

Global sensitivity analysis can unveil the hidden universe of uncertainty in multiverse studies

R code

Alessio Lachi and Arnald Puy

Contents

1	Load required packages	2
2	Read data	3
3	Sample matrix	6
4	Run the model	10
5	Uncertainty analysis	11
6	Sensitivity analysis	14
7	Session information	19
	References	20

1 Load required packages

This code snippet loads all the packages required for the analysis and defines a plotting theme to ensure graphical consistency across all ggplot figures.

```
# LOAD PACKAGES #####  
  
library(sensobol)  
load_packages(c("haven", "tidyverse", "data.table", "margins", "miceadds",  
              "doParallel", "foreach", "cowplot", "benchmarkme", "sandwich",  
              "countrycode"))  
  
# THEME FOR PLOTTING #####  
  
theme_AP <- function() {  
  theme_bw() +  
    theme(panel.grid.major = element_blank(),  
          panel.grid.minor = element_blank(),  
          legend.background = element_rect(fill = "transparent", color = NA),  
          legend.key = element_rect(fill = "transparent", color = NA),  
          strip.background = element_rect(fill = "white"),  
          legend.margin = margin(0.5, 0.1, 0.1, 0.1),  
          legend.box.margin = margin(0.2,-4,-7,-7),  
          plot.margin = margin(3, 4, 0, 4),  
          legend.text = element_text(size = 8),  
          axis.title = element_text(size = 10),  
          legend.key.width = unit(0.4, "cm"),  
          legend.key.height = unit(0.4, "cm"),  
          legend.title = element_text(size = 9))  
}  
#####
```

2 Read data

Here we read in the dataset required for the analysis and turn it into a `data.table` format, and explore the proportion of `NA` values in each column. Then, we transform some columns that require standardization or need to be converted into numeric columns. Finally, in the ‘define variables’ section, we define the variables that will form the uncertainty space of the analysis. Note that each vector represents a specific uncertain parameter, and the elements of the vector are the different levels that will be explored.

```
# READ DATA AND STANDARDIZE #####
#  
data <- readRDS("df.rds")  
setDT(data)  
  
# Columns with migration data -----  
  
cols <- c("foreignpct", "migstock_wb", "migstock_un", "migstock_oecd", "netmigpct")  
  
# Standardize migration data -----  
  
data[, (cols) := lapply(.SD, function(x) as.numeric(scale(as.numeric(x)))), .SDcols = cols]  
  
## Warning in scale(as.numeric(x)): NAs introduced by coercion  
  
## Warning in scale(as.numeric(x)): NAs introduced by coercion  
# TURN DOTS AND EMPTY CELLS INTO NA #####  
  
data[data == "."] <- NA  
data[data == "")] <- NA  
  
# IDENTIFY NUMERIC COLUMNS AND TRANSFORM #####  
  
numeric_columns <- sapply(data, is.numeric)  
numeric_columns <- names(numeric_columns[numeric_columns])  
data[, (numeric_columns):= lapply(.SD, as.numeric), .SDcols = (numeric_columns)]  
data[, iso_country:= as.numeric(iso_country)]  
  
to.numeric <- c("gdp_wb", "gdp_twn", "gni_wb", "gini_wid", "ginim_dolt", "top10_wid",  
    "al_ethnic", "dpi_tf", "wdi_empprilo", "mcp", "mignet_un")  
  
data[, (to.numeric):= lapply(.SD, as.numeric), .SDcols = (to.numeric)]  
  
# IDENTIFY THE PROPORTION OF MISSING VALUES PER COLUMN AND YEAR #####  
  
data[, lapply(.SD, function(x) sum(is.na(x)) / nrow(.SD)), year]  
  
##      year      X country      female      age      age_sq education income  
##      <num> <num> <num>      <num>      <num>      <num>      <num>      <num>
```

```

## 1: 1996      0      0 0.003137372 0.005901247 0.005901247 0.02453873 0.1663928
## 2: 2006      0      0 0.002611440 0.006419789 0.006419789 0.01280331 0.9116826
## 3: 2016      0      0 0.005163590 0.057804789 0.057804789 0.01567355 0.2022482
##   reduce_income_diff  incdiff_c      jobs      jobs_c old_age_care  oldage_c
##                  <num>      <num>      <num>      <num>      <num>      <num>
## 1:      0.06655711 0.06655711 0.04859192 0.04859192 0.02483753 0.02483753
## 2:      0.05766929 0.05766929 0.04667948 0.04667948 0.02368431 0.02368431
## 3:      0.04724913 0.04724913 0.04437032 0.04437032 0.02312192 0.02312192
##   unemployed  unemp_c    housing  housing_c     health  health_c
##              <num>      <num>      <num>      <num>      <num>      <num>
## 1: 0.06439083 0.06439083 0.06278479 0.06278479 0.35362665 0.02696646
## 2: 0.06227558 0.06227558 0.05480396 0.05480396 0.02346669 0.02346669
## 3: 0.05254981 0.05254981 0.05209285 0.05209285 0.02367026 0.02367026
##   iso_country foreignpct netmigpct year_t_minus_1 gdp_oecd    gdp_wb
##              <num>      <num>      <num>      <num>      <num>      <num>
## 1: 0.04108463 0.04108463 0.04108463 0.04108463 0.04108463 0.04108463
## 2: 0.04356026 0.04356026 0.04356026 0.04356026 0.04356026 0.04356026
## 3: 0.06397368 0.06397368 0.06397368 0.06397368 0.06397368 0.06397368
##   gdp_twn     gni_wb ginid_solt ginim_dolt  gini_wb  gini_wid top10_wid
##              <num>      <num>      <num>      <num>      <num>      <num>
## 1: 0.9502502 0.04108463 0.04108463 0.04108463 0.9437888 0.8570628 0.2268992
## 2: 0.9449421 0.04356026 0.04356026 0.04356026 0.4142033 0.8318886 0.1319502
## 3: 0.9364833 0.06397368 0.43684884 0.43684884 0.9364833 1.0000000 1.0000000
##   mcp migstock_wb migstock_un migstock_oecd  mignet_un    pop_wb
##              <num>      <num>      <num>      <num>      <num>      <num>
## 1: 0.2350041 0.04108463 0.04108463 0.3677822 0.04108463 0.04108463
## 2: 0.2457655 0.04356026 0.04356026 0.2537811 0.04356026 0.04356026
## 3: 1.0000000 1.00000000 0.06397368 1.0000000 1.00000000 0.06397368
##   al_ethnic    dpi_tf wdi_empprilo wdi_unempilo socx_oecd  latent6
##              <num>      <num>      <num>      <num>      <num>      <num>
## 1: 0.04108463 0.04108463 0.04108463 0.04108463 0.09710914 0.1514529
## 2: 0.04356026 0.04356026 0.04356026 0.04356026 0.08708426 0.1402561
## 3: 0.06397368 1.00000000 0.06397368 0.06397368 0.24616158 0.1266222
##   latent4
##              <num>
## 1: 0.1514529
## 2: 0.1402561
## 3: 0.1266222

# COUNTRY VARIABLE #####
# DEFINE VARIABLES #####

```

```

country <- unique(data$country)
setDT(data)
data[, country_cl:= countrycode(country, "country.name", "continent")]
data[, country_cl:= ifelse(country_cl == "Oceania" | country_cl == "Asia", "Asia_Oceania",
                           ifelse(country_cl == "Americas", "America", country_cl))]

# DEFINE VARIABLES #####

```

```

# Dependent variable (logit and linear) ----

Y.vector <- c("jobs", "unemployed", "reduce_income_diff", "old_age_care", "housing",
            "health", "jobs_c", "unemp_c", "incdiff_c", "oldage_c", "housing_c",
            "health_c", "latent6", "latent4")

# Policy variables ----

X <- c("foreignpct", "migstock_un", "netmigpct", "migstock_oecd")

# Controls variables ----

controls.ind = c("female", "education", "income", "iso_country", "year_t_minus_1",
                 "al_ethnic", "socx_oecd", "country")

# Controls country ----

controls.country <- c("America - Asia_Oceania", "Asia_Oceania - Europe",
                      "America - Europe", "America - Asia_Oceania - Europe")

# Controls age ----

controls.age <- c("age", "age_sq", "no")

# Controls gini ----

controls.gini <- c("ginid_solt", "ginim_dolt")

# Robust standard errors ----

se.vec <- c("no", "country")

# Yes / no vector ----
yes.no.vec <- c("yes", "no")

```

3 Sample matrix

Here we define the sample matrix representing the uncertainty space of the analysis. Since we also want to conduct a variance-based sensitivity analysis (SA) later on to pinpoint which uncertain parameters convey the most uncertainty to the model output, we define an A, B and AB matrices using the `sensobol` package (Puy et al. 2022). We refer the reader to Puy et al. (2022) and Saltelli et al. (2008) to know more about SA. Then, in the “transform to appropriate columns” section, we transform each column (initially in $\mathcal{U}(0, 1)$) to its appropriate distribution, defined based on the number of levels each column includes. For instance, the column Y represents the `Y.vector` object, which has 14 different levels, and hence it should be distributed between 1 and 14, or $\mathcal{D}(1, 14)$.

The “define the model” section codes the model, which should be tailored to the analysis. The important thing to remember is that the model should run rowwise throughout the sample matrix with the parameter specifications as set in each row. Each row in the sample matrix represents a specific combination of parameter values and a specific model structure.

```
# DEFINE SAMPLE MATRIX #####  
  
# Settings -----  
  
matrices <- c("A", "B", "AB")  
order <- "second"  
N <- 2^13  
params <- c("Y", "X1", controls.ind, "age", "SE", "bootstrap", "GINI")  
  
# Sample matrix -----  
  
mat <- data.table(sobol_matrices(matrices = matrices, params = params, N = N,  
                                order = order))  
  
# To remove -----  
  
to.remove <- "country"  
specific.columns <- setdiff(controls.ind, to.remove)  
  
# Transform to appropriate columns -----  
  
mat[, Y:= floor(Y * 14) + 1]  
mat[, Y:= Y.vector[Y]]  
mat[, X1:= floor(X1 * length(X)) + 1]  
mat[, X1:= X[X1]]  
mat[, (specific.columns):= lapply(.SD, function(x) ifelse(x > 0.5, "yes", "no")), .SDcols = (sp  
mat[, country:= floor(country * length(controls.country)) + 1]  
mat[, country:= controls.country[country]]  
mat[, age:= floor(age * length(controls.age)) + 1]  
mat[, age:= controls.age[age]]  
mat[, SE:= floor(SE * length(se.vec)) + 1]  
mat[, SE:= se.vec[SE]]  
mat[, bootstrap:= floor(bootstrap * length(yes.no.vec)) + 1]
```

```

mat[, bootstrap:= yes.no.vec[bootstrap]]
mat[, GINI:= floor(GINI * length(controls.gini)) + 1]
mat[, GINI:= controls.gini[GINI]]

# DEFINE MODEL #####
# Function to create the glm formula -----
formula_fun <- function(mat) {

  formula = paste0(mat[, Y], " ~ ", mat[,X1], collapse = " ")

  # Conditional on age -----
  formula <- switch(mat[, age],
    "age" = paste0(formula, " + age"),
    "age_sq" = paste0(formula, " + age + age_sq"),
    formula)

  # Add GINI -----
  formula <- paste0(formula, " + ", paste0(mat[, GINI], collapse = " + "))

  # List of additional variables to include if they are marked as "yes" -----
  additional_vars <- c("female", "education", "income", "year_t_minus_1", "iso_country",
    "al_ethnic", "socx_oecd", "country")

  # Search variables marked as "yes" in mat -----
  included_vars <- names(mat[, ..additional_vars])[which(mat[, ..additional_vars] == "yes")]

  # Append included variables to the formula -----
  if (length(included_vars) > 0) {

    formula <- paste0(formula, " + ", paste(included_vars, collapse = " + "))

  }

  return(formula)
}

# Wrap up function ----

full_fun <- function(data, mat, Y.vector) {

```

```

formula <- as.formula(formula_fun(mat = mat))

# Country filtering -----

countries_to_filter <- switch(as.character(mat[, country]),
                               "America - Asia_Oceania" = c("America", "Asia_Oceania"),
                               "Asia_Oceania - Europe" = c("Asia_Oceania", "Europe"),
                               "America - Europe" = c("America", "Europe"),
                               "America - Asia_Oceania - Europe" = c("America", "Asia_Oceania"),
                               NULL)

if (!is.null(countries_to_filter)) {

  data <- data[country_cl %in% countries_to_filter]
}

# Bootstrap -----

if (mat[, bootstrap] == "yes") {

  data <- data[sample(1:nrow(data), nrow(data), replace = TRUE), ]
}

# Determine model family and cluster variable -----

family <- if (mat[, Y] %in% Y.vector[1:6]) "binomial" else "gaussian"

cluster_var <- switch(as.character(mat[, SE]),
                      "no" = NULL,
                      "country" = "country",
                      "year")

# Use only complete cases -----

cols_formula <- all.vars(formula)

if(!is.null(cluster_var)) {

  colnames_selected <- c(cols_formula, cluster_var)

} else {

  colnames_selected <- cols_formula
}

data <- data[, ..colnames_selected][complete.cases(data[, ..colnames_selected])]
```

```

# Execute model ----

if (family == "binomial") {

  if (is.null(cluster_var)) {

    model <- glm(formula, family = binomial, data = data)
    out <- margins(model, variables = mat[, X1])

  } else {

    model <- glm.cluster(data = data[, ..cols_formula], formula = formula,
                          cluster = data[[cluster_var]], family = "binomial")
    out <- margins(model$glm_res, variables = mat[, X1], vcov = model$vcov)

  }

} else {

  if (is.null(cluster_var)) {

    model <- lm(formula = formula, data = data)
    out <- margins(model, variables = mat[, X1])

  } else {

    model <- lm.cluster(data = data[, ..cols_formula], formula = formula,
                          cluster = data[[cluster_var]])
    out <- margins(model$lm_res, variables = mat[, X1], vcov = model$vcov)

  }

}

out <- as.numeric(summary(out)[c("AME", "lower", "upper")])

return(out)
}

```

4 Run the model

This code snippet defines a parallel backend to conduct the simulations in parallel, and executes the model rowwise throughout the sample matrix defined in the previous step. Once the simulations are completed, we extract only the A matrix (to conduct an uncertainty analysis later on) and export the simulations in .csv.

```
# PARALLEL COMPUTING #####
# Define parallel computing -----
cl <- makeCluster(floor(detectCores() * 0.75))
registerDoParallel(cl)

# Run the simulations -----
out <- foreach(i = 1:nrow(mat),
               .packages = c("margins", "miceadds", "data.table"),
               .combine = "rbind") %dopar% {
  full_fun(data = data, mat = mat[i, ], Y.vector = Y.vector)
}

# Stop the parallel backend -----
stopCluster(cl)

# ARRANGE DATA #####
out.dt <- data.table(out) %>%
  setnames(., colnames(.), c("AME", "low", "high"))

AME.dt <- out.dt[1:(2 * N), ]

# EXPORT DATA #####
full.dt <- cbind(mat, out.dt)
fwrite(full.dt, "full.dt.csv")
```

5 Uncertainty analysis

Here we do some data cleaning and adding of new color columns for better plotting later on. We also calculate the 2.5 and 97.5 quantiles of the output distribution, and plot the output following the layout defined in Breznau et al. (2022). Finally, we plot a scatterplot of each uncertain parameter against the model output, a standard practice in SA that often helps to visualize, in a swift way, possible influential parameters and the presence of interactions.

```
# PLOT UNCERTAINTY #####
tmp <- out.dt[order(AME.dt)] %>%
  .[, Model := .I] %>%
  .[, color := ifelse(low < 0 & high < 0, "negative",
                      ifelse(low < 0 & high > 0, "includes zero", "positive"))] %>%
  .[, color := factor(color, levels = c("negative", "includes zero", "positive"))]

tmp[, .N, color] %>%
  .[, total.n := nrow(tmp)] %>%
  .[, fraction := N / total.n] %>%
  print()

##          color      N total.n    fraction
##          <fctr> <int>    <int>      <num>
## 1:    negative 10633   16384 0.64898682
## 2:        <NA>     216   16384 0.01318359
## 3: includes zero  3443   16384 0.21014404
## 4:    positive   2092   16384 0.12768555

# Calculate 2.5% and 97.5% quantiles ----

quantiles <- quantile(AME.dt$AME, c(0.025, 0.975))
quantiles

##      2.5%      97.5%
## -0.254461  0.198296

plot.uncertainty <- tmp %>%
  na.omit() %>%
  ggplot(., aes(x = Model, y = AME, color = color)) +
  geom_point(size = 0.3) +
  geom_errorbar(aes(ymin = low, ymax = high), size = 0.5) +
  scale_color_manual(values = c("orange", "grey", "blue"), name = "") +
  geom_hline(yintercept = 0, lty = 2) +
  labs(x = "Models Ordered by AME", y = "AME") +
  theme_AP() +
  theme(legend.position = "top",
        axis.title.x = element_text(size = 8),
        axis.text.x = element_text(size = 6.5))

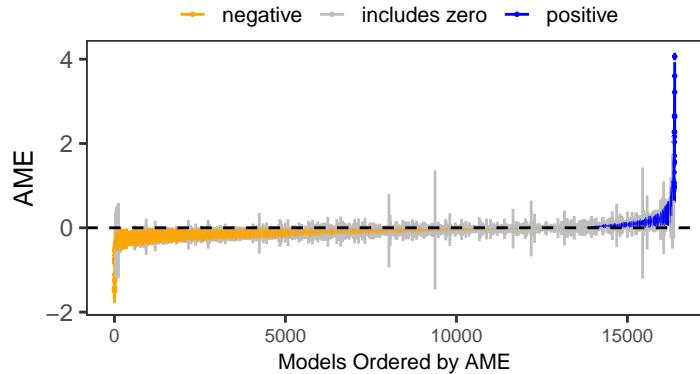
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
```

```

## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

plot.uncertainty

```



```

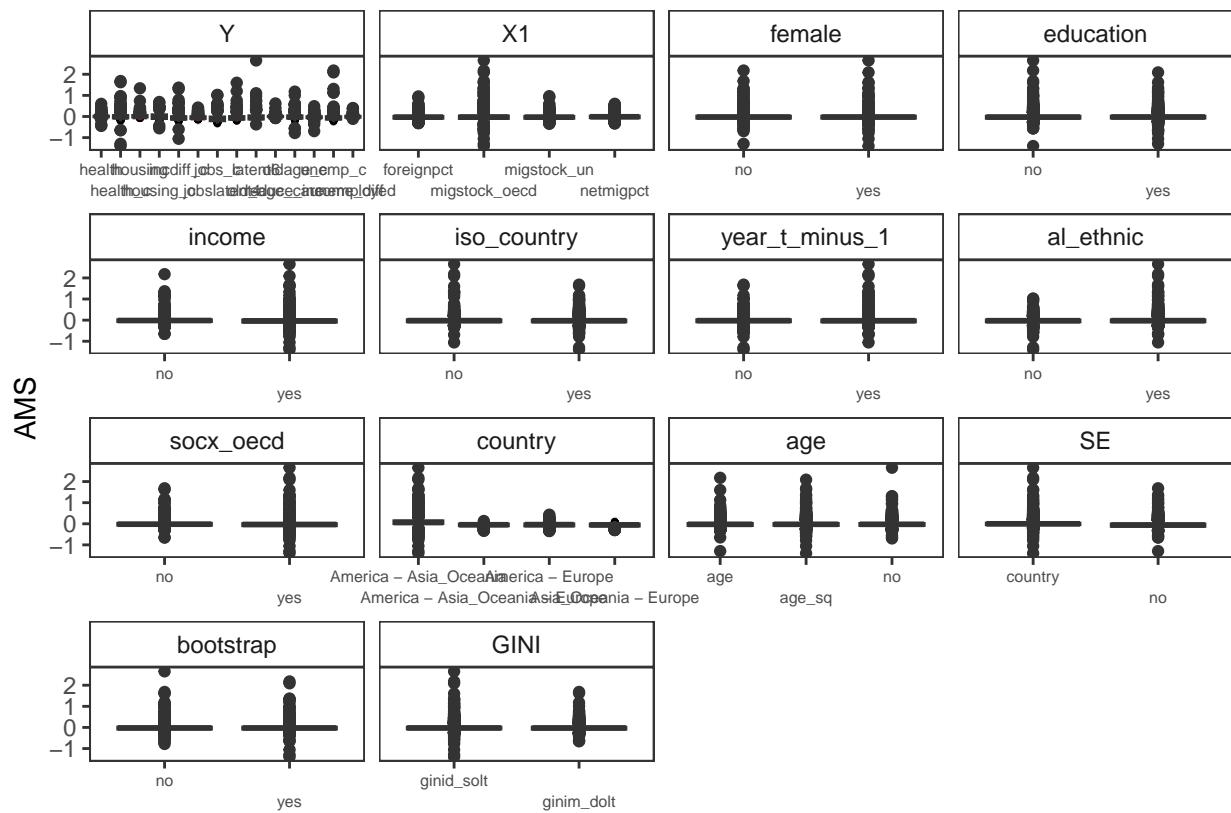
# PLOT SCATTERPLOT #####
plot_scatter(data = mat, N = N, Y = out, params = params) +
  geom_boxplot() +
  theme(axis.text.x = element_text(size = 6)) +
  labs(x = "", y = "AMS") +
  scale_x_discrete(guide = guide_axis(n.dodge = 2))

## Warning: Removed 1568 rows containing non-finite outside the scale range
## (`stat_summary_bin()`).

## Warning: Removed 1568 rows containing non-finite outside the scale range
## (`stat_boxplot()`).

## Warning: Removed 1568 rows containing missing values or values outside the scale range
## (`geom_point()`).

```



6 Sensitivity analysis

Here we conduct a variance-based SA to know which parameters are responsible for the most uncertainty in the final model estimation. We again rely on the workflow defined in `sensobol` (Puy et al. 2022). We use the saltelli and the jansen estimators to estimate first and total-order indices respectively. See Puy et al. (2022) for other possible options in the selection of estimators. We finally plot the Sobol' indices and export the resulting data.

```
# SENSITIVITY ANALYSIS #####
# Define setting -----
first <- "saltelli"
total <- "jansen"
R <- 10^3

# Modify Y for y in the params vector for notation consistency -----
params[1] <- "y"

# Run SA -----
ind <- sobol_indices(matrices = matrices, params = params, N = N, Y = out.dt$AME,
boot = TRUE, R = R, first = first, total = total, order = order)

ind

##
## First-order estimator: saltelli | Total-order estimator: jansen
##
## Total number of model runs: 876544
##
## Sum of first order indices: 0.3586659
##      original      bias std.error     low.ci     high.ci
##      <num>      <num>    <num>      <num>      <num>
##  1:  0.0750353990 1.341904e-03 0.036109054  0.002921050 0.144465939
##  2:  0.0069654961 8.891063e-04 0.016941268 -0.027127886 0.039280666
##  3: -0.0006153313 -2.790831e-05 0.001713501 -0.003945824 0.002770978
##  4:  0.0002372609 1.118523e-04 0.002367474 -0.004514756 0.004765573
##  5:  0.0059921499 1.383670e-04 0.008540752 -0.010885783 0.022593349
##  ---
## 115: -0.0004827168 1.637230e-05 0.003733946 -0.007817488 0.006819310
## 116: -0.0014278684 3.891521e-05 0.002720994 -0.006799833 0.003866266
## 117:  0.0010945122 2.273352e-05 0.003025458 -0.004858010 0.007001567
## 118: -0.0014603753 2.541833e-05 0.002071488 -0.005545836 0.002574249
## 119:  0.0019034463 4.632741e-05 0.002098185 -0.002255249 0.005969487
##      sensitivity      parameters
##      <char>          <char>
##  1:           Si            y
```

```

## 2:      Si      X1
## 3:      Si      female
## 4:      Si      education
## 5:      Si      income
## ---
## 115:    Sij  age.bootstrap
## 116:    Sij  age.GINI
## 117:    Sij  SE.bootstrap
## 118:    Sij  SE.GINI
## 119:    Sij bootstrap.GINI

# Calculate sum of first and total-order indices -----
# (only indices >0.05 are considered)

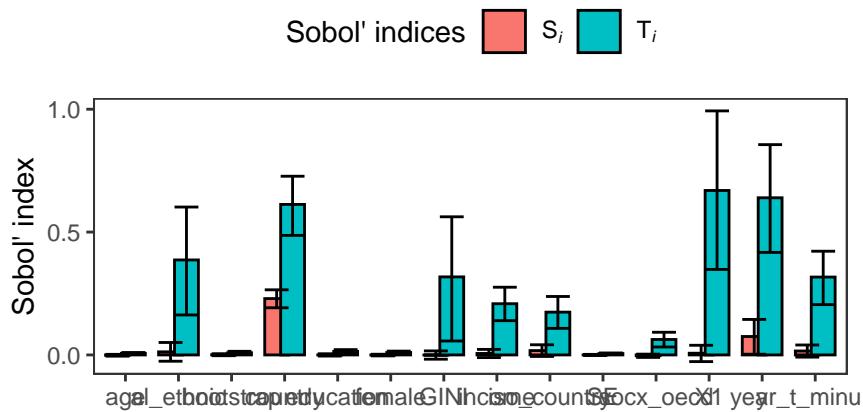
ind$results[sensitivity %in% c("Si", "Sij")] %>%
  .[original > 0.05, sum(original)]

```

```
## [1] 0.4003714
```

```
# Plot Sobol' indices -----
```

```
plot(ind)
```

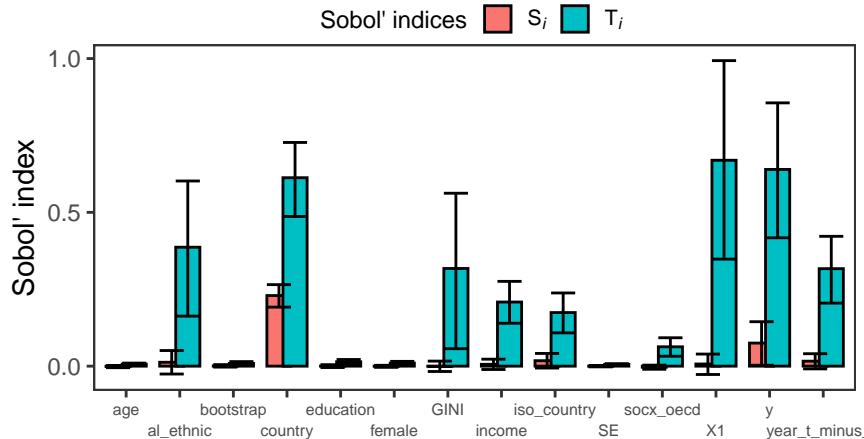


```

plot.sobel <- plot(ind) +
  scale_x_discrete(guide = guide_axis(n.dodge = 2)) +
  theme(legend.position = "top",
        legend.margin = margin(0.5, 0.1, 0.1, 0.1),
        legend.box.margin = margin(0.2, -4, -7, -7),
        plot.margin = margin(3, 4, 0, 4),
        legend.text = element_text(size = 8),
        legend.key.width = unit(0.4, "cm"),
        legend.key.height = unit(0.4, "cm"),
        legend.title = element_text(size = 9),
        axis.text.x = element_text(size = 6.2))

```

```
plot.sobel
```



```

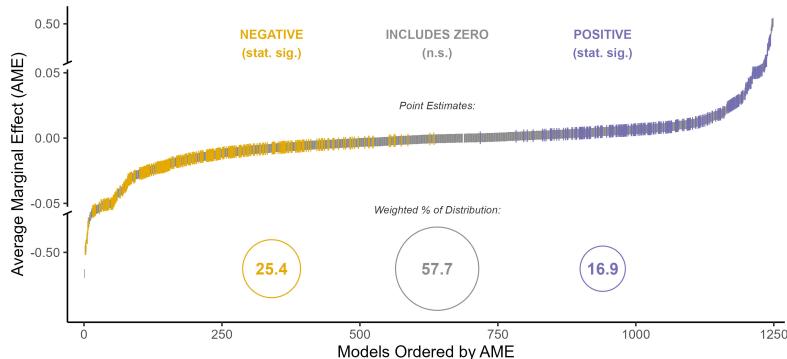
plot.second <- plot(ind, order = order) +
  theme(axis.text.x = element_text(size = 6.5))

# Export sensitivity indices -----
fwrite(ind$results, "ind.csv")

# ADD ORIGINAL BREZNAU ET AL PLOT #####
original.plot<- cowplot::ggdraw() + cowplot::draw_image("pnas.2203150119fig01.png", scale = 0.9)

original.plot

```

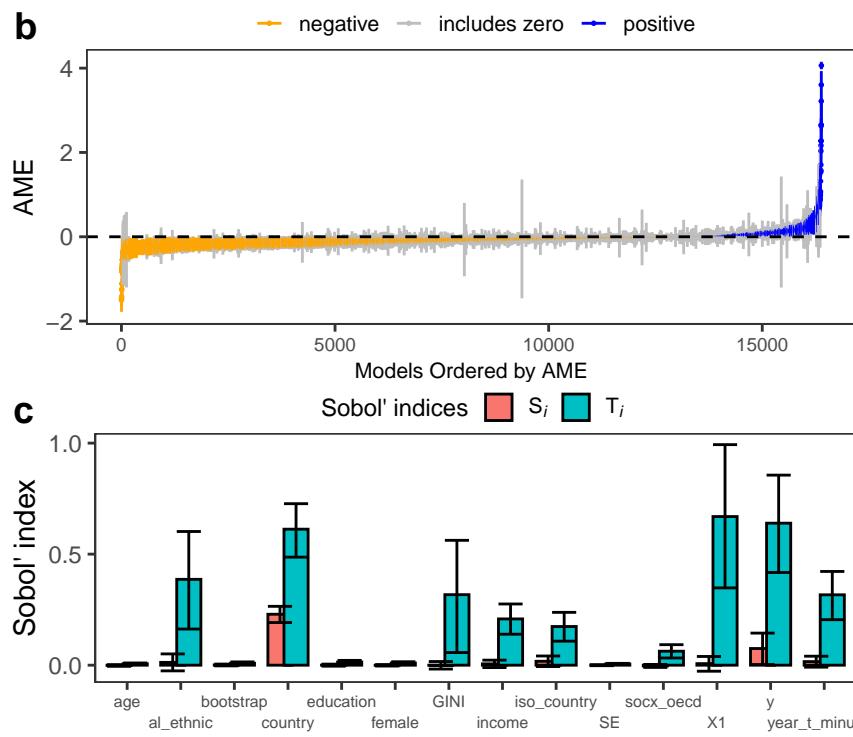


```

p2 <- plot_grid(plot.uncertainty, plot.sobel, ncol = 1, labels = c("b", "c"))

p2

```



```
# MERGE ALL PLOTS #####
plot_grid(original.plot, p2, plot.second, ncol = 1, labels = c("a", "", "d"),
          rel_heights = c(0.37, 0.63))
```

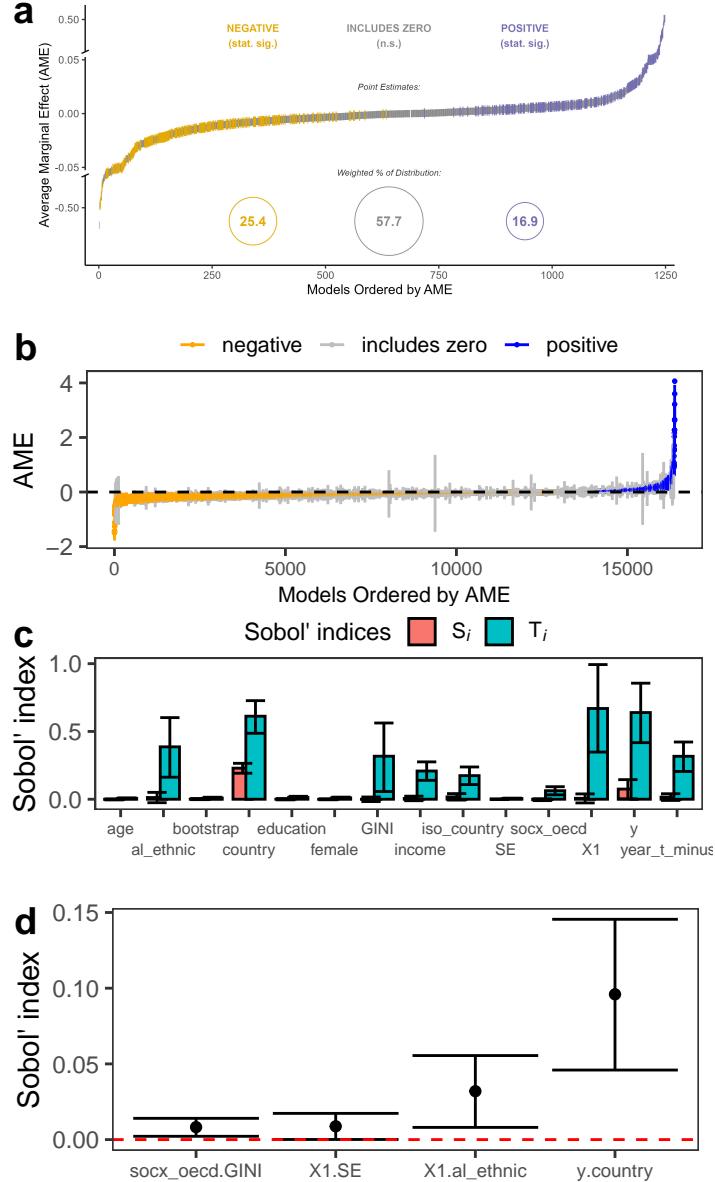


Figure 1: Global sensitivity analysis. a) Distribution of estimated Average Marginal Effects (AME) in Brezna et al. (2022). Reproduced with permission. b) Distribution of AME after a global sensitivity analysis. c) Sobol' first (S_i) and total-order (T_i) effects. d) Second-order (S_{ij}) effects. All error bars show the 95% confidence intervals after bootstrapping the Sobol' indices using the normal method. The red, horizontal dashed line is the approximation error threshold and is at 0.05. We only show pairwise interactions whose contribution to the output variance is higher than the approximation error.

7 Session information

```
# SESSION INFORMATION #####
sessionInfo()

## R version 4.3.3 (2024-02-29)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Sonoma 14.2.1
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRlapack.dylib;
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
## 
## time zone: Europe/London
## tzcode source: internal
##
## attached base packages:
## [1] parallel stats      graphics grDevices utils      datasets methods
## [8] base
##
## other attached packages:
## [1] countrycode_1.5.0 sandwich_3.0-2   benchmarkme_1.0.8 cowplot_1.1.3
## [5] doParallel_1.0.17 iterators_1.0.14 foreach_1.5.2       miceadds_3.17-44
## [9] mice_3.15.0        margins_0.3.27    data.table_1.14.99 lubridate_1.9.2
## [13]forcats_1.0.0      stringr_1.5.1     dplyr_1.1.4       purrr_1.0.2
## [17]readr_2.1.4        tidyverse_1.3.1   tibble_3.2.1      ggplot2_3.5.1
## [21]tidyverse_2.0.0     haven_2.5.2      sensobol_1.1.5
##
## loaded via a namespace (and not attached):
## [1] benchmarkmeData_1.0.4 gtable_0.3.5      prediction_0.3.18
## [4] xfun_0.39            lattice_0.22-5    tzdb_0.3.0
## [7] vctrs_0.6.5          tools_4.3.3       Rdpack_2.6
## [10]generics_0.1.3       fansi_1.0.6       highr_0.10
## [13]pkgconfig_2.0.3      Matrix_1.6-5     filehash_2.4-5
## [16]lifecycle_1.0.4      farver_2.1.2     compiler_4.3.3
## [19]munsell_0.5.1        mitools_2.4      codetools_0.2-19
## [22]htmltools_0.5.5     yaml_2.3.7       crayon_1.5.2
## [25]pillar_1.9.0         MASS_7.3-60.0.1   randtoolbox_2.0.4
## [28]magick_2.7.4          boot_1.3-29      tidyselect_1.2.0
## [31]digest_0.6.34        stringi_1.8.3    labeling_0.4.3
## [34]fastmap_1.1.1        grid_4.3.3       colorspace_2.1-0
## [37]cli_3.6.3            magrittr_2.0.3    Rfast_2.0.7
## [40]utf8_1.2.4           broom_1.0.5      withr_3.0.0
## [43]scales_1.3.0          backports_1.4.1   RcppZiggurat_0.1.6
```

```

## [46] timechange_0.2.0      rmarkdown_2.21       httr_1.4.5
## [49] rngWELL_0.10-9        zoo_1.8-12        hms_1.1.3
## [52] evaluate_0.20         knitr_1.42        rbibutils_2.2.16
## [55] rlang_1.1.4           Rcpp_1.0.12        glue_1.7.0
## [58] DBI_1.1.3             tikzDevice_0.12.4 rstudioapi_0.15.0
## [61] R6_2.5.1

## Return the machine CPU
cat("Machine:      "); print(get_cpu()$model_name)

## Machine:
## [1] "Apple M1 Max"

## Return number of true cores
cat("Num cores:   "); print(detectCores(logical = FALSE))

## Num cores:
## [1] 10

## Return number of threads
cat("Num threads: "); print(detectCores(logical = FALSE))

## Num threads:
## [1] 10

```

References

- Breznau, Nate, Eike Mark Rinke, Alexander Wuttke, Hung H. V. Nguyen, Muna Adem, Jule Adriaans, Amalia Alvarez-Benjumea, et al. 2022. “Observing Many Researchers Using the Same Data and Hypothesis Reveals a Hidden Universe of Uncertainty.” *Proceedings of the National Academy of Sciences* 119 (44): e2203150119. <https://doi.org/10.1073/pnas.2203150119>.
- Puy, Arnald, Samuele Lo Piano, Andrea Saltelli, and Simon A. Levin. 2022. “Sensobol: An R Package to Compute Variance-Based Sensitivity Indices.” *Journal of Statistical Software* 102 (5): 1–37. <https://doi.org/10.18637/jss.v102.i05>.
- Saltelli, Andrea, Marco Ratto, Terry Andres, Francesca Campolongo, Jessica Cariboni, Debora Gatelli, Michaela Saisana, and Stefano Tarantola. 2008. *Global Sensitivity Analysis. The Primer*. Chichester, UK: John Wiley & Sons, Ltd. <https://doi.org/10.1002/9780470725184>.