Supplemental materials containing causal mediation effect definitions for subgroups, a simulated data example, and SAS and R syntax and code for three examples for “A novel approach to estimate moderated treatment effects and moderated mediated effects with continuous moderators”, Submitted to *Tutorials in* *Psychological Methods*.

**Simulation Study**

A Monte Carlo simulation study was conducted to compare the statistical performance of two methods for detecting a moderated treatment effect on a continuous outcome variable moderated by a continuous variable. The effects of interest include the test of a moderated treatment effect, the treatment effect for individuals that scored below the median on the continuous moderator variable, and the treatment effect for individuals that scored above the median on the continuous moderator variable. The effects were estimated by dichotomizing the continuous moderator variable via a median split and by applying the simulation-based approach. It is hypothesized that the simulation-based approach will result in higher statistical power to detect moderated treatment effects and subgroup effects compared to dichotomization, since it does not require dichotomizing the naturally continuous moderator variable prior to the estimation of moderated treatment effects.

**Data-Generating Model**

SAS 9.4 was used to conduct a Monte Carlo simulation. The following equations represent the linear regression model used to generate the data where is an observed value of and is the sample median.

|  |  |
| --- | --- |
|  | (S1) |
|  |  (S2) |
|   | (S3) |

Data were simulated for sample sizes *N* = 100 and *N* = 250 with 500 replications per condition. The following six conditions were simulated for each sample size: zero treatment effect and zero interaction (), zero treatment effect at *C* = 0 and small interaction (), zero treatment effect at *C* = 0 and no interaction (), medium treatment effect and zero interaction (), medium treatment effect at *C* = 0 and small interaction (), and medium treatment effect at *C* = 0 and medium interaction (). The effect of the covariate on the outcome was held constant at an approximately medium effect size for all conditions (*d* = .39). Treatment effects for those that scored below the median of the covariate and for those that scored above the median of the covariate were estimated in two ways as summarized below.

**Effect Estimation**

**Dichotomization.** The following steps were carried out to estimate treatment effects for those that scored below the median of the covariate and for those that scored above the median of the covariate by dichotomizing the continuous covariate, *C*.

|  |  |
| --- | --- |
|  | (S4) |
|  |  (S5) |

The treatment effect for those below the median of *C* was estimated as and the treatment effect for those above the median of *C* as . The test of the moderated treatment effect (difference between the two groups) was estimated as . Confidence intervals for all effects were computed using percentile bootstrapping with 1,000 bootstrap samples. The sample median used for dichotomization was estimated within each bootstrap sample.

**Simulation-based Approach.** The following steps were carried out to estimate treatment effects for those that scored below the median of the covariate and the treatment effect for those that scored above the median of the covariate by applying the simulation-based approach and averaging over ranges of the continuous covariate, *C*. First, the following equation was estimated using Ordinary Least Squares:

|  |  |
| --- | --- |
|  | (S6) |

Next, we saved the parameters from the above equation and saved the RMSE (). For each observation, we simulated the potential outcomes *Yi*(1) and *Yi*(0) where :

|  |  |
| --- | --- |
|   | (S7) |
|   |  (S8) |

The treatment effect for those below the median of *C* = , the treatment effect for those above the median of *C* = , and the test of a moderated treatment effect (difference between the two groups) was estimated as . We simulated 1000 predictions per observation. Confidence intervals for all effects were computed using percentile bootstrapping with 1,000 bootstrap samples. That is, the parameters used to simulate the potential outcomes were estimated within each bootstrap sample.

**True values**

Assuming a continuous covariate, *C*, follows a normal distribution We can dichotomize it using a median split such assuming is the median for example, now contains two discrete categories. Alternatively, we can consider truncating the normal distribution and estimating the moments of the truncated normal distribution. The mean and variance of a truncated normal distribution corresponding to those that score above the median are and . Where *pdf* is the probability density function evaluated at point and *cdf* is the cumulative distribution function at point

The mean and variance for those that score below the median are and . Therefore, we can see that truncating the normal distribution at the median leaves us with two categories that have their own mean and variance. As opposed to dichotomization which implicitly assumes the variance within each discrete category is equal to zero.

Assuming the data-generating model from above and knowing the expected value of the covariate above the median is equal to and the expected value below the median is equal to , the true treatment effects are summarized in the Table S1 for each simulation condition. The closed form solution above holds for linear models and assuming the covariate follows a normal distribution.

|  |
| --- |
| Table S1 |
| *True values for the treatment effect below the median and the treatment effect above the median* |
| Condition | *a* | *h* | True effect below median | True effect above median  |
| 1 | 0 | 0 | .000 | .000 |
| 2 | 0 | .14 | -.112 | .112 |
| 3 | 0 | .39 | -.311 | .311 |
| 4 | .39 | 0 | .390 | .390 |
| 5 | .39 | .14 | .278 | .502 |
| 6 | .39 | .39 | .079 | .701 |

**Results**

**Type 1 Error and Statistical Power**

Treatment effect estimates and moderated treatment effect estimates were deemed statistically significant if zero was not contained between the 2.5th and 97.5th percentiles of the bootstrap confidence intervals in each simulation replication. Type I error rates were computed as the proportion of times across the 500 simulation replications per condition that estimates were statistically significant when true values of the respective effects were equal to zero. Type 1 error rates and statistical power were summarized for each treatment effect and moderated treatment effect separately. Table S2 summarizes the Type 1 error and statistical power for the moderated treatment effect. The only instance of Type 1 error rates not falling within the robustness interval was for the dichotomization method (Type 1 error rate = .022) for *N* = 100, *a* = 0, and *h* = 0. The simulation-based method resulted in high statistical power to detect the moderated treatment effect across all conditions and both sample sizes. When the true interaction effect was equal to .622 (*a* = 0 and *h* = .39) and *N* = 100, the power to detect the effect with dichotomization was equal to .248 and the power to detect the effect with the simulation-based approach was equal to .454. When *N* = 250 the power increased for both methods to .582 and .828, respectively.

|  |
| --- |
| Table S2 |
| *Type 1 error rates and statistical power for detecting the moderated treatment effect using dichotomization and simulation-based approach for N = 100 and N = 250.* |
|  |  |  | Type 1 error rates/Statistical power |
|  |  |  | *N* = 100 | *N* = 250 |
| *a* | *h* | True effect | Dichotomization | Simulation | Dichotomization | Simulation |
| 0 | 0 | 0 | 0.022 | 0.058 | 0.036 | 0.052 |
| 0 | 0.14 | 0.223 | 0.064 | 0.138 | 0.114 | 0.198 |
| 0 | 0.39 | 0.622 | 0.248 | 0.454 | 0.582 | 0.828 |
| 0.39 | 0 | 0 | 0.030 | 0.048 | 0.040 | 0.054 |
| 0.39 | 0.14 | 0.223 | 0.056 | 0.104 | 0.130 | 0.208 |
| 0.39 | 0.39 | 0.622 | 0.288 | 0.508 | 0.586 | 0.852 |

Table S3 summarizes the Type 1 error and statistical power for the treatment effect below the median of the continuous covariate. The Type 1 error rates for both methods were within the bounds of liberal robustness (.025 - .075; Bradley, 1978). The simulation-based approach resulted in high statistical power to detect the treatment effect below the median across all conditions and both sample sizes. When the true treatment effect was equal to .278 and *N* = 100, the power to detect the effect with dichotomization was equal to .130 and the power to detect the effect with empirical integration was equal to .196. When *N* = 250 the power increased for both methods to .284 and .384, respectively.

|  |
| --- |
| Table S3 |
| *Type 1 error rates and statistical power for detecting the treatment effect below the median of the continuous moderator variable using dichotomization and simulation-based approach for N = 100 and N = 250.* |
|  |  |  | Type 1 error rate/Statistical power |
|  |  |  | *N* = 100 | *N* = 250 |
| *a* | *h* | True effect  | Dichotomization | Simulation | Dichotomization | Simulation |
| 0 | 0 | 0 | 0.034 | 0.046 | 0.050 | 0.050 |
| 0 | 0.14 | -0.112 | 0.060 | 0.088 | 0.072 | 0.088 |
| 0 | 0.39 | -0.311 | 0.146 | 0.212 | 0.318 | 0.412 |
| 0.39 | 0.39 | 0.079 | 0.040 | 0.068 | 0.082 | 0.080 |
| 0.39 | 0.14 | 0.278 | 0.130 | 0.196 | 0.284 | 0.384 |
| 0.39 | 0 | 0.390 | 0.232 | 0.330 | 0.540 | 0.654 |

Table S4 summarizes the Type 1 error and statistical power for the treatment effect above the median of the continuous covariate. The Type 1 error rates for both methods were within the robustness interval (.025 - .075). The simulation-based approach resulted in high statistical power to detect the treatment effect above the median across all conditions and both sample sizes. When the true treatment effect was equal to .502 and *N* = 100, the power to detect the effect with dichotomization was equal to .366 and the power to detect the effect with the simulation-based approach was equal to .488. When *N* = 250 the power increased for both methods to .756 and .866, respectively.

|  |
| --- |
| Table S4 |
| *Type 1 error rates and statistical power for detecting the treatment effect above the median of the continuous moderator variable using dichotomization and the simulation-based approach for N = 100 and N = 250.* |
|  |  |  | Type 1 error rates/Statistical power |
|  |  |  | *N* = 100 | *N* = 250 |
| *a* | *h* | True effect  | Dichotomization | Simulation | Dichotomization | Simulation |
| 0 | 0 | 0 | 0.036 | 0.054 | 0.048 | 0.048 |
| 0 | 0.14 | 0.112 | 0.062 | 0.092 | 0.068 | 0.108 |
| 0 | 0.39 | 0.311 | 0.152 | 0.232 | 0.368 | 0.496 |
| 0.39 | 0 | 0.390 | 0.226 | 0.294 | 0.542 | 0.668 |
| 0.39 | 0.14 | 0.502 | 0.366 | 0.488 | 0.756 | 0.866 |
| 0.39 | 0.39 | 0.701 | 0.586 | 0.758 | 0.964 | 0.994 |

**Bias**

Bias was computed as the difference between the treatment effect estimate and the true value of the respective treatment effect within each simulation replication. Bias was summarized for each treatment effect and the moderated treatment effect separately (Tables S5 – S7). For both treatment effects and the test of the moderated treatment effect, the bias for dichotomization and the simulation-based approach were similar in magnitude. On average, the bias was higher for the moderated treatment effect using dichotomization vs. the simulation-based approach (-0.003 vs. <0.001; see Table S5).

|  |
| --- |
| Table S5 |
| *Bias of the estimated moderated treatment effect using dichotomization and the simulation-based approach for N = 100 and N = 250.* |
|  |  |  | Bias |
|  |  |  | *N* = 100 | *N* = 250 |
| *a* | *h* | True effect | Dichotomization | Simulation | Dichotomization | Simulation |
| 0 | 0 | 0 | -0.021 | -0.007 | <-0.001 | -0.006 |
| 0 | 0.14 | 0.223 | 0.006 | 0.009 | -0.004 | 0.006 |
| 0 | 0.39 | 0.622 | -0.023 | -0.018 | -0.021 | -0.013 |
| 0.39 | 0 | 0 | -0.009 | -0.002 | 0.010 | 0.012 |
| 0.39 | 0.14 | 0.223 | -0.016 | -0.012 | 0.019 | 0.022 |
| 0.39 | 0.39 | 0.622 | 0.033 | 0.019 | -0.012 | -0.004 |

On average, the bias was lower for the treatment effect below the median using dichotomization vs. the simulation-based approach (<-0.001 vs. -0.003; see Table S6).

|  |
| --- |
| Table S6 |
| *Bias of the estimated treatment effect below the median using dichotomization and the simulation-based approach for N = 100 and N = 250.* |
|  |  |  | Bias |
|  |  |  | *N* = 100 | *N* = 250 |
| *a* | *h* | True effect  | Dichotomization | Simulation | Dichotomization | Simulation |
| 0 | 0 | 0 | 0.018 | 0.013 | -0.011 | -0.008 |
| 0 | 0.14 | -0.112 | 0.007 | 0.005 | 0.013 | 0.009 |
| 0 | 0.39 | -0.311 | 0.011 | 0.011 | 0.018 | 0.017 |
| 0.39 | 0.39 | 0.079 | -0.033 | -0.027 | 0.017 | 0.009 |
| 0.39 | 0.14 | 0.278 | -0.012 | -0.013 | -0.014 | -0.015 |
| 0.39 | 0 | 0.390 | -0.015 | -0.019 | -0.008 | -0.011 |

On average, the bias was higher for the treatment effect above the median using dichotomization vs. the simulation-based approach (-0.004 vs. -0.002; see Table S7).

|  |
| --- |
| Table S7 |
| *Bias of the estimated treatment effect above the median using dichotomization and the simulation-based approach for N = 100 and N = 250.* |
|  |  |  | Bias |
|  |  |  | *N* = 100 | *N* = 250 |
| *a* | *h* | True effect  | Dichotomization | Simulation | Dichotomization | Simulation |
| 0 | 0 | 0 | -0.003 | 0.006 | -0.011 | -0.014 |
| 0 | 0.14 | 0.112 | 0.013 | 0.014 | 0.009 | 0.014 |
| 0 | 0.39 | 0.311 | -0.013 | -0.007 | -0.003 | 0.004 |
| 0.39 | 0 | 0.390 | -0.024 | -0.021 | 0.001 | <0.001 |
| 0.39 | 0.14 | 0.502 | -0.028 | -0.025 | 0.006 | 0.007 |
| 0.39 | 0.39 | 0.701 | <-0.001 | -0.008 | 0.004 | 0.005 |

**Summary**

Overall, dichotomization and the simulation-based approach resulted in similar Type 1 error rates and bias. The simulation-based approach resulted in higher statistical power across all conditions. On average, the power to detect the moderated treatment effect using dichotomization was .214 and using the simulation-based approach was .339 which was approximately a 58% increase in statistical power. On average, the power to detect the treatment effect below the median using dichotomization was .190 and using the simulation-based approach was .251 which was approximately a 32% increase in statistical power. On average, the power to detect the treatment effect above the median using dichotomization was .409 and using the simulation-based approach was .500 which was approximately a 22% increase in statistical power. In summary, the simulation-based approach is a promising method for detecting moderated treatment effects and estimating subgroup effects with continuous variable moderators.

**Mediation Formula and XM Interaction**

**Mediation Formula**

Pearl (2001, 2012) described the mediation formula which is a non-parametric definition for mediated effects. The mediation formula for the conditional indirect (mediated) effect (IE) assuming *X* is a treatment variable:

|  |  |
| --- | --- |
|  | (S9) |

Subgroup mediated effects can then be defined similarly as described for the treatment effect in Equation 3 in the main manuscript by averaging over a specific range (e.g., subgroup of individuals who scored above *c* of *C* would sum over values > *c* to ∞) of the continuous moderator that corresponds to the defined subgroup instead of averaging over the full distribution of the continuous moderator. The average mediated effect is then defined as the weighted average of the subgroup (*s* of S) specific effects:

|  |  |
| --- | --- |
|  | (S10) |

In the main paper, we only defined subgroup effects for the Pure Natural Indirect effect (PNIE). In linear models with no treatment-mediator (*XM*) interaction, the PNIE=*ab*. The subgroup effects can be extended to all causal mediation effects including the total natural indirect effect (TNIE), the pure natural direct effect (PNDE), the total natural direct effect (TNDE), the controlled direct effect (CDE), and the total effect (TE).

The mediation formula for the conditional TNIE assuming *X* is a treatment variable:

|  |  |
| --- | --- |
|  | (S11) |

The average TNIE is then defined as the weighted average of the subgroup (*s* of S) specific effects:

|  |  |
| --- | --- |
|  | (S12) |

The mediation formula for the conditional PNDE assuming *X* is a treatment variable:

|  |  |
| --- | --- |
|  | (S13) |

The average PNDE is then defined as the weighted average of the subgroup (*s* of S) specific effects:

|  |  |
| --- | --- |
|  | (S14) |

The mediation formula for the conditional TNDE assuming *X* is a treatment variable:

|  |  |
| --- | --- |
|  | (S15) |

The average TNDE is then defined as the weighted average of the subgroup (*s* of S) specific effects:

|  |  |
| --- | --- |
|  | (S16) |

The mediation formula for the conditional CDE assuming *X* is a treatment variable:

|  |  |
| --- | --- |
|  | (S17) |

The average CDE is then defined as the weighted average of the subgroup (*s* of S) specific effects:

|  |  |
| --- | --- |
|  | (S18) |

Finally, the average TE is then defined as the sum of PNIE+TNDE or TNIE+PNDE which can be estimated for each subgroup specifically or averaged over the subgroups for the average TE.

**Simulated Empirical Example**

SAS 9.4 was used to simulate data based on a recent study that investigated mediating mechanisms of package shape (*X*) on perceived product healthiness (*Y*) through its effect on perceived package slimness (*M*) and the moderating effect of BMI on the perceived package slimness-perceived product healthiness relation in a group of adult females (Study 2; Yarar, Machiels, & Orth, 2019). Data were simulated for *N* = 300 individuals using the reported model coefficients and descriptive statistics from Study 2 (Yarar, et al., 2019). The following equations represent the linear regression model used to generate the data where is an observed value of and is the sample median.

|  |  |
| --- | --- |
|  | (S19) |
|  | (S20) |
|  | (S21) |
|  | (S22) |

**Syntax and Simulated Datasets for Demonstration**

The annotated code SAS and R syntax is applied to three simulated examples. Five-hundred observations were simulated from the following equations using SAS 9.4.

|  |  |
| --- | --- |
|  | (S23) |
|  | (S24) |
|  | (S25) |
|  | (S26) |
|  | (S27) |

The first example in the syntax is a moderated treatment effect example. The focal outcome for the first example is *M2*. The treatment effect is the effect of *X* on *M2* and this effect is moderated by *M1*. Researchers can replace *X* with the binary treatment variable in their own dataset, *M2* with the continuous outcome variable in their dataset, and *M1* with the continuous moderator variable in their own dataset.

 The second example in the syntax is a moderated mediated effect example. The focal mediator is *M2* and the focal outcome is *Y2.* The effect of treatment on the mediator (*X*->*M2*; *a* path) is moderated by *M1* through the interaction term *XM1*. The effect of the mediator on the outcome (*M2*->*Y2*; *b* path) is moderated by *M1* through the interaction term *M1M2*. The direct effect of treatment on the outcome (*X*->*Y2*; *c’* path) is moderated by *M1* through the interaction term *XM1*. Although the data were simulated with an *XM2* interaction, this interaction term is ignored in this example to demonstrate moderated mediation of the traditional mediated effect *ab.* Finally, researchers do not have to have two-waves of data to apply this syntax. Researchers may have one measure of the mediator which can replace *M2*, one measure of the outcome which can replace *Y2*, and one or more continuous covariates which can replace *M1* and *Y1*. The only interaction terms involve one of these covariates, i.e., *M1*.

The third example in the syntax is a moderated causal mediation effects example. The focal mediator is *M2* and the focal outcome is *Y2.* The effect of treatment on the mediator (*X*->*M2*; *a* path) is moderated by *M1* through the interaction term *XM1*. The effect of the mediator on the outcome (*M2*->*Y2*; *b* path) is moderated by *M1* through the interaction term *M1M2*. The direct effect of treatment on the outcome (*X*->*Y2*; *c’* path) is moderated by *M1* through the interaction term *XM1*, and the effect of the mediator on the outcome (*M2*->*Y2*) is moderated by treatment through the *XM2* interaction term*.* In this example, there are six causal effects that are estimated, the PNIE, TNIE, PNDE, TNDE, CDE, and TE. The CDE is estimated holding the *M2* at the mean of the baseline *M1*.Finally, researchers do now have to have two-waves of data to apply this syntax. Researchers may have one measure of the mediator which can replace *M2*, one measure of the outcome which can replace *Y2*, and one or more continuous covariates which can replace *M1* and *Y1*. The only interaction terms involve one of these covariates, i.e., *M1*.

**Dataset**

A csv file called *suppdata.csv* can be directly retrieved from the following Open Science Framework link (https://osf.io/r4e5y/?view\_only=239d81da3d6d4bd09496c5b31139e17d).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | x | m1 | y1 | m2 | y2 | m1m2 | xm1 | xm2 |
| 1 | 1 | 0.40383 | 2.3885 | -0.28274 | 0.4614 | -0.11418 | 0.40383 | -0.28274 |
| 2 | 0 | 1.34236 | -0.18502 | 1.6876 | 2.16279 | 2.26536 | 0 | 0 |
| 3 | 1 | 0.26283 | 0.4917 | 0.0179 | 1.9373 | 0.00471 | 0.26283 | 0.0179 |
| 4 | 0 | -0.04937 | 1.52412 | -0.05713 | -0.50062 | 0.00282 | 0 | 0 |
| 5 | 0 | -1.27735 | -0.95668 | 0.37601 | -0.42403 | -0.48029 | 0 | 0 |
| 6 | 0 | -1.28895 | 0.32438 | -1.10241 | -1.66039 | 1.42095 | 0 | 0 |
| 7 | 1 | -0.86471 | 1.51754 | 0.4095 | 1.06966 | -0.3541 | -0.86471 | 0.4095 |
| 8 | 0 | 0.92124 | 1.49475 | 0.94008 | 0.26562 | 0.86604 | 0 | 0 |
| 9 | 1 | -0.18568 | 4.25477 | 0.72651 | 2.70942 | -0.1349 | -0.18568 | 0.72651 |
| 10 | 1 | 0.30707 | 0.23748 | -0.02614 | 1.76333 | -0.00803 | 0.30707 | -0.02614 |
| 11 | 1 | -1.06922 | 0.0376 | -0.78251 | -0.95275 | 0.83668 | -1.06922 | -0.78251 |
| 12 | 0 | 0.74741 | 1.11544 | -0.23502 | 2.51474 | -0.17566 | 0 | 0 |
| 13 | 1 | 0.6378 | -2.51767 | -0.42782 | -0.96827 | -0.27287 | 0.6378 | -0.42782 |
| 14 | 0 | 0.96683 | -1.25781 | -1.08546 | -0.21255 | -1.04946 | 0 | 0 |
| 15 | 1 | 0.00063 | -0.30486 | -0.23562 | -0.10337 | -0.00015 | 0.00063 | -0.23562 |
| 16 | 0 | -0.50976 | 0.93332 | -0.5207 | -1.75913 | 0.26543 | 0 | 0 |
| 17 | 0 | 1.02002 | 2.42122 | 1.14946 | 0.9272 | 1.17247 | 0 | 0 |
| 18 | 0 | -0.12914 | 1.59075 | -0.93794 | 0.39159 | 0.12113 | 0 | 0 |
| 19 | 1 | 0.83071 | 3.29001 | 1.16207 | 3.0159 | 0.96534 | 0.83071 | 1.16207 |
| 20 | 1 | -0.39064 | -0.86058 | 1.71445 | -0.14326 | -0.66973 | -0.39064 | 1.71445 |
| 21 | 1 | -0.53868 | -0.16197 | 0.12885 | -0.54902 | -0.06941 | -0.53868 | 0.12885 |
| 22 | 1 | -0.6241 | -3.24868 | 0.49688 | -1.92076 | -0.3101 | -0.6241 | 0.49688 |
| 23 | 1 | -1.23695 | 0.68677 | -0.82904 | -0.79792 | 1.02549 | -1.23695 | -0.82904 |
| 24 | 0 | -1.01189 | 0.32598 | 1.40589 | -1.16842 | -1.42261 | 0 | 0 |
| 25 | 0 | 1.6245 | 3.09262 | 0.271 | 3.07251 | 0.44023 | 0 | 0 |
| 26 | 0 | -0.27232 | -0.24053 | 0.24223 | -1.90455 | -0.06597 | 0 | 0 |
| 27 | 1 | 0.57444 | -2.2531 | 0.91995 | 1.63367 | 0.52845 | 0.57444 | 0.91995 |
| 28 | 1 | -0.37758 | 0.23596 | -0.05603 | 1.06714 | 0.02115 | -0.37758 | -0.05603 |
| 29 | 1 | -1.38204 | 0.63538 | -1.09472 | -0.75307 | 1.51294 | -1.38204 | -1.09472 |
| 30 | 1 | -0.00816 | 5.81853 | 2.13157 | 4.85993 | -0.0174 | -0.00816 | 2.13157 |
| 31 | 1 | 1.30557 | -1.85219 | 0.36033 | 1.33164 | 0.47043 | 1.30557 | 0.36033 |
| 32 | 0 | -1.65099 | 2.0475 | 0.23261 | -0.70969 | -0.38404 | 0 | 0 |
| 33 | 1 | 0.33366 | -3.23541 | 0.01061 | -0.18618 | 0.00354 | 0.33366 | 0.01061 |
| 34 | 0 | -0.47614 | -2.61073 | 1.30507 | -1.64071 | -0.6214 | 0 | 0 |
| 35 | 0 | -0.61353 | -0.91959 | -1.81974 | -1.32244 | 1.11646 | 0 | 0 |
| 36 | 1 | -0.26717 | 0.26385 | 0.76797 | -0.0913 | -0.20518 | -0.26717 | 0.76797 |
| 37 | 0 | 0.26856 | 0.45353 | -1.7748 | -1.37159 | -0.47664 | 0 | 0 |
| 38 | 0 | 0.71102 | -1.42078 | 2.00481 | -1.57351 | 1.42546 | 0 | 0 |
| 39 | 0 | 0.69558 | -1.59813 | 1.12216 | 0.42766 | 0.78055 | 0 | 0 |
| 40 | 1 | 0.57925 | -3.10139 | -0.08955 | -0.65981 | -0.05187 | 0.57925 | -0.08955 |
| 41 | 0 | -0.67194 | -5.09115 | 2.76778 | -0.57824 | -1.85978 | 0 | 0 |
| 42 | 0 | 0.09981 | 0.02907 | 1.78623 | -0.17374 | 0.17828 | 0 | 0 |
| 43 | 1 | -0.44192 | -0.81549 | 2.02688 | 1.02485 | -0.89572 | -0.44192 | 2.02688 |
| 44 | 0 | 0.6281 | -0.29395 | 0.57474 | 1.50571 | 0.36099 | 0 | 0 |
| 45 | 1 | 1.18488 | 0.74407 | 1.43138 | 3.80986 | 1.69601 | 1.18488 | 1.43138 |
| 46 | 1 | 0.6758 | -1.63563 | 1.29941 | 0.74792 | 0.87814 | 0.6758 | 1.29941 |
| 47 | 1 | 0.77901 | 2.39736 | 1.58716 | 3.27546 | 1.23641 | 0.77901 | 1.58716 |
| 48 | 0 | -1.1695 | -2.75122 | 0.61439 | -1.04214 | -0.71853 | 0 | 0 |
| 49 | 1 | 1.29371 | 0.91726 | 1.16016 | 1.06583 | 1.50091 | 1.29371 | 1.16016 |
| 50 | 1 | -1.11582 | -3.63239 | 0.91516 | 1.1911 | -1.02115 | -1.11582 | 0.91516 |
| 51 | 0 | -0.48779 | 1.44002 | -1.72157 | -0.81197 | 0.83977 | 0 | 0 |
| 52 | 1 | 0.24626 | -0.74361 | -0.55412 | 1.75508 | -0.13645 | 0.24626 | -0.55412 |
| 53 | 1 | -0.56506 | -0.61619 | -0.3737 | 0.73802 | 0.21116 | -0.56506 | -0.3737 |
| 54 | 1 | -0.17502 | 0.63487 | -0.66266 | 0.28237 | 0.11598 | -0.17502 | -0.66266 |
| 55 | 1 | -0.88206 | -4.59996 | -1.34432 | -4.33708 | 1.18578 | -0.88206 | -1.34432 |
| 56 | 1 | -1.91805 | -3.211 | -0.81143 | -2.53184 | 1.55637 | -1.91805 | -0.81143 |
| 57 | 1 | -1.58629 | 0.0018 | 0.38558 | -0.64104 | -0.61164 | -1.58629 | 0.38558 |
| 58 | 0 | 0.25131 | 0.39586 | -2.14724 | -1.7811 | -0.53962 | 0 | 0 |
| 59 | 1 | -0.69488 | -1.65674 | -0.30079 | -1.31313 | 0.20902 | -0.69488 | -0.30079 |
| 60 | 0 | 0.95586 | 0.76436 | 1.21637 | 2.75668 | 1.16269 | 0 | 0 |
| 61 | 0 | 0.41267 | -2.07637 | 0.92121 | -0.39923 | 0.38016 | 0 | 0 |
| 62 | 1 | 0.26043 | 0.96248 | 0.88499 | 1.3027 | 0.23047 | 0.26043 | 0.88499 |
| 63 | 0 | -2.48801 | -2.60662 | -1.29449 | -1.81272 | 3.22069 | 0 | 0 |
| 64 | 1 | -1.68717 | 0.22586 | -0.98672 | -0.89013 | 1.66477 | -1.68717 | -0.98672 |
| 65 | 0 | -1.07723 | 2.79013 | -3.93511 | -0.2841 | 4.23902 | 0 | 0 |
| 66 | 0 | -2.32343 | -0.88099 | -1.49864 | -0.69264 | 3.48199 | 0 | 0 |
| 67 | 0 | 0.18555 | -1.70967 | -0.70116 | -0.55755 | -0.1301 | 0 | 0 |
| 68 | 0 | -0.99833 | -2.13588 | -0.23424 | -0.12909 | 0.23385 | 0 | 0 |
| 69 | 0 | 1.12546 | 1.13699 | 1.87051 | 3.47301 | 2.10519 | 0 | 0 |
| 70 | 1 | -1.32339 | 2.8037 | -2.09798 | -0.46079 | 2.77644 | -1.32339 | -2.09798 |
| 71 | 0 | -1.26772 | -3.26979 | -0.20961 | -0.87475 | 0.26573 | 0 | 0 |
| 72 | 0 | -0.26061 | -1.63306 | 1.78264 | 0.79394 | -0.46458 | 0 | 0 |
| 73 | 0 | -1.28928 | 0.12412 | 0.78992 | -0.59582 | -1.01842 | 0 | 0 |
| 74 | 0 | -1.30959 | -0.48837 | 0.77072 | -1.40554 | -1.00933 | 0 | 0 |
| 75 | 0 | 1.59193 | -0.00822 | -0.36789 | -0.85622 | -0.58565 | 0 | 0 |
| 76 | 1 | -1.03331 | 3.99945 | -0.38569 | 1.35535 | 0.39854 | -1.03331 | -0.38569 |
| 77 | 0 | -0.03895 | 0.56679 | 1.56189 | 2.11641 | -0.06084 | 0 | 0 |
| 78 | 1 | 1.08431 | -0.10822 | 1.41323 | 2.63876 | 1.53237 | 1.08431 | 1.41323 |
| 79 | 1 | 1.90384 | 2.19192 | 2.32128 | 5.6479 | 4.41935 | 1.90384 | 2.32128 |
| 80 | 1 | 1.62007 | 0.12022 | 1.31092 | 2.35317 | 2.12377 | 1.62007 | 1.31092 |
| 81 | 0 | -1.27704 | -1.06357 | -1.41775 | -2.22362 | 1.81052 | 0 | 0 |
| 82 | 1 | -1.36302 | 1.17791 | -0.37899 | -1.03867 | 0.51657 | -1.36302 | -0.37899 |
| 83 | 1 | -0.51724 | -0.02095 | 0.68137 | -1.07732 | -0.35243 | -0.51724 | 0.68137 |
| 84 | 1 | -1.79156 | -0.73751 | -0.51816 | -1.90004 | 0.92832 | -1.79156 | -0.51816 |
| 85 | 1 | -0.8384 | -4.28101 | -0.57236 | -1.00291 | 0.47987 | -0.8384 | -0.57236 |
| 86 | 0 | 0.34123 | -0.82891 | 0.94485 | -1.03288 | 0.32242 | 0 | 0 |
| 87 | 0 | -0.46403 | 0.22375 | -0.24488 | 0.30627 | 0.11363 | 0 | 0 |
| 88 | 0 | -0.57214 | 0.17388 | 0.76715 | 0.86921 | -0.43891 | 0 | 0 |
| 89 | 1 | 0.51297 | -0.92049 | 1.88721 | 1.26124 | 0.96808 | 0.51297 | 1.88721 |
| 90 | 1 | 0.45527 | -3.13025 | 1.08636 | -0.35799 | 0.49459 | 0.45527 | 1.08636 |
| 91 | 0 | -1.1878 | 3.06073 | -0.3914 | -0.21646 | 0.4649 | 0 | 0 |
| 92 | 0 | -0.26913 | -1.50311 | -1.69449 | -1.69904 | 0.45603 | 0 | 0 |
| 93 | 1 | -0.70579 | 2.63974 | -0.09061 | -0.46127 | 0.06395 | -0.70579 | -0.09061 |
| 94 | 1 | -0.82684 | -0.1139 | 1.11144 | 0.17144 | -0.91899 | -0.82684 | 1.11144 |
| 95 | 0 | -1.3388 | -0.90539 | 2.18902 | -0.59789 | -2.93066 | 0 | 0 |
| 96 | 0 | -0.49228 | 0.62159 | 1.52022 | -0.02298 | -0.74837 | 0 | 0 |
| 97 | 0 | -0.71204 | -1.52761 | -0.76921 | -3.247 | 0.5477 | 0 | 0 |
| 98 | 1 | -0.20688 | -0.09692 | -0.35929 | 1.43987 | 0.07433 | -0.20688 | -0.35929 |
| 99 | 1 | -0.43261 | -2.02706 | -0.39765 | -1.03646 | 0.17203 | -0.43261 | -0.39765 |
| 100 | 0 | -0.76884 | 0.87064 | -0.62361 | -0.36071 | 0.47946 | 0 | 0 |
| 101 | 0 | -1.19363 | -0.44351 | -0.69207 | -1.21767 | 0.82607 | 0 | 0 |
| 102 | 1 | 0.32639 | -0.00058 | -0.53418 | 0.83248 | -0.17435 | 0.32639 | -0.53418 |
| 103 | 1 | 0.39516 | -0.30562 | 0.43668 | 2.14167 | 0.17256 | 0.39516 | 0.43668 |
| 104 | 1 | -0.24604 | -2.03218 | 1.38343 | -0.24858 | -0.34038 | -0.24604 | 1.38343 |
| 105 | 1 | 1.53688 | 0.89048 | 1.8459 | 3.19796 | 2.83694 | 1.53688 | 1.8459 |
| 106 | 1 | 1.37768 | -2.23699 | 2.21586 | 3.36015 | 3.05275 | 1.37768 | 2.21586 |
| 107 | 1 | 0.37294 | 1.93121 | 1.10576 | 0.29044 | 0.41238 | 0.37294 | 1.10576 |
| 108 | 1 | 2.22597 | 4.96374 | 1.36701 | 4.95199 | 3.04293 | 2.22597 | 1.36701 |
| 109 | 0 | 0.06249 | 0.88955 | 1.25341 | 0.84637 | 0.07832 | 0 | 0 |
| 110 | 1 | 0.65334 | 1.4455 | 1.39393 | 2.94691 | 0.91072 | 0.65334 | 1.39393 |
| 111 | 1 | 0.2868 | 0.8106 | -0.99838 | 1.01467 | -0.28633 | 0.2868 | -0.99838 |
| 112 | 0 | 0.98561 | -0.38321 | 0.77526 | 1.6561 | 0.7641 | 0 | 0 |
| 113 | 1 | -0.47672 | 0.91172 | -1.1108 | -0.54837 | 0.52954 | -0.47672 | -1.1108 |
| 114 | 0 | 0.26083 | 0.10607 | -0.412 | 0.8001 | -0.10746 | 0 | 0 |
| 115 | 1 | 0.73917 | 1.85168 | 0.31926 | 3.26647 | 0.23599 | 0.73917 | 0.31926 |
| 116 | 0 | -0.38421 | 2.577 | 0.52569 | 0.2299 | -0.20198 | 0 | 0 |
| 117 | 1 | -0.43099 | 0.94468 | -0.47079 | -0.02099 | 0.20291 | -0.43099 | -0.47079 |
| 118 | 0 | 1.46437 | 4.11784 | 0.8749 | 0.43228 | 1.28118 | 0 | 0 |
| 119 | 0 | -1.30095 | -2.46533 | -1.54927 | -2.58872 | 2.01552 | 0 | 0 |
| 120 | 1 | 0.46571 | -0.01234 | 2.13616 | 4.79153 | 0.99484 | 0.46571 | 2.13616 |
| 121 | 1 | -1.27319 | 1.12764 | -0.6182 | -1.37697 | 0.78708 | -1.27319 | -0.6182 |
| 122 | 1 | -0.63126 | 0.48038 | -0.34335 | -0.35024 | 0.21674 | -0.63126 | -0.34335 |
| 123 | 1 | 0.06687 | 0.55653 | 2.0049 | 3.5166 | 0.13406 | 0.06687 | 2.0049 |
| 124 | 0 | 1.15662 | 4.23411 | 0.21248 | 3.68701 | 0.24576 | 0 | 0 |
| 125 | 1 | 1.93807 | 0.23505 | 1.24808 | 4.02565 | 2.41887 | 1.93807 | 1.24808 |
| 126 | 1 | 1.31241 | 1.38056 | 3.20854 | 5.32349 | 4.21092 | 1.31241 | 3.20854 |
| 127 | 0 | -1.06934 | -3.0429 | -2.01187 | -1.18793 | 2.15138 | 0 | 0 |
| 128 | 0 | -0.28323 | -0.52556 | 0.1525 | -1.90643 | -0.04319 | 0 | 0 |
| 129 | 0 | -1.86008 | -3.68214 | 0.75813 | -1.52943 | -1.41018 | 0 | 0 |
| 130 | 1 | 0.30469 | -5.23489 | -0.76862 | -1.89796 | -0.23419 | 0.30469 | -0.76862 |
| 131 | 0 | -0.49216 | 1.81528 | 0.32777 | 1.68415 | -0.16131 | 0 | 0 |
| 132 | 1 | 0.37553 | -1.30907 | 2.43036 | 2.28808 | 0.91266 | 0.37553 | 2.43036 |
| 133 | 1 | 0.17195 | -0.81925 | -0.18245 | 0.11106 | -0.03137 | 0.17195 | -0.18245 |
| 134 | 0 | -1.18945 | -2.32065 | 0.13673 | -0.84654 | -0.16263 | 0 | 0 |
| 135 | 1 | -1.50363 | -3.54958 | -0.84259 | -3.97545 | 1.26694 | -1.50363 | -0.84259 |
| 136 | 0 | 1.90385 | -2.29533 | 0.1897 | -1.31425 | 0.36116 | 0 | 0 |
| 137 | 0 | 1.0816 | -2.81944 | -1.32569 | -0.35979 | -1.43387 | 0 | 0 |
| 138 | 0 | -0.153 | -0.26 | -0.52256 | -0.41246 | 0.07995 | 0 | 0 |
| 139 | 0 | -1.64972 | 0.71016 | -0.78177 | 0.72971 | 1.2897 | 0 | 0 |
| 140 | 0 | 0.51084 | -0.61942 | 0.42688 | 0.36923 | 0.21807 | 0 | 0 |
| 141 | 1 | 0.48826 | 6.13539 | 1.20716 | 2.70925 | 0.58941 | 0.48826 | 1.20716 |
| 142 | 1 | -2.13481 | -2.54932 | 0.95021 | -2.09191 | -2.02852 | -2.13481 | 0.95021 |
| 143 | 1 | -0.35913 | 1.44062 | -0.77593 | 1.17984 | 0.27866 | -0.35913 | -0.77593 |
| 144 | 1 | -0.55287 | -0.12198 | 1.38026 | 1.841 | -0.7631 | -0.55287 | 1.38026 |
| 145 | 0 | -1.05413 | -1.88476 | -0.31324 | -1.34579 | 0.33019 | 0 | 0 |
| 146 | 1 | 1.59209 | 2.53792 | 0.25491 | 2.55079 | 0.40584 | 1.59209 | 0.25491 |
| 147 | 0 | -1.0413 | 1.59241 | -1.82234 | -0.06286 | 1.8976 | 0 | 0 |
| 148 | 1 | -0.87113 | -0.54056 | -0.22951 | -3.12085 | 0.19994 | -0.87113 | -0.22951 |
| 149 | 0 | 0.20633 | 3.29569 | 0.41575 | 2.77728 | 0.08578 | 0 | 0 |
| 150 | 0 | 0.10359 | -1.01059 | 0.88166 | 0.70381 | 0.09133 | 0 | 0 |
| 151 | 0 | -0.5874 | -2.66456 | -0.32737 | -1.34677 | 0.1923 | 0 | 0 |
| 152 | 0 | -0.28542 | -0.14152 | 0.48929 | -1.02064 | -0.13965 | 0 | 0 |
| 153 | 1 | 0.92063 | -3.19541 | 2.02203 | -0.04145 | 1.86153 | 0.92063 | 2.02203 |
| 154 | 0 | 1.2384 | 1.39037 | -0.15405 | 2.48021 | -0.19077 | 0 | 0 |
| 155 | 1 | 0.04737 | 1.17784 | 0.50239 | 0.64662 | 0.0238 | 0.04737 | 0.50239 |
| 156 | 1 | 0.19723 | -4.98483 | 0.53216 | -2.88297 | 0.10496 | 0.19723 | 0.53216 |
| 157 | 0 | -0.59237 | -0.70339 | 0.61599 | 0.89431 | -0.36489 | 0 | 0 |
| 158 | 0 | -0.7286 | -1.82868 | -0.57528 | -1.20156 | 0.41915 | 0 | 0 |
| 159 | 1 | 0.64899 | -3.41281 | 1.78382 | -0.2751 | 1.15768 | 0.64899 | 1.78382 |
| 160 | 0 | 0.54389 | 3.08908 | 0.35131 | 1.96069 | 0.19107 | 0 | 0 |
| 161 | 1 | 0.2519 | 0.33731 | 0.87871 | 3.10945 | 0.22135 | 0.2519 | 0.87871 |
| 162 | 1 | -1.18942 | 0.1164 | -0.60746 | -1.85649 | 0.72253 | -1.18942 | -0.60746 |
| 163 | 0 | 1.37753 | 1.89445 | -0.52184 | 0.58658 | -0.71885 | 0 | 0 |
| 164 | 1 | 0.35854 | 0.44333 | -0.07643 | 0.63376 | -0.0274 | 0.35854 | -0.07643 |
| 165 | 1 | -0.51862 | -1.23582 | -0.32535 | 0.70607 | 0.16873 | -0.51862 | -0.32535 |
| 166 | 1 | 0.85684 | -2.89443 | 2.29586 | 1.03252 | 1.96719 | 0.85684 | 2.29586 |
| 167 | 1 | 0.24477 | -0.13954 | -0.66321 | 0.64068 | -0.16233 | 0.24477 | -0.66321 |
| 168 | 1 | 0.67674 | -3.24614 | -0.94201 | 0.56581 | -0.6375 | 0.67674 | -0.94201 |
| 169 | 1 | -0.42741 | -0.05325 | -0.36564 | 1.29593 | 0.15628 | -0.42741 | -0.36564 |
| 170 | 0 | 2.76957 | 1.98944 | -0.37927 | 1.51003 | -1.05041 | 0 | 0 |
| 171 | 1 | 1.39562 | -0.90272 | 1.84676 | 2.92007 | 2.57738 | 1.39562 | 1.84676 |
| 172 | 1 | -0.3673 | -3.60318 | -0.57831 | -3.96692 | 0.21241 | -0.3673 | -0.57831 |
| 173 | 1 | -1.23915 | -0.55827 | 1.57678 | -0.70546 | -1.95387 | -1.23915 | 1.57678 |
| 174 | 1 | -0.60576 | -1.18939 | -0.10533 | -0.48873 | 0.0638 | -0.60576 | -0.10533 |
| 175 | 1 | 0.75949 | 0.05201 | 0.76458 | 2.53843 | 0.58069 | 0.75949 | 0.76458 |
| 176 | 0 | -0.70548 | -0.48593 | -1.51248 | 0.18123 | 1.06702 | 0 | 0 |
| 177 | 1 | -1.16658 | 3.38753 | -0.44004 | 2.7764 | 0.51335 | -1.16658 | -0.44004 |
| 178 | 0 | 0.63838 | -2.70638 | -0.89049 | -2.52495 | -0.56847 | 0 | 0 |
| 179 | 0 | -0.54653 | 3.62637 | -0.11669 | -0.03589 | 0.06377 | 0 | 0 |
| 180 | 1 | 1.02161 | -0.23991 | 1.95967 | 3.42297 | 2.00201 | 1.02161 | 1.95967 |
| 181 | 1 | -1.70378 | -1.67138 | -1.93322 | -1.95502 | 3.29378 | -1.70378 | -1.93322 |
| 182 | 0 | -1.1632 | -1.76807 | 0.30742 | -0.58803 | -0.35759 | 0 | 0 |
| 183 | 0 | -0.40865 | -2.89119 | -0.97914 | 1.25376 | 0.40012 | 0 | 0 |
| 184 | 1 | -1.55415 | 1.71503 | -0.02629 | -0.47624 | 0.04086 | -1.55415 | -0.02629 |
| 185 | 1 | 0.43688 | 5.00478 | 1.32776 | 4.96556 | 0.58007 | 0.43688 | 1.32776 |
| 186 | 0 | -0.16376 | -2.49611 | 0.48353 | -0.02443 | -0.07918 | 0 | 0 |
| 187 | 0 | 1.2267 | -1.18882 | 1.90526 | 0.63172 | 2.33717 | 0 | 0 |
| 188 | 0 | -0.03967 | 0.11291 | -3.04574 | -0.93951 | 0.12081 | 0 | 0 |
| 189 | 0 | 1.87736 | 2.61552 | 0.88884 | 1.47943 | 1.66867 | 0 | 0 |
| 190 | 0 | -2.89241 | 2.98051 | -3.32363 | -0.45467 | 9.61332 | 0 | 0 |
| 191 | 1 | 0.79932 | -1.29583 | 1.83352 | 2.23995 | 1.46558 | 0.79932 | 1.83352 |
| 192 | 1 | -1.01712 | -0.40684 | -0.43409 | -0.59705 | 0.44152 | -1.01712 | -0.43409 |
| 193 | 1 | 0.60093 | 0.42202 | 1.99357 | 1.55009 | 1.198 | 0.60093 | 1.99357 |
| 194 | 1 | 0.23303 | -0.54065 | 0.46457 | 0.8007 | 0.10826 | 0.23303 | 0.46457 |
| 195 | 0 | 0.25088 | 0.17807 | 1.39753 | 2.01537 | 0.35061 | 0 | 0 |
| 196 | 0 | 0.48091 | 0.69467 | -0.98418 | -1.31119 | -0.4733 | 0 | 0 |
| 197 | 0 | -1.63037 | 1.74983 | -1.06835 | -0.49607 | 1.74181 | 0 | 0 |
| 198 | 0 | 0.48941 | 0.62119 | -0.09302 | -0.38189 | -0.04553 | 0 | 0 |
| 199 | 0 | -0.57429 | 2.39268 | 1.07093 | 2.64753 | -0.61503 | 0 | 0 |
| 200 | 1 | 0.53089 | 0.42899 | 2.73588 | 4.79834 | 1.45245 | 0.53089 | 2.73588 |
| 201 | 1 | 1.62856 | 0.42847 | 1.30349 | 3.26065 | 2.12281 | 1.62856 | 1.30349 |
| 202 | 0 | -0.85021 | -0.36204 | -1.67385 | -2.5545 | 1.42312 | 0 | 0 |
| 203 | 0 | 1.39677 | 0.77214 | -0.50691 | 0.26556 | -0.70804 | 0 | 0 |
| 204 | 0 | -1.15858 | 2.91242 | -1.00728 | 0.79529 | 1.16702 | 0 | 0 |
| 205 | 0 | -1.35522 | 1.83083 | -0.09904 | 0.16206 | 0.13422 | 0 | 0 |
| 206 | 0 | 0.92675 | 0.97934 | 2.2908 | 2.19328 | 2.123 | 0 | 0 |
| 207 | 0 | 0.07856 | -3.12207 | 0.10812 | 0.35137 | 0.00849 | 0 | 0 |
| 208 | 1 | -0.37488 | 1.06568 | 2.87029 | 3.22822 | -1.07601 | -0.37488 | 2.87029 |
| 209 | 1 | -0.44349 | 0.42374 | 2.02986 | 2.5233 | -0.90022 | -0.44349 | 2.02986 |
| 210 | 1 | 0.61885 | 0.40667 | 0.33609 | 2.03558 | 0.20799 | 0.61885 | 0.33609 |
| 211 | 1 | -0.34647 | 1.93655 | 0.91535 | 1.22621 | -0.31714 | -0.34647 | 0.91535 |
| 212 | 1 | 1.18799 | 3.86154 | -0.13553 | 1.32173 | -0.16101 | 1.18799 | -0.13553 |
| 213 | 0 | 1.59875 | 0.30303 | 1.33877 | 1.01222 | 2.14037 | 0 | 0 |
| 214 | 0 | 0.1359 | 0.51899 | -1.09841 | 0.16818 | -0.14927 | 0 | 0 |
| 215 | 1 | 0.89647 | -0.9894 | 1.07935 | 2.26874 | 0.9676 | 0.89647 | 1.07935 |
| 216 | 1 | 1.47805 | 1.8675 | 0.72261 | 4.17731 | 1.06806 | 1.47805 | 0.72261 |
| 217 | 1 | -0.03875 | -1.25927 | 0.26241 | 0.62809 | -0.01017 | -0.03875 | 0.26241 |
| 218 | 1 | 0.31767 | 2.10294 | 1.05705 | 1.19244 | 0.33579 | 0.31767 | 1.05705 |
| 219 | 0 | -0.71921 | 0.16994 | -2.35379 | 0.40614 | 1.69286 | 0 | 0 |
| 220 | 0 | -0.83165 | -0.6377 | 0.6181 | -0.13754 | -0.51404 | 0 | 0 |
| 221 | 1 | 1.18796 | -2.33738 | 0.78506 | 1.11157 | 0.93262 | 1.18796 | 0.78506 |
| 222 | 0 | -0.97892 | -1.58479 | 1.2968 | -0.3132 | -1.26947 | 0 | 0 |
| 223 | 0 | 0.32344 | -1.68989 | -0.0891 | -0.17052 | -0.02882 | 0 | 0 |
| 224 | 1 | 1.04711 | -2.32198 | 0.73192 | 2.4784 | 0.7664 | 1.04711 | 0.73192 |
| 225 | 1 | -0.17953 | 1.64277 | 0.62432 | 2.87993 | -0.11209 | -0.17953 | 0.62432 |
| 226 | 1 | -0.58426 | 1.44043 | -0.78887 | -1.01123 | 0.46091 | -0.58426 | -0.78887 |
| 227 | 1 | 2.17928 | -0.40051 | 1.43139 | 4.01346 | 3.1194 | 2.17928 | 1.43139 |
| 228 | 1 | 0.69581 | -0.42086 | 0.09721 | -0.88698 | 0.06764 | 0.69581 | 0.09721 |
| 229 | 1 | -2.11629 | 0.76497 | -0.47614 | -3.07449 | 1.00766 | -2.11629 | -0.47614 |
| 230 | 1 | -0.74065 | 1.92927 | 0.21829 | 0.60029 | -0.16167 | -0.74065 | 0.21829 |
| 231 | 1 | -1.01577 | -0.45368 | 1.48167 | -0.14618 | -1.50504 | -1.01577 | 1.48167 |
| 232 | 1 | -0.28233 | 0.80828 | -0.91718 | 0.09406 | 0.25895 | -0.28233 | -0.91718 |
| 233 | 1 | -0.78385 | 2.12989 | 0.04287 | 0.95371 | -0.0336 | -0.78385 | 0.04287 |
| 234 | 0 | 1.88018 | 1.5123 | 0.77999 | 2.1033 | 1.46652 | 0 | 0 |
| 235 | 0 | -0.83441 | 3.20959 | -1.19769 | -0.37143 | 0.99936 | 0 | 0 |
| 236 | 0 | -0.23637 | -0.25691 | -0.72327 | 0.82054 | 0.17096 | 0 | 0 |
| 237 | 0 | 0.47177 | -0.37261 | -0.62472 | -0.63762 | -0.29472 | 0 | 0 |
| 238 | 0 | -0.50766 | -0.1004 | -0.69232 | 0.0586 | 0.35146 | 0 | 0 |
| 239 | 1 | 0.13795 | 0.8334 | 1.38036 | 2.06156 | 0.19043 | 0.13795 | 1.38036 |
| 240 | 1 | -0.71099 | 1.4617 | -0.2024 | -1.16204 | 0.1439 | -0.71099 | -0.2024 |
| 241 | 1 | -0.4801 | 0.79256 | 0.62767 | 1.32979 | -0.30134 | -0.4801 | 0.62767 |
| 242 | 1 | -0.86379 | -1.73361 | -1.54806 | -2.52551 | 1.33719 | -0.86379 | -1.54806 |
| 243 | 0 | -1.03882 | -2.87052 | -0.06572 | -1.24539 | 0.06827 | 0 | 0 |
| 244 | 1 | -1.49526 | -0.68949 | -2.09763 | -3.14751 | 3.1365 | -1.49526 | -2.09763 |
| 245 | 1 | 0.10412 | 3.02589 | 1.85328 | 2.55403 | 0.19297 | 0.10412 | 1.85328 |
| 246 | 1 | 1.22542 | 1.73896 | 2.81119 | 6.06955 | 3.44488 | 1.22542 | 2.81119 |
| 247 | 0 | 0.83142 | -1.39928 | 0.2722 | -1.10339 | 0.22631 | 0 | 0 |
| 248 | 0 | 0.86649 | 0.92655 | -1.79733 | -2.1238 | -1.55737 | 0 | 0 |
| 249 | 1 | -0.80763 | -0.16654 | 0.99544 | 2.17221 | -0.80394 | -0.80763 | 0.99544 |
| 250 | 0 | 0.96768 | -0.48136 | -0.41718 | 2.58691 | -0.4037 | 0 | 0 |
| 251 | 1 | -1.52061 | -0.10195 | -0.04073 | -3.50181 | 0.06194 | -1.52061 | -0.04073 |
| 252 | 1 | 1.4815 | -1.85415 | 2.23541 | 3.17163 | 3.31177 | 1.4815 | 2.23541 |
| 253 | 0 | -0.49814 | -1.32427 | 1.28388 | 0.26718 | -0.63956 | 0 | 0 |
| 254 | 0 | -0.56737 | 0.29907 | -0.58355 | 0.28936 | 0.33109 | 0 | 0 |
| 255 | 1 | 0.71507 | 1.09455 | -0.0165 | 2.2726 | -0.0118 | 0.71507 | -0.0165 |
| 256 | 0 | -0.60524 | -4.89471 | 1.56038 | -2.61804 | -0.9444 | 0 | 0 |
| 257 | 0 | 0.36162 | 1.94664 | -0.20119 | 1.11149 | -0.07275 | 0 | 0 |
| 258 | 1 | 2.57254 | 1.5201 | 3.32447 | 7.33916 | 8.55233 | 2.57254 | 3.32447 |
| 259 | 0 | 0.94338 | -0.96966 | 0.15639 | 0.28252 | 0.14753 | 0 | 0 |
| 260 | 1 | -0.62686 | -2.51463 | 1.08394 | -0.72528 | -0.67947 | -0.62686 | 1.08394 |
| 261 | 1 | 0.92524 | 1.83236 | 2.80103 | 3.71836 | 2.59163 | 0.92524 | 2.80103 |
| 262 | 0 | 0.25776 | 3.27872 | 0.74304 | 0.73931 | 0.19153 | 0 | 0 |
| 263 | 0 | -0.56317 | 0.09452 | -0.98903 | 1.4593 | 0.55699 | 0 | 0 |
| 264 | 1 | -1.34155 | 0.00744 | 0.87113 | -1.02307 | -1.16867 | -1.34155 | 0.87113 |
| 265 | 0 | -0.77445 | -1.233 | 1.48572 | -1.95347 | -1.15062 | 0 | 0 |
| 266 | 1 | 0.15557 | -0.0196 | 4.19757 | 4.21819 | 0.653 | 0.15557 | 4.19757 |
| 267 | 0 | -1.57954 | 2.64653 | -0.20298 | 1.51516 | 0.32061 | 0 | 0 |
| 268 | 1 | 0.3531 | 0.56502 | 0.10761 | 0.8669 | 0.038 | 0.3531 | 0.10761 |
| 269 | 1 | -0.94738 | -0.22841 | -0.81496 | 0.04193 | 0.77207 | -0.94738 | -0.81496 |
| 270 | 1 | -0.65823 | -1.61094 | -1.23434 | -1.95047 | 0.81248 | -0.65823 | -1.23434 |
| 271 | 0 | 0.69929 | -1.36736 | 0.36222 | 0.55168 | 0.2533 | 0 | 0 |
| 272 | 1 | 0.34466 | -4.14904 | 1.57851 | 0.99954 | 0.54405 | 0.34466 | 1.57851 |
| 273 | 1 | -1.65803 | 0.39059 | -0.86273 | -1.00476 | 1.43044 | -1.65803 | -0.86273 |
| 274 | 1 | 0.06778 | 1.17825 | 2.47891 | 4.07045 | 0.16803 | 0.06778 | 2.47891 |
| 275 | 1 | 2.1512 | 0.08452 | 3.14092 | 5.55581 | 6.75675 | 2.1512 | 3.14092 |
| 276 | 1 | 0.13056 | -1.10361 | 3.69537 | 1.9635 | 0.48247 | 0.13056 | 3.69537 |
| 277 | 0 | 1.3489 | 4.28509 | -0.93813 | 1.60411 | -1.26545 | 0 | 0 |
| 278 | 1 | -0.0425 | 0.40522 | 0.56976 | 1.12714 | -0.02421 | -0.0425 | 0.56976 |
| 279 | 1 | -0.01052 | 5.05086 | 2.38592 | 3.49572 | -0.0251 | -0.01052 | 2.38592 |
| 280 | 1 | -0.25497 | -0.1661 | 0.04169 | 0.12235 | -0.01063 | -0.25497 | 0.04169 |
| 281 | 1 | 0.14749 | 0.46032 | 0.2264 | 0.83374 | 0.03339 | 0.14749 | 0.2264 |
| 282 | 1 | -1.15554 | 2.11806 | 0.95679 | 0.70483 | -1.10561 | -1.15554 | 0.95679 |
| 283 | 1 | -1.40544 | -0.927 | -0.66315 | -1.44201 | 0.93202 | -1.40544 | -0.66315 |
| 284 | 1 | 1.45678 | -0.44165 | 3.09489 | 4.05922 | 4.50858 | 1.45678 | 3.09489 |
| 285 | 0 | 0.24962 | -0.95376 | -1.08615 | 0.18964 | -0.27112 | 0 | 0 |
| 286 | 0 | -1.44162 | -1.30523 | 0.13752 | 2.00091 | -0.19825 | 0 | 0 |
| 287 | 1 | 0.18561 | 0.04782 | -0.1881 | -0.30023 | -0.03491 | 0.18561 | -0.1881 |
| 288 | 0 | -2.15249 | -0.57218 | 0.77782 | 0.04757 | -1.67426 | 0 | 0 |
| 289 | 0 | 0.66082 | 1.06314 | 0.55919 | 0.05832 | 0.36952 | 0 | 0 |
| 290 | 1 | 0.1803 | 1.17274 | 1.66003 | 2.2337 | 0.29931 | 0.1803 | 1.66003 |
| 291 | 0 | 0.15289 | 0.49151 | 0.6046 | -0.33025 | 0.09244 | 0 | 0 |
| 292 | 1 | -0.01392 | -1.09289 | 0.47685 | 1.73644 | -0.00664 | -0.01392 | 0.47685 |
| 293 | 1 | 2.15272 | 3.50721 | 1.58412 | 6.3337 | 3.41016 | 2.15272 | 1.58412 |
| 294 | 0 | -0.85344 | 0.23048 | 0.09347 | -1.42943 | -0.07977 | 0 | 0 |
| 295 | 1 | 1.75899 | 3.61809 | 2.31791 | 7.60653 | 4.07718 | 1.75899 | 2.31791 |
| 296 | 0 | 0.40031 | -0.1795 | -1.49686 | 1.09621 | -0.59921 | 0 | 0 |
| 297 | 0 | 1.23366 | 2.69601 | 2.04349 | 1.26776 | 2.52098 | 0 | 0 |
| 298 | 1 | 0.92715 | -0.91 | 2.89031 | 4.34915 | 2.67974 | 0.92715 | 2.89031 |
| 299 | 0 | -1.2425 | -2.10287 | -0.6586 | -0.88127 | 0.81831 | 0 | 0 |
| 300 | 0 | -0.69867 | 4.45931 | -0.96427 | -0.5115 | 0.6737 | 0 | 0 |
| 301 | 0 | -2.31772 | -0.99202 | -0.68822 | -2.16337 | 1.5951 | 0 | 0 |
| 302 | 1 | -0.26237 | 0.3441 | 0.46604 | -0.94395 | -0.12228 | -0.26237 | 0.46604 |
| 303 | 0 | -0.12374 | -0.55702 | -0.95421 | -1.0916 | 0.11807 | 0 | 0 |
| 304 | 1 | -0.53918 | -2.62898 | -3.14119 | -2.8104 | 1.69368 | -0.53918 | -3.14119 |
| 305 | 0 | -0.39061 | 1.22096 | -1.24256 | -2.26555 | 0.48536 | 0 | 0 |
| 306 | 0 | -1.60312 | 3.48131 | -0.6977 | 3.67169 | 1.1185 | 0 | 0 |
| 307 | 1 | 1.10928 | -0.39069 | -0.09861 | 0.53607 | -0.10938 | 1.10928 | -0.09861 |
| 308 | 1 | 2.25286 | -1.13512 | 1.67825 | 2.97623 | 3.78086 | 2.25286 | 1.67825 |
| 309 | 0 | 1.06577 | -3.22293 | 0.15382 | -0.9352 | 0.16393 | 0 | 0 |
| 310 | 1 | 0.89152 | 4.74414 | 0.4435 | 3.24703 | 0.39539 | 0.89152 | 0.4435 |
| 311 | 0 | 0.79961 | -3.39002 | 1.97821 | -1.83916 | 1.5818 | 0 | 0 |
| 312 | 0 | 0.16688 | -1.65155 | -1.70372 | -0.79729 | -0.28432 | 0 | 0 |
| 313 | 1 | 1.54594 | -1.65178 | 0.90428 | 0.38221 | 1.39796 | 1.54594 | 0.90428 |
| 314 | 0 | -2.3143 | -0.91974 | -0.55655 | -2.11189 | 1.28802 | 0 | 0 |
| 315 | 1 | -0.84771 | 4.01048 | 0.17651 | 2.79033 | -0.14963 | -0.84771 | 0.17651 |
| 316 | 1 | -0.4739 | 3.09625 | -0.72543 | 0.51601 | 0.34378 | -0.4739 | -0.72543 |
| 317 | 0 | 1.61014 | 4.13893 | 0.91874 | 2.73224 | 1.4793 | 0 | 0 |
| 318 | 0 | 0.21529 | -3.2648 | 2.57125 | 1.3672 | 0.55356 | 0 | 0 |
| 319 | 0 | 1.36683 | -0.20705 | -0.04917 | -2.06102 | -0.0672 | 0 | 0 |
| 320 | 1 | 1.73274 | 3.45736 | 3.29312 | 7.26637 | 5.70614 | 1.73274 | 3.29312 |
| 321 | 1 | -1.58261 | -1.30635 | 0.07091 | -2.63629 | -0.11223 | -1.58261 | 0.07091 |
| 322 | 1 | -0.70336 | -3.34177 | 0.37779 | -2.03446 | -0.26572 | -0.70336 | 0.37779 |
| 323 | 0 | -0.22708 | 3.08314 | 1.48268 | 1.87735 | -0.33669 | 0 | 0 |
| 324 | 1 | 0.10488 | -0.00659 | 0.64134 | 0.79265 | 0.06727 | 0.10488 | 0.64134 |
| 325 | 0 | 0.45969 | -0.73763 | 0.56631 | 0.89564 | 0.26033 | 0 | 0 |
| 326 | 0 | 0.79213 | 4.35551 | -0.69745 | 3.67431 | -0.55247 | 0 | 0 |
| 327 | 1 | -1.5168 | -0.95909 | -2.41059 | -3.16144 | 3.65637 | -1.5168 | -2.41059 |
| 328 | 0 | 1.38159 | 0.23396 | 0.40849 | 1.86112 | 0.56437 | 0 | 0 |
| 329 | 0 | -1.94256 | 2.20136 | 0.5138 | -0.39455 | -0.9981 | 0 | 0 |
| 330 | 1 | -0.51475 | -0.11348 | -0.39347 | -0.58168 | 0.20254 | -0.51475 | -0.39347 |
| 331 | 1 | -0.3978 | -2.68907 | 0.02903 | -0.72985 | -0.01155 | -0.3978 | 0.02903 |
| 332 | 1 | 0.30555 | 0.97957 | -0.29324 | 0.34076 | -0.0896 | 0.30555 | -0.29324 |
| 333 | 0 | -1.03498 | 2.32446 | -0.8502 | 0.82091 | 0.87994 | 0 | 0 |
| 334 | 0 | 0.51964 | 1.18072 | 0.90788 | 0.51402 | 0.47177 | 0 | 0 |
| 335 | 0 | -1.37689 | -1.33901 | -1.41648 | -1.75806 | 1.95033 | 0 | 0 |
| 336 | 1 | -0.11982 | -0.28382 | 0.34103 | 0.2689 | -0.04086 | -0.11982 | 0.34103 |
| 337 | 1 | 0.07195 | 0.33329 | 1.42846 | 2.95957 | 0.10277 | 0.07195 | 1.42846 |
| 338 | 0 | 0.00751 | 0.43818 | 0.79718 | 0.77477 | 0.00598 | 0 | 0 |
| 339 | 0 | -0.07998 | -0.13915 | -1.75341 | -0.73895 | 0.14024 | 0 | 0 |
| 340 | 0 | -1.42861 | 0.5661 | 0.17231 | 0.16276 | -0.24616 | 0 | 0 |
| 341 | 0 | -0.0455 | 2.7531 | -0.27734 | 0.87816 | 0.01262 | 0 | 0 |
| 342 | 1 | -0.40067 | 0.71202 | 0.99712 | 2.32497 | -0.39952 | -0.40067 | 0.99712 |
| 343 | 0 | 1.00041 | -1.65123 | 0.42488 | 1.51408 | 0.42506 | 0 | 0 |
| 344 | 1 | 0.79503 | -0.14422 | 2.87697 | 3.54162 | 2.28729 | 0.79503 | 2.87697 |
| 345 | 0 | -0.25434 | -1.27645 | 0.1197 | 0.77576 | -0.03044 | 0 | 0 |
| 346 | 0 | 0.11511 | -0.50543 | -0.08852 | -0.42763 | -0.01019 | 0 | 0 |
| 347 | 1 | 0.71742 | -1.10929 | 1.4746 | -0.02297 | 1.0579 | 0.71742 | 1.4746 |
| 348 | 0 | -0.28747 | 1.6995 | -0.23751 | -0.42688 | 0.06828 | 0 | 0 |
| 349 | 0 | 0.99086 | 0.93638 | -0.10617 | -1.18616 | -0.10519 | 0 | 0 |
| 350 | 1 | 1.33655 | 2.33788 | -0.28157 | 2.69031 | -0.37633 | 1.33655 | -0.28157 |
| 351 | 1 | -1.51959 | 1.13443 | -0.46275 | -1.64001 | 0.70319 | -1.51959 | -0.46275 |
| 352 | 1 | -0.54638 | 0.09519 | 0.18602 | -0.91287 | -0.10164 | -0.54638 | 0.18602 |
| 353 | 1 | -0.69648 | -0.20498 | 2.09186 | 1.65548 | -1.45694 | -0.69648 | 2.09186 |
| 354 | 1 | 1.16189 | 3.87967 | 0.05463 | 2.69993 | 0.06347 | 1.16189 | 0.05463 |
| 355 | 0 | 0.55133 | -1.32888 | 0.58332 | -1.17929 | 0.3216 | 0 | 0 |
| 356 | 1 | 0.93571 | -1.83423 | 1.03813 | 1.5473 | 0.9714 | 0.93571 | 1.03813 |
| 357 | 0 | -0.91852 | -2.09647 | 0.78682 | -0.94002 | -0.72271 | 0 | 0 |
| 358 | 1 | 0.22839 | 0.43914 | 3.22432 | 1.8143 | 0.73641 | 0.22839 | 3.22432 |
| 359 | 0 | -2.1107 | -0.18352 | -2.10506 | -1.59287 | 4.44315 | 0 | 0 |
| 360 | 0 | -1.84283 | 2.72593 | -0.47402 | 0.57691 | 0.87353 | 0 | 0 |
| 361 | 0 | -1.35681 | 0.38007 | -0.4237 | 0.8796 | 0.57489 | 0 | 0 |
| 362 | 1 | -1.54084 | -0.33536 | -0.35324 | -2.61984 | 0.54429 | -1.54084 | -0.35324 |
| 363 | 0 | 0.18939 | 0.98004 | -0.55398 | -1.22502 | -0.10492 | 0 | 0 |
| 364 | 1 | 1.36908 | 0.9483 | 0.51935 | 3.04893 | 0.71104 | 1.36908 | 0.51935 |
| 365 | 0 | -1.14211 | -1.25031 | -1.37533 | -1.21762 | 1.57079 | 0 | 0 |
| 366 | 1 | -1.32233 | 4.97116 | 0.04653 | 1.41544 | -0.06153 | -1.32233 | 0.04653 |
| 367 | 1 | 0.12132 | -0.08578 | 1.32273 | 1.27792 | 0.16048 | 0.12132 | 1.32273 |
| 368 | 0 | 0.11199 | -0.56075 | 0.44555 | -0.81534 | 0.0499 | 0 | 0 |
| 369 | 1 | 0.94387 | 1.08169 | 1.75539 | 3.87489 | 1.65686 | 0.94387 | 1.75539 |
| 370 | 0 | 0.61172 | 1.30573 | 0.52365 | 1.5341 | 0.32033 | 0 | 0 |
| 371 | 1 | -0.5119 | -0.0989 | -1.22862 | -1.26199 | 0.62893 | -0.5119 | -1.22862 |
| 372 | 1 | 1.04506 | -0.02825 | 1.23531 | 2.41283 | 1.29098 | 1.04506 | 1.23531 |
| 373 | 0 | -2.50492 | 0.01689 | -0.37686 | -0.80957 | 0.94401 | 0 | 0 |
| 374 | 0 | -0.32708 | 2.3376 | -0.13641 | -1.30448 | 0.04462 | 0 | 0 |
| 375 | 0 | -0.9499 | 1.77706 | -1.53309 | -0.06739 | 1.45628 | 0 | 0 |
| 376 | 0 | 0.19223 | -0.98076 | 1.87473 | -1.21271 | 0.36039 | 0 | 0 |
| 377 | 1 | -0.60645 | 1.18647 | 0.14318 | -0.8851 | -0.08683 | -0.60645 | 0.14318 |
| 378 | 1 | -0.34896 | -1.37523 | 1.59874 | 0.1726 | -0.55789 | -0.34896 | 1.59874 |
| 379 | 0 | -1.04907 | 0.89572 | -1.17214 | 0.54 | 1.22965 | 0 | 0 |
| 380 | 0 | 0.38775 | -0.26076 | -0.70924 | -0.87496 | -0.27501 | 0 | 0 |
| 381 | 1 | -0.92668 | 1.10516 | -0.8887 | -0.42112 | 0.82354 | -0.92668 | -0.8887 |
| 382 | 0 | -1.56063 | -0.63044 | 1.52583 | -1.4052 | -2.38126 | 0 | 0 |
| 383 | 0 | 0.55024 | -3.47911 | 1.35717 | -2.22413 | 0.74677 | 0 | 0 |
| 384 | 1 | 1.3318 | 3.49562 | -0.30965 | 1.77018 | -0.4124 | 1.3318 | -0.30965 |
| 385 | 1 | -1.86851 | 1.66911 | -0.68832 | -0.20262 | 1.28613 | -1.86851 | -0.68832 |
| 386 | 1 | -0.80015 | 0.00868 | 0.63167 | -0.39409 | -0.50543 | -0.80015 | 0.63167 |
| 387 | 0 | -0.6472 | -1.84173 | 1.88258 | -0.06577 | -1.21841 | 0 | 0 |
| 388 | 0 | 0.76702 | 3.44244 | -0.881 | 1.44426 | -0.67574 | 0 | 0 |
| 389 | 1 | -0.05477 | 2.31565 | -0.8277 | 1.08611 | 0.04533 | -0.05477 | -0.8277 |
| 390 | 0 | -0.77433 | -1.66521 | -2.58149 | -0.81263 | 1.99892 | 0 | 0 |
| 391 | 1 | -0.53826 | 2.58842 | -0.72094 | -0.96503 | 0.38805 | -0.53826 | -0.72094 |
| 392 | 1 | 0.56311 | -1.24766 | 1.91468 | 0.47149 | 1.07818 | 0.56311 | 1.91468 |
| 393 | 1 | 0.13791 | 0.21299 | 1.38995 | 2.31105 | 0.19169 | 0.13791 | 1.38995 |
| 394 | 0 | 0.56629 | -1.25888 | 0.57172 | -0.26058 | 0.32376 | 0 | 0 |
| 395 | 1 | -0.14987 | 0.45141 | 0.82793 | 2.6189 | -0.12408 | -0.14987 | 0.82793 |
| 396 | 1 | -0.70175 | -2.32821 | -0.33399 | -0.94672 | 0.23438 | -0.70175 | -0.33399 |
| 397 | 0 | 0.60003 | 0.05412 | -0.41502 | 1.45933 | -0.24902 | 0 | 0 |
| 398 | 1 | -0.56339 | 0.00196 | -0.72161 | -1.88751 | 0.40655 | -0.56339 | -0.72161 |
| 399 | 1 | -1.30789 | -1.62063 | -1.5839 | 0.14463 | 2.07156 | -1.30789 | -1.5839 |
| 400 | 1 | 0.77223 | 0.18519 | -0.49938 | 0.77663 | -0.38563 | 0.77223 | -0.49938 |
| 401 | 0 | -2.09847 | 2.78923 | -1.15972 | 2.68184 | 2.43365 | 0 | 0 |
| 402 | 0 | 1.03251 | -1.95115 | -1.09605 | 0.91589 | -1.13168 | 0 | 0 |
| 403 | 0 | -0.30322 | 0.8192 | 0.49538 | 0.23184 | -0.15021 | 0 | 0 |
| 404 | 1 | -0.49534 | 0.40897 | 1.57561 | 1.37698 | -0.78046 | -0.49534 | 1.57561 |
| 405 | 0 | 0.13568 | 1.75392 | -0.72885 | 0.14416 | -0.09889 | 0 | 0 |
| 406 | 1 | 0.85111 | 0.70094 | 0.49461 | 2.19066 | 0.42097 | 0.85111 | 0.49461 |
| 407 | 0 | 1.21508 | 0.37197 | 0.47334 | 1.20304 | 0.57515 | 0 | 0 |
| 408 | 0 | 1.31836 | 0.77316 | 0.9364 | 1.06998 | 1.23451 | 0 | 0 |
| 409 | 0 | 0.34082 | 0.07605 | 2.47918 | 0.65426 | 0.84495 | 0 | 0 |
| 410 | 1 | -0.14753 | -2.20899 | 0.29967 | -0.56231 | -0.04421 | -0.14753 | 0.29967 |
| 411 | 1 | -1.01745 | 0.97905 | 0.64512 | 0.01583 | -0.65638 | -1.01745 | 0.64512 |
| 412 | 1 | -1.65341 | -0.23726 | -2.05113 | -1.93916 | 3.39135 | -1.65341 | -2.05113 |
| 413 | 1 | -0.16568 | 4.37471 | 0.91449 | 2.82043 | -0.15151 | -0.16568 | 0.91449 |
| 414 | 1 | 0.51095 | 1.62847 | 0.43669 | 3.20667 | 0.22313 | 0.51095 | 0.43669 |
| 415 | 0 | -1.67495 | -1.86827 | 0.14651 | -1.37352 | -0.2454 | 0 | 0 |
| 416 | 1 | 0.87712 | -2.4805 | 0.15904 | -0.83452 | 0.13949 | 0.87712 | 0.15904 |
| 417 | 1 | -1.2765 | 2.75797 | -0.37402 | -0.29994 | 0.47744 | -1.2765 | -0.37402 |
| 418 | 1 | -0.61684 | -1.24756 | -0.25508 | -2.37502 | 0.15734 | -0.61684 | -0.25508 |
| 419 | 1 | -1.26121 | 1.04856 | -0.09929 | 1.47767 | 0.12523 | -1.26121 | -0.09929 |
| 420 | 1 | 0.77117 | -3.20386 | -2.10219 | -3.12618 | -1.62116 | 0.77117 | -2.10219 |
| 421 | 0 | 0.55654 | -3.93044 | -1.15536 | -1.54934 | -0.64301 | 0 | 0 |
| 422 | 0 | 0.17192 | 0.87323 | 1.34822 | -0.25248 | 0.23179 | 0 | 0 |
| 423 | 0 | 1.18908 | 0.10416 | -0.48056 | -1.23527 | -0.57142 | 0 | 0 |
| 424 | 1 | -1.12735 | -0.30132 | -2.19771 | -2.02467 | 2.4776 | -1.12735 | -2.19771 |
| 425 | 0 | 0.84717 | 2.92979 | -0.14181 | 1.37467 | -0.12014 | 0 | 0 |
| 426 | 0 | 1.05471 | -0.11978 | -0.4972 | -1.43294 | -0.52441 | 0 | 0 |
| 427 | 0 | 1.22671 | -0.14245 | -0.40649 | -1.02517 | -0.49865 | 0 | 0 |
| 428 | 0 | -0.5919 | -0.37474 | -0.46259 | -2.30233 | 0.27381 | 0 | 0 |
| 429 | 0 | 0.22798 | -0.8228 | -1.30267 | -3.07315 | -0.29698 | 0 | 0 |
| 430 | 0 | 0.76149 | -2.7629 | 1.04261 | 0.63704 | 0.79393 | 0 | 0 |
| 431 | 0 | -0.35937 | 2.06518 | 0.14143 | 1.77974 | -0.05083 | 0 | 0 |
| 432 | 1 | 0.08607 | 0.01199 | 1.52886 | 1.37229 | 0.1316 | 0.08607 | 1.52886 |
| 433 | 0 | -1.24751 | 0.68728 | -0.15308 | 0.24775 | 0.19097 | 0 | 0 |
| 434 | 0 | 0.3699 | -3.23295 | -0.09393 | -2.67784 | -0.03474 | 0 | 0 |
| 435 | 1 | 0.41239 | -4.15766 | 0.20285 | -0.91077 | 0.08366 | 0.41239 | 0.20285 |
| 436 | 1 | 0.41763 | 1.76594 | 0.86704 | 2.31764 | 0.3621 | 0.41763 | 0.86704 |
| 437 | 1 | -0.71412 | -1.77766 | 0.15116 | -0.65491 | -0.10794 | -0.71412 | 0.15116 |
| 438 | 0 | 1.30338 | 0.67593 | 0.6623 | 1.95285 | 0.86322 | 0 | 0 |
| 439 | 1 | 1.73706 | -2.33601 | 1.7289 | 4.537 | 3.00321 | 1.73706 | 1.7289 |
| 440 | 0 | 1.17966 | 1.70496 | -0.95329 | 2.64623 | -1.12456 | 0 | 0 |
| 441 | 1 | -0.33525 | -1.53248 | 1.25508 | 0.62508 | -0.42077 | -0.33525 | 1.25508 |
| 442 | 0 | -0.75168 | 0.02556 | -0.83624 | 0.22547 | 0.62859 | 0 | 0 |
| 443 | 1 | 0.4583 | 1.26012 | 1.3083 | 2.20084 | 0.5996 | 0.4583 | 1.3083 |
| 444 | 0 | 0.00364 | -2.10438 | 1.21055 | -0.57944 | 0.00441 | 0 | 0 |
| 445 | 0 | -0.4436 | 0.39945 | -0.47919 | -2.83879 | 0.21257 | 0 | 0 |
| 446 | 1 | -0.82196 | 1.36928 | 0.26223 | 0.55304 | -0.21554 | -0.82196 | 0.26223 |
| 447 | 0 | -0.04462 | 1.4247 | 0.12639 | 0.70895 | -0.00564 | 0 | 0 |
| 448 | 0 | -0.41664 | -4.27392 | 0.40484 | -1.00648 | -0.16867 | 0 | 0 |
| 449 | 1 | 1.97899 | 3.73444 | 3.4869 | 7.40408 | 6.90054 | 1.97899 | 3.4869 |
| 450 | 1 | 0.03234 | 1.51634 | 0.24242 | 1.25137 | 0.00784 | 0.03234 | 0.24242 |
| 451 | 0 | 0.89737 | 1.1231 | 0.528 | 1.33278 | 0.47381 | 0 | 0 |
| 452 | 1 | 0.74048 | -1.07531 | 2.3413 | 2.11518 | 1.73369 | 0.74048 | 2.3413 |
| 453 | 1 | 0.70201 | -1.51238 | 1.24396 | 2.59587 | 0.87328 | 0.70201 | 1.24396 |
| 454 | 1 | 0.42267 | -0.53493 | 0.4633 | -0.22176 | 0.19582 | 0.42267 | 0.4633 |
| 455 | 0 | -1.92493 | 1.44861 | 0.31031 | -0.35955 | -0.59733 | 0 | 0 |
| 456 | 1 | 1.34349 | 0.1473 | 1.14033 | 3.55932 | 1.53201 | 1.34349 | 1.14033 |
| 457 | 1 | -0.28922 | -0.96954 | 1.42017 | 1.43996 | -0.41075 | -0.28922 | 1.42017 |
| 458 | 0 | 0.78874 | -0.67557 | -0.37169 | 1.43838 | -0.29317 | 0 | 0 |
| 459 | 0 | 0.1115 | -1.86136 | 0.88028 | -0.61413 | 0.09815 | 0 | 0 |
| 460 | 1 | 0.30001 | -0.20799 | 2.18606 | 5.2141 | 0.65584 | 0.30001 | 2.18606 |
| 461 | 1 | 1.72433 | 0.2877 | 1.70634 | 5.09924 | 2.94229 | 1.72433 | 1.70634 |
| 462 | 0 | -2.16471 | -1.11894 | -0.2232 | -2.08453 | 0.48316 | 0 | 0 |
| 463 | 1 | 1.89403 | 2.17856 | 2.5634 | 6.88474 | 4.85515 | 1.89403 | 2.5634 |
| 464 | 1 | 1.72053 | 3.16869 | 1.17235 | 6.06922 | 2.01706 | 1.72053 | 1.17235 |
| 465 | 1 | 1.85654 | 2.12613 | 2.17732 | 4.28162 | 4.04228 | 1.85654 | 2.17732 |
| 466 | 0 | 1.2044 | -2.47914 | 1.24264 | -0.23917 | 1.49664 | 0 | 0 |
| 467 | 0 | 1.01617 | -2.4169 | -0.96795 | -0.43604 | -0.9836 | 0 | 0 |
| 468 | 0 | 0.05308 | 0.12356 | 0.85705 | -1.28064 | 0.0455 | 0 | 0 |
| 469 | 0 | -2.09901 | -1.16562 | 0.39454 | -2.51179 | -0.82813 | 0 | 0 |
| 470 | 0 | -0.61968 | -0.58454 | 1.05921 | 0.63824 | -0.65638 | 0 | 0 |
| 471 | 1 | -0.25939 | 3.71473 | 2.6401 | 2.99151 | -0.68481 | -0.25939 | 2.6401 |
| 472 | 1 | 1.24379 | 0.96357 | 0.99757 | 2.93233 | 1.24076 | 1.24379 | 0.99757 |
| 473 | 1 | -0.33043 | -0.94558 | 0.16912 | 1.2523 | -0.05588 | -0.33043 | 0.16912 |
| 474 | 0 | -1.02995 | 0.77949 | -1.05767 | 0.07299 | 1.08935 | 0 | 0 |
| 475 | 1 | 0.97075 | -1.52447 | -0.21481 | 1.73915 | -0.20853 | 0.97075 | -0.21481 |
| 476 | 1 | -0.929 | -5.19835 | 0.40632 | -2.87817 | -0.37747 | -0.929 | 0.40632 |
| 477 | 1 | 0.41553 | -0.69977 | 0.44516 | 1.1624 | 0.18498 | 0.41553 | 0.44516 |
| 478 | 1 | -0.16413 | 2.92853 | -0.00012 | 2.38797 | 0.00002 | -0.16413 | -0.00012 |
| 479 | 0 | 0.81104 | -1.92111 | -0.06244 | -1.00953 | -0.05065 | 0 | 0 |
| 480 | 0 | 0.01382 | -1.83682 | -0.92763 | 0.02094 | -0.01282 | 0 | 0 |
| 481 | 1 | -0.42912 | 0.72858 | 0.11636 | -0.88869 | -0.04993 | -0.42912 | 0.11636 |
| 482 | 0 | -0.20677 | 2.23406 | 1.27495 | 0.82776 | -0.26362 | 0 | 0 |
| 483 | 1 | -1.03351 | -1.72509 | 0.74277 | -1.9569 | -0.76766 | -1.03351 | 0.74277 |
| 484 | 1 | -0.31648 | 1.64634 | 0.02331 | -0.62399 | -0.00738 | -0.31648 | 0.02331 |
| 485 | 0 | 0.75279 | 0.04567 | 2.23008 | 1.48234 | 1.67877 | 0 | 0 |
| 486 | 1 | -0.62222 | 0.81232 | -0.00306 | 0.02925 | 0.0019 | -0.62222 | -0.00306 |
| 487 | 1 | 0.55527 | -1.78411 | 1.79868 | 0.17154 | 0.99876 | 0.55527 | 1.79868 |
| 488 | 0 | -0.89006 | 0.45514 | -0.36247 | 0.01698 | 0.32262 | 0 | 0 |
| 489 | 1 | 1.08623 | 0.58761 | 1.22007 | 1.33168 | 1.32528 | 1.08623 | 1.22007 |
| 490 | 0 | -0.32167 | 0.03473 | 0.8335 | -0.58595 | -0.26811 | 0 | 0 |
| 491 | 1 | -1.32067 | -2.33577 | -0.52983 | -1.76718 | 0.69973 | -1.32067 | -0.52983 |
| 492 | 0 | -0.06942 | 0.22021 | 2.15574 | 0.53732 | -0.14965 | 0 | 0 |
| 493 | 0 | 1.27593 | 1.02745 | 1.05497 | 3.91462 | 1.34607 | 0 | 0 |
| 494 | 1 | -1.03038 | -3.65962 | -0.6725 | -4.15547 | 0.69293 | -1.03038 | -0.6725 |
| 495 | 1 | 1.75804 | 0.15889 | 0.90172 | 3.43402 | 1.58526 | 1.75804 | 0.90172 |
| 496 | 0 | 1.4726 | 0.93557 | 1.32912 | 1.23437 | 1.95726 | 0 | 0 |
| 497 | 0 | 0.72812 | 2.16946 | 0.61372 | 1.42607 | 0.44686 | 0 | 0 |
| 498 | 1 | 0.14151 | 0.52612 | 2.63522 | 0.69418 | 0.37291 | 0.14151 | 2.63522 |
| 499 | 0 | 2.27675 | 3.66816 | 1.85515 | 2.81085 | 4.2237 | 0 | 0 |
| 500 | 1 | -0.62414 | 1.42678 | -0.13895 | 0.29835 | 0.08672 | -0.62414 | -0.13895 |

R syntax and output

An R syntax file named *Examplecodemarkdown.R* to replicate these analyses can be directly retrieved from the following Open Science Framework link (https://osf.io/r4e5y/?view\_only=239d81da3d6d4bd09496c5b31139e17d).

## Dataset

Importing example dataset which contains the following variables: X (0 - control, 1 - treatment); M1, Y1 - baseline values of the mediator and outcome - these could also be other baseline covariates; M2 - the focal outcome variable in Example 1 for moderated treatment effects below and the focal mediator in examples 2 and 3 for moderated mediation; Y2 - the focal outcome variable in examples 2 and 3; XM1 - interaction term of baseline M1 with X. This is the interaction term for the a path and the c’ paths in examples 2 and 3; M1M2 - interaction term of baseline and post-treatment M. This is the interaction term for the b path in examples 2 and 3; XM2 - interaction term of post-treatment M2 with X for use in Example 3.

data<-read.csv("C:/suppdata.csv",header=TRUE)
library(splitstackshape)

## Example 1

Example 1: Moderated treatment effect Moderated treatment effect is X -> M2 with XM1 interaction term X is binary treatment variable, M2 is continuous outcome, M1 is continuous moderator.

#Saving means and standard deviations for use in defined subgroups;
meanm1<-mean(data$m1)
sdm1<-sd(data$m1)

med1<-lm(m2~x+m1+xm1,data=data)

datamodel<-data
#Saving the coefficients and rmse;
datamodel$im<-med1$coefficients[1]
datamodel$a1<-med1$coefficients[2]
datamodel$z1<-med1$coefficients[3]
datamodel$a2<-med1$coefficients[4]
rmsem<-sqrt(sum(med1$residuals^2)/med1$df)

#Duplicating the rows of the dataset so each participant gets 1000 simulated values
datamodel = expandRows(datamodel, count=1000, count.is.col=FALSE)
datamodel$em=rnorm(500000,mean=0,sd=rmsem)
datagcomp<-datamodel

#Simulating the potential outcomes for each participant
attach(datagcomp)
datagcomp$M2\_0<-im+a1\*0+z1\*m1+a2\*0\*m1+em
datagcomp$M2\_1<-im+a1\*1+z1\*m1+a2\*1\*m1+em

#Estimating the treatment effect
datagcomp$trt<-datagcomp$M2\_1-datagcomp$M2\_0

#Averaging over the 1000 observations per participant
datagcomp<-aggregate(datagcomp$trt,list(datagcomp$i), FUN=mean)
names(datagcomp)<-c("i", "trt")
datagcomp<-cbind(datagcomp,data)

#Subsetting the dataset to sort the observations around the cutpoints of +1/-1SD above/below the mean of m1
sda <- datagcomp[which(datagcomp$m1 >= (sdm1+meanm1)),]
sdb <- datagcomp[which(datagcomp$m1 <= (meanm1-sdm1)),]
mid <- datagcomp[which(datagcomp$m1 < (meanm1+sdm1) & datagcomp$m1 > (meanm1-sdm1)),]

#Saving the point estimates of the average treatment effect (trt)
#treatment effect for subgroup who scored 1 SD below the mean of m1 or lower (sdbtrt)
#treatment effect for subgroup who scored 1 SD above the mean of m1 or higher (sdatrt)
#treatment effect for subgroup who scored between 1 SD below and 1 SD above the mean of m1 or lower (midtrt)
trt<-mean(datagcomp$trt)
sdbtrt<-mean(sdb$trt)
nsdb<-nrow(sdb)
sdatrt<-mean(sda$trt)
nsda<-nrow(sda)
midtrt<-mean(mid$trt)
nmid<-nrow(mid)

detach(datagcomp)

Bootstrap procedure begins here. The above code is repeated for B bootstrap samples.

#Begin bootstrap procedure
#set.seed(2021)
trtboot = NULL
sdbtrtboot = NULL
nsdbboot= NULL
sdatrtboot = NULL
nsdaboot= NULL
midtrtboot = NULL
nmidboot= NULL
n = nrow(data); B = 1000
for(draw in 1:B){
 # Randomly sample from the rows of dataset, with replacement
 sim = data[sample(1:n,size=n,replace=T),]
 sim$person<-seq(1,n)

 meanm1rep<-mean(sim$m1)
 sdm1rep<-sd(sim$m1)

 modelrep<-sim
 med1rep<-lm(m2~x+m1+xm1,data=modelrep)

 #Saving the coefficients and rmse;
 modelrep$imrep<-med1rep$coefficients[1]
 modelrep$a1rep<-med1rep$coefficients[2]
 modelrep$z1rep<-med1rep$coefficients[3]
 modelrep$a2rep<-med1rep$coefficients[4]
 rmsemrep<-sqrt(sum(med1rep$residuals^2)/med1rep$df)

 #Duplicating the rows of the dataset
 datagmodelrep <- expandRows(modelrep, count=1000, count.is.col=FALSE)
 datagmodelrep$emrep<-rnorm(500000,mean=0,sd=rmsemrep)
 datagcomprep<-datagmodelrep

 attach(datagcomprep)
 datagcomprep$M2\_0rep<-imrep+a1rep\*0+z1rep\*m1+a2rep\*0\*m1+emrep
 datagcomprep$M2\_1rep<-imrep+a1rep\*1+z1rep\*m1+a2rep\*1\*m1+emrep

 datagcomprep$trtrep<-datagcomprep$M2\_1rep-datagcomprep$M2\_0rep

 datagcomprep<-aggregate(datagcomprep$trtrep,list(datagcomprep$person), FUN=mean)
 names(datagcomprep)<-c("person", "trtrep")
 datagcomprep<-cbind(sim, datagcomprep)

 sdarep <- datagcomprep[which(datagcomprep$m1 >= (meanm1rep+sdm1rep)),]
 sdbrep <- datagcomprep[which(datagcomprep$m1 <= (meanm1rep-sdm1rep)),]
 midrep <- datagcomprep[which(datagcomprep$m1 < (meanm1rep+sdm1rep)
 & datagcomprep$m1 > (meanm1rep-sdm1rep)),]

 trtrep<-mean(datagcomprep$trtrep)
 sdbtrtrep<-mean(sdbrep$trtrep)
 nsdbrep<-nrow(sdbrep)
 sdatrtrep<-mean(sdarep$trtrep)
 nsdarep<-nrow(sdarep)
 midtrtrep<-mean(midrep$trtrep)
 nmidrep<-nrow(midrep)

 trtboot=rbind(trtboot, trtrep)
 sdbtrtboot=rbind(sdbtrtboot, sdbtrtrep)
 nsdbboot=rbind(nsdbboot, nsdbrep)
 sdatrtboot=rbind(sdatrtboot, sdatrtrep)
 nsdaboot=rbind(nsdaboot, nsdarep)
 midtrtboot=rbind(midtrtboot, midtrtrep)
 nmidboot=rbind(nmidboot, nmidrep)

 detach(datagcomprep)

}
#Prints point estimates and 95% CIs;
trt

## [1] 0.5049117

quantile(trtboot,c(0.025,0.975))

## 2.5% 97.5%
## 0.3154639 0.6868639

sdbtrt

## [1] -0.08708206

quantile(sdbtrtboot,c(0.025,0.975))

## 2.5% 97.5%
## -0.4050194 0.2404680

sdatrt

## [1] 1.094372

quantile(sdatrtboot,c(0.025,0.975))

## 2.5% 97.5%
## 0.7895882 1.4318249

midtrt

## [1] 0.5037737

quantile(midtrtboot,c(0.025,0.975))

## 2.5% 97.5%
## 0.3206141 0.6915081

## Example 2

Moderated Mediation with a-path and b-path moderated by baseline covariate M1; The a path is X->M2 and is moderated by baseline M1, the c’ path is X->Y2 and is moderated by baseline M1, the b path is M2->Y2 and is moderated by baseline M1. There is an additional baseline covariate, Y1. This covariate can be excluded from the model if the model is not a two-wave mediation model.

#Saving means and standard deviations for use in defined subgroups;
meanm1<-mean(data$m1)
sdm1<-sd(data$m1)

med1<-lm(m2~x+m1+y1+xm1,data=data)
out1<-lm(y2~x+m2+m1+y1+xm1+m1m2,data=data)

datamodel<-data
#Saving the coefficients and rmse;
datamodel$im<-med1$coefficients[1]
datamodel$a1<-med1$coefficients[2]
datamodel$z1<-med1$coefficients[3]
datamodel$z2<-med1$coefficients[4]
datamodel$a2<-med1$coefficients[5]
rmsem<-sqrt(sum(med1$residuals^2)/med1$df)

datamodel$iy<-out1$coefficients[1]
datamodel$cp1<-out1$coefficients[2]
datamodel$b1<-out1$coefficients[3]
datamodel$z3<-out1$coefficients[4]
datamodel$z4<-out1$coefficients[5]
datamodel$cp2<-out1$coefficients[6]
datamodel$b2<-out1$coefficients[7]
rmsey<-sqrt(sum(out1$residuals^2)/out1$df)

#Duplicating the rows of the dataset so each participant gets 1000 simulated values
datamodel = expandRows(datamodel, count=1000, count.is.col=FALSE)

#The residuals are simulated from a normal distribution with standard deviations equal to
#the estimates root mean-squared error from the regression models above;
datamodel$em=rnorm(500000,mean=0,sd=rmsem)
datamodel$ey=rnorm(500000,mean=0,sd=rmsey)

datagcomp<-datamodel

#Simulating the potential outcomes for each participant
attach(datagcomp)
#M2\_0 is the predicted value of the mediator at time 2 holding the treatment fixed at X=0
datagcomp$M2\_0<-im+a1\*0+z1\*m1+z2\*y1+a2\*0\*m1+em
#M2\_1 is the predicted value of the mediator at time 2 holding the treatment fixed at X=1
datagcomp$M2\_1<-im+a1\*1+z1\*m1+z2\*y1+a2\*1\*m1+em
detach(datagcomp)

attach(datagcomp)
#Y2\_0\_M2\_0 is the predicted value of the outcome at time 2 holding the treatment fixed at X=0
#and holding the mediator constant at the individual's predicted value of M2 holding the treatment fixed at X=0
datagcomp$Y2\_0\_M2\_0<-iy+cp1\*0+b1\*M2\_0+z3\*m1+z4\*y1+cp2\*0\*m1+b2\*m1\*M2\_0+ey

#Y2\_0\_M2\_1 is the predicted value of the outcome at time 2 holding the treatment fixed at X=0
#and holding the mediator constant at the individual's predicted value of M2 holding the treatment fixed at X=1;
datagcomp$Y2\_0\_M2\_1<-iy+cp1\*0+b1\*M2\_1+z3\*m1+z4\*y1+cp2\*0\*m1+b2\*m1\*M2\_1+ey

#Estimating the mediated effect
datagcomp$nie<-datagcomp$Y2\_0\_M2\_1-datagcomp$Y2\_0\_M2\_0

#Averaging over the 1000 observations per participant
datagcomp<-aggregate(datagcomp$nie,list(datagcomp$i), FUN=mean)
names(datagcomp)<-c("i", "nie")
datagcomp<-cbind(datagcomp,data)

#Subsetting the dataset to sort the observations around the cutpoints of +1/-1SD above/below the mean of m1
sda <- datagcomp[which(datagcomp$m1 >= (sdm1+meanm1)),]
sdb <- datagcomp[which(datagcomp$m1 <= (meanm1-sdm1)),]
mid <- datagcomp[which(datagcomp$m1 < (meanm1+sdm1) & datagcomp$m1 > (meanm1-sdm1)),]

#Saving the point estimates of the average mediated effect (nie)
#mediated effect for subgroup who scored 1 SD below the mean of m1 or lower (sdniet)
#mediated effect for subgroup who scored 1 SD above the mean of m1 or higher (sdanie)
#mediated effect for subgroup who scored between 1 SD below and 1 SD above the mean of m1 or lower (midnie)
nie<-mean(datagcomp$nie)
sdbnie<-mean(sdb$nie)
nsdb<-nrow(sdb)
sdanie<-mean(sda$nie)
nsda<-nrow(sda)
midnie<-mean(mid$nie)
nmid<-nrow(mid)

detach(datagcomp)

Bootstrap procedure begins here. The above code is repeated for B bootstrap samples.

#Begin bootstrap procedure
#set.seed(2021)
nieboot = NULL
sdbnieboot = NULL
nsdbboot= NULL
sdanieboot = NULL
nsdaboot= NULL
midnieboot = NULL
nmidboot= NULL
n = nrow(data); B = 1000
for(draw in 1:B){
 # Randomly sample from the rows of dataset, with replacement
 sim = data[sample(1:n,size=n,replace=T),]
 sim$person<-seq(1,n)

 meanm1rep<-mean(sim$m1)
 sdm1rep<-sd(sim$m1)

 modelrep<-sim
 med1rep<-lm(m2~x+m1+y1+xm1,data=modelrep)
 out1rep<-lm(y2~x+m2+m1+y1+xm1+m1m2,data=data)

 #Saving the coefficients and rmse;
 modelrep$imrep<-med1rep$coefficients[1]
 modelrep$a1rep<-med1rep$coefficients[2]
 modelrep$z1rep<-med1rep$coefficients[3]
 modelrep$z2rep<-med1rep$coefficients[4]
 modelrep$a2rep<-med1rep$coefficients[5]
 rmsemrep<-sqrt(sum(med1rep$residuals^2)/med1rep$df)

 modelrep$iyrep<-out1rep$coefficients[1]
 modelrep$cp1rep<-out1rep$coefficients[2]
 modelrep$b1rep<-out1rep$coefficients[3]
 modelrep$z3rep<-out1rep$coefficients[4]
 modelrep$z4rep<-out1rep$coefficients[5]
 modelrep$cp2rep<-out1rep$coefficients[6]
 modelrep$b2rep<-out1rep$coefficients[7]
 rmseyrep<-sqrt(sum(out1rep$residuals^2)/out1rep$df)

 #Duplicating the rows of the dataset
 datagmodelrep <- expandRows(modelrep, count=1000, count.is.col=FALSE)
 datagmodelrep$emrep<-rnorm(500000,mean=0,sd=rmsemrep)
 datagmodelrep$eyrep<-rnorm(500000,mean=0,sd=rmseyrep)
 datagcomprep<-datagmodelrep

 attach(datagcomprep)
 datagcomprep$M2\_0rep<-imrep+a1rep\*0+z1rep\*m1+z2rep\*y1+a2rep\*0\*m1+emrep
 datagcomprep$M2\_1rep<-imrep+a1rep\*1+z1rep\*m1+z2rep\*y1+a2rep\*1\*m1+emrep
 detach(datagcomprep)

 attach(datagcomprep)
 datagcomprep$Y2\_0\_M2\_0rep<-iyrep+cp1rep\*0+b1rep\*M2\_0rep+z3rep\*m1+z4rep\*y1+cp2rep\*0\*m1+b2rep\*m1\*M2\_0rep+eyrep
 datagcomprep$Y2\_0\_M2\_1rep<-iyrep+cp1rep\*0+b1rep\*M2\_1rep+z3rep\*m1+z4rep\*y1+cp2rep\*0\*m1+b2rep\*m1\*M2\_1rep+eyrep

 #Estimating the mediated effect
 datagcomprep$nierep<-datagcomprep$Y2\_0\_M2\_1rep-datagcomprep$Y2\_0\_M2\_0rep

 datagcomprep<-aggregate(datagcomprep$nierep,list(datagcomprep$person), FUN=mean)
 names(datagcomprep)<-c("person", "nierep")
 datagcomprep<-cbind(sim, datagcomprep)

 sdarep <- datagcomprep[which(datagcomprep$m1 >= (meanm1rep+sdm1rep)),]
 sdbrep <- datagcomprep[which(datagcomprep$m1 <= (meanm1rep-sdm1rep)),]
 midrep <- datagcomprep[which(datagcomprep$m1 < (meanm1rep+sdm1rep)
 & datagcomprep$m1 > (meanm1rep-sdm1rep)),]

 nierep<-mean(datagcomprep$nierep)
 sdbnierep<-mean(sdbrep$nierep)
 nsdbrep<-nrow(sdbrep)
 sdanierep<-mean(sdarep$nierep)
 nsdarep<-nrow(sdarep)
 midnierep<-mean(midrep$nierep)
 nmidrep<-nrow(midrep)

 nieboot=rbind(nieboot, nierep)
 sdbnieboot=rbind(sdbnieboot, sdbnierep)
 nsdbboot=rbind(nsdbboot, nsdbrep)
 sdanieboot=rbind(sdanieboot, sdanierep)
 nsdaboot=rbind(nsdaboot, nsdarep)
 midnieboot=rbind(midnieboot, midnierep)
 nmidboot=rbind(nmidboot, nmidrep)

 detach(datagcomprep)

}
#Prints point estimates and 95% CIs;
nie

## [1] 0.3389093

quantile(nieboot,c(0.025,0.975))

## 2.5% 97.5%
## 0.2285651 0.4465614

sdbnie

## [1] -0.01676138

quantile(sdbnieboot,c(0.025,0.975))

## 2.5% 97.5%
## -0.11254344 0.08235438

sdanie

## [1] 0.8662655

quantile(sdanieboot,c(0.025,0.975))

## 2.5% 97.5%
## 0.6047281 1.1115969

midnie

## [1] 0.2937464

quantile(midnieboot,c(0.025,0.975))

## 2.5% 97.5%
## 0.1888427 0.3996969

## Example 3

Moderated Causal Mediation effects: PNIE, TNIE, PNDE, TNDE, CDE, and TE; The a path is X->M2 and is moderated by baseline M1, the c’ path is X->Y2 and is moderated by baseline M1, the b path is M2->Y2 and is moderated by baseline M1. There is a treatment-mediator interaction XM2 which results in different mediated effects, PNIE, TNIE, and different direct effects, PNDE, TNDE, and CDE. There is an additional baseline covariate, Y1. This covariate can be excluded from the model if the model is not a two-wave mediation model.

#Saving means and standard deviations for use in defined subgroups;
meanm1<-mean(data$m1)
sdm1<-sd(data$m1)

med1<-lm(m2~x+m1+y1+xm1,data=data)
out1<-lm(y2~x+m2+m1+y1+xm1+m1m2+xm2,data=data)

datamodel<-data
#Saving the coefficients and rmse;
datamodel$im<-med1$coefficients[1]
datamodel$a1<-med1$coefficients[2]
datamodel$z1<-med1$coefficients[3]
datamodel$z2<-med1$coefficients[4]
datamodel$a2<-med1$coefficients[5]
rmsem<-sqrt(sum(med1$residuals^2)/med1$df)

datamodel$iy<-out1$coefficients[1]
datamodel$cp1<-out1$coefficients[2]
datamodel$b1<-out1$coefficients[3]
datamodel$z3<-out1$coefficients[4]
datamodel$z4<-out1$coefficients[5]
datamodel$cp2<-out1$coefficients[6]
datamodel$b2<-out1$coefficients[7]
datamodel$h<-out1$coefficients[8]
rmsey<-sqrt(sum(out1$residuals^2)/out1$df)

#Duplicating the rows of the dataset so each participant gets 1000 simulated values
datamodel = expandRows(datamodel, count=1000, count.is.col=FALSE)

#The residuals are simulated from a normal distribution with standard deviations equal to
#the estimates root mean-squared error from the regression models above;
datamodel$em=rnorm(500000,mean=0,sd=rmsem)
datamodel$ey=rnorm(500000,mean=0,sd=rmsey)

datagcomp<-datamodel

#Simulating the potential outcomes for each participant
attach(datagcomp)
#M2\_0 is the predicted value of the mediator at time 2 holding the treatment fixed at X=0
datagcomp$M2\_0<-im+a1\*0+z1\*m1+z2\*y1+a2\*0\*m1+em
#M2\_1 is the predicted value of the mediator at time 2 holding the treatment fixed at X=1
datagcomp$M2\_1<-im+a1\*1+z1\*m1+z2\*y1+a2\*1\*m1+em
detach(datagcomp)

attach(datagcomp)
#Y2\_0\_M2\_0 is the predicted value of the outcome at time 2 holding the treatment fixed at X=0
#and holding the mediator constant at the individual's predicted value of M2 holding the treatment fixed at X=0
datagcomp$Y2\_0\_M2\_0<-iy+cp1\*0+b1\*M2\_0+z1\*m1+z2\*y1+cp2\*0\*m1+b2\*m1\*M2\_0+h\*0\*M2\_0+ey

#Y2\_1\_M2\_0 is the predicted value of the outcome at time 2 holding the treatment fixed at X=1
#and holding the mediator constant at the individual's predicted value of M2 holding the treatment fixed at X=0;
datagcomp$Y2\_1\_M2\_0<-iy+cp1\*1+b1\*M2\_0+z3\*m1+z3\*y1+cp2\*1\*m1+b2\*m1\*M2\_0+h\*1\*M2\_0+ey

#Y2\_0\_M2\_1 is the predicted value of the outcome at time 2 holding the treatment fixed at X=0
#and holding the mediator constant at the individual's predicted value of M2 holding the treatment fixed at X=1;
datagcomp$Y2\_0\_M2\_1<-iy+cp1\*0+b1\*M2\_1+z3\*m1+z4\*y1+cp2\*0\*m1+b2\*m1\*M2\_1+h\*0\*M2\_1+ey

#Y2\_1\_M2\_1 is the predicted value of the outcome at time 2 holding the treatment fixed at X=1
#and holding the mediator constant at the individual's predicted value of M2 holding the treatment fixed at X=1;
datagcomp$Y2\_1\_M2\_1<-iy+cp1\*1+b1\*M2\_1+z3\*m1+z4\*y1+cp2\*1\*m1+b2\*m1\*M2\_1+h\*1\*M2\_1+ey

#Value of the mediator variable at time 2 that the CDE is estimated;
m=meanm1
#Y2\_0\_m is the predicted value of the outcome at time 2 holding the treatment fixed at X=0
#and holding the mediator constant at a fixed value for all individuals;
datagcomp$Y2\_0\_m<-iy+cp1\*0+b1\*m+z3\*m1+z4\*y1+cp2\*0\*m1+b2\*m1\*m+h\*0\*m+ey

#Y2\_1\_m is the predicted value of the outcome at time 2 holding the treatment fixed at X=1
#and holding the mediator constant at a fixed value for all individuals;
datagcomp$Y2\_1\_m<-iy+cp1\*1+b1\*m+z3\*m1+z4\*y1+cp2\*1\*m1+b2\*m1\*m+h\*1\*m+ey

#Estimating all causal mediation effects
datagcomp$pnie<-datagcomp$Y2\_0\_M2\_1-datagcomp$Y2\_0\_M2\_0
datagcomp$tnie<-datagcomp$Y2\_1\_M2\_1-datagcomp$Y2\_1\_M2\_0
datagcomp$pnde<-datagcomp$Y2\_1\_M2\_0-datagcomp$Y2\_0\_M2\_0
datagcomp$tnde<-datagcomp$Y2\_1\_M2\_1-datagcomp$Y2\_0\_M2\_1
datagcomp$cde<-datagcomp$Y2\_1\_m-datagcomp$Y2\_0\_m
datagcomp$te<-datagcomp$pnie+datagcomp$tnde

#Averaging over the 1000 observations per participant
datagcomp<-aggregate(cbind(datagcomp$pnie,datagcomp$tnie,datagcomp$pnde,datagcomp$tnde,datagcomp$cde,datagcomp$te),
 list(datagcomp$i), FUN=mean)
names(datagcomp)<-c("i", "pnie","tnie","pnde","tnde","cde","te")
datagcomp<-cbind(datagcomp,data)

#Subsetting the dataset to sort the observations around the cutpoints of +1/-1SD above/below the mean of m1
sda <- datagcomp[which(datagcomp$m1 >= (sdm1+meanm1)),]
sdb <- datagcomp[which(datagcomp$m1 <= (meanm1-sdm1)),]
mid <- datagcomp[which(datagcomp$m1 < (meanm1+sdm1) & datagcomp$m1 > (meanm1-sdm1)),]

#Saving the point estimates of the average mediated effect (nie)
#mediated effect for subgroup who scored 1 SD below the mean of m1 or lower (sdniet)
#mediated effect for subgroup who scored 1 SD above the mean of m1 or higher (sdanie)
#mediated effect for subgroup who scored between 1 SD below and 1 SD above the mean of m1 or lower (midnie)
nsdb<-nrow(sdb)
nsda<-nrow(sda)
nmid<-nrow(mid)

pnie<-mean(datagcomp$pnie)
sdbpnie<-mean(sdb$pnie)
sdapnie<-mean(sda$pnie)
midpnie<-mean(mid$pnie)
tnie<-mean(datagcomp$tnie)
sdbtnie<-mean(sdb$tnie)
sdatnie<-mean(sda$tnie)
midtnie<-mean(mid$tnie)
pnde<-mean(datagcomp$pnde)
sdbpnde<-mean(sdb$pnde)
sdapnde<-mean(sda$pnde)
midpnde<-mean(mid$pnde)
tnde<-mean(datagcomp$tnde)
sdbtnde<-mean(sdb$tnde)
sdatnde<-mean(sda$tnde)
midtnde<-mean(mid$tnde)
cde<-mean(datagcomp$cde)
sdbcde<-mean(sdb$cde)
sdacde<-mean(sda$cde)
midcde<-mean(mid$cde)
te<-mean(datagcomp$te)
sdbte<-mean(sdb$te)
sdate<-mean(sda$te)
midte<-mean(mid$te)

detach(datagcomp)

Bootstrap procedure begins here. The above code is repeated for B bootstrap samples.

#Begin bootstrap procedure
#set.seed(2021)
pnieboot = NULL
sdbpnieboot = NULL
sdapnieboot = NULL
midpnieboot = NULL
tnieboot = NULL
sdbtnieboot = NULL
sdatnieboot = NULL
midtnieboot = NULL
pndeboot = NULL
sdbpndeboot = NULL
sdapndeboot = NULL
midpndeboot = NULL
tndeboot = NULL
sdbtndeboot = NULL
sdatndeboot = NULL
midtndeboot = NULL
cdeboot = NULL
sdbcdeboot = NULL
sdacdeboot = NULL
midcdeboot = NULL
teboot = NULL
sdbteboot = NULL
sdateboot = NULL
midteboot = NULL
nsdbboot= NULL
nsdaboot= NULL
nmidboot= NULL
n = nrow(data); B = 1000
for(draw in 1:B){
 # Randomly sample from the rows of dataset, with replacement
 sim = data[sample(1:n,size=n,replace=T),]
 sim$person<-seq(1,n)

 meanm1rep<-mean(sim$m1)
 sdm1rep<-sd(sim$m1)

 modelrep<-sim
 med1rep<-lm(m2~x+m1+y1+xm1,data=modelrep)
 out1rep<-lm(y2~x+m2+m1+y1+xm1+m1m2+xm2,data=data)

 #Saving the coefficients and rmse;
 modelrep$imrep<-med1rep$coefficients[1]
 modelrep$a1rep<-med1rep$coefficients[2]
 modelrep$z1rep<-med1rep$coefficients[3]
 modelrep$z2rep<-med1rep$coefficients[4]
 modelrep$a2rep<-med1rep$coefficients[5]
 rmsemrep<-sqrt(sum(med1rep$residuals^2)/med1rep$df)

 modelrep$iyrep<-out1rep$coefficients[1]
 modelrep$cp1rep<-out1rep$coefficients[2]
 modelrep$b1rep<-out1rep$coefficients[3]
 modelrep$z3rep<-out1rep$coefficients[4]
 modelrep$z4rep<-out1rep$coefficients[5]
 modelrep$cp2rep<-out1rep$coefficients[6]
 modelrep$b2rep<-out1rep$coefficients[7]
 modelrep$hrep<-out1rep$coefficients[8]
 rmseyrep<-sqrt(sum(out1rep$residuals^2)/out1rep$df)

 #Duplicating the rows of the dataset
 datagmodelrep <- expandRows(modelrep, count=1000, count.is.col=FALSE)
 datagmodelrep$emrep<-rnorm(500000,mean=0,sd=rmsemrep)
 datagmodelrep$eyrep<-rnorm(500000,mean=0,sd=rmseyrep)
 datagcomprep<-datagmodelrep

 attach(datagcomprep)
 datagcomprep$M2\_0rep<-imrep+a1rep\*0+z1rep\*m1+z2rep\*y1+a2rep\*0\*m1+emrep
 datagcomprep$M2\_1rep<-imrep+a1rep\*1+z1rep\*m1+z2rep\*y1+a2rep\*1\*m1+emrep
 detach(datagcomprep)

 attach(datagcomprep)
 datagcomprep$Y2\_0\_M2\_0rep<-iyrep+cp1rep\*0+b1rep\*M2\_0rep+z3rep\*m1+z4rep\*y1+cp2rep\*0\*m1+b2rep\*m1\*M2\_0rep+hrep\*0\*M2\_0rep+eyrep
 datagcomprep$Y2\_1\_M2\_0rep<-iyrep+cp1rep\*1+b1rep\*M2\_0rep+z3rep\*m1+z4rep\*y1+cp2rep\*1\*m1+b2rep\*m1\*M2\_0rep+hrep\*1\*M2\_0rep+eyrep
 datagcomprep$Y2\_0\_M2\_1rep<-iyrep+cp1rep\*0+b1rep\*M2\_1rep+z3rep\*m1+z4rep\*y1+cp2rep\*0\*m1+b2rep\*m1\*M2\_1rep+hrep\*0\*M2\_1rep+eyrep
 datagcomprep$Y2\_1\_M2\_1rep<-iyrep+cp1rep\*1+b1rep\*M2\_1rep+z3rep\*m1+z4rep\*y1+cp2rep\*1\*m1+b2rep\*m1\*M2\_1rep+hrep\*1\*M2\_1rep+eyrep

 mrep=meanm1rep
 datagcomprep$Y2\_0\_mrep<-iyrep+cp1rep\*0+b1rep\*mrep+z3rep\*m1+z4rep\*y1+cp2rep\*0\*m1+b2rep\*m1\*mrep+hrep\*0\*mrep+eyrep
 datagcomprep$Y2\_1\_mrep<-iyrep+cp1rep\*1+b1rep\*mrep+z3rep\*m1+z4rep\*y1+cp2rep\*1\*m1+b2rep\*m1\*mrep+hrep\*1\*mrep+eyrep

 #Estimating the causal mediation effects
 datagcomprep$pnierep<-datagcomprep$Y2\_0\_M2\_1rep-datagcomprep$Y2\_0\_M2\_0rep
 datagcomprep$tnierep<-datagcomprep$Y2\_1\_M2\_1rep-datagcomprep$Y2\_1\_M2\_0rep
 datagcomprep$pnderep<-datagcomprep$Y2\_1\_M2\_0rep-datagcomprep$Y2\_0\_M2\_0rep
 datagcomprep$tnderep<-datagcomprep$Y2\_1\_M2\_1rep-datagcomprep$Y2\_0\_M2\_1rep
 datagcomprep$cderep<-datagcomprep$Y2\_1\_mrep-datagcomprep$Y2\_0\_mrep
 datagcomprep$terep<-datagcomprep$pnierep+datagcomprep$tnderep

 datagcomprep<-aggregate(cbind(datagcomprep$pnierep,datagcomprep$tnierep,datagcomprep$pnderep,
 datagcomprep$tnderep,datagcomprep$cderep,datagcomprep$terep),
 list(datagcomprep$person), FUN=mean)
 names(datagcomprep)<-c("person","pnierep","tnierep","pnderep","tnderep","cderep","terep")
 datagcomprep<-cbind(sim,datagcomprep)

 sdarep <- datagcomprep[which(datagcomprep$m1 >= (meanm1rep+sdm1rep)),]
 sdbrep <- datagcomprep[which(datagcomprep$m1 <= (meanm1rep-sdm1rep)),]
 midrep <- datagcomprep[which(datagcomprep$m1 < (meanm1rep+sdm1rep)
 & datagcomprep$m1 > (meanm1rep-sdm1rep)),]

 nsdbrep<-nrow(sdbrep)
 nsdarep<-nrow(sdarep)
 nmidrep<-nrow(midrep)
 pnierep<-mean(datagcomprep$pnierep)
 sdbpnierep<-mean(sdbrep$pnierep)
 sdapnierep<-mean(sdarep$pnierep)
 midpnierep<-mean(midrep$pnierep)
 tnierep<-mean(datagcomprep$tnierep)
 sdbtnierep<-mean(sdbrep$tnierep)
 sdatnierep<-mean(sdarep$tnierep)
 midtnierep<-mean(midrep$tnierep)
 pnderep<-mean(datagcomprep$pnderep)
 sdbpnderep<-mean(sdbrep$pnderep)
 sdapnderep<-mean(sdarep$pnderep)
 midpnderep<-mean(midrep$pnderep)
 tnderep<-mean(datagcomprep$tnderep)
 sdbtnderep<-mean(sdbrep$tnderep)
 sdatnderep<-mean(sdarep$tnderep)
 midtnderep<-mean(midrep$tnderep)
 cderep<-mean(datagcomprep$cderep)
 sdbcderep<-mean(sdbrep$cderep)
 sdacderep<-mean(sdarep$cderep)
 midcderep<-mean(midrep$cderep)
 terep<-mean(datagcomprep$terep)
 sdbterep<-mean(sdbrep$terep)
 sdaterep<-mean(sdarep$terep)
 midterep<-mean(midrep$terep)

 nsdbboot=rbind(nsdbboot, nsdbrep)
 nsdaboot=rbind(nsdaboot, nsdarep)
 nmidboot=rbind(nmidboot, nmidrep)

 pnieboot=rbind(pnieboot, pnierep)
 sdbpnieboot=rbind(sdbpnieboot, sdbpnierep)
 sdapnieboot=rbind(sdapnieboot, sdapnierep)
 midpnieboot=rbind(midpnieboot, midpnierep)

 tnieboot=rbind(tnieboot, tnierep)
 sdbtnieboot=rbind(sdbtnieboot, sdbtnierep)
 sdatnieboot=rbind(sdatnieboot, sdatnierep)
 midtnieboot=rbind(midtnieboot, midtnierep)

 pndeboot=rbind(pndeboot, pnderep)
 sdbpndeboot=rbind(sdbpndeboot, sdbpnderep)
 sdapndeboot=rbind(sdapndeboot, sdapnderep)
 midpndeboot=rbind(midpndeboot, midpnderep)

 tndeboot=rbind(tndeboot, tnderep)
 sdbtndeboot=rbind(sdbtndeboot, sdbtnderep)
 sdatndeboot=rbind(sdatndeboot, sdatnderep)
 midtndeboot=rbind(midtndeboot, midtnderep)

 cdeboot=rbind(cdeboot, cderep)
 sdbcdeboot=rbind(sdbcdeboot, sdbcderep)
 sdacdeboot=rbind(sdacdeboot, sdacderep)
 midcdeboot=rbind(midcdeboot, midcderep)

 teboot=rbind(teboot, terep)
 sdbteboot=rbind(sdbteboot, sdbterep)
 sdateboot=rbind(sdateboot, sdaterep)
 midteboot=rbind(midteboot, midterep)

 detach(datagcomprep)

}
#Prints point estimates and 95% CIs;
pnie

## [1] 0.2585909

quantile(pnieboot,c(0.025,0.975))

## 2.5% 97.5%
## 0.1613586 0.3054813

sdbpnie

## [1] -0.09519359

quantile(sdbpnieboot,c(0.025,0.975))

## 2.5% 97.5%
## -0.04338132 0.04055563

sdapnie

## [1] 1.009634

quantile(sdapnieboot,c(0.025,0.975))

## 2.5% 97.5%
## 0.4376189 0.8198541

midpnie

## [1] 0.1555071

quantile(midpnieboot,c(0.025,0.975))

## 2.5% 97.5%
## 0.1273242 0.2643633

tnie

## [1] 0.4357596

quantile(tnieboot,c(0.025,0.975))

## 2.5% 97.5%
## 0.2954306 0.5824107

sdbtnie

## [1] -0.04092471

quantile(sdbtnieboot,c(0.025,0.975))

## 2.5% 97.5%
## -0.1984791 0.1440392

sdatnie

## [1] 1.098056

quantile(sdatnieboot,c(0.025,0.975))

## 2.5% 97.5%
## 0.7488151 1.3992449

midtnie

## [1] 0.3866547

quantile(midtnieboot,c(0.025,0.975))

## 2.5% 97.5%
## 0.2575046 0.5458744

pnde

## [1] 0.4524605

quantile(pndeboot,c(0.025,0.975))

## 2.5% 97.5%
## 0.3437048 0.5077040

sdbpnde

## [1] -0.7733283

quantile(sdbpndeboot,c(0.025,0.975))

## 2.5% 97.5%
## -0.8154522 -0.5317762

sdapnde

## [1] 1.880911

quantile(sdapndeboot,c(0.025,0.975))

## 2.5% 97.5%
## 1.389567 1.656812

midpnde

## [1] 0.3967142

quantile(midpndeboot,c(0.025,0.975))

## 2.5% 97.5%
## 0.3256037 0.5263678

tnde

## [1] 0.6296291

quantile(tndeboot,c(0.025,0.975))

## 2.5% 97.5%
## 0.5308967 0.7187823

sdbtnde

## [1] -0.7190594

quantile(sdbtndeboot,c(0.025,0.975))

## 2.5% 97.5%
## -0.8510278 -0.5645019

sdatnde

## [1] 1.969333

quantile(sdatndeboot,c(0.025,0.975))

## 2.5% 97.5%
## 1.818089 2.124925

midtnde

## [1] 0.6278618

quantile(midtndeboot,c(0.025,0.975))

## 2.5% 97.5%
## 0.5167987 0.7455716

cde

## [1] 0.3993394

quantile(cdeboot,c(0.025,0.975))

## 2.5% 97.5%
## 0.3068830 0.4879941

sdbcde

## [1] -0.5321362

quantile(sdbcdeboot,c(0.025,0.975))

## 2.5% 97.5%
## -0.6235503 -0.4204078

sdacde

## [1] 1.326829

quantile(sdacdeboot,c(0.025,0.975))

## 2.5% 97.5%
## 1.21645 1.43037

midcde

## [1] 0.3975488

quantile(midcdeboot,c(0.025,0.975))

## 2.5% 97.5%
## 0.2929812 0.4991821

te

## [1] 0.8882201

quantile(teboot,c(0.025,0.975))

## 2.5% 97.5%
## 0.7187039 1.0117636

sdbte

## [1] -0.814253

quantile(sdbteboot,c(0.025,0.975))

## 2.5% 97.5%
## -0.8690131 -0.5412757

sdate

## [1] 2.978967

quantile(sdateboot,c(0.025,0.975))

## 2.5% 97.5%
## 2.290606 2.891790

midte

## [1] 0.7833689

quantile(midteboot,c(0.025,0.975))

## 2.5% 97.5%
## 0.6647610 0.9882731

SAS syntax and output

A SAS syntax file named *Examplecode.sas* to replicate these analyses can be directly retrieved from the following Open Science Framework link (https://osf.io/r4e5y/?view\_only=239d81da3d6d4bd09496c5b31139e17d).

## Dataset

Importing example dataset which contains the following variables: X (0 - control, 1 - treatment); M1, Y1 - baseline values of the mediator and outcome - these could also be other baseline covariates; M2 - the focal outcome variable in Example 1 for moderated treatment effects below and the focal mediator in examples 2 and 3 for moderated mediation; Y2 - the focal outcome variable in examples 2 and 3; XM1 - interaction term of baseline M1 with X. This is the interaction term for the a path and the c’ paths in examples 2 and 3; M1M2 - interaction term of baseline and post-treatment M. This is the interaction term for the b path in examples 2 and 3; XM2 - interaction term of post-treatment M2 with X for use in Example 3.

PROC IMPORT OUT= WORK.Full
 DATAFILE= "C:\SuppData.csv"
 DBMS=CSV REPLACE;
 GETNAMES=YES;
 DATAROW=2;
RUN;

Filename NULLOG DUMMY ‘C:\NULL’;

PROC PRINTTO LOG=NULLOG;

## Example 1

Example 1: Moderated treatment effect. Moderated treatment effect is X -> M2 with XM1 interaction term. X is binary treatment variable, M2 is continuous outcome, M1 is continuous moderator

\*Saving means and standard deviations for use in defined subgroups;
proc means data=full noprint;
var m1 m2 y1;
output out=mmeans mean=meanm1 meanm2 meany1 std=stdm1 stdm2 stdy1;
run;

\*Saving the means and standard deviations as macro variables for use later on;
data means; set mmeans;
call symput ("meanm1", meanm1);
call symput ("meanm2", meanm2);
call symput ("meany1", meany1);
call symput ("stdm1", stdm1);
call symput ("stdm2", stdm2);
call symput ("stdy1", stdy1);
run;

data full; set full;
merger=1;
data mmeans; set mmeans;
merger=1;
run;

proc reg data=full outest=file1 noprint covout;
model m2=x m1 xm1;
run;quit;
\*Saving the intercept, regression coefficients, and mean-squared error from each equation;
data out1; set file1;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
im=intercept;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
a1=x;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
z1=m1;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
a2=xm1;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
msem=\_RMSE\_\*\_RMSE\_;
keep im a1 z1 a2 msem;
data out; merge out1 mmeans;
merger=1;
run;
data all; merge out full;
id=\_N\_;
by merger;
run;

\*Empirical Integration;
data gcomp; set all;
\*do loop computes 1000 predicted values for each observation in the dataset;
do j = 1 to 1000;

\*The residual is simulated from a normal distribution with standard deviation equal to the estimates root mean-squared error from the regression model above;
em = sqrt(msem)\*rannor(0);

\*M2\_0 is the predicted value of the mediator at time 2 holding the treatment fixed at X=0;
M2\_0=im + a1\*0+ z1\*m1+ a2\*0\*m1+em;
\*M2\_1 is the predicted value of the mediator at time 2 holding the treatment fixed at X=1;
M2\_1=im + a1\*1+ z1\*m1+ a2\*1\*m1+em;

\*The treatment effect is the difference between the potential outcomes M2\_1-M2-0;
trt = M2\_1-M2\_0;

output;
end;
run;

\*Estimating the mean of baseline variables m1 y1 and m2\_0 m2\_1 and the treatment effect
across all 1000 simulations per observation;
proc means data=gcomp noprint;
class id;
var m1 y1 m2\_0 m2\_1 trt;
output out=mean mean=m1 y1 m2\_0 m2\_1 trt;
run;
data effects; set mean;
if Id = . then delete;
run;

\*Estimating the subgroup effect for individuals that scored 1 SD below the mean of m1 also saving the number of individuals that fall within this subgroup sdbn;
proc means data=effects(where=(m1 le (&meanm1-&stdm1))) noprint;
var trt;
output out=sdb mean=sdbtrt n = sdbn;
run;
\*Estimating the subgroup effect for individuals that scored 1 SD above the mean of m1
also saving the number of individuals that fall within this subgroup sdan;
proc means data=effects(where=(m1 ge (&meanm1+&stdm1))) noprint;
var trt;
output out=sda mean=sdatrt n = sdan;
run;
\*Estimating the subgroup effect for individuals that scored greater than 1 SD below the mean of m1 and less than 1 SD above the mean of m1
also saving the number of individuals that fall within this subgroup midn;
proc means data=effects(where=(m1 lt (&meanm1+&stdm1) & m1 gt (&meanm1-&stdm1))) noprint;
var trt;
output out=mid mean=midtrt n = midn;
run;
\*Estimating the average treatment effect also saving the total number of individuals in the sample n;
proc means data=effects noprint;
var trt;
output out=avg mean=trt n =n;
run;

data allmean; merge sdb sda mid avg;
\*Verifying the average(marginal) treatment effect is a weighted average of the subgroup effects above;
marg=midtrt\*(midn/n)+sdbtrt\*(sdbn/n)+sdatrt\*(sdan/n);
run;

proc means data=allmean;
var trt sdbtrt sdatrt midtrt sdbn sdan midn n marg;
run;

\*Saving the point estimates;
data allmean; set allmean;
call symput ("trt", trt);
call symput ("sdbtrt", sdbtrt);
call symput ("sdatrt", sdatrt);
call symput ("midtrt", midtrt);
run;

Bootstrapping procedure starts here. The above code is repeated for each bootstrap sample.

\*Bootstrap procedure;
\*Nboot is the number of bootstrap replications;
%let nboot=1000;
\*sampsize is the observed sample size;
%let sampsize=500;
proc surveyselect data=full noprint out=outtemp method=urs sampsize=&sampsize rep=&nboot outhits;
run;
quit;

proc means data=outtemp noprint;
by replicate;
var m1 m2 y1;
output out=mmeans mean=meanm1 meanm2 meany1 std=stdm1 stdm2 stdy1;
run;

data mmeans; set mmeans;
merger=1;
run;

\*Estimating linear models;
proc reg data=outtemp outest=file1 noprint covout;
by replicate;
model m2=x m1 xm1;
run;quit;
\*Saving the intercept, regression coefficients, and mean-squared error from each equation;
data out1; set file1;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
im=intercept;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
a1=x;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
z1=m1;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
a2=xm1;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
msem=\_RMSE\_\*\_RMSE\_;
keep im a1 z1 a2 msem replicate;
data out; merge out1 mmeans;
id=\_N\_;
run;
data all; merge out outtemp;
id=\_N\_;
by replicate;
run;

%MACRO bootstrap;
data summary; set \_null\_;
%Do r=1 %to &nboot;
\*G-formula;
data gcomp; set all(where=(replicate=&r));
do j = 1 to 1000;
em = sqrt(msem)\*rannor(0);

M2\_0=im + a1\*0 + z1\*m1 + a2\*0\*m1 + em;
M2\_1=im + a1\*1 + z1\*m1 + a2\*1\*m1 + em;

trt = M2\_1-M2\_0;

output;
end;
run;

data meansim; set gcomp;
call symput ("meanm1", meanm1);
call symput ("meanm2", meanm2);
call symput ("meany1", meany1);
call symput ("stdm1", stdm1);
call symput ("stdm2", stdm2);
call symput ("stdy1", stdy1);
run;

proc means data=gcomp noprint;
class replicate id;
var m1 y1 m2\_0 m2\_1 trt;
output out=mean mean=m1 y1 m2\_0 m2\_1 trt;
run;
data effects; set mean;
if Id = . then delete;
if replicate = . then delete;
run;

proc means data=effects(where=(m1 le (&meanm1-&stdm1))) noprint;
var trt;
output out=sdb mean=sdbtrt n = sdbn;
run;
proc means data=effects(where=(m1 ge (&meanm1+&stdm1))) noprint;
var trt;
output out=sda mean=sdatrt n = sdan;
run;
proc means data=effects(where=(m1 lt (&meanm1+&stdm1) & m1 gt (&meanm1-&stdm1))) noprint;
var trt;
output out=mid mean=midtrt n = midn;
run;
proc means data=effects noprint;
var trt;
output out=avg mean=trt n =n;
run;

data allmean; merge sdb sda mid avg;
run;

data new; set summary;
data summary; set allmean new;
run;
%end;

%mend bootstrap;
%bootstrap;

data summary; set summary;
marg=midtrt\*(midn/n)+sdbtrt\*(sdbn/n)+sdatrt\*(sdan/n);
run;

\*Procedure to compute the 2.5% and 97.5% percentiles for each potential outcome and treatment effect;
proc sort data=summary;
 by trt;
run;
data trt; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("trt\_LCL95", trt);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("trt\_UCL95", trt);
run;

proc sort data=summary;
 by sdbtrt;
run;
data sdbtrt; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("sdbtrt\_LCL95", sdbtrt);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("sdbtrt\_UCL95", sdbtrt);
run;

proc sort data=summary;
 by sdatrt;
run;
data sdatrt; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("sdatrt\_LCL95", sdatrt);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("sdatrt\_UCL95", sdatrt);
run;

proc sort data=summary;
 by midtrt;
run;
data midtrt; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("midtrt\_LCL95", midtrt);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("midtrt\_UCL95", midtrt);
run;

data ci;
trt=&trt;
trt\_LCL95=&trt\_LCL95;
trt\_UCL95=&trt\_UCL95;

sdatrt=&sdatrt;
sdatrt\_LCL95=&sdatrt\_LCL95;
sdatrt\_UCL95=&sdatrt\_UCL95;

sdbtrt=&sdbtrt;
sdbtrt\_LCL95=&sdbtrt\_LCL95;
sdbtrt\_UCL95=&sdbtrt\_UCL95;

midtrt=&midtrt;
midtrt\_LCL95=&midtrt\_LCL95;
midtrt\_UCL95=&midtrt\_UCL95;

run;
\*Printing the point estimates and 95% percentile bootstrap CIs;
\*Prints point estimates and 95% CIs;
/\*Trt is the average treatment effect, sdbtrt is the treatment effect for the subgroup who scored 1 SD below the mean of M1 or lower, sdatrt is the treatment effect for the subgroup who scored 1 SD above the mean of M1 or higher, midtrt is the treatment effect for the subgroup who scored between 1 SD below and 1 SD above the mean or higher \*/
proc print data=ci;
run;

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| trt | trt\_LCL95 | trt\_UCL95 | sdatrt | sdatrt\_LCL95 | sdatrt\_UCL95 | sdbtrt | sdbtrt\_LCL95 | sdbtrt\_UCL95 | midtrt | midtrt\_LCL95 | midtrt\_UCL95 |
| 0.50491 | 0.31896 | 0.68508 | 1.09437 | 0.77764 | 1.41646 | -0.087082 | -0.38529 | 0.25226 | 0.50377 | 0.32101 | 0.69213 |

**Example 2**

Example 2 - Moderated Mediation with a-path and b-path moderated by baseline covariate M1. The a path is X->M2 and is moderated by baseline M1,the c’ path is X->Y2 and is moderated by baseline M1, and the b path is M2->Y2 and is moderated by baseline M1. There is an additional baseline covariate, Y1. This covariate can be excluded from the model if the model is not a two-wave mediation model.

\*Saving means and standard deviations for use in defined subgroups;
proc means data=full noprint;
var m1 m2 y1;
output out=mmeans mean=meanm1 meanm2 meany1 std=stdm1 stdm2 stdy1;
run;

\*Saving the means and standard deviations as macro variables for use later on;
data means; set mmeans;
call symput ("meanm1", meanm1);
call symput ("meanm2", meanm2);
call symput ("meany1", meany1);
call symput ("stdm1", stdm1);
call symput ("stdm2", stdm2);
call symput ("stdy1", stdy1);
run;

data mmeans; set mmeans;
merger=1;
run;

\*Estimating mediator and outcome models with xm1 and m1m2 interaction terms;
proc reg data=full outest=file1 covout noprint;
model m2=x m1 y1 xm1;
model y2=x m2 m1 y1 xm1 m1m2;
run;quit;
\*Saving the intercept, regression coefficients, and mean-squared error from each equation;
data out1; set file1;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
im=intercept;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
a1=x;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
z1=m1;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
z2=y1;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
a2=xm1;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
msem=\_RMSE\_\*\_RMSE\_;
keep im a1 z1 z2 a2 msem;
data out2; set file1;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
iy=intercept;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
cp1=x;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
b1=m2;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
z3=m1;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
z4=y1;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
cp2=xm1;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
b2=m1m2;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
msey=\_RMSE\_\*\_RMSE\_;
keep iy cp1 b1 z3 z4 cp2 b2 msey;

data full; set full;
merger=1;
data out; merge out1 out2 mmeans;
merger=1;
run;

data all; merge out full;
id=\_N\_;
by merger;
run;

\*Integration;
data gcomp; set all;
by merger;
\*Creating 1000 predicted values of the (nested) potential outcomes for each observation;
do j = 1 to 1000;
\*The residuals are simulated from a normal distribution with standard deviations equal to the estimates root mean-squared error from the regression models above;
em = sqrt(msem)\*rannor(0);
ey = sqrt(msey)\*rannor(0);

\*M2\_0 is the predicted value of the mediator at time 2 holding the treatment fixed at X=0;
M2\_0=im + a1\*0 + z1\*m1 + z2\*y1 + a2\*0\*m1 + em;
\*M2\_1 is the predicted value of the mediator at time 2 holding the treatment fixed at X=1;
M2\_1=im + a1\*1 + z1\*m1 + z2\*y1 + a2\*1\*m1 + em;

\*Y2\_0\_M2\_0 is the predicted value of the outcome at time 2 holding the treatment fixed at X=0 and holding the mediator constant at the individual's predicted value of M2 holding the treatment fixed at X=0;
Y2\_0\_M2\_0 = iy + cp1\*0 + b1\*M2\_0 + z3\*m1 + z4\*y1 + cp2\*0\*m1 + b2\*m1\*m2\_0 + ey;
\*Y2\_0\_M2\_1 is the predicted value of the outcome at time 2 holding the treatment fixed at X=0 and holding the mediator constant at the individual's predicted value of M2 holding the treatment fixed at X=1;
Y2\_0\_M2\_1 = iy + cp1\*0 + b1\*M2\_1 + z3\*m1 + z4\*y1 + cp2\*0\*m1 + b2\*m1\*m2\_1 + ey;

\*Estimating all mediated effect;
NIE = Y2\_0\_M2\_1-Y2\_0\_M2\_0;

output;
end;
run;

\*Estimating the mean of the potential outcomes and causal mediation effects across all 1000 predictions per observation;
proc means data=gcomp noprint;
class id;
var m1 y1 m2\_0 m2\_1 Y2\_0\_M2\_1 Y2\_0\_M2\_0 nie;
output out=mean mean=m1 y1 m2\_0 m2\_1 Y2\_0\_M2\_1 Y2\_0\_M2\_0 nie;
run;
data effects; set mean;
if Id = . then delete;
run;

\*Estimating the causal mediation effects for subgroup of individuals that scored 1SD below the mean or lower of m1;
proc means data=effects(where=(m1 le (&meanm1-&stdm1))) noprint;
var nie;
output out=sdb mean=sdbnie n = sdbn;
run;
\*Estimating the causal mediation effects for subgroup of individuals that scored 1SD above the mean or higher of m1;
proc means data=effects(where=(m1 ge (&meanm1+&stdm1))) noprint;
var nie;
output out=sda mean=sdanie n = sdan;
run;
\*Estimating the causal mediation effects for subgroup of individuals that scored above 1SD below the mean or below 1SD above the mean of m1;
proc means data=effects(where=(m1 lt (&meanm1+&stdm1) & m1 gt (&meanm1-&stdm1))) noprint;
var nie;
output out=mid mean=midnie n = midn;
run;
\*Estimating the average mediated effect;
proc means data=effects noprint;
var nie;
output out=avg mean=nie n =n;
run;

data med; merge sdb sda mid avg;
run;

\*Saving the point estimates of the mediated effects;
data medeffs1; set med;
call symput ("NIE", NIE);
call symput ("sdbNIE", sdbNIE);
call symput ("sdaNIE", sdaNIE);
call symput ("midNIE", midNIE);
run;

Bootstrapping procedure starts here. The above code is repeated for each bootstrap sample.

\*Bootstrap procedure;
\*Nboot is the number of bootstrap replications;
%let nboot=1000;
\*sampsize is the observed sample size;
%let sampsize=500;
proc surveyselect data=full noprint out=outtemp method=urs sampsize=&sampsize rep=&nboot outhits;
run;
quit;

proc means data=outtemp noprint;
by replicate;
var m1 m2 y1;
output out=mmeans mean=meanm1 meanm2 meany1 std=stdm1 stdm2 stdy1;
run;

data mmeans; set mmeans;
merger=1;
run;

\*Estimating linear models;
proc reg data=outtemp outest=file1 noprint covout;
by replicate;
model m2=x m1 y1 xm1;
model y2=x m2 m1 y1 xm1 m1m2;
run;quit;
\*Saving the intercept, regression coefficients, and mean-squared error from each equation;
data out1; set file1;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
im=intercept;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
a1=x;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
z1=m1;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
z2=y1;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
a2=xm1;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
msem=\_RMSE\_\*\_RMSE\_;
keep im a1 z1 z2 a2 msem replicate;
data out2; set file1;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
iy=intercept;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
cp1=x;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
b1=m2;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
z3=m1;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
z4=y1;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
cp2=xm1;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
b2=m1m2;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
msey=\_RMSE\_\*\_RMSE\_;
keep iy cp1 b1 z3 z4 cp2 b2 msey replicate;
data out; merge out1 out2 mmeans;
id=\_N\_;
run;
data all; merge out outtemp;
id=\_N\_;
by replicate;
run;

%MACRO bootstrap;
data summary; set \_null\_;
%Do r=1 %to &nboot;
\*G-formula;
data gcomp; set all(where=(replicate=&r));
do j = 1 to 1000;
em = sqrt(msem)\*rannor(0);
ey = sqrt(msey)\*rannor(0);
M2\_0=im + a1\*0 + z1\*m1 + z2\*y1 + a2\*0\*m1 + em;
M2\_1=im + a1\*1 + z1\*m1 + z2\*y1 + a2\*1\*m1 + em;

Y2\_0\_M2\_0 = iy + cp1\*0 + b1\*M2\_0 + z3\*m1 + z4\*y1 + cp2\*0\*m1 + b2\*m1\*m2\_0 + ey;
Y2\_0\_M2\_1 = iy + cp1\*0 + b1\*M2\_1 + z3\*m1 + z4\*y1 + cp2\*0\*m1 + b2\*m1\*m2\_1 + ey;

\*Estimating the mediated effect;
NIE = Y2\_0\_M2\_1-Y2\_0\_M2\_0;

output;
end;
run;

data meansim; set gcomp;
call symput ("meanm1", meanm1);
call symput ("meanm2", meanm2);
call symput ("meany1", meany1);
call symput ("stdm1", stdm1);
call symput ("stdm2", stdm2);
call symput ("stdy1", stdy1);
run;

proc means data=gcomp noprint;
class replicate id;
var m1 y1 m2\_0 m2\_1 Y2\_0\_M2\_1 Y2\_0\_M2\_0 nie;
output out=mean mean=m1 y1 m2\_0 m2\_1 Y2\_0\_M2\_1 Y2\_0\_M2\_0 nie;
run;
data effects; set mean;
if Id = . then delete;
if replicate = . then delete;
run;

proc means data=effects(where=(m1 le (&meanm1-&stdm1))) noprint;
var nie;
output out=sdb mean=sdbnie n = sdbn;
run;
proc means data=effects(where=(m1 ge (&meanm1+&stdm1))) noprint;
var nie;
output out=sda mean=sdanie n = sdan;
run;
proc means data=effects(where=(m1 lt (&meanm1+&stdm1) & m1 gt (&meanm1-&stdm1))) noprint;
var nie;
output out=mid mean=midnie n = midn;
run;
proc means data=effects noprint;
var nie;
output out=avg mean=nie n =n;
run;

data allmean; merge sdb sda mid avg;
run;

data new; set summary;
data summary; set allmean new;
run;
%end;

%mend bootstrap;
%bootstrap;

\*Procedure to compute the 2.5% and 97.5% percentiles for the mediated effect;
proc sort data=summary;
 by NIE;
run;
data NIE; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("NIE\_LCL95", NIE);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("NIE\_UCL95", NIE);
run;

proc sort data=summary;
 by sdbnie;
run;
data sdbnie; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("sdbnie\_LCL95", sdbnie);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("sdbnie\_UCL95", sdbnie);
run;

proc sort data=summary;
 by sdanie;
run;
data sdapnie; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("sdanie\_LCL95", sdanie);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("sdanie\_UCL95", sdanie);
run;

proc sort data=summary;
 by midnie;
run;
data midnie; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("midnie\_LCL95", midnie);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("midnie\_UCL95", midnie);
run;

\*Saving point estimates and 95% percentile bootstrap CIs;
data ci;

NIE=&NIE;
NIE\_LCL95=&NIE\_LCL95;
NIE\_UCL95=&NIE\_UCL95;

sdbnie=&sdbnie;
sdbNIE\_LCL95=&sdbNIE\_LCL95;
sdbNIE\_UCL95=&sdbNIE\_UCL95;

sdanie=&sdanie;
sdaNIE\_LCL95=&sdaNIE\_LCL95;
sdaNIE\_UCL95=&sdaNIE\_UCL95;

midnie=&midnie;
midNIE\_LCL95=&midNIE\_LCL95;
midNIE\_UCL95=&midNIE\_UCL95;

run;

\*Prints point estimates and 95% CIs;
/\*NIE is the average mediated effect,
 sdbNIE is the mediated effect for the subgroup who scored 1 SD below the mean of M1 or lower,
 sdaNIE is the mediated effect for the subgroup who scored 1 SD above the mean of M1 or higher,
 midNIE is the mediated effect for the subgroup who scored between 1 SD below and 1 SD abve the mean or higher \*/
proc print data=ci;
var NIE NIE\_LCL95 NIE\_UCL95 sdbNIE sdbNIE\_LCL95 sdbNIE\_UCL95 sdaNIE sdaNIE\_LCL95 sdaNIE\_UCL95
midNIE midNIE\_LCL95 midNIE\_UCL95;
run;

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| NIE | NIE\_LCL95 | NIE\_UCL95 | sdbnie | sdbNIE\_LCL95 | sdbNIE\_UCL95 | sdanie | sdaNIE\_LCL95 | sdaNIE\_UCL95 | midnie | midNIE\_LCL95 | midNIE\_UCL95 |
| 0.33891 | 0.22733 | 0.46601 | -0.016761 | -0.11527 | 0.083207 | 0.86627 | 0.58073 | 1.18026 | 0.29375 | 0.18888 | 0.42053 |

## Example 3

Example 3 - Moderated Causal Mediation effects: PNIE, TNIE, PNDE, TNDE, CDE, and TE. The a path is X->M2 and is moderated by baseline M1,the c’ path is X->Y2 and is moderated by baseline M1, and the b path is M2->Y2 and is moderated by baseline M1. There is a treatment-mediator interaction XM2 which results in different mediated effects, PNIE, TNIE, and different direct effects, PNDE, TNDE, and CDE. There is an additional baseline covariate, Y1. This covariate can be excluded from the model if the model is not a two-wave mediation model.

\*Saving means and standard deviations for use in defined subgroups;
proc means data=full noprint;
var m1 m2 y1;
output out=mmeans mean=meanm1 meanm2 meany1 std=stdm1 stdm2 stdy1;
run;

\*Saving the means and standard deviations as macro variables for use later on;
data means; set mmeans;
call symput ("meanm1", meanm1);
call symput ("meanm2", meanm2);
call symput ("meany1", meany1);
call symput ("stdm1", stdm1);
call symput ("stdm2", stdm2);
call symput ("stdy1", stdy1);
run;

data mmeans; set mmeans;
merger=1;
run;

\*Estimating mediator and outcome models with xm1, m1m2, and xm2 interaction terms;
proc reg data=full outest=file1 noprint covout;
model m2=x m1 y1 xm1;
model y2=x m2 m1 y1 xm1 m1m2 xm2;
run;quit;
\*Saving the intercept, regression coefficients, and mean-squared error from each equation;
data out1; set file1;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
im=intercept;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
a1=x;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
z1=m1;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
z2=y1;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
a2=xm1;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
msem=\_RMSE\_\*\_RMSE\_;
keep im a1 z1 z2 a2 msem;
data out2; set file1;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
iy=intercept;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
cp1=x;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
b1=m2;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
z3=m1;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
z4=y1;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
cp2=xm1;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
b2=m1m2;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
h=xm2;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
msey=\_RMSE\_\*\_RMSE\_;
keep iy cp1 b1 z3 z4 cp2 b2 h msey;

data out; merge out1 out2 mmeans;
merger=1;
run;

data full; set full;
merger=1;
data all; merge out full;
id=\_N\_;
by merger;
run;

\*Integration;
data gcomp; set all;
by merger;
\*Creating 1000 predicted values of the (nested) potential outcomes for each observation;
do j = 1 to 1000;
\*The residuals are simulated from a normal distribution with standard deviations equal to the estimates root mean-squared error from the regression models above;
em = sqrt(msem)\*rannor(0);
ey = sqrt(msey)\*rannor(0);

\*M2\_0 is the predicted value of the mediator at time 2 holding the treatment fixed at X=0;
M2\_0=im + a1\*0 + z1\*m1 + z2\*y1 + a2\*0\*m1 + em;
\*M2\_1 is the predicted value of the mediator at time 2 holding the treatment fixed at X=1;
M2\_1=im + a1\*1 + z1\*m1 + z2\*y1 + a2\*1\*m1 + em;

\*Y2\_0\_M2\_0 is the predicted value of the outcome at time 2 holding the treatment fixed at X=0 and holding the mediator constant at the individual's predicted value of M2 holding the treatment fixed at X=0;
Y2\_0\_M2\_0 = iy + cp1\*0 + b1\*M2\_0 + z3\*m1 + z4\*y1 + cp2\*0\*m1 + b2\*m1\*m2\_0 + h\*0\*M2\_0 + ey;
\*Y2\_1\_M2\_0 is the predicted value of the outcome at time 2 holding the treatment fixed at X=1 and holding the mediator constant at the individual's predicted value of M2 holding the treatment fixed at X=0;
Y2\_1\_M2\_0 = iy + cp1\*1 + b1\*M2\_0 + z3\*m1 + z4\*y1 + cp2\*1\*m1 + b2\*m1\*m2\_0 + h\*1\*M2\_0 + ey;
\*Y2\_0\_M2\_1 is the predicted value of the outcome at time 2 holding the treatment fixed at X=0 and holding the mediator constant at the individual's predicted value of M2 holding the treatment fixed at X=1;
Y2\_0\_M2\_1 = iy + cp1\*0 + b1\*M2\_1 + z3\*m1 + z4\*y1 + cp2\*0\*m1 + b2\*m1\*m2\_1 + h\*0\*M2\_1 + ey;
\*Y2\_1\_M2\_1 is the predicted value of the outcome at time 2 holding the treatment fixed at X=1 and holding the mediator constant at the individual's predicted value of M2 holding the treatment fixed at X=1;
Y2\_1\_M2\_1 = iy + cp1\*1 + b1\*M2\_1 + z3\*m1 + z4\*y1 + cp2\*1\*m1 + b2\*m1\*m2\_1 + h\*1\*M2\_1 + ey;

\*Value of the mediator variable at time 2 that the CDE is estimated;
m=&meanm1;
\*Y2\_0\_m is the predicted value of the outcome at time 2 holding the treatment fixed at X=0 and holding the mediator constant at a fixed value for all individuals;
Y2\_0\_m = iy + cp1\*0 + b1\*m + z3\*m1 + z4\*y1 + cp2\*0\*m1 + b2\*m1\*m + h\*0\*m + ey;
\*Y2\_1\_m is the predicted value of the outcome at time 2 holding the treatment fixed at X=1 and holding the mediator constant at a fixed value for all individuals;
Y2\_1\_m = iy + cp1\*1 + b1\*m + z3\*m1 + z4\*y1 + cp2\*1\*m1 + b2\*m1\*m + h\*1\*m + ey;

\*Estimating all causal mediation effects;
PNIE = Y2\_0\_M2\_1-Y2\_0\_M2\_0;
TNIE = Y2\_1\_M2\_1-Y2\_1\_M2\_0;
PNDE = Y2\_1\_M2\_0-Y2\_0\_M2\_0;
TNDE = Y2\_1\_M2\_1-Y2\_0\_M2\_1;
CDE= Y2\_1\_m-Y2\_0\_m;
TE=PNIE+TNDE;

output;
end;
run;

\*Estimating the mean of the potential outcomes and causal mediation effects across all 1000 predictions per observation;
proc means data=gcomp noprint;
class id;
var m1 y1 m2\_0 m2\_1 Y2\_0\_M2\_1 Y2\_0\_M2\_0 Y2\_1\_M2\_1 Y2\_1\_M2\_0 Y2\_1\_m Y2\_0\_m
pnie tnie pnde tnde cde te;
output out=mean mean=m1 y1 m2\_0 m2\_1 Y2\_0\_M2\_1 Y2\_0\_M2\_0 Y2\_1\_M2\_1 Y2\_1\_M2\_0 Y2\_1\_m Y2\_0\_m
pnie tnie pnde tnde cde te;
run;
data effects; set mean;
if Id = . then delete;
run;

\*Estimating the causal mediation effects for subgroup of individuals that scored 1SD below the mean or lower of m1;
proc means data=effects(where=(m1 le (&meanm1-&stdm1))) noprint;
var pnie tnie pnde tnde cde te;
output out=sdb mean=sdbpnie sdbtnie sdbpnde sdbtnde sdbcde sdbte n = sdbn;
run;
\*Estimating the causal mediation effects for subgroup of individuals that scored 1SD above the mean or higher of m1;
proc means data=effects(where=(m1 ge (&meanm1+&stdm1))) noprint;
var pnie tnie pnde tnde cde te;
output out=sda mean=sdapnie sdatnie sdapnde sdatnde sdacde sdate n = sdan;
run;
\*Estimating the causal mediation effects for subgroup of individuals that scored above 1SD below the mean or below 1SD above the mean of m1;
proc means data=effects(where=(m1 lt (&meanm1+&stdm1) & m1 gt (&meanm1-&stdm1))) noprint;
var pnie tnie pnde tnde cde te;
output out=mid mean=midpnie midtnie midpnde midtnde midcde midte n = midn;
run;
\*Estimating the average causal mediation effects;
proc means data=effects noprint;
var pnie tnie pnde tnde cde te;
output out=avg mean=pnie tnie pnde tnde cde te n =n;
run;

data med; merge sdb sda mid avg;
run;

\*Saving the point estimates of the causal mediation effects;
data medeffs1; set med;
call symput ("PNIE", PNIE);
call symput ("sdbPNIE", sdbPNIE);
call symput ("sdaPNIE", sdaPNIE);
call symput ("midPNIE", midPNIE);
call symput ("TNIE", TNIE);
call symput ("sdbTNIE", sdbTNIE);
call symput ("sdaTNIE", sdaTNIE);
call symput ("midTNIE", midTNIE);
call symput ("PNDE", PNDE);
call symput ("sdbPNDE", sdbPNDE);
call symput ("sdaPNDE", sdaPNDE);
call symput ("midPNDE", midPNDE);
call symput ("TNDE", TNDE);
call symput ("sdbTNDE", sdbTNDE);
call symput ("sdaTNDE", sdaTNDE);
call symput ("midTNDE", midTNDE);
call symput ("CDE", CDE);
call symput ("sdbCDE", sdbCDE);
call symput ("sdaCDE", sdaCDE);
call symput ("midCDE", midCDE);
call symput ("TE", TE);
call symput ("sdbTE", sdbTE);
call symput ("sdaTE", sdaTE);
call symput ("midTE", midTE);
run;

Bootstrapping procedure starts here. The above code is repeated for each bootstrap sample.

\*Bootstrap procedure;
\*Nboot is the number of bootstrap replications;
%let nboot=1000;
\*sampsize is the observed sample size;
%let sampsize=500;
proc surveyselect data=full noprint out=outtemp method=urs sampsize=&sampsize rep=&nboot outhits;
run;
quit;

proc means data=outtemp noprint;
by replicate;
var m1 m2 y1;
output out=mmeans mean=meanm1 meanm2 meany1 std=stdm1 stdm2 stdy1;
run;

data mmeans; set mmeans;
merger=1;
run;

proc reg data=outtemp outest=file1 noprint covout;
by replicate;
model m2=x m1 y1 xm1;
model y2=x m2 m1 y1 xm1 m1m2 xm2;
run;quit;
\*Saving the intercept, regression coefficients, and mean-squared error from each equation;
data out1; set file1;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
im=intercept;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
a1=x;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
z1=m1;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
z2=y1;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
a2=xm1;
if \_MODEL\_ ='MODEL1'& \_TYPE\_='PARMS';
msem=\_RMSE\_\*\_RMSE\_;
keep im a1 z1 z2 a2 msem replicate;
data out2; set file1;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
iy=intercept;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
cp1=x;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
b1=m2;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
z3=m1;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
z4=y1;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
cp2=xm1;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
b2=m1m2;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
h=xm2;
if \_MODEL\_ ='MODEL2'& \_TYPE\_='PARMS';
msey=\_RMSE\_\*\_RMSE\_;
keep iy cp1 b1 z3 z4 cp2 b2 h msey replicate;
data out; merge out1 out2 mmeans;
id=\_N\_;
run;
data all; merge out outtemp;
id=\_N\_;
by replicate;
run;

%MACRO bootstrap;
data summary; set \_null\_;
%Do r=1 %to &nboot;
\*G-formula;
data gcomp; set all(where=(replicate=&r));
do j = 1 to 1000;
em = sqrt(msem)\*rannor(0);
ey = sqrt(msey)\*rannor(0);
M2\_0=im + a1\*0 + z1\*m1 + z2\*y1 + a2\*0\*m1 + em;
M2\_1=im + a1\*1 + z1\*m1 + z2\*y1 + a2\*1\*m1 + em;

Y2\_0\_M2\_0 = iy + cp1\*0 + b1\*M2\_0 + z3\*m1 + z4\*y1 + cp2\*0\*m1 + b2\*m1\*m2\_0 + h\*0\*M2\_0 + ey;
Y2\_1\_M2\_0 = iy + cp1\*1 + b1\*M2\_0 + z3\*m1 + z4\*y1 + cp2\*1\*m1 + b2\*m1\*m2\_0 + h\*1\*M2\_0 + ey;
Y2\_0\_M2\_1 = iy + cp1\*0 + b1\*M2\_1 + z3\*m1 + z4\*y1 + cp2\*0\*m1 + b2\*m1\*m2\_1 + h\*0\*M2\_1 + ey;
Y2\_1\_M2\_1 = iy + cp1\*1 + b1\*M2\_1 + z3\*m1 + z4\*y1 + cp2\*1\*m1 + b2\*m1\*m2\_1 + h\*1\*M2\_1 + ey;

\*Value of the meadiator variable at time 2 that the CDE is estimated;
m=&meanm1;
Y2\_0\_m = iy + cp1\*0 + b1\*m + z3\*m1 + z4\*y1 + cp2\*0\*m1 + b2\*m1\*m + h\*0\*m + ey;
Y2\_1\_m = iy + cp1\*1 + b1\*m + z3\*m1 + z4\*y1 + cp2\*1\*m1 + b2\*m1\*m + h\*1\*m + ey;

\*Estimating all causal mediation effects;
PNIE = Y2\_0\_M2\_1-Y2\_0\_M2\_0;
TNIE = Y2\_1\_M2\_1-Y2\_1\_M2\_0;
PNDE = Y2\_1\_M2\_0-Y2\_0\_M2\_0;
TNDE = Y2\_1\_M2\_1-Y2\_0\_M2\_1;
CDE= Y2\_1\_m-Y2\_0\_m;
TE=PNIE+TNDE;

output;
end;
run;

data meansim; set gcomp;
call symput ("meanm1", meanm1);
call symput ("meanm2", meanm2);
call symput ("meany1", meany1);
call symput ("stdm1", stdm1);
call symput ("stdm2", stdm2);
call symput ("stdy1", stdy1);
run;

proc means data=gcomp noprint;
class replicate id;
var m1 y1 m2\_0 m2\_1 Y2\_0\_M2\_1 Y2\_0\_M2\_0 Y2\_1\_M2\_1 Y2\_1\_M2\_0 Y2\_1\_m Y2\_0\_m
pnie tnie pnde tnde cde te;
output out=mean mean=m1 y1 m2\_0 m2\_1 Y2\_0\_M2\_1 Y2\_0\_M2\_0 Y2\_1\_M2\_1 Y2\_1\_M2\_0 Y2\_1\_m Y2\_0\_m
pnie tnie pnde tnde cde te;
run;
data effects; set mean;
if Id = . then delete;
if replicate = . then delete;
run;

proc means data=effects(where=(m1 le (&meanm1-&stdm1))) noprint;
var pnie tnie pnde tnde cde te;
output out=sdb mean=sdbpnie sdbtnie sdbpnde sdbtnde sdbcde sdbte n = sdbn;
run;
proc means data=effects(where=(m1 ge (&meanm1+&stdm1))) noprint;
var pnie tnie pnde tnde cde te;
output out=sda mean=sdapnie sdatnie sdapnde sdatnde sdacde sdate n = sdan;
run;
proc means data=effects(where=(m1 lt (&meanm1+&stdm1) & m1 gt (&meanm1-&stdm1))) noprint;
var pnie tnie pnde tnde cde te;
output out=mid mean=midpnie midtnie midpnde midtnde midcde midte n = midn;
run;
proc means data=effects noprint;
var pnie tnie pnde tnde cde te;
output out=avg mean=pnie tnie pnde tnde cde te n =n;
run;

data allmean; merge sdb sda mid avg;
run;

data new; set summary;
data summary; set allmean new;
run;
%end;

%mend bootstrap;
%bootstrap;

\*Procedure to compute the 2.5% and 97.5% percentiles for the causal mediation effects;
proc sort data=summary;
 by PNIE;
run;
data PNIE; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("PNIE\_LCL95", PNIE);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("PNIE\_UCL95", PNIE);
run;

proc sort data=summary;
 by sdbpnie;
run;
data sdbpnie; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("sdbpnie\_LCL95", sdbpnie);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("sdbpnie\_UCL95", sdbpnie);
run;

proc sort data=summary;
 by sdapnie;
run;
data sdapnie; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("sdapnie\_LCL95", sdapnie);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("sdapnie\_UCL95", sdapnie);
run;

proc sort data=summary;
 by midpnie;
run;
data midpnie; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("midpnie\_LCL95", midpnie);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("midpnie\_UCL95", midpnie);
run;

proc sort data=summary;
 by TNIE;
run;
data TNIE; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("TNIE\_LCL95", TNIE);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("TNIE\_UCL95", TNIE);
run;

proc sort data=summary;
 by sdbtnie;
run;
data sdbtnie; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("sdbtnie\_LCL95", sdbtnie);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("sdbtnie\_UCL95", sdbtnie);
run;

proc sort data=summary;
 by sdatnie;
run;
data sdatnie; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("sdatnie\_LCL95", sdatnie);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("sdatnie\_UCL95", sdatnie);
run;

proc sort data=summary;
 by midtnie;
run;
data midtnie; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("midtnie\_LCL95", midtnie);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("midtnie\_UCL95", midtnie);
run;

proc sort data=summary;
 by PNDE;
run;
data PNDE; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("PNDE\_LCL95", PNDE);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("PNDE\_UCL95", PNDE);
run;

proc sort data=summary;
 by sdbPNDE;
run;
data sdbPNDE; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("sdbPNDE\_LCL95", sdbPNDE);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("sdbPNDE\_UCL95", sdbPNDE);
run;

proc sort data=summary;
 by sdaPNDE;
run;
data sdaPNDE; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("sdaPNDE\_LCL95", sdaPNDE);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("sdaPNDE\_UCL95", sdaPNDE);
run;

proc sort data=summary;
 by midPNDE;
run;
data midPNDE; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("midPNDE\_LCL95", midPNDE);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("midPNDE\_UCL95", midPNDE);
run;

proc sort data=summary;
 by TNDE;
run;
data TNDE; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("TNDE\_LCL95", TNDE);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("TNDE\_UCL95", TNDE);
run;

proc sort data=summary;
 by sdbTNDE;
run;
data sdbTNDE; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("sdbTNDE\_LCL95", sdbTNDE);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("sdbTNDE\_UCL95", sdbTNDE);
run;

proc sort data=summary;
 by sdaTNDE;
run;
data sdaTNDE; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("sdaTNDE\_LCL95", sdaTNDE);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("sdaTNDE\_UCL95", sdaTNDE);
run;

proc sort data=summary;
 by midTNDE;
run;
data midTNDE; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("midTNDE\_LCL95", midTNDE);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("midTNDE\_UCL95", midTNDE);
run;

proc sort data=summary;
 by CDE;
run;
data CDE; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("CDE\_LCL95", CDE);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("CDE\_UCL95", CDE);
run;

proc sort data=summary;
 by sdbCDE;
run;
data sdbCDE; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("sdbCDE\_LCL95", sdbCDE);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("sdbCDE\_UCL95", sdbCDE);
run;

proc sort data=summary;
 by sdaCDE;
run;
data sdaCDE; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("sdaCDE\_LCL95", sdaCDE);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("sdaCDE\_UCL95", sdaCDE);
run;

proc sort data=summary;
 by midCDE;
run;
data midCDE; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("midCDE\_LCL95", midCDE);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("midCDE\_UCL95", midCDE);
run;

proc sort data=summary;
 by TE;
run;
data TE; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("TE\_LCL95", TE);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("TE\_UCL95", TE);
run;

proc sort data=summary;
 by sdbTE;
run;
data sdbTE; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("sdbTE\_LCL95", sdbTE);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("sdbTE\_UCL95", sdbTE);
run;

proc sort data=summary;
 by sdaTE;
run;
data sdaTE; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("sdaTE\_LCL95", sdaTE);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("sdaTE\_UCL95", sdaTE);
run;

proc sort data=summary;
 by midTE;
run;
data midTE; set summary;
 if \_N\_=(ceil((0.05/2)\*&nboot)) then call symput("midTE\_LCL95", midTE);
 if \_N\_=(ceil((1-(0.05/2))\*&nboot)) then call symput("midTE\_UCL95", midTE);
run;

\*Saving point estimates and 95% percentile bootstrap CIs;
data ci;

PNIE=&PNIE;
PNIE\_LCL95=&PNIE\_LCL95;
PNIE\_UCL95=&PNIE\_UCL95;

sdbpnie=&sdbpnie;
sdbPNIE\_LCL95=&sdbPNIE\_LCL95;
sdbPNIE\_UCL95=&sdbPNIE\_UCL95;

sdapnie=&sdapnie;
sdaPNIE\_LCL95=&sdaPNIE\_LCL95;
sdaPNIE\_UCL95=&sdaPNIE\_UCL95;

midpnie=&midpnie;
midPNIE\_LCL95=&midPNIE\_LCL95;
midPNIE\_UCL95=&midPNIE\_UCL95;

TNIE=&TNIE;
tnie\_LCL95=&tnie\_LCL95;
tnie\_UCL95=&tnie\_UCL95;

sdbTNIE=&sdbTNIE;
sdbtnie\_LCL95=&sdbtnie\_LCL95;
sdbtnie\_UCL95=&sdbtnie\_UCL95;

sdaTNIE=&sdaTNIE;
sdatnie\_LCL95=&sdatnie\_LCL95;
sdatnie\_UCL95=&sdatnie\_UCL95;

midTNIE=&midTNIE;
midtnie\_LCL95=&midtnie\_LCL95;
midtnie\_UCL95=&midtnie\_UCL95;

PNDE=&PNDE;
PNDE\_LCL95=&PNDE\_LCL95;
PNDE\_UCL95=&PNDE\_UCL95;

sdbPNDE=&sdbPNDE;
sdbPNDE\_LCL95=&sdbPNDE\_LCL95;
sdbPNDE\_UCL95=&sdbPNDE\_UCL95;

sdaPNDE=&sdaPNDE;
sdaPNDE\_LCL95=&sdaPNDE\_LCL95;
sdaPNDE\_UCL95=&sdaPNDE\_UCL95;

midPNDE=&midPNDE;
midPNDE\_LCL95=&midPNDE\_LCL95;
midPNDE\_UCL95=&midPNDE\_UCL95;

TNDE=&TNDE;
TNDE\_LCL95=&TNDE\_LCL95;
TNDE\_UCL95=&TNDE\_UCL95;

sdbTNDE=&sdbTNDE;
sdbTNDE\_LCL95=&sdbTNDE\_LCL95;
sdbTNDE\_UCL95=&sdbTNDE\_UCL95;

sdaTNDE=&sdaTNDE;
sdaTNDE\_LCL95=&sdaTNDE\_LCL95;
sdaTNDE\_UCL95=&sdaTNDE\_UCL95;

midTNDE=&midTNDE;
midTNDE\_LCL95=&midTNDE\_LCL95;
midTNDE\_UCL95=&midTNDE\_UCL95;

CDE=&CDE;
CDE\_LCL95=&CDE\_LCL95;
CDE\_UCL95=&CDE\_UCL95;

sdbCDE=&sdbCDE;
sdbCDE\_LCL95=&sdbCDE\_LCL95;
sdbCDE\_UCL95=&sdbCDE\_UCL95;

sdaCDE=&sdaCDE;
sdaCDE\_LCL95=&sdaCDE\_LCL95;
sdaCDE\_UCL95=&sdaCDE\_UCL95;

midCDE=&midCDE;
midCDE\_LCL95=&midCDE\_LCL95;
midCDE\_UCL95=&midCDE\_UCL95;

TE=&TE;
TE\_LCL95=&TE\_LCL95;
TE\_UCL95=&TE\_UCL95;

sdbTE=&sdbTE;
sdbTE\_LCL95=&sdbTE\_LCL95;
sdbTE\_UCL95=&sdbTE\_UCL95;

sdaTE=&sdaTE;
sdaTE\_LCL95=&sdaTE\_LCL95;
sdaTE\_UCL95=&sdaTE\_UCL95;

midTE=&midTE;
midTE\_LCL95=&midTE\_LCL95;
midTE\_UCL95=&midTE\_UCL95;
run;

\*Prints point estimates and 95% CIs;
/\*PNIE is the average pure natural indirect effect,
 sdbPNIE is the pure natural indirect effect for the subgroup who scored 1 SD below the mean of M1 or lower,
 sdaPNIE is the pure natural indirect effect for the subgroup who scored 1 SD above the mean of M1 or higher,
 midPNIE is the pure natural indirect effect for the subgroup who scored between 1 SD below and 1 SD above the mean or higher
The TNIE, PNDE, TNDE, CDE, and TE follow the same naming convention \*/
proc print data=ci;
var PNIE PNIE\_LCL95 PNIE\_UCL95 sdbpnie sdbPNIE\_LCL95 sdbPNIE\_UCL95 sdapnie sdaPNIE\_LCL95 sdaPNIE\_UCL95
midpnie midPNIE\_LCL95 midPNIE\_UCL95;
run;
proc print data=ci;
var TNIE TNIE\_LCL95 TNIE\_UCL95 sdbTNIE sdbTNIE\_LCL95 sdbTNIE\_UCL95 sdaTNIE sdaTNIE\_LCL95 sdaTNIE\_UCL95
midTNIE midTNIE\_LCL95 midTNIE\_UCL95;
run;
proc print data=ci;
var PNDE PNDE\_LCL95 PNDE\_UCL95 sdbPNDE sdbPNDE\_LCL95 sdbPNDE\_UCL95 sdaPNDE sdaPNDE\_LCL95 sdaPNDE\_UCL95
midPNDE midPNDE\_LCL95 midPNDE\_UCL95;
run;
proc print data=ci;
var TNDE TNDE\_LCL95 TNDE\_UCL95 sdbTNDE sdbTNDE\_LCL95 sdbTNDE\_UCL95 sdaTNDE sdaTNDE\_LCL95 sdaTNDE\_UCL95
midTNDE midTNDE\_LCL95 midTNDE\_UCL95;
run;
proc print data=ci;
var CDE CDE\_LCL95 CDE\_UCL95 sdbCDE sdbCDE\_LCL95 sdbCDE\_UCL95 sdaCDE sdaCDE\_LCL95 sdaCDE\_UCL95
midCDE midCDE\_LCL95 midCDE\_UCL95;
run;
proc print data=ci;
var TE TE\_LCL95 TE\_UCL95 sdbTE sdbTE\_LCL95 sdbTE\_UCL95 sdaTE sdaTE\_LCL95 sdaTE\_UCL95
midTE midTE\_LCL95 midTE\_UCL95;
run;

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PNIE | PNIE\_LCL95 | PNIE\_UCL95 | sdbpnie | sdbPNIE\_LCL95 | sdbPNIE\_UCL95 | sdapnie | sdaPNIE\_LCL95 | sdaPNIE\_UCL95 | midpnie | midPNIE\_LCL95 | midPNIE\_UCL95 |
| 0.22968 | 0.13675 | 0.33796 | -.003019142 | -0.054717 | 0.043023 | 0.61347 | 0.35571 | 0.91875 | 0.19018 | 0.10501 | 0.28878 |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TNIE | tnie\_LCL95 | tnie\_UCL95 | sdbTNIE | sdbtnie\_LCL95 | sdbtnie\_UCL95 | sdaTNIE | sdatnie\_LCL95 | sdatnie\_UCL95 | midTNIE | midtnie\_LCL95 | midtnie\_UCL95 |
| 0.43137 | 0.28237 | 0.59367 | -0.037771 | -0.20188 | 0.13973 | 1.05058 | 0.70153 | 1.44814 | 0.39141 | 0.24890 | 0.55355 |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PNDE | PNDE\_LCL95 | PNDE\_UCL95 | sdbPNDE | sdbPNDE\_LCL95 | sdbPNDE\_UCL95 | sdaPNDE | sdaPNDE\_LCL95 | sdaPNDE\_UCL95 | midPNDE | midPNDE\_LCL95 | midPNDE\_UCL95 |
| 0.43367 | 0.21961 | 0.64731 | -0.67400 | -1.04506 | -0.29701 | 1.53425 | 1.14031 | 1.95569 | 0.43214 | 0.22652 | 0.65412 |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TNDE | TNDE\_LCL95 | TNDE\_UCL95 | sdbTNDE | sdbTNDE\_LCL95 | sdbTNDE\_UCL95 | sdaTNDE | sdaTNDE\_LCL95 | sdaTNDE\_UCL95 | midTNDE | midTNDE\_LCL95 | midTNDE\_UCL95 |
| 0.63535 | 0.42385 | 0.85826 | -0.70875 | -1.10866 | -0.35770 | 1.97135 | 1.53994 | 2.40403 | 0.63336 | 0.40613 | 0.86859 |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CDE | CDE\_LCL95 | CDE\_UCL95 | sdbCDE | sdbCDE\_LCL95 | sdbCDE\_UCL95 | sdaCDE | sdaCDE\_LCL95 | sdaCDE\_UCL95 | midCDE | midCDE\_LCL95 | midCDE\_UCL95 |
| 0.39934 | 0.19305 | 0.59977 | -0.53214 | -0.90072 | -0.16062 | 1.32683 | 0.94100 | 1.77297 | 0.39755 | 0.18155 | 0.61528 |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TE | TE\_LCL95 | TE\_UCL95 | sdbTE | sdbTE\_LCL95 | sdbTE\_UCL95 | sdaTE | sdaTE\_LCL95 | sdaTE\_UCL95 | midTE | midTE\_LCL95 | midTE\_UCL95 |
| 0.86503 | 0.63218 | 1.09406 | -0.71177 | -1.10516 | -0.35121 | 2.58483 | 2.10907 | 3.03092 | 0.82355 | 0.57472 | 1.08090 |

**References**

Bradley, J. v. (1978). Robustness? *British Journal of Mathematical and Statistical Psychology*, *31*(2), 144–152. https://doi.org/10.1111/j.2044-8317.1978.tb00581.x

Pearl, J. (2001). *Direct and indirect effects.* Paper presented at the Proceedings of the seventeenth conference on uncertainty in artifical intelligence.

Pearl, J. (2012). The Causal Mediation Formula-A Guide to the Assessment of Pathways and Mechanisms. *Prevention Science*, *13*(4), 426–436. https://doi.org/10.1007/s11121-011-0270-1

Yarar, N., Machiels, C. J. A., & Orth, U. R. (2019). Shaping up: How package shape and consumer body conspire to affect food healthiness evaluation. *Food Quality and Preference*, *75*(March), 209–219. https://doi.org/10.1016/j.foodqual.2019.03.004