Binary segmentation

Toby Dylan Hocking

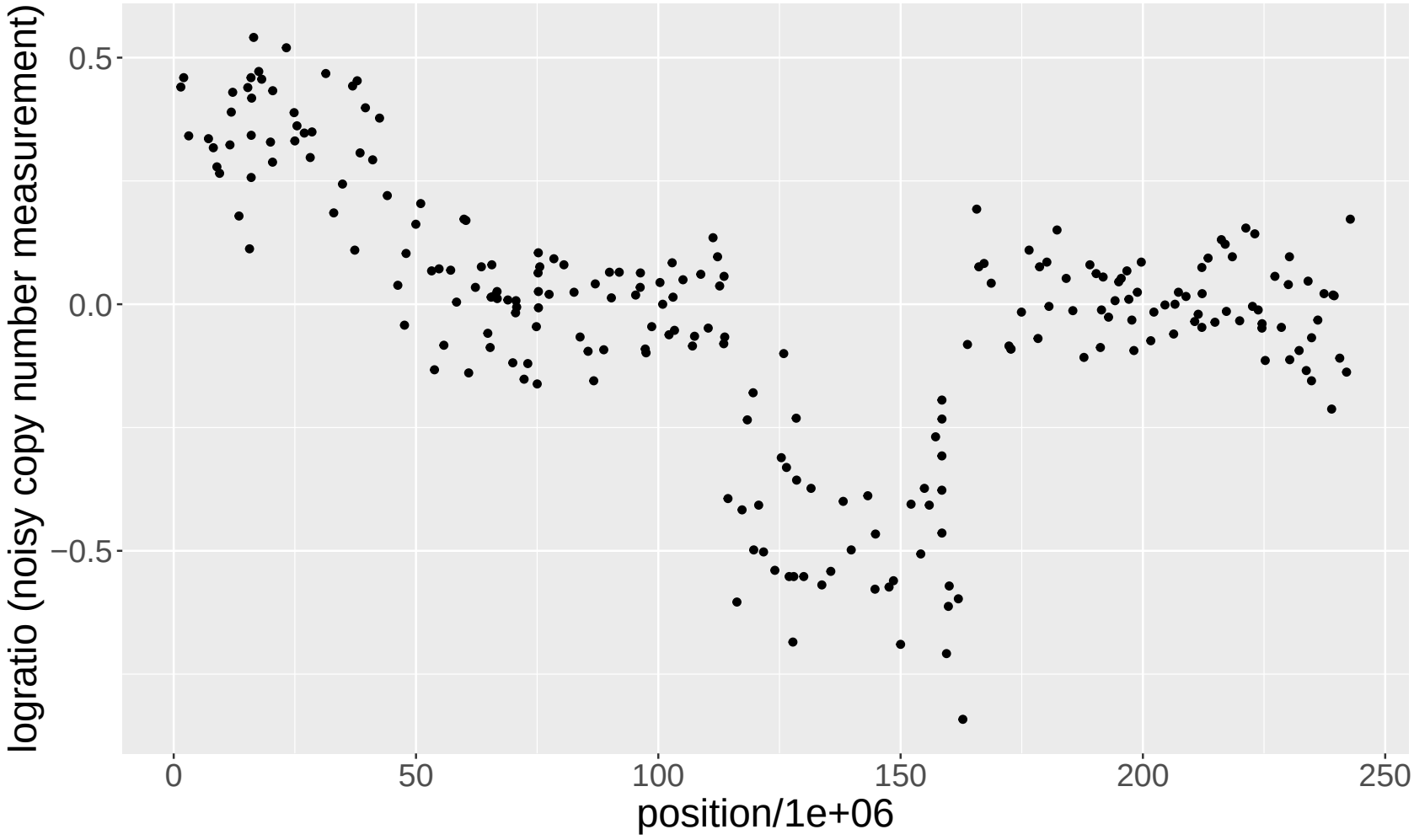
Segmentation / changepoint detection framework

- Let $x_1, \ldots, x_n \in \mathbb{R}$ be a data sequence over space or time.
- Where are the abrupt changes in the data sequence?

```
##       profile.id chromosome  position    logratio
##           <fctr>     <fctr>     <int>       <num>
##    1:          4          2   1472476  0.44042072
##    2:          4          2   2063049  0.45943162
##    3:          4          2   3098882  0.34141652
##    4:          4          2   7177474  0.33571191
##    5:          4          2   8179390  0.31730407
##   ---
## 230:          4          2 239227603  0.01863417
## 231:          4          2 239471307  0.01720929
## 232:          4          2 240618997 -0.10935876
## 233:          4          2 242024751 -0.13764780
## 234:          4          2 242801018  0.17248752
```

Segmentation / changepoint data visualization

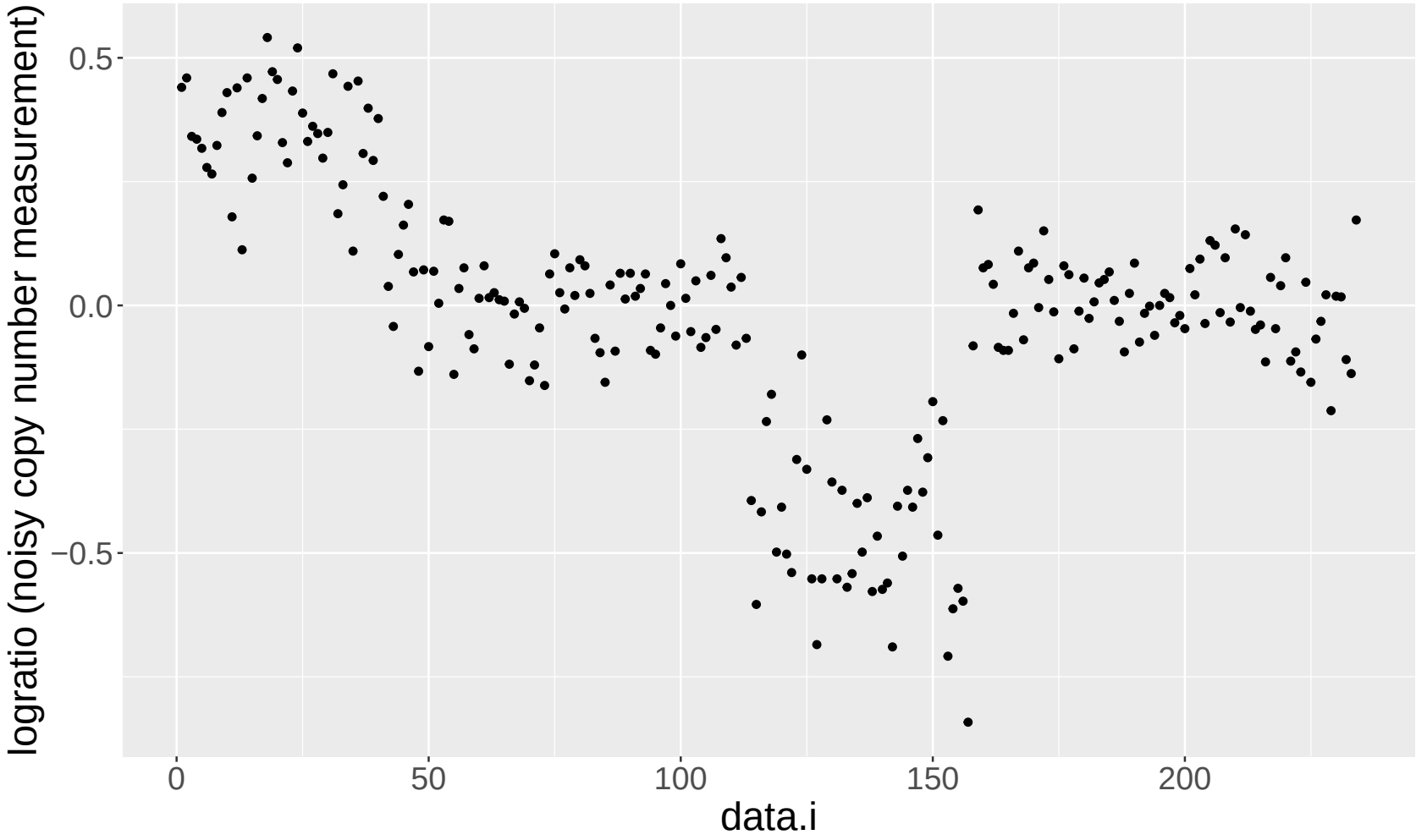
- Let $x_1, \ldots, x_n \in \mathbb{R}$ be a data sequence over space or time.
- Where are the abrupt changes in the data sequence?



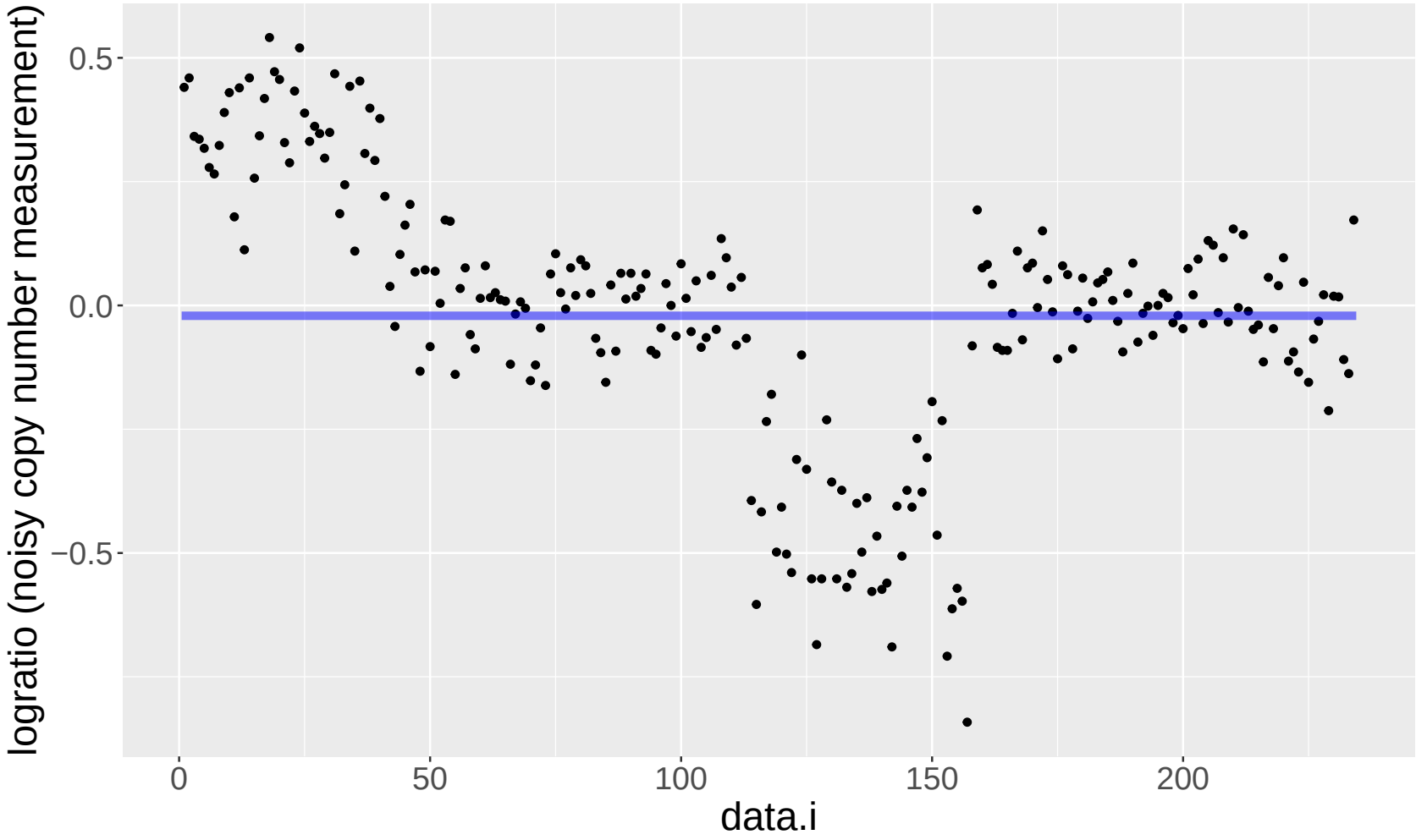
Assume normal distribution with change in mean, constant variance

- There are a certain number of clusters/segments $K \in \{1, \ldots, n\}$.
- Each segment $k \in \{1, \ldots, K\}$ has its own mean parameter $\mu_k \in \mathbb{R}$.
- There is some constant variance parameter $\sigma^2 > 0$ which is common to all segments.
- For each data point i on segment $k \in \{1, \ldots, K\}$ we have $x_i \sim N(\mu_k, \sigma^2)$ – normal distribution.
- This normal distribution assumption means that we want to find segments/changepoints with mean m that minimize the square loss, $(x - m)^2$.

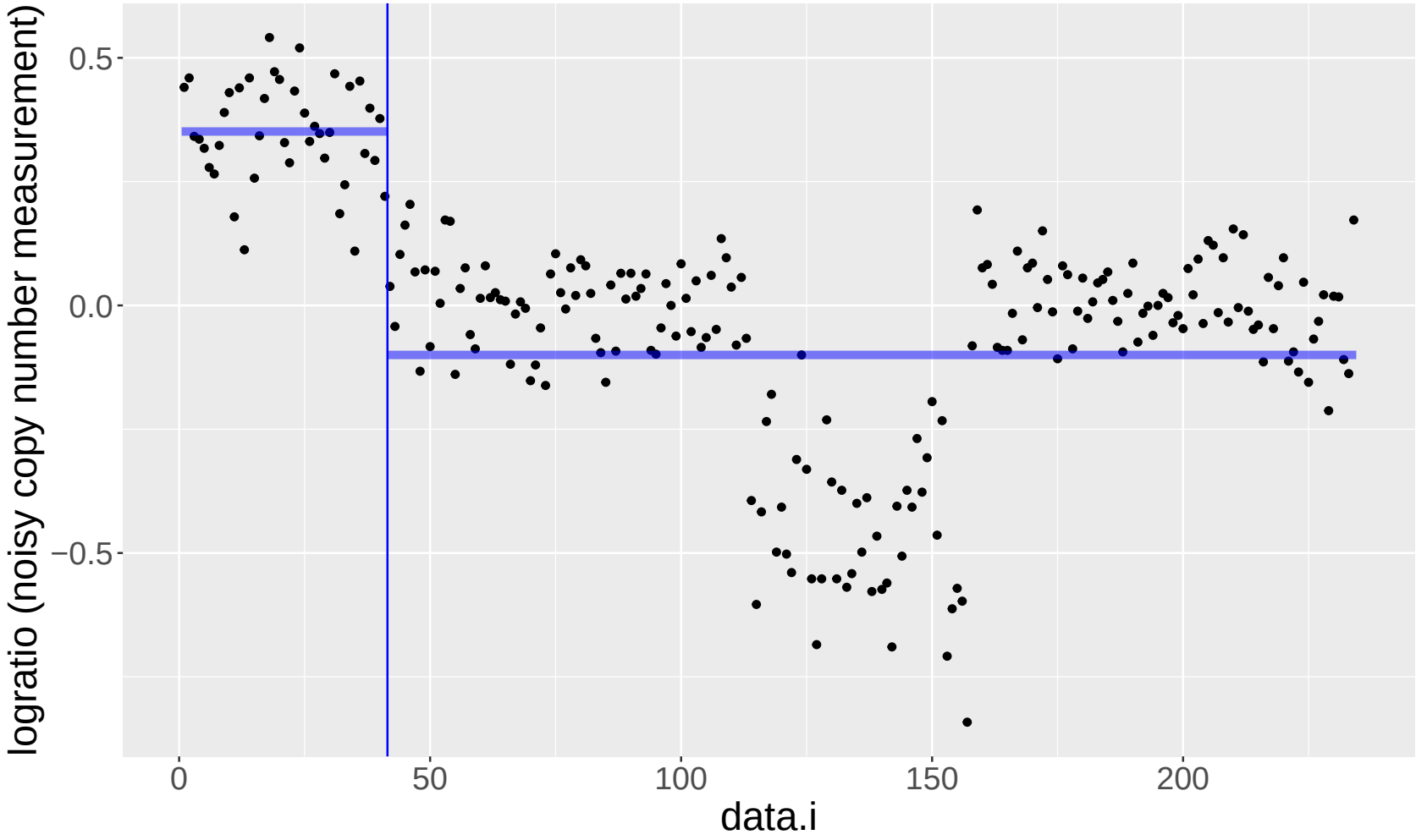
Visualize data sequence



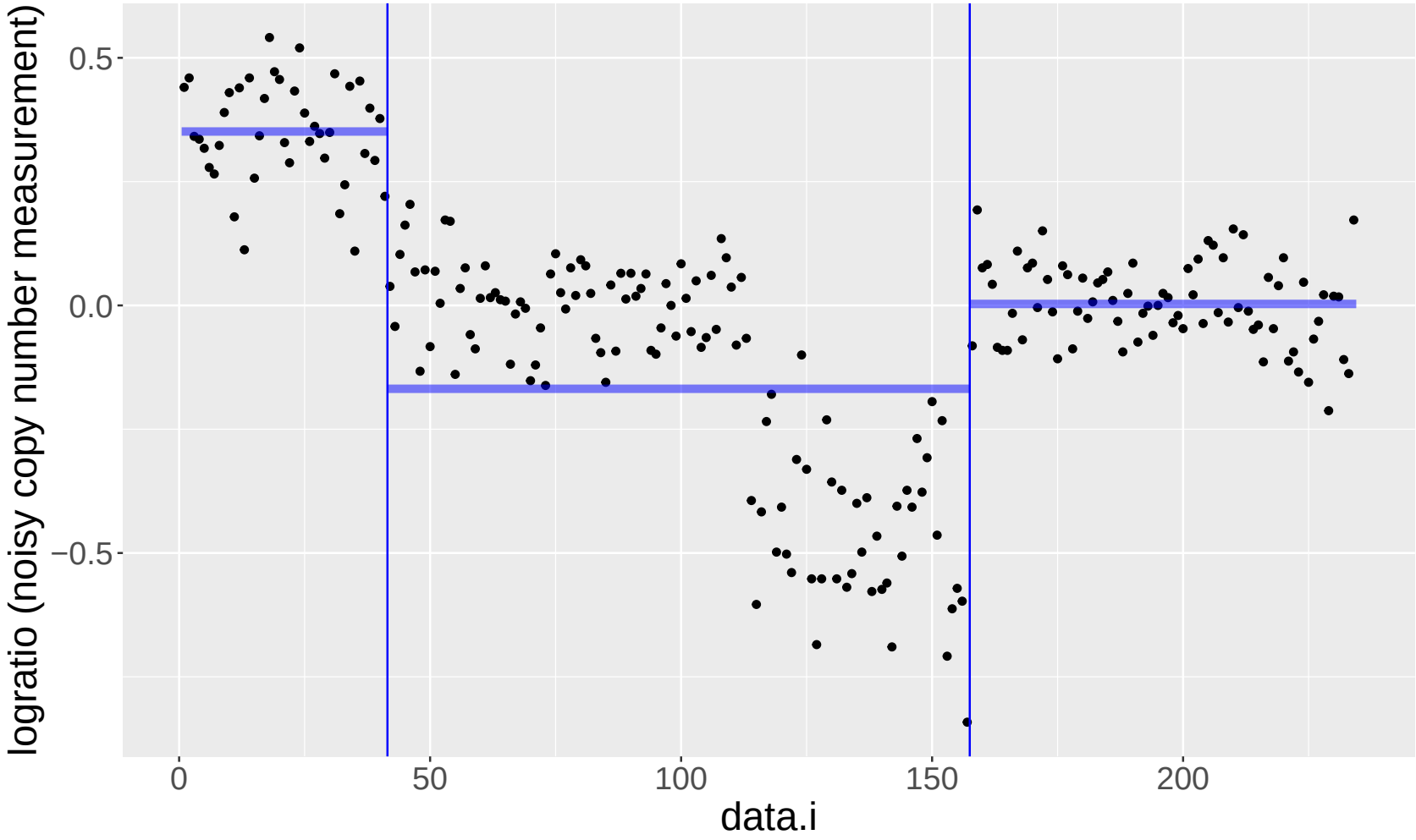
Fit a mean model



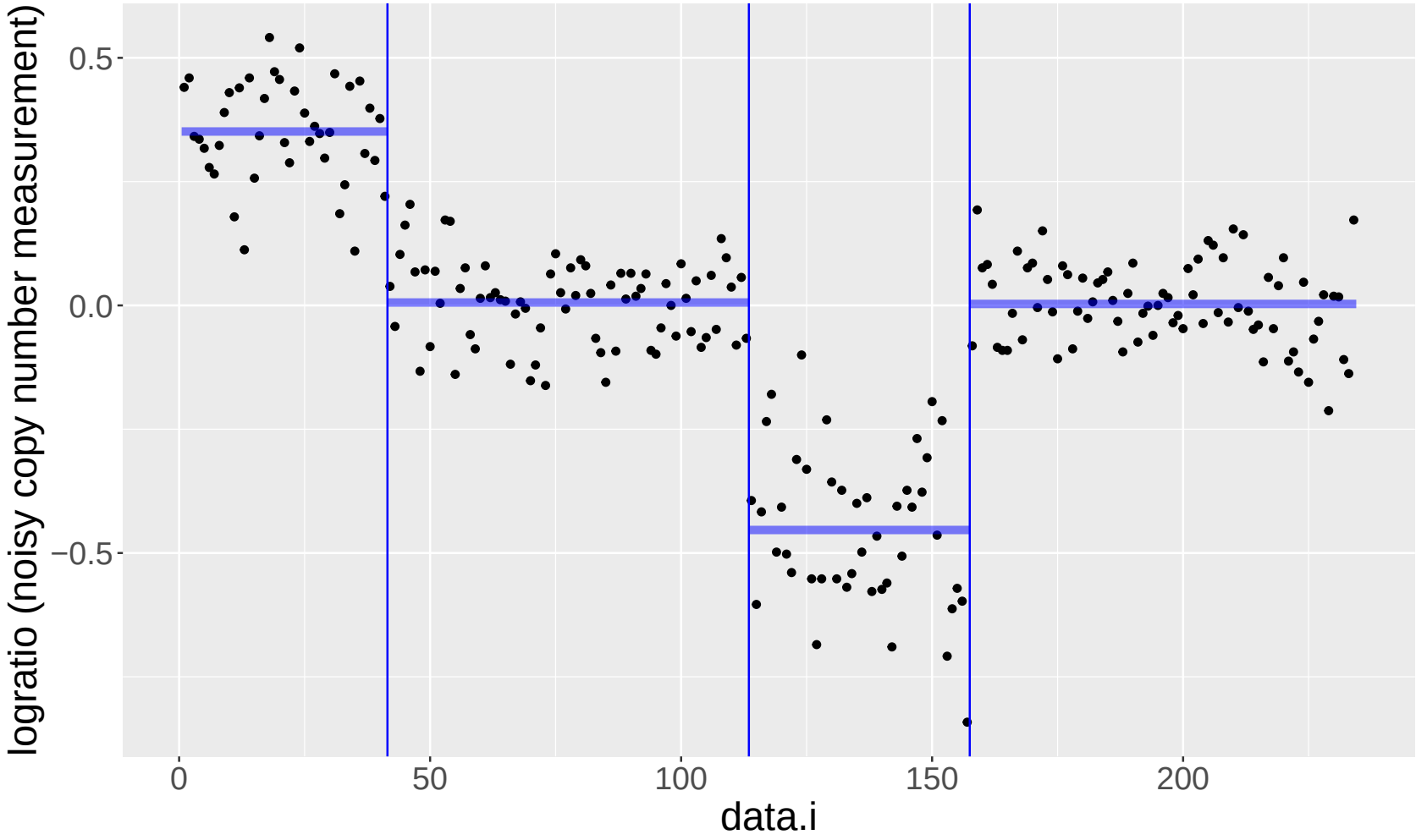
Find best single changepoint (two segments)



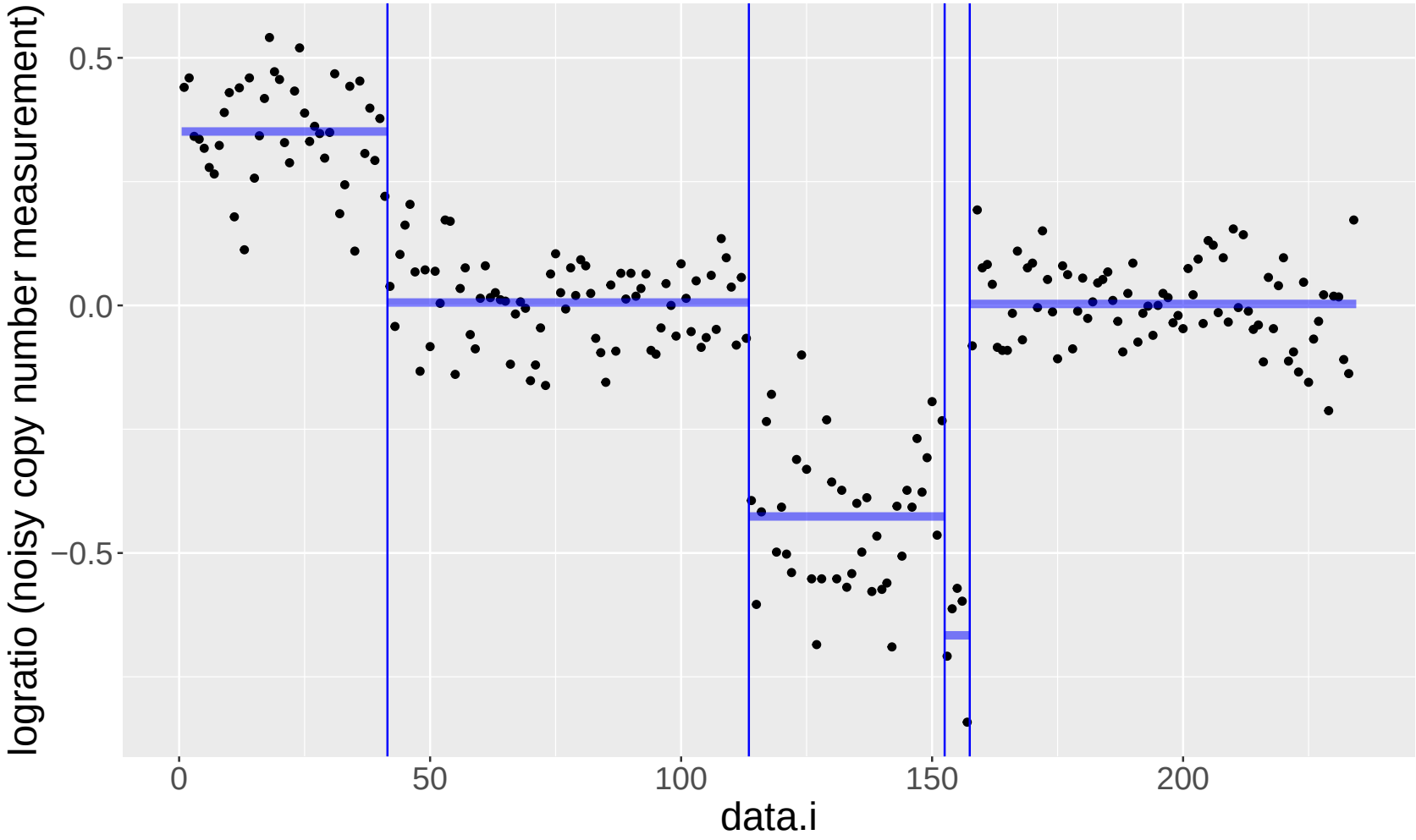
Find two changepoints (three segments)



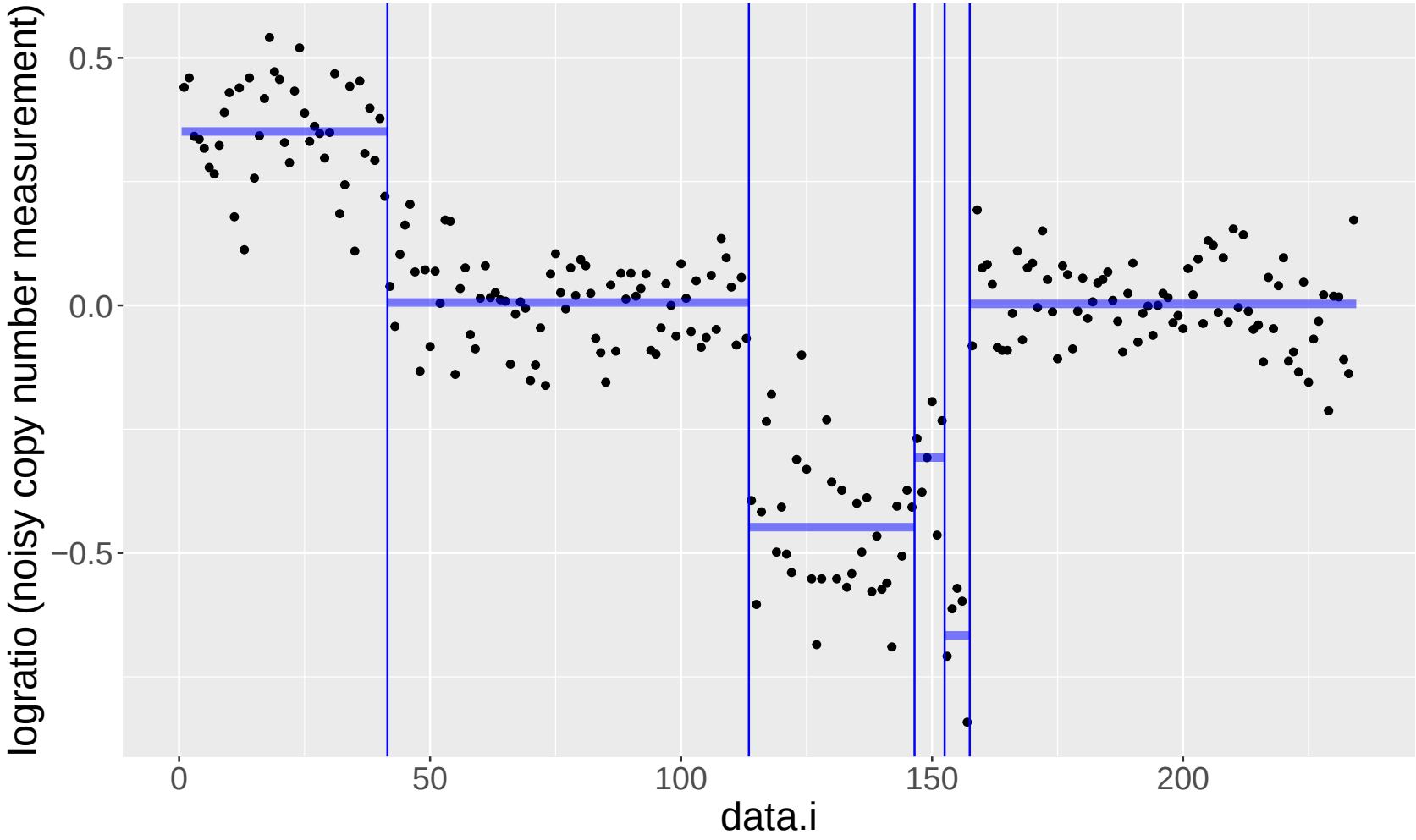
Find four segments



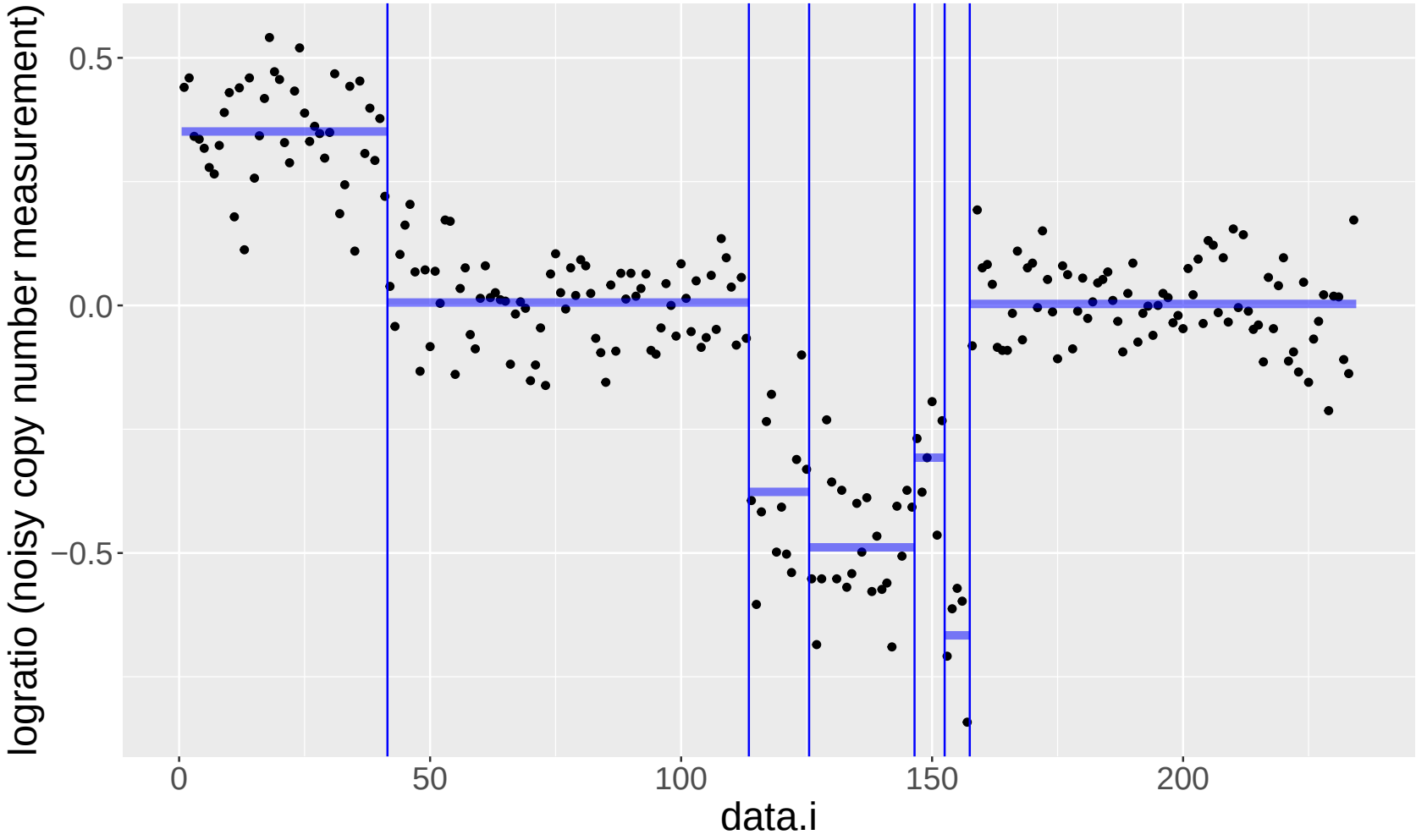
Find five segments



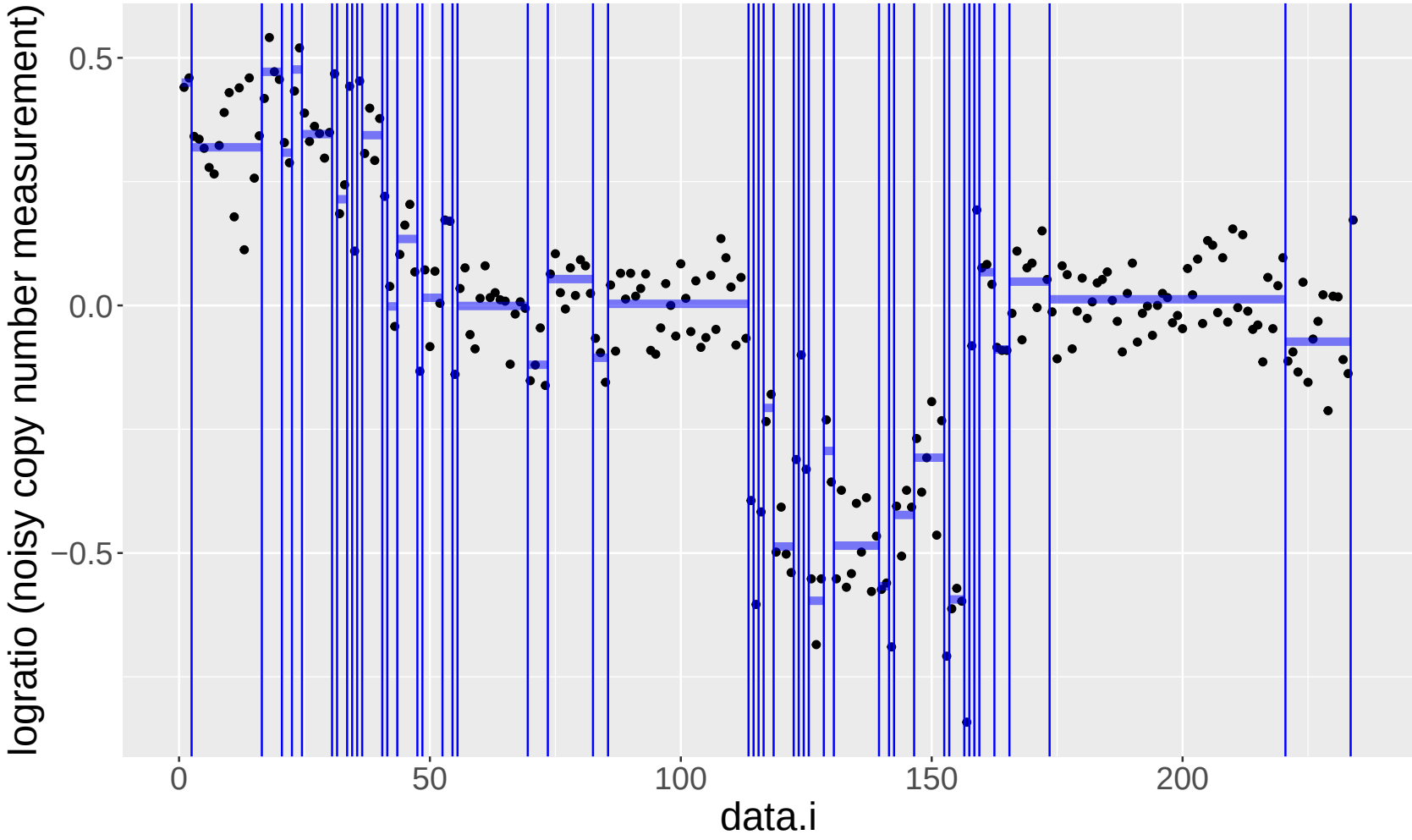
Find six segments



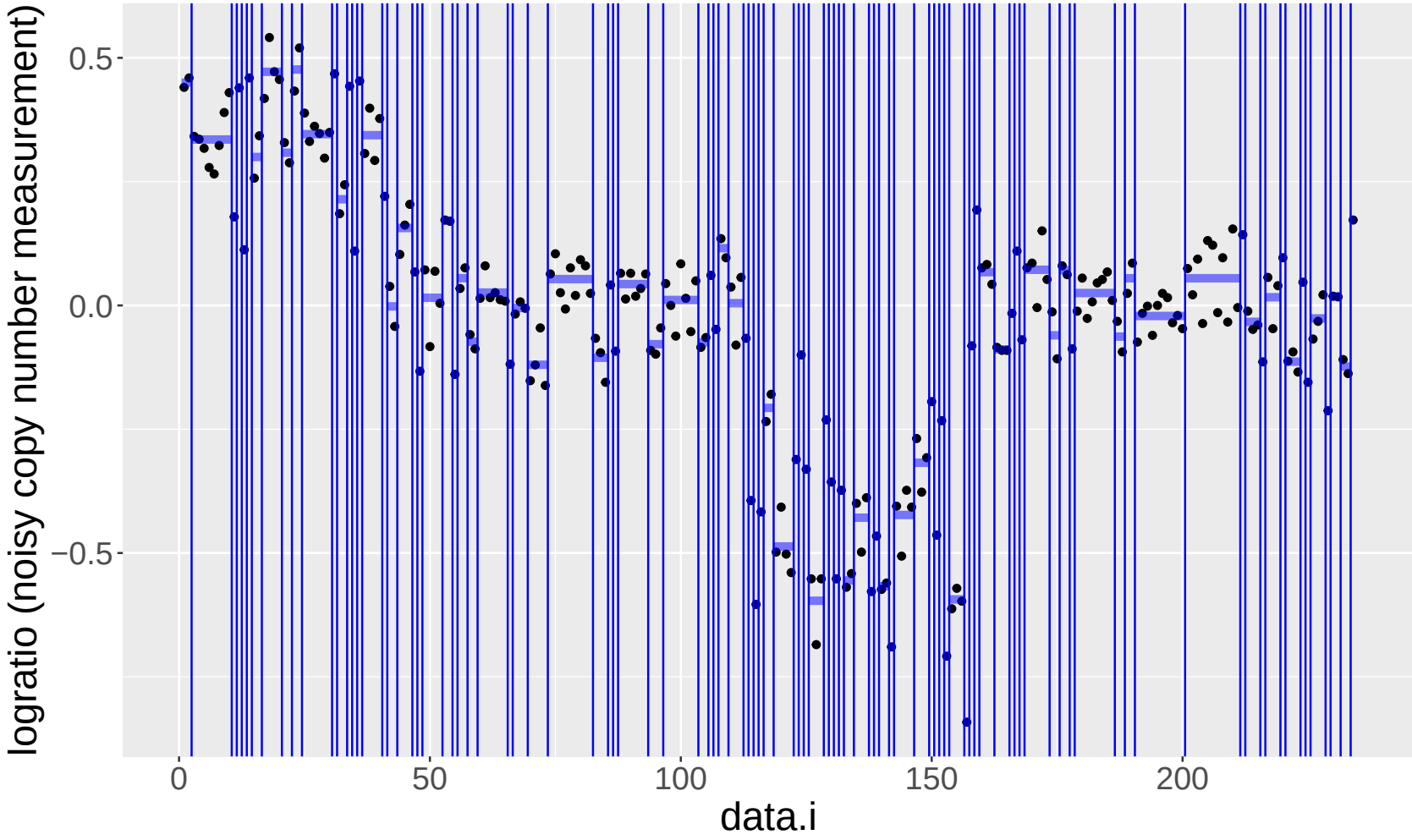
Find seven segments



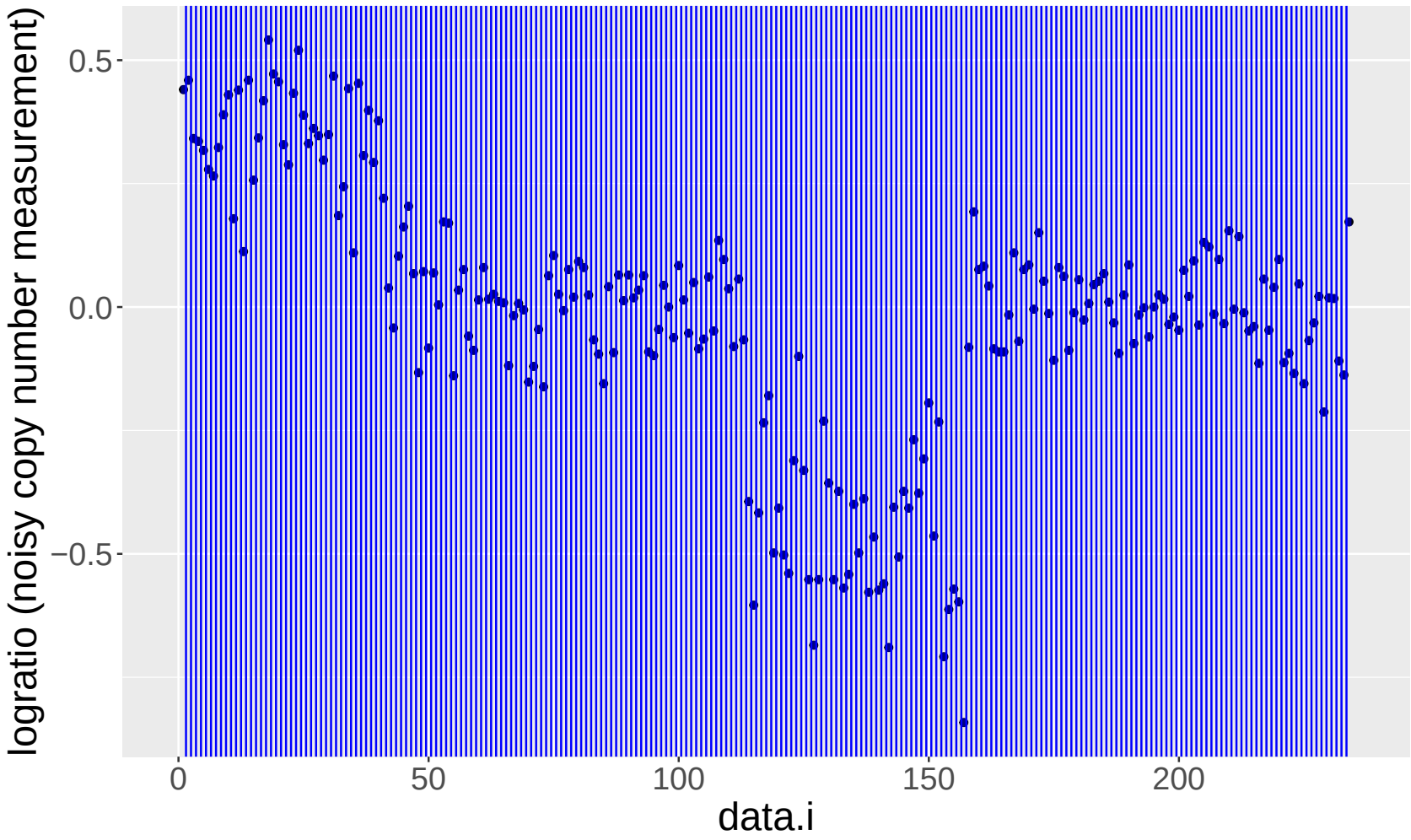
Find 50 segments



Find 100 segments



Find 234 segments

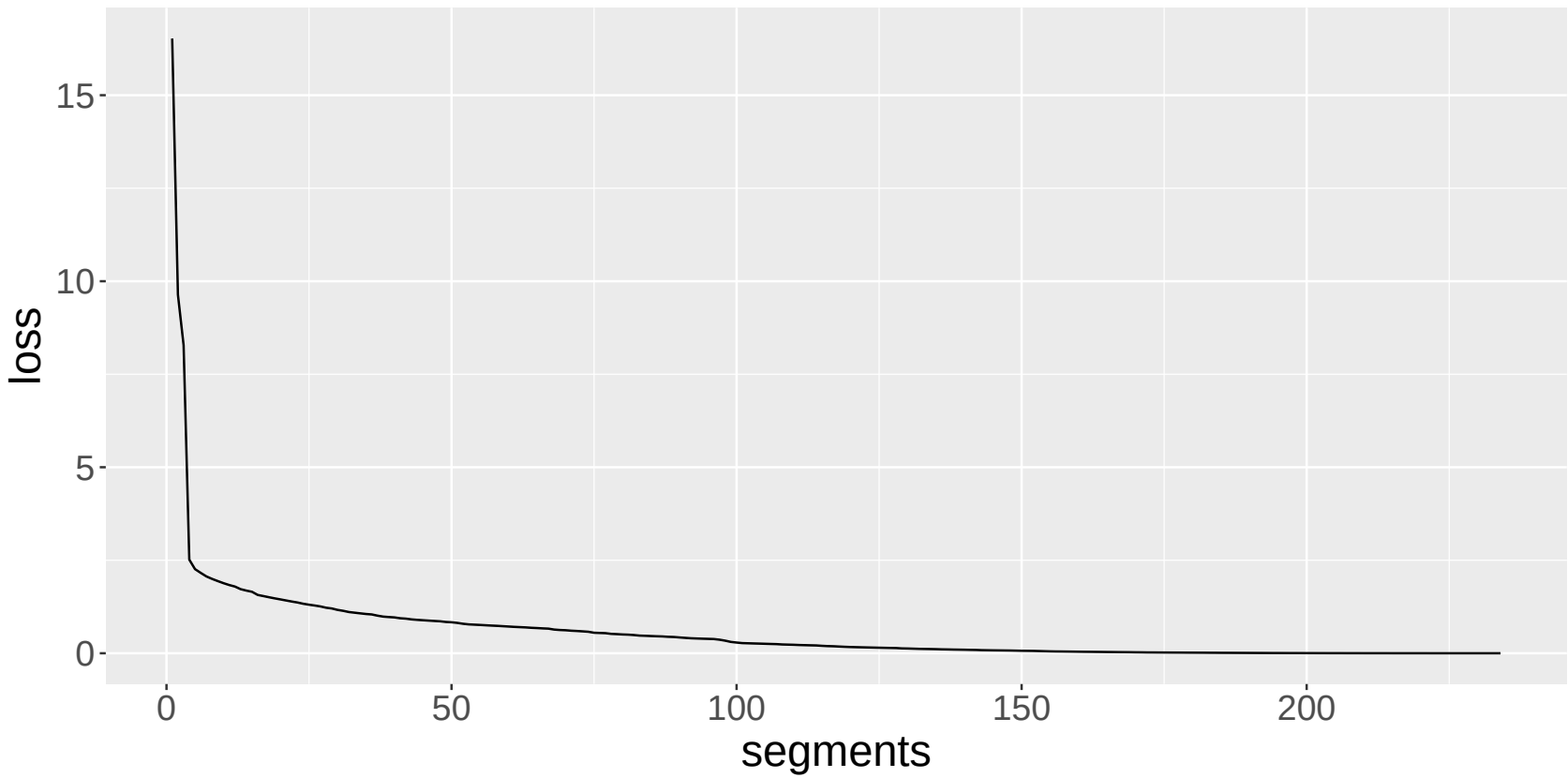


Comparison with previous algorithms from clustering

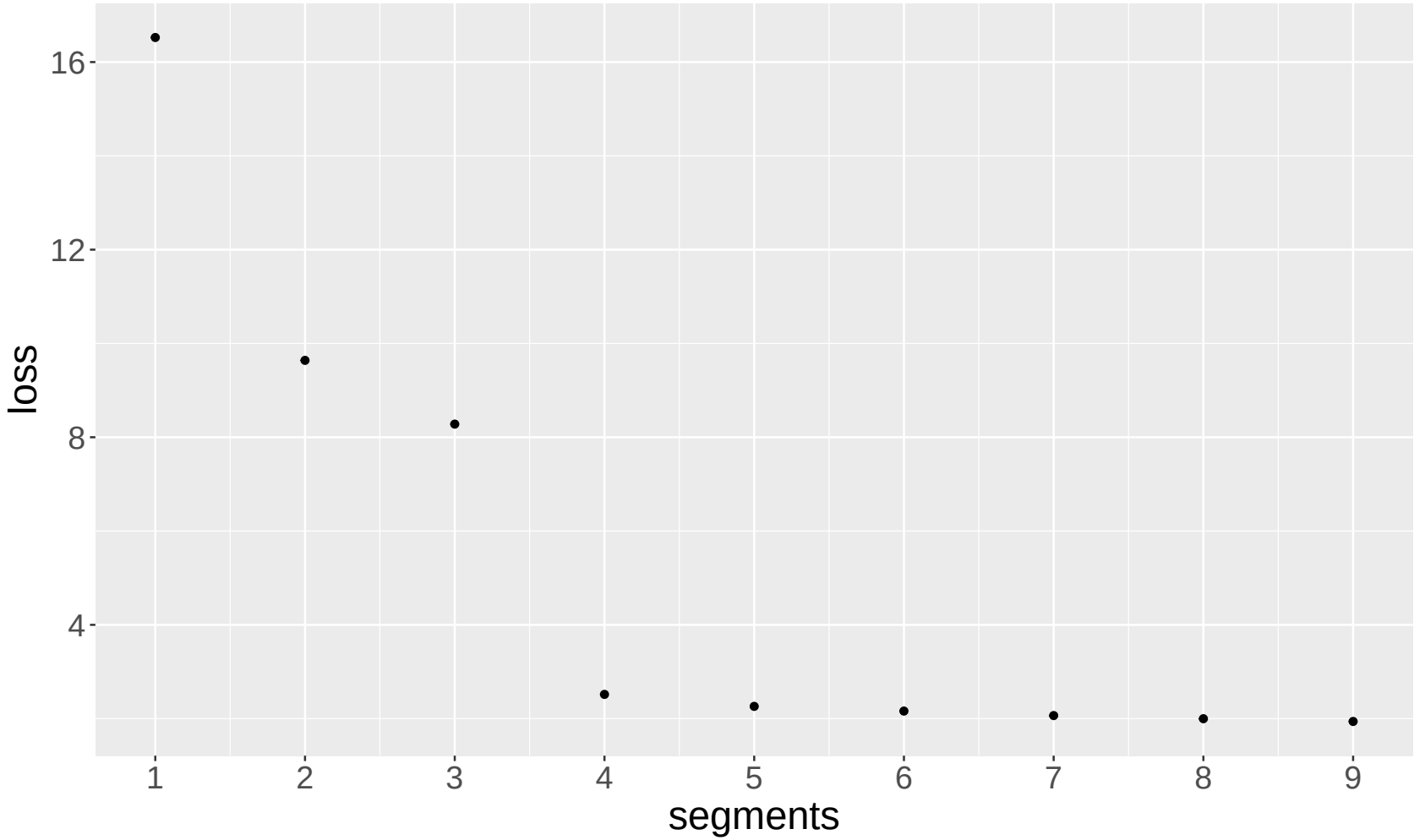
- Binary segmentation has segment/cluster-specific mean parameter, as in K-means and Gaussian mixture models. These algorithms attempt optimization of an error function which measures how well the means fit the data (but are not guaranteed to compute the globally optimal/best model).
- Binary segmentation is deterministic (different from K-means/GMM which requires random initialization). It performs a sequence of greedy minimizations (as in hierarchical clustering).
- Binary segmentation defines a sequence of split operations (from 1 segment to N segments), whereas agglomerative hierarchical clustering defines a sequence of join operations (from N clusters to 1 cluster). Data with common segment mean must be adjacent in time/space; hierarchical clustering joins may happen between any pair of data points (no space/time dimension).

Error/loss function visualization

- Let $m^{(k)} \in \mathbb{R}^n$ be the mean vector with k segments.
- Error for k segments is defined as sum of squared difference between data x and mean $m^{(k)}$ vectors, $E_k = \sum_{i=1}^{n} (x_i - m_i^{(k)})^2$
- As in previous clustering models, kink in the error curve can be used as model selection criterion.



Error/loss function zoom



Learning algorithm

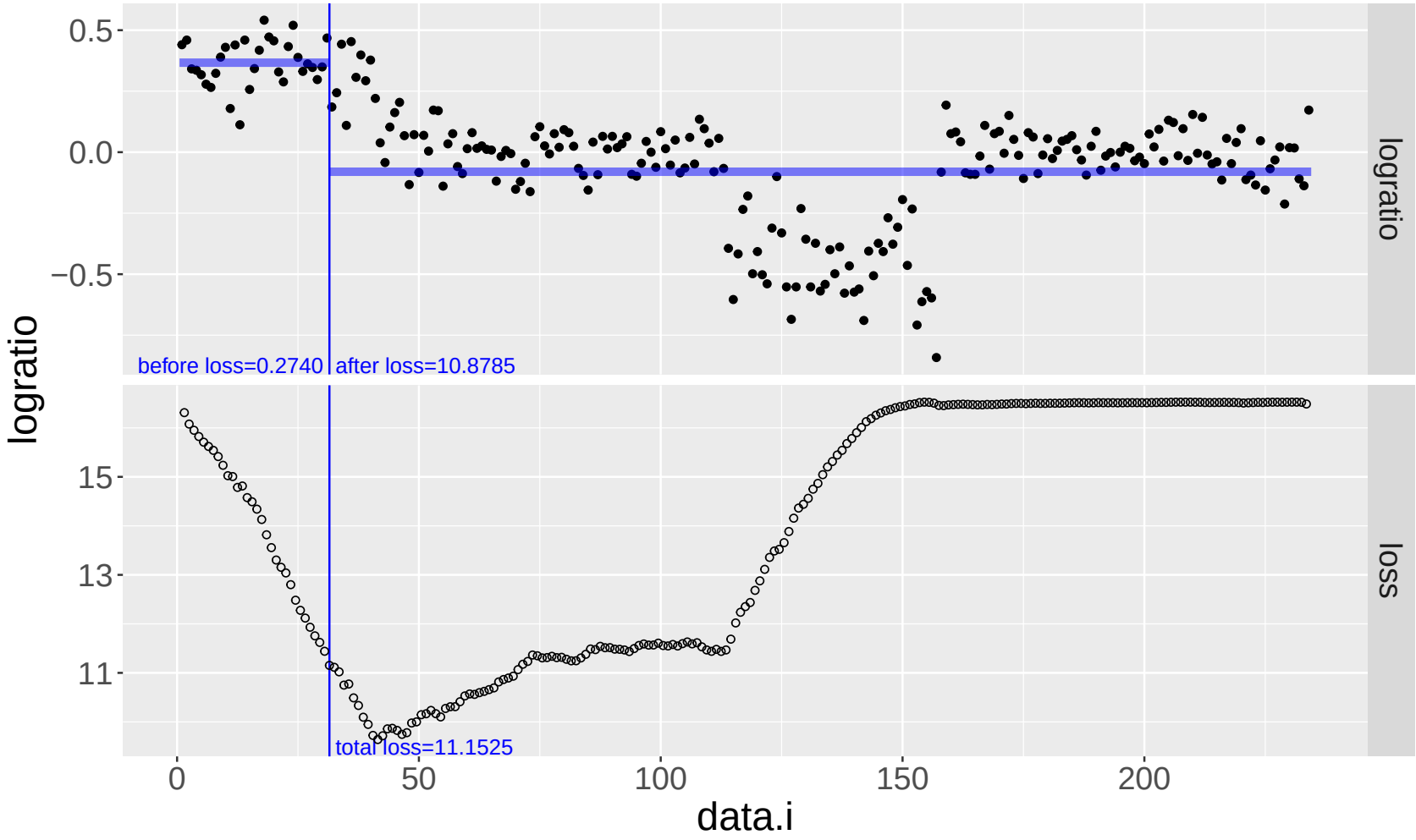
- Start with one segment, then repeat:
- Compute loss of each possible split.
- Choose split which results in largest loss decrease.
- If $s = \sum_{i=1}^{n} x_i$ is the sum over n data points, then the mean is s/n and the square loss (from 1 to n) is

$$L_{1,n} = \sum_{i=1}^{n}(x_i - s/n)^2 = \sum_{i=1}^{n}[x_i^2] - 2(s/n)s + n(s/n)^2$$

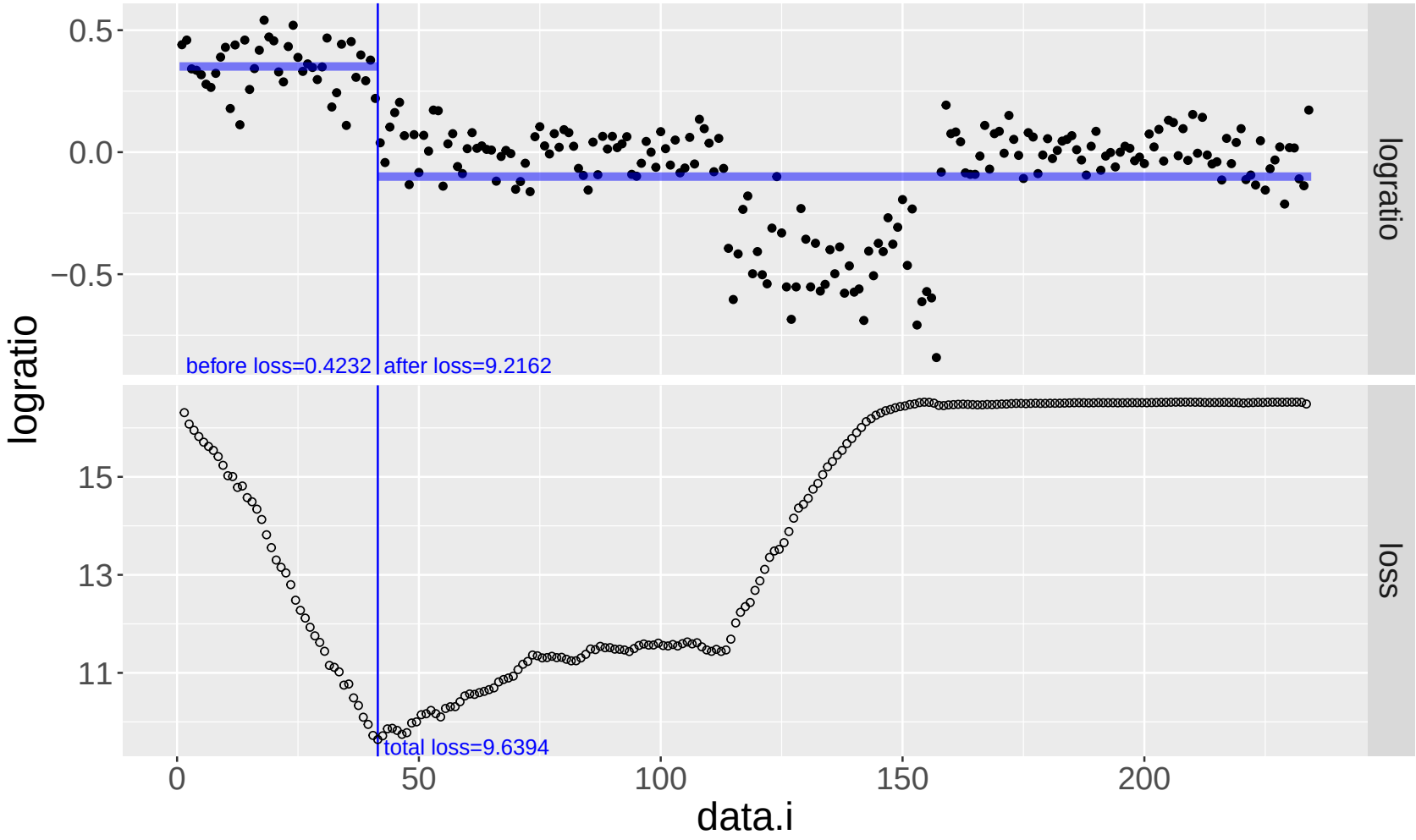
- Use cumulative sum to compute square loss from 1 to t, $L_{1,t}$, and from $t+1$ to n, $L_{t+1,n}$, and minimize over all changepoints t,

$$\min_{t \in \{1,\ldots,n-1\}} L_{1,t} + L_{t+1,n}$$

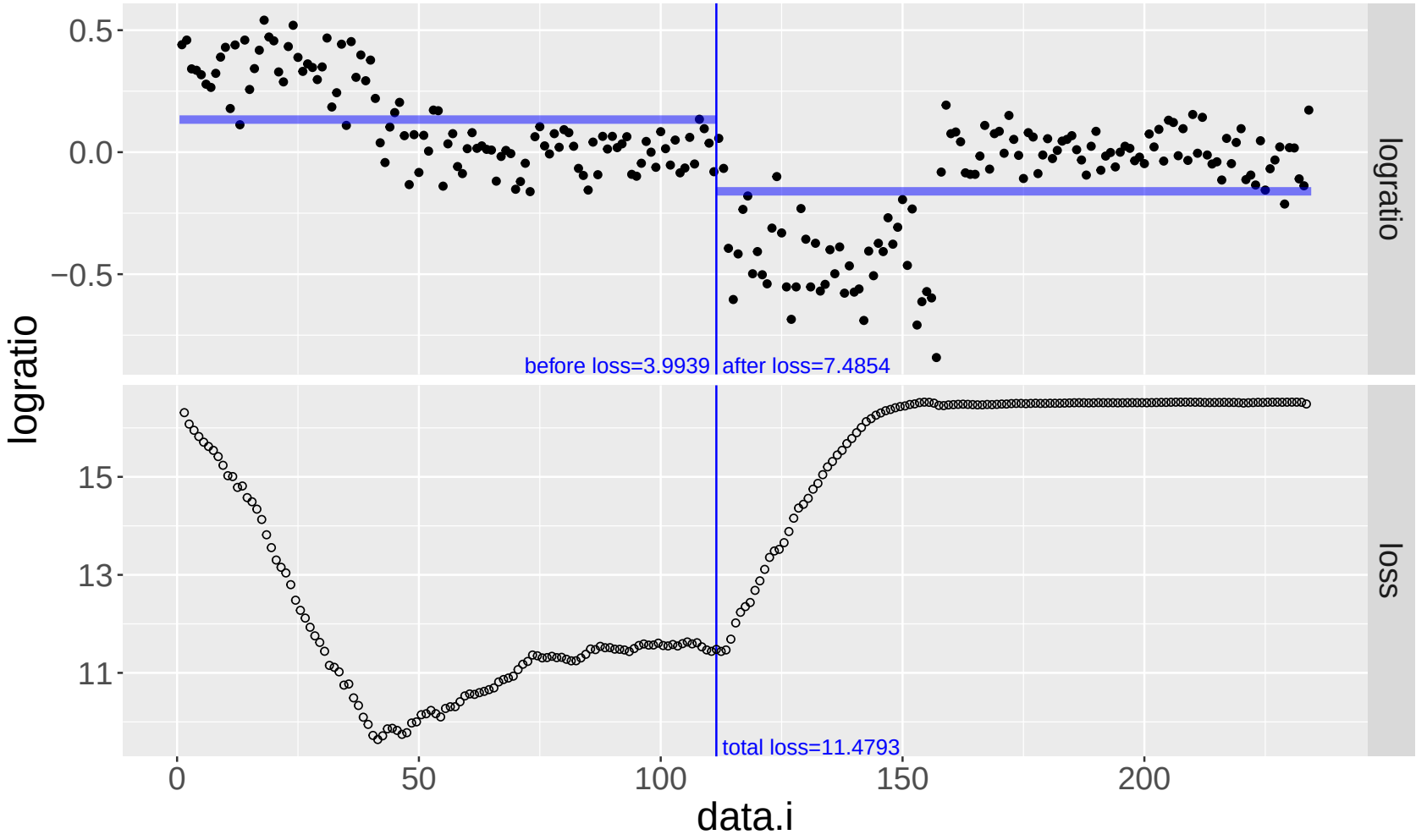
First step of binary segmentation



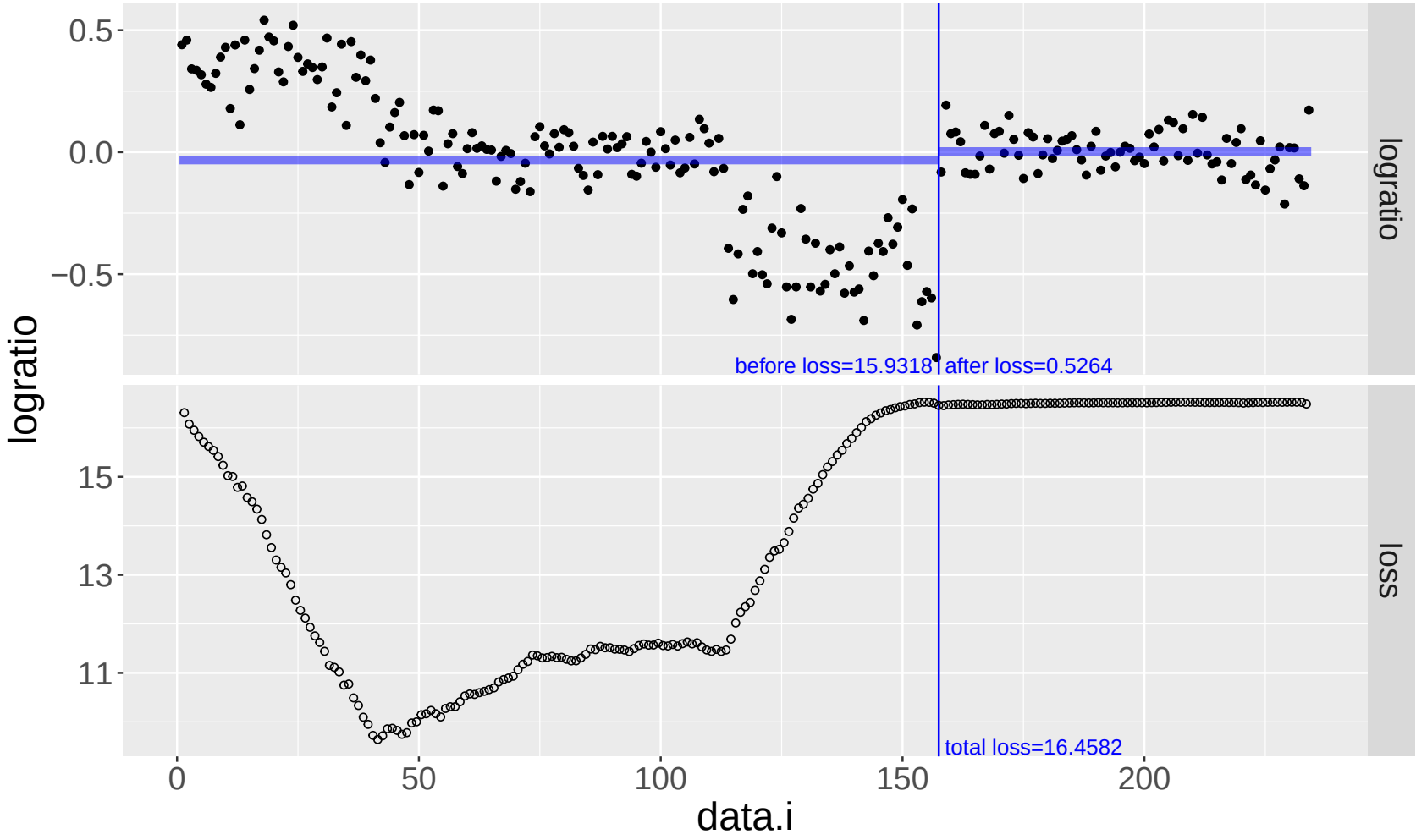
First step of binary segmentation



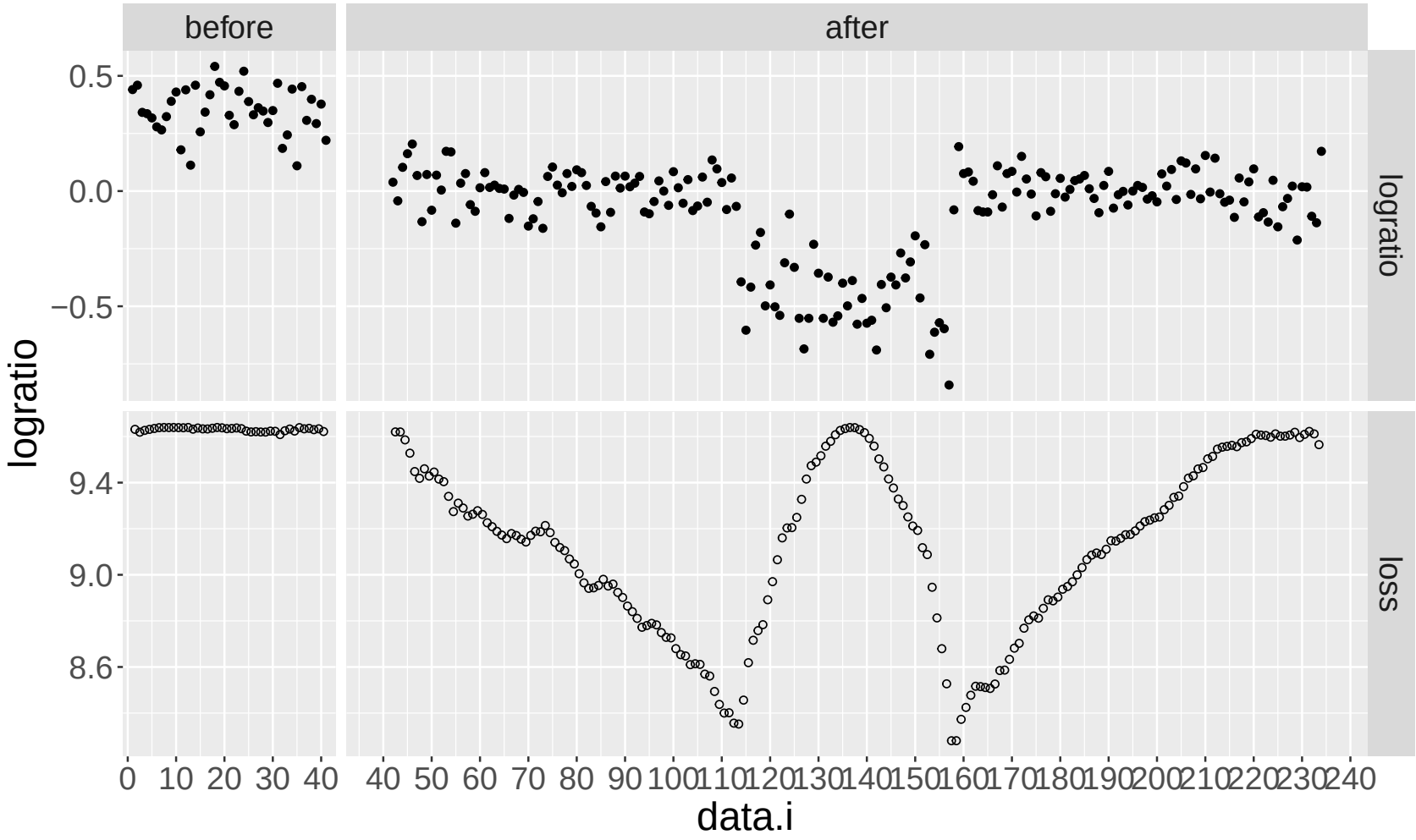
First step of binary segmentation



First step of binary segmentation

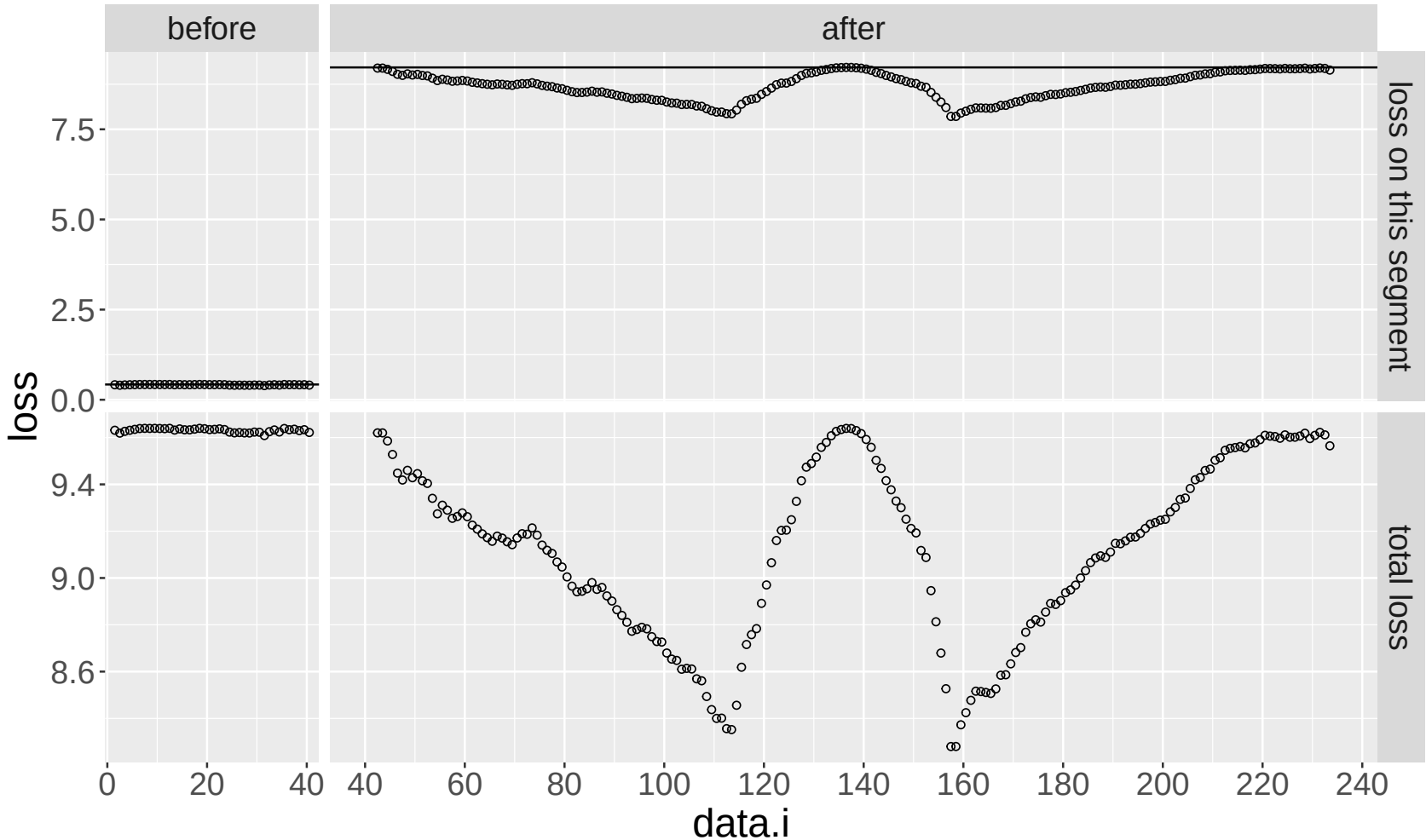


Second step of binary segmentation



Loss computation

- Minimization can be performed by choosing the split which maximizes the decrease in loss with respect to previous model (with no split).



Complexity analysis

- Assume n data and K segments.
- Computing best loss decrease and split point for a segment with t data takes $O(t)$ time.
- Keep a list of segments which could be split, sorted by loss decrease values.
- Best case is when segments get cut in half each time, $O(n \log K)$ time.
- Worst case is when splits are very unequal $(1, t-1)$, $O(nK)$ time.

Possible exam questions

- TODO