The visualization, unbiased estimation and interpretation of distributional regression models

By

Stanislaus Stadlmann

A thesis jointly submitted to

Macquarie University, Sydney
Department of Mathematics and Statistics
Faculty of Science and Engineering

and

Georg-August Universität Göttingen
Chairs of Statistics and Econometrics
Faculty of Business and Economics

for the degrees of

Doctor of Philosophy (Sydney)
Dr. rer. pol. (Göttingen)

31st of July 2021

MACQUARIE
University
SYDNEY·AUSTRALIA

GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Typeset in LaTeX 2_ε.

Statement of Originality

(Macquarie University)

This thesis is submitted jointly to Macquarie University, Sydney and Georg-August University, Göttingen in fulfilment of the requirements for the degrees of Doctor of Philosophy (Macquarie University) and Doktor der Wirtschaftswissenschaften (Universität Göttingen) in accordance with the Cotutelle agreement dated 22nd of August 2019.

This work has not previously been submitted for a degree or diploma in any university. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

Versicherung bei Zulassung zur Promotionsprüfung,

engl: Declaration for admission to the doctoral examination

(University of Göttingen)

Ich versichere,

1. dass ich die eingereichte Dissertation "The visualization, unbiased estimation and interpretation of distributional regression models" selbstständig angefertigt habe und nicht die Hilfe Dritter in einer dem Prüfungsrecht und wissenschaftlicher Redlichkeit widersprechenden Weise in Anspruch genommen habe,

2. dass ich das Prüfungsrecht einschließlich der wissenschaftlichen Redlichkeit – hierzu gehört die strikte Beachtung des Zitiergebots, so dass die Übernahme fremden Gedankenguts in der Dissertation deutlich gekennzeichnet ist – beachtet habe,

3. dass beim vorliegenden Promotionsverfahren kein Vermittler gegen Entgelt eingeschaltet worden ist sowie im Zusammenhang mit dem Promotionsverfahren und seiner Vorbereitung

    - kein Entgelt gezahlt oder entgeltgleiche Leistungen erbracht worden sind

- keine Dienste unentgeltlich in Anspruch genommen wurden, die dem Sinn und Zweck eines Prüfungsverfahrens widersprechen

4. dass ich eine entsprechende Promotion nicht anderweitig beantragt und hierbei die eingereichte Dissertation oder Teile daraus vorgelegt habe.

Mir ist bekannt, dass Unwahrheiten hinsichtlich der vorstehenden Versicherung die Zulassung zur Promotionsprüfung ausschließen und im Falle eines späteren Bekanntwerdens die Promotionsprüfung für ungültig erklärt werden oder der Doktorgrad aberkannt werden kann.

Stanislaus Stadlmann
30 July 2021

Contributions

This chapter provides a brief summary of the three papers comprised in this thesis. Moreover, the contributions of each author are declared in detail.

Interactively visualizing distributional regression models with distreg.vis

Stadlmann, S. & Kneib, T. (2021). Interactively visualizing distributional regression models with distreg.vis. *Statistical Modelling*, doi: https://doi.org/10.1177/1471082X211007308

A newly emerging field in statistics is distributional regression, where not only the mean but each parameter of a parametric response distribution can be modeled using a set of predictors. As an extension of generalized additive models, distributional regression utilizes the known link functions (log, logit, etc.), model terms (fixed, random, spatial, smooth, etc.) and available types of distributions but allows us to go well beyond the exponential family and to model potentially all distributional parameters. Due to this increase in model flexibility, the interpretation of covariate effects on the shape of the conditional response distribution including its moments and other features derived from this distribution is more challenging than with traditional mean-based methods. In particular, such quantities of interest often do not directly equate to the modeled parameters but are rather a (potentially complex) combination of them. To ease the post-estimation model analysis, we propose a framework and subsequently feature an

implementation in `R` for the visualization of Bayesian and frequentist distributional regression models fitted using the bamlss (Umlauf, Klein, & Zeileis, 2018), gamlss (Stasinopoulos & Rigby, 2007) and betareg (Cribari-Neto & Zeileis, 2010) `R` packages.

The first author's contributions to this paper are as follows:

- Designing the graphical user interface (GUI)
- Writing code for the GUI and underlying computational methods
- Programming the `R` package including troubleshooting, bug fixing and error handling
- Writing the manuscript
- Revising the manuscript

Furthermore, Thomas Kneib contributed to the project by a) conceiving the project idea, b) designing the methodological section about distributional regression and c) thoroughly revising the manuscript. Sections 3.2, 3.5 and A.1.2 of this chapter, though modified, are based on sections of the master's thesis "bamlss.vis: An R package to Interactively Analyze and Visualize Bayesian Additive Models for Location, Scale and Shape (BAMLSS)", submitted by myself (Stanislaus Stadlmann) on 18th January 2017 at Georg-August Universität Göttingen for the degree "M.Sc. Applied Statistics". Some of the code written as part of this master's thesis was also re-used for `distreg.vis`.

Thesis inclusion

This paper is included as Chapter 3 in this thesis. While the paper content stays mostly the same, Section 3.4 was moved back into the main text body from the paper appendix, and an additional Section 3.6 was included. Section 3.5, also found in the paper appendix, replaced a shorter version of itself found in the paper, acting as a short summary. Further, some of the notation was changed to cohere with the rest of the chapters.

Parameter orthogonality transformations in distributional regression models

Stadlmann, S., Heller, G. Z., Kneib, T. & Koens, L. (2021). Parameter orthogonality transformations in distributional regression models. *Working Paper*.

Distributional regression has become an increasingly popular tool in regression analysis, providing the ability to link distributional parameters beyond the mean to additive predictors, for a practically unlimited choice of response distributions. For many of these response distributions, maximum likelihood estimates (MLEs) of the distributional parameters are correlated and, as a result, misspecifying the regression predictor for one distributional parameter induces a bias in the estimates for the predictors of other parameters, even if these are correctly specified. The aim of this article is to develop a framework for the reparametrization of any two-parameter distribution, such that its parameters have asymptotically uncorrelated MLEs i.e., are orthogonal. In some cases, this is achievable analytically; however more frequently the mathematics of the reparametrization proves intractable. Addressing this issue, we propose to retain the location parameter, and replace the second (usually scale) parameter of the chosen distribution with a function that provides a numeric solution to a system of ordinary differential equations (ODE) based on the covariance matrix of the MLEs, creating an orthogonal parametrization. This approach is validated in simulation studies where (a) parameters are assumed to be constant and (b) parameters depend on covariates. Finally, our proposed method is demonstrated on an extreme rainfall dataset from Tasmania, Australia.

The first author's contributions to this paper are as follows:

- Initial idea exploration
- Writing of project reports
- Execution and testing of research ideas
- Drafting and revising the manuscript

- Writing `R` code to implement the method
- Designing and producing graphs
- Formatting supplementary material

Furthermore, the following contributions were made: Thomas Kneib assisted with a) the conception of the project idea, b) designing the broad vision of the project when hurdles were presented and c) revising the manuscript. Gillian Heller a) also helped conceive the original project idea, b) followed up on my research progress, c) proposed important ideas when hurdles were encountered, d) helped revise the manuscript and e) helped revise the supplementary material. Lyndon Koens was crucial to a) solving mathematical challenges, and b) helping implement them in software as well as c) revising the manuscript.

Thesis inclusion

This paper is included in this thesis as Chapter 4. The differences between the thesis chapter and original working paper are Section 4.4.1 and parts of Section 4.4.3, which were originally in the published paper appendix but were moved into the content body.

Variable importance in likelihood-based regression models

Stadlmann, S. & Kneib, T. (2021). Variable importance in likelihood-based regression models. *Working Paper*.

Classical regression methods are popular for quantifying relations between dependent and independent variables and making statements about their significance, but not made for ranking explanatory variables by their relative importance. A viable relative importance metric should a) take into account variable cross-correlation, b) be independent of their order of inclusion, c) measure error-reduction conditional on other covariates and d) combine effects with multiple degrees of freedom. Within the field

of linear models, two metrics fulfill these requirements: "hierarchical partitioning" , in which the average contribution of each variable in all possible model subsets is calculated and "relative weights", a fast approximation of hierarchical partitioning based on orthogonal predictors. Beyond linear models, however, a sizeable gap of variable importance measures remains. To fill this vacuum, we propose an extension of both previously mentioned metrics to likelihood-based models with linear predictors including generalized linear models (GLM) and generalized additive models for location, scale and shape (GAMLSS) with linear predictors. We present an implementation using `R` called `vibe`. Our proposal's effectiveness is proven with an extensive simulation and presented with two analyses: health-care patient satisfaction scores in North Macedonia described using a latent ordinal GLM and childhood malnutrition in India modeled with a multi-parametric GAMLSS.

The first author's contributions to this paper are as follows:

- Drafting, revising and writing of manuscript
- Design, programming, development, maintaining and publishing of accompanying `R` package
- Creation of graphs and conception of application ideas
- Obtaining, cleaning and including the datasets used in application chapters

The following contributions were made: Thomas Kneib conceived the original project idea and helped overcome critical hurdles in the project. Further, Thomas Kneib helped with critical revisions.

Thesis inclusion

This paper is included in the thesis as Chapter 5, without any amendments.

Stanislaus Stadlmann, 31 July 2021

Acknowledgements

This rollercoaster of a journey that has been my PhD would not have been possible without the support of some extraordinary people. First and foremost I want to express my biggest thanks and deep gratitude to Thomas Kneib. Thomas, you were the first person who sparked some joy in me when it came to statistics teaching Linear Models back in 2014. To this day I have yet to meet a teacher who has a more down-to-earth, funnier and more approachable way of teaching statistics. When it came to my PhD, you always had an open door and ear when it came to dramatic hurdles that were encountered and believed in our joint projects even when I didn't. You created an office environment that I haven't found anywhere else, by inviting everyone for coffee breaks or feierabend-beers, sometimes even in Annes and my office when we were working. Of course I can also not forget for supporting me in coming to Australia - I wouldn't be here without you! Further, I want to say thanks to Gillian Heller, who made me feel welcome in Sydney, was always involved in my research in a detailed way and whose feedback on my writing I learned a lot from. I also feel very grateful for you always providing a reference when it was needed.

Thank you to all the office colleagues that have made my PhD life such an incredible time - especially Anne Berner, Hannes Riebl, Isa Marques, Paul Wiemann, Bruno Santos and Anne Sørensen. To my parents, Nicole Jaufer and Markus Stadlmann, who always believed in me even in my darkest moments. To all of the people who I was fortunate to share a house with and shaped who I am today, especially Annelene Kruse, Jonas Schmitt, Isabel Simon and Rafael Chaves. To the friends who made my time

in Göttingen and Sydney extraordinary, including: Timon Bunnenberg, Aditi Kuber, Paul Höft, Julius Hartmann, Anna Natlacen. To Maurizio Manuguerra and Samuel Muller, who provided mentorship in tricky situations. And to Kitty-Jean Laginha, whose multi-faceted support was invaluable and who continues to expand my horizon academically and non-academically. To the internet, the R programming language and LaTeX including BibTeX, which have saved me countless hours of manual labor.

List of Publications

- Stadlmann, S. & Kneib, T. (2021). Interactively visualizing distributional regression models with distreg.vis. *Statistical Modelling*, doi: https://doi.org/10.1177/1471082X211007308

- Stadlmann, S., Heller, G. Z., Kneib, T. & Koens, L. (2021). Parameter orthogonality transformations in distributional regression models. *Working Paper*.

- Stadlmann, S. & Kneib, T. (2021). Variable importance in likelihood-based regression models. *Working Paper*.

Contents

List of Notation

n	Number of samples
N	Number of simulation repetitions
Q	Number of available covariates in $\mathbf{X}$
Q_k	Number of available covariates in $\mathbf{X}_k$
i	Index of observations
θ_k	Distributional parameters
ω_k	Distributional parameters, orthogonalized
μ	First distributional parameter, equal to θ_1 (used in Chapter 3 to comply with software notation)
σ	Second distributional parameter, equal to θ_2 (used in Chapter 3)
K	Number of distributional parameters
y	Response, dependent variable
$\mathbf{X}$	Matrix of explanatory covariates
$\mathbf{X}_k$	Subset of $\mathbf{X}$ connected to parameter θ_k
x_q	Covariate, univariate
$\mathbf{x}_{kq}$	Covariate, possibly multivariate
$\boldsymbol{\beta}$	Vector of full regression coefficients
$\boldsymbol{\beta}_k$	Subset of $\boldsymbol{\beta}$ corresponding to $\mathbf{X}_k$
β_{k0}	Intercept term of parameter θ_k
β_{kq}	Regression coefficient of covariate x_q for modeling θ_k

$\boldsymbol{\beta}_{kq}$	Regression coefficient(s) of $\mathbf{x}_q$ for modeling θ_k
$f_{kq}(\cdot)$	Covariate effect of $\mathbf{x}_q$ for describing parameter θ_k, possibly nonparametric
$p(\cdot)$	Probability density function (PDF)
$g_k(\cdot)$	Link function to uphold support of θ_k
$h(\cdot)$	Response function $h(\cdot) = g^{-1}(\cdot)$
$s(\cdot)$	Orthogonality function, surface

List of Acronyms

BAMLSS Bayesian additive models for location, scale and shape

CDF cumulative distribution function

CG Cole and Green

DNN deep neural networks

FIZ Fitness- und Gesundheitszentrum

GAM generalized additive models

GAMLSS generalized additive models for location, scale and shape

GAMM generalized additive mixed models

GLM generalized linear models

GUI graphical user interface

HP hierarchical partitioning

IRLS iteratively reweighted least squares

LASSO least absolute shrinkage and selection operator

MADAM mean and dispersion additive models

MCMC Markov chain Monte Carlo

MLE maximum-likelihood estimate

ODE ordinary differential equation

PDE partial differential equation

PDF probability density function

PLS penalized weighted least squares

RS Rigby and Stasinopoulos

RW relative weights

STAR structured additive regression

VGAM vector generalized linear and additive models

WLS weighted least squares

List of Figures

List of Tables

Summary

Distributional regression represents a modern approach to regression modeling that yields the ability to simultaneously connect multiple parameters beyond the mean of any parametric response distribution to structured additive predictors that can take parametric and non-parametric forms. This thesis proposes contributions to this field in three unique ways: in 1) a framework for the visualization of distributional regression models is developed, which focuses on predicted conditional moments and the shape of the whole distribution, instead of solely relying on distributional parameters as is commonly done. It is implemented as an extensive interactive `R` package named `distreg.vis`, focused on usability. The second contribution 2) recognizes a bias in the estimation of distributional regression model coefficients of all parameters if the model equation of one parameter is incorrectly specified. A solution for two-parameter distributions based on a numerically solved system of ordinary differential equations (ODE) created with the parameters' maximum likelihood estimate (MLE) covariance matrix is outlined, implemented and tested in a simulation study. Contribution 3) fills a gap in the interpretation of fitted distributional regression models. Existing metrics for ranking the importance of variables in linear regression models are discussed, with "relative weights" and "hierarchical partitioning" standing out as the most suitable due to their robustness to the scale of covariates, the consideration of variable cross-correlation, order independency and suitability for effects with more than one degree of freedom. These metrics are subsequently extended to generalized linear models (GLM) and generalized additive models for location, scale and shape (GAMLSS) with linear

predictors taking into account the possibly multi-parametric response structure and likelihood-based nature of the fitted regression models. These extensions are implemented in an `R` package called `vibe`, providing methods compatible with several other packages. The above contributions are showcased using several datasets about wages in the Mid-Atlantic region of the USA, gym visitor numbers in Göttingen, extreme rainfall in Tasmania of Australia, patient satisfaction with a health care provider in North Macedonia and malnutrition scores in India.

1

Introduction

With the rise of personal computing and the increasing availability of digital devices, from mobile phones, cameras, smartwatches to electric toothbrushes, the amount of digital information available nowadays is ever-increasing. As stated by Arthur C. Clarke, however, "[...] it is vital to remember that information – in the sense of raw data – is not knowledge; that knowledge is not wisdom; and that wisdom is not foresight" (Gunawardene, 2003). This transfer of data into knowledge is the key role of a statistician, as outlined by Senn (2003).

Crucial tools in a statistician's toolkit are statistical regression models, which have seen a tremendous amount of research activity and innovation throughout the last half-century. From around 3,300 publications related to the field in 1970, almost 600,000 publications have been released to peer-reviewed journals in 2020, according to the database "Dimensions" (Hook et al., 2018).

One culmination of modeling innovation is distributional regression, which represents a natural extension of generalized linear models (GLM, Nelder & Wedderburn, 1972) and generalized additive models (GAM, Hastie & Tibshirani, 1990) in its ability to connect any parameter θ_k of a K-parametric distribution $y \sim D(\theta_1, \ldots, \theta_K)$ to a set of additive and possibly non-parametric predictors. Conceived as generalized additive models for location, scale and shape (GAMLSS, Rigby & Stasinopoulos, 2005), the term distributional regression was introduced by Klein, Kneib, Lang, and Sohn (2015) to also account for parameters beyond location, scale or shape. As visible in Figure 1.1, the number of publications related to distributional regression has increased rapidly since its original conception.

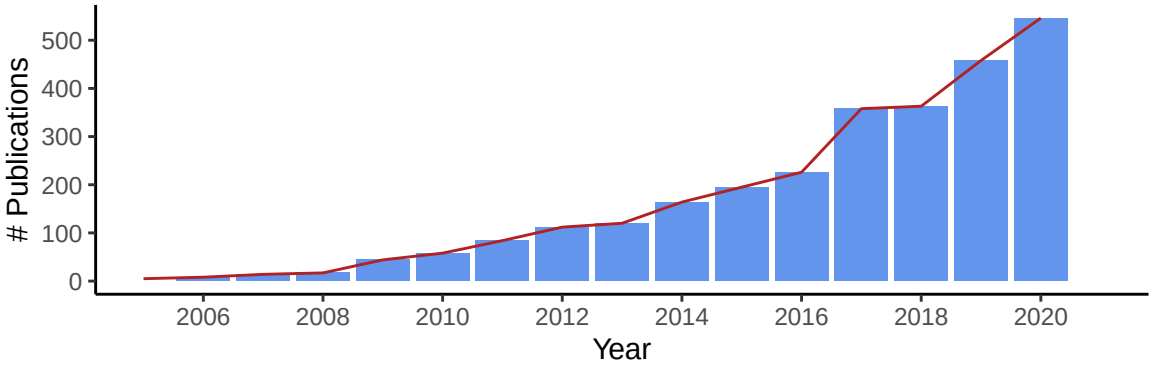

FIGURE 1.1: Number of scientific publications per year using search terms "distributional regression", "GAMLSS" or "BAMLSS" collected using portal "Dimensions" (Hook et al., 2018).

The emergence of distributional regression comes with new-found flexibility in statistical modeling, but also adds a new layer of complexity that generates its own set of complications, with gaps that remain to be filled. Three of those, related to the visualization, unbiased estimation and interpretation of distributional regression models, are discussed and resolved in this thesis.

The first project of this thesis concerns the graphical display of distributional regression models. In contrast to linear models, GLM and GAM, for which numerous visualization tools exist, the visualization of distributional regression models is more difficult since covariate effects might have an influence on multiple distributional parameters and using a non-identity link function makes the interpretation depend on

other covariates. Further, modeled distributional parameters do not always equate to moments, so a transformation is necessary to arrive at interpretable figures. To resolve these challenges, in Chapter 3 we develop a framework which is implemented using the statistical software language `R` called `distreg.vis` (Stadlmann & Kneib, 2021).

In the second project, the main focus is shifted towards the estimation of distributional regression models. We recognize that most distributions outside the range of the exponential family have a non-diagonal covariance matrix of the distributional parameter maximum-likelihood estimates (MLEs). While not problematic when assumed that all parameters are constant, expressing θ_k as a function of additive covariate functions $f_{kq}(\mathbf{X}_{kq}; \boldsymbol{\beta}_{kq})$ leads to a bias of regression coefficients $\boldsymbol{\beta}_{kq}$ if some parameters θ_k are not correctly specified. The misspecification bias as found in linear models (Zimmerman, 2020) gains another layer in distributional regression with correlated parameters, since regression coefficients of one parameter model specification are found to be biased even if the misspecification occurs in another parameter. As shown in Chapter 4, we propose a solution for two-parameter distributions by reparametrizing the probability density function (PDF) based on a numeric solution of a system of ordinary differential equations (ODEs) constructed using the MLE covariance matrix.

The last contribution of this thesis is motivated by a previous lack of tools to rank the relative importance of covariate effects in a fitted distributional regression model. While components of a traditional regression output including regression coefficients, p values and goodness-of-fit figures are useful to make statements about the model fit, effect significances and (possibly nonlinear) relationships between $\mathbf{X}$ and y, they cannot be used to rank variable importance, since the covariate cross-correlation is not taken into account, effects might have multiple coefficients or they are dependent on the covariates' scale. In Chapter 5, we review existing variable importance metrics for linear models, including their strengths and weaknesses, and propose an extension of these to GLM and GAMLSS. This work is implemented in the `R` package `vibe`.

The tools and frameworks developed in this thesis concerning the visualization, estimation and interpretation of fitted regression models fill important gaps in distributional regression analysis, with which researchers will be better equipped to tackle

their inquiries. While the proposal of Chapter 4 aims to ensure that regression coefficients are unbiased, Chapters 3 and 5 are targeted at making statistical models more accessible, thereby reducing the obstacles presented to applied researchers with the increased complexity of distributional regression.

The remainder of this thesis will be structured as follows: Chapter 2 gives an introduction to distributional regression models, including a brief history and mathematical foundation as well as current developments. Chapters 3 to 5 represent the main contributions of this thesis, concerning visualization (3), unbiased estimation (4) and interpretation (5) of distributional regression models. The thesis concludes and gives an outlook in Chapter 6, while additional material about some of the chapters is found in Appendix A.

2

Distributional regression models

2.1 Motivation

In introductory statistics courses, students learn that the realizations of many phenomenona in the outside world can follow a variety of different distributions, which depend on parameters that can change depending on the exact scenario described. However, many popular regression methods only measure the change in one parameter, like the expected value of a distribution, and disregard others like the scale or shape.

Figure 2.1 shows a scenario where kernel density estimates of yearly wages in the Mid-Atlantic region on the east coast of the US are displayed by individuals' education level (James, Witten, Hastie, & Tibshirani, 2017). The distributions are noticeably different: expected wages as well as wage variation increases with a higher education

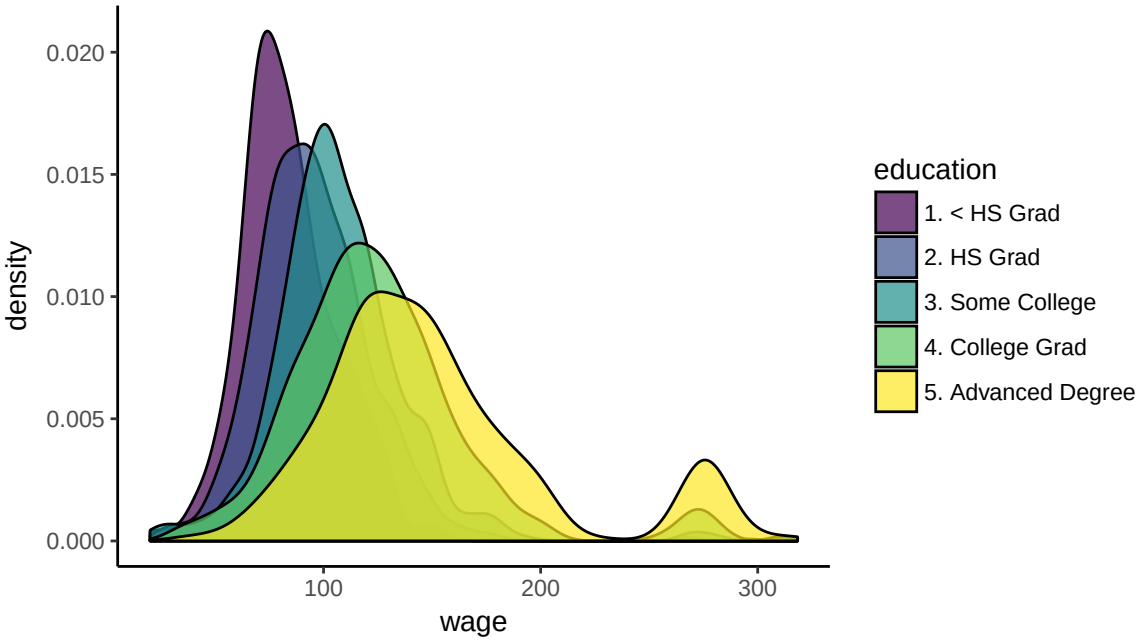

FIGURE 2.1: Kernel density estimates of yearly wages (in 1000$) of workers in the Mid-Atlantic region of the United States, divided and colored by their education level.

level.

In a classical regression model, the expected value of an individual's wage would be connected to the education level, with the intercept varying depending on the level. This leads to a shift in location of the target distribution, correctly picking up the difference in average wages. However, classical regression approaches such as linear models or generalized linear models (GLM, Nelder & Wedderburn, 1972) assume that the dispersion parameter of the distribution is homogeneous, an assumption which is very likely violated in this case. This leads to incorrect hypothesis testing and confidence intervals, as outlined by Fahrmeir, Kneib, Lang, and Marx (2013, p. 80). Models which can take into account variance differences depending on covariates are therefore necessary to ensure correct inference.

2.2 Conception

Initial pursuits to connect parameters beyond the mean to explanatory covariates were undertaken as early as 1987 with Murray Aitkin's contribution "Modelling variance

heterogeneity in normal regression using GLIM" (Aitkin, 1987), in which it was recognized that in classical linear models the dispersion parameter of the dependent variable is assumed to be homogeneous with respect to explanatory covariates, which is often not the case. This is resolved by connecting the dispersion parameter to a user-specified subset of predictors using a log-link function, and estimating it simultaneously with the expected value. Aitkin published code to estimate such a model with normally distributed errors using the statistical software GLIM (Nelder, 1975).

Taking it one step further, Smyth (1989) proposes a generalization of Aitkin's approach to allow the target distribution to be of the exponential family, rather than simply Gaussian. Branded "Generalized Linear Models with Varying Dispersion", this model class utilizes the GLM model specification for $\mathbb{E}(y)$ including its link function but assumes the dispersion parameter ϕ to have its own connection to a subset of additive predictors, including a link function specified independently of $\mathbb{E}(y)$.

Further down this road to distributional regression, Rigby and Stasinopoulos (1996) are concerned with effect types. Motivated by the flexibility of generalized additive models (GAM, Hastie & Tibshirani, 1990), they propose replacing the linear predictors used to describe both the mean and dispersion parameters μ, ϕ with additive and possibly non-parametric functions of the explanatory covariates. Employing the same user-specified link functions as Smyth (1989), their contribution is coined mean and dispersion additive models (MADAM).

Later, Rigby and Stasinopoulos further developed MADAM into its current form called generalized additive models for location, scale and shape (GAMLSS, Rigby & Stasinopoulos, 2005). In this seminal paper, the constraint of using exponential family distributions in modeling the target variable is removed to allow any distributional choice, as long as the probability density function (PDF) is twice continuously differentiable with respect to its parameters. Further, the parameters which can be connected to explanatory variables are no longer limited to expected value and variance; they can be any distributional parameter including kurtosis, skewness and shape. Model estimation is conducted using penalized likelihood maximization.

Finally, distributional regression was coined as an umbrella term by Klein, Kneib,

Lang, and Sohn (2015), encompassing the previously mentioned model classes. Furthermore, it reflects the fact that not all distributional parameters may represent the location, scale or shape of a distribution. Some models in which the predictors assume special non-parametric forms models such as spatial or random effects are also referred to as "structured additive distributional regression". In those cases, estimation is usually conducted using Bayesian methods (see e.g. Klein & Kneib, 2016a; Klein, Kneib, Lang, & Sohn, 2015; Voncken, Kneib, Albers, Umlauf, & Timmerman, 2021).

2.3 Model specification and implementations

To formally introduce the class of distributional regression models, we assume a K-parametric distribution $y \sim D(\theta_1, \ldots, \theta_K)$. Each of the parameters θ_k can be connected to a set of predictors $\mathbf{X}_k$ representing a subset of the full matrix of available covariates $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_Q)$. Due to the "curse of dimensionality" (Fahrmeir et al., 2013, p. 531), we assume an additive structure of covariate effects. This leads to the following model specification:

$$\begin{aligned} \eta_k &= \beta_{k0} + \sum_{q=1}^{Q_k} f_{kq}(\mathbf{x}_{kq}; \boldsymbol{\beta}_{kq}) \\ h_k(\eta_k) &= \theta_k, \end{aligned} \tag{2.1}$$

where $k = 1, \ldots, K$, $q = 1, \ldots, Q_k$ and $Q_k \leqslant Q$ with Q_k, Q representing the number of available covariates in $\mathbf{X}_k$ and $\mathbf{X}$ respectively. Explanatory covariates x_q used to model θ_k are represented in a possibly multivariate manner by $\mathbf{x}_{kq}$. Further, $h_k(\cdot)$ is a pre-specified response function intended to uphold the support of parameter θ_k, and $\boldsymbol{\beta}_{kq}$ are regression coefficients of structured effect f_{kq}. These structured effects can have parametric (linear, categorical, interactions) or non-parametric (spatial, random, smooth) as well as multivariate (tensor product splines) forms. This flexible model equation means that distributional regression encompasses many known model classes, including GLM, GAM, GAMLSS and generalized additive mixed models (GAMM, Lin & Zhang, 1999).

2.3.1 gamlss

Software packages for fitting distributional regression models are manifold. Acting as the companion to Rigby and Stasinopoulos (2005), the `R` (R Core Team, 2020) package `gamlss` (Stasinopoulos & Rigby, 2007) presents a comprehensive suite of programs to estimate distributional regression models based on maximizing the likelihood function with respect to the response variable and the covariate effects.

The first of `gamlss`' two presented algorithms for model estimation called Rigby and Stasinopoulos (RS) combines backfitted iteratively reweighted least squares (IRLS) with a process that iterates over the modeled parameters. This algorithm is divided into two parts: the inner iterations (within parameter θ_k) and outer iterations (between parameters $\theta_k, \theta_{k+1}, \ldots$). In the inner iterations, a working variable $\mathbf{z}_k$ is created, comprised of the predictor vector $\boldsymbol{\eta}_k = g_k(\boldsymbol{\theta}_k)$, the score function $\mathbf{u}_k = \frac{\partial \ell}{\partial \boldsymbol{\eta}_k}$ with respect to the predictor and weights $\mathbf{w}_k$ which include different forms of the second derivative of the log-likelihood function (e.g. observed or expected Fisher information), depending on what is available for the specific distribution[1]. Then, the coefficients $\boldsymbol{\beta}_k$ are updated by backfitting the covariate effects $f_k(\cdot)$ to working variable $\mathbf{z}_k$ using weights $\mathbf{w}_k$ with weighted least squares (WLS) for linear effects or penalized weighted least squares (PLS) for non-parametric terms, until the inner iterations converge. Then, the outer iterations repeat this step for every parameter θ_k until its deviance also converges (Stasinopoulos, Rigby, Heller, Voudouris, & De Bastiani, 2017).

In the second algorithm implemented by `gamlss` labelled Cole and Green (CG), each (expected or approximate) cross-derivative of the log-likelihood function with respect to the parameters is required to calculate iterative weights $\mathbf{z}_k$ in each update of the regression coefficients. Further, in contrast to RS, the fitting of covariate effects to working variable $\mathbf{z}_k$ using weights $\mathbf{w}_k$ is not repeated until converging within one parameter θ_k, but also represents an outer iteration. Updates of regression coefficients are however again calculated using a modified backfitting algorithm. This contrast to RS makes the algorithm faster in some cases, but is also generally more unstable. For

[1] The bold-letter notation for $\boldsymbol{\theta}$ taken from (Rigby & Stasinopoulos, 2005) refers to a vector of observations, not parameters. The rest of this thesis refers to $\boldsymbol{\theta}$ as a vector of parameters $(\theta_1, \ldots, \theta_K)^\intercal$.

this reason, RS is preferred and set as default by the developers (Stasinopoulos et al., 2017).

2.3.2 bamlss

An extensive alternative to `gamlss` is given by the `R` package `bamlss`, implementing Bayesian additive models for location, scale and shape (BAMLSS, Umlauf et al., 2018). With many built-in distributions including an interface to the `gamlss` distributions in the package `gamlss.dist` (Stasinopoulos & Rigby, 2019) and a wide variety of available effect types, it offers similar flexibility to `gamlss`. As the name suggests, model estimation is conducted using Bayesian methods: coefficients are assumed to follow a prior distribution, rather than being fixed. Combined with the likelihood function with respect to the regression coefficients, response variable and explanatory covariates, this creates the log-posterior density $\log \pi$:

$$\log \pi(\boldsymbol{\beta}, \boldsymbol{\tau} \mid \mathbf{y}, \mathbf{X}, \boldsymbol{\alpha}) \propto \underbrace{l(\boldsymbol{\beta} \mid \mathbf{y}, \mathbf{X})}_{\text{log-likelihood}} + \underbrace{\sum_{k=1}^{K} \sum_{q=1}^{Q_k} \log p_{qk}\left(\boldsymbol{\beta}_{qk} \mid \boldsymbol{\tau}_{qk}, \boldsymbol{\alpha}_{qk}\right)}_{\text{prior distributions}}, \tag{2.2}$$

where $\boldsymbol{\beta}_{qk}$ are regression coefficients for the qth covariate effect on distributional parameter θ_k and therefore are a subset of $\boldsymbol{\beta}$, and $\mathbf{y}, \mathbf{X}$ are response and design matrices of explanatory variables respectively. Vectors $\boldsymbol{\tau}_{qk}$ and $\boldsymbol{\alpha}_{qk}$ are both parameters of the prior distributions $p_{qk}(\cdot)$; $\boldsymbol{\tau}$ combining all $\boldsymbol{\tau}_{qk}$ represents hyper-parameters having their own distribution $d_{\boldsymbol{\tau}_{qk}}(\cdot)$, while $\boldsymbol{\alpha}$ consisting of $\boldsymbol{\alpha}_{qk}$ are fixed. Hyper-parameters $\boldsymbol{\tau}_{jl}$ are usually variances, which can be used to control the degree of smoothness of $f_{jl}(\cdot)$ or the amount of correlation between observations (Umlauf et al., 2018).

When estimating regression coefficients in `bamlss`, the posterior mode is first maximized with respect to the coefficients $\boldsymbol{\beta}$ of interest. The values found after optimization are then used as starting values for Markov chain Monte Carlo (MCMC) simulation, which results in a matrix with samples from the posterior distribution of $\boldsymbol{\beta}$. While the two-stage approach of `bamlss` is computationally more expensive than maximum-likelihood estimation, the usage of MCMC samples and subsequent ease of constructing

credible intervals yields more reliable uncertainty measures than asymptotic confidence intervals.

2.3.3 VGAM

Vector generalized linear and additive models (VGAM, Yee, 2015) and the corresponding `R` package `VGAM` (Yee, 2010) represent an implementation of distributional regression models that has been available since the early days of `R` programming (Yee & Hastie, 2003). While originally focused on multivariate responses, it has since grown to include a wide range of modeling approaches, including ordinal regression models, copulas combining distributions of different types and distributions outside of the exponential family. A wide range of smoothing functions exist to connect any distributional parameter to explanatory covariates. Estimation is conducted with IRLS.

2.3.4 Other implementations

A list of implementations of distributional regression would not be complete without `mgcv` (Wood, 2011). Originating as a software to fit GAMs, it has evolved to allow for multi-parameter equations, albeit only for a select number of distributions mostly within the exponential family. `BayesX` (Brezger, Kneib, & Lang, 2005) implements Bayesian structured additive regression (STAR) models including multi-parameter effects, and provides an `R` interface named `R2BayesX` (Umlauf, Adler, Kneib, Lang, & Zeileis, 2015). Its strengths lie in its customizability and special effects, e.g. spatial terms. The package `betareg` (Cribari-Neto & Zeileis, 2010) provides multi-parameter additive regression tailored to the beta distribution. In `gamboostLSS` (Thomas et al., 2018) a regularized estimation in the form of gradient boosting is combined with GAMLSS. The result is a package well suited for high-dimensional predictor scenarios. In `GJRM` (Marra & Radice, 2017) the specification of bi- and trivariate response distributions is possible, without the need to have the same marginal distributions. All parameters of the multivariate response distribution can then be connected to additive predictors.

2.4 Current developments

Distributional regression remains a method with an active research community and numerous ongoing developments. While `gamlss` already implements a large amount of distributions, it is limited to a univariate response. Klein, Kneib, Klasen, and Lang (2015) propose a Bayesian multivariate distributional regression framework, which supports continuous, discrete and latent responses as well as non-parametric predictors.

Many other projects are centered around model estimation and effect selection: Thomas et al. (2018) improve the existing `gamboostLSS` estimation technique with a gradient-boosting algorithm that incorporates "stability selection", a method that tests the stability of covariates by repeatedly subsampling the data. An alternative effect selection for distributional regression building on the least absolute shrinkage and selection operator (LASSO, Tishbirani, 1996) is proposed for linear and categorical covariates in Groll, Hambuckers, Kneib, and Umlauf (2019). Tackling a similar issue from a Bayesian perspective, Klein, Carlan, Kneib, Lang, and Wagner (2021) introduce a Spike and Slab prior specification into the distributional regression model class.

Since effect types in distributional regression can be non-parametric, the available options are ever-increasing. In Klein, Nott, and Smith (2021), deep neural networks (DNN) are fused with distributional regression to cater for audiences with high-dimensional predictor spaces desiring high prediction accuracies. An implementation is showcased in Rügamer et al. (2021). Schlosser, Hothorn, Stauffer, and Zeileis (2019) propose a framework and implementation called `disttree` to connect decision trees to distributional parameters.

3

Interactively visualizing distributional regression models

3.1 Introduction

For modeling parameters beyond the mean of a target distribution, generalized additive models for location, scale and shape (GAMLSS, Rigby & Stasinopoulos, 2005) as introduced by Rigby and Stasinopoulos (2005) provide the ability to link all parameters characterizing the response distribution to a set of explanatory variables via an additive predictor, similar in spirit to generalized additive models but without its distributional limitations. Overcoming some of GAMLSS' earlier restrictions, distributional regression as coined by Klein, Kneib, Lang, and Sohn (2015) presents a highly flexible modeling framework with a variety of possible target distributions and a wide

range of effects including parametric penalized splines, random and spatial effects as well as non-parametric effects such as regression trees.

Implementations of distributional regression models feature `gamlss` (Stasinopoulos & Rigby, 2007) and `bamlss` (Umlauf et al., 2018), the most prominent extensions with a vast selection of available distributions and effects. A number of software extensions have also been developed to support specific instances of the distributional regression class, such as `betareg` (Cribari-Neto & Zeileis, 2010) for beta regression. While `gamlss` and `betareg` employ different types of maximum likelihood estimation, `bamlss` implements a Bayesian approach featuring posterior mode estimates which are subsequently used as starting values for Markov chain Monte Carlo (MCMC) sampling. This has the added benefit of being able to construct the posterior distribution of each parameter as well as potentially complex functions of these parameters.

Moving beyond single-parameter regression models naturally leads to additional challenges when it comes to the interpretation of the estimated effects, since the same covariate may show up in multiple regression predictors and applying a non-identity link function additionally makes the interpretation depend on the remaining set of covariates. Furthermore, in many cases the parameters employed to characterize the distribution of interest do not directly equate to the moments or other interpretable characteristics of a distribution, making another transformation necessary to arrive at interpretable figures.

As a consequence, applied researchers often appreciate the practical appeal of distributional regression where regression relations beyond the mean can be investigated, but they struggle when it comes to understanding the output of the regression estimates. To facilitate this process, we introduce a new package `distreg.vis` that can deal with model classes from the `bamlss`, `gamlss` and `betareg` packages to achieve the following tasks:

- `moments()`: Obtain predicted moments (expected value, variance) of the target distribution based on user-specified values of the explanatory variables, if they exist.

- `plot_dist()`: Create a graph displaying the predicted probability density function (PDF) or cumulative distribution function (CDF) based on the same user-specified values.

- `plot_moments()`: View the marginal influence of a selected effect on the predicted moments of the target distribution.

With those functions, applied scientists can directly translate regression objects into publication-ready graphs without the need to worry about package-specific predict and plotting functions, the connection between predicted parameters and moments or the correct display of predicted PDFs. To make the process of interpreting fitted distributional regression models even more accessible, `distreg.vis` features a rich graphical user interface (GUI) built on the `shiny` framework (Chang, Cheng, Allaire, Xie, & McPherson, 2018). Using this GUI, the user can (a) obtain an overview of the selected model fit and then use the functions mentioned above to (b) easily select explanatory values for which to display the predicted distributions, (c) obtain marginal influences of selected covariates and (d) change aesthetic components of each displayed graph. Following a successful analysis, the user can obtain the R code needed to reproduce all displayed plots without having to start the application again.

The idea of calculating conditional values of the response distribution by way of `plot_moments()` is not new. In fact, many other R packages feature aspects of the functionality of `distreg.vis`. Notably, the `effects` package (Fox & Weisberg, 2019) is an extensive library for calculating conditional means of the response distribution depending on varying explanatory covariates in linear, generalized linear and mixed effects-type models. Lacking the graphing capabilites of `effects` while putting more focus on the estimation of marginal means, the `prediction` (Leeper, 2019) package offers a bigger range of supported model classes and even non-parametric effects.

Certainly the most general marginal effects packages are `emmeans` (R. Lenth, 2019), formerly called `lmmeans` (R. V. Lenth, 2016), and `ggeffects` (Lüdecke, 2018), as they not only offer a large variety of supported model classes and predictor effects but, contrary to `effects` and `prediction`, also include the compatibility to all of `distreg.vis`'

supported distributional regression model classes: `gamlss`, `betareg` and `bamlss` (only supported by `ggeffects`). However, even the latter two effects packages are only able to compute marginal means of moment values for `betareg` and fail to provide the same functionality for `bamlss` and `gamlss`, where predictions remain restricted to the parameter-level. As such, `distreg.vis` is the only package which supports both `gamlss`, `bamlss` and `betareg` in full generality and can calculate marginal effects on the moments and not only the parameters of a distribution.

The package `distreg.vis` is available on the Comprehensive R Archive Network (CRAN) at https://cran.r-project.org/web/packages/distreg.vis/index.html.

The remainder of this paper is structured as follows: Section 3.2 provides an example tutorial on analysing income distributions in which using `distreg.vis` yields a benefit. Section 3.3 covers the methodological background of the distributional regression model class and its implementations, while Section 3.4 details how `distreg.vis` itself is implemented in `R`. The GUI is showcased in Section 3.5, while Section 3.6 features an application using hourly visitor data of a gym in Göttingen. Section 3.7 concludes the paper. The appendices show how `distreg.vis` can be extended (Section A.1.1), display special cases of `distreg.vis`' functions (Section A.1.2) and provide complementary graphs (Section A.1.3). The datasets, `R` scripts as well as the appendices are available in the online supplemental materials at www.statmod.org/smij/archive.html.

3.2 Motivating example: Drivers of yearly income

To illustrate the usefulness of `distreg.vis`, we will give a short example based on a dataset on the yearly income of 3,000 male workers in the Mid-Atlantic region of the United States, provided by the `ISLR` package (James et al., 2017). This dataset will then be used continuously throughout Section 3.2 and 3.5 as well as parts of the Appendix to illustrate `distreg.vis`' abilities.

The dataset consists of 11 variables, both continuous and categorical. Our variable of main interest, i.e. the response in the following regression analyses, is given by individuals' raw yearly income in thousand US$ (variable `wage`). The remaining variables

mostly contain socioeconomic information, including the person's age in years (`age`), their ethnic origin (`ethnicity`), their level of education (`education`) and the year in which the observation was recorded (`year`). Considering its non-negative nature, we start by assuming a log-normal distribution for the wages, $y \sim \text{Lognormal}(\mu, \sigma^2)$. Though more complex distributions would provide a slightly better fit than the log-normal, it was chosen for its simplicity and popularity.

To find out what drives a person's income, we will link both parameters of the log-normal distribution to the aforementioned explanatory variables. The model specification therefore has the following form:[1]

$$\begin{aligned} \mu &= \beta_{10} + f_{11}(\texttt{age}) + \beta_{12} \cdot \texttt{year} + f_{13}(\texttt{education}) + f_{14}(\texttt{ethnicity}) \\ \log(\sigma) &= \beta_{20} + f_{21}(\texttt{age}) + \beta_{22} \cdot \texttt{year} + f_{23}(\texttt{education}) + f_{24}(\texttt{ethnicity}) \end{aligned} \tag{3.1}$$

where the functions f_{11}, f_{21} are modeled non-parametrically using penalized splines (Eilers & Marx, 1996), while f_{13}, f_{23}, f_{14} and f_{24} are modeled as simple fixed categorical effects of the respective variables. All effects with more than one degree of freedom are therefore consistently denoted with $f_{kq}(\cdot)$ to emphasize that the original input variable has to be recoded prior to inclusion in the model. The standard deviation parameter σ is connected to the explanatory variables using a log-link function to ensure a positive support.

Combined with the sample-based estimation technique of `bamlss`, `distreg.vis` can produce credible intervals around marginal influence plots. For this reason, we choose `bamlss` for model estimation, using the following code:

```
R> wage_model <- bamlss(
+    list(
+      wage ~ s(age) + ethnicity + year + education,
+      sigma ~ s(age) + ethnicity + year + education
+    ),
+    data = Wage,
```

[1] As mentioned in the List of Notation, Chapter 3 mostly uses μ and σ for the first and second parameter of a distribution, respectively, whereas the rest of the thesis uses θ_1 and θ_2 (or higher indices). This is due to compatibility with supported software packages which use the μ, σ notation.

```
+    family = lognormal_bamlss()
+  )
```

After successful estimation, it makes sense to have a look at the model summary. Using the built-in `summary.bamlss()` function, we can take a look at the results:

```
R> print(summary(wage_model), digits = 1)

[output shortened]
Formula mu:
---
wage ~ s(age) + ethnicity + year + education
-
Parametric coefficients:

                             Mean   2.5%    50%  97.5% parameters
(Intercept)                36.834 14.075 36.606 61.579       38.0
ethnicity2. Black           0.008 -0.077  0.007  0.101        0.0
[output shortened]
-
Smooth terms:
             Mean 2.5%  50% 97.5% parameters
s(age).tau21 0.33 0.06 0.23  1.21        0.3
s(age).alpha 1.00 1.00 1.00  1.00         NA
s(age).edf   5.99 4.52 5.96  7.63        6.4
---
Formula sigma:
---
sigma ~ s(age) + ethnicity + year + education
-
Parametric coefficients:
[output shortened]
-
Smooth terms:
             Mean 2.5% 50% 97.5% parameters
s(age).tau21  2.5  0.5 2.0   7.7          2
s(age).alpha  0.9  0.4 1.0   1.0         NA
```

```
s(age).edf    6.3  4.7 6.3   7.6          6
---
[output shortened]
```

As visible in the code output, `bamlss` provides estimation results for each effect used to describe the target distribution parameters (μ and σ in this case). Even though making statements about each effect's significant difference from zero is possible, the interpretation of such statements as well as for the raw effect estimates is difficult. Model results of penalized splines, for example, only show estimates for their degree of smoothness (τ and α) and estimated degrees of freedom. Without appropriate graphs showing the marginal effect they can therefore not be interpreted. Furthermore, even the absolute coefficients for the categorical effects cannot be taken at face value, since they only affect the distributional parameters μ and σ, and not the moments.

To overcome these limitations, we use `distreg.vis`. If, for example, we are interested in the impact of `education` on the marginal income distribution, we create a `data.frame` object in which all different `education` categories are present and all other numeric variables are set to their mean. This can be easily achieved by the function `set_mean()` in combination with `model_data`, which obtains the explanatory covariates of the model and sets them to the mean. Further defining the `row.names` of the `data.frame` to be the different education levels ensures improved legends in further graphs.

```
R> df <- set_mean(
+    model_data(wage_model),
+    vary_by = "education"
+  )
R> row.names(df) <- levels(Wage$education)
R> df

                   ethnicity year           education age
1. < HS Grad        1. White 2006       1. < HS Grad  42
2. HS Grad          1. White 2006         2. HS Grad  42
3. Some College     1. White 2006    3. Some College  42
4. College Grad     1. White 2006    4. College Grad  42
```

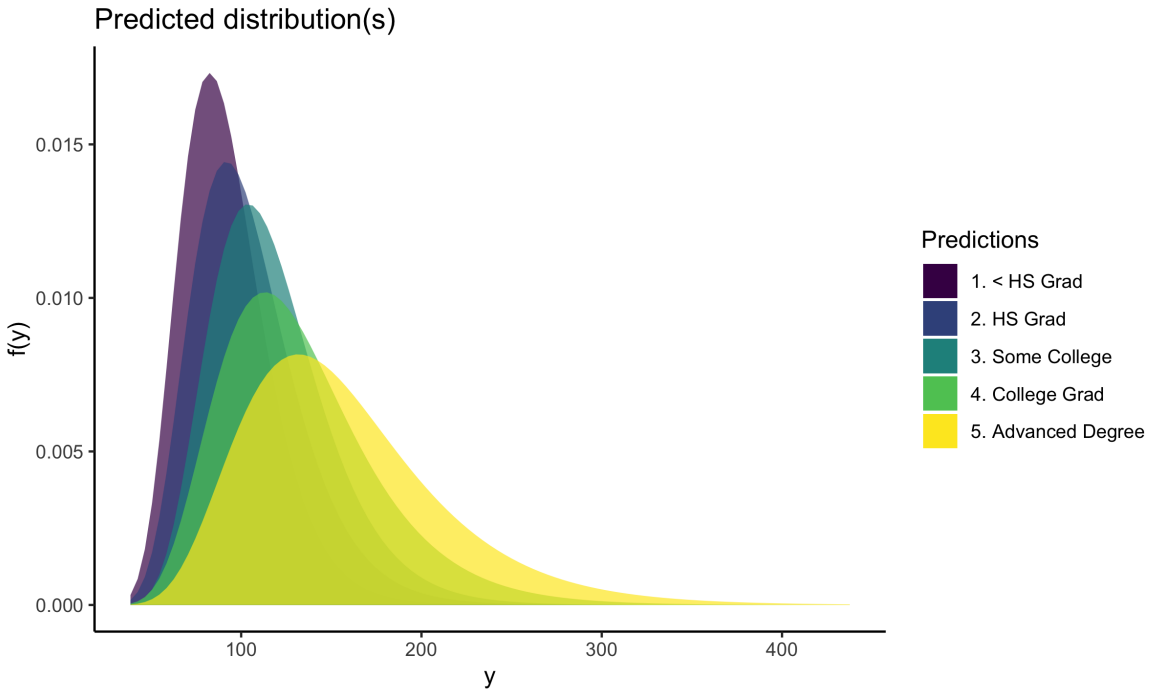

FIGURE 3.1: Predicted distributions for each covariate combination specified in the `df` data.frame object.

```
5. Advanced Degree  1. White 2006 5. Advanced Degree  42
```

Now, it will be interesting to see the predicted distribution for all five cases. To achieve this, we first use `distreg.vis`' `preds()` function and obtain the predicted distributional parameters. Then, we include them into `plot_dist()` to plot the complete distributions:

```
R> pp <- preds(model = wage_model, newdata = df)
R> pp
                          mu     sigma
1. < HS Grad        4.485241 0.2691822
2. HS Grad          4.603004 0.2886467
3. Some College     4.722890 0.2826761
4. College Grad     4.836627 0.3278559
5. Advanced Degree 5.002570 0.3490710

R> plot_dist(wage_model, pred_params = pp)
```

Figure 3.1 displays the predicted distributions for each covariate combination specified in `df`. We can see that the predicted income not only changes in location (higher education shifts the distribution to the right), but also in the variance (higher education leads to a higher variance).

Figure 3.1 is useful to get a visual feel of the influence of `education` on the modeled distribution. However, we would also like to know the influence of `age`, which is not a categorical covariate, on the predicted moments of the log-normal distribution. Normally, this would be a tedious task, as the modeled non-parametric effect has to be transformed two times: first, via the link function to ensure the correct support of the modeled parameters. And second, from the parameters to the distributional moments, as the parameters of the log-normal distribution do not directly equate to its moments.

The function `plot_moments()` was written to solve this task. It takes both a fitted distributional regression object and combinations of explanatory variables for which the influence is of interest. In our case, we specify both the `df` and `wage_model` objects as arguments of the function:

```
R> plot_moments(
+    wage_model,
+    int_var = "age",
+    pred_data = df,
+    rug = TRUE,
+    samples = TRUE,
+    palette = "viridis",
+    uncertainty = TRUE
+  )
```

Executing the above code results in what can be seen in Figure 3.2. The plot is divided into two parts representing the first two moments of the target distribution labelled “Expected_Value” and “Variance”. On both graphs, the y-axis depicts the moment values, while x-axis displays the variable of interest, which is `age` in our case.

The lines seen in each graph represent the previously specified covariate combinations, and then display how the moment changes over the whole range of the variable of interest. In our case the five different lines represent different education levels. We

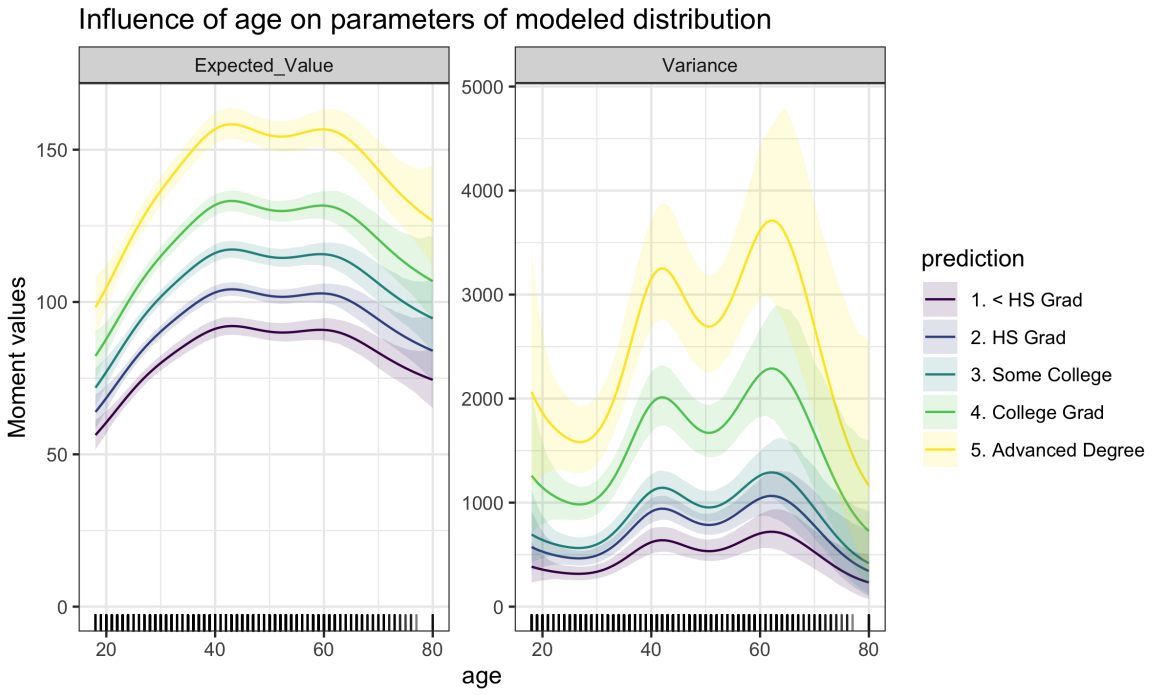

FIGURE 3.2: Impact of variable `age` on the first two predicted moments of the target wage distribution, including its 95% credible intervals.

can now see that the expected `wage` level first rises with age until around the age of 40, when it is lowered a bit. Then the income levels increase again up until the age of 60, at which point the `wage` then decreases. We can also see that this shape roughly stays the same for each education level, which stems from the lack of a modeled interaction effect between `age` and `education`.

A strong advantage of the sample-based approach of `bamlss` is its ability to easily construct credible intervals around the parameter estimates. In Figure 3.2 we can see small shadows representing credible intervals above and below each of the five lines describing the `age` effect on the first two moments in each `education` category. Since the first and highest `education` categories do not overlap, we can conclude that their effect is significantly different from each other, just from observing the graph.

Looking at the right part of Figure 3.2, we can see that the modeled variance levels of the target distribution show similar linkage to `age` as the expected value: two high points can be observed around the age of 40 and 60, both at which the modeled distribution reaches the highest `wage` variance.

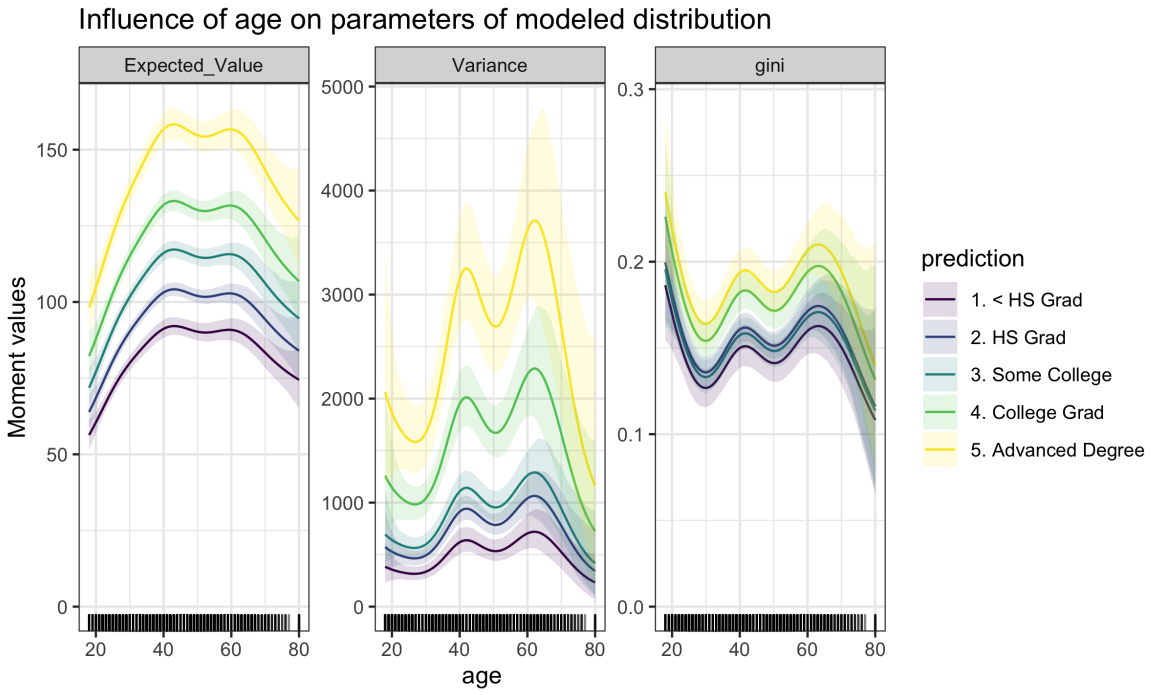

Figure 3.3: Impact of variable `age` on the first two predicted moments of the target wage distribution (equivalent to Figure 3.2) as well as a user-specified function (`gini()`), including its credible intervals.

Even though Figure 3.2 only shows the influence of `age` on the first two moments, `plot_moments()` is easily able to include other metrics that depend on the predicted parameters, using its argument `ex_fun`. A good example in wage distributions would be the Gini coefficient (Lerman & Yitzhaki, 1984), an economic figure measuring a distributions' inequality. Including this measure in our effect plots can be done using the following code:

```
R> gini <- function(par) {
+    2 * pnorm((par[["sigma"]] / 2) * sqrt(2)) - 1
+  }
R> moments_plot_exfun <- plot_moments(
+    wage_model, int_var = "age",
+    pred_data = df, samples = TRUE,
+    uncertainty = TRUE, ex_fun = "gini",
+    rug = TRUE
+  )
```

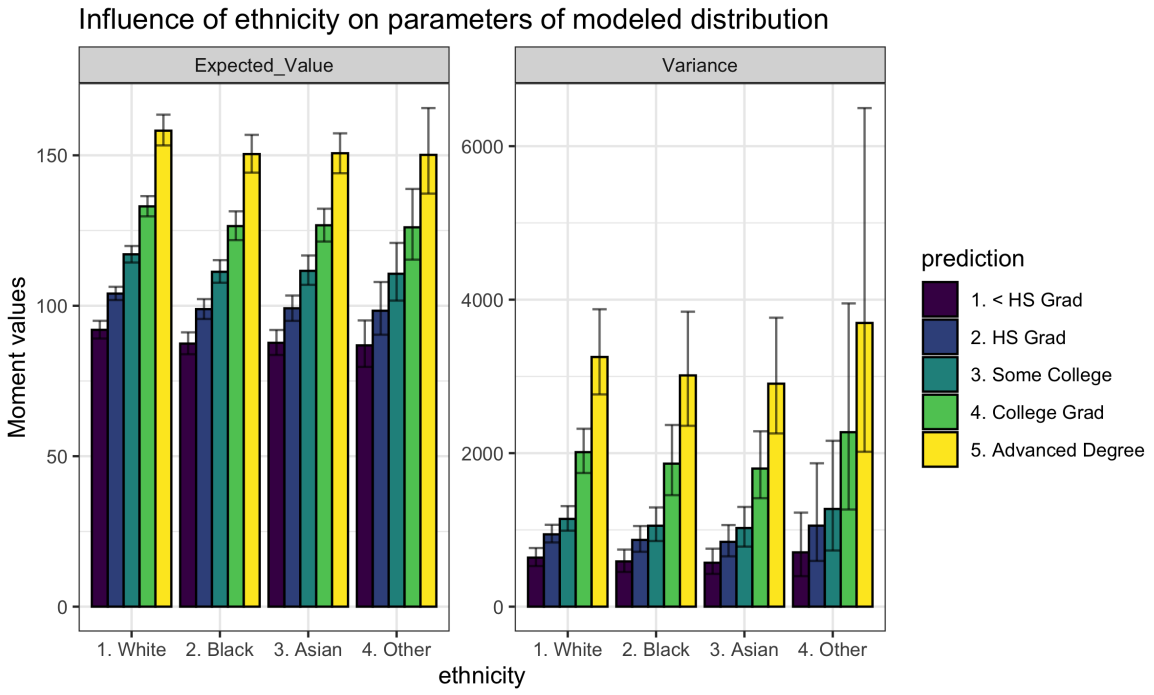

FIGURE 3.4: Impact of variable `ethnicity` on the first two predicted moments of the target wage distribution.

As visible in Figure 3.3, a new graph window was added in comparison to 3.2 depicting the influence of `age` on our specified metric, the Gini coefficient. Further noticable are the credible intervals which we can also obtain if we combine `plot_moments()` with the sample-based approach of `bamlss`.

The function `plot_moments()` is also able to display the difference in moments depending on a categorical covariate. No other arguments have to be specified - `distreg.vis` is able to detect the variable type automatically. In the following code chunk, the variable `ethnicity` is selected as the variable of interest.

```
R> plot_moments(
+    wage_model, int_var = "ethnicity",
+    pred_data = df, samples = TRUE,
+    palette = "viridis", uncertainty = TRUE
+  )
```

Figure 3.4 now shows the result of the code chunk above. On the x-axis we can see the categories of our variable of interest, `ethnicity`. The y-axis now denotes the moments broken up by both the variable of interest and the categories of `education`,

which we specified in the code chunk on page 19. The error bars on the ends of each bar represent credible intervals (95%). From analysing the bars with their error fields we can conclude that the variable `education` mostly yields significantly different expected values of `wage`, while `ethnicity` does not.

From the provided example on modeling wages, it became apparent that distributional regression is a useful tool to handle complex model scenarios. Nonetheless, visualizing the fitted regression models using existing tools is difficult, as transformations are necessary to arrive at interpretable figures. This process is made easier by `distreg.vis`, which gives the abilities to quickly visualize predicted distributions and display marginal moment influences.

3.3 The distributional regression model class

Distributional regression models represent an umbrella class for models where it is possible to link the parameters beyond the mean of a target distribution to available predictors (Klein, Kneib, Lang, & Sohn, 2015). Any choice of the underlying parametric target distribution is valid, as long as the PDF is twice continuously differentiable with respect to the parameters and, in particular, is not limited to the exponential family typically employed in generalized additive models. Assuming a parametric distribution $y \sim D(\theta_1, \ldots, \theta_K)$ with K distributional parameters $\theta_1, \ldots, \theta_K$, we arrive at the model specification

$$\begin{aligned} \eta_k &= \beta_{k0} + \sum_{q=1}^{Q_k} f_{kq}(\mathbf{x}_{kq}; \boldsymbol{\beta}_{kq}) \\ h_k(\eta_k) &= \theta_k, \end{aligned} \tag{3.2}$$

where $h_k(\cdot)$, $k = 1, \ldots, K$ denotes the response function used to uphold the support of parameter θ_k, $f_{kq}(\cdot)$, $q = 1, \ldots, Q_k$ represents the possibly non-parametric effect of covariate(s) $\mathbf{x}_{kq}$ on the modeled parameter and $\boldsymbol{\beta}_{kq}$, $q = 1, \ldots, Q_k$ depict the regression parameters which are to be estimated.

Depending on the software package used to fit a distributional regression model, a

vast selection of possible effects are available, including but not limited to penalized splines (also in multivariate forms), spatial effects based on Gaussian random fields or Markov random fields, varying coefficient terms and random effects (see Fahrmeir et al., 2013, Ch. 9 for an overview). More recently, new research is done on connecting the distributional regression framework to effects known from Machine Learning, e.g., Random Forest (Schlosser et al., 2019). Multivariate extensions have also gained considerable interest in the past, see for example Klein, Kneib, Klasen, and Lang (2015), Klein and Kneib (2016b) or Marra and Radice (2017).

Due to the high flexibility in the target distributions, the predictors and the specified link function, distributional regression encompasses many well-known regression approaches, such as generalized linear models (Nelder & Wedderburn, 1972, GLM), generalized additive models (Hastie & Tibshirani, 1990, GAM), generalized additive mixed models (Lin & Zhang, 1999, GAMM) and generalized additive models for location, scale and shape (Rigby & Stasinopoulos, 2005, GAMLSS). Although in theoretical proximity to GAMLSS, the term distributional regression was coined since distributional parameters do not always represent either the location, scale or shape of a distribution (Klein, Kneib, Lang, & Sohn, 2015).

3.3.1 Implementations of GAMLSS

For estimating distributional regression models in `R`, the two most capable software implementations are `gamlss` (Stasinopoulos & Rigby, 2007) and `bamlss` (Umlauf et al., 2018), with the package `betareg` (Cribari-Neto & Zeileis, 2010) only focusing on beta regression. The main difference in `gamlss` and `bamlss` is rooted in their estimation techniques, which will be described below:

gamlss

The `gamlss` package features a frequentist approach employing (penalized) maximum likelihood inference. Its main estimation algorithm, RS, short for Rigby and Stasinopoulos, uses iteratively reweighted least squares (IRLS) in combination with a modified

backfitting algorithm to arrive at coefficient estimates. The algorithm system is broken up into inner and outer iterations, with each inner iteration depicting the fitting of one distributional parameter θ_k. Here, a working variable $\mathbf{z}_k$ consisting of all used predictors, the first derivative of the likelihood (score function) and "iterative weights" $\mathbf{w}_k$, determined with a local scoring algorithm, is calculated. Then, the working variable is fitted 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_k, 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, Ch. 3).

Estimating parameters with backfitting has the advantage of avoiding the need for cross-derivatives and is therefore quite efficient. Uncertainty assessments typically rely on asymptotic normality assumptions and have been found to be pretty conservative in simulation studies (see for example Klein, Kneib, & Lang, 2015).

The collection of `gamlss` packages provides a vast number of distributions via its accompanying package `gamlss.dist` (Stasinopoulos & Rigby, 2019) as well as several extensions concerning spatial data effects (`gamlss.spatial`, De Bastiani, Stasinopoulos, & Rigby, 2018) or truncated distributions (`gamlss.tr`, Stasinopoulos & Rigby, 2018), for example.

bamlss

The `bamlss` package (Umlauf et al., 2018, BAMLSS) provides a highly customizable Bayesian estimation framework with both posterior mode estimates via penalized likelihood and fully Bayesian inference implemented via MCMC simulation techniques. By default, it revolves mainly around two functions: `bamlss::bfit()` and `bamlss::GMCMC()`. The first function, `bfit()`, utilizes an optimizing function which seeks to find the mode of the posterior distribution via penalized likelihood with respect to the effect coefficients. Then, those values are used as starting numbers for MCMC simulations (function `GMCMC()`), which are based on iteratively weighted least squares

proposals that rely on multivariate normal proposals obtained from locally quadratic approximations of the log-full conditional (Brezger & Lang, 2006).

Both functions can be swapped by the user with optimizer and sampler functions that more closely resemble the subjective preference. As such, the implementation of `bamlss` represents a lego-type toolbox that enables replacing specific parts of the model specification with alternative and potentially more flexible variants without altering the rest of the model implementation. Compared to `gamlss`, the number of supported distributions is more limited but the access to posterior samples facilitates finite sample inference also for complex functionals of the original parameters. Default priors are assigned to all parameters of the model specification, but these can also be controlled by the user.

3.3.2 Distributional compatibility

To ensure a wide user audience, `distreg.vis` is able to support a variety of distributions from the `gamlss`, `bamlss` and `betareg` packages. Table 3.1 gives an overview, and divides the available distributions into those that can be used in both `plot_dist()` and `plot_moments()` (Table 3.1a), and those that can only be used in combination with `plot_dist()` (Table 3.1b). Table 3.1b consists of distributions which do not have existing moments or do not have any implementations yet, rendering them incompatible with `plot_moments()`. To include as many distributional families as possible, we worked together with both the authors of `gamlss` (Stasinopoulos & Rigby, 2007) and `bamlss` (Umlauf et al., 2018) and implemented the moment functions for almost all available distributions in their respective packages.

3.4 Implementation of `distreg.vis`

Understanding the inner workings of `distreg.vis` is best done by starting at the two most important functions, `plot_dist()` and `plot_moments()`. Each function is a direct answer to the two central questions which led to the creation of `distreg.vis`:

Name of distribution	class	Name of distribution	class
beta	bamlss	NO2	gamlss
binomial	bamlss	NOF	gamlss
cnorm	bamlss	PARETO2	gamlss
gamma	bamlss	PARETO2o	gamlss
gaussian	bamlss	PE	gamlss
gaussian2	bamlss	PE2	gamlss
glogis	bamlss	PIG	gamlss
gpareto	bamlss	PO	gamlss
lognormal	bamlss	RG	gamlss
multinomial	bamlss	SHASHo	gamlss
poisson	bamlss	SICHEL	gamlss
BE	gamlss	SN1	gamlss
BEo	gamlss	SN2	gamlss
BNB	gamlss	SST	gamlss
DEL	gamlss	ST2	gamlss
EGB2	gamlss	ST3	gamlss
exGAUS	gamlss	ST3C	gamlss
EXP	gamlss	ST4	gamlss
GA	gamlss	ST5	gamlss
GB2	gamlss	TF	gamlss
GEOM	gamlss	TF2	gamlss
GEOMo	gamlss	WEI	gamlss
GG	gamlss	WEI2	gamlss
GIG	gamlss	WEI3	gamlss
GPO	gamlss	ZAGA	gamlss
GT	gamlss	ZALG	gamlss
GU	gamlss	ZANBI	gamlss
IG	gamlss	ZAP	gamlss
IGAMMA	gamlss	ZAPIG	gamlss
JSU	gamlss	ZASICHEL	gamlss
JSUo	gamlss	ZAZIPF	gamlss
LG	gamlss	ZIBNB	gamlss
LO	gamlss	ZINBI	gamlss
LOGNO	gamlss	ZIP	gamlss
NBF	gamlss	ZIP2	gamlss
NBI	gamlss	ZIPF	gamlss
NBII	gamlss	ZIPIG	gamlss
NO	gamlss	ZISICHEL	gamlss
betareg	betareg	-	-

(a)

Name of distribution	class	Name of distribution	class
BCCG	gamlss	LOGNO2	gamlss
BCCGo	gamlss	LQNO	gamlss
BCPE	gamlss	SEP	gamlss
BCPEo	gamlss	SEP1	gamlss
BCT	gamlss	SEP2	gamlss
BCTo	gamlss	SEP3	gamlss
BEINF	gamlss	SEP4	gamlss
BEINF0	gamlss	SHASH	gamlss
BEINF1	gamlss	SHASHo2	gamlss
BEOI	gamlss	ST1	gamlss
BEZI	gamlss	ZABB	gamlss
BI	gamlss	ZABI	gamlss
DPO	gamlss	ZAIG	gamlss
GB1	gamlss	ZIBB	gamlss
GP	gamlss	ZIBI	gamlss
LOGITNO	gamlss	ZINBF	gamlss

(b)

Table 3.1: Supported dependent variable distributions in `distreg.vis`. Table b) shows distributions where only visualizing the PDF/CDF depending on specified covariate combinations is possible (`plot_dist()`). Table a) lists the distributions where both the PDF/CDF function and the "moments plot" detailing the influence of a covariate on the distributional moments (`plot_moments()`) are supported.

1. What exactly does my predicted distribution look like, based on selected covariate combinations?

2. How do the expected moments of my target distribution vary over the range of a specific variable of interest?

The remaining functions of the `distreg.vis` solely feature a supportive character

easing the overall computative process. Here, we start with `plot_dist()`:

3.4.1 Visualization of distribution predictions

To accurately display the predicted distribution of a covariate combination, the first step is to gather the appropriate distributional parameters from which to build the PDF or CDF. These parameters are obtained by choosing certain covariate combinations and then using the estimated model for prediction. These covariate combinations are usually user-picked, but `distreg.vis` offers a few tools to ease the choosing process, as shown below.

Choosing covariate combinations

This section introduces tools to make using both `plot_moments()` and `plot_dist()` easier by creating datasets to predict with in the correct format. To quickly obtain a `data.frame` object with all variables set to their mean, one can use `model_data()` to obtain the covariates with which the model was estimated, and then `set_mean()` to calculate the average values of each covariate. Both functions and their arguments are repeatedly used in and as such are part of the main functions in `distreg.vis`, `plot_dist()` and `plot_moments()`.

The usage of `model_data()` is as follows:

```
model_data(model, dep = FALSE, varname = NULL)
```

where `model` represents one of `distreg.vis`' supported model classes. It returns a `data.frame` object containing the explanatory variables, except when `dep = TRUE` (returns vector of dependent variable) or when `varname` is specified in character form (returns a vector of a specifically named explanatory variable).

After obtaining the explanatory variables, reducing its dimensions to the mean is achieved with `set_mean()`, the usage of which is as follows:

```
set_mean(input, vary_by = NULL)
```

With the argument `input`, a `data.frame` object is specified, which can conveniently be the output of a `model_data()` call. Then, average values are calculated of each

column in `input`. For categorical variables, the first level of the underlying factor class is taken, which in model estimation typically corresponds to the reference category.

Using the argument `vary_by()`, the user can specify an explanatory variable in character form, over which the returned `data.frame` is "varied". This means that the returned object consists of the same number of columns as before, but with multiple rows and varying values in the covariate of choice. If this variable is categorical, then the varying values consist of the possible categories. In numeric cases, a sequence of five values, ranging from the 2.5% to the 97.5% quantile is created. All other variables stay constant at their mean or reference category. This argument is very useful for a display of marginal distributions, based on the resulting `data.frame` object.

To show an example of the ease of use of the aforementioned functions, we consider the same model that was fit in Section 3.2. To retrieve the explanatory variables of that model and set them to the mean with varying values in the categorical covariate `education`, we can run the code shown on Page 19. From the code chunk results, we can now quickly make parameter predictions and obtain either graphs with the marginal distribution (`plot_moments()`), or marginal influence plots (`plot_moments()`). Further defining the `row.names` of the `data.frame` to be the different education levels ensures improved legends in further graphs.

Parameter predictions

Predicted parameters of the target distribution are usually outputs of `predict.object()` functions for each regression model class. This is no different in the case of `bamlss`, `gamlss` or `betareg`, for which those functions also exist. Unfortunately, the usage of all three `predict` functions is not consistent over its respective packages, and sometimes not even within the package throughout parameters of different target distributions.

For example, `predict.bamlss()` returns a vector if the predicted distribution only consists of one parameter, a list of vectors if it consists of more than one, and a list of matrices if the MCMC samples of the predicted parameters are desired. To ensure unity and guarantee type consistency over the supported packages outcome classes, `preds()` was written. Without worrying about class-specific function arguments, it offers a

consistent way of obtaining predictions based on specific covariate combinations. Its usage is as follows:

```
preds(model, newdata = NULL, what = "mean", vary_by = NULL)
```

where `model` represents a fitted `gamlss`, `bamlss` or `betareg` object and `newdata` a `data.frame` with different covariate combinations in each row. If `newdata` is omitted (`= NULL`), then the mean of the explanatory variables are used for prediction, utilizing `set_mean()`. This can be used in combination with the argument `vary_by`, which then creates a `newdata` that consists of varying values in one specific variable (see Chapter 3.4.1 for details).

The argument `what` specifies whether the predicted parameters should be directly calculated (`what = "mean"`), or the output should consist of samples of the predicted parameter (`what = "samples"`). This option is only available for the `bamlss` model class, and is necessary for exact estimates of the predicted moments or other measures derived from the parameters and for the construction of credible intervals at a later stage (e.g., for plotting).

Obtaining samples for the predicted parameters of our target distribution, based on the different education levels defined in `df` can now be done as follows, using the option `what = "samples"`:

```
R> pp_samples <- preds(model = wage_model,
+                      newdata = df,
+                      what = "samples")
R> lapply(pp_samples, head, 2)

$`1. < HS Grad`
           mu     sigma
[1,] 4.506987 0.2850489
[2,] 4.485811 0.2498074

$`2. HS Grad`
           mu     sigma
[1,] 4.630472 0.2872791
[2,] 4.592440 0.2883318
```

```
$`3. Some College`
           mu     sigma
[1,] 4.759729 0.2878243
[2,] 4.712548 0.2792889

$`4. College Grad`
           mu     sigma
[1,] 4.864325 0.3270728
[2,] 4.816144 0.3139536

$`5. Advanced Degree`
           mu     sigma
[1,] 4.999823 0.3329738
[2,] 4.991542 0.3368283
```

In this case, the object `pp_samples` was produced as a `list` with as many elements as preselected covariate combinations. Each element is a matrix containing samples in rows (by default `bamlss` keeps $n = 1001$, of which the first two rows are shown above) for all distributional parameters (in this case two: μ, σ). The two plotting functions of `distreg.vis` automatically recognize the object and can extract information from both maximum-likelihood estimates (MLEs) (`gamlss` or `betareg`) or MCMC samples (`bamlss`). If the predicted parameters are provided in the form of samples (as a `list` object in `R`), `plot_moments()` is able to display credible intervals above and below the predicted moments.

Visualize predicted distributions

To get a feel for the predicted distributions and their differences, it is best to visualize them. In combination with the obtained parameters from `preds()`, the function `plot_dist()` finds the necessary distribution functions (PDF or CDF) from the respective packages and then displays them graphically.

Creating the distributional graph is a five step process, beginning with the right input; the arguments. The usage of `plot_dist()` is as follows:

```
plot_dist(model, pred_params = NULL, palette = "viridis",
  type = "pdf", rug = FALSE, vary_by = NULL, newdata = NULL)
```

As before, `model` depicts the fitted distributional regression model. The argument `pred_params` takes the previously predicted parameters of the fitted distribution. Options `palette` and `rug` yield aesthetic changes; the first one is able to process any character string that resembles either `"default"`, which results in displaying the default color palette of `ggplot2` (Wickham, 2016), `"viridis"` for a colorblind-friendly palette from the `viridis` package (Garnier, 2017) or a palette from the `RColorBrewer` package (Neuwirth, 2014).

If the user wishes to leave out the step of specifying predicted parameters with `pred_params`, the argument `newdata` can be provided instead. This argument will then be passed onto `preds`, which internally computes the predicted parameters used for plotting the distribution functions. If neither `pred_params` nor `newdata` are specified, then `plot_dist()` uses the mean values of the explanatory variables with `set_mean()` to predict the parameters used for plotting. The argument `vary_by` is also passed on to `set_mean()` which, if specified, leads to `plot_dist()` displaying marginal distributions over the range or categories of a specified variable (see Section 3.4.1 for details).

The last argument `rug`, as the name implies, adds a "rug graph" below the distribution plots, which shows small vertical strips indicating where actual observations of the dependent variable lie. Using the argument `type`, one can switch between PDFs (`"pdf"`) or CDFs (`"cdf"`).

After `plot_dist()` has received all necessary arguments, it executes validity checks to ensure the argument's correct specification. This includes controlling for the correct `model` class, checking whether the distributional family can be used safely and whether CDF or PDF functions for the modeled distribution are present and ready to be graphically displayed. If this is the case, the internal `fam_fun_getter()` is used to create a `list` with two functions pointing to the correct PDF and CDF functions for the distributions from either the `gamlss`, `bamlss` or `betareg` namespace.

To graphically display the previously obtained functions, the only remaining unanswered question is one about the limits: what should be the axis' ranges? To accommodate this task, the internal function `limits()` was created. The default option in `distreg.vis` for the x-axis on both CDF and PDF function displays are the 0.1% and the 99.9% quantiles of the predicted distribution. If multiple distributions are displayed, always the minimum and maximum values of all calculated quantiles are used.

Both the validity checks and the `limits()` function make use of specific information about every available distribution in the supported distributional regression packages. This information is stored in a `data.frame` object called `dists` with the following form:

```
R> str(dists)
'data.frame': 125 obs. of  8 variables:
$ dist_name  : chr  "BB" "BCCG" "BCCGo" "BCPE" ...
$ class      : chr  "gamlss" "gamlss" "gamlss" "gamlss" ...
$ implemented: logi  FALSE TRUE TRUE TRUE TRUE TRUE ...
$ moment_funs: logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
$ type_limits: chr  "both_limits" "one_limit" "one_limit" "one_limit" ...
$ l_limit    : int  0 0 0 0 0 0 0 0 0 0 ...
$ u_limit    : int  10 NA NA NA NA NA NA 1 1 1 ...
$ type       : chr  "Discrete" "Continuous" "Continuous" "Continuous" ...
```

The `dists` object contains one row for each distribution, and columns with the following content:

- `dist_name`: Name of the distribution.
- `class`: Either `"bamlss"`, `"gamlss"` or `"betareg"` detailing from which package the target distribution comes from.
- `implemented`: Is this distribution generally usable for `plot_dist()`, and was this usage already tested?
- `moment_funs`: Are functions implemented with which to calculate the moments

of the distribution, given the parameters? This column is notably relevant for `plot_moments()`, in which the predicted moments are displayed.

- `type_limits`: Details the range the values from the distribution can have. Can be `"both_limits"`, `"one_limit"`, `"no_limit"` and `"cat_limit"` (for categorical distributions).
- `l_limit, u_limit`: Integers detailing where the limits of the distributions lie.
- `type`: Character string for the type of distribution. Can be `"Discrete"`, `"Continuous"`, `"Mixed"` and `"Categorical"`.

Following a successful calculation of the plot limits, the graph itself can be created. Internally, `distreg.vis` divides between continuous, discrete and categorical distributions. Continuous distributions are displayed as filled line plots, while discrete and categorical distributions take bar graph shapes.

For plotting, `distreg.vis` relies on the `ggplot2` package (Wickham, 2016). After an empty graph is constructed, the previously obtained CDF or PDF functions are evaluated for each predicted parameter combination and all values inside the calculated plot limits.

Figure 3.5 lays out examples of the five different shapes of `plot_dist()`: the PDF of continuous, discrete and categorical distributions, as well as the CDF of continuous and discrete distributions.[2] In each of those cases (subplots a) to e)), a dataset was simulated of an appropriately chosen target distribution, which was then used for model estimation. Based on these models and five arbitrary covariate combinations from the datasets, the predicted PDFs and CDFs were computed and displayed.

3.4.2 Visualization of effect influence

The target of the second main function, `plot_moments()`, is to display the influence of a selected effect on the predicted moments of the modeled distribution. The motivation

[2] In contrast to the notation of this thesis, the graphs in `distreg.vis` use $f(\cdot)$ and $F(\cdot)$ to denote PDF and CDF functions, respectively.

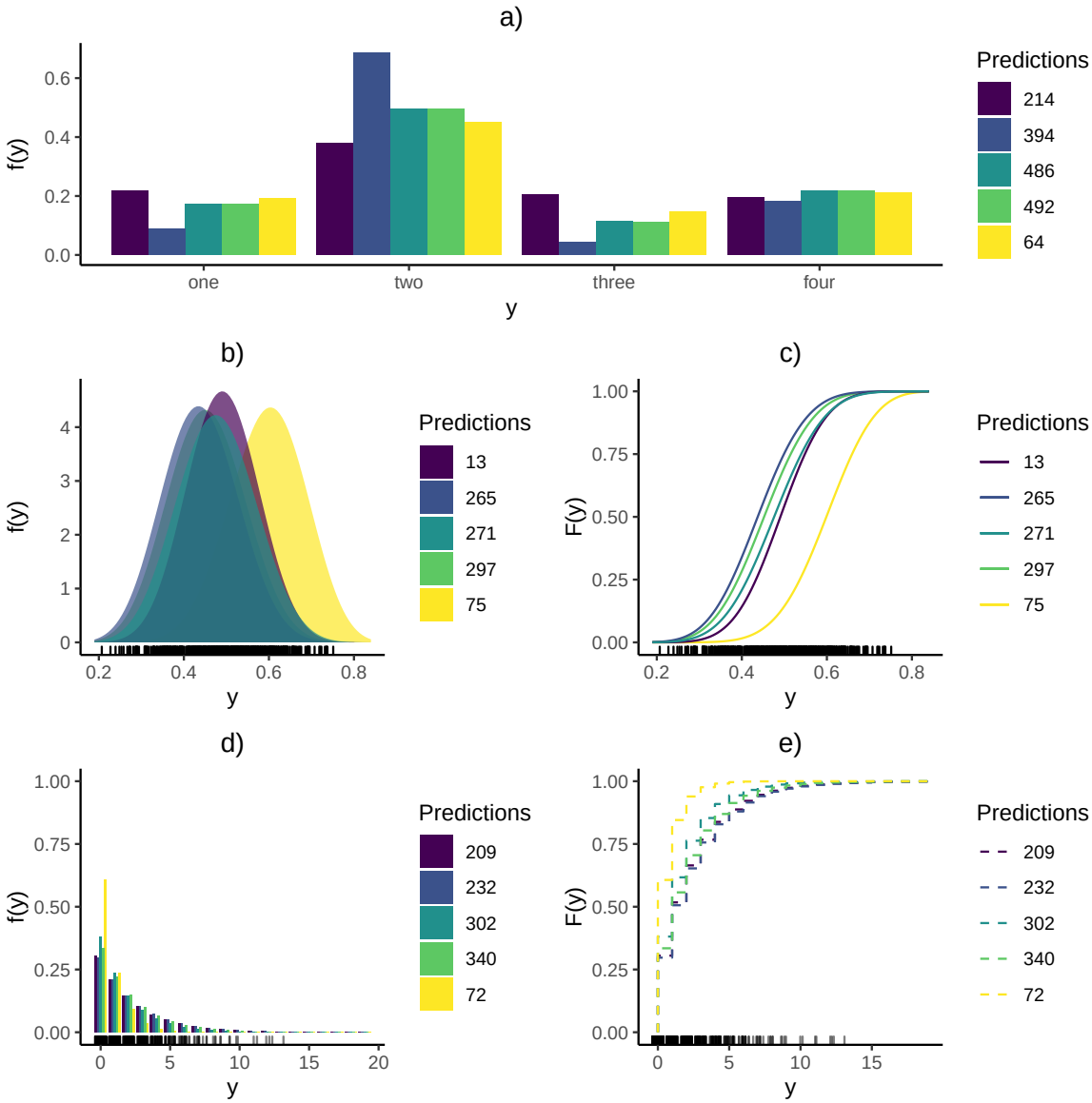

Figure 3.5: Plot outcome possibilities in `plot_dist()` displayed with five different predictions in five examples: a) a multinomial PDF, b) an expected PDF with the respective CDF of a beta distribution shown in c), and the expected PDF and CDF pair of a geometric distribution in d) and e), respectively.

for computing influences on the moments of a distribution is its interpretability: in most cases, the parameters of a distribution do not equate to the moments and as such are only indirectly location, scale or shape properties, making the computed effects hard to understand.

When `plot_moments()` is called, the provided function arguments are evaluated first. They are specified as follows:

```
plot_moments(model, int_var, pred_data = NULL, rug = FALSE,
```

```
  samples = FALSE, uncertainty = FALSE, ex_fun = NULL,
  palette = "viridis", vary_by = NULL)
```

Here, `model` represents the fitted distributional regression object (as before). The argument `int_var` stands for "variable of interest" and consists of a character string with the name of the explanatory variable, whose effect influence should be plotted. This covariate can be either categorical or continuous. The argument `pred_data` is specified using a `data.frame` object that consists of the values that the remaining variables should attain, while `int_var` varies. If this `data.frame` has multiple rows, the influence plot is re-drawn for each different covariate combination. This is a neat way to, for example, visualize interaction effects between two variables.

If `pred_data` is not specified (left at `NULL`), `plot_moments()` will utilize `set_mean()` in combination with `model_data()` to use mean values of the explanatory variables for prediction. If the argument `vary_by` is specified, it will be passed onto `set_mean()` for creating a `data.frame` object in which one variable has varying values, while all others stay at their mean (see Chapter 3.4.1 for details).

Using the arguments `samples` and `uncertainty`, one can exploit the advantages that Bayesian regression models have to offer. With `samples = TRUE` the predicted moments are computed by transforming the samples of the predicted parameters to the moments, and then calculating the mean of the samples over the range of `int_var`. If the argument `samples` was chosen to be `FALSE` (the default case), then the mean of the samples is taken before they are transformed to the predicted moments.

This distinction is especially relevant in nonlinear parameter-to-moment transformations, where taking the average before transforming the samples yields inaccurate results. Setting `samples = FALSE` should therefore only be used in situations where one is interested in quick and rough effect influences. For complete accuracy, setting `samples = TRUE` is preferred.

With `uncertainty`, the user can construct credible intervals for the influence plots. If this argument is set to `TRUE`, then empirical 2.5% and 97.5% quantiles are computed for the predicted moments of the entire range of our variable of interest, `int_var`. Using this method, it is possible to detect complex significant differences between the

moments of a categorical variable, for example. Both `samples` and `uncertainty` can only be set to `TRUE` for `bamlss` model classes.

The argument `ex_fun` takes a character string with the name of an R function that currently exists in the user's global environment. This function can be any function which takes the predicted parameters as input and yields a single number as an output. For situations where metrics depend on the entire predicted distribution of the target variable, like the Gini coefficient (Lerman & Yitzhaki, 1984), this is especially useful. Combined with the sample-based approach of `bamlss`, we can even obtain uncertainty measures for those user-defined metrics.

After typical argument validity checks, a sequence is created that covers the entire range of the variable of interest, `int_var`. This sequence is then combined with `pred_data` to obtain a `data.frame` object where the main variable varies but the previously specified other variables stay constant. The newly created `data.frame` is further passed to the `preds()` function, which computes predicted parameters for every row.

Depending on whether `samples` was set to `TRUE` or not, the output of the called `preds()` function is now either a `data.frame` with the same dimensions as `pred_data` and with the parameters replaced by the predicted moments, or a `list` with one element for each user-specified covariate combination consisting of the parameter samples which were transformed to the expected moments of the target distribution. Afterwards, the mean of all moment samples at each point of `int_var`'s range is calculated, resulting in all necessary values to create the resulting influence graph. If `uncertainty` is also `TRUE`, the upper and lower limits of the credible intervals are constructed.

Obtaining the predicted moments and possibly lower and upper bounds of the credible intervals means that the resulting graph can then be created. To illustrate the potential of `plot_moments()`, we will again focus on the introductory example, where the question of interest is the relationship between annual income and different socioeconomic variables. Calling `plot_moments()` for both a numeric (`age`) and a categorical variable (`ethnicity`) can be done as follows:

```
R> gini <- function(par) {
+    2 * pnorm((par[["sigma"]] / 2) * sqrt(2)) - 1
```

```
+  }
R> plot_moments(
+    wage_model,
+    int_var = "age",
+    pred_data = df_new,
+    samples = TRUE,
+    uncertainty = TRUE,
+    ex_fun = "gini",
+    rug = TRUE
+  )
R> plot_moments(
+    wage_model,
+    int_var = "ethnicity",
+    pred_data = df_new,
+    samples = TRUE,
+    uncertainty = TRUE,
+    ex_fun = "gini"
+  )
```

Figure 3.6 shows the outcome of the above code, which represents two of the three different graph outcomes that `plot_moments()` can have. Part a) focuses on the relationship between the `age` of observed individuals, a numeric variable, and the first two expected moments of the modeled income distribution (leftmost and middle graph). The different lines represent multiple education levels, which are based on the `data.frame` object called `df` (p. 19). For a less cluttered visualization, this portrayal of the `age` influence features only three different education levels resulting in three different lines (hence the subset `df_new`).

Even though the first two expected moments are already informative, one might be interested in other specific measures being dependent on the parameters of a distribution. This could include higher-order moments like skewness or more specific measures, like income inequality. Previously, finding out how this measure varies over a variable would involve many lines of code. With `distreg.vis`, specifying an external function is easy. The very right graph of part a) portrays the external function (`gini()`) that was specified with the argument `ex_fun`. It represents the Gini coefficient of the

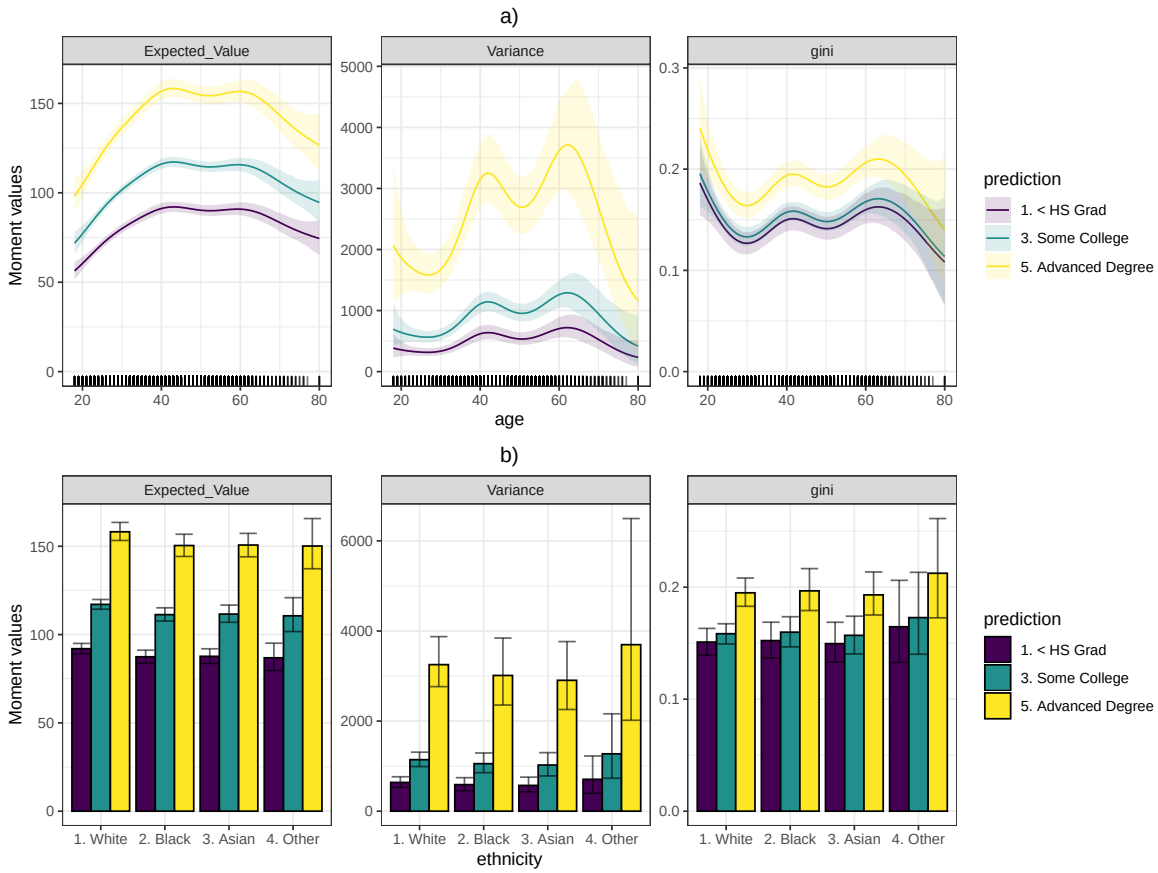

Figure 3.6: Plot outcome possibilities in `plot_moments()`: part a) shows the expected moments (`Expected_Value` and `Variance`) and a user-specified function (`gini()`) over the range of a continuous covariate, including its credible intervals. Part b) displays the expected moments and the external function over the range of a categorical variable, `ethnicity`.

distribution, which is a measure of inequality of a distribution (Lerman & Yitzhaki, 1984). Similarly to the expected moments, we also construct credible intervals around the self-specified function `gini()`, as shown in Figure 3.6. It is similar to Graphs 3.3 and 3.4, with two categories removed to increase visibility.

Below and above the lines of the varying moment and external function levels lie shaded areas which represent the uncertainty of the influence. These are credible intervals, i.e. empirical 2.5% and 97.5% quantiles of the obtained MCMC samples transformed to be expected moments. The advantages of having uncertainty areas are numerous, but include the direct visibility of significant differences between two different categories chosen with the `data.frame` object called `pred_data`. For example, the left plot of part a) shows a significant difference in the expected income for all

three chosen education levels, because the credible intervals do not overlap in almost the entire range of `age`.

The bar graphs of Figure 3.6 part b) appear if a categorical variable is chosen as the variable of interest `int_var` (in this case it was chosen to be `ethnicity`). The x-axis portrays all different categories that the variable can attain. Then, the height of each bar features the expected mean or variance of the predicted distribution when `int_var` reaches the specific category. The different bars in each category represent again the covariate combinations specified in the `pred_data` argument. Due to the `pred_data` covariate values only varying in the `education` levels of the individuals, one could understand part b) of Figure 3.6 as the predicted expected value, variance and external function of the modeled distribution, broken down to each combination of both `education` and `ethnicity`.

Similar to the continuous case, an external function can also be included for discrete variables of interest. The function called `gini()` was used again, calculating the Gini coefficient for the log-normal distribution. The small error bars, also only available for `bamlss` objects, show the credible intervals around the predicted moments in each category. The significance between categories can now be checked in two ways - between the categories of `int_var` and between the different covariate combinations chosen in `pred_data`. In this case, we can see a significant wage difference between the `education` levels but not the `ethnicity` categories.

3.5 The graphical user interface

After executing `vis()`, a new browser window with the started application is opened up. Figure 3.7 shows the layout of the application, which is then displayed in the user's browser.

As visible in Figure 3.7, the layout of `distreg.vis`' GUI 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 right segment, with

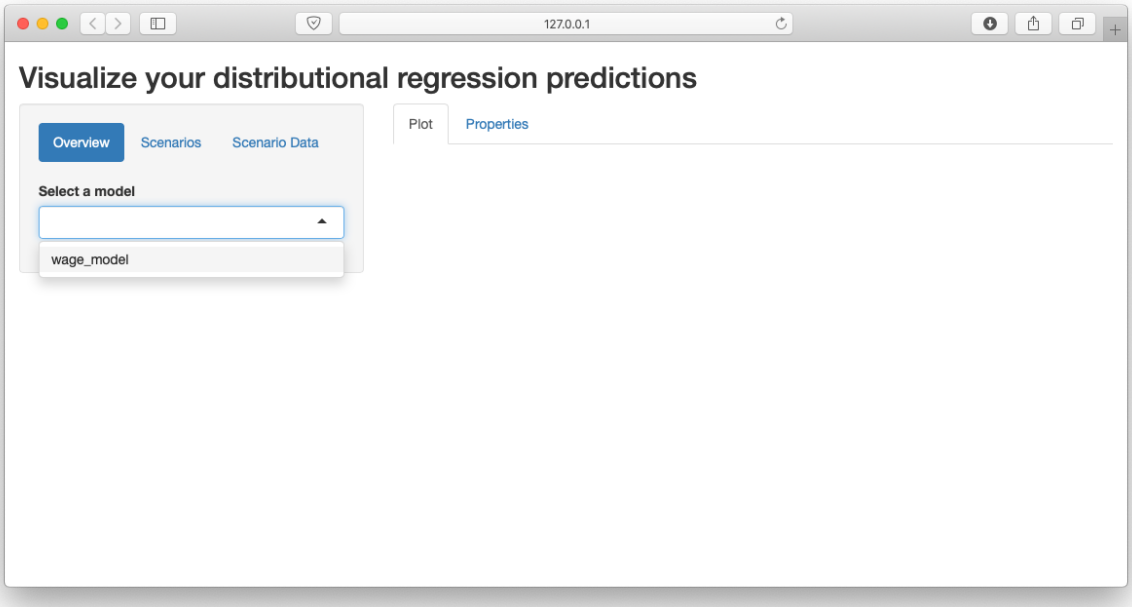

Figure 3.7: Layout of `distreg.vis` after starting the application.

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

3.5.1 Overview tab

The purpose of the overview tab is to display descriptive details about fitted distributional regression models. After the GUI of `distreg.vis` is started up, it solely consists of a drop-down list where the user can select the model on which the following analysis is to be based. Entries in this list are created by the internal function `search_distreg()` which searches the working directory of the current user for any object of the classes `bamlss`, `gamlss`, `betareg` or `betatree` (also from the `betareg` package). Figure 3.7 shows only one entry, `wage_model`, representing the model fitted with the code provided in Section 3.2 of the main paper.

After a model is selected, the overview tab expands to show an outline of the fitted `bamlss` model. Specifically, as shown in Figure 3.8, the tab displays two parts, called "Model Family" and "Model Equations". "Model Family" shows the family of the model's target distribution, as well as the parameters which can be modeled including their link functions. In the case of `wage_model`, the family "lognormal" with parameters

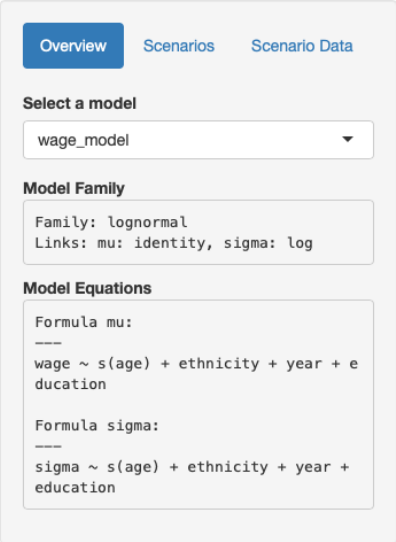

FIGURE 3.8: Expanded overview tab after model selection.

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

3.5.2 Scenarios tab

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

As displayed in Figure 3.9, the top of the tab consists of two buttons, "Create Scenario" and "Clear Scenarios". Right below, web widgets for each explanatory variable are visible. Here, `distreg.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 variable's possible categories, while for numeric variables slider modules are created (`shiny::numericInput()`), ranging from the variable's minimum to maximum

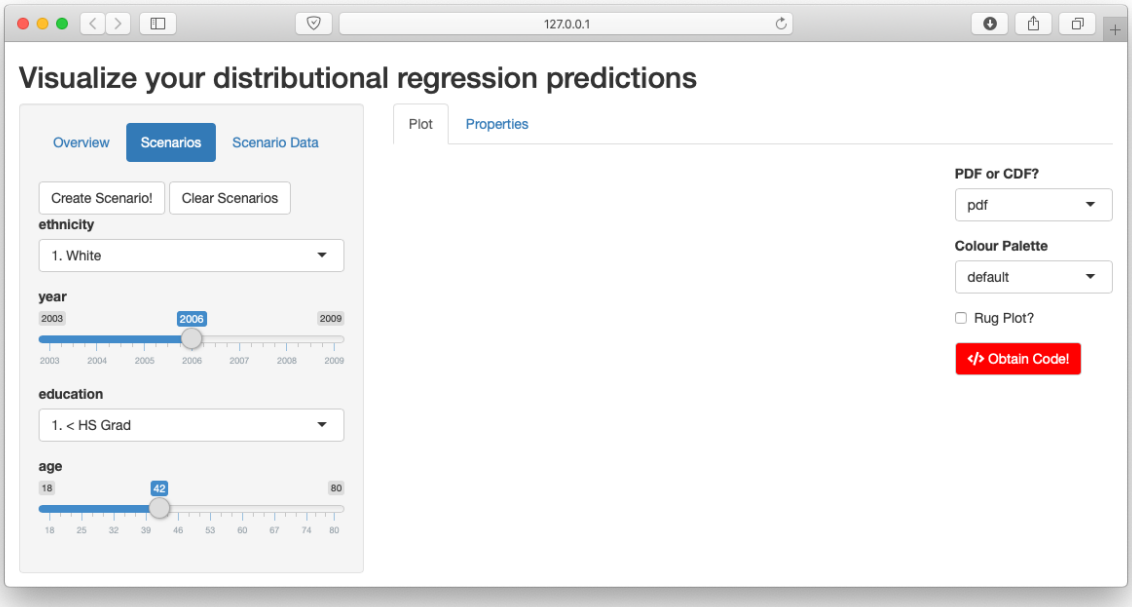

Figure 3.9: "Scenarios" tab of the `distreg.vis` GUI.

value. The default value for numeric covariates is its arithmetic average, while for categorical covariates the first level of the resulting `factor` variable is used.

To add a new scenario, the user can now specify a value for each variable and click on "Create Scenario". This triggers the plotting window, which in turn displays the predicted distribution for the specified covariate combination. To compare two or more scenarios, the user can change the specified covariate values and again press "Create Scenario". This way, the graph window will display predicted PDF/CDF functions for each specified covariate combination. Deleting all previously specified combinations can be achieved by clicking "Delete Scenario".

Beneath the visual components of the "scenarios" lies 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.

We can now easily create a graph for the predicted distribution of wages depending on every `education` level. To do so, we set the covariates to the following values:

- `ethnicity: 1. White`

- `year: 2006`
- `education: 1. < HS grad`
- `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 step. Thus, we can now view the according wage distributions for a 42-year-old white male across all levels of education (Figure 3.10). Furthermore, the "rug plot" option was selected.

3.5.3 Plot tab

The "Plot" tab, which is located in the right-hand segment of `distreg.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 `distreg.vis::preds()`, which then computes predictions for the response distribution parameters. Afterwards, the predicted parameters are inserted into the PDF or the CDF, which is then graphically displayed. This procedure is repeated for each covariate combination.

Figure 3.10 shows the plot output for five different scenarios based on the `Wage` dataset. Noticeable in Figure 3.10 to the right side of the plot are multiple web elements for user interaction. The first element, found below the description "PDF or CDF?", provides the ability to switch between displaying the PDF (the default value) or the CDF. Figure A.3 in the Appendix shows Figure 3.10 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, 2016). The selected color palette in Figure 3.10 is "viridis" (from R package `viridis`), a colorblind-friendly palette (Garnier, 2017).

Using the third element labelled "Rug plot?", a small strip chart at the bottom of the graph can be added, displaying the observation frequency at each level of the target variable's range. The fourth web element on the right side of the plot output,

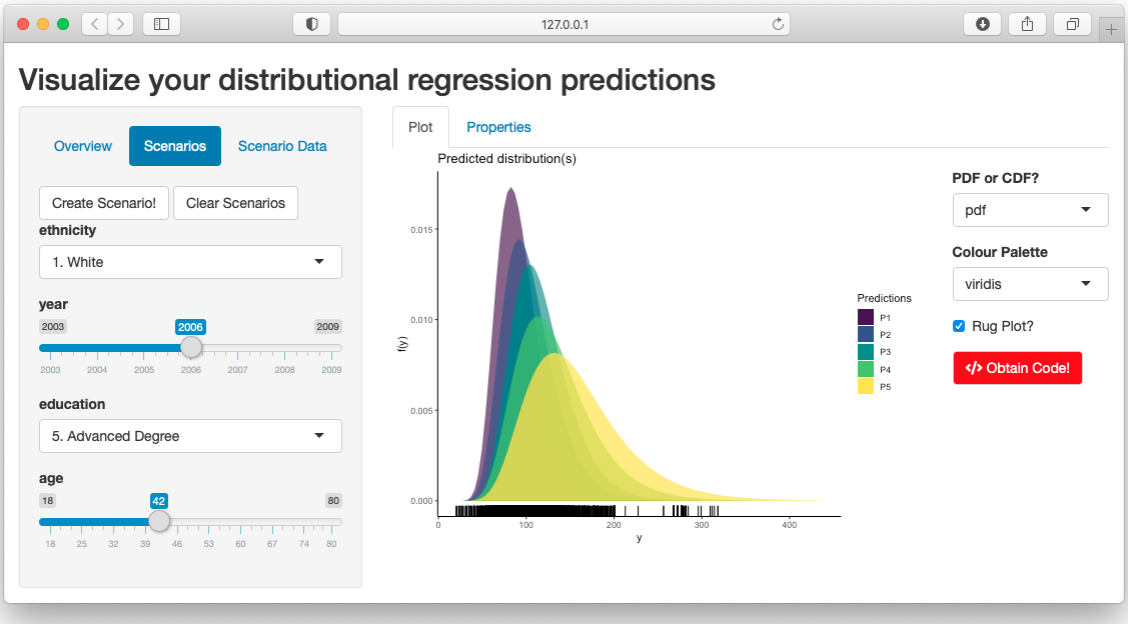

Figure 3.10: Plot tab output when specifying five different scenarios with different education levels.

a red button with 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 A.4 in the Appendix shows the modal window which arises when pressing the "Obtain Code!" button in the interface of Figure 3.10.

3.5.4 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 with 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 `distreg.vis`' left segment.

Figure 3.11 shows the tab's layout. As visible, 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,

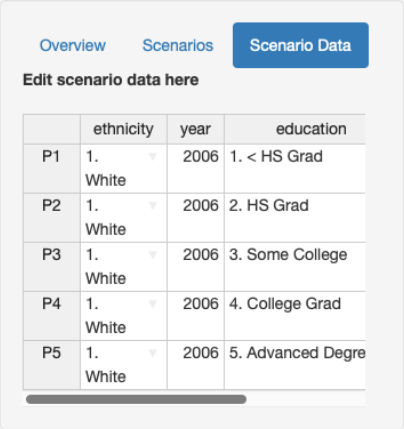

FIGURE 3.11: "Scenario Data" tab.

while one row counts as one "scenario". In Figure 3.11, 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. One additional column is added to the existing covariates, labelled "rownames". With this column, each "scenario" obtains a new label, which then automatically transfers to the legend of each graph.

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 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 un-checking 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 prevent the irresponsible usage of model predictions, `distreg.vis` will display a warning pop-up message with the name of the covariate combination (`P` plus the respective number if not changed with column "rownames") where values were specified which are out of the original variable's range (Appendix: Figure A.5).

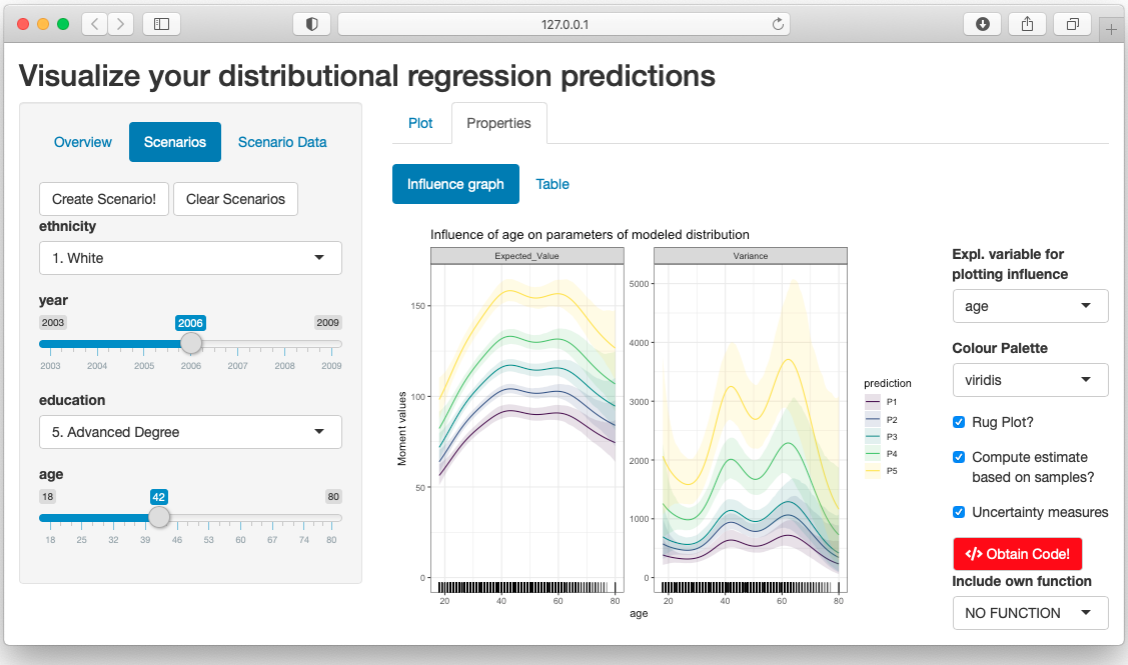

Figure 3.12: Influence of `age` on the first two moments of the predicted distributions for `wage_model`.

3.5.5 Properties tab

Previous sections have described that the "Plot" tab visualizes the predicted distributions for each covariate combination specified in the "Scenarios" tab. However, while differences in distributions for each combination were visible (e.g., in Figure 3.10), it is only indirectly possible to infer influences of the complete range of a numerical covariate on the distributional moments.

To provide this functionality, the tab "Properties" was implemented in `distreg.vis`, located in the right-hand segment and greatly building upon the function `plot_moments()`. When opened, the "Properties" tab reveals two sub-tabs: "Influence graph" and "Table".

As visible in Figure 3.12, the "Influence graph" tab consists of a graph on the left side next to a small bar with additional options. On the right side, the continuous variable `age` was selected. This leads to a recreation of Figure 3.6 part a) without the specification of the external function. Selecting the categorical variable `ethnicity` would lead to part b) of Figure 3.6.

Next to the plot in the "Influence graph" tab multiple web elements are displayed.

The first, located under the description "Expl. variable for plotting influence", lets the user select the explanatory variable for which the influence plot shall be created. The second element gives options to choose a color palette, similar to Figure 3.10. In Figures 3.12 the "viridis" color palette was specified to account for potential colorblindness. Below the palette selector the same red button as in Figure 3.10 is placed, which when pressed, presents code to reproduce the influence plot with (Appendix: Figure A.6).

At the bottom end of the web elements, we see a selector element titled "Include own function", which corresponds to the `plot_moments()` argument `ex_fun`. If a user-written function calculating a measure based on the expected parameters of the target distribution is specified here, the influence of the selected covariate on this measure is calculated and included in the plot (similar to part a) of Figure 3.6).

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" without it depending on a selected covariate's range. Figure A.7 shows this tab's layout.

The only present element in this tab is a table showing 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.

3.6 Prediction of gym attendance

This section will focus on highlighting the features of `distreg.vis` using a real-world example. Specifically, a dataset on visitor numbers in a sports gym in the German town of Göttingen will be analyzed.

In the sports life of Göttingen, a university town with a population of 120,000, the Fitness- und Gesundheitszentrum (FIZ) plays a significant role. Adjacent to and cooperating with the university's sports faculty, it is one of the biggest gyms in the city and provides a wide range of training opportunities.

Due to the FIZ's "high quality yet affordable price" policy, its visitor numbers are often very high, leading to longer waiting times for fitness machines and therefore a

variables	type	content
no_people	integer	Number of checked-in customers at the Fitnesszentrum (FIZ).
weekday	factor	Day of the week (1-7). Starts with 1 (Monday).
hm_num	numeric	Numeric representation of the 5 minute period **end time**. Is calculated as $hm\_num = hour * 60 + minute$.
stime	logical	Was this datapoint recorded during the lecture period (`TRUE` or `FALSE`).
day_collect	numeric	Number of days since attendance recording began.
day_of_year	integer	Number of days since the beginning of the year.

Table 3.2: Type and description of variables contained in the FIZ dataset used in this chapter. $n = 179{,}004$.

decrease in training efficiency. Finding out when those situations arise and what drives visitor numbers is of general interest and will be the focus of this analysis.

The dataset was provided by the "Verein für Freizeitsport und Gesundheitstraining an der Georg-August-Universität Göttingen e.V", who also operate the FIZ. Each check-in and check-out of a single visitor since January 2016 was recorded. We then counted the current number of visitors in sequences of five minutes. Table 3.2 shows the description of covariates utilized in this section.

As the number of visitors cannot be negative but otherwise does not have an upper limit, only distributions with a positive support can be considered. For ease of interpretation, we choose a censored normal distribution to describe attendance numbers. The resulting model specification therefore has the following form:

$$\begin{aligned} no\_people &\sim \text{CensoredNormal}(\mu, \sigma) \\ \mu &= \beta_{10} + f_{11}(weekday) + f_{12}(hm\_num, weekday) \\ &\quad + f_{13}(stime) + \beta_{14} \cdot day\_collect \\ &\quad + f_{15}(day\_of\_year) \\ \log(\sigma) &= \beta_{20} + f_{21}(weekday) + f_{22}(hm\_num) \end{aligned} \tag{3.3}$$

where f_{11}, f_{13}, f_{21} represent categorical effects. The effects f_{15} and f_{22} are modeled using simple penalized splines (P-splines), while $f_{12}(hm\_num, weekday)$ stands for a

P-spline that is estimated independently for each category in `weekday`.

Estimating the model with `bamlss` can then be done using the following code:

```
bam <- bamlss(list(
  no_people ~ weekday +
    s(hm_num, bs = "ps", by = weekday) +
    stime +
    day_collect +
    s(day_of_year, bs = "ps"),
  sigma ~ weekday +
    s(hm_num, bs = "ps")),
  data = fiz_number,
  family = cnorm_bamlss()
)
```

After completion of model estimation, we can start analyzing it with `distreg.vis`. First we call the GUI with `distreg.vis::vis()`, and then select the previously created `bam` object. Further, we head to the "Scenarios" tab to obtain predictions of the visitor number distribution.

Anecdotal evidence shows that evenings are usually busier than afternoons, which are in the middle of the workday. We can now find out whether the model predicts the same behavior. In this example, we compare a Monday afternoon to a Monday evening.

Table 3.3 shows the two scenarios we are comparing. The values $hm\_num = 902, 1112$ represent the times of 15:02 and 18:32, while $day\_of\_year = 190$ stands for the 9th of July, which in 2018 was the 919th day since recording of visitor numbers started ($day\_collect = 919$). Furthermore, the beginning of July is still within of the lecture period ($stime = TRUE$).

Scenario	weekday	stime	day_collect	hm_num	day_of_year
afternoon	1	TRUE	919.00	902.00	190
evening	1	TRUE	919.00	1112.00	190

TABLE 3.3: Covariate values corresponding to the two scenarios in Figure 3.13, in which the distribution of gym attendance on a Monday afternoon is compared to the evening on the same day.

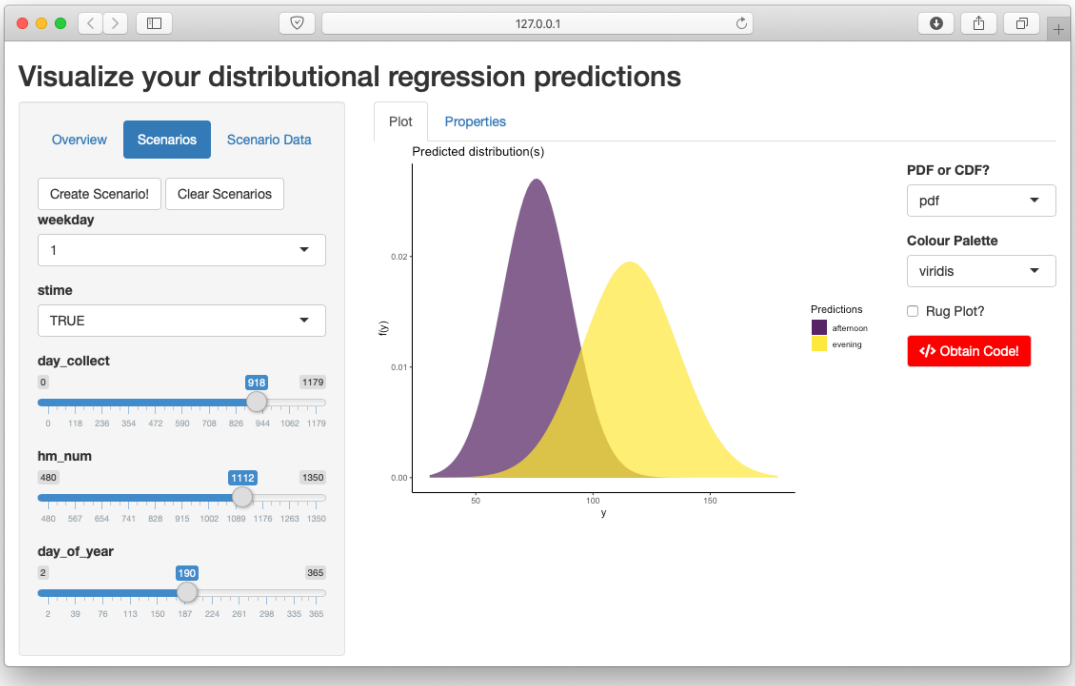

Figure 3.13: "Scenarios" tab of the `distreg.vis` GUI showing the predicted PDFs for the two scenarios detailed in Table 3.3. Different colors correspond to different scenarios.

After the two aforementioned scenarios were specified in the "Scenarios" tab (left plot side), the resulting expected distributions are displayed (right plot side). Figure 3.13 shows the result inside the GUI. We can see that there is a strong difference between the expected visitor number on a Monday afternoon and a Monday evening.

Using `distreg.vis`, one could now visually answer a wide range of interesting questions. For example, how does the expected distribution of gym visitors compare between a weekday inside of the lecture period (*stime* = TRUE) and outside one? Or how do distributions change between working days and on the weekend? Finding the answers to these questions is now possible with just a few clicks, although they will only partially be included here.

Seeing directly how the gym visitor distribution changes during the day is not straightforward using the "Plot" tab. More specifically, we would like to know the influence of *hm_num* on the expected parameters of the target distribution, holding other variables constant. Without `distreg.vis`, this would require many lines of code, since the censored normal distribution uses the parameters μ, σ that do not equate to

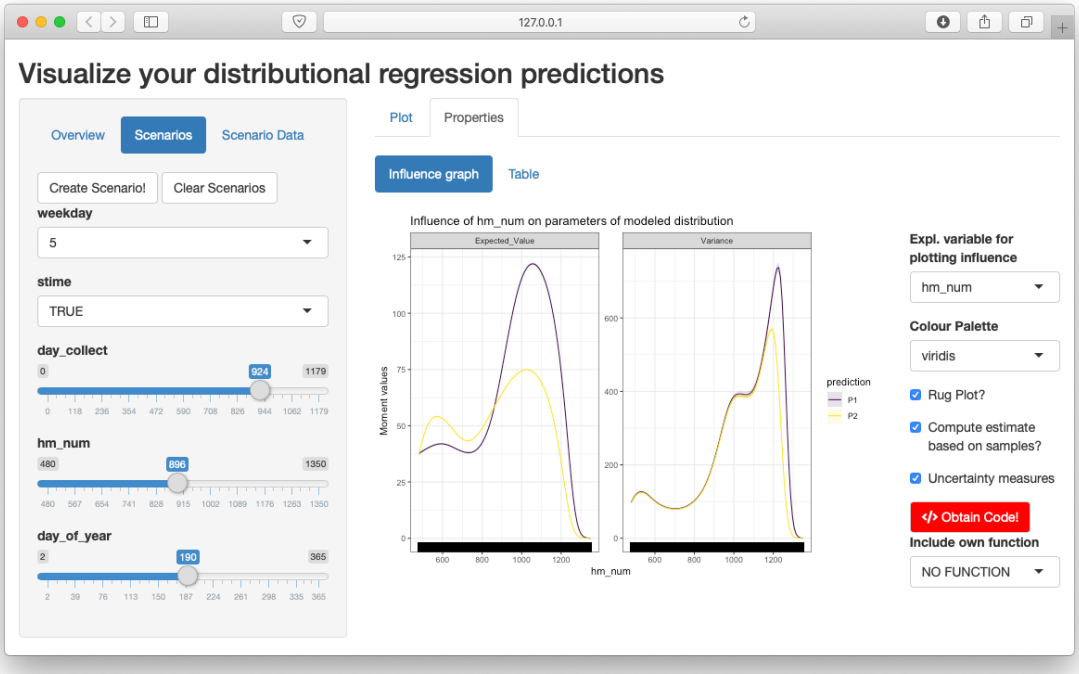

FIGURE 3.14: GUI of `distreg.vis` detailing the influence of `hm_num`, the numerical time, on the first two expected moments of the number of gym visitors. Credible intervals are included but hardly visible due to the high sample size.

the moments of the distribution. Now, we can just head to the "Influence graph" tab.

To visualize both the influence of time during the day as well as how this effect varies between the days, we specify a scenario which compares a Monday in July to the following Friday. Table 3.4 shows the newly specified covariate combinations.

Scenario	weekday	stime	day_collect	hm_num	day_of_year
monday	1	TRUE	919.00	896.00	190
friday	5	TRUE	924.00	896.00	195

TABLE 3.4: Covariate values corresponding to Figure 3.14, in which a Monday is compared to a Friday.

Following the specification of scenarios, Figure 3.14 shows the state of `distreg.vis`' GUI "Properties" tab, where `hm_num` was selected as the variable of interest. Furthermore, we ticked the options for estimating a rug plot and sample-based estimation including the display of its credible intervals.

As visible in Figure 3.14, the expected gym attendance for Monday and Friday substantially differs. From the time when the gym opens ($hm\_num = 480$, 08:00)

to around noon ($hm\_num = 720$) both on Monday and Friday a peak is expected, whereas the peak for Friday is significantly higher. After a dip at noon, both Monday and Friday exhibit the highest daily expected visitor numbers in the evening, around 6 p.m. ($hm\_num = 720$). The pattern of the morning is reversed here - significantly more people are expected on Monday than on Friday. Afterwards the expected attendance on both days quickly diminishes, though this pattern occurs about half an hour earlier on Friday.

Even though we included the option to draw credible intervals, it is hard to see them on the `Expected_Value` graph with the naked eye. This is due to them being narrow; a result of the high sample size. The modeled variance on the right side of the graph does have visible credible intervals. We see that the pattern of both Monday and Friday is similar - the distribution variance is lowest in the morning before noon, and then rises until its peak around 20:00 ($hm\_num = 1200$). This might be due to the gym visitor numbers being highly dependent on other evening activities, like popular football games.

Beyond the previous analysis, we would also like to know how the expected attendance changes between days within the lecture period and days within the semester breaks. To display this graphically, we now just have to select the right variable, `stime`.

Figure 3.15 shows the resulting graph. Based on the previously selected covariate combination (Table 3.4), we see the expected value and the variance broken down into all categories of `stime`. The results show that the expected number of gym visitors is higher in the lecture period than outside it, which is an expected result. Outside of the lecture period many students are travelling or undertaking internships, leading to a lower amount of students in the city. On the right side of the plot, the variance stays constant within the lecture period and outside it, which originates from the model specification.

Both the results of Figure 3.14 and Figure 3.15 can be easily reproduced by using the dynamic code that is provided when the "Obtain Code" button is pressed. Due to all graphical results being based on the R package `ggplot2` (Wickham, 2016), one can change the aesthetic quickly. For example, the theme of the graph can be changed to

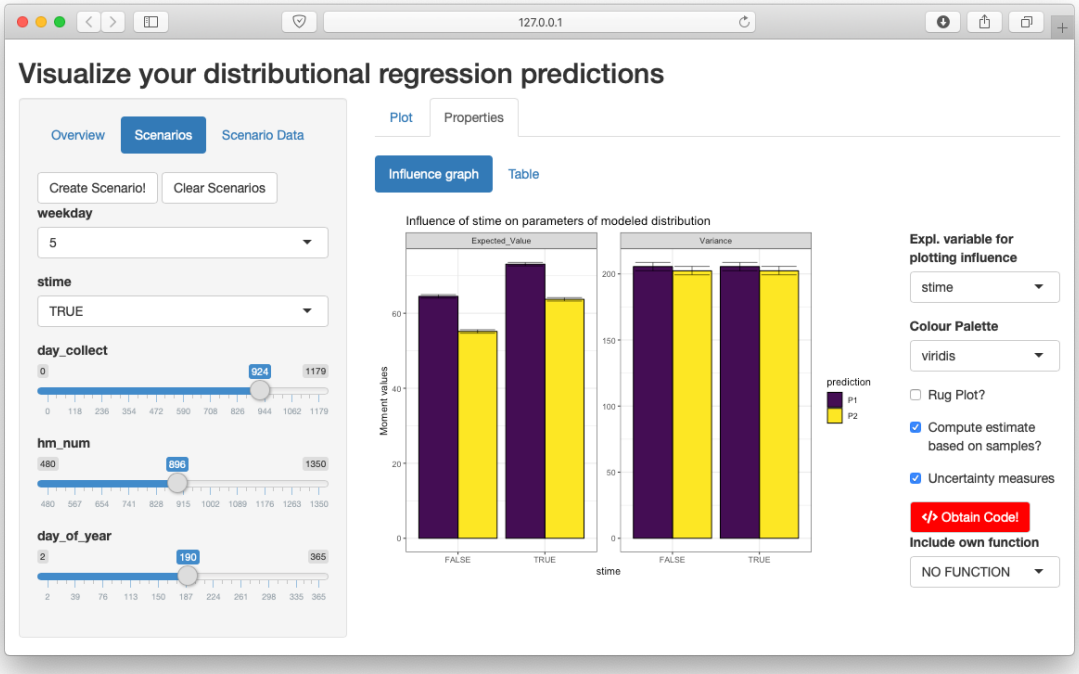

FIGURE 3.15: State of the `distreg.vis` GUI when variable `stime`, detailing whether the given gym attendance day falls within the lecture period or not, is selected as the variable of interest.

a more classic look with adding `+ theme_classic()` (a function from `ggplot2`) to the obtained function call. Furthermore, user-subjective information can be added, like the opening hours of the gym or the meanings behind the variable names.

3.7 Discussion

This paper laid out the foundations of distributional regression and its implementations in R. It became clear that distributional regression provides a highly flexible modeling framework, from which researchers will increasingly benefit in the future. To make fitted models more accessible for users, this paper introduced the software `distreg.vis` based on the `bamlss`, `gamlss` and `betareg` R packages.

Using `distreg.vis`, the user can visualize predicted response distributions based on interactively chosen covariate combinations. Furthermore, `distreg.vis` provides the ability to show the influence of a selected explanatory covariate on the first two moments of any response variable, including continuous and discrete distributions.

Both of these key functions help the user draw useful inference from fitted Bayesian and frequentist distributional regression models.

Moreover, `distreg.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). Following the finished analysis, the user can obtain R code to reproduce the displayed plot with all chosen options.

With this new tool at hand, users will hopefully be more encouraged to apply distributional regression models and better able to draw resilient conclusions. Nevertheless, further work on making distributional regression models more accessible can be done. The idea of visualizing predicted distributions and making influence plots could also be applied to other variations of distributional regression, for example BayesX (Brezger et al., 2005), VGAM (Yee, 2015) or GJRM (Marra & Radice, 2017), which offer a large variety of supported distributions.

4

Parameter orthogonality transformations in distributional regression models

4.1 Introduction

The distributional regression framework, first proposed as generalized additive models for location, scale and shape (GAMLSS, Rigby & Stasinopoulos, 2005) by Rigby and Stasinopoulos (2005), represents a highly flexible modeling environment. It extends generalized additive models (GAM, Hastie & Tibshirani, 1990) by providing the ability to connect distributional parameters beyond the mean with structured additive model terms (fixed, random, spatial, smooth, etc.) using known link functions (log, logit, etc.) while supporting a large variety of response distributions including those outside the exponential family. The term "distributional regression" was later coined to reflect

the fact that not all distributional parameters are either of location, scale or shape.

In most distributions beyond the exponential family, the distributional parameters which are to be connected to explanatory covariates have maximum-likelihood estimates (MLEs) that are asymptotically correlated (non-orthogonal) to varying degrees. While this phenomenon is not severely problematic in the case of constant parameters, adding predictors to the model causes a bias if its model equations are not correctly specified. Although model misspecification is well known and studied (see e.g. Zimmerman, 2020), it is given a new layer in distributional regression since the estimated coefficients for one parameter are biased even if the misspecification occurs only in other parameters. This provides an interesting new perspective on mean-focused regression analysis, which typically neglects modeling distributional parameters beyond the mean.

To arrive at distributional parameters with independent model specifications, we propose a framework that finds a new and orthogonal parametrization of any two-parameter distribution by (a) solving a system of ordinary differential equations (ODEs) based on the covariance of the parameters' MLE and then (b) connecting the original parameters to the parameters obtained in this solution, which effectively removes the bias spillover from one parameter to the other. Due to its flexibility, our framework can be applied to all two-parameter distributions, even those whose MLEs' covariance matrix is not in closed form.

The issue of parameter non-orthogonality for marginal distributions (no covariates) as well as within a regression-type setting has received attention over a considerable timespan. Jeffreys (1948) first recognized the importance of parameter orthogonality for finding estimates more quickly. Based on this finding, Huzurbazar (1950) proposed a framework for finding orthogonal parametrizations based on solutions of an ODE in two-parametric probability density functions (PDFs). More recently, Stein, Zucchini, and Juritz (1987) found strong parameter non-orthogonality to lead to inaccurate estimation and proposed an orthogonal parametrization of the Sichel distribution. Raising orthogonality for the first time as an issue in regression analysis, Heller, Couturier, and Heritier (2019) showed that parameter non-orthogonality can induce a bias in estimated

regression coefficients: with a Poisson-inverse Gaussian distributed response variable, the misspecification of the scale parameter led to a false non-significance of a medical treatment effect. Even though most of the above literature proposes solutions to the non-orthogonality problem, they require closed form solutions of formulas relying on the parameter MLEs' covariance matrix. Our proposition is applicable even in distributions where the MLE covariance matrix can only be approximated or estimated locally, or where the ODE proposed by Huzurbazar (1950) cannot be solved analytically.

The structure of this paper is as follows: Section 4.2 introduces the distributional regression model class. In Section 4.3, the extent of the non-orthogonality problem and its consequences are outlined. Section 4.4 will outline our proposed universal solution to establish orthogonality in two-parameter distributions while Section 4.5 confirms the effectiveness of our proposal. An application to extreme rainfall in Tasmania, Australia is given in Section 4.6, after which Section 4.7 concludes the paper.

4.2 Distributional regression

Distributional regression is an umbrella class for models in which it is possible to link not only the mean but also other parameters of a distribution to a set of additive predictors which can be parametric (fixed, random, spatial) or non-parametric (regression trees, local curves, splines) as outlined by Klein, Kneib, Klasen, and Lang (2015). All choices of response distribution are valid, as long as they are twice continuously differentiable with respect to their parameters. In particular, they are not limited to the exponential family of distributions used in generalized linear models (GLMs) and GAMs. Assuming a parametric response distribution with distributional parameters $\theta_1, \ldots, \theta_K$, i.e. $y \sim D(\theta_1, \ldots, \theta_K)$, the model specification is:

$$\begin{aligned} \eta_k &= \beta_{k0} + \sum_{q=1}^{Q_k} f_{kq}(\mathbf{x}_{kq}; \boldsymbol{\beta}_{kq}) \\ h_k(\eta_k) &= \theta_k, \end{aligned} \tag{4.1}$$

where $h_k(\cdot)$, $k = 1, \ldots, K$ denotes the response function used to uphold the support of parameter θ_k; $f_{kq}(\cdot)$, $q = 1, \ldots, Q_k$ represents the (possibly non-parametric) effect of covariate(s) $\mathbf{x}_{kq}$ on the modeled parameter; and $\boldsymbol{\beta}_{kq}$, $q = 1, \ldots, Q_k$ denotes the regression parameters which are to be estimated.

Due to the high level of flexibility in the response distributions, the predictors and the specified link function, distributional regression encompasses many well-known regression approaches, such as generalized linear models (GLM, Nelder & Wedderburn, 1972), GAM, generalized additive mixed models (GAMM, Lin & Zhang, 1999), Bayesian additive models for location, scale and shape (BAMLSS, Umlauf et al., 2018) and GAMLSS. Closely related to GAMLSS, the term distributional regression was coined by Klein, Kneib, Lang, and Sohn (2015) to overcome some of its earlier limitations and to reflect the fact that distributional parameters do not always represent either the location, scale or shape of a distribution.

4.3 Orthogonality

4.3.1 General case

Consider a two-parameter distribution $y \sim D(\theta_1, \theta_2)$ with PDF $p(y|\theta_1, \theta_2)$ and log-likelihood function $\ell(\theta_1, \theta_2|y)$. It is well known (Millar, 2011) that the MLEs $\hat{\boldsymbol{\theta}} = (\hat{\theta}_1, \hat{\theta}_2)^\top$ are, under regularity conditions, asymptotically distributed as

$$\hat{\boldsymbol{\theta}} \overset{a}{\sim} N\left(\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix}, \mathcal{I}^{-1}\right), \tag{4.2}$$

where $\mathcal{I}$ (the information matrix) is defined as

$$\mathcal{I} = -\mathbb{E}\begin{bmatrix}\frac{\partial^2 \ell(\theta_1,\theta_2|y)}{\partial \theta_1^2} & \frac{\partial^2 \ell(\theta_1,\theta_2|y)}{\partial \theta_1 \partial \theta_2}\\ \frac{\partial^2 \ell(\theta_1,\theta_2|y)}{\partial \theta_1 \partial \theta_2} & \frac{\partial^2 \ell(\theta_1,\theta_2|y)}{\partial \theta_2^2}\end{bmatrix} = -\begin{bmatrix}\mathbb{E}_{11} & \mathbb{E}_{12}\\ \mathbb{E}_{12} & \mathbb{E}_{22}\end{bmatrix}. \tag{4.3}$$

The asymptotic correlation between $\hat{\theta}_1$ and $\hat{\theta}_2$ is

$$\text{Corr}(\hat{\theta}_1, \hat{\theta}_2) = \frac{\mathbb{E}_{12}}{\sqrt{\mathbb{E}_{11}\mathbb{E}_{22}}} \, . \tag{4.4}$$

$\mathbb{E}_{12} = 0$ yields $\text{Corr}(\hat{\theta}_1, \hat{\theta}_2) = 0$, and in this case the parameters $\boldsymbol{\theta} = (\theta_1, \theta_2)^\top$ are called orthogonal.

4.3.2 Orthogonality in regression analysis

When optimizing the log-likelihood function with respect to only the distributional parameters, non-orthogonality of the parameters is not strongly problematic, but merely slows down the optimization process (Huzurbazar, 1950). However, in situations where we assume the parameters to have a connection to explanatory variables, we show that misspecification of the regression equation for one parameter has an impact on the estimated regression coefficients of the other parameter, even if that other parameter is correctly specified. This is highly relevant in situations where the applied researcher is interested in only one parameter of the distribution, typically location, and does not take into account possible effects on other parameters. Note that distributions which are members of the exponential family have orthogonal parameters (Barndorff-Nielsen, 1978, p. 111 for exponential family definition, p. 183 for orthogonality statement). Therefore, GLMs and GAMs are not subject to the problem of non-orthogonality.

As an illustrative example, we consider a target variable y following an inverse gamma distribution. We use the following, non-exponential family parametrization, (Stasinopoulos et al., 2017):

$$p(y|\theta_1, \theta_2) = \frac{\theta_2^{\theta_1}}{\Gamma(\theta_1)} \cdot y^{-\theta_1 - 1} \cdot \exp\left(-\frac{\theta_2}{y}\right), \quad \theta_1, \theta_2 > 0 \text{ and } y > 0 \, . \tag{4.5}$$

We define the second parameter θ_2 as being dependent on a covariate x, while θ_1 is

constant:

$$\theta_1 = \beta_0$$
$$\theta_2 = \beta_1\, x.$$

Further, we derive the analytical MLE using the correct model specification. It uses an approximate inverse of the digamma function $\psi(\cdot)$ (Batir, 2018) and is therefore an approximate result:

$$\hat{\boldsymbol{\beta}}_{\mathrm{ML}}^{\mathrm{correct}} \approx \begin{bmatrix} \left(\log(1+\exp(-\frac{1}{n}\sum_{i=1}^{n}\log(\beta_1\, x_i) + \frac{1}{n}\sum_{i=1}^{n}\log(y_i)))\right)^{-1} \\ n\beta_0 \left(\sum_{i=1}^{n}\frac{x_i}{y_i}\right)^{-1} \end{bmatrix}. \tag{4.6}$$

Note that x appears in the equation for $\hat{\theta}_1$, even though it is not in its model equation. If we now incorrectly assume that θ_2 is not dependent on x, the MLE assumes a different form:

$$\hat{\boldsymbol{\beta}}_{\mathrm{ML}}^{\mathrm{incorrect}} \approx \begin{bmatrix} \left(\log(1+\exp(-\frac{1}{n}\sum_{i=1}^{n}\log(\beta_1) + \frac{1}{n}\sum_{i=1}^{n}\log(y_i)))\right)^{-1} \\ n\beta_0 \left(\sum_{i=1}^{n}\frac{1}{y_i}\right)^{-1} \end{bmatrix}. \tag{4.7}$$

Comparing these estimators, we can see that the information of x is missing in both elements of $\hat{\boldsymbol{\beta}}_{\mathrm{ML}}^{\mathrm{incorrect}}$.

Figure 4.1 shows the results of a simulation in which we estimated the correct and incorrect MLEs for a grid of true coefficients $\boldsymbol{\beta}_0 \otimes \boldsymbol{\beta}_1$, where $\boldsymbol{\beta}_0 = (0.5, \ldots, 10)^\intercal$ and $\boldsymbol{\beta}_1 = (1, \ldots, 5)^\intercal$, and then computed the mean absolute bias. In the second row of plots the parameters were estimated omitting the information in x analogous to (4.7). It is clear that the first parameter, even though correctly specified, exhibits an increasingly strong bias as β_1, the influence of x on θ_2, increases.

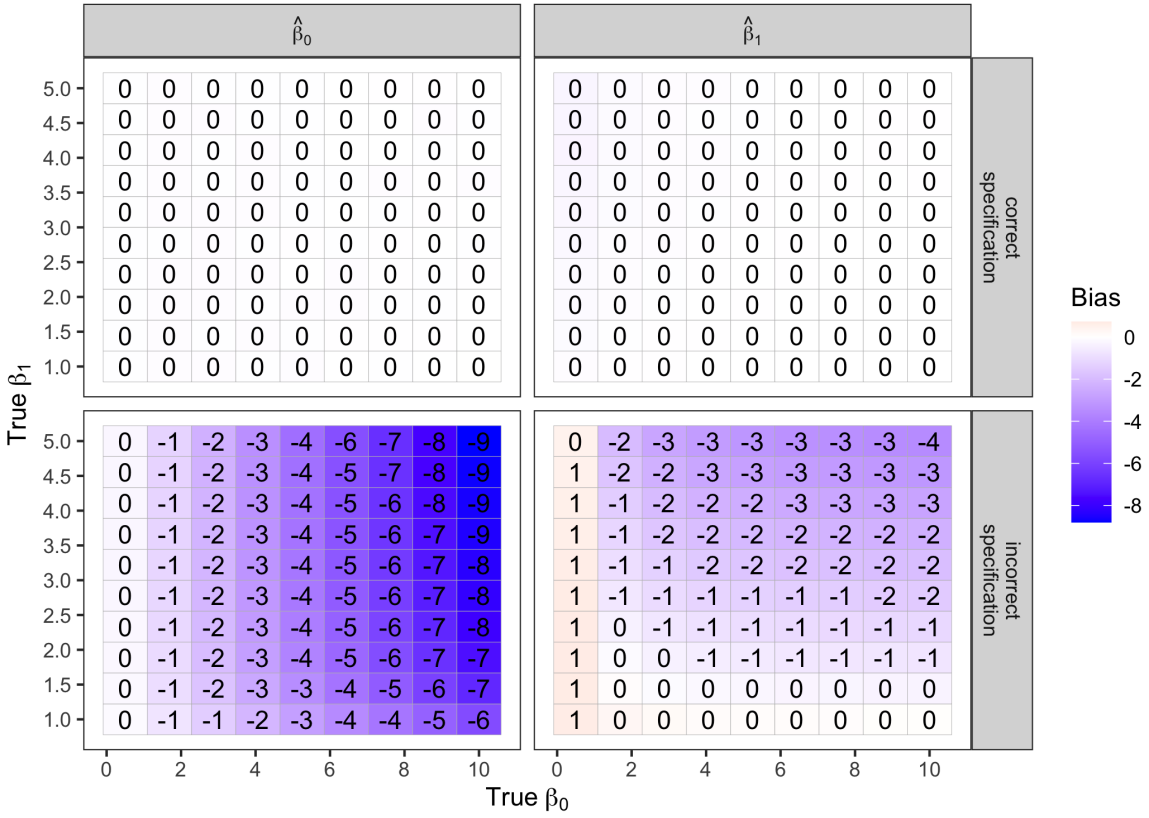

FIGURE 4.1: Simulation study depicting the mean absolute bias of β_0 and β_1, if β_1 is correctly and incorrectly specified. The x- and y-axes represent the true values of β_0, β_1. The columns depict the estimators $\hat{\beta}_0$ and $\hat{\beta}_1$, whereas the rows specify whether the information of x was included (`correct specification`, top row) or omitted (`incorrect specification`, bottom row). $n = 500$.

4.4 Establishing orthogonality

In this section, we develop a solution for establishing orthogonality in two-parameter distributions with non-orthogonal parameters.

Our reparametrization framework relies heavily on the findings of Huzurbazar (1950), who drew his motivation for orthogonalization from the difficulty of obtaining maximum likelihood optimization iterations for highly correlated estimators in an era when scientific computing power was limited. From (4.4), the parameters $\boldsymbol{\theta}$ of a two-parameter PDF are orthogonal if and only if the following statement holds true:

$$\mathbb{E}_{12} = \mathbb{E}\left(\frac{\partial^2}{\partial\theta_1\partial\theta_2}\log p\right) = 0\ . \tag{4.8}$$

This is not always the case for distributions outside the exponential family. We may reparametrize by considering a new parametrization $p(y|\omega_1, \omega_2)$, where

$$\omega_1 = g_1(\theta_1, \theta_2)$$
$$\omega_2 = g_2(\theta_1, \theta_2)$$

and $g_1(\cdot)$ and $g_2(\cdot)$ are functions to be chosen such that

$$\mathbb{E}\left(\frac{\partial^2}{\partial\omega_1\partial\omega_2}\log p\right) = 0\ . \tag{4.9}$$

The method of Huzurbazar (1950) is to retain the first parameter (usually location), i.e. set $\omega_1 = \theta_1$, and choose ω_2 such that (4.9) is satisfied. This leads to an ODE, the solution of which yields the new, orthogonal parameters:

$$\mathbb{E}_{12} + \mathbb{E}_{22}\frac{\partial\theta_2}{\partial\theta_1} = 0. \tag{4.10}$$

4.4.1 Derivation of the orthogonality statement

We derive the statement 4.10 here. First, the cross-derivative of the likelihood function with respect to the new parameters is obtained. Huzurbazar (1950) applies the chain rule and shows the following equation:

$$\frac{\partial^2}{\partial\omega_k\partial\omega_l}\log p = \frac{\partial\theta_i}{\partial\omega_k}\frac{\partial\theta_j}{\partial\omega_l}\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log p + \frac{\partial^2\theta_i}{\partial\omega_k\partial\omega_l}\frac{\partial}{\partial\theta_i}\log p, \tag{4.11}$$

while stating: "The summation convention is used with respect to i and j." What he implies is: the first term is summed over i and j; the second term is summed just

over i. For ω_1 ($k=1$) and ω_2 ($l=2$), this is:

$$\begin{aligned}\frac{\partial^2}{\partial\omega_1\partial\omega_2}\log p &= \sum_{i=1}^{2}\sum_{j=1}^{2}\frac{\partial\theta_i}{\partial\omega_1}\frac{\partial\theta_j}{\partial\omega_2}\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log p + \sum_{i=1}^{2}\frac{\partial^2\theta_i}{\partial\omega_1\partial\omega_2}\frac{\partial}{\partial\theta_i}\log p \\ &= \frac{\partial\theta_1}{\partial\omega_1}\frac{\partial\theta_1}{\partial\omega_2}\frac{\partial^2}{\partial\theta_1^2}\log p + \frac{\partial^2\theta_1}{\partial\omega_1\partial\omega_2}\frac{\partial}{\partial\theta_1}\log p && i=1, j=1 \\ &+ \frac{\partial\theta_1}{\partial\omega_1}\frac{\partial\theta_2}{\partial\omega_2}\frac{\partial^2}{\partial\theta_1\partial\theta_2}\log p && i=1, j=2 \\ &+ \frac{\partial\theta_2}{\partial\omega_1}\frac{\partial\theta_1}{\partial\omega_2}\frac{\partial^2}{\partial\theta_2\partial\theta_1}\log p + \frac{\partial^2\theta_2}{\partial\omega_1\partial\omega_2}\frac{\partial}{\partial\theta_2}\log p && i=2, j=1 \\ &+ \frac{\partial\theta_2}{\partial\omega_1}\frac{\partial\theta_2}{\partial\omega_2}\frac{\partial^2}{\partial\theta_2^2}\log p. && i=2, j=2\end{aligned} \tag{4.12}$$

Using (4.12), the covariance of $\hat{\omega}_1, \hat{\omega}_2$ can be obtained as follows:

$$\begin{aligned}\mathrm{Cov}(\hat{\omega}_1, \hat{\omega}_2) &= \mathbb{E}\left(\frac{\partial^2\ell}{\partial\omega_1\,\partial\omega_2}\right) \\ &= \mathbb{E}\left(\frac{\partial^2\ell}{\partial\theta_1^2}\right)\cdot\frac{\partial\theta_1}{\partial\omega_1}\cdot\frac{\partial\theta_1}{\partial\omega_2} \\ &+ \mathbb{E}\left(\frac{\partial^2\ell}{\partial\theta_2^2}\right)\cdot\frac{\partial\theta_2}{\partial\omega_1}\cdot\frac{\partial\theta_2}{\partial\omega_2} \\ &+ \mathbb{E}\left(\frac{\partial^2\ell}{\partial\theta_1\,\partial\theta_2}\right)\cdot\left(\frac{\partial\theta_1}{\partial\omega_1}\cdot\frac{\partial\theta_2}{\partial\omega_2} + \frac{\partial\theta_1}{\partial\omega_2}\cdot\frac{\partial\theta_2}{\partial\omega_1}\right),\end{aligned} \tag{4.13}$$

since $\mathbb{E}\left(\frac{\partial\ell}{\partial\theta_1}\right) = \mathbb{E}\left(\frac{\partial\ell}{\partial\theta_2}\right) = 0$.

In the case where we retain θ_1, i.e. $\omega_1 = \theta_1$, we have

$$\begin{aligned}\theta_1 &= \omega_1 \\ \frac{\partial\theta_1}{\partial\omega_1} &= 1 \\ \frac{\partial\theta_1}{\partial\omega_2} &= 0 \\ \frac{\partial\theta_2}{\partial\omega_1} &= \frac{\partial\theta_2}{\partial\theta_1}\end{aligned}$$

and

$$\mathbb{E}\left(\frac{\partial^2 \ell}{\partial\theta_1\,\partial\omega_2}\right) = \mathbb{E}\left(\frac{\partial^2 \ell}{\partial\theta_2^2}\right)\cdot\frac{\partial\theta_2}{\partial\theta_1}\cdot\frac{\partial\theta_2}{\partial\omega_2} + \mathbb{E}\left(\frac{\partial^2 \ell}{\partial\theta_1\,\partial\theta_2}\right)\cdot\frac{\partial\theta_2}{\partial\omega_2}. \tag{4.14}$$

Setting (4.14) to zero since the covariance should be zero, we get

$$\begin{aligned} &\mathbb{E}\left(\frac{\partial^2 \ell}{\partial\theta_2^2}\right)\cdot\frac{\partial\theta_2}{\partial\theta_1}\cdot\frac{\partial\theta_2}{\partial\omega_2} + \mathbb{E}\left(\frac{\partial^2 \ell}{\partial\theta_1\,\partial\theta_2}\right)\cdot\frac{\partial\theta_2}{\partial\omega_2} = 0 \\ \Rightarrow\quad &\mathbb{E}\left(\frac{\partial^2 \ell}{\partial\theta_2^2}\right)\cdot\frac{\partial\theta_2}{\partial\theta_1} + \mathbb{E}\left(\frac{\partial^2 \ell}{\partial\theta_1\,\partial\theta_2}\right) = 0, \end{aligned} \tag{4.15}$$

which equals the ODE stated in 4.10 (Huzurbazar, 1950):

$$\mathbb{E}_{12} + \mathbb{E}_{22}\frac{\partial\theta_2}{\partial\theta_1} = 0. \tag{4.16}$$

Computationally, using (4.10) in the form of a partial differential equation (PDE) is advantageous. We therefore introduce ω_2 to rewrite it as follows:

$$\mathbb{E}\left(-\frac{\partial^2}{\partial\omega_1\partial\omega_2}\log p\right) = \mathbb{E}_{12}\frac{\partial\omega_2}{\partial\theta_2} + \mathbb{E}_{22}\frac{\partial\omega_2}{\partial\theta_1} = 0. \tag{4.17}$$

4.4.2 Invertibility

The simplified orthogonality condition in (4.17), when solved, yields a result for $g_2(\cdot)$:

$$\omega_1 = \theta_1, \quad \omega_2 = g_2(\theta_1, \theta_2),$$

which needs to be inverted if we want to express the PDF in terms of ω_1 and ω_2:

$$\theta_1 = \omega_1, \quad \theta_2 = s(\omega_1, \omega_2)\ . \tag{4.18}$$

The existence of a closed form solution of an orthogonal PDF therefore depends on closed-form solutions of $\mathbb{E}_{ij}$ (4.3), the PDE (4.17) and its invertibility (4.18).

4.4.3 The inverse gamma distribution

We apply Huzurbazar's (1950) method to the inverse gamma distribution. Using the parametrization specified in (4.5), we obtain

$$\mathbb{E}_{12} = \mathbb{E}\left(-\frac{1}{\theta_2}\right) = -\frac{1}{\theta_2}$$
$$\mathbb{E}_{22} = \mathbb{E}\left(-\left[-\frac{\theta_1}{\theta_2^2}\right]\right) = \frac{\theta_1}{\theta_2^2},$$

which we deploy in (4.17). We start with the PDF, described in (4.5). The likelihood function can be written as:

$$L(\theta_1, \theta_2) = \frac{\theta_2^{\theta_1 \cdot n}}{\Gamma(\theta_1)^n} \cdot \prod_{i=1}^{n} (y_i^{-\theta_1 - 1}) \cdot \exp(-\sum_{i=1}^{n} \frac{\theta_2}{y_i}). \tag{4.19}$$

Then, the log-likelihood function is:

$$l(\theta_1, \theta_2) = \theta_1 \cdot n \cdot \log(\theta_2) - n \cdot \log(\Gamma(\theta_1)) \tag{4.20}$$
$$- \theta_1 \cdot \sum_{i=1}^{n} \log(y_i) - \sum_{i=1}^{n} \log(y_i) - \sum_{i=1}^{n} \frac{\theta_2}{y_i}. \tag{4.21}$$

Differentiating the log-likelihood function leads to the score function:

$$\text{score}\begin{pmatrix}\theta_1 \\ \theta_2\end{pmatrix} = \begin{pmatrix}\frac{\partial l(\theta_1,\theta_2)}{\partial \theta_1} \\ \frac{\partial l(\theta_1,\theta_2)}{\partial \theta_2}\end{pmatrix} = \begin{pmatrix} n \cdot \log(\theta_2) - n \cdot \psi(\theta_1) - \sum_{i=1}^{n} \log(y_i) \\ \frac{\theta_1 \cdot n}{\theta_2} - \sum_{i=1}^{n} \frac{1}{y_i}, \end{pmatrix} \tag{4.22}$$

where $\psi(\cdot)$ represents the digamma function. Further deriving the score function leads to the observed Fisher information:

$$F\begin{pmatrix}\theta_1 \\ \theta_2\end{pmatrix} = -\begin{pmatrix}\frac{\partial^2 l(\theta_1,\theta_2)}{\partial^2 \theta_1} & \frac{\partial^2 l(\theta_1,\theta_2)}{\partial \theta_1 \partial \theta_2} \\ \frac{\partial^2 l(\theta_1,\theta_2)}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 l(\theta_1,\theta_2)}{\partial^2 \theta_2}\end{pmatrix} = \begin{pmatrix} n \cdot \psi_1(\theta_1) & -\frac{n}{\theta_2} \\ -\frac{n}{\theta_2} & \frac{\theta_1 \cdot n}{\theta_2^2}\end{pmatrix}, \tag{4.23}$$

where $F(\cdot)$ is also the expected Fisher information $F^*(\cdot)$ and $\psi_1(\cdot)$ represents the trigamma function (second derivative of log-gamma function). For it to represent the covariance matrix of the MLEs, we have to invert $F^*(\cdot)$. First, we calculate the

determinant of this matrix:

$$det(F^* \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}) = \frac{n^2(\psi_1(\theta_1) \cdot \theta_1 - 1)}{\theta_2^2}. \tag{4.24}$$

Then, we obtain the inverse expected Fisher information, which equals the covariance matrix of the parameter MLEs:

$$F^* \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}^{-1} = \begin{pmatrix} \frac{\theta_1}{n(\psi_1(\theta_1)\cdot\theta_1-1)} & \frac{\theta_2}{n(\psi_1(\theta_1)\cdot\theta_1-1)} \\ \frac{\theta_2}{n(\psi_1(\theta_1)\cdot\theta_1-1)} & \frac{\psi_1(\theta_1)\cdot\theta_2^2}{n(\psi_1(\theta_1)\cdot\theta_1-1)} \end{pmatrix}. \tag{4.25}$$

From Huzurbazar (1950) we know that in order to orthogonalize distributional parameters, we have to solve the ODE in (4.10). Applying it to the inverse gamma distribution, we obtain the following values:

$$\begin{aligned} \mathbb{E}_{12} &= \mathbb{E}(-\frac{1}{\theta_2}) = -\frac{1}{\theta_2} \\ \mathbb{E}_{22} &= \mathbb{E}(-(-\frac{\theta_1}{\theta_2^2})) = \frac{\theta_1}{\theta_2^2}. \end{aligned}$$

This means we have to solve the following differential equation:

$$-\frac{1}{\theta_2} + \frac{\theta_1}{\theta_2^2} \cdot \frac{\partial\theta_2}{\partial\theta_1} = 0, \tag{4.26}$$

which we can solve analytically:

$$\begin{aligned} \frac{\partial\theta_2}{\partial\theta_1} &= \frac{\theta_2}{\theta_1} \\ \theta_2 &= \omega_2 \cdot \theta_1, \end{aligned}$$

which, since we set $\theta_1 = \omega_1$ is

$$\theta_2 = \omega_2 \cdot \omega_1. \tag{4.27}$$

Substituting (4.27) into (4.5) leads to the following PDF:

$$p(y|\omega_1,\omega_2) = \frac{(\omega_1\omega_2)^{\omega_1}}{\Gamma(\omega_1)} \cdot x^{-\omega_1-1} \cdot \exp\left(-\frac{\omega_1\omega_2}{y}\right), \quad \omega_1,\omega_2,y>0, \tag{4.28}$$

where ω_1 and ω_2 are orthogonal parameters. We analytically check the orthogonality statement in (4.8) to confirm our new parametrization:

$$\mathbb{E}\left(\frac{\partial^2 \log p}{\partial\omega_1\partial\omega_2}\right) = \frac{1}{\omega_2} - \mathbb{E}\left(\frac{1}{y}\right) = \frac{1}{\omega_2} - \frac{\omega_1}{\omega_1\cdot\omega_2} = 0\ , \tag{4.29}$$

where $\mathbb{E}\left(\frac{1}{y}\right) = \frac{\theta_1}{\theta_2} = \frac{\omega_1}{\omega_1\omega_2}$. Note that this parametrization of the inverse gamma distribution is part of the exponential family (in non-canonical form), which explains why it has orthogonal parameters. A short derivation is given in A.2.2.

4.4.4 Beyond analytic solutions

It is easy to find non-exponential family distributions with non-orthogonal parameters for which Equation (4.17) cannot be solved analytically. To illustrate, we again use the inverse gamma distribution, but with the location/scale parametrization more widely used in regression analysis (Rigby, Stasinopoulos, Heller, & De Bastiani, 2019):

$$\begin{gathered} p(y|\theta_1,\theta_2) = \frac{\theta_1^{\gamma}(\gamma+1)^{\gamma}y^{-(\gamma+1)}}{\Gamma(\gamma)}\exp\left[-\frac{\theta_1(\gamma+1)}{y}\right] \quad \text{with} \\ \gamma = \frac{1}{\theta_2^2}, \quad \theta_1,\theta_2>0, \quad y>0. \end{gathered} \tag{4.30}$$

The parametrization of (4.30) has the convenient property that θ_1 is a location parameter (the mode), making it suitable for regression analysis. The second parameter θ_2 changes the scale of the distribution. To obtain an orthogonal PDF while preserving this property, we use the method outlined by Huzurbazar (1950) and derive $\mathbb{E}_{12}$ and $\mathbb{E}_{22}$ as in (4.3), to set up the ODE on the basis of (4.10). The result of this procedure,

derived in Section A.2.1 in the Appendix, has the following form:

$$\begin{aligned}
\mathbb{E}_{12} &= -\frac{2}{\theta_2^3} \cdot \left(\frac{1}{\theta_1} + \frac{1}{\theta_2^2 \theta_1} \right) + \frac{2}{\theta_1 \theta_2^3} \\
\mathbb{E}_{22} &= \frac{6 \left(\log(\theta_1) - \log\left(\frac{1}{\theta_2^2} + 1 \right) \right)}{\theta_2^4} + \frac{10}{\theta_2^6} + \frac{4}{\left(\frac{1}{\theta_2^2} + 1 \right)^2 \theta_2^8} - \frac{14}{\left(\frac{1}{\theta_2^2} + 1 \right) \theta_2^6} \\
&\quad + \frac{6}{\theta_2^4} + \frac{6}{\theta_2^2} + \frac{6 \log(\theta_1)}{\theta_2^4} + \frac{6 \log\left(\frac{1}{\theta_2^2} + 1 \right)}{\theta_2^4} + \frac{6 \psi\left(\frac{1}{\theta_2^2} \right)}{\theta_2^4} \\
0 &= \mathbb{E}_{12} + \mathbb{E}_{22} \frac{\partial \theta_2}{\partial \theta_1} ,
\end{aligned} \tag{4.31}$$

where the last line represents a re-arranged version of (4.17). The resulting ODE is highly non-linear and not separable. Further, even if we were able to obtain an analytical solution, it might not be invertible, despite the necessity to express $\theta_2 = s(\omega_1, \omega_2)$. There would be added difficulties if the MLE covariance matrix (4.2) were not available in closed form, which is relevant in distributions without moments, for example the Cauchy distribution (N. L. Johnson, Kotz, & Balakrishnan, 1995, p. 167). However, this does not apply in this case.

Resolving these situations, should they arise, is possible with the use of numeric solution of the PDE in (4.17). This paper focuses on the use of the `R` package `deSolve` (Soetaert, Petzoldt, & Setzer, 2010) for this purpose. We avoid the numerical difficulties with $\mathbb{E}_{ij} \to 0$ by using the method of characteristics (Olver, 2013) to reformulate Equation (4.17) into a coupled system of ODEs:

$$\frac{\partial \theta_2}{\partial \tau} = \mathbb{E}_{12}, \quad \frac{\partial \theta_1}{\partial \tau} = \mathbb{E}_{22}, \quad \frac{\partial \omega_2}{\partial \tau} = 0 \ . \tag{4.32}$$

The first two of the above equations describe a path, parameterized by τ, in the $\theta_1 - \theta_2$ plane, while the last equation tells us how ω_2 changes along this path. Since $\frac{\partial \omega_2}{\partial \tau} = 0$, ω_2 is constant along these parametrized paths (called the characteristic paths). Hence if we specify a different ω_2 on each characteristic path, we can completely specify the coordinate transformation from $(\theta_1, \theta_2) \to (\omega_1 = \theta_1, \omega_2)$ that orthogonalizes the system. We have to specify initial conditions for $\omega_2, \theta_2, \theta_1$; as these values can be arbitrary, we

specify a constant for θ_1 and $\omega_2 = \theta_2$ to compute θ_2 values. We continue varying θ_1 to get different values for θ_2. Each time we solve for one of the paths, in which each path's proposed length is determined by the range of τ. To ensure that our path is solving in the right area, we further specify what ranges we would like to have our original parameters θ_1, θ_2 solved in and then check whether this was achieved after we obtain the solved path. If the maximum or minimum values of the ODE solutions are too far off, we re-adjust τ such that it falls within the pre-specified θ_1, θ_2 range.

4.4.5 ODE solution surface

After successfully solving the system of ODEs in (4.32), we obtain a number of combinations for the original parameters θ_1, θ_2 and ω_2 (θ_2's orthogonal counterpart). We display our results graphically in a three-dimensional scatterplot in Figure 4.2, from

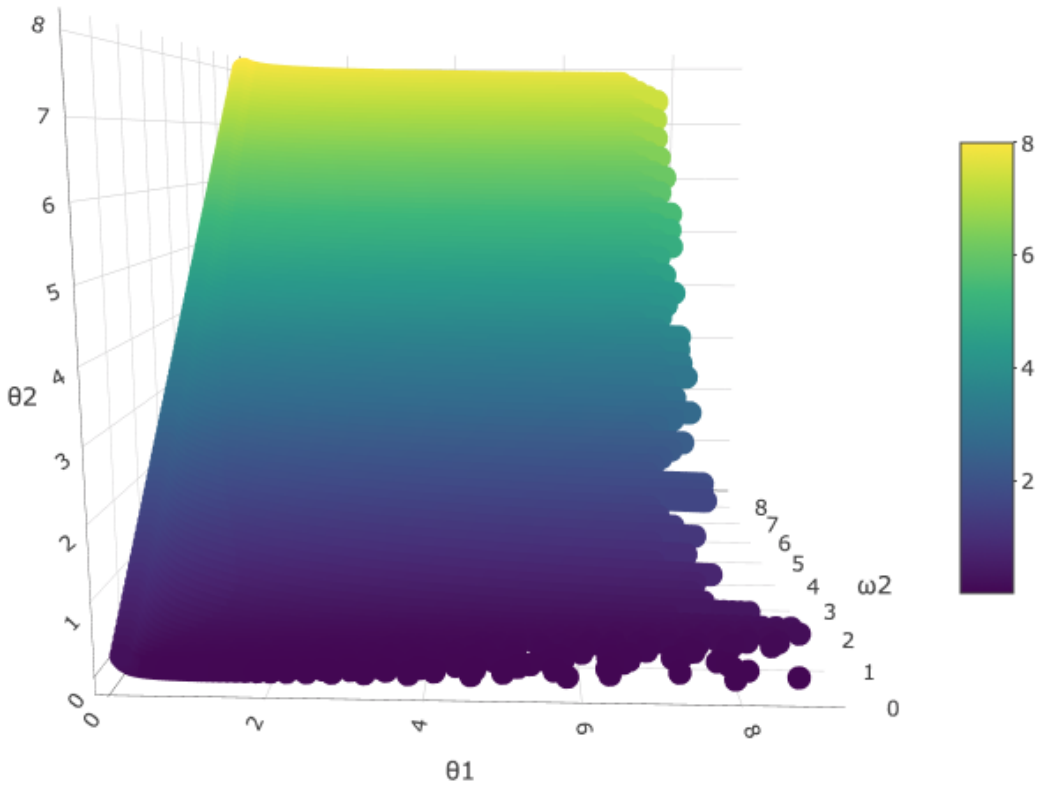

Figure 4.2: ODE system solution surface for parameters $\theta_1, \theta_2, \omega_2$ of the location-scale parametrization of the inverse gamma distribution.

which we are able to visually confirm the existence of unique solutions for each $\theta_1, \theta_2, \omega_2$

combination.

Since we are interested in finding a PDF that we can reparametrize with new, orthogonal parameters ω_1, ω_2 while still retaining the property $\theta_1 = \omega_1$, we estimate the surface given in Figure 4.2, connecting θ_2 to the orthogonal parameters:

$$\hat{\theta}_2 = \hat{s}(\theta_1, \omega_2) \tag{4.33}$$

To do so, we use a tensor product spline surface (Wood, 2006) as implemented in the `R` package `mgcv` (Wood, 2011). The estimated PDF replaces θ_2 with the estimate given by (4.33).

4.4.6 The new probability density function

The new PDF has the original parameters θ_1, θ_2 replaced by orthogonal parameters ω_1, ω_2. Since the first parameter stays the same, we are only replacing θ_2 with its value on the newly fitted surface in (4.33). Applied to the PDF introduced in (4.30), its transformed version has the following form:

$$\begin{aligned} p(y \mid \omega_1, \omega_2) &= \frac{\omega_1^{\gamma}(\gamma+1)^{\gamma} y^{-(\gamma+1)}}{\Gamma(\gamma)} \exp\left(-\frac{\omega_1 \cdot (\gamma+1)}{y}\right), \\ &\text{where } \gamma = \frac{1}{\hat{s}(\omega_1, \omega_2)^2} \ . \end{aligned} \tag{4.34}$$

Figure 4.3 shows a visual representation of the new parametrization, with both the first and second parameter of the orthogonal PDF varying. The heights of the PDF are obtained by using the specific ω_i parameter values to calculate the values for original parameters θ_i and then using those to evaluate the original PDF. In this figure, a change in color represents a change in ω_1, while ω_2 is varied over linetype. We can see that ω_1 retains the role of location parameter, while ω_2 scales the distribution.

The estimation of parameters with the new parametrization is similar in spirit to the original one where we maximize the likelihood function with respect to ω_1, ω_2. However, due to the replacement of θ_2 with the fitted surface $\hat{s}(\omega_1, \omega_2)$, the optimization is a two-step process: every proposal of ω_1, ω_2 leads to an evaluation of $\hat{s}(\omega_1, \omega_2)$ first, which

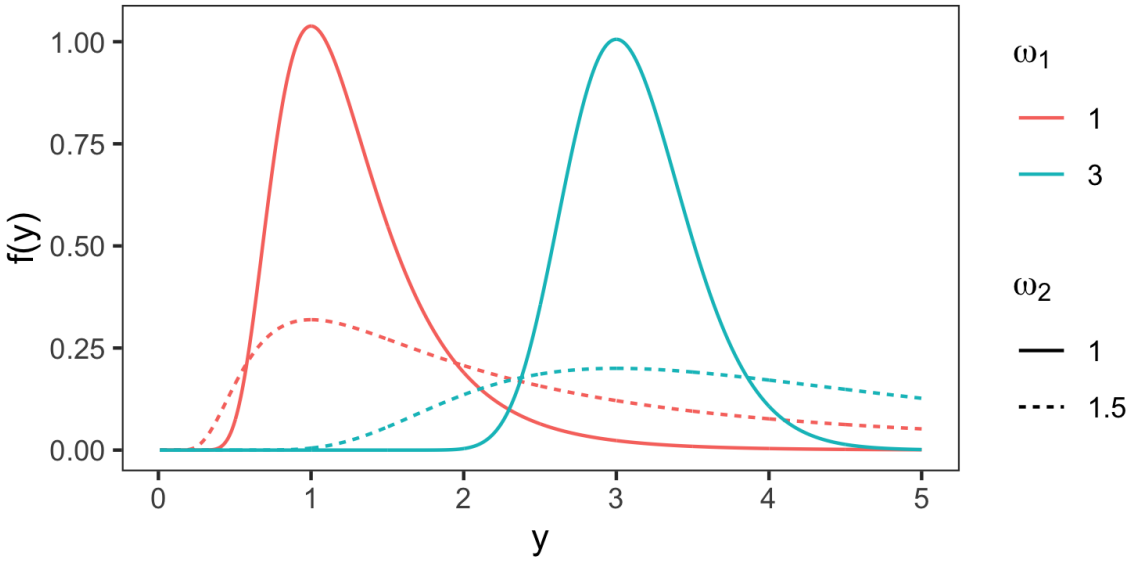

Figure 4.3: PDFs of the orthogonalized inverse gamma distribution for $\omega_1 = \{1, 3\}$ and $\omega_2 = \{1, 1.5\}$.

translates ω_1, ω_2 to θ_1, θ_2, from which we evaluate the likelihood function $L(y; \theta_1, \theta_2)$.

4.5 Numerical confirmation

4.5.1 General case

We now confirm the above results via simulation. In order to test whether a distribution's parametrization is orthogonal, we draw a random sample from the original, non-orthogonal, distribution; estimate old and new parameters; and compute empirical correlations between the MLEs for both parametrizations. We again focus on the inverse gamma distribution and its orthogonalized version introduced in (4.34).

For our simulation study, we choose 10 equally spaced values in the ranges $\theta_1 \in \{0.2, 8\}$ and $\theta_2 \in \{0.5, 8\}$. We draw a random sample of $n = 500$ observations and maximize the likelihood functions of the old and new parametrizations. Repeating this procedure $N = 500$ times, we compute the correlation between the maximized parameters. The results are shown in Figure 4.4. The left and right panels show correlations of the MLEs $(\hat{\theta}_1, \hat{\theta}_2)$ of the original non-orthogonal parameters, and the MLEs $(\hat{\omega}_1, \hat{\omega}_2)$ of the orthogonalized parameters, respectively. In almost all areas where MLEs of the

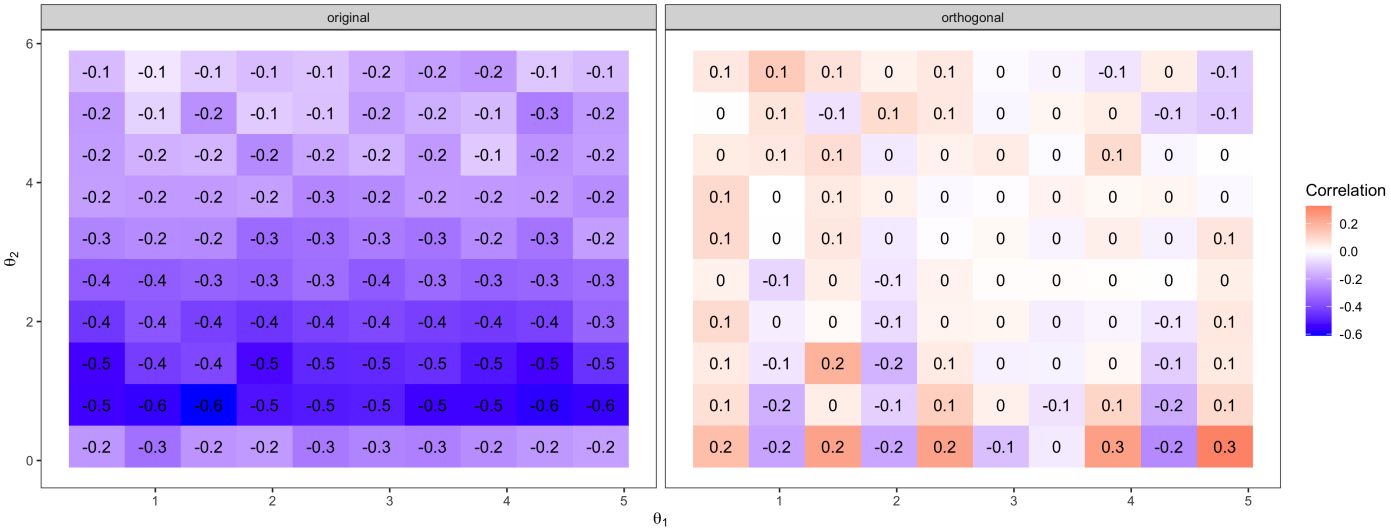

FIGURE 4.4: Empirical correlation of MLEs with different θ_1, θ_2 combinations. Left panel: empirical correlations of $(\hat{\theta}_1, \hat{\theta}_2)$; right panel: empirical correlations of $(\hat{\omega}_1, \hat{\omega}_2)$.

original parameters exhibit high, negative correlations, the new parametrization leads to significantly reduced correlations. Thus, the MLEs of the orthogonalized parameters will be approximately independent of each other.

4.5.2 Regression setting

The motivation for orthogonal parameters stems from the desire to interpret estimated regression coefficients for each parameter independently. When considering the covariance matrix of regression coefficient estimates, orthogonality of the response distribution leads to a block-diagonal Fisher information matrix (Heller et al., 2019), ensuring orthogonality between regression coefficients of different parameters. This prevents scenarios such as that presented in Figure 4.1, where misspecification of one parameter causes a bias in the estimation of regression coefficients in other parameters.

To confirm this result we present another simulation study based on the inverse gamma distribution, in which we analyze the robustness of the estimated coefficients of the location parameter to misspecification of the model for the scale parameter, in orthogonal and non-orthogonal parametrizations. We specify the following connection

between the parameters of the distribution of y and explanatory covariates:

$$y \sim \text{InverseGamma}(\theta_1, \theta_2) \tag{4.35}$$

$$\theta_1 = \beta_{10} + \beta_{11} x \tag{4.36}$$

$$\theta_2 = \beta_{20} + \beta_{21} x, \tag{4.37}$$

where θ_1, θ_2 represent the original parameters of the non-orthogonal PDF (4.30), and covariate x is distributed as $x \sim N(2, 0.5^2)$. Even though θ_1, θ_2 are constrained to be positive, we used identity link functions for simplicity. No computational problems were encountered.

We fix $\beta_{10} = \beta_{11} = \beta_{20} = 0.15$ and vary the θ_2 slope parameter $\beta_{21} \in \{0.05, 0.1, 0.15\}$. In each of those three variations of β_{21}, we simulate as follows: (a) generate x from $N(2, 0.5^2)$, (b) calculate true distributional parameters (θ_1, θ_2) according to (4.36), (4.37) and $\boldsymbol{\beta}$, (c) simulate y from (4.35) and (d) estimate regression coefficients according to the following four model specifications:

$$\begin{aligned}
&\text{S1.1: } \theta_1 = \beta_{10} + \beta_{11} x \quad \theta_2 = \beta_{20} \\
&\text{S1.2: } \theta_1 = \beta_{10} + \beta_{11} x \quad \theta_2 = \beta_{20} + \beta_{21} x \\
&\text{S2.1: } \omega_1 = \beta'_{10} + \beta'_{11} x \quad \omega_2 = \beta'_{20} \\
&\text{S2.2: } \omega_1 = \beta'_{10} + \beta'_{11} x \quad \omega_2 = \beta'_{20} + \beta'_{21} x,
\end{aligned} \tag{4.38}$$

where ω_i correspond to the orthogonal, and θ_i to the original PDF. The focus of interest is the unbiased estimation of coefficients of the first (location) parameter, so only $\hat{\beta}_{11}$ will be graphically displayed. Finally, we compute the difference between the true coefficients and the estimated ones: $\beta_{11}^{\text{diff}} = \beta_{11} - \hat{\beta}_{11}^{(\prime)}$.

The results are shown in Figure 4.5. While the four different groups of boxplots denoted on the x axis represent estimated model coefficients under different specifications (S1.1-S2.1) corresponding to (4.38), the differently colored boxplots represent varying values of β_{21} for simulation of (4.38). On the y-axis we observe the difference between our estimated and true values $\beta_{11} - \hat{\beta}'_{11}$.

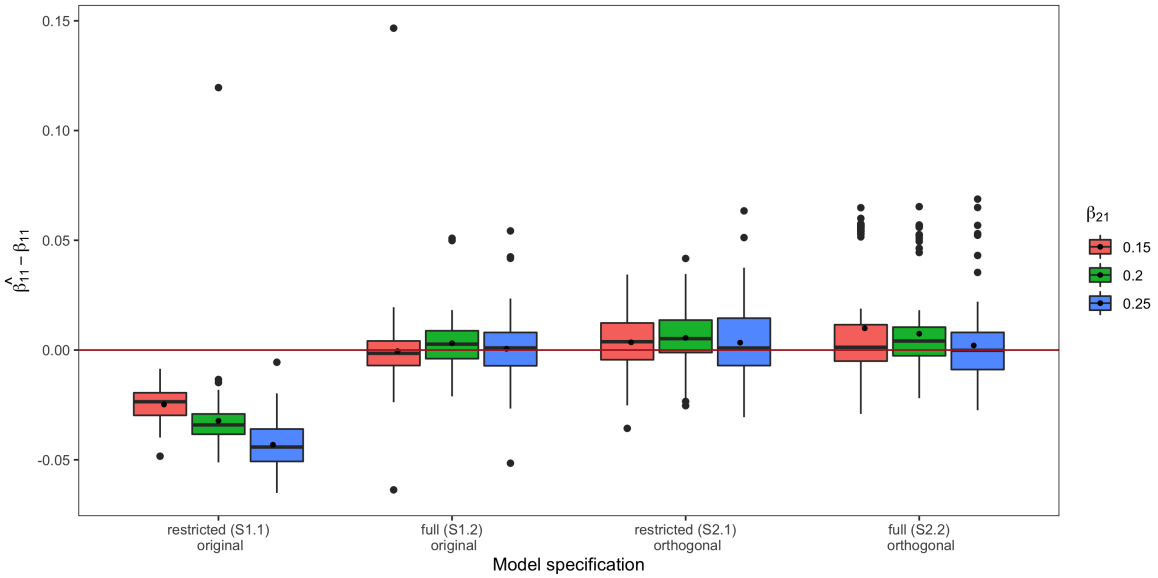

FIGURE 4.5: Results of simulation study comparing estimated regression coefficients in misspecified and correctly specified models across each of the original and orthogonal PDF of the inverse gamma distribution. Note that the results are slightly biased towards the non-orthogonal model, since this is how the data was originally specified. $n = 100$.

For all three simulated values of β_{21}, the misspecification leads to a bias of $\hat{\beta}_{11}$ in the original likelihood, as expected. The greater the value of β_{21}, i.e. the greater the influence of x on θ_2, the greater the bias. The MLEs based on the orthogonal likelihood, however, are either unbiased or have a substantially lower bias. This confirms the superiority of the model based on the orthogonal PDF, and is consistent with the results of similar simulations by Heller et al. (2019) of the Poisson-inverse Gaussian distribution, for which a closed-form orthogonalization was available.

4.6 Illustration: Extreme rainfall in Tasmania

The statistical modeling of extreme weather events plays a significant role for the organisation of any metropolitan area, small or large, since they represent a major threat to people's livelihoods. In this paper, we focus on the upper extreme of rainfall, as flooding is "the world's most frequent natural hazard affecting the largest number of

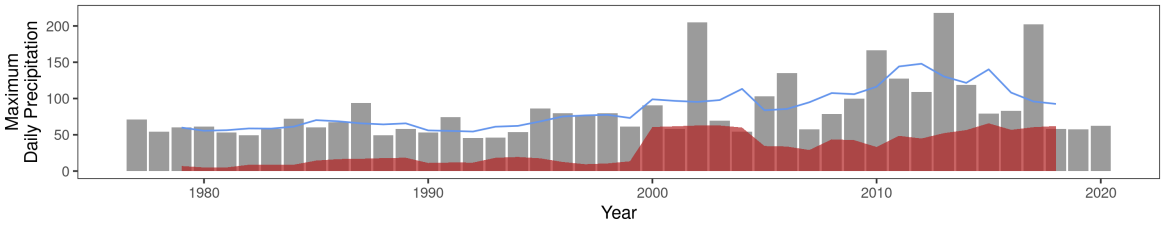

Figure 4.6: Maximum daily precipitation per year in Queenstown, Tasmania from 1977 to 2020. The rolling 10-day mean is represented using a blue line, while the red shaded area represents the rolling 10-day standard deviation.

people worldwide" (Leal, Boavida-Portugal, Fragoso, & Ramos, 2019). In particular, we concentrate on the annual maximum daily rainfall in Queenstown, Tasmania, Australia.

4.6.1 The dataset

Queenstown is one of the wettest cities in Australia with an average yearly rainfall of 2404 mm. The city's rainfall data were obtained using the online service "Climate data online" from the Bureau of Meteorology (2021).

Figure 4.6 shows a graphical summary of the annual maximum daily rainfall during the period 1977 to 2020. Evidently, precipitation levels are mostly stable during the first half of the period, after which more variation is seen (2000 to 2020). It appears that both the scale and shape of the distribution changes over time.

For modeling extreme weather events, the Gumbel distribution (Gumbel, 1935) has proven to be very useful: recent applications include hail size in the United States (Allen et al., 2017) and extreme wind values in South Korea (Kang, Ko, & Huh, 2015). The following parametrization is used in this section (Gumbel, 1935):

$$p(y|\ \theta_1, \theta_2) = \frac{1}{\theta_2} e^{-\left(\frac{x-\theta_1}{\theta_2} + e^{-\frac{x-\theta_1}{\theta_2}}\right)} \quad \text{with } \theta_1 \in \mathbb{R}, \theta_2 \in \mathbb{R}^+, y \in \mathbb{R}, \tag{4.39}$$

where θ_1 is a location parameter (the mode) and θ_2 is a scale parameter.

	Restricted Model	Full Model
$\hat{\beta}_{10}$	-54.137^{***} (11.6)	-34.784^{**} (14.0)
$\hat{\beta}_{11}$	2.761^{***} (0.6)	1.789^{**} (0.7)
$\hat{\beta}_{20}$	-0.824^{***} (0.2)	-118.782^{*} (39.3)
$\hat{\beta}_{21}$		5.892^{*} (2.0)

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

TABLE 4.1: Estimated coefficients of non-orthogonal models, with bootstrap standard errors in parentheses. Estimates and standard errors are rounded to three and one decimal point(s), respectively.

4.6.2 Modeling

Before using our method to orthogonalize the Gumbel distribution, we first implement traditional modeling approaches. Let's assume we are interested in finding out whether θ_1 (location) varies over time, ignoring possible scale parameter (θ_2) time changes (Restricted Model). We then compare these results with a model in which we also let θ_2 vary over time (Full Model). Our model specifications have the following form:

$$\begin{aligned} \theta_1 &= \beta_{10} + \beta_{11} \cdot year \\ \log(\theta_2) &= \underbrace{\underbrace{\beta_{20}}_{\text{Restricted Model}} + \beta_{21} \cdot year}_{\text{Full Model}} . \end{aligned} \tag{4.40}$$

Table 4.1 shows the estimated coefficients of both models. P values are obtained using a standard non-parametric Bootstrap procedure (Efron, 1979). In the restricted model, we observe that `year` has a significant influence on θ_1 ($\hat{\beta}_{11} = 2.761, p < 0.05$), which points to a positive mode shift over time. The full model has an additional coefficient of `year` $\hat{\beta}_{21}$, which is significantly different from zero ($\hat{\beta}_{21} = 5.892, p < 0.05$). This indicates that the scale of the distribution is also connected to `year`. Nonetheless, both coefficient estimates for the mode have altered considerably to $\hat{\beta}_{10} = -34.784$ and $\hat{\beta}_{11} = 1.789$, without any difference in the model specification for θ_1. These changes are caused by the misspecification bias induced by the nonzero correlation $\text{Corr}(\hat{\theta}_1, \hat{\theta}_2)$, described in Section 4.3.

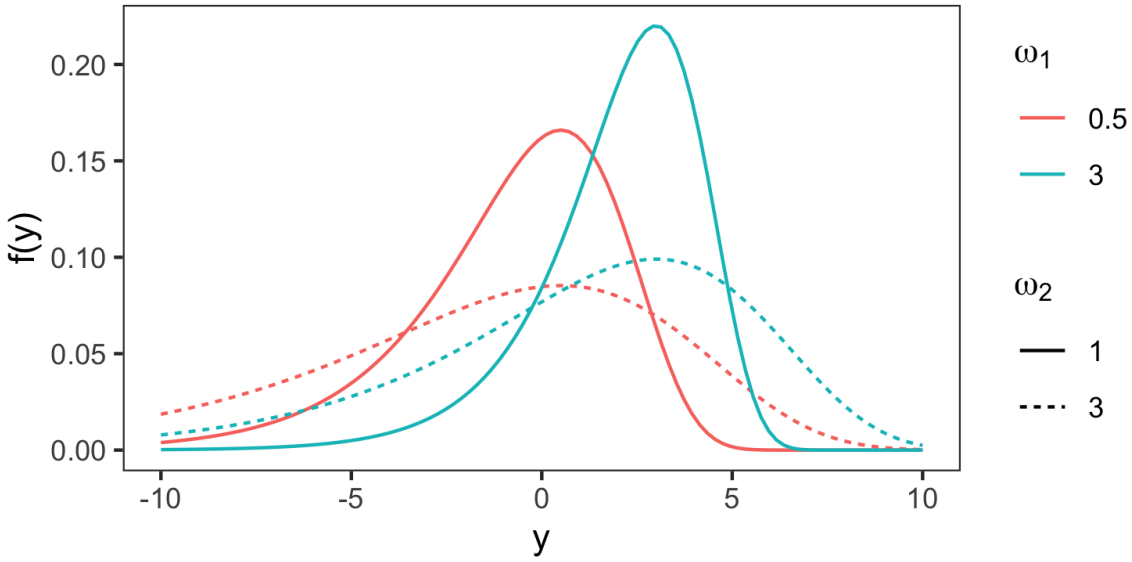

Figure 4.7: PDFs of the orthogonalized Gumbel distribution for $\omega_1 = \{0.5, 3\}$ and $\omega_2 = \{1, 3\}$.

The large changes in estimated coefficients of θ_1 due to the difference in specification of θ_2 highlights the unreliable estimation of regression coefficients in situations where their parent parameters are highly non-orthogonal. This means that we are not able to confidently interpret parameters independently. Seeking independent interpretation, we use a representation of the Gumbel distribution that was orthogonalized using our proposed method (Section 4.4.4). Figure 4.7 shows the PDF for different values of the orthogonal parameters ω_1 and ω_2. The calculation of the PDF and how ω_1, ω_2 are varied is similar in style to Figure 4.3. We can see that an increase of ω_1 leads to a difference in location, while ω_2 is a scale parameter.

Linking our effects to the new orthogonal representation, we use the following model specification:

$$\begin{aligned} \omega_1 &= \beta'_{10} + \beta'_{11} \cdot year \\ \log(\omega_2) &= \underbrace{\underbrace{\beta'_{20}}_{\text{Restricted Model}} + \beta'_{21} \cdot year}_{\text{Full Model}} . \end{aligned} \tag{4.41}$$

Table 4.2 shows the estimated coefficients of the two orthogonal models. It is apparent

	Restricted Model	Full Model
$\hat{\beta}'_{10}$	-31.827^{***} (10.5)	-31.827^{***} (10.4)
$\hat{\beta}'_{11}$	1.651^{***} (0.5)	1.651^{***} (0.5)
$\hat{\beta}'_{20}$	-32.709^{***} (112.7)	-85.269^{*} (419.0)
$\hat{\beta}'_{21}$		-79.882 (1532.8)

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

TABLE 4.2: Estimated coefficients of orthogonal models, with bootstrap standard errors in parentheses. Estimates and standard errors are rounded to three and one decimal point(s), respectively.

that the `year` coefficient estimate on the first parameter ω_1 does not change, irrespective of how ω_2 was modeled. Further, we can see that $\hat{\beta}'_{11} = 1.651$ is very similar to $\hat{\beta}_{11} = 1.789$ from the full original model (which we assume to have the full information). Additionally, we observe that the estimated coefficient of `year` on the scale parameter is not significantly different from zero ($p > 0.05$). The shift in time seems therefore mostly attributed to a shift in location, but not scale.

Due to the orthogonalization of our PDF, we can now safely interpret the coefficient estimates on the first parameter, without concern about misspecification of other parameter(s). While the mode of the distribution of extreme rainfall is significantly shifted over time, the apparent increase in scale is not statistically significant.

4.7 Conclusion

In this paper, we outline the existence of a bias in distributional regression model estimation that arises when the distributional parametrization is non-orthogonal and one parameter is incorrectly specified. Addressing this issue, we develop a framework to orthogonalize two-parameter distributions such that the resulting regression coefficient estimates are uncorrelated between parameters. To do so, we numerically solve a system of ODEs based on the original parameters' MLE covariance matrix and connect the results to the second parameter of our chosen distribution. The result of this procedure is a parametrization of the chosen distribution which is robust to the bias coming from a misspecification of parameters which the applied researcher may not be interested in.

There are limitations to our present framework. Firstly, more work on the orthogonalization of distributions with three or more parameters would be beneficial. While Huzurbazar (1950) proposes a method for analytically orthogonalizing three-parameter distributions, he deems solvable only the case where two of the three parameters are already orthogonal, due to the complexity of the differential equations. However, with modern computational power, this complexity could conceivably be overcome. Secondly, the optimization with respect to the new, orthogonal parameters is computationally expensive. Lastly, the interpretation of effects on any distributional parameters, whether they are orthogonal or not, remains difficult if they do not equate to distributional moments or other interpretable quantities. More work on the translation of estimated effects on the parameters to effects on the distributional location or shape would therefore be beneficial.

Notwithstanding, our proposed work provides a solution to a striking issue in regression analysis beyond the mean. Using our framework, applied researchers can parametrize distributions in order to produce reliable regression coefficient estimates and draw robust conclusions.

5

Variable importance in likelihood-based regression models

5.1 Introduction

The field of regression analysis beyond the classic linear model framework has seen a substantial amount of innovation over the last decades. Modern prominent methods include generalized linear models (GLM, Nelder & Wedderburn, 1972) and generalized additive models for location, scale and shape (GAMLSS, Rigby & Stasinopoulos, 2005). This combined incremental increase of modeling freedom in choosing target distributions (GLM) and the ability to simultaneously model multiple distributional parameters (GAMLSS) forms a toolkit that can be tailored for individual modeling

scenarios. Furthermore, the development of model estimation techniques incorporating model selection like the least absolute shrinkage and selection operator (LASSO, Tishbirani, 1996) or boosting (Freund, Schapire, & Abe, 1999; Schapire, 1990) ensures that applied researchers find the most optimal model to resolve their hypotheses.

While an optimal model specification ensures the correct interpretation of estimated coefficients, many applied researchers further require the assessment of the predictors' relative importance (Green, Carroll, & Desarbo, 1978). Examples include ranking contributions to customer satisfaction scores (Garver & Williams, 2020; J. Johnson, 2000) or finding drivers of species richness in conservation biology (MacNally, 1996). Common components of a fitted regression model, like coefficients and t or p values, are however not useful for comparing explanatory covariates across their variable importance, since they are subject to either

a) covariate scale-changes (raw coefficients),

b) over-stating the importance of correlated predictors (standardized coefficients, t values),

c) order-dependency (sequential sum of squares), or

d) unsuitability for coefficients with degrees of freedom greater than one (raw and standardized coefficients).

This lacking ability to compare and rank explanatory covariates has motivated many researchers to propose alternative metrics. In a comprehensive summary of relative importance in linear regression, Grömping (2015) points to two metrics that can be considered best-practice for ranking variable importance: "hierarchical partitioning" (also called "LMG" after the authors in Lindeman, Merenda, & Gold, 1980) calculates the goodness-of-fit measure of all possible submodels of the "full" regression model to then find each variable's average contribution, while a procedure called "relative weights" (RW) analysis (Fabbris, 1980; Genizi, 1993; J. Johnson, 2000) is based on an orthogonal representation of the model design matrix. Grömping (2015) shows that

while hierarchical partitioning (HP) is able to take into account the most information and therefore is the most complete, relative weights (RW) are a fast and strong approximation in higher-dimensional predictor spaces ($Q > 10$).

Unfortunately, the aforementioned metrics have only been developed for linear models and logistic regression (see Tonidandel & LeBreton, 2010). Since this represents just a small fraction of the vast modeling and target distribution landscape (Fahrmeir et al., 2013), we propose an extension of both the all-subset metric hierarchical partitioning and relative weights analysis to likelihood-based models with linear predictors including GLMs beyond logistic regression and GAMLSS. We present a user-friendly implementation using the statistical programming language `R` (R Core Team, 2020). With this new extension, researchers can use modern modeling techniques while also ranking each effect's covariate importance.

The remainder of this paper is structured as follows: Section 5.2 describes the supported model classes, while Section 5.3 gives an example of where our `R` package `vibe` is useful. In Section 5.4 we review the most effective variable importance metrics and present our extension to GLMs and GAMLSS in Section 5.5, while Section 5.6 introduces the corresponding `R` package. Our proposal is validated in simulations found in Section 5.7, after which Section 5.8 shows an application of `vibe` to a dataset on malnourished children in India. Section 5.9 concludes the paper.

5.2 Model classes

Using generalized additive models for location, scale and shape (GAMLSS, Rigby & Stasinopoulos, 2005), multiple parameters of a K-parametric response distribution $y \sim D(\theta_1, \ldots, \theta_K)$ can be simultaneously linked to additive predictors of different forms. Each parameter is assigned its own model equation that can have an independent subset of predictor effects. More importantly, distributional choice is not limited to the exponential family, as long as the probability density function (PDF) is twice continuously differentiable with respect to its parameters. Effect choice is similar to those of generalized additive models (GAM, Hastie & Tibshirani, 1990) - they can

be parametric (linear, categorical, with interactions etc.) or non-parametric (random, smooth splines, decision trees, spatial effects, etc.). This amounts to the following model equation:

$$\begin{aligned} \eta_k &= \beta_{k0} + \sum_{q=1}^{Q_k} f_{kq}(\mathbf{x}_{kq}; \boldsymbol{\beta}_{kq}) \\ h_k(\eta_k) &= \theta_k, \end{aligned} \tag{5.1}$$

where $h_k(\cdot)$ is a pre-specified response function intended to uphold the support of parameter θ_k and $\boldsymbol{\beta}_{kq}$ are regression coefficients of effect f_{kq}. The result of (5.1) is a highly flexible modeling environment beyond the mean. Due to parameters of many distributions not being equal to either the location, scale or shape of a distribution, the term "distributional regression" was coined by Klein, Kneib, Lang, and Sohn (2015).

Many different software packages implement versions of distributional regression. The original `R` package `gamlss` with coefficients obtained via maximum-likelihood estimate (MLE) was developed by Stasinopoulos and Rigby (2007), while the `R` extension `bamlss` (Umlauf et al., 2018) uses Posterior-Mode maximization and subsequent Markov chain Monte Carlo (MCMC) sampling to present a posterior distribution for each regression coefficient. `betareg` (Cribari-Neto & Zeileis, 2010) presents a distributional regression framework specifically for beta regression, whereas `gamboostLSS` (Thomas et al., 2018) includes boosting algorithms in its estimation of distributional regression models, geared towards potentially high-dimensional datasets.

The generality of distributions, effect choices and link functions means that distributional regression encompasses many known regression models, such as generalized linear models (GLM, Nelder & Wedderburn, 1972), GAM and generalized additive mixed models (GAMM, Lin & Zhang, 1999).

To support the estimation of variable importance in a wide range of regression scenarios, the proposal of this paper is built on GLM and GAMLSS model classes, although it is applicable to any likelihood-based model class using linear and parametric predictors. The implemented variable importance `R` package `vibe` introduced in this paper is compatible with `R` packages `mgcv`, `gamlss` and the built-in `R` function `glm()`.

The `mgcv` compatibility ensures a wider range of distributions than what is possible using `glm()`, as seen in Section 5.3. However, the compatibility of effect types from both `gamlss` and `mgcv` are currently limited to linear and categorical effects only.

5.3 Motivational example

Rajendra and Machhindra (2018) aim to determine drivers of resident satisfaction in three university student hostels in Kolhapur (India) from information gathered in a survey with 183 participants. The target variable measures the overall satisfaction level in five distinct categories, ranging from "very dissatisfied" to "very satisfied". The study attempts to analyze this variable labelled `rsat` and its connection to explanatory variables measuring the residents' ratings of individual parts of their hostel experience, on a scale of 1 (very dissatisfied) to 5 (very satisfied). These 29 explanatory variables include questions about the quality of the facilities, the environment cleanliness, residents' personal rooms and food provided by the hostel. Control variables including resident's gender and the specific hostel students were living in are also recorded.

In contrast to the original publication by Rajendra and Machhindra (2018), a regression model to connect `rsat` to all explanatory variables is employed. Due to the ordinal nature of the dependent variable we apply an ordinal logistic regression model. As suggested by McCullagh (1980), we assume a latent variable that drives the categorical cuts in `rsat` as follows:

$$y^* = \beta_0 + \mathbf{X}\boldsymbol{\beta} + e, \quad e \mid \mathbf{X} \sim \text{Logistic}(\mu, s) \tag{5.2}$$

where $\phi_1 < \phi_2 < \phi_3 < \phi_4$ are cut-points for translating y^* into `rsat`:

$$\begin{aligned} rsat &= \text{“very dissatisfied”} && \text{if } y^* \leqslant \phi_1 \\ rsat &= \text{“dissatisfied”} && \text{if } \phi_1 < y^* \leqslant \phi_2 \\ rsat &= \text{“neutral”} && \text{if } \phi_2 < y^* \leqslant \phi_3 \\ rsat &= \text{“satisfied”} && \text{if } \phi_3 < y^* \leqslant \phi_4 \\ rsat &= \text{“very satisfied”} && \text{if } y^* > \phi_4, \end{aligned} \tag{5.3}$$

where $\boldsymbol{\beta}$ and $\phi_i, i = 1, \ldots, 4$ are to be estimated. We can then connect the latent variable y^* to explanatory variables as in (5.2). Conducting model estimation using the `R` package `mgcv` (Wood, 2011), we execute the following code:

```
gam_ocat <- gam(
  rsat ~ Gender + hostel_no + Q.1 + Q.2 + Q.3 + Q.4 + Q.5 + Q.6 +
    Q.7 + Q.8 + Q.9 + Q.10 + Q.11 + Q.12 + Q.13 + Q.14 + Q.15 +
    Q.16 + Q.17 + Q.18 + Q.19 + Q.20 + Q.21 + Q.22 + Q.23 + Q.24 +
    Q.25 + Q.26 + Q.27 + Q.28 + Q.29,
  data = sat, family = ocat(R = 5)
)
```

In the above code chunk, variables $Q.o, o = 1, \ldots, 29$ refer to the questions residents were asked in the satisfaction questionnaire, with Table A.1 in the Appendix containing the full questions. As visible in the obtained model summary, Table 5.1, there are seven variables that are significantly associated with resident satisfaction ($p < 0.05$): `hostel_no` (hostel number), `Q.2` (security systems), `Q.9` (parking lot), `Q.10` (study room), `Q.18` (annual functions), `Q.26` (comfortable beds) and `Q.28` (availability of drinking water).

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-5.90	1.34	-4.41	**0.00**
hostel_no	0.60	0.22	2.77	**0.01**
Q.2	-0.38	0.18	-2.11	**0.04**
Q.9	0.44	0.20	2.23	**0.03**
Q.10	-0.41	0.19	-2.14	**0.03**
Q.18	0.43	0.17	2.54	**0.01**
Q.26	0.44	0.17	2.55	**0.01**
Q.28	0.47	0.18	2.62	**0.01**

TABLE 5.1: Coefficients of the ordinal resident satisfaction model, only containing variables with $p < 0.05$. The full version can be found as Table A.2 in the Appendix.

The information of significant association between the dependent variable and those seven explanatory variables is useful for the hostel management, yet it remains unknown which of these covariates has the biggest impact. One could look at the absolute coefficients in Table 5.1, but since we can assume that covariates are correlated (the "hostel number" and "comfortable beds" or "security systems" could share some information),

the relative importance is unclear.

To rank the variables in their contribution to resident satisfaction, we use our package `vibe`, which provides two variable importance metrics: hierarchical partitioning and relative weights. Since hierarchical partitioning would have to calculate 2^{31} sub-models (which amounts to more than 21 billion), we use the less computationally expensive relative weights (more detail in Section 5.4).

The `vibe` package automatically recognizes the model type and calculates the specified metrics. To calculate it for the ordinal categorical model and obtain a formatted table of results, we use the following code:

```
rw_gam <- vibe(gam_ocat, metric = "relweights", gofmetric = "R2e")
rw_gam$results
```

Covariate Name	Parameter	I. Effects	I. Effects (fraction)
hostel_no	mu	0.038	0.09
Q.6	mu	0.022	0.051
Q.7	mu	0.024	0.057
Q.9	mu	0.037	0.088
Q.18	mu	0.03	0.072
Q.26	mu	0.039	0.092
Q.28	**mu**	**0.04**	**0.096**

Table 5.2: Relative weights analysis results for the model specified in 5.2. Column "Parameter" descibes which parameter was modeled, while "I. Effects" stands for independent effects and describes each variable's raw contribution to a reduction in the goodness-of-fit figure. In the fourth column, independent contributions are scaled to sum to one. Covariates with an independent contribution lower than 0.05 were excluded from this table (full table available as A.3 in the Appendix).

Table 5.2 shows the formatted output of Line 2 of the above code chunk.[1]. This allows us to obtain each variable's relative contribution towards reducing the model's error rate. The model goodness-of-fit measure used is a likelihood-based "R^2" proposed by Estrella in 1998 (more detail is provided in Section 5.5). Leading effects are `Q.28` (drinking water) and `Q.26` (comfortable beds), with `hostel_no` and `Q.9` (parking lot) in third and fourth place, respectively. We can conclude that the availability of

[1]The parameter is denoted as "mu" due to notation of the underlying software, but it is equal to θ_1. This recurs occasionally in forthcoming graphs.

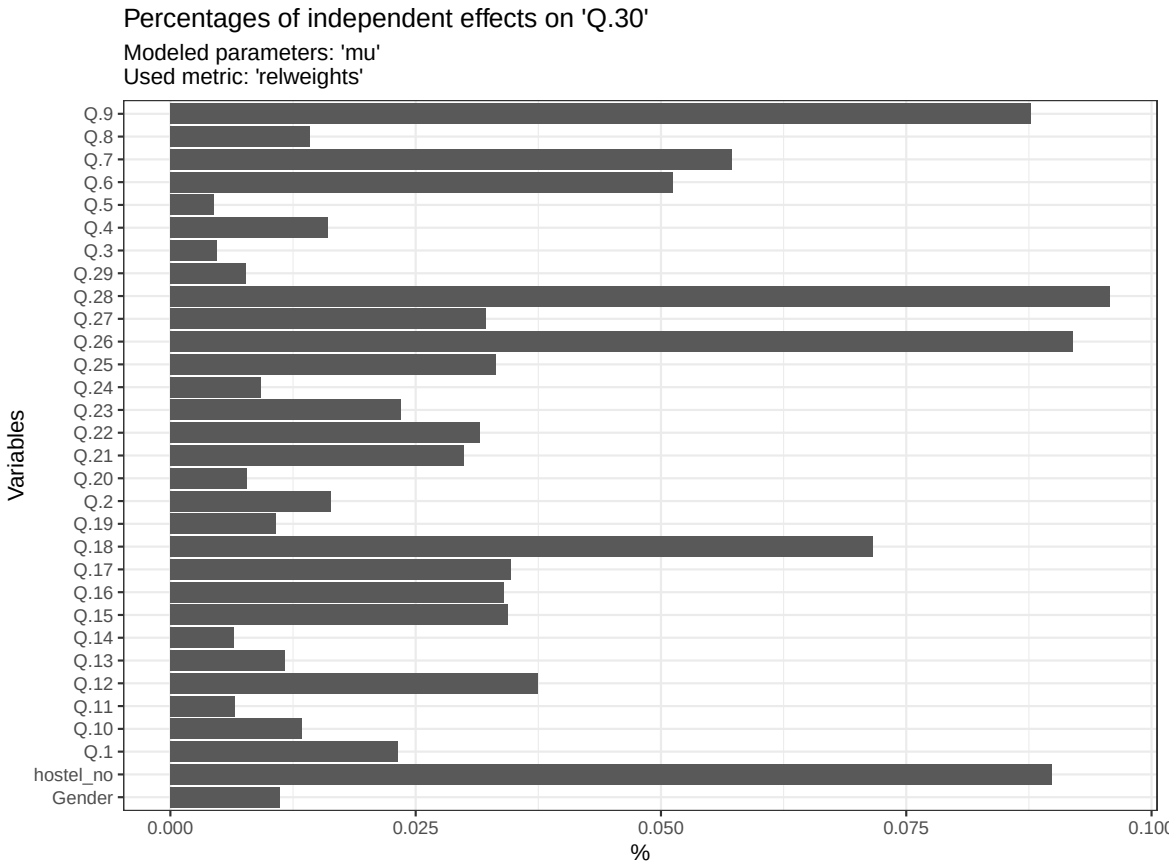

FIGURE 5.1: Variable importance results by explanatory covariates with metric "relative weights", measuring the share (%) of error reduction for the fitted ordinal categorical GLM, produced with package `vibe`.

good drinking water, pleasant bedding and the supply of parking spots are the most important drivers of resident satisfaction. The hostel manager having to allocate new funds could therefore prioritize improving those elements of the hostel environment.

To produce publication-ready graphs, the applied researcher can use the built-in plot function for visualization of results as follows:

```
plot(rw_gam)
```

The result of this short code-snippet can be seen in Figure 5.1.

5.4 Variable importance metrics

5.4.1 Hierarchical partitioning

The mathematical theory of hierarchical partitioning as proposed by Chevan and Sutherland (1991) considers one target variable y paired with Q predictors. The predictors can be included into a model in 2^Q unique ways, from the intercept model (no predictors are included) to the full model (all predictors are included). A user-specified goodness-of-fit, denoted by R_x with $x \in \{0, 1, 2, 12, \ldots, 123 \cdots Q\}$, is extracted from each possible sub-model.

For the 2^Q model combinations there are $Q!$ ways of ordering the variables (permutations). In the three-variable case, the permutations are:

$$123, 132, 312, 321, 231, 213.$$

Each of these permutations represents how a full model can sequentially be built. The first number represents the variable for which the average contribution is calculated. In each of the sequential steps, the variable of interest is removed and the difference in the goodness-of-fit obtained. For the last of the above permutations, 213, the following differences would be calculated:

$$\begin{aligned} R_2 - R_0 &= G^{\text{Diff}}_{221} \\ R_{21} - R_1 &= G^{\text{Diff}}_{222} \\ R_{213} - R_{13} &= G^{\text{Diff}}_{223}, \end{aligned} \tag{5.4}$$

where R_{vars} corresponds to a specified goodness-of-fit of a model with included covariate effects in the index (R_{12} would link to a model with covariates 1 and 2, whereas R_0 refers to the goodness of fit of an "only intercept" model) and $G^{\text{Diff}}_{qjl}, q = 1, \ldots, Q;\ j = 1, \ldots, (Q-1)!; l = 1, \ldots, Q$ corresponds to the lth difference in goodness of fit calculated in the jth permutation linked to the qth covariate effect. After repeating this procedure

for every permutation, the average contribution of variable q is obtained as follows:

$$\bar{G}_q^{\text{Diff}} = \sum_{j=1}^{(Q-1)!} \sum_{l=1}^{Q} \frac{G_{qjl}^{\text{Diff}}}{Q!} \tag{5.5}$$

Originally created to decompose the R^2 of a linear model, hierarchical partitioning can also be used in combination with other goodness-of-fit measures or model types by simply swapping out R_{vars}. A convenient choice for likelihood-based regression models is one that takes the evaluated likelihood-function into account. Examples include the model deviance, or a "Pseudo-R^2" as proposed by Estrella (1998).

5.4.2 Relative weights

Although in theory hierarchical partitioning can be generalized for Q predictors and K parameters, the computational time increases exponentially with the number of predictors.

To solve this shortcoming of hierarchical partitioning, an approach labeled "relative weights" was proposed by Fabbris (1980), Genizi (1993) and later J. Johnson (2000). In their proposal, the contribution of each covariate on the dependent variable is calculated using a two-step procedure: first, an orthogonalization of the design matrix occurs, which is then connected to the dependent variable using simple linear regression. Second, the standardized coefficients of the orthogonal model are then weighted based on the relationship between the orthogonalized and original variables.

The most recent version modernized by J. Johnson (2000) starts with the design matrix $\mathbf{X}$ of dimension $N \times Q$, centered to 0 and scaled to $\text{Var}(x_q) = 1$ and the dependent variable y with dimension $N \times 1$. Using a singular value decomposition, as long as $\mathbf{X}$ is of full rank and has more rows than columns, one obtains an orthogonal matrix $\mathbf{Z}$ that shares the same dimension as $\mathbf{X}$:

$$\begin{aligned} \mathbf{X} &= \mathbf{U}\mathbf{D}\mathbf{V}^\intercal \\ \mathbf{Z} &= \mathbf{U}\mathbf{V}^\intercal, \end{aligned} \tag{5.6}$$

where $\mathbf{U}_{N\times N}$ consists of the eigenvectors of $\mathbf{X}\mathbf{X}^\intercal$, $\mathbf{V}^\intercal_{N\times Q}$ holds the eigenvectors of $\mathbf{X}\mathbf{X}^\intercal$ and $\mathbf{D}_{Q\times Q}$ is a diagonal matrix containing the singular values of $\mathbf{X}$. Using the orthogonal design matrix $\mathbf{Z}$, the ordinary least squares (OLS) coefficients $\boldsymbol{\beta}^* = (\mathbf{Z}^\intercal\mathbf{Z})^{-1}\mathbf{Z}^\intercal y$. Even though $\boldsymbol{\beta}^*$ are standardized and now represent estimators of uncorrelated variables in $\mathbf{Z}$, removing the issue of shared variable error reduction, they do not represent good approximations for $\mathbf{X} \rightarrow y$ contributions, since $\mathbf{Z}$ itself might not be a good representation of $\mathbf{X}$, especially in scenarios with high correlation.

The missing link of $\mathbf{X} \rightarrow \mathbf{Z}$ is reinstated by a linear model with explanatory variables $\mathbf{X}$ and dependent variables $\mathbf{Z}$. The coefficients $\boldsymbol{\Gamma}$ are calculated as follows:

$$\boldsymbol{\Gamma} = (\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\mathbf{Z} \tag{5.7}$$

The result is a matrix $\boldsymbol{\Gamma}$ with dimensions $Q \times Q$, where each column consists of the coefficients for a regression model of $\mathbf{X}$ on z_q. Those columns are converted into weights by dividing each element by the column sum as follows:

$$\gamma^{*2}_{mq} = \frac{\gamma^2_{mq}}{\sum_{m=1}^{Q}\gamma^2_{mq}} \tag{5.8}$$

where $\gamma^2_{mq}, m = 1,\ldots,Q, q = 1,\ldots,Q$ are the squares of general entries in $\boldsymbol{\Gamma}$. The relative weights of $\mathbf{X}$ on y are finally calculated as

$$\delta^2_q = \sum_{m=1}^{Q}\gamma^{*2}_{mq}\beta^{*2}_q \tag{5.9}$$

and represent the share of R^2 that each variable contributes to the model. The biggest advantage of this procedure is the small number of regression models that need to be estimated: independently of the predictor space dimensionality, we only need to calculate the two model coefficient matrices $\boldsymbol{\beta}^*$ and $\boldsymbol{\Gamma}$, instead of 2^Q model combinations when calculating the results of hierarchical partitioning.

However, the procedure of calculating relative weights as proposed by J. Johnson (2000) is limited to multiple linear regression models only. As soon as more complex

methods are used, as in the case with non-normally distributed target variables or in situations where we are interested in more than just the mean, this metric is no longer applicable without amendments.

The limitations of relative weights in its original contribution were recognized by Tonidandel and LeBreton (2010), who extended the concept to ensure compatibility with logistic regression models. Their proposed method follows that of J. Johnson (2000) in orthogonalizing $\mathbf{X}$ to $\mathbf{Z}$ and subsequently connecting both matrices as described in (5.7), although $\mathbf{\Gamma}$ changes to $(\mathbf{Z}^\intercal\mathbf{Z})^{-1}\mathbf{Z}^\intercal\mathbf{X}$ ($\mathbf{Z}, \mathbf{X}$ are flipped). The coefficients $\boldsymbol{\beta}^*$ are replaced with their equivalent logistic regression coefficients $\hat{\boldsymbol{\beta}}^*_{\text{logistic}} = \arg\max_{\boldsymbol{\beta}^*} \prod_{i=1}^n L_i(\boldsymbol{\beta}^*; \mathbf{Z})$, where $L_i(\cdot)$ are the individual likelihood contributions of $\mathbf{Z}$ based on the binomial distribution. Finally, the regression coefficients are standardized by dividing them by the standard error of the predicted parameter $\hat{\pi}$ and multiplying them by the R^2 of the orthogonal model. The relative weights are again weighted using $\mathbf{\Gamma}$ as in (5.9).

Further, Tonidandel and LeBreton (2010) show that the results obtained from the above procedure are very similar to those obtained using hierarchical partitioning, at a fraction of the computing time.

5.5 Extensions to GLM and GAMLSS

Previous sections have shown that viable proposals exist to assign importance to variables in linear models and logistic regression models. However, a sizeable gap still prevails in variable importance beyond these two model classes. Although hierarchical partitioning is readily applicable to all different types of regression models, there is not currently any available software that an applied researcher can use. Further, the issue of expensive computation with large numbers of covariates perseveres. To bridge this gap, this paper a) extends relative weights analysis (RWA) and the all-subset metric hierarchical partitioning to likelihood-based models with linear predictors, specifically GLMs beyond logistic regression and GAMLSS with linear predictors (this Section),

and b) features an easy-to-use `R` extension called `vibe` that implements the proposed extensions (Section 5.6).

To generalize relative weights beyond linear models, we change the algorithm proposed by Tonidandel and LeBreton (2010) on four accounts: first, the estimated coefficients of the orthogonal model are based on the same model class as the full model with a mirrored model specification. For example, this entails that if the regression specification was $y \sim D(g_1(\theta_1) = \mathbf{X}\boldsymbol{\beta}_1, g_2(\theta_2) = \mathbf{X}\boldsymbol{\beta}_2)$ with distribution $D(\theta_1, \theta_2)$ and link functions $g_1(\cdot)$, $g_2(\cdot)$, then $\boldsymbol{\beta}^*$ will be estimated with specification $y \sim D(g_1(\theta_1) = \mathbf{Z}\boldsymbol{\beta}_1^*, g_2(\theta_2) = \mathbf{Z}\boldsymbol{\beta}_2^*)$. Second, we alter the goodness-of-fit used to scale the regression coefficients $\boldsymbol{\beta}^*$, R^2 to a likelihood-based one suggested by Estrella (1998):

$$R^2_{\mathrm{E}} = 1 - \left[\frac{\ln\left(L^{\max}\right)}{\ln\left(L_0^{\max}\right)}\right]^{-\left(\frac{2}{n}\right)\ln\left(L_0^{\max}\right)}, \tag{5.10}$$

where $L_0^{\max}$ and $L^{\max}$ represent the (maximum) evaluated likelihood function of a model with no explanatory covariates, and of a model containing the orthogonalized design matrix $\mathbf{Z}$, respectively. This change from a goodness-of-fit based on $\hat{y}$ (R^2) to one that is likelihood-based reflects the fact that modeled parameters are not always the mean, as is the case in GAMLSS, for example. Third, we allow for the use of categorical effects as part of the model equation. Lastly, our algorithm takes into account the simultaneous modeling of multiple parameters, as is done in distributional regression.

In detail, we propose the following Algorithm 1, where $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_Q)^\intercal$ represents a matrix containing all available predictors and $\mathbf{X}_k = \mathbf{X}_1, \ldots, \mathbf{X}_{Q_k}$ represents a subset of $\mathbf{X}$ consisting of the covariates connected to parameters θ_k which can be distributional parameters as known from distributional regression but also moments such as the mean used in GLM and GAM. In the latter two model classes, $\mathbf{X}_k$ always represents the full data matrix $\mathbf{X}$. $\boldsymbol{\theta}$ contains the parameters of y that should be connected to predictors $\mathbf{X}_k$.

In Algorithm 1, we first convert categorical effects to dummy variables, if they exist (Lines 3–5). Then, we prepare and orthogonalize $\mathbf{X}$ into $\mathbf{Z}$ (Lines 7–10), after which the $\mathbf{X} \rightarrow \mathbf{Z}$ weights are obtained. In Lines 11–12 the regression coefficients with regards to

Algorithm 1: Generalized relative weights

Input: $\mathbf{y}, \mathbf{X}, \boldsymbol{\theta}$, $L(\cdot)$

1 **for** $k = 1$ *to* K **do**
2 $\quad \mathbf{X}_k = \text{subset}(\mathbf{X})$; // Obtain relevant predictors
3 $\quad cat = \text{has\_categorical}(\mathbf{X}_k)$; // binary: any categorical vars?
4 $\quad$ **if** cat = *TRUE* **then**
5 $\quad\quad \mathbf{X}_k = \text{convert}(\mathbf{X}_k)$; // convert to dummy-coding
6 $\quad$ **end**
7 $\quad \mathbf{X}_k^s = \text{standardize}(\mathbf{X}_k)$; // substract mean and divide by sd
8 $\quad \mathbf{Z}_k = \text{svd}(\mathbf{X}_k^s)$; // obtain orthogonal rep. of X
9 $\quad \mathbf{Z}_k^s = \text{standardize}(\mathbf{Z}_k)$
10 $\quad \Lambda = (\mathbf{Z}_k^{s\intercal}\mathbf{Z}_k^s)^{-1}\mathbf{Z}_k^s\mathbf{X}_k^s$; // obtain $\mathbf{X} \to \mathbf{Z}$ contributions
11 $\quad \boldsymbol{\beta}^*, L^{\max} = \arg\max_{\boldsymbol{\beta}^*} \prod_{i=1}^n L_i(\boldsymbol{\beta}^*|\mathbf{y}, \mathbf{Z})$; // get coefs, likelihood
12 $\quad L_0^{\max} = \arg\max_{\boldsymbol{\beta}_0^*} \prod_{i=1}^n L_i(\boldsymbol{\beta}_0^*|\mathbf{y}, \mathbf{Z})$; // obtain empty likelihood
13 $\quad s_{\hat{\eta}} = \text{sd}(\mathbf{Z}_k^s\boldsymbol{\beta}^*)$; // Obtain sd of linear predictor.
14 $\quad g = \text{gof}(L_0^{\max}, L^{\max})$; // Compute goodness-of-fit
15 $\quad \mathbf{w} = \Lambda^2 \left(\boldsymbol{\beta}^* \frac{g}{s_{\hat{\eta}}}\right)^2$; // Obtain relative weights
16 $\quad$ **if** cat = *TRUE* **then**
17 $\quad\quad \mathbf{w} = \text{coerce}(\mathbf{w})$; // coerce weights into one effect
18 $\quad$ **end**
19 $\quad$ return($\mathbf{w}$)
20 **end**

Output: Relative weights $\mathbf{w}$

$\mathbf{Z}$ ($\boldsymbol{\beta}^*$) and the coefficients for an "intercept model" ($\boldsymbol{\beta}_0^*$, no covariates) are calculated. The corresponding maximized likelihood function values $L^{\max}, L_0^{\max}$ are stored. Lines 13 and 14 calculate the standard error of the predicted response function and the user-chosen goodness-of-fit respectively, while in Line 15 the coefficients are combined with the goodness-of-fit, the previously calculated predicted response standard error and Λ to obtain the relative weights. Lines 16–18 sums the relative weights of dummy-coded variables to form unified categorical effects. If the user-chosen goodness-of-fit is not dependent on the empty model, Line 12 can be disregarded.

With our changes, the proposed algorithm is applicable to any likelihood-based

regression model with linear predictors and scaleable to a large number of variables. If the original, full model is a distributional regression model, all of the above steps are repeated for each parameter independently (seen as a `for` loop ranging from Lines 1–20).

5.6 The `vibe` package

We present a user-friendly implementation of the metrics introduced in Sections 5.4 and 5.5 in `R` called `vibe` (**V**ariable **I**mportance **Be**yond Linear Models), which was briefly showcased in Section 5.3. Code, tests and an installation manual can be found at https://github.com/Stan125/vibe.

The software extension `vibe` is designed to be lightweight, simple and easy to use. It revolves around its main function, also called `vibe()`, which conducts the main calculations. Depending on the full model provided, `vibe` uses one of three different methods: `vibe.glm()`, `vibe.gam()` or `vibe.gamlss()`. The arguments which can be provided are as follows:

```
function (object, metric = "hp",
          gofmetric = "R2e", ncores = 1,
          progress = TRUE, ...)
```

With the argument `object`, a fitted regression model is specified. This can be one of either class `"glm"`, `"gam"` or `"gamlss"`. From this object, `vibe` automatically recognizes the provided model class. If the object is of class `"gamlss"` and more than one parameter is linked to predictors, it is automatically recognized by `vibe` and variable importance is calculated for each parameter individually. With the argument `metric`, one of the two proposed metrics can be chosen: `"hp"` stands for "hierarchical partitioning" and calculates the all-subset metric hierarchical partitioning, whereas specifying `"relweights"` will calculate relative weights. With the argument `gofmetric`, the user can specify different goodness-of-fit figures, although only `"r2e"` (Pseudo-R^2 as defined in 5.10) is currently available. The arguments `ncores` and `progress` control the parallelization if `metric = "hp"` is specified, while multiple cores for computation

of sub-models can be specified with `ncores`. With `progress`, a progress bar can be displayed.

After the `vibe()` function is executed, it returns an object of class `vibe`. Next to built-in `print` and `summary` functions, the extension also features a `plot` function, which creates a neat display of the results. See Figures 5.1 and 5.3 for examples.

5.7 Simulation

In both original relative weights proposals by J. Johnson (2000) and its logistic regression extension by Tonidandel and LeBreton (2010), as well as in Grömping (2015), the authors were able to show that relative weights analysis yields very similar results to the full-subset metric hierarchical partitioning. This section fulfills a similar purpose by confirming that extending relative weights beyond linear models does not diminish their accuracy.

As part of the proposed simulation study, the performance of RWA and hierarchical partitioning will be tested on the two different model classes: GLM and GAMLSS (with linear predictors). In the first scenario (a), which concerns only GLM, we assume the expected value of the parametric distribution to be connected to explanatory variables x_1, x_2, x_3 in the following way:

$$\begin{aligned} \eta_1 &= \mathbf{X}\boldsymbol{\beta}_1 \\ h(\eta_1) &= \mathbb{E}(y)x \end{aligned} \tag{5.11}$$

where $h(\cdot)$ represents a pre-specified response function that connects the explanatory variables to the expected value of y, upholding the correct parameter space. It varies between distributions, and equals to the default functions provided by the respective estimation software. The covariates $x_1, \ldots, x_3$ are distributed as follows:

$$\mathbf{X} = (x_1, x_2, x_3)^\intercal \sim N\left(\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0.50 & 0.20 & 0.40 \\ 0.20 & 0.50 & 0.20 \\ 0.40 & 0.20 & 0.50 \end{bmatrix}\right), \tag{5.12}$$

which includes correlations of 0.4 (x_1&x_2, x_2&x_3) and 0.8 (x_1&x_3). The corresponding regression coefficients decrease in size and sum to one to simulate a "relative share": $\boldsymbol{\beta}_1 = (\beta_1, \beta_2, \beta_3)^\intercal = (0.5, 0.\overline{33}, 0.1\overline{67})$. We estimate a GLM using the `R` function `gam()` of package `mgcv` (Wood, 2011) using the correct model specification, and calculate both RWA and hierarchical partitioning to obtain the share of relative importance of each variable. The difference between RWA and hierarchical partitioning is calculated $m_{\text{RWA}} - m_{\text{HP}}$.

In the second scenario (b), we use the same coefficients $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3)^\intercal$ to connect x_q to y, but this time to its first parameter θ_1 in $y \sim D(\theta_1, \ldots, \theta_K)$, instead of the expected value as in a). We choose a variety of different response distributions, visible in Table 5.3. The model specification is as follows:

$$\begin{aligned} \eta_1 &= \mathbf{X}\boldsymbol{\beta}_1 \\ h_1(\eta_1) &= \theta_1, \end{aligned} \tag{5.13}$$

while all other parameters of the distribution designed as fixed. Similar to the first scenario, we estimate the full model using the correct model specification and obtain the difference $m_{\text{RWA}} - m_{\text{HP}}$. This time we utilize `gamlss` (Stasinopoulos & Rigby, 2007) for model estimation, which yields the possibility of connecting distributional parameters to predictors.

In the third scenario (c), we focus on variable importance for the second parameter of y, which typically corresponds to the scale of the distribution. Our chosen distributions are equal as in (a). The simulation specification for all models is as follows:

$$\begin{aligned} \eta_1 &= \mathbf{X}\boldsymbol{\beta}_1 \\ \eta_2 &= \mathbf{X}\boldsymbol{\beta}_2 \\ h_1(\eta_1) &= \theta_1 \\ h_2(\eta_2) &= \theta_2, \end{aligned} \tag{5.14}$$

with the coefficients $\boldsymbol{\beta}_1$ retaining the same values as in (5.11) and $\boldsymbol{\beta}_2$ being equal

to $\boldsymbol{\beta}_1$: $\boldsymbol{\beta}_1 = \boldsymbol{\beta}_2 = (0.5, 0.\overline{33}, 0.1\overline{67})$. We obtain both variable importance metrics RW and HP for the second parameter only, causing the submodels of HP to vary in predictors for θ_2 only, while the model specification for θ_1 stays as in (5.14). Further, the orthogonalization and models estimated for the relative weights metric also concern only changes in predictors for the second parameter θ_2, while the predictors for θ_1 are not orthogonalized.

In Figure 5.2 the results of all three scenarios are shown. On the left side of the plot, boxplots depict the difference between estimated relative weights (m_{RWA}) and estimated variable importance according to hierarchical partitioning (m_{HP}) (x-axes) as well as the specific target distribution (y-axes). The boxplots are displayed in groups of three, which correspond to the three explanatory variables x_1, x_2, x_3 (color code in legend). The right side of the graph includes a flipped bar plot depicting the average absolute difference over all three explanatory variables $m_{\text{RWA}} - m_{\text{HP}}$ on the x-axis. Average standard errors are included in parentheses.

As visible in Figure 5.2, the first and second scenarios see very close results between HP and RWA, with most boxplots completely enclosed in the dashed lines at $x = \{-0.05, 0.05\}$. Most distributions see an average difference of around 0.015 percentage points (visible on the right side), with only the beta distribution showing a slightly lower average difference of 0.006. This strengthens the approximation accuracy of relative weights, indicating that the all-subset metric hierarchical partitioning is not needed. In the third scenario, the difference between RWA and HP is slightly greater. Most distributions exhibit a gap between 0.01 and 0.045, with the Gamma distribution reporting the highest average deviation of 4.5% (percentage points) between RWA and the all-subset metric HP. The lowest average difference is found in the Inverse Gamma distribution models with 0.014 percentage points. It has to be noted that although the average difference $m_{\text{RWA}} - m_{\text{HP}}$ is low, it varies from covariate to covariate: x_1 mostly experiences negative differences, while the difference is mostly positive for x_2 and x_3.

In conclusion, we can see that the approximation of hierarchical partitioning with relative weights is effective, although the accuracy is higher in variable importance for the first parameter. We can thus confidently use relative weights as an alternative to

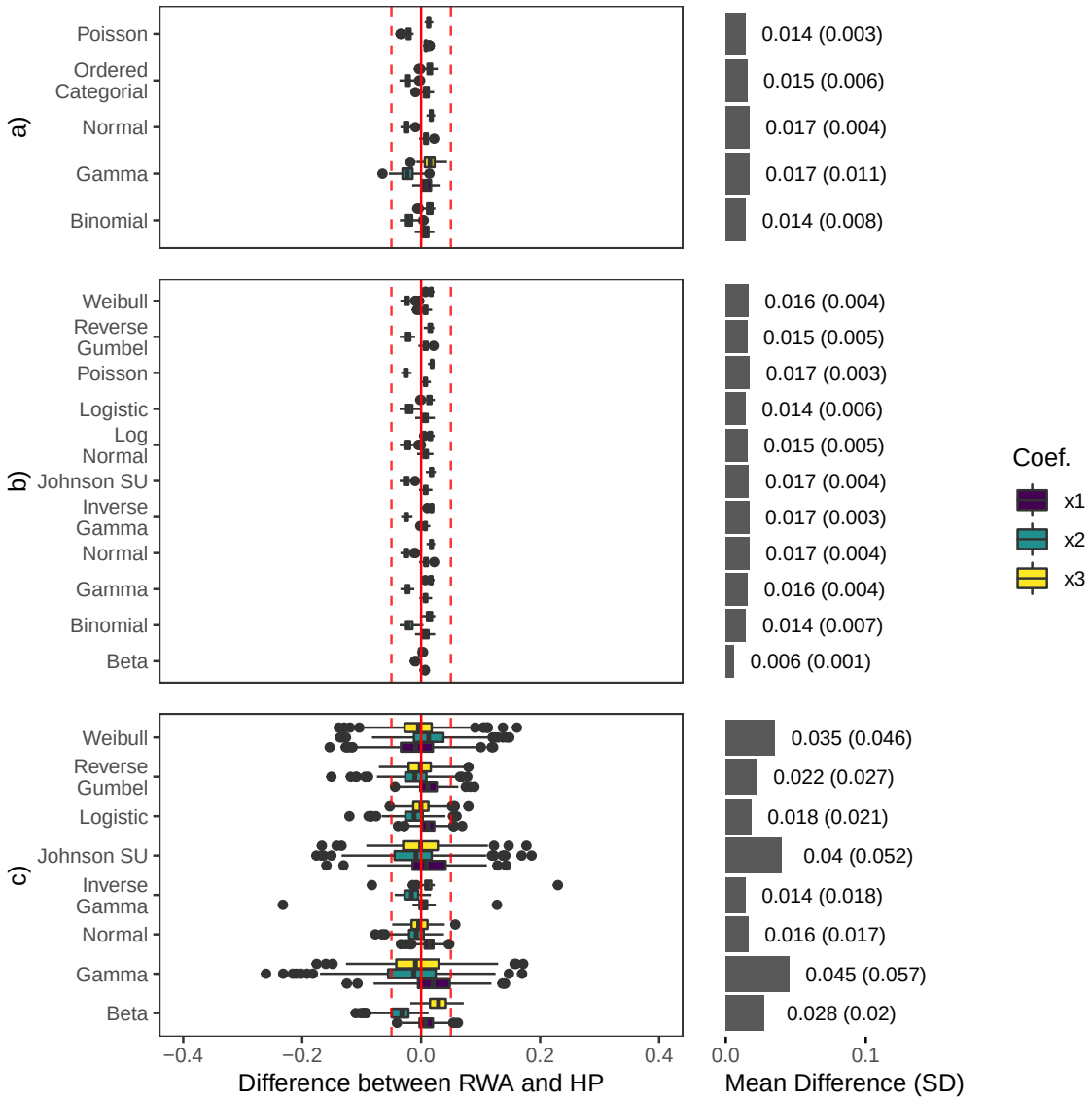

Figure 5.2: Results of simulations according to Scenarios (5.11) labelled a, (5.13) labelled b and (5.14) labelled c. The y-axes in all three plots denote different target distributions, whereas there are two x-axes, denoting the difference between relative weights and hierarchical partitioning $m_{\text{RWA}} - m_{\text{HP}}$ (left) and the mean absolute difference on the right side. Colors represent different covariates x_1, x_2, x_3. A vertical zero line is included in red, with dashed lines at $x = \{-0.05, 0.05\}$. The right side of the plots includes average deviations between the two metrics displayed in bars, with the average standard deviation in parentheses. $N = 200$.

hierarchical partitioning.

Table 5.3 provides an overview of the model classes and distributions used in the simulation. Additionally, the mean computing time of both relative weights (RW)

and hierarchical partitioning (HP) in seconds using a single-core Intel©Xeon©CPU E5-4660 v4 @ 2.20GHz is given in columns 3 and 4, with relative weights usually taking only a quarter of the time of hierarchical partitioning, even in this case of low predictor count. We expect to see an even bigger difference in computing time when the predictor space is high-dimensional. For the many distributions and model classes chosen, our relative weights extension thus provides an effective alternative to hierarchical partitioning, using only a fraction of the computing time.

	Class	Family	HP time	RW time
1	gam	Binomial	0.33	**0.07**
2	gam	Gamma	1.65	**0.33**
3	gam	Normal	0.11	**0.03**
4	gam	Ordered Categorial	1.15	**0.32**
5	gam	Poisson	0.36	**0.07**
6	gamlss	Beta	0.39	**0.08**
7	gamlss	Binomial	0.22	**0.05**
8	gamlss	Gamma	0.44	**0.09**
9	gamlss	Inverse Gamma	0.26	**0.07**
10	gamlss	Johnson SU	0.41	**0.09**
11	gamlss	Log Normal	0.18	**0.04**
12	gamlss	Logistic	0.37	**0.08**
13	gamlss	Normal	0.71	**0.17**
14	gamlss	Poisson	0.18	**0.04**
15	gamlss	Reverse Gumbel	0.43	**0.09**
16	gamlss	Weibull	0.53	**0.12**

TABLE 5.3: Distributions included in simulation and corresponding variable importance computing time in seconds per estimated variable importance metric.

5.8 Application: Childhood malnutrition in India

In a report provided by the Demographic and Health Surveys (International Institute for Population Sciences, 2007), socioeconomic and health-related information of individual children and their parents was published. Hofner, Mayr, and Schmid (2016) utilize the published dataset to understand the relations behind childhood malnutrition and socioeconomic variables. In detail, they focus on a response variable called `stunting`, measuring the child's stunted growth in relation to the population average.

It is measured as a Z score in the following way:

$$Z_i = \frac{\text{he}_i - \text{he}_{\text{med}}}{s}, \tag{5.15}$$

where he_i measures the child's current height and $\text{he}_{\text{med}}, s$ denote the median and standard deviation of the reference population height, respectively. Hofner et al. (2016) then connect this dependent variable to

- `cbmi`: The child's body mass index,
- `cage`: The child's age in months,
- `mbmi`: The mother's body mass index,
- `mage`: The mother's age in years and
- `mcdist`: The families' district of living

with nonlinear smooth splines using the `R` extension `gamboostLSS` (Thomas et al., 2018). After estimation of their model, which due to incorporating a boosting approach also includes an effect selection part, all of the above variables were selected to be connected to both the location and scale parameters of the dependent variable.

In this section we use the model fitted by Hofner et al. (2016) and determine which of the variables most significantly contribute to explaining the dependent variable. For ease of estimation, we restrict ourselves to three Indian districts: Mumbai, Delhi and Aizawl. Similar to the above authors, we assume the dependent variable y (`stunting`) to follow a Gaussian distribution. We then connect both parameters to variables in the following way:

$$\begin{aligned} y &\sim N(\theta_1, \theta_2) \\ \theta_1 &= \beta_{10} + \sum_{q=1}^{4} \beta_{1q} x_q + f_{15}(x_5; \beta_{15}, \beta_{16}) \\ \log(\theta_2) &= \beta_{20} + \sum_{q=1}^{4} \beta_{2q} x_q + f_{25}(x_5; \beta_{25}, \beta_{26}), \end{aligned} \tag{5.16}$$

Covariates	θ_1	θ_2
(Intercept)	**2.19***	0.32
	(**1.10**)	(0.55)
cbmi	**−0.20*****	−0.03
	(**0.05**)	(0.02)
cage	**−0.05*****	0.01
	(**0.01**)	(0.01)
mbmi	0.04	−0.00
	(0.03)	(0.01)
mage	0.01	0.01
	(0.02)	(0.01)
mcdist_labDelhi	**−0.75****	**0.32***
	(**0.28**)	(**0.14**)
mcdist_labMumbai (Greater Mumbai)	−0.46	−0.03
	(0.24)	(0.14)

***$p < 0.001$; **$p < 0.01$; *$p < 0.05$

TABLE 5.4: Fitted regression coefficients of the Indian malnutrition model specified in (5.16) for modeling parameters θ_1 ($\beta_{10}, \ldots, \beta_{16}$) and θ_2 ($\beta_{20}, \ldots, \beta_{26}$) in columns two and three, respectively. Standard errors are given in parentheses.

where $\mathbb{E}(y) = \theta_1$, $\text{Var}(y) = \theta_2^2$ and $x_1 \ldots x_4$ represent the numeric variables `cbmi`, `cage`, `mbmi`, `mage` and x_5 denotes the categorical district variable `mcdist`, modeled using a categorical term $f_{k5}(\cdot)$. To ensure compatibility with the `vibe` package, a linear connection between y and x_q is assumed.

Table 5.4 provides an overview of the coefficients, estimated using `gamlss` (Rigby & Stasinopoulos, 2005). Even though all variables were selected in the boosted regression model by Hofner et al. (2016), only a few of their coefficients remain significantly different from zero when included in the model as a linear term.

Although we obtain regression results in Table 5.4, it is still unclear which variable contributes the most to childhood malnutrition, both in location and shape. To answer this question, we use the package `vibe`. In contrast to the motivational example in Section 5.3, the number of variables per parameter is relatively low, which means that the all-subset metric hierarchical partitioning is not out of reach. To obtain variable importance results, we execute the following code:

```
vb_gm <- vibe(mod_gm, metric = "hp", ncores = 3, progress = FALSE)
```

Covariate Name	Parameter	I. Effects	I. Effects (fraction)
cbmi	mu	0.06	0.26
cage	**mu**	**0.12**	**0.53**
mbmi	mu	0.01	0.05
mage	mu	0.00	0.02
mcdist_lab	mu	0.03	0.14
cbmi	sigma	0.02	0.29
cage	sigma	0.00	0.02
mbmi	sigma	0.00	0.00
mage	sigma	0.01	0.07
mcdist_lab	**sigma**	**0.05**	**0.62**

Table 5.5: Variable importance results for the Indian malnutrition model specified in (5.16), with independent variable contributions and its corresponding scaled fractions, displayed in columns three and four, respectively. This software output uses the μ, σ notation which we define as θ_1, θ_2.

```
summary(vb_gm)
plot(vb_gm, perc = FALSE)
```

Table 5.5 shows the raw and percentage variable importance results, whereas Figure 5.3 displays them graphically. We can now see that the impact of the child's age is the most important factor for malnutrition, while the body mass index comes second. For the variance parameter, the child's district seems to have the most impact.

5.9 Conclusion

With this paper, we propose an extension of hierarchical partitioning and relative weights for use with GLM and GAMLSS with linear predictors. Using both metrics, applied researchers can make better inferences with regard to the covariate effects with the highest impact on the dependent variable's location, scale and shape parameters in relation to other variables. Moreover, relative weights are highly scalable with the number of covariates, making them suitable for high-dimensional applications.

Furthermore, an implementation in `R` is presented, designed to be easy-to-use when the fitted regression model is estimated. The applied researcher simply needs to provide the fitted `glm`, `gam` or `gamlss` object and select the preferred variable importance metric. A function to visualize the results is readily available.

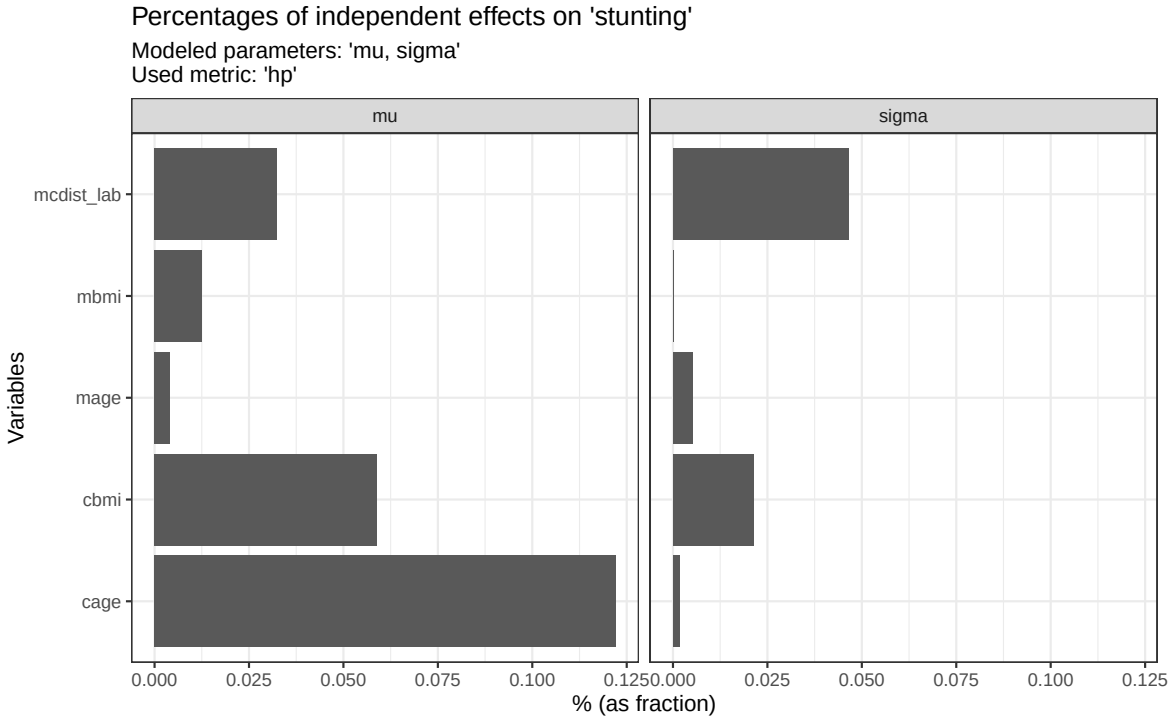

FIGURE 5.3: Plot output of variable importance figures for the Indian malnutrition model, with μ, σ notation which we define as θ_1, θ_2.

Some drawbacks remain. First, even though many of the model classes are supported, the effects are assumed to be linear. In many cases, this is a strong simplification of reality. While hierarchical partitioning can in theory easily be expanded to non-linear effects, further research on extending the relative weights algorithm proposed in this paper to non-linear effects (e.g. parametric base splines) would be beneficial. Second, the discussed variable importance metrics assume that the optimal model has already been found. The variable selection step in applied modeling is therefore not replaced. Third, `vibe` could be extended to support even more `R` packages and goodness-of-fit metrics.

Ultimately, the proposed variable importance metrics provide a reliable foundation to provide researchers with more information about their included variables on a distribution of interest. Due to the open distributional interface of GAMLSS, variable importance can be applied to any conceivable metric of interest of the target distribution, such as the median or the kurtosis. Thanks to its flexibility, the variable

importance implementation `vibe` has the potential to become regularly utilized as a crucial tool for the applied researcher.

6

Conclusion and discussion

Following an introduction to the class of distributional regression models, including a formal introduction and a cross-sectional overview of current developments, this thesis went on to make three distinct contributions to the field.

The first contribution was concerned with visualizing distributional regression models. It was recognized that the increased parameter specification flexibility of the model class brings additional challenges in understanding how estimated effects influence the moments of a distribution, since effects can be non-parametric, link functions transform the effects and modeled parameters may not directly equate to distributional moments. A framework and subsequent implementation in `R` was proposed and published. This contribution fills a gap in the model class and has the potential to become the prime visualization tool for regression beyond the mean. However, although already extensive, it can be developed further. Multivariate (copula) distributions are receiving increased

attention but are currently incompatible with `distreg.vis`. Also, `R` packages with new effect types are being developed and also not currently compatible with `distreg.vis`. That could be targeted next.

In Chapter 4, the focus shifted to the estimation of distributional regression models. The distribution of parameter maximum-likelihood estimates (MLEs) was analyzed and found to have a non-diagonal covariance matrix in most distributions outside the exponential family. It was observed that this causes a misspecification bias with a spillover effect, which influences regression coefficient estimates even when parameters were correctly specified. An orthogonalization framework for two-parameter distributions that relies on the numeric solution of a system of ordinary differential equations was laid out and verified using simulations, achieving unbiased estimation. While the proposed framework is robust and effective, it remains limited to two-parameter distributions. The extension to three and more parameters could be researched. Further, implementation of the method in well-known regression software, like `gamlss` or `bamlss`, would make it easier to use this contribution.

The third contribution of this thesis concerns the interpretation of distributional regression models. While modern regularization techniques are useful in arriving at correct and optimal model specifications, the question of variable importance remained to be answered. In Chapter 5, common methods for ranking variable importance in linear regression models were reviewed, with two approaches standing out as the most suitable. These two algorithms were extended to generalized linear models (GLM, Nelder & Wedderburn, 1972) and generalized additive models for location, scale and shape (GAMLSS, Rigby & Stasinopoulos, 2005) with some adjustments, verified in simulations and implemented in an `R` package called `vibe`. Potential improvements concern compatibility: currently, `vibe` supports the base `R` function `glm()` as well as the packages `mgcv` and `gamlss`. This could be extended to other, more specialized distributional regression implementations. Further, the proposed algorithm could be extended to parametric non-linear effects like smooth splines.

On the whole, the contributions proposed in this thesis close important gaps in the field of distributional regression and will hopefully be well utilized in the future.

A

Appendix

A.1 Supplement to Chapter 3: *Interactively visualizing distributional regression models*

A.1.1 Extending `distreg.vis`

In its original conception, `distreg.vis` was written to accommodate both `bamlss` and `gamlss` models, which are both well suited for solving most distributional regression tasks. However, users might prefer other R packages for their model estimation. To ensure future package compatibility, `distreg.vis` was written such that it can be easily extended to work with new modules, thanks to a rather minimalistic interface with the distributional regression packages. This section serves as a reference for its realization, with a direct application to the `betareg` (Cribari-Neto & Zeileis, 2010) package.

distreg_checker()

The first function that requires extending is `distreg_checker`, which inspects whether the used model class is supported by `distreg.vis`. It always returns a Boolean value, and is part of `distreg.vis`' most core functions' error handling. Extension to the two new `betareg` model classes (`betareg`, `betatree`) is achieved as follows:

```
distreg_checker <- function(x) {
  if (is.character(x))
    obj <- get(x, envir = .GlobalEnv)
  else
    obj <- x
  if (is(obj, "bamlss"))
    return(TRUE)
  else if (is(obj, "gamlss"))
    return(TRUE)
  else if (is(obj, "betareg") | is(obj, "betatree"))
    return(TRUE)
  else
    return(FALSE)
}
```

dists.csv

After officially registering the new package in `distreg_checker`, more information about the newly supported distributions has to be provided. The basis of all supported distributions is formed by the file `/inst/extdata/dist_df.csv`, which, after executing `data-raw/create_dist_table.R` becomes the `distreg.vis::dists` dataset, in which plot limits, the corresponding model class, distribution type and other knowledge about the distribution are stored. Due to `betareg` only focusing on the beta distribution, it extends `dist_df.csv` by just one line. Shown below are the first and last two lines of `dist_df.csv`:

```
R> file <- "inst/extdata/dist_df.csv"
R> c(head(readLines(file, warn = FALSE), 2),
+    tail(readLines(file, warn = FALSE), 2))
```

```
[1] "dist_name;class;implemented;moment_funs;type_limits;l_limit;
u_limit;type"
[2] "BB;gamlss;FALSE;FALSE;both_limits;0;10;Discrete"
[3] "quant;bamlss;FALSE;FALSE;NA;NA;NA;NA"
[4] "betareg;betareg;TRUE;TRUE;both_limits;0;1;Continuous"
```

For further information on `dists`, visit Section 3.4.1 and its description of Table 3.1.

fam_fun_getter()

In order for `plot_dist()` to be able to display the probability density function (PDF) or cumulative distribution function (CDF), the correct underlying R functions need to be implemented. This is achieved by `fam_fun_getter()`, which, by demand, generates functions that yield the corresponding PDF (`type == "d"`), PDF (`type == "p"`) and quantile values (`type == "q"`). Fortunately, both `bamlss` and `gamlss` already provide these functions inside their family objects (e.g. `gamlss.dist::dNO()`). For `betareg`, these functions are implemented directly into *fam_fun_getter()*. An excerpt of the function, returning the PDF, is shown below.

```
fam_fun_getter <- function(fam_name, type) {
  ...
  if (is.betareg(fam_name)) {
    if (type == "d") {
      fun <- function(x, par) {
        alpha <- par[["mu"]] * par[["phi"]]
        beta <- (1 - par[["mu"]]) * par[["phi"]]
        return(dbeta(x, alpha, beta))
      }
    }
  ...
  }
  ...
  return(fun)
}
```

preds()

With `preds()`, predicted distributional parameters are computed. Due to different

implementations of the distributional regression packages' `predict.object()` functions, some code is needed to consolidate the output. In `preds()` this means that the return value consists of a `data.frame` object with the columns representing the distributional parameters, while the rows stand for an individual prediction based on a unique covariate combination.

The only exception to this rule is allowed for sample-based predictions, where each covariate combination consists of one element in a list, which in turn stores a `data.frame` object with parameters in columns, and one sample per row. Currently, this is only possible with `bamlss` models.

The following code is part of `preds()` ensuring the correct output of `predict.betareg` when used with a `betareg` model:

```
preds <- function(model, newdata = NULL, what = "mean",
                  vary_by = NULL) {
...
    else if (is(model, "betareg") | is(model, "betatree")) {
    if (what == "mean") {
      pred_par <- data.frame(
        mu = predict(model, newdata = newdata, type = "response"),
        phi = predict(model, newdata = newdata, type = "precision")
      )
    } else if (what != "mean") {
      stop("betareg uses ML, so no samples available.")
    }
...
return(pred_par)
}
```

moments()

The advantage of `distreg.vis` over other packages is that it can calculate marginal effects on the expected moments of a distribution, not only on the parameters. Transforming the parameters to the moments is done with the `moments()` function. When used in combination with `gamlss` and `bamlss` models, `moments()` looks for the correct distributional family object (e.g. `bamlss::lognormal_bamlss()` and then retrieves

the correct functions to then calculate the first two moments from the parameters.

In case parameter predictions include samples from the posterior distribution (currently only possible with `bamlss`), the samples are first transformed to the moments, after which the average is calculated. Independently of whether samples are provided or only mean values, the output of the `moments()` function consists of two columns, with the names `Expected_Value` and `Variance`. Each row stands for one distinct parameter combination provided in `par`. A third column is added if the argument `ex_fun` was provided, which calculates user-defined metrics that are functions of the samples.

In the case of `betareg`, the moment functions were implemented directly in `distreg.vis`, as detailed in the following code snippet:

```
  moments <- function(par, fam_name, what = "mean", ex_fun = NULL) {

  ...
    if (is.betareg(fam_name)) {

      if (what != "mean")
        stop("In betareg models only option for argument ‘what’
is ‘mean’.")

      if (!funworks) {
        moms_raw <- apply(par, 1, FUN = function(x) {
          ex <- x[["mu"]]
          vx <- x[["mu"]] * (1 - x[["mu"]]) / (1 + x[["phi"]])
          return_vec <- c(Expected_Value = ex, Variance = vx)
          return(return_vec)
        })
      }

      if (funworks) {
        moms_raw <- apply(par, 1, FUN = function(x) {
          ex <- x[["mu"]]
          vx <- x[["mu"]] * (1 - x[["mu"]]) / (1 + x[["phi"]])
          ex_fun_vals <- fun(as.list(x))
          return_vec <- c(Expected_Value = ex,
```

```
                          Variance = vx,
                          ex_fun = ex_fun_vals)
          names(return_vec)[names(return_vec) == "ex_fun"] <- ex_fun
        })
      }
      moms <- as.data.frame(t(moms_raw), row.names = rnames)
    }
    if (exists("moms"))
      return(as.data.frame(moms))
...
}
```

To complete the support for a new distributional regression package, one has to extend `moments()` such that it knows where to find the functions which calculate the moments of the distributional parameters, and how to execute them. If this step is completed, the functionalities of `plot_dist()` and `plot_moments()` will be fully available to the new distributional regression package.

A.1.2 Special case of plot_dist()

One more special case exists within using `plot_moments()`, which appears when the family of multinomial distributions is used for describing the dependent variable. Due to the nature of the variable's parameters π_i as the probability to fall into category i always summing up to 1, we can visualize the impact of a continuous variable differently. Figure A.1 shows such a case, where the influence of a continuous variable named `norm1` on a categorical variable, both stemming from a simulated dataset, was graphically displayed.

The three windows in Figure A.1 represent three different covariate combinations specified in `pred_data`. On the x-axis, we see the range of `int_var`, while the y-axis denotes probabilities. Every colored area represents the expected probability for a new observation to fall into any of the categories of the dependent variable. Due to the target distribution being highly dependent on the simulated explanatory variables, its probabilities change considerably over the range of the variable `norm2`.

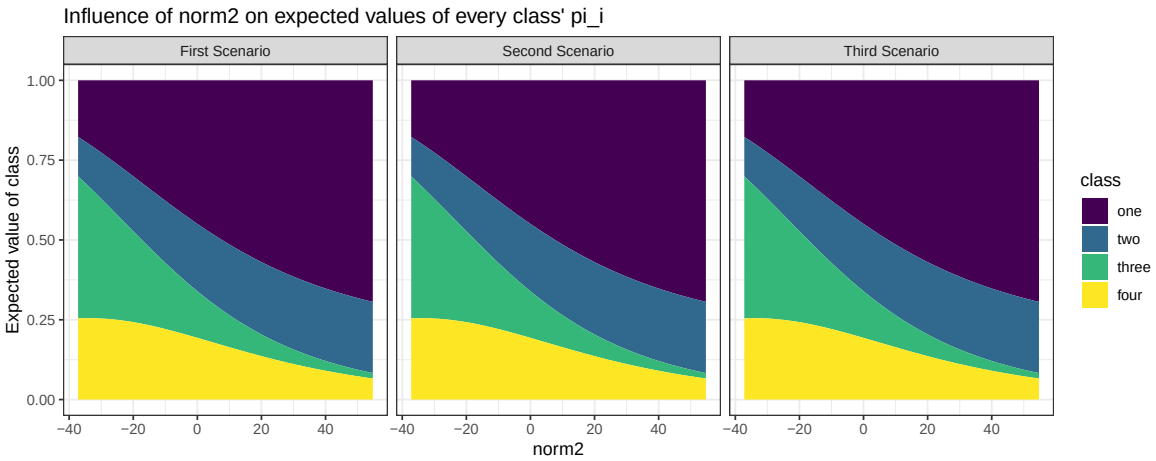

Figure A.1: Outcome of `plot_moments()` for a model with a simulated multinomial target distribution. In this case, each scenario gets its own window, with the expected probabilities to fall into each category ranging over the variable of interest.

A.1.3 Additional graphs

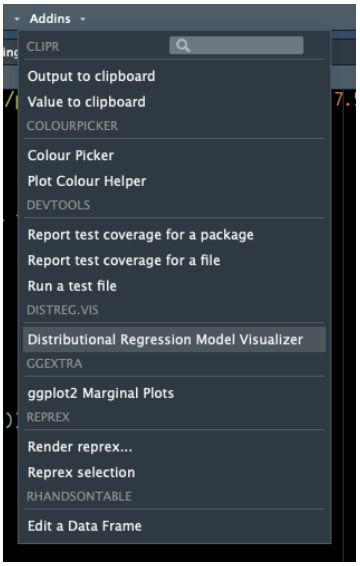

Figure A.2: Button to start the main application of distreg.vis in RStudio.

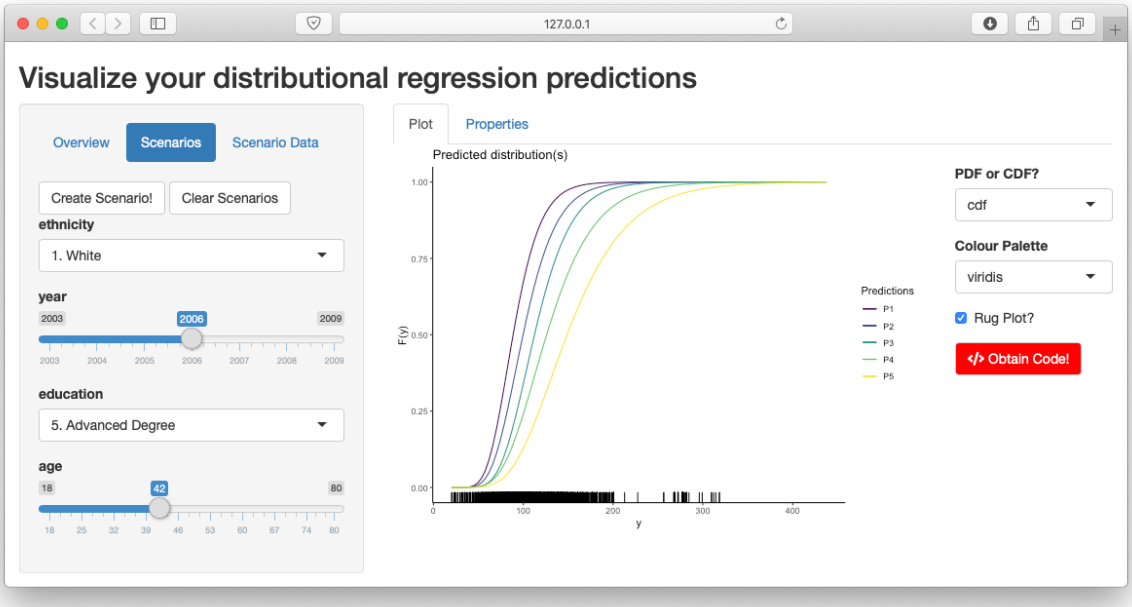

FIGURE A.3: CDF plot output for different education levels based on the `Wage` dataset.

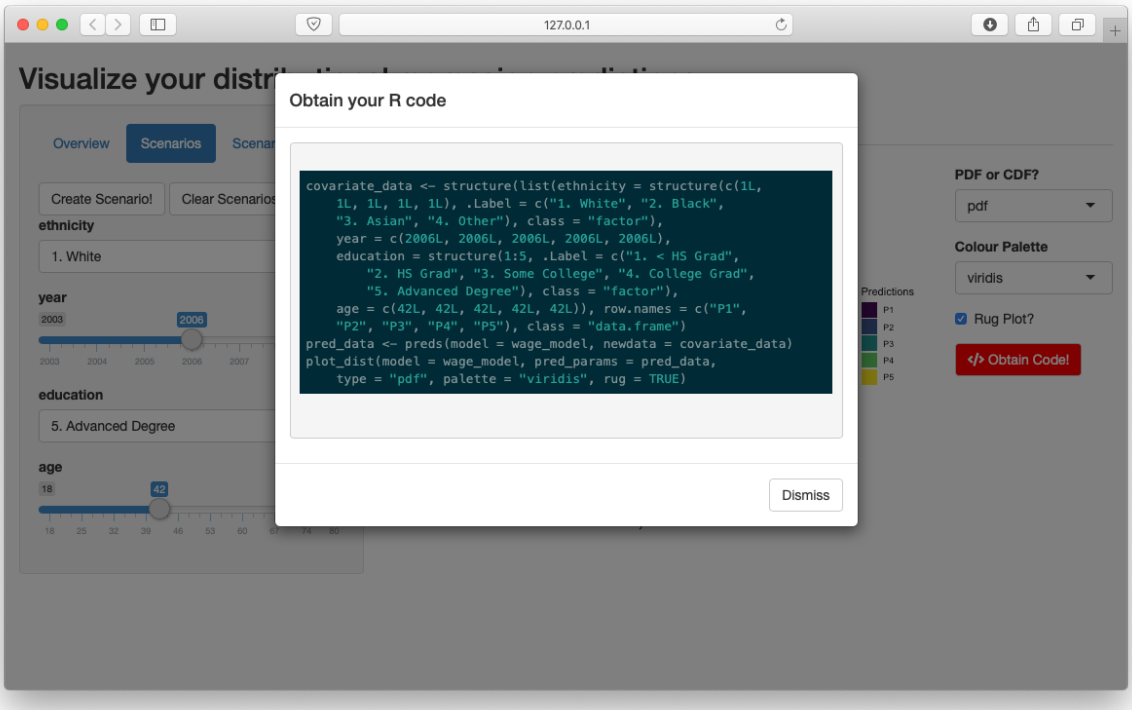

FIGURE A.4: Modal window with formatted and highlighted code after pressing the "Obtain Code!" button

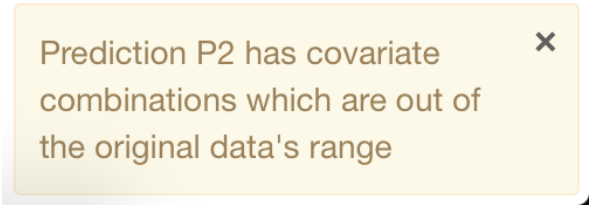

Figure A.5: Warning message when specifying covariate combinations which are out of range.

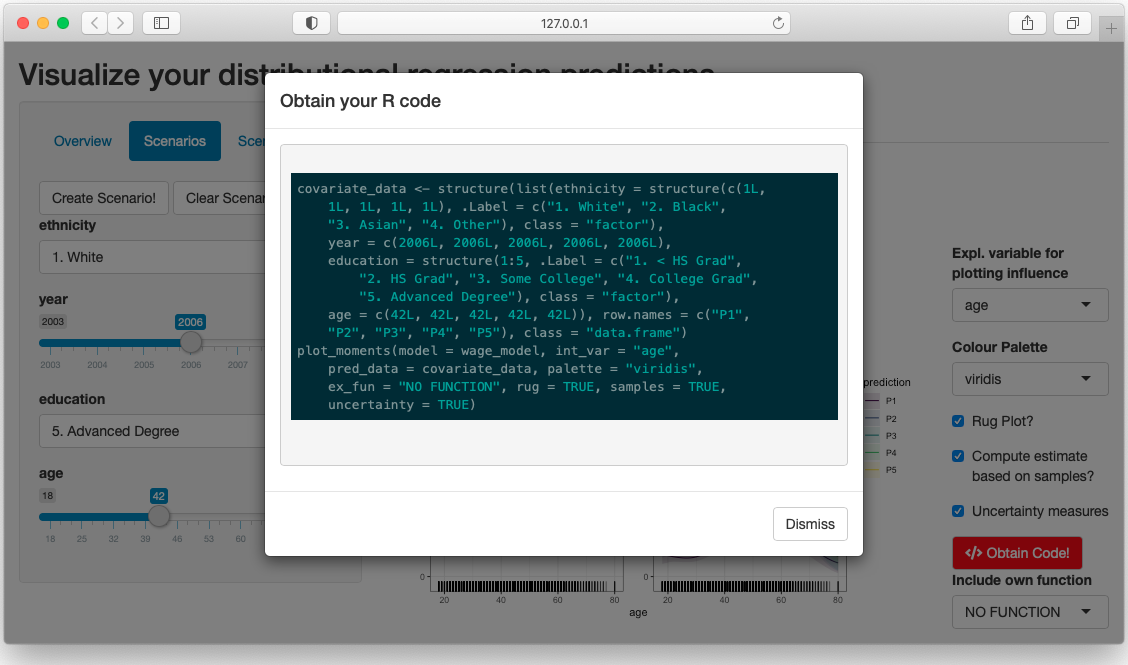

Figure A.6: Modal window to display code for reproducing the influence plot.

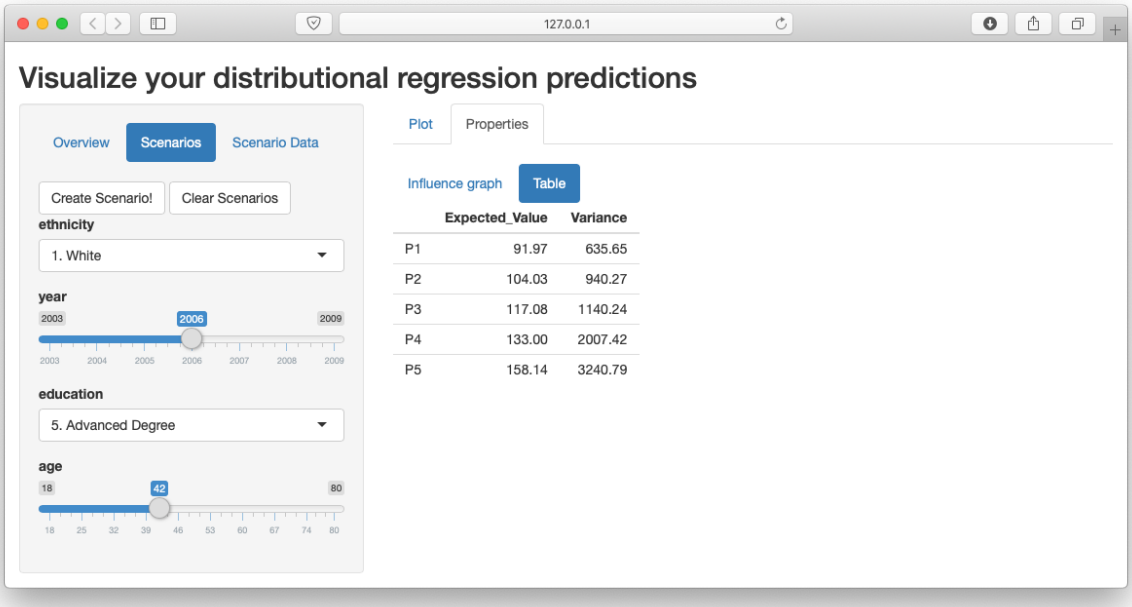

FIGURE A.7: Expected value and variance for predicted distributions based on specified covariate combinations.

A.2 Supplement to Chapter 4: *Parameter orthogonality transformations in distributional regression models*

A.2.1 Derivatives of the inverse-gamma location/scale PDF and its ODE

This parametrization of the inverse gamma distribution as given by Stasinopoulos et al. (2017) has the following PDF:

$$p(y \mid \theta_1, \theta_2) = \frac{\theta_1^{\frac{1}{\theta_2^2}} \left(\frac{1}{\theta_2^2}+1\right)^{\frac{1}{\theta_2^2}} y^{-\left(\frac{1}{\theta_2^2}+1\right)}}{\Gamma\left(\frac{1}{\theta_2^2}\right)} \exp\left(-\frac{\theta_1 \cdot \left(\frac{1}{\theta_2^2}+1\right)}{y}\right). \tag{A.1}$$

Its log-density function is:

$$\log p(y \mid \theta_1, \theta_2) = \frac{\log(\theta_1) + \log\left(\frac{1}{\theta_2^2} + 1\right)}{\theta_2^2} - \log\left(\Gamma\left(\frac{1}{\theta_2^2}\right)\right) - \frac{\theta_1\left(\frac{1}{\theta_2^2} + 1\right)}{y} + \left(-\frac{1}{\theta_2^2} - 1\right)\log(y).$$

The first derivative of $\log p(y \mid \theta_1, \theta_2)$ with regard to the second parameter is:

$$\frac{\partial \log p}{\partial \theta_2} = -\frac{2\left(\log(\theta_1) + \log\left(\frac{1}{\theta_2^2} + 1\right)\right)}{\theta_2^3} - \frac{2}{\left(\frac{1}{\theta_2^2} + 1\right)\theta_2^5} + \frac{2\psi^{(0)}\left(\frac{1}{\theta_2^2}\right)}{\theta_2^3} + \frac{2\theta_1}{\theta_2^3 y} + \frac{2\log(y)}{\theta_2^3}. \tag{A.2}$$

while the second cross-derivative is as follows:

$$\frac{\partial \log p}{\partial \theta_1 \partial \theta_2} = \frac{2}{\theta_2^3 y} - \frac{2}{\theta_1 \theta_2^3}\ . \tag{A.3}$$

Taking the negative expected value of this term we arrive at:

$$\begin{aligned} \mathbb{E}_{12} = \mathbb{E}\left(-\frac{\partial^2 \log p}{\partial \theta_1 \partial \theta_2}\right) &= -\frac{2}{\theta_2^3} \cdot \mathbb{E}\left(\frac{1}{y}\right) + \frac{2}{\theta_1 \theta_2^3} \\ &= -\frac{2}{\theta_2^3} \cdot \frac{\frac{1}{\theta_2^2}}{\theta_1\left(\frac{1}{\theta_2^2} + 1\right)} + \frac{2}{\theta_1 \theta_2^3}, \end{aligned} \tag{A.4}$$

since $\mathbb{E}\left(\frac{1}{y}\right) = \frac{\frac{1}{\theta_2^2}}{\theta_1\left(\frac{1}{\theta_2^2}+1\right)}$.

The second derivative of $\log p$ with respect to θ_2 is:

$$\begin{aligned} \frac{\partial \log p}{\partial \theta_2^2} = &\frac{6\left(\log(\theta_1) + \log\left(\frac{1}{\theta_2^2} + 1\right)\right)}{\theta_2^4} - \frac{10}{\theta_2^6} \\ &- \frac{4}{\left(\frac{1}{\theta_2^2} + 1\right)^2 \theta_2^8} + \frac{14}{\left(\frac{1}{\theta_2^2} + 1\right)\theta_2^6} - \frac{6\theta_1}{\theta_2^4 y} - \frac{6\log(y)}{\theta_2^4}. \end{aligned} \tag{A.5}$$

which has the following negative expected value:

$$\begin{aligned}\mathbb{E}_{22} = \mathbb{E}\left(-\frac{\partial^2 \log p}{\partial \theta_2^2}\right) &= -\frac{6\left(\log(\theta_1) - \log\left(\frac{1}{\theta_2^2}+1\right)\right)}{\theta_2^4} + \frac{10}{\theta_2^6} + \frac{4}{\left(\frac{1}{\theta_2^2}+1\right)^2 \theta_2^8} \\ &\quad - \frac{14}{\left(\frac{1}{\theta_2^2}+1\right)\theta_2^6} + \frac{6\theta_1}{\theta_2^4} \cdot \mathbb{E}\left(\frac{1}{y}\right) + \frac{6 \cdot \mathbb{E}(\log(y))}{\theta_2^4} \\ &= -\frac{6\left(\log(\theta_1) - \log\left(\frac{1}{\theta_2^2}+1\right)\right)}{\theta_2^4} + \frac{10}{\theta_2^6} + \frac{4}{\left(\frac{1}{\theta_2^2}+1\right)^2 \theta_2^8} \\ &\quad - \frac{14}{\left(\frac{1}{\theta_2^2}+1\right)\theta_2^6} + \frac{6\theta_1}{\theta_2^4} \cdot \frac{\frac{1}{\theta_2^2}}{\theta_1\left(\frac{1}{\theta_2^2}+1\right)} + \frac{6 \cdot \log\left(\theta_1\left(\frac{1}{\theta_2^2}+1\right)\right) - \psi\left(\frac{1}{\theta_2^2}\right)}{\theta_2^4},\end{aligned} \tag{A.6}$$

where $\mathbb{E}\left(\frac{1}{y}\right) = \frac{\frac{1}{\theta_2^2}}{\theta_1\left(\frac{1}{\theta_2^2}+1\right)}$ and $\mathbb{E}(\log(y)) = \log\left(\theta_1\left(\frac{1}{\theta_2^2}+1\right)\right) - \psi\left(\frac{1}{\theta_2^2}\right)$.

Including the terms $\mathbb{E}_{12}$ (A.4) and $\mathbb{E}_{22}$ (A.6) in

$$\mathbb{E}_{12} + \mathbb{E}_{22}\frac{\partial \theta_2}{\partial \theta_1}, \tag{A.7}$$

which was stated also in (4.10) on Page 66, leads to the following ordinary differential equation (ODE):

$$\begin{aligned}-\frac{2}{\theta_2^3} \cdot \left(\frac{1}{\theta_1} + \frac{1}{\theta_2^2 \theta_1}\right) + \frac{2}{\theta_1 \theta_2^3} - \Bigg[&\frac{6\left(\log(\theta_1) - \log\left(\frac{1}{\theta_2^2}+1\right)\right)}{\theta_2^4} + \frac{10}{\theta_2^6} + \\ &\frac{4}{\left(\frac{1}{\theta_2^2}+1\right)^2 \theta_2^8} - \frac{14}{\left(\frac{1}{\theta_2^2}+1\right)\theta_2^6} + \frac{6}{\theta_2^4} + \frac{6}{\theta_2^2} + \frac{6\log(\theta_1)}{\theta_2^4} + \\ &\frac{6\log\left(\frac{1}{\theta_2^2}+1\right)}{\theta_2^4} + \frac{6\psi\left(\frac{1}{\theta_2^2}\right)}{\theta_2^4}\Bigg] \cdot \frac{\partial \theta_2}{\partial \theta_1} = 0.\end{aligned} \tag{A.8}$$

A.2.2 Orthogonal inverse gamma distribution as part of the exponential family

This is to show that the orthogonalized version of the inverse gamma distribution is indeed part of the exponential family. Here, we will use the canonical form definition as stated by McCullagh and Nelder (1983, p. 28):

$$p(y) = \exp[\{y\theta - b(\theta)\}/a(\phi) + c(y, \phi)] \tag{A.9}$$

Then, the probability density function as defined by 4.28 can be re-written as follows:

$$\begin{aligned} p\left(y \mid \omega_1\omega_2\right) &= \frac{\left(\omega_1, \omega_2\right)^{\omega_1}}{\Gamma\left(\omega_1\right)} y^{-\omega_1-1} \exp\left(-\frac{\omega_1\omega_2}{y}\right) \\ &= \exp\left[\omega_1 \log\left(\omega_1, \omega_2\right) - \log\left(\Gamma\left(\omega_1\right)\right) - \left(\omega_1 + 1\right)\log(y) - \frac{\omega_1\omega_2}{y}\right] \\ &= \exp\left[\omega_1 \log\left(\omega_1\right) + \omega_1 \log\left(\omega_2\right) - \log\left(\Gamma\left(\omega_1\right)\right) - \omega_1 \log(y) - \log(y) - \frac{\omega_1\omega_2}{y}\right] \\ &= \exp\left[\frac{-\frac{1}{y} \cdot (-\omega_2) + \log\left(\omega_2\right)}{1} + \left\{\omega_1 \log\left(\omega_1\right) - \log\left(\Gamma\left(\omega_1\right)\right) - \omega_1 \log(y) - \log(y)\right\}\right] \end{aligned} \tag{A.10}$$

A.3 Supplement to Chapter 5: *Variable importance in likelihood-based regression models*

A.3.1 Tables of subsection "Motivational Example"

Question label	Question content
Q.1	Rule & Regulations
Q.2	Security Systems
Q.3	Repair & Maintenance
Q.4	Behavior Of Warden
Q.5	Behavior Of Rector
Q.6	Satisfaction Of Management
Q.7	Calm & Peaceful Environment
Q.8	Clean Area
Q.9	Parking Lot
Q.10	Study Room
Q.11	Hostel Library
Q.12	Bathroom And Toilet
Q.13	Entertainment Facility
Q.14	Freedom
Q.15	Fine & Extra Charges
Q.16	24 Hrs. Electricity
Q.17	High Speed Wi-Fi Facility
Q.18	Annual Functions
Q.19	Quantity Of Food
Q.20	Meals Menu
Q.21	Quality Of Food
Q.22	Washroom
Q.23	Cleanness Of Mess
Q.24	Management System Of Mess
Q.25	Comfortable And Well Furnished Room
Q.26	Comfortable Beds
Q.27	Purity Of Drinking Water
Q.28	Availability Of Water
Q.29	Overall Satisfaction About Mess
rsat	Overall Satisfaction About Hostel

TABLE A.1: Survey question labels and content of the resident satisfaction dataset in Kolhapur, India. The column "question column" represent the hostel properties which residents rated. Data collection was conducted by Rajendra and Machhindra (2018).

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-5.90	1.34	-4.41	0.00
Gender	0.52	0.44	1.18	0.24
hostel_no	0.60	0.22	2.77	0.01
Q.1	0.23	0.19	1.23	0.22
Q.2	-0.38	0.18	-2.11	0.04
Q.3	-0.10	0.17	-0.61	0.54
Q.4	-0.02	0.19	-0.09	0.93
Q.5	-0.12	0.18	-0.66	0.51
Q.6	0.33	0.21	1.55	0.12
Q.7	0.26	0.17	1.55	0.12
Q.8	-0.18	0.18	-1.03	0.30
Q.9	0.44	0.20	2.23	0.03
Q.10	-0.41	0.19	-2.14	0.03
Q.11	-0.00	0.15	-0.02	0.98
Q.12	0.23	0.18	1.32	0.19
Q.13	0.01	0.14	0.05	0.96
Q.14	-0.16	0.20	-0.83	0.41
Q.15	0.23	0.16	1.42	0.15
Q.16	0.17	0.19	0.91	0.36
Q.17	0.12	0.14	0.90	0.37
Q.18	0.43	0.17	2.54	0.01
Q.19	0.06	0.19	0.29	0.77
Q.20	-0.29	0.24	-1.19	0.23
Q.21	0.21	0.17	1.22	0.22
Q.22	0.15	0.21	0.71	0.48
Q.23	0.22	0.21	1.03	0.30
Q.24	-0.29	0.21	-1.39	0.17
Q.25	-0.01	0.18	-0.06	0.95
Q.26	0.44	0.17	2.55	0.01
Q.27	0.14	0.18	0.75	0.45
Q.28	0.47	0.18	2.62	0.01
Q.29	0.16	0.20	0.80	0.42

Table A.2: Full regression output of model specified in (5.2). Long version of Table 5.1.

Covariate Name	Parameter	I. Effects	I. Effects (fraction)
Gender	mu	0.00	0.01
hostel_no	mu	0.04	0.09
Q.1	mu	0.01	0.02
Q.2	mu	0.01	0.02
Q.3	mu	0.00	0.00
Q.4	mu	0.01	0.02
Q.5	mu	0.00	0.00
Q.6	mu	0.02	0.05
Q.7	mu	0.02	0.06
Q.8	mu	0.01	0.01
Q.9	mu	0.04	0.09
Q.10	mu	0.01	0.01
Q.11	mu	0.00	0.01
Q.12	mu	0.02	0.04
Q.13	mu	0.00	0.01
Q.14	mu	0.00	0.01
Q.15	mu	0.01	0.03
Q.16	mu	0.01	0.03
Q.17	mu	0.01	0.03
Q.18	mu	0.03	0.07
Q.19	mu	0.00	0.01
Q.20	mu	0.00	0.01
Q.21	mu	0.01	0.03
Q.22	mu	0.01	0.03
Q.23	mu	0.01	0.02
Q.24	mu	0.00	0.01
Q.25	mu	0.01	0.03
Q.26	mu	0.04	0.09
Q.27	mu	0.01	0.03
Q.28	mu	0.04	0.10
Q.29	mu	0.00	0.01

TABLE A.3: Full output of "relative weights" calculated on the basis of the model specified in (5.2). Long version of Table 5.2.

References

Aitkin, M. (1987). Modelling variance heterogeneity in normal regression using GLIM. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *36*(3), 332–339.

7

Allen, J. T., Tippett, M. K., Kaheil, Y., Sobel, A. H., Lepore, C., Nong, S., & Muehlbauer, A. (2017). An Extreme Value Model for U.S. Hail Size. *Monthly Weather Review*, *145*(11), 4501–4519.

79

Barndorff-Nielsen, O. (1978). *Information and exponential families: In statistical theory*. New York: Wiley.

63

Batir, N. (2018). Inequalities for the inverses of the polygamma functions. *Archiv der Mathematik*, *110*(6), 581–589.

64

Brezger, A., Kneib, T., & Lang, S. (2005). BayesX: Analyzing Bayesian structured additive regression models. *Journal of Statistical Software*, *14*(11), 1–22.

11, 57

Brezger, A., & Lang, S. (2006). Generalized structured additive regression based on Bayesian P-splines. *Computational Statistics & Data Analysis*, *50*(4), 967–991.

28

Bureau of Meteorology. (2021). *Summary statistics Queenstown (7xs)*. Retrieved from

http://www.bom.gov.au/climate/averages/tables/cw_097034.shtml
79

Chang, W., Cheng, J., Allaire, J., Xie, Y., & McPherson, J. (2018). shiny: Web application framework for R [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=shiny (R package version 1.2.0)
15

Chevan, A., & Sutherland, M. (1991). Hierarchical partitioning. *The American Statistician*, *45*(2), 90–96.
93

Cribari-Neto, F., & Zeileis, A. (2010). Beta regression in R. *Journal of Statistical Software*, *34*(2), 1–24.
vi, 11, 14, 26, 88, 113

De Bastiani, F., Stasinopoulos, M. D., & Rigby, R. A. (2018). gamlss.spatial: Spatial terms in generalized additive models for location scale and shape models [Computer software manual]. (R package version 2.0.0)
27

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, *7*(1), 1–26.
80

Eilers, P. H., & Marx, B. D. (1996). Flexible smoothing using B-splines and penalized likelihood. *Statistical Science*, *11*(2), 89–121.
17

Estrella, A. (1998). A new measure of fit for equations with dichotomous dependent variables. *Journal of Business & Economic Statistics*, *16*(2), 198–205.
91, 94, 97

Fabbris, L. (1980). Measures of predictor variable importance in multiple regression: An additional suggestion. *Quality and Quantity*, *14*(6), 787–792.
86, 94

Fahrmeir, L., Kneib, T., Lang, S., & Marx, B. (2013). *Regression: Models, methods and applications*. Berlin: Springer.

6, 8, 26, 87

Fox, J., & Weisberg, S. (2019). *An R companion to applied regression* (3rd ed.). Thousand Oaks, California: Sage.

15

Freund, Y., Schapire, R., & Abe, N. (1999). A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, *14*(5), 771–780.

86

Garnier, S. (2017). viridis: Default color maps from 'matplotlib' [Computer software manual]. (R package version 0.4.0)

34, 46

Garver, M., & Williams, Z. (2020). Utilizing relative weight analysis in customer satisfaction research. *International Journal of Market Research*, *62*(2), 158–175.

86

Genizi, A. (1993). Decomposition of R^2 in multiple regression with correlated regressors. *Statistica Sinica*, *3*(2), 407–420.

86, 94

Green, P. E., Carroll, J. D., & Desarbo, W. S. (1978). A new measure of predictor variable importance in multiple regression. *Journal of Marketing Research*, *15*(3), 356–360.

86

Groll, A., Hambuckers, J., Kneib, T., & Umlauf, N. (2019). LASSO-type penalization in the framework of generalized additive models for location, scale and shape. *Computational Statistics & Data Analysis*, *140*(1), 59–73.

12

Grömping, U. (2015). Variable importance in regression models. *Wiley Interdisciplinary Reviews: Computational Statistics*, *7*(2), 137–152.

86, 100

Gumbel, E. J. (1935). Les valeurs extrêmes des distributions statistiques. *Annales de l'institut Henri Poincaré*, *5*(2), 115–158.

79

Gunawardene, N. (2003). *Humanity will survive information deluge – Sir Arthur C Clarke.* Retrieved from https://bazaarmodel.net/phorum/read.php?1,462

1

Hastie, T. J., & Tibshirani, R. J. (1990). *Generalized additive models.* London: Taylor & Francis.

2, 7, 26, 59, 87

Heller, G. Z., Couturier, D.-L., & Heritier, S. R. (2019). Beyond mean modelling: Bias due to misspecification of dispersion in Poisson-inverse Gaussian regression. *Biometrical Journal*, *61*(2), 333–342.

60, 76, 78

Hofner, B., Mayr, A., & Schmid, M. (2016). gamboostLSS: An R Package for Model Building and Variable Selection in the GAMLSS Framework. *Journal of Statistical Software, Articles*, *74*(1), 1–31.

104, 105, 106

Hook, D. W., Porter, S. J., & Herzog, C. (2018). Dimensions: Building context for search and evaluation. *Frontiers in Research Metrics and Analytics*, *3*(23), 1–11.

xxiii, 1, 2

Huzurbazar, V. S. (1950). Probability Distributions and Orthogonal Parameters. *Mathematical Proceedings of the Cambridge Philosophical Society*, *46*(2), 281–284.

60, 61, 63, 65, 66, 68, 69, 70, 71, 83

International Institute for Population Sciences. (2007). *India national family health survey (nfhs-3) 2005-06.* IIPS and Macro International.

104

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). Islr: Data for 'introduction to statistical learning with applications in R' [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=ISLR (R package version 1.2)

5, 16

Jeffreys, H. (1948). *Theory of probability* (2nd ed.). Oxford: Clarendon Press.

60

Johnson, J. (2000). A heuristic method for estimating the relative weight of predictor variables in multiple regression. *Multivariate Behavioral Research*, *35*(1), 1-19.

86, 94, 95, 96, 100

Johnson, N. L., Kotz, S., & Balakrishnan, N. (1995). *Continuous univariate distributions* (2nd ed.). New York: Wiley.

72

Kang, D., Ko, K., & Huh, J. (2015). Determination of extreme wind values using the Gumbel distribution. *Energy*, *86*(1), 51–58.

79

Klein, N., Carlan, M., Kneib, T., Lang, S., & Wagner, H. (2021). Bayesian effect selection in structured additive distributional regression models. *Bayesian Analysis*, *16*(2), 545–573.

12

Klein, N., & Kneib, T. (2016a). Scale-dependent priors for variance parameters in structured additive distributional regression. *Bayesian Analysis*, *11*(4), 1071–1106.

8

Klein, N., & Kneib, T. (2016b). Simultaneous inference in structured additive conditional copula regression models: A unifying Bayesian approach. *Statistics and Computing*, *26*(4), 841–860.

26

Klein, N., Kneib, T., Klasen, S., & Lang, S. (2015). Bayesian structured additive distributional regression for multivariate responses. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *64*(4), 569–591.

12, 26, 61

Klein, N., Kneib, T., & Lang, S. (2015). Bayesian generalized additive models for location, scale and shape for zero-inflated and overdispersed count data. *Journal of the American Statistical Association*, *110*(509), 405–419.

27

Klein, N., Kneib, T., Lang, S., & Sohn, A. (2015). Bayesian structured additive distributional regression with an application to regional income inequality in Germany. *Annals of Applied Statistics*, *9*(2), 1024–1052.

2, 7, 8, 13, 25, 26, 62, 88

Klein, N., Nott, D. J., & Smith, M. S. (2021). Marginally calibrated deep distributional regression. *Journal of Computational and Graphical Statistics*, *30*(2), 467–483.

12

Leal, M., Boavida-Portugal, I., Fragoso, M., & Ramos, C. (2019). How much does an extreme rainfall event cost? Material damage and relationships between insurance, rainfall, land cover and urban flooding. *Hydrological Sciences Journal*, *64*(6), 673–689.

79

Leeper, T. J. (2019). prediction: Tidy, type-safe 'prediction()' methods [Computer software manual]. (R package version 0.3.14)

15

Lenth, R. (2019). emmeans: Estimated marginal means, aka least-squares means [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=emmeans (R package version 1.4)

15

Lenth, R. V. (2016). Least-squares means: The R package lsmeans. *Journal of Statistical Software*, *69*(1), 1–33.

15

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

23, 39, 41

Lin, X., & Zhang, D. (1999). Inference in generalized additive mixed models by using smoothing splines. *Journal of the Royal Statistical Society: Series B (Methodological)*, *61*(2), 381–400.

8, 26, 62, 88

Lindeman, R., Merenda, P., & Gold, R. (1980). *Introduction to bivariate and multivariate analysis*. Glenview, Illinois: Scott, Foresman.
86

Lüdecke, D. (2018). ggeffects: Tidy data frames of marginal effects from regression models. *Journal of Open Source Software*, *3*(26), 772.
15

MacNally, R. (1996). Hierarchical partitioning as an interpretative tool in multivariate inference. *Australian Journal of Ecology*, *21*(2), 224–228.
86

Marra, G., & Radice, R. (2017). Bivariate copula additive models for location, scale and shape. *Computational Statistics and Data Analysis*, *112*(1), 99–113.
11, 26, 57

McCullagh, P. (1980). Regression models for ordinal data. *Journal of the Royal Statistical Society: Series B (Methodological)*, *42*(2), 109–142.
89

McCullagh, P., & Nelder, J. A. (1983). *Generalized linear models*. Boca Raton, Florida: Springer US.
125

Millar, R. B. (2011). *Maximum likelihood estimation and inference: With examples in R, SAS and ADMB* (Vol. 111). John Wiley & Sons.
62

Nelder, J. A. (1975). Announcement by the working party on statistical computing: GLIM (generalized linear interactive modelling program). *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *24*(2), 259–261.
7

Nelder, J. A., & Wedderburn, R. W. M. (1972). Generalized linear models. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, *135*(3), 370–384.
2, 6, 26, 62, 85, 88, 112

Neuwirth, E. (2014). `RColorBrewer`: Colorbrewer palettes [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=RColorBrewer`

(R package version 1.1-2)

34

Olver, P. (2013). *Introduction to partial differential equations.* Cham, Switzerland: Springer International Publishing.

72

Owen, J. (2016). rhandsontable: Interface to the 'handsontable.js' library [Computer software manual]. (R package version 0.3.4)

47

R Core Team. (2020). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria.

9, 87

Rajendra, B. D., & Machhindra, S. V. (2018). *Study of hostel life* (Master's thesis). Shivaji University, Kolhapur, Kolhapur, India.

xxviii, 89, 126

Rigby, R. A., & Stasinopoulos, M. D. (1996). Mean and dispersion additive models. In W. Härdle & M. G. Schimek (Eds.), *Statistical theory and computational aspects of smoothing* (pp. 215–230). Berlin: Physica-Verlag HD.

7

Rigby, R. A., & Stasinopoulos, M. D. (2005). Generalized Additive Models for Location, Scale and Shape. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *54*(3), 507–554.

2, 7, 9, 13, 26, 59, 85, 87, 106, 112

Rigby, R. A., Stasinopoulos, M. D., Heller, G. Z., & De Bastiani, F. (2019). *Distributions for modeling location, scale, and shape: Using GAMLSS in R.* Boca Raton, Florida: Chapman and Hall/CRC.

71

Rügamer, D., Shen, R., Bukas, C., Barros de Andrade e Sousa, L., Thalmeier, D., Klein, N., ... Müller, C. L. (2021). *deepregression: a flexible neural network framework for semi-structured deep distributional regression.*

12

Schapire, R. E. (1990). The strength of weak learnability. *Mach. Learn.*, *5*(2), 197—227.

86

Schlosser, L., Hothorn, T., Stauffer, R., & Zeileis, A. (2019). Distributional regression forests for probabilistic precipitation forecasting in complex terrain. *Ann. Appl. Stat.*, *13*(3), 1564–1589.

12, 26

Senn, S. (2003). *Dicing with death: Chance, risk and health.* Cambridge: Cambridge University Press.

1

Smyth, G. K. (1989). Generalized linear models with varying dispersion. *Journal of the Royal Statistical Society: Series B (Methodological)*, *51*(1), 47–60.

7

Soetaert, K., Petzoldt, T., & Setzer, R. W. (2010). Solving differential equations in R: Package deSolve. *Journal of Statistical Software*, *33*(9), 1–25.

72

Stadlmann, S., & Kneib, T. (2021). Interactively visualizing distributional regression models with distreg.vis. *Statistical Modelling*. doi: https://doi.org/10.1177/1471082X211007308

3

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

vi, 9, 14, 26, 28, 88, 101

Stasinopoulos, M. D., & Rigby, R. A. (2018). gamlss.tr: Generating and fitting truncated 'gamlss.family' distributions [Computer software manual]. (R package version 5.1-0)

27

Stasinopoulos, M. D., & Rigby, R. A. (2019). gamlss.dist: Distributions for generalized additive models for location scale and shape [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=gamlss.dist` (R package

version 5.1-4)

10, 27

Stasinopoulos, M. D., Rigby, R. A., Heller, G. Z., Voudouris, V., & De Bastiani, F. (2017). *Flexible regression and smoothing: Using GAMLSS in R*. Boca Raton, Florida: CRC Press.

9, 10, 27, 63, 122

Stein, G. Z., Zucchini, W., & Juritz, J. M. (1987). Parameter estimation for the Sichel distribution and its multivariate extension. *Journal of the American Statistical Association*, *82*(399), 938–944.

60

Thomas, J., Mayr, A., Bischl, B., Schmid, M., Smith, A., & Hofner, B. (2018). Gradient boosting for distributional regression - faster tuning and improved variable selection via noncyclical updates. *Statistics and Computing*, *28*(3), 673–687.

11, 12, 88, 105

Tishbirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288.

12, 86

Tonidandel, S., & LeBreton, J. (2010). Determining the relative importance of predictors in logistic regression: An extension of relative weight analysis. *Organizational Research Methods*, *13*(4), 767–781.

87, 96, 97, 100

Umlauf, N., Adler, D., Kneib, T., Lang, S., & Zeileis, A. (2015). Structured additive regression models: An R interface to BayesX. *Journal of Statistical Software*, *63*(21), 1–46.

11

Umlauf, N., Klein, N., & Zeileis, A. (2018). BAMLSS: Bayesian additive models for location, scale, and shape (and beyond). *Journal of Computational and Graphical Statistics*, *27*(3), 612–627.

vi, 10, 14, 26, 27, 28, 62, 88

Voncken, L., Kneib, T., Albers, C. J., Umlauf, N., & Timmerman, M. E. (2021).

Bayesian Gaussian distributional regression models for more efficient norm estimation. *British Journal of Mathematical and Statistical Psychology*, *74*(1), 99–117.

8

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. New York: Springer-Verlag. Retrieved from http://ggplot2.org

34, 36, 46, 55

Wood, S. N. (2006). Low-rank scale-invariant tensor product smooths for generalized additive mixed models. *Biometrics*, *62*(4), 1025–1036.

74

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

11, 74, 90, 101

Yee, T. W. (2010). The VGAM package for categorical data analysis. *Journal of Statistical Software*, *32*(10), 1–34.

11

Yee, T. W. (2015). *Vector generalized linear and additive models: With an implementation in R*. New York: Springer.

11, 57

Yee, T. W., & Hastie, T. J. (2003). Reduced-rank vector generalized linear models. *Statistical Modelling*, *3*(1), 15–41.

11

Zimmerman, D. L. (2020). *Linear model theory: With examples and exercises*. Cham, Switzerland: Springer International Publishing.

3, 60