

Testing for Structural Change

Theory, Implementation and Applications

Achim Zeileis



Dortmund, 2003-11-14



Testen auf Strukturveränderung

Theorie, Implementierung und Anwendungen

Achim Zeileis



Dortmund, 2003-11-14



Überblick

- ❄ Theorie:
Verallgemeinerte M-Fluktuationstests auf Parameterinstabilität
- ❄ Implementierung:
Strukturbruchtests im R-Paket `strucchange`
- ❄ Anwendung:
Testen und Datierung von Strukturbrüchen

Was ist ein Strukturbruch?

Betrachtet wird das lineare Regressionsmodell

$$y_i = x_i^\top \beta_i + u_i \quad (i = 1, \dots, n),$$

wobei zum Zeitpunkt i :

- ❄ y_i — abhängige Variable,
- ❄ x_i — Regressorvektor mit k Elementen,
- ❄ β_i — Vektor von k unbekanntem Regressionskoeffizienten,
- ❄ u_i — Fehlerterm.

Was ist ein Strukturbruch?

Getestet wird die Nullhypothese

$$H_0 : \beta_i = \beta_0 \quad (i = 1, \dots, n)$$

gegen

$$H_1 : \beta_i \text{ ändert sich mit der Zeit}$$

→ verallgemeinerte Fluktuationstests.

Was ist ein Strukturbruch?

Getestet wird die Nullhypothese

$$H_0 : \beta_i = \beta_0 \quad (i = 1, \dots, n)$$

gegen

$$H_1 : \beta_i \text{ ändert sich mit der Zeit}$$

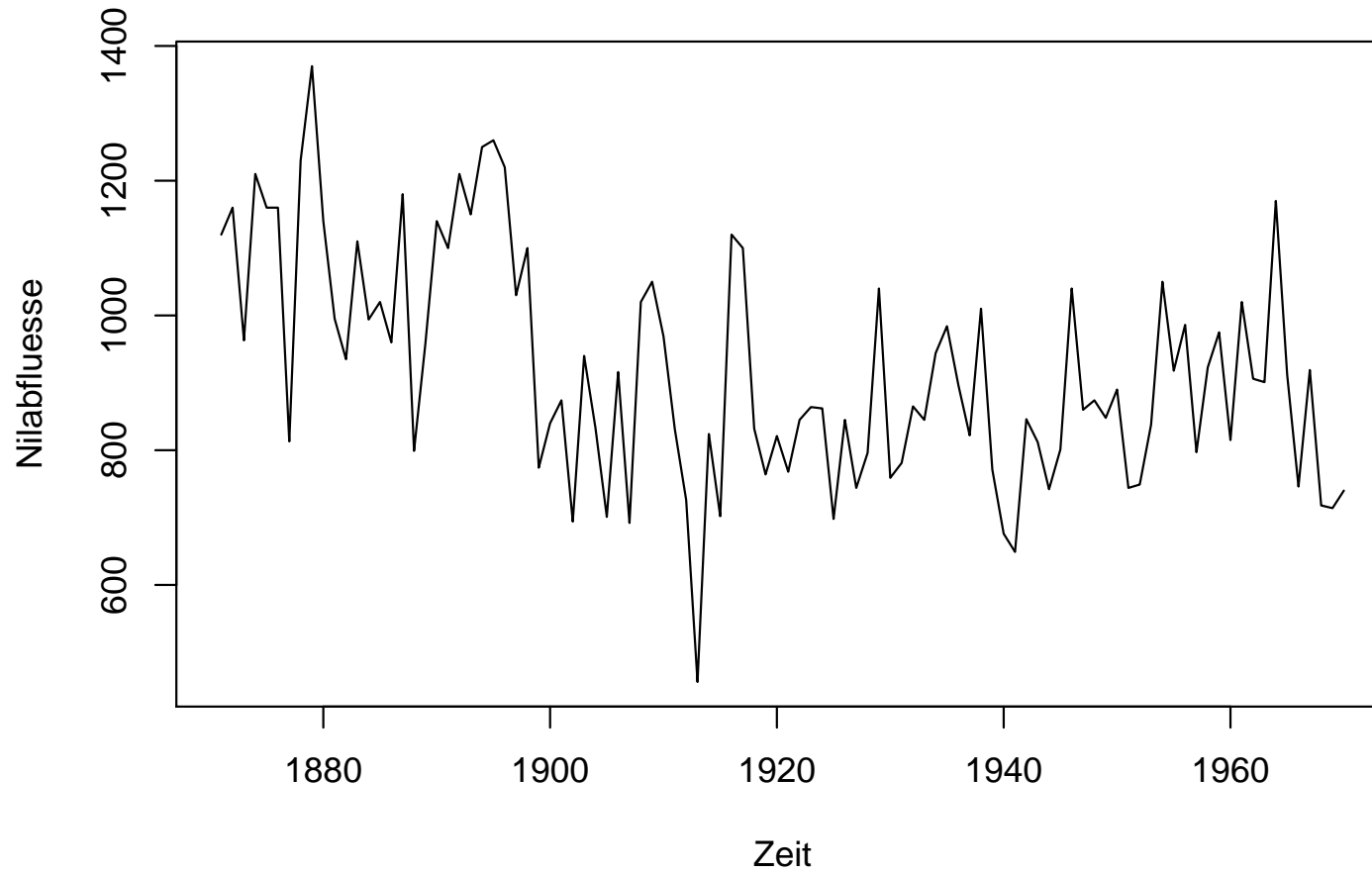
→ verallgemeinerte Fluktuationstests.

bzw.

$$H_1^* : \beta_i = \begin{cases} \beta_A & (1 \leq i \leq i_0) \\ \beta_B & (i_0 < i \leq n) \end{cases}$$

→ F -Tests.

Warum Strukturbrüche?



Warum Strukturbrüche?

Weshalb sollte man auf Strukturveränderung testen wollen?

- ❄ Zentrales Interesse in vielen Anwendungen: ob, wann und wie ändert sich der datengenerierende Prozeß?
- ❄ Nachweis einer Änderung durch Signifikanztest,
- ❄ auch als diagnostischer Test, da Modelle mit nicht beachteter Parameterinstabilität zu uninterpretierbaren Parameterschätzungen, verzerrter Inferenz und ungenauen Prognosen führen,
- ❄ exploratives Hilfsmittel zur Entdeckung von Strukturveränderungen.

Verallgemeinerte Fluktuationstests

- ❄ Empirischer Fluktuationsprozeß spiegelt Fluktuation wider in
 - ❖ Residuen,
 - ❖ Koeffizientenschätzungen,
 - ❖ *M-scores* (bspw. KQ or ML scores).
- ❄ Theoretischer Grenzprozeß ist bekannt.
- ❄ Wahl von Schranken, die vom Grenzprozeß (oder einem Funktional dessen) mit einer bekannten Wahrscheinlichkeit α überschritten werden.
- ❄ Überschreitet der empirische Fluktuationsprozeß die theoretischen Schranken, so ist die Fluktuation unwahrscheinlich groß \Rightarrow Nullhypothese verwerfen.

Philosophie

The generalized fluctuation test framework ...

*“... includes formal significance tests but its philosophy is basically that of data analysis as expounded by Tukey. Essentially, the techniques are designed to **bring out departures from constancy in a graphic way** instead of parametrizing particular types of departure in advance and then developing formal significance tests intended to have high power against these particular alternatives.”* (Brown, Durbin, Evans, 1975)

Generalisierte M-Fluktuationstests

Wie können Abweichungen von der Annahme $\theta_i = \theta_0$ eingefangen werden? Betrachte eine k - bzw. $k + 1$ -dimensionale Score-Funktion $\psi(\cdot)$ mit

$$E_{\theta_i = \theta_0}[\psi(y_i, x_i, \theta_0)] = 0$$

Annahmen:

- ❄ Erwartungswert 0 bzgl. $f(y_i | x_i, \theta_0)$
- ❄ Stabilisierende Varianzen:

$$\frac{1}{n} \sum_{i=1}^n \text{COV}[\psi(y_i, x_i, \theta_0)] = J_n \xrightarrow{p} J,$$

Generalisierte M-Fluktuationstests

Typische Wahl für ψ ist die partielle Ableitung einer Zielfunktion Ψ

$$\psi(y, x, \theta) = \frac{\partial \Psi(y, x, \theta)}{\partial \theta}.$$

Beispielsweise: KQ oder ML.

Anstatt der Spezifikation einer vollen Likelihood:

- ❄ Mißspezifikationsmodelle: Quasi-ML, robuste M-Schätzung.
- ❄ Schätzgleichungen: Hilfsvariablen (IV), Verallgemeinerte Momentenmethode (GMM), Verallgemeinerte Schätzgleichungen (GEE).

Theor. M-Fluktuationsprozesse

Aggregation über die Zeit zu einem kumulativen Fluktuationsprozeß:

$$W_n(t, \theta) = n^{-1/2} \sum_{i=1}^{\lfloor nt \rfloor} \psi(y_i, x_i, \theta).$$

Theor. M-Fluktuationsprozesse

Aggregation über die Zeit zu einem kumulativen Fluktuationsprozeß:

$$W_n(t, \theta) = n^{-1/2} \sum_{i=1}^{\lfloor nt \rfloor} \psi(y_i, x_i, \theta).$$

Unter H_0 gilt der folgende zentrale Grenzwertsatz:

$$J^{-1/2} W_n(\cdot, \theta_0) \xrightarrow{d} W(\cdot).$$

Empirische M-Fluktuationsprozesse

Eine geeignete Schätzung $\hat{\theta}_n$ von θ_0 ist gegeben durch:

$$\sum_{i=1}^n \psi(y_i, x_i, \hat{\theta}_n) = 0.$$

Empirische M-Fluktuationsprozesse

Eine geeignete Schätzung $\hat{\theta}_n$ von θ_0 ist gegeben durch:

$$\sum_{i=1}^n \psi(y_i, x_i, \hat{\theta}_n) = 0.$$

Unter H_0 gilt der folgende zentrale Grenzwertsatz:

$$\hat{J}_n^{-1/2} W_n(\cdot, \hat{\theta}_n) \xrightarrow{d} W^0(\cdot),$$

falls \hat{J}_n ein konsistenter Schätzer der Kovarianzmatrix J .

Warum Implementierung?

- ❄️ potentiellen Benutzern ermöglichen theoretische Resultate auf Daten anzuwenden,
- ❄️ Reproduzierbarkeit von Forschungsergebnissen,
- ❄️ Kommunikation von statistischen Ideen und dem Verständnis über die Umsetzung von Theorie in Praxis.

strucchange

strucchange bietet vereinheitlichte und flexible Werkzeuge, um:

- ❄ Fluktuationsprozesse und F -Teststatistiken anzupassen,
- ❄ die Prozesse/Statistiken zusammen mit ihren Schranken darzustellen,
- ❄ so potentielle Bruchpunkte graphisch zu illustrieren,
- ❄ Signifikanztests durchzuführen.

strucchange

Weitere Funktionalität:

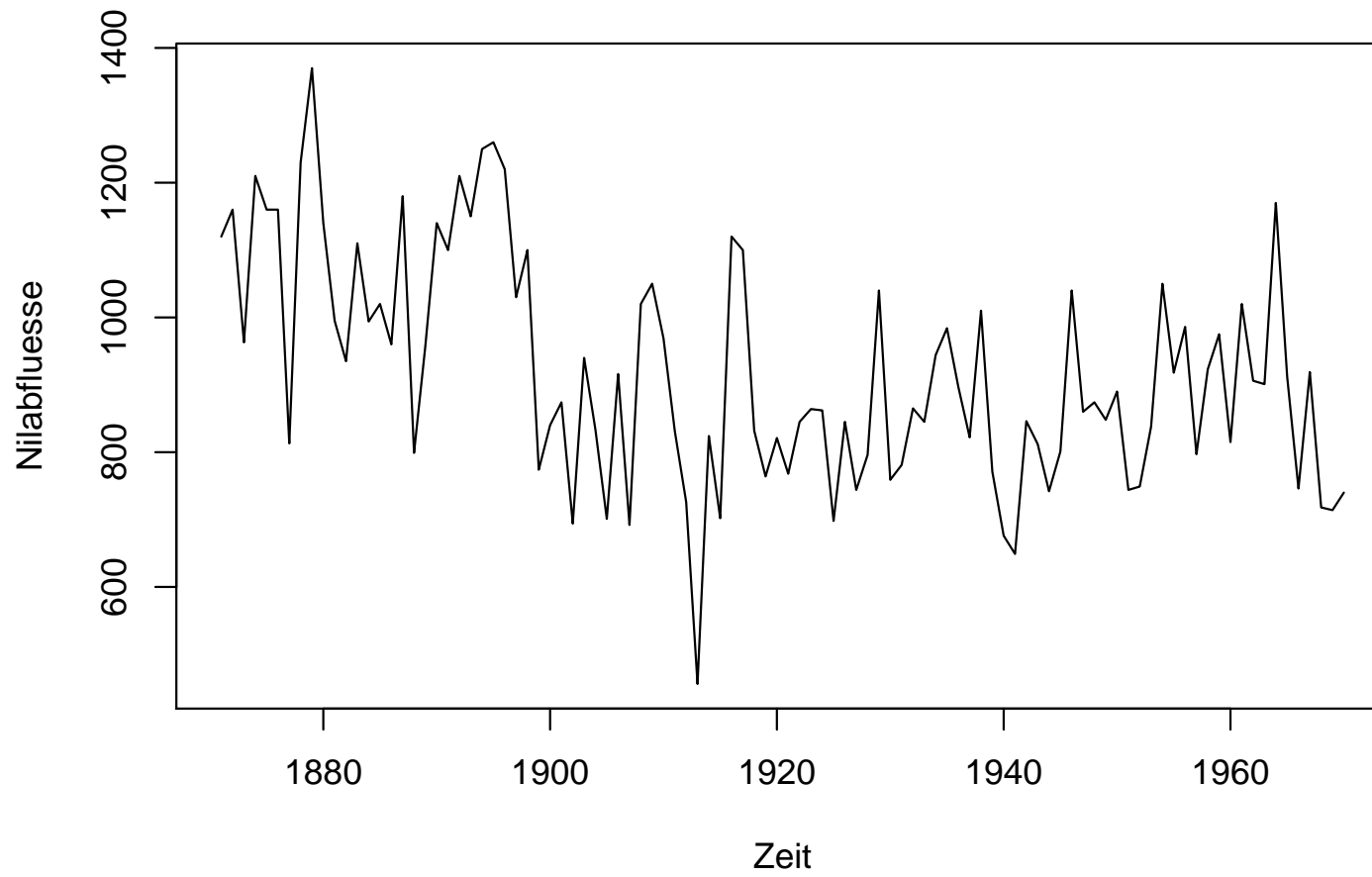
❄ Überwachung von Strukturveränderung mit Fluktuationstests basierend auf:

- ❖ KQ-Residuen,
- ❖ Koeffizientenschätzungen.

❄ Datierung von Strukturbrüchen:

- ❖ Bruchpunktschätzung,
- ❖ Konfidenzintervalle.

Nildaten

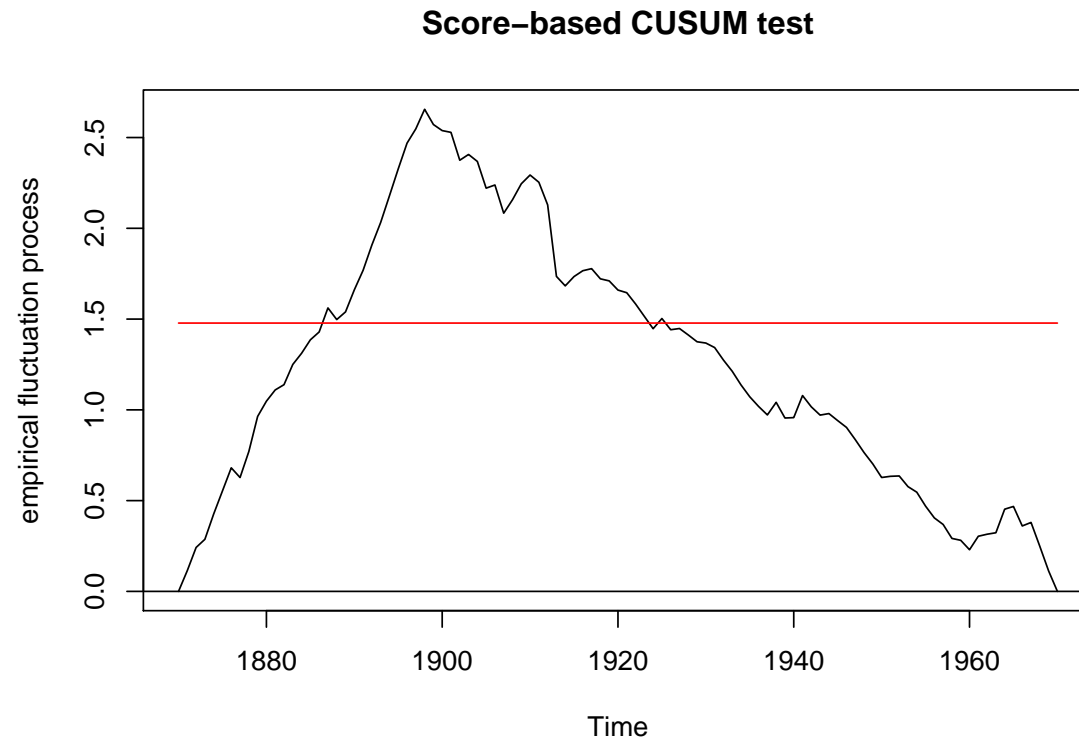


Nildaten

```
R> scus.nile <- efp(Nile ~ 1, type = "Score-CUSUM")  
R> plot(scus.nile)
```

Nildaten

```
R> scus.nile <- efp(Nile ~ 1, type = "Score-CUSUM")  
R> plot(scus.nile)
```

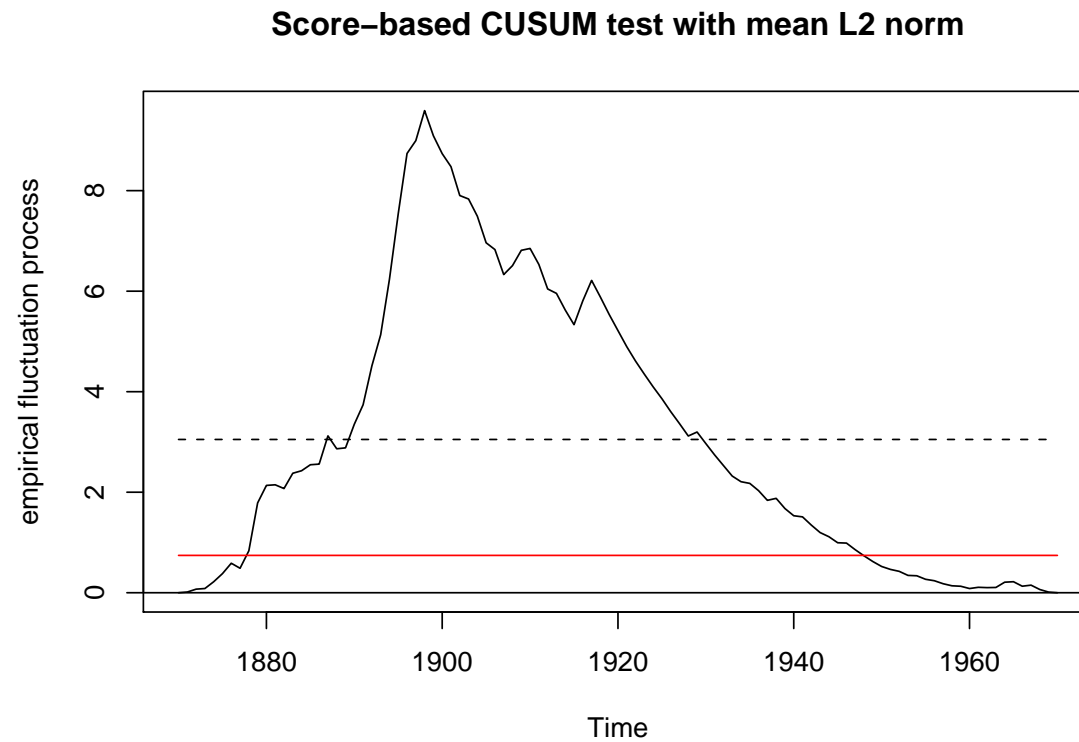


Nildaten

```
R> plot(scus.nile, functional = "meanL2")
```

Nildaten

```
R> plot(scus.nile, functional = "meanL2")
```

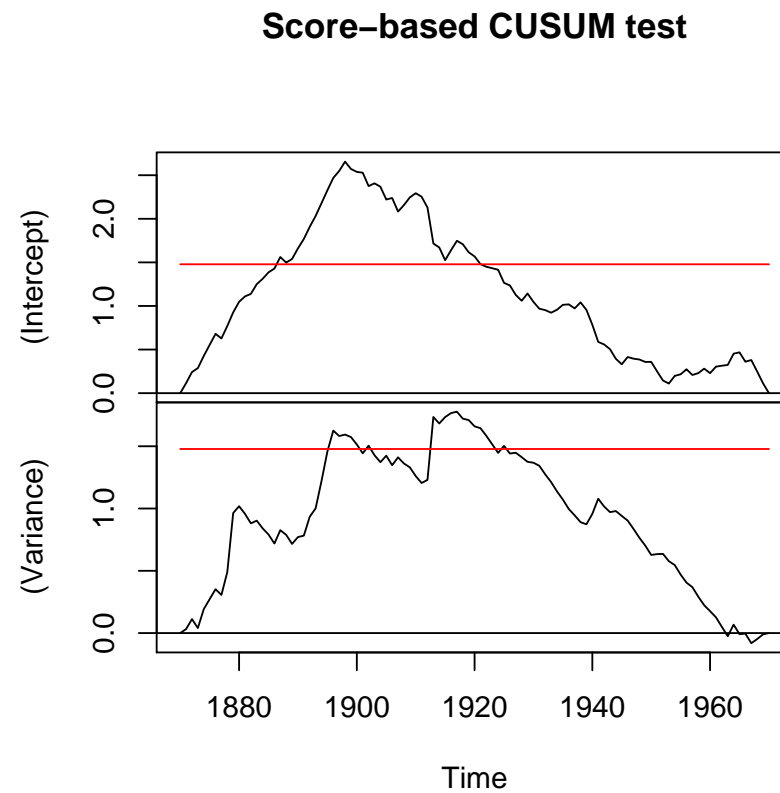


Nildaten

```
R> plot(scus.nile, functional = NULL)
```

Nildaten

```
R> plot(scus.nile, functional = NULL)
```



Nildaten

```
R> sctest(scus.nile)
```

```
Score-based CUSUM test
```

```
data: scus.nile
```

```
f(efp) = 2.6557, p-value = 2.992e-06
```

Nildaten

```
R> sctest(scus.nile)
```

```
Score-based CUSUM test
```

```
data: scus.nile  
f(efp) = 2.6557, p-value = 2.992e-06
```

```
R> sctest(scus.nile, functional = "meanL2")
```

```
Score-based CUSUM test with mean L2 norm
```

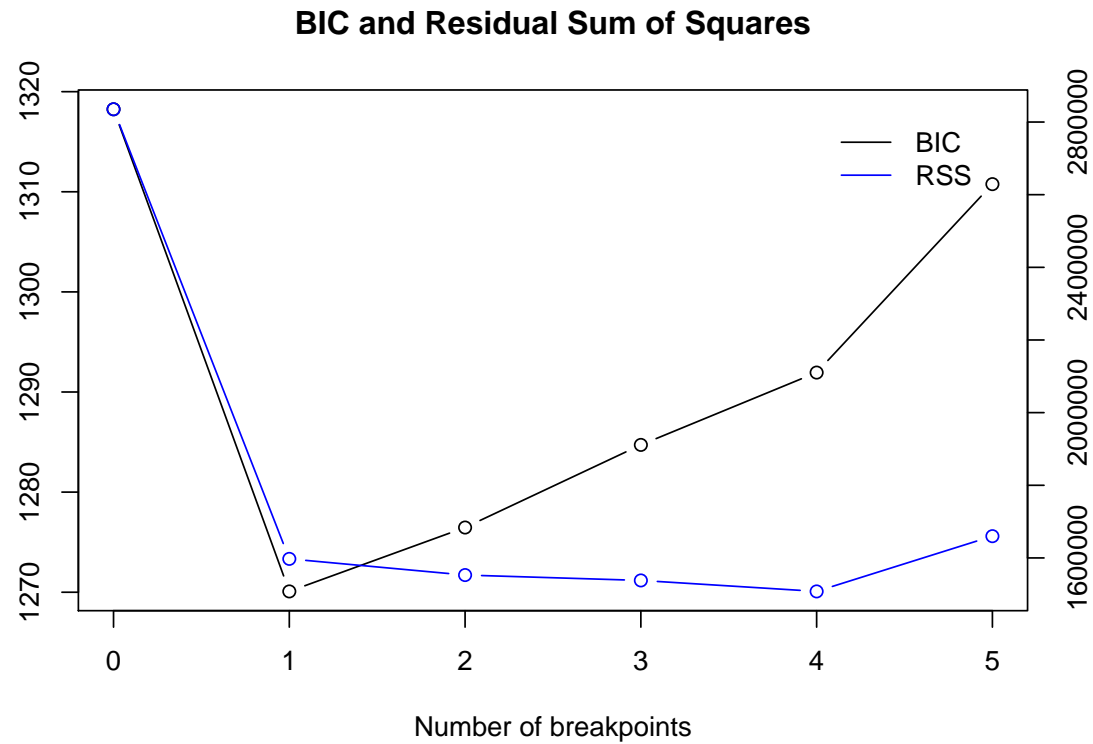
```
data: scus.nile  
f(efp) = 3.08, p-value = 0.005
```

Nildaten

```
R> bp.nile <- breakpoints(Nile ~ 1)
R> plot(bp.nile)
```

Nildaten

```
R> bp.nile <- breakpoints(Nile ~ 1)
R> plot(bp.nile)
```



Nildaten

```
R> summary(bp.nile, breaks = 3)
```

```
    Optimal (m+1)-segment partition:
```

```
Call:
```

```
breakpoints.formula(formula = Nile ~ 1)
```

```
Breakpoints at observation number:
```

```
m = 1  28
m = 2  28  83
m = 3  28 68 83
```

```
Corresponding to breakdates:
```

```
m = 1  1898
m = 2  1898  1953
m = 3  1898 1938 1953
```

```
Fit:
```

```
 m   0           1           2           3
RSS 2835156.750 1597457.194 1552923.616 1538096.513
BIC  1318.242   1270.084   1276.467   1284.718
```

Nildaten

```
R> confint(bp.nile)
```

```
Confidence intervals for breakpoints  
of optimal 2-segment partition:
```

```
Call:
```

```
confint.breakpointsfull(object = bp.nile)
```

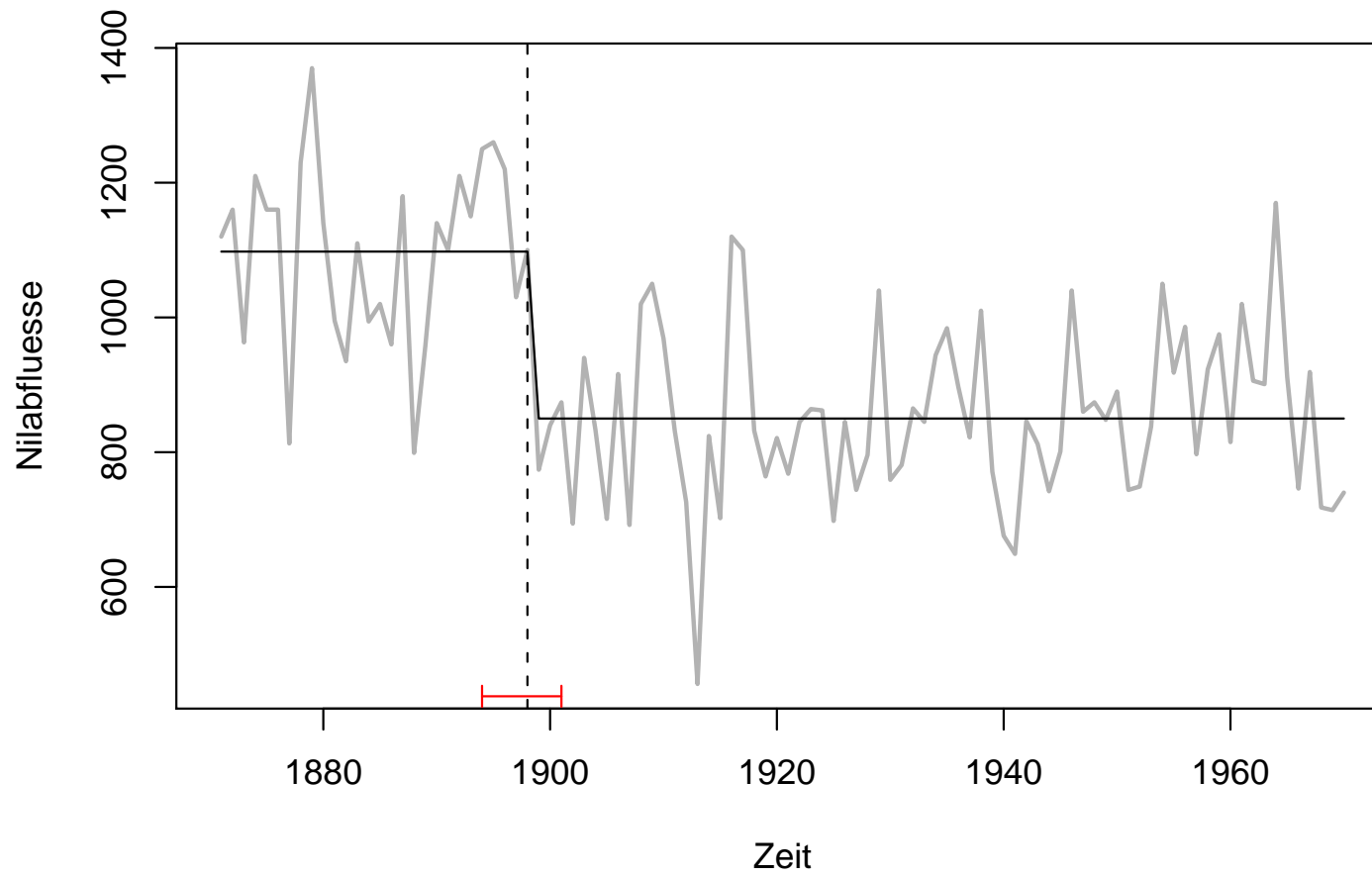
```
Breakpoints at observation number:
```

```
  2.5 % breakpoints 97.5 %  
1    24           28    31
```

```
Corresponding to breakdates:
```

```
  2.5 % breakpoints 97.5 %  
1 1894           1898 1901
```


Nildaten



Bruchpunktschätzung

Segmentiertes Regressionsmodell:

$$y_i = x_i^\top \beta_j + u_i \quad (i = i_{j-1} + 1, \dots, i_j),$$

wobei $j = 1, \dots, m + 1$ der Segmentindex ist.

m -Partition: die Menge der Bruchpunkte $\mathcal{I}_{m,n} = \{i_1, \dots, i_m\}$

Konvention: $i_0 = 0$ und $i_{m+1} = n$

Bruchpunktschätzung

Segmentiertes Regressionsmodell:

$$y_i = x_i^\top \beta_j + u_i \quad (i = i_{j-1} + 1, \dots, i_j),$$

wobei $j = 1, \dots, m + 1$ der Segmentindex ist.

m -Partition: die Menge der Bruchpunkte $\mathcal{I}_{m,n} = \{i_1, \dots, i_m\}$

Konvention: $i_0 = 0$ und $i_{m+1} = n$

Die Fehlerquadratsumme des segmentierten Modells ist:

$$RSS(i_1, \dots, i_m) = \sum_{j=1}^{m+1} r_{SS}(i_{j-1} + 1, i_j)$$

Bruchpunktschätzung

Dabei ist $rss(i_{j-1} + 1, i_j)$ die minimale Fehlerquadratsumme im j -ten Segment.

Gesucht:

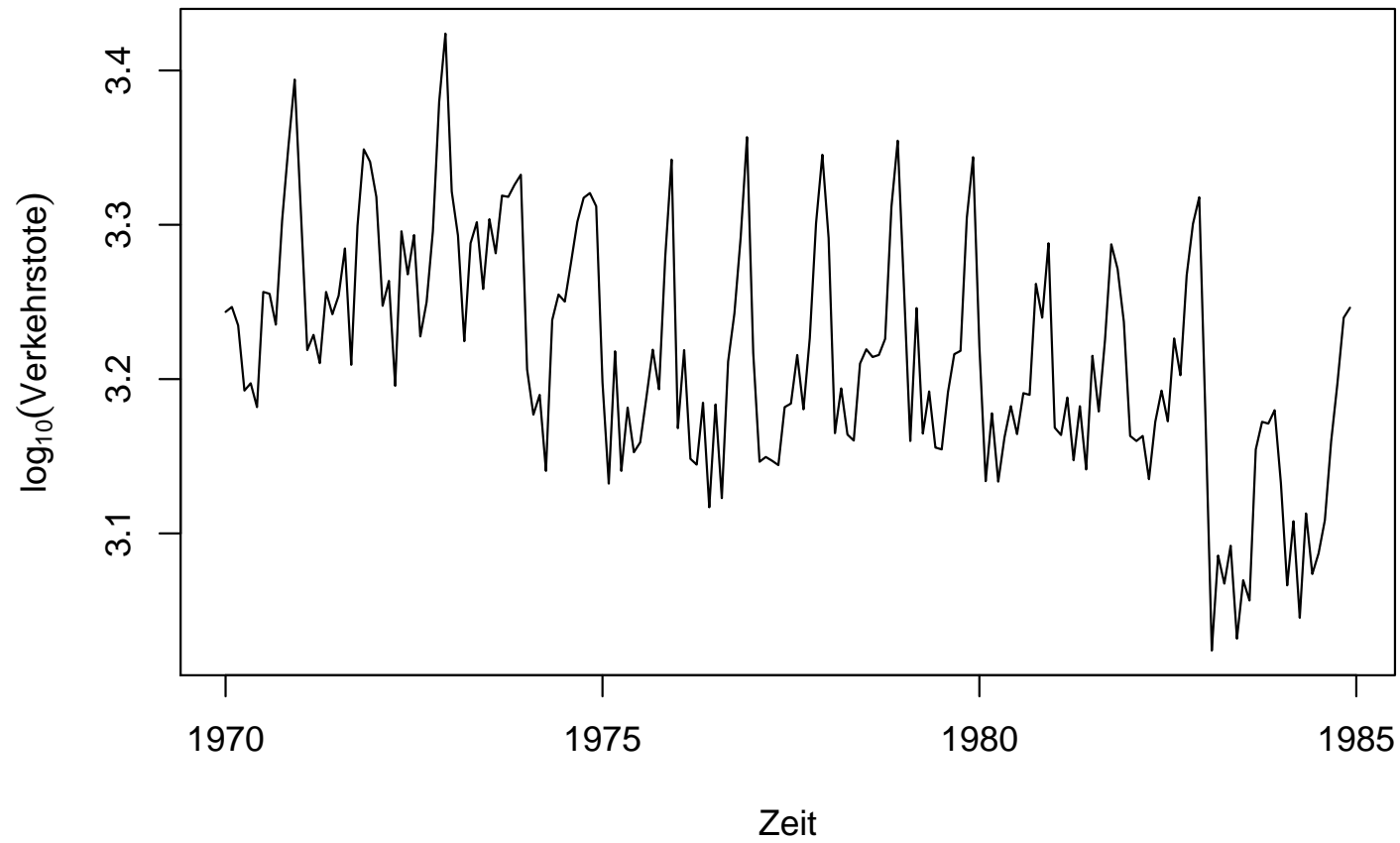
$$(\hat{i}_1, \dots, \hat{i}_m) = \operatorname{argmin}_{(i_1, \dots, i_m)} RSS(i_1, \dots, i_m)$$

über alle Partitionen (i_1, \dots, i_m) mit $i_j - i_{j-1} \geq \lfloor nh \rfloor \geq k$

Optimalitätsprinzip von Bellman:

$$RSS(\mathcal{I}_{m,n}) = \min_{mn_h \leq i \leq n - n_h} [RSS(\mathcal{I}_{m-1,i}) + rss(i + 1, n)]$$

Verkehrstote in Großbritannien

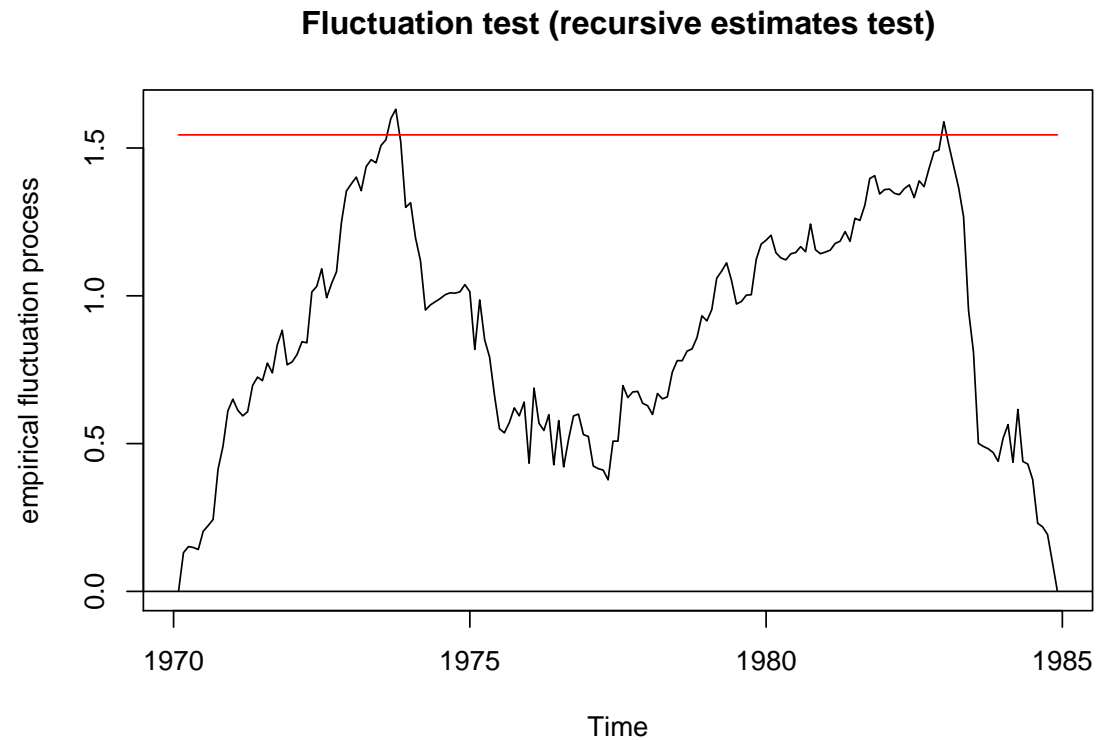


Verkehrstote in Großbritannien

```
R> re.seat <- efp(y ~ ylag1 + ylag12, data = seatbelt, type = "RE")  
R> plot(re.seat)
```

Verkehrstote in Großbritannien

```
R> re.seat <- efp(y ~ ylag1 + ylag12, data = seatbelt, type = "RE")  
R> plot(re.seat)
```

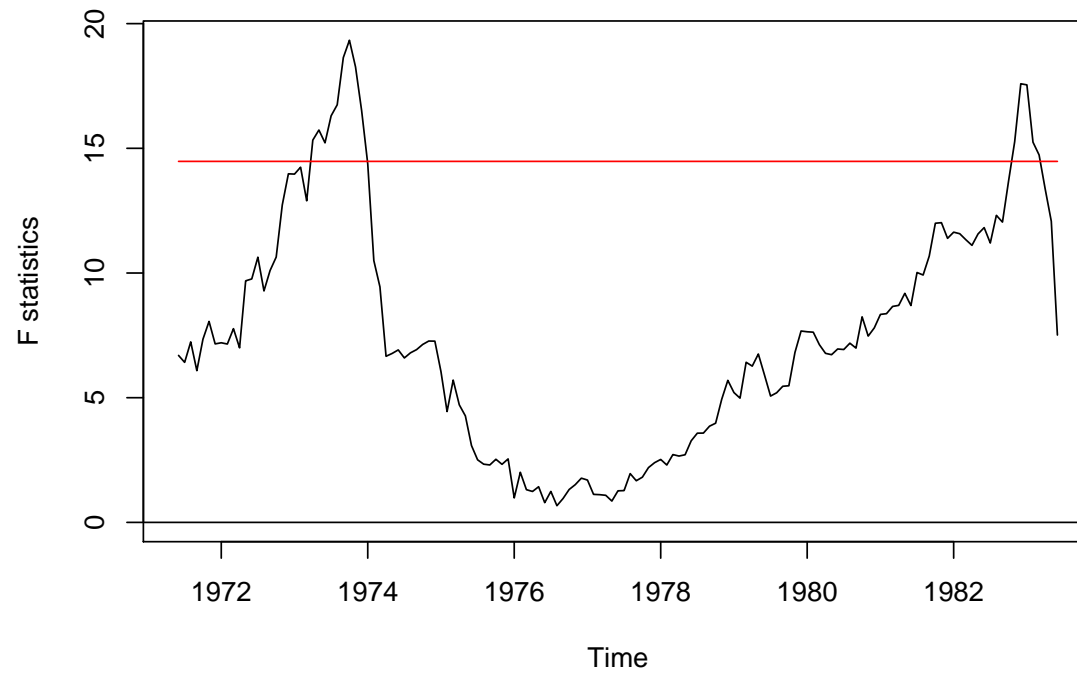


Verkehrstote in Großbritannien

```
R> fs.seat <- Fstats(y ~ ylag1 + ylag12, data = seatbelt, from = 0.1)
R> plot(fs.seat)
```


Verkehrstote in Großbritannien

```
R> fs.seat <- Fstats(y ~ ylag1 + ylag12, data = seatbelt, from = 0.1)
R> plot(fs.seat)
```

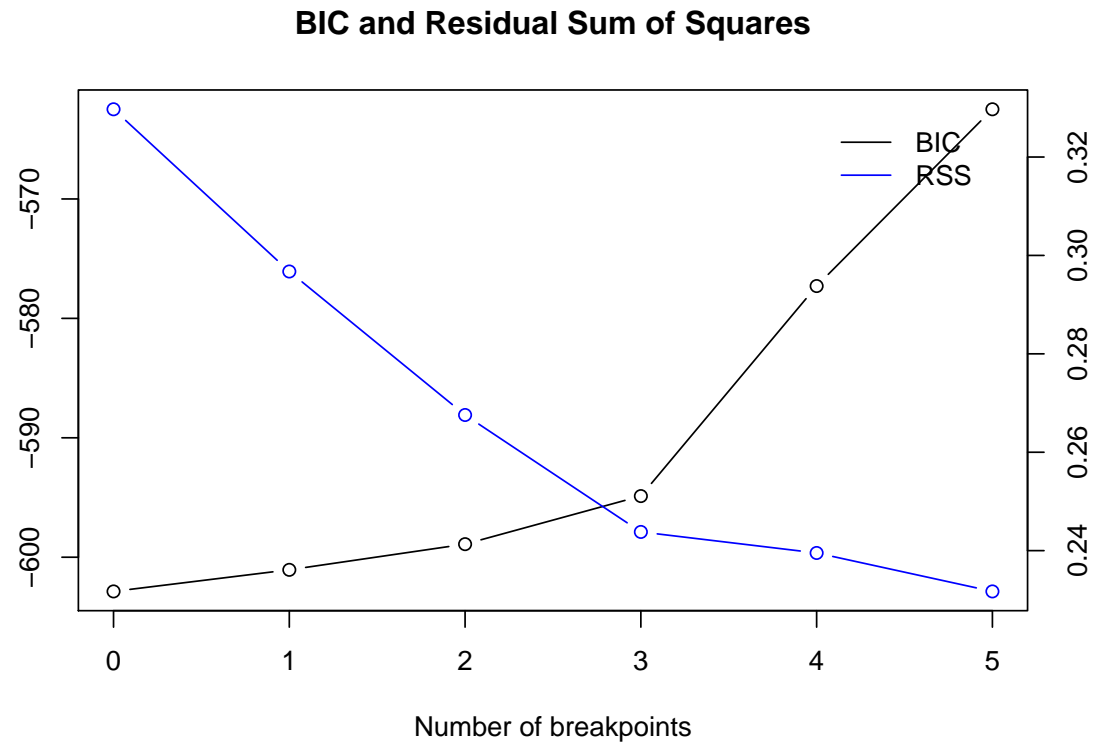


Verkehrstote in Großbritannien

```
R> bp.seat <- breakpoints(y ~ ylag1 + ylag12, data = seatbelt, h = 0.1)
R> plot(bp.seat, breaks = 5)
```

Verkehrstote in Großbritannien

```
R> bp.seat <- breakpoints(y ~ ylag1 + ylag12, data = seatbelt, h = 0.1)
R> plot(bp.seat, breaks = 5)
```



Verkehrstote in Großbritannien

