Quantile GAM modelling with qgam

Matteo Fasiolo

matteo.fasiolo@bristol.ac.uk

Material available at:

`https://github.com/mfasiolo/GAM_Workshop_Dortmund_25`

These slides cover:

1. Intro to quantile GAM models
2. Fitting GAMs with mgcv
3. Fitting GAMs with qgam
4. Big Data methods
5. Quantile GAM modelling with qgam

What is quantile regression?

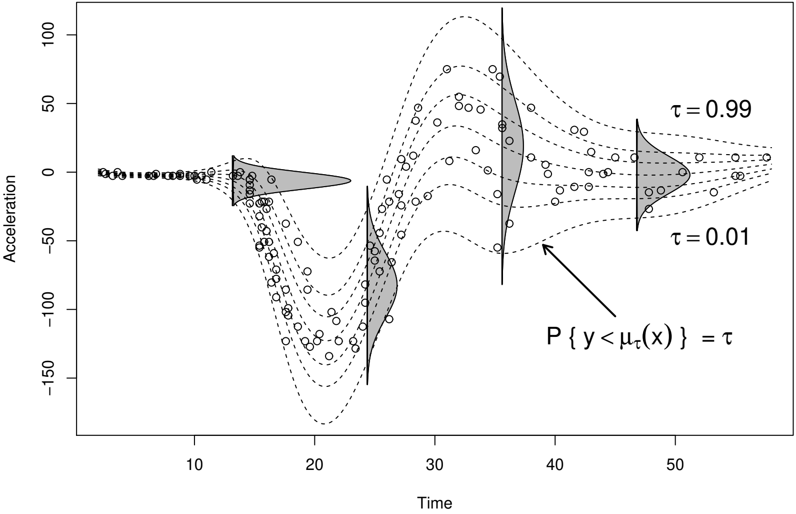
Regression setting:

- y is our response or dependent variable
- $\boldsymbol{x}$ is a vector of covariates or independent variables

In **distributional regression** we want a good model for $\text{Distr}(y|\boldsymbol{x})$.

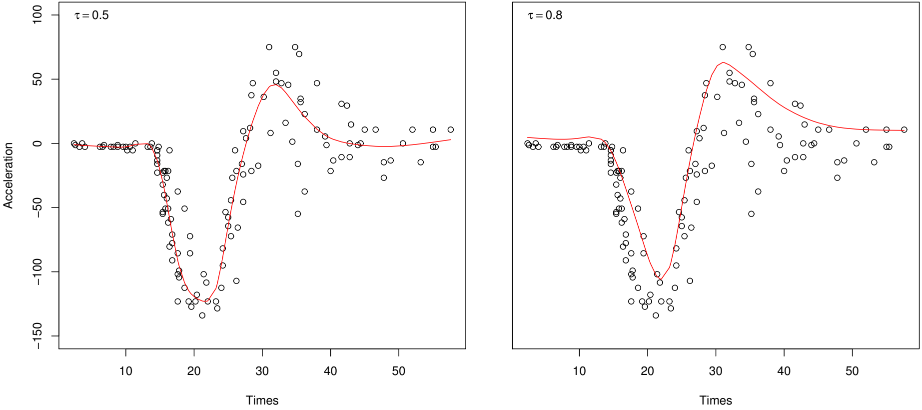
Model is $\text{Distr}_m\{y|\theta_1(\boldsymbol{x}), \dots, \theta_q(\boldsymbol{x})\}$, where $\theta_1(\boldsymbol{x}), \dots, \theta_q(\boldsymbol{x})$ are parameters.

Given $\text{Distr}_m(y|\boldsymbol{x})$ we can get the conditional quantiles $\mu_\tau(\boldsymbol{x})$.



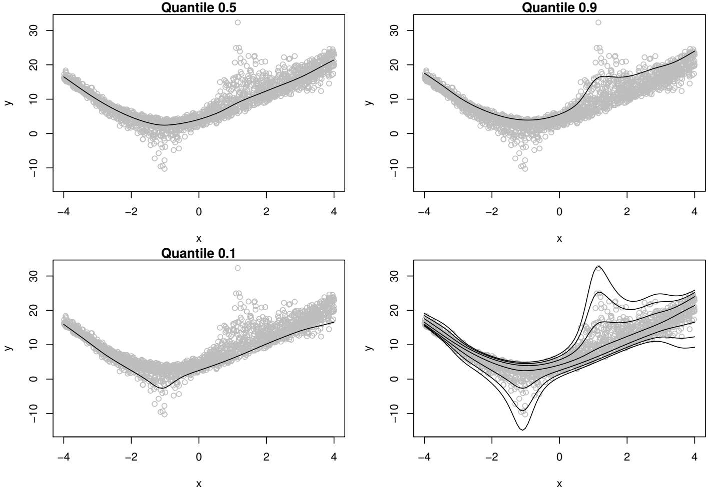
Quantile regression estimates conditional quantiles $\mu_\tau(\boldsymbol{x})$ directly.

No model for Distr$(y|\boldsymbol{x})$.

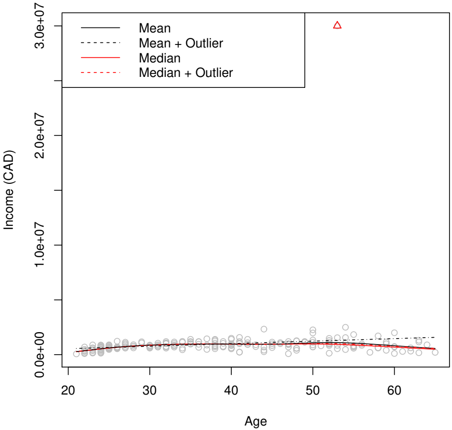


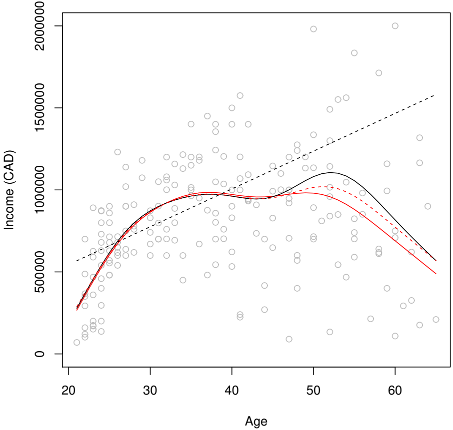
No assumptions on Distr$(y|\boldsymbol{x})$:

- no need to find good model for Distr$(y|\boldsymbol{x})$;
- no need to find normalizing transformations (e.g. Box-Cox);



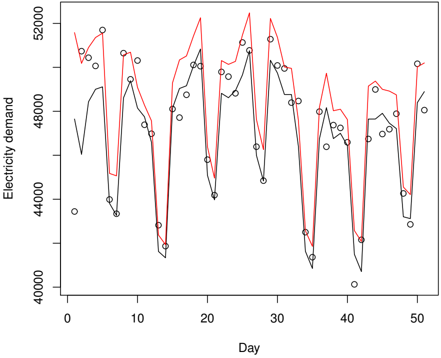
The median is also more **resistant to outliers**.

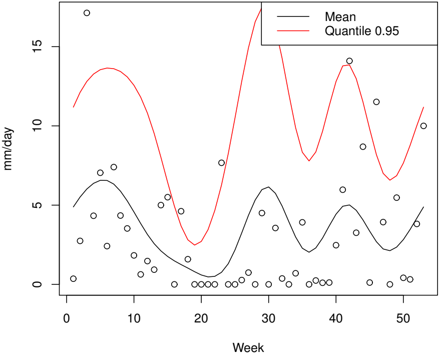




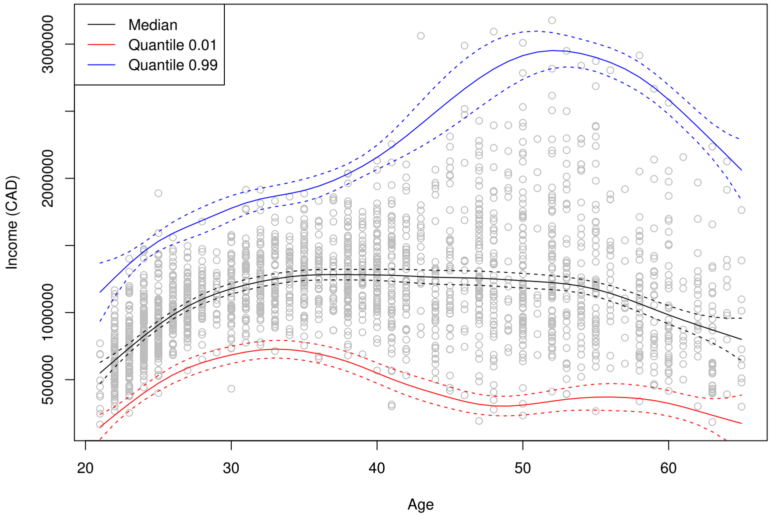
Some quantiles are more important than others:

- electricity producers need to satisfy high electricity demand
- urban planners need estimates of extreme rainfall





Effect of explanatory variables may depend on quantile



These slides cover:

1. Intro to quantile GAM models
2. Fitting GAMs with mgcv
3. Fitting GAMs with qgam
4. Big Data methods
5. Quantile GAM modelling with qgam

Model fitting

Recall the GAM model structure:

$$y|\boldsymbol{x} \sim \text{Distr}\{y|\mu(\boldsymbol{x}), \boldsymbol{\theta}\} \quad \text{where} \quad g(\mu(\boldsymbol{x})) = \sum_{j=1}^{m} f_j(\boldsymbol{x}).$$

In `mgcv` $\boldsymbol{\beta}$ estimated by Maximum a Posterior (MAP)

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\text{argmax}} \ \log p(\boldsymbol{\beta}|\mathbf{y}, \boldsymbol{\lambda}) = \underset{\boldsymbol{\beta}}{\text{argmax}} \Big\{ \overbrace{\log p(\mathbf{y}|\boldsymbol{\beta})}^{\text{goodness of fit}} + \underbrace{\log p(\boldsymbol{\beta}|\boldsymbol{\lambda})}_{\text{prior penalising complexity}} \Big\}$$

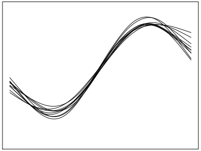
where:

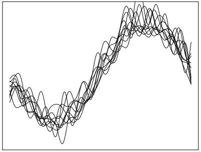
- $\log p(\boldsymbol{y}|\boldsymbol{\beta})$ is log-likelihood
- $\log p(\boldsymbol{\beta}|\boldsymbol{\lambda})$ penalizes the complexity of the f_j's
- $\boldsymbol{\lambda} > 0$ smoothing parameters ($\uparrow \boldsymbol{\lambda}$ $\uparrow$smoothness)

`mgcv` uses a hierarchical fitting framework:

1. Select $\boldsymbol{\lambda}$ to determine smoothness

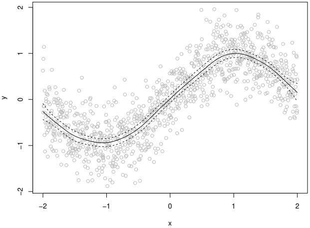
$$\hat{\boldsymbol{\lambda}} = \underset{\boldsymbol{\lambda}}{\operatorname{argmax}} \, \text{LAML}(\boldsymbol{\lambda}).$$





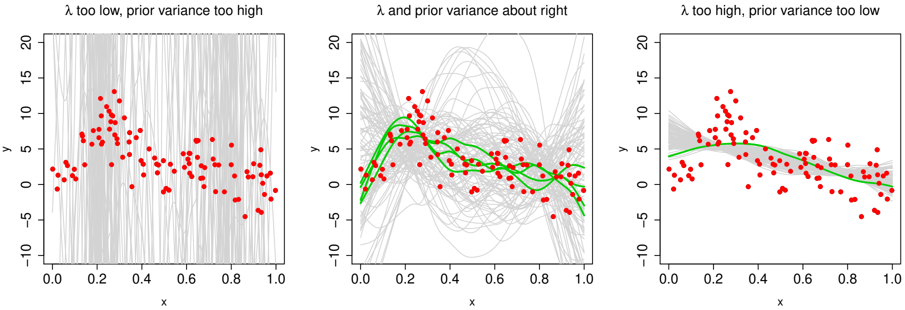
2. For fixed $\boldsymbol{\lambda}$, estimate $\boldsymbol{\beta}$ to determine actual fit

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmax}} \, \log p(\boldsymbol{\beta}|\boldsymbol{\lambda}).$$



What is the Laplace Approximate Marginal Likelihood?

$$\text{LAML}(\boldsymbol{\lambda}) \approx p(\boldsymbol{y}|\boldsymbol{\lambda}) = \int p(\boldsymbol{y}|\boldsymbol{\beta})p(\boldsymbol{\beta}|\boldsymbol{\lambda})d\boldsymbol{\beta}.$$



Alternatives LAML for $\boldsymbol{\lambda}$ selection:

- Generalized Cross-Validation (GCV)
- Akaike Infomation Criterion (AIC)

but LAML is most widely applicable in `mgcv`.

To choose $\boldsymbol{\lambda}$ estimation method in `mgcv`

```
fit <- gam(y ~ ..., method = "REML")
```

see `?gam`.

LAML is the default for multi-parameter GAMs.

These slides cover:

1. Intro to quantile GAM models
2. Fitting GAMs with mgcv
3. Fitting GAMs with qgam
4. Big Data methods
5. Quantile GAM modelling with qgam

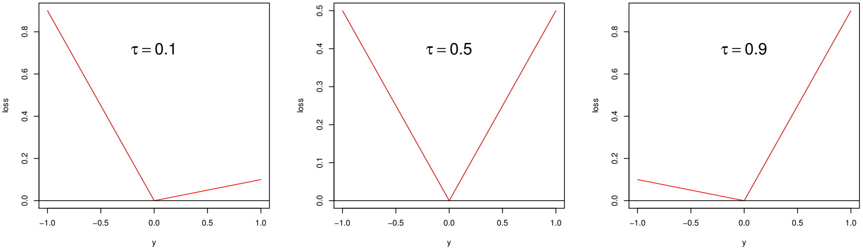
Quantile GAM fitting

In parametric GAMs $\mu_\tau(\boldsymbol{x}) = F^{-1}(\tau|\boldsymbol{x})$.

Key fact: $\mu_\tau(\boldsymbol{x})$ is the minimizer of

$$L(\mu|\boldsymbol{x}) = \mathbb{E}\{ \, \rho_\tau(y-\mu) \, | \, \boldsymbol{x} \, \},$$

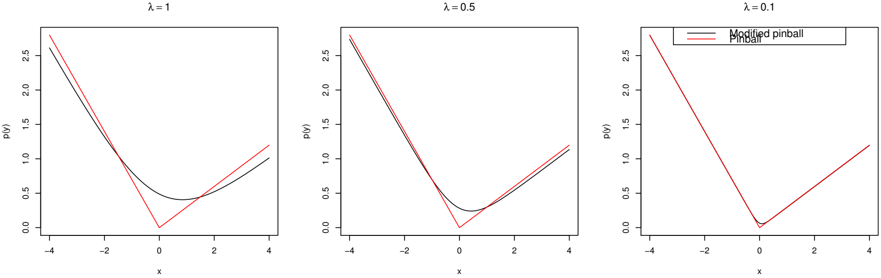
where ρ_τ is the "pinball" loss (Koenker, 2005):



In additive modelling context $\mu_\tau(\boldsymbol{x}) = \sum_{j=1}^m f_j(\boldsymbol{x}) = \mu_\tau(\boldsymbol{\beta})$.

qgam uses a modified loss which we call Extended log-F (ELF) loss.

This is smooth and convex and, as $\gamma \to 0$, we recover pinball loss.



In the plots above λ should be γ.

Since qgam 1.3.0, γ is selected automatically.

Smoothing the loss has statistical advantages, see Fasiolo et al. (2021a).

Recall $\boldsymbol{\beta}$ estimated by maximusing log-posterior

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\text{argmax}}\ \log p(\boldsymbol{\beta}|\boldsymbol{\lambda}) = \underset{\boldsymbol{\beta}}{\text{argmax}} \Big\{ \overbrace{\log p(\boldsymbol{y}|\boldsymbol{\beta})}^{\text{goodness of fit}} + \underbrace{\log p(\boldsymbol{\beta}|\boldsymbol{\lambda})}_{\text{prior penalising complexity}} \Big\}.$$

We plug the negative ELF loss in place of $\log p(\boldsymbol{y}|\boldsymbol{\beta})$ so

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\text{argmax}}\ \log p(\boldsymbol{\beta}|\boldsymbol{\lambda}) = \underset{\boldsymbol{\beta}}{\text{argmax}} \Big\{ \underbrace{-\text{ELFLoss}(\boldsymbol{y}|\boldsymbol{\beta})}_{\text{Pseudo log-likelihood}} + \log p(\boldsymbol{\beta}|\boldsymbol{\lambda}) \Big\}.$$

See Fasiolo et al. (2021a) for justification.

Getting a good fit requires adding a new parameter, the **learning rate** σ.

We use a hierarchical fitting framework:

1. Select σ to optimise coverage

$$\hat{\sigma} = \underset{\sigma}{\text{argmin}}\, \text{CalibrLoss}(\sigma).$$

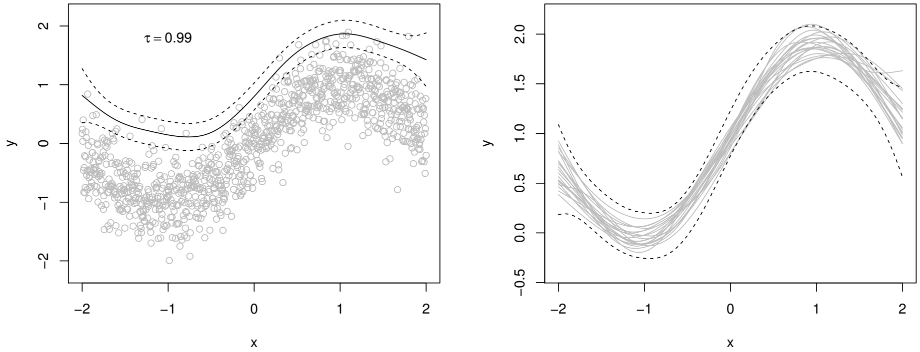
2. For fixed σ, select $\boldsymbol{\lambda}$ to determine smoothness

$$\hat{\boldsymbol{\lambda}} = \underset{\boldsymbol{\lambda}}{\text{argmax}}\, \text{LAML}(\boldsymbol{\lambda}).$$

3. For fixed $\boldsymbol{\lambda}$ and σ, estimate $\boldsymbol{\beta}$

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\text{argmin}}\;\; \log p(\boldsymbol{\beta}|\boldsymbol{\lambda})$$

Minimise CalibrLoss(σ) to match model-based and sampling uncertainty.



NOTE: we can let σ and γ vary with $\boldsymbol{x}$ (see R demo)

These slides cover:

1. Intro to quantile GAM models
2. Fitting GAMs with mgcv
3. Fitting GAMs with qgam
4. Big Data methods
5. Quantile GAM modelling with qgam

GAMs for Big Data

Let $\texttt{Load}_i$ be electricity demand at hour h_i.

Consider Gaussian GAM where

$$\begin{aligned}\mathbb{E}(\texttt{Load}_i) &= \cdots && \cdot \text{ Some effects}\\ &+ f_1(T_i) && \cdot \text{ Temperature}\\ &+ f_2(\texttt{toy}_i), && \cdot \text{ Time-of-year}\end{aligned}$$

It is standard practice to model the 24 hours separately.

So we fit 24 models.

A more ambitious model is

$$\begin{aligned}\mathbb{E}(\texttt{Load}_i) &= \cdots & &\cdot \text{ Some effects}\\ &+ \text{te}_1(T_i, h_i) & &\cdot \text{ Temperature}\\ &+ \text{te}_2(\texttt{toy}_i, h_i), & &\cdot \text{ Time-of-year}\end{aligned}$$

where `te`'s are 2D tensor product smooths.

Why is this useful? Some answers:

- statistical efficiency $\rightarrow$ share information across time-of-day
- ease of use and interpretation

Do we need Big Data methods? Notice that:

- n is 24 times bigger
- tensor product construction

$$\text{te}(\text{T}, h) = \sum_{j=1}^{J} \sum_{k=1}^{K} \beta_{ij} b_j(\text{T}) b_k(h) = \sum_{j=1}^{J} \sum_{k=1}^{K} \beta_{ij} \tilde{b}_{jk}(\text{T}, h)$$

  so tensor effect has $J \times K$ coefficients.

$\mathbb{E}(\text{load}_i) = \mathbf{X}_i^{\text{T}} \boldsymbol{\beta}$, where $\mathbf{X}_i^{\text{T}}$ is row of model matrix $\mathbf{X}$.

Block of $\mathbf{X}$ corresponding to $\text{te}(\text{T}, h)$ is $n \times (K \times J)$.

Bottom line: $\mathbf{X}$ can get very big so

- storing $\mathbf{X}$ takes too much memory
- computing with $\mathbf{X}$ (e.g. $\mathbf{X}^\mathsf{T}\mathbf{X}$) takes time

`mgcv::bam()` uses **memory-saving** methods of Wood et al. (2015):

- do not create $\mathbf{X}$ but only sub-blocks:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_{11} & \mathbf{X}_{12} \\ \mathbf{X}_{21} & \mathbf{X}_{22} \\ \vdots & \vdots \\ \mathbf{X}_{B1} & \mathbf{X}_{B2} \end{bmatrix}$$

  do not store them either, but build them when needed
- any computation involving $\mathbf{X}$ is based on the blocks

Faster computation and memory savings using Wood et al. (2017).

Simple observation is that many variables are discrete in nature:

- time of day (tod) $\in \{1, \dots, 24\}$
- time of year (toy) $\in \{1, \dots, 365\}$
- temperature (T) $\in \{\dots, -0.1, 0, 0.1, 0.2, \dots\}$

There is room for data compression, example:

- we have 10 year of data and 24×365 obs per year
- effect of toy is

$$s(\text{toy}) = \sum_{i=1}^{p} \beta_i b_i(\text{toy}).$$

  so model matrix part $\mathbf{X}$ of toy is $(10 * 24 * 365) \times p$
- compressed model matrix $\bar{\mathbf{X}}$ is $365 \times p$
- saving factor $\#elem(\mathbf{X})/\#elem(\bar{\mathbf{X}}) = 10 * 24$

Discretization can be applied to variables that are not "naturally" discrete.

Sampling variability is $O(n^{-\frac{1}{2}})$, so discretizing in $m = O(n^{\frac{1}{2}})$ bins is ok.

Discrete methods are enabled by:

```
bam(..., discrete = TRUE)
```

Or in `qgam` version $\geq$ 2.0:

```
qgam(..., discrete = TRUE)
```

NOTE: `bam` does not support multi-parameter GAMs.

These slides cover:

1. Intro to quantile GAM models
2. Fitting GAMs with mgcv
3. Fitting GAMs with qgam
4. Big Data methods
5. Quantile GAM modelling with qgam

Demonstration in R

For more details on methodology, see Fasiolo et al. (2021a) and Fasiolo et al. (2021b).

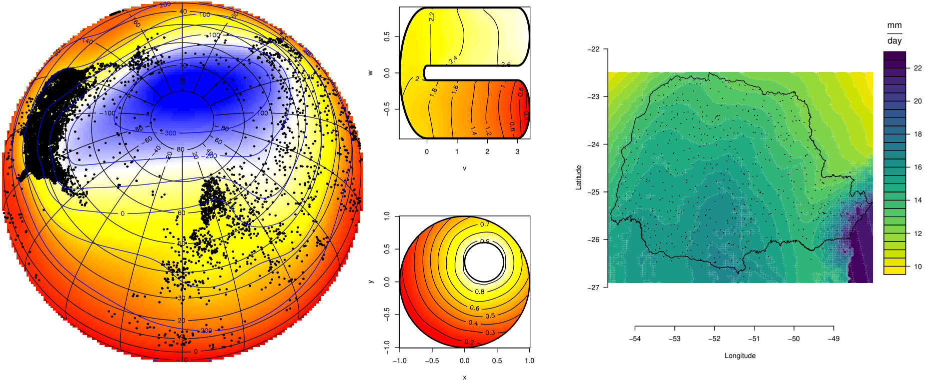
Ben Griffiths is working on Big Data (`bam`) methods for QGAMs.

For more software training material, see

`http://mfasiolo.github.io/qgam/articles/qgam.html`

`https://mfasiolo.github.io/mgcViz/articles/qgam_mgcViz.html`

THANK YOU!



Examples of quantile GAMs from Fasiolo et al. (2021a).

References I

Fasiolo, M., S. N. Wood, M. Zaffran, R. Nedellec, and Y. Goude (2021a). Fast calibrated additive quantile regression. *Journal of the American Statistical Association 116*(535), 1402–1412.

Fasiolo, M., S. N. Wood, M. Zaffran, R. Nedellec, and Y. Goude (2021b). qgam: Bayesian nonparametric quantile regression modeling in r. *Journal of statistical software 100*(9).

Koenker, R. (2005). *Quantile regression*. Number 38. Cambridge university press.

Wood, S. N. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73*(1), 3–36.

Wood, S. N., Y. Goude, and S. Shaw (2015). Generalized additive models for large data sets. *Journal of the Royal Statistical Society: Series C (Applied Statistics) 64*(1), 139–155.

Wood, S. N., Z. Li, G. Shaddick, and N. H. Augustin (2017). Generalized additive models for gigadata: modeling the uk black smoke network daily data. *Journal of the American Statistical Association 112*(519), 1199–1210.

References II

Wood, S. N., N. Pya, and B. Säfken (2016). Smoothing parameter and model selection for general smooth models. *Journal of the American Statistical Association 111*(516), 1548–1575.