

Structured Bayesian Regression Tree Models for Estimating Distributed Lag Effects: The R Package *dlmtree*

by Seongwon Im, Ander Wilson, and Daniel Mork

Abstract When examining the relationship between an exposure and an outcome, there is often a time lag between exposure and the observed effect on the outcome. A common statistical approach for estimating the relationship between the outcome and lagged measurements of exposure is a distributed lag model (DLM). Because repeated measurements are often autocorrelated, the lagged effects are typically constrained to vary smoothly over time. A recent statistical development on the smoothing constraint is a tree structured DLM framework. We present an R package *dlmtree*, available on CRAN, that integrates tree structured DLM and extensions into a comprehensive software package with user-friendly implementation. A conceptual background on tree structured DLMs and a demonstration of the fitting process of each model using simulated data are provided. We also demonstrate inference and interpretation using the fitted models, including summary and visualization. Additionally, a built-in shiny app for heterogeneity analysis is included.

1 Introduction

In many fields, there is interest in estimating the lagged relationship between an exposure (or treatment) and an outcome. In such cases, the length of the lag or how the exposure effect varies is often unknown. The lagged relationship can take either of two forms. The first form is the association between the exposure at one time point and an outcome distributed across several subsequent times. For example, the impact of advertisement persists beyond that single time point of the investment (Koyck, 1954; Palda, 1965) or exposure to an environmental pollutant affects mortality on the same day and each of the following days (Schwartz, 2000). The second form is an exposure assessed longitudinally and an outcome assessed post-exposure. Examples of this form are maternal exposure to environmental chemicals during pregnancy on birth outcomes and children's health and development (Wilson et al., 2017a; Chiu et al., 2023; Hsu et al., 2023) and the effect of training and recovery activities over multiple days on athlete wellness (Schliep et al., 2021). A popular statistical method to estimate the time-varying association between exposure and outcome is a distributed lag model (DLM).

The DLM regresses a scalar outcome on the exposure measured at preceding time points (Schwartz, 2000; Gasparrini et al., 2010). The DLM framework is particularly useful as it allows for estimating time-resolved exposure effects and quantifies the temporal relationship between the exposure and the outcome. Because the repeated measurements of exposure are often correlated, DLMs typically include a smoothing constraint to add temporal structure to the estimated time-specific effects and to regularize the exposure effects in the presence of multicollinearity across the measurements. Constraints include polynomials, splines, and Gaussian processes (Zanobetti et al., 2000; Warren et al., 2012). Notably, Mork and Wilson (2023) introduced an approach for constrained DLM using regression tree structures based on the Bayesian additive regression tree (BART) framework (Chipman et al., 2010). More recent methods have extended the tree structured DLM framework in several directions, including nonlinear exposure-response relationships (Mork and Wilson, 2022), models with lagged interactions among multiple exposures (Mork and Wilson, 2023), and models with heterogeneous lag effects (Mork et al., 2024).

There are several related methods and packages for DLM implementation. Table 1 lists currently available R packages for DLMs that are generally appropriate for the type of epidemiology studies considered in this paper. The *dlm* package contains software for estimating a DLM with linear or nonlinear exposure-response relationships using splines for

the smoothing constraint (Gasparrini, 2011). Mork and Wilson (2022, 2023) compare model performance between the spline-based and tree structured DLM implementations. The **dlm** package does not consider heterogeneity or multiple exposures. There are several DLM methods for the linear effect of a single exposure with modification by a single covariate. The **dlim** package allows for modification by a single continuous factor (Demateis et al., 2024) and the **bdlim** package allows for modification by a single categorical factor (Wilson et al., 2017b). However, these packages do not allow for multiple candidate modifying factors or multiple exposures. The package **DiscreteDLM** extends Bayesian estimation to categorical outcomes but does not consider mixtures or heterogeneity (Dempsey and Wyse, 2025). Hence, the **dlmtree** package fills several gaps including heterogeneous effects of multiple candidate modifiers and analysis of mixture or multivariate lagged exposures.

Table 1: Available DLM related R packages with functionalities

| Package | GLM | Nonlinearity | Mixture | Heterogeneity |
|--------------------|-----|--------------|---------|---------------|
| bdlim | ✓ | | | ✓ |
| DiscreteDLM | ✓ | | | |
| dlim | ✓ | | | ✓ |
| dlmtree | ✓ | ✓ | ✓ | ✓ |
| dlm | ✓ | ✓ | | |

Note: This table excludes packages designed for autoregressive DLMs, which are used in a different context from the models proposed here.

In this article, we introduce a comprehensive R package **dlmtree** which consolidates a wide range of tree structured DLMs. The package offers a user-friendly and computationally efficient environment to fit tree structured DLMs and to address multiple potential research assumptions including nonlinearity, monotonicity and prior information, multiple simultaneous exposures, and heterogeneous lag effects. We first provide a conceptual review of a regression tree as a smoothing constraint in a DLM framework and an overview of the extensions of tree structured DLMs. We present a decision tree to help users select an appropriate model for their analysis. We illustrate the model fitting process with detailed descriptions and syntax of functions for implementing models, obtaining the summary output and inferential information, and plotting the fitted models for visualization. The package is available in the comprehensive R archive network (CRAN), and the installation instructions are provided at <https://danielmork.github.io/dlmtree/>.

2 Tree structured DLMs

2.1 DLM tree as a smoothing constraint

In this section, we review the regression tree approach to constrained DLM estimation, a key idea underlying the tree structured DLMs in the **dlmtree** package. Classically, the DLM model is applied to time-series studies, and time is defined in terms of the outcome time. Let y_t be the observed outcome and x_t the observed exposure at time t for $t = 1, \dots, T$. For clarity, we assume a continuous outcome when presenting the models. We discuss extensions to generalized linear models in Section 2.4. The DLM applied to time-series is

$$y_t = \sum_{l=1}^L x_{t-l} \beta_l + \mathbf{z}_t' \gamma + \varepsilon_t, \quad (1)$$

where β_l is the linear effect of exposure at lag l (l days prior to the outcome assessment), \mathbf{z}_t is a vector of covariates including the intercept, γ is a vector of regression coefficients for the covariates, ε_t is independent error assumed to follow $\text{Normal}(0, \sigma^2)$.

In this paper, we focus on an alternative and more general representation of the DLM that does not assume repeated assessments of an outcome in a time-series design. Consider a vector of outcomes $\mathbf{y} = (y_1, \dots, y_n)$ for a sample $i = 1, \dots, n$. Suppose we are interested in

the lagged association between the outcome and a single longitudinally assessed exposure $x_i = (x_{i1}, \dots, x_{iT})$ measured at equally spaced time points, $t = 1, \dots, T$. Here, exposure time is not explicitly defined relative to the outcome. Most commonly, the exposures are in the T time points prior to outcome assessment, as in (1), but that is not required. In the illustrations in this manuscript, we consider exposure during the first T weeks of pregnancy and a birth or early childhood health outcome as our motivating example. Using this representation, the DLM is

$$y_i = f(x_i) + z_i' \gamma + \varepsilon_i, \quad f(x_i) = \sum_{t=1}^T x_{it} \theta_t, \quad (2)$$

where z_i , γ and ε_i are as defined above, and f is a distributed lag function parameterized by θ_t representing the linear effect of exposure at time t . The time-series design can also be modeled using this format, and we discuss the data processing needed to convert time-series data to this format in Section 4.2.

The estimation of θ_t for $t = 1, \dots, T$ in (2) requires an appropriate temporal structure, such as a smooth or piecewise-smooth constraint on θ_t , to account for autocorrelation within the exposure measurements. Mork and Wilson (2023) introduced a regression tree-structure, shown in Figure 1, referred to as a *DLM tree*. A DLM tree, denoted \mathcal{T} , is a binary tree that

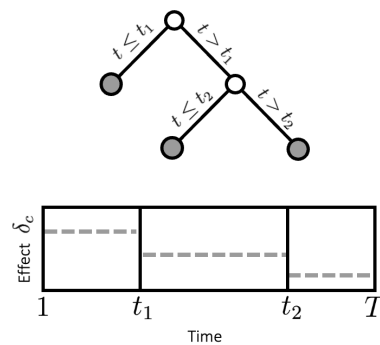


Figure 1: A DLM tree, \mathcal{T} . A binary tree splits the time span into non-overlapping intervals, here resulting in three terminal nodes representing three time segments (gray nodes). Each terminal node is assigned a constant effect (gray dashed lines).

splits the exposure time span into non-overlapping segments. We denote the terminal nodes of \mathcal{T} as λ_c for $c = 1, \dots, C$ where C is the total number of terminal nodes. Each terminal node of the DLM tree is assigned a scalar parameter δ_c that represents the effect of the time lags contained in that terminal node. We denote a set of scalar parameters of the DLM tree \mathcal{T} as $\mathcal{D} = \{\delta_1, \dots, \delta_C\}$. A DLM tree defines a function of a time lag,

$$g(t|\mathcal{T}, \mathcal{D}) = \delta_c \quad \text{if } t \in \lambda_c. \quad (3)$$

The treed distributed lag model (TDLM) is a tree structured DLM in its simplest form and is a Bayesian additive model that consists of an ensemble of A DLM trees, indexed as \mathcal{T}_a , with a corresponding set of scalar parameters \mathcal{D}_a for $a = 1, \dots, A$. TDLM defines θ_t in (2) as

$$\theta_t = \sum_{a=1}^A g(t|\mathcal{T}_a, \mathcal{D}_a). \quad (4)$$

The representation of the DLM tree ensemble provides several advantages. Each DLM tree provides a temporal structure on the exposure-time-response function with data-driven learning of the change points and time spans related to the outcome. The ensemble structure allows for flexibility to approximate smoothness in the exposure-time-response function as each DLM tree in the ensemble splits the time span differently.

2.2 DLM tree pair for lagged multivariate exposures

A distributed lag mixture model extends the DLM framework to a multivariate exposure (also referred to as *mixture exposures* in the environmental literature) assessed longitudinally. For mixture exposures with $M \geq 2$ exposures, we denote the vector of longitudinally assessed measurements of the m^{th} exposure, for $m = 1, \dots, M$, as $\mathbf{x}_{im} = (x_{im1}, \dots, x_{imT})$. The exposure-time-response function $f(x_i)$ in (2) is replaced with a mixture-exposure-time-response function that incorporates the main effect of each exposure and the pairwise lagged interaction effects between exposures. The mixture-exposure-time-response function is

$$f(x_{i1}, \dots, x_{iM}) = \sum_{m=1}^M \sum_{t=1}^T x_{imt} \theta_{mt} + \sum_{m_1=1}^M \sum_{m_2=m_1+1}^M \sum_{t_1=1}^T \sum_{t_2=1}^T x_{im_1 t_1} x_{im_2 t_2} \theta_{m_1 m_2 t_1 t_2}, \quad (5)$$

where θ_{mt} is a main effect of exposure m at time t and $\theta_{m_1 m_2 t_1 t_2}$ is an interaction effect of exposure m_1 at time t_1 and exposure m_2 at time t_2 .

Estimating the function in (5) is challenging due to autocorrelation across repeated exposure measurements, correlation at the same time lag between exposures, and a high-dimensional parameter space. [Mork and Wilson \(2023\)](#) introduced the treed distributed lag mixture model (TDLMM) that structures the parameters in (5) using a *DLM tree pair*. Figure 2 illustrates a single DLM tree pair, denoted $\{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_1 \times \mathcal{T}_2\}$ with corresponding parameters sets $\{\mathcal{D}_1, \mathcal{D}_2, \Omega\}$, where \mathcal{D}_1 and \mathcal{D}_2 represent the scalar main effects for \mathcal{T}_1 and \mathcal{T}_2 , respectively, and Ω contains the scalar interaction effects for interaction surface, denoted $\mathcal{T}_1 \times \mathcal{T}_2$. Each DLM tree in the pair is associated with one component of the mixture

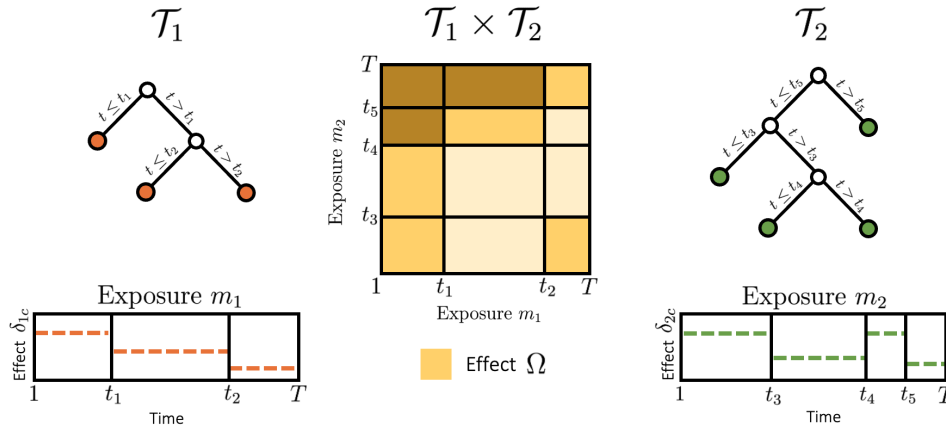


Figure 2: A DLM tree pair. Two DLM trees split the time span of the assigned exposure into non-overlapping intervals, here resulting in three time segments for exposure m_1 and four time segments for exposure m_2 (colored nodes). Each terminal node is assigned a scalar parameter that represents a constant effect of the assigned exposure (colored dashed lines). The interaction surface is fully defined by two DLM trees and each combination of time segments is assigned a scalar parameter (color-shaded boxes).

exposures. Similar to the DLM tree described above, each DLM tree in the pair partitions the time span of its assigned component. Each DLM tree in a tree pair is defined similarly to (3) such that

$$g(t|\mathcal{T}_p, \mathcal{D}_p) = \delta_{pc} \quad \text{if } t \in \lambda_{pc}, \quad p = 1, 2. \quad (6)$$

The DLM tree pair structure allows for an interaction surface to model lagged interactions between exposures. The interaction surface $\mathcal{T}_1 \times \mathcal{T}_2$ shown in the middle of Figure 2, is fully defined by the two DLM trees in a pair. Each time interval of the first DLM tree is paired with every interval of the second DLM tree. Each combination is assigned a scalar parameter $\omega_{c_1 c_2}$ that represents the lagged interaction effect where c_1 and c_2 are indices for terminal nodes of the first and second DLM tree, respectively. The interaction surface of a DLM tree

pair defines a function

$$g_I(t_1, t_2 | \mathcal{T}_1 \times \mathcal{T}_2, \Omega) = \omega_{c_1 c_2} \quad \text{if } t_1 \in \lambda_{c_1}, t_2 \in \lambda_{c_2}. \quad (7)$$

A DLM tree pair with an interaction surface provides a structure to regularize the exposure-time-response function for two components and their lagged interaction effects.

TDLMM employs an ensemble representation of A DLM tree pairs denoted $\{\mathcal{T}_{a1}, \mathcal{T}_{a2}, \mathcal{T}_{a1} \times \mathcal{T}_{a2}\}$ with the corresponding set of scalar parameters $\{\mathcal{D}_{a1}, \mathcal{D}_{a2}, \Omega_a\}$, using the formulation in (6) and (7). Each DLM tree \mathcal{T}_{ap} is also associated with exposure m , indicated by $S_{ap} = m$. With TDLMM, the main effect of exposure m at time t in (5) is

$$\theta_{mt} = \sum_{a=1}^A \sum_{p=1}^2 g(t | \mathcal{T}_{ap}, \mathcal{D}_{ap}) \mathbb{I}(S_{ap} = m). \quad (8)$$

The pairwise interaction effect between exposure m_1 at time t_1 and exposure m_2 at time t_2 in (5) is

$$\theta_{m_1 m_2 t_1 t_2} = \sum_{a=1}^A g_I(t_1, t_2 | \mathcal{T}_{a1} \times \mathcal{T}_{a2}, \Omega_a) \mathbb{I}(S_{a1} = m_1, S_{a2} = m_2). \quad (9)$$

TDLMM has three representations depending on different assumptions of lagged interactions between exposures. The simplest form, TDLMMadd, assumes $\theta_{m_1 m_2 t_1 t_2} = 0$ in (5), implying no interaction between exposures, resulting in an additive model of the main effects of exposures. The second form, TDLMMns, accounts for lagged interaction between exposures but not within exposures. The last, TDLMMall, allows for all lagged interactions between and within exposures, implying a nonlinear effect of exposure.

2.3 Extensions to nonlinear exposure-time-response functions

Mork and Wilson (2022) introduced the treed distributed lag nonlinear model (TDLNM) to estimate the nonlinear association between a single longitudinally assessed exposure and an outcome. The DLM tree, as shown in (3), assumes a linear association between exposure at each time point and the outcome. To relax the linearity assumption for a distributed lag nonlinear model framework, TDLNM modifies the DLM tree to additionally split exposure concentration levels along with the exposure time span, partitioning the bi-dimensional space of exposure concentration and time lags. Mork and Wilson (2024) further extended TDLNM to include a monotonicity assumption, where the exposure-response is constrained to be non-decreasing at each time point. See Mork and Wilson (2022, 2024) for details.

2.4 Extensions to generalized linear models

In various applications of DLMs, the response variables may be binary or counts. Examples of binary outcomes include the occurrence of conditions such as asthma or preterm birth, and an example of a count-valued outcome is the daily number of deaths in a county. The linear representation of the DLM framework allows the tree structured DLMs to extend to the generalized linear model setting. TDLM, TDLNM, and TDLMM have been extended to incorporate binary response variables via logistic regression (Mork and Wilson, 2022, 2023). Further extensions on these models allow for count data via negative binomial regression, including an option for zero-inflated negative binomial data. The extensions to binary and count data rely on a framework based on the Pólya-Gamma data augmentation approach (Polson et al., 2013; Neelon, 2019).

2.5 Extensions to heterogeneous models

Another extension to the DLM framework is to assume heterogeneous exposure effects. The exposure effects may be heterogeneous due to a single modifying factor or a set of factors. For example, the impact of prenatal exposure to air pollution may be governed by genetic

factors such as fetal sex (Rosa et al., 2019). The set of factors, referred to as *modifiers*, may be continuous, categorical, or ordinal. The general approach is to introduce an additional tree, known as a *modifier tree*, that partitions the modifier space and has a DLM tree or a DLM tree pair affixed to each terminal node. Mork et al. (2024) extended TDLM to the

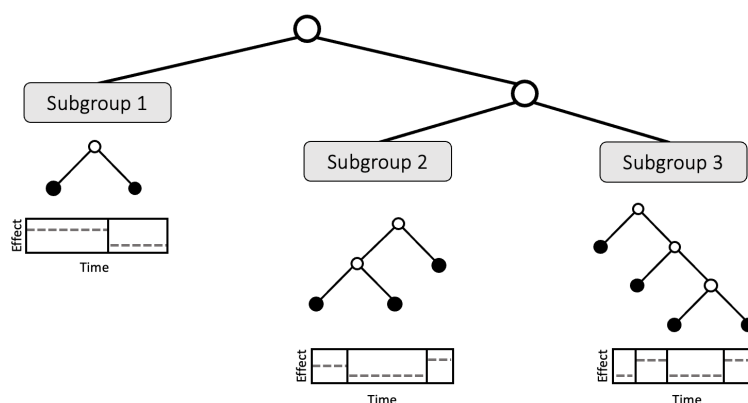


Figure 3: A nested tree structure for heterogeneous tree structured DLMs. The top tree is a modifier tree that is applied to candidate modifiers, here resulting in three subgroups. DLM trees are affixed to the terminal nodes of a modifier tree to estimate the exposure-time-response relationship specific to the subgroups.

heterogeneous distributed lag model (HDLM) by introducing a nested tree structure, as shown in Figure 3. In a nested tree, a modifier tree is applied to a set of candidate modifiers, defining mutually exclusive subgroups of the sample at each terminal node. A DLM tree is then attached to each terminal node of the modifier tree to define subgroup-specific effects with unique parameters for each subgroup. Other extensions include a heterogeneous distributed lag mixture model (HDLMM) that extends TDLM to incorporate heterogeneity based on a set of multiple candidate modifiers using an ensemble of tree triplet structures.

3 Implementation

The tree structured DLMs are classified with three main criteria: 1) linear or nonlinear exposure-time-response function, 2) one lagged component or a mixture of more than one lagged component, and 3) homogeneous or heterogeneous exposure-time-response relationship. Figure 4 illustrates a decision tree as a guide to choosing an appropriate tree structured DLM.

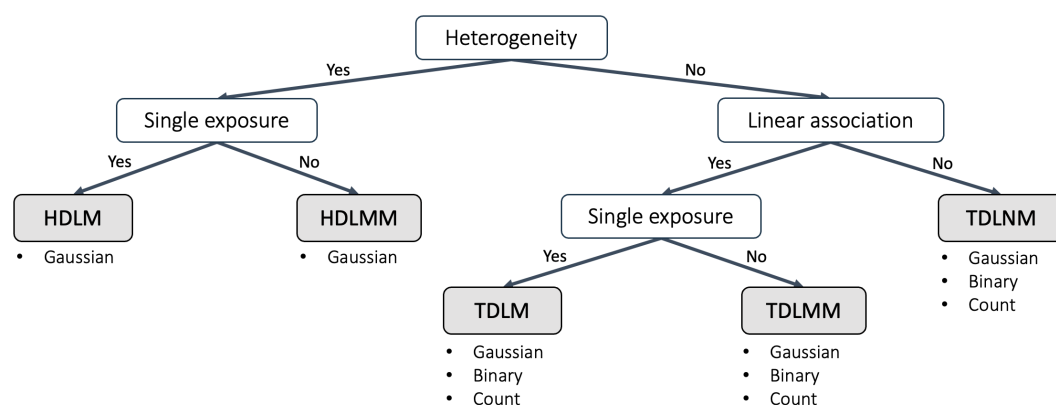


Figure 4: A decision tree for choosing tree structured DLMs. The bullet points below the models list the data types of response variables that each model can incorporate.

The `dmltree` package offers tree structured DLMs shown in Figure 4. A main function `dmltree` is designed to fit various tree structured DLMs, offering customizable analysis

through its arguments. The function `dmltree` with its main arguments is as follows.

```
dmltree(formula, data, exposure.data, family, dlm.type, mixture, het)
```

The function requires a formula specifying an outcome and covariates for the fixed effect, a data source, the data type of response variable, and three arguments for model specification. The data for the formula, including covariates and the outcome, are provided as an $(n \times p)$ data frame in the `data` argument. The exposures are not specified in the formula and are not needed in the `data` argument. Rather, the exposure data is specified separately as an $(n \times T)$ matrix of exposure measurements or a list of $(n \times T)$ matrices of exposure measurements in the `exposure.data` argument.

Table 2: Arguments for `dmltree` function

| Argument | Description |
|----------------------------|--|
| <code>formula</code> | Object of class <code>formula</code> for the fixed effect |
| <code>data</code> | A data frame containing covariates and an outcome used in <code>formula</code> |
| <code>exposure.data</code> | A numerical matrix of exposure data with the same length as <code>data</code> . For a mixture setting, a named list containing equally sized numerical matrices of exposure data having the same length as the <code>data</code> |
| <code>family</code> | 'gaussian' for a continuous response, 'logit' for binomial, 'zinb' for count data |
| <code>dlm.type</code> | DLM type specification: 'linear', 'nonlinear', 'monotone' |
| <code>mixture</code> | logical; A flag for mixture exposures if TRUE |
| <code>het</code> | logical; A flag for heterogeneity if TRUE |

Table 2 provides descriptions of the arguments. All tree structured DLMs allow for continuous response variables, while TDLM, TDLNM, and TDLMM additionally allow for binary and count-valued response variables. Additional parameters, omitted here, include MCMC sampling parameters, the number of trees in the ensemble, effect shrinkage, sparsity parameters for exposure and modifier selection, and model-specific hyperparameters. These parameters can be fine-tuned and passed as a list to the corresponding control helper functions, each suffixed with its purpose (e.g., `control.mcmc`, `control.hyper`).

A fitted model is assigned a class, determined by the model-specifying arguments. The classes are: `tdlm`, `tdlmm`, `tdlnm`, `hdlm`, and `hdlmm`. Each class uses an S3 object oriented system with summary method. The summary method returns the model information, estimates and credible intervals for the fixed effects and lag effects, and time points of a significant effect, where the 95% credible intervals of the lag effects do not contain zero (in some applications this is referred to as *critical window*) of the exposure or mixture exposures. The `plot` method on the summary object further returns the visualization of the estimated main exposure effects (and interaction effects if applicable) with 95% credible intervals. Additionally for classes of models with heterogeneity: `hdlm` and `hdlmm`, the `shiny` method is built for various types of statistical inference. The [shiny](#) app interface provides tools for identifying important modifiers and their splitting points that contribute to heterogeneity. It also includes features for evaluating personalized exposure effects with a set of user-specified modifiers, and exposure effects specific to a subgroup defined by a set of modifiers of interest.

4 Example usage

We illustrate the example usage of tree structured DLMs through a set of vignettes based on simulated data. We demonstrate data preparation and the model fitting process for TDLM, TDLMM, HDLM, and HDLMM. An example of fitting TDLNM is provided in the supplementary material.

4.1 Simulated dataset

We use the simulated dataset 'sbd_dlmtree', which is publicly available in the [dlmtree](#) package GitHub repository. The dataset contains 10,000 simulated mother-child dyads with descriptions of maternal and birth information. The maternal covariates include maternal age, height, prior weight, prior body mass index (BMI), race, Hispanic designation, education attainment, smoking habits, marital status, and yearly income. Additionally, the birth information includes birth weight for gestational age z-scores (BWGAZ), gestational age, sex of a child, and estimated date of conception. The dataset contains five environmental chemicals measured at 37 weeks preceding the birth for each dyad: fine particulate matter (PM_{2.5}), temperature, sulfur dioxide (SO₂), carbon monoxide (CO), and nitrogen dioxide (NO₂). Each exposure measurement is scaled by its interquartile range value. The dataset is constructed to have realistic distributions and correlations of the covariates and exposures. It contains a complex exposure-response relationship that includes interactions and heterogeneous effects.

In the following examples, we examine the distributed lag effects of maternal exposure to the environmental mixtures during the 37 weeks of gestation on BWGAZ. The results provided in the example usage are solely for demonstrative purposes of the model fitting process using the model parameters and simulated data and do not represent any actual findings.

4.2 Data preparation for model fitting

We first load the required packages: [dlmtree](#) and [dplyr](#). We used [dplyr](#) for a better presentation of the data processing. We set the seed for reproducibility and load the external dataset 'sbd_dlmtree' from [dlmtree](#) package repository using an embedded function `get_sbd_dlmtree`.

```
# Libraries and seed
library(dlmtree)
library(dplyr)
set.seed(1)

# Download data as 'sbd'
sbd <- get_sbd_dlmtree()
```

The data frame `sbd` has a lagged format where each row has columns of lagged measurements of each exposure (e.g. wide format data). Often, datasets have a time-series format, which contains a single column of dates and multiple columns of corresponding measurements of response variables and exposures on that specific date only. This is particularly common in time-series studies, whereas the wide format data is more common in cohort studies. The model fitting function `dlmtree` is designed to use a data frame in a wide format; hence, a data frame with a time-series format must be pivoted to a wide format. An example data frame of time-series format and a function for data pivoting are provided in the supplementary materials. In addition to the wide format, the software in this package requires that all individuals and all exposures (for multi-exposure models) have the same number of exposure time points.

To prepare the birth data in `sbd` for model fitting, we create a data frame including the covariates and the response variable, BWGAZ. We note that the categorical columns in the dataset are of class `factor`. We consider five components for exposure data: PM_{2.5}, temperature, SO₂, CO, and NO₂, and store them as a list of exposure matrices with a size of (10,000 × 37). Each matrix is already in wide format, where rows represent observations and columns represent exposure measurements at different lags.

```
# Response and covariates
sbd_cov <- sbd %>% select(bwgaz, ChildSex, MomAge, GestAge, MomPriorBMI, Race,
```



```

      Hispanic, MomEdu, SmkAny, Marital, Income,
      EstDateConcept, EstMonthConcept, EstYearConcept)

# Exposure data
sbd_exp <- list(PM25 = sbd %>% select(starts_with("pm25_")),
               TEMP = sbd %>% select(starts_with("temp_")),
               SO2 = sbd %>% select(starts_with("so2_")),
               CO = sbd %>% select(starts_with("co_")),
               NO2 = sbd %>% select(starts_with("no2_")))

sbd_exp <- sbd_exp %>% lapply(as.matrix)

```

Each matrix of the exposure data list can be centered and scaled with caution to different interpretations of the resulting estimates. Specifically, when using the TDLMM with lagged interaction, it is crucial to avoid centering the exposure data as it can lead to an inaccurate estimate of the marginal exposure effect when considering co-exposures.

4.3 TDLM: Estimating linear relationship between an outcome and a single exposure

Model fitting and summary

We first assume that we are interested in the linear association between BWGAZ and weekly exposure to PM_{2.5} during the first 37 gestational weeks. We include the following covariates to control for fixed effects: child sex, maternal age, BMI, race, Hispanic designation, smoking habits, and month of conception. We fit TDLM with the following code.

```

tdlm.fit <- dlmtree(
  formula = bwgaz ~ ChildSex + MomAge + MomPriorBMI + Race +
              Hispanic + SmkAny + EstMonthConcept,
  data = sbd_cov,
  exposure.data = sbd_exp[["PM25"]], # A single numeric matrix
  family = "gaussian",
  dlm.type = "linear",
  control.mcmc = list(n.burn = 2500, n.iter = 10000, n.thin = 5)
)

```

The resulting fitted object of class `tdlm` has attributes of the fitted model information and posterior samples of parameters of interest. The summary method applied to the object `tdlm.fit` returns a clear overview of the model fit. The summary of the model is obtained with the following code.

```

tdlm.sum <- summary(tdlm.fit)
print(tdlm.sum)

```

```
---
```

```
TDLM summary
```

```
Model run info:
```

```

- bwgaz ~ ChildSex + MomAge + MomPriorBMI + Race + Hispanic + SmkAny + EstMonthConcept
- sample size: 10,000
- family: gaussian
- 20 trees
- 2500 burn-in iterations
- 10000 post-burn iterations
- 5 thinning factor
- exposure measured at 37 time points

```

- 0.95 confidence level

Fixed effect coefficients:

| | Mean | Lower | Upper |
|----------------------|--------|--------|--------|
| *(Intercept) | 2.289 | 2.032 | 2.542 |
| *ChildSexM | -2.105 | -2.126 | -2.085 |
| MomAge | 0.000 | -0.001 | 0.002 |
| *MomPriorBMI | -0.021 | -0.022 | -0.019 |
| RaceAsianPI | 0.069 | -0.057 | 0.192 |
| RaceBlack | 0.078 | -0.050 | 0.205 |
| Racewhite | 0.059 | -0.060 | 0.181 |
| *HispanicNonHispanic | 0.255 | 0.233 | 0.278 |
| *SmkAnyY | -0.403 | -0.451 | -0.356 |
| EstMonthConcept2 | -0.049 | -0.109 | 0.010 |
| *EstMonthConcept3 | -0.145 | -0.211 | -0.077 |
| *EstMonthConcept4 | -0.230 | -0.295 | -0.160 |
| *EstMonthConcept5 | -0.207 | -0.265 | -0.147 |
| *EstMonthConcept6 | -0.205 | -0.260 | -0.153 |
| EstMonthConcept7 | -0.032 | -0.083 | 0.023 |
| *EstMonthConcept8 | 0.145 | 0.081 | 0.210 |
| *EstMonthConcept9 | 0.393 | 0.326 | 0.460 |
| *EstMonthConcept10 | 0.372 | 0.311 | 0.437 |
| *EstMonthConcept11 | 0.330 | 0.271 | 0.387 |
| *EstMonthConcept12 | 0.129 | 0.078 | 0.181 |

* = CI does not contain zero

DLM effect:

range = [-0.019, 0.008]

signal-to-noise = 0.021

critical windows: 11-20,36-37

| | Mean | Lower | Upper |
|------------|--------|--------|--------|
| Period 1 | 0.003 | -0.005 | 0.015 |
| Period 2 | 0.000 | -0.006 | 0.010 |
| Period 3 | -0.002 | -0.010 | 0.004 |
| Period 4 | -0.002 | -0.010 | 0.003 |
| Period 5 | -0.001 | -0.007 | 0.004 |
| Period 6 | -0.001 | -0.006 | 0.005 |
| Period 7 | -0.001 | -0.006 | 0.006 |
| Period 8 | -0.001 | -0.008 | 0.004 |
| Period 9 | -0.002 | -0.011 | 0.003 |
| Period 10 | -0.002 | -0.012 | 0.006 |
| *Period 11 | -0.016 | -0.024 | -0.007 |
| *Period 12 | -0.017 | -0.024 | -0.010 |
| *Period 13 | -0.017 | -0.024 | -0.012 |
| *Period 14 | -0.017 | -0.022 | -0.010 |
| *Period 15 | -0.017 | -0.022 | -0.011 |
| *Period 16 | -0.017 | -0.022 | -0.011 |
| *Period 17 | -0.017 | -0.024 | -0.011 |
| *Period 18 | -0.019 | -0.030 | -0.013 |
| *Period 19 | -0.018 | -0.027 | -0.011 |
| *Period 20 | -0.015 | -0.024 | -0.003 |
| Period 21 | -0.007 | -0.019 | 0.002 |
| Period 22 | -0.002 | -0.010 | 0.006 |
| Period 23 | -0.003 | -0.012 | 0.003 |
| Period 24 | -0.002 | -0.008 | 0.004 |

```

Period 25  0.000 -0.005  0.006
Period 26  0.000 -0.005  0.006
Period 27 -0.001 -0.005  0.005
Period 28 -0.001 -0.006  0.004
Period 29 -0.001 -0.007  0.004
Period 30 -0.002 -0.008  0.003
Period 31 -0.002 -0.008  0.004
Period 32 -0.001 -0.008  0.005
Period 33  0.002 -0.004  0.010
Period 34  0.004 -0.003  0.012
Period 35  0.006 -0.001  0.014
*Period 36  0.008  0.000  0.017
*Period 37  0.008  0.000  0.019

```

```
---
```

```
* = CI does not contain zero
```

```
residual standard errors: 0.004
```

```
---
```

The summary output first presents a section ‘Model run info’ with the model fitting information including the formula, sample size, data type of the response variable, number of trees in the ensemble, MCMC parameters, number of lags, and a confidence level. The next section, ‘Fixed effect coefficients’, shows the estimates with credible intervals for the regression coefficients of the covariates. Lastly, the summary output returns ‘DLM effect’ section including the range of DLM effects, signal-to-noise ratio, and estimated lagged effects with credible intervals. Each lag is marked with an asterisk if it is identified as a critical window based on a pointwise 0.95 probability credible interval. In context, TDLM estimated a negative association between BWGAZ and PM_{2.5} with gestational weeks 11–20 as critical windows.

The cumulative effect of the exposure is defined as the effect of a one unit increment of the exposure across all time points. The following code returns the estimated cumulative effect with its 95% credible interval from an attribute `cumulative.effect` of the summary object `tdlm.sum`.

```

tdlm.sum$cumulative.effect
      mean      2.5%      97.5%
-0.1738753 -0.2124918 -0.1367665

```

In this context, TDLM estimates that a unit increment of exposure to PM_{2.5} across all gestational weeks has a cumulative exposure effect of -0.17 (95% CrI: [-0.21, -0.14]) on BWGAZ.

Visualizing exposure effects

For a more intuitive view of the overall trend of the distributed lag effects, the `plot` method may be applied to the summary object `tdlm.sum` to visualize the effect estimates as the following.

```
plot(tdlm.sum, main = "Estimated effect of PM2.5", xlab = "Time", ylab = "Effect")
```

Figure 5 shows the plot of the estimated exposure effect of PM_{2.5} in the simulated dataset where the x-axis is the lags (or weeks) and the y-axis is the estimated exposure effect. The gray area showing the 95% credible intervals of exposure effects during weeks 11–20 does not cover the red line of a null effect, which indicates a critical window. The plot method also includes additional arguments of `main`, `xlab`, and `ylab` for customizing the main title, x-axis, and y-axis label. For a customized plot, the estimated values can be obtained from the attributes of the summary object `tdlm.sum`: `matfit`, `cilower`, and `ciupper`.

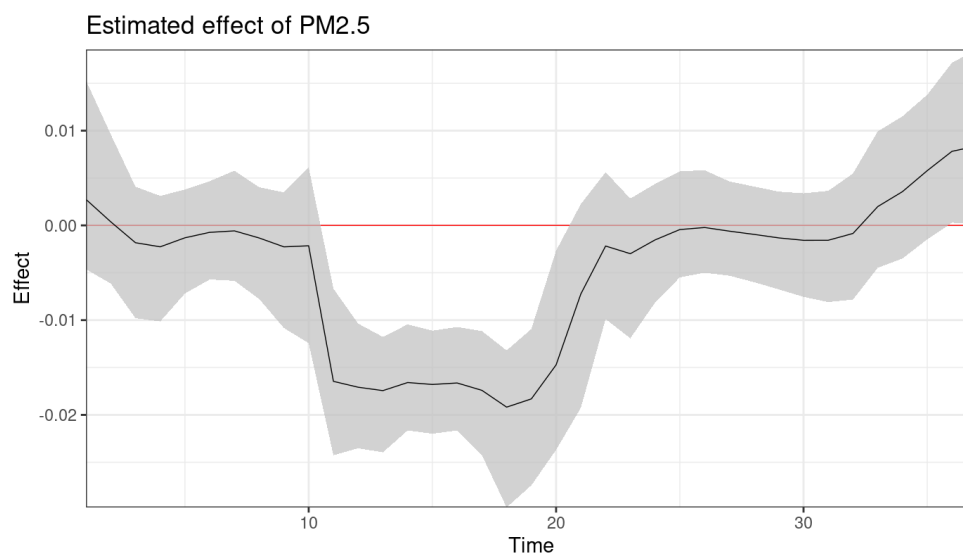


Figure 5: Estimated distributed lag effects of $PM_{2.5}$ on BWGAZ during 37 gestational weeks, using TDLM. These results are based on simulated data.

4.4 TDLMM: Analyzing linear relationship between an outcome and multiple exposures

Model fitting with pairwise lagged interactions

Suppose we are interested in the linear association between BWGAZ and mixture exposures of five exposures: $PM_{2.5}$, temperature, SO_2 , CO, and NO_2 . We are mainly interested in the marginal effects of each exposure on BWGAZ, the pairwise lagged interactions between exposures, and which exposures are most correlated with BWGAZ. Here, we use all five exposures to illustrate the flexibility and capability of our package to handle multiple exposures. In practice, users should carefully determine the number of exposures included in the model by considering the complexity of the model relative to the number of observations and the expected signal-to-noise ratio. The key differences between the code for the model for multiple exposures below and the code for the previous single exposure model are that the `exposure.data` is provided with a list of matrices of five different exposures and we specify `mixture = TRUE`. The code is as follows.

```
tdlmm.fit <- dlmtree(
  formula = bwgaz ~ ChildSex + MomAge + MomPriorBMI +
    Race + Hispanic + SmkAny + EstMonthConcept,
  data = sbd_cov,
  exposure.data = sbd_exp,
  family = "gaussian",
  dlm.type = "linear",
  mixture = TRUE,
  control.mix = list(interactions = "noself"),
  control.mcmc = list(n.burn = 2500, n.iter = 10000, n.thin = 5)
)
```

The model assumption regarding lagged interaction can be additionally specified for the TDLMM fitting with interactions argument within `control.mix`. The options include no interaction ('none'), no-self interactions ('noself'), and all interactions ('all'). We use TDLMM with no-self lagged interaction, which is the default argument.

Model summary accounting for co-exposures

The summary method can be applied to the fitted object `tdlmm.fit`. As TDLMM allows for pairwise lagged interaction between mixture exposures, the estimated exposure effect for each exposure varies with the levels of the co-exposures. The summary method on class `tdlmm` offers additional control argument `marginalize` to address this. The argument `marginalize` requires a fixed level used for co-exposure marginalization. The default is the empirical means of co-exposures, which provides the distributed lag function for a single exposure estimated when all other exposures are fixed at their means. This is equivalent to integrating out all co-exposures. The following code returns the summary of the `tdlmm.fit` with marginalization using the empirical means of co-exposures.

```
# Marginalization with co-exposure fixed at the empirical means
tdlmm.sum <- summary(tdlmm.fit, marginalize = "mean")
```

Other marginalization options are available for conducting different types of inferences. The second option is to specify a number between 0 and 100, representing a percentile of the co-exposures that will be used for marginalization. The following code returns the summary of the `tdlmm.fit` with marginalization fixing all co-exposures at their 25th percentile values.

```
# Marginalization with co-exposure fixed at 25th percentile
tdlmm.sum.percentile <- summary(tdlmm.fit, marginalize = 25)
```

The last option is to specify the exact levels of co-exposures. This option requires the argument `marginalize` to be a numeric vector of the same length as the number of exposures used for the model fitting. The specified values in the vector must also follow the order of the exposures in the fitted model. This method of marginalization offers flexibility as these exposure levels can be specified based on pre-informed levels using existing data or to address hypothetical questions. For example, the following code can be used to obtain the marginal exposure effect of $PM_{2.5}$ when temperature, SO_2 , CO, and NO_2 are all fixed to 1. The marginalized effects of other exposures are calculated in a similar manner.

```
# Marginalization with co-exposure fixed at exact levels for each exposure
tdlmm.sum.level <- summary(tdlmm.fit, marginalize = c(1, 1, 1, 1, 1))
```

Below is the summary result of `tdlmm.sum` with the default argument using empirical means.

```
print(tdlmm.sum)
```

```
---
```

```
TDLMM summary
```

```
Model run info:
```

```
- bwgaz ~ ChildSex + MomAge + MomPriorBMI + Race + Hispanic + SmkAny + EstMonthConcept
- sample size: 10,000
- family: gaussian
- 20 trees (alpha = 0.95, beta = 2)
- 2500 burn-in iterations
- 10000 post-burn iterations
- 5 thinning factor
- 5 exposures measured at 37 time points
- 10 two-way interactions (no-self interactions)
- 1 kappa sparsity prior
- 0.95 confidence level
```

```
Fixed effects:
```

| | Mean | Lower | Upper |
|--------------|--------|--------|--------|
| *(Intercept) | 0.172 | 0.043 | 0.307 |
| *ChildSexM | -2.063 | -2.085 | -2.041 |

| | | | |
|----------------------|--------|--------|--------|
| MomAge | 0.001 | -0.001 | 0.002 |
| *MomPriorBMI | -0.020 | -0.022 | -0.019 |
| RaceAsianPI | 0.027 | -0.058 | 0.117 |
| RaceBlack | 0.033 | -0.063 | 0.124 |
| Racewhite | 0.016 | -0.067 | 0.100 |
| *HispanicNonHispanic | 0.248 | 0.224 | 0.272 |
| *SmkAnyY | -0.393 | -0.441 | -0.346 |
| EstMonthConcept2 | 0.073 | -0.003 | 0.145 |
| *EstMonthConcept3 | 0.107 | 0.009 | 0.211 |
| *EstMonthConcept4 | 0.158 | 0.038 | 0.282 |
| *EstMonthConcept5 | 0.255 | 0.126 | 0.388 |
| *EstMonthConcept6 | 0.200 | 0.064 | 0.333 |
| *EstMonthConcept7 | 0.223 | 0.084 | 0.354 |
| *EstMonthConcept8 | 0.199 | 0.068 | 0.331 |
| *EstMonthConcept9 | 0.291 | 0.164 | 0.418 |
| *EstMonthConcept10 | 0.182 | 0.070 | 0.296 |
| *EstMonthConcept11 | 0.135 | 0.040 | 0.236 |
| EstMonthConcept12 | 0.006 | -0.062 | 0.077 |

* = CI does not contain zero

--

Exposure effects: critical windows

* = Exposure selected by Bayes Factor

(x.xx) = Relative effect size

*PM25 (0.7): 11-20

*TEMP (0.7): 5-19

*SO2 (0.21):

*CO (0.63):

*NO2 (0.26): 23

--

Interaction effects: critical windows

PM25/TEMP (0.8):

12/6-19

13/6-19

14/6-20

15/6-20

16/6-20

17/6-21

18/5-22

19/5-22

20/6-21

residual standard errors: 0.005

The summary output of the TDLMM fit contains similar information to that of the TDLM. The 'Model run info' section in the output includes additional information specific to mixture exposures: the number of exposures included in the model, the number of pairwise interactions, and a sparsity parameter for exposure selection. The summary output does not include the lagged effects for each exposure to prevent overwhelming the output with excessive information. The summary presents the critical window of the marginal effects of each exposure with the relative effect size, indicating the effect size of exposure relative to that of other exposures. The summary also returns the critical window of lagged interaction effects with relative effect size. In our context, all five exposures are considered to be

significantly associated with BWGAZ, based on a Bayes factor threshold of 0.5. The TDLMM estimated gestational weeks 11–20, 5–19, and 23 as critical windows of PM_{2.5}, temperature, and NO₂, respectively. The model fit also identified significant PM_{2.5}–temperature lagged interaction effects.

Additional statistical inferences using TDLMM

More useful statistical inferences are possible with `tdlmm.fit` and `tdlmm.sum`. First, a function `adj_coexposure` can be used to obtain the marginalized exposure effect while accounting for the expected change in co-exposures. In comparison to using the argument `marginalize` in `summary` method, the function `adj_coexposure` uses a spline-based method to predict the expected changes in co-exposures corresponding with a pre-defined change in an exposure of interest and calculates the marginalized effects of the primary exposure with co-exposure at the predicted levels. The following code shows an example usage of the function with its output omitted.

```
# Lower and upper exposure levels specified as 25th and 75th percentiles
tdlmm.coexp <- adj_coexposure(sbd_exp, tdlmm.fit, contrast_perc = c(0.25, 0.75))
```

The function requires exposure data, the model fit of class `tdlmm`, and an argument `contrast_perc` which can be specified with a vector of two percentiles used for co-exposure prediction. Another argument `contrast_exp` is available for specifying exact exposure levels for each exposure.

Second, the marginal cumulative effect of each exposure can be obtained with the summary object `tdlmm.sum`. The object has a list attribute `DLM`, which contains estimates of marginal exposure effect and cumulative effect with their credible intervals. These estimated effects correspond to the argument `marginalize` specified within `summary` method. The code below returns the estimates of the cumulative effect of PM_{2.5}.

```
tdlmm.sum$DLM$PM25$cumulative
$mean
[1] -0.3790024

$ci.lower
      2.5%
-0.5318544

$ci.upper
      97.5%
-0.2305391
```

PM_{2.5} can be replaced with other exposures if desired, e.g., `tdlmm.sumDLMTEMP$cumulative` for temperature. In this context, the TDLMM estimates that the marginal cumulative effect of a unit increase of PM_{2.5} across all gestational weeks is -0.38 (95% CrI: [-0.53, -0.23]).

Visualizing main exposure effects and lagged interaction effects

The `plot` method on the summary object `tdlmm.sum` requires a single exposure or a pair of exposures. The `plot` method returns the marginal effect of an exposure when specified with a single exposure. For instance, for the three exposures, PM_{2.5}, temperature, and NO₂, we can use the following code.

```
library(gridExtra)

p1 <- plot(tdlmm.sum, exposure1 = "PM25", main = "PM2.5")
```

```
p2 <- plot(tdlmm.sum, exposure1 = "TEMP", main = "Temperature")
p3 <- plot(tdlmm.sum, exposure1 = "NO2", main = "NO2")

grid.arrange(p1, p2, p3, nrow = 1)
```

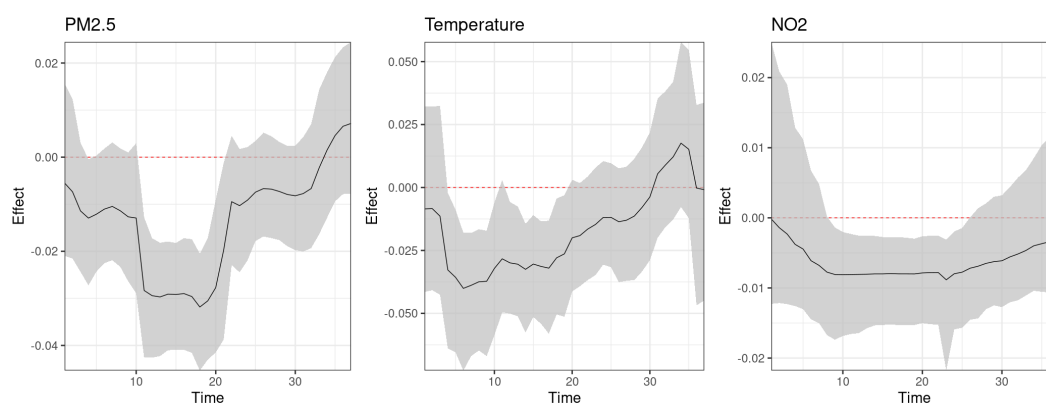


Figure 6: Estimated marginal distributed lag effects of PM_{2.5}, temperature, and NO₂ on BWGAZ during 37 gestational weeks, using TDLMM. These results are based on simulated data.

The plot method with arguments of two exposures visualizes an interaction surface of two specified exposures. The following code plots an estimated pairwise interaction surface of two specified exposures:

```
plot(tdlmm.sum, exposure1 = "PM25", exposure2 = "TEMP")
```

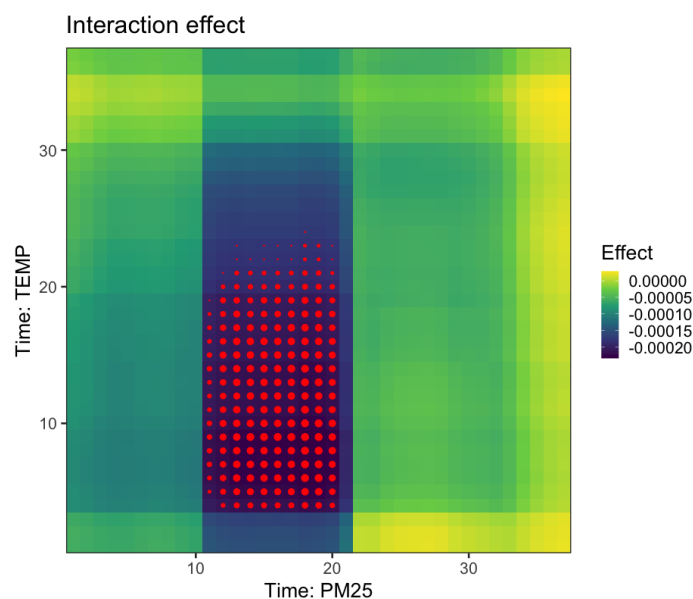


Figure 7: Estimated lagged interaction effects between PM_{2.5} and temperature, using TDLMM. These results are based on simulated data.

Figure 7 shows the estimated interaction surface between PM_{2.5} and temperature in the simulated data, estimated with TDLMM. The gradient colors on the grid indicate the estimated interaction effects where the x-axis is the exposure time span of PM_{2.5} and the y-axis is that of temperature. The red dots indicate that the credible interval of the effect does not contain zero, with larger dots indicating a higher probability of non-null effect. Figure 7 indicates significant negative interaction effects at weeks 12–20 of PM_{2.5} and 6–19 weeks of temperature at 95% confidence level, implying that increased exposure to PM_{2.5} may decrease the exposure effect of temperature, and vice versa.

4.5 HDLM & HDLMM: Introducing heterogeneity to distributed lag effects

We illustrate heterogeneous models for a single exposure (HDLM) and for a mixture exposure (HDLMM) for estimating heterogeneous exposure effects. We focus our demonstration on a [shiny](#) interface built for HDLM and HDLMM for examining the most significant modifying factors, the personalized exposure effects, and the subgroup-specific exposure effects.

Model fitting and summary with selected modifiers

Suppose we are interested in the linear association between BWGAZ and a single exposure $PM_{2.5}$. We additionally assume that the exposure effect of $PM_{2.5}$ may be modified by child sex, maternal age, BMI, and smoking habits across the population. Specifically, to fit heterogeneous models, additional arguments can be passed to `control.het` as a list. An argument `modifiers` can be set to a vector of modifier names. These modifiers must be included in the data frame provided to the `data` argument (`sbd_cov` in this example). By default, the argument is set to include all covariates included in the formula argument as modifiers. Also, the possible number of splitting points of modifiers for heterogeneity can be specified with an integer argument `modifier.splits` to manage the computational cost. We specify `het = TRUE` to fit the HDLM with the following code.

```
hdlm.fit <- dlmtree(
  formula = bwgaz ~ ChildSex + MomAge + MomPriorBMI +
    Race + Hispanic + SmkAny + EstMonthConcept,
  data = sbd_cov,
  exposure.data = sbd_exp[["PM25"]],
  family = "gaussian",
  dlm.type = "linear",
  het = TRUE,
  control.het = list(
    modifiers = c("ChildSex", "MomAge", "MomPriorBMI", "SmkAny"),
    modifier.splits = 10
  ),
  control.mcmc = list(n.burn = 2500, n.iter = 10000, n.thin = 5)
)
```

The `summary` method applied to the object `hdlm.fit` returns the overview of the model fit. The following code returns the summary.

```
hdlm.sum <- summary(hdlm.fit)
print(hdlm.sum)
```

```
---
HDLM summary

Model run info:
- bwgaz ~ ChildSex + MomAge + MomPriorBMI + Race + Hispanic + SmkAny + EstMonthConcept
- sample size: 10,000
- family: gaussian
- 20 trees
- 2500 burn-in iterations
- 10000 post-burn iterations
- 5 thinning factor
- exposure measured at 37 time points
- 0.5 modifier sparsity prior
- 0.95 confidence level
```

Fixed effects:

| | Mean | Lower | Upper |
|----------------------|--------|--------|--------|
| *(Intercept) | 1.272 | 0.914 | 1.629 |
| ChildSexM | 0.127 | -0.302 | 0.566 |
| MomAge | 0.001 | -0.002 | 0.004 |
| *MomPriorBMI | -0.021 | -0.024 | -0.018 |
| RaceAsianPI | 0.045 | -0.074 | 0.171 |
| RaceBlack | 0.055 | -0.070 | 0.182 |
| Racewhite | 0.034 | -0.082 | 0.157 |
| *HispanicNonHispanic | 0.255 | 0.233 | 0.278 |
| *SmkAnyY | -0.406 | -0.453 | -0.359 |
| EstMonthConcept2 | -0.050 | -0.104 | 0.004 |
| *EstMonthConcept3 | -0.129 | -0.188 | -0.070 |
| *EstMonthConcept4 | -0.201 | -0.265 | -0.139 |
| *EstMonthConcept5 | -0.195 | -0.246 | -0.144 |
| *EstMonthConcept6 | -0.199 | -0.249 | -0.144 |
| EstMonthConcept7 | -0.035 | -0.087 | 0.019 |
| *EstMonthConcept8 | 0.147 | 0.088 | 0.208 |
| *EstMonthConcept9 | 0.395 | 0.331 | 0.455 |
| *EstMonthConcept10 | 0.388 | 0.328 | 0.446 |
| *EstMonthConcept11 | 0.343 | 0.290 | 0.397 |
| *EstMonthConcept12 | 0.141 | 0.091 | 0.190 |

* = CI does not contain zero

Modifiers:

| | PIP |
|-------------|--------|
| ChildSex | 1.0000 |
| MomAge | 0.6305 |
| MomPriorBMI | 0.9055 |
| SmkAny | 0.0975 |

PIP = Posterior inclusion probability

residual standard errors: 0.004

To obtain exposure effect estimates, use the 'shiny(fit)' function.

As before, the summary output includes 'Model run info' and 'Fixed effects' sections with the estimates and the 95% credible intervals of the regression coefficients of the fixed effect. The summary additionally shows the sparsity hyperparameter set for modifier selection and the posterior inclusion probability (PIP) of the modifiers included in the model. The fitted HDLM identified child sex, maternal age, and BMI to be the modifiers that contributed the most heterogeneity to the exposure effect of PM_{2.5}. It is important to note that a high PIP does not necessarily indicate modification. Instead, it suggests that their posterior should be examined to determine if any meaningful modification is present.

A similar model fitting process can be done when examining the heterogeneous exposure effect of a mixture of five exposures on BWGAZ. The following code additionally specifies `mixture = TRUE` and fits HDLMM with the same potential modifiers:

```
hdlmm.fit <- dlmtree(
  formula = bwgaz ~ ChildSex + MomAge + MomPriorBMI +
    Race + Hispanic + SmkAny + EstMonthConcept,
  data = sbd_cov,
  exposure.data = sbd_exp,
```

```

family = "gaussian",
dml.type = "linear",
mixture = TRUE,
het = TRUE,
control.het = list(
  modifiers = c("ChildSex", "MomAge", "MomPriorBMI", "SmkAny"),
  modifier.splits = 10),
),
control.mcmc = list(n.burn = 2500, n.iter = 10000, n.thin = 5)
)

```

As previously, the summary method on `hdlmm` model object similarly returns the summary of the fitted HDLMM. The summary output, omitted here, is similar to that of HDLM, with additional information such as the number of exposures, number of pairwise interactions, and sparsity parameters for modifier selection and exposure selection. Unlike the summary method applied to the model of class `tdlmm` in Section 4.4, marginalization methods are unavailable for the fitted model `hdlmm.fit` as the marginalization of co-exposure with heterogeneity is not well defined.

Using shiny app to investigate heterogeneous distributed lag effects

Exposure effects are estimated at the individual level and can be summarized at either the individual or subgroup level. This flexibility makes summarizing and visualizing the estimated effects challenging. A built-in [shiny](#) app with an object of class `hdlm` and `hdlmm` provides a comprehensive analysis of the exposure effects. HDLM and HDLMM share the same [shiny](#) interface but the shiny method applied to class `hdlmm` additionally includes an option in the panel to select an exposure of interest from mixture exposures. We present the [shiny](#) app interface using the fitted HDLM for a single exposure, using the same argument specification for fitting the model `hdlmm.fit`. The [shiny](#) app is launched with the following code.

```
shiny(hdlm.fit)
```

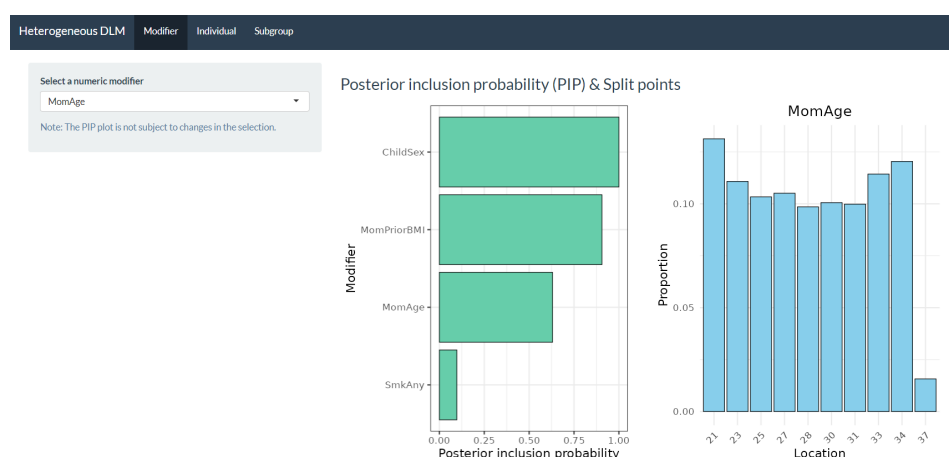


Figure 8: Example of R [shiny](#) user interface for the fitted HDLM. The displayed results are based on simulated data.

Figure 8 shows the main screen, also the first tab, of the [shiny](#) app for the fitted HDLM. The [shiny](#) interface includes three tabs. The first tab labeled 'Modifier' presents two panels including a bar plot of modifier PIPs and the proportions of split points of a user-selected continuous modifier used to split the internal nodes of modifier trees.

In the 'Individual' tab, the user can adjust the levels of modifiers to obtain the individualized estimate of the distributed lag effects. In our context, the [shiny](#) app provides

the personalized exposure effect and critical windows when the sex of a child, age, BMI, and smoking habits of a mother are specified. Figure 9 shows the estimated personalized exposure effect of PM_{2.5} during gestational weeks for a 29-year-old mother with a BMI of 24, whose child is male and who does not smoke.

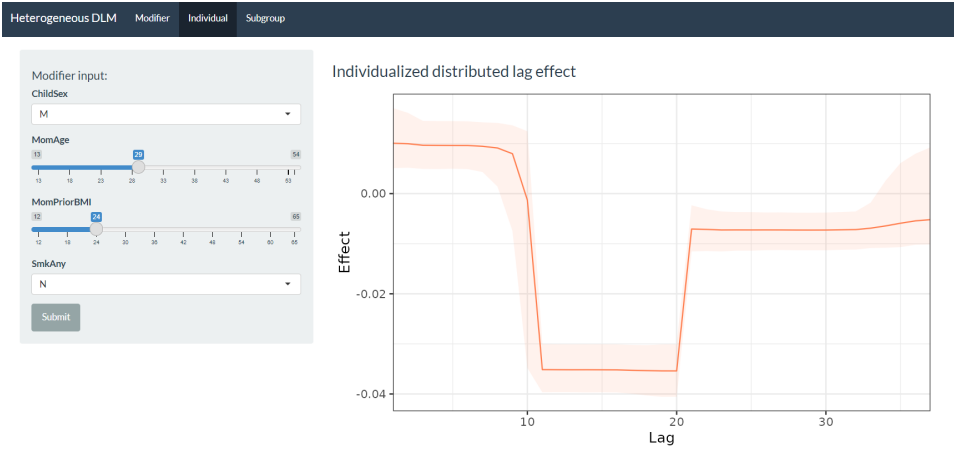


Figure 9: Example of estimated personalized exposure effect in R shiny app. The displayed results are based on simulated data.

The last tab labeled ‘Subgroup’ offers subgroup-specific analyses. In the top panel, shown in Figure 10, the user can select one or two modifiers to group the samples into multiple subgroups and obtain personalized exposure effects for individuals in each subgroup. Each line of the resulting plot represents an exposure effect of an individual accounting for all modifiers of that individual. This is useful for simultaneously assessing how much exposure effects vary among individuals and across subgroups. The bottom panel allows users to

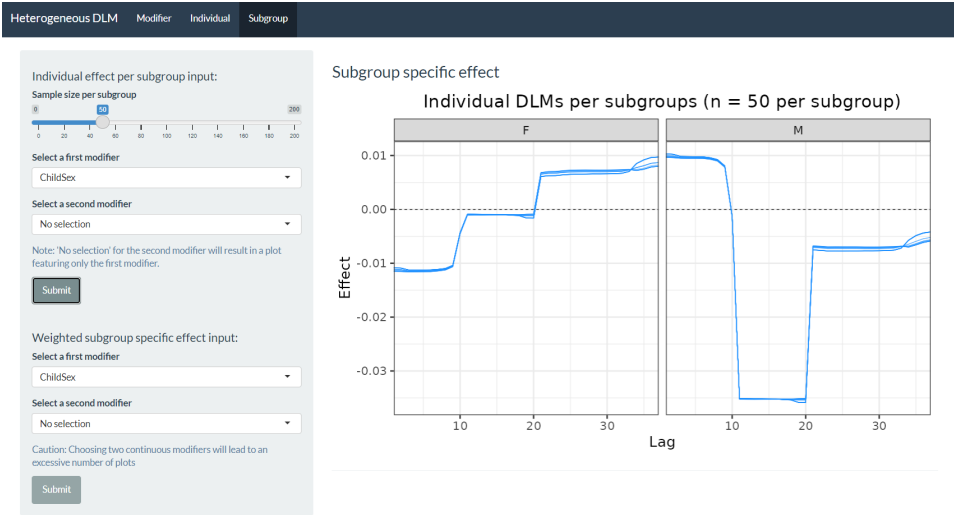


Figure 10: Example of estimated personalized exposure effects with subgroups in R shiny app. The displayed results are based on simulated data.

select one or two modifiers for subgroup-specific distributed lag effects. Subgroup-specific distributed lag effects are calculated by marginalizing out the modifiers not specified in the panel. Two modifiers at most can be specified for analyzing how much each modifier affects heterogeneity in the exposure effects. For example, Figure 11 shows the estimated exposure-time-response function for four subgroups, grouped by two categorical modifiers: child sex and smoking habit. The difference in subgroup-specific exposure effects indicates that smoking habit does not contribute much heterogeneity in the exposure effect of PM_{2.5}, while child sex introduces a considerable amount, which aligns with the modifier PIPs in

the summary output. Using the simulated dataset, the subgroup-specific effects suggest that mothers with a male child may be more vulnerable to $PM_{2.5}$.

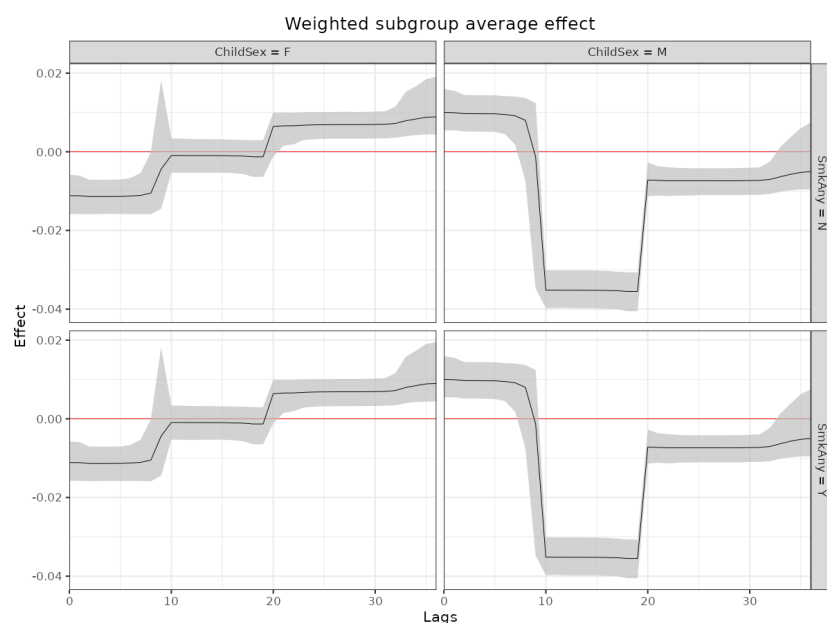


Figure 11: Example of estimated subgroup-specific effects, grouped by child sex and maternal smoking status, in R [shiny](#) app. These results are based on simulated data.

5 Practical considerations

R package [dlmtree](#) offers highly flexible and integrative models for analyzing the effects of repeatedly measured exposures. The package allows user specification for adjusting the flexibility of the exposure-lag-response relationship, number of exposures, types of lagged interaction terms, modifiers, and MCMC simulation control. We highlight important considerations and best practices when using the package to obtain reliable results.

The models included in [dlmtree](#) are very flexible and highly parametric. It is, therefore, important to be mindful of whether such a complex model is relevant to a research question. Factors that increase model complexity are: increased number of time points the exposure is assessed at, more unique exposures and allowing for additional interactions between or within exposures across time, and including heterogeneity and the number of candidate modifiers. The BART framework and shrinkage and selection priors are effective at regularizing these models. However, more complex models can still result in increased variance, identifiability issues where the model becomes sensitive to model specification, and challenges in interpretation. Therefore, we recommend users employ standard best practices in model building, such as carefully assessing what should be included in the model, considering pre-selection of exposures or combining related exposures into a single index, aggregating exposure measurements to a coarser time interval, being mindful of extreme multicollinearity, and conducting sensitivity analyses with simpler models where possible.

When running heterogeneous distributed lag models, potentially important modifiers can be identified using the PIP. When the number of modifiers is small relative to the number of trees, all modifiers will have a higher baseline PIP. Additionally, a large PIP does not necessarily imply meaningful effect modification, and the posterior distribution of the exposure-response function should be explored to better identify important modifiers. Related to model complexity, when two or more modifiers are highly correlated, one or both may be identified as driving heterogeneity. Therefore, considering the mechanisms of effect modification is essential when selecting modifiers and interpreting results.

For practical implementation, we recommend running longer MCMC chains than are used in the examples in this paper. While the examples presented use short iterations for demonstration, longer chains are necessary for stable estimates and to improve mixing and convergence. Given the high dimensionality of the tree structured DLMs, a longer chain will sample more reliable full posterior distributions for inference. Additionally, for logistic and negative binomial models, longer burn-in periods are important to ensure convergence. For example, in our previous large data analyses with this software (Mork and Wilson, 2023; Mork et al., 2024; Mork and Wilson, 2022), we have used upwards of 10,000 burn-in iterations.

Convergence can be assessed in several ways. We have relied on trace plots, which graph the chain of posterior draws for key parameters. When possible, another approach is to run multiple Markov chains with different seeds and compare results across chains. To help users assess model convergence and identify potential issues with mixing, our package provides a `diagnose` S3 method for model summary objects which returns a comprehensive convergence diagnostics panel customized for the `treed` DLM framework. It provides an overview of key outputs including trace plots and posterior density plots of distributed lag effects, Metropolis-Hastings acceptance rate for tree updates, and changes in tree sizes throughout MCMC sampling. The example usage of `diagnose` is demonstrated in the supplementary materials.

6 Summary

We introduced the R package `d1mtree`, a user-friendly software for addressing a wide range of research questions regarding the relationship between a longitudinally assessed exposure or mixture and a scalar outcome. Our software provides functionality to estimate distributed lag linear or nonlinear models, quantify main and interaction effects, and account for heterogeneity in the exposure-time-response function. Furthermore, the methods in our package have been carefully optimized for computational efficiency under a custom C++ language framework with a convenient R wrapper function, `d1mtree`, designed for accessibility by all researchers. In this paper, we provided an overview of the regression tree approaches used for estimating DLMs, and through a collection of vignettes, we highlighted a variety of tools available for processing data, fitting models, conducting inference, and visualizing the results. Our goal in making this package available is to bring robust data science tools to expand the range of questions that can be asked and answered with longitudinally assessed data.

7 Acknowledgements

Research reported in this publication was supported by National Institute of Environmental Health Sciences of the National Institutes of Health under award numbers ES035735, ES029943, ES028811, AG066793, and ES034021. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

- H. A. Chipman, E. I. George, and R. E. McCulloch. BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010. URL <https://doi.org/10.1214/09-AOAS285>. [p136]
- Y.-H. M. Chiu, A. Wilson, H.-H. L. Hsu, H. Jamal, N. Mathews, I. Kloog, J. Schwartz, D. C. Bellinger, N. Khani, R. O. Wright, et al. Prenatal ambient air pollutant mixture exposure and neurodevelopment in urban children in the northeastern United States. *Environmental Research*, 233:116394, 2023. URL <https://doi.org/j.envres.2023.116394>. [p136]

- D. Demateis, K. P. Keller, D. Rojas-Rueda, M.-A. Kioumourtoglou, and A. Wilson. Penalized distributed lag interaction model: Air pollution, birth weight, and neighborhood vulnerability. *Environmetrics*, page e2843, 2024. URL <https://doi.org/10.1002/env.2843>. [p137]
- D. Dempsey and J. Wyse. Bayesian variable selection in distributed lag models: A focus on binary quantile and count data regressions, 2025. URL <https://arxiv.org/abs/2403.03646>. [p137]
- A. Gasparrini. Distributed lag linear and non-linear models in R: The package dlnm. *Journal of Statistical Software*, 43(8):1–20, 2011. URL <https://doi.org/10.18637/jss.v043.i08>. [p137]
- A. Gasparrini, B. Armstrong, and M. G. Kenward. Distributed lag non-linear models. *Statistics in Medicine*, 29(21):2224–2234, 2010. URL <https://doi.org/10.1002/sim.3940>. [p136]
- H.-H. L. Hsu, A. Wilson, J. Schwartz, I. Kloog, R. O. Wright, B. A. Coull, and R. J. Wright. Prenatal ambient air pollutant mixture exposure and early school-age lung function. *Environmental Epidemiology*, 7(2):e249, 2023. URL <https://doi.org/10.1097/EE9.000000000000249>. [p136]
- L. M. Koyck. *Distributed lags and investment analysis*, volume 4. North-Holland Publishing Company Amsterdam, 1954. [p136]
- D. Mork and A. Wilson. Treed distributed lag nonlinear models. *Biostatistics*, 23(3):754–771, 2022. URL <https://doi.org/10.1093/biostatistics/kxaa051>. [p136, 137, 140, 157]
- D. Mork and A. Wilson. Estimating perinatal critical windows of susceptibility to environmental mixtures via structured Bayesian regression tree pairs. *Biometrics*, 79(1):449–461, 2023. URL <https://doi.org/10.1111/biom.13568>. [p136, 137, 138, 139, 140, 157]
- D. Mork and A. Wilson. Incorporating prior information into distributed lag nonlinear models with zero-inflated monotone regression trees. *Bayesian Analysis*, 1(1):1–29, 2024. URL <https://doi.org/10.1214/23-BA1412>. [p140]
- D. Mork, M.-A. Kioumourtoglou, M. Weisskopf, B. A. Coull, and A. Wilson. Heterogeneous distributed lag models to estimate personalized effects of maternal exposures to air pollution. *Journal of the American Statistical Association*, 119(545):14–26, 2024. URL <https://doi.org/10.1080/01621459.2023.2258595>. [p136, 141, 157]
- B. Neelon. Bayesian zero-inflated negative binomial regression based on pólya-gamma mixtures. *Bayesian Analysis*, 14(3):829, 2019. URL <https://doi.org/10.1214/18-ba1132>. [p140]
- K. S. Palda. The measurement of cumulative advertising effects. *The Journal of Business*, 38(2):162–179, 1965. [p136]
- N. G. Polson, J. G. Scott, and J. Windle. Bayesian inference for logistic models using pólya-gamma latent variables. *Journal of the American Statistical Association*, 108(504):1339–1349, 2013. URL <https://doi.org/10.1080/01621459.2013.829001>. [p140]
- M. J. Rosa, H.-H. L. Hsu, A. C. Just, K. J. Brennan, T. Bloomquist, I. Kloog, I. Pantic, A. M. García, A. Wilson, B. A. Coull, et al. Association between prenatal particulate air pollution exposure and telomere length in cord blood: Effect modification by fetal sex. *Environmental Research*, 172:495–501, 2019. URL <https://doi.org/10.1016/j.envres.2019.03.003>. [p141]
- E. M. Schliep, T. L. J. Schafer, and M. Hawkey. Distributed lag models to identify the cumulative effects of training and recovery in athletes using multivariate ordinal wellness data. *Journal of Quantitative Analysis in Sports*, 17(3):241–254, 2021. URL <https://doi.org/10.1515/jqas-2020-0051>. [p136]

- J. Schwartz. The distributed lag between air pollution and daily deaths. *Epidemiology*, 11(3): 320–326, 2000. URL <https://doi.org/10.1097/00001648-200005000-00016>. [p136]
- J. Warren, M. Fuentes, A. Herring, and P. Langlois. Spatial-temporal modeling of the association between air pollution exposure and preterm birth: Identifying critical windows of exposure. *Biometrics*, 68(4):1157–1167, 2012. URL <https://doi.org/10.1111/j.1541-0420.2012.01774.x>. [p136]
- A. Wilson, Y.-H. M. Chiu, H.-H. L. Hsu, R. O. Wright, R. J. Wright, and B. A. Coull. Potential for bias when estimating critical windows for air pollution in children’s health. *American Journal of Epidemiology*, 186(11):1281–1289, 06 2017a. ISSN 0002-9262. doi: 10.1093/aje/kwx184. URL <https://doi.org/10.1093/aje/kwx184>. [p136]
- A. Wilson, Y.-H. M. Chiu, H.-H. L. Hsu, R. O. Wright, R. J. Wright, and B. A. Coull. Bayesian distributed lag interaction models to identify perinatal windows of vulnerability in children’s health. *Biostatistics*, 18(3):537–552, 2017b. URL <https://doi.org/10.1093/biostatistics/kxx002>. [p137]
- A. Zanobetti, M. P. Wand, J. Schwartz, and L. M. Ryan. Generalized additive distributed lag models: Quantifying mortality displacement. *Biostatistics*, 1(3):279–292, 2000. URL <https://doi.org/10.1093/biostatistics/1.3.279>. [p136]

Seongwon Im

Department of Statistics
Colorado State University
United States of America
(0009-0000-8447-5852)
seongwon.im@colostate.edu

Ander Wilson

Department of Statistics
Colorado State University
United States of America
(0000-0003-4774-3883)
ander.wilson@colostate.edu

Daniel Mork

Department of Biostatistics
Harvard T.H. Chan School of Public Health
United States of America
(0000-0002-7924-0706)
dmork@hsph.harvard.edu