

Bayesian Semiparametric Univariate and Multivariate Density Deconvolution for Continuous and Zero-Inflated Data with the R Package BayesDecon

BLAKE MOYA¹, MAINAK MANNA¹, AND ABHRA SARKAR^{1,*}

¹*Department of Statistics and Data Sciences, The University of Texas at Austin, Austin, Texas, USA*

Abstract

Density deconvolution is a fundamental problem in measurement error settings, arising when the goal is to recover the distribution of unobserved latent variables based on their error-contaminated proxies. An important application arises in nutritional epidemiology, where the objective is to estimate the long-term average intake of nutritional components using 24-hour dietary recall data. This task involves complex multivariate models that account for conditionally heteroscedastic measurement errors, zero-inflated proxies for episodically consumed dietary components, diverse marginal distribution shapes across components, etc. While flexible Bayesian hierarchical methods, coupled with Markov chain Monte Carlo techniques, have been increasingly successful in recent years to enable deconvolution under such intricate but realistic scenarios, the development of rapid and user-friendly software implementations has lagged behind. The R package `BayesDecon` has been developed to address this challenge, providing a fast and accessible implementation of flexible Bayesian deconvolution methods for practitioners working with measurement errors. In the process, we have introduced substantial improvements to several aspects of the original models and algorithms, and have also added options to fit simpler parametric models. Although illustrated on challenges in nutritional epidemiology, the package addresses general deconvolution problems and hence is broadly applicable to other domains as well.

Keywords *Measurement error; NHANES data*

1 Introduction

In many real-world problems, the goal is to estimate the distribution of a latent variable X that is not directly observable, but instead inferred through a related proxy variable Y , which serves as a noisy or indirect proxy for X . If the observed variable Y is contaminated by measurement error U , such that its distribution arises from the convolution of the densities of X and U , then estimating the density of X from observed values of Y becomes a problem of density deconvolution.

Motivating examples arise in nutritional epidemiology (see, e.g., Carroll et al., 2006; Keogh et al., 2020; Shaw et al., 2020), where one seeks to estimate the proportion of individuals in a population whose long-term average intake of nutrients such as sodium or vitamins meets recommended guidelines. Since long-term intake is not directly observable, researchers rely on short-term dietary assessment, most commonly 24-hour dietary recalls, as proxies. These recalls

*Corresponding author. Email: abhra.sarkar@utexas.edu.

provide imperfect and noisy measurements of the underlying long-term intake, necessitating appropriate deconvolution methods to recover the latent distribution of interest.

Early work on deconvolution typically assumed X and U are independent and/or the distribution of U is known, often taken to be Gaussian; see, for example, Chapter 12 in Carroll et al. (2006). However, the independence assumption is often unrealistic. In many real-world settings, including especially in dietary recalls, the variability of the measurement errors U may depend on the underlying value of X . Bayesian methods and associated Markov chain Monte Carlo (MCMC) machinery have proven useful tools in accommodating these scenarios (Shaw et al., 2020) and are the main focus of the article. While these approaches necessitate a comprehensive specification of the likelihood – including elements which are often of secondary interest, such as the error distribution – and hence can be both methodologically and computationally demanding, recent modeling advances enable highly flexible and realistic treatment of these components. In return, such methods also produce estimates for such elements, not just the main density of interest, thereby offering a thorough understanding of the problem and facilitating informal assessments of deviations from parametric assumptions. An important first work in this direction is Staudenmayer et al. (2008), who proposed a method that allows the variance of a normally distributed error to vary as a function of X , modeling both the distribution of X and the variance function of U using mixtures of B-splines. Nonetheless, the assumption of normality may still be inappropriate for many settings. Sarkar et al. (2014) relaxed this assumption by modeling the error distribution as a structured mixture of normals that enforces zero mean while allowing for substantial flexibility in shape and tail behavior.

Extending these methods to multivariate settings, with vector-valued $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{U}$, presents significant additional challenges. Sarkar et al. (2018) addressed the multivariate case by modeling both the latent and error distributions using mixtures of multivariate normals with careful control of the number of parameters, achieved through low-rank approximations of the covariance matrices. However, this approach could not handle zero-inflated proxies and was also limited in accurately estimating marginal distributions, particularly when they exhibit steep gradients that are not well captured by multivariate normal mixtures. To overcome these limitations, Sarkar et al. (2021) developed a latent variable framework to explicitly account for zero-inflated recalls, and introduced a copula-based modeling strategy that separates the modeling of marginal distributions from the dependence structure. Specifically, the marginals are modeled flexibly, tailored to the distributional characteristics of individual components of $\mathbf{X}$, while a Gaussian copula captures the joint dependence.

This formulation allows for more accurate and adaptable modeling in settings with heterogeneous and structurally complex dietary data. Consider, for example, the nutritional epidemiology application discussed above. Some dietary components, such as sodium and total grains, are consumed almost daily, resulting in continuous, strictly positive surrogate measurements. In contrast, other components like milk and whole grains are consumed episodically, producing surrogate variables with a substantial proportion of zeros, posing significant challenges for density estimation near zero. To address these complexities, we adopt the approach of Sarkar et al. (2021), which first models the marginal distributions of the strictly continuous dietary components using flexible mixtures of truncated normals and the marginal distributions of the zero-inflated dietary components using flexible normalized mixtures of B-splines (see Section 2), and then constructs the joint distribution through a copula-based framework that captures the dependence structure between the different dietary components.

The literature on density deconvolution in complex multivariate settings, particularly those involving zero-inflation, remains limited, and to our knowledge, no dedicated R package cur-

rently addresses such scenarios. We identified a few R packages—for example, `spedecon` (Kent and Ruppert, 2023), `pmledecon` (Cai et al., 2022), `deconvolve` (Delaigle et al., 2026) and `deconvolveR` (Narasimhan and Efron, 2020)—that address deconvolution problems in much simpler, univariate, and strictly continuous settings, each with several notable limitations. The package `spedecon` assumes a known, identically distributed U that is independent of X (Kent and Ruppert, 2023). The package `pmledecon` uses a similar framework, attempting to relax the assumption of known error by requiring a pure error sample for estimation, a condition often infeasible in real data applications. In contrast, the package `deconvolve` allows for an unknown nonparametric error distribution with heteroscedasticity, but it still assumes no dependence between X and U (Delaigle and Meister, 2008). The package `deconvolveR` does not assume a linear relationship of the form $Y = X + U$; instead, it models $Y | X$ as coming from an exponential family and assumes that X takes values in a finite set (Efron, 2016), making it unsuitable for application in the settings discussed above. The methodological limitations of these packages do not allow them to be readily extended to complex multivariate settings described above.

While R codes were originally made available alongside the publication of Sarkar et al. (2014, 2021), broader adoption of their approaches in applied settings may have been hindered by several factors, including their limited accessibility to non-experts in Bayesian methods, long runtimes, and lack of detailed accompanying documentation. To fill these gaps, the *BayesDecon* package provides fast, streamlined, user-friendly methods for the univariate and multivariate density deconvolution frameworks developed in Sarkar et al. (2014, 2021), including support for settings where the data may exhibit zero inflation. The package is already available on the Comprehensive R Archive Network (CRAN). It uses C++ to perform an efficient MCMC-based implementation under the hood, offering significantly faster performance compared to the implementation relying solely on R.

Additionally, our new implementation includes a few modeling enhancements such as enforcing monotonicity in B-spline basis coefficients, when appropriate, with related MCMC sampling carried out using the method of Geweke (1991). Furthermore, we implement an improved algorithm to update the parameters of truncated normal distributions using Robert (1995). We have fixed minor bugs in the original implementations in the process, and also introduced additional options for fitting various simpler parametric models, such as homoscedastic errors (i.e., measurement errors with constant variance not dependent on the underlying X values) and Gaussian or Laplace distributed errors.

For benchmarking, we used synthetic data comprising 3 variables with 3 replicates per individual for 1000 individuals. The original implementation of these deconvolution methods already demonstrated excellent recovery of target densities in realistic simulation settings. In replicated simulation experiments, the new implementation provided further improvements for variables where the original approach had slightly underperformed, while maintaining similar performance for the others. On an Apple M3 Max with 36GB of unified memory, while the original R codes required approximately 33 minutes to complete 5000 MCMC iterations, *BayesDecon* completed the same task in just under 3 minutes, demonstrating a substantial gain in efficiency.

We demonstrate the package’s capability to estimate long-term averages of various dietary components by deconvolving their measurement error-contaminated 24-hour recalls from the open-access National Health and Nutrition Examination Survey (NHANES) data set, distinct from prior work using the Eating at America’s Table Study (EATS) in Sarkar et al. (2014, 2021).

The article is organized as follows. For the sake of completeness, Section 2 provides a compact description of the models implemented in our package, along with the improvements incorporated. Section 3 provides a brief outline of the posterior sampler. Section 4 introduces

different objects and functions available in the package. Finally, Section 5 provides an overall discussion of the package and outlines potential future directions. Supplementary materials present additional numerical experiments in simulation and real-data settings to further validate the efficiency and reliability of the new package.

2 Model Description

To maintain clarity in a notation-heavy manuscript, throughout we denote both random variables (or vectors or parameters) and their specific realizations by the same symbols (e.g., X , $\mathbf{X}$, $\boldsymbol{\theta}$, etc.). While this represents a slight abuse of notation, the intended meaning should remain clear from the context. Throughout this paper, when a density is nonstandard, we specify its functional form either explicitly or up to a normalizing constant as $f(\boldsymbol{\theta} \mid \boldsymbol{\xi}) \propto p(\boldsymbol{\theta} \mid \boldsymbol{\xi})$, where $\boldsymbol{\xi}$ denotes its characterizing parameters. When we have a standard and/or closed-form distribution instead, we use the tilde notation, $(\boldsymbol{\theta} \mid \boldsymbol{\xi}) \sim f(\boldsymbol{\xi})$, where f denotes the distribution family. In such cases, to streamline notation, we use the same symbol f to represent both the distribution family name and its corresponding probability density function, evaluated as $f(\boldsymbol{\theta} \mid \boldsymbol{\xi})$. For mixture densities, we leave the parameters $\boldsymbol{\xi}$ initially implicit before they are formally introduced later in the text. More specifically, $\text{Normal}(\mu, \sigma^2)$ denotes a Normal distribution with mean μ and standard deviation σ ; $\text{Dir}(\alpha_1, \dots, \alpha_K)$ denotes a Dirichlet distribution on a K -dimensional simplex with shape parameters $\alpha_1, \dots, \alpha_K$; $\text{Uniform}(a, b)$ denotes a uniform distribution supported on $[a, b]$; $\text{Inv-Ga}(a, b)$ denotes an inverse gamma distribution with shape parameter a and scale parameter b , having probability density function $f(s) = b^a s^{-a-1} \exp(-b/s) / \Gamma(a)$, $s > 0$; $\text{MVN}_d(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a d -dimensional multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$; $\text{TN}(\mu, \sigma^2, [a, b])$ denotes a truncated normal distribution with location parameter μ and scale parameter σ , restricted to the interval $[a, b]$; and $\text{TMVN}_d(\boldsymbol{\mu}, \boldsymbol{\Sigma}, [\mathbf{l}, \mathbf{u}])$ denotes a d -dimensional truncated multivariate normal distribution with location vector $\boldsymbol{\mu}$, scale matrix $\boldsymbol{\Sigma}$, and support restricted to the hyper-rectangle defined by the lower and upper bounds $\mathbf{l}$ and $\mathbf{u}$, respectively.

2.1 Latent Variable Framework

We assume that there are q zero-inflated components along with p strictly positive components of interest. There are n individuals and m_i replicates of the surrogate are available for the i -th individual. Let $\mathbf{Y}_{i,j} = (Y_{1,i,j}, \dots, Y_{2q+p,i,j})^\top$ represent the observed data for the j -th replicate of the i -th individual. For $l = 1, \dots, q$, $Y_{l,i,j}$ is a binary variable which takes the value 1 if a positive value is observed corresponding to the l -th zero-inflated component and zero otherwise. For $l = q + 1, \dots, 2q$, $Y_{l,i,j}$ is the observed value of the l -th zero-inflated component. For $l = 2q + 1, \dots, 2q + p$, $Y_{l,i,j}$ is the observed value of the l -th regular component.

Also, let $\mathbf{X}_i = (X_{1,i}, \dots, X_{q+p,i})^\top$, where $X_{l,i}$ is the true latent value of the l -th component of the i -th individual. Next, some additional semi-latent variables $\mathbf{W}_{i,j} = (W_{1,i,j}, \dots, W_{2q+p,i,j})^\top$ are defined, all continuous and connected with the $Y_{l,i,j}$'s by

$$\begin{aligned} Y_{l,i,j} &= 1(W_{l,i,j} > 0), & \text{for } l = 1, \dots, q, \\ Y_{l,i,j} &= Y_{l-q,i,j} W_{l,i,j}, & \text{for } l = q + 1, \dots, 2q, \\ Y_{l,i,j} &= W_{l,i,j}, & \text{for } l = 2q + 1, \dots, 2q + p. \end{aligned}$$

For $l = 1, \dots, q$, $W_{l,i,j}$ determines whether the observed value of the l -th zero-inflated component for the j -th replicate of the i -th individual is positive. It is latent up to its sign and

can be thought of as a contaminated version of $\tilde{X}_{l,i} = h_l(X_{l,i})$, a function of $X_{l,i}$, and is modeled as

$$W_{l,i,j} = \tilde{X}_{l,i} + U_{l,i,j} = h_l(X_{l,i}) + U_{l,i,j}, \quad l = 1, \dots, q,$$

so that the probability of observing a positive value for the l -th zero-inflated component of the i -th individual is obtained as $P_l(X_{l,i}) = \mathbb{P}(W_{l,i,j} > 0 \mid X_{l,i}) = \mathbb{P}\{U_{l,i,j} > -h_l(X_{l,i})\}$.

For $l = q + 1, \dots, 2q$, when $Y_{l,i,j}$ is positive, $W_{l,i,j}$ equals the observed value $Y_{l,i,j}$ of the $(l - q)$ -th zero-inflated component in the j -th replicate of the i -th individual. However, when the observed value $Y_{l,i,j}$ is zero, $W_{l,i,j}$ is latent. Assuming zero mean for $U_{l,i,j}$ and independence of $Y_{l-q,i,j}$ and $W_{l,i,j}$, these semi-latent $W_{l,i,j}$'s can then be established as contaminated proxies for $\tilde{X}_{l-q,i} = X_{l-q,i}/P_{l-q}(X_{l-q,i})$ (see Section 2.1 of Sarkar et al., 2021, for more explanation), and are therefore modeled as

$$W_{l,i,j} = \tilde{X}_{l,i} + U_{l,i,j} = X_{l-q,i}/P_{l-q}(X_{l-q,i}) + U_{l,i,j}, \quad l = q + 1, \dots, 2q.$$

Finally, for $l = 2q + 1, \dots, 2q + p$, $W_{l,i,j}$ equals the observed value $Y_{l,i,j}$ of the $(l - q)$ -th regular component, the contaminated proxy for $\tilde{X}_{l,i} = X_{l-q,i}$, and is modeled as

$$W_{l,i,j} = \tilde{X}_{l,i} + U_{l,i,j} = X_{l-q,i} + U_{l,i,j}, \quad l = 2q + 1, \dots, 2q + p.$$

2.2 Modeling the Density of $\mathbf{X}$

The density $f_{\mathbf{X}}$ is specified using a Gaussian copula as

$$f_{\mathbf{X}}(\mathbf{X}) = |\mathbf{R}_{\mathbf{X}}|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} \mathbf{Y}_{\mathbf{X}}^{\top} (\mathbf{R}_{\mathbf{X}}^{-1} - \mathbf{I}_{q+p}) \mathbf{Y}_{\mathbf{X}} \right\} \prod_{l=1}^{q+p} f_{X,l}(X_l),$$

with $F_{X,l}(X_l) = \Phi(Y_{X,l})$ for all l and $\mathbf{R}_{\mathbf{X}}$ is the correlation matrix of $\mathbf{X}$. Notably, when the components of $\mathbf{X}$ are independent, $\mathbf{R}_{\mathbf{X}} = \mathbf{I}_{q+p}$ and the joint density factors into the product of its marginal densities.

For $l = 1, \dots, q$, the marginals of X_l need to capture the local variations in shape. To be precise, most of the W_l 's will be latent when X_l is near zero and hence we pull information from the observed W_l 's to get the smoothness of the density near zero. Sarkar et al. (2021) showed that quadratic B-spline bases can do the job efficiently. Let $[A_l, B_l]$ be the support of interest for X_l . We divide the interval $[A_l, B_l]$ into L subintervals using the knot points $A_l = t_{l,1} = t_{l,2} = t_{l,3} < t_{l,4} < \dots < t_{l,L+3} = t_{l,L+4} = t_{l,L+5} = B_l$. Using this, we define $K_t = L + 2$ quadratic B-spline bases $\mathbf{B}_{2,l,K_t} = \{b_{2,l,1}, \dots, b_{2,l,K_t}\}$ (de Boor, 2001). Now we specify the density as

$$f_{X,l}(X_l \mid \boldsymbol{\xi}_l) \propto \mathbf{B}_{2,l,K_t} \exp(\boldsymbol{\xi}_l), \quad f(\boldsymbol{\xi}_l \mid \sigma_{\boldsymbol{\xi},l}^2) \propto \exp\{-\boldsymbol{\xi}_l^{\top} \boldsymbol{\Lambda}_l \boldsymbol{\xi}_l / (2\sigma_{\boldsymbol{\xi},l}^2)\}, \quad \sigma_{\boldsymbol{\xi},l}^2 \sim \text{Inv-Ga}(a_{\boldsymbol{\xi}}, b_{\boldsymbol{\xi}}),$$

where $\boldsymbol{\xi}_l = \{\xi_{l,1}, \dots, \xi_{l,K_t}\}$; $\exp(\boldsymbol{\xi}_l) = \{\exp(\xi_{l,1}), \dots, \exp(\xi_{l,K_t})\}$; and $\boldsymbol{\Lambda}_l = \mathbf{D}_2^{\top} \mathbf{D}_2$ where $\mathbf{D}_2$ is a $(K_t + 2) \times K_t$ matrix such that $\mathbf{D}_2 \boldsymbol{\xi}_l$ computes the second-order difference in $\boldsymbol{\xi}_l$ (Eilers and Marx, 1996).

For $l = q + 1, \dots, q + p$, marginals of X_l are specified by mixtures of truncated normals as

$$f_{X,l}(X_l \mid \{\pi_{X,l,k}, \mu_{X,k}, \sigma_{X,k}^2\}_{k=1}^{K_X}) = \sum_{k=1}^{K_X} \pi_{X,l,k} \text{TN}(X_l \mid \mu_{X,k}, \sigma_{X,k}^2, [A_l, B_l]),$$

$$\boldsymbol{\pi}_{X,l} \sim \text{Dir}(\boldsymbol{\alpha}_{X_l}/K_X, \dots, \boldsymbol{\alpha}_{X_l}/K_X), \quad \mu_{X,k} \sim \text{Normal}(\mu_{X,0}, \sigma_{X,0}^2), \quad \sigma_{X,k}^2 \sim \text{Inv-Ga}(a_{\sigma_{X,0}^2}, b_{\sigma_{X,0}^2}).$$

To specify $\mathbf{R}_X$, we use Cholesky decomposition to write $\mathbf{R} = \mathbf{V}\mathbf{V}^\top$, where $\mathbf{V}$ is a $(q + p) \times (q + p)$ matrix of the form

$$\mathbf{V} = \begin{pmatrix} v_{1,1} & 0 & \cdots & 0 \\ v_{2,1} & v_{2,2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ v_{q+p,1} & v_{q+p,2} & \cdots & v_{q+p,q+p} \end{pmatrix}.$$

The various elements of $\mathbf{V}$ are specified using spherical coordinates as follows, ensuring that $\mathbf{R}$ is a valid correlation matrix.

$$\begin{aligned} v_{1,1} &= 1, \\ v_{2,1} &= b_1, \quad v_{2,2} = \sqrt{1 - b_1^2}, \\ v_{3,1} &= b_2 \sin \theta_1, \quad v_{3,2} = b_2 \cos \theta_1, \quad v_{3,3} = \sqrt{1 - b_2^2}. \end{aligned}$$

More generally, with $i(l) = 1 + \{1 + 2 + \cdots + (l - 3)\} = (l^2 - 5l + 8)/2$, the l -th row of $\mathbf{V}$ is

$$\begin{aligned} v_{l,1} &= b_{l-1} \sin \theta_{i(l)}, \\ v_{l,k} &= b_{l-1} \cos \theta_{i(l)} \cos \theta_{i(l)+1} \cdots \cos \theta_{i(l)+k-2} \sin \theta_{i(l)+k-1}, \quad \text{for } k = 2, \dots, (l - 2), \\ v_{l,l-1} &= b_{l-1} \cos \theta_{i(l)} \cos \theta_{i(l)+1} \cdots \cos \theta_{i(l)+l-4} \cos \theta_{i(l)+l-3}, \quad \text{and} \\ v_{l,l} &= \sqrt{1 - b_{l-1}^2}. \end{aligned}$$

We impose uniform priors on b_l and θ_s as

$$\begin{aligned} b_l &\stackrel{\text{iid}}{\sim} \text{Uniform}(-1, 1) \text{ for } l = 1, \dots, (q + p - 1), \\ \theta_s &\stackrel{\text{iid}}{\sim} \text{Uniform}(-\pi, \pi) \text{ for } s = 1, \dots, i(q + p) + (q + p - 3). \end{aligned}$$

2.3 Modeling the Density of $\mathbf{U} \mid \mathbf{X}$

To flexibly model the conditional distribution $\mathbf{U}_{i,j} = (U_{1,i,j}, \dots, U_{2q+p,i,j})^\top$ given $\mathbf{X}$, a multivariate structure $\mathbf{U}_{ij} = \mathbf{S}(\mathbf{X}_i)\boldsymbol{\epsilon}_{i,j}$ is assumed, where $\boldsymbol{\epsilon}_{i,j} = (\epsilon_{1,i,j}, \dots, \epsilon_{2q+p,i,j})^\top$ is independent of $\mathbf{X}_i$ with $\mathbb{E}(\boldsymbol{\epsilon}_{i,j}) = \mathbf{0}$ and $\mathbf{S}(\mathbf{X}_i)$ is a matrix that captures the conditional covariance patterns as $\text{cov}(\mathbf{U}_{i,j} \mid \mathbf{X}_i) = \mathbf{S}(\mathbf{X}_i)\text{cov}(\boldsymbol{\epsilon}_{i,j})\mathbf{S}(\mathbf{X}_i)^\top$.

The density of $\boldsymbol{\epsilon}$ is specified using a Gaussian copula, similar to Section 2.2, as

$$f_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}) = \prod_{l=1}^q f_{\epsilon,l}(\epsilon_l) \times |\mathbf{R}_{\boldsymbol{\epsilon}}|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} \mathbf{Y}_X^\top (\mathbf{R}_{\boldsymbol{\epsilon}}^{-1} - \mathbf{I}_{q+p}) \mathbf{Y}_X \right\} \prod_{l=q+1}^{2q+p} f_{\epsilon,l}(\epsilon_l),$$

where $F_{\epsilon,l}(\epsilon_l) = \Phi(Y_{\epsilon,l})$ for all l .

For $l = 1, \dots, q$, marginal densities $f_{\epsilon,l}$ are thus modeled as Normal(0, 1).

For $l = q + 1, \dots, 2q + p$, the marginals $f_{\epsilon,l}$ are modeled as a flexible mixture where each

mixture component itself is a stylized mixture of two normals. Specifically,

$$f_{\epsilon,l}(\epsilon_l \mid \{\pi_{\epsilon,l,k}, p_{\epsilon,k}, \tilde{\mu}_{\epsilon,k}, \sigma_{\epsilon,k,1}^2, \sigma_{\epsilon,k,2}^2\}_{k=1}^{K_\epsilon}) = \sum_{k=1}^{K_\epsilon} \pi_{\epsilon,l,k} f_{c\epsilon}(\epsilon_l \mid p_{\epsilon,k}, \tilde{\mu}_{\epsilon,k}, \sigma_{\epsilon,k,1}^2, \sigma_{\epsilon,k,2}^2),$$

$$\pi_{\epsilon,l,k} \sim \text{Dirichlet}\left(\frac{\alpha_{\epsilon,l}}{K_\epsilon}, \dots, \frac{\alpha_{\epsilon,l}}{K_\epsilon}\right),$$

$$p_{\epsilon,k} \sim \text{Uniform}(0, 1), \quad \tilde{\mu}_{\epsilon,k} \sim \text{Normal}(0, \sigma_{\epsilon,\tilde{\mu}}^2), \quad \sigma_{\epsilon,k,1}^2 \sim \text{Inv-Ga}(a_\epsilon, b_\epsilon), \quad \sigma_{\epsilon,k,2}^2 \sim \text{Inv-Ga}(a_\epsilon, b_\epsilon),$$

where $p_{\epsilon,k}$, $\tilde{\mu}_{\epsilon,k}$, $\sigma_{\epsilon,k,1}^2$, and $\sigma_{\epsilon,k,2}^2$ are assumed to be mutually independent for each $k = 1, \dots, K_\epsilon$, and

$$f_{c\epsilon}(\epsilon \mid p, \tilde{\mu}, \sigma_1^2, \sigma_2^2) = p \text{Normal}(\epsilon \mid \mu_1, \sigma_1^2) + (1 - p) \text{Normal}(\epsilon \mid \mu_2, \sigma_2^2),$$

with $\mu_1 = c_1 \tilde{\mu}$, $\mu_2 = c_2 \tilde{\mu}$, $c_1 = (1 - p)/\{p^2 + (1 - p)^2\}^{1/2}$, and $c_2 = -p/\{p^2 + (1 - p)^2\}^{1/2}$.

This specification ensures $\mathbb{E}(\epsilon_{l,i,j}) = 0$, hence also $\mathbb{E}(\mathbf{U}_{i,j} \mid \mathbf{X}_i) = 0$, while allowing the shape and tail to have wide-ranging behavior, including skewness, multimodality, and heavy tails.

The correlation matrix $\mathbf{R}_\epsilon$ is modeled analogously to $\mathbf{R}_X$ with $\mathbf{R}_\epsilon = \mathbf{V}_\epsilon \mathbf{V}_\epsilon^\top$ and the elements of $\mathbf{V}_\epsilon$ parameterized using spherical coordinates as before.

It is often natural to assume that X_l has no direct significant influence on $U_{l'}$ for $l \neq l'$, and hence $\mathbf{S}(\mathbf{X}_i)$ may be considered to be a diagonal matrix (see Sarkar et al., 2018, for a detailed discussion). That is,

$$\mathbf{S}(\mathbf{X}_i) = \text{diag}\{s_1(X_{1,i}), \dots, s_q(X_{q,i}), s_{q+1}(X_{1,i}), \dots, s_{2q}(X_{q,i}), s_{2q+1}(X_{q+1,i}), s_{2q+p}(X_{q+p,i})\}.$$

Furthermore, it is assumed that $s_l(X_{l,i}) = 1$ for $l = 1, \dots, q$. Also, for $l = q + 1, \dots, 2q$, $s_l(X_{l-q,i})$ is modeled as a function of $X_{l-q,i}/P_{l-q}(X_{l-q,i})$ in place of $X_{l-q,i}$ because $U_{l,i,j}$ is the measurement error associated with $X_{l-q,i}/P_{l-q}(X_{l-q,i})$. Finally, for $l = 2q + 1, \dots, 2q + p$, $s_l(X_{l-q,i})$ is specified as a function of $X_{l-q,i}$. Flexible penalized mixtures of quadratic B-spline bases are used to model these functions as

$$s_l^2(X_{l-q} \mid \boldsymbol{\vartheta}_l) = \sum_{k=1}^{K_l} b_{2,l,k}(\tilde{X}_{l,i}) \exp(\vartheta_{l,k}) \text{ for } l = q + 1, \dots, 2q + p,$$

$$f(\boldsymbol{\vartheta}_l \mid \sigma_{\boldsymbol{\vartheta},l}^2) \propto (2\pi\sigma_{\boldsymbol{\vartheta},l}^2)^{-K_l/2} \exp\{-\boldsymbol{\vartheta}_l^\top \boldsymbol{\Lambda}_l \boldsymbol{\vartheta}_l / (2\sigma_{\boldsymbol{\vartheta},l}^2)\}, \quad \sigma_{\boldsymbol{\vartheta},l}^2 \sim \text{Inv-Ga}(a_{\boldsymbol{\vartheta}}, b_{\boldsymbol{\vartheta}}),$$

where $\boldsymbol{\Lambda}_l$ is defined similarly to its definition in Section 2.2.

2.4 Modeling the Probabilities $P_l(X_l)$

As discussed in Section 2.1, $P_l(X_l)$ is the probability of observing a positive value for the l -th zero-inflated component of the i -th individual. With $f_{\epsilon,l}(\epsilon) = \text{Normal}(0, 1)$ and $s_l(X_{l,i}) = 1$ for $l = 1, \dots, q$, as specified in Section 2.3, we have $P_l(X_l) = \Phi\{h_l(X_l)\}$. While a probit link is used to model $P_l(X_l)$, the overall mapping can be made highly flexible by allowing $h_l(X_l)$ to be a flexible function of X_l . Recycling the quadratic B-splines introduced earlier, $h_l(X_l)$ is thus again modeled as a penalized mixture of these basis functions as

$$h_l(X_l \mid \boldsymbol{\beta}_l) = \sum_{k=1}^{K_l} b_{2,l,k}(X_l) \beta_{l,k} = \mathbf{B}_{2,l,K_l} \boldsymbol{\beta}_l,$$

$$f(\boldsymbol{\beta}_l \mid \sigma_{\boldsymbol{\beta},l}^2, \boldsymbol{\mu}_{\boldsymbol{\beta},l}, \boldsymbol{\Sigma}_{\boldsymbol{\beta},l}) \propto (2\pi\sigma_{\boldsymbol{\beta},l}^2)^{-K_l/2} \exp\{-\boldsymbol{\beta}_l^\top \boldsymbol{\Lambda}_l \boldsymbol{\beta}_l / (2\sigma_{\boldsymbol{\beta},l}^2)\} \text{MVN}_{K_l}(\boldsymbol{\beta}_l \mid \boldsymbol{\mu}_{\boldsymbol{\beta},l,0}, \boldsymbol{\Sigma}_{\boldsymbol{\beta},l,0}),$$

$$\sigma_{\boldsymbol{\beta},l}^2 \sim \text{Inv-Ga}(a_{\boldsymbol{\beta}}, b_{\boldsymbol{\beta}}),$$

where $\mathbf{\Lambda}_l$ is defined similarly to its definition in Section 2.2.

However, it is often natural for $P_l(X_l)$ to be an increasing function of X_l . For instance, in our nutritional epidemiology example, where X_l represents long-term average intake of an episodic component, $P_l(X_l)$ is the probability of having a positive 24-hour recall for that component and is therefore expected to increase with X_l . Now, for $P_l(X_l)$ to be monotonically increasing in X_l is equivalent to requiring that $h_l(X_l)$ is monotonically increasing in X_l . We thus enforce the additional constraints $-\infty < \beta_{l,1} \leq \beta_{l,2} \leq \dots \leq \beta_{l,K_l} < \infty$ on the mixing coefficients which ensure that $h_l(X_l)$ is monotonically increasing (see Appendix A). The restriction enforced on β_l can also be written in the form $\mathbf{a} \leq \mathbf{D}\beta_l \leq \mathbf{b}$, where

$$\mathbf{a} = \begin{pmatrix} -\infty \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} \infty \\ \infty \\ \vdots \\ \infty \end{pmatrix}, \quad \mathbf{D} = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ -1 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & -1 & 1 \end{pmatrix}.$$

2.5 Model Flexibility

The aforementioned likelihood construction, along with the associated prior specifications, yields a highly flexible and extensible Bayesian framework for both univariate and multivariate density deconvolution in realistic measurement error settings, including when the surrogates are zero-inflated. Specifically, the mixture models used for the marginal densities $f_{X,l}(X_l)$ of interest can approximate broad classes of continuous distributions (Norets and Pelenis, 2012; Sarkar et al., 2018). Similarly, the mixture models for the marginal scaled error densities $f_{\epsilon,l}(\epsilon_l)$ can approximate very large classes of densities having zero means (Pelenis, 2014; Sarkar et al., 2018). Additionally, the B-spline mixture models for the variance functions $s_l^2(X_{l-q})$ and the probabilities $P_l(X_l)$ can approximate large classes of smooth continuous functions (Ruppert, 2002). While theoretical approximation guarantees require the number of mixture components and B-spline basis functions to grow asymptotically, in practice, suitably large but finite choices yield highly flexible, smooth, data-adaptive estimates with excellent empirical performance (Ishwaran and James, 2002; Ishwaran and Zarepour, 2002). Furthermore, the model for the correlation matrices has been shown to have full support over the space of all valid correlation matrices, enabling flexible modeling of dependence structures (Sarkar et al., 2021). Next, the model $(\mathbf{U} | \mathbf{X}) = \mathbf{S}(\mathbf{X})\boldsymbol{\epsilon}$, with a diagonal $\mathbf{S}(\mathbf{X})$, does impose some assumptions on how $\mathbf{U}$ varies with $\mathbf{X}$. In principle, more flexible models that allow all aspects of $\mathbf{U}$ to vary arbitrarily with $\mathbf{X}$ can potentially be conceived. However, such approaches are generally not practical for measurement error problems, as they tend to be numerically highly unstable, even in univariate settings (Sarkar et al., 2014). Additionally, as discussed in detail in Sarkar et al. (2018), the multiplicative structural assumption is in fact often quite meaningful in multivariate measurement problems. Furthermore, note that the framework also automatically accommodates a multiplicative error model, $\mathbf{W} = \mathbf{X}\tilde{\mathbf{U}}$ with $\tilde{\mathbf{U}}$ independent of $\mathbf{X}$ and $\mathbb{E}(\tilde{\mathbf{U}}) = 1$ (Iturria et al., 2002), since it can be reformulated as an additive model with conditionally heteroscedastic errors as discussed above as $\mathbf{W} = \mathbf{X} + \mathbf{X}(\tilde{\mathbf{U}} - 1) = \mathbf{X} + s(\mathbf{X})\boldsymbol{\epsilon}$ with $s(\mathbf{X}) = \mathbf{X}$, $\boldsymbol{\epsilon} = \tilde{\mathbf{U}} - 1$ independent of $\mathbf{X}$ and $\mathbb{E}(\boldsymbol{\epsilon}) = 0$ (Sarkar et al., 2018). Finally, while the assumption of a parametric Gaussian copula may seem restrictive, distinguishing between different copula families has been shown to be extremely challenging, even in much simpler settings without measurement error (Silva and Lopes, 2008). In complex deconvolution contexts, the Gaussian copula thus offers a pragmatic and computationally efficient compromise.

Altogether, the Bayesian hierarchical framework implemented by the **BayesDecon** package provides a highly flexible and practically effective tool for density deconvolution, accommodating both additive *and* multiplicative measurement errors in complex but realistic settings. In numerical experiments, such frameworks have consistently vastly outperformed pre-existing alternatives across diverse scenarios, including in simpler parametric cases for which those alternatives were specifically designed (Sarkar et al., 2014, 2018, 2021). While not the main focus of the package, options for fitting various simpler parametric models, such as homoscedastic and/or Gaussian or Laplace distributed errors, are also included for interested methodologists and practitioners.

3 Posterior Sampling

In practice, the different components X_l for $l = 1, \dots, q + p$ can have very different units of measurement and ranges of numerical values. To make the variables unit-free and bring them onto a common scale, we transform the corresponding observed semi-continuous and continuous proxies $Y_{l,i,j}$ for $l = q + 1, \dots, 2q + p$ as

$$Y_{l,i,j,trans} = 10 \times \frac{Y_{l,i,j} - \min\{Y_{l,i,j}\}}{\max\{Y_{l,i,j}\} - \min\{Y_{l,i,j}\}}.$$

Henceforth, for simplicity but with some abuse of notation, we will denote both the original and the transformed variables by the same notation $Y_{l,i,j}$. Such simple linear transformations do not adversely affect analysis (Sarkar et al., 2014) unlike non-linear Box-Cox transformations used in previous literature (Zhang et al., 2011), which often result in loss of information and introduce bias. Transforming the variables to be unit-free and supported on a common domain also greatly facilitates model construction and prior specification across different components. Due to the additive nature of the errors, the observed surrogate $Y_{l+q,i,j}$ typically exhibits a wider range than the corresponding latent variable $X_{l,i}$. To account for this, we choose the bounds $[A_l, B_l]$ for the distributions of $X_{l,i}$ as $A_l = 0$ and $B_l = 10 \times r_l$ for $l = 1, \dots, q + p$. For dietary recall data, a value of $r_l = 0.5$ usually provides a good choice.

The choices of hyperparameters and MCMC sampling steps in our multivariate model mostly follow those described in Sarkar et al. (2021). However, the implementation now relies entirely on a C++ backend. We now also employ an improved algorithm, introduced in Robert (1995) and discussed in Section B of this article, for fitting the truncated normal components in the mixture models for $f_{X,l}(X_l)$, $l = q + 1, \dots, q + p$. Additionally, the updates for β_l , $l = 1, \dots, q$ are also different since they now satisfy monotonicity constraints discussed in Section 2.4. Specifically, the full conditionals of β_l 's are available in closed form as

$$\begin{aligned} f(\beta_l | \cdot) &\propto \text{MVN}_{K_l}(\beta_l | \mu_{\beta,l}, \Sigma_{\beta,l}) 1(a \leq D\beta_l \leq b), \quad \text{where} \\ \mu_{\beta,l} &= \Sigma_{\beta,l}^{-1} \{ \Sigma_{\beta,l,0}^{-1} \mu_{\beta,l,0} + \sum_{i=1}^n \sum_{j=1}^{m_i} W_{l,i,j} \mathbf{B}_{2,l,K_l}(\tilde{X}_{l-q,i})^\top \mathbf{B}_{2,l,K_l}(\tilde{X}_{l-q,i}) \}, \\ \Sigma_{\beta,l} &= \{ \sigma_{\beta,l}^{-2} \mathbf{A}_l + \Sigma_{\beta,l,0}^{-1} + \sum_{i=1}^n \mathbf{B}_{2,l,K_l}(\tilde{X}_{l-q,i})^\top \mathbf{B}_{2,l,K_l}(\tilde{X}_{l-q,i}) \}^{-1}. \end{aligned}$$

Sampling from the above full conditional can be done using the idea of truncated multivariate normal sampling by Geweke (1991). We first sample $\mathbf{V}$ from $\text{TMVN}_{K_l}(\mathbf{0}, D\Sigma_{\beta,l}D^\top, [a - \mu_{\beta,l}, b - \mu_{\beta,l}])$, then the transformation $\mu_{\beta,l} + D^{-1}\mathbf{V}$ yields a sample from the full conditional distribution of β_l . Sampling from a truncated normal distribution can again be done using the algorithm from Robert (1995) discussed in Section B.

4 The BayesDecon Package

The **BayesDecon** package requires a data frame containing the surrogate variables as input, carries out a simple transformation of the variables to make them unit-free and have a shared support, runs MCMC to fit the Bayesian hierarchical model described in Section 2, and returns posterior samples of all latent variables and model parameters in their original scale. It also produces summarized plots of the density of X_l , the density of the ‘scaled’ errors ε_l and the variance function $s_\ell^2(X_l)$.

For illustration, we use the NHANES data, which provides dietary information for the US population. NHANES is conducted by the Centers for Disease Control and Prevention (CDC) every two years. The What We Eat in America (WWEIA) interview component of NHANES collects two 24-hour dietary recalls from each selected individual, along with their demographic information. These 24-hour recalls can be treated as surrogates for an individual’s long-term average intake, from which the distribution of long-term average intake in the US population can be estimated.

In the following subsections, we discuss the preprocessing of the data set and then apply various functions available in the **BayesDecon** package to the data set, along with their brief descriptions. All analyses demonstrating the functionality of the package are based on MCMC chains of 7000 iterations, of which the initial 2000 are discarded as burn-in and the remaining 5000 are thinned by a factor of 5, resulting in a final sample of 1000 draws. With the above-specified MCMC parameters, univariate analysis completes in under 5 minutes, whereas a multivariate analysis, incorporating all 13 variables listed in Table 1, requires less than 3 hours to run. Further discussions regarding computational runtime are provided in Section 4.4.

4.1 Preprocessing the NHANES Data

A convenient way to obtain the NHANES data set is from the package **heiscore** (Ramdas et al., 2024; Ho et al., 2024). The `selectDataset` function from the **heiscore** package fetches a tibble containing demographic information and unprocessed 24-hour recall data from the 2005-2006 NHANES cycle through the 2017-2018 cycle. To demonstrate our package, we work with the data set corresponding to the 2017–2018 cycle. This data set gives 24-hour dietary recalls for 8704 individuals with 2 replicates available for each component. After excluding records with missing responses, data for 6482 individuals remain. Among those, 2 rows are further eliminated due to lack of complete information. There are 22 dietary components in the data set, of which 13 are used to calculate the Healthy Eating Index (HEI)-2020 (Shams-White et al., 2023; Ho et al., 2024). The selected variables and their code names are given in Table 1. The proportion of zero values was computed for each variable, and variables with a proportion greater than 0.05 were classified as episodic, consistent with previous literature (Zhang et al., 2011). The final data set consists of 6482 individuals, each with two replicates, comprising seven episodic components and six regular components.

```
library(heiscore)
library(tidyverse)
library(BayesDecon)

data.raw <- selectDataset("1718") %>%
```

Table 1: Description of variables.

Sl	Variable label	Variable code	Consump- tion Pattern	Proportion of zeros in the data set
1	Total Fruit	T_F_TOTAL	episodic	0.30
2	Whole Fruits	FWHOLEFRT	episodic	0.44
3	Total Vegetables	VTOTALLEG	episodic	0.08
4	Greens and Beans	VDRKGRLEG	episodic	0.66
5	Whole Grains	T_G_WHOLE	episodic	0.48
6	Total Dairy	T_D_TOTAL	episodic	0.10
7	Seafood/Plant Proteins	PFSEAPLANTLEG	episodic	0.45
8	Total Protein	PFALLPROTLEG	regular	0.04
9	Fatty Acids	TFACIDS	regular	0.00
10	Refined Grains	T_G_REFINED	regular	0.04
11	Sodium	TSODI	regular	0.00
12	Added Sugars	T_ADD_SUGARS	regular	0.03
13	Saturated Fat	TSFAT	regular	0.00

```

select(-SEQN, -WTDRD1, -WTDR2D, -SEX, -RACE_ETH, -AGE, -FAMINC) %>%
na.omit()
data.raw <- data.raw[-c(4437, 5987),]
d <- ncol(data.raw)/2
n <- nrow(data.raw)
var.name <- unique(gsub("^DR1_?"|^DR2_?", "", names(data.raw)))
data <- as.data.frame(rbind(as.matrix(data.raw[, (1:d)*2-1]),
                           as.matrix(data.raw[, (1:d)*2])))
colnames(data) <- var.name
SEQN <- rep(1:n, times=2)

data <- cbind(SEQN, data) %>%
  sort_by(SEQN) %>%
  select(SEQN, T_F_TOTAL, FWHOLEFRT, VTOTALLEG, VDRKGRLEG,
         T_G_WHOLE, T_D_TOTAL, PFSEAPLANTLEG, PFALLPROTLEG,
         TFACIDS, T_G_REFINED, TSODI, T_ADD_SUGARS, TSFAT)
print(head(data), row.names = FALSE)

```

4.2 Description of Various Functions

4.2.1 Obtaining MCMC Samples

The application of the *BayesDecon* package begins with its core function, `bdeconv`. It takes a data frame as input, where the first column represents an identifier for the individual and the remaining columns represent different surrogates for that individual (as shown in Section 4.1), and returns MCMC samples of all model parameters of the desired size. Several arguments to

Table 2: Arguments of `bdeconv` function.

Argument	Description
<code>err_type</code>	Specifies the error type: <code>"flex"</code> corresponds to the model described in Section 2.3, <code>"norm"</code> assumes normally distributed errors, and <code>"lapl"</code> assumes Laplace distributed errors.
<code>var_type</code>	Specifies whether the error is homoscedastic (<code>"homoscedastic"</code>) or heteroscedastic (<code>"heteroscedastic"</code>), as defined in Section 2.3.
<code>res_type</code>	Specifies the format of the MCMC output: <code>"mean"</code> returns the average of all MCMC samples, <code>"final"</code> returns the last sample, and <code>"all"</code> returns the full set of samples.
<code>regular_var</code>	Specifies which variables to consider as regular; all others are treated as zero-inflated. If any zero values exist in the regular variables, those are replaced with half of the minimum nonzero value observed for that variable (see Zhang et al., 2011). If not provided, variables with all strictly continuous surrogates are considered regular.
<code>scaling</code>	A logical value indicating whether the observed variables are linearly transformed to the common interval $[0,10]$ before fitting the model, as described in Section 3. The default is <code>TRUE</code> .
<code>upr</code>	Specifies the upper bound for the deconvolved estimates. If <code>scaling = TRUE</code> , it specifies the ratio of this upper bound to the upper bound of the observed surrogates. If <code>scaling = FALSE</code> , it directly specifies the upper bound. By default, the upper bound is 0.5 times the upper bound of the surrogates.
<code>lwr</code>	Specifies the lower bound for the deconvolved estimates when <code>scaling = FALSE</code> ; it is not used when <code>scaling = TRUE</code> . For zero-inflated variables, the lower bound is automatically set to 0. The default is 0.

specify the model can be passed to this function, all of which have default values as discussed in Sections 2 and 3, and they are listed in Table 2. All hyperparameters of the model and variables necessary for the MCMC sampler can be specified through a list via the `control` argument; some of the most important ones are listed in Table 3. The full list is available in the package documentation on CRAN but is curtailed here to conserve space. Generally speaking, we employed informative priors that induce strong smoothness on the various functions modeled by mixtures of B-splines (specifically, the variance function, and the density and probability-of-consumption functions for episodic variables), and non-informative or weakly informative priors for other parameters.

For multivariate data, the marginal parameters are initialized by first running the corresponding univariate marginal models for a few iterations. The number of these iterations can be specified using the `niter_uni` argument within the `control` list. These initial analyses for different components can be parallelized by setting `parallel = TRUE`. The number of cores to be used in parallelization can also be specified via the `core_num` argument. The progress of the MCMC chain can be seen using `progress = TRUE`. The output obtained from the function can

Table 3: Additional arguments passed through `control` argument of `bdeconv` function.

Argument	Description	Default value
<code>nsamp</code>	Number of samples.	1000
<code>nburn</code>	Burn-in period.	2000
<code>nthin</code>	Thinning interval.	5
<code>K_t</code>	K_t is the number of knot points for the B-splines.	10
<code>K_x</code>	Number of mixture components in the model for the density of X when the surrogates are strictly continuous. Not applicable when the surrogates are zero-inflated in which case the density is modeled using a normalized mixture of B-splines.	10
<code>K_epsilon</code>	Number of mixture components in the model for the density of scaled errors ϵ when applicable.	10

be used for estimating the density or cumulative density of the latent variables of interest and associated errors, plotting these estimated densities, the associated variance functions, etc.

```
## Sample for univariate analysis of the variable Added Sugars (regular)
set.seed(123)
res_uni_1 <- bdeconv(data[,c(1,13)], res_type = 'all', upr = 0.3,
                     regular_var = "T_ADD_SUGARS", progress = TRUE)
```

```
## Sample for univariate analysis of the variable Total Dairy (zero-inflated)
set.seed(123)
res_uni_2 <- bdeconv(data[,c(1,7)], res_type = 'all', upr = 0.5,
                     progress = TRUE)
```

```
## Sample for multivariate analysis of all the variables
set.seed(123)
res_mv <- bdeconv(data, res_type = 'all', upr = 0.5,
                  regular_var = c("PFALLPROTLEG", "TFACIDS", "T_G_REFINED",
                                "TSODI", "T_ADD_SUGARS", "TSFAT"),
                  control = list(nburn = 5000), progress = TRUE, core_num = 8,
                  parallel = TRUE)
```

4.2.2 Visualizing Estimates

The object returned by the `bdeconv` function can be directly passed to the `plot` function, which produces various types of visualizations for univariate zero-inflated, univariate regular, or multivariate data. By default, the `plot` function utilizes the `ggplot2` package (Wickham, 2016), if it is available in the system library; otherwise, it falls back to the base R plot function. This behavior can be modified by setting the argument `use_ggplot = FALSE` within the plot call. The same plot can also be generated using the `ggplot2::autoplot` function, which is commonly used for visualizing a wide range of object classes in a consistent and extensible manner.

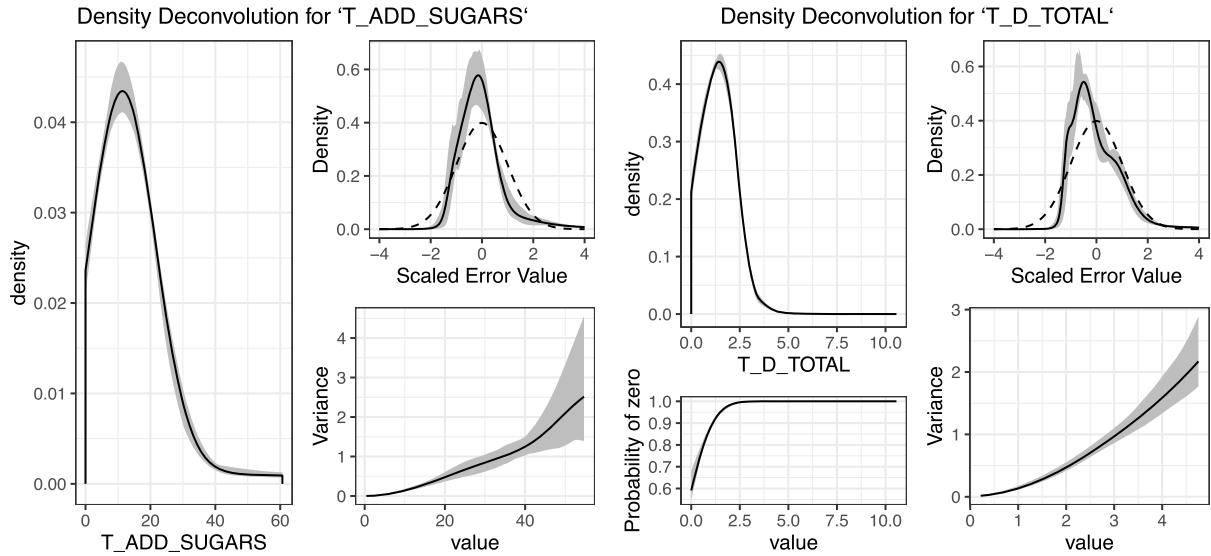


Figure 1: Plots generated by `plot(res_uni_1)` (left) and `ggplot2::autoplot(res_uni_2)` (right).

From the plots in Figure 1, it can be seen that the `plot` function produces different outputs for regular (Added Sugars) and zero-inflated (Total Dairy) variables. For regular variables, the plots include the estimated densities $f_{X,l}(X_l)$ and $f_{\epsilon,l}(\epsilon_l)$ and their 95% pointwise credible intervals, along with sample estimates of individual variances and the estimated variance functions $s_l^2(X_{l-q})$. In the plots for the densities $f_{\epsilon,l}(\epsilon_l)$ of the scaled errors, the standard Gaussian is also shown as a dashed line for reference. In contrast, for zero-inflated variables, an additional plot showing the probability $P_l(X_l)$ of observing a continuous surrogate is also provided. For multivariate data, the `plot` function can also display the estimated densities of the constituent components along the diagonal, scatter plots for variable pairs in the lower triangle, and contour plots of their estimated bivariate joint densities in the upper triangle (see right panel of Figure 2).

```
plot(res_uni_1)
ggplot2::autoplot(res_uni_2)
```

4.2.3 Subsetting MCMC Samples

The output of the `bdeconv` function for multivariate data can be indexed in the same way as data frames. This feature is particularly useful for extracting samples of marginals or the joint distribution of selected variables and visualizing them accurately, as illustrated below.

```
res_sub_1 <- res_mv[,13]
res_sub_2 <- res_mv[,c(3,8,11)]
plot(res_sub_1)
plot(res_sub_2, use_ggplot = FALSE)
```

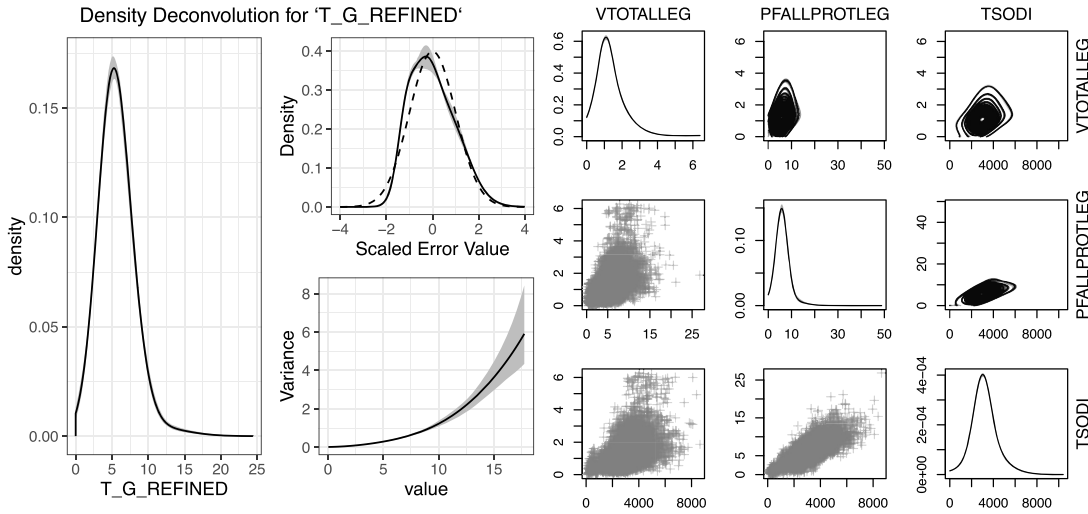


Figure 2: Plots generated by `plot(res_sub_1)` (left) and `plot(res_sub_2, use_ggplot = FALSE)` (right).

4.2.4 Printing a Summary of MCMC Samples

The five-number summary of all variables and the corresponding errors can be obtained using the `print` function, or simply by calling the output of the `bdeconv` function—similar to how results are displayed for model-fitting functions like `lm` or `glm`.

```
res_uni_1
```

```
Call: bdeconv.data.frame(obj = data[, c(1, 13)], upr = 0.3,
  regular_var = "T_ADD_SUGARS", res_type = "all", progress = TRUE)
Number of samples: 1000
Number of variables: 1
```

```
Variable: T_ADD_SUGARS
```

	Min.	1Q	Med.	3Q	Max.
y	5.000000e-03	4.870000	10.945000	20.010000	202.24000
x	1.163347e-05	7.845026	13.4674509	20.003974	60.67175
u	-6.061909e+01	-5.271390	-0.7750663	2.915994	166.92112

4.3 Internal Object Structures and class Assignments

The output of the `bdeconv` function is assigned to a different `class` depending on the nature of the variable: univariate zero-inflated, univariate regular, or multivariate. The assigned `class` for different variable types, summarized in Table 4, determines the behavior of various generic functions such as `plot`, `autoplot`, `dens_x` and `dist_x`, when applied to the output of the `bdeconv` function. For example, as we have already seen, the behavior of the `plot` function varies depending on the type of variable; this is achieved through the use of specific S3 methods—`plot.zi_res`, `plot.nc_res`, and `plot.mv_res`, which are automatically dispatched based on the nature of the input data set.

Table 4: `class` assigned to `bdeconv` output.

Variable type	<code>class</code> assigned
Univariate zero-inflated	<code>bdeconv_result</code> , <code>uv_res</code> , <code>zi_res</code>
Univariate regular	<code>bdeconv_result</code> , <code>uv_res</code> , <code>nc_res</code>
Multivariate	<code>bdeconv_result</code> , <code>mv_res</code>

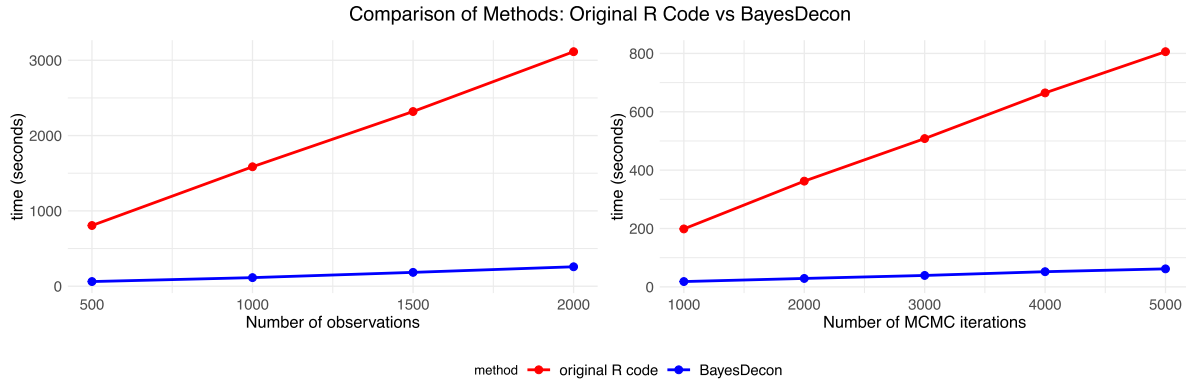


Figure 3: Run times for (a) 5000 MCMC iterations for subsets of varying sizes from the original data (left), and (b) MCMC with a varying number of iterations for a fixed subset of size 1000 from the original data (right).

4.4 Runtime Comparisons

As discussed in the Introduction, implementing the algorithm in C++ using Rcpp yields substantial performance gains over the original R implementation, as demonstrated on a synthetic data set. Here, we demonstrate similar gains with the NHANES data set. We used subsets with 500, 1000, 1500, and 2000 randomly sampled individuals from the complete data set and three dietary variables: Total Fruits, Sodium, and Saturated Fat. For each subset, 5000 MCMC iterations were run to compare the performance of the package with that of the baseline R implementation. We also experimented with MCMC runs of different lengths on a fixed data set of size 1000, and once again, the package implementation demonstrated substantially greater efficiency. For 2000 observations and 5000 MCMC iterations, the package proved to be nearly 12 times more efficient than the original codes. Figure 3 depicts the results of the above experiments.

4.5 Additional Experiments and Illustrations

Additional numerical experiments and illustrations, emphasizing recovery and reproducibility in both simulated and NHANES data settings, are provided in the supplementary materials. Specifically, we replicated the simulation study from Sarkar et al. (2021) and compared the integrated squared errors (ISEs) for recovering true marginal and joint densities in three-component deconvolution scenarios with varying shapes for the marginal densities of both $\mathbf{X}$ and associated errors ϵ and varying levels of zero-inflation across the components. The BayesDecon package generally performed comparably to the original R codes but also delivered some improvements for components where the previous implementation was relatively less effective. These gains may be attributed to enhancements in modeling features, such as enforcing monotonicity in consumption

probabilities and adopting a numerically more stable algorithm for sampling truncated normals. We further assessed reproducibility across multiple runs and when applied to different subsets of variables in both synthetic and NHANES data. Overall, *BayesDecon*'s performance was robust and consistent across all cases, particularly in simulations with three replicates per subject. For NHANES, with only two replicates per subject, we observed some variability in the estimated density of the scaled errors ϵ across multiple seeds. This can be attributed to the challenge of clearly delineating the scaled errors ϵ from the variance functions $s^2(X)$ from limited data, especially when X has a large mass near zero where the measurement errors $U = s(X)\epsilon$ are inherently very small, and can be further complicated by the presence of many exact zero recalls in this region in episodic cases. In such cases, strong prior regularization of the error distribution can stabilize its estimates. Crucially, the estimated f_X 's, which are of main interest, still remain highly consistent across all scenarios tested. Their updates depend on the product $U = s(X)\epsilon$, and when these values are small, the corresponding recalls are in fact highly informative about the underlying X values, making them robust against uncertainty in the scaled error distribution estimates. See the supplementary materials for additional details.

5 Discussion

The *BayesDecon* package provides an efficient implementation of the methods developed in Sarkar et al. (2014, 2018, 2021), leveraging C++ and the *RcppArmadillo* library for computational speed gains. The package offers user-friendly tools for estimating and visualizing deconvolved densities from real data subject to measurement error, such as the NHANES data set discussed in Section 4, with enhancements in certain modeling aspects and numerical stability as well as substantial improvements in computational efficiency.

Additionally, the package supports classical measurement error models, including both homogeneous and heterogeneous error structures, which can be specified via optional arguments (e.g., `var_type = "homoscedastic"`, `err_type = "norm"`) in the `bdeconv` function. While these features are not the primary focus of the package, they are included to enhance flexibility and compatibility with a broader range of modeling scenarios. We anticipate that the package will facilitate simulation-based analyses for assessing model performance, guiding further methodological development in deconvolution research, building on the frameworks established in the aforementioned papers as well as supporting advancements in other directions.

The current version of the *BayesDecon* package is designed to estimate deconvolved densities for both univariate and multivariate data, including settings with zero inflation in the error-contaminated surrogates. We remain committed to incorporating user feedback as we receive it. We also plan to expand and update the package to reflect advances in this area, such as accommodating survey weights (Wu and Stephenson, 2024), incorporating covariate information to estimate group-specific deconvolved densities (Sarkar, 2022).

Supplementary Material

Supplementary materials present results from additional simulation experiments and further analyses of the NHANES data set, providing additional validation of the efficiency and reliability of the *BayesDecon* package. A separate zip file includes data and codes to replicate the results therein.

Appendix

A Monotonicity of $h(x)$ Under Additional Constraints

Lemma A.1. *If $-\infty < \beta_1 \leq \beta_2 \leq \dots \leq \beta_J < \infty$, then $h(x) = \sum_{j=1}^J b_{2,j}(x)\beta_j = \mathbf{B}_{2,J}\boldsymbol{\beta}$ is an increasing function within its support.*

Proof. Consider knot points $t_1 = t_2 = t_3 < t_4 < \dots < t_{J-1} < t_J < t_{J+1} = t_{J+2} = t_{J+3}$ where $t_{3:J+1}$ are equidistant with $\delta = t_4 - t_3$. For $j = 3, \dots, J-2$, the basis functions $b_{2,j}$ are defined as

$$b_{2,j} = \begin{cases} \{(x - t_j)/\delta\}^2/2, & \text{if } t_j < x < t_{j+1}, \\ -\{(x - t_{j+1})/\delta\}^2/2 + (x - t_{j+1})/2 + 1/2, & \text{if } t_{j+1} < x < t_{j+2}, \\ \{1 - (x - t_{j+2})/\delta\}^2/2, & \text{if } t_{j+2} < x < t_{j+3}, \\ 0 & \text{otherwise.} \end{cases}$$

For $j = 1, 2, J-1, J$, the basis functions are defined as follows.

$$\begin{aligned} b_{2,1} &= \begin{cases} \{1 - (x - t_3)/\delta\}^2/2, & \text{if } t_3 < x < t_4, \\ 0, & \text{otherwise,} \end{cases} \\ b_{2,2} &= \begin{cases} -\{(x - t_3)/\delta\}^2/2 + (x - t_3)/2 + 1/2, & \text{if } t_3 < x < t_4, \\ \{1 - (x - t_4)/\delta\}^2/2, & \text{if } t_4 < x < t_5, \\ 0, & \text{otherwise,} \end{cases} \\ b_{2,J-1} &= \begin{cases} \{(x - t_{J-1})/\delta\}^2/2, & \text{if } t_{J-1} < x < t_J, \\ -\{(x - t_J)/\delta\}^2/2 + (x - t_J)/2 + 1/2, & \text{if } t_J < x < t_{J+1}, \\ 0, & \text{otherwise,} \end{cases} \\ b_{2,J} &= \begin{cases} \{(x - t_J)/\delta\}^2/2, & \text{if } t_J < x < t_{J+1}, \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

From the above equations, we can express $h(x)$ for $x \in (t_j, t_{j+1})$ with $j = 3, \dots, J$, as $h(x) = \beta_{j-2}[\{1 - (x - t_j)/\delta\}^2/2] + \beta_{j-1}[-\{(x - t_j)/\delta\}^2/2 + (x - t_j)/2 + 1/2] + \beta_j[\{1 - (x - t_j)/\delta\}^2/2]$. So, to check whether $h(x)$ is increasing within the interval (t_j, t_{j+1}) , it is sufficient to show that the function $g(y) = \beta_{j-2}(1 - y)^2/2 + \beta_{j-1}[-y^2 + y + 1/2] + \beta_j y^2/2$ is increasing for $y \in (0, 1)$ which is true since $g'(y) = -\beta_{j-2}(1 - y) + \beta_{j-1}(1 - 2y) + \beta_j y = (\beta_{j-1} - \beta_{j-2})(1 - y) + (\beta_j - \beta_{j-1})y \geq 0$. Now, using the continuity of $h(x)$ at the knot points, we conclude that $h(x)$ is increasing within its support. $\square$

B Sampling from a Truncated Normal Distribution

Consider the problem of sampling from $TN(0, 1, [a, b])$. Let $\Phi(x)$ denote the cumulative distribution function of a standard normal distribution evaluated at x . Naive accept-reject sampling using the distribution $\text{Normal}(0, 1)$ is a possible solution, but it has an acceptance probability $\Phi(b) - \Phi(a)$ which becomes very small when $\Phi(a)$ and $\Phi(b)$ are very close. If v is a sample drawn from $\text{Uniform}(\Phi(a), \Phi(b))$, then the transformation $z = \Phi^{-1}(v)$ yields a sample from $TN(0, 1, [a, b])$. Though this method can avoid rejection sampling, it requires accurate computation of the function $\Phi(x)$ which is not possible even for moderately large values of x , for example $|x| > 6$. Robert (1995) proposed an efficient algorithm to sample from an univariate truncated normal using three different accept-reject algorithms based on the values of a and b . Based on the support of the distribution $[a, b]$, they found the optimal algorithm that maximizes the acceptance probability of the sample. The algorithm is described below.

1. **If** $a < 0$, $b = \infty$, **or** $a = -\infty$, $b > 0$, **or** $a < 0$, $b > 0$, $b - a > \sqrt{2\pi}$: Sample from $\text{Normal}(0, 1)$ and accept if it lies within $[a, b]$, otherwise reject and sample again.
2. **If** $a > 0$, $b > a + \frac{2\sqrt{e}}{a+\sqrt{a^2+4}} \exp\left\{\frac{a^2-a\sqrt{a^2+4}}{4}\right\}$: Sample z from $\text{Shifted Exponential}(\lambda, a)$ where $\lambda = (a + \sqrt{a^2 + 4})/2$. If $z > b$, reject it and sample again. If $z \leq b$, accept it with probability $e^{-\frac{(z-\lambda)^2}{2}}$, otherwise reject it and sample again.
3. **If** $b < 0$, $-a > -b + \frac{2\sqrt{e}}{-b+\sqrt{b^2+4}} \exp\left\{\frac{b^2+b\sqrt{b^2+4}}{4}\right\}$: Sample z from $\text{Shifted Exponential}(\lambda, -b)$ where $\lambda = (-b + \sqrt{b^2 + 4})/2$. If $z > -a$, reject it and sample again. If $z \leq -a$, accept it with probability $e^{-\frac{(z-\lambda)^2}{2}}$, otherwise reject it and sample again.
4. **Else**: Sample z from $\text{Uniform}(a, b)$. Define

$$\rho(z) = \begin{cases} e^{-\frac{z^2}{2}}, & \text{if } a < 0, b > 0, \\ e^{-\frac{b^2-z^2}{2}}, & \text{if } b < 0, \\ e^{-\frac{a^2-z^2}{2}}, & \text{if } a > 0. \end{cases}$$

Accept z with probability $\rho(z)$, otherwise reject it and sample again.

Here $\text{Shifted Exponential}(\lambda, \mu)$ denotes a shifted exponential distribution with location μ and rate λ given by the pdf

$$f(x) = \frac{1}{\lambda} e^{-\lambda(x-\mu)} 1(x > \mu).$$

To generate a sample from $\text{TN}(\mu, \sigma^2, [a, b])$, we can generate a sample z from $\text{TN}(0, 1, [\frac{a-\mu}{\sigma}, \frac{b-\mu}{\sigma}])$ and then use the transformation $t = \mu + \sigma z$. To sample from a truncated multivariate normal distribution with location $\mathbf{0}$, scale $\tilde{\Sigma}$, and support restricted to the hyper-rectangle defined by the lower and upper bounds $\mathbf{l}$ and $\mathbf{u}$, we use a Gibbs sampler as done in Robert (1995), Geweke (1991).

Acknowledgments

We also thank the NHANES investigators, staff, and participants for their dedicated work in conducting the survey and for providing the data used in this research.

Funding

We gratefully acknowledge support for this research provided by the National Science Foundation (NSF) Grant DMS-2515902.

References

- Cai Y, Gu H, Kenney T (2022). *pmldecon: Deconvolution Density Estimation using Penalized MLE*. R package version 0.2.1.
- Carroll RJ, Ruppert D, Stefanski LA, Crainiceanu CM (2006). *Measurement Error in Nonlinear Models: A Modern Perspective*. Chapman and Hall/CRC.
- de Boor C (2001). *A Practical Guide to Splines*, volume 27 of *Applied Mathematical Sciences*. Springer, New York.
- Delaigle A, Hyndman T, Wang T (2026). *deconvolve: Deconvolution Tools for Measurement Error Problems*. R package version 0.1.0.

- Delaigle A, Meister A (2008). Density estimation with heteroscedastic error. *Bernoulli*, 14: 562–579. <https://doi.org/10.3150/08-BEJ121>
- Efron B (2016). Empirical Bayes deconvolution estimates. *Biometrika*, 103: 1–20. <https://doi.org/10.1093/biomet/asv068>
- Eilers PHC, Marx BD (1996). Flexible smoothing with B-splines and penalties. *Statistical Science*, 11: 89–121. <https://doi.org/10.1214/ss/1038425655>
- Geweke J (1991). Efficient simulation from the multivariate normal and student-t distributions subject to linear constraints and the evaluation of constraint probabilities. In: *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, volume 571, 578. Interface Foundation of North America, Inc., Fairfax, Virginia.
- Ho B, Ramdas V, Sarkar A (2024). HEI analysis of NHANES dietary data: Exploring the diet quality of Americans with R package heiscore. *Journal of Data Science*, 1: 1–20. <https://doi.org/10.6339/24-JDS1160>
- Ishwaran H, James L (2002). Approximate Dirichlet processes computing in finite normal mixtures: Smoothing and prior information. *Journal of Computational and Graphical Statistics*, 11: 209–230.
- Ishwaran H, Zarepour M (2002). Exact and approximate sum representations for the Dirichlet process. *Canadian Journal of Statistics*, 30: 269–283. <https://doi.org/10.2307/3315951>
- Iturria SJ, Carroll RJ, Firth D (2002). Polynomial regression and estimating functions in the presence of multiplicative measurement error. *Journal of the Royal Statistical Society, Series B, Statistical Methodology*, 61: 547–561. <https://doi.org/10.1111/1467-9868.00192>
- Kent D, Ruppert D (2023). Smoothness penalized deconvolution (SPeD) of a density estimate. *Journal of the American Statistical Association*, 119: 2407–2417. <https://doi.org/10.1080/01621459.2023.2259028>
- Keogh RH, Shaw PA, Gustafson P, Carroll RJ, Deffner V, . . . , Freedman LS (2020). STRATOS guidance document on measurement error and misclassification of variables in observational epidemiology: Part 1—basic theory and simple methods of adjustment. *Statistics in Medicine*, 39: 2197–2231. <https://doi.org/10.1002/sim.8532>
- Narasimhan B, Efron B (2020). deconvolveR: A g-modeling program for deconvolution and empirical Bayes estimation. *Journal of Statistical Software*, 94: 1–20.
- Norets A, Pelenis J (2012). Bayesian modeling of joint and conditional distributions. *Journal of Econometrics*, 168: 332–346. <https://doi.org/10.1016/j.jeconom.2012.02.001>
- Pelenis J (2014). Bayesian regression with heteroscedastic error density and parametric mean function. *Journal of Econometrics*, 178: 624–638. <https://doi.org/10.1016/j.jeconom.2013.10.006>
- Ramdas V, Ho B, Sarkar A (2024). *heiscore: Score and Plot the Healthy Eating Index from NHANES Data*. R package version 0.1.4.
- Robert CP (1995). Simulation of truncated normal variables. *Statistics and Computing*, 5: 121–125. <https://doi.org/10.1007/BF00143942>
- Ruppert D (2002). Selecting the number of knots for penalized splines. *Journal of Computational and Graphical Statistics*, 11: 735–757. <https://doi.org/10.1198/106186002853>
- Sarkar A (2022). Bayesian semiparametric covariate informed multivariate density deconvolution. *Journal of Computational and Graphical Statistics*, 31: 1153–1163. <https://doi.org/10.1080/10618600.2022.2060239>
- Sarkar A, Mallick BK, Staudenmayer J, Pati D, Carroll RJ (2014). Bayesian semi-parametric density deconvolution in the presence of conditionally heteroscedastic mea-

- surement errors. *Journal of Computational and Graphical Statistics*, 23: 1101–1125. <https://doi.org/10.1080/10618600.2014.899237>
- Sarkar A, Pati D, Chakraborty A, Mallick BK, Carroll RJ (2018). Bayesian semiparametric multivariate density deconvolution. *Journal of the American Statistical Association*, 113: 401–416. <https://doi.org/10.1080/01621459.2016.1260467>
- Sarkar A, Pati D, Mallick BK, Carroll RJ (2021). Bayesian copula density deconvolution for zero-inflated data in nutritional epidemiology. *Journal of the American Statistical Association*, 116: 1075–1087. <https://doi.org/10.1080/01621459.2020.1782220>
- Shams-White MM, Pannucci TE, Lerman JL, Herrick KA, Zimmer M, . . . , Reedy J (2023). Healthy eating index-2020: Review and update process to reflect the dietary guidelines for Americans, 2020–2025. *Journal of the Academy of Nutrition and Dietetics*, 123: 1280–1288. <https://doi.org/10.1016/j.jand.2023.05.015>
- Shaw PA, Gustafson P, Carroll RJ, Deffner V, Dodd KW, . . . , Freedman LS (2020). STRATOS guidance document on measurement error and misclassification of variables in observational epidemiology: Part 2—more complex methods of adjustment and advanced topics. *Statistics in Medicine*, 39: 2232–2263. <https://doi.org/10.1002/sim.8531>
- Silva RD, Lopes HF (2008). Copula, marginal distributions and model selection: A Bayesian note. *Statistics and Computing*, 18: 313–320. <https://doi.org/10.1007/s11222-008-9058-y>
- Staudenmayer J, Ruppert D, Buonaccorsi JP (2008). Density estimation in the presence of heteroscedastic measurement error. *Journal of the American Statistical Association*, 103: 726–736. <https://doi.org/10.1198/0162145080000000328>
- Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag, New York.
- Wu SM, Stephenson BJK (2024). Bayesian estimation methods for survey data with potential applications to health disparities research. *Wiley Interdisciplinary Reviews. Computational Statistics*, 16: e1633. <https://doi.org/10.1002/wics.1633>
- Zhang S, Midthune D, Guenther PM, Krebs-Smith SM, Kipnis V, Carroll RJ (2011). A new multivariate measurement error model with zero-inflated dietary data, and its application to dietary assessment. *Annals of Applied Statistics*, 5: 1456–1487.