The Journal of Open Source Software

gratia: An R package for exploring generalized additive models

Gavin L. Simpson[1]

1 Department of Animal and Veterinary Science, Aarhus University, Denmark

DOI: DOIunavailable

Software

- Review
- Repository
- Archive

Editor: Pending Editor

Reviewers:

- @Pending Reviewers

Submitted: N/A
Published: N/A

Summary

Generalized additive models (GAMs, Hastie & Tibshirani, 1990; Wood, 2017) are an extension of generalized linear models that allows the effects of covariates to be modelled as smooth functions. GAMs are increasingly used in many areas of science (e.g. Pedersen et al., 2019; Simpson, 2018) because the smooth functions allow nonlinear relationships between covariates and the response to be learned from the data through the use of penalized splines. Within the R (R Core Team, 2024) ecosystem, Simon Wood's *mgcv* package (Wood, 2017) is widely used to fit GAMs and is a *Recommended* package that ships with R as part of the default install. A growing number of other R packages build upon *mgcv*, for example as an engine to fit specialised models not handled by *mgcv* itself (e.g. *GJMR*, Marra & Radice, 2023), or to make use of the wide range of splines available in *mgcv* (e.g. *brms*, Bürkner, 2017).

The *gratia* package builds upon *mgcv* by providing functions that make working with GAMs easier. *gratia* takes a *tidy* approach (Wickham, 2014) providing *ggplot2* (Wickham, 2016) replacements for *mgcv*'s base graphics-based plots, functions for model diagnostics and exploration of fitted models, and a family of functions for drawing samples from the posterior distribution of a fitted GAM. Additional functionality is provided to facilitate the teaching and understanding of GAMs. The overall aim of *gratia* is to abstract away some of the complexity of working with GAMs fitted using *mgcv* to allow researchers to focus on using and interrogating their model rather than the technical R programming needed to achieve this.

Generalized additive models

A GAM has the form

$$\begin{aligned} y_i &\sim \mathcal{D}(\mu_i, \phi) \\ g(\mu_i) &= \mathbf{A}_i\boldsymbol{\gamma} + f_1(x_{1i}) + f_2(x_{2i}) + f_3(x_{3i}, x_{4i}) + \cdots \end{aligned}$$

where observations y_i are assumed to be conditionally distributed $\mathcal{D}$ with expectation $\mathbb{E}(y_i) = \mu_i$ and dispersion parameter ϕ. The expectation of y_i is given by a linear predictor of strictly parametric terms, whose model matrix is $\mathbf{A}_i$ with parameters $\boldsymbol{\gamma}$, plus a sum of $j = 1, \ldots, J$ smooth functions of covariates $f_j()$. $g()$ is a link function mapping values on the linear predictor to the scale of the response.

The smooth functions f_j are represented in the GAM using penalised splines, which are themselves formed as weighted sums of basis functions, $b_k()$, (De Boor, 2001) e.g.

$$f_j(x_{ij}) = \sum_{k=1}^{K} \beta_{jk} b_{jk}(x_{ij})$$

for a univariate spline. The weights, β_k, are model coefficients to be estimated alongside $\boldsymbol{\gamma}$. To avoid overfitting, estimates $\hat{\beta}_{jk}$ and $\hat{\boldsymbol{\gamma}}$ are sought to minimise the penalised log-likelihood

of the model

$$\ell_p(\boldsymbol{\beta}) = \ell(\boldsymbol{\beta}) - \frac{1}{2\phi}\sum_j \lambda_j \boldsymbol{\beta}_j^{\mathsf{T}} \mathbf{S}_j \boldsymbol{\beta}_j$$

where ℓ and ℓ_p are the log likelihood and penalized log likelihood, respectively, of the data at the parameter estimates, $\mathbf{S}_j$ are penalty matrices and λ_j are smoothing parameters associated with each smooth. Note that $\boldsymbol{\beta}$ now contains the coefficients $\boldsymbol{\gamma}$ and β_{jk}. $\boldsymbol{\beta}_j^{\mathsf{T}} \mathbf{S}_j \boldsymbol{\beta}_j$ measures the wiggliness of f_j, which, with the default basis-penalty combination, is the integrated squared second derivative of f_j. The smoothing parameters, $\boldsymbol{\lambda}$, control the trade-off between fit to the data and the complexity of the estimated functions.

The default spline created by *mgcv*'s `s()` is a low rank, thin plate regression spline (Wood, 2003). Figure 1, shows the basis functions for such a spline fitted to data simulated from the function

$$f = 0.2x^{11}\{10(1-x)\}^6 + 10(10x)^3(1-x)^{10}$$

with additive Gaussian noise ($\mu = 0, \sigma = 1$), and the associated penalty matrix, prepared using functions from *gratia*.

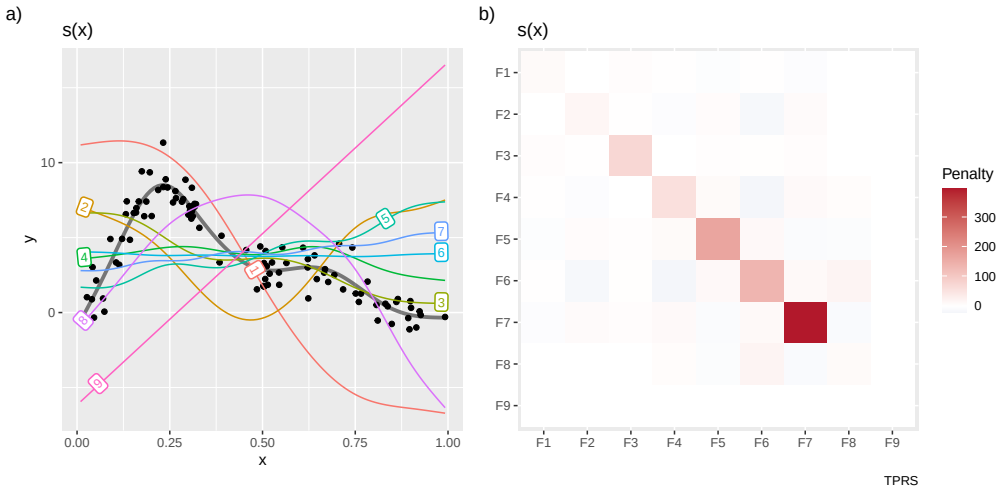

Figure 1: Basis functions (a) and associated penalty matrix (b) for a penalised, low rank, thin plate regression spline. a) shows the individual basis functions (thin coloured lines; numbers indicate which basis function each line represents) multiplied by their respective model coefficients, as well as the data (black points) to which the GAM was fitted. The estimated smooth is shown as the thick grey line. b) shows the penalty matrix for the basis shown in a). Note the 9th basis function (labelled 'F9,' which is the linear function at the lower left to upper right in a), is not affected by the penalty as it has 0 second derivative everywhere, and hence the resulting penalty for this function is 0.

Statement of need

mgcv is state-of-the-art software for fitting GAMs and their extensions to data sets on the order of millions of observations (e.g. Li & Wood, 2020; Wood, 2011; Wood et al., 2016). *mgcv* provides functions for plotting estimated smooth functions, as well as for producing model diagnostic plots. These functions produce plots using base graphics, the original plotting system for R. Additionally, *mgcv* returns fitted GAMs as complex list objects (see `?mgcv::gamObject`), the contents of which are not easily used for downstream analysis without careful study of *mgcv* and its help pages, plus a good understanding of GAMs themselves. The overall aim of *gratia* is to abstract away some of the complexity of working with GAMs fitted using *mgcv* to allow researchers to focus on using and interrogating their model rather than the technical R programming needed to achieve this. As a result, *gratia* is also increasingly being

used by researchers in many fields, and has, at the time of writing, been cited over 250 times (data from Google Scholar).

One of the motivations driving the development of *gratia* was to provide equivalent plotting capabilities using the *ggplot2* package (Wickham, 2016). To facilitate this, *gratia* provides functions for representing the model components as objects using *tidy* principles, which are suitable for plotting with *ggplot2* or manipulation by packages in the *tidyverse* (e.g. Wickham et al., 2023). This functionality allows for high-level plotting using the `draw()` method, as well as easily customisable plot creation using lower-level functionality.

Taking a Bayesian approach to smoothing with penalized splines (Kimeldorf & Wahba, 1970; Silverman, 1985; Wahba, 1983, 1985; see Miller, 2021 for a summary), it can be shown that GAMs fitted by *mgcv* are an empirical Bayesian model with a (usually improper) multivariate normal prior on the basis function coefficients. Samples from the posterior distribution of these models can be used to estimate the uncertainty in quantities derived from a GAM. This can be invaluable in applied research, where, for example, a quantity of interest may arise as an operation on predictions from the model. *gratia* provides functions for sampling from the posterior distribution of estimated smooths and from the model as a whole, where sampling can include the uncertainty in the estimated coefficients (`fitted_samples()`), the sampling uncertainty of the response (`predicted_samples()`), or both (`posterior_samples()`). By default, a Gaussian approximation to the posterior distribution is used, but a simple Metropolis Hastings sampler can be substituted (using `mgcv::gam.mh()`), which has better performance when the posterior is not well approximated by a Gaussian approximation.

The teaching of GAMs can benefit from visualisation of the spline basis functions and associated penalty matrices. *gratia* provides this functionality via `basis()` and `penalty()`, which can be applied either to a smooth specification (e.g. `s(x, z, bs = "ds")`) or to a fitted GAM (see Figure 1). These functions expose functionality already available in *mgcv*, but supply outputs in a tidy format, which makes access to these features more intuitive than the original implementations in *mgcv*. Additional utility functions are provided, for example: `model_constant()`, `edf()`, `model_edf()`, `overview()`, and `inv_link()`, which extract the model intercept term (or terms), the effective degrees of freedom of individual smooths and the overall model, shows a summary of the fitted GAM, and extracts the inverse of the link function(s) used, respectively.

State of the field

Several R packages provide similar functionality to that of *gratia*. Notably, the *mgcViz* package (Fasiolo, Nedellec, et al., 2020) provides sophisticated capabilities for plotting estimated smooths for models fitted by *mgcv* and *qgam* (Fasiolo, Wood, et al., 2020), and model diagnostics plots. A particular feature of *mgcViz* is that it was designed to be scalable, easily handling models fitted to data with millions of observations, a use case not currently well handled by *gratia*. *tidymv* (Coretta, 2023b) and its successor, *tidygam* (Coretta, 2023a), provide plots of estimated smooths and model predictions respectively, as well as differences of smooths, but the focus is on univariate smooths, in contrast to *gratia*'s ability to plot multivariate and specialist smooths (e.g. soap film smooths). Like *gratia*, the *itsadug* package (van Rij et al., 2022) is motivated to make working with estimated GAMs and related models easier. *itsadug* uses base graphics for plotting, but in place of partial effects plots, produces plots of adjusted predictions, and while useful for a broad range of models, is especially focused on GAMs fitted to longitudinal data with random effects.

Areas where *gratia* stands out from these other packages are i) the consistent functionality for a range of posterior simulations (*mgcViz* has some support for simulating new response values) as illustrated below, ii) support for computing derivatives of smooths, partial derivatives, and derivatives on the response scale (i.e. conditional derivatives), iii) tools for learning or teaching how GAMs work, iv) the range of utility functions that make working with GAMs easier, and

v) the simple and consistent *tidy* interface to all functionality.

Example usage

In this short example, I illustrate a few of the features of *gratia* using a data set of sea surface chlorophyll *a* measurements at a number of locations in the Atlantic Ocean, whose spatial locations are given as geographical coordinates (`lat` and `lon`), plus two additional covariates; `bathy`, the depth of the ocean, in metres, at the sampling location, and `jul.day`, the day of the year in which the observation was made. These data are in the `chl` dataset provided by the *gamair* package accompanying Wood (2017).

The packages required for this example are loaded, as is the data set `chl` with

```
pkgs <- c("mgcv", "gamair", "gratia", "ggplot2", "dplyr", "ggdist")
loaded <- vapply(pkgs, library, logical(1L), logical.return = TRUE,
  character.only = TRUE)
data(chl, package = "gamair")
```

A simple GAM for these data is to model the response (`chl`) with a spatial smooth of latitude (`lat`) and longitude (`lon`) as covariates. Here, I use a spline on the sphere (SOS) smoother built using a Duchon spline with second order derivative penalty (Duchon, 1977). Additional terms included in the linear predictor are a smooth of the day of year of sample collection (`jul.day`) and a smooth of ocean depth (`bath`). The response is assumed to be conditionally Tweedie distributed, with the power parameter (p) of the distribution estimated during fitting. Model coefficients and smoothing parameters are estimated using restricted maximum likelihood (Wood, 2011). To use parallel processing in some parts of the model fitting algorithm, the `nthreads` control parameter is set to 10 (set this lower if your machine doesn't have this many physical CPU cores).

```
ctrl <- gam.control(nthreads = 10)
m1 <- gam(
  chl ~ s(lat, lon, bs = "sos", m = -1, k = 150) +
    s(jul.day, bs = "cr", k = 20) +
    s(bath, k = 10),
  data = chl, method = "REML", control = ctrl, family = tw()
)
```

Model diagnostic plots can be produced using `appraise()`, which by default produces four plots: i) a QQ plot of model residuals, with theoretical quantiles and reference bands generated following Augustin et al. (2012), ii) a plot of residuals (deviance residuals are the default) against linear predictor values, iii) a histogram of residuals, and iv) a plot of observed versus fitted values. Model diagnostic plots for the model, with simulated residuals-based reference bands on the QQ plot, are produced with

```
appraise(m1, method = "simulate")
```

which show significant heteroscedasticity and departure from the conditional distribution of the response given the model, as seen in the increasing spread of the deviance residuals at larger values of the linear predictor in the upper right panel of Figure 2.

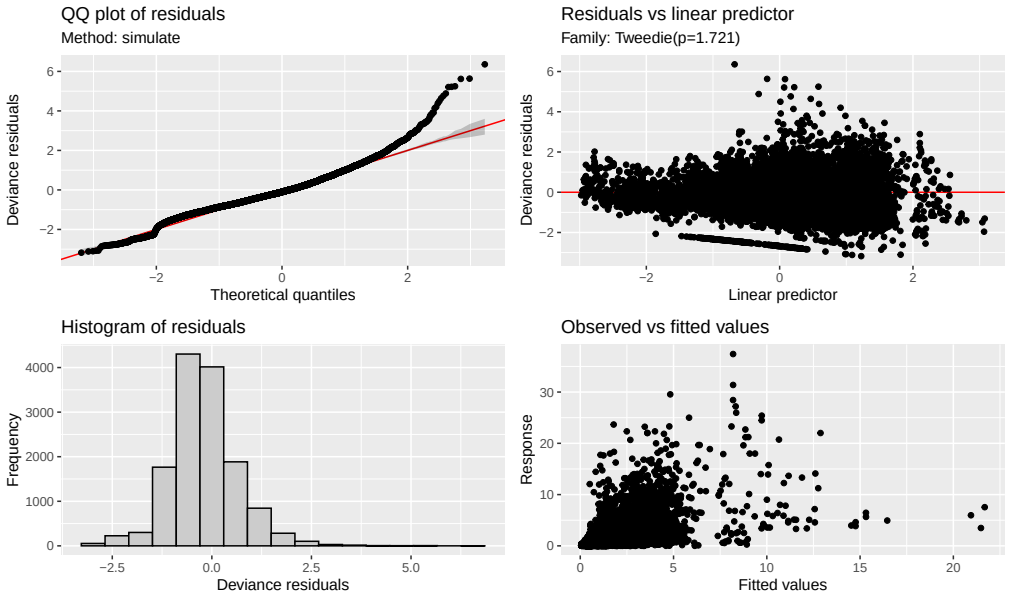

Figure 2: Model diagnostic plots for the GAM fitted to the ocean chlorophyll *a* data produced by the `appraise()` function. The four plots produced are: i) a QQ plot of model residuals, with theoretical quantiles and reference bands generated following Augustin et al. (2012) (upper left), ii) a plot of residuals (deviance residuals are default) against linear predictor values (upper right), iii) a histogram of deviance residuals (lower left), and iv) a plot of observed versus fitted values (lower right)

The problems with the model apparent in the diagnostics plots are probably due to important controls on chlorophyll *a* missing from the covariates available in the example data. However, the original model assumed constant values for the dispersion, ϕ, and the power parameter p, which may be too inflexible if missing covariates mean that important effects are not included in the model.

A distributional GAM, containing linear predictors for all distributional parameters,

$$
\begin{aligned}
y_i &\sim \mathcal{D}(\mu_i, p_i, \varphi_i) \\
g(\mu_i) &= \beta_1 + f_1(\text{lat}_i, \text{lon}_i) + f_2(\text{jul.day}_i) + f_3(\text{bath}_i) \\
g(p_i) &= \beta_2 + f_4(\text{lat}_i, \text{lon}_i) + f_5(\text{jul.day}_i) + f_6(\text{bath}_i) \\
g(\varphi_i) &= \beta_3 + f_7(\text{lat}_i, \text{lon}_i) + f_8(\text{jul.day}_i) + f_9(\text{bath}_i)
\end{aligned}
$$

may improve the model diagnostics.

A distributional GAM for $\mathcal{D}$ Tweedie with linear predictors for μ, p, and φ is fitted below using *mgcv*'s `twlss()` family

```
m2 <- gam(
  list(
    chl ~ s(lat, lon, bs = "sos", m = -1, k = 150) + # location
      s(jul.day, bs = "cr", k = 20) + s(bath, k = 10),
    ~ s(lat, lon, bs = "sos", m = -1, k = 100) +     # power
      s(jul.day, bs = "cr", k = 20) + s(bath, k = 10),
    ~ s(lat, lon, bs = "sos", m = -1, k = 100) +     # scale
      s(jul.day, bs = "cr", k = 20) + s(bath, k = 10)
  ),
  data = chl, method = "REML", control = ctrl, family = twlss()
)
```

This model has much better model diagnostics although some large residuals remain (Figure 3). Note that the QQ plot uses theoretical quantiles from a standard normal distribution as the simulation-based values are not currently available in *mgcv* or *gratia* for some of the distributional families, including the `twlss()` family, and as such, the reference bands may not be appropriate.

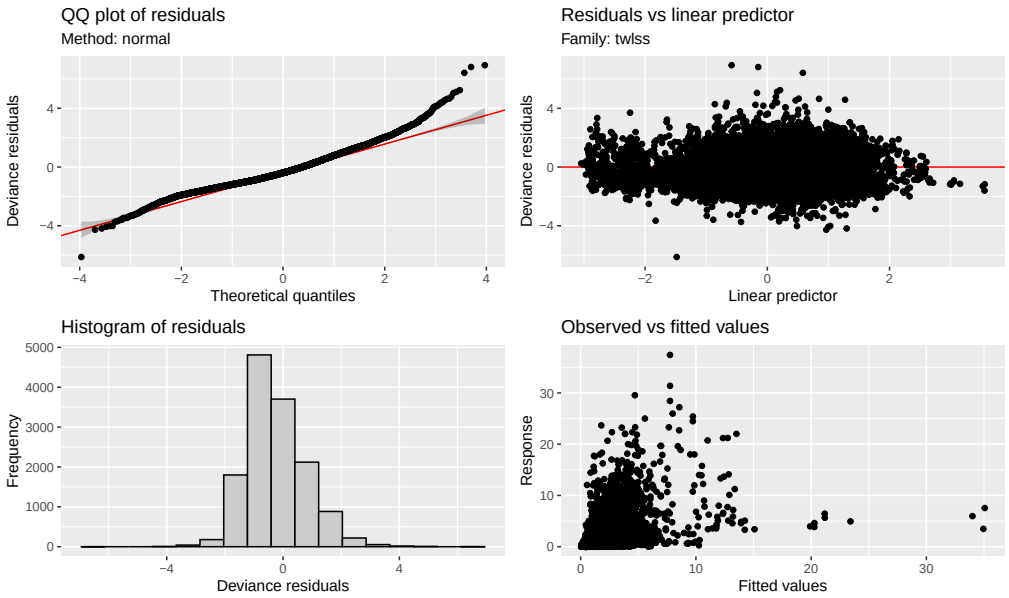

Figure 3: Model diagnostic plots for the distributional GAM fitted to the ocean chlorophyll *a* data produced by the `appraise()` function. Refer to the caption for Figure 2 for a description of the plots shown.

gratia can handle distributional GAMs fitted with *mgcv*, and also those fitted using *GJRM*'s `gamlss()`. Below, the estimated smooths from `m2` are plotted using `draw()`

```
# Define the coordinate system to use for plotting on a map
crs <- "+proj=ortho +lat_0=20 +lon_0=-40"
draw(m2, crs = crs, default_crs = 4326, dist = 0.05, rug = FALSE)
```

The plots produced by `draw()` from a fitted model are known as *partial effect plots* (Figure 4), which show the component contributions, on the link scale, of each model term to the linear predictor. The y axis on these plots is typically centred around 0 due to most smooths having a sum-to-zero identifiability constraint applied to them. This constraint is what allows the model to include multiple smooths and remain identifiable. These plots allow you to read off the contributions of each smooth to the fitted response (on the link scale); they show link-scale *predictions* of the response for each smooth, conditional upon all other terms in the model, including any parametric effects and the intercept, having zero contribution. In the parlance of the *marginaleffects* package (Arel-Bundock et al., Forthcoming), these plots show *adjusted predictions*, just where the adjustment includes setting the contribution of *all* other model terms to the predicted value to zero. For *partial derivatives* (what *marginaleffects* would call a *marginal effect* or slope), *gratia* provides `derivatives()`.

Here, we see a specialised plot drawn for spline-on-the-sphere smooths $f(\texttt{lat}_i, \texttt{lon}_i)$ (Figure 4, left-hand column), which uses `ggplot2::coord_sf()` and functionality from the *sf* package (Pebesma, 2018; Pebesma & Bivand, 2023) to visualise the smooth via an orthographic projection.

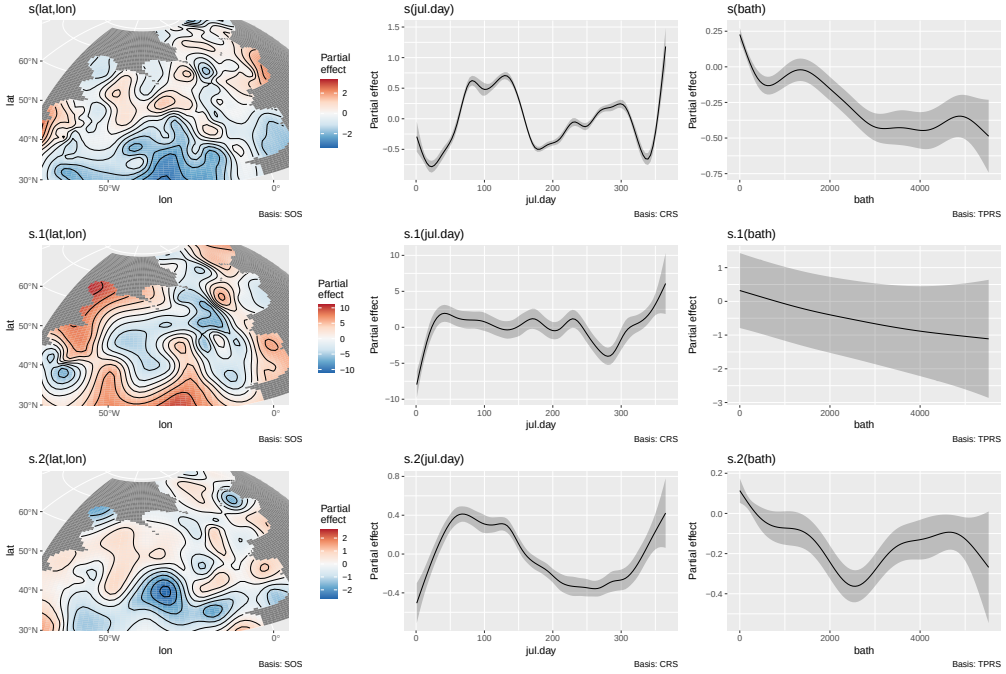

Figure 4: Estimated smooth functions for the distributional GAM, `m2`, fitted to the ocean chlorophyll *a* data. The first row of plots is for the linear predictor of the conditional mean chlorophyll *a*, while the second and third rows are for the conditional power parameter and conditional scale, respectively. The shaded ribbons are 95% Bayesian credible intervals.

If the provided plots are insufficient for users' needs, lower-level functionality is provided by *gratia* to facilitate bespoke plotting with *ggplot2*. For example, to evaluate the SOS smooth at a grid (50x50) of values over the range of the covariates, we use `smooth_estimates()` and add a Bayesian credible interval with `add_confint()`:

```
smooth_estimates(m2, select = "s(lat,lon)", n = 50) |>
  add_confint()
```

This returns a data frame of the requested values that is amenable for plotted with `ggplot()`.

Posterior sampling

Perhaps we are interested in the average expected chlorophyll *a* between 40–50 degrees N and 40–50 degrees W. An estimate for this value can be computed directly from the fitted model as follows. First create a slice through the data for the spatial locations were are interested in using the `data_slice()` function, which ensures that `ds` contains everything we need to predict from the fitted model

```
ds <- data_slice(m2,
  lat = evenly(lat, lower = 40, upper = 50, by = 0.5),
  lon = evenly(lon, lower = -50, upper = -40, by = 0.5))
```

Next, `fitted_values()` returns the predicted values at the specified locations. I only include the spatial effects, excluding the effects of ocean depth and day of year, and ignore terms in the other linear predictors as they do not affect the expected chlorophyll *a*:

```
use <- c("(Intercept)", "s(lat,lon)")
fv <- fitted_values(m2, data = ds, terms = use) # predict
```

Finally, I summarise the predictions for the location parameter to yield the average of the predicted values

```
fv |>
  filter(.parameter == "location") |>
  summarise(chl_a = mean(.fitted))
```

```
## # A tibble: 1 x 1
##   chl_a
##   <dbl>
## 1  1.07
```

While this is an acceptable answer to the question, it lacks an uncertainty estimate. This is where posterior sampling is useful. With a small modification of the above code and a little data wrangling, we can produce an uncertainty estimate, using `fitted_samples()` to generate posterior draws of the expected chlorophyll *a*:

```
fs <- fitted_samples(m2,      # model
  data = ds,                  # values of covariates to predict at
  terms = use,                # which terms to include in predictions
  n = 10000,                  # number of posterior draws
  method = "gaussian",        # Gaussian approximation to the posterior
  unconditional = TRUE,       # incl uncertainty for estimating lambda
  n_cores = 4,                # how many CPU cores to compute MVN samples
  seed = 342)                 # set the random number seed, used internally
```

The posterior draws can then be summarised as before, except now the average chlorophyll *a* is calculated separately for each posterior draw (`.draw`)

```
fs |>                                 # take the posterior draws
  group_by(.draw) |>                  # group them by `.draw`
  summarise(chl_a = mean(.fitted)) |> # compute mean of fitted chl a
  ggdist::median_qi()                 # summarise posterior
```

```
## # A tibble: 1 x 6
##   chl_a .lower .upper .width .point .interval
##   <dbl>  <dbl>  <dbl>  <dbl> <chr>  <chr>
## 1 0.932  0.757   1.17   0.95 median qi
```

The posterior distribution of average chlorophyll *a* is summarised using `median_qi()` from the *ggdist* package (Kay, 2024a, 2024b). While we could use the base R function `quantile()` to compute the interval, the use of `median_qi()` illustrates how *gratia* tries to interact with other packages.

Conclusion

gratia provides a range of functionality to make working with estimated GAMs easier for users. It is designed to take some of the pain out of working with models, simplifying plotting of smooths and related features (differences, derivatives) and exposing the powerful machinery of the *mgcv* package without the need for a deep understanding of GAMs, splines, and the inner structure of *mgcv*'s model objects. As such, it provides a useful addition for anyone wanting to use GAMs in their data analyses without requiring them to be a GAM expert.

Acknowledgements

Development of *gratia* was supported by a Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant to the author (RGPIN-2014-04032).

References

Arel-Bundock, V., Greifer, N., & Heiss, A. (Forthcoming). How to interpret statistical models using marginaleffects in R and Python. *Journal of Statistical Software*.

Augustin, N. H., Sauleau, E.-A., & Wood, S. N. (2012). On quantile quantile plots for generalized linear models. *Computational Statistics & Data Analysis*, *56*(8), 2404–2409. https://doi.org/10.1016/j.csda.2012.01.026

Bürkner, P.-C. (2017). brms: An R package for bayesian multilevel models using stan. *Journal of Statistical Software, Articles*, *80*(1), 1–28. https://doi.org/10.18637/jss.v080.i01

Coretta, S. (2023a). *tidygam: Tidy prediction and plotting of generalised additive models*.

Coretta, S. (2023b). *tidymv: Tidy model visualisation for generalised additive models*.

De Boor, C. (2001). *A practical guide to splines* (1st ed.). Springer. ISBN: 9780387953663

Duchon, J. (1977). Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In *Constructive theory of functions of several variables* (pp. 85–100). Springer, Berlin, Heidelberg. https://doi.org/10.1007/BFb0086566

Fasiolo, M., Nedellec, R., Goude, Y., & Wood, S. N. (2020). Scalable visualization methods for modern generalized additive models. *Journal of Computational and Graphical Statistics*, *29*(1), 78–86. https://doi.org/10.1080/10618600.2019.1629942

Fasiolo, M., Wood, S. N., Zaffran, M., Nedellec, R., & Goude, Y. (2020). Fast calibrated additive quantile regression. *Journal of the American Statistical Association*, 1–11. https://doi.org/10.1080/01621459.2020.1725521

Hastie, T. J., & Tibshirani, R. J. (1990). *Generalized additive models*. Chapman & Hall / CRC. ISBN: 9781351445962

Kay, M. (2024a). ggdist: Visualizations of distributions and uncertainty in the grammar of graphics. *IEEE Transactions on Visualization and Computer Graphics*, *30*(1), 414–424. https://doi.org/10.1109/TVCG.2023.3327195

Kay, M. (2024b). *ggdist: Visualizations of distributions and uncertainty*. Zenodo. https://doi.org/10.5281/ZENODO.10782896

Kimeldorf, G. S., & Wahba, G. (1970). A correspondence between bayesian estimation on stochastic processes and smoothing by splines. *Annals of Mathematical Statistics*, *41*(2), 495–502.

Li, Z., & Wood, S. N. (2020). Faster model matrix crossproducts for large generalized linear models with discretized covariates. *Statistics and Computing*, *30*(1), 19–25. https://doi.org/10.1007/s11222-019-09864-2

Marra, G., & Radice, R. (2023). *GJRM: Generalised joint regression modelling* (pp. R package version 0.2–6.4).

Miller, D. L. (2021). Bayesian views of generalized additive modelling. *arXiv [Stat.ME]*. http://arxiv.org/abs/1902.01330

Pebesma, E. (2018). Simple features for R: Standardized support for spatial vector data. *The R Journal*, *10*(1), 439. https://doi.org/10.32614/rj-2018-009

Pebesma, E., & Bivand, R. (2023). *Spatial data science: With applications in R* (1st Edition). Chapman; Hall/CRC. https://doi.org/10.1201/9780429459016

Pedersen, E. J., Miller, D. L., Simpson, G. L., & Ross, N. (2019). Hierarchical generalized additive models in ecology: An introduction with mgcv. *PeerJ*, *7*, e6876. https://doi.org/10.7717/peerj.6876

R Core Team. (2024). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Silverman, B. W. (1985). Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, *47*(1), 1–52.

Simpson, G. L. (2018). Modelling palaeoecological time series using generalised additive models. *Frontiers in Ecology and Evolution*, *6*, 149. https://doi.org/10.3389/fevo.2018.00149

van Rij, J., Wieling, M., Baayen, R. H., & van Rijn, H. (2022). *itsadug: Interpreting time series and autocorrelated data using GAMMs*.

Wahba, G. (1983). Bayesian "confidence intervals" for the cross-validated smoothing spline. *Journal of the Royal Statistical Society*, *45*(1), 133–150. https://doi.org/10.1111/j.2517-6161.1983.tb01239.x

Wahba, G. (1985). A comparison of GCV and GML for choosing the smoothing parameter in the generalized spline smoothing problem. *Annals of Statistics*, *13*(4), 1378–1402. https://doi.org/10.1214/AOS/1176349743

Wickham, H. (2014). Tidy data. *Journal of Statistical Software*, *59*(10), 1–23. https://doi.org/10.18637/jss.v059.i10

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer International Publishing. https://doi.org/10.1007/978-3-319-24277-4

Wickham, H., Cetinkaya-Rundel, M., & Grolemund, G. (2023). *R for data science: Import, tidy, transform, visualize, and model data* (2nd ed.). O'Reilly Media. ISBN: 9781492097402

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. https://doi.org/10.1111/j.1467-9868.2010.00749.x

Wood, S. N. (2003). Thin plate regression splines. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, *65*(1), 95–114. https://doi.org/10.1111/1467-9868.00374

Wood, S. N. (2017). *Generalized additive models: An introduction with R, second edition*. CRC Press. ISBN: 9781498728379

Wood, S. N., Pya, N., & Säfken, B. (2016). Smoothing parameter and model selection for general smooth models. *Journal of the American Statistical Association*, *111*(516), 1548–1563. https://doi.org/10.1080/01621459.2016.1180986