Chapter 8
Causal Mediation Analysis Using R

K. Imai, L. Keele, D. Tingley, and T. Yamamoto

Abstract Causal mediation analysis is widely used across many disciplines to investigate possible causal mechanisms. Such an analysis allows researchers to explore various causal pathways, going beyond the estimation of simple causal effects. Recently, Imai et al. (2008) [3] and Imai et al. (2009) [2] developed general algorithms to estimate causal mediation effects with the variety of data types that are often encountered in practice. The new algorithms can estimate causal mediation effects for linear and nonlinear relationships, with parametric and nonparametric models, with continuous and discrete mediators, and with various types of outcome variables. In this paper, we show how to implement these algorithms in the statistical computing language **R**. Our easy-to-use software, **mediation**, takes advantage of the object-oriented programming nature of the **R** language and allows researchers to estimate causal mediation effects in a straightforward manner. Finally, **mediation** also implements sensitivity analyses which can be used to formally assess the robustness of findings to the potential violations of the key identifying assumption. After describing the basic structure of the software, we illustrate its use with several empirical examples.

Kosuke Imai
Department of Politics, Princeton University, Princeton, NJ 08544, USA
e-mail: kimai@princeton.edu

Luke Keele
Department of Political Science, Ohio State University, Columbus, OH 43210, USA
e-mail: keele.4@polisci.osu.edu

Dustin Tingley
Department of Politics, Princeton University, Princeton, NJ 08544, USA
e-mail: dtingley@princeton.edu

Teppei Yamamoto
Department of Politics, Princeton University, Princeton, NJ 08544, USA
e-mail: tyamamot@princeton.edu

8.1 Introduction

Causal mediation analysis is important for quantitative social science research because it allows researchers to identify possible causal mechanisms, thereby going beyond the simple estimation of causal effects. As social scientists, we are often interested in empirically testing a theoretical explanation of a particular causal phenomenon. This is the primary goal of causal mediation analysis. Thus, causal mediation analysis has a potential to overcome the common criticism of quantitative social science research that it only provides a black-box view of causality.

Recently, Imai et al. (2008) [3] and Imai et al. (2009) [2] developed general algorithms for the estimation of causal mediation effects with a wide variety of data that are often encountered in practice. The new algorithms can estimate causal mediation effects for linear and nonlinear relationships, with parametric and nonparametric models, with continuous and discrete mediators, and with various types of outcome variables. These papers [3, 2] also develop sensitivity analyses which can be used to formally assess the robustness of findings to the potential violations of the key identifying assumption. In this paper, we describe the easy-to-use software, **mediation**, which allows researchers to conduct causal mediation analysis within the statistical computing language **R** [8]. We illustrate the use of the software with some of the empirical examples presented in Imai et al. [2].

8.1.1 Installation and Updating

Before we begin, we explain how to install and update the software. First, researchers need to install **R** which is available freely at the Comprehensive R Archive Network (`http://cran.r-project.org`). Next, open **R** and then type the following at the prompt:

```
R> install.packages("mediation")
```

Once **mediation** is installed, the following command will load the package:

```
R> library("mediation")
```

Finally, to update **mediation** to its latest version, try the following command:

```
R> update.packages("mediation")
```

8.2 The Software

In this section, we give an overview of the software by describing its design and architecture. To avoid duplication, we do not provide the details of the methods that are implemented by **mediation** and the assumptions that underline them. Readers are encouraged to read Imai et al. [3, 2] for more information about the methodology implemented in **mediation**.

8.2.1 Overview

The methods implemented via **mediation** rely on the following identification result obtained under the sequential ignorability assumption of Imai et al. [3]:

$$\bar{\delta}(t) = \int\int \mathbb{E}(Y_i \mid M_i = m, T_i = t, X_i = x) \\ \left\{ dF_{M_i|T_i=1,X_i=x}(m) - dF_{M_i|T_i=0,X_i=x}(m) \right\} dF_{X_i}(x), \tag{8.1}$$

$$\bar{\zeta}(t) = \int\int \{\mathbb{E}(Y_i \mid M_i = m, T_i = 1, X_i = x) \\ -\mathbb{E}(Y_i \mid M_i = m, T_i = 0, X_i = x)\} \, dF_{M_i|T_i=t,X_i=x}(m) \, dF_{X_i}(x), \tag{8.2}$$

where $\bar{\delta}(t)$ and $\bar{\zeta}(t)$ are the average causal mediation and average (natural) direct effects, respectively, and (Y_i, M_i, T_i, X_i) represents the observed outcome, mediator, treatment, and pretreatment covariates. The sequential ignorability assumption states that the observed mediator status is as if randomly assigned conditional on the randomized treatment variable and the pretreatment covariates. Causal mediation analysis under this assumption requires two statistical models: one for the mediator $f(M_i \mid T_i, X_i)$ and the other for the outcome variable $f(Y_i \mid T_i, M_i, X_i)$. (Note that we use the empirical distribution of X_i to approximate F_{X_i}.) Once these models are chosen and fitted by researchers, then **mediation** will compute the estimated causal mediation and other relevant estimates using the algorithms proposed in Imai et al. [2]. The algorithms also produce confidence intervals based on either a nonparametric bootstrap procedure (for parametric or nonparametric models) or a quasi-Bayesian Monte Carlo approximation (for parametric models).

Figure 8.1 graphically illustrates the three steps required for a mediation analysis. The first step is to fit the mediator and outcome models using, for example, regression models with the usual `lm()` or `glm()` functions. In the second step, the analyst takes the output objects from these models, which in Figure 8.1 we call `model.m` and `model.y`, and use them as inputs for the main function, `mediate()`. This function then estimates the causal mediation effects, direct effects, and total effect along with their uncertainty estimates. Finally, sensitivity analysis can be conducted via the function `medsens()` which takes the output of `mediate()` as an input. For the output of the

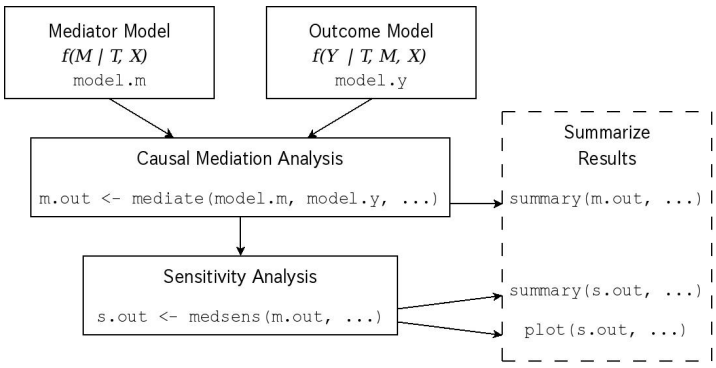

Fig. 8.1 Diagram illustrating the use of the software **mediation**. Users first fit the mediator and outcome models. Then, the function `mediate()` conducts causal mediation analysis while `medsens()` implements sensitivity analysis. The functions `summary()` and `plot()` help users interpret the results of these analyses.

`mediate()` function, a `summary()` method reports its key results in tabular form. For the output of the `medsens()` function, there are both `summary()` and `plot()` functions to display numerical and graphical summaries of the sensitivity analysis, respectively.

8.2.2 Estimation of the Causal Mediation Effects

Estimation of the causal mediation effects is based on Algorithms 1 and 2 of Imai et al. [2]. These are general algorithms in that they can be applied to any parametric (Algorithm 1 or 2) or semi/nonparametric models (Algorithm 2) for the mediator and outcome variables. Here, we briefly describe how these algorithms have been implemented in **mediation** by taking advantage of the object-oriented nature of the **R** programming language.

Algorithm 1 for Parametric Models

We begin by explaining how to implement Algorithm 1 of Imai et al. [2] for standard parametric models. First, analysts fit parametric models for the mediator and outcome variables. That is, we model the observed mediator M_i given the treatment T_i and pretreatment covariates X_i. Similarly, we model the observed outcome Y_i given the treatment, mediator, and pretreatment covariates. For example, to implement the Baron–Kenny procedure [1] in **mediation**, linear models are fitted for both the mediator and outcome models using the `lm()` command.

The model objects from these two parametric models form the inputs for the `mediate()` function. The user must also supply the names for the mediator and outcome variables along with how many simulations should be used for inference, and whether the mediator variable interacts with the treatment variable in the outcome model. Given these model objects, the estimation proceeds by simulating the model parameters based on their approximate asymptotic distribution (i.e., the multivariate normal distribution with the mean equal to the parameter estimates and the variance equal to the asymptotic variance estimate), and then computing causal mediation effects of interest for each parameter draw (e.g., using equations (8.1) and (8.2) for average causal mediation and (natural) direct effects, respectively). This method of inference can be viewed as an approximation to the Bayesian posterior distribution due to the Bernstein–von Mises Theorem [6]. The advantage of this procedure is that it is relatively computationally efficient (when compared to Algorithm 2).

We take advantage of the object-oriented nature of the **R** programming language at several steps in the function `mediate()`. For example, functions like `coef()` and `vcov()` are useful for extracting the point and uncertainty estimates from the model objects to form the multivariate normal distribution from which the parameter draws are sampled. In addition, the computation of the estimated causal mediation effects of interest requires the prediction of the mediator values under different treatment regimes as well as the prediction of the outcome values under different treatment and mediator values. This can be done by using `model.frame()` to set the treatment and/or mediator values to specific levels while keeping the values of the other variables unchanged. We then use the `model.matrix()` and matrix multiplication with the distribution of simulated parameters to compute the mediation and direct effects. The main advantage of this approach is that it is applicable to a wide range of parametric models and allows us to avoid coding a completely separate function for different models.

Algorithm 2 for Non/Semiparametric Inference

The disadvantage of Algorithm 1 is that it cannot be easily applied to non and semiparametric models. For such models, Algorithm 2, which is based on nonparametric bootstrap, can be used although it is more computationally intensive. Algorithm 2 may also be used for the usual parametric models. Specifically, in Algorithm 2, we resample the observed data with replacement. Then, for each of the bootstrapped samples, we fit both the outcome and mediator models and compute the quantities of interest. As before, the computation requires the prediction of the mediator values under different treatment regimes as well as the prediction of the outcome values under different treatment and mediator values. To take advantage of the object-oriented nature of the **R** language, Algorithm 2 relies on the `predict()` function to compute these predictions, while we again manipulate the treatment and me-

diator status using the `model.frame()` function. This process is repeated a large number of times and returns a bootstrap distribution of the mediation, direct, and total effects. We use the percentiles of the bootstrap distribution for confidence intervals. Thus, Algorithm 2 allows analysts to estimate mediation effects with more flexible model specifications or to estimate mediation effects for quantiles of the distribution.

8.2.3 Sensitivity Analysis

Causal mediation analysis relies on the sequential ignorability assumption that cannot be directly verified with the observed data. The assumption implies that the treatment is ignorable given the observed pretreatment confounders and that the mediator is ignorable given the observed treatment and the observed pretreatment covariates. In order to probe the plausibility of such a key identification assumption, analysts must perform a sensitivity analysis [9]. Unfortunately, it is difficult to construct a sensitivity analysis that is generally applicable to any parametric or nonparametric model. Thus, Imai et al. [3, 2] develop sensitivity analyses for commonly used parametric models, which we implement in **mediation**.

The Baron–Kenny Procedure

Imai et al. [3] develop a sensitivity analysis for the Baron–Kenny procedure and Imai et al. [2] generalize it to the linear structural equation model (LSEM) with an interaction term. This general model is given by

$$M_i = \alpha_2 + \beta_2 T_i + \xi_2^\top X_i + \varepsilon_{i2}, \tag{8.3}$$

$$Y_i = \alpha_3 + \beta_3 T_i + \gamma M_i + \kappa T_i M_i + \xi_3^\top X_i + \varepsilon_{i3}, \tag{8.4}$$

where the sensitivity parameter is the correlation between ε_{i2} and ε_{i3}, which we denote by ρ. Under sequential ignorability, ρ is equal to zero and thus the magnitude of this correlation coefficient represents the departure from the ignorability assumption (about the mediator). Note that the treatment is assumed to be ignorable as it would be the case in randomized experiments where the treatment is randomized but the mediator is not. Theorem 2 of [2] shows how the average causal mediation effects change as a function of ρ.

To obtain the confidence intervals for the sensitivity analysis, we apply the following iterative algorithm to equations (8.3) and (8.4) for a fixed value of ρ. At the tth iteration, given the current values of the coefficients, i.e., $\theta^{(t)} = (\alpha_2^{(t)}, \beta_2^{(t)}, \xi_2^{(t)}, \dots)$, and a given error correlation ρ, we compute the variance–covariance matrix of $(\varepsilon_{i2}, \varepsilon_{i3})$, which is denoted by $\Sigma^{(t)}$. The matrix is computed by setting ${\sigma_j^{(t)}}^2 = ||\hat{\varepsilon}_j^{(t)}||^2/(n - L_j)$ and $\sigma_{23}^{(t)} = \rho \sigma_2^{(t)} \sigma_3^{(t)}$, where $\hat{\varepsilon}_j^{(t)}$

is the residual vector and L_j is the number of coefficients for the mediator model ($j = 2$) and the outcome model ($j = 3$) at the tth iteration. We then update the parameters via generalized least squares, i.e.,

$$\theta^{(t+1)} = \{V^\top (\Sigma^{(t)^{-1}} \otimes I_n) V\}^{-1} V^\top (\Sigma^{(t)^{-1}} \otimes I_n) W$$

where $V = \begin{bmatrix} \mathbf{1} & T & X & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & T & M & TM & X \end{bmatrix}$, $W = \begin{bmatrix} M \\ Y \end{bmatrix}$, $T = (T_1, \ldots, T_n)^\top$, $M = (M_1, \ldots, M_n)^\top$ and $Y = (Y_1, \ldots, Y_n)^\top$ are column vectors of length n, and $X = (X_1, \ldots, X_n)^\top$ are the $(n \times K)$ matrix of observed pretreatment covariates, and $\otimes$ represents the Kronecker product. We typically use equation-by-equation least squares estimates as the starting values of θ and iterate these two steps until convergence. This is essentially an application of the iterative feasible generalized least square algorithm of the seemingly unrelated regression [12], and thus the asymptotic variance of $\hat{\theta}$ is given by $\mathrm{Var}(\hat{\theta}) = \{V^\top (\Sigma^{-1} \otimes I_n) V\}^{-1}$. Then, for a given value of ρ, the asymptotic variance of the estimated average causal mediation effects is found, for example, by the Delta method and the confidence intervals can be constructed.

The Binary Outcome Case

The sensitivity analysis for binary outcomes parallels the case when both the mediator and outcome are continuous. Here, we assume that the model for the outcome is a probit regression. Using a probit regression for the outcome allows us to assume the error terms are jointly normal with a possibly nonzero correlation ρ. Imai et al. [2] derive the average causal mediation effects as a function of ρ and a set of parameters that are identifiable due to randomization of the treatment. This lets us use ρ as a sensitivity parameter in the same way as in the Baron–Kenny procedure. For the calculation of confidence intervals, we rely on the quasi-Bayesian approach of Algorithm 1 by approximating the posterior distribution with the sampling distribution of the maximum likelihood estimates.

The Binary Mediator Case

Finally, a similar sensitivity analysis can also be conducted in a situation where the mediator variable is dichotomous and the outcome is continuous. In this case, we assume that the mediator can be modeled as a probit regression where the error term is independently and identically distributed as standard normal distribution. A linear normal regression with error variance equal to σ_3^2 is used to model the continuous outcome variable. We further assume that the two error terms jointly follow a bivariate normal distribution with mean zero and covariance $\rho\sigma_3$. Then, as in the other two cases, we use the correlation between the two error terms ρ as the sensitivity parameter. Imai et al. [2] show that under this setup, the causal mediation effects can

be expressed as a function of the model parameters that can be consistently estimated given a fixed value of ρ. Uncertainty estimates are computed based on the quasi-Bayesian approach, as in the binary outcome case. The results can be graphically summarized via the `plot()` function in a manner similar to the other two cases.

Alternative Interpretations Based on R^2

The main advantage of using ρ as a sensitivity parameter is its simplicity. However, applied researchers may find it difficult to interpret the magnitude of this correlation coefficient. To overcome this limitation, Imai et al. [3] proposed alternative interpretations of ρ based on the coefficients of determination or R^2 and Imai et al. [2] extended them to the binary mediator and binary outcome cases. In that formulation, it is assumed that there exists a common unobserved pretreatment confounder in both mediator and outcome models. Applied researchers are then required to specify whether the coefficients of this unobserved confounder in the two models have the same sign or not; i.e., $\text{sgn}(\lambda_2\lambda_3) = 1$ or -1 where λ_2 and λ_3 are the coefficients in the mediator and outcome models, respectively. Once this information is provided, the average causal mediation effect can be expressed as the function of "the proportions of original variances explained by the unobserved confounder" where the original variances refer to the variances of the mediator and the outcome (or the variance of latent variable in the case of binary dependent variable). Alternatively, the average causal mediation effect can also be expressed in terms of "the proportions of the previously unexplained variances explained by the unobserved confounder" (see [1] for details). These alternative interpretations allow researchers to quantify how large the unobserved confounder must be (relative to the observed pretreatment covariates in the model) in order for the original conclusions to be reversed.

8.2.4 Current Limitations

Our software, **mediation**, is quite flexible and can handle many of the model types that researchers are likely to use in practice. Table 8.1 categorizes the types of the mediator and outcome variables and lists whether **mediation** can produce the point and uncertainty estimates of causal mediation effects. For example, while **mediation** can estimate average causal mediation effects when the mediator is ordered and the outcome is continuous, it has not yet been extended to other cases involving ordered variables. In each situation handled by **mediation**, it is possible to have an interaction term between treatment status and the mediator variable, in which case the estimated quantities of interest will be reported separately for the treatment and control groups.

Table 8.1 The types of data that can be currently handled by **mediation** for the estimation of causal mediation effects

	Outcome Variable Types		
Mediator Types	Continuous	Ordered	Binary
Continuous	Yes	No	Yes
Ordered	Yes	No	No
Binary	Yes	No	Yes

Table 8.2 The types of data that can be currently handled by **mediation** for sensitivity analysis. For continuous variables, the linear regression model is assumed. For binary variables, the probit regression model is assumed

	Outcome Variable Types		
Mediator Types	Continuous	Ordered	Binary
Continuous	Yes	No	Yes
Ordered	No	No	No
Binary	Yes	No	No

Our software provides a convenient way to probe the sensitivity of results to potential violations of the ignorability assumption for certain model types. The sensitivity analysis requires the specific derivations for each combination of models, making it difficult to develop a general sensitivity analysis method. As summarized in Table 8.2, **mediation** can handle several cases that are likely to be encountered by applied researchers. When the mediator is continuous, then sensitivity analysis can be conducted with continuous and binary outcome variables. In addition, when the mediator is binary, sensitivity analysis is available for continuous outcome variables. For sensitivity analyses that combine binary or continuous mediators and outcomes, analysts must use a probit regression model with a linear regression model. This allows for jointly normal errors in the analysis. Unlike the estimation of causal mediation effects, sensitivity analysis with treatment/mediator interactions can only be done for the continuous outcome/continuous mediator and continuous outcome/binary mediator cases. In the future, we hope to expand the range of models that are available for sensitivity analysis.

8.3 Examples

Next, we provide several examples to illustrate the use of **mediation** for the estimation of causal mediation effects and sensitivity analysis. The data used are available as part of the package so that readers can replicate the results reported below. We demonstrate the variety of models that can be used for the outcome and mediating variables.

Before presenting our examples, we load the **mediation** library and the example data set included with the library.

```
R> library("mediation")
mediation: R Package for Causal Mediation Analysis
Version: 2.0
R> data("jobs")
```

This dataset is from the Job Search Intervention Study (JOBS II) [10]. In the JOBS II field experiment, 1,801 unemployed workers received a pre-screening questionnaire and were then randomly assigned to treatment and control groups. Those in the treatment group participated in job-skills workshops. Those in the control condition received a booklet describing job-search tips. In follow-up interviews, two key outcome variables were measured: a continuous measure of depressive symptoms based on the Hopkins Symptom Checklist (`depress2`), and a binary variable representing whether the respondent had become employed (`work1`). In the JOBS II data, a continuous measure of job-search self-efficacy represents a key mediating variable (`job_seek`). In addition to the outcome and mediators, the JOBS II data also include the following list of baseline covariates that were measured prior to the administration of the treatment: pretreatment level of depression (`depress1`), education (`educ`), income, race (`nonwhite`), marital status (`marital`), age, sex, previous occupation (`occp`), and the level of economic hardship (`econ_hard`).

8.3.1 Estimation of Causal Mediation Effects

The Baron–Kenny Procedure

We start with an example when both the mediator and the outcome are continuous. In this instance, the results from either algorithm will return point estimates essentially identical to the usual Baron and Kenny procedure though the quasi-Bayesian or nonparametric bootstrap approximation is used. Using the JOBS II data, we first estimate two linear regressions for both the mediator and the outcome using the `lm()` function.

```
R> model.m <- lm(job_seek ~ treat + depress1 + econ_hard
+ sex + age + occp + marital + nonwhite + educ + income,
```

```
data = jobs)
R> model.y <- lm(depress2 ~ treat + job_seek + depress1
+ econ_hard + sex + age + occp + marital + nonwhite
+ educ + income, data = jobs)
```

These two model objects, `model.m` and `model.y`, become the arguments for the `mediate()` function. The analyst must take some care with missing values before estimating the models above. While model functions in **R** handle missing values in the data using the usual listwise deletion procedures, the functions in **mediation** assume that missing values have been removed from the data before the estimation of these two models. Thus the data for the two models must have identical observations sorted in the same order with all missing values removed. The **R** function `na.omit()` can be used to remove missing values from the data frame.

In the first call to `mediate()` below, we specify `boot = TRUE` to call the nonparametric bootstrap with 1000 resamples (`sims = 1000`). When this option is set to `FALSE` in the second call, inference proceeds via the quasi-Bayesian Monte Carlo approximation using Algorithm 1 rather than Algorithm 2. We must also specify the variable names for the treatment indicator and the mediator variable using `treat` and `mediator`, respectively.

```
R> out.1 <- mediate(model.m, model.y, sims = 1000,
boot = TRUE, treat = "treat", mediator = "job_seek")
R> out.2 <- mediate(model.m, model.y, sims = 1000,
treat = "treat", mediator = "job_seek")
```

The objects from a call to `mediate()`, i.e., `out.1` and `out.2` above, are lists which contain several different quantities from the analysis. For example, `out.1$d0` returns the point estimate for the average causal mediation effect based on Algorithm 1. The help file contains a full list of values that are contained in `mediate()` objects. The `summary()` function prints out the results of the analysis in tabular form:

```
R> summary(out.1)

 Causal Mediation Analysis

Confidence Intervals Based on Nonparametric Bootstrap

Mediation Effect:  -0.01593 95% CI  -0.031140 -0.002341
Direct Effect:  -0.03125 95% CI  -0.1045  0.0408
Total Effect:  -0.04718 95% CI  -0.11996  0.02453
Proportion of Total Effect via Mediation:
0.2882 95% CI  -2.412  3.419

R> summary(out.2)
.
```

```
.
Output Omitted
```

The output from the `summary()` function displays the estimates for the average causal mediation effect, direct effect, total effect, and proportion of total effect mediated. The first column displays the quantity of interest, the second column displays the point estimate, and the other columns present the 95% confidence intervals. Researchers can then easily report these point estimates and corresponding uncertainty estimates in their work. In this case, we find that job search self-efficacy mediated the effect of the treatment on depression in the negative direction. This effect, however, was small with a point estimate of $-.016$ but the 95% confidence intervals $(-.031, -.002)$ still do not contain 0.

The Baron–Kenny Procedure with the Interaction Term

Analysts can also allow the causal mediation effect to vary with treatment status. Here, the model for the outcome must be altered by including an interaction term between the treatment indicator, `treat`, and the mediator variable, `job_seek`:

```
R> model.y <- lm(depress2 ~ treat + job_seek
+ treat:job_seek + depress1 + econ_hard + sex
+ age + occp + marital + nonwhite + educ
+ income, data = jobs)
```

Users should note that under our current implementation, the interaction term must be specified in the form of `treat.name:med.name` where `treat.name` and `med.name` are the names of the treatment variable and mediator in the model, respectively. Then, a call is again made to `mediate()`, but now the option `INT = TRUE` must be specified:

```
R> out.3 <- mediate(model.m, model.y, sims = 1000,
boot = TRUE, INT = TRUE, treat = "treat", mediator =
"job_seek")
R> out.4 <- mediate(model.m, model.y, sims=1000,
INT = TRUE, treat = "treat", mediator =
"job_seek")
R> summary(out.3)

 Causal Mediation Analysis

Confidence Intervals Based on Nonparametric Bootstrap

Mediation Effect_0:  -0.02056 95% CI  -0.0425 -0.0038
Mediation Effect_1:  -0.01350 95% CI  -0.0281 -0.0023
Direct Effect_0:  -0.03318 95% CI  -0.10496  0.03592
```

```
Direct Effect_1:  -0.02611 95% CI  -0.09612  0.04454
Total Effect:  -0.04668 95% CI  -0.11594  0.02135
Proportion of Total Effect via Mediation:
0.3053 95% CI  -3.578  3.593

R> summary(out.4)
.
.
Output Omitted
```

Again using the `summary()` function provides a table of the results. Now estimates for the mediation and direct effects correspond to the levels of the treatment and are printed as such in the tabular summary. In this case, the mediation effect under the treatment condition, listed as `Mediation Effect_1`, is estimated to be $-.014$ while the mediation effect under the control condition, `Mediation Effect_0`, is $-.021$.

Use of Non/Semiparametric Regression

The flexibility of **mediation** becomes readily apparent when we move beyond standard linear regression models. For example, we might suspect that the mediator has a nonlinear effect on the outcome. Generalized Additive Models (GAMs) allow analysts to use splines for flexible nonlinear fits. This presents no difficulties for the `mediate()` function. We model the mediator as before, but we alter the outcome model using the `gam()` function from the `mgcv` library.

```
R> library(mgcv)
This is mgcv 1.4-1
R> model.m <- lm(job_seek ~ treat + depress1
+ econ_hard + sex + age + occp + marital
+ nonwhite + educ + income, data = jobs)
R> model.y <- gam(depress2 ~ treat + s(job_seek,
bs = "cr")  + depress1 + econ_hard + sex + age
+ occp + marital + nonwhite + educ + income,
data = jobs)
```

In this case we fit a Generalized Additive Model for the outcome variable, and allow the effect of the `job_seek` variable to be nonlinear and determined by the data. This is done by using the `s()` notation which allows the fit between the mediator and the outcome to be modeled with a spline. Using the spline for the fit allows the estimate for the mediator on the outcome to be a series of piecewise polynomial regression fits. This semiparametric regression model is a more general version of nonparametric regression models such as lowess. The model above allows the estimate to vary across the range of the predictor variable. Here, we specify the model with a cubic basis function (`bs = "cr"`) for the smoothing spline and leave the smoothing selection to be done at the

program defaults which is generalized cross-validation. Fully understanding how to fit such models is beyond the scope here. Interested readers should consult Wood 2006 [11] for full technical details and Keele 2008 [5] provides coverage of these models from a social science perspective.

The call to `mediate()` with a `gam()` fit remains unchanged except that when the outcome model is a semiparametric regression only the nonparametric bootstrap is valid for calculating uncertainty estimates, i.e., `boot = TRUE`.

```
R> out.5 <- mediate(model.m, model.y, sims = 1000,
boot = TRUE, treat = "treat", mediator = "job_seek")

R> summary(out.5)
.
.
Output Omitted
```

The model for the mediator can also be modeled with the `gam()` function as well. The `gam()` function also allows analysts to include interactions; thus analysts can still allow the mediation effects to vary with treatment status. This simply requires altering the model specification by using the `by` option in the `gam()` function and using two separate indicator variables for treatment status. To fit this model we need one variable that indicates whether the observation was in the treatment group and a second variable that indicates whether the observation was in the control group. To allow the mediation effect to vary with treatment status, the call to `gam()` takes the following form:

```
R> model.y <- gam(depress2 ~ treat + s(job_seek, by = treat)
+ s(job_seek, by = control) + depress1 + econ_hard + sex
+ age + occp + marital + nonwhite + educ + income,
data = jobs)
```

In this case, we must also alter the options in the `mediate()` function by specifying `INT = TRUE` and provide the variable name for the control group indicator using the `control` option.

```
R> out.6 <- mediate(model.m, model.y, sims = 1000,
boot = TRUE, INT = TRUE, treat = "treat",
mediator = "job_seek", control = "control")

R> summary(out.6)

 Causal Mediation Analysis

Confidence Intervals Based on Nonparametric Bootstrap

Mediation Effect_0:  -0.02328 95% CI  -0.059138  0.006138
```

```
Mediation Effect_1:  -0.01622 95% CI  -0.041565  0.004363
Direct Effect_0:  -0.01408 95% CI  -0.09369  0.05672
Direct Effect_1:  -0.007025 95% CI  -0.08481  0.06114
Total Effect:  -0.0303 95% CI  -0.13065  0.04744
Proportion of Total Effect via Mediation:
0.3395% CI  -8.514  4.391
```

As the reader can see, despite the fact that the mediator was specified as a nonparametric function, one still receives point estimates and confidence intervals for the mediation effect across each treatment level. In the table, `Mediation Effect_0` and `Direct Effect_0` are the mediation and direct effects respectively under the control condition, while `Mediation Effect_1` and `Direct Effect_1` are the mediation and direct effects under treatment.

Quantile Causal Mediation Effects

Researchers might also be interested in modeling mediation effects for quantiles of the outcome. Quantile regression allows for a convenient way to model the quantiles of the outcome distribution while adjusting for a variety of covariates [7]. For example, a researcher might be interested in the 0.5 quantile (i.e., median) of the distribution. This also presents no difficulties for the `mediate()` function. Again for these models, uncertainty estimates are calculated using the nonparametric bootstrap. To use quantile regression, we load the `quantreg` library and model the median of the outcome, though other quantiles are also permissible. Analysts can also relax the no-interaction assumption for the quantile regression models as well. Below we estimate the mediator with a standard linear regression, while for the outcome we use `rq()` to model the median.

```
R> library(quantreg)
Loading required package: SparseM
Package SparseM (0.78) loaded.
To cite, see citation("SparseM")
Package quantreg (4.26) loaded.
To cite, see citation("quantreg")

R> model.m <- lm(job_seek ~ treat + depress1 + econ_hard
+ sex + age + occp + marital + nonwhite + educ + income,
data = jobs)
R> model.y <- rq(depress2 ~ treat + job_seek + depress1
+ econ_hard + sex + age + occp + marital + nonwhite
+ educ + income, tau= 0.5, data = jobs)
R> out.7 <- mediate(model.m, model.y, sims = 1000,
boot = TRUE, treat = "treat", M = "job_seek")

R> summary(out.7)
```

```
 Causal Mediation Analysis

Confidence Intervals Based on Nonparametric Bootstrap

Mediation Effect:  -0.01470 95% CI  -0.027235 -0.001534
Direct Effect:  -0.02489 95% CI  -0.09637  0.04309
Total Effect:  -0.03959 95% CI  -0.11523  0.02857
Proportion of Total Effect via Mediation:
0.3337 95% CI  -3.069  1.902
```

where the `summary()` command gives the estimated median causal mediation effect along with the estimates for the other quantities of interest.

It is also possible to estimate mediation effects for quantiles of the outcome other than the median. This is done simply by specifying a different outcome quantile in the quantile regression model. For example, if the 10th percentile of the outcome were of interest, then the user can change the `tau` option,

```
R> model.y <- rq(depress2 ~ treat + job_seek + depress1
+ econ_hard + sex + age + occp + marital + nonwhite
+ educ + income, tau = 0.1, data = jobs)
```

Furthermore, it is straightforward to loop over a set of quantiles and graph the mediation effects for a range of quantiles, as done in [2].

Discrete Mediator and Outcome Data

Often analysts use measures for the mediator and outcome that are discrete. For standard methods, this has presented a number of complications requiring individually tailored techniques. The **mediation** software, however, can handle a number of different discrete data types using the general algorithms developed in Imai et al. [2]. For example, one outcome of interest in the JOBS II study is a binary indicator (`work1`) for whether the subject became employed after the training sessions. To estimate the mediation effect, we simply use a probit regression instead of a linear regression for the outcome and then call `mediate()` as before:

```
R> model.m <- lm(job_seek ~ treat + depress1 + econ_hard
+ sex + age + occp + marital + nonwhite + educ + income,
data = jobs)
R> model.y <- glm(work1 ~ treat + job_seek + depress1
+ econ_hard + sex + age + occp + marital + nonwhite + educ
+ income, family = binomial(link = "probit"), data = jobs)
R> out.8 <- mediate(model.m, model.y, sims = 1000,
boot = TRUE, treat = "treat", mediator = "job_seek")
R> out.9 <- mediate(model.m, model.y, sims = 1000,
treat = "treat", mediator = "job_seek")
```

```
R> summary(out.8)
.
.
Output Omitted

R> summary(out.9)

 Causal Mediation Analysis

Quasi-Bayesian Confidence Intervals

Mediation Effect:  0.003780 95% CI  -0.0005248  0.0109583
Direct Effect:  0.05573 95% CI  -0.007416  0.119900
Total Effect:  0.05951 95% CI  -0.004037  0.123071
Proportion of Total Effect via Mediation:
0.05804 95% CI  -0.2405  0.4498
```

In the table printed by the `summary()` function, the estimated average causal mediation effect along with the quasi-Bayesian confidence interval are printed on the first line followed by the direct and total effects, and the proportion of the total effect due to mediation. It is also possible to use a logit model for the outcome instead of a probit model. However, we recommend the use of a probit model because our implementation of the sensitivity analysis below requires a probit model for analytical tractability.

The mediator can also be binary or an ordered measure as well. This simply requires modeling the mediator with either a probit or ordered probit model. For demonstration purposes, the `jobs` data contains two variables, `job_dich` and `job_disc`, which are recoded versions of `job_seek`. The first measure is simply the continuous scale divided at the median into a binary variable. The second measure, `job_disc`, recodes the continuous scale into a discrete four-point scale. We emphasize that this is for demonstration purposes only, and analysts in general should not recode continuous measures into discrete measures. Estimating mediation effects with a binary mediator is quite similar to the case above with a binary outcome. We simply now use a probit model for the mediator and a linear regression for the outcome:

```
R> model.m <- glm(job_dich ~ treat + depress1 + econ_hard
+ sex + age + occp + marital + nonwhite + educ + income,
data = job, family = binomial(link = "probit"))
R> model.y <- lm(depress2 ~ treat + job_dich + treat:job_dich
+ depress1 + econ_hard + sex + age + occp + marital
+ nonwhite + educ + income, data = jobs)
```

In this example we allow the effect of the mediator to vary with treatment status. The user now calls `mediate()` and can use either the quasi-Bayesian approximation or nonparametric bootstrap.

```
R> out.10 <- mediate(model.m, model.y, sims = 1000,
boot=TRUE, treat="treat", mediator="job_dich", INT=TRUE)
R> out.11 <- mediate(model.m, model.y, sims = 1000,
treat = "treat", mediator = "job_dich", INT = TRUE)
R> summary(out.10)
.
.
Output Omitted
R> summary(out.11)
 Causal Mediation Analysis

Quasi-Bayesian Confidence Intervals

Mediation Effect_0:  -0.01809 95% CI  -0.035290 -0.005589
Mediation Effect_1:  -0.01968 95% CI  -0.034518 -0.007263
Direct Effect_0:  -0.02849 95% CI  -0.1008  0.0393
Direct Effect_1:  -0.03009 95% CI  -0.10111  0.03791
Total Effect:  -0.04817 95% CI  -0.11962  0.01729
Proportion of Total Effect via Mediation:
0.3431 95% CI  -3.330  3.756
```

In the table, we see that `Mediation Effect_0` is the mediation effect under the control condition, while `Mediation Effect_1` is the mediation effect under the treatment condition. The same notation applies to the direct effects. As the reader can see, the output also indicates which algorithm is used for the 95% confidence intervals.

When the mediator is an ordered variable, we switch to an ordered probit model for the mediator. In **R**, the `polr()` function in the `MASS` library provides this functionality. The `MASS` library is automatically loaded with **mediation** so the `polr()` function is readily available to users. Thus, we fit the outcome and mediator models below:

```
R> model.m <- polr(job_disc ~ treat + depress1 + econ_hard
+ sex + age + occp + marital + nonwhite + educ + income,
data = jobs, method = "probit", Hess = TRUE)
R> model.y <- lm(depress2 ~ treat + job_disc + depress1
+ econ_hard + sex + age + occp + marital + nonwhite
+ educ + income, data = jobs)
```

The reader should note that in the call to `polr()` the `Hess = TRUE` needs to be specified to use the quasi-Bayesian approximation in the `mediate()` function. Once we have estimated these two models, analysis proceeds as before:

```
R> out.12 <- mediate(model.m, model.y, sims = 1000,
boot = TRUE, treat = "treat", mediator = "job_disc")
R> out.13 <- mediate(model.m, model.y, sims = 1000,
```

```
treat = "treat", mediator = "job_disc")

R> summary(out.12)
.
.
Output Omitted
R> summary(out.13)
.
.
Output Omitted
```

Again, for any of these data types, analysts can relax the no-interaction assumption as before by including the interaction between treatment and the mediator variable in the outcome model and using the `INT = TRUE` option.

8.3.2 Sensitivity Analysis

Once analysts have estimated mediation effects, they should always explore how robust their finding is to the ignorability assumption. The `medsens()` function allows analysts to conduct sensitivity analyses for mediation effects. Next, we provide a demonstration of the functionality for the sensitivity analysis. Currently, **mediation** can conduct sensitivity analyses for the continuous–continuous case, the binary–continuous case, and the continuous–binary case.

The Baron–Kenny Procedure

As before, one must first fit models for the mediator and outcome and then pass these model objects through the mediate function:

```
R> model.m <- lm(job_seek ~ treat + depress1 + econ_hard
+ sex + age + occp + marital + nonwhite + educ + income,
data = jobs)
R> model.y <- lm(depress2 ~ treat + job_seek + depress1
+ econ_hard + sex + age + occp
+ marital + nonwhite + educ + income, data = jobs)
R> med.cont <- mediate(model.m, model.y, sims=1000,
treat = "treat", mediator = "job_seek")
```

Once the analyst estimates the mediation effects, the output from the `mediate()` function becomes the argument for `medsens()`, which is the workhorse function. The `medsens()` function recognizes the options specified in the `mediate()` function and thus there is no need to specify the `treat`, `mediator`, or `INT` options.

```
R> sens.cont <- medsens(med.cont, rho.by = 0.05)
```

The `rho.by` option specifies how finely incremented the parameter ρ is for the sensitivity analysis. Using a coarser grid for ρ speeds up estimation considerably, but this comes at the cost of estimating the robustness of the original conclusion only imprecisely.

After running the sensitivity analysis via `medsens()`, the `summary()` function can be used to produce a table with the values of ρ for which the confidence interval contains zero. This allows the analyst to immediately see the approximate range of ρ where the sign of the causal mediation effect is indeterminate. The second section of the table contains the value of ρ for which the mediation effect is exactly zero, which in this application is -0.19. The table also presents coefficients of determination that correspond to the critical value of ρ where the mediation effect is zero. First, $R^{*2}_M R^{*2}_Y$ is the product of coefficients of determination which represents the proportion of the *previously unexplained* variance in the mediator and outcome variables that is explained by an unobservable pretreatment unconfounder. An alternative formulation is in terms of the proportion of the *original* variance explained by an unobserved confounder, which we denote as $\widetilde{R}^2_M \widetilde{R}^2_Y$.

```
R> summary(sens.cont)

Mediation Sensitivity Analysis

Sensitivity Region
    Rho Med. Eff. 95% CI    95% CI  R^2_M*R^2_Y* R^2_M~R^2_Y
                        Lower     Upper
[1,] -0.25   0.0056  -0.0008   0.0120   0.0625       0.0403
[2,] -0.20   0.0012  -0.0035   0.0058   0.0400       0.0258
[3,] -0.15  -0.0032  -0.0084   0.0020   0.0225       0.0145
[4,] -0.10  -0.0074  -0.0150   0.0001   0.0100       0.0064

Rho at which ACME = 0: -0.1867

R^2_M*R^2_Y* at which ACME = 0: 0.0349

R^2_M~R^2_Y~ at which ACME = 0: 0.0225
```

The table above presents the estimated mediation effect along with its confidence interval for each value of ρ. The reader can verify that when ρ is equal to zero, the reported mediation effect matches the estimate produced by the `mediate()` function. For other values of ρ, the mediation effect is calculated under different levels of unobserved confounding.

The information from the sensitivity analysis can also be summarized graphically using the `plot()` function. First, passing the `medsens` object to `plot()` and specifying the `sens.par` option to `"rho"`, i.e.,

```
R> plot(sens.cont, sens.par = "rho")
```

produces the left-hand side of Figure 8.2. In the plot, the dashed horizontal line represents the estimated mediation effect under the sequential ignorability assumption, and the solid line represents the mediation effect under various values of ρ. The gray region represents the 95% confidence bands.

Similarly, we can also plot the sensitivity analysis in terms of the coefficients of determination as discussed above. Here we specify `sens.par` option to `"R2"`. We also need to specify two additional pieces of information. First, `r.type` option tells the plot function whether to plot $R^{*2}_M R^{*2}_Y$ or $\widetilde{R}^2_M \widetilde{R}^2_Y$. To plot the former `r.type` is set to `1` and to plot the latter `r.type` is set to `2`. Finally, the `sign.prod` option specifies the sign of the product of the coefficients of the unobserved confounder in the mediator and outcome models. This product indicates whether the unobserved confounder affects both mediator and outcome variables in the same direction (`1`) or different directions (`-1`), thereby reflecting the analyst's expectation about the nature of confounding.

For example, the following command produces the plot representing the sensitivity of estimates with respect to the proportion of the original variances explained by the unobserved confounder when the confounder is hypothesized to affect the mediator and outcome variables in opposite directions.

```
R> plot(sens.cont, sens.par = "R2", r.type = 2,
sign.prod = -1)
```

The resulting plot is shown on the right-hand side of Figure 8.2. Each contour line represents the mediation effect for the corresponding values of $\widetilde{R}^2_M$ and $\widetilde{R}^2_Y$. For example, the 0 contour line corresponds to values of the product $\widetilde{R}^2_M \widetilde{R}^2_Y$ such that the average causal mediation effect is 0. As reported in the table, even a small proportion of original variance unexplained by the confounder, .02%, produces mediation effects of 0. Accordingly, the right-hand side of Figure 8.2 shows how increases in $\widetilde{R}^2_M \widetilde{R}^2_Y$ (moving from the lower left to upper right) produce *positive* mediation effects.

For both types of sensitivity plots, the user can specify additional options available in the plot function such as alternative title (`main`) and axis labels (`xlab`, `ylab`) or manipulate common graphical options (e.g., `xlim`).

Binary Outcome

The `medsens()` function also extends to analyses where the mediator is binary and the outcome is continuous, as well as when the mediator is continuous and the outcome is binary. If either variable is binary, `medsens()` takes an additional argument. For example, recall the binary outcome model estimated earlier:

```
R> model.y <- glm(work1 ~ treat + job_seek + depress1
+ econ_hard + sex + age + occp + marital + nonwhite
```

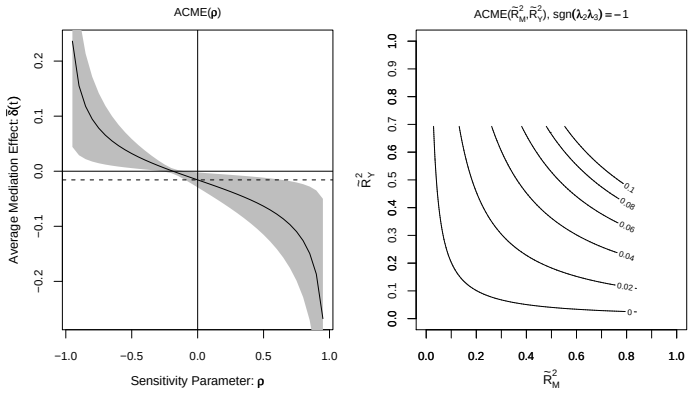

Fig. 8.2 Sensitivity analysis with continuous outcome and mediator.

```
+ educ  + income, family = binomial(link = "probit"),
data = jobs)
R> med.bout <- mediate(model.m, model.y, sims = 1000,
treat = "treat", mediator = "job_seek")
```

The call to `medsens()` works as before, with the output from the `mediate()` function passed through `medsens()`.

```
R> sens.bout <- medsens(med.bout, rho.by = 0.05,
sims = 1000)
```

The `sims` option provides control over the number of draws in the parametric bootstrap procedure which is used to compute confidence bands. When either the mediator or outcome is binary, the exact values of sensitivity parameters where the mediation effects are zero cannot be analytically obtained as in the fully continuous case (see [3] Section 4). Thus, this information is reported based on the signs of the estimated mediation effects under various values of ρ and corresponding coefficients of determination. The usage of the `summary()` function, however, remains identical to the fully continuous case in that the output table contains the estimated mediation effects and the corresponding values of ρ for which the confidence region contains zero.

As in the case with continuous mediator and outcome variables, we can plot the results of the sensitivity analysis. The following code produces Figure 8.3.

```
R> plot(sens.bout, sens.par = "rho")
R> plot(sens.bout, sens.par = "R2", r.type = 2,
sign.prod = 1)
```

On the left-hand side we plot the average causal mediation effects in terms of ρ, while we use $\widetilde{R}^2_M$ and $\widetilde{R}^2_Y$ on the right-hand side. In the ρ plot, the dashed line represents the estimated mediation effect under sequential ignorability,

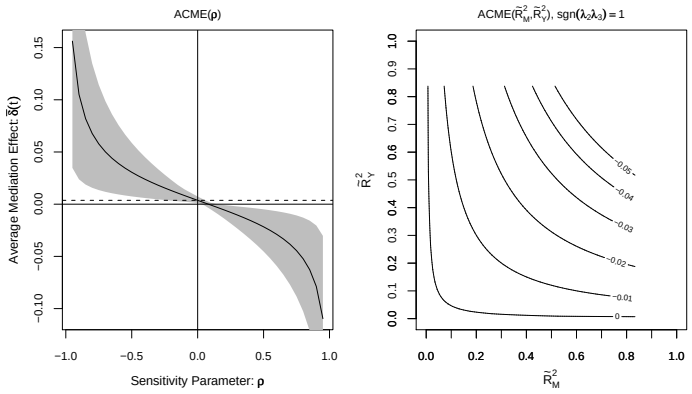

Fig. 8.3 Sensitivity analysis with continuous outcome and binary mediator.

and the solid line represents the mediation effect under various values of ρ. The gray region represents the 95% confidence bands. In the $\widetilde{R}^2$ plot the average causal mediation effect is plotted against various values of $\widetilde{R}^2_M$ and $\widetilde{R}^2_Y$ and is interpreted in the same way as above.

When the outcome is binary, the proportion of the total effect due to mediation can also be calculated as a function of the sensitivity parameter ρ. The `pr.plot` option in the plot command (in conjunction with the `sens.par = "rho"` option) allows users to plot a summary of the sensitivity analysis for the proportion mediated. For example, the following call would provide a plot of this quantity:

```
R> plot(sens.bout, sens.par = "rho", pr.plot = TRUE)
```

Binary Mediator

The final form of sensitivity analysis deals with the case where the outcome variable is continuous but the mediator is binary. For the purpose of illustration, we simply dichotomize the `job_seek` variable to produce a binary measure `job_dich`. We fit a probit model for the mediator and linear regression for the outcome variable.

```
R> model.m <- glm(job_dich ~ treat + depress1
+ econ_hard + sex + age + occp + marital + nonwhite
+ educ + income, data = jobs,
family = binomial(link = "probit"))
R> model.y <- lm(depress2 ~ treat + job_dich+ depress1
+ econ_hard + sex + age + occp
+ marital + nonwhite + educ + income, data = jobs)
R> med.bmed <- mediate(model.m, model.y, sims = 1000,
treat = "treat", mediator = "job_dich")
```

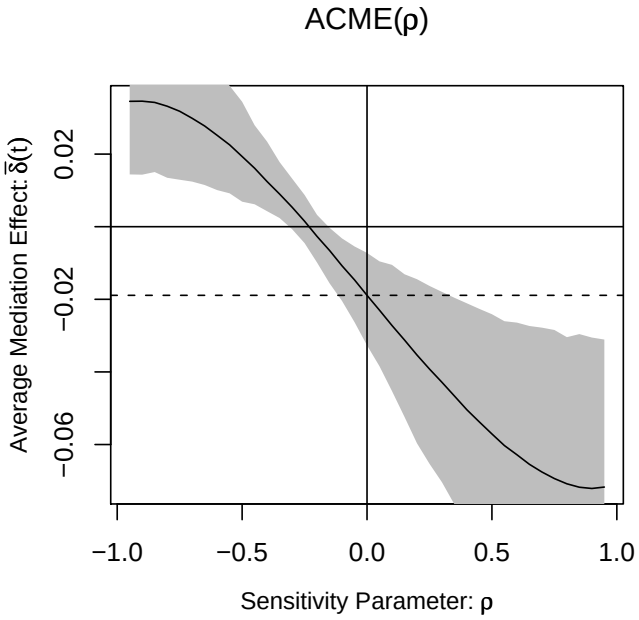

Fig. 8.4 Sensitivity analysis with continuous outcome and binary mediator.

```
R> sens.bmed <- medsens(med.bmed, rho.by = 0.05,
sims = 1000)
```

Again we can pass the output of the `medsens()` function through the `plot()` function:

```
R> plot(sens.bmed, sens.par = "rho")
```

producing Figure 8.4. The plot is interpreted in the same way as the above cases. The user also has the option to plot sensitivity results in terms of the coefficients of determination just as in the case with continuous outcome and mediator variables.

When the mediator variable is binary, the plotted values of the mediation effect and their confidence bands may not be perfectly smooth curves due to simulation errors. This is especially likely when the number of simulations (`sims`) is set to a small value. In such situations, the user can choose to set the `smooth.effect` and `smooth.ci` options to `TRUE` in the `plot()` function so that the corresponding values become smoothed out via a lowess smoother before being plotted. Although this option often makes the produced graph look nicer, the user should be cautious as the adjustment could affect one's

substantive conclusions in a significant way. A recommended alternative is to increase the number of simulations.

8.4 Concluding Remarks

Causal mediation analysis is a key tool for social scientific research. In this paper, we describe our easy-to-use software for causal mediation analysis, **mediation**, that implements the new methods and algorithms introduced by Imai et al. 2008 [3] and Imai et al. 2009 [2]. The software provides a flexible, unified approach to causal mediation analysis in various situations encountered by applied researchers. The object-oriented nature of the **R** programming made it possible for us to implement these algorithms in a fairly general way. In addition to the estimation of causal mediation effects, **mediation** implements formal sensitivity analyses so that researchers can assess the robustness of their findings to the potential violations of the key identifying assumption. This is an important contribution for at least two reasons. First, even in experiments with randomize treatments, causal mediation analysis requires an additional assumption that is not directly testable from the observed data. Thus, researchers must evaluate the consequences of potential violations of the assumption via sensitivity analysis. Alternatively, researchers might use other experimental designs though this entails making other assumptions [4]. Second, the accumulation of such sensitivity analyses is essential for interpreting the relative degree of robustness across different studies. Thus, the development of easy-to-use software, such as **mediation**, facilitates causal mediation analysis in applied social science research in several critical directions.

8.5 Notes and Acknowledgment

The most recent version (along with all previous versions) of the **R** package, **mediation**, is available for download at the Comprehensive R Archive Network (`http://cran.r-project.org/web/packages/mediation`). This article is based on version 2.1 of **mediation**. Financial support from the National Science Foundation (SES-0849715 and SES-0918968) is acknowledged.

References

1. Baron, R., Kenny, D.: The moderator-mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations. Journal

of Personality and Social Psychology **51**(4), 1173–1182 (1986)
2. Imai, K., Keele, L., Tingley, D.: A general approach to causal mediation analysis. Tech. rep., Department of Politics, Princeton University (2009). Available at `http://imai.princeton.edu/research/BaronKenny.html`
3. Imai, K., Keele, L., Yamamoto, T.: Identification, inference and sensitivity analysis for causal mediation effects. Tech. rep., Department of Politics, Princeton University (2008). Available at `http://imai.princeton.edu/research/mediation.html`
4. Imai, K., Tingley, D., Yamamoto, T.: Experimental identification of causal mechanisms. Tech. rep., Department of Politics, Princeton University (2009). Available at `http://imai.princeton.edu/research/Design.html`
5. Keele, L.: Semiparametric Regression for the Social Sciences. Wiley and Sons, Chichester, UK (2008)
6. King, G., Tomz, M., Wittenberg, J.: Making the most of statistical analyses: Improving interpretation and presentation. American Journal of Political Science **44**, 341–355 (2000)
7. Koenker, R.: Quantile Regression. Cambridge University Press, Cambridge (2008)
8. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2009). URL `http://www.R-project.org`. ISBN 3-900051-07-0
9. Rosenbaum, P.R.: Observational Studies, 2nd edn. Springer-Verlag, New York (2002)
10. Vinokur, A., Schul, Y.: Mastery and inoculation against setbacks as active ingredients in the jobs intervention for the unemployed. Journal of Consulting and Clinical Psychology **65**(5), 867–877 (1997)
11. Wood, S.: Generalized Additive Models: An Introduction With R. Chapman & Hall/CRC, Boca Raton (2006)
12. Zellner, A.: An efficient method of estimating seemingly unrelated regressions and tests for aggregation bias. Journal of the American Statistical Association **57**, 348–368 (1962)