Reparameterizing a Time Series in terms of levels

Carl Schmertmann

04-Apr-2022

Main Point

It's convenient to have model parameters that are easy to interpret. In the case of a time series, the *levels* that the variable takes are often easier to understand than the mathematical constants underlying the series.

Here I show an example, in which we re-write a spline model

$$\theta_t = \sum_k \alpha_k \, s_{kt} \quad , \quad t = 1 \ldots T$$

so that its parameters are the levels of theta at the beginning, middle, and end of the time series.

Example: Natural Spline Time Series

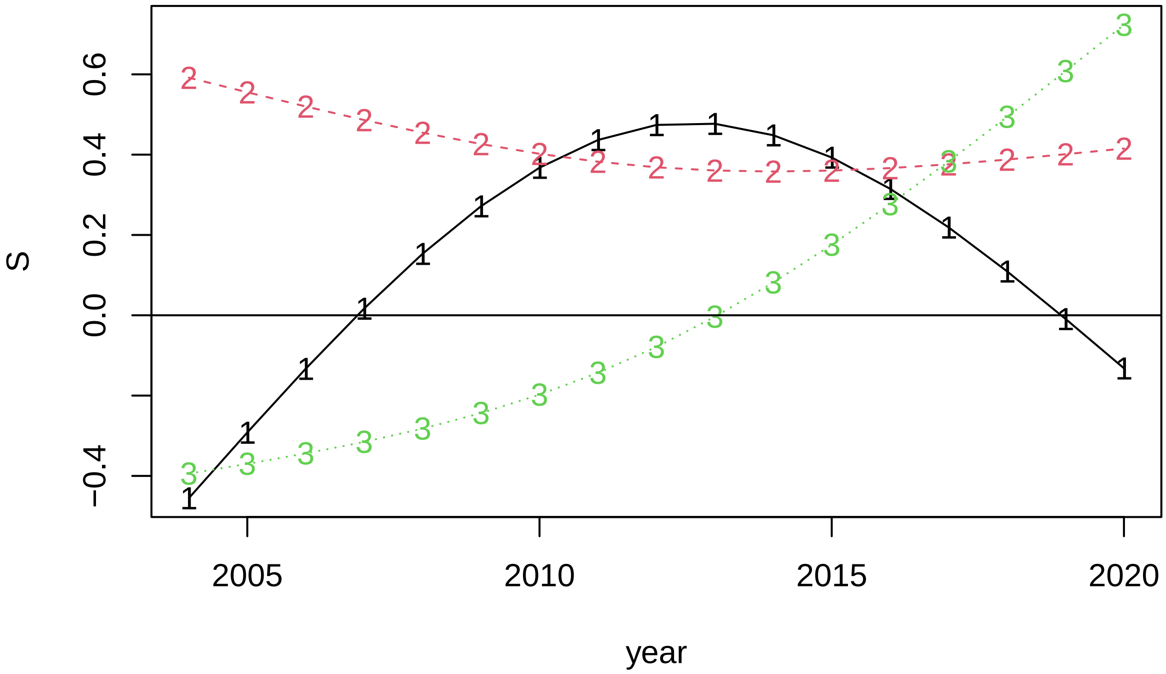
As a specific example, suppose that we are modeling a time series over years 2004–2020, so that the time index runs over $t = 1 \ldots 17$. Suppose also that we have decided to use a natural spline with $K = 3$ degrees of freedom, and an intercept.

In this case the model is

$$\theta = S\alpha$$

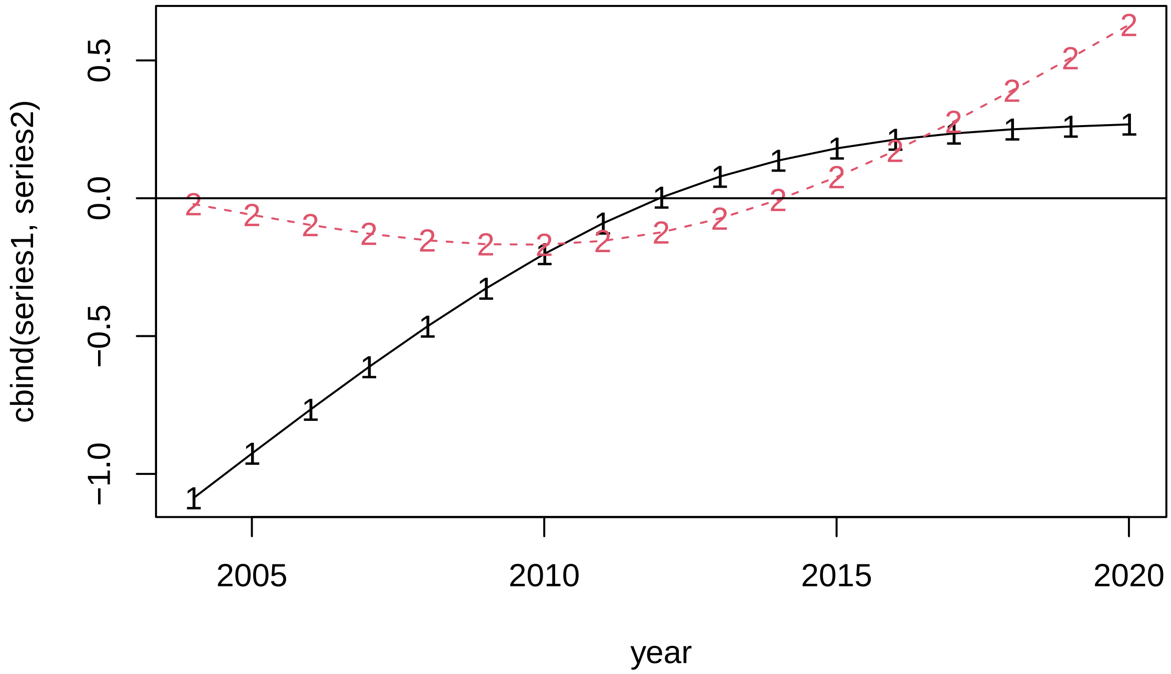
and a time series is a weighted sum of the columns of S, with parameters $(\alpha_1, \alpha_2, \alpha_3)$ as weights.

```r
year = 2004:2020

S = splines::ns( x=seq(year), df=3, intercept=TRUE)

matplot(year, S, type='o')
abline(h=0)
```



For example, if $\alpha = (0.7, -0.7, 0.9)$ or $(-0.3, 0.2, 0.7)$ then the time series would look like

```
series1 = S %*% c(+.70, -.70, +.90)
series2 = S %*% c(-.30, +.20, +.70)
matplot(year, cbind(series1, series2), type='o')
abline(h=0)
```



This is a nice, flexible way of modeling a time series. But the parameters don't really mean anything. With a little matrix algebra, it's easy to reparameterize the same set of possible time series in terms of the value of the series at specified times.

Suppose we want to have the parameters be the levels of the series in 2004, 2012, and 2020 ($t = 1, 9, 17$). Let's call those levels τ (for *t*argets) and write our objective as

$$\begin{bmatrix} e_1' \\ e_9' \\ e_{17}' \end{bmatrix} S\alpha = \begin{bmatrix} \tau_1 \\ \tau_9 \\ \tau_{17} \end{bmatrix}$$

where e_t' is a vector with a 1 in the tth position and 0s everywhere else (i.e., the tth row of a $T \times T$ identity matrix).

$$\underbrace{E}_{3\times17}\underbrace{S}_{17\times3}\underbrace{\alpha}_{3\times1} = \underbrace{\tau}_{3\times1}$$

The first question to answer is "Given τ, how do we find the α that generates it?" That is

$$\alpha = (ES)^{-1}\tau$$

Then the reparameterized time series is

$$\theta = S\alpha = [S(ES)^{-1}]\tau = S^*\tau$$

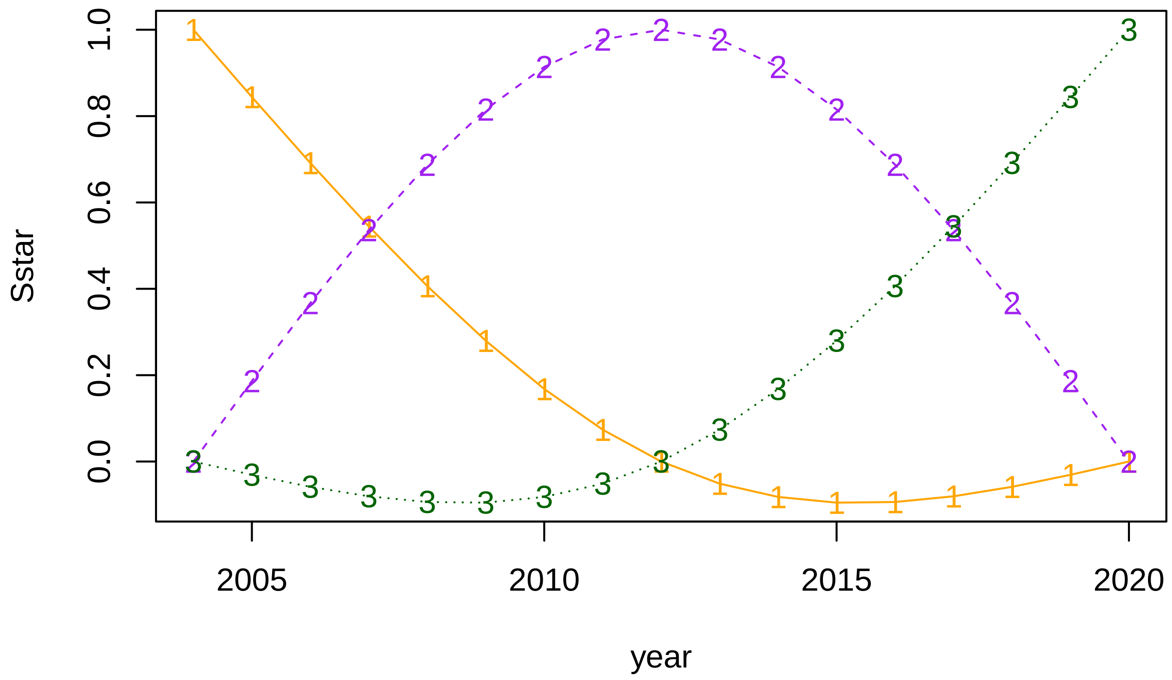
with a new basis $S^* = S(ES)^{-1}$.

In our example this is

```
E     = diag(17)[ c(1,9,17), ]
Sstar = S %*% solve(E %*% S)
```

which has new columns that look like

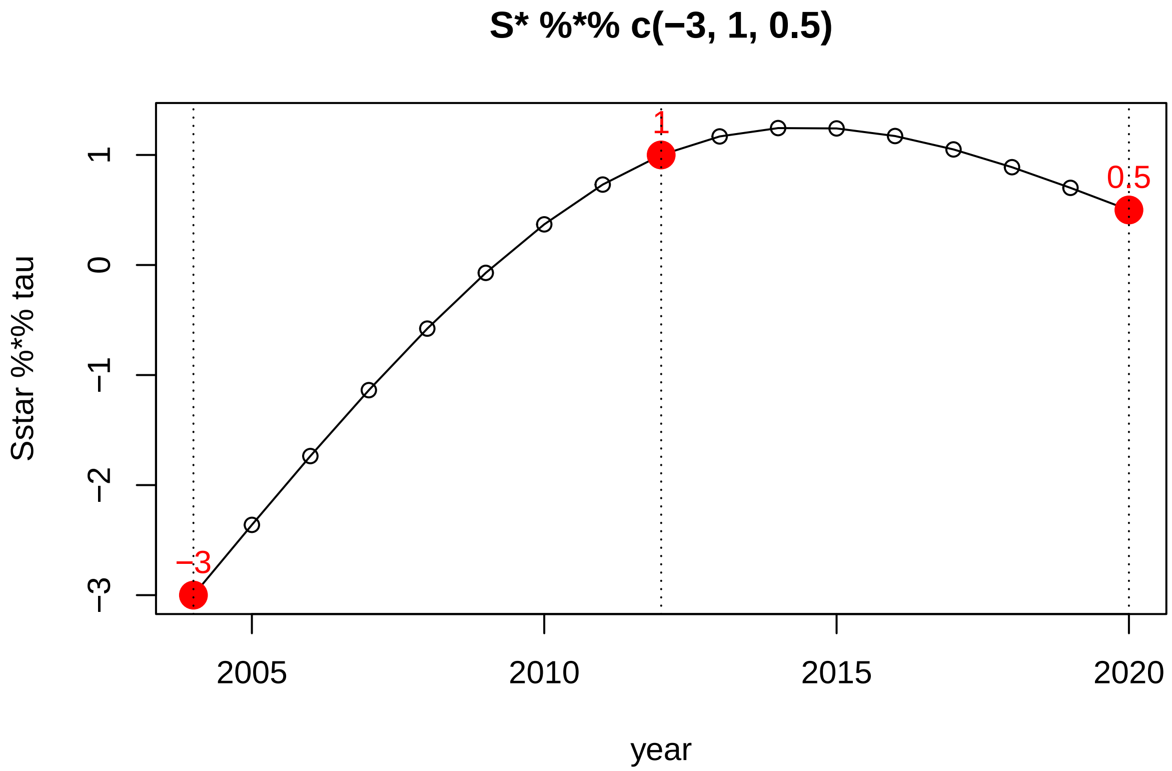
```
matplot(year, Sstar, type='o', col=c('orange','purple','darkgreen'))
```



# Using the reparameterized system

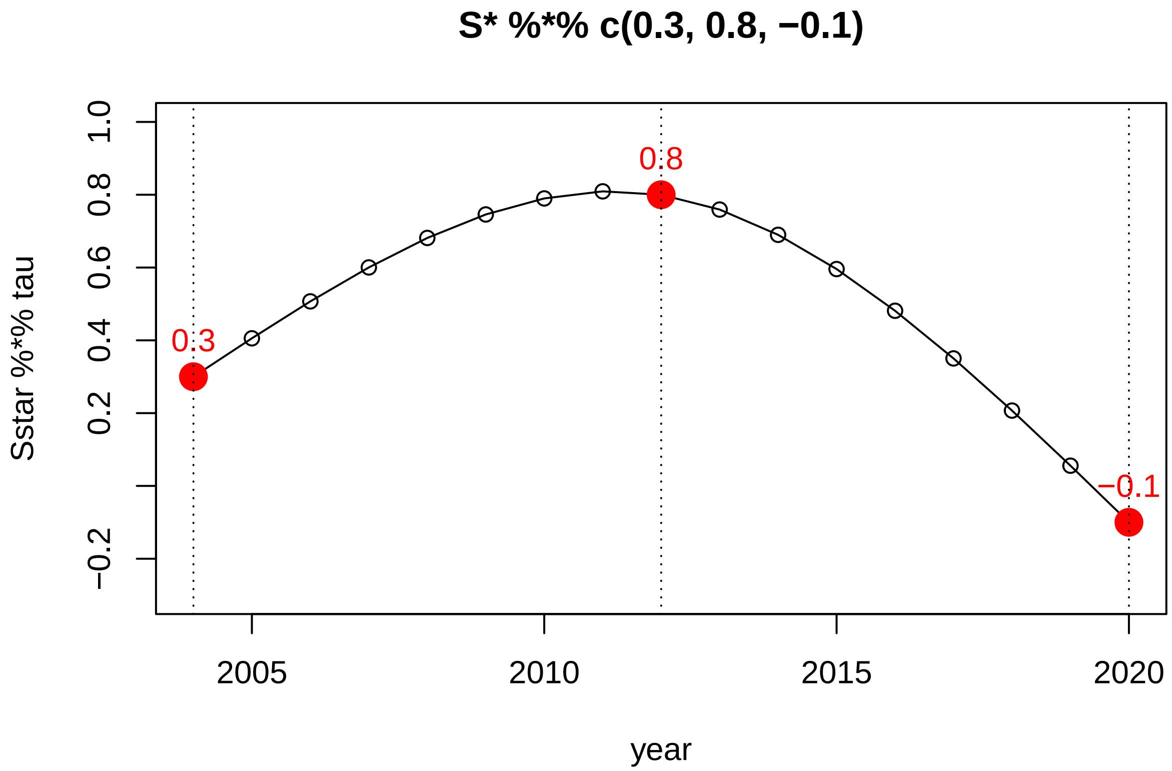
Now the parameters of the spline equal the θ values at specified times. For example, if we want a series that has (start, middle, end) values of $(-3, 1, 0)$ we use

```
tau = c( -3, 1, 0.5)

plot(year, Sstar %*% tau, type='o',ylim=c(-3,1.3),
     main='S* %*% c(-3, 1, 0.5)')
points(c(2004,2012,2020), tau, pch=16, cex=2, col='red',)
text( c(2004, 2012, 2020), tau+.30, label=tau, col='red', cex=1)
abline(v = c(2004,2012,2020), lty='dotted')
```



Or if we want a series that has (start, middle, end) values of $(0.3, 0.8, -0.1)$ we use

```
tau = c( 0.3, 0.8, -0.1)

plot(year, Sstar %*% tau, type='o',ylim=c(-0.3,1),
     main='S* %*% c(0.3, 0.8, -0.1)')
points(c(2004,2012,2020), tau, pch=16, cex=2, col='red',)
text( c(2004, 2012, 2020), tau+.10, label=tau, col='red', cex=1)
abline(v = c(2004,2012,2020), lty='dotted')
```


S* %*% c(0.3, 0.8, −0.1)
Sstar %*% tau
1.0
0.8
0.6
0.4
0.2
−0.2
0.3
0.8
−0.1
2005
2010
2015
2020
year