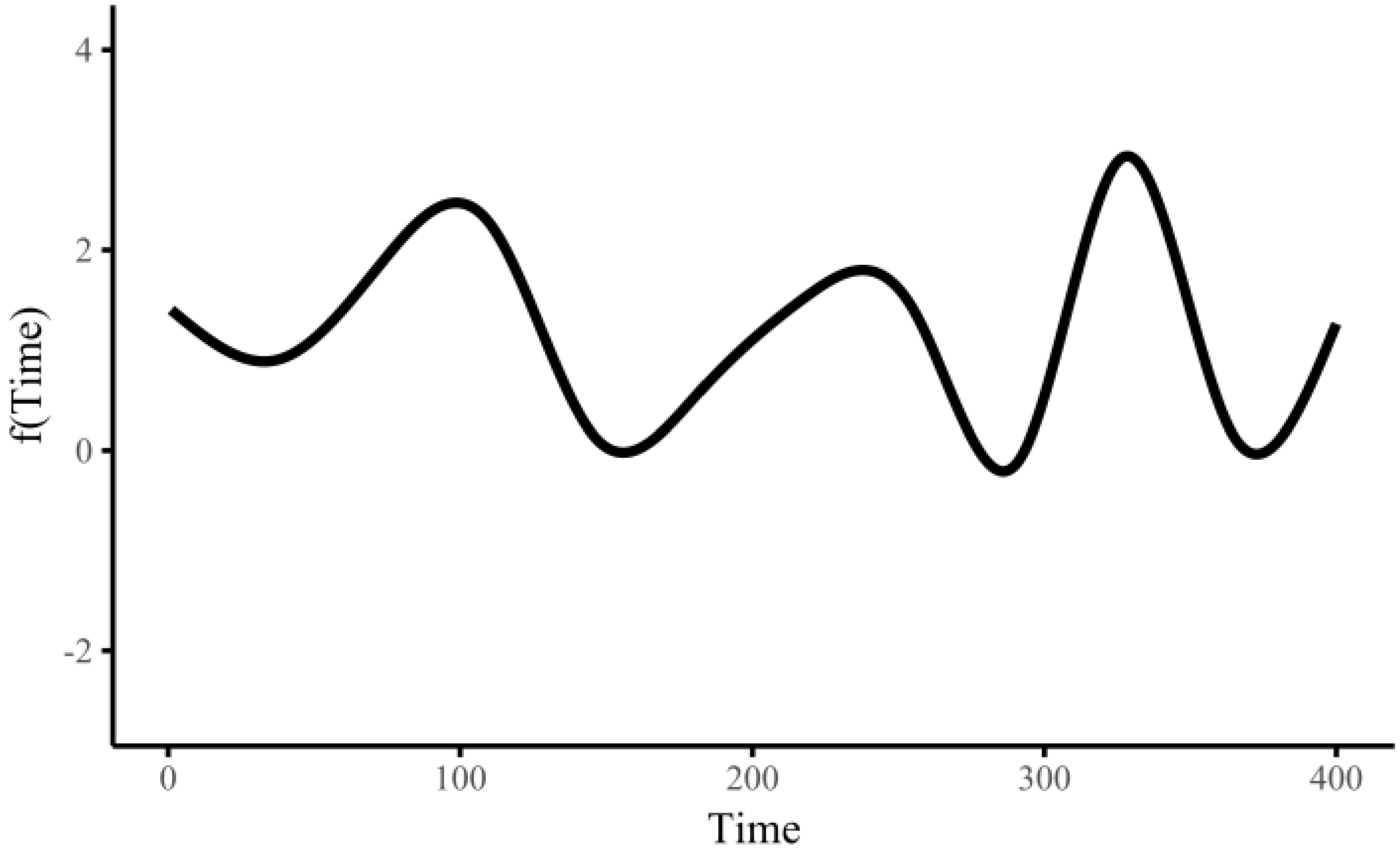
Time series modeling with Bayesian Dynamic Generalized Additive Models

Nicholas Clark

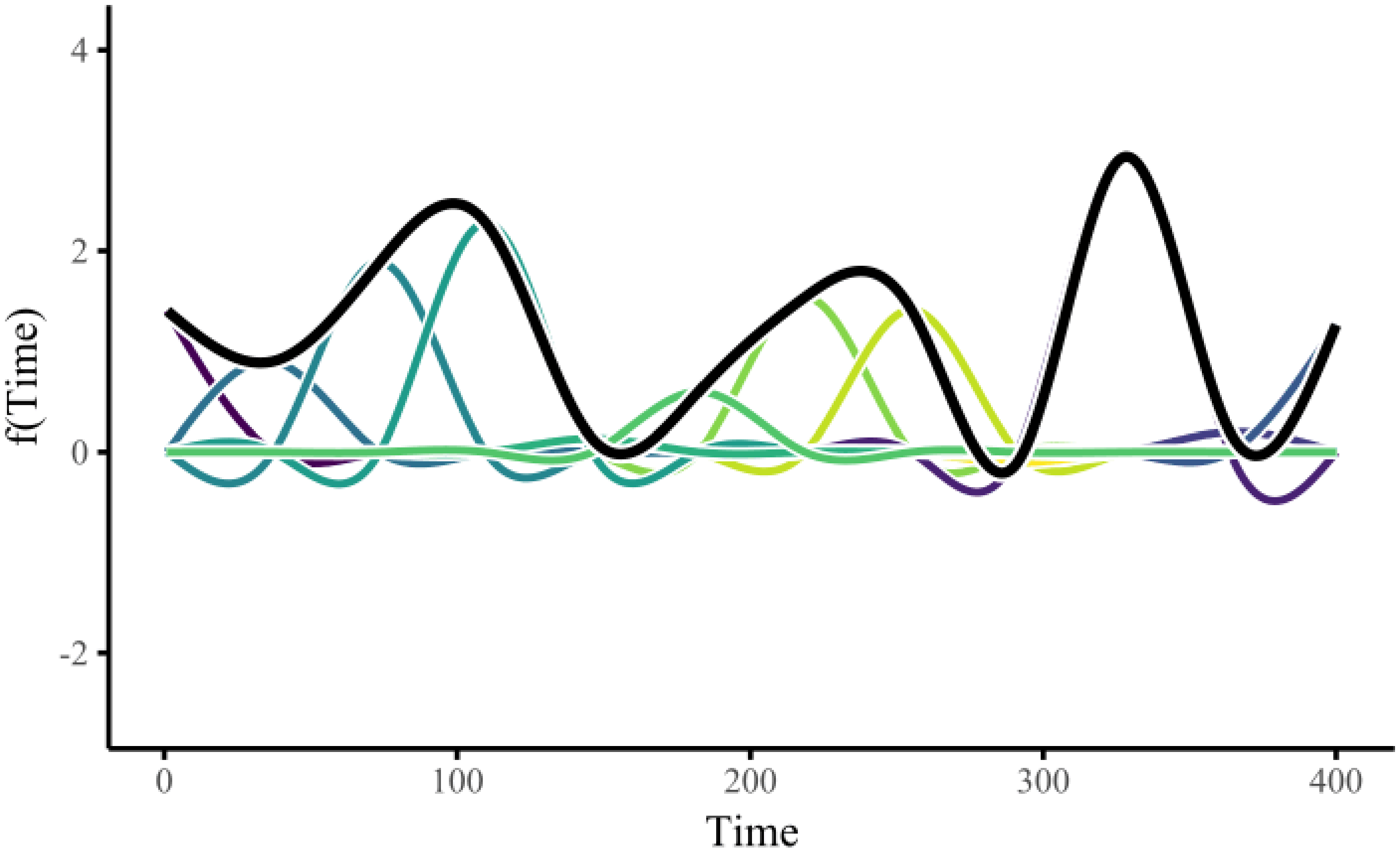
School of Veterinary Science, University of Queensland, Australia

Wednesday 13th December, 2023

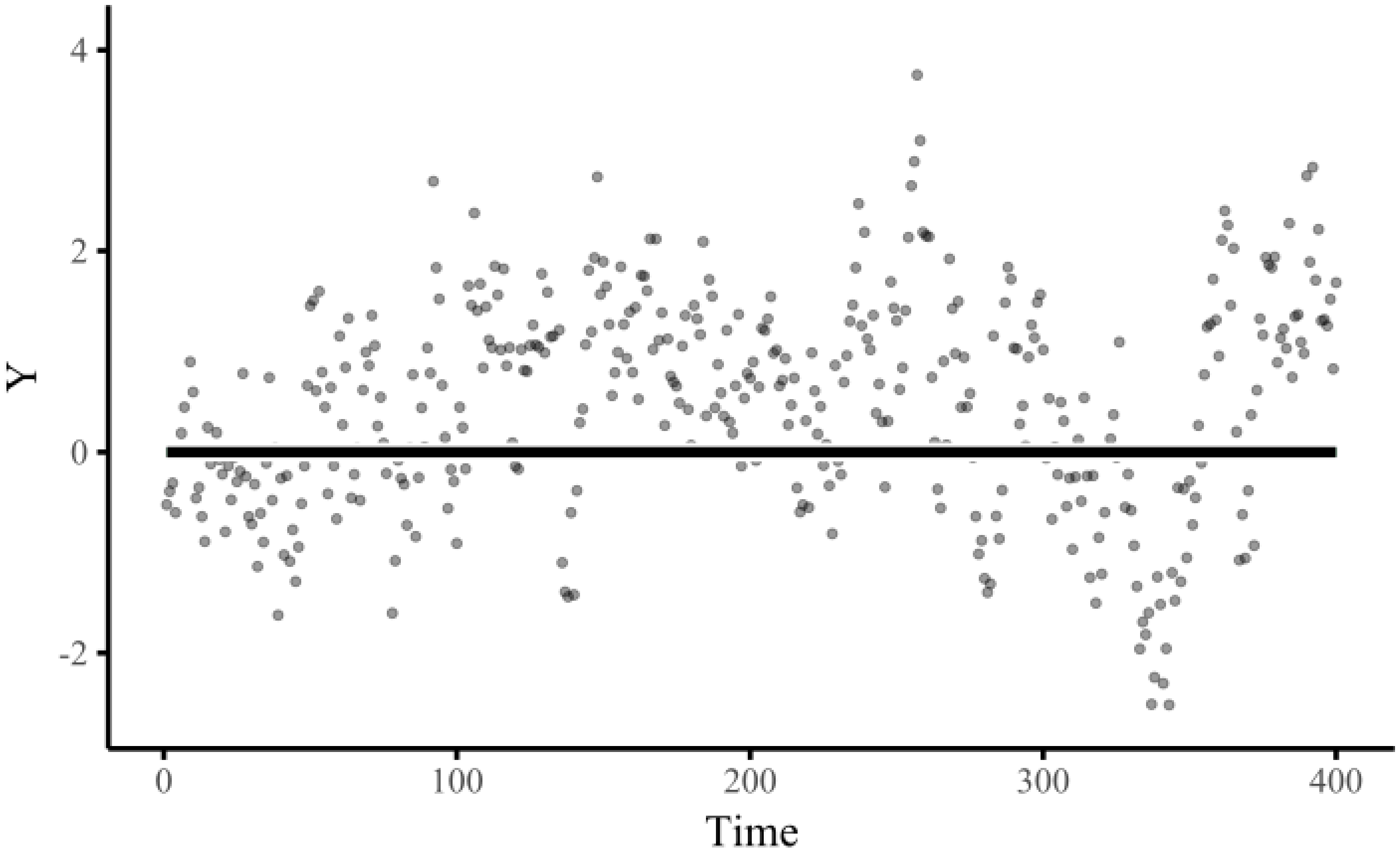
GAMs use splines $(f(x))$ ...



...made of weighted basis functions



Penalize $f''(x)$ to learn weights



Easy to fit in R

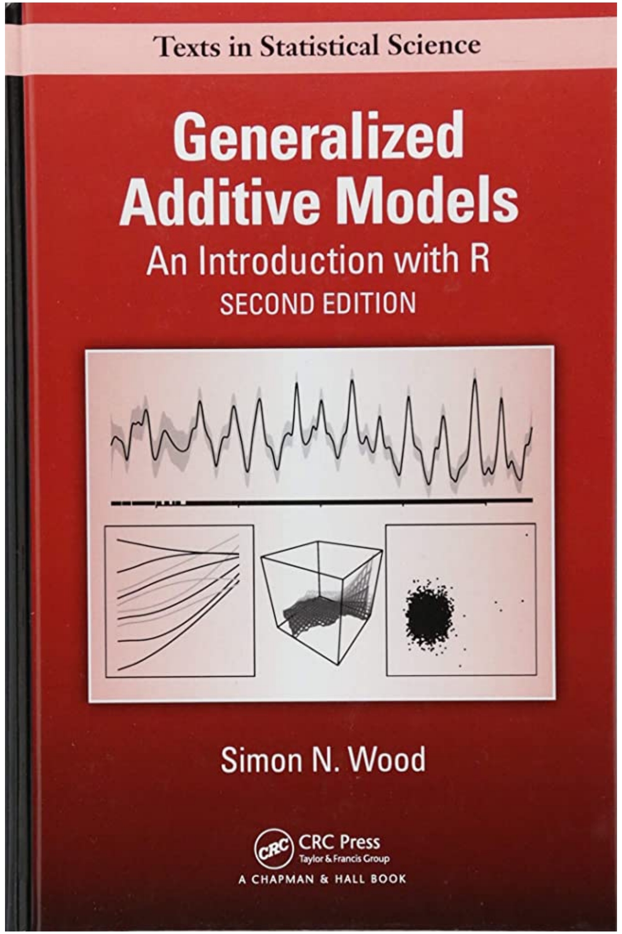
$$\mathbb{E}(\boldsymbol{Y}_t|\boldsymbol{X}_t) = g^{-1}(\alpha + \sum_{j=1}^{J} f(x_{jt}))$$

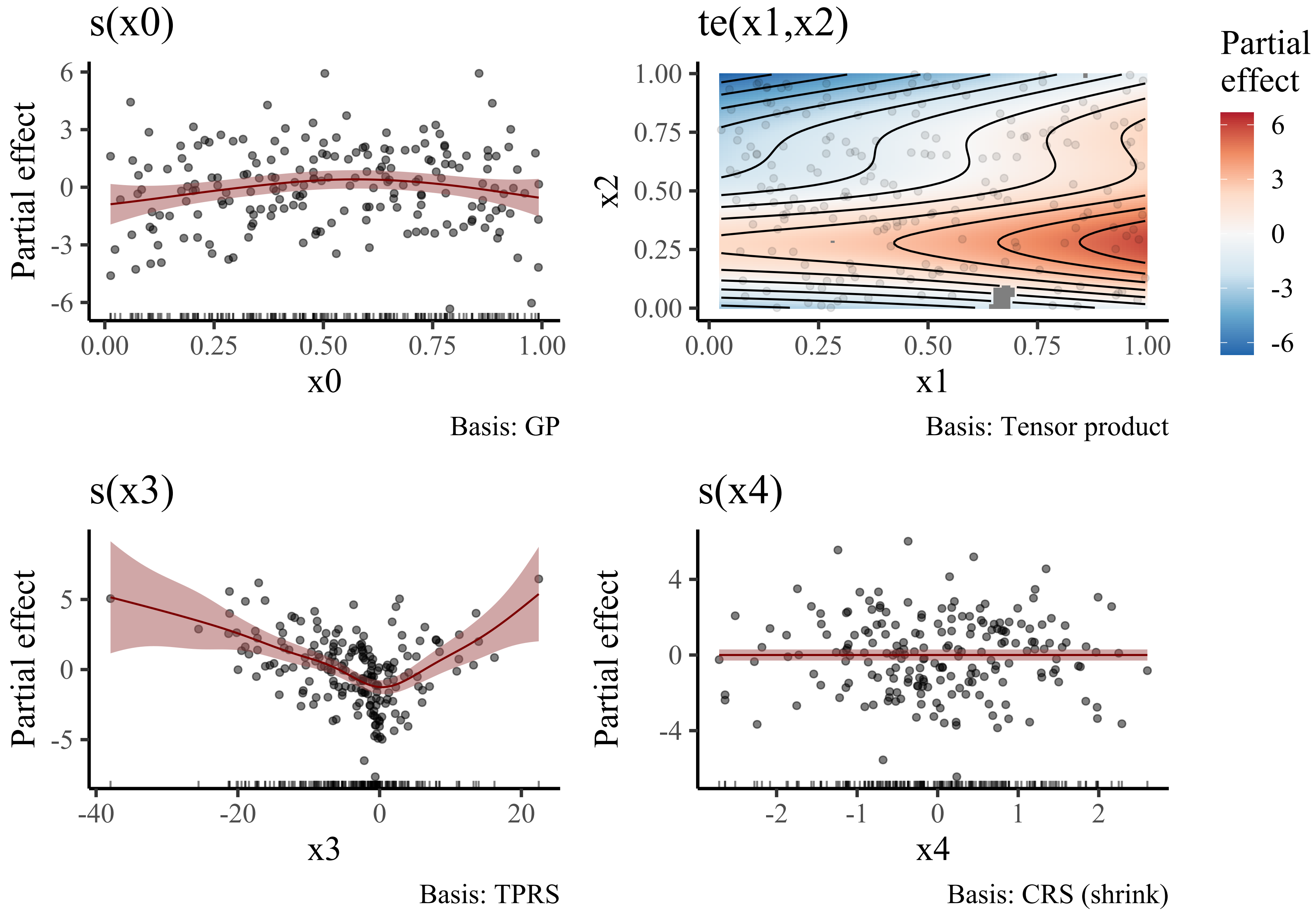
Where:

g^{-1} is the inverse of the link function

α is the intercept

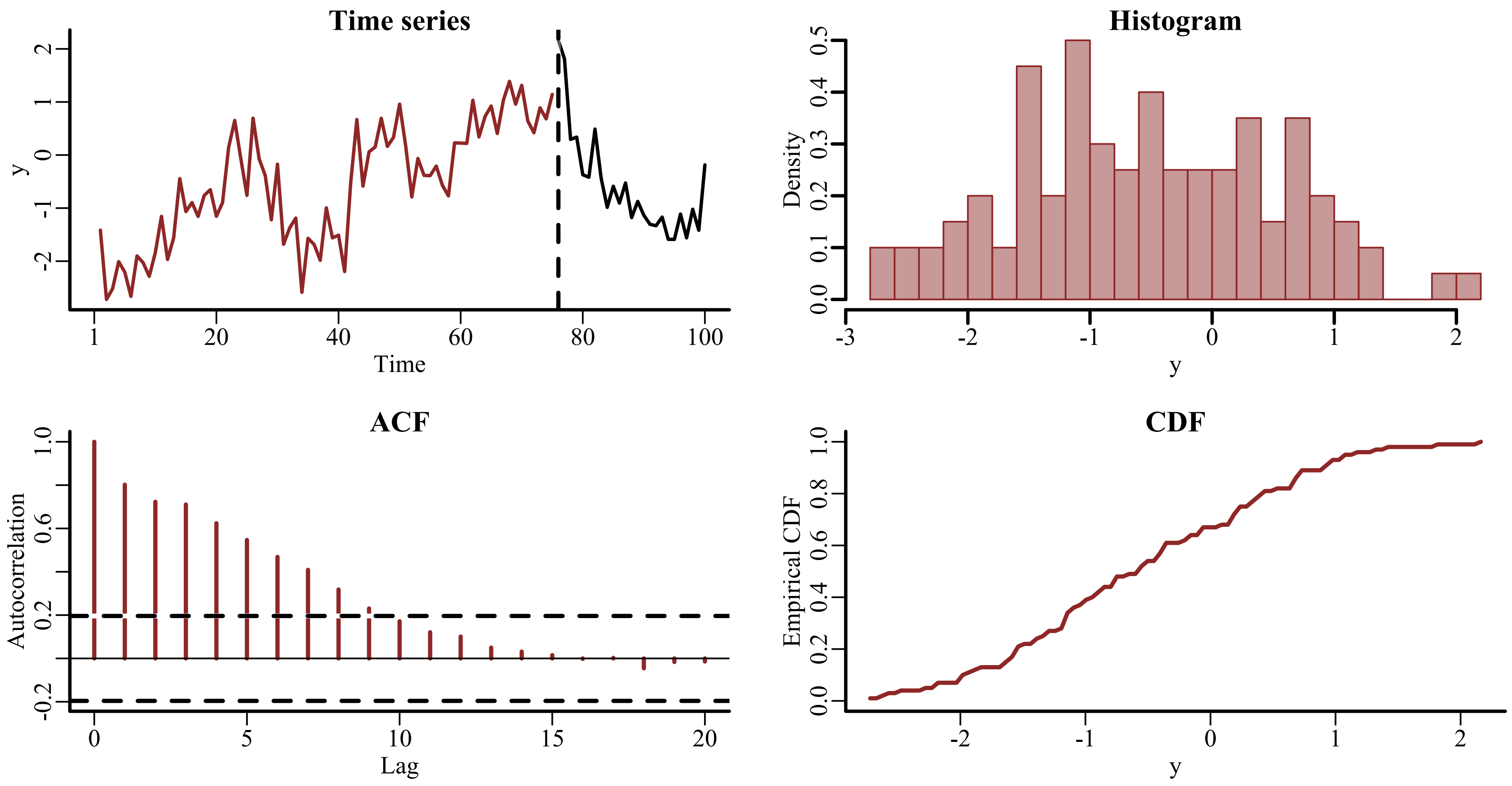
$f(x)$ are potentially nonlinear functions of the J predictors





s(x0)
Partial effect
6
3
0
-3
-6
0.00
0.25
0.50
0.75
1.00
x0
Basis: GP
te(x1,x2)
x2
1.00
0.75
0.50
0.25
0.00
x1
Basis: Tensor product
Partial effect
6
3
0
-3
-6
s(x3)
Partial effect
5
0
-5
-40
-20
0
20
x3
Basis: TPRS
s(x4)
Partial effect
4
0
-4
-2
-1
0
1
2
x4
Basis: CRS (shrink)

What's the catch?

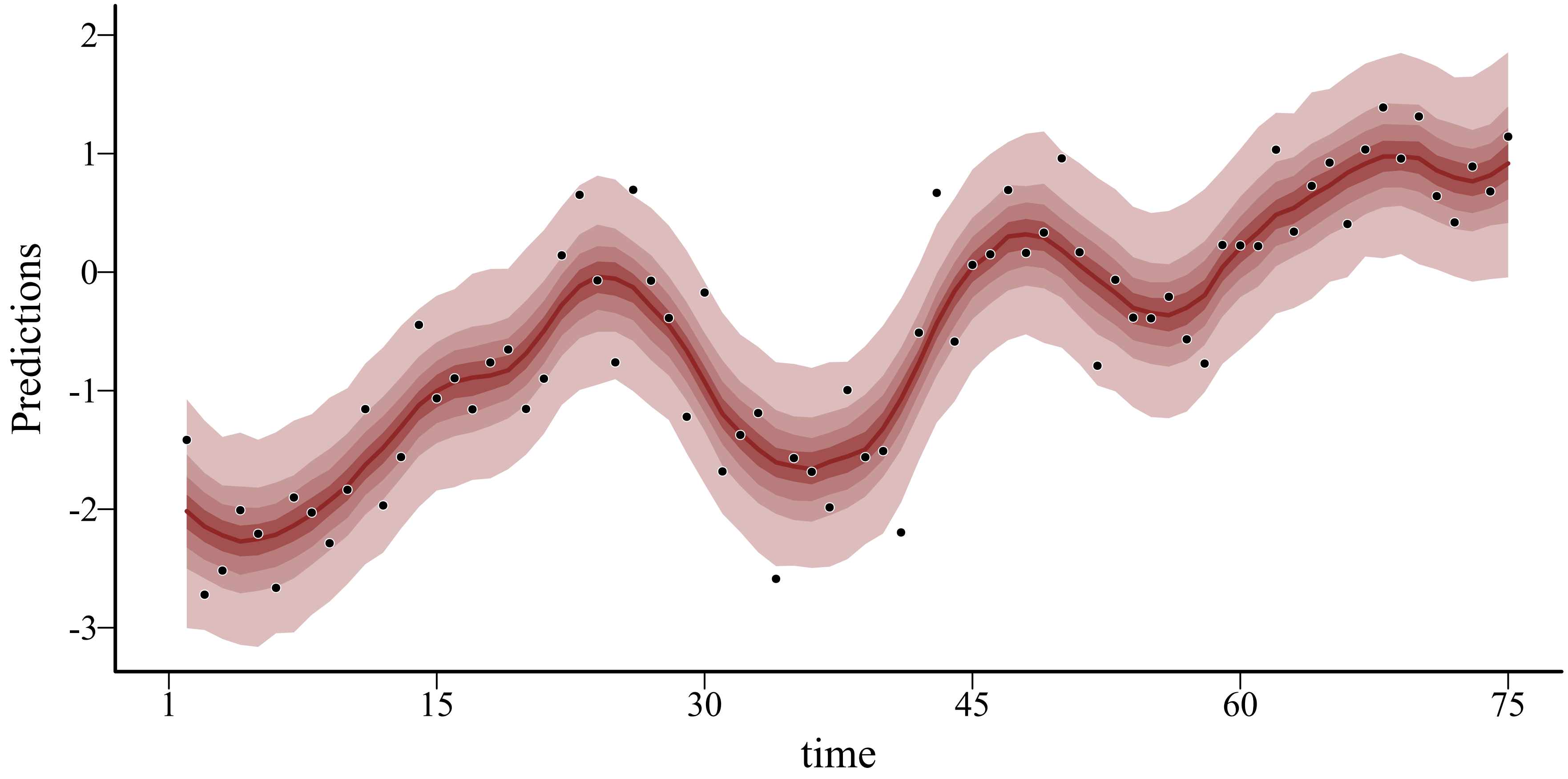


A spline of `time`

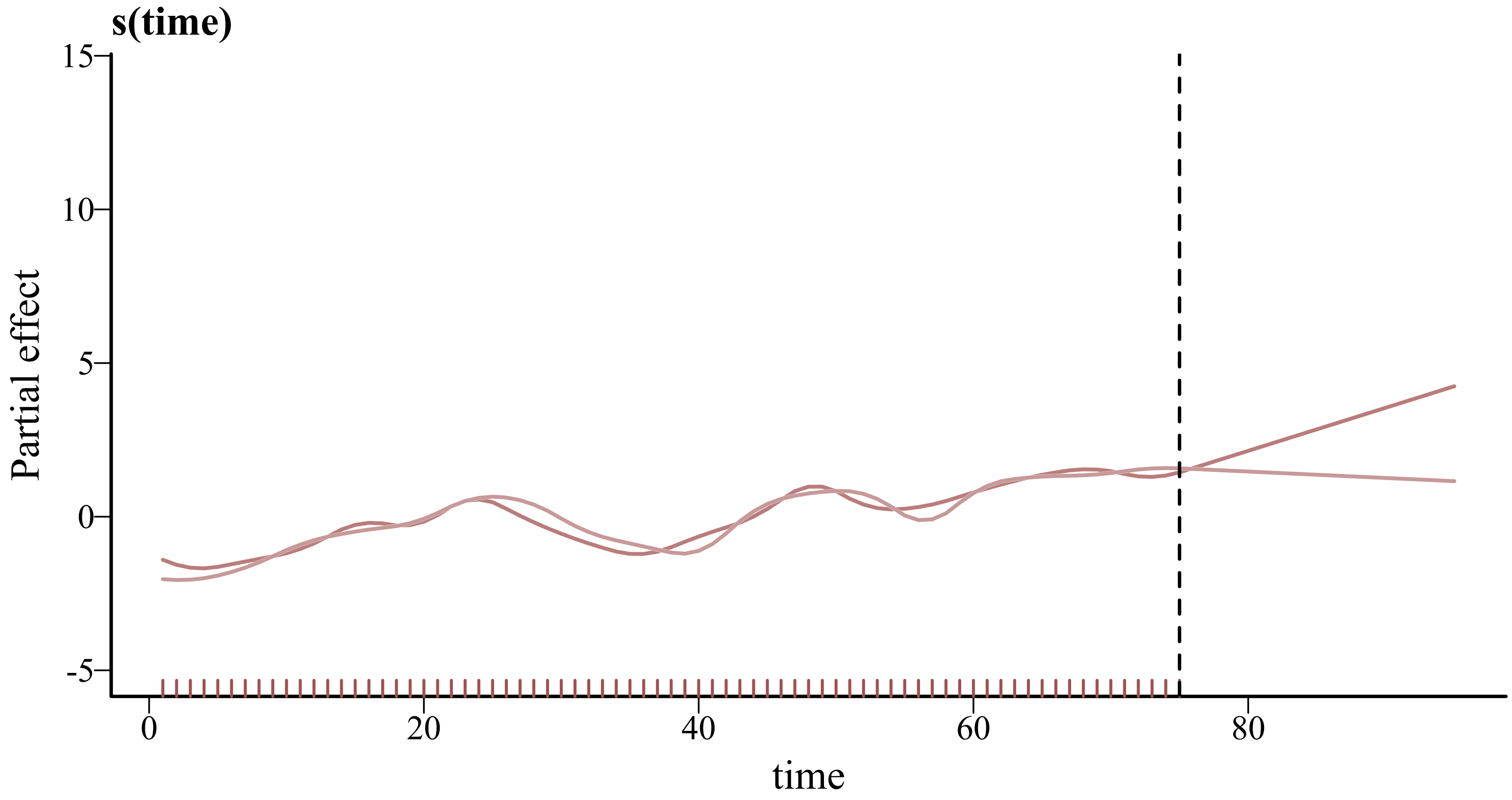
```r
library(mgcv)
model <- gam(y ~ s(time, k = 20, bs = 'bs', m = 2),
             data = data,
             family = gaussian())
```

A B-spline (`bs = 'bs'`) with `m = 2` sets the penalty on the second derivative

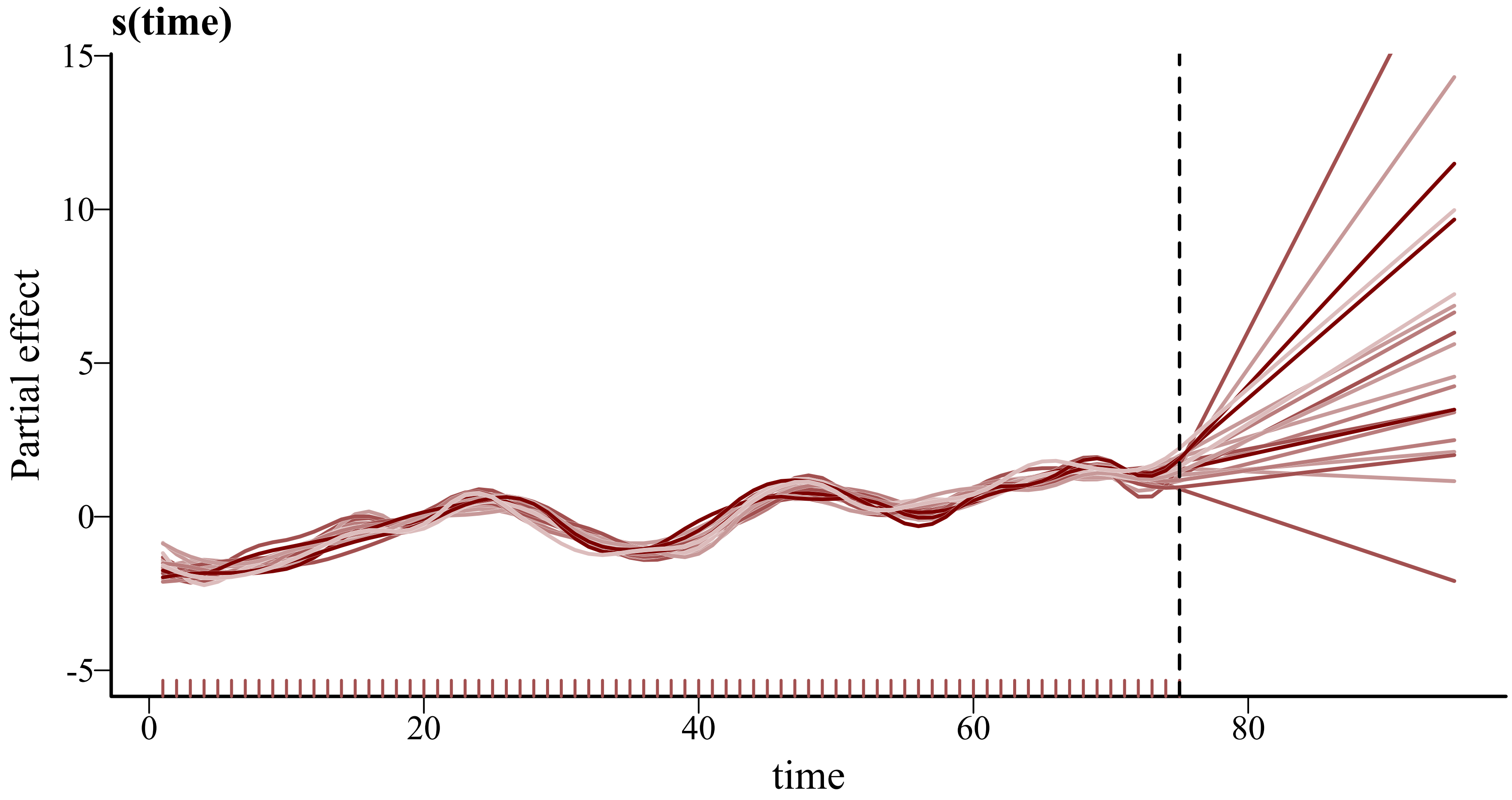
Hindcasts 😊



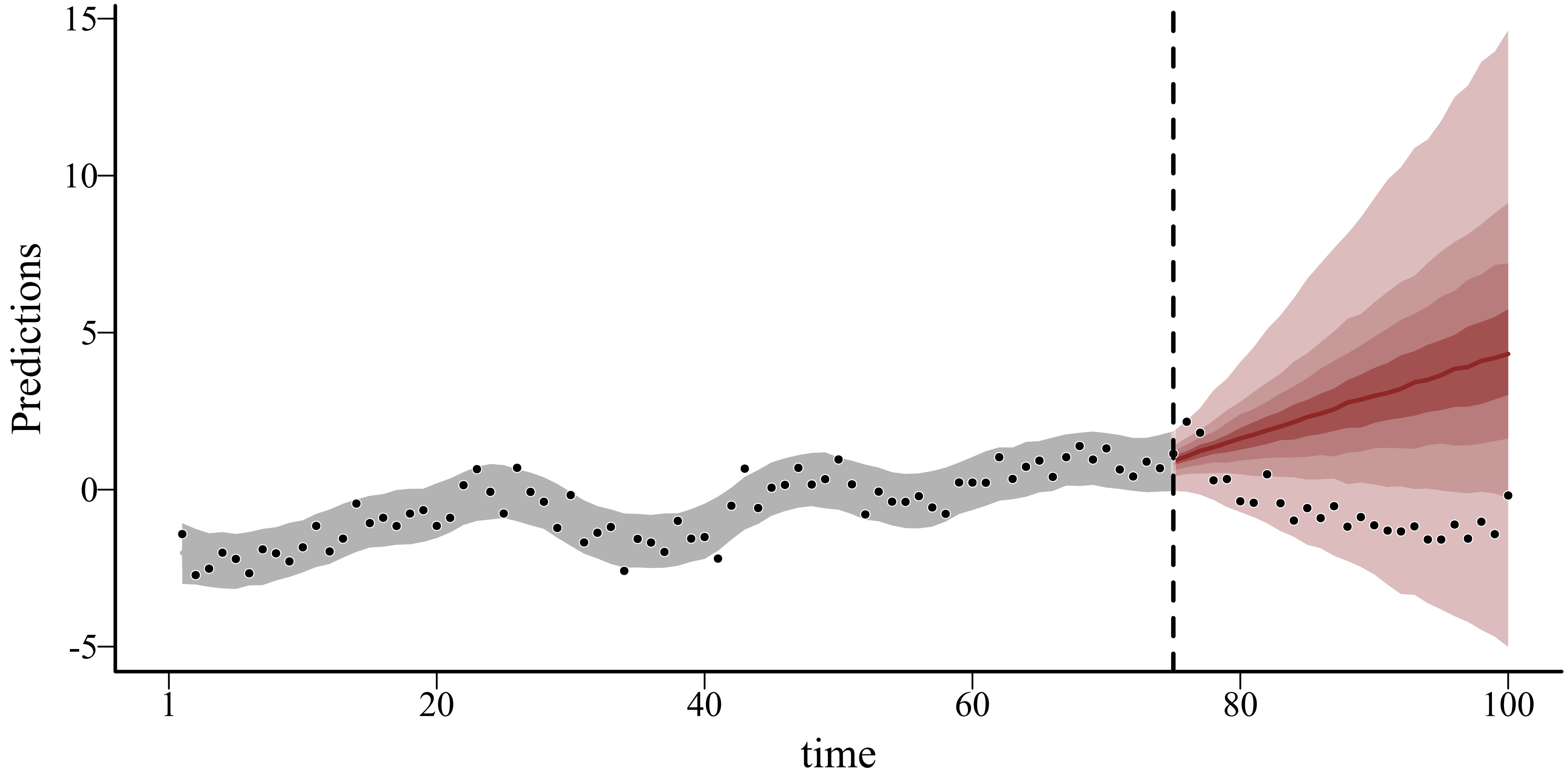
Basis functions ⇨ local knowledge



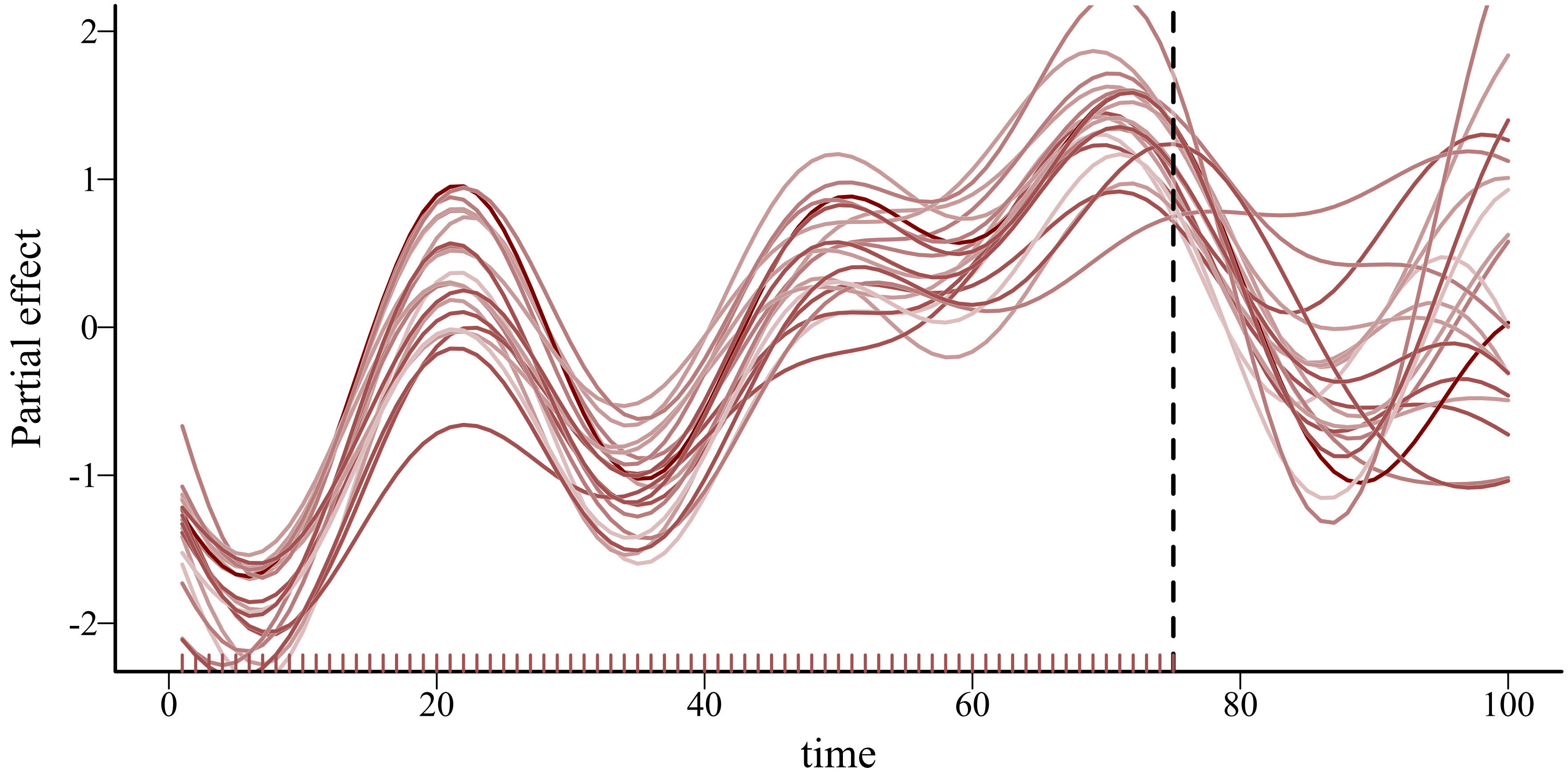
Basis functions ⇨ local knowledge



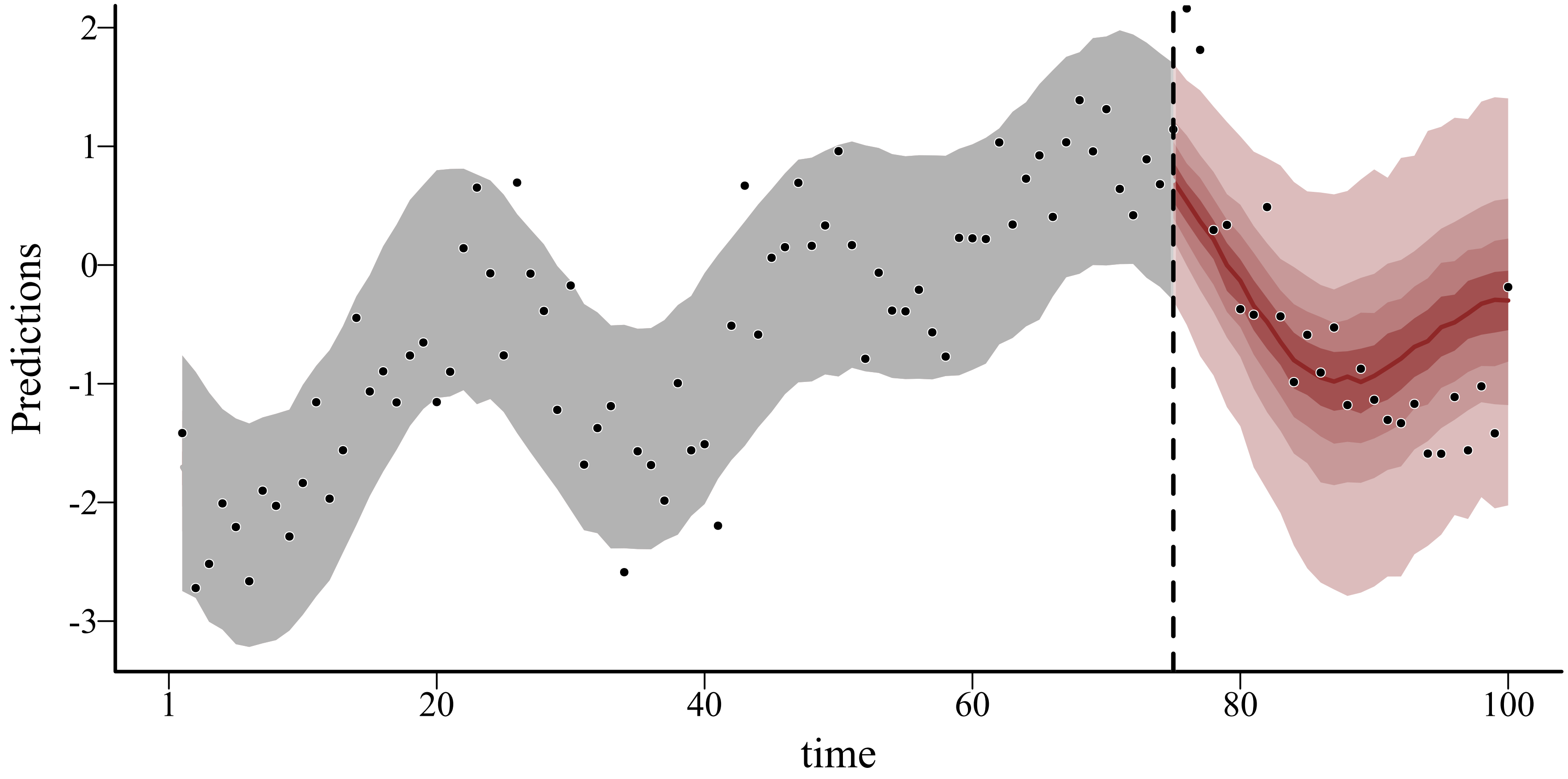
Forecasts 😭



We need *global* knowledge



We need *global* knowledge



Dynamic GAMs

$$\mathbb{E}(\boldsymbol{Y_t}|\boldsymbol{X_t}) = g^{-1}(\alpha + \sum_{j=1}^{J} f(x_{jt}) + z_t)$$

Where:

g^{-1} is the inverse of the link function

α is the intercept

$f(x)$ are potentially nonlinear functions of the J predictors

z_t is a ***latent dynamic process***

Modelling with the `mvgam` 📦

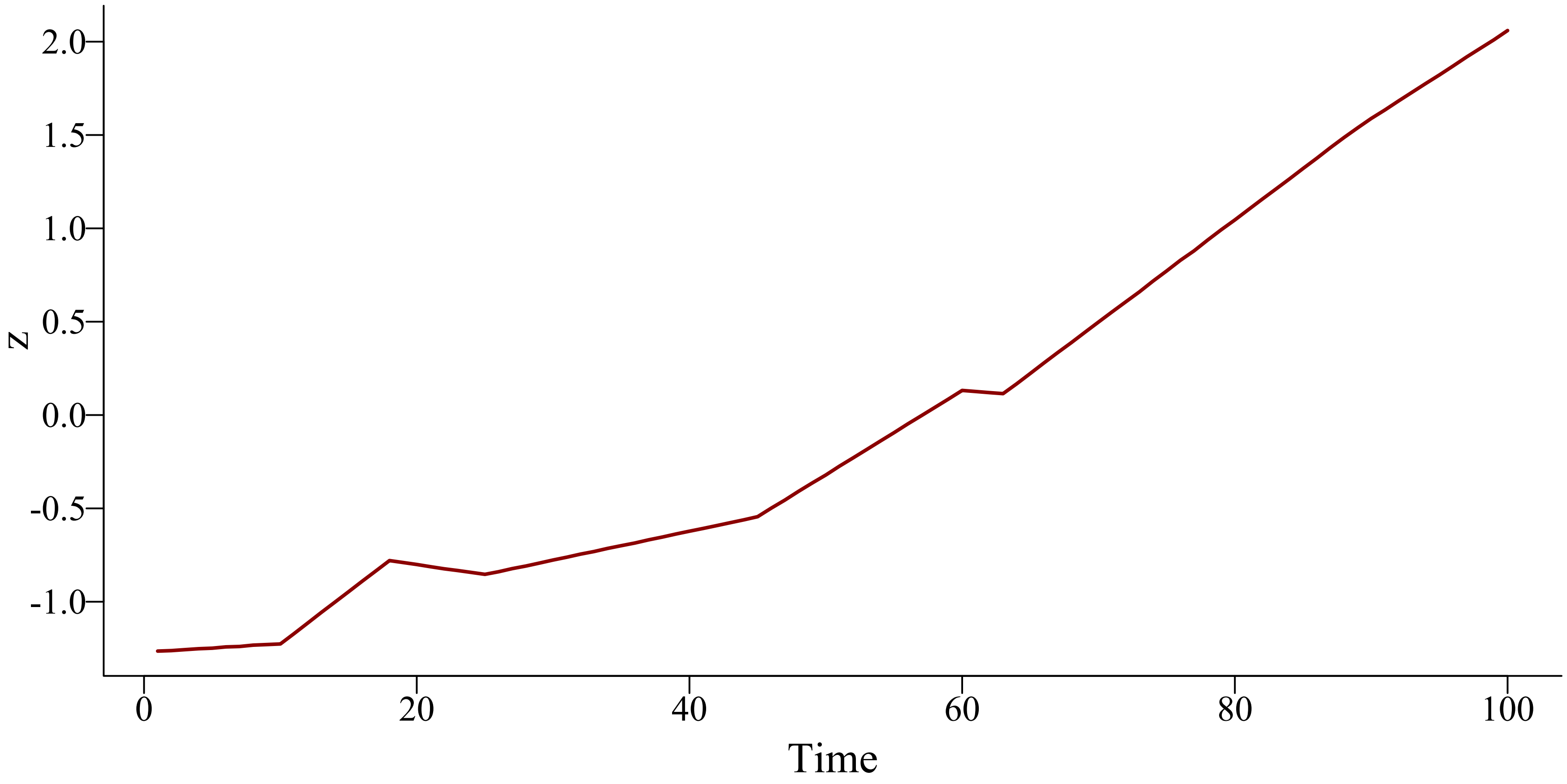
- Bayesian framework to fit Dynamic GLMs and Dynamic GAMs
  - Hierarchical intercepts, slopes and smooths
  - Latent dynamic processes
  - State-Space models with measurement error
- Built off the `mgcv` 📦 to construct penalized smoothing splines
- Familiar R formula interface
- Uni- or multivariate series from a range of response distributions
- Uses Stan for ADVI, Laplace or full Hamiltonian Monte Carlo

We can fit models that include random effects, nonlinear effects, time-varying effects and complex multidimensional smooth functions. All these effects can operate *on both process and observation models*

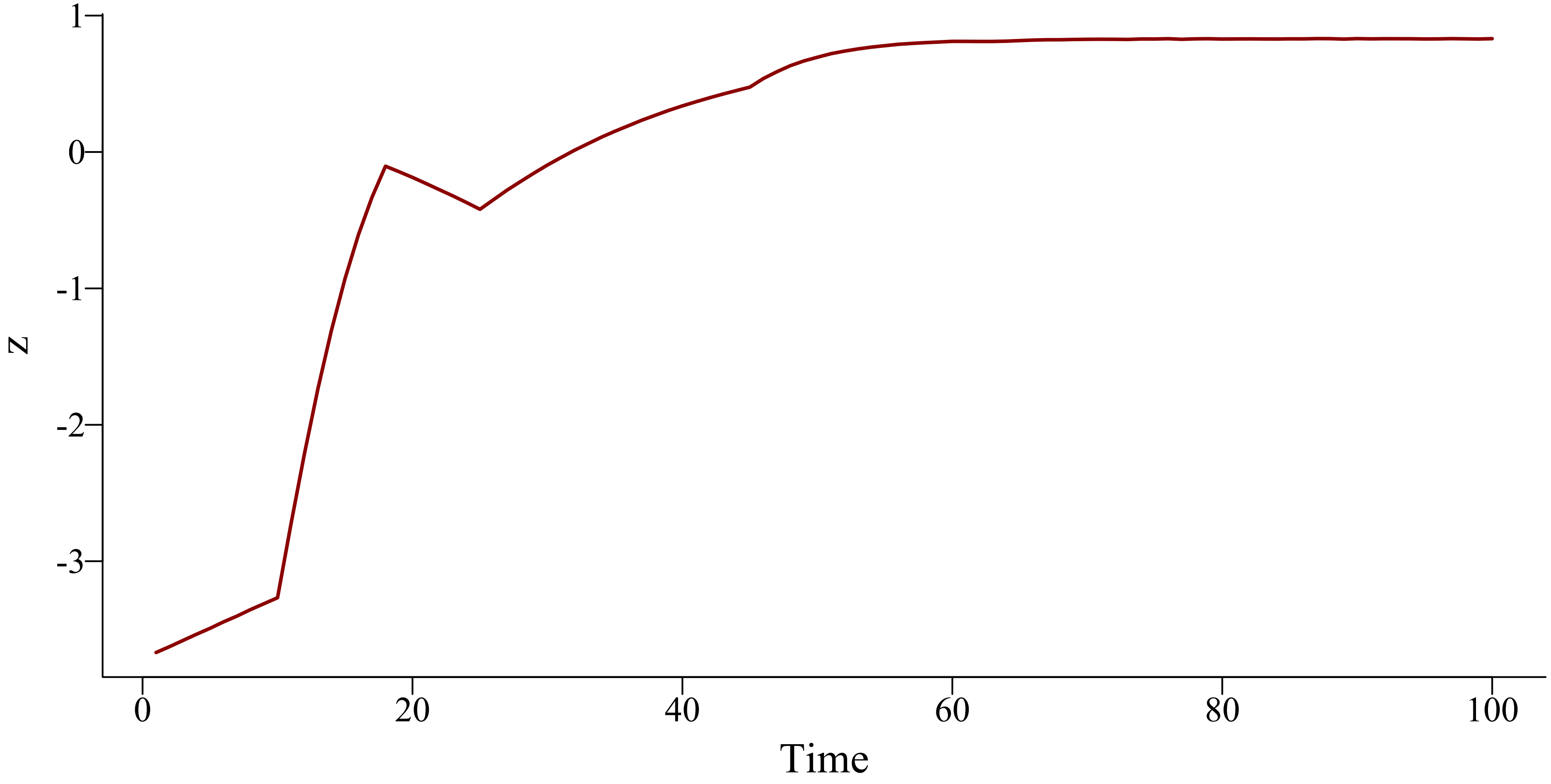
Can incorporate unobserved temporal dynamics; no need to regress the outcome on past values or resort to transformations

What kinds of dynamic processes are available in the `mvgam` 📦 ?

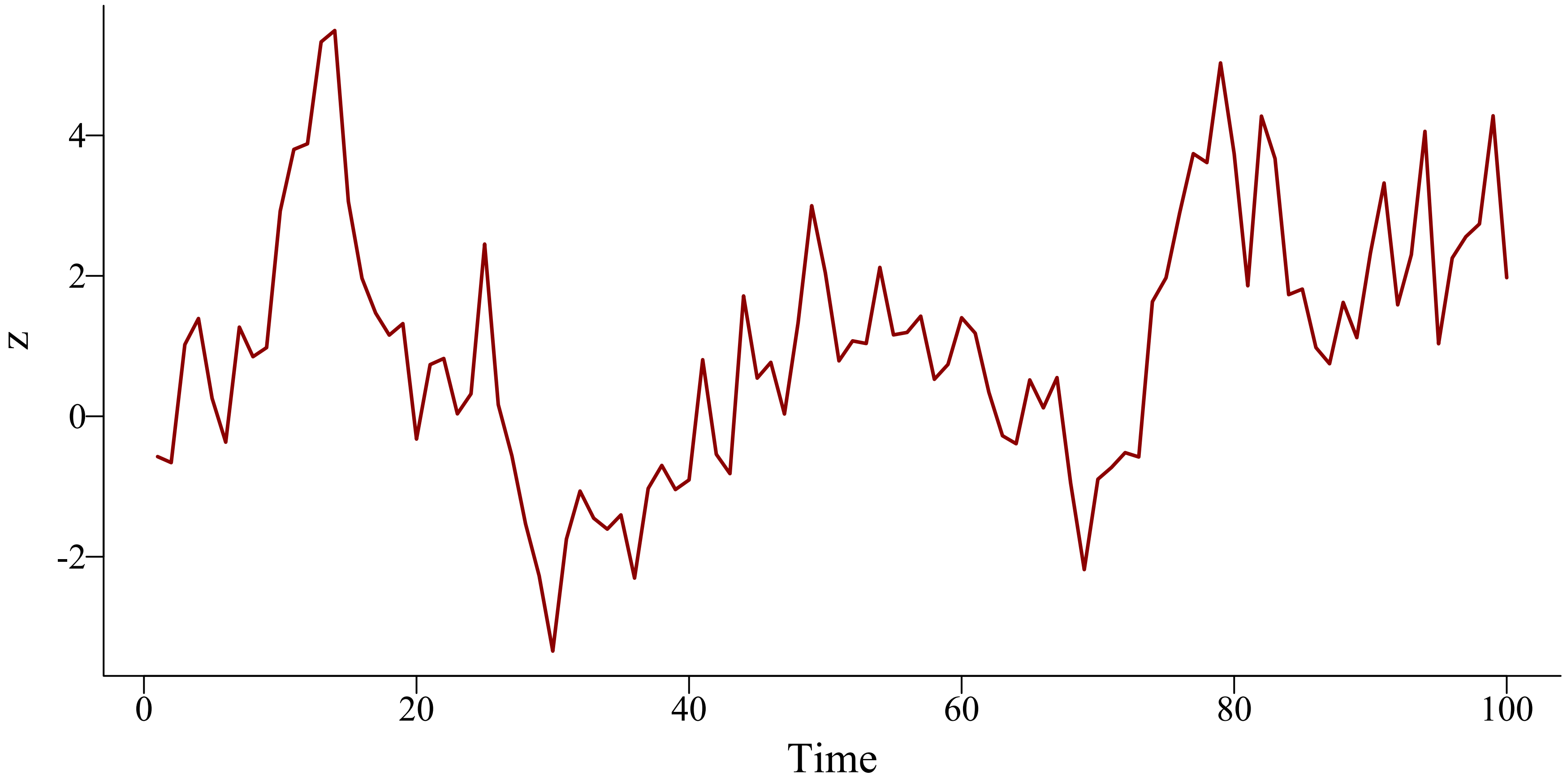
Piecewise linear...



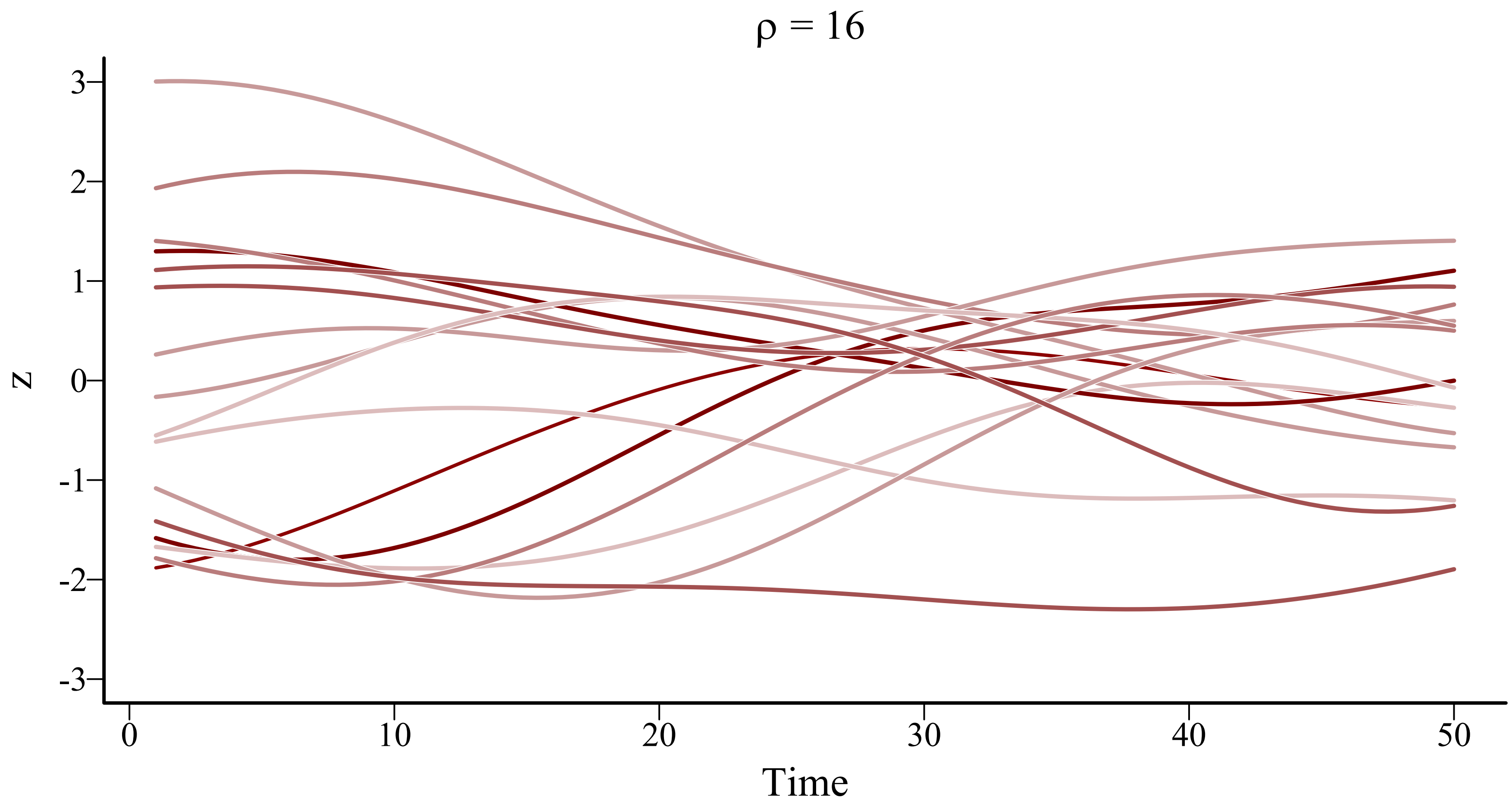
...or logistic with upper saturation



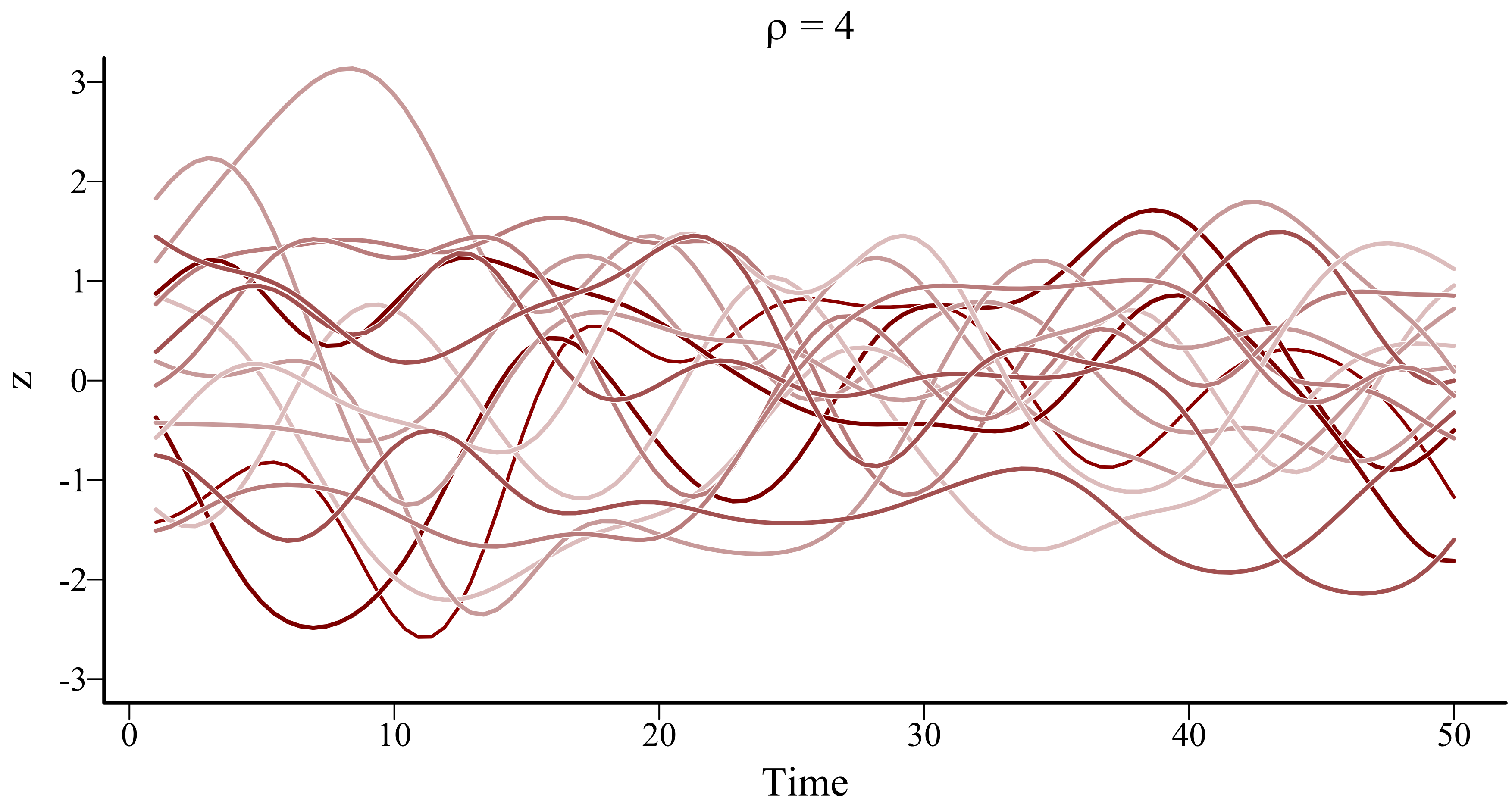
RW or ARMA(p = 1-3, q = 0-1)



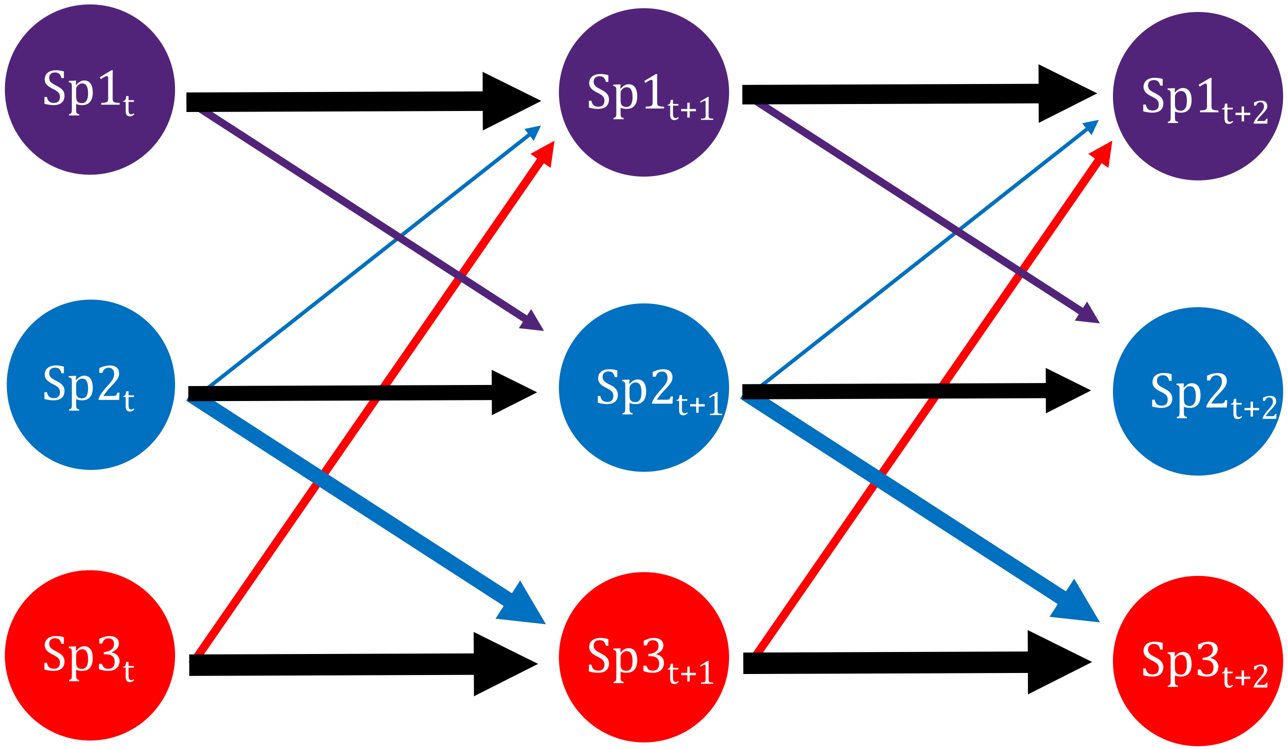
Gaussian Process...



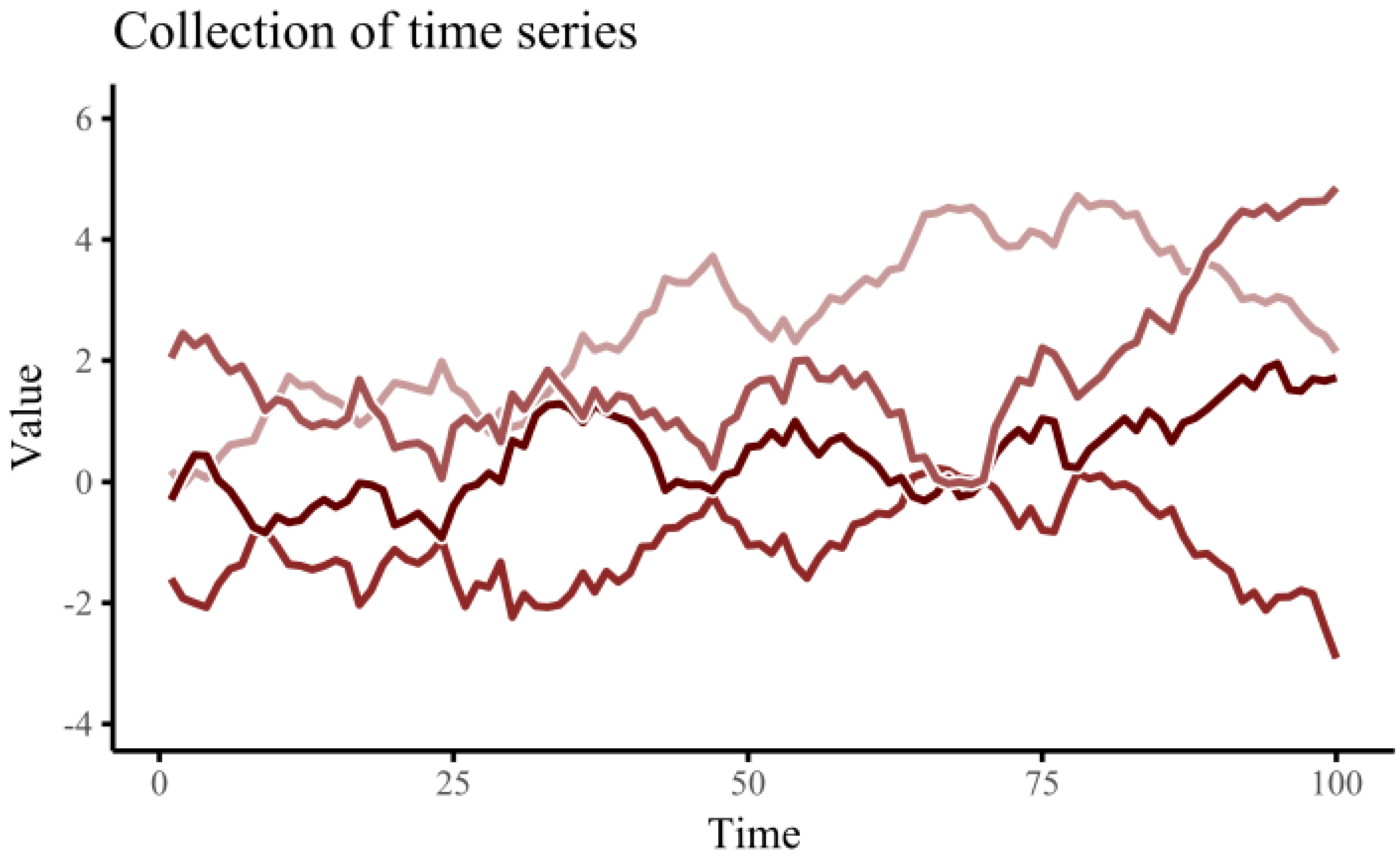
...where length scale ⇨ *memory*



VAR1 ⇨ Granger causality



Factors ⇨ induced correlations



Example of the interface

```
model <- mvgam(
  formula = y ~
    s(series, bs = 're') +
    s(x0, series, bs = 're') +
    x1 +
    gp(x2) +
    te(x3, x4, bs = c('cr', 'tp')),
  data = data,
  family = poisson(),
  trend_model = AR(p = 1, ma = TRUE, cor = TRUE),
  algorithm = 'sampling',
  burnin = 500,
  samples = 500,
  chains = 4,
  parallel = TRUE)
```

Produce all Stan code and objects

```
code(model)
```

```
## // Stan model code generated by package mvgam
## functions {
##   /* Spectral density function of a Gaussian process
##   * with squared exponential covariance kernel
##   * Args:
##   *   l_gp: numeric eigenvalues of an SPD GP
##   *   alpha_gp: marginal SD parameter
##   *   rho_gp: length-scale parameter
##   * Returns:
##   *   numeric values of the GP function evaluated at l_gp
##   */
##   vector spd_cov_exp_quad(data vector l_gp, real alpha_gp, real rho_gp) {
##     int NB = size(l_gp);
##     vector[NB] out;
##     real constant = square(alpha_gp) * (sqrt(2 * pi()) * rho_gp);
##     real neg_half_lscale2 = -0.5 * square(rho_gp);
##     for (m in 1 : NB) {
##       out[m] = constant * exp(neg_half_lscale2 * square(l_gp[m]));
##     }
##     return out;
```

Workflow in `mvgam`

Fit models with splines, GPs, and multivariate dynamic processes to sets of time series; use informative priors for effective regularization

Use posterior predictive checks and Randomized Quantile residuals to assess model failures

Use `marginaleffects` to generate interpretable (and reportable) model predictions

Produce probabilistic forecasts

Evaluate forecasts from using proper scoring rules

More resources

Cheatsheet ⇨ Overview of `mvgam`

Vignette ⇨ Introduction to the package

Vignette ⇨ Shared latent process models

Vignette ⇨ Time-varying effects

Vignette ⇨ Multivariate State-Space models

Motivating publication ⇨ Clark & Wells 2023 *Methods in Ecology and Evolution*

Relevant links

`mvgam` 📦 website

nicholasjclark

slides for this talk

personal website

n.clark@uq.edu.au