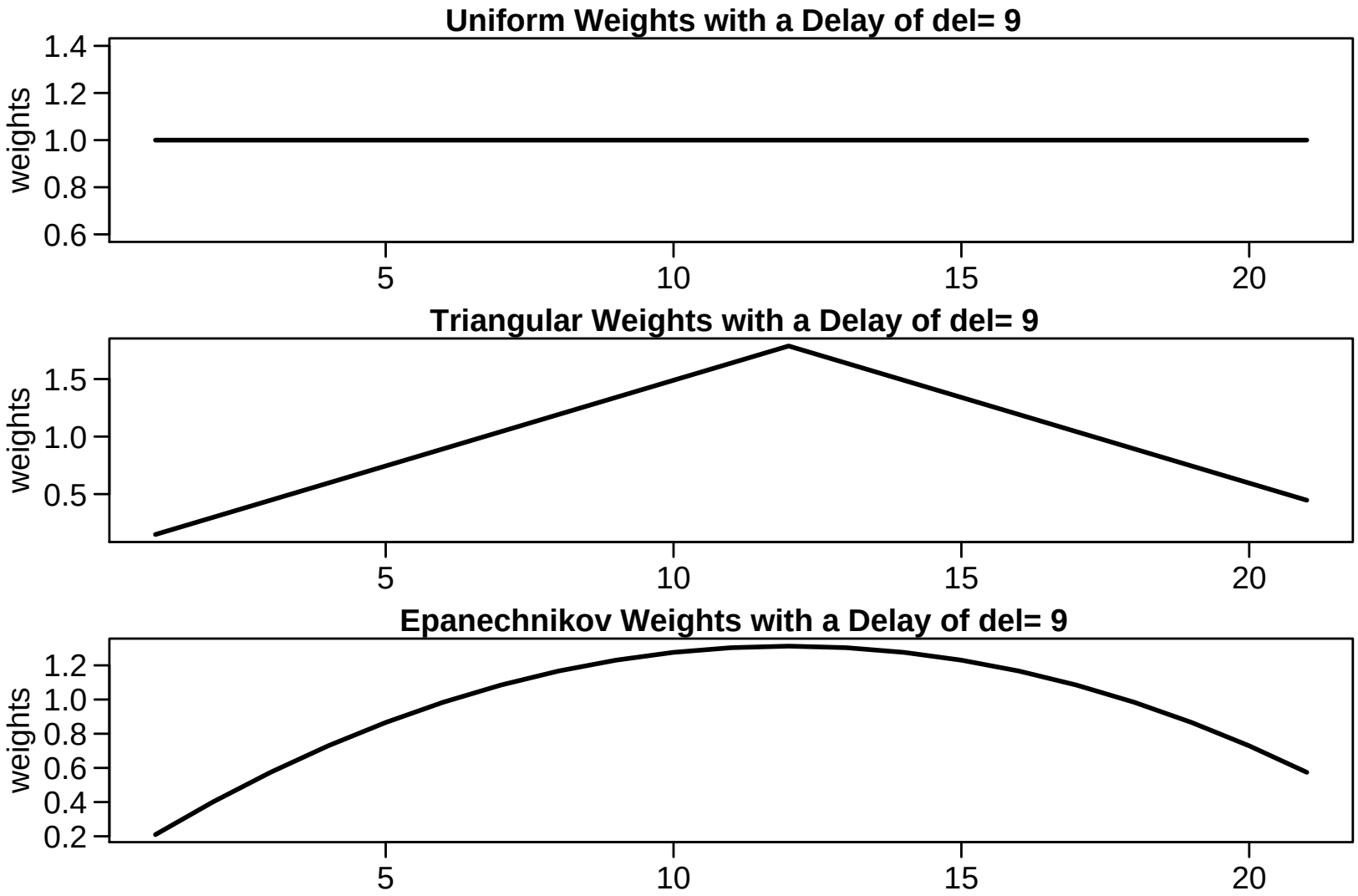
`wrm.filter` – Weight Functions

Window `width = 21`



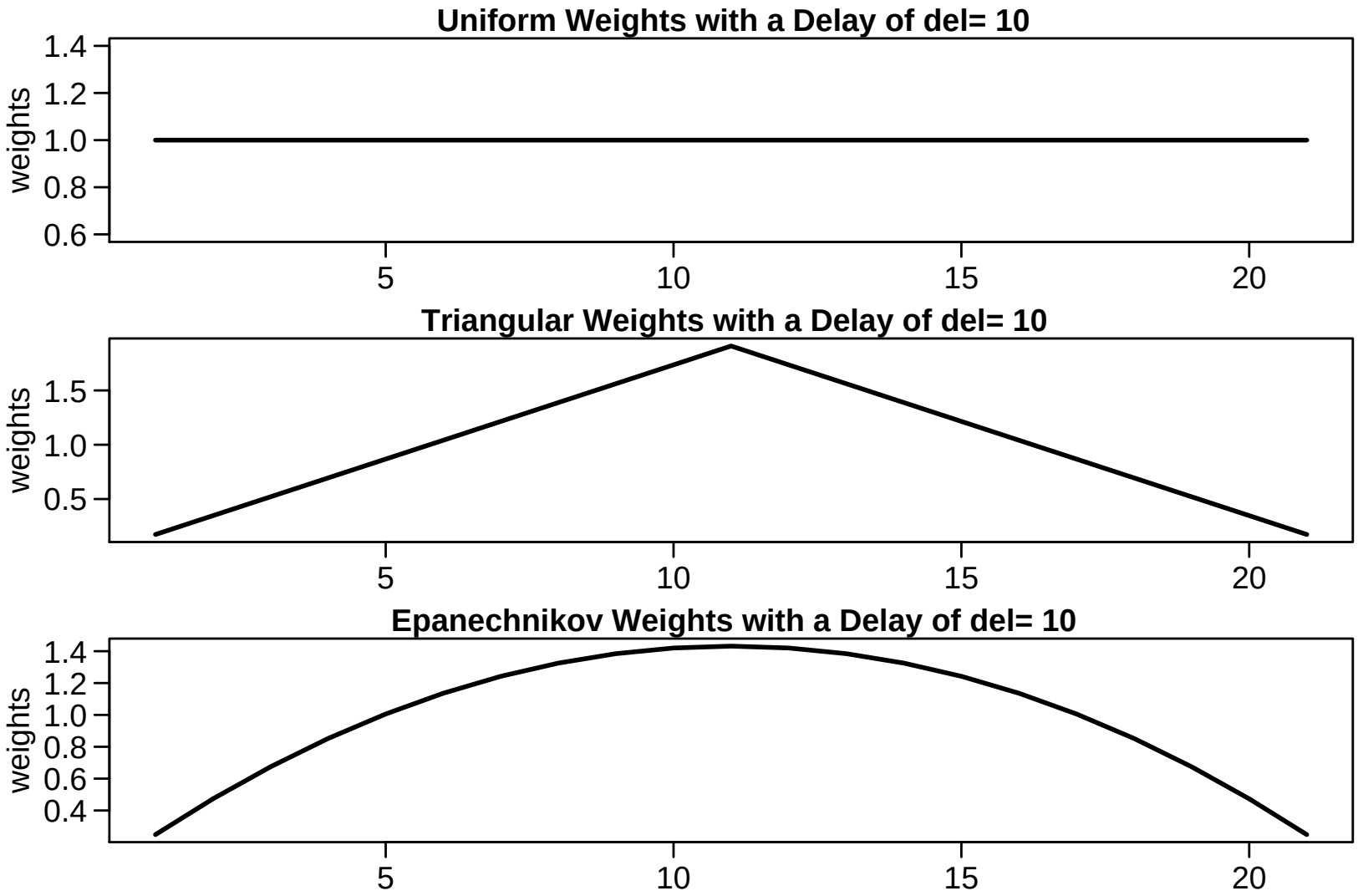
`wrm.filter` – Weight Functions

Window `width = 21`

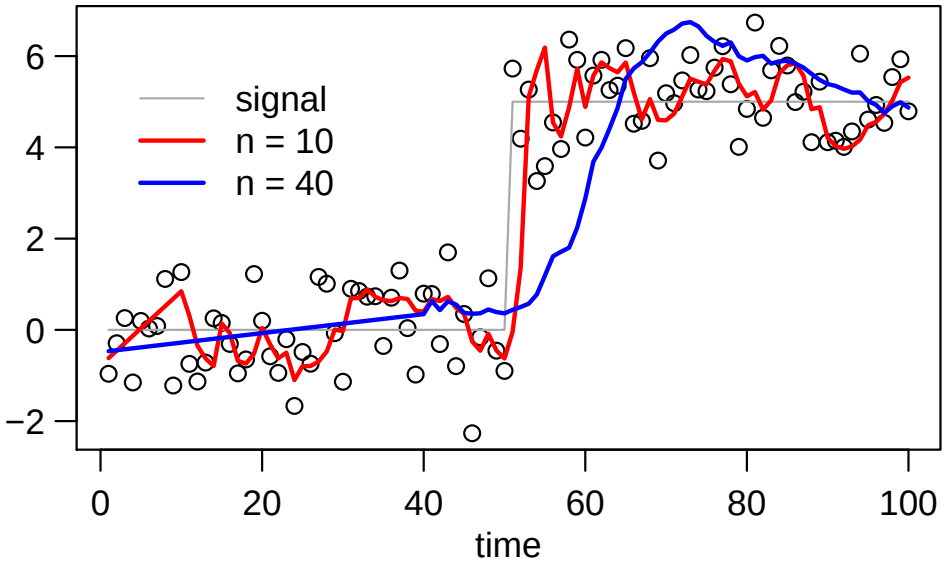


`wrm.filter` – Weight Functions

Window `width = 21`



Influence of the Window Width



n small

- \+ small bias
- \+ adapts quickly to changes
- \+ short computation time

⇔

n large

- \+ small variance
- \+ smooth
- \+ robust

➡ **Data adaptive choice of window width**

adore.filter

(Schettlinger, Fried, Gather, 2008)

adaptive **o**nline **re**peated median filtering:

Window width adaptation by a test
using the 'balance' of the residual signs $\left(\sum_{i=1}^{n} \text{sign}(r_{t,i}) = 0 \right)$

RM approximation in current time window

↓

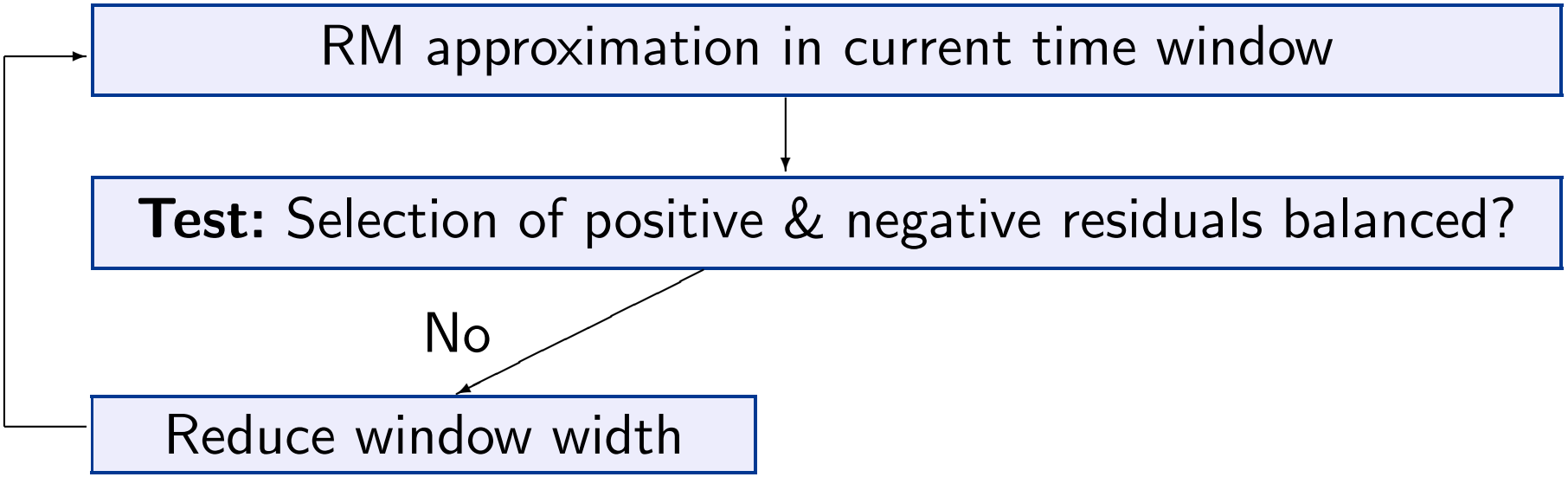
Test: Selection of positive & negative residuals balanced?

adore.filter

(Schettlinger, Fried, Gather, 2008)

adaptive **o**nline **re**peated median filtering:

Window width adaptation by a test
using the 'balance' of the residual signs $\left(\sum_{i=1}^{n} \text{sign}(r_{t,i}) = 0 \right)$

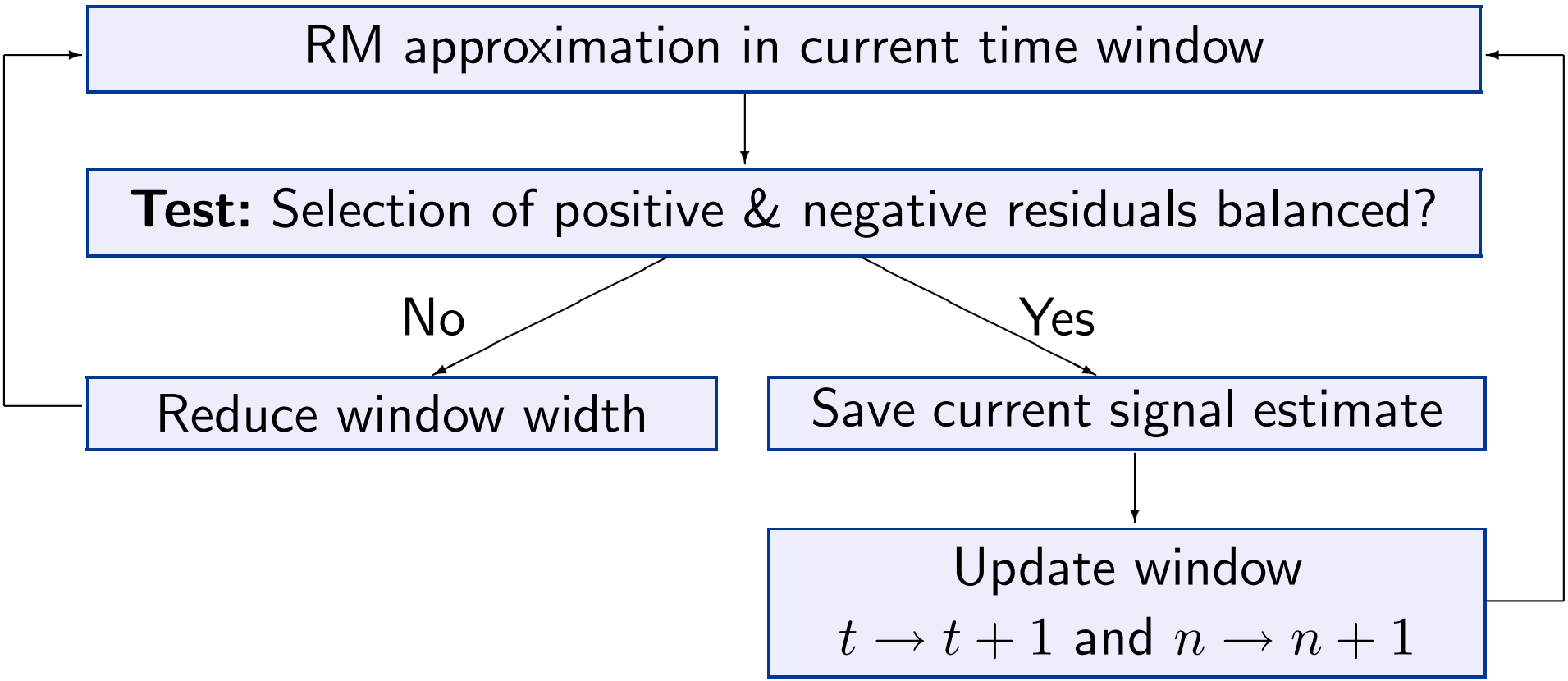


adore.filter

(Schettlinger, Fried, Gather, 2008)

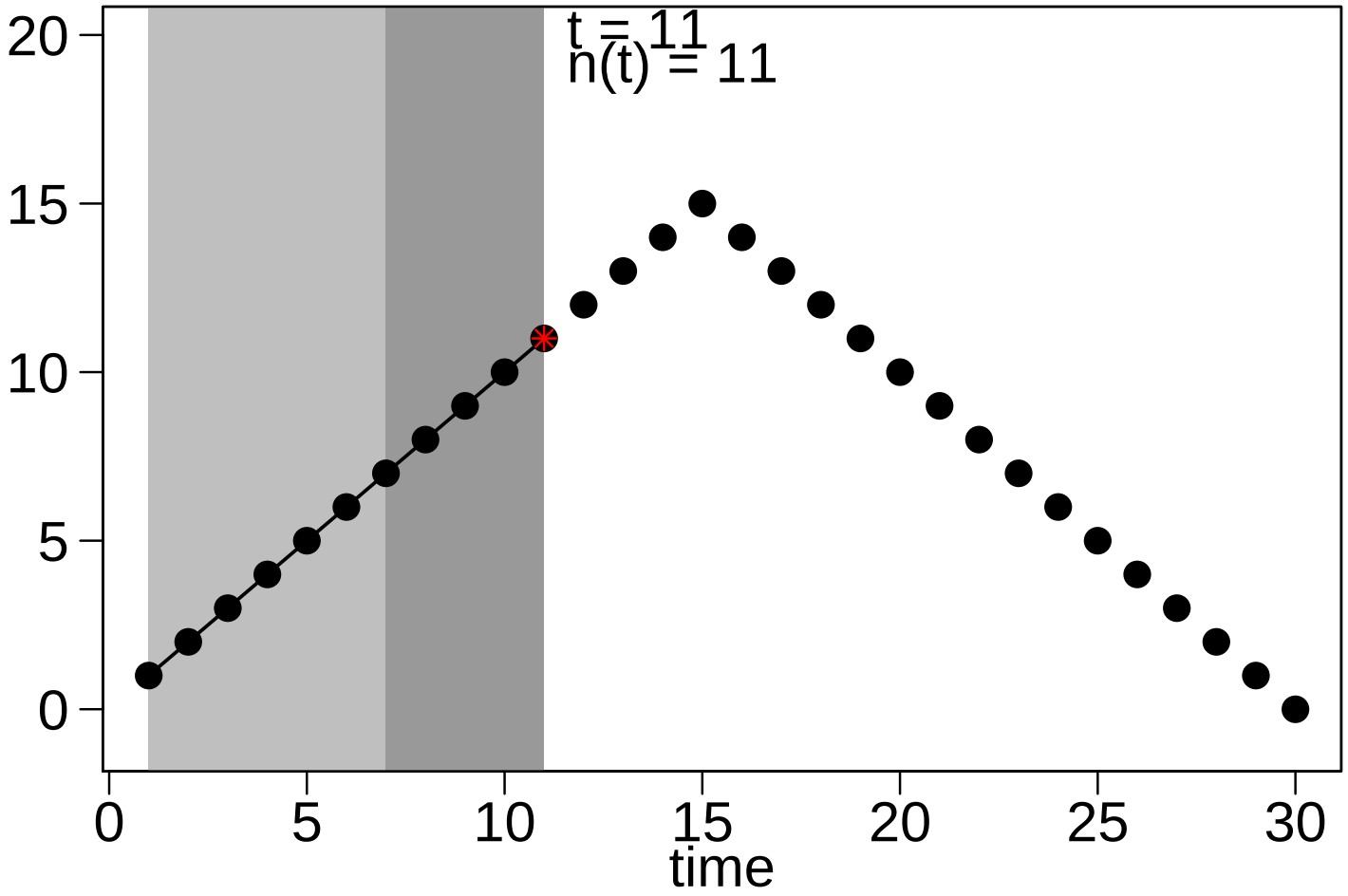
adaptive **o**nline **re**peated median filtering:

Window width adaptation by a test
using the 'balance' of the residual signs $\left(\sum_{i=1}^{n} \text{sign}(r_{t,i}) = 0 \right)$



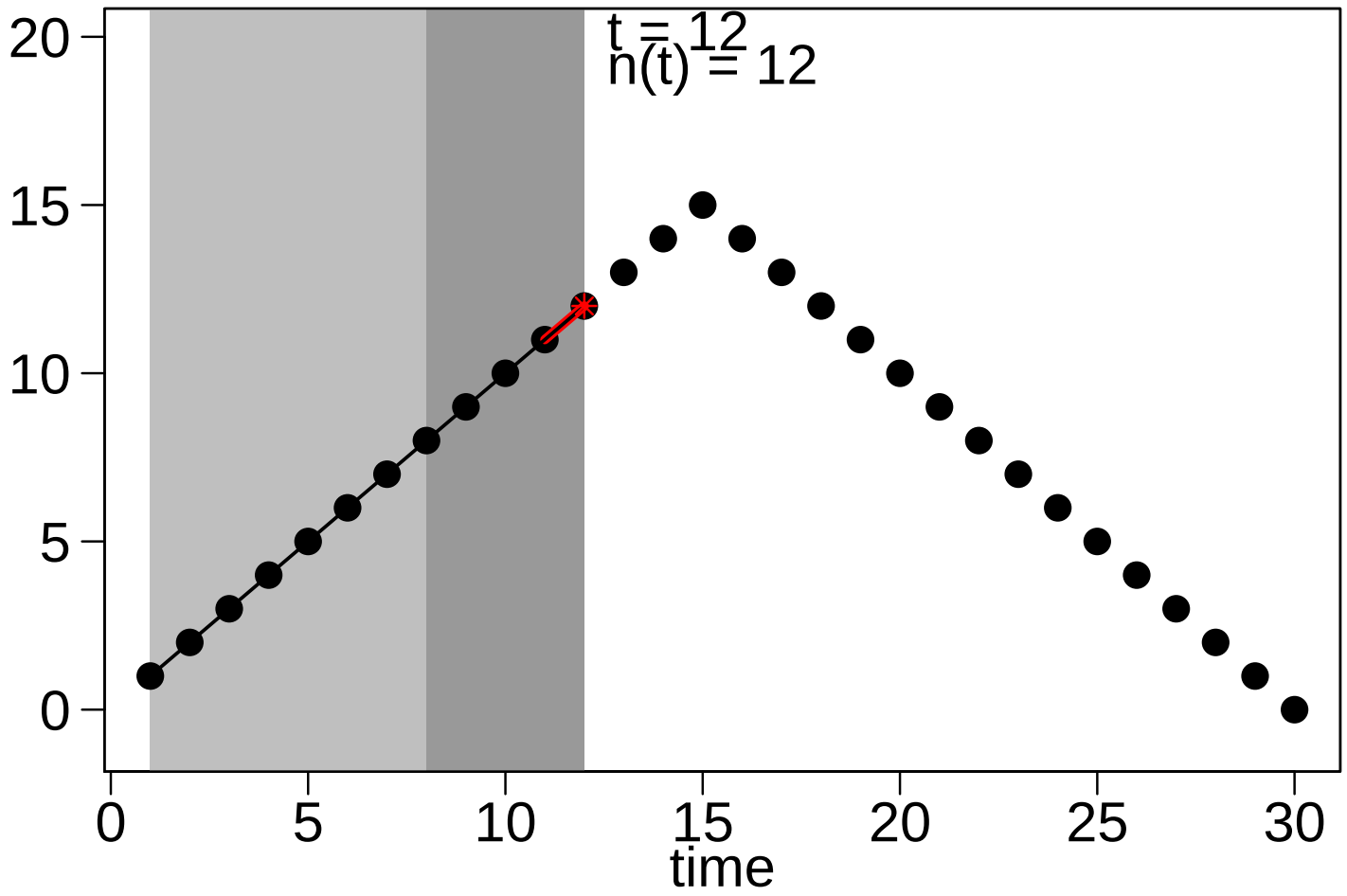
adore.filter – Illustration

```
min.width=11    p.test=5    max.width=100
```



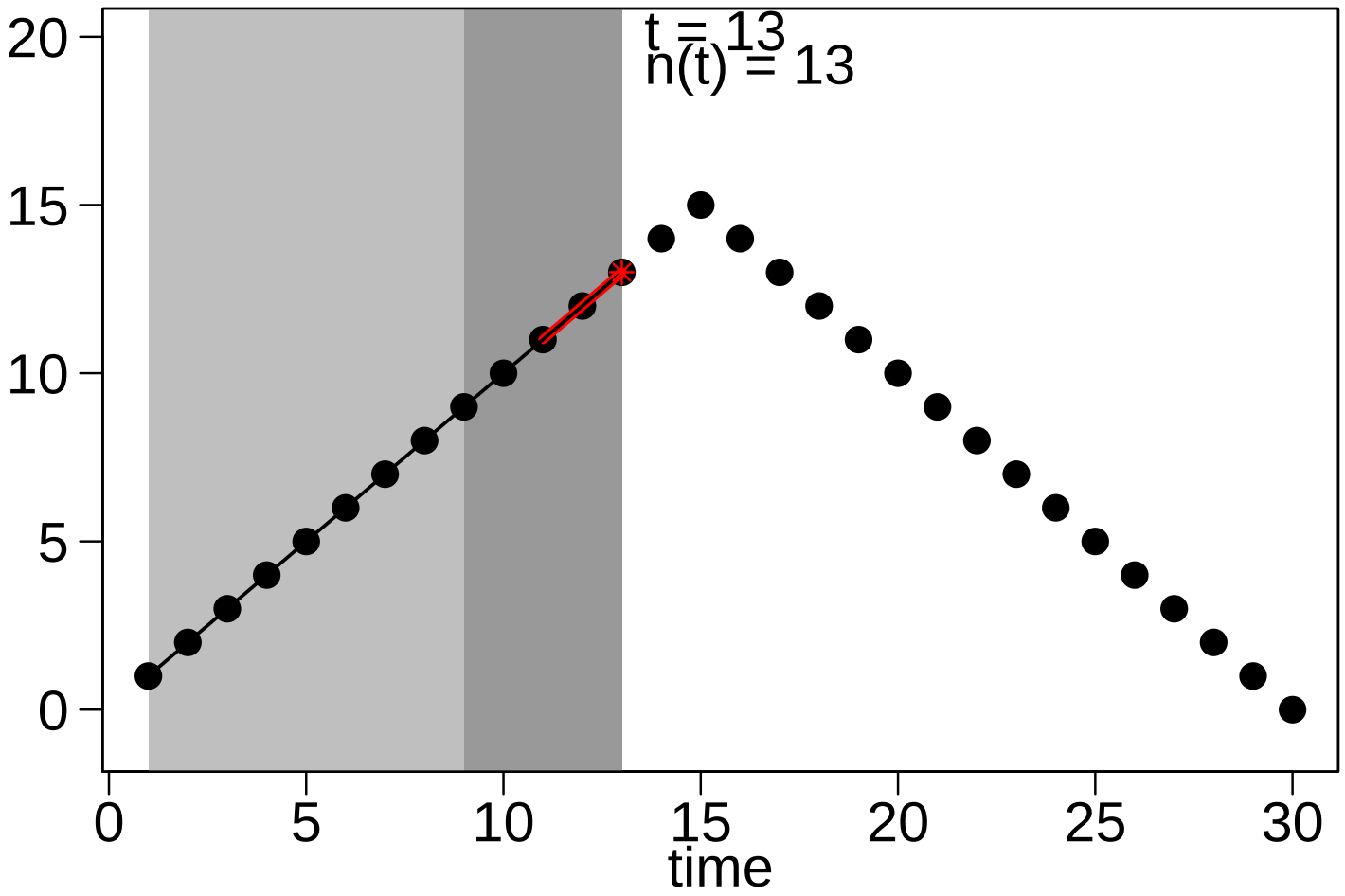
adore.filter – Illustration

`min.width=11  p.test=5  max.width=100`



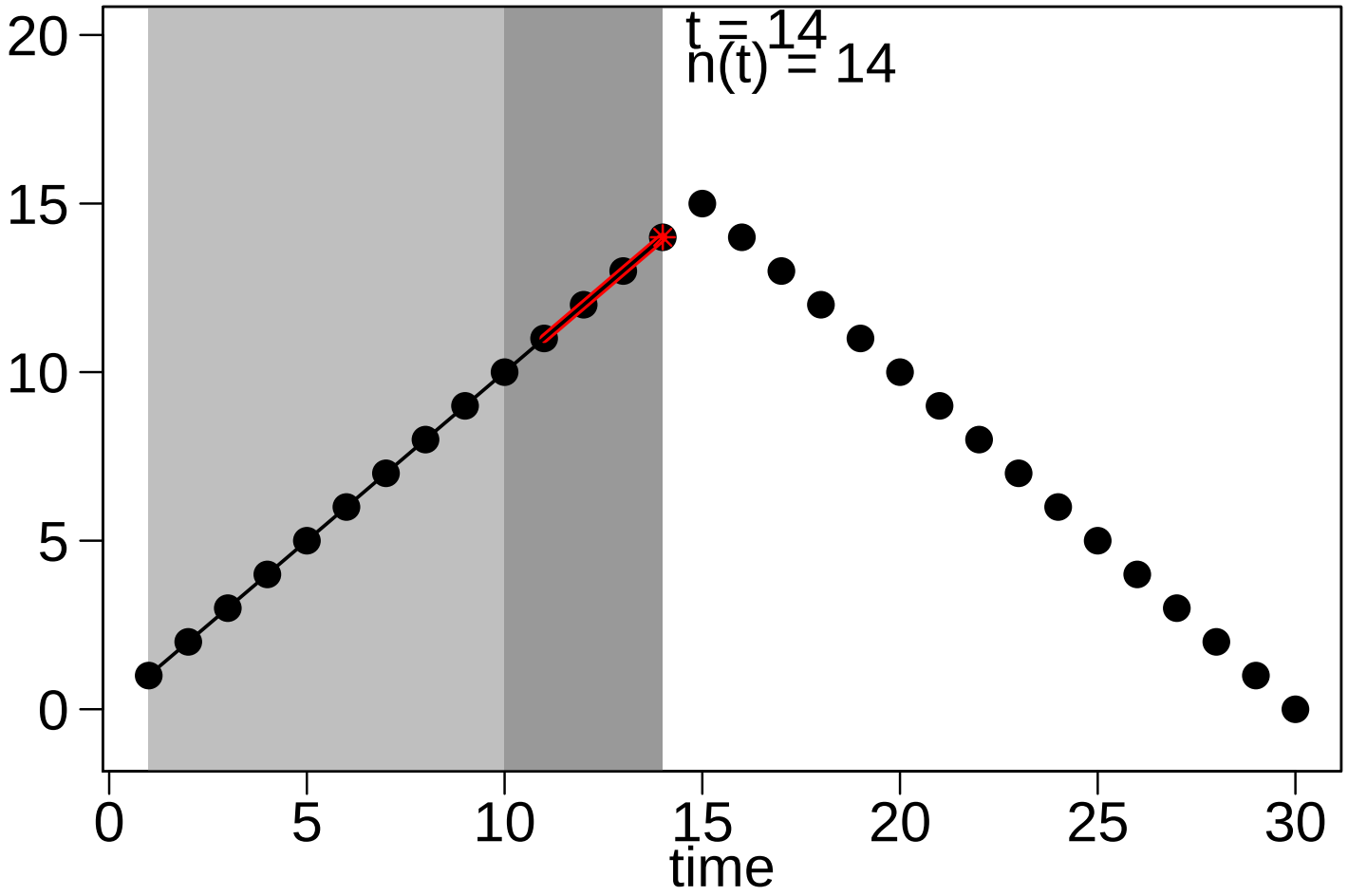
adore.filter – Illustration

```
min.width=11    p.test=5    max.width=100
```



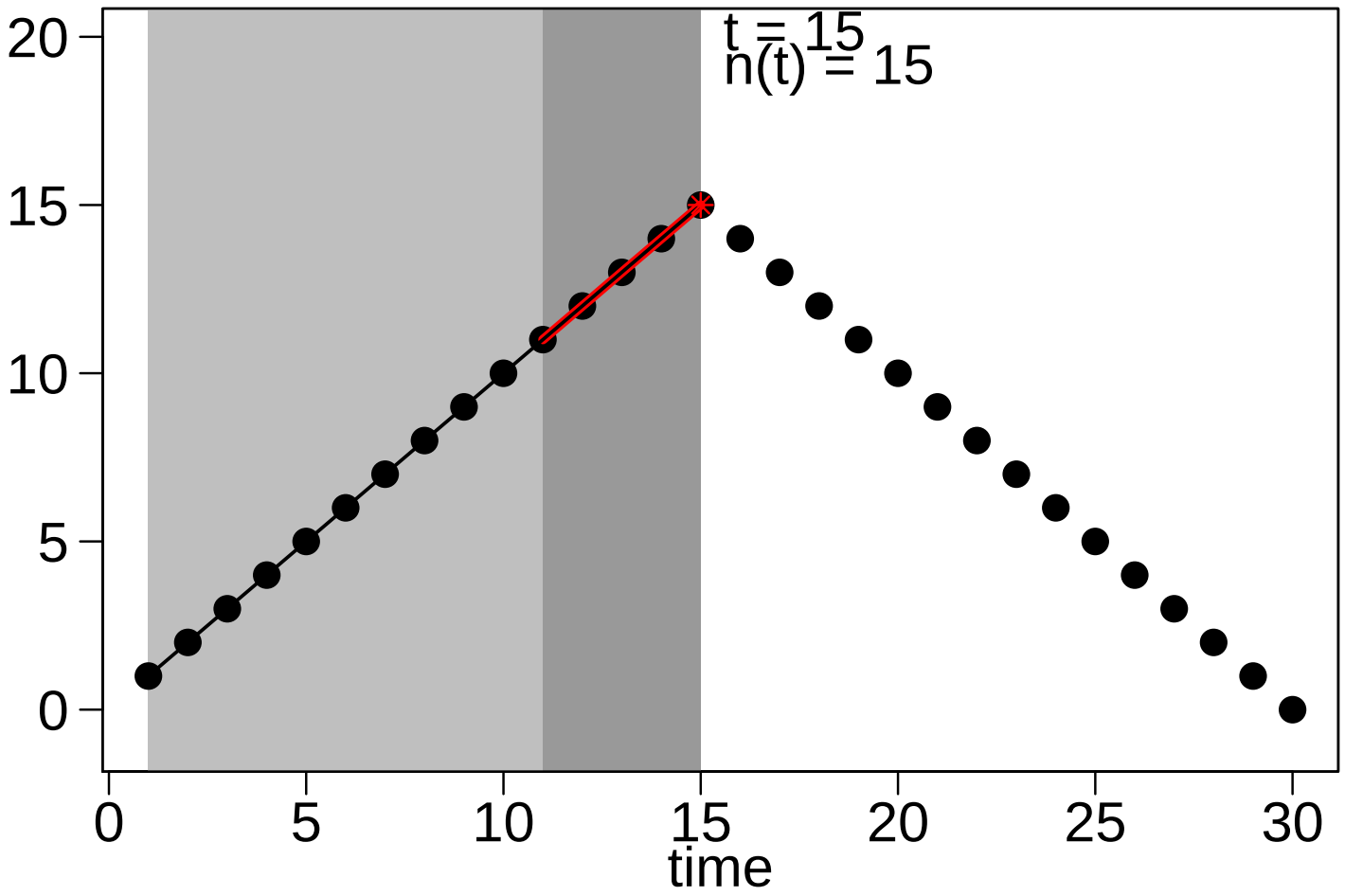
adore.filter – Illustration

```
min.width=11    p.test=5    max.width=100
```



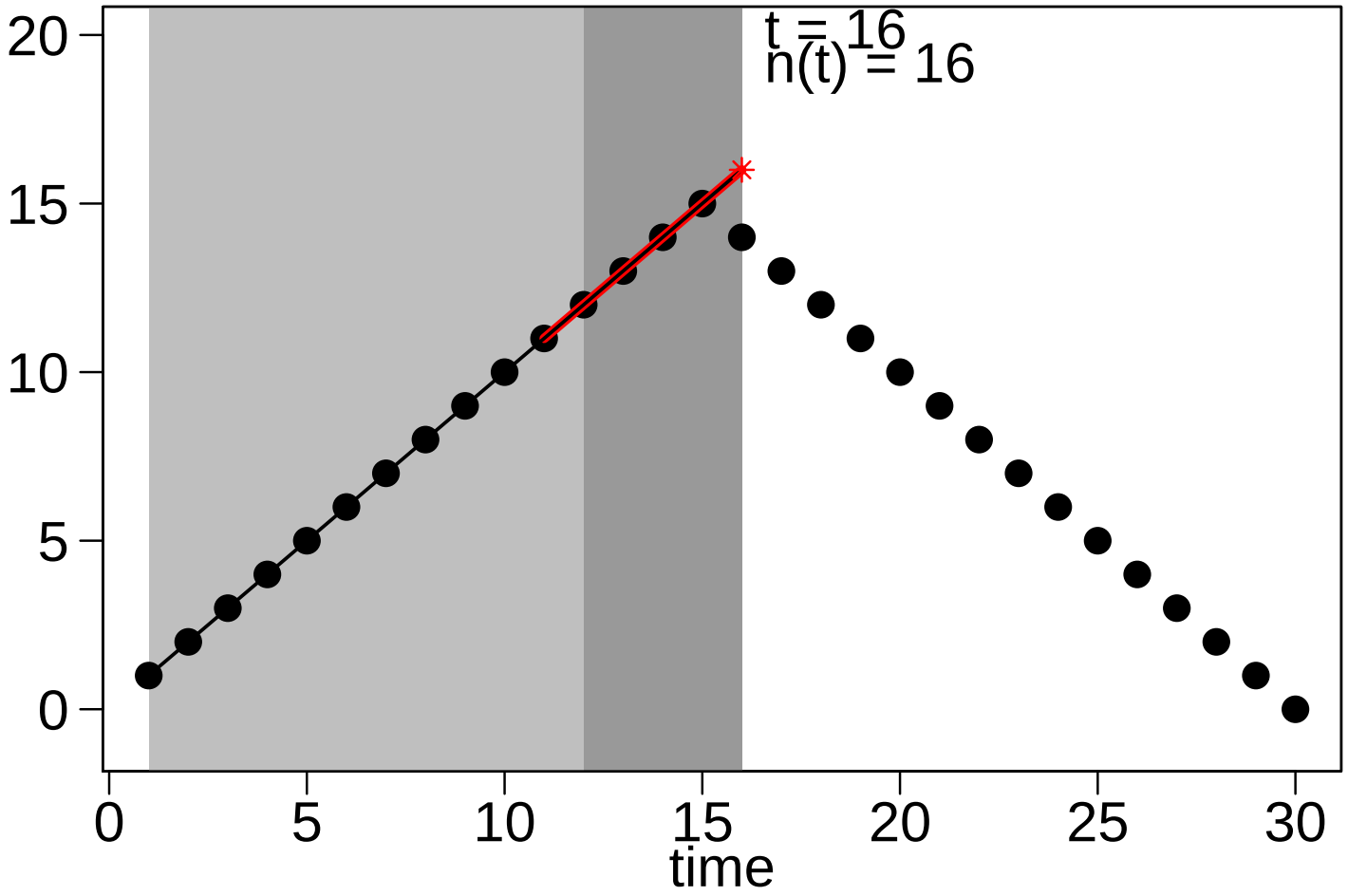
adore.filter – Illustration

```
min.width=11    p.test=5    max.width=100
```



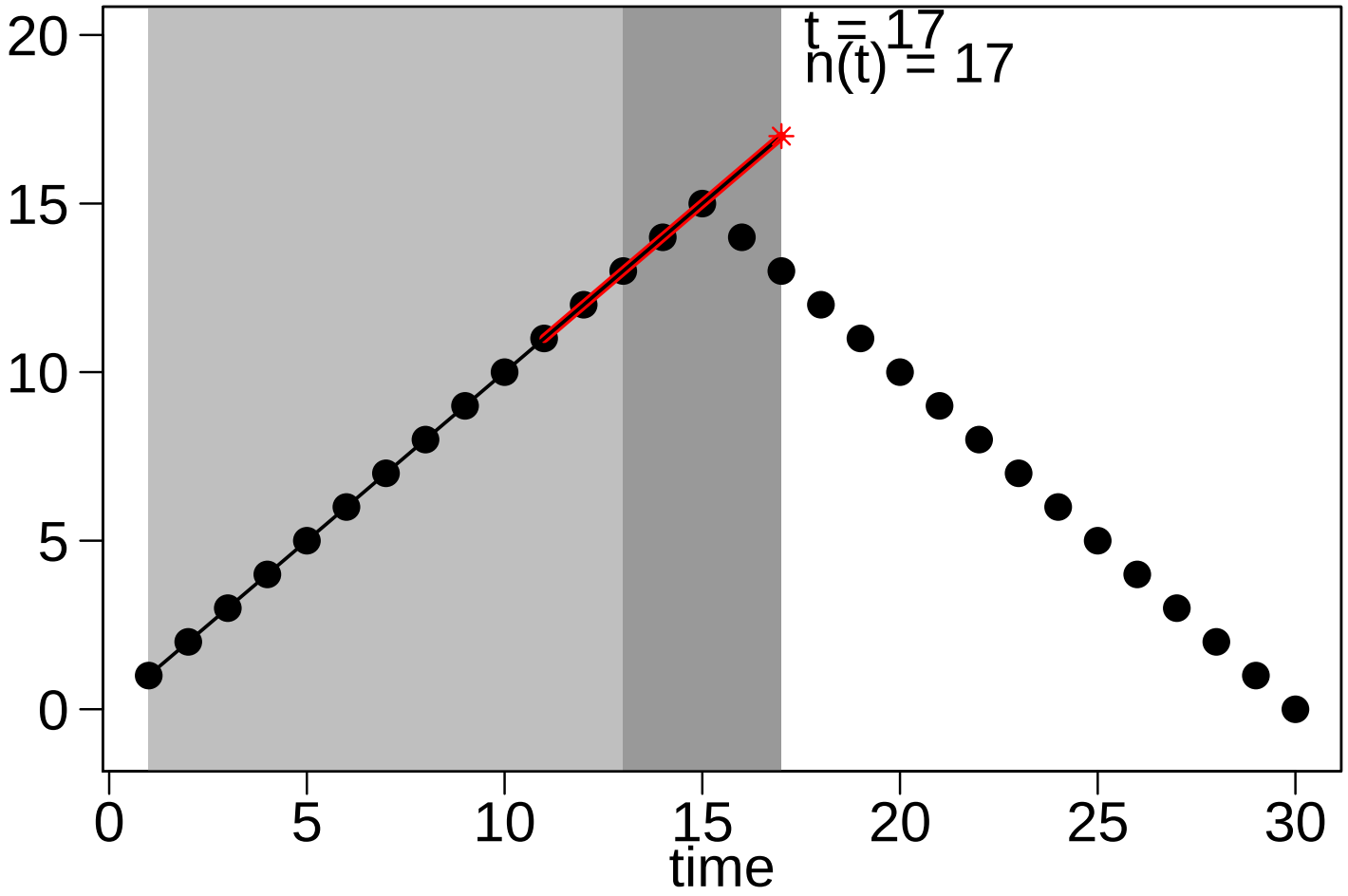
adore.filter – Illustration

```
min.width=11    p.test=5    max.width=100
```



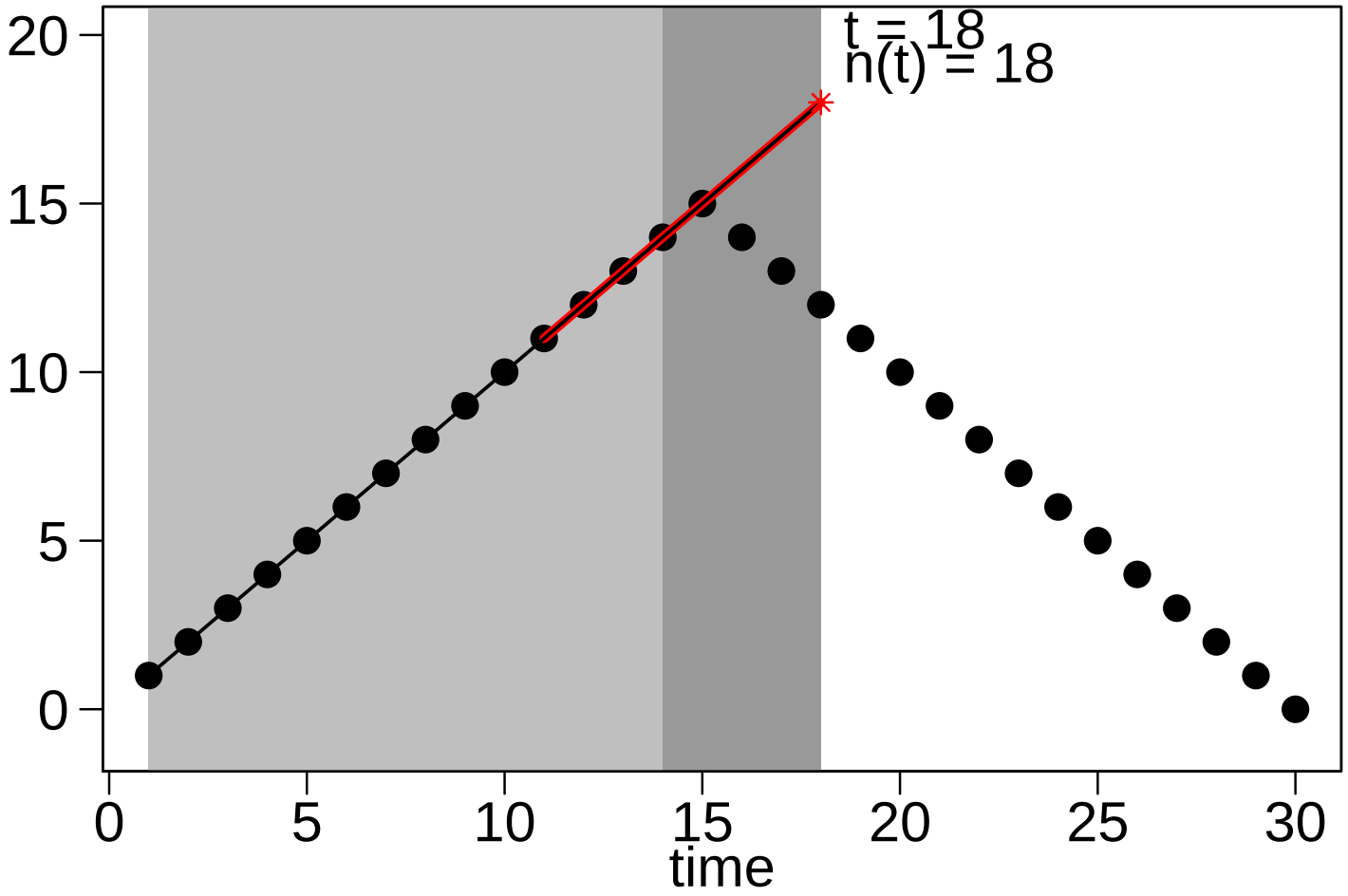
adore.filter – Illustration

```
min.width=11    p.test=5    max.width=100
```



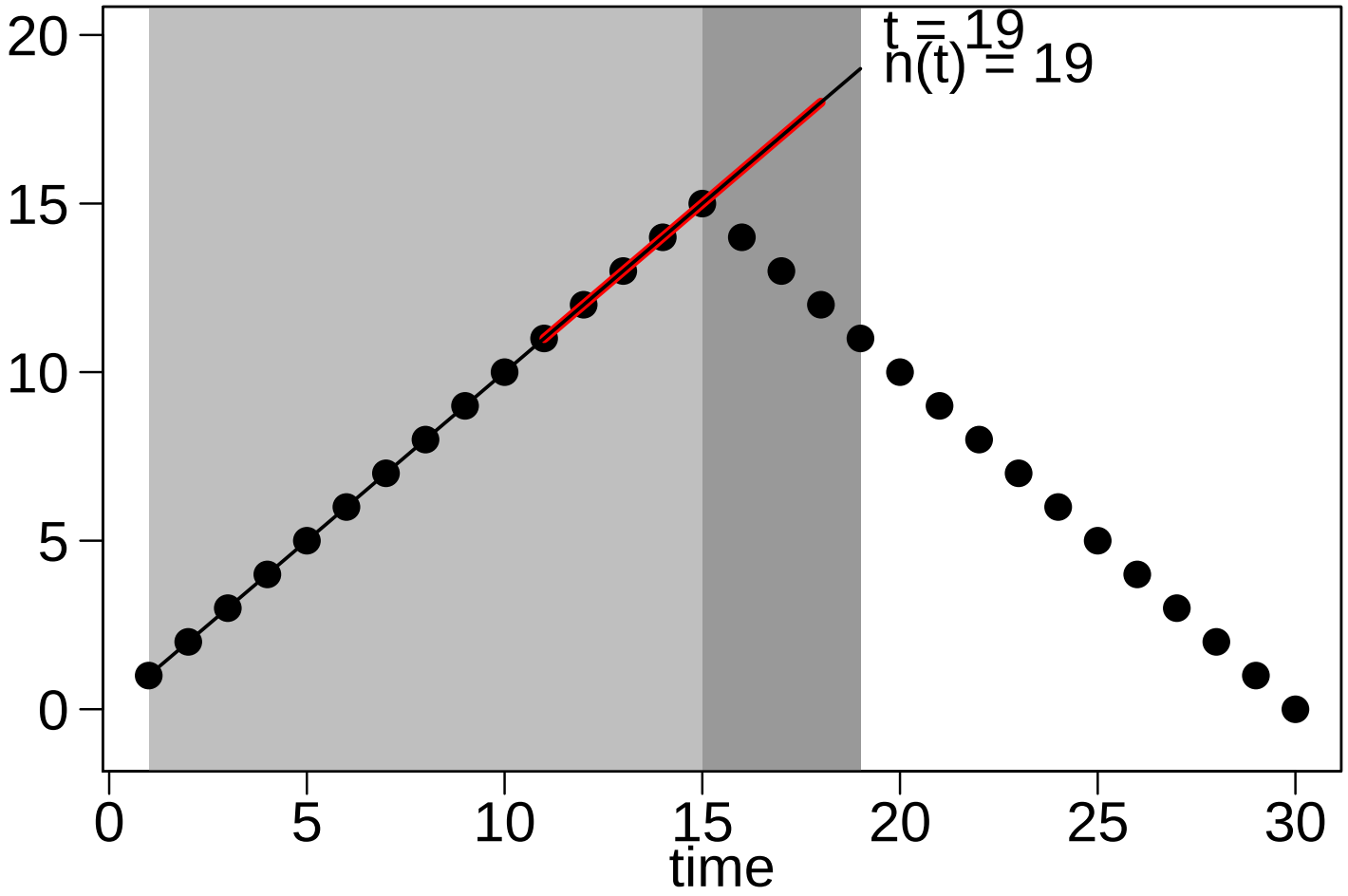
adore.filter – Illustration

`min.width=11  p.test=5  max.width=100`



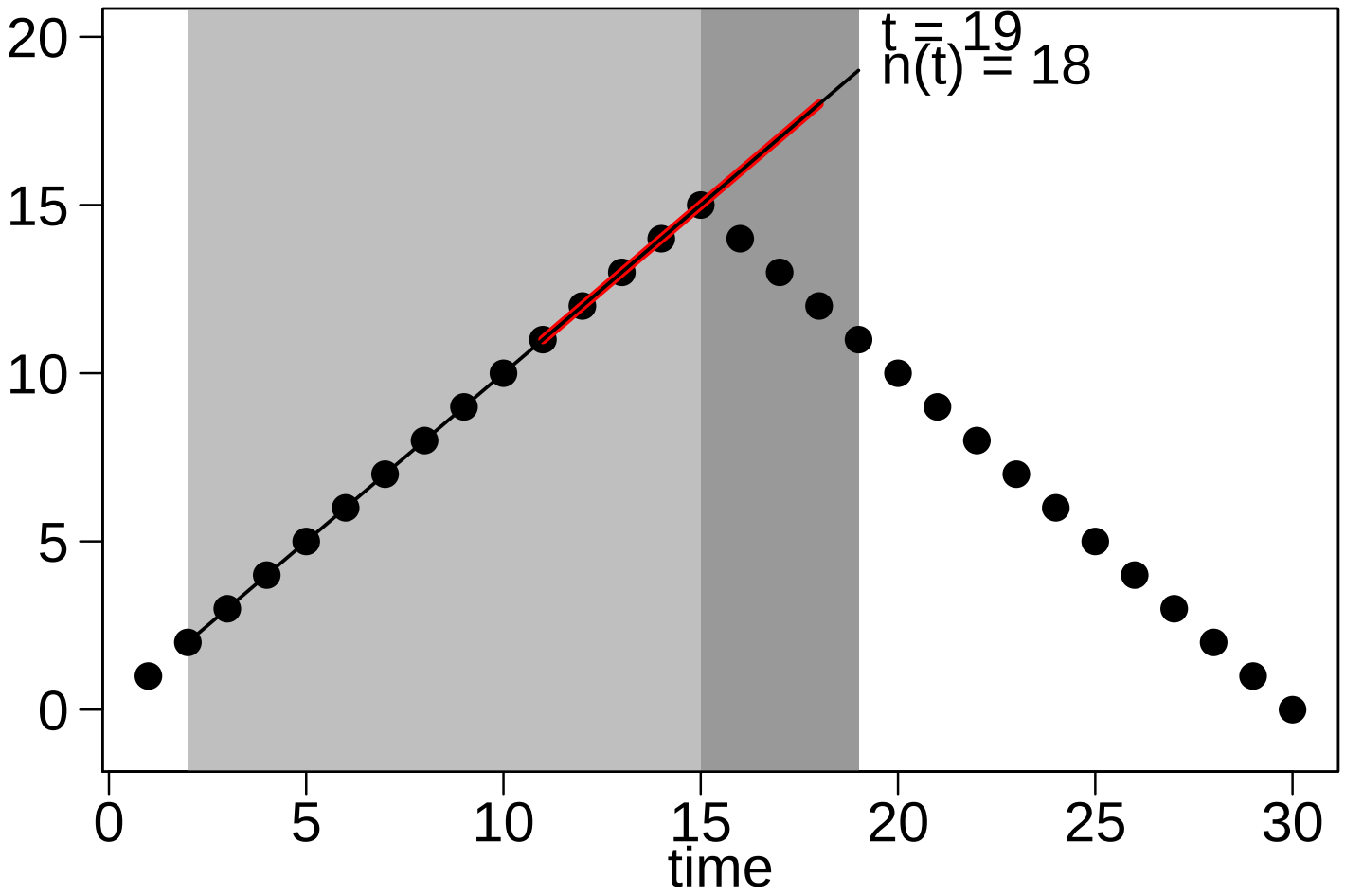
adore.filter – Illustration

`min.width=11  p.test=5  max.width=100`



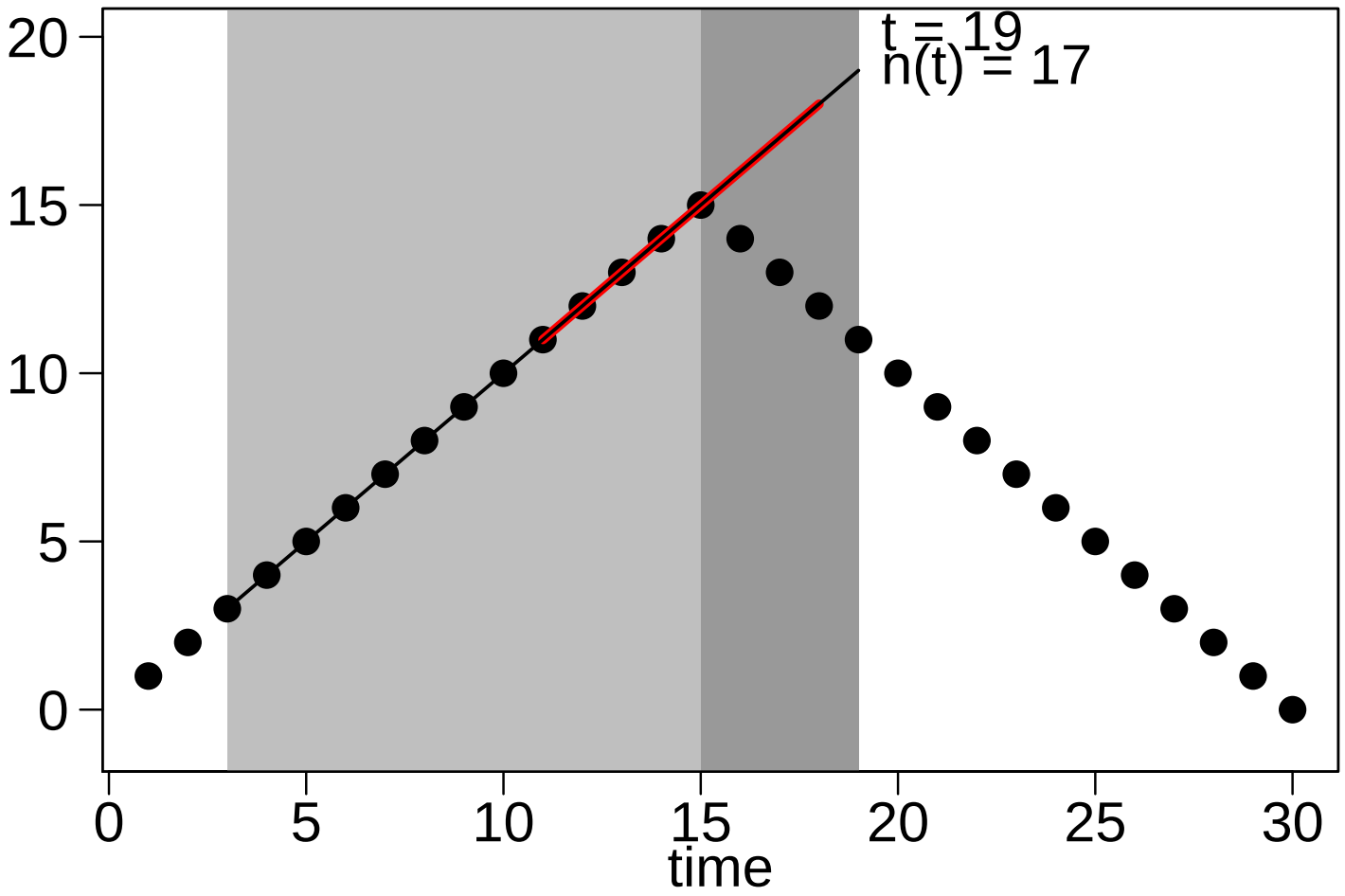
adore.filter – Illustration

```
min.width=11    p.test=5    max.width=100
```



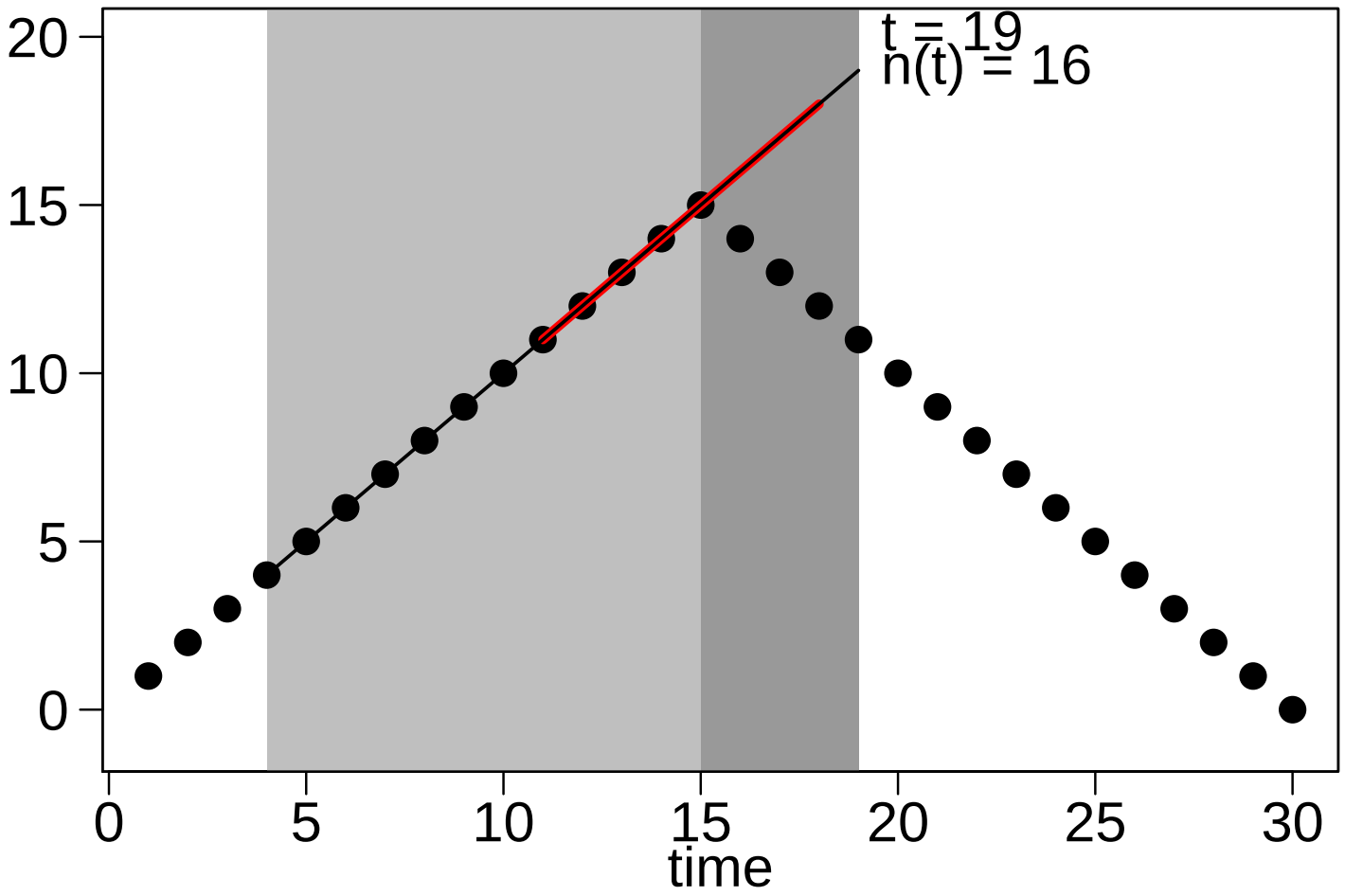
adore.filter – Illustration

`min.width=11  p.test=5  max.width=100`



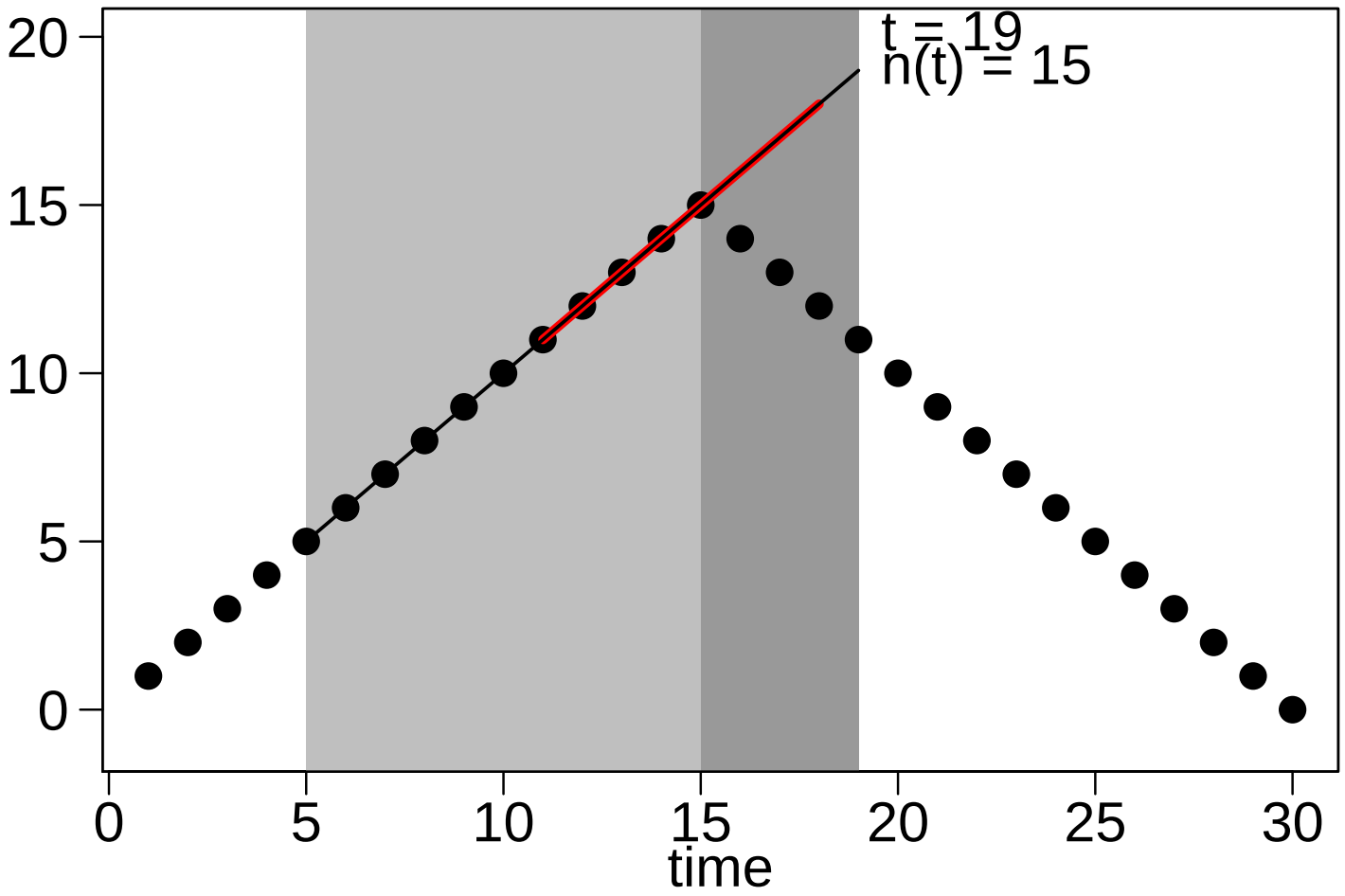
adore.filter – Illustration

```
min.width=11    p.test=5    max.width=100
```



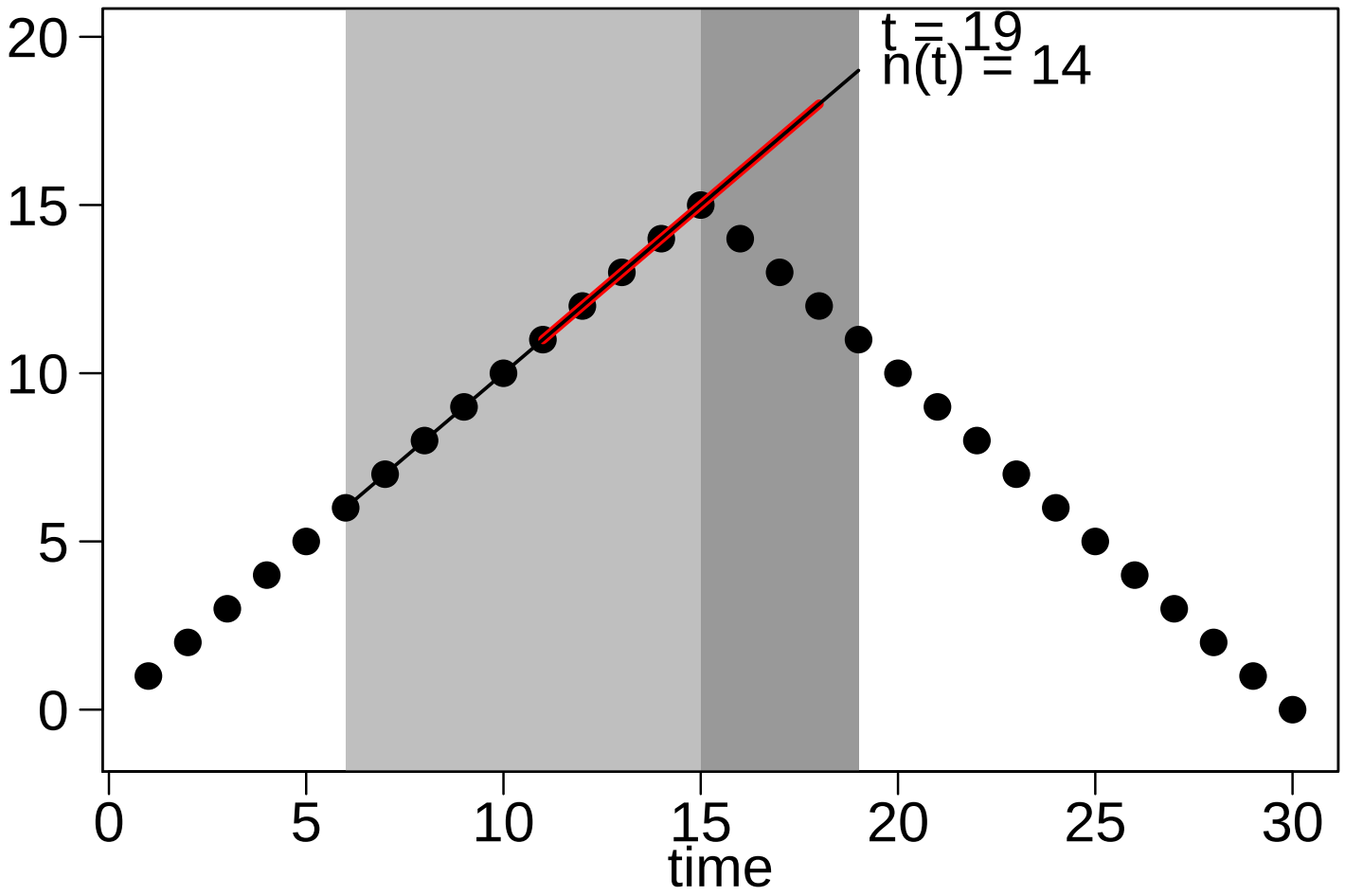
adore.filter – Illustration

```
min.width=11    p.test=5    max.width=100
```



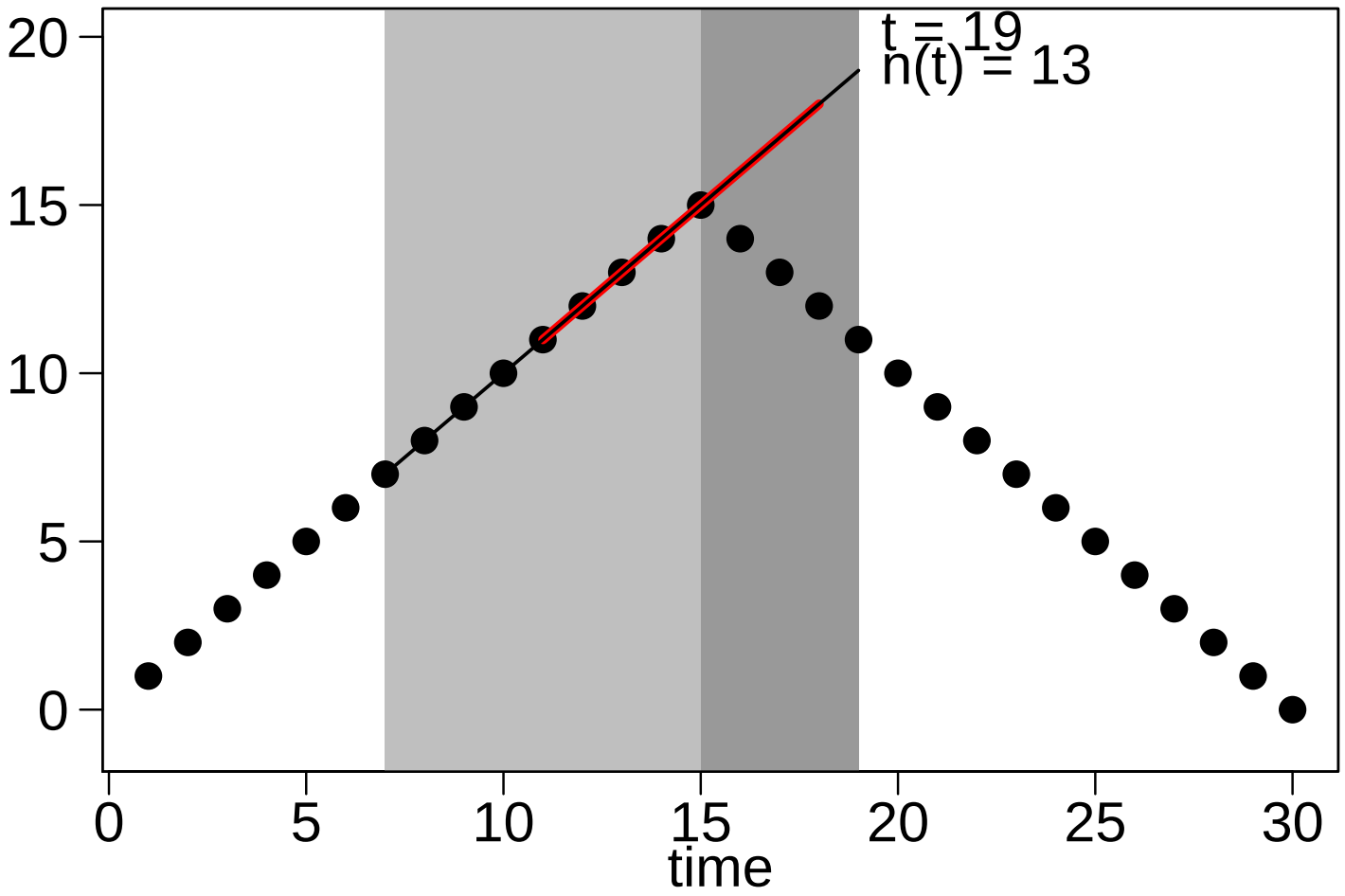
adore.filter – Illustration

`min.width=11  p.test=5  max.width=100`



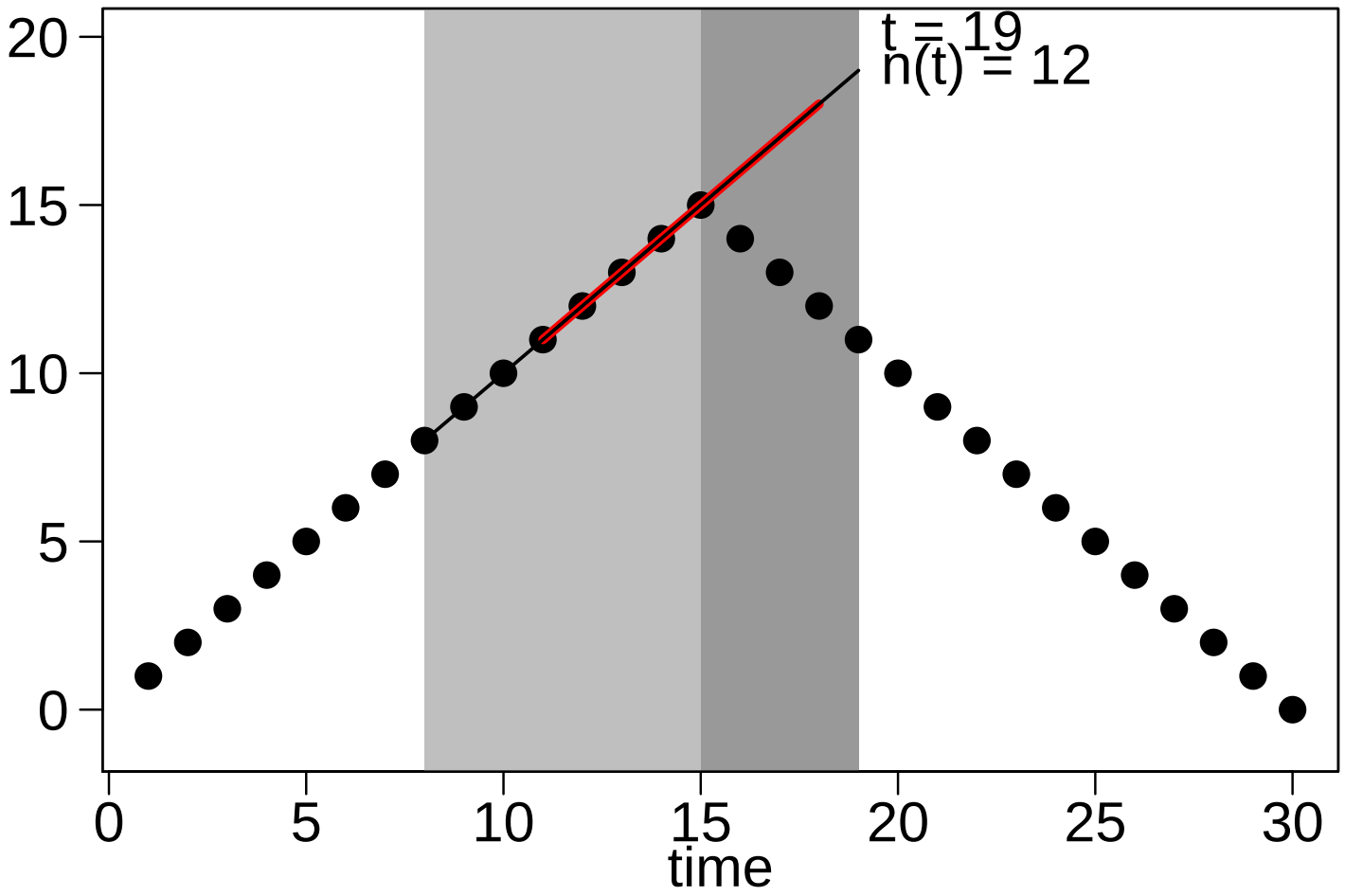
adore.filter – Illustration

```
min.width=11    p.test=5    max.width=100
```



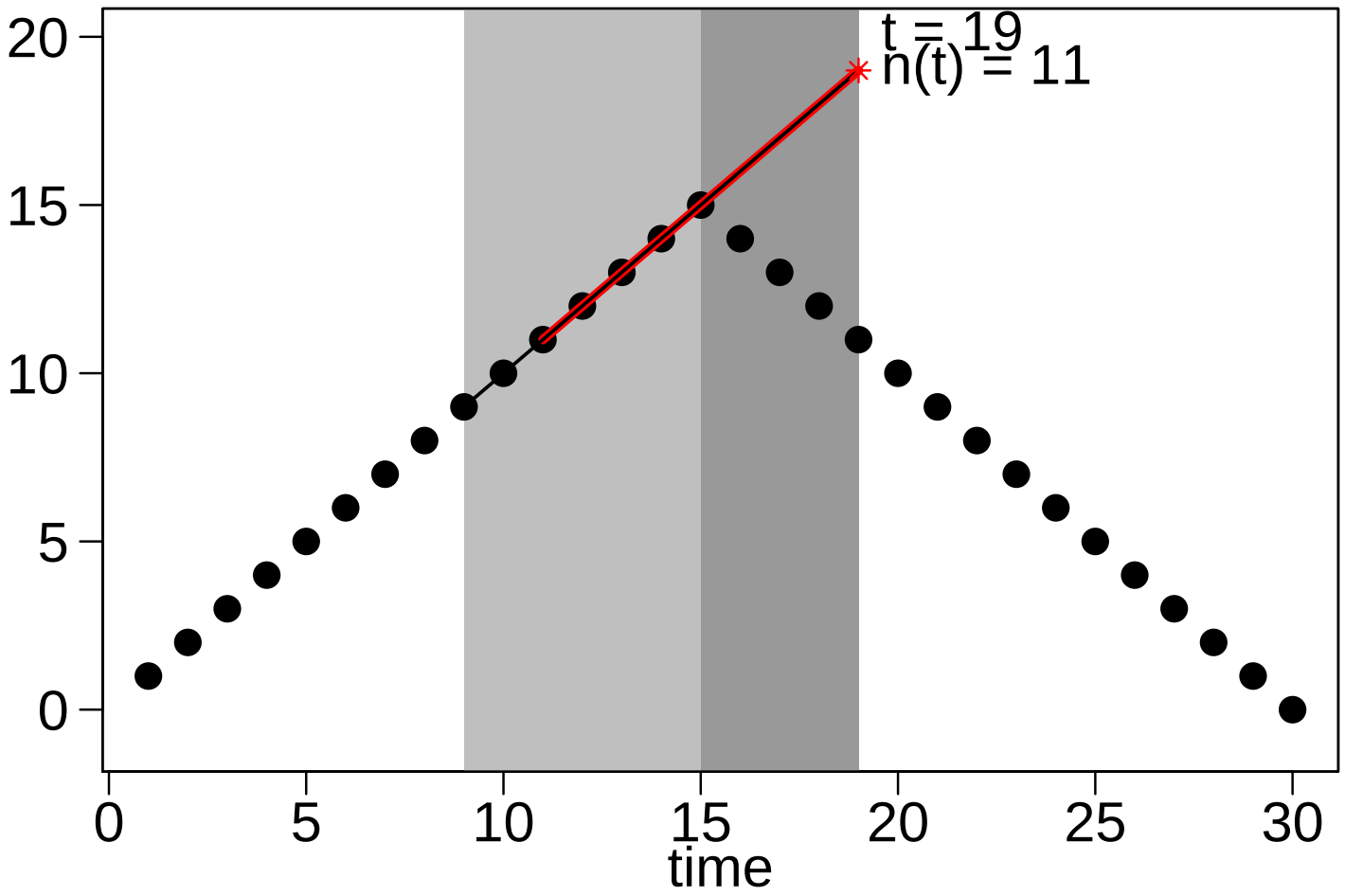
adore.filter – Illustration

```
min.width=11    p.test=5    max.width=100
```



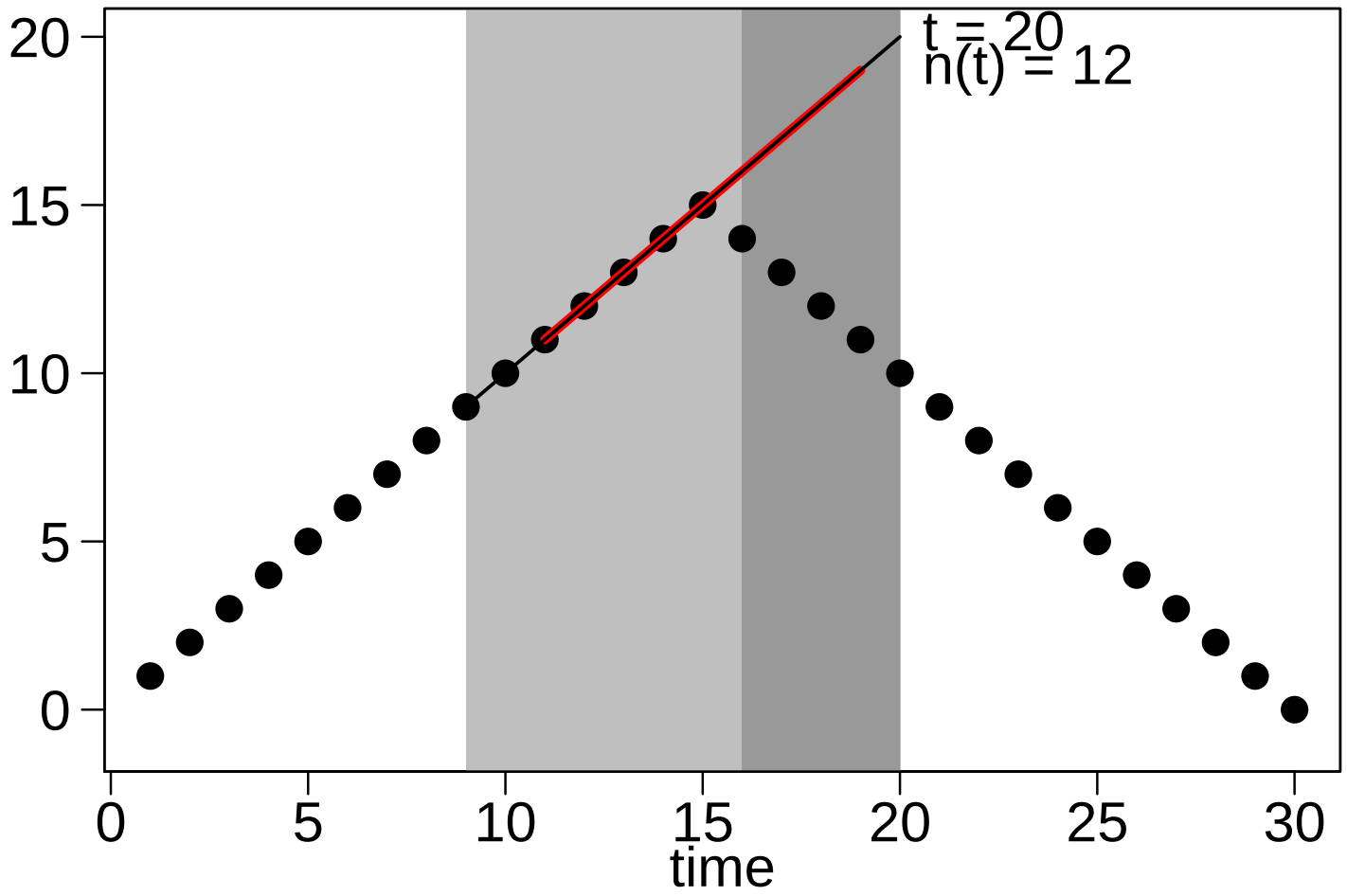
adore.filter – Illustration

```
min.width=11    p.test=5    max.width=100
```



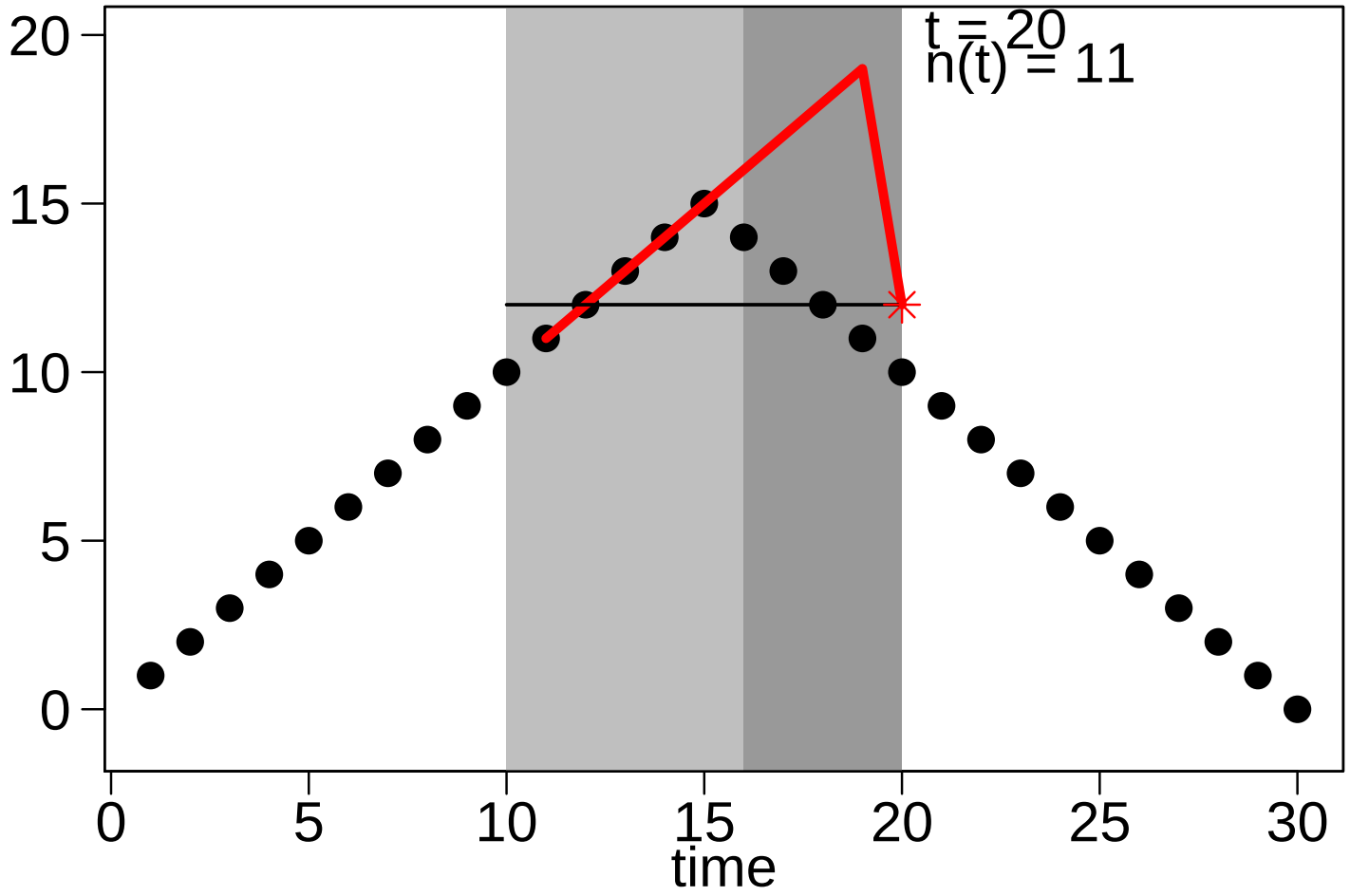
adore.filter – Illustration

```
min.width=11    p.test=5    max.width=100
```



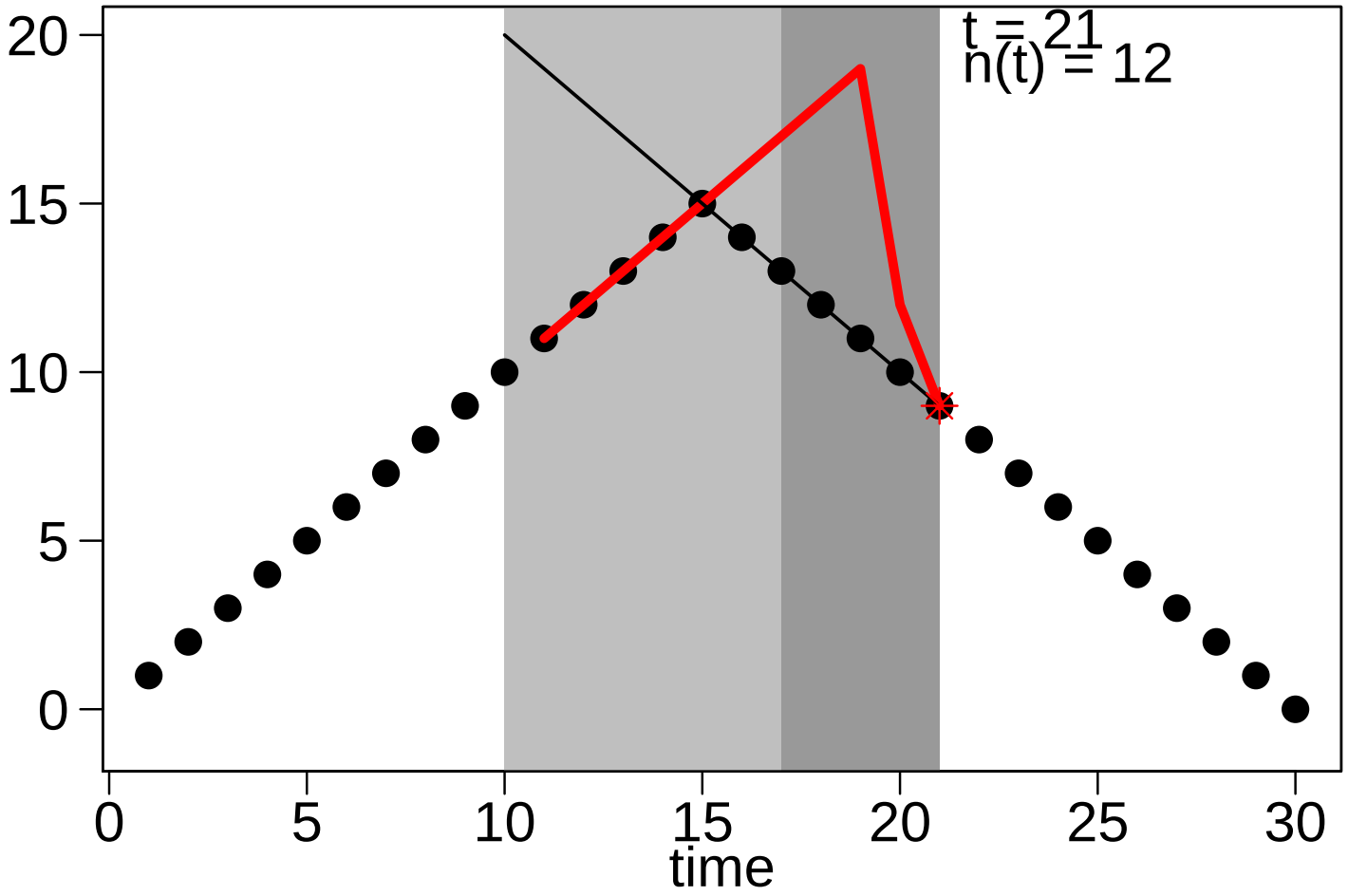
adore.filter – Illustration

`min.width=11  p.test=5  max.width=100`



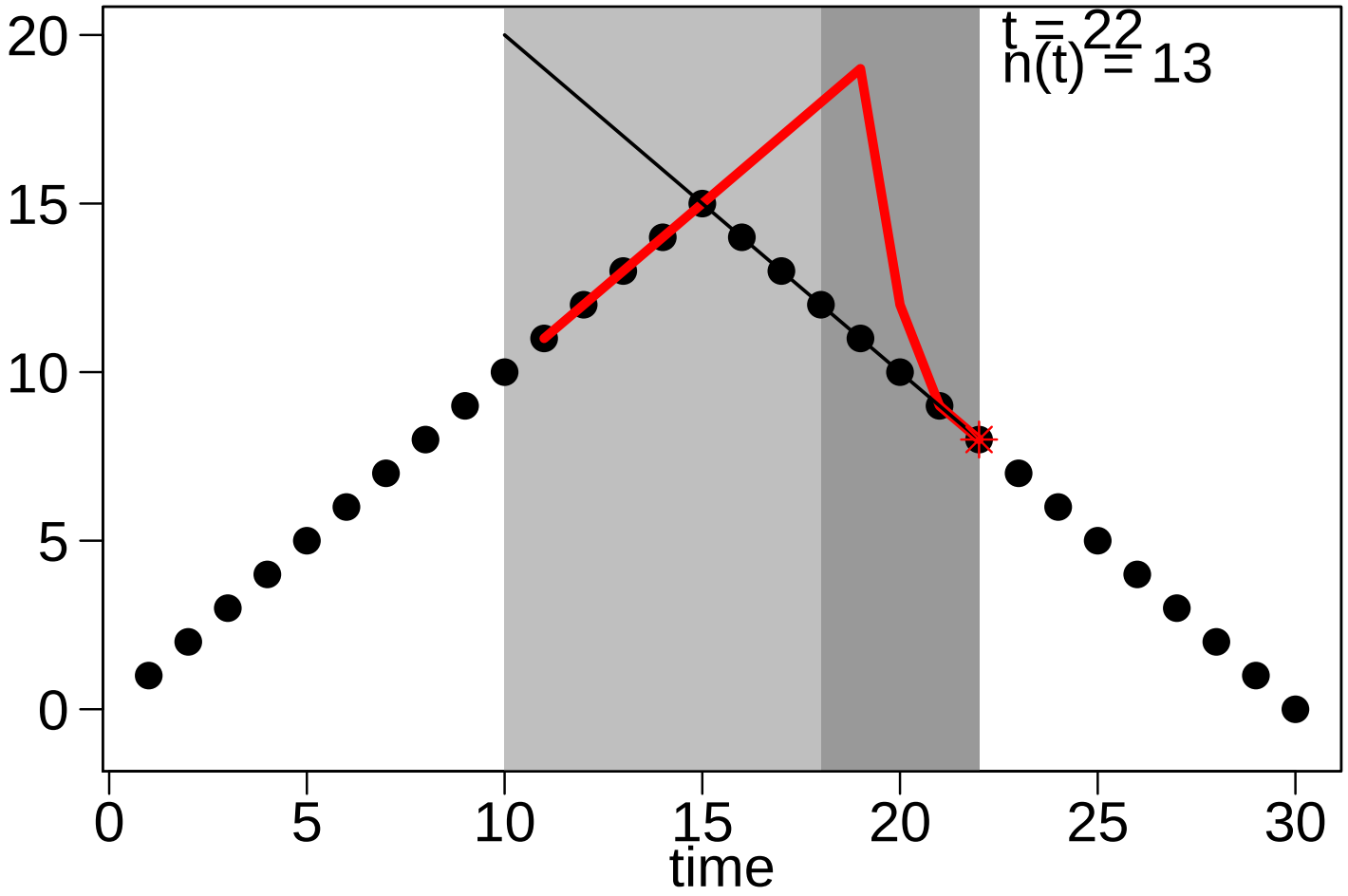
adore.filter – Illustration

```
min.width=11    p.test=5    max.width=100
```



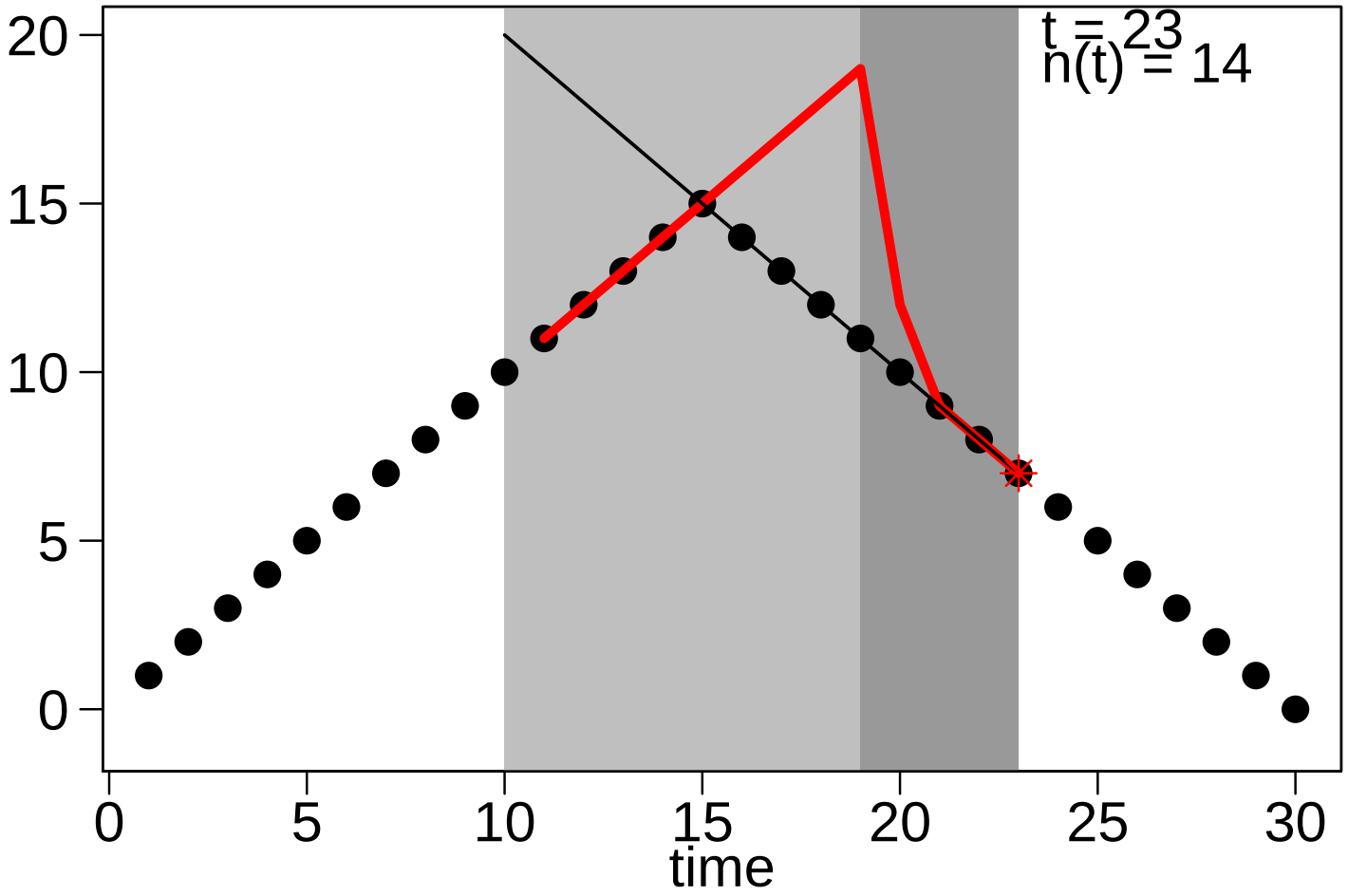
adore.filter – Illustration

```
min.width=11    p.test=5    max.width=100
```



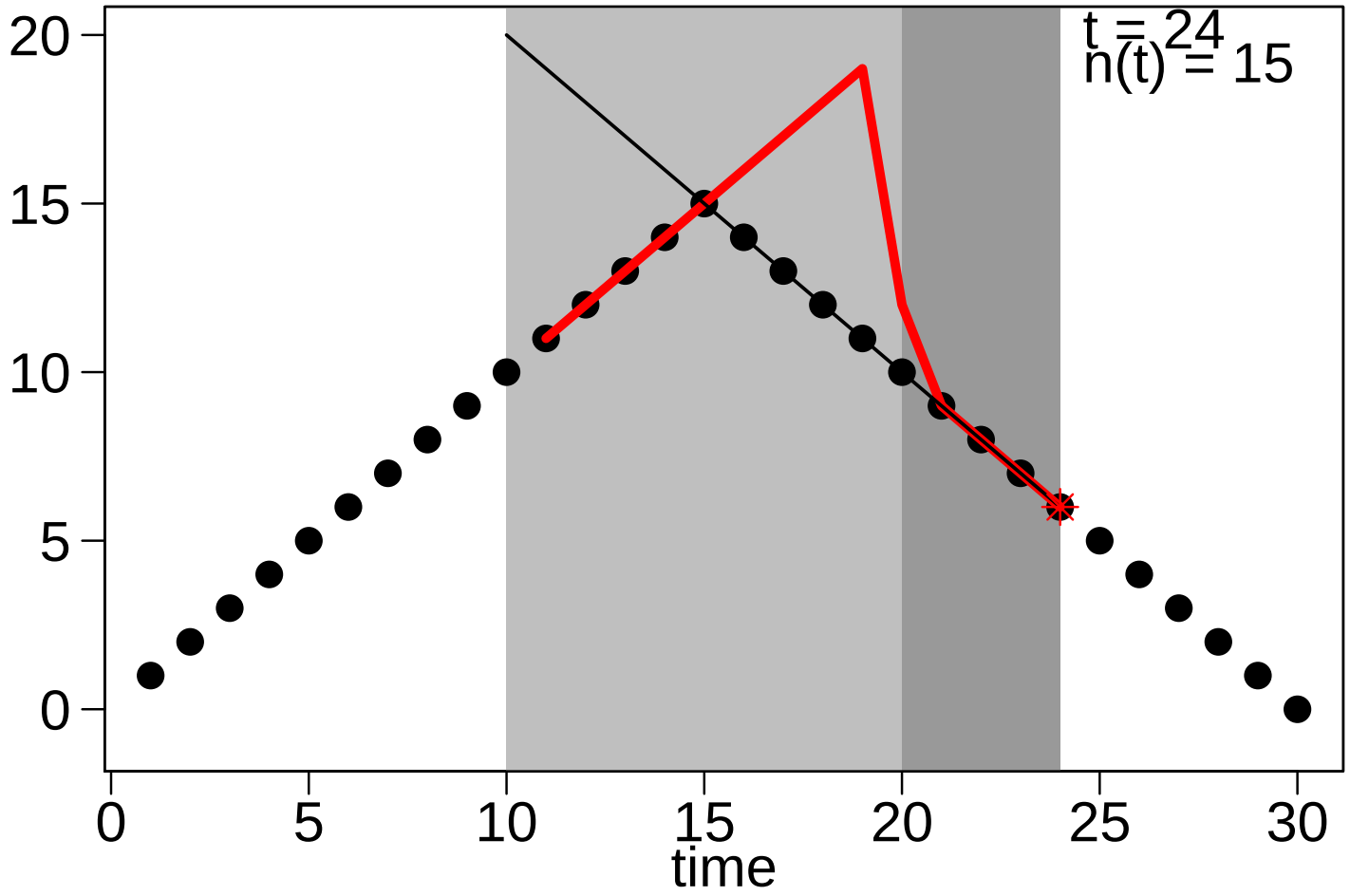
adore.filter – Illustration

```
min.width=11    p.test=5    max.width=100
```



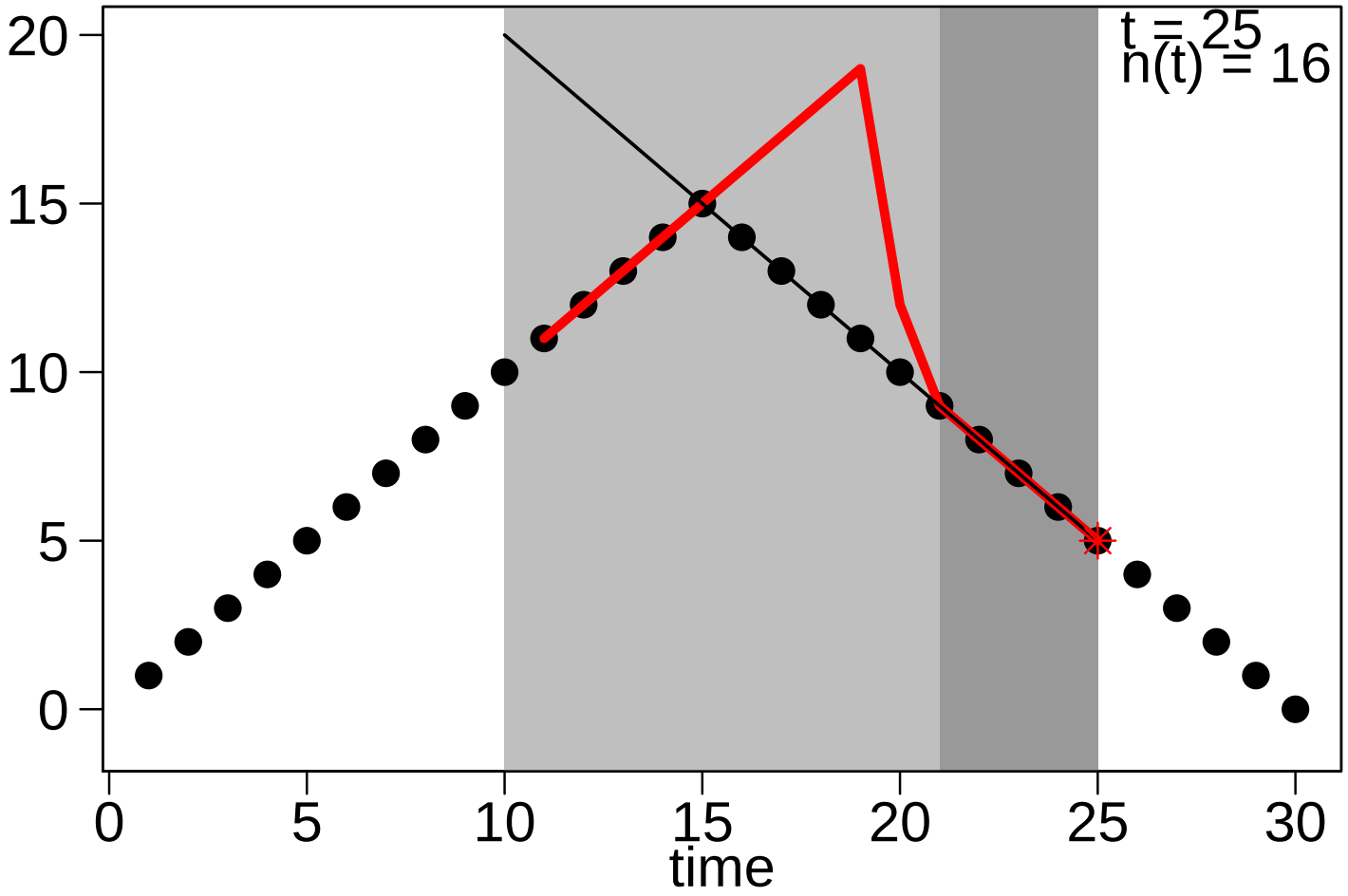
adore.filter – Illustration

```
min.width=11    p.test=5    max.width=100
```



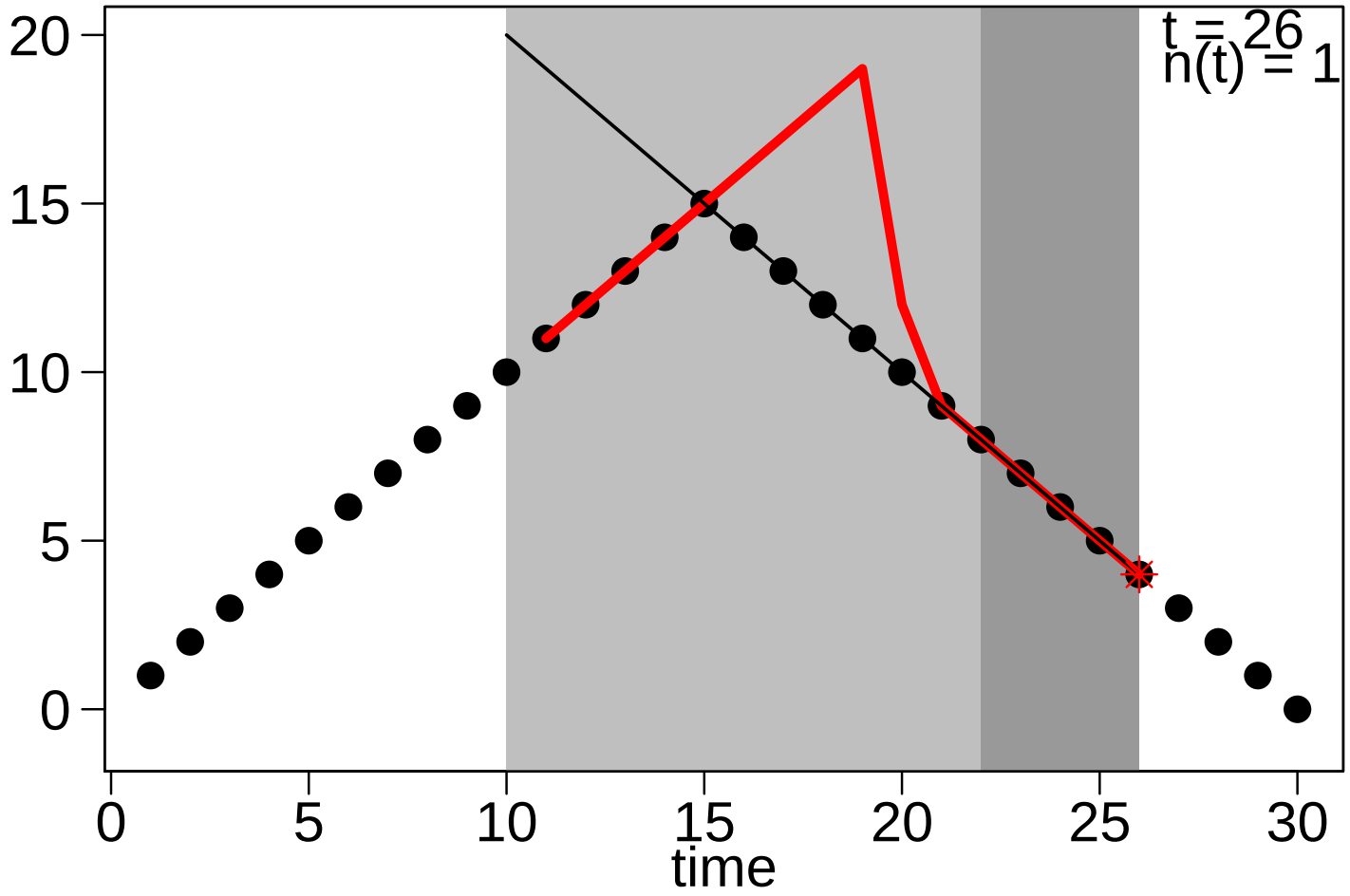
adore.filter – Illustration

```
min.width=11    p.test=5    max.width=100
```



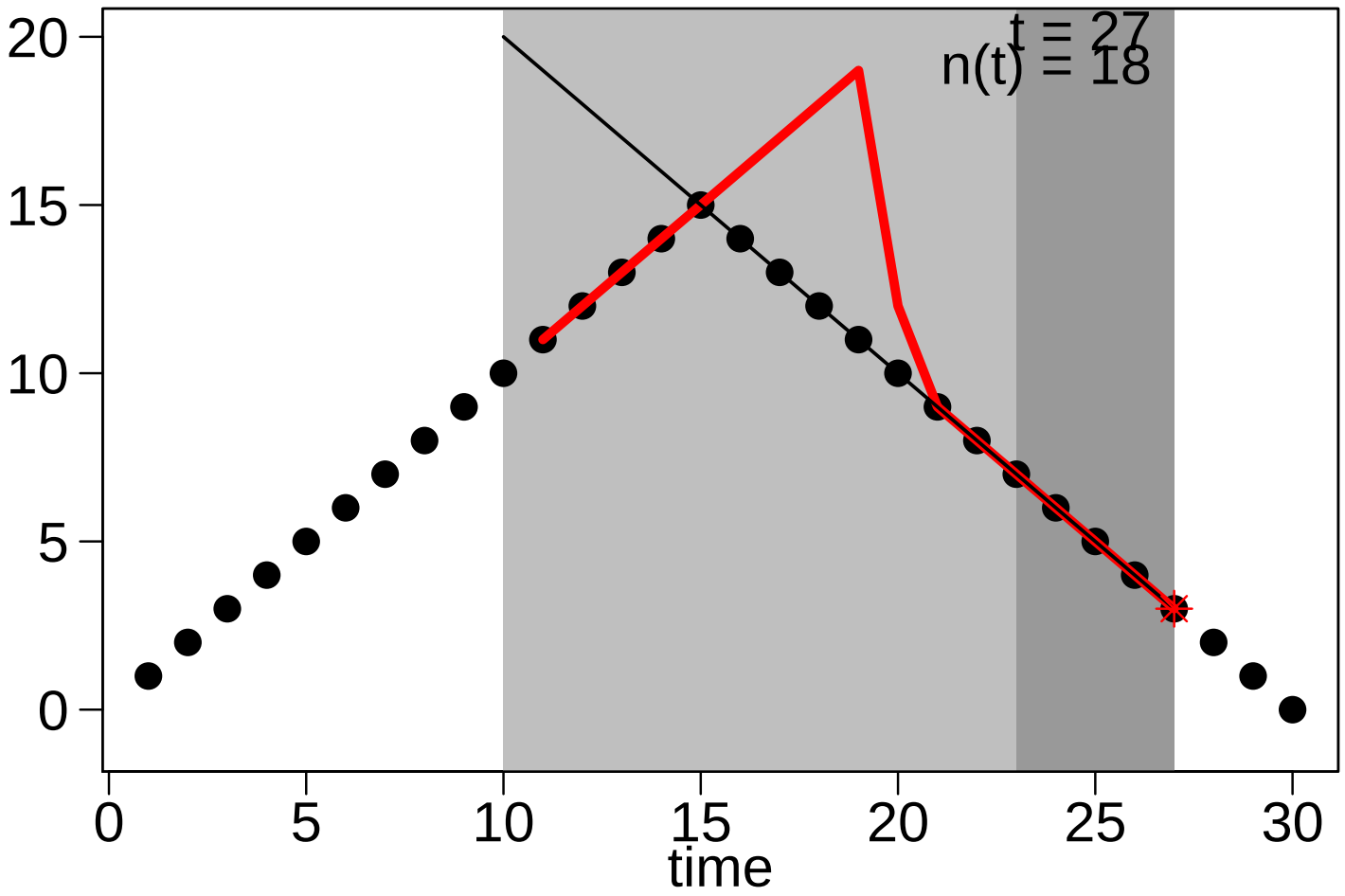
adore.filter – Illustration

```
min.width=11    p.test=5    max.width=100
```



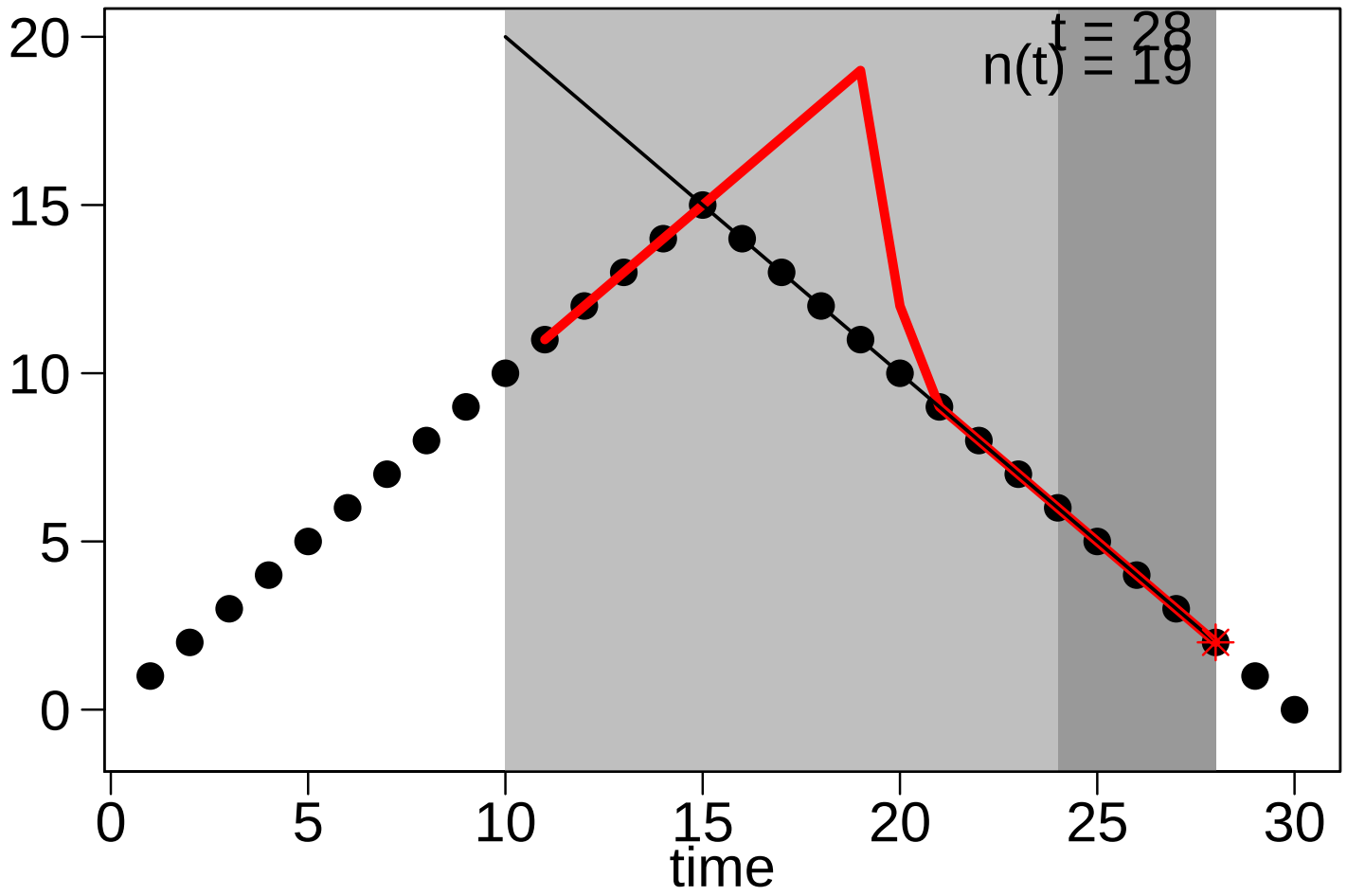
adore.filter – Illustration

```
min.width=11    p.test=5    max.width=100
```



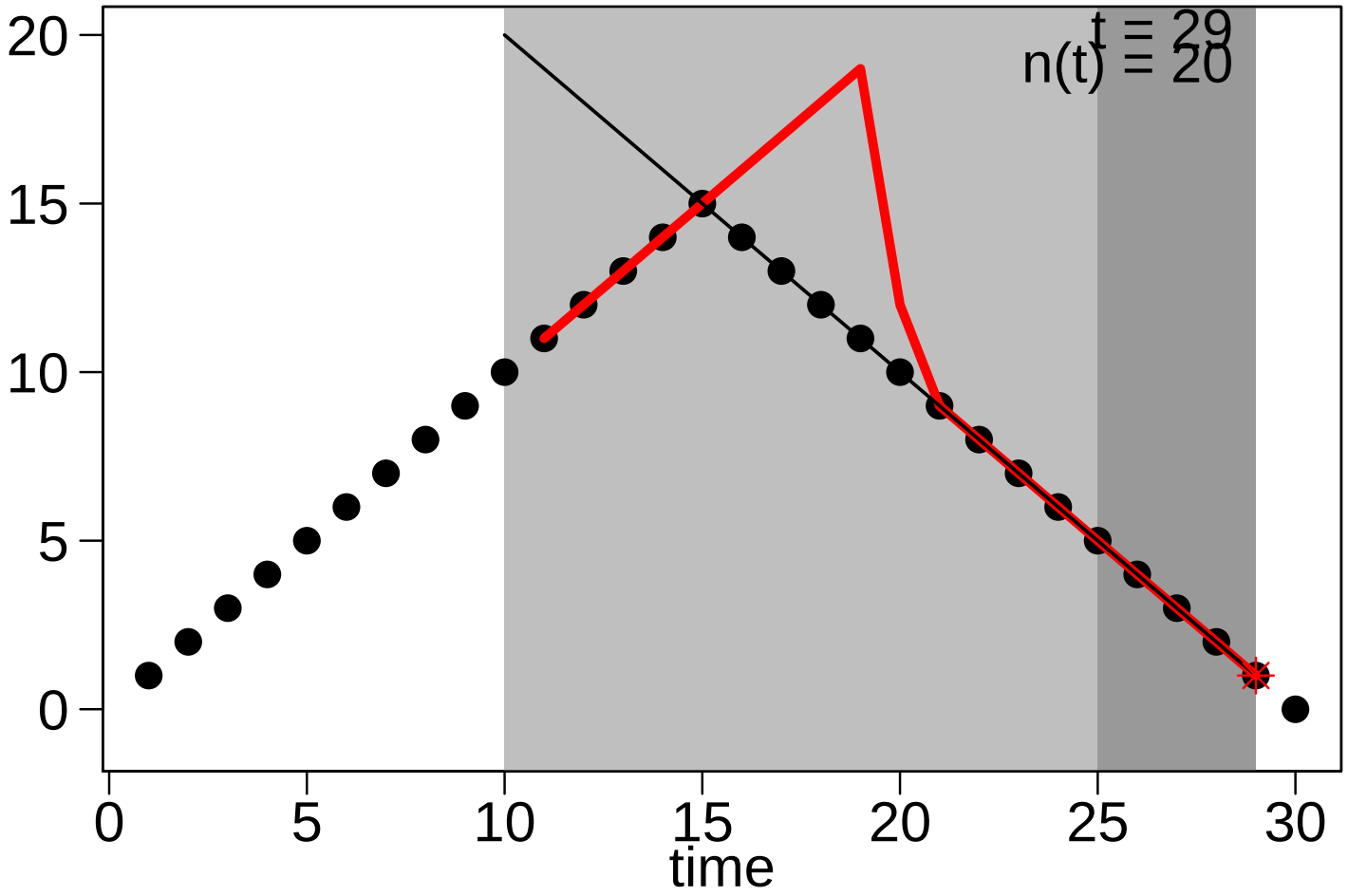
adore.filter – Illustration

```
min.width=11    p.test=5    max.width=100
```



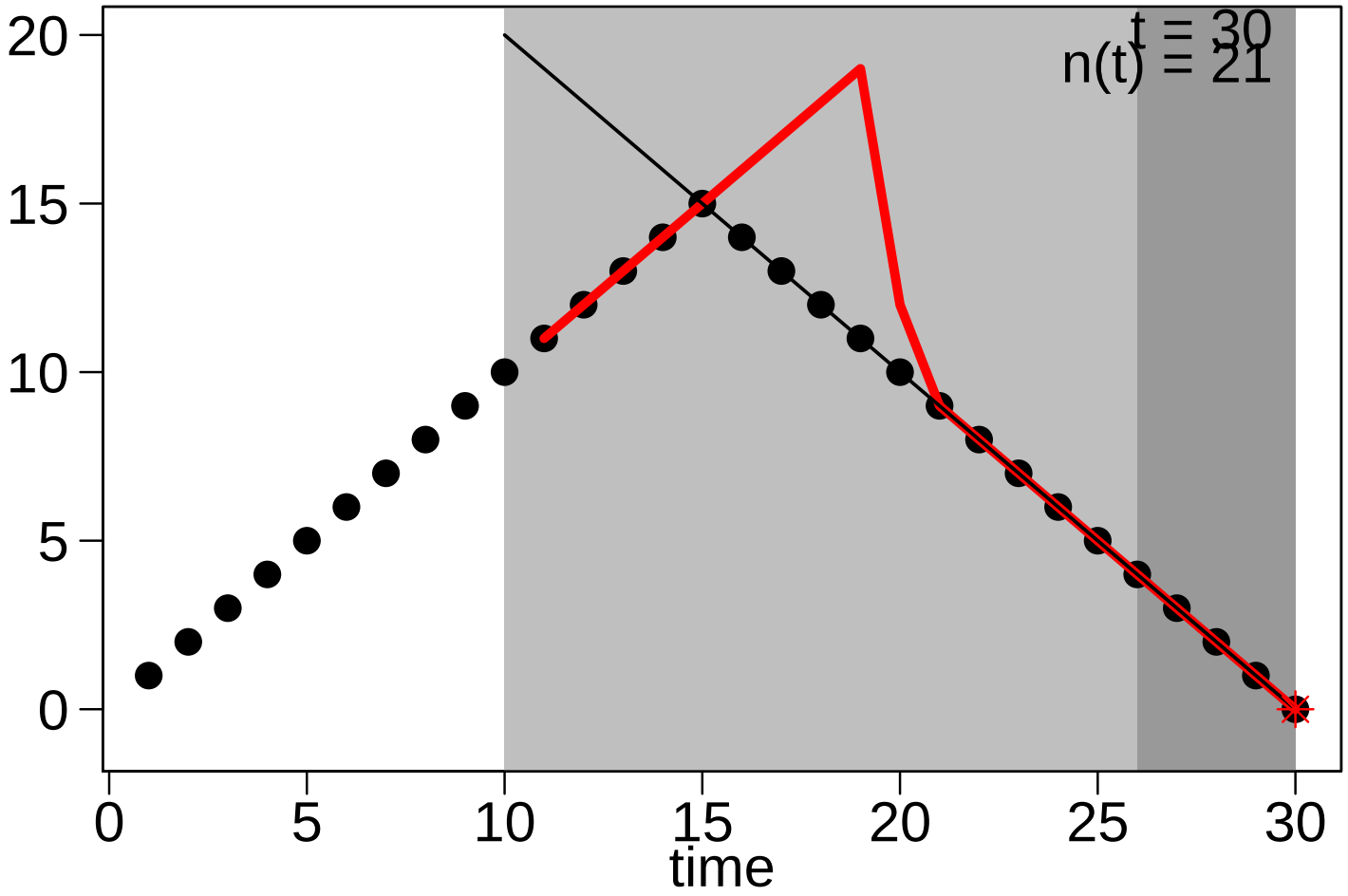
adore.filter – Illustration

```
min.width=11    p.test=5    max.width=100
```



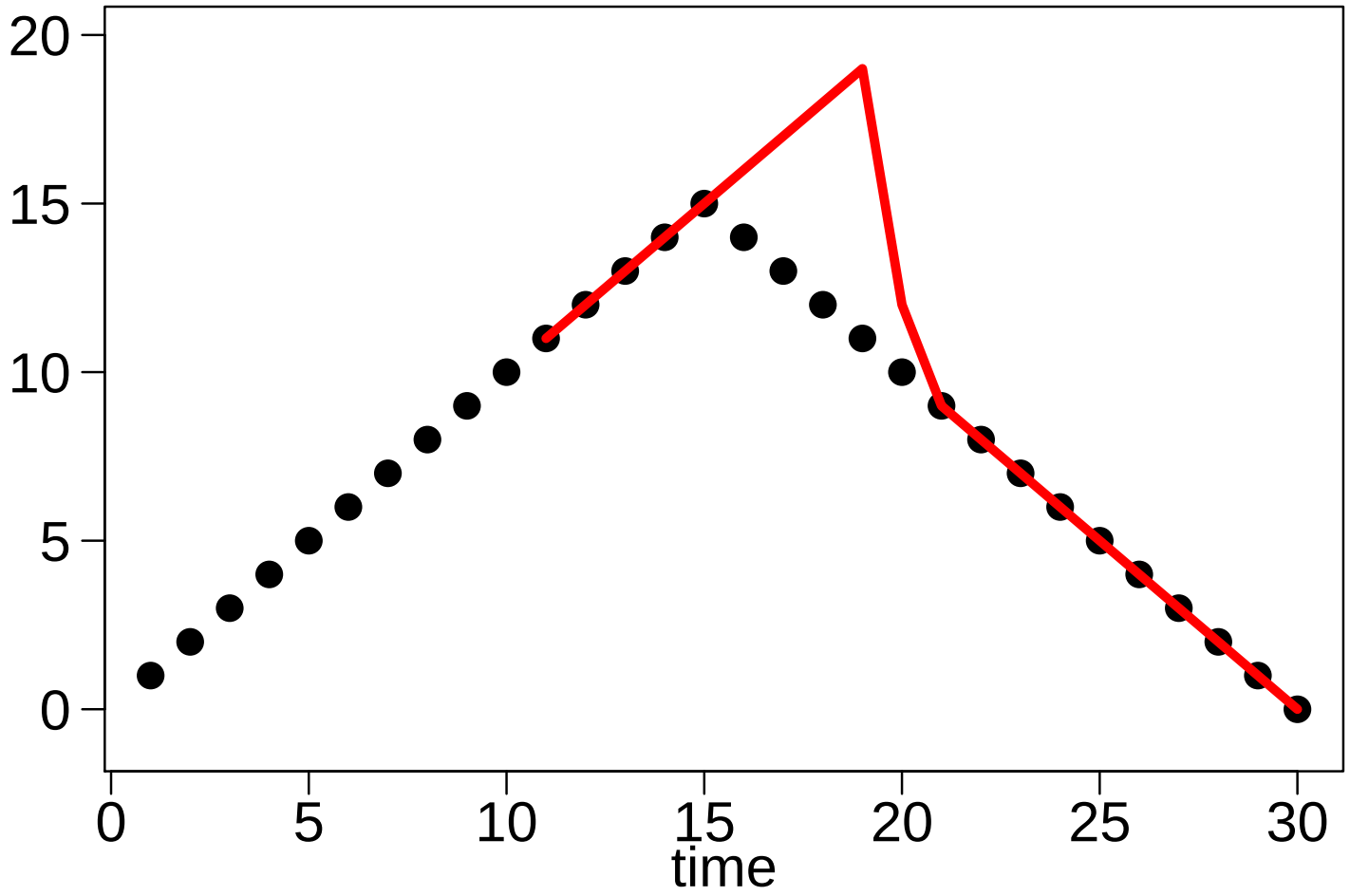
adore.filter – Illustration

```
min.width=11    p.test=5    max.width=100
```



adore.filter – Illustration

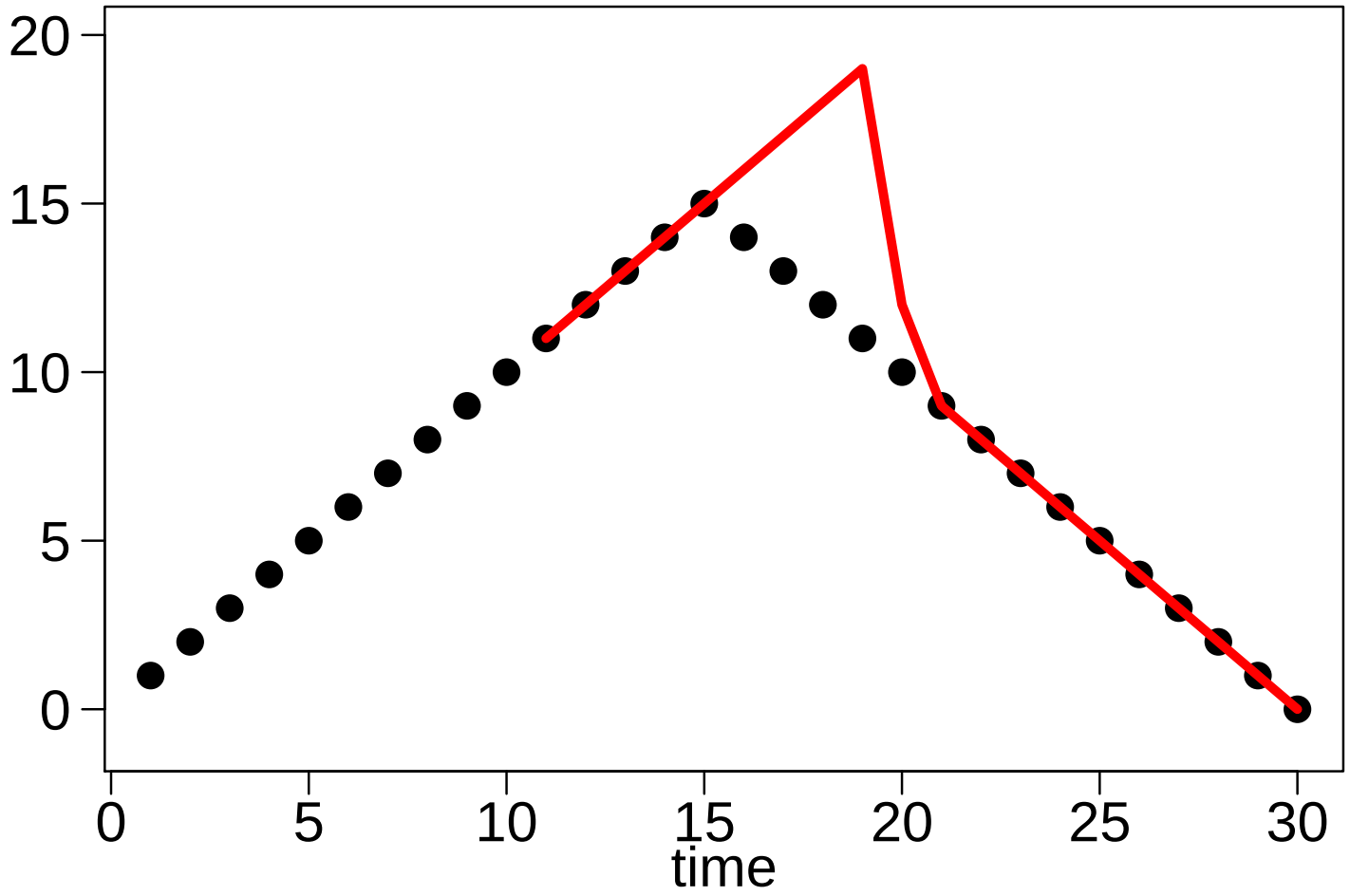
`min.width=11  p.test=5  max.width=100`



`adore.filter` – Illustration

`min.width=11` `p.test=5` `max.width=100`

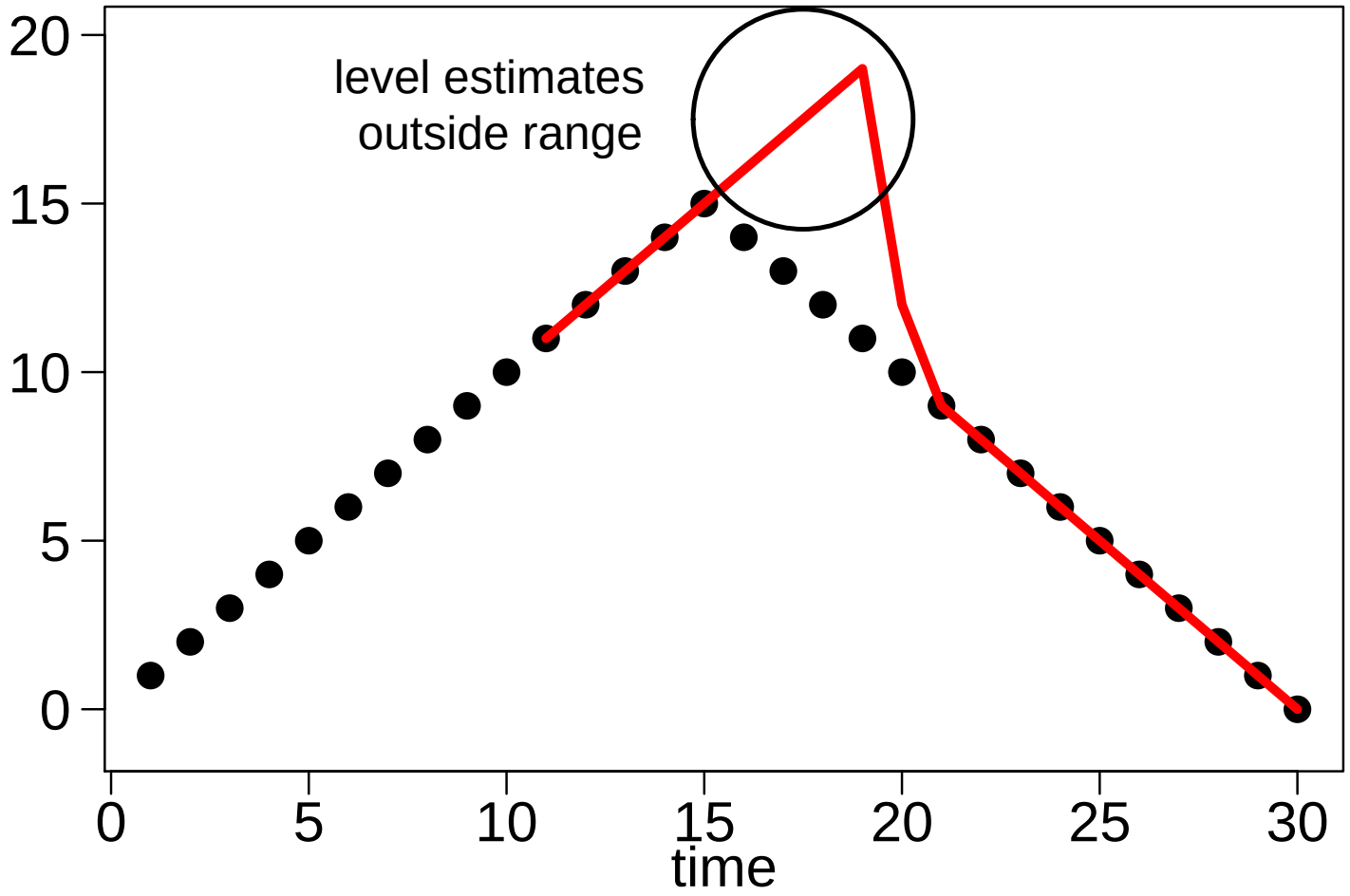
`rtr=0`: No restriction on the estimated signal level



adore.filter – Illustration

`min.width=11` `p.test=5` `max.width=100`

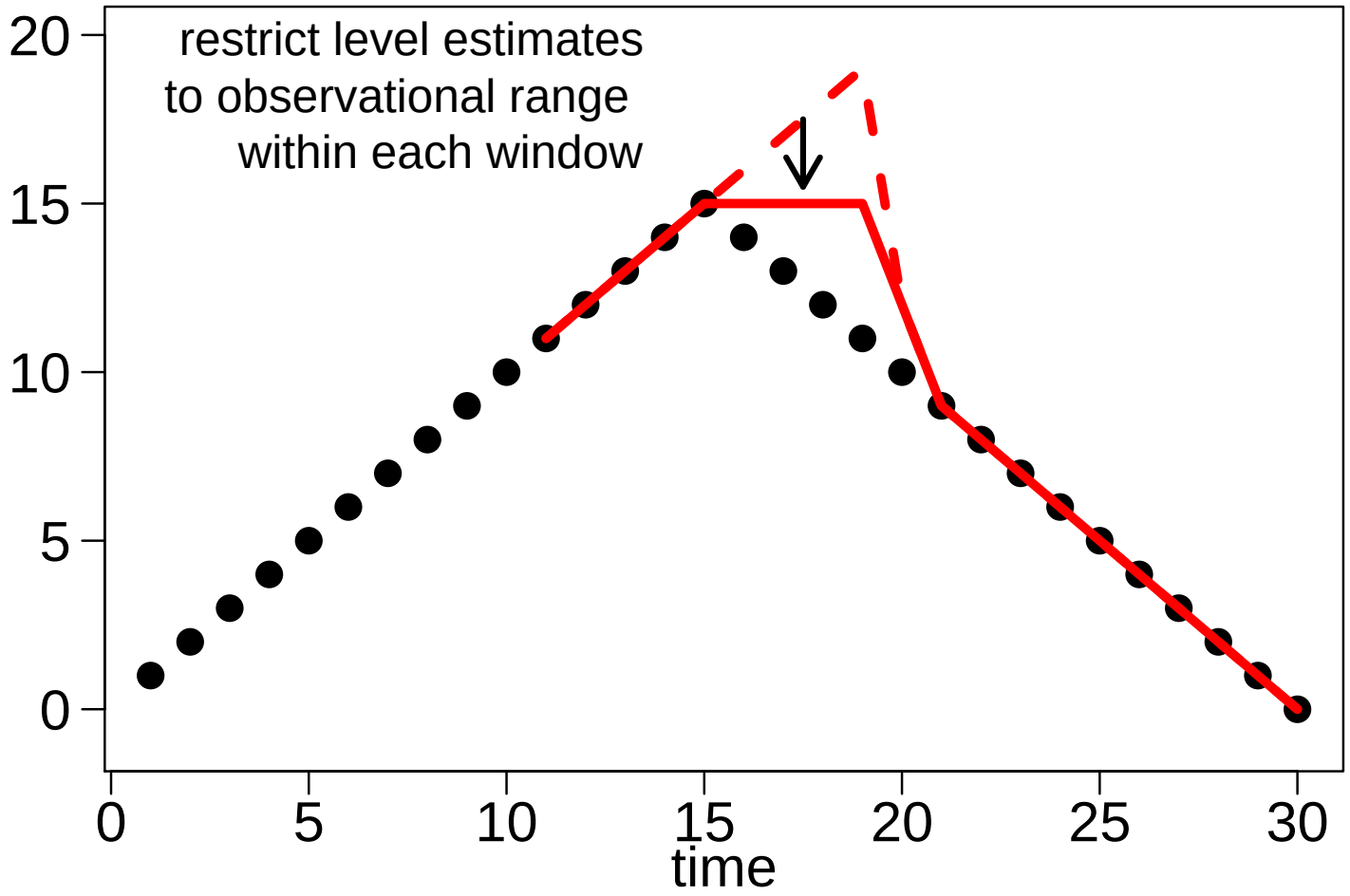
`rtr=0`: No restriction on the estimated signal level



adore.filter – Illustration

`min.width=11` `p.test=5` `max.width=100`

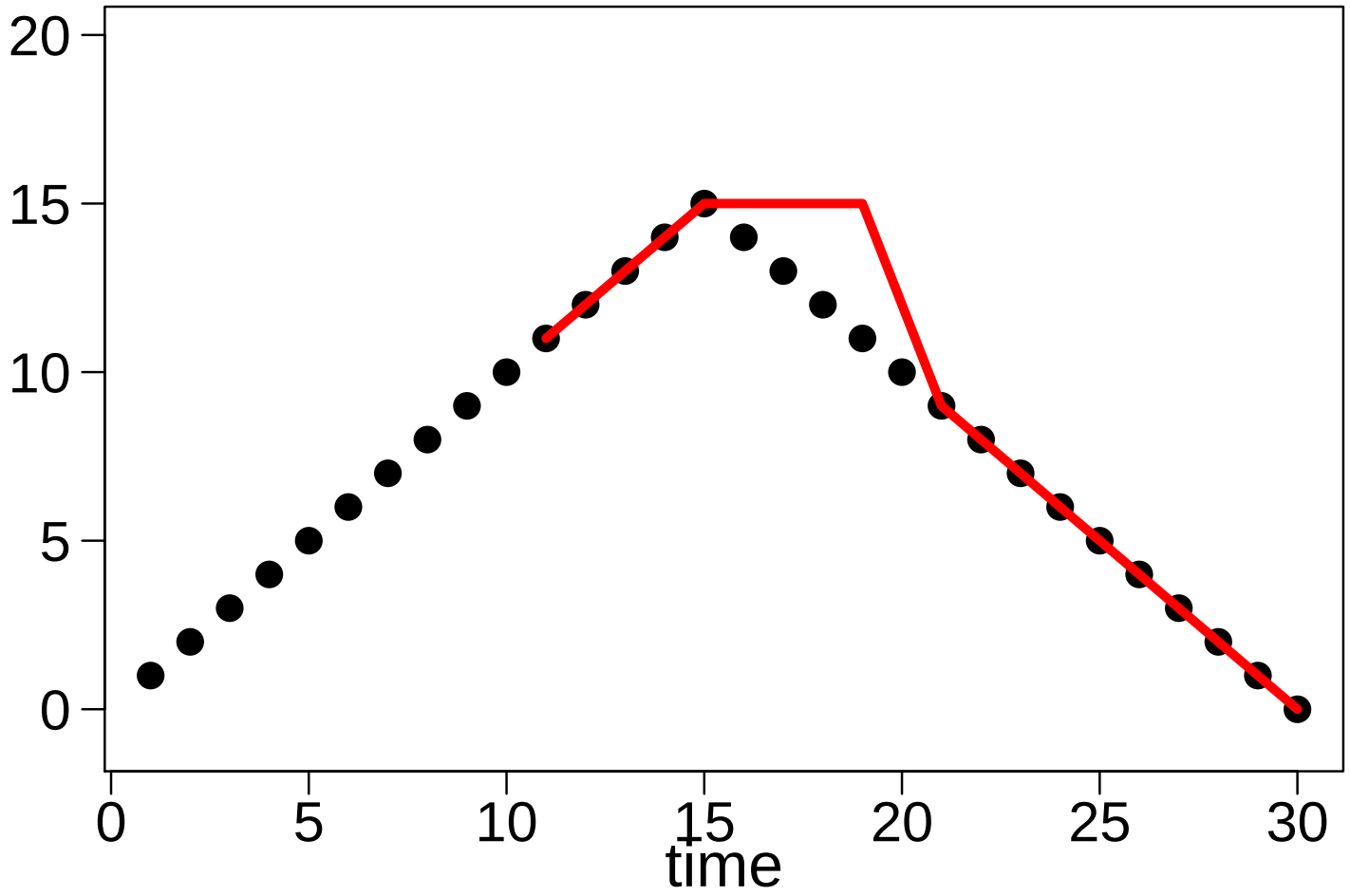
`rtr=0`: No restriction on the estimated signal level



adore.filter – Illustration

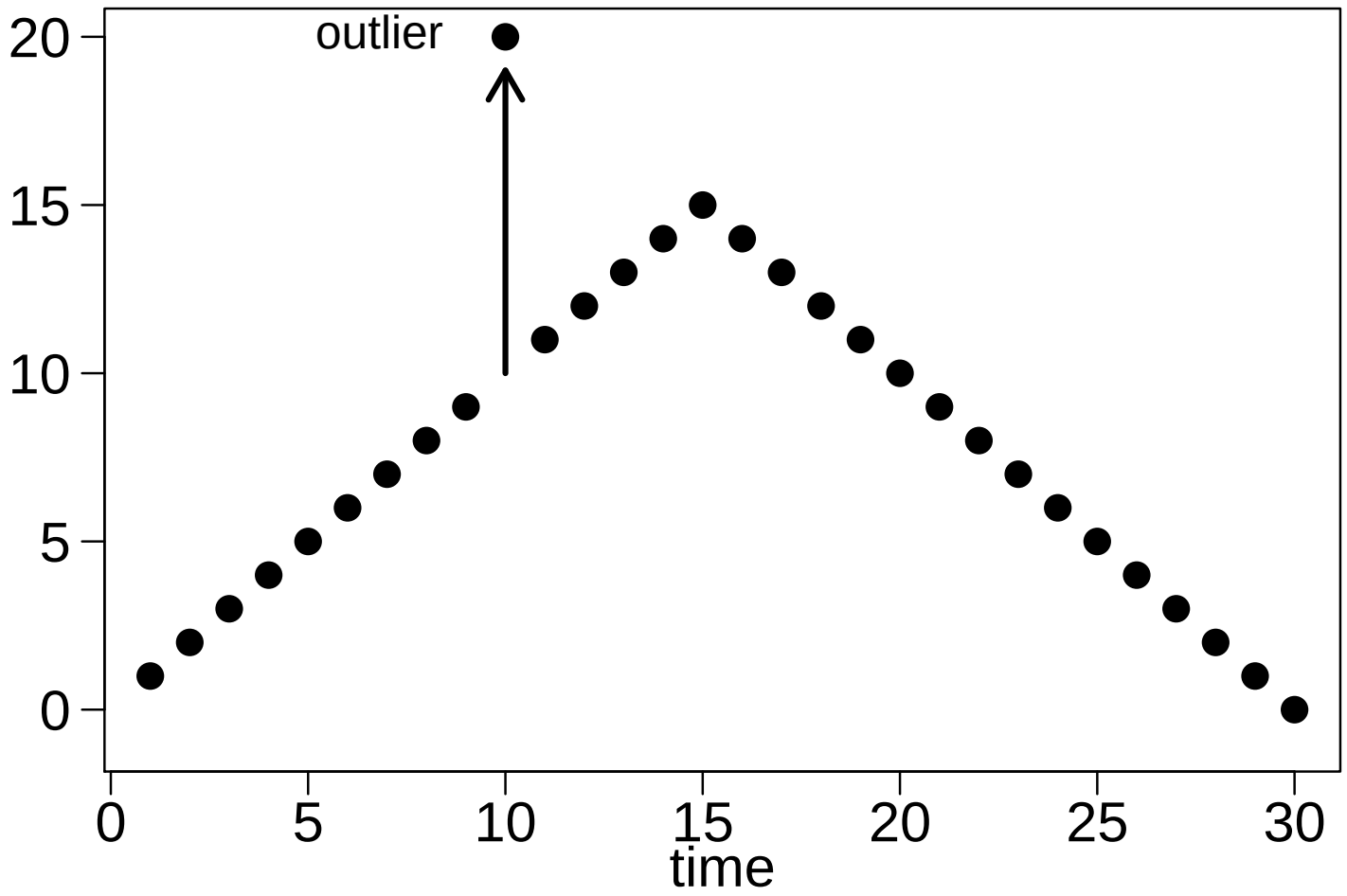
`min.width=11` `p.test=5` `max.width=100`

`rtr=1`: 'restrict to range' of observations in the current window



`adore.filter` – Illustration

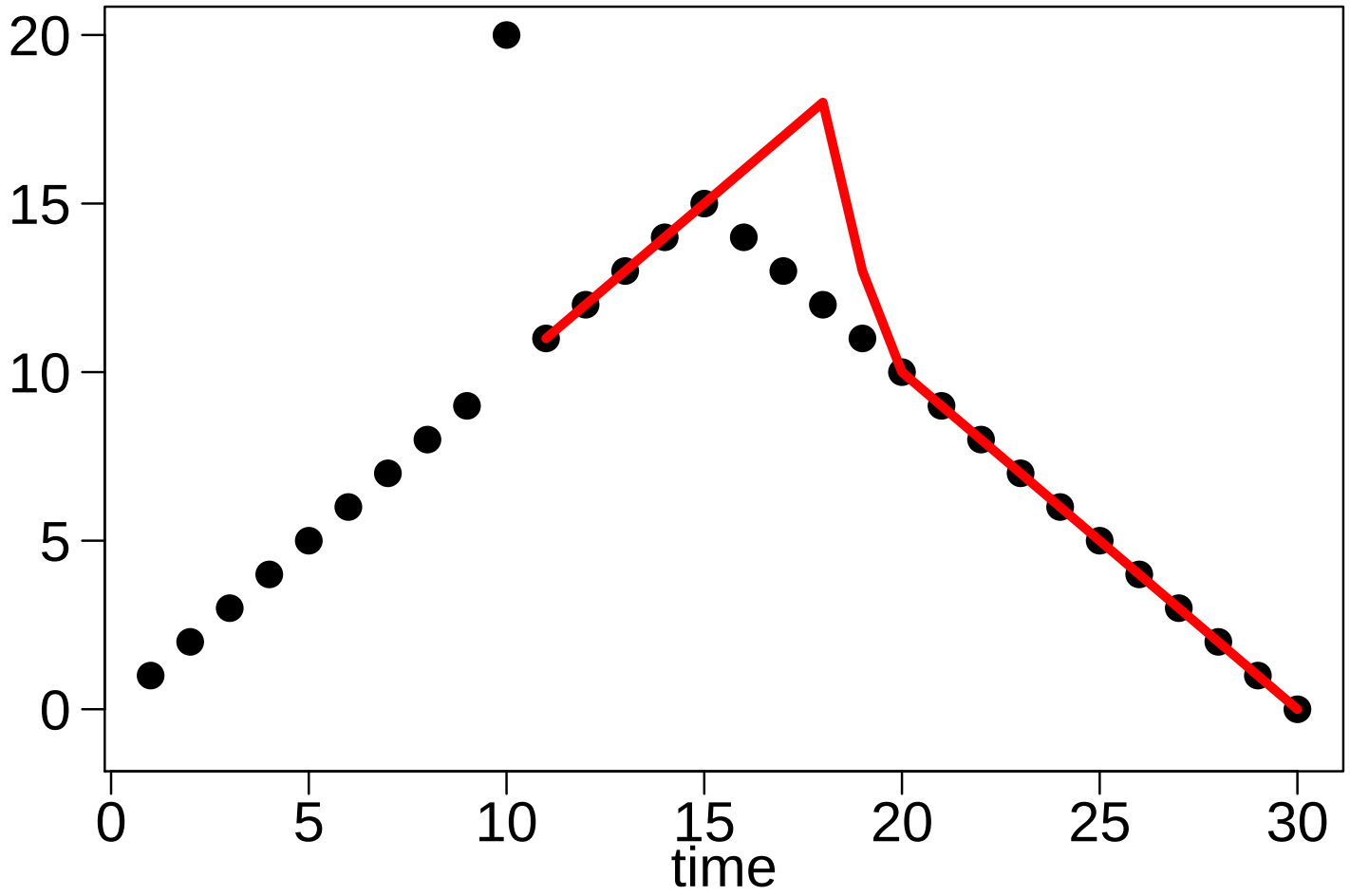
`min.width=11` `p.test=5` `max.width=100`



adore.filter – Illustration

`min.width=11` `p.test=5` `max.width=100`

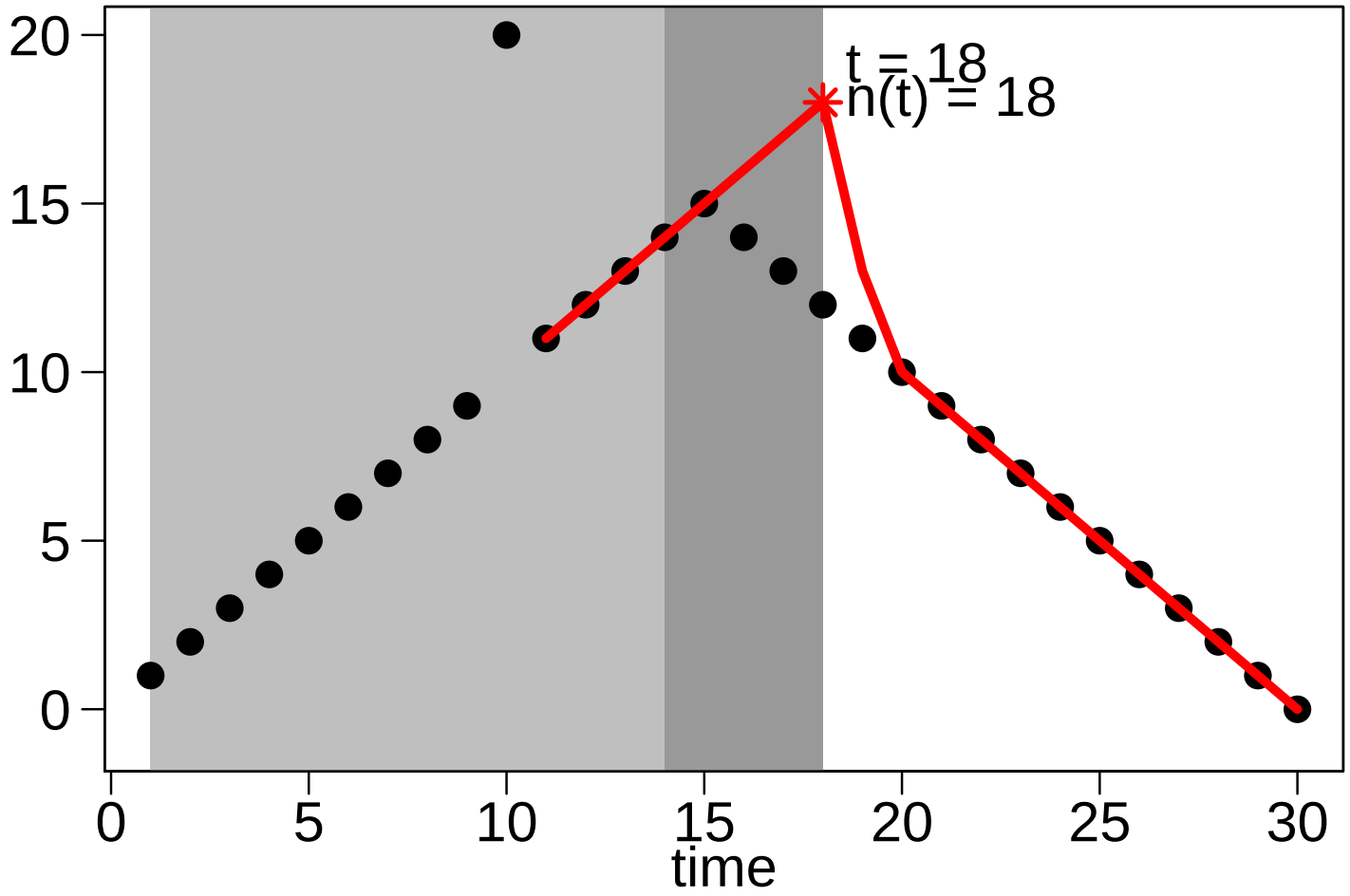
`rtr=1`: 'restrict to range' of observations in the window



adore.filter – Illustration

`min.width=11` `p.test=5` `max.width=100`

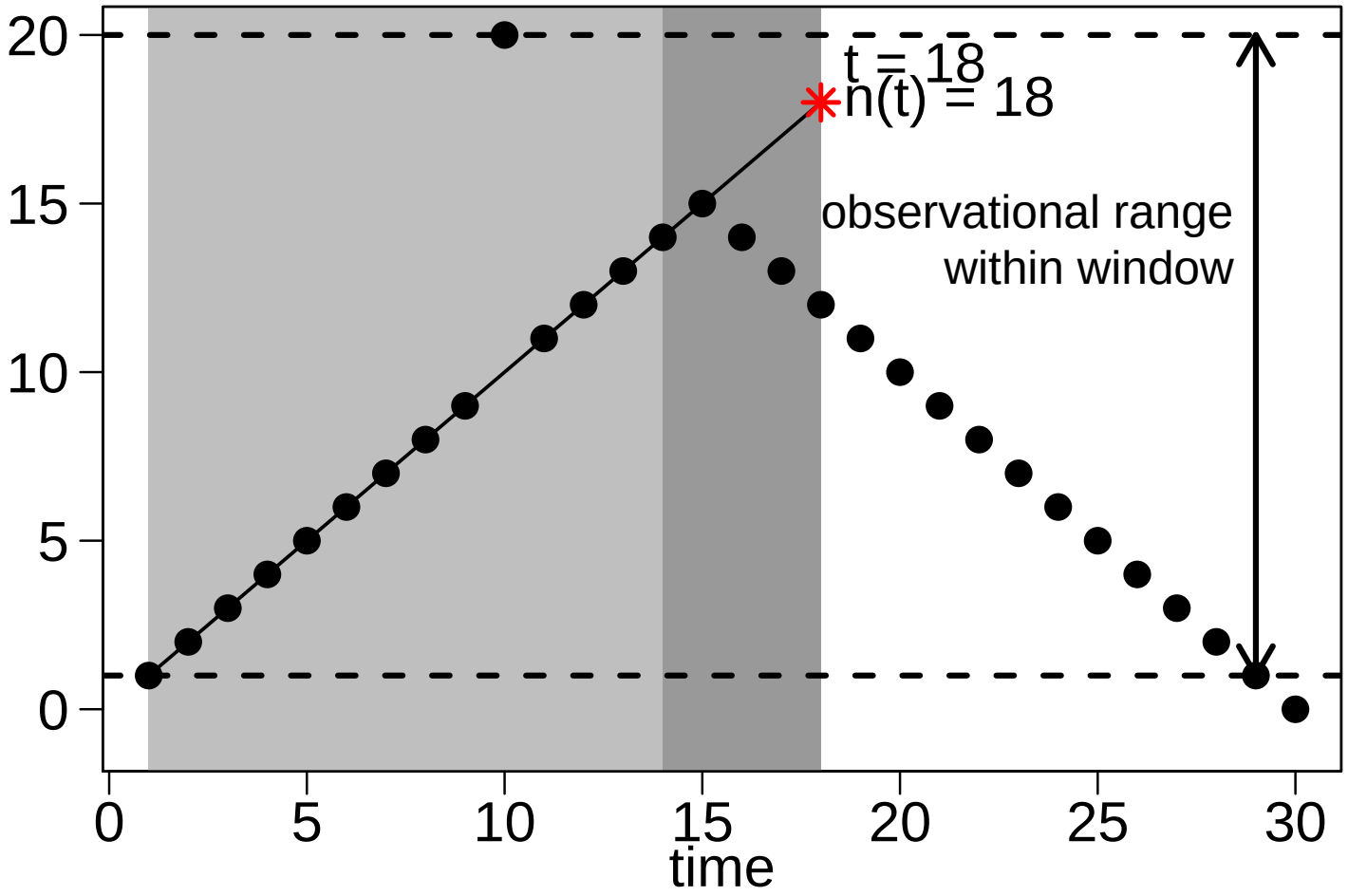
`rtr=1`: 'restrict to range' of observations in the window



adore.filter – Illustration

`min.width=11` `p.test=5` `max.width=100`

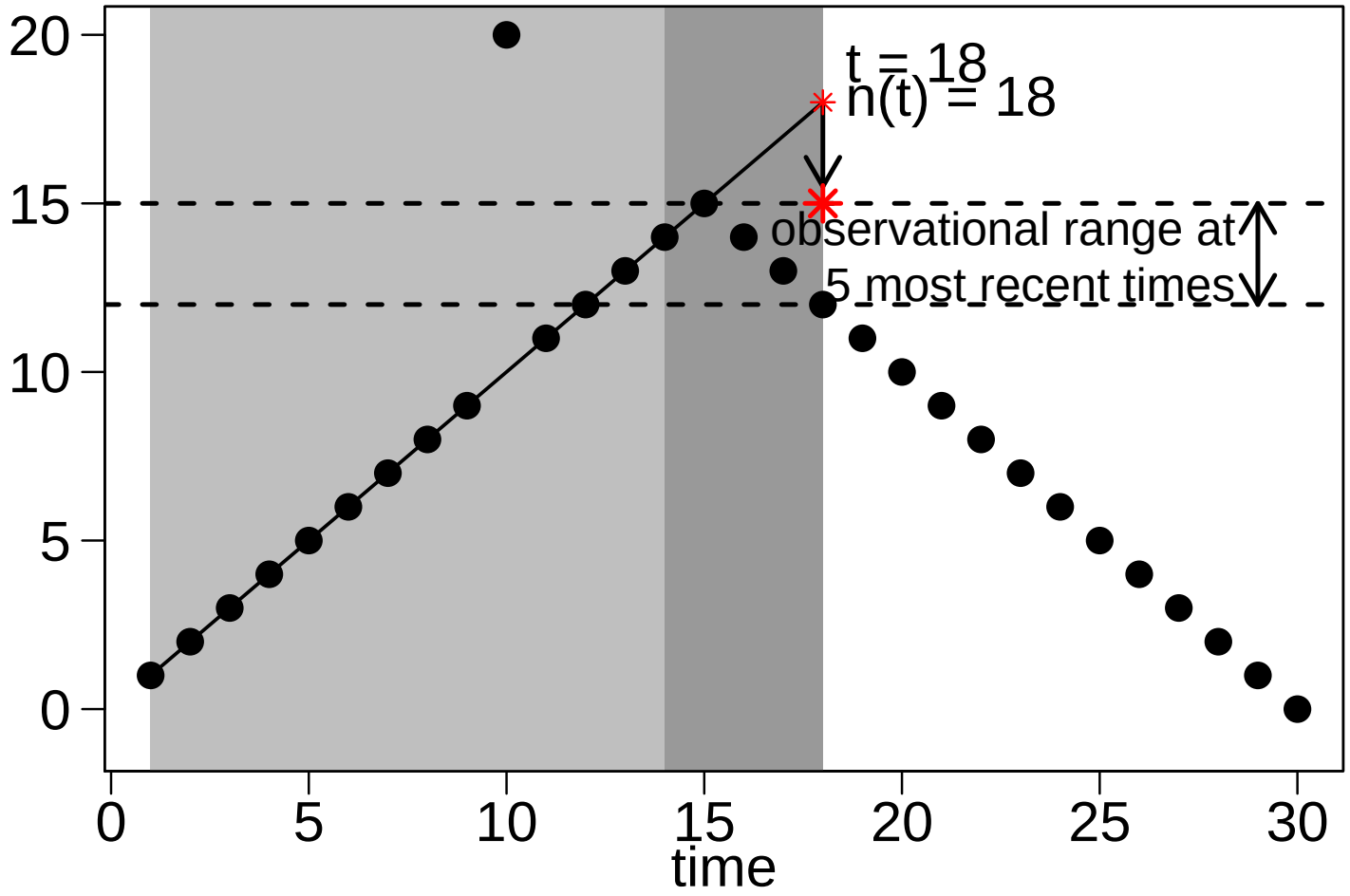
`rtr=1`: 'restrict to range' of observations in the window



adore.filter – Illustration

`min.width=11` `p.test=5` `max.width=100`

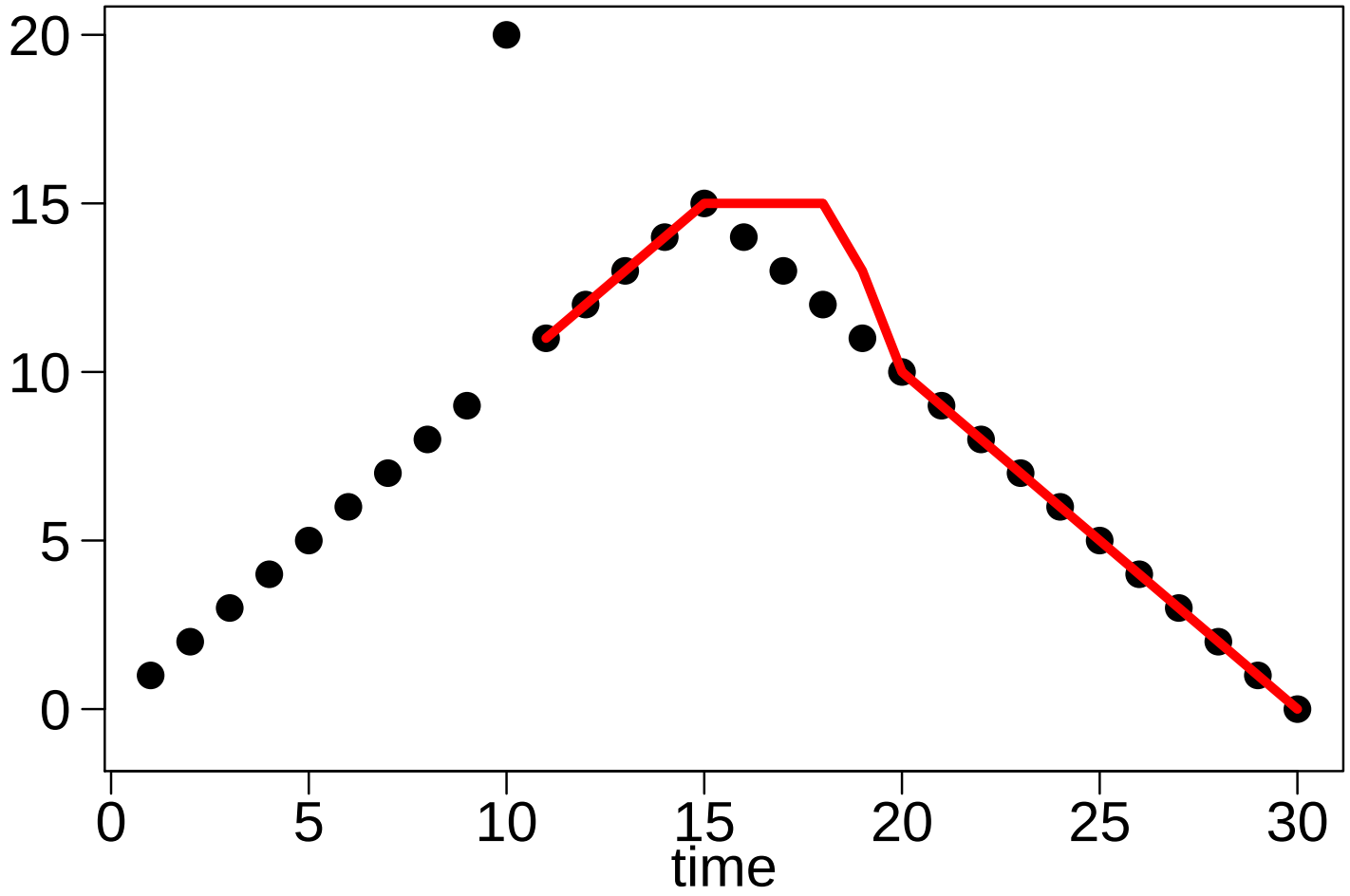
`rtr=2`: 'restrict to range' of `p.test` most recent observations



adore.filter – Illustration

`min.width=11` `p.test=5` `max.width=100`

`rtr=2`: 'restrict to range' of `p.test` most recent observations



madore.filter

(Borowski, Schettlinger, Gather 2008)

Time window of length n_t / k-variate sample with n_t observations

$$(\mathbf{y}_{t+i}) = \begin{pmatrix} y_{1,\,t-n_t+1} & \cdots & y_{1,\,t} \\ y_{2,\,t-n_t+1} & \cdots & y_{2,\,t} \\ \vdots & & \vdots \\ y_{k,\,t-n_t+1} & \cdots & y_{k,\,t} \end{pmatrix}$$

with $i = t - n_t + 1 \qquad , \ldots, t$

madore.filter

(Borowski, Schettlinger, Gather 2008)

Time window of length n_t / k-variate sample with n_t observations

$$(\mathbf{y}_{t+i}) = \begin{pmatrix} y_{1,\, t-n_t+1+\text{lag}} & \cdots & y_{1,\, t+\text{lag}} \\ y_{2,\, t-n_t+1+\text{lag}} & \cdots & y_{2,\, t+\text{lag}} \\ \vdots & & \vdots \\ y_{k,\, t-n_t+1+\text{lag}} & \cdots & y_{k,\, t+\text{lag}} \end{pmatrix}$$

with $i = t - n_t + 1 + \text{lag}, \ldots, t + \text{lag}$

and $\text{lag} = 0$ for online estimation

`madore.filter`

(Borowski, Schettlinger, Gather 2008)

Within a time window of length n_t / a k-variate sample $(\mathbf{y}_{t+i})$:

1. Use adaptive univariate **RM** to find $\widehat{\boldsymbol{\mu}}_t$ and $\widehat{\boldsymbol{\beta}}_t$
and to determine overall window width n_t ($\rightarrow$ `adore.filter`)

2. Estimate local covariance matrix of multivariate residuals $\mathbf{r}_{t+i}$

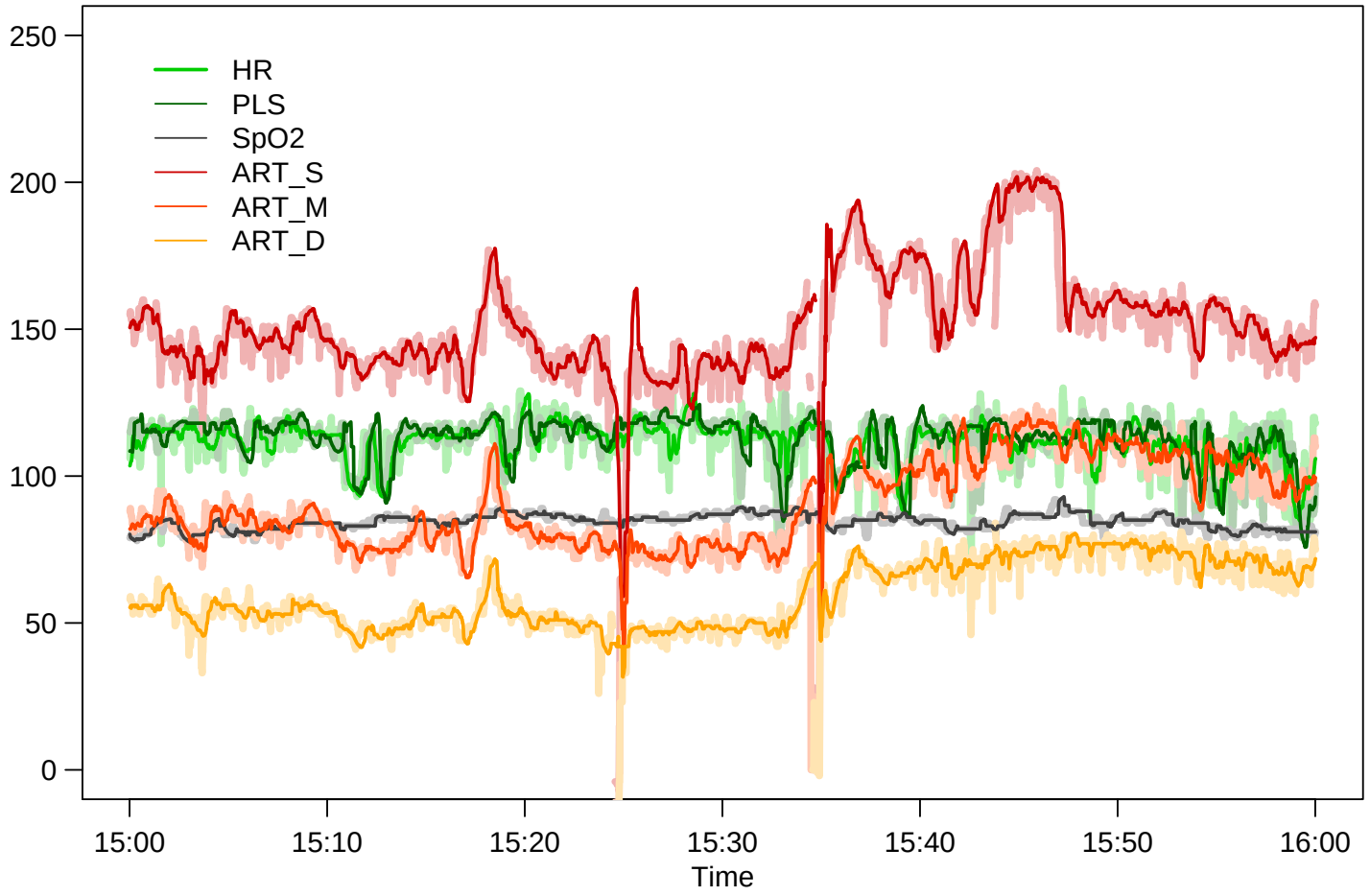
4. Trim observations $\mathbf{y}_{t+i}$ where the Mahalanobis distance of the corresponding residuals $\mathbf{r}_{t+i}$ is larger than d_{n_t} $\left(\stackrel{e.g.}{=} \chi^2_{k;\alpha} \right)$

5. Use multivariate Least Squares on the trimmed observations
to find the level estimate $\widehat{\boldsymbol{\mu}}_t$
($\rightarrow$ `dw.filter`)

(m)adore.filter – Application

`adore.filter(..., min.width=61, p.test=10, rtr=0)`

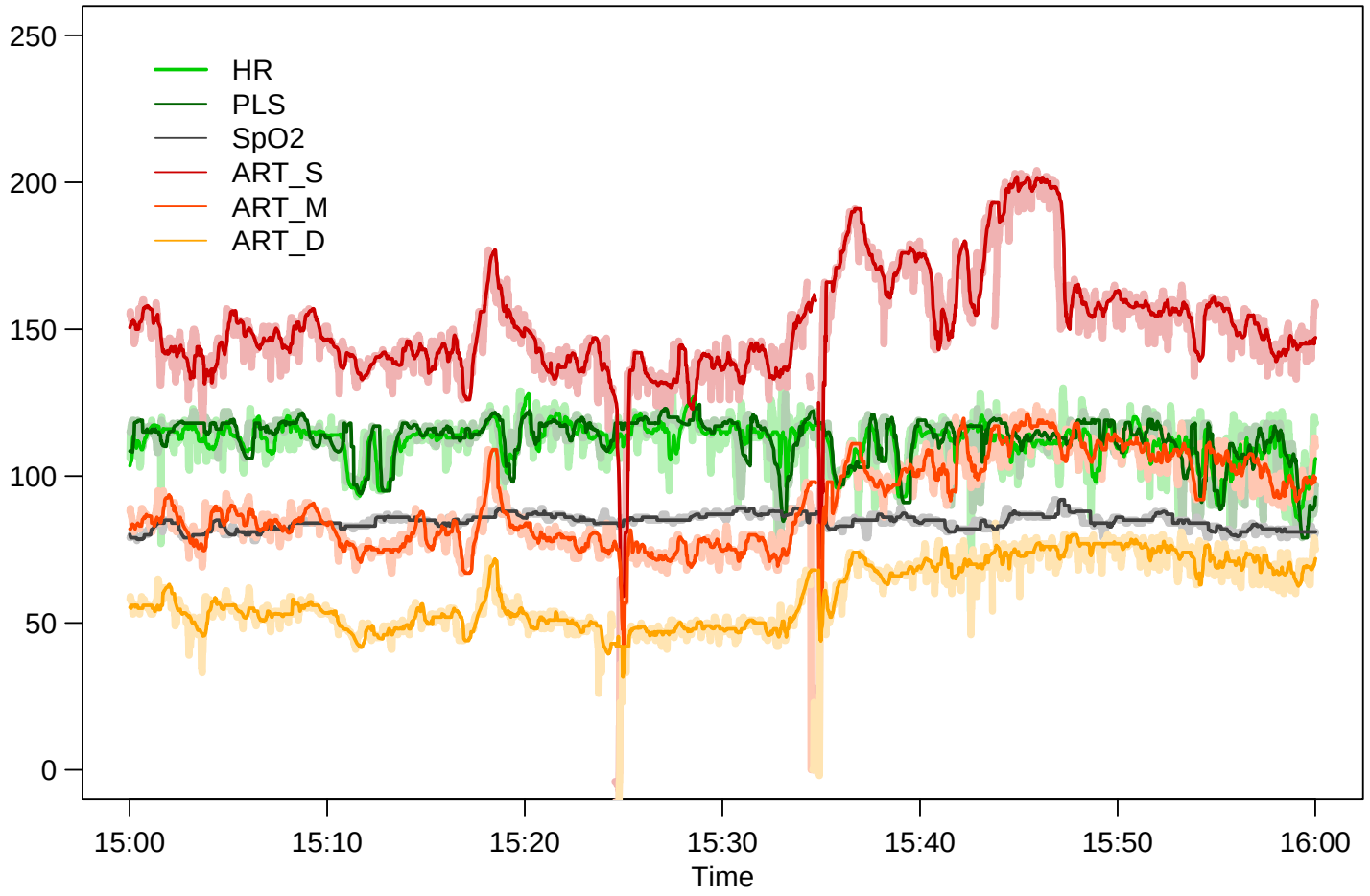
component-wise application



(m)adore.filter – Application

```
adore.filter(..., min.width=61, p.test=10, rtr=1)
```

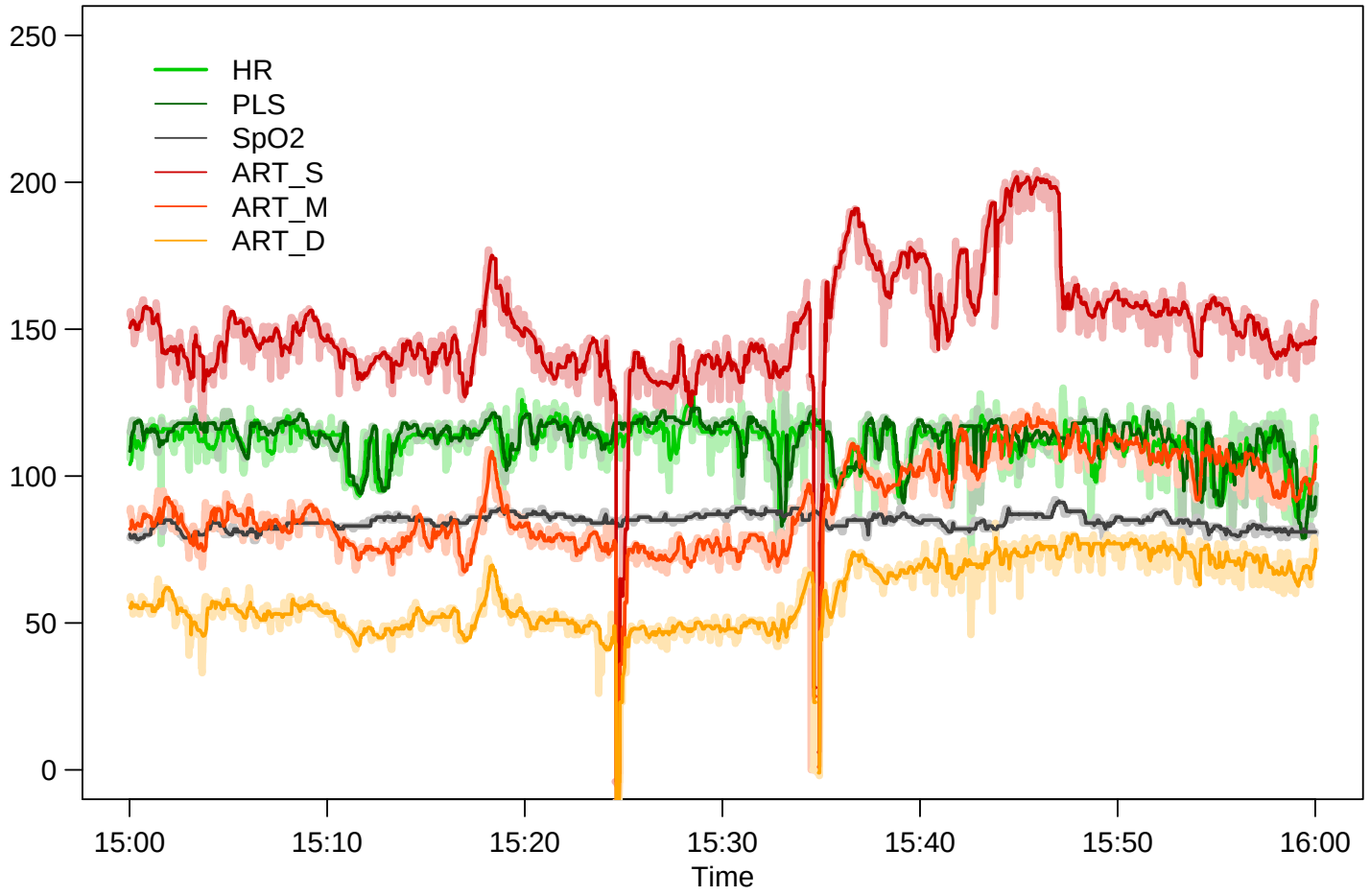
component-wise application



(m)adore.filter – Application

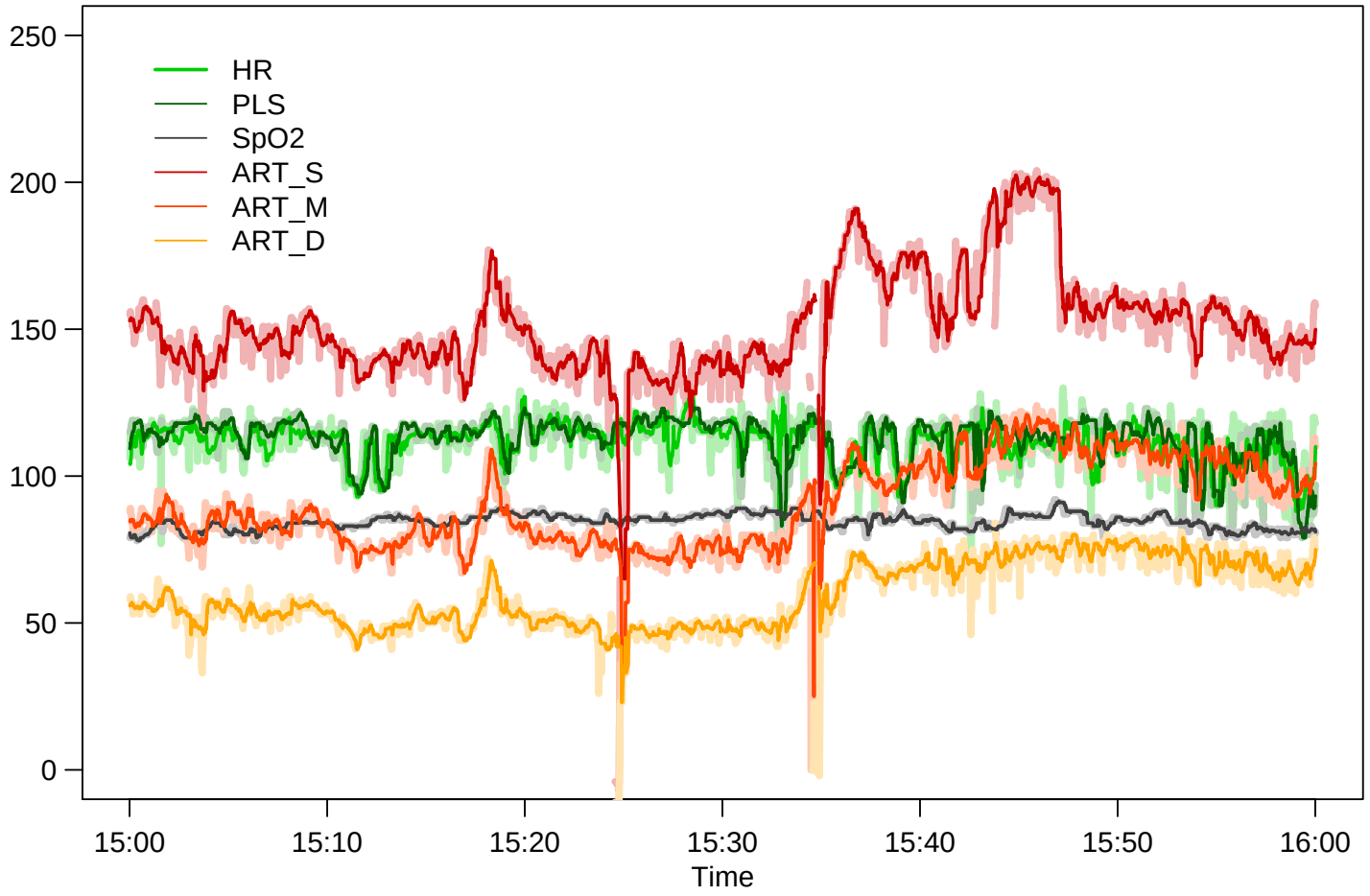
```
adore.filter(..., min.width=61, p.test=10, rtr=2)
```

component-wise application



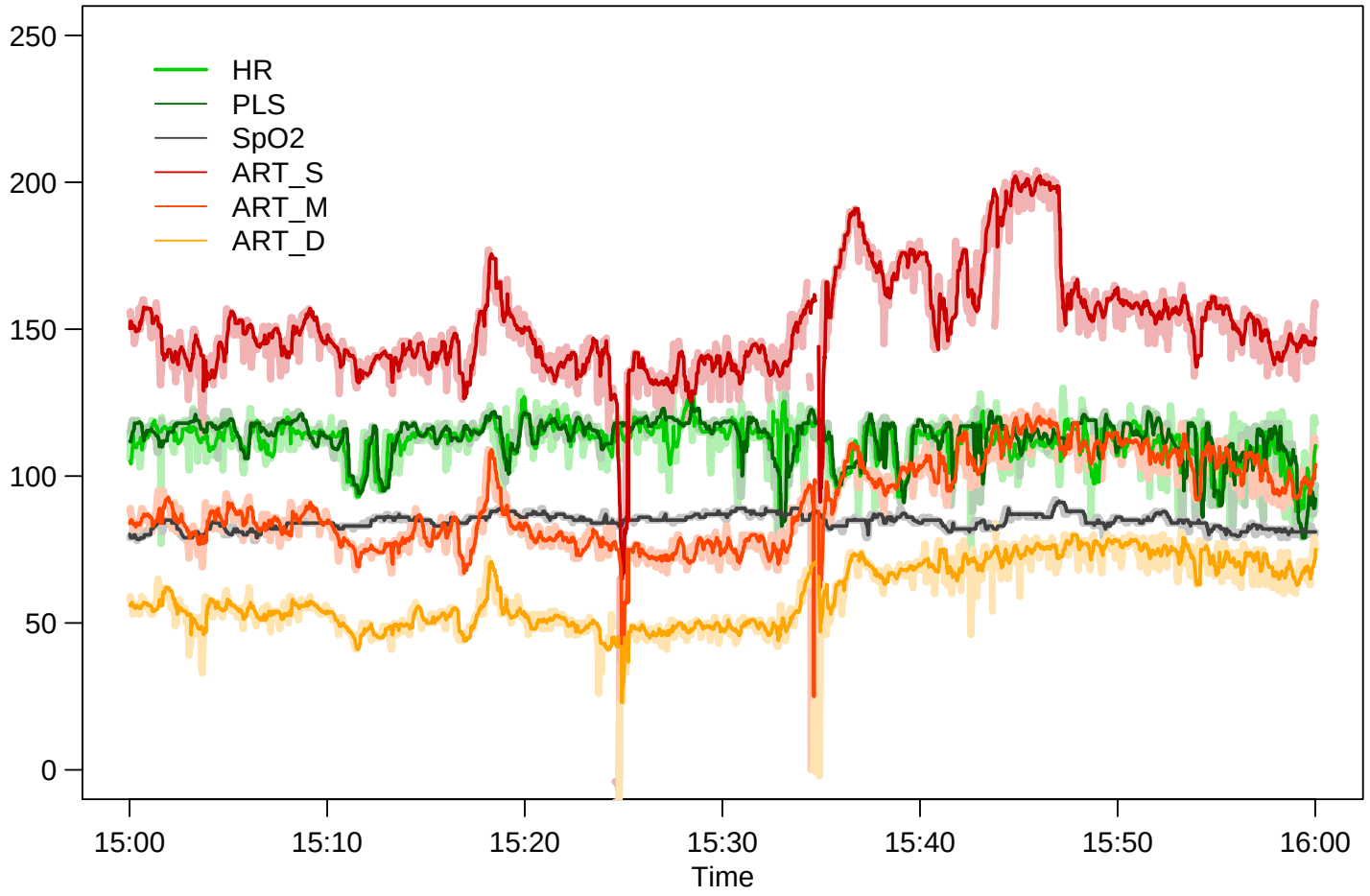
(m)adore.filter – Application

```
madore.filter(..., min.width=61, p.test=10, rtr=2)
```



(m)adore.filter – Application

`madore.filter(..., min.width=61, p.test=10, rtr=2)`
applied to blocks of highly correlated variables



Filter Output

- An object of class `...filter` / a list of the elements:

  `level` filtered signal level

  `slope` corresponding slope within each time window

  Each of these elements is a `data.frame`
  with column names specified by the applied `method`(s).

Filter Output

- An object of class `...filter` / a list of the elements:

  `level` filtered signal level
  `slope` corresponding slope within each time window

  Each of these elements is a `data.frame`
  with column names specified by the applied `method`(s).

- Input parameters

- `scale` output within each time window possible for `hybrid.filter`, `dw.filter`, `robust.filter`, and `adore.filter`

- Filter specific output

Summary

	with delay	online	missing values	adaptive widths	multivariate
`robreg.filter`	✓	✓	✓	✗	✗
`hybrid.filter`	✓	✗	✓	✗	✗
`dw.filter`	✓	✓	✓	✗	✗
`robust.filter`	✓	✓	✗	✓	✗
`wrm.filter`	✓	✓	✓	✗	✗
`adore.filter`	✗	✓	✓	✓	✗
`madore.filter`	✓	✓	✓	✓	✓

Background References

Croux, C., Rousseeuw, P.J. and Hössjer, O. (1994) Generalized S-Estimators. *J. Am. Stat. Assoc.* **89** 1271-1281.

Hampel, F.R. (1975) Beyond Location Parameters: Robust Concepts and Methods. *Bulletin ISI* **46**, 375-382.

Rousseeuw, P.J. (1983) Multivariate Estimation with High Breakdown Point. In: W. Grossmann, G. Pflug, I. Vincze, W. Wertz (eds.) *Proceedings of the 4th Pannonian Symposium on Mathematical Statistics and Probability, Vol. B*, D. Reidel Publishing Company, Dordrecht (The Netherlands).

Rousseeuw, P.J. (1984) Least Median of Squares Regression. *J. Am. Stat. Assoc.* **79**, 871-880.

Rousseeuw, P.J. and Hubert, M. (1999) Regression Depth. *J. Am. Stat. Assoc.* **94**, 388-402.

Rousseeuw, P.J., Croux, C. (1993) Alternatives to the Median Absolute Deviation. *J. Am. Stat. Assoc.* **88**, 1273-1283.

Siegel, A.F. (1982) Robust Regression Using Repeated Medians. *Biometrika* **69**, 242-244.

robfilter-References

robreg.filter:

Davies, P.L., Fried, R., Gather, U. (2004) Robust Signal Extraction for On-line Monitoring Data. *J. Stat. Plann. Inference* **122** (1-2), 65-78.

Gather, U., Schettlinger, K., Fried, R. (2006) Online Signal Extraction by Robust Linear Regression. *Computation. Stat.* **21** (1), 33-52.

hybrid.filter:

Fried, R., Bernholt, T., Gather, U. (2006) Repeated Median and Hybrid Filters. *Comput. Stat. Data An.* **50**, 2313-2338.

dw.filter:

Bernholt, T., Fried, R., Gather, U., Wegener, I. (2006) Modified Repeated Median Filters. *Stat. Comput.* **16**, 177-192.

robust.filter:

Fried, R. (2004) Robust Filtering of Time Series with Trends. *J. Nonparametr. Stat.* **16**, 313-328.

Gather, U., Fried, R. (2004) Methods and Algorithms for Robust Filtering. *COMPSTAT 2004: Proceedings in Computational Statistics*, J. Antoch (eds.), Physika-Verlag, Heidelberg, 159-170.

Fried, R., Gather, U. (2007) On Rank Tests for Shift Detection in Time Series. *Comput. Stat. Data An.* **52**, 221-233.

robfilter-References

wrm.filter and wrm.smooth:

Fried, R., Einbeck, J., Gather, U. (2007) Weighted Repeated Median Smoothing and Filtering. *J. Am. Stat. Assoc.* **102**, 1300-1308.

adore.filter:

Schettlinger, K., Fried, R., Gather, U. (2008) Real Time Signal Processing by Adaptive Repeated Median Filters. *Submitted*.

madore.filter:

Borowski, M., Schettlinger, K., Gather, U. (2008) Multivariate Real Time Signal Processing by a Robust Adaptive Regression Filter. *Submitted*.

Lanius, V., Gather, U. (2007) Robust Online Signal Extraction from Multivariate Time Series. *Technical Report 38/2007*, SFB 475, Technische Universität Dortmund.

Overview

Schettlinger, K., Fried, R., Gather, U. (2006) Robust Filters for Intensive Care Monitoring: Beyond the Running Median, *Biomedizinische Technik* **51**(2), 49-56.