A sensitivity analysis of the PAWN sensitivity index

Arnald Puy, Samuele Lo Piano, Andrea Saltelli

Contents

1 Preliminary functions

```r
# PRELIMINARY FUNCTIONS ---------------------------------------------------

# Install the development version of the pawnr package
devtools::install_github("arnaldpuy/pawnr", build_vignettes = TRUE)

# Function to read in all required packages in one go:
loadPackages <- function(x) {
  for(i in x) {
    if(!require(i, character.only = TRUE)) {
      install.packages(i, dependencies = TRUE)
      library(i, character.only = TRUE)
    }
  }
}

# Load the packages
loadPackages(c("tidyverse", "data.table", "randtoolbox", "sensitivity",
               "boot", "parallel", "doParallel", "scales", "cowplot",
               "overlapping", "pawnr", "sensobol", "sensitivity", "wesanderson"))

# Set checkpoint

dir.create(".checkpoint")
library("checkpoint")

checkpoint("2019-09-22",
           R.version ="3.6.1",
           checkpointLocation = getwd())
```

2 Check convergence of Sobol' indices and PAWN

```r
# DEFINE SETTINGS ---------------------------------------------------------

N <- seq(500, 10000, 250) # Sample sizes
n <- 15 # Number of conditioning intervals
k <- c(2, 3, 8, 20) # Vector with number of model inputs
R <- 100 # Bootstrap replicas
n_cores <- floor(detectCores() * 0.75) # Use 75% of the cores available
type <- "norm" # Define the confidence interval method
conf <- 0.95 # Define the ci
models <- c("Liu", "Ishigami", "Sobol' G", "Morris")
params <- lapply(k, function(x) paste("X", 1:x, sep = ""))
names(params) <- models
```

```r
# Function to compute the Liu et al. function
liu <- function(X1, X2) {
  X1 / X2
}

liu_Mapply <- function(X) {
  X[, 1] <- qchisq(X[, 1], df = 10)
  X[, 2] <- qchisq(X[, 2], df = 13.978)
  return(mapply(liu, X[, 1], X[, 2]))
}
```

2.1 Sample matrix

```r
# CONSTRUCT SAMPLE MATRICES -----------------------------------------------

A <- list()
for(i in k) {
  A[[i]] <-mclapply(N, function(N) sobol_matrices(n = N, k = i), mc.cores = n_cores)
}

A <- A[!sapply(A, is.null)]
names(A) <- models

for(i in names(A)) {
  names(A[[i]]) <- N
}
```

2.2 Model output

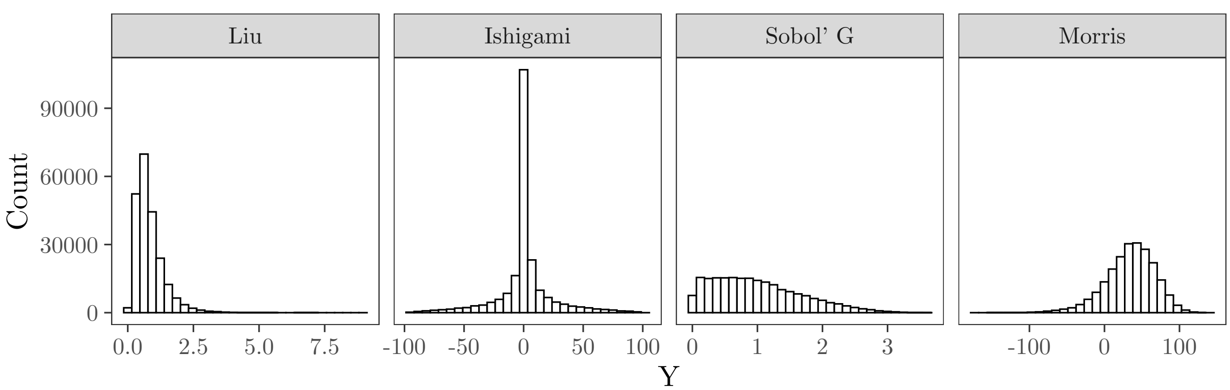
```r
# COMPUTE MODEL OUTPUT ----------------------------------------------------

Y <- list()
for(i in names(A)) {
  if(i == "Liu") {
    Y[[i]] <- lapply(A[[i]], function(x) liu_Mapply(x))
  } else if(i == "Ishigami") {
    Y[[i]] <- lapply(A[[i]], function(x) sensobol::ishigami_Mapply(x))
  } else if(i == "Sobol' G") {
    Y[[i]] <- lapply(A[[i]], function(x) sensobol::sobol_Fun(x))
  } else {
    Y[[i]] <- lapply(A[[i]], function(x) sensitivity::morris.fun(x))
  }
}

names(Y) <- models
for(i in names(Y)) {
  names(Y[[i]]) <- N
```

```
}
```

```
# PLOT MODEL UNCERTAINTY ----------------------------------------------------

lapply(models, function(models) Y[[models]]$`10000`) %>%
  do.call(cbind, .) %>%
  data.table() %>%
  setnames(., 1:4, models) %>%
  melt(., measure.vars = 1:4) %>%
  .[, variable:= factor(variable, levels = models)] %>%
  ggplot(., aes(value)) +
  geom_histogram(color = "black",
                 fill = "white") +
  labs(x = "Y",
       y = "Count") +
  facet_wrap(~ variable,
             scales = "free_x",
             ncol = 4) +
  theme_bw() +
  theme(aspect.ratio = 1,
        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))
```

```
## Warning in (function (..., deparse.level = 1) : number of rows of result is
## not a multiple of vector length (arg 1)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



2.3 Sobol' indices

```
# COMPUTE SOBOL' INDICES AND THEIR CONFIDENCE INTERVALS -----------------------

out <- out.ci <- list()
```

```r
for(i in names(A)) {
  for(j in names(A[[i]])) {
    out[[i]][[j]] <- sobol_indices(Y[[i]][[j]],
                                   params = params[[i]],
                                   n = as.numeric(j),
                                   type = "saltelli",
                                   R = R,
                                   parallel = "multicore",
                                   ncpus = n_cores)
    out.ci[[i]][[j]] <- sobol_ci(out[[i]][[j]],
                                 params = params[[i]],
                                 type = type,
                                 conf = conf)
  }
}
```

```r
# SOBOL INDICES AND CONFIDENCE INTERVALS OF DUMMY PARAMETER -------------------

sobol.dummy <- sobol.dummy.ci <- list()
for(i in names(A)) {
  for(j in names(A[[i]])) {
    sobol.dummy[[i]][[j]] <- sobol_dummy(Y[[i]][[j]],
                                         params = params[[i]],
                                         R = R,
                                         n = as.numeric(j),
                                         parallel = "multicore",
                                         ncpus = n_cores)
    sobol.dummy.ci[[i]][[j]] <- sobol_ci_dummy(sobol.dummy[[i]][[j]],
                                               type = type,
                                               conf = conf)
  }
}

sobol.dummy.final <- lapply(sobol.dummy.ci, function(x) rbindlist(x, idcol = "N")) %>%
  rbindlist(., idcol = "model") %>%
  .[, model:= factor(model, levels = c("Liu", "Ishigami",
                                       "Sobol' G", "Morris"))]
```

```r
# SOBOL' CONVERGENCE ---------------------------------------------------------

sobol.convergence <- lapply(out.ci, function(x) rbindlist(x, idcol = "N")) %>%
  rbindlist(., idcol = "model") %>%
  .[, N:= as.numeric(N)] %>%
  .[, diff:= high.ci - low.ci] %>%
  .[, model:= factor(model, levels = c("Liu", "Ishigami",
                                       "Sobol' G", "Morris"))] %>%
  .[, parameters:= factor(parameters,
                          levels = paste("X", 1:20, sep = ""))] %>%
```

```
  .[, method:= "Sobol' $S_{Ti}$"] %>%
  .[, .(model, N, parameters, original, low.ci, high.ci, diff, method, sensitivity)]
```

2.4 PAWN

```
# COMPUTE PAWN INDICES AND THEIR CONFIDENCE INTERVALS -------------------------

# Subset to take only the A matrix and the model output of the A matrix
Y.pawn <- A.pawn <- list()
for(i in names(Y)) {
  for(j in names(Y[[i]])) {
    Y.pawn[[i]][[j]] <- Y[[i]][[j]][1:j]
    A.pawn[[i]][[j]] <- A[[i]][[j]][1:j, ]
  }
}

# Compute PAWN indices and their confidence intervals
pawn.indices <- pawn.ci <- list()
for(i in names(A.pawn)) {
  for(j in names(A.pawn[[i]]) ) {
    pawn.indices[[i]][[j]] <- pawn_generic(data = A.pawn[[i]][[j]],
                                           Y = Y.pawn[[i]][[j]],
                                           n = n,
                                           test = median,
                                           R = R)
    pawn.ci[[i]][[j]] <- pawn_ci(pawn.indices[[i]][[j]])
  }
}
```

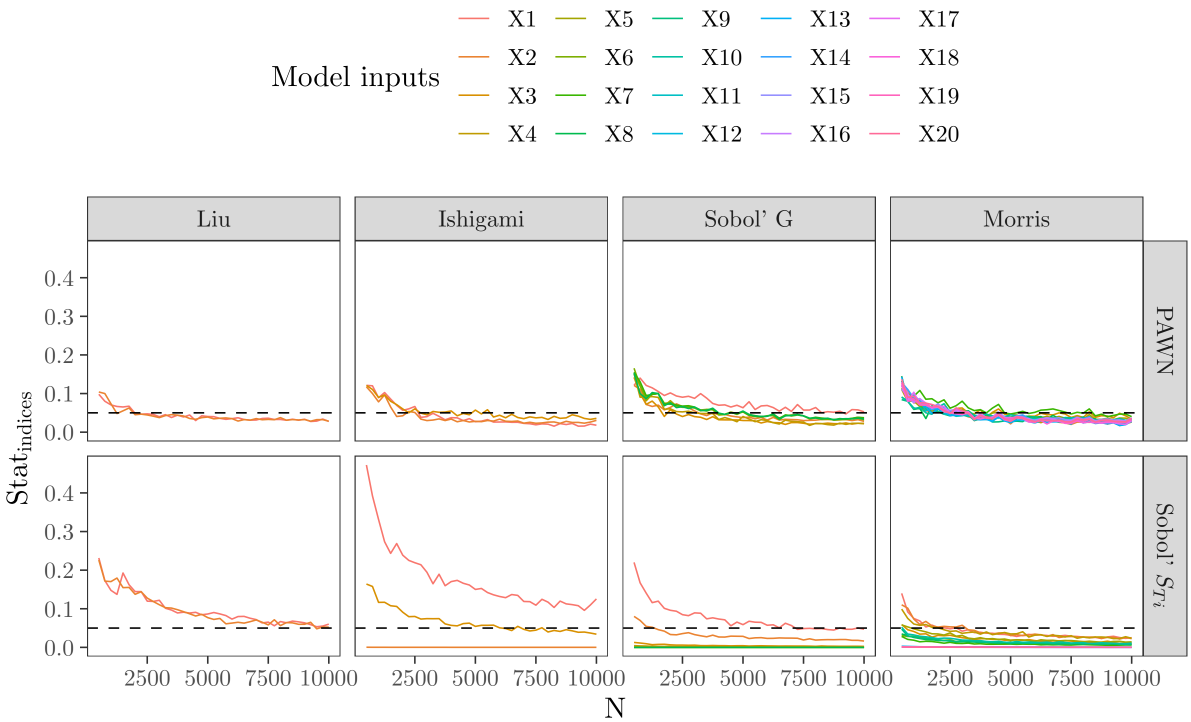
```
# PAWN AND CONFIDENCE INTERVALS OF DUMMY PARAMETER ----------------------------

pawn.index.dummy <- list()
for(i in names(Y)) {
  for(j in names(Y[[i]]) ) {
    pawn.index.dummy[[i]][[j]] <- pawn_dummy(Y = Y[[i]][[j]],
                                             n = n,
                                             R = R)
  }
}

pawn.index.dummy <- lapply(pawn.index.dummy, function(x) rbindlist(x, idcol = "N")) %>%
  rbindlist(., idcol = "model") %>%
  .[, model:= factor(model, levels = c(c("Liu", "Ishigami",
                                         "Sobol' G", "Morris")))]
```

```
# PAWN CONVERGENCE ------------------------------------------------------------

pawn.convergence <- lapply(pawn.ci, function(x) rbindlist(x, idcol = "N")) %>%
```

```r
  rbindlist(., idcol = "model") %>%
  .[, N:= as.numeric(N)] %>%
  .[, diff:= high.ci - low.ci] %>%
  .[, model:= factor(model, levels = c("Liu", "Ishigami",
                                         "Sobol' G", "Morris"))] %>%
  .[, parameters:= gsub("V", "X", parameters)] %>%
  .[, parameters:= factor(parameters,
                           levels = paste("X", 1:20, sep = ""))] %>%
  .[, method:= "PAWN"]
```

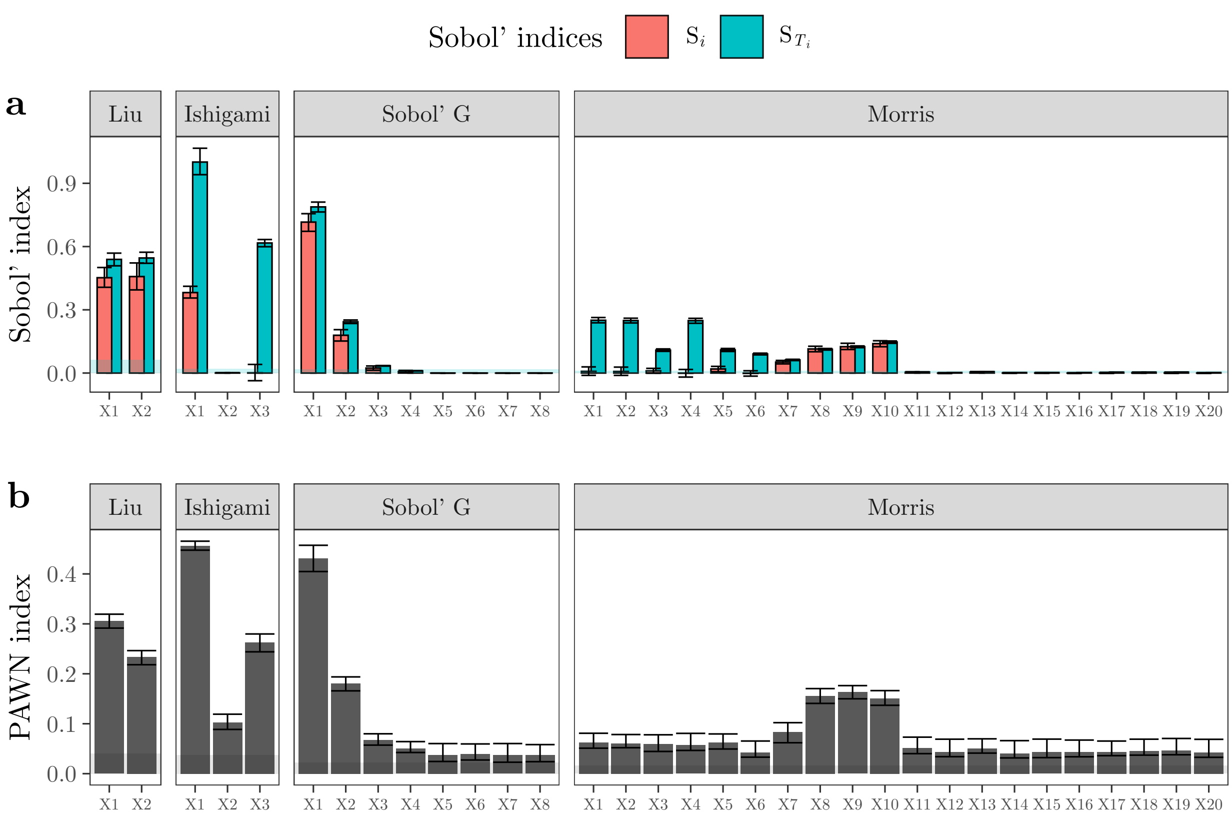
2.5 Plot convergence

```r
# PLOT CONVERGENCE ------------------------------------------------------------

sobol.convergence[sensitivity == "STi"] %>%
  .[, sensitivity:= NULL] %>%
  rbind(., pawn.convergence) %>%
  ggplot(., aes(N, diff,
                group = parameters,
                color = parameters)) +
  geom_line() +
  geom_hline(yintercept = 0.05,
             lty = 2) +
  scale_color_discrete(name = "Model inputs") +
  labs(y = expression(Stat[indices]),
       x = "N") +
  facet_grid(method~model) +
  theme_bw() +
  theme(legend.position = "top",
        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))
```

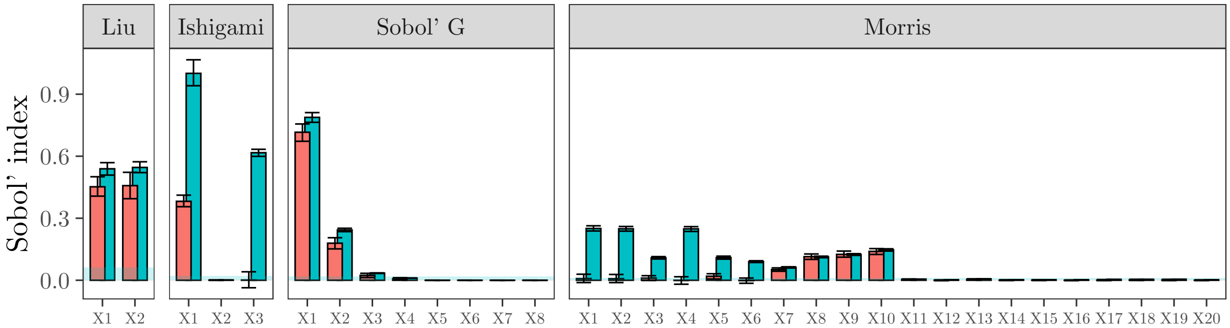



```
# PLOT SOBOL' AND PAWN INDICES ----------------------------------------------

# Sobol' indices
a <- plot_sobol(sobol.convergence[N==max(N)],
                dummy = sobol.dummy.final[N==max(N)]) +
  facet_grid(~model,
             scales = "free_x",
             space = "free_x") +
  labs(x = "",
       y = "Sobol' index") +
  theme(axis.text.x = element_text(size = 6),
        legend.position = "none")

# Get legend
legend <- get_legend(a + theme(legend.position = "top"))


# PAWN indices
b <- pawn.convergence[N==max(N)] %>%
  plot_pawn(.) +
  geom_rect(data = pawn.index.dummy[N==max(N)],
            aes(ymin = 0,
                ymax = high.ci,
                xmin = -Inf,
                xmax = Inf),
            fill = "black",
            alpha = 0.1,
```

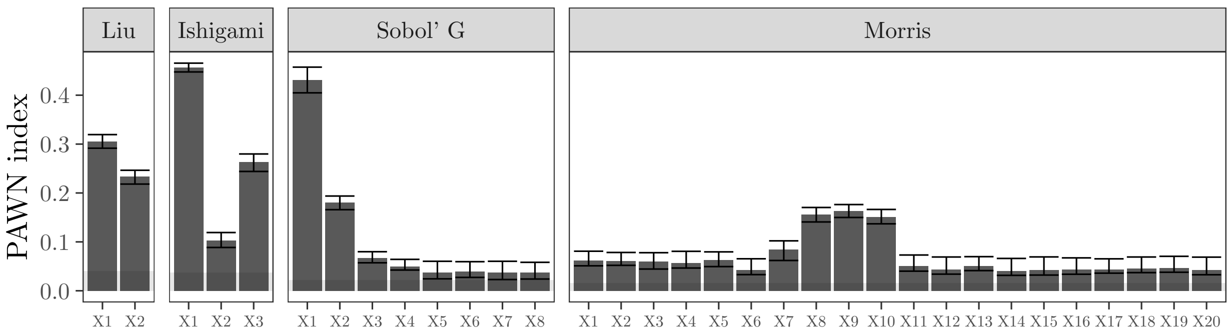
```
             inherit.aes = FALSE) +
  labs(x = "",
       y = "PAWN index") +
  facet_grid(~ model,
             scales = "free_x",
             space = "free_x") +
  theme(axis.text.x = element_text(size = 6))

# Merge
bottom <- plot_grid(a, b,
                    ncol = 1,
                    labels = "auto",
                    align = "h")

plot_grid(legend, bottom,
          labels = c("", ""),
          ncol = 1,
          align = "",
          rel_heights = c(0.1, 1))
```




```
# PLOT SOBOL' AND PAWN INDICES (INDIVIDUAL PLOTS) -----------------------------

a
```




```
b
```



3 Sensitivity of PAWN to its design parameters

3.1 The model

```
# THE MODEL -------------------------------------------------------------------

# Function to divide a vector into chunks
chunks <- function(x,n) split(x, cut(seq_along(x), n, labels = FALSE))

# The model
model_pawn <- function(Model, N, n, epsilon, theta) {
  # Check which model to apply to set the number of
  # parameters
  if(Model == 1) {
    k <- 2
  } else if(Model == 2) {
    k <- 3
  } else if(Model == 3) {
    k <- 8
  } else {
    k <- 20
  }
  # Create the Sobol' matrix
  data <- randtoolbox::sobol(n = N, dim = k)
  # Transform distribution:
  if(Model == 1) {
```

```r
    ModelRun <- liu_Mapply
  } else if(Model == 2) {
    ModelRun <- sensobol::ishigami_Mapply
  } else if(Model == 3) {
    ModelRun <- sensobol::sobol_Fun
  } else {
    ModelRun <- sensitivity::morris.fun
  }
  # Run the model
  Y <- ModelRun(data)
  # Set seed to fix the random number generator
  # for the sample function below
  set.seed(epsilon)
  # Sample the unconditional model output
  index <- sample(1:nrow(data),
                  size = floor(nrow(data) / n),
                  # Without replacement
                  replace = FALSE)
  # Bind model inputs and model output
  dt <- data.table::data.table(cbind(data, Y))
  # Subset and obtain the unconditional model output
  Y_unc <- dt[index, Y]
  # Create the intervals
  melted <- data.table::melt(dt,
                             measure.vars = 1:(ncol(dt) - 1),
                             variable.name = "parameters")
  # Compute PAWN indices
  out <- melted[order(parameters, value)][
    , .(chunks(Y, n)), parameters][
    , Y_unc:= .(rep(.(Y_unc), times = n * ncol(data)))][
    , ID:= .I][
    , results:= .(.(mapply(stats::ks.test, Y_unc, V1))), ID][
    , statistic:= sapply(results, function(x) x[, 1]$statistic)]
  if(theta == 1) {
    final <-  out[, mean(statistic), parameters][, V1]
  } else if(theta == 2) {
    final <-  out[, median(statistic), parameters][, V1]
  } else {
    final <-  out[, max(statistic), parameters][, V1]
  }
  return(final)
}
```

3.2 Settings

```r
# DEFINE SETTINGS ------------------------------------------------------------
```

```
# Set sample size
n <- 2 ^ 12

# Define N.min and N.max
N.min <- 200
N.max <- 2000

# Set parameters
parameters <- c("N", "n", "epsilon", "theta")

# Vector with name of functions
models <- c("Liu", "Ishigami", "Sobol' G", "Morris")
```

3.3 Sample matrix

```
# CREATION OF THE MATRICES ---------------------------------------------------

# Create the A, B and AB matrices, also for the
# computation of second and third-order indices
tmp <- lapply(1:4, function(x)
  sobol_matrices(n = n,
                 k = length(parameters),
                 second = TRUE,
                 third = TRUE) %>%
    data.table())

# Name the slots
names(tmp) <- 1:4

# Rename columns
tmp <- lapply(tmp, setnames, parameters) %>%
  rbindlist(., idcol = "Model")

# Create two copies of the sample matrix and list the
# original and the copies. One would be to run the
# calculations in the max in theta setting; the
# other one for the max not in theta setting,
# and the other in the optimum setting
max <- copy(tmp)
A <- list(tmp, max, copy(tmp))

# Name the slots
names(A) <- c("max", "no.max", "optimum")

# Transform all distributions
for(i in names(A)) {
  if(i == "max") {
```

```
    # where 1=mean, 2=median, 3=max in the model
    A[[i]][, N:= floor(qunif(N, N.min, N.max))]
    A[[i]][, n:= floor(qunif(n, 5, 20))]
    A[[i]][, theta:= floor(theta * (3 - 1 + 1)) + 1]
  } else if(i == "no.max") {
    A[[i]][, N:= floor(qunif(N, N.min, N.max))]
    A[[i]][, n:= floor(qunif(n, 5, 20))]
    A[[i]][, theta:= floor(theta * (2 - 1 + 1)) + 1]
  } else {
    A[[i]][, N:= floor(qunif(N, N.max, 4000))]
    A[[i]][, n:= floor(qunif(n, 15, 20))]
    A[[i]][, theta:= floor(theta * (2 - 1 + 1)) + 1]
  }
}

# Transform all the other distributions
A.pawn <- rbindlist(A, idcol = "setting")[
  , epsilon:= floor(qunif(epsilon, 1, 1000))][
  , Model:= as.numeric(Model)]

print(A.pawn)
```

```
##         setting Model    N  n epsilon theta
##      1:     max     1 1100 12     500     2
##      2:     max     1 1550  8     750     1
##      3:     max     1  650 16     250     3
##      4:     max     1  875 10     625     1
##      5:     max     1 1775 18     125     2
##     ---
## 786428: optimum     4 2500 16     864     2
## 786429: optimum     4 3500 18     364     1
## 786430: optimum     4 3000 17     114     2
## 786431: optimum     4 2000 19     614     1
## 786432: optimum     4 2000 17     372     1
```

3.4 Run the model

```
# RUN MODEL -------------------------------------------------------------------

# Define parallel computing
cl <- makeCluster(n_cores)
registerDoParallel(cl)

# Compute
Y.pawn <- foreach(i=1:nrow(A.pawn),
             .packages = "data.table") %dopar%
  {
```

```r
    model_pawn(epsilon = A.pawn[[i, "epsilon"]],
               N = A.pawn[[i, "N"]],
               n = A.pawn[[i, "n"]],
               theta = A.pawn[[i, "theta"]],
               Model = A.pawn[[i, "Model"]])
  }

# Stop parallel cluster
stopCluster(cl)
```

```r
# EXTRACT DATA ----------------------------------------------------------------

rowNumber <- lapply(1:4, function(x) A.pawn[, .I[Model == x]])
names(rowNumber) <- models

out <- list()
for(i in models) {
  out[[i]] <- Y.pawn[rowNumber[[i]]]
}

dt.models <- list()
for(i in seq_along(1:4)) {
  dt.models[[i]] <- cbind(A.pawn[Model == i], data.table(do.call(rbind, out[[i]])))
}
```

3.5 Uncertainty analysis

```r
# DATASET FOR UNCERTAINTY ANALYSIS --------------------------------------------

AB.pawn <- lapply(dt.models, function(x)  {
  x[, .SD[1: (2 * (2 ^ 12))], setting] %>%
    melt(., measure.vars = patterns("V"),
         variable.name = "parameter")
  }) %>%
  rbindlist() %>%
  .[, Model:= ifelse(Model == 1, models[1],
                     ifelse(Model == 2, models[2],
                            ifelse(Model == 3, models[3], models[4])))] %>%
  .[, parameter:= gsub("V", "X", parameter)] %>%
  .[, parameter:= factor(parameter,
                         levels = paste("X", 1:20, sep = ""))] %>%
  .[, Model:= factor(Model,
                     levels = c("Liu", "Ishigami", "Sobol' G", "Morris"))] %>%
  .[, setting:= ifelse(setting == "max", "$max \\in \\theta$",
                      ifelse(setting == "no.max", "$max \\notin \\theta$", "Optimum"))]
```

```r
# CHECK OVERLAP ----------------------------------------------------------

overlap.dt <- split(AB.pawn, AB.pawn$setting)

overlap.results <- mclapply(overlap.dt, function(x) {
  split(x, x$Model, drop = TRUE) %>%
    lapply(., function(x) split(x, x$parameter, drop = TRUE)) %>%
    lapply(., function(x) lapply(x, function(y) y[, value])) %>%
    lapply(., function(x) overlap(x))},
  mc.cores = n_cores)

tmp <- lapply(overlap.results, function(x) lapply(x, function(y) {
  cbind(y$OV) %>%
    data.frame() %>%
    setDT(., keep.rownames = TRUE)
  }))

pawn.overlap.results <- lapply(tmp, function(x)
  rbindlist(x, idcol = "Model")) %>%
  rbindlist(., idcol = "setting") %>%
  setnames(., ".", "overlap")

par.overlap <- paste("X", 1:6, sep = "")

final.overlap <- lapply(models, function(x) pawn.overlap.results[Model==x, .SD, setting]) %>%
  lapply(., function(x) x[, "overlap":= round(.SD, 3), .SDcols = "overlap"])

final.overlap
```

```
## [[1]]
##                   setting Model    rn overlap
## 1:    $max \\in \\theta$   Liu X1-X2   0.268
## 2: $max \\notin \\theta$   Liu X1-X2   0.126
## 3:                Optimum   Liu X1-X2   0.012
##
## [[2]]
##                   setting    Model    rn overlap
## 1:    $max \\in \\theta$ Ishigami X1-X2   0.009
## 2:    $max \\in \\theta$ Ishigami X1-X3   0.052
## 3:    $max \\in \\theta$ Ishigami X2-X3   0.094
## 4: $max \\notin \\theta$ Ishigami X1-X2   0.001
## 5: $max \\notin \\theta$ Ishigami X1-X3   0.016
## 6: $max \\notin \\theta$ Ishigami X2-X3   0.040
## 7:                Optimum Ishigami X1-X2   0.000
## 8:                Optimum Ishigami X1-X3   0.000
## 9:                Optimum Ishigami X2-X3   0.000
##
```

```
## [[3]]
##                       setting    Model    rn overlap
##  1:    $max \\in \\theta$ Sobol' G X1-X2   0.106
##  2:    $max \\in \\theta$ Sobol' G X1-X3   0.011
##  3:    $max \\in \\theta$ Sobol' G X1-X4   0.012
##  4:    $max \\in \\theta$ Sobol' G X1-X5   0.007
##  5:    $max \\in \\theta$ Sobol' G X1-X6   0.008
##  6:    $max \\in \\theta$ Sobol' G X1-X7   0.007
##  7:    $max \\in \\theta$ Sobol' G X1-X8   0.009
##  8:    $max \\in \\theta$ Sobol' G X2-X3   0.196
##  9:    $max \\in \\theta$ Sobol' G X2-X4   0.141
## 10:    $max \\in \\theta$ Sobol' G X2-X5   0.109
## 11:    $max \\in \\theta$ Sobol' G X2-X6   0.109
## 12:    $max \\in \\theta$ Sobol' G X2-X7   0.102
## 13:    $max \\in \\theta$ Sobol' G X2-X8   0.104
## 14:    $max \\in \\theta$ Sobol' G X3-X4   0.703
## 15:    $max \\in \\theta$ Sobol' G X3-X5   0.589
## 16:    $max \\in \\theta$ Sobol' G X3-X6   0.569
## 17:    $max \\in \\theta$ Sobol' G X3-X7   0.543
## 18:    $max \\in \\theta$ Sobol' G X3-X8   0.545
## 19:    $max \\in \\theta$ Sobol' G X4-X5   0.836
## 20:    $max \\in \\theta$ Sobol' G X4-X6   0.808
## 21:    $max \\in \\theta$ Sobol' G X4-X7   0.778
## 22:    $max \\in \\theta$ Sobol' G X4-X8   0.778
## 23:    $max \\in \\theta$ Sobol' G X5-X6   0.954
## 24:    $max \\in \\theta$ Sobol' G X5-X7   0.921
## 25:    $max \\in \\theta$ Sobol' G X5-X8   0.916
## 26:    $max \\in \\theta$ Sobol' G X6-X7   0.947
## 27:    $max \\in \\theta$ Sobol' G X6-X8   0.953
## 28:    $max \\in \\theta$ Sobol' G X7-X8   0.954
## 29: $max \\notin \\theta$ Sobol' G X1-X2   0.003
## 30: $max \\notin \\theta$ Sobol' G X1-X3   0.001
## 31: $max \\notin \\theta$ Sobol' G X1-X4   0.002
## 32: $max \\notin \\theta$ Sobol' G X1-X5   0.001
## 33: $max \\notin \\theta$ Sobol' G X1-X6   0.001
## 34: $max \\notin \\theta$ Sobol' G X1-X7   0.001
## 35: $max \\notin \\theta$ Sobol' G X1-X8   0.002
## 36: $max \\notin \\theta$ Sobol' G X2-X3   0.095
## 37: $max \\notin \\theta$ Sobol' G X2-X4   0.088
## 38: $max \\notin \\theta$ Sobol' G X2-X5   0.075
## 39: $max \\notin \\theta$ Sobol' G X2-X6   0.077
## 40: $max \\notin \\theta$ Sobol' G X2-X7   0.075
## 41: $max \\notin \\theta$ Sobol' G X2-X8   0.075
## 42: $max \\notin \\theta$ Sobol' G X3-X4   0.644
## 43: $max \\notin \\theta$ Sobol' G X3-X5   0.567
## 44: $max \\notin \\theta$ Sobol' G X3-X6   0.547
## 45: $max \\notin \\theta$ Sobol' G X3-X7   0.525
## 46: $max \\notin \\theta$ Sobol' G X3-X8   0.525
```

```
## 47: $max \\notin \\theta$ Sobol' G X4-X5   0.852
## 48: $max \\notin \\theta$ Sobol' G X4-X6   0.824
## 49: $max \\notin \\theta$ Sobol' G X4-X7   0.800
## 50: $max \\notin \\theta$ Sobol' G X4-X8   0.794
## 51: $max \\notin \\theta$ Sobol' G X5-X6   0.951
## 52: $max \\notin \\theta$ Sobol' G X5-X7   0.926
## 53: $max \\notin \\theta$ Sobol' G X5-X8   0.919
## 54: $max \\notin \\theta$ Sobol' G X6-X7   0.953
## 55: $max \\notin \\theta$ Sobol' G X6-X8   0.954
## 56: $max \\notin \\theta$ Sobol' G X7-X8   0.957
## 57:               Optimum Sobol' G X1-X2   0.000
## 58:               Optimum Sobol' G X1-X3   0.000
## 59:               Optimum Sobol' G X1-X4   0.000
## 60:               Optimum Sobol' G X1-X5   0.000
## 61:               Optimum Sobol' G X1-X6   0.000
## 62:               Optimum Sobol' G X1-X7   0.000
## 63:               Optimum Sobol' G X1-X8   0.000
## 64:               Optimum Sobol' G X2-X3   0.001
## 65:               Optimum Sobol' G X2-X4   0.001
## 66:               Optimum Sobol' G X2-X5   0.001
## 67:               Optimum Sobol' G X2-X6   0.001
## 68:               Optimum Sobol' G X2-X7   0.001
## 69:               Optimum Sobol' G X2-X8   0.001
## 70:               Optimum Sobol' G X3-X4   0.278
## 71:               Optimum Sobol' G X3-X5   0.190
## 72:               Optimum Sobol' G X3-X6   0.182
## 73:               Optimum Sobol' G X3-X7   0.183
## 74:               Optimum Sobol' G X3-X8   0.172
## 75:               Optimum Sobol' G X4-X5   0.545
## 76:               Optimum Sobol' G X4-X6   0.528
## 77:               Optimum Sobol' G X4-X7   0.520
## 78:               Optimum Sobol' G X4-X8   0.493
## 79:               Optimum Sobol' G X5-X6   0.956
## 80:               Optimum Sobol' G X5-X7   0.945
## 81:               Optimum Sobol' G X5-X8   0.907
## 82:               Optimum Sobol' G X6-X7   0.953
## 83:               Optimum Sobol' G X6-X8   0.922
## 84:               Optimum Sobol' G X7-X8   0.954
##                   setting    Model    rn overlap
##
## [[4]]
##                 setting  Model      rn overlap
##   1: $max \\in \\theta$ Morris   X1-X2   0.907
##   2: $max \\in \\theta$ Morris   X1-X3   0.937
##   3: $max \\in \\theta$ Morris   X1-X4   0.889
##   4: $max \\in \\theta$ Morris   X1-X5   0.888
##   5: $max \\in \\theta$ Morris   X1-X6   0.814
##  ---
```

```
## 566:             Optimum Morris X17-X19   0.880
## 567:             Optimum Morris X17-X20   0.963
## 568:             Optimum Morris X18-X19   0.751
## 569:             Optimum Morris X18-X20   0.679
## 570:             Optimum Morris X19-X20   0.888
```

```
lapply(par.overlap, function(x) final.overlap[[4]][rn %like% x])
```

```
## [[1]]
##                  setting  Model      rn overlap
##    1: $max \\in \\theta$ Morris   X1-X2   0.907
##    2: $max \\in \\theta$ Morris   X1-X3   0.937
##    3: $max \\in \\theta$ Morris   X1-X4   0.889
##    4: $max \\in \\theta$ Morris   X1-X5   0.888
##    5: $max \\in \\theta$ Morris   X1-X6   0.814
##   ---
## 458:             Optimum Morris X17-X19   0.880
## 459:             Optimum Morris X17-X20   0.963
## 460:             Optimum Morris X18-X19   0.751
## 461:             Optimum Morris X18-X20   0.679
## 462:             Optimum Morris X19-X20   0.888
##
## [[2]]
##                  setting  Model      rn overlap
##    1: $max \\in \\theta$ Morris   X1-X2   0.907
##    2: $max \\in \\theta$ Morris  X1-X20   0.776
##    3: $max \\in \\theta$ Morris   X2-X3   0.957
##    4: $max \\in \\theta$ Morris   X2-X4   0.936
##    5: $max \\in \\theta$ Morris   X2-X5   0.846
##   ---
## 107:             Optimum Morris X15-X20   0.955
## 108:             Optimum Morris X16-X20   0.972
## 109:             Optimum Morris X17-X20   0.963
## 110:             Optimum Morris X18-X20   0.679
## 111:             Optimum Morris X19-X20   0.888
##
## [[3]]
##                    setting  Model     rn overlap
##   1:    $max \\in \\theta$ Morris  X1-X3   0.937
##   2:    $max \\in \\theta$ Morris  X2-X3   0.957
##   3:    $max \\in \\theta$ Morris  X3-X4   0.928
##   4:    $max \\in \\theta$ Morris  X3-X5   0.871
##   5:    $max \\in \\theta$ Morris  X3-X6   0.854
##   6:    $max \\in \\theta$ Morris  X3-X7   0.643
##   7:    $max \\in \\theta$ Morris  X3-X8   0.236
##   8:    $max \\in \\theta$ Morris  X3-X9   0.224
##   9:    $max \\in \\theta$ Morris X3-X10   0.251
##  10:    $max \\in \\theta$ Morris X3-X11   0.900
```

```
## 11:    $max \\in \\theta$ Morris X3-X12   0.805
## 12:    $max \\in \\theta$ Morris X3-X13   0.877
## 13:    $max \\in \\theta$ Morris X3-X14   0.845
## 14:    $max \\in \\theta$ Morris X3-X15   0.805
## 15:    $max \\in \\theta$ Morris X3-X16   0.844
## 16:    $max \\in \\theta$ Morris X3-X17   0.841
## 17:    $max \\in \\theta$ Morris X3-X18   0.932
## 18:    $max \\in \\theta$ Morris X3-X19   0.819
## 19:    $max \\in \\theta$ Morris X3-X20   0.818
## 20: $max \\notin \\theta$ Morris  X1-X3   0.917
## 21: $max \\notin \\theta$ Morris  X2-X3   0.961
## 22: $max \\notin \\theta$ Morris  X3-X4   0.934
## 23: $max \\notin \\theta$ Morris  X3-X5   0.914
## 24: $max \\notin \\theta$ Morris  X3-X6   0.853
## 25: $max \\notin \\theta$ Morris  X3-X7   0.590
## 26: $max \\notin \\theta$ Morris  X3-X8   0.134
## 27: $max \\notin \\theta$ Morris  X3-X9   0.123
## 28: $max \\notin \\theta$ Morris X3-X10   0.160
## 29: $max \\notin \\theta$ Morris X3-X11   0.885
## 30: $max \\notin \\theta$ Morris X3-X12   0.803
## 31: $max \\notin \\theta$ Morris X3-X13   0.882
## 32: $max \\notin \\theta$ Morris X3-X14   0.826
## 33: $max \\notin \\theta$ Morris X3-X15   0.793
## 34: $max \\notin \\theta$ Morris X3-X16   0.828
## 35: $max \\notin \\theta$ Morris X3-X17   0.835
## 36: $max \\notin \\theta$ Morris X3-X18   0.909
## 37: $max \\notin \\theta$ Morris X3-X19   0.807
## 38: $max \\notin \\theta$ Morris X3-X20   0.809
## 39:               Optimum Morris  X1-X3   0.800
## 40:               Optimum Morris  X2-X3   0.726
## 41:               Optimum Morris  X3-X4   0.745
## 42:               Optimum Morris  X3-X5   0.783
## 43:               Optimum Morris  X3-X6   0.527
## 44:               Optimum Morris  X3-X7   0.247
## 45:               Optimum Morris  X3-X8   0.006
## 46:               Optimum Morris  X3-X9   0.008
## 47:               Optimum Morris X3-X10   0.007
## 48:               Optimum Morris X3-X11   0.701
## 49:               Optimum Morris X3-X12   0.444
## 50:               Optimum Morris X3-X13   0.626
## 51:               Optimum Morris X3-X14   0.456
## 52:               Optimum Morris X3-X15   0.426
## 53:               Optimum Morris X3-X16   0.444
## 54:               Optimum Morris X3-X17   0.450
## 55:               Optimum Morris X3-X18   0.595
## 56:               Optimum Morris X3-X19   0.478
## 57:               Optimum Morris X3-X20   0.442
##                   setting  Model     rn overlap
```

```
##
## [[4]]
##                  setting  Model     rn overlap
##  1:    $max \\in \\theta$ Morris  X1-X4   0.889
##  2:    $max \\in \\theta$ Morris  X2-X4   0.936
##  3:    $max \\in \\theta$ Morris  X3-X4   0.928
##  4:    $max \\in \\theta$ Morris  X4-X5   0.870
##  5:    $max \\in \\theta$ Morris  X4-X6   0.904
##  6:    $max \\in \\theta$ Morris  X4-X7   0.636
##  7:    $max \\in \\theta$ Morris  X4-X8   0.235
##  8:    $max \\in \\theta$ Morris  X4-X9   0.225
##  9:    $max \\in \\theta$ Morris X4-X10   0.251
## 10:    $max \\in \\theta$ Morris X4-X11   0.947
## 11:    $max \\in \\theta$ Morris X4-X12   0.853
## 12:    $max \\in \\theta$ Morris X4-X13   0.921
## 13:    $max \\in \\theta$ Morris X4-X14   0.876
## 14:    $max \\in \\theta$ Morris X4-X15   0.851
## 15:    $max \\in \\theta$ Morris X4-X16   0.890
## 16:    $max \\in \\theta$ Morris X4-X17   0.889
## 17:    $max \\in \\theta$ Morris X4-X18   0.964
## 18:    $max \\in \\theta$ Morris X4-X19   0.865
## 19:    $max \\in \\theta$ Morris X4-X20   0.863
## 20: $max \\notin \\theta$ Morris  X1-X4   0.870
## 21: $max \\notin \\theta$ Morris  X2-X4   0.943
## 22: $max \\notin \\theta$ Morris  X3-X4   0.934
## 23: $max \\notin \\theta$ Morris  X4-X5   0.867
## 24: $max \\notin \\theta$ Morris  X4-X6   0.893
## 25: $max \\notin \\theta$ Morris  X4-X7   0.574
## 26: $max \\notin \\theta$ Morris  X4-X8   0.134
## 27: $max \\notin \\theta$ Morris  X4-X9   0.124
## 28: $max \\notin \\theta$ Morris X4-X10   0.160
## 29: $max \\notin \\theta$ Morris X4-X11   0.929
## 30: $max \\notin \\theta$ Morris X4-X12   0.845
## 31: $max \\notin \\theta$ Morris X4-X13   0.923
## 32: $max \\notin \\theta$ Morris X4-X14   0.872
## 33: $max \\notin \\theta$ Morris X4-X15   0.835
## 34: $max \\notin \\theta$ Morris X4-X16   0.873
## 35: $max \\notin \\theta$ Morris X4-X17   0.876
## 36: $max \\notin \\theta$ Morris X4-X18   0.950
## 37: $max \\notin \\theta$ Morris X4-X19   0.850
## 38: $max \\notin \\theta$ Morris X4-X20   0.849
## 39:                Optimum Morris  X1-X4   0.893
## 40:                Optimum Morris  X2-X4   0.958
## 41:                Optimum Morris  X3-X4   0.745
## 42:                Optimum Morris  X4-X5   0.586
## 43:                Optimum Morris  X4-X6   0.646
## 44:                Optimum Morris  X4-X7   0.200
## 45:                Optimum Morris  X4-X8   0.005
```

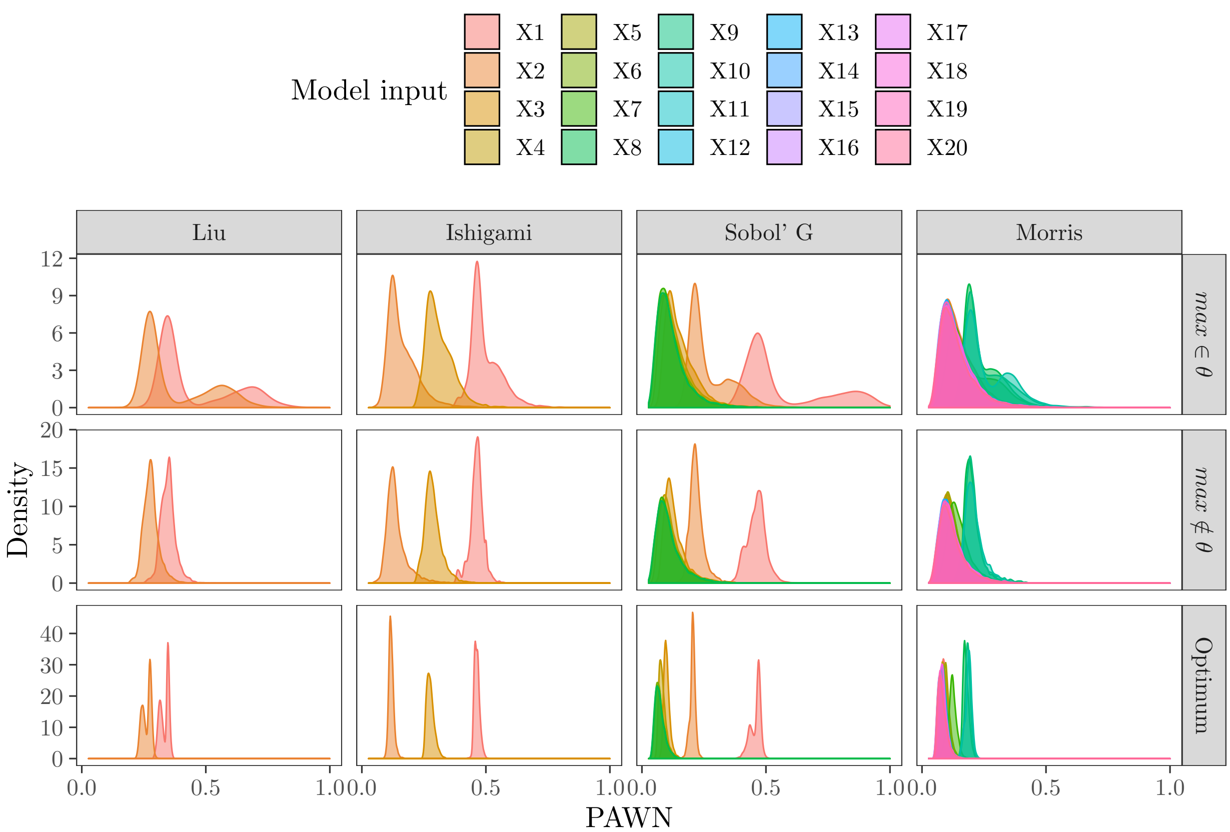
```
## 46:               Optimum Morris  X4-X9   0.007
## 47:               Optimum Morris X4-X10   0.006
## 48:               Optimum Morris X4-X11   0.905
## 49:               Optimum Morris X4-X12   0.541
## 50:               Optimum Morris X4-X13   0.801
## 51:               Optimum Morris X4-X14   0.556
## 52:               Optimum Morris X4-X15   0.515
## 53:               Optimum Morris X4-X16   0.541
## 54:               Optimum Morris X4-X17   0.545
## 55:               Optimum Morris X4-X18   0.779
## 56:               Optimum Morris X4-X19   0.592
## 57:               Optimum Morris X4-X20   0.536
##                   setting  Model     rn overlap
##
## [[5]]
##                   setting  Model     rn overlap
##  1:    $max \\in \\theta$ Morris  X1-X5   0.888
##  2:    $max \\in \\theta$ Morris  X2-X5   0.846
##  3:    $max \\in \\theta$ Morris  X3-X5   0.871
##  4:    $max \\in \\theta$ Morris  X4-X5   0.870
##  5:    $max \\in \\theta$ Morris  X5-X6   0.806
##  6:    $max \\in \\theta$ Morris  X5-X7   0.687
##  7:    $max \\in \\theta$ Morris  X5-X8   0.289
##  8:    $max \\in \\theta$ Morris  X5-X9   0.278
##  9:    $max \\in \\theta$ Morris X5-X10   0.307
## 10:    $max \\in \\theta$ Morris X5-X11   0.860
## 11:    $max \\in \\theta$ Morris X5-X12   0.753
## 12:    $max \\in \\theta$ Morris X5-X13   0.808
## 13:    $max \\in \\theta$ Morris X5-X14   0.771
## 14:    $max \\in \\theta$ Morris X5-X15   0.751
## 15:    $max \\in \\theta$ Morris X5-X16   0.794
## 16:    $max \\in \\theta$ Morris X5-X17   0.786
## 17:    $max \\in \\theta$ Morris X5-X18   0.867
## 18:    $max \\in \\theta$ Morris X5-X19   0.763
## 19:    $max \\in \\theta$ Morris X5-X20   0.768
## 20: $max \\notin \\theta$ Morris  X1-X5   0.950
## 21: $max \\notin \\theta$ Morris  X2-X5   0.909
## 22: $max \\notin \\theta$ Morris  X3-X5   0.914
## 23: $max \\notin \\theta$ Morris  X4-X5   0.867
## 24: $max \\notin \\theta$ Morris  X5-X6   0.794
## 25: $max \\notin \\theta$ Morris  X5-X7   0.632
## 26: $max \\notin \\theta$ Morris  X5-X8   0.137
## 27: $max \\notin \\theta$ Morris  X5-X9   0.126
## 28: $max \\notin \\theta$ Morris X5-X10   0.163
## 29: $max \\notin \\theta$ Morris X5-X11   0.833
## 30: $max \\notin \\theta$ Morris X5-X12   0.748
## 31: $max \\notin \\theta$ Morris X5-X13   0.823
## 32: $max \\notin \\theta$ Morris X5-X14   0.763
```

```
## 33: $max \\notin \\theta$ Morris X5-X15   0.740
## 34: $max \\notin \\theta$ Morris X5-X16   0.782
## 35: $max \\notin \\theta$ Morris X5-X17   0.784
## 36: $max \\notin \\theta$ Morris X5-X18   0.852
## 37: $max \\notin \\theta$ Morris X5-X19   0.752
## 38: $max \\notin \\theta$ Morris X5-X20   0.757
## 39:               Optimum Morris  X1-X5   0.640
## 40:               Optimum Morris  X2-X5   0.573
## 41:               Optimum Morris  X3-X5   0.783
## 42:               Optimum Morris  X4-X5   0.586
## 43:               Optimum Morris  X5-X6   0.419
## 44:               Optimum Morris  X5-X7   0.282
## 45:               Optimum Morris  X5-X8   0.006
## 46:               Optimum Morris  X5-X9   0.009
## 47:               Optimum Morris X5-X10   0.008
## 48:               Optimum Morris X5-X11   0.548
## 49:               Optimum Morris X5-X12   0.354
## 50:               Optimum Morris X5-X13   0.491
## 51:               Optimum Morris X5-X14   0.365
## 52:               Optimum Morris X5-X15   0.341
## 53:               Optimum Morris X5-X16   0.354
## 54:               Optimum Morris X5-X17   0.360
## 55:               Optimum Morris X5-X18   0.464
## 56:               Optimum Morris X5-X19   0.378
## 57:               Optimum Morris X5-X20   0.355
##                   setting  Model     rn overlap
##
## [[6]]
##                   setting  Model     rn overlap
##  1:    $max \\in \\theta$ Morris  X1-X6   0.814
##  2:    $max \\in \\theta$ Morris  X2-X6   0.865
##  3:    $max \\in \\theta$ Morris  X3-X6   0.854
##  4:    $max \\in \\theta$ Morris  X4-X6   0.904
##  5:    $max \\in \\theta$ Morris  X5-X6   0.806
##  6:    $max \\in \\theta$ Morris  X6-X7   0.587
##  7:    $max \\in \\theta$ Morris  X6-X8   0.213
##  8:    $max \\in \\theta$ Morris  X6-X9   0.204
##  9:    $max \\in \\theta$ Morris X6-X10   0.230
## 10:    $max \\in \\theta$ Morris X6-X11   0.930
## 11:    $max \\in \\theta$ Morris X6-X12   0.924
## 12:    $max \\in \\theta$ Morris X6-X13   0.936
## 13:    $max \\in \\theta$ Morris X6-X14   0.928
## 14:    $max \\in \\theta$ Morris X6-X15   0.931
## 15:    $max \\in \\theta$ Morris X6-X16   0.959
## 16:    $max \\in \\theta$ Morris X6-X17   0.969
## 17:    $max \\in \\theta$ Morris X6-X18   0.905
## 18:    $max \\in \\theta$ Morris X6-X19   0.931
## 19:    $max \\in \\theta$ Morris X6-X20   0.947
```

```
## 20: $max \\notin \\theta$ Morris  X1-X6   0.792
## 21: $max \\notin \\theta$ Morris  X2-X6   0.862
## 22: $max \\notin \\theta$ Morris  X3-X6   0.853
## 23: $max \\notin \\theta$ Morris  X4-X6   0.893
## 24: $max \\notin \\theta$ Morris  X5-X6   0.794
## 25: $max \\notin \\theta$ Morris  X6-X7   0.537
## 26: $max \\notin \\theta$ Morris  X6-X8   0.128
## 27: $max \\notin \\theta$ Morris  X6-X9   0.119
## 28: $max \\notin \\theta$ Morris X6-X10   0.153
## 29: $max \\notin \\theta$ Morris X6-X11   0.937
## 30: $max \\notin \\theta$ Morris X6-X12   0.932
## 31: $max \\notin \\theta$ Morris X6-X13   0.946
## 32: $max \\notin \\theta$ Morris X6-X14   0.940
## 33: $max \\notin \\theta$ Morris X6-X15   0.921
## 34: $max \\notin \\theta$ Morris X6-X16   0.943
## 35: $max \\notin \\theta$ Morris X6-X17   0.966
## 36: $max \\notin \\theta$ Morris X6-X18   0.913
## 37: $max \\notin \\theta$ Morris X6-X19   0.933
## 38: $max \\notin \\theta$ Morris X6-X20   0.943
## 39:                Optimum Morris  X1-X6   0.589
## 40:                Optimum Morris  X2-X6   0.663
## 41:                Optimum Morris  X3-X6   0.527
## 42:                Optimum Morris  X4-X6   0.646
## 43:                Optimum Morris  X5-X6   0.419
## 44:                Optimum Morris  X6-X7   0.170
## 45:                Optimum Morris  X6-X8   0.005
## 46:                Optimum Morris  X6-X9   0.006
## 47:                Optimum Morris X6-X10   0.006
## 48:                Optimum Morris X6-X11   0.713
## 49:                Optimum Morris X6-X12   0.843
## 50:                Optimum Morris X6-X13   0.795
## 51:                Optimum Morris X6-X14   0.862
## 52:                Optimum Morris X6-X15   0.808
## 53:                Optimum Morris X6-X16   0.840
## 54:                Optimum Morris X6-X17   0.848
## 55:                Optimum Morris X6-X18   0.797
## 56:                Optimum Morris X6-X19   0.896
## 57:                Optimum Morris X6-X20   0.836
##                    setting  Model     rn overlap
```

```r
# PLOT UNCERTAINTY ----------------------------------------------------------

plot.uncertainty.pawn <- ggplot(AB.pawn, aes(value,
                                             fill = parameter,
                                             color = parameter)) +
  geom_density(alpha = 0.5,
               position = "identity") +
  facet_grid(setting~Model,
```

```
              scales = "free_y") +
  scale_fill_discrete(name = "Model input") +
  scale_color_discrete(guide = FALSE) +
  labs(x = "PAWN",
       y = "Density") +
  scale_x_continuous(breaks = pretty_breaks(n = 3)) +
  theme_bw() +
  theme(legend.position = "top",
        legend.box = "horizontal",
        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))

plot.uncertainty.pawn
```




```
# EXPORT AB MATRIX FOR PAWN ---------------------------------------------------

fwrite(AB.pawn, "AB.pawn.csv")
```

3.6 Sensitivity analysis

```r
# DATASET FOR SENSITIVITY ANALYSIS ---------------------------------------------

dt.pawn.sens <- lapply(dt.models, function(x)
  melt(x, measure.vars = patterns("V"), variable.name = "model.input")) %>%
  rbindlist() %>%
  .[, Model:= ifelse(Model == 1, models[1],
                     ifelse(Model == 2, models[2],
                            ifelse(Model == 3, models[3], models[4])))] %>%
  .[, model.input:= gsub("V", "X", model.input)] %>%
  .[, model.input:= factor(model.input,
                           levels = paste("X", 1:20, sep = ""))] %>%
  .[, Model:= factor(Model,
                     levels = c("Liu", "Ishigami", "Sobol' G", "Morris"))] %>%
  setnames(., "value", "Y")

# EXPORT AB MATRIX FOR SENSITIVITY ---------------------------------------------

fwrite(dt.pawn.sens, "dt.pawn.sens.csv")
```

```r
# SENSITIVITY ANALYSIS --------------------------------------------------------

pawn.sensitivity <- dt.pawn.sens[, sobol_indices(Y,
                                                 params = parameters,
                                                 R = R,
                                                 n = 2 ^ 12,
                                                 parallel = "multicore",
                                                 second = TRUE,
                                                 third = TRUE,
                                                 ncpus = n_cores),
                                 .(setting, Model, model.input)]
```

```r
# CONFIDENCE INTERVALS --------------------------------------------------------

# Arrange data
tmp3 <- split(pawn.sensitivity, pawn.sensitivity$setting) %>%
  lapply(., function(x) split(x, x$Model)) %>%
  lapply(., function(x) lapply(x, function(y) split(y, y$model.input, drop = TRUE)))

# Compute confidence intervals
pawn.ci <- list()
for(i in names(tmp3)) {
  for(j in names(tmp3[[i]])) {
    for(k in names(tmp3[[i]][[j]])) {
      pawn.ci[[i]][[j]][[k]] <- sobol_ci(tmp3[[i]][[j]][[k]],
                                         params = parameters,
                                         type = type,
```

```r
                                                       conf = conf,
                                                       second = TRUE,
                                                       third = TRUE)
     }
   }
}

# Rearrange data
final.pawn.ci <- lapply(pawn.ci, function(x)
  lapply(x, function(y) rbindlist(y, idcol = "model.input"))) %>%
  lapply(., function(x) rbindlist(x, idcol = "model")) %>%
  rbindlist(., idcol = "setting") %>%
  .[, model:= factor(model, levels = c("Liu", "Ishigami",
                                       "Sobol' G", "Morris"))] %>%
  .[, model.input:= factor(model.input, levels = paste("X", 1:20, sep = ""))] %>%
  .[, parameters:= gsub("epsilon", "$\\\\varepsilon$", parameters)] %>%
  .[, parameters:= gsub("theta", "$\\\\theta$", parameters)] %>%
  .[, setting:= ifelse(setting == "max", "$max \\in \\theta$",
                       ifelse(setting == "no.max", "$max \\notin \\theta$", "Optimum"))]

# EXPORT DATA ---------------------------------------------------------------

fwrite(final.pawn.ci, "final.pawn.ci.csv")
```

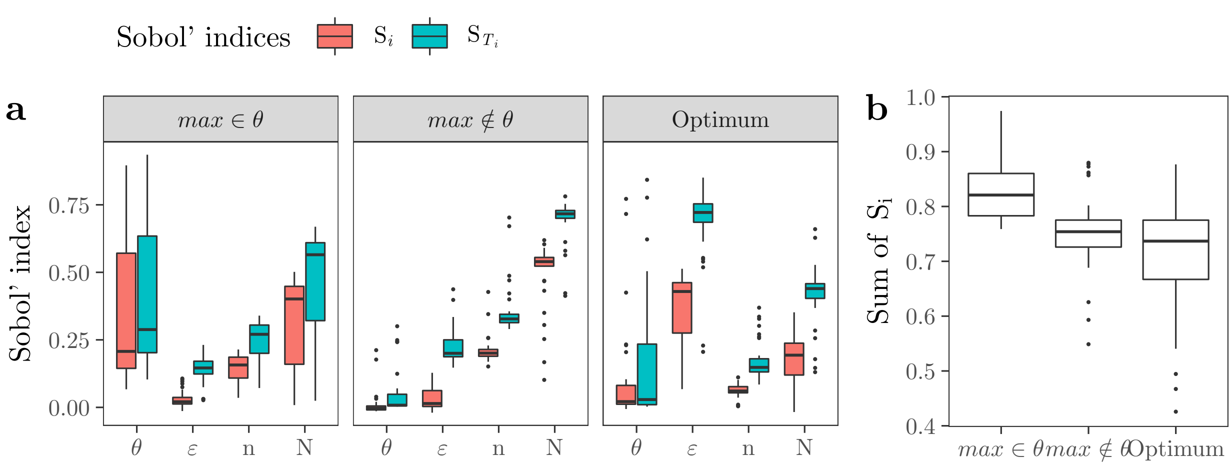
```r
# PLOT AGGREGATED SOBOL' INDICES ---------------------------------------------

a <- final.pawn.ci[sensitivity == "Si" | sensitivity == "STi"] %>%
  ggplot(., aes(parameters, original,
                fill = sensitivity)) +
  geom_boxplot(outlier.size = 0.2) +
  labs(x = "",
       y = "Sobol' index") +
  scale_fill_discrete(name = "Sobol' indices",
                      labels = c(expression(S[italic(i)]),
                                 expression(S[italic(T[i])]))) +
  theme_bw() +
  facet_wrap(~ setting) +
  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),
        legend.position = "none")

legend <- get_legend(a + theme(legend.position = "top"))
```

```r
# PLOT SUM OF SOBOL' SI ----------------------------------------------------

b <- final.pawn.ci[sensitivity == "Si"][
  , sum(original), .(setting, model, model.input)] %>%
  ggplot(., aes(setting, V1)) +
  geom_boxplot(outlier.size = 0.2) +
  labs(x = "",
       y = expression(paste("Sum of"~S[i]))) +
  theme_bw() +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

# MERGE AGGREGATED SOBOL' AND SUM OF SOBOL' ------------------------------------

up <- plot_grid(legend, NULL,
                ncol = 2)

bottom <- plot_grid(a, b,
                    ncol = 2,
                    align = "hv",
                    labels = "auto",
                    rel_widths = c(2.2, 1))
```

```
## Warning: Graphs cannot be vertically aligned unless the axis parameter is
## set. Placing graphs unaligned.
```

```
## Warning: Graphs cannot be horizontally aligned unless the axis parameter is
## set. Placing graphs unaligned.
```

```r
plot_grid(up, bottom,
          ncol = 1,
          align = "hv",
          rel_heights = c(0.21, 1))
```



4 Sensitivity of Sobol' indices to its design parameters

4.1 The model

```
# THE MODEL -------------------------------------------------------------------

# Functions to create A and AB matrices to compute Ti
scrambled_sobol <- function(A, B) {
  X <- rbind(A, B)
  for(i in 1:ncol(A)) {
    AB <- A
    AB[, i] <- B[, i]
    X <- rbind(X, AB)
  }
  AB <- rbind(A, X[((2*nrow(A)) + 1):nrow(X), ])
  return(AB)
}

sobol_matrix <- function(n, k) {
  df <- randtoolbox::sobol(n = n, dim = k * 2)
  A <- df[, 1:k]
  B <- df[, (k + 1) : (k * 2)]
  out <- scrambled_sobol(A = A, B = B)
  return(out)
}

# Functions to estimate Ti
sobol_all <- function(Y_A, Y_AB, type) {
  n <- length(Y_A[!is.na(Y_A)])
  f0 <- (1 / n) * sum(Y_A)
  VY <- 1 / n * sum((Y_A - f0) ^ 2)
  if(type == "jansen") {
    STi <- ((1 / (2 * n)) * sum((Y_A - Y_AB) ^ 2)) / VY
  } else if(type == "homma") {
    STi <- (VY - (1 / n) * sum(Y_A * Y_AB) + f0 ^ 2) / VY
  } else if(type == "sobol") {
    STi <- ((1 / n) * sum(Y_A * (Y_A - Y_AB))) / VY
  } else {
    stop("type should  be either jansen, sobol or homma")
  }
  return(STi)
}

sobol_Ti_Mapply <- function(d, type) {
  return(mapply(sobol_all,
                MoreArgs = list(type = type),
                d[, "Y_A"],
                d[, "Y_AB"]))
```

```r
}

sobol_Ti <- function(Y, params, type) {
  k <- length(params)
  p <- length(1:(length(Y) / (k + 1)))
  Y_A <- Y[1:p]
  Y_AB <- Y[(p + 1):length(Y)]
  parameters <- rep(params, each = length(Y_A))
  vec <- cbind(Y_A, Y_AB)
  out <- data.table(vec, parameters)
  output <- out[, sobol_Ti_Mapply(.SD, type = type), parameters][, V1]
  return(output)
}

# The model
model_sobol <- function(Model, N, k, Theta) {
  data <- sobol_matrix(n = N, k = k)
  if(Model == 1) {
    Y <- liu_Mapply(data)
  } else if(Model == 2) {
    Y <- sensobol::ishigami_Mapply(data)
  } else if(Model == 3) {
    Y <- sensobol::sobol_Fun(data)
  } else {
    Y <- sensitivity::morris.fun(data)
  }
  out <- sobol_Ti(Y, params = paste("X", 1:k, sep = ""), type = Theta)
  return(out)
}
```

4.2 Settings

```r
# DEFINE SETTINGS -------------------------------------------------------------

# Set parameters
parameters.sobol <- c("N", "Theta")
```

4.3 Sample matrix

```r
# CREATION OF THE MATRICES -----------------------------------------------------

# Create the A and AB matrices, also for the
# computation of second and third-order indices
tmp <- lapply(models, function(x)
  sobol_matrices(n = n, k = length(parameters.sobol)) %>%
    data.table())
```

```r
# Rename columns and transform distributions
A <- lapply(tmp, setnames, parameters.sobol) %>%
  rbindlist(., idcol = "Model")

# Create two copies of the sample matrix and list the
# original and the copies. One would be to run the
# calculations with uncertainty in N and Theta,
# the other with uncertainty in N only.
N.only <- copy(A)
A.DT <- list(A, N.only)
names(A.DT) <- c("N.Theta", "N")

A <- rbindlist(A.DT, idcol = "setting")

A.sobol <- A[, k:= ifelse(Model == 1, 2, ifelse(Model == 2, 3, ifelse(Model == 3, 8, 20)))][
    , N:= floor(qunif(N, N.min, N.max))][
    , Model:= as.numeric(Model)][
    , Theta:= floor(Theta * (3 - 1 + 1)) + 1][
    , Theta:= ifelse(Theta == 1, "jansen", ifelse(Theta == 2, "homma", "sobol"))][
    , Theta:= ifelse(setting == "N", "jansen", Theta)]

print(A.sobol)
```

```
##         setting Model    N  Theta  k
##      1: N.Theta     1 1100  homma  2
##      2: N.Theta     1 1550 jansen  2
##      3: N.Theta     1  650  sobol  2
##      4: N.Theta     1  875  homma  2
##      5: N.Theta     1 1775  sobol  2
##     ---
## 131068:       N     4  650 jansen 20
## 131069:       N     4 1550 jansen 20
## 131070:       N     4 1100 jansen 20
## 131071:       N     4  200 jansen 20
## 131072:       N     4  200 jansen 20
```

```r
print(n)
```

```
## [1] 4096
```

4.4 Run the model

```r
# RUN MODEL ---------------------------------------------------------------

# Define parallel computing
cl <- makeCluster(n_cores)
registerDoParallel(cl)
```

```r
# Compute
Y.sobol <- foreach(i=1:nrow(A.sobol),
                   .packages = "data.table") %dopar%
  {
    model_sobol(N = A.sobol[[i, "N"]],
                Theta = A.sobol[[i, "Theta"]],
                Model = A.sobol[[i, "Model"]],
                k = A.sobol[[i, "k"]])
  }

# Stop parallel cluster
stopCluster(cl)
```

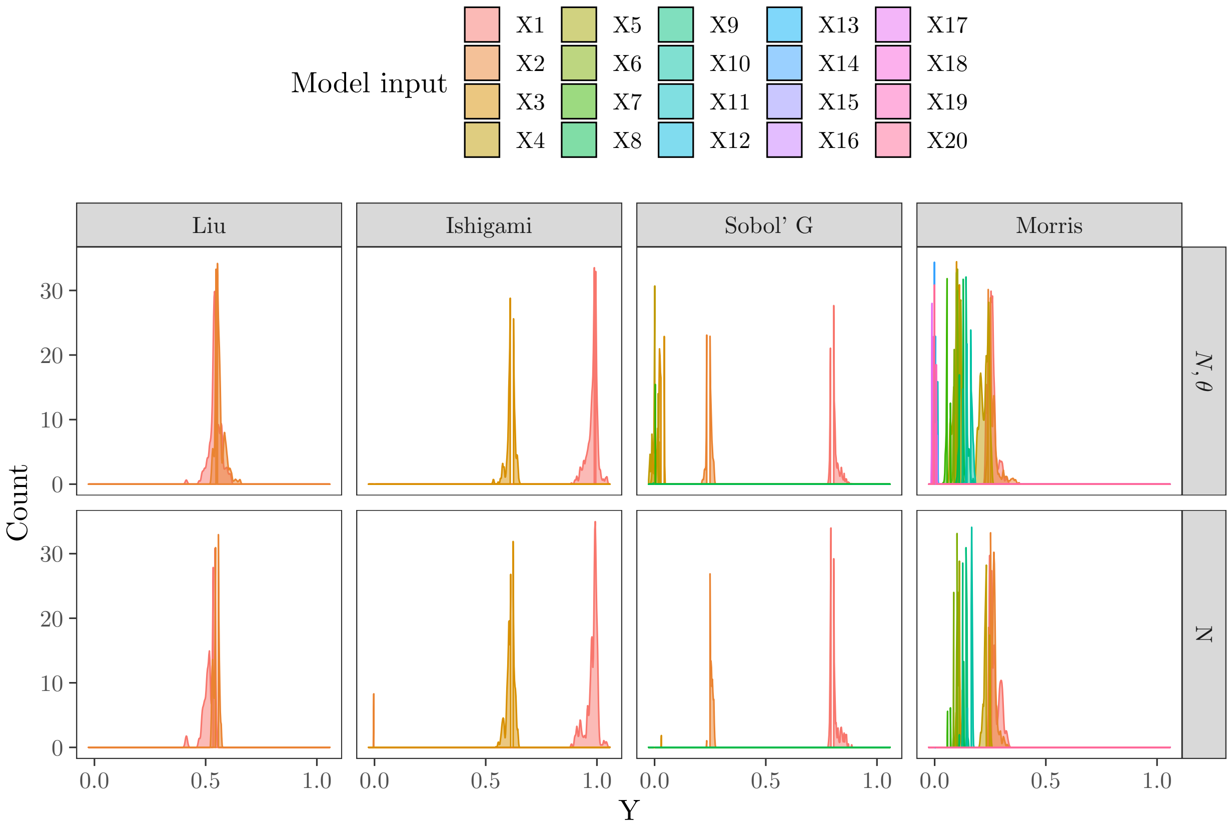
```r
# EXTRACT MODEL OUTPUT -----------------------------------------------------

rowNumber <- lapply(1:4, function(x) A.sobol[, .I[Model == x]])
names(rowNumber) <- models

out <- list()
for(i in models) {
  out[[i]] <- Y.sobol[rowNumber[[i]]]
}

dt.models <- list()
for(i in seq_along(1:4)) {
  dt.models[[i]] <- cbind(A[Model == i], data.table(do.call(rbind, out[[i]])))
}
```

4.5 Uncertainty analysis

```r
# DATASET FOR UNCERTAINTY ANALYSIS -------------------------------------------

AB.sobol <- lapply(dt.models, function(x) {
  x[, .SD[1: (2 * (2 ^ 12))], setting] %>%
  melt(., measure.vars = patterns("V"),
       variable.name = "parameter")}) %>%
  rbindlist(.) %>%
  .[, Model:= ifelse(Model == 1, models[1],
                     ifelse(Model == 2, models[2],
                            ifelse(Model == 3, models[3], models[4])))] %>%
  .[, k:= NULL] %>%
  .[, parameter:= gsub("V", "X", parameter)] %>%
  .[, parameter:= factor(parameter,
                         levels = paste("X", 1:20, sep = ""))] %>%
  .[, Model:= factor(Model,
                     levels = c("Liu", "Ishigami", "Sobol' G", "Morris"))] %>%
  .[, setting:= ifelse(setting == "N.Theta", "$N,\\theta$", setting)]
```

```r
# EXPORT AB MATRIX FOR SOBOL' --------------------------------------------------

fwrite(AB.sobol, "AB.sobol.csv")
```

```r
# PLOT UNCERTAINTY ------------------------------------------------------------

AB.sobol %>%
  ggplot(., aes(value,
                fill = parameter,
                color = parameter)) +
  geom_density(alpha = 0.5,
               position = "identity") +
  facet_grid(setting~Model) +
  scale_fill_discrete(name = "Model input") +
  scale_color_discrete(guide = FALSE) +
  labs(x = "Y",
       y = "Count") +
  scale_x_continuous(breaks = pretty_breaks(n = 3)) +
  scale_y_continuous(limits = c(0, 35)) +
  theme_bw() +
  theme(legend.position = "top",
        legend.box = "horizontal",
        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))
```




```r
# CHECK OVERLAP ----------------------------------------------------------

overlap.dt <- split(AB.sobol, AB.sobol$setting)

overlap.results <- mclapply(overlap.dt, function(x) {
  split(x, x$Model, drop = TRUE) %>%
    lapply(., function(x) split(x, x$parameter, drop = TRUE)) %>%
    lapply(., function(x) lapply(x, function(y) y[, value])) %>%
    lapply(., function(x) overlap(x))},
  mc.cores = n_cores)

tmp <- lapply(overlap.results, function(x) lapply(x, function(y) {
  cbind(y$OV) %>%
    data.frame() %>%
    setDT(., keep.rownames = TRUE)
}))

sobol.overlap.results <- lapply(tmp, function(x)
  rbindlist(x, idcol = "Model")) %>%
  rbindlist(., idcol = "setting") %>%
  setnames(., ".", "overlap")

par.overlap <- paste("X", 1:6, sep = "")
```

```r
final.overlap <- lapply(models, function(x) sobol.overlap.results[Model==x, .SD, setting]) %>%
  lapply(., function(x) x[, "overlap":= round(.SD, 3), .SDcols = "overlap"])

final.overlap
```

```
## [[1]]
##         setting Model    rn overlap
## 1: $N,\\theta$   Liu X1-X2   0.348
## 2:           N   Liu X1-X2   0.193
##
## [[2]]
##         setting    Model    rn overlap
## 1: $N,\\theta$ Ishigami X1-X2       0
## 2: $N,\\theta$ Ishigami X1-X3       0
## 3: $N,\\theta$ Ishigami X2-X3       0
## 4:           N Ishigami X1-X2       0
## 5:           N Ishigami X1-X3       0
## 6:           N Ishigami X2-X3       0
##
## [[3]]
##          setting    Model    rn overlap
##  1: $N,\\theta$ Sobol' G X1-X2   0.000
##  2: $N,\\theta$ Sobol' G X1-X3   0.000
##  3: $N,\\theta$ Sobol' G X1-X4   0.000
##  4: $N,\\theta$ Sobol' G X1-X5   0.000
##  5: $N,\\theta$ Sobol' G X1-X6   0.000
##  6: $N,\\theta$ Sobol' G X1-X7   0.000
##  7: $N,\\theta$ Sobol' G X1-X8   0.000
##  8: $N,\\theta$ Sobol' G X2-X3   0.000
##  9: $N,\\theta$ Sobol' G X2-X4   0.000
## 10: $N,\\theta$ Sobol' G X2-X5   0.000
## 11: $N,\\theta$ Sobol' G X2-X6   0.000
## 12: $N,\\theta$ Sobol' G X2-X7   0.000
## 13: $N,\\theta$ Sobol' G X2-X8   0.000
## 14: $N,\\theta$ Sobol' G X3-X4   0.035
## 15: $N,\\theta$ Sobol' G X3-X5   0.001
## 16: $N,\\theta$ Sobol' G X3-X6   0.002
## 17: $N,\\theta$ Sobol' G X3-X7   0.001
## 18: $N,\\theta$ Sobol' G X3-X8   0.001
## 19: $N,\\theta$ Sobol' G X4-X5   0.025
## 20: $N,\\theta$ Sobol' G X4-X6   0.037
## 21: $N,\\theta$ Sobol' G X4-X7   0.027
## 22: $N,\\theta$ Sobol' G X4-X8   0.026
## 23: $N,\\theta$ Sobol' G X5-X6   0.780
## 24: $N,\\theta$ Sobol' G X5-X7   0.208
## 25: $N,\\theta$ Sobol' G X5-X8   0.674
## 26: $N,\\theta$ Sobol' G X6-X7   0.260
```

```
## 27: $N,\\theta$ Sobol' G X6-X8   0.595
## 28: $N,\\theta$ Sobol' G X7-X8   0.227
## 29:           N Sobol' G X1-X2   0.000
## 30:           N Sobol' G X1-X3   0.000
## 31:           N Sobol' G X1-X4   0.000
## 32:           N Sobol' G X1-X5   0.000
## 33:           N Sobol' G X1-X6   0.000
## 34:           N Sobol' G X1-X7   0.000
## 35:           N Sobol' G X1-X8   0.000
## 36:           N Sobol' G X2-X3   0.000
## 37:           N Sobol' G X2-X4   0.000
## 38:           N Sobol' G X2-X5   0.000
## 39:           N Sobol' G X2-X6   0.000
## 40:           N Sobol' G X2-X7   0.000
## 41:           N Sobol' G X2-X8   0.000
## 42:           N Sobol' G X3-X4   0.000
## 43:           N Sobol' G X3-X5   0.000
## 44:           N Sobol' G X3-X6   0.000
## 45:           N Sobol' G X3-X7   0.000
## 46:           N Sobol' G X3-X8   0.000
## 47:           N Sobol' G X4-X5   0.000
## 48:           N Sobol' G X4-X6   0.000
## 49:           N Sobol' G X4-X7   0.000
## 50:           N Sobol' G X4-X8   0.000
## 51:           N Sobol' G X5-X6   0.374
## 52:           N Sobol' G X5-X7   0.615
## 53:           N Sobol' G X5-X8   0.685
## 54:           N Sobol' G X6-X7   0.270
## 55:           N Sobol' G X6-X8   0.264
## 56:           N Sobol' G X7-X8   0.695
##         setting    Model    rn overlap
##
## [[4]]
##          setting  Model      rn overlap
##   1: $N,\\theta$ Morris   X1-X2   0.458
##   2: $N,\\theta$ Morris   X1-X3   0.000
##   3: $N,\\theta$ Morris   X1-X4   0.292
##   4: $N,\\theta$ Morris   X1-X5   0.000
##   5: $N,\\theta$ Morris   X1-X6   0.000
##  ---
## 376:           N Morris X17-X19   0.063
## 377:           N Morris X17-X20   0.000
## 378:           N Morris X18-X19   0.028
## 379:           N Morris X18-X20   0.000
## 380:           N Morris X19-X20   0.000
```

```
lapply(par.overlap, function(x) final.overlap[[4]][rn %like% x])
```

```
## [[1]]
##          setting  Model      rn overlap
##   1: $N,\\theta$ Morris   X1-X2   0.458
##   2: $N,\\theta$ Morris   X1-X3   0.000
##   3: $N,\\theta$ Morris   X1-X4   0.292
##   4: $N,\\theta$ Morris   X1-X5   0.000
##   5: $N,\\theta$ Morris   X1-X6   0.000
##  ---
## 304:           N Morris X17-X19   0.063
## 305:           N Morris X17-X20   0.000
## 306:           N Morris X18-X19   0.028
## 307:           N Morris X18-X20   0.000
## 308:           N Morris X19-X20   0.000
##
## [[2]]
##         setting  Model      rn overlap
##  1: $N,\\theta$ Morris   X1-X2   0.458
##  2: $N,\\theta$ Morris  X1-X20   0.000
##  3: $N,\\theta$ Morris   X2-X3   0.000
##  4: $N,\\theta$ Morris   X2-X4   0.261
##  5: $N,\\theta$ Morris   X2-X5   0.000
##  6: $N,\\theta$ Morris   X2-X6   0.000
##  7: $N,\\theta$ Morris   X2-X7   0.000
##  8: $N,\\theta$ Morris   X2-X8   0.000
##  9: $N,\\theta$ Morris   X2-X9   0.000
## 10: $N,\\theta$ Morris  X2-X10   0.000
## 11: $N,\\theta$ Morris  X2-X11   0.000
## 12: $N,\\theta$ Morris  X2-X12   0.000
## 13: $N,\\theta$ Morris  X2-X13   0.000
## 14: $N,\\theta$ Morris  X2-X14   0.000
## 15: $N,\\theta$ Morris  X2-X15   0.000
## 16: $N,\\theta$ Morris  X2-X16   0.000
## 17: $N,\\theta$ Morris  X2-X17   0.000
## 18: $N,\\theta$ Morris  X2-X18   0.000
## 19: $N,\\theta$ Morris  X2-X19   0.000
## 20: $N,\\theta$ Morris  X2-X20   0.000
## 21: $N,\\theta$ Morris  X3-X20   0.000
## 22: $N,\\theta$ Morris  X4-X20   0.000
## 23: $N,\\theta$ Morris  X5-X20   0.000
## 24: $N,\\theta$ Morris  X6-X20   0.000
## 25: $N,\\theta$ Morris  X7-X20   0.000
## 26: $N,\\theta$ Morris  X8-X20   0.000
## 27: $N,\\theta$ Morris  X9-X20   0.000
## 28: $N,\\theta$ Morris X10-X20   0.000
## 29: $N,\\theta$ Morris X11-X20   0.076
## 30: $N,\\theta$ Morris X12-X20   0.224
## 31: $N,\\theta$ Morris X13-X20   0.003
## 32: $N,\\theta$ Morris X14-X20   0.565
```

```
## 33: $N,\\theta$ Morris X15-X20   0.355
## 34: $N,\\theta$ Morris X16-X20   0.289
## 35: $N,\\theta$ Morris X17-X20   0.298
## 36: $N,\\theta$ Morris X18-X20   0.261
## 37: $N,\\theta$ Morris X19-X20   0.394
## 38:           N Morris   X1-X2   0.554
## 39:           N Morris  X1-X20   0.000
## 40:           N Morris   X2-X3   0.000
## 41:           N Morris   X2-X4   0.131
## 42:           N Morris   X2-X5   0.000
## 43:           N Morris   X2-X6   0.000
## 44:           N Morris   X2-X7   0.000
## 45:           N Morris   X2-X8   0.000
## 46:           N Morris   X2-X9   0.000
## 47:           N Morris  X2-X10   0.000
## 48:           N Morris  X2-X11   0.000
## 49:           N Morris  X2-X12   0.000
## 50:           N Morris  X2-X13   0.000
## 51:           N Morris  X2-X14   0.000
## 52:           N Morris  X2-X15   0.000
## 53:           N Morris  X2-X16   0.000
## 54:           N Morris  X2-X17   0.000
## 55:           N Morris  X2-X18   0.000
## 56:           N Morris  X2-X19   0.000
## 57:           N Morris  X2-X20   0.000
## 58:           N Morris  X3-X20   0.000
## 59:           N Morris  X4-X20   0.000
## 60:           N Morris  X5-X20   0.000
## 61:           N Morris  X6-X20   0.000
## 62:           N Morris  X7-X20   0.000
## 63:           N Morris  X8-X20   0.000
## 64:           N Morris  X9-X20   0.000
## 65:           N Morris X10-X20   0.000
## 66:           N Morris X11-X20   0.000
## 67:           N Morris X12-X20   0.312
## 68:           N Morris X13-X20   0.000
## 69:           N Morris X14-X20   0.516
## 70:           N Morris X15-X20   0.235
## 71:           N Morris X16-X20   0.617
## 72:           N Morris X17-X20   0.000
## 73:           N Morris X18-X20   0.000
## 74:           N Morris X19-X20   0.000
##         setting  Model      rn overlap
##
## [[3]]
##         setting  Model     rn overlap
##  1: $N,\\theta$ Morris  X1-X3   0.000
##  2: $N,\\theta$ Morris  X2-X3   0.000
```

```
##  3: $N,\\theta$ Morris  X3-X4   0.000
##  4: $N,\\theta$ Morris  X3-X5   0.521
##  5: $N,\\theta$ Morris  X3-X6   0.189
##  6: $N,\\theta$ Morris  X3-X7   0.000
##  7: $N,\\theta$ Morris  X3-X8   0.153
##  8: $N,\\theta$ Morris  X3-X9   0.006
##  9: $N,\\theta$ Morris X3-X10   0.001
## 10: $N,\\theta$ Morris X3-X11   0.000
## 11: $N,\\theta$ Morris X3-X12   0.000
## 12: $N,\\theta$ Morris X3-X13   0.000
## 13: $N,\\theta$ Morris X3-X14   0.000
## 14: $N,\\theta$ Morris X3-X15   0.000
## 15: $N,\\theta$ Morris X3-X16   0.000
## 16: $N,\\theta$ Morris X3-X17   0.000
## 17: $N,\\theta$ Morris X3-X18   0.000
## 18: $N,\\theta$ Morris X3-X19   0.000
## 19: $N,\\theta$ Morris X3-X20   0.000
## 20:           N Morris  X1-X3   0.000
## 21:           N Morris  X2-X3   0.000
## 22:           N Morris  X3-X4   0.000
## 23:           N Morris  X3-X5   0.098
## 24:           N Morris  X3-X6   0.034
## 25:           N Morris  X3-X7   0.000
## 26:           N Morris  X3-X8   0.011
## 27:           N Morris  X3-X9   0.000
## 28:           N Morris X3-X10   0.000
## 29:           N Morris X3-X11   0.000
## 30:           N Morris X3-X12   0.000
## 31:           N Morris X3-X13   0.000
## 32:           N Morris X3-X14   0.000
## 33:           N Morris X3-X15   0.000
## 34:           N Morris X3-X16   0.000
## 35:           N Morris X3-X17   0.000
## 36:           N Morris X3-X18   0.000
## 37:           N Morris X3-X19   0.000
## 38:           N Morris X3-X20   0.000
##         setting  Model     rn overlap
##
## [[4]]
##         setting  Model     rn overlap
##  1: $N,\\theta$ Morris  X1-X4   0.292
##  2: $N,\\theta$ Morris  X2-X4   0.261
##  3: $N,\\theta$ Morris  X3-X4   0.000
##  4: $N,\\theta$ Morris  X4-X5   0.000
##  5: $N,\\theta$ Morris  X4-X6   0.000
##  6: $N,\\theta$ Morris  X4-X7   0.000
##  7: $N,\\theta$ Morris  X4-X8   0.000
##  8: $N,\\theta$ Morris  X4-X9   0.000
```

```
##  9: $N,\\theta$ Morris X4-X10   0.001
## 10: $N,\\theta$ Morris X4-X11   0.000
## 11: $N,\\theta$ Morris X4-X12   0.000
## 12: $N,\\theta$ Morris X4-X13   0.000
## 13: $N,\\theta$ Morris X4-X14   0.000
## 14: $N,\\theta$ Morris X4-X15   0.000
## 15: $N,\\theta$ Morris X4-X16   0.000
## 16: $N,\\theta$ Morris X4-X17   0.000
## 17: $N,\\theta$ Morris X4-X18   0.000
## 18: $N,\\theta$ Morris X4-X19   0.000
## 19: $N,\\theta$ Morris X4-X20   0.000
## 20:           N Morris  X1-X4   0.107
## 21:           N Morris  X2-X4   0.131
## 22:           N Morris  X3-X4   0.000
## 23:           N Morris  X4-X5   0.000
## 24:           N Morris  X4-X6   0.000
## 25:           N Morris  X4-X7   0.000
## 26:           N Morris  X4-X8   0.000
## 27:           N Morris  X4-X9   0.000
## 28:           N Morris X4-X10   0.000
## 29:           N Morris X4-X11   0.000
## 30:           N Morris X4-X12   0.000
## 31:           N Morris X4-X13   0.000
## 32:           N Morris X4-X14   0.000
## 33:           N Morris X4-X15   0.000
## 34:           N Morris X4-X16   0.000
## 35:           N Morris X4-X17   0.000
## 36:           N Morris X4-X18   0.000
## 37:           N Morris X4-X19   0.000
## 38:           N Morris X4-X20   0.000
##         setting  Model     rn overlap
##
## [[5]]
##         setting  Model     rn overlap
##  1: $N,\\theta$ Morris  X1-X5   0.000
##  2: $N,\\theta$ Morris  X2-X5   0.000
##  3: $N,\\theta$ Morris  X3-X5   0.521
##  4: $N,\\theta$ Morris  X4-X5   0.000
##  5: $N,\\theta$ Morris  X5-X6   0.260
##  6: $N,\\theta$ Morris  X5-X7   0.017
##  7: $N,\\theta$ Morris  X5-X8   0.176
##  8: $N,\\theta$ Morris  X5-X9   0.002
##  9: $N,\\theta$ Morris X5-X10   0.000
## 10: $N,\\theta$ Morris X5-X11   0.000
## 11: $N,\\theta$ Morris X5-X12   0.000
## 12: $N,\\theta$ Morris X5-X13   0.000
## 13: $N,\\theta$ Morris X5-X14   0.000
## 14: $N,\\theta$ Morris X5-X15   0.000
```

```
## 15: $N,\\theta$ Morris X5-X16   0.000
## 16: $N,\\theta$ Morris X5-X17   0.000
## 17: $N,\\theta$ Morris X5-X18   0.000
## 18: $N,\\theta$ Morris X5-X19   0.000
## 19: $N,\\theta$ Morris X5-X20   0.000
## 20:           N Morris  X1-X5   0.000
## 21:           N Morris  X2-X5   0.000
## 22:           N Morris  X3-X5   0.098
## 23:           N Morris  X4-X5   0.000
## 24:           N Morris  X5-X6   0.026
## 25:           N Morris  X5-X7   0.000
## 26:           N Morris  X5-X8   0.534
## 27:           N Morris  X5-X9   0.000
## 28:           N Morris X5-X10   0.000
## 29:           N Morris X5-X11   0.000
## 30:           N Morris X5-X12   0.000
## 31:           N Morris X5-X13   0.000
## 32:           N Morris X5-X14   0.000
## 33:           N Morris X5-X15   0.000
## 34:           N Morris X5-X16   0.000
## 35:           N Morris X5-X17   0.000
## 36:           N Morris X5-X18   0.000
## 37:           N Morris X5-X19   0.000
## 38:           N Morris X5-X20   0.000
##         setting  Model     rn overlap
##
## [[6]]
##         setting  Model     rn overlap
##  1: $N,\\theta$ Morris  X1-X6   0.000
##  2: $N,\\theta$ Morris  X2-X6   0.000
##  3: $N,\\theta$ Morris  X3-X6   0.189
##  4: $N,\\theta$ Morris  X4-X6   0.000
##  5: $N,\\theta$ Morris  X5-X6   0.260
##  6: $N,\\theta$ Morris  X6-X7   0.000
##  7: $N,\\theta$ Morris  X6-X8   0.015
##  8: $N,\\theta$ Morris  X6-X9   0.000
##  9: $N,\\theta$ Morris X6-X10   0.000
## 10: $N,\\theta$ Morris X6-X11   0.000
## 11: $N,\\theta$ Morris X6-X12   0.000
## 12: $N,\\theta$ Morris X6-X13   0.000
## 13: $N,\\theta$ Morris X6-X14   0.000
## 14: $N,\\theta$ Morris X6-X15   0.000
## 15: $N,\\theta$ Morris X6-X16   0.000
## 16: $N,\\theta$ Morris X6-X17   0.000
## 17: $N,\\theta$ Morris X6-X18   0.000
## 18: $N,\\theta$ Morris X6-X19   0.000
## 19: $N,\\theta$ Morris X6-X20   0.000
## 20:           N Morris  X1-X6   0.000
```

```
## 21:           N Morris  X2-X6   0.000
## 22:           N Morris  X3-X6   0.034
## 23:           N Morris  X4-X6   0.000
## 24:           N Morris  X5-X6   0.026
## 25:           N Morris  X6-X7   0.000
## 26:           N Morris  X6-X8   0.000
## 27:           N Morris  X6-X9   0.000
## 28:           N Morris X6-X10   0.000
## 29:           N Morris X6-X11   0.000
## 30:           N Morris X6-X12   0.000
## 31:           N Morris X6-X13   0.000
## 32:           N Morris X6-X14   0.000
## 33:           N Morris X6-X15   0.000
## 34:           N Morris X6-X16   0.000
## 35:           N Morris X6-X17   0.000
## 36:           N Morris X6-X18   0.000
## 37:           N Morris X6-X19   0.000
## 38:           N Morris X6-X20   0.000
##         setting  Model     rn overlap
```

4.6 Sensitivity analysis

```
# DATASET FOR SENSITIVITY ANALYSIS --------------------------------------------

full.dataset.sobol <- lapply(dt.models, function(x)
  melt(x, measure.vars = patterns("V"),
       variable.name = "parameter")) %>%
  rbindlist(.) %>%
  .[, Model:= ifelse(Model == 1, models[1],
                     ifelse(Model == 2, models[2],
                            ifelse(Model == 3, models[3], models[4])))] %>%
  .[, k:= NULL] %>%
  .[, parameter:= gsub("V", "X", parameter)] %>%
  .[, parameter:= factor(parameter,
                         levels = paste("X", 1:20, sep = ""))] %>%
  .[, Model:= factor(Model,
                     levels = c("Liu", "Ishigami", "Sobol' G", "Morris"))] %>%
  .[, setting:= ifelse(setting == "N.Theta", "$N,\\theta$", setting)]

# EXPORT SENSITIVITY MATRIX ---------------------------------------------------

fwrite(full.dataset.sobol, "full.dataset.sobol.csv")
```

```
# SENSITIVITY ANALYSIS --------------------------------------------------------

sobol.sensitivity <- full.dataset.sobol[, sobol_indices(value,
                                                        type = "jansen",
                                                        params = parameters.sobol,
```

```
                                                n = 2 ^ 12,
                                                R = R,
                                                parallel = "multicore",
                                                ncpus = n_cores),
                                     .(Model, parameter, setting)]
```

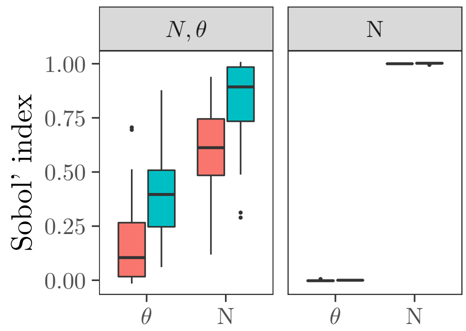
```
# CONFIDENCE INTERVALS ---------------------------------------------------------

# Arrange data
tmp3 <- split(sobol.sensitivity, sobol.sensitivity$setting) %>%
  lapply(., function(x) split(x, x$Model)) %>%
  lapply(., function(x) lapply(x, function(y) split(y, y$parameter, drop = TRUE)))

# Compute confidence intervals
out <- list()
for(i in names(tmp3)) {
  for(j in names(tmp3[[i]])) {
    for(k in names(tmp3[[i]][[j]])) {
      out[[i]][[j]][[k]] <- sobol_ci(tmp3[[i]][[j]][[k]],
                                      params = parameters.sobol,
                                      type = type,
                                      conf = conf)
    }
  }
}
```

```
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
```

```
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
## [1] "All values of t are equal to  1 \n Cannot calculate confidence intervals"
```

```
# ARRANGE DATA ----------------------------------------------------------------

final.sobol <- lapply(out, function(x)
  lapply(x, function(y) rbindlist(y, idcol = "model.input"))) %>%
  lapply(., function(x) rbindlist(x, idcol = "Model")) %>%
  rbindlist(., idcol = "setting") %>%
  .[, Model:= factor(Model, levels = c("Liu", "Ishigami", "Sobol' G", "Morris"))] %>%
  .[, model.input:= factor(model.input, levels = paste("X", 1:20, sep = ""))] %>%
  .[, parameters:= gsub("Theta", "$\\\\theta$", parameters)]

# EXPORT DATA -----------------------------------------------------------------

fwrite(final.sobol, "final.sobol.csv")
```

```
# PLOT SOBOL INDICES ----------------------------------------------------------

ggplot(final.sobol, aes(parameters, original,
                        fill = sensitivity)) +
  geom_boxplot(outlier.size = 0.2) +
  labs(x = "",
       y = "Sobol' index") +
  scale_fill_discrete(name = "Sobol' indices",
                      labels = c(expression(S[italic(i)]),
                                 expression(S[italic(T[i])]))) +
  theme_bw() +
  facet_wrap(~setting) +
  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),
        legend.position = "none")
```



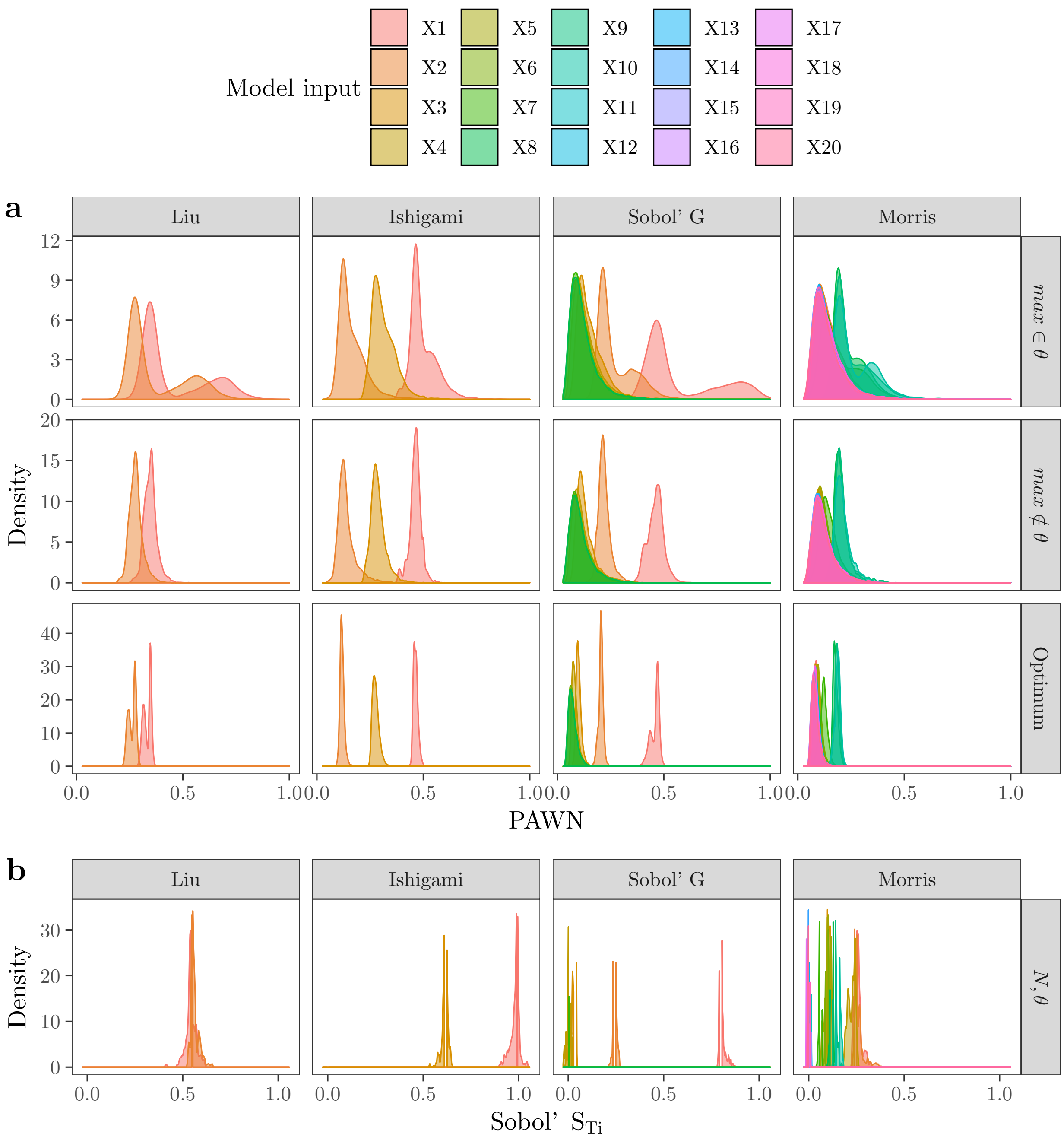
5 Extra plots

```
# MERGE UNCERTAINTY IN PAWN AND SOBOL'---------------------------------------

a <- plot.uncertainty.pawn +
  theme(legend.position = "none")

b <- AB.sobol[!setting == "N"] %>%
  ggplot(., aes(value,
                fill = parameter,
                color = parameter)) +
  geom_density(alpha = 0.5,
               position = "identity") +
  facet_grid(setting~Model) +
  scale_fill_discrete(name = "Model input") +
  scale_color_discrete(guide = FALSE) +
  labs(x = expression(paste("Sobol'"~S[Ti])),
       y = "Density") +
  scale_x_continuous(breaks = pretty_breaks(n = 3)) +
  scale_y_continuous(limits = c(0, 35)) +
  theme_bw() +
  theme(legend.position = "none",
        legend.box = "horizontal",
        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))

# Get legend
legend <- get_legend(a + theme(legend.position = "top"))

# Merge
bottom <- plot_grid(a, b,
                    ncol = 1,
                    labels = "auto",
```

```r
                     align = "h",
                     rel_heights = c(1, 0.46))
```

```
## Warning: Graphs cannot be horizontally aligned unless the axis parameter is
## set. Placing graphs unaligned.
```

```r
plot_grid(legend, bottom,
          labels = c("", ""),
          ncol = 1,
          align = "",
          rel_heights = c(0.2, 1))
```

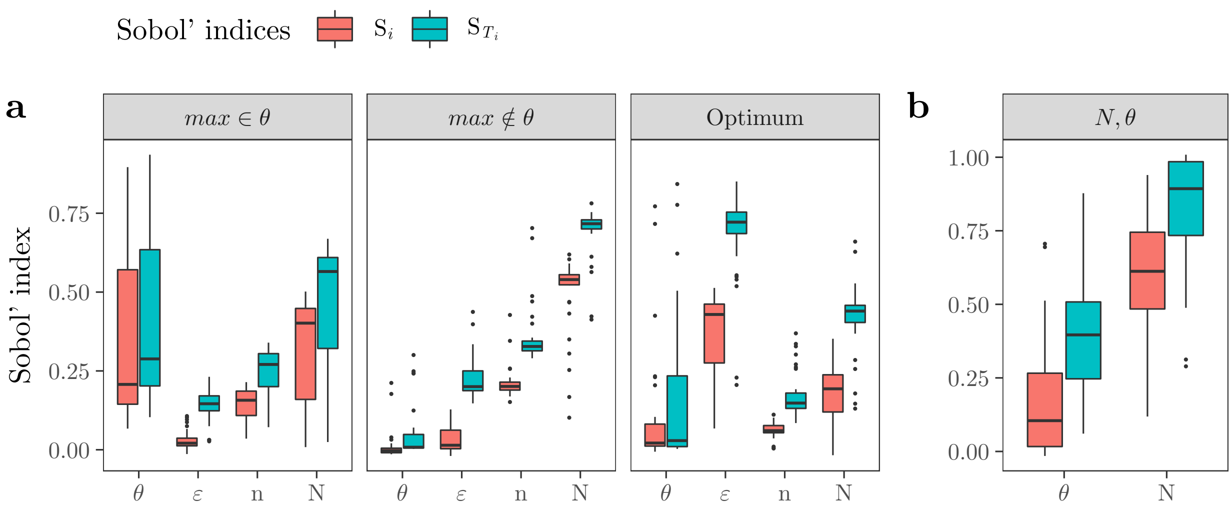


```r
# PLOT AGGREGATED SOBOL' INDICES ----------------------------------------------

a <- final.pawn.ci[sensitivity == "Si" | sensitivity == "STi"] %>%
  ggplot(., aes(parameters, original,
                fill = sensitivity)) +
  geom_boxplot(outlier.size = 0.2) +
  labs(x = "",
       y = "Sobol' index") +
  scale_fill_discrete(name = "Sobol' indices",
                      labels = c(expression(S[italic(i)]),
                                 expression(S[italic(T[i])]))) +
  theme_bw() +
  facet_wrap(~ setting) +
  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),
        legend.position = "none")

legend <- get_legend(a + theme(legend.position = "top"))


b <- final.sobol[!setting == "N"] %>%
  ggplot(., aes(parameters, original, fill = sensitivity)) +
  geom_boxplot(outlier.size = 0.2) +
  labs(x = "",
       y = "") +
  scale_fill_discrete(name = expression(paste("Sobol'"~T[i])),
                      labels = c(expression(S[italic(i)]),
                                 expression(S[italic(T[i])]))) +
  facet_wrap(~ setting) +
  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),
        legend.position = "none")

up <- plot_grid(legend, NULL,
                ncol = 2)

bottom <- plot_grid(a, b,
                    ncol = 2,
```

```r
                    align = "v",
                    labels = "auto",
                    rel_widths = c(2.58, 1))
```

```
## Warning: Graphs cannot be vertically aligned unless the axis parameter is
## set. Placing graphs unaligned.
```

```r
plot_grid(up, bottom,
          ncol = 1,
          align = "hv",
          rel_heights = c(0.21, 1))
```




```r
# PLOT AGGREGATED SUM OF SI ---------------------------------------------------

a <- final.pawn.ci[sensitivity == "Si"][
  , sum(original), .(setting, model, model.input)] %>%
  ggplot(., aes(setting, V1)) +
  geom_boxplot(outlier.size = 0.2) +
  labs(x = "",
       y = expression(paste("Sum of"~S[i]))) +
  theme_bw() +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())

b <- final.sobol[sensitivity == "Si" & !setting == "N"] %>%
  .[, sum(original), .(Model, model.input, setting)] %>%
  ggplot(., aes(setting, V1)) +
  geom_boxplot(outlier.size = 0.2) +
  labs(x = "",
       y = "") +
  theme_bw() +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.title.x=element_blank())
```

```
plot_grid(a, b,
          ncol = 2,
          labels = "auto",
          align = "hv",
          rel_widths = c(1, 0.4))
```