Generalized additive (mixed) models

18 Nov 2024

```r
library(mgcv)
library(gratia)
library(tidyverse)
```

Additive models

- generally a way to specify more complex (smooth) terms based on *individual* covariates: $\mu = \beta_0 + f_1(x_1) + f_2(x_2) + \ldots$
- lots of ways to generate $f_i(x_i)$: kernel estimators, locally weighted polynomials, … see T. J. Hastie and Tibshirani (1990), T. Hastie, Tibshirani, and Friedman (2009) (*backfitting algorithm* etc.)
- we will focus on the approach of Simon N. Wood (2017), which is in some ways more restricted (everything is done explicitly via bases + latent Gaussian variables)

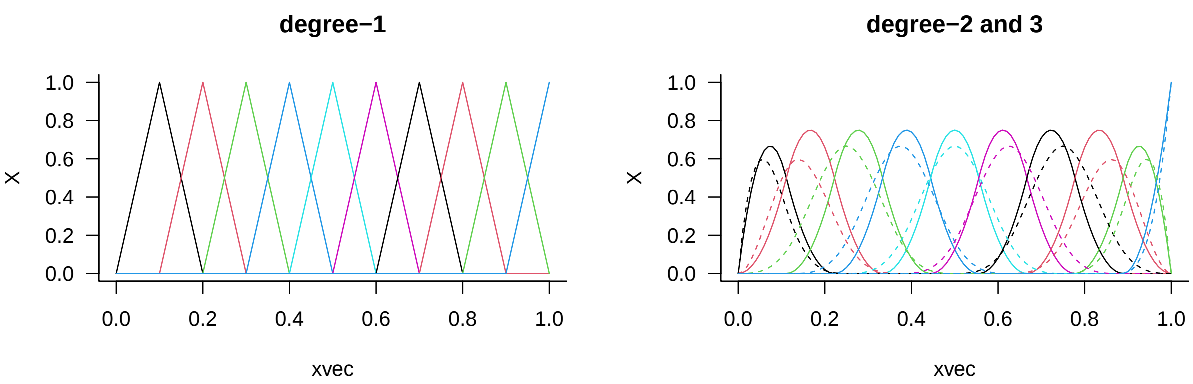
Basis expansions

- (theoretically) infinitely expandable
- e.g. polynomials ('regular'/raw, orthogonal, Legendre, Hermite)
- wavelet, Fourier
- splines: **piecewise polynomial** with continuity/smoothness constraints

Spline degree

```r
xvec <- seq(0, 1, length.out = 101)
sfun <- function(d = 3, type = c("bs", "ns"), lty = 1, ...) {
    type <- match.arg(type)
    X <- switch(type,
```

```r
                 bs = splines::bs(xvec, df = 10, degree = d),
                 ns = splines::ns(xvec, df = 10) ## only cubic
                 )
    par(bty = "l", las = 1)
    matplot(xvec, X, type = "l", lty = lty, ...)
}
par(mfrow=c(1,2))
sfun(d = 1, main = "degree-1")
sfun(d = 2, main = "degree-2 and 3")
sfun(d = 3, add = TRUE, lty = 2)
```



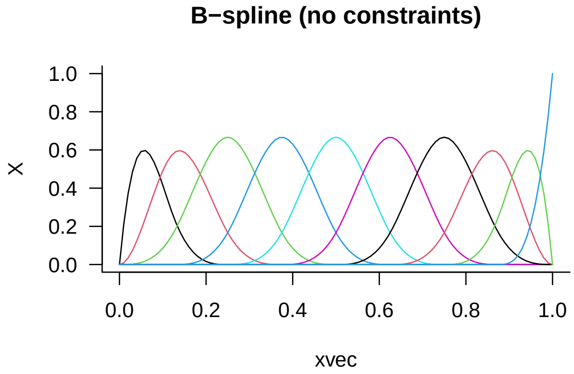
spline terminology

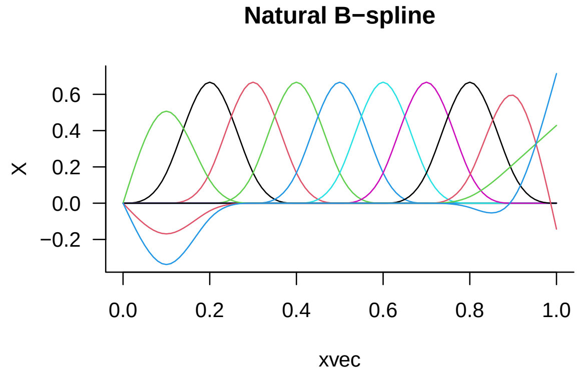
- **knots**: breakpoints (boundary, interior)
- order-M (ESL): continuous derivatives up to order $M-2$ (cubic, $M=4$)
- typically $M = 1, 2, 4$
- number of knots = df (degrees of freedom) -1 -intercept

Spline choices

- continuous derivatives up to $d-1$
- truncated polynomial basis (simple)
- B-splines: complex, but *minimal support*/maximum sparsity
- *natural* splines: extra constraint, derivatives > 1 vanish at boundaries

```r
par(mfrow = c(1,2))
sfun(main = "B-spline (no constraints)")
sfun(type = "ns", main = "Natural B-spline")
```





choosing knot locations

- generally not that important: evenly spaced, *or* evenly spaced based on quantiles

choosing basis dimension

- in principle could expand dimension to match total number of points (*interpolation spline*)
- ... but that would overfit
- AIC, adjusted R^2, cross-validation ...

smoothing splines

- as many knots as data points
- plus squared-second-derivative ("wiggliness") penalty

$$\text{RSS} + \lambda \int (f''(t))^2 \, dt$$

- defined on an infinite-dimensional space
- minimizer is a (natural?) cubic spline with knots at x_i

$$(\mathbf{y} - \mathbf{Z}\mathbf{b})^\top (\mathbf{y} - \mathbf{Z}\mathbf{b}) + \lambda \mathbf{b}^\top \Omega \mathbf{b}$$

with $\{\Omega\}_{jk} = \int \mathbf{Z}_j''(t) \mathbf{Z}_k''(t) \, dt$

- **generalized** ridge regression: penalize by $\lambda \Omega_N$ rather than λI
- could use same data augmentation methods as before except that now we use $\sqrt{\lambda} C$ where C is a matrix, and the "square root" (Cholesky factor) of Ω_N

See Simon N. Wood (2017), Perperoglou et al. (2019)

generalized cross-validation score

- very close to AIC
- Larsen (2015), Golub, Heath, and Wahba (1979)
- minimize RSS$/(\mathrm{Tr}(\mathbf{I}-\mathbf{S}(\lambda)))^2$, where S is "a rotation-invariant version of PRESS [predicted residual error sum of squares]" ($\sum(e_i/(1-h_{ii}))^2$)
- replace RSS with approximation of deviance,

  $$||\sqrt{\mathbf{W}}(\mathbf{z}-\mathbf{X}\beta)||^2$$

  for generalized (non-Gaussian) models

connection to mixed models

- note that $\lambda\mathbf{b}^\top\Omega\mathbf{b}$ is equivalent to $(1/\sigma^2)\mathbf{b}^\top\Sigma'^{-1}\mathbf{b}$; if Σ' is a *scaled* covariance matrix (i.e. $\Sigma=\sigma^2\Sigma$), then this is the core of the MVN log-likelihood $\log L(\mathbf{b}|\Sigma)$ (all we're missing is a factor of $\mathrm{Det}(\Sigma)^{-1/2}$ and a normalization constant)
- So we can fit this with any of the mixed model machinery, provided we can set up the correct covariance matrix

ML criterion, REML criterion

- treat spline smoothing as a *mixed model* problem
- spline (penalized) parameters are $\mathbf{b}$
- $y|u \sim N(\mathbf{X}\beta+\mathbf{Z}\mathbf{b},\sigma^2\mathbf{I})$; $\mathbf{b} \sim N(0,(\sigma^2/\lambda)\mathbf{W}^{-1})$ where the $\mathbf{W}$ is the penalty matrix
- corresponds to minimizing $||\mathbf{y}-\mathbf{X}\beta-\mathbf{Z}\mathbf{b}||^2+\lambda\mathbf{b}^\top\mathbf{W}\mathbf{b}$
- REML: "fixed effects are viewed as random effects with improper uniform priors and are integrated out" (Wood 2011)
- Laplace approximation
- slower but generally preferred now

practical stuff

- Simon Wood is insanely smart, and `mgcv` is insanely powerful and flexible
- gratia package (named after Grace Wahba
- available 'smooths' (bases + penalty terms): look for strings of the form `smooth.construct.*.smooth.spe`
- although you can *theoretically* have as many knots as data points, fewer is often good enough/computationally efficient

Available bases (using `apropos("smooth.construct")`):

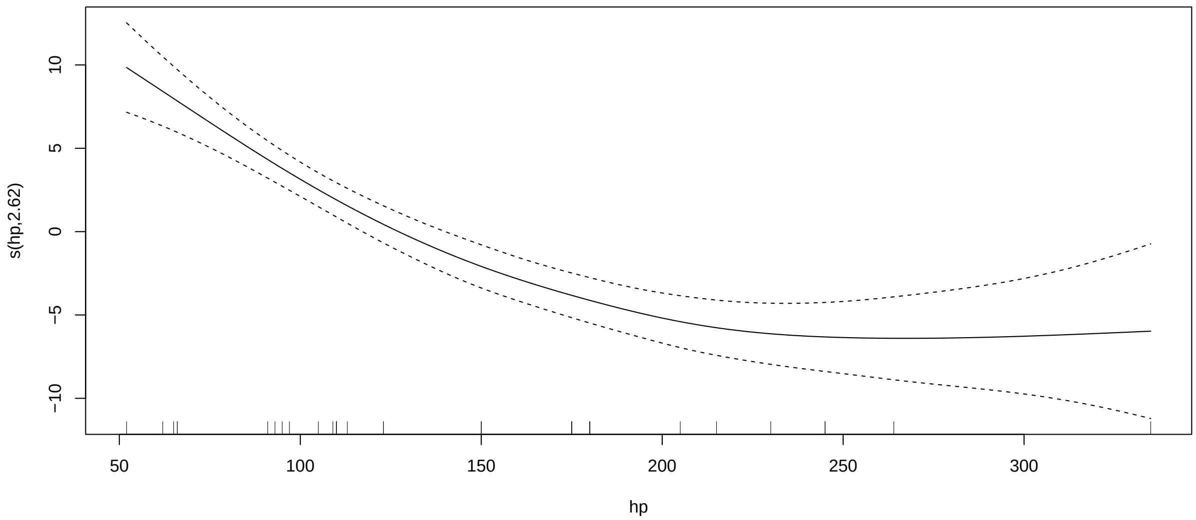
```
 [1] "ad"     "bs"     "cc"     "cp"     "cr"     "cs"     "ds"     "gp"
 [9] "mrf"    "ps"     "re"     "sf"     "so"     "sos"    "sw"     "t2"
[17] "tensor" "tp"     "ts"
```

```r
g1 <- gam(mpg ~ s(hp), data = mtcars)
summary(g1)
```

```

Family: gaussian
Link function: identity

Formula:
mpg ~ s(hp)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  20.0906     0.5487   36.62   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
        edf Ref.df     F p-value
s(hp) 2.618  3.263 26.26  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.735   Deviance explained = 75.7%
GCV = 10.862  Scale est. = 9.6335    n = 32
```

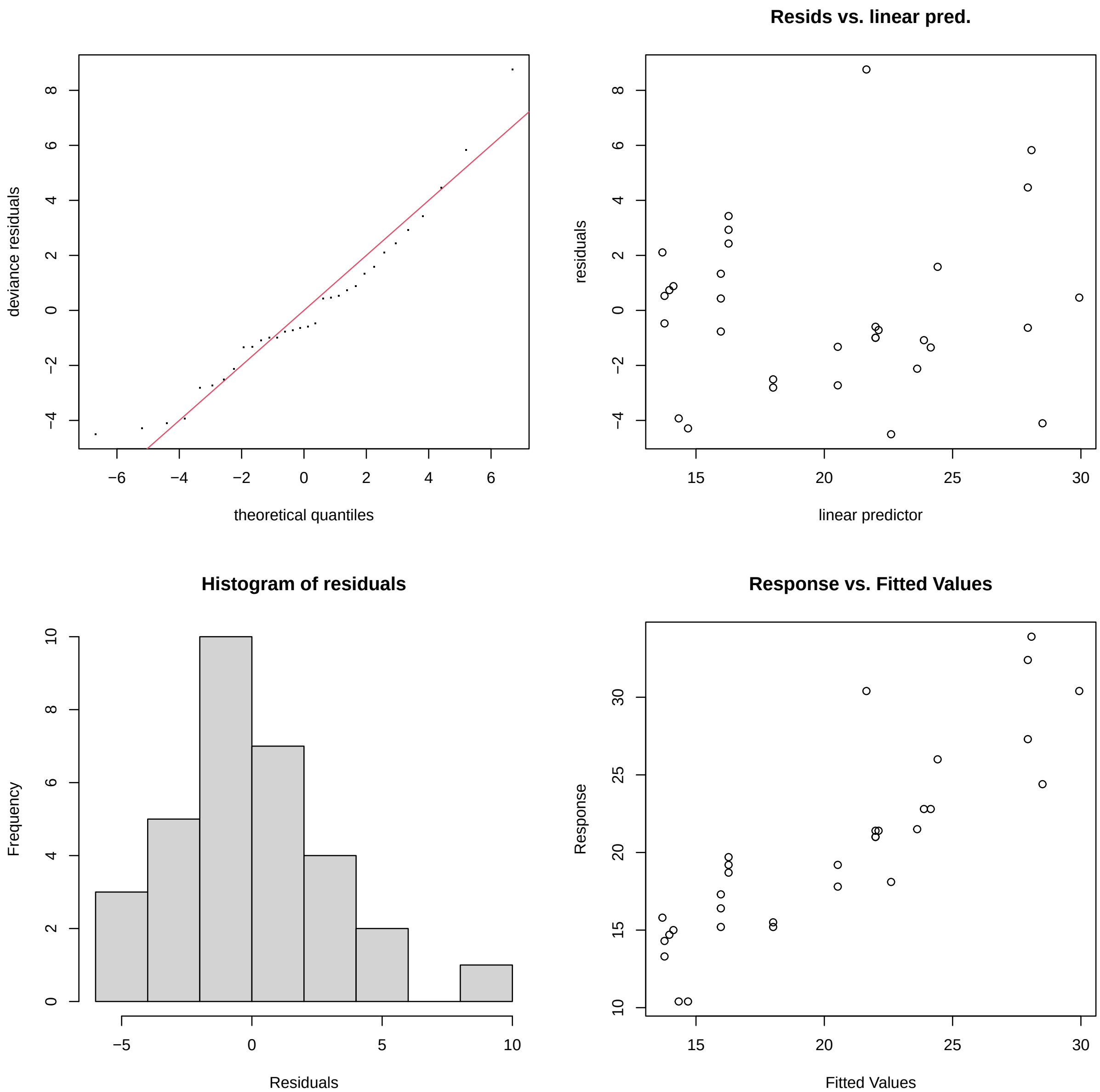
Plot:

```r
plot(g1)
```



Check:

```
gam.check(g1)
```

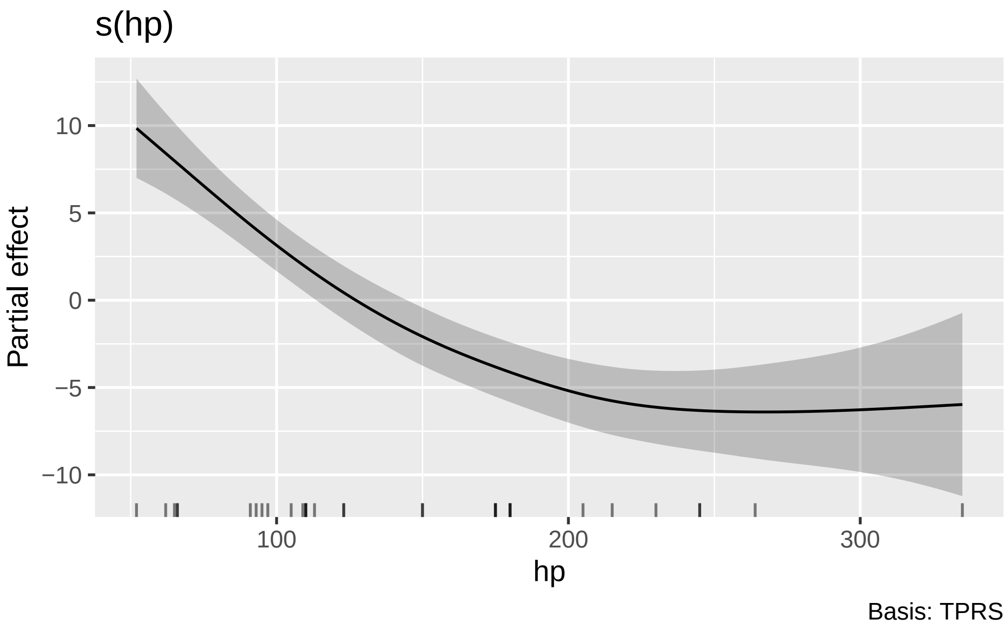



```
Method: GCV   Optimizer: magic
Smoothing parameter selection converged after 4 iterations.
The RMS GCV score gradient at convergence was 4.290111e-05 .
The Hessian was positive definite.
Model rank =  10 / 10

Basis dimension (k) checking results. Low p-value (k-index<1) may
indicate that k is too low, especially if edf is close to k'.
```

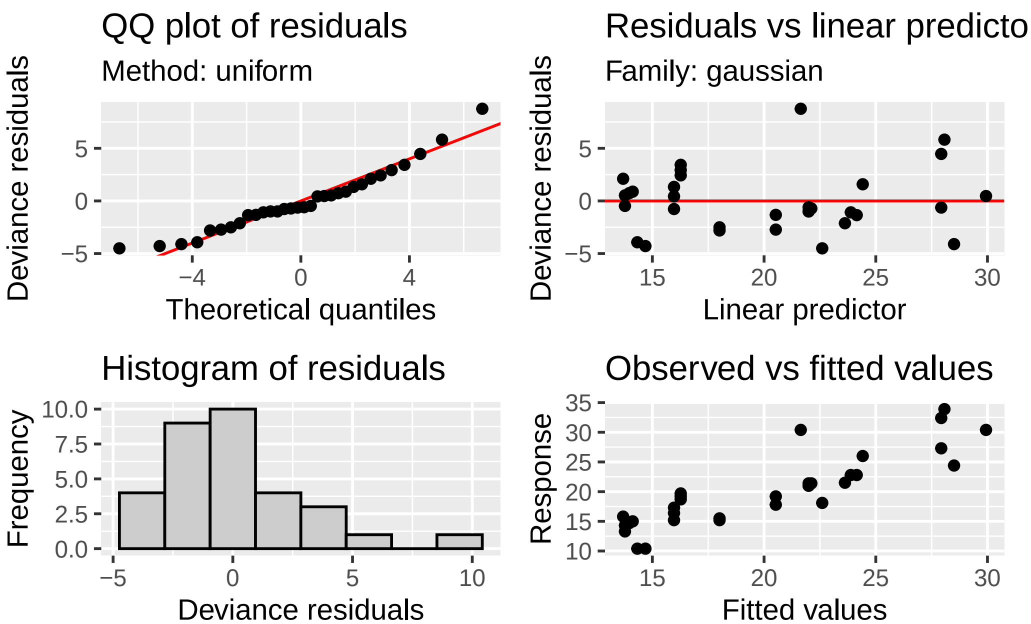
```
        k'  edf k-index p-value
s(hp) 9.00 2.62    0.87    0.18
```

The `gratia` package has prettier versions:

```r
draw(g1)
```

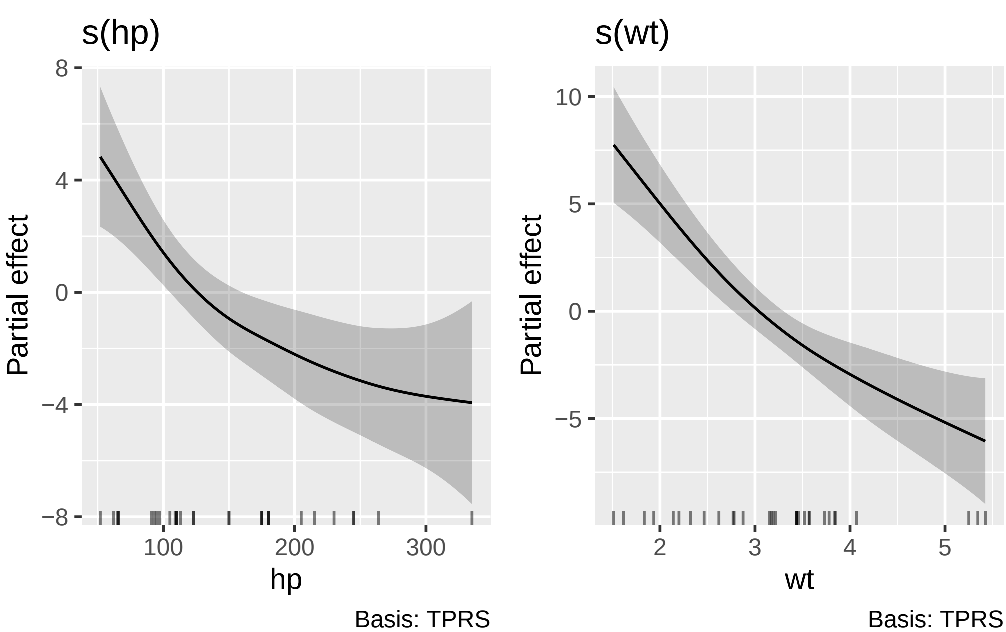



```r
appraise(g1)
```



concurvity (analogous to "collinearity"): CV question, Ramsay, Burnett, and Krewski (2003); rule of thumb is that a value of (0.3? 0.5? 0.8?) suggests trouble ...

```
g3 <- gam(mpg ~ s(hp) + s(wt), data = mtcars)
draw(g3)
```

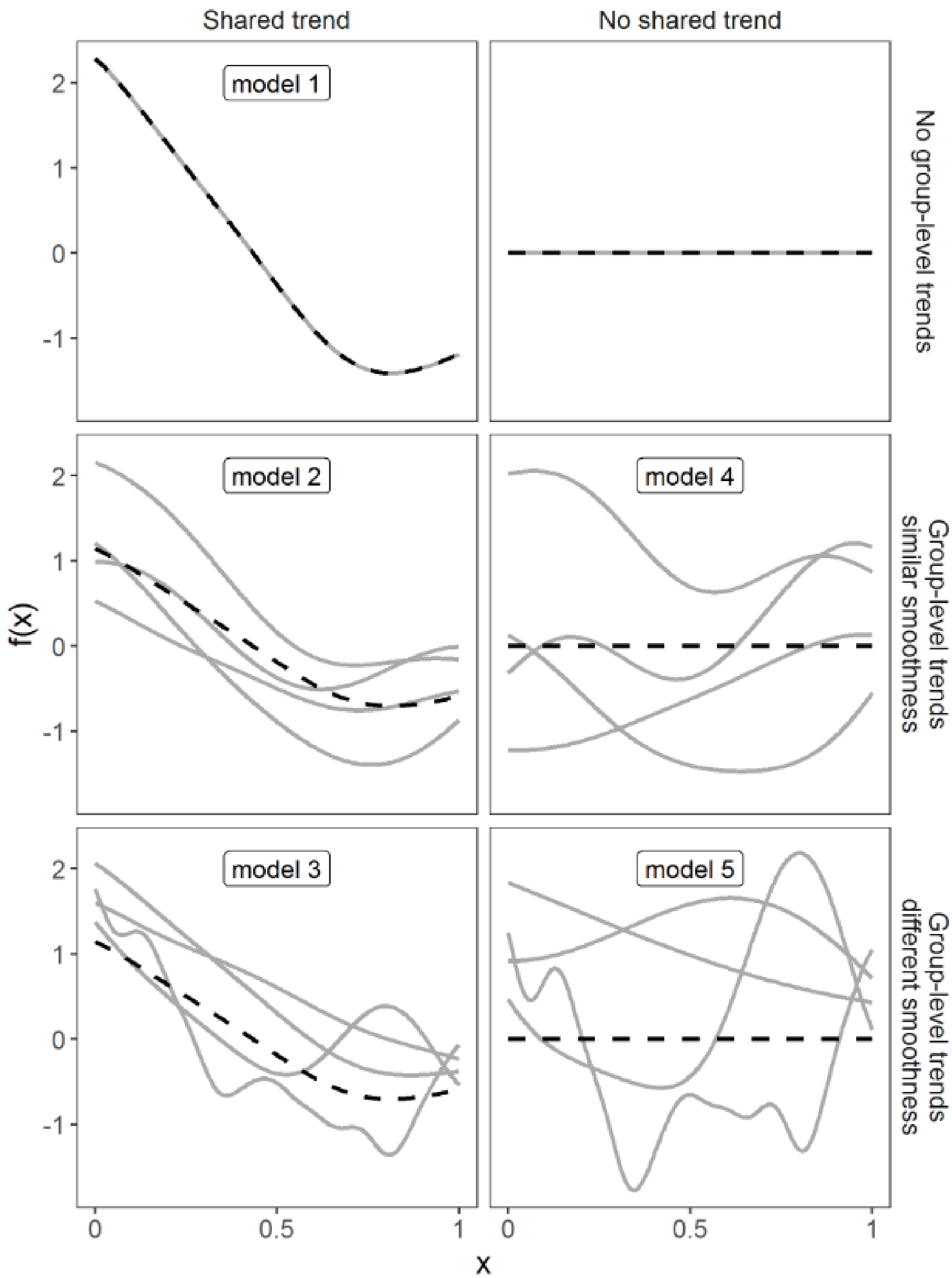


```r
concurvity(g3)
```

```
                 para     s(hp)     s(wt)
worst    2.534415e-18 0.9575001 0.9575001
observed 2.534415e-18 0.6883189 0.6968782
estimate 2.534415e-18 0.4784240 0.7978191
```

Many options: simple random effects (`bs = "re"`); *cyclic* splines (make $x(0) = x(T)$; `bs="cc"`) ; multidimensional splines (thin-plate, *tensor product* (`te()`); spherical (*Duchon*) splines (`bs = "sos"`); Markov random fields (`bs = "mrf"`); Gaussian processes (`bs = "gp"`); splines by category (`by=` argument); constrained splines (`scam` package, Pya and Wood (2015)); *soap film* splines; etc etc etc etc ...

The paper by Pedersen et al. (2019) on **hierarchical splines** is especially important.


Shared trend
No shared trend
model 1
model 2
model 3
model 4
model 5
No group-level trends
Group-level trends
similar smoothness
Group-level trends
different smoothness
f(x)
x
2
1
0
-1
0
0.5
1