Mathematics & Statistics | Lancaster University

Introduction to Changepoint Analysis

Rebecca Killick(r.killick@lancs.ac.uk)
CASI 2024

Workshop Plan

- What are changepoints?
- Notation
- Likelihood based changepoints
  - Change in mean
  - Change in mean and/or variance
  - Change in trend and autocorrelation
- How many changes?
- Non-parametric changepoints
- Checking assumptions (if time allows)

There will be tasks throughout the sections.

What are Changepoints?

Changepoints are also known as:

- breakpoints
- segmentation
- structural breaks
- regime switching
- detecting disorder

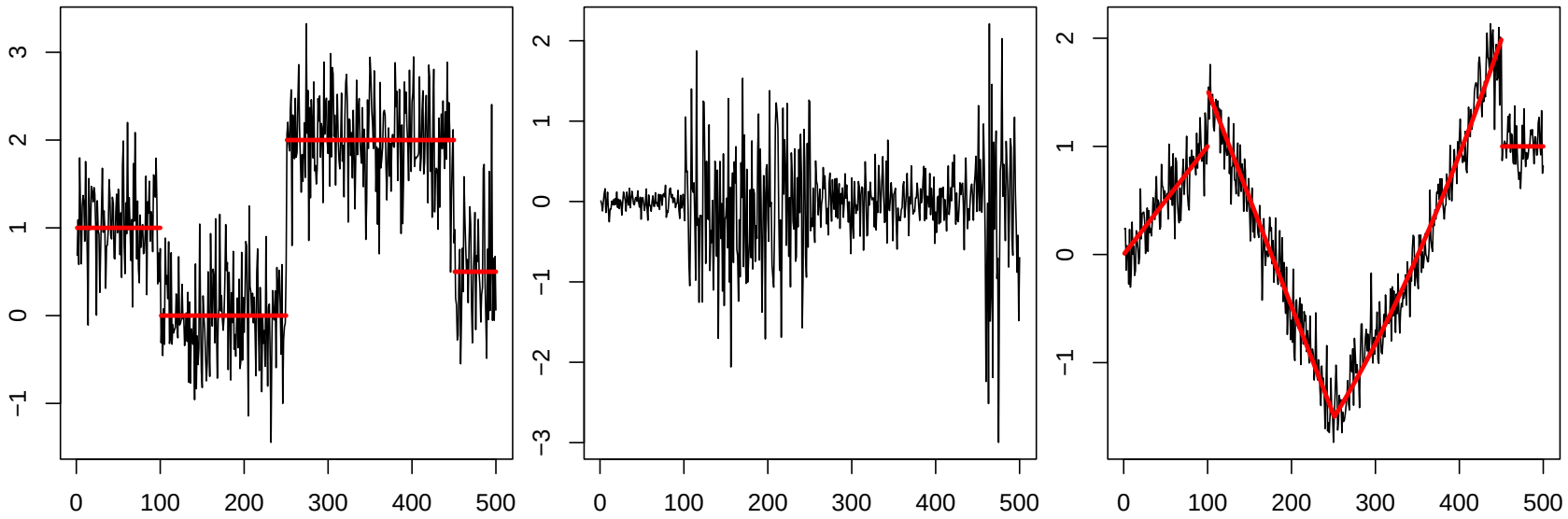
and can be found in a wide range of literature including

- quality control
- economics
- medicine
- environment
- linguistics
- ...

What are changepoints?

For data $y_1, \ldots, y_n$, if a changepoint exists at τ, then $y_1, \ldots, y_\tau$ differ from $y_{\tau+1}, \ldots, y_n$ in some way.

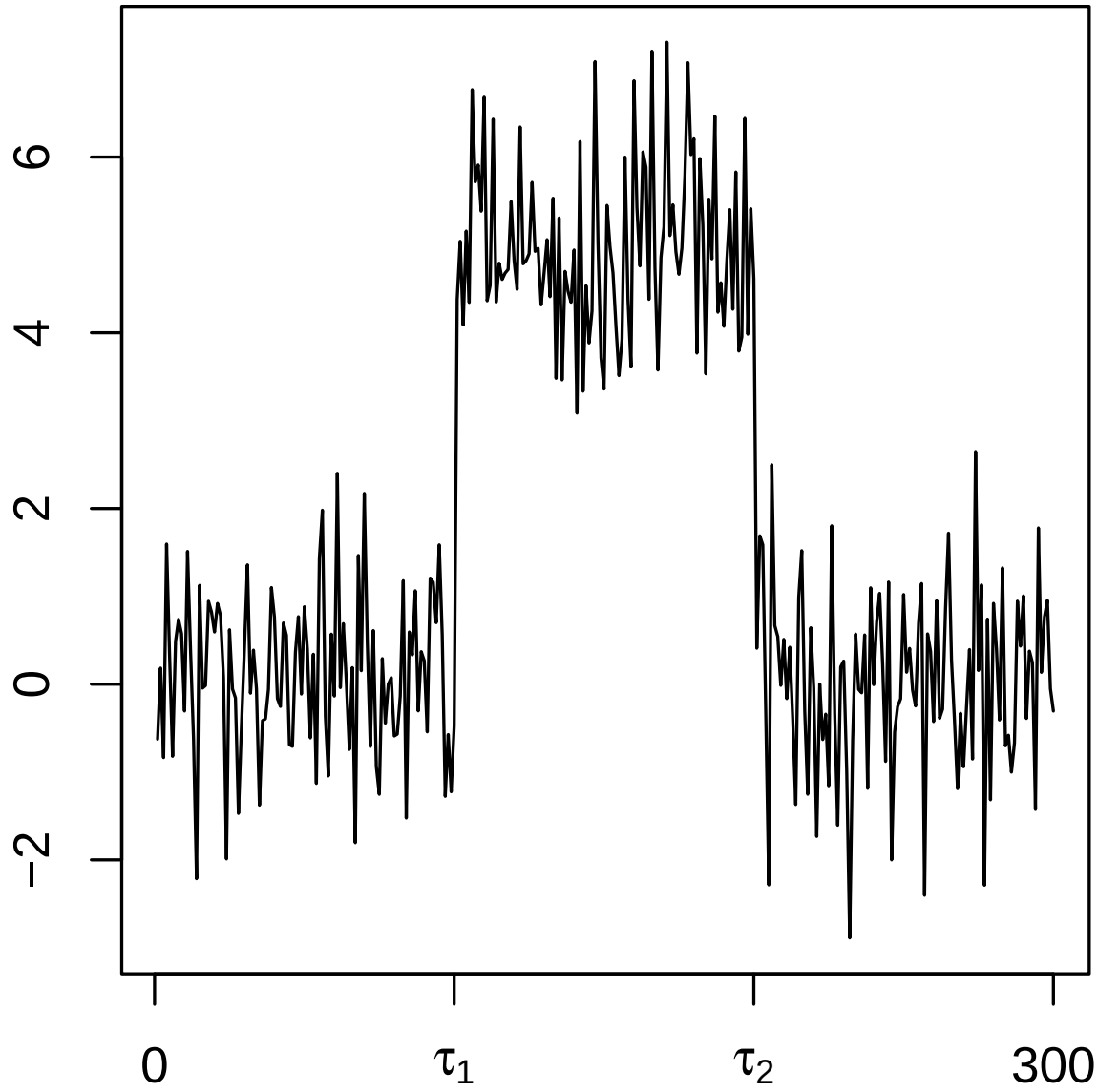
There are many different types of change.



What is the goal?

- Has a change occurred?
- If yes, where is the change?
- What is the difference between the pre and post change data?
  - Maybe this is the type of change
  - Maybe it is the parameter values before and after the change
- What is the probability that a change has occured?
- How certain are we of the changepoint location?
- How many changes have occurred (+ all the above for each change)?
- Why has there been a change?

Notation and Concepts

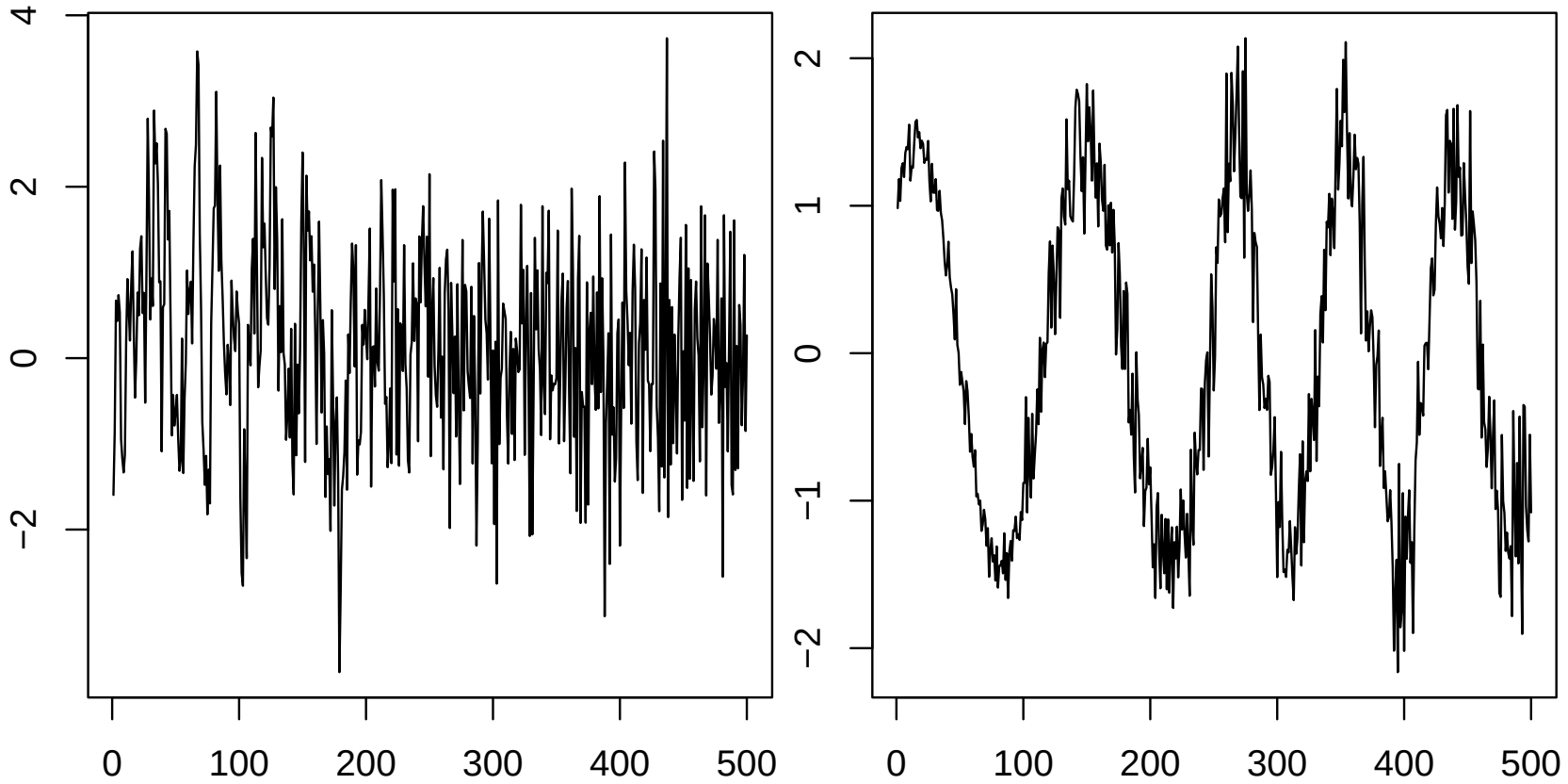


Notation and Concepts

Thus a changepoint model for a change in mean has the following formulation:

$$y_t = \begin{cases} \mu_1 & \text{if} \quad 1 \leq t \leq \tau_1 \\ \mu_2 & \text{if} \quad \tau_1 < t \leq \tau_2 \\ \vdots & \quad \vdots \\ \mu_{m+1} & \text{if} \quad \tau_m < t \leq \tau_{m+1} = n \end{cases}$$

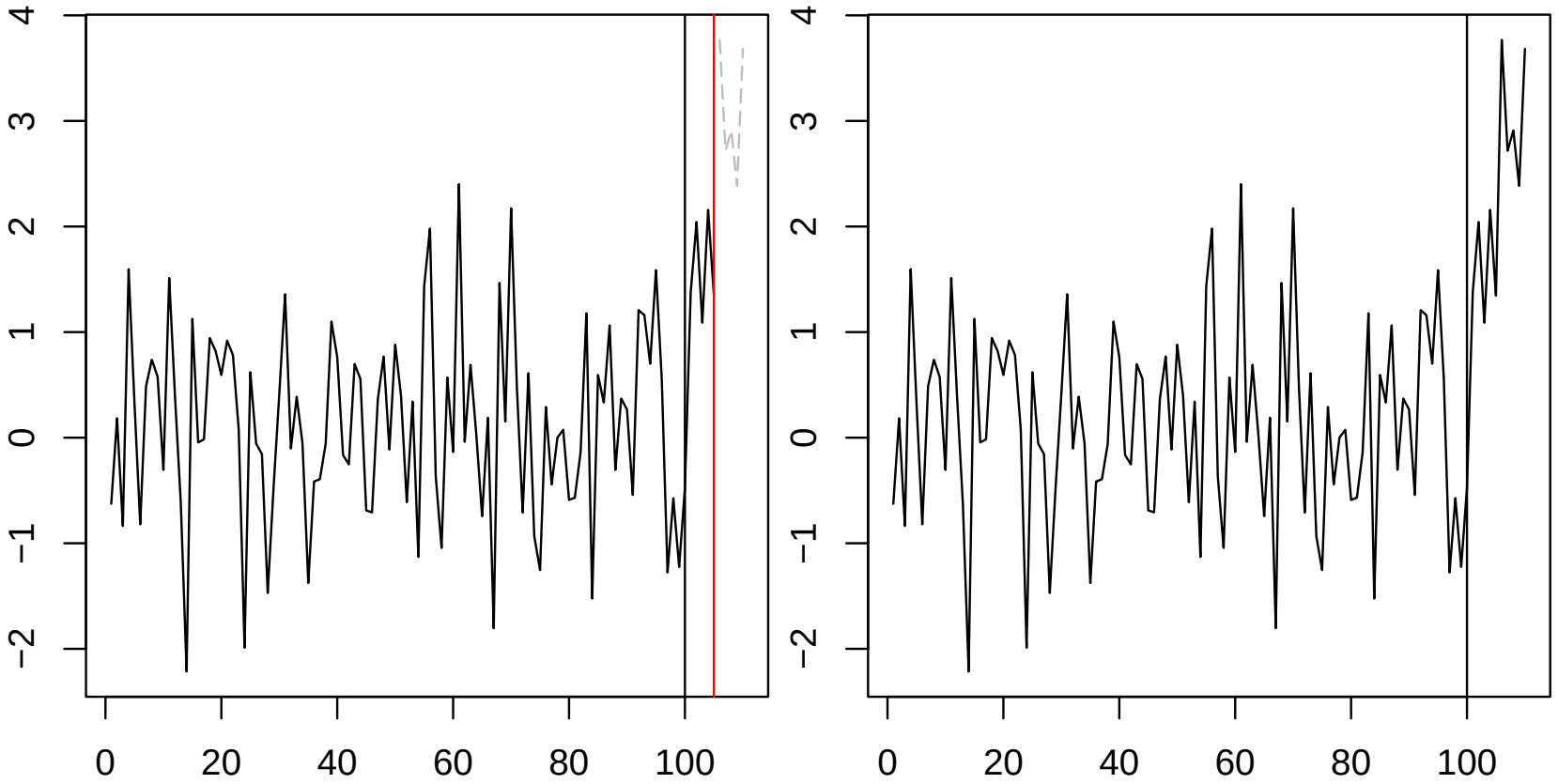
More complicated changes



Online vs Offline

- Online
  - Processes data as it arrives or in batches
  - Goal is quickest detection of a change
  - Often used in processing control, intrusion detection
- Offline
  - Processes all the data in one go
  - Goal is accurate detection of a change
  - Often used in genome analysis, audiology

Online vs Offline



Packages

Today we will use the

```
library(changepoint)
```

```
library(changepoint.np)
```

```
library(EnvCpt)
```

packages.

Other notable `R` packages are available for changepoint analysis including

- `strucchange` - for changes in regression
- `bcp` - if you want to be Bayesian
- `cpm` - for online changes (`changepoint.online` coming soon)
- `fastcpd` - for long series with few or clustered changepoints

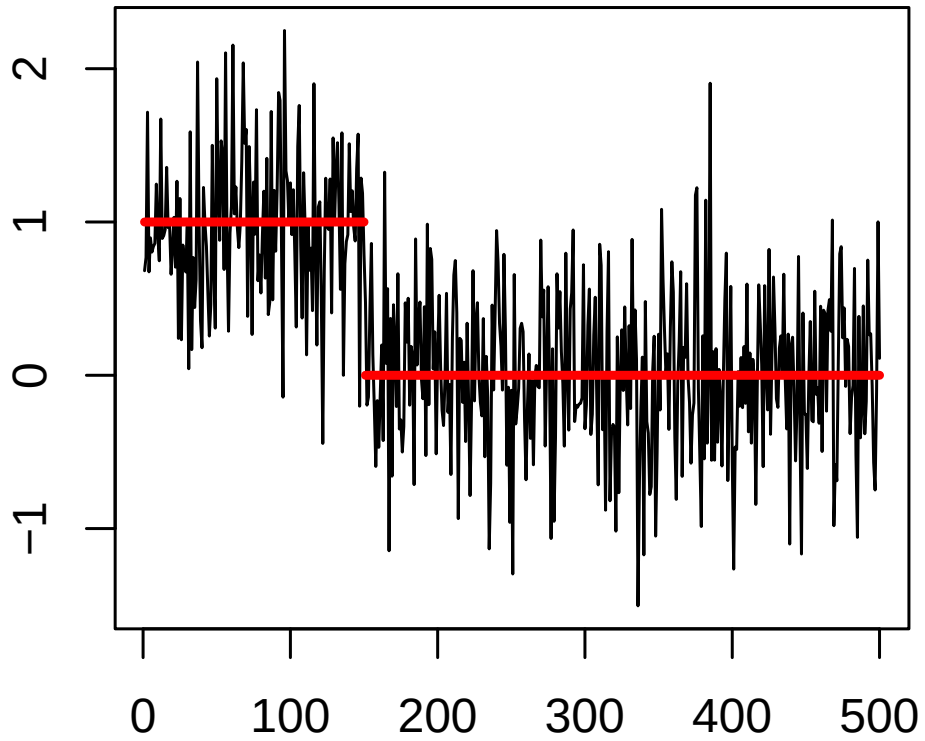
Single Changepoint

Mathematics & Statistics | Lancaster University

Assume we have time-series data where

$$y_t|\theta_t \sim \mathrm{N}(\theta_t, 1),$$

but where the means, θ_t, are piecewise constant through time with a single change.

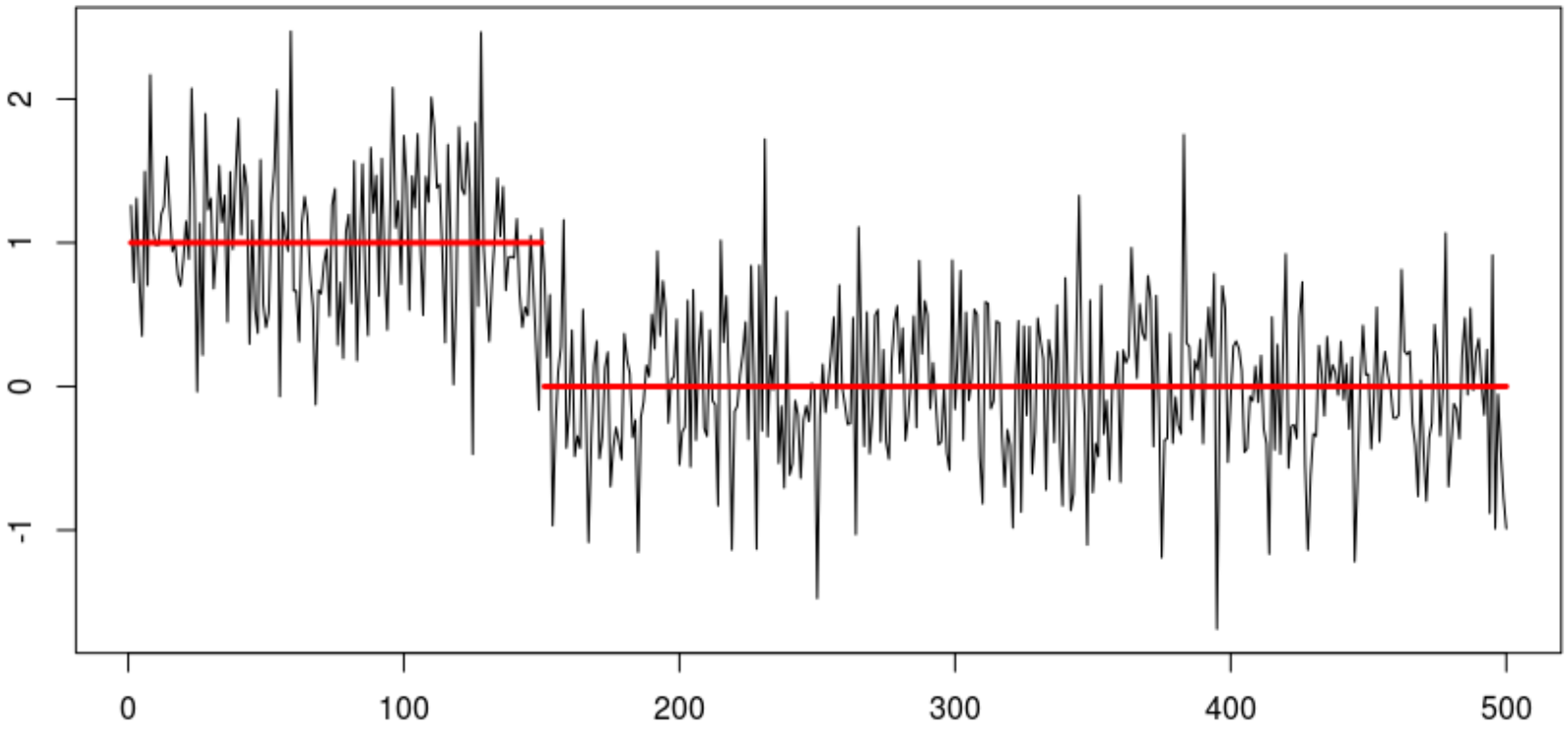


Single Changepoint

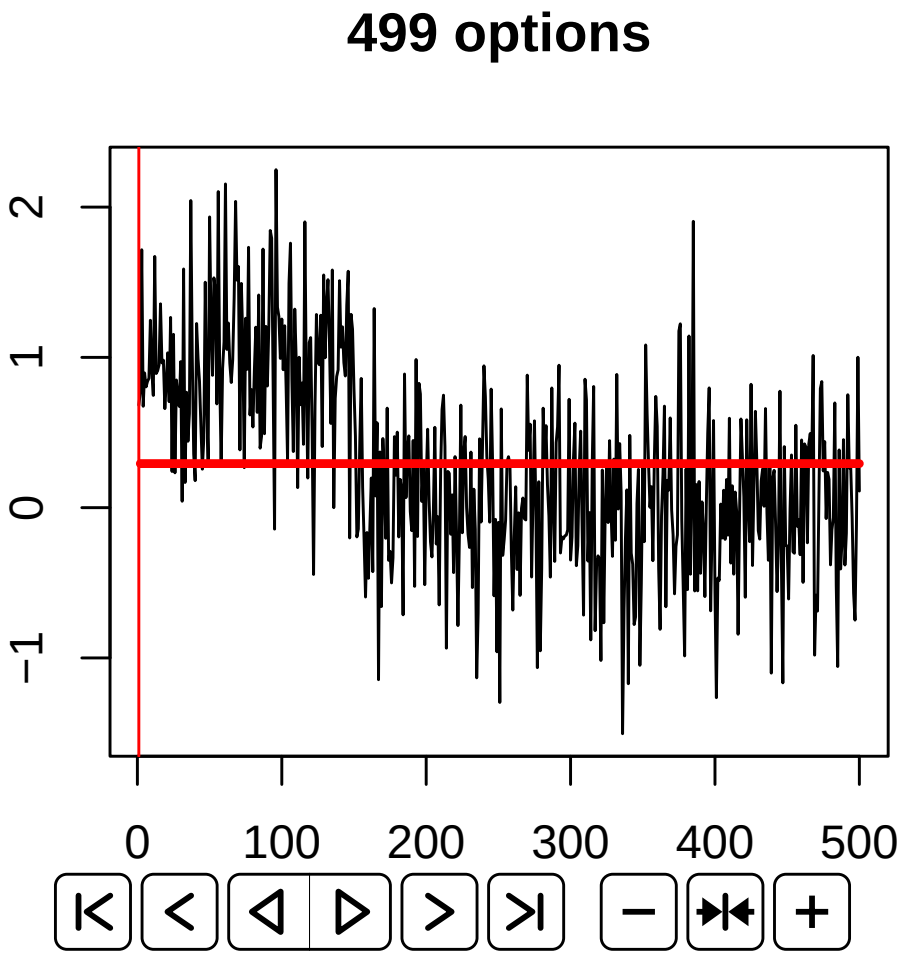
Fit for whole data

=

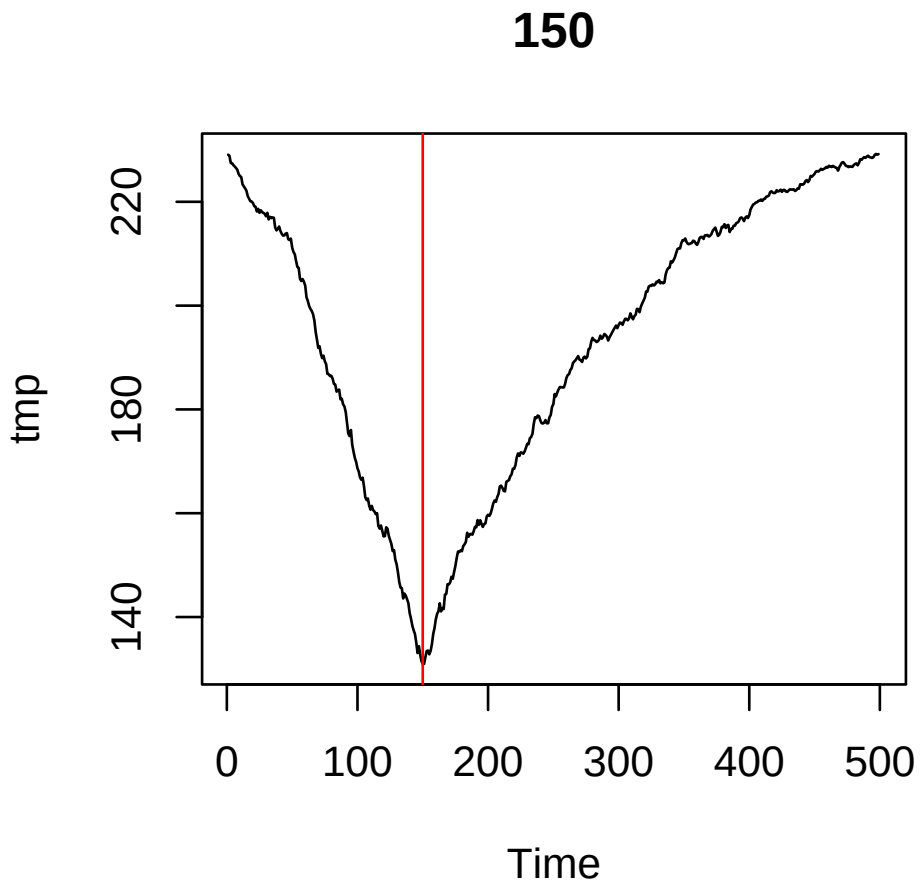
Cost of segment + Cost of segment



Finding a single change



Finding a single change



Finding a single change

How do we know if the changepoint found is “significant” or not?

- We also calculate the cost of the whole data with no change
- If the difference is **large enough** then we say there is a change

In practice **large enough** is hard to define as it is application dependent.

The default in the `changepoint` package is MBIC - a Modified Bayesian Information Criterion.

This will not be appropriate for all data sets! More later . . .

changepoint R package

The changepoint R package contains 3 wrapper functions:

- cpt.mean - mean only changes
- cpt.var - variance only changes
- cpt.meanvar - mean and variance changes

The package also contains:

- functions/methods for the cpt S4 class
- 5 data sets
- other R functions that are made available for those who know what they are doing and might want to extend/modify the package.

The `cpt` class

- S4 class
- Slots store all the information from the analysis
  - e.g. `data.set`, `cpts`, `param.est`, `pen.value`, `ncpts.max`
- Slots are accessed via their names e.g. `cpts(x)`
- Standard methods are available for the class e.g. `plot`, `summary`
- Additional generic functions are available e.g. `seg.len`, `ncpts`
- Each core function outputs a `cpt` object

cpt.mean


```
cpt.mean(data, penalty="MBIC", pen.value=0,
method="AMOC", Q=5, test.stat="Normal", class=TRUE,
param.estimates=TRUE,minseglen=1)
```

- `data` - vector or `ts` object
- `penalty` - cut-off point, MBIC, SIC, BIC, AIC, Hannan-Quinn, Asymptotic, Manual.
- `pen.value` - Type I error for Asymptotic, number or character for manual.
- `method` - AMOC, PELT, SegNeigh, BinSeg.
- `Q` - max number of changes for SegNeigh or BinSeg.
- `test.stat` - Test statistic, Normal or CUSUM.
- `class` - return a `cpt` object or not.
- `param.estimates` - return parameter estimates or not.
- `minseglen` - minimum number of data points between changes.

Single Change in Mean

IMPORTANT: The `cpt.mean` function assumes that the variance of a time series is 1. If this is not the case then you need to scale the data prior to analysis.

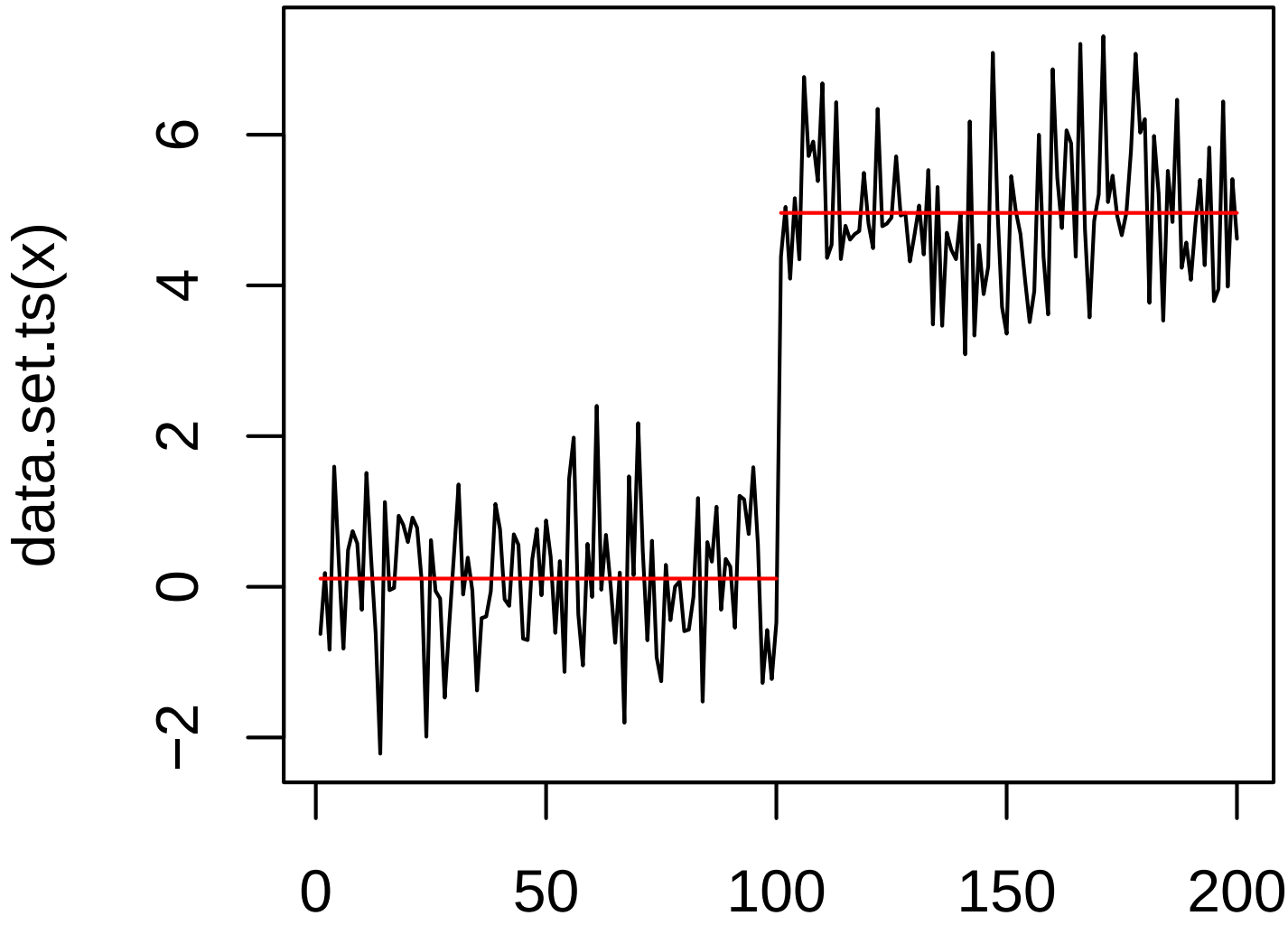
```
set.seed(1)
m1=c(rnorm(100,0,1),rnorm(100,5,1))
m1.amoc=cpt.mean(m1)
cpts(m1.amoc)
```

```
## [1] 100
```

Single Change in Mean

Mathematics & Statistics | Lancaster University

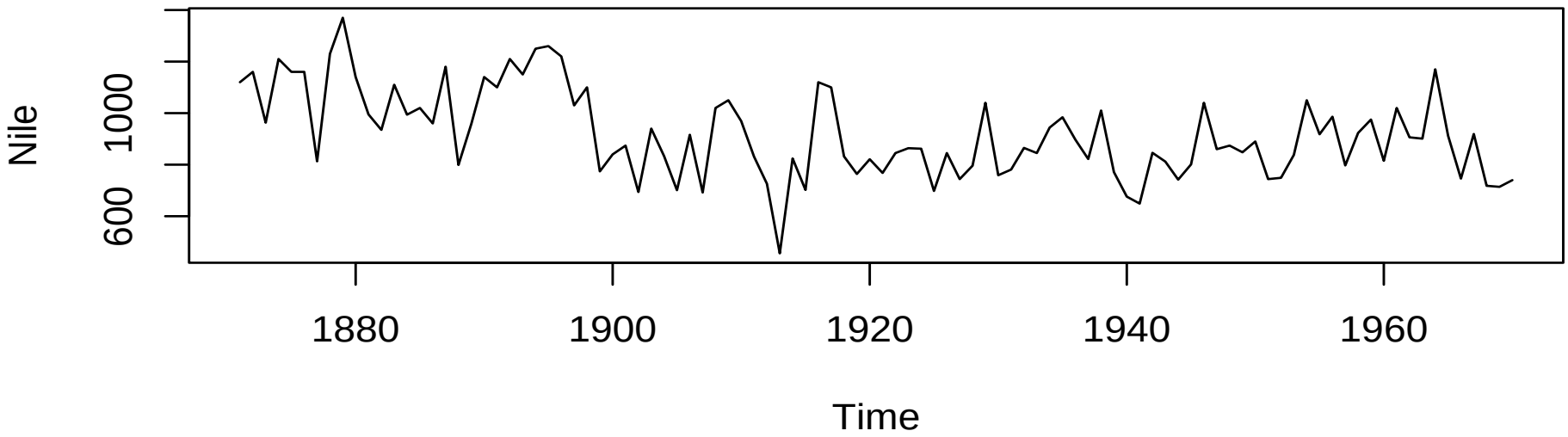
```
plot(m1.amoc)
```



Task: Nile

Data from Cobb (1978): readings of the annual flow volume of the Nile River at Aswan from 1871 to 1970.

```
data(Nile)
ts.plot(Nile)
```



Hypothesized that there was a change around the turn of the century.

Task: Nile

Use the `cpt.mean` function to see if there is evidence for a change in mean in the Nile river data.

```
data(Nile)
```

If you identify a change, where is it and what are the pre and post change means?

Don't forget to

```
library(changepoint)
```

before you start

Task: Nile

Annual flow volume of the Nile River at Aswan from 1871 to 1970 studied in Cobb(1978).

```
nile.default=cpt.mean(Nile,method="AMOC")
cpts(nile.default)
```

```
## [1] 28
```

```
cpts.ts(nile.default)
```

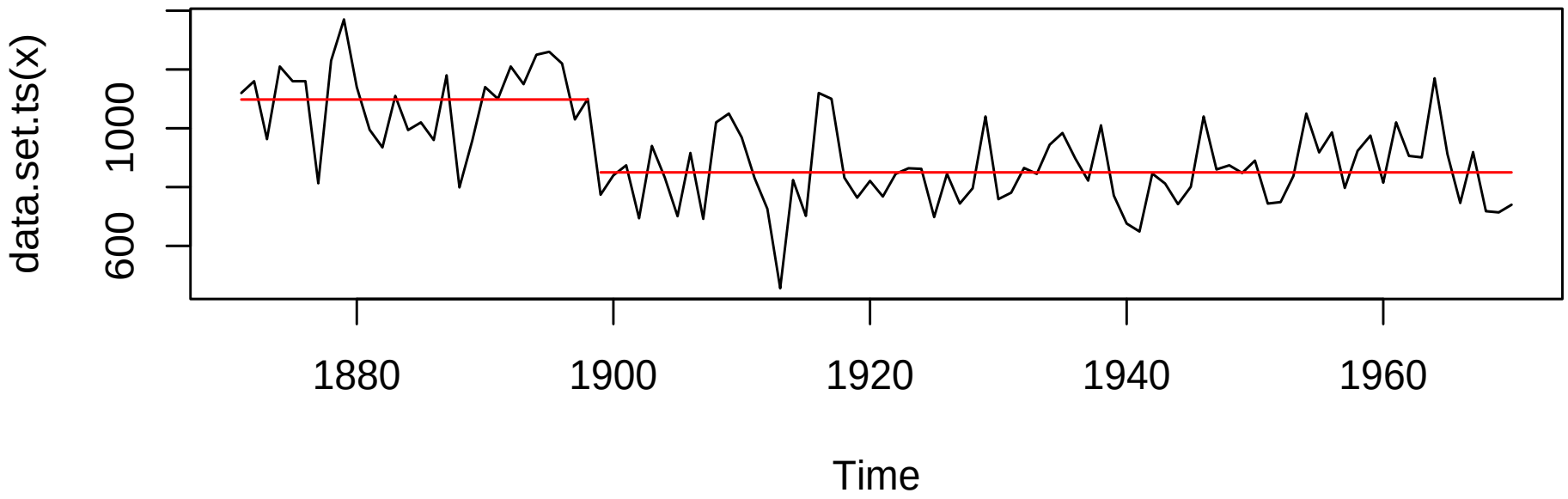
```
## [1] 1898
```

```
param.est(nile.default)
```

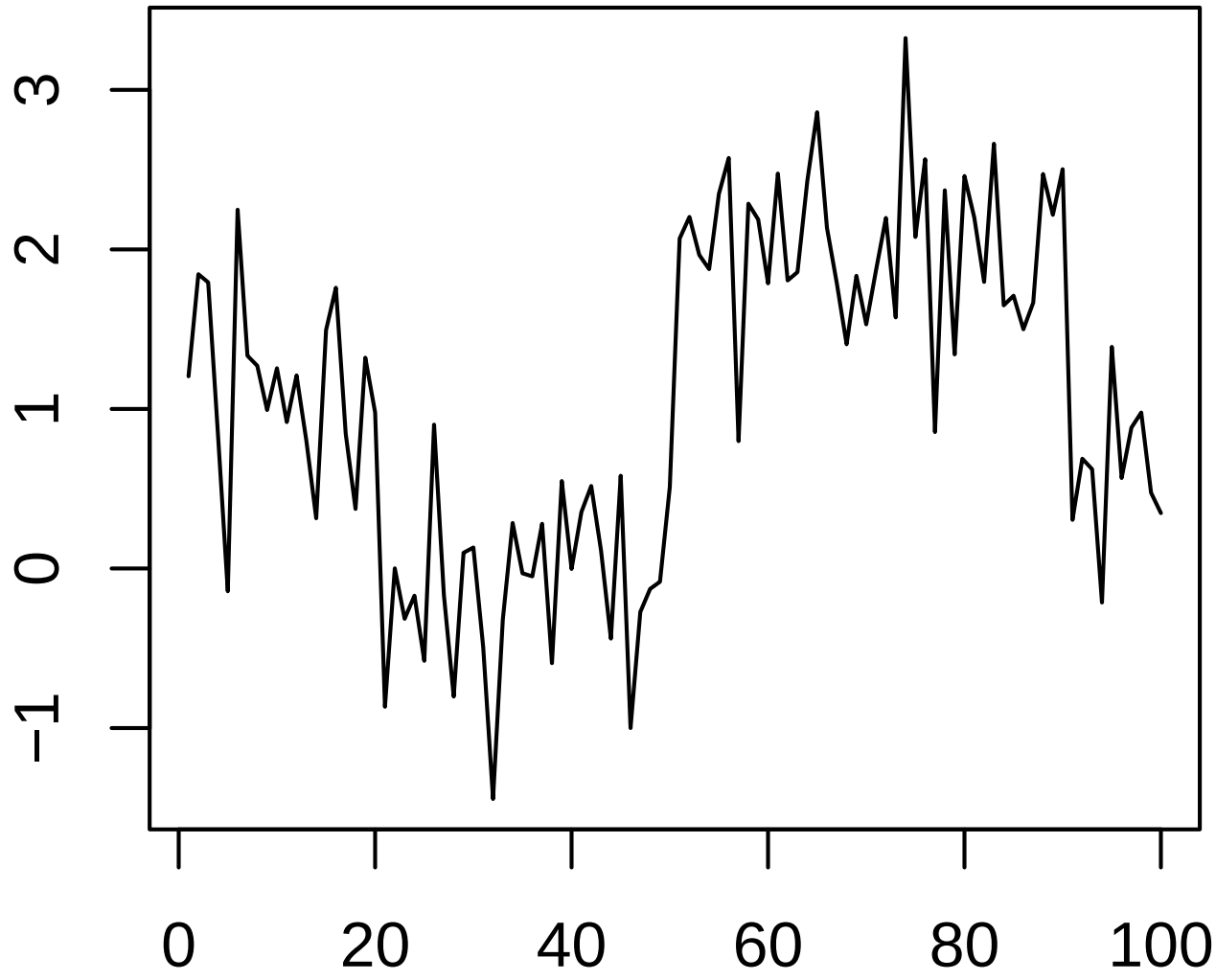
```
## $mean
## [1] 1097.7500  849.9722
```

Task: Nile

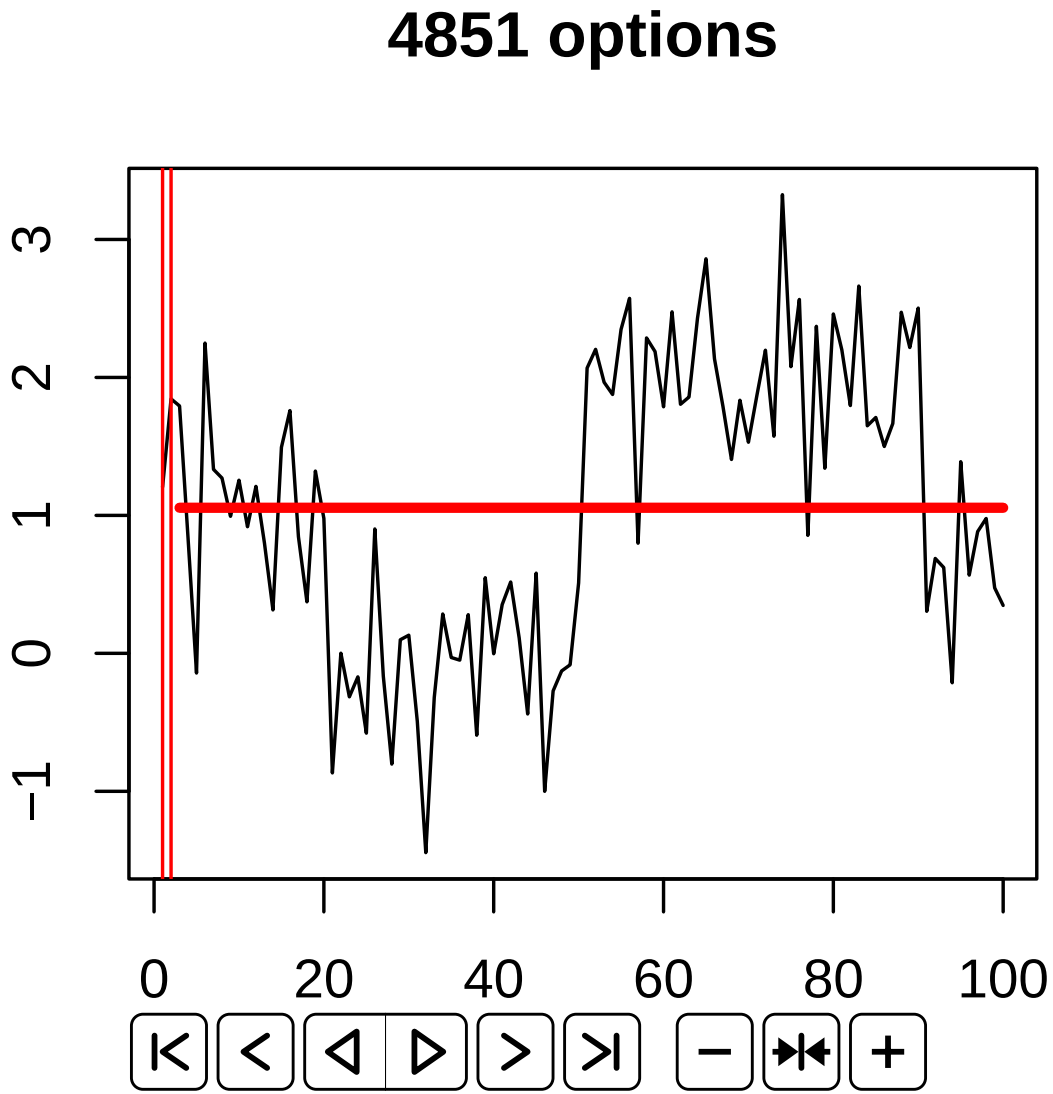

```
plot(nile.default)
```



Multiple Changepoints



Multiple Changepoints



The Challenge

- What are the values of $\tau_1, \ldots, \tau_m$?
- What is m?
- For n data points there are 2^{n-1} possible solutions
- If m is known there are still $\binom{n-1}{m}$ solutions
- If $n = 1000$ and $m = 10$, 2.607755×10^{23} solutions
- How do we search the solution space efficiently?

Methods in `changepoint`

- At Most One Change (`AMOC`)

Approximate but computationally **fast**:

- Binary Segmentation (`BinSeg`) (Scott and Knott (1974)) which is $\mathcal{O}(n \log n)$ in CPU time.

Slower but **exact**:

- Segment Neighbourhood (`SegNeigh`) (Auger and Lawrence (1989)) is $\mathcal{O}(Qn^2)$.

Fast and **exact**:

- Pruned Exact Linear Time (`PELT`) (Killick et al. (2012)) At worst $\mathcal{O}(n^2)$. For linear penalties, scaling changes, $\mathcal{O}(n)$.

cpt.var


```
cpt.var(data, penalty, pen.value, know.mean=FALSE, mu=NA,
method, Q, test.stat="Normal", class, param.estimates,
minseglen=2)
```

Majority of arguments are the same as for `cpt.mean`

- `know.mean` - if known we don't count it as an estimated parameter when calculating penalties.
- `mu` - Mean if known.
- `test.stat` - Normal or CSS (cumulative sums of squares)
- `minseglen` - Default is 2

Changes in Variance

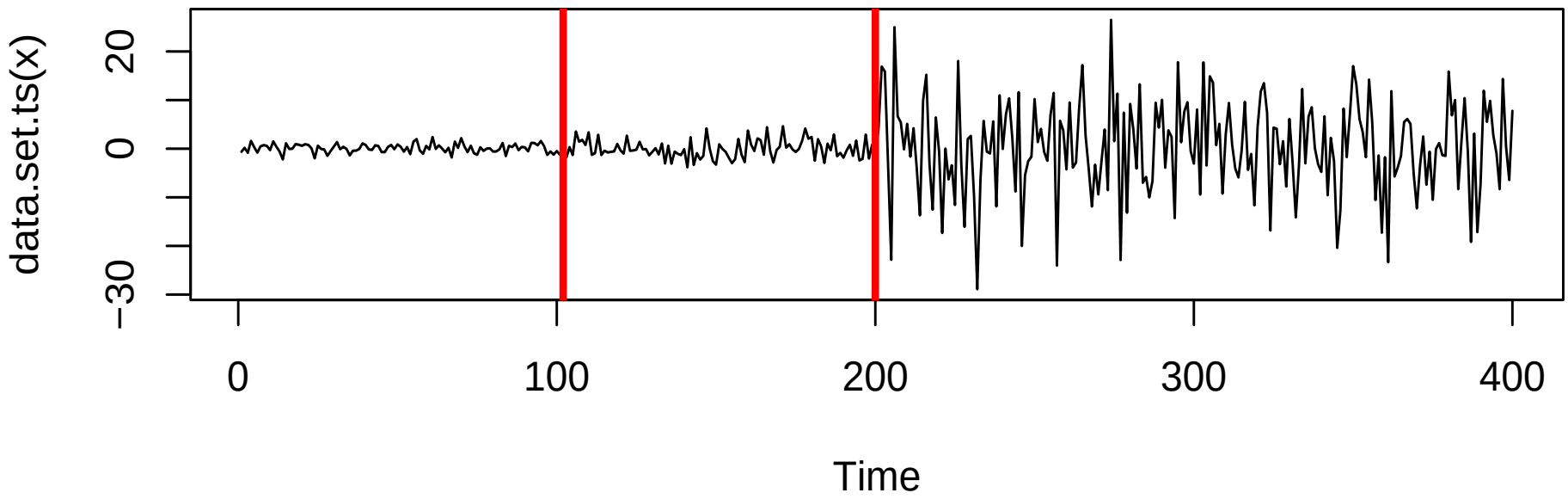

```
set.seed(1)
v1=c(rnorm(100,0,1),rnorm(100,0,2),rnorm(100,0,10),
     rnorm(100,0,9))
v1.man=cpt.var(v1,method='PELT',penalty='Manual',
     pen.value='2*log(n)')
cpts(v1.man)
param.est(v1.man)
```

```
## [1] 102 200
## $variance
## [1]  0.8007158  3.6933616 92.3876410
##
## $mean
## [1] 0.1986058
```

Changes in Variance

Ratios of true variances (4, 25, 0.81)

```
plot(v1.man,cpt.width=3)
```



cpt.meanvar


```
cpt.meanvar(data, penalty, pen.value, method, Q,
test.stat="Normal", class, param.estimates,
shape=1,minseglen=2)
```

Again the same underlying structure as `cpt.mean`.

- `test.stat` - choice of Normal, Gamma, Exponential, Poisson.
- `shape` - assumed shape parameter for Gamma.
- `minseglen` - minimum segment length of 2

Mean & Variance


```
set.seed(1)
mv1=c(rexp(50,rate=1),rexp(50,5),rexp(50,2),rexp(50,7))
mv1.pelt=cpt.meanvar(mv1,test.stat='Exponential',
      method='BinSeg',Q=10,penalty="SIC")
cpts(mv1.pelt)
```

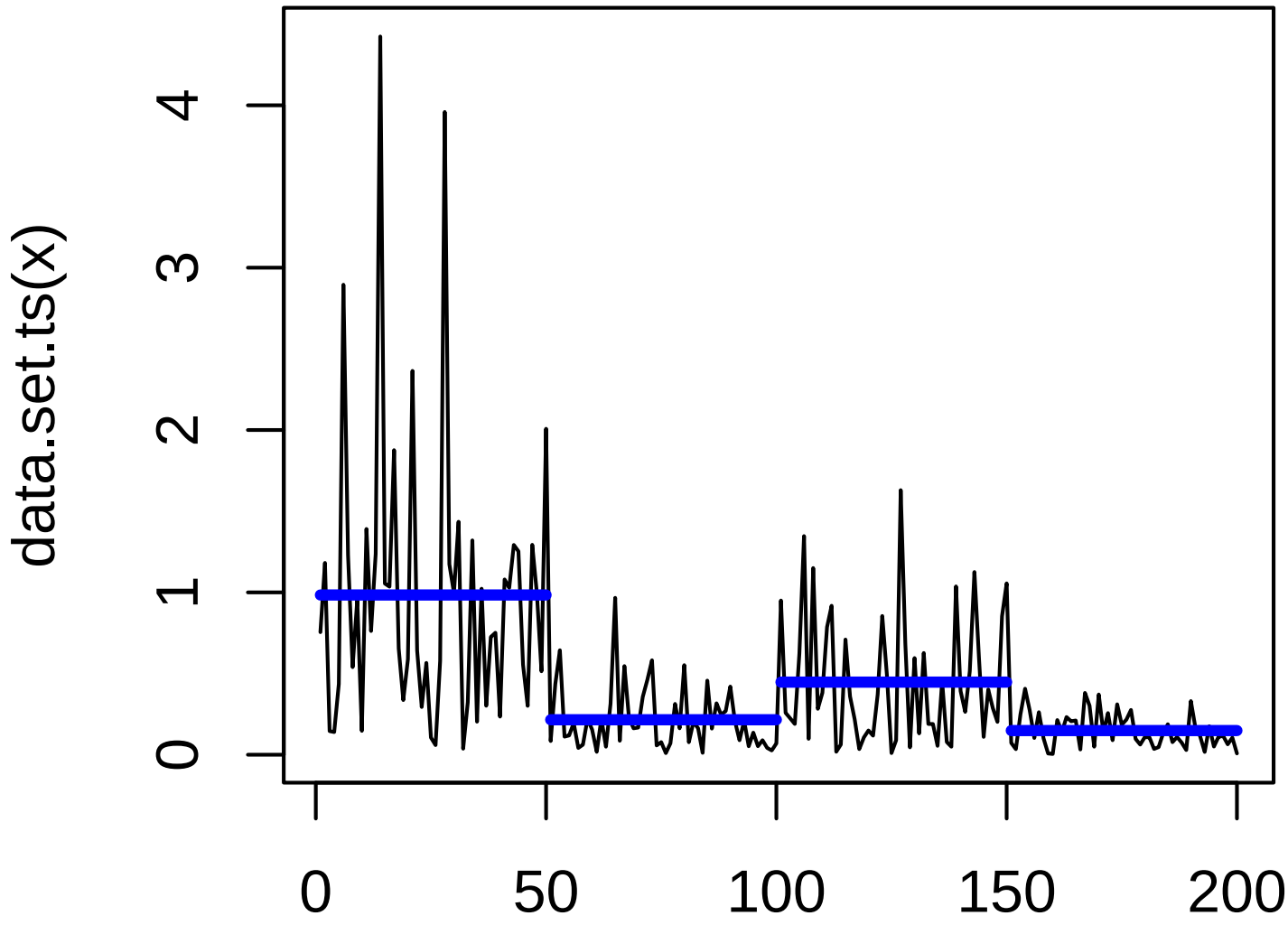
```
## [1]  50 100 150
```

```
param.est(mv1.pelt)
```

```
## $rate
## [1] 1.016217 4.641184 2.235431 6.705612
```

Mean & Variance


```
plot(mv1.pelt,cpt.width=3,cpt.col='blue')
```

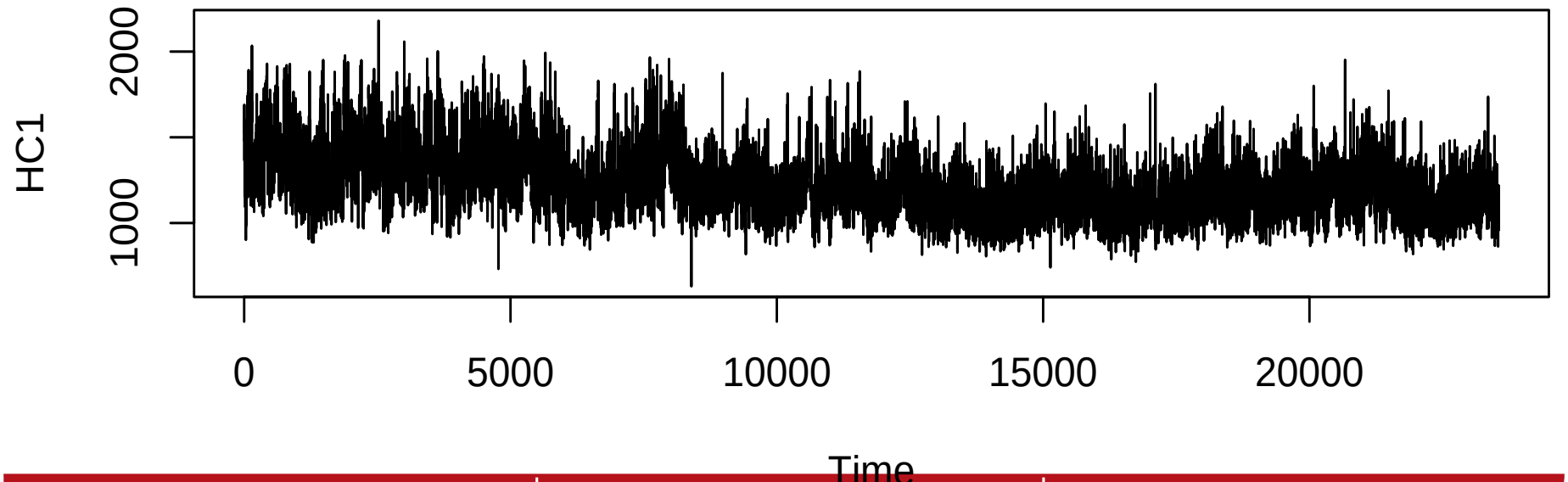


Task HC1

G+C content within part of Human Chromosome 1, data from NCBI. 3kb windows along the Human Chromosome from 10Mb to 33Mb.

Use the `cpt.meanvar` function to identify regions with different C+G content.

```
data(HC1)
ts.plot(HC1)
```



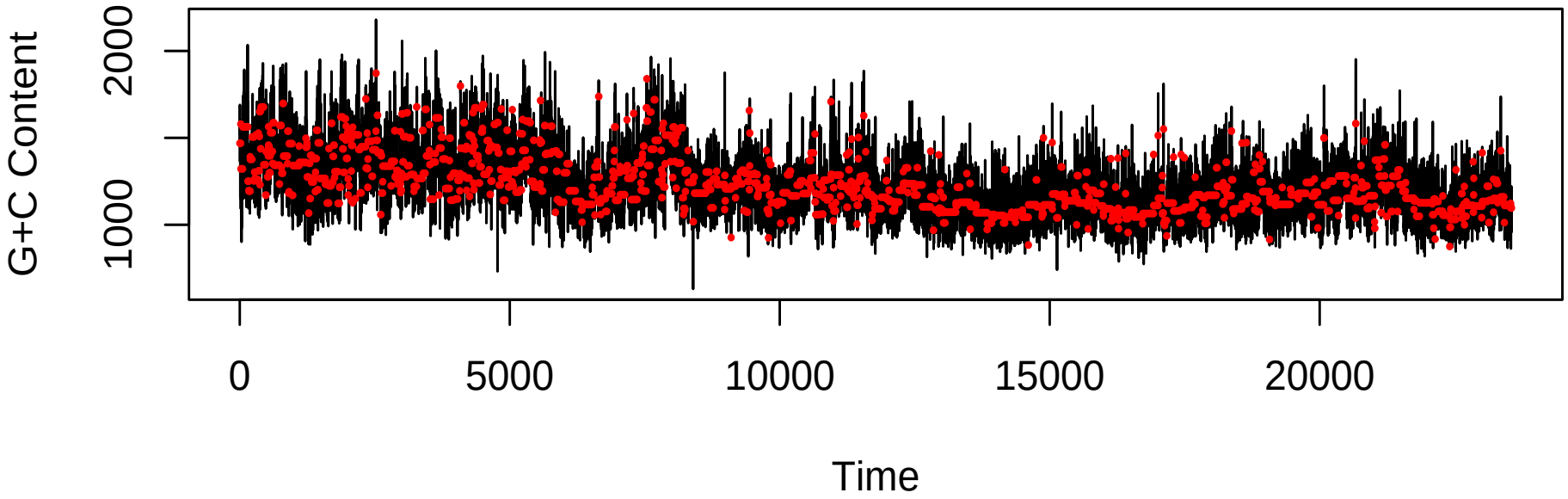
A solution: HC1


```
data(HC1)
hc1.pelt=cpt.meanvar(HC1,method='PELT',penalty='Manual',
         pen.value=14)
ncpts(hc1.pelt)
```

```
## [1] 805
```

A solution: HC1


```
plot(hc1.pelt,ylab='G+C Content',cpt.width=3)
```

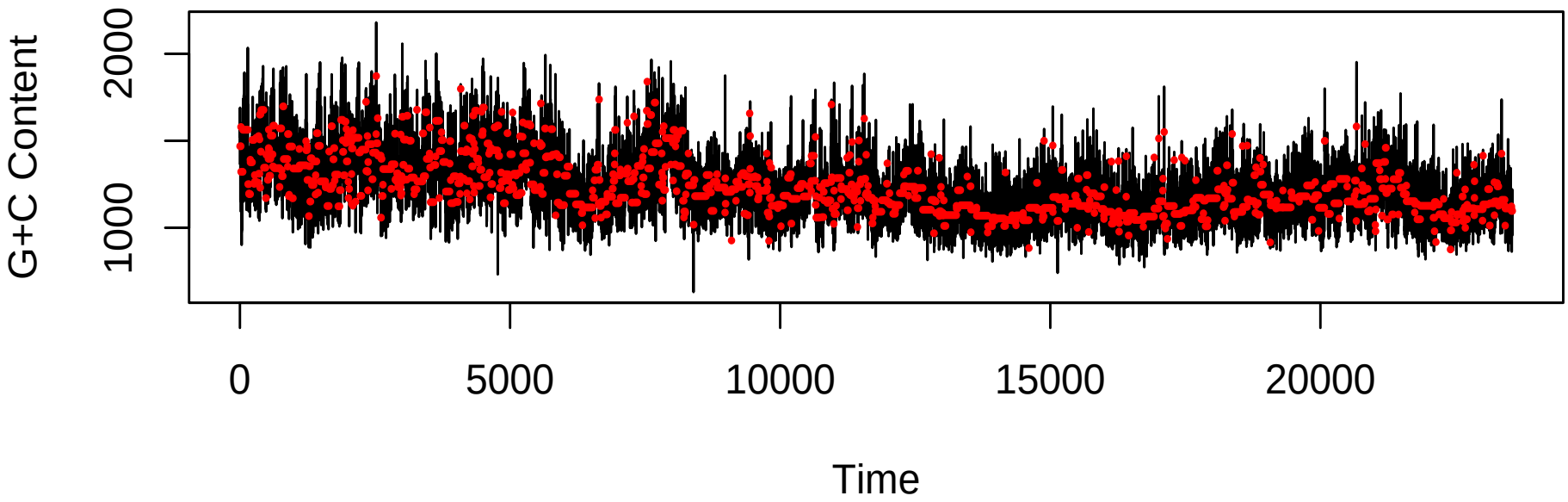


Number of changes?

Mathematics & Statistics | Lancaster University

Does the number of changes appear reasonable?

```
plot(hc1.pelt,ylab='G+C Content',cpt.width=3)
```



CROPS

Changepoints for a **r**ange **o**f **p**enalti**es**

Use `penalty='CROPS'` with `method='PELT'` to get all segmentations for a range of penalty values.

```
v1.crops=cpt.var(v1,method="PELT",penalty="CROPS",
                 pen.value=c(5,500))
```

```
## [1] "Maximum number of runs of algorithm = 10"
## [1] "Completed runs = 2"
## [1] "Completed runs = 3"
## [1] "Completed runs = 4"
## [1] "Completed runs = 5"
## [1] "Completed runs = 6"
## [1] "Completed runs = 8"
## [1] "Completed runs = 9"
```

CROPS


```
cpts.full(v1.crops)
```

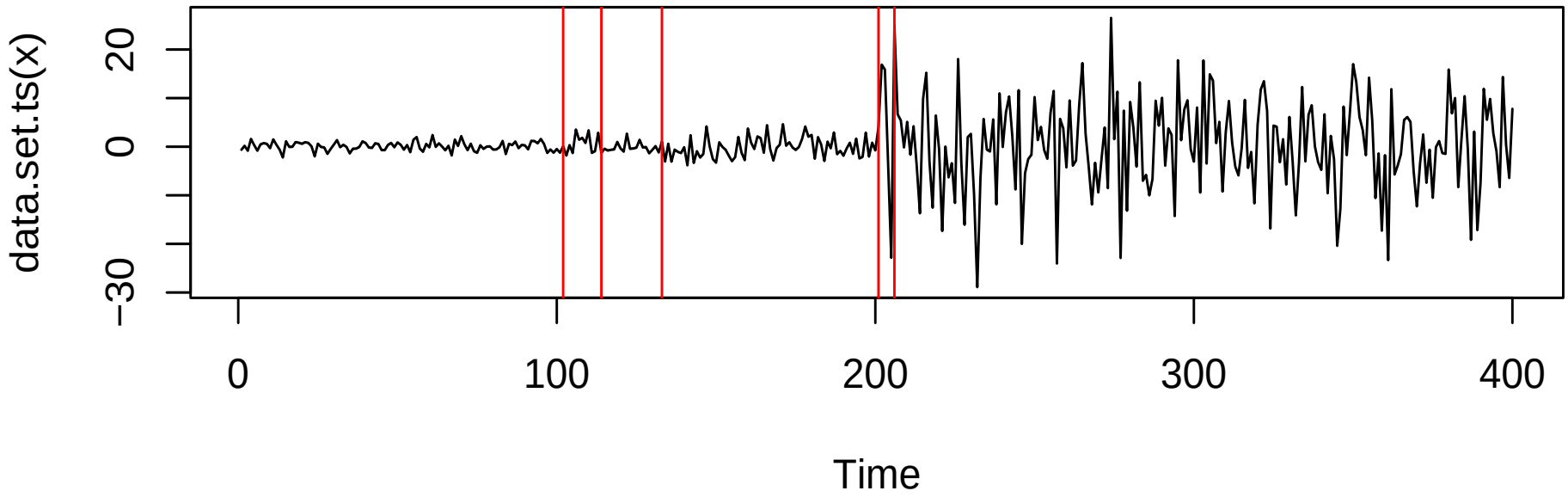
```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,]  102  114  133  201  206  213  375  379
## [2,]  102  114  133  201  206  375  379   NA
## [3,]  102  114  133  201  206   NA   NA   NA
## [4,]   96  133  201  206   NA   NA   NA   NA
## [5,]  102  201  206   NA   NA   NA   NA   NA
## [6,]  102  200   NA   NA   NA   NA   NA   NA
## [7,]  200   NA   NA   NA   NA   NA   NA   NA
## [8,]   NA   NA   NA   NA   NA   NA   NA   NA
```

CROPS


```
pen.value.full(v1.crops)
```

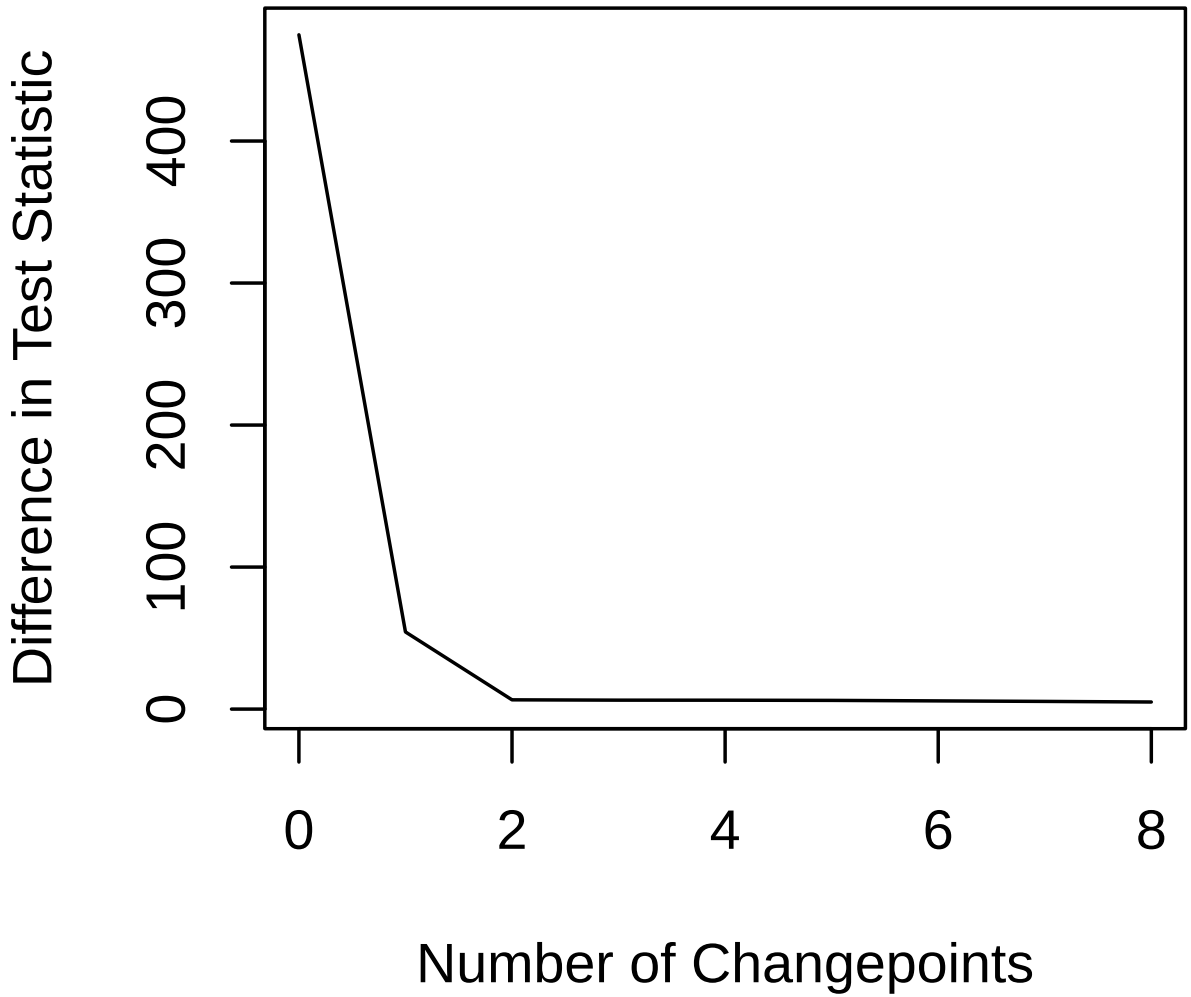
```
## [1]   5.000000   5.431360   6.151053   6.270164   6.314013
## [8] 474.797364
```

```
plot(v1.crops,ncpts=5)
```



CROPS


```
plot(v1.crops,diagnostic=TRUE)
```



cpt.np


```
cpt.np(data, penalty, pen.value, method,
test.stat="empirical_distribution", class, minseglen=1,
nquantiles=10)
```

Again the same underlying structure as `cpt.mean`.

- `test.stat` - choice of empirical_distribution
- `minseglen` - minimum segment length of 1
- `nquantiles` - number of quantiles to use

Example


```
set.seed(12)
J <- function(x){(1+sign(x))/2}
n <- 1000
tau <- c(0.1,0.13,0.15,0.23,0.25,0.4,0.44,0.65,0.76,0.78,
      0.81)*n
h <- c(2.01, -2.51, 1.51, -2.01, 2.51, -2.11, 1.05, 2.16,
      -1.56, 2.56, -2.11)
sigma <- 0.5
t <- seq(0,1,length.out = n)
data <- array()
for (i in 1:n){
   data[i] <- sum(h*J(n*t[i] - tau)) + (sigma * rnorm(1))
}
```

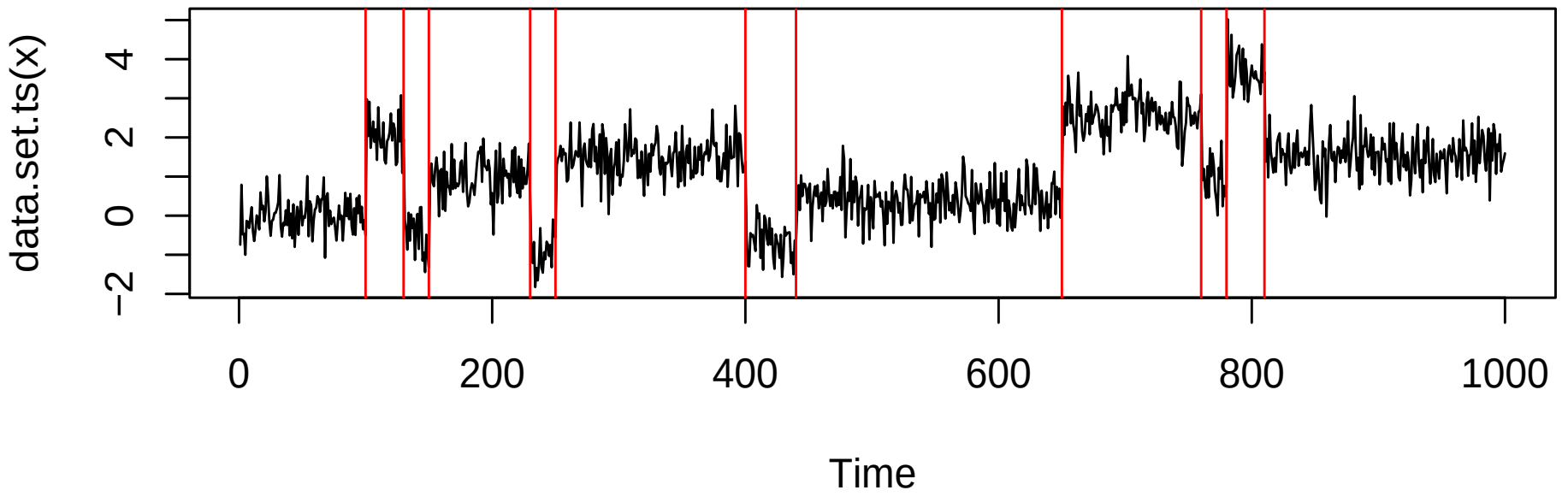
Example


```
out <- cpt.np(data, method="PELT",minseglen=2,
        nquantiles =4*log(length(data)))
cpts(out)
```

```
##  [1] 100 130 150 230 250 400 440 650 760 780 810
```

Example


```
plot(out)
```



Task: `cpt.np`

Look at the `HeartRate` data from the `changepoint.np` package. Use one of the non-parametric functions to see if there is evidence for changes in heart rate.

```
data(HeartRate)
```

A solution: HeartRate

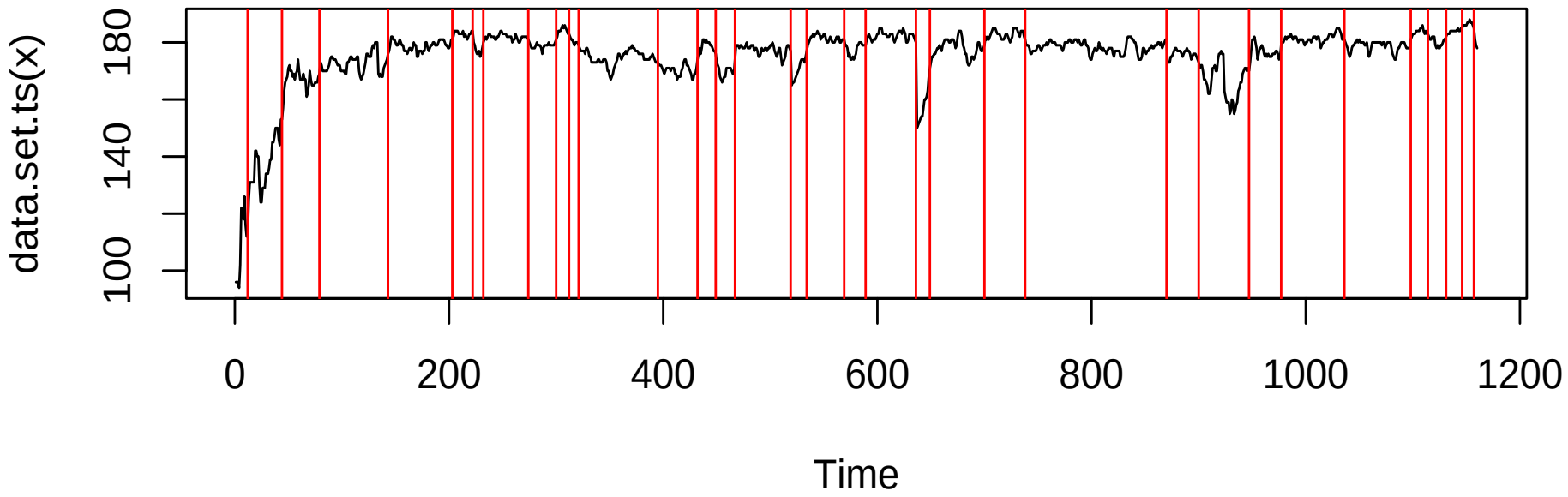
Mathematics & Statistics | Lancaster University

```
HR.pelt=cpt.np(HeartRate,method='PELT',
               nquantiles=4*log(length(HeartRate)))
ncpts(HR.pelt)
```

```
## [1] 33
```

A solution: HeartRate


```
plot(HR.pelt)
```



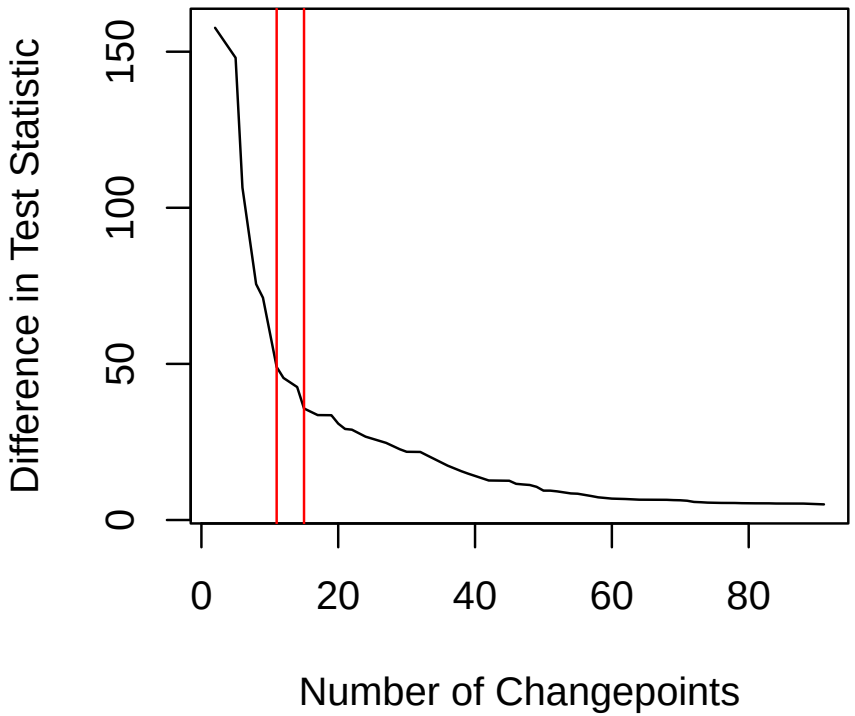
A solution: HeartRate


```
HR.crops=cpt.np(HeartRate, penalty = "CROPS",
        pen.value = c(5,200), method="PELT",minseglen=2,
        nquantiles =4*log(length(HeartRate)))
```

```
## [1] "Maximum number of runs of algorithm = 91"
## [1] "Completed runs = 2"
## [1] "Completed runs = 3"
## [1] "Completed runs = 5"
## [1] "Completed runs = 9"
## [1] "Completed runs = 17"
## [1] "Completed runs = 32"
## [1] "Completed runs = 52"
## [1] "Completed runs = 75"
## [1] "Completed runs = 84"
```

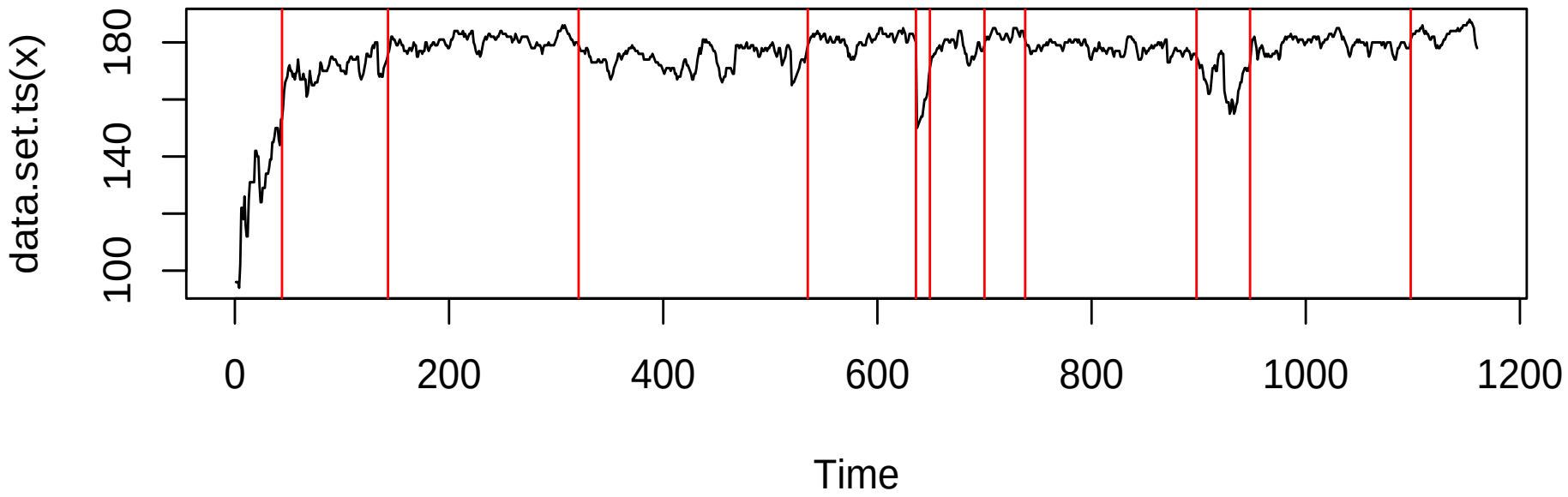
A solution: HeartRate


```
plot(HR.crops, diagnostic = TRUE)
abline(v=11,col='red')
abline(v=15,col='red')
```



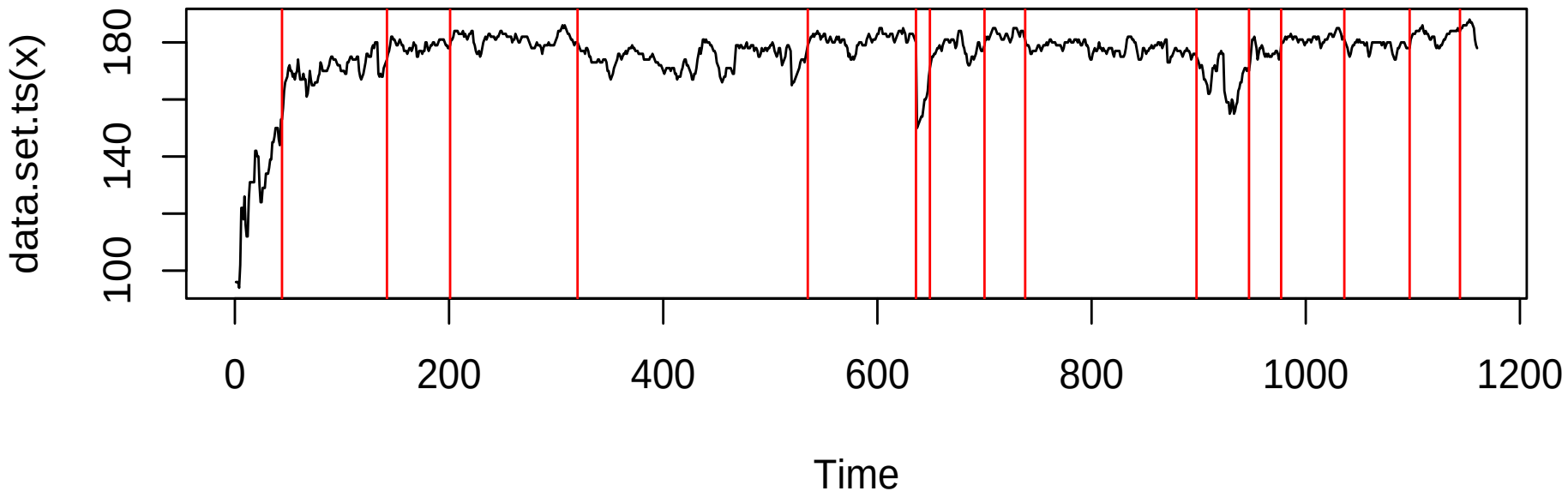
A solution: HeartRate


```
plot(HR.crops, ncpts = 11)
```



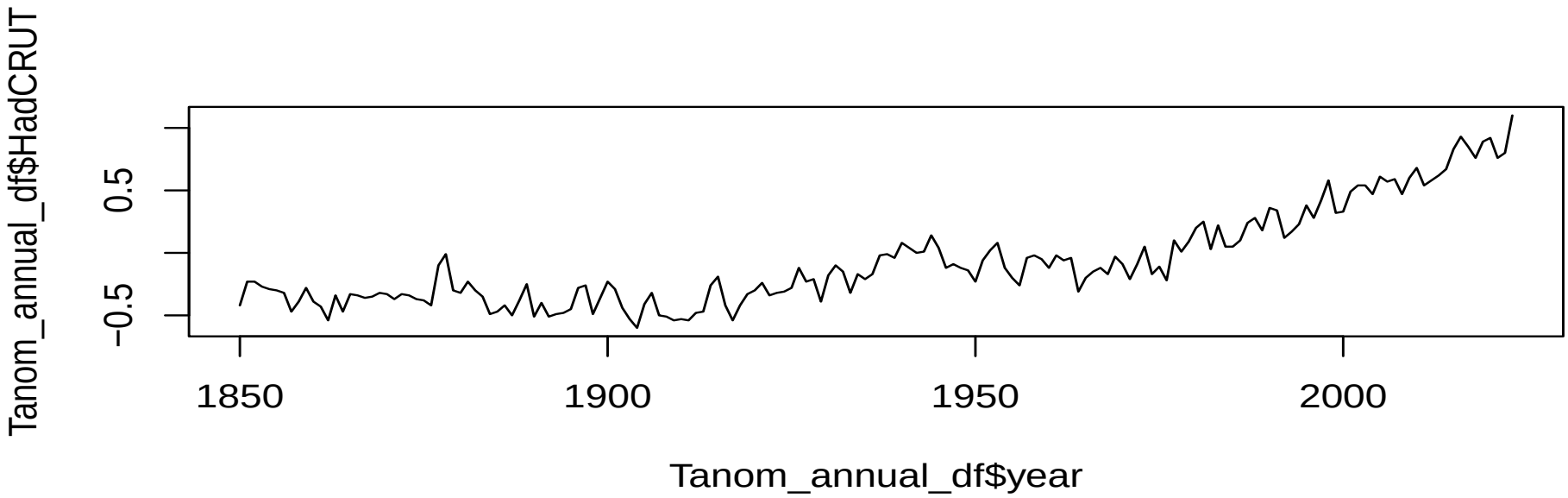
A solution: HeartRate


```
plot(HR.crops, ncpts = 15)
```



Trend and Autocorrelation


```
load('temperature_anomalies_updated.RData')
plot(Tanom_annual_df$year,Tanom_annual_df$HadCRUT,type='l')
```



T+AC Model

A simple linear trend and AR(1) model would be:

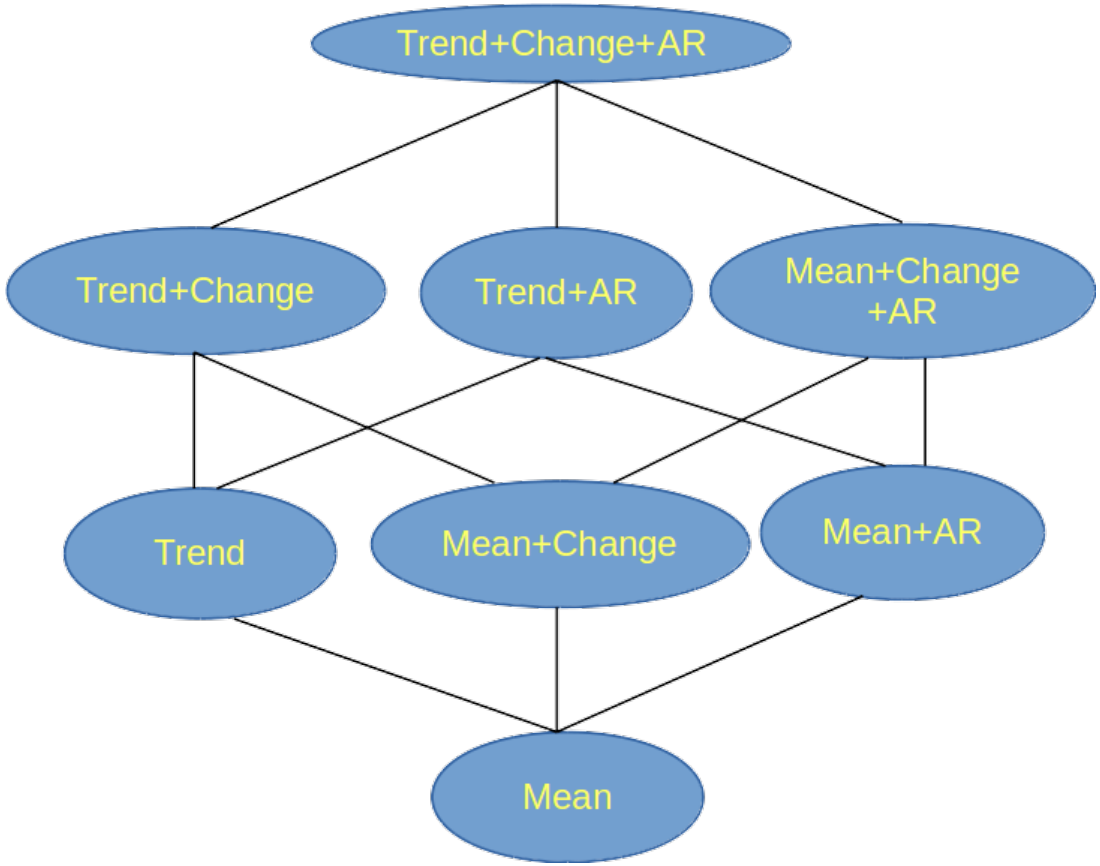
$$y_t = \begin{cases} \alpha_1 + \beta_1 t + \epsilon_t & \text{if} \quad 1 \leq t \leq \tau_1 \\ \alpha_2 + \beta_2 t + \epsilon_t & \text{if} \quad \tau_1 < t \leq \tau_2 \\ \vdots & \vdots \\ \alpha_{m+1} + \beta_{m+1} + \epsilon_t & \text{if} \quad \tau_m < t \leq \tau_{m+1} = n \end{cases}$$

where

$$\epsilon_t = \begin{cases} \phi_{1,1}(y_t - \alpha_1 - \beta_1 t) + Z_t & \text{if} \quad 1 \leq t \leq \tau_1 \\ \phi_{2,1}(y_t - \alpha_2 - \beta_2 t) + Z_t & \text{if} \quad \tau_1 < t \leq \tau_2 \\ \vdots & \vdots \\ \phi_{m+1,1}(y_t - \alpha_{m+1} - \beta_{m+1} t) + Z_t & \text{if} \quad \tau_m < t \leq \tau_{m+1} = n \end{cases}$$

and $Z_t \sim N(0, \sigma^2)$.

AIM: select the most parsimonious but accurate model for the data.



Simple to extend with other types of models.

EnvCpt

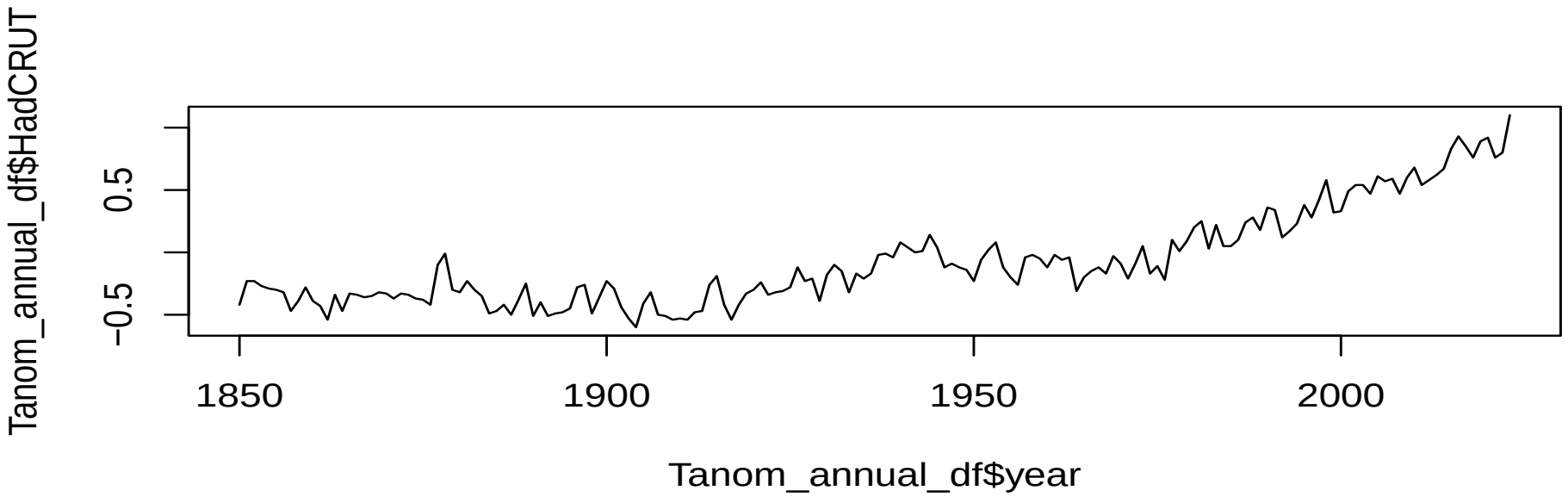

```
envcpt(data,models=c("mean","meancpt","meanar1","meanar2",
"meanar1cpt","meanar2cpt","trend","trendcpt","trendar1",
"trendar2","trendar1cpt","trendar2cpt"), minseglen=5,...,
verbose=TRUE)
```

- data is a vector or `ts` object
- models is a character vector of models to fit, default is all
- minseglen is the minimum number of observations between changes
- verbose prints progress through model fitting
- `...` are any other parameters to the `cpt.*` functions

Returns S3 object, has AIC and plot methods.

Task: Annual Temperatures


```
load('temperature_anomalies_updated.RData')
plot(Tanom_annual_df$year,Tanom_annual_df$HadCRUT,type='l')
```



Task: Annual Temperatures

Use the `envcpt` function to see if there is evidence for a change in the temperature data.

```
load('temperature_anomalies_updated.RData')
```

The data is available in `Tanom_annual_df` with several groups (columns) to choose from.

What is the best model? How many changepoints?

Don't forget to

```
library(EnvCpt)
```

```
## Loading required package: MASS
```

before you start.

A solution: NASA


```
library(EnvCpt)
nasa.envcpt=envcpt(Tanom_annual_df$NASA[
    !is.na(Tanom_annual_df$NASA)], verbose=F)
```

```
## Warning in arima(data, order = c(1, 0, 0), method = "CSS-M
## convergence problem: optim gave code = 1
```

```
which.min(AIC(nasa.envcpt))
```

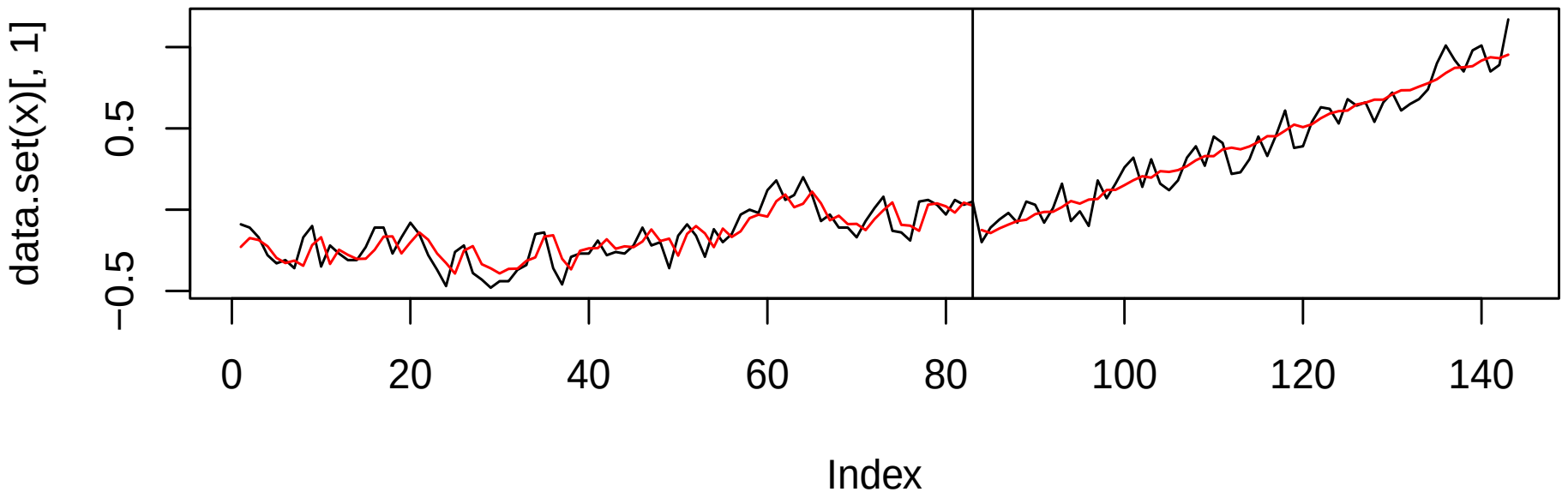
```
## trendar1cpt
##          11
```

```
Tanom_annual_df$year[!is.na(Tanom_annual_df$NASA)
                ][cpts(nasa.envcpt$trendar1cpt)]
```

```
## [1] 1962
```

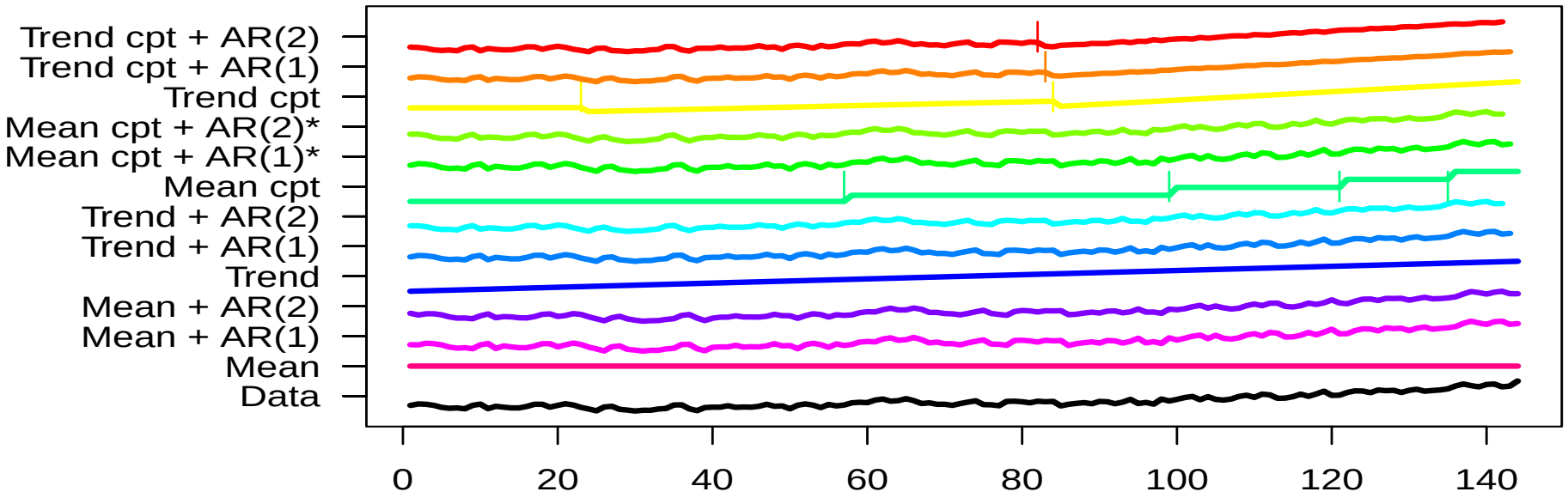
A solution: NASA


```
plot(nasa.envcpt$trendar1cpt)
abline(v=cpts(nasa.envcpt$trendar1cpt))
```



A solution: NASA


```
plot(nasa.envcpt)
```



Checking Assumptions

The main assumptions for a Normal likelihood ratio test for a change in mean are:

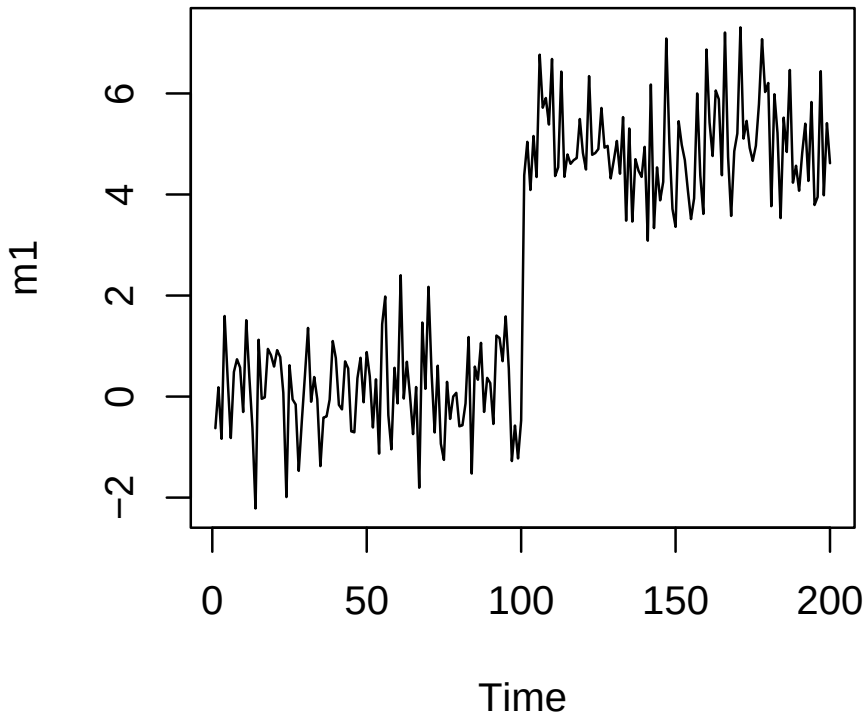
- Independent data points;
- Normal distributed points pre and post change;
- Constant variance across the data.

How can we check these?

Checking Assumptions

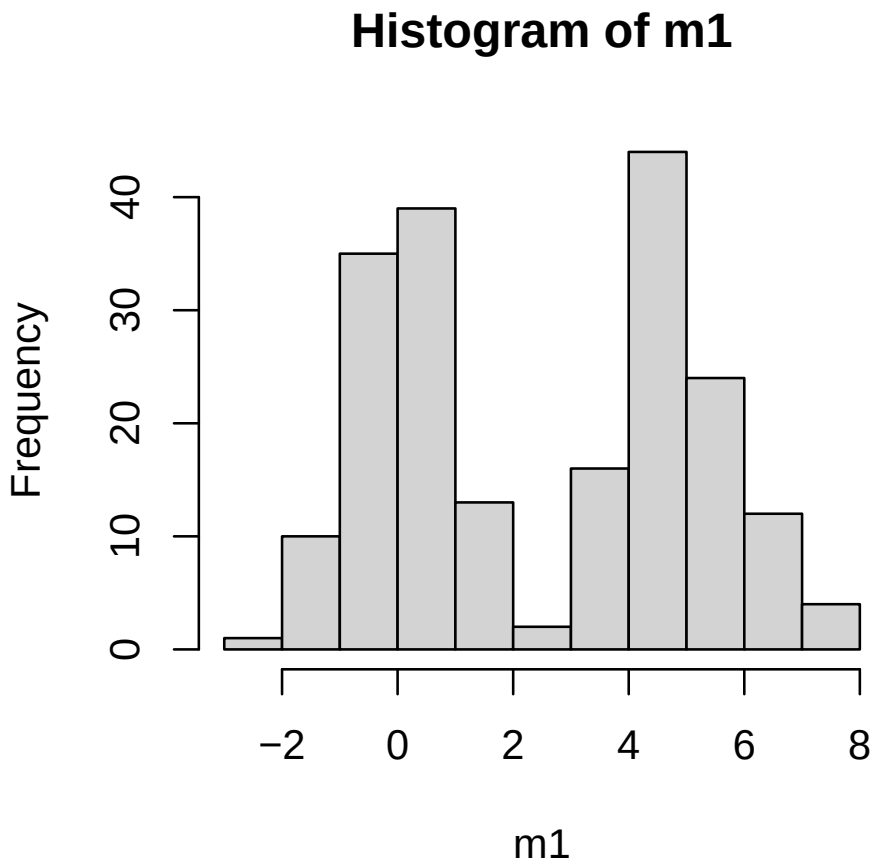
In reality we can't check assumptions prior to analysis.

```
ts.plot(m1)
```



Checking Assumptions


```
hist(m1)
```



Checking Assumptions


```
shapiro.test(m1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  m1
## W = 0.91086, p-value = 1.31e-09
```

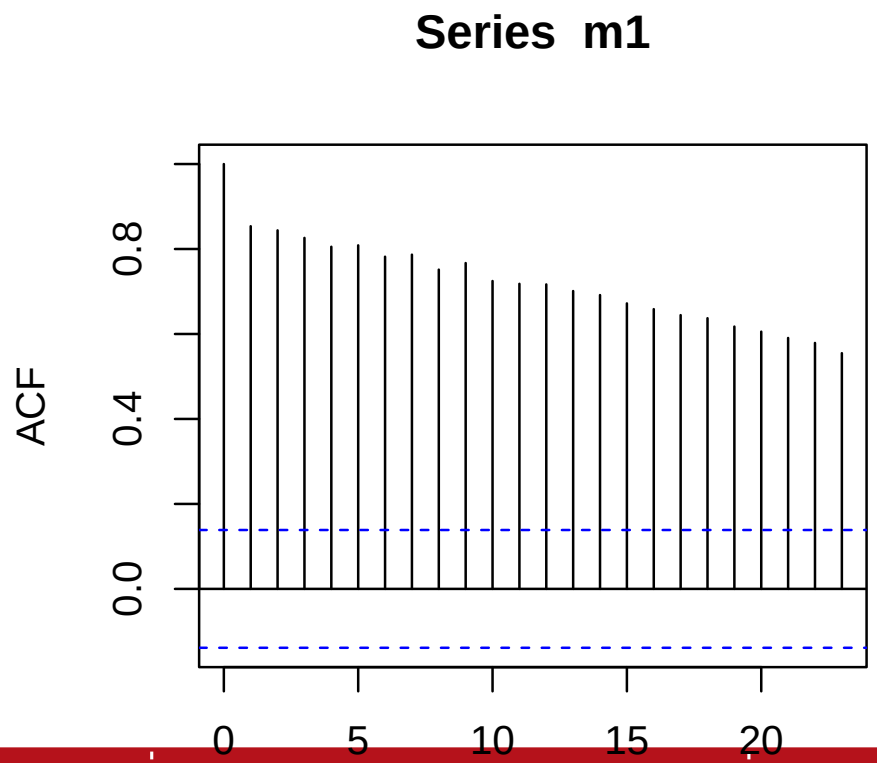
```
ks.test(m1,pnorm,mean=mean(m1),sd=sd(m1))
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  m1
## D = 0.15491, p-value = 0.0001355
## alternative hypothesis: two-sided
```

Checking Assumptions


```
acf(m1)
library(WeightedPortTest)
Weighted.Box.test(m1, lag=min(50,length(m1)/4))$p.value
```

```
## Approximate p-value
##                       0
```



How to check

- Check the residuals

```
means=param.est(m1.amoc)$mean
m1.resid=m1-rep(means,seg.len(m1.amoc))
shapiro.test(m1.resid)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  m1.resid
## W = 0.99228, p-value = 0.3721
```

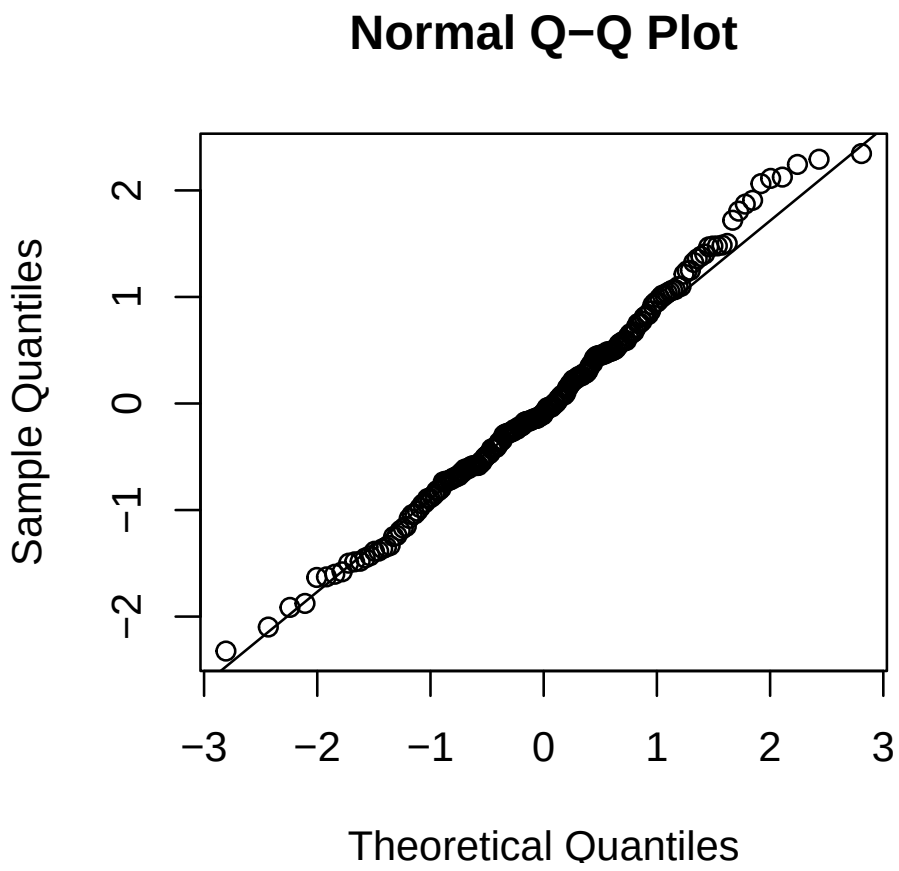
Residual Check


```
ks.test(m1.resid,pnorm,mean=mean(m1.resid),sd=sd(m1.resid))

##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  m1.resid
## D = 0.045812, p-value = 0.7953
## alternative hypothesis: two-sided
```

Residual Check

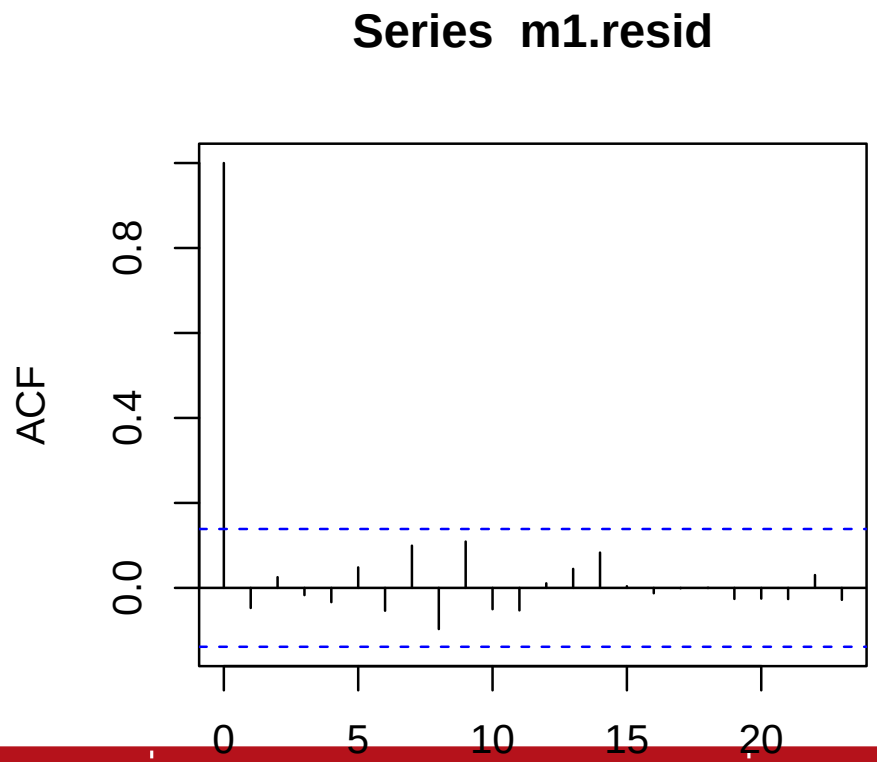

```
qqnorm(m1.resid)
qqline(m1.resid)
```



Residual Check


```
acf(m1.resid)
Weighted.Box.test(m1.resid,
            lag=min(50,length(m1.resid)/4))$p.value
```

```
## Approximate p-value
##           0.9313805
```



Task

Check the assumptions you have made on the simulated, Nile, HC1, HeartRate, and Temperature data using the residual check.

What effect might any invalid assumptions have on the inference?

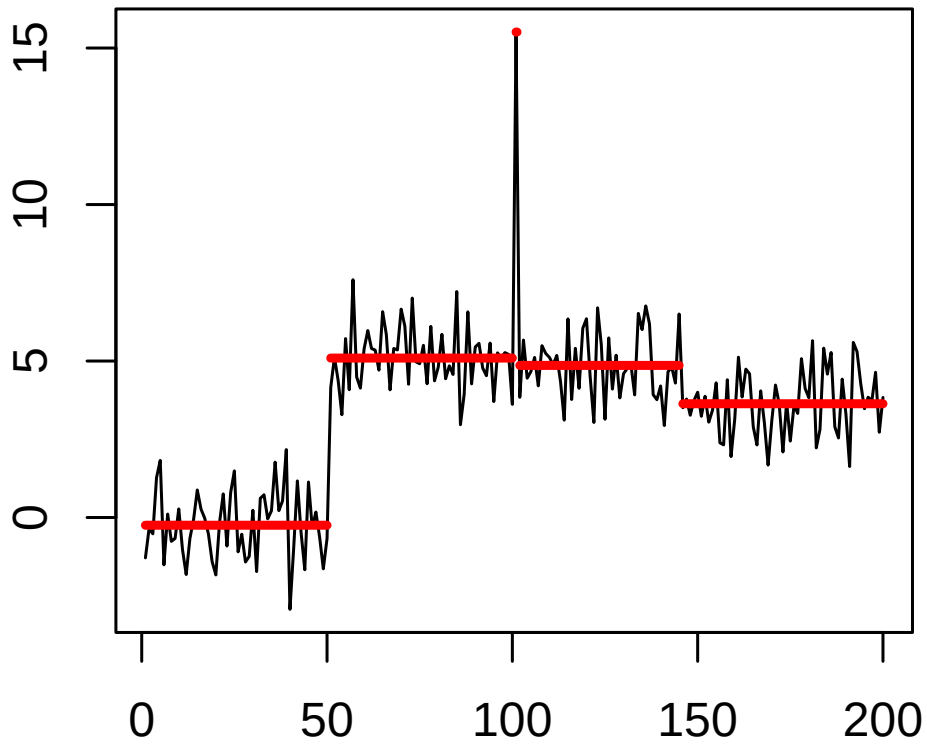
Influence

- Which data points are *influential* for obtaining the segmentation?
  - Changepoints versus Outliers
  - How to measure influence?
- How *stable* is the obtained segmentation?

Note: Currently the package considers mean changes but we are working on expansions. The premise is applicable to all model and changepoint forms. It is just the calulation of what you “expect” to happen that can be tricky.

Influence: Example


```
set.seed(30)
x=c(rnorm(50),rnorm(50,mean=5),rnorm(1,mean=15),
  rnorm(49,mean=5),rnorm(50,mean=4))
xcpt=cpt.mean(x,method='PELT')
plot(xcpt,cpt.width=3,ylab='')
```



How to measure?

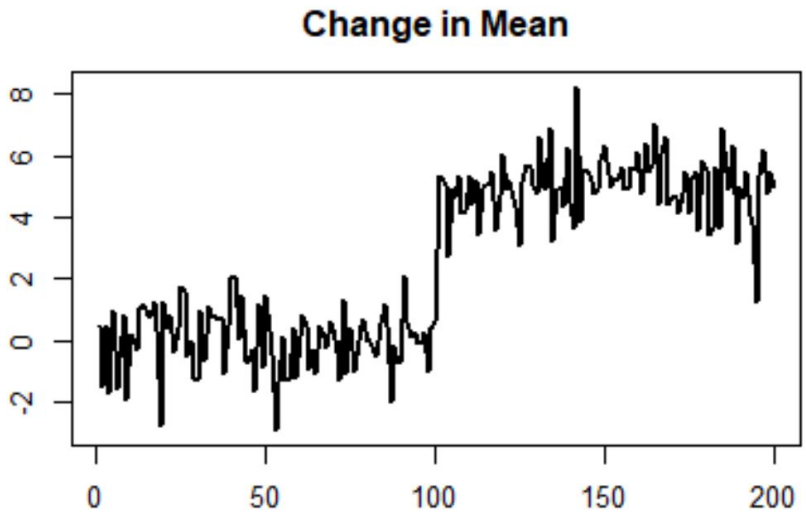
Sources of Inspiration:

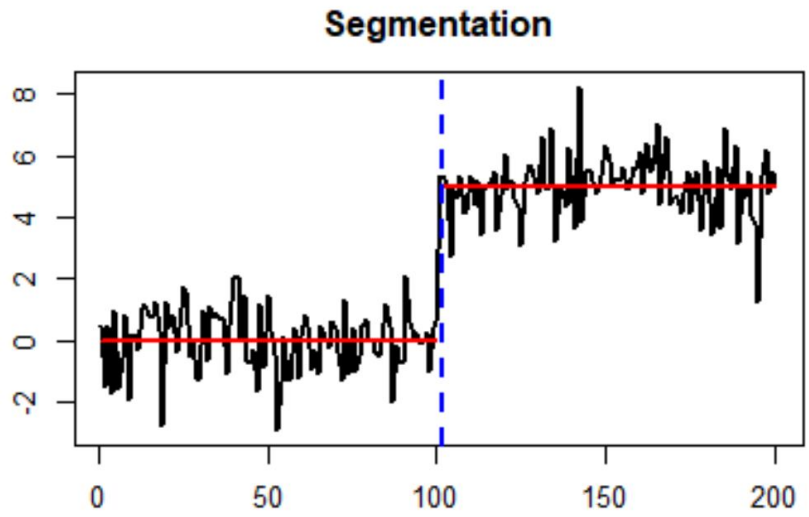
- Regression Analysis: Measures of Influence (e.g., Cook's distance)
- Robust Statistics: Influence Functions

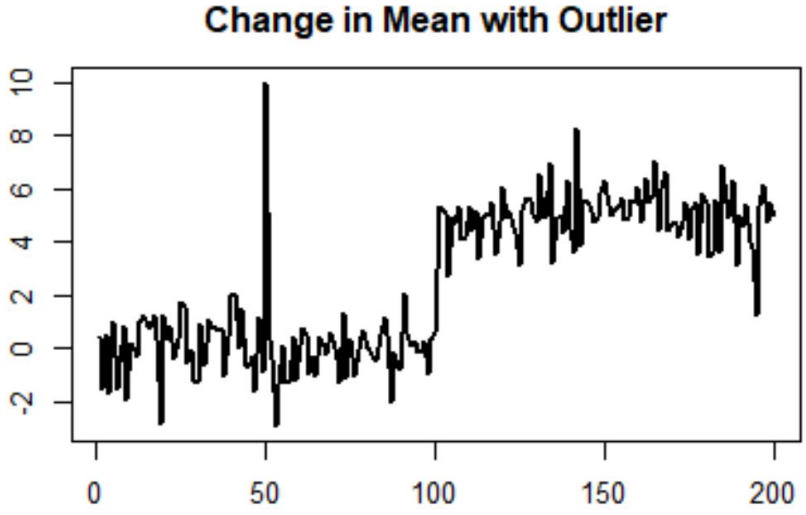
Two routes:

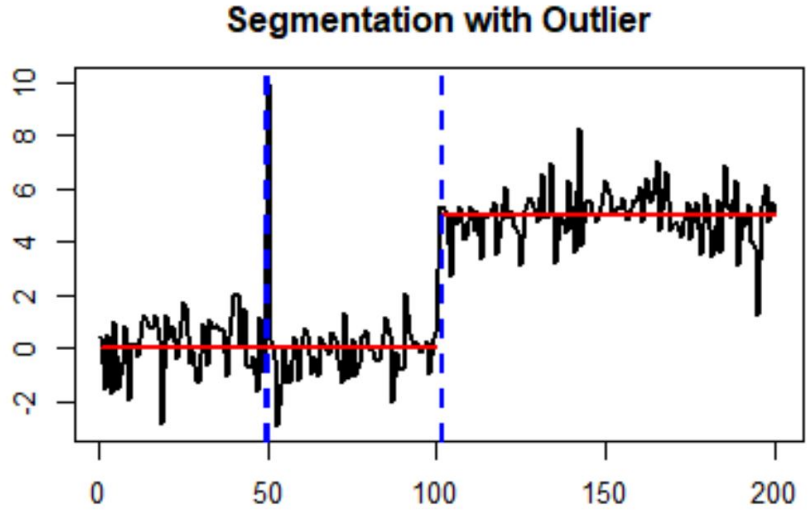
- Modifying an observation
- Leaving out an observation

Modify


Change in Mean
-2
0
2
4
6
8
0
50
100
150
200

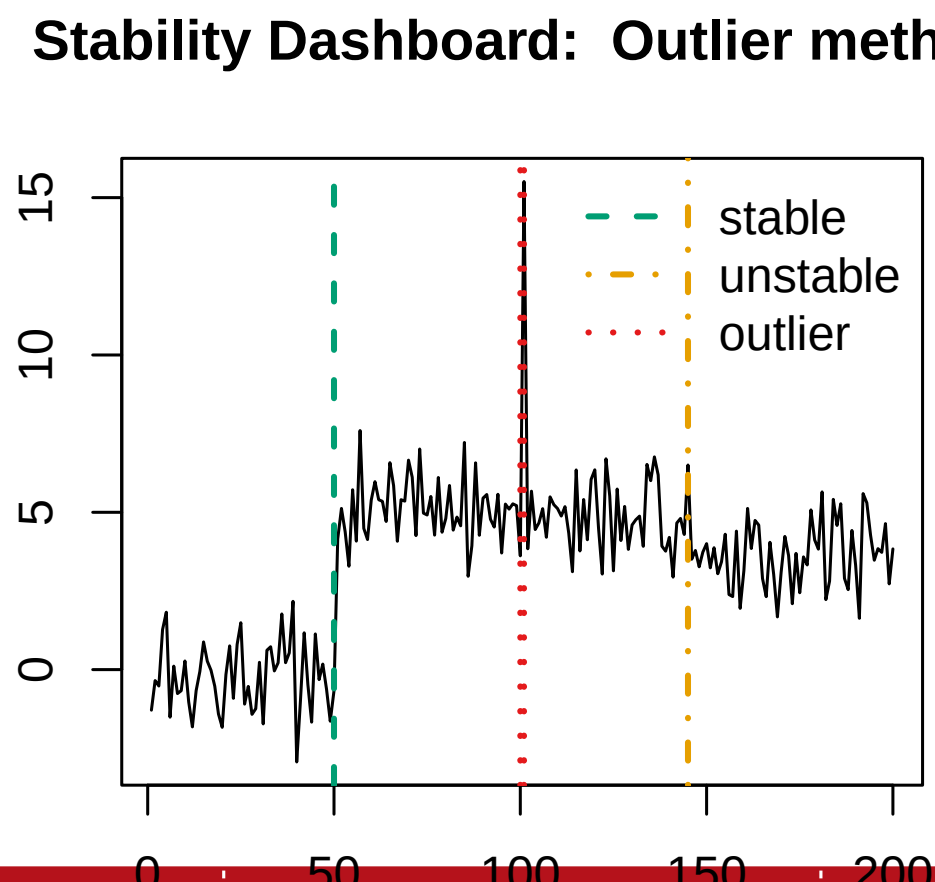

Segmentation
-2
0
2
4
6
8
0
50
100
150
200


Change in Mean with Outlier
-2
0
2
4
6
8
10
0
50
100
150
200


Segmentation with Outlier
-2
0
2
4
6
8
10
0
50
100
150
200

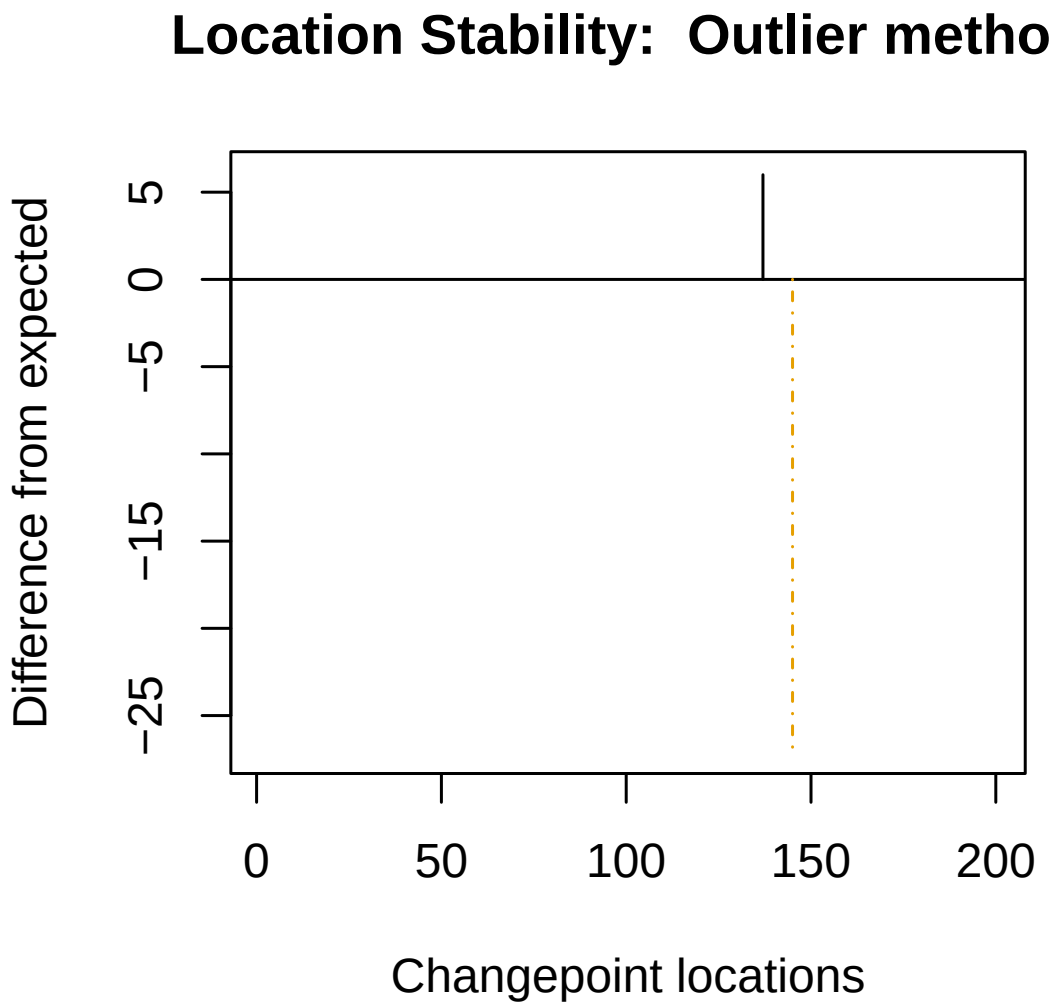
Stability Dashboard: Out


```
x.inf.out=influence(xcpt,method='outlier')
out.Stability=StabilityOverview(x,cpts(xcpt),x.inf.out,
  legend.args=list(display=TRUE,x="topright",y=NULL,cex=1,
  horiz=FALSE,xpd=FALSE,bty='n'))
```



Location Stability: Out

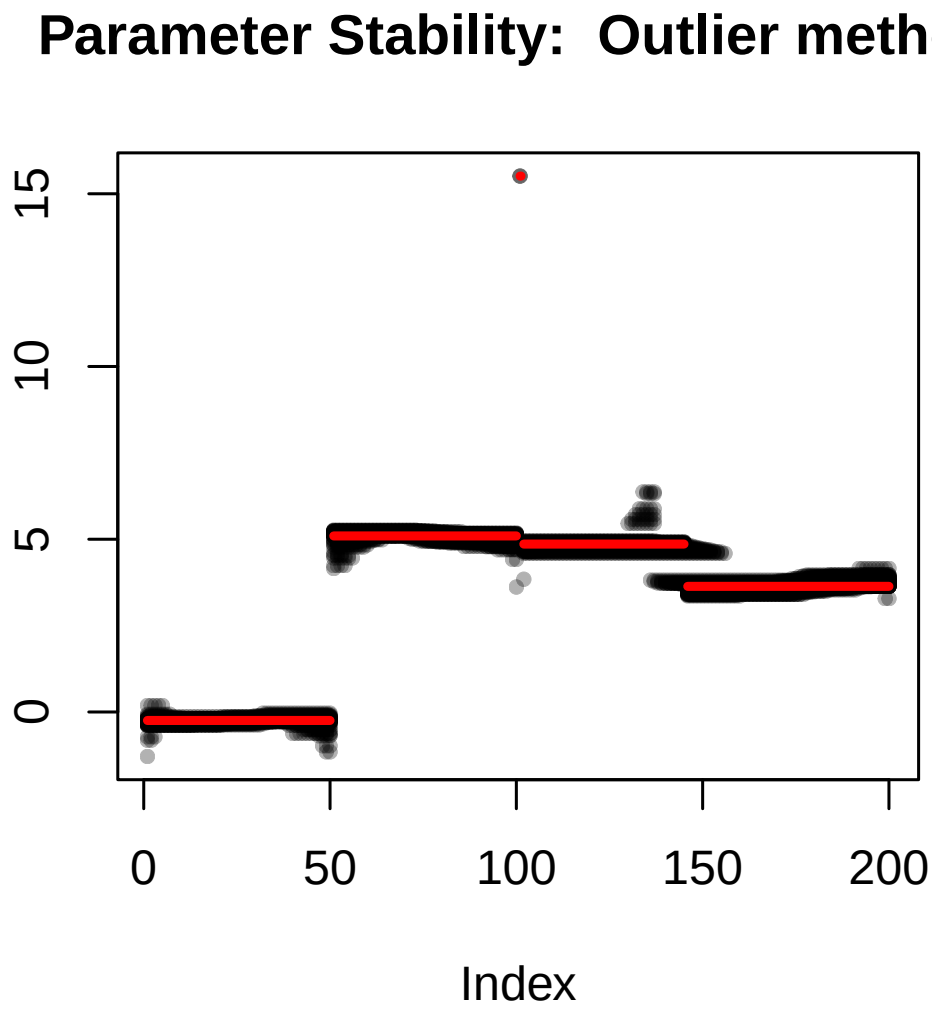

```
out.location=LocationStability(cpts(xcpt),x.inf.out,
  type='Difference')
```



Parameter Stability: Out

Mathematics & Statistics | Lancaster University

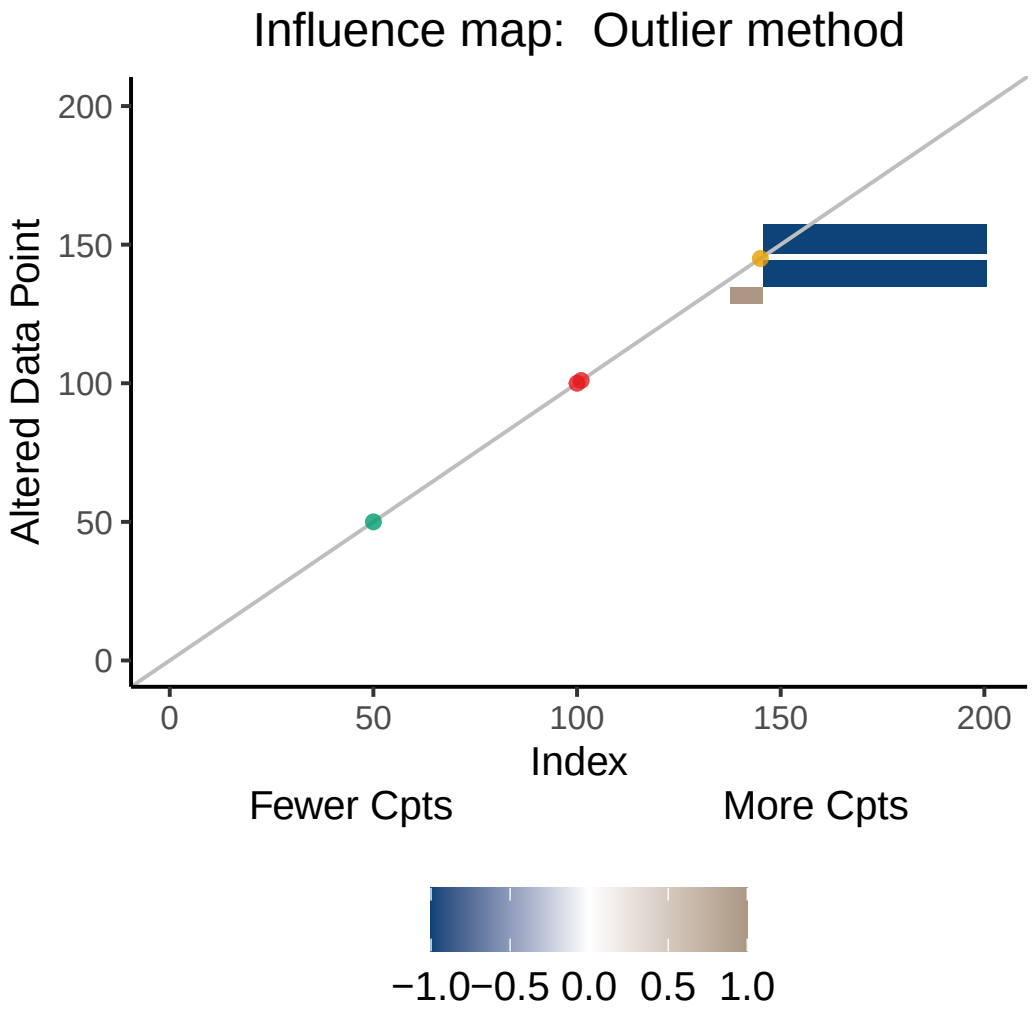
```
ParameterStability(x.inf.out,original.mean=rep(
  param.est(xcpt)$mean,times=diff(c(0,xcpt@cpts))))
```



Influence Map: Out

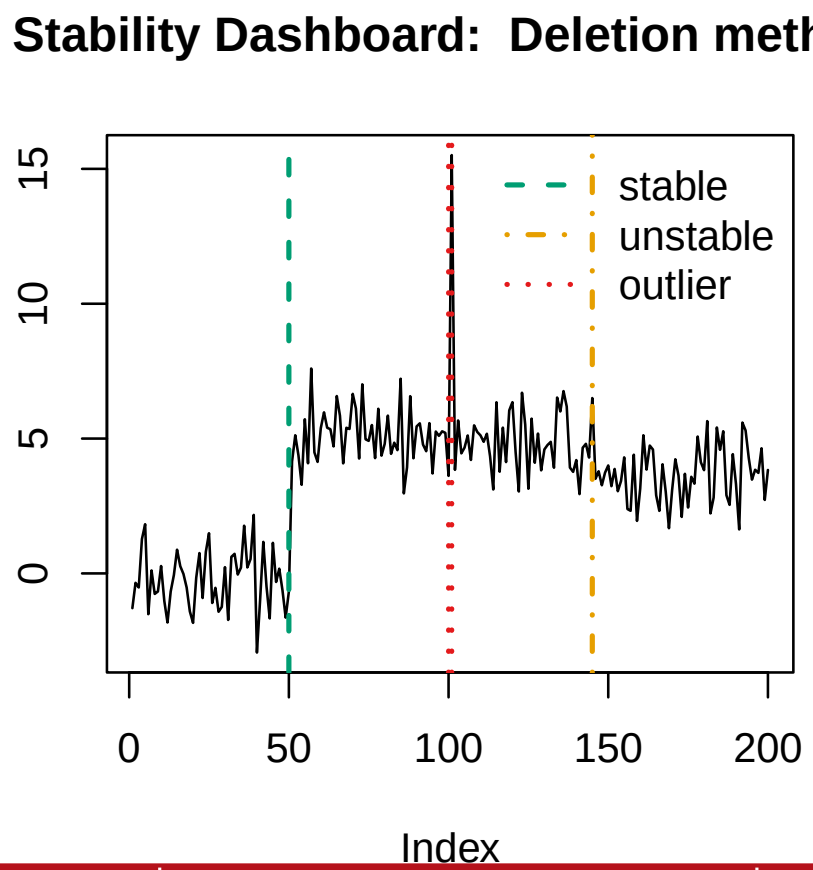
Mathematics & Statistics | Lancaster University

```
out.map=InfluenceMap(cpts(xcpt),x.inf.out)
```



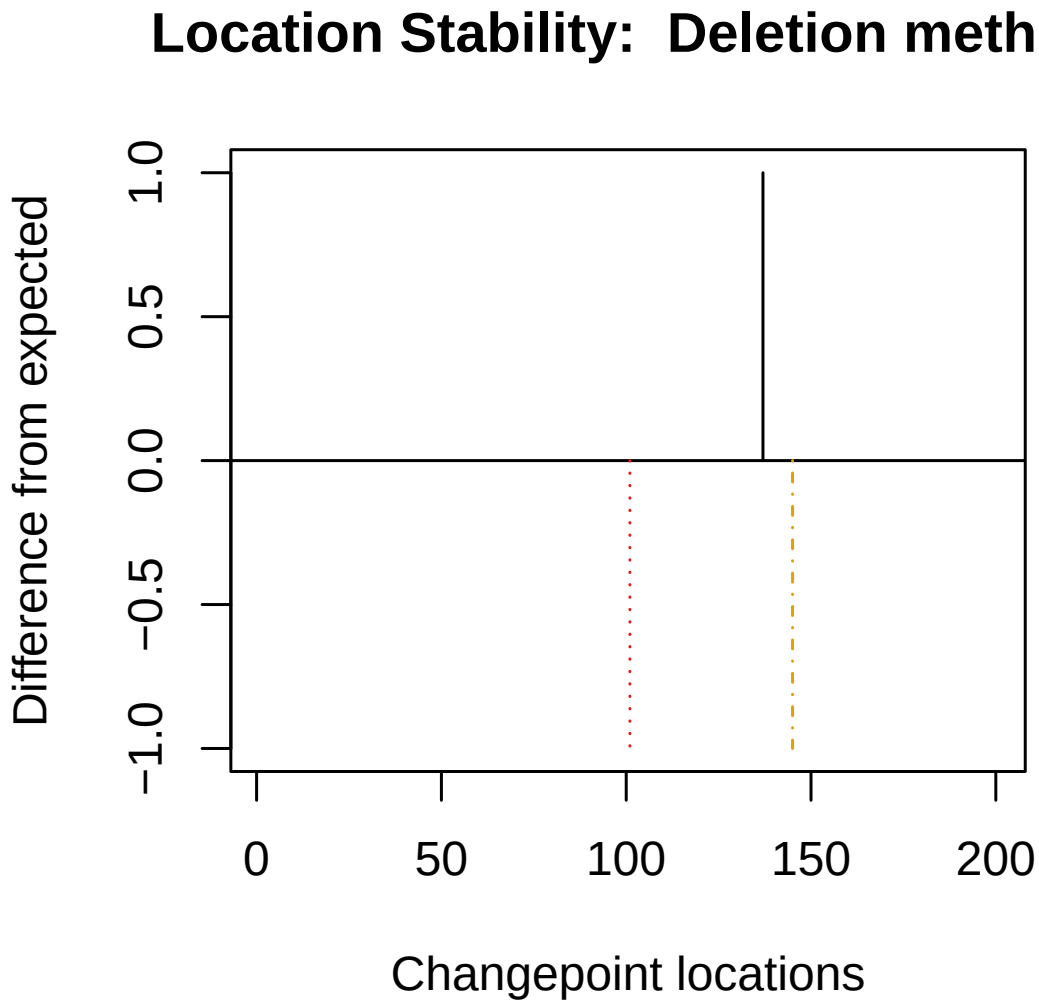
Stability Dashboard: Del


```
x.inf.del=influence(xcpt,method='delete')
del.Stability=StabilityOverview(x,cpts(xcpt),x.inf.del,
  legend.args=list(display=TRUE,x="topright",y=NULL,cex=1,
  horiz=FALSE,xpd=FALSE,bty='n'))
```



Location Stability: Del

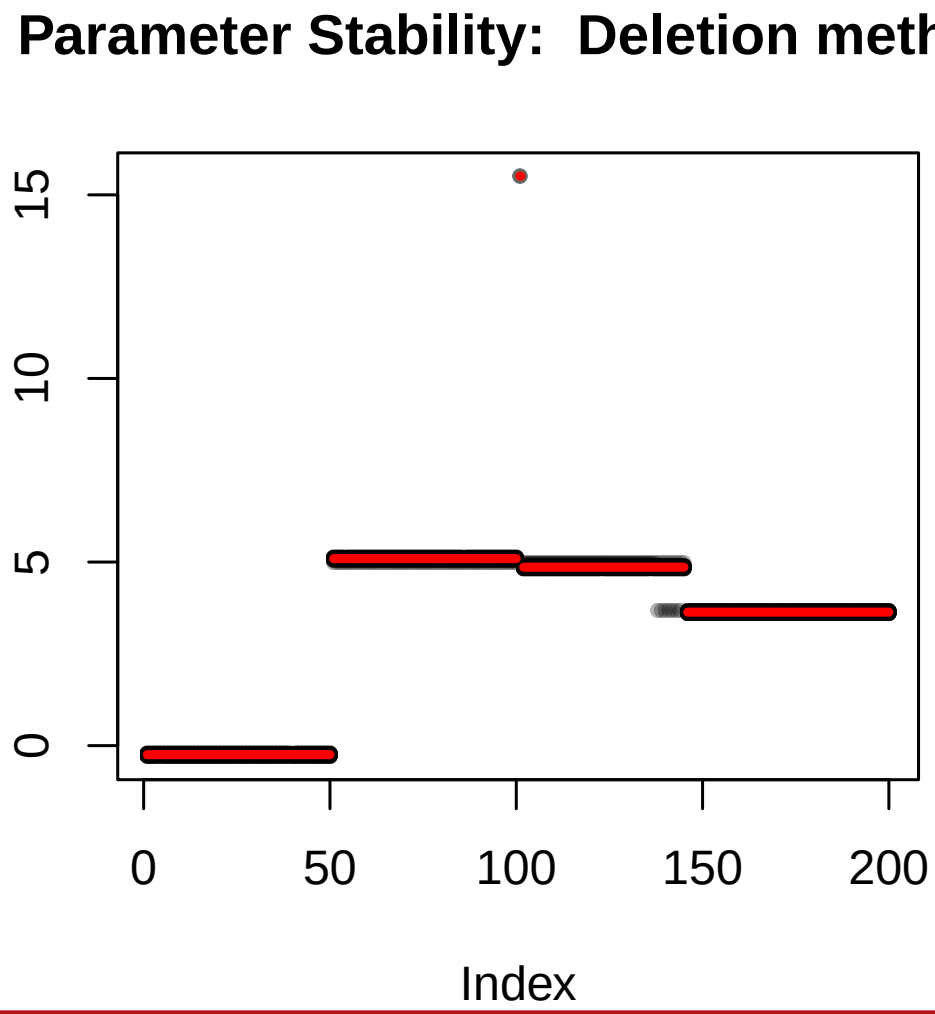

```
del.location=LocationStability(cpts(xcpt),x.inf.del,
  type='Difference')
```



Parameter Stability: Del

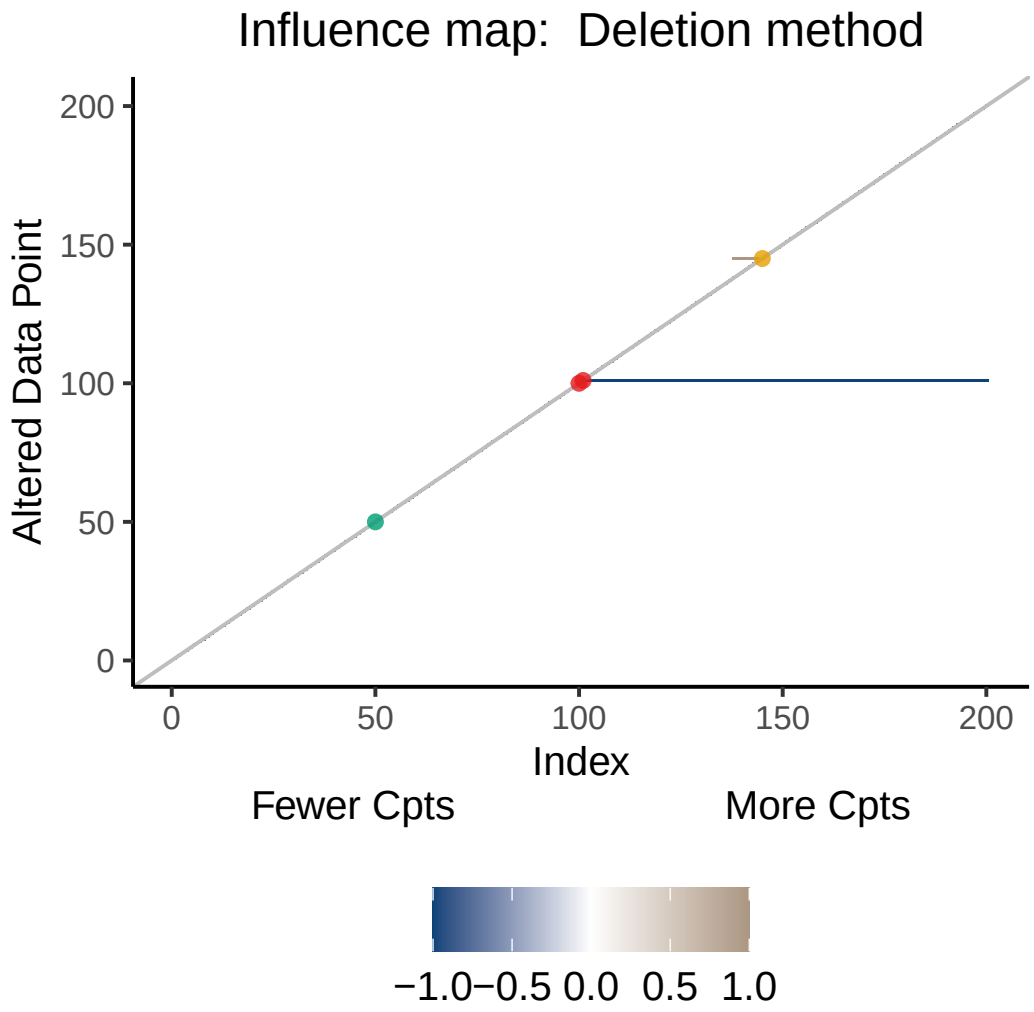
Mathematics & Statistics | Lancaster University

```
ParameterStability(x.inf.del,original.mean=rep(
  param.est(xcpt)$mean,times=diff(c(0,xcpt@cpts))))
```



Influence Map: Del


```
del.map=InfluenceMap(cpts(xcpt),x.inf.del)
```



Task: Nile Influence

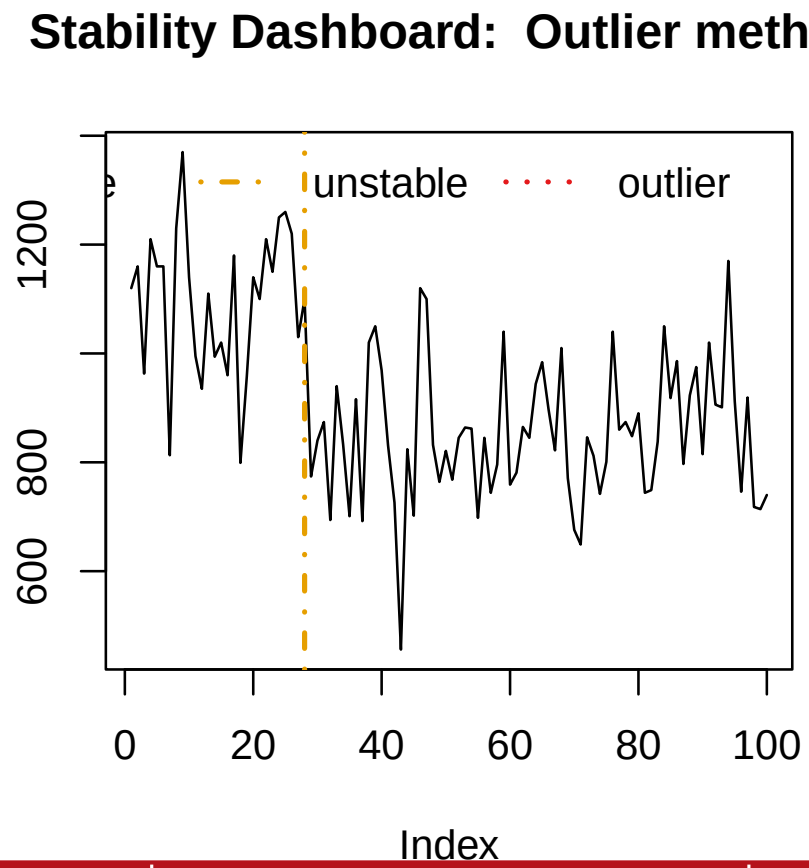

```
nile.cpts=cpt.mean(Nile,penalty="Manual",
                        pen.value=100000,method="PELT")
```

Look at the influence on the simulated, and Nile.

What effect might any invalid assumptions have on the influences you are seeing?

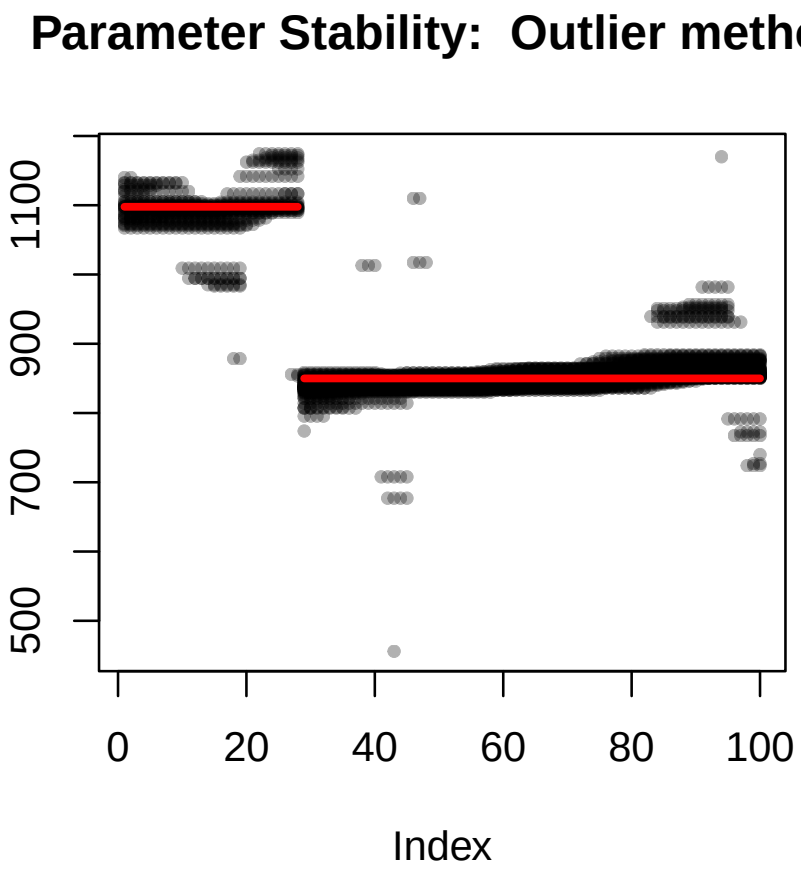
Task: Nile Influence


```
nile.inf=influence(nile.cpts,method="outlier")
stab=StabilityOverview(data.set(nile.cpts), cpts(nile.cpts),
    nile.inf,legend.args=list(display=T,x="topright",
    horiz=T,y=NULL,cex=1,bty="n",xpd=F))
```



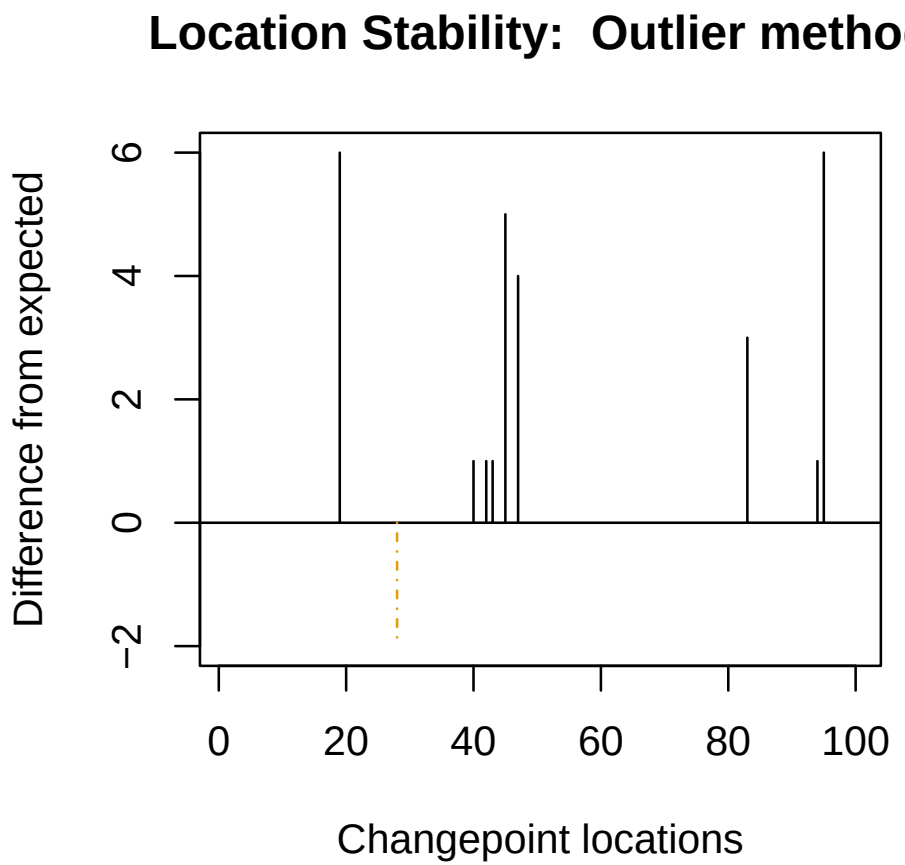
Task: Nile Influence


```
ParameterStability(nile.inf, original.mean=rep(
  param.est(nile.cpts)$mean,times=diff(c(0,nile.cpts@cpts)))
```



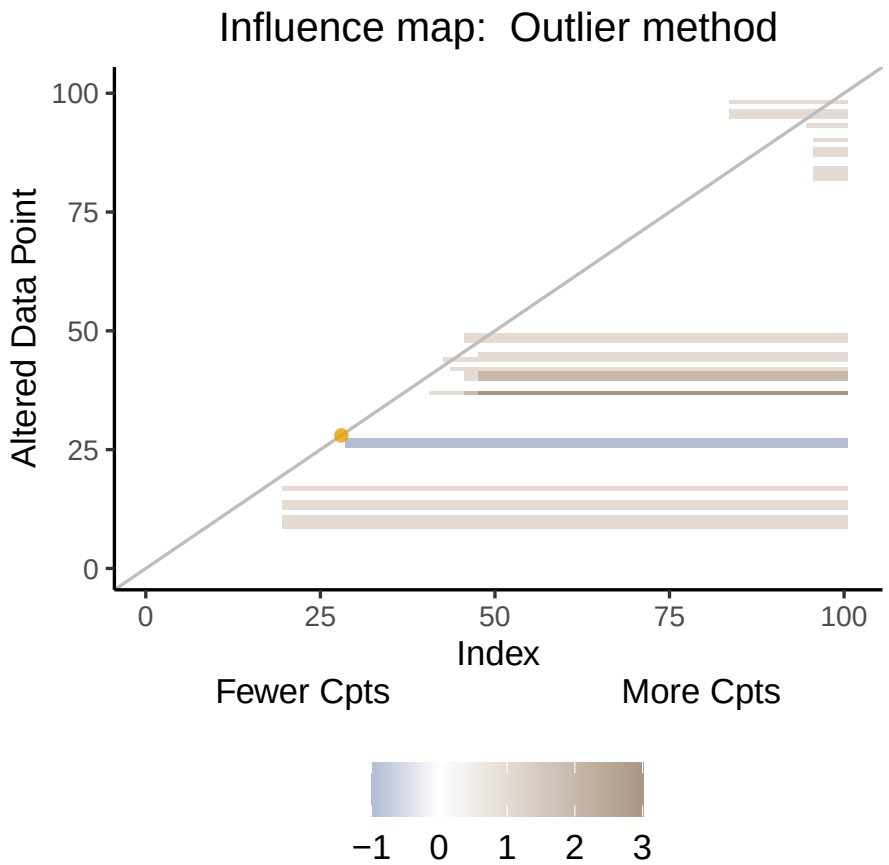
Task: Nile Influence


```
locstab=LocationStability(cpts(nile.cpts), nile.inf,
                type="Difference")
```



Nile Influence


```
inf=InfluenceMap(cpts(nile.cpts),nile.inf)
```



Consolidating Task

Mathematics & Statistics | Lancaster University

Download the ratings for the following TV shows from the IMDB and analyze the series using some of the techniques you have learnt from today. For each series, do you identify any changes? Are the assumptions you are making valid? What effect might any invalid assumptions have on the inference? Do the changepoints remain stable under influence?

- Doctor Who
- Grey's Anaytomy
- Mistresses
- The Simpsons
- Top Gear

(Understandably IMBD does not allow screen scraping nor downloads of information for redistribution so you will have to copy and paste the table into Excel, or equivalent, yourself in order to get the ratings data into R.)

Bonus

Just from looking at the data, can you predict which shows have been cancelled?

References

JSS: Killick, Eckley (2014)

PELT: Killick, Fearnhead, Eckley (2012)

CROPS: Kaynes, Eckley, Fearnhead (2015)

cpt.np: Haynes, Fearnhead, Eckley (2016)

EnvCpt: Beaulieu, Killick (2018)

Influence: Wilms, Killick, Matteson (2022)