Flexible models for clustered data

Helen Ogden, University of Southampton

APBG mid year meeting, 29 July, 2022

A simple clustered data modelling setup

Setup:

- observations y_i, $i = 1, \ldots, n$.
- each observation i belongs to a cluster c_i.
- also have a continuous covariate x_i for each i

We model how the distribution of response Y_i depends on x_i and c_i.

Suppose there are n_j observations in cluster j, and d clusters in total.

Response distributions

Given a predictor $\eta(x_i, c_i)$, we will model Y_i as in a GLM:

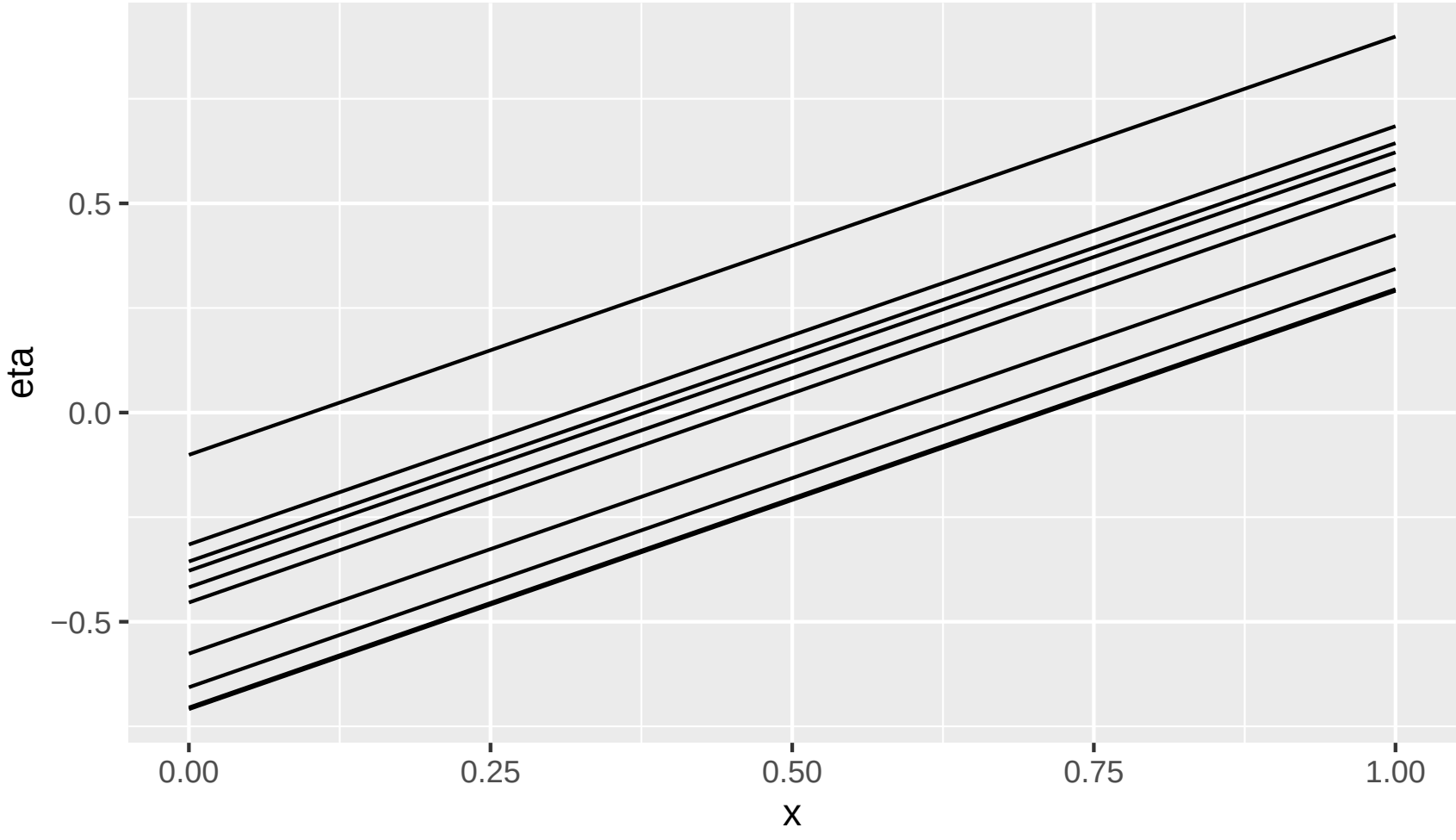
e.g.

$$
\begin{aligned}
Y_i &\sim N(\mu_i, \sigma^2), \quad \mu_i = \eta(x_i, c_i), \\
Y_i &\sim \text{Bernoulli}(\mu_i), \quad \text{logit}(\mu_i) = \eta(x_i, c_i), \\
Y_i &\sim \text{Poisson}(\mu_i), \quad \log(\mu_i) = \eta(x_i, c_i).
\end{aligned}
$$

Focus here on normal responses, for simplicity.

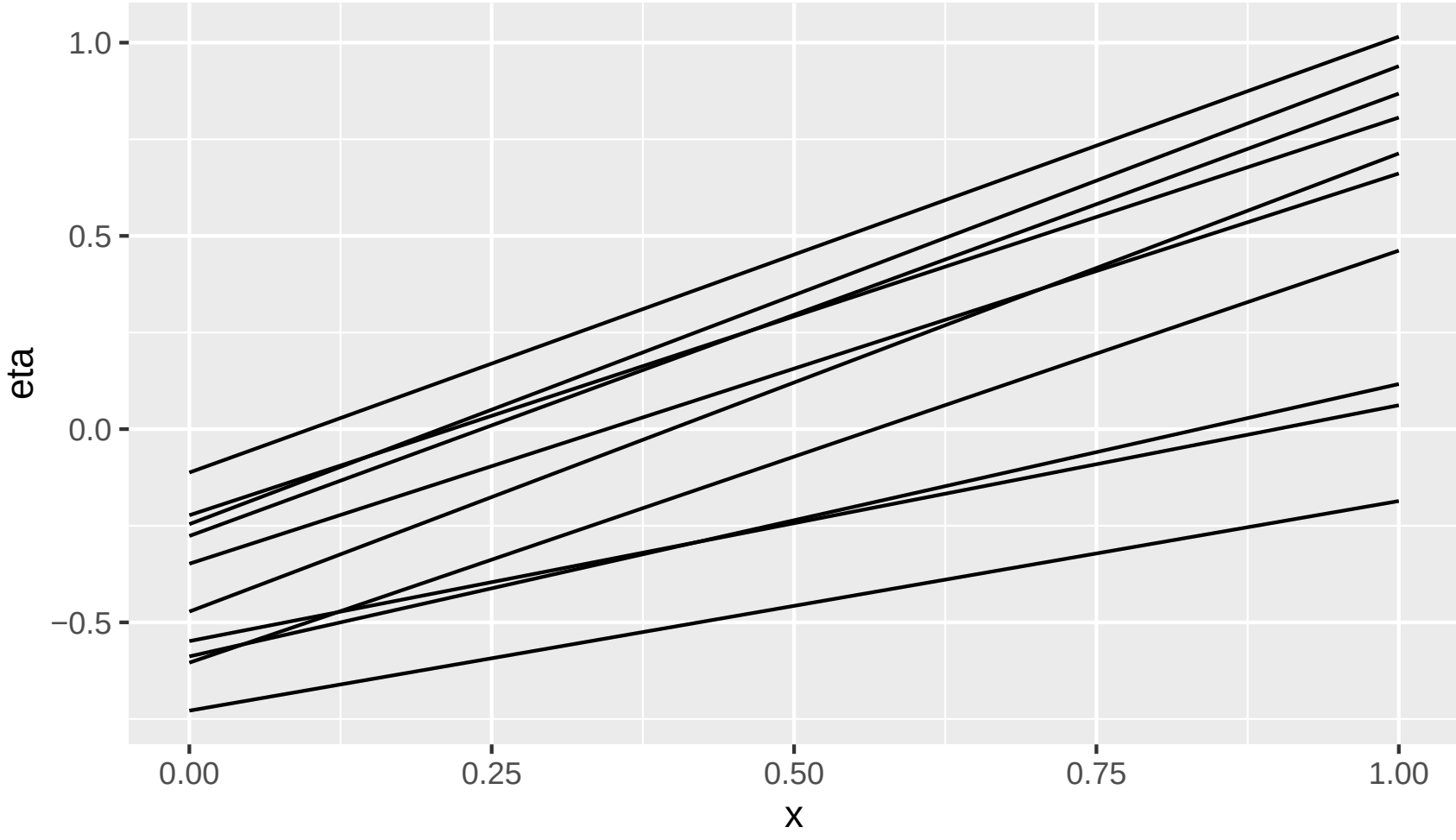
GLMM (random intercept)

$$\eta(x, c) = \beta_0 + \beta_1 x + u_c, \;\; u_c \sim N(0, \sigma_u^2)$$



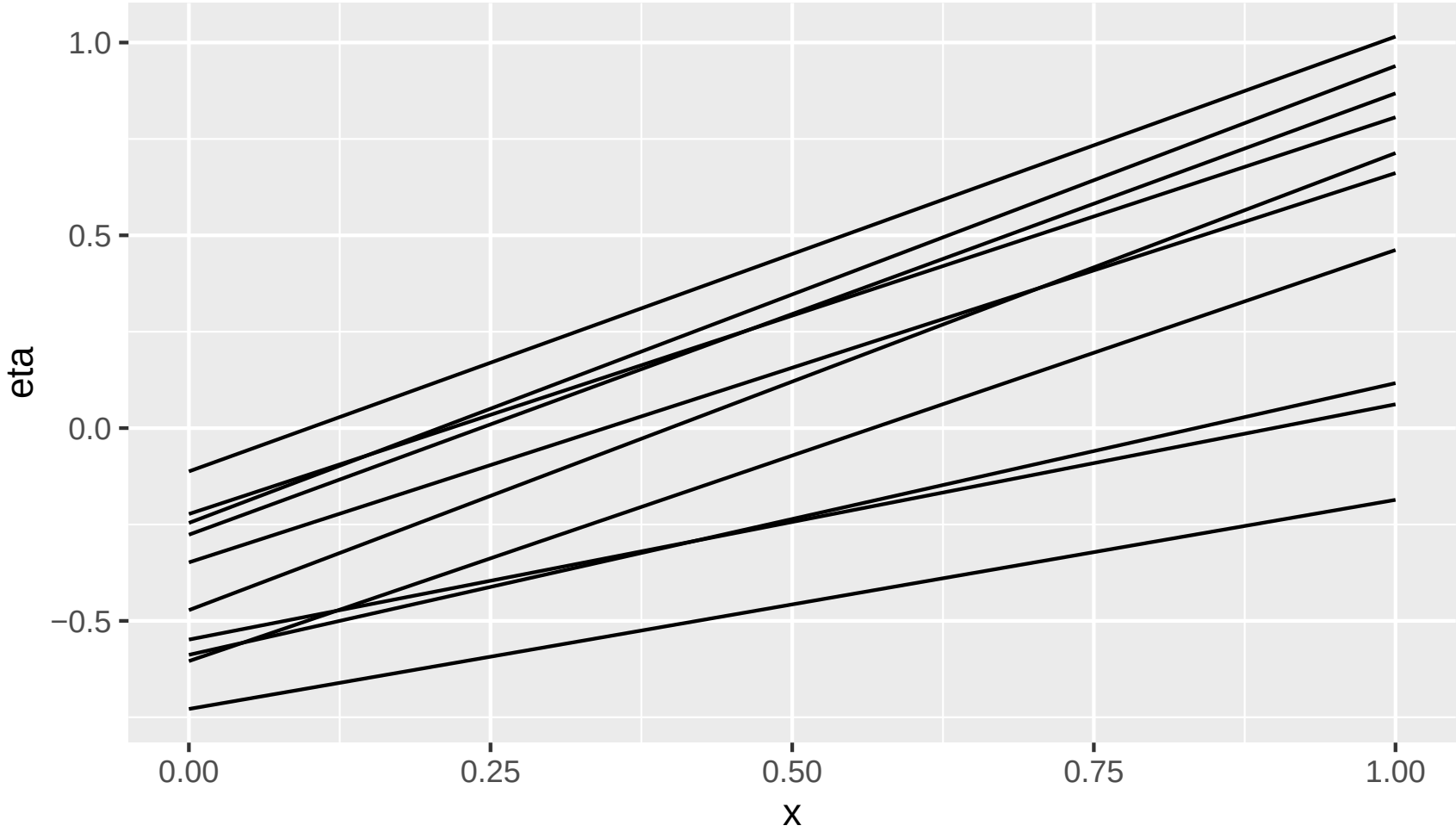
GLMM (random slopes)

$$\eta(x, c) = \beta_0 + \beta_1 x + u_{c1} + u_{c2} x, \;\; u_c \sim N_2(0, \Sigma_u)$$



GLMM (random slopes)

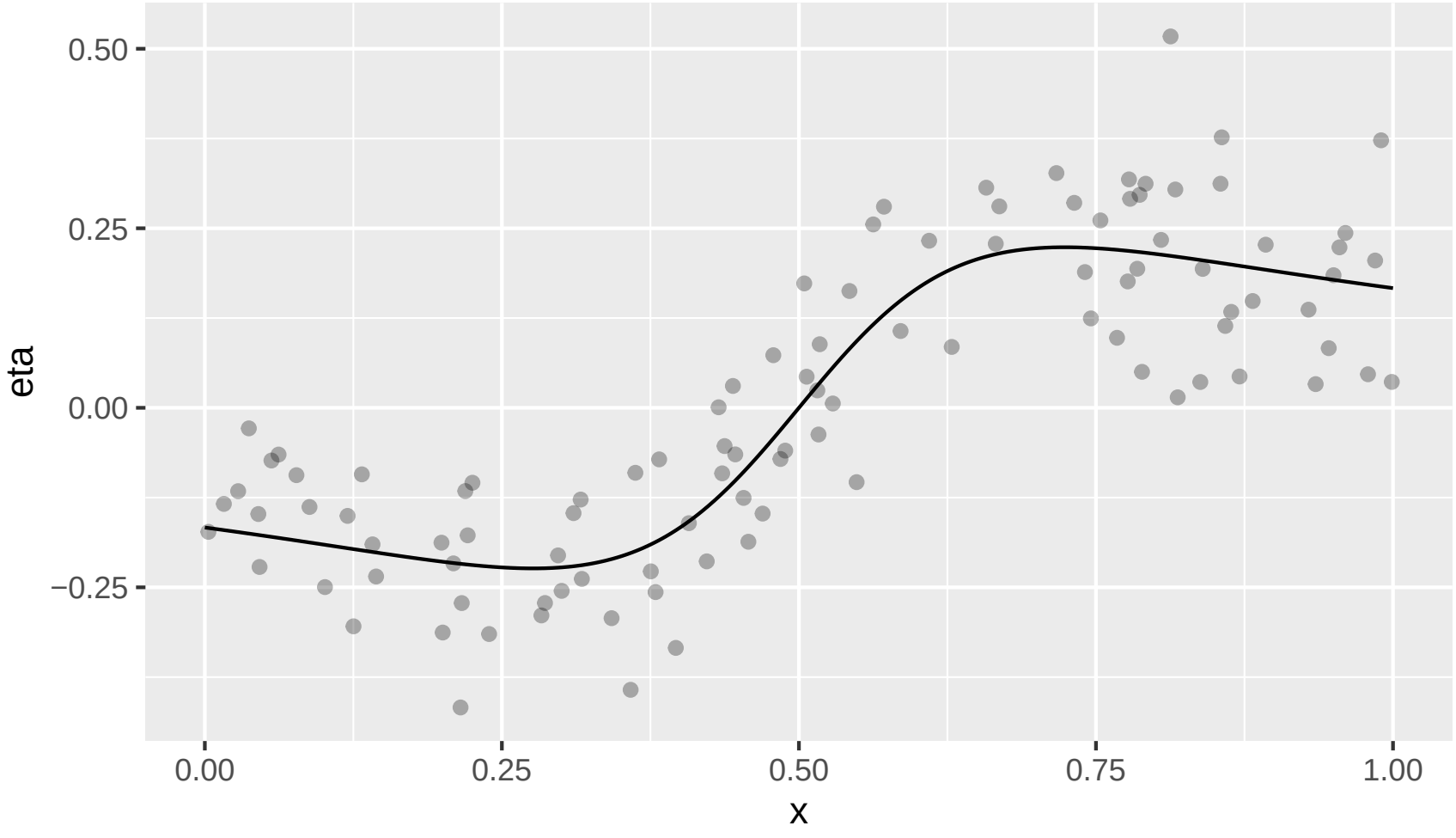
$$\eta(x, c) = \beta_0 + \beta_1 x + u_{c1} + u_{c2} x, \;\; u_c \sim N_2(0, \Sigma_u)$$



What if $\eta(x, c)$ is not linear in x?

Generalised Additive Models

Without clustering, could model $\eta(x)$ as a smooth function of x.



Generalised Additive Models

```
library(mgcv)
mod_gam <- gam(y ~ s(x))
```

Here $\eta(x;\beta) = \beta^T b(x)$ for k basis functions $b(.)$.

Choose β to maximise the penalised loglikelihood

$$\ell_{\text{pen}}(\beta; y) = \ell(\beta; y) - \gamma w_\eta(\beta),$$

where

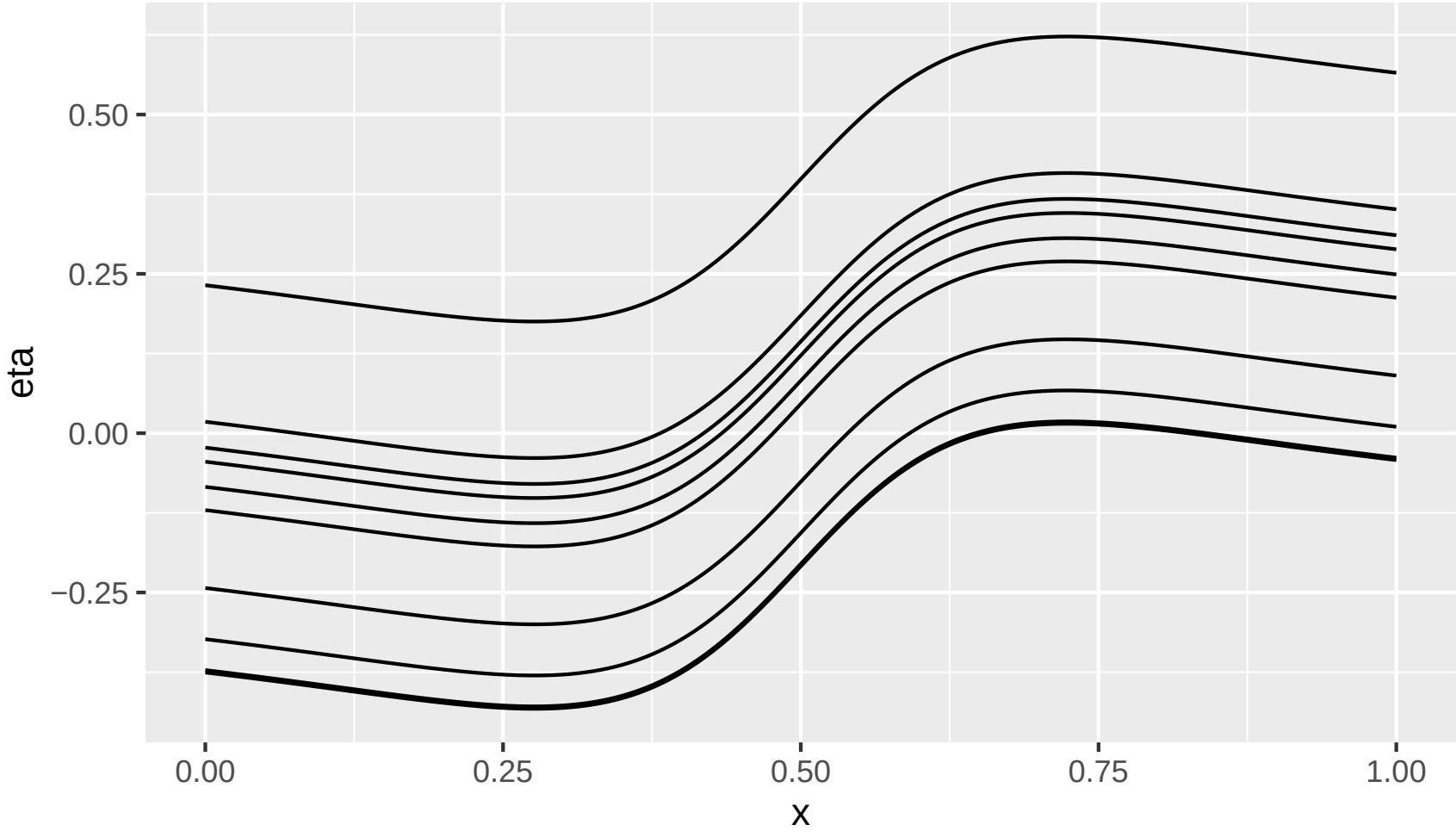
$$w_\eta(\beta) = \int [\eta''(x;\beta)]^2 dx$$

is the wiggliness of $\eta(.;\beta)$ and where the smoothing parameter γ must be chosen (e.g. by cross validation). The smoothing parameter is chosen automatically by `mgcv`.

How can this be extended to include clustering?

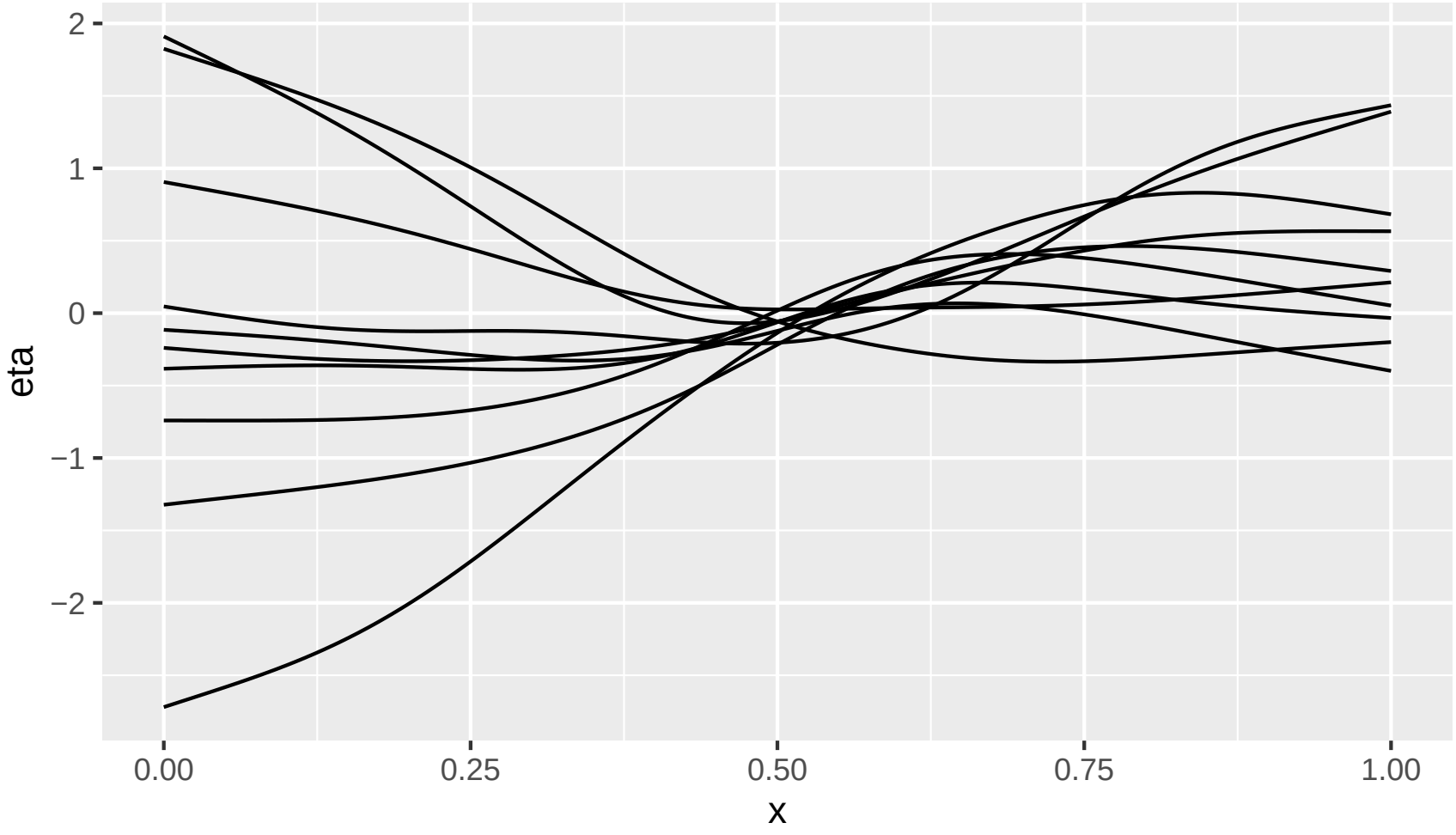
GAMM (random intercept)

$$\eta(x, c) = \beta^T b(x) + u_c, \; u_c \sim N(0, \sigma_u^2)$$

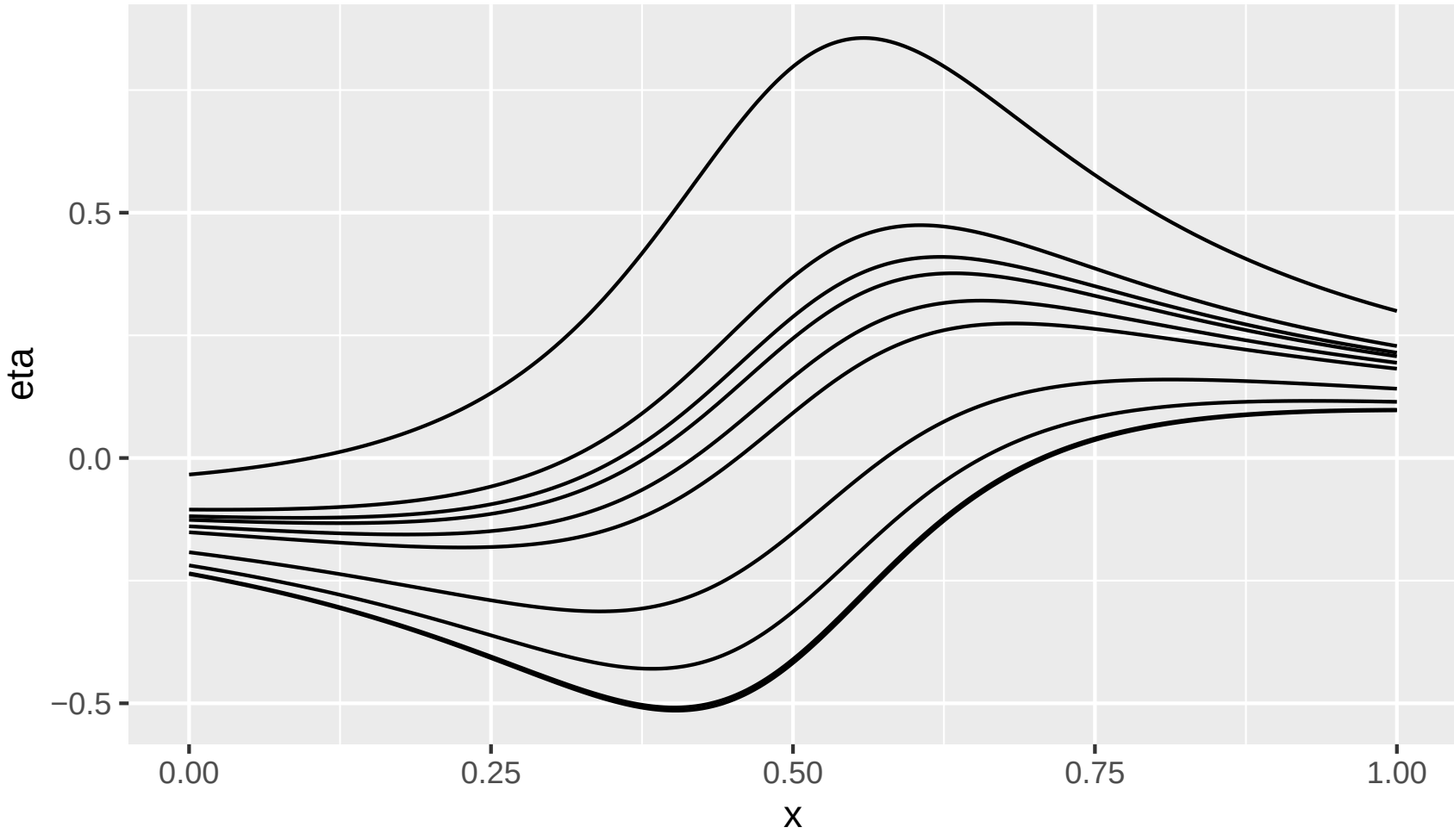


GAMM (random effects for all basis coefficients)

$$\eta(x, c) = \beta^T b(x) + u_c^T b(x), \; u_c \sim N_k(0, \Sigma_u).$$



Example of $\eta(x, c)$ with one-dimensional variation



Binary data generated by thresholding normal data

Suppose

$$Z_i = \beta_0 + \beta_1 x_i + u_{c_i} + \epsilon_i,$$

where $u_j \sim N(0, \sigma_u^2)$ and $\epsilon_i \sim N(0, \sigma^2(x_i))$, and let

$$Y_i = \begin{cases} 1 & \text{if } Z_i > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Then $Y_i \sim \text{Bernoulli}(\mu_i)$ where

$$\Phi^{-1}(\mu_i) = \eta(x, c) = \frac{\beta_0 + \beta_1 x + u_c}{\sigma(x)}.$$

If $\sigma(x) = \sigma$, then a random intercept model is correctly specified, but if $\sigma(x)$ varies with x it is not.

Form of $\eta(x, c)$ in heteroscedastic case

Here we have

$$\eta(x, c) = \frac{\beta_0 + \beta_1 x + u_c}{\sigma(x)}$$

for some non-constant function $\sigma(x)$.

We can rewrite as

$$\eta(x, c) = \mu(x) + u_c \delta(x),$$

where

$$\mu(x) = \frac{\beta_0 + \beta_1 x}{\sigma(x)}, \quad \delta(x) = \frac{1}{\sigma(x)}.$$

We only need one parameter u_c to represent the variation between clusters.

Extending the random intercept model

Instead of a random intercept model

$$\eta(x, c) = \mu(x) + u_c, \quad u_c \sim N(0, \sigma_u^2)$$

where $\mu(.)$ is an unknown smooth functions, we could model

$$\eta(x, c) = \mu(x) + u_c\delta(x), \quad u_c \sim N(0, \sigma_u^2)$$

where $\mu(.)$ and $\delta(.)$ are both unknown smooth functions.

Problem: many choices of $(\mu(.), \delta(.), \sigma_u^2)$ parameterise the same process.

Extending to multiple $\delta(.)$ terms

More generally, we could allow

$$\eta(x, c) = \mu(x) + \sum_{l=1}^{L} u_{cl}\delta_l(x), \quad u_c \sim N_L(0, \Sigma_u)$$

where $\mu(.)$ and each $\delta_l(.)$ are unknown smooth functions.

Problem: many choices of $(\mu(.), \delta_1(.), \ldots, \delta_L(.), \Sigma_u)$ parameterise the same process.

Link to functional data analysis

Recall we are modelling

$$\eta(x, c) = \mu(x) + \sum_{l=1}^{L} u_{cl}\delta_l(x),$$

where $\mu(.)$ and each $\delta_l(.)$ are unknown smooth functions.

We don't observe the functions $\eta(., c)$ themselves, but what if we did? Could we then estimate $\mu(.)$ and $\delta_l(.)$?

The Karhunen–Loève decomposition

We think of the functions $\eta(., c)$ as independent draws from a process.

The Karhunen–Loève decomposition tells us

$$\eta(x, c) = \mu(x) + \sum_{l=1}^{\infty} u_{cl}\delta_l(x),$$

the u_{cl} are uncorrelated random variables, with $\text{var}(u_{cl}) = \lambda_l$, $\lambda_1 \geq \lambda_2, \ldots$ and where $\delta_l(.)$ are orthonormal functions:

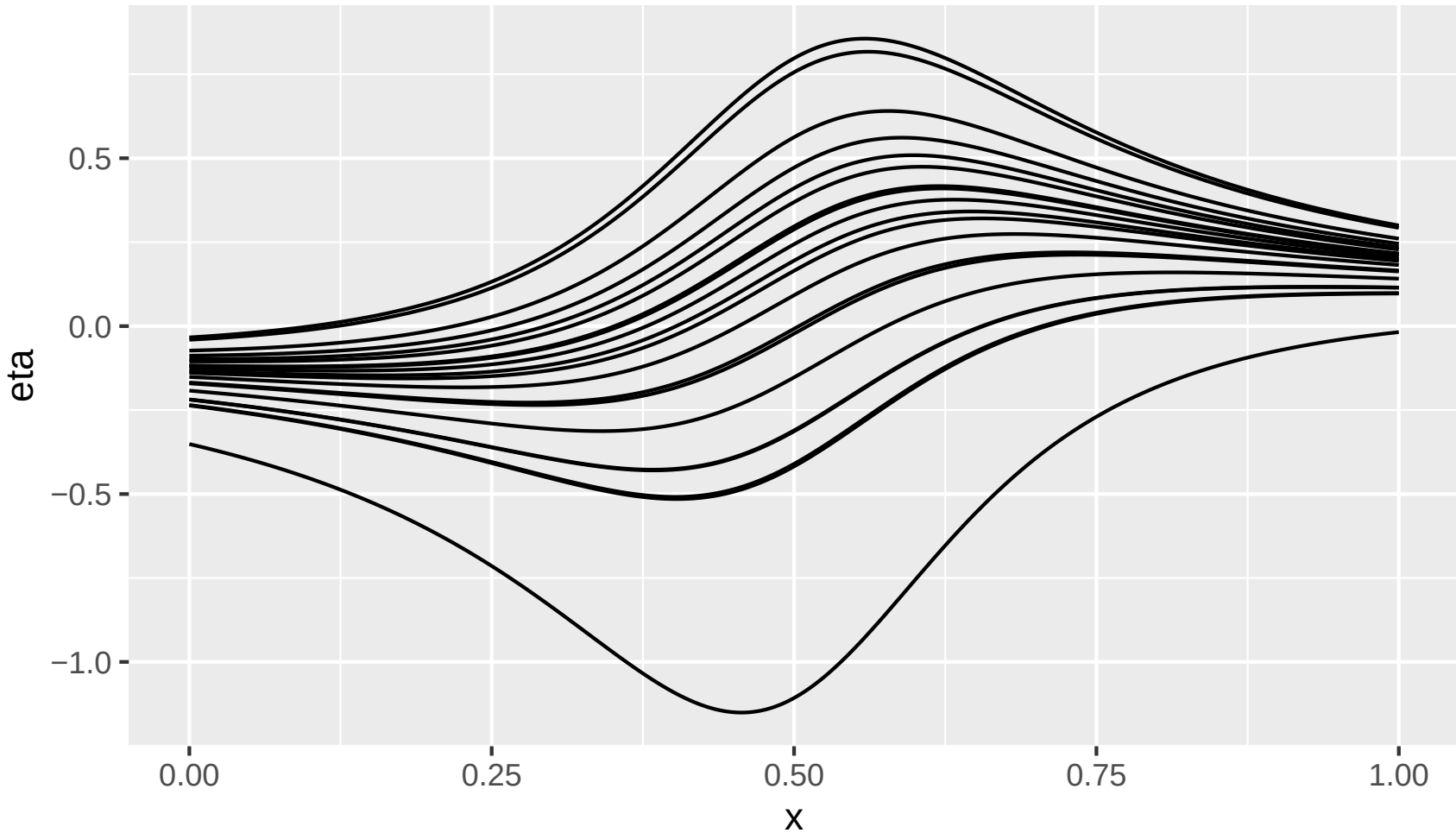
$$\int \delta_l(x)^2 dx = 1,$$

$$\int \delta_l(x)\delta_k(x) = 0, \quad l \neq k.$$

The $\delta_l(.)$ are the functional principal components (FPCs) of the process.

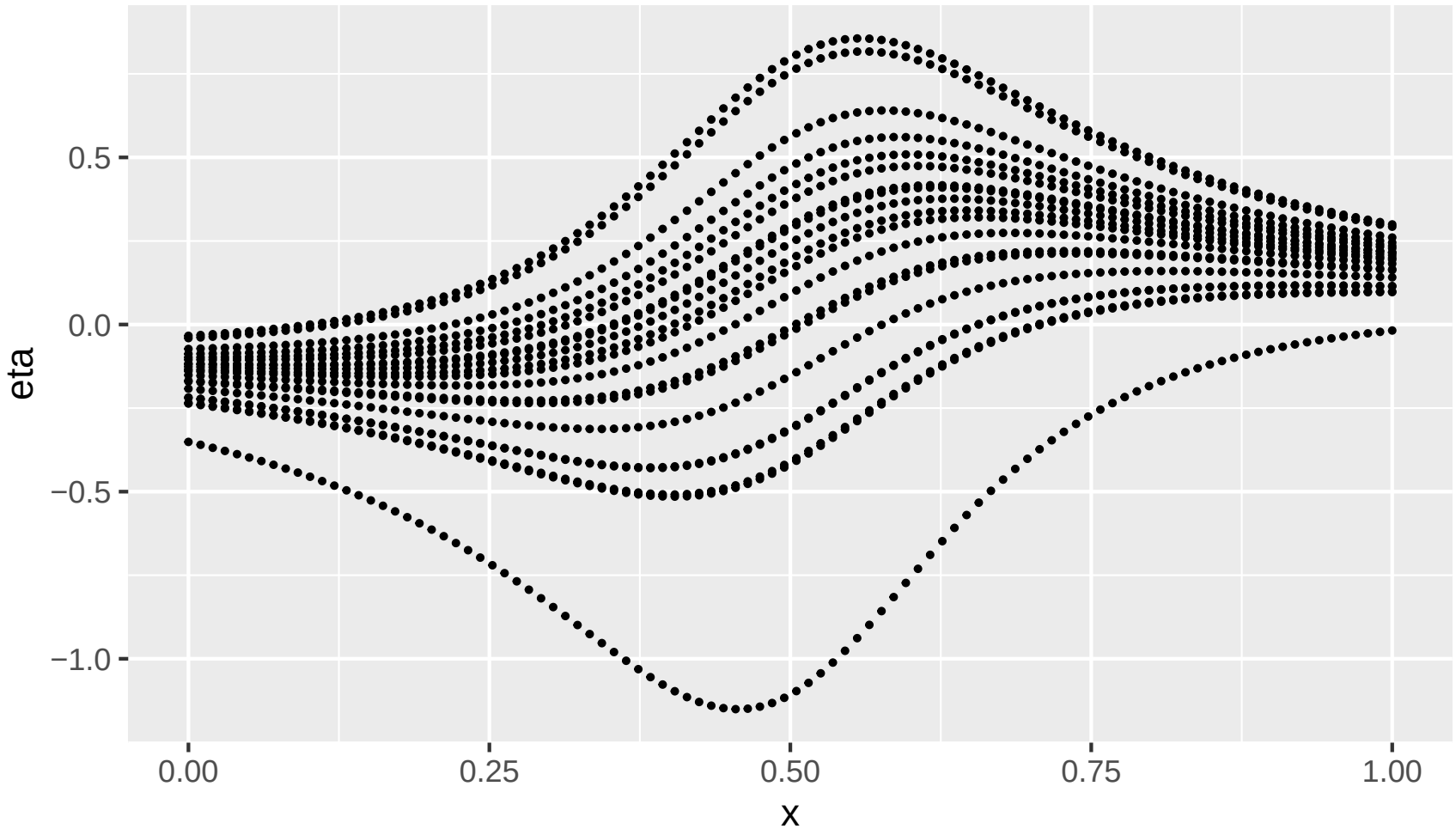
Finding FPCs given $\eta(., c)$

Suppose we observe $\eta(x, c)$ at a fine grid of values for x, and store values as a matrix $N_{cj} = \eta(c, x_j)$.



Estimating FPCs given $\eta(., c)$

Suppose we observe $\eta(x, c)$ at a fine grid of values for x, and store values as a matrix $N_{cj} = \eta(c, x_j)$.



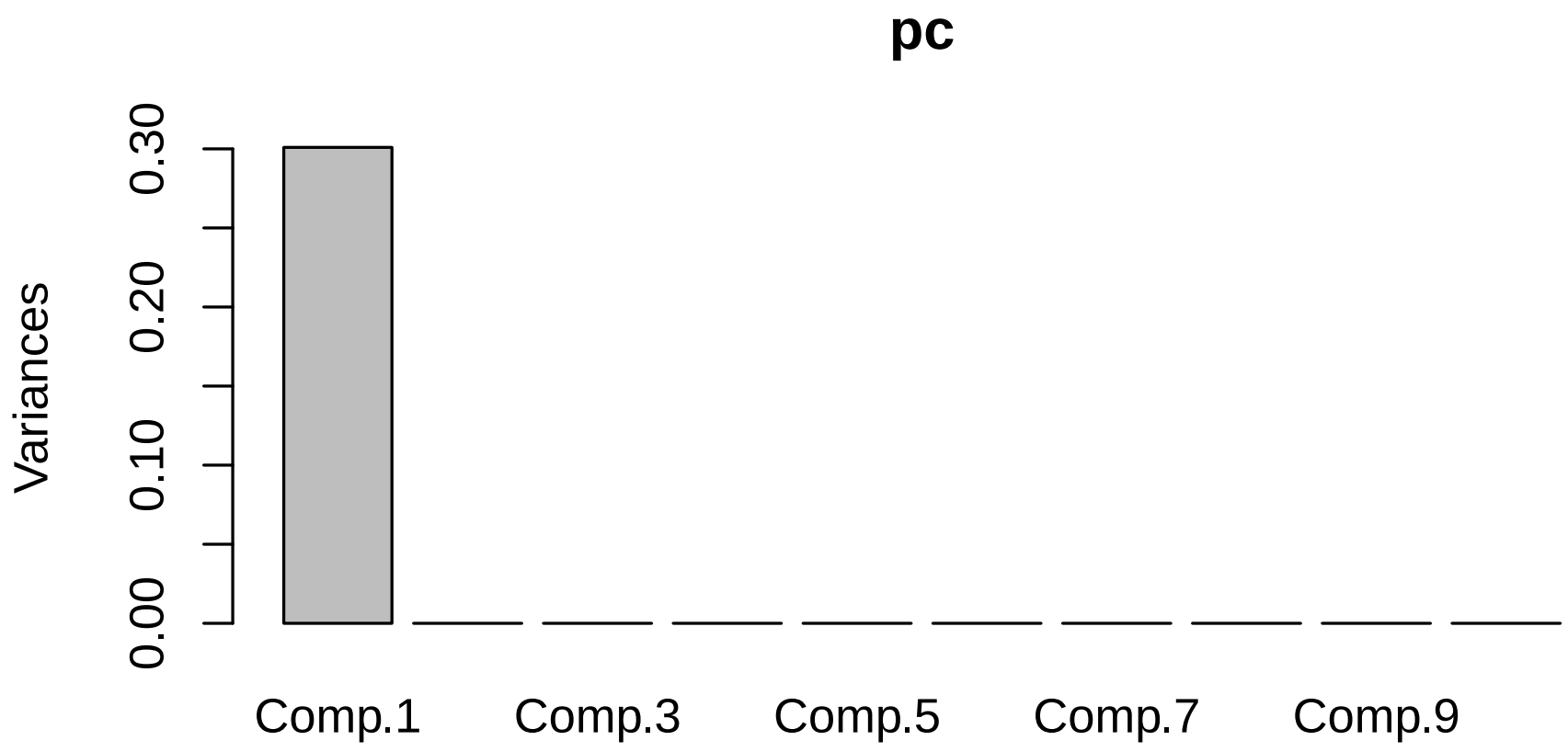
Estimating FPCs given $\eta(., c)$

After subtracting off the mean function, taking the principal components of this matrix gives us the functional principal components evaluated at the fine grid of x values.

```
mu <- rowMeans(N)
N_norm <- N - mu
pc <- princomp(N_norm)
```

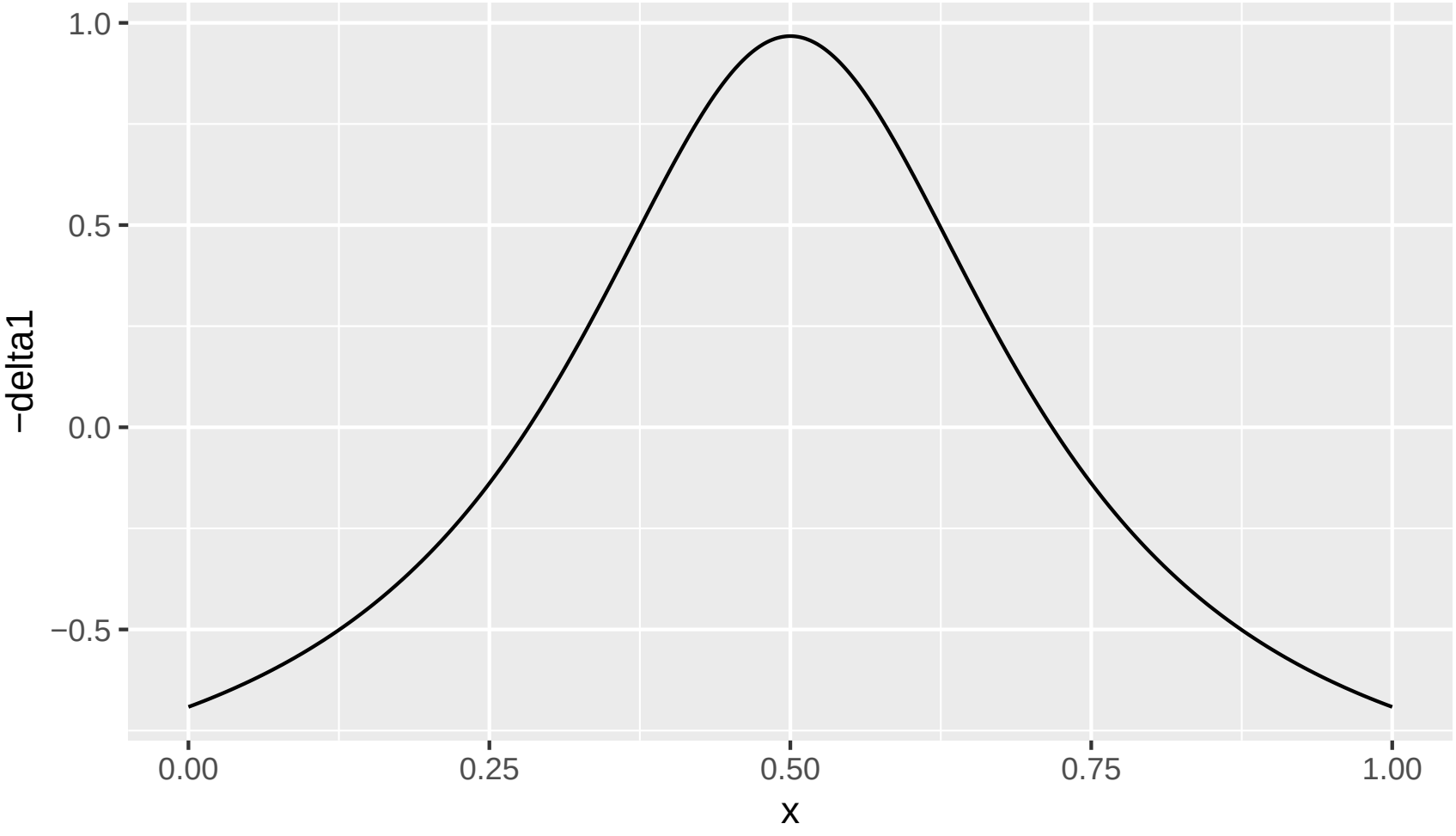
Estimating FPCs given $\eta(., c)$

```
screeplot(pc)
```



Estimating FPCs given $\eta(., c)$

```
delta1 <- pc$scores[,1]
```



Functional principal components modelling

Unfortunately, we don't observe $\eta(x, c)$ for a fine grid of x values. We observe a noisy version of $\eta(x, c)$ at a sparse and irregular grid.

We truncate the Karhunen–Loève decomposition and model

$$\eta(x, c) = \mu(x) + \sum_{l=1}^{L} u_{cl}\delta_l(x), \quad u_{cl} \sim N(0, \lambda_l)$$

for some L (which we must choose), where $\delta_l(.)$ are orthonormal functions. We have to estimate $\mu(.)$ and each $\delta_l(.)$ and λ_l.

For given $\mu(.)$ and $\delta_l(.)$ we could fit the model as a GLMM.

A two-step approach

The `refund` `R` package (Goldsmith et al. 2022) suggests a two-step approach (for normal responses):

1. Obtain estimates of $\mu(.)$ and $\delta_1(.), \ldots, \delta_L(.)$. See `fpca.sc` for details.
2. Fit the model with $\mu(.)$ and $\delta_1(.), \ldots, \delta_L(.)$ fixed at these estimates.

Gertheiss, Goldsmith, and Staicu (2017) suggest a two-step approach in the case of non-normal responses.

Choosing L in the two-step approach

We must choose the number of FPCs to include in the model with this two-step approach.

This is typically done by choosing L in the first step to explain a certain proportion of the variation. Writing FVE_L for the fraction of variance explained by first L PCs, we have

$$\text{FVE}_L = \frac{\sum_{l=1}^{L} \lambda_l}{\sum_{l=1}^{\infty} \lambda_l},$$

and can choose L to be the smallest value so that (e.g.) $\text{FVE}_L \geq 0.95$.

An alternative approach

Instead of fixing $\mu(.)$ and $\delta_l(.)$ at the first step, we can estimate them as part of a larger model.

The marginal likelihood

$$L(\mu, \delta, \lambda, \sigma; y) = \int f(y|\eta = \mu(x) + u^T \delta(x); \sigma) \prod_{l=1}^{L} \phi(u_l, 0, \lambda_l) du$$

may be found by using standard mixed-effects model software, e.g. the `lme4` `R` package (Bates et al. 2015).

A penalised likelihood approach

We penalise the expected wiggliness of $\eta(x) = \mu(x) + \sum_{l=1}^{L} u_{cl}\delta_l(x)$:

$$\text{pen}(\mu, \delta, \lambda) = E(w_\eta) = w_\mu + \sum_{l=1}^{L} \lambda_l w_{\delta_l}.$$

The penalised loglikelihood is

$$\ell_{\text{pen}}(\mu, \delta, \lambda, \sigma; y) = \ell(\mu, \delta, \lambda, \sigma; y) - \gamma \text{pen}(\mu, \delta, \lambda),$$

where $\ell(.; y) = \log L(.; y)$ is the marginal loglikelihood and γ is a smoothing parameter.

Basis functions

We could aim to maximise the penalised loglikelihood over all functions $\mu(.)$ and $\delta(.) = (\delta_1(.), \ldots, \delta_L(.))$ such that $\delta_l(x)$ are orthonormal functions.

To make this easier, we restrict to $\mu(.)$ and $\delta_l(.)$ of the form $\mu(x) = \beta_\mu^T b(x)$ and $\delta_l(x) = \beta_{\delta_l}^T b(x)$ for fixed basis functions $b(.)$.

Write $\beta = (\beta_\mu, \beta_{\delta_1}, \ldots, \beta_{\delta_L})$, and $\theta = (\beta, \lambda, \sigma)$ for the model parameters.

Removing constraint on optimisation

We aim to maximise the penalised loglikelihood

$$\ell_{\text{pen}}(\theta; y) = \ell(\theta; y) - \gamma \, \text{pen}(\theta),$$

subject to $\delta_l(x) = \beta_{\delta_l}^T b(x)$ being orthonormal functions.

This is a relatively complex constrained optimization problem.

Instead, it is possible to reparameterise $\delta_l(.)$ with new parameters ϕ_l which ensure orthonormality.

Choosing the tuning parameters

We use a Leave-One-Cluster-Out Cross Validation criterion for γ and L.

$$\text{LOCO}(\gamma, L) = \sum_c \log f(y_c | x_c; \hat{\theta}_{-c}(\gamma, L)),$$

where:

- $y_c = (y_i : c_i = c)$, $x_c = (x_i : c_i = c)$ is the data for cluster c
- $f_{-c}(.|x, c, \theta)$ is the marginal density for y_c given x_c given parameter value θ
- $\hat{\theta}_{-c}(\gamma, L)$ is the parameter estimate from the data with cluster c removed, given γ and L.

LOCO can be very time-consuming to compute, as we have to refit the model with each cluster removed.

Speeding up LOCO

To speed up computation, we make use of the fact that $\hat{\theta}_{-c}$ is usually close to $\hat{\theta}$.

The penalised loglikelihood is

$$\ell_{\text{pen}}(\theta; y) = \sum_c \ell(\theta; y_c) - \gamma \text{pen}(\theta)$$

so if we store the component parts $\ell(\hat{\theta}; y_c)$ and $\text{pen}(\hat{\theta})$ we can compute

$$\ell_{\text{pen}}(\hat{\theta}; y_{-c}) = \sum_{c' \neq c} \ell(\hat{\theta}; y_{c'}) - \gamma \text{pen}(\hat{\theta})$$

directly from the stored parts.

Similarly, if we store the derivatives of the component parts at $\hat{\theta}$, we can compute the derivatives of $\ell_{\text{pen}}(.; y_{-c})$ at $\hat{\theta}$.

Using all this information, it typically only takes a few steps of an optimization algorithm to find each $\hat{\theta}_{-c}$.

Simulation study

Simulate 100 datasets each with 50 clusters and 10 observations per cluster, where $x_i \sim U(0,1)$,

$$Y_i \sim N(\eta(x_i, c_i), (0.1)^2),$$

$$\eta(x,c) = \frac{-1 + x + u_c}{\sigma(x)}, \quad u_c \sim N(0, (0.5)^2)$$

and

$$\sigma(x) = 1 + 0.2(10x - 5)^2.$$

Fit with two-step approach, either fixing $L = 1$ or choosing L to explain 95% of the variance, and with the new penalised likelihood approach, with L and γ chosen by the LOCO criterion.

Root Mean Squared Error

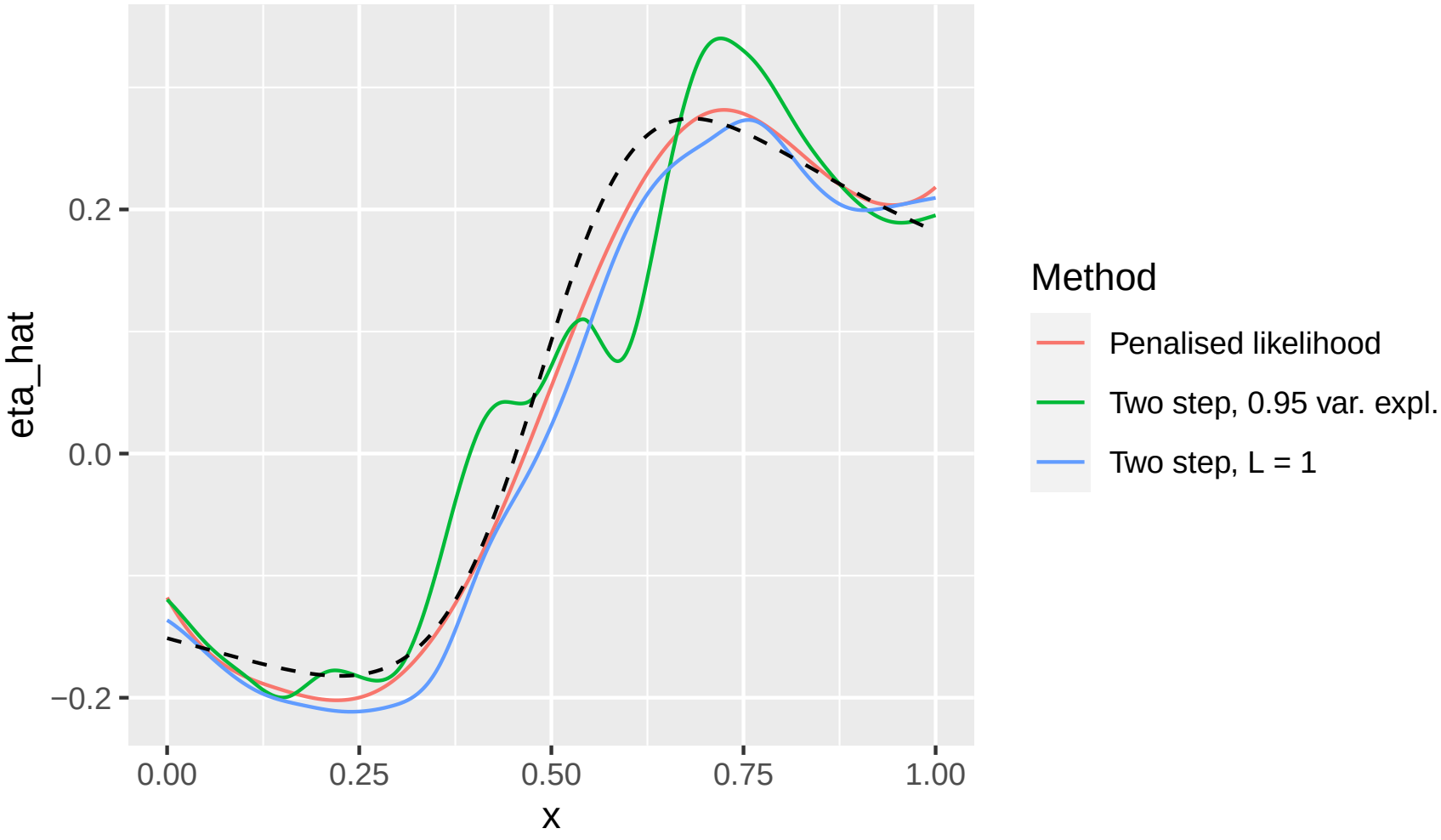
For each cluster c, we find an integrated mean squared error

$$\mathrm{MSE}(\hat{\eta}(,c),\eta(,c)) = \int_0^1 (\hat{\eta}(x,c) - \eta(x,c))^2 dx,$$

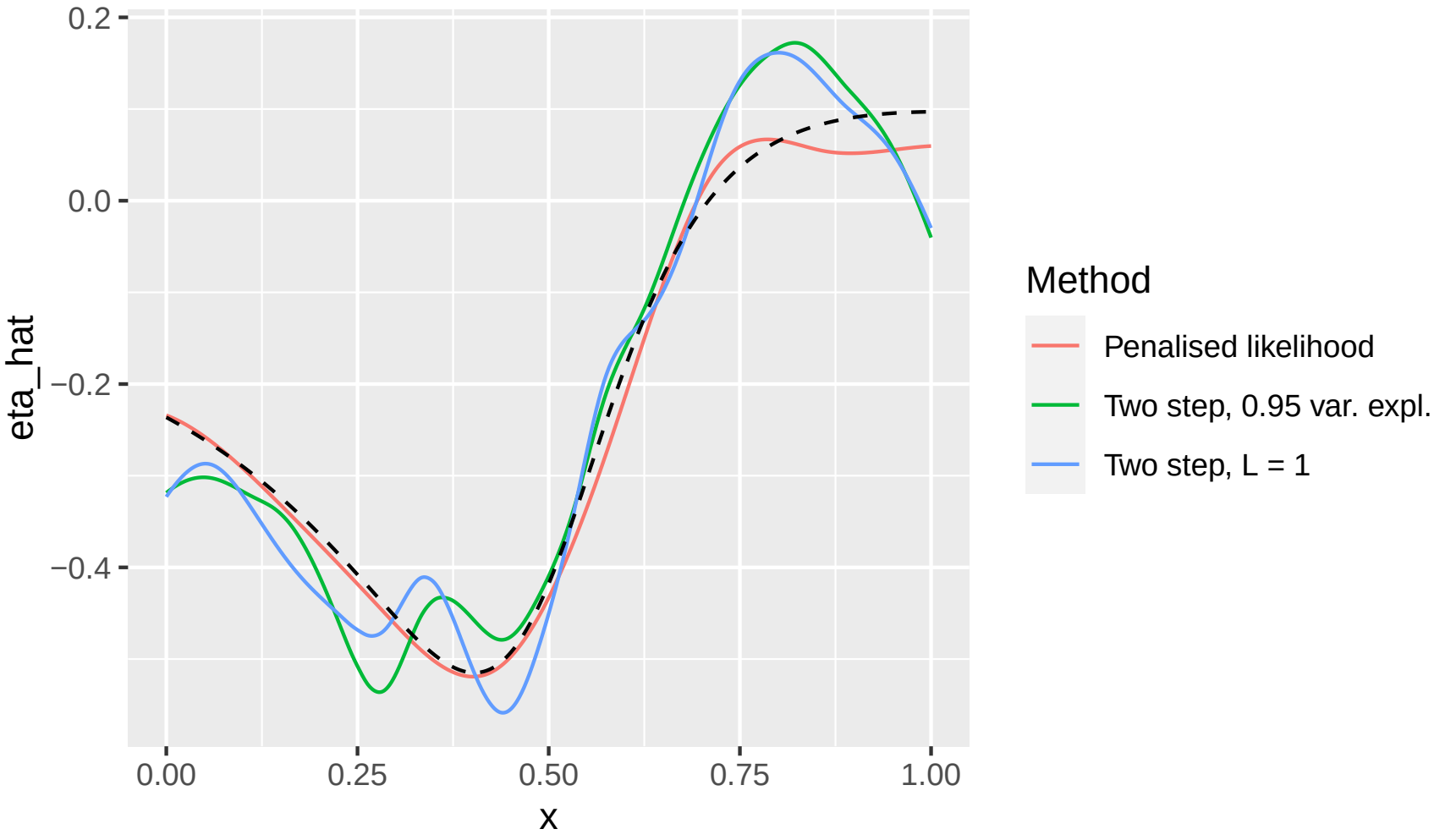
and average over all clusters and take square roots, to give an overall root mean square error (RMSE) for each method.

Method	RMSE
Penalised likelihood	0.037
Two step, 0.95 var. expl.	0.070
Two step, L = 1	0.064

Example fitted $\eta(x, c)$



Example fitted $\eta(x, c)$



Conclusions

- Random intercept models not particularly natural when we move away from straight-line dependence on x.
- Methods from functional data analysis may be applied in this setting, but two-step approach does not always work well.
- Our penalised likelihood methods seem to provide good performance across data structures in this simple setup, with automated smoothing parameter selection.
- Penalised likelihood method extends to non-normal responses.
- Challenge: extension to multiple continuous covariates.
- Challenge: extension to multiple clustering variables.

References

Bates, Douglas, Martin Mächler, Ben Bolker, and Steve Walker. 2015. "Fitting Linear Mixed-Effects Models Using lme4." *Journal of Statistical Software* 67 (1): 1–48. https://doi.org/10.18637/jss.v067.i01.

Gertheiss, Jan, Jeff Goldsmith, and Ana Maria Staicu. 2017. "A note on modeling sparse exponential-family functional response curves." *Computational Statistics and Data Analysis* 105: 46–52. https://doi.org/10.1016/j.csda.2016.07.010.

Goldsmith, Jeff, Fabian Scheipl, Lei Huang, Julia Wrobel, Chongzhi Di, Jonathan Gellar, Jaroslaw Harezlak, et al. 2022. *Refund: Regression with Functional Data*. https://CRAN.R-project.org/package=refund.