

# Onlinebeispiel 8.7. Wachstumsmodelle von Seehunden (*Phoca vitulina*) mittels multivariaten State-Space-Modellen

Kapitel 8.3.1 aus Henle, K., A. Grimm-Seyfarth & B. Gruber: Erfassung und Analyse von Tierpopulationen. Ulmer Verlag

Annegret Grimm-Seyfarth

22.04.2025

In diesem Beispiel zeigen wir multivariate State-Space-Modelle (MARSS) unter Verwendung des MARSS-Pakets (Holmes et al. 2012, 2024a,b). Diese Klasse von Zeitreihenmodellen wird auch als vektorautoregressive State-Space-Modelle (VARSS) bezeichnet. Hier zeigen wir ein Beispiel, bei dem die Modellauswahl verwendet wird, um verschiedene Populationsstrukturen bei Seehunden (*Phoca vitulina*) an der Westküste Nordamerikas zu testen. Dieses und viele weitere Beispiele können mithilfe von Holmes et al. (2021) nachgelesen und erstellt werden.

```
# check.packages function: install and load multiple R packages.
# Function from: https://gist.github.com/smithdanielle/9913897
check.packages <- function(pkg){
  new.pkg <- pkg[!(pkg %in% installed.packages()[, "Package"])]
  if (length(new.pkg))
    install.packages(new.pkg, dependencies = TRUE, type = "source")
  sapply(pkg, require, character.only = TRUE)
}

# benoetigte R pakete
pakete <- c("MARSS", "ggplot2", "scales")

# Pruefe und installiere
check.packages(pakete)
```

```
##   MARSS ggplot2  scales
##   TRUE   TRUE   TRUE
```

## Daten einladen

Die Daten sind im R-Paket MARSS als Beispieldatensatz integriert (Jeffries et al. 2003). Inzwischen wurde aber auch eine umfangreichere Version der Daten bis einschließlich 2023 publiziert (Pearson et al. 2024). Dort gibt es jedoch viele fehlende Datenpunkte, weshalb wir in diesem Beispiel den Datensatz bis 1999 verwenden.

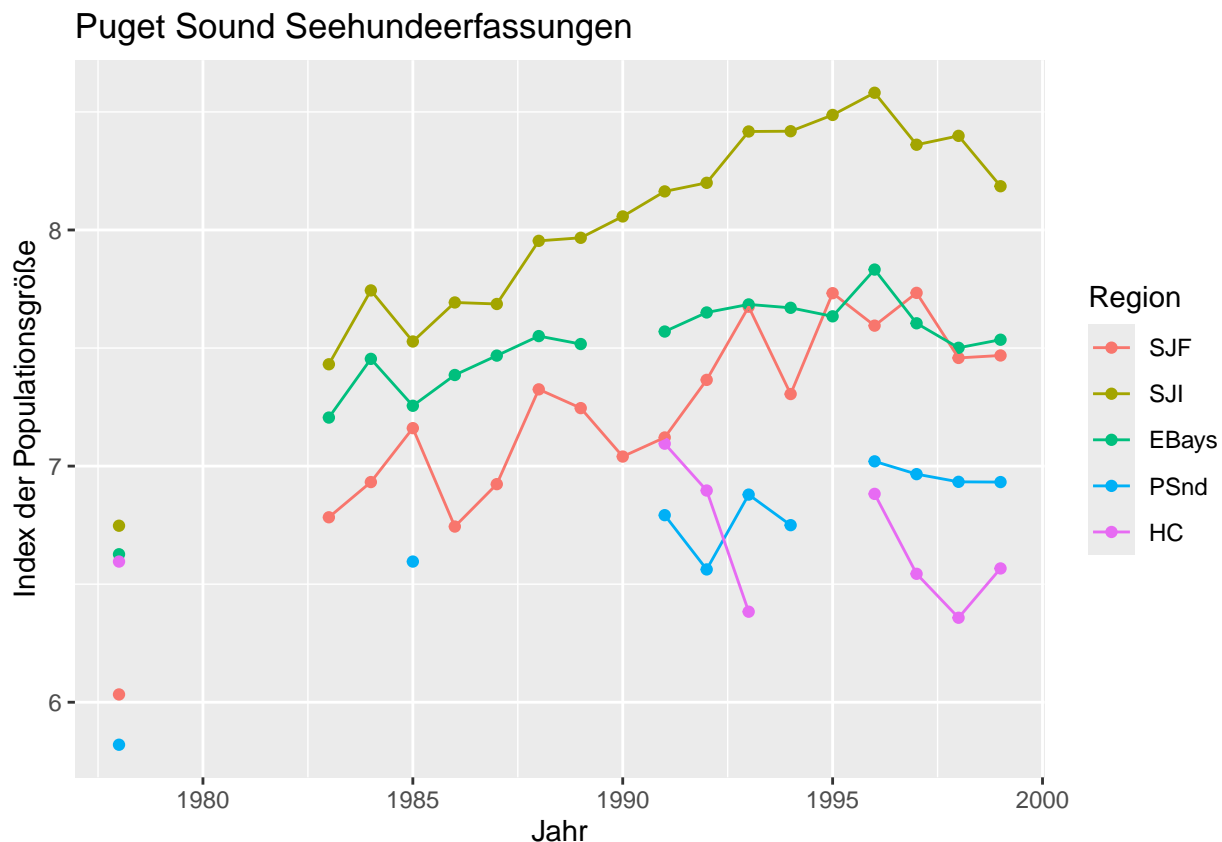
```
data(harborSealWA, package = "MARSS")
```

## Grafische Darstellung

Die Daten stammen aus fünf Regionen (oder Standorten). Dargestellt ist jeweils ein normierter Index für die Populationsgröße. Seehunde wurden von 1978 bis 1999 gezählt, während sie an Land waren. Nachdem sie vor deren Unterschutzstellung durch die Jagd auf ein niedriges Niveau reduziert worden waren, erholten sich die Seehunde während des Zeitraums dieses Datensatzes stetig. Wir gehen davon aus, dass der zugrundeliegende Populationsprozess ein stochastischer exponentieller Wachstumsprozess mit mittleren Wachstumsraten ist, die sich in den Jahren 1978-1999 nicht verändert haben.

Die Erhebungsmethoden waren in den 20 Jahren der Datenerfassung einheitlich, aber wir wissen weder, welchen Anteil der Population die einzelnen Regionen repräsentieren, noch kennen wir die Varianz der Beobachtungsfehler für jede Region. In Anbetracht der unterschiedlichen Anzahl von Erfassungen in den einzelnen Regionen können die Beobachtungsfehler sehr unterschiedlich sein. Die Regionen wurden unterschiedlich stark beprobt; in der am besten beprobten Region fehlen nur 4 Jahre, während in der schlechtesten mehr als die Hälfte der Jahre fehlen. Stellen wir diese Rohdaten einmal für die verschiedenen Regionen dar.

```
temp <- as.data.frame(MARSS::harborSealWA)
pdat <- reshape2::melt(temp, id.vars = "Year", variable.name = "region")
p <- ggplot(pdat, aes(x = Year, y = value, col = region)) +
  xlab("Jahr") +
  ylab("Index der Populationsgröße") +
  labs(color = "Region") +
  geom_point() +
  geom_line()
p + ggtitle("Puget Sound Seehundeerfassungen")
```



## Daten vorbereiten

Wir werden die Population am Hood Canal (HC) auslassen, da diese Region von den anderen etwas isoliert ist und aufgrund von hypoxischen Ereignissen und periodisch auftretenden intensiven Killerwal-Raubzügen ganz andere Bedingungen aufweist. Die Daten werden wie folgt aufbereitet:

```
dat <- MARSS::harborSealWA
years <- dat[, "Year"]
dat <- dat[, !(colnames(dat) %in% c("Year", "HC"))]
dat <- t(dat) # Daten transponieren, um Jahre als Spalten zu haben
colnames(dat) <- years
n <- nrow(dat) - 1
```

## Eine große Population

Wenn wir Daten über eine große geografische Region betrachten, könnten wir davon ausgehen, dass die verschiedenen Erhebungsregionen eine einzige Population messen, wenn wir glauben, dass sich die Tiere ausreichend bewegen, sodass das gesamte Gebiet (mehrere Regionen zusammen) „gut durchmischt“ ist. Außerdem gehen wir davon aus, dass es sich bei allen vier regionalen Zeitreihen um Beobachtungen dieser einen Population handelt, die jedoch relativ zu dieser Entwicklung nach oben oder unten skaliert sind. Folglich betrachten wir jede regionale Erhebung als einen Index der Gesamtpopulationsgröße. Bei diesem Modell gehen wir nicht davon aus, dass die Regionen unabhängige Teilpopulationen darstellen, sondern vielmehr unabhängige Beobachtungen einer Population.

In MARSS wird folgendes Modell angepasst:

$$x_t = B_t \cdot x_{t-1} + u + w_t \text{ mit } w_t \sim N(0, Q),$$

$$y_t = Z_t \cdot x_t + a + v_t \text{ mit } v_t \sim N(0, R),$$

$$x_0 = \mu.$$

Dabei ist:

$x_t$  eine  $m \times T$ -Matrix (**states**), jedes  $x_t$  ist eine Realisierung der Zufallsvariablen  $X_t$  zum Zeitpunkt  $t$ ;

$w_t$  eine  $m \times T$ -Matrix der Prozessfehler; Die Prozessfehler zum Zeitpunkt  $t$  sind multivariat normalverteilt mit Mittelwert 0 und Kovarianzmatrix  $Q_t$ ;

$y_T$  eine  $n \times T$ -Matrix der Beobachtungen (einige Beobachtungen können fehlen);

$v_t$  ein  $n \times T$ -Spaltenvektor der prozessunabhängigen Fehler; die Beobachtungsfehler zum Zeitpunkt  $t$  sind multivariat normalverteilt mit Mittelwert 0 und Kovarianzmatrix  $R_t$ ;

$B_t$  und  $Z_t$  sind Parameter und  $m \times m$  bzw.  $n \times m$  Matrizen;

$u$  und  $a$  sind Parameter und  $m \times 1$  bzw.  $n \times 1$  Spaltenvektoren;

$Q$  und  $R$  sind Parameter und  $g \times g$  (typischerweise  $m \times m$ ) bzw.  $h \times h$  (typischerweise  $n \times n$ ) Varianz-Kovarianz-Matrizen.

Zur Anpassung unseres Modells mit MARSS() erstellen wir eine Liste, die die Größe und Struktur jeder Parametermatrix genau beschreibt. Feste Werte in einer Matrix werden mit ihrem numerischen Wert bezeichnet, und geschätzte Werte werden mit einem Namen versehen und in Anführungszeichen gesetzt. Unsere Modellliste für eine einzelne gut gemischte Population lautet:

```
mod.list.0 <- list(
  B = matrix(1),
  U = matrix("u"),
```

```

Q = matrix("q"),
Z = matrix(1, 4, 1),
A = "scaling",
R = "diagonal and unequal",
x0 = matrix("mu"),
tinitx = 0
)

```

Hierbei bedeutet die Abkürzung `diagonal and unequal` eine Diagonalmatrix, bei der jedes Diagonalelement einen anderen Wert hat. Die Abkürzung `scaling` bezeichnet die Form von  $a$  (in der Gleichung  $y_t = Z_t \cdot x_t + a + v_t$ ), wobei ein Wert auf 0 gesetzt und der Rest geschätzt wird. Wir sollten den Code in der Liste ausführen, um sicherzustellen, dass jeder Parameter in der Liste die gleiche Form hat wie in unserer mathematischen Gleichung für das Modell.

Dann passen wir das Modell an:

```

fit.0 <- MARSS(dat, model = mod.list.0)

## Success! abstol and log-log tests passed at 32 iterations.
## Alert: conv.test.slope.tol is 0.5.
## Test with smaller values (<0.1) to ensure convergence.
##
## MARSS fit is
## Estimation method: kem
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001
## Estimation converged in 32 iterations.
## Log-likelihood: 21.62931
## AIC: -23.25863   AICc: -19.02786
##
##              Estimate
## A.SJI          0.79583
## A.EBays         0.27528
## A.PSnd         -0.54335
## R.(SJF,SJF)     0.02883
## R.(SJI,SJI)     0.03063
## R.(EBays,EBays) 0.01661
## R.(PSnd,PSnd)   0.01168
## U.u            0.05537
## Q.q            0.00642
## x0.mu           6.22810
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.

```

## Residuen

Zur Überprüfung des Modells schauen wir uns die Residuen an:

```

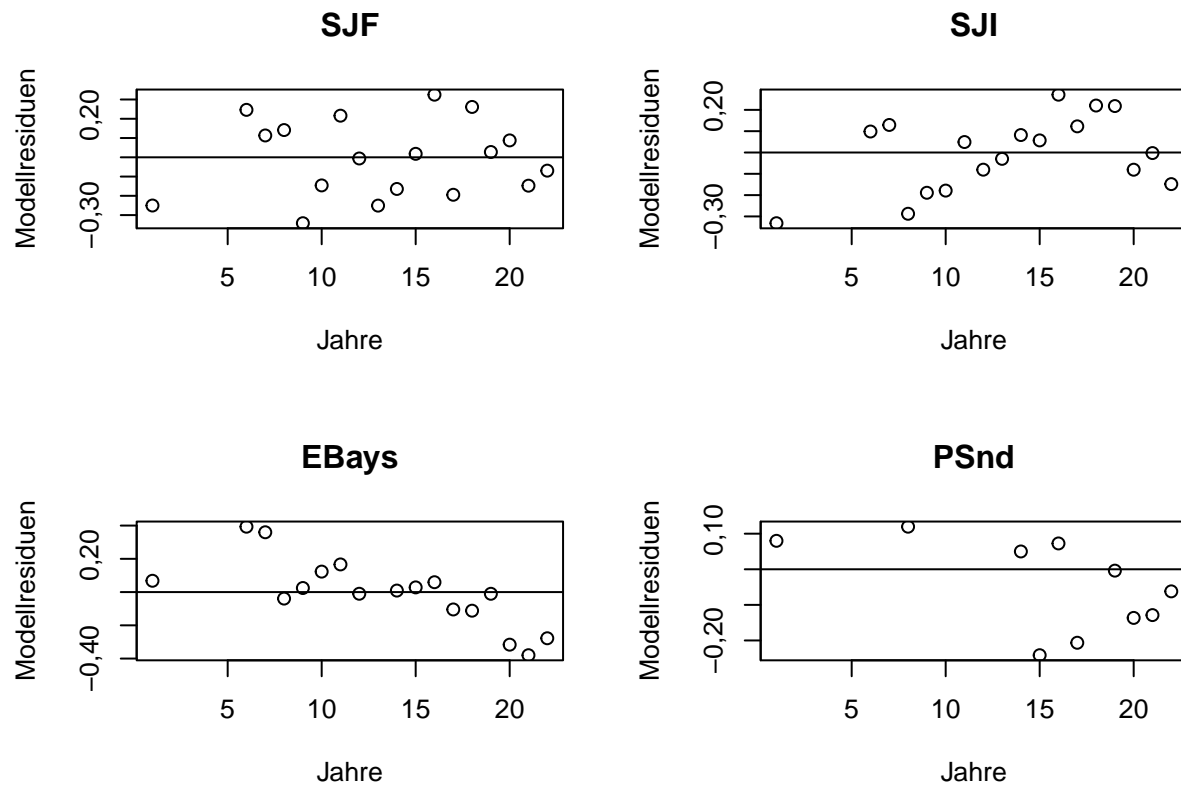
par(mfrow = c(2, 2))
resids <- MARSSresiduals(fit.0, type="tt1")
for (i in 1:4) {

```

```

plot(resids$model.residuals[i, ], yaxt = "n",
     ylab = "Modellresiduen", xlab = "Jahre")
abline(h = 0)
title(rownames(dat)[i])
axis(
  side = 2,
  at = pretty(resids$model.residuals[i, ]),
  labels = format(pretty(resids$model.residuals[i, ]),
                  decimal.mark = ",", nsmall = 2)
)
}

```



Auf der x-Achse sind die Jahre aufgetragen, auf der y-Achse die Residuen. Das Modell sieht nicht schlecht aus, aber die Residuen für SJI und EBays sind problematisch, da sie eine bestimmte Form (hier: Korrelation mit den Jahren) aufweisen.

## Vier Einzelpopulationen ohne zeitliche Autokorrelation

Das Modell für eine gut gemischte Population war laut Modellvalidierung mittels Residuen nicht sehr gut. Eine weitere vernünftige Annahme wäre, dass die verschiedenen Regionen vier verschiedene, zeitlich unabhängige Teilpopulationen darstellen.

Die  $Q$ -Matrix ist diagonal mit einem Varianzwert. Das bedeutet, dass die Prozessvarianz (Varianz der jährlichen Wachstumsraten) unabhängig ist (gute und schlechte Jahre sind nicht korreliert), aber der Grad der Variabilität ist in allen Regionen gleich. Wir erstellen die  $u$ -Matrix mit einem  $u$  Wert (`equal`). Dies

bedeutet, dass wir davon ausgehen, dass die Wachstumsraten in allen Regionen gleich sind. Wir setzen die  $B$ -Matrix gleich einer Diagonalmatrix mit 1 auf der Diagonalen (d. h., keine Dichteabhängigkeit). Dies ist die Identitäts-Matrix (`identity`) und entspricht einer 1, allerdings für Matrizen. Wir brauchen  $B$  nicht für unser Modell, aber `MARSS()` benötigt einen Wert. Hinsichtlich der Erfassungen setzen wir in diesem Modell jede Erfassung ( $x$ ) verschieden (`unequal`). Die Modellliste lautet dann:

```
mod.list.1 <- list(
  B = "identity",
  U = "equal",
  Q = "diagonal and equal",
  Z = "identity",
  A = "scaling",
  R = "diagonal and unequal",
  x0 = "unequal",
  tinitx = 0
)
```

Die neuen Abkürzungen bedeuten das Folgende: `identity` entspricht dem Wert 1 für Matrizen; `equal` bedeutet, dass alle Werte in der Matrix gleich sind; `diagonal and equal` bedeutet, dass die Matrix diagonal ist und ein Wert auf der Diagonale liegt; `unequal` bedeutet, dass alle Werte in der Matrix unterschiedlich sind.

Modell anpassen:

```
fit.1 <- MARSS(dat, model = mod.list.1)
```

```
## Success! abstol and log-log tests passed at 24 iterations.
## Alert: conv.test.slope.tol is 0.5.
## Test with smaller values (<0.1) to ensure convergence.
##
## MARSS fit is
## Estimation method: kem
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001
## Estimation converged in 24 iterations.
## Log-likelihood: 23.21635
## AIC: -26.43271   AICc: -22.20194
##
##               Estimate
## R.(SJF,SJF)      0.02912
## R.(SJI,SJI)      0.00638
## R.(EBays,EBays)  0.00365
## R.(PSnd,PSnd)    0.00974
## U.1              0.05837
## Q.diag           0.00966
## x0.X.SJF         6.13673
## x0.X.SJI         6.73763
## x0.X.EBays       6.58977
## x0.X.PSnd        5.80029
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.
```

## Vier Einzelpopulationen mit zeitlicher Autokorrelation

Eine weitere vernünftige Annahme wäre, dass die verschiedenen Regionen unterschiedliche Teilpopulationen darstellen, dass aber die jährlichen Wachstumsraten korreliert sind (gutes und schlechtes Jahr fallen zusammen). Der einzige Parameter, der sich ändert, ist die  $Q$ -Matrix. Dies passen wir wie folgt an:

```
mod.list.2 <- mod.list.1
mod.list.2$Q <- "equalvarcov"
```

Die Abkürzung `equalvarcov` modelliert zeitliche Autokorrelation.

Wir passen das Modell an:

```
fit.2 <- MARSS(dat, model = mod.list.2)

## Success! abstol and log-log tests passed at 31 iterations.
## Alert: conv.test.slope.tol is 0.5.
## Test with smaller values (<0.1) to ensure convergence.
##
## MARSS fit is
## Estimation method: kem
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001
## Estimation converged in 31 iterations.
## Log-likelihood: 34.09079
## AIC: -46.18158   AICc: -41.00511
##
##              Estimate
## R.(SJF,SJF)    0.02439
## R.(SJI,SJI)    0.00282
## R.(EBays,EBays) 0.00186
## R.(PSnd,PSnd)   0.00797
## U.1            0.05736
## Q.diag         0.01306
## Q.offdiag      0.01049
## x0.X.SJF       6.05525
## x0.X.SJI       6.70630
## x0.X.EBays     6.56580
## x0.X.PSnd      5.75608
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.
```

## Modellvergleiche

Wie bei anderen Modellen auch, können diese Modelle über AICc verglichen werden (siehe Kapitel 9.3 im Buch).

```
c(fit.0$AICc, fit.1$AICc, fit.2$AICc)
```

```
## [1] -19.02786 -22.20194 -41.00511
```

Wir sehen, dass das letzte Modell deutlich das beste ist und den geringsten AICc aufweist.

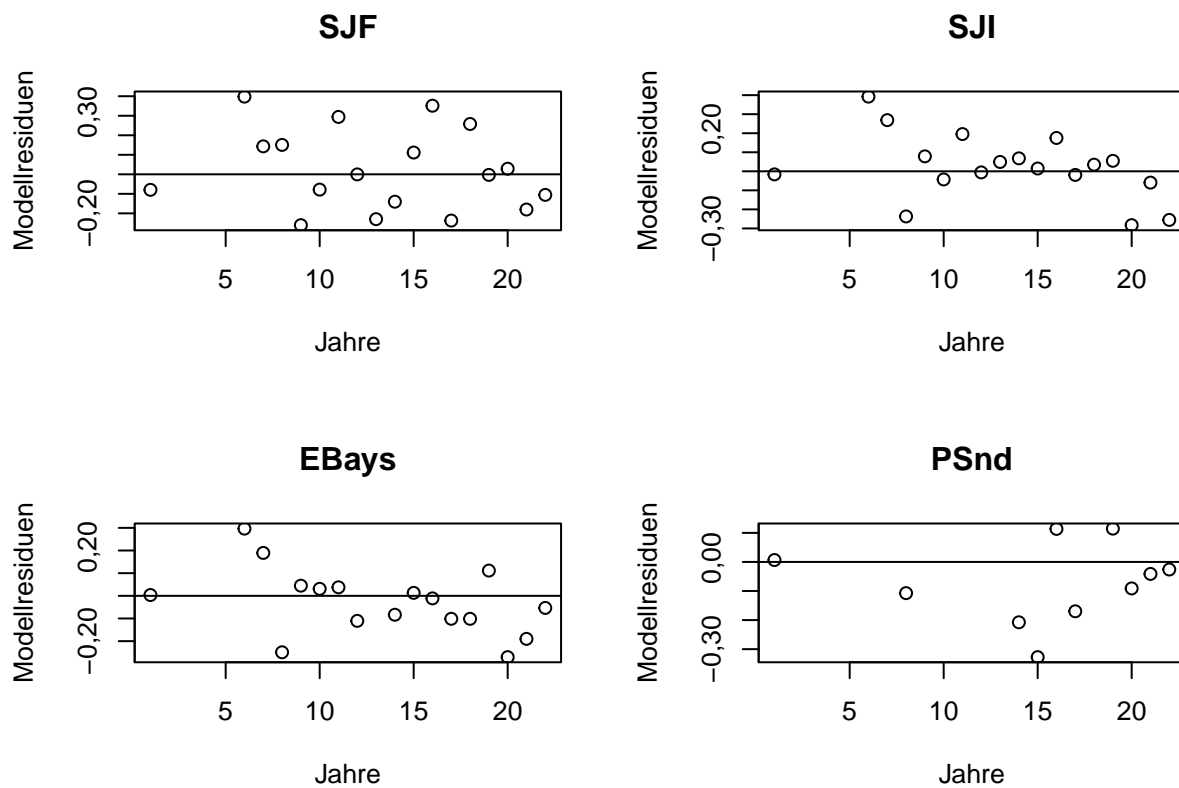
## Residuen

Schauen wir uns nun die Residuen des letzten Modells an, um die Modellanpassung zu validieren:

```
par(mfrow = c(2, 2))
resids <- MARSSresiduals(fit.2, type="tt1")
```

```
## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.
```

```
for (i in 1:4) {
  plot(resids$model.residuals[i, ], yaxt = "n",
       ylab = "Modellresiduen", xlab = "Jahre")
  abline(h = 0)
  title(rownames(dat)[i])
  axis(
    side = 2,
    at = pretty(resids$model.residuals[i, ]),
    labels = format(pretty(resids$model.residuals[i, ]),
                    decimal.mark = ",", nsmall = 2)
  )
}
```



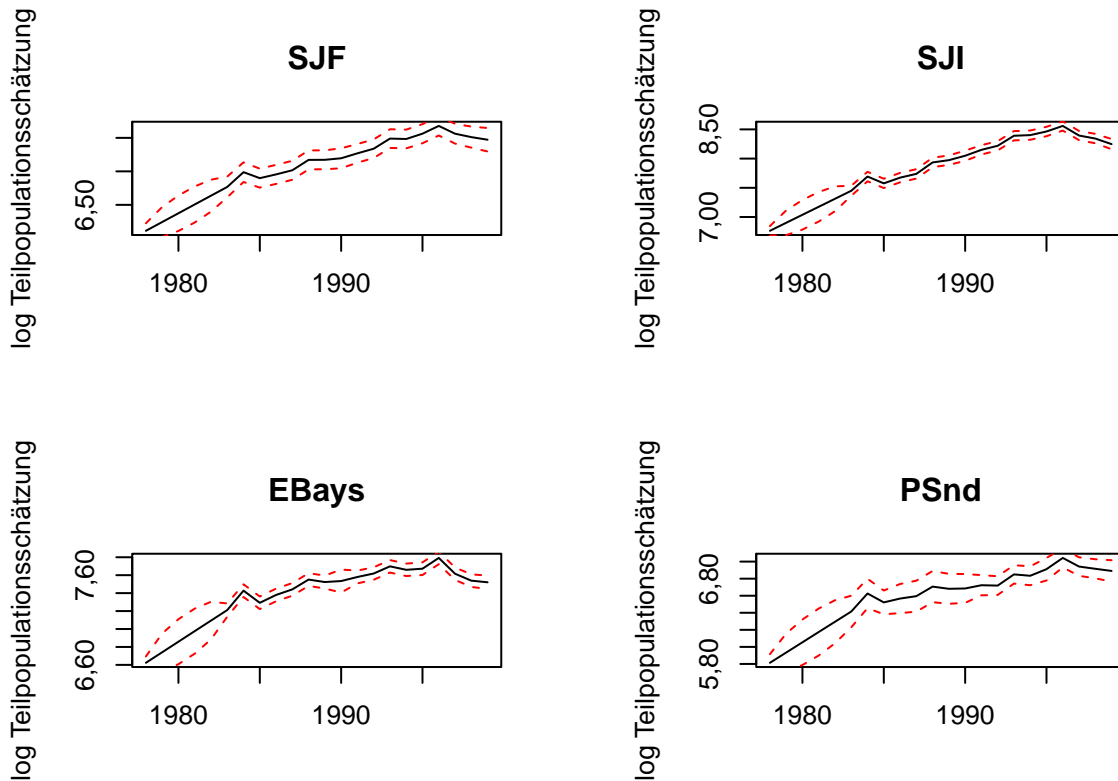
Die Residuen sehen wesentlich besser aus.



## Grafische Darstellung

Nun wollen wir die einzelnen Wachstumsraten der vier Regionen noch grafisch darstellen:

```
par(mfrow = c(2, 2), mar=c(6,6,4,2))
for (i in 1:4) {
  plot(years, fit.2$states[i, ], yaxt = "n",
       ylab = "log Teilpopulationsschätzung",
       xlab = "", type = "l",
       #cex=2, cex.lab=2, cex.axis=2
  )
  lines(years, fit.2$states[i, ] - 1.96 * fit.2$states.se[i, ],
        type = "l", lwd = 1, lty = 2, col = "red")
  lines(years, fit.2$states[i, ] + 1.96 * fit.2$states.se[i, ],
        type = "l", lwd = 1, lty = 2, col = "red")
  title(rownames(dat)[i], cex.main = 1.2)
  axis(
    side = 2, #cex.axis=2,
    at = pretty(fit.2$states[i, ]),
    labels = format(pretty(fit.2$states[i, ]),
                    decimal.mark = ",", nsmall = 2)
  )
}
```



Die schwarze Linie zeigt das Wachstum der Populationen, die roten gestrichelten Linien die 95 %-Konfidenzintervalle.

Dieses Modell kann auch um räumliche Strukturen erweitert werden. Ideen dafür finden sich hier: <https://atsa-es.github.io/atsa-labs/sec-mss-using-marss-models-to-study-spatial-structure.html>

Das Benutzerhandbuch von Holmes et al. (2021) beinhaltet auch Anpassungen über Baysische Modelle und Bootstrapping mittels JAGS und stan. Bei weiterführenden Analysen ist diese Literatur sehr empfehlenswert.

## Literaturverzeichnis

- Holmes, E.E., Ward, E.J., Wills, K. (2012): MARSS: Multivariate autoregressive state-space models for analyzing time-series data. — *The R Journal* 4(1): 11–19.
- Holmes, E.E., Scheuerell, M.D., Ward, E.J. (2021): Applied time series analysis for fisheries and environmental data. Edition 2021. — University of Washington, <https://atsa-es.github.io/atsa-labs/>
- Holmes, E.E., Ward, E.J., Scheuerell, M.D., Wills, K. (2024a): MARSS: Multivariate autoregressive state-space modeling. R package version 3.11.9.
- Holmes, E.E., Scheuerell, M.D., Ward, E.J. (2024b): Analysis of multivariate time-series using the MARSS package. Version 3.11.9. — NOAA Fisheries, Northwest Fisheries Science Center, Seattle, Washington.
- Jeffries, S., Huber, H., Calambokidis, J., Laake, J. (2003): Trends and status of harbor seals in Washington State: 1978–1999. — *Journal of Wildlife Management* 67(1): 207–218.
- Pearson, S.F., Amburgey, S.M., Clark, C.T., Tanedo, S.A., Londin, J.M., Huber, H.R., Jeffries, S.J. (2024): Trends and status of harbor seals in Washington State, USA (1977–2023). — *Marine Mammals Science* 41: e13161.
- Wickham, H. (2016): *ggplot2: Elegant graphics for data analysis*. — Springer-Verlag New York.