# Flexible Bayesian Regression Models for Quantifying Complex Interactions in Exposure Mixture Studies

*Elizabeth Zhang*
April DD, 20YY

Submitted to the Department of
Mathematics and Statistics
of Amherst College in partial fulfillment
of the requirements for the degree of
Bachelor of Arts with honors.

Advisor:
*Amy Wagaman*

# Abstract

The abstract should be a short summary of your thesis work. A paragraph is usually sufficient here.

# Acknowledgments

Use this space to thank those who have helped you in the thesis process (professors, staff, friends, family, etc.). If you had special funding to conduct your thesis work, that should be acknowledged here as well.

This work was performed in part using high-performance computing equipment at Amherst College obtained under National Science Foundation Grant Number 2117377. The data used for simulations in this thesis was supported by the National Institute of Environmental Health Sciences of the National Institutes of Health under Award Numbers U2CES026555 and U2CES026553. The content is solely the responsibility of the author and does not necessarily represent the official views of the National Science Foundation or the National Institutes of Health.

# Table of Contents

**Abstract** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **i**

**Acknowledgments** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **iii**

**List of Tables** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **vii**

**List of Figures** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **ix**

**Chapter 1: Introduction** . . . . . . . . . . . . . . . . . . . . . . . . . **1**

**Chapter 2: Humanistic Perspective** . . . . . . . . . . . . . . . . . **5**

**Chapter 3: Bayesian Regression Methods** . . . . . . . . . . . . . **7**

- 3.1 Motivation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
  - 3.1.1 Interactions from a statistical perspective . . . . . . . . . . . 7
  - 3.1.2 Mechanistic and public health relevance . . . . . . . . . . . . 9
- 3.2 Bayesian kernel machine regression (BKMR) . . . . . . . . . . . . . 10
  - 3.2.1 Kernel machine regression . . . . . . . . . . . . . . . . . . . . 11
  - 3.2.2 Connection to mixed models . . . . . . . . . . . . . . . . . . . 14
  - 3.2.3 Toy example . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
  - 3.2.4 Variable selection . . . . . . . . . . . . . . . . . . . . . . . . 18
  - 3.2.5 Prior specification . . . . . . . . . . . . . . . . . . . . . . . . 20
  - 3.2.6 The MCMC algorithm . . . . . . . . . . . . . . . . . . . . . . 21
- 3.3 Bayesian semiparametric regression (BSR) . . . . . . . . . . . . . . 22
  - 3.3.1 Spline regression . . . . . . . . . . . . . . . . . . . . . . . . . 23

3.3.2 Toy example . . . . . . . . . . . . . . . . . . . . . . . . . . . 26
3.3.3 Model formulation in BSR . . . . . . . . . . . . . . . . . . . . 29
3.3.4 Sparsity inducing priors . . . . . . . . . . . . . . . . . . . . . 30
3.3.5 Prior specification . . . . . . . . . . . . . . . . . . . . . . . . 31
3.3.6 The MCMC algorithm . . . . . . . . . . . . . . . . . . . . . . 32
3.4 Detecting interactions . . . . . . . . . . . . . . . . . . . . . . . . . 33
3.4.1 BKMR . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33
3.4.2 BSR . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
3.4.3 Differences between BKMR and BSR . . . . . . . . . . . . . . 35
3.4.4 Exposure-covariate interactions . . . . . . . . . . . . . . . . . 36
**Chapter 4: Simulations** . . . . . . . . . . . . . . . . . . . . . . . . . **37**
4.1 Past simulation studies . . . . . . . . . . . . . . . . . . . . . . . . 37
4.2 Methods . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39
4.2.1 MADRES data . . . . . . . . . . . . . . . . . . . . . . . . . . 39
4.2.2 Using copulas to simulate predictor data . . . . . . . . . . . . 42
4.2.3 Simulating predictor-response relationships . . . . . . . . . . . 47
4.2.4 Models . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
4.2.5 Model assessment . . . . . . . . . . . . . . . . . . . . . . . . . 49
4.3 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
**Conclusion** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **51**
**Appendix A: Supplemental output** . . . . . . . . . . . . . . . . . . . **53**
A.1 Methods . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53
A.2 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59
**Appendix B: Code** . . . . . . . . . . . . . . . . . . . . . . . . . . . . **61**
B.1 Code for Chapter 3: . . . . . . . . . . . . . . . . . . . . . . . . . . 61
B.2 Code for Chapter 4: . . . . . . . . . . . . . . . . . . . . . . . . . . 66

B.2.1 Code for Chapter 4.2.1: . . . . . . . . . . . . . . . . . . . . . . . 66
B.2.2 Code for Chapter 4.2.2: . . . . . . . . . . . . . . . . . . . . . . . 69
B.2.3 Code for Chapter 4.2.3: . . . . . . . . . . . . . . . . . . . . . . . 73
B.2.4 Code for Chapter 4.2.4: . . . . . . . . . . . . . . . . . . . . . . . 83
B.2.5 Code for Chapter 4.3: . . . . . . . . . . . . . . . . . . . . . . . . 83
**Corrections** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **85**
**References** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **87**

# List of Tables

4.1 Specification of interaction terms in simulations. . . . . . . . . . . . . . 48

# List of Figures

3.1 Non-linear data with a true relationship (orange) and a fitted linear regression (blue). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

3.2 A query point of 12.5 and the weights of neighboring observations based on a Gaussian kernel . . . . . . . . . . . . . . . . . . . . . . . . . . . 16

3.3 Fitted kernel machine regression (blue) with $\rho = 2$ compared to the true relationship (orange). . . . . . . . . . . . . . . . . . . . . . . . 17

3.4 Fitted kernel machine regression with $\rho = 0.02$ and $\rho = 50$. . . . . . . 17

3.5 Linear spline regression (blue) with four knots (dotted lines) compared to the true relationship (orange). . . . . . . . . . . . . . . . . . . . . 26

3.6 Cubic spline regression (blue) with four knots (dotted lines) compared to the true relationship (orange). . . . . . . . . . . . . . . . . . . . . 27

3.7 Natural spline regression (blue) with four knots (dotted lines) compared to the true relationship (orange). . . . . . . . . . . . . . . . . . 28

3.8 Natural and cubic spline regression (blue) compared to the true relationship (orange) extrapolated outside the bounds of $x$ (dotted lines). 29

4.1 Distributions of original (a) and natural log transformed (b) concentrations of metals in MADRES cohort (n=252). . . . . . . . . . . . 40

4.2 Distributions of continuous (a) and categorical (b) covariates in the MADRES cohort (n=252). . . . . . . . . . . . . . . . . . . . . . . . . 41

4.3 Association between race by ethnicity and birth place and metal exposures in the MADRES cohort (n=252). . . . . . . . . . . . . . . . . . . 43
4.4 Distributions of exposures from observed data (blue) and simulated data (gray). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 44
4.5 Distributions of covariates from observed data (blue) and simulated data (gray). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45
4.6 Correlation heat maps of exposures from observed data (a) and averaged from simulated data (b), as well as distribution of correlations from simulated data (c). . . . . . . . . . . . . . . . . . . . . . . . . . 46

A.1 Surface for base case. . . . . . . . . . . . . . . . . . . . . . . . . . 53
A.2 Surface for Hg, Ni, mult low eff. . . . . . . . . . . . . . . . . . . . . 54
A.3 Surface for Hg, Ni, mult high eff. . . . . . . . . . . . . . . . . . . . 54
A.4 Surface for Hg, Ni, poly low eff. . . . . . . . . . . . . . . . . . . . . 55
A.5 Surface for Hg, Ni, poly high eff. . . . . . . . . . . . . . . . . . . . . 55
A.6 Surface for Cd, As, mult low eff. . . . . . . . . . . . . . . . . . . . . 56
A.7 Surface for Cd, As, mult high eff. . . . . . . . . . . . . . . . . . . . 56
A.8 Surface for Cd, As, poly low eff. . . . . . . . . . . . . . . . . . . . . 57
A.9 Surface for Cd, As, poly high eff. . . . . . . . . . . . . . . . . . . . 57
A.10 Surface for Ni, Co, mult low eff. . . . . . . . . . . . . . . . . . . . . 58
A.11 Surface for Ni, Co, mult high eff. . . . . . . . . . . . . . . . . . . . 58
A.12 Surface for Ni, Co, poly low eff. . . . . . . . . . . . . . . . . . . . . 59
A.13 Surface for Ni, Co, poly low eff. . . . . . . . . . . . . . . . . . . . . 59

# Chapter 1 Introduction

Rapid industrial development has created conditions of cumulative chronic toxicity which pose an acute risk to the wellbeing of humans and our living environment. In fact, it has been estimated that, globally, human activity releases chemicals at a rate of 220 billion tons per annum (Cribb, 2016). These staggering levels of pollution have led scholars to formally declare that humanity has surpassed the safe operating space of the planetary boundary for novel entities (Persson et al., 2022). As a result, exposure to low levels of pollutants has become an inevitable peril of daily life (Naidu et al., 2021; Vineis, 2018). In this new era of pervasive toxicity, understanding the nature and severity of health effects associated with chemical exposures is especially timely.

For this, we turn to epidemiological studies. The broad field of preventive epidemiology involves the identification of potentially modifiable risk factors that contribute to the burden of disease within human populations. Environmental epidemiology, in particular, considers the effect of environmental exposures — chemical or otherwise. However, studies concerning chemical pollutants in environmental epidemiology have historically focused on elucidating the effect and mechanisms of exposures to a single pollutant. In reality, humans are invariably exposed to numerous complex exposure mixtures which together contribute to the progression of adverse health outcomes. Therefore, risk assessments of single pollutants likely fail to capture the true consequences of these complex exposures (Heys, Shore, Pereira, Jones, & Martin, 2016). Assessing mixtures of chemicals can also have more direct implications for public health interventions. The United States Environmental Protection Agency (U.S. EPA) currently passes regulations for individual pollutants. In practice, though, regulation occurs by controlling the source of pollution, which is responsible for the production of a whole mixture of chemicals with specific joint effects on human health. As a result, the National Academies of Science has advocated for a multipollutant regulatory approach, which is likely to be more protective of human health (National Academies of Sciences, Engineering, and Medicine, Division on Earth and Life Studies, Board on Environmental Studies and Toxicology, & Committee on Incorporating 21st Century Science into Risk-Based Evaluations, 2017).

There are clear practical motivations for studies that examine the health effects of exposure to co-occurring mixtures of chemicals, hereafter referred to as exposure mixtures. However, expanding the focus of analysis from one exposure to multiple exposures introduces unique statistical challenges. In addition to a common issue of small effect sizes and small sample sizes present in most exposure analyses, multiple exposure analyses must also contend with high-dimensionality, collinearity, non-linear effects, and non-additive interactions (Yu et al., 2022). In particular, data with numerous pollutants, or predictors, require exponentially greater levels of complexity and time cost in analysis. Collinearity between exposures is common when analyzing pollutants from a single source and can lead to unstable estimates in a generalized linear model if left unaccounted for. Finally, exposures can have both non-linear single effects and non-additive interaction effects, which are difficult to capture unless explicitly specified in the model.

The classic multiple linear regression framework often fails to capture the true effects in this setting. In the past few years, a wide variety of statistical methods have been developed to overcome these challenges (see reviews at Gibson et al., 2019; Yu et al., 2022), which have been accompanied by a host of comparative simulation studies for general mixture scenarios (e.g., Hoskovec, Benka-Coker, Severson, Magzamen, & Wilson, 2021; Lazarevic, Knibbs, Sly, & Barnett, 2020; Pesenti et al., 2023). However, to our knowledge, there has yet to be a simulation study which provides conclusive guidance about the ability of recently developed methods to conduct inference on non-additive interactions between exposures when the nature and effect sizes of these interactions vary.

The goal of this thesis is to fill this gap in the literature by exploring the theory and performance of Bayesian regression techniques for quantifying complex interactions between multiple environmental exposures and related covariates. Specifically, we will compare two recently developed models for estimating the health effects of exposure mixtures: Bayesian Kernel Machine Regression (BKMR) (Bobb et al., 2015) and Bayesian Semiparametric Regression (BSR) (Antonelli et al., 2020).

In an age where anthropogenic actions have radically reshaped the earth, humanistic inquiry can offer critical insights into how we navigate the hazards of our rapidly changing environment. We begin in Chapter 2 by contextualizing this thesis with a brief overview of cultural and social understandings of toxicity. Chapter 3 explains the motivation for studying interactions and provides background on the theory of Bayesian methods for analyzing exposure mixtures. Chapter 4 assesses the performance of these methods using a simulation study, based on a dataset with information on the relationship between prenatal exposure to heavy metals and gestational weight. We conclude with a discussion of the implications of this work for the future study of complex interactions in exposure mixture studies.

# Chapter 2 Humanistic Perspective

List of loose ideas:

- goal of modernism: define single causes for single effects
    - study chemicals in isolation to obtain purely mechanistic explanation of their toxicity
- contrast with idea of relationality: entities cannot be understood without considering their *relationality* to other surrounding entities
    - relationality disrupts the notion of bounded objects
    - motivation for chemical mixtures: chemicals themselves are not independent from surrounding chemicals
    - motivation for chemical mixtures in the context of social epidemiology: the effects of chemicals are modulated by structural/social conditions
- relationality also disrupts Cartesian split between body and mind
    - racial hierarchy positions certain groups closer to the bounds of the corporal body, while other groups have transcended these bounds and are defined by their intellect (i.e., the mind)
    - result –> some bodies are seen as inherently more susceptible to chemical exposure, more "porous"
    - leads to damage centered research which, while well-intentioned, inadvertently de-humanizes marginalized groups

- remedy: relationality leads into concept of alterlife, modern life is inseparable from alteration due to chemical exposure

# Chapter 3 Bayesian Regression Methods

## 3.1 Motivation

We are interested in using Bayesian regression techniques to characterize the nature of non-additive interactions in exposure mixture studies. We begin by reviewing definitions for what constitutes an interaction and why interactions are relevant from public health and biological relevance.

### 3.1.1 Interactions from a statistical perspective

First, we define additivity and non-additivity in the traditional statistical paradigm (Siemiatycki & Thomas, 1981). Suppose we have two variables $x_1$ and $x_2$, and we want to consider their effect on some outcome of interest. If specifying [effect due to $x_1$ and $x_2$] = [effect due to $x_1$] + [effect due to $x_2$] can adequately capture this relationship, then we say that $x_1$ and $x_2$ each have an **additive effect** on the outcome and that there is no interaction between them. On the other hand, if there is variability in the outcome that can be captured by an additional term equal to some function of $x_1$ and $x_2$, we say that there is a **non-additive interaction** between $x_1$ and $x_2$. In this case, [effect due to $x_1$ and $x_2$] = [effect due to $x_1$] + [effect due to $x_2$] + [effect due to $f(x_1, x_2)$], where $f$ is a non-zero function.

For our sake, when we refer to "interaction," we mean any non-additive interaction. We consider such non-additive interactions to be complex, meaning that they are difficult to detect. To see why, let us consider running a linear regression for $Y$ on $x_1$ and $x_2$. The theoretical model would be defined as

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} f(x_1, x_2),$$

where the $\beta$'s represent the effect sizes. We can see that the form of the interaction must be explicitly specified in the formulation of the model. Most commonly, a multiplicative interaction is assessed, where $f(x_1, x_2) = x_1 * x_2$. However, a non-additive interaction can take on many different forms, the true nature of which is difficult to determine analytically.

We used a two-predictor case above, but interactions can also exist between more variables (i.e., two-way by $f(x_1, x_2)$, three-way by $f(x_1, x_2, x_3)$, etc.). So, if we wanted to assess all possible interactions, the number to consider quickly becomes intractable in high-dimensional settings. For instance, consider modelling 10 predictors in the above linear regression setting. In order to be assessed, each interaction must be explicitly specified as a new term in the model. Even if we only considered one form for each interaction, including all possible two-way interactions would involve adding $\binom{10}{2} = 45$ additional terms to the model, and all possible three-way interactions would add $\binom{10}{3} = 120$ additional terms.

It is important also to acknowledge, here, that there is a limit to how many variables can be included in an interaction before it becomes incomprehensible to most humans. For instance, Halford, Baker, McCredden, & Bain (2005) suggest that there is a steep decline in interpretability from three- to four-way interactions, and that five-way interactions are only interpreted correctly at chance level (Halford et al., 2005). Hence, for practical purposes, we will limit our exploration to two- and three-way interactions.

### 3.1.2 Mechanistic and public health relevance

Thus far, we have discussed interactions within a statistical paradigm. However, in addition to being an interesting estimation challenge, non-additive interactions are also relevant in exposure mixture studies from both a mechanistic and public health point of view.

From a mechanistic perspective, a non-additive statistical interaction between two chemical exposures suggests that these compounds may be functionally interacting with each other. Theoretical models propose that such interactions can be classified as either synergistic or antagonistic (Heys et al., 2016; Plackett & Hewlett, 1952). In a synergistic interaction, the joint effects of a mixture exceed the independent effects of each component. This usually occurs if a chemical induces an enzyme involved with the activation of a second chemical or if a chemical inhibits an enzyme that would have otherwise degraded a second chemical. For example, it has been shown that organophosphates slow the degradation of pyrethoids by inhibiting detoxifying enzymes — these two classes of chemicals are often found together in commercial insecticide mixtures (Hernández et al., 2013).

On the other hand, in an antagonistic interaction, the joint effects of a mixture are less than their independent effects. This can occur either through competition at the target site of an enzyme or through direct chemical reactions with each other. In general, synergistic interactions are more concerning in risk assessments, as they lead to underestimation of the true toxicity of a mixture.

It should be noted, though, that while statistical interactions may provide some insight into how exposure mixtures are related to health, they cannot confirm their underlying biology (VanderWeele & Knol, 2014). If the goal is to assess a meaningful biological interaction, then the discovery of a statistical interaction should be followed up by a functional study.

Now, from a public health perspective, we might be interested in how exposure mixtures interact with other covariates, or, in other words, how social and health factors might mediate the relationship between a health outcome and chemical exposures (VanderWeele & Knol, 2014). In our case, we can include these additional covariates in the exposure mixture model, where, statistically, they would contribute to the model in the same manner as another chemical exposure: a predictor. A statistical interaction in our model between a covariate and an exposure would indicate that the *magnitude* of the effect of reducing the level of an exposure might differ across various levels of the covariate. This finding could be relevant to public health policy makers, as the potential benefit of regulating a pollutant might differ across groups. For instance, it has been suggested that nutritional intake may modify susceptibility to chemical exposures (e.g., Kannan, Misra, Dvonch, & Krishnakumar, 2006; Kordas, Lönnerdal, & Stoltzfus, 2007).

In many cases, we might assess a covariate related to health inequity, such as socioeconomic status. We provide a cautionary comment, here, that an interaction term should not be the *sole* measure used to measure a health disparity (Ward et al., 2019). In this case, we should first consider the independent, additive association between the covariate and levels of exposure or rates of a health outcome, in order to contextualize the meaning of a potential interaction term.

## 3.2 Bayesian kernel machine regression (BKMR)

In this section, we introduce the theory of BKMR. First, we define the notation that we will be using for kernel machine regression:

- $X_m$ is an exposure in the exposure matrix $\mathbf{X}$ with $m = 1, \ldots, M$

- $\mathbf{x}_i$ is a vector of values for a single observation in $\mathbf{X}$ with $i = 1, \ldots, n$
- $x_{im}$ is the $i$th observation of $X_m$
- $\mathbf{z}_i$ is a vector of covariates for a single observation in the matrix $\mathbf{Z}$, which contains a set of covariates, with $i = 1, \ldots, n$
- $Y_i$ is an observation of $\mathbf{Y}$, measuring the health outcome in this case
- $h(\cdot)$ is the flexible function relating $\mathbf{x}$ to $\mathbf{Y}$
- $k$ is the kernel function, the Gaussian in this case
- $\mathbf{K}$ is the $n \times n$ kernel matrix, with $(i, j)$th element $k(\mathbf{x}_i, \mathbf{x}_j)$
- $\rho$ is the parameter which controls smoothnessm, associated with the kernel function
- $\tau$ is the parameter multiplied by the kernel matrix to relate $\mathbf{K}$ to $h$
- $\boldsymbol{\beta}_{\mathbf{z}}$ is a vector of the weights on the covariates, and
- $\boldsymbol{\varepsilon}_i \overset{\text{iid}}{\sim} N(0, \sigma^2)$ are the residuals of the response.

And, we define the notation that we will be using specific to BKMR:

- $r_m = 1/\rho_m$ is an augmented variable in $\mathbf{r}$ in the kernel matrix, controlling smoothness
- $\delta_m$ is an indicator variable in $\boldsymbol{\delta}$ which represents inclusion in the model
- $\mathcal{S}_g$ is a group of partitioned predictors with $g = 1, \ldots, G$
- $\{\delta_m | \mathbf{x}_m \in \mathcal{S}_g\}$ is an indicator variable in $\boldsymbol{\delta}_{\mathcal{S}_g}$ which represents inclusion of a parameter in group $g$ in the model
- $\pi$ is the prior probability of inclusion of a predictor in the model, and
- $\lambda \equiv \tau\sigma^{-2}$ is a convenient way to define the prior on $\tau$.

### 3.2.1 Kernel machine regression

We begin by introducing kernel machine regression, with attention to its specific implementation in BKMR. First proposed by Nadaraya (1964) and Watson (1964), kernel machine regression is a nonparametric regression technique that can be used to capture non-linear effects and non-additive interactions. In this introduction, we follow the presentation of kernel machine regression provided by Bobb et al. (2015).

To contextualize this method, we start at the typical linear regression setting,

$$Y_i = \mathbf{x}_i^\top \boldsymbol{\beta}_\mathbf{x} + \mathbf{z}_i^\top \boldsymbol{\beta}_\mathbf{z} + \varepsilon_i,$$

where $Y_i$ measures a health outcome at a given point, $\mathbf{x}_i = [x_{i1}, \ldots, x_{iM}]$ is a vector of M exposures, $\mathbf{z}_i$ is a vector of covariates, $\boldsymbol{\beta}_\mathbf{x}$ and $\boldsymbol{\beta}_\mathbf{z}$ are vectors of weights for the exposures and covariates, respectively, and $\varepsilon_i$ is a random variable from $\boldsymbol{\varepsilon} \overset{\text{iid}}{\sim} N(0, \sigma^2)$. We can see that this function assumes that there is a linear relationship between the exposure and the response, and that the combined effects of multiple exposures are additive.

Kernel machine regression defines this relationship using a flexible function $h : \mathbb{R}^M \to \mathbb{R}$, where

$$Y_i = h(\mathbf{x}_i) + \mathbf{z}_i^\top \boldsymbol{\beta}_\mathbf{z} + \varepsilon_i.$$

Here, $h(\cdot)$ is represented by the function $k(\cdot, \cdot)$, a kernel. The kernel controls the covariance, or the similarity, between values of $h(\mathbf{x})$ and as such ensures that points near each other on the prediction surface will have similar values — or, in other words, that the prediction surface will be smooth. In the case of kernel machine regression, we define a positive definite kernel where $k : \mathbb{R}^M \times \mathbb{R}^M \to \mathbb{R}$. Note also that covariates are assumed to have a linear, non-additive effect on the response.

There are many choices of functions for $k$. BKMR uses the Gaussian kernel, also known as the radial basis function or, sometimes, the squared exponential kernel. The Gaussian kernel is defined as

$$k(\mathbf{x},\mathbf{x}') = \exp\left\{ -\frac{||\mathbf{x}-\mathbf{x}'||^2}{\rho} \right\},$$

where $||\mathbf{x}-\mathbf{x}'||^2 = \sum_{m=1}^{M}(x_m - x'_m)^2$ for a set of exposure values $\mathbf{x}$ and the exposure values of another subject $\mathbf{x}'$, and $\rho$ is a tuning parameter that controls the relationship between the correlation between two points and their distance. Greater values of $\rho$ will enforce more dependence between points and make the resulting function smoother. $h$ is related to $k$ by a multiplicative constant $\tau$, a tuning parameter which controls the vertical scale of $h$.

Now that we have defined $h$ and $k$, we can think about how to characterize the relationship between our response and exposures Kernel machine regression is a nonparametric technique because it does not specify a functional form for this relationship. Hence, we will think about estimating the response at a particular query point. Operationally, Müller (1987) demonstrated that kernel machine regression uses a weighted average of all the observations in the dataset to estimate the response, defined as

$$\bar{Y} = \frac{\sum_{i=1}^{n} w_i Y_i}{\sum_{i=1}^{n} w_i},$$

with some set of weights $\{w_i\}_{i=1}^{n}$. Intuitively, we want to weight the observations that are closer to the query point more heavily. Using the Gaussian kernel as a weight allows us to achieve this. Replacing the weight with the Gaussian kernel, we get

$$\bar{Y} = \frac{\sum_{i=1}^{n} k(\mathbf{x},\mathbf{x}_i) Y_i}{\sum_{i=1}^{n} k(\mathbf{x},\mathbf{x}_i)}.$$

As we move through the predictor space, we can think of the prediction as a continuous moving average of local points in the dataset. The correlation between two values of $h$ is defined as

$$\operatorname{cor}(h_i, h_j) = \exp\bigg\{ -\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\rho} \bigg\},$$

which allows us to see that values of $h$ near each other will have a higher correlation and thus similar values. This is also why the resulting function is smooth.

### 3.2.2 Connection to mixed models

It is useful to make connections between this definition of kernel machine regression and mixed models. Liu, Lin, & Ghosh (2007) demonstrated this by representing $h(\mathbf{x})$ as following a Gaussian process probability distribution,

$$h(\mathbf{x}) \sim \mathcal{GP}(\mathbf{0}, \tau k(\mathbf{x}, \mathbf{x}')),$$

with covariance function $k$, where $\mathbf{x}$ is a vector of the exposure values, and $\mathbf{x}'$ contains the exposure values of another subject. A Gaussian process is a collection of random variables, of which any finite number follow a multivariate normal distribution (Schulz, Speekenbrink, & Krause, 2018). Here, we assume that the expected value of the $h$ function with input $\mathbf{x}$ is $\mathbf{0}$. We use $k$ for the covariance function, which represents the dependence between the function values with inputs $\mathbf{x}$ and $\mathbf{x}'$: $k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(h(\mathbf{x}) - \mathbf{0})(h(\mathbf{x}') - \mathbf{0})]$.

Now, we can represent $h$ as a collection of variables from a Gaussian process. $h$ follows a multivariate normal distribution,

$$h(\mathbf{x}) \sim N(\mathbf{0}, \tau \mathbf{K}),$$

where $h(\mathbf{x}) = [h(\mathbf{x}_1), h(\mathbf{x}_2), \ldots, h(\mathbf{x}_n)]^\top$ and $\mathbf{K}$ is the kernel matrix. The kernel matrix is an $n \times n$ matrix with $(i, j)$th element $k(\mathbf{x}_i, \mathbf{x}_j)$. Now, returning back to the regression view, we can think of each $Y_i$ as following the distribution,

$$Y_i \overset{\text{ind}}{\sim} N(h(\mathbf{x}_i) + \mathbf{z}_i^\top \boldsymbol{\beta}_\mathbf{z}, \sigma^2) \text{ for } i = 1, \dots, n,$$

where $\sigma^2$ comes from the variance of the residuals. Here, $h$ can be interpreted as a random effect.

### 3.2.3 Toy example

In the following section, we illustrate kernel machine regression with a toy example.

Consider the following case where we want to model the relationship between a single predictor and a response variable. Suppose the true relationship between $x$ and $Y$ is defined $Y = e^{\frac{x}{10}} + 2\sin(\frac{x}{2})$. We simulate 51 equally spaced observations of $x$ from 0 to 25, with error $\varepsilon_i \overset{\text{iid}}{\sim} N(0, 0.25)$.

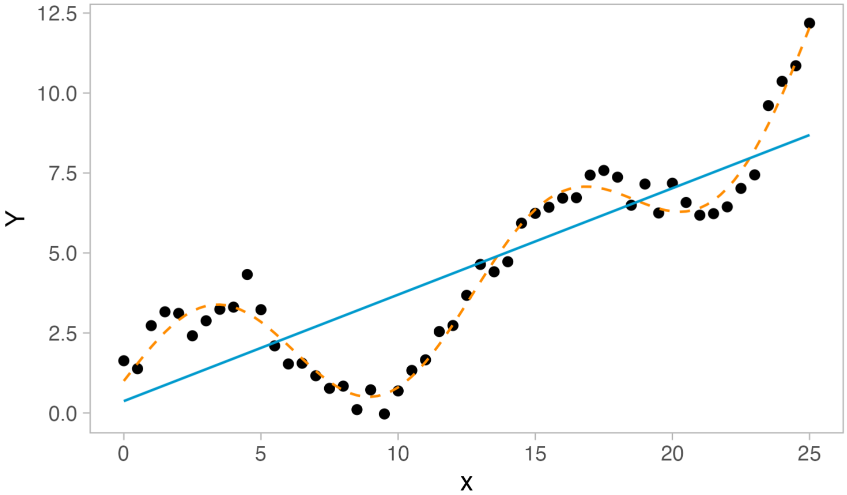

Figure 3.1: Non-linear data with a true relationship (orange) and a fitted linear regression (blue).

Figure 3.1 illustrates the shape of our simulated non-linear data and the fit proposed by a simple linear regression. We can observe that the linear regression fails to capture the true non-linear relationship. In this case, this would lead to an underestimation of the true association between $x$ and $Y$. Now, we will try to capture this relationship using kernel machine regression.

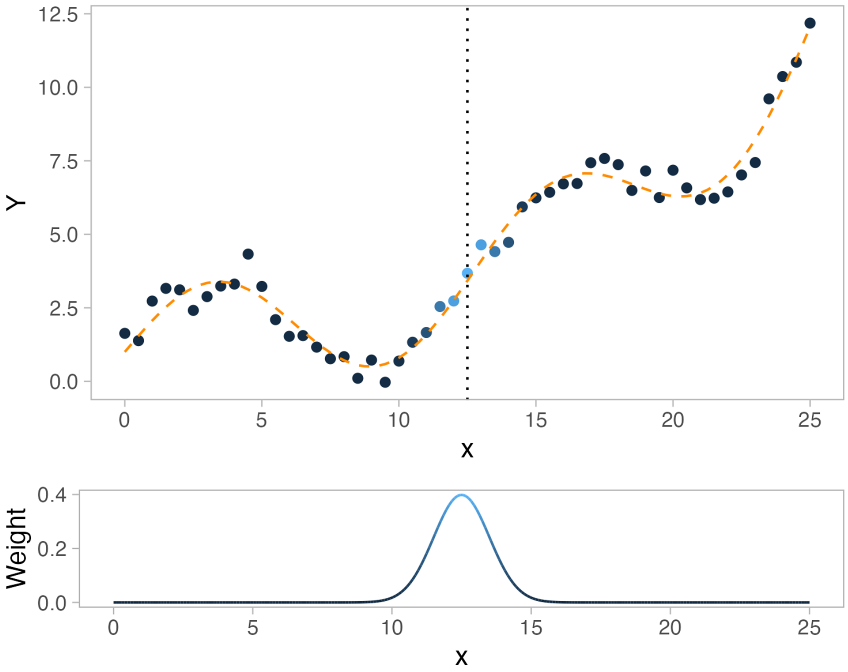

Figure 3.2: A query point of 12.5 and the weights of neighboring observations based on a Gaussian kernel

To visualize how kernel machine regression works as a moving weighted average, we can consider a query point of 12.5. Figure 3.2 identifies the query point and assigns corresponding weights to the neighboring points based on a normal distribution, which shares the same density as the Gaussian kernel. In this case, we will specify $\rho = 2$, which is synonymous with assigning the weights using a normal distribution with $\sigma^2 = 1$. We can see how an appropriate estimate for $h(12.5)$ can be obtained by taking a weighted average of the $Y$'s, with those observations nearby weighted the most heavily.

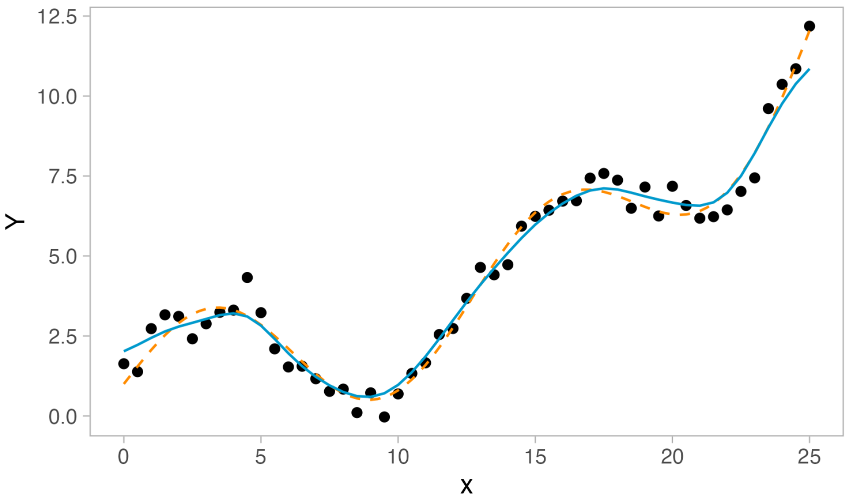

Figure 3.3: Fitted kernel machine regression (blue) with $\rho = 2$ compared to the true relationship (orange).

Now, we fit a kernel machine regression on this data with $\rho = 2$ using the `stats` package in R. We can see in Figure 3.3 that kernel machine regression captures the complex non-linear relationship between $Y$ and $x$ and closely follows the true relationship. We do note, though, that the estimation is less precise at the tails, where there is less information provided by local observations. We can also use this example to consider the effect of various values of $\rho$ on the smoothness of the $h$ function.

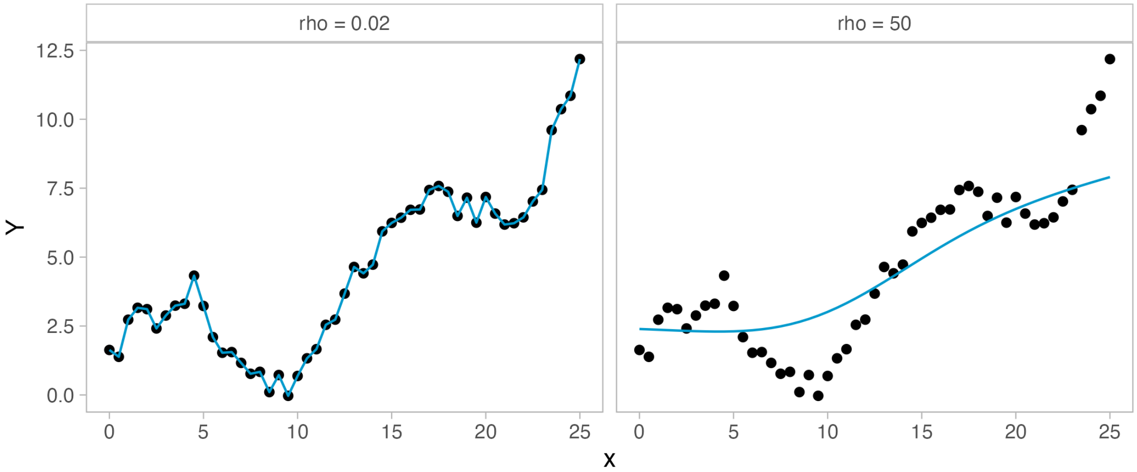

Figure 3.4: Fitted kernel machine regression with $\rho = 0.02$ and $\rho = 50$.

Figure 3.4 demonstrates the effect of relatively smaller and larger values of $\rho$ on $h$. Decreasing the value of $\rho$ allows kernel machine regression to overfit to the noise in the data by relaxing the dependence of neighboring values of $h$ to each other. On the other hand, increasing the value of $\rho$ enforces more dependence in $h$ and as such results in an underfit estimation. Hence, the choice of $\rho$ has a strong effect on the performance of kernel machine regression.

### 3.2.4 Variable selection

Now that we have defined kernel machine regression, we can extend it to the Bayesian paradigm. Bobb et al. (2015) showed that the Bayesian approach can outperform frequentist kernel machine regression because simultaneous variable selection and estimation can better capture the exposure-response relationship. In this section, we discuss the two methods for Bayesian variable selection in BKMR: hierarchical variable selection and component-wise variable selection (Bobb et al., 2015).

In order to perform variable selection, we define a parameter that puts a weight on each exposure. Each weight controls the degree to which its associated exposure contributes to the model. In component-wise selection, we do this by augmenting the kernel function as

$$k(\mathbf{x}, \mathbf{x}'|\mathbf{r}) = \exp\left\{ -\sum_{m=1}^{M} r_m (x_m - x'_m)^2 \right\},$$

where $\mathbf{r} = [r_1, \ldots, r_M]^\top$. We define $r_m = 1/\rho_m$, the inverse of the tuning parameter $\rho_m$ for each $\mathbf{x}_m$. Now, we can imagine that an exposure that is not closely associated with the response will be be assigned a value of $r_m$ close to 0, which corresponds to a larger value of $\rho_m$. This larger value of $\rho_m$ means that this exposure would contribute less to the exposure-response relationship, as depicted in the second panel of Figure

3.4.

We now define the kernel matrix $\mathbf{K}_{\mathbf{X},\mathbf{r}}$ as the $n \times n$ matrix with $(i, j)$th element $k(\mathbf{x}, \mathbf{x}'|\mathbf{r})$. To allow $r_m$ to equal 0 with non-zero probability, we first define an indicator variable determining whether or not a predictor is included in the variable, which is distributed as

$$\delta_m \sim \text{Bernoulli}(\pi),$$

where $\pi$ is the prior probability of inclusion. Now, we can assume a "slab-and-spike" prior on $r_m$, distributed as

$$r_m|\delta_m \sim \delta_m f(r_m) + (1 - \delta_m)P_0,$$

where $f(\cdot)$ is some pdf with support $\mathbb{R}^+$, and $P_0$ denotes the density with point mass at 0.

While this process of component-wise variable selection works well in a typical multiple regression setting, it can lead to unreliable estimates in situations where the exposures are highly correlated with each other, which is common in exposure mixture studies. In this case, the correlated components contribute similar information to the model, and component-wise variable selection is not able to distinguish which exposure is important. BKMR deals with this problem by introducing hierarchical variable selection.

Hierarchical variable selection involves partitioning the predictors $\mathbf{x}_1, \ldots, \mathbf{x}_M$ into $G$ groups, denoted $\mathcal{S}_g$ with $g = 1, \ldots, G$. These groups should be selected by the user based on prior knowledge, with the aim of keeping within-group correlation high and between-group correlation low. For instance, consider a situation with 4 chemicals, Hg, Pb, As, and Sn. If Hg, Pb, and As were strongly correlated with each other and each weakly correlated with Sn, we might define $\mathcal{S}_1 = \{\text{Hg, Pb, As}\}$ and $\mathcal{S}_1 = \{\text{Sn}\}$.

The indicators from $r_m|\delta_m$ are now distributed as

$$\boldsymbol{\delta}_{\mathcal{S}_g}|\omega_g \sim \text{Multinomial}(\omega_g, \boldsymbol{\pi}_{\mathcal{S}_g}), g = 1, \ldots, G,$$

$$\omega_g \sim \text{Bernoulli}(\pi),$$

where $\boldsymbol{\delta}_{\mathcal{S}_g} = \{\delta_m|\mathbf{x}_m \in \mathcal{S}_g\}$ and $\boldsymbol{\pi}_{\mathcal{S}_g}$ are vectors of indicator variables and prior probabilities, respectively, of a exposure $\mathbf{x}_m$ in group $\mathcal{S}_g$ entering the model. By this approach, at most one exposure in each group is allowed to enter the model.

While hierarchical variable selection resolves the issue of multicollinearity, it requires specifying subgroups of predictors a priori and assumes that one exposure in each group can capture the information of the rest. Hence, care should be taken to justify the partitioning of exposures when taking this approach.

Note also that the posterior means of $\delta_m$ generated from these variable selection procedures represent the posterior probability of inclusion of $\mathbf{x}_m$. We can interpret these posterior inclusion probabilities (PIPs) as measures of the relative importance of each exposure These measures can be used to understand the contribution of each exposure to the health outcome of interest in the model.

### 3.2.5 Prior specification

In this section, we specify the default prior distributions and parameters used by the BKMR algorithm (Bobb et al., 2015).

BKMR, by default, assumes $\rho_m = 1/r_m \sim \text{Unif}(a_r, b_r)$, a flat prior between $a_r$ and $b_r$ for which the default values are 0 and 100, respectively (Bobb, 2017a). This defines the prior probability of $\rho$ as equally distributed across any value from 0 to 100. This inverse of this prior corresponds to the slab component of the "slab-and-spike" prior, where $r_m|\delta_m \sim \delta_m \text{Unif}^{-1}(a_r, b_r) + (1-\delta_m)P_0$. As a flat prior, this distribution should be chosen when we have no prior knowledge about the smoothness of the exposure-response function, with hyperparameters $a_r$ and $b_r$ selected to represent the range of values we expect $\rho$ to potentially span.

We have seen that the smoothness of a kernel machine regression responds strongly to different values of $r_m = 1/\rho$, and, accordingly, the model fit of BKMR is sensitive to their prior distribution. In general, the PIPs generated from the variable selection procedure are particularly sensitive to this prior, though their relative importance tends to remain stable (Bobb, Claus Henn, Valeri, & Coull, 2018). As such, the BKMR algorithm also offers the options to define uniform and gamma priors for the $r_m = 1/\rho$.

Moreover, BKMR assumes that the prior probability of including a predictor ($\delta_m$) or group of predictors ($\omega_g$) in the model is distributed $\pi \sim \text{Beta}(a_\pi, b_\pi)$. The default hyperparameters are $a_\pi = b_\pi = 1$, which represent a flat, uninformative prior between 0 and 1. When the hierarchical selection approach is applied, equal values for $\boldsymbol{\pi}_{\mathcal{S}_g}$, the probabilities of inclusion for each component in group $\mathcal{S}_g$, are assumed.

Finally, BKMR assumes that the inverse of the variance of the residuals is distributed $\sigma^{-2} \sim \text{Gamma}(a_\sigma, b_\sigma)$, with default values of $a_\sigma = b_\sigma = 0.001$, and that the vertical scale of $h$ is parameterized by $\lambda \equiv \tau\sigma^{-2} \sim \text{Gamma}(a_\lambda, b_\lambda)$, with default values of $a_\lambda, b_\lambda$ such that the mean and variance of $\lambda$ are both equal to 10.

### 3.2.6 The MCMC algorithm

Briefly, we discuss the algorithm used to find the solution in the BKMR package (Bobb et al., 2015, 2018), with commentary on its implications for the model fitting process.

BKMR uses a Markov chain Monte Carlo (MCMC) algorithm with a mix of Gibbs and Metropolis-Hastings samplers to estimate the posterior distributions of the parameters. In particular, a Gibbs step is used to update the distribution of $\sigma^2$ while a Metropolis-Hastings step is used to update the distribution of $\lambda$. For component-wise and hierarchical variable selection, $(\mathbf{r}, \boldsymbol{\delta}, \boldsymbol{\omega})$ are sampled jointly using a Metropolis-Hastings sampling scheme.

While each distribution generated by the Gibbs step is always accepted, the distributions for $\lambda$ and $r_m$ generated by the Metropolis-Hastings steps are accepted based on an acceptance rate (Wagaman & Dobrow, 2021). The standard deviation of the proposal distribution controls the acceptance rate and as such acts as a tuning parameter (Bobb, 2017b). In general, increasing the standard deviation leads to lower acceptance rates. Acceptance rates that are too low lead to slower convergence, but rates that are too high can cause convergence to a non-optimal distribution.

A major computational limitation of BKMR is that at each iteration of the MCMC algorithm, the $n \times n$ augmented kernel matrix $\mathbf{K}_{\mathbf{Z},\mathbf{r}}$ must be inverted multiple times. To offset this, BKMR can employ a Gaussian predictive process which involves specifying a set of $l$ points, or "knots," that are a subset of the predictor space. The vector of predictors can be approximated by projection onto this lower dimensional space, which allows the algorithm to perform inversions on an $l \times l$ matrix. A general suggestion is to use this approach to speed up the algorithm when $n$ is large and to specify $l \approx n/10$ (Bellavia, 2021).

## 3.3 Bayesian semiparametric regression (BSR)

In this section, we introduce the theory of BSR. First, we define the notation that we will be using for spline regression:

- $X_m$ is a predictor variable in the predictor matrix $\mathbf{X}$ with $m = 1, \ldots, M$, measuring exposure variables or covariates

- $\mathbf{x}_i$ is a vector of values for a single observation in $\mathbf{X}$ with $i = 1, \ldots, n$
- $\mathbf{z}_i$ is a vector of covariates for a single observation in the matrix $\mathbf{Z}$, which contains a set of covariates, with $i = 1, \ldots, n$
- $Y_i$ is an observation of $\mathbf{Y}$, measuring the health outcome in this case
- $f(\cdot)$ relates $\mathbf{x}_i$ to $Y_i$ by a set of basis functions, $b_j(X)$
- $\beta_j$ is a weight on the $j$th basis function
- $P$ is the order of the basis expansion
- $K$ is a set of $\xi_k$, $k = 1, \ldots, K$, interior knots defining $K + 1$ disjoint intervals, and
- $\boldsymbol{\varepsilon}_i \overset{\text{iid}}{\sim} N(0, \sigma^2)$ are the residuals of the response.

And, we define the notation that we will be using specific to BSR:

- $\widetilde{X}_m$ is a $d$-dimensional basis function expansion of $X_m$
- $\widetilde{X}_{m_1 m_2}$ is a $d^2$-dimensional basis expansion of the interaction between $X_{m_1}$ and $X_{m_2}$
- $f^{(h)}(\cdot)$, where $h = 1, \ldots, H$ are a set of functions that sum up to $f(\cdot)$
- $\boldsymbol{\zeta} = \{\zeta_{mh}\}$ is an indicator for whether the $m$th predictor is included in the $h$th function
- $\boldsymbol{\beta}_S^{(h)}$ is a vector of all the coefficients on the predictors in function $h$
- $\sigma_\beta^2$ is the prior variance on the coefficients, and
- $\boldsymbol{\Sigma}_\beta$ is a diagonal matrix with the variances of the multivariate slab prior, $\sigma^2\sigma_\beta^2$, on the diagonals.

### 3.3.1 Spline regression

We begin by introducing spline regression, with attention to its specific implementation in BSR. Spline regression is a semiparametric regression technique that can be used to capture non-linear effects. In this introduction, we follow the presentation of spline regression provided by Antonelli et al. (2020), with additional details and explanation from Hastie, Tibshirani, & Friedman (2009).

BSR uses spline regression to define the regression relationship as

$$Y_i = f(\mathbf{x}_i) + \mathbf{z}_i^\top \boldsymbol{\beta}_\mathbf{z} + \varepsilon_i,$$

where $f$ is defined by a set of basis functions on the exposures, $\mathbf{x}_i$, $\mathbf{z}_i$ and $\boldsymbol{\beta}_\mathbf{z}$ are the covariates and their associated weights, and $\varepsilon_i$ is a random variable from $\boldsymbol{\varepsilon} \overset{\text{iid}}{\sim} N(0, \sigma^2)$. BSR uses natural spline bases. In order to understand how these are constructed, we start with a broad definition of basis expansions, before exploring linear, cubic, and then natural spline bases.

We determine the basis expansion by considering a piece-wise function of $X_m$ with some order $P$ and some set of $K$ knots defining $K + 1$ disjoint intervals. BSR places knots at uniformly sized quantiles within the boundaries of $X_m$. The most commonly used orders are $P = 1, 2, \text{and } 4$, the constant, linear, and cubic splines, respectively. To begin, let us consider a continuous piece-wise linear spline basis (i.e., $P = 2$) of a one-dimensional $X$ with two interior knots. In this case, we use the following four basis functions:

$$b_1(X) = 1, \qquad b_2(X) = X, \qquad b_3(X) = (X - \xi_1)_+, \qquad b_4(X) = (X - \xi_2)_+,$$

where $\xi_1$ and $\xi_2$ are the two interior knots, and $t_+$ denotes the positive part. These bases are used to construct the regression model $f(X) = \sum_{j=1}^{4} \beta_j b_j(X)$, which requires estimating $K + P = 4$ parameters. We can check the continuity restrictions at the knots by seeing that $f(\xi_1^-) = \beta_1 + \xi_1\beta_2$ and $f(\xi_1^+) = \beta_1 + \xi_1\beta_2 + (\xi_1 - \xi_1)\beta_3$ are equal, and likewise at the second knot.

Now, in the case of exposure mixtures, we want smoother functions that can capture the non-linear relationship between the response and the predictors. We can achieve this by increasing the order to $P = 4$ and using a cubic spline, with continuous first and second derivatives at the knots. The cubic spline is the lowest-order spline for which knot-discontinuity cannot be detected by the human eye. For example, for one $X$ with two interior knots, we use the following six basis functions:

$$\begin{gathered} b_1(X) = 1, \qquad b_2(X) = X, \qquad b_3(X) = X^2, \\ b_4(X) = X^3, \qquad b_5(X) = (X - \xi_1)^3_+, \qquad b_6(X) = (X - \xi_2)^3_+. \end{gathered}$$

Now, the regression model is defined as $f(X) = \sum_{j=1}^{6} \beta_j b_j(X)$ and requires estimating $K + P = 6$ parameters. It can be shown that $f'(\xi_i^-) = f'(\xi_i^+)$ and $f''(\xi_i^-) = f''(\xi_i^+)$, and so forth.

However, the behavior of polynomials near the boundaries of $X$, where there is less information, can be erratic. Natural cubic splines, also referred to as just natural splines, address this by imposing an additional restriction of linearity at the boundaries of $X$. Paradoxically, this also leads to a simpler model with four fewer parameters to estimate. A general definition of the $K$ basis functions for a natural spline with interior knots $\xi_j$, $j = 1, \ldots, K$, is given by:

$$\begin{gathered} b_1(X) = 1, \qquad b_2(X) = X, \qquad b_{k+2}(X) = d_k(X) - d_{K-1}(X), \\ d_k(X) = \frac{(X - \xi_k)^3_+ - (X - \xi_K)^3_+}{\xi_K - \xi_k}. \end{gathered}$$

Here, the regression model is defined as $f(X) = \sum_{j=1}^{K} \beta_j b_j(X)$, with $K$ parameters. BSR uses natural splines to specify the regression relationship.

### 3.3.2 Toy example

In the following section, we illustrate spline regression using the same toy example used to introduce kernel machine regression. See Section 3.2.3 and Figure 3.1 for details on the parameters used to generate simulated data.

As in Section 3.2.3, we consider a case where we want to model the relationship between a single exposure and a response variable, where the true relationship between $x$ and $Y$ is defined as $Y = e^{\frac{x}{10}} + 2\sin(\frac{x}{2})$. We fit a series of linear, cubic, and then natural spline regressions to illustrate the general framework of a natural spline regression.

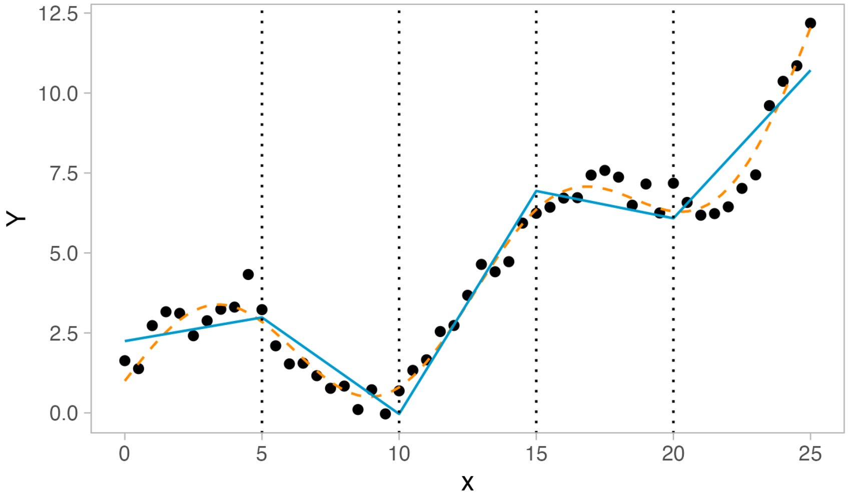

Figure 3.5: Linear spline regression (blue) with four knots (dotted lines) compared to the true relationship (orange).

Figure 3.5 illustrates the fit proposed by a linear spline regression with order $P = 2$. We can see that the implementation of knots allows for even a linear fit to capture more of the nuances in this nonlinear relationship, as compared to a standard linear regression.

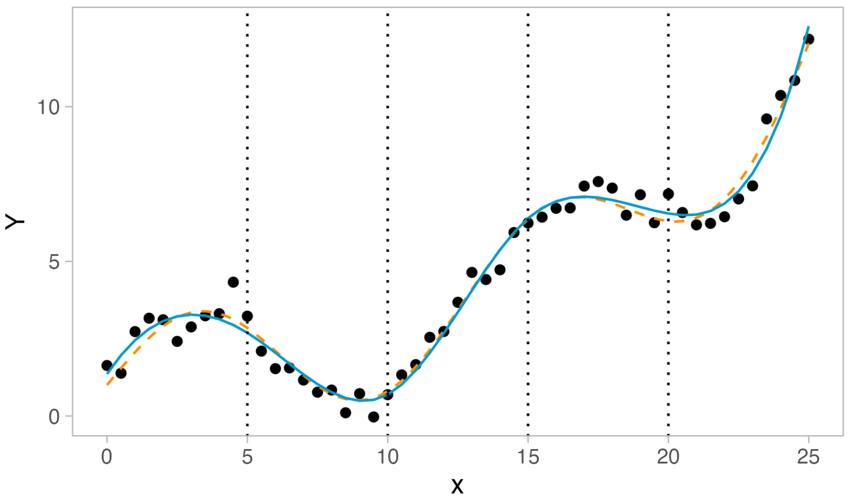

Figure 3.6: Cubic spline regression (blue) with four knots (dotted lines) compared to the true relationship (orange).

However, this linear spline regression is still unable to fully estimate the nonlinearity in our example. Increasing the order to $P = 4$ with a cubic spline regression offers additional flexibility. Figure 3.6 illustrates the fit proposed by this model. Here, we can see the benefits of using a cubic polynomial relationship in a nonlinear setting: the estimated relationship is continuous at the knots, and the nonlinear relationship has been flexibly captured.

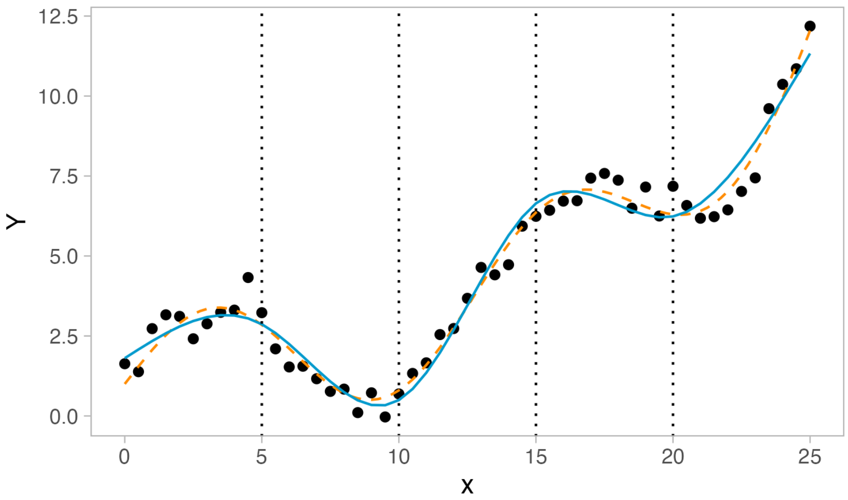

Figure 3.7: Natural spline regression (blue) with four knots (dotted lines) compared to the true relationship (orange).

Our final modification involves imposing linearity constraints on the boundaries of $x$ to implement a natural spline regression. Figure 3.7 shows that the fit estimated using a natural spline regression. The fitted line is only slightly different than that proposed by a cubic spline regression in Figure 3.6. The most noticeable difference is that the slopes of the tails of the natural spline regression are less extreme than for the cubic spline regression.

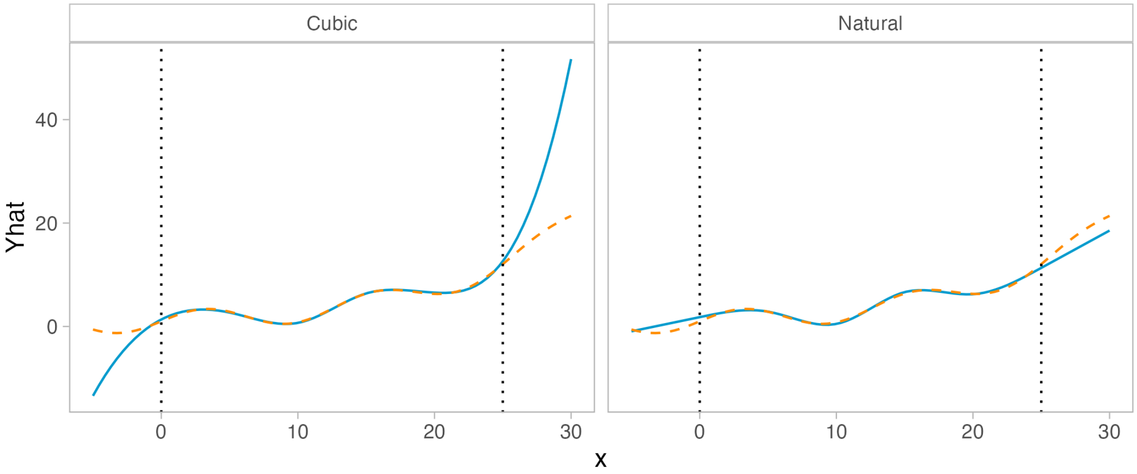

Figure 3.8: Natural and cubic spline regression (blue) compared to the true relationship (orange) extrapolated outside the bounds of $x$ (dotted lines).

Extrapolation outside the scope of $x$ allows us to see the effect of the linearity constraints imposed by natural regression. Figure 3.8 demonstrates how the cubic spline regression behaves erratically outside the bounds of $x$, as the cubic polynomial lines tend toward $\pm\infty$, while the natural spline regression follows a more appropriate linear trend. As the natural spline regression is more reliable near the boundaries of $x$ and also simpler to estimate, BSR adopts the use of natural splines.

### 3.3.3 Model formulation in BSR

Now that we have defined natural splines, we introduce BSR, following the presentation in Antonelli et al. (2020). We demonstrate the construction of $f$ in BSR by first assuming a two-dimensional case with exposures $X_1$ and $X_2$. We define

$$f(X_1) = \widetilde{X}_1\boldsymbol{\beta}_1, \qquad f(X_2) = \widetilde{X}_2\boldsymbol{\beta}_2,$$

$$f(X_1, X_2) = \widetilde{X}_1\boldsymbol{\beta}_1 + \widetilde{X}_2\boldsymbol{\beta}_2 + \widetilde{X}_{12}\boldsymbol{\beta}_{12},$$

where $\widetilde{X}_m = [b_{m1}(X_m), \ldots, b_{md}(X_m)]$ represents a $d$-dimensional basis function expansion for $m = 1, 2$, and $\widetilde{X}_{12} = [b_{11}(X_1)b_{21}(X_2), b_{11}(X_1)b_{22}(X_2), \ldots, b_{1d}(X_1)b_{2d}(X_2)]$ represents a $d^2$-dimensional basis expansion of the interaction between $X_1$ and $X_2$. $d$ is an influential tuning parameter. BSR by default assumes that all exposures have the same number of degrees of freedom and uses the Watanabe-Akaike (WAIC) model selection criterion to select $d$, which approximates leave one out cross validation. Note that we must explicitly model the effect of the interaction term by assuming a multiplicative interaction between the basis functions of the predictors.

Extending to the multi-dimensional setting, BSR assumes the following general model formulation:

$$f(\mathbf{x}_i) = \sum_{h=1}^{H} f^{(h)}(\mathbf{x}_i),$$
$$f^{(h)}(\mathbf{x}_i) = \sum_{m_1=1}^{M} \widetilde{x}_{im_1}\boldsymbol{\beta}^{(h)}_{m_1} + \sum_{m_1=2}^{M}\sum_{m_2<m_1} \widetilde{x}_{im_1m_2}\boldsymbol{\beta}^{(h)}_{m_1m_2} + \ldots,$$

where $f^{(h)}(\mathbf{x}_i)$ includes a summation of all $M$-way interactions. The inclusion of all $M$-way interactions makes the model far too overparameterized. Moreover, $f(\mathbf{x}_i)$ is a sum of $k$ different functions $f^{(h)}(\mathbf{x}_i)$ where a value for $H$ is selected in order to capture all exposure effects in the model. Each of the $H$ functions has the same functional form, and so the regression coefficients for a function $f^{(h)}(\mathbf{x}_i)$ are only identifiable up to a constant — this means that there are multiple sets of coefficients that could be estimated from the same data.

### 3.3.4 Sparsity inducing priors

In order to handle the overparameterization and non-identifiability of the model, BSR implements multivariate sparsity inducing priors. In this section, we follow the presentation provided in Antonelli et al. (2020).

First, we define indicators $\boldsymbol{\zeta} = \{\zeta_{mh}\}$ representing whether the $m$th exposure is included in the $h$th function:

$$P(\zeta_{mh} = 1) = \tau_h^{\zeta_{mh}}(1-\tau_h)^{1-\zeta_{mh}} I(A_h \not\subset A_{h'} \forall h' \neq h \text{ or } A_h = \{\}),$$
$$\text{where } A_h = \{m : \zeta_h = 1\}.$$

Here, the indicators follow a Bernoulli distribution with prior probability of inclusion $\tau_h$. The posterior means of $\boldsymbol{\zeta}$, i.e. the PIPs, can be interpreted as measures of relative variable importance. We include an indicator function $I()$ that represents whether the function $h$ contains a unique set of predictors. This indicator ensures that no function contains exposures that are a subset of those in another function, $h'$, in which case this function would be redundant and thus removed from the model entirely.

Now, we assume a multivariate slab-and-spike prior on the regression coefficients:

$$P(\boldsymbol{\beta}_S^{(h)}|\boldsymbol{\zeta}) = \left(1 - \prod_{m \in S} \zeta_{mh}\right) P_{\mathbf{0}} + \left(\prod_{m \in S} \zeta_{mh}\right) \psi_1(\boldsymbol{\beta}_S^{(h)}),$$
$$\text{where } S \text{ is some subset of } 1, 2, \ldots, m.$$

Here, $P_{\mathbf{0}}$ denotes the density with point mass at $\mathbf{0}$, and $\psi_1()$ is a multivariate normal distribution with mean $\mathbf{0}$ and covariance $\boldsymbol{\Sigma}_\beta$, a diagonal matrix with $\sigma^2\sigma_{\boldsymbol{\beta}}^2$ on the diagonals.

### 3.3.5 Prior specification

In this section, we discuss the priors and their default specifications in BSR (Antonelli et al., 2020).

The priors on $\boldsymbol{\Sigma}_\beta$, the diagonal matrix with $\sigma^2\sigma_{\boldsymbol{\beta}}^2$ on the diagonals, control the shrinkage of $\boldsymbol{\beta}_S^{(h)}$. Variable selection is sensitive to the choice of prior on this parameter, so BSR implements an empirical Bayes strategy to obtain a prior distribution for $\sigma^2_\beta$ based on the data. While this is not a fully Bayesian approach, it has been shown that this strategy works better in practice (Antonelli et al., 2020). Additionally, the default prior for $\sigma^2$ is assumed to follow a $\text{Gamma}(0.001, 0.001)$ distribution.

However, when there is a weak relationship between the exposures and relationship, the estimated prior variance for the slab $\sigma^2_\beta$ can be very small. In this case, the shape of the slab approximates the point mass of 0 at the spike, and the PIPs become difficult to accurately estimate. BSR avoids this by imposing a lower bound on the variance. This is determined by establishing a constant value for $\tau_h$, the prior probability of inclusion, for all $h$ and then permuting the rows of $Y$ (i.e., breaking up the relationship). Then, a grid of values for $\sigma^2_\beta$ are tested until some predefined threshold of the posterior probability of inclusion is obtained (e.g., 0.25 for a main effect and 0.05 for a two-way interaction). If the empirical Bayes estimate for $\sigma^2_\beta$ is less than this lower bound, then the lower bound is used instead.

Finally, BSR assumes that $\tau_h \sim \text{Beta}(L, \gamma)$, which defines the prior probability of including a predictor for all functions $h$. If $L$ is some predefined constant, and $\gamma = m$, the number of predictors, then the prior amount of sparsity should increase as the number of predictors increases (Antonelli et al., 2020).

### 3.3.6 The MCMC algorithm

We also briefly discuss the MCMC algorithm employed by BSR (Antonelli et al., 2020).

BSR uses an MCMC algorithm to obtain posterior distributions of $\sigma^2$ and $\tau_h$. In particular, Gibbs samplers are employed to sample $\sigma^2$ and $\tau_h$ from their full distributions and to update $\boldsymbol{\zeta}$ and $\boldsymbol{\beta}_S^{(h)}$. Every $T$ MCMC iterations, BSR uses a Monte Carlo expectation maximization algorithm with a Gibbs sampler to update $\sigma^2_\beta$. The empirical Bayes estimate is obtained once $\sigma_\beta^2$ converges, at which point the MCMC runs conditional on this estimated variance.

Notably, this algorithm must deal with the explicit specification of interaction terms in the model. Any additive univariate effect or lower-order interaction term is, by definition, a subset of some higher-order interaction term. As the MCMC algorithm searches the model space, it might accept a move to a higher-order interaction and get stuck in a local mode when, in reality, a simpler model should be preferred. BSR handles this challenge by imposing a constraint in the MCMC algorithm: if the inclusion of a $p$th order interaction term is being considered, then the algorithm must also evaluate all $(p-1)$th order models. If the truth is some lower-order model, then this strategy avoids the undesirable convergence to a local mode. When the model is complex, maintaining reversibility of updates under this strategy can be computationally challenging with a Gibbs sampler; in this case, using a Metropolis Hastings sampler is computationally faster (Antonelli et al., 2020).

## 3.4 Detecting interactions

In Section 3.1, we highlighted the challenges of analytically testing for the presence of interactions in exposure mixture studies. These challenges motivated a theoretical exploration of BKMR and BSR in Sections 3.2 and 3.3. Now, we discuss and compare the options that BKMR and BSR provide for inference on the presence of interactions. We also include discussion on theoretical advantages and disadvantages to each.

### 3.4.1 BKMR

Since the flexible $h$ function in kernel machine regression allows us to forgo any assumptions about the nature of the relationship between the health outome and exposures, BKMR can potentially capture complex interactions between exposures The challenge with using BKMR to do this, however, is that there is no formal framework for conducting inference on the presence of interactions.

Currently, the most common approach to detecting interactions is through a qualitative assessment of visual diagnostic plots (Bobb, 2017a). Two- or three-way interactions can be assessed by plotting the estimated exposure-response relation for one/two exposures at various quantiles of another exposure, while setting all other exposures at fixed quantile values. For instance, if we are interested in the interaction between $X_1$ and $X_2$, we can plot the estimated regression line against $X_1$ at the 0.25, 0.5, and 0.75 quantiles of $X_2$ and vice versa. In the three-way case of $X_1$, $X_2$, and $X_3$, we can plot the estimated regression surface against $X_1$ and $X_2$ at the 0.25, 0.5, and 0.75 quantiles of $X_3$. If the shape of the estimation changes meaningfully, then there might be evidence of an interaction.

A slightly more formal inferential approach for two-way interactions involves using summary statistics (Bobb, 2017a, 2017b). In this case, we can calculate the difference in estimated response values for $X_1$ at two quantiles, say, 0.25 and 0.75, of $X_2$ and then generate a confidence interval. If we observe that the interval does not contain 0, then there is evidence of an interaction. The choice of quantiles here is important. If there is a parachute-like regression surface between the response and two exposures, the summary statistics might mask the true nature of the relationship.

Notably, if we specify hierarchical variable selection to handle multicollinearity, then only one exposure in each a priori defined group can enter the model. If there exists some true interaction between exposures in one group, then BKMR will be unable to incorporate it into the final model. Moreover, interactions between exposures in separate groups can only be identified if both are selected into the final model based on their within-group PIPs. Hence, if detecting interactions is a goal when using BKMR with hierarchical variable selection, groups should be carefully selected, and the influence of group membership should be considered in model interpretation.

### 3.4.2 BSR

Providing formal inference on the presence of interactions was one of the primary motivations for the development of BSR. BSR explicitly incorporates interaction terms in its model formulation, and the model fitting process assigns PIPs for any $m$-dimensional interaction from the posterior means of the $\boldsymbol{\zeta}$ matrix. Such probabilities can be used as a quantifiable measure of the strength of a potential interaction. We can also compare PIPs for interactions with the PIPs for their individual components, which can be used to compare exposures' interactive effects with their marginal effects.

The visualization and summary statistics approaches available in BKMR are also possible in BSR. In particular, it is helpful to follow up the identification of a potential interaction with a visual assessment using the estimated exposure-response relationship at fixed quantiles of other predictors. The major benefit of BSR is that the PIPs serve as a quantifiable uncertainty metric for the presence of interactions.

### 3.4.3 Differences between BKMR and BSR

We have explained how the inferential framework of BSR improves upon the BKMR approach for the detection of complex interactions. We also briefly compare general features of their model formulations.

While BKMR is a fully nonparametric approach, BSR is a semiparametric approach because it makes distributional assumptions about the data (i.e., that the relationship can be adequately captured by a $d$-dimensional natural spline basis expansion). As BKMR uses the kernel technique, its implementation can become computationally intensive for datasets with large $n$, as it scales with $n^2$, while BSR is able to scale with $n$.

BSR is highly sensitive to the choice of $d$, the degrees of freedom. While it employs a WAIC approach to selecting the best $d$, this parameter introduces an additional tuning step in the analysis. Both approaches can be highly sensitive to the specification of certain priors. BSR offers an empirical Bayes strategy for $\sigma_\beta^2$, the variance of a exposure's probability of inclusion. On the other hand, the choice of prior on the influential smoothing parameter, $\rho$, in BKMR is up to the user.

We also note that BKMR offers a hierarchical variable selection approach to dealing with collinearity between exposures. While this is an additional decision that must be specified by the user, it offers a formal approach to dealing with collinear exposures. BSR does not explicitly account for collinearity in its model formulation, which can lead to erroneous interpretations of variable importance if unaccounted for.

### 3.4.4 Exposure-covariate interactions

*add exposure-covariate interaction challenges*

# Chapter 4 Simulations

## 4.1 Past simulation studies

Here, we preface our simulation study with a brief overview of examples in the literature which compare various methods for exposure mixtures using simulations. Taylor et al. (2016) conclude that, in general for exposure mixture studies, no single method consistently outperforms others across all situations and, importantly, that a method should be chosen based on the question of interest. Thus, for each study, we highlight not only the findings, but also the data-generating scenarios and the identified question of interest.

Lazarevic et al. (2020) compare the performance of a broad range of methods for accurate variable selection of important exposures. They simulated exposure data using a multivariate copula based on real-world data and the response by specifying a regression relationship with only a subset of truly significant exposures and a normal error term. Two correlation structures were considered — one with the original Spearman correlation matrix and one with the values halved — as well as two signal-to-noise ratios — one with an $R^2$ for the true model at 10% and one at 30%. They found that BKMR, along with three other flexible regression methods that allow for nonlinearity, provided more accurate variable selection results compared to two machine learning methods. Moreover, they observed that, in general, low signal-to-noise ratios had a stronger impact on performance than did increasing multicollinearity.

Hoskovec et al. (2021) compare Bayesian methods, including BKMR, while considering 4 research questions: accurate estimation, selection of important exposures, exclusion of unimportant exposures, and identification of interactions. They use observed exposure and covariate data to simulate response data using regression relationships; they considered three exposure-response scenarios of varying complexity and included two-way multiplicative interaction terms. For each simulated dataset, they randomly assigned exposures to be active components of the mixture to incorporate variability in the data. Overall, they found that Bayesian methods outperformed traditional linear regressions, and that BKMR performed best when the exposure-response function takes on a complex form.

Most recently, Pesenti et al. (2023) compare BKMR, BSR, and the Bayesian Least Absolute Shrinkage and Selection Operator (LASSO) for variable selection. Data were generated using a multivariate normal with moderate and strong correlation structures specified manually by the researchers. They found that, in situations with additivity and linearity, Bayesian LASSO was appropriate. Across the other scenarios, BKMR generally performed best, while BSR selected exposures with high heterogeneity when the sample size was smaller due to the influence of the degrees of freedom, $d$, tuning parameter. Notably, multicollinearity did not generally lead to spurious variable selection.

Finally, we briefly comment on studies by Sun et al. (2013) and Barrera-Gómez et al. (2017), whose explicit goal is to compare methods for identifying interactions. Both studies generate exposure data using the correlation structure from an existing dataset; Sun et al. (2013) uses a multivariate lognormal, while Barrera-Gómez et al. (2017) uses a multivariate normal. Both only consider two-way, multiplicative interactions. While neither of these studies consider the methods used in this thesis, they find that, in general, models that formally allow for interaction effects perform better than models that only allow for univariate additive effects.

## 4.2 Methods

The goal of our simulation study is to provide guidance on the choice between BSR and BKMR for characterizing a diverse range of complex interactions between predictors. In particular, we aim to extend findings from previous simulation studies by considering a more comprehensive range of interaction types, including different effect sizes, non-multiplicative interactions, and three-way interactions. We also explore interactions between exposures and categorical covariates, a previously understudied form of interaction in exposure mixture studies.

### 4.2.1 MADRES data

In order to make our simulations comparable to real-world exposure mixture studies, we based our simulation data on the Maternal And Developmental Risks from Environmental and Social Stressors (MADRES) pregnancy cohort. The MADRES cohort is an ongoing, prospective pregnancy cohort of predominantly lower-income, Hispanic women in Los Angeles, California, which began in 2015 (Bastain et al., 2019). Urine samples were collected by participants at their first visit, and questionnaires were administered during their first visit, with follow-ups at the first, second, and third trimesters. See Bastain et al. (2019) for further details on study design.

Howe et al. (2020) previously examined the effect of prenatal metal mixtures of birth weight (BW) for gestational age (GA) in this cohort. They used BKMR to identify associations between metal mixtures and BW for GA, as well as BSR to conduct inference on interactions between metals. Briefly, using BKMR, they found that, of the metals in the mixture, Hg and Ni were most strongly associated with BW for GA. Moreover, BKMR results suggested that a potential interaction between Hg and Ni; however, when run through BSR, the PIP for this interaction was extremely small, despite being the highest of all two-way interactions.

Data from the study by Howe et al. (2020) were obtained from publicly available data in the Human Health Exposure Resource (HHEAR) Data Repository, which has been approved under Icahn School of Medicine at Mount Sinai IRB Protocol #16-00947. The Digital Object Identifiers associated with the urinary trace element data and epidemiological data are 10.36043/1945_159 and 10.36043/1945_177, respectively. All analyses were conducted in R v4.3.2 (R Core Team, 2013).

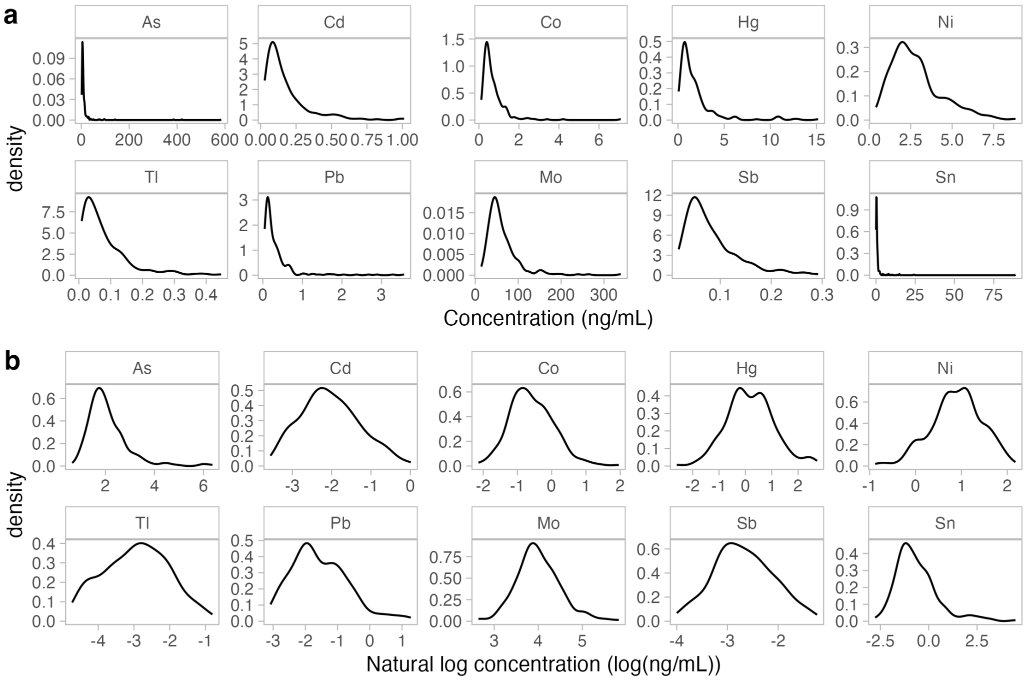

Figure 4.1: Distributions of original (a) and natural log transformed (b) concentrations of metals in MADRES cohort (n=252).

We followed the approach by Howe et al. (2020) for preparing the data for analysis. This resulted in retaining 10 metals in analysis: arsenic (As), cadmium (Cd), cobalt (Co), mercury (Hg), nickel (Ni), molybdenum (Mo), lead (Pb), antimony (Sb), tin (Sn), and thallium (Tl). Howe et al. (2020) used speciated As, but this was not available in HHEAR, so we used total As. Metals were expressed in nanograms per milliliter (ng/mL) and natural log transformed to reduce right-skewness (Figure 4.1). Among the full range of covariates considered by Howe et al. (2020), we used the subset of 4 that were available in HHEAR: any smoke exposure during pregnancy, maternal prepregnancy body mass index (BMI), maternal age during firt trimester, and maternal race by ethnicity and birth place. We chose not to include study site, as there was a study site with only 1 participant. Race by ethnicity and birth place was collapsed into the following categories: non-Hispanic white, non-Hispanic black, non-Hispanic other, Hispanic born in the US, and Hispanic born outside the US. We observed 8 missing values for BMI in the data from HHEAR, which were not reported by Howe et al. (2020). We mean imputed these missing values. Distributions of covariates are shown in Figure 4.2. Our final analytic dataset included 252 participants, which was 10 fewer than in Howe et al. (2020), likely due to small discrepancies in their dataset and the one made available in HHEAR.

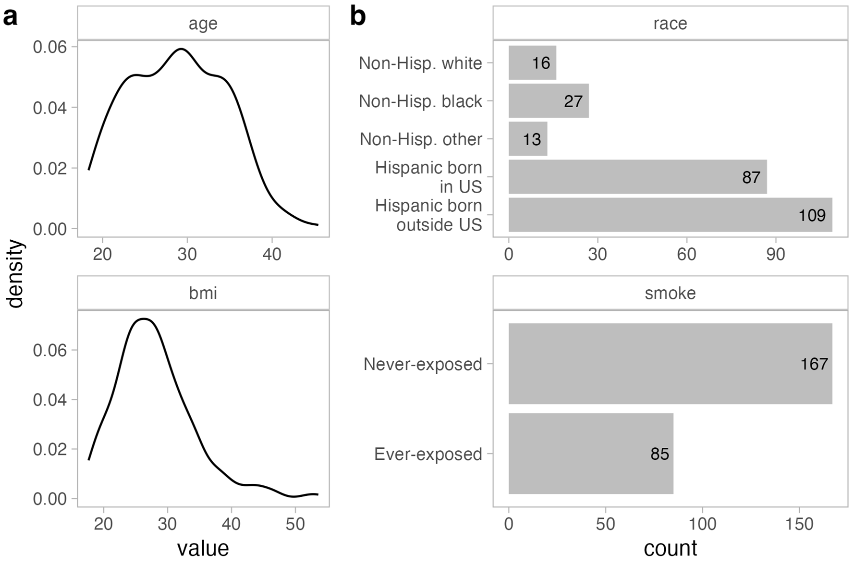

Figure 4.2: Distributions of continuous (a) and categorical (b) covariates in the MADRES cohort (n=252).

### 4.2.2 Using copulas to simulate predictor data

We simulated exposure and covariate data (hereafter referred to collectively as predictors) using a multivariate Gaussian copula fit on the 252 participants in the MADRES cohort. We used copulas as they can preserve both the correlation structure and marginal distributions from the observed data, allowing us to replicate conditions in a real-world scenario.

First, we briefly introduce copulas in the context of their use in this simulation, based on the presentation in Nelsen (2006). Copulas are joint cumulative distribution functions (CDFs) defined on the unit cube $[0, 1]^n$ that capture the dependence between $n$ uniformly distributed marginals. Sklar's theorem allows us to apply copulas to our observed data. Sklar's theorem states that, if $H(x_1, \ldots x_n)$ is a joint CDF of the marginal CDFs $F_1(x_1), \ldots, F_n(x_n)$, then there exists a copula $C$ such that, for all $(x_1, \ldots, x_n)$ in $(X_1, \ldots, X_n)$,

$$H(x_1, \ldots x_n) = C(F_1(x_1), \ldots, F_n(x_n)).$$

Note that, by the probability integral transform, or the universality of the uniform, the CDFs $F_1(x_1), \ldots, F_n(x_n)$ are distributed uniformly.

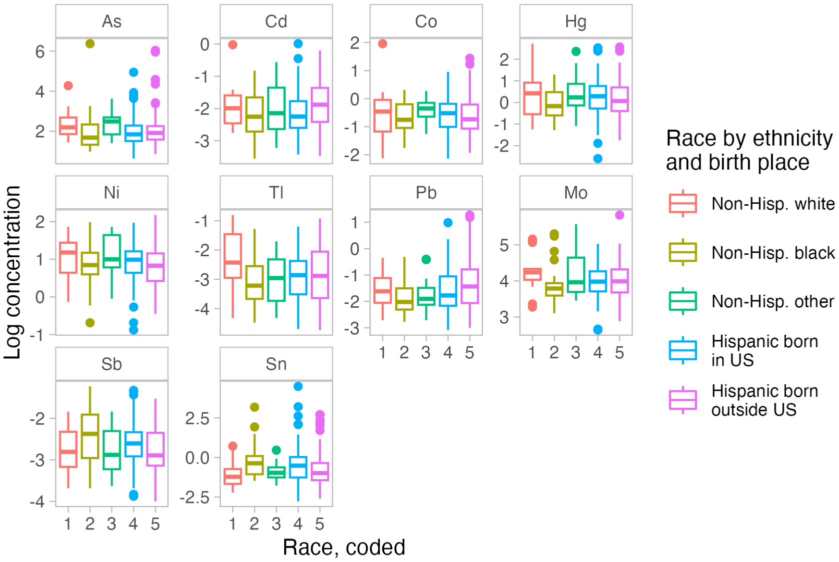

Figure 4.3: Association between race by ethnicity and birth place and metal exposures in the MADRES cohort (n=252).

We used the `copula` package in R to fit copulas and generate random data (Hofert, Kojadinovic, Maechler, & Yan, 2023). We transformed the observed continuous predictor values to uniform distributions based on their empirical marginal CDFs, a process called generating "pseudo-random" samples. We used the checkerboard copula approach for generating pseudo-random samples for smoke exposure, a binary variable (Genest & Nešlehovà, 2007). We coded smoke exposure as 0's and 1's, generated a pseudo-random sample, and then "jittered" the values with uniform random noise. There is currently no widely accepted approach for generating pseudo-random samples from unordered categorical variables with more than two levels. Thus, we excluded race by ethnicity and birthplace from the copula model. While this means that our simulated datasets did not preserve any potential association between race and exposures, Figure 4.3 suggests that there is little to no visible association between race and exposures in the observed dataset.

Various families of copulas have been described, each of which specifies a different shape for the dependence structure. We performed model selection to identify the copula that best approximates the dependence structure of our data. We fit the set of multivariate copulas used by Lazarevic et al. (2020) in their simulation study, which included the Gaussian, $t$, Gumbel, Frank, Clayton, and Joe copulas. We fit two $t$ copulas with 4 and 10 degrees of freedom, which controls dependence at the tails of the distributions, as well as a $t$ copula where the degrees of freedom was determined during the fitting process. The Gumbel, Frank, Clayton, and Joe copulas require a $\theta$ parameter, which controls dependence between the distributions. We fit two versions of these copulas with $\theta = \{2, 4\}$. Among these, the Gaussian copula minimized Akaike information criterion and maximized likelihood, so we proceeded with this model. The Gaussian copula assumes a bivariate normal dependence structure between the marginal CDFs.

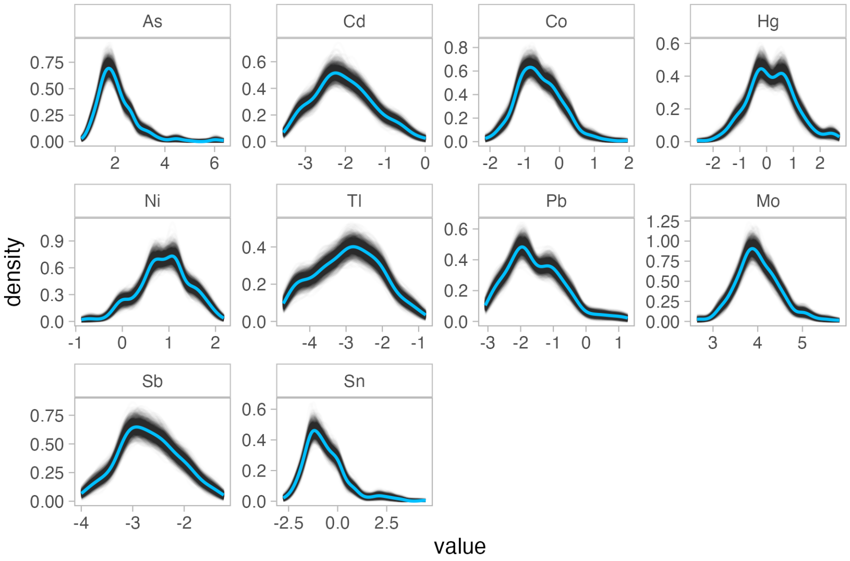

Figure 4.4: Distributions of exposures from observed data (blue) and simulated data (gray).

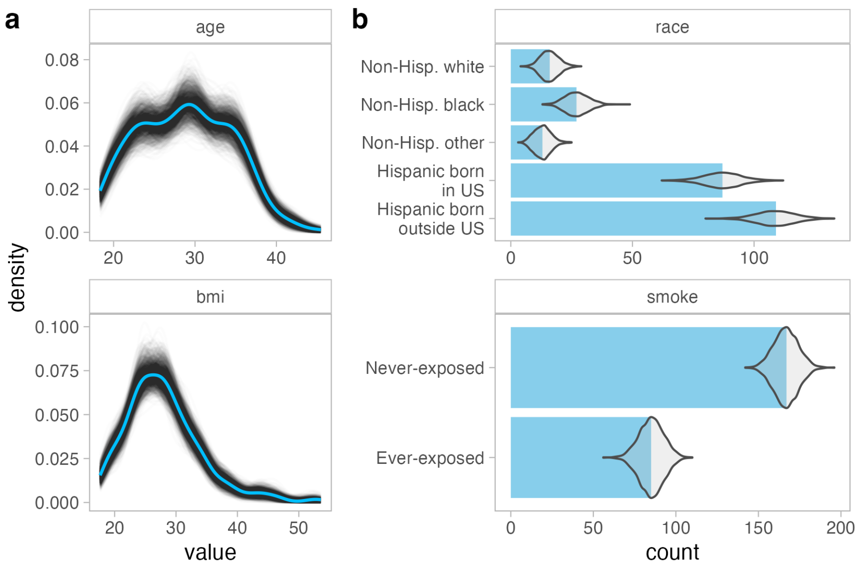

Figure 4.5: Distributions of covariates from observed data (blue) and simulated data (gray).

We simulated predictor data by randomly sampling from the fitted multivariate Gaussian copula distribution. All pseudo-random samples were then back-transformed to their original distributions using empirical marginal CDFs. We simulated the race by ethnicity and birthplace variable by randomly assigning observations to each of the five categories based on proportions in the observed dataset. We verified that the original structure of the observed dataset were preserved by visually comparing univariate distributions of exposures (Figure 4.4) and covariates (Figure @(ref:univcovsim)), as well as the correlation structure using Spearman's $\rho$ (Figure 4.6).

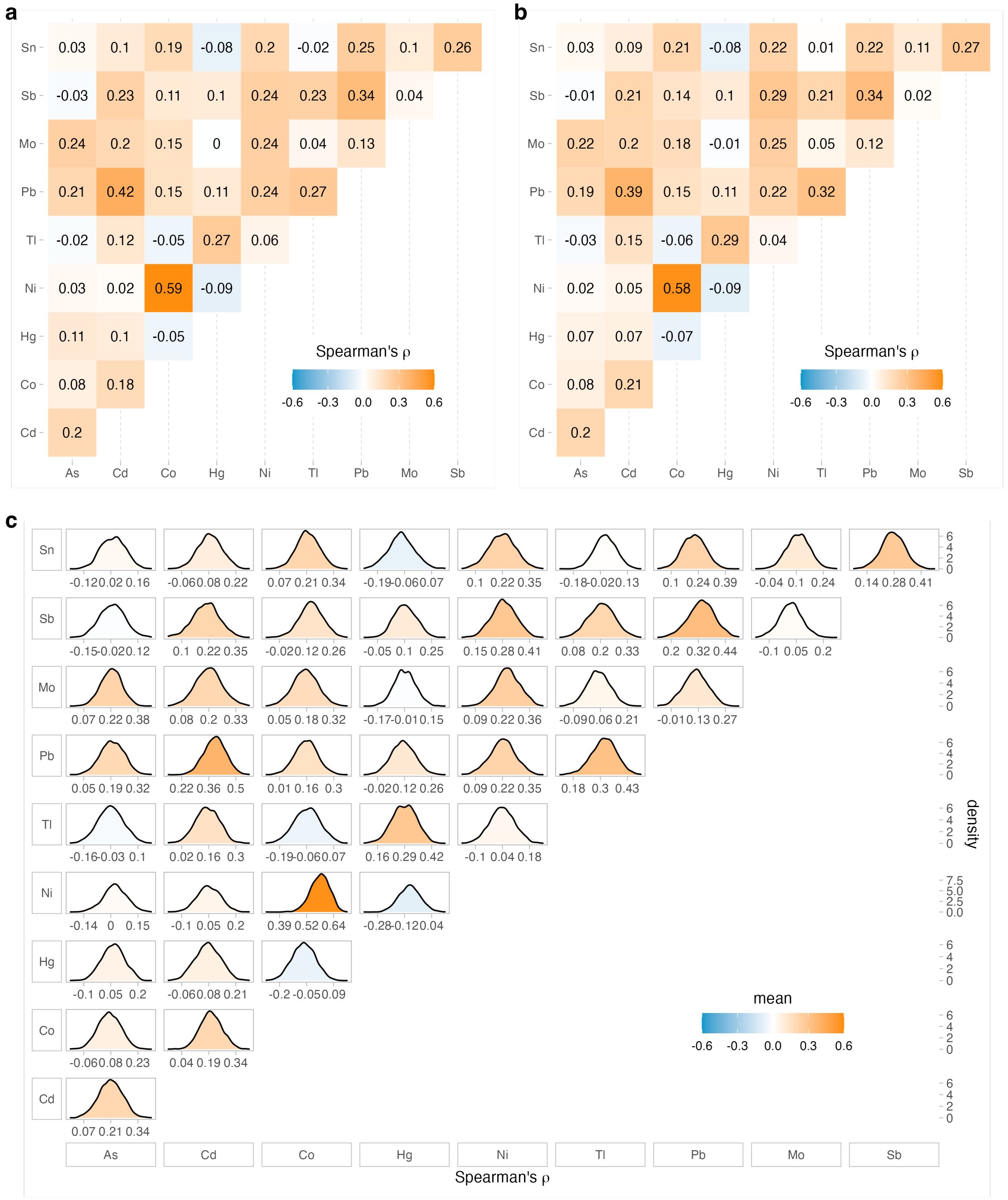

Figure 4.6: Correlation heat maps of exposures from observed data (a) and averaged from simulated data (b), as well as distribution of correlations from simulated data (c).

We generated one set of simulated datasets with the same sample size as the observed dataset (n=252), which is typical in many cohort studies. We also generated another set of simulated datasets with a larger sample size (n=1000), which has become increasingly common with the rise of larger-scale studies and big data. The goal of this choice is to inform sample size considerations in study design.

*include similar plots for larger dataset?*

### 4.2.3 Simulating predictor-response relationships

Health outcome responses were simulated under several different scenarios, each of which included different effect sizes and functional forms for the interactions, as well as different sample sizes. In the first scenario, we specified a "base case" model:

$$\begin{aligned} Y_i = \mathrm{Hg}_i &+ \frac{3}{1+\exp(-4\mathrm{Ni}_i)} + \frac{1.5}{1+\exp(-4\mathrm{Sn}_i)} - \mathrm{Sb}_i^2 + 0.5\mathrm{Sb}_i \\ &+ \mathrm{age} + 0.5\mathrm{bmi} + 0.5\mathrm{race}_{\mathrm{black}} + 0.5\mathrm{race}_{\mathrm{hisp.non}} + 1.5\mathrm{smoke} + \varepsilon_i, \end{aligned}$$

where $\varepsilon_i \overset{\text{iid}}{\sim} N(0,5)$. This model includes a linear term for Hg, two S-shaped logistic terms for Ni and Sn with varying effect sizes, and a symmetric inverse U-shaped quadratic term for Sb. Moreover, we included covariate terms as linear effects in the model. We chose the standard deviation on the normal random error term in order to achieve an $R^2$ of around 0.2-0.3 in a multiple linear regression including only the true functional form of the chemicals, which approximates realistic signal-to-noise ratios in exposure mixture studies (Lazarevic et al. (2020)).

In subsequent scenarios, we added an additional interaction term to the base case model.

The larger effect sizes were selected in order to achieve a power of approximately 0.5 at $\alpha = 0.05$ in a multiple linear regression with the true functional form of the chemicals specified, as well as with the covariate terms included. All cases:

Table 4.1: Specification of interaction terms in simulations.

| | Effect size | |
|---|---|---|
| | Small | Large |
| **Univariately significant** | | |
| Multiplicative | $0.35\text{Hg}*\text{Ni}$ | $0.7\text{Hg}*\text{Ni}$ |
| Polynomial | $0.3\text{Hg}*(\text{Ni}-1)^2$ | $0.26\text{Hg}*(\text{Ni}-1)^2$ |
| **Univariately insignificant** | | |
| Multiplicative | $0.35\text{Cd}*\text{As}$ | $0.7\text{Cd}*\text{As}$ |
| Polynomial | $0.125\text{Cd}*(\text{As}-1)^2$ | $0.25\text{Cd}*(\text{As}-1)^2$ |
| **Highly correlated** | | |
| Multiplicative | $0.3\text{Hg}*\text{Co}$ | $0.6\text{Hg}*\text{Co}$ |
| Polynomial | $0.09\text{Hg}*(\text{Co}-1)^2$ | $0.2\text{Hg}*(\text{Co}-1)^2$ |
| **Three-way interaction** | | |
| Multiplicative | $0.3\text{Hg}*\text{Ni}*\text{Tl}$ | $0.6\text{Hg}*\text{Ni}*\text{Tl}$ |
| Polynomial | $0.1\text{Hg}*(\text{Ni}-1)^2*\text{Tl}$ | $0.18\text{Hg}*(\text{Ni}-1)^2*\text{Tl}$ |

- 252 and 1000 sample sizes
- multiplicative and polynomial interaction
- small and large effect size
- two-way interactions between univariately significant, univariately insignificant, and highly correlated chemicals
- three-way interaction
- interaction between covariate and exposure
- see appendix for surfaces

This resulted in a total of 42 scenarios. For each scenario, we generated 100 simulated datasets, resulting in 4200 datasets.

### 4.2.4 Models

Software: Bobb et al. (2018) on CRAN, Antonelli et al. (2020) on GitHub

Models compared, specify the parameters for each (justify them!).

All metal concentrations were standardized to keep values scale-free.

- MLR
- MLR with known form of interactions specified (oracle method)
- BKMR with component-wise
- BSR

BKMR: Howe ran 200,000 Markov chain Monte Carlo (MCMC) iterations using the default priors. The first half of iterations was used as burn-in. To reduce potential autocorrelation, we thinned the chains, selecting every 25th iteration. - bobb recommends 50,000 iterations at least

check convergence with trace plots

### 4.2.5 Model assessment

- use median probability model threshold — marginal PIP of at least 0.5
- how many times is interaction picked up?
    - sensitivity and false discovery rate
- potentially explore mpower package

## 4.3 Results

- example output from representative model
- figures + tables w/ model performance

# Conclusion

If we don't want the conclusion to have a chapter number next to it, we can add the `{-}` attribute.

**More info**

And here's some other random info: the first paragraph after a chapter title or section head *shouldn't be* indented, because indents are to tell the reader that you're starting a new paragraph. Since that's obvious after a chapter or section title, proper typesetting doesn't add an indent there.

# Appendix A Supplemental output

## A.1 Methods

3d surfaces of interactions.

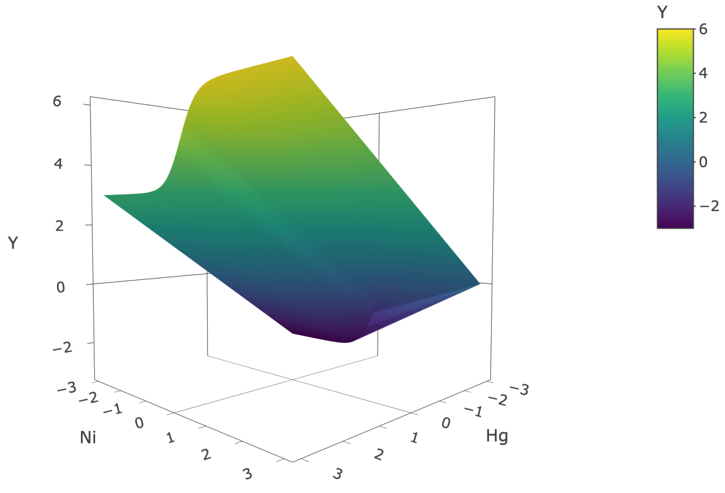

Figure A.1: Surface for base case.

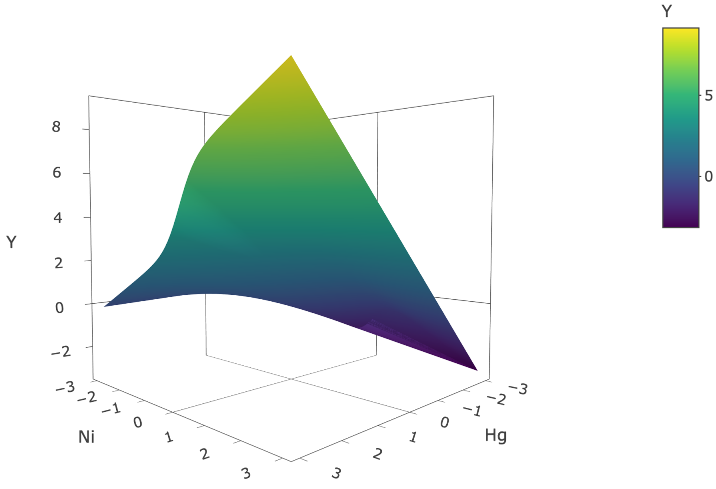

Figure A.2: Surface for Hg, Ni, mult low eff.

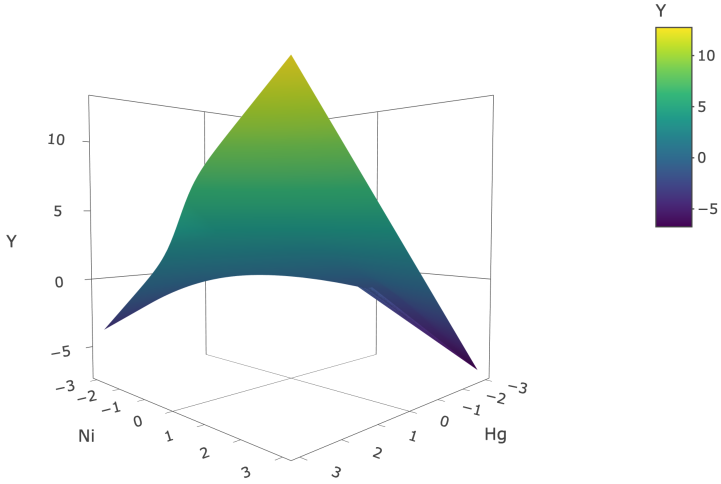

Figure A.3: Surface for Hg, Ni, mult high eff.

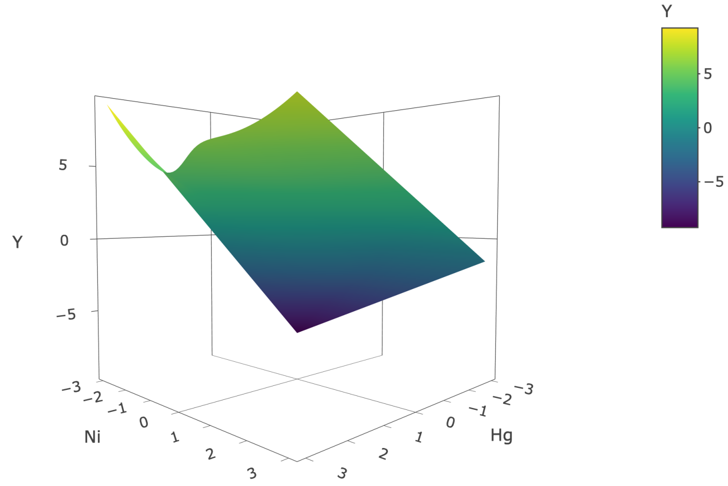

Figure A.4: Surface for Hg, Ni, poly low eff.

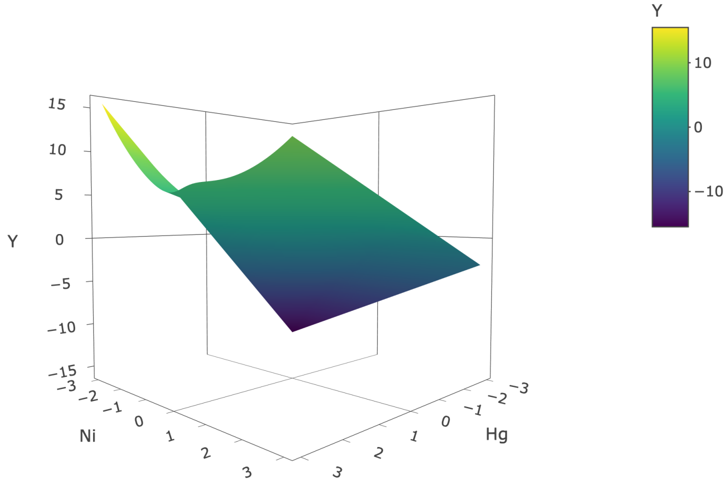

Figure A.5: Surface for Hg, Ni, poly high eff.

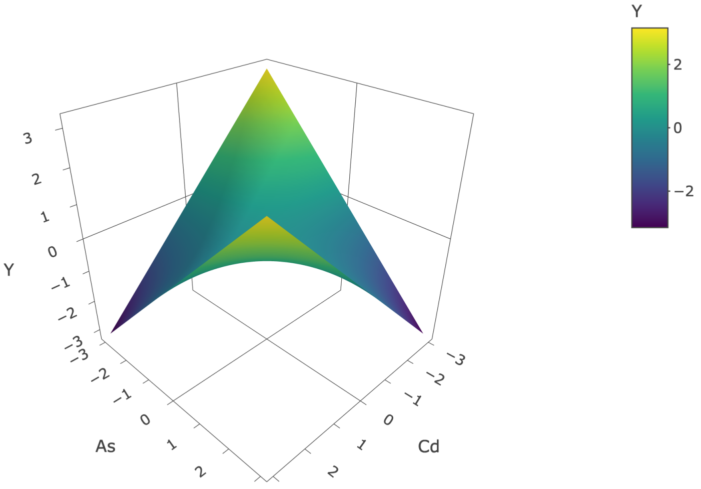

Figure A.6: Surface for Cd, As, mult low eff.

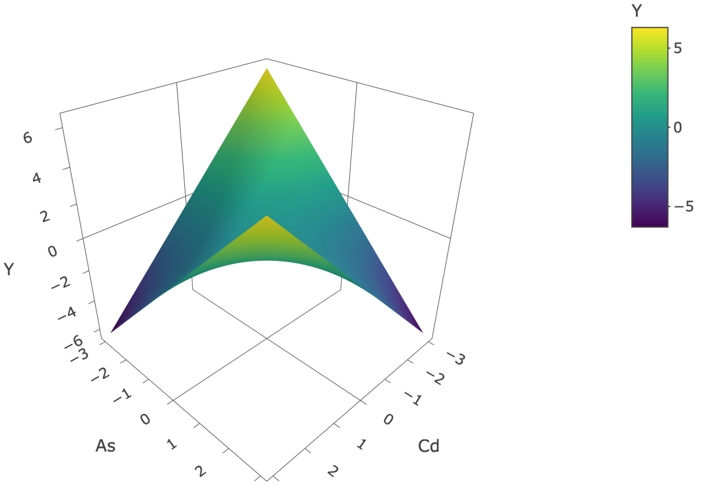

Figure A.7: Surface for Cd, As, mult high eff.

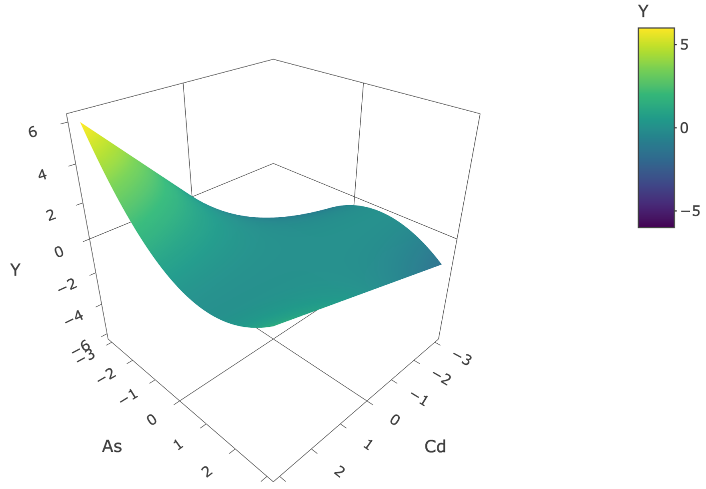

Figure A.8: Surface for Cd, As, poly low eff.

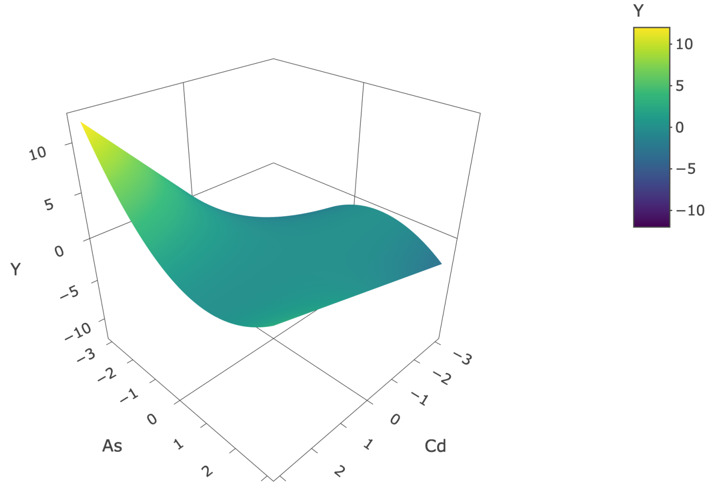

Figure A.9: Surface for Cd, As, poly high eff.

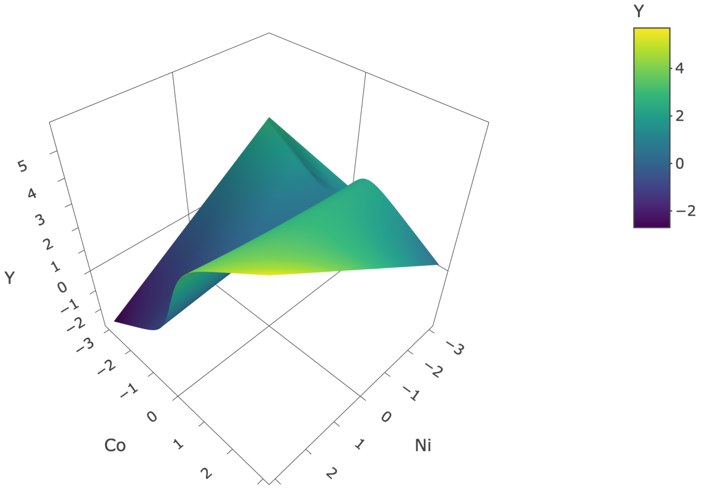

Figure A.10: Surface for Ni, Co, mult low eff.

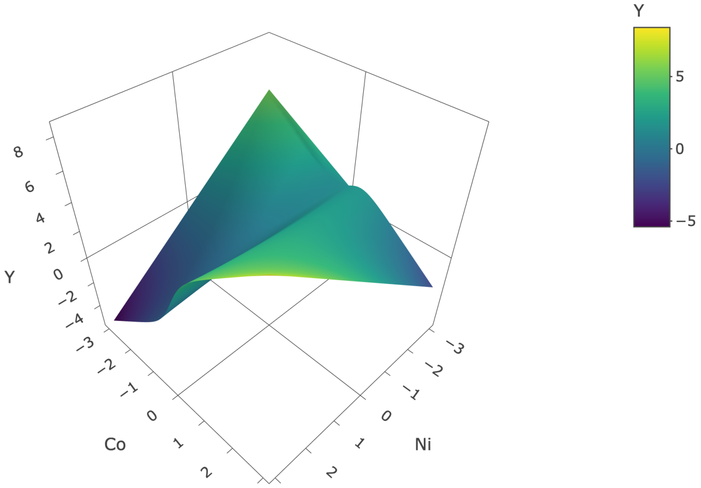

Figure A.11: Surface for Ni, Co, mult high eff.

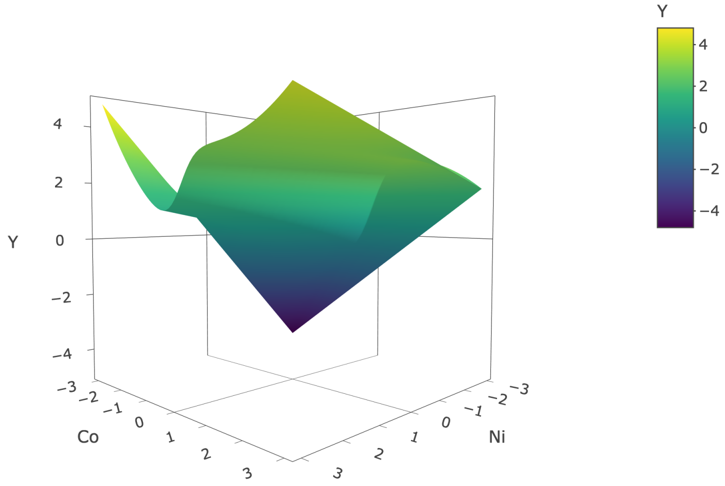

Figure A.12: Surface for Ni, Co, poly low eff.

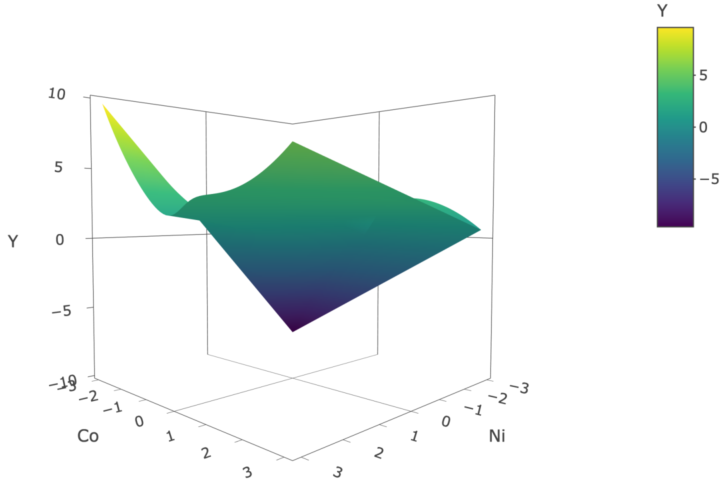

Figure A.13: Surface for Ni, Co, poly low eff.

## A.2 Results

Examples of trace plots for BKMR and BSR

# Appendix B Code

This second appendix includes all of the R chunks of code that were hidden throughout the document.

## B.1 Code for Chapter 3:

The code for this chapter generates a toy example, used to demonstrate the kernel machine regression and spline regression techniques.

```r
#load packages
library(tidyverse)
library(stats)
library(splines)
```

```r
# set theme for plots
theme_set(theme_light())
theme_update(panel.grid.major = element_blank(),
             panel.grid.minor = element_blank())
theme_update(
  strip.background = element_rect(color="gray", fill="white"),
  strip.text = element_text(color = "gray30")
  )
```

```r
########
# generate simulated points
########

# generate data from distribution
```

```r
set.seed(0) # reproducibility
x <- seq(0, 25, length.out = 51)
Y <- exp(x/10) + 2*sin(x/2) + rnorm(51, mean = 0, sd = 0.5)
df <- data.frame(x, Y)

# plot data and linear regression line
q1 <- ggplot(df, aes(x, Y)) +
  geom_point() +
  geom_function(fun = function(x) exp(x/10) + 2*sin(x/2),
                linetype = "dashed", color = "darkorange") +
  geom_smooth(method = "lm", formula = "y~x",
              color = "deepskyblue3", fill = "gray70",
              linewidth = 0.5, se = F)

# save plot
ggsave("index/figures/ch3_toy1.png", plot = q1, device = "png",
       width = 5, height = 3)
```

```r
########
# kernel regression
########

# get normal distribution of weights around query points
df$Weight <- dnorm(df$x, mean = 12.5, sd = 1)

# plot points colored by their weights
p1 <- ggplot(df, aes(x, Y)) +
  geom_point(aes(color = Weight)) +
  geom_function(fun = function(x) exp(x/10) + 2*sin(x/2),
                linetype = "dashed", color = "darkorange") +
  geom_vline(xintercept = 12.5, linetype = "dotted") +
  theme(legend.position = "none")

# plot a curve of weights
normcurv <- data.frame(x = seq(0, 25, length.out = 250))
normcurv$Weight <- dnorm(normcurv$x, mean = 12.5, sd = 1)
p2 <- ggplot(normcurv, aes(x, Weight, color = Weight)) +
  geom_line() +
  scale_y_continuous(breaks = c(0, 0.2, 0.4)) +
  theme(legend.position = "none")
```

```r
# stitch plots together
q2 <- cowplot::plot_grid(p1, p2, ncol = 1, rel_heights = c(0.7, 0.3))
q2

# save plot
ggsave("index/figures/ch3_toy2.png", plot = q2, device = "png",
       width = 5, height = 4)

# fit kernel regression with sigma = 1, bandwidth = 8/3
kmr_toy <- ksmooth(df$x, df$Y, kernel = "normal",
                   bandwidth = 8/3, x.points = df$x)
df <- df |>
  left_join(as.data.frame(kmr_toy), by = "x") |>
  rename(Yhat = y)

# plot kernel regression estimation
q3 <- ggplot(df) +
  geom_point(aes(x, Y)) +
  geom_function(fun = function(x) exp(x/10) + 2*sin(x/2),
                linetype = "dashed", color = "darkorange") +
  geom_line(aes(x, Yhat), color = "deepskyblue3")
q3

# save plot
ggsave("index/figures/ch3_toy3.png", plot = q3, device = "png",
       width = 5, height = 3)

# fit kernel regression with sigma = 5, bandwidth = 40/3
kmr_toy_5 <- ksmooth(df$x, df$Y, kernel = "normal",
                     bandwidth = 40/3, x.points = df$x)

# fit kernel regression with sigma = 0.1, bandwith = 8/30
kmr_toy_1 <- ksmooth(df$x, df$Y, kernel = "normal",
                     bandwidth = 8/30, x.points = df$x)

# re-join data
dfrho <- df |>
  left_join(as.data.frame(kmr_toy_5), by = "x") |>
  rename("rho = 50" = y) |>
  left_join(as.data.frame(kmr_toy_1), by = "x") |>
  rename("rho = 0.02" = y) |>
  select(-Yhat) |>
  pivot_longer(cols = c("rho = 50", "rho = 0.02"), values_to = "Yhat")

# plot kernel regression with two values of rho
qrho <- ggplot(dfrho) +
  geom_point(aes(x, Y)) +
  geom_line(aes(x, Yhat), color = "deepskyblue3") +
  facet_wrap(~name)
qrho

# save plot
ggsave("index/figures/ch3_toyrho.png", plot = qrho, device = "png",
       width = 7, height = 3)
```

```r
########
# spline regression
########

kn <- c(5, 10, 15, 20) # 4 knots of equal width

# fit linear spline regression
spline_toy_line <- lm(Y ~ bs(x, knots = kn, degree = 1), data = df)
p_line <- predict(spline_toy_line, se = T)
df$Yhats_line <- p_line$fit

q4 <- ggplot(df) +
  geom_point(aes(x, Y)) +
  geom_function(fun = function(x) exp(x/10) + 2*sin(x/2),
                linetype = "dashed", color = "darkorange") +
  geom_line(aes(x, Yhats_line), color = "deepskyblue3") +
  geom_vline(xintercept = kn, linetype = "dotted")
q4

# save plot
ggsave("index/figures/ch3_toy4.png", plot = q4, device = "png",
       width = 5, height = 3)

# fit cubic spline regression
spline_toy_cub <- lm(Y ~ bs(x, knots = kn), data = df)
p_cub <- predict(spline_toy_cub, se = T)
df$Yhats_cub <- p_cub$fit
```

```r
# plot spline regression estimation
q5 <- ggplot(df) +
  geom_point(aes(x, Y)) +
  geom_function(fun = function(x) exp(x/10) + 2*sin(x/2),
                linetype = "dashed", color = "darkorange") +
  geom_line(aes(x, Yhats_cub), color = "deepskyblue3") +
  geom_vline(xintercept = kn, linetype = "dotted")
q5

# save plot
ggsave("index/figures/ch3_toy5.png", plot = q5, device = "png",
       width = 5, height = 3)

# fit natural spline regression
spline_toy_nat <- lm(Y ~ ns(x, knots = kn), data = df)
p_nat <- predict(spline_toy_nat, se = T)
df$Yhats_nat <- p_nat$fit

# plot spline regression estimation
q6 <- ggplot(df) +
  geom_point(aes(x, Y)) +
  geom_function(fun = function(x) exp(x/10) + 2*sin(x/2),
                linetype = "dashed", color = "darkorange") +
  geom_line(aes(x, Yhats_nat), color = "deepskyblue3") +
  geom_vline(xintercept = c(5, 10, 15, 20), linetype = "dotted")
q6

# save plot
ggsave("index/figures/ch3_toy6.png", plot = q6, device = "png",
       width = 5, height = 3)

# see what happens outside of the bounds
x_longer <- seq(-5, 30, length.out = 81)
y_longer_cub <- predict(spline_toy_cub,
                        newdata = data.frame(x = x_longer))
y_longer_nat <- predict(spline_toy_nat,
                        newdata = data.frame(x = x_longer))

df_longer <- data.frame(
  x = c(x_longer, x_longer),
  spline = c(rep("Cubic", 81), rep("Natural", 81)),
  Yhat = c(y_longer_cub, y_longer_nat)
)

# plot outside of bounds
qbounds <- ggplot(df_longer) +
  geom_line(aes(x, Yhat), color = "deepskyblue3") +
  geom_function(fun = function(x) exp(x/10) + 2*sin(x/2),
                linetype = "dashed", color = "darkorange") +
  geom_vline(xintercept = c(0, 25), linetype = "dotted") +
  facet_wrap(~spline)
qbounds

# save plot
ggsave("index/figures/ch3_toybounds.png", plot = qbounds,
       device = "png", width = 7, height = 3)
```

## B.2 Code for Chapter 4:

The code for this chapter prepares the data from the MADRES study, generates simulated data, fits multiple linear regressions, BKMR, and BSR on the simulated data, and produces model output.

### B.2.1 Code for Chapter 4.2.1:

First, we clean the data from the MADRES study.

```
# load packages
library(tidyverse)
```

```
# read in data
target <- read_csv("madres_data/1945_TARGETED_DATA.csv")
epi <- read_csv("madres_data/1945_EPI_DATA.csv")
```

```
########
# clean target data
########

target_small <- target |>
  # if below LOD, use LOD / sqrt(2)
  mutate(conc_mod = ifelse(Comment_code == 37,
                           LOD / sqrt(2),
                           Concentration)) |>
  # adjust for urine specific gravity: Ac = A × [(SGmean -1)/(SG-1)]
  mutate(conc_mod = conc_mod * ((mean(target$SG)-1)/(SG-1))) |>
  select(Project_ID, SID, PID, child_PID, Analyte_Code, conc_mod) |>
  group_by(SID) |>
  mutate(Project_ID = min(Project_ID)) |>
  ungroup() |>
  pivot_wider(names_from = Analyte_Code, values_from = conc_mod) |>
  # howe kept As, Cd, Co, Hg, Ni, Tl, and Pb in main, Mo, Sb, and Sn in supp
  # don't have modified version of As used in their paper
  select(Project_ID, SID, PID, child_PID,
         As, Cd, Co, Hg, Ni, Tl, Pb, Mo, Sb, Sn)

# save
write_csv(target_small, "madres_data/target_small.csv")

# only keep data from first trimester
target_first <- target_small |>
  group_by(child_PID) |>
  filter(Project_ID == min(Project_ID)) |>
  ungroup()

# save
write_csv(target_first, "madres_data/target_first.csv")
```

```
########
# clean epi data
########

# select relevant variables
epi_small <- epi |>
  # make new categorical variables
  mutate(mom_site = as.factor(mom_site),
```

```r
         race = as.factor(case_when(
           t1_demo_hispanic == 0 & t1_demo_race == 2 ~ 1, #non-hisp white
           t1_demo_hispanic == 0 & t1_demo_race == 4 ~ 2, #non-hisp black
           t1_demo_hispanic == 0 ~ 3, #other, non-hispanic
           t1_demo_hispanic == 1 & t1_demo_usa == 1 ~ 4, #hispanic in US
           t1_demo_hispanic == 1 & t1_demo_usa == 0 ~ 5, #hispanic NOT in US
           .default = NA
         )),
         smoke = as.factor(ifelse(
           t1_smoke_preg == 1 | t2_smoke_preg == 1 | t3_smoke_preg == 1 |
             t1_smoke == 1 | t2_smoke == 1 | t3_smoke == 1, 1, 0
         ))) |>
  # replace -99 with NA
  mutate(across(where(is.numeric), ~ifelse(. == -99, NA, .)))  |>
  dplyr::select(child_pid, mom_site,
                age = t1_mat_age, # age, trimester 1
                bmi = t1_pre_BMI, # bmi
                race, # maternal r/e
                smoke, # ever-exposure to smoke
                gender, birthweight, GA # birthweight + gestational age
                # can't find anemia measure or AsB
  )

# handle NA values
epi_imp <- epi_small |>
  # exclude birthweight (observed response)
  # exclude study site because of small categories
  select(-c(gender, birthweight, GA, mom_site)) |>
  # na's for smoke during preg, set to 0
  mutate(smoke = as.factor(ifelse(is.na(smoke), 0, smoke))) |>
  # impute mean for BMI
  mutate(across(where(is.numeric),
                ~ifelse(is.na(.), mean(.,na.rm = TRUE), .)))
```

```r
########
# combine epi and target data
########
comb <- epi_imp |>
  left_join(target_first, by = c("child_pid" = "child_PID")) |>
  relocate(child_pid, Project_ID, SID, PID, mom_site, race, smoke)
```

```r
# remove outliers
comb_small <- comb |>
  filter(Mo >=1, Sb <= 1.4)

# save
write_csv(comb_small, "madres_data/base_data.csv")
```

### B.2.2 Code for Chapter 4.2.2:

Next, we use copulas to simulate predictor data. We use the `copula` and `rslurm` packages in this section. This code was run on the Amherst HPC RStudio server.

```r
# load packages
library(tidyverse)
library(copula)
library(rslurm)
```

```r
# read data back in
comb_small <- read_csv("madres_data/base_data.csv")

# log-transform target data
comb_log <- comb_small |>
  mutate(across(10:19, log)) |>
  # factors back to numeric
  mutate(across(where(is.factor), as.numeric))

# check spearman's rho
cor(comb_log[, 7:19], method = "spearman")
```

```r
########
# fit copulas
########

# create pseudo observations for continuous variables
u <- pobs(comb_log[, 7:19])

# fit checkerboard copula on smoke
prop_smoke0 <- 1 - mean(comb_log$smoke)
```

```
# jitter 0's and 1's uniformly within quantile
set.seed(0)
u_smoke <- comb_log$smoke |>
  map_dbl(\(x) {
    ifelse(x == 0, runif(1, 0, prop_smoke0), runif(1, prop_smoke0, 1))
  })
u[, 1] <- u_smoke

# fit copulas
cfit_gaus <- fitCopula(normalCopula(dim = 13, dispstr = "un"), u)
cfit_t1 <- fitCopula(tCopula(dim = 13, dispstr = "un",
                             df.fixed = FALSE), u)
cfit_t2 <- fitCopula(tCopula(dim = 13, dispstr = "un",
                             df = 4, df.fixed = TRUE), u)
cfit_t3 <- fitCopula(tCopula(dim = 13, dispstr = "un",
                             df = 10, df.fixed = TRUE), u)
cfit_gum1 <- fitCopula(gumbelCopula(4, dim = 13), u)
cfit_gum2 <- fitCopula(gumbelCopula(2, dim = 13), u)
cfit_frank1 <- fitCopula(frankCopula(4, dim = 13), u)
cfit_frank2 <- fitCopula(frankCopula(2, dim = 13), u)
cfit_clay1 <- fitCopula(claytonCopula(4, dim = 13), u)
cfit_clay2 <- fitCopula(claytonCopula(2, dim = 13), u)
cfit_joe1 <- fitCopula(joeCopula(4, dim = 13), u)
cfit_joe2 <- fitCopula(joeCopula(2, dim = 13), u)

# evaluate fit using AIC
aic_values <- sapply(list(cfit_gaus, cfit_t1, cfit_t2, cfit_t3,
                          cfit_gum1, cfit_gum2, cfit_frank1, cfit_frank2,
                          cfit_clay1, cfit_clay2, cfit_joe1, cfit_joe2
                          ), AIC)
names(aic_values) <- c("cfit_gaus", "cfit_t1", "cfit_t2", "cfit_t3",
                       "cfit_gum", "cfit_gum2", "cfit_frank1", "cfit_frank2",
                       "cfit_clay1", "cfit_clay2", "cfit_joe1", "cfit_joe2")
sort(aic_values)

# evaluate fit using likelihood
aic_values <- sapply(list(cfit_gaus, cfit_t1, cfit_t2, cfit_t3,
                          cfit_gum1, cfit_gum2, cfit_frank1, cfit_frank2,
                          cfit_clay1, cfit_clay2, cfit_joe1, cfit_joe2
                          ), logLik)
names(lik_values) <- c("cfit_gaus", "cfit_t1", "cfit_t2", "cfit_t3",
                       "cfit_gum", "cfit_gum2", "cfit_frank1", "cfit_frank2",
                       "cfit_clay1", "cfit_clay2", "cfit_joe1", "cfit_joe2")
sort(lik_values)

# guassian copula performs best, proceed with this
write_rds(cfit_gaus, "sim/gauscop.RDS")
```

```
########
# simulate predictor data
########

# read copula back in
cfit_gaus <- read_rds("sim/gauscop.RDS")

# extract rho
rho <- coef(cfit_gaus)

# create function for simulation
simulate_data <- function(data, n, rho, prop_smoke, prop_race) {
  #' data = original observed data
  #' n = sample size
  #' rho = rho values from normal copula
  #' prop_smoke = proportion smoke from observed dataset
  #' prop_race = table with race/eth values

  # simulate pseudo-observations from copula
  samp <- rCopula(n, normalCopula(rho, dim = ncol(data), dispstr = "un"))

  # transform pseudo-observations to observed marginal distributions
  sampt <- 1:ncol(data) |>
    purrr::map_dfc(
      \(x) {
        if(names(data)[x] == "smoke") {
          # use observed probability threshold for smoke
          df <- data.frame(ifelse(samp[,x] < prop_smoke, 0, 1),
                           row.names = NULL)
        } else {
          # use empirical marginal CDF's for continuous
          df <- data.frame(quantile(data[[x]], probs = samp[,x]),
                           row.names = NULL)
        }
        names(df) <- names(data)[x]
        return(df)
      }
    ) |>
    # randomly sample race
    mutate(race = sample(x = names(prop_race), prob = prop_race,
                         size = n, replace = T)) |>
    relocate(race)
  return(sampt)
}

# create function to run size 252 samples on hpc
run_sim1 <- function() {
  set.seed(0)
  out <- 1:2100 |>
    purrr::map(\(x) {
      mutate(simulate_data(comb_log_clip, n = nrow(comb_log_clip), rho = rho,
                           prop_smoke = 1-mean(comb_log_clip$smoke),
                           prop_race = table(comb_log$race)),
             race = as.numeric(race),
             sim = x)
    })
  return(out)
}

# send job to hpc for size 252 samples
sjob1 <- slurm_call(run_sim1,
                    global_objects = c('comb_log', 'comb_log_clip',
                                       'rho', 'simulate_data'),
                    jobname = 'sim_data1')

# get output
out1 <- get_slurm_out(sjob1)
write_rds(out1, "sim/sim_preds_sm.RDS")

# create function to run size 1000 samples on hpc
run_sim2 <- function() {
  set.seed(1)
  out <- 1:2100 |>
    purrr::map(\(x) {
      mutate(simulate_data(comb_log_clip, n = 1000, rho = rho,
                           prop_smoke = 1-mean(comb_log_clip$smoke),
                           prop_race = table(comb_log$race)),
               race = as.numeric(race),
               sim = x)
    })
  return(out)
}

# send job to hpc for size 1000 samples
sjob2 <- slurm_call(run_sim2,
                    global_objects = c('comb_log', 'comb_log_clip',
                                       'rho', 'simulate_data'),
                    jobname = 'sim_data2')

# get output
out2 <- get_slurm_out(sjob2)
write_rds(out2, "sim/sim_preds_lg.RDS")
```

### B.2.3 Code for Chapter 4.2.3:

Next, we simulate the response data. We use the `rslurm` package in this section. This code was run on the Amherst HPC RStudio server.

```
# load packages
library(tidyverse)
library(rslurm)
```

```
##############
# create functions for various response variables
##############

# base case, no interactions
base_case <- function(df) {
  mutate(df, y =
           Hg + 3/(1+exp(-4*Ni)) - (Sb^2) + 0.5*Sb + 1.5/(1+exp(-4*Sn)) +
           age + 0.5*bmi +
           case_when(race == 1 ~ 1,
                     race == 2 ~ 1.5,
                     race == 3 ~ 1,
                     race == 4 ~ 1,
                     race == 5 ~ 1.5) +
           ifelse(smoke == 1, -1, 0.5) +
           rnorm(nrow(df), 0, 5))
}

am1 <- function(df) {
  mutate(df, y =
           Hg + 3/(1+exp(-4*Ni)) - (Sb^2) + 0.5*Sb + 1.5/(1+exp(-4*Sn)) +
           0.35*Hg*Ni +
           age + 0.5*bmi +
           case_when(race == 1 ~ 1,
                     race == 2 ~ 1.5,
                     race == 3 ~ 1,
                     race == 4 ~ 1,
                     race == 5 ~ 1.5) +
           ifelse(smoke == 1, -1, 0.5) +
           rnorm(nrow(df), 0, 5))
}

am2 <- function(df) {
  mutate(df, y =
           Hg + 3/(1+exp(-4*Ni)) - (Sb^2) + 0.5*Sb + 1.5/(1+exp(-4*Sn)) +
           0.7*Hg*Ni +
           age + 0.5*bmi +
           case_when(race == 1 ~ 1,
                     race == 2 ~ 1.5,
                     race == 3 ~ 1,
                     race == 4 ~ 1,
                     race == 5 ~ 1.5) +
           ifelse(smoke == 1, -1, 0.5) +
           rnorm(nrow(df), 0, 5))
}

ap1 <- function(df) {
  mutate(df, y =
           Hg + 3/(1+exp(-4*Ni)) - (Sb^2) + 0.5*Sb + 1.5/(1+exp(-4*Sn)) +
           0.13*Hg*((Ni-1)^2) +
           age + 0.5*bmi +
           case_when(race == 1 ~ 1,
                     race == 2 ~ 1.5,
                     race == 3 ~ 1,
                     race == 4 ~ 1,
                     race == 5 ~ 1.5) +
            ifelse(smoke == 1, -1, 0.5) +
            rnorm(nrow(df), 0, 5))
}

ap2 <- function(df) {
  mutate(df, y =
            Hg + 3/(1+exp(-4*Ni)) - (Sb^2) + 0.5*Sb + 1.5/(1+exp(-4*Sn)) +
            0.26*Hg*((Ni-1)^2) +
            age + 0.5*bmi +
            case_when(race == 1 ~ 1,
                      race == 2 ~ 1.5,
                      race == 3 ~ 1,
                      race == 4 ~ 1,
                      race == 5 ~ 1.5) +
            ifelse(smoke == 1, -1, 0.5) +
            rnorm(nrow(df), 0, 5))
}

bm1 <- function(df) {
  mutate(df, y =
            Hg + 3/(1+exp(-4*Ni)) - (Sb^2) + 0.5*Sb + 1.5/(1+exp(-4*Sn)) +
            0.35*Cd*As +
            age + 0.5*bmi +
            case_when(race == 1 ~ 1,
                      race == 2 ~ 1.5,
                      race == 3 ~ 1,
                      race == 4 ~ 1,
                      race == 5 ~ 1.5) +
            ifelse(smoke == 1, -1, 0.5) +
            rnorm(nrow(df), 0, 5))
}

bm2 <- function(df) {
  mutate(df, y =
            Hg + 3/(1+exp(-4*Ni)) - (Sb^2) + 0.5*Sb + 1.5/(1+exp(-4*Sn)) +
            0.7*Cd*As +
            age + 0.5*bmi +
            case_when(race == 1 ~ 1,
                      race == 2 ~ 1.5,
                      race == 3 ~ 1,
                      race == 4 ~ 1,
                      race == 5 ~ 1.5) +
           ifelse(smoke == 1, -1, 0.5) +
           rnorm(nrow(df), 0, 5))
}

bp1 <- function(df) {
  mutate(df, y =
           Hg + 3/(1+exp(-4*Ni)) - (Sb^2) + 0.5*Sb + 1.5/(1+exp(-4*Sn)) +
           0.125*Cd*((As-1)^2) +
           age + 0.5*bmi +
           case_when(race == 1 ~ 1,
                     race == 2 ~ 1.5,
                     race == 3 ~ 1,
                     race == 4 ~ 1,
                     race == 5 ~ 1.5) +
           ifelse(smoke == 1, -1, 0.5) +
           rnorm(nrow(df), 0, 5))
}

bp2 <- function(df) {
  mutate(df, y =
           Hg + 3/(1+exp(-4*Ni)) - (Sb^2) + 0.5*Sb + 1.5/(1+exp(-4*Sn)) +
           0.25*Cd*((As-1)^2) +
           age + 0.5*bmi +
           case_when(race == 1 ~ 1,
                     race == 2 ~ 1.5,
                     race == 3 ~ 1,
                     race == 4 ~ 1,
                     race == 5 ~ 1.5) +
           ifelse(smoke == 1, -1, 0.5) +
           rnorm(nrow(df), 0, 5))
}

cm1 <- function(df) {
  mutate(df, y =
           Hg + 3/(1+exp(-4*Ni)) - (Sb^2) + 0.5*Sb + 1.5/(1+exp(-4*Sn)) +
           0.3*Hg*Co +
           age + 0.5*bmi +
           case_when(race == 1 ~ 1,
                     race == 2 ~ 1.5,
                     race == 3 ~ 1,
                     race == 4 ~ 1,
                     race == 5 ~ 1.5) +
           ifelse(smoke == 1, -1, 0.5) +
           rnorm(nrow(df), 0, 5))
}

cm2 <- function(df) {
  mutate(df, y =
           Hg + 3/(1+exp(-4*Ni)) - (Sb^2) + 0.5*Sb + 1.5/(1+exp(-4*Sn)) +
           0.6*Hg*Co +
           age + 0.5*bmi +
           case_when(race == 1 ~ 1,
                     race == 2 ~ 1.5,
                     race == 3 ~ 1,
                     race == 4 ~ 1,
                     race == 5 ~ 1.5) +
           ifelse(smoke == 1, -1, 0.5) +
           rnorm(nrow(df), 0, 5))
}

cp1 <- function(df) {
  mutate(df, y =
           Hg + 3/(1+exp(-4*Ni)) - (Sb^2) + 0.5*Sb + 1.5/(1+exp(-4*Sn)) +
           0.15*Hg*((Co-1)^2) +
           age + 0.5*bmi +
           case_when(race == 1 ~ 1,
                     race == 2 ~ 1.5,
                     race == 3 ~ 1,
                     race == 4 ~ 1,
                     race == 5 ~ 1.5) +
           ifelse(smoke == 1, -1, 0.5) +
           rnorm(nrow(df), 0, 5))
}

cp2 <- function(df) {
  mutate(df, y =
           Hg + 3/(1+exp(-4*Ni)) - (Sb^2) + 0.5*Sb + 1.5/(1+exp(-4*Sn)) +
           0.3*Hg*((Co-1)^2) +
           age + 0.5*bmi +
           case_when(race == 1 ~ 1,
                     race == 2 ~ 1.5,
                     race == 3 ~ 1,
                     race == 4 ~ 1,
                     race == 5 ~ 1.5) +
           ifelse(smoke == 1, -1, 0.5) +
           rnorm(nrow(df), 0, 5))
}

dm1 <- function(df) {
  mutate(df, y =
           Hg + 3/(1+exp(-4*Ni)) - (Sb^2) + 0.5*Sb + 1.5/(1+exp(-4*Sn)) +
           0.3*Hg*Ni*Tl +
           age + 0.5*bmi +
           case_when(race == 1 ~ 1,
                     race == 2 ~ 1.5,
                     race == 3 ~ 1,
                     race == 4 ~ 1,
                     race == 5 ~ 1.5) +
           ifelse(smoke == 1, -1, 0.5) +
           rnorm(nrow(df), 0, 5))
}

dm2 <- function(df) {
  mutate(df, y =
           Hg + 3/(1+exp(-4*Ni)) - (Sb^2) + 0.5*Sb + 1.5/(1+exp(-4*Sn)) +
           0.6*Hg*Ni*Tl +
           age + 0.5*bmi +
           case_when(race == 1 ~ 1,
                     race == 2 ~ 1.5,
                     race == 3 ~ 1,
                     race == 4 ~ 1,
                     race == 5 ~ 1.5) +
           ifelse(smoke == 1, -1, 0.5) +
           rnorm(nrow(df), 0, 5))
}

dp1 <- function(df) {
  mutate(df, y =
           Hg + 3/(1+exp(-4*Ni)) - (Sb^2) + 0.5*Sb + 1.5/(1+exp(-4*Sn)) +
           0.09*Hg*((Ni-1)^2)*Tl +
           age + 0.5*bmi +
           case_when(race == 1 ~ 1,
                     race == 2 ~ 1.5,
                     race == 3 ~ 1,
                     race == 4 ~ 1,
                     race == 5 ~ 1.5) +
           ifelse(smoke == 1, -1, 0.5) +
           rnorm(nrow(df), 0, 5))
}

dp2 <- function(df) {
  mutate(df, y =
           Hg + 3/(1+exp(-4*Ni)) - (Sb^2) + 0.5*Sb + 1.5/(1+exp(-4*Sn)) +
           0.18*Hg*((Ni-1)^2)*Tl +
           age + 0.5*bmi +
           case_when(race == 1 ~ 1,
                     race == 2 ~ 1.5,
                     race == 3 ~ 1,
                     race == 4 ~ 1,
                     race == 5 ~ 1.5) +
           ifelse(smoke == 1, -1, 0.5) +
           rnorm(nrow(df), 0, 5))
}
```

```
#############
# create response variables for exposure-exposure interxn
#############

# read output back in, size 252
out1 <- read_rds("sim/sim_preds_sm.RDS")

# create function for responses at size 252
run_resp1 <- function() {
  set.seed(0)
  out1_resp1 <- out1 |>
    purrr::map(\(x) {
      # get dataset number
      no <- x$sim[1]
      x <- x |>
        # scale log-transformed exposures
        mutate(across(As:Sn, ~c(scale(.))))
      df <- case_when(
        no <= 100 ~ base_case(x),
        no <= 200 ~ am1(x),
        no <= 300 ~ am2(x),
        no <= 400 ~ ap1(x),
        no <= 500 ~ ap2(x),
        no <= 600 ~ bm1(x),
        no <= 700 ~ bm2(x),
        no <= 800 ~ bp1(x),
        no <= 900 ~ bp2(x),
        no <= 1000 ~ cm1(x),
        no <= 1100 ~ cm2(x),
        no <= 1200 ~ cp1(x),
        no <= 1300 ~ cp2(x),
        no <= 1400 ~ dm1(x),
        no <= 1500 ~ dm2(x),
        no <= 1600 ~ dp1(x),
        no <= 1700 ~ dp2(x),
        .default = x #note 1701 - 2100 is for cov-exp interxn
      )
    }) |>
    purrr::set_names(nm = c(
      rep("_base", 100),
      rep("am1", 100),
      rep("am2", 100),
      rep("ap1", 100),
      rep("ap2", 100),
      rep("bm1", 100),
      rep("bm2", 100),
      rep("bp1", 100),
      rep("bp2", 100),
      rep("cm1", 100),
      rep("cm2", 100),
      rep("cp1", 100),
      rep("cp2", 100),
      rep("dm1", 100),
      rep("dm2", 100),
      rep("dp1", 100),
      rep("dp2", 100),
      rep("unset", 400)
    ))
  return(out1_resp1)
}

# run to hpc
runrespsm <- slurm_call(
  run_resp1,
  global_objects = c('out1', 'base_case',
                     'am1', 'am2', 'ap1', 'ap2',
                     'bm1', 'bm2', 'bp1', 'bp2',
                     'cm1', 'cm2', 'cp1', 'cp2',
                     'dm1', 'dm2', 'dp1', 'dp2'),
  jobname = 'sim_resp1')

# get output
out1_resp1 <- get_slurm_out(runrespsm)
# only save for exp-exp interxns
out1_resp1 <- out1_resp1[1:1700]
write_rds(out1_resp1, "sim/sim_resp_sm_a.RDS")

# read output back in, size 1000
out2 <- read_rds("sim/sim_preds_lg.RDS")

# create function for response at size 1000
run_resp2 <- function() {
  set.seed(0)
  out2_resp1 <- out2 |>
    purrr::map(\(x) {
      # get dataset number
      no <- x$sim[1]
      x <- x |>
        mutate(across(As:Sn, ~c(scale(.))))
      df <- case_when(
        no <= 100 ~ base_case(x),
        no <= 200 ~ am1(x),
        no <= 300 ~ am2(x),
        no <= 400 ~ ap1(x),
        no <= 500 ~ ap2(x),
        no <= 600 ~ bm1(x),
        no <= 700 ~ bm2(x),
        no <= 800 ~ bp1(x),
        no <= 900 ~ bp2(x),
        no <= 1000 ~ cm1(x),
        no <= 1100 ~ cm2(x),
        no <= 1200 ~ cp1(x),
        no <= 1300 ~ cp2(x),
        no <= 1400 ~ dm1(x),
        no <= 1500 ~ dm2(x),
        no <= 1600 ~ dp1(x),
        no <= 1700 ~ dp2(x),
```

```r
        .default = x #note 1701 - 2100 is for cov-exp interxn
      )
    }) |>
    purrr::set_names(nm = c(
      rep("_base", 100),
      rep("am1", 100),
      rep("am2", 100),
      rep("ap1", 100),
      rep("ap2", 100),
      rep("bm1", 100),
      rep("bm2", 100),
      rep("bp1", 100),
      rep("bp2", 100),
      rep("cm1", 100),
      rep("cm2", 100),
      rep("cp1", 100),
      rep("cp2", 100),
      rep("dm1", 100),
      rep("dm2", 100),
      rep("dp1", 100),
      rep("dp2", 100),
      rep("unset", 400)
    ))
  return(out2_resp1)
}

# send to HPC
runresplg <- slurm_call(
  run_resp2,
  global_objects = c('out2', 'base_case',
                     'am1', 'am2', 'ap1', 'ap2',
                     'bm1', 'bm2', 'bp1', 'bp2',
                     'cm1', 'cm2', 'cp1', 'cp2',
                     'dm1', 'dm2', 'dp1', 'dp2'),
  jobname = 'sim_resp2')

# get output
out2_resp1 <- get_slurm_out(runresplg)
# only save output for exp-exp interxns for now
out2_resp1 <- out2_resp1[1:1700]
write_rds(out2_resp1, "sim/sim_resp_lg_a.RDS")
```

```
#############
# create response variables for exposure-covariate interxn
#############
```

### B.2.4 Code for Chapter 4.2.4:

Here, we fit the models to our simulated data. We use the `rslurm`, `bkmr`, and `NLinteraction` packages in this section. This code was run on the Amherst HPC RStudio server.

### B.2.5 Code for Chapter 4.3:

Here, we extract results from our simulation. We use the `rslurm`, `bkmr`, and `NLinteraction` packages in this section. This code was run on the Amherst HPC RStudio server.

# Corrections

A list of corrections after submission to department.

Corrections may be made to the body of the thesis, but every such correction will be acknowledged in a list under the heading “Corrections,” along with the statement “When originally submitted, this honors thesis contained some errors which have been corrected in the current version. Here is a list of the errors that were corrected.” This list will be given on a sheet or sheets to be appended to the thesis. Corrections to spelling, grammar, or typography may be acknowledged by a general statement such as “30 spellings were corrected in various places in the thesis, and the notation for definite integral was changed in approximately 10 places.” However, any correction that affects the meaning of a sentence or paragraph should be described in careful detail. The files samplethesis.tex and samplethesis.pdf show what the “Corrections” section should look like. Questions about what should appear in the “Corrections” should be directed to the Chair.

# References

Antonelli, J., Mazumdar, M., Bellinger, D., Christiani, D., Wright, R., & Coull, B. (2020). Estimating the health effects of environmental mixtures using Bayesian semiparametric regression and sparsity inducing priors. *The Annals of Applied Statistics*, *14*(1), 257–275. http://doi.org/10.1214/19-AOAS1307

Barrera-Gómez, J., Agier, L., Portengen, L., Chadeau-Hyam, M., Giorgis-Allemand, L., Siroux, V., … Basagaña, X. (2017). A systematic comparison of statistical methods to detect interactions in exposome-health associations. *Environmental Health*, *16*(1), 74. http://doi.org/10.1186/s12940-017-0277-6

Bastain, T. M., Chavez, T., Habre, R., Girguis, M. S., Grubbs, B., Toledo-Corral, C., … Breton, C. (2019). Study Design, Protocol and Profile of the Maternal And Developmental Risks from Environmental and Social Stressors (MADRES) Pregnancy Cohort: A Prospective Cohort Study in Predominantly Low-Income Hispanic Women in Urban Los Angeles. *BMC Pregnancy and Childbirth*, *19*(1), 189. http://doi.org/10.1186/s12884-019-2330-7

Bellavia, A. (2021). *Statistical Methods for Environmental Mixtures*. Retrieved from `https://bookdown.org/andreabellavia/mixtures/preface.html`

Bobb, J. F. (2017a, March). Introduction to Bayesian kernel machine regression and the bkmr R package. Retrieved from `https://jenfb.github.io/bkmr/`

`overview.html`

Bobb, J. F. (2017b, December). Example using the bkmr R package with simulated data from the NIEHS mixtures workshop. Retrieved from `https://jenfb.github.io/bkmr/SimData1.html#1_load_packages_and_download_data`

Bobb, J. F., Claus Henn, B., Valeri, L., & Coull, B. A. (2018). Statistical software for analyzing the health effects of multiple concurrent exposures via Bayesian kernel machine regression. *Environmental Health*, *17*(1), 67. http://doi.org/10.1186/s12940-018-0413-y

Bobb, J. F., Valeri, L., Claus Henn, B., Christiani, D. C., Wright, R. O., Mazumdar, M., … Coull, B. A. (2015). Bayesian kernel machine regression for estimating the health effects of multi-pollutant mixtures. *Biostatistics*, *16*(3), 493–508. http://doi.org/10.1093/biostatistics/kxu058

Cribb, J. (2016). *Surviving the 21st century: Humanity's ten great challenges and how we can overcome them*. Springer.

Genest, C., & Nešlehovà, J. (2007). A Primer on Copulas for Count Data. *ASTIN Bulletin*, *37*(2), 475–515. http://doi.org/`https://doi.org/10.2143/AST.37.2.2024077`

Gibson, E. A., Nunez, Y., Abuawad, A., Zota, A. R., Renzetti, S., Devick, K. L., … Kioumourtzoglou, M.-A. (2019). An overview of methods to address distinct research questions on environmental mixtures: An application to persistent organic pollutants and leukocyte telomere length. *Environmental Health*, *18*(1), 76. http://doi.org/10.1186/s12940-019-0515-1

Halford, G. S., Baker, R., McCredden, J. E., & Bain, J. D. (2005). How Many Variables Can Humans Process? *Psychological Science*, *16*(1), 70–76.

http://doi.org/10.1111/j.0956-7976.2005.00782.x

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. New York, NY: Springer New York. http://doi.org/10.1007/978-0-387-84858-7

Hernández, A. F., Parrón, T., Tsatsakis, A. M., Requena, M., Alarcón, R., & López-Guarnido, O. (2013). Toxic effects of pesticide mixtures at a molecular level: Their relevance to human health. *Toxicology*, *307*, 136–145. http://doi.org/10.1016/j.tox.2012.06.009

Heys, K., Shore, R., Pereira, M., Jones, K., & Martin, F. (2016). Risk assessment of environmental mixture effects. *RSC Advances*, *6*(53), 47844–47857. http://doi.org/10.1039/C6RA05406D

Hofert, M., Kojadinovic, I., Maechler, M., & Yan, J. (2023). Copula: Multivariate Dependence with Copulas. Retrieved from `https://CRAN.R-project.org/package=copula`

Hoskovec, L., Benka-Coker, W., Severson, R., Magzamen, S., & Wilson, A. (2021). Model choice for estimating the association between exposure to chemical mixtures and health outcomes: A simulation study. *PLOS ONE*, *16*(3), e0249236. http://doi.org/10.1371/journal.pone.0249236

Howe, C. G., Claus, H. B., Eckel, S. P., Farzan, S. F., Grubbs, B. H., Chavez, T. A., … Breton, C. V. (2020). Prenatal Metal Mixtures and Birth Weight for Gestational Age in a Predominately Lower-Income Hispanic Pregnancy Cohort in Los Angeles. *Environmental Health Perspectives*, *128*(11), 117001. http://doi.org/10.1289/EHP7201

Kannan, S., Misra, D. P., Dvonch, J. T., & Krishnakumar, A. (2006). Exposures to Airborne Particulate Matter and Adverse Perinatal Outcomes: A Biologically Plausible Mechanistic Framework for Exploring Potential Effect Modification by Nutrition. *Environmental Health Perspectives*, *114*(11), 1636–1642. http://doi.org/10.1289/ehp.9081

Kordas, K., Lönnerdal, B., & Stoltzfus, R. J. (2007). Interactions between nutrition and environmental exposures: Effects on health outcomes in women and children. *The Journal of Nutrition*, *137*(12), 2794–2797. http://doi.org/10.1093/jn/137.12.2794

Lazarevic, N., Knibbs, L. D., Sly, P. D., & Barnett, A. G. (2020). Performance of variable and function selection methods for estimating the nonlinear health effects of correlated chemical mixtures: A simulation study. *Statistics in Medicine*, *39*(27), 3947–3967. http://doi.org/10.1002/sim.8701

Liu, D., Lin, X., & Ghosh, D. (2007). Semiparametric Regression of Multidimensional Genetic Pathway Data: Least-Squares Kernel Machines and Linear Mixed Models. *Biometrics*, *63*(4), 1079–1088. http://doi.org/10.1111/j.1541-0420.2007.00799.x

Müller, H.-G. (1987). Weighted Local Regression and Kernel Methods for Nonparametric Curve Fitting. *Journal of the American Statistical Association*, *82*(397), 231–238. http://doi.org/10.1080/01621459.1987.10478425

Nadaraya, E. A. (1964). On Estimating Regression. *Theory of Probability & Its Applications*, *9*(1), 141–142. http://doi.org/10.1137/1109020

Naidu, R., Biswas, B., Willett, I. R., Cribb, J., Kumar Singh, B., Paul Nathanail, C., … Aitken, R. J. (2021). Chemical pollution: A growing peril and potential catastrophic risk to humanity. *Environment International*, *156*, 106616. http://doi.org/10.1016/j.envint.2021.106616

National Academies of Sciences, Engineering, and Medicine, Division on Earth and Life Studies, Board on Environmental Studies and Toxicology, & Committee on Incorporating 21st Century Science into Risk-Based Evaluations. (2017). *Using 21st Century Science to Improve Risk-Related Evaluations.* Washington, D.C.: National Academies Press. http://doi.org/10.17226/24635

Nelsen, R. B. (2006). *An introduction to copulas* (2nd ed). New York: Springer.

Persson, L., Carney Almroth, B. M., Collins, C. D., Cornell, S., De Wit, C. A., Diamond, M. L., . . . Hauschild, M. Z. (2022). Outside the Safe Operating Space of the Planetary Boundary for Novel Entities. *Environmental Science & Technology*, *56*(3), 1510–1521. http://doi.org/10.1021/acs.est.1c04158

Pesenti, N., Quatto, P., Colicino, E., Cancello, R., Scacchi, M., & Zambon, A. (2023). Comparative efficacy of three Bayesian variable selection methods in the context of weight loss in obese women. *Frontiers in Nutrition*, *10*, 1203925. http://doi.org/10.3389/fnut.2023.1203925

Plackett, R. L., & Hewlett, P. S. (1952). Quantal Responses to Mixtures of Poisons. *Journal of the Royal Statistical Society: Series B (Methodological)*, *14*(2), 141–154. http://doi.org/10.1111/j.2517-6161.1952.tb00108.x

R Core Team. (2013). *R: A language and environment for statistical computing: Reference index.* Vienna: R Foundation for Statistical Computing.

Schulz, E., Speekenbrink, M., & Krause, A. (2018). A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *Journal of Mathematical Psychology*, *85*, 1–16. http://doi.org/10.1016/j.jmp.2018.03.001

Siemiatycki, J., & Thomas, D. C. (1981). Biological Models and Statistical Interactions: An Example from Multistage Carcinogenesis. *International Journal of*

*Epidemiology*, *10*(4), 383–387. http://doi.org/10.1093/ije/10.4.383

Sun, Z., Tao, Y., Li, S., Ferguson, K. K., Meeker, J. D., Park, S. K., … Mukherjee, B. (2013). Statistical strategies for constructing health risk models with multiple pollutants and their interactions: Possible choices and comparisons. *Environmental Health*, *12*(1), 85. http://doi.org/10.1186/1476-069X-12-85

Taylor, K. W., Joubert, B. R., Braun, J. M., Dilworth, C., Gennings, C., Hauser, R., … Carlin, D. J. (2016). Statistical Approaches for Assessing Health Effects of Environmental Chemical Mixtures in Epidemiology: Lessons from an Innovative Workshop. *Environmental Health Perspectives*, *124*(12), A227–A229. http://doi.org/10.1289/EHP547

VanderWeele, T. J., & Knol, M. J. (2014). A Tutorial on Interaction. *Epidemiologic Methods*, *3*(1), 33–72. http://doi.org/10.1515/em-2013-0005

Vineis, P. (2018). From John Snow to omics: The long journey of environmental epidemiology. *European Journal of Epidemiology*, *33*(4), 355–363. http://doi.org/10.1007/s10654-018-0398-4

Wagaman, A. S., & Dobrow, R. P. (2021). *Probability: With Applications and R* (1st ed.). Wiley. http://doi.org/10.1002/9781119692430

Ward, J. B., Gartner, D. R., Keyes, K. M., Fliss, M. D., McClure, E. S., & Robinson, W. R. (2019). How do we assess a racial disparity in health? Distribution, interaction, and interpretation in epidemiological studies. *Annals of Epidemiology*, *29*, 1–7. http://doi.org/10.1016/j.annepidem.2018.09.007

Watson, G. S. (1964). Smooth Regression Analysis. *Sankhyā: The Indian Journal of Statistics*, *26*(4), 359–372.

Yu, L., Liu, W., Wang, X., Ye, Z., Tan, Q., Qiu, W., . . . Chen, W. (2022). A review of practical statistical methods used in epidemiological studies to estimate the health effects of multi-pollutant mixture. *Environmental Pollution*, *306*, 119356. http://doi.org/10.1016/j.envpol.2022.119356