

ASSIST: A Suite of S functions Implementing Spline smoothing Techniques

Yuedong Wang

Department of Statistics and Applied Probability
University of California-Santa Barbara

Chunlei Ke

Amgen

Outline

- 1 ASSIST Package
- 2 Smoothing Spline Regression Models
- 3 Non-Parametric Nonlinear Regression Models
- 4 Semi-parametric Nonlinear Regression Models
- 5 Semi-parametric Nonlinear Mixed-Effects Models

Outline

- 1 ASSIST Package
- 2 Smoothing Spline Regression Models
- 3 Non-Parametric Nonlinear Regression Models
- 4 Semi-parametric Nonlinear Regression Models
- 5 Semi-parametric Nonlinear Mixed-Effects Models

Outline

- 1 ASSIST Package
- 2 Smoothing Spline Regression Models
- 3 Non-Parametric Nonlinear Regression Models
- 4 Semi-parametric Nonlinear Regression Models
- 5 Semi-parametric Nonlinear Mixed-Effects Models

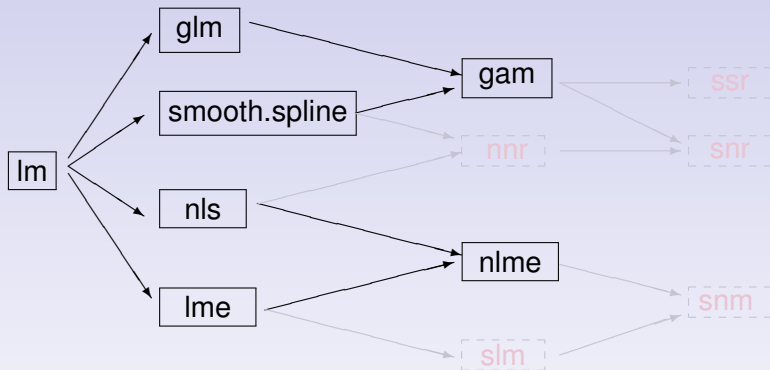
Outline

- 1 ASSIST Package
- 2 Smoothing Spline Regression Models
- 3 Non-Parametric Nonlinear Regression Models
- 4 Semi-parametric Nonlinear Regression Models
- 5 Semi-parametric Nonlinear Mixed-Effects Models

Outline

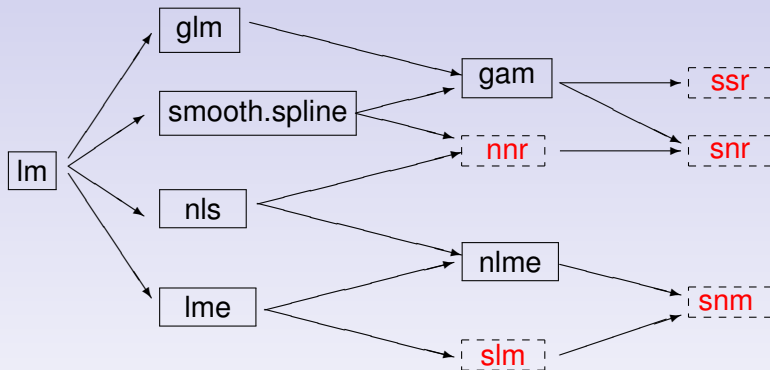
- 1 ASSIST Package
- 2 Smoothing Spline Regression Models
- 3 Non-Parametric Nonlinear Regression Models
- 4 Semi-parametric Nonlinear Regression Models
- 5 Semi-parametric Nonlinear Mixed-Effects Models

An Overview



- SSR: smoothing spline regression model
- NNR: nonlinear nonparametric regression model
- SNR: semi-parametric nonlinear regression model
- SLM: semi-parametric linear mixed effects model
- SNM: semi-parametric nonlinear mixed effects model

An Overview



- **SSR**: smoothing spline regression model
- **NNR**: nonlinear nonparametric regression model
- **SNR**: semi-parametric nonlinear regression model
- **SLM**: semi-parametric linear mixed effects model
- **SNM**: semi-parametric nonlinear mixed effects model

Smoothing Spline Regression Models

A smoothing spline regression model assumes that (Wahba, 1990)

$$y_i = f(t_i) + \epsilon_i, \quad i = 1, \dots, n$$

- The domain of the function $\mathcal{T}$ is an arbitrary set
- f belongs to a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$
- $\epsilon_i \stackrel{iid}{\sim} N(0, \sigma^2)$

Reproducing Kernel Hilbert Space

Definition Let $\mathcal{H}$ be a Hilbert space of real-valued functions from $\mathcal{T}$ to $\mathbb{R}$. For a fixed $t \in \mathcal{T}$, the *evaluational functional* $L_t : \mathcal{H} \rightarrow \mathbb{R}$ is defined as

$$L_t f = f(t)$$

Definition A Hilbert space of real-valued functions $\mathcal{H}$ is a RKHS if every evaluational functional is continuous.

Fact For every RKHS there exist a bivariate function $R(s, t)$ such that

$$(R(t, \cdot), f) = f(t)$$

Definition The bivariate function $R(s, t)$ is called the *reproducing kernel* of the RKHS.

Some Examples

	Polynomial spline	Periodic spline	Thin-plate spline
$\mathcal{T}$	$[0, 1]$	unit circle	$\mathbb{R}^d$
$\mathcal{H}$	$W_2^m[0, 1]$	$W_2^m(per)$	$L_2^m(\mathbb{R}^d)$

$$W_2^m[0, 1] = \{f : f, \dots, f^{(m-1)} \text{ are absolutely continuous, } \int_0^1 (f^{(m)})^2 dt < \infty\}$$

$$W_2^m(per) = \{f : f^{(j)} \text{ are absolutely continuous, } f^{(j)}(0) = f^{(j)}(1), j = 0, \dots, m-1, \\ \int_0^1 (f^{(m-1)})^2 < \infty\}$$

$$L_2^m(\mathbb{R}^d) = \{f : \sum_{\alpha_1 + \dots + \alpha_d = m} \frac{m!}{\alpha_1! \dots \alpha_d!} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \left(\frac{\partial^m f}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}} \right)^2 \prod_{j=1}^d dx_j < \infty\} \\ 2m - d > 0$$

Some Notations

Let

$$\mathcal{H}_0 = \text{span}\{\phi_1, \dots, \phi_M\}$$

be the *null space* consisting functions that will not be penalized.

Let

$$\mathcal{H}_1 = \mathcal{H} \ominus \mathcal{H}_0.$$

Denote $R_1(s, t)$ as the *reproducing kernel* of $\mathcal{H}_1$ and P_1 as the orthogonal projection of f onto $\mathcal{H}_1$.

The choices of $\mathcal{H}_0$, $\mathcal{H}_1$ and P_1 depend on

- domain of the function $\mathcal{T}$
- prior knowledge such as smoothness
- purpose of the study

For commonly used splines, the forms of $\phi_1, \dots, \phi_M$, R_1 and P_1 are known.

Some Examples

	$\phi_1, \dots, \phi_M$	R_1
cubic spline $W_2^2[0, 1]$	$1, t$	$\int_0^{\min(s,t)} (s-u)(t-u)du$
cubic periodic spline $W_2^2(per)$	1	$\sum_{v=1}^{\infty} 2 \cos 2\pi v(s-t)/(2\pi)^4$
thin-plate spline $L_2^2(\mathbb{R}^2)$	$1, t_1, t_2$	$\ \mathbf{s} - \mathbf{t}\ ^2 \ln \ \mathbf{s} - \mathbf{t}\ $

Estimation for Smoothing Spline Regression

The *penalized least squares* (PLS) estimate of f is the solution to

$$\min_{f \in \mathcal{H}} \left\{ \frac{1}{n} \sum_{i=1}^n (y_i - f(t_i))^2 + \lambda \|P_1 f\|^2 \right\}$$

- The first part (LS) measures the goodness-of-fit
- The second part penalizes the departure to the null space $\mathcal{H}_0$
- λ controls the trade-off between goodness-of-fit and the departure to $\mathcal{H}_0$

Kimeldorf-Wahba Representer Theorem

Fixed λ , the solution to the PLS has the form

$$\hat{f}_\lambda(t) = \sum_{i=1}^M d_i \phi_i(t) + \sum_{j=1}^n c_j R_1(t_j, t)$$

Let $T_{n \times M} = \{\phi_v(t_i)\}_{i=1}^n_{v=1}^M$ and $\Sigma = \{R_1(t_i, t_j)\}_{i,j=1}^n$. The coefficients $\mathbf{d} = (d_1, \dots, d_M)^T$ and $\mathbf{c} = (c_1, \dots, c_n)^T$ are solutions to

$$\begin{aligned} (\Sigma + n\lambda I)\mathbf{c} + T\mathbf{d} &= \mathbf{y} \\ T^T\mathbf{c} &= 0 \end{aligned}$$

Estimate of the Smoothing Parameter

The choice of the smoothing parameter λ is critical to the performance of a spline estimate. Three data-adaptive methods, Generalized Cross Validation (GCV), Generalized Maximum Likelihood (GML) and Unbiased Risk (UBR), estimate λ as the minimizers of the following functions

$$\text{GCV}(\lambda) = \frac{\frac{1}{n} \|(I - A(\lambda))\mathbf{y}\|^2}{\left[\frac{1}{n} \text{tr}(I - A(\lambda))\right]^2}$$

$$\text{GML}(\lambda) = \frac{\mathbf{y}^T (I - A(\lambda)) \mathbf{y}}{[\det^+((I - A(\lambda)))]^{1/(n-M)}}$$

$$U(\lambda) = \frac{1}{n} \|(I - A(\lambda))\mathbf{y}\|^2 + \frac{2\sigma^2}{n} \text{tr}A(\lambda)$$

where $A(\lambda)$ is the hat matrix

The `ssr` Function

To fit a smoothing spline regression model, we need to specify

- basis functions $\phi_1, \dots, \phi_M$ of the null space $\mathcal{H}_0$
- Reproducing kernel $R_1(s, t)$ of $\mathcal{H}_1$

```
ssr(formula, rk, spar)
```

Required

- `formula` specifies the response on the left of a $\sim$ operator, and bases $\phi_1(t), \dots, \phi_M(t)$ on the right
- `rk` specifies $R_1(s, t)$. Expressions of commonly used reproducing kernels are available. Users can easily add their own kernels

Optional

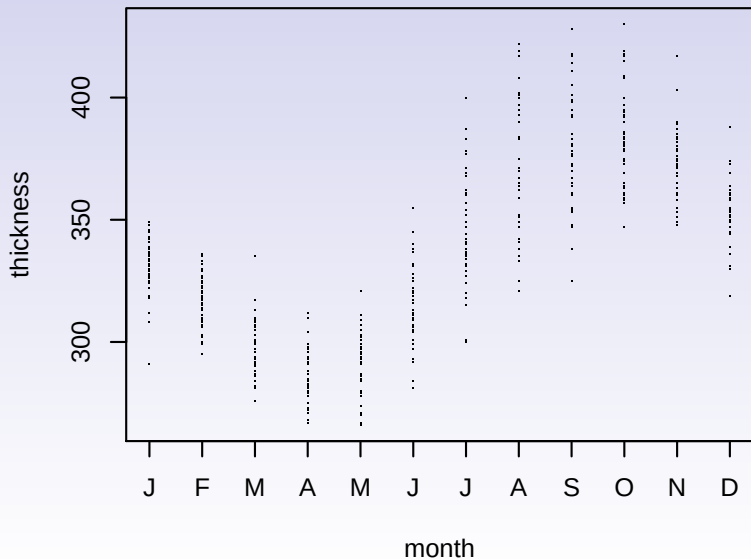
- `spar` specifies the method for selecting the smoothing parameter(s). The default is the GCV
- There are many other options

Ozone Data

We use a ozone data set to illustrate the `ssr` function. Monthly mean ozone thickness (Dobson units) in Arosa, Switzerland from 1926 to 1971 were recorded. First, we ignore the year effect and investigate how ozone thickness changes in a year.

```
> library(assist)
> data(Arosa)
> attach(Arosa)
> Arosa[1:10,]
   year month thick
1     1     1   312
2     1     2   300
3     1     3   281
.     .     .     .
.     .     .     .
.     .     .     .
```

Scatter Plot



Fit a Cubic Spline

```
> fit1 <- ssr(thick~month, rk=cubic(month),  
              scale=T, spar="m", data=Arosa)
```

- `cubic`: function calculates the RK for a cubic spline on $[0, 1]$
- `scale=T`: transforms the `month` variable into $[0, 1]$
- `spar="m"`: use the GML method to choose the smoothing parameter

Summarizing `ssr` Fits

```
> summary(fit1)
```

```
Smoothing spline regression fit by GML method
Call: ssr(formula=thick~month, rk=cubic(month),
  data=Arosa, scale=T, spar='`m`')
```

```
Coefficients (d):
```

(Intercept)	month
326.01578	21.33211

```
GML estimate(s) of smoothing parameter(s) :
```

```
2.149083e-06
```

```
Equivalent Degrees of Freedom (DF): 9.566868
```

```
Estimate of sigma: 16.79393
```

- $DF = tr(A(\hat{\lambda}))$

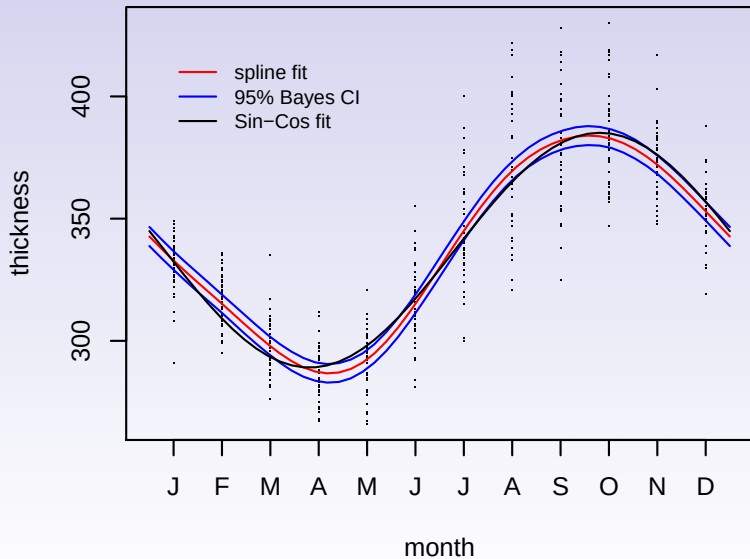
Fit a Periodic Spline

Since the mean function is periodic, it is more natural to fit a periodic spline.

```
> Arosa$csmonth <- (Arosa$month-0.5)/12  
> fit2 <- ssr(thick~1, rk=periodic(csmonth),  
              data=Arosa)
```

- `csmonth`: transform of month variable into $[0, 1]$
- `periodic`: function calculates the RK for a periodic spline on $[0, 1]$
- Use the default method (GCV) to select the smoothing parameter

Fits and Confidence Intervals



Partial Spline

To check if the simple sine-cosine model fits the data adequately, one may use the following *partial spline* model

$$\begin{aligned} & \text{thickness}(\text{csmnth}) \\ = & \mu + \beta_1 \sin(2\pi \text{csmnth}) + \beta_2 \cos(2\pi \text{csmnth}) + \\ & f(\text{csmnth}) + \epsilon(\text{csmnth}), \end{aligned}$$

- Sine and cosine functions are added to the null space
- $f \in W_2^2(per) \ominus \{1\}$ represents part of the function that is not explained by the simple sine-cosine model

Fit the Partial Spline and Test

```
> fit3<-ssr(thick~sin(2*pi*csmnth)+cos(2*pi*csmnth),
            rk=periodic(csmnth), spar='m', data=Arosa)
> anova(fit3,simu.size=500)
```

Testing H_0 : f in the NULL space

	test.value	simu.size	simu.p-value	approx.p-value
LMP	0.001262064	500	0	
GML	0.9553638	500	0	3.490208e-12

- `anova`: test the hypothesis that the unknown function lies in the null space, i.e. $f = 0$
- `simu.size`: the number of simulations to calculate p-values based on simulation
- LMP: locally most powerful (score) test
- GML: generalized maximum likelihood (likelihood ratio) test

A New Penalty

- The penalty for the above partial spline is

$$||P_1 f||^2 = \int_0^1 (f'')^2 dt$$

- $P_1 f \neq 0$ for sine and cosine functions
- The parametric part of the partial spline model is not orthogonal to $W_2^2(per) \ominus \{1\}$
- A more efficient approach to test the parametric model is to use a periodic *L-spline* model with penalty

$$||P_1 f||^2 = \int_0^1 (Lf)^2 dt$$

where $L = D[D^2 + (2\pi)^2]$. It is easy to see that $Lf = 0$ for sine and cosine functions

Periodic L -Spline Model

$$\begin{aligned} & \text{thickness}(\text{csmo}n\text{th}) \\ = & \mu + \beta_1 \sin(2\pi \text{csmo}n\text{th}) + \beta_2 \cos(2\pi \text{csmo}n\text{th}) + \\ & f(\text{csmo}n\text{th}) + \epsilon(\text{csmo}n\text{th}), \end{aligned}$$

- $f \in W_2^2(per) \ominus \{1, \sin(2\pi \text{csmo}n\text{th}), \cos(2\pi \text{csmo}n\text{th})\}$

Fit the *L*-Spline and Test

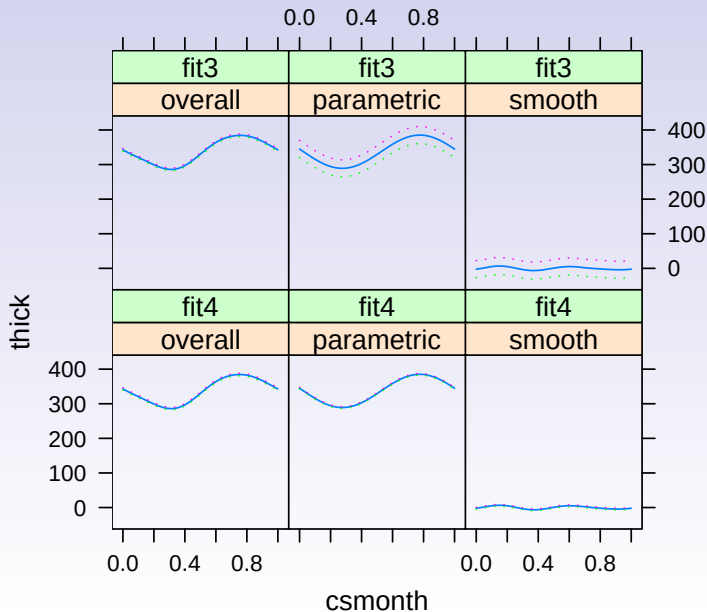
```
> fit4 <- update(fit3,
                  rk=lspline(csmnth,type='sinel'))
> anova(fit4,simu.size=500)
```

Testing H_0 : f in the NULL space

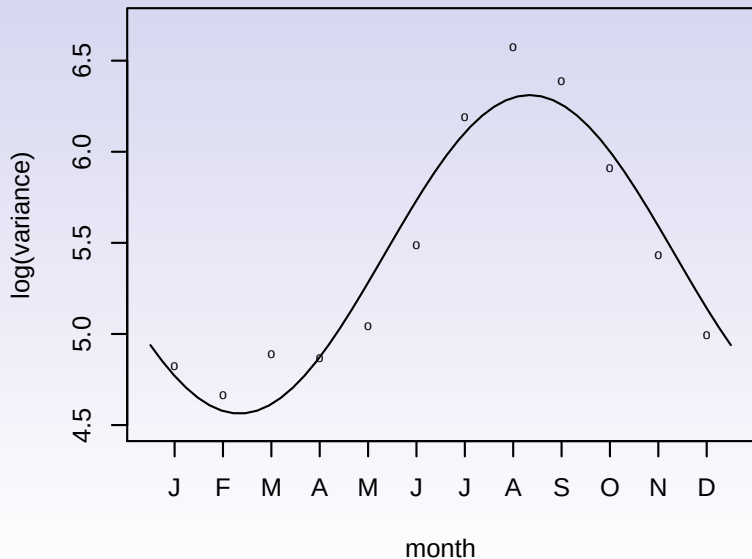
	test.value	simu.size	simu.p-value	approx.p-value
LMP	2.5391e-06	500	0	
GML	0.9533696	500	0	1.164513e-12

- `lspline`: calculates RKs for some L-splines
- `type`: specifies the type of L-splines

Components and Confidence Intervals



Unequal Variances



Fit a L -spline Model With Unequal Variances

The line represents fits from a simple sine and cosine model. It suggests the following variance function

$$\begin{aligned} & \text{Var}(\epsilon(\text{csmmonth})) \\ &= \sigma^2 \exp\{2[a \sin(2\pi \text{csmmonth}) + b \cos(2\pi \text{csmmonth})]\} \end{aligned}$$

```
> fit5 <- update(fit4,
  weights=varComb(varExp(form=~sin(2*pi*csmmonth)),
    varExp(form=~cos(2*pi*csmmonth)))
```

- `weights`: a vector or a matrix specifying known weights for weighted smoothing, or a `varFunc` structure specifying a variance function structure
- `varFunc` structures: same as in `nlme`

Summary

```
> summary(fit5)
```

```
...
```

```
Coefficients (d):
```

```
(Intercept) sin(2*pi*csmnth) cos(2*pi*csmnth)
      337.0753      -47.42951       7.773344
```

```
GML estimate(s) of smoothing parameter(s) :
```

```
3.676933e-09
```

```
Equivalent Degrees of Freedom (DF): 6.84808
```

```
Estimate of sigma: 15.22469
```

```
Combination of:
```

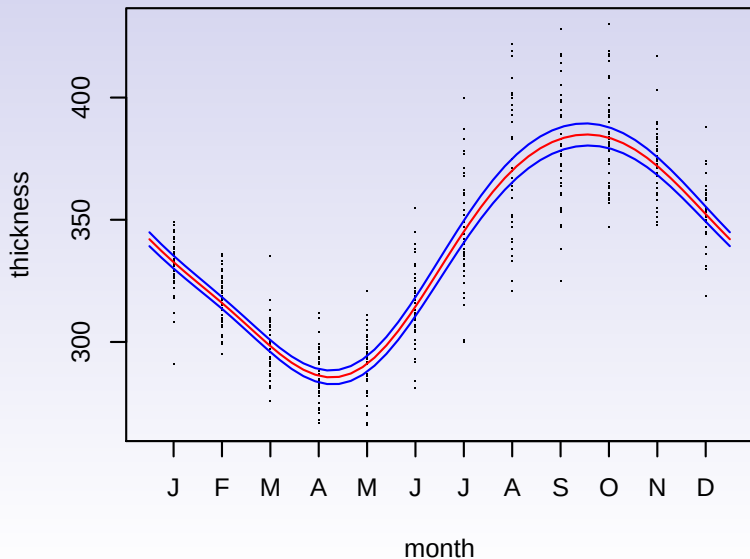
```
Variance function structure of class varExp representing
      expon
```

```
-0.3555964
```

```
Variance function structure of class varExp representing
      expon
```

```
-0.2496355
```

L-Spline Fits With Unequal Variances



Correlation

Observations close in time may be correlated. Suppose that we want to use a continuous first-order autoregressive structure for random errors: the covariance between the random error of month i_1 in year j_1 and the random error of month i_2 in year j_2 is

$$\sqrt{\sigma^2((i_1 - .5)/12)\sigma^2((i_2 - .5)/12)\rho^{|i_1 - i_2| + 12|j_1 - j_2|}}$$

```
> Arosa$z <- month+12*(year-1)
> fit6 <- update(fit5,corr=corCAR1(form=~z))
```

- **corr**: a **corStruct** object describing the correlation structure
- **corClasses**: same as in **nlme**

Summary

```
> summary(fit6)
```

```
...
```

```
Coefficients (d):
```

```
(Intercept) sin(2*pi*csmnth) cos(2*pi*csmnth)
  336.896826   -47.345807         7.788973
```

```
GML estimate(s) of smoothing parameter(s): 3.57509e-09
```

```
Equivalent Degrees of Freedom (DF): 7.327918
```

```
Estimate of sigma: 15.31094
```

```
Correlation structure of class corCAR1 representing
```

```
Phi
```

```
0.3411413
```

```
Combination of:
```

```
Variance function structure of class varExp representing
```

```
expon
```

```
-0.3602905
```

```
Variance function structure of class varExp representing
```

```
expon
```

```
-0.3009272
```

Smoothing Spline ANOVA Decomposition

Now consider the effects of both `month` and `year` variables. Modeling the joint function as the tensor product of a periodic spline for `month` and a cubic spline for `year`:

$$\begin{aligned}
 & W_2^2(per) \otimes W_2^2[0, 1] \\
 = & (\{1\} \oplus H_1^1) \otimes (\{1\} \oplus \{year\} \oplus H_2^2) \\
 = & (\{1\}) && \text{constant} \\
 \oplus & (\{year\}) && \text{linear}(year) \\
 \oplus & (H_1^1) && \text{smooth}(month) \\
 \oplus & (H_2^2) && \text{smooth}(year) \\
 \oplus & (H_1^1 \otimes \{year\}) && s(month) * l(year) \\
 \oplus & (H_1^1 \otimes H_2^2) && s(month) * s(year)
 \end{aligned}$$

Smoothing Spline ANOVA Model

We have the following SS ANOVA model

$$\begin{aligned} & \text{thickness}(\text{csmo}n\text{th}, \text{csyear}) \\ = & \mu + \beta \text{csyear} + \mathbf{s}_1(\text{csmo}n\text{th}) + \mathbf{s}_2(\text{csyear}) + \\ & \mathbf{s}/(\text{csmo}n\text{th}, \text{csyear}) + \mathbf{ss}(\text{csmo}n\text{th}, \text{csyear}) + \\ & \epsilon(\text{csmo}n\text{th}, \text{csyear}) \end{aligned}$$

- μ : overall mean
- $\beta \text{csyear} + \mathbf{s}_2(\text{csyear})$: year main effect
- $\mathbf{s}_1(\text{csmo}n\text{th})$: month main effect
- $\mathbf{s}/(\text{csmo}n\text{th}, \text{csyear}) + \mathbf{ss}(\text{csmo}n\text{th}, \text{csyear})$: interaction

Smoothing Spline ANOVA Model

```
> fit7 <- ssr(thick~I(cyear-0.5),
  rk=list(periodic(csmnth), cubic(cyear),
  rk.prod(periodic(csmnth), kron(cyear-.5)),
  rk.prod(periodic(csmnth), cubic(cyear))),
  method="m", data=Arosa)
> summary(fit7)
...
Coefficients (d):
  (Intercept) I(cyear - 0.5)
    336.829814      6.031531

GML estimate(s) of smoothing parameter(s) :
  1.53135e-06 2.77910e-07 6.17499e-01 2.02682e-02
Equivalent Degrees of Freedom (DF): 24.59540
Estimate of sigma: 15.84154
```

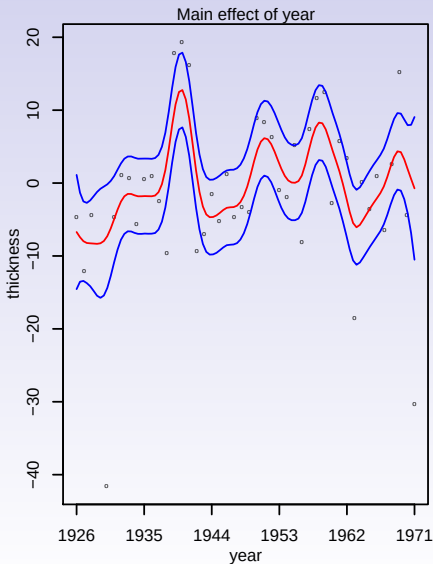
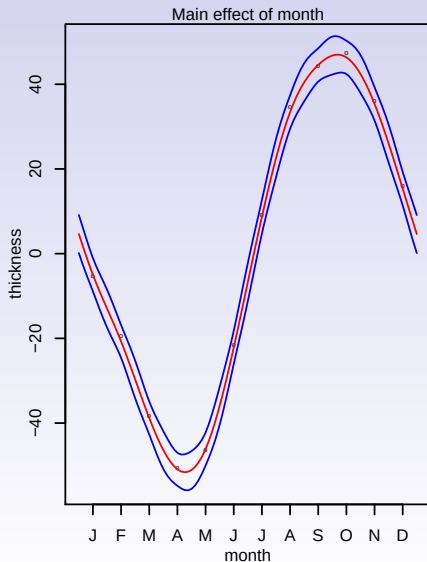
- `rk.prod`: computes the products of RKs

Fit An Additive Model

The estimate of the interaction is quite small. We delete the interaction terms and fit an additive model.

```
> fit8 <- update(fit7,  
  rk=list(periodic(csmoonth), cubic(csyearch)))  
> summary(fit8)  
...  
Coefficients (d):  
  (Intercept) I(csyearch - 0.5)  
    336.82874         5.98981  
  
GML estimate(s) of smoothing parameter(s) :  
  1.537247e-06 2.726700e-07  
Equivalent Degrees of Freedom (DF):  24.66142  
Estimate of sigma:  15.83780
```

Plots of Yearly Trend and Seasonal Trend



Other Applications of `ssr`

We have shown how to use `ssr` to fit cubic, periodic, partial and L splines. `ssr` can also be used to fit

- 1 thin-plate splines, `rk=tps`
- 2 splines on a sphere, `rk=sphere`
- 3 tensor products of all these splines, i.e. smoothing spline ANOVA models, `rk.prod`
- 4 Binomial, Poisson and Gamma data, `family=""`
- 5 Functional linear models (Ramsay and Silverman, 1997)

Non-Parametric Nonlinear Regression Models

$$y_i = \eta(\mathbf{f}; \mathbf{t}_i) + \epsilon_i$$

- η : a known function of $\mathbf{t}_i = (t_{1i}, \dots, t_{di})$ in an arbitrary domain $\mathcal{T}$
- $\mathbf{f} = (f_1, \dots, f_q)$: a vector of unknown non-parametric functions
- η depends on $\mathbf{f}$ nonlinearly
- For example, we may use $f = \exp(g)$ to enforce positivity on f
- Sometimes f is observed indirectly through a nonlinear functional, e.g. in remote sensing

The `nnr` Function

The `nnr` function fits a NNR model:

```
nnr(formula, func, start, data)
```

Required

- `formula` specifies the response on the left of a $\sim$ operator, and the η function on the right
- `func` specifies basis functions of the null space and RK for each non-parametric function
- `start` specifies initial values for the unknown functions

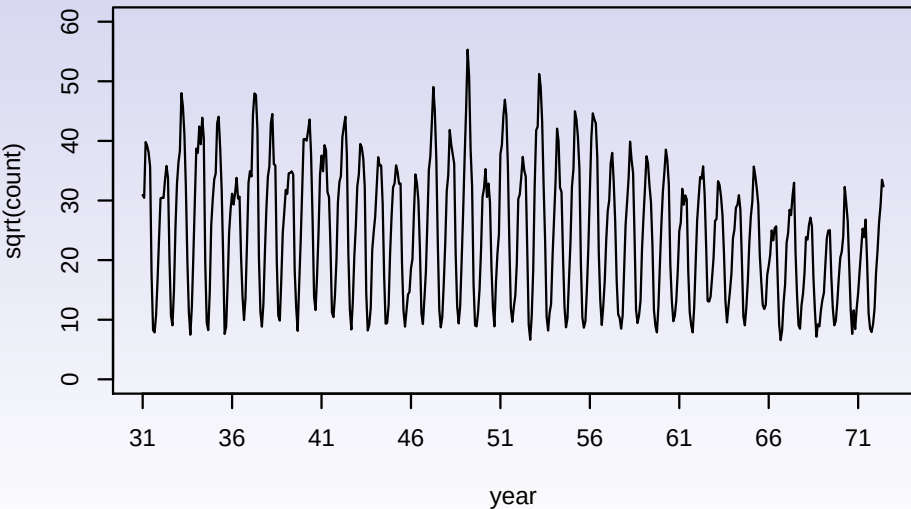
Optional

- `spar` specifies the method for selecting the smoothing parameter(s). The default is the GCV

Chickenpox epidemic

The data set contains monthly number of reported cases of chickenpox in New York City from 1931 to the first six months of 1972. The goal is to investigate dynamics in an epidemic: long term trend over years, seasonal trend and their interactions.

Time series plot



Seasonal variation

- The seasonal variation was mainly caused by two factors:
 - social behavior of children who made close contacts when school was in session
 - temperature and humidity which may affect the survival and transmission of dispersal stages
- Thus the seasonal variations were similar over the years
- We assume that the seasonal variation has the same shape after vertical shift and vertical scale transformations

A multiplicative model for chickenpox epidemic

We assume the following **multiplicative model**

$$y(t_1, t_2) = g_1(t_2) + \exp(g_2(t_2)) \times g_3(t_1) + \epsilon(t_1, t_2)$$

- $y(t_1, t_2)$ is the square root of reported cases in month t_1 of year t_2
- Both t_1 and t_2 are transformed into the interval $[0, 1]$
- g_1 represents yearly mean cases
- g_2 represents magnitude of the seasonal variation. $\exp(g_2(t_2))$ is the amplitude. A bigger amplitude corresponds to a bigger seasonal variation
- g_3 represents seasonal trend
- All component functions have nice interpretations
- g_1 , g_2 and g_3 are unknown and to be modeled nonparametrically
- A extension of the additive models and varying coefficient models

Identifiability and model spaces

- g_1 is modeled using a cubic spline: $g_1 \in W_2^2[0, 1]$
- The exponential transformation of g_2 makes the amplitude positive. Again, g_2 is modeled using a cubic spline. To make g_2 and g_3 identifiable, we need the side condition $\int_0^1 g_2(t)dt = 0$. We achieve this by removing the constant functions from the model space: $g_2 \in W_2^2[0, 1] \ominus \{1\}$
- It has been recognized g_3 periodic and is close to a sinusoidal function, but a simple sinusoidal model may be inappropriate. We use the L -spline with $L = D^2 + (2\pi)^2$ to model g_3 . To make model g_3 identifiable with g_1 , we need the side condition $\int_0^1 g_3(t)dt = 0$. Again, we achieve this by removing the constant functions from the model space: $g_3 \in W_2^2(per) \ominus \{1, \sin 2\pi x, \cos 2\pi x\}$

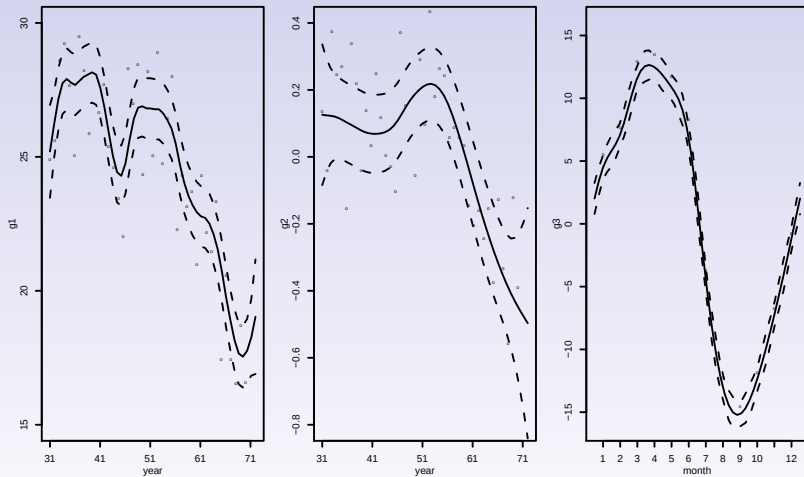
R code

```

S3 <- periodic(chickenpox$csmonth)
g3.tmp <- ssr(ct~1, rk=S3, data=chickenpox,
              spar=' 'm' ')
g3.ini <- as.vector(S3%*%g3.tmp$rkpk.obj$c)
nnr(ct~g1(csyear)+exp(g2(csyear))*g3(csmonth),
     func=list(g1(x)~list(~I(x-.5), cubic(x)),
               g2(x)~list(~I(x-.5)-1, cubic(x)),
               g3(x)~list(~sin(2*pi*x)+
               cos(2*pi*x)-1, lspline(x, type=' 'sine0' '))),
     data=chickenpox,
     start=list(g1=mean(sqrt(count)),
                g2=0, g3=g3.ini),
     control=list(converg=' 'coef' ', spar=' 'm' '))

```

Fits



Semi-parametric Nonlinear Regression Models

$$y_i = \eta(\phi, \mathbf{f}; \mathbf{t}_i) + \epsilon_i$$

- η depends not only on a vector of unknown non-parametric functions, but also on a vector of parameters
- For example, if f is a strictly increasing function, it can be written as

$$f(t) = \eta(\phi, g; t) = \phi + \int_0^t \exp\{g(s)\} ds$$

- Other well known special cases are projection pursuit, single and multiple index, nonlinear partial spline, varying coefficient and self-modeling nonlinear regression models

The `snr` Function

The `snr` function fits an SNR model:

`snr(formula, func, params, start, data)`

Required

- `formula` specifies the response on the left of a $\sim$ operator, and the η function on the right
- `func` specifies basis functions of the null space and RK for each non-parametric function
- `params` specifies models for the parameters
- `start` specifies initial values for the unknown functions and parameters

Optional

- `spar` specifies the method for selecting the smoothing parameter(s). The default is the GCV

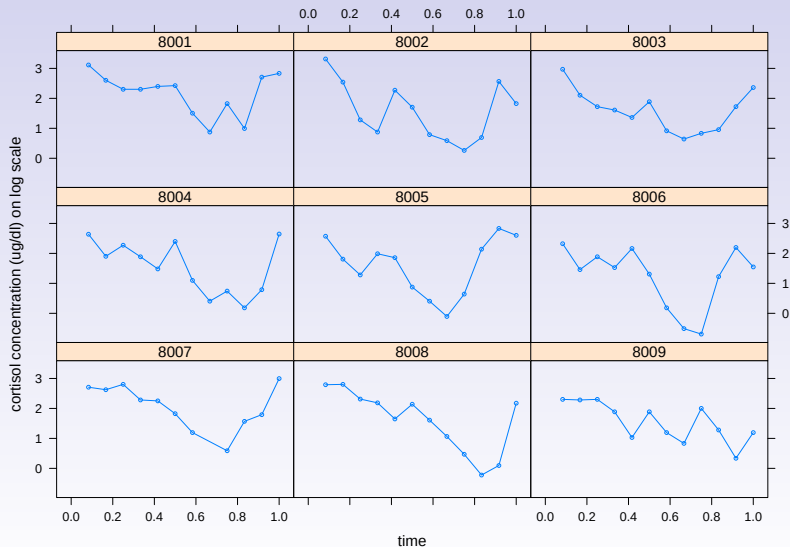
Circadian Rhythm of Cortisol

In an experiment to study immunological responses in humans, blood samples were collected every two hours for 24 hours from 9 healthy normal volunteers, 11 patients with major depression and 16 patients with Cushing's syndrome. These blood samples were analyzed for parameters that measure immune functions and hormones. We will concentrate on cortisol, a hormone that is affected by stress.

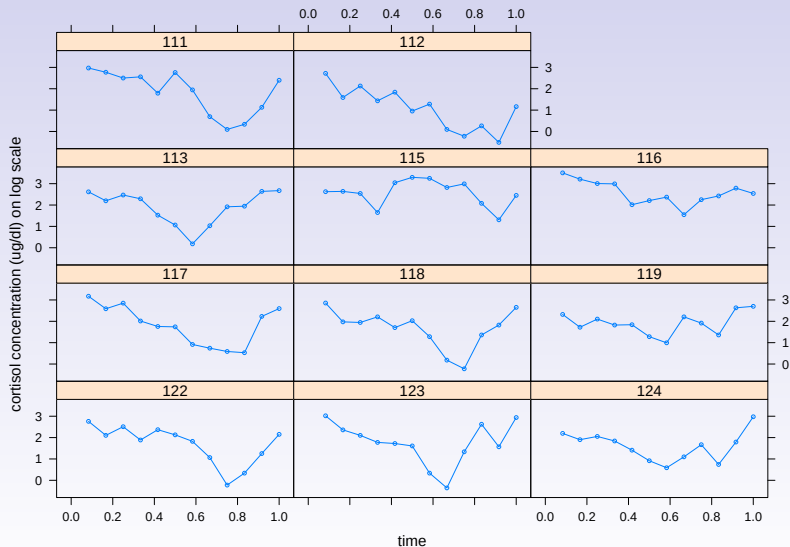
Cushing's syndrome is a hormonal disorder caused by prolonged exposure of the body's tissues to high levels of the hormone cortisol.

Scientific Questions: Do major depression and Cushing's syndrome affect circadian rhythms, and if so, how?

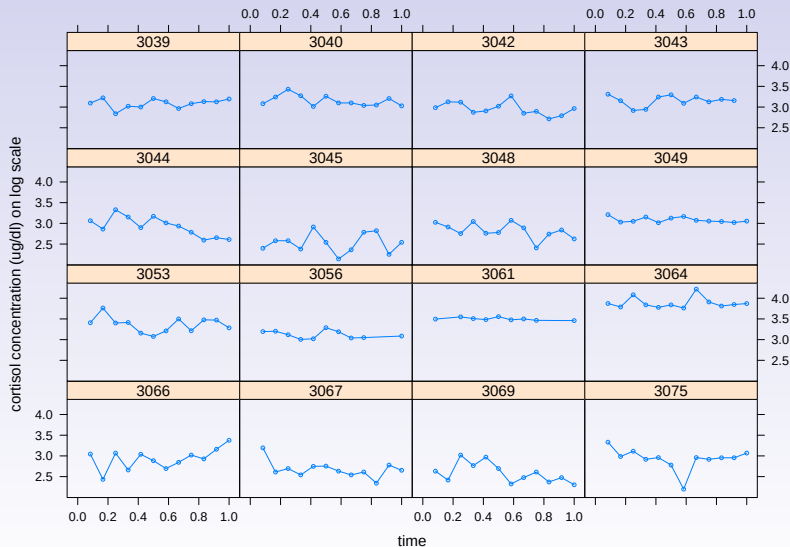
Cortisol Levels from Normal Volunteers



Cortisol Levels from Patients with Major Depression



Cortisol Levels from Patients with Cushing's Syndrome



Assumptions

- There is a common curve for all individuals
- Time axis may be shifted for each subject, i.e. each subject has his own *phase*
- Each subject has his own magnitude of rhythm (*amplitude*)
- Each subject has his own *24-hour mean level*
- Period = 24 h
- For simplicity, we transform the 24-h period into $[0,1]$

Shape Invariant Model

Consider one group of subjects only. The shape invariant model (SIM) assumes that

$$y_{ij} = \beta_{1i} + \exp(\beta_{2i})f(t_{ij} - \text{alogit}(\beta_{3i})) + \epsilon_{ij},$$

$$i = 1, \dots, m, \quad j = 1, \dots, n_i$$

- β_{1i} is the 24-h mean of subject i
- $\exp(\beta_{2i})$ is the amplitude of subject i . The exponential transformation is used to force the amplitude to be positive
- $\text{alogit}(\beta_{3i})$ is the phase of subject i . The inverse logistic transformation, $\text{alogit}(x) = \exp(x)/(1 + \exp(x))$, is used to force the phase to be inside interval $[0,1]$
- f is the common curve. Circadian rhythm exists if $f \neq 0$. We model f using periodic spline $f \in W_2^2(per) \ominus \{1\}$
- $\epsilon_{ij} \stackrel{iid}{\sim} N(0, \sigma^2)$

Shape Invariant Model

SIM is a special case of SNR. Thus it can be fitted using the `snr` function:

```
nor <- horm.cort[horm.cort$type=="normal",]
M <- model.matrix(~as.factor(ID), data=nor)
nor.snr.fit1 <- snr(conc~b1+exp(b2)*f(time-alogit(b3)),
  func=f(u)~list(periodic(u)),
  params=list(b1~M-1, b2+b3~M[, -1]-1),
  start=list(params=c(mean(nor$conc), rep(0, 24))),
  data=nor, spar="m",
  control=list(prec.out=0.001, converg="PRSS"))
```

Semi-parametric Nonlinear Mixed-Effects Models

$$\begin{aligned}y_{ij} &= \eta(\phi_i, \mathbf{f}; \mathbf{t}_{ij}) + \epsilon_{ij} \\ \phi_i &= \mathbf{A}_i\boldsymbol{\beta} + \mathbf{B}_i\mathbf{b}_i\end{aligned}$$

- The first-stage model is the same as a SNR model
- The second-stage model is the same as one for a nonlinear mixed-effect model
- A second-stage model may also be constructed for $\mathbf{f}$

The `snm` Function

The `snm` function fits an SNM model:

`snm(formula, func, fixed, random, start, data)`

Required

- `formula` specifies the response on the left of a $\sim$ operator, and the η function on the right
- `func` specifies basis functions of the null space and RK for each non-parametric function
- `fixed` specifies fixed effects
- `start` specifies initial values for fixed effects

Optional

- `random` specifies random effects
- `group` specifies grouping for random effects
- `spar` specifies the method for selecting the smoothing parameter(s). The default is the GCV
- `correlation` specifies within group correlation structure
- `weights` specifies within group heteroscedasticity structure

Shape Invariant Mixed Effects Model

The SIM ignores correlations within an individual. For cortisol data, it is more appropriate to consider a shape invariant mixed effects model (SIMM):

$$y_{ij} = \mu + b_{1i} + \exp(b_{2i})f(t_{ij} - \text{alogit}(b_{3i})) + \epsilon_{ij},$$

$$i = 1, \dots, m, \quad j = 1, \dots, n_i$$

- fixed effect μ represents 24-hour mean of the population
- random effects b_{1i} , b_{2i} and b_{3i} represent the i th subject's deviation of 24-hour mean, amplitude and phase
- $\mu + b_{1i}$ is the 24-h mean of subject i
- $\exp(b_{2i})$ is the amplitude of subject i
- $\text{alogit}(b_{3i})$ is the phase of subject i
- we assume that $\mathbf{b}_i = (b_{1i}, b_{2i}, b_{3i})^T \stackrel{iid}{\sim} N(0, \sigma^2 D)$, where D is an unstructured positive-definite matrix
- f : same as in SIM
- $\epsilon_{ij} \stackrel{iid}{\sim} N(0, \sigma^2)$

Data Analyses

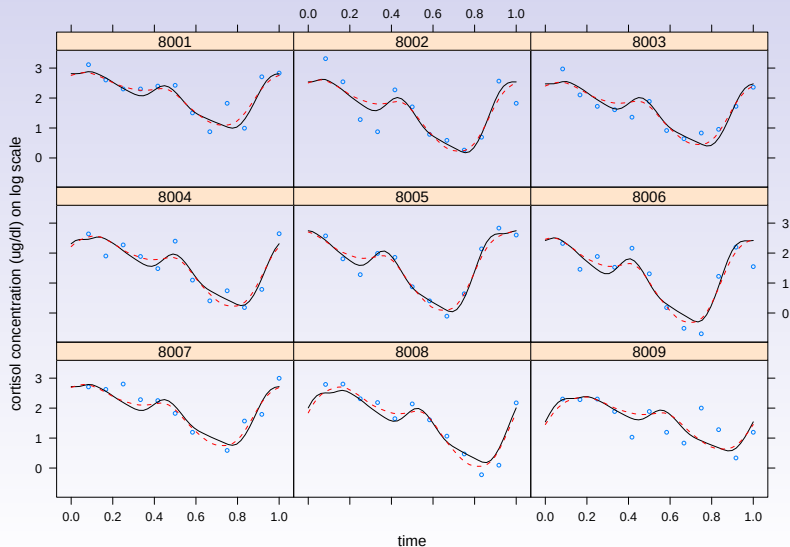
The data set consists the following variables:

- ID: subjects ID number
- time: time when measurements were taken
- conc: hormone cortisol concentration
- group: three groups of subjects: normal, depression and cushing

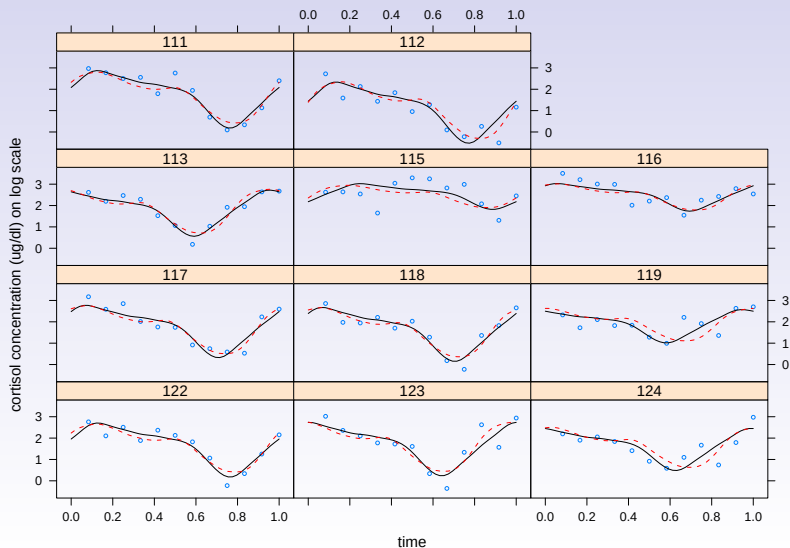
We first fit a SIMM for each group. The following code fits the SIMM to the normal group using `snm`.

```
snm(conc~b1+exp(b2)*f(time-alogit(b3)),
    func=f(u)~list(periodic(u)),
    fixed=list(b1~1), random=list(b1+b2+b3~1),
    groups=~ID, spar='`m`', data=nor,
    start=c(mean(nor$conc)))
```

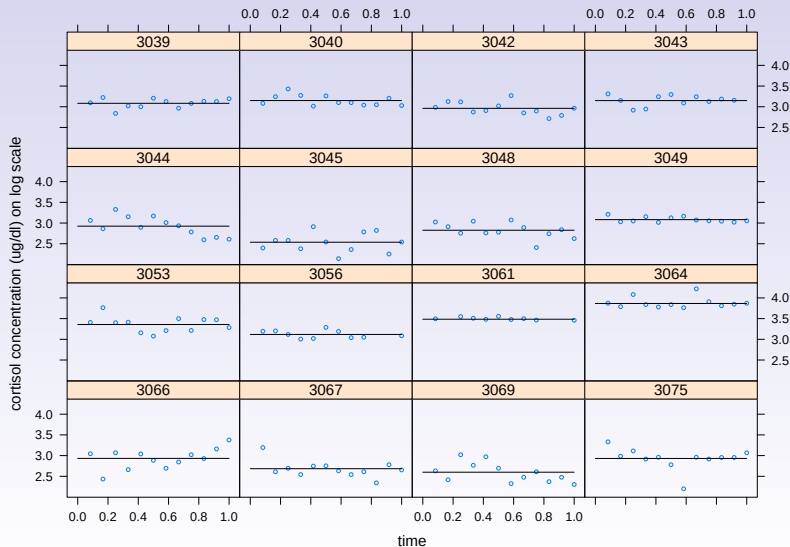
Observations and Fits for Normal Volunteers



Observations and Fits for Patients with Major Depression



Cortisol Levels from Patients with Cushing's Syndrome



Estimates of the Shape Functions

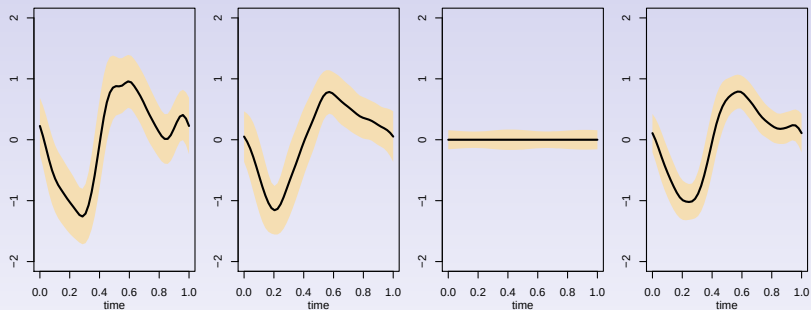


Figure: Solid lines are estimates of the shape functions. Shaded regions are 95% Bayesian confidence intervals. The left three panels are estimates from separate fits, and the right panel is the estimate from the joint fit.

Conclusions

For patients with Cushing's syndrome,

- 24-hour means were elevated to much higher levels
- the common function is almost zero, which indicates that circadian rhythms were lost

The absence of a circadian rhythm has been considered as hallmark of the diagnosis of Cushing's syndrome. Unlike normal subjects, patients with Cushing's syndrome fail to decrease cortisol secretion in the late evening. Therefore, the measurement of elevated late evening cortisol is a very simple and useful way to screen patients for Cushing's syndrome.

Testing Differences Between Groups

To compare `depression` group with the `normal` group, we need to investigate potential effects of depression

- 1 on the shape function
- 2 on the parameters if the shape functions are the same

Same Shape?

From plots we can see that the shape functions for the `normal` and `depression` groups are similar. To formally test this hypothesis, we fit data from these two groups simultaneous

$$y_{ijk} = \mu_k + b_{1ik} + \exp(b_{2ik})f(k, t_{ijk} - \text{alogit}(b_{3ik})) + \epsilon_{ijk},$$

$$i = 1, \dots, m, \quad j = 1, \dots, n_i, \quad k = 1, 2,$$

$$\mathbf{b}_{ik} = (b_{1ik}, b_{2ik}, b_{3ik})^T \stackrel{iid}{\sim} N(0, \sigma_k^2 D_k), \quad \epsilon_{ijk} \stackrel{iid}{\sim} N(0, \sigma_k^2)$$

- $k = 1$ and $k = 2$ correspond to `depression` and `normal` groups respectively
- fixed effects μ_k is the population 24-hour mean of group k
- random effects b_{1ik} , b_{2ik} and b_{3ik} represent the i th subject's deviation of 24-hour mean, amplitude and phase
- different correlation structures for the random effects in each group
- subjects are nested within group

Smoothing Spline ANOVA decomposition of f

f is a function of both `group` (denoted as k) and `time` (denoted as t). Thus we have different common functions for each `group`. We want to test

$$H_0 : f(1, t) = f(2, t)$$

We use the following SS ANOVA decomposition

$$f(k, t) = s(t) + ss(k, t)$$

- $s(t) = (f(1, t) + f(2, t))/2$ is the main effect of `time`
- $ss(k, t) = f(k, t) - s(t)$ is the interaction between `group` and `time`

Since $ss(k, t) = (f(1, t) - f(2, t)) * (I(k = 1) - I(k = 2))$, the hypothesis

$$H_0 : f(1, t) = f(2, t)$$

is equivalent to

$$H_0 : ss(k, t) = 0$$

Same Shape?

```

nordep <- horm.cort[horm.cort$type!="cushing",]
nordep$type <- as.factor(as.vector(nordep$type))
snm(conc~b1+exp(b2)*f(type,time-alogit(b3)),
    func=f(g,u)~list(list(periodic(u),
    rk.prod(shrink1(g),periodic(u)))),
    data=nordep, fixed=list(b1~type),
    random=list(b1+b2+b3~1), groups=~ID,
    weights=varIdent(form=~1|type),
    spar="m", start=c(1.8,-.2))

```

- The estimate of $ss(k, t)$ is essentially zero. Thus $f(1, t) = f(2, t)$

Differences in Means, Amplitudes and Phases

Under the assumption of one shape function for both groups, we now can investigate differences of 24-hour mean, amplitude, and phase between two groups. For this purpose, consider the following model

$$y_{ijk} = \mu_k + b_{1ik} + \exp(b_{2ik} + d_1 \times I(k = 2)) \times \\ f(t_{ijk} - \text{alogit}(b_{3ik} + d_2 \times I(k = 2))) + \epsilon_{ijk}, \\ i = 1, \dots, m, \quad j = 1, \dots, n_i, \quad k = 1, 2,$$

where d_1 and d_2 measures the differences of amplitude and phase between `normal` and `depression` groups.

Differences in Means, Amplitudes and Phases

```
snm(conc~b1+exp(b2+d1*I(type=="normal"))*
    f(time-alogit(b3+d2*I(type=="normal"))),
    func=f(u)~list(periodic(u)),
    data=nordep, fixed=list(b1~type,d1+d2~1),
    random=list(b1+b2+b3~1), groups=~ID,
    weights=varIdent(form=~1|type),
    spar="m", start=c(1.9,-0.3,0,0))
```

- Differences of 24-hour mean ($\mu_1 - \mu_2$) and amplitude (d_1) between two groups are significant. For patients with major depression, 24-hour means were elevated and amplitudes were dampened.
- Difference of phase between two groups (d_2) is not significant