Georg-August Universität Göttingen

bamlss.vis: An R Package to Interactively Analyze and Visualize Bayesian Additive Models for Location, Scale and Shape (BAMLSS) Using the Shiny Framework

20 week Master thesis as part of the
Master of Science (M.Sc.) course "Applied Statistics"
at the University of Göttingen

Author:
Stanislaus Stadlmann,
Student ID: 21144637

Supervisors
Prof. Dr. Thomas Kneib
Dr. Nadja Klein

Submitted on December 18, 2017
by Stanislaus Stadlmann,
born in Vienna, Austria

GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Contents

List of Figures

Acronyms

AM Additive Models

BAMLSS Bayesian Additive Models for Location, Scale and Shape

cdf cumulative distribution function

CG Cole and Green

GAM Generalized Additive Models

GAMLSS Generalized Additive Models for Location, Scale and Shape

GLM Generalized Linear Models

i.i.d. independent and identically distributed

IRLS Iteratively Reweighted Least Squares

mcmc Markov Chain Monte Carlo

OLS Ordinary Least Squares

pdf probability density function

RS Rigby and Stasinopoulos

STAR Structured Additive Regression

1 Introduction

Aided by an exponential increase in gathered data and dramatic improvements in computing power, the recent past has seen numerous advancements in statistical sciences. New computational-heavy methods, such as Random Forests or Neural Networks can finally be applied on an affordable scale, while more established models are used by a growing number of statistical users. In the United States of America, for example, the number of jobs classified as statisticians has increased by more than 120% in the years from 1997 to 2016 (Bureau of Labor Statistics, 2016).

One of the new emerging fields is distributional regression, where not only the mean but each parameter of a response distribution can be modeled using a set of predictors (Klein et al., 2015). Notable frameworks called Generalized Additive Models for Location, Scale and Shape (GAMLSS) and Bayesian Additive Models for Location, Scale and Shape (BAMLSS) were invented by Rigby and Stasinopoulos (2001) in the form of a frequentist perspective and Umlauf, Klein, and Zeileis (2017) using a Bayesian approach, respectively.

Since methods have become increasingly more complex and capable over the years, it is important to make them accessible and understandable to the growing number of statistical users. In the case of distributional regression models, the interpretation of covariate effects on response moments and the expected conditional response distribution is harder than with traditional methods such as Ordinary Least Squares (OLS) or Generalized Linear Models (GLM), since the moments of a distribution often do not directly equate the modeled parameters but are rather a combination of them with a varying degree of complexity.

This thesis will introduce a framework for the visualization of distributional regression models fitted using the `bamlss` R package (Umlauf et al., 2017) as well as feature an implementation as an R extension titled bamlss.vis. The main goals of this framework are the abilities to:

1. See and compare the expected distribution for chosen sets of covariates
2. View the direct relationship between moments of the response distribution and a chosen explanatory variable, given a set of covariates.

Additionally, the user can obtain the code which created the visualizations described above to potentially reproduce them later. The implementation will be

done using the statistical software R (R Core Team, 2017) in the form of a Shiny application (Chang et al., 2017).

This thesis will be further structured as follows: Section 2 and its subchapters will give an overview of the models which Bayesian Additive Models for Location, Scale and Shape are related to and built upon. Section 2.5 explains the estimation procedures behind the introduced models. In Section 3, bamlss.vis is described by first creating a model in a case-study (Section 3.2) with a real dataset. Then, this model is used in Section 3.3 for a walk-through of bamlss.vis' main abilities. Afterwards, Section 3.4 expands Section 3.3 by showcasing additional functions, which cannot be seen using the case-study model.

2 Bayesian Additive Models for Location, Scale and Shape

Bayesian Additive Models for Location, Scale and Shape are a form of Bayesian regression models in which every parameter of a parametric distribution with K parameters is related to a set of additive predictors. The distribution does not have to follow the exponential family, which extends the distributions available for modeling beyond the ones used in Generalized Linear Models (GLM). In similar fashion to Generalized Additive Models (GAM, Hastie and Tibshirani, 1990), the additive predictors can assume different shapes, including non-linear, fixed, random and spatial effects (Umlauf et al., 2017).

In the ability to additively model multiple parameters of one distribution, BAMLSS bear many similarities with Generalized Additive Models for Location, Scale and Shape (GAMLSS, Rigby and Stasinopoulos, 2001). Disparities lie in the estimation of model parameters, where BAMLSS utilize Markov Chain Monte Carlo (mcmc) simulations which provide credible intervals in situations where asymptotic maximum likelihood confidence intervals often fail, as well as BAMLSS' ability to model multivariate parametric distributions (Umlauf et al., 2017).

To give a sufficient depiction of the BAMLSS model class, this section will start with explaining Additive Models and then gradually generalize the broader frameworks to finally arrive at BAMLSS. Furthermore, a brief overview of the different estimation techniques for the covered model frameworks will be given (Section 2.5).

2.1 Additive Models

BAMLSS can be seen as a generalization of Structured Additive Regression (STAR) models, which are in turn a generalization of Additive Models (AM). Additive Models, first proposed by Friedman and Stuetzle (1981) represent a model type in which a dependent variable y is related to a set of non-parametric predictors in an additive way. Assuming conditional independence of $y_1, \ldots, y_n$ given the explanatory variables $z_1, \ldots, z_K$, we obtain the following model equation:

$$y_i = f_1(z_{i1}) + f_2(z_{i2}) + \ldots + f_k(z_{iK}) + \epsilon_i \tag{2.1}$$

where $f_j(\cdot)$ depicts unspecified non-parametric functions of covariate z_j, which can include smoothing splines or local regression approaches. Specifying the predictors additively is more efficient than using multi-dimensional smoothers, as the computational power and size of the dataset needed for fitting would increase exponentially with every new dimension. This makes additive models more flexible compared to standard linear regression, while still being more interpretable than non-additive models (Buja et al., 1989).

Fahrmeir et al. (2013) suggest that an Additive Model can also include parametric components. Given covariates $x_1, \ldots, x_Q$ with linear effects on y, we can extend (2.1) to a semi-parametric regression model with the following specification:

$$y_i = \sum_{j=1}^{K} f_j(z_{ij}) + \underbrace{\mathbf{x}_i'\boldsymbol{\beta}}_{\beta_0+\beta_1 x_{i1}+\ldots+\beta_Q x_{iQ}} + \epsilon_i \tag{2.2}$$

Eq. (2.2) combines non-parametric and parametric components. Because the model would otherwise not be identified, functions $f_j(\cdot)$ now have to be centered around zero, such that the restriction

$$\sum_{i=1}^{n} f_1(x_{i1}) = \ldots = \sum_{i=1}^{n} f_K(x_{iK}) = 0$$

holds. The functions $f_j(\cdot)$ are approximated using Basis functions in the following scheme

$$f_j(z_j) = \sum_{m=1}^{d_j} \mathbf{B}_m(z_j)\gamma_{jm}$$

where $\mathbf{B}_m(z_j)$ represents a smooth function base for variable z_j and γ_{jm} denotes

its regression coefficient. This allows to write the Additive Model in a matrix form, indifferent of the chosen basis:

$$\mathbf{y} = \sum_{j=1}^{K} \mathbf{Z}_j \boldsymbol{\gamma}_j + \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{2.3}$$

Here, the design matrices $\mathbf{Z}_1, \ldots, \mathbf{Z}_K$ represent the Basis functions assessed at different covariates. $\mathbf{X}$ is constructed in equivalence to the standard linear regression model. Assumptions about the error term of a semi-parametric Additive Model are also similar to the classic linear model, where ϵ_i are independent and identically distributed (i.i.d.) normally distributed with $E(\epsilon_i) = 0$ and $Var(\epsilon_i) = \sigma^2$. These properties are then also valid for the response variable, so that $y_i \overset{i.i.d.}{\sim} N(\hat{y}, \sigma_y^2)$ (Fahrmeir et al., 2013, chap. 9.1).

2.2 Structured Additive Regression Models

The non-parametric components in Additive Models open the possibility for more flexible relationships between the dependent variable and single explanatory variables, which standard linear regression methods might not capture correctly. However, sometimes the area of model application requires even more flexibility, for example by including spatial covariates, fixed/random effects or interaction terms. These specific types of effects extend the Additive Model to a Structured Additive Regression Model (Fahrmeir et al., 2003, STAR). This chapter will briefly describe its different components.

2.2.1 Spatial Effects

Similarly to Section 2.1, observations (y_i, z_{ij}, x_{iq}) are given, where x_j and z_j depict covariates whose effects are to be modeled using linear and non-parametric predictors, respectively. Additionally, a geographic location index s is known with observations s_i, which can be either discrete (e.g. region or country) as well as continuous (e.g. longitude/latitude). Extending the semi-parametric Additive

Model as specified in (2.2) a geospatial effect is now added:

$$\begin{aligned} y_i &= \sum_{j=1}^{K} f_j(z_{ij}) + \mathbf{x}_i'\boldsymbol{\beta} + f_{geo}(s_i) + \epsilon_i \\ &= \kappa^{add} + f_{geo}(s_i) + \epsilon_i \end{aligned} \tag{2.4}$$

κ^{add} includes the non-spatial effects from (2.2). The newly added spatial effect, $f_{geo}(\cdot)$, is often viewed as a proxy for unknown covariates, such as altitude or climate data. If the geographic location index s is tracked using discrete values, $f_{geo}(\cdot)$ could represent a Markov random field. For continuous values, smoothing techniques such as Kriging (Matheron, 1963) or a multivariate tensor product spline are available. In both the discrete and the continuous case, the vector of geo-additive components $\boldsymbol{f}_{geo}$ can be written as

$$\boldsymbol{f}_{geo} = \mathbf{Z}_{geo}\boldsymbol{\gamma}_{geo}$$

so that it can be incorporated into the geoadditive model in matrix notation in the following way

$$\mathbf{y} = \sum_{j=1}^{K} \mathbf{Z}_j\boldsymbol{\gamma}_j + \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}_{geo}\boldsymbol{\gamma}_{geo} + \boldsymbol{\epsilon} \tag{2.5}$$

which bears similarities to the Basis function approach in (2.3) (Fahrmeir et al., 2013, chap. 9.2).

2.2.2 Interaction Terms

The regression equation (2.2) of Additive Models included main non-parametric and parametric effects but no interactions between covariates. When incorporating interaction effects, one has to differentiate between an interaction among a continuous and a categorical variable, as well as one where two continuous variables share a common effect (Fahrmeir et al., 2013, chap. 9.3).

To illustrate the first case, it is assumed that z_1 and x_1 are continuous and binary ($x_i \in (0, 1)$) covariates, respectively. Then, the interaction term $f_{z_1|x_1}(z_1) \cdot x_1$ can

be included in the Additive Model from (2.2) in the following way:

$$y_i = \sum_{j=1}^{K} f_j(z_{ij}) + \mathbf{x}_i'\boldsymbol{\beta} + \underbrace{\begin{matrix} 0 \\ f_{z_1|x_1}(z_{i1}) \end{matrix}}_{\begin{matrix} \text{if } x_{i1}=0 \\ \text{if } x_{i1}=1 \end{matrix}} f_{z_1|x_1}(z_{i1})x_{i1} + \epsilon_i$$

If $x_1 = 0$, the non-linear effects of z_1 are now

$$\begin{matrix} f_1(z_1) & \text{if } x_1 = 0 \\ f_1(z_1) + f_{z_1|x_1}(z_1) + \beta_1 & \text{if } x_1 = 1 \end{matrix}$$

This framework can also incorporate spatially covarying terms, where the interaction term $f_{geo|x_1}(s)$ represents an interaction between the location variable s and a categorical variable x_1 (Fahrmeir et al., 2013).

Using a Basis function approach, the vector of interaction effects

$$\boldsymbol{f}_{int} = (f_{z_1|x_1}(z_{11})x_{11}, \ldots, f_{z_1|x_1}(z_{n1})x_{n1})'$$

can be described in matrix notation to extend (2.3) in the following way:

$$\mathbf{y} = \sum_{j=1}^{K} \mathbf{Z}_j\boldsymbol{\gamma}_j + \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}_{int}\boldsymbol{\gamma}_{int} + \boldsymbol{\epsilon}$$

Here, the design matrix $\boldsymbol{Z}_{int}$ represents the Basis function values multiplied with x_1 observations (Fahrmeir et al., 2013, chap. 9.3).

The possibility of interactions between two continuous covariates is also given. In this case, the interaction between z_1 and z_2 is modeled using a two-dimensional non-parametric function $f_{z_1|z_2}(z_1, z_2)$. Common two-dimensional functions include bi-variate smooth splines and Kriging techniques. When only the two-dimensional functions without main effects ($f_1(z_1)$, $f_2(z_2)$) should be included, the model equation assumes the following form:

$$y_i = f_{z_1|z_2}(z_{i1}, z_{i2}) + f_3(z_{i3}) + \ldots + f_K(z_{iK}) + \mathbf{x}_i'\boldsymbol{\beta} + \epsilon_i \tag{2.6}$$

For reasons of identifiability, $f_{z_1|z_2}(z_1, z_2)$ also needs to be centered around zero. Fahrmeir, Kneib, Lang, and Marx (2013, chap. 9.3.2) warn that for estimation of models with two-dimensional surfaces a high sample size with combinations of z_1 and z_2 is required. In cases where this requirement is not fulfilled, a simple

main effects model as in (2.2) is preferred.

It is also possible to model the interaction effect of z_1 and z_2 using the two-dimensional surface $f_{z_1|z_2}(z_1, z_2)$ while still including the main effects. In this scenario, the model is specified as follows:

$$y_i = f_{z_1|z_2}(z_{i1}, z_{i2}) + f_1(z_{i1}) + f_2(z_{i2}) + \sum_{j=3}^{K} f_j(z_{ij}) + \sum_{l=1}^{Q} \beta_l x_{il} + \epsilon_i \tag{2.7}$$

The identifiability problem in this model is now more complex than before. To solve it, Fahrmeir et al. (2013, chap. 9.3) state that not only all included functions have to be centered around zero but also "all slices of the interaction $f_{z_1|z_2}(z_1, z_2)$, i.e. all one-dimensional smooths with fixed value of z_1 or z_2".

Using the non-parametric function approach, the matrix representation of the model can be obtained:

$$\mathbf{y} = \sum_{j=1}^{K} \mathbf{Z}_j \boldsymbol{\gamma}_j + \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}_{z_1|z_2} \boldsymbol{\gamma}_{z_1|z_2} + \boldsymbol{\epsilon} \tag{2.8}$$

with interaction term design matrix $\mathbf{Z}_{z_1|z_2}$ (Fahrmeir et al., 2013, chap. 9.3).

2.2.3 Random Effects

When dealing with longitudinal datasets it is often necessary to model cluster-specific similarities using Random Effects (Laird and Ware, 1982). Additive Models can also be expanded upon using Random Effects to arrive at so called Additive Mixed Models. Assuming a longitudinal data structure with subjects $j = 1, ..., n_i$ in clusters $i = 1, ..., m$ and covariates $x_1, \ldots, x_Q$, a parametric random coefficient model possesses the following structure:

$$y_{ij} = (\beta_0 + \nu_{0i}) + (\beta_1 + \nu_{1i})x_{ij1} + \ldots + (\beta_Q + \nu_{Qi})x_{ijQ} + \epsilon_i$$

The "random" coefficients ν_{0i} (intercept) and $\nu_{1i}, \ldots, \nu_{Qi}$ (slopes) represent the cluster-specific deviations from the main effects. To obtain Additive Mixed Models, the main effects are then replaced with non-parametric functions:

$$y_{ij} = f_1(x_{ij1}) + \ldots + f_Q(x_{ijQ}) + \nu_{0i} + \nu_{1i}x_{ij1} + \ldots + \nu_{Qi}x_{ijQ} + \epsilon_i \tag{2.9}$$

Like non-parametric main effects, Random Effects also have a matrix notation. In the case where every main effect is also modeled with cluster-specific effects, the matrix form of Additive Mixed Models is as follows:

$$\mathbf{y} = \sum_{j=1}^{K} \mathbf{Z}_j \boldsymbol{\gamma}_j + \mathbf{R}_0 \boldsymbol{\nu}_0 + \sum_{j=1}^{K} \mathbf{R}_j \boldsymbol{\nu}_j + \boldsymbol{\epsilon}$$

Here, $\boldsymbol{\nu}_0 = (\nu_{01}, \ldots, \nu_{0m})'$ and $\boldsymbol{\nu}_j = (\nu_{j1}, \ldots, \nu_{jm})'$ represent the Random Effects coefficients. A more in-depth look at the structure of the design matrices is given by Fahrmeir et al. (2013, chap. 9.4, p. 550)

2.3 Generalized Structured Additive Regression Models

Structured Additive Regression (STAR) models extend simple Additive Models with special model terms briefly introduced in the previous sections. These effects include:

- Nonlinear effects of a univariate continuous variable
- Spatial effects of location index s
- Interactions between a continuous covariate and a categorical variable
- Nonlinear interactions between two continuous covariates
- Random Effects with intercept ν_0 and slope ν_j deviations from main effects

All of the aforementioned model terms can be included in a STAR interchangeably, including parametric predictors $\mathbf{X}\boldsymbol{\beta}$ (Fahrmeir et al., 2013, chap 9.5).

STAR models provide very flexible ways of modeling the influence of explanatory variables on a given response variable y. Note that while the components can be non-parametric, the direct modeling of y assumes that the response variable follows a Gaussian distribution. However, when dealing with binary or categorical responses for example, this assumption is violated. In this case, a type of model specification is needed that directly upholds the dependent variables' support (Olsson, 2002, chap. 2). To solve this challenge STAR models are merged with Generalized Linear Models to Generalized STAR models.

Generalized Linear Models (GLM), first coined by Nelder and Wedderburn (1972), introduce a framework where the expectation of response y is related to a lin-

ear predictor $\eta_i = \mathbf{x}_i'\boldsymbol{\beta}$ via a link function $g(E(y_i)) = g(\mu_i) = \eta_i$ or a response function $h = g^{-1}$ to arrive at the following model specification:

$$\begin{aligned} \mu_i &= h(\mathbf{x}_i'\boldsymbol{\beta}) \quad \text{or} \\ g(\mu_i) &= \mathbf{x}_i'\boldsymbol{\beta} \end{aligned} \tag{2.10}$$

For example, when modeling a binomially distributed response the probability parameter π, which has a support of $\pi \in [0, 1]$, is related to linear predictors $\mathbf{X}\boldsymbol{\beta}$. Using a logistic link function, we obtain a Logistic Regression Model:

$$\begin{aligned} \eta_i &= \mathbf{x}_i'\boldsymbol{\beta} \\ E(y_i) = \pi_i &= \frac{exp(\eta_i)}{1 + exp(\eta_i)} \end{aligned}$$

Here, the response function ensures the correct support of π (Fahrmeir et al., 2013, chap. 5). Using a link function, the expectation of y can be the first moment of any discrete or continuous distribution part of the exponential family, such as the Poisson, Binomial and Gamma distribution (Rigby and Stasinopoulos, 2005).

Note in (2.10) that the effects of covariates $x_1, \ldots, x_Q$ are modeled parametrically. Generalized Additive Models, as suggested by Hastie and Tibshirani (1990), extend the class of Generalized Linear Models to allow for non-parametric effects. In particular, the linear predictor $\eta_i = \mathbf{x}_i'\boldsymbol{\beta}$ is interchanged by smooth non-parametric functions $f_j(\cdot)$. Given response variable y and covariates $z_1, \ldots, z_K$, the following model specification is obtained:

$$\begin{aligned} \eta_i &= \sum_{j=1}^{K} f_j(x_{ij}) \\ \mu_i &= E(y_i) = h(\eta_i) \end{aligned} \tag{2.11}$$

Now, many different response distributions as well as flexible effects for explanatory variables are supported to create a highly flexible model framework. In (2.11) only non-parametric effects are linked to η_i. However, given response y and covariates (x_q, z_j) all specific effects of STAR models (spatial effects $f_{geo}(\cdot)$, interactions $f_{int}(\cdot)$, etc.) as well as parametric coefficients can be combined to form a

Generalized Structured Additive Regression Model (Generalized STAR):

$$\begin{aligned} \eta_i &= f_1(z_{i1}) + \ldots + f_K(z_{iK}) + \mathbf{x}_i'\boldsymbol{\beta} \\ \mu_i &= h(\eta_i) \end{aligned} \tag{2.12}$$

In semi-parametric Generalized STAR models, $f_j(\cdot)$ can have any of the structural forms described in Chapter 2.2. Nevertheless, the model assumptions for Generalized Linear Models including the exponential family requirement for the response variable still hold (Fahrmeir et al., 2013, chap. 9.5).

2.4 Structured Additive Distributional Regression

Generalized STAR models provide a framework to flexibly estimate the expected value of a previously specified distributional parameter. However, in many cases not only the first moment but also higher-order moments are of special interest. In modeling income, for example, not only the expected income but also the shape of the overall distribution is important. A common measure for income inequality is the Gini coefficient, which can be calculated using the cumulative distribution function (cdf) requiring estimates for all distributional parameters (Lerman and Yitzhaki, 1984).

2.4.1 GAMLSS

First modeling approaches which go beyond the mean of a distribution were suggested by Nelder and Pregibon (1987) using parametric functions of explanatory covariates related to the dispersion parameter ϕ of an exponential family distribution. Building upon this approach, Generalized Additive Models for Location, Scale and Shape (GAMLSS) were introduced by Rigby and Stasinopoulos (2001). Gamlss combine the flexibility of being able to model multiple distributions with parametric or non-parametric explanatory effects and extend them for multiple response distribution parameters such that not only the location but also the scale and shape of a distribution can be modeled simultaneously. Furthermore, GAMLSS relax the assumption of the dependent variable having to follow an exponential family distribution, which significantly increases the number of response modeling possibilities.

Assuming a dependent variable from a distribution with parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_L)'$

and observations $y_1, \ldots, y_n$, given covariates $\mathbf{z}_1, \ldots, \mathbf{z}_K$ and $\mathbf{x}_1, \ldots, \mathbf{x}_Q$, a GAMLSS can be described with the following model specification:

$$g_l(\boldsymbol{\theta}_l) = \boldsymbol{\eta}_l = \mathbf{X}_l\boldsymbol{\beta}_l + \sum_{j=1}^{K_l} \mathbf{Z}_{jl}\boldsymbol{\gamma}_{jl} \tag{2.13}$$

In Equation (2.13), $g_l(\cdot)$ represents a known monotonic link function, which can be different for each parameter. $\mathbf{X}_l$ depicts the subset of variables $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_Q)'$ used to model parameter θ_l, while $\mathbf{Z}_{jl}\boldsymbol{\gamma}_{jl}$ serves as the Basis function matrix for a non-parametric effect of covariate z_j on parameter θ_l, taken from a subset of variables $z_1, \ldots, z_K$ variables. The specific subset of covariates z with non-parametric effects on parameter θ_l has a length of K_l variables (Stasinopoulos and Rigby, 2007).

As shown above, GAMLSS can utilize different combinations of parametric and non-parametric effects to model each distributional parameter. Equation 2.13 displays a case in which every parameter is modeled using a non-empty subset of variables x and z. However, some parameters can also be set to a constant and not be dependent on covariates. For example, when assuming the Gaussian distribution for the dependent variable and connecting μ to parametric effects x_q using the identity link function ($g(\mu) = \mu$) and the variance parameter σ^2 to a constant, we arrive at a linear model specification (Stasinopoulos and Rigby, 2007).

2.4.2 BAMLSS

As mentioned in the introduction of this thesis, the modeled parameters do not always directly equate to the moments (location, scale and shape) of a distribution but rather a combination of them. For this reason, approaches to simultaneously model the parameters of a distribution are often referred to as distributional regression, which includes GAMLSS. However, as seen in (2.13), GAMLSS in its normal form only incorporate main effect modeling. To further integrate structured additive terms, such as spatial effects, random effects and interaction terms (Brezger and Lang, 2006), distributional regression is extended to Structured Additive Distributional Regression (Klein et al., 2015).

In 2013, Klein et al. introduced Bayesian Additive Distributional Regression, which is a model type extending GAMLSS to include structured additive pre-

dictors for modeling parameters of a specified distribution. It represents a fully Bayesian approach, in which coefficients are obtained by drawing samples from the approximate posterior effect distribution using Markov Chain Monte Carlo (MCMC) simulations.

An implementation of Bayesian Additive Distributional Regression, called Bayesian Additive Models for Location, Scale and Shape (BAMLSS) was since created by Umlauf et al. (2017). As the authors pointed out, the name bears resemblance to GAMLSS, because of many similarities in its modeling approach. However, extensions of BAMLSS over GAMLSS are manifold. First, parallel to the proposed framework of Klein et al. (2013), MCMC simulations are utilized for estimation of coefficients. This is done in contrast to GAMLSS, where predictor coefficient estimates are retrieved via several forms of penalized likelihood maximization. Advantages of using MCMC simulations over likelihood-based approaches include the sample-based inference, which yields more reliable confidence intervals than the intervals of GAMLSS estimates based on asymptotic properties. Second, BAMLSS offer more flexibility of specifying covariate effects with the support of structured additive predictors, like spatial effects or two-dimensional splines. Third, BAMLSS also support multivariate response distributions, which enhances GAMLSS' univariate response framework. Furthermore, the implementation of BAMLSS is designed in a way that allows for the usage of external estimation algorithms and software packages like JAGS or BayesX.

The model specification of BAMLSS is similar to the GAMLSS class. The parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_L)'$ of a parametric distribution with observations $\mathbf{y} = (y_1, \ldots, y_n)'$ are linked to structured additive predictors using monotonic and twice-differentiable link functions $g_l(\theta_l)$ [1]. Based on covariates $\mathbf{x}_1, \ldots, \mathbf{x}_Q$, the following model equation can be obtained:

$$g_l(\boldsymbol{\theta}_l) = f_{1l}(\mathbf{X}_{1l}; \boldsymbol{\beta}_{1l}) + \ldots + f_{Q_l l}(\mathbf{X}_{Q_l l}; \boldsymbol{\beta}_{Q_l l}) \tag{2.14}$$

Here, $f_{jl}(\cdot)$ represent unspecified functions with regression coefficients $\boldsymbol{\beta}_{jl}$ that can attain any structured additive predictor forms, including non-parametric effects. The corresponding model matrices used for each parameter are $\mathbf{X}_{1l}, \ldots, \mathbf{X}_{1Q_l}$, which are constructed depending on the type of covariate(s) and the prior as-

[1] note that Umlauf, Klein, and Zeileis (2017) use $h_l(\cdot)$. It was changed to $g_l(\cdot)$ in this thesis for continuity reasons.

sumptions about $f_{jl}(\cdot)$. It is also possible to describe the effects in vector form:

$$\boldsymbol{f}_{jl} = \begin{bmatrix} f_{jl}(\mathbf{x}_1; \boldsymbol{\beta}_{jl}) \\ \vdots \\ f_{jl}(\mathbf{x}_n; \boldsymbol{\beta}_{jl}) \end{bmatrix} = f_{jl}(\mathbf{X}_{jl}; \boldsymbol{\beta}_{jl})$$

with $\mathbf{X}_{jl}$ $(n \times m_{jl})$ specifying the design matrix for effect $f_{jl}(\cdot)$ so that they integrate themselves into the following model equation

$$g_l(\boldsymbol{\theta}_l) = \boldsymbol{\eta}_l = \boldsymbol{f}_{1k} + \ldots + \boldsymbol{f}_{J_l l} \tag{2.15}$$

where $\boldsymbol{f}_{jl}$ represents the vector of function evaluations for the jth effect on parameter θ_l. Similar to Chapters 2.1 and 2.2, effects in BAMLSS can also be derived through a Basis function approach, in which case they can be written as $\boldsymbol{f}_{jl} = \mathbf{X}_{jl}\boldsymbol{\beta}_{jl}$ (Umlauf et al., 2017).

As mentioned earlier in this chapter, BAMLSS offer very flexible ways of specifying covariate effects. Breaking through the framework of Basis function approaches, BAMLSS also allow covariate functions $f_{jl}(\cdot)$ which are non-linear in its parameters $\boldsymbol{\beta}_{jl}$. An example of this is the Gompertz growth curve

$$\boldsymbol{f}_{jl} = \beta_1 \cdot \exp(-\exp(\beta_2 + \mathbf{X}_{jl}\beta_3))$$

with non-linear parameters $\boldsymbol{\beta}_{jl}$(Umlauf et al., 2017).

2.5 Estimation

In this section, estimation techniques for obtaining the coefficients for regression models introduced in Section 2 will be discussed.

2.5.1 STAR and Generalized STAR models

Structured Additive Regression Models (STAR) including simple Additive Models (AM) can be estimated using backfitting, an algorithm first coined by Friedman and Stuetzle (1981). The intention of backfitting is to compute estimates for the model terms by the iterative smoothing of partial residuals. Originating

from the STAR model equation

$$\mathbf{y} = \sum_{j=1}^{K} \boldsymbol{f}_j + \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

one can describe $\boldsymbol{f}_j$ as

$$\boldsymbol{f}_j \approx \mathbf{y} - \sum_{l \neq j} \boldsymbol{f}_l - \mathbf{X}\boldsymbol{\beta}$$

Thus, estimates $\hat{\boldsymbol{f}}_j$, $\hat{\boldsymbol{\beta}}$ can be obtained by running the following steps:

1. Intitialize by setting $\hat{\boldsymbol{f}}_1 = \ldots = \hat{\boldsymbol{f}}_K = \hat{\boldsymbol{\beta}} = 0$.
2. Compute estimates $\hat{\boldsymbol{f}}_j$ by

$$\hat{\boldsymbol{f}}_j = \mathbf{S}_j(\mathbf{y} - \sum_{i \neq j} \hat{\boldsymbol{f}}_l - \mathbf{X}\hat{\boldsymbol{\beta}})$$

3. Compute estimates $\hat{\boldsymbol{\beta}}$ through

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'(\mathbf{y} - \sum_{j=1}^{K} \hat{\boldsymbol{f}}_j)$$

4. Repeat Steps 2-3 until the iteration deviations between estimates $\hat{\boldsymbol{f}}_j$ are sufficiently small.

Note that $\mathbf{S}_j$ can be any smoothing function. In the case of penalized smooth splines, this could be the penalized least squares estimate $\mathbf{S}_j = (\mathbf{Z}'\mathbf{Z} + \lambda\mathbf{K})^{-1}\mathbf{Z}'$ with smoothing parameter λ, design matrix $\mathbf{Z}$ and penalty matrix $\mathbf{K}$. The estimated smoothing parameter λ can be computed by optimizing a model choice criterion, for example GCV or AIC (Fahrmeir et al., 2013).

Generalized STAR models, which use a link function to relate structured additive predictors to expected values of exponential family distributions, are estimated based on a maximum-likelihood approach. To obtain coefficient estimates, the penalized log-likelihood criterion

$$l_{pen}(\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_K, \boldsymbol{\beta}) = l(\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_K, \boldsymbol{\beta}) - \frac{1}{2}\sum_{j=1}^{K} \lambda_j \boldsymbol{\gamma}_j' \mathbf{K}_j \boldsymbol{\gamma}_j \tag{2.16}$$

is maximized in regard to $\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_K, \boldsymbol{\beta}$. The first part of the criterion, $l(\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_K, \boldsymbol{\beta})$

depicts the log-likelihood known from Generalized Linear Models but with structural additive effects.

As suggested by Fahrmeir et al. (2013), the penalized log-likelihood criterion of (2.16) can in some cases be directly maximized. For example, an R implementation for Generalized Additive Models (which can be seen as a sub-variant of STAR models), `mgcv` (Wood, 2011), uses penalized iteratively reweighted least squares to directly obtain coefficient estimates. Optimization of the penalized log-likelihood can also be combined with backfitting, as it is done in the R package `gam`, written by Hastie (2017).

2.5.2 GAMLSS

Generalized Additive Models for Location, Scale and Shape (GAMLSS) also rely on the penalized likelihood in (2.16) to obtain coefficient estimates. For maximization, Rigby and Stasinopoulos (2005) present two algorithms: the "RS" algorithm, a generalization of the algorithm suggested by Rigby and Stasinopoulos (1996) for fitting mean and dispersion additive models, and "CG", which represents a generalization of the algorithm coined by Cole and Green (1992) relying on first and second (cross) derivatives of the likelihood function with regard to the distributional parameters.

RS, abbreviated for Rigby and Stasinopoulos, uses iteratively Iteratively Reweighted Least Squares (IRLS) in combination with a modified backfitting algorithm to arrive at coefficient estimates. To illustrate the algorithm system, it is broken up into inner and outer iterations, with each inner iteration depicting the fitting of one distributional parameter θ_l. Here, a working variable $\mathbf{z}_l$ consisting of all used predictors, the first derivative of the likelihood (score function) and "iterative weights" $\mathbf{w}_l$, determined with a local scoring algorithm, is calculated. Then, the working variable is fit to the explanatory variables using backfitted weighted least squares and penalized weighted least squares for parametric and non-parametric coefficients, respectively. The inner iteration is repeated until the inner global deviance has converged. This procedure is done for every θ_l, after which one outer iteration is finished. The outer iterations are further repeated, until the outer global deviance has also converged (Stasinopoulos et al., 2017, Chap. 3).

Note that while the RS algorithm uses score functions to update the working

variable, the cross derivatives of the log likelihood functions with respect to the distributional parameters $\boldsymbol{\theta}_l$ are not needed. The Cole and Green (CG) algorithm, however, uses cross derivatives to obtain new weights for the iteratively reweighted least squares estimation. Other contrasts between CG and RS include the iteration system of CG, where new weights $\mathbf{w}_l$ and the working variable $\mathbf{z}_l$ for iteratively reweighted least squares are not updated for fitting of every parameter but represents the outer iteration. In the inner iterations, distributional parameters are fitted after each other relying on given weights depending on the current working variable. Here, backfitting is again used to compute current estimates of regression coefficients. After the inner and outer iterations have converged, model estimation is finished (Stasinopoulos et al., 2017, Chap. 3). As stated by Stasinopoulos et al. (2017), the RS algorithm is preferred in most cases since it is usually faster and converges more consistently than CG. In cases where the modeled distribution posesses highly correlated parameters, however, RS can be slower and suffer from premature convergence.

2.5.3 BAMLSS

As the name already suggests, the framework for estimating Bayesian Additive Models for Location, Scale and Shape (BAMLSS) is based upon Bayesian principles. In this case, model coefficient estimates are assumed to follow a prior distribution, rather than be fixed and unknown quantities. Combined with the likelihood function of the data given parameters $\boldsymbol{\beta}$ and explanatory variables $\mathbf{X}$, this yields the posterior density, which is typically logarithmized to form the log-posterior density

$$\log \pi(\boldsymbol{\beta}, \boldsymbol{\tau}|\mathbf{y}, \mathbf{X}, \boldsymbol{\alpha}) \propto \underbrace{l(\boldsymbol{\beta}|\mathbf{y}, \mathbf{X})}_{\text{log-likelihood}} + \underbrace{\sum_{l=1}^{L} \sum_{j=1}^{J_l} \log p_{jl}(\boldsymbol{\beta}_{jl}|\boldsymbol{\tau}_{jl}, \boldsymbol{\alpha}_{jl})}_{\text{prior distributions}} \tag{2.17}$$

where $\boldsymbol{\beta}_{jl}$ depict regression coefficients for the jth predictor on distributional parameter θ_l and therefore serves as a subset of $\boldsymbol{\beta}$, and $\mathbf{y}, \mathbf{X}$ are response and design matrices of explanatory variables respectively. Vectors $\boldsymbol{\tau}$ and $\boldsymbol{\alpha}$ are both parameters of the prior distributions $p_{jl}(\cdot)$, whereas $\boldsymbol{\tau}$ represents hyper-parameters which can have their own distribution $d_{\boldsymbol{\tau}_{jl}}(\cdot)$, while $\boldsymbol{\alpha}$ is fixed and not varying. Thus, the prior distributions in the BAMLSS framework can be further dissected

into

$$p_{jk}(\boldsymbol{\beta}_{jl}|\boldsymbol{\tau}_{jl}, \boldsymbol{\alpha}_{jl}) \propto d_{\boldsymbol{\beta}_{jl}}(\boldsymbol{\beta}_{jl}|\boldsymbol{\tau}_{jl}, \boldsymbol{\alpha}_{\boldsymbol{\beta}_{jl}}) \cdot d_{\boldsymbol{\tau}_{jl}}(\boldsymbol{\tau}_{jl}|\boldsymbol{\alpha}_{\boldsymbol{\tau}_{jl}}) \tag{2.18}$$

As the authors of BAMLSS state, $\boldsymbol{\tau}_{jl}$ are usually variances, which can be used to control the degree of smoothness of $f_{jl}(\cdot)$ or the amount of correlation between observations. For example, when using a normal distribution prior for regression splines, $\boldsymbol{\tau}_{jl}$ represent inverse smoothing parameters $\boldsymbol{\lambda}$ from a penalized likelihood framework (Umlauf et al., 2017).

While the prior distributions in (2.18) can be chosen arbitrarily by the user, Umlauf, Klein, and Zeileis (2017) give some suggestions and default distributions for priors of commonly-used model terms. Linear effects, for example, are typically modeled using very uninformative priors with a high variance. In this case, the hyper-parameters $\boldsymbol{\tau}_{jl}$ represent fixed values. Priors of non-linear effects usually take the form of multivariate normal distributions for $\boldsymbol{\beta}_{jl}$ with non-fixed parameter $\boldsymbol{\tau}_{jl}$, which controls the amount of smoothness of $f_{jl}(\cdot)$. Specifying the prior density of $\boldsymbol{\tau}_{jl}$ for non-linear effects is typically done using inverse gamma distributions or the half-Cauchy distribution which is preferred when it can be assumed that $\boldsymbol{\tau}_{jl}$ reaches values close to zero. Incorporating geographical effects, which might have additional covariate information available at each of the geographical levels, can be done using compound priors, which contain a vector of Gaussian random effects and the nested predictors.

The BAMLSS R implementation, `bamlss`, provides a highly customizable Bayesian estimation framework, which by default revolves around two main functions: `bamlss::bfit()` and `bamlss::GMCMC()`. The first function, `bfit()`, utilizes an optimizing function which seeks to find the mode of the posterior distribution with respect to $\boldsymbol{\beta}$ and $\boldsymbol{\tau}$. Then, those values are used as starting numbers for Markov Chain Monte Carlo (MCMC) simulations (function `GMCMC()`), which samples from the posterior distribution of the coefficients and allows the calculation of its posterior median or mean. Both functions can be swapped by the user with optimizer and sampler functions that more closely resemble the subjective preference. The reason for this full Bayesian two-step procedure over direct MCMC sampling is the high difficulty of finding good MCMC starting values. Directly starting MCMC using insufficient starting values would often yield a high rejection frequency, a long burn-in phase or divergence of the sampler. This is especially true in cases where highly complex hyper-parameters are used, for example bivariate smooth splines (Umlauf et al., 2017).

Optimizing the log-posterior density to obtain its mode with respect to the regression coefficients $\boldsymbol{\beta}$ and smoothing variances $\boldsymbol{\tau}$ in ł`bamlss` is completed using a nested iterative weighted least squares approach in combination with backfitting. In the inner iterations, optimal regression coefficients are obtained by fitting the first effect $f_{1l}(\cdot)$ on the current working variable of parameter θ_l with weighted least squares, then using the partial residuals from its subtraction with the current predicted working variable to fit all other effects. This is applied to every parameter θ_l until the inner iteration converges. In the outer iteration, the current working variable $\mathbf{z}_l$ and new weights $\mathbf{W}_{ll}$ are updated and re-used in backfitting until it also converges (Umlauf et al., 2017).

While backfitting the estimated effects f_{jl}, the variance parameter $\boldsymbol{\tau}_{jl}$ is also optimized at each step. After specifying a certain search interval the best solution is found based on a one-dimensional search given a certain goodness-of-fit criterion, e.g. BIC or AIC. Umlauf, Klein, and Zeileis (2017) state that while this does not assure that the found $\boldsymbol{\tau}_{jl}$ are also global minima, it will at least deliver good starting values for a later MCMC sampling (Umlauf et al., 2017).

Note that the fitting of $f_{jl}(\cdot)$ inside the backfitting algorithm represents a single Newton-Raphson step since weights $\mathbf{W}_{ll}$ are constructed using only second but no cross derivatives of the likelihood with respect to the transformed parameters $\boldsymbol{\eta}_l$ by default. Therefore, the `bamlss` log-posterior maximization framework bears many similarities to the RS algorithm proposed by Rigby and Stasinopoulos (2005) for estimation of GAMLSS (Umlauf et al., 2017). Furthermore, the nesting procedure of the backfitted iterative weighted least squares algorithm can be changed. For example, the procedure of backfitting effects for every parameter can be repeated until it converges within the current parameter before starting the backfitting procedure for the next θ_l. Also, it is possible to limit the number of outer iterations to one, such that the weights and working variable are not further updated (Umlauf et al., 2017).

After optimal parameters $\boldsymbol{\beta}$ and $\boldsymbol{\tau}$ for maximizing the log-posterior are found, they are used as starting values for the MCMC algorithm. Using MCMC simulations to obtain samples from the posterior distribution to approximate its mean is essential since the expectation of the posterior

$$E(\boldsymbol{\beta}, \boldsymbol{\tau} \,|\mathbf{y}, \mathbf{X}, \boldsymbol{\alpha}) = \int \left(\begin{smallmatrix}\boldsymbol{\beta}\\ \boldsymbol{\tau}\end{smallmatrix}\right) \pi(\boldsymbol{\beta}, \boldsymbol{\tau}|\mathbf{y}, \mathbf{X}, \boldsymbol{\alpha})\, d\left(\begin{smallmatrix}\boldsymbol{\beta}\\ \boldsymbol{\tau}\end{smallmatrix}\right)$$

can rarely be solved analytically. Umlauf et al. (2017) give some suggestions of MCMC algorithms from which samples for $\boldsymbol{\beta}$ can be obtained. One of those is defined by the "Random-Walk Metropolis" algorithm, which is preferred in most cases due to its generality and ease of implementation. In its core, the sampler draws a candidate $\boldsymbol{\beta}_{jl}^{*}$ from a symmetric "jumping" distribution $q(\boldsymbol{\beta}_{jl}^{*}|\boldsymbol{\beta}_{jl}^{(t)})$, which is then accepted as a new state of the Markov Chain with probability

$$\alpha(\boldsymbol{\beta}_{jl}^{*}|\boldsymbol{\beta}_{jl}^{(t)}) = \min[\frac{\pi(\boldsymbol{\beta}_{jl}^{*}|\ \cdot)}{\pi(\boldsymbol{\beta}_{jl}^{(t)}|\cdot)}, 1]$$

where the log-posterior density is evaluated at both the current and proposed value. In common cases, the jumping distribution is denoted by a normal distribution with expectation $\boldsymbol{\beta}_{jl}^{(t)}$ and covariance matrix Σ_{jl}. Nevertheless, in complex model specifications this sampling framework is not efficient and does not lead to i.i.d. behavior of the Markov chain. Therefore, Umlauf et al. (2017) suggest to set the covariance matrix to the local curvature information $\Sigma_{jl} = -(\partial^2\pi(\boldsymbol{\theta}|\mathbf{y},\mathbf{X})/\partial\boldsymbol{\beta}_{jl}\boldsymbol{\beta}_{jl}')^{-1}$ or its expectation, evaluated at the posterior mode estimates $\hat{\boldsymbol{\beta}}_{jl}$.

However, having a fixed covariance matrix during MCMC simulation might still yield undesirable chain behavior, especially when moving into posterior distribution regions with a low density. As a solution, the authors of BAMLSS suggest the use of a "Derivative-based Metropolis-Hastings" sampler, where the jumping distribution is again a normal distribution but its expectation and covariance matrix are iteratively recalculated using iterative weighted least squares, where at each step a new working variable $\mathbf{z}_l$ and weights $\mathbf{W}_{ll}$ depending on current evaluations of the log-likelihood function $l(\boldsymbol{\beta}|\mathbf{y},\mathbf{X})$ are defined (Umlauf et al., 2017).

By default, the R implementation of BAMLSS obtains samples using the "Derivative-based Metropolis-Hastings" algorithm. In cases where the smoothing variances $\boldsymbol{\tau}$ are very complex (e.g. in multivariate smooth splines), `bamlss` uses "Slice" sampling (Umlauf et al., 2017).

3 bamlss.vis

The previous Sections 2.1 to 2.5 gave a description of Bayesian Additive Models for Location, Scale and Shape (BAMLSS) and the underlying sub-models on which they are based. This section will introduce a framework to interactively visualize covariate effects and distributional predictions of fitted `bamlss` models and feature its implementation as an R package. Due to its visual component, the tool will be called bamlss.vis.

The proceeding sub-chapter 3.1 will explain the motivation for bamlss.vis. Afterwards, a small case-study based on wages of male workers will be presented in Section 3.2. The resulting fitted `bamlss` object is then used to feature most of bamlss.vis' abilities in Section 3.3. In Section 3.4, additional bamlss.vis functions which go beyond the model created in the case-study are presented.

3.1 Motivation

As discussed in previous sections, distributional regression is concerned with modeling the parameters of a known parametric distribution. After estimation of the model, the user obtains coefficients which measure the influence of an explanatory variable on η_l, representing the transformed parameter θ_l. Nonetheless, in most cases the user is more interested in the inference on the expected moments of a distribution. Yet, in many cases the moments do not directly match the expected parameters.

To further illustrate the problem, consider the censored normal distribution y^* constructed from a normally distributed variable, $y \sim N(\mu, \sigma^2)$. Assuming a left-censored y^* with cut-off point $a = 0$, its probability density function can be obtained by:

$$f(y^* = x) = \begin{cases} f(y = x) & x > 0 \\ F(y = \frac{-\mu}{\sigma}) & x \leq 0 \end{cases}$$

where $f(y)$ and $F(y)$ are the probability density function (pdf) and the probability density function (pdf) of normally distributed variable y, respectively. It is visible in the above equation that the censored normal distribution is both discrete and continuous. While y^* shares the density with y above the cut-off point, the full remaining density in the censored normal distribution is assigned

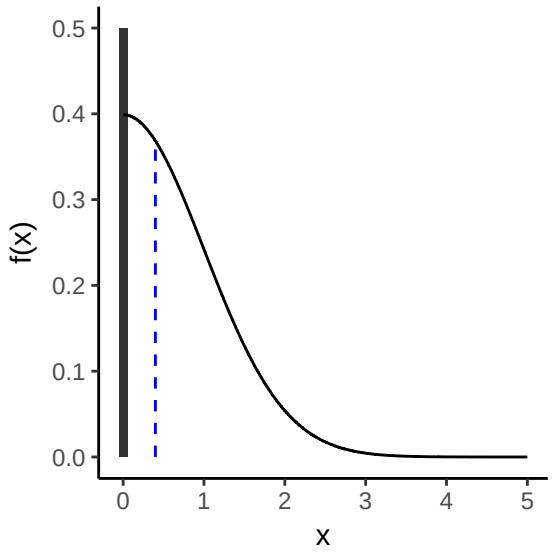

Figure 1: Probability Density Function of a left-censored normal distribution with the expected value drawn as a blue line.

to the cut-off point a. Figure 1 shows a sample left-censored normal distribution y^* created from $y \sim N(0,1)$ with $a = 0$ (Greene, 2012).

As visible in Figure 1, the moments of the standard normal distribution do not carry over to the censored normal distribution. In fact, while $E(y) = 0$, the expected value of y^* is $E(y^*) \approx 0.399$. To be exact, the censored normal distributions first two moments with cut-off $a = 0$ can be calculated as follows:

$$
\begin{aligned}
E(y^*) &= (1-\alpha) \cdot (\mu + \sigma\beta) \quad \text{and} \\
Var(y^*) &= \sigma^2(1-\alpha) \cdot [(1-\gamma) + (\frac{-\mu}{\sigma} - \beta)^2 \cdot \alpha]
\end{aligned}
$$

$$
\begin{aligned}
\text{while:} \quad \alpha &= \Phi(\frac{-\mu}{\sigma}) \\
\beta &= \frac{\phi(\frac{\mu}{\sigma})}{1-\alpha} \\
\gamma &= \beta^2 - \beta \cdot (\frac{-\mu}{\sigma})
\end{aligned}
\tag{3.1}
$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ are the pdf and cdf of the standard normal distribution and μ and σ^2 are the parameters of y, respectively (Greene, 2012). Equation (3.1) shows that both the expected value and the variance of y^* are computed by a combination of the parameters of the original variable y, μ and $\sigma^2$and are not equal. Thus, an explanatory variable that has a positive effect on μ has both an impact on $E(y^*)$ and $Var(y^*)$. Therefore, coefficients for measuring covariate

influences on those parameters are not directly translatable to the moments of the modeled distribution and might even have critically different estimates.

Furthermore, even in cases where the desired moments directly equate the modeled parameters (e.g. in Gaussian or Poisson-distributed responses), different link functions for their transformation $g_l(\theta_l)$ and possibly highly complex non-parametric effects of explanatory variables can lead to coefficient estimates that are hard to interpret. In this case, a visual comparison of predicted distributions is needed.

To tackle both of the aforementioned interpreting problems with fitted `bamlss` models, bamlss.vis possesses two main objectives:

1. Visually compare the predicted distributions (pdf or cdf) based on interactively selected covariates
2. View the changes of distribution moments over the whole range of a selected variable, based on chosen explanatory covariates.

Using bamlss.vis, one can then observe the influence of a covariate on the distribution by its cdf or pdf (1) and its moments (2).

3.2 Case-Study

Presenting bamlss.vis' abilities is best achieved using a `bamlss` fitted with a dataset of real observations. The objective for a suitable dataset is that its response variable and explanatory variables are easy to understand for users without a specific scientific background. The chosen dataset, "Wage" from the ISLR R package (James et al., 2017), fulfills these requirements. "Wage", collected by the United States Census Bureau (2011), includes 3000 male individuals living in the Mid-Atlantic region of the United States of America with records of the following variables:

- **wage**: Worker's raw wage (in $1000)
- **age**: Age of worker
- **year**: Year that wage information was recorded
- **race**: A factor with levels `1. White, 2. Black, 3. Asian` and `4. Other`

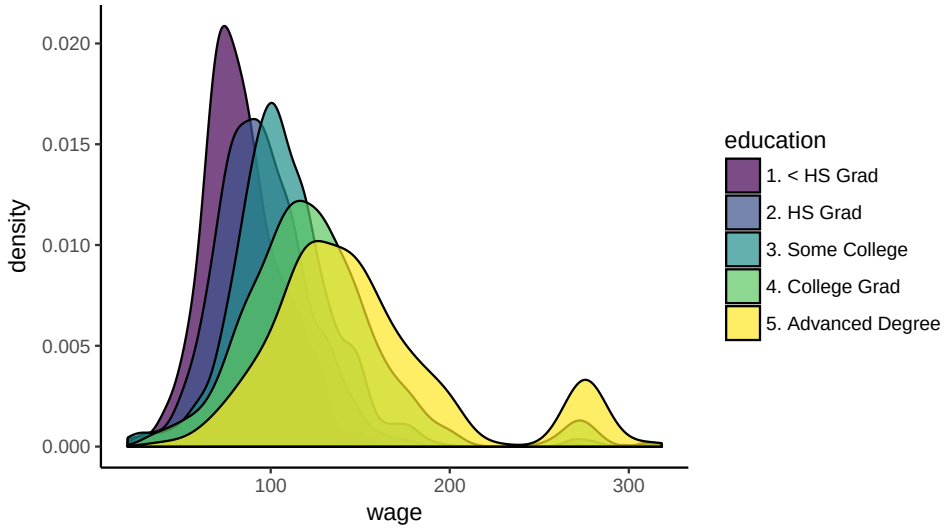

Figure 2: Gaussian kernel density estimates for wages split up by education level.

- **education**: A factor with levels `1. < HS Grad`, `2. HS Grad`, `3. Some College`, `4. College Grad` and `5. Advanced Degree`
- **health**: A factor with levels `1. <= Good` and `2. >= Very Good`, indicating the health level of worker

The variable, whose distribution is of interest and shall be modeled is the male worker's wage. While doing first analyses, it is clear that the wage is highly dependent on the given variables. Figure 2 shows kernel density estimates (Gaussian) for the estimated wage distribution depending on education level.

As visible in Figure 2, the kernel density estimates are critically different for each education level. In general, we can observe that a higher education level is linked to a higher expected income but also to an increased variance. Therefore, both location and shape should be modeled when fitting the `bamlss`. As income cannot be smaller than zero but does otherwise not have upper limits, the censored normal distribution with cut-off $a = 0$ is chosen as the response family. After some data preparation, model fitting is completed by running the following R code: (Umlauf et al., 2017):

```
cnorm_model <- bamlss(
  list(wage ~ s(age) + race + year + education + health,
       sigma ~ s(age) + race + year + education + health),
  data = wage_sub,
  family = cnorm_bamlss()
```

```
)
```

Code-Chunk 1: R code for fitting the `bamlss` based on Wage dataset

As visible in Code-Chunk 1, both μ and σ are modeled such that they relate to explanatory variables additively. Both parameters are connected to parametric effects `race`, `year`, `education` and `health`. The influence of `age` is specified with a thin-plate smooth spline.

3.3 Application Structure & Guide

As previously mentioned, bamlss.vis is implemented in the form of an R extension. For building and maintaining the package, GitHub is used. This allows users to easily install the package with the following R commands:

```
if (!require(devtools))
  install.packages("devtools")
devtools::install_github("Stan125/bamlss.vis")
```

Furthermore, bamlss.vis is strongly based on the Shiny framework (Chang et al., 2017), which is an R package designed to create interactive visualizations with HTML code and R functions. In the words of the author, Shiny combines "the computational power of R with the interactivity of the modern web" (RStudio, Inc., 2017).

In its core, a Shiny application is built using R functions and can therefore be called similarly. In the case of bamlss.vis, there are two ways one can start the application. First, the user can run the code `bamlss.vis::vis()`. Second, bamlss.vis can also be called using the open source General User Interface RStudio. When opened, one can click on the "Add-Ins" button and then select "BAMLSS Model Visualizer" if bamlss.vis is installed (Figure A1 in the Appendix). This will also trigger the command `bamlss.vis::vis()`.

After executing the R code, a new browser window with the started application is opened up. Figure 3 shows the layout of the application, which is then displayed in the user's browser. As visible in Figure 3, the layout of bamlss.vis is divided into two segments, which have their own tabs the user can click on. In each segment, one of those tabs is always displayed. The left segment, with tabs "Overview", "Scenarios" and "Scenario Data", is concerned with model-related settings. The

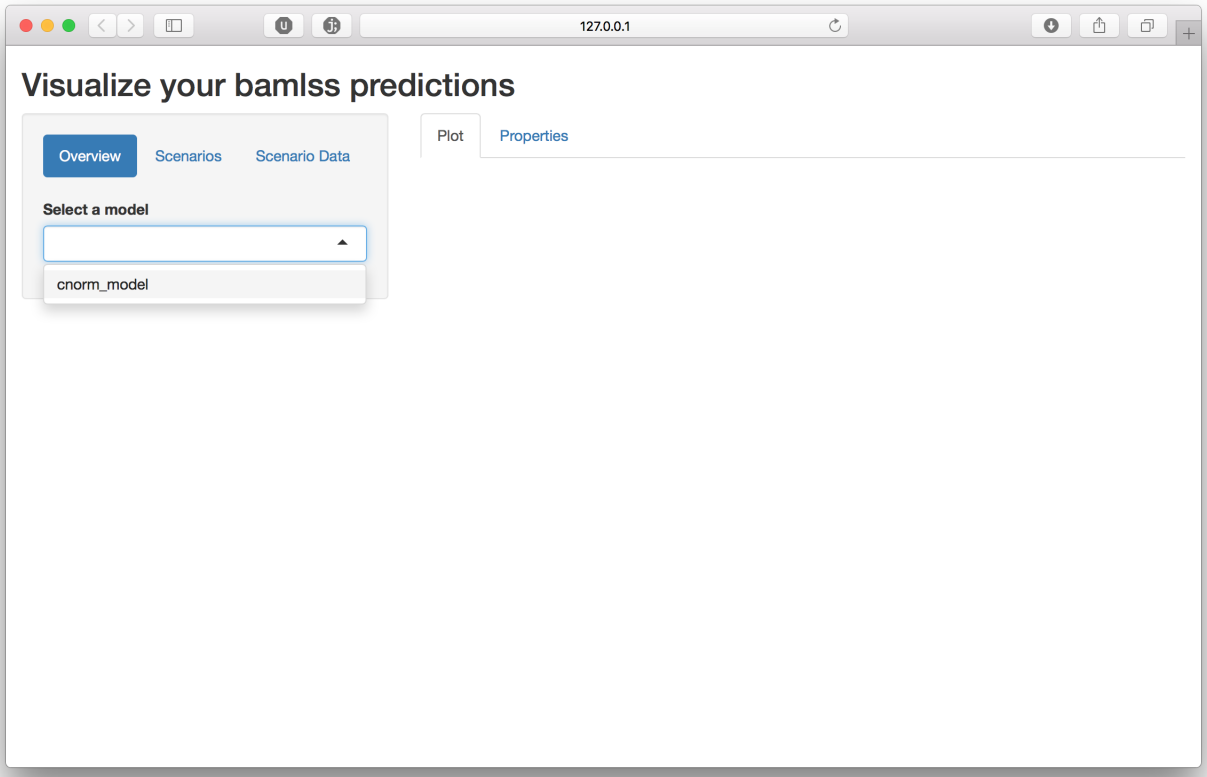

Figure 3: Layout of bamlss.vis after starting the application.

right segment, with tabs "Plot" and "Properties" is used to display graphs and properties in reaction to user inputs on the left segment.

3.3.1 Overview Tab

The purpose of the overview tab is to display descriptive details about fitted `bamlss`. After bamlss.vis is started up, it only consists of a drop-down list where the user can select the model on which the further analysis is to be based. Entries in this list are created by an R function which searches the working directory of the current user for any object of the class `bamlss`. Figure 3 shows only one entry, `cnorm_model`, representing the model fitted with the code provided in Section 3.2.

After a model is selected, the overview tab expands to show an outline of the fitted `bamlss`. Specifically, as shown in Figure 4, the tab displays two parts, called "Model Family" and "Model Equations". "Model Family" shows the distributional family of the model, as well as the parameters which can be modeled including their link functions. In the case of `cnorm_model`, the family "cnorm" (for censored normal distribution) with parameters μ and σ and link functions

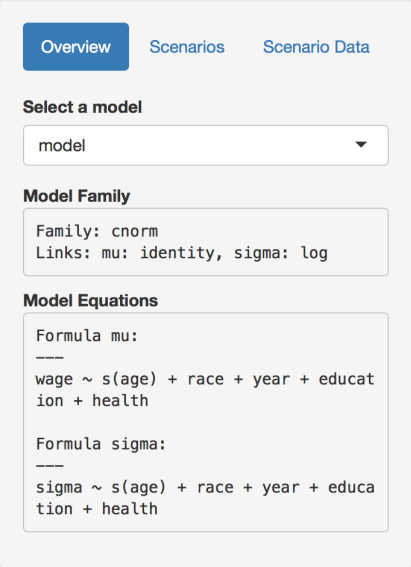

Figure 4: Expanded overview tab after model selection.

"identity" and "log" can be obtained. "Model Equations" displays the way covariate effects were specified. We can confirm that for `cnorm_model`, the effect of age on both μ and σ is specified using a smooth spline.

3.3.2 Scenarios Tab

In this tab, the user can interactively specify covariate values for each explanatory variable, thereby creating a "Scenario". After these combinations are "locked in" by clicking a button, the predicted distribution based on the previously selected model is then plotted. This can be repeated with different covariate values to compare the predictions for multiple covariate combinations.

As displayed in Figure 5, the top of the tab consists of two buttons, "Create Scenario" and "Clear Scenarios". Right below these buttons a box for including the intercept in predictions is portrayed. Further below, web widgets for each explanatory variable are visible. Bamlss.vis executes a check for the type of each explanatory variable and then constructs different web application elements depending on that information. Categorical covariates receive selector boxes (R function `shiny::selectInput()`) with the variables' possible categories, while for numeric variables slider modules are created (`shiny::numericInput()`), ranging from the variable's minimum to maximum value. The default value for numeric covariates is its arithmetic average.

To add a new scenario, one can select a value for each variable and then click on "Create Scenario". Then, covariate combinations can be changed to create a new

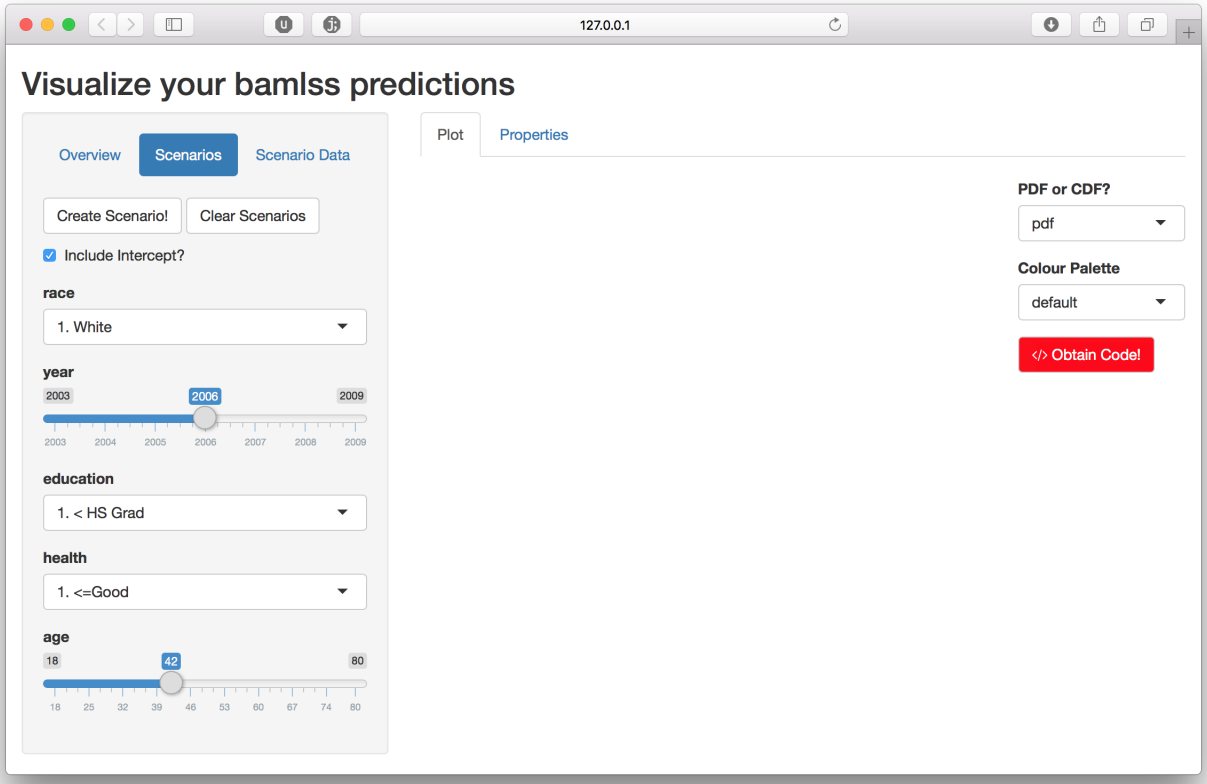

Figure 5: Scenarios tab of bamlss.vis.

scenario. This way, predictions will be computed for every specified combination. Deleting all previously specified combinations can be achieved by clicking "Delete Scenario".

Beneath the visual components of the "Scenarios" rests a small database, which stores each covariate combination the user has specified. This database, created with `shiny::reactiveValues()` forms the basis for all other tabs which analyze those covariate combinations. It has a "reactive" nature, such that all tabs depending on it are automatically updated when a database value changes.

Because critical differences in wage distributions depending on education were already observed in Figure 2 (p. 23), it will now be interesting to recreate this plot based on the fitted `cnorm_bamlss` by visually comparing the modeled distributions depending on `education`, while controlling for other variables. To achieve this, the following covariate values will be specified first:

- `race: 1. White`
- `year: 2006`
- `education: 1. < HS grad`

- `health: 2. >= Very Good`
- `age: 42` ($= \overline{age}$)

Then, the "Create Scenario" Button is pressed. This is done four more times, with each time seeing a rise in education level by one category. Thus, we can now view the according wage distributions for a 42-year-old white male with very good health across all levels of education (Figure 6).

3.3.3 Plot Tab

The "Plot" tab, which is located in the right-hand segment of bamlss.vis, reacts to user interaction in the Scenarios tab. Every time the "Create Scenarios" button is pressed, the tab is updated. Specifically, the data which the user inputs in the left tab is passed onto `bamlss.vis::preds()` (a customized version of `bamlss::predict.bamlss()`, which then computes predictions for the response distribution parameters by taking the arithmetic average of transformed MCMC samples from the posterior distribution. Afterwards, the predicted parameters are inserted into the probability density function for graphically visualizing the predicted distribution. This procedure is repeated for each "Scenario".

Figure 6 shows the plot output for five different scenarios based on the `Wage` dataset described in Section 3.3.2. As visible in the aforementioned figure, the predicted distributions have both a higher expected value and a higher variance as the education level rises, similar to the kernel density estimates in Figure 2.

Also noticeable in Figure 6 to the right side of the plot are three web elements for user interaction. The first element, found below the description "PDF or CDF?", provides the ability to switch between displaying the pdf (default value) or the cdf. Figure A2 in the Appendix shows Figure 6 after the "cdf" option was selected.

The second element gives the option to select a different color palette. Its default value is "default", which uses the built-in color palette provided in ggplot2 (Wickham, 2009). The selected color palette in Figure 6 is "viridis", which is a colorblind-friendly palette spanning over a high range of different colors (Garnier, 2017).

The third web element on the right side of the plot output, a red button with

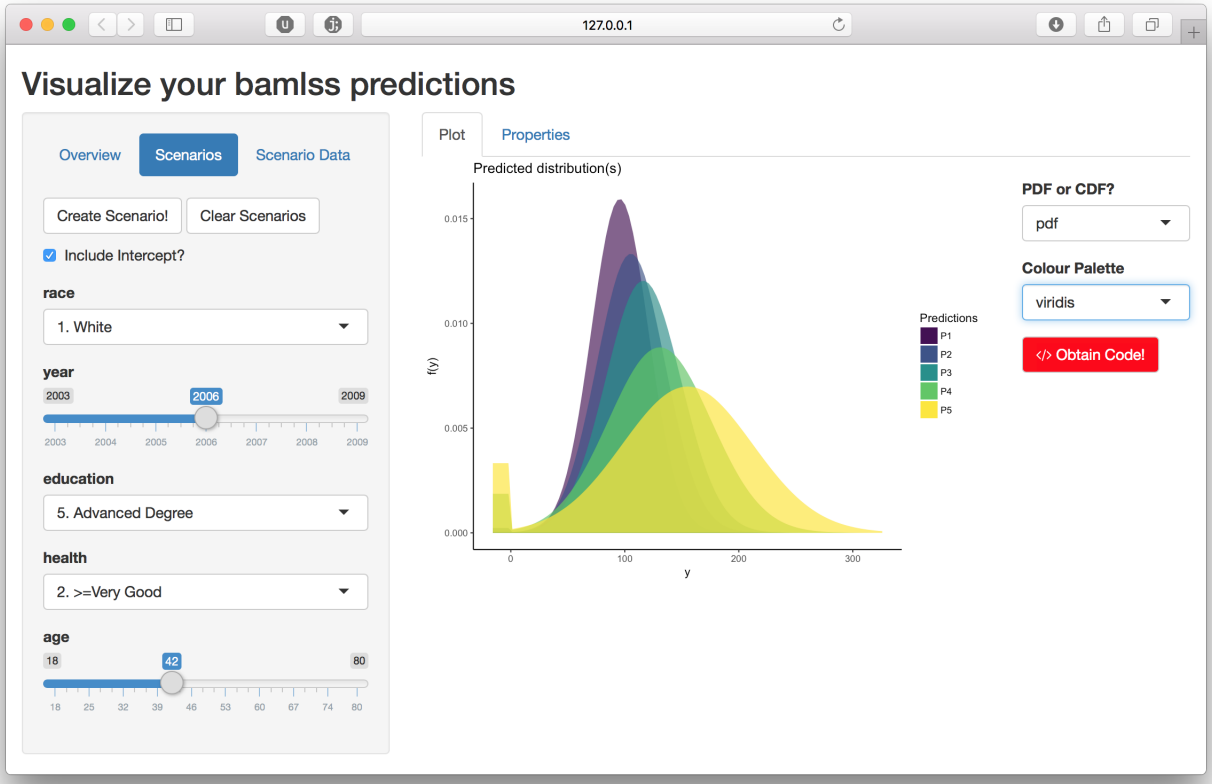

Figure 6: Plot tab output when specifying five different scenarios with different education levels.

the description "Obtain Code!", adds reproducibility to the plot. When clicked, a modal window pops up with R commands that, if executed in the main R console with the user's current working environment, directly recreates the graph currently being displayed in the "Plot" tab. Figure 7 shows the modal window which arises when pressing the "Obtain Code!" button in the interface of Figure 6. Below the description "Obtain your R code!" it is possible to see a text window with a dark background. In it are three syntax-highlighted R commands which can be copied and stored for future recreation. The first R command creates a `data.frame` object from the"Scenarios" tabs user input. The second one uses `bamlss.vis::preds()` in combination with the provided models name to compute predicted parameters for the data, which are then re-used in line three to plot the results graphically with `bamlss.vis::plot_dist()`.

Furthermore, bamlss.vis checks for specified plot options (type of distribution, color palette) and includes them as arguments in the last line. The formatting of code relies on the R package `formatR` (Xie, 2017). Clicking on the "Dismiss" button closes the modal window.

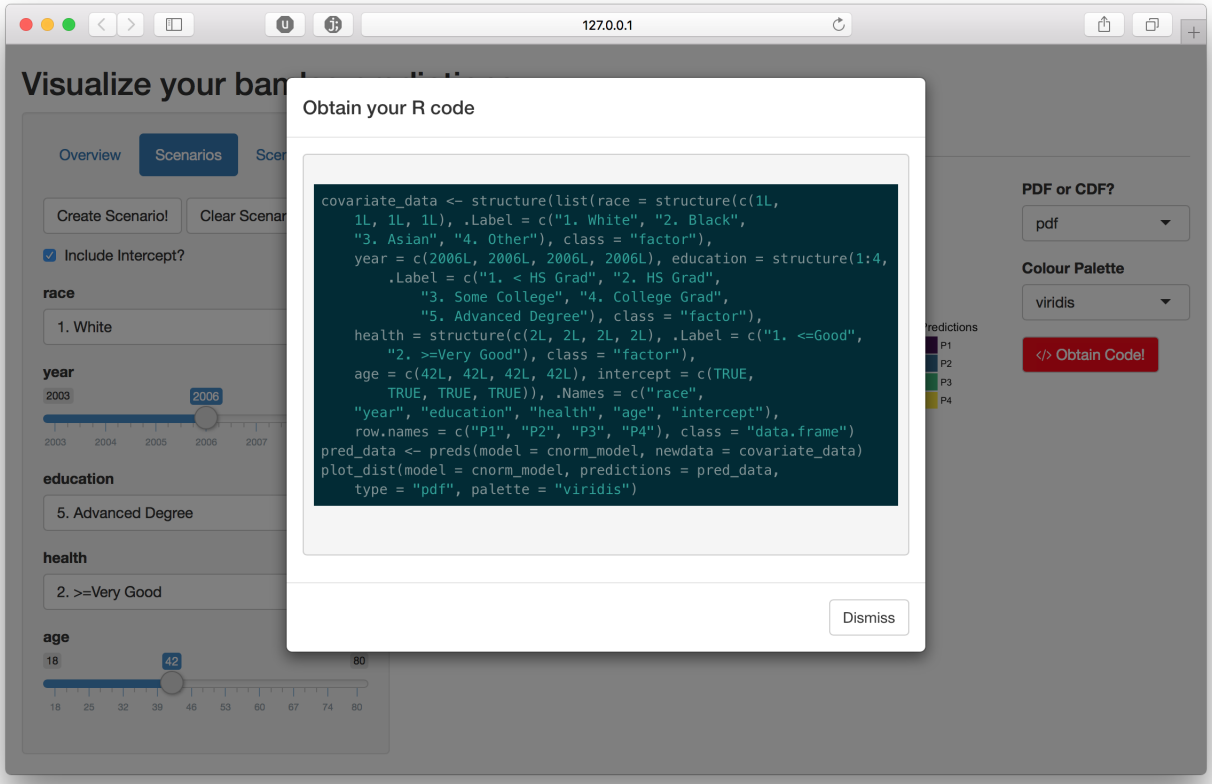

Figure 7: Modal window with formatted and highlighted code after pressing the "Obtain Code!" button

3.3.4 Scenario Data Tab

While the "Scenario" tab gives the ability to quickly specify covariate values, sometimes the user might want to type in exact values on which to make predictions. Furthermore, one might want to see what values were previously specified in the "Scenarios" tab. For both reasons the tab "Scenario Data" was created in bamlss.vis' left segment.

Figure 8 shows the tabs layout. As is shown, the only present web element is a table placed underneath the description "Edit scenario data here". This table, created with the R package `rhandsontable` (Owen, 2016) represents the editable version of all data input from the "Scenarios" tab. Columns represent the specified covariates, while one row counts as one "scenario". In Figure 8, it is possible to see that only three columns of the original six are currently visible. This cut-off was integrated to prevent an overlap with the plot. Nevertheless, the user can use the scrolling bar at the bottom to reach other columns.

To edit an observation from categorical variables, users can click on a small drop-down button in the cell which can be edited. The table recognizes categorical

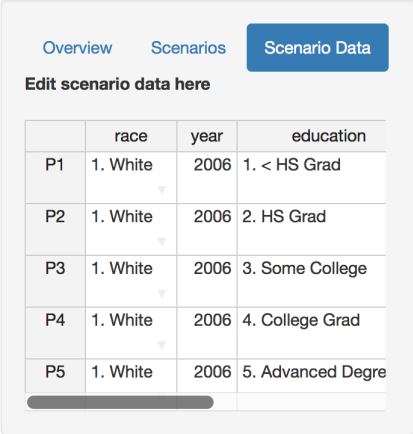

Figure 8: Scenario data tab.

variables and will then provide a menu where the desired value is to be selected. With numeric variables, users can select the cell and then input the value they wish to make predictions with. Values in logical variables can be specified by checking or unchecking a box in the cell.

With the ability to specify own covariate values also comes the ability in numeric cases to type in numbers that are not in the original variable's range. In the case of `cnorm_model`, this could mean that one tries to make predictions for incomes in the year `2050`, which is far beyond $max(year) = 2009$. To circumvent the unresponsible usage of model predictions, bamlss.vis will display a warning pop-up message with the name of the covariate combination (`P#`) where values were specified which are out of the original variable's range (Appendix: Figure A3).

3.3.5 Properties Tab

Previous chapters have described that the "Plot" tab visualizes the predicted distributions for each covariate combination specified in the "Properties" tab. However, while differences in distributions for each combination were visible (e.g. in Figure 6), it is not possible to directly infer influences of each covariate on the distributional moments.

To provide this functionality, the tab "Properties" was implemented in bamlss.vis, located in the right-hand segment. When opened, the "Properties" tab reveals two sub-tabs: "Influence graph" and "Table". As visible in Figure 9, the "Influence graph" tab consists of a graph on the left side next to a small bar with

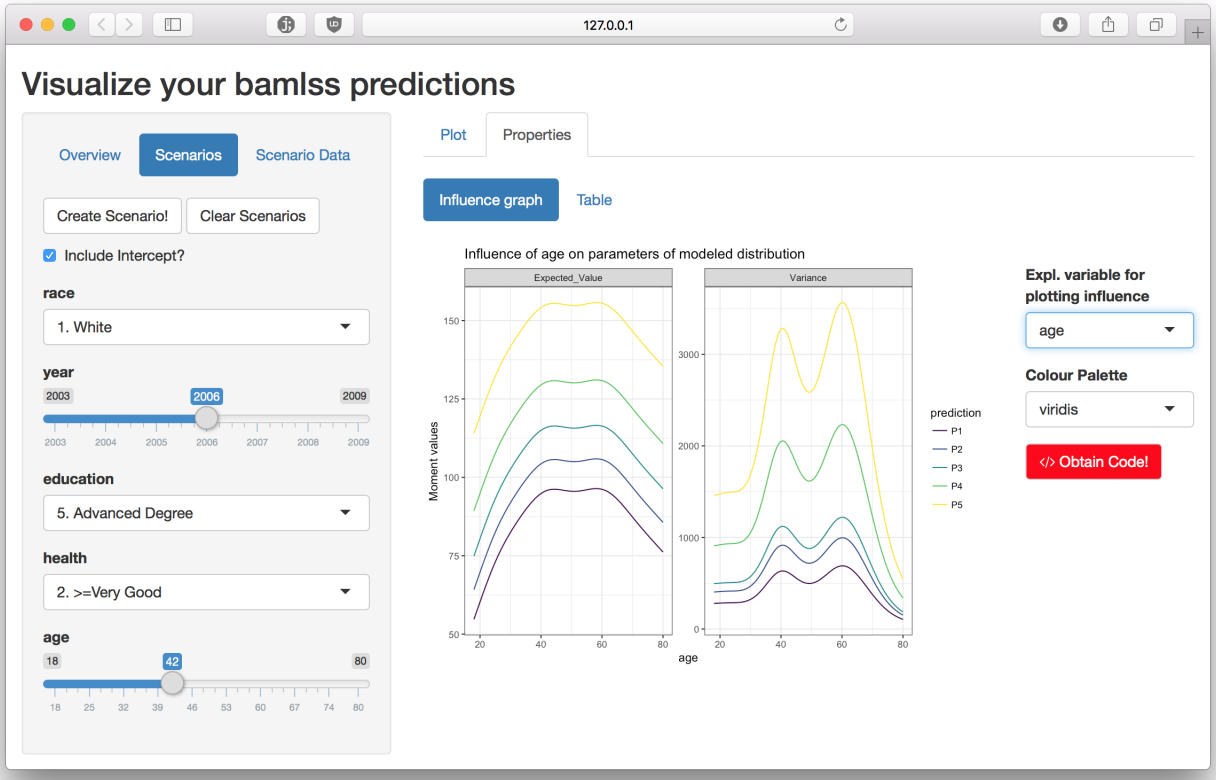

Figure 9: Influence of `age` on the first two moments of the predicted distributions for `cnorm_model`

additional options. The graph displays the changes in the first two moments over the whole range of a covariate that was chosen on the right side. This is repeated for each "scenario" specified on bamlss.vis' left-hand segment. As seen in Figure 9, the plot is divided into a `Expected_Value` and a `Variance` section. In this case, variable `age` was chosen and therefore represents the x-axis from the variables minimum to maximum value. The y-axis depicts values for each of the first two distributional moments. Each line displays the level which the predicted moments would assume for `age`'s possible values, depending on the chosen covariate combination in the "Scenarios" tab.

In Figure 9, five different scenarios which represent varying `education` levels were already specified. So in summary, this plot measures the influence of `age` on each of the first two moments of the modeled income distribution for healthy 42-year-old white males, depending on the `education` level.

For the influence of `age` on the first moment, it is possible to observe on all five lines that up until around 40 years, `age` has a positive effect on the expected income, then takes a small downturn which is recovered around age 60, after which a rising age decreases the expected income. This influence structure is

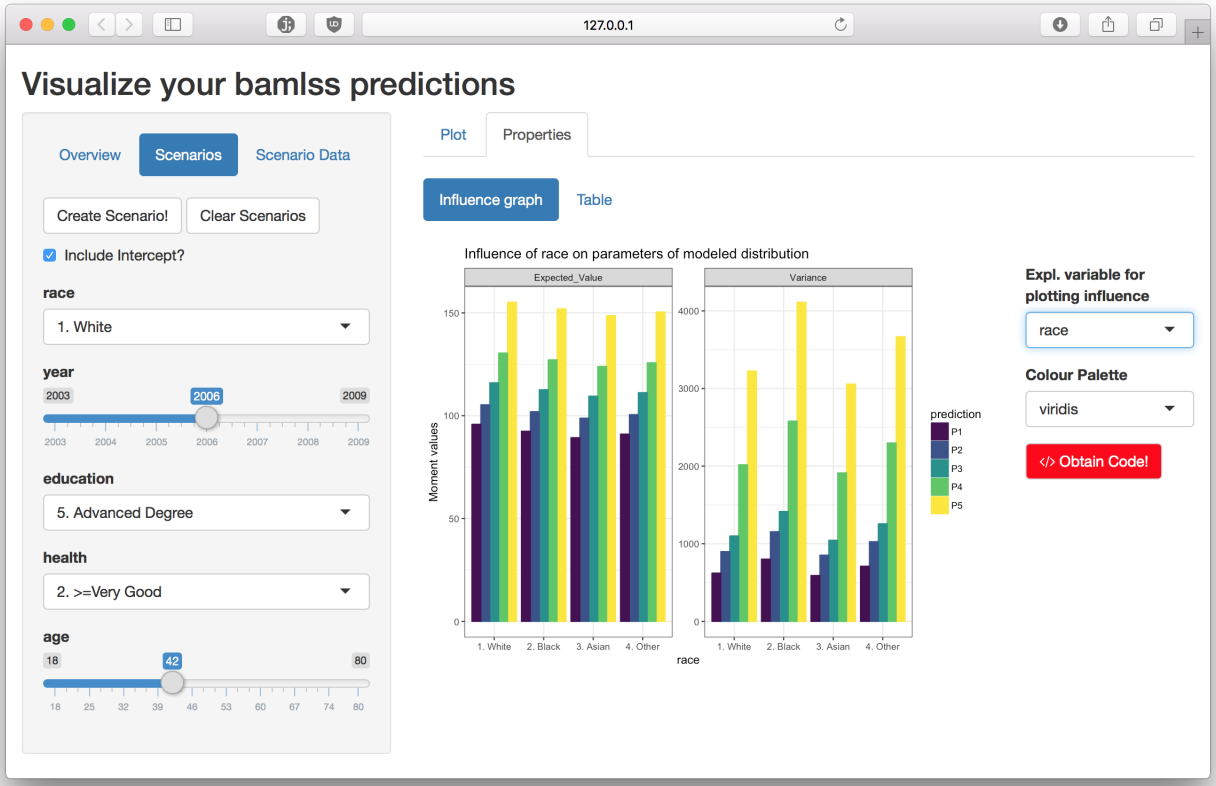

Figure 10: Influence of `race` categories on the first two moments of the predicted distributions for `cnorm_model`.

similar for all education levels because the effect of `age` on both parameters of the censored normal distribution was specified as a simple smooth spline with no `education` interaction.

For effects on the variance, a similar influence structure is observed, although the ups- and downs become more extreme as education level rises. This can be explained by the original model specification, where the variance σ^2 is not but only the standard deviation σ is modeled in relation to explanatory covariates. Thus, its effects have to be squared.

The effects of `age` on distributional moments can be visualized in lines because the underlying variable is measured in integers. However, bamlss.vis can also display differences in both moments depending on a categorical covariate. Figure 10 illustrates the influence graph when `race`, a qualitative covariate, is selected. In Figure 10, the x-axis again depicts the selected variable, `race`, with its possible outcomes `1.White` to `4. Other`. Every outcome displays five bars, one for each covariate combination specified in the "Scenarios" tab. As the five different covariate combinations depict increasing education levels, this plot shows the effects of both `education` and `race` on the expected income level and its variance.

As already observed in Figure 9, both the expected income and income variance increase with a higher education level. Furthermore, Figure 10 shows that 42-year old males are expected to earn less if they belong to the `2. Black` or `3. Asian` category than when they assume the `1. White` category. Still, it is visible that the effect of `education` leads to larger differences in the expected income than `race`.

Next to the graph in the "Influence graph" sub-tab three web elements are displayed. The first one, located under the description "Expl. variable for plotting influence", yields the ability to select the explanatory variable for which the influence plot shall be created. The second element lets the user choose a color palette, similar to Figure 6. In Figures 9 and 10, the "viridis" color palette was specified to account for colorblindness compatibility. Located below the palette selector the same red button as in Figure 6 is placed, which opens a modal window to reproduce the influence plot when it is pressed (Appendix: Figure A4).

The other sub-tab of "Properties" called "Table" is useful if the user solely wants to see the differences in distributional moment values across specified covariate combinations in the "Scenarios tab". Figure 11 shows the tab's layout. the only present element is a table which shows two columns, `Expected_Value` and `Variance` with computed values for the first two moments. Every row depicts one covariate combination specified in the "Scenario" tab.

3.4 Additional Functions

Based on `cnorm_model`, the previous chapters gave an overview of bamlss.vis' functionalities for continuous and univariate response distributions. However, the abilities of bamlss.vis go beyond the censored normal distribution and are applicable to almost all distributional families that the `bamlss` package provides. This chapter will demonstrate how bamlss.vis handles non-continuous or bivariate response distributions.

The fitted `bamlss` objects which bamlss.vis uses his section rely on simulated data generated by `bamlss.vis::model_fam_data()`. To sustain a correlation structure between covariates such that they can be exploited for useful modeling purposes, `model_fam_data()` simulates a three-dimensional uniform distribution with a sample space of $x_i \in [0, 1]$, $i = 1, 2, 3$. Then, inverse transform sampling with x_1 is used to obtain sample data for all supported `bamlss` distribution families.

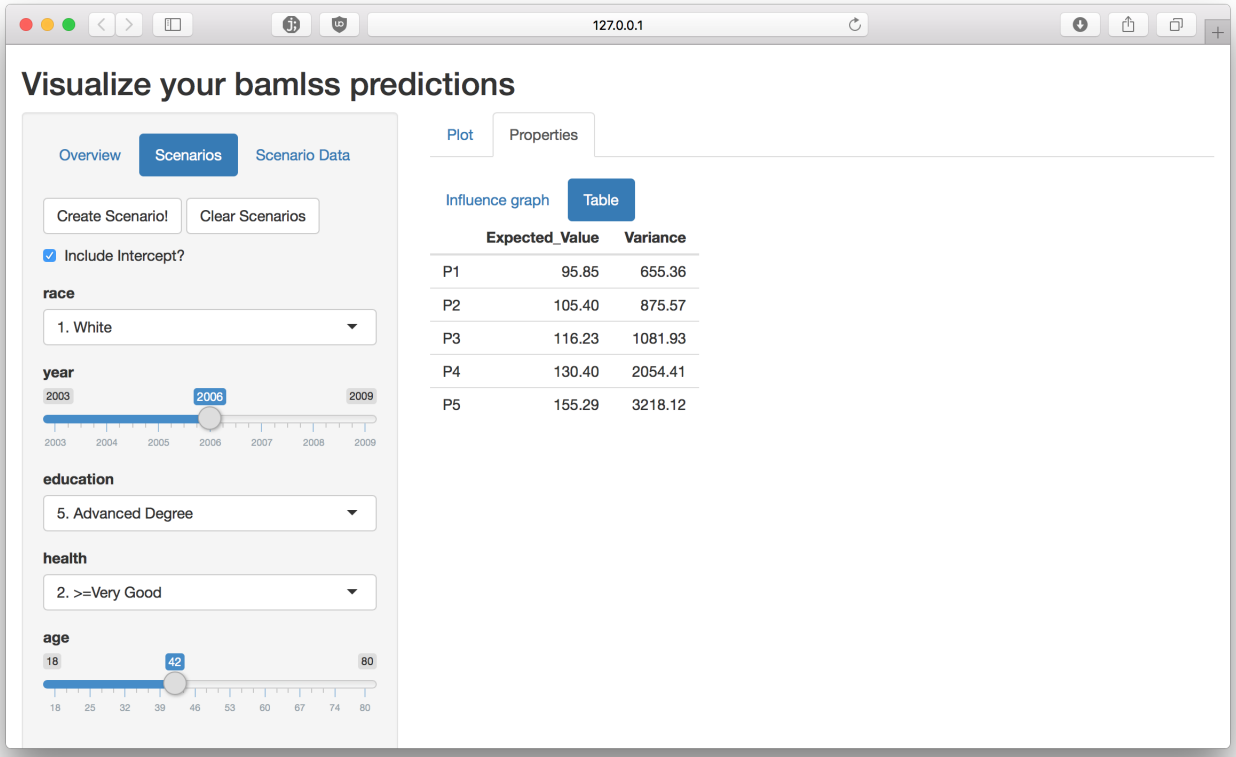

Figure 11: Expected Value and Variance for predicted distributions based on specified covariate combinations.

x_2 and x_3 are transformed to standard normally distributed variables. Covariates x_1 and x_2 are then used as explanatory variables for modeling the transformed response variable x_1.

3.4.1 Discrete Responses

Models with discrete responses are automatically recognized by bamlss.vis using the function `is.discrete`. In contrast to the continuous case, the application structure of bamlss.vis does not change but the distribution graph is transformed into a bar plot. Figure 12 shows the "Plot" tab of bamlss.vis for a fitted `bamlss` object modeling a Poisson-distributed response where three distinct covariate combinations were already specified in the "Scenarios" tab. Here, the three predicted probability mass functions for the different Poisson distributions are visible. As the Poisson distribution is a discrete family, probabilities $f(y)$ are displayed in bars on the y-axis. Possible outcomes of the Poisson distribution are shown on the x-axis. For every outcome, three bars for three different covariate combinations specified in the "Scenarios" are displayed. For example, prediction `P2` has the lowest expected probability $P(y = 0)$ compared to the other two

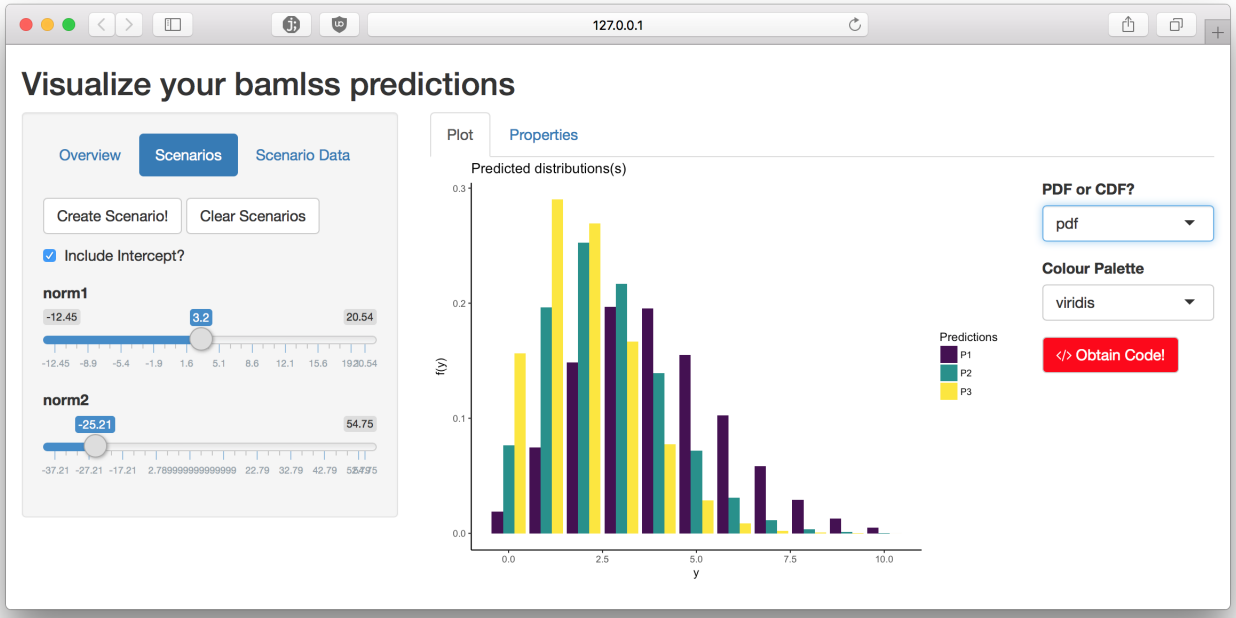

Figure 12: Plot tab with predicted probability mass functions for Poisson distributions with three specified covariate combinations in the "Scenarios" tab

combinations.

In theory, the Poisson distribution can assume all values $\{y \in \mathbb{Z} \mid y \geq 0\}$ up to ∞. To appropriately visualize different expected Poisson distribution, the x-axis of the graph is limited to $0, \ldots, x_{lim}$ where $x_{lim} = (\max(\lambda_1, \ldots, \lambda_K) \cdot 2) + 3$ for predicted parameters λ_i from covariate combinations $1, \ldots, K$.

Cumulative distribution functions for discrete response distributions can also be obtained by using the the same list selector element as with continuous response distributions to the graphs right side. The cdf is then displayed as a step-wise graph (Appendix: Figure A5). Moreover, specifying other color palettes and obtaining plot reproduction code is still also possible.

While the Poisson distribution has discrete observations, its first two moments $E(y) = Var(y) = \lambda$ assume values on a continuous scale. Therefore, the appearance of the "Properties" tab with "Influence graph" and "Table" sub-tabs are identical to the continuous case described in Section 3.3.5.

3.4.2 Multivariate Responses

In contrast to the `gamlss` R package without extensions, `bamlss` also supports multivariate response distributions. From the range of multivariate distributions, only the multivariate normal distribution is currently implemented in `bamlss`

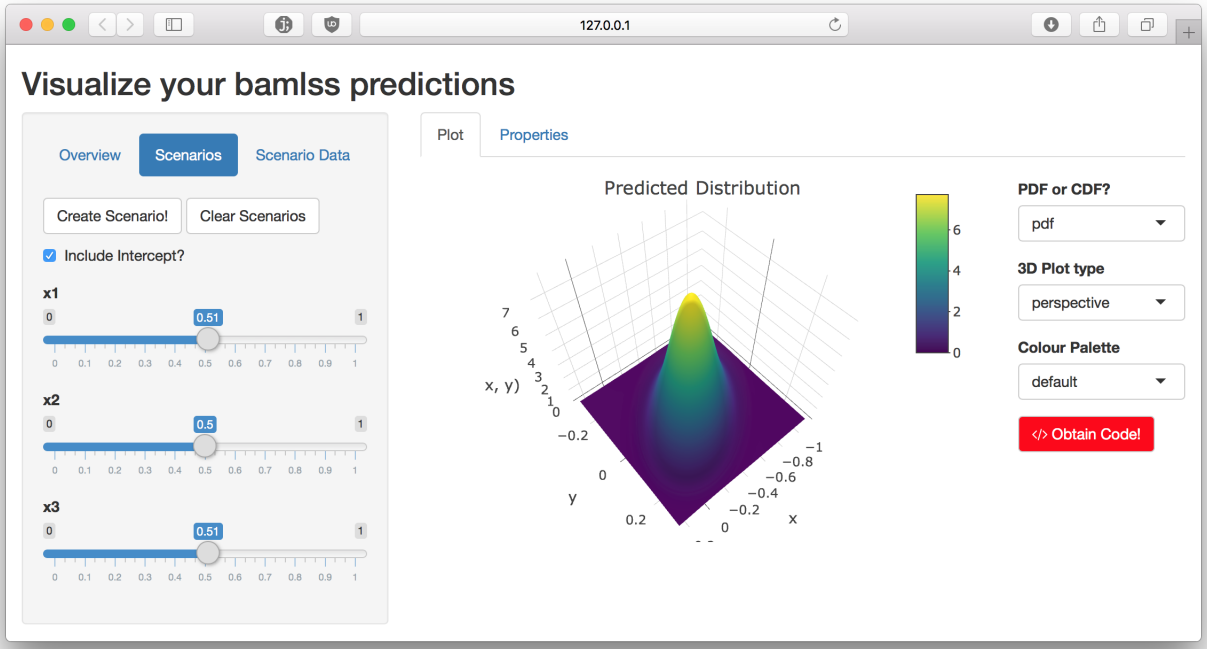

Figure 13: Predicted distribution for a multivariate normal distributed response based on the covariate specification of the "Scenarios" tab.

and therefore supported by bamlss.vis. Nevertheless, because of its existing 3D graphical framework, bamlss.vis can easily be extended for new multivariate distributions.

For graphical reasons, bamlss.vis can only display bivariate responses. Figure 13 shows the "Plot" and "Scenarios" tab with a specified scenario from the left tab. As visible in Figure 13, a bivariate normal distribution is displayed. Multivariate response predictions in bamlss.vis are always only dependent on the last specified covariate combination without any prediction comparisons. Furthermore, multivariate predictions are displayed using the R package `plotly` (Sievert et al., 2017), which adds a layer of interactivity to the graph. Using a graph generated with `plotly`, the user can hold and drag the distribution to view the shape from all perspectives. Moreover, hovering over the distribution surface yields current x, y and z values.

Located on the right-hand side of the plot are web elements mostly known from previous Sections. Bamlss.vis supports the display of multivariate cumulative distribution functions, which can be selected in the first element titled "PDF or CDF?". An example is given by Figure A6 in the Appendix. Placed below the first element and the description "3D Plot type" is another drop-down list for specifying the type of 3D Plot. In addition to "perspective", the options "contour" and "image" can be selected. Examples for both options can be found

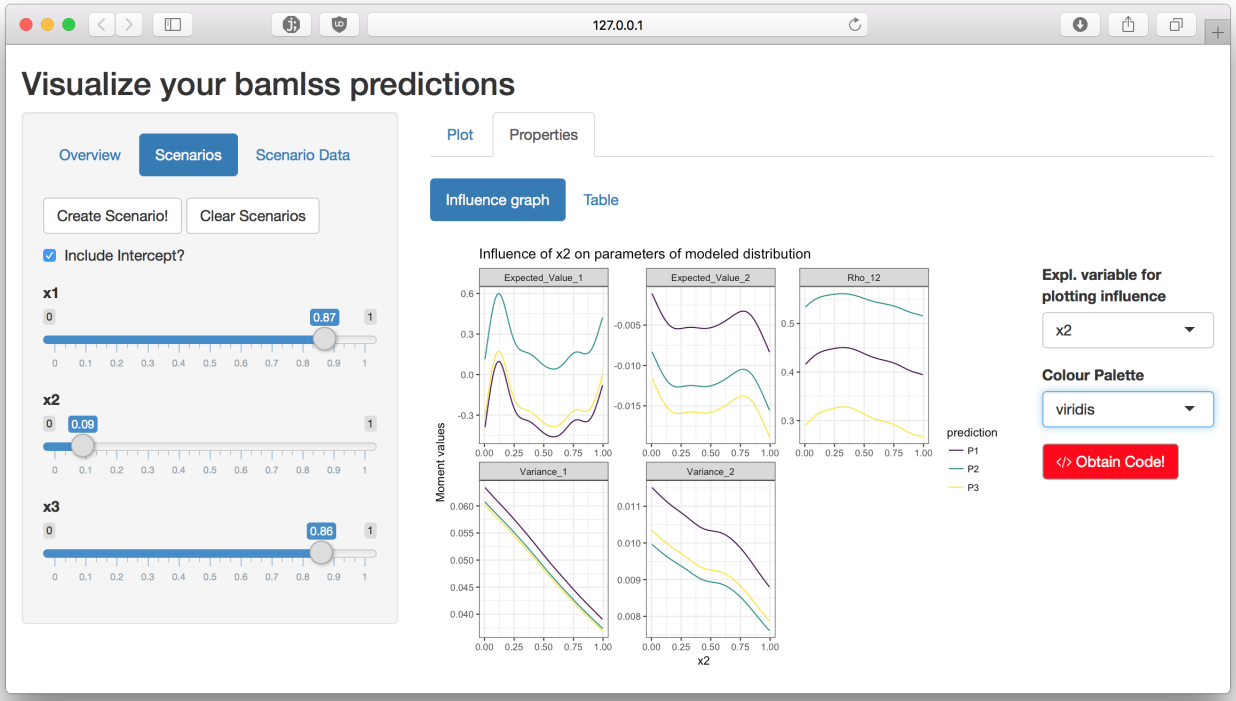

Figure 14: "Influence graph" tab for a selected bivariate normal `bamlss` with three specified covariate combinations.

in Appendix Figures A7 and A8 for a contour and image plot, respectively.

Given that the bivariate normal distribution consists of two univariate normal distributions with covariance $\rho\sigma_1\sigma_2$, its first two moments are $\boldsymbol{\mu} = \binom{\mu_1}{\mu_2}$ and $\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$. In total, this amounts to five parameters for which covariate influences can be graphically displayed. Figure 14 shows the "Influence graph" sub-tab for a specified multivariate normal `bamlss`. As shown, the displayed graph is divided into five subgraphs, instead of two in the univariate case. Similar to the influence plot for univariate response distributions, the relationship between the moments and `x2` (explanatory variable selected in this case) is plotted for every previously specified covariate combination. Furthermore, the user can select different color palettes and obtain R code for plot reproduction.

The "Table" sub-tab of the "Properties" tab again differs for bi-variate normal distributions. Here, the displayed table for distributional moments depending on covariate combinations specified in the "Scenarios" tab is expanded to the five moments of the bivariate normal distribution. For an example, Figure A9 in the Appendix is provided.

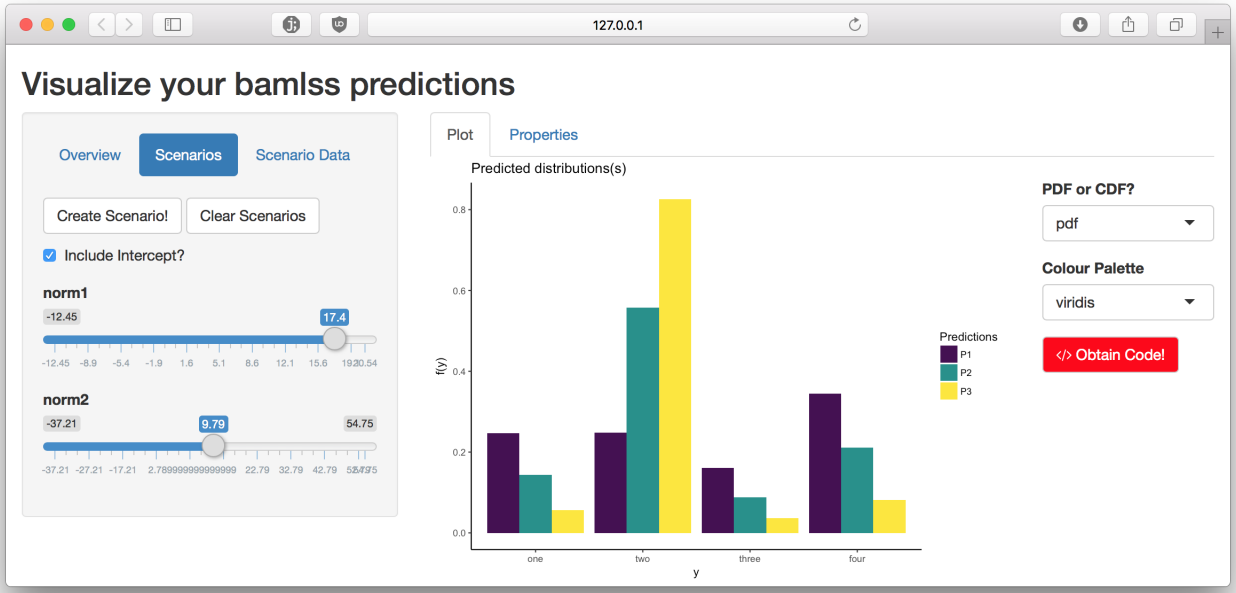

Figure 15: Predicted multinomial response in the "Plot tab" based on three specified covariate combinations

3.4.3 Multinomial Responses

For modeling unordered categorical response variables, `bamlss` supports the multinomial distribution family, a generalization of the binomial distribution. Loading a `bamlss` with a multinomial response is possible in bamlss.vis, although the graphical appearance of predicted distributions and the influence plot differs from usual discrete cases.

Figure 15 shows the "Plot" tab with three specified covariate combinations after selecting a sample `bamlss` with multinomial response in the "Overview" tab. The modeled sample response variable can have four different outcomes `one`, `two` , `three` and `four` which are visible on the x-axis. The predicted probabilities for each of those outcomes are displayed on the y-axis. This is realized graphically in bars for each of the specified covariate combinations, which are three in the present case. Interpreting Figure 15, one can see that with each higher covariate combination (`P1` to `P3`), the probability to assume category `two` increases, while the probabilities for all other categories decrease.

Moreover, tweaking the plot with the web elements in the right sidebar is possible, similar to bamlss.vis' behavior for discrete or continuous univariate responses. The option for displaying the cumulative distribution function is inoperative, due to the lack of a reasonable graphical representation.

Using own covariate combinations and then visualizing the probability differences

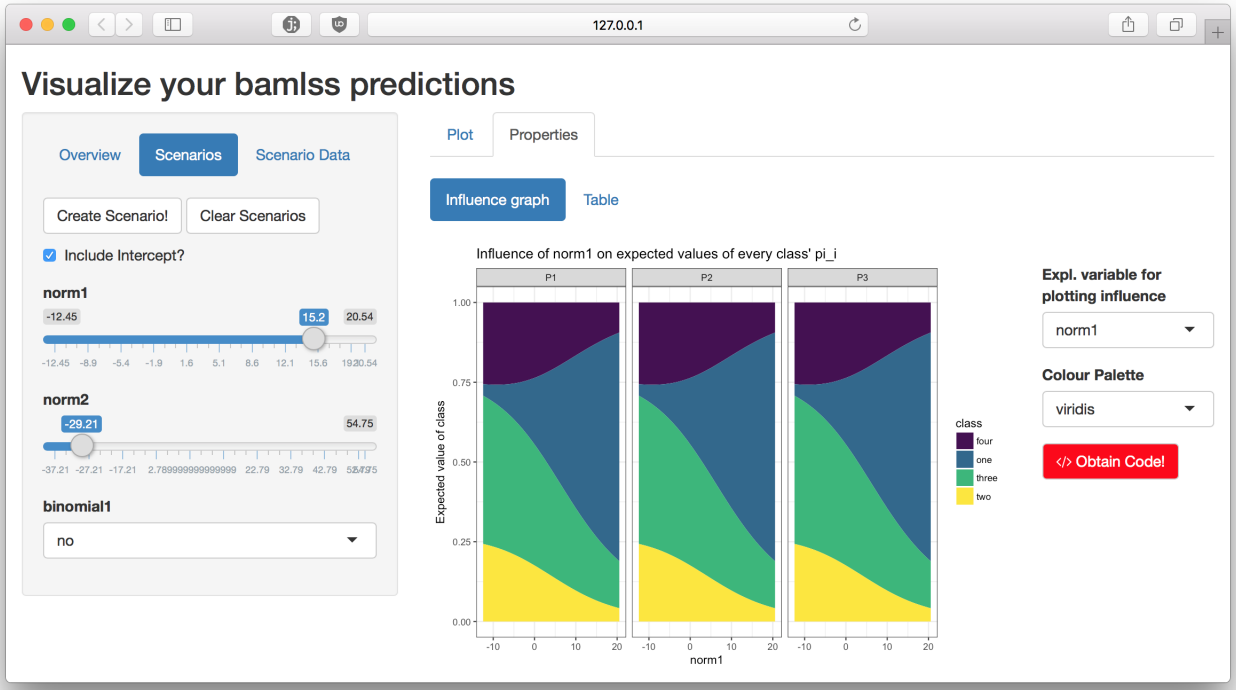

Figure 16: Influence of variable `norm1` on expected first moments of multinomial distribution

for each class as in Figure 15 can already be used to obtain influence tendencies of covariates on the distributional moments, because for a multinomial distribution y with K classes and $n = 1$ draws $E(y) = \boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)'$ holds, which means the first moment is already graphically displayed. However, the variation of $\boldsymbol{\pi}$ over the whole range of an explanatory variable might be of interest. To yield this functionality, the "influence graph" subtab is provided. Figure 16 shows the influence plot for a sample multinomial model. In Figure 16, the predicted values for $\pi_1, \ldots, \pi_4$ are displayed over the range of the selected variable, `norm1`.

This is then repeated for every covariate combination which was specified in the "Scenarios" tab. As visible, the y-axis limits are always 0 and 1, due to $\pi_1 = \ldots = \pi_4 \geq 0$ and $\sum_{i=1}^{4} \pi_i = 1$ holding true. Furthermore, one can see that an increase of `norm1` is associated with an increase of the probability to fall into category "one" (π_1) and a decrease for all other categories, across all three covariate combinations.

Displaying the influence of explanatory covariates on probabilities π_i is also possible for categorical variables. For an example, Figure A10 in the Appendix is provided.

4 Conclusion

This thesis laid out the foundations of Bayesian Additive Models for Location Scale and Shape by first describing its ancestors (AM, STAR, Generalized STAR) and relatives (GAMLSS) and then the BAMLSS model framework itself. It became clear that distributional regression provides a highly flexible modeling framework, from which researchers will presumably increasingly benefit in the future. To make fitted models more accessible for users this thesis introduced the interactive application bamlss.vis based on the `bamlss` R package. With this new tool at hand, users will hopefully be encouraged about applying distributional regression models, specifically BAMLSS, and then draw more resilient conclusions.

Using bamlss.vis, the user can visualize predicted response distributions based on interactively chosen covariate combinations. Furthermore, bamlss.vis provides the ability to show the influence of a selected explanatory covariate on the first two moments of any modeled response variable, including continuous, discrete and bivariate distributions. Both of these key functions help the user draw useful inference from fitted `bamlss` objects.

Moreover, bamlss.vis is designed to be highly customizable. The user-specified covariate combinations (that represent the core of the analysis) can be individually edited and expanded. The graphical appearance of all plots can be changed to display other color palettes or other distribution types (pdf/cdf). Proceeding the finished analysis, the user can obtain R code to reproduce the displayed plot with all chosen options.

Nevertheless, further work on making distributional regression models more understandable can be done. The idea of visualizing predicted distributions and making influence plots should also be applied to other variations of distributional regression, for example BayesX or GAMLSS, which offer a large variety of supported distributions. Moreover, the implementation of bamlss.vis can be further developed. For example, the user might want to know the influence of an explanatory variable on a self-specified figure which can be constructed from the estimated parameters, like the Gini coefficient or a higher-order distributional moment. This could be implemented by adding a text field which computes the desired figure.

Lastly, the framework of bamlss.vis relies heavily on the quality of the fitted

model. Wrongly specified response distributions could, for example, also lead to wrong conclusions about the covariate influence. Further work on the selection of response distributions, model choice and variable selection would therefore be highly beneficial.

A Appendix

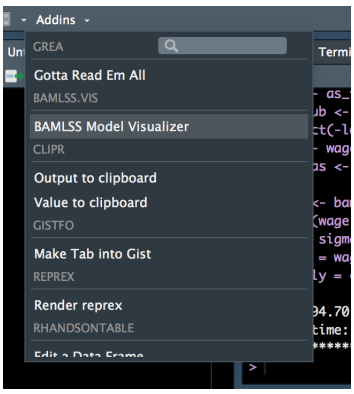

Figure A1: Button to start the main application of bamlss.vis in RStudio.

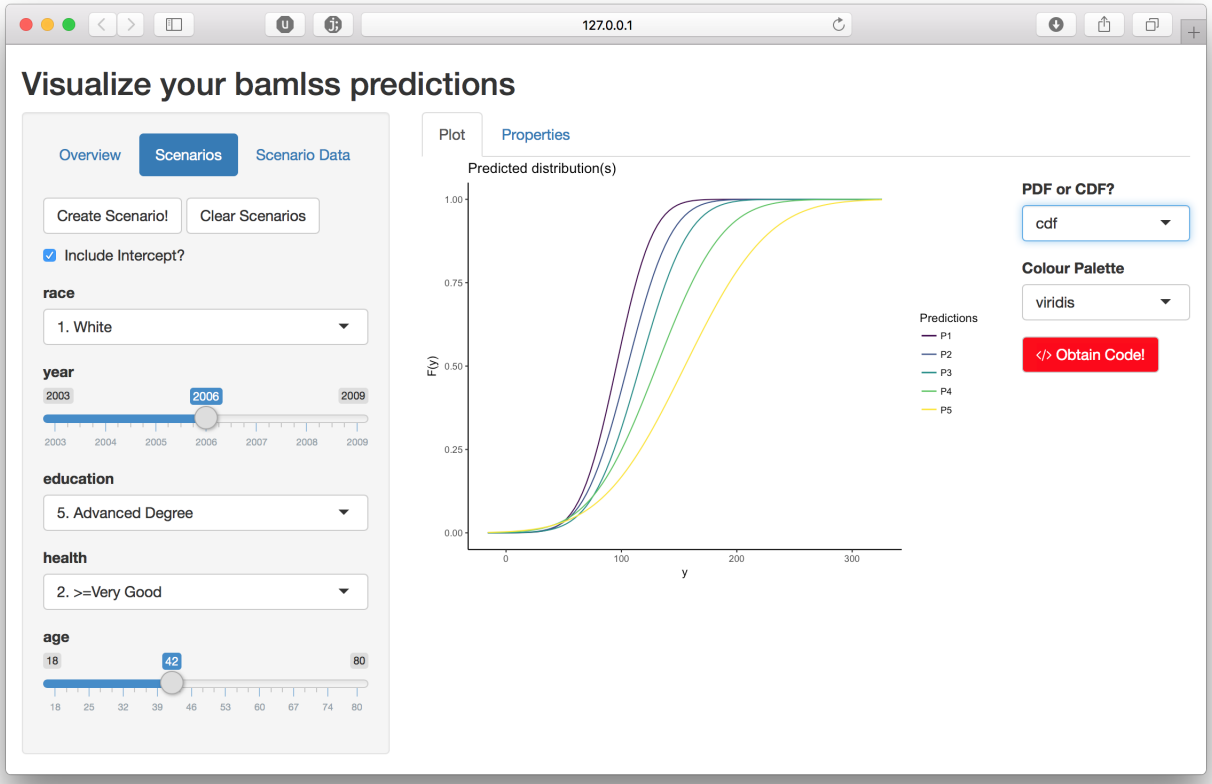

Figure A2: Cumulative Distribution Function plot output for different education levels based on the `Wage` dataset.

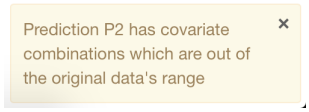

Figure A3: Warning message when specifying covariate combinations which are out of range.

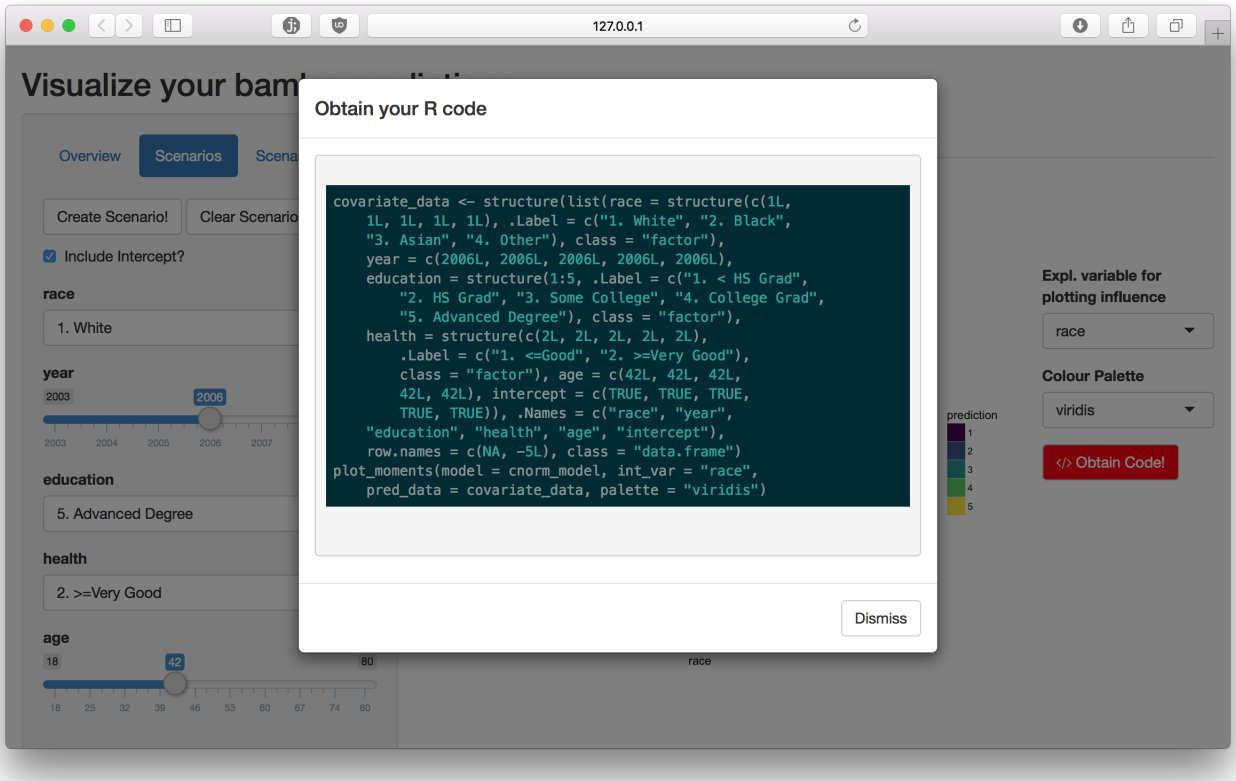

Figure A4: Modal window to display code for reproducing the influence plot.

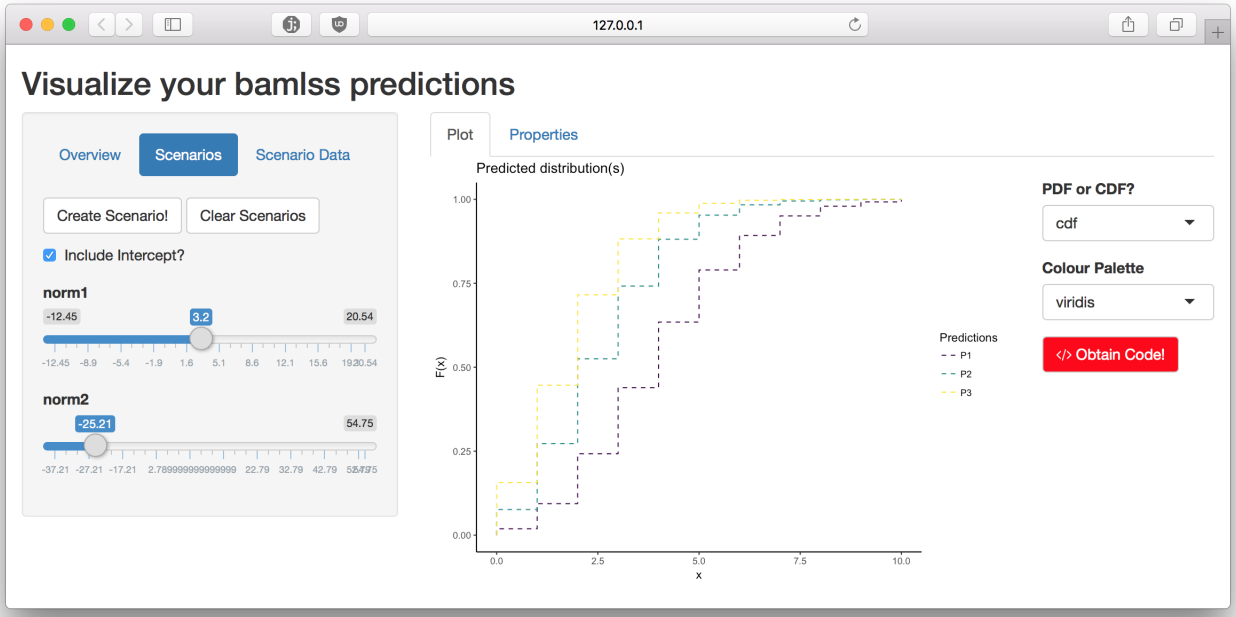

Figure A5: Predicted Cumulative Distribution Function for three Poisson distributions based on user-input covariate values.

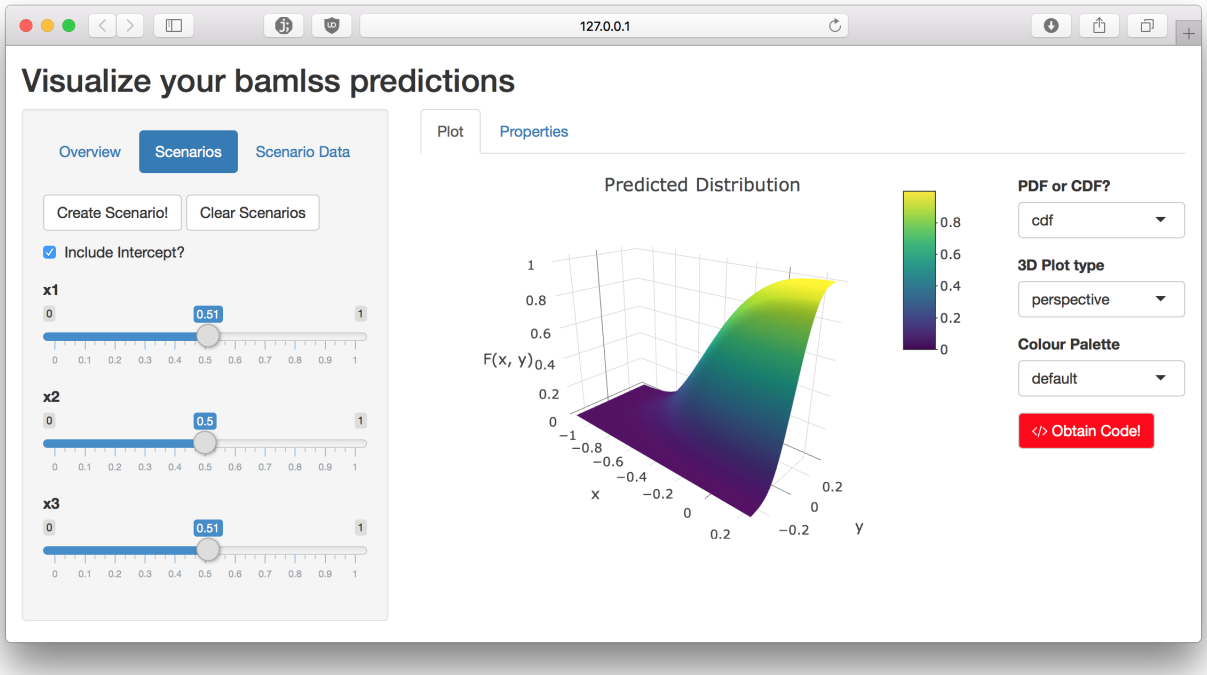

Figure A6: Predicted distribution for a multivariate normal distributed response based on the covariate specification of the "Scenarios" tab.

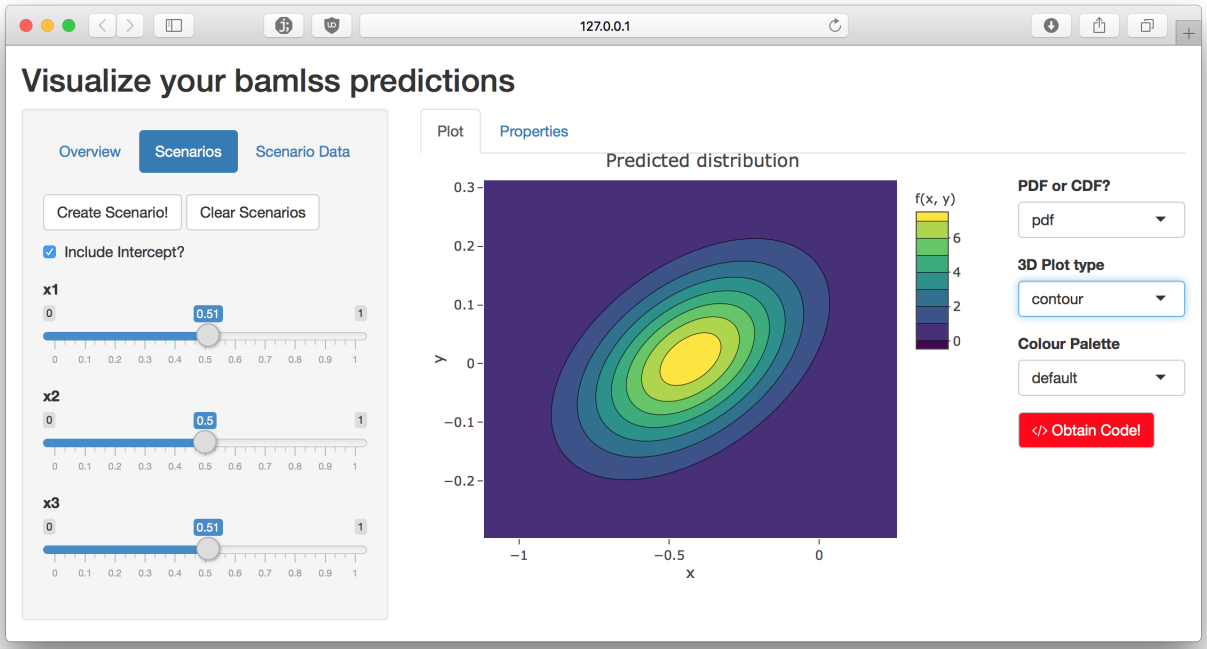

Figure A7: Contour plot for the predicted multivariate normal distribution based on user inputs in "Scenarios" tab.

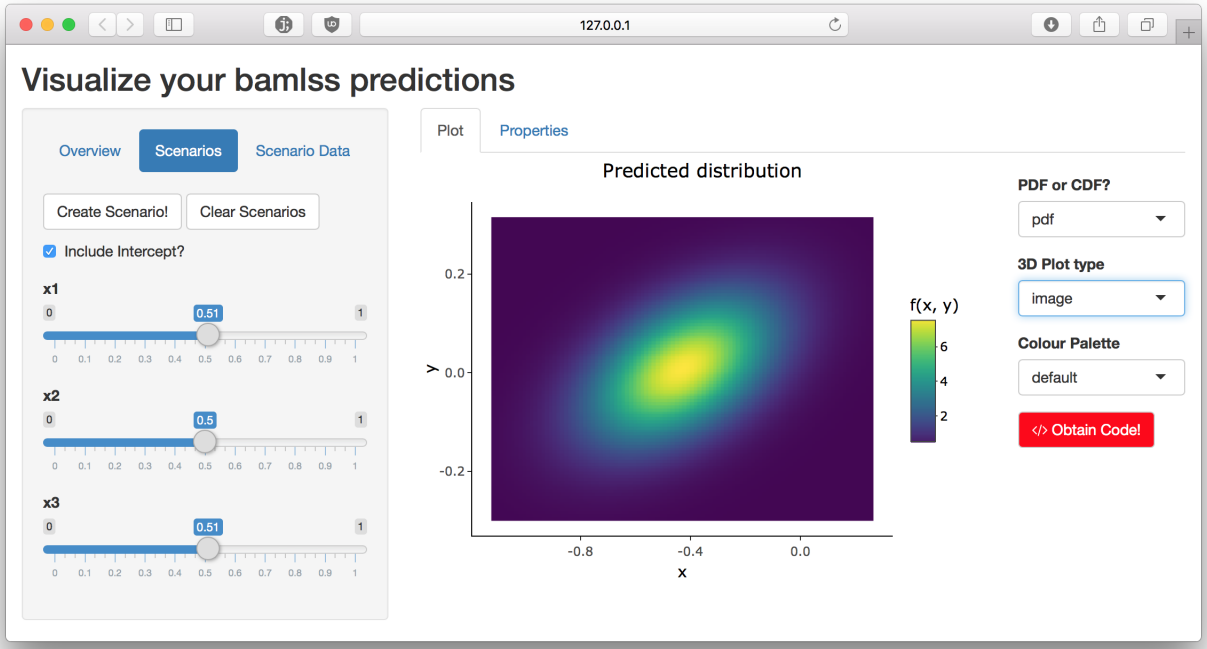

Figure A8: Image plot for the predicted multivariate normal distribution based on user inputs in "Scenarios" tab.

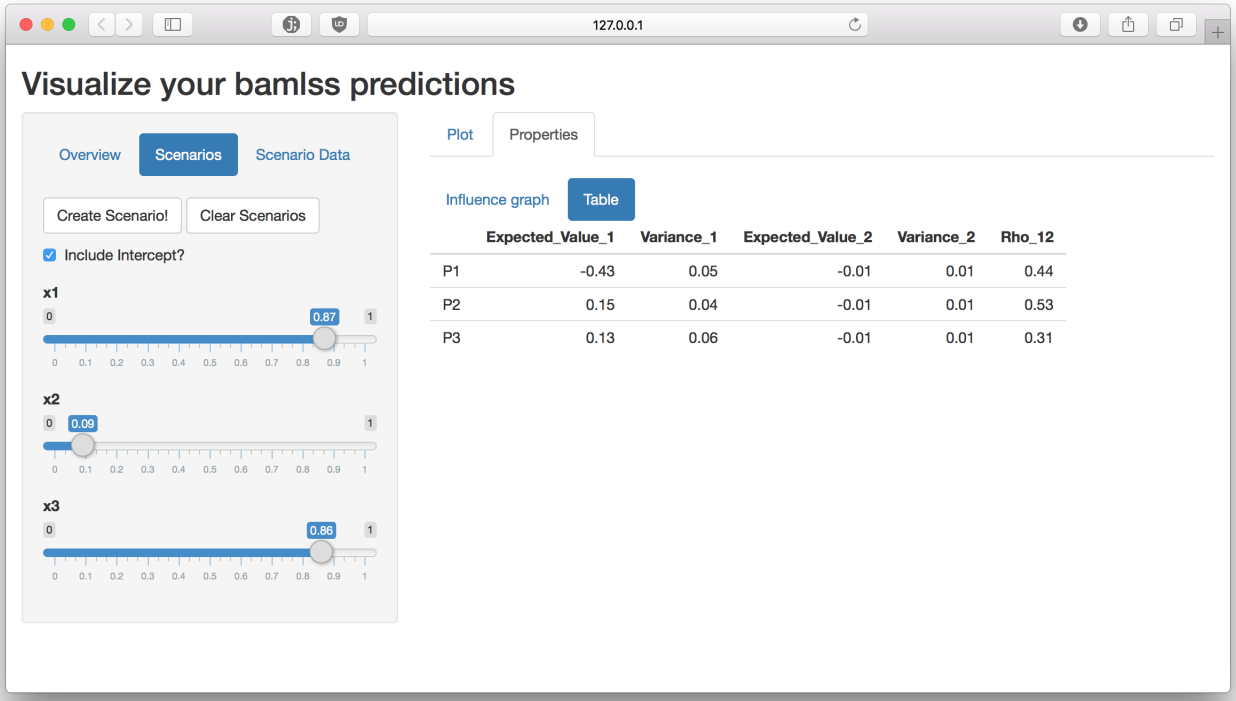

Figure A9: Table view for expected moments of the bivariate normal distribution based on three specified covariate combinations in the "Scenarios" tab.

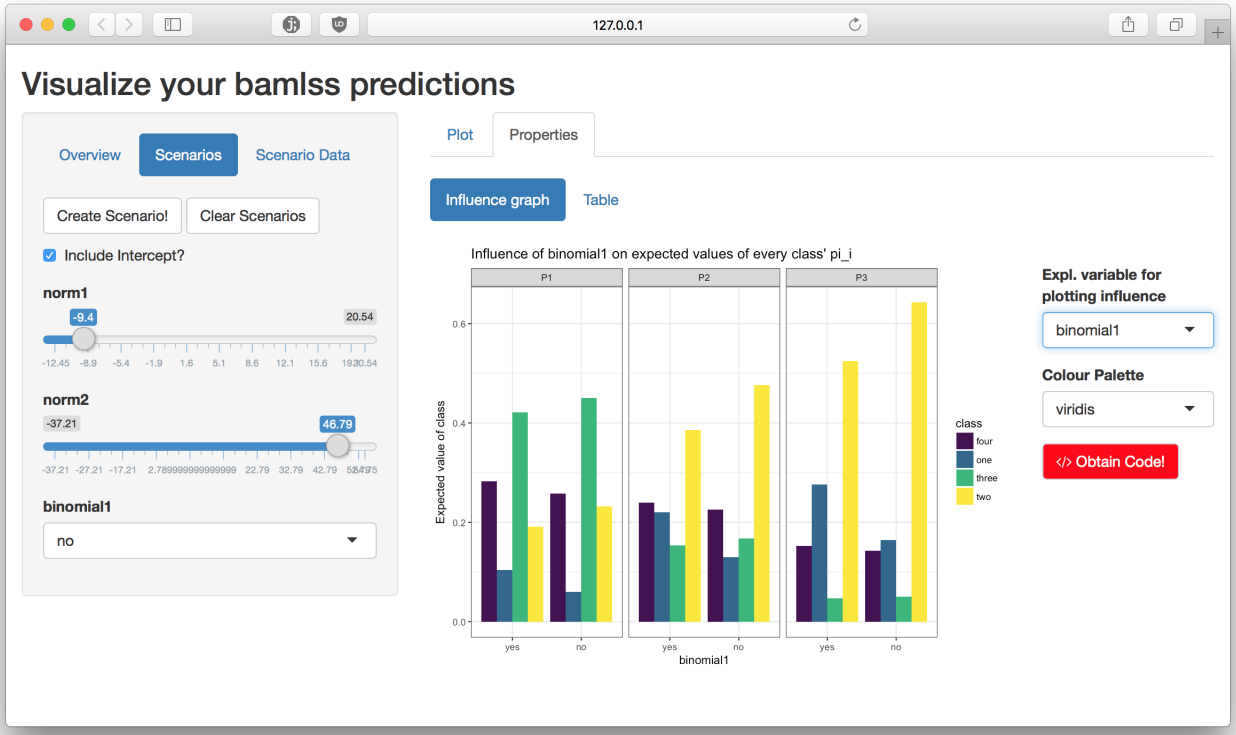

Figure A10: Sample influence plot for categorical covariates on expected class probabilities in multinomial response `bamlss`

Bibliography

A. Brezger and S. Lang. Generalized structured additive regression based on bayesian p-splines. *Computational Statistics & Data Analysis*, 50:967–991, February 2006.

A. Buja, T. Hastie, and R. Tibshirani. Linear smoothers and additive models. *The Annals of Statistics*, pages 453–510, 1989.

Bureau of Labor Statistics. OES Data, 2016. URL `https://www.bls.gov/oes/tables.htm`. [Online; accessed 28-Nov-2017].

W. Chang, W. Cheng, J. J. Allaire, Yihui X., and J. McPherson. *shiny: Web Application Framework for R*, 2017. URL `https://CRAN.R-project.org/package=shiny`. R package version 1.0.5.

T. J. Cole and P. J. Green. Smoothing reference centile curves: The lms method and penalized likelihood. *Statistics in Medicine*, 11(10):1305–1319, 1992. ISSN 1097-0258. doi: 10.1002/sim.4780111005. URL `http://dx.doi.org/10.1002/sim.4780111005`.

L. Fahrmeir, T. Kneib, and S. Lang. Penalized additive regression for space-time data: a bayesian perspective, 2003. URL `http://nbn-resolving.de/urn/resolver.pl?urn=nbn:de:bvb:19-epub-1687-9`.

L. Fahrmeir, T. Kneib, S. Lang, and B. Marx. *Regression: Models, Methods and Applications*. Springer Berlin Heidelberg, 2013. ISBN 9783642343339. URL `https://books.google.de/books?id=EQxU9iJtipAC`.

J. H. Friedman and W. Stuetzle. Projection pursuit regression. *Journal of the American statistical Association*, 76(376):817–823, 1981.

S. Garnier. *viridis: Default Color Maps from 'matplotlib'*, 2017. URL `https://CRAN.R-project.org/package=viridis`. R package version 0.4.0.

W. H. Greene. *Econometric Analysis*. Pearson International Edition. Pearson Education, Limited, 2012. ISBN 9780273753568. URL `https://books.google.de/books?id=-WFPYgEACAAJ`.

T. Hastie. *gam: Generalized Additive Models*, 2017. URL `https://CRAN.R-project.org/package=gam`. R package version 1.14-4.

T. J. Hastie and R. J. Tibshirani. *Generalized Additive Models*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 1990. ISBN 9780412343902. URL `https://books.google.de/books?id=qa29r1Ze1coC`.

G. James, D. Witten, T. Hastie, and R. Tibshirani. *ISLR: Data for an Introduc-*

tion to Statistical Learning with Applications in R, 2017. URL https://CRAN.R-project.org/package=ISLR. R package version 1.2.

N. Klein, T. Kneib, and S. Lang. Bayesian structured additive distributional regression. Working Papers in Economics and Statistics 2013-23, Universität Innsbruck, Institut für Finanzwissenschaft, Innsbruck, 2013. URL http://hdl.handle.net/10419/101101.

N. Klein, T. Kneib, S. Lang, and A. Sohn. Bayesian structured additive distributional regression with an application to regional income inequality in germany. *Ann. Appl. Stat.*, 9(2):1024–1052, June 2015. doi: 10.1214/15-AOAS823. URL https://doi.org/10.1214/15-AOAS823.

N. M. Laird and J. H. Ware. Random-effects models for longitudinal data. *Biometrics*, pages 963–974, 1982.

R. I. Lerman and S. Yitzhaki. A note on the calculation and interpretation of the gini index. *Economics Letters*, 15(3-4):363–368, 1984.

G. Matheron. Principles of geostatistics. *Economic geology*, 58(8):1246–1266, 1963.

J. A. Nelder and D. Pregibon. An extended quasi-likelihood function. *Biometrika*, 74 (2):221–232, 1987.

J. A. Nelder and R. W. M. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3):370–384, 1972.

U. Olsson. *Generalized Linear Models: An Applied Approach*. Lightning Source, 2002. ISBN 9789144041551. URL https://books.google.de/books?id=SPljHQAACAAJ.

J. Owen. *rhandsontable: Interface to the 'Handsontable.js' Library*, 2016. URL https://CRAN.R-project.org/package=rhandsontable. R package version 0.3.4.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017. URL https://www.R-project.org/.

R. A. Rigby and D. M. Stasinopoulos. *Mean and Dispersion Additive Models*, pages 215–230. Physica-Verlag HD, Heidelberg, 1996. ISBN 978-3-642-48425-4. doi: 10.1007/978-3-642-48425-4_16. URL https://doi.org/10.1007/978-3-642-48425-4_16.

R. A. Rigby and D. M. Stasinopoulos. The gamlss project: a flexible approach to statistical modelling. In *New trends in statistical modelling: Proceedings of the 16th international workshop on statistical modelling*, volume 337, page 345. University of Southern Denmark, June 2001.

R. A. Rigby and D. M. Stasinopoulos. Generalized additive models for location, scale and shape. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 54 (3):507–554, 2005.

RStudio, Inc. Shiny, 2017. URL `https://shiny.rstudio.com`. [Online; accessed 28-Nov-2017].

C. Sievert, C. Parmer, T. Hocking, S. Chamberlain, K. Ram, M. Corvellec, and P. Despouy. *plotly: Create Interactive Web Graphics via 'plotly.js'*, 2017. URL `https://CRAN.R-project.org/package=plotly`. R package version 4.7.1.

D. M. Stasinopoulos and R. A. Rigby. Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, 23(7):1–46, 2007.

M. D. Stasinopoulos, R. A. Rigby, G. Z. Heller, V. Voudouris, and F. De Bastiani. *Flexible Regression and Smoothing: Using GAMLSS in R.* Chapman & Hall/CRC The R Series. CRC Press, 2017. ISBN 9781351980371. URL `https://books.google.de/books?id=1h-9DgAAQBAJ`.

N. Umlauf, N. Klein, and A. Zeileis. Bamlss: Bayesian additive models for location, scale and shape (and beyond). Working papers, Working Papers in Economics and Statistics, 2017. URL `https://EconPapers.repec.org/RePEc:inn:wpaper:2017-05`.

United States Census Bureau. Supplement to current population survey, March 2011. URL `http://www.nber.org/cps/cpsmar11.pdf`. [Online; accessed 28-Nov-2017].

H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2009. ISBN 978-0-387-98140-6. URL `http://ggplot2.org`.

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

Y. Xie. *formatR: Format R Code Automatically*, 2017. URL `https://CRAN.R-project.org/package=formatR`. R package version 1.5.

Ich versichere, dass ich die Arbeit selbständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen oder anderen Quellen entnommen sind, sind als solche kenntlich gemacht. Die schriftliche und elektronische Form der Arbeit stimmen überein. Ich stimme der Überprüfung der Arbeit durch eine Plagiatssoftware zu.

Stanislaus Stadlmann, 18. Dezember 2017