Global Epidemiology 9 (2025) 100179

Contents lists available at ScienceDirect

Global Epidemiology

journal homepage: www.sciencedirect.com/journal/global-epidemiology

Commentary

AI-assisted exposure-response data analysis: Quantifying heterogeneous causal effects of exposures on survival times

Louis Anthony Cox Jr. [a,*], R. Jeffrey Lewis [b], Saumitra V. Rege [c], Shubham Singh [d]

[a] Cox Associates, Entanglement, and University of Colorado. 503 N. Franklin Street, Denver, Colorado, 80218, USA
[b] Kelly Services, Epidemiology Contractor (retired ExxonMobil Biomedical Sciences, Inc.), Lavallette, New Jersey, USA
[c] Epidemiology, ExxonMobil Biomedical Sciences, Inc.1545 U.S. Highway 22 East Annandale, NJ 08801-3059, USA
[d] Business Analytics (BANA) Program, Business School, University of Colorado, 1475 Lawrence St. Denver, CO 80217-3364, USA

ARTICLE INFO

Keywords
AI-assisted data analysis
ICE plots
Exposure-response modeling
Survival trees
Random survival Forest
Causal Survival Forest
Heterogeneous treatment effects

ABSTRACT

AI-assisted data analysis can help risk analysts better understand exposure-response relationships by making it relatively easy to apply advanced statistical and machine learning methods, check their assumptions, and interpret their results. This paper demonstrates the potential of large language models (LLMs), such as ChatGPT, to facilitate statistical analyses, including survival data analyses, for health risk assessments. Through AI-guided analyses using relatively recent and advanced methods such as Individual Conditional Expectation (ICE) plots using Random Survival Forests and Heterogeneous Treatment Effects (HTEs) estimated using Causal Survival Forests, population-level exposure-response functions can be disaggregated into individual-level exposure-response functions. These reveal the extent of heterogeneity in risks across individuals for different levels of exposure, holding other variables fixed. By applying these methods to an illustrative dataset on blood lead levels (BLL) and mortality risk among never-smoker men from the NHANES III survey, we show how AI can clarify inter-individual variations in exposure-associated risks. The results add insights not easily obtained from traditional parametric or semi-parametric models such as logistic regression and Cox proportional hazards models, illustrating the advantages of non-parametric approaches for quantifying heterogeneous causal effects on survival times. This paper also suggests some practical implications of using AI in regulatory health risk assessments and public policy decisions.

Introduction

AI-assisted data analysis has great potential to improve the practice of exposure-response modeling and data analysis. AI assistance makes it easier than ever before for epidemiologists and risk analysts to apply advanced statistical methods, check whether they are appropriate for a given dataset, and gain more detailed and accurate insights into health outcomes associated with exposures by performing risk assessment modeling disaggregated down to the individual level. AI can help risk assessors select, apply, and optimize or "tune" appropriate machine learning (ML) and non-parametric statistical modeling methods to obtain better insights from data. "Better" here means insights that are less assumption-dependent (or, conversely, that are more robust to modeling choices and assumptions) than those from traditional parametric statistical modeling methods such as regression modeling. AI-assisted data analysis can also help to visualize, interpret, and communicate results from these advanced methods in relatively simple, intuitive ways that facilitate comprehension and appropriate use of the results in health risk assessment. This paper illustrates and critically assesses the extent to which current large language models (LLMs), represented by ChatGPT, can already be used to obtain the above advantages both for relatively simple exposure-response regression models and also for more advanced survival curve models.

This paper has two complementary objectives: to illustrate the advantages of current LLMs in facilitating relatively advanced data analyses, thereby reducing the barriers to using these methods in practice; and to show how LLM-assisted data analysis can support practical application of multiple complementary methods for clarifying causal impacts of variables on health outcomes over time. While AI's usefulness in automating statistical tasks and generating insights from statistical modeling results is a key focus, the paper also demonstrates how combining AI tools with traditional and emerging methods (e.g.,

* Corresponding author.
E-mail address: tcoxdenver@aol.com (L.A. Cox).

https://doi.org/10.1016/j.gloepi.2024.100179
Received 13 September 2024; Received in revised form 13 December 2024; Accepted 21 December 2024
Available online 25 December 2024

regression models, survival forests, causal survival analyses) can enhance our understanding of causation. Combining these two perspectives suggests practical ways to use currently available AI to greatly simplify application of appropriate advanced analytics methods to quantitative health risk assessments.

To quickly convey some of the main ideas, we will first show them in the simple context of the mtcars dataset, a well-known dataset often used in statistical and machine learning courses to introduce key ideas and methods to students. The dataset consists of 32 observations of automobile design and performance metrics. Key variables include mpg (miles per gallon, a measure of fuel efficiency) and wt (weight in thousands of pounds, an indicator of vehicle mass). The dataset is frequently used to illustrate regression and other machine learning methods due to its simplicity and interpretability. We use it now to demonstrate AI LLM-assisted regression modeling and diagnostic techniques in a very simple, well-understood setting: simple linear regression (SLR) modeling with one continuous dependent variable and one continuous independent variable. Later sections then consider more sophisticated analyses and more complex health data.

A simple example: AI-assisted simple linear regression

Before turning to exposure-response modeling, it may be useful to illustrate the AI-assisted approach to data analysis for mtcars. Entering the prompt "*Please show me a linear regression model of mpg vs. wt in mtcars, complete with 95% confidence bands*" into ChatGPT 4o (as of August 2024) produced Fig. 1 on the first attempt.

(Not all runs are equally successful. On the third attempt, the same version of ChatGPT produced "There seems to be a persistent issue with processing the data types, which is preventing the successful generation of the plot with confidence intervals.") In this case, the AI had no difficulty retrieving the mtcars dataset, identifying that "*wt*" is a car weight variable and that "*mpg*" is miles per gallon, and correctly labeled the axes and performing the requested analysis. A follow-up prompt to check modeling assumptions, namely,

User: *Please check whether the assumptions of the linear regression model hold and report the corresponding regression diagnostics. (The R package GVLMA automates this assumption-checking. Please use something similar for Python if possible.)*

elicited this response:

ChatGPT: "I have provided the regression diagnostics, including tests for normality, homoscedasticity, multicollinearity, and autocorrelation of residuals. Here are the key results:

1. **Shapiro-Wilk Test for Normality**: The *p*-value indicates that the residuals do not significantly deviate from normality (*p*-value >0.05).
2. **Breusch-Pagan Test for Heteroscedasticity**: The *p*-value suggests no evidence of heteroscedasticity, meaning the variance of residuals appears constant (p-value >0.05).
3. **Variance Inflation Factor (VIF)**: The VIF values suggest that there is no significant multicollinearity (a VIF value of 1 is ideal).
4. **Durbin-Watson Statistic**: This value indicates that there might be some positive autocorrelation in the residuals (values close to 2 suggest no autocorrelation).

These diagnostics help to validate the assumptions of the linear regression model…".

Comment: This response is only partly adequate, as it fails to check whether the key assumption of a linear relationship between the independent and dependent variables, *wt* and *mpg*, is consistent with the data. A human expert noticing this omission can direct the AI to correct it:

User: …*Please apply the Ramsey RESET test yourself in Python to get a p-value for the above linear regression model*

ChatGPT: "The Ramsey RESET test has returned a p-value of approximately 0.0029. This low p-value indicates that there is statistically significant evidence to suggest that the identity link function may not be appropriate, implying potential non-linearities in the relationship between mpg and wt. In other words, the model might be missing some non-linear terms or interactions, which could be better captured with a different link function or by including higher-order terms in the model."

User: *Please show me a LOESS curve instead*

ChatGPT: "Here is the LOESS curve for the relationship between miles per gallon (mpg) and weight (wt) in the mtcars dataset. The red line represents the smoothed curve, which is locally fitted to the data, allowing for a flexible, non-linear relationship between the variables. This visualization can help you see if there are any non-linear patterns in the data that were not captured by the linear regression model." (The full chat is at https://chatgpt.com/share/a10b35ee-6087-4388-bf03-9c00b115bca0).

This short chat illustrates some of the power and some of the

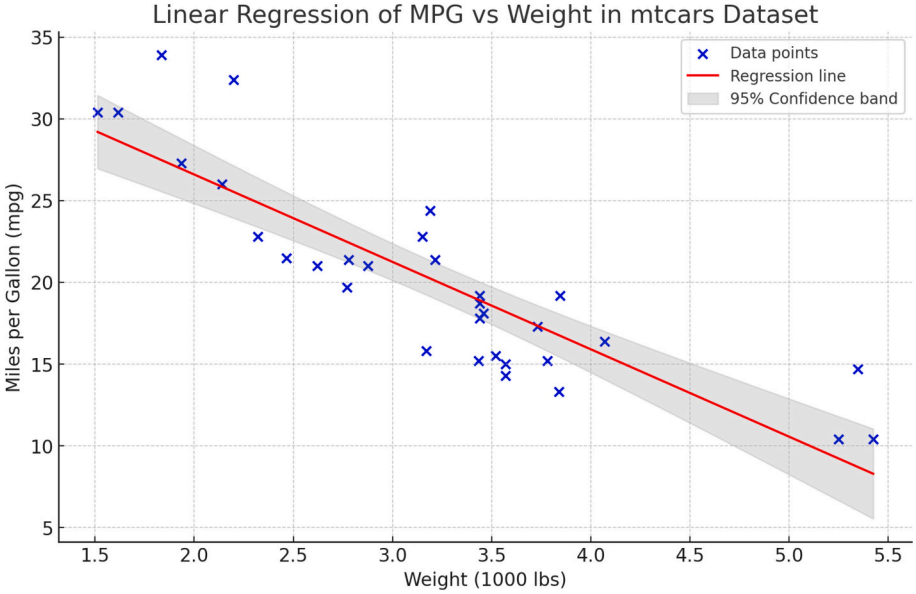

Fig. 1. A simple linear regression (SLR) model.

limitations of current conversational AI. ChatGPT is able to successfully run basic statistical analyses, such as linear regression, model diagnostics, and LOESS (non-parametric smoothing regression) analyses, when directed to do so. It takes the initiative to identify numerous needed details, such as which specific model diagnostic tests to perform, how to label the axes of plots, and what colors to use for plotting, which is very convenient and a very useful time-saver for busy analysts. It also interprets the results, which can help communicate findings to non-specialists. On the other hand, its choices are not always fully adequate. For example, it neglected to check the assumption of a linear link function until prompted to do so. Its unguided interpretations may also be simplistic or wrong. For example, an initial assessment that a model fits the data adequately may have to be revised when a user points out additional tests that should be run. This mix of skills and limitations positions current AI to be a very useful partner and assistant to a human data analyst, but not a reliable substitute for human experts.

For statistical programming, an extremely useful capability of current AI is that it can show the code it uses in its analyses. It can also generate new code upon demand for performing advanced data analyses, even in cases where it cannot run the code itself, e.g., because of lack of access to needed packages. For example, ChatGPT does its analyses in Python, but it can also generate code in R that a user can then paste into R, or inspect and modify as needed. Getting the code right for complex analyses frequently takes many iterations, strong direction, and patience from the user, again emphasizing the importance of human-AI collaboration. But the fact that ChatGPT and other large language models (LLMs) such as Gemini can appropriately draw on thousands of statistics packages in R and Python to help create new code to accomplish user-specified analyses vastly increases productivity in performing advanced analyses. Checking that different implementations using different statistics packages in R and Python produce the same answers help build confidence in the results.

As an example, Fig. 2 shows an Individual Conditional Expectation (ICE) plot [11] for the same data as in Fig. 1. Appendix A gives the code, developed with the help of ChatGPT, to generate this plot in R using existing ICE plot packages. In an ICE plot, each individual case is represented by an entire curve showing the predicted expected values of the dependent variable (here, *mpg*) for different values of the independent variable (*wt*), holding all other variables at their current levels for each individual. Predictions are made using an ML algorithm. The ICE plot in Fig. 2 used the popular Random Forest algorithm for its predictions. The dots show the predicted value of *mpg* for the current value of *wt* for each case (make of car) in the dataset. The curve going through each dot shows how the predicted *mpg* value changes as *wt* changes, but none of the other variables changes. Thus, an ICE plot is closely analogous to the concept of the natural direct effect (of *wt* on *mpg*) used in epidemiology and mediation analysis [5].

The heavy curve in the middle is a Partial Dependence Plot (PDP). This shows the average (marginal) values of the ICE plots averaged over all individual cases. Both the PDP and the individual curves in the ICE plot are less steep than the cloud of dots, meaning that most of the individual data points fall above the PDP on the left (positive residuals) and below it on the right (negative residuals). This reflects the fact that various confounders (including displacement and number of cylinders) that are positively correlated with *wt* and negatively correlated with *mpg* strengthen the negative statistical association between *wt* and *mpg*. Cars with higher values of weight also tend to have more cylinders and greater displacement. The points in the scatter plot of predicted *mpg* vs. *wt* reflect not only how *mpg* covaries with *wt* alone, as in a PDP or ICE plot, but also the effects on *mpg* of these correlates of *wt*. ICE plots remove the effects of such observed confounders by holding their levels fixed as *wt* varies.

Fig. 2 also illustrates an important conceptual limitation of all analytic methods. The meaning and utility of an analysis is limited by the clarity of the concepts and variables it uses [14]. Operational definitions can show exactly what is and is not implied by analytic results, including ICE plots. For example, while it is surely intuitive that lighter cars get higher mpg, other things held fixed, as quantified in Fig. 2, this does not imply that lightening the load by removing its engine or its wheels would cause a car to get better gas mileage. Such common-sense caveats are beyond what data analysis alone can show, but are well within the grasp of current LLMs, suggesting another way in which LLMs can

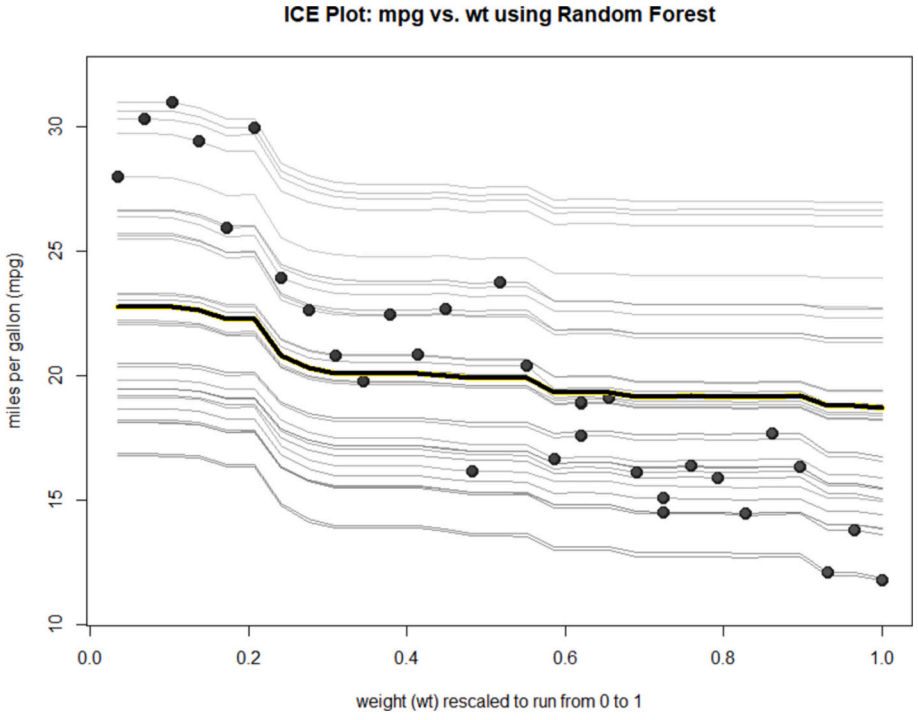

Fig. 2. An Individual Conditional Expectation (ICE) plot.
Source: The code to generate this ICE plot is in Appendix A.

complement purely statistical and ML methods in making sense out of data.

Figs. 1–3 are intended to illustrate for a non-controversial dataset how current AI-assisted analysis can help perform both elementary and more sophisticated analyses with only a small fraction of the effort that was required before the advent of LLMs. The following sections suggest and illustrate how this technology can be applied to help clarify exposure-response relationships and health risks.

An example data set for health risk assessment blood lead and mortality risk in the NHANES III data

To explore how well current AI-assisted data analysis can be applied to realistic exposure-response data, we use a previously analyzed illustrative dataset assembled from the NHANES III survey data (Third National Health and Nutrition Examination Survey (NHANES III), 1988–94, https://wwwn.cdc.gov/nchs/nhanes/nhanes3/default.aspx), focusing on blood lead levels (BLL) and their impact on mortality risk among self-reported never-smoker men. This example dataset is described and analyzed further in [5,7]). Table 1 summarizes its variables. Table 2 shows the layout of the data. We use this dataset purely for purposes of illustration, and do not seek to draw conclusions from it for any larger or different population (e.g., the general US population), so we do not use the weights for various subpopulations developed by NHANES.

Table 1
Variables in the Example NHANES III Blood Lead Data Set.

Variable	Type	Description
Dependent Variables		
Status	Binary	Indicates whether an individual had died by the end of the follow-up period. Value of 1 if the individual had died, 0 if still alive.
Death Age (deathage)	Continuous	Age at death, measured in years, derived as: deathage = age + survival.time/12, where survival time is in months and age is in years.
Independent Variables		
Exposure = Blood Lead Level (BLL)	Continuous	Primary exposure variable, measured in micrograms per deciliter (μg/dL).
Age	Continuous	Age at the start of follow-up, measured in years.
Grade	Continuous	Highest grade of education attained, measured as a continuous variable.
Income Ratio	Continuous	Income expressed as a multiple of the poverty level.
Ethnicity and Race	Categorical	
Not Hispanic (notHispanic)	Binary	Indicates whether the individual is non-Hispanic (1) or Hispanic (0).
Black	Binary	Indicates whether the individual is Black (1) or not (0).
Geographic Location		
Small Metropolitan Area (small.metro)	Binary	Indicates residence in a small metropolitan area (1) or not (0).
South	Binary	Indicates residence in the South region (1) or not (0).
West	Binary	Indicates residence in the West region (1) or not (0).
Marital Status		
Never Married (never.married)	Binary	Indicates whether the individual has never been married (1) or otherwise (0).

AI-assisted data analysis and results

The following sections show how AI-assisted data analysis can be used to obtain insights into BLL-mortality exposure-response relationships in the example dataset. The interplay between human user knowledge (knowing what to ask for) and ChatGPT knowledge (knowing how to produce it) is clearly illustrated throughout the process of producing and refining results.

AI-assisted logistic regression analysis

An epidemiologist might start exploring this data set by fitting a logistic regression model to the exposure (BLL) and response (mortality) data, as in Fig. 3.

Using an LLM, this can be done by attaching the data file and entering a prompt such as "Please show me the logistic regression curve of mortality probability vs. BLL implied by the logistic regression model, with 95% confidence bands." This prompt induced ChatGPT to produce Fig. 3. (The full chat is at the link provided under Fig. 3. The key word

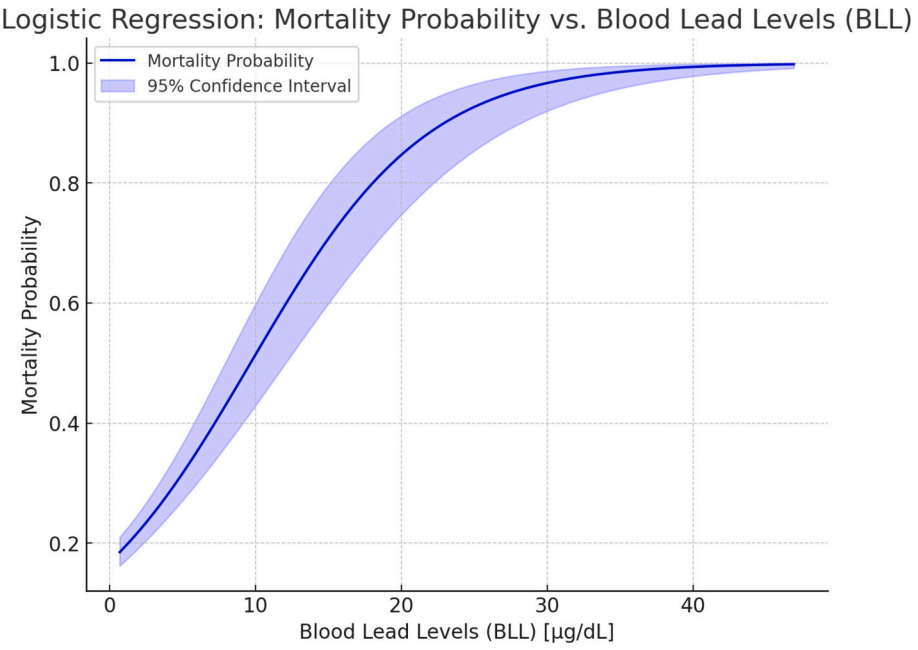

Fig. 3. A simple logistic regression model fit to the BLL-mortality exposure-response data.
Source: Appendix B, https://chatgpt.com/share/ddc0b554-755f-4e73-adae-d076abb0b801

Table 2
Layout of the Data (first 10 of 2631 cases) for Male Non-Smokers.

age	grade	income.ratio	Exposure	survival.time	Status	Not Hispanic	small. metro	Black	South	West	never. married
21	12	0.641	5	293	0	0	0	0	0	1	1
35	17	5.406	2	306	0	1	0	0	0	0	0
50	12	3.416	7.3	292	0	0	0	0	0	1	0
31	14	3.358	5.6	311	0	1	0	0	0	0	0
55	16	2.578	3.2	313	0	1	1	0	1	0	0
27	14	3.918	3.3	314	0	1	0	0	0	1	0
72	12	2.569	4.9	182	1	0	0	0	0	1	0
43	8	1.19	4	292	0	0	0	0	0	1	0
32	17	5.46	2.5	313	0	1	0	0	0	1	0
21	12	2.123	1.3	308	0	0	1	0	0	1	1

"please" is used in this and subsequent chats to indicate specific tasks that the user asks the LLM to complete.) Since creating and validating the code can be a time-consuming, iterative process when the LLM makes mistakes, it is convenient to harvest and save the final versions of the code segments used to produce the results shown. Appendix B does this for the logistic regression modeling results, starting with the simple (univariate) logistic regression model shown in Fig. 3.

Likewise, the LLM can perform multivariate logistic regression modeling in Python, or generate R code to do it. It can then select and run model diagnostics and interpret the results. Doing so in an extension of the chat that produced Fig. 3 produced the forest plot in Fig. 4. In this plot, 95 % confidence intervals (the horizontal "whiskers" around the dots representing point estimates of odds ratios) entirely to the right of 1 indicate a significant (at the 5 % significance level) increase in predicted risk per unit of the corresponding variable. This is the case for *never. married* and *age.* Confidence intervals entirely to the left of 1 indicate a significant decrease in predicted risk per unit of the corresponding variable, as is the case for *income.ratio* and *grade.* Variables with confidence intervals that include 1 do not have any clear statistically significant association with mortality probability.

The same chat that produced the R code for Figs. 3 and 4 also contains the following AI-generated interpretations, conclusions, and caveats for the multivariate logistic regression model:

"1. **Model Fit**: The **Hosmer-Lemeshow test** suggests a poor fit for the logistic regression model. This means the model might not capture the relationship between the predictors and mortality risk well. You may want to explore alternative models (e.g., generalized additive models or interaction terms).

2. **Significant Predictors**:

- **Age**, **education (grade)**, **income ratio**, and **marital status (never married)** are significantly associated with mortality risk.
- **Blood Lead Levels (Exposure)** do not appear to be significantly associated with mortality risk.

3. **Multicollinearity**: There is no serious multicollinearity issue based on the VIF values.

4. **Influential Points**: A large number of influential points may be affecting the model. These should be reviewed and potentially addressed (e.g., through robust regression techniques or sensitivity analysis)."

These conclusions suggest that the positive exposure-response association in Fig. 3 might plausibly be due to confounding, since there is no significant exposure-response association in a multivariate logistic regression model. Further exploration reveals that *age* is positively

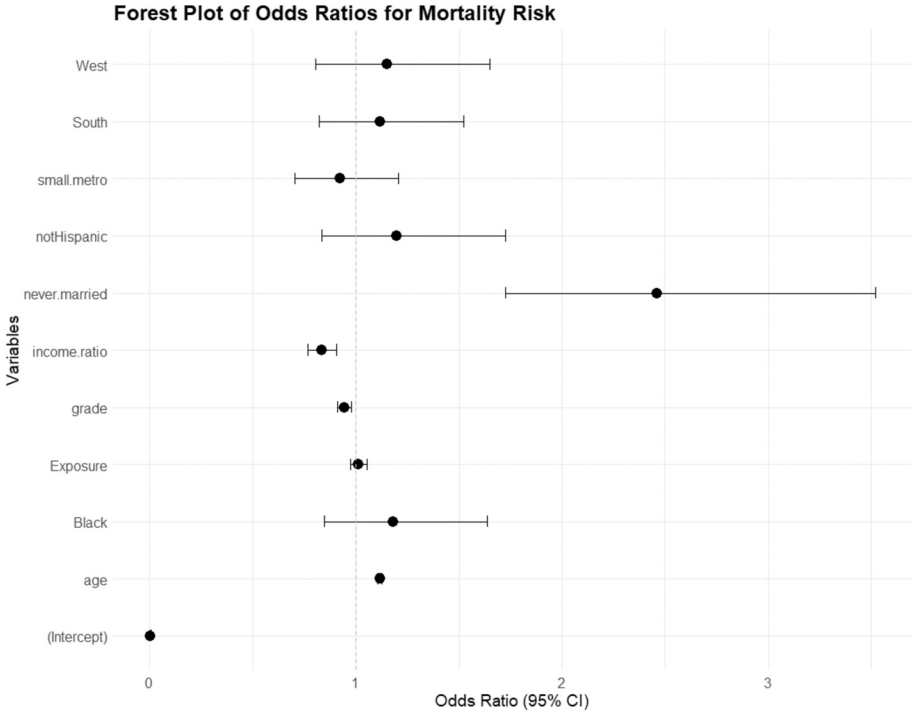

Fig. 4. A forest plot for the multivariate logistic regression model.
Source: Appendix provides the R code used to generate this figure, and the URL for the chat that produced the R code.

correlated with both exposure and mortality risk. This confounder was not controlled (held fixed) in Fig. 3, and much of the positive association between exposure and mortality probability is explained by *age*. However, regression diagnostics suggest that multivariate logistic regression is not the most appropriate model for this dataset (p-value = 1.036e-06 in the Hosmer-Lemeshow test).

AI-assisted ICE plot and PDP analysis

Fig. 5 shows a PDP and ICE plot as a non-parametric alternative to multivariate logistic regression modeling. The PDP suggests an increase in mortality risk from about 0.30 on average at low exposures (BLL levels) to about 0.37 at high exposures, holding all other variables in the dataset fixed for each individual. (The possibility of unmeasured or residual confounding is not excluded by such PDP analysis, however.) The ICE curves show that much of the inter-individual heterogeneity stems from a cluster of high-risk individuals – those who were old at the start of follow-up – against a background of relatively lower-risk individuals. To better show inter-individual variability, Fig. 5b shows a *centered ICE plot*, meaning one in which all individual ICE curves are given a nominal starting value of 0 at the left end (for exposure = 0) and the curves show predicted deviations from this zero-exposure baseline level of risk as exposure increases. This display has a striking implication. Although predicted mortality risk increases with exposure *on average*, as shown by the PDP, for a substantial fraction of the individuals, the ICE curves decrease with exposure, other variables in the dataset being held fixed. If the individual-level exposure-response relationships predicted by these curves are assumed to be causal, meaning that the changes in mortality risk along each curve are *caused* by changes in exposure (rather than by unmeasured or residual confounding), then reductions in exposure reduce average risk in the population but nonetheless increase it for a substantial fraction of the individuals in the population. Such inter-individual heterogeneity is obscured by statistical analyses that only consider average effects in populations. The vast majority of exposure-response curves and analyses published to date do not quantify inter-individual heterogeneity at the individual level. AI-assisted data analysis now makes such individual-level ICE plot calculations relatively easy to perform, but it does not tell risk managers and policy-makers how to use the results, which make visible potential trade-offs between risks to different individuals that are left comfortably invisible in more traditional, aggregate, statistical modeling.

Background on survival data analysis

We now pivot from considering mortality probability during follow up as the main outcome of interest to considering survival times of individuals as the main outcome of interest. Quantifying the probabilistic relationship between exposure and survival times requires survival data analysis. This section introduces key ideas from this branch of statistics.

Survival data analysis comprises a set of methods that are concerned with predicting the time (typically called the "survival time") until a well-defined event of interest, such as death, occurs [18]. Traditional approaches for survival data analysis include non-parametric methods such as the Kaplan-Meier estimator (see Fig. 7); semi-parametric methods such as Cox Proportional Hazards (CPH) models (Fig. 6); and parametric methods such as parametric regression models for survival times. CPH is the most widely used regression technique for survival analysis [16]. It addresses the limitations of purely empirical (e.g., Kaplan-Meier) descriptive methods by modeling the effects of several variables while also enabling the quantification of survival differences between groups [9]. The CPH relies on the proportional hazards assumption which states that the hazard function for the comparison groups is proportional to the hazard functions for exposed people, so that the hazard ratio between them remains constant over time [20]. CPH cannot easily model nonlinearities and interaction effects in data correctly [15]. By contrast, Random Survival Forest (RSF) is a nonparametric ensemble method constructed by fitting multiple decision trees and aggregating their predictions (see Fig. 8). Key advantages of RSF include its ability to evaluate multiple covariates, non-linearities and interactions, and non-reliance on the proportionality assumption [17]. Traditional survival methodologies assume the same probability of the outcomes for all included individuals and do not explicitly address the key issue of inter-individual variability.

Accurately describing the relationship between exposure(s) and outcome(s) is central to policy and regulatory decisions. In many instances however, treatment effects vary considerably over the population, making it difficult to translate inferences from the average population effects to the individual level [3,8]. Understanding heterogeneity of effects requires addressing challenges due to high volume and complexity of individual-level characteristics [12,13]. Furthermore,

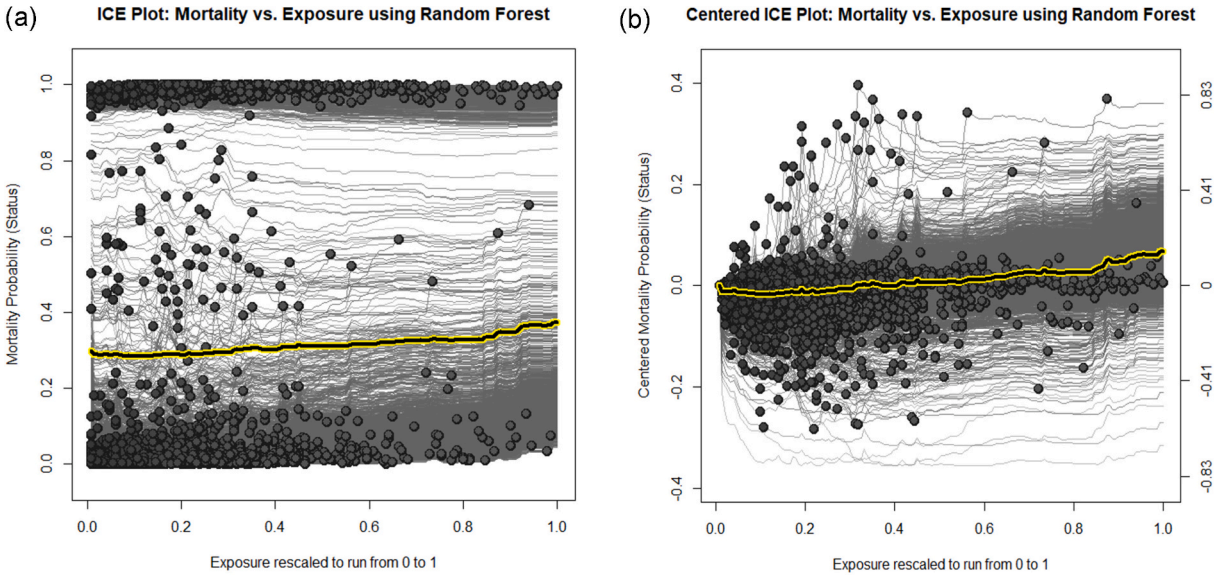

Fig. 5. a. ICE plot (individual curves) and PDP (heavy curve) for mortality probability vs. exposure (BLL).
Fig. 5b. A centered ICE plot for the same information as in Fig. 5.
Source: Appendix C gives R code for generating these figures.

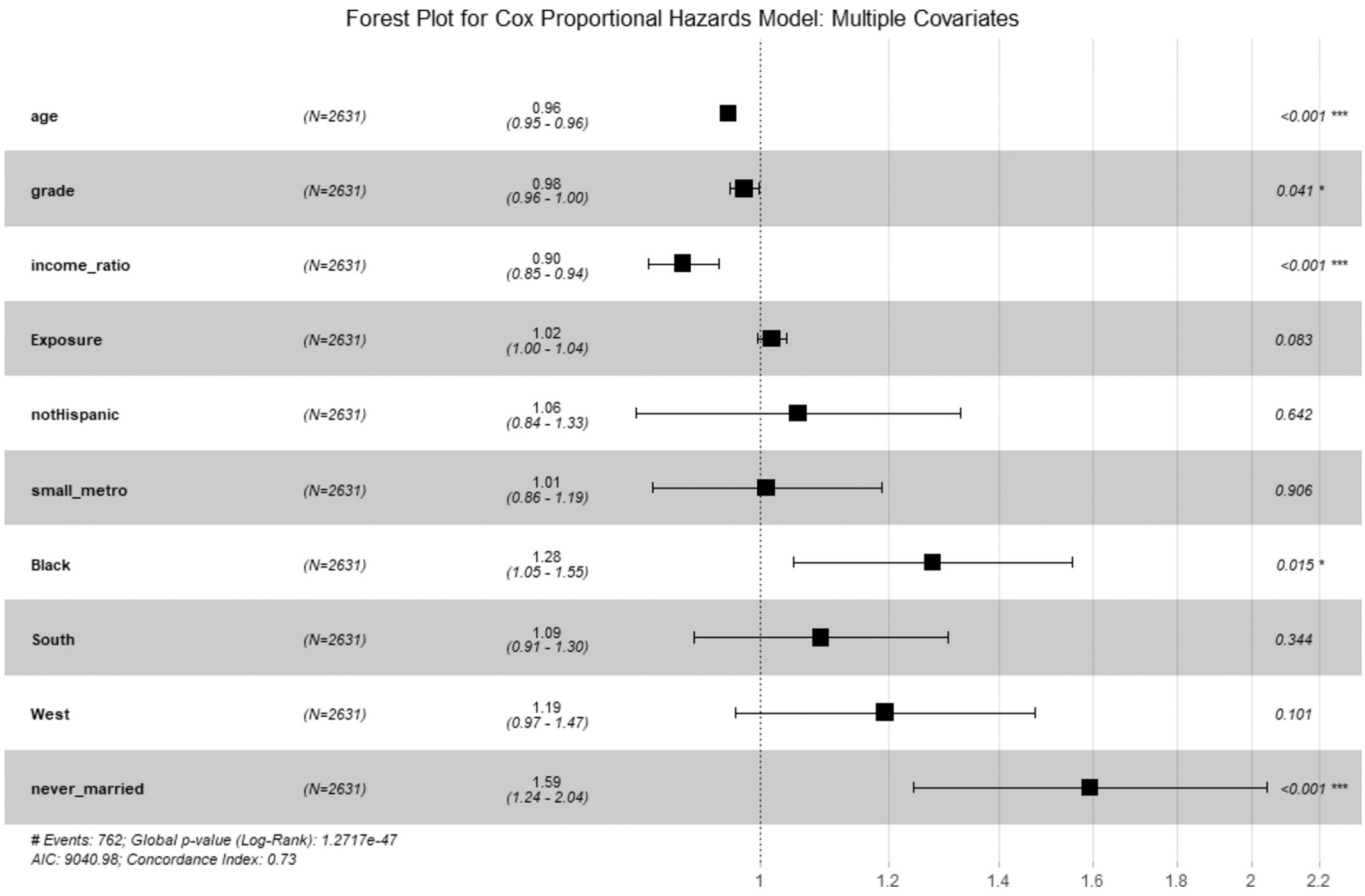

Fig. 6. Forest Plot for Cox Proportional Hazards (CPH) model.
Source: Appendix D gives the R code and a link to the chat that produced it for this figure.

censored survival data has several challenges for use. Therefore, using flexible approaches like survival trees within a counterfactual modeling framework in which exposure alone is varied, holding other variables fixed, is a promising approach as survival trees can uncover complex relationships that are beyond the scope of parametric models and yet are easy to interpret [2,4]. Survival trees (see Fig. 7) are a method that identifies homogenous groups predefined by specific covariates by a recursive binary partitioning of the data into subsets with distinct outcome patterns. This results in smaller subsets of relatively homogeneous outcomes called 'nodes' with the final smallest subset being termed "terminal" or "leaf" nodes [19]. Several different splitting criteria have been previously used within survival trees. Existing statistical techniques such as Kaplan–Meier curve estimates can then be applied to understand the survival distribution in the final subset (terminal node/leaf). Survival trees provide a flexible approach to characterize the effects of covariates and their interactions on outcomes.

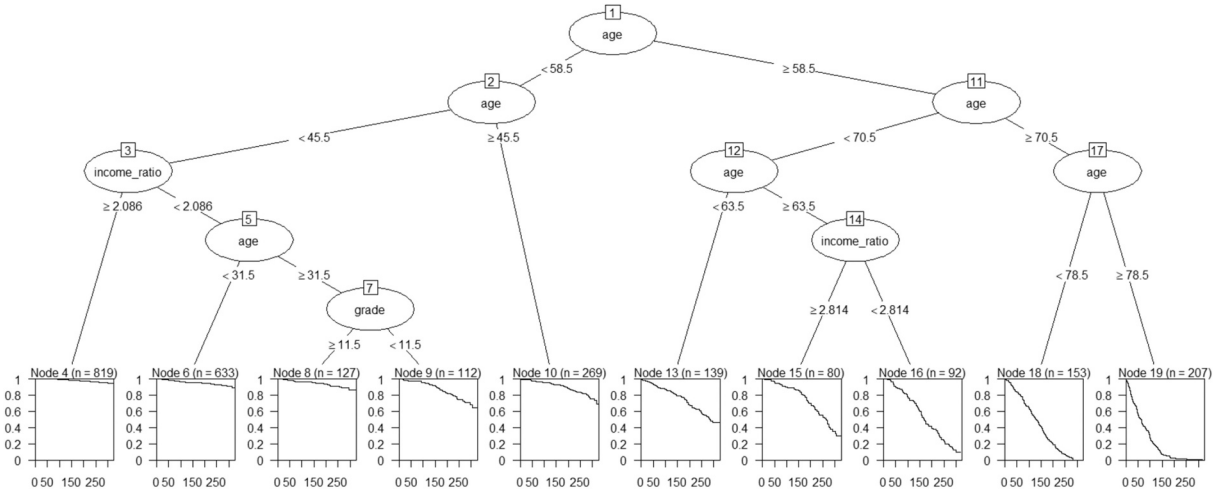

Fig. 7. A survival tree showing empirical (Kaplan-Meier) estimates of survival curves (shown in the bottom-most nodes) for different sets of conditioning information.
Source: Appendix E gives the R code and a link to the chat that produced it for this figure.

Exposure response functions are a key consideration in regulatory risk assessments and policymaking with the assumption that historical exposure-response relationships can be used to predict future impacts of regulation change or interventions. There are fundamental issues that are yet to be fully understood in this context including causal interpretation of historical associations and spatiotemporal differences in the shapes of exposure response curves [6]. If they are not carefully addressed, such issues risk creating inaccurate perceptions of the public health benefits to be expected from tighter regulations. Regression based exposure response curves largely represent the average responses for different levels of exposure in a population. They do not provide clarity on heterogeneity in individual risks or estimate total or direct causal effects [5].

Several non-parametric machine learning methods have been previously validated to provide clarity and precision to the exposure-response curves. These methods have been widely used in other fields and can be applied in health research for a deeper understanding of causality which is not well addressed by existing methods. The partial dependence plots (PDPs) and individual conditional expectation (ICE) plots are model-agnostic tools that can visualize inter-individual variability in exposure-response curves and help interpret the impact of variables in the prediction of a machine learning model [1,5]. A partial dependence plot illustrates the average marginal effects between model inputs and predictions [21]. ICE plots, on the other hand, disaggregate the average exposure response curve into individual exposure response curves to help visualize the individual differences by u]sing each prediction separately [11]. Further details of these computational methods have been described previously [5]. The following sections demonstrate how aggregate exposure-response curves can be "de-averaged" or decomposed into clusters of individual-level exposure-response curves of different shapes using these methods while allowing for time-varying responses modeled by survival curves estimated from data using non-parametric machine-learning methods.

AI-assisted Cox Proportional Hazards modeling

Perhaps the best-known survival data model among epidemiologists and health risk analysts is the Cox Proportional Hazards (CPH) model. Fig. 6 shows a forest plot for the results of CPH modeling, and Appendix D contains the R code and a link to the chat used to produce it. That chat provides the following AI-generated interpretation of the CPH results: "The Cox Proportional Hazards model indicates that higher… education level, and income are associated with a significantly reduced risk of death, while being Black and never having been married are linked to a higher mortality risk. Blood lead levels (BLL) show a marginally significant positive association with mortality ($p = 0.083$), but ethnicity, geographic location, and living in small metropolitan areas do not have a significant impact." However, when prompted to interpret regression diagnostics for the CPH model, ChaGPT warns that "The global Schoenfeld residuals test indicates a violation of the proportional hazards assumption for the overall model, with significant issues for covariates such as age, grade, income ratio, ethnicity (non-Hispanic), and race (Black), while the assumption holds for blood lead levels (BLL). Martingale residuals suggest a generally adequate model fit for continuous covariates, but potential non-linearities should be further explored. Deviance residuals identify possible outliers that could disproportionately affect the model, and DFBeta values highlight influential observations that may require closer examination for data issues or undue influence." Thus, similar to the logistic regression model for mortality probability, the CPH model produces plausible results (Fig. 6), but its assumptions (especially, the proportional hazards assumption) are not well supported for this dataset.

One option is to seek a better-fitting CPH model by introducing interaction terms, time-varying covariates (e.g., effects of age on mortality risk may depend on year of birth), dummy variables for ranges of the variables, and other adjustments. We will instead follow a more flexible modeling strategy by dropping the CPH modeling framework and exploring non-parametric approaches to survival data analysis and analyses of longevity.

AI-assisted survival tree and random survival forest analysis

Fig. 7 shows a survival tree depicting how non-parametric (Kaplan-Meier) survival curves (shown in the bottom-most or "leaf" nodes) vary based on different combinations of information (ranges of values) for the variables in the paths from the root node (*age*, at the top of the tree) to each leaf node. Appendix E presents the R code that produced this tree and gives a link to the chat used to produce the R code. Prompted to explain the tree to someone unfamiliar with Classification and Regression Tree analysis and survival data analysis, ChatGPT generated the following explanation and interpretation:

"The attached image [Fig. 7] represents a survival tree generated using CART (Classification and Regression Tree) analysis to identify subgroups of individuals based on predictors that influence mortality risk. The dataset consists of non-smoking men from NHANES III (1988–1994), and the primary outcome of interest is survival time. The tree identifies key variables that stratify the population into groups with differing survival probabilities over time, represented by Kaplan-Meier curves at the terminal nodes.

The model begins by splitting the population based on **age**. Individuals younger than **58.5 years** are placed in the left branch, while those older than **58.5 years** are placed in the right. Subsequent splits are made using variables such as **income ratio**, **age**, and **education level (grade)**. Each terminal node represents a distinct subgroup based on these variables, with the size of the subgroup (n) indicated, along with a survival curve that depicts the probability of survival over time.

The **x-axis** of each survival curve shows time (in months), while the **y-axis** represents survival probability. Flatter survival curves indicate a lower mortality risk, while steeper curves reflect a higher risk of death over time. Key patterns observed include:

- **Younger individuals** with a high **income ratio** show better survival outcomes, as seen in **Node 4**, which has a flat survival curve and represents individuals with lower mortality risk.
- **Older individuals** and those with lower **income ratios** and **education levels** exhibit steeper survival curves, suggesting higher mortality risk. For instance, **Node 9** shows a steep curve, indicating shorter survival times for middle-aged individuals with lower incomes and education.

Overall, the analysis highlights **age** as the most important predictor of mortality risk, followed by **income ratio** and **education level**. Younger individuals with higher incomes and education have better survival probabilities, whereas older individuals with lower socioeconomic status are at greater risk of mortality."

Although not stated by ChatGPT, another salient feature of Fig. 7 is that *Exposure* does not appear in the tree, suggesting that it is not one of the most important predictors of mortality risk. In addition, there is clearly substantial inter-individual heterogeneity in survival curves, as shown by the wide range of dissimilar curves at the leaf (bottom) nodes of Fig. 7.

Single CART trees can be extended to "random forests," i.e., ensembles of hundreds of trees trained on different subsets of a data frame. Their predictions are then averaged ("bagged") in an effort to obtain more accurate predictions. Random forest predictions for each individual case as an independent variable is varied, holding all other variable values fixed, generate partial dependence plots (PDPs), as in Fig. 2. Similarly, single survival trees can be extended to "survival forest" ensembles of hundreds of survival trees. These can also be used to generate partial dependence plots (PDPs) and ICE plots for the effects of individual predictors on expected survival times or, conversely, cumulative hazard for mortality. Fig. 8 shows the PDPs from a Random Survival

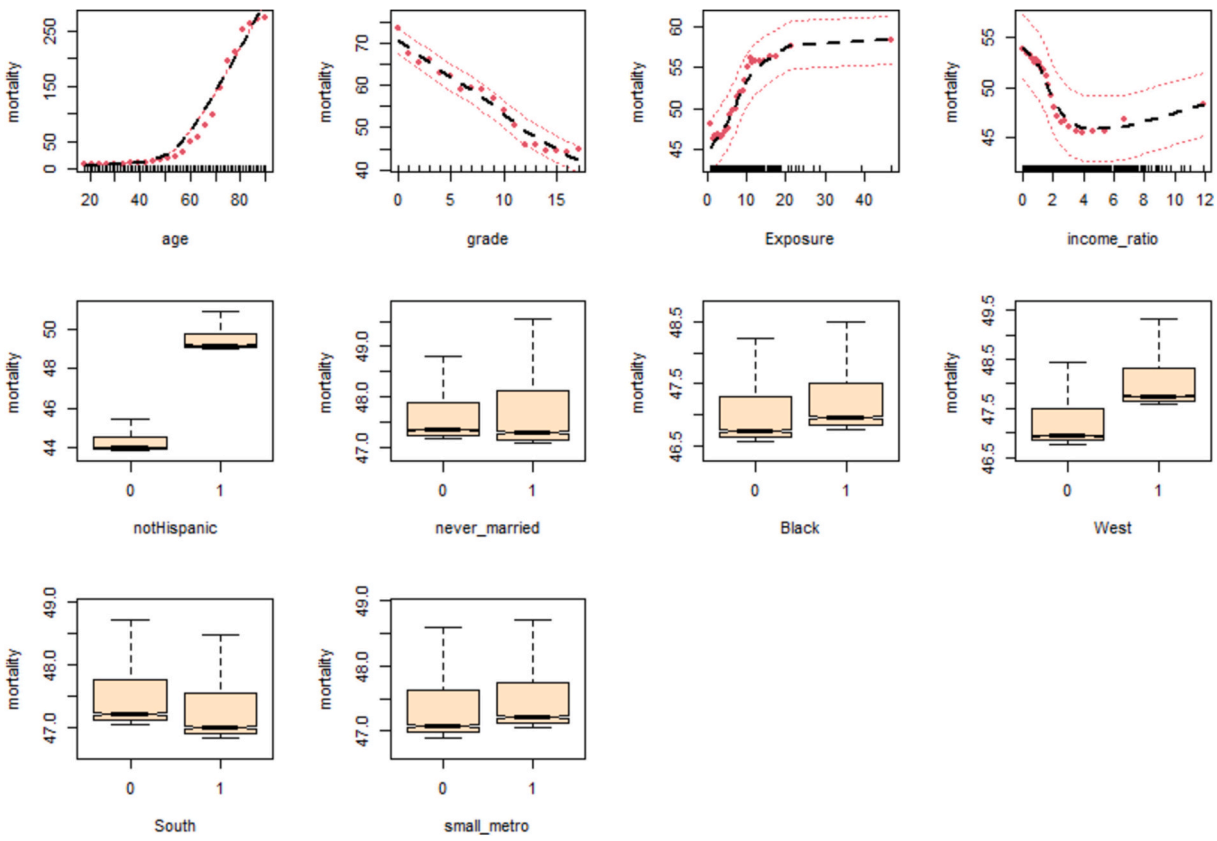

Fig. 8. Partial Dependence Plots (PDPs) for cumulative hazard of mortality, as estimated by Random Survival Forest (RSF) analysis.

Forest for each independent variable, with bootstrapped 95 % uncertainty bands for continuous variables such as *age* and *Exposure*, and box plots for discrete variables such as never married. Fig. 9 shows an ICE plot for predicted median survival times for individuals as *Exposure* is varied. Appendix F contains the code that produced these figures. Asked to explain Fig. 8, ChatGPT generated the following response:

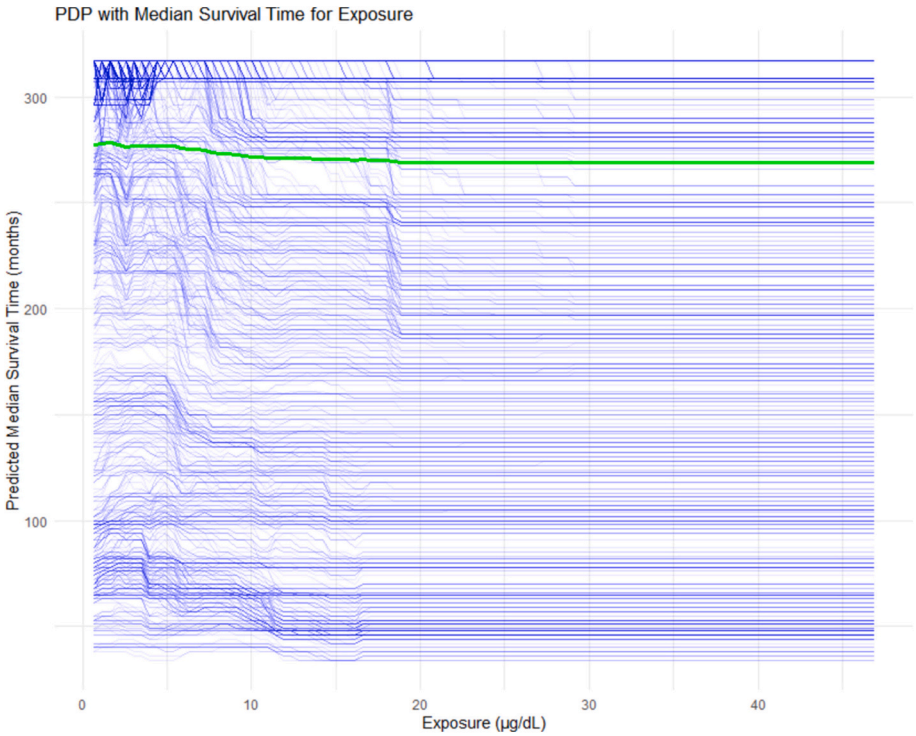

Fig. 9. ICE plots and PDP for median survival times vs. exposure(BLL in µg/dL).
Source: Appendix F gives code for Figs. 8 and 9.

"The partial dependence plots (PDPs) in the top row illustrate the marginal effects of continuous variables (age, education, blood lead level, and income ratio) on predicted mortality risk, averaged over the distribution of all other variables in the model. These plots demonstrate how mortality risk increases significantly with age and blood lead levels, while higher levels of education and income ratio are associated with lower mortality risk. The dashed red uncertainty bands represent variability in predictions, calculated using resampling methods within the random forest survival model. The box plots in the bottom row compare predicted mortality risks across categorical variables, such as ethnicity, marital status, race, and geographic location. For instance, non-Hispanic, never-married, and Black individuals, as well as those living in the South, tend to have higher predicted mortality risks, whereas individuals in the Western US and small metropolitan areas exhibit lower risks. The y-axis represents relative mortality risk, derived from the model's predictions, rather than absolute probabilities or time units, offering an abstract measure of risk over the follow-up period."

Fig. 9 shows how predicted median survival time in months varies with exposure, holding other variables fixed, for different individuals. It shows substantial heterogeneity in individual exposure-response ICE curves around the population average exposure-response PDP curve. The causal interpretation of such curves is not self-evident, since, for example, exposure (BLL) may be correlated with unmeasured confounders such as environmental tobacco smoke or unreported smoking. But even without a clear causal interpretation, it is clear that inter-individual-heterogeneity is large enough so that the green PDP curve for population average exposure-response is not highly representative of most of the individual-level ICE curves.

AI-assisted survival causal survival forest analysis

Causal effects of exposure on survival times can be clarified using causal survival forest analysis. A causal survival forest [8] is an ensemble of causal survival trees. A causal survival tree is similar to a regular survival tree, such as the one in Fig. 7, except that instead of seeking to predict survival times or survival probabilities based on the values of independent variables, a causal survival tree seeks to estimate heterogeneous treatment effects (HTEs), i.e., the individual-level differences in survival times or probabilities made by different levels of exposure. The leaf nodes and splits (interior nodes) in a causal survival tree are selected to try to maximize treatment effect differences instead of trying to maximize a measure of homogeneity in survival outcomes. For the predictions at the leaf nodes to have valid causal interpretations, several key assumptions must hold. A commonly used set of assumptions is the following: (1) No unmeasured confounding; (2) Positivity, meaning that for each individual (or subgroup), there must be a positive probability of receiving each level of exposure. This ensures that there is sufficient variability in treatment levels across the covariates to estimate causal effects. (3) Stable Unit Treatment Value Assumption (SUTVA), meaning that each individual's outcome must be independent of the exposures of others, i.e., there is no interference between individuals. This is critical for ensuring that the estimated treatment effects reflect individual-level causal effects. (4) Correct model specification, meaning that the tree (or forest) should correctly capture the relationship between treatment, covariates, and survival outcomes. (5) The censoring mechanism (e.g., the end date for data collection) must be independent of the exposure and the outcome, conditional on covariates. If these assumptions hold, the causal survival tree provides valid estimates of treatment effects at the leaf nodes.

Fig. 10 shows the estimated causal (if the above assumptions hold) relationship between exposure level (x axis) and mean causal effect on survival time in months over a 10-year horizon (y axis). The dots show the estimated mean causal effect (HTEs) of blood lead levels on survival for groups of individuals with the same exposure levels. Specifically, each dot corresponds to a specific exposure level (blood lead concentration) and shows the average predicted causal effect of that exposure

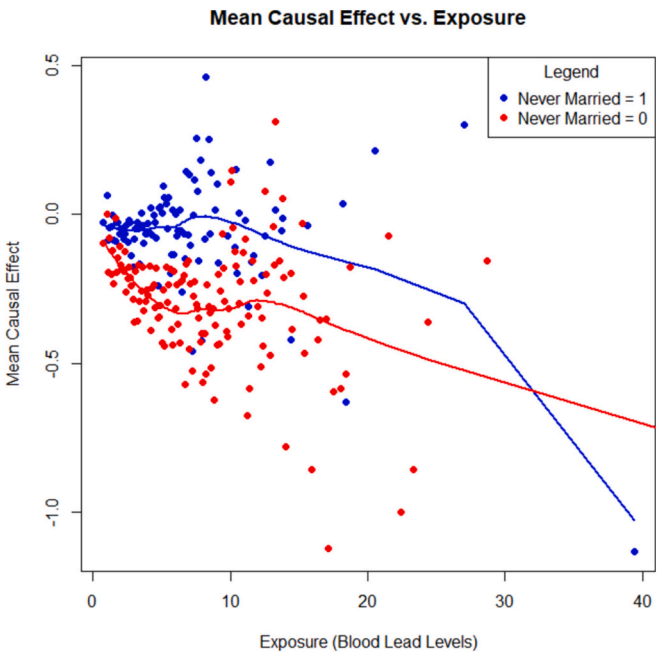

Fig. 10. Results of a Causal Survival Forest analysis of Heterogeneous Treatment Effects (HTEs).
Source: Appendix G gives code for Fig. 10.

level on mean survival time over a horizon of 10 years for individuals with that level of exposure. Red dots represent results for individuals who were never married and blue dots are for individuals who are or have been married. The red and blue curves fit to this scatter plot are LOESS (non-parametric smoothing regression) curves. Exposure causes different estimated effects (HTEs) on the mean survival times of different subsets of individuals, ranging from reductions on the order of about a month over the next 10 years in predicted mean survival times to gains on the order of up to half a month over the next 10 years. With the exception of one outlier at the lower right, it appears that estimated HTEs of lead exposure are less for never-married (blue) than for ever-married (red) people.

Other sources of heterogeneity in the estimated individual-level causal effects (HTEs) of exposure can be explored, e.g., via CART tree or Random Forest analysis. The three most important sources of heterogeneity in HTEs identified in Random Forest analysis (not shown) are age, grade, and income ratio. As shown in Fig. 10, individual differences in never-married status also help to explain heterogeneity in individual HTEs, but differences in age, grade, and income are even more important.

An attractive benefit of AI-assisted data analysis using LLMs is that they are "living" analyses [10], meaning that the LLM remembers them in detail, making it easy for the same or other investigators to return to an analysis and to extend it as new questions arise. Simply resuming the interactive chat used to create an analysis with an LLM, even months or years after it is first done, makes it easy to revisit, interrogate, and update its data, assumptions, reasoning, conclusions, and explanations. For example, an anonymous reviewer of an earlier version of this article asked the following two questions: "1. How were tuning parameters chosen in the survival tree and causal survival forest? 2. Can the authors also present a forest plot using causal survival forest?" Returning after several months to the chat used to develop the causal survival forest and entering those questions elicited the following response from ChatGPT:

"**Addressing Reviewer Comments:**

1. How were tuning parameters chosen in the survival tree and causal survival forest?

Tuning parameters for the **causal survival forest** were chosen based on the default settings of the grf package, which are designed to provide

robust performance for general use cases. However, users can customize these parameters depending on the problem at hand. The default parameters include:

- **num.trees = 2000**: A sufficient number of trees to ensure stability and robustness in predictions. Increasing the number of trees may reduce variance but increases computational cost.
- **min.node.size = 5**: The minimum number of observations in each leaf node, which balances bias and variance.
- **sample.fraction = 0.5**: Half of the data is used for growing each tree.
- **honesty = TRUE**: Ensures that the splits in trees are determined using a disjoint subset of the data to reduce overfitting.
- **horizon = 120 months**: The time point for the restricted mean survival time (RMST) calculation, corresponding to the research question.

No cross-validation or parameter tuning was explicitly performed in the presented analysis, but this can be done by using the tune.parameters argument in the causal_survival_forest function.

2. Can the authors also present a forest plot using the causal survival forest?

A **forest plot** is a common visualization in survival analysis that summarizes the effects of predictors or treatments on outcomes. Here's how to present a forest plot based on the causal survival forest model…" The LLM then generated code which, after slight refinement (e.g., to sort the confidence interval bars), produced Fig. 11 and the following explanation:

"**Explanation of the Forest Plot:**

1. **Covariate Groups**:

   ∘ The y-axis represents combinations of binary covariates (e.g., never.married, small.metro, Black, South) to show how the HTE [Heterogenous Treatment Effect] varies across subpopulations.

2. **Mean Causal Effects**:

   ∘ The x-axis shows the mean causal effect (HTE) for each group.

3. **Confidence Intervals**:

   ∘ Horizontal bars indicate the 95 % confidence intervals for the mean causal effects."

These kinds of responses to follow-up questions to clarify or extend an analysis can greatly increase its value and the return on the investments of time, data, and analysis and expositions skills that went into the initial analysis [10].

Discussion

The following questions about inter-individual heterogeneity in risks are of frequent interest in public health risk assessment and epidemiology.

- *Do some people have greater exposure-associated risks than others, even if their exposures are the same?*
- *If so, how do their individual exposure-response curves differ?*
- *To what extent do differences in covariates predict differences in exposure-response functions?*
- *How would changing exposure (but nothing else, such as income) change the distribution of individual life lengths in the population?*
- *How would changing exposure (but nothing else) change each individual's probability distribution of life length, given the values of other covariates for the individual?*

Computationally practical methods for answering such questions based on data are comparatively new. ICE plot analyses (Figs. 5 and 9) and heterogeneous treatment effects (HTEs) of exposure (Fig. 10) provide promising frameworks for using survival data to answer such

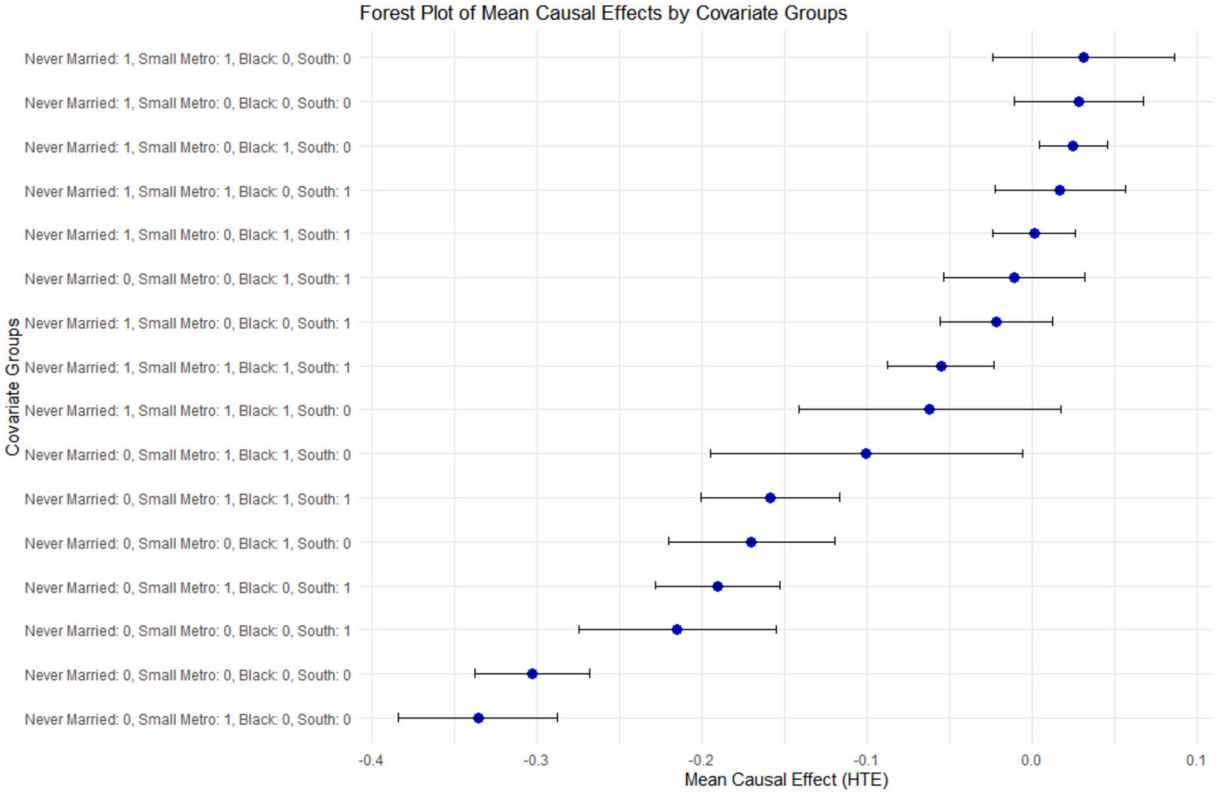

Fig. 11. Forest Plot for Causal Survival Forest, Generated Upon Request.

questions. They show that for the same value of exposure on the x axis, there are very different individual mortality risks. Moreover, the HTEs, as well as the average slopes of the individual exposure-response curves (i.e., the ICE curves) as exposure increases, e.g., from 0 to 10, are very different for different individuals. Since it is not convenient to visualize how entire probability distributions of individual life lengths vary with exposure, Fig. 9 shows only how the median predicted survival time (in months) varies with exposure. If desired, other quantiles can be plotted similarly. Likewise, Fig. 10 shows the estimated changes in life expectancy over the next 10 years caused by different levels of exposure for different individuals. Similar-looking plots can be shown for other time horizons. A useful insight from the non-parametric survival data analysis results based on Random Survival Forest predictions is that they show clearly that exposure is significantly positively associated with increased mortality risk (Figs. 8 and 9). This was not as clear in logistic regression (Fig. 4) or CPH modeling results (Fig. 6).

This paper has illustrated how AI-powered data analysis can enable subject matter experts (SMEs) to quickly complete a variety of analyses to show how predicted individual-level risks vary with exposure, holding other covariates fixed. Disaggregating population exposure-response functions to show inter-individual variability in risk for each level of exposure enables risk managers to understand quantitatively the extent of inequalities in exposure-associated risks and in exposure-response relationships within a population. AI-assisted data analysis makes it relatively easy to produce these highly resolved risk assessments using relatively recent and sophisticated methods such as ICE plots and HTE estimates. The capability to produce advanced analyses easily frees risk analysts to focus on what questions to ask and the extent to which underlying modeling assumptions are appropriate for the data being analyzed, while delegating many of the details of how the answers are computed to reliable R and Python packages that are selected, run, and interpreted under the supervision of an AI.

We have sought to show how AI-assisted data analysis can be used to accelerate and enhance key analytics tasks such as the following:

- *Calculate and visualize individual-level outcomes.* Tools such as Individual Conditional Expectation (ICE) plots and HTEs allow for the visualization of the dependence of individual-level health risks on exposure and other variables. This goes beyond what traditional methods such as regression modeling and Partial Dependence Plots (PDP) do by quantifying average effects in a population. By showing individual-level effects, these methods enable practitioners to better understand heterogeneous responses to exposures, potentially enabling more personalized and precise risk assessments.
- *Check assumptions of statistical models.* Advanced diagnostic tests, such as martingale residual tests for Cox Proportional Hazards (CPH) models, allow practitioners to test and verify (or refute) the assumptions underlying their statistical models. This ensures that the models used are appropriate for the data, improving the validity of the results.
- *Apply more advanced methods* when assumptions of simpler models are found not to hold: When diagnostic tests indicate that the assumptions of simpler models (such as linear or logistic regression) do not hold, more sophisticated techniques (such as Random Forests or Random Survival Forests) can be used instead. These methods are non-parametric. They can handle non-linear relationships and interactions between variables, providing more robust and reliable results.
- *Cross-check validity and robustness of conclusions.* Using multiple analytical methods allows practitioners to cross-validate their findings. By comparing results from different models, they can assess the consistency and robustness of their conclusions, leading to greater confidence in their results.

However, as illustrated in the chats at the links in the appendices, it is currently still prudent – and even essential, in order to achieve trustworthy results – for statistics and ML experts to carefully review AI-generated results and code to assure that statistical software is being used appropriately.

Conclusions and recommendations

This paper has demonstrated the current potential of AI-assisted data analysis to help clarify statistical exposure-response relationships using survival data. ICE plots and sophisticated machine learning analyses using predictors such as Random Survival Forest are now relatively easy to prepare using off-the-shelf statistics packages. Large language models (LLMs) make those packages much more accessible and easier to use than ever before. The resulting individual-level risk predictions provide insights into inter-individual variability in exposure-response relationships that are not easy to quantify and visualize using traditional aggregate statistical regression models.

The illustrative analysis of blood lead levels (BLL) and mortality risk for male nonsmokers in the NHANES III dataset illustrates how non-parametric machine learning methods can reveal inter-individual differences in predicted survival times as functions of exposure, offering a deeper, data-driven perspective on exposure-associated health risks. These findings suggest several key conclusions:

1. **Limitations of Traditional Models**: Logistic regression and Cox Proportional Hazards models, may not detect significant exposure-response relationships, especially in the presence of non-linearities and interactions among variables.
2. **Non-Parametric Methods**: Combining non-parametric prediction methods such as survival forests with ICE plot methods for disaggregating average exposure response curves down to the individual level provides a flexible and powerful approach to modeling survival data. Regulatory agencies and risk management policy- and decision-makers can take advantage of such methods to quantify differences in estimated individual-level exposure-response functions rather than relying solely on population-averaged estimates. Our illustrative analysis of the NHANES III dataset using blood lead levels (BLL) and mortality risk illustrates the value of these methods in providing a deeper, more personalized understanding of risk profiles across populations.
3. **AI-Assisted Analysis**: The use of AI, particularly large language models like ChatGPT, can greatly facilitate the application of machine learning methods and advanced statistical analyses for risk analysts and epidemiologists. We have demonstrated ChatGPT's ability to help conduct complex analyses and generate new insights with minimal user input, making advanced statistical methods more accessible to researchers. It is timely for public health researchers and risk assessors to consider integrating AI-assisted data analysis into their workflows to take advantage of recent machine learning approaches and to help check the assumptions and interpret the results of advanced statistical analyses.
4. **Human-AI partnership**: While AI-assisted data analysis appears very promising, continued validation of results is necessary to ensure accuracy. AI can automate much of the analysis, but expert oversight is critical for ensuring the appropriate application of models and the correct interpretation of results, as illustrated by some of the chats referenced in the appendixes. Researchers should view AI as a valuable tool for assisting human analysis rather than a replacement for expert judgment. Standardizing the use of these tools in health risk assessments might help to improve their credibility and reliability in policy contexts.

In conclusion, AI-assisted data analysis represents a potentially transformative step forward in health risk assessment, offering more granular insights into exposure-response relationships and enabling better-informed decisions for public health and regulatory policies.

AI Acknowledgements

We used OpenAI's ChatGPT throughout the preparation of this paper in performing data analyses, generating code for advanced statistical models, and drafting initial interpretations of results and text explaining their significance. The authors are responsible for all aspects of the final text. We believe that such human-AI collaborations highlight the potential of AI to help increase productivity and advance the quality of research.

CRediT authorship contribution statement

Louis Anthony Cox: Writing – original draft, Software, Project administration, Methodology, Funding acquisition, Formal analysis, Conceptualization. **R. Jeffrey Lewis:** Writing – original draft, Software, Project administration, Methodology, Funding acquisition, Formal analysis, Conceptualization. **Saumitra V. Rege:** Writing – original draft, Software, Project administration, Methodology, Funding acquisition, Formal analysis, Conceptualization. **Shubham Singh:** Writing – original draft, Software, Project administration, Methodology, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors thank two anonymous reviewers for suggestions that improved the final exposition. The work of Cox and Singh was supported by Cox Associates LLC; the work of Rege and Lewis on this paper was not funded. Cox Associates received funding in 2022-2024 from ExxonMobil Biomedical Sciences Inc. (EMBSI) to develop and apply statistical methods to quantify inter-individual variability in exposure-response relationships. Cox Associates also received funding in 2019-2022 from the US Department of Agriculture (USDA) to support research on statistical and machine learning methods of causal inference from observational data. The methods, examples, and applications discussed and conclusions reached are solely those of the authors.

Appendix A. R code for generating an ICE plot

```
simple_ice <− function(data = mtcars) {.
# Load required packages.
if (!requireNamespace("randomForest", quietly = TRUE)) {.
install.packages("randomForest").
}
if (!requireNamespace("ICEbox", quietly = TRUE)) {.
install.packages("ICEbox").
}
library(randomForest).
library(ICEbox).
# Train a Random Forest model on mpg vs. all other variables in the dataset.
model <− randomForest(mpg ~ ., data = data).
# Generate ICE data for the predictor wt.
ice <− ice(object = model, X = data, y = data$mpg, predictor = "wt").
# Plot the ICE curves without centering and plot all curves.
plot(ice, centered = FALSE, frac_to_plot = 1,
xlab = "weight (wt) rescaled to run from 0 to 1",
ylab = "miles per gallon (mpg)").
# Add a title to the plot.
title("ICE Plot: mpg vs. wt using Random Forest").
}
# Example usage:
simple_ice().
```

Appendix B. R code for Fig. 3 (simple logistic regression) and logistic regression analyses

```
# Load necessary libraries.
library(ggplot2).
# Assuming the dataset is already loaded in R as df, where "Exposure" is the variable for blood lead levels (BLL) and "Status" is the mortality status.
# Fit a logistic regression model.
model <− glm(Status ~ Exposure, data = df, family = binomial).
# Create a sequence of Exposure values for prediction.
Exposure_range < − seq(min(df$Exposure), max(df$Exposure), length.out = 100).
# Create a dataframe for prediction.
pred_df < − data.frame(Exposure = Exposure_range).
# Predict mortality probabilities based on the logistic regression model.
pred_df$predicted_prob <− predict(model, newdata = pred_df, type = "response").
# Get confidence intervals for the predictions.
pred <− predict(model, newdata = pred_df, type = "link", se.fit = TRUE).
critval <− 1.96 # Critical value for 95 % confidence intervals.
pred_df$upper <− plogis(pred$fit + critval * pred$se.fit).
pred_df$lower <− plogis(pred$fit - critval * pred$se.fit).
# Plot the logistic regression curve along with observed data points (jittered).
ggplot() +.
geom_point(data = df, aes(x = Exposure, y = Status), color = "red", alpha = 0.5, position = position_jitter(height = 0.02)) +.
geom_line(data = pred_df, aes(x = Exposure, y = predicted_prob), color = "blue") +.
geom_ribbon(data = pred_df, aes(x = Exposure, ymin = lower, ymax = upper), fill = "blue", alpha = 0.2) +.
labs(title = "Logistic Regression: Mortality Probability vs. Blood Lead Levels (Exposure)",
x = "Blood Lead Levels (Exposure) [μg/dL]",
y = "Mortality Probability") +.
theme_minimal().
# Code for regression diagnostics for simple logistic regression model.
# Load necessary libraries.
library(ResourceSelection) # For Hosmer-Lemeshow test.
library(ggplot2).
library(car) # For Cook's distance.
# Fit the logistic regression model.
model <− glm(Status ~ Exposure, data = df, family = binomial).
# 1. Hosmer-Lemeshow Goodness-of-Fit Test.
# Dividing data into deciles of fitted values for the test.
hoslem_test <− hoslem.test(df$Status, fitted(model), g = 10).
print(hoslem_test).
# 2. Residual Analysis: Plot of deviance residuals vs. fitted values.
df$deviance_residuals <− residuals(model, type = "deviance").
ggplot(df, aes(x = fitted(model), y = deviance_residuals)) +.
geom_point(color = "red") +.
geom_hline(yintercept = 0, linetype = "dashed") +.
labs(title = "Deviance Residuals vs. Fitted Values",
x = "Fitted Values", y = "Deviance Residuals") +.
theme_minimal().
# 3. Linearity in the Logit: Plot logit vs. Exposure.
df$logit <− log(fitted(model) / (1 - fitted(model))) # logit of fitted values.
```

```
ggplot(df, aes(x = Exposure, y = logit)) +.
geom_point(color = "blue") +.
geom_smooth(method = "loess", color = "red") +.
labs(title = "Logit (Predicted Probability) vs. Exposure",
x = "Exposure", y = "Logit (Predicted Probability)") +.
theme_minimal().
# 4. Influential Points: Cook's distance.
cooks_d < − cooks.distance(model).
influential_points < − which(cooks_d > (4 / nrow(df))) # Rule of thumb for identifying influential points.
# Plot Cook's distance.
ggplot(data.frame(index = 1:nrow(df), cooks_d = cooks_d), aes(x = index, y = cooks_d)) +.
geom_bar(stat = "identity", fill = "skyblue") +.
geom_hline(yintercept = (4 / nrow(df)), linetype = "dashed", color = "red") +.
labs(title = "Cook's Distance for Influential Points",
x = "Observation Index", y = "Cook's Distance") +.
theme_minimal().
# Print influential points.
print(paste("Influential points:", paste(influential_points, collapse = ", "))).
# Code for simple logistic regression and loess curve (not shown in text).
# Load necessary library.
library(ggplot2).
# Fit the logistic regression model.
model <− glm(Status ~ Exposure, data = df, family = binomial).
# Create a dataframe for prediction.
Exposure_range < − seq(min(df$Exposure), max(df$Exposure), length.out = 100).
pred_df < − data.frame(Exposure = Exposure_range).
pred_df$logistic_pred <− predict(model, newdata = pred_df, type = "response").
# Plot the LOESS curve and logistic regression curve together.
ggplot(df, aes(x = Exposure, y = Status)) +.
geom_point(color = "red", alpha = 0.5) + # Observed data points.
geom_smooth(method = "loess", color = "blue", se = FALSE) + # LOESS curve.
geom_line(data = pred_df, aes(x = Exposure, y = logistic_pred), color = "darkgreen", size = 1.5) + # Thicker, darker logistic regression curve.
labs(title = "LOESS and Logistic Regression Curves for Mortality Risk vs. Exposure",
x = "Exposure (Blood Lead Levels) [μg/dL]",
y = "Mortality Risk (Status)") +.
theme_minimal().
# Code for multivariate logistic regression.
# Fit the multivariate logistic regression model.
model <− glm(Status ~ Exposure + age + grade + income.ratio + notHispanic + Black + small.metro + South + West + never.married,
data = df, family = binomial).
# Summarize the model.
summary(model).
# Calculate odds ratios and confidence intervals.
odds_ratios <− exp.(coef(model)) # Odds Ratios.
conf_int < − exp.(confint(model)) # 95 % Confidence Intervals.
# Extract p-values.
p_values <− summary(model)$coefficients[, "Pr(>|z|)"].
# Combine results into a single dataframe.
results <− data.frame(.
Variable = rownames(summary(model)$coefficients),
Odds_Ratio = odds_ratios,
CI_Lower = conf_int[,1],
CI_Upper = conf_int[,2],
p_value = p_values.
)
# Print the results.
print(results).
# Code for multivariate logistic regression model diagnostics.
# Load necessary libraries.
library(ResourceSelection) # For Hosmer-Lemeshow test.
library(car) # For Variance Inflation Factor (VIF).
library(ggplot2).
# Fit the multivariate logistic regression model.
model <− glm(Status ~ Exposure + age + grade + income.ratio + notHispanic + Black + small.metro + South + West + never.married,
data = df, family = binomial).
# 1. Hosmer-Lemeshow Goodness-of-Fit Test.
hoslem_test <− hoslem.test(df$Status, fitted(model), g = 10).
print(hoslem_test).
# 2. Check for multicollinearity using VIF.
vif_values <− vif(model).
print(vif_values).
# 3. Deviance Residuals vs Fitted Values Plot.
df$deviance_residuals <− residuals(model, type = "deviance").
ggplot(df, aes(x = fitted(model), y = deviance_residuals)) +.
geom_point(color = "red") +.
geom_hline(yintercept = 0, linetype = "dashed") +.
labs(title = "Deviance Residuals vs. Fitted Values",
x = "Fitted Values", y = "Deviance Residuals") +.
theme_minimal().
# 4. Linearity of logit (log odds) with predictors.
df$logit <− log(fitted(model) / (1 - fitted(model))).
ggplot(df, aes(x = Exposure, y = logit)) +.
geom_point(color = "blue") +.
geom_smooth(method = "loess", color = "red") +.
labs(title = "Logit (Predicted Probability) vs. Exposure",
x = "Exposure", y = "Logit (Predicted Probability)") +.
theme_minimal().
# 5. Cook's Distance for influential points.
cooks_d < − cooks.distance(model).
influential_points < − which(cooks_d > (4 / nrow(df))) # Identify influential points.
ggplot(data.frame(index = 1:nrow(df), cooks_d = cooks_d), aes(x = index, y = cooks_d)) +.
geom_bar(stat = "identity", fill = "skyblue") +.
geom_hline(yintercept = (4 / nrow(df)), linetype = "dashed", color = "red") +.
labs(title = "Cook's Distance for Influential Points",
x = "Observation Index", y = "Cook's Distance") +.
theme_minimal().
# Print influential points.
print(paste("Influential points:", paste(influential_points, collapse = ", "))).
# Code for logistic regression forest plot.
# Load necessary libraries.
library(ggplot2).
# Fit the multivariate logistic regression model.
model <− glm(Status ~ Exposure + age + grade + income.ratio + notHispanic + Black + small.metro + South + West + never.married,
data = df, family = binomial).
# Summarize the model.
summary(model).
# Calculate odds ratios and confidence intervals.
odds_ratios <− exp.(coef(model)) # Odds Ratios.
conf_int < − exp.(confint(model)) # 95 % Confidence Intervals.
# Extract p-values.
p_values <− summary(model)$coefficients[, "Pr(>|z|)"].
# Combine results into a single dataframe.
results <− data.frame(.
Variable = rownames(summary(model)$coefficients),
```

```
Odds_Ratio = odds_ratios,
CI_Lower = conf_int[, 1],
CI_Upper = conf_int[, 2],
p_value = p_values.
)
# Create the forest plot using ggplot2.
ggplot(results, aes(x = Variable, y = Odds_Ratio)) +.
geom_point(size = 4, color = "black") + # Larger points for better visibility.
geom_errorbar(aes(ymin = CI_Lower, ymax = CI_Upper), width = 0.2, color = "black") +.
geom_hline(yintercept = 1, linetype = "dashed", color = "grey") + # Reference line at OR = 1.
coord_flip() + # Flip coordinates to make it horizontal.
labs(title = "Forest Plot of Odds Ratios for Mortality Risk",
x = "Variables",
y = "Odds Ratio (95% CI)") +.
theme_minimal() +.
theme(.
plot.title = element_text(size = 18, face = "bold"), # Larger title font.
axis.title.x = element_text(size = 14), # Larger x-axis title font.
axis.title.y = element_text(size = 14), # Larger y-axis title font.
axis.text.x = element_text(size = 12), # Larger x-axis text font.
axis.text.y = element_text(size = 12), # Larger y-axis text font.
legend.position = "none" # Remove legend if not needed.
)
```

Appendix C. R code for ICE plots in Fig. 5

```
simple_ice <− function(data) {.
# Load required packages.
if (!requireNamespace("randomForest", quietly = TRUE)) {.
install.packages("randomForest").
}
if (!requireNamespace("ICEbox", quietly = TRUE)) {.
install.packages("ICEbox").
}
library(randomForest).
library(ICEbox).
# Train a Random Forest model on Status (mortality) vs. all other variables in the dataset.
model <− randomForest(Status ~ ., data = data).
# Generate ICE data for the predictor Exposure.
ice <− ice(object = model, X = data, y = data$Status, predictor = "Exposure").
# Plot the ICE curves without centering and plot all curves.
plot(ice, centered = FALSE, frac_to_plot = 1,
xlab = "Exposure rescaled to run from 0 to 1",
ylab = "Mortality Probability (Status)").
# Add a title to the plot.
title("ICE Plot: Mortality vs. Exposure using Random Forest").
}
# Example usage:
sampled_data <− read.csv("df.csv").
simple_ice(sampled_data).
# To obtain a centered ICE plot, change "centered = FALSE" to "centered = TRUE" above.
```

Appendix D. R Code for Cox Proportional Hazards (CPH) model forest plot in Fig. 6

https://chatgpt.com/share/89c68911-ffdb-49e0-9f22-003775e5b2e3

```
# Quietly install and load necessary packages if not already installed.
if (!require(survival)) {.
install.packages("survival", quiet = TRUE).
library(survival).
}
if (!require(survminer)) {.
install.packages("survminer", quiet = TRUE).
library(survminer).
}
# Load the dataset.
df < − read.csv("df.csv").
# Rename columns to replace '.' with '_'.
names(df) < − gsub("\\.", "_", names(df)).
# Derive the 'deathage' as mentioned in the dataset description.
df$deathage <− df$age + df$survival_time / 12.
# Fit a Cox Proportional Hazards model.
cox_model <− coxph(Surv(deathage, Status) ~ age + grade + income_ratio + Exposure + notHispanic + small_metro + Black + South + West + never_married, data = df).
# Display the summary of the Cox model.
summary(cox_model).
# Visualize hazard ratios using a forest plot with a custom title.
ggforest(cox_model, data = df, main = "Forest Plot for Cox Proportional Hazards Model: Multiple Covariates").
```

Appendix E. R Code for the survival tree in Fig. 7

https://chatgpt.com/share/25011467-cd4f-4cd2-8d99-3baab9492d1c

```
# Load required libraries.
library(survival).
library(rpart).
library(rpart.plot).
library(partykit).
# Read the data.
df < − read.csv("df.csv").
# Rename columns to replace any '.' with '_'.
names(df) < − gsub("\\.", "_", names(df)).
# Fit the survival tree using the rpart method with cp = 0.003.
surv_tree <− rpart(Surv(survival_time, Status) ~ age + grade + income_ratio + Exposure +.
notHispanic + small_metro + Black + South + West + never_married,
data = df, method = "exp", control = rpart.control(cp = 0.003)).
# Plot the survival tree using rpart.plot.
rpart.plot(surv_tree, type = 3, extra = 101, fallen.leaves = TRUE).
# Convert rpart object to partykit object to visualize survival curves at the leaf nodes.
surv_tree_party <− as.party(surv_tree).
# Plot the tree with survival curves at the leaf nodes.
plot(surv_tree_party, tp_args = list(type = "surv")).
```

Appendix F. R code for Survival Forest Analysis

```
# NOTE: This code may take several minutes to run.
# Install the randomForestSRC package if not already installed.
if (!require(randomForestSRC)) {.
install.packages("randomForestSRC").
}
# Load the necessary library.
library(randomForestSRC).
# Load the dataset (assuming it's saved as df.csv).
df < − read.csv("df.csv").
# Rename columns to replace "." with "_", if needed.
names(df) < − gsub("\\.", "_", names(df)).
# Fit a survival random forest model.
surv_forest <− rfsrc(Surv(survival_time, Status) ~ age + grade + income_ratio + Exposure + notHispanic + small_metro + Black + South + West + never_married,
```

```
data = df, ntree = 1000, importance = TRUE).
# Generate Partial Dependence Plots for all variables.
plot.variable(surv_forest, partial = TRUE, smooth.lines = TRUE, show.plots = TRUE).
# Appendix F (cont.) R code for survival forest ICE plots.
# Install necessary packages.
if (!require(randomForestSRC)) {.
install.packages("randomForestSRC").
}
if (!require(ggplot2)) {.
install.packages("ggplot2").
}
if (!require(dplyr)) {.
install.packages("dplyr").
}
# Load the libraries.
library(randomForestSRC).
library(ggplot2).
library(dplyr).
# Load your dataset (assuming it's saved as df.csv).
df < − read.csv("df.csv").
# Rename columns to replace "." with "_", if needed.
names(df) < − gsub("\\.", "_", names(df)).
# Fit a survival random forest model.
surv_forest <− rfsrc(Surv(survival_time, Status) ~ age + grade + income_ratio + Exposure + notHispanic + small_metro + Black + South + West + never_married,
data = df, ntree = 1000, importance = TRUE).
# Function to extract median survival time from the survival curve.
# This function finds the time at which survival probability = 0.5 (median survival time).
extract_median_survival_time < − function(surv_object, newdata) {.
pred <− predict(surv_object, newdata = newdata).
apply(pred$survival, 1, function(surv_probs) {.
time_points < − pred$time.interest.
# Find the first time point where survival probability drops below 0.5.
median_time_index <− which.min(abs(surv_probs - 0.5)).
time_points[median_time_index].
})
}
# Set up a grid of values for the variable you want to generate PDPs for (e.g., Exposure).
exposure_values <− seq(min(df$Exposure), max(df$Exposure), length.out = 100).
# Create an empty list to store predictions.
pdp_data <− list().
# Loop through each individual in the dataset and calculate the median survival time for each exposure level.
for (i in 1:nrow(df)) {.
# Create a copy of the original dataset and modify the "Exposure" column for each grid value.
newdata <− df[rep(i, length(exposure_values)),].
newdata$Exposure <− exposure_values.
# Extract median survival times for each exposure value.
median_survival_times <− extract_median_survival_time(surv_forest, newdata).
# Store the results in a dataframe.
pdp_data[[i]] < − data.frame(id = i, Exposure = exposure_values, Predicted = median_survival_times).
}
# Combine all PDP data into a single dataframe.
pdp_df < − do.call(rbind, pdp_data).
# Calculate the average median survival time at each level of Exposure (PDP curve).
pdp_avg_df < − pdp_df % > %.
group_by(Exposure) % > %.
summarize(Predicted = mean(Predicted, na.rm. = TRUE)) # Taking the mean of the median times.
# Plot the PDP with Median Survival Time on the y-axis.
ggplot() +.
geom_line(data = pdp_df, aes(x = Exposure, y = Predicted, group = id), alpha = 0.1, color = "blue") +.
geom_line(data = pdp_avg_df, aes(x = Exposure, y = Predicted), color = "green", size = 1.2) +.
theme_minimal() +.
labs(title = "PDP with Median Survival Time for Exposure", x = "Exposure (μg/dL)", y = "Predicted Median Survival Time (months)").
```

Appendix G. R code for Causal Survival Forest Analysis

```
# Install and load necessary packages.
if (!require(grf)) install.packages("grf", dependencies = TRUE).
if (!require(dplyr)) install.packages("dplyr", dependencies = TRUE).
library(grf).
library(dplyr).
# Assuming you have a dataset 'df' loaded with the following columns:
# "age", "grade", "income.ratio", "Exposure", "survival.time", "Status", "notHispanic",
# "small.metro", "Black", "South", "West", "never.married".
# Remove rows with missing values from the dataset to ensure matching lengths.
df_clean <− df[complete.cases(df[, c("age", "grade", "income.ratio", "notHispanic", "small.metro",
"Black", "South", "West", "never.married", "survival.time",
"Exposure", "Status")]),].
# Ensure Y, W, and D are numeric vectors and X is a matrix.
X < - as.matrix(df_clean[, c("age", "grade", "income.ratio", "notHispanic", "small.metro",
"Black", "South", "West", "never.married")]).
Y < - as.numeric(df_clean$survival.time) # Event time (survival time in months).
W < - as.numeric(df_clean$Exposure) # Treatment assignment (Blood Lead Levels).
D < - as.numeric(df_clean$Status) # Event type (1: death, 0: censored).
# Define the time horizon (set it to 120 months).
horizon <− 120.
# Fit the causal survival forest.
cs_forest <− causal_survival_forest(X = X, Y = Y, W = W, D = D, horizon = horizon).
# Predict causal effects on out-of-bag (OOB) samples.
cs_pred_oob < − predict(cs_forest).
# Create a data frame to store Exposure and corresponding predictions, including 'never.married'.
df_results <− data.frame(.
Exposure = df_clean$Exposure,
Causal_Effect = cs_pred_oob$predictions,
Never_Married = df_clean$never.married.
)
# Group by Exposure levels and calculate the mean causal effect.
df_mean_effects <− df_results % > %.
group_by(Exposure) % > %.
summarize(Mean_Causal_Effect = mean(Causal_Effect)).
# Assign colors based on 'never_married' status: red for 'never_married = 1', blue otherwise.
dot_colors <− ifelse(df_results$Never_Married == 1, "red", "blue").
# Fit a LOESS curve to the data and calculate standard errors.
loess_fit <− loess(Mean_Causal_Effect ~ Exposure, data = df_mean_effects, span = 0.75).
loess_pred <− predict(loess_fit, se = TRUE).
```

```
# Add the fitted values and standard errors to the dataframe.
df_mean_effects$Fitted <− loess_pred$fit.
df_mean_effects$SE < - loess_pred$se.fit.
# Calculate the 95 % confidence interval.
df_mean_effects$Upper_CI < - df_mean_effects$Fitted +1.96 * df_mean_effects$SE.
df_mean_effects$Lower_CI < - df_mean_effects$Fitted - 1.96 * df_mean_effects$SE.
# Plot the LOESS smoothed curve with grey uncertainty bands.
plot(df_mean_effects$Exposure, df_mean_effects$Mean_Causal_Effect,
xlab = "Exposure (Blood Lead Levels)",
ylab = "Mean Causal Effect",
main = "Mean Causal Effect vs. Exposure (LOESS Curve with 95% CI)",
col. = "blue", pch = 16).
# Add the LOESS fitted curve.
lines(df_mean_effects$Exposure, df_mean_effects$Fitted, col. = "red", lwd = 2).
# Add the uncertainty bands (95% CI) with grey shading.
polygon(c(df_mean_effects$Exposure, rev(df_mean_effects$Exposure)),
c(df_mean_effects$Upper_CI, rev(df_mean_effects$Lower_CI)),
col. = rgb(0.5, 0.5, 0.5, 0.4), border = NA) # Grey shaded area for CI.
# Plot the individual dots with colors based on 'never.married' status.
points(df_results$Exposure, df_results$Causal_Effect, col. = dot_colors, pch = 16).
# Comparison of estimated HTEs by never_married status.
# Install and load necessary packages.
if (!require(grf)) install.packages("grf", dependencies = TRUE).
if (!require(dplyr)) install.packages("dplyr", dependencies = TRUE).
library(grf).
library(dplyr).
# Assuming you have a dataset 'df' loaded with the following columns:
# "age", "grade", "income.ratio", "Exposure", "survival.time", "Status", "notHispanic",
# "small.metro", "Black", "South", "West", "never.married".
# Remove rows with missing values from the dataset to ensure matching lengths.
df_clean <− df[complete.cases(df[, c("age", "grade", "income.ratio", "notHispanic", "small.metro",
"Black", "South", "West", "never.married", "survival.time",
"Exposure", "Status")]),].
# Function to process and plot for a given 'never.married' group.
plot_group <− function(df_filtered, color, line_color, add = FALSE) {.
# Ensure Y, W, and D are numeric vectors and X is a matrix.
X < - as.matrix(df_filtered[, c("age", "grade", "income.ratio", "notHispanic", "small.metro",
"Black", "South", "West", "never.married")]).
Y < - as.numeric(df_filtered$survival.time) # Event time (survival time in months).
W < - as.numeric(df_filtered$Exposure) # Treatment assignment (Blood Lead Levels).
D < - as.numeric(df_filtered$Status) # Event type (1: death, 0: censored).
# Define the time horizon (set it to 120 months).
horizon <− 120.
# Fit the causal survival forest.
cs_forest <− causal_survival_forest(X = X, Y = Y, W = W, D = D, horizon = horizon).
# Predict causal effects on out-of-bag (OOB) samples.
cs_pred_oob < − predict(cs_forest).
# Create a data frame to store Exposure and corresponding predictions.
df_results <− data.frame(.
Exposure = df_filtered$Exposure,
Causal_Effect = cs_pred_oob$predictions.
)
# Group by Exposure levels and calculate the mean causal effect for this group.
df_mean_effects <− df_results % > %.
group_by(Exposure) % > %.
summarize(Mean_Causal_Effect = mean(Causal_Effect)).
# Fit a LOESS curve to the data and calculate standard errors.
loess_fit <− loess(Mean_Causal_Effect ~ Exposure, data = df_mean_effects, span = 0.75).
loess_pred <− predict(loess_fit, se = TRUE).
# Add the fitted values and standard errors to the dataframe.
df_mean_effects$Fitted <− loess_pred$fit.
df_mean_effects$SE < - loess_pred$se.fit.
# Calculate the 95 % confidence interval.
df_mean_effects$Upper_CI < - df_mean_effects$Fitted +1.96 * df_mean_effects$SE.
df_mean_effects$Lower_CI < - df_mean_effects$Fitted - 1.96 * df_mean_effects$SE.
# Plot or add the points and LOESS curve.
if (add) {.
# Add points and LOESS curve to an existing plot.
points(df_mean_effects$Exposure, df_mean_effects$Mean_Causal_Effect, col. = color, pch = 16).
lines(df_mean_effects$Exposure, df_mean_effects$Fitted, col. = line_color, lwd = 2).
} else {.
# Create a new plot.
plot(df_mean_effects$Exposure, df_mean_effects$Mean_Causal_Effect,
xlab = "Exposure (Blood Lead Levels)",
ylab = "Mean Causal Effect",
main = "Mean Causal Effect vs. Exposure",
col. = color, pch = 16).
lines(df_mean_effects$Exposure, df_mean_effects$Fitted, col. = line_color, lwd = 2).
}
}
# Filter for individuals with never_married = 1 (blue) and never_married = 0 (red).
df_never_married_1 < − df_clean % > % filter(never.married == 1).
df_never_married_0 < − df_clean % > % filter(never.married == 0).
# Plot for never_married = 1 (blue).
plot_group(df_never_married_1, color = "blue", line_color = "blue").
# Overlay plot for never_married = 0 (red).
plot_group(df_never_married_0, color = "red", line_color = "red", add = TRUE).
# Add legend to explain what the red and blue dots represent.
legend("topright", legend = c("Never Married = 1", "Never Married = 0"),
col. = c("blue", "red"), pch = 16, title = "Legend").
```

References

[1] Angelini M, Blasilli G, Lenti S, Santucci G. A visual analytics conceptual framework for Explorable and steerable partial dependence analysis. IEEE Trans Vis Comput Graph 2024 Aug;30(8):4497–513. https://doi.org/10.1109/TVCG.2023.3263739. Epub 2024 Jul 1. PMID: 37027262.

[2] Bertsimas D, Dunn J, Gibson E, Orfanoudaki A. Optimal survival trees Machine learning 2022;111(8):2951–3023.

[3] Caron A, Baio G, Manolopoulou I. Shrinkage Bayesian causal forests for heterogeneous treatment effects estimation. J Comput Graph Stat 2022;31(4): 1202–14. https://doi.org/10.1080/10618600.2022.2067549.

[4] Carrizosa E, Molero-Río C, Romero Morales D. Mathematical optimization in classification and regression trees. TOP 2021;29:5–33. https://doi.org/10.1007/s11750-021-00594-1.

[5] Cox LA Jr. What is an exposure-response curve? Glob Epidemiol 2023 Jun15;6: 100114. doi: 10.1016/j.gloepi.2023.100114. PMID: 37637716; PMCID: PMC10445976.

[6] Cox Jr LA. Quantitative risk analysis of air pollution health effects. New York, NY: Springer; 2021.
[7] Cox Jr LA. Using Bayesian networks to clarify interpretation of exposure-response regression coefficients: blood lead-mortality association as an example. Crit Rev Toxicol 2020 Aug;50(7):539–50. https://doi.org/10.1080/10408444.2020.1787329 [PMID: 32903110].
[8] Cui Y, Kosorok MR, Sverdrup E, Wager S, Zhu R. Estimating heterogeneous treatment effects with right-censored data via causal survival forests. J R Stat Soc Series B Stat Methodology 2023 Apr;85(2):179–211. https://doi.org/10.1093/jrsssb/qkac001.
[9] Deo SV, Deo V, Sundaram V. Survival analysis-part 2: Cox proportional hazards model. Indian J Thorac Cardiovasc Surg 2021 Mar;37(2):229–33. https://doi.org/10.1007/s12055-020-01108-7. Epub 2021 Jan 2. PMID: 33642726; PMCID: PMC7876211.
[10] Goble R, Bier VM. Risk assessment can be a game-changing information technology–but too often it isn't. Risk Anal 2013 Nov;33(11):1942–51. https://doi.org/10.1111/risa.12055. Epub 2013 Apr 24. PMID: 23614689.
[11] Goldstein A, Kapelner A, Bleich J, Pitkin E. Peeking inside the black box: visualizing statistical learning with plots of individual conditional expectation. J Comput Graph Stat 2015;24(1):44–65. https://doi.org/10.1080/10618600.2014.907095.
[12] Hu L, Ji J, Li F. Estimating heterogeneous survival treatment effect in observational data using machine learning. Stat Med 2021;40(21):4691–713.
[13] Kent DM, Rothwell PM, Ioannidis JP, Altman DG, Hayward RA. Assessing and reporting heterogeneity in treatment effects in clinical trials: a proposal. Trials 2010;11:1–11.
[14] Kline RB. Principles and practice of structural equation modeling. 5th ed. New York, NY: The Guilford Press; 2023.
[15] Moncada-Torres A, van Maaren MC, Hendriks MP, Siesling S, Geleijnse G. Explainable machine learning can outperform Cox regression predictions and provide insights in breast cancer survival. Sci Rep 2021 Mar 26;11(1):6968. https://doi.org/10.1038/s41598-021-86327-7. PMID: 33772109; PMCID: PMC7998037.
[16] Moolgavkar SH, Chang ET, Watson HN, Lau EC. An assessment of the Cox proportional hazards regression model for epidemiologic studies. Risk Anal 2018 Apr;38(4):777–94. https://doi.org/10.1111/risa.12865. Epub 2017 Nov 23. PMID: 29168991.
[17] Pickett KL, Suresh K, Campbell KR, Davis S, Juarez-Colunga E. Random survival forests for dynamic predictions of a time-to-event outcome using a longitudinal biomarker. BMC Med Res Methodol 2021 Oct 17;21(1):216. https://doi.org/10.1186/s12874-021-01375-x. PMID: 34657597; PMCID: PMC8520610.
[18] Schober P, Vetter TR. Survival analysis and interpretation of time-to-event data: the tortoise and the hare. Anesth Analg 2018 Sep;127(3):792–8. https://doi.org/10.1213/ANE.0000000000003653. PMID: 30015653; PMCID: PMC6110618.
[19] Zhou Y, McArdle JJ. Rationale and Applications of Survival Tree and Survival Ensemble Methods. Psychometrika 2015 Sep;80(3):811–33. https://doi.org/10.1007/s11336-014-9413-1. Epub 2014 Sep 17. PMID: 25228495; PMCID: PMC4409541.
[20] Dessai S, Patil V. Testing and interpreting assumptions of Cox regression analysis. Cancer Res Stat Treat 2019;2(1):108–11. https://doi.org/10.4103/CRST.CRST_40_19.
[21] Friedman JH. Greedy function approximation: A gradient boosting machine. Ann Stat 2001;29(5):1189–232. https://doi.org/10.1214/aos/1013203451.